PARTICLE TRACKING USING THE UNSCENTED KALMAN FILTER IN HIGH ENERGY PHYSICS EXPERIMENTS

Jahanzeb Akhtar

A thesis submitted for the degree of Doctor of Philosophy

Brunel University London

September 2015

Acknowledgments

I would like to thank my supervisor, Dr. Roger Powell, for the support and encouragement and advice. Thanks for always being there in all those meetings and fruitful discussions, I am inspired by your passion and enthusiasm towards research. I am also thankful to my second supervisor, Dr. Paul Kyberd, for his support.

I would also like to thank my family especially my parents and uncle, Mr. Muhammad Jamal, for their encouragement and support throughout my research.

Abstract

The extended Kalman filter (EKF) has a long history in the field of non-linear tracking. More recently, statistically-based estimators have emerged that avoid the need for a deterministic linearisation process. The Unscented Kalman filter (UKF) is one such technique that has been shown to perform favourably for some non-linear systems when compared to an EKF implementation, both in terms of accuracy and robustness.
In this Thesis, the UKF is applied to a high energy physics particle tracking problem where currently the EKF is being implemented. The effects of measurement redundancy are investigated to determine improvements in accuracy of particle track reconstruction. The relationship between measurement redundancy and relative observability is also investigated through an experimental and theoretical analysis.
Smoothing (backward filtering), in the high energy physics experiments, is implemented using the Rauch Tung Striebel (RTS) smoother with the EKF , however, in Unscented Kalman filter algorithms, the Jacobian matrices required by the RTS method, are not available. The Unscented Rauch Tung Striebel (URTS) smoother addresses this problem by avoiding the use of Jacobian matrices but is not efficient for large dimensional systems such as high energy physics experiments. A technique is implemented in the RTS smoother to make it suitable for the UKF. The method is given the name the Jacobian Equivalent Rauch Tung Striebel (JE-RTS) smoother. The implementation of this method is quite straight forward when the UKF is used as an estimator.

Contents

List of Figures

List of Tables

List of acronyms

HEP	High Energy Physics
LHC	Large Hadron Collider
MICE	Muon Ionization Cooling Experiment
TPC	Time Projection Chamber
ITS	Inner Tracking System
TRT	Transition Radiation Tracker
TOF	Time Of Flight
VLPC	Visible Light Photon Counter
EKF	Extended Kalman Filter
CKF	Combinatorial Kalman Filter
SPKF	Sigma Point Kalman Filter
UKF	Unscented Kalman Filter
CDKF	Central Difference Kalman Filter
SSSPKF	Spherical Simplex Sigma Point Kalman Filter
SS-UKF	Spherical Simplex Unscented Kalman Filter
SR-UKF	Square Root Unscented Kalman Filter
SR-CDKF	Square Root Central Difference Kalman Filter
UPF	Unscented Particle Filter

AUPF	Adaptive Unscented Particle Filter
WLS	Weighted Least Square
MSE	Mean Squared Error
VdP	Van der Pol
RTS	Rauch Tung Striebel
URTS	Unscented Rauch Tung Striebel
JE-RTS	Jacobian Equivalent Rauch Tung Striebel

Chapter 1

Introduction

1.1 Track reconstruction in high energy physics experiments

In High Energy Physics (HEP) experiments such as Large Hadron Collider (LHC) experiments running at CERN [1], millions of particles traverse the particle detectors and for each particle position, momentum, and energy measurements are needed for particle identification. Therefore sorting out the measurements for each particle and estimating them afterwards in such large dimensional problems is a big challenge. Typically this is achieved using track reconstruction which is decomposed into two tasks; track finding and track fitting. Track finding deals with sorting out the measurements in terms of their origin whereas track fitting estimates these set of measurements as accurately as possible. The desire of high accuracy and the issue of computational complexity makes track reconstruction a challenging task in HEP experiments.

Currently, in most of the LHC experiments, track finding is achieved through pattern recognition [2, 3] where measurements are divided into subsets, each subset contains measurements that belong to the same particle. The subsets are passed to the Extended Kalman Filter (EKF) for track fitting [4]. The overall description of the track reconstruction process is shown in figure 1.1. Track fitting problem is a challenging task and a focus of this thesis. Track fitting mainly depends on track model, how closely the mathematical function replicates the actual system. This includes mainly the incorporation of ionization energy loss into the overall system dynamics. Track fitting also depends on the efficiency of the algorithm, how efficiently an algorithm deals with the material effects (multiple scattering) inside strong magnetic field and ionization energy loss. After all the tracks are fitted, smoothing (backward filtering) is implemented to improve the accuracy [5]. Smoothing is achieved using RTS which was designed to work with the EKF.

The focus here is to investigate particle tracking problem (track fitting and smoothing) and analyse how it can be improved. One way of achieving the improvement could be to replace the current track fitting algorithm with a more efficient one by keeping in mind the challenges related to the system in question.

1.2 Challenges

1.2.1 Uncertainties

The main source of uncertainty, in any engineering problem, that contributes as measurement noise is the accuracy of the sensors. In HEP experiments, highly accurate sensors are needed since the measurements are subsequently used for particle identification and the detector alignment [6]. Particle identification is highly important task since it is the core of such experiments. Alignment on the other hand is important because it uses the fitted data provided by track fitting to improve the accuracy by aligning the detector [7]. Data is taken from small sensors, units of the large detector,

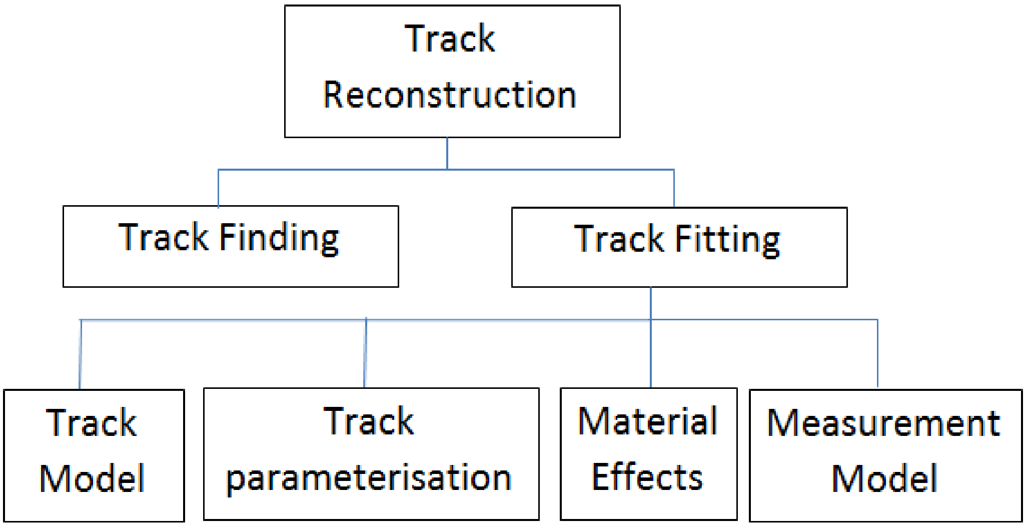


Figure 1.1: Overall structure of the track reconstruction process

to identify the position of each sensor where maximum hits are found and the detector is accordingly aligned. Typically these sensors have a resolution in microns.

The source of uncertainty in the process is multiple scattering [8], caused by the very thin detector layers that are in fact used to measure the position and momentum of a particle. This affects the trajectory of a particle when it hits the detector layers which does not only affect the position but also the momentum of a particle. Multiple scattering in fact is a great source of non-linearity as well since it happens inside a very strong magnetic field.

1.2.2 Non-linearity

Generally strong magnets encapsulates the particle detectors and consequently the particles traversing the detector follows a helical trajectory due to the uniform magnetic field. Multiple scattering inside such strong magnetic field makes track fitting a highly non-linear problem. This happens due to the fact that the more the magnetic field strength is, the more the curvature of the helical path that a particle follows. Consequently, when the particle hits the detector layer, the particle deviates with more uncertainty and it becomes difficult for the estimator to provide accurate results. Moreover, a very small change in the process noise can cause an estimator to diverge and raises the issue of relative observability.

1.2.3 Large dimensionality

As explained above track fitting is a large dimensional problem since data from millions of particles has to be fitted and to determine each particle, six measurements are needed. Since covariances are also needed in order to take account of the accuracies and for other computation purposes, the storage and computations of the large dimensional matrices are huge computational burdens. Furthermore, these large dimensional covariances are also needed for smoothing computations which also contributes towards computational costs.

To cope with such challenges, any track fitting algorithm should be accurate and robust against outliers and non-linearities. The algorithm should achieve this by maintaining the computational cost. EKF is providing reasonably acceptable results but an improvement can be achieved by replacing a more efficient algorithm.

1.3 Research objectives

1.3.1 Implementing the suitable algorithm

Investigate different algorithms that can be used to improve track fitting by keeping minimum the computational cost. Implement the selected algorithm to HEP problem and compare it with the EKF, currently being implemented for track fitting, in terms of accuracy and robustness against non-linearities. Analyse how the selected algorithm would behave in high non-linearities and what other advantages, apart from good accuracy, it could provide.

1.3.2 Accuracy and relative observability analysis

After the implementation of an efficient algorithm to track fitting, investigate how the accuracy and relative observability of the estimates produced by the algorithm can be improved. Relative observability is a measure of the degree of observability which means that if a system is observable, how closely observable. The issue of relative observability is raised by the non-linearities in the HEP experiments and causes an algorithm to perform badly or even diverge.

1.3.3 Derivation of an efficient smoothing algorithm

Smoothing is an algorithm that is implemented with the estimator to improve the accuracy of the estimates. Currently, RTS smoother [5] is being implemented with the EKF in almost all the HEP experiments for track fitting, it cannot be directly implemented with the UKF. Although a smoother exists that can be directly implemented with the UKF, its computational cost is high and therefore is not a suitable choice for such problems. Therefore, the objective was to derive a smoother that can be implemented more efficiently with the UKF in large dimensional problems such as HEP experiments.

1.4 Contributions

1.4.1 Implementation of the UKF to track fitting and comparison with the EKF

The HEP experiments were studied to understand the structure, dynamics and uncertainties involved in track fitting problem. It also involved the performance analysis of the existing algorithms currently being implemented. As explained above, EKF is currently being used as an algorithm for track fitting but an efficient algorithm can improve track fitting. An implementation of the UKF to track fitting for the MICE is presented along with its comparison with the EKF. It is worth mentioning here that the UKF has never been implemented to track fitting problem and, therefore, never been compared with the EKF in terms of accuracy and robustness against non-linearities within HEP experiments. The comparison presented shows that the EKF is less robust against non-linearities and the UKF is a better choice for track fitting.

1.4.2 Accuracy and relative observability analysis and improvement

A detailed mathematical and experimental analysis of the performance of the UKF is presented in order to shed light on the phenomena of redundancy, accuracy and observability. The relationship between measurement redundancy and accuracy and relative observability is investigated and it is presented that measurement redundancy does not always improve the quality of the estimates. The effects of different redundant measurements on each measurement are discussed and it is presented that the performance of the UKF in terms of accuracy and observability can be increased by carefully selecting the redundant measurements. Two cases are considered where firstly duplicate measurements are introduced as redundant measurements and secondly an independent redundant measurement (measurement not included in the state vector) is introduced. UKF has never been analysed before in such a way according to the best of our knowledge.

1.4.3 Improvements through time of flight measurement

As explained in the last section, accuracy and relative observability was improved by introducing duplicate measurements and independent redundant measurements. The independent redundant measurement that was considered and introduced is Time Of Flight (TOF) measurement. TOF was introduced due to the fact that firstly it is already being measured (easily available) and secondly it contains information that can contribute towards an improvement in position and momentum estimates. This is a potential contribution because TOF measurement has never been used in such a way to improve the position and momentum estimates in the HEP experiments. Moreover, no such theoretical analysis has been presented with an independent redundant measurement according to best of our knowledge.

1.4.4 Derivation of a suitable smoother for UKF in large dimensional problems

The thesis presents an improvement over an earlier method called the URTS smoother [12]. A smoother is derived for the UKF that works more efficiently as compared to the URTS smoother. The derived smoother is given the name "Jacobian Equivalent RTS (JE-RTS) smoother" since it uses, like URTS smoother, the RTS smoother's standard update equations. The JE-RTS smoother and URTS smoother have been implemented using the MICE system of equations and compared in terms of accuracy and computational cost. These smoothers provide similar accuracy but the JE-RTS smoother outperforms the URTS smoother in terms of computational cost.

1.5 Thesis outline

Chapter 2 discusses the track reconstruction problem in HEP experiments. Different techniques that are currently being used for track finding and track fitting processes are discussed. Track fitting being the main focus of this research, the EKF as an existing track fitting algorithm is briefly explained. Other advanced algorithms which are currently being implemented for tracking outside particle track fitting field are reviewed in terms of their performance as compared to the EKF. Keeping the main challenges of HEP experiments, accuracy and computational complexity, in mind each algorithm is discussed in the context of track fitting. Finally smoothing in HEP experiments is discussed.

Chapter 3 firstly explains the MICE [11]. Since position and momentum for each muon is measured by the scintillating fibre trackers and these measurements are passed to the Kalman filter for track fitting, they are discussed in terms of construction and measurement uncertainties. Finally track reconstruction in the context of MICE is explained and the sources of uncertainty inside tracker are discussed. Secondly, the EKF and the forms of the UKF, augmented and non-augmented UKFs [9], and Spherical Simplex Sigma point Kalman Filter (SSSPKF) [10] are presented. These algorithms have been implemented to a two dimensional non-linear system called Van der Pol oscillator and compared in terms of robustness against noise and non-linearity. Since the augmented and non-augmented UKFs and SSSPKF generates and propagate different number of sigma points, they are compared in terms of computational complexity. And finally, the UKF and EKF are compared in terms of accuracy and robustness against non-linearity for track fitting at MICE.

Chapter 4 presents the analysis of the UKF to MICE for track fitting. It is shown that how UKF works with the measurement redundancy and how this can be used to improve the accuracy of the estimates. UKF has also been analysed in terms of relative observability, and analysis about the relationship between relative observability and degree of measurement redundancy is provided.

In chapter 5 the RTS smoother is presented and its implementation with the EKF is discussed. It is explained why RTS smoother cannot directly be implemented with the UKF and why the URTS smoother is not suitable for large dimensional systems. A technique, implemented in the RTS smoother to make it suitable for the UKF and large dimensional systems, is presented and the resulting smoother is given a name "the JE-RTS smoother". Finally a comparison of the URTS and JE-RTS smoothers in terms of accuracy and computational complexity is provided.

Chapter 6 provides a detailed conclusions and discussions of this thesis.

Chapter 2

Literature review

2.1 Track reconstruction

Track reconstruction, in HEP experiments, is divided into two parts called track finding and track fitting [4]. Track finding is achieved using pattern recognition or classification technique. Measurement set provided by the tracking detector is divided into subsets where each subset contains measurements related to the same particle. These subsets are called track candidates. Track fitting aims at estimating a set of parameters, as accurately as possible, describing the state of the particle according to the set of measurements in a track candidate.

2.1.1 Track finding

Track finding is classified into two types; local and global methods. Hansroul, H. Jeremie, and D. Savard Global, [13], presented that conformal mapping can be implemented as a global method of track finding. Hough, P. V. C, [14], showed that Hough transform can be implemented as a global method. And recently T. Alexopoulos et al., [15], presented that Legendre transform can also be implemented as a global method. Local methods are track road and track following. Track road is an approach that needs a measurement set, created by the same charged particle, to be initiated. Then track model can be used to find the path between different measurements. The measurements that lie inside the boundaries of the track path (road) are included in a track candidate. Track following is initiated with a seed build from few measurements. These seeds are constructed in the inner region of the tracking detector close to interaction region or in the outer region. The purpose of constructing seed in the inner region is that the measurements are of high precision whereas in the outer region the track density is lower. The seed is used to extrapolate the track to the next layer of the detector containing a measurement and the closest measurement to the predicted one is selected for the track candidate. This is repeated until too many detector layers with missing measurements are found/encountered or until the whole detector system is traversed.

2.1.2 Track fitting

In the track fitting algorithm, a set of parameters representing the kinematic state of the charged particle are estimated using the measurements. Since the measurements have uncertainties attached to them, the track fit also provides measure of uncertainty about these measurements in the shape of a covariance matrix. Track fitting process is decomposed into the following steps.

Track parametrization

Five parameters may be used to uniquely describe the state of a charged particle [46, 47, 48]. These parameters can be defines by a vector as follows,

$$\mathbf{x} = \begin{bmatrix} d_p \\ \phi_0 \\ k \\ d_z \\ tan\lambda \end{bmatrix} \tag{2.1}$$

where d_p is the signed distance of the pivot from the helix in the x-y plane. ϕ_0 is an azimuthal angle that specifies the pivot with respect to the center of the helix. k is the inverse of transverse momentum, it also provides the charge of the track assigned during the track fitting. d_z is a longitudinal parameter that represents a signed distance of the helix, in z direction, from the pivot. And the final parameter, $tan\lambda$ represents track's slope.

A charged particle follows helical trajectory inside uniform magnetic field and its position is determined by;

$$x = x_0 + d_p cos\phi_0 + \alpha/\kappa(cos\phi_0 - cos(\phi_0 + \phi)) \tag{2.2}$$

$$y = y_0 + d_p sin\phi_0 + \alpha/\kappa(sin\phi_0 - sin(\phi_0 + \phi)) \tag{2.3}$$

$$z = z_0 + d_z - \alpha/\kappa tan\lambda\phi \tag{2.4}$$

where α is the constant of magnetic field. ϕ is turning angle that is used to determine the location. ϕ is an internal parameter with a sign, positive or negative, that represents the polarity of the charged particle. $\mathbf{x}_0 = (x_0, y_0, z_0)^T$ is the pivot where the helix parameters and the error covariance matrix are defined. Figure 2.1 shows the dynamics

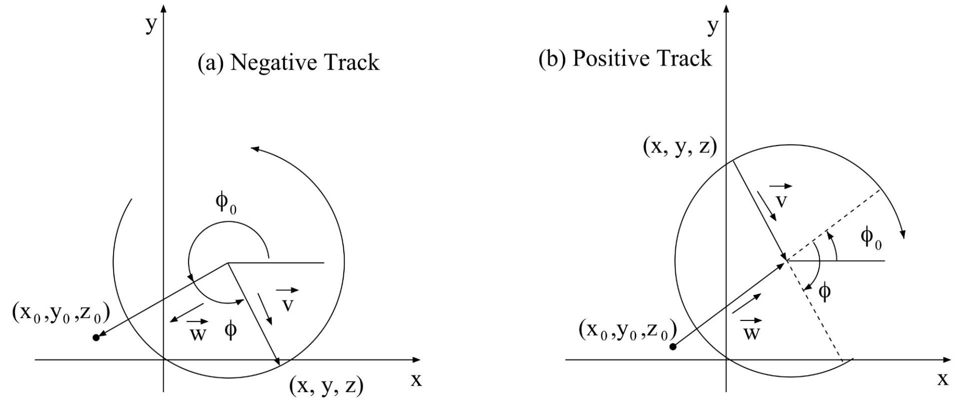


Figure 2.1: A schematic representation of helix parameterization. (a) represents a track of a negatively charged particle whereas (b) represents a track for a positively charged particle. It is clear from (a) and (b) that if the particles are oppositely charged, they travel an opposite direction. The vectors w and v in the figure can be defined as $w = (cos\phi_0, sin\phi_0)^T$ and $v = (cos(\phi_0 + \phi), sin(\phi_0 + \phi))^T$. Adapted from Yukiyoshi Ohnishi, [48].

of the charged particle in track parametrisation in the case of a helix.

The track parameters are chosen according to the geometry of the tracking detector. If cylindrical detector layers are used, then the reference surface is often cylindrical and makes the radius times azimuthal angle the natural choice of one of the position parameters, whereas in the case of planar detector layers it is shown by L. Bugge and J. Myrheim, [52], that Cartesian position co-ordinates are frequently used.

Track model

The dependence of a state vector at a given surface k on the state vector at a different surface i can be determined using the track model,

$$\mathbf{q}_k = \mathbf{f}_{k|i}(\mathbf{q}_i) + \mathbf{v}_k \tag{2.5}$$

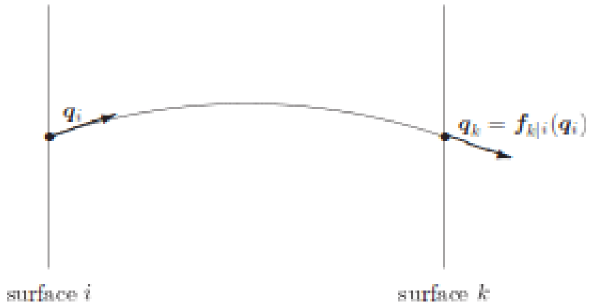


Figure 2.2: An illustration of the track model and propagation concepts. The function $\mathbf{f}_{k|i}$ is the track propagator from surface i to surface k. Its mathematical form depends on the track model, i.e., the solution of the equation of motion in the actual magnetic field. Adapted from Are Strandile and Rudolf Fruhwirth, [24].

where $\mathbf{f}_{k|i}$ is the track propagator from surface i to surface k and $\mathbf{q}_i$ is the state vector containing parameters explained above and $\mathbf{v}_k$ is the process noise. The sources of process noise encountered inside the particle detector are briefly explained in the Material effects section below where it is also described how different process noises are treated in the track fitting process. An illustration is provided in figure 2.2.

Error propagation

Track fitting process requires the propagation of the track parameters along with the corresponding covariances. This linear error propagation is typically achieved using a similarity transformation between layer i and k.

$$\mathbf{C}_k = \mathbf{F}_{k|i}\mathbf{C}_i\mathbf{F}_{k|i}^T \tag{2.6}$$

where **C** is the covariance matrix and **F**$_{k|i}$ is the propagated Jacobian matrix from layer i to k,

$$\mathbf{F}_{k|i} = \frac{\partial q_k}{\partial q_i} \tag{2.7}$$

A. Strandlie and W. Wittek, [53], presented that in the case of analytical track models the Jacobian is also analytical. Whereas in the inhomogeneous magnetic fields, the derivatives are calculated purely by numerical schemes. L. Bugge and J. Myrheim, [54], presented that they can also be calculated by semi-analytical propagation of the derivatives in parallel with the Runge-Kutta propagation of the track parameters.

Material effects

C. Amsler et al., [8], presented that particle trajectory is mainly affected by, in the detector volume, ionization energy loss and multiple coulomb scattering. Multiple Coulomb scattering is considered as an elastic process, which disturbs the direction of a charged particle traversing a detector in the case of a thin scatterer. However Amsler et al, [55], presented that in the case of a sufficiently thick scatterer the position in a plane transversal to the incident direction is also changed. Multiple coulomb scattering does not only disturbs the direction but also the momentum of a charged particle. To account the effects of multiple coulomb scattering a covariance is updated in the track fitting procedure. In order to include the effects of multiple scattering, the reason for only updating the process covariance matrix is that the mean value of the scattering angle and eventual offset is zero.

Radiation energy loss also contributes as noise in the case of light particles such as electrons. Fruhwirth et al, [52], showed that the fluctuation of ionization energy loss is very small and treated in the track fitting process to correct the state vector. However in the case of Bremsstrahlung energy loss, H. Bethe and W. Heitler, [56], presented that it suffers large fluctuations and, therefore, affects both the state vector and corresponding covariance matrix.

Measurement model

The measurement model h_k represents how actual measurements (hits) $\mathbf{m}_k$ in a detector layer k are functionally dependent on the state vector $\mathbf{q}_k$ at the same detector layer,

$$\mathbf{m}_k = \mathbf{h}_k(\mathbf{q}_k) + \mathbf{w}_k \tag{2.8}$$

Usually, the measurement vector $\mathbf{m}_k$ contains position measurements but, according to the requirement of the overall experiment, may also contain other quantities such as momentum and energy. $\mathbf{w}_k$ is the measurement noise that occurs due to the accuracy of the sensors and is considered as Gaussian noise. Since track fitting is typically being achieved using the EKF and linear least squares method, the Jacobian $\mathbf{H}_k$ is essential and is given by,

$$\mathbf{H}_k = \frac{\partial \mathbf{m}_k}{\partial \mathbf{q}_k} \tag{2.9}$$

In the case of Jacobian containing only rotations and projections, which is mostly the case, it can be computed analytically.

