Nonlinear State Estimation Algorithms and their Applications

Thesis submitted for the degree of
Doctor of Philosophy
at the University of Leicester

by

Bharani Chandra Kumar Pakki

Department of Engineering
University of Leicester
Leicester, LE1 7RH, U.K.

November 2012

Abstract

Nonlinear State Estimation Algorithms and their Applications

Bharani Chandra Kumar Pakki

State estimation is a process of estimating the unmeasured or noisy states using the measured outputs and control inputs along with process and measurement models. The extended Kalman filter (EKF) has been an important approach for nonlinear state estimation over the last five decades. However, EKFs are only suitable for 'mild' nonlinearities where the first-order approximations of the nonlinear functions are available and they also require evaluation of state and measurement Jacobians at every iteration.

This thesis presents a few linear and nonlinear state estimation methods and their applications. To start with, we investigate the use of the linear H_∞ filter, which can deal with non-Gaussian noises, in a control application. The efficacy of the linear H_∞ filter based sliding mode controller is verified on a quadruple tank system. The main tools for nonlinear state estimation are cubature Kalman filter (CKF) and its variants. A solution to simultaneous localisation and mapping (SLAM) problem using CKF is proposed. The effectiveness of the nonlinear CKF-SLAM over EKF- and UKF-SLAM is demonstrated.

We propose a couple of new nonlinear state estimation algorithms, namely, cubature information filters (CIFs) and cubature H_∞ filters (CH_∞Fs), and their square root versions. The CIF is derived from an extended information filter and a CKF. The CIF is further extended for use in multi-sensor state estimation and its square root version is derived using a unitary transformation. For non-linear and non-Gaussian systems, we fuse an extended H_∞ filter and CKF to form CH_∞F which has the desirable features of both CKF and an extended H_∞ filter. Further, we derive a square root CH_∞F using a J-unitary transformation for numerical stability. The efficacies of the proposed algorithms are evaluated on simulation examples.

Acknowledgments

First of all I would like to thank Professor Da-Wei Gu for his time, valuable guidance, patience, kindness and humbleness. He was always ready to discuss both technical and non-technical issues. He has motivated me to explore different research areas ranging from robotics, control, estimation, etc. I am thankful to Professor Ian Postlethwaite for his help and support in improving the technicality and readability of this thesis. His comments and recommendations on my overall PhD research work was quite helpful. I thank GOD for providing enough energy and courage to overcome all the difficulties came across during the PhD.

My thanks goes to my lab members; Mangal, Paresh, Mirza, Rihan, Indira, Rosali, Peter and Xiaoxing, who made my life comfortable in Leicester. Mangal and I had several discussions on various topics, which were quite useful. I express my gratitude to Dr. Emmanuel Prempain for providing some crucial comments during APG transfer. Constructive comments during final viva from Professor Victor M. Becerra and Dr. Andrea Lecchini-Visintini were also helpful.

I acknowledge the help of Professor Bijnan Bandyopadhyay in the sliding mode control design for a quadruple-tank system. I would like to thank my previous and present flatmates of 3A Egginton street for their help, which definitely allowed me to focus more on my research; Dr. Srini Bandi and his family for the valuable time spent with us; UKIERI partners - Professor Debasish Ghose, Professor Radhakant Padhi and their families, and Dr. Ramakalyan Ayyagari for having several discussions at Leicester.

My thanks goes to my *gurus*, Dr. N. Ananthkrishnan, Dr. Gopaljee, Dr. G. R. Bindu , Professor P. S. Chandramohanan Nair and Mr. Narayan Bal, who helped me in improving my research skills at different stages of my life.

A special gratitude and love goes to my wife Sirisha, my son Sovish, my parents and all my family members for their unconditional love and support.

Finally I would like to thank UKIERI (United Kingdom - India Education Research Initiative) for funding this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Control systems design is necessary for almost all practical systems. The task of a control engineer is to design the controller to stabilise an unstable system and/or to achieve the desired performance in the presence of dynamic perturbations including parametric uncertainties, and disturbance and noise. In general, there are two categories of control systems, the open-loop and closed-loop control systems [1]. In an open-loop control system, the output has no effect on the control action. For a given input, the system gives a certain output. In the presence of disturbances or dynamic perturbations, the stability and tracking property of open-loop systems are not guaranteed. In open-loop systems, no measurements are made at the output for control purpose and hence it does not have the feedback mechanism. Closed-loop control systems are also known as feedback control systems. In a closed-loop system, the outputs are measured and compared with the reference signals to generate the errors. Based on these error signals, the controller generates the inputs to the system which help the outputs to reach their desirable values.

Any dynamical system can be represented by a set of state variables. If all the state variables are used to obtain the control signals, it is called as state feedback control system and if the controller is based on the measured outputs, then it is called as output feedback

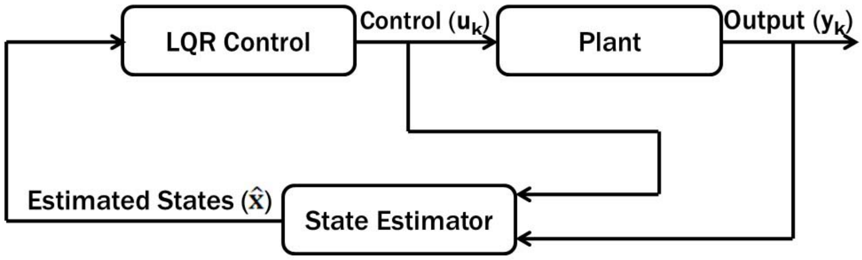

Figure 1.1: An example of a combined state estimation - control approach.

systems. In most of the real-world applications it is always not possible to access the complete state information due to the limitations on sensors and or cost consideration. If *some* of the sensors are very noisy or expensive or heavy, then it is not advisable to use them to measure the states. Rather, one can use the state estimation methods to obtain the unavailable states using the output information.

The control systems can be classified in to several ways like; linear and nonlinear, deterministic and stochastic, time-varying and time-invariant, lumped and distributive parameter systems, etc. These control systems can be designed based on the output or state information. State feedback controllers include pole placement control, linear quadratic regulator (LQR), dynamic inversion, etc. and output feedback controllers include proportional-integral-derivative (PID) control, linear quadratic Gaussian control, etc. Many controllers can be designed based on either state feedback or output feedback like sliding mode control, H_∞ control, model predictive control, etc. One can also note that; even if the complete state information is not available, the state feedback controllers can be designed using estimated states from state estimation methods. A combined state estimation - control approach is shown in Figure 1.1. If the state estimator in Figure 1.1 is the Kalman filter, then this approach reduces to linear quadratic Gaussian (LQG) control. A similar kind of approach has been explored in Chapter 2 using a sliding mode control and an H_∞ filter.

State estimation is a process of estimating the unmeasured states using the noisy outputs, control inputs along with process and measurement models. It has been an active

Wiener filter	Kalman filter
Mainly used for signal estimation.	Can be used for both signal and state estimation.
Both signals and processes noises should be stationary.	Kalman filter is a generalisation of Wiener filter for non-stationary signals.
Can be obtained by spectral factorisation methods.	Requires the solution of the matrix Riccati equation.
Basically, Wiener filter is a frequency domain approach.	Kalman filter is a time domain (state space) approach.

Table 1.1: Key differences between Wiener and Kalman filters [9].

research area for several decades. Similar to control systems, state estimation can also be classified as linear and nonlinear, deterministic and stochastic, etc. The earliest state estimation problem was considered in the field of astronomical studies by *Karl Friedrich Gauss* in 1795, where the planet and comet motion was studied using the telescopic measurements [2]. Gauss used the least square method as the estimation tool. After more than 140 years of Gauss' invention, *Andrey Nikolaevich Kolmogorov* [3] and *Norbert Wiener* [4] solved the linear least-square estimation problem for stochastic systems. Kolmogorov studied discrete least-estimation problems, whereas, Wiener studied the continuous-time problems [5]. Wiener filter[1] is a useful tool in signal processing and communication theory. But when it specially comes to the state estimation, Wiener filter is seldom used as it only deals with the stationary processes. *Rudolf Emil Kalman* extended the Wiener's work for more generic non-stationary processes in the path breaking paper [7]. The Wiener filter was developed in the frequency domain and is mainly used for signal estimation, whereas, the Kalman filter was developed in the time domain for state estimation. Key differences between Wiener and Kalman filters are given in Table 1.1.

As the main emphasis of this thesis is on the state estimation, the Wiener filter will not be further discussed.

The Kalman filter can be defined as "*an estimator used to estimate the state of a*

[1] In general, the term 'filter' is frequently used for state estimators in the estimation literature. This is due to Wiener, who studied the continuous-time estimation problem and noted that his algorithm can be implemented using a linear circuit. In circuit theory, the filters are used to separate the signals over different frequency ranges. Wiener's solution extended the classical theory of filter design to problems of obtaining the filtered signals from noisy measurements [6].

linear dynamic system perturbed by Gaussian white noise using measurements that are linear functions of the system state but corrupted by additive Gaussian white noise [8]". The Kalman filter and its variants are the main estimation tool for practical systems in the past several decades. The Kalman filter can be represented in an alternative form as the *information filter*, where the parameters of interest are the information states and the inverse of covariance matrix rather than states and covariance. Information filters are easier in initialisation compared to conventional Kalman filters and the update stage is computationally economic, and it can be easily extended for multi-sensor fusion; for more details please see [9, 10]. Both Kalman and information filters can be derived in the Gaussian framework and they need accurate process and measurement models. The early success of the Kalman filter in 1960s in aerospace applications is due to the availability of accurate system models, which are obtained after spending millions of dollars on the space program [11]. However, it is not worth to spend that huge amount of money in most other industrial applications to get an accurate model. One of the alternatives to the Kalman filter is to develop the estimator using the concepts of robust control. Several researchers have explored the robust control theory, specially an H_∞ theory, to develop robust state estimators [12, 13, 14, 15, 16, 17]. In H_∞ filters, the requirements on the accurate models or 'apriori' statistical noise properties can be relaxed to certain extent.

In real-time implementation of Kalman filters, the propagated error covariance matrices may become ill-conditioned, which eventually hinders the filter operation. This can happen if some of the states are measured with greater precision than others, where the elements of covariance matrix corresponding to accurately measured states will have lower values, while the other entries will have higher values. These types of ill-conditioned covariance matrices may cause numerical instability during the online implementation. To circumvent these difficulties, one can use square root Kalman filters, where the square root of the error covariance matrices are propagated. Some of the key properties of square root filters are symmetric positive definiteness of error covariances, availability of square root factors, doubled order precision, improved numerical accuracy, etc. [6, 23, 24, 25, 26].

Similar to square root Kalman filters, the information and H_∞ filters were also explored as square root information filters [27, 28, 29] and square root H_∞ filters [6, 30].

Initially the Kalman filter was developed for the linear systems. However, most of the real-world problems are nonlinear and hence the Kalman filter has been further extended for nonlinear systems. *Stanley F. Schmidt* was the first researcher to explore the Kalman filter for nonlinear systems, while doing so he developed the so called *extended Kalman filter* (EKF), see [31] for fascinating historical facts about the development of the EKF for practical applications. However, EKFs are only suitable for 'mild' nonlinearities where the first-order approximations of the nonlinear functions are available and they also require evaluation of state Jacobians at every iterations. To overcome some of the limitations of EKF, an unscented Kalman filter (UKF) has been proposed [32, 33], which is a derivative free filter. The UKF uses the deterministic sampling approach to capture the mean and covariances with sigma points and in general has been shown to perform better than EKF in nonlinear state estimation problems. The UKFs are further explored in information domain for decentralised estimation [34, 83, 36]. There are a few other nonlinear estimation techniques found in the literature, to name a few, Rao-Blackwellised particle filters [38], which are the improved version of particle filters [39], Gaussian filters [40], state dependent Riccati equation filters [41, 42], sliding mode observers [43], Fourier-Hermite Kalman filter [44], etc.

Recently, the cubature Kalman filter (CKF) [45] has been proposed for nonlinear state estimation. CKF is a Gaussian approximation of Bayesian filter, but provides a more accurate filtering estimate than existing Gaussian filters. In this thesis, we explore the CKF for multi-sensor state estimation and for non-Gaussian noises. The efficacy of the proposed methods are demonstrated on various simulation examples.

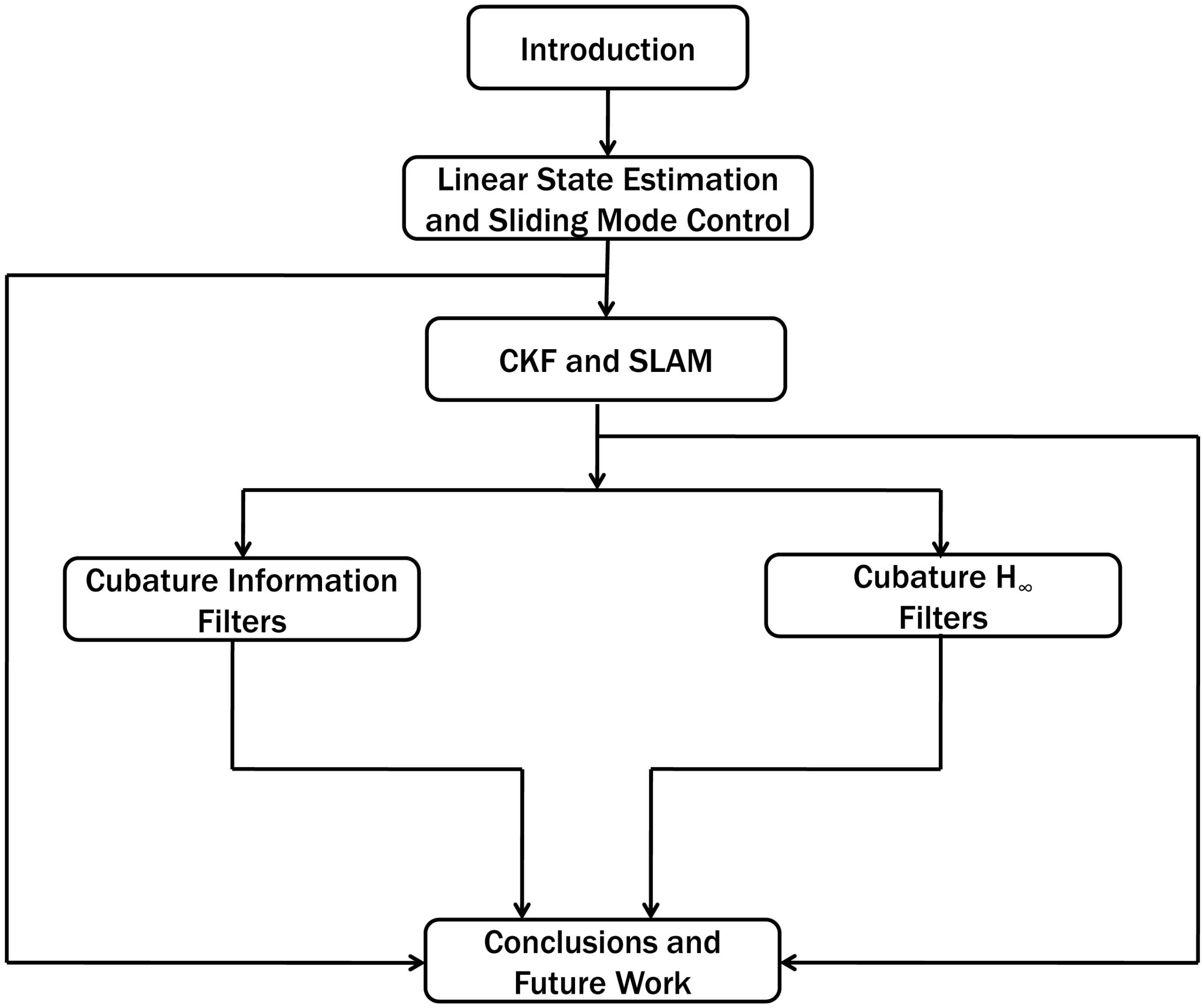

Figure 1.2: Flow-chart of the thesis.

1.2 Thesis Organisation and Contributions

A graphical representation of the thesis is shown in Figure 1.2. The thesis is organised in the following manner.

Chapter 2 begins with mathematical preliminaries of Kalman and H_∞ filters. The detailed derivation of Kalman filter and the game theory approach to the discrete H_∞ filter is briefly discussed. In this Chapter, we propose a combined use of sliding mode control (SMC) and an H_∞ filter for a quadruple-tank system. It is assumed that, out of the four states only two states are available. The complete state vector is estimated using an H_∞ filter and the SMC is designed based on the estimated states. The proposed H_∞ filter based SMC can be easily extended for other practical systems.

Chapter 3 discusses the nonlinear state estimation methods like EKF, UKF and, most focussed on CKF. A solution to the Simultaneous Localisation and Mapping (SLAM) using CKF is presented. Different simulations are performed to compare EKF-, UKF- and CKF-SLAM.

In Chapter 4, the cubature information filter (CIF) is first derived from an extended information filter and a CKF. The CIF is then extended to multi-sensor state estimation, where the data from various nonlinear sensors are fused. For numerical accuracy, square root cubature information filter is further developed for the single sensor as well as multi-sensor cases. The efficacy of the multi-sensor square root CIF is validated on a permanent magnet synchronous motor example.

Chapter 5 deals with the fusion of an extended H_∞ filter and CKF to form a cubature H_∞ filter (CH_∞F). The CH_∞F is derived for state estimation of nonlinear systems with general noises; not limited to Gaussian noises. The square root CH_∞F is then derived using the J-unitary transformation. The effectiveness of the square root CH_∞F is demonstrated on a continuous stirred tank reactor problem. The combined control and estimation problem is considered and a number of simulations are performed to verify the efficacy of the square root CH_∞F in the presence of Gaussian and non-Gaussian noises.

Finally, the concluding remarks on the proposed methods and a detailed future work scheme are presented in Chapter 6.

1.3 List of Publications

1. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, “Square−root cubature information filter,” IEEE Sensors Journal, vol. 13, no. 2, pp. 750−758, 2013.

2. A. Bajodah, H. M. Tariq, K. P. Bharani Chandra, R. Ahmed and D.−W Gu, “Fault tolerant control of aircraft actuating surfaces using generalized DI and integral SM Control,” Journal of Intelligent and Robotic Systems, vol. 69, pp. 181−188, 2013.

3. K. P. Bharani Chandra and D.−W Gu, "Nonlinear state estimation algorithms in aerospace control systems," Book chapter in Computational Intelligence in Aerospace Sciences, (Editors: M. Vasile and V. M. Becerra), AIAA, 2014.

4. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "SLAM using EKF, EH_∞ and mixed EH_2/H_∞ filter," in Proceedings of IEEE International Symposium on Intelligent Control (Multi−Conference on Systems and Control), Yokohama, Japan, Sept., 2010.

5. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature information filter and its applications," in Proceedings of IEEE American Control Conference, California, July, 2011.

6. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Fusion of an extended H_∞ filter and cubature Kalman filter," in Proceedings of 18th IFAC World Congress, Italy, Sept., 2011.

7. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature Kalman filter based localization and mapping," in Proceedings of 18th IFAC World Congress, Italy, Sept., 2011.

8. K. P. Bharani Chandra, H. Srikanthan, D.−W Gu, B. Bandyopadhyay and I. Postlethwaite, "Discrete−time sliding mode control using an H_∞ filter for a quadruple tank system," in Proceedings of 12th IEEE workshop on Variable structure systems (VSS), Bombay, India, Jan., 2012.

9. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Nonlinear state estimation for induction and permanent magnet synchronous motors," in Proceedings of IEEE International workshop on Electronics Machine, Power Electronics and Engineering, Lushan, China, April, 2012.

10. K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature H_∞ information filter," European Control Conference, Zurich, Switzerland, July 2013.

Another couple of papers have been submitted for possible publication in journals.

Chapter 2

Linear State Estimation and its Application in Control Theory

2.1 Introduction

State estimation is the process of estimating the state vector using uncertain and inaccurate measurements along with the control inputs, process and measurement models. It has been an active research area for several decades and plays a key role in practical applications. Real-life applications of the state estimation include state and parameter estimation in chemical process plants, electrical machinery, data assimilation, econometrics, fault detection and isolation, control system design, etc., where either sensors are noisy or it is difficult to measure the states. Linear state estimation deals with the state estimation of linear systems and linear measurement models. Although, all the practical systems have nonlinearities in process and measurement models; it is worth to explore some of the linear state estimation methods, which can be easily extended to nonlinear systems. In this chapter, we will consider the description on the Kalman and H_∞ filters. These two methods are the most relevant linear estimation methods required for further chapters. The usage of state estimation in a control application is also explored in this chapter. The control mechanism used to control the heights of the quadruple-tank system is sliding mode control. The combined control and estimation for a quadruple tank is considered. It is assumed that only two states of the quadruple-tank are available and the complete state vector is estimated using the state estimation methods. The estimated states from the filters are then used for sliding mode control.

The rest of this chapter is structured as follows. Section 2.2 introduces the discrete-time Kalman filter. In particular, Section 2.2.1 details the process and measurement models, Section 2.2.2 derives the Kalman filter and the basic equations of the Kalman filter are summarised in Algorithm 1. Section 2.3 deals with the H_∞ filter and is summarised in Algorithm 2. Section 2.4 examines the combined state estimation and sliding mode control for a quadruple-tank system, detailed simulations in the presence of Gaussian and non-Gaussian noises are presented in Section 2.5 and this chapter is concluded in Section 2.6.

2.2 The Discrete-Time Kalman Filter

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) way to estimate the state vector of a system. In this chapter, Kalman filter is described in discrete-time.

2.2.1 Process and Measurement Models

Consider the discrete linear process and measurement models as

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \tag{2.1}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{2.2}$$

where $k \geq 1$ is the time index, $\mathbf{x}_k$ is the state vector, $\mathbf{u}_k$ is the control input, $\mathbf{z}_k$ is the measurement, $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the process and measurement noises, respectively.

The process and measurement noises are assumed to be a Gaussian-distributed[1] random variables, which have zero means and covariances of $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$,

$$\mathbf{w}_{k-1} = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$$
$$\mathbf{v}_k = \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

where $\mathcal{N}$ represents the Gaussian or normal probability distribution.

The process and measurement noises at any time are assumed to be independent of the state of the system

$$\mathbb{E}\left[\mathbf{w}_i \mathbf{x}_j^T\right] = \mathbf{0}, \quad \mathbb{E}\left[\mathbf{w}_i \mathbf{w}_j^T\right] = \mathbf{Q}_i \delta_{ij} \quad \forall \quad i, j$$

and

$$\mathbb{E}\left[\mathbf{v}_i \mathbf{x}_j^T\right] = \mathbf{0}, \quad \mathbb{E}\left[\mathbf{v}_i \mathbf{v}_j^T\right] = \mathbf{R}_i \delta_{ij} \quad \forall \quad i, j$$

where δ_{ij} is the Dirac function and $\mathbb{E}\left[\cdot\right]$ is the expectation operator.

2.2.2 Derivation of the Kalman filter

The derivation in this section closely follows the one given in [50] and [51]. The Kalman filter gives an estimate, $\hat{\mathbf{x}}_{i|j}$, which minimises the mean-squared estimation error conditioned on the measurements sequence, $\mathbf{Z}_j = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_j]$. The estimated state is the expected value of state conditioned on the measurements sequence and is given by

$$\hat{\mathbf{x}}_{i|j} = \mathbb{E}[\mathbf{x}_i|\mathbf{Z}_j]. \tag{2.3}$$

[1]The Kalman's original derivation did not use the Baye's rule and does not require the exploitation of any specific error distribution information. The Kalman filter is the minimum variance estimator if the noise is Gaussian, and it is the ***linear*** minimum variance estimator for linear systems with non-Gaussian noises [7, 11, 32, 46].

The error between the actual and estimated states is

$$\mathbf{x}_{e,i|j} = \mathbf{x}_i - \hat{\mathbf{x}}_{i|j} \tag{2.4}$$

and the covariance of $\hat{\mathbf{x}}_{i|j}$ is

$$\mathbf{P}_{i|j} = \mathbb{E}[\mathbf{x}_{e,i|j}\mathbf{x}_{e,i|j}^T|\mathbf{Z}_j]. \tag{2.5}$$

The Kalman filter is a recursive algorithm consisting of prediction and update stages. In the prediction stage, the state and the covariance at k^{th} instant are predicted based on the information at $(k-1)^{th}$ instant. Once the measurement is obtained at k^{th} instant, the predicted state and covariance are used to form the updated state estimate and updated covariance at k^{th} instant. This process then repeats recursively.

The predicted state can be obtained by taking the expectation of the state model conditioned on measurements up to $(k-1)^{th}$ instant and is given by

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}[\mathbf{x}_k|\mathbf{Z}_{k-1}] \\
&= \mathbb{E}[(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1})|\mathbf{Z}_{k-1}] \\
&= \mathbf{F}_{k-1}\mathbb{E}[\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}] + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbb{E}[\mathbf{w}_{k-1}|\mathbf{Z}_{k-1}] \\
&= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}.
\end{aligned} \tag{2.6}$$

Similarly, the predicted covariance at k^{th} instant based on $(k-1)^{th}$ can be found as

$$\begin{aligned}
\mathbf{P}_{k|k-1} &= \mathbb{E}[(\mathbf{x}_{e,k|k-1}\mathbf{x}_{e,k|k-1}^T)|\mathbf{Z}_{k-1}] \\
&= \mathbb{E}[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})^T|\mathbf{Z}_{k-1}] \\
&= \mathbb{E}[(\mathbf{F}_{k-1}\mathbf{x}_{k-1} - \mathbf{F}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{w}_{k-1})(\mathbf{F}_{k-1}\mathbf{x}_{k-1} - \mathbf{F}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{w}_{k-1})^T|\mathbf{Z}_{k-1}] \\
&= \mathbf{F}_{k-1}\mathbb{E}[(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1})^T|\mathbf{Z}_{k-1}]\mathbf{F}_{k-1}^T + \mathbb{E}[(\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T)|\mathbf{Z}_{k-1}] \\
&= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}.
\end{aligned} \tag{2.7}$$

The predicted measurement and the innovation vector, which is the difference between actual measurement and the predicted measurement, can be obtained as

$$\begin{aligned}
\widehat{\mathbf{z}}_{k|k-1} &= \mathbb{E}[\mathbf{z}_k|\mathbf{Z}_{k-1}] \\
&= \mathbb{E}\left[\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k|\mathbf{Z}_{k-1}\right] \\
&= \mathbf{H}_k(\mathbb{E}[\mathbf{x}_k|\mathbf{Z}_{k-1}] + \mathbb{E}[\mathbf{v}_k|\mathbf{Z}_{k-1}] \\
&= \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1} \qquad (2.8)
\end{aligned}$$

and

$$\begin{aligned}
\nu_k &= \mathbf{z}_k - \widehat{\mathbf{z}}_{k|k-1} \\
&= \mathbf{z}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}. \qquad (2.9)
\end{aligned}$$

After obtaining the predicted state and covariance, the next task is to obtain the updated state and covariance for the recursive process. The updated state vector can be obtained from the predicted state and the innovation vector and is given as

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\nu_k \qquad (2.10)$$

where, $\mathbf{K}_k$ is the Kalman gain, which dictates the influence of the innovation on the updated state vector. The error between the actual and updated states are given by

$$\begin{aligned}
\mathbf{x}_{e,k|k} &= \mathbf{x}_k - \widehat{\mathbf{x}}_{k|k} \\
&= \mathbf{x}_k - [\widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\nu_k] \\
&= \mathbf{x}_{e,k|k-1} - \mathbf{K}_k\nu_k \\
&= \mathbf{x}_{e,k|k-1} - \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}) \\
&= \mathbf{x}_{e,k|k-1} - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}) \\
&= \mathbf{x}_{e,k|k-1} - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_{e,k|k-1} + \mathbf{v}_k) \\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{x}_{e,k|k-1} - \mathbf{K}_k\mathbf{v}_k \qquad (2.11)
\end{aligned}$$

where, $\mathbf{I}$ is the identity matrix of an appropriate size.

The covariance update can be written as

$$\begin{aligned}
\mathbf{P}_{k|k} &= \mathbb{E}[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})^T|\mathbf{Z}_k] \\
&= \mathbb{E}[\mathbf{x}_{e,k|k}\mathbf{x}_{e,k|k}^T|\mathbf{Z}_k] \\
&= \mathbb{E}[\{(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{x}_{e,k|k-1} - \mathbf{K}_k\mathbf{v}_k\}\{(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{x}_{e,k|k-1} - \mathbf{K}_k\mathbf{v}_k\}^T|\mathbf{Z}_k] \\
&= \mathbb{E}[(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{x}_{e,k|k-1}\mathbf{x}_{e,k|k-1}^T(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^T - (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{x}_{e,k|k-1}\mathbf{v}_k^T\mathbf{K}_k^T \\
&\quad -\mathbf{K}_k\mathbf{v}_k\mathbf{x}_{e,k|k-1}^T(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{v}_k\mathbf{v}_k^T\mathbf{K}_k^T|\mathbf{Z}_k] \\
&= (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbb{E}[\mathbf{x}_{e,k|k-1}\mathbf{x}_{e,k|k-1}^T](\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^T - (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbb{E}[\mathbf{x}_{e,k|k-1}\mathbf{v}_k^T|\mathbf{Z}_k]\mathbf{K}_k^T \\
&\quad -\mathbf{K}_k\mathbb{E}[\mathbf{v}_k\mathbf{x}_{e,k|k-1}^T|\mathbf{Z}_k](\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbb{E}[\mathbf{v}_k\mathbf{v}_k^T|\mathbf{Z}_k]\mathbf{K}_k^T \\
&= (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\
&= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T. \qquad (2.12)
\end{aligned}$$

By taking the trace[2] on both sides of Eq. (2.12) yields

$$Tr(\mathbf{P}_{k|k}) = Tr(\mathbf{P}_{k|k-1}) - 2Tr(\mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1}) + Tr(\mathbf{K}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T)\mathbf{K}_k^T) + Tr(\mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T) \qquad (2.13)$$

To evaluate the updated covariance, $\mathbf{P}_{k|k}$, the Kalman gain , $\mathbf{K}_k$ is also required. The next task is to find the Kalman gain. The Kalman filter aims at minimising the mean-square

[2]

$$\begin{aligned}
(\mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T)^T &= \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} \\
Tr(\mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T)^T &= Tr(\mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1})
\end{aligned}$$

estimation error. By taking the partial derivative[3] of Eq.(2.12) with respect to $\mathbf{K}_k$ gives

$$\frac{\partial Tr(\mathbf{P}_{k|k})}{\partial \mathbf{K}_k} = -2\mathbf{P}_{k|k-1}\mathbf{H}_k^T + 2\mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + 2\mathbf{K}_k\mathbf{R}_k. \tag{2.14}$$

By equating the right hand side of Eq. (2.14) to 0 yields

$$-2(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}\mathbf{H}_k^T + 2\mathbf{K}_k\mathbf{R}_k = 0. \tag{2.15}$$

By solving Eq.(2.15) for $\mathbf{K}_k$ yields

$$\begin{aligned} \mathbf{K}_k\mathbf{R}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T \\ \mathbf{P}_{k|k-1}\mathbf{H}_k^T &= \mathbf{K}_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T) \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \end{aligned} \tag{2.16}$$

Alternative form of updated covariance can be obtained by substituting Eq.(2.16) in Eq.(2.12)

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.17}$$

and

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k. \tag{2.18}$$

The Kalman filter is now summarised in Algorithm 1.

2.3 Discrete-Time H_∞ Filter

The Kalman filter assumes the process model has known dynamics and the noise sources has known statistics. However, these assumptions may limit the application of estimators

[3]For any matrix, M, and a symmetric matrix, N,

$$\frac{\partial Tr(MNM^T)}{\partial M} = 2MN.$$

Algorithm 1 The Kalman Filter

Initialise the state vector, $\widehat{\mathbf{x}}_{0|0}$, and the covariance, $\mathbf{P}_{0|0}$ (set $k = 1$).

Prediction

Evaluate the predicted state and covariance using

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}\\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}.\end{aligned}$$

Measurement Update

The updated state and covariance can be obtained as

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \nu_k\\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}\end{aligned}$$

where,

$$\begin{aligned}\nu_k &= \mathbf{z}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}\\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}.\end{aligned}$$

in many applications, as the process dynamics and noise statistics are not exactly known or may not be available. To overcome some of these limitations of Kalman filters, a few researchers have proposed H_∞ filters [12, 13, 14, 15, 16, 17, 29].

This section presents a brief introduction to an H_∞[4] filter which minimises the worst-case estimation error. This is in contrast to the Kalman filter which minimises the expected value of the variance of the estimation error. Furthermore, H_∞ does not make any assumptions about the statistics of the process and measurement noise. For a detailed formulation and derivation see for example [11], [15], [48] and [47].

The discrete-time process and observation models can be written as

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \tag{2.19}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{2.20}$$

where k is a time index, $\mathbf{x}_k$ is a state vector, $\mathbf{u}_k$ is a control input, $\mathbf{z}_k$ is a measurement

[4] H_∞ filters minimizes the worst case energy gain from the noise input to the estimation error; which is equivalent to minimising the H_∞ norm.

vector, and $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the process and measurement noises. These models are almost similar to that of the Kalman filter described in the Section 2.2.1; except the assumption on noises. The noise terms $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ may be random with possibly unknown statistics, or even they may be deterministic. They may have a non-zero mean.

In H_∞ filter, instead of directly estimating the state one may estimate a linear combination of states

$$\mathbf{n}_k = \mathbf{L}_k \mathbf{x}_k. \tag{2.21}$$

By replacing $\mathbf{L}$ with the identity matrix, one can directly estimate the state vector. In the game theory approach to H_∞ filtering, the performance measure is given by

$$\mathbf{J}_\infty = \frac{\sum_{k=1}^{N} \|\mathbf{n}_k - \widehat{\mathbf{n}}_k\|^2_{\mathbf{M}_k}}{\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2_{\mathbf{P}_0^{-1}} + \sum_{k=1}^{N} (\|\mathbf{w}_k\|^2_{\mathbf{Q}_k^{-1}} + \|\mathbf{v}_k\|^2_{\mathbf{R}_k^{-1}})} \tag{2.22}$$

where $\mathbf{P}_0$, $\mathbf{Q}_k$, $\mathbf{R}_k$, and $\mathbf{M}_k$ are symmetric positive definite weighing matrices chosen by the user based on the problem at hand. The norm notation used in this section is $\|e\|^2_{S_k} = e^T S_k e$.

In this dynamic game theory framework, there are essentially two players: the designer and the nature. The designer's goal is to find the estimate of the error $\mathbf{n}_k - \widehat{\mathbf{n}}_k$ so that the cost $\mathbf{J}_\infty$ is minimised, while the nature's goal is to maximise $\mathbf{J}_\infty$. The numerator of $\mathbf{J}_\infty$ is the energy of the estimation error and the denominator can be considered as the energy of the unknown disturbances. The nature can simply put large magnitudes of $\mathbf{w}_k$, $\mathbf{v}_k$, and $\mathbf{x}_0$ to achieve its ultimate goal, and this makes the game unfair to the designer. Thus, the $\mathbf{J}_\infty$ is defined with $(\mathbf{x}_0 - \hat{\mathbf{x}}_0)$,$\mathbf{w}_k$, and $\mathbf{v}_k$ in the denominator. Then, the nature needs to cleverly choose those disturbances in order to maximise $\mathbf{n}_k - \widehat{\mathbf{n}}_k$; likewise the designer also should be smart to find an estimation strategy to minimise $\mathbf{n}_k - \widehat{\mathbf{n}}_k$.

The task of the H_∞ filter is to minimise the state estimation error so that $\mathbf{J}_\infty$ is bounded by a prescribed threshold under the worst case $\mathbf{w}_k$, $\mathbf{v}_k$, and $\mathbf{x}_0$

$$\sup \mathbf{J}_\infty < \gamma^2 \tag{2.23}$$

where “sup” stands for supremum, $\gamma > 0$ is the error attenuation parameter.

Based on Eq.(2.23), the designer should find $\hat{\mathbf{x}}_k$ so that $\mathbf{J}_\infty < \gamma^2$ holds for any disturbances in $\mathbf{w}_k$, $\mathbf{v}_k$, and $\mathbf{x}_0$. The best the designer can do is to minimise $\mathbf{J}_\infty$ under worst case disturbances, then the $\mathbf{H}_\infty$ filter can be interpreted as the following ‘minmax’ problem

$$\min_{\hat{\mathbf{x}}_k} \quad \max_{\mathbf{w}_k,\mathbf{v}_k,\mathbf{x}_0} \quad \mathbf{J}_\infty \tag{2.24}$$

For detailed analysis and solution procedure to the H_∞ filtering problem see [47] and [11]. In this section, we use the H_∞ filter algorithm given in [49], as the relevant equations in this approach are closely related to that of the Kalman filter.

The predicted state vector and auxiliary matrix of H_∞ filter are

$$\begin{aligned}
\widehat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}
\end{aligned}$$

and the updated state and inverse of the updated auxiliary matrix can be obtained as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_\infty[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}] \tag{2.25}$$

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k - \gamma^{-2}\mathbf{I}_n \tag{2.26}$$

where

$$\mathbf{K}_\infty = \mathbf{P}_{k|k-1}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{2.27}$$

and $\mathbf{I}_n$ denotes the identity matrix of dimension $n \times n$.

An H_∞ filter is summarised in Algorithm 2, which has the similar structure to that of the Kalman filter.

It is interesting to note that, for very high values of γ, the updated auxiliary matrix of H_∞ filter in Eq.(2.26) and the covariance matrix of the Kalman filter in Eq.(2.18) are equivalent. Hence, the H_∞ filter’s performance can be matched with that of the Kalman

Algorithm 2 H_∞ Filter

Initialise the state vector, $\widehat{\mathbf{x}}_{0|0}$, and the auxiliary matrix, $\mathbf{P}_{0|0}$ (set $k = 1$).

Prediction

1: Evaluate the predicted state and auxiliary matrix using

$$\begin{aligned}
\widehat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}^T_{k-1} + \mathbf{Q}_{k-1}.
\end{aligned}$$

Measurement Update

1: The updated state and the inverse of auxiliary matrix can be obtained as

$$\begin{aligned}
\widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_\infty \nu_k \\
\mathbf{P}^{-1}_{k|k} &= \mathbf{P}^{-1}_{k|k-1} + \mathbf{H}^T_k \mathbf{R}^{-1}_k \mathbf{H}_k - \gamma^{-2}\mathbf{I}_n
\end{aligned}$$

where,

$$\begin{aligned}
\nu_k &= \mathbf{z}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1} \\
\mathbf{K}_\infty &= \mathbf{P}_{k|k-1}\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}^T_k + \mathbf{R}_k)^{-1}
\end{aligned}$$

filter, but the reverse is not true.

2.4 State Estimation and Control of a Quadruple-Tank System

In the previous sections, the Kalman and H_∞ filters were discussed. This section deals with the combined state estimation and control problem for a quadruple-tank system using Kalman filter and H_∞ filter. The controller design is based on the sliding mode control (SMC), which involves a stable sliding surface design followed by a control law design to ensure the system states onto the chosen surface [63]. We present the combined SMC and H_∞ filter for the quadruple-tank. Although we will only explore this approach for the quadruple-tank, it can be applicable to many other practical systems. The basic structure of this approach is shown in Figure 2.1. In this section, firstly, a brief literature survey on quadruple-tank system is given. The mathematical model of the quadruple-tank system is then presented, which is followed by SMC design and combined SMC-H_∞ filter closed-

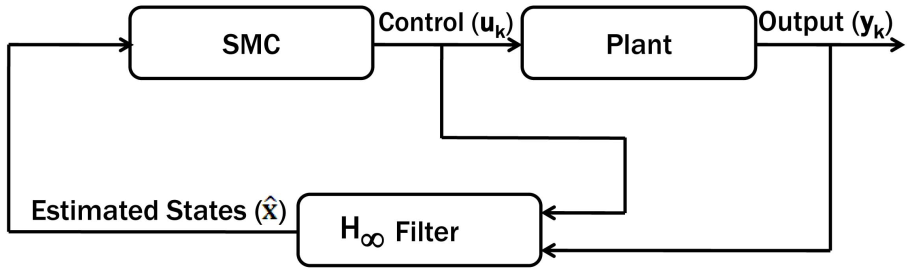

Figure 2.1: A new combined SMC-H_∞ filter approach.

loop simulations.

2.4.1 Quadruple-Tank System

The quadruple-tank is an interesting multivariable plant consisting of four interconnected tanks and two pumps [53] that some researchers have used to explore different control and estimation methods. Decentralised proportional-integral (PI) control, internal model control (IMC) and H_∞ controllers have been designed for the quadruple-tank system [52], and it was shown that the IMC and H_∞ controllers provided better performance than the PI controller. Different nonlinear model predictive controllers were proposed in [54] and [55], interconnection and damping assignment passivity based control for quadruple-tank system was given in [56]. A nonlinear sliding mode control (SMC) with feedback linearisation was proposed and implemented in [57].

It is a well known fact that all the states are required for state feedback controller design. However, in most of the practical applications, the states are not always available for feedback. Similarly, in the quadruple-tank system, only the first two states are assumed to be accessible for feedback and hence either one has to rely on output feedback control methods or the remaining two states are to be estimated for state feedback SMC design. The usage of an extended Kalman filter and high gain observers for the state estimation of a quadruple-tank system is given in [58], state estimation in non-Gaussian domain using particle filter is described in [59]. A very few researchers have demonstrated combined controller-observer scheme for quadruple-tank. In [60], Kalman filter is used for the estimation of unavailable states of a quadruple-tank system and the controller is based on PID and IMC.

