Technische Universität München

Institute of Flight System Dynamics

Comparison of Gaussian Based Filters with Bayesian Filters for Nonlinear State Estimation

–

"Vergleich von Gauß- basierte Filter mit Bayes- Filter für Nichtlineare Zustandsschätzung"

Internship Report

Author: Mohit Mehndiratta

Matriculation Number: 03645448

Supervisor: Christoph Göttlicher

February 2015

Abstract

The Gaussian based Extended Kalman Filter (EKF) has become a widely used nonlinear state estimator for Flight Path Reconstruction purposes. But owing to its inherent flaw of linearizing the nonlinear system around recent state estimates, Sigma-Point Kalman Filter also known as Unscented Kalman Filter (UKF), was investigated as an alternative in (Mehndiratta, 2014), which is based on the propagation of the carefully created sigma points through the nonlinear system rather than its linearization. However, the Gaussian filters are primarily approximate nonlinear estimators as they assume the Gaussianity of the system, and thus only delay the inevitable divergence of the filter that will occur either when nonlinearities of the system become too severe or the system under investigation has non-Gaussian posterior density. To deal with this problem, Particle Filters (PFs) have been proposed, which are the sequential Monte Carlo methods based on point mass representation of probability densities.

This work executes a performance evaluation of the commonly used Gaussian filters (Extended Kalman Filter and Unscented Kalman Filter), with comparatively general Bayesian estimators (Particle Filters) regarding the nonlinear state estimation problem.

The descriptions of Extended Kalman Filter, Unscented Kalman Filter, and Particle Filters are presented in the second chapter with derivations included only for PFs. Firstly, the standard Kalman Filter for linear systems is discussed, followed by its further extension to the nonlinear systems in the form of Extended Kalman Filter. Thereafter, the Unscented Transformations are illustrated along with their application in the form of UKF. Later, the fundamentals of particle filtering are treated in detail which is followed by the implementation description of various types of PFs.

In the application chapter, the state estimation process is performed using Gaussian (EKF and UKF), and Bayesian filters for two example problems. First is the tracking problem of a Re-Entry vehicle, where the estimation performances of EKF and UKF are compared to the performance of Extended Particle Filter (a combined type of PF) on various grounds. Subsequently, an algebraic system with bimodal posterior density is analyzed, where EKF and various types of PFs are employed. The shortcomings of each filter, as evident from the results, are outlined and discussed in detail. Moreover, PFs are witnessed to deliver better results particularly for highly nonlinear/non-Gaussian state estimation (or tracking) problems.

Table of Contents

List of Figures

List of Tables

Table of Acronyms

Acronym	Description
APF	Auxiliary Particle Filter
BF	Bayesian Filters
EKF	Extended Kalman Filter

EPF	Extended Particle Filter
FPR	Flight Path Reconstruction
GF	Gaussian Filters
GRV	Gaussian Random Variable
i.i.d	independent identically distributed
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LKF	Linearised Kalman Filter
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MISE	Mean Integrated Square Error
MSE	Mean-square estimation error
OEM	Output Error Method
PF	Particle Filter
pdf	Probability density function
R-K	Runge-Kutta
RPF	Regularized Particle Filter
RV	Random variables
SIR	Sampling Importance Resampling Filter
SIS	Sequential Importance Sampling
SMC	sequential Monte Carlo
TSM	Two-step method
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
UT	Unscented Transform

Table of Symbols

Latin Letters		
Symbol	**Unit**	**Description**
$\boldsymbol{A}_k$	varying	Linearized system matrix
$\boldsymbol{B}_k$	varying	Linearized input matrix
$\boldsymbol{C}, \boldsymbol{C}_k$	varying	Linearized output matrix
$\boldsymbol{D}_k$	varying	Linearized feed-through matrix
$D_{\tilde{\boldsymbol{x}}}\boldsymbol{f}$	–	Differential operator
$\boldsymbol{E}_k$	varying	Linearized output process noise matrix
$\boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})$	varying	Dynamic system equation

$\boldsymbol{F}_d, \boldsymbol{F}_k$	varying	Linearized process noise distribution matrix
g	m/s^2	Gravitational acceleration
$\boldsymbol{G}_d, \boldsymbol{G}_k$	varying	Linearized measurement noise distribution matrix
$\boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}))$	varying	Dynamic output equation
$\boldsymbol{I}$	varying	Identity matrix
$K(\cdot)$	–	Rescaled Kernel density
$\boldsymbol{K}_k$	varying	Kalman filter gain matrix
L	–	Size of state vector
N_{eff}	varying	Effective sample size
N_{th}	varying	Threshold sample size
$P(\cdot \mid \cdot)$	–	Conditional probability
$p(\cdot)$	–	Probability density function
$\boldsymbol{P}_x$, $\boldsymbol{P}_k$	varying	State covariance matrix
$\boldsymbol{P}_{xy}$	varying	Cross covariance matrix
$\boldsymbol{P}_{yy}$	varying	Output covariance matrix
$q(\cdot)$	–	Proposal density function
$\boldsymbol{Q}_k$	varying	Process noise covariance matrix
$\boldsymbol{R}_k$	varying	Measurement noise covariance matrix
$\boldsymbol{u}_k$	varying	Input vector
w_k^i	varying	Associated weight of i^{th} particle
$\overline{\mathrm{w}}_k^i$	varying	Normalized associated weight of i^{th} particle
$\boldsymbol{w}(t), \boldsymbol{w}_k$	varying	Process noise vector
$W^{(m)}$	–	Weight of sigma point for computing mean
$W^{(c)}$	–	Weight of sigma point for computing covariance matrix
$\boldsymbol{x}_k$	varying	State vector at time k
$\boldsymbol{X}_k$	varying	State vector set from time $0, \dots, k$
$\hat{\boldsymbol{x}}_k$	varying	Estimated state vector
$\hat{\boldsymbol{y}}_k$	varying	Estimated output vector
$\boldsymbol{z}_k$	varying	Measurements at time k
$\boldsymbol{Z}_k$	varying	Measurement set from time $0, \dots, k$

Greek Letters

Symbol	Unit	Description
α	–	Spread of the sigma points around mean
β	–	Incorporate prior knowledge about the distribution
λ	–	Composite scaling parameter for sigma points
$\delta(\cdot)$	–	Impulse function
κ	–	Secondary scaling parameter for sigma points
$\boldsymbol{v}(t), \boldsymbol{v}_k$	varying	Measurement noise vector
$\boldsymbol{\Phi}, \boldsymbol{\Phi}_k$	varying	State transition matrix
$\boldsymbol{\Psi}, \boldsymbol{\Psi}_k$	varying	Discrete input matrix

$\boldsymbol{\chi}_i$	varying	Sigma point vector
$\boldsymbol{\chi}_{k-1}^{a}$	varying	Augmented form of sigma point vector
$\boldsymbol{\chi}_{k-1}^{w}$	varying	Sigma point vector of the process noise
$\boldsymbol{\chi}_{k-1}^{x}$	varying	Sigma point vector of the states
$\boldsymbol{\chi}_{k-1}^{v}$	varying	Sigma point vector of the measurement noise
$\boldsymbol{\Upsilon}_i$	varying	Transformed sigma point vector

Indices

Symbol	**Description**
0	Initial quantities
a	Augmented quantities used in case of general UKF
d	Discrete time variable
i	Quantity related to i^{th} particle
k	Time instant
w	Process noise quantities
v	Measurement noise quantities

1 Introduction

The technological advancements in modern computing and in flight test techniques, during the last few decades have made a remarkable impact on the implementation and theory of Aircraft parameter identification; wherein, the primary objectives of various flight tests are to obtain airworthiness certification and also, to estimate aircraft's performance and stability and control characteristics using linearized equations of motion. Numerous techniques for aircraft parameter identification, both in time and frequency domain, have been applied in the past and some of them are Maximum-Likelihood methods, Filter Error method and Two-Step method (TSM).

The Two-Step method is one of the prominent techniques that have been proved computationally efficient for aircraft system identification purpose. According to (J.A. Mulder, 1999), TSM disintegrates the state-parameter identification problem into a nonlinear state estimation problem and a subsequent parameter identification problem. The first step of TSM is often known as *Flight Path Reconstruction* (FPR). This report is entirely focused on Flight Path Reconstruction step of the TSM approach.

The relevance of the FPR (or the process of state estimation) is not only limited to the aerial vehicles, but also to any dynamical system which operates with uncertainties in the process and measurement models. Moreover, the accuracy of estimation depends entirely on the state estimator employed. For nonlinear state estimation in particular, numerous suboptimal techniques have been developed and most of them are based on Gaussian assumptions, and hence the name Gaussian based filters. One of the extensively used Gaussian filters, in the past and even today, is the *Extended Kalman Filter*. But, as it is based on the linearization of the nonlinear system, other new techniques of Gaussian filtering have been successfully implemented and amongst them is the Sigma Point Filtering (or *Unscented Kalman Filter*), which is based on the propagation of the sigma points rather than linearizing the system.

However, regardless of their easy implementation, Gaussian filters tend to be unreliable for highly nonlinear/non-Gaussian systems and also for the systems with modeling errors. This is due to their dependence on Gaussian assumption for posterior density and noises (process noise and measurement noise). To overcome this limitation (Gaussian assumption) and for accurate estimation of non-Gaussian posterior densities (especially, bimodal and heavily skewed distributions), non-Gaussian filters (or Bayesian estimators) were proposed; amongst which *Particle filters* are the most prominent. The Particle filter is a more generalized approach and does not need Gaussian assumption for posterior density, as it characterizes posterior density by a set of weighted samples (or particles), which were selected randomly. Therefore, the motivation of this report is to compare the performance of frequently used Gaussian filters (Extended Kalman Filter and Unscented Kalman filter) with the performance of a comparatively newer and more complicated approach of the Bayesian Filtering (Particle filters) for nonlinear state estimation.

The comparison was done with the help of results obtained for two considered examples. The first example involves the tracking problem of a Re-entry vehicle, wherein the performances of three filters (Extended Kalman Filter, Unscented Kalman filter and Particle filter) were analyzed together, based on various aspects. In the second example, the effectiveness of various type of Particle filters compared to Extended Kalman filter, was exploited for an algebraic system.

A thorough description of each topic was desired. The derivations of all the necessary equations (particularly, Particle filters) are provided in the respective sections. The focus was to provide an explicit description as far as possible, so that any engineering background personnel should be able to have an insight into the topic without referring to any other information source.

The work is broadly distributed among four chapters, starting with a short description about the topic in chapter 1 (Introduction). Chapter 2 (Theory) forms the main theoretical part of this report, explaining all the theoretical concepts. First, it briefs about the Flight Path Reconstruction step of the Two-Step method (section 2.1). Then the Linear Kalman Filter (section 2.2) is discussed without derivation, but a brief algorithm is presented. Afterwards, the Extended Kalman Filter (section 2.3) is explained using an algorithm. Successively, the Unscented Transforms (section 2.4) are illustrated and its application to an extension, as the Unscented Kalman Filter is shown in section 2.4.1. After that, a motivation for non-Gaussian filtering is provided in the form of shortcomings of Gaussian filtering (section 2.5), which is then accompanied by the theory of Bayesian estimation technique (section 2.6.1), introduction of Particle filters (2.6.2) and illustration of other (and combined) types of particle filtering approaches (section 2.6.3 and section 2.6.4).

An interpretation for all the obtained results is provided in chapter 3 (Application). Numerous observed behavior of all the filters, for both the examples are mentioned, along with their reasoning, in various sub-sections of this chapter.

Finally, the drawn conclusions are provided in chapter 4 (Conclusions and perspective).

2 Theory

2.1 Flight Path Reconstruction (FPR)

During a flight testing, quantities like *'rate of climb'* can be measured directly while some are rather difficult to measure (for example '*angle of attack*') and may lead to problems (presence of large measurement noise) if they are measured directly with the available instruments i.e. a pitot-tube and gyros. This forms the basis for flights path reconstruction. In simple terms, it is a technique of accurately determining or reconstructing the time history of the aircraft's position, attitude, velocities and rotational rates (or states to be precise) using the measurements made during a flight and simultaneously assuring consistency between the aircraft's kinematic equations and the flight data.

The FPR utilizes the redundancy present in the available measurements from various sensors, for instance inertial and air data sensors, which is crucial for accuracy, as the dynamic flight data is often susceptible to bias and scale factor errors along with the presence of process and measurement noise; and thus FPR gives the best estimate for the states along with an estimate of scale factors and bias errors. In a nutshell, FPR involves the reconstruction of states and sensors errors (if the state vector is augmented with additional states) using the kinematic equations that relate all the states (position, velocity and acceleration) including scale and bias errors, as unknown parameters.

Following the FPR, which forms the first step of TSM, the reconstructed states are used for aircraft dynamic modeling and subsequently, several performance, stability and control characteristics can then be obtained directly from the aerodynamic model. This is the second step of the TSM approach and one of the renowned methods used is the '*Output Error method (OEM)*', details of which can be seen in (Simon, 2006).

The maximum likelihood methods (like OEM) as stated above, are efficient in evaluating the states of a *deterministic system* i.e., a system in which the only uncertainty entering the dynamic system is the measurement noise. But for *stochastic systems*, these methods often yield poor results on account of the presence of an additional noise source, that is known as the 'process noise'; which signifies the presence of an uncertainty in the system model and can be regarded as 'Turbulence' for example. The stochastic system considered, is of the form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) + \boldsymbol{F}\boldsymbol{w}(t); \qquad \boldsymbol{x}(t=0) = \boldsymbol{x}_0 \\ \boldsymbol{y}(t) &= \boldsymbol{h}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) \\ \boldsymbol{z}_k &= \boldsymbol{y}_k + \boldsymbol{G}_d \boldsymbol{v}_k \end{aligned} \tag{2-1}$$

$$\boldsymbol{x} \in \mathbb{R}^L; \quad \boldsymbol{y}, \boldsymbol{z}_k \in \mathbb{R}^m; \quad \boldsymbol{u} \in \mathbb{R}^n; \quad \boldsymbol{w} \in \mathbb{R}^{n_w}; \quad \boldsymbol{v}_k \in \mathbb{R}^{n_v}; \quad \boldsymbol{F} \in \mathbb{R}^{L \times n_w}; \quad \boldsymbol{G}_d \in \mathbb{R}^{m \times n_v},$$

where, $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ are the assumed, additive Gaussian process noise and additive Gaussian measurement noise respectively. Also, $E[\boldsymbol{w}(t)] = 0$ and $E[\boldsymbol{v}(t)] = 0$.

The problem that often arises in estimating the states of a stochastic system is that the system equations can longer be directly integrated over time even with known initial condition $\boldsymbol{x}_0$ and inputs. This is on account of the presence of a random variable (process noise) in the

system's dynamic equations and thus leads to the requirement of *statistical methods* for its state estimation. Most of the statistical state estimators (often called 'filters') are based on the 'Bayesian formulation' that utilizes '*probability density functions (pdf)'* in describing the states and measured variables. The goal of these state estimators is to perform the 'optimal filtering', which refers to a methodology that estimates the states of a time varying system using the available noisy measurements. One of the widely used state estimator based on this approach is the 'Kalman Filter'.

2.2 The Kalman Filter

The goal of Kalman filtering is to have an unbiased, minimum variance state estimator which is realized by assuming the *posterior density* $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ to be Gaussian at every time step and thus can be fully parameterized by a mean and a covariance matrix. For the present case, rather than considering the complicated Bayesian (discussed later) derivation of the Kalman filter (also for Extended Kalman Filter and Unscented Kalman Filter, discussed subsequently), a simpler least-square approach is followed.

It is a two-step procedure which has a '*prediction step'* and a '*correction update step'*. The role of prediction step is to estimate the states based on the knowledge about the system. However, as the process noise is not measurable, only the deterministic part of the states are propagated in the prediction step and later these erroneous predictions are corrected at regular intervals with the measured data, during the *correction update*. Considering a Linear time-invariant, continuous time dynamic system

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{F}\boldsymbol{w}(t) \\ \boldsymbol{z}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{v}(t). \end{aligned} \tag{2-2}$$

The discrete time equivalent of the above equation is (see appendix A of (Mehndiratta, 2014) for detailed derivation)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}(\Delta t)\boldsymbol{x}_k + \boldsymbol{\Psi}(\Delta t) \cdot \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{\Psi}(\Delta t) \cdot \boldsymbol{F}_d\boldsymbol{w}_k \\ \boldsymbol{z}_k &= \boldsymbol{C}\boldsymbol{x}_k + \underbrace{\frac{1}{\sqrt{\Delta t}}\boldsymbol{G}}_{=\boldsymbol{G}_d}\boldsymbol{v}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{G}_d\boldsymbol{v}_k, \end{aligned} \tag{2-3}$$

where,

$$\boldsymbol{\Phi}_k = e^{\boldsymbol{A}_k\Delta t} = \sum_{i=0}^{\infty} \frac{(\boldsymbol{A}_k\Delta t)^i}{i!} \tag{2-4}$$

$$\boldsymbol{\Psi}_k = \int_0^{\Delta t} e^{\boldsymbol{A}_k\tau} d\tau = \sum_{i=1}^{\infty} \frac{\boldsymbol{A}_{\mathrm{k}}^{\mathrm{i}-1}\Delta t^i}{i!}. \tag{2-5}$$

The complete derivation of the standard Kalman filter for a linear system is omitted here, but can be seen in appendix A.2 of (Mehndiratta, 2014). Nevertheless, a brief algorithm of the Kalman filter for linear discrete system is mentioned below. Here, in representation of state vector $\hat{\boldsymbol{x}}_k^-$, the superscript '$-$' means that the estimation is an '*uncorrected prediction'* or a *'priori estimate'* whereas the subscript represents the time instant, k. On the contrary,

superscript '$+$' would mean that estimation is a '*corrected'* or *'a posteriori estimate'* with corresponding measurement update. This nomenclature is followed throughout the report.

1. The L dimensional discrete time dynamic system (Δt is removed for readability)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi x}_k + \boldsymbol{\Psi} \cdot \boldsymbol{B u}_k + \boldsymbol{\Psi} \cdot \boldsymbol{F}_d \boldsymbol{w}_k \\ \boldsymbol{z}_k &= \boldsymbol{C x}_k + \boldsymbol{G}_d \boldsymbol{v}_k \\ \boldsymbol{w}_k &\sim (\boldsymbol{0}, \boldsymbol{Q}_k) \\ \boldsymbol{v}_k &\sim (\boldsymbol{0}, \boldsymbol{R}_k) \end{aligned} \tag{2-6}$$

2. Initialization

$$\begin{aligned} \hat{\boldsymbol{x}}_0^+ &= E(\boldsymbol{x}_0) \\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T] \end{aligned} \tag{2-7}$$

3. Prediction
   From time $k = 1, \dots, N-1$

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^- &= \boldsymbol{\Phi} \hat{\boldsymbol{x}}_k^+ + \boldsymbol{\Psi} \cdot \boldsymbol{B u}_k \\ \boldsymbol{P}_{k+1}^- &= \boldsymbol{\Phi P}_k^+ \boldsymbol{\Phi}^T + \boldsymbol{\Psi F}_d \boldsymbol{Q}_k \boldsymbol{F}_d^T \boldsymbol{\Psi}^T \end{aligned} \tag{2-8}$$

4. Correction update

$$\begin{aligned} \boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^- \boldsymbol{C}^T \left(\boldsymbol{C P}_{k+1}^- \boldsymbol{C}^T + \boldsymbol{G}_d \boldsymbol{R}_k \boldsymbol{G}_d^T \right)^{-1} \\ \hat{\boldsymbol{x}}_{k+1}^+ &= \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{K}_{k+1} [\boldsymbol{z}_k - \boldsymbol{C x}_{k+1}^-] \\ \boldsymbol{P}_{k+1}^+ &= (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{C}) \boldsymbol{P}_{k+1}^- \end{aligned} \tag{2-9}$$

It may be worth noting that the Kalman filter behaves as an optimal state estimator, provided the state transition model and measurements are both linear and the present noise is Gaussian.