2.2 Background of track fitting in HEP experiments

Track fitting is mainly achieved using linear least squares method and the Kalman filtering.

2.2.1 Linear least squares method

Track fitting is generally performed using linear least squares methods which is an optimal approach if the system model is linear. In particle physics terms, this approach is optimal if, in equation 2.5, the track propagator $\mathbf{f}_{k|i}$ is a linear function of the state vector $\mathbf{q}_i$ from layer k to i (detector layers) and the noise encountered during estimation process is Gaussian. One of the advantages of using linear least squares method is that these are easy to compute. However P. J. Rousseeuw and A. M. Leroy , [16], showed that they lack robustness.

In the case of thick scatter or substantial multiple scattering, the estimated states, representing a track, can significantly deviate from the actual track. H. Eichinger and M. Regler, [57], presented that this issue can be resolved by estimating two projected scattering angles either at each detector layer or at a set of virtual breakpoints inside a continuous scatterer. These two methods, global least square method and breakpoint method, provide similar results for the estimates of the state vector. Therefore in this sense these algorithms are equivalent [58].

The problem with least squares method is that they require a linearized track model if the systems dynamics are non-linear. The linearization is normally acquired by an approximation procedure such as first of approximation using Taylor expansion. Since an approximated model is used, accuracy is sacrificed. Another issue is that, due to the static nature of this method, the noise is not taken into consideration locally. The source of process noise in the HEP detectors is the interaction of the charged particles with the detector layers.

These methods become numerically inefficient when applied to large dimensional systems such as track fitting. This inefficiency arises due to the requirement of inverting large matrices which also contributes significantly towards computational cost. However for Kalman filtering, P. Billoir, [59], and R. Fruhwirth, [60], presented that the kalman filter contains the same features of breakpoint method in terms of estimation quality and does not require the inversion of large matrices. The additional advantage of the Kalman filter is that the material effects (process noise) such as ionization energy losses and multiple scattering can be incorporated locally.

2.2.2 Extended Kalman filter

The Kalman filter may be divided into two steps: prediction and update. The track parameters $\mathbf{q}_{k-1|k-1}$ are propagated through the track model in order to predict track parameters $\mathbf{q}_{k|k-1}$ in the next detector layer containing a measurement:

$$\mathbf{q}_{k|k-1} = \mathbf{f}_{k|k-1}(\mathbf{q}_{k-1|k-1}) \tag{2.10}$$

The associated covariance is calculated as

$$\mathbf{C}_{k|k-1} = \mathbf{F}_{k|k-1}\mathbf{C}_{k-1|k-1}\mathbf{F}_{k|k-1}{}^{T} + \mathbf{Q}_{k} \tag{2.11}$$

where $\mathbf{F}_{k|k-1}$ is computed by linearizing the function $\mathbf{f}_{k|k-1}$. The mathematical function that is linearized by the EKF is presented by equations (2.2)-(2.4). $\mathbf{C}_{k-1|k-1}$ is the updated covariance computed at detector layer $k-1$. $\mathbf{Q}_k$ accounts for the multiple scattering after layer $k-1$ up to and including layer k.

The update step adds the information from the actual measurement in order to correct the state vector:

$$\mathbf{q}_{k|k} = \mathbf{q}_{k|k-1} + \mathbf{K}_{k}(\mathbf{m}_{k} - \mathbf{h}_{k}\mathbf{q}_{k|k-1}) \tag{2.12}$$

where $\mathbf{m}_k$ is the vector of actual measurements and $\mathbf{h}_k$ is the measurement function. $\mathbf{K}_k$ is the Kalman gain matrix given by

$$\mathbf{K}_{k} = \mathbf{C}_{k|k-1}\mathbf{H}_{k}{}^{T}(\mathbf{v}_{k} + \mathbf{H}_{k}\mathbf{C}_{k|k-1}\mathbf{H}_{k}{}^{T})^{-1} \tag{2.13}$$

where $\mathbf{H}_k$ is computed by linearising the measurement function $\mathbf{h}_k$. The updated covariance is calculated by

$$\mathbf{C}_{k|k} = (\mathbf{I} - \mathbf{K}_{k}\mathbf{H}_{k})\mathbf{C}_{k|k-1} \tag{2.14}$$

The process of the Kalman filtering can be seen in figure 2.3.

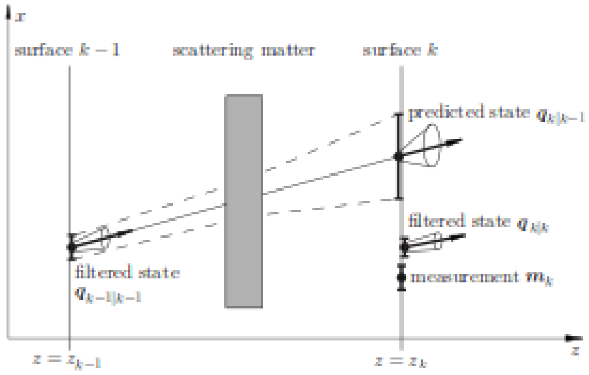


Figure 2.3: Prediction and filter step of the Kalman filter. The propagation proceeds in the z direction, while the x coordinate is measured. Adapted from Regler et al.

2.2.3 Existence of outliers and their removal

Outliers can often be found in a track candidate produced by the track finding algorithm. The origin of these outlying observations can be distorted hits, irrelevant hits from some other tracks or electronic noise. To deal with such outliers, R. Fruhwirth, [60], presented that an obvious and more practical way is to continuously analyse the results (how close the observations and the predicted track positions are) provided by the estimator using all the measurements but the one under consideration. If only one outliers is found, it is more feasible solution to reject the outliers.

However, if several outliers are found in a track candidate, the smoothed predictions become biased and there arises a possibility that good observations are rejected. This issue can be dealt with by making the track fit more robust, the effect of the outliers on the overall solution can be reduced by incorporating down-weighting techniques. Recently, adaptive estimators have been presented that are more robust in a sense that they automatically down-weight the influence of the outlying observations on the overall solution. I.A. Golutvin et. al., [61], presented a similar approach that is based on a re-descending M-estimator and uses Tukeys bi-square function [62].

Once the track finding process is completed and track candidates have been sorted, track candidates may be found with one or more hits in common. Mostly, it happens if the track finding has been done sequentially. Such incompatible tracks has to be sorted in a way that their maximal or optimal subset is found. This problem can be solved by finding a graph in which each track corresponds to a node and two nodes are connected to make an edge if compatibility is found between two corresponding tracks. However, finding all maximal node sets in a graph was an issue that is resolved by S. R. Das, [63].

Since millions of tracks are being dealt with in the HEP problems, several solutions may be found when one searches for a maximal set of compatible tracks. Additionally, it is also required to take into account the quality of the track. To accomplish this, quality index is assigned to each track which is based on the track length, Chi square statistics, direction of the track and the distance of the track from interaction region. The maximal compatible node set, which has the maximum quality index, is considered the best one. The best node set can be found by using a recurrent neural network (Hopfield network). R. Fruhwirth, [64], presented this implementation to DELPHI experiment in its forward chambers. D. Wicke, [65], explained an algorithm which was developed to provide global solutions for tracking ambiguities faced in DELPHI.

2.2.4 Hybrid methods

To deal efficiently with the outliers, Billoir, P., and S. Qian, [2], presented that the Kalman filter can be used to perform track finding and track fitting. It starts out with finding the seed in couple of adjacent layers then follows each seed through the detector and picks up compatible hits. Figure 2.4 provides an idea about the function of the hybrid methods.

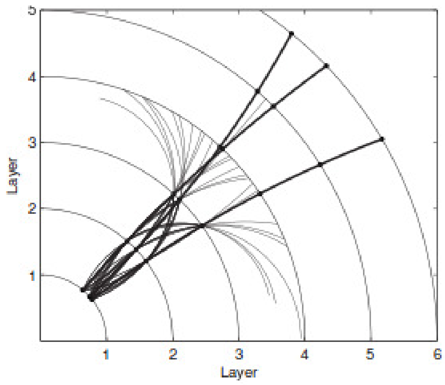


Figure 2.4: Example of progressive track recognition with three tracks. Seeds are formed in layers 1-3 by finding all possible combinations of hits. Each of the seeds is extrapolated to the outer layers thin circles. Seeds that do not reach the next layer or are not compatible with an observation in some layer are discarded. The thick circles represent the three seeds that are successfully propagated to the outermost layer. Adapted from Are Strandile and Rudolf Fruhwirth, 2010

Combinatorial Kalman filter

If there are many nearby tracks or high density of noisy measurements, the measurement closest to the predicted track might not necessarily belong to the track under consideration. Mankel, [3], implemented the combinatorial Kalman filter in order to cope with such issue.

CKF, similar to progressive track finding process, starts from a short track segment (seed) measured either at the inner end of the tracking detector or at the outer end. While propagating forward to the first layer after the sees, several Kalman filter branches are generated if several compatible measurements are found. Each of these branches contain a unique compatible measurement. Since the detector inefficiencies in such large dimensional systems are inevitable, a branch, with a missing measurement, is also created to cope with potential detector inefficiencies.

After all the branches are produced and sorted at the current detector layer, the branches with at least one compatible measurement are propagated to the next detector layer. At this stage new branches are created for the combinations of predicted states compatible with the new measurements found. Propagating further by following this process of generating new branches makes a combinatorial tree where Kalman filters run in parallel.

A criteria is defined within the CKF algorithm to assess different branches in terms of their quality. Branches that fall below a defined quality value are automatically removed. Figures 2.5 and 2.6 describes the function of the CKF. Since millions of tracks have to be processed, computational complexity is an issue that is also taken into account. The way CKF achieves this is by removing the branches that traverse several consecutive layers without a compatible measurement. The branch with the highest quality is selected at the end of the process when all the detector layers have been traversed. Therefore, a complete account of the branches is taken and analysed in parallel with the track fitting process.

An algorithm which is quite similar to the CKF has been coded in a language called Cellular Au-tomata [66, 68, 67]. The approach is called Cellular Automaton and can be regarded as local due to the fact that it generates branches using the measurement found in the nearby detectors layers. The approach does not need to be initiated from a seed. Cellular Automaton processes all the measurements in parallel and this functionality makes it a hybrid approach between a global and local method.

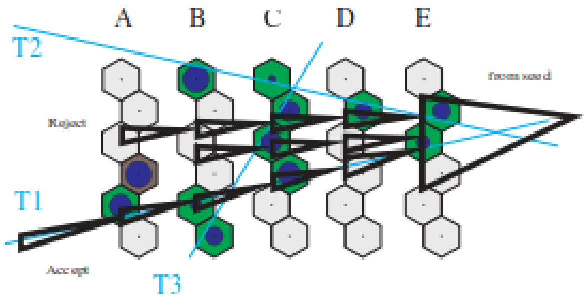


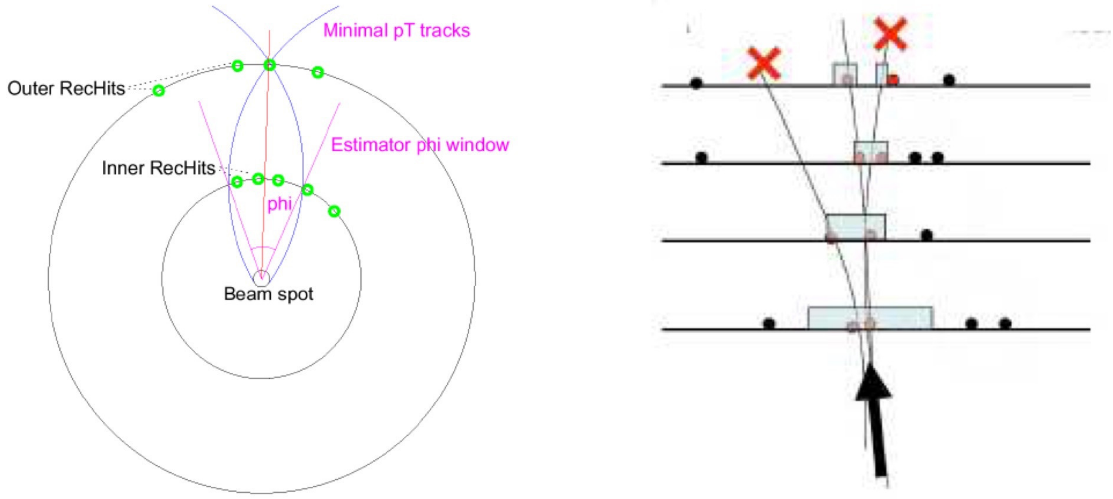
Figure 2.5: Illustration of the combinatorial Kalman filter for an ambiguous situation caused by three nearby tracks in a super layer of the HERA-B outer tracker. The propagation proceeds upstream from the right to the left. It is assumed that the propagation started with a seed of hits from track T1. From Mankel, [3].

2.3 Role of the Kalman filter in the LHC experiments

CKF is currently being used as a track re-constructor in almost all of the HEP experiments especially in four LHC experiments named ATLAS, ALICE, CMS and LHCb [1, 50, 51, 49]. These experiments are different in terms of the nature of the experiment and implementation of the track reconstruction. But the overall track reconstruction process is decomposed into the following steps irrespective of the nature of the experiment.

2.3.1 Track finding

CKF is being used for track finding in all the LHC experiments except LHCb [1]. In LHCb, track finding is performed by searching the peaks, representing the distances from measurement to a parametrised trajectory, in the histogram. Global approaches are not being used except in ALICE, ATLAS and CMS. ALICE has implemented Hough transform and Hopfield neural networks in the Time Projection Chamber (TPC) and



(a)

(b)

Figure 2.6: (a) Schematic view of the CKF seeding with hit pairs. CKF starts with the hit in the outer layer and searches for the hits found in a phi window that corresponds to a minimum p_T and are compatible with the beam spot. (b) CKF starts from the seed, processes compatible hits iteratively leading to trajectory candidate. The candidate on left can be seen as discarded since no compatible hits were found whereas the track on right is also stopped because the error becomes too high after the last hit is added. Red dots shown in the figure are compatible measurements whereas black dots are incompatible. From G.B. Cerati, [51].

Inner Tracking System (ITS) respectively, and ATLAS used Hough transform in Transition Radiation Tracker (TRT). CMS implemented Hopfield net several years ago but dont currently use it.

2.3.2 Track fitting

Although a global least squares approach is still being used in the ATLAS inner detector and, by default, in muon system of the ATLAS experiment, as an alternative, the Kalman filter is by far the most commonly used track fitting algorithm.

2.3.3 Post processing

In the CMS experiment, the track candidates are removed that have too many measurements in common for the purpose of trajectory cleaning. ATLAS implements outlier rejection at various stages during the track reconstruction process.

Like in any of the LHC experiments, track reconstruction in MICE is divided into two parts; track finding and track fitting [23]. Space points are constructed from the hits on different fibre channels. Pattern recognition is used to generate seed which is subsequently passed to Kalman filter to fit it with the space points in the next channels. Kalman filter predicts and smooth the given track at the remaining space points. MICE is briefly explained in chapter 3.

2.3.4 Alignment

The data provided by track fitting is used afterwards in alignment [6, 7]. Alignment is a problem of aligning the detectors according to the best fit trajectory of the particles determined in track fitting. The alignment procedure is carried out to optimize the measurement system. Alignment involves an inversion of large matrices to obtain corrections. It is a typical $\mathbf{Ax} = \mathbf{b}$ problem where any large matrix A makes it difficult to solve x. To overcome this issue, different methods have been considered that can be used in order solve large matrix inverses efficiently [6]. The Kalman filter avoids these large inverses and is currently being used for alignment [24, 25]. Alignment is out of scope of this thesis and the main focus is track fitting.

2.4 Tracking in non-HEP experiments

Since Gaussian approximate solutions are relatively easier to implement and they provide modest computational cost, they have gained attention of different scientific communities over the last 40 years. Under the assumption that system dynamics are linear and all the probability densities are Gaussian, the Kalman filter [69] is the optimal solution for the Bayesian estimation problems. The solution is optimal, if the assumptions hold, in the Minimum Mean Squared Error (MMSE) sense, the Maximum Likelihood (ML) sense, the Maximum A Posteriori (MAP) sense and it also asymptotically achieves the Cramer-Rao lower bound [70]. However, the original Kalman filter can only be implemented to a linear system. Since almost all the systems are non-linear and the Kalman filter cannot be directly implemented, actual non-linear systems are linearized using first order Taylor series expansion. The algorithm is known as the EKF [71, 72, 73].

The EKF has widely been used in nonlinear estimation problems for the last forty years. It has been successfully implemented to the problems of probabilistic inference, state estimation, parameter estimation and dual estimation [71, 74, 75]. The EKF is extensively being used in integrated navigation systems which are used to measure position, altitude and velocity in the aerospace industry for different aircrafts [76, 77, 78, 79]. These systems receive measurements from avionic sensors such as

Global Positioning System (GPS) and magnetic compass, and use the EKF for tracking. In tracking, one of the first implementation of the EKF was to the Apollo moon mission in the navigation computer /citeHoag1963.

In parameter estimation, Puskorius and Feldkamp from Ford Research Laboratories applied the EKF to the training of recurrent neural networks for real-world automotive engineering problems such as engine misfire detection, drop-out rejection, sensor catalyst modeling and on-vehicle idle speed control [81]. The results for this work was a custom VLSI based on a recurrent neural network framework [82].

Sequential Monte Carlo methods often use extended Kalman filters as subcomponents [83] for more powerful inference. However the implementation of the EKF is suboptimal due to the fact that it uses first order Taylor series linearization, the EKF ignores the fact that the prior and predicted system state variables are random variables themselves. This issue of not properly accounting the state variable's probabilistic spread severely affect the accuracy of the posterior predictions. Subsequently, updated state estimates provided by the filter are also affected. Consequently, a filter can diverge in a case if it keeps generating inconsistent estimates of the estimation error covariance. This happens due to the fact that filter trust more in its own generated estimates as compared to the true state space evolution and actual measurements. These issues directly affect the performance of the EKF and also any inference system that is based on the EKF or uses EKF as a component.

To overcome the issues related to the EKF, new algorithms have been proposed in the literature that are based on deterministic sampling methods. These methods avoid the step of linearization, instead they propagate Gaussian random variables through the actual non-linear systems. Julier and Uhlmann derived the Unscented Kalman Filter

(UKF) in the general context of state estimation for automatic control applications [84, 85, 86, 17]. The UKF uses the unscented transformation [35] to transform the sampling points through the actual non-linear systems by avoiding the need to linearize it. This method works directly with the non-smooth and non-analytical systems due to the fact that it only need functional evaluations of the true non-linear dynamics.

Two similar algorithms, based on the Stirling's interpolations formula [87], to the UKF were introduced in the late nineties. These filters are similar to the UKF in a sense that firstly they are non-derivative based and secondly they use deterministic sampling approach to propagate Gaussian mean and covariance through the actual non-linear system. These approaches were published approximately at the same time by Ito and Xiong [88], who called their algorithm the Central Difference Kalman Filter (CDKF) and by Norgaard et al [89], who gave their algorithm a name Divided Difference filter (DDF). It is worth mentioning here that although these approaches, the CDKF and the DDF, were published independently at the same time, they are the same algorithm essentially. However, we will be using the CDKF for the review purposes below.

The algorithms discussed above, the UKF, the CDKF and the DDF, are similar in a sense that they are non-derivative based approaches based on deterministic sampling framework. All these algorithms generate sigma points and propagate them through the actual non-linear systems. The similarities in these algorithms allow to group them together in a family called sigma point kalman filters (SPKFs) [22].

Later on, Rudolph van der Merwe and Eric Wan [90, 91] presented numerically robust versions of the UKF and CDKF. They applied a technique within these algorithms and due to that technique gave these algorithms a name Square root UKF (SRUKF) and Square Root CDKF (SRCDKF). They have also shown that there are some interesting parallels between SPKFs and a statistical approach called weighted statistical linear regression (stochastic linearization) [71, 92] and it provides an insight how SPKFs are a better choice as compared to the EKF in terms of robustness for Gaussian probabilistic inference problems.

Until now a brief history of the current approaches included in the family of SPKFs has been provided to shed some light on their origin. However, now a review on these approaches in terms of their advantages, disadvantages and comparisons will be provided. All these algorithms have been published and compared extensively in literature and it is worth mentioning here that they all have advantages and disadvantages. We will try to cover a decent amount of literature to give an insight into these algorithms by also keeping in mind their applicability to HEP problems. It is also worth mentioning here that new tracking algorithms, has not been discussed before in this thesis, called particle filters will also be reviewed.

Although EKF is widely being used for non-linear state estimation in various applications, relatively new non-derivative based approaches have been presented in the literature. SPKFs are a type of algorithms that have extensively been compared with the EKF in terms of accuracy. The SPKFs perform better because, unlike the EKF, the actual non-linear function is not linearized and instead the sampling points are used. Two approaches considered to be within SPKFs framework are the Central Difference Kalman Filter (CDKF) and the UKF and they are similar in terms of accuracy [21, 22].

Although the UKF and the CDKF provides similar accuracy, the CDKF, as compared to the UKF, is quite complex in its formulation. The difference arises in the calculations of covariances and the Weights for sigma points. The UKF, therefore, is frequently being used in several non-linear problems, excluding particle physics, and compared with the EKF in terms of accuracy.

Simon J. Julier and Jeffrey K. Uhlmann [17] presented the UKF to avoid the issues faced by the EKF. The main issue associated with the EKF is its difficult implementation in highly non-linear systems. The other issues are EKF's less accuracy and its divergence in severe non-linearities. They compared these algorithms and it is shown that the UKF performs better. Julier showed how, for the same computational cost, the UKF consistently outperforms the EKF in terms of state estimation accuracy and estimate consistency [93].

Eric A. Wan and Rudolph van der Merwe [18] also compared the EKF and UKF and it is shown that EKF, due to linearization, provides less accurate results. EKF approximates the actual non-linear system to first order and consequently posterior mean and covariance of the transformed Gaussian random variable is affected and the accuracy is sacrificed. It is shown that the UKF provides better results in terms of accuracy at a comparable computational complexity.

Fredrik Orderud [19] compared the EKF and UKF for a system with non-linear measurements. Before, these algorithms have been extensively compared when the process model was non-linear. It is shown that UKF performs better for the radar model but the results are similar for the tracking model. The reason is that the radar model is highly non-linear as compared to the tracking model which was relatively linear.

Joseph, J. LaViola [20] compared the EKF and UKF using a non-linear tracking. Due to the near linearity of the dynamics, EKF is a better choice because it provides similar results in terms of accuracy but outperforms UKF in terms of computational complexity. Hence EKF is a better choice for Nearly linear system.

Rudolph van der Merwe and Eric Wan [22] presented the updated versions of the SPKFs. It is shown that a technique can be implemented within the UKF and CDKF frameworks for numerically robust and efficient implementation. The UKF and CDKF that incorporate it are called Squarer Root UKF (SR-UKF) and Squarer Root CDKF (SR-CDKF). Square Root SPKF perform more efficiently as compared to their original forms in terms of robustness.

Although the square root versions of the SPKF are numerically stable and are more robust as compared to their original counterparts, extra incorporated steps cause computational complexity. So robustness is gained over the additional computational cost. Moreover an additional inverse in the Kalman gain calculations can be a bottleneck especially in the case of a system that is not well behaved.

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp [26] presented an approach in the non-linear filtering called the particle filters. It is shown that the particle filters can cope with any non-linearities and Gaussian and non-Gaussian distribution.

Katalin Gyrgy, Andrs Kelemen and Lszl Dvid [27] has compared the particle filter with the UKF and EKF. It is shown that the particle filter and the UKF are easier to implement as compared to the EKF and outperforms it in terms of accuracy. The UKF and EKF has comparable computational load whereas it is comparatively high in the particle filters due to a large number of particles involved. Eleni N. Chatzi, and Andrew W. Smyth [28] has also compared the particle filters with the UKF and has shown that the UKF is computationally more efficient.

Kung-Chung Lee, Anand Oka, Emmanuel Pollakis and Lutz Lampe [29] have compared the UKF and particle filters in terms of robustness and concluded that particle filters are more robust because they can cope with any non-linearities and Gaussian and non-Gaussian distributions.

