PERFORMANCE ANALYSIS OF MULTIPLE MODEL KALMAN FILTERS FOR TARGET TRACKING

by

Yağız Akalın

B.S., in Computer Engineering, Boğaziçi University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical & Electronics Engineering

Boğaziçi University

2021

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Prof. Emin Anarım, whose expertise was invaluable in formulating the research questions and methodology. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would also like to thank Prof. Mutlu Koca for his valuable guidance throughout my studies. You provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.

I would like to forward my appreciation to all my friends and colleagues who contributed to my thesis with their continuous encouragement.

In addition, I would also like to give special thanks to my parents and my sister as a whole for their continious support and understanding when undertaking my research and writing my thesis. Your prayer for me was what sustained me this far.

ABSTRACT

PERFORMANCE ANALYSIS OF MULTIPLE MODEL KALMAN FILTERS FOR TARGET TRACKING

Target tracking is a computer vision problem on which many studies have been done and research on this topic is still ongoing. The main tasks of target tracking systems can be listed as determining the position, velocity or acceleration of one or more moving targets. Target tracking relies on a recursive prediction using noisy measurements from the radar in order to calculate the next movement of the target. Usually the data measured by monitoring devices is not precise, as the measurements have some kind of measurement noise depending on the sensor. Therefore, the measurement noise of the sensors complicates the target tracking and it is necessary to filter the noise in order to estimate the real path of the moving targets and improve the estimation of the trajectories of the targets. Although the measurement coming from the radar is in polar coordinates, in modern target tracking applications, since the motion of the target is linear in the Cartesian system, the state estimation of the next movement of the target is done in the Cartesian coordinate system. In this thesis, various Kalman Filters were investigated to monitor polar measurements with Cartesian coordinates, remodel noise and then compare the performances of these filters. In addition, since the target does not depend on a single movement model in real life, systems that enable the interaction of more than one movement model are used. Furthermore, the motion models and sample tracking scenarios used to evaluate the performance of these filters were defined, and the performance of multiple filter systems was evaluated through simulations made on different scenarios.

ÖZET

ÇOKLU MODEL KALMAN SÜZGEÇLERİNİN HEDEF İZLEMEDEKİ BAŞARIM ANALİZİ

Hedef izleme, üzerinde birçok çalışmanın yapıldığı ve araştırmaların halen devam ettiği bir bilgisayarla görü problemdir. Hedef izleme sistemlerinin ana görevleri, bir veya daha fazla hareketli hedefin konumunu, hızını veya ivmesini belirlemek olarak listelenebilir. Hedef takibi, hedefin bir sonraki hareketinin özyinelemeli şekilde hesaplanması amacıyla radardan gelen gürültülü ölçümlerin kullanılarak ardışık bir kestirimde bulunmasına dayanır. Ölçümler, sensöre bağlı olarak bir tür ölçüm gürültüsüne sahip olduğundan ötürü genellikle izleme cihazları tarafından ölçülen veriler kesin olmamaktadır. Bu nedenle sensörlerin ölçüm gürültüsü, hedef takibini karmaşık hale getirmekte olup, hareketli hedeflerin gerçek yolunu tahmin etmek ve hedeflerin yörüngelerinin tahminini geliştirmek için gürültüyü süzgeçlemek gerekmektedir. Radardan gelen ölçüm bilgilerinin polar koordinatta olmasına karşın, modern hedef takibi uygulamalarında hedefe ait hareketin kartezyen sistemde doğrusal olması sebebiyle hedefin sonraki hareketine ait durum tahmini kartezyen koordinat sisteminde yapılır. Bu tezde, polar koordinattaki ölçümlerin kartezyen koordinatlar ile takip edilmesi ve gürültünün tekrar modellenmesi için çeşitli Kalman Süzgeçleri araştırılmış ve bu süzgeçlerin performansları karşılaştırılmıştır. Ayrıca gerçek hayatta hedef, tek bir hareket modeline bağlı kalmaması nedeniyle birden fazla hareket modelinin etkileşimini içeren sistemler kullanılmıştır. İkinci olarak ise bu filtrelerin başarımlarını değerlendirmede kullanılan hareket modelleri ve örnek takip senaryoları tanımlanmış olup, farklı senaryolar üzerinde yapılan simülasyonlar vasıtasıyla çoklu filtre sistemlerinin başarımı değerlendirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

R^2	Set of all pairs of real numbers
R^3	Set of all ordered triples of real numbers
X_t	Motion of a target at time t
W_t	White Gaussian noise
x_t	x-axis location of the target at time t
y_t	y-axis location of the target at time t
$\dot{x}_t$	Velocity component of a target at x-axis
$\dot{y}_t$	Velocity component of a target at x-axis
Q_t	Covariance matrix of White Gaussian noise
Z_t^P	Measurement of the position vector
r_t	Measurement of the range at time t
$h_t^P(X_t)$	Measurement function of the position
V_t^P	Measurement noise vector
R^P	Covariance matrix of the measurement noise
P(A)	Prior probability
P(B)	Marginal probability
P($B \mid A$)	Likelihood probability
$P_{k\|k-1}$(x)	Probability that the target is at the x-axis
$f_{k\|k-1}$(x,y)	Model that dictates how the target propagated over time
P_{k-1}(y)	Probability that the target was at y-axis at time k-1
$g_k(z \mid x)$	Likelihood probability
x_k	State vector at time k
w_k	Process noise
z_k	Measurement vector
H_k	Coefficient matrix of measurement
v_k	Measurement noise
P_k	Covariance matrix of the estimation
Q_k	Process noise covariance matrix

K_k	Kalman gain at time k
R_k	Measurement noise covariance matrix
I	Identity matrix
f_k	Nonlinear state transition function
h_k	Nonlinear measurement function
P_0	Covariance of the initial state
F_k	Jacobian of the state transition matrix
H_k	Jacobian of the measurement matrix
W_i^m	Weights for the mean
W_i^c	Weights for the covariance
S_k	Predicted measurement covariance
C_k	Cross-correlation matrix
Z^{CM}	Converted measurements
R^{CM}	Covariance of the converted measurements
r_m	Range measurement
x_m^c	Cartesian coordinate measurements at x-axis
y_m^c	Cartesian coordinate measurements at y-axis
Z_k^{uP}	Unbiased converted measurement vector
R_u^P	Covariance matrix of the unbiased conversion
Z_k^{dP}	Decorrelated unbiased converted measurement vector
R_{du}^P	Covariance matrix of the decorrelated unbiased conversion
$\bar{c}_j$	Normalization coefficient
p_{ij}	Model transition probability
m	Number of models in IMM
$\bar{z}_j$	Converted measurement
S_j	Covariance matrix of the converted measurement
$X_{k\|k}^j$	Estimation output of IMM
$P_{k\|k}^j$	Covariance output of IMM
$\mathbf{Z}^{k-1}$	Measurement estimation residue at index k
w_γ	Weight for spawning targets
J_γ	Number of components to be added

m_γ	Mean state vector for spawning targets
ϕ_t	State transition matrix
Γ_t	Coefficient matrix
θ_t	Measurement of the bearing at time t
σ_r	Standard deviation of range measurement
σ_θ	Standard deviation of bearing measurement
ϕ_k	State transition matrix
μ_0	Mean of the initial state
χ	Sigma points
ξ	Propagated points
θ_m	Bearing measurement
μ_i	Model probability
κ	Decision-making threshold

LIST OF ACRONYMS/ABBREVIATIONS

ADS-B	Automatic Dependent Surveillance-Broadcast
CA	Constant Acceleration
CMKF	Converted Measurement Kalman Filter
COG	Center Of Gravity
CT	Constant Turn
CUSUM	Cumulative Sum
CV	Constant Velocity
DB-SCAN	Density-Based Spatial Clustering of Applications with Noise
DUCMKF	Decorelated Unbiased Converted Measurement Kalman Filter
EKF	Extended Kalman Filter
GM-IEKF	Gaussian Mixture Iterative Extended Kalman Filter
GM-PHD	Gaussian Mixture Probability Hypothesis Density Filter
GMM	Gaussian Mixture Model
IEKF	Iterative Extended Kalman Filter
IMM	Interacting Multiple Model
JPDA	Joint Probabilistic Data Association
KF	Kalman Filter
MHT	Multiple Hypothesis Tracking
MMSE	Minimum Mean Square Error
NN	Nearest Neighbor
OSPA	Optimal Subpattern Assignment
OTE	Object Tracking Error
OTEstd	Object Tracking Error Standard Deviation
PFE	Performance Fit Error
PHD	Probability Hypothesis Density
PHDF	Probability Hypothesis Density Filter
RMSE	Root Mean Square Error
TCAS	Traffic Alert and Collision Avoidance System

UCMKF	Unbiased Converted Measurement Kalman Filter
UKF	Unscented Kalman Filter
UT	Unscented Transform
WN	White Noise
WSS	Within-cluster Sum of Square
WT	Weight Threshold

1. INTRODUCTION

Target tracking has been one of the principal and crucial task in the field of computer vision. It is commonly used in several applications like surveillance, guidance and obstacle avoidance systems. The main tasks of target tracking systems can be listed as determining the position, velocity or acceleration of one or multiple moving targets. In a tracking system like this, sensors are needed for measurements which depend on the target. Such sensors which give such measurements can be radars, sonars, or cameras, etc. Those measurements have measurement noise and hence; often the data measured by tracking devices are not exact. One sensor can give inaccurate results on far range whereas some other can be inaccurate on short range and good measurements on far range can be obtained from those type of sensors. The measurement noise of sensors can make target tracking complicated. In order to predict the true path of moving targets, we must filter out the noise to improve the estimatation of the targets' trajectories. The noise can be listed as two types; the first type is the measurement noise which is caused by inaccuracies in the tracking device whereas the second type is the state noise that is caused by environmental factors as well as human errors. Furthermore, several factors can be the reason of noise which can be listed as imprecise radar measurements, turbulence or atmospheric interference. In addition, the estimation process can become more complicated if the measurements are taken at irregular time intervals. Since tracking of targets such as aircrafts and marine vessels is still a big problem for safety of the voyage and prevention from the possible attacks, so many methods and solutions were developed for increasing the accuracy and sensitivity of estimations for targets' trajectories.

Data-filtering target trackers are used in several cases in order to obtain an estimation of both the position and velocity of a target when the measurements are taken. They are commonly used in the control of the traffic in the air or obstacle avoidance systems. Such fields have always pursued improved accuracy in their predictions. Many different filtering methods have been proposed as a solution for target tracking.

from a sensor are in spherical coordinates, in general and this results in a non-linear measurement model with the conversion from Cartesian coordinate to spherical coordinate. The tracking problem can usually be decomposed into two subproblems such as

- Detection: Finding targets' position for the first time.
- Prediction (Tracking): Guessing the position of targets where they're going to be in the next frame

In 1940, Norbert Wiener approached the filtering problem stated above by building a model so-called minimum mean square error (MMSE) estimation by assuming the process was stationary. However, this model as known as the Wiener Filter had some limitations such as being applicable only for stationary processes and having need of solving tricky calculations in the continuous case. In 1960, Rudolf Kalman improved the Wiener Filter by adding extension for solution of non-stationary processes and simplifying the algebraic calculations. This new model was assumed great discovery in the statistical state estimation field and named after as Kalman Filter (KF). Thanks to this discovery of Rudolf Kalman, many problems that had no solution until then, especially problems related to space exploration, were solved.

Later, filtering in nonlinear systems has become more important; however, since linear system equations are required for KF, the nonlinear equations must be linearized in order to apply KF algorithm which results in the implementation of Extended Kalman Filter (EKF). Moreover, another nonlinear approach is known as Unscented Kalman Filter (UKF). After obtaining the measurement information from the radars, the next step is to correlate this position information with the movements of the vessels. Although the measurement information obtained is polar, the estimation of the movement of the vessels is made in the Cartesian coordinate system. The reason for this is that the motion models made by vessels are linear in the Cartesian system. However; as a result of this conversion, measurement errors in polar no longer lose their white and uncorrelated features. In this study, we will give information about

target tracking filters used for linear and nonlinear systems where various Kalman filter variants are used to monitor polar measurements with Cartesian coordinates and to remodel noise. We will present our study on the tracking performance of various Kalman filters in different tracking scenarios where there is both a single motion model and multiple motion models. In Chapter 2, we will discuss the previous studies on the target tracking problem with single and multiple motion models, as well as the proposed methods used for the target tracking problem. In Chapter 3, we will give a background information of the modeling of the target tracking problem and the Optimal Bayes Filter. In Chapter 4, we will examine the Kalman filter and derivative approaches used when the system is linear or nonlinear in scenarios based on a single motion model. In Chapter 5, we will study the Interacting Multiple Model (IMM) and Cumulative Sum (CUSUM) algorithms used in situations where the target motion is constantly changing and more than one motion model is valid. In Chapter 6, we will discuss the Probability Hypothesis Density Filter (PHDF) suggested in the literature for multiple target tracking and the estimation of the target number in case of tracking multiple targets. In Chapter 7, we will analyze the tracking performance of all Kalman filter variants in different scenarios and different motion models, then we present the results of our studies. In the last chapter, we will summarize the results of all these studies and give information about the works we will do in the future.

2. LITERATURE REVIEW

As an efficient application of estimating the state of a target, there are many studies proposed in the literature on target tracking with noisy measurements. Several cases like tracking a maneuvering target, decoupled tracking and tracking in cluttered environment as well as tracking both single and multiple targets have been studied in the literature. However, the proposed solutions for some problems are still not sufficient and in many practical applications it is very important to reduce the computational requirements while achieving the intended tracking performance. When we examine current target tracking algorithms in the literature, it is common to find applications using range and bearing measurements obtained from radar. Although the range and bearing measurements are in polar coordinates, target tracking algorithms are not linear because the target state is expressed in Cartesian coordinates. Despite this problem, Extended Kalman filter (EKF), Unscented Kalman filter (UKF), Unbiased Converted measurements Kalman Filter (UCMKF) and Decorrelated Unbiased Conversion Measurement Kalman Filter (DUCMKF) in different variations have been developed in order to apply to nonlinear systems in the literature.