As known from experience, perfectly linear systems seldom exist; i.e., in reality almost all the systems are nonlinear. Also, the problem lies in the fact that still the optimum solution only exist for some particular nonlinear systems and there is no general solution available (as of yet) like in existence for the linear estimation. However, many suboptimal solutions had been found for nonlinear systems that lead to descent results.

Among various suboptimal approaches, some are based on the assumptions of Gaussianity of both, the posterior density and the present noise (process and measurement), and are often called as Gaussian Based Filters. The techniques that employ these assumptions are Linearized Kalman Filter (LKF), Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF).

Whereas, other approaches that do not employ Gaussian assumptions for the estimation of posterior densities are commonly known as Bayesian State Estimators and one prominent technique is the Particle Filter. In the present work, first the Extended Kalman Filter and Unscented Kalman Filter are discussed briefly along with their algorithms, followed by a detailed description of the Particle filters (including derivations).

2.3 The Extended Kalman Filter

It is a modified form of the Kalman filter (discussed previously), often used for nonlinear systems. As known from experience, in case of nonlinear estimation, the given nonlinear system is linearized about the nominal states, controls, outputs and noise values (for instance, if the system equations represent the dynamics of a missile then the nominal control would be those control surface deflections, given to achieve the states (position, velocity and attitude) and output, required to be on the planned mission trajectory to hit the target). In Linearized Kalman filter, deviation of true states from the nominal states is estimated instead of estimating the true states, which is the basis of Extended Kalman filter. Another implementation difference of Linearized Kalman filter from Extended Kalman filter is, that for EKF the nonlinear system is linearized about the most recent prediction of the Kalman filter instead of the nominal states and controls because in reality, the latter are very difficult to measure.

The fundamental goal behind EKF is to use the whole nonlinear dynamic system in state $\hat{\boldsymbol{x}}_k^-$ propagation, whereas the state error covariance matrix $\boldsymbol{P}_k^-$ and measurement updates are propagated in the form of Linearized system matrices. Here, only the algorithmic representation of equations for the standard EKF formulation (i.e. for system without feed-through) is provided; whereas, one can refer (Mehndiratta, 2014) for the detailed derivation of the equations of EKF, for the system with feed-through.

Algorithm of EKF for a standard system (without feed-through)

1. The considered $L-$ dimensional continuous-discrete time dynamic system is

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big) \tag{2-10}$$

$$\boldsymbol{z} = \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{v}(t)\big), \tag{2-11}$$

   also, $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ are the assumed additive white Gaussian process and measurement noise, which are defined as

$$\begin{aligned} \boldsymbol{w}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{Q}(t)\big) \\ \boldsymbol{v}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{R}(t)\big). \end{aligned} \tag{2-12}$$

2. The EKF filter is initialized as

$$\begin{aligned} \hat{\boldsymbol{x}}_0^+ &= E(\boldsymbol{x}_0) \\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T]. \end{aligned} \tag{2-13}$$

3. The states and error covariance matrix are propagated based on prediction step and later corrected from one measurement time to the next, till the end of simulation time N. That is, for $k \in \{1, \dots, N-1\}$

   a) Prediction step: propagation of the states and error covariance matrix at time instant k.

$$\hat{\boldsymbol{x}}_{k+1}^- = \hat{\boldsymbol{x}}_k^+ + \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}_{meas}(\tau), \boldsymbol{0}) d\tau \tag{2-14}$$

$$\begin{aligned} \boldsymbol{P}_{k+1}^{-} &= \boldsymbol{\Phi}_k \boldsymbol{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{\Psi}_k \boldsymbol{F}_k E\left[\boldsymbol{w}_k \boldsymbol{w}_k^T\right] \boldsymbol{F}_k^T \boldsymbol{\Psi}_k^T \\ &= \boldsymbol{\Phi}_k \boldsymbol{P}_k^{+} \boldsymbol{\Phi}_k^T + \boldsymbol{\Psi}_k \boldsymbol{F}_k \boldsymbol{Q}_k \boldsymbol{F}_k^T \boldsymbol{\Psi}_k^T, \end{aligned} \tag{2-15}$$

where,

$$\boldsymbol{A}_k = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t))}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^{+},\ \boldsymbol{u}=\boldsymbol{u}_k} \tag{2-16}$$

$$\boldsymbol{F}_k = \left. \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\right)}{\partial \boldsymbol{w}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^{+},\ \boldsymbol{u}=\boldsymbol{u}_k}, \tag{2-17}$$

and matrices $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Psi}_k$ are same as defined in equations (2-4) and (2-5), respectively.

b) Correction step: correction update of the states using available measurements at time $k+1$.

$$\begin{aligned} \boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^{-} \boldsymbol{C}_{k+1}^T \left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1}^{-} \boldsymbol{C}_{k+1}^T + \boldsymbol{G}_{k+1} E\left[\boldsymbol{v}_{k+1} \boldsymbol{v}_{k+1}^T\right] \boldsymbol{G}_{k+1}^T\right)^{-1} \\ &= \boldsymbol{P}_{k+1}^{-} \boldsymbol{C}_{k+1}^T \left(\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1}^{-} \boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1} \end{aligned} \tag{2-18}$$

$$\begin{aligned} \boldsymbol{P}_{k+1}^{+} &= (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{C}_{k+1}) \boldsymbol{P}_{k+1}^{-} (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{C}_{k+1})^T \\ &\quad + \boldsymbol{K}_{k+1} \boldsymbol{G}_{k+1} \boldsymbol{R}_{k+1} \boldsymbol{G}_{k+1}^T \boldsymbol{K}_{k+1}^T \\ &= (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{C}_{k+1}) \boldsymbol{P}_{k+1}^{-} \end{aligned} \tag{2-19}$$

$$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} + \boldsymbol{K}_{\boldsymbol{k+1}} \left(\boldsymbol{z}_{\boldsymbol{k+1}} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k+1}^{-}, \boldsymbol{0})\right), \tag{2-20}$$

where,

$$\boldsymbol{C}_{k+1} = \left. \frac{\partial \boldsymbol{h}\left(\boldsymbol{x}(t), \boldsymbol{\nu}(t)\right)}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1}^{-},\ \boldsymbol{u}=\boldsymbol{u}_{k+1}} \tag{2-21}$$

$$\boldsymbol{G}_{k+1} = \left. \frac{\partial \boldsymbol{h}\left(\boldsymbol{x}(t), \boldsymbol{\nu}(t)\right)}{\partial \boldsymbol{\nu}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1}^{-},\ \boldsymbol{u}=\boldsymbol{u}_{k+1}} \tag{2-22}$$

2.4 Unscented Transformation

The problem that often arises with a nonlinear system is, that it is difficult to transform a probability density function (pdf) through a general nonlinear function. The EKF works on the principle that "a linearized transformation of mean and covariance is approximately equal to the true nonlinear transformations" as mentioned in (Simon, 2006), but the approximations could sometimes be unsatisfactory. To overcome this shortcoming, the method of Unscented Transformations (UT) was suggested to estimate the *posterior density* $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ using a set of some carefully chosen *Sigma* points.

In the subsequent description of UT, a state vector $\boldsymbol{x} \in \mathbb{R}^L$ is considered, where L are the number of states and $\boldsymbol{f}(\boldsymbol{x})$ is a vector function of the nonlinear dynamic system, which is represented as

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}). \tag{2-23}$$

Using the Taylor series, the nonlinear system can be approximated around the mean $\bar{\boldsymbol{x}}$ as

$$\boldsymbol{y} = \boldsymbol{f}(\bar{\boldsymbol{x}}) + D_{\tilde{\boldsymbol{x}}}\boldsymbol{f} + \frac{1}{2!}D_{\tilde{\boldsymbol{x}}}^2\boldsymbol{f} + \frac{1}{3!}D_{\tilde{\boldsymbol{x}}}^3\boldsymbol{f} + \ \ldots, \tag{2-24}$$

where, $D_{\tilde{\boldsymbol{x}}}\boldsymbol{f}$ is a *differential operator* operating on the nonlinear function $\boldsymbol{f}(\boldsymbol{x})$ and is defined as

$$D_{\tilde{\boldsymbol{x}}}^k\boldsymbol{f} = \left(\sum_{i=1}^{L}\tilde{x}_i\frac{\partial}{\partial x_i}\right)^k \boldsymbol{f}(\boldsymbol{x})\Bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}, \tag{2-25}$$

where, $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$ and x_i represents the i^{th} state of the state vector $\boldsymbol{x}$.

According to (Hayking, 2001) "The unscented transformation (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation". Here, a random variable $\boldsymbol{x} \in \mathbb{R}^L$ is propagated through a nonlinear vector function, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ with the assumption that, $\boldsymbol{x}$ has a mean equal to $\bar{\boldsymbol{x}}$ and a covariance matrix given by $\boldsymbol{P}_{\boldsymbol{x}}$. The complete derivation of UT is out of scope, but can be found in (Mehndiratta, 2014); moreover, a brief description is presented to illustrate the concept.

To calculate the statistics of $\boldsymbol{y}$, a matrix $\boldsymbol{\chi}$ is formed with $2L + 1$ sigma vectors according to the following equation.

$$\begin{aligned}
\boldsymbol{\chi}_0 &= \bar{\boldsymbol{x}} \\
\boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} + \left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_i && i = 1, \ldots, L \\
\boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} - \left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_{i-L} && i = L+1, \ldots, 2L
\end{aligned} \tag{2-26}$$

where, $\boldsymbol{\chi}_i \in \mathbb{R}^L$ is the i^{th} sigma vector of the matrix $\boldsymbol{\chi}$ and $\left(\sqrt{(L+\lambda)\boldsymbol{P}_x}\right)_i$ represents the i^{th} column of a matrix square root of the covariance matrix $\boldsymbol{P}_x$. This matrix square root can be computed using 'Cholesky' or 'Schur' decomposition algorithm, details of which can be seen in (Simon, 2006).

These generated sigma vectors are then propagated through the nonlinear function yielding a cloud of transformed points which subsequently leads to the evaluation of the estimated mean and covariance matrix.

$$\boldsymbol{\Upsilon}_i = \boldsymbol{f}(\boldsymbol{\chi}_i) \qquad i = 0, \dots, 2L \tag{2-27}$$

$$\overline{\boldsymbol{y}} \approx \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\Upsilon}_i \tag{2-28}$$

$$\boldsymbol{P}_y = \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\Upsilon}_i - \overline{\boldsymbol{y}})(\boldsymbol{\Upsilon}_i - \overline{\boldsymbol{y}})^T, \tag{2-29}$$

where the weights W_i are evaluated as,

$$W_0^{(m)} = \frac{\lambda}{L+\lambda}$$

$$W_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \tag{2-30}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)} \qquad i = 1, \dots, 2L$$

and the constants used in the above equations are defined below.

$\lambda = \alpha^2(L + \kappa) - L$

$\alpha =$ Spread of the sigma points around $\overline{\boldsymbol{x}}$. It is usually set to a small positive value as a larger value would set sigma points far away from the mean.

$\approx (10^{-4} \leq \alpha < 1)$

$\kappa =$ Secondary scaling parameter and is set equal to $(3 - L)$, based on a heuristic approach.

$\beta =$ Incorporate a prior knowledge about the distribution

≈ 2 (Optimal for Gaussian distribution)

2.4.1 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is an extension of the Unscented Transform (UT) to the recursive estimation of the states. The basic difference between EKF and UKF lies in the propagation of Gaussian Random Variable (GRV), through the system dynamics. As known from (Simon J. Julier, 1995), “In EKF the state distribution is approximated by a GRV which is then propagated analytically through the first order linearization of the nonlinear system”. On the other hand, UKF is based on the intuition that, “With a fixed number of parameters it should be easier to approximate a Gaussian Distribution than it is to approximate an arbitrary nonlinear function” as mentioned in (Simon J. Julier, 1995). Following this intuition, a minimal set of sample (*Sigma*) points are carefully chosen that completely captures the true mean and covariance of the state distribution (that is, first and second order moments of the prior distribution), and when propagated through the true nonlinear system, they capture the posterior mean and covariance exactly to third order for any nonlinearity. This is better than

EKF which only estimates the mean accurately to first order and covariance to third order (the proof can be found in (Mehndiratta, 2014)).

2.4.1.1 UKF Algorithm for a General Case

A special notation is used here in representing the sigma points, that is, in $\boldsymbol{\chi}^{x}_{k|k-1}$ the superscript means that it is the sigma point about $\boldsymbol{x}$ whereas, subscript $k|k-1$ means that the sigma point is evaluated at time k, when the measurements are available till time $k-1$. In the same way, $\boldsymbol{\chi}^{w}_{k|k-5}$ would mean that the sigma point about $\boldsymbol{w}$ (process noise) are evaluated at time k using all the measurements available till time $k-5$.

1. The L-state continuous time dynamic system is given as,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big)\\ \boldsymbol{z} &= \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\big)\\ \boldsymbol{w}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{Q}(t)\big)\\ \boldsymbol{v}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{R}(t)\big)\end{aligned} \tag{2-31}$$

2. The UKF is initialized as follows.

$$\begin{aligned}\hat{\boldsymbol{x}}_0^{+} &= E[\boldsymbol{x}_0]\\ \boldsymbol{P}_0^{+} &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^{+})(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^{+})^T]\\ \hat{\boldsymbol{x}}_0^{a+} &= \begin{bmatrix}\hat{\boldsymbol{x}}_0^{+T} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}^T\\ \boldsymbol{P}_0^{a+} &= E[(\boldsymbol{x}_0^{a+} - \hat{\boldsymbol{x}}_0^{a+})(\boldsymbol{x}_0^{a+} - \hat{\boldsymbol{x}}_0^{a+})^T]\\ &= \begin{bmatrix}\boldsymbol{P}_0^{+} & 0 & 0\\ 0 & \boldsymbol{Q}_0 & 0\\ 0 & 0 & \boldsymbol{R}_0\end{bmatrix},\end{aligned} \tag{2-32}$$

   where, $\hat{\boldsymbol{x}}_k^a$ is the augmented form of the state vector which is the vertical concatenation of the original state vector, process noise vector and measurement noise vector; and is used as,

$$\hat{\boldsymbol{x}}_k^a = \begin{bmatrix}\hat{\boldsymbol{x}}_k{}^T & \boldsymbol{w}_k{}^T & \boldsymbol{v}_k{}^T\end{bmatrix}^T. \tag{2-33}$$

   Also, here L represents the dimension of the augmented state vector.

3. The following time update equations are used to propagate the state estimate and covariance from one measurement time to the next, till the end of the sampling time N.

   For $k \in \{1, \dots, N-1\}$

   Sigma points are calculated as,

$$\boldsymbol{\chi}_{k-1}^a = \begin{bmatrix}\hat{\boldsymbol{x}}_{k-1}^a & \hat{\boldsymbol{x}}_{k-1}^a + \gamma\sqrt{\boldsymbol{P}_{k-1}^a} & \hat{\boldsymbol{x}}_{k-1}^a - \gamma\sqrt{\boldsymbol{P}_{k-1}^a}\end{bmatrix}^T, \tag{2-34}$$

where, $\boldsymbol{\chi}_{k-1}^{a}$ is the augmented sigma point vector and is of the form,

$$\boldsymbol{\chi}_{k}^{a} = [(\boldsymbol{\chi}_{k}^{x})^{T} \quad (\boldsymbol{\chi}_{k}^{w})^{T} \quad (\boldsymbol{\chi}_{k}^{v})^{T}]^{T}. \tag{2-35}$$

Also, the scaling parameters γ and λ are

$$\begin{aligned} \gamma &= \sqrt{L+\lambda} \\ \lambda &= \alpha^{2}(L+\kappa) - L \end{aligned} \tag{2-36}$$

and α, κ and β are same as defined in UT.

The time update equations are:

$$\boldsymbol{\chi}_{k|k-1}^{x} = \boldsymbol{f}(\boldsymbol{\chi}_{k-1}^{x}, \boldsymbol{u}_{k-1}, \boldsymbol{\chi}_{k-1}^{w}) \tag{2-37}$$

$$\widehat{\boldsymbol{x}}_{k}^{-} = \sum_{i=0}^{2L} W_{i}^{(m)} \boldsymbol{\chi}_{i,k|k-1}^{x} \tag{2-38}$$

$$\boldsymbol{P}_{k}^{-} = \sum_{i=0}^{2L} W_{i}^{(c)} \left(\boldsymbol{\chi}_{i,k|k-1}^{x} - \widehat{\boldsymbol{x}}_{k}^{-}\right) \left(\boldsymbol{\chi}_{i,k|k-1}^{x} - \widehat{\boldsymbol{x}}_{k}^{-}\right)^{T} \tag{2-39}$$

$$\boldsymbol{\Upsilon}_{k|k-1} = \boldsymbol{h}\left(\boldsymbol{\chi}_{k|k-1}^{x}, \boldsymbol{u}_{k}, \boldsymbol{\chi}_{k-1}^{v}\right) \tag{2-40}$$

$$\widehat{\boldsymbol{y}}_{k}^{-} = \sum_{i=0}^{2L} W_{i}^{(m)} \boldsymbol{\Upsilon}_{i,k|k-1} \tag{2-41}$$

The measurements update equations are:

$$\boldsymbol{P}_{\tilde{\boldsymbol{y}}_{k}\tilde{\boldsymbol{y}}_{k}} = \sum_{i=0}^{2L} W_{i}^{(c)} \left(\boldsymbol{\Upsilon}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{k}^{-}\right) \left(\boldsymbol{\Upsilon}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{k}^{-}\right)^{T} \tag{2-42}$$

$$\boldsymbol{P}_{\boldsymbol{x}_{k}\boldsymbol{y}_{k}} = \sum_{i=0}^{2L} W_{i}^{(c)} \left(\boldsymbol{\chi}_{i,k|k-1}^{x} - \widehat{\boldsymbol{x}}_{k}^{-}\right) \left(\boldsymbol{\Upsilon}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{k}^{-}\right)^{T} \tag{2-43}$$

The update equations, similar to the standard Kalman filter equations, are written as

$$\boldsymbol{K}_{k} = \boldsymbol{P}_{\boldsymbol{x}_{k}\boldsymbol{y}_{k}} \boldsymbol{P}_{\tilde{\boldsymbol{y}}_{k}\tilde{\boldsymbol{y}}_{k}}^{-1} \tag{2-44}$$

$$\widehat{\boldsymbol{x}}_{k}^{+} = \widehat{\boldsymbol{x}}_{k}^{-} + \boldsymbol{K}_{k}(\boldsymbol{z}_{k} - \widehat{\boldsymbol{y}}_{k}^{-}) \tag{2-45}$$

$$\boldsymbol{P}_{k}^{+} = \boldsymbol{P}_{k}^{-} - \boldsymbol{K}_{k} \boldsymbol{P}_{\tilde{\boldsymbol{y}}_{k}\tilde{\boldsymbol{y}}_{k}} \boldsymbol{K}_{k}^{T}, \tag{2-46}$$

where the weights W_i are given by equation (2-30).