In this section, discrete-time SMC using the Kalman and H_∞ filters is designed for the quadruple-tank. Nonlinear sliding surface proposed in [63] and [64] is considered to achieve better performance. The SMC for a quadruple-tank is also reported in [57]. But, our approach is different from [57] in two aspects. In [57], the objective is to control only two states and linear sliding surfaces are constructed using only the first two states. However, in this section the main emphasis is to control all the four states and nonlinear sliding surfaces are designed using complete state information. Secondly, SMC given in Theorem 1 of [57] require the full state information. Whereas, we have assumed only two states are available from sensors and the complete state vector is estimated using Kalman and H_∞ filters.

The quadruple-tank system is shown in Figure 2.2, which consists of four interconnected tanks, two pumps and two level sensors. For more details, see [53] for continuous time plant model. The quadruple-tank is discretised using Euler's method with sampling time of $\Delta T = 0.1s$. The discrete-time nonlinear quadruple-tank model is

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{1k} + \Delta T(-\frac{a_1}{A_1}\sqrt{2gx_{1k}} + \frac{a_3}{A_1}\sqrt{2gx_{3k}} + \frac{\gamma_1 k_1}{A_1}u_1) \\ x_{2k} + \Delta T(-\frac{a_2}{A_2}\sqrt{2g_{2k}} + \frac{a_4}{A_2}\sqrt{2gx_{4k}} + \frac{\gamma_2 k_2}{A_2}u_2) \\ x_{3k} + \Delta T(-\frac{a_3}{A_3}\sqrt{2gx_{2k}} + \frac{(1-\gamma_2)k_2}{A_3}u_2) \\ x_{4k} + \Delta T(-\frac{a_4}{A_4}\sqrt{2gx_{4k}} + \frac{(1-\gamma_1)k_1}{A_4}u_1) \end{bmatrix}$$

where, the state vector $\mathbf{x}$, consist of water levels of all tanks is $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$. The inputs are the voltages to the two pumps, $[u_1, u_2]^T$ and the outputs are the voltage from level measurements of the first two tanks. A_i is the cross section of Tank i and a_i is cross section of outlet hole. The first input u_1, directly effects the first and fourth states, whereas the second input u_2, has direct influence on second and third states. The outputs are the

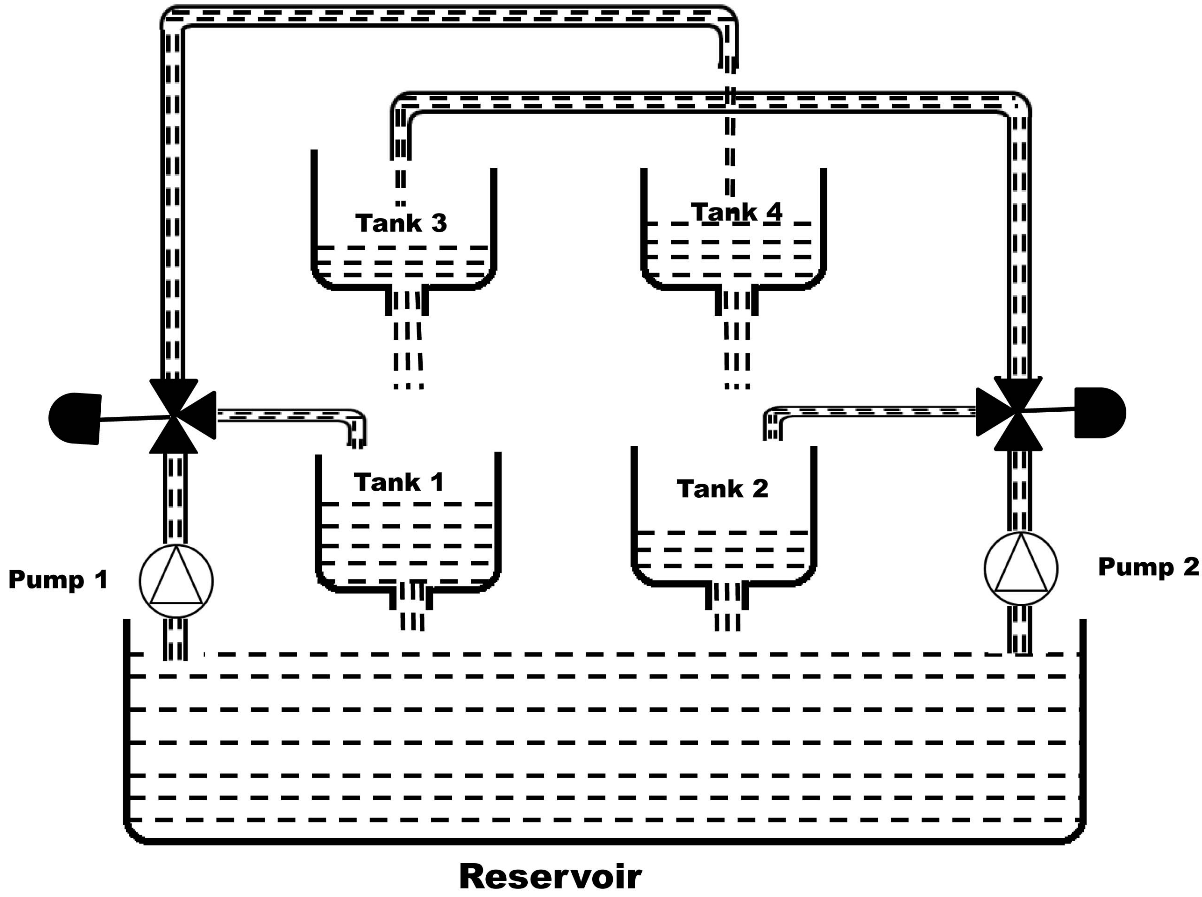

Figure 2.2: Quadruple-Tank System [53].

Parameters	Values
$A_1, A_3\ (cm^2)$	28
$A_2, A_4\ (cm^2)$	32
$a_1, a_3\ (cm^2)$	0.071
$a_2, a_4\ (cm^2)$	0.057
$k_c\ (V/cm)$	0.5
$g\ (cm/s^2)$	981

Table 2.1: Quadruple-tank parameters.

measured level signals, $k_c x_1$ and $k_c x_2$. The additive process and sensor noises are added to the state and output vectors, respectively. The parameter values used for simulation are given in Table 2.1.

One of the typical features of this quadruple-tank process is that, the plant can exhibit both minimum and non-minimum phase characteristics [53]. In this section, the control and estimator design are done at the non-minimum phase operating point. The corresponding parameters for this operating point are

$$\begin{aligned}
[x_1^0, x_2^0, x_3^0, x_4^0] &= [12.6, 13, 4.8, 4.9] \\
[k_1, k_2] &= [3.14, 3.29] \\
[\gamma_1, \gamma_2] &= [0.43, 0.34]
\end{aligned}$$

The quadruple-tank is linearised at the above operating point and is given below

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{w}_k \tag{2.28}$$

$$\mathbf{z}_k = \mathbf{H} + \mathbf{v}_k \tag{2.29}$$

where, the state space matrices are

$$\mathbf{F} = \begin{bmatrix} 1+\frac{a_{11}\Delta T}{2\sqrt{x_1^0}} & 0 & \frac{a_{13}\Delta T}{2\sqrt{x_3^0}} & 0 \\ 0 & 1+\frac{a_{21}\Delta T}{2\sqrt{x_2^0}} & 0 & \frac{a_{24}\Delta T}{2\sqrt{x_4^0}} \\ 0 & 0 & 1+\frac{a_{31}\Delta T}{2\sqrt{x_3^0}} & 0 \\ 0 & 0 & 0 & 1+\frac{a_{41}\Delta T}{2\sqrt{x_4^0}} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} b_1\Delta T & 0 \\ 0 & b_2\Delta T \\ 0 & b_3\Delta T \\ b_4\Delta T & 0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix}$$

and the parameters are

$a_{11} = -\frac{a_1}{A_1}\sqrt{2g}$, $a_{12} = \frac{a_3}{A_1}\sqrt{2g}$, $a_{21} = -\frac{a_2}{A_2}\sqrt{2g}$, $a_{24} = \frac{a_4}{A_2}\sqrt{2g}$, $a_{31} = -\frac{a_3}{A_3}\sqrt{2g}$, $a_{41} = -\frac{a_4}{A_4}\sqrt{2g}$, $b_1 = \frac{\gamma_1 k_1}{A_1}$, $b_2 = \frac{\gamma_2 k_2}{A_2}$, $b_3 = \frac{(1-\gamma_2)k_2}{A_3}$, and $b_4 = \frac{(1-\gamma_1)k_1}{A_1}$.

2.4.2 Sliding Mode Control of Quadruple-Tank System

This section deals with the SMC of quadruple-tank system. The objective is to control all the four states, unlike in [57], where only two states are controlled. For an improved performance, nonlinear sliding surfaces are considered during SMC design. Using a nonlinear sliding surface, the damping ratio of a system can be varied from its initial low value to final high value. The initial low damping ratio results in a faster response and the later high damping avoids overshoot. Thus the nonlinear surface ascertains the reduction in settling time without any overshoot [63], [64].

2.4.2.1 SMC Design

The plant in Eq.(2.28) is not in the regular form, as the input matrix **G** does not have any zero-row vectors. Before designing the SMC control, the plant need to be transformed in the regular form. The plant can be transformed into the regular form by using the transformation matrix

$$T = \begin{bmatrix} -\frac{1}{b_1} & 0 & 0 & \frac{1}{b_4} \\ 0 & -\frac{1}{b_2} & \frac{1}{b_3} & 0 \\ 0 & 0 & \frac{1}{b_3} & 0 \\ 0 & 0 & 0 & \frac{1}{b_4} \end{bmatrix}.$$

The new transformed system can be written as

$$\mathbf{y}_{k+1} = (T\mathbf{F}T^{-1})\mathbf{y}_k + T\mathbf{G}\mathbf{u}_k + T\mathbf{G}\mathbf{w}_k \tag{2.30}$$

where $\mathbf{y}_k = T\mathbf{x}_k$ and it can be further expressed as

$$\mathbf{y}_{k+1} = \begin{bmatrix} \mathbf{y}_{u,k+1} \\ \mathbf{y}_{l,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{11} & \mathbf{y}_{12} \\ \mathbf{y}_{21} & \mathbf{y}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{u,k} \\ \mathbf{y}_{l,k} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_l \end{bmatrix} \mathbf{u}_k + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_l \end{bmatrix} \mathbf{w}_k \tag{2.31}$$

where

$$
\begin{aligned}
\mathbf{y}_{11} &= \begin{bmatrix} 1+\frac{a_{11}\Delta T}{2\sqrt{x_1^0}} & 0 \\ 0 & 1+\frac{a_{21}\Delta T}{2\sqrt{x_2^0}} \end{bmatrix} \\
\mathbf{y}_{12} &= \begin{bmatrix} \frac{a_{13}b_3\Delta T}{2b_1\sqrt{x_3^0}} & \frac{\Delta T}{2}\left(\frac{a_{41}}{\sqrt{x_4^0}}-\frac{a_{11}}{\sqrt{x_1^0}}\right) \\ \frac{\Delta T}{2}\left(\frac{a_{31}}{\sqrt{x_3^0}}-\frac{a_{21}}{\sqrt{x_2^0}}\right) & \frac{a_{24}b_4\Delta T}{2b_2\sqrt{x_4^0}} \end{bmatrix} \\
\mathbf{y}_{21} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
\mathbf{y}_{22} &= \begin{bmatrix} 1+\frac{a_{31}\Delta T}{2\sqrt{x_3^0}} & 0 \\ 0 & 1+\frac{a_{41}\Delta T}{2\sqrt{x_4^0}} \end{bmatrix} \\
\mathbf{G}_l &= \begin{bmatrix} 0 & \Delta T \\ \Delta T & 0 \end{bmatrix}.
\end{aligned}
$$

2.4.2.2 Nonlinear Sliding Surface Design

The performance of the SMC based closed loop system can be enhanced by using the nonlinear surfaces [63], [64] and hence in the present work the nonlinear sliding surface is considered.

Let the nonlinear sliding surface [63] be

$$s_k = c_k^T \mathbf{y}_k \tag{2.32}$$

where

$$c_k^T = [K - \psi(\mathbf{z}_k)\mathbf{y}_{12}^T P(\mathbf{y}_{11} - \mathbf{y}_{12}K) \quad I_2] \tag{2.33}$$

and I_2 is the second order identity matrix.

The gain matrix,

$$K = \begin{bmatrix} -0.66651 & -0.2026 \\ 0.06417 & -0.6641 \end{bmatrix} \tag{2.34}$$

is obtained by solving the linear quadratic regulator (LQR) problem for $\mathbf{y}_{11}$ and $\mathbf{y}_{12}$, such that $(\mathbf{y}_{11} - \mathbf{y}_{12}K)$ have stable eigenvalues. The weighting matrices considered for LQR design are

$$Q_{lqr} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \; and \;\; R_{lqr} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

The matrix P, required for Eq.(2.33) is obtained by solving the following Lyapunov equation

$$P = (\mathbf{y}_{11} - \mathbf{y}_{12}K)^T P(\mathbf{y}_{11} - \mathbf{y}_{12}K) + W. \tag{2.35}$$

By selecting W as I_2, the corresponding P matrix is

$$\begin{bmatrix} 1223.39 & -32.47 \\ -32.57 & 162.17 \end{bmatrix}.$$

The nonlinear function, $\psi(\mathbf{z}_k)$, can be chosen as [63]

$$\psi(\mathbf{z}_k) = \begin{bmatrix} -\beta_1 e^{-m_1|x^0_{1,k-1}|} & 0 \\ 0 & -\beta_2 e^{-m_2|x^0_{2,k-1}|} \end{bmatrix} \tag{2.36}$$

where β_1, β_2, m_1 and m_2 are tuning parameters. These parameters are chosen as unity and $\psi(\mathbf{z}_k)$ is chosen such that it satisfies the below condition [64], [65]

$$2\psi(\mathbf{z}_k) + \psi(\mathbf{z}_k)\mathbf{y}_{12}^T P \mathbf{y}_{12}\psi(\mathbf{z}_k) \leq 0. \tag{2.37}$$

2.4.2.3 Control Law

In this sub-section, the control law will be derived. From Eq.(2.32), s_{k+1} can be written as

$$\begin{aligned} s_{k+1} &= c_{k+1}^T \mathbf{y}_{k+1} \\ s_{k+1} &= c_{k+1}^T T \mathbf{x}_{k+1}. \end{aligned} \tag{2.38}$$

By using Eq.(2.28) and (Eq.2.38), s_{k+1} can be written as

$$s_{k+1} = c_{k+1}^T T \mathbf{F} \mathbf{x}_k + c_{k+1}^T T \mathbf{G} \mathbf{u}_k + c_{k+1}^T T \mathbf{G} \mathbf{w}_k. \tag{2.39}$$

In discrete-time sliding mode control, one of the objectives is to achieve the sliding surfaces, $s_k = 0$ in finite time [63]. This can be achieved by an equivalent control law [66] by setting

$$s_{k+1} = 0. \tag{2.40}$$

By using Eq.(2.39) and Eq.(2.40), the control law can be derived as

$$c_{k+1}^T T \mathbf{F} \mathbf{x}_k + c_{k+1}^T T \mathbf{G} \mathbf{u}_k + c_{k+1}^T T \mathbf{G} \mathbf{w}_k = 0 \tag{2.41}$$

and

$$\mathbf{u}_k = -(c_{k+1}^T T \mathbf{G})^{-1} (c_{k+1}^T T \mathbf{F} \mathbf{x}_k + c_{k+1}^T T \mathbf{G} \mathbf{w}_k). \tag{2.42}$$

The aforementioned control law contains uncertain terms or process noise. However, in general these uncertain terms are not known and hence they can be replaced by the average of known bounds of these terms [64]. The modified control law can be written as

$$\mathbf{u}_k = -(c_{k+1}^T T \mathbf{G})^{-1} (c_{k+1}^T T \mathbf{F} \mathbf{x}_k + d_m). \tag{2.43}$$

where d_m is the average of lower and upper bounds of $c_{k+1}^T T\mathbf{G}\mathbf{w}_k$. From Eq.(2.43), it can be seen that the control law at k^{th} time instant require the information at $k+1^{th}$ time instant, and in general is not feasible. However, from Eq.(2.33) and Eq.(2.36), only the output information at k^{th} instant ($x_{1,k}$ and $x_{2,k}$) is required to find c_{k+1} and hence the control input, $\mathbf{u}_k$, can be evaluated using the output information at k^{th} instant.

2.4.3 Stability of the nonlinear sliding surface

From Eq.(2.31),

$$\mathbf{y}_{u,k+1} = \mathbf{y}_{11}\mathbf{y}_{u,k} + \mathbf{y}_{12}\mathbf{y}_{l,k} \tag{2.44}$$

During the sliding, $s_k = 0$, and hence Eq.(2.32) can be written as

$$c_k^T \mathbf{y}_k = 0 \tag{2.45}$$

$$\Rightarrow \quad \begin{bmatrix} K - \psi(\mathbf{z}_k)\mathbf{y}_{12}^T P \mathbf{y}_{eq} & I_2 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{u,k} & \mathbf{y}_{l_k} \end{bmatrix}^T = 0 \tag{2.46}$$

$$\Rightarrow \quad \mathbf{y}_{l,k} = -[K - \psi(\mathbf{z}_k)\mathbf{y}_{12}^T P \mathbf{y}_{eq}]\mathbf{y}_{u,k} \tag{2.47}$$

where $\mathbf{y}_{eq} = \mathbf{y}_{11} - \mathbf{y}_{12}K$.

From Eq.(2.44) and Eq.(2.47)

$$\mathbf{y}_{u,k+1} = \mathbf{y}_{11}\mathbf{y}_{u,k} + \mathbf{y}_{12}(-[K - \psi(\mathbf{z}_k)\mathbf{y}_{12}^T P \mathbf{y}_{eq}])\mathbf{y}_{u,k} \tag{2.48}$$

$$= (\mathbf{y}_{11} - \mathbf{y}_{12}K)\mathbf{y}_{u,k} + \mathbf{y}_{12}\psi(\mathbf{z}_k)\mathbf{y}_{12}^T P \mathbf{y}_{eq}\mathbf{y}_{u,k} \tag{2.49}$$

$$= (\mathbf{y}_{12}\psi(\mathbf{z}_k)\mathbf{y}_{12}^T P + I)\mathbf{y}_{eq}\mathbf{y}_{u,k}. \tag{2.50}$$

The stability of the nonlinear sliding surface can be proved by using the Lyapunov theory (see Appendix A for more details). Let us assume the Lyapunov function for the system defined in Eq.(2.50) is

$$V_k = \mathbf{y}_{u,k}^T P \mathbf{y}_{u,k}. \tag{2.51}$$

The increment of V_k is

$$\begin{aligned}
\triangle V_k &= V_{k+1} - V_k \qquad (2.52)\\
&= \mathbf{y}_{u,k+1}^T P \mathbf{y}_{u,k+1} - \mathbf{y}_{u,k}^T P \mathbf{y}_{u,k}\\
&= (\mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T + \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T) P (\mathbf{y}_{eq} \mathbf{y}_{u,k} + \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k}) - \mathbf{y}_{u,k}^T P \mathbf{y}_{u,k}\\
&= \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k} + \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k} + \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k} +\\
&\quad \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k} - \mathbf{y}_{u,k}^T P \mathbf{y}_{u,k}\\
&= -\mathbf{y}_{u,k}^T (P - \mathbf{y}_{eq}^T P \mathbf{y}_{eq}) \mathbf{y}_{u,k} + \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12} [2\psi(\mathbf{z}_k) + \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k)] \mathbf{y}_{12}^T P \mathbf{y}_{eq} \mathbf{y}_{u,k}\\
&= -\mathbf{y}_{u,k}^T W \mathbf{y}_{u,k} + M^T [2\psi(\mathbf{z}_k) + \psi(\mathbf{z}_k) \mathbf{y}_{12}^T P \mathbf{y}_{12} \psi(\mathbf{z}_k)] M \qquad (2.53)
\end{aligned}$$

where $M = \mathbf{y}_{u,k}^T \mathbf{y}_{eq}^T P \mathbf{y}_{12}$.

From Eq.(2.37) and Eq.(2.53), one can write

$$\nabla V \leq -\mathbf{y}_{u,k}^T W \mathbf{y}_{u,k}. \qquad (2.54)$$

Since the increment of the Lyapunov function is negative definite, the equilibrium point for Eq.(2.50) is stable, and hence the designed nonlinear sliding surface is stable. In a similar way by constructing the Lyapunov function of sliding surface, s_{k+1}, the increment of the Lyapunov function can be easily shown as negative definite which in turn proves the existence of sliding mode [64].

2.5 Simulations and Results

By using the filter Algorithms described in Section 2.2 and Section 5.2, all the four states of quadruple-tank are estimated using x_1 and x_2 . One may note that, the control input $\mathbf{u}$ required for the quadruple-tank is obtained from SMC controller given in Section 2.4.2. This proposed scheme for a quadruple-tank system is shown in Figure 2.3. The process noise is added to all the four states, whereas the measurement noise is added to the last

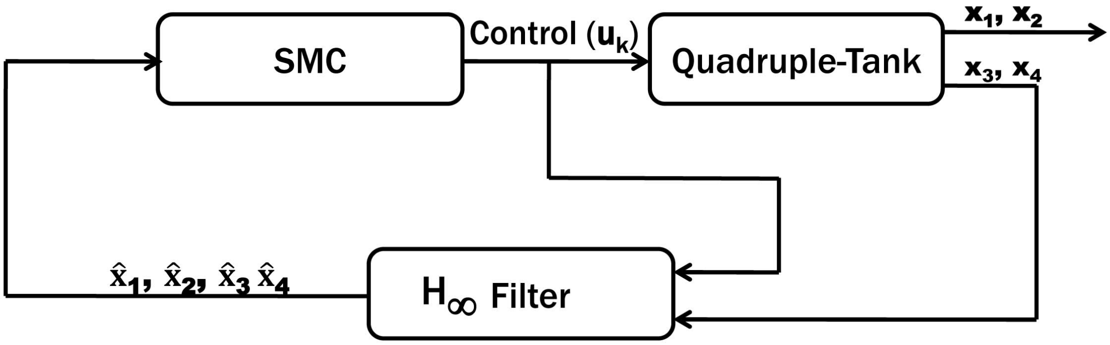

Figure 2.3: Proposed scheme using SMC and H_∞ filter for quadruple-tank system

two states only; the quadruple-tank block in Figure 2.3 assumed to have additive noises and hence separate noises are not added in the block diagram. The usage of the H_∞ filter in this work is required for two purposes; it is used to estimate the two unavailable states, and to inherently filter out the process and sensor noises. One may note that in this work although the first two states of the quadruple-tank are available from sensors, we are still using the full order state estimation rather than reduced order state estimation. The first two estimated states are less noisy than actual measured states and are beneficial for the state feedback SMC control design.

This section describes the various simulations done for closed loop quadruple-tank system. Although, the controller and estimator designs are done for linearised model, the simulations in this section are performed on full nonlinear model. The SMC given in Section 2.4.2 and the estimator in Sections 2.2 and 5.2 are considered in the simulations. The first two sensed states from quadruple-tank are given to filters, which then estimates all the four states. These estimated states from estimators are used by the SMC, which then provides the input to the quadruple-tank system. The initial values of the plant are perturbed by $+15\%$ of their nominal values, and the objective is to bring back the perturbed states to the actual initial conditions. The chosen initial covariance matrix, $\mathbf{P}_{0|0}$, is

$$\mathbf{P}_{0|0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Two sets of simulations are performed to show the efficacy of the proposed method and are compared with the Kalman filter's response. The first set involves the closed loop simulation in the presence of Gaussian noises and the second one involves the simulation with non-Gaussian noises.

2.5.1 Simulation in the presence of Gaussian noises

In this subsection, it is assumed that plant and sensor noises, $\mathbf{w}_k$ and $\mathbf{v}_k$, are zero-mean Gaussian. The standard deviations for all the four states and the measurements are 0.0316. The corresponding covariance matrices for the Kalman filter are

$$\mathbf{Q} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}.$$

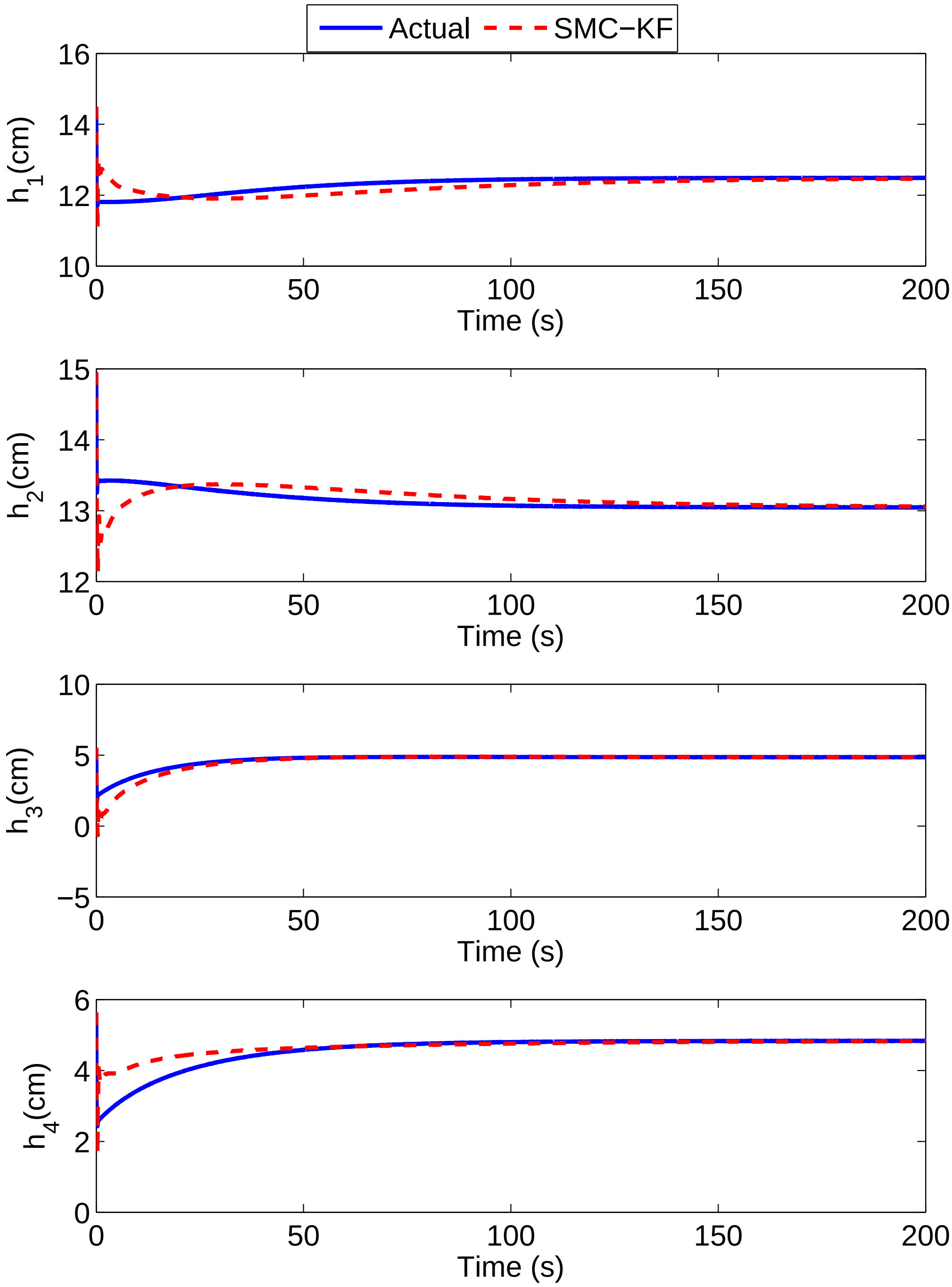

Figure 2.4: Actual and estimated states of the quadruple-tank using the Kalman filter in the presence of Gaussian noises.

The tuning parameters for the H_∞ filter simulations are

$$\mathbf{Q} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}.$$

and the performance bound, γ, is chosen as 1. These tuning parameters are chosen by trial and error method. One can note that, the chosen standard deviations for process and measurement noises using H_∞ filter are higher than the Kalman filter. The closed-loop performance can further be improved by using rigorous tuning methods; at the cost of increased computation complexity.

The SMC based quadruple-tank levels using the Kalman filter are shown in Figure 2.4, whereas for the H_∞ filter are shown in Figure 2.5. In both figures, the actual and estimated perturbed states reaches their actual values in the finite time. The estimated states closely follows the actual states and the error between them decreases with time. From the initial transient response, it can be seen that the H_∞ filter's response is faster than that of the Kalman filter's. The estimation errors for the SMC based Kalman and H_∞ filters for Gaussian noises are shown in Figure 2.6. The root mean square error (RMSE) plots are shown in Figure 2.7, where the SMC based on H_∞ filter shows the better performance in the presence of the Gaussian noises. The maximum state estimation errors over the simulation time (∞-norm) for SMC based Kalman filter's four states are 1.2051, 1.3111, 2.8799 and 1.4693, respectively and for the SMC based H_∞ filter are 0.7185, 0.7254, 1.6648 and 0.8921, respectively. In simulations, the quadruple-tank is excited by Gaussian noises for both the Kalman and H_∞ filters'. In the Kalman filter, if the standard deviation of noises once fixed then the corresponding covariances are the square of the

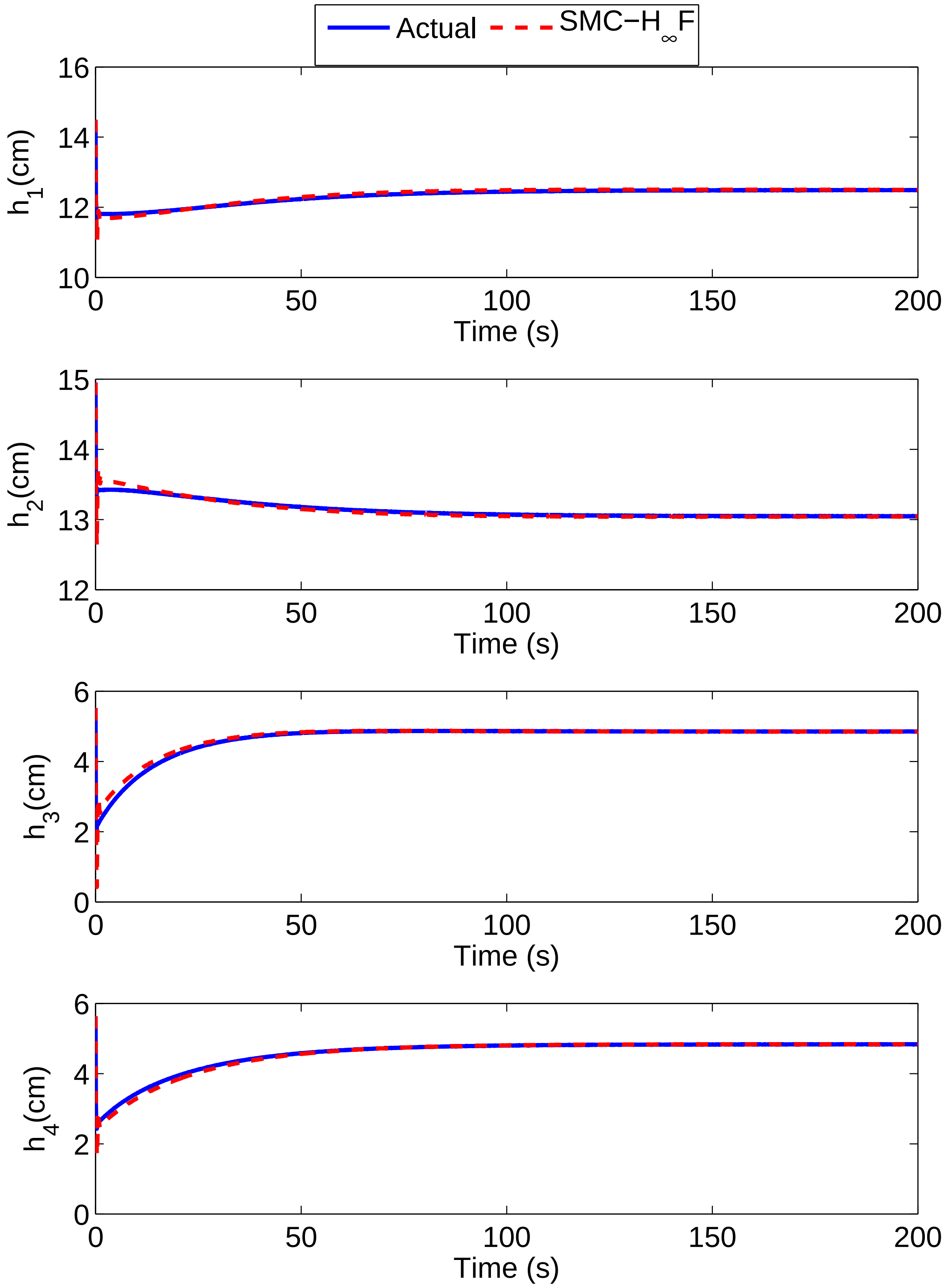

Figure 2.5: Actual and estimated states of the quadruple-tank using H_∞ filter in the presence of Gaussian noises.

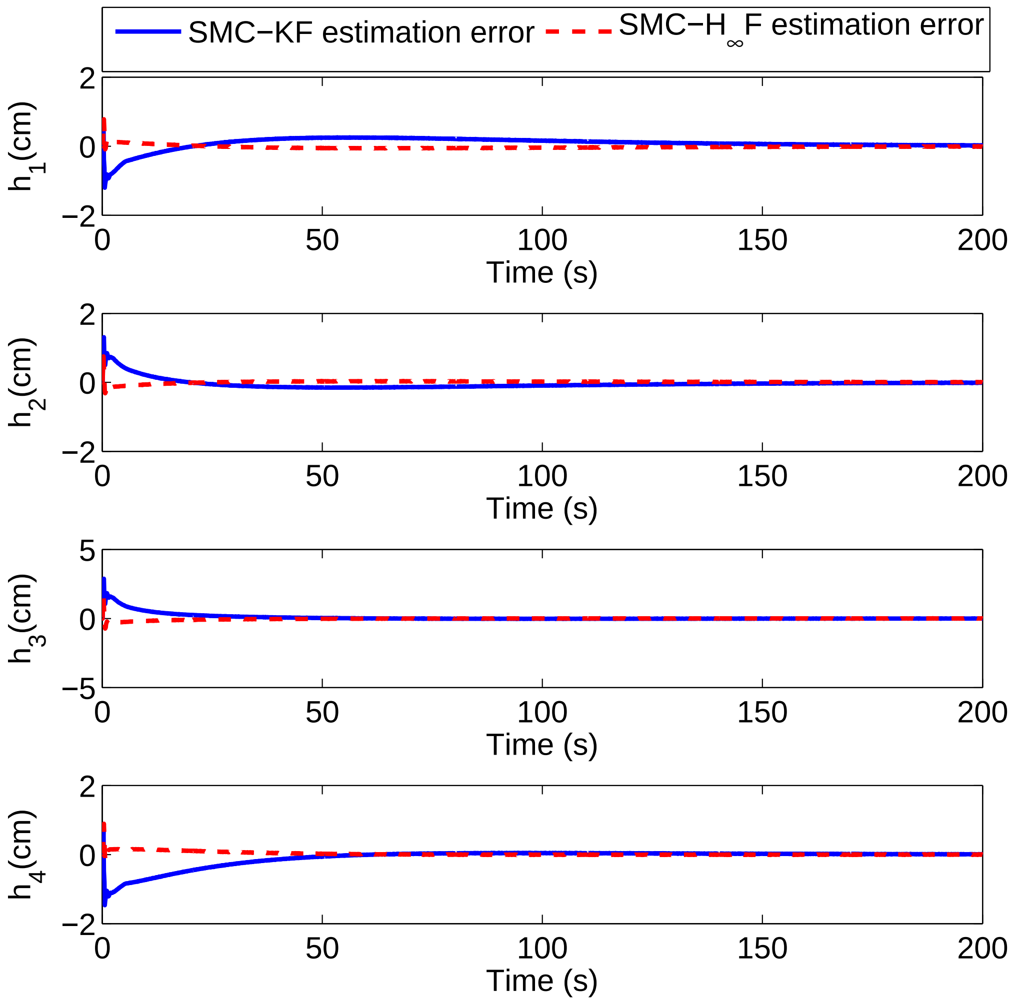

Figure 2.6: Estimation errors for the quadruple-tank using Kalman and H_∞ filters in the presence of Gaussian noises.

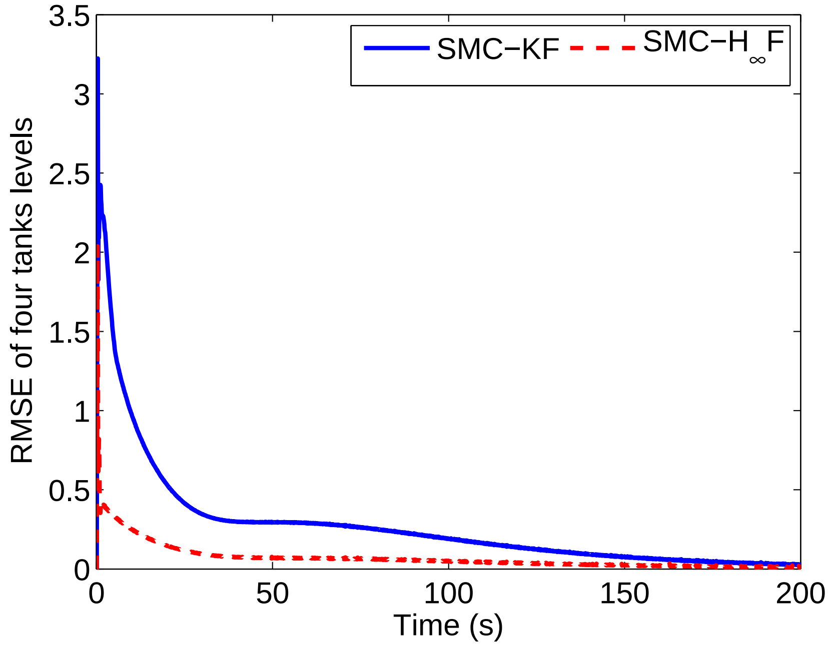

Figure 2.7: RMSEs using Kalman and H_∞ filters in the presence of non-Gaussian noises.

standard deviations. However, in the H_∞ filter, there is a freedom to select tuning parameters irrespective of the noises. Both Kalman and H_∞ filters' performances can further improved by tuning $\mathbf{Q}$ and $\mathbf{R}$. The results in this section are shown for the full nonlinear model. When the same simulations are repeated with linear open-loop plant model, the Kalman filter's response is better than the H_∞ filter's response, as the Kalman filter is the optimal estimator for linear-Gaussian systems.

2.5.2 Simulation in the presence of non-Gaussian noises

In most of the real-life applications, the assumption of zero-mean and Gaussian noises are not valid. To validate the proposed approach; non-zero mean, non-Gaussian noises

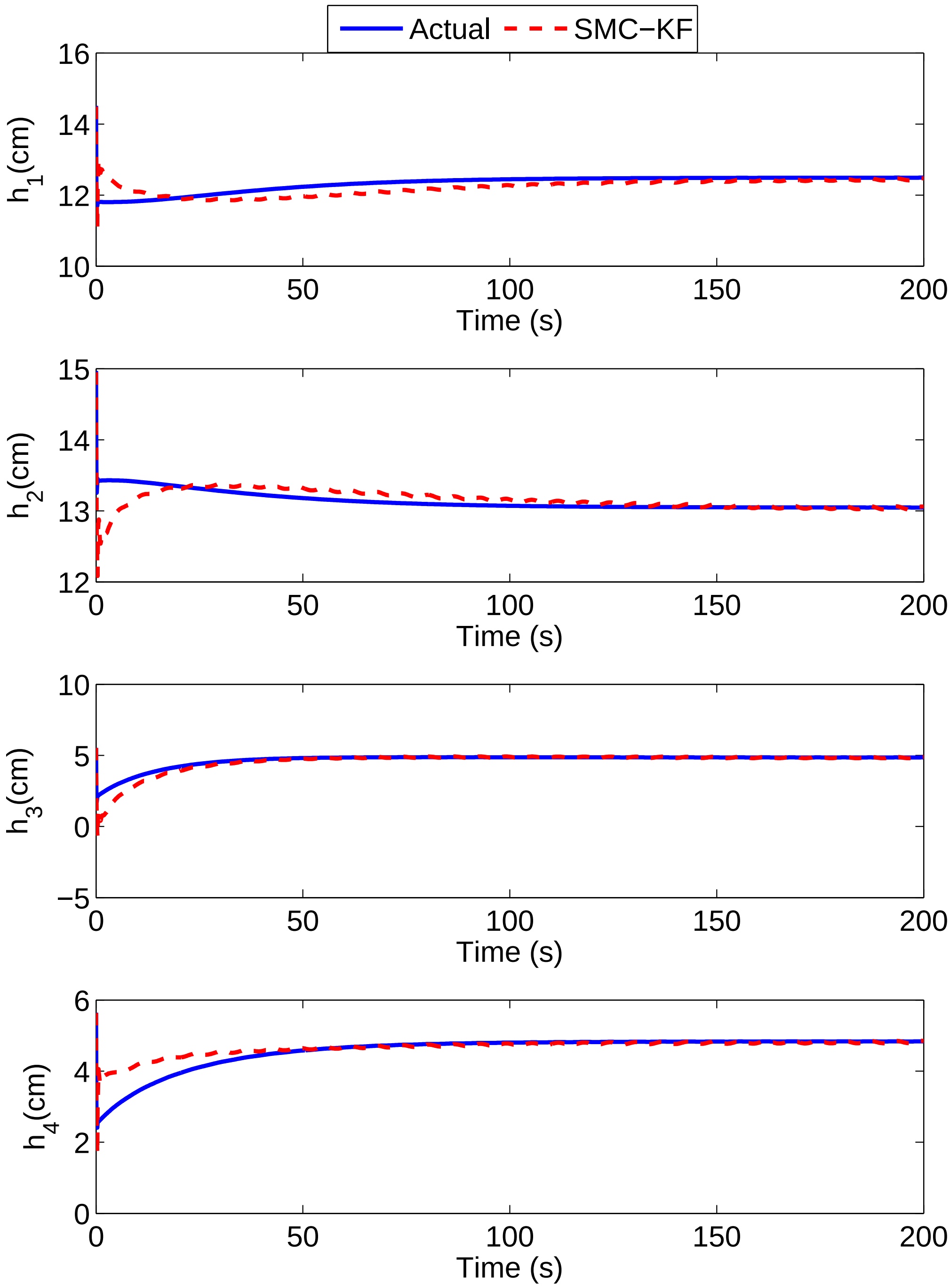

Figure 2.8: Actual and estimated states of the quadruple-tank using the Kalman filter in the presence of non-Gaussian noises.

are considered for the simulations. The process and measurement noises are

$$\mathbf{w}_k = \begin{bmatrix} 0.01+0.01\times sin(0.1k) \\ 0.01+0.01\times sin(0.1k) \\ 0.01+0.01\times sin(0.1k) \\ 0.01+0.01\times sin(0.1k) \end{bmatrix}$$

$$\mathbf{v}_k = \begin{bmatrix} 0.01+0.01\times sin(0.1k) \\ 0.01+0.01\times sin(0.1k) \end{bmatrix}.$$

The maximum magnitudes of $\mathbf{w}_k$ and $\mathbf{v}_k$ are used in the noise covariance matrices for the Kalman filter. For the H_∞ filter, the tuning parameters given in the Section 2.5.1 are considered. The closed loop simulations with the sinusoidal noises are shown in Figures 2.8 and 2.9. One can notice that the simulations with the non-Gaussian noises are similar to the Gaussian case. The H_∞ filter's response converges faster than the Kalman filter's response.