Although particle filters, as compared to the UKF, are more accurate, they have other issues associated with them. One of the main issues is that the particle filters have a high computational cost. Another issue is that particle filters rely on importance sampling, they require the design of proposal distribution that can approximate the posterior distribution really well which is difficult to design in general. To overcome this issue, Rudolph van der Merwe , Arnaud Doucet, Nando de Freitas and Eric Wan [30] presented the Unscented particle filter (UPF). This method incorporates UKF's technique to produce samples which helps in using the latest available information efficiently. The UPF uses the UKF for proposal distribution to incorporate the latest observations and match the true posterior distribution more closely. UPF is well suited for engineering problems where sensors are very accurate but non-linear.

UPF avoids the issue faced by the particle filters at an extra computational cost. The reason is that each particle uses the UKF to obtain the importance proposal. Since UPF struggles with the large dimensional problems, its real time capability is also questionable. To overcome this, Wentao Yu, Jun Peng, Xiaoyong Zhang, Shuo Li, and Weirong Liu [31] presented a new algorithm called the Adaptive UPF (AUPF) that adaptively adjust the number of particles during filtering process to reduce the unnecessary computations.

Although AUPF has been shown to perform better as compared to the UPF in terms of computational cost, adjusting the number of particles might affect the accuracy.

Therefore, all the discussed tracking algorithms have merits and DE-merits and they should be chosen according to the challenges faced due to the dynamics of the system. In the next section, the above discussed tracking algorithms are discussed in terms of their suitability to HEP experiments for track fitting.

2.5 A critical review of some tracking algorithms within HEP experiments

As discussed above, EKF is currently being used for track fitting and is performing reasonably well. However, there are some issues related to track fitting that are currently being faced. One of the main issues are high non-linearity of the track fitting problem that is caused by the multiple scattering inside a strong magnetic field. Multiple scattering disturbs the position and momentum of the charged particle and provided the fact that it happens inside trackers encapsulated in very strong magnets, the EKF performs badly or even diverges due to the linearization process. The EKF also diverges due to very small changes that occur in the momentum of the particle caused by the internal dynamics such as the ionization energy loss.

In the Muon ionization cooling experiment, scintillating fibre trackers are used to measure the position and the momentum of a particle. The purpose of this experiment is to reduce the emittance of the muon beam where cooling is typically measured using these scintillating fibre trackers. This is achieved by installing one tracker at the start of the experiment (upstream tracker) and one at the end (downstream tracker) to compare the momentum measurements. If the transverse momentum is decreased, the muon beam is believed to have been cooled down and obviously the emittance is reduced. The main issue is that the downstream tracker is expected to receive straight particle tracks or at least tracks with very low transverse momentum, which is not always true. Due to such uncertainty that is very difficult to model, the issue of observability arises. The current track fitting algorithm, the EKF, performs poorly or even diverges.

These issues can be resolved by replacing a more efficient tracking algorithm. The algorithms that can be used to increase the accuracy of the estimates and improve track fitting are the UKF, CDKF and particle filters and their updated versions. However, in such large dimensional problems, the filter should be chosen that can improve track fitting by also keeping the computational cost minimum. This section shed some light on the algorithms discussed above in HEP experiment's context.

2.5.1 UKF and track fitting

THE UKF outperforms the EKF in terms of accuracy with a comparable computational cost. Besides, the UKF is more robust against non-linearities as compared to the EKF. Therefore, the UKF will be a better choice to cope with the above mentioned issue. The additional advantage of the UKF is that it is easy to implement and is not going to rise the computational cost.

2.5.2 CDKF and track fitting

THE UKF and CDKF belong to the same family and perform similarly in terms of accuracy. Whereas the CDKF, as compared to the UKF, is more complex algorithm as it involves more steps which obviously contribute towards computational cost rise. Therefore, the CDKF is not suitable for the track fitting at HEP experiments.

2.5.3 Particle filters and track fitting

The main issue associated with the particle filters is that they have high computational cost, and obviously is not very suitable for the HEP experiments where millions of particles are needed to be processed. Using particle filter for alignment, which is even worst in terms of computational burden, would make it worst. UPF is even worst choice for track fitting since it is more computationally complex. Although the particle filters might outperform the UKF and the CDKF in terms of accuracy and robustness against non-linearities, the computational cost would be increased substantially.

The above analysis shows that UKF is a more suitable choice for track fitting in HEP experiments since it outperforms EKF in terms of accuracy and has comparable computational cost. Whereas all other approaches will give rise to computational complexity. In the next chapter, UKF will be compared to the EKF in terms of robustness against noise and non-linearities.

2.6 Smoothing

Kalman smoothing has an important role to play in HEP particle tracking experiments as accurate estimates are essential in order to determine the track of a particle in terms of position and momentum over time. It should also be noted that detector alignment using known incidence cosmic rays can be accomplished using either parameter estimation (e.g. non-linear weighted least squares (WLS)) or Kalman filtering. Owing to the inherent non-linearities, the EKF is used as a track re-constructor for particle tracking whereas smoothing is typically achieved using RTS smoother in particle tracking problems [4, 94, 60]. The RTS smoother is implemented using the Jacobian matrices that are naturally available with EKF.

The unscented Kalman filter (UKF) is a relatively new and comparatively efficient form of Kalman filter for non-linear systems, however, since the main benefit of this algorithm is that Jacobian matrices are not computed, the question of how to implement an RTS smoother naturally arises.

Eric. A. Wan and R. van der Merwe [96], has implemented a bi-directional UKF smoother, where firstly it estimates the future states (runs forward) and then estimates the past states (runs backward). The problem with the backward filtering is that it involves the inverse of the dynamic model which can lead to inaccurate results [95]. The unscented Rauch Tung Striebel (URTS) smoother is different from RTS in that sigma points are calculated from the updated estimates and the covariances provided by the filter are used to calculate smoother gains in place of the Jacobian terms.

Although URTS smoother can be implemented directly with UKF, the calculation of sigma points and cross covariances for smoother gains is a computational burden. And keeping in mind the computational issues related to large dimensional systems such as HEP problem, URTS is not a suitable choice. Therefore a computationally efficient smoothing algorithms is needed to be used with the UKF.

2.7 Conclusions

The EKF has widely been used in the non-linear tracking applications and recently sampling based methods has been presented. These methods avoid the linearization step and instead they generate sampling points and propagate them through the actual non-linear system. A family of Kalman filters that incorporate this method is called Sigma Point Kalman Filters (SPKFs) and has been shown to perform better as compared to the EKF in terms of accuracy and robustness against non-linearities in various applications except track fitting at HEP experiments.

Although, the EKF is being used for track fitting for the last forty years within the HEP experiments and is performing reasonably well, there are some inefficiencies that can be resolved by replacing an efficient tracking algorithm. The EKF starts performing poorly and even diverges at very small changes in momentum caused by the internal dynamics of the trackers such as multiple scattering and ionization energy loss. Multiple scattering inside a very strong magnetic field makes track fitting a highly non-linear problem and it becomes very difficult for the EKF to predict such changes reasonably well since it already linearizes the actual non-linear system and the actual system dynamics are not being taken into account. Another problem that makes the EKF to perform poorly is ionization energy loss. The particle loses energy and the momentum is affected. Again, the EKF diverges since it works on approximations and this fact accumulates towards its bad performance while facing such problems caused by the internal dynamics.

Different algorithms have been presented and compared with the EKF in the literature. In a search for a suitable algorithm for track fitting, two challenges should be considered; accuracy and computational complexity. One of the most suitable type of algorithms for non-linear particle tracking are SPKFs and particle filters. The SPKFs are a family of kalman filters that contain two main members, the CDKF and the UKF. These two algorithms perform similar in terms of accuracy, however the UKF is relatively easy to implement due to the fact that the CDKF is quite complex in its formulation. Therefor UKF is a better choice when comparing these two approaches. On the other hand, the particle filters are more efficient and robust as compared to the UKF since they cope with any type of non-linearities and work with both Gaussian and non-Gaussian noises. Undoubtedly, the particle filter would be a better choice for track fitting only if the accuracy had to be increased. However, computational complexity is also a challenge that should not be avoided while choosing a suitable algorithm for track fitting. Particle filters will improve the accuracy without preserving the computational cost within a range that is being currently achieved using the EKF. On the other hand, the UKF provide better accuracy as compared to the EKF while maintaining similar computational cost. Therefore the UKF is a better choice for track fitting since it is computationally more efficient as compared to the particle filters.

Smoothing is an algorithm that is implemented with an estimator to improve the accuracy within track fitting process. Since the EKF is currently being used for track fitting, smoothing is being achieved using the RTS smoother. RTS smoother uses Jacobians computed by the EKF and therefore cannot directly be implemented with the UKF. A smoothing algorithm has to be designed that could efficiently work with the UKF.

Chapter 3

A comparison of the UKF and EKF for particle tracking

The focus of this chapter is, firstly, to explain scintillating fibre trackers used at Muon Ionization Cooling Experiment (MICE) in terms of construction, measurement specifications and uncertainties.

Secondly, before the UKF is implemented to track fitting at MICE, forms of the UKF are compared with the EKF in terms of robustness against non-linearity. As explained in the previous chapter, the extended Kalman filter has widely been used for non-linear state estimation but the unscented Kalman filter [17, 32, 18], is frequently being used due to its advantages over the EKF. The forms of the UKF, augmented and non-augmented UKFs, have previously been analysed [36, 9]. Yuanxin Wu and Dewen Hu [9] investigated these forms of the UKF and concluded that the augmented form is more accurate as compared to the non-augmented form, whereas Fuming Sun, Guanglin Li and Jingli Wang [36] described that augmented UKF does not always have preference over the non-augmented UKF and that their performance is affected by the level of noise. However, these algorithms have never been compared with the EKF in terms of robustness against non-linearity. Therefore we compare the performance of the forms of the UKF in terms of robustness against non-linearities with the EKF. The Van der Pol oscillator, a small dimensional problem, is chosen for the comparison because this will be helpful in investigating and selecting a suitable algorithm for the particle tracking (track fitting) problem. Although this system does not replicate the particle tracking system but provides a platform for these algorithms to be tested at different non-linearity levels.

Finally, the EKF is compared with the UKF in this chapter in terms of accuracy and non-linearity for a particle tracking problem at MICE. Due to the fact that the UKF has never been implemented to particle tracking, it has never been compared with the EKF.

The structure of this chapter is as follows. Section 3.1 briefly explains the muon ionization cooling experiment. The construction of the scintillating fibre trackers is discussed in Section 3.2. Sections 3.3 describes the track reconstruction process at MICE and in Section 3.4, tracker uncertainties are discussed. Section 3.5 provides the mathematical description of the EKF. In Section 3.6, a general overview about the UKF is presented which subsequently covers the mathematical description of its versions, augmented and non-augmented. An approach that requires the computation of fewer sigma points called spherical simplex sigma point Kalman filters (SSSPKF) [10], based on its unscented transformation [33], is presented in Section 3.7. These algorithms are compared in terms of their robustness against measurement and process noise with the help of a simulation. A comparison of these approaches and the results are presented in Section 3.8. The implementation of the UKF to particle tracking within MICE is presented in Section 3.9. The EKF and the UKF are compared in terms of accuracy and robustness against non-linearity for the MICE in Section 3.10. Finally the conclusions can be found in Section 3.11.

3.1 MICE overview

Muon beams, as conventionally produced, occupy a large volume of phase space. A future particle physics facility, the Neutrino Factory, depends on the ability to produce beams of muons along with the cost of the accelerator needed to produce such beams that increases with the volume of phase space occupied be the beam. MICE will use a novel technique to demonstrate that it is possible to achieve a significant reduction in the beam's phase space within the muon's 2 microsecond lifetime. Conventional cooling techniques are far too slow and all the muons would have decayed before there was any reduction in the phase space.

Figure 3.1 shows a schematic of MICE. The upstream instrumentation includes a particle identification system. Particle identification is achieved using time of flight hodoscopes, TOF0 and TOF1, and cherenkov counters to select a pure muon beam. Downstream of the cooling channel, there is a final hodoscope, TOF2, and a calorimeter that allows muon decays to be identified and rejected. The upstream and downstream trackers measure the position and momentum of each particle before and after the cooling channel [37, 38, 39].

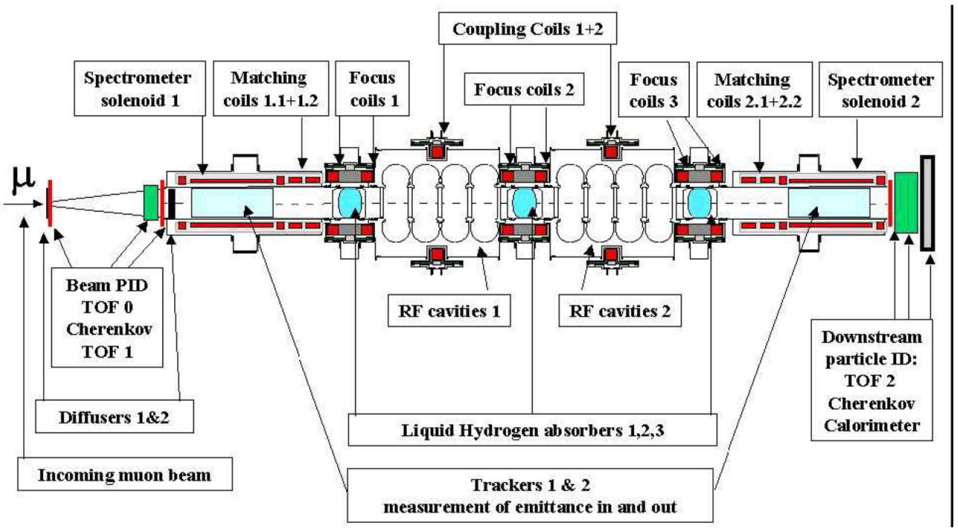


Figure 3.1: Schematic of MICE. Adapted from H. Sakamoto, [11]

3.2 Scintillating fibre tracker

MICE uses two Scintillating fibre trackers that sit inside the spectrometer solenoid, shown in figure 3.2, that provides a uniform magnetic field of 4T [38]. The center coil has a uniform magnetic field volume of 1m length and 30cm diameter. Therefore trackers also cover 1m length and 30cm of diameter. The centre coil can be seen in the cross section of the spectrometer solenoid shown in figure 3.3. Since the muons follow a helical trajectory while traversing trackers due to the uniform magnetic field, the tracker measures the position and momentum in each x, y and z direction. Table 3.1 shows the specifications for the tracker and spectrometer solenoid. The construction of the scintillating fibre tracker is explained in this section.

3.2.1 Scintillating fibre doublets

The scintillating fibres of 350 μm in diameter are glued with 427 μm pitch to make a single layer [39]. Such fibres are used because they provide the position measurement without disturbing the trajectory of the muons. Two layers of sintillating fibres are

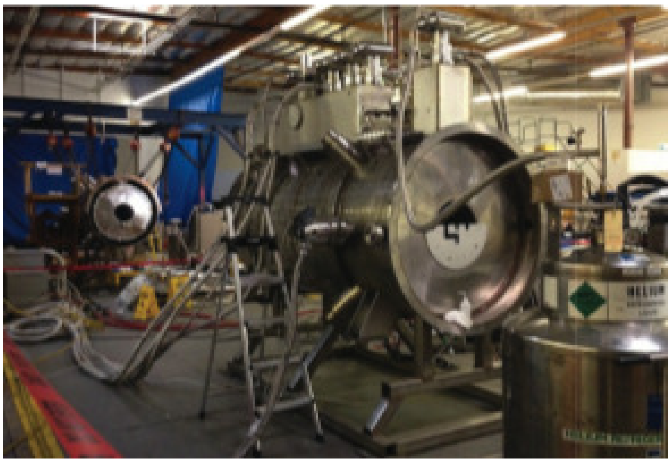


Figure 3.2: MICE spectrometer solenoids magnet. Adapted from D. Rajaram and P. Snopok, [38]

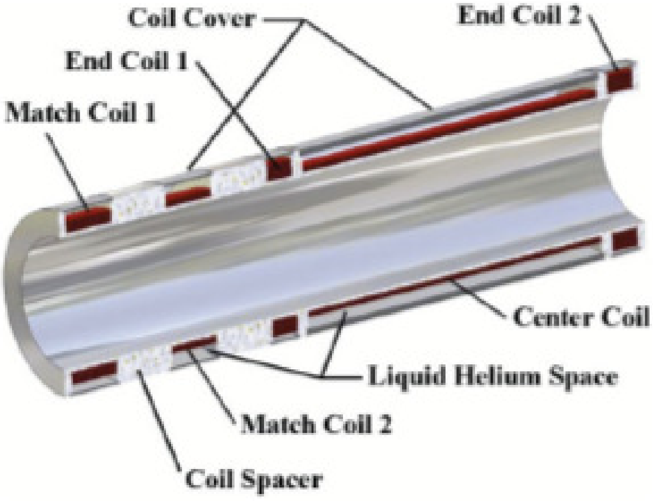


Figure 3.3: Cross section of a spectrometer solenoid magnet. Adapted from D. Rajaram and P. Snopok, [38]

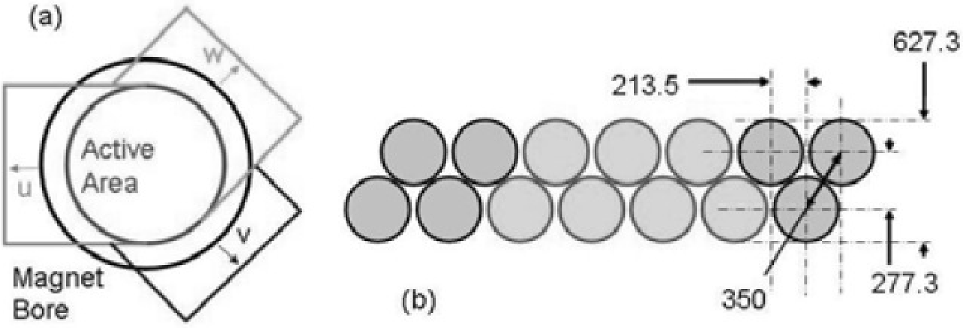


Figure 3.4: (a) Three views per station. (b) Scintillating fibre doublet. Adapted from P. Kyberd, [39]

glued to make a doublet in order to minimise the dead space between fibres. Figure 3.4(b) shows the arrangement of fibres in a doublet.

3.2.2 Scintillating fibre stations

Each tracker consists of five stations as shown in figure 3.5. Each station is made up of carbon fibre frame consisting three scintillating fibre doublets arranged at 120 degrees to each other as shown in figure 3.4(a) [11]. Thus each station provides three views. Light produced by the fibres due to particle hits is transferred through the light guides connected to the optical connectors found on the circumference of each station. A single frame is shown in figure 3.6 where three lines show the arrangement of three fibre doublets.

3.2.3 Light guide system

The light-guide, a bundle of 128 clear fibres, is decomposed into two parts, internal and external light-guide. Internal light-guides connect stations to the patch panel whereas external light-guide connects patch panel and VLPC cassettes [11].

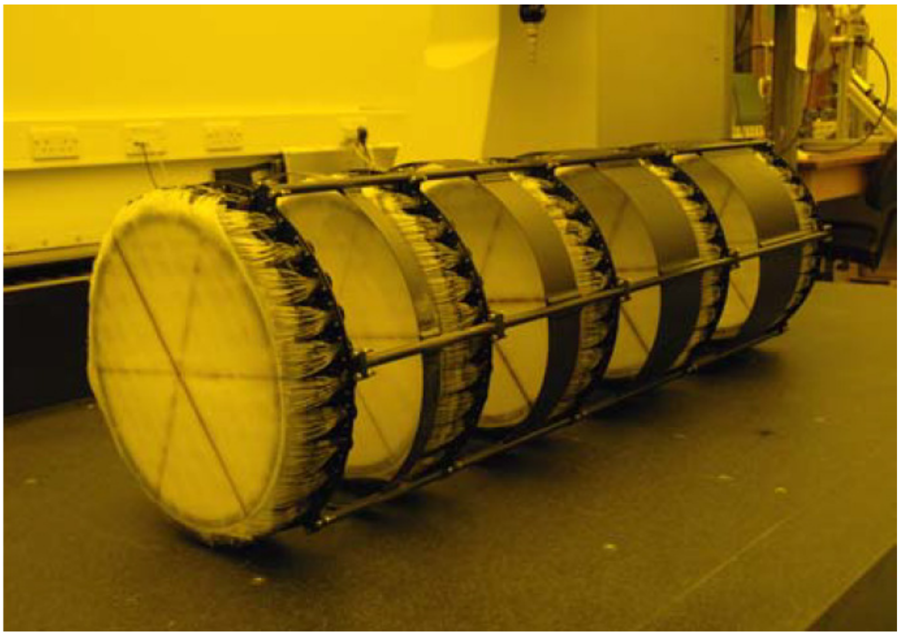


Figure 3.5: Scintillating fibre tracker. Adapted from P. Kyberd, [39]

Component	Parameter	Specification
Scintillating fibre tracker	Scintillating fibre diameter	350 μm
	Fibre pitch	427 μm
	Number of fibres per optical readout channel	7
	Position resolution per plane	470 μm
	Views per station	3
	Stations per tracker	5
Tracking volume	Length	110 cm
	Diameter	30 cm
Spectrometer solenoid	Magnetic field in the tracking volume	4 T
	Tracking volume's field uniformity	3 per mil

Table 3.1: Specicifications of the tracking system

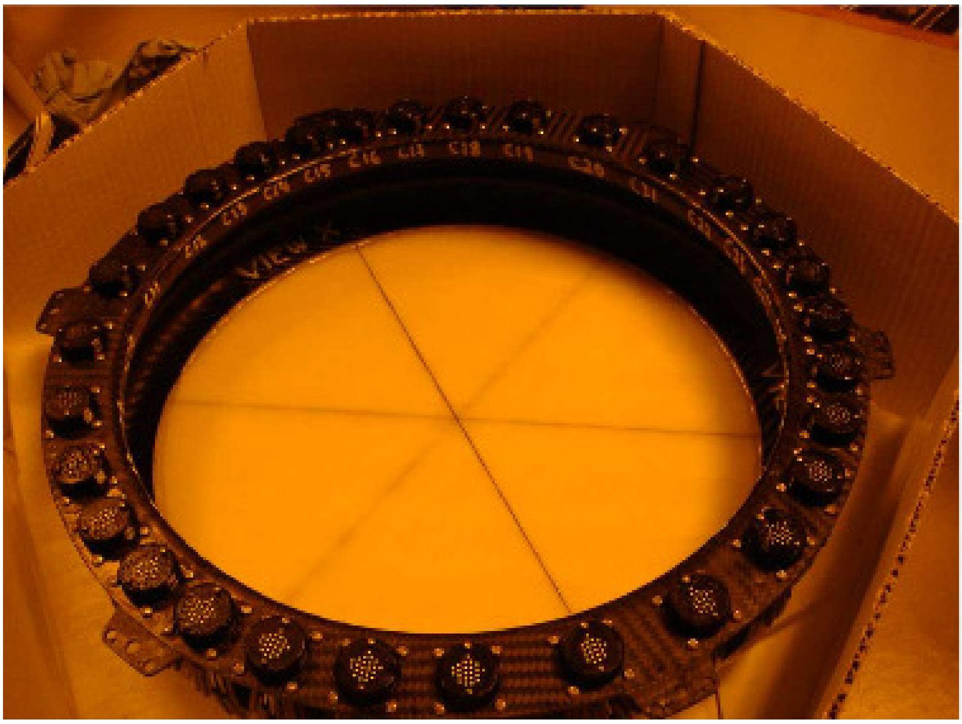


Figure 3.6: Single station where optical connectors can be seen on the circumference. Three back lines seen in the double layer corresponds to center fibers in each views, Adapted from H. Sakamoto, [11]

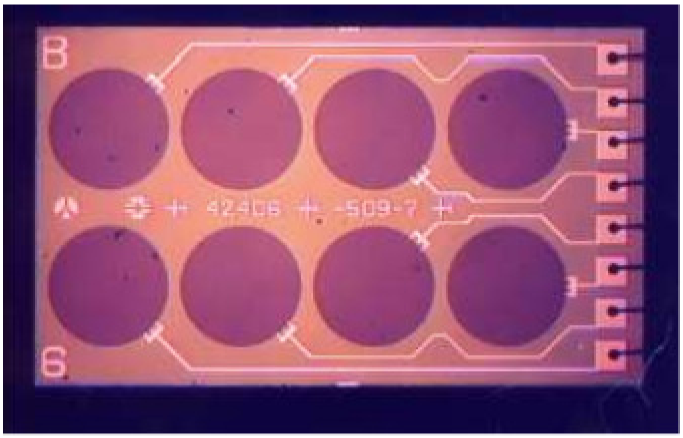


Figure 3.7: Visible light photon counter. Adapted from H. Sakamoto, [11]

3.2.4 Photon detecting system

Although scintillating fibres does not disturb the trajectory of the particle, they produce very little light [39]. To read these low light signals an efficient system was required and thus VLPCs are used. The light signal is transferred through the light guides to VLPCs, shown in figure 3.7, which convert it into an electrical signal. Electrical signal helps in identifying where the hits are and the position resolutuion is 470 μm [38, 39].