Furthermore, a special case which is often encountered in practice, where both process and measurement noise are purely additive, the computational complexity of the UKF algorithm can be reduced. For such cases, the state vector need not be augmented by noise RV's, which in turn reduces the dimension of sigma vector. The detailed algorithm for this situation, similar to the one provided before, can be found in (Mehndiratta, 2014).

2.5 Short comings of Gaussian filters (EKF and UKF)

Even though Gaussian based Extended Kalman Filter is the most widely used filtering strategy for the nonlinear systems, it can still be difficult to implement with tuned parameters and often gives unreliable estimates if the system under observation has severe nonlinearities. This is on account of the dependence of EKF algorithm on linearization to propagate the mean and covariance of the state. Other aspects of Linearization are:

- EKF can produce highly unstable filtering performance if the time of propagation is not sufficiently small; while a small time step would lead to high computational burden.
- Furthermore, the linearization can only be applied if the Jacobian matrices exist; as sometimes the nonlinear function is not differentiable.

On the other hand, Unscented Kalman Filter reduced linearization errors of EKF, by estimating the mean of a nonlinear system with higher order accuracy (third order for Gaussian systems) while, EKF estimates it correctly only till first order and therefore, UKF can provide significant improvements in estimation accuracy over EKF. However, it (UKF) is still an approximate nonlinear estimator, as it always assumes the Gaussianity of the system as well as the noise (both process and measurement), and thus merely defers the impending divergence of the filter that will occur when nonlinearities of the system or measurement become too severe.

In order to overcome the problem of divergence resulting from Gaussian assumptions and to include elements of nonlinearity for modeling the critical dynamics of a system accurately, the *Particle filters* (PFs) were proposed. A Particle filter is completely a nonlinear estimator as it is based on the non-Gaussian filtering technique. The fundamental principles and derivations of various types of particle filters are discussed in the following section.

2.6 Non-Gaussian Filtering

The origin of particle filtering dates back to early 1940s contributed by the work of Norbert Wiener and Metropolis, but was not fully explored then due to inadequate technological availability. Only by 1980s the technological advancements led to its implementation, but even then and now the computational burden of the particle filter is a main hindrance in its extensive use. The particle filter, which is a statistical method based on a brute-force approach, often produces good state estimates even for the system containing extreme nonlinearities in the system and measurement models for which the conventional Kalman filter either breaks down or obtains highly erroneous predictions. It is commonly known with some other names, including bootstrap filtering, Monte Carlo filtering, the condensation algorithm and survival of the fittest method.

The PF is a probability-based estimator and therefore to illustrate basics, the Bayesian approach for state estimation is discussed first in section 2.6.1, which is followed by a detailed description of the particle filters in section 2.6.2.

2.6.1 Bayesian State Estimation

The Bayesian approach to state estimation is briefly discussed in the current section. The mentioned approach is based on the Bayes' Rule which gives the conditional probability of an event given that another event has already occurred. That is,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2-47}$$

Where, $P(A|B)$ indicates the probability of occurring A when B has already occurred.

The L-dimensional nonlinear discrete-time dynamic system under consideration is of the form,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \\ \boldsymbol{z}_k &= \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \end{aligned} \tag{2-48}$$

$$\boldsymbol{x}_k \in \mathbb{R}^L; \quad \boldsymbol{u}_k \in \mathbb{R}^n; \quad \boldsymbol{w}_k \in \mathbb{R}^{n_w}; \quad \boldsymbol{z}_k \in \mathbb{R}^m; \quad \boldsymbol{v}_k \in \mathbb{R}^{n_v},$$

where, k indicates the time instant and $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ and $\boldsymbol{z}_k$ are the state vector, input vector and the measurement vector respectively. Also, $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the assumed independent, white process and measurement noise with known pdf's, $p_w(\boldsymbol{w}_k)$ and $p_v(\boldsymbol{v}_k)$, respectively. It is worth noting that, although the derivations of Bayesian estimator and particle filter (subsequent chapter) are provided for discrete time systems but they are equally applicable in case of continuous systems, as will be illustrated later in the algorithms of various particle filters.

The main aim of a Bayesian state estimator (or a filter) is to acquire a recursive way for computing the posterior conditional probability density function (pdf) of $\boldsymbol{x}_k$ on the basis of the available measurements $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_k$ and is represented as $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ which implies, probability density function of $\boldsymbol{x}_k$ conditional to all the measurements including at time k.

The first measurement is obtained at $k = 1$, hence the filter is initialized as

$$p(\boldsymbol{x}_0) = p(\boldsymbol{x}_0|\boldsymbol{Z}_0), \tag{2-49}$$

where, $\boldsymbol{Z}_0$ indicates the set containing no measurements.

Before achieving the aforementioned goal, another conditional pdf $p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})$ which is the conditional density of $\boldsymbol{x}_k$ provided all the measurements given prior to time k, is evaluated first using the law of total probability and is given as,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1}) &= \int p[(\boldsymbol{x}_k, \boldsymbol{x}_{k-1})|\boldsymbol{Z}_{k-1}] d\boldsymbol{x}_{k-1} \\ &= \int p[\boldsymbol{x}_k|(\boldsymbol{x}_{k-1}, \boldsymbol{Z}_{k-1})] p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{k-1}) d\boldsymbol{x}_{k-1}. \end{aligned} \tag{2-50}$$

Also, from the definition of the dynamic system in equation (2-48), it is evident that $\boldsymbol{x}_k$ is dependent on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{w}_{k-1}$ but independent of any measurement term. Thus,

$$p[\boldsymbol{x}_k|(\boldsymbol{x}_{k-1}, \boldsymbol{Z}_{k-1})] = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{2-51}$$

and its substitution back in equation (2-51) would result in,

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{k-1})d\boldsymbol{x}_{k-1}. \quad \textbf{(2-52)}$$

The pdf $p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{k-1})$ is the same posterior conditional pdf which is needed to be evaluated recursively but, it is available at the starting of the estimator. Also, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ represents the pdf of the state at time k given the states at the previous time $k-1$ and can be evaluated based on the knowledge of the system given in equation (2-48).

Now, the final posterior conditional pdf of the states $\boldsymbol{x}_k$ at time k can be represented using the Bayes' rule of equation (2-47) as,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_k) &= \frac{p(\boldsymbol{Z}_k|\boldsymbol{x}_k)}{p(\boldsymbol{Z}_k)}p(\boldsymbol{x}_k) \\ &= \frac{p[(\boldsymbol{z}_k,\boldsymbol{Z}_{k-1})|\boldsymbol{x}_k]}{p(\boldsymbol{z}_k,\boldsymbol{Z}_{k-1})}\underbrace{\frac{p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})p(\boldsymbol{Z}_{k-1})}{p(\boldsymbol{Z}_{k-1}|\boldsymbol{x}_k)}}_{p(\boldsymbol{x}_k)} \\ &= \frac{p(\boldsymbol{x}_k,\boldsymbol{z}_k,\boldsymbol{Z}_{k-1})}{p(\boldsymbol{x}_k)p(\boldsymbol{z}_k,\boldsymbol{Z}_{k-1})}\frac{p(\boldsymbol{x}_k,\boldsymbol{Z}_{k-1})p(\boldsymbol{Z}_{k-1})}{p(\boldsymbol{Z}_{k-1})p(\boldsymbol{Z}_{k-1}|\boldsymbol{x}_k)} \times \frac{p(\boldsymbol{x}_k,\boldsymbol{z}_k)}{p(\boldsymbol{x}_k,\boldsymbol{z}_k)} \end{aligned} \quad \textbf{(2-53)}$$

The above equation can be reduced to the following form by converting the ratios of various pdfs to marginal pdfs, where the details can be seen in (Simon, 2006).

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_k) = \frac{p[\boldsymbol{Z}_{k-1}|(\boldsymbol{x}_k,\boldsymbol{z}_k)]p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})p(\boldsymbol{Z}_{k-1}|\boldsymbol{x}_k)} \quad \textbf{(2-54)}$$

Also, it can be witnessed from equation (2-48) that $\boldsymbol{y}_k$ is a function of $\boldsymbol{x}_k$ and $\boldsymbol{\nu}_k$ only. Therefore,

$$p[\boldsymbol{Z}_{k-1}|(\boldsymbol{x}_k,\boldsymbol{z}_k)] = p(\boldsymbol{Z}_{k-1}|\boldsymbol{x}_k) \quad \textbf{(2-55)}$$

and thus, the previous equation reduces to

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_k) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})}. \quad \textbf{(2-56)}$$

The pdf $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ is obtained from the system's measurement equations and $p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})$ is computed from equation (2-52). Also, the pdf $p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})$ is evaluated in a similar way as used for pdf $p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})$. That is,

$$\begin{aligned} p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1}) &= \int p[(\boldsymbol{z}_k,\boldsymbol{x}_k)|\boldsymbol{Z}_{k-1}]d\boldsymbol{x}_k \\ &= \int p[\boldsymbol{z}_k|(\boldsymbol{x}_k,\boldsymbol{Z}_{k-1})]p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{x}_k \end{aligned} \quad \textbf{(2-57)}$$

and, as said before that $\boldsymbol{z}_k$ is independent of $\boldsymbol{Z}_{k-1}$, implies

$$p[\boldsymbol{z}_k|(\boldsymbol{x}_k,\boldsymbol{Z}_{k-1})] = p(\boldsymbol{z}_k|\boldsymbol{x}_k). \quad \textbf{(2-58)}$$

By substituting the above expression back in equation (2-57), a final expression for the posterior conditional pdf of the state vector $\boldsymbol{x}_k$ is obtained.

$$p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1}) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{x}_k \quad \textbf{(2-59)}$$

The analytical solutions for the above Bayesian state estimator is only available for some special cases. For instance, if the system and process models are linear and the process noise $\boldsymbol{w}_k$ and measurement $\boldsymbol{\nu}_k$ noise are assumed to be additive Gaussian white noise, then the achieved solution is the known Kalman filter. The above mentioned approach for obtaining the standard Kalman filter is far more complex than the one obtained based on the least-square methodology, as presented in (Simon, 2006). It is worth noting that the Bayesian approach apparently proves the optimality of the Kalman filter provided the noise (both process noise and measurement noise) are Gaussian. Whereas, the least-square derivation given in (Simon, 2006), justifies the Kalman filter to be optimal linear filter irrespective of the noise pdfs.

2.6.2 The Particle Filter

The particle filter was proposed in 1993 as an approximation to numerically implement the nonlinear Bayesian estimator. According to (Arulampalam, 2002) “Particle filters are sequential Monte Carlo methods based on the point mass (or particle) representations of probability densities, which can be applied to any state-space model and also generalize the traditional Kalman filtering methods”. These methods result in a complete representation of the states posterior distribution, which eventually lead to an easy computation of any statistical estimate and thus can treat any nonlinearity or distribution. Another advantage of sequential Monte Carlo method (or particle filter) is that, the approximations are centered on probability distributions (or pdfs) in spite of compromising the accuracy of the dynamic model as done in Gaussian approximation based Kalman filters.

2.6.2.1 Sequential Importance Sampling

The algorithm that forms the basis in implementing the sequential MC (SMC) methods is known as *Sequential Importance Sampling* (SIS) and its main idea is to represent the required posterior density using a set of random samples (or particles) along with their associated weights, which later results in the computation of the state estimates (Arulampalam, 2002). Also, it is worth mentioning that as the number of samples increases infinitely, the mentioned representation of the posterior density becomes equivalent to its true value and SIS based filter approaches an optimal Bayesian estimate.

In the present context, $\{\boldsymbol{x}_{0:k}^i, \mathrm{w}_k^i\}_{i=1}^{N_s}$ denote the *random measure* which characterizes the complete posterior pdf $p(\boldsymbol{x}_{0:k}|\boldsymbol{Z}_k)$, where N_s are the number of samples, $\{\boldsymbol{x}_{0:k}^i, i = 0, \dots, N_s\}$ is a set representing the support points along with their associated weights $\{\mathrm{w}_k^i, i = 0, \dots, N_s\}$ and $\boldsymbol{x}_{0:k} = \{\boldsymbol{x}_j, j = 0, \dots, k\}$ is a set of the state vector up to time k. Using the mentioned terms and representing the set $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k$ (or $\boldsymbol{x}_{0:k}$) as $\boldsymbol{X}_k$ (also done for the measurements), the joint posterior density at time k can be represented as,

$$p(\boldsymbol{X}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \delta(\boldsymbol{X}_k - \boldsymbol{X}_k^i), \quad \textbf{(2-60)}$$

where, $\overline{\mathrm{w}}_k^i$ are the normalized weights such that

$$\sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i = 1. \tag{2-61}$$

The PF recursively estimates the location and associated weight of each particle, where its location and weight at each time simultaneously reflects the posterior density value in that particular region of the state space. Before presenting the equations for SIS, a special sampling technique known as the Importance Sampling, is described below as it constitutes a vital role in SIS based filters.

2.6.2.1.1 Importance Sampling

The focus in the current section is to approximate the expected mean of a nonlinear function based on the known probability density $p(\cdot)$. But in certain cases it is troublesome to draw samples from the underlying density $p(\cdot)$ and hence the idea is to use a proposal density $q(\cdot)$ instead, for producing samples. Therefore, the expected mean of an arbitrary function $g(\cdot)$ also given in (Karlsson, 2005) can be represented using the proposal density as,

$$E[g(\boldsymbol{x})] = \int g(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = \int g(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x}) d\boldsymbol{x} \tag{2-62}$$

Now, in the above equation the mean value of $g(\boldsymbol{x})$ would be evaluated by computing the mean of $g(\boldsymbol{x})p(\boldsymbol{x})/q(\boldsymbol{x})$ with samples taken from the density $q(\boldsymbol{x})$. Also, the integral can be approximated as,

$$E[g(\boldsymbol{x})] = \frac{1}{N_s} \sum_{i=1}^{N_s} \underbrace{\frac{p(\boldsymbol{x}^i)}{q(\boldsymbol{x}^i)}}_{\mathrm{w}^i} g(\boldsymbol{x}^i), \tag{2-63}$$

where, $\{\boldsymbol{x}_k^i\}_{i=1}^{N_s}$ are the same sample set, which is also known as *independent identically distributed* (i.i.d.) samples, from the *importance density* $q(\cdot)$ and the i^{th} *importance weight* is defined as $\mathrm{w}^i = p(\boldsymbol{x}^i)/q(\boldsymbol{x}^i)$.

Now, if the samples $\boldsymbol{X}_k^i$ are drawn from the importance density $q(\boldsymbol{X}_k|\boldsymbol{Z}_k)$, then the normalized weights in equation (2-60) can be written as,

$$\overline{\mathrm{w}}_k^i \propto \frac{p(\boldsymbol{X}_k^i|\boldsymbol{Z}_k)}{q(\boldsymbol{X}_k^i|\boldsymbol{Z}_k)}. \tag{2-64}$$

To obtain the weight update equation an expression for the complete posterior density can be written using Bayes' theorem, conditional probability definition and the inherent Markov property in the state space model, as can be seen in (Karlsson, 2005). That is,

$$
\begin{aligned}
p(\boldsymbol{X}_k|\boldsymbol{Z}_k) &= \frac{p(\boldsymbol{z}_k|\boldsymbol{X}_k,\boldsymbol{Z}_{k-1})p(\boldsymbol{X}_k|\boldsymbol{Z}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})} \\
&= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{X}_{k-1},\boldsymbol{Z}_{k-1})p(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})} \\
&= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})}p(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{k-1})
\end{aligned}
\tag{2-65}
$$

and if the normalization factor, that is the denominator, is ignored the expression of the complete posterior density reduces further to,

$$
p(\boldsymbol{X}_k|\boldsymbol{Z}_k) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{k-1}). \tag{2-66}
$$

Now if the considered proposal density is written using Bayes' theorem,

$$
q(\boldsymbol{X}_k|\boldsymbol{Z}_k) = q(\boldsymbol{x}_k|\boldsymbol{X}_{k-1},\boldsymbol{Z}_k)q(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_k) \tag{2-67}
$$

and assuming that it takes the final form,

$$
q(\boldsymbol{X}_k|\boldsymbol{Z}_k) = q(\boldsymbol{x}_k|\boldsymbol{X}_{k-1},\boldsymbol{Z}_k)q(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{k-1}), \tag{2-68}
$$

a recursive expression for the weight updates can be obtained by substituting equation (2-66) and equation (2-68) back in equation (2-64).

$$
\begin{aligned}
\overline{\mathrm{w}}_k^i &\propto \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)p(\boldsymbol{X}_{k-1}^i|\boldsymbol{Z}_{k-1})}{q(\boldsymbol{x}_k^i|\boldsymbol{X}_{k-1}^i,\boldsymbol{Z}_k)q(\boldsymbol{X}_{k-1}^i|\boldsymbol{Z}_{k-1})} \\
&= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)}{q(\boldsymbol{x}_k^i|\boldsymbol{X}_{k-1}^i,\boldsymbol{Z}_k)}\overline{\mathrm{w}}_{k-1}^i
\end{aligned}
\tag{2-69}
$$

Furthermore, if the proposal density is chosen to be only dependent on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{z}_k$, which is particularly useful in the more likely case when only filtered estimate of $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ is required at each time step, the modified weight update equation becomes

$$
\overline{\mathrm{w}}_k^i = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i,\boldsymbol{z}_k)}\overline{\mathrm{w}}_{k-1}^i \tag{2-70}
$$

But for a particular choice of the proposal density which is often more intuitive and convenient to implement,

$$
q(\boldsymbol{x}_k^i|\boldsymbol{X}_{k-1}^i,\boldsymbol{Z}_k) = p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i), \tag{2-71}
$$

a simplified expression of the weight updates can be obtained.

$$
\overline{\mathrm{w}}_k^i \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)\overline{\mathrm{w}}_{k-1}^i, \tag{2-72}
$$

where, $\overline{\mathrm{w}}_k^i$ depicts the normalized weight of the i^{th} particle at time k.