EKF, which is one of the widely used target tracking algorithms in nonlinear systems, is based on linearizing the nonlinear measurement function by using the Taylor transformation of the measurement function [1]. As the degree of nonlinearity in the measurement function increases, the effect of the linearization approach decreases. Another variation of the EKF, the Iterative Extended Kalman filter (I-EKF), tries to make the measurement function linear over the most recent state. If, between two cycles, the amount of improvement in the prediction of the target's position decreases, the cycle is terminated.

In their study, Shen and Liu [2] compared the EKF and the UKF, which are nonlinear filtering algorithms, and found a better tracking performance of the UKF compared to the EKF in target tracking made in line with the measurements from

the radar. They showed that UKF exhibited more consistent results and the reason for this is explained that both the decrease in complexity due to the fact that the Jacobian matrix is not calculated and the better estimation of the mean and variance of the nonlinear function by means of the unscented transform increases the tracking performance of the UKF.

Mochnac, Marchevsky and Kocan [3] compared the target tracking performance of the two filtering methods regarding the KF, which is an optimal estimator for linear systems, and the EKF, which first makes the system linear and then filters, in nonlinear systems. It has been stated that the KF is the best working target tracking algorithm when the noise is Gaussian and the system is linear, and it has been shown through simulations that using the EKF has a better tracking performance due to the invalidity of the assumptions in the KF as a result of the fact that noise is not always Gaussian in real-time target tracking scenarios. UKF is another target tracking algorithm based on the principle of approaching the transformation as linear and noise as Gaussian by obtaining sigma points in systems where the noise is Gaussian and the transformation is non-linear. Compared to the EKF, it has the advantage of not requiring the Jacobian matrix and thus differential equations, however; the UKF requires more calculation steps compared to the EKF.

In cases where the nonlinearity of the EKF is high, the I-EKF has been proposed as an alternative method in order to overcome the first-order approximation errors. Specifically, the I-EKF recursively linearizes nonlinear equations to compensate for higher order terms. Zhao, Netto and Milli [4] proposed the Gaussian Mixture Iterative Extended Kalman Filter (GM-IEKF) based on the Generalized Maximum Likelihood approach in order to predict the situation dynamics in a noisy environment. The GM-IEKF dynamic state estimator proposed in this study is able to monitor system transitions faster and more reliably than the traditional EKF and UKF, thanks to its robustness against observation outliers and the batch regression form in which observations and predictions are processed simultaneously. is observed.

Since the dynamic equations are generally linear and independent, the Cartesian coordinate system is used to define the dynamics of a target. One of the most widely used tools for linear state estimation from noisy measurements is using KF. However, if the dynamic model of a target location is not expressed in linear equations, the tracking problem is associated with nonlinear estimation. One way to generalize the methods we apply to linear estimation is to make KF nonlinear recursive equations, this extended algorithm is called EKF. Although these two algorithms form the basis of target tracking algorithms, they have important flaws that need to be addressed.

In [5], the authors proposed a robust version of a linear KF for target tracking in R^2 and R^3 by using converted radar sensor based measurement systems. They proved that the state estimation error is bounded in a probabilistic sense. The performance of their robust filter was shown to perform favorably and it had much simpler implementation compared to another existing filters.

Li, Corchado, Chen and Bajo [6] accepted the tracking problem as an optimization problem. In their study, it was desired to find a trajectory function which fits the observation the best over a sliding window. For smoothing, filtering and prediction, the trajectory function was used. Their approach was applicable to target motion patterns like an aircraft where the sensor observation was limited except the information of smoothly manuevering target. However, sharply manuevering target were not considered and the authors endorsed sliding window method for scenarios where the target has a complex trajectory.

In [7], the authors conducted a study on an ADS-B-based collision avoidance radar to detect cooperative and non-cooperative targets in the airspace. The radar system in this paper was effective against almost all air traffic with or without ADS-B (Automatic Dependent Surveillance-Broadcast) and TCAS (Traffic Alert and Collision Avoidance System) which are transponder-based collison avoidance systems. Moreover, the radar system was based on the standard ADS-B signal without needing an additional spectrum and didn't require any modifications to the existing air traffic control

system which makes getting the Federal Aviation Administration's approval easier.

Patel and Takore [8] conducted the study of a visual surveillance system with detecting a moving object as well as tracking capability was conducted. The authors proposed an implementation on standard surveillance dataset using Kalman Filter in order to track any single moving object. Their applications could be applied to any computer vision application for detecting and tracking a moving object. Additionally, their methods worked on videos of both indoor and outdoor environments captured using a static camera which alleviates complex background conditions.

Shantaiya, Verma and Mehta [9] presented a tracking method for processing video data to perform tracking by a machine vision system. The authors proposed a method to track multiple objects simultaneously and it could control problems of multi-object tracking like appearance and disappearance of objects as well as missing of an object. According to their study, improved optical flow algorithm was found to be more promising since it's more accurate and efficient in terms of computation time.

In [10], a KF was used for simultaneous tracking in two dimensional image coordinates and three dimensional world coordinates for each camera. In case of occlusions, each camera's two dimensional or three dimensional trackers share information to improve both trajectory prediction and performance.

Kang et al. [11] proposed a method in which they used a combination of motion model and appearance model to keep track of each individual. Motion models were obtained using the Kalman Filter, which estimates the position in both 2D and 3D. In this study, target tracking was done by maximizing a joint probability model, considering both the appearance and the motion model.

One of the target tracking algorithms used for nonlinear transformations is the Converted Measurement Kalman Filter (CMKF), in which the transformed criteria are used and the Kalman equations are revised accordingly. In our problem, this means

that the filtering is done in Cartesian coordinates by converting the polar measurements we have into Cartesian coordinates. However, due to the nature of the noise, the transformed measurements are biased and therefore the covariance matrix is correlated. For this reason, the Unbiased Converted Measurement Kalman Filter (UCMKF) and the Decorrelated Converted Measurement Kalman Filter (DUCMKF) have been proposed. All three target tracking algorithms are based on using the linear KF after converting the range and bearing measurements from the radar to Cartesian coordinates before filtering.

Tracking the maneuvering target requires an accurate definition of the dynamic model of the filter. In such a case, the common approach in the literature is to use filters compatible with the target's motion model or multi-model tracking filters. In his thesis, Murat Efe [12] examined the coherent tracking methods on target movement when the target maneuvers. We see that multi-model tracking performance is improved in studies where IMM and Joint Probabilistic Data Association (JPDA) are used together [13,14]. Although Multiple Hypothesis Tracking (MHT) [15, 16] gave a better result in terms of tracking performance, it is not used much in practice due to its complexity and high memory requirement. Since nearly optimal solutions are sought for the target tracking problem in general, the IMM-JPDA method proposed by M. De Feo has been compared with the Nearest Neighborhood (NN) and MHT data association methods [13].

3. MODELING OF TRACKING PROBLEM

3.1. Formulation of the Target Tracking Problem

In two-dimensionel (2D) plane, the dynamic model of the target can be expressed in Cartesian coordinate system as

$$X_{t+1} = \phi_t X_t + \Gamma_t W_t \tag{3.1}$$

where $\mathbf{X}_t = \left[x_t, y_t, \dot{x}_t, \dot{y}_t\right]$ is the motion of the target at time t, x_t and y_t are location components whereas $\dot{x}_t$ and $\dot{y}_t$ are the velocity components of the target. The state transition matrix and the coefficient matrices are stated with ϕ_t and Γ_t, respectively. In addition, W_t corresponds to White Gaussian noise process with zero mean and covariance matrix Q_t. The measurement of the position vector can be expressed as $Z_t^P = \left[r_t, \theta_t\right]$ where r_t is the measurement of the range and θ_t is the measurement of the bearing. The measurement equation can be defined as

$$Z_t^P = h_t^P(X_t) + V_t^P \tag{3.2}$$

where $h_t^P(X_t)$ is the measurement function of the position such that

$$h_t^P(X_t) = \begin{bmatrix} \sqrt{x_t^2 + y_t^2} \\ arctan(y_t/x_t) \end{bmatrix} \tag{3.3}$$

with the assumption that the radar is located at the origin. In Equation 3.2, V_t^P is White Gaussian measurement noise vector with zero mean, having the covariance matrix $R^P = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\theta \end{bmatrix}$ where σ_r and σ_θ represent the standard deviations of range and bearing measurements, respectively.

3.2. Bayes' Theorem

Bayes' Theorem is the most widely used probability-based classification method among probability theories. This theorem, named after the 18th century English mathematician Thomas Bayes, is the mathematical formula used to determine conditional probability. Bayes' Theorem, which makes use of conditional probability, helps to find out what probabilities and reasons the result may have arisen while searching for the cause of an effect. This theorem is also a very useful method in target tracking problem because it suggests that the next location of the target will logically be close to this location, based on the information about the previous location of the target. The Bayes' Theorem can be formulated expressed as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.4}$$

where P(A) is the prior probability, P(B) is the marginal probability and $P(B|A)$ is the likelihood probability.

3.3. Optimal Bayes Filtering

In target tracking application, Bayes Filter adapts the Bayes Rule into estimating the states of the target based on a sequence of (partial) observations from a sensor. As the sensor measurements are not perfect and contain noise, we cannot be sure of the exact location or state of a target. That's the reason that we employ probability density functions that help us quantify our uncertainty. We must also pay attention to the fact that the state vector can have more than one elements and thus we will be dealing with a multivariate probability density function.

If we want to make use of the Bayes' Theorem, one of the things we need is the prior probability, P(A). Here, the prior probability will represent our knowledge of where we think our target could be, while taking into account the uncertainty associated with that guess. We can imagine that our target has reached the current state from the

previous state following some rules that we know from before. For example, for a truck traveling on a road, we can say that at the next time step the truck maintained the same velocity and thus covered the corresponding distance. In the Bayes Filter, these rules are considered as the Dynamical Model that the truck has to follow to achieve the next state, and the Dynamical Model helps us predict where our target could be. We can express the Dynamical Model of a target as

$$P_{k|k-1}(x) = \int_0^{k-1} f_{k|k-1}(x, y) P_{k-1}(y)\, dy \tag{3.5}$$

where $P_{k|k-1}(x)$ is the probablity that the target is at the x-axis when it transitioned from $(k-1)^{th}$ time step to k^{th}, $f_{k|k-1}(x, y)$ is the model that dictates how the target propagated over time and $P_{k-1}(y)$ is the probability that the target was at y-axis at $(k-1)^{th}$ time step.

Equation 3.5 is called the Chapman-Kolmogorov equation. With the help of the Dynamical Model as represened by the Chapman-Kolmogorov equation, we propagate our target and estimate the prior probability, $P_{k|k-1}(x)$, of our target being at x coordinate at time step k.

The Observational Model is also required to relate the state of the target to the observations of the target, while taking into account the uncertainty associated with it. According to the Bayes' Theorem, the Observational Model helps us to find the likelihood probability represented by $g_k(z|x)$. At the end, our Bayes Filter takes the following form:

$$P_k(x|z) = \frac{P_{k|k-1}(x) g_k(z|x)}{\int_0^{k-1} P_{k|k-1}(y) g_k(z|y)\, dy} \tag{3.6}$$

where the denominator term in this equation corresponds to the Marginal Probability, P(B), in the Bayes Filter. This is just a normalization term which helps us account for all the propagation of that the target underwent since its birth.

4. SINGLE MODEL TARGET TRACKING

4.1. Target Tracking in Linear Systems

4.1.1. Kalman Filter

In the study of linear filtering, the KF is a recursive linear filtering model used for estimating targets' trajectories [17]. Various cases such as maneuvering targets, multiple targets and collision avoidance can be examined using KF algorithm. KF is successful in smoothing random deviations from the true path of the targets which improves its ability to predict the path of each target as more measurements from the tracker are processed. Furthermore, KF which enables making estimations from noisy data and observations estimates the position of a target and then updates this decision by making probabilistic inference from previous data. Therefore; it increases the success rate as much as possible using every feature of existing information. Furthermore, since KF uses only information just before estimation, its background calculations are computationally efficient.

The KF is the optimal filter in the sense of least mean squared error for track prediction and it has common usage area from the space shuttle to the patriot missile system to the New York stock exchange. However, it is usually utilized for filtering the noisy observations and then producing estimates of position and velocity of the moving target. After the estimation process, KF updates its decision by comparing it with the observation for that state. This update is carried out by using the residual between the estimation and the observation, and by the Kalman Gain which is the weight factor for that residue. Finally covariance matrix is updated based on the Kalman Gain and the predicted covariance matrix. Working principle of the KF which is a combination of a predictor and a filter where both of them are linear systems can be explained with

the following KF equations: The dynamic model of a system is described as

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1} \tag{4.1}$$

where x_k denotes the state vector at time k, Φ_k represents the state transition matrix, w_k is the Gaussian zero-mean white process noise. Moreover, the measurement model can be expressed as

$$z_k = H_k x_k + v_k \tag{4.2}$$

where z_k denotes the corresponding measurement, H_k represents the measurement matrix and v_k is the Gaussian zero-mean white measurement noise. Next step is estimating the state such that

$$\hat{x}_k^- = \Phi_{k-1} \hat{x}_{k-1}^+ \tag{4.3}$$

where $-$ in the superscript represents estimation without update whereas $+$ represents estimation after update and hat symbol denotes estimations. The next step is covariance matrix estimation such that

$$P_k^- = \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + Q_{k-1} \tag{4.4}$$

where P_k^- denotes the covariance matrix. After the state estimation and covariance matrix estimation, the Kalman Gain is calculated using the following equation:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}. \tag{4.5}$$

According to calculated Kalman Gain, both the state estimate and covariance matrix

are updated as following:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - H_k\hat{x}_k^-] \tag{4.6}$$

$$P_k^+ = [I - K_k H_k]P_k^-. \tag{4.7}$$

The predictor estimates the state vector which contains all required information of the system to describe it. Many systems may contain hidden measured variables as well as states and this makes estimation of the state vector obtain better knowledge of the state. There will always be unmodeled dynamics due to unknown properties of a physical system, which can qualify as unmodeled dynamics noise. Noise can be described by White Gaussian noise, which is a probabilistic process and includes random variables. The recursive KF is a probabilistic extension of Bayes estimation and it consists of five equations where the two of them stands for updating and predicting whereas the three of them stands for updating measurement, i.e. the corrector step. Implementing the KF requires some knowledge of the system to be estimated, with its measurements, or observations.