3.3 Track reconstruction

3.3.1 Space point construction

Particle hits detected on a given station in one plane make a cluster if the hits are of the same particle [23]. Clusters are made from the hits found on two or three planes. Space points are formed using these clusters that intersect each other in at least two planes from the same station. The space points from distinct stations are analysed in terms of collinearity and the selected points are extrapolated to the next stations. Finally the space points are required by the Kalman filter for track fitting.

3.3.2 Track fitting with Kalman filtering

The Kalman filter requires, along with the group of tracker measurements, the track model, the resolution for each measurement, the geometry of the detector itself and the material effects to correct the noise introduced due to multiple scattering. The space points and the associated tracks are passed to the Kalman filter which produces fits by also taking into account the multiple scattering.

3.4 Uncertainty analysis

Multiple scattering directly affects the trajectory of the particles and is considered as non-Gaussian process noise. Apart from disturbing the position of the particle, it also affects the momentum. However in MICE, the multiple scattering is minimized by selecting the thin fibres and thus the process noise is negligible [39]. Another source of uncertainty that could participate as the process noise is the non-uniformity of the magnetic field strength as it could affect the position and momentum of the particle. This issue is avoided by using the superconducting magnets that provides a uniform magnetic field. The uniform field is due to the geometry and design of the coil, and also the precision with which the coil is wound. On the other hand, the source for measurement noise is the position resolution which is considered as zero mean Gaussian noise.

3.5 Extended Kalman filter

Let a general non-linear system represented by the following discrete time equations:

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

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{w}_k \tag{3.2}$$

where $\mathbf{x}_k \in \mathfrak{R}^n$, and v_k and w_k are process and measurement noises respectively. $\mathbf{u}_{k-1}$ is the input which is the magnetic field in the case of particle tracking. Predicted state vector is calculated as,

$$\hat{\mathbf{x}}_k^- = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \quad \hat{\mathbf{x}}_k^- \in \mathfrak{R}^n \tag{3.3}$$

where $\mathbf{u}_{k-1}$ is the input signal. The predicted covariance is calculated as,

$$\mathbf{P}_{x_k}^- = \mathbf{F}_k \mathbf{P}_{x_{k-1}} \mathbf{F}_k^T + \mathbf{P}_v, \quad \mathbf{P}_{x_k}^- \in \mathfrak{R}^{n \times n} \tag{3.4}$$

The Jacobian matrices (F_k) are the first derivative term in the Taylor expansion of the non-linear function and $\mathbf{P}_v$ is the process noise. Predicted measurements are calculated as,

$$\hat{\mathbf{y}}_k = h(\hat{\mathbf{x}}_k^-), \quad \hat{\mathbf{y}}_k \in \mathfrak{R}^m \tag{3.5}$$

Measurement covariance is calculated as,

$$\mathbf{P}_{y_k} = \mathbf{H}_k \mathbf{P}_{x_k}^- \mathbf{H}_k^T + \mathbf{P}_n, \quad \mathbf{P}_{y_k} \in \mathfrak{R}^{m \times m} \tag{3.6}$$

where $\mathbf{H}_k$ are calculated by linearising the non-linear measurement function and $\mathbf{P}_n$ is the measurement noise covariance. Finally the Kalman gain, state estimates and update covariance are calculated using equations below.

$$\mathbf{K}_k = \mathbf{P}_{x_k}^- \mathbf{H}_k^T (\mathbf{P}_{y_k})^{-1}, \qquad \mathbf{K}_k \in \mathfrak{R}^{n \times m} \tag{3.7}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k), \qquad \hat{\mathbf{x}}_k \in \mathfrak{R}^n \tag{3.8}$$

$$\mathbf{P}_{x_k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{x_k}^-, \qquad \mathbf{P}_{x_k} \in \mathfrak{R}^{n \times n} \tag{3.9}$$

3.6 Unscented transformation

In unscented Kalman filtering, sampling points are calculated to completely capture the mean and covariance of the Gaussian random variable. These sampling points are called sigma points because of their calculation which is described below in this section. Sigma points are propagated through the actual non-linear process and measurement functions. The following non-linear function "h" provides the mapping between the states (**x**) and the measurements (**y**).

$$\mathbf{y} = h(\mathbf{x}) \tag{3.10}$$

The non-linear function is applied to each point in turn to yield a cloud of transformed points.

3.6.1 Non-augmented UKF

Let a general non-linear system represented by the following discrete time equations:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1} + \mathbf{u}_{k-1}) + \mathbf{v}_k, \qquad \mathbf{x}_k \in \Re^n \tag{3.11}$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{w}_k, \qquad \mathbf{y}_k \in \Re^m \tag{3.12}$$

where $\mathbf{v}_k$ and $\mathbf{w}_k$ are process and measurement noise vectors respectively. $\mathbf{u}_{k-1}$ is the input which is the magnetic field in the case of particle tracking. It is worth mentioning here that the notation used for UKF are taken from [34]. The n-dimensional random variable $\mathbf{x}$ with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}} \in \Re^{n\times n}$ is approximated by $2n+1$ weighted points given by [17]:

$$\begin{aligned}
\mathbf{X}_{i,k-1} &= \hat{\mathbf{x}}_{k-1}, && i = 0, \\
&= \hat{\mathbf{x}}_{k-1} + \gamma \mathbf{S}_{i,k-1}, && i = 1, \ldots, n, \\
&= \hat{\mathbf{x}}_{k-1} - \gamma \mathbf{S}_{i,k-1}, && i = n+1, \ldots, 2n,
\end{aligned} \tag{3.13}$$

where $\mathbf{S}_{i,k-1}$ is the i^{th} column of the matrix $\mathbf{S}_{k-1} = \sqrt{\mathbf{P_{x_{k-1}}}}$ and $\gamma = \sqrt{n+\lambda}$, where $\lambda = \alpha^2(n+\kappa) - n$ and α and κ are tuning parameters. These tuning parameters allow scaling of sigma points towards or away from the origin. One must choose $\kappa > 0$, to guarantee the positive definiteness of the covariance matrix. And α should be a value between 0 and 1 to avoid non-local effects when the system is highly non-linear. For more explanation about these parameters, refer to [22]. The sigma points have weights assigned by

$$\begin{aligned} w_m^0 &= \frac{\lambda}{n+\lambda}, \\ w_c^0 &= \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \\ w_m^i = w_c^i &= \frac{\lambda}{2(n+\lambda)} \end{aligned} \tag{3.14}$$

w_m is used to reconstruct the mean and w_c the covariance. β is a non-negative weighting parameter which affects the weighting of the 0^{th} sigma point (the mean) for the calculation of the covariance. β can be used to incorporate knowledge of the higher order moments of the distribution. In the case of Gaussian distribution, $\beta = 2$ is optimal [35]. The time update is calculated as:

$$\mathbf{X}_{i,k|k-1} = f(\mathbf{X}_{i,k-1}, \mathbf{u}_{k-1}), \qquad \mathbf{X}_{i,k|k-1} \in \Re^n \tag{3.15}$$

$$\hat{\mathbf{x}}_k^- = \Sigma_{i=0}^{2n}(w_m^i \mathbf{X}_{i,k|k-1}) \tag{3.16}$$

$$\mathbf{P}_{x_k}^- = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T) + \mathbf{P}_v, \qquad \mathbf{P}_{x_k}^- \in \Re^{n \times n} \tag{3.17}$$

where $\mathbf{P}_v$ is the process noise covariance. At this stage, sigma points are calculated again, using $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_{x_k}^-$, to incorporate the effects of process noise.

Map each point through the measurement function to yield the set of transformed sigma points as,

$$\mathbf{Y}_{i,k|k-1} = h(\mathbf{X}_{i,k|k-1}), \qquad \mathbf{Y}_{i,k|k-1} \in \Re^m \tag{3.18}$$

where superscript $m \geq n$ is the number of measurements. The mean is given by the weighted average of transformed points,

$$\hat{\mathbf{y}}_k = \Sigma_{i=0}^{2n}(w_m^i \mathbf{Y}_{i,k|k-1}) \tag{3.19}$$

And the measurement covariance is the weighted outer product of the transformed points,

$$\mathbf{P}_{y_k} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)^T) + \mathbf{P}_n, \quad \mathbf{P}_{y_k} \in \Re^{m\times m} \tag{3.20}$$

where $\mathbf{P}_n$ is the (diagonal) measurement noise covariance matrix. The cross covariance is calculated as,

$$\mathbf{P}_{x_k y_k} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)^T), \quad \mathbf{P}_{x_k y_k} \in \Re^{n\times m} \tag{3.21}$$

And finally the Kalman gain, UKF estimates and updated covariance are calculated using equations below.

$$\mathbf{K}_k = \mathbf{P}_{x_k y_k}\mathbf{P}_{y_k}^{-1}, \qquad \mathbf{K}_k \in \Re^{n\times m} \tag{3.22}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k), \qquad \hat{\mathbf{x}}_k \in \Re^{n} \tag{3.23}$$

$$\mathbf{P}_{x_k} = \mathbf{P}_{x_k}^- - \mathbf{K}_k\mathbf{P}_{y_k}\mathbf{K}_k^T, \qquad \mathbf{P}_{x_k} \in \Re^{n\times n} \tag{3.24}$$

3.6.2 Augmented UKF

Let a general nonlinear system represented by the following discrete time equations:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{u}_{k-1}) \tag{3.25}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{w}_k) \tag{3.26}$$

where $\mathbf{x}_k \in \Re^{n_x}$, $\mathbf{v}_k \in \Re^{n_v}$ is the process noise vector and $\mathbf{w}_k \in \Re^{n_w}$ is the measurement noise vector. $\mathbf{u}_{k-1}$ is the input which is the magnetic field in the case of particle tracking. The superscripts n_x, n_v and n_w are the dimension of the state, process noise and measurement noise vectors respectively.

The augmented state vector and covariance at step k are structured as,

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{v}_k \\ \mathbf{w}_k \end{bmatrix} \in \Re^N \tag{3.27}$$

$$\mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_x & 0 & 0 \\ 0 & \mathbf{P}_v & 0 \\ 0 & 0 & \mathbf{P}_w \end{bmatrix} \in \Re^{N\times N} \tag{3.28}$$

where superscript "a" in (3.27) and (3.28) is an abbreviation of augmented and $N = n_x + n_v + n_w$ is the dimention of the augmented state vector. It should be noted here that $2N + 1$ sigma points are required as compared to the non-augmented form where only $2n + 1$ sigma points are computed. Sigma points are calculated using (3.13) and the sigma point vector is structured as,

$$\mathbf{X}_{i,k-1}^a = \begin{bmatrix} \mathbf{X}_{i,k-1}^x \\ \mathbf{X}_{i,k-1}^v \\ \mathbf{X}_{i,k-1}^w \end{bmatrix} \tag{3.29}$$

where $\mathbf{X}_{i,k-1}^x$, $\mathbf{X}_{i,k-1}^v$ and $\mathbf{X}_{i,k-1}^w$ are the sigma points belong to states, process noise and measurement noise respectively. Time update is calculated as,

$$\mathbf{X}_{i,k|k-1}^x = f(\mathbf{X}_{i,k-1}^x, \mathbf{X}_{i,k-1}^v, \mathbf{u}_{k-1}), \qquad \mathbf{X}_{i,k|k-1}^x \in \Re^{n_x} \tag{3.30}$$

$$\hat{\mathbf{x}}_k^- = \Sigma_{i=0}^{2n}(w_m^i \mathbf{X}_{i,k|k-1}^x) \tag{3.31}$$

$$\mathbf{P}_{x_k}^- = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-)(X_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-)^T), \quad \mathbf{P}_{x_k}^- \in \Re^{n_x \times n_x} \tag{3.32}$$

Sigma points are propagated through the measurement function to yield the set of transformed sigma points as,

$$\mathbf{Y}_{i,k|k-1} = h(\mathbf{X}_{i,k|k-1}^x, \mathbf{X}_{i,k|k-1}^w), \quad \mathbf{Y}_{i,k|k-1} \in \Re^{n_w} \tag{3.33}$$

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

$$\hat{\mathbf{y}}_k = \Sigma_{i=0}^{2n}(w_m^i \mathbf{Y}_{i,k|k-1}) \tag{3.34}$$

And the measurement covariance is the weighted outer product of the transformed points,

$$\mathbf{P}_{y_k} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)^T), \quad \mathbf{P}_{y_k} \in \Re^{n_w \times n_w} \tag{3.35}$$

The cross covariance is calculated as,

$$\mathbf{P}_{x_k y_k} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k)^T), \quad \mathbf{P}_{x_k y_k} \in \Re^{n_x \times n_w} \tag{3.36}$$

And finally the Kalman gain, UKF estimates and updated covariance are calculated using (3.22), (3.23) and (3.24) respectively.

3.7 Spherical simplex unscented Kalman filter

SS-UKF is a relatively new approach of the unscented transformation that allows the computation of fewer sigma points as compared to the general UKF. Like the UKF, SS-UKF can also be implemented in augmented and non-augmented forms.

3.7.1 SS-UKF additive noise case

In SS-UKF formulation, only $n+2$ sigma points are required for a system described by (3.11) and (3.12). The sigma points are calculated by:

$$\mathbf{X}_{i,k-1} = \hat{\mathbf{x}}_{k-1} + \sqrt{\mathbf{P}_{k-1}}\mathbf{Z}_i, \quad i = 0, \ldots, n+1, \quad \mathbf{X}_{i,k-1} \in \Re^n \tag{3.37}$$

The weights are calculated by following the steps below,

1) Choose the value of W_0 between 0 and 1

2) The rest of weights are chosen as;

$$W_i = \frac{1 - W_0}{n+1}$$

3) The scaling parameters are used to control sigma point spread

4)

$$w_i = 1 + \frac{W_0 - 1}{\alpha^2}, \qquad i = 0 \tag{3.38}$$

$$= \frac{W_i}{\alpha^2}, \qquad i \neq 1 \tag{3.39}$$

In order to incorporate higher order information and minimize higher order errors, another parameter β is included in the weights.

$$\begin{aligned} w_m^0 &= w_0, \\ w_c^0 &= w_0 + (1 - \alpha^2 + \beta), \\ w_m^i &= w_c^i = w_i \end{aligned} \tag{3.40}$$

where w_m and w_c are used to reconstruct the mean and covariance respectively.

5) The vector sequence z_i is initialized as;

$$\begin{aligned} z_0^1 &= [0], \\ z_1^1 &= \left[\frac{-1}{\sqrt{2w_1}}\right], \\ z_2^1 &= \left[\frac{1}{\sqrt{2w_1}}\right] \end{aligned}$$

6) The sequence is expanded for $j = 2, \ldots, n$ as;

$$\begin{aligned} \mathbf{Z}_i^j &= \begin{bmatrix} z_0^{j-1} \\ 0 \end{bmatrix}, && i = 0 \\ &= \begin{bmatrix} z_i^{j-1} \\ \frac{-1}{\sqrt{j(j+1)w_1}} \end{bmatrix}, && i = 1, \ldots, j, \\ &= \begin{bmatrix} 0_{j-1} \\ \frac{j}{j(j+1)w_1} \end{bmatrix}, && i = j + 1 \end{aligned}$$

The prediction calculations can be calculated by (3.15)-(3.20). The cross covariance can be calculated by (3.21) and Kalman gain by (3.22). And finally the standard Kalman update equations are used to calculate the estimate and its covariance described by (3.23) and (3.24).

3.7.2 Augmented SS-UKF

For an n-dimensional system represented by (3.25) and (3.26), $N+2$ sigma points are required. Sigma points are calculated as,

$$\mathbf{X}^a_{i,k-1} = \hat{\mathbf{x}}^a_{k-1} + \sqrt{\mathbf{P}^a_{k-1}}\mathbf{Z}_i, \quad i = 0, \ldots, n+1, \quad \mathbf{X}^a_{i,k-1} \in \Re^N \tag{3.41}$$

The vector $\mathbf{Z}_i$ is the ith column of spherical simplex sigma point matrix and their calculation procedure is given in the section above. These sigma points are transformed through the non-linear process function, as described by (3.30). The predicted state and its covariance is calculated using (3.31) and (3.32) respectively. Then the sigma points are transformed through the measurement model described by (3.33). The mean and measurement covariance are calculated using (3.34) and (3.35). The cross covariance is calculated using (3.36). The Kalman gain is calculated by (3.22), and SS-UKF estimate and its covariance are calculated using standard Kalman filter update equations represented by (3.23) and (3.24).

3.8 Simulation

The EKF, augmented, non-augmented and spherical simplex UKFs were implemented on a 2-dimensional non-linear system called Van der Pol (VdP) oscillator. The purpose, firstly, is to test the EKF and forms of the UKF under different levels of non-linearity, and secondly to analyse the forms of the UKF in terms of accuracy and computational complexity. This analysis will help in selecting the suitable form for track fitting, comparatively large dimensional problem, at MICE. VdP oscillator has a non-linearity parameter "μ" that can be varied to change the non-linearity of the system. The state and measurement equations for VdP oscillator are,

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - x_1 + \mathbf{v}_k \qquad (3.42) \\ \mathbf{y} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \mathbf{w}_k \qquad (3.43) \end{aligned}$$

where $\mathbf{v}_k$ and $\mathbf{w}_k$ are the process and measurement noise respectively.

3.8.1 Problem formulation

The scaling parameters were chosen as, $\kappa = 0$, $\alpha = 0.7$ and $\beta = 2$. A sampling interval of 0.1 is selected to discretize the system. The initial state estimate and covariance is chosen as,

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{P}_{x_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The process and measurement noise covariances are chosen as,

$$\mathbf{P}_v = \mathbf{P}_n = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}$$

Since the performance of the augmented UKF, non-augmented UKF and SS-UKF is being compared in terms of their robustness against process and measurement noises, the uncertainty is worth explaining here. In the following analysis weak and strong noise terms are used, where weak and strong additive noises are $N(0,1)$ and $N(0,5)$ respectively.

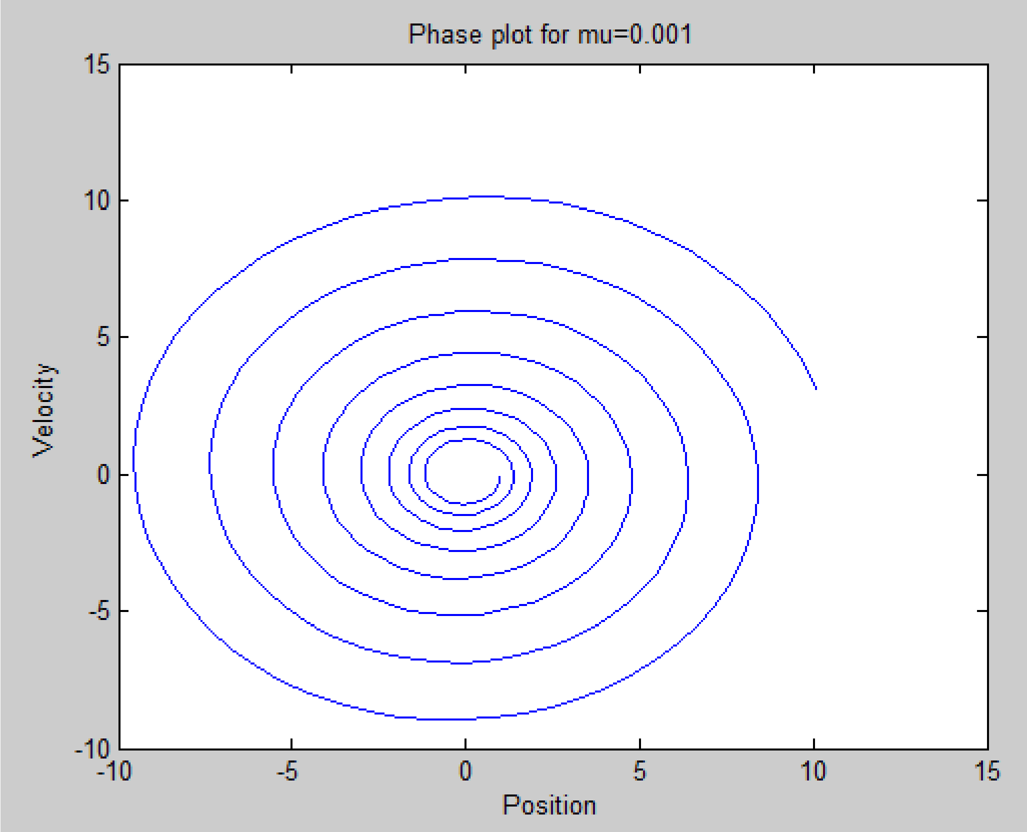


Figure 3.8: Van der Pol phase plot at $\mu = 0.001$

3.8.2 Comparison in terms of robustness against non-linearity

The EKF, augmented and non-augmented UKFs and SS-UKF have been tested by varying the non-linearity parameter "μ" in (3.42). Figures 3.8-3.16 show the phase space diagrams of VdP oscillator plotted for different values of "μ". It should be noted here that the behaviour of the system changes as the value of "μ" changes. In fact as "μ" increases, the system becomes more discontinuous especially after $\mu = 1$.