Finally, an expression for the marginal posterior distribution can be obtained by integrating the approximated joint posterior distribution,

$$p(\boldsymbol{X}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \delta\left(\boldsymbol{X}_k - \boldsymbol{X}_k^i\right) \tag{2-73}$$

keeping in mind a mathematical result for impulse functions

$$\delta(\boldsymbol{X}_k) = \delta(\boldsymbol{x}_{0:k}) = \prod_{i=0}^{k} \delta(\boldsymbol{x}_i) = \delta(\boldsymbol{x}_0)\delta(\boldsymbol{x}_1) \ldots \delta(\boldsymbol{x}_k) \tag{2-74}$$

Therefore,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_k) &= \int p(\boldsymbol{X}_k|\boldsymbol{Z}_k)\, d\boldsymbol{X}_{k-1} \\ &= \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \int \delta\left(\boldsymbol{X}_k - \boldsymbol{X}_k^i\right) d\boldsymbol{X}_{k-1} \\ &= \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \delta\left(\boldsymbol{X}_k - \boldsymbol{X}_k^i\right) \end{aligned} \tag{2-75}$$

2.6.2.1.2 Resampling

A common problem that arises in SIS particle filter is the *degeneracy phenomena* on account of which all but one particle will have negligible weight after few iterations. As shown in (Doucet, 1998), the variance of the importance weights can only increase over time and thus the degeneracy problem is inevitable and would later lead to divergence. To avoid it, a selection or resampling strategy is implemented in particle filtering. As suggested in (Bergman, 1999), a suitable measure to account for degeneracy is the *effective sample size* N_{eff} and is defined as,

$$N_{eff} = \frac{N_s}{1 + Var\left(\overline{\mathrm{w}}_k^{*i}\right)}, \tag{2-76}$$

where, $\overline{\mathrm{w}}_k^{*i} = p\left(\boldsymbol{x}_k^i|\boldsymbol{Z}_k\right)/q\left(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i, \boldsymbol{z}_k\right)$ and is known as the *true weight*. But the above expression cannot be evaluated analytically and hence an approximation is often used which is given as,

$$\widehat{N}_{eff} \approx \frac{1}{\sum_{i=1}^{N_s}\left(\overline{\mathrm{w}}_k^i\right)^2}, \tag{2-77}$$

where, $\overline{\mathrm{w}}_k^i$ is the normalized weight as defined previously.

A physical way of looking at $\widehat{N}_{eff}$ is that it quantifies the number of samples (or particles) which are effectively contributing in the estimation process, as most of the samples will have lower weights due to degeneracy. Moreover, the two extreme cases for $\widehat{N}_{eff}$ can be accounted as

- Case 1: Equally weighted samples, that is $\overline{\mathrm{w}}_k^i = \frac{1}{N_s}$, then

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}\left(\frac{1}{N_s}\right)^2} = \frac{1}{N_s\left(\frac{1}{N_s}\right)^2} = N_s \tag{2-78}$$

- Case 2: All weights are zero except one (severe degeneracy), then

$$\widehat{N}_{eff} = \frac{1}{1} = 1 \tag{2-79}$$

Therefore, $\widehat{N}_{eff}$ is always bounded, $1 \leq \widehat{N}_{eff} \leq N_s$ and whenever a significant degeneracy is observed that is, the effective number of samples $\widehat{N}_{eff}$ falls below a certain threshold N_{th}, the resampling is applied to reduce the effect of degeneracy.

The fundamental idea behind resampling is to apply a concept called *Survival of the fittest* which means replacing the unlikely samples (or particles) that have smaller weights with the more likely ones so as to avoid the computation effort spent on updating the particles that have almost zero influence in approximating the posterior distribution. In its application according to (Arulampalam, 2002), a new sample set $\{\boldsymbol{x}_k^{j*}\}_{j=1}^{N_s}$ is generated by resampling with replacement N_s times from an approximate discrete representation of the posterior $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ given as,

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), \tag{2-80}$$

such that, $\mathrm{Prob}(\boldsymbol{x}_k^{j*} = \boldsymbol{x}_k^j) = \overline{\mathrm{w}}_k^j$. Also, the resulting sample from the posterior density of equation (2-80) is an i.i.d. (independent identically distributed) sample and thus the new weights are reset to $\overline{\mathrm{w}}_k^j = 1/N_s$. One of the resampling scheme used to implement the mentioned resampling technique is *Systematic Resampling* (also known as minimum variance resampling). One can note that although other efficient resampling schemes like *Stratified Resampling, Residual Resampling* are available but systematic resampling is preferred in the report on account of its easier implementation, performing $O(N_s)$ operations and also its minimum variance approach. A brief algorithm describing its working is presented below, where $\mathbb{U}[a,b]$ represents a uniform distribution on the interval $[a,b]$ (including the end limits). One may note that the following algorithm also stores the parent index i^j (although not required here), as it is utilized by one of the filter described later (i.e. *Auxiliary particle filter*).

Systematic Resampling algorithm

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

- Initialization of cumulative distribution function (CDF) : $c_1 = 0$
- $For\ \ i = 2 : N_s$
  - Construction of CDF : $c_i = c_{i-1} + \overline{\mathrm{w}}_k^i$
- $End\ For$
- Iteration is started from the bottom of the CDF : $i = 1$
- Starting point is drawn : $u_1 \sim \mathbb{U}[0, N_s^{-1}]$
- $For\ \ j = 1 : N_s$

- CDF is traversed : $u_j = u_1 + N_s^{-1}(j-1)$
- $While\ u_j > c_i$
  - Increment of index : $i = i + 1$
- $End\ While$
- Updated sample : $\boldsymbol{x}_k^{j*} = \boldsymbol{x}_k^i$
- Updated weight : $\overline{w}_k^j = N_s^{-1}$
- Recorded parent index : $i^j = i$

➢ $End\ For$

Even though resampling step helps in reducing the effects of degeneracy, but at the same time it introduces some other practical problems. They are

1. Resampling limits the ability to parallelize the particle filter implementation since all the particle must be combined (Arulampalam, 2002).
2. Sample Impoverishment: The particles that have high weights $\overline{w}_k^i$ are more likely to get selected during resampling and hence result into a loss of diversity among the particles as the obtained distribution will have many repeated points. This effect is more prominent in the estimation of systems with small process noise, and in fact all the particles will collapse to a single point within few iterations if the system under consideration has negligible process noise (Arulampalam, 2002).

Various techniques have been developed to overcome these issues and some of them are *Roughening*, *Prior Editing*, *Regularized Particle Filter*, *Markov Chain Monte Carlo (MCMC) Resampling* and *Auxiliary Particle Filter*. Among all the mentioned schemes, only *Regularized Particle Filter* and *Auxiliary Particle Filter* are discussed in detail in the later sections, whereas for further reading on other approaches one may refer (Simon, 2006).

2.6.3 Other Types of Particle Filters

The sequential importance sampling as mentioned in the previous section forms the basis for most of the particle filters. Hence, the various versions of PFs found in the literature can be accounted as special cases of the general SIS algorithm, which are usually derived either by appropriate choice of importance density or/and by modified resampling step (Arulampalam, 2002).

In the current section, three PFs that were proposed in the literature are presented, along with a brief derivation to illustrate the modifications made to obtain their final forms from the SIS algorithm. The considered PFs are

1. Sampling Importance Resampling (SIR) Filter;
2. Regularized Particle Filter (RPF);
3. Auxiliary Particle Filter (APF).

Among all the mentioned filters, the SIR particle filter is a direct implementation of the Sequential importance sampling method and hence, is easier to implement; whereas the other two filters are computationally more complex, but at the same time show improved results in counteracting the problem of sample impoverishment, described previously.

2.6.3.1 Sampling Importance Resampling Filter (SIR)

The SIR particle filter is a sequential Monte Carlo method which can be used for recursive Bayesian filtering required in case of nonlinear state estimation problems. As the SIR PF is a direct execution of the SIS algorithm, therefore all the assumptions stated in case of SIS method are also required for its implementation. Some of the assumptions are stated again:

- The considered system's dynamic model, $\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k)$ and measurement model, $\boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k)$ provided in equation (2-48), is needed to be known.
- It is possible to be able to draw samples from the *prior*, based upon the knowledge of the system $\boldsymbol{f}_k(\cdot)$ and the process noise distribution of $\boldsymbol{w}_k$.
- Furthermore, availability of the likelihood function $p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)$ is required for particle wise evaluation of the importance weights using equation (2-72).

To obtain the SIR filtering algorithm, the following modifications are made in implementing the SIS algorithm.

i. The appropriate choice of the importance density $q(\boldsymbol{x}_k|\boldsymbol{X}_{k-1}^i, \boldsymbol{Z}_k)$ is taken to be prior $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$, as also mentioned in equation (2-71).
ii. The resampling step (described in section 2.6.2.1.2) is to be applied at every time step.

The implication of the above choice of importance density in a need to perform the sampling step using $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$. A sample $\boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$ can be obtained by first generating a process noise sample $\boldsymbol{w}_{k-1}^i \sim p_w(\boldsymbol{w}_{k-1})$ and thereafter propagating the particles as $\boldsymbol{x}_k^i = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1})$; where $p_w(\cdot)$ is the known pdf of process noise $\boldsymbol{w}_k$. As mentioned before, this particular choice of importance density leads to a weight update in the form (already obtained in equation (2-72)),

$$\overline{\mathrm{w}}_k^i \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)\overline{\mathrm{w}}_{k-1}^i. \tag{2-81}$$

However, based upon the choice of applying resampling at every time step, the previous weight will always be $\overline{\mathrm{w}}_{k-1}^i = 1/N_s$. Therefore, the above weight update equation further reduces to,

$$\overline{\mathrm{w}}_k^i \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k^i), \tag{2-82}$$

where all the weights are normalized before the resampling step.

Finally, an estimate for corrected state value can be obtained from the expected value of the marginal posterior distribution. That is,

$$\hat{\boldsymbol{x}}_k^+ = E[p(\boldsymbol{x}_k|\boldsymbol{Z}_k)] = \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \cdot \boldsymbol{x}_k^i \tag{2-83}$$

Even though, SIR filter has the advantage of straightforward implementation due to a simple evaluation of importance weights and easily sampled importance density, this filter is however sensitive to outliers. This is due to a choice of the importance density, which is independent of the measurements $\boldsymbol{z}_k$ and thus makes the state space to be explored without any knowledge of the observations (Arulampalam, 2002). Furthermore, it can also be sometimes inefficient, as it resamples all the particles every time even if not much degeneracy was accounted.

A brief algorithm illustrating the working of SIR filter is presented below. One may note that in the following algorithms, additional superscripts ($-$ and $+$) are used which have the same meaning as described in case of EKF and UKF; that is, $\boldsymbol{x}_k^{i-}$ represents *a priori* particle whereas $\boldsymbol{x}_k^{i+}$ would represent *a posteriori* particle.

1. The L-state continuous time dynamic system is given as,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big) \\ \boldsymbol{z} &= \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\big) \\ \boldsymbol{w}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{Q}(t)\big) \\ \boldsymbol{v}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{R}(t)\big) \end{aligned} \tag{2-84}$$

2. The SIR filter is initialized at $k = 0$ as,

$$\begin{aligned} \widehat{\boldsymbol{x}}_0^+ &= E[\boldsymbol{x}_0] \\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_0^+)^T] \end{aligned} \tag{2-85}$$

   and N_s number of initial samples (or particles) $\{\boldsymbol{x}_0^{\boldsymbol{i}}\}_{i=1}^{N_s}$ are randomly generated around initial state $\widehat{\boldsymbol{x}}_0^+$, using the known initial state pdf $p(\boldsymbol{x}_0)$.

3. The following time update equations are used to estimate the corrected states ($\widehat{\boldsymbol{x}}_k^+$) by propagating the particles and their associated weights from one measurement time to the next, till the end of the sampling time N. That is, for $k \in \{1, \dots, N-1\}$

   a) The time propagation (or *prediction*) step is performed to obtain the *priori* particles $\boldsymbol{x}_k^{i-}$ based on the dynamic system equations and the process noise $\boldsymbol{w}_{k-1}$ generated using the known process noise pdf $p_w(\boldsymbol{w}_{k-1})$.

$$\boldsymbol{x}_k^{i-} = \boldsymbol{x}_{k-1}^{i+} + \int_{t_k}^{t_{k+1}} \boldsymbol{f}\big(\boldsymbol{x}_{k-1}^{i+}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}\big) d\tau\,; \qquad i = 1, \dots, N_s \tag{2-86}$$

   b) The relative likelihood (or importance weights) w_k^i of each particle $\boldsymbol{x}_k^{i-}$, conditioned on the measurement $\boldsymbol{z}_k$ is computed as,

$$\mathrm{w}_k^i = p\big(\boldsymbol{z}_k | \boldsymbol{x}_k^{i-}\big); \qquad i = 1, \dots, N_s \tag{2-87}$$

   c) All the obtained weights are then normalized

$$\bar{\mathrm{w}}_k^i = \frac{\mathrm{w}_k^i}{\sum_{j=1}^{N_s} \mathrm{w}_k^j}; \qquad i = 1, \dots, N_s \tag{2-88}$$

   d) Using the Systematic Resampling algorithm (section 2.6.2.1.2), the new set of *posteriori* particles $\boldsymbol{x}_k^{i+}$ and corresponding weights $\bar{\mathrm{w}}_k^i$ are generated.

e) The corrected value of the state $\hat{\boldsymbol{x}}_k^+$ is obtained as

$$\hat{\boldsymbol{x}}_k^+ = \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \cdot \boldsymbol{x}_k^{i+} \tag{2-89}$$

2.6.3.2 Regularized Particle Filter (RPF)

The Regularized particle filter is one of the artifice suggested to counter the problem of sample impoverishment (described at the end of section 2.6.2.1) which emerged as a consequence of the resampling step, performed to tackle the degeneracy among particles. The problem of sample impoverishment (loss of diversity among particles) was understood to be occurring on account of drawing the samples from a discrete distribution rather than from a continuous one, during the resampling step. And furthermore, lack of proper rectification of the problem may lead to "*particle collapse*" that signifies a severe case of sample impoverishment, where all particles are identically located in the state space and thus provides a poor representation of the posterior density, as also stated in (Arulampalam, 2002).

The RPF is identical in implementation to the SIR filter, except for the resampling step. The SIR filter resamples at every time step from the discrete approximation of the posterior density,

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^i\right), \tag{2-90}$$

whereas, the RPF resamples from a continuous approximation of the posterior density in the form,

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i K_h\left(\boldsymbol{x}_k - \boldsymbol{x}_k^i\right), \tag{2-91}$$

based upon the criteria of effective sample size ($\widehat{N}_{eff} < N_{th}$), as mentioned before. In the above equation,

$$K_h(\boldsymbol{x}) = \frac{1}{h^L} K\left(\frac{\boldsymbol{x}}{h}\right), \tag{2-92}$$

is the rescaled Kernel density $K(\cdot)$, h is the positive scalar Kernel bandwidth, L is the dimension of the state vector $\boldsymbol{x}_k$ and $\overline{\mathrm{w}}_k^i$ are the same normalized weights, as used previously.

The Kernel density is a symmetric pdf such that

$$\int \boldsymbol{x} K(\boldsymbol{x}) d\boldsymbol{x} = 0, \qquad \int \|\boldsymbol{x}\|^2 K(\boldsymbol{x}) d\boldsymbol{x} < \infty. \tag{2-93}$$

The Kernel $K(\cdot)$ and bandwidth h are chosen to minimize the *Mean Integrated Square Error* (MISE) between the assumed true density $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ and the corresponding regularized empirical representation of equation (2-91), which is defined as (Arulampalam, 2002)

$$MISE(\hat{p}) = E\left[\int \{\hat{p}(\boldsymbol{x}_k|\boldsymbol{Z}_k) - p(\boldsymbol{x}_k|\boldsymbol{Z}_k)\}^2 d\boldsymbol{x}_k\right], \tag{2-94}$$

where, $\hat{p}(\cdot \mid \cdot)$ denotes the approximate continuous density given in equation (2-91). In the special case of equal weights ($\overline{\mathrm{w}}_k^i = \frac{1}{N_s}$ for $i = 1, \dots, N_s$), the optimal choice of the Kernel is Epanechnikov Kernel (Simon, 2006)

$$K(\boldsymbol{x}) = \begin{cases} \frac{L+2}{2v_L}(1 - \|\boldsymbol{x}\|^2), & if\ \|\boldsymbol{x}\| < 1 \\ 0, & otherwise \end{cases} \quad \textbf{(2-95)}$$

where, v_L is the volume of the L-dimensional unit hypersphere, which is given by $v_L = 2\pi v_{L-2}/L$ (for further reading one may refer (Simon, 2006)). Also, if the underlying posterior density is Gaussian with an identity covariance matrix, then the optimal bandwidth is given as (Simon, 2006)

$$h^* = \left[8v_L^{-1}(L+4)\left(2\sqrt{\pi}\right)^L\right]^{1/(L+4)} N_s^{-1/(L+4)}, \quad \textbf{(2-96)}$$

Although these choices of Kernel and bandwidth are optimal only for the considered special case (i.e. equal weights and a Gaussian pdf), but still they are often used in general cases to obtain a suboptimal filter (Arulampalam, 2002).

Since it was explicitly assumed that the true posterior density $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ has a unit covariance matrix, numerical computation of the covariance $\boldsymbol{S}_k$ of each particle $\boldsymbol{x}_k^i$ at every time step is required. Thereafter its matrix square root $\boldsymbol{A}_k$ (such that, $\boldsymbol{A}_k\boldsymbol{A}_k^T = \boldsymbol{S}_k$) is evaluated using the Cholesky decomposition, to later compute the Kernel as

$$K_h(\boldsymbol{x}) = \left(det(\boldsymbol{A}_k)\right)^{-1} h^{-L} K\left(\boldsymbol{A}_k^{-1}\frac{\boldsymbol{x}}{h}\right). \quad \textbf{(2-97)}$$

Eventually the RPF has a theoretical disadvantage that the samples are no longer assured to be asymptotically approximating the posterior density. However, in case of serious sample impoverishment (for instance, if system has small process noise), the RPF shows improved performance in state estimation compared to the SIR filter.

The RPF is summarized as

1. The L-state continuous time dynamic system is given as,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\right) \\ \boldsymbol{z} &= \boldsymbol{h}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\right) \\ \boldsymbol{w}(t) &\sim \left(\boldsymbol{0}, \boldsymbol{Q}(t)\right) \\ \boldsymbol{v}(t) &\sim \left(\boldsymbol{0}, \boldsymbol{R}(t)\right) \end{aligned} \quad \textbf{(2-98)}$$

2. The filter is initialized at $k = 0$ as,

$$\begin{aligned} \hat{\boldsymbol{x}}_0^+ &= E[\boldsymbol{x}_0] \\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T] \end{aligned} \quad \textbf{(2-99)}$$

and N_s number of initial samples (or particles) $\{\boldsymbol{x}_0^i\}_{i=1}^{N_s}$ are randomly generated around initial state $\hat{\boldsymbol{x}}_0^+$, using the known initial state pdf $p(\boldsymbol{x}_0)$.

3. The following time update equations are used to propagate the particles and their associated weights from one measurement time to the next, till the end of the sampling time N. That is, for $k \in \{1, \dots, N-1\}$

   a) The time propagation (or *prediction*) step is performed to obtain the *priori* particles $\boldsymbol{x}_k^{i-}$ based on the dynamic system equations and the process noise $\boldsymbol{w}_{k-1}$ generated using the known process noise pdf $p_w(\boldsymbol{w}_{k-1})$.

$$\boldsymbol{x}_k^{i-} = \boldsymbol{x}_{k-1}^{i+} + \int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{x}_{k-1}^{i+}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}\right) d\tau\,; \qquad i = 1, \dots, N_s \tag{2-100}$$

   b) The relative likelihood (or importance weights) w_k^i of each particle $\boldsymbol{x}_k^{i-}$, conditioned on the measurement $\boldsymbol{z}_k$ is computed as,

$$\mathrm{w}_k^i = p\left(\boldsymbol{z}_k | \boldsymbol{x}_k^{i-}\right); \qquad i = 1, \dots, N_s \tag{2-101}$$

   c) All the obtained weights are then normalized

$$\overline{\mathrm{w}}_k^i = \frac{\mathrm{w}_k^i}{\sum_{j=1}^{N_s} \mathrm{w}_k^j}\,; \qquad i = 1, \dots, N_s \tag{2-102}$$

Now, $\widehat{N}_{eff}$ is calculated based on

$$\widehat{N}_{eff} \approx \frac{1}{\sum_{i=1}^{N_s} \left(\overline{\mathrm{w}}_k^i\right)^2} \tag{2-103}$$

and if $\widehat{N}_{eff} < N_{th}$ (for example, to resample at 50% reduction, $N_{th} = 0.5 * N_s$) then the resampling step (step 4) is executed otherwise step 5 is directly followed.