4.2. Target Tracking in Non-linear Systems

4.2.1. Extended Kalman Filter

For nonlinear systems, one of the approaches for tracking task is using The Extended Kalman Filter [18]. In this filter, local linearizations of nonlinear system and measurement functions are obtained by Taylor Series. Following that the KF equations are used to predict state vector and covariance matrix. Considering the following nonlinear system which is described by the difference equation and the observation model with additive noise

$$\mathbf{x}_k = \mathbf{f}_{k-1}(x_{k-1}, u_{k-1}, \mathbf{w}_{k-1}) \tag{4.8}$$

$$\mathbf{z}_k = \mathbf{h}_k(x_k, \mathbf{v}_k) \tag{4.9}$$

where both $\mathbf{w}_k$ and $\mathbf{v}_k$ are WN with zero mean and covariances with Q_k and R_k, respectively. Initially, we only have the mean and the covariance of the initial state, and therefore; we can initialize the system as

$$\mu_0 = E[\mathbf{x}_0] \tag{4.10}$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \mu_0)(\mathbf{x}_0 - \mu_0)^T] \tag{4.11}$$

where the initial state x_0 is a random vector having μ_0 and $\mathbf{P}_0$ as the known mean and covariances, respectively. Our assumption is about the random vector w_k which catches the uncertainties in the model and $\mathbf{v}_k$ is the measurement noise where w_k and $\mathbf{v}_k$ are zero mean WN with known variances and both of them are uncorrelated. The two random vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are also not correlated such that $E[\mathbf{w}_k\mathbf{v}_j^T] = 0$ for all k and j. For each time step k, we compute the partial derivative matrices (Jacobian) as

$$F_{k-1} = \left.\frac{\partial f_{k-1}}{\partial x}\right|_{\hat{x}_{k-1}^+} \tag{4.12}$$

$$L_{k-1} = \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{x}_{k-1}^+}. \tag{4.13}$$

In the next step, we perform the time update of the state estimate and the covariance of the estimation error as

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^T \tag{4.14}$$

$$\mathbf{x}_k^- = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^+, \mathbf{u}_{k-1}, 0). \tag{4.15}$$

Following the model predictor, we compute the partial derivative matrices, i.e. Jacobians for measurement function as

$$H_k = \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}_k^-} \tag{4.16}$$

$$M_k = \left.\frac{\partial h_k}{\partial v}\right|_{\hat{x}_k^-}. \tag{4.17}$$

In the final step, we measurement update of the state estimation and the estimation error covariance are performed as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \Big(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T \Big)^{-1} \tag{4.18}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, 0) \tag{4.19}$$

$$\mathbf{P}_k = \Big(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \Big) \mathbf{P}_k^- \tag{4.20}$$

where the Kalman Gain is denoted by $\mathbf{K}_k$ in these equations. In this section, we extend the KF which is an linear optimal state estimator to real world problems where the system cannot be described in a linear manner. Since a lot of real world models are not linear, EKF is better option compared to KF and provides better tracking performance in nonlinear systems. However; EKF is not optimal and the Jacobians might diverge if the nonlinearities are large. On the other hand, EKF is highly efficient and the complexity of the EKF algorithm is $O(k^{2.376} + n^2)$, i.e. polynomial in measurement dimensionality k and the state dimensionality n. In the next section, we will study another KF variant which uses sigma points instead of Taylor series expansion and Jacobians. As a result, we expect to get better approximations as well as better tracking performance by using further points to approximate nonlinearity.

4.2.2. Unscented Kalman Filter

Another approach for tracking in nonlinear systems, namely the Unscented Filter (UKF) uses the unscented transform (UT) method to estimate the distribution of the

new state [19]. Here, the sigma points are carefully selected and propagated through time, and final state and covariance estimations are computed as a weighted sum of these points. UKF consists of two steps which are prediction and update steps.

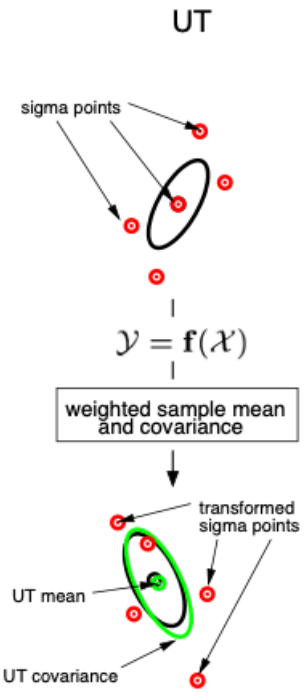

Figure 4.1. The Unscented Transform (UT).

In the prediction step, sigma points $\boldsymbol{\chi}$ are calculated as follows:

$$\boldsymbol{\chi}_{k-1}^{(0)} = \mathbf{m}_{k-1} \tag{4.21}$$

$$\boldsymbol{\chi}_{k-1}^{(i)} = \mathbf{m}_{k-1} + \sqrt{n+\lambda}(\sqrt{\mathbf{P}_{k-1}})_i \qquad i = 1, ..., n \tag{4.22}$$

$$\boldsymbol{\chi}_{k-1}^{(i+n)} = \mathbf{m}_{k-1} - \sqrt{n+\lambda}(\sqrt{\mathbf{P}_{k-1}})_i \qquad i = 1, ..., n \tag{4.23}$$

where $\mathbf{m}_{k-1}$ and $\mathbf{P}_{k-1}$ are the state mean and the covariance matrix from the previous time step $(k-1)$, respectively. n is the state dimension and λ is calculated with control parameters α and κ as in

$$\lambda = \alpha^2(n+\kappa) - n. \tag{4.24}$$

In the following step, these sigma points are propagated with nonlinear state transition function $\boldsymbol{f}$ using the following equation:

$$\hat{\boldsymbol{\chi}}_k^{(i)} = \boldsymbol{f}\boldsymbol{\chi}_{k-1}^{(i)} \qquad i = 0, ..., 2n. \tag{4.25}$$

Lastly in the prediction step, the predicted state mean $\mathbf{m}_k^-$ and the predicted state covariance $\mathbf{P}_k^-$ are computed as

$$\mathbf{m}_k^- = \sum_{i=1}^{2n} W_i^{(m)} \hat{\boldsymbol{\chi}}_k^{(i)} \tag{4.26}$$

$$\mathbf{P}_k^- = \sum_{i=1}^{2n} W_i^{(c)} (\hat{\boldsymbol{\chi}}_k^{(i)} - \mathbf{m}_k^-)(\hat{\boldsymbol{\chi}}_k^{(i)} - \mathbf{m}_k^-)' + \mathbf{Q}_{k-1} \tag{4.27}$$

where $W_i^{(m)}$ and $W_i^{(c)}$ are the weights for the mean and the covariance calculations can be obtained as

$$W_0^{(m)} = \frac{\lambda}{\lambda + n} \tag{4.28}$$

$$W_0^{(c)} = \frac{\lambda}{\lambda + n} + (1 - \alpha^2 + \beta) \tag{4.29}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(\lambda + n)} \qquad i = 1.....2n \tag{4.30}$$

where the parameter β is selected as 2 for a Gaussian distribution. In the update step of the UKF, new sigma points are calculated using $\mathbf{m}_k^-$ and $\mathbf{P}_k^-$ which are the predicted state mean and the predicted state covariance, respectively:

$$\boldsymbol{\chi}_k^{-(0)} = \mathbf{m}_k^- \tag{4.31}$$

$$\boldsymbol{\chi}_k^{-(i)} = \mathbf{m}_k^- + \sqrt{n + \lambda}(\sqrt{\mathbf{P}_k^-})_i \qquad i = 1.....n \tag{4.32}$$

$$\boldsymbol{\chi}_k^{-(i+n)} = \mathbf{m}_k^- - \sqrt{n + \lambda}(\sqrt{\mathbf{P}_k^-})_i \qquad i = 1.....n. \tag{4.33}$$

The predicted measurement mean $\boldsymbol{\mu}_k$ and the predicted measurement covariance $\mathbf{S}_k$ are calculated from these propagated points $\hat{\boldsymbol{\xi}}$ using the following equations:

$$\boldsymbol{\mu}_k = \sum_{i=1}^{2n} W_i^{(m)} \hat{\boldsymbol{\xi}}_k^{(i)} \tag{4.34}$$

$$\mathbf{S}_k = \sum_{i=1}^{2n} W_i^{(c)} (\hat{\boldsymbol{\xi}}_k^{(i)} - \boldsymbol{\mu}_k)(\hat{\boldsymbol{\xi}}_k^{(i)} - \boldsymbol{\mu}_k)' + \mathbf{R}. \tag{4.35}$$

The cross-correlation matrix $\mathbf{C}_k$ is computed as

$$\mathbf{C}_k = \sum_{i=1}^{2n} W_i^{(c)} (\hat{\boldsymbol{\chi}}_k^{-(i)} - \mathbf{m}_k^-)(\hat{\boldsymbol{\xi}}_k^{(i)} - \boldsymbol{\mu}_k)' \tag{4.36}$$

to obtain Kalman gain $\mathbf{K}_k$ as follows:

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1}. \tag{4.37}$$

Finally the updated mean and the updated covariance can be achieved with the equations below:

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k[\,\mathbf{Z}_k - \boldsymbol{\mu}_k] \tag{4.38}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k'. \tag{4.39}$$

When we compare KF, UKF and EKF; we can say that KF is an optimal filter for systems where the dynamic model of a target can be expressed as a linear equation. However, for nonlinear systems, EKF or UKF is a better choice compared to KF. Furthermore, if we compare UKF and EKF, we can say that UKF might be better than EKF due to the fact that UKF does not make a linear approximation at one point but it uses further points in order to approximate nonlinearity. As a result, the UT can catch the higher moments better than EKF which uses Taylor series approximations to approximate the nonlinearity.

4.2.3. Converted Measurement Kalman Filter

Another method used for nonlinear transformations is the Converted Measurement Kalman Filter (CMKF) [20], in which the converted measurements are used and the Kalman equations are revised accordingly. In our case, this means passing polar measurements to Cartesian and filtering is done in Cartesian coordinates. They are expressed mathematically as

$$Z^{CM} = \begin{bmatrix} r_k cos(\theta_k) \\ r_k sin(\theta_k) \end{bmatrix}, R^{CM} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \tag{4.40}$$

where Z^{CM} corresponds to the converted measurements and R^{CM} corresponds to the covariance of the converted measurements. However, due to the nature of the noise, our converted measurements are biased and the covariance matrix is correlated. For this reason, Unbiased Converted Measurement Kalman Filter (UCMKF) and Decorrelated Unbiased Converted Measurement Kalman Filter (DUCMKF) have been proposed.

4.2.3.1. Unbiased Converted Measurement Kalman Filter. The CMKF algorithm is a common tracking algorithm for the target state in Cartesian coordinate system and the measurement in the polar coordinate system. However, the CMKF has a conversion bias; hence, if we adopt the conversion bias to the KF, then this results in performance degradation. The Unbiased Converted Measurement Kalman Filter (UCMKF) is proposed in [21] to eliminate the bias by multiplicative compensation.

For the standard conversion, the range measurement r_m and bearing measurement θ_m in polar coordinates are converted into Cartesian coordinate measurements x_m^c and

y_m^c as follows before they are processed with the KF such that

$$r_m = r + \hat{r} \tag{4.41}$$

$$\theta_m = \theta + \hat{\theta} \tag{4.42}$$

$$\begin{bmatrix} x_m^c \\ y_m^c \end{bmatrix} = E \begin{bmatrix} (r+\hat{r})cos(\theta+\hat{\theta}) \\ (r+\hat{r})sin(\theta+\hat{\theta}) \end{bmatrix} = e^{-\sigma_\theta^2/2} \begin{bmatrix} rcos(\theta) \\ rsin(\theta) \end{bmatrix} \tag{4.43}$$

where r and θ are the true values of range and bearing. The measurement errors $\hat{r}$ and $\hat{\theta}$ are assumed to be independent Gaussian White Noise process with zero mean and standard deviation σ_r and σ_θ. In order to get the unbiased conversion,

$$\mathbf{Z}_k^{uP} = \begin{bmatrix} x_k^u \\ y_k^u \end{bmatrix} = \begin{bmatrix} \lambda_\theta^{-1} r_k cos(\theta_k) \\ \lambda_\theta^{-1} r_k sin(\theta_k) \end{bmatrix}, \lambda_\theta = e^{-\sigma_\theta^2/2} \tag{4.44}$$

where the converted measurement is denoted as $\mathbf{Z}_k^{uP}$. The covariance matrix of the unbiased conversion we use in this case is:

$$\mathbf{R}_u^P = \begin{bmatrix} R_{u11}^P & R_{u12}^P \\ \\ R_{u21}^P & R_{u22}^P \end{bmatrix} \tag{4.45}$$

where the entries of the covariance matrix are calculated using the following equations:

$$R_{u11}^P = (\lambda_\theta^{-2} - 2) r_k^2 cos^2\theta_k + \frac{1}{2}(r_k^2 + \sigma_r^2)(1 + \lambda'_\theta cos2\theta_k) \tag{4.46}$$

$$R_{u12}^P = R_{u21}^P = (\lambda_\theta^{-2} - 2) r_k^2 sin\theta_k cos\theta_k + \frac{1}{2}(r_k^2 + \sigma_r^2)\lambda'_\theta sin2\theta_k \tag{4.47}$$

$$R_{u22}^P = (\lambda_\theta^{-2} - 2) r_k^2 sin^2\theta_k + \frac{1}{2}(r_k^2 + \sigma_r^2)(1 - \lambda'_\theta cos2\theta_k) \tag{4.48}$$

where $\lambda'_\theta = e^{-2\sigma_\theta^2}$. In order to obtain UCMKF, $\mathbf{Z}_k^{uP}$ and $\mathbf{R}_u^P$ are combined with the KF.