The tests initially started with $\mu = 0.001$ and gradually increased up to $\mu = 3.8$. For each value of μ the MSEs were recorded for these algorithms in table 3.2 and 3.3 for position and velocity respectivly. It is clear from these tables that EKF is affected badly in terms of accuracy between $\mu = 0.001 - 0.2$ and eventually diverges at $\mu = 0.25$

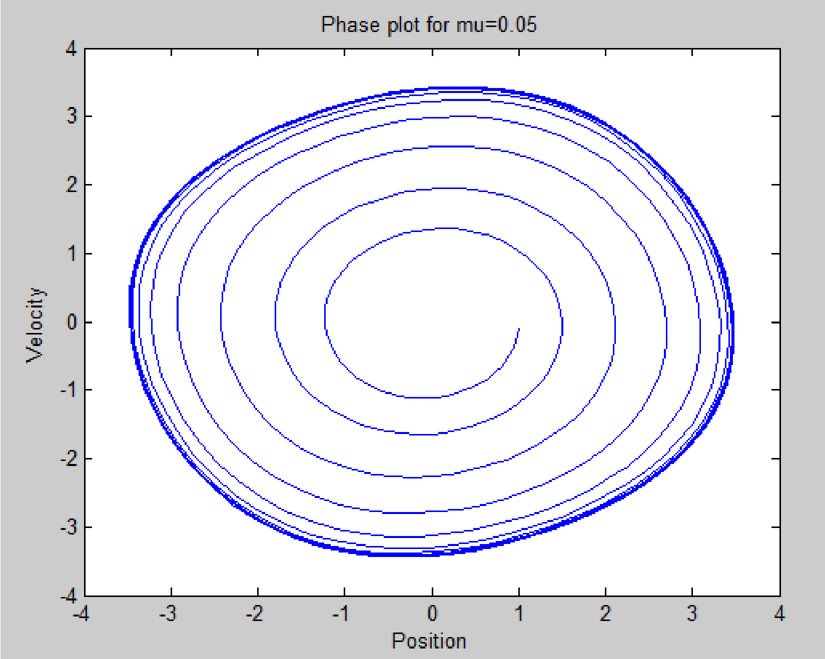


Figure 3.9: Van der Pol phase plot at $\mu = 0.05$

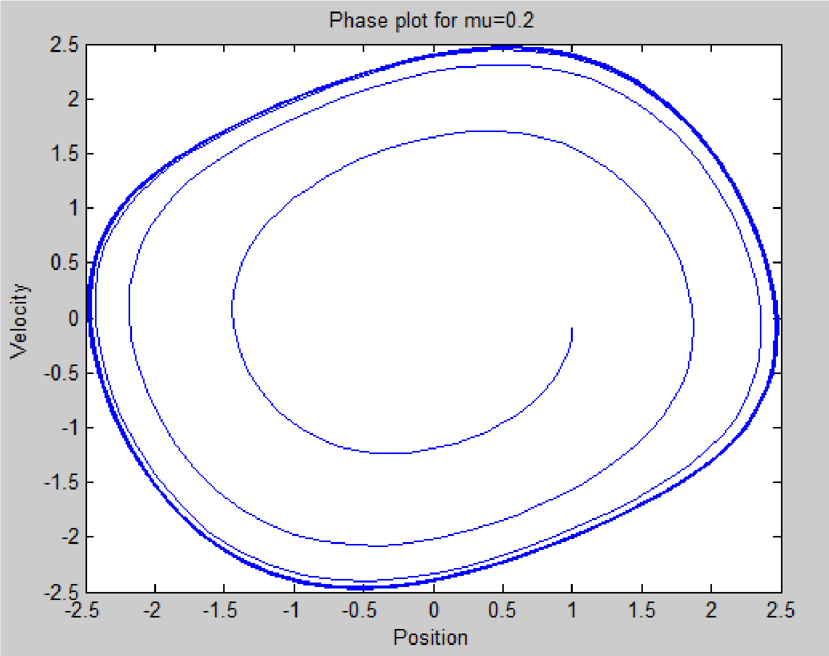


Figure 3.10: Van der Pol phase plot at $\mu = 0.2$

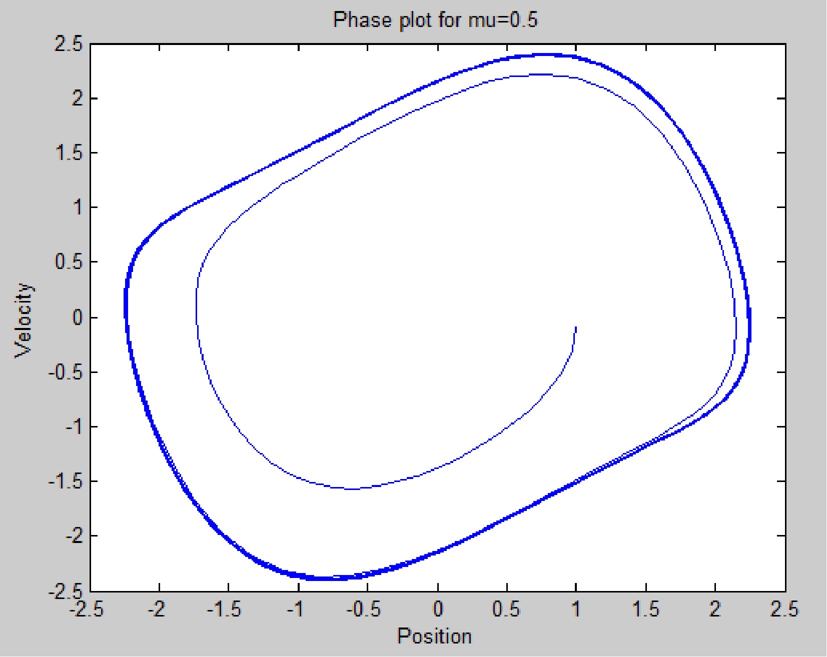


Figure 3.11: Van der Pol phase plot at $\mu = 0.5$

whereas the forms of the UKF are robust until the system becomes highly non-linear ($\mu \geq 2.5$). The SS-UKF, the non-augmented UKF and the augmented UKF diverge at $\mu = 3$, $\mu = 3.3$ and $\mu = 3.8$ respectively.

3.8.3 Comparison in terms of accuracy

The estimates for the algorithms are plotted with the real trajectory for position and velocity in figures 3.17 and 3.18 respectively with guassian noise of $v_k\ N(0,1)$ and $w_k\ N(0,1)$. In figures 3.19 and 3.20, the MSEs are plotted with guassian noise of $v_k\ N(0,5)$ and $w_k\ N(0,1)$.

It can be observed from the figures 3.17 and 3.18 that the non-augmented UKF is more accurate as compared to augmented and spherical simplex UKFs. But in the case of strong noise, the augmented UKF outperforms the other two forms. The strong noise affected the SS-UKF the most, as can be observed from figures 3.19

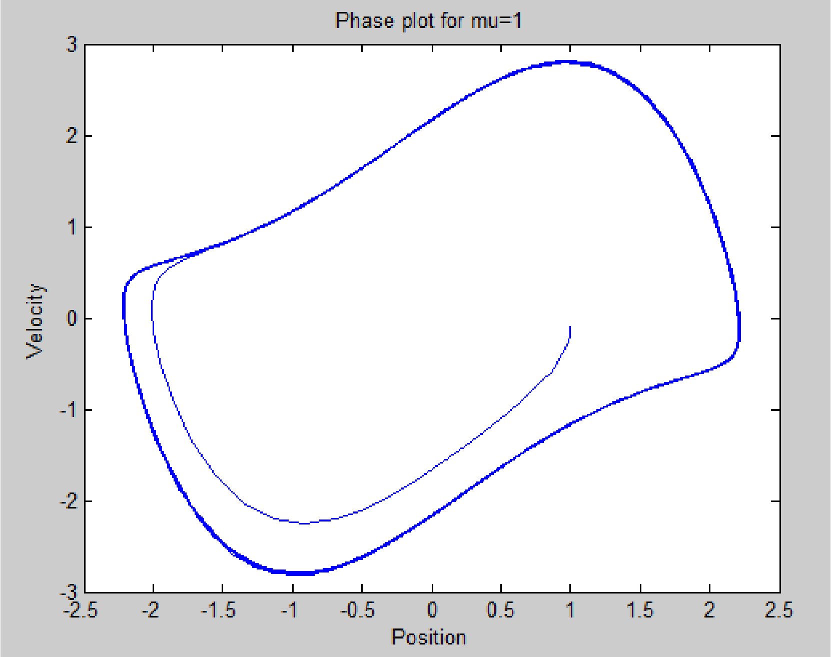


Figure 3.12: Van der Pol phase plot at $\mu = 1$

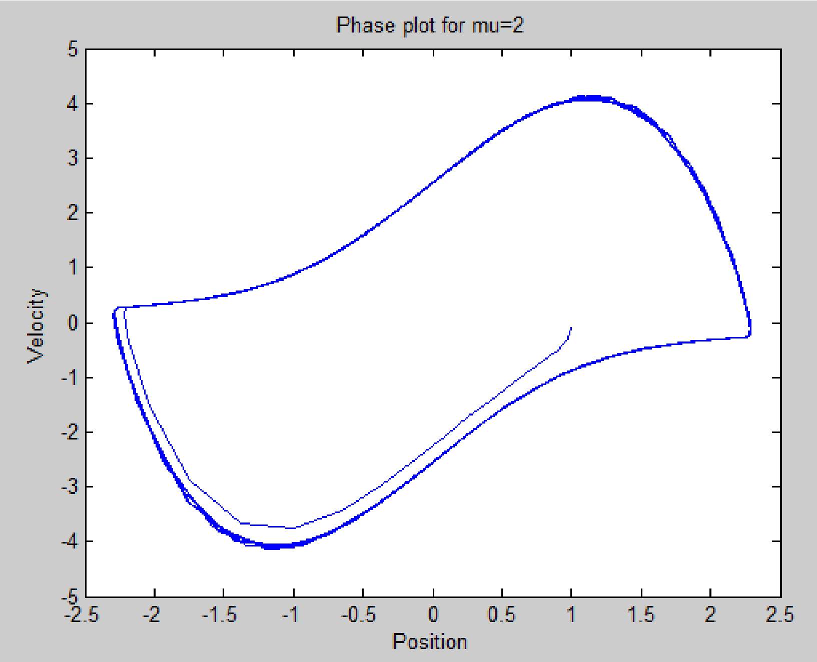


Figure 3.13: Van der Pol phase plot at $\mu = 2$

μ	MSE for Aug UKF	MSE for Non-aug UKF	MSE for SS-UKF	MSE for EKF
0.001	288×10^{-9}	219×10^{-9}	846×10^{-6}	188×10^{-3}
0.01	374×10^{-9}	281×10^{-9}	856×10^{-6}	1.284
0.05	438×10^{-9}	299×10^{-9}	866×10^{-6}	1.758
0.1	399×10^{-9}	323×10^{-9}	861×10^{-6}	2.650
0.15	429×10^{-9}	334×10^{-9}	864×10^{-6}	4.363
0.175	433×10^{-9}	341×10^{-9}	871×10^{-6}	5.048
0.2	424×10^{-9}	344×10^{-9}	857×10^{-6}	5.560
0.22	454×10^{-9}	344×10^{-9}	861×10^{-6}	5.859
0.24	439×10^{-9}	351×10^{-9}	853×10^{-6}	6.113
0.25	439×10^{-9}	351×10^{-9}	853×10^{-6}	*Diverges*
0.5	450×10^{-9}	371×10^{-9}	873×10^{-6}	-
1	470×10^{-9}	481×10^{-9}	883×10^{-6}	-
1.5	483×10^{-9}	576×10^{-9}	959×10^{-6}	-
2	494×10^{-9}	1.06×10^{-6}	3.77×10^{-3}	-
2.5	487×10^{-9}	2.36×10^{-6}	4.91×10^{-3}	-
2.8	486×10^{-9}	5×10^{-6}	7.65×10^{-3}	-
3	484×10^{-9}	18×10^{-6}	*Diverges*	-
3.2	484×10^{-9}	154×10^{-6}	-	-
3.4	497×10^{-9}	*Diverges*	-	-
3.6	3×10^{-6}	-	-	-
3.8	6×10^{-6}	-	-	-
3.9	*Diverges*	-	-	-

Table 3.2: MSEs in position estimates for the augmented UKF, non-augmented UKF, SS-UKF and EKF

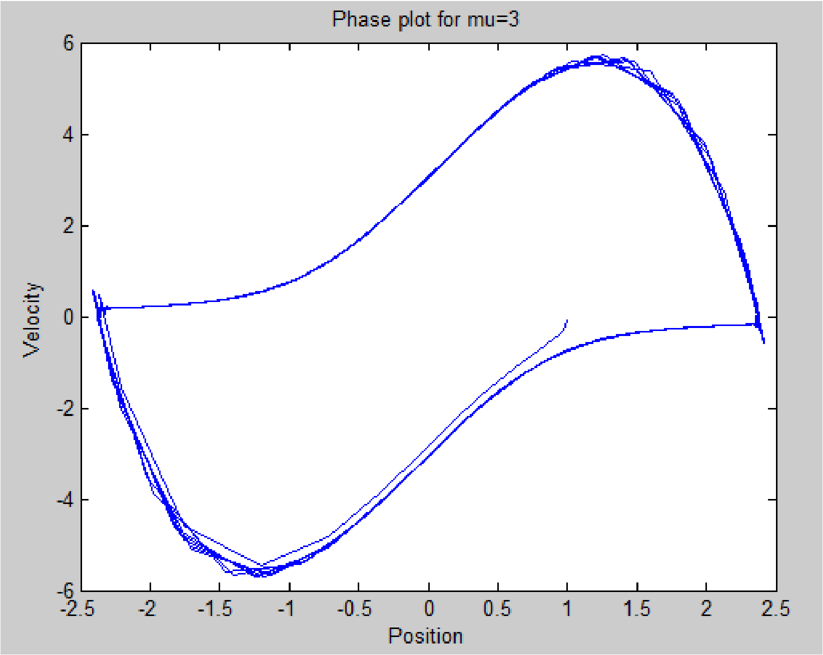


Figure 3.14: Van der Pol phase plot at $\mu = 3$

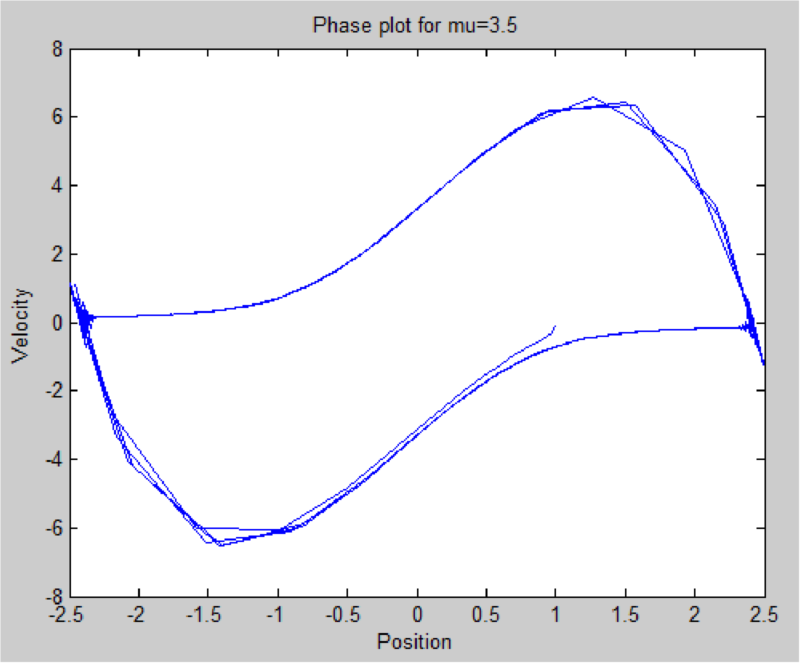


Figure 3.15: Van der Pol phase plot at $\mu = 3.5$

μ	MSE for Aug UKF	MSE for Non-aug UKF	MSE for SS-UKF	MSE for EKF
0.001	256×10^{-9}	271×10^{-9}	73×10^{-6}	132×10^{-3}
0.01	263×10^{-9}	273×10^{-9}	72×10^{-6}	1.032
0.05	260×10^{-9}	277×10^{-9}	74×10^{-6}	1.504
0.1	238×10^{-9}	309×10^{-9}	74×10^{-6}	2.2470
0.15	244×10^{-9}	308×10^{-9}	76×10^{-6}	3.64
0.175	272×10^{-9}	331×10^{-9}	75×10^{-6}	4.228
0.2	266×10^{-9}	292×10^{-9}	78×10^{-6}	4.667
0.22	211×10^{-9}	286×10^{-9}	79.08×10^{-6}	4.924
0.24	204×10^{-9}	316×10^{-9}	80×10^{-6}	5.124
0.25	236×10^{-9}	345×10^{-9}	81.6×10^{-6}	*Diverges*
0.5	310×10^{-9}	335×10^{-9}	104×10^{-6}	-
1	251×10^{-9}	570×10^{-9}	220×10^{-6}	-
1.5	324×10^{-9}	974×10^{-9}	447×10^{-6}	-
2	276×10^{-9}	5.41×10^{-6}	3.6×10^{-3}	-
2.5	261×10^{-9}	35×10^{-6}	6×10^{-3}	-
2.8	267×10^{-9}	98×10^{-6}	7.53×10^{-3}	-
3	292×10^{-9}	289×10^{-6}	*Diverges*	-
3.2	270×10^{-9}	849×10^{-6}	-	-
3.4	497×10^{-9}	*Diverges*	-	-
3.6	7.63×10^{-6}	-	-	-
3.8	13×10^{-6}	-	-	-
3.9	*Diverges*	-	-	-

Table 3.3: MSEs in velocity estimates for the augmented UKF, non-augmented UKF, SS-UKF and EKF

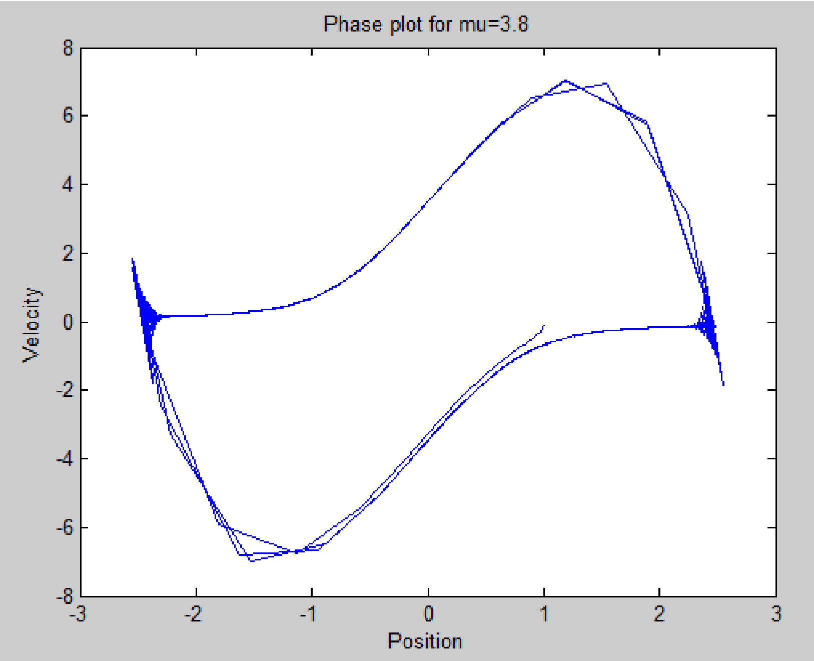


Figure 3.16: Van der Pol phase plot at $\mu = 3.8$

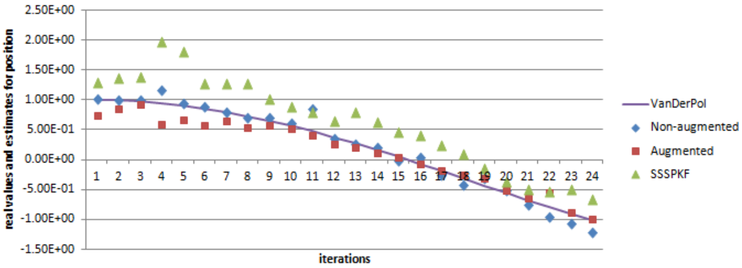


Figure 3.17: Trajectory of real values and estimates for position in the presence of weak noise

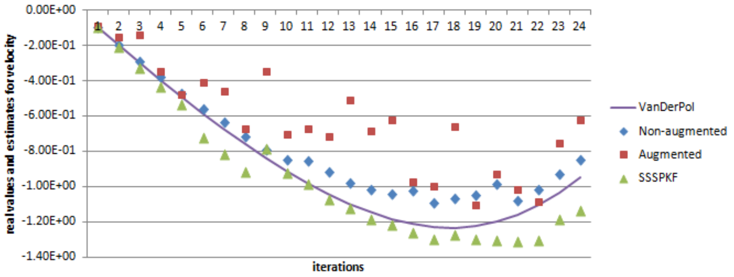


Figure 3.18: Trajectory of real values and estimates for velocity in the presence of weak noise

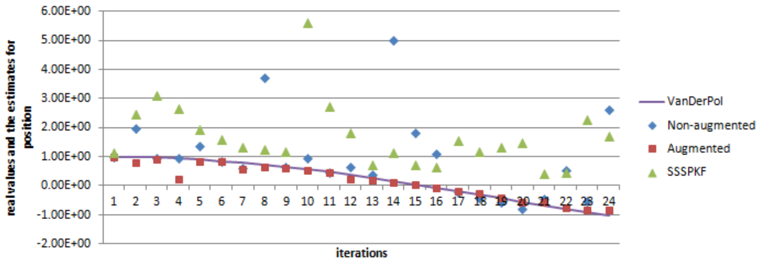


Figure 3.19: Trajectory of real values and estimates for position in the presence of strong noise

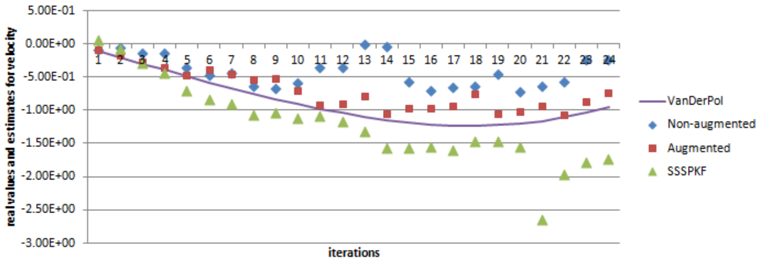


Figure 3.20: Trajectory of real values and estimates for velocity in the presence of strong noise

Algorithm	Average elapsed time
Non-augmented UKF	0.0398 seconds
Augmented UKF	0.0515 seconds
SS-UKF	0.0381 seconds

Table 3.4: Computational time

and 3.20. These algorithms were analysed by varying different noise levels and it was observed that none of these always have preference over the other but they are scenario dependent.

3.8.4 Comparison in terms of computational complexity

Fifteen computer runs were carried out and average elapsed times for the non-augmented, augmented and spherical simplex UKFs are tabulated in table 3.4. SS-UKF is computationally efficient as compared to the other two algorithms since fewer sigma points are calculated and propagated through nonlinear process and measurement functions. Comparing the augmented and non-augmented UKFs, the non-augmented UKF is computationally less expensive because less number of sigma points are calculated and propagated. For a 2-dimensional system, 8, 10 and 13 sigma points are required to be calculated and propagated through the nonlinear functions on each iteration, for SS-UKF, non-augmented UKF and augmented UKF respectively.

In the light of this analysis, the non-augmented UKF is more suitable algorithm for particle tracking because, firstly, the noises are additive and secondly implementing an augmented UKF would increase the computational cost.

3.9 Particle tracking problem and simulation of MICE

EKF is currently being used for track fitting in the MICE experiment and is performing reasonably well [23, 40, 41]. However there is a room for improvement in the process of track fitting which can be achieved by selecting an algorithm that can cope more efficiently with the scintillating fibre tracker wrapped in a very strong magnetic field, a highly non-linear system. It is evident from the previous literature [17, 42] and the comparison made in this chapter that the EKF under-performs in terms of accuracy, as compared to the UKF, when the system is highly non-linear. A detailed analysis on the performance of the UKF and EKF in different levels of non-linearity is provided above in this chapter where it is shown that an increase in non-linearity affects the performance of the EKF and it diverges when non-linearities become severe. However, the EKF and the UKF has never been compared on a particle tracking problem. In this section, firstly the MICE tracking problem is explained with initial conditions. Secondly the implementation of the EKF and UKF to particle tracking on MICE is described where these are compared in terms of accuracy and non-linearity.

The states pertinent to a charged particle in a uniform magnetic field following a helical trajectory may be represented by a 6-dimensional state vector, known as the track vector.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ p_x \\ p_y \\ p_z \end{bmatrix} \tag{3.44}$$

where x, y and z are the coordinates and p_x, p_y and p_z are the quantities of momentum in each x, y and z direction respectively. The track vector is transformed through the track propagator "f_k" that defines how a state vector at surface $k-1$ determines the state vector at surface k.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{v}_k, \qquad \mathbf{x}_k \in \Re^6 \tag{3.45}$$

where $\mathbf{v}_k$ is the process noise. Figure 3.21 shows the direction of the particle through a tracker which clearly demonstrates that the particle traverse the fibre trackers towards z direction whereas the x and y directions represents the helical path. The stations in the diagram measures the moving particle in terms of its position and momentum. These stations are in fact the source of process noise as well since the particle deviates in terms of direction while hitting each station. A detailed analysis of the structure, dynamics and uncertainties for MICE has already been discussed earlier in this chapter, reader is referred to the beginning of this chapter for more explanation.

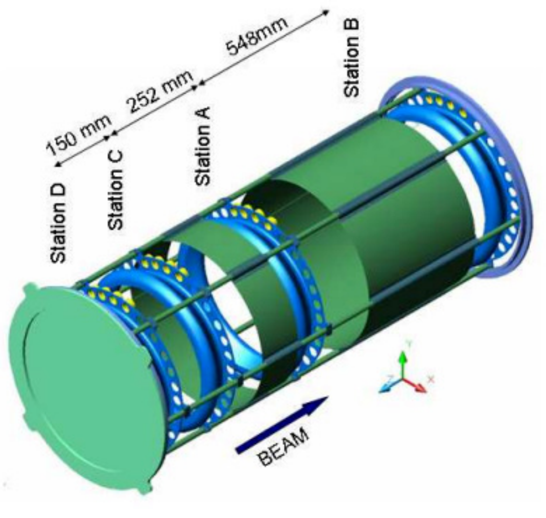


Figure 3.21: Direction of the particle moving through a tracker, Adapted from H. Sakamoto, 2010.