4. The following steps are performed to have Regularized resampling.
   a) The ensemble mean μ and the covariance matrix $\boldsymbol{S}_k$ is computed as

$$\boldsymbol{\mu} = \frac{1}{N_s} \sum_{i=1}^{N_s} \boldsymbol{x}_k^{i-}$$

$$\boldsymbol{S}_k = \frac{1}{N_s - 1} \sum_{i=1}^{N_s} \left(\boldsymbol{x}_k^{i-} - \boldsymbol{\mu}\right)\left(\boldsymbol{x}_k^{i-} - \boldsymbol{\mu}\right)^T \tag{2-104}$$

   b) A matrix square root $\boldsymbol{A}_k$ of $\boldsymbol{S}_k$ is obtained using the Cholesky factorization, such that, $\boldsymbol{A}_k \boldsymbol{A}_k^T = \boldsymbol{S}_k$.
   c) The volume of L-dimensional unit hypersphere is computed as $v_L = 2\pi v_{L-2}/L$, starting the recursion with value $v_1 = 2$, $v_2 = \pi$ and $v_3 = 4\pi/3$.
   d) The optimal Kernel bandwidth h is obtained as

$$h = \frac{1}{2} \cdot \left[8 v_L^{-1} (L+4) \left(2\sqrt{\pi}\right)^L \right]^{1/(L+4)} N_s^{-1/(L+4)} \quad \textbf{(2-105)}$$

where, a factor $1/2$ is multiplied on the right hand side of equation (2-96), in order to handle multimodal pdf's (a pdf with more than one local maxima).

e) Finally a continuous form of the posterior density $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ is approximated as

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{Z}_k) \approx \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i K_h\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{\boldsymbol{i}}\right), \quad \textbf{(2-106)}$$

where, the Kernel $K_h(\boldsymbol{x})$ and the Epanechnikov Kernel $K(\boldsymbol{x})$ are the same as provided in equation (2-97) and equation (2-95) respectively.

f) Now, as the continuous pdf approximation $\hat{p}(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ is available, the *posteriori* particles can be obtained by first performing the Systematic Resampling (section 2.6.2.1.2) on $\hat{p}(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ to obtain the new set of particles ($\boldsymbol{x}_k^{i*}$) and then using the following equation,

$$\boldsymbol{x}_k^{i+} = \boldsymbol{x}_k^{i-} + h\boldsymbol{A}_k \boldsymbol{x}_k^{i*}; \qquad i = 1, \dots, N_s \quad \textbf{(2-107)}$$

Also, setting $\overline{\mathrm{w}}_k^i = N_s^{-1}$ for $i = 1, \dots, N_s$.

5. The corrected value of the state $\hat{\boldsymbol{x}}_k^+$ is obtained as

$$\hat{\boldsymbol{x}}_k^+ = \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \cdot \boldsymbol{x}_k^{i+} \quad \textbf{(2-108)}$$

2.6.3.3 Auxiliary Particle Filter (APF)

The APF was introduced by (M. Pitt, 1999), as a variant of standard SIR filter to even out the probability of *priori* particles, thus further enhancing the diversity among *posteriori* particles. "The idea of this filter is to increase the influence of particles with a large future likelihood, assuming that the estimate is not needed until after the next measurement" (Karlsson, 2005). In other words, it is a technique of selecting most suitable parent particles, (particles at previous time) that have maximum likelihood with the current measurement, using a "pre-sampling step"; before propagating them forward in time. This is performed by augmenting each particle with an *auxiliary variable* (corresponding to particle index), such that the particle's origin is traced simultaneous to the evaluation of its compatibility with the coming observation. To obtain the APF from SIS framework, an importance density of the form $q(\boldsymbol{x}_k, i|\boldsymbol{Z}_k)$ is introduced, that samples the pair $\left\{\boldsymbol{x}_k^j, i^j\right\}_{j=1}^{N_s}$, where i^j refers to the parent particle's index (i.e. at time $k-1$).

Considering the joint density of i^{th} particle, the following proportionality can be deduced by applying Bayes' rule,

$$\begin{aligned} p(\boldsymbol{x}_k, i|\boldsymbol{Z}_k) &\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k, i|\boldsymbol{Z}_{k-1}) \\ &= p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|i, \boldsymbol{Z}_{k-1}) p(i|\boldsymbol{Z}_{k-1}) \\ &= p(\boldsymbol{z}_k|\boldsymbol{x}_k) p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i\right) \overline{\mathrm{w}}_{k-1}^i. \end{aligned} \quad \textbf{(2-109)}$$

The APF works by drawing samples from the above joint distribution $p(\boldsymbol{x}_k, i|\boldsymbol{Z}_k)$ and thereafter discarding the index i in the pair $(\boldsymbol{x}_k, i)$ to generate samples $\{\boldsymbol{x}_k^j\}_{j=1}^{N_s}$ from the marginalized density $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$ (Arulampalam, 2002). Also, the importance density used to generate samples $\{\boldsymbol{x}_k^j, i^j\}_{j=1}^{N_s}$ is defined such as, to satisfy the following proportionality

$$q(\boldsymbol{x}_k, i|\boldsymbol{Z}_k) \propto p(\boldsymbol{z}_k|\boldsymbol{\mu}_k^i)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)\overline{\mathrm{w}}_{k-1}^i, \tag{2-110}$$

where $\boldsymbol{\mu}_k^i$ is some characterization of $\boldsymbol{x}_k$ subject to $\boldsymbol{x}_{k-1}^i$and can be defined as:

a Mean,

$$\boldsymbol{\mu}_k^i \sim E[\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i], \tag{2-111}$$

or a Sample,

$$\boldsymbol{\mu}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i). \tag{2-112}$$

Again applying Bayes' rule, the importance density can also be written as

$$q(\boldsymbol{x}_k, i|\boldsymbol{Z}_k) = q(i|\boldsymbol{Z}_k)q(\boldsymbol{x}_k|i, \boldsymbol{Z}_k). \tag{2-113}$$

Now, by defining

$$q(\boldsymbol{x}_k|i.\boldsymbol{Z}_k) \triangleq p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i) \tag{2-114}$$

and substituting equation (2-113) back in equation (2-110), the following expression can be obtained

$$q(i|\boldsymbol{Z}_k) \propto p(\boldsymbol{z}_k|\boldsymbol{\mu}_k^i)\overline{\mathrm{w}}_{k-1}^i. \tag{2-115}$$

Using the importance weight definition of equation (2-64), weight of the sample $\{\boldsymbol{x}_k^j, i^j\}_{j=1}^{N_s}$ is assigned as

$$\begin{aligned} \overline{\mathrm{w}}_k^j &\propto \frac{p(\boldsymbol{x}_k^j, i^j|\boldsymbol{Z}_k)}{q(\boldsymbol{x}_k^j, i^j|\boldsymbol{Z}_k)} \\ \overline{\mathrm{w}}_k^j &\propto \frac{\overline{\mathrm{w}}_{k-1}^{i^j} p(\boldsymbol{z}_k|\boldsymbol{x}_k^j) p\left(\boldsymbol{x}_k^j|\boldsymbol{x}_{k-1}^{i^j}\right)}{q(i^j|\boldsymbol{Z}_k)q(\boldsymbol{x}_k^j|i^j, \boldsymbol{Z}_k)}, \end{aligned} \tag{2-116}$$

also, by using equations (2-114) and (2-115), a final weight expression can be obtained

$$\overline{\mathrm{w}}_k^j \propto \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^j)}{p\left(\boldsymbol{z}_k\middle|\boldsymbol{\mu}_k^{i^j}\right)}. \tag{2-117}$$

An advantage of APF is, that as it uses current measurement to pre-select the particles, the estimated state is most likely to be close to the true state, compared to the state estimate produced by a normal SIR filter. APF can also be regarded as a pre-sampling (resampling at previous time step) filter based on some points estimates $\boldsymbol{\mu}_k^i$ that characterizes $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$. The APF has been stated (Arulampalam, 2002) to be less sensitive to outliers in case of small

process noise, as then $\boldsymbol{\mu}_k^i$ characterizes the pdf $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$ well and the weights $\bar{\mathrm{w}}_k^i$ are more even. However, for the large process noise, the previous assumption of deterministic process model breaks down and wrong parent particles can get selected (as now a single particle would poorly characterize the pdf $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$). And thus in this situation, the APF would result into a degraded performance over the SIR filter.

The APF is summarized as

1. The L-state continuous time dynamic system is given as,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big)\\ \boldsymbol{z} &= \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\big)\\ \boldsymbol{w}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{Q}(t)\big)\\ \boldsymbol{v}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{R}(t)\big)\end{aligned} \quad \textbf{(2-118)}$$

2. The filter is initialized at $k = 0$ as,

$$\begin{aligned}\hat{\boldsymbol{x}}_0^+ &= E[\boldsymbol{x}_0]\\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T]\end{aligned} \quad \textbf{(2-119)}$$

   and N_s number of initial samples (or particles) $\{\boldsymbol{x}_0^i\}_{i=1}^{N_s}$ are randomly generated around initial state $\hat{\boldsymbol{x}}_0^+$, using the known initial state pdf $p(\boldsymbol{x}_0)$.

3. The following time update equations are used to propagate the particles and their associated weights from one measurement time to the next, till the end of the sampling time N. That is, for $k \in \{1, \dots, N-1\}$
   a) The characteristic variable $\boldsymbol{\mu}_k^i$ is first calculated based on the dynamic system equations and the zero process noise. That is,

$$\boldsymbol{\mu}_k^i = \boldsymbol{x}_{k-1}^{i+} + \int_{t_k}^{t_{k+1}} \boldsymbol{f}\big(\boldsymbol{x}_{k-1}^{i+}, \boldsymbol{u}_{k-1}, \boldsymbol{0}\big) d\tau\,; \qquad i = 1, \dots, N_s \quad \textbf{(2-120)}$$

   b) The parent weights are computed as

$$\mathrm{w}_k^i = p\big(\boldsymbol{z}_k|\boldsymbol{\mu}_k^i\big)\bar{\mathrm{w}}_{k-1}^i; \qquad i = 1, \dots, N_s \quad \textbf{(2-121)}$$

   c) All the obtained weights are normalized

$$\bar{\mathrm{w}}_k^i = \frac{\mathrm{w}_k^i}{\sum_{j=1}^{N_s} \mathrm{w}_k^j}; \qquad i = 1, \dots, N_s \quad \textbf{(2-122)}$$

   d) Then the Systematic Resampling algorithm is applied over the above weights to obtain the indices of the parent particles (with maximum probabilities). The function call (according to the resampling algorithm of section 2.6.2.1.2) is made as
   $\left[\{\sim, \sim, i^j\}_{j=1}^{N_s}\right] = \mathrm{RESAMPLE}\left[\{\boldsymbol{\mu}_k^i, \bar{\mathrm{w}}_k^i\}_{i=1}^{N_s}\right]$.

e) Now, based on the parent indices, the time propagation (or *prediction*) step is performed to obtain the *priori* particles $\boldsymbol{x}_k^{j-}$

$$\boldsymbol{x}_k^{j-} = \boldsymbol{x}_{k-1}^{i^j+} + \int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\boldsymbol{x}_{k-1}^{i^j+}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}\right) d\tau\,; \qquad j = 1, \dots, N_s \tag{2-123}$$

f) Also, their assigned weights are evaluated

$$\bar{\mathrm{w}}_k^j = \frac{p\left(\boldsymbol{z}_k | \boldsymbol{x}_k^{j-}\right)}{p\left(\boldsymbol{z}_k \middle| \boldsymbol{\mu}_k^{i^j}\right)}; \qquad j = 1, \dots, N_s \tag{2-124}$$

where, $\bar{\mathrm{w}}_k^j$ are the normalized weights obtained in the same way as in equation (2-122).

4. (Optional)
   $\widehat{N}_{eff}$ is calculated based on equation (2-77) and
   - If $\widehat{N}_{eff} < N_{th}$ (for example, $N_{th} = 0.5 * N_s$) then the same Systematic resampling step is executed
     $\left[\left\{\boldsymbol{x}_k^{j+}, \bar{\mathrm{w}}_k^j, \sim\right\}_{j=1}^{N_s}\right] = \mathrm{RESAMPLE}\left[\left\{\boldsymbol{x}_k^{j-}, \bar{\mathrm{w}}_k^j\right\}_{j=1}^{N_s}\right]$
   - Otherwise,
     $\left\{\boldsymbol{x}_k^{j+}, \bar{\mathrm{w}}_k^j\right\}_{j=1}^{N_s} = \left\{\boldsymbol{x}_k^{j-}, \bar{\mathrm{w}}_k^j\right\}_{j=1}^{N_s}$

5. The corrected value of the state $\widehat{\boldsymbol{x}}_k^+$ is obtained as

$$\widehat{\boldsymbol{x}}_k^+ = \sum_{j=1}^{N_s} \bar{\mathrm{w}}_k^j \cdot \boldsymbol{x}_k^{j+} \tag{2-125}$$

2.6.4 Combined form of Particle filters

All the forms of particle filters discussed in the previous section are obtained by the reformation of systematic importance sampling, i.e. either by varying the choice of importance density or by modifying resampling algorithms. Another approach that has been proposed to enhance the quality of particle filtering, is to use it in combination with other filters like the EKF or the UKF. According to this approach, N_s number of Kalman filters (one for each particle) run together to propagate and subsequently correct all the particles at each measurement time (time at which measurement is available) and thereafter, resampling is performed using measurement. Although, these combined type of PFs consume higher computational effort, but however they often show a superior performance over a simple PF (for instance SIR), for the systems with severe nonlinearities or containing higher process noise.

Only the algorithm incorporating the combination of EKF with PF, commonly known as *Extended Particle Filter* (EPF), is mentioned below; however, the algorithm for combined UKF and PF, *Unscented Particle Filter* (UPF), can also be written in a similar manner and can be referred in (Hayking, 2001).

Algorithm for Extended PF (EPF)

1. The L-state continuous time dynamic system is given as,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big)\\ \boldsymbol{z} &= \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\big)\\ \boldsymbol{w}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{Q}(t)\big)\\ \boldsymbol{v}(t) &\sim \big(\boldsymbol{0}, \boldsymbol{R}(t)\big)\end{aligned} \quad \textbf{(2-126)}$$

2. The filter is initialized at $k = 0$ as,

$$\begin{aligned}\hat{\boldsymbol{x}}_0^+ &= E[\boldsymbol{x}_0]\\ \boldsymbol{P}_0^+ &= E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T]\end{aligned} \quad \textbf{(2-127)}$$

and N_s number of initial samples (or particles) $\{\boldsymbol{x}_0^i\}_{i=1}^{N_s}$ are randomly generated around initial state $\hat{\boldsymbol{x}}_0^+$ using the known initial state pdf $p(\boldsymbol{x}_0)$. Also, the initial covariance matrices of these particles are made equal to the initial state covariance. i.e.

$$\boldsymbol{P}_0^{i+} = \boldsymbol{P}_0^+; \quad i = 1, \dots, N_s \quad \textbf{(2-128)}$$

3. The following time update equations are used to estimate the corrected states ($\hat{\boldsymbol{x}}_k^+$) by propagating the particles using EKF equations from one measurement time to the next, till the end of the sampling time N. That is, for $k \in \{1, \dots, N-1\}$

   a) The time propagation (or *prediction*) step is performed to obtain the *priori* particles $\boldsymbol{x}_k^{i-}$ and their respective covariance matrices.

$$\begin{aligned}\boldsymbol{x}_k^{i-} &= \boldsymbol{x}_{k-1}^{i+} + \int_{t_{k-1}}^{t_k} \boldsymbol{f}\big(\boldsymbol{x}_{k-1}^{i+}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}\big) d\tau; \qquad i = 1, \dots, N_s\\ \boldsymbol{P}_k^{i-} &= \boldsymbol{\Phi}_{k-1}^i \boldsymbol{P}_{k-1}^{i+} \boldsymbol{\Phi}_{k-1}^{i^T} + \boldsymbol{\Psi}_{k-1}^i \boldsymbol{F}_{k-1}^i \boldsymbol{Q}_{k-1} \boldsymbol{F}_{k-1}^{i^T} \boldsymbol{\Psi}_{k-1}^{i^T} \qquad i = 1, \dots, N_s\end{aligned} \quad \textbf{(2-129)}$$

   where,

$$\boldsymbol{F}_{k-1}^i = \left.\frac{\partial \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\big)}{\partial \boldsymbol{w}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{k-1}^{i+},\ \boldsymbol{u}=\boldsymbol{u}_{k-1}}, \quad \textbf{(2-130)}$$

   and matrices $\boldsymbol{\Phi}_{k-1}^i$ and $\boldsymbol{\Psi}_{k-1}^i$ are defined similar to equations (2-4) and (2-5), respectively. Also, each noise vector $\boldsymbol{w}_{k-1}$ is randomly generated using $p_w(\boldsymbol{w}_{k-1})$.

   b) The *priori* particles and covariance matrices are corrected to obtain *posteriori* particles and covariance matrices.

$$\begin{aligned}\boldsymbol{K}_k^i &= \boldsymbol{P}_k^{i-} \boldsymbol{C}_k^{i^T} \left(\boldsymbol{C}_k^i \boldsymbol{P}_k^{i-} \boldsymbol{C}_k^{i^T} + \boldsymbol{R}_k\right)^{-1}\\ \boldsymbol{x}_k^{i+} &= \boldsymbol{x}_k^{i-} + \boldsymbol{K}_k^i \left(\boldsymbol{z}_k - \boldsymbol{h}\big(\boldsymbol{x}_k^{i-}, \boldsymbol{u}_k, \boldsymbol{0}\big)\right)\\ \boldsymbol{P}_k^{i+} &= \big(\boldsymbol{I} - \boldsymbol{K}_k^i \boldsymbol{C}_k^i\big) \boldsymbol{P}_k^{i-},\end{aligned} \quad \textbf{(2-131)}$$

where,

$$\boldsymbol{C}_k^i = \left.\frac{\partial \boldsymbol{h}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_k^{i-},\ \boldsymbol{u}=\boldsymbol{u}_k}. \quad \textbf{(2-132)}$$

c) The relative likelihood w_k^i of each particle $\boldsymbol{x}_k^{i+}$ is computed as,

$$\mathrm{w}_k^i = p\left(\boldsymbol{z}_k | \boldsymbol{x}_k^{i+}\right); \qquad i = 1, \ldots, N_s \quad \textbf{(2-133)}$$

d) All the obtained weights are then normalized

$$\overline{\mathrm{w}}_k^i = \frac{\mathrm{w}_k^i}{\sum_{j=1}^{N_s} \mathrm{w}_k^j}; \qquad i = 1, \ldots, N_s \quad \textbf{(2-134)}$$

e) Using the Systematic Resampling algorithm (section 2.6.2.1.2), the new set of *posteriori* particles $\boldsymbol{x}_k^{i+}$ and corresponding weights $\overline{\mathrm{w}}_k^i$ are generated along with their respective covariance matrices $\boldsymbol{P}_k^{i+}$.

f) The corrected value of the state $\hat{\boldsymbol{x}}_k^+$ is obtained as

$$\hat{\boldsymbol{x}}_k^+ = \sum_{i=1}^{N_s} \overline{\mathrm{w}}_k^i \cdot \boldsymbol{x}_k^{i+} \quad \textbf{(2-135)}$$

3 Application

In the current chapter, two example problems are discussed to illustrate separate scenarios. The first one is a tracking problem of a Re-entry vehicle (also taken in (Mehndiratta, 2014)) for which, the performance of previously discussed filters (EKF, UKF and PF), in estimating the states of the re-entry vehicle, is compared to one another. Various parameters used for comparison are

a) Computational Time.
b) Lower sampling rates (Linearization of the system over larger time step).
c) Poor initial guess.