4.2.3.2. Decorrelated Unbiased Converted Measurement Kalman Filter. In order to remove the correlation between the measurement noise and the covariance matrix of the conversion measurement, the DUCKMF substitutes the current measurement with the one step state estimate. Like the conversions in UCMKF, the unbiased conversions in DUCMKF can be expressed as

$$\mathbf{Z}_k^{dP} = \begin{bmatrix} x_k^u \\ y_k^u \end{bmatrix} = \begin{bmatrix} \lambda_\theta^{-1} r_k cos(\theta_k) \\ \lambda_\theta^{-1} r_k sin(\theta_k) \end{bmatrix}, \lambda_\theta = e^{-\sigma_\theta^2/2} \tag{4.49}$$

where $\mathbf{Z}_k^{dP}$ is the unbiased converted measurement. Conditional measurement values are used in DUCMKF in order to extinguish the correlation. Range and bearing measurement estimates are expressed using $r_{k+1|k}$ and $\theta_{k+1|k}$ such as

$$r_{k+1|k} = \sqrt{x_{k+1|k}^2 + y_{k+1|k}^2} \tag{4.50}$$

$$\theta_{k+1|k} = arctan \frac{y_{k+1|k}}{x_{k+1|k}}. \tag{4.51}$$

The state estimation is denoted as $\hat{\mathbf{X}}_{k+1|k}$ whereas the estimation at k+1 is denoted as $\hat{\mathbf{X}}_{k+1|k} = \left[x_{k+1|k}, y_{k+1|k}, \dot{x}_{k+1|k}, \dot{y}_{k+1|k}\right]'$. The standard deviations of the range and bearing estimates are $\sigma_{r,k+1}^2$ and $\sigma_{\theta,k+1}^2$, respectively and are expressed by

$$\sigma_{r,k+1}^2 = \frac{1}{r_{k+1|k}^2} \begin{bmatrix} x_{k+1|k} & y_{k+1|k} \end{bmatrix} \boldsymbol{P}_{k+1|k}^P \begin{bmatrix} x_{k+1|k} & y_{k+1|k} \end{bmatrix}' \tag{4.52}$$

$$\sigma_{\theta,k+1}^2 = \frac{1}{r_{k+1|k}^4} \begin{bmatrix} -y_{k+1|k} & x_{k+1|k} \end{bmatrix} \boldsymbol{P}_{k+1|k}^P \begin{bmatrix} -y_{k+1|k} & x_{k+1|k} \end{bmatrix}' \tag{4.53}$$

where the covariance matrix of the estimation is denoted as $\mathbf{P}_{k+1|k}^P$. The covariance matrix of DUCMKF can be expressed as following matrix:

$$\mathbf{R}_{du}^P = \begin{bmatrix} R_{du11}^P & R_{du12}^P \\ \\ R_{du21}^P & R_{du22}^P \end{bmatrix} \tag{4.54}$$

where the entries of this covariance matrix is calculated using the following equations:

$$\begin{aligned} R^P_{du11} &= \tfrac{1}{2}(r^2_{k+1|k} + \sigma_r^2 + \sigma^2_{k+1|k}) \\ &\quad [1 + cos(2\theta_{k+1|k})e^{-2\sigma^2_\theta}e^{-2\sigma^2_{\theta,k+1|k}}]e^{-\sigma^2_\theta} \\ &\quad -\tfrac{1}{2}(r^2_{k+1|k} + \sigma^2_{r,k+1|k})[1 + cos(2\theta_{k+1|k})e^{-2\sigma^2_{\theta,k+1|k}}] \end{aligned} \tag{4.55}$$

$$\begin{aligned} R^P_{du12} = R^P_{du21} &= \tfrac{1}{2}(r^2_{k+1|k} + \sigma_r^2 + \sigma^2_{k+1|k}) \\ &\quad [sin(2\theta_{k+1|k})e^{-2\sigma^2_\theta}e^{-2\sigma^2_{\theta,k+1|k}}]e^{-\sigma^2_\theta} \\ &\quad -\tfrac{1}{2}(r^2_{k+1|k} + \sigma^2_{r,k+1|k})[sin(2\theta_{k+1|k})e^{-2\sigma^2_{\theta,k+1|k}}] \end{aligned} \tag{4.56}$$

$$\begin{aligned} R^P_{du22} &= \tfrac{1}{2}(r^2_{k+1|k} + \sigma_r^2 + \sigma^2_{k+1|k}) \\ &\quad [1 - cos(2\theta_{k+1|k})e^{-2\sigma^2_\theta}e^{-2\sigma^2_{\theta,k+1|k}}]e^{-\sigma^2_\theta} \\ &\quad -\tfrac{1}{2}(r^2_{k+1|k} + \sigma^2_{r,k+1|k})[1 - cos(2\theta_{k+1|k})e^{-2\sigma^2_{\theta,k+1|k}}]. \end{aligned} \tag{4.57}$$

By combining the $\mathbf{Z}_k^{dP}$ and $\mathbf{R}_{du}^P$ with the KF, the DUCMKF can be obtained. The DUCMKF can be obtained through combining the converted measurement $\mathbf{Z}_k^{dP}$ and the covariance matrix $\mathbf{R}_{du}^P$ with the KF. Despite the feature of DUCMKF that keeps away from the correlation between measurement error covariance matrix and measurement noise, we see that the state estimation error may also occur. When the state prediction error is greater than the measurement error, the performance of UCMKF is better than that of DUCMKF. This can be observed when the measurement error is small or when there is a mismatch between the actual movement of the target and the movement pattern.

5. MULTIPLE MODEL TARGET TRACKING

As we mentioned at the beginning, the movements of vessels are generally modeled as constant velocity motion, constant acceleration motion or constant rotational motion. It is better to use all models in interaction with each other, rather than using one of these models, when it is not known which movement the vessel makes or when it is considered that the vessels can make several of these movements during the whole journey. For this purpose, two different methods using the Multiple Kalman Model are proposed: the first is Cumulative Sum (CUSUM), which chooses one of the models according to the log-likelihood ratio, and the other is the Interacting Multiple Models (IMM) method that establishes a Markovian connection between model transitions and interacts with each other [22].

5.1. Interacting Multiple Model

IMM has a general structure that allows it to work with the interaction of more than one Kalman Filter. Unlike CUSUM, instead of using the output of the most likely of the filters at any given moment, the output of all filters is mixed according to their likelihood functions. In cases where the target is maneuvering, IMM is one of the most effective methods used in target tracking. This method runs multiple filters with different process noise or different state matrix in parallel at all time steps. Like other multi-model algorithms, although the IMM consists of many filters, the transition between models is performed on a probabilistic basis, keeping the number of assumptions between filters and observations constant. Each iteration of the IMM algorithm consists of three basic steps: state estimation, filter probability calculations, and estimation combination.

The equations for the Converted Measurement Kalman Filter for state estimations can be written as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.1}$$

$$\mathbf{X}^j_{k|k-1} = \mathbf{F}^j_{k-1}\mathbf{X}^{0j}_{k-1|k-1} \tag{5.2}$$

$$\mathbf{P}^j_{k|k-1} = \mathbf{F}\mathbf{P}^{0j}_{k-1|k-1}\mathbf{F}^T + \mathbf{B}^j\mathbf{Q}\mathbf{B}^{\mathbf{j}T} \tag{5.3}$$

$$\mathbf{K}^j_k = \mathbf{P}^j_{k|k-1}\mathbf{H}^{\mathbf{j}}{}^T_k(\mathbf{H}^{\mathbf{j}}{}_k\mathbf{P}^j_{k|k-1}\mathbf{H}^{\mathbf{j}}{}_k + \mathbf{R}_k)^{-1} \tag{5.4}$$

$$\mathbf{X}^j_{k|k} = \mathbf{X}^j_{k|k-1} + \mathbf{K}^j_k(\mathbf{Z}^{CM}_k - \mathbf{H}^{\mathbf{j}}{}_k\mathbf{X}_{k|k-1}) \tag{5.5}$$

$$\mathbf{P}^j_{k|k} = [\mathbf{I} - \mathbf{K}^j_k\mathbf{H}^{\mathbf{j}}{}_k]\mathbf{P}^j_{k|k-1} \tag{5.6}$$

The input of each filter here is found by the outputs of the other filters and the transition probabilities:

$$\hat{x}_{0j}(k|k) = \sum_{i=1}^{m} x_i(k|k)\mu_{ij}(k|k) \tag{5.7}$$

$$\mu_{i|j}(k|k) = \frac{1}{\bar{c}_j}p_{ij}\mu_i(k) \tag{5.8}$$

where $\bar{c}_j$ is the normalization coefficient, μ_i is the model probability, p_{ij} is the model transition probability and m is the number of models. The calculation of likelihood function can be found as follows:

$$\Lambda_j(k) = \frac{1}{\sqrt{2\pi|S_j(k)|}}e^{\frac{-1}{2}\bar{z}^T_j(k)S_j(k)^{-1}\bar{z}_j(k)} \tag{5.9}$$

where $\bar{z}_j$ is the converted measurement and S_j stands for the covariance matrix. The probability of each model is updated using the simulation outputs:

$$\mu_j(k+1) = \frac{1}{c}\Lambda_j(k+1)\bar{c}_j \tag{5.10}$$

Finally, for any j Kalman filter, $X^j_{k|k}$ and $P^j_{k|k}$ represent the estimation and covariance outputs, respectively. The mixture of outputs is obtained as

$$\hat{x}(k+1|k+1) = \sum_{j=1}^{m} \hat{x}_j(k+1|k+1)\mu_j(k+1) \tag{5.11}$$

$$\hat{P}(k+1|k+1) = \sum_{j=1}^{m} \hat{P}_j(k+1|k+1)\mu_j(k+1). \tag{5.12}$$

5.2. Cumulative Sum

CUSUM detector is a sequential detection algorithm to exchange between different types of filter models. The standard CUSUM detector is used as follows:

$$L_k = max\left\{L_{k-1} + log\frac{p(\tilde{\mathbf{Z}}_k \mid H_1, \mathbf{Z}^{k-1})}{p(\tilde{\mathbf{Z}}_k \mid H_0, \mathbf{Z}^{k-1})}, 0\right\}, L_0 = 0 \tag{5.13}$$

where $\mathbf{Z}^{k-1}$=$(Z_1, Z_2....Z_{k-1})$ is the measurement estimation residue at index k. The conditional probability $p(. \mid .)$ represents the hypothesis H_0 that the vessel does not return, and H_1 represents the hypothesis that the vessel turns. Decision making rules are as follows:

1)If $L_k \geq \kappa$, H_1 is accepted and a Constant Turn Kalman Filter (CTKF) is used;

2)If $L_k < \kappa$, H_0 is accepted and Constant Acceleration or Constant Speed Kalman Filter is used. Here, κ, is chosen as the decision-making threshold.

6. MULTI-TARGET TRACKING TECHNIQUES

6.1. The Probability Hypothesis Density Filter

Single target tracking algorithm can be used on multiple targets with a solution to measurement track association problem. The Probability Hypothesis Density Filter(PHDF) is one of the multitarget tracking algorithms used in the literature [23]. It can estimate the states of unknown number of targets given a measurement set which has false alarms and missing detections.

In the PHDF, only the first order moment of the multi-target posterior distribution, namely PHD, is used for estimation, different from other Multi-target Bayes Filters. For a linear Gaussian system, two closed form solutions of PHD algorithm are presented. One of them is Gaussian Mixture Probability Hypothesis Density Filter (GM-PHDF). In this filter, the PHD is expressed with a Gaussian Mixture Model. Therefore; the KF equations can be used in state estimation.

Input of the GM-PHDF for time step k is a Gaussian Mixture Model given in (6.1) which consists of Gaussian components, belongs to previous time step which has state vectors $\mathbf{m}_{k-1}$, covariance matrices $\mathbf{P}_{k-1}$ and an associated weights w_{k-1} to each Gaussian component. Superscript i indicates that information belongs to i'th Gaussian component below:

$$\{w_{k-1}^{i}, \mathbf{m}_{k-1}^{i}, \mathbf{P}_{k-1}^{i}\},\ i = 1, ..., J_{k-1}. \tag{6.1}$$

In each time step, additional Gaussian components for spawned and born targets, which have their own mean vectors, covariance matrices and weights are added to Gaussian Mixture Model.