3.9.1 Process model

The track propagator provides a 3-dimensional vector of the position coordinates (x, y and z) and another 3-dimensional vector of momentum. The process function (helix equation) is as below,

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \frac{\gamma}{\rho}(\theta - \sin\theta)\mathbf{b} + \frac{\sin\theta}{\rho}\mathbf{u}_{k-1} + \frac{\alpha}{\rho}(1 - \cos\theta)\mathbf{n}_{k-1} \tag{3.46}$$

$$\mathbf{u}_k = \gamma(1 - \cos\theta)\mathbf{b} + \cos\theta\mathbf{u}_{k-1} + \alpha\sin\theta\mathbf{n}_{k-1} \tag{3.47}$$

Equation (3.46) provides the position (x, y and z) and (3.47) gives momentum (p_x, p_y and p_z). Therefore, track vector represented by the equation (3.44) can also be written as,

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_k \\ \mathbf{u}_k \end{bmatrix} \tag{3.48}$$

b = $\frac{\mathbf{B}}{|\mathbf{B}|}$, where $\mathbf{B}$ is magnetic field strength vector which can be written as, $\mathbf{B} = \left[\, {\scriptstyle B_x, B_y, B_z} \,\right]^T$ and $|\mathbf{B}|$ is the 2-norm. γ and α are $cosine$ and $sine$ of the angles between momentum and magnetic field respectively, where $\alpha = \sqrt{1 - \gamma^2}$. $\rho = -q(\frac{|\mathbf{B}|}{p_0})c$, where ρ is the curvature, q is charge, p_0 is the transverse momentum and c is a constant that is calculated as, $c = 0.3\frac{GeV}{Tm}$. $\theta = \rho l$, where l is the length. $\mathbf{n}_{k-1}$ is the cross product of **b** and $\mathbf{u}_{k-1}$.

3.9.2 Measurement model

The measurement model h_k describes the functional dependence of the measurement vector $\mathbf{y}_k$ in layer k on the state vector at the same layer.

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{w}_k, \qquad \mathbf{y}_k \in \Re^m \tag{3.49}$$

where $\mathbf{w}_k$ is the measurement noise. In the MICE detector, positions and corresponding momenta are measured by scintillating fibre trackers more details of which are in [41].

Visible light photon counters (VLPC's) convert the light, produced by charged particles passing through scintillating fibres, to electrical signals with a position resolution of 470 μm.

3.9.3 Initial setup

The UKF and EKF is used to estimate the track vector. The initial track vector $(\mathbf{x}_k)$ and covariance $(\mathbf{P}_{x_k})$ are chosen according to the initial data provided by MICE as,

$$\mathbf{x}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.2573 \\ -2.8230 \\ 1 \end{bmatrix} \tag{3.50}$$

$$\mathbf{P}_{x_k} = \begin{bmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{bmatrix} \tag{3.51}$$

The charge is set to 1 and the magnetic field strength is 4T hence $B = \left[\,{\scriptstyle 0,0,4}\,\right]^T$. The UKF scaling parameters are chosen as $\alpha = 0.7$, $\kappa = 0$ and $\beta = 2$. The process noise is chosen as $10^-6 \times I_n$, where I_n represents an identity matrix of dimension of the state vector. Measurement noise is 0 mean with a standard deviation of $0.001m$.

The position estimates are plotted in figure 3.22 and the quality of each estimate is shown in table 3.5 for the UKF and EKF, where Mean Squared Error (MSE) for each of the state estimates is given.

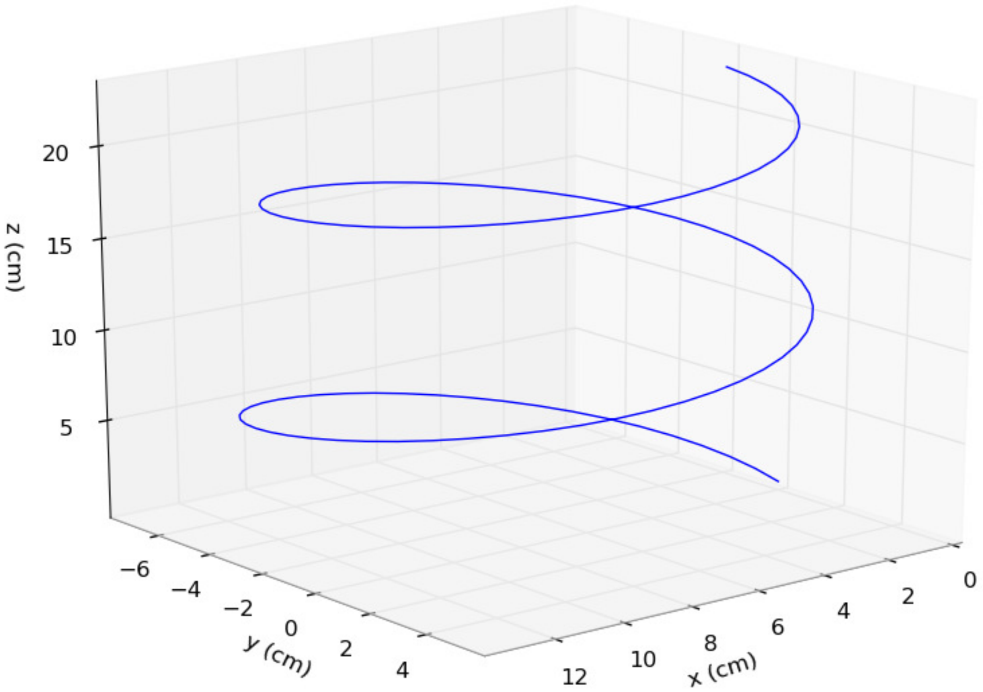


Figure 3.22: Particle moving in a uniform magnetic field

3.10 Comparison of the EKF and UKF for MICE

The EKF and the UKF has been implemented to MICE and are compared in this section. It is clear from the table 3.5 that UKF outperforms the EKF in terms of accuracy. However, it will be more helpful to see how these algorithms behave when the measurement noise is increased. It would also be helpful to analyse these algorithms in terms of robustness against non-linearities in MICE. The experiments has been carried out to shed some light on these question.

System's states	MSE for the UKF	MSE for the EKF
x	$6.74 \times 10^{-7} m$	$3.8 \times 10^{-3} m$
y	$7.93 \times 10^{-7} m$	$0.6 \times 10^{-3} m$
z	$2.27 \times 10^{-7} m$	$4.5 \times 10^{-3} m$
p_x	$7.49 \times 10^{-7} MeV/c$	$4.2 \times 10^{-3} MeV/c$
p_y	$9.33 \times 10^{-7} MeV/c$	$3.9 \times 10^{-3} MeV/c$
p_z	$7.26 \times 10^{-7} MeV/c$	$2.7 \times 10^{-3} MeV/c$

Table 3.5: Mean squared error of each state estimate

3.10.1 Comparison in terms of accuracy

In this section, the effects of increased measurement noise on the performance of the EKF and UKF are investigated. Table 3.6 shows the MSE in the estimates for the EKF and UKF when the measurement noise was increased to 0 mean with a standard deviation of $0.01m$. A better performance and consistancy of the UKF can be observed in this table. The measurement noise was further increased to 0 mean with a standard deviation of $0.1m$ and the results are provided in table 3.7. It is clear from these tables that, although the performance of both the algorithms is affected as the noise is increased, in the case of higher noises the EKF is more likely to diverge.

3.10.2 Comparison in terms of robustness against non-linearity

In this section, the effects of non-linearities on the performance of the EKF and UKF are analysed. As explained above, the main source of non-linearity in the MICE is the process noise (multiple scattering) inside the strong magnetic field. The non-linearity increases as the magnetic field strength increases due to the fact that the curvature of the helix increases with the magnetic field strength and consequently, a small change in the process noise can substantially deviate the path of a charged particle. This drift

System's states	**MSE for the UKF**	**MSE for the EKF**
x	$7.01 \times 10^{-5} m$	$5.18 \times 10^{-2} m$
y	$5.86 \times 10^{-5} m$	$4.6 \times 10^{-2} m$
z	$3.63 \times 10^{-5} m$	$3.7 \times 10^{-2} m$
p_x	$8.41 \times 10^{-5} MeV/c$	$6.1 \times 10^{-2} MeV/c$
p_y	$7.12 \times 10^{-5} MeV/c$	$3.4 \times 10^{-2} MeV/c$
p_z	$6.37 \times 10^{-5} MeV/c$	$3.1 \times 10^{-2} MeV/c$

Table 3.6: Mean squared error of each state estimate for 0 mean with a standard deviation of 0.01 of measurement noise

System's states	**MSE for the UKF**	**MSE for the EKF**
x	$8.07 \times 10^{-4} m$	$8.18 \times 10^{-1} m$
y	$7.46 \times 10^{-4} m$	$6.5 \times 10^{-1} m$
z	$6.83 \times 10^{-4} m$	$5.3 \times 10^{-1} m$
p_x	$8.81 \times 10^{-4} MeV/c$	$7.4 \times 10^{-1} MeV/c$
p_y	$6.82 \times 10^{-4} MeV/c$	$4.8 \times 10^{-1} MeV/c$
p_z	$5.71 \times 10^{-4} MeV/c$	$3.9 \times 10^{-1} MeV/c$

Table 3.7: Mean squared error of each state estimate for 0 mean with a standard deviation of 0.1 of measurement noise

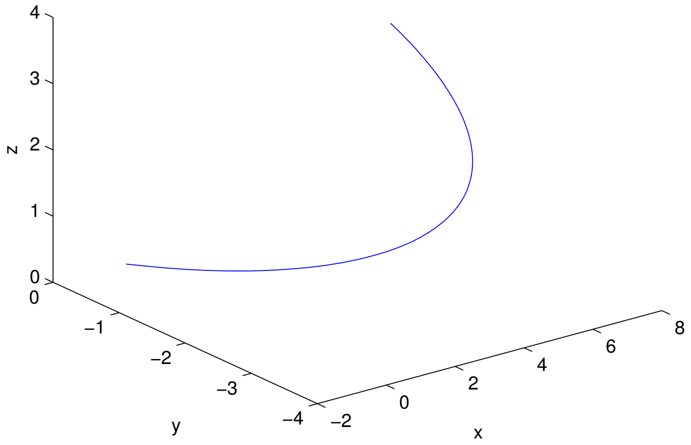


Figure 3.23: Particle moving in a uniform magnetic field of 1T

in the path does not only affect the position but also the momentum of the particle. Figures 3.23-3.25 show that how the curvature of the helix changes as the magnetic field strength increases from $1T$ to $3T$ respectively. It is clear from these figures that the less the magnetic field strength is, the less the path that a particle follows is curved and consequently the less the measurements would be affected. In other words, the deviation in the position and momentum will be less in a relatively weak magnetic field strength as compared to a system with a strong magnetic field strength for the same level of process noise. The non-linearity also increases if the process noise increases within a uniform strong magnetic field.

The EKF and UKF were tested, in terms of robustness against non-linearity, by varying the process noise in the uniform magnetic field strength of $4T$. The process noise was increased to 0 mean with a standard deviation of $0.00001m$ and the results are recorded in table 3.8. It can be observed that a very small change in the process noise has significantly increased the MSE for the EKF and UKF, comparing it with table

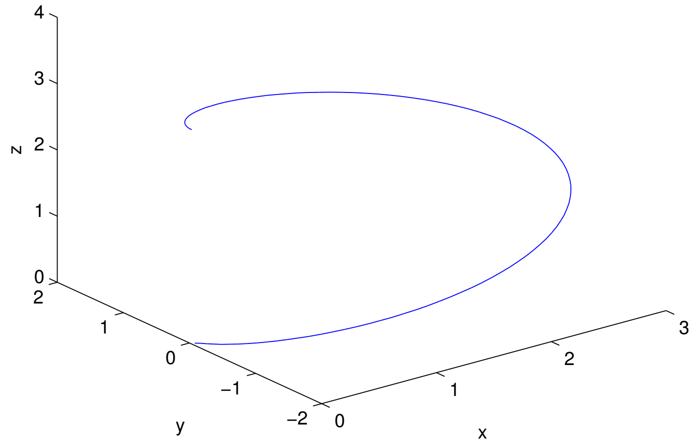


Figure 3.24: Particle moving in a uniform magnetic field of 2T

3.5. The process noise was further increased to 0 mean with a standard deviation of $0.0001m$ and the results recorded in table 3.9 show that a further small change in the process noise again affect the performance of the algorithms badly as is apparent by the MSE. Although the EKF is still trying to estimate but these results are not acceptable in such systems where high level of accuracy is essential. It can also be inferred from these tables that the EKF is less robust against such non-linearities and is more likely to diverge if process noise is further increased. Therefore, the UKF is a better choice for MICE and other particle tracking problems since it is more accurate and robust against non-linearities.

System's states	**MSE for the UKF**	**MSE for the EKF**
x	$5.41 \times 10^{-4} m$	$6.48 \times 10^{-1} m$
y	$6.46 \times 10^{-4} m$	$7.6 \times 10^{-1} m$
z	$4.08 \times 10^{-4} m$	$5.7 \times 10^{-1} m$
p_x	$9.11 \times 10^{-4} MeV/c$	$5.5 \times 10^{-1} MeV/c$
p_y	$5.72 \times 10^{-4} MeV/c$	$4.3 \times 10^{-1} MeV/c$
p_z	$4.82 \times 10^{-4} MeV/c$	$3.6 \times 10^{-1} MeV/c$

Table 3.8: Mean squared error of each state estimate for 0 mean with a standard deviation of 0.00001 of process noise

System's states	**MSE for the UKF**	**MSE for the EKF**
x	$4.47 \times 10^{-2} m$	$3.18 m$
y	$3.16 \times 10^{-2} m$	$3.8 m$
z	$2.82 \times 10^{-2} m$	$2.4 m$
p_x	$5.21 \times 10^{-2} MeV/c$	$4.3 MeV/c$
p_y	$4.22 \times 10^{-2} MeV/c$	$3.4 MeV/c$
p_z	$3.61 \times 10^{-2} MeV/c$	$3.1 MeV/c$

Table 3.9: Mean squared error of each state estimate for 0 mean with a standard deviation of 0.0001 of process noise

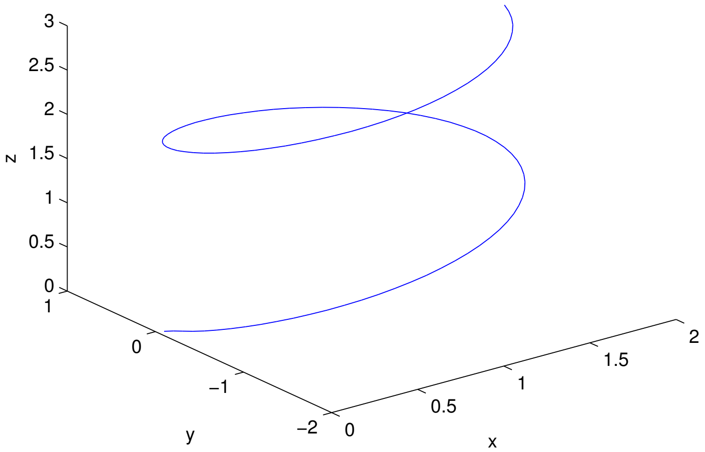


Figure 3.25: Particle moving in a uniform magnetic field of 3T

3.11 Conclusions

There is a trade-off between accuracy and computational complexity when augmented and non-augmented UKFs and SS-UKF are compared. Non-augmented UKF performs better in the presence of weak additive noise, whereas augmented UKF provides better results when the noise is strong. SS-UKF is computationally less expensive as compared to the augmented and non-augmented UKFs, but is less accurate.

The non-augmented UKF is implemented to track fitting problem at MICE, where currently EKF is being used. The UKF is clearly a better choice for the MICE as it has outperformed the EKF in terms of accuracy and robustness against non-linearity. Due to the fact that the EKF does not use the actual non-linear system, it cannot cope well with the uncertainties that occur due to the internal dynamics of the system. The UKF will be analysed in terms of accuracy and relative observability in the next chapter.

Chapter 4

Accuracy and observability analysis of the UKF

In high energy physics (HEP) experiments, as discussed previously in this thesis and in chapter 3, accurate observations are essential in order to determine the momentum of a particle. Trajectories of charged particles are non-linear in a magnetic field owing to the quantum nature of decay and particle interactions. Tracking a single particle can be quite computationally expensive despite the low dimensionality but a particle collider can produce millions of events and each one with hundreds of tracks, so the accuracy and computational efficiency of any algorithm is of paramount importance.

Track fitting (particle tracking) has typically been accomplished using non-linear weighted least squares (WLS), Kalman filtering and extended Kalman filtering (EKF) [4], as is the case for Muon Ionization Cooling Experiment (MICE), is obtained using EKF. It has been explained in chapter 3 that the EKF is less robust against non-linearity and the UKF is a better algorithm for MICE. Therefore in this chapter, UKF is further analysed for MICE in order to better understand the performance of the UKF with particle tracking problem.

The main work of this chapter is to investigate the effects of measurement redundancy on the performance of the UKF, and to analyse the relationship between measurement redundancy and observability. The structure of this chapter is as follows. Measurement redundancy and the performance of the UKF in terms of accuracy is discussed in Section 4.1. Section 4.2 discusses the relationship between measurement redundancy and relative observability where it is shown how a weak state can be observed with a good accuracy. And finally conclusions are presented in Section 4.3.

4.1 Effects of measurement redundancy on the performance of UKF

It is worth mentioning here that the results and analysis presented here are based on the data provided by MICE set-up as described in chapter 3, where an implementation of the UKF to MICE is also provided. Before the effects of measurement redundancy [43] on UKF's performance are discussed, we explain how the UKF accommodates redundancy quite naturally.

4.1.1 UKF and measurement redundancy

Consider a redundant system whose measurement covariance ($\mathbf{P}_{y_k}$) is computed using 3.20, represented by $\mathbf{P}_{y_k} \in \Re^{m \times m}$. The cross covariance ($\mathbf{P}_{x_k y_k}$) is calculated using 3.21 is represented by $\mathbf{P}_{x_k y_k} \in \Re^{n \times m}$. Then computing the Kalman gain using 3.22 gives $\mathbf{K}_k \in \Re^{n \times m}$. Then $\mathbf{K}_k$ is directly substituted in 3.23 and 3.24 to calculate the updated estimates and covariance. It is clear from the above dimensions that measurement redundancy is incorporated quite naturally by the UKF without using projection techniques.

4.1.2 Problem formulation

Two cases are being considered. The first is the fairly trivial but unrealistic addition of one or more measurements of the same quantity (using the same type of sensor). This is unrealistic in the context of the MICE detector because it would require an independent fibre bundle. Nevertheless, of interest here is to understand the effect of redundancy irrespective of practicalities. Essentially, adding a duplicate measurement reduces measurement noise by virtue of its averaging effect.

The second and more realistic case is where we add measurements of otherwise unmeasured variables. In what follows we choose a measurement of the "Time Of Flight" (TOF) of a particle moving through the detector. Such information is available independently of the other measurements.

4.1.3 Case 1: Effects of measured redundant measurements on accuracy

In general, increasing the degree of redundancy further improves the results. But it is not always the case. Effects of different redundant measurements on every single estimate are investigated and it is noted that measurement redundancy doesn't always improve the results. If a specific measurement i is affected by comparatively high noise and has a high residual, $\propto \|y_i - \hat{y}_i\|$, it is not the case that this residual will be reduced by the effect of some other redundant measurement elsewhere. Clearly, measurement noise related residuals associated with measurement i will be reduced if the redundant measurement simply duplicates the affected measurement with a different noise process. To demonstrate this, the effects of measurement redundancy on the accuracy of z measurement are discussed.

red. degree	no_{red}	$z1_{red}$	$z2_{red}$	$z3_{red}$
MSE $\times 10^{-5}$	$5.11m$	$3.48m$	$2.16m$	$1.77m$

Table 4.1: Mean squared error in z measurements

It is worth mentioning here that the initial setup and results for MICE are presented in chapter 3 which are being taken as a reference. A comparatively high noise (0 mean with a standard deviation of $0.01m$) is affecting the z measurements at each time step. Table 4.1 shows the relationship between the MSE in z direction and degree of redundancy. In this table no_{red}, $z1_{red}$, $z2_{red}$ and $z3_{red}$ represent no redundancy, 1, 2 and 3 redundant measurements of z respectively. It is clear from the tabulated results that in the presence of measurement redundancy, a comparatively low mean squared error (MSE) is achieved which is further reduced if more redundant measurements are introduced.

4.1.4 Case 2: Effects of independent redundant measurement "TOF" on accuracy

Time of flight (TOF) is taken as an independent redundant measurement to analyse its effects on the quality of measurements. With the following choice of units, TOF is measured in nanoseconds, and using a relativistic formula for particle velocity, $\mathbf{v} = \mathbf{p}.c/E$ we can calculate the time of flight, along a helical path, between two detectors separated by a distance D, [44], where D is the distance between two detectors, $\mathbf{v}$ and $\mathbf{p}$ are velocity and momentum vectors respectively, E is energy and c the speed of light.

In the MICE experiment, the distance between two successive hits is dominated by the z measurement, whereas the effect of the curved path on the TOF, although much

Degree of redundancy	no_{red}	1_{redTOF}	2_{redTOF}
MSE in $\hat{p}_z$ (MeV/c)	8.6×10^{-4}	1.01×10^{-5}	9.4×10^{-6}

Table 4.2: MSE in pz measurement

smaller, depends on the amount of curvature, which is determined from p_x and p_y.

A comparatively high noise (0 mean with a standard deviation of 0.01MeV/c) is affecting the measurement of p_z at each time step. The effects of TOF on each measurement have been investigated and it is concluded that this independent redundant measurement has smaller effects on all the estimates except $\hat{p}_z$, which is significantly improved. Table 4.2 shows the MSE in $\hat{p}_z$ for the system with and without the redundant measurements of TOF. no_{red}, 1_{redTOF} and 2_{redTOF} represents no redundancy, 1 redundant measurement of TOF and 2 redundant measurements of TOF respectively. It can be seen that the MSE error is low when a measurement of TOF is introduced as a redundant measurement. Another measurement of TOF further reduces the MSE.

4.2 Measurement redundancy and relative observability

A relationship between the measurement redundancy and relative observability is discussed in this section. Again, two cases are being considred. Firstly, in case 1, relative observability in the presence of redundant measurements (directly measured) is discussed. Secondly, in case 2, the effects of an independent redundant measurement (TOF) on the relative observability are discussed. However before we present these results, it is necessary to discuss relative observability in relation with the convergence of an algorithm. The next section briefly discusses relative observability for particle physics problems and its effects on the performance of an algorithm in terms of convergence.

4.2.1 Relative observability and convergence

As discussed previously in this thesis, the main challenge faced in particle tracking is the divergence of the EKF due to multiple scattering and ionization energy loss. Multiple scattering, that affects position and momentum of each particle inside a strong magnetic field, is a non-linear problem and in addition the uncertainty caused by multiple scattering at each sensor accumulates and also causes the EKF to diverge.

On the other hand, ionization energy loss is caused by a particle that loses energy while traversing a tracker. A particle might decay and do not hit the sensor or significantly drift from its actual path due to delays. Consequently, uncertainty caused by ionization energy loss is very difficult to model since the estimator have no data or only inaccurate data available to it.

We define a system being unobservable if a particle decays and does not hit a sensor. On the other hand, we define a system as weakly observable in the case of inaccurate data due to particle delays. Since in the MICE decayed particles that do not hit the detector are of no interest and are discarded, we will be discussing relative observability.

Before the performance of the UKF is analysed for weakly observable states, it is necessary to discuss relative observability theoretically. A system can become weakly observable if the system of equation is ill-conditioned to an extent that computer precision prevents convergence. Weak observability also arises when the measurement uncertainty is large with respect to some measure such as the condition number.