Subsequently in section 3.2, the state estimation is performed over an algebraic system using the EKF and various forms of particle filters, described in section 2.6.3. This was done to depict the effects of increasing complexities, which arise while efficiently dealing with the problems of degeneracy and sample impoverishment. The analysis was based on the computational time consumed to deliver a certain order of accuracy. The parameter used to check accuracy for comparison is the 'mean-square error' and is defined as

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{e}_i^2, \tag{3-1}$$

where, $\boldsymbol{e}_i$ is the estimation error (difference between the true states and the estimated states) in the i^{th} sample and N represents the total number of samples.

The programming was done using MathWorks software Matlab based on *object oriented approach*. Also, to achieve better accuracy in computing derivatives (Jacobian matrices), symbolic manipulations were favored, but due to extremely slow nature of symbolic calculations in Matlab, another platform of the same software MuPAD was used. The detailed software documentation is omitted here, but can be seen in (Mehndiratta, 2014). Based upon the programming approach mentioned in (Mehndiratta, 2014), a third sub-class named *PAR_F* of *PROBLEM* class was created, similar to *EKF* and *UKF* sub-classes, with similar attributes and methods. Also, one may note that for both the examples, prior distribution was regarded as the importance density in implementing all forms of particle filers.

3.1 Re-entry Vehicle Example

In the current section, an example problem regarding tracking of a Re-entry vehicle is described. The idea is to evaluate the problem using a PF also, as it had been extensively analyzed in past using EKF and UKF only. The particle filter used was the combined type EPF (Extended particle filter), due to implementation issues observed with the other types.

3.1.1 Problem description

The aim is to estimate the states of a body as it re-enters the earth's atmosphere, at a very high altitude and with high velocity. The state vector is composed of position $x_1(t)$, velocity $x_2(t)$ and constant ballistic coefficient $x_3(t)$. Furthermore, its motion is determined by altitude and velocity dependent drag terms, and it is constrained to fall vertically.

The continuous time dynamics of the system is given as

$$\dot{x}_1 = -x_2(t) + w_1(t)$$
$$\dot{x}_2 = -e_1^{\gamma x_1(t)} x_2(t)^2 x_3(t) + w_2(t) \quad \textbf{(3-2)}$$
$$\dot{x}_3 = w_3(t),$$

where, γ is a constant that relates the air density with altitude. The data for the real states was generated by initializing the state vector as,

$$\boldsymbol{x}(0) = [3 \times 10^5 \;\; 2 \times 10^4 \;\; 10^{-3}] \quad \textbf{(3-3)}$$

and integrating the above equations over time with $\Delta t = 1/64\ sec$.

The Measurement equation is

$$z_1 = \sqrt{(M^2 + [x_1(t) - a]^2)} + v_1(t) \quad \textbf{(3-4)}$$

where, M represents the Radar height and a signifies the horizontal range between the body and radar. All the constants were chosen as

$$\gamma = 5 \times 10^{-5}; \quad M = 100{,}000\ ft; \quad a = 100{,}000\ ft \quad \textbf{(3-5)}$$

Also, $\boldsymbol{Q}$ and R are the process noise covariance matrix and measurement noise covariance matrix respectively and are defined as,

$$\boldsymbol{w} \approx (\boldsymbol{0}, \boldsymbol{Q})$$
$$v \approx (0, R) \quad \textbf{(3-6)}$$
$$E[\boldsymbol{w}v^T] = \boldsymbol{0}$$

3.1.2 Results

The plots were obtained after setting up the filter in a similar manner as done in (Mehndiratta, 2014). For convenience, the initial state vector (*xi_uncorr_0*), initial covariance matrix (*Pi_uncorr_0*), process noise covariance matrix (*Q_PSD*) and measurement noise covariance (*R*) are explicitly mentioned below. Also for Extended particle filter, $N_s = 1000$ (number of particles) was used along with the utilization of Systematic resampling algorithm of section 2.6.2.1.2.

$$xi\_uncorr\_0 = [3 \times 10^5 \;\; -2 \times 10^4 \;\; 10^{-3}]^T \quad \textbf{(3-7)}$$

$$Pi\_uncorr\_0 = \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 4 \times 10^6 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \textbf{(3-8)}$$

$$Q\_PSD = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \textbf{(3-9)}$$

$$R = 10000\ ft^2 \quad \textbf{(3-10)}$$

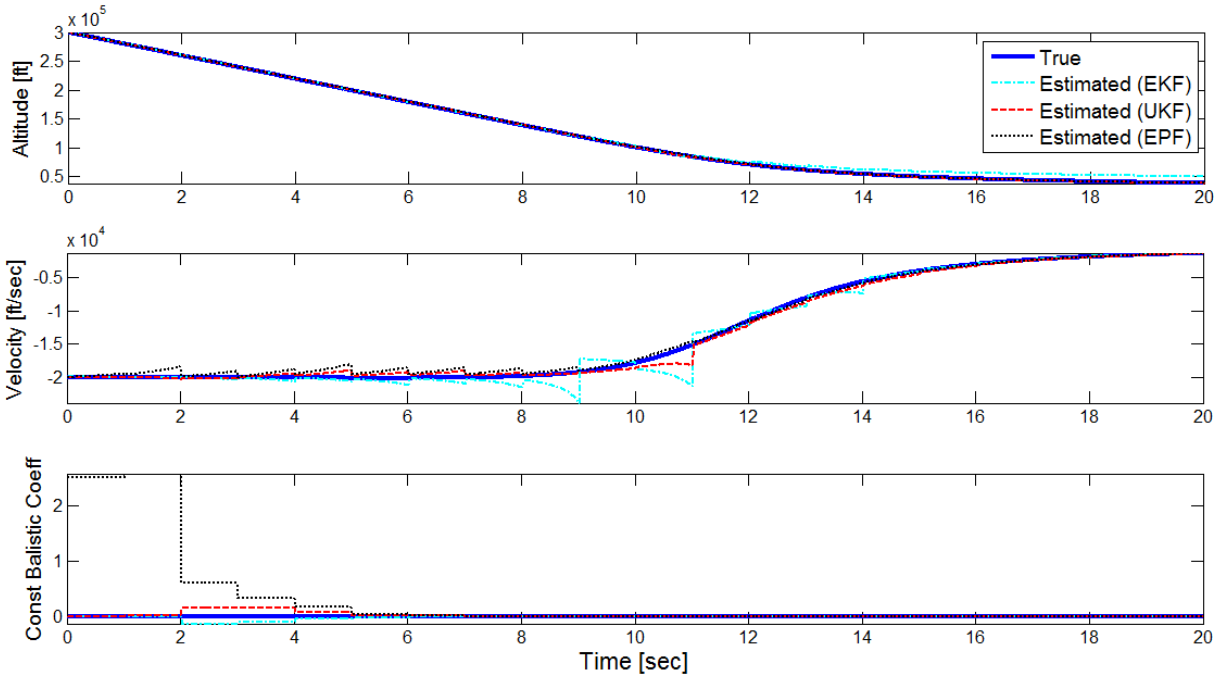

Figure 3-1: Estimated States

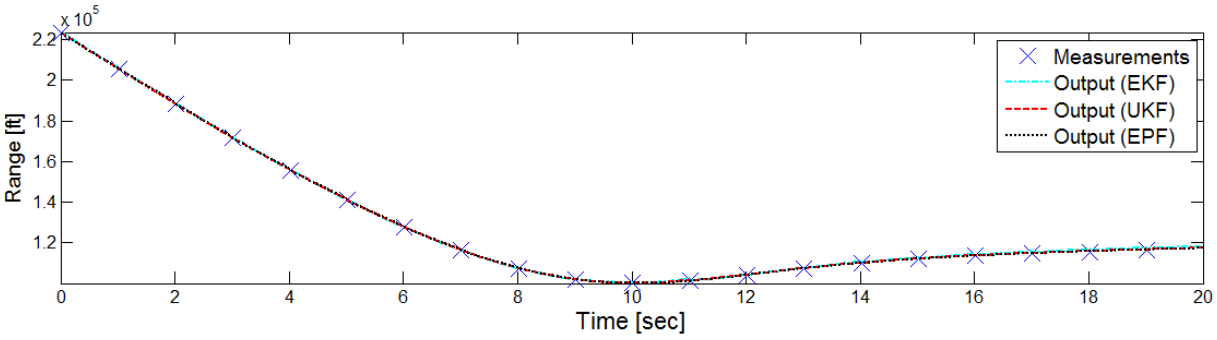

Figure 3-2: Reconstructed Output

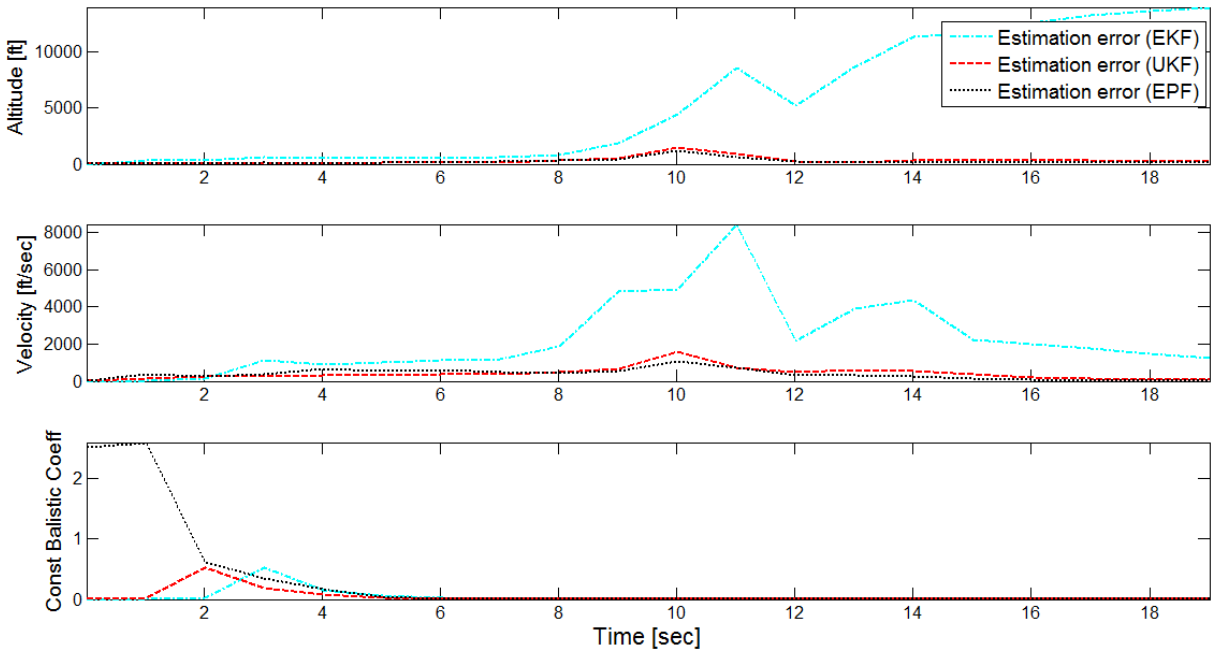

Figure 3-3: Absolute estimation error

For a through comparison, the results were obtained across a Monte Carlo simulation consisting of 10 runs. The corresponding estimated states and reconstructed output are displayed in Figure 3-1 and Figure 3-2, along with their respective true states and measurements. Thereafter, the absolute value of the estimation errors for EKF, UKF and EPF are shown in Figure 3-3.

As seen in Figure 3-3, the absolute estimation error for EPF is almost of the same order as that of UKF while, the error for EKF is $2-3$ orders higher than that for the other two. This observation justifies that the combined form of PF (i.e. EPF) shows better performance than the individual filters as for this particular example, the simple PF was producing highly erroneous results (not explicitly presented).

3.1.2.1 Computational Time

The results were obtained using the original Matlab scripts (*'m-files'*). Current section presents the computational time (in terms of the CPU time) used by all filters along with their mean-square estimation error (MSE), which are listed in Table 3-1 and Table 3-2, respectively.

EKF	UKF	EPF
0.559	1.371	368.928

Table 3-1: Computational time [sec]

	EKF	UKF	EPF
Altitude $[ft]$	$4.5926e+08$	$1.0133e+06$	$5.6107e+05$
Velocity $[ft/sec]$	$0.1293e+08$	$1.4114e+06$	$7.4168e+05$
Const. ballistic coefficient	0.0217	0.0230	0.6887

Table 3-2: Mean-square estimation error (MSE)

It is illustrated from Table 3-1, that the time taken by EKF and UKF for the present application is nearly same. The reason for this is the usage of specialized UKF algorithm (that assumes the additive nature of the uncertainties), which is a lot faster than the generalized UKF algorithm and also in case of EKF, the Jacobians were evaluated analytically using MuPAD, instead of the numerical computation (finite difference technique) which is believed to be slower than the former. On the contrary, EPF utilizes roughly 370 times the computational time as taken by the other two. This was expected as for the Extended PF, N_s number of EKFs run together to propagate (and correct) all N_s particles at each simulation time and for the considered case, large value of N_s was taken to obtain better estimation. To evaluate performance of EPF with varying sample size (N_s), number of particles are plotted with simulation time and corresponding MSEs in Figure 3-4 and Figure 3-5 respectively. Also, one may note that if the compiled *'mex-file'* is used for estimation by EPF, the run time was reduced to $12.1081\ secs$.

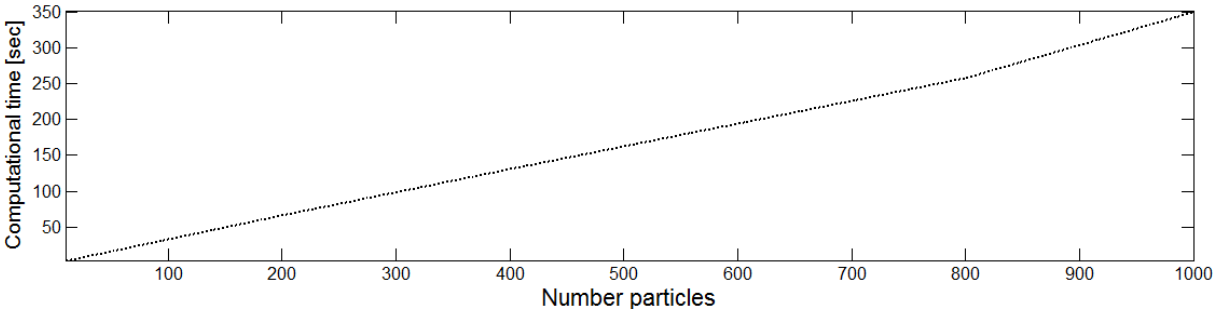

Figure 3-4: Average Computational time with number of particles

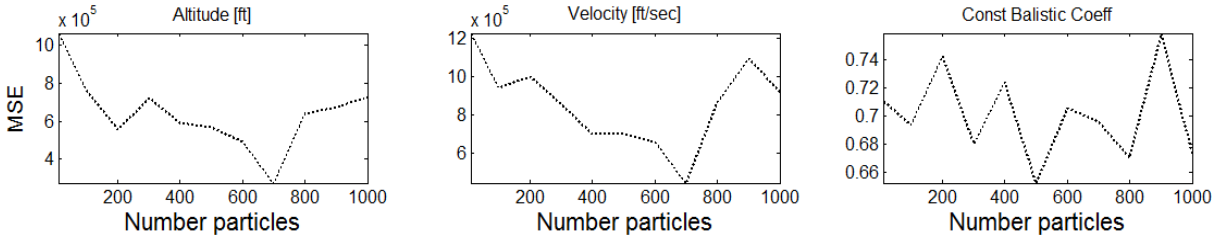

Figure 3-5: Average MSE with number of particles

3.1.2.2 Lower sampling rates

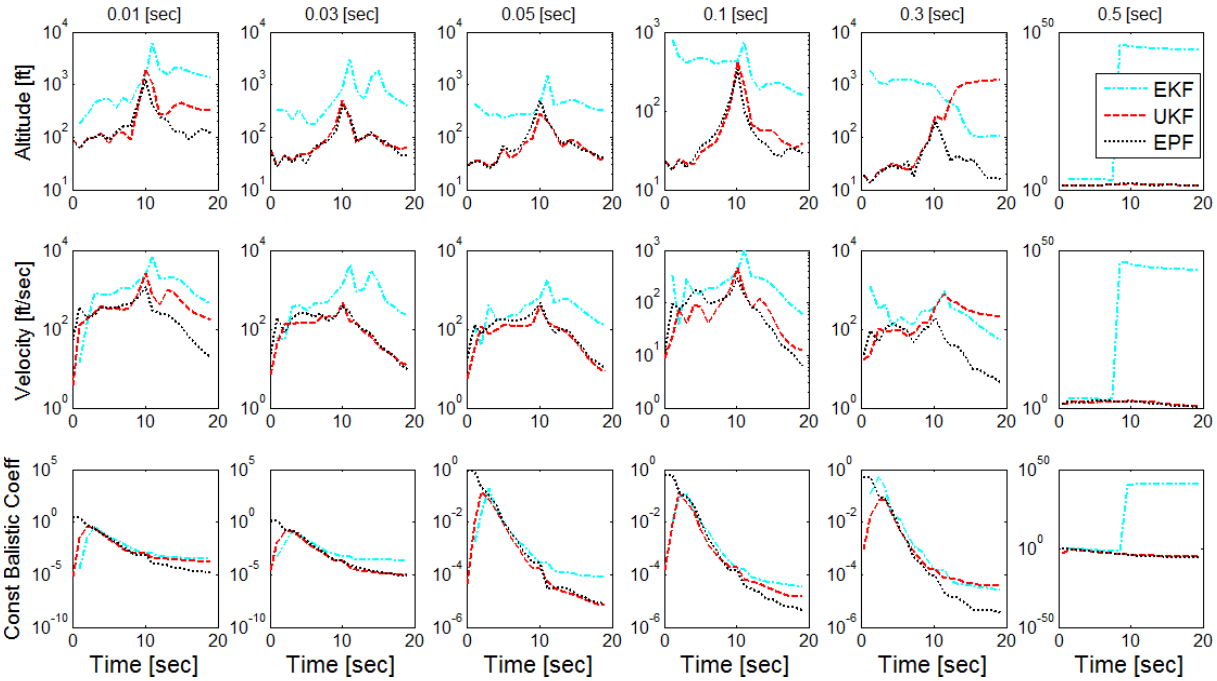

Figure 3-6: Average Estimation error [log] for various time steps

Next, the linearization robustness of EKF, UKF and EPF was explored by integrating the system's differential equations over various time steps. The corresponding estimated states and reconstructed output were obtained along with the absolute estimation errors, which were averaged over 10 Monte Carlo simulations and are displayed in Figure 3-6. In the mentioned figure, the title over each column, indicates the linearization time step, used for the corresponding estimation process.