In (6.2), J_γ represents the number of component to be added, $\mathbf{m}_\gamma$ is the mean state vector, $\mathbf{P}_\gamma$ is the covariance matrix and w_γ is the weight for spawning targets.

$$w^i_{k|k-1} = w_\gamma \tag{6.2}$$

$$\mathbf{m}^i_{k|k-1} = \mathbf{m}_\gamma \tag{6.3}$$

$$\mathbf{P}^i_{k|k-1} = \mathbf{P}_\gamma \tag{6.4}$$

where $i = J_{k-1} + 1, ..., J_{k-1} + J_\gamma$. Same values for new born targets are given in (6.5) with subscript β.

$$w^i_{k|k-1} = w^l_{k-1} w_\beta \tag{6.5}$$

$$\mathbf{m}^i_{k|k-1} = \mathbf{F}\mathbf{m}^l_{k-1} \tag{6.6}$$

$$\mathbf{P}^i_{k|k-1} = \mathbf{Q} + \mathbf{F}\mathbf{P}^l_{k-1}\mathbf{F}^T \tag{6.7}$$

where $i = J_{k-1} + J_\gamma + 1, ..., J_{k-1} + J_\gamma + J_\beta$ and $l = 1, ..., J_{k-1}$. Following the addition of new components, the GM-PHDF starts the prediction step using components from previous time step as

$$w^i_{k|k-1} = p_s w^i_{k-1}, \quad \mathbf{m}^i_{k|k-1} = \mathbf{F}\mathbf{m}^i_{k-1}, \quad \mathbf{P}^i_{k|k-1} = \mathbf{Q} + \mathbf{F}\mathbf{P}^i_{k-1}\mathbf{F}^T \tag{6.8}$$

where p_s indicates probability of survival of the targets and $i = 1, ..., J_{k-1}$. Calculations of values for update step of the GM-PHDF are done with the following equations:

$$\boldsymbol{\eta}^j_{k|k-1} = \mathbf{H}\mathbf{m}^j_{k|k-1} \tag{6.9}$$

$$\mathbf{S}^j_k = \mathbf{R} + \mathbf{H}\mathbf{P}^j_{k|k-1}\mathbf{H}^T \tag{6.10}$$

$$\mathbf{K}^j_k = \mathbf{P}^j_{k|k-1}\mathbf{H}^T[\mathbf{S}^j_k]^{-1} \tag{6.11}$$

$$\mathbf{P}^j_{k|k} = [\mathbf{I} - \mathbf{K}^j_k\mathbf{H}]\mathbf{P}^j_{k|k-1}. \tag{6.12}$$

Update step of the GM-PHDF when no measurements are taken are:

$$w_k^j = (1 - p_d) w_{k|k-1}^j \qquad j = 1, ..., J_{k|k-1} \tag{6.13}$$

$$\mathbf{m}_k^j = \mathbf{m}_{k|k-1}^j \qquad j = 1, ..., J_{k|k-1} \tag{6.14}$$

$$\mathbf{P}_k^j = \mathbf{P}_{k|k-1}^j \qquad j = 1, ..., J_{k|k-1} \tag{6.15}$$

where p_d represents the probability of detection. In the existence of a measurement set z, the update step is performed with the additional steps below

$$w_k^{l*J_{k|k-1}+j} = p_d w_{k|k-1}^j \mathcal{N}(z; \boldsymbol{\eta}_{k|k-1}^j, \mathbf{S}_k^j) \qquad j = 1, ..., J_{k|k-1} \tag{6.16}$$

$$\mathbf{m}_k^{l*J_{k|k-1}+j} = \mathbf{m}_{k|k-1}^j + \mathbf{K}_k^j (z - \boldsymbol{\eta}_{k|k-1}^j) \qquad j = 1, ..., J_{k|k-1} \tag{6.17}$$

$$\mathbf{P}_k^{l*J_{k|k-1}+j} = \mathbf{P}_{k|k}^j \qquad j = 1, ..., J_{k|k-1} \tag{6.18}$$

where $\mathcal{N}$ indicates a normal probability distribution. Finally weights are normalized using the following equation

$$w_k^{l*J_{k|k-1}+j} = \frac{w_k^{l*J_{k|k-1}+j}}{\kappa_k + \sum_{i=1}^{J_{k|k-1}+i} w_k^{l*J_{k|k-1}+i}} \tag{6.19}$$

where κ_k, shows false measurement rate at time k.

6.1.1. Estimation of Number of Targets

A thresholding operation is proposed [21] for extracting state estimations for multiple targets, as in algorithm below.

Algorithm 1 Extracting state estimations for multiple targets

Input: $\{w_k^i, m_k^i, P_k^i\},\ i = 1, ...J_k$
Output: $\hat{m_k}$
  $\hat{m_k} \leftarrow \{\}$
  for $i = 1, ...J_k$ **do**
    if $w_k^i \geq 0.5$ **then**
      for $j = 1, ...\text{round}(w_k^i)$ **do**
        $\hat{m_k} \leftarrow \{\hat{m_k}, m_k^i\}$
      end for
    end if
  end for

Figure 6.1. Algorithm for estimating the number of targets.

This method selects the means(m) of dominant Gaussian components in the sense of having greater weight(w) than some predetermined threshold(κ) so an improper threshold selection causes serious performance degradation. In the presence of high clutter rate and missed detections determining a threshold value gets trickier. Even a threshold is selected successfully, this threshold can be problem dependent or can be useful only on some initial weights of the Gaussian components. This means that a method which is initial weight independent and generalizable can help to extract best possible the GM-PHDF estimation. So we have tried to try alternative methods for state extraction.

In this study, we will examine three methods for GM-PHDF estimation such as Elbow, Average Silhouette and Gap Statistic which have the following properties:

- Use a weight threshold as stated above.
- Find centers of gravity of the sorted Gaussian weights at the current time step and cluster them into two clusters using this center point then selecting the cluster consisting the heaviest weight.
- Combine two methods by selecting the cluster consisting of the heaviest weight like above after using a weight threshold.
- Use DB-SCAN algorithm for clustering weights and then select the cluster that has the highest weight sum.

- Use several K-Means clustering methods given below on set of Gaussian components weights.

6.1.1.1. Elbow Method.

- Computes k-means clustering for different values of k.
- For each k, calculates the total within-cluster sum of square (WSS),
- Plots the curve of wss according to the number of clusters k.
- Selects the location of a bend (knee) as number of clusters.

6.1.1.2. Average Silhouette Method.

- Computes k-means clustering for different values of k.
- For each k, calculates Silhouette Coefficient (-1, 1),
  Silhouette Coefficient = $\frac{(b-a)}{max(a,b)}$
  a : the mean intra-cluster distance
  b : the mean nearest-cluster distance
- Plots the curve of Silhouette Coefficients according to the number of clusters k.
- Selects the location of a peak as number of clusters.

6.1.1.3. Gap Statistic Method.

- Computes k-means clustering for different values of k.
- For each k, calculates intra-cluster variance W_k.
- Genarates B data sets with random uniform distribution in range $[min(data), max(data)]$.
- Computes k-means clustering for each data set with different values of k.
- For each k, calculates intra-cluster variance $W_k b$.
- Computes the estimated Gap Statistic finally.

$$Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}) - log(W_k). \tag{6.20}$$

7. EXPERIMENTS AND RESULTS

7.1. Performance Evaluation Metrics

7.1.1. Root Mean Square Error

Various metrics are used to measure the performance of the methods used in the target tracking system. The most preferred of these metrics is the Root Mean Square Error (RMSE). RMSE for position or velocity can be formulated as

$$RMSE(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \sqrt{[\,(\mathbf{x} - \hat{\mathbf{x}})^2 + (\mathbf{y} - \hat{\mathbf{y}})^2]} \tag{7.1}$$

$$RMSE(\hat{\dot{\mathbf{x}}}, \hat{\dot{\mathbf{y}}}) = [\,(\dot{\mathbf{x}} - \hat{\dot{\mathbf{x}}})^2 + (\dot{\mathbf{y}} - \hat{\dot{\mathbf{y}}})^2] \tag{7.2}$$

where $\mathbf{x}, \mathbf{y}, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \dot{\mathbf{x}}, \dot{\mathbf{y}}, \hat{\dot{x}}, \hat{\dot{y}}$ are the vectors representing the position in the x-axis, the position in the y-axis, the position in the x-axis, the position in the y-axis, the velocity in the x-axis, the velocity in the y-axis, and the estimation of the velocities in these axes, respectively.

7.1.2. Object Tracking Error

Object Tracking Error (OTE) has been defined as another measurement metric [24]. The recommended metric for target tracking mostly from video is (7.3) for position and (7.4) for speed in an estimate of N steps. Apart from this metric, which is also equivalent to the mean of the RMSE, the standard deviation (OTDstd), minimum, maximum values and medians of the RMSE are also defined as the performance criteria metric.

$$OTE(\hat{x}, \hat{y}) = \frac{1}{N} \sum_{n=1}^{N} \sqrt{[\,(x_n - \hat{x}_n)^2 + (y_n - \hat{y}_n)^2]} \tag{7.3}$$

$$OTE(\hat{x},\hat{y}) = \frac{1}{N}\sum_{n=1}^{N}\sqrt{[\,(\dot{x}_n - \hat{\dot{x}}_n)^2 + (\dot{y}_n - \hat{\dot{y}}_n)^2]}. \tag{7.4}$$

7.1.3. Percentage Fit Error

Percentage Fit Error (PFE) is another measurement tool used to benchmark the performance of filters [25]. In this measurement tool, performance is obtained by using the error rate rather than the amount of error. It is expressed mathematically by the following formulas:

$$PFE_x = \frac{1}{N}\sum_{n=1}^{N} 100\frac{\|(x_n - \hat{x}_n)\|}{\|(x)_n\|}\% \tag{7.5}$$

$$PFE_y = \frac{1}{N}\sum_{n=1}^{N} 100\frac{\|(y_n - \hat{y}_n)\|}{\|(y)_n\|}\% \tag{7.6}$$

$$PFE = \sqrt{PFE_x^2 + PFE_y^2}. \tag{7.7}$$

7.1.4. The Optimal Subpattern Assignment (OSPA)

The Optimal Subpattern Assignment (OSPA) [26] is a commonly used performance evaluation metric for multi-target tracking algorithms. It can compare two sets which have different number of elements, thus; it can be implemented for comparison of the estimation sets which have false and missed detections with ground truths set.

In multi-target tracking, we can define two vector sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ for estimations $\boldsymbol{x}$ and ground truths vectors $\boldsymbol{y}$, respectively as below.

$$\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_m}\}, \boldsymbol{Y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_n}\}. \tag{7.8}$$

The cut-off distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ can be defined as

$$d_c(\boldsymbol{x}, \boldsymbol{y}) = \min\{c, d(\boldsymbol{x}, \boldsymbol{y})\} \tag{7.9}$$

where $c > 0$ is a cut-off parameter. The p'th order OSPA metric with a cut-off parameter c is denoted as $D_{p,c}$ which can be expressed as the following formula:

$$D_{p,c} = e_{p,c}^{loc} + e_{p,c}^{card} \tag{7.10}$$

where $e_{p,c}^{loc}$ is a location error whereas $e_{p,c}^{card}$ is a cardinality error. These two error statements are derived as follows,

$$e_{p,c}^{loc}(\boldsymbol{X}, \boldsymbol{Y}) = \left[\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}(d_c(\boldsymbol{x_i}, \boldsymbol{y}_{\pi_i}))^p\right)\right]^{1/p} \tag{7.11}$$

where Π_n is the set permutations of lenght m choosen out of n elements such as

$$e_{p,c}^{card}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{(n-m)c^p}{n}\right)^{1/p}. \tag{7.12}$$

7.2. Simulations

7.2.1. Kalman Filter Simulations

The tracking application was performed with five virtual targets whose initial and final positions were determined randomly. We tracked them after detecting their first positions and then estimated their probable positions after each interval. We repeated these steps until five targets reached their final positions. Following each iteration, we plotted our estimation and the real position of the targets as well as the error rate of our estimation. The variances were selected as 1.44 when adding the white noise for both determining the actual routes from the intended routes, and the observations

from those actual routes. Randomly selected initial and random positions, intended route, actual target positions and KF estimations for one target can be seen in Figure 7.1.

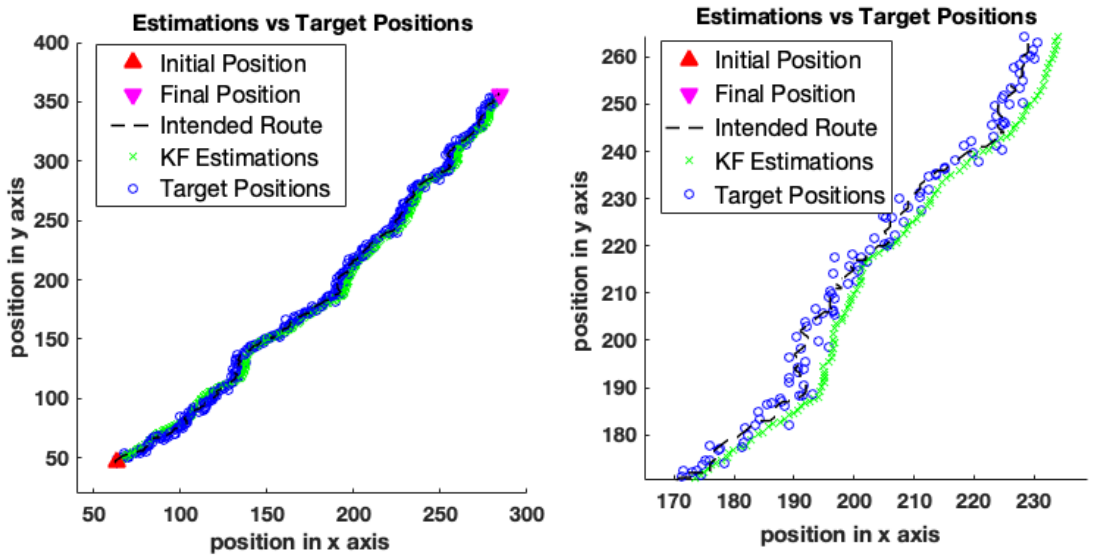

Figure 7.1. Estimation vs Target Positions.

As we stated above, the error rate of KF Estimations decreased gradually when there was no sudden change in route of the target. Figure 7.2 shows the error weighted target positions. It can be seen clearly that error rates were peaked when the target altered its route.