On the other hand, for convergence of an algorithm sufficient data of good quality is required. In the case of insufficient data the issue of unobservability arises whereas a poor quality data makes a system weakly observable. In other words, a system must be observable in order for an estimation algorithm to converge. This implies that observability is a necessary condition for convergence. However, an algorithm might diverge even if the system is practically observable. An algorithm might fail to converge for a system with severe non-linearity and another efficient algorithm might still converge. For instance the UKF still provides reasonable solutions for the systems where the EKF diverged.

The above implies that observability is a property of a system and convergence is a property of an algorithm. Divergence of an algorithm can be avoided by increasing the relative observability of a system. The focus here is to analyse the performance of the UKF in relation with relative observability and redundancy.

4.2.2 Measure of relative observability

The well-known Kalman observability test is a stop/go test but two phenomena make the situation less clear in practice. If an extra measurement is added to a system with observability rank$(\mathcal{O}_b) = n-1$ such as to take $\mathcal{O}_b$ to full rank, the practical value of this depends on the both the signal to noise ratio of that measurement and whether the new hyperplane leads to a system that is not ill-conditioned. The former point relates directly to measurement noise but the latter tends to amplify the effect of process noise and can lead to computational difficulties.

An observable system can become unobservable due to any of its weakly observable states. A state is said to be weak or weakly observable if its corresponding singular value of the Fisher information matrix (FIM) [45] (inverse of the covariance matrix) is comparatively small. The singular values of the Fisher information matrix provide information about the relative observability of the corresponding part of the estimated state. An estimated state associated with a strong (comparatively large) singular value indicates good accuracy whereas an estimated state with a weak corresponding singular value implies that it could become unobservable under certain conditions.

As explained above, there is a strong relationship between the accuracy of an estimate and relative observability. If measurement residuals corresponding to the weak state can be reduced to minimise its difference from the strongly observable state's residuals, then the condition number of the updated covariance and FIM can be improved. Therefore, MSE is being used as a measure of relative observability.

4.2.3 Case 1: Relationship between measured redundant measurements and relative observability

A duplicate measurement of the weak state reduces the measurement residual and helps in estimating the state with good accuracy. And if a redundant measurement (not a duplicate of weak mode) exists in a system, then the residuals of the weak state are not reduced and its observability can't be improved. For example, if the above described tracking system is weakly observable due to (say) weak state x, then an additional measurement of x improves the quality of this measurement by virtue of its noise reduction effects (averaging). But if the duplicate measurement is of y then more weight is given to y instead of x and the measurement residuals are still high for x.

To demonstrate, the relative observability of the weak state p_x is being discussed with different redundant measurements. Figures 4.1, 4.2, 4.3 and 4.4 show the MSE in p_x for each combination of redundant measurements for $0 \leq \rho \leq 1.5$, $\rho = \frac{m}{n}$, where m and n represent the number of measurements and states respectively. In these figures,

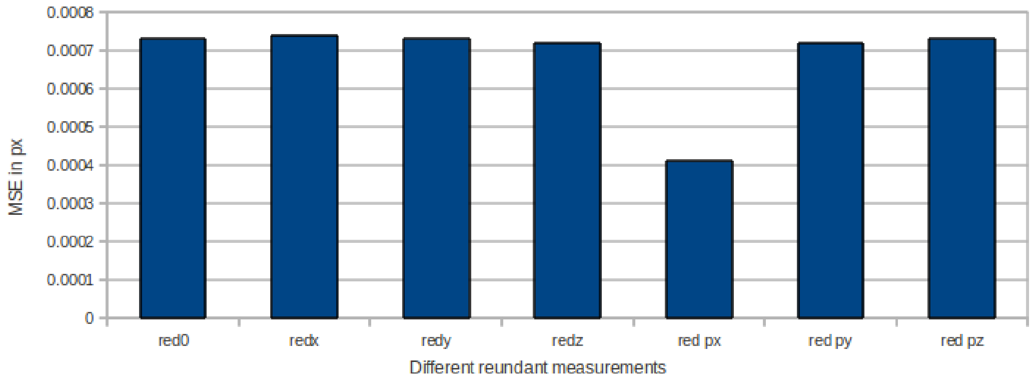


Figure 4.1: MSE in the presence of different redundant measurements

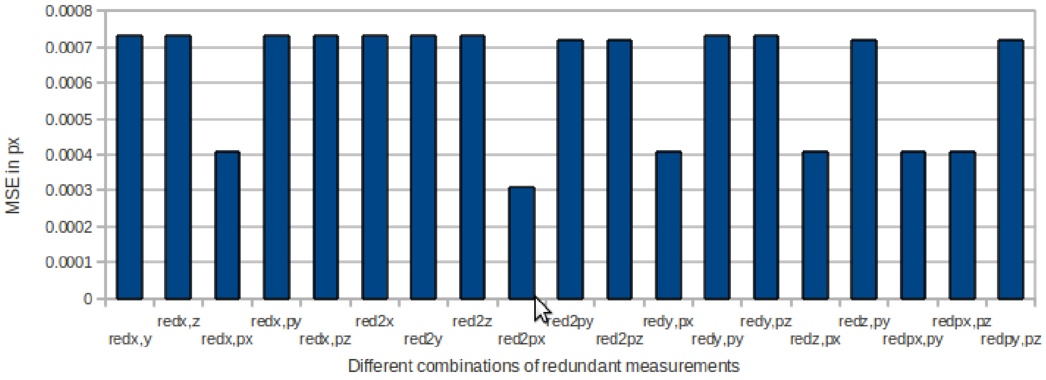


Figure 4.2: MSE in the presence of different combinations of 2 redundant measurement

each entry on x-axis starts with the abbreviation "red" indicating a redundant measurement. The abbreviation "red" is written with different combinations of redundant measurements such as red0, redx,y and red2y representing no redundant measurement; redundant measurements of x and y; and 2 redundant measurements of y, respectively. It is clear from these figures that the MSE is low when the duplicate measurement of p_x or a combination of redundant measurements with duplicate P_x are introduced.

Figure 4.1 shows the MSE in the measurement p_x for different redundant measurements. A lower MSE can be observed when the duplicate measurement of p_x is

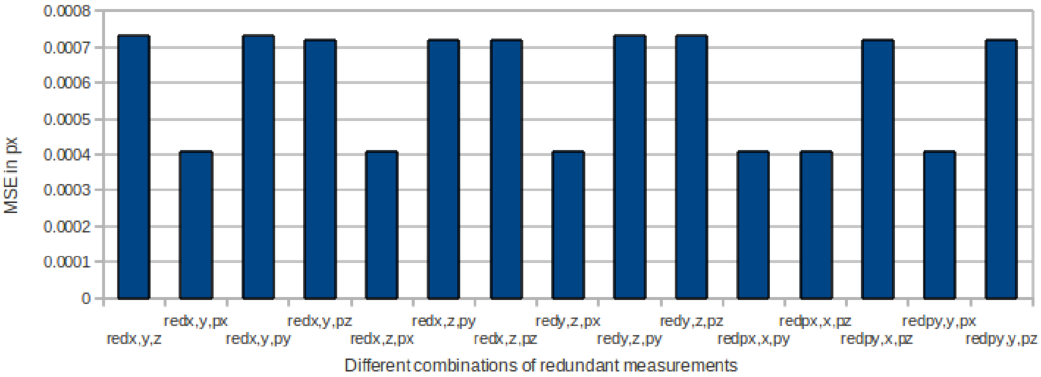


Figure 4.3: MSE in the presence of different combinations of 3 redundant measurement

introduced whereas no other redundant measurement decreased the MSE. Figure 4.2 shows the MSE in the measurement p_x for every possible combination of 2 redundant measurements. Again the MSE is low when a combination of redundant measurements with duplicate measurements of p_x is introduced. It can also be observed that MSE is further reduced when 2 duplicate measurements of p_x introduced. No improvements can be seen when a combination of redundant measurements, that doesn't involve a duplicate p_x, is introduced.

Figures 4.3 and 4.4 show the effects of each combination of 3 redundant measurements on the MSE of measurement p_x. A lower MSE is obtained by introducing a duplicate measurement of the weak state (p_x) or any of its combinations with other redundant measurements. The MSE is further decreased when more duplicate measurements of p_x are introduced. Again, the MSE is not decreased when a combination that doesn't involve duplicate measurement of p_x is introduced.

In figures 4.1-4.4 the bars that represent MSE switch between two levels: high or low. The reason is that when the system is weakly observable or has not converged, measurement residuals are high. Whereas in the presence of the duplicate measure-

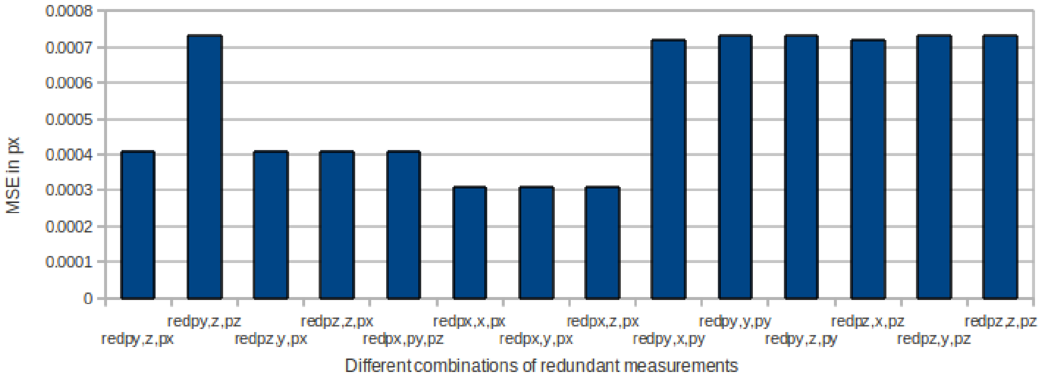


Figure 4.4: MSE in the presence of different combinations of 3 redundant measurement

ments, the measurement error is comparatively low before convergence. It should be noted at this stage that once the system has converged or becomes completely observable, measurement redundancy doesn't affect the system in terms of reducing measurement residuals. So the difference in MSEs comes from the error reduction before convergence.

4.2.4 Case 2: Relationship between measurement redundancy of TOF and relative observability

TOF is being introduced to the system to analyze its effects on relative observability. Figure 4.5 shows the MSE in z and p_z measurements when system is weakly observable due to the state p_z. In this figure, $z0red$ and $pz0red$ represents MSE in z and pz when there is no redundant measurement of TOF present in the system. Similarly, $z1red$ and $Pz1red$ represents MSE in z and p_z measurements when there is one redundant measurement of TOF present in the system, and $z2red$ and $Pz2red$ represents the MSE when two redundant measurements introduced. It is clear that a redundant

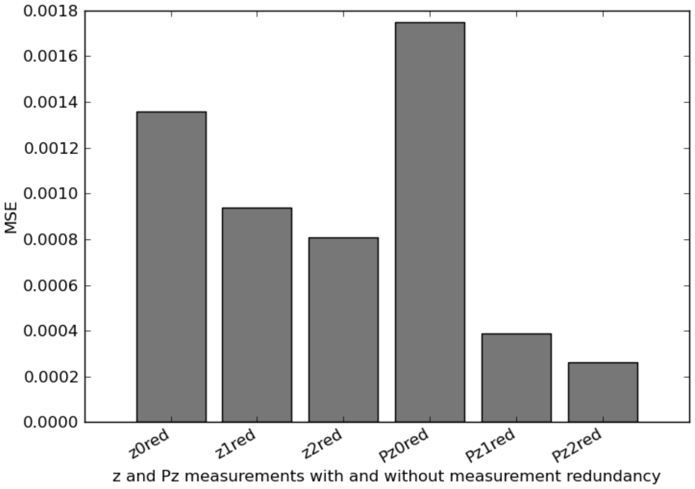


Figure 4.5: TOF as a redundant measurement and it's effects on z and Pz measurements

measurement of TOF improves the quality of the measurements by decreasing the MSE. MSE is further reduced when another redundant measurement is introduced to the system.

4.3 Conclusion

UKF does not need any projection techniques to cope with measurement redundancy, it's incorporated naturally. Accuracy of UKF and degree of redundancy has direct relationship; an increase in degree of redundancy also increases estimator's accuracy. But the accuracy of an estimate is only increased when a redundant measurement of itself is present in the system. Relative observability of a weakly observable state can be improved by reducing its measurement residuals. If a weak state is present in a set where all other states are strongly observable, then a large difference in measurement residuals can cause a condition number of the updated covariance to become too large and make the overall system ill-conditioned. Therefore the condition number can be reduced by introducing redundant measurements. A redundant measurement of a weakly observable state strengthens the weak state, and its estimate is obtained with good accuracy. The quality of the estimate corresponding to a weakly observable state can further be increased by carefully selecting redundant measurements and increasing the degree of redundancy. For instance, TOF measurement improved the results of p_z because TOF measurement depends on the momentum in z direction.

Chapter 5

Smoothing

The UKF was implemented in a particle tracking problem, explained in chapter 3. Particle tracking, in HEP experiments, is improved by implementing an algorithm called smoothing [5] (backward filtering) with the estimator. The overall idea of smoothing is to improve the accuracy of the estimates by using the data acquired during the estimation process (forward filtering). Since the smoothing algorithm is given converged data (data provided by the estimator) at the start, it does not only provide more accurate results by running backwards in time but also gives an insight about any events that were missed otherwise during the estimation process. Smoothing is currently being achieved by implementing the RTS smoother [5] with the EKF in these experiments. The RTS smoother is provided with all the necessary data computed by the estimator (EKF), hence the RTS smoother works directly with the EKF.

On the other hand, the RTS smoother cannot be directly implemented in the UKF formulation since Jacobians are not used. An extension of RTS smoother that can directly be implemented in the UKF formulation is called the URTS smoother [12]. Although URTS smoother can be implemented directly with UKF, the calculation of sigma points and cross covariances for smoother gains is a computational burden. In this chapter, a technique is presented that enables the RTS smoother to be implemented in the UKF using a Jacobian equivalent (JE) and the resulting algorithm is termed as JE-RTS smoother. The JE-RTS smoother is compared with the URTS smoother in terms of accuracy and computational complexity.

The sructure of this chapter is as follows: In Section 6.1 the EKF-based smoothing algorithm is described. In Section 6.2 smoothing is discussed in the context of UKF. The URTS and JE-RTS methods of smoothing are compared in section 6.3, and their implementation on particle tracking problem and results are presented in Section 6.4.

5.1 EKF and smoothing

The full information at the end of track as provided by the filter can be propagated back to all previous estimates by another iterative procedure, the Kalman smoother. This is also known as RTS smoother. A step of the smoother from layer $k+1$ to layer k for the state vector is given as,

$$\mathbf{x}_k^s = \hat{\mathbf{x}}_k + \mathbf{A}_k(\mathbf{x}_{k+1}^s - \hat{\mathbf{x}}_{k+1}^-), \qquad \mathbf{x}_k^s \in \Re^n \tag{5.1}$$

where $\mathbf{x}_{k+1}^s$ is the smoothed vector at step $k+1$ and $\mathbf{A}_k$ is smoother gain matrix given by:

$$\mathbf{A}_k = \mathbf{P}_{x_k}\mathbf{F}_{k+1|k}^T(\mathbf{P}_{x_{k+1}}^-)^{-1}, \qquad \mathbf{A}_k \in \Re^{n\times n} \tag{5.2}$$

where $\mathbf{F}_{k+1|k}$ and $\mathbf{P}_{x_{k+1}}^-$ are the Jacobian and predicted covariance at step $k+1$ respectively. The smoothed covariance matrix is given by:

$$\mathbf{P}_k^s = \mathbf{P}_{x_k} - \mathbf{A}_k(\mathbf{P}_{x_{k+1}}^- - \mathbf{P}_{k+1}^s)\mathbf{A}_k^T, \qquad \mathbf{P}_k^s \in \Re^{n\times n} \tag{5.3}$$

where $\mathbf{P}_{k+1}^s$ is the smoothed covariance at step $k+1$. The RTS smoother works directly with the EKF since the information (vectors and covariances) required in 5.1

-5.3, except $\mathbf{x}^s_{k+1}$ and $\mathbf{P}^s_{k+1}$, are calculated and stored during estimation process. This information is then available to the RTS smoother. Hence, No new calculations are needed.

5.2 UKF and smoothing

As described above, the RTS smoother uses Jacobian matrices in its smoother gain calculations. These Jacobian terms are stored during the estimation process and subsequently used in smoothing. Whereas in the UKF formulation, the Jacobian terms are not available and hence the RTS smoother can not directly be implemented. Hence, in order for smoothing to be implemented with UKF, a Jacobian equivalent of EKF for UKF is required. Section 6.2.1 explains an existing smoothing algorithm, the URTS smoother, that avoids Jacobian terms and can directly be implemented with UKF. Section 6.2.2 describes a new extension of the RTS smoother in the context of the UKF termed the JE-RTS smoother.

5.2.1 URTS smoother

In the URTS smoother, sigma points instead of Jacobians are computed for smoother gain calculations. These sigma points are calculated as,

$$\begin{aligned} \mathbf{X}_{i,k} &= \hat{\mathbf{x}}_k, & i &= 0, & (5.4)\\ &= \hat{\mathbf{x}}_k + \gamma\mathbf{S}_{i,k}, & i &= 1,\dots,n, & (5.5)\\ &= \hat{\mathbf{x}}_k - \gamma\mathbf{S}_{i,k}, & i &= n+1,\dots,2n, & (5.6) \end{aligned}$$

$\hat{\mathbf{x}}_k$ is the updated state vector and $\mathbf{S}_{i,k}$ is the ith column of the matrix $\mathbf{S}_k$, where $\mathbf{S}_k = \sqrt{\mathbf{P}_{x_k}}$, the square root is solved using Cholesky decomposition. $\mathbf{P}_{x_k}$ is the updated covariance. These sigma points are propagated through the dynamic model,

$$\mathbf{X}_{i,k+1} = f(\mathbf{X}_{i,k}, \mathbf{u}_k), \quad \mathbf{X}_{i,k+1} \in \Re^n \tag{5.7}$$

Predicted mean $\hat{\mathbf{x}}_{k+1}^{-}$, predicted covariance $\mathbf{P}_{x_{k+1}}^{-}$ and cross covariance $\mathbf{C}_{x_{k+1}}$ is calculated as,

$$\hat{\mathbf{x}}_{k+1}^{-} = \Sigma_{i=0}^{2n}(w_m^i \mathbf{X}_{i,k+1}) \tag{5.8}$$

$$\mathbf{P}_{x_{k+1}}^{-} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^{-})(\mathbf{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^{-})^T) + \mathbf{P}_v \tag{5.9}$$

$$\mathbf{C}_{x_{k+1}} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k} - \hat{\mathbf{x}}_k)(\mathbf{X}_{i,k+1} - \hat{\mathbf{x}}_{k+1}^{-})^T) \tag{5.10}$$

where $\mathbf{C}_{x_{k+1}} \in \Re^{n \times n}$. Smoother gain is calculated as,

$$\mathbf{D}_k = \mathbf{C}_{x_{k+1}}(\mathbf{P}_{x_{k+1}}^{-})^{-1}, \quad \mathbf{D}_k \in \Re^{n \times n} \tag{5.11}$$

And finally the standard RTS smoother equations (5.1) and (5.3) are used to calculate smoothed state and covariance respectively.

5.2.2 JE-RTS smoother

The Jacobian equivalent is derived by comparing (3.4) and (3.17), and is given by:

$$\mathbf{P}_{x_k}^{-} = \mathbf{F}_k \mathbf{P}_{x_{k-1}} \mathbf{F}_k^T = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^{-})(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^{-})^T) \tag{5.12}$$

assuming process noise being equal for both the estimators. Simplifying (5.12) gives;

$$\mathbf{F}_k \mathbf{F}_k^T = (\mathbf{P}_{x_k}^{-} - \mathbf{P}_v)(\mathbf{P}_{x_{k-1}})^{-1} \tag{5.13}$$

Further simplification gives;

$$\mathbf{F}_k^T = \sqrt{(\mathbf{P}_{x_k}^{-} - \mathbf{P}_v)(\mathbf{P}_{x_{k-1}})^{-1}} \tag{5.14}$$

The Jacobian equivalent is substituted in (5.2) in order to calculate smoother gains which are subsequently required in (5.1) and (5.3) for smoother state and covariance computations. From now on the EKF's equivalent Jacobian for the UKF will be written as $\mathbf{F}^{UKF}$. Some tests carried out to compare the Jacobian and its eqivalent. Both the estimators implemented on Van der Pol oscillator and the updated covariances were found equal in terms of eigenvalues and structure.

5.3 A comparison of URTS and JE-RTS smoothers

The URTS smoother is different from the JE-RTS smoother in smoother gain calculations. In the URTS smoother, the smoother gain is calculated as,

$$\mathbf{D}_k = \mathbf{C}_{x_{k+1}}(\mathbf{P}^-_{x_{k+1}})^{-1}$$

where $\mathbf{C}_{x_{k+1}}$ represented by (5.10) can also be written as;

$$\mathbf{C}_{x_{k+1}} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T) \tag{5.15}$$

It should be noted here that (5.15) is an equivalent of (5.10) and that the difference is only in subscripts. $\mathbf{C}_{x_{k+1}}$ is being written differently in this section so that the smoother gains in URTS smoother can be compared with the ones in JE-RTS smoother. Substituting (5.15) into (5.11);

$$\mathbf{D}_k = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T)(\mathbf{P}^-_{x_{k+1}})^{-1} \tag{5.16}$$

whereas in JE-RTS smoother, smoother gain is calculated as;

$$\mathbf{A}_k = \mathbf{P}_{x_k}(\mathbf{F}^{UKF}_{k+1|k})^T(\mathbf{P}^-_{x_{k+1}})^{-1} \tag{5.17}$$

Substituting $\mathbf{F}^{UKF}_{k+1|k}$ gives;

$$\mathbf{A}_k = \mathbf{P}_{x_k}\sqrt{\frac{(\mathbf{P}^-_{x_{k+1}} - \mathbf{P}_v)}{\mathbf{P}_{x_k}}(\mathbf{P}^-_{x_{k+1}})^{-1}} \tag{5.18}$$

Now substitution of $(\mathbf{P}^-_{x_{k+1}} - \mathbf{P}_v)$ gives;

$$\mathbf{A}_k = \frac{\mathbf{P}_{x_k}}{\sqrt{\mathbf{P}_{x_k}}}(\mathbf{P}^-_{x_{k+1}})^{-1}\sqrt{\Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T)} \tag{5.19}$$

$\mathbf{P}_{x_k}$ can be written as;

$$\mathbf{P}_{x_k} = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{X}_{i,k-1} - \hat{\mathbf{x}}_{k-1})^T) \tag{5.20}$$

Substituting (5.20) into (5.19);

$$\mathbf{A}_k = \Sigma_{i=0}^{2n}(w_c^i(\mathbf{X}_{i,k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T)(\mathbf{P}_{x_{k+1}}^-)^{-1} \tag{5.21}$$

Equations (5.16) and (5.21) provide an insight about the smoother gains of URTS and JE-RTS smoothers respectively. It is clear from these equations that smoother gain D_K (for URTS smoother) and A_k (for JE-RTS smoother) are equivalent. Therefore these approaches are similar in terms of the information that they add to the system for smoothing.

The difference arises when these approaches are analysed in terms of computational cost. The URTS smoother involves the calculation and propagation of sigma points which are calculated using the updated states and covariances provided by the filter. These sigma points are required for cross covariances which are used in smoother gain calculations. So the information required in (5.1) and (5.3) is calculated during the smoothing process. Whereas in the JE-RTS smoother the information required for smoothing, except $\mathbf{x}_{k+1}^s$ and $\mathbf{P}_{k+1}^s$, in (5.1) and (5.3) is stored while filtering process.

5.4 Implementation of URTS and JE-RTS smoothers

The URTS and JE-RTS methods of smoothing are implemented with the UKF in the particle tracking problem briefly described in chapter 5. In this section, firstly, the JE-RTS and URTS smoothers are being compared in terms of accuracy. The measurement noise is 0 mean with a standard deviation of 0.01m for position measurements and 0.01Mev/c for momentum measurements. Secondly these smoothing algorithms are analysed in terms of computational cost.