The estimation errors for the SMC based Kalman and H_∞ filters for non-Gaussian noises are shown in Figure 2.10. The RMSE plots are shown in Figure 2.11, where the SMC based on H_∞ filter shows the better performance in the presence of the non-Gaussian noises. The small offsets are due to the non-zero bias of the noises, which were intentionally added to verify the effectiveness of the H_∞ filter in the presence of non-zero mean noises. The $\infty - norms$ of state estimation errors over the simulation time for SMC based Kalman filter's four states are 1.2004, 1.3677, 2.9674 and 1.4793, respectively and for the SMC based H_∞ filter are 0.7185, 0.7254, 1.6648 and 0.8921, respectively.

2.6 Conclusions

In this chapter, the basic concepts and algorithms for the Kalman and H_∞ filters, and their application in the control theory have been presented. The combined sliding mode control and H_∞ filter scheme for practical systems are proposed. The proposed scheme

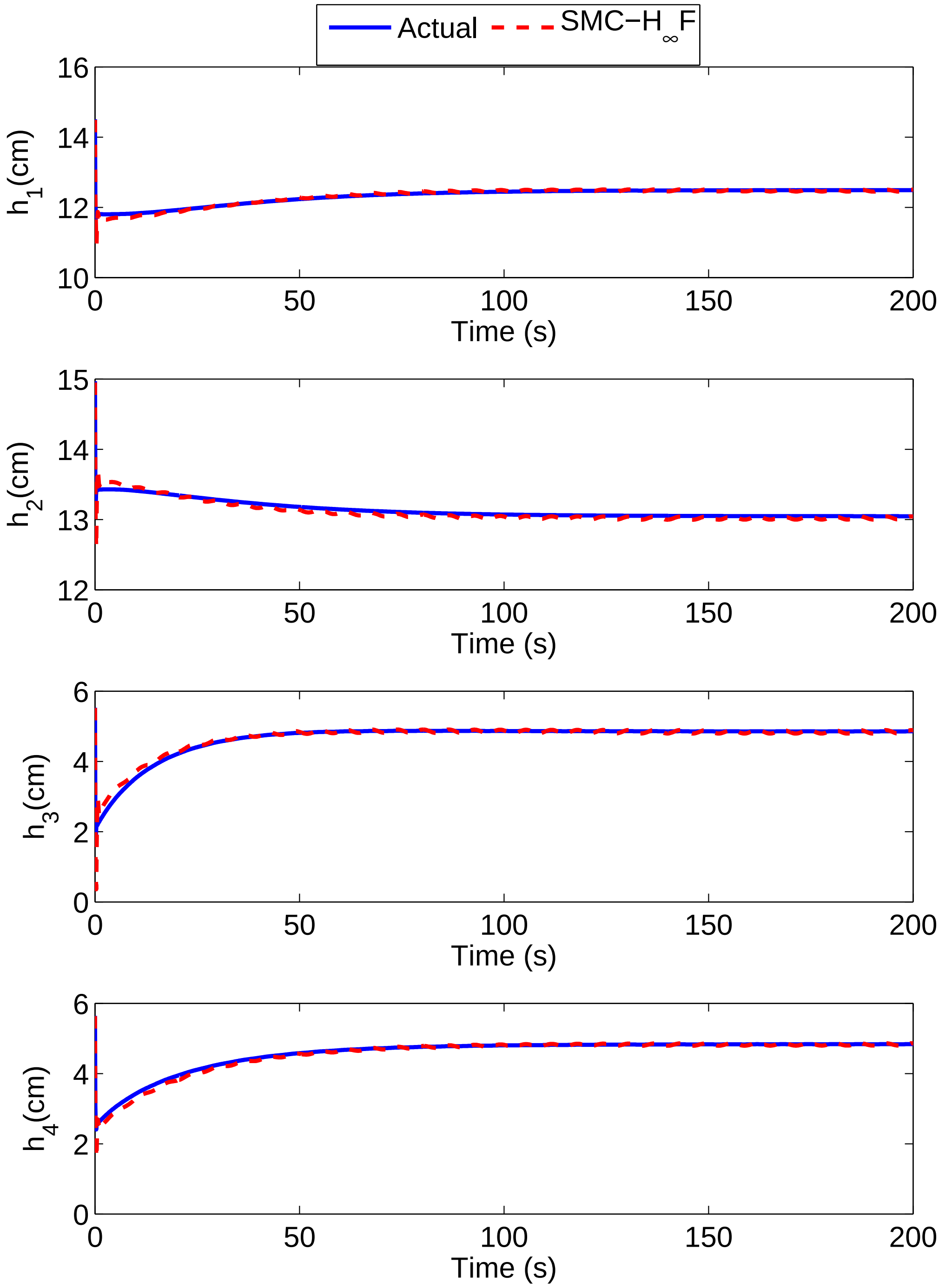

Figure 2.9: Actual and estimated states of the quadruple-tank using H_∞ filter in the presence of non-Gaussian noises.

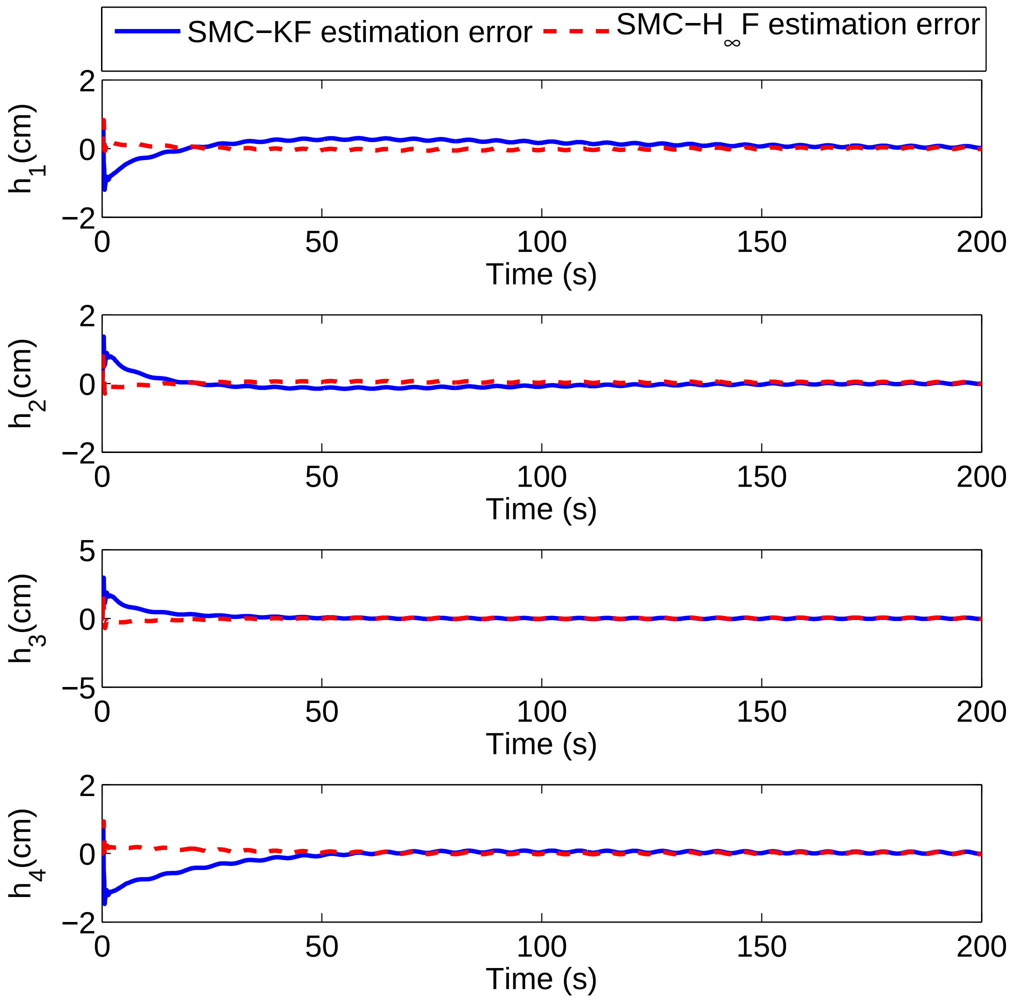

Figure 2.10: Estimation errors for the quadruple-tank using Kalman and H_∞ filters in the presence of non-Gaussian noises.

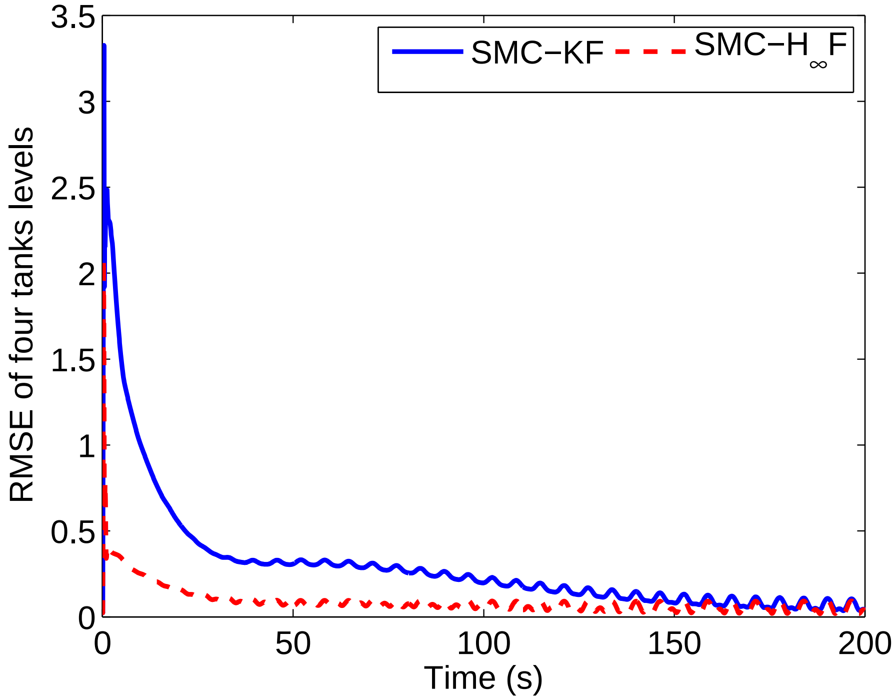

Figure 2.11: RMSEs using Kalman and H_∞ filters in the presence of non-Gaussian noises.

is implemented on a simulation example of a full nonlinear quadruple-tank system. The first two states of the quadruple-tank are assumed to be available and the remaining two states are estimated using the filters; the estimated states are then used for the sliding mode control design. The efficacy of the proposed approach for quadruple-tank is verified by extensive simulations. The Kalman and H_∞ filters based sliding mode control are compared for a quadruple-tank and it was found that the H_∞ filter based sliding mode control outperforms the Kalman filter based sliding mode control. It was also shown that the proposed scheme not only works for Gaussian and non-Gaussian noises, but also works for non-zero mean noises. This chapter mainly explores the linear state estimation methods and their applicability to the control theory; the next chapter will focus on non-linear state estimation methods and their application.

Chapter 3

Nonlinear State Estimation and CKF SLAM

3.1 Introduction

Nonlinearity can be a challenging issues in the controllers and observers design. Almost all the practical systems are inherently nonlinear [18], [19], and there are cases when linear controllers or estimators designs for nonlinear systems are tedious. In this chapter, the main emphasis is given to the nonlinear state estimation methods and their application. The Kalman filter and its variants are the main estimation tool for practical systems from the past several decades. However, Kalman filter was actually derived for linear systems [7] and later it has been extended for nonlinear applications [31]. The extended version of the Kalman filter for nonlinear systems is known as an extended Kalman filter (EKF). Similar to Kalman filter, EKF also has the prediction and measurement update stages. In EKF, the plant and measurement models are linearised about the best available estimate. EKFs are only suitable for 'mild' nonlinearities where the first-order approximations of the nonlinear functions are available and they also require evaluation of state Jacobians at each iteration. In some of the practical applications, these approximations will degrade the overall performance. To handle some of the issues with the EKF, a derivative free unscented Kalman filter (UKF) [32] was proposed; which uses the sigma point to capture the mean and covariance of the nonlinear system. More recently, the cubature Kalman filter (CKF) was proposed as an alternative to the UKF. CKF is a Gaussian approximation of Bayesian filter, but provides a more accurate filtering estimates than existing Gaussian filters. There are a few other nonlinear estimation techniques found in the literature, namely, Rao-Blackwellised particle filters [38], which are the improvised version of particle filters [39], Gaussian filters [40], state dependent Riccati equation filters [41,42], sliding mode observers [43], Fourier-Hermite Kalman filter [44], adaptive filters [21,22], etc.

This chapter is divided in to two parts; the first part deals with the EKF, UKF and CKF. The means and covariances of the polar-to-rectangular coordinate transformation using linearised, unscented and cubature transformations are investigated. In the second part, we propose a solution to simultaneous localisation and mapping (SLAM) using CKF and is compared with EKF- and UKF-SLAM.

The rest of this chapter is structured as follows. Section 3.2 deals with the discrete-time EKF and Section 3.3 deals with the unscented transformation and UKF. Cubature transformation and CKF are briefed in Section 3.4. A solution to SLAM using CKF is detailed in Section 3.5.2. Finally, Section 3.6 concludes this chapter.

3.2 Extended Kalman Filter

Consider the discrete nonlinear process and measurement models as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{3.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \tag{3.2}$$

where k is the time index, $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector, $\mathbf{u}_k$ is control input, $\mathbf{z}_k$ is the measurement, $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the process and measurement noises, respectively. These noises are assumed to be zero mean Gaussian-distributed random variables with covariances of

$\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$.

Similar to Kalman filter, EKF is also a recursive process consisting of prediction and measurement update but it requires Jacobians of the process and measurement models.

The predicted state vector and covariance matrix can be written as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{3.3}$$

$$\mathbf{P}_{k|k-1} = \nabla\mathbf{f}_x \mathbf{P}_{k-1|k-1} \nabla\mathbf{f}_x^T + \mathbf{Q}_{k-1} \tag{3.4}$$

and the updated state and covariance can be obtained as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k[\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})] \tag{3.5}$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_n - \mathbf{K}_k \nabla\mathbf{h}_x)\mathbf{P}_{k|k-1} \tag{3.6}$$

where $\mathbf{I}_n$ denotes the identity matrix of dimension $n \times n$ and the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T \left[\nabla\mathbf{h}_x \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T + \mathbf{R}_k\right]^{-1}. \tag{3.7}$$

The Jacobians of $\mathbf{f}$ and $\mathbf{h}$, $\nabla\mathbf{f}_x$ and $\nabla\mathbf{h}_x$, are evaluated at $\hat{\mathbf{x}}_{k-1|k-1}$ and $\hat{\mathbf{x}}_{k|k-1}$, respectively. EKF is summarised in Algorithm 3; for detailed formulation and derivation of EKF, please see [9] and [11].

3.2.1 Nonlinear Transformation and the effects of Linearisation

The EKF described in Section 3.2 was based on the first order Taylor series approximation of nonlinear functions. This subsection investigates the effects of linearisation on nonlinear transformation of polar to cartesian coordinates and the estimation error analysis. One can expect the similar error in EKF, when it applied to nonlinear systems as it uses the first-order linearisation. The similar discussion has been considered in [11,32,33,46]. In mapping application, the vehicle (robot/UAV) takes the observation of the landmarks and

Algorithm 3 Extended Kalman Filter

Initialise the state vector, $\hat{\mathbf{x}}_{0|0}$, and the covariance matrix, $\mathbf{P}_{0|0}$ (set $k = 1$).

Prediction

1: The predicted state and covariance matrix are

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \mathbf{P}_{k|k-1} &= \nabla\mathbf{f}_x \mathbf{P}_{k-1|k-1} \nabla\mathbf{f}_x^T + \mathbf{Q}_{k-1}.\end{aligned}$$

Measurement Update

1: The updated state and covariance can be obtained as

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}\left[\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\right] \\ \mathbf{P}_{k|k} &= (\mathbf{I}_n - \mathbf{K}_k \nabla\mathbf{h}_x)\,\mathbf{P}_{k|k-1}\end{aligned}$$

where the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T \left(\nabla\mathbf{h}_x \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T + \mathbf{R}_k\right)^{-1}.$$

outputs the range and bearing of the landmarks. While continuing in motion, the vehicle builds a complete map of landmarks. For the successful completion of this mapping task, one of the key steps is to detect the landmarks. The most common sensors used in robotic mapping is laser [37], which outputs the range and bearing of the landmarks. Mostly, these outputs have to be converted to the cartesian coordinates for further analysis and control design.

Consider the polar to cartesian nonlinear transformation given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} \tag{3.8}$$

where $\mathbf{x}$, consisting of range and bearing is $\mathbf{x} = [r \quad \theta]^T$, and $[x \quad y]^T$ are the cartesian coordinates of the target. It is assumed that r and θ are two independent variables with means $\bar{r}$ and $\bar{\theta}$, and the corresponding standard deviations are σ_r and σ_θ, respectively.

The range and bearings in polar coordinates frame can be further written as

$$r = \bar{r} + r_e \tag{3.9}$$

$$\theta = \bar{\theta} + \theta_e \tag{3.10}$$

with $\bar{r} = 1$ and $\bar{\theta} = \frac{\pi}{2}$. r_e and θ_e are the corresponding zero-mean deviations from their means. It is assumed that r_e and θ_e are uniformly distributed[1] between $\pm r_m$ and $\pm\theta_m$, respectively. A similar scenario has been considered in [11].

The means of x and y, $\bar{x}$ and $\bar{y}$, can be obtained by taking the expectations of x and y as given below

$$\begin{aligned}
\bar{x} &= \mathbb{E}(r\cos\theta) \\
&= \mathbb{E}[(\bar{r}+r_e)(\cos(\bar{\theta}+\theta_e))] \\
&= \mathbb{E}[(\bar{r}+r_e)(\cos\bar{\theta}\cos\theta_e - \sin\bar{\theta}\sin\theta_e)] \\
&= \mathbb{E}[-\bar{r}\sin\bar{\theta}\sin\theta_e - r_e\sin\bar{\theta}\sin\theta_e] \\
&= \mathbb{E}[-\sin\theta_e] \qquad (\because r_e \text{ and } \theta_e \text{ are independent}) \\
&= \frac{1}{2\theta_m}[\cos\theta_e]_{-\theta_m}^{\theta_m} \qquad (\text{from } 3.11) \\
&= 0
\end{aligned} \tag{3.12}$$

[1]If a variable x is uniformly distributed between a and b, $\mathbb{U}(a,b)$, then the n^{th} moment of x is

$$\mathbb{E}(x^n) = \frac{1}{b-a}\int_a^b x^n dx. \tag{3.11}$$

and

$$
\begin{aligned}
\bar{y} &= \mathbb{E}(r\sin\theta) \\
&= \mathbb{E}[(\bar{r}+r_e)\sin(\bar{\theta}+\theta_e)] \\
&= \mathbb{E}[\bar{r}\sin\bar{\theta}\cos\theta_e + r_e\sin\bar{\theta}\cos\theta_e + \bar{r}\cos\bar{\theta}\sin\theta_e + r_e\cos\bar{\theta}\sin\theta_e] \\
&= \mathbb{E}(\bar{r}\sin\bar{\theta}\cos\theta_e) \\
&= \mathbb{E}(\cos\theta_e) \\
&= \frac{1}{2\theta_m}[\sin\theta_e]_{-\theta_m}^{\theta_m} \qquad \text{(from 3.11)} \\
&= \frac{\sin\theta_m}{\theta_m} < 1.
\end{aligned} \tag{3.13}
$$

Similarly, the covariance can be obtained as [11]

$$
\begin{aligned}
\mathbf{P}_{x,y} &= \mathbb{E}\left[\begin{bmatrix} (x-\bar{x}) \\ (y-\bar{y}) \end{bmatrix}\begin{bmatrix} (x-\bar{x}) \\ (y-\bar{y}) \end{bmatrix}^T\right] \\
&= \mathbb{E}\left[\begin{bmatrix} r\cos\theta \\ r\sin\theta - \frac{\sin\theta_m}{\theta_m} \end{bmatrix}\begin{bmatrix} r\cos\theta \\ r\sin\theta - \frac{\sin\theta_m}{\theta_m} \end{bmatrix}^T\right] \\
&= \begin{bmatrix} \frac{1}{2}(1+\sigma_r^2)\left(1-\frac{\sin 2\theta_m}{2\theta_m}\right) & 0 \\ 0 & \frac{1}{2}(1+\sigma_r^2)\left(1+\frac{\sin 2\theta_m}{2\theta_m}\right) - \frac{\sin^2\theta_m}{\theta_m^2} \end{bmatrix}
\end{aligned} \tag{3.14}
$$

Simulations were performed to see the true mean and covariance ellipse of the nonlinear polar to cartesian coordinate transformation. From now onwards, the means of x and y given in Eqs. (3.12) and (3.13) are called as true mean and the ellipse formed by the first and fourth elements of the $\mathbf{P}_{x,y}$ given in Eq.(3.14) is called as true ellipse. 2000 measurement samples were generated by taking the true range and bearing values of the target location and adding a zero-mean r_e and θ_e, which are uniform distributed between ± 0.02 and $\pm 20^\circ$. The corresponding plot is shown in Figure 3.1. The range is varying from $\bar{r} \pm r_m$ i.e. (1 ± 0.02) and the bearing is varying from $\bar{\theta} \pm \theta_m$ i.e. $\left(\frac{\pi}{2} \pm 0.3491\text{rad}\right)$.

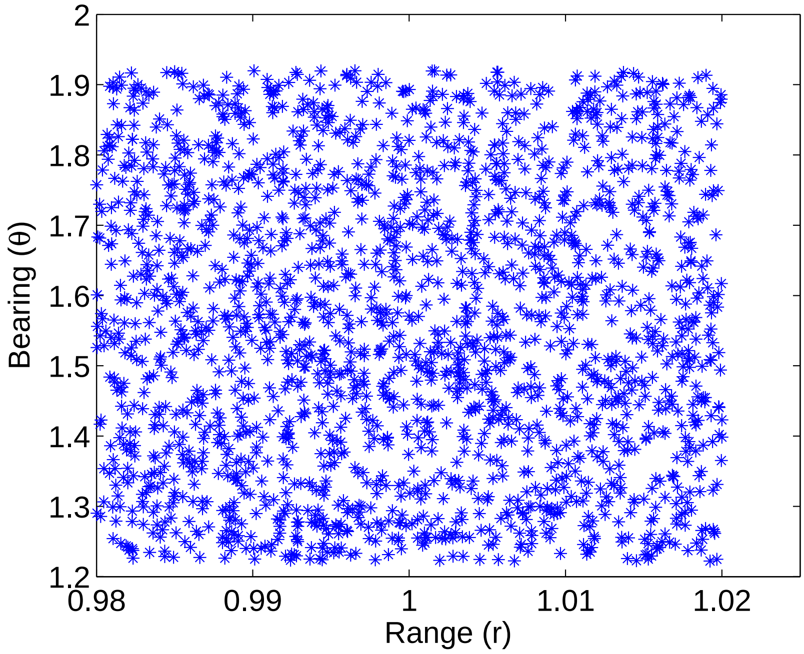

Figure 3.1: 2000 random points ($*$) are generated with range and bearings, which are uniformly distributed between ± 0.02 and $\pm 20°$.

These random points are then processed through the nonlinear polar to cartesian coordinate transformation and are shown in Figure 3.2. The nonlinear mean and the standard deviation ellipse are also shown in Figure 3.2.

From Eqs.(3.12) and (3.13), the mean of x is 0 and for the y is less than 1; the same can be seen in Figure 3.2, where the means of x and y are 0 and 0.9798 (which is less than 1), respectively.

3.2.1.1 Polar to Cartesian Coordinates Transformation: First order linearisation

In this subsection, the polar to cartesian coordinates transformation using first order linearisation will be analysed. The mean of Eq.(3.8) can be obtained by taking the expected values on both sides and can be written as

$$\mathbb{E}\begin{bmatrix} x \\ y \end{bmatrix} = \mathbb{E}\begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} \tag{3.15}$$

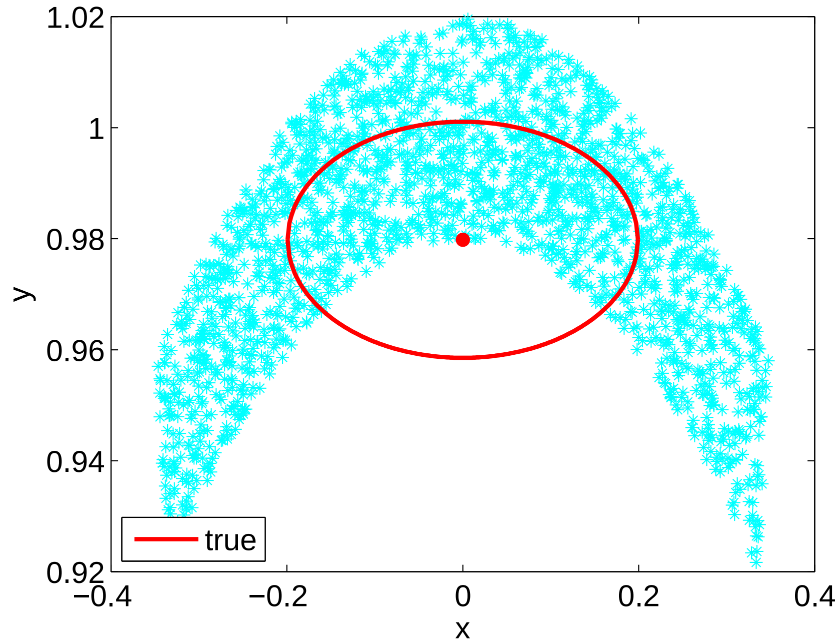

Figure 3.2: 2000 random measurements are generated with range and bearings, which are uniformly distributed between ± 0.02 and $\pm 20°$. These random points are then processed through the nonlinear polar to cartesian coordinate transformation and are shown as $*$. The true mean and the uncertainty ellipse are represented by $\bullet$ and solid line, respectively.

To analyse the effects of linearisation, the nonlinear terms in Eq.(3.15) are expanded using Taylor's series (the second and higher order derivative terms are neglected).

$$\bar{\mathbf{h}}(\mathbf{x}) \simeq \mathbb{E}\left[\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} + \nabla_S\Big|_{\bar{\mathbf{x}}} \begin{pmatrix} x-\bar{x} \\ y-\bar{y} \end{pmatrix}\right] \tag{3.16}$$

$$= \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} + \nabla_S\Big|_{\bar{r},\bar{\theta}} \mathbb{E}\begin{bmatrix} x-\bar{x} \\ y-\bar{y} \end{bmatrix} \tag{3.17}$$

$$= \begin{bmatrix} \bar{r}\cos\bar{\theta} \\ \bar{r}\sin\bar{\theta} \end{bmatrix} \tag{3.18}$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.19}$$

where $\nabla_{\mathbf{s}}\Big|_{\bar{r},\bar{\theta}}$ is the Jacobian of $\mathbf{s}$ evaluated at $(\bar{r},\bar{\theta})$ and is given by

$$\nabla_{\mathbf{s}}\Big|_{\bar{r},\bar{\theta}} = \left.\begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}\right|_{\bar{r},\bar{\theta}} \tag{3.20}$$

$$= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{3.21}$$

The linearised covariance [11] of Eq.(3.8) is

$$\mathbf{P} = \nabla_{\mathbf{s}}\Big|_{\bar{r},\bar{\theta}} \mathbf{P}_{r,\theta} \nabla_{s}\Big|_{\bar{r},\bar{\theta}}^{T} \tag{3.22}$$

where

$$\mathbf{P}_{r,\theta} = \mathbb{E}\left[\begin{pmatrix} r-\bar{r} \\ \theta-\bar{\theta} \end{pmatrix}\begin{pmatrix} r-\bar{r} \\ \theta-\bar{\theta} \end{pmatrix}^{T}\right] \tag{3.23}$$

$$= \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \tag{3.24}$$

and, σ_r and σ_θ are the standard deviations of r and θ, respectively.

The linearised mean and the standard deviation ellipse along with the true mean and true uncertainty ellipse are shown in Figure 3.3. It can be seen that the linearised mean and standard deviation ellipse are not consistent with the true mean and true uncertainty ellipse. The true mean is located at $(0, 0.9798)$, whereas the linearised mean is located at $(0, 1)$. One can see similar linearisation errors in EKF, which uses the first order Jacobians of the state and measurement models.

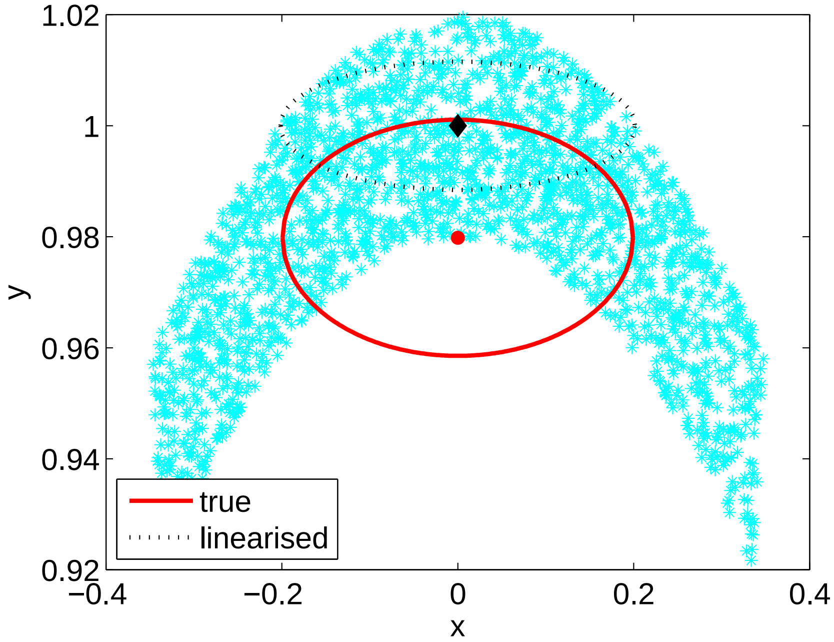

Figure 3.3: 2000 random measurements are generated with range and bearings, which are uniformly distributed between ± 0.02 and $\pm 20°$. These random points are then processed through the nonlinear polar to cartesian coordinate transformation and are shown as $*$. The true mean and the linearised mean are represented by • and ♦, and true and linearised uncertainty ellipses are represented by solid and dotted lines, respectively.

3.3 Unscented Kalman Filter

In Section 3.2.1, effects of the Jacobi linearisation in calculating the mean and covariance of a nonlinear transformation was anlaysed. In this section, a derivative free unscented transformation and unscented Kalman filter (UKF) will be discussed. Unscented transformation is founded on the intuition that "*it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function*" [32]. In unscented transformation, a set of deterministic sigma points are chosen and are propagated through the nonlinear function and then a weighted mean and covariance are evaluated. The unscented transform ensures the higher accuracy than linearisation approach.

3.3.1 Unscented Transformation

Consider the nonlinear function given by $\mathbf{s} = \mathbf{h}(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$. The mean and covariance of the random variable $\mathbf{x}$ are $\bar{\mathbf{x}}$ and $\mathbf{P}_\mathbf{x}$, respectively. The mean and covariance of $\mathbf{s}$, $\bar{\mathbf{s}}$ and $\mathbf{P}_s$, using unscented transformation can be obtained using Algorithm 4.

Algorithm 4 Unscented Transform

1: Compute the $2n+1$ weighted sigma points

$$\begin{aligned} \chi_0 &= \bar{\mathbf{x}} \\ \chi_i &= \bar{\mathbf{x}} + \left[\sqrt{(n+\lambda)\mathbf{P}_\mathbf{x}}\right]_i, \quad i = 1, \ldots, n \\ \chi_i &= \bar{\mathbf{x}} - \left[\sqrt{(n+\lambda)\mathbf{P}_\mathbf{x}}\right]_i, \quad i = n+1, \ldots, 2n \end{aligned} \tag{3.25}$$

where, $[\ldots]_i$ denotes the $i-th$ column of $[\ldots]$. Set the corresponding weights as

$$\begin{aligned} W_0^\mathbf{x} &= \frac{\lambda}{n+\lambda} \\ W_0^P &= \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) \\ W_i^\mathbf{x} &= \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n \\ W_i^P &= W_i^\mathbf{x}, \quad i = 1, \ldots, 2n \end{aligned} \tag{3.26}$$

where

$$\lambda = \alpha^2(n+\kappa) - n. \tag{3.27}$$

The suggested values for α, β and κ are $1 \times 10^{-2} or 1 \times 10^{-3}$, 2 and $3-n$, respectively [46, 33].

2: Propagate the sigma points through the nonlinear function

$$\mathfrak{s}_i = \mathbf{h}(\chi_i), \quad i = 0, \ldots, 2n. \tag{3.28}$$

3: The mean and covariance of $\mathbf{s}$ are

$$\bar{\mathbf{s}} \approx \sum_{i=0}^{2n} W_i^\mathbf{x} \mathfrak{s}_i, \quad i = 0, \ldots, 2n \tag{3.29}$$

$$\mathbf{P}_s \approx \sum_{i=0}^{2n} W_i^P (\mathfrak{s}_i - \bar{\mathbf{s}})(\mathfrak{s}_i - \bar{\mathbf{s}})^T, \quad i = 0, \ldots, 2n.. \tag{3.30}$$

3.3.1.1 Polar to Cartesian Coordinate Transformation - Unscented Transformation

Consider the nonlinear polar to cartesian coordination given in Eq.(3.8). In this section, the unscented transformation given in Algorithm 4 will be used to obtain the mean and covariance of Eq. (3.8). The size of the state vector is $n = 2$, the parameters α, β and κ are selected as 0.01, 2 and 1, respectively and λ using Eq.(3.37) is -1.9997. The remaining parameters used in the simulations are the same as given in Section 3.2.1. The sigma points using Eq.(3.25) can be calculated as

$$\begin{aligned}
\chi_0 &= \begin{bmatrix} \bar{r} \\ \bar{\theta} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\pi}{2} \end{bmatrix} \\
\chi_1 &= \bar{\mathbf{x}} + \left[\sqrt{(n+\lambda)\mathbf{P_x}}\right]_1 = \begin{bmatrix} 1+\sigma_r\sqrt{n+\lambda} \\ \frac{\pi}{2} \end{bmatrix} \\
\chi_2 &= \bar{\mathbf{x}} + \left[\sqrt{(n+\lambda)\mathbf{P_x}}\right]_2 = \begin{bmatrix} 1 \\ \frac{\pi}{2}+\sigma_\theta\sqrt{n+\lambda} \end{bmatrix} \\
\chi_3 &= \bar{\mathbf{x}} - \left[\sqrt{(n+\lambda)\mathbf{P_x}}\right]_3 = \begin{bmatrix} 1-\sigma_r\sqrt{n+\lambda} \\ \frac{\pi}{2} \end{bmatrix} \\
\chi_4 &= \bar{\mathbf{x}} - \left[\sqrt{(n+\lambda)\mathbf{P_x}}\right]_4 = \begin{bmatrix} 1 \\ \frac{\pi}{2}-\sigma_\theta\sqrt{n+\lambda} \end{bmatrix}
\end{aligned} \tag{3.31}$$

where,

$$\mathbf{P_x} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \tag{3.32}$$

The corresponding weights are

$$\begin{aligned}
W_0^{\mathbf{x}} &= \frac{\lambda}{n+\lambda} = -6665.66 &(3.33)\\
W_0^{P} &= \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) = -6662.66 &(3.34)\\
W_1^{\mathbf{x}} &= W_2^{\mathbf{x}} = W_3^{\mathbf{x}} = W_4^{\mathbf{x}} = 1666 &(3.35)\\
W_1^{P} &= W_2^{P} = W_3^{P} = W_4^{P} = 1666 &(3.36)
\end{aligned}$$

where

$$\lambda = \alpha^2(n+\kappa) - n = -1.9997. \tag{3.37}$$

The transformed sigma points using Eq.(3.28) are

$$\begin{aligned}
\mathfrak{s}_0 &= \mathbf{h}(\chi_0) = \mathbf{h}\left(\begin{bmatrix} \bar{r} \\ \bar{\theta} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\\
\mathfrak{s}_1 &= \mathbf{h}(\chi_1) = \mathbf{h}\left(\begin{bmatrix} 1+\sigma_r\sqrt{n+\lambda} \\ \frac{\pi}{2} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1+\sigma_r\sqrt{(n+\lambda)} \end{bmatrix}\\
\mathfrak{s}_2 &= \mathbf{h}(\chi_2) = \mathbf{h}\left(\begin{bmatrix} 1 \\ \frac{\pi}{2}+\sigma_\theta\sqrt{n+\lambda} \end{bmatrix}\right) = \begin{bmatrix} \cos\left(\frac{\pi}{2}+\sigma_\theta\sqrt{(n+\lambda)}\right) \\ \sin\left(\frac{\pi}{2}+\sigma_\theta\sqrt{(n+\lambda)}\right) \end{bmatrix}\\
\mathfrak{s}_3 &= \mathbf{h}(\chi_3) = \mathbf{h}\left(\begin{bmatrix} 1-\sigma_r\sqrt{n+\lambda} \\ \frac{\pi}{2} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1-\sigma_r\sqrt{(n+\lambda)} \end{bmatrix}\\
\mathfrak{s}_4 &= \mathbf{h}(\chi_4) = \mathbf{h}\left(\begin{bmatrix} 1 \\ \frac{\pi}{2}-\sigma_\theta\sqrt{n+\lambda} \end{bmatrix}\right) = \begin{bmatrix} \cos\left(\frac{\pi}{2}-\sigma_\theta\sqrt{(n+\lambda)}\right) \\ \sin\left(\frac{\pi}{2}-\sigma_\theta\sqrt{(n+\lambda)}\right) \end{bmatrix}. &(3.38)
\end{aligned}$$

Once the sigma points and its corresponding weights, and the transformed sigma points are obtained; the mean and covariance of **s** can be calculated as

$$\bar{\mathbf{s}} \approx \sum_{i=0}^{4} W_i^{\mathbf{x}} \mathfrak{s}_i \tag{3.39}$$

$$\mathbf{P}_s \approx \sum_{i=0}^{4} W_i^{P} (\mathfrak{s}_i - \bar{\mathbf{s}})(\mathfrak{s}_i - \bar{\mathbf{s}})^T. \tag{3.40}$$

A similar simulation scenario given in Section 3.2.1 is repeated with unscented transformation and the corresponding results are shown in Figure 3.4. The mean and covariance of unscented transformation along the true and linearised are shown in Figure 3.4. It is very hard to see the true mean as it is hidden behind the unscented transform. The true, linearised and unscented transformation means are located at $(0, 0.9798)$, $(0, 1)$ and $(0, 0.9797)$, respectively. The error covariances for true, linearised and unscented transformation are

$$\begin{bmatrix} 0.1991 & 0 \\ 0 & 0.0213 \end{bmatrix}, \begin{bmatrix} 0.2015 & 0 \\ 0 & 0.0115 \end{bmatrix} \text{and} \begin{bmatrix} 0.2015 & 0 \\ 0 & 0.0310 \end{bmatrix}. \tag{3.41}$$

The uncertainty ellipse using unscented transformation is comparatively unbiased as compared to that of the linearised uncertainty ellipse. It can also be seen that the uncertainty ellipse for unscented transformation given in Figure 3.4 does not match with that of true ellipse along the y-axis. One of the reasons for this mismatch is due to the negative λ. For more details please see the last paragraph of Section 3.4. The unscented transform response can be further improved by tuning α, β and κ.

3.3.2 Unscented Kalman Filter

Unscented Kalman filter (UKF) is a recursive filter based on unscented transformation. In Section 3.3.1, the advantages of unscented transform over the linear approximation were demonstrated.