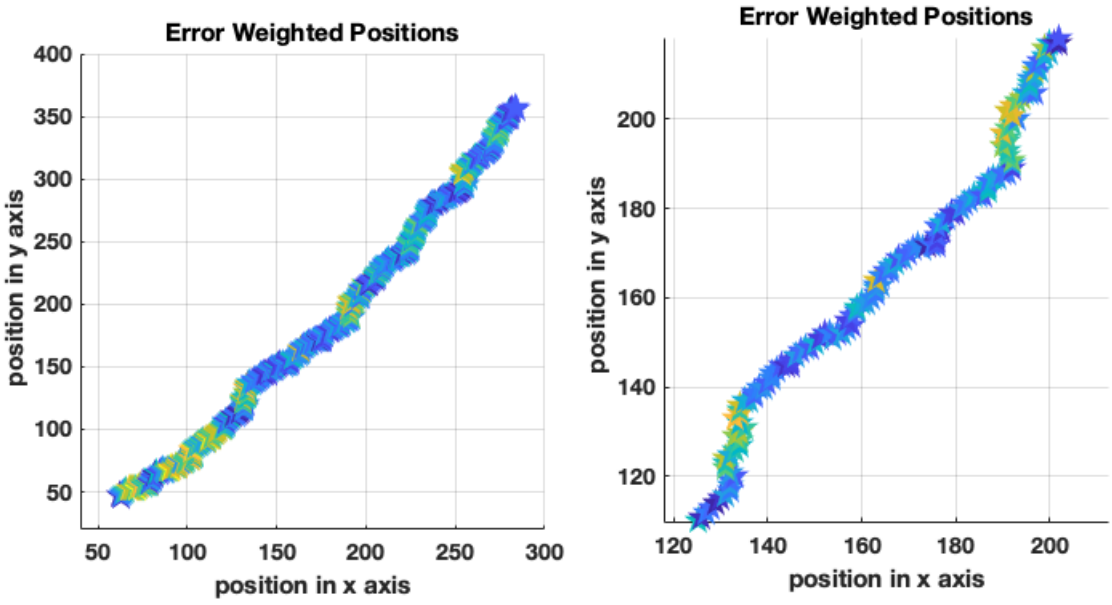

Figure 7.2. Error Weighted Positions (Darker colours represent lower error rates).

For visualizing multiple target tracking, we plotted all targets together with their true paths and the corresponding estimated paths in Figure 7.3.

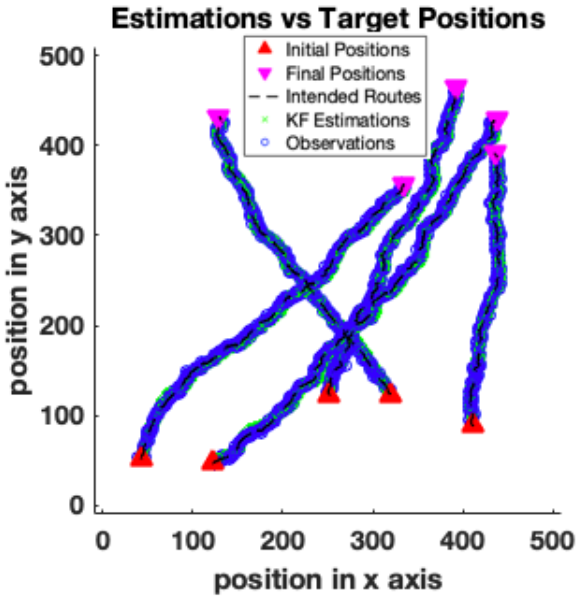

Figure 7.3. Final Plot of Initial and Final Positions, Intended Routes, Observations and Estimated Paths of KF for five targets.

In addition, we obtained the mean of the MSE (Mean Squared Error) for five experiments of tracking targets from initial positions to final positions, as 3.72 for the position in x-axis and as 4.64 in y-axis.

7.2.2. Performance Comparison of KF, UKF and EKF

In order to compare the tracking performances KF and UKF, the example tracking scenario in Figure 7.4 was generated using MATLAB. In this scenario, it is assumed that the target is moving on a linear path and it is desired to show visually how successful KF and UKF are in following the target. While UKF, which can be seen in Figure 7.4, followed the target with a better performance from the very first moment, it was determined that KF missed the target a little at first, but achieved better performance towards the end.

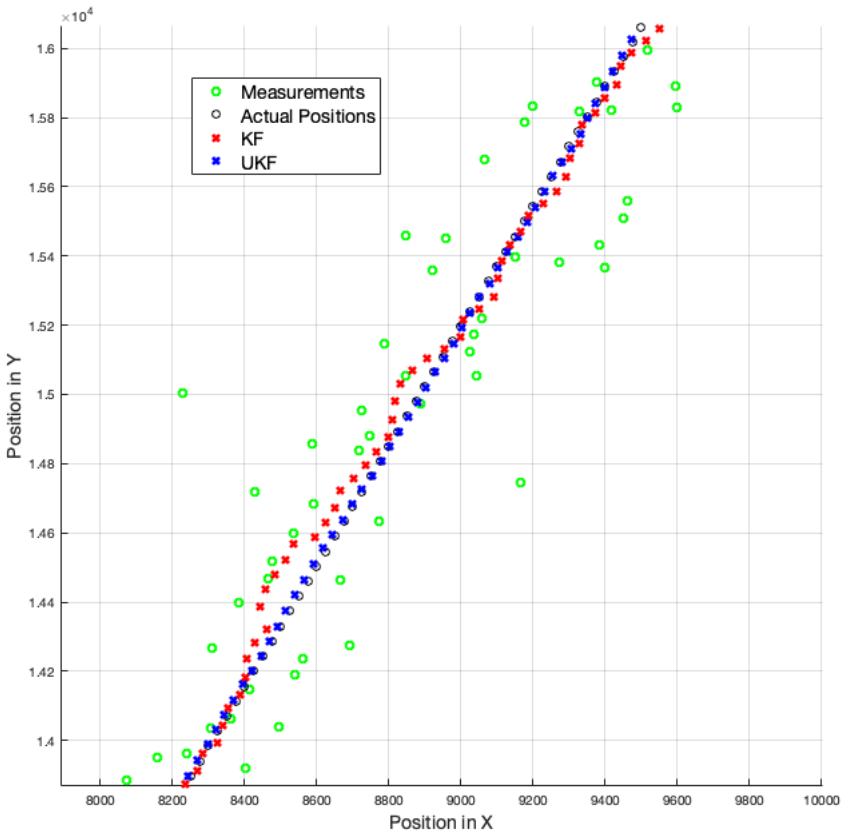

Figure 7.4. Comparison of KF and UKF.

Again in the same tracking scenario, in addition to the KF and UKF performances, the performance graph of the EKF, UCMKF and DUCMKF filters is shown in Figure 7.5 in terms of MSE metric. As can be seen here, it is clearly seen that UKF performs much better than others, and KF is the filter type with the lowest performance among the five filters. While EKF has average performance, it can be seen that DUCMKF and UCMKF performed close to UKF at first, but failed to follow the target as the target moves.

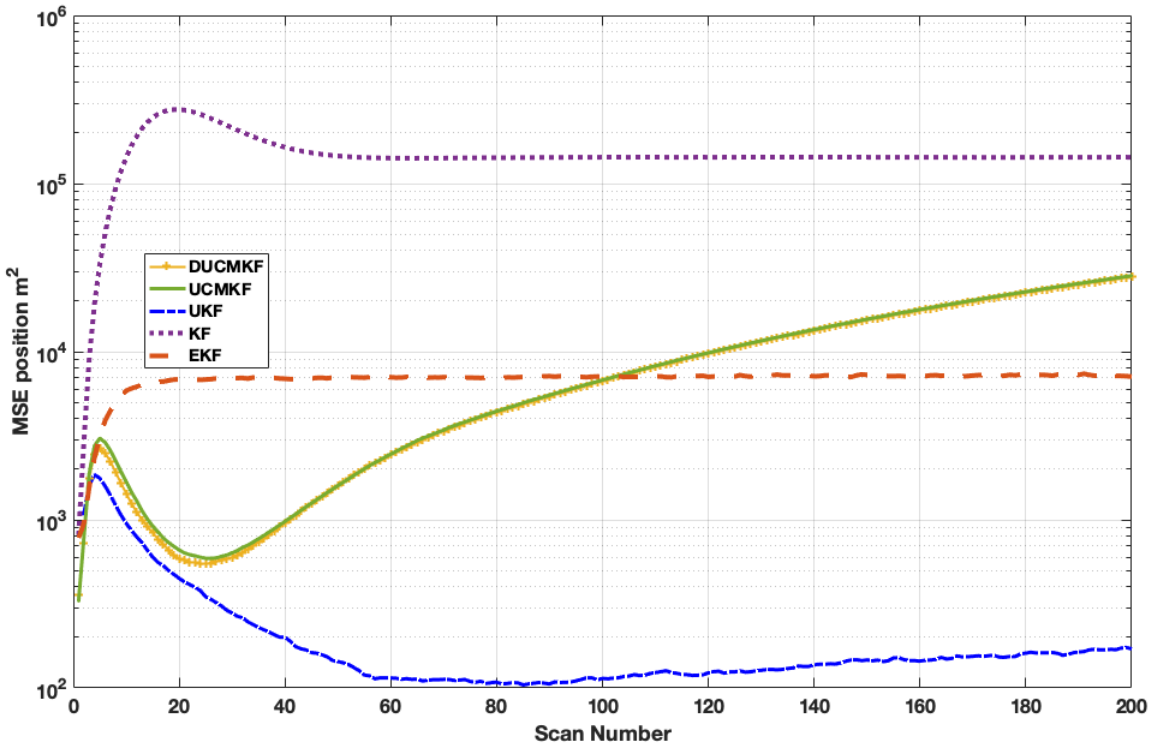

Figure 7.5. MSE Comparison of Different Kalman Variants.

7.2.3. IMM Simulations

In order to compare the IMM model with Kalman filters, the motion model shown in Figure 7.6 was created.

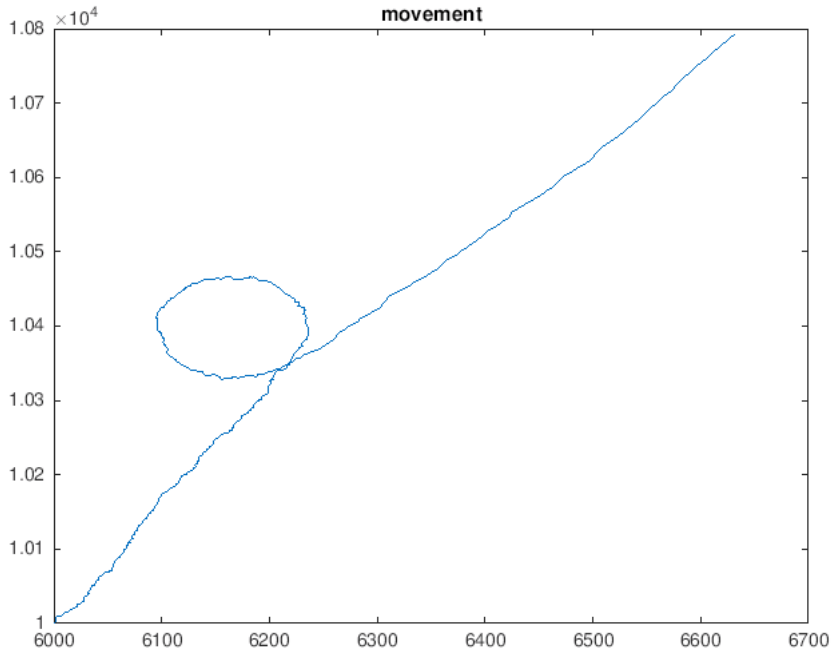

Figure 7.6. Motion model used in IMM simulations.

In this model, the vessel moves with a constant speed of the first 200 steps, a constant rotational motion in the next 100 steps, and a constant acceleration motion in the next 100 steps. Only when we adapted the UKF model we got the model probabilities shown in Figure 7.7.

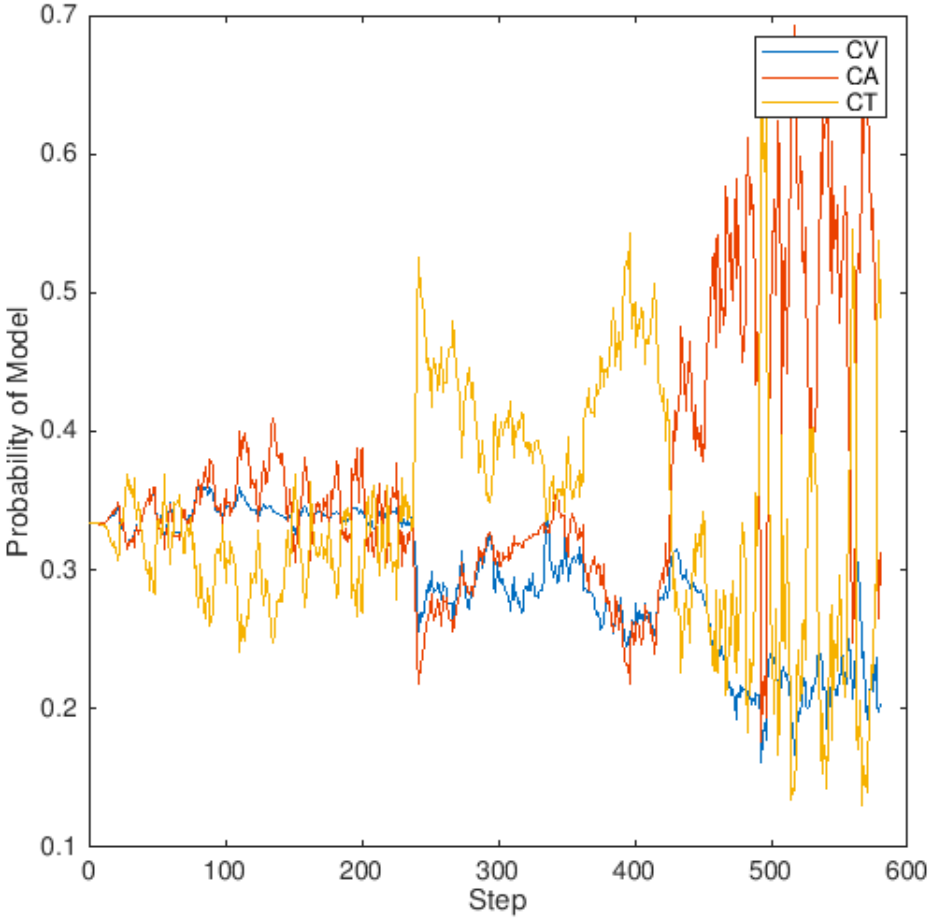

Figure 7.7. Probability distribution of different motion models in the IMM model.