It is evident from Figure 3-6 that the estimation accuracy of EKF degraded substantially with increasing linearization time. This is on account of breakdown of the linearization assumptions at larger time steps. Although, UKF estimated the states as accurately as EPF for smaller time steps, its estimation accuracy also started degrading after $0.1\ sec$. The reason for this behavior is that, at larger time steps Gaussian assumptions for pdf approximation doesn't hold good and the posterior pdf $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$, can no longer be represented by Gaussian distribution. Therefore, a Gaussian filter (UKF and EKF) which approximates posterior density based on Gaussian approximations, yields poor results for larger linearization times, over a Bayesian estimator (EPF).

Furthermore, the mean-square errors averaged over 10 Monte Carlo simulations were also obtained for all the states, by each filter and are shown in Figure 3-7. The 'x-axis' in mentioned figure signifies the linearization time step and the 'y-axis' depicts the average mean-square

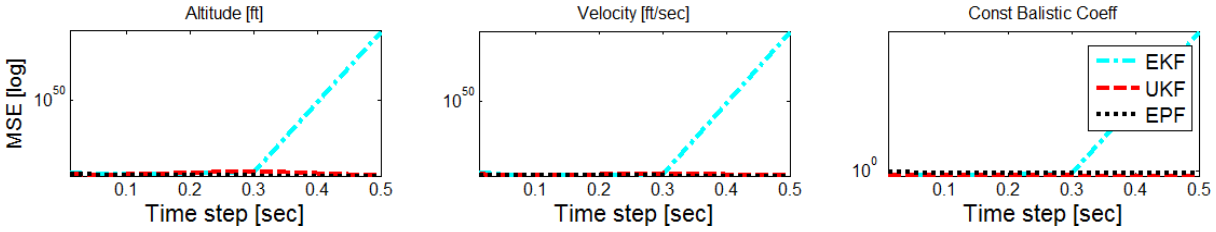

Figure 3-7: Average MSE [log] over various linearization times

error of the state (in log scale) and the title over each plot gives the state name. As witnessed in the figure, the linearization error in EKF proves to be devastating to its estimation accuracy, as the average mean-square error in EKF increases drastically with the time step more than $0.3\ sec$. However, the average mean-square error in case UKF and EPF remains low and comparable throughout the variation domain.

3.1.2.3 Poor initial guess

The estimation accuracy for changing initial conditions are analyzed in the current section. Various initial state vectors were obtained by percentage addition (and subtraction) of the original state vector (used before) to original state vector itself and thereafter performing the state estimation process by all the filters, over 10 MC simulations. The resulting estimated states, output and average estimation error (absolute value) are presented in Figure 3-8, Figure 3-9 and Figure 3-10, respectively. In these figures, the title over each column, indicates the percentage variation of the initial conditions used for the corresponding estimation process, from the original initial conditions, given in equation (3-7).

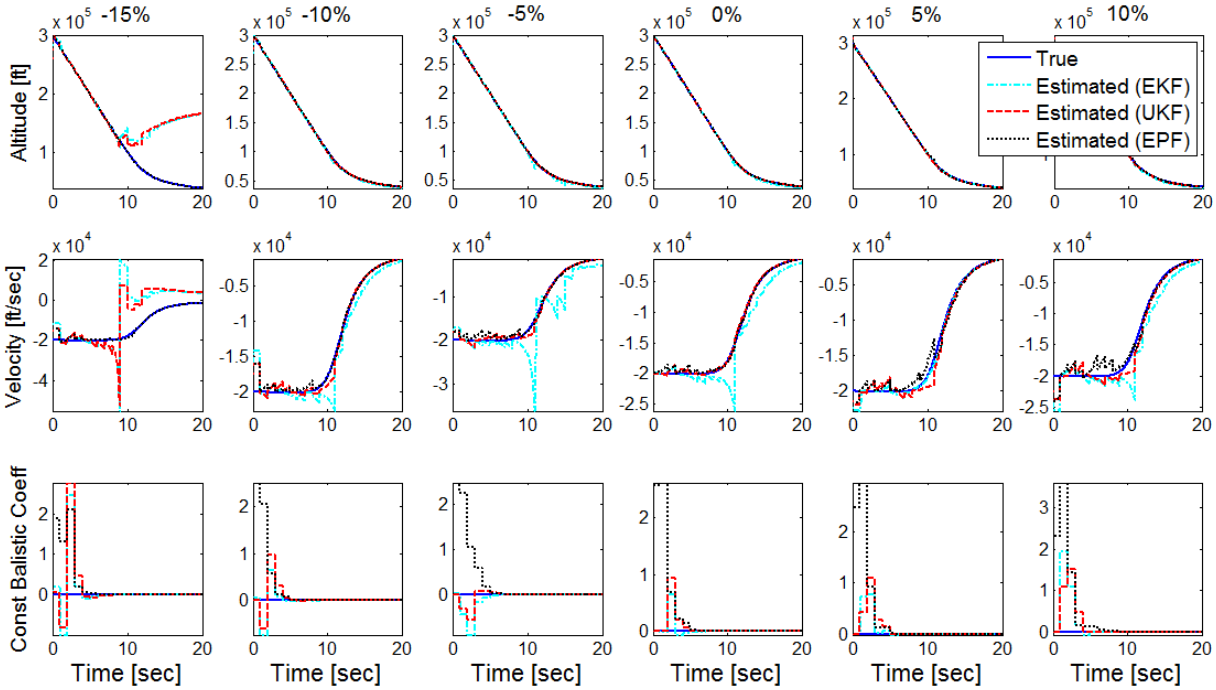

Figure 3-8: Estimated States for varying initial state vector

As illustrated from Figure 3-10, the estimation accuracy of Gaussian filters (EKF and UKF) significantly deteriorated for higher variations (-15% and 10%) compared to the Bayesian filter (EPF). Along with EKF, which throughout resulted inferior performance, UKF also started showing poor convergence to true value, after $\pm 5\%$ variation of initial state vector. This behavior can also be explained by a similar reason given before. That is, as the initial state approximations get poorer, it becomes difficult for the Gaussian filters to converge back to true values and also, their criticality for divergence towards nonlinearities increase (they tend to be

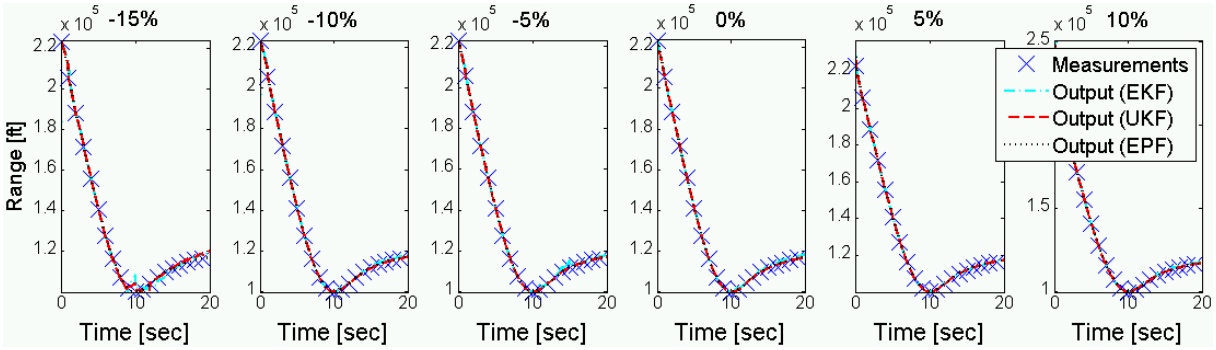

Figure 3-9: Reconstructed Output for varying initial state vector

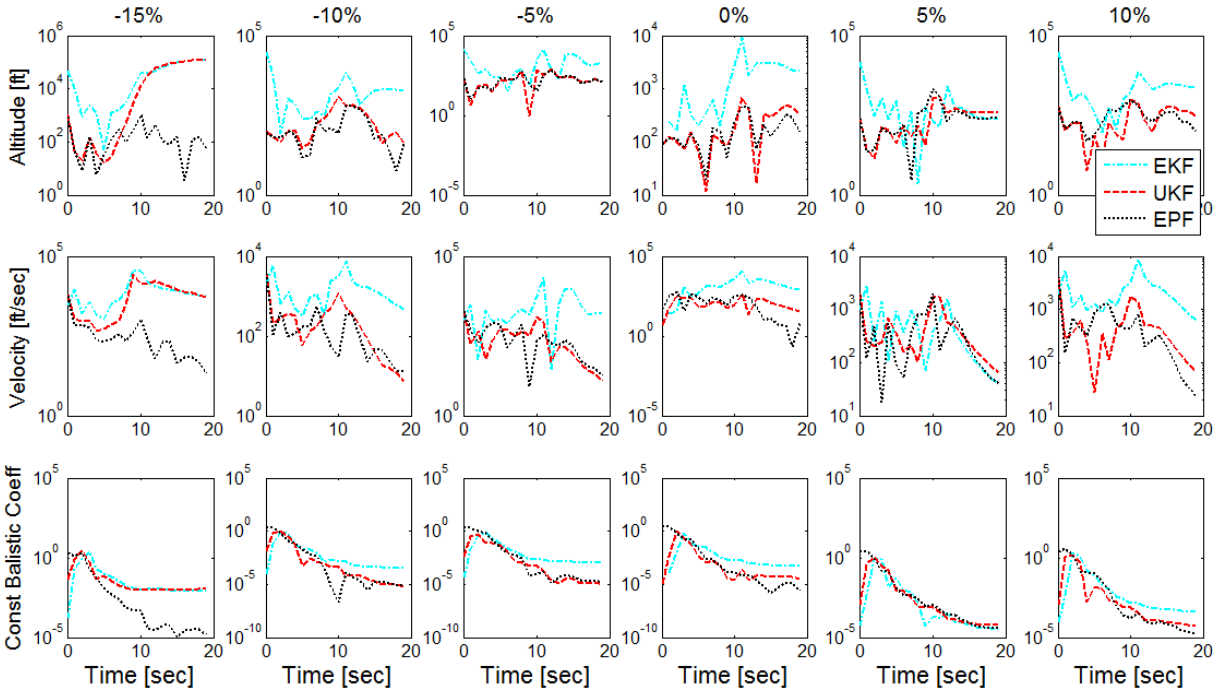

Figure 3-10: Average Estimation error [log] for varying initial state vector

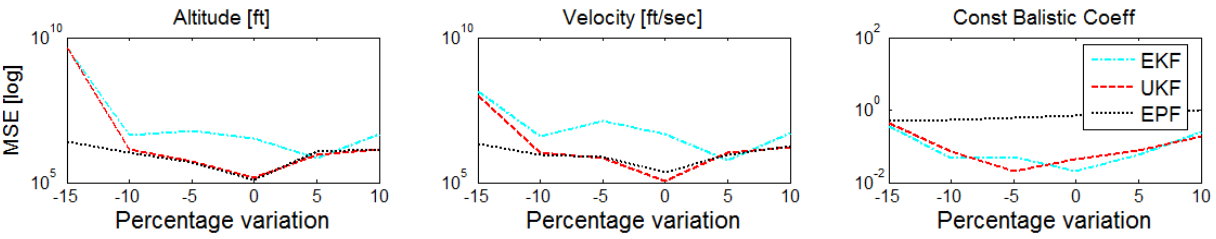

Figure 3-11: Average MSE [log] over varying initial state vector

very sensitive towards nonlinearities). On the other hand, since Bayesian filters do not have any Gaussian assumptions, they converge to true values at faster rates and do not diverge even in the presence of severe nonlinearities.

Another measure to account for accuracy, i.e. MSEs are also plotted with varying initial conditions and are shown in Figure 3-11. Sudden increase of errors at maximum variations was observed for EKF and UKF whereas, MSE of EPF showed a smoother variation, gradually increasing outwards (both sides).

3.1.2.4 Conclusion

As evident from the presented figures, all the filters obtained similar results till $9\,secs$ of simulation time, when the body was at high altitude with minimal drag effects and was

approximately falling linearly. Thereafter, the drag increased significantly, which resulted in a highly nonlinear motion and eventually lead the filters to produce significantly different estimations. Figure 3-3 illustrates that, even in the presence of substantial nonlinearities, UKF being a Gaussian filter, still estimates with a comparable accuracy to a Bayesian filter (EPF), while the estimates of EKF were mostly poor. Furthermore, the performance of UKF and EPF were primarily equivalent for most of the cases except, when large linearization time (time step more than $0.3\ secs$) and broad variations of initial condition (-15% and 10%) were analyzed. For the considered example, this was anticipated because for most of the cases Gaussian assumptions hold and hence, a Gaussian filter (UKF) was expected to produce good results at very less computational effort. Moreover, as the computational time taken by EPF was very high compared to EKF or UKF, hence its usage for this particular example is worth only for mentioned extreme cases while for rest of the large application domain UKF proved efficient enough.

3.2 Example Problem

In the current section, state estimation problem of a scalar system is described. Reason for considering following example is the presence of significant nonlinearities in process and measurement models which is also described in (Arulampalam, 2002). In the particular case, comparison of EKF with various type of particle filters is done on the basis of computational time and accuracy. Furthermore, the performance of all PFs is exploited by comparison among each other.

3.2.1 Problem description

The algebraic process model of the system is given as

$$x_k = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + w_k. \tag{3-11}$$

Also, the measurement equation is

$$z_k = \frac{x_k^2}{20} + v_k, \tag{3-12}$$

where, w_k and v_k are zero mean Gaussian process and measurement noise respectively.

The initial state vector (*xi_uncorr_0*) and initial covariance matrix (*Pi_uncorr_0*) are

$$xi\_uncorr\_0 = 0.1 \tag{3-13}$$

$$Pi\_uncorr\_0 = 10 \tag{3-14}$$

Also, Q and R are the process noise covariance matrix and measurement noise covariance matrix, respectively and are initialized as,

$$Q\_PSD = 10; \quad R = 1 \tag{3-15}$$

3.2.2 Results

The plots were obtained after initializing the filter in a similar manner as done for previous example. Also for all types of particle filters, $N_s = 1000$ (number of particles) was used. For better analysis, the results were obtained across a Monte Carlo simulation consisting of 5 runs.

The corresponding estimated state and output by all the filters are displayed in Figure 3-12, along with their true state and measurements. Thereafter, the absolute value of the estimation error for all the filters are shown in Figure 3-13.

3.2.2.1 Comparison with EKF

It is evident from Figure 3-13 that the estimation error of EKF is $7 - 8$ times (in absolute scale) as that of any PF used. The reason is the local linearization and Gaussian assumptions of EKF due to which, it is incapable of appropriately approximating the underlying bimodal posterior density and hence is prone to either choosing a wrong mean or just taking the average between the means. This eventually justifies the statement that Gaussian filters (EKF) are unreliable for the highly nonlinear systems that have non-Gaussian posterior density (bimodal in this case), and if tried to estimate based on Gaussian assumptions, will produce extremely erroneous results.

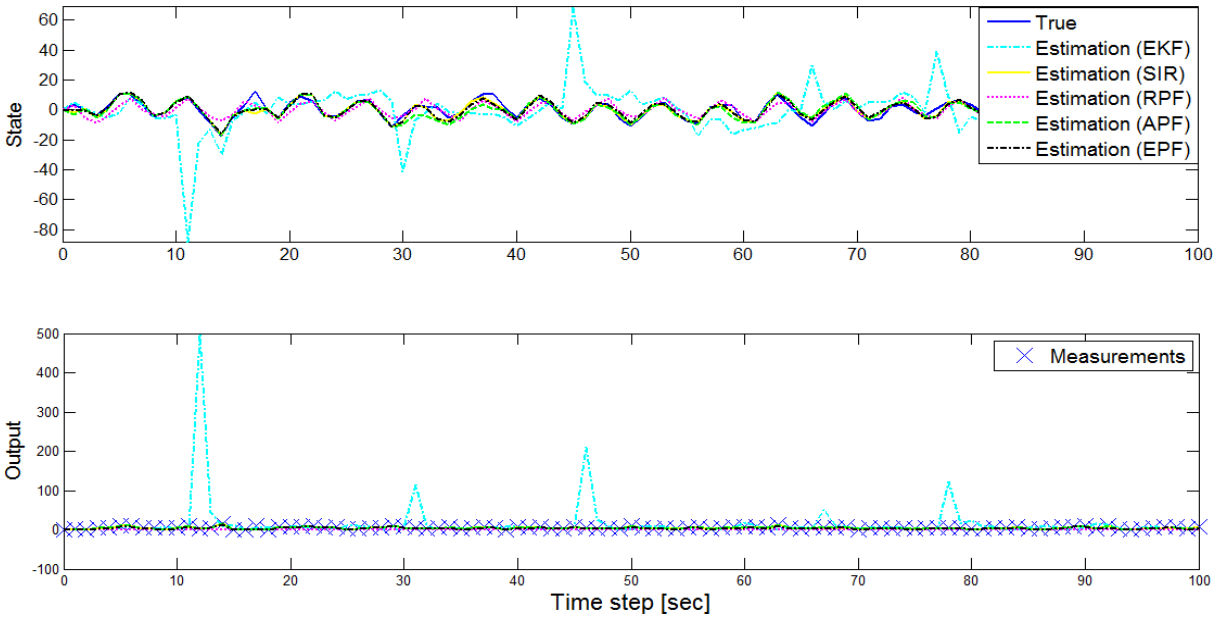

Figure 3-12: Estimated State and Output

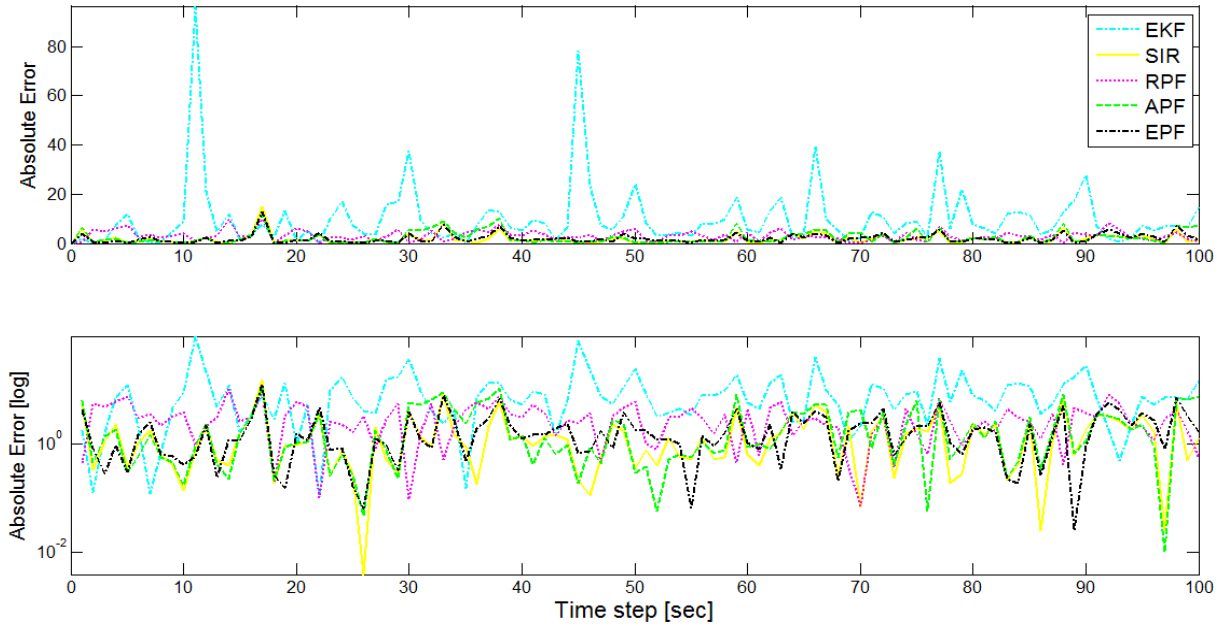

Figure 3-13: Absolute estimation error

These results were also obtained using the original Matlab scripts (*'m-files'*) and the corresponding computational time used by each filter along with their mean-square estimation error (MSE) are listed in Table 3-3 and Table 3-4, respectively.