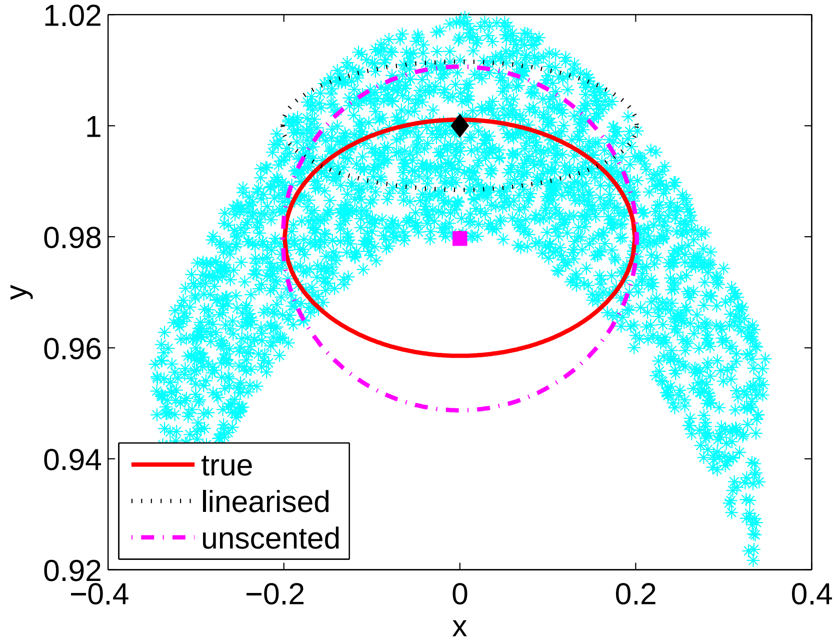

Figure 3.4: 2000 random measurements are generated with range and bearings, which are uniformly distributed between ± 0.02 and $\pm 20°$. These random points are then processed through the nonlinear polar to cartesian coordinate transformation and are shown as $*$. The true, linearised and unscented transformation means are represented by $\bullet$, $\blacklozenge$ and $\blacksquare$, respectively. True, linearised and unscented transformation uncertainty ellipses are represented by solid, dotted and dashed-dotted lines, respectively.

Consider the discrete process and measurement models give in Eqs.(3.1) and (3.2). Similar to EKF, UKF can also be expressed in two stages, prediction and measurement update, and is briefed in Algorithm 5. For more details on UKF, please see, for example [11] and [33].

Algorithm 5 Unscented Kalman Filter

1: Initialise the state vector, $\hat{\mathbf{x}}_{0|0}$, and the covariance matrix, $\mathbf{P}_{0|0}$ (set $k = 1$)

Prediction

2: Calculate the prediction sigma points

$$\begin{aligned}
\chi_{0,k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} \\
\chi_{i,k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \left[\sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}\right]_i, \quad i = 1,\ldots,n \\
\chi_{i,k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} - \left[\sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}\right]_i, \quad i = n+1,\ldots,2n
\end{aligned} \tag{3.42}$$

where λ can be calculated using Eq.(3.27).

3: Propagate the sigma points through the nonlinear process model

$$\chi^*_{i,k|k-1} = \mathbf{f}(\chi_{i,k-1|k-1}, \mathbf{u}_{k-1}), \quad i = 0,\ldots,2n. \tag{3.43}$$

4: Predicted state and covariance can be obtained as

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_i^x \chi^*_{i,k|k-1} \tag{3.44}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_i^P (\chi^*_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\chi^*_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{Q}_{k-1} \tag{3.45}$$

where the weights, W_i^x and W_i^P, can be calculated using Eq.(3.26).

Measurement Update

1: Calculate the update sigma points (these sigma points are calculated using predicted mean and covariance, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$)

$$\begin{aligned}
\chi_{0,k|k-1} &= \hat{\mathbf{x}}_{k|k-1} \\
\chi_{i,k|k-1} &= \hat{\mathbf{x}}_{k|k-1} + \left[\sqrt{(n+\lambda)\mathbf{P}_{k|k-1}}\right]_i, \quad i = 1,\ldots,n \\
\chi_{i,k|k-1} &= \hat{\mathbf{x}}_{k|k-1} - \left[\sqrt{(n+\lambda)\mathbf{P}_{k|k-1}}\right]_i, \quad i = n+1,\ldots,2n
\end{aligned} \tag{3.46}$$

2: Propagate the sigma points through the nonlinear measurement model

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k), \quad i = 0,\ldots,2n. \tag{3.47}$$

3: Predicted measurement, covariance and cross-covariance can be calculated as

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} W_i^x \mathbf{z}_{i,k|k-1} \tag{3.48}$$

$$\mathbf{P}_{zz,k|k-1} = \sum_{i=0}^{2n} W_i^P (\mathbf{z}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})^T + \mathbf{R}_k \tag{3.49}$$

$$\mathbf{P}_{xz,k|k-1} = \sum_{i=0}^{2n} W_i^P (\chi_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})^T \tag{3.50}$$

where the weights, W_i^x and W_i^P, can be calculated using Eq.(3.26).

4: Update mean and error covariance can be obtained as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \tag{3.51}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{zz,k|k-1} \mathbf{K}_k^T \tag{3.52}$$

where the Kalman gain, $\mathbf{K}_k$, is

$$\mathbf{K}_k = \mathbf{P}_{xz,k|k-1} \mathbf{P}_{zz,k|k-1}^{-1}. \tag{3.53}$$

3.4 Cubature Kalman Filter

The CKF is the closest known approximation to the Bayesian filter that could be designed in a nonlinear setting under the Gaussian assumption. Unlike EKF, CKF filter does not require evaluation of Jacobians during the estimation process. EKF require the first order Taylor's series approximation, where the nonlinear functions are approximated by Jacobians, and the UKF performance is completely dominated by the tuning parameters, α, β and κ. Whereas, CKF neither require Jacobians like EKF nor the additional tuning parameters like UKF. Hence CKF is an appealing option for nonlinear state estimation when compared with EKF or UKF [45]. The basic steps required for CKF are described in this section. One can see [45] for more details.

3.4.1 CKF Theory

Consider a nonlinear system with additive noise defined by process and measurement models in (3.1) and (3.2).

The key assumption of the CKF is that the predictive density $p(\mathbf{x}_k|D_{k-1})$, where $D_{k-1} = (\mathbf{u}_l, \mathbf{z}_l)_{l=1}^{k-1}$ denotes the history of input-measurement pairs up to $k-1$, and the filter likelihood density $p(\mathbf{z}_k|D_k)$ are both Gaussian, which eventually leads to a Gaussian posterior density $p(\mathbf{x}_k|D_k)$. Under this assumption, the CKF solution reduces to how to compute their means and covariances more accurately.

The CKF is a two stage procedure comprising of prediction and update.

3.4.1.1 Prediction

In the prediction step, the CKF computes the mean $\hat{\mathbf{x}}_{k|k-1}$ and the associated covariance $\mathbf{P}_{k|k-1}$ of the Gaussian predictive density numerically using cubature rules. The predicted mean can be written as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\left[\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}|D_{k-1}\right] \tag{3.54}$$

Since $\mathbf{w}_{k-1}$ is assumed to be zero-mean and uncorrelated with the measurement sequence, we get

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\left[\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})|D_{k-1}\right] \\
&= \int_{R^n} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1}|D_{k-1}) d\mathbf{x}_{k-1} \\
&= \int_{R^n} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) d\mathbf{x}_{k-1}.
\end{aligned} \tag{3.55}$$

Similarly, the associated error covariance can be represented as

$$\begin{aligned}
\mathbf{P}_{k|k-1} &= \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T|\mathbf{z}_{k-1}\right] \\
&= \int_{R^n} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})\mathbf{f}^T(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})\mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})d\mathbf{x}_{k-1} \\
&\quad -\hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1}.
\end{aligned} \tag{3.56}$$

3.4.1.2 Measurement Update

The predicted measurement density can be represented by

$$p(\mathbf{z}_k|D_{k-1}) = \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, \mathbf{P}_{zz,k|k-1}) \tag{3.57}$$

where the predicted measurement and associated covariance are given by

$$\hat{\mathbf{z}}_{k|k-1} = \int_{R^n} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k \tag{3.58}$$

$$\mathbf{P}_{zz,k|k-1} = \int_{R^n} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T + \mathbf{R}_k \tag{3.59}$$

and the cross-covariance is

$$\mathbf{P}_{xz,k|k-1} = \int_{R^n} \mathbf{x}_k\mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T. \tag{3.60}$$

Once the new measurement $\mathbf{z}_k$ is received, the CKF computes the posterior density $p(\mathbf{x}_k|D_k)$ and can be obtained as

$$p(\mathbf{x}_k|D_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}), \tag{3.61}$$

where

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \tag{3.62}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{zz,k|k-1}\mathbf{K}_k^T \tag{3.63}$$

with the Kalman gain given as

$$\mathbf{K}_k = \mathbf{P}_{xz,k|k-1}\mathbf{P}_{zz,k|k-1}^{-1} \tag{3.64}$$

It can be seen that in the above prediction and measurement update equations, the Bayesian filter solution reduces to computing the multi-dimensional integrals, whose integrands are of the form $nonlinear function \times Gaussian$. The heart of the CKF is to find the multi-dimensional integrals using cubature rules.

3.4.1.3 Cubature Rules

The cubature rule to approximate an n-dimensional Gaussian weighted integral is

$$\int_{R^n} \mathbf{f(x)}\mathcal{N}(\mathbf{x};\mu,\mathbf{P})d\mathbf{x} \approx \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{f}(\mu+\mathbf{P}^{\frac{1}{2}}\xi_i) \tag{3.65}$$

where $\mathbf{P}^{\frac{1}{2}}$ is a square root factor of the covariance $\mathbf{P}$ satisfying the relation $\mathbf{P} = \mathbf{P}^{\frac{1}{2}}\mathbf{P}^{\frac{T}{2}}$; the set of $2n$ cubature points are given by $\{\xi_i\}$ where ξ_i is the $i-th$ element of the following set

$$\sqrt{n}\left[\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix},\ldots,\begin{pmatrix}0\\\vdots\\0\\1\end{pmatrix},\begin{pmatrix}-1\\0\\\vdots\\0\end{pmatrix},\ldots,\begin{pmatrix}0\\\vdots\\0\\-1\end{pmatrix}\right] \tag{3.66}$$

These cubature rules are required to numerically evaluate the multi-integrands in the prediction and update stage of the CKF.
The cubature points required for prediction step are

$$\chi_{i,k|k-1} = \mathbf{P}_{k-1|k-1}^{\frac{1}{2}}\xi_i + \hat{\mathbf{x}}_{k-1|k-1} \tag{3.67}$$

where $i = 1, 2, ..., 2n$ and n is the size of the state vector.

The propagated cubature points through the process model are

$$\chi^*_{i,k|k-1} = \mathbf{f}(\chi_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{3.68}$$

The predicted mean and error covariance matrix from (3.55), (3.56) and (3.65) are

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k|k-1} \tag{3.69}$$

$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k|k-1}\chi^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + \mathbf{Q}_{k-1}. \tag{3.70}$$

By using (3.58)$-$(3.60) and (3.65), the predicted measurement and its associated covariances are

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1} \tag{3.71}$$

$$\mathbf{P}_{zz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} + \mathbf{R}_k \tag{3.72}$$

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} \tag{3.73}$$

where

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k) \tag{3.74}$$

$$\chi_{i,k|k-1} = \mathbf{P}^{\frac{1}{2}}_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1}. \tag{3.75}$$

The updated state and covariance can be obtained using Eqs.(3.62)-(3.64). The CKF is summarised in Algorithm 6.

Algorithm 6 Cubature Kalman Filter

1: Initialise the state vector, $\hat{\mathbf{x}}_{0|0}$, and the covariance matrix, $\mathbf{P}_{0|0}$ (set $k = 1$).

Prediction

2: Factorise the covariance matrix, $\mathbf{P}_{k-1|k-1}$

$$\mathbf{P}_{k-1|k-1} = \mathbf{P}^{\frac{1}{2}}_{k-1|k-1}\mathbf{P}^{\frac{T}{2}}_{k-1,k-1}$$

where $\mathbf{P}^{\frac{1}{2}}_{k-1|k-1}$ is the square root factor of $\mathbf{P}_{k-1|k-1}$.

3: Calculate the cubature points

$$\chi_{i,k-1|k-1} = \mathbf{P}^{\frac{1}{2}}_{k-1|k-1}\xi_i + \hat{\mathbf{x}}_{k-1|k-1}, \quad i = 1,\dots,2n$$

where ξ_i is the $i-th$ element of the following set

$$\sqrt{n}\left[\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix},\dots,\begin{pmatrix}0\\\vdots\\0\\1\end{pmatrix},\begin{pmatrix}-1\\0\\\vdots\\0\end{pmatrix},\dots,\begin{pmatrix}0\\\vdots\\0\\-1\end{pmatrix}\right].$$

4: Propagate the cubature points through the nonlinear process model

$$\chi^*_{i,k-1|k-1} = \mathbf{f}(\chi_{i,k-1|k-1},\mathbf{u}_{k-1}), \quad i = 1,\dots,2n.$$

5: Predicted state and covariance can be obtained as

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k-1|k-1}\\ \mathbf{P}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k-1|k-1}\chi^{*T}_{i,k-1|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + \mathbf{Q}_{k-1}.\end{aligned}$$

Measurement Update

1: Factorise the predicted covariance, $\mathbf{P}_{k|k-1}$

$$\mathbf{P}_{k|k-1} = \mathbf{P}^{\frac{1}{2}}_{k|k-1}\mathbf{P}^{\frac{T}{2}}_{k|k-1}.$$

2: Evaluate the cubature points (these cubature points are calculated using predicted mean and covariance, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$)

$$\chi_{i,k|k-1} = \mathbf{P}^{\frac{1}{2}}_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1}.$$

3: Propagate the cubature points through the nonlinear measurement model

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k).$$

4: Predicted measurement, covariance and cross-covariance can be calculated as

$$\begin{aligned}
\hat{\mathbf{z}}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1} \\
\mathbf{P}_{zz,k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} + \mathbf{R}_k \\
\mathbf{P}_{xz,k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1}
\end{aligned}$$

where the Kalman Gain is

$$\mathbf{K}_k = \mathbf{P}_{xz,k|k-1}\mathbf{P}^{-1}_{zz,k|k-1}.$$

5: Update mean and error covariance can be obtained as

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}^T_{k|k-1}) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{zz,k|k-1}\mathbf{K}^T_k.
\end{aligned}$$

3.4.2 Cubature Transform

Consider the nonlinear function given by $\mathbf{s} = \mathbf{h}(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$. The mean and covariance of the random variable $\mathbf{x}$ are $\bar{\mathbf{x}}$ and $\mathbf{P}_\mathbf{x}$, respectively. The mean and covariance of $\mathbf{s}$, $\bar{\mathbf{s}}$

and $\mathbf{P}_s$, using cubature transformation can be calculated using Algorithm 7.

Algorithm 7 Cubature Transform

1: Compute the $2n$ cubature points

$$\chi_i = \bar{\mathbf{x}} + \mathbf{P}_x^{\frac{1}{2}} \xi_i, \quad i = 1, \ldots, 2n \tag{3.76}$$

where $\mathbf{P}^{\frac{1}{2}}$ is the square root factor or $\mathbf{P}$ and ξ_i is the i^{th} column of

$$\sqrt{n}\left[\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}, \ldots, \begin{pmatrix}0\\\vdots\\0\\1\end{pmatrix}, \begin{pmatrix}-1\\0\\\vdots\\0\end{pmatrix}, \ldots, \begin{pmatrix}0\\\vdots\\0\\-1\end{pmatrix}\right]. \tag{3.77}$$

The corresponding weights are

$$W_i = \frac{1}{2n}, \quad i = 1, \ldots, 2n \tag{3.78}$$

2: Propagate the cubature points through the nonlinear function

$$\mathfrak{s}_i = \mathbf{h}(\chi_i), \quad i = 1, \ldots, 2n. \tag{3.79}$$

3: The mean and covariance for $\mathbf{s}$ can be calculated as

$$\bar{\mathbf{s}} \approx \sum_{i=0}^{2n} W_i \mathfrak{s}_i, \quad i = 0, \ldots, 2n \tag{3.80}$$

$$\mathbf{P}_s \approx \sum_{i=0}^{2n} W_i (\mathfrak{s}_i \mathfrak{s}_i^T - \bar{\mathbf{s}}\bar{\mathbf{s}}^T), \quad i = 0, \ldots, 2n. \tag{3.81}$$

3.4.2.1 Polar to Cartesian Coordinate Transformation - Cubature Transformation

In this section, the polar to cartesian coordinate transformation given in Section 3.2.1 will be analysed using cubature transformation. The cubature transformation given in Algorithm 7 will be used to obtain the mean and covariance of Eq. (3.8). The parameters in this section are the same as given in Section 3.2.1.

The cubature points using Eq.(3.76) can be calculated as:

$$
\begin{aligned}
\chi_1 &= \bar{\mathbf{x}} + \mathbf{P}_x^{\frac{1}{2}}\xi_1 &= \bar{\mathbf{x}} + \left[\sqrt{n}\mathbf{P}_{\mathbf{x}}^{\frac{1}{2}}\right]_1 &= \begin{bmatrix} 1+\sigma_r\sqrt{n} \\ \frac{\pi}{2} \end{bmatrix} \\
\chi_2 &= \bar{\mathbf{x}} + \mathbf{P}_x^{\frac{1}{2}}\xi_2 &= \bar{\mathbf{x}} + \left[\sqrt{n}\mathbf{P}_{\mathbf{x}}^{\frac{1}{2}}\right]_2 &= \begin{bmatrix} 1 \\ \frac{\pi}{2}+\sigma_\theta\sqrt{n} \end{bmatrix} \\
\chi_3 &= \bar{\mathbf{x}} + \mathbf{P}_x^{\frac{1}{2}}\xi_3 &= \bar{\mathbf{x}} - \left[\sqrt{n}\mathbf{P}_{\mathbf{x}}^{\frac{1}{2}}\right]_3 &= \begin{bmatrix} 1-\sigma_r\sqrt{n} \\ \frac{\pi}{2} \end{bmatrix} \\
\chi_4 &= \bar{\mathbf{x}} + \mathbf{P}_x^{\frac{1}{2}}\xi_4 &= \bar{\mathbf{x}} - \left[\sqrt{n}\mathbf{P}_{\mathbf{x}}^{\frac{1}{2}}\right]_4 &= \begin{bmatrix} 1 \\ \frac{\pi}{2}-\sigma_\theta\sqrt{n} \end{bmatrix}
\end{aligned}
\tag{3.82}
$$

The transformed cubature points using Eq.(3.79) are

$$
\begin{aligned}
\mathfrak{s}_1 &= \mathbf{h}(\chi_1) = \mathbf{h}\left(\begin{bmatrix} 1+\sigma_r\sqrt{n} \\ \frac{\pi}{2} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1+\sigma_r\sqrt{(n)} \end{bmatrix} \\
\mathfrak{s}_2 &= \mathbf{h}(\chi_2) = \mathbf{h}\left(\begin{bmatrix} 1 \\ \frac{\pi}{2}+\sigma_\theta\sqrt{n} \end{bmatrix}\right) = \begin{bmatrix} \cos\left(\frac{\pi}{2}+\sigma_\theta\sqrt{(n)}\right) \\ \sin\left(\frac{\pi}{2}+\sigma_\theta\sqrt{(n)}\right) \end{bmatrix} \\
\mathfrak{s}_3 &= \mathbf{h}(\chi_3) = \mathbf{h}\left(\begin{bmatrix} 1 \\ \frac{\pi}{2}+\sigma_\theta\sqrt{n} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1-\sigma_r\sqrt{(n+)} \end{bmatrix} \\
\mathfrak{s}_4 &= \mathbf{h}(\chi_4) = \mathbf{h}\left(\begin{bmatrix} 1 \\ \frac{\pi}{2}-\sigma_\theta\sqrt{n} \end{bmatrix}\right) = \begin{bmatrix} \cos\left(\frac{\pi}{2}-\sigma_\theta\sqrt{(n)}\right) \\ \sin\left(\frac{\pi}{2}-\sigma_\theta\sqrt{(n)}\right) \end{bmatrix}.
\end{aligned}
\tag{3.83}
$$

Once the cubature points and its corresponding weights, and the transformed cubature

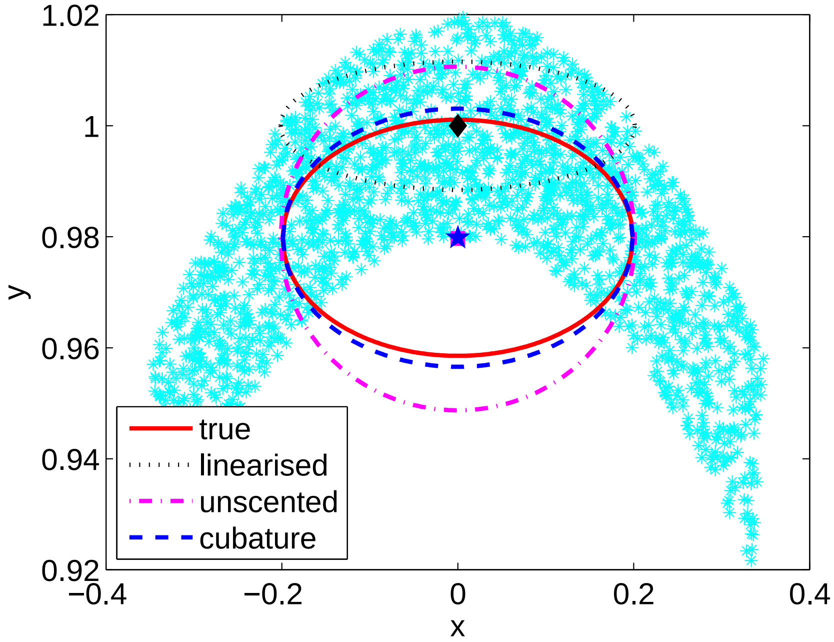

Figure 3.5: 2000 random measurements are generated with range and bearings, which are uniformly distributed between ± 0.02 and $\pm 20°$. These random points are then processed through the nonlinear polar to cartesian coordinate transformation and are shown as $*$. The true-, linearised-, unscented transformation- and cubature transformation means are represented by $\bullet$, $\blacklozenge$, $\blacksquare$ and $\star$, respectively. True, linearised, unscented transformation and cubature transformation uncertainty ellipses are represented by solid, dotted, dashed-dotted and dashed lines, respectively.

points are obtained; the mean and covariance of $\mathbf{s}$ can be calculated as

$$\bar{\mathbf{s}} \approx \sum_{i=1}^{4} W_i \mathfrak{s}_i \tag{3.84}$$

$$\mathbf{P}_s \approx \sum_{i=1}^{4} W_i (\mathfrak{s}_i \mathfrak{s}_i^T - \bar{\mathbf{s}}\bar{\mathbf{s}}^T). \tag{3.85}$$

Simulations in Sections 3.2.1 and 3.3.1 are repeated along with cubature transformation and the corresponding results are show in Figure 3.5. The true mean is at $(0, 0.9798)$, the linearised mean is at $(0, 1)$, the unscented mean is at $(0, 0.9797)$ and the cubature mean is at $(0, 0.9798)$. It is very hard to see the means of different transformations in Figure 3.5 as they are overlapped. By zooming the area around the means, they can be clearly distinguished and are shown in Figure 3.6. The uncertainty ellipse using cubature trans-

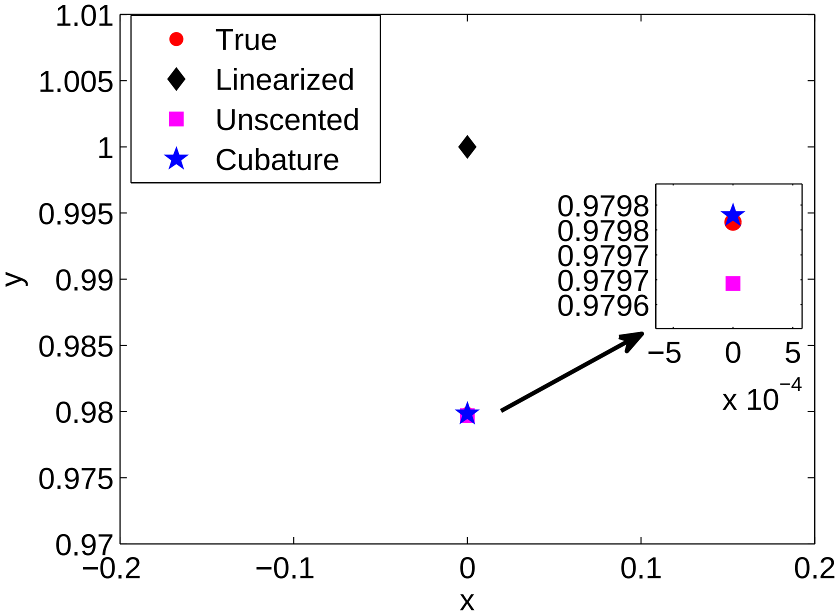

Figure 3.6: Zoomed view of means in Figure 3.5.

formation is very close to the true one and is consistent as compared to that of unscented or linearised uncertainty ellipses. By choosing $\kappa = 0$, unscented transformation response can be matched with cubature transformation response [11].

Although UKF and CKF are derived or proposed from different philosophies, they can be compared in several aspects. UKF uses $2n+1$ sigma points, where as the CKF require $2n$ points. UKF requires more tuning parameters than the CKF (CKF only requires filter initial conditions, and process and measurement covariance matrices, whereas UKF requires extra a few additional parameters). The suggested tuning parameter for UKF is $\kappa = 3 - n$ [33]. If the number of states are more than three, the tuning parameter κ, becomes negative and may halt the UKF operation. It is quite interesting to see that, by using $\kappa = 0$ in the UKF, and $\alpha = \pm 1$, $\beta = 0$ and $\kappa = 0$ in the scaled UKF, the sigma- and cubature-points are the same [45,67]. However, there is no mathematical justification for choosing these parameters for UKF.

3.5 Simultaneous Localisation and Mapping

This section presents the mathematical framework employed in the study of the Simultaneous Localisation and Mapping (SLAM) problem and presents a solution to SLAM using CKF.

Autonomous vehicles are required to determine their own states, since most of their actions (such as surveillance, reconnaissance, autonomous navigation etc.) depends on the state information. Localisation is the process of estimating a vehicle's position and orientation based on landmarks or beacons. Localisation based on an *a priori* map requires the knowledge of the environment defined by the location of different landmarks and hence the environment needs to be explored in advance. As a consequence, the autonomous vehicle is limited to operating within the known environment. If it is necessary to extend the environment of operation, new areas have to be surveyed before autonomous localisation can take place. This difficulty can be overcome by SLAM, see [68] for more details.

The SLAM problem asks: Is it possible for a vehicle to be placed at an unknown location in an unknown environment and to build incrementally a consistent map of the environment while simultaneously determining its location within this map? SLAM has been an active research area for several years and its solution is seen as the "holy grail" by the robotics community [37]. Initial research into this area began with a landmark paper by Smith, Self and Cheeseman which introduced the concept of a stochastic map in which a mobile robot acquires knowledge about its location and organises its environment by making sensor observations in different places and at different times, see [69].

The most common representation used in SLAM is a state-space model with additive Gaussian noise leading to the well known EKF, see [37]. Nonlinear functions are used to represent the process and the measurement model and states are estimated using a recursive process (time update and measurement update). During the state prediction and update stage, linearisation is needed to compute an estimate of the new robot position in the map as well as the correlated position of the landmarks inside the covariance matrix. However, the EKF approach for SLAM is only suitable for 'mild' nonlinearities where first-order approximations of the nonlinear functions are suitable. In order to address problems caused by linearisation, the use of an UKF, appears to be an appealing option, see [33]. The UKF uses the deterministic sampling approach to capture the mean and covariances with sigma points and in general been shown to perform better than the EKF in nonlinear estimation problems. The usage of UKF for large scale outdoor environments SLAM was proposed in [70].

Rao-Blackwellised particle filter (RBPF) is also used to solve the SLAM problem (FastSLAM) [71], [72] and [73]. The UKF and RBPF is fused in SLAM application to form Unscented FastSLAM and deals with some of the limitations of FastSLAM, see [74]. The mean and covariances are updated by using the UKF to avoid the linearisation errors and Jacobian calculations in the feature estimates. The state-dependent Riccati equation (SDRE) filtering has also been used for UAV localisation as an alternative of EKF localisation, see [42].

In this section, we propose the usage of CKF for nonlinear state estimation of SLAM. The augmented state vector of vehicle states and the location of landmarks are estimated using CKF.

3.5.1 The Vehicle, Landmark and Sensor Models

SLAM can be performed by storing the vehicle pose and landmarks in a single state vector, and estimating it by a recursive process of prediction and measurement update. In SLAM, the vehicle starts typically at an unknown location without *a priori* knowledge of landmark locations. The vehicle is mounted with a sensor which is capable of identifying the landmarks. The most common sensor used for SLAM is a laser, which takes the observation of the landmarks and outputs the range and bearing of the landmarks. While continuing in motion, the vehicle builds a complete map of landmarks and uses these to provide estimates of the vehicle location. By using the relative position between the vehicle and landmarks in the environment, both the position of the vehicle and the position of the features or landmarks can be estimated simultaneously.

3.5.1.1 Vehicle Model

Bicycle model is one of the most common vehicle models used for the SLAM application. Several researchers has demonstrated their SLAM solutions in bicycle models [75]. The vehicle model used in this chapter is the common bicycle model, assuming that the control inputs are given by the wheel velocity, V_{k-1}, and steering angle, γ_{k-1} and L is the distance between the front or rear set of wheels and the time interval ΔT denotes the time from $k-1$ to k, see [32], [75] for more details. The vehicle's state vector represents its location and orientation and is given by

$$\begin{bmatrix} \mathbf{x}_{v_k} \end{bmatrix} = \begin{bmatrix} x_{v_k} \\ y_{v_k} \\ \phi_{v_k} \end{bmatrix} = \begin{bmatrix} x_{v_{k-1}} + \Delta T V_{k-1} \cos(\phi_{v_{k-1}} + \gamma_{k-1}) \\ y_{v_{k-1}} + \Delta T V_{k-1} \sin(\phi_{v_{k-1}} + \gamma_{k-1}) \\ \phi_{v_{k-1}} + \Delta T V_{k-1} \frac{\sin(\gamma_{k-1})}{L} \end{bmatrix}$$

In the above equations, the process noise $\mathbf{w}_{k-1}$ is eliminated. One popular way to include the process noise in the process model is to insert the noise terms into the control signal $\mathbf{u}$ such that

$$\mathbf{u}_{k-1} = \mathbf{u}_{n_{k-1}} + \mathbf{w}_{k-1} \tag{3.86}$$

where $\mathbf{u}_{n_{k-1}}$ is a nominal control signal and $\mathbf{w}_{k-1}$ is a zero mean Gaussian distribution noise vector with covariance matrix, $\mathbf{Q}_{k-1}$.

3.5.1.2 Landmark Model

In the context of SLAM, a landmark is a feature of the environment that can be observed using vehicle's sensor. Different kinds of landmark are used in SLAM like point landmarks, corners, lines, etc. For the SLAM algorithm, the feature states are assumed to be stationary. Landmarks can be represented by the following expression

$$\mathbf{x}_{m_k} = \mathbf{x}_{m_{k-1}} \tag{3.87}$$

The SLAM map is defined by an augmented state vector formed by the concatenation of the vehicle and feature map state.

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}_{v_k}^T & \mathbf{x}_{m(k)}^T \end{bmatrix}^T \tag{3.88}$$

3.5.1.3 Sensor Model

It is assumed that the vehicle is equipped with a range-bearing sensor that takes observations of the features of the environment. Laser and sonar sensors are two examples of range-bearing sensors that can be used on a vehicle. Given the current vehicle position $\mathbf{x}_{v_k} = [x_{v_k} \quad y_{v_k}]^T$ and the position of an observed feature $\mathbf{x}_{m_k} = [x_{i_k} \quad y_{i_k}]^T$, the range and bearing can be modelled as

$$\mathbf{z}_{i_k} = \begin{bmatrix} \sqrt{(x_{v_k} - x_{i_k})^2 + (y_{v_k} - y_{i_k})^2} \\ \tan^{-1}\left(\frac{y_{v_k} - y_{i_k}}{x_{v_k} - x_{i_k}}\right) - \phi(k) \end{bmatrix} + \begin{bmatrix} v_{r_k} \\ v_{\theta_k} \end{bmatrix} \tag{3.89}$$

where 'i' denotes the feature number and, v_{r_k} and v_{θ_k} represents the noises in range and bearing measurements.

3.5.2 CKF SLAM

This section describes the use of CKF for estimating the state vector of SLAM. As compared to EKF SLAM, this approach need not requires the evaluation of Jacobians during the prediction and update stages, which makes this approach more promising for achieving the better accuracy. In this chapter, we have used the CKF for state estimation. The CKF SLAM is detailed in Algorithm 8. During the prediction stage, the state vector is augmented with the control inputs and the error covariance matrix with process noise covariance matrix, $\mathbf{Q}_{k-1}$. This augmentation accounts the uncertainty in the control inputs, which then effects the state vector. The estimated state and covariance can then be obtained using the `cub_cal` function. The *cub_cal* function described in Algorithm 9, and *veh_model* represents the state equations of the vehicle model. In the measurement update stage, first the cubature point array and square root factor of the error covariance matrix are obtained as given in steps 1-2. Once the cubature points are evaluated, they are propagated through the nonlinear measurement model (steps 3). Then the predicted measurement, covariance matrices and Kalman gain are evaluated in step 4. The updated state vector and corresponding covariance matrix can be evaluated using step 5.

Algorithm 8 CKF SLAM

Prediction

1: Augment the state vector and covariance matrix

$$\begin{aligned}
\hat{\mathbf{x}}^{\mathbf{v}}_{k-1|k-1} &= \begin{bmatrix} \hat{\mathbf{x}}^{T}_{v,k-1|k-1} & V_{k-1|k-1} & \gamma_{k-1|k-1} \end{bmatrix}^{T} \\
\mathbf{P}^{v}_{k-1|k-1} &= \begin{bmatrix} \mathbf{P}^{v}_{k-1|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k-1} \end{bmatrix}
\end{aligned}$$

2: Predict the state vector and covariance matrix

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} & \mathbf{P}_{k|k-1} \end{bmatrix}$$

using the *cub_cal* function given in Algorithm 9. The inputs to the *cub_cal* function are *veh_model*, $\hat{\mathbf{x}}_{v,k-1|k-1}$, $\mathbf{P}^{v}_{k-1|k-1}$ and the outputs are predicted state vector and covariance matrix.

Measurement Update

1: Factorise the predicted covariance, $\mathbf{P}_{k|k-1}$

$$\mathbf{P}_{k|k-1} = \mathbf{P}^{\frac{1}{2}}_{k|k-1}\mathbf{P}^{\frac{T}{2}}_{k|k-1}.$$

2: Evaluate the cubature points

$$\mathbf{cp}_{i,k|k-1} = \mathbf{P}^{\frac{1}{2}}_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1}.$$

3: Propagate the cubature points through the nonlinear measurement model

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\mathbf{cp}_{i,k|k-1}, \mathbf{u}_k).$$

4: Predicted measurement, covariance and cross-covariance can be calculated as

$$\begin{aligned}
\hat{\mathbf{z}}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1} \\
\mathbf{P}_{zz,k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} + \mathbf{R}_k \\
\mathbf{P}_{xz,k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}\chi^T_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1}
\end{aligned}$$

where the Kalman Gain is

$$\mathbf{K}_k = \mathbf{P}_{xz,k|k-1}\mathbf{P}^{-1}_{zz,k|k-1}.$$

5: Update mean and error covariance can be obtained as

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}^T_{k|k-1}) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{zz,k|k-1}\mathbf{K}^T_k.
\end{aligned}$$

Once the landmarks are detected, they can be augmented with the vehicle states. This can be obtained by using state augmentation algorithm given in Algorithm 10. During the process of state augmentation, first the state vector is augmented with observations and the corresponding error covariance matrix with measurement noise covariance matrix. The augmented state vector of vehicle states and observed landmarks, and the corresponding covariance matrix can be evaluated using which is detailed in Algorithm 10. Once the set of landmarks are observed and augmented in the state vector, the prediction and measurement update given in Algorithm 8 are repeated. One should also note that, once the landmarks are detected, the new augmented state vector with landmarks should be processed, rather than vehicle states alone, in Algorithm 8. The new augmented state vector will now have the vehicle states and the detected landmarks' locations. SLAM efficiency can be improved by revisiting the landmarks and is known as loop closing in SLAM literature.

Algorithm 9 $[\hat{\mathbf{x}}_o \quad \hat{\mathbf{P}}_o]$=`cub_cal(f, x`$_{in}$`, P`$_{in}$`)`

1: Calculate the cubature point array, ξ_i .

2: Factorise the covariance, $\mathbf{P}_{in}$

$$\mathbf{P}_{in} = \mathbf{P}_{in}^{\frac{1}{2}} \mathbf{P}_{in}^{\frac{T}{2}}.$$

3: Evaluate the cubature points

$$\mathbf{cp}_i = \hat{\mathbf{x}}_{in} + \mathbf{P}_{in}^{\frac{1}{2}} \xi_i.$$

4: Propagated the cubature points through nonlinear model

$$\chi_i = \mathbf{f}(\mathbf{cp}_i).$$

5: The state vector and corresponding covariance matrix

$$\begin{aligned} \hat{\mathbf{x}}_o &= \frac{1}{2n} \sum_{i=1}^{2n} \chi_i \\ \mathbf{P}_o &= \frac{1}{2n} \sum_{i=1}^{2n} \chi_i \chi_i^T - \hat{\mathbf{x}} \hat{\mathbf{x}}^T. \end{aligned}$$

3.5.3 Simulation Results

This section includes simulation results of SLAM using EKF, UKF and CKF. The basic SLAM package is available in [76], and is modified for this work. The process and

Algorithm 10 State Augmentation

1: Augment the state vector(with observation) and covariance matrix(with measurement noise covariance matrix)

$$\begin{aligned} \hat{\mathbf{x}}_k^a &= \begin{bmatrix} \hat{\mathbf{x}}_k^T & \mathbf{z}_k^T \end{bmatrix}^T \\ \mathbf{P}_k^a &= \begin{bmatrix} \mathbf{P}_{x,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix}. \end{aligned}$$

2: Evaluate the augmented model

$$aug\_model = \begin{bmatrix} \hat{\mathbf{x}}_k^a \\ x_v + z_r \cos(z_\theta + \phi_v) \\ y_v + z_r \sin(z_\theta + \phi_v) \end{bmatrix}$$

3: The augmented state vector and corresponding covariance matrix, $\hat{\mathbf{x}}_a$ and $\mathbf{P}_a$, can be calculated using the *cub_cal* function given in Algorithm 9. The inputs to the *cub_cal* function are aug_model, $\hat{\mathbf{x}}_k^a$, and $\mathbf{P}_k^a$ and the outputs are $\hat{\mathbf{x}}_a$ and $\mathbf{P}_a$.

observation models used for the simulations are given in Section 3.5.1. In the following numerical experiments, the velocity of the vehicle is $V = 3m/s$, the steering angle range is from $-30^\circ < \gamma < 30^\circ$ and the maximum rate of change in steer angle is 20 deg/sec. The controls are updated at every 0.025 seconds and observations occur at every 0.2 seconds. The range-bearing sensor has a forward-facing 180° field-of-view and maximum range of 30 metres. Similar parameters has also been considered in [75]. In our case, we assumed the landmarks as point features as they are the simplest representation of any landmarks. One can represent the landmarks by lines, etc. The trajectory of the vehicle is known and thirty seven landmarks were randomly spread, the simulation scenario is shown in Figure 3.7. It is assumed that the vehicle starts from origin. Once it detects any landmarks, then those landmarks positions are augmented in the state vector and this process is called as mapping. This process continues until the robot completes its trajectory. During this process, the robot localise itself in the landmarks map, which is called as localisation. In SLAM, the robot performs both localisation and mapping simultaneously.

The measure of the filter consistency is examined over the average error norm ($\bar{J}_k$)

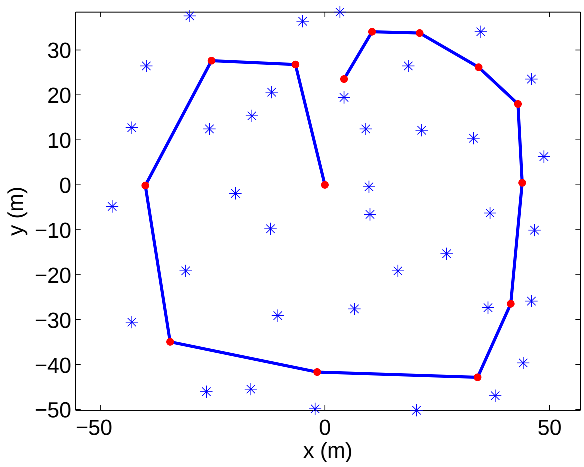

Figure 3.7: Simulation scenario showing the trajectory of the vehicle (solid line) and landmarks ($*$).

over N Monte Carlo simulations. The error norm of the positions is given by

$$J_k^{(i)} = \sqrt{(x_k^i - \hat{x}_k{}^i)^2 + (y_k^i - \hat{y}_k{}^i)^2} \tag{3.90}$$

where 'i' shows the $i-th$ simulation.