MSE	UKF	$URTS$	$JE-RTS$
x	$1.20mm$	$1.19mm$	$1.18mm$
y	$4.36mm$	$4.34mm$	$4.29mm$
z	$3.68mm$	$3.50mm$	$3.64mm$
p_x	$0.000349MeV/c$	$0.00034MeV/c$	$0.00029MeV/c$
p_y	$0.00132MeV/c$	$0.00133MeV/c$	$0.00137MeV/c$
p_z	$0.00038MeV/c$	$0.000373MeV/c$	$0.000371MeV/c$

Table 5.1: Mean squared error in each measurement for UKF, URTS smoother and JE-RTS smoother

5.4.1 Accuracy analysis

Table 5.1 shows the Mean Squared Error (MSE) in each measurement. The second column of 5.1 shows the MSE in the estimates achieved using the UKF with no smoothing applied. The third and fourth column of table 5.1 represents MSEs in each smoothed estimate achieved using the URTS and JE-RTS smoothers respectively. It is clear from these results that smoothing provides better results when compared to the original UKF estimates. This happens because the smoothers are provided with more accurate estimates at the start. The MSEs of the smoothed estimates achieved using URTS and JE-RTS mothods of smoothing are similar. None of the presented approaches have advantages over the other when their accuracy is analysed, as can be seen from the analytical comparison presented in section 6.3.

5.4.2 Computational cost analysis

Table 5.2 shows the time consumed by the estimator (UKF), URTS smoother and JE-RTS smoother. The second row of this table represents the time consumed by the estimator when required to be implemented with these smoothing approaches. Time

Time consumed	$URTSsmoother$	$JE-RTSsmoother$
Estimator	$2.866sec$	$2.946sec$
Smoother	$0.456sec$	$0.219sec$
Estimator+Smoother	$3.322sec$	$3.165sec$

Table 5.2: Average time consumed by URTS smoother and JE-RTS smoother while estimation and smoothing processes

consumed by the estimator is important since the URTS and JE-RTS smoothers are different in terms of their dependence on the data stored during the estimation process. It should be noted here that time consumed by the estimator includes the time taken for information storage required for smoothing. It is clear from table 5.2 that estimator consumes less time when it is required to be implemented with URTS smoother as compared to JE-RTS mothod. The reason is that URTS smoother does not rely heavily on stored information.

The third row of table 5.2 represents the time consumed by the smoothers. Clearly JE-RTS smoother outperforms the URTS in terms of the time consumed. JE-RTS smoother is fast because, unlike URTS smoother, it does not require the recalculation and propagation of sigma points for its smoother gain calculation.

The fourth row of table 5.2 shows the addition of the time consumed by the estimator and smoothers. The overall time consumed is less when JE-RTS smoother is implemented with UKF. Reader is refered to section 6.3 that briefly discusses these approaches in terms of accuracy and computational complexity.

5.5 Conclusion

URTS and JE-RTS smoothers perform similarly in terms of accuracy since the information added by the smoother gains is approximately the same. The smoother gains in the JE-RTS smoother are calculated using the states and covariances stored during the filtering process, whereas sigma points are re-calculated in URTS smoother. Therefore, JE-RTS smoother requires fewer computations than URTS. On the other hand, the UKF (as an estimator) when required to be implemented with URTS smoother requires less storage of states and covariances. JE-RTS smoother, when implemented with UKF, performs better in terms of overall time consumed.

Chapter 6

Discussions and conclusions

6.1 Concluding summary

In HEP experiments, EKF is currently being used for track fitting. The main challenges in track fitting are the large dimensionality, high non-linearity due to magnetic field and multiple scattering and requirement of high accuracy. Therefore, any track fitting algorithm should be computationally efficient, accurate and robust against outliers and non-linearities. Although the EKF has been implemented for track fitting for almost forty years, but research shows that it is less accurate and diverges when exposed to severe non-linearities. Implementation of an efficient algorithm could improve the track fitting and therefore UKF was selected due to its better performance and comparable computational cost to the EKF.

The UKF is implemented to MICE and is analysed in terms of accuracy against outliers. It is shown that how the UKF works with measurement redundancy and how this can down-weight the effects of an outlier. Each measurement is introduced as a redundant measurement and it is concluded that not all the redundant measurements improve the results of the affected measurement but only the duplicate one. Degree of redundancy further improve the results, but again, if the redundant measurement is the duplicate one.

The relationship between measurement redundancy and relative observability is analysed. And it is concluded that the relative observability of a weakly observable state can only be improved if its duplicate measurement is introduced as a redundant measurement. Relative observability is also investigated by increasing the degree of redundancy, different combination of redundant measurements introduced. The relative observability of the weakly observable state improved whenever the combination of the redundant measurements had involved its duplicate measurement. To have more insight about this relationship, an independent redundant measurement, TOF, introduced as a redundant measurement and its effects on the quality of the measurement are investigated. TOF, as a redundant measurement, only improves the quality of z and P_z measurements. It was concluded that since TOF measurement involves P_z measurement in its calculations, the information added by it is in z direction.

Smoothing is achieved using RTS smoother within the EKF framework. Since UKF has never been implemented to HEP problem before, no efficient smoother was available. Although a smoother existed that could have been implemented with the UKF called URTS, it is less efficient in terms of computational cost. A technique incorporated in the RTS smoother to make it suitable for the UKF and large dimensional problems. We gave it a name “JE-RTS smoother”. JE-RTS smoother outperforms the URTS smoother in terms of computational complexity by preserving the similar accuracy.

6.2 Discussions

When an extra measurement is added as a redundant measurement, it adds new information. But the question arises how this information affects different existing measurements, which measurements are affected and which ones remain unchanged or are least affected.

The UKF works naturally with measurement redundancy. When a redundant measurement is added to the system, its associated column appears in the measurement error covariance of the UKF where each column reflects the uncertainties in the associated measurement. When the redundant measurement is a duplicate of one of the existing measurements, there will be two identical columns in the measurement covariance. This implies that the column vector associated to the redundant measurement is not linearly independent to its identical column but to all the column vectors associated to the other measurements. Therefore, the redundant measurement only affects its duplicate one (information is added only in one direction) and all other measurments are preserved.

In the case of measurement redundancy, the Kalman gain matrix also becomes a nonsquare matrix containing two identical columns. Now in the innovation step of the UKF, the information for redundant measurement is added twice and the results are improved. The extra column in the Kalman gain does not add information into any other measurement.

Having said that, a redundant measurement can have affects on some of the measurement which are not duplicate ones due to the system dynamics or measurement dependencies. To analyse measurement dependencies, an independent redundant measurement, TOF, was introduced. TOF only affects P_z, momentum is z direction. The reason is that TOF measurement depends on the momentum in z direction.

It can be inferred from the above findings that measurement redundancy does not always improve the results in terms of accuracy and observability and that before introducing it a careful consideration should be given to the available measurements, system dynamics and measurement dependencies.

Although the EKF handles redundancy, it does it at the cost of projection. So adding redundancy in the EKF framework also invites the extra computational cost whereas the UKF does not use any projection technique and works naturally with the measurement redundancy. This is an additional advantage of the UKF especially in large dimensional problems.

6.3 Conclusions

The UKF is a more efficient algorithm as compared to the EKF in terms of accuracy and robustness against non-linearities such as faced in particle physics problems. The EKF performs poorly when compared with the UKF for weak observability that arises due to ionisation energy loss.

A system must be observable in order for an algorithm to converge. However an algorithm might fail to converge even if the system is practically observable. This happens if the system is not well behaved (badly conditioned) or the algorithm is not efficient enough to deal with it. The convergence of an algorithm can be improved by improving the relative observability of the system. One way of doing this is through introducing measurement redundancy. It is worth mentioning here that measurement redundancy is incorporated naturally in the UKF. An independent redundant measurement (not from the state vector) improves the relative observability of the system and subsequently helps an algorithm to converge to an improved solution.

The RTS smoother cannot directly be implemented with the UKF. Although the URTS smoother is suitable for the UKF, it is not computationally efficient especially in large dimensional systems. The JE-RTS smoother is computationally more efficient as compared to the URTS smoother and provides similar accuracy. Therefore JE-RTS smoother is a better choice in the large dimensional systems such as HEP experiments.

6.4 Future work

The extension of this research is to take UKF forward and analyse it in the LHC experiments which, as compared to MICE, are very large dimensional. It would be interesting to see how the UKF will work with such systems where million of particle traverse the detectors simultaneously and several kalman filters are implemented.

Another possible extension is to implement UKF as an alignment algorithm, a hugely large dimensional problem. To deal with the issue of computational complexity, UKF can be parallelized and this also is possible area of future research.

Bibliography

[1] A. Strandile, "Track reconstruction in the LHC experiments," Proceedings of the first LHC detector alignment workshop, CERN, Geneva, Switzerland, September 2006.

[2] Billoir, P., and S. Qian, "Simultaneous Pattern recognition and track fitting by the Kalman filtering method," Nuclear Instruments and Methods in Physics Research A 294, 219-228, 1990.

[3] Mankel, R, "A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system," Nuclear Instruments and Methods in Physics Research A 395, 1997.

[4] Are Strandile and Rudolf Fruhwirth, "Track and vertex reconstruction: From classical to adaptive methods," Reviews of modern physics, Volume 82, April-June 2010.

[5] Robert Grover Brown and Patrick Y.C. Hwang, "Introduction to random signals and applied Kalman filtering, fourth edition. Wiley, 2001, ch. 7.

[6] V. Blobel, "Alignment algorithms, Proceedings of the first LHC detector alignment workshop, CERN, Geneva, Switzerland, September 2006.

[7] R. Fruhwirth and E. Widl, "Track based alignment using a Kalman filter technique, Proceedings of the first LHC detector alignment workshop, CERN, Geneva, Switzerland, September 2006.

[8] C. Amsler et al., “Review of particle Physics” physics letters B667, 2008.

[9] Yuanxin Wu, Dewen Hu,“ Unscented Kalman Filtering for Additive Noise Case: Augmented versus Nonaugmented”, 2005.

[10] Eric A. Wan and Rudolph van der Merwe, “Spherical Simplex Sigma-Point Kalman Filters: A comparison in the inertial navigation of a terrestrial vehicle”, 2008.

[11] H. Sakamoto, “ A Study of Muon Ionization Cooling at MICE,” Ph.D dissertation, Department of Physics, Graduate School of Science, Osaka University, 2010.

[12] Simo Sarkka, “Unscented RauchTungStriebel Smoother,” IEEE Transactions on Automatic Control, VOL. 53, NO. 3, APRIL 2008.

[13] Hansroul, H. Jeremie, and D. Savard, , “Fast circle fit with the conformal mapping method,” Nuclear Instruments and Methods in Physics Research A 270, 498, 1988.

[14] Hough, P. V. C, “Machine analysis of bubble chamber pictures,” Proceedings of the International Conference on High Energy Accelerators and Instrumentation, CERN, Geneva, 1959.

[15] T. Alexopoulos et al., “Identification of circles from data points using the Legendre transform,” Nuclear Instruments and Methods in Physics Research A 745, 2014.

[16] P. J. Rousseeuw and A. M. Leroy, “Robust regression and outlier detection, Wiley, New York, 1987.

[17] Simon J. Julier and Jeffrey K. Uhlmann, “A New Extension of the Kalman Filter to non-linear Systems,” Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Deffence Sensing, Simulation and Control, 1997.

[18] Eric A. Wan and Rudolph van der Merwe, “The Unscented Kalman Filter for Nonlinear Estimation”, 2000.

[19] Fredrik Orderud, "Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements", 2005.

[20] Joseph, J. LaViola, "A Comparison of Unscented and Extended Kalrnan Filtering for Estimating Quaternion Motion," IEEE Proceedings of the American Control Conference, Denver, Colorado June 4-6, 2003.

[21] Rudolph van der Merwe and Eric Wan, "Sigma point Kalman filters for non-linear estimation and sensor fusion", Guidance, Navigation, and Control Conference, Rhode island, 2005.

[22] Rudolph van der Merwe and Eric Wan, "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models," Ph.D dissertation, OGI School of Science and Engineering, Oregon Health and Science University, Portland, Oregon, 2004.

[23] A Dobbs et al., " The Reconstruction Software for the Muon Ionization Cooling Experiment Trackers," Proceedings of 20th International Conference on Computing in High Energy and Nuclear Physics, 1418 October 2013, Amsterdam, Netherlands, 2014.

[24] Are Strandile and Rudolf Fruhwirth, "Application of the Kalman alignment algorithm to the CMS tracker," Journal of physics: Conference series 219, 2010.

[25] Louis Nicolas, Adlene Hicheur, Matthew Needham, Jan Amoraal, Wouter Hulsbergen and Gerhard Raven, "Alignment of LHCb tracking stations with tracks fitted with a Kalman filter," IEEE Nuclear Science Symposium, 2008.

[26] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2, FEBRUARY 2002.

[27] Katalin Gyrgy, Andrs Kelemen and Lszl Dvid, "Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation," The 7th International Conference Interdisciplinarity in Engineering, INTER-ENG 2013. 10-11 October 2013, Petru Maior University of Tirgu Mures, Romania

[28] Eleni N. Chatzi, and Andrew W. Smyth, "The Unscented Kalman Filter and Particle Filter Methods for Nonlinear Structural System Identi

cation with Non-Collocated Heterogeneous Sensing," JOURNAL OF STRUCTURAL CONTROL, 2002.

[29] Kung-Chung Lee, Anand Oka, Emmanuel Pollakis and Lutz Lampe, "A Comparison between Unscented Kalman Filtering and Particle Filtering for RSSI-Based Tracking," Positioning Navigation and Communication (WPNC) 7th Workshop, 2010.

[30] Rudolph van der Merwe , Arnaud Doucet, Nando de Freitas and Eric Wan, "The Unscented Particle Filter ," Adv. Neural Inform. Process. Syst., Dec. 2000.

[31] Wentao Yu, Jun Peng, Xiaoyong Zhang, Shuo Li, and Weirong Liu, "An Adaptive Unscented Particle Filter Algorithm through Relative Entropy for Mobile Robot Self-Localization," Mathematical Problems in Engineering Volume 2013, Article ID 567373, 2013.

[32] ColinMcManus, "The Unscented Kalman Filter for State Estimation", 2010.

[33] Simon J. Julier, "The Spherical Simplex Unscented Transformation", 2003.

[34] Rambabu Kandepu, Bjarne Foss and Lars Imsland, "Applying the unscented Kalman filter for nonlinear state estimation," 2008.

[35] Simon J. Julier, "The Scaled Unscented Transformation", 2002.

[36] Fuming Sun, Guanglin Li, Jingli Wang," Unscented Kalman Filter using Augmented State in the Presence of Additive Noise", 2009.

[37] C. Heidt, "The tracker system for the muon ionization cooling experiment," Nuclear Instruments and Methods in Physics Research A, Volume 718 (2013), 560-562.

[38] D. Rajaram and P. Snopok, " Muon Ionization Cooling Experiment step VI," proceedings of IPAC 2013, Shanghai, China, 2013.

[39] P. Kyberd, " Emittance measurement devices in the Muon Ionization Cooling Experiment," Proceedings of 9th European Workshop, DIPAC 2009, Basel, Switzerland, May 25-27, 2009.

[40] D. Adey, " Measuring Emittance with The MICE Scintillating Fibre Trackers ," Proceedings of IPAC2011, San Sebastin, Spain, 2011.

[41] M. Ellis et al., "The Design, construction and performance of the MICE scintillating fibre trackers," Nuclear Instruments and Methods in Physics Research A, Volume 659, Issue 1, p. 136-153, June 2011.

[42] Stefano Mariani and Aldo Ghisi, "Unscented Kalman filtering for non-linear structural dynamics," non-linear Dyn. 49:131150, 2007.

[43] Oliver M. Collins and N. Vasudev, "The Effect of Redundancy on Measurement," IEEE Transactions on Information Theory, pp.3390-3096, 2001.

[44] M C S Williams, "Particle identification using time of flight," Journal of Physics G: Nuclear and Particle Physics, 19 October 2012.

[45] A. G. O. Mutambara, "Decentralized Estimation and Control for Multisensor Systems," CRC Press, Boca Raton, 1998.

[46] A. Salzburger, "A parameterization for fast simulation of muon tracks in the ATLAS inner detector and muon systems," CERN-Thesis-2004-051, 2003.

[47] Y. Ohnishi, "Track parameterization," KEK, National laboratory of high energy physics, Tsukuba, Japan, 1997.

[48] M. Limper, "Track and vertex reconstruction in the ATLAS inner detector," PhD Thesis, October 2009.

[49] S. Cucciarelli, "Track and vertex reconstruction with the CMS detector," 10th international conference in B-Physics at hadron machines, Assisi, Italy, 20-24 June, 2005.

[50] G. B. Cerati, "CMS tracking performance and sensitivity for the MSSM decay," PhD thesis, School of mathematical, physical and natural sciences, University of Milan, 2008.

[51] M. Campanelli, A. Clark, X. Wu, "Hadron collider physics," Proceedings of the 1st hadron collider physics symposium, Les Diablerets, Switzerland, July 4-9, 2005.

[52] Fruhwirth et al, "Data analysis techniques for high energy physics," 2nd edition, Cambridge University press, 2000.

[53] A. Strandlie and W. Wittek, "Propagation of Covariance Matrices of Track Parameters in Homogeneous Magnetic Fields in CMS," Nuclear Instruments and Methods in Physics Research A 566, 687, 2006.

[54] L. Bugge and J. Myrheim, "Tracking and track fitting," Nuclear Instruments and Methods, vol 179, 365-381, 1981.

[55] Amsler et al, "particle data group," physics letters B, 667,1, 2008.

[56] H. Bethe and W. Heitler, "On the Stopping of Fast Particles and on the Creation of Positive Electrons ," Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, Vol. 146, No. 856, pp. 83-112, August 1934.

[57] H. Eichinger and M. Regler, "Review of track fitting methods in counter experiments," CERN technical report, 81-06, 1981.

[58] P. Billoir and R. Fruhwirth and M. Regler, "Track element merging strategy and vertex fitting in complex modular detectors," Nuclear Instruments and methods in Physics research A 241, 115-131, 1985.

[59] P. Billoir, "Track Fitting With Multiple Scattering: A New Method," Nuclear Instruments and methods in Physics research Volume 225, Issue 2, Pages 352366, 15 August 1984.

[60] R. Fruhwirth, "Application of Kalman filtering to track and vertex fitting," Nuclear Instruments and methods in Physics research A, Volume 262, Issues 23, Pages 444-450, 15 December 1987.

[61] I.A. Golutvin et. al, "Robust estimates of track parameters and spatial resolution for CMS muon chambers," Computer Physics Communications, Volume 126, Issues 12, April 2000, Pages 7276, 2000.

[62] F. R. Hampel et. al, "Robust statistics, the approach based on the influence function," John Wiley and sons, New York, 1986.

[63] S. R. Das, "On a New Approach for Finding All the Modified Cut-Sets in an Incompatibility Graph," IEEE Transactions on Computers, Volume 22, Issue 2, February 1973, Pages 187-193, 1973.

[64] R. Fruhwirth, "Selection of optimal subsets of tracks with a feed-back neural network," Computer Physics Communications, Volume 78, Issues 12, December 1993, Pages 2328, 1993.

[65] D. Wicke, "A new algorithm for solving tracking ambiguities," technical report DELPHI 96-163, PROG 236, track 92, CERN, 1998.

[66] A. Glazov, I. Kisel, E. Konotopskaya and G. Ososkov, "Filtering tracks in discrete detectors using a cellular automaton," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 329, Issues 12, 15 May 1993, Pages 262268, 1993.

[67] I. Kisel, "Event reconstruction in the CBM experiment," Nuclear Instruments and Methods in Physics Research Section A, Volume 566, pages 85-88, 2006.

[68] I. Kisel et. al, "Cellular automaton and elastic net for event reconstruction in the NEMO-2 experiment," Nuclear Instruments and Methods in Physics Research Section A, Volume 387, page 433, 1997.

[69] R. E. Kalman, "A new approach to linear filtering and prediction problems," ASME Journal of basic Engineering, 82 (Series D): 35-45, 1960.

[70] S. M. Kay, "Estimation theory," Volume 1 of fundamentals of statistical signal processing, Prentice hall PTR, 1993.

[71] A. Gelb, "Applied optimal estimation," MIT press, 1974.

[72] B. Anderson and J. Moore, "Optimal filtering," prentice-Hall, 1979.

[73] A. Jazwinsky, "Stochastic process and filtering theory," Academic press, New York, 1970.

[74] S. Haykin, "Kalman filtering and neural networks," John Wiley and sons, 2001.

[75] A. T. Nelson, "Non-linear estimation and modeling of noisy time series by dual Kalman filtering methods," PhD thesis, Oregon graduate institute, 2000.

[76] NAVSYS Corporation: GPS Inertial products. http://www.navsys.com.

[77] Inertial science INC, ISIS-GPS: Integrated INS/GPS GNS systems. http://www.inertialscience.com.

[78] CROSSBOW technology, INC. Crossbow: Smarter sensors in silicon. http://www.xbow.com.

[79] Thales navigation, INC. Thales navigation and Magellan GPS systems. http://www.thalesnavigation.com.

[80] D. Hoag, "Appolo guidance and navigation: Consideration of APPOLO IMU Gimbal lock," technical report E1344, MIT instrumentation lab, April 1963.

[81] G. V. Puskorius, L. A. Feldkamp and and L. I. Davis, "Dynamic neural network methods applied to on vehicle idle speed control," proceedings of the IEEE 84, pp. 1407-1419 October 1996.

[82] R. Tawel, N. Aranki, G. V. Puskorius, K. A. Marko, L. A. Feldkamp, J. V. James, G. Jesion and T. M. Feldkamp, "Custom VLSI ASIC for automotive applications with recurrent networks," proceedings of the IEEE international joint conference on neural networks, pp. 598-602, Anchorage, AK, May 1998.

[83] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," technical report CUED/F-INFENG/TR 310, Department of Engineering, University of Cambridge, 1998.

[84] S. Julier, J. Uhlmann and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," proceddings of the american control conference, pp. 1628-1632, 1995.

[85] S. Julier and J. Uhlmann, "A general method for approximating non-linear transformation of probability distribution," Tech. Rep. RRG, Department of Engineering Science, University of Oxford, November 1996.

[86] S. Julier, "A skewed approach to filtering," SPIE conference on signal and data processing of small targets, vol 3373, SPIE, pp. 271-282, Orlando, Florida, April 1998.

[87] J. Stirling, "Methodus differentials, sive tractatus de summation et interpolation serierum infinitarium," 1730.

[88] K. ITO and K. Xiong, "Gaussian filters for non-linear filtering problems," IEEE transactions on automatic control 45, 5. pp 910-927, May 2000.

[89] M. Norgaard and N. Poulsen and O. Ravn, "New developments in state estimation for non-linear systems," Automatica 36, 11. pp 1627-1638, November 2000.

[90] R. V. D. Marwe and E. Wan, "Efficient derivative-free Kalman filters for online learning," proceedings of the 9th European symposium on artificial nural networks (ESANN), pp. 205-210, Belgium, April 2001.

[91] R. V. D. Marwe and E. Wan, "The square-root unscented Kalman filter for state and parameter estimation," proceedings of the IEEE international conference on Acoustics, speech and signal processing, vol. 6, pp. 3461-3464, Salt lake city, UT, May 2001.

[92] T. Lefebvre, H. Bruyninckx and D. Schutter, "A new method for the non-linear transformation of means and covariances in filters and estimators," IEEE transactions on automatic control 47, 8, pp. 1406-1409, Belgium, August 2002.

[93] S. J. Julier, "Comprehensive process models for high speed navigation," PhD thesis, University of Oxford, England, 1997.

[94] A. Cervera-Villanueva et al., "Kalman filter tracking and vertexing in a silicon detector for neutrino physics," Nuclear Instruments and Methods in Physics Research A 486 (2002) 639-662, November 2001.

[95] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If I had a million particles, in Proc. ICML 2006, Pittsburgh, PA, pp. 481488.

[96] E. A. Wan and R. van der Merwe, The unscented Kalman filter, in Kalman Filtering and Neural Networks, S. Haykin, Ed. Wiley ch. 7, 2001.