EKF	SIR	RPF	APF	EPF
0.0257	8.2889	8.1242	11.9258	29.3328

Table 3-3: Computational time [sec]

EKF	SIR	RPF	APF	EPF
1328.3524	21.1087	79.7852	37.4004	27.3446

Table 3-4: Mean-square estimation error

It can be observed from Table 3-3 that the computational time taken by EKF is the lowest and the time taken by EPF is the highest whereas, other particle filters SIR PF, RPF and APF are almost similar. Even though the run time of EKF is very less compare to any PF, but its high MSE value, as seen from Table 3-4, provides enough motivation to bear high computational cost of employing a PF for severely nonlinear systems.

3.2.2.2 Comparison among each other

In the current subsection, performances of previously discussed PFs are evaluated and compared to one another. Although from Figure 3-13, one can hardly find any discrepancy in the estimation accuracy of all four particle filters but, the distinctiveness gets more evident from the table of MSEs (Table 3-4). To analyze the effect of changing number of particles on the run time and estimation accuracy, the variation of computational time and MSE for all the PFs are shown in Figure 3-14 and Figure 3-15, respectively. The observations of each PF is discussed sequentially and also, for better visualization, the estimated state by each PF is plotted again, separately in Figure 3-16, Figure 3-17, Figure 3-18 and Figure 3-19.

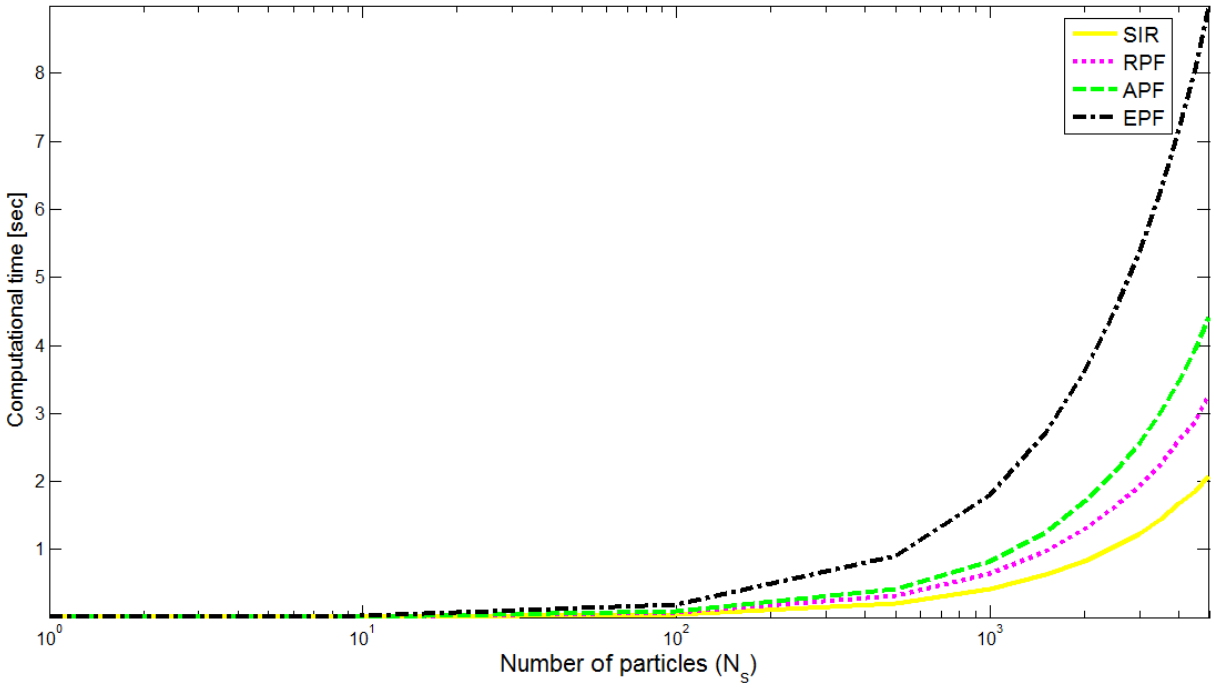

Figure 3-14: Average Computational time over varying number of particles

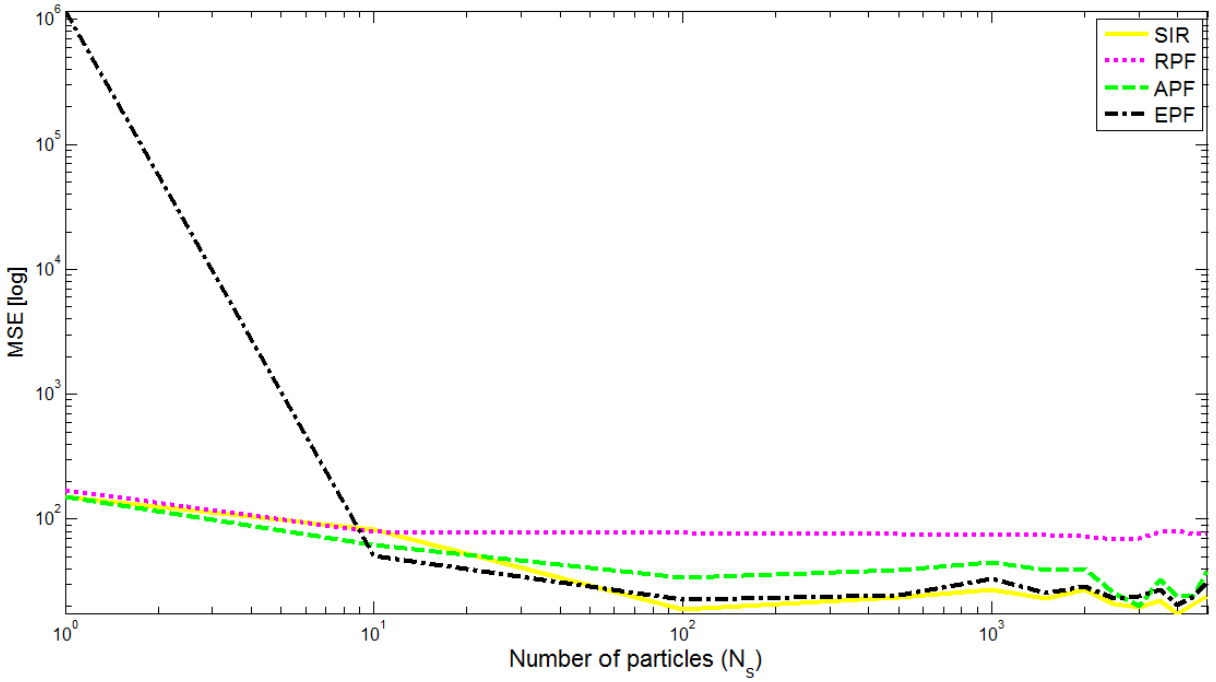

Figure 3-15: Average mean-square error [log] over varying number of particles

SIR particle filter

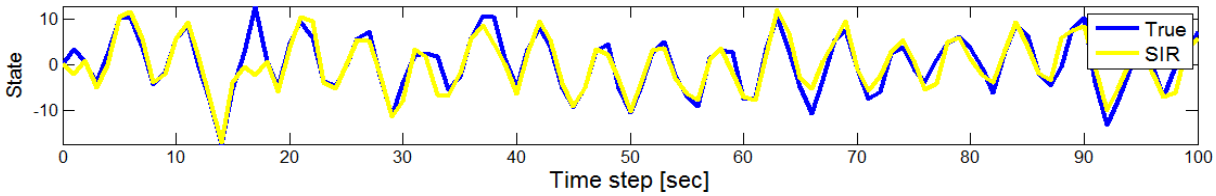

Figure 3-16: Estimated State by SIR particle filter

The SIR PF employing systematic resampling algorithm, resulted in the most accurate estimation for $N_s = 1000$, over other PFs, as can be observed from Table 3-4. However, as displayed in Figure 3-15, the SIR obtained unsatisfactory results for less number of particle ($N_s < 50$). The reason for this is inadequate amount of sampling due to the availability of low number of particles from the prior distribution (broad in this case), which is regarded as the importance density and because of which the posterior density in not fully approximated.

Furthermore, as seen in Figure 3-14, the variation of computational time, though exponentially rising with number of particles, is still less than other types, on account of its simplest execution.

Regularized particle filter

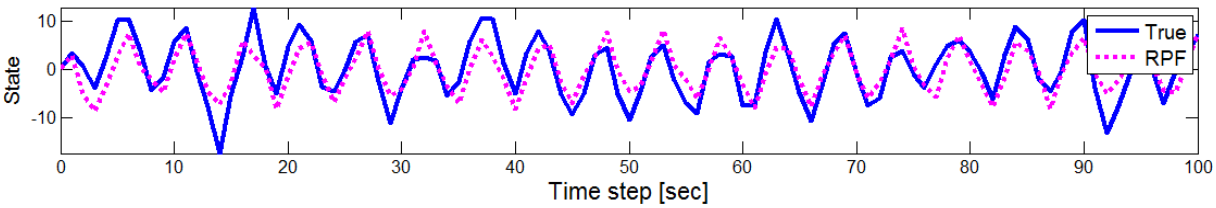

Figure 3-17: Estimated State by RPF

In this example, the RPF did not result in substantial reduction of errors over SIR for $N_s = 1000$, as visible from Table 3-4. This might be because of the presence of significant process

noise, due to which the regularized samples (samples obtained from continuous distribution) are no longer guaranteed to asymptotically approximate those from the true posterior density. The observation of almost constant MSE (Figure 3-15) with increasing number of particles can also be explained by a similar reason.

The variation of run time with N_s for RPF is comparable to SIR PF, as shown in Figure 3-14, because RPF only requires N_s additional computations for generations from kernel density at each time step.

Auxiliary particle filter

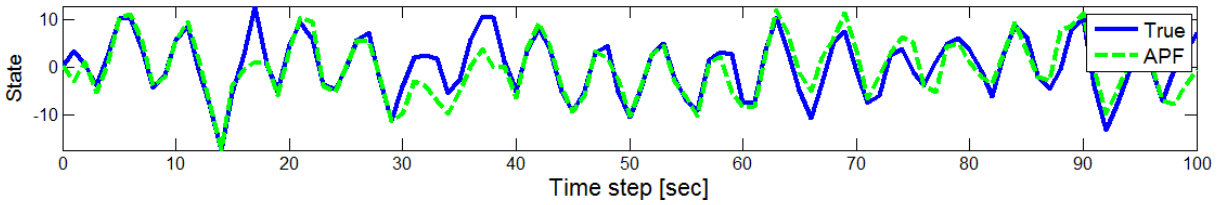

Figure 3-18: Estimated State by APF

As one may anticipate, one of the artifice to reduce estimation errors made by SIR PF, is to choose proposed particles in a more intelligent manner. Hence the APF did a descent estimation for same number of particles and also its MSE decreases with increasing number of particles, compared to SIR, as depicted in Table 3-4 and Figure 3-15, respectively. However, as substantial process noise enters the system equations, a single point $\boldsymbol{\mu}_k^i$ (auxiliary variable) could not characterize the prior distribution well, which pushed the APF to resample based on a poor approximation of the prior and hence did not further improve the estimation accuracy over SIR.

In addition, the computational time increases more in comparison to SIR, as seen in Figure 3-14, because of additional computations involved for evaluating auxiliary variable and parent weights.

Extended particle filter

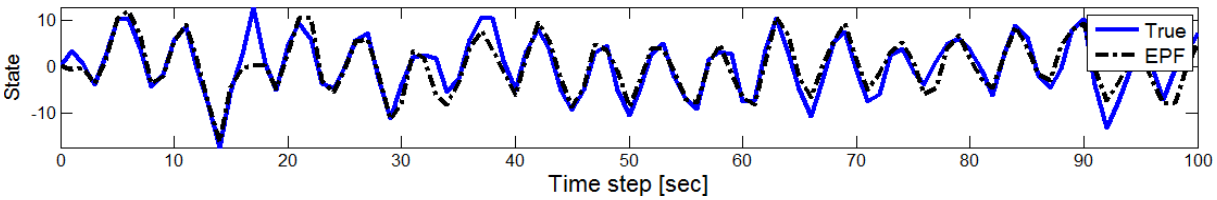

Figure 3-19: Estimated State by EPF

Although, EKF was seen to obtain poor estimates for the considered bimodal posterior density, the combined type of PF overcame its deficiency almost completely. This is in accordance with the statement made before, that combined type of PFs often perform better than the individual filters. The estimation accuracy of EKF for $N_s = 1000$ is primarily of the order of SIR PF, as also witnessed in Table 3-4. However, the slightly positive offset of the MSE for EPF over SIR PF can be explained by the bimodal nature of posterior density, which EKF tries to approximate using Gaussian distribution to propagate (and correct) particles for each measurement time. Therefore, the usage of EKF for propagating particles in EPF for non-Gaussian systems may introduce additional error in state estimation over normal SIR particle filter.

Figure 3-15 brings out that for very small sample size ($N_s < 10$), EPF estimates the state equally poor as estimated by EKF. This is due to the unavailability of enough samples for

supporting the execution of PF in adequately approximating the posterior density using weighted samples.

Furthermore, the computational time of EPF increases at the highest exponential rate compared to any other PF, as depicted in Figure 3-14. Reason for this are the N_s number of EKFs, that are running together to propagate (and correct) particles at all measurement times, as stated previously as well.

3.2.2.3 Conclusion

By the means of above example, estimation performances of Gaussian (EKF) and Bayesian (PFs) estimators were analyzed particularly for a non-Gaussian system, with bimodal posterior density. On one hand, EKF linearizes the nonlinear process and measurement models in order to approximate the probability density by a Gaussian distribution, while on the other hand, particle filters directly approximate the density using weighted random samples.

Four types of particle filters, the SIR PF, the RPF, the APF, and the EPF were analyzed based on various aspects. For the particular example with 1000 particles, SIR PF was seen to outperform other PFs by obtaining the most accurate estimations (with minimum MSE) in least amount of computational time. However, the designing of any PF for a particular problem is critical to the selection of importance density. This is because, a well-tuned importance density is in a way an adequate tradeoff between the number of particles and the computational expense necessary for each particle, yielding the best qualitative performance with economical computational effort (Arulampalam, 2002).

4 Conclusions and perspective

The Flight Path Reconstruction procedure is effectively a nonlinear estimation problem whose preciseness is dependent upon the chosen state estimator to a great extent. Further challenges may arise for the state estimator with low-quality sensors and higher levels of process noise during a flight test.

The Gaussian filters (particularly Extended Kalman Filter) have been extensively used as state estimators for the FPR application, but owing to their intrinsic limitation of explicit calculations of Jacobian to linearize the system, unsatisfactory results may arise. Also, the Unscented Kalman Filter exhibits certain advantages over EKF due to the propagation of exclusively selected sigma points for approximating the Gaussian distribution rather than truncating the Taylor series to an arbitrary order, as done in EKF. However, on account of the employment of Gaussian assumptions for approximating posterior density in all Gaussian filters, the divergence is merely deferred.

In the application part of this report, the filtering algorithms were first implemented for the FPR of a Re-Entry vehicle example using EKF, UKF, and Extended particle filter, and then on a nonlinear state estimation of an algebraic system, employing EKF and other types of PFs (Sampling Importance Resampling PF, Regularized PF, Auxiliary PF and EPF). For the first problem, the performance comparison between the three filters, based on the obtained estimation errors, revealed that the estimation accuracy of UKF and EPF were comparable whereas the accuracy of EKF was below satisfactory level.

In other execution, performance of EKF was observed to be extremely poor over all PFs. Apparently it can be said that, in severe nonlinear environments, a nonlinear filter such as PF offers an improvement in accuracy over an approximate nonlinear filter like EKF. The mentioned improvement resulted on account of approximating the density, rather than modeling the nonlinear system. In addition, when the PFs (SIR PF, RPF, APF, and EPF) were compared among each other, the SIR PF outperformed other PFs by obtaining minimum MSE estimates in least computational time. As a remark, the performance of a PF for a particular problem is more affected by an optimum choice of importance density than an indefinite selection of number of particles.

For future work, it is envisioned to implement both the filtering algorithms (Gaussian and Bayesian) for *Flight Path Reconstruction of a simulation model* of an aircraft using actual flight test data for additional characterization of the performance benefits of one over other. In far future, there is also a possibility of further extension of the work for real-time flight applications.

5 References

Arulampalam, M. S. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2, FEBRUARY 2002*, 174-188.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tr acking Applications.* Linköping, Sweden: Dissertations No. 579, Linköping Studies in Science and Technology, Linköping University.

Doucet, A. (1998). *On sequential Monte Carlo methods for Bayesian filtering.* Dept. Eng., Univ. Cambridge, UK, Tech. Rep., 1998.

Hayking, S. (2001). *KALMAN FILTERING AND NEURAL NETWORKS.* New York: Wiley-Interscience, 2001.

J.A. Mulder, Q. C. (1999). Non-linear Flight Path Reconstruction review and new advances. *Progress in Aerospace Science (1999)* (pp. 673-726). HS delft, Neitherlands: Alsevier Science Ltd.

Karlsson, R. (2005). *Particle Filtering for Positioning and T racking Applications.* Linköping 2005: Department of Electrical Engineering Linköping University , SE-581 83 Linköping, Sweden.

M. Pitt, N. S. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association. June 1999, vol. 94, no. 446*, 590–599.

Mehndiratta, M. (2014). *Master's Thesis : Comparison of Extended Kalman Filter and Unscented Kalman Filter for Flight Path Reconstruction in System Identification.* Garching, Munich: Institute of Flight System Dynamics, TUM.

Simon J. Julier, J. K. (1995). A New Approach for Filtering Nonlinear Systems. *American Control Conference*, (pp. 1628 - 1632). Seattle, Washinton.

Simon, D. (2006). *Optimal State Estimation: Kalman,H Infinity and Nonlinear Approaches.* New Jersey: Wiley-Interscience, 2006.