500 Monte Carlo simulations were performed for SLAM algorithms for low and high Gaussian noisy environments. For the first scenario, the process and observation noises are $\sigma_v = 0.1m/s, \sigma_\gamma = 1^\circ$, and $\sigma_r = 0.1m, \sigma_\theta = 1^\circ$, respectively. A sample EKF SLAM simulation results with low intensity noises are shown in Figures 3.8 and 3.9. Figure 3.8 shows the reference, actual and EKF estimated trajectories along with the actual and estimated landmarks using EKF SLAM. The reference trajectory (dotted line) in Fgiure 3.8 is generated based on the given way-points, and the actual trajectory (solid line) shows the actual path traveled by the vehicle. The actual path is different from the reference trajectory is due to the constraints imposed on the vehicle parameters like the bounds on maximum steering angle and its rate. The first column in Figure 3.9 shows the actual and estimated states and the second column shows the estimation errors in the three states. In the presence of low intensity noises, the actual and estimated states are very close to each

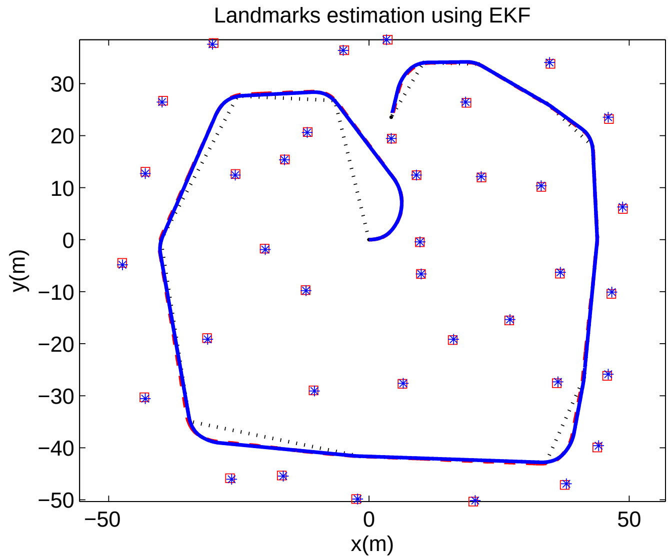

Figure 3.8: Simulation scenario showing the reference (dotted line), actual (solid line) and EKF estimated (dashed line) trajectories in the presence of the low intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

other. The estimation errors in x_v, y_v and ϕ_v using EKF SLAM are 0.5235, 0.4203 and 6.2801, respectively. UKF SLAM with low intensity noises are shown in Figures 3.10 and 3.11. Figure 3.10 shows the reference, actual and UKF estimated trajectories along with the actual and estimated landmarks using UKF SLAM. The selected UKF tuning parameters α, β and κ are selected as 0.001, 2 and $3-n$, respectively. The estimation errors in x_v, y_v and ϕ_v using UKF SLAM are 0.2970, 0.2912 and 6.2792, respectively. CKF SLAM with low intensity noises are shown in Figures 3.12 and 3.13. Figure 3.12 shows the reference, actual and EKF estimated trajectories along with the actual and estimated landmarks using EKF SLAM. The estimation errors in x_v, y_v and ϕ_v using CKF SLAM are 0.1645, 0.1505 and 6.2790, respectively. In the presence of low intensity noises, the estimation errors using CKF SLAM are lower as compared to that of EKF and UKF SLAM. The average RMSE plots of 500 Monte Carlo simulation using EKF, UKF and CKF SLAM are shown in Figure 3.14. The maximum RMSEs for EKF, UKF and CKF SLAM are 0.3390, 0.1446 and 0.0370, respectively.

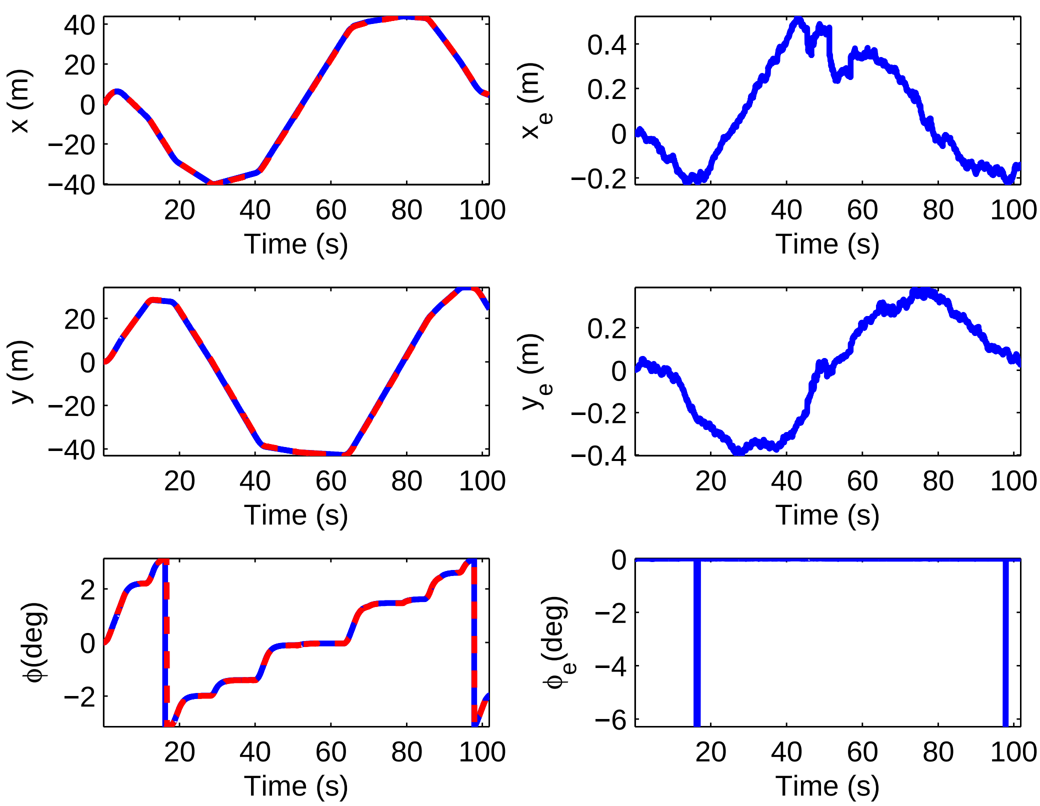

Figure 3.9: EKF SLAM with low intensity noises ($\sigma_v = 0.1m/s, \sigma_\gamma = 1^\circ$ and $\sigma_r = 0.1m, \sigma_\theta = 1^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

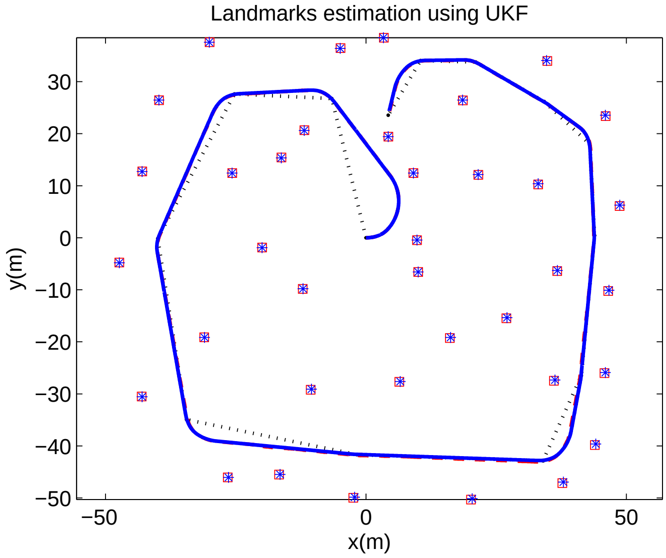

Figure 3.10: Simulation scenario showing the reference (dotted line), actual (solid line) and UKF estimated (dashed line) trajectories in the presence of the low intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

The average error norm over N Monte Carlo simulations is computed as

$$\bar{J}_i = \frac{1}{N}\sum_{i=1}^{N} J_k^{(i)} \tag{3.91}$$

For the second scenario, the process and observation noises are $\sigma_v = 1m/s, \sigma_\gamma = 10°$, and $\sigma_r = 1m, \sigma_\theta = 10°$, respectively. A sample EKF SLAM simulation results with high intensity noises are shown in Figures 3.15 and 3.16. Figure 3.15 shows the reference, actual and EKF estimated trajectories along with the actual and estimated landmarks using EKF SLAM. The first column plots shows the actual and estimated states and the second column shows the estimation errors in the three states. In the presence of low intensity noises, the actual and estimated states are very close to each other. The estimation errors in x_v, y_v and ϕ_v using EKF SLAM are 6.7578, 2.9467 and 6.2882, respectively. UKF SLAM with low intensity noises are shown in Figures 3.17 and 3.18. Figure 3.17 shows the reference, actual and UKF estimated trajectories along with the actual and estimated landmarks using UKF SLAM. The estimation errors in x_v, y_v and ϕ_v using UKF SLAM are 3.0894, 3.7740 and 6.2512, respectively. CKF SLAM with high intensity noises are

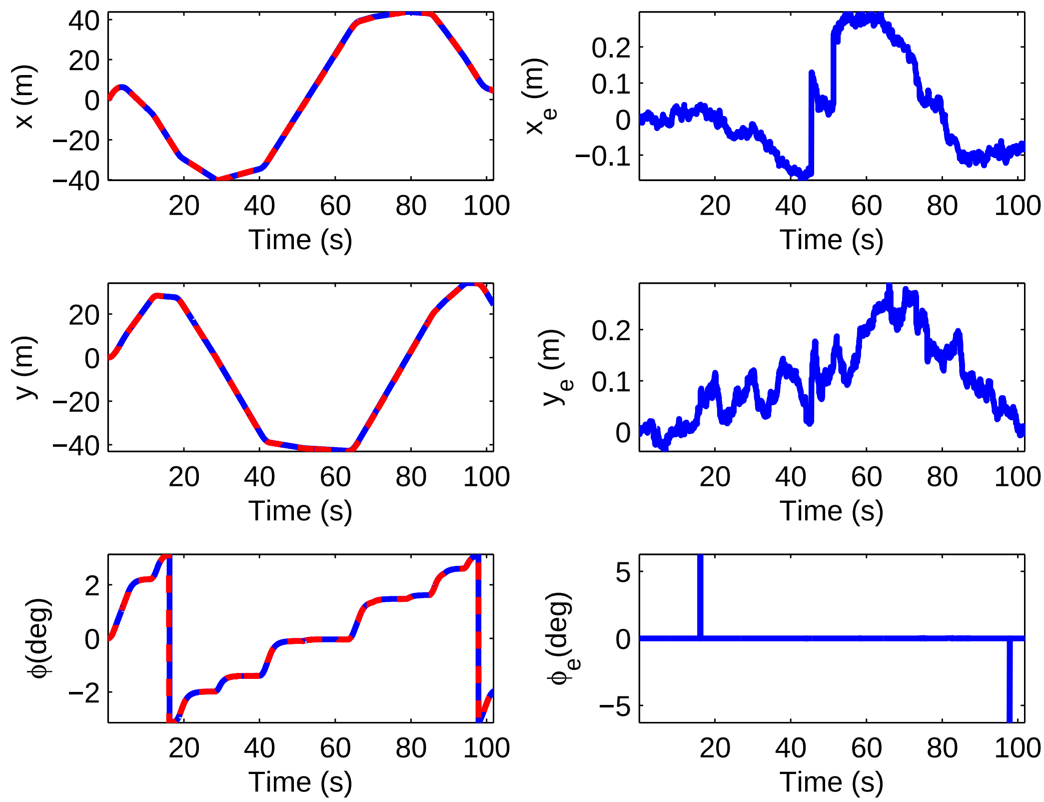

Figure 3.11: UKF SLAM with low intensity noises ($\sigma_v = 0.1m/s, \sigma_\gamma = 1^\circ$ and $\sigma_r = 0.1m, \sigma_\theta = 1^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

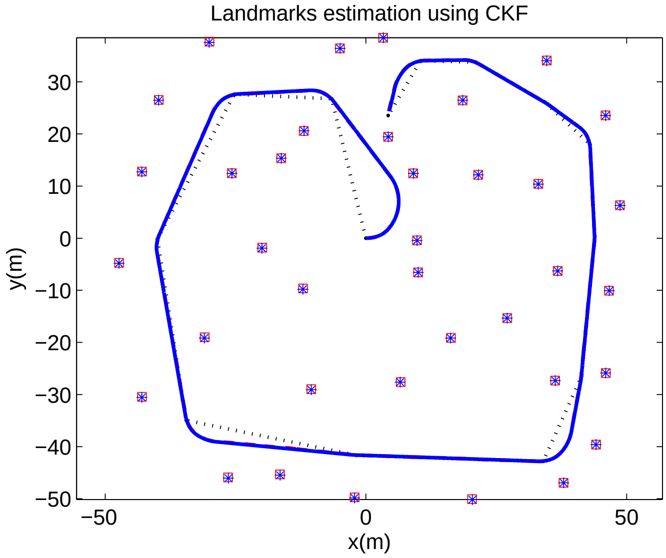

Figure 3.12: Simulation scenario showing the reference (dotted line), actual (solid line) and CKF estimated (dashed line) trajectories in the presence of the low intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

shown in Figures 3.19 and 3.20. Figure 3.19 shows the reference, actual and CKF estimated trajectories along with the actual and estimated landmarks using ECKF SLAM.. The estimation errors in x_v, y_v and ϕ_v using CKF SLAM are 1.5148, 2.7968 and 6.2402, respectively. In the presence of high intensity noises, the estimation errors using CKF SLAM are lower as compared to that of EKF and UKF SLAM. The average RMSE plots of 500 Monte Carlo simulation using EKF, UKF and CKF SLAM are shown in Figure 3.21. The maximum RMSEs for EKF, UKF and CKF SLAM are 45.8431, 22.5368 and 7.8337, respectively.

In some of the Monte Carlo simulations, the vehicle could not able to finish the full trajectory in the UKF SLAM; due to the unavailability a square root factor of the error covariance matrix. The similar instability of UKFs are discussed in [45]. The average simulation times for EKF-, UKF- and CKF-SLAM simulations are 28.43 s, 51.23s and 50.12 s, respectively. EKF SLAM requires the least average simulation time as compared to UKF and CKF SLAM. CKF SLAM is slightly faster than the UKF SLAM; as UKF propagates $2n+1$ sigma points whereas CKF propagates $2n$ cubature points. In all the simulations, the CKF SLAM outperforms the EKF and UKF SLAM.

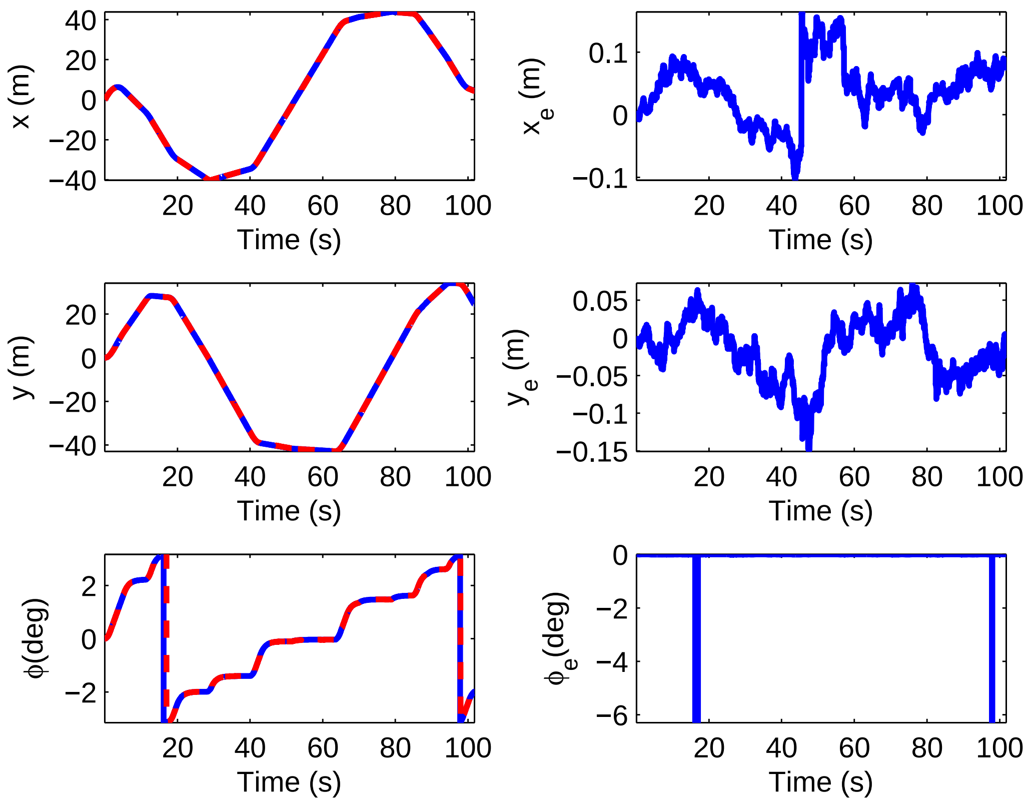

Figure 3.13: CKF SLAM with low intensity noises ($\sigma_v = 0.1m/s, \sigma_\gamma = 1^\circ$ and $\sigma_r = 0.1m, \sigma_\theta = 1^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

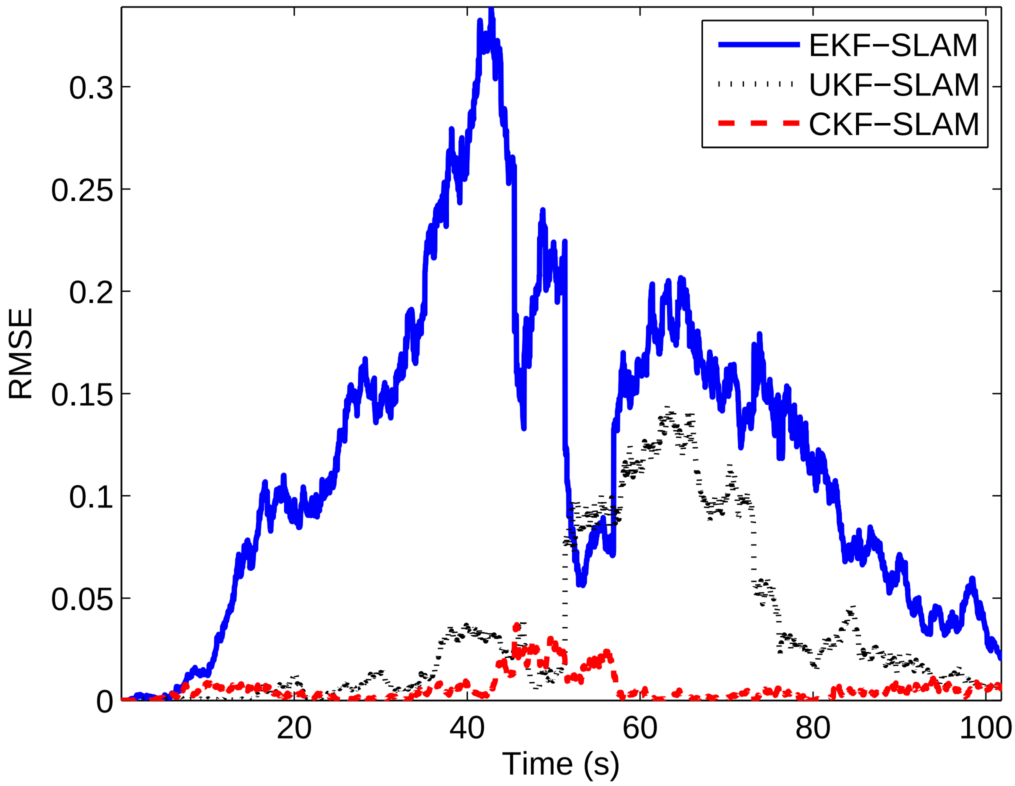

Figure 3.14: Average RMSE of the vehicle positions $[x_v, y_v]^T$ over 500 simulations with $\sigma_v = 0.1m/s, \sigma_\gamma = 1^\circ$ and $\sigma_r = 0.1m, \sigma_\theta = 1^\circ$. Solid, dotted and dashed lines represents EKF, UKF and CKF SLAM, respectively.

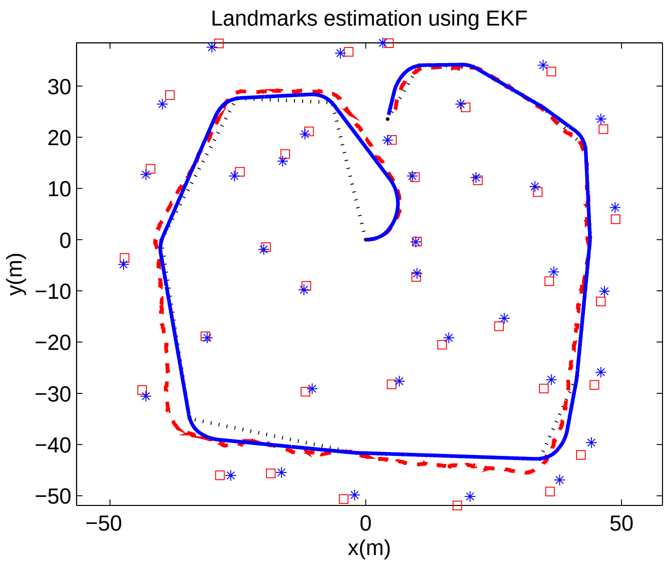

Figure 3.15: Simulation scenario showing the reference (dotted line), actual (solid line) and EKF estimated (dashed line) trajectories in the presence of the high intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

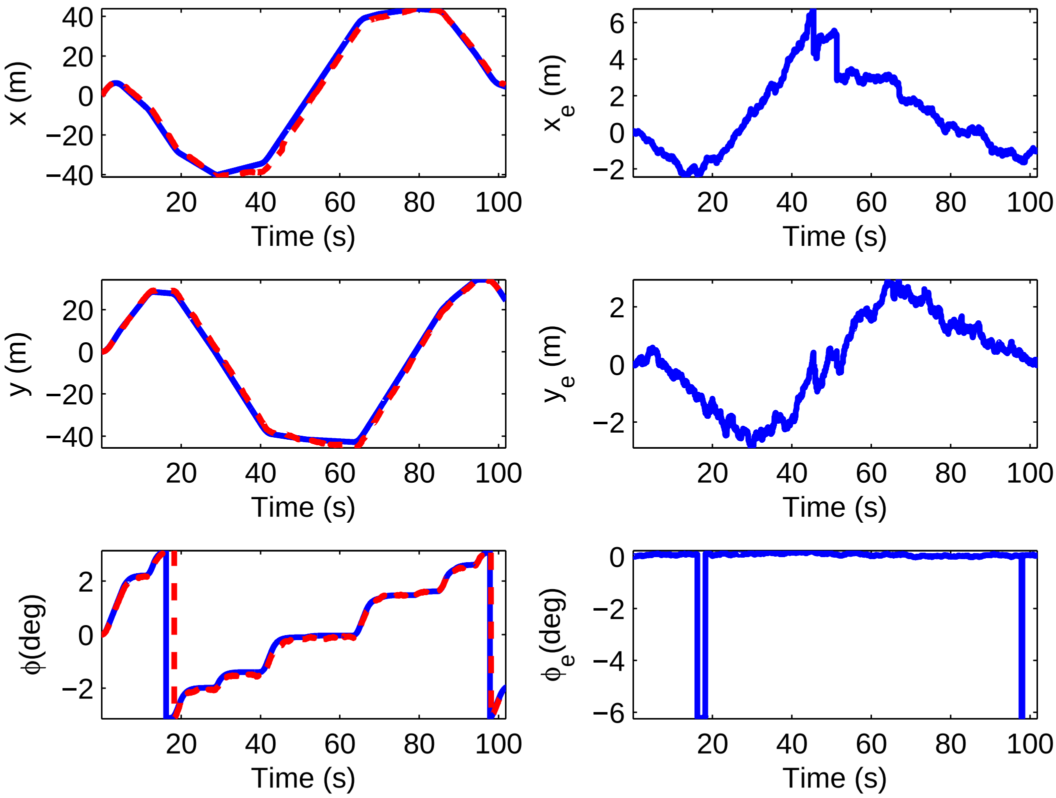

Figure 3.16: EKF SLAM with high intensity noises ($\sigma_v = m/s, \sigma_\gamma = 10^\circ$ and $\sigma_r = m, \sigma_\theta = 10^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

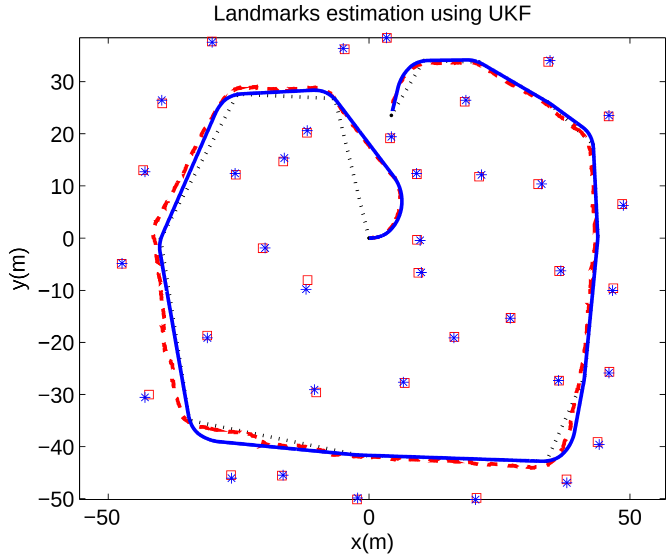

Figure 3.17: Simulation scenario showing the reference (dotted line), actual (solid line) and UKF estimated (dashed line) trajectories in the presence of the high intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

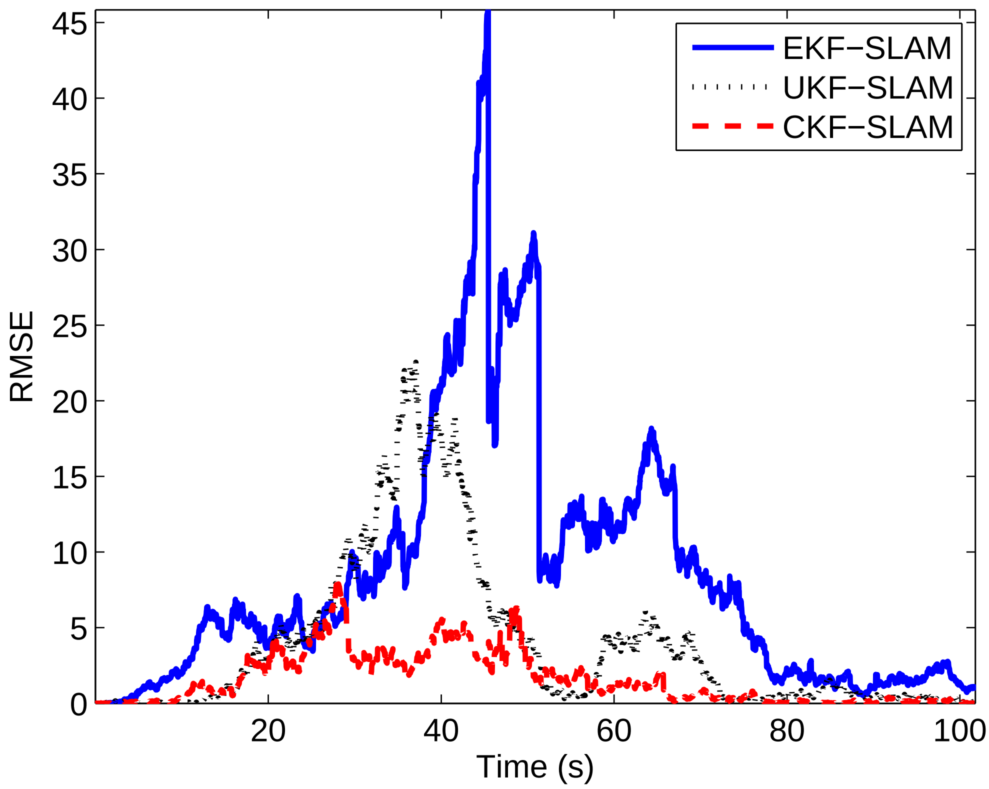

Figure 3.21: Average RMSE of the vehicle positions $[x_v, y_v]^T$ over 500 simulations with $\sigma_v = 1m/s, \sigma_\gamma = 10^\circ$ and $\sigma_r = 1m, \sigma_\theta = 10^\circ$. Solid, dotted and dashed lines represents EKF, UKF and CKF SLAM, respectively.

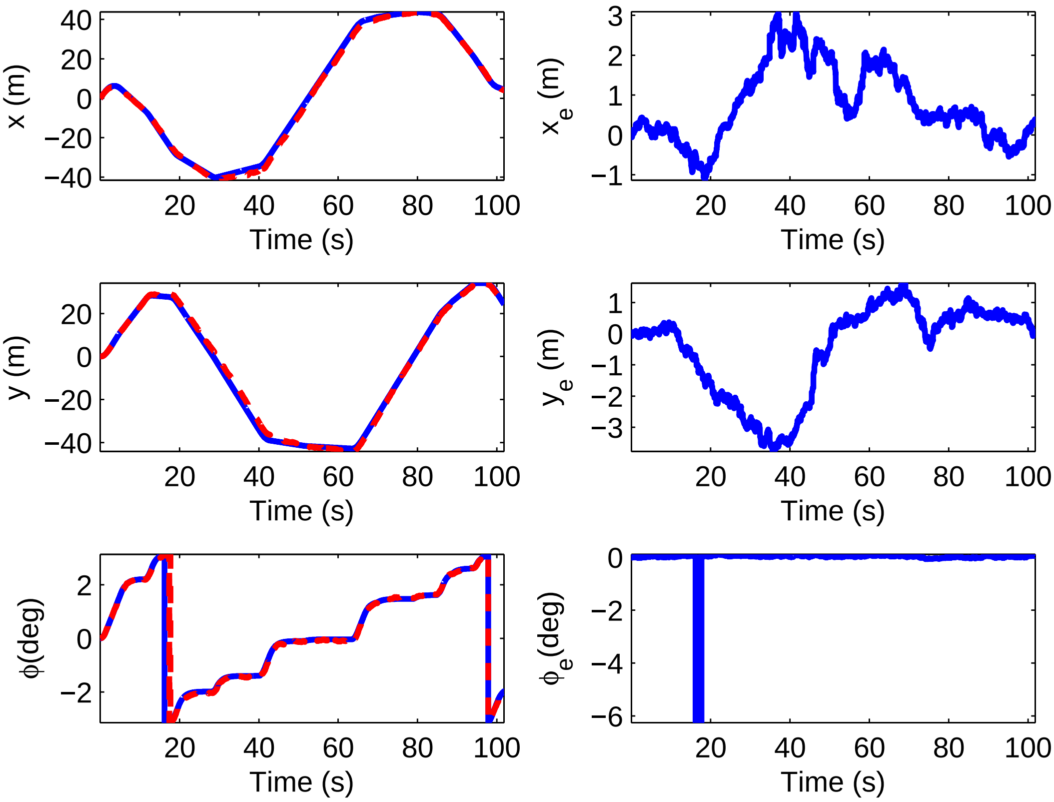

Figure 3.18: UKF SLAM with low intensity noises ($\sigma_v = 1m/s, \sigma_\gamma = 10^\circ$ and $\sigma_r = 1m, \sigma_\theta = 10^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

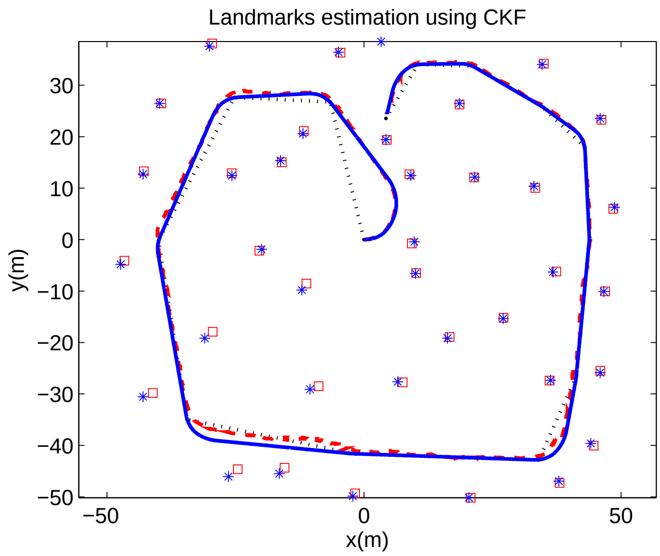

Figure 3.19: Simulation scenario showing the reference (dotted line), actual (solid line) and CKF estimated (dashed line) trajectories in the presence of the high intensity noises. The actual and estimated landmarks are represented by $*$ and $\square$, respectively.

3.6 Conclusions

This chapter has presented and analysed a few nonlinear state estimation methods and their application. The considered nonlinear estimation methods are EKF, UKF and CKF. To analyse the effects of the linearsation and other transforms, polar to cartesian coordinates transformation example was considered. It was shown the means and standard deviation ellipses using unscented and cubature transforms outperforms the nonlinear transformation using linearisation. While estimating means, unscented transform has more estimation error than the cubature transform. These nonlinear state methods are further explored in SLAM problem. We proposed the use of the cubature Kalman filter for SLAM. The proposed algorithm does not requires the evaluation of Jacobians during the prediction and update stage and hence is a derivative free SLAM. The efficacy of the algorithm is verified by simulations. Two types of Gaussian noises are used in the simulations and it was shown that CKF SLAM outperforms EKF and UKF SLAM, in both cases.

In this chapter, estimation algorithms based on single sensor in the presence of Gaus-

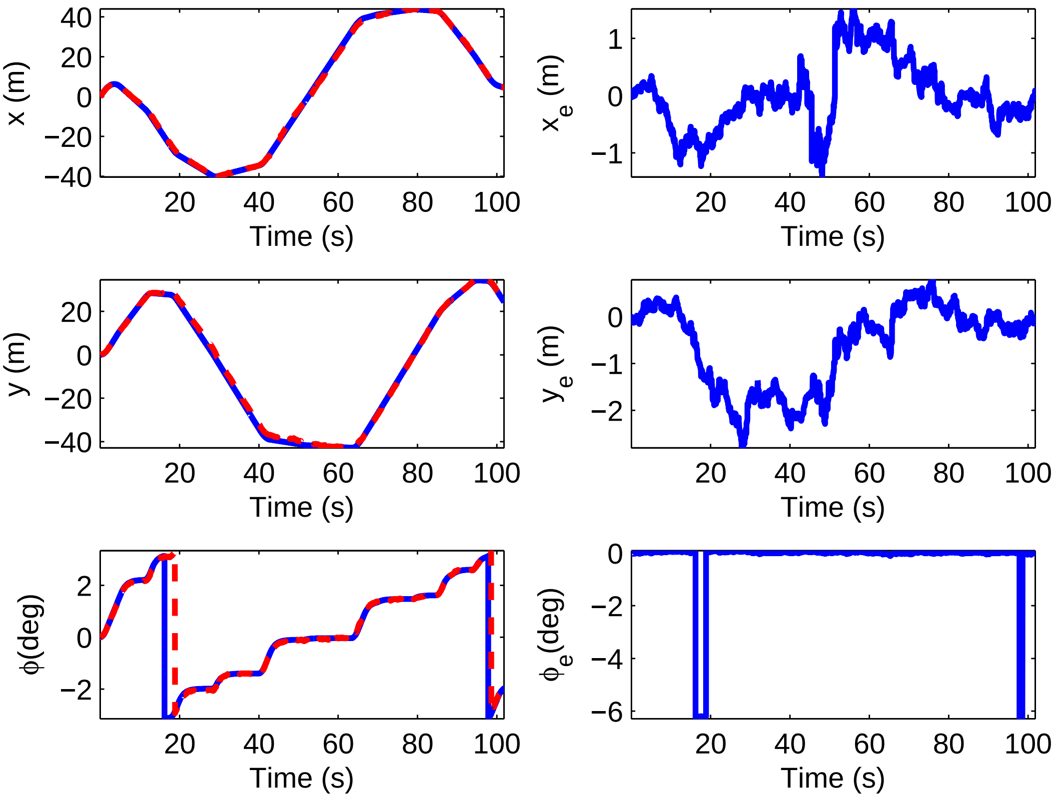

Figure 3.20: CKF SLAM with low intensity noises ($\sigma_v = 1m/s, \sigma_\gamma = 10^\circ$ and $\sigma_r = 1m, \sigma_\theta = 10^\circ$). The solid and dashed lines in the first column represents the actual and estimated vehicle states and the second column shows the corresponding error plots.

sian noises are presented. However, in many real-life applications multi-sensor state estimation provides a better solution and hence in the next chapter, multi-sensor state estimation will be explored.

Chapter 4

Cubature Information Filters

4.1 Introduction

In Chapter 3, extensions of Kalman filter for nonlinear systems were discussed. In those methods, the state vector and the covariance matrix were propagated at different stages to estimate the state vector. An algebraically equivalent form of extended Kalman filter (EKF), the extended information filter (EIF), has been proposed in the literature to cope with some of the issues of EKF [9, 10]. In EIFs, the parameters of interest are the information states and the inverse of covariance rather than states and covariance. Information filters are easy in initialisation compared to conventional Kalman filters, the update stage is computationally economic and it can be easily extended for multi-sensor fusion. Kalman filter can deal with multi-sensor state estimation; but the update stage becomes quite cumbersome during the process of fusing the data from different sensors. Compared to the Kalman filter, the major advantage of information space is its structural and computational simplicity which makes it applicable to multi-sensor and decentralised estimation [10]. One of the key features of information filters are their ability to effectively handle the multi-sensor state estimation. Indeed, EIF has several advantages over EKF; for more details see [9, 10]. However, both EKFs and EIFs are only suitable for 'mild' nonlinearities (where the first-order approximations of the nonlinear functions are available) and they also require evaluation of state Jacobians at every iteration. In this chapter, we propose a cubature information filter (CIF) by embedding cubature Kalman filter (CKF) with an EIF architecture for nonlinear systems. A square root version of cubature information filter (SRCIF), is derived for numerical efficiency. Both CIF and SRCIF are further developed for multi-sensor state estimation. The applicability of the proposed SRCIF is demonstrated on multi-sensor state estimation of a permanent magnet synchronous motor model. The rest of the chapter is structured as follows. Section 4.2 includes the preliminaries of the EIF and some important equations of CKF, and Section 4.3 describes the CIF. Section 4.4 is devoted to SRCIF. Section 4.5 includes numerical simulations and concluding remarks are presented in Section 4.6.

4.2 Extended Information Filter and Cubature Kalman Filter

This section presents a brief introduction to EIF and CKF. For detailed formulations and derivations of these filtering algorithms, please see for example [10] for EIF and [45] for CKF.

4.2.1 Extended information filter

EIF is an algebraic equivalent of EKF, in which the parameters of interest are information states and the inverse of the covariance matrix (information matrix) rather than the states and covariance. EIF can be represented by a recursive process of prediction and measurement updates. The EIF equations are summarised below.

Consider the discrete nonlinear process and measurement models as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{4.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \tag{4.2}$$

where k is the time index, $\mathbf{x}_k$ is the state vector, $\mathbf{u}_k$ is the control input, $\mathbf{z}_k$ is the measurement, $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the process and measurement noises, respectively. These noises are assumed to be zero mean Gaussian-distributed random variables with covariances of $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$.

The conventional filter deals with the estimation of state vector, $\hat{\mathbf{x}}$ along with the corresponding variance matrix, $\mathbf{P}$. Whereas, the information filter deals with the information state, $\mathbf{y}$, and the corresponding information matrix (inverse of the covariance matrix), $\mathbf{Y}$. The predicted information state vector, $\hat{\mathbf{y}}_{k|k-1}$, and the predicted information matrix, $\mathbf{Y}_{k|k-1}$, are given as

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{Y}_{k|k-1}\hat{\mathbf{x}}_{k|k-1} \tag{4.3}$$

$$\mathbf{Y}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1} = \left[\nabla\mathbf{f}_x\mathbf{Y}_{k-1|k-1}^{-1}\nabla\mathbf{f}_x^T + \mathbf{Q}_{k-1}\right]^{-1} \tag{4.4}$$

where $\mathbf{P}_{k|k-1}$ is the predicted covariance matrix and

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}). \tag{4.5}$$

The updated information state vector, $\hat{\mathbf{y}}_{k|k}$, and the updated information matrix, $\mathbf{Y}_{k|k}$, are

$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \mathbf{i}_k \tag{4.6}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \mathbf{I}_k. \tag{4.7}$$

The information state contribution, $\mathbf{i}_k$, and its associated information matrix, $\mathbf{I}_k$, are

$$\mathbf{i}_k = \nabla\mathbf{h}_x^T\mathbf{R}_k^{-1}\left[\nu_k + \nabla\mathbf{h}_x\hat{\mathbf{x}}_{k|k-1}\right] \tag{4.8}$$

$$\mathbf{I}_k = \nabla\mathbf{h}_x^T\mathbf{R}_k^{-1}\nabla\mathbf{h}_x \tag{4.9}$$

where the measurement residual, ν_k, is

$$\nu_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k) \tag{4.10}$$

and $\nabla\mathbf{f}_x$, and $\nabla\mathbf{h}_x$ are the Jacobians of $\mathbf{f}$ and $\mathbf{h}$ evaluated at the best available state (Jacobians for prediction and update equations are evaluated at $\hat{\mathbf{x}}_{k-1|k-1}$ and $\hat{\mathbf{x}}_{k|k-1}$, respectively).

One of the key advantages of the information filter over Kalman filter is the update stage, where the updated information state and information matrix can be obtained by simply adding the associated information contributions to the predicted information state and information matrix. One can refer [10] for a detailed derivation of the information filter.