As a result of the simulations, it was observed that these possibilities were compatible with the established movement model. Then we simulated the IMM with CMKF filters and compared it with the UKF-IMM as shown in Figure 7.8.

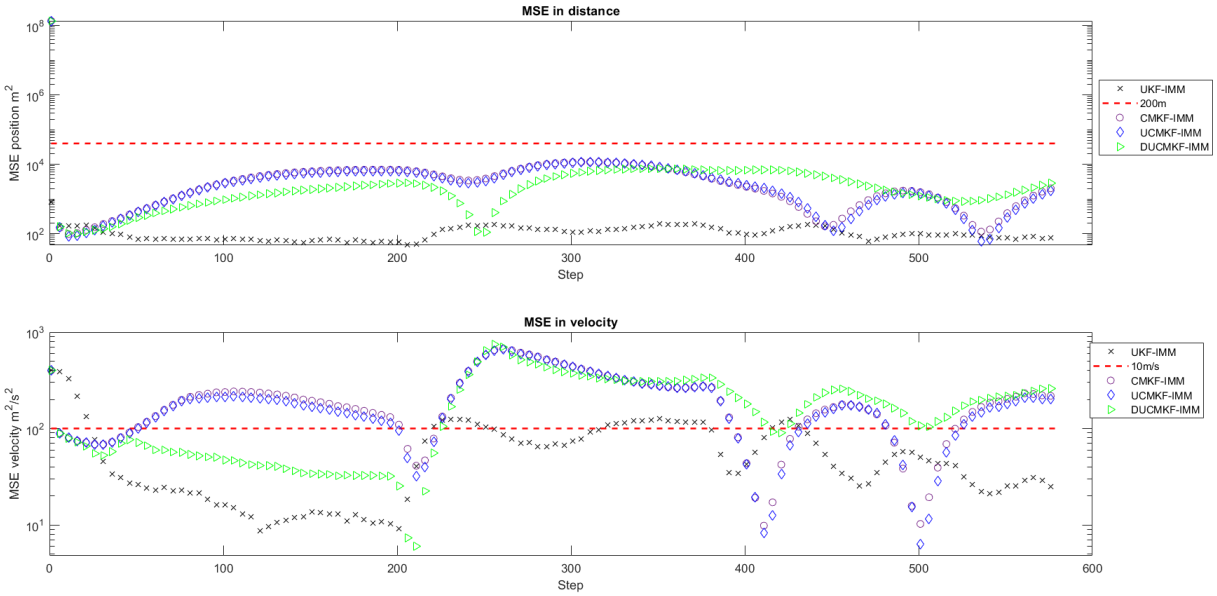

Figure 7.8. MSE comparison of different IMM filters.

As can be seen in Figure 7.8, UKF-IMM has the best performance in terms of MSE, while CMKF-IMM and UCMKF-IMM performed close to each other. DUCKMF, on the other hand, performed better than CMKF-IMM and UCMKF-IMM, while it was below UKF-IMM in terms of tracking performance.

7.2.4. CUSUM Simulations

Using the CUSUM model with CMKF filters, we obtained the log-similarity function seen in figure 7.9. In this section, the tracking scenario in the previous section was taken as a reference, and the tracking performances and log-likelihood ratio of CUSUM were created in line with this model. Then, the tracking results of CUSUM-CMKF/EKF and IEKF results are given in figures 7.10, 7.11 and 7.12.

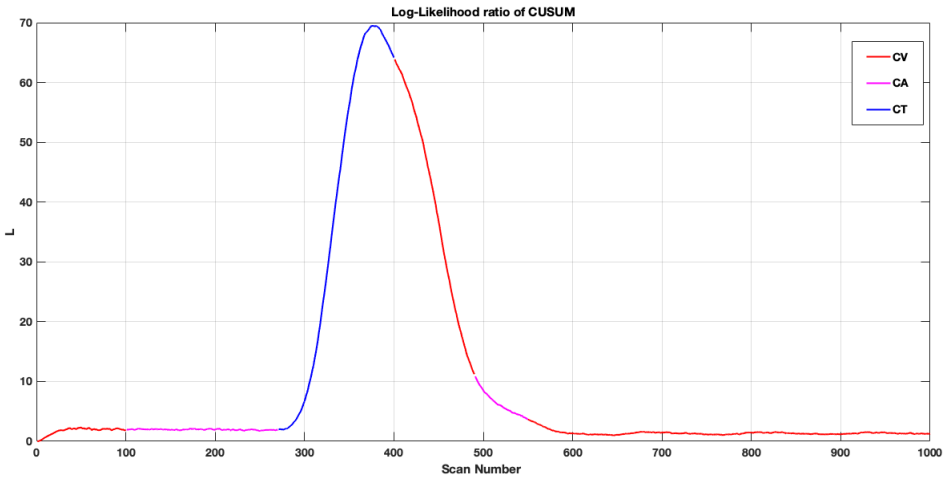

Figure 7.9. Determining the motion model with the log-likelihood function.

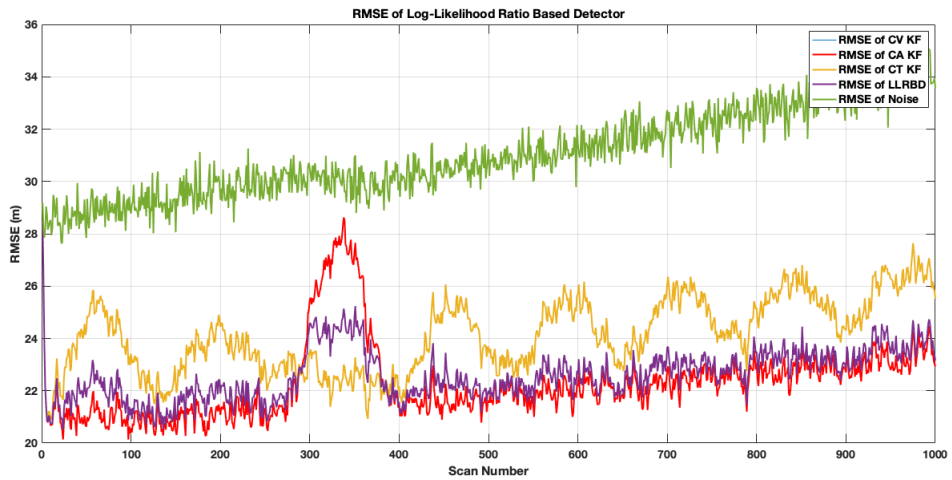

Figure 7.10. Simulation of CUSUM and Converted Measurement Kalman Filter together.

When we examine the simulations in which three different filters are used together with CUSUM, it is understood that IEKF has the lowest RMSE value. It is seen that the CMKF and CUSUM models are far behind the other filters in terms of tracking performance. In the CUSUM-CMKF model, it is seen that CA-KF performs better when the target moves at constant speed and constant acceleration, while the margin of error suddenly increases when the target makes a constant turn. While we see that CUSUM performs better than CT-KF in general, CT-KF seems to perform better when the target makes a constant turn.

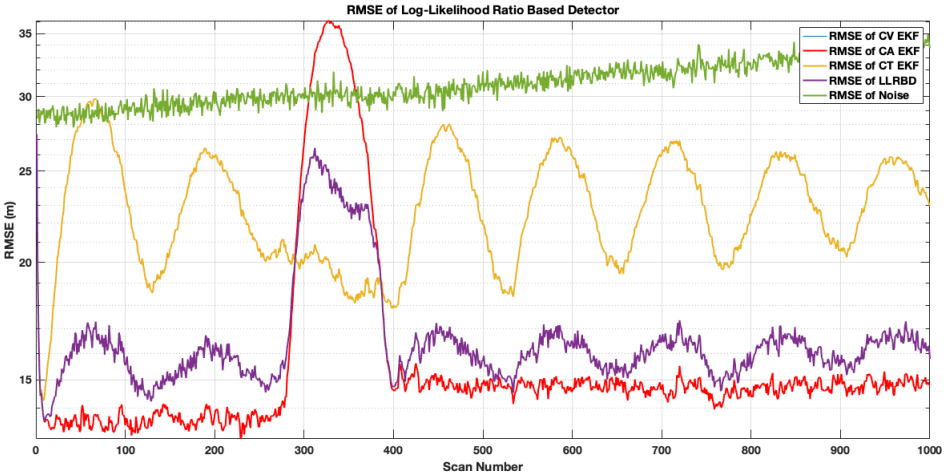

Figure 7.11. Simulation of CUSUM and Extended Kalman Filter together.

When we look at the EKF-CUSUM simulations, it is seen that CA-EKF has the best performance among all filters, except when the target makes a constant turn. Although CUSUM outperforms CT-EKF in general, the margin of error is greater than CT-EKF when the target turns.

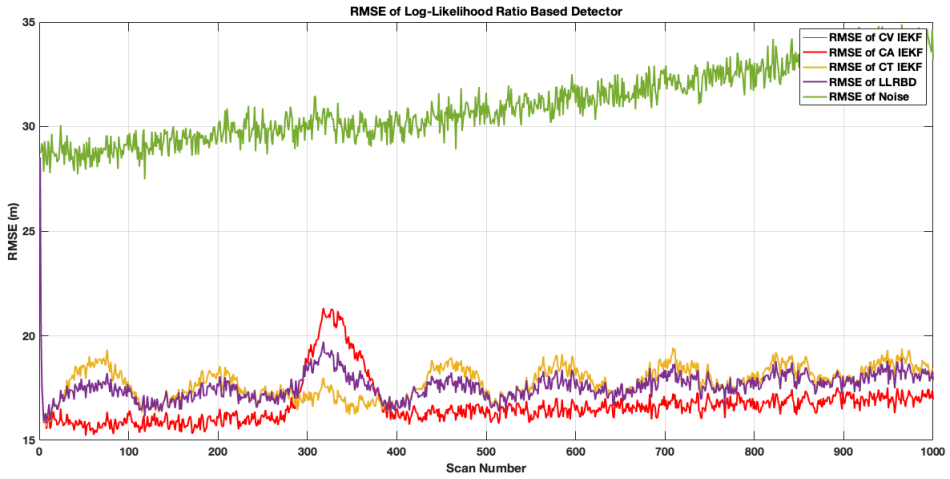

Figure 7.12. Simulation of CUSUM and Iterated Extended Kalman Filter together.

When we examine the IEKF-CUSUM simulations, it is seen that all filters are close to each other in terms of performance. Although the comparison of the filters with each other is similar to the simulations in EKF, it is seen that there is an improvement compared to EKF in terms of tracking performance.

7.2.5. CUSUM/IMM Simulations with Range and Bearing Measurements

In order to observe the performance of the filters in a scenario where the target has multiple motion models, tracking simulations were made with various scenarios in computer environment. In these simulations, while the radar was accepted at the origin point, the measured target which is initially at the point (6000 km, 10000 km) started to move in $\pi/6$ radian direction with an initial velocity of 20 m/s. It has an acceleration of 1 m/s^2 in addition to the process noise and it is moving with a constant acceleration. In filters, Cartesian conversion of the first measurement value of the target is used as the starting position for each target, and zero is used for the velocity value. The erroneous start, although it gives a negative result for the Kalman Filter, has been chosen because it offers an applicable reality. The diag(104,104,50,50,1,1) values were used as the initial value of the covariance matrix. In the IMM model, the probability of all models was initially taken equal and the probability of transition between models was determined as 0.05. The threshold values checked for changing the model in CUSUM were also determined as a result of the observation. Parameters in IMM are used as $\alpha = 10^{-3}$ and $\beta = 2$. In all simulations, the average of 100 Monte Carlo simulations was taken.

Various metrics are used to compare the performance of the filters used. One of these metrics is the Root Mean Square Error (RMSE). Percentage Fit Error (PFE) is another measurement tool used to compare the performance of filters. In this measurement tool, performance is achieved by using the error rate rather than the amount of error. Object Tracking Error (OTE) has been defined as another measurement metric. The recommended metric for target tracking from literature corresponds to the average of RMSEs for position and velocity in an estimate consisting of N steps. Again, the standard deviation (OTDstd), minimum, maximum values and medians of RMSE are defined as performance criteria metric.

In the first simulation, the target moves the first 200 steps with constant velocity, the next 200 steps with coordinated rotational motion, and the last 100 steps with

constant acceleration. Our measurement period is 1 measurement per second. Process noise was also taken as having standard deviation in both axes. The motion model was created, the measurements taken are shown in Figure 7.13, and the RMSE values of the errors are shown in Figure 7.14 and Figure 7.15. When the results are examined, while the filter using CUSUM has lower performance in CV and CA motion models, it has higher success in CT motion model.