For the nonlinear information filter, recovery of state and covariance matrices are required at different stages and is an active area of research [77−80]. The state vector and covariance matrix can be recovered by using left division [1] [79]

$$\hat{\mathbf{x}}_{k|k} = \mathbf{Y}_{k|k} \backslash \hat{\mathbf{y}}_{k|k} \tag{4.11}$$

$$\mathbf{P}_{k|k} = \mathbf{Y}_{k|k} \backslash \mathbf{I}_n \tag{4.12}$$

where $\mathbf{I}_n$ is the state vector sized identity matrix. Initialisation in the information space is easier than in the Kalman filter and the update stage of information filter is computationally simpler than the Kalman filter. EIF can be shown to be more efficient than the EKF. But some of the drawbacks inherent in the EKF still affect the EIF. These include the nontrivial nature of the derivations of the Jacobian matrices (and computation) and linearisation instability [10].

[1] $\mathbf{x} = A \backslash B$ solves the least square solution for $A\mathbf{x} = B$ such that $||A\mathbf{x} - b||$ is minimal.

4.2.2 Cubature Kalman filter

Although the CKF has been described in Chapter 3, for completeness some of the key prediction equations required for the CIF's derivation are repeated in this section.

The cubature points required for the prediction step are

$$\chi_{i,k-1|k-1} = \sqrt{\mathbf{P}_{k-1|k-1}}\xi_i + \hat{\mathbf{x}}_{k-1|k-1} \tag{4.13}$$

where $i = 1, 2, ..., 2n$, n is the size of the state vector and ξ_i is the $i-th$ element of the following set

$$\sqrt{n}\left[\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}\right] \tag{4.14}$$

.

The propagated cubature points through the process model are

$$\chi^*_{i,k|k-1} = \mathbf{f}(\chi_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{4.15}$$

The evaluated mean and error covariance matrix are

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k|k-1} \tag{4.16}$$

$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi^*_{i,k|k-1}\chi^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + \mathbf{Q}_{k-1} \tag{4.17}$$

The predicted measurement and its associated covariances are

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1} \tag{4.18}$$

$$\mathbf{P}_{zz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1}\mathbf{z}_{i,k|k-1}^T - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T + \mathbf{R}_k \tag{4.19}$$

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}\mathbf{z}_{i,k|k-1}^T - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T \tag{4.20}$$

where

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k) \tag{4.21}$$

$$\chi_{i,k|k-1} = \sqrt{\mathbf{P}_{k|k-1}}\xi_i + \hat{\mathbf{x}}_{k|k-1}. \tag{4.22}$$

4.3 Cubature Information Filter

This section presents the CIF algorithm, which uses CKF in an EIF framework. The main idea is to derive the prediction step from CKF and the update step from EIF.

Let the information state vector and information matrix be given by $\hat{\mathbf{y}}_{k|k-1}$ and $\mathbf{Y}_{k|k-1}$. The factorisation of the inverse information matrix is required to evaluate $\mathbf{S}_{k-1|k-1}$, which is then required for the propagated cubature points.

$$\left[\mathbf{Y}_{k-1|k-1}\right]^{-1} = \mathbf{S}_{k-1|k-1}\mathbf{S}_{k-1|k-1}^T \tag{4.23}$$

where $\mathbf{S}_{k-1|k-1}$ is a square root factor of $\left[\mathbf{Y}_{k-1|k-1}\right]^{-1}$. The evaluation of cubature points and propagated cubature points can then be given as

$$\chi_{i,k-1|k-1} = \mathbf{S}_{k-1|k-1}\xi_i + \hat{\mathbf{x}}_{k-1|k-1} \tag{4.24}$$

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

where $i = 1, 2, ..., 2n$ and n is the size of the state vector.

The factorisation of the error covariance matrix, the evaluation of cubature points and propagated cubature points for the process model, as required for CIF, are shown in (4.23)−(4.25).

From (4.4) and (4.17), and (4.3) and (4.16)

$$\mathbf{Y}_{k|k-1} = \mathbf{P}^{-1}_{k|k-1} = \left[\frac{1}{2n}\sum_{i=1}^{2n}\chi^{*}_{i,k|k-1}\chi^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^{T}_{k|k-1} + \mathbf{Q}_{k-1}\right]^{-1} \tag{4.26}$$

and

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{P}^{-1}_{k|k-1}\hat{\mathbf{x}}_{k|k-1} \tag{4.27}$$

$$= \mathbf{Y}_{k|k-1}\hat{\mathbf{x}}_{k|k-1} \tag{4.28}$$

$$= \frac{1}{2n}\left[\mathbf{Y}_{k|k-1}\sum_{i=1}^{2n}\chi^{*}_{i,k|k-1}\right]. \tag{4.29}$$

In the measurement update of CIF, the first two steps involve the evaluation of propagated cubature points and the predicted measurement is given below. The propagated cubature points for the measurement model can be evaluated as

$$\chi_{i,k|k-1} = \mathbf{S}_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1} \tag{4.30}$$

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k). \tag{4.31}$$

The predicted measurement is

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{z}_{i,k|k-1}. \tag{4.32}$$

The information state contribution and its associated information matrix in (4.8) and (4.9) are explicit functions of the linearised Jacobian of the measurement model. But the CKF algorithm does not require the Jacobians for measurement update and hence it cannot be directly used in the EIF framework. However, by using the following linear error propagation property [36, 82], it is possible to embed the CKF update in the EIF framework. The linear error propagation property for the error cross covariance matrix can be approximated as [82]

$$\mathbf{P}_{xz,k|k-1} \simeq \mathbf{P}_{k|k-1} \nabla \mathbf{h}_x^T. \tag{4.33}$$

By multiplying $\mathbf{P}_{k|k-1}^{-1}$ and $\mathbf{P}_{k|k-1}$ on the RHS of (4.8) and (4.9) we get

$$\mathbf{i}_k = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{k|k-1} \nabla \mathbf{h}_x^T \mathbf{R}_k^{-1} \left[\nu_k + \nabla \mathbf{h}_x \mathbf{P}_{k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \hat{\mathbf{x}}_{k|k-1} \right] \tag{4.34}$$

$$\mathbf{I}_k = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{k|k-1} \nabla \mathbf{h}_x^T \mathbf{R}_k^{-1} \nabla \mathbf{h}_x \mathbf{P}_{k|k-1}^T \mathbf{P}_{k|k-1}^{-T}. \tag{4.35}$$

Using (4.33) in (4.34) and (4.35) we get

$$\mathbf{i}_k = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{xz,k|k-1} \mathbf{R}_k^{-1} \left[\nu_k + \mathbf{P}_{xz,k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \hat{\mathbf{x}}_{k|k-1} \right] \tag{4.36}$$

$$\mathbf{I}_k = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{xz,k|k-1} \mathbf{R}_k^{-1} \mathbf{P}_{xz,k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \tag{4.37}$$

where

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n} \sum_{i=1}^{2n} \chi_{i,k|k-1} \mathbf{z}_{i,k|k-1}^T - \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{z}}_{k|k-1}^T. \tag{4.38}$$

The updated information state vector and information matrix for the CIF can be obtained by using $\mathbf{i}_k$ and $\mathbf{I}_k$ from (4.36) and (4.37) in (4.6) and (4.7).

One may note that, unlike for the information filter, zero initialisation is not possible in nonlinear information filter. The evaluation of cubature points requires the square root of the covariance matrix. The state vector and covariance matrix can be recovered by (4.11) and (4.12). The CIF algorithm is summarised in Algorithm 11.

4.3.1 CIF in Multi-Sensor State Estimation

One of the main advantages of the information filter is its ability to deal with multi-sensor data fusion [10, 85]. The information from different sensors can be easily fused by simply adding the information contributions to the information matrix and information

Algorithm 11 Cubature Information Filter

Prediction

1: Evaluate the information matrix and the information state vector

$$\begin{aligned}\mathbf{Y}_{k|k-1} &= \mathbf{P}_{k|k-1}^{-1}\\ \hat{\mathbf{y}}_{k|k-1} &= \mathbf{Y}_{k|k-1}\frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k-1|k-1}^{*}\end{aligned}$$

where

$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}^{*}\chi_{i,k|k-1}^{*T} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}_{k|k-1}^{T} + \mathbf{Q}_{k-1}$$

and $\chi_{i,k|k-1}$ can be obtained from (4.25).

Measurement Update

1: Evaluate the information state contribution and its associated information matrix

$$\begin{aligned}\mathbf{I}_k =& \mathbf{Y}_{k|k-1}\mathbf{P}_{xz,k|k-1}\mathbf{R}_k^{-1}\mathbf{P}_{xz,k|k-1}^{T}\mathbf{Y}_{k|k-1}^{T}\\ \mathbf{i}_k =& \mathbf{Y}_{k|k-1}\mathbf{P}_{xz,k|k-1}\mathbf{R}_k^{-1}\left[\nu_k + \mathbf{P}_{xz,k|k-1}^{T}\mathbf{Y}_{k|k-1}^{T}\hat{\mathbf{x}}_{k|k-1}\right]\end{aligned}$$

where

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\chi_{i,k|k-1}\mathbf{z}_{i,k|k-1}^{T} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^{T}.$$

2: The estimated information vector and information matrix of CIF are

$$\begin{aligned}\mathbf{Y}_{k|k} &= \mathbf{Y}_{k|k-1} + \mathbf{I}_k\\ \hat{\mathbf{y}}_{k|k} &= \hat{\mathbf{y}}_{k|k-1} + \mathbf{i}_k.\end{aligned}$$

The state and covariance can be recovered using (4.11) and (4.12).

vector [10, 85]. In multi-sensor state estimation, the available observations consist of measurements taken from different sensors. The prediction step for multi-sensor state estimation is similar to that of the Kalman or information filter. In the measurement update step, the data from different sensors are fused for an efficient and reliable estimation [81].

Let the different sensors used for state estimation be given by

$$\mathbf{z}_{j,k} = \mathbf{h}_{j,k}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_{j,k}; \qquad j = 1,2,...D \tag{4.39}$$

where 'D' is the number of sensors.

The CIF algorithm can be easily extended for multi-sensor data fusion in which the basic update step of CIF is similar to EIF [10]. The updated information vector and information matrix for multi-sensor CIF are

$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \sum_{j=1}^{D} \mathbf{i}_{j,k} \tag{4.40}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{j=1}^{D} \mathbf{I}_{j,k}. \tag{4.41}$$

By using (4.36) and (4.37), the information contributions of multi-sensor CIF are

$$\mathbf{I}_{j,k} = \mathbf{M}_{j,k|k-1}^{T} \mathbf{R}_{j,k}^{-1} \mathbf{M}_{j,k|k-1} \tag{4.42}$$

$$\mathbf{i}_{j,k} = \mathbf{M}_{j,k|k-1}^{T} \mathbf{R}_{j,k}^{-1} [\nu_{j,k} + \mathbf{M}_{j,k|k-1} \hat{\mathbf{x}}_{k|k-1}] \tag{4.43}$$

where

$$\mathbf{M}_{j,k|k-1}^{T} = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{j,xz,k|k-1}. \tag{4.44}$$

4.4 Square Root Cubature Information Filter

This section presents a brief description of the square root extended information filter (SREIF) and a derivation of the square root cubature information filter (SRCIF). One of the most stable and numerically reliable implementations of the Kalman filter is its square root version [8]. In this the square root covariance matrix is propagated to make the overall filter robust against round-off errors. Some of the key properties of the square root filter are: symmetric positive definite error covariances, availability of square root factors, doubled order precision, improved numerical accuracy [29, 8, 11, 45]. Similarly, in the information domain square root versions of information filters are preferred [29]. These added advantages of square root filters are the motivation for the development of SRCIF.

The following notation is used throughout this chapter. Given a positive definite matrix F, then F and F^{-1} can be factorised as

$$\mathbf{F} = (\mathbf{F}^{1/2})(\mathbf{F}^{T/2}) \tag{4.45}$$

$$\mathbf{F}^{-1} = (\mathbf{F}^{-T/2})(\mathbf{F}^{-1/2}). \tag{4.46}$$

4.4.1 Square Root Extended Information Filter

In this subsection, the square root extended information filter (SREIF), which is required for the SRCIF derivation is briefly discussed. The prediction step of SREIF is not required in this work and hence it is not presented. For more details on SREIF see [29, 6]. The measurement update step for SREIF is [6]

$$\begin{bmatrix} \mathbf{P}_{k|k-1}^{-T/2} & \nabla\mathbf{h}_x^T \mathbf{R}_k^{-T/2} \\ \hat{\mathbf{x}}_{k|k-1}^T \mathbf{P}_{k|k-1}^{-T/2} & \mathbf{z}_k^T \mathbf{R}_k^{-T/2} \end{bmatrix} \Theta = \begin{bmatrix} \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \hat{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{-T/2} & \star \end{bmatrix} \tag{4.47}$$

and in information space, it is

$$\begin{bmatrix} \mathbf{Y}_{S,k|k-1} & \nabla\mathbf{h}_x^T \mathbf{Y}_{\mathbf{R}} \\ \hat{\mathbf{y}}_{S,k|k-1}^T & \mathbf{z}_k^T \mathbf{Y}_{\mathbf{R}} \end{bmatrix} \Theta = \begin{bmatrix} \mathbf{Y}_{S,k|k} & \mathbf{0} \\ \hat{\mathbf{y}}_{S,k|k}^T & \star \end{bmatrix} \tag{4.48}$$

where $\hat{\mathbf{y}}_S$, $\mathbf{Y}_S$ and $\mathbf{Y}_{\mathbf{R}}$ are the square root factors[2] of $\mathbf{y}$, $\mathbf{Y}$ and $\mathbf{R}^{-1}$, respectively. '$\star$' represents the terms which are irrelevant for SREIF and Θ is a unitary matrix which can be found using Givens rotations or Householder reflections[3] [8]. If Θ is partitioned

[2]Square root factors of information matrix and information state

$$\begin{aligned} \mathbf{P}^{-1} &= \mathbf{P}^{-T/2}\mathbf{P}^{-1/2} \\ \Rightarrow \mathbf{Y} &= \mathbf{Y}_s \mathbf{Y}_s^T \\ \mathbf{P}^{-1}\mathbf{x} &= \mathbf{P}^{-T/2}\mathbf{P}^{-1/2}\mathbf{x} \\ \Rightarrow \mathbf{y} &= \mathbf{Y}_s \mathbf{y}_s. \end{aligned}$$

[3]The basic structure of the Householder matrix, Θ, is

$$\Theta = \mathbf{I} - \frac{2}{c^T c} c c^T$$

as $[\Theta_1 \quad \Theta_2; \Theta_3 \quad \Theta_4]$, then the updated information state vector and the corresponding information matrix can be written as

$$\begin{aligned} \hat{\mathbf{y}}_{S,k|k}^T &= \mathbf{y}_{S,k|k-1}^T \Theta_1 + \mathbf{z}_k^T \mathbf{Y}_{\mathbf{R}} \Theta_3 && (4.50) \\ \mathbf{Y}_{S,k|k} &= \mathbf{Y}_{S,k|k-1} \Theta_1 + \nabla \mathbf{h}_x^T \mathbf{Y}_{\mathbf{R}} \Theta_3. && (4.51) \end{aligned}$$

The measurement update stage of SREIF can be extended for multi-sensor state estimation [81]. In this, the data from different sensors are fused for an efficient and reliable estimation.

The measurement update step for SREIF using 'D' sensors is

$$\begin{bmatrix} \mathbf{Y}_{S,k|k-1} & \nabla \mathbf{h}_{1,k}^T \mathbf{Y}_{\mathbf{R},1,k} & \nabla \mathbf{h}_{2,k}^T \mathbf{Y}_{\mathbf{R},2,k} & \cdots & \nabla \mathbf{h}_{D,k}^T \mathbf{Y}_{\mathbf{R},D,k} \\ \hat{\mathbf{y}}_{S,k|k-1}^T & \mathbf{z}_{1,k}^T \mathbf{Y}_{\mathbf{R},1,k} & \mathbf{z}_{2,k}^T \mathbf{Y}_{\mathbf{R},2,k} & \cdots & \mathbf{z}_{D,k}^T \mathbf{Y}_{\mathbf{R},D,k} \end{bmatrix} \Theta = \begin{bmatrix} \mathbf{Y}_{S,k|k} & \mathbf{0} \\ \hat{\mathbf{y}}_{S,k|k}^T & \star \end{bmatrix}. \quad (4.52)$$

4.4.2 Square Root Cubature Information Filter

In this subsection, the square root information filter for nonlinear systems is derived. The proposed algorithm is derived from SREIF [29] and CKF [45], and is called as square root cubature information filter. The square root factors of covariance matrices can be found using the **QR**[4] decomposition and the *leftdivide* operator. Furthermore, this approach can be extended to multi-sensor data fusion.

4.4.2.1 SRCIF Prediction

1: Evaluate the cubature points

$$\chi_{i,k-1|k-1} = \mathbf{P}_{k-1|k-1}^{1/2} \xi_i + \mathbf{x}_{k-1|k-1}. \quad (4.53)$$

where c is a column vector and $\mathbf{I}$ is the identity matrix of the same dimension.

[4] **QR** is orthogonal triangular decomposition and can be found in MATLAB using the command 'qr'.

2: Evaluate the propagated cubature points

$$\chi^*_{i,k|k-1} = \mathbf{f}(\chi_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{4.54}$$

3: Estimate the predicted state

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \chi^*_{i,k|k-1}. \tag{4.55}$$

4: Estimate the square root factor of the predicted error covariance and information matrix

$$\mathbf{S}_{k|k-1} = \left[\mathbf{qr}\left(\mathscr{X}^*_{i,k|k-1} \quad \mathbf{Y}_{Q,k-1}\right)^T\right]^T \tag{4.56}$$

$$\mathbf{Y}_{S,k|k-1} = \mathbf{S}_{k|k-1}\backslash\mathbf{I} \tag{4.57}$$

where $\mathbf{Y}_{Q,k-1}$ is a square root factor of $\mathbf{Q}_{k-1}$ and

$$\mathscr{X}^*_{i,k|k-1} = \frac{1}{\sqrt{2n}}\left[\chi^*_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \quad \chi^*_{2,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \ldots \quad \chi^*_{2n,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]. \tag{4.58}$$

5: Evaluate the square root information state vector

$$\hat{\mathbf{y}}_{S,k|k-1} = \mathbf{Y}_{S,k|k-1}\hat{\mathbf{x}}_{k|k-1}. \tag{4.59}$$

4.4.2.2 SRCIF Measurement Update

From (4.47), one can see that the measurement update stage of SREIF requires the linearised measurement model, $\nabla\mathbf{H}$. By using the below statistical error propagation property [82], the derivative-less measurement update step for the SRCIF can be derived.

$$
\begin{aligned}
\mathbf{P}_{zz} &= cov(\mathbf{z}) \\
&= E[(\mathbf{z}-\hat{\mathbf{z}})(\mathbf{z}-\hat{\mathbf{z}})^T] \\
&= E[\nabla\mathbf{h}_x\mathbf{x}-\nabla\mathbf{h}_x\hat{\mathbf{x}})(\nabla\mathbf{h}_x\mathbf{x}-\nabla\mathbf{h}_x\hat{\mathbf{x}})^T] \\
&= \nabla\mathbf{h}_x cov(\mathbf{x})\nabla\mathbf{h}_x^T \\
&= \nabla\mathbf{h}_x\mathbf{P}\nabla\mathbf{h}_x^T.
\end{aligned} \tag{4.60}
$$

Let $\mathbf{P}_{zz}^{1/2}$ and $\mathbf{P}^{1/2}$ denote the square root factor of $\mathbf{P}_{zz}$ and $\mathbf{P}$. Then (4.60) can be expressed as

$$\mathbf{P}_{zz}^{1/2}\mathbf{P}_{zz}^{T/2} = \nabla\mathbf{h}_x\mathbf{P}^{1/2}\mathbf{P}^{T/2}\nabla\mathbf{h}_x^T \tag{4.61}$$

and the covariance matrix in (4.61) can be factorised as

$$\mathbf{P}_{zz}^{1/2} = \nabla\mathbf{h}_x\mathbf{P}^{1/2} \tag{4.62}$$

$$\mathbf{P}_{zz}^{T/2} = \mathbf{P}^{T/2}\nabla\mathbf{h}_x^T. \tag{4.63}$$

Pre-multiplying $\nabla\mathbf{h}_x^T\mathbf{Y}_R$ by $\mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{k|k-1}^{T/2}$ gives

$$
\begin{aligned}
\nabla\mathbf{h}_x^T\mathbf{Y}_R &= \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{k|k-1}^{T/2}\nabla\mathbf{h}_x^T\mathbf{Y}_R \\
&= \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{Y}_R.
\end{aligned} \tag{4.64}
$$

From (4.48) and (4.64) we get

$$\begin{bmatrix} \mathbf{Y}_{S,k|k-1} & \mathbf{Y}_{M,k|k-1} \\ \hat{\mathbf{y}}_{S,k|k-1}^T & \mathbf{z}_k^T\mathbf{Y}_R \end{bmatrix} \Theta \;=\; \begin{bmatrix} \mathbf{Y}_{S,k|k} & \mathbf{0} \\ \hat{\mathbf{y}}_{S,k|k}^T & \star \end{bmatrix} \tag{4.65}$$

where

$$\mathbf{Y}_{M,k|k-1} = \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{Y}_R. \tag{4.66}$$

As in (4.56), the square root factor of the predicted error covariance matrix is evaluated. Similarly, the square root factor of the measurement error covariance matrix, $\mathbf{P}^{T/2}_{zz,k|k-1}$, can be evaluated. The estimated state vector and the square root factor of the error covariance matrix required for SRCIF prediction can be recovered using (4.65), (4.11) and (4.12). The derived SRCIF can then be extended to multi-sensor data fusion. The prediction stage of the SRCIF in multi-sensor state estimation is similar to SRCIF.

By using (4.64), (4.66) and (4.52), the multi-sensor SRCIF measurement update step for multi-sensor SRCIF is

$$\begin{bmatrix} \mathbf{Y}_{S,k|k-1} & \mathbf{Y}_{1,M,k|k-1} & \mathbf{Y}_{2,M,k|k-1} & \cdots & \mathbf{Y}_{D,M,k|k-1} \\ \hat{\mathbf{y}}^T_{S,k|k-1} & \mathbf{z}^T_{1,k}\mathbf{Y}_{\mathbf{R},1,k} & \mathbf{z}^T_{2,k}\mathbf{Y}_{\mathbf{R},2,k} & \cdots & \mathbf{z}^T_{D,k}\mathbf{Y}_{\mathbf{R},D,k} \end{bmatrix} \Theta = \begin{bmatrix} \mathbf{Y}_{S,k|k} & \mathbf{0} \\ \hat{\mathbf{y}}^T_{S,k|k} & \star \end{bmatrix}. \tag{4.67}$$

The derivation of the update stage of the multi-sensor SRCIF is given below. By expanding LHS of (4.67), yields (4.68)

$$\begin{bmatrix} \mathbf{Y}_{S,k|k-1} & \mathbf{Y}_{1,M,k|k-1} & \mathbf{Y}_{2,M,k|k-1} & \cdots & \mathbf{Y}_{D,M,k|k-1} \\ \hat{\mathbf{y}}^T_{S,k|k-1} & \mathbf{z}^T_{1,k}\mathbf{Y}_{\mathbf{R},1,k} & \mathbf{z}^T_{2,k}\mathbf{Y}_{\mathbf{R},2,k} & \cdots & \mathbf{z}^T_{D,k}\mathbf{Y}_{\mathbf{R},D,k} \end{bmatrix} \Theta\Theta^T \begin{bmatrix} \mathbf{Y}^T_{S,k|k-1} & \hat{\mathbf{y}}_{S,k|k-1} \\ \mathbf{Y}^T_{1,M,k|k-1} & \mathbf{Y}^T_{\mathbf{R},1,k}\mathbf{z}_{1,k} \\ \mathbf{Y}^T_{2,M,k|k-1} & \mathbf{Y}^T_{\mathbf{R},2,k}\mathbf{z}_{2,k} \\ \vdots & \vdots \\ \mathbf{Y}^T_{D,M,k|k-1} & \mathbf{Y}^T_{\mathbf{R},1,k}\mathbf{z}_{D,k} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{Y}_{S,k|k} & \mathbf{0} \\ \hat{\mathbf{y}}^T_{S,k|k} & \star \end{bmatrix} \begin{bmatrix} \mathbf{Y}^T_{S,k|k} & \hat{\mathbf{y}}_{S,k|k} \\ \mathbf{0} & \star \end{bmatrix} \tag{4.68}$$

By using $\Theta\Theta^T = \mathbf{I}$, (4.68) can be further expressed as

$$\begin{bmatrix} \sum_{j=1}^{D} \mathbf{Y}_{j,M,k|k-1}\mathbf{Y}^T_{j,M,k|k-1} + \mathbf{Y}_{S,k|k-1}\mathbf{Y}^T_{S,k|k-1} & \sum_{j=1}^{D} \mathbf{Y}_{j,M,k|k-1}\mathbf{Y}^T_{R,j,k}\mathbf{z}_{j,k} + \mathbf{Y}_{S,k|k-1}\hat{\mathbf{y}}_{S,k|k-1} \\ \sum_{j=1}^{D} \mathbf{z}^T_{j,k}\mathbf{Y}_{R,j,k}\mathbf{Y}^T_{j,M,k|k-1} + \hat{\mathbf{y}}^T_{S,k|k-1}\mathbf{Y}^T_{S,k|k-1} & \sum_{j=1}^{D} \mathbf{z}_{j,k}\mathbf{Y}_{R,j,k}\mathbf{Y}^T_{R,j,k}\mathbf{z}_{j,k} + \hat{\mathbf{y}}^T_{S,k|k-1}\mathbf{Y}^T_{S,k|k-1} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{Y}_{S,k|k}\mathbf{Y}^T_{S,k|k} & \mathbf{Y}_{S,k|k}\hat{\mathbf{y}}_{S,k|k} \\ \hat{\mathbf{y}}^T_{S,k|k}\mathbf{Y}^T_{S,k|k} & \star \end{bmatrix} \tag{4.69}$$

By equating the corresponding terms of LHS and RHS of (4.69) and by using

$$\mathbf{Y} = \mathbf{Y}_s \mathbf{Y}_s^T \tag{4.70}$$

$$\mathbf{y} = \mathbf{Y}_s \mathbf{y}_s \tag{4.71}$$

we get

$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \sum_{j=1}^{D} \mathbf{i}_{s,j,k} \tag{4.72}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{j=1}^{D} \mathbf{I}_{s,j,k} \tag{4.73}$$

where

$$\mathbf{I}_{s,j,k} = \mathbf{Y}_{j,M,k|k-1} \mathbf{Y}_{j,M,k|k-1}^T \tag{4.74}$$

$$\mathbf{i}_{s,j,k} = \mathbf{Y}_{j,M,k|k-1} \mathbf{Y}_{\mathbf{R},j,k}^T \mathbf{z}_{j,k}. \tag{4.75}$$

From (4.66) and (4.74), we get

$$\mathbf{I}_{s,j,k} = \mathbf{P}_{k|k-1}^{-T/2} \mathbf{P}_{j,zz,k|k-1}^{T/2} \mathbf{R}_j^{-T/2} \mathbf{R}_j^{-1/2} \mathbf{P}_{j,zz,k|k-1}^{1/2} \mathbf{P}_{j,k|k-1}^{-1/2}. \tag{4.76}$$

Using (4.63) in (4.76), we get

$$\mathbf{I}_{s,j,k} = \nabla \mathbf{h}_{j,x}^T \mathbf{R}_{j,k}^{-1} \nabla \mathbf{h}_{j,x}. \tag{4.77}$$

From (4.33) and (4.77), we get

$$\begin{aligned} \mathbf{I}_{s,j,k} &= \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{j,xz,k|k-1} \mathbf{R}_{j,k}^{-1} \mathbf{P}_{j,xz,k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \\ &= \mathbf{M}_{j,k|k-1}^T \mathbf{R}_{j,k}^{-1} \mathbf{M}_{j,k|k-1} \end{aligned} \tag{4.78}$$

which is the same as (4.42) and hence the decentralised SRCIF is equivalent to the decentralised CIF. Similarly, the corresponding analysis can be easily done for the information vector contribution, $\mathbf{i}_{s,j,k}$.

In a similar way, the square root UIF can be easily derived.

4.5 Speed and Rotor Position Estimation of a Two Phase Permanent Magnet Synchronous Motor

In this section, we will consider the state estimation of a two phase permanent magnet synchronous motor (PMSM) [11]. The PMSM has four states, the first two states are currents through the two windings, the third state is speed and the fourth state is rotor angular position. The inputs to the motor are the voltages, $u_{1,k}$ and $u_{2,k}$. The objective is to estimate the rotor angular position and speed of PMSM using the two winding currents.

The discrete-time nonlinear model of PMSM is [11]

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \\ x_{4,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} + T_s(-\frac{R}{L}x_{1,k} + \frac{\omega\lambda}{L}\sin x_{4,k} + \frac{1}{L}u_{1,k}) \\ x_{2,k} + T_s(-\frac{R}{L}x_{2,k} - \frac{\omega\lambda}{L}\cos x_{4,k} + \frac{1}{L}u_{2,k}) \\ x_{3,k} + T_s(-\frac{3\lambda}{2J}x_{1,k}\sin x_{4,k} + \frac{3\lambda}{2J}x_{2,k}\cos x_{4,k} - \frac{Fx_{3,k}}{J}) \\ x_{4,k} + T_s x_{3,k} \end{bmatrix}$$

the outputs and inputs are

$$\begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix} = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad \begin{bmatrix} u_{1,k} \\ u_{2,k} \end{bmatrix} = \begin{bmatrix} \sin(0.002\pi k) \\ \cos(0.002\pi k) \end{bmatrix}.$$

Note that, in this thesis one of the main aims is to show the efficacy of the proposed state estimation algorithms. The plant models are discretised using Euler's method. However, one can investigate other discretisation methods like Runge-Kutta methods, etc.

The following parameters are considered for the simulations: R = 1.9Ω, λ=0.1, L =

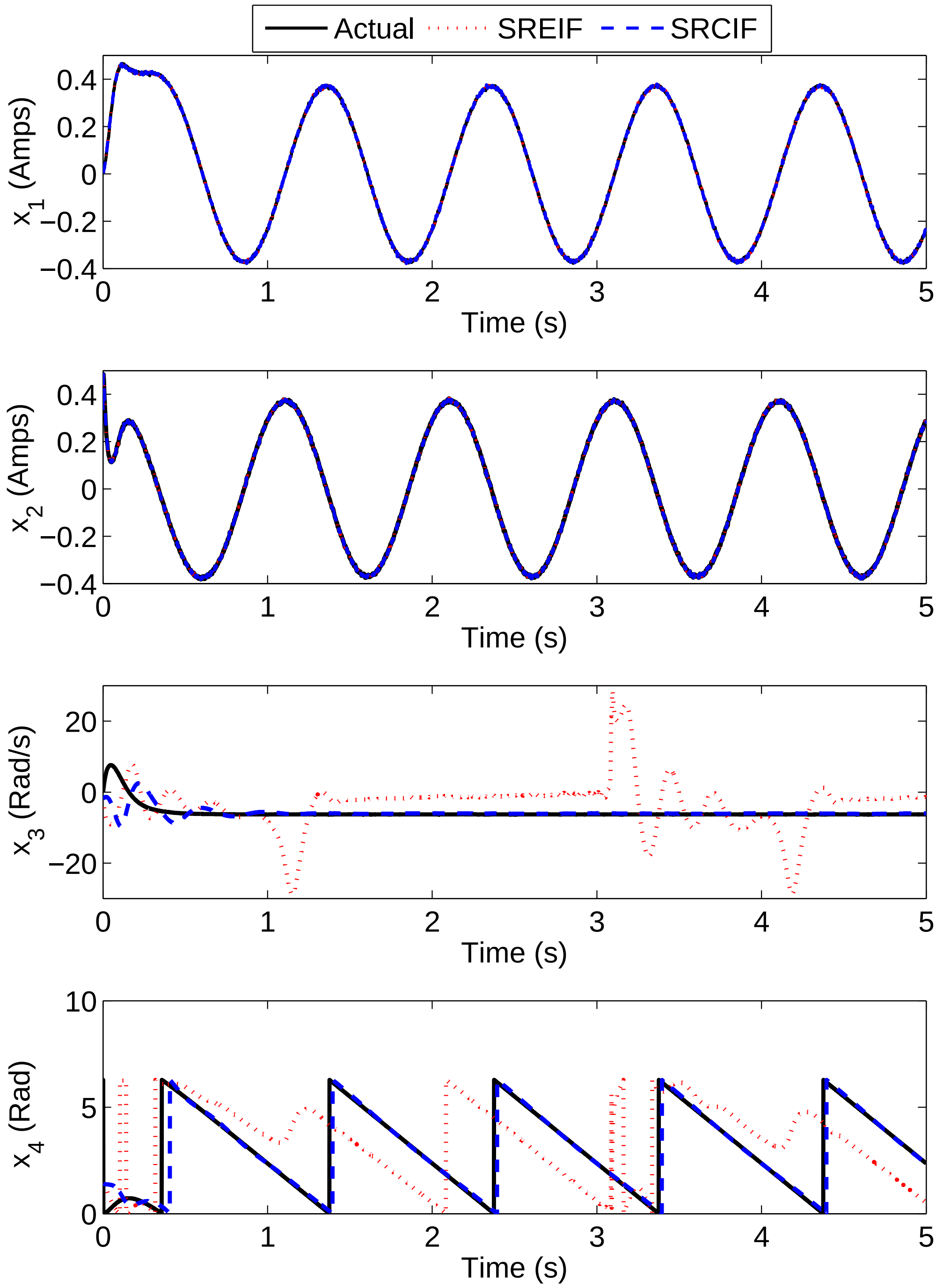

Figure 4.1: Actual and estimated states using SREIF and SRCIF.

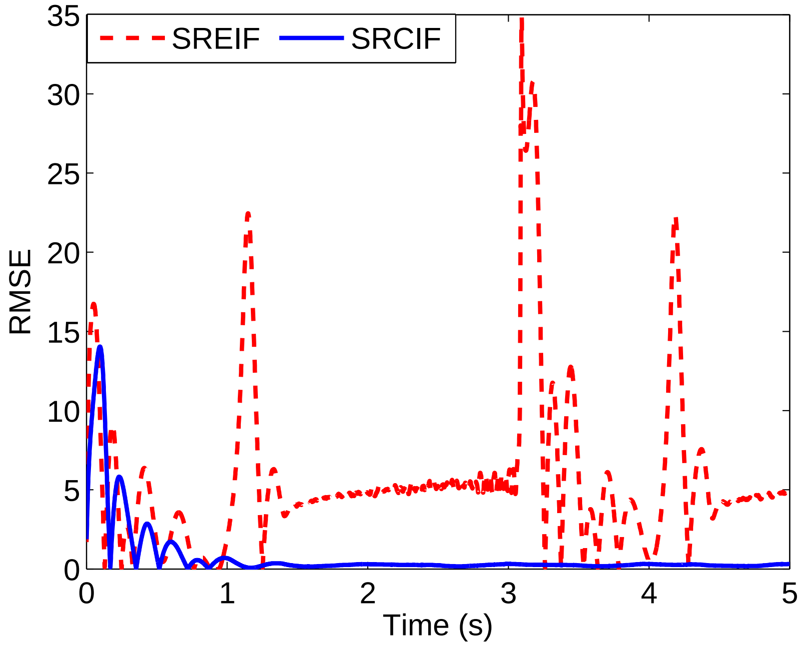

Figure 4.2: RMSE of PMSM using SREIF and SRCIF.

0.003H, J = 0.00018, F = 0.001 and $T_s = 0.001$ s. The covariance matrices for the process and measurement noises

$$\mathbf{Q} = \begin{bmatrix} 11.11 & 0 & 0 & 0 \\ 0 & 11.11 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 1\times 10^{-6} \end{bmatrix}, \quad \mathbf{R} = 1\times 10^{-6}\mathbf{I}_2$$

are added to the plant and measurement models. The initial conditions for all the plant states are 0, the initial information vector is selected from $\mathcal{N}\left(\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T, \mathbf{I}_4\right)$. The speed and the rotor angular position are estimated using SREIF, SRUIF and SRCIF. The SRUIF tuning parameters are $\alpha = 0.001$, $\beta = 2$ and $\kappa = 3 - n$ [84]. Over 500 Monte-Carlo runs were performed to analyse the performance of the estimates. Figure 4.1 shows a typical result of one of the Monte-Carlo simulations. In Figure 4.1 the estimated SRCIF states are very close to the actual states, whereas estimated states using SREIF fail to converge to the actual states. Note that the speed in Figure 4.1 is negative as the PMSM

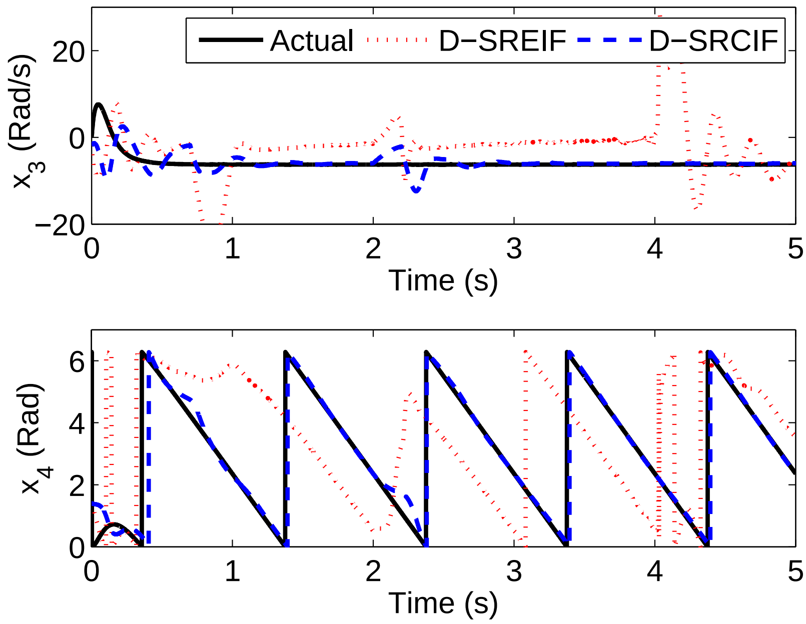

Figure 4.3: Actual and estimated states of PMSM using decentralised SREIF and decentralised SRCIF.

is simulated based on an open-loop control strategy. The desired speed tracking can be achieved by designing a closed-loop control system, which is beyond the scope of this work. The root mean square error (RMSE) plot for the PMSM's speed is shown in Figure 4.2. The peaks in Figure 4.2 are due to the sudden change in rotor angle which occurs at rotor angle of 2π, where SREIF estimation is poor. The average RMSE values over simulated time are 8.1874 and 2.2909 for SREIF and SRCIF, respectively. In some of the simulations, the SRUIF response is unstable and hence its performance is not shown in the plots. One of the reasons for this divergent behaviour of SRUIF is the unavailability of the positive definite square root matrices, which halts the filter. The similar instability of unscented filters is discussed in [45]. To show the effectiveness of the multi-sensor SRCIF, data from two different sets of sensors are used in the simulations. The noise covariances of the two sensors are

$$\mathbf{R}_1 = 1 \times 10^{-6} I_2 \quad \text{and} \quad \mathbf{R}_2 = 2.5 \times 10^{-5} I_2.$$

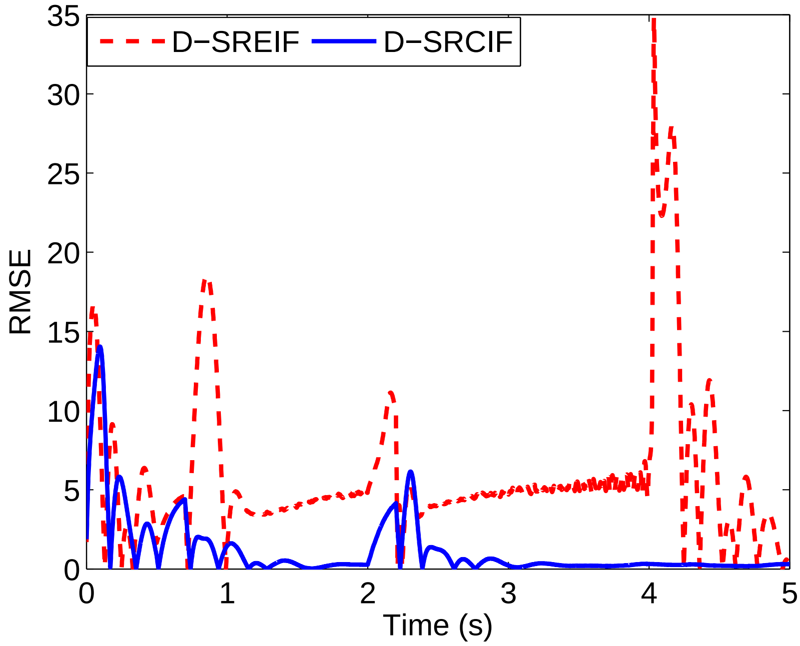

Figure 4.4: RMSE of PMSM using decentralised SREIF and decentralised SRCIF.

The simulation results for state estimation of the PMSM using multi-sensor SREIF and SRCIF are shown in Figure 4.3. Although fault detection and isolation are not the main scope of this work, state estimation in the presence of data loss is simulated in this case. It is assumed that the first and second sensor fails to give any output from 0.5 to 0.7 s and from 2 to 2.5 s, respectively. The first two actual states and its estimates using decentralised SREIF, SRCIF almost overlap and hence they are not shown in Figure 4.3. It can be seen that state estimates using multi-sensor SRCIF in presence of data loss are very close to each other. The sudden changes in the rotor position and angle at 0.5 and 2 s are due to the data loss. The average RMSE values over simulated time are 8.0975 and 2.5747 for multi-sensor SREIF and SRCIF, respectively. The RMSE plot for PMSM in the presence of data loss is shown in Figure 4.4 and it can be seen that, using two different sensors, the multi-sensor SRCIF outperforms multi-sensor SREIF.

4.5.1 Simulations of PMSM without limiting the rotor position from 0 to 2π radians

In this section, the simulations given in the previous section are repeated without limiting the rotor position from 0 to 2π radians. Over 500 Monte-Carlo runs were performed to analyse the performance of the estimates. Figure 4.5 shows a typical result of one of the Monte-Carlo simulations. In Figure 4.5 the estimated SRCIF states are very close to the actual states, whereas estimated states using SREIF fail to converge to the actual states. The RMSE plot for the PMSM's speed is shown in Figure 4.5. The average RMSE values over simulated time are 15.17 and 0.5582 for SREIF and SRCIF, respectively.

The simulation results for state estimation of the PMSM, without limiting the rotor position from 0 to 2π, using multi-sensor SREIF and SRCIF are shown in Figure 4.7. It is assumed that the first and second sensor fails to give any output from 0.5 to 0.7 s and from 2 to 2.5 s, respectively. The first two actual states and its estimates using decentralised SREIF, SRCIF almost overlap and hence they are not shown in Figure 4.7. It can be seen that state estimates using multi-sensor SRCIF in presence of data loss are very close to each other. The sudden changes in the rotor position and angle at 0.5 and 2 s are due to the data loss. The average RMSE values over simulated time are 14.3804 and 1.5487 for multi-sensor SREIF and SRCIF, respectively. The RMSE plot for PMSM in the presence of data loss is shown in Figure 4.8 and it can be seen that, using two different sensors, the multi-sensor SRCIF outperforms multi-sensor SREIF. Note that the RMSE plots shows the RMSE of the PMSM's speed only and hence the plots given in Figure 4.2 and 4.6, and 4.4 and 4.8 looks similar.

4.6 Conclusions

In this chapter, we have proposed a cubature information filter (CIF) and its square root version (SRCIF) for nonlinear systems. The proposed filters are derived from an extended

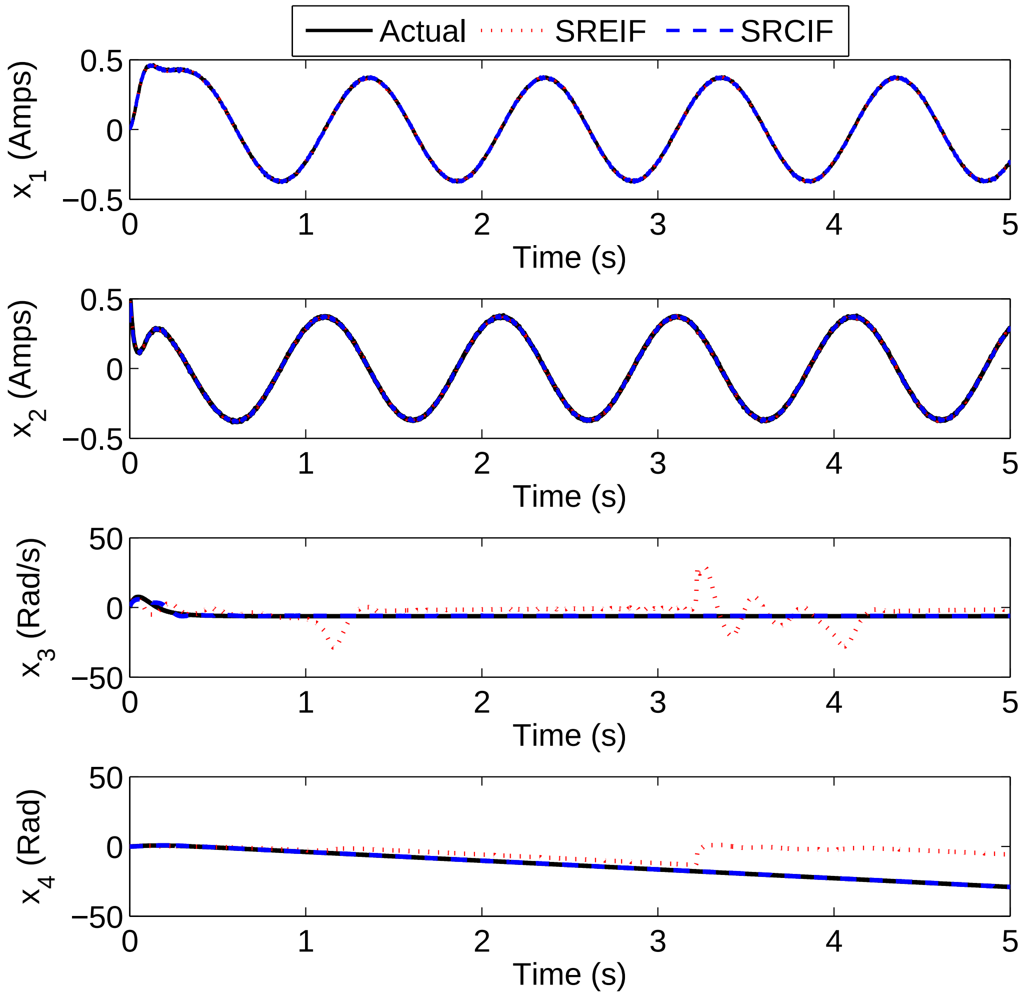

Figure 4.5: Actual and estimated states using SREIF and SRCIF; without limiting the rotor position to 2π radians.

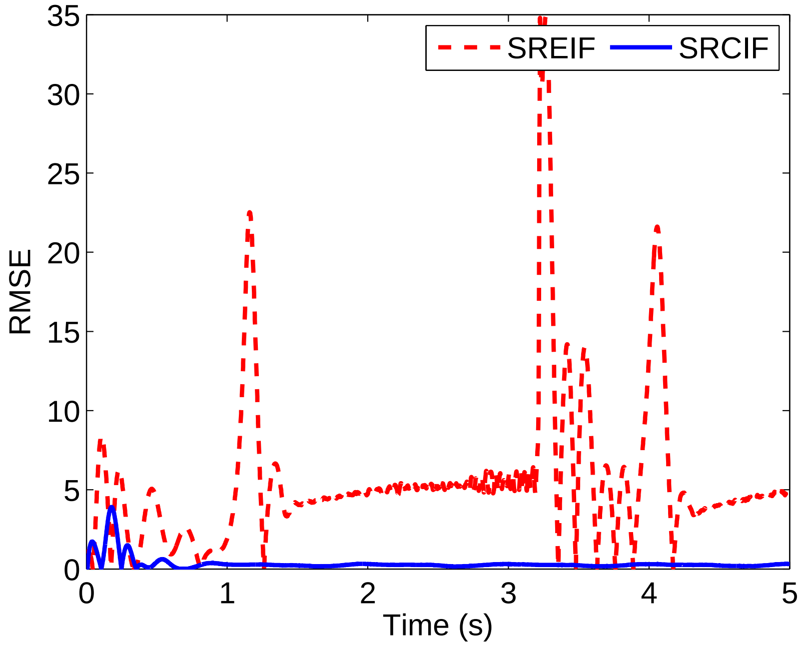

Figure 4.6: RMSE of PMSM using SREIF and SRCIF; without limiting the rotor position to 2π radians.

information filter, a cubature Kalman filter and their square root versions. The CIF and SRCIF have the following desirable properties

1. They do not require the evaluation of Jacobians during the prediction and measurement update stages.

2. The update step is computationally simpler.

3. The SRCIF is numerically stable and reliable.

4. They are easy to extend for multi-sensor state estimation.

The efficacy of the proposed algorithms are verified by simulations. The multi-sensor SRCIF is applied to state estimation of PMSM and is compared with SREIF and SRUIF. It is also shown that the SRCIF, when applied to multi-sensor state estimation outperforms SREIF and SRUIF. One of the advantages of information filters is to deal with state estimation with multiple sensors.

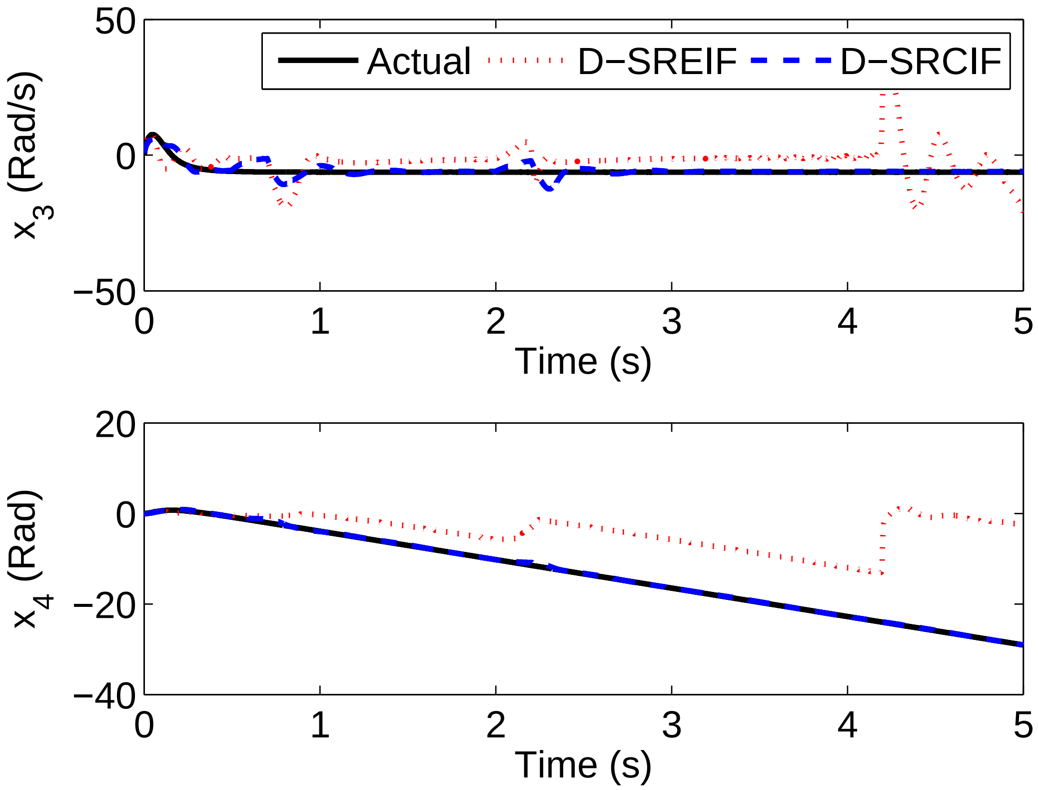

Figure 4.7: Actual and estimated states of PMSM using decentralised SREIF and decentralised SRCIF; without limiting the rotor position to 2π radians.

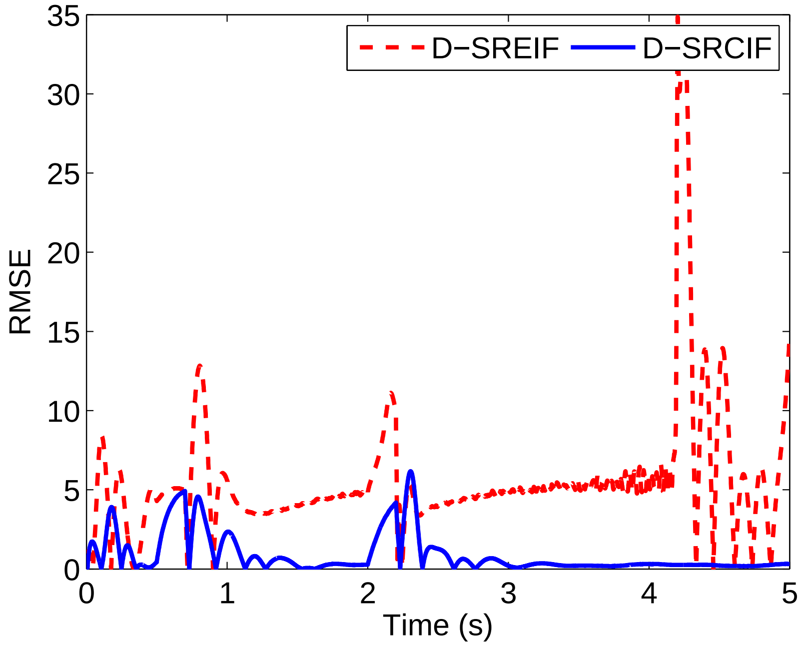

Figure 4.8: RMSE of PMSM using decentralised SREIF and decentralised SRCIF; without limiting the rotor position to 2π radians.

The proposed algorithms can be further explored in robotics and aerospace applications, where data from different sensors are fused to achieve reliable state estimates.

Chapter 5

Cubature H_∞ Filters

5.1 Introduction

In Chapter 3, EKF, UKF and CKF were discussed and in Chapter 4, CKF was further extended in information domain to deal with multi-sensor state estimation. However in the previously discussed nonlinear state estimators, the statistical properties of the noises are assumed to be known apriori and, in addition, they may not be robust against parametric uncertainties. This chapter deals with the derivative free state estimation for nonlinear systems with non-Gaussian noises using CKF and EH_∞F. In H_∞ filters, neither the accurate model nor the 'apriori' statistical noise properties are required [17], [48], [47], [49], [11]. The extended version of H_∞ filters [49] and extended H_∞ filter (EH_∞F), still require Jacobians during the state estimation of nonlinear systems, which may degrade the performance of highly nonlinear systems. The mixed H_2/H_∞ filter combines the best features of Kalman filtering and H_∞ filtering [86-88].

In this chapter, we present a cubature H_∞ filter by embedding CKF with EH_∞F for nonlinear, non-Gaussian systems, furthermore, a square root version is derived. The applicability of the square root cubature H_∞ filter is demonstrated on the nonlinear state estimation of a closed loop continuous stirred tank reactor (CSTR), in the presence of Gaussian and non-Gaussian noises. The control variable of CSTR is a function of the estimated states and is computed by a feedback linearisation method [93]. The rest of the Chapter is structured as follows. Sections 5.2 and 5.3 include the preliminaries of the extended H_∞ filter and CKF, respectively. Section 5.4 describes the cubature H_∞ filter and its square root version is derived in Section 5.5. Section 5.6 includes numerical simulations and concluding remarks are presented in Section 5.7.

5.2 An Extended H_∞ Filter

This section presents a brief introduction to an EH_∞F. An H_∞ filter for linear systems was discussed in Chapter 2, for a detailed formulation and derivation of EH_∞F see for example [48] and [49].

The discrete-time process and observation models can be written as

$$\mathbf{x}_k = \mathbf{f}[\mathbf{x}_{k-1}, \mathbf{u}_{k-1}] + \mathbf{w}_{k-1} \tag{5.1}$$

$$\mathbf{z}_k = \mathbf{h}[\mathbf{x}_k, \mathbf{u}_k] + \mathbf{v}_k \tag{5.2}$$

where k is a current time index, $\mathbf{x}_k \in \mathscr{R}^n$ is a state vector, $\mathbf{u}_k \in \mathscr{R}^q$ is a control input, $\mathbf{z}_k \in \mathscr{R}^p$ is a measurement vector, and $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are the process and observation noises. The noise terms $\mathbf{w}_k$ and $\mathbf{v}_k$ may be random with possibly unknown statistics, or they may be deterministic. They may have a non-zero mean. Instead of directly estimating the state one can estimate a linear combination of states.

In the game theory approach to H_∞ filtering [17], [49], the performance measure is given by

$$\mathbf{J}_\infty = \frac{\sum_{k=1}^{N} ||\mathbf{n}_k - \widehat{\mathbf{n}}_k||^2_{\mathbf{M}_k}}{||\mathbf{x}_0 - \hat{\mathbf{x}}_0||^2_{\mathbf{P}_0^{-1}} + \sum_{k=1}^{N}(||\mathbf{w}_k||^2_{\mathbf{Q}_k^{-1}} + ||\mathbf{v}_k||^2_{\mathbf{R}_k^{-1}})} \tag{5.3}$$

where $\mathbf{P}_0$, $\mathbf{Q}_k$, $\mathbf{R}_k$, and $\mathbf{M}_k$ are symmetric positive definite weighing matrices chosen by the user based on the problem at hand. The norm notation used in this section is $||e||^2_{S_k} = e^T S_k e$. This is the same performance measure which was discussed in Section for linear systems. A linear H_∞ filter can be easily extended to nonlinear systems by replacing the linear state and measurement matrices with their Jacobians, and by slightly modifying the predicted state equation. In this Chapter, we have used the EH_∞F algorithm given in [49] and is given below.

The predicted state vector and auxiliary matrix are

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\widehat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{5.4}$$

$$\mathbf{P}_{k|k-1} = \nabla\mathbf{f}_x \mathbf{P}_{k-1|k-1} \nabla\mathbf{f}_x^T + \mathbf{Q}_k \tag{5.5}$$

and the inverse of the updated auxiliary matrix can be obtained as

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \nabla\mathbf{h}_x^T \mathbf{R}_k^{-1} \nabla\mathbf{h}_x - \gamma^{-2}\mathbf{I}_n \tag{5.6}$$

where $\mathbf{I}_n$ denotes the identity matrix of dimension $n \times n$.
The updated state is

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_\infty[\mathbf{z}_k - \mathbf{h}(\widehat{\mathbf{x}}_{k|k-1})] \tag{5.7}$$

where

$$\mathbf{K}_\infty = \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T [\nabla\mathbf{h}_x \mathbf{P}_{k|k-1} \nabla\mathbf{h}_x^T + \mathbf{R}_k]^{-1} \tag{5.8}$$

The Jacobians of $\mathbf{f}$ and $\mathbf{h}$, $\nabla\mathbf{f}_x$ and $\nabla\mathbf{h}_x$, are evaluated at $\widehat{\mathbf{x}}_{k-1|k-1}$ and $\widehat{\mathbf{x}}_{k|k-1}$, respectively.

5.3 Cubature Kalman filter

For the completeness, some of the key prediction equations described in previous chapters are repeated in this section. These prediction equations are required for the CH_∞F's derivation.

The cubature points required for the prediction step are

$$\chi_{i,k-1|k-1} = \sqrt{\mathbf{P}_{k-1|k-1}}\xi_i + \hat{\mathbf{x}}_{k-1|k-1} \tag{5.9}$$

where $i = 1, 2, ..., 2n$ and n is the size of the state vector.

The propagated cubature points through the process model are

$$\chi^*_{i,k|k-1} = \mathbf{f}(\chi_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{5.10}$$

The evaluated mean and error covariance matrix are

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \chi^*_{i,k|k-1} \tag{5.11}$$

$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \chi^*_{i,k|k-1}\chi^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + \mathbf{Q}_{k-1} \tag{5.12}$$

The predicted measurement and its associated covariances are

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{z}_{i,k|k-1} \tag{5.13}$$

$$\mathbf{P}_{zz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{z}_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} + \mathbf{R}_k \tag{5.14}$$

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \chi_{i,k|k-1}\mathbf{z}^T_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} \tag{5.15}$$

where

$$\mathbf{z}_{i,k|k-1} = \mathbf{h}(\chi_{i,k|k-1}, \mathbf{u}_k) \tag{5.16}$$

$$\chi_{i,k|k-1} = \sqrt{\mathbf{P}_{k|k-1}}\xi_i + \hat{\mathbf{x}}_{k|k-1}. \tag{5.17}$$

5.4 Cubature H_∞ filter

This section describes the cubature H_∞ filter algorithm, which uses an EH_∞F in the CKF framework. It can be seen from the Section 5.2, that EH_∞F requires Jacobians, and the CKF detailed in Section 5.3 assumes that the statistical noise properties are known during the state estimation. In this section, we present a nonlinear estimation algorithm in which neither the Jacobians nor the statistical noise properties are required for state estimation. It can be useful in the presence of parametric uncertainties. If some of the statistical properties of noises are known apriori, then it can be incorporated in the proposed method. The proposed cubature H_∞ filter is an heuristic approach, which has the advantages of both CKF and an EH_∞F. The main aim is to develop a filter which should be a derivative free filter like CKF and have the robustness properties of an EH_∞F. The idea is to use the prediction step of CKF and the update step of H_∞ filter. It is not straight forward to just replace the prediction stage from the CKF and update stage from EH_∞F to form the cubature H_∞ filter. The main difficulty is due to the update stage of EH_∞F, where one of the key steps is the evaluation of Jacobians. For the derivative free cubature H_∞ filter, these Jacobians are approximated using the linear propagation property. In the cubature H_∞ filter, the prediction step is similar to CKF, which involves the factorisation of the error auxiliary matrix, evaluation of cubature and propagated cubature points for the process model and estimation of the predicted state and predicted error auxiliary matrix, see [45] for more details. The aim of this section is to fuse an EH_∞F and CKF to obtain a filter which will have the desirable properties of both filters. The update step of the EH_∞F requires linearised Jacobian of the measured model, while it does not explicitly exist in the CKF framework. However, by using the following linear propagation property [82] and [36], it is possible to embed the EH_∞F in the CKF framework. The linear error propagation property for the error covariance and cross covariance can be approximated as [82]

$$\mathbf{P}_{xz,k|k-1} \simeq \mathbf{P}_{k|k-1}\nabla\mathbf{h}_x^T \tag{5.18}$$

$$\mathbf{P}_{zz,k|k-1} \simeq \nabla\mathbf{h}_x\mathbf{P}_{k|k-1}\nabla\mathbf{h}_x^T \tag{5.19}$$

Now we will use (5.18) to determine the update step of the cubature H_∞ filter. By multiplying $\mathbf{P}_{k|k-1}^{-1}$ and $\mathbf{P}_{k|k-1}$, and their transposes on the second term of RHS of (5.6) we get

$$\nabla \mathbf{h}_x^T \mathbf{R}_k^{-1} \nabla \mathbf{h}_x = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{k|k-1} \nabla \mathbf{h}_x^T \mathbf{R}_k^{-1} \nabla \mathbf{h}_x \mathbf{P}_{k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \tag{5.20}$$

and by using (5.18) in (5.20) we get

$$\nabla \mathbf{h}_x^T \mathbf{R}_k^{-1} \nabla \mathbf{h}_x = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{xz,k|k-1} \mathbf{R}_k^{-1} \mathbf{P}_{xz,k|k-1}^T \mathbf{P}_{k|k-1}^{-T} \tag{5.21}$$

By using the inverse of the updated auxiliary matrix and EH_∞F gain , (5.6) and (5.8), (5.18) and (5.19) we get

$$\mathbf{K}_{c\infty} = \mathbf{P}_{xz,k|k-1} (\mathbf{P}_{zz,k|k-1} + \mathbf{R}_k)^{-1} \tag{5.22}$$

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{xz,k|k-1} \mathbf{R}_k^{-1} \mathbf{P}_{xz,k|k-1}^T \mathbf{P}_{k|k-1}^{-T} - \gamma^{-2} \mathbf{I}_n \tag{5.23}$$

The recovery of a covariance matrix from an information matrix is an active area of research [77-80]. The auxiliary matrix $\mathbf{P}_{k|k}$ can be recovered from (5.23) by using MATLAB's *leftdivide* operator [79]

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{-1} \backslash \mathbf{I}_n. \tag{5.24}$$

The predicted step of the cubature H_∞ filter is similar to the prediction step of CKF [45]. The cubature H_∞ filter algorithm is summarised in Algorithm 12.

Algorithm 12 Cubature H_∞ Filter

Initialise the state vector, $\widehat{\mathbf{x}}$, and the auxiliary matrix, $\mathbf{P}$ (set $k = 1$).

Prediction

1: Factorise

$$\mathbf{P}_{k-1|k-1} = \mathbf{P}_{k-1|k-1}^{1/2} \mathbf{P}_{k-1,k-1}^{T/2}. \tag{5.25}$$

2: Evaluate the cubature points, $\boldsymbol{X}_{i,k-1|k-1}$

$$\boldsymbol{X}_{i,k-1|k-1} = \mathbf{P}_{k-1|k-1}^{1/2} \xi_i + \widehat{\mathbf{x}}_{k-1|k-1} \tag{5.26}$$

3: Evaluate the propagated cubature points, $\boldsymbol{X}^*_{i,k|k-1}$

$$\boldsymbol{X}^*_{i,k|k-1} = \mathbf{f}(\boldsymbol{X}_{i,k-1|k-1}, \mathbf{u}_{k-1}) \tag{5.27}$$

4: Estimate the predicted state, $\widehat{\mathbf{x}}_{k|k-1}$

$$\widehat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{X}^*_{i,k-1|k-1} \tag{5.28}$$

5: Estimate the predicted auxiliary matrix, $\mathbf{P}_{k|k-1}$

$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{X}^*_{i,k-1|k-1}\boldsymbol{X}^{*T}_{i,k-1|k-1} - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{x}}^T_{k|k-1} + Q_{k-1} \tag{5.29}$$

Measurement Update

1: Factorise

$$\mathbf{P}_{k|k-1} = \mathbf{P}^{1/2}_{k|k-1}\mathbf{P}^{T/2}_{k|k-1}. \tag{5.30}$$

2: Evaluate the cubature points

$$\boldsymbol{X}_{i,k|k-1} = \mathbf{P}^{1/2}_{k|k-1}\xi_i + \widehat{\mathbf{x}}_{k|k-1}. \tag{5.31}$$

3: Evaluate the propagated cubature points of measurement model, $\boldsymbol{Z}_{i,k|k-1}$

$$\boldsymbol{Z}_{i,k|k-1} = \mathbf{h}(\boldsymbol{X}_{i,k|k-1}, \mathbf{u}_k) \tag{5.32}$$

4: Estimate the predicted measurement, $\widehat{\mathbf{z}}_{k|k-1}$

$$\widehat{\mathbf{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{Z}_{i,k|k-1} \tag{5.33}$$

5: Estimate the measurement auxiliary matrix, $\mathbf{P}_{zz,k|k-1}$

$$\mathbf{P}_{zz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{m}\mathbf{Z}_{i,k|k-1}\mathbf{Z}_{i,k|k-1}^T - \widehat{\mathbf{z}}_{k|k-1}\widehat{\mathbf{z}}_{k|k-1}^T + \mathbf{R}_k \tag{5.34}$$

6: Estimate the cross auxiliary matrix, $\mathbf{P}_{xz,k|k-1}$

$$\mathbf{P}_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{m}\boldsymbol{X}_{i,k|k-1}\mathbf{Z}_{i,k|k-1}^T - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{z}}_{k|k-1} \tag{5.35}$$

7: Estimate the gain matrix, $\mathbf{K}_{c\infty}$

$$\mathbf{K}_{c\infty} = \mathbf{P}_{xz,k|k-1}(\mathbf{P}_{zz,k|k-1} + \mathbf{R}_k)^{-1}. \tag{5.36}$$

8: Estimate the updated state

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{c\infty}(\mathbf{z}_k - \widehat{\mathbf{z}}_{k|k-1}^T). \tag{5.37}$$

9: Estimate the inverse of the updated auxiliary matrix, $\mathbf{P}_{k|k}^{-1}$

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{P}_{k|k-1}^{-1}\mathbf{P}_{xz,k|k-1}\mathbf{R}_k^{-1}\mathbf{P}_{xz,k|k-1}^T\mathbf{P}_{k|k-1}^{-T} - \gamma^{-2}\mathbf{I}_n. \tag{5.38}$$

10: Recover the auxiliary matrix, $\mathbf{P}_{k|k}$, using (5.24).

If the statistical properties of noises are known, then in the cubature H_∞ filter they can be directly used as noise covariance matrices. If the statistical properties of noises are not known, then the desired performance can be achieved by using '**Q**' and '**R**' as tuning parameters. The selection of an attenuation parameter, γ, is also crucial for the existence of the filter. One can either use an appropriate fixed γ or a time-varying γ described in [47].

5.5 Square root Cubature H_∞ Filter

During the real-time implementation of state estimation algorithms, the propagated error covariance matrices may become ill conditioned, which eventually halts the filter operation. This can happen if some of the states are measured with greater precision than other states, where the corresponding elements of covariance matrix with accurately measured states will have lower values, while the other entries will have higher values [11]. These type of ill-conditioned covariance matrix may cause numerical instability during the online implementation. To circumvent these difficulties, one can use square root filters, where the square root of the error covariance matrices are propagated. Some of the key properties of square root filters are symmetric positive definite error covariances, availability of square root factors, doubled order precision, improved numerical accuracy, etc. [45], [11], [6], [26], [8].

Along the lines of square root Kalman filters, a few researchers have explored square root H_∞ filters. In [30], square root algorithms for H_∞ apriori, aposteriori and filtering problems are developed in Krein space. The square root H_∞ information estimation for a rectangular discrete-time descriptor system is described in [89], where the inverse of the covariance matrices (information matrices) are propagated. In [90], square root H_∞ estimators for time-variant descriptor systems are developed. One of the main differences in deriving the square root Kalman and H_∞ filters is the use of a rotation matrix. The square root Kalman filter uses the unitary matrix[1], whereas the square root H_∞ filter use the $\mathbf{J}$-unitary matrix.

In this section, we will first derive the update step of the square root H_∞ filter and then the square root cubature H_∞ filter will be discussed. The following notations for the matrices are used.

[1]When $\mathbf{J}$ is the identity matrix, unitary matrix is a special case of $\mathbf{J}$-unitary matrix.

Given a symmetric matrix $\mathbf{F}$, then $\mathbf{F}$ and $\mathbf{F}^{-1}$ can be factorised as

$$\mathbf{F} = (\mathbf{F}^{1/2})S(\mathbf{F}^{T/2}) \tag{5.39}$$

$$\mathbf{F}^{-1} = (\mathbf{F}^{-T/2})S(\mathbf{F}^{-1/2}) \tag{5.40}$$

where S is the signature matrix.

5.5.1 Square root H_∞ filter

The prediction step of the square root H_∞ filter is omitted here as it is not required for the derivation of the square root cubature H_∞ filter. The measurement update step of the square root H_∞ filter is

$$\begin{bmatrix} -\nabla\mathbf{h}_x^T\mathbf{R}_k^{-T/2} & \mathbf{P}_{k|k-1}^{-T/2} & \gamma^{-1}\mathbf{I}_n \\ \mathbf{R}_k^{1/2} & \nabla\mathbf{h}_x\mathbf{P}_{k|k-1}^{1/2} & \mathbf{0} \end{bmatrix}\Theta_J = \begin{bmatrix} \mathbf{0} & \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \mathbf{R}_e^{1/2} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.41}$$

where Θ_J is a $\mathbf{J}$-unitary matrix.

Once the square root factor of the auxiliary matrices $\mathbf{P}_{k|k}$ and $\mathbf{R}_e = \nabla\mathbf{h}_x\mathbf{P}_{k|k-1}\nabla\mathbf{h}_x^T + \mathbf{R}_k$ are obtained, the gain matrix of square root H_∞ filter can be obtained as

$$\mathbf{K}_{S\infty,k} = \mathbf{P}_{k|k-1}^{1/2}\mathbf{P}_{k|k-1}^{T/2}\nabla\mathbf{h}_x^T\mathbf{R}_e^{-T/2}\mathbf{R}_e^{-1/2} \tag{5.42}$$

The update state of square root H_∞ filter is

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{S\infty,k}(\mathbf{z}_k - \mathbf{h}(\widehat{\mathbf{x}}_{k|k-1})) \tag{5.43}$$

The measurement update in (5.41) can be easily derived by squaring both sides[2],

[2] If $b = a\Theta$, with Θ as J-unitary matrix, then [30]

$$bJb^T = a\Theta J\Theta^T a^T = aJa^T$$

$$\begin{bmatrix} -\nabla\mathbf{h}_x^T\mathbf{R}_k^{-T/2} & \mathbf{P}_{k|k-1}^{-T/2} & \gamma^{-1}\mathbf{I}_n \\ \mathbf{R}_k^{1/2} & \nabla\mathbf{h}_x\mathbf{P}_{k|k-1}^{1/2} & \mathbf{0} \end{bmatrix} \Theta_J\mathbf{J}\Theta_J^T \begin{bmatrix} -\mathbf{R}_k^{-1/2}\nabla\mathbf{h}_x & \mathbf{R}_k^{T/2} \\ \mathbf{P}_{k|k-1}^{-1/2} & \mathbf{P}_{k|k-1}^{T/2}\nabla\mathbf{h}_x^T \\ \gamma^{-1}\mathbf{I}_n & \mathbf{0} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{0} & \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \mathbf{R}_e^{1/2} & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{J} \begin{bmatrix} \mathbf{0} & \mathbf{R}_e^{T/2} \\ \mathbf{P}_{k|k}^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (5.44)$$

The $\mathbf{J}$-unitary matrix for (5.44) can be chosen as

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_n \end{bmatrix} \quad (5.45)$$

where $\mathbf{I}_p$ and $\mathbf{I}_n$ denotes the identity matrices of dimension $p \times p$ and $n \times n$, respectively. By using $\Theta_J\mathbf{J}\Theta_J^T = \mathbf{J}$, (5.44) can further be written as

$$\begin{bmatrix} \mathbf{P}_{k|k-1}^{-1} + \nabla\mathbf{h}_x^T\mathbf{R}_k^{-1}\nabla\mathbf{h}_x - \gamma^{-2}\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \nabla\mathbf{h}_x\mathbf{P}_{k|k-1}\nabla\mathbf{h}_x^T + \mathbf{R}_k \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{k|k}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_e \end{bmatrix} \quad (5.46)$$

One can see that the first entries of (5.46), and (5.42) are the same as the inverse of the updated auxiliary matrix and the gain of the H_∞ filter described in Section 5.2 and hence the square root extended H_∞ filter derived in this section is equivalent to the extended H_∞ filter in Section 5.2.

5.5.2 Square root Cubature H_∞ Filter

In this subsection, the derivative-free square root cubature H_∞ filter for nonlinear systems is derived. The square root CKF is a nonlinear filter which deals with only Gaussian noises [45] and the square root H_∞ filter detailed in Section 5.1 can deal with non-Gaussian noises but it requires Jacobians during the nonlinear state estimation. The proposed algorithm is derived from the square root CKF and square root H_∞ filter, and is called as square root cubature H_∞ filter. The main advantages of the square root cubature H_∞ filter are derivative-less nonlinear estimation and its capability to deal with the non-Gaussian noises. The prediction step of the square root cubature H_∞ filter is the same as for the square root CKF [45].

5.5.2.1 Square root Cubature H_∞ Filter Measurement Update

From (5.41), one can see that the measurement update stage of square root H_∞ filter requires the linearised measurement model, $\nabla\mathbf{h}$. By using the below statistical error propagation property [82], the derivative-free measurement update step for the square root cubature H_∞ filter can be derived as

$$\begin{aligned}\mathbf{P}_{zz} &\simeq \mathbb{E}[(\mathbf{z}-\hat{\mathbf{z}})(\mathbf{z}-\hat{\mathbf{z}})^T] \\ &\simeq \mathbb{E}[\nabla\mathbf{h}_x\mathbf{x}-\nabla\mathbf{h}_x\widehat{\mathbf{x}})(\nabla\mathbf{h}_x\mathbf{x}-\nabla\mathbf{h}_x\hat{\mathbf{x}})^T] \\ &\simeq \nabla\mathbf{h}_x cov(\mathbf{x})\nabla\mathbf{h}_x^T \\ &\simeq \nabla\mathbf{h}_x\mathbf{P}\nabla\mathbf{h}_x^T \end{aligned} \tag{5.47}$$

Let $\mathbf{P}_{zz}^{1/2}$ and $\mathbf{P}^{1/2}$ be the square root factors of $\mathbf{P}_{zz}$ and $\mathbf{P}$. Then (5.47) can be expressed as

$$\mathbf{P}_{zz}^{1/2}\mathbf{P}_{zz}^{T/2} = \nabla\mathbf{h}_x\mathbf{P}^{1/2}\mathbf{P}^{T/2}\nabla\mathbf{h}_x^T \tag{5.48}$$

and further (5.48) can be written as

$$\mathbf{P}_{zz}^{1/2} = \nabla\mathbf{h}_x\mathbf{P}^{1/2} \tag{5.49}$$

$$\mathbf{P}_{zz}^{T/2} = \mathbf{P}^{T/2}\nabla\mathbf{h}_x^T \tag{5.50}$$

By pre-multiplying $\mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{k|k-1}^{T/2}$ to $\nabla\mathbf{h}_x^T\mathbf{R}_k^{-T/2}$ and by using (5.50) we get

$$\begin{aligned}\nabla\mathbf{h}_x^T\mathbf{R}_k^{-T/2} &= \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{k|k-1}^{T/2}\nabla\mathbf{h}_x^T\mathbf{R}_k^{-T/2}\\ &= \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{R}_k^{-T/2}\end{aligned} \tag{5.51}$$

By using (5.49) and (5.51) in (5.41), the measurement update of the square root cubature H_∞ filter can be written as

$$\begin{bmatrix} -\mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{R}_k^{-T/2} & \mathbf{P}_{k|k-1}^{-T/2} & \gamma^{-1}\mathbf{I}_n \\ \mathbf{R}_k^{1/2} & \mathbf{P}_{zz,k|k-1}^{1/2} & \mathbf{0}\end{bmatrix}\Theta_J = \begin{bmatrix} \mathbf{0} & \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \mathbf{R}_e^{1/2} & \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{5.52}$$

The derivation of (5.52) is given in Section 5.5.3 and the computation of square root cubature H_∞ algorithm is summarised in Algorithm 13.

5.5.3 Derivation of Update Step in Square root Cubature H_∞ Filter

By expanding LHS of (5.52), we get

$$\begin{aligned}&\begin{bmatrix} -\mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{R}^{-T/2} & \mathbf{P}_{k|k-1}^{-T/2} & \gamma^{-1}\mathbf{I}_n \\ \mathbf{R}^{1/2} & \mathbf{P}_{zz,k|k-1}^{1/2} & \mathbf{0}\end{bmatrix}\Theta_J\mathbf{J}\Theta_J^T\begin{bmatrix} -\mathbf{R}^{-1/2}\mathbf{P}_{zz,k|k-1}^{1/2}\mathbf{P}_{k|k-1}^{-1/2} & \mathbf{R}^{T/2} \\ \mathbf{P}_{k|k-1}^{-1/2} & \mathbf{P}_{zz,k|k-1}^{T/2} \\ \gamma^{-1}\mathbf{I}_n & 0\end{bmatrix}\\ &= \begin{bmatrix} \mathbf{0} & \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \mathbf{R}_e^{1/2} & \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{J}\begin{bmatrix} \mathbf{0} & \mathbf{R}_e^{T/2} \\ \mathbf{P}_{k|k}^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\end{aligned} \tag{5.53}$$

By using $\Theta_J\mathbf{J}\Theta_J^T = \mathbf{J}$ and (5.45); (5.53) can be written as

$$\begin{bmatrix} \mathbf{P}_{k|k-1}^{-1} + \mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{R}_k^{-1}\mathbf{P}_{zz,k|k-1}^{1/2}\mathbf{P}_{k|k-1}^{-1/2} - \gamma^{-2}\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{zz,k|k-1} + \mathbf{R}_k\end{bmatrix} = \begin{bmatrix} \mathbf{P}_{k|k}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_e\end{bmatrix} \tag{5.54}$$

By using the statistical approximations (5.18) and (5.50) we get

$$\begin{aligned}
\mathbf{P}_{xz,k|k-1} &\simeq \mathbf{P}_{k|k-1}\nabla\mathbf{h}_x^T \\
&= \mathbf{P}_{k|k-1}^{1/2}\mathbf{P}_{k|k-1}^{T/2}\nabla\mathbf{h}_x^T \\
&= \mathbf{P}_{k|k-1}^{1/2}(\nabla\mathbf{h}_x\mathbf{P}_{k|k-1}^{1/2})^T \\
&= \mathbf{P}_{k|k-1}^{1/2}\mathbf{P}_{zz,k|k-1}^{T/2} \qquad (5.55)
\end{aligned}$$

By equating the corresponding terms of LHS and RHS of (5.54) and by using (5.55) in (5.54) we get

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{P}_{k|k-1}^{-1}\mathbf{P}_{xz,k|k-1}\mathbf{R}_k^{-1}\mathbf{P}_{xz,k|k-1}^T\mathbf{P}_{k|k-1}^{-T} - \gamma^{-2}\mathbf{I}_n \qquad (5.56)$$

and

$$\mathbf{R}_e = \mathbf{P}_{zz,k|k-1} + \mathbf{R}_k \qquad (5.57)$$

which is same as (5.23) in the cubature H_∞ filter.

Algorithm 13 Square root Cubature H_∞ Filter

Initialise the state vector, $\mathbf{x}$, and the square root of the auxiliary matrix, $\mathbf{P}^{1/2}$ (set $k=1$). **Prediction**

1: Evaluate the cubature points

$$X_{i,k-1|k-1} = \mathbf{P}_{k-1|k-1}^{1/2}\xi_i + \mathbf{x}_{k-1|k-1} \qquad (5.58)$$

2: Evaluate the propagated cubature points

$$X_{i,k|k-1}^* = \mathbf{f}(X_{i,k-1|k-1}, \mathbf{u}_{k-1}) \qquad (5.59)$$

3: Estimate the predicted state

$$\widehat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{X}^*_{i,k|k-1} \tag{5.60}$$

4: Estimate the square root factor of the predicted auxiliary matrix using the **QR**[3] decomposition

$$\mathbf{P}^{1/2}_{k|k-1} = \left[\mathbf{qr}\left(\mathscr{X}_{i,k|k-1} \qquad \mathbf{Q}^{1/2}\right)^T\right]^T \tag{5.61}$$

where

$$\mathscr{X}_{i,k|k-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{X}_{1,k|k-1} - \widehat{\mathbf{x}}_{k|k-1} \quad \boldsymbol{X}_{2,k|k-1} - \widehat{\mathbf{x}}_{k|k-1} \ldots \boldsymbol{X}_{2n,k|k-1} - \widehat{\mathbf{x}}_{k|k-1}\right] \tag{5.62}$$

Measurement Update

1: Evaluate the cubature points

$$\boldsymbol{X}_{i,k|k-1} = \mathbf{P}^{1/2}_{k|k-1}\xi_i + \widehat{\mathbf{x}}_{k|k-1}. \tag{5.63}$$

2: Evaluate the propagated cubature points

$$\boldsymbol{Z}_{i,k|k-1} = \mathbf{h}(\boldsymbol{X}_{i,k|k-1}, \mathbf{u}_k) \tag{5.64}$$

3: Evaluate the square root factor of measurement auxiliary matrix

$$\mathbf{P}^{1/2}_{zz,k|k-1} = \left[\mathbf{qr}\left(\mathscr{Z}_{i,k|k-1} \qquad \mathbf{R}^{1/2}_k\right)^T\right]^T \tag{5.65}$$

where

$$\mathscr{Z}_{i,k|k-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{Z}_{1,k|k-1} - \widehat{\mathbf{z}}_{k|k-1} \quad \boldsymbol{Z}_{2,k|k-1} - \widehat{\mathbf{z}}_{k|k-1} \quad \boldsymbol{X}_{2n,k|k-1} - \widehat{\mathbf{z}}_{k|k-1}\right] \tag{5.66}$$

[3]**QR** is orthogonal triangular decomposition and can be evaluated by using MATLAB command ‘qr’.