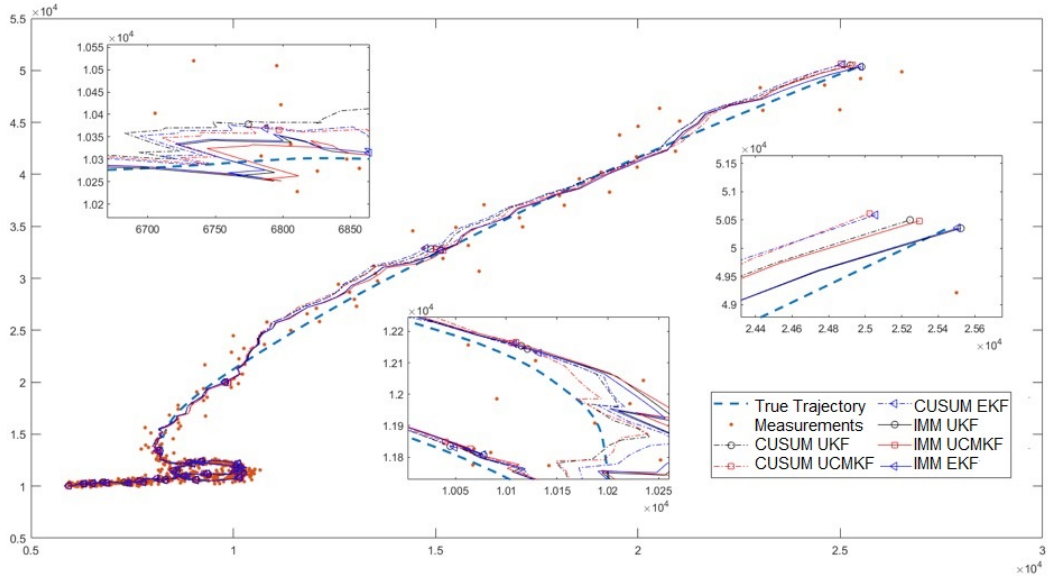

Figure 7.13. The actual location of the target and filter estimates in the multiple motion model.

When a simulation consisting of only the CT motion model is made, it was observed that the RMSE values of the filters using CUSUM are lower than those using IMM. However, when the measurement errors are increased and the values of $\sigma_r = 100$ and $\sigma_\theta = 2$ are used, it is observed in Figure 7.17 that the models using CUSUM approach the IMM more slowly. Comparison of all filters according to OTE OTEstd and PFE metrics in the aforementioned scenarios is shown in Table 1 and Table 2.

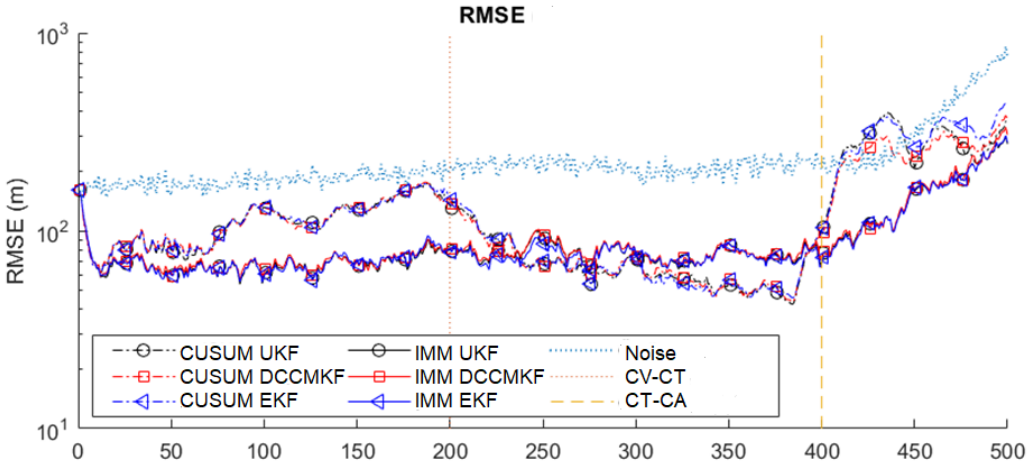

Figure 7.14. RMSE (Position) outputs of filters in CT motion model ($\sigma_r = 20$ and $\sigma_\theta = 1$).

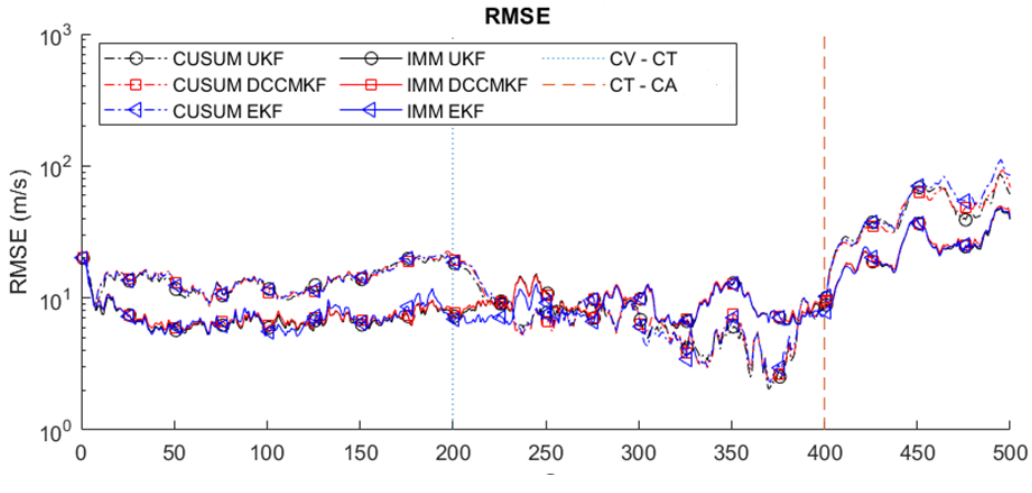

Figure 7.15. RMSE (Velocity) outputs of filters in CT motion model ($\sigma_r = 20$ and $\sigma_\theta = 1$).

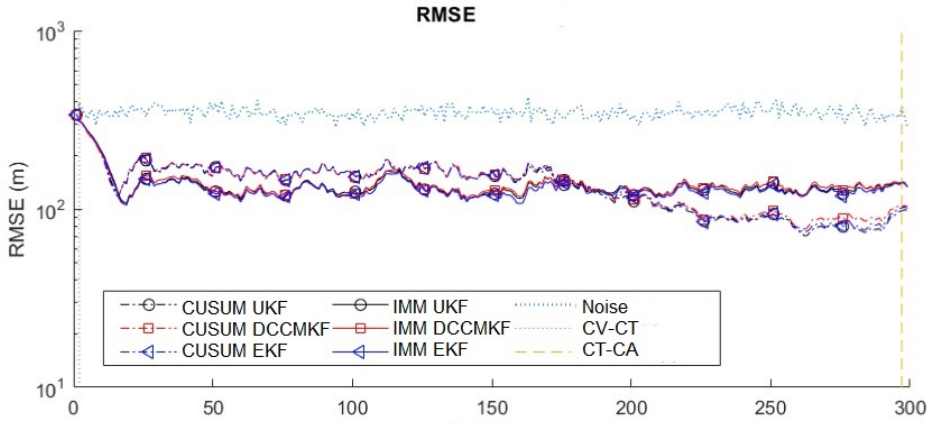

Figure 7.16. RMSE (Position) outputs of filters in CT motion model ($\sigma_r = 100$ and $\sigma_\theta = 2$).

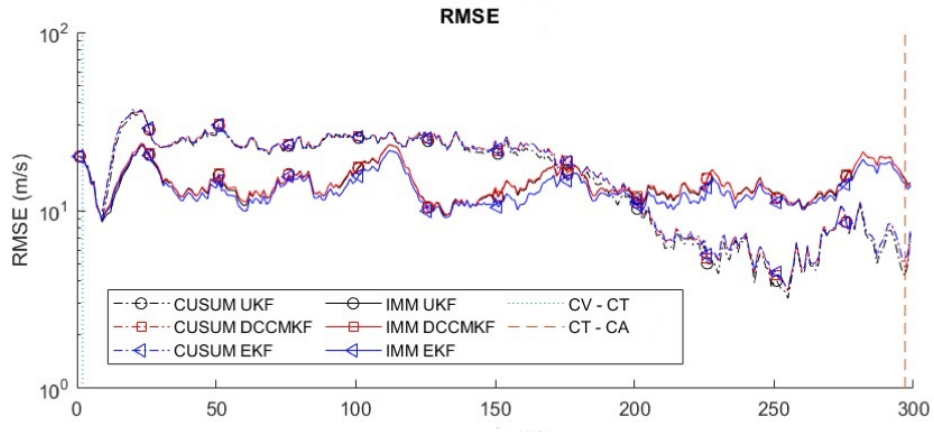

Figure 7.17. RMSE (Velocity) outputs of filters in CT motion model ($\sigma_r = 100$ and $\sigma_\theta = 2$).

Table 7.1. Comparison of performance of different filters in multiple motion model.

	σ_r=20 σ_θ=1			σ_r=100 σ_θ=2		
	PFE	OTE	OTEstd	PFE	OTE	OTEstd
CUSUM-UKF	1.98%	128.62	86.58	3.89%	249.26	255.14
CUSUM-DCCMKF	1.86%	124.33	77.12	4.03%	256.77	269.26
CUSUM-EKF	2.12%	134.13	96.48	4.29%	265.95	298.57
IMM-UKF	1.26%	89.49	11.49	2.30%	170.18	**97.02**
IMM-DCCMKF	1.30%	91.44	11.79	2.42%	174.32	111.39
IMM-EKF	**1.26**%	**88.95**	**11.40**	**2.28**	**166.84**	100.83
Noise	3.31%	233.20	110.53	6.22%	469.71	216.22

Table 7.2. Comparison of the performance of different filters in the CT model.

	σ_r=20 σ_θ=1deg			σ_r=100 σ_θ=2deg		
	PFE	OTE	OTEstd	PFE	OTE	OTEstd
CUSUM-UKF	**1.17**%	**56.66**	19.74	2.99%	138.55	43.92
CUSUM-DCCMKF	1.18%	57.49	19.47	3.02%	141.19	41.77
CUSUM-EKF	1.17%	56.95	19.75	3.01%	139.50	43.13
IMM-UKF	1.30%	65.03	10.58	2.85%	136.36	28.51
IMM-DCCMKF	1.31%	65.80	**10.43**	2.87%	137.54	**27.80**
IMM-EKF	1.28%	64.28	10.71	**2.78**%	**132.97**	29.02
Noise	3.38%	167.67	13.68	7.29%	348.42	24.51

According to the simulation results, if UKF, EKF and DC-CMKF filters are used with IMM and CUSUM models, it is possible to make successful estimations in Cartesian coordinates by taking measurements from polar coordinates. Although there is not much difference in performance between UKF, EKF and DC-CMKF filters, it has been observed that DC-CKMF filter works faster in the installed system. When CUSUM and IMM models created for multiple models are compared with each other, it has been noted that CUSUM cannot achieve the same performance as IMM. Again, when the amount of error was increased, the performance difference between IMM and CUSUM also increased. CUSUM fails in many scenarios to accurately predict the

new model, especially after transitions between models. In our study, it was assumed that the rotation angle in the coordinated rotation, the amount of process noise and the amount of measurement noise were known correctly, and a method could not be developed for their estimation. However, in order to be more realistic, the estimation of these parameters may be necessary for radar applications.

7.2.6. The GM-PHDF Experiments for Single Maneuvering Target Tracking

In this simulation, we generated an example scenario below where ground truths are plotted in blue, measurements are plotted in green and sensor positions are plotted in black markers.

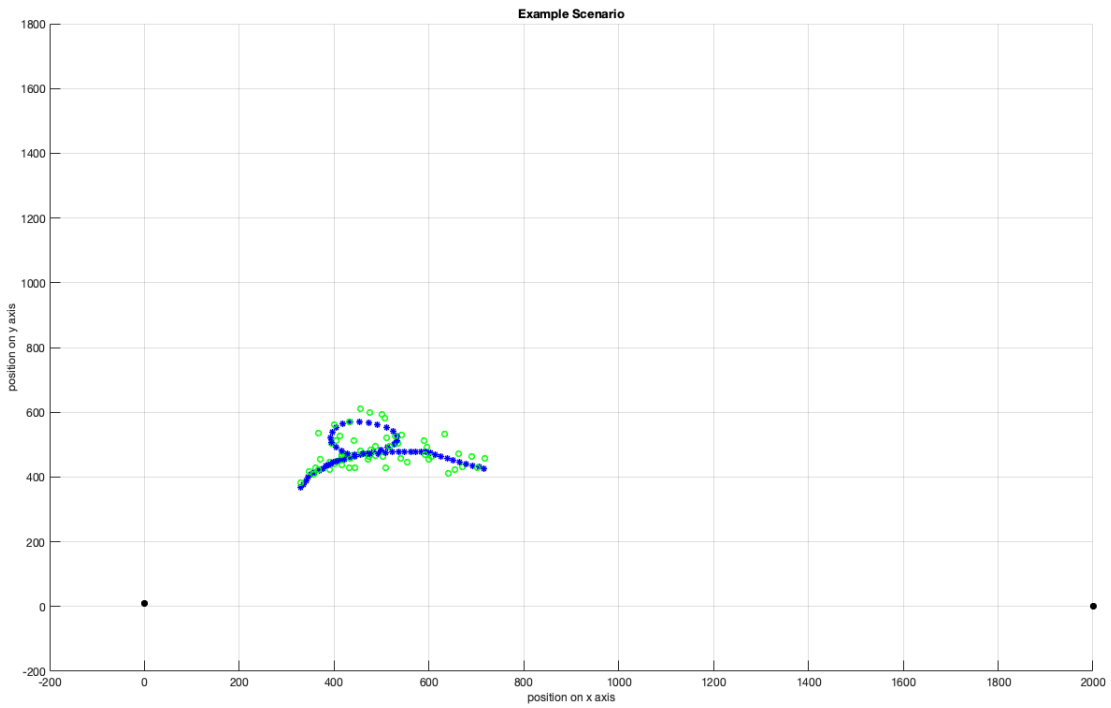

Figure 7.18. The trajectory of the target.

The EKF and the UKF algorithms and their implementation with the GM-PHDF in both constant velocity and constant turn models are implemented for this experiment. OSPA metric results are given 7.19 below. Generally the GM-PHD implemen-

tations of the filters showed better performance in both models in comparison with the common filtering algorithms.