4: Evaluate the square root of updated auxiliary matrix using

$$\begin{bmatrix} -\mathbf{P}_{k|k-1}^{-T/2}\mathbf{P}_{zz,k|k-1}^{T/2}\mathbf{R}_k^{-T/2} & \mathbf{P}_{k|k-1}^{-T/2} & \gamma^{-1}\mathbf{I}_n \\ \mathbf{R}_k^{1/2} & \mathbf{P}_{zz,k|k-1}^{1/2} & \mathbf{0} \end{bmatrix} \Theta_J = \begin{bmatrix} \mathbf{0} & \mathbf{P}_{k|k}^{-T/2} & \mathbf{0} \\ \mathbf{R}_e^{1/2} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.67}$$

5: Evaluate the gain matrix using

$$\mathbf{K}_{SC\infty,k} = \mathbf{P}_{xz,k|k-1}\mathbf{R}_e^{-T/2}\mathbf{R}_e^{-1/2} \tag{5.68}$$

where

$$\mathbf{P}_{xz,k|k-1} = \mathbf{P}_{k|k-1}^{1/2}\mathbf{P}_{zz,k|k-1}^{T/2}$$

6: Evaluate the updated state using

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{SC\infty,k}(\mathbf{z}_k - \widehat{\mathbf{z}}_{k|k-1}) \tag{5.69}$$

7: Recover the square root factor of the updated auxiliary matrix, $\mathbf{P}_{k|k}^{1/2}$, using

$$\mathbf{P}_{k|k}^{1/2} = \mathbf{P}_{k|k}^{-1/2}\backslash\mathbf{I}_n. \tag{5.70}$$

5.6 State Estimation of a CSTR

In order to evaluate the performance of the square root cubature H_∞ filter, the state estimation of a continuous stirred tank reactor (CSTR) in the presence of Gaussian and non-Gaussian noises is considered. The process model for an irreversible, first-order chemical reaction, $A \rightarrow B$ which occurs in a CSTR is [91-92]

$$\dot{C}_A = \frac{q}{V}(C_{Af} - C_A) - k_0 \exp\left(\frac{-E}{RT}\right) C_A \tag{5.71}$$

$$\dot{T} = \frac{q}{V}(T_f - T) + \frac{-\nabla H}{\rho C_p} k_0 \exp\left(\frac{-E}{RT}\right) C_A + \frac{UA}{V\rho C_p}(T_c - T) \tag{5.72}$$

The state vector (C_A and T) consists of concentration and temperature of the reactor and the measured output, h is temperature of the CSTR, T.

The following parameters are considered for the simulations [92]:
$q = 100L/min$, $\frac{E}{R} = 8750K$, $C_{Af} = 1mol/L$, $K_0 = 7.5 \times 10^{10} min^{-1}$, $T_f = 350K$, $UA = 5000J/minK$, $V = 100L$, $T_c = 300K$, $\rho = 1000g/L$, $C_A = 0.5mol/L$, $C_p = 0.239J/gK$, $T = 350K$ and $(\nabla H) = 5000J/mol$.

The process and measurement noises, w_k and v_k, are added to the process and measurement models. The control variable, coolant temperature T_c, is computed using input-output feedback linearisation [92]

$$T_c = \frac{\zeta - L_f h(x)}{L_g h(x)} \tag{5.73}$$

The Lie derivatives L_f and L_h, and ζ are

$$\begin{aligned} L_f h(x) &= \frac{q}{V}(T_f - T) + \frac{-\nabla H}{\rho C_p} k_0 \exp\left(\frac{-E}{RT}\right) C_A - \frac{UA}{V\rho C_p}(T) && (5.74) \\ L_g h(x) &= \frac{UA}{V\rho C_p} && (5.75) \\ \zeta &= 50z + 10(T_{setpoint} - T) && (5.76) \end{aligned}$$

where z can be obtained by integrating

$$\dot{z} = T_{setpoint} - T.$$

The control input is evaluated at the estimated states. The plant is discretised using Euler's method with a sampling time of $0.01s$ and the reactor set-point is 400 K. The objective is to estimate the full state vector of closed-loop CSTR using noisy temperature measurements. The simulations with perfect measurement noise are also performed. The selected

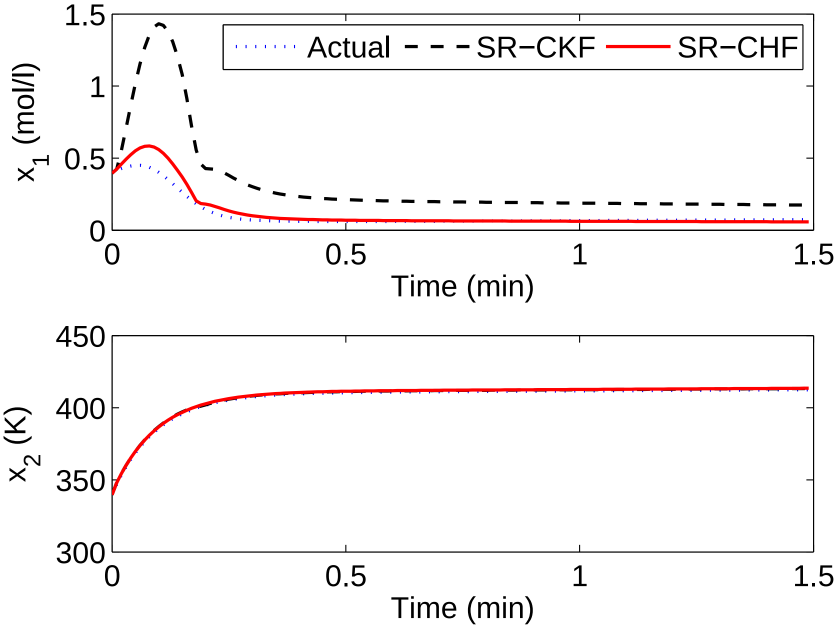

Figure 5.1: Actual and estimated states in the presence of non-Gaussian noises.

tuning parameters for the ***square root cubature H_∞ filter*** are $\gamma = 1$ and

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.5 \end{bmatrix} \quad and \quad R = 0.001$$

5.6.1 State estimation in the presence of non-Gaussian noises

In this sub-section, state estimation of the CSTR using a square root cubature H_∞ filter in the presence of non-Gaussian, non-zero mean, time-varying noises is considered. The process and measurement noises are

$$w_k = \begin{bmatrix} 0.01 + 0.05 \times sin(0.1k) & 0 \\ 0 & 1 + sin(0.1k) \end{bmatrix} \tag{5.77}$$

$$v_k = 1 + 0.1 \times sin(0.1k). \tag{5.78}$$

The CSTR model is initialised at $\mathbf{x}_0 = [0.4 \quad 340]^T$ and the chosen associated auxiliary matrix is $\mathbf{P}_{0|0} = [0.001 \quad 0; 0 \quad 0.01]$. The initial state estimate is randomly selected from

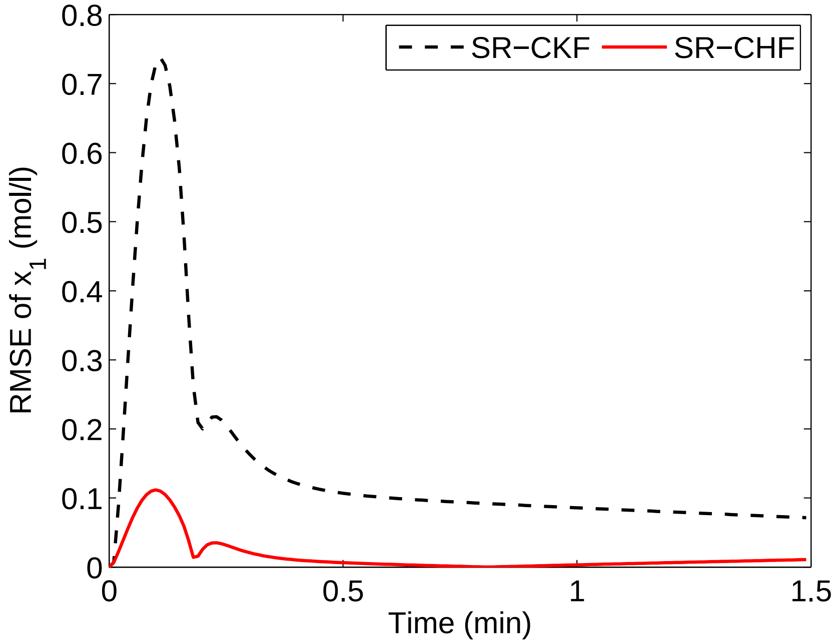

Figure 5.2: RMSEs of x_1 in the presence of non-Gaussian noises.

a uniform distribution of $\mathbb{U}(\mathbf{x}_0, \mathbf{P}_{0|0})$. The maximum magnitudes of w_k and v_k are used in the noise covariance matrices for the square root CKF. The actual and estimated states, and the RMSEs using square root CKF and square root cubature H_∞ filter are shown in Figures 5.1 and 5.2, respectively. The offsets in RMSE plots are due to the non-zero bias of the noises, which were intentionally added to verify the effectiveness of the square root cubature H_∞ filter in the presence of non-zero mean noises. The maximum RMSEs for the square root CKF and the square root cubature H_∞ filter are 0.7372 and 0.1116, respectively. Hence, the square root cubature H_∞ filter would appear to be well suited for non-linear state estimation in the presence of non-Gaussian noises. We have not compared the filters response in the presence of Gaussian noises as any Kalman filter's performance can be achieved by tuning the H_∞ filter parameters. Similarly, the square root CKF's performance can be replicated using the square root cubature H_∞ filter, but the reverse is not possible.

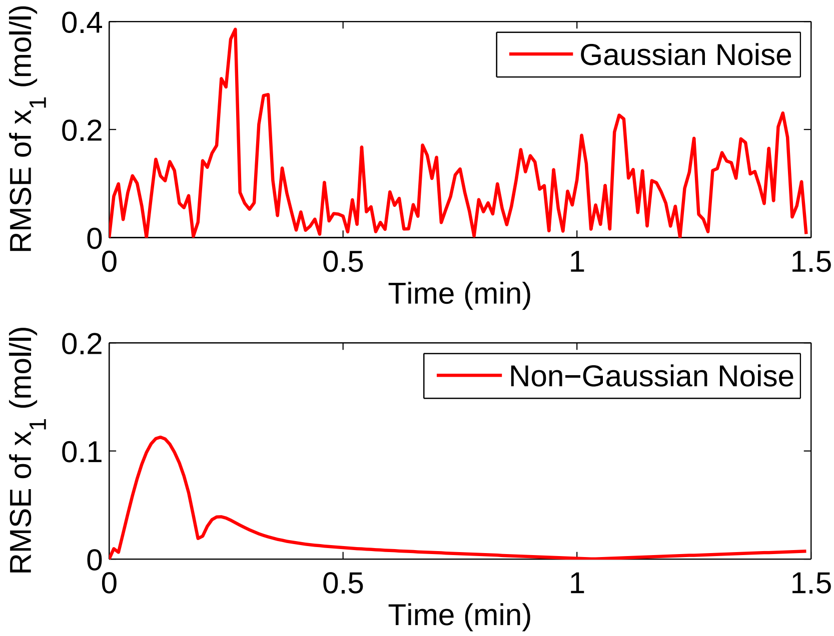

Figure 5.3: RMSEs of x_1 with perfect measurements.

5.6.2 State estimation with perfect measurements

To illustrate the effectiveness of the square root cubature H_∞ filter, the case of perfect measurements is considered. Non-square root estimators have the tendency to diverge due to singularity issues. There are many ad-hoc methods available to circumvent these problems. However, in most cases, the presence of the square root filter inherently solves these numerical issues [9, 8, 11]. Simulations using the *non-square root cubature H_∞ filter* for the perfect case are not presented, as its response diverges after $3-4$ time steps. In this section, for the non-Gaussian case, the simulations in the Section 6.1 are repeated with perfect measurements. In the Gaussian case, the standard deviations of the two states for process noise are considered as 0.1. The magnitude of the measurement noise is assumed to be zero (perfect measurement) for both Gaussian and non-Gaussian simulations. The corresponding results are shown in Figure 5.3. It can be seen that the RMSEs, using square root cubature H_∞ filters in the presence of Gaussian and non-Gaussian are within an acceptable range.

5.7 Conclusions

In this Chapter, we have presented a cubature H_∞ filter and its square root version for nonlinear systems. The proposed filter is derived from an extended H_∞ filter fused with a cubature Kalman filter. The advantages of the proposed filter are

1. It can deal with highly nonlinear systems with Gaussian and non-Gaussian noises.

2. It does not require the evaluation of Jacobians for nonlinear state estimation.

3. If the statistical properties of the noises (Gaussian) are known, then they can be incorporated in the proposed estimation method. If they are not known, then the $\mathbf{Q}$ and $\mathbf{R}$ matrices can be used as tuning parameters.

4. The overall robustness of the filter can be enforced by tuning the attenuation parameter, γ.

5. The square root cubature H_∞ filter is inherently numerically stable filter, as it propagates the square root of the auxiliary matrix.

The effectiveness of the square root cubature H_∞ filter was demonstrated by numerical simulations. The states of a continuous stirred tank reactor were estimated by using the square root cubature H_∞ filter in the presence of Gaussian as well as non-Gaussian noises.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis has presented a few linear and nonlinear state estimation methods, their extensions and applications. This section summarises the main findings and contributions.

6.1.1 An H_∞ filter based Sliding Mode Control

Linear state estimation methods and their application in control theory were explored. The Kalman filter and H_∞ filter were the main tools for linear estimation methods. Both Kalman and H_∞ are optimal filters, Kalman filter minimises the variance of the estimation error and H_∞ filter minimises the worst-case estimation errors. An H_∞ filter has several advantages over the Kalman filter, like its ability to deal with uncertain systems, non-Gaussian noises, etc. but its usage in control applications is not fully explored. The use of an H_∞ filter for a sliding mode controller (SMC) was proposed. The efficacy of the combined SMC-H_∞ filter was demonstrated on a quadruple-tank system. The controller and estimator designs were done for linearised model, but the simulations were performed on full nonlinear model. This combined approach was successful in controlling all four levels of the tank using only two states. The proposed scheme not only worked for Gaussian and non-Gaussian noises, but also worked for non-zero mean noises. More details can be seen in Chapter 2.

6.1.2 Simultaneous Localisation And Mapping (SLAM) using Cubature Kalman Filter (CKF)

Recently, the CKF was proposed as an alternative to EKF and UKF. It has not yet fully explored in the estimation, control and robotics communities. A solution to SLAM using CKF was proposed. The proposed solution does not requires the evaluation of Jacobians during the prediction and update stage and hence is a derivative free SLAM. The efficacy of the proposed algorithm is verified by simulations. Two types of Gaussian noises are used in the simulations and it was shown that CKF SLAM outperforms EKF and UKF SLAM, in both cases. More details are given in Chapter 3.

6.1.3 Cubature Information Filters

An information filter is a key tool to handle multi-sensor state estimation, where the information state and information matrix are propagated. One of the main advantages of the information over conventional filter is its simpler measurement update. When it comes to multi-sensor state estimation nonlinear systems, the preferred method EIF. EIF is an extension of information filter for nonlinear systems. One of the main limitations of EIF is the use of Jacobains in the prediction and measurement update and hence are suitable for only mild nonlinearities. It require evaluation of state and measurement Jacobians at every iteration. In Chapter 4, we have proposed a derivative-free cubature information filter (CIF) for nonlinear systems. The CIF was formed by embedding CKF with an EIF architecture. A square root version of CIF (SRCIF) was also proposed for numerical efficiency. Both CIF and SRCIF are further developed for multi-sensor state estimation. The CIF and SRCIF have the following desirable properties

1. They do not require the evaluation of Jacobians during the prediction and measurement update stages.

2. The update step is computationally simpler.
3. The SRCIF is numerically stable and reliable.
4. They are easy to extend for multi-sensor state estimation.

The applicability of the proposed SRCIF was demonstrated on multi-sensor state estimation of a permanent magnet synchronous motor model (PMSM). It was shown that, SRCIF when applied to PMSM outperforms square root extended information filter and square root unscented information filter. More details on CIFs can be seen in Chapter 4.

6.1.4 Cubature H_∞ Filters

An H_∞ filter has several advantages over conventional Kalman filter. A linear H_∞ filter can be easily extended to nonlinear systems by replacing the linear state and measurement matrices with their Jacobians, and by slightly modifying the predicted state equation. Similar to EIF, due to the requirement of Jacobians, EH_∞F is only suitable for mild nonlinearities. To circumvent this issue we proposed derivative-free cubature H_∞ filters (CH_∞F's) in Chapter 5. The CH_∞F was formed by embedding CKF with EH_∞F. Similar to SRCIF, a square root CH_∞F (SRCH_∞F) was derived for numerical efficiency. The CH_∞ and SRCH_∞F have the following desirable properties

1. It can deal with nonlinear systems with Gaussian and non-Gaussian noises.
2. It does not require the evaluation of Jacobians for nonlinear state estimation.
3. If the statistical properties of the noises (Gaussian) are known, then they can be incorporated in the proposed estimation method. If they are not known, then the **Q** and **R** matrices can be used as tuning parameters.
4. The overall robustness of the filter can be enforced by tuning the attenuation parameter, γ.

5. The square root cubature H_∞ filter is inherently numerically stable filter, as it propagates the square root of the auxiliary matrix.

The effectiveness of the SRCH_∞F was demonstrated by numerical simulations. The combined control-estimation problem for a continuous stirred tank reactor (CSTR) was considered. A feedback linearsation method was opted for a control design. The states of a CSTR were estimated by using the SRCH_∞F in the presence of Gaussian as well as non-Gaussian noises. These estimated states were fed back to feedback linearisation controller. More details on CH_∞Fs can be seen in Chapter 5.

Most of the work in this thesis has already been accepted or published in [98-107].

6.2 Future Work

The presented work can be further extended in several directions. One can broadly extend the work in the following areas

- Cubature H_∞ information filters
- Cubature information smoothers
- Stability analysis of the proposed methods
- Tuning of filter parameters
- Applications

6.2.1 Cubature H_∞ Information Filters

In this thesis, the CIF was derived to deal multi-sensor state estimation and the CH_∞F was derived for generic noises. These two methods can be fused to form a derivative-free cubature H_∞ information filter. The cubature H_∞ information filter can be handy for multi-sensor state estimation with generic noises for nonlinear systems. Preliminary work on cubature H_∞ information filter is given in the [106]; a square root version of this filter can further derived for numerical efficiency.

6.2.2 Cubature Information Smoothers

In estimation literature, smoother is basically an estimator which estimates the states using the future measurements. Smoothing can do a better job than the Kalman filter by using additional measurements made after the time of the estimated state vector [8]. Smoother plays an important role in several practical applications. One of our immediate future work is to develop a derivative-free cubature information smoother for nonlinear systems. Cubature information smoother propagates the information state and information vector and hence they can be easily extended for multi-sensor state estimation.

6.2.3 Stability Analysis of Proposed Methods

A few researchers have considered the stability analysis of EKF. From the stability analysis, it is possible to know the conditions for which the estimation error diverges. One of the obvious extensions of our work is to perform the stability analysis of the proposed methods. According to our knowledge, stability analysis of the CKF is not yet done. Before considering the stability analysis of the proposed methods, it is worth to perform the stability analysis of the CKF followed by CIF and CH_∞F.

6.2.4 Tuning of Filter Parameters

One of the important issues in the filtering methods is tuning. In this thesis, not much attention was given to the selection of tuning parameters, $\mathbf{Q}, \mathbf{R}, \hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0}, \gamma$, etc. If these tuning parameters are properly tuned based on some optimisation methods, then the overall performance can be further improved. One of the future research directions is to use some advanced optimisation methods to tune these parameters. In this thesis we have also not explored the parametric uncertainty analysis for the filters; specially, H_∞ filters have the capability to deal uncertain systems. This aspect of derivative-free CH_∞F can further be explored.

6.2.5 Applications

The proposed methods can be further extended to several challenging applications. In Chapter 4, the state estimation of open-loop PMSM is considered. For real-time implementation, high fidelity models with complete control-estimation mechanism is required and hence one can explore the proposed methods for real electrical machines. A couple of important applications of multi-sensor state are re-entry vehicle tracking and air traffic control [96,95], where the SRCIF can be further explored. The SRCH_∞F can further be explored in real-life applications where the Gaussian assumptions are not valid.

Appendix A

Lyapunov Stability for Discrete-time Systems

The discrete-time system can be written as

$$\mathbf{x}_k = \mathbf{f}[\mathbf{x}_{k-1}, k] \tag{A.1}$$

where k is a current time index, $\mathbf{x}_k \in \mathscr{R}^n$ is a state vector. Lyapunov stability theorem for discrete-time system [97] states that, if in a neighbourhood of the equilibrium point, $\mathbf{x}_e$, there exists a function, $\mathbf{V}$, such that

- $\mathbf{V}(x,k)$ is positive definite
- The rate of change of $\mathbf{V}(x,k)$, $\Delta\mathbf{V}(x,k)$, along any solution of Eq.(A.1) is negative semi-definite, *then the equilibrium point, $\boldsymbol{x}_e$, is stable*.

Bibliography

[1] D.−W. Gu, P. H. Petkov and M. M. Konstantinov, *Robust control design with MATLAB*, Springer, 2005.

[2] H. W. Sorenson, "Least−squares estimation: from Gauss to Kalman," IEEE Spectrum, vol.7, no. 7, pp. 63−68, 1970.

[3] A. N. Kolmogorov, W. L. Doyle and I. Selin, "Interpolation and extrapolation of stationary random sequences" Bulletin of academic sciences, Mathematics series, USSR, vol. 5, 1941. Translation by W. Doyle and J. Selin, RM−3090−PR, Rand Corporation, Santa Monica, California, 1962.

[4] N. Wiener, "Extrapolation, interpolation, and smoothing of stationary time series with engineering applications" Technology Press of Massachusetts Institute of Technology, and New York, Wiley, 1950.

[5] T. Kailath, "An innovations approach to least−squares estimation−Part I: Linear filtering in additive white noise," IEEE Transactions on Automatic Control, vol. 13, no. 6, pp. 646−655, 1968.

[6] T. Kailath, A. H. Sayed and B. Hassibi, *Linear estimation*, Prentice Hall, 2000.

[7] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Transactions of the ASME − Journal of Basic Engineering, vol. 82, pp. 35−45, 1960.

[8] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*, 3^{rd} edition, John Wiley & Sons, 2008.

[9] B. D. O. Anderson and J.B. Moore, *Optimal filtering*, Prentice−Hall, 1979.

[10] A. G. O. Mutambara, *Decentralized estimation and control for multi−sensor systems*, CRC Press, 1998.

[11] D. Simon, *Optimal estimation: Kalman, H_∞, and nonlinear approaches*, John Wiley & Sons, 2006.

[12] M. J. Grimble and A. El−Sayed, "Solution of the H_∞ optimal linear filtering problem for discrete−time systems," IEEE Transactions on Acoustics, Speech and Signal Processing, vol.38, no.7, pp.1092−1104, 1990 .

[13] K. M. Nagpal and P. P. Khargonekar, "Filtering and smoothing in an H_∞ setting," IEEE Transactions on Automatic Control, vol.36, no.2, pp.152−166, 1991.

[14] I. Yaesh and U. Shaked, "Game theory approach to optimal linear state estimation and its relation to the minimum H_∞-norm estimation," IEEE Transactions on Automatic Control, vol.37, no.6, pp.828−831, 1992.

[15] U. Shaked, "H_∞-minimum error state estimation of linear stationary processes," IEEE Transactions on Automatic Control, vol.35, no.5, pp.554−558, 1990.

[16] T. Basar, "Optimum performance levels for minimax filters, predictors and smoothers," Systems and Control Letters, vol.16, no.5, pp.309−317, 1991.

[17] R. N. Banavar and J. L. Speyer, "A linear−quadratic game approach to estimation and smoothing," in Proceedings of the American Control Conference, pp. 2818−2822, 1991.

[18] J.−J. E. Slotine and W. Li, *Applied nonlinear control*, Prantice−Hall, Englewood Cliffs, 1991.

[19] H. K. Khalil, *Nonlinear systems*, 3rd edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[20] R. L. Moose, “An adaptive state estimation solution to the maneuvering target problem,” IEEE Transactions on Automatic Control, vol.20, no.3, pp. 359−362, June 1975.

[21] R. Mehra, “On the identification of variances and adaptive Kalman filtering,” IEEE Transactions on Automatic Control, vol.15, no.2, pp. 175−184, April 1970.

[22] A. H. Mohamed and K. P. Schwarz, “Adaptive Kalman filtering for INS/GPS,” Journal of Geodesy, vol.73, pp. 193−203, 1999.

[23] P. G. Kaminski, A. E. Bryson and S. F. Schmidt, “Discrete square−root filtering−A survey of current techniques,” IEEE Transactions on Automatic Control, vol.16, no.6, pp.727−736, 1971.

[24] M. Morf and T. Kailath, “Square−root algorithms for least squares estimation,” IEEE Transactions on Automatic Control, vol.20, no.4, pp.487−497, 1975.

[25] M. Verhaegen and P. Van Dooren, “Numerical aspects of different Kalman filter implementations,” IEEE Transactions on Automatic Control , vol. 31, no. 10, pp. 907−917, 1986.

[26] N. A. Carlson, “Federated square root filter for decentralized parallel processors,” IEEE Transactions on Aerospace and Electronic Systems, vol. 26, no. 3, pp. 517−525, 1990.

[27] G. J. Bierman, M. R. Belzer, J. S. Vandergraft and D. W. Porter, “Maximum likelihood estimation using square root information filters,” IEEE Transactions on Automatic Control , vol. 35, no. 12, pp. 1293−1298, 1990.

[28] Mark L. Psiaki, “Square−root information filtering and fixed−interval smoothing with singularities,” Automatica, vol. 35, no. 7, pp. 1323−1331, 1999.

[29] P. Park and T. Kailath, “New square root algorithms for Kalman filtering,” IEEE Transactions on Automatic Control, vol. 40, no. 5, pp. 895−899, 1995.

[30] B. Hassibi, T. Kailath and A. S. Sayed, “Array algorithms for H_∞ estimation,” IEEE Transactions on Automatic Control, vol. 45, no. 4, pp. 702−706, 2000.

[31] L. A. McGee and S. F. Schmidt, “Discovery of the Kalman filter as a practical tool for aerospace and industry,” NASA Technical report, NASA−TM−86847, 1985.

[32] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proceedings of the IEEE, vol. 92, no.3, pp. 401−422, 2004.

[33] S. Julier and J. K. Uhlmann, “A New Method of the Nonlinear Transformation of Means and Covariances in Filters and Estimators,” IEEE Transactions on Automatic Control, vol. 45, no. 3, pp. 477−482, 2000.

[34] T. Vercauteren and X. Wang, “Decentralized sigma−point information filters for target tracking in collaborative sensor networks,” IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 2997 − 3009, 2005.

[35] M. Campbell and W. Whitacre, “Cooperative tracking using vision measurements on seascan uavs,” IEEE Transactions on Control Systems Technology, vol. 15, no. 4, pp. 613 − 626, 2007.

[36] D.−J Lee, “Nonlinear estimation and Multiple sensor fusion using unscented information filtering,” IEEE Signal Processing Letters, vol. 15, pp. 861−864, 2008.

[37] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant−Whyte and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” IEEE Transactions on Robotics and Automation, vol. 17, no.3, pp. 229−241, 2001.

[38] A. Doucet, S. Godsill and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” Statistics and Computing, vol. 10, no. 3, pp. 197−208, 2000.

[39] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, “A tutorial on particle filters for on−line nonlinear/non−Gaussian Bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174−188, 2002.

[40] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 910−927, 2000.

[41] C. P. Mracek, J. R. Cloutier and C. A. D’Souza, “A New Technique for Nonlinear Estimation,” Proceedings of IEEE Conference on Control Applications, pp.338−343, 1996.

[42] A. Nemra and N. Aouf, “Robust INS/GPS Sensor Fusion for UAV Localization Using SDRE Nonlinear Filtering,” IEEE Sensors Journal, vol. 10, no. 4, pp. 789−798, 2010.

[43] Y. Xiong and M. Saif, “Sliding mode observer for nonlinear uncertain systems,” IEEE Transactions on Automatic Control,, vol. 46, no. 12, pp. 2012−2017, 2001.

[44] J. Sarmavuori and S. Sarkka, “Fourier−Hermite Kalman filter,” IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1511−1515, 2012.

[45] I. Arasaratnam and S. Haykin, “Cubature Kalman Filters,” IEEE Transactions on Automatic Control, vol. 54, no. 6, pp. 1254−1269, 2009.

[46] R. van der Merwe, “Sigma−Point Kalman filters for probabilistic inference in dynamic state−Space models,” Ph.D Thesis, OGI School of Science and Engineering, Oregon Health and Science University, April 2004.

[47] X. M. Shen and L. Deng, “Game theory approach to discrete H_∞ filter design,” IEEE Transactions on Signal Processing, vol. 45, no. 4, pp. 1092−1095, 1997.

[48] G. A. Einicke and L.B. White, “Robust extended Kalman filtering,” IEEE Transactions on Signal Processing, vol. 47, no. 9, pp. 2596−2600, 1999.

[49] F. Yang, Z. Wang, S. Lauria and X. Liu, "Mobile robot localization using robust extended H_∞ filtering," Journal Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 23, no. 8, pp. 1067−1080, 2009.

[50] M. Csorba, "Simultaneous localisation and map building," Ph.D Thesis, University of Oxford, 1997.

[51] H. D.−Whyte, "Introduction to estimation and the kalman filter," Lecture notes, The university of Sydney, 2001 (Last accessed on 19/08/2012 from http://www.acfr.usyd.edu.au/pdfs/training/estKalFilter/EstimationNotesKC1.pdf).

[52] R. Vadigepalli, E. P. Gatzkem and F. J. Doyle III, "Robust control of a multivariable experimental four−tank system," Industrial and Engineering Chemistry Research, vol. 40, no. 8, pp. 1916−1927, 2001.

[53] K. H. Johansson, "The quadruple−tank process: a multivariable laboratory process with an adjustable zero," IEEE Transactions on Control Systems Technology, vol. 8, no. 3, pp. 456−465, 2000.

[54] M. M. Mercangoza and F. J. Doyle III, "Distributed model predictive control of an experimental four−tank system," Journal of Process control, vol. 17, no. 3, pp. 297−308, 2007.

[55] T. Raff, S. Huber, Z. K. Nagy and F. Allgower, "Nonlinear model predictive control of a four tank system: an experimental stability study," in Proceedings of the IEEE International Conference on Control Applications, Munich, 2006.

[56] J. K. Johnsen and F. Allgower, "Interconnection and damping assignment passivity−based control of a four−tank system," Lecture Notes in Control and Information Sciences, vol. 366, pp. 111−122, 2007.

[57] P. P. Biswas, R. Srivastava, S. Raya and A. Samanta, “Sliding mode control of quadruple tank process,” Mechatronics, vol. 19, no. 4, pp. 548−561, 2009.

[58] S. H. Said and F. M. Sahli, “A set of observers design to a quadruple tank process,” IEEE International Conference on Control Applications (IEEE Multi−conference on Systems and Control), USA, 2008.

[59] S. A. Imtiaz, K. Roy, B. Huang, S. L. Shah and P Jampana, “Estimation of states of nonlinear systems using a particle Filter,” IEEE International Conference on Industrial Technology, India, 2006.

[60] A. V. Okpanachi, “Developing Advanced Control strategies for a 4−Tank Laboratory process,” Master’s Thesis, Telemark University College, 2010.

[61] D. Luenberger, “An introduction to observers,” IEEE Transaction on Automatic Control, vol. 16, no.6, pp. 596−602, 1971.

[62] C. Edwards, S. K. Spurgeon and R. J. Patton, “Sliding mode observers for fault detection and isolation,” Automatica, vol. 36, no. 4, pp. 541−553, 2000.

[63] B. Bandyopadhyay, D. Fulwani and K. K.−Soo, “Sliding mode control using novel sliding surfaces,” Lecture Notes in Control and Information Sciences, Springer, vol. 392, 2009.

[64] B. Bandyopadhyay and D. Fulwani, “High−performance tracking controller for discrete plant using nonlinear sliding surface,” IEEE Transaction on Industrial Electronics, vol. 56, no. 9, pp. 3628−3637, 2009.

[65] G. Cheng and K. Peng, “Robust composite nonlinear feedback control with application to a servo positioning system,” IEEE Transactions on Industrial Electronics, vol. 54, no. 2, pp. 1132−1140, 2007.

[66] V. I. Utkin, J. Guldner, J. and Shi, J, “Sliding Mode Control in Electromechanical Systems,” second edition, CRC Press, 2009.

[67] A. Solin, “Cubature integration methods in non−linear Kalman filtering and smoothing,” Bachelor Thesis, Faculty of Information and Natural Sciences, Aalto University, Finland, 2010.

[68] T. Bailey, “Mobile robot localisation and mapping in extensive outdoor environments,” Ph.D Thesis, Australian Centre for Field Robotics, The university of Sydney, 2002.

[69] R. Smith and P. Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics,” in Autonomous Robot Vehicles, I.J. Cox and G.T. Wilfon, Eds. New York: Springer−Verlag, pp. 167¬193, 1990.

[70] R. Martinez−Cantin and J.A. Castellanos, “Unscented SLAM for Large−scale Outdoor Environments,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3427−3432, 2005.

[71] K. Murphy, “Bayesian map learning in dynamic environments,” in Proceedings of the Neural Information Processing Systems, pp. 1015−1021, 1999.

[72] M. Montemerlo, S. T. D. Koller and B. Wegbreit, “FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in Proceedings of International Conference on Artificial Intelligence, pp. 1151−1156, 2003.

[73] G. Grisetti, C. Stachniss, W. Burgard, “Improved techniques for grid mapping with Rao−Blackwellized particle filters,” IEEE Transactions on Robotics, vol. 23, no. 1, pp.34−46, 2007.

[74] C. Kim, R. Sakthivel and W. K. Chung, “Unscented FastSLAM: A robust and efficient solution to the SLAM Problem,” IEEE Transactions on Robotics, vol. 24, no. 4, pp. 808−820, 2008.

[75] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nobot, "Consistency of EKF−SLAM Algorithm," in Proceedings of the Intelligent Robots and Systems, pp. 3562−3568, 2006.

[76] [Online]. http://www−personal.acfr.usyd.edu.au/tbailey/software/slam_simulations.htm. Last accessed on 19/9/2012.

[77] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," Robotics and Autonomous Systems, vol. 57, no. 12, pp. 1198−1210, 2009.

[78] M. R. Walter, R. M. Eustice and J. J. Leonard, "Exactly sparse extended information filters for feature−based SLAM," The International Journal of Robotics Research, vol. 26, no. 4, pp. 335−359, 2007.

[79] R. M. Eustice, H. Singh, J. J. Leonard and M. R. Walter, "Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters," The International Journal of Robotics Research, vol. 25, no. 4, pp. 1223−1242, 2006.

[80] S. Huang, Z. Wang and G. Dissanayake, "Exact state and covariance sub−matrix recovery for submap based sparse EIF SLAM algorithm," in proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, 2008.

[81] J. R. Raol and G. Girija, "Sensor data fusion algorithms using square−root information filtering," IEE Proceedings − Radar, Sonar and Navigation, vol. 149, no. 2, pp. 89−96, 2002.

[82] G. Sibley, G. Sukhatme and L. Matthies, "The iterated sigma point filter with applications to long range stereo," in Proceddings of Robotics: Science and Systems, Philadelphia, USA, 2006.

[83] M. Campbell and W. Whitacre, "Cooperative tracking using vision measurements on seascan uavs," IEEE Trans. on Control Systems Technology, vol. 15, no. 4, pp. $613-626$, 2007.

[84] S. Julier, "The scaled unscented transformation," in Proceedings of IEEE American Control Conf., Anchorage, USA, 2002.

[85] Y. Bar−Shalom, X.−R. Li and T. Kirubarajan, *Estimation with applications to tracking and navigation*, John Wiley, 2001.

[86] N. Berman and U. Shaked , "H_∞ nonlinear filtering," International Journal of Robust and Nonlinear Control, vol. 6, no. 4, pp. 281−295, 1996.

[87] Y. Theodor and U. Shaked, "A dynamic game approach to mixed H_∞/H_2 estimation," International Journal of Robust and Nonlinear Control, vol. 6, no. 4, pp. 331−345, 1996.

[88] M. D. S. Aliyu and E. K. Boukas, " Discrete−time mixed H_2/H_∞ nonlinear filtering," International Journal of Robust and Nonlinear Control, vol. 21, no. 11, pp. 1257−1282, 2011.

[89] M. H. Terra, J. Y. Ishihara and G. Jesus, "Fast array algorithms for H_∞ information estimation of rectangular discrete−time descriptor systems," IEEE Conference on Control Applications and Intelligent Control, pp. 637−642, 2009.

[90] J. Y. Ishihara, B. Macchiavello and M. H. Terra, "H_∞ Estimation and array algorithms for discrete−time descriptor systems," 45th IEEE Conference on Decision and Control, pp. 4740−4745, 2006.

[91] A. Uppal A, W. H. Ray and A. B. Poore, "On the dynamic behavior of continuous stirred tank reactors," Chemical Engineering Science, vol. 29, no. 4, 967−985, 1974.

[92] M. J. Kurtz and M. A. Henson, “Nonlinear output feedback control of chemical reactors,” Proceedings of the American Control Conference, pp. 2667−2671, 1995.

[93] F. R. Garces, V. M. Becerra, C. Kambhampati and K. Warwick, *Strategies for feedback linearisation: a dynamic neural network approach*, Springer, 2003.

[94] L. Lang, W.−S. Chen, B. R. Bakshi, P. K. Goel and S. Ungarala, “Bayesian estimation via sequential Monte Carlo sampling − Constrained dynamic systems,” Automatica, vol. 43, no. 9, pp. 1615−1622, 2007.

[95] A. Lecchini, W. Glover, J. Lygeros and J. Maciejowski, “Air−traffic control in approach sectors: Simulation examples and optimisation ,” Lecture Notes in Computer Science, vol. 3414, pp. 433−448, 2005.

[96] X. R. Li and Y. Bar−Shalom, “Design of an interacting multiple model algorithm for air traffic control tracking,” IEEE Transactions Control Systems Technology, vol. 1, no. 3, pp. 186−194, 1993.

[97] Z.−Ping and Y. Wang, “A converse Lyapunov theorem for discrete−time systems with disturbances,” Systems and Control Letters, vol. 45, no. 1, pp. 49 − 58, 2002.

[98] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, “Square−root cubature information filter,” IEEE Sensors Journal, vol. 13, no. 2, pp. 750−758, 2013.

[99] A. Bajodah, H. M. Tariq, K. P. Bharani Chandra, R. Ahmed and D.−W Gu, “Fault tolerant control of aircraft actuating surfaces using generalized DI and integral SM Control,” Journal of Intelligent and Robotic Systems, vol. 69, pp. 181−188, 2013.

[100] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, “SLAM using EKF, EH_∞ and mixed EH_2/H_∞ filter,” in Proceedings of IEEE International Symposium on Intelligent Control (Multi−Conference on Systems and Control), Yokohama, Japan, Sept., 2010.

[101] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature information filter and its applications," in Proceedings of IEEE American Control Conference, California, July, 2011.

[102] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Fusion of an extended H_∞ filter and cubature Kalman filter," in Proceedings of 18th IFAC World Congress, Italy, Sept., 2011.

[103] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature Kalman filter based localization and mapping," in Proceedings of 18th IFAC World Congress, Italy, Sept., 2011.

[104] K. P. Bharani Chandra, H. Srikanthan, D.−W Gu, B. Bandyopadhyay and I. Postlethwaite, "Discrete−time sliding mode control using an H_∞ filter for a quadruple tank system," in Proceedings of 12th IEEE workshop on Variable structure systems (VSS), Bombay, India, Jan., 2012.

[105] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Nonlinear state estimation for induction and permanent magnet synchronous motors," in Proceedings of IEEE International workshop on Electronics Machine, Power Electronics and Engineering, Lushan, China, April, 2012.

[106] K. P. Bharani Chandra, D.−W Gu and I. Postlethwaite, "Cubature H_∞ information filter," European Control Conference, Zurich, Switzerland, July 2013.

[107] K. P. Bharani Chandra and D.−W Gu, "Nonlinear state estimation algorithms in aerospace control systems," Book chapter in Computational Intelligence in Aerospace Sciences, (Editors: M. Vasile and V. M. Becerra), AIAA, 2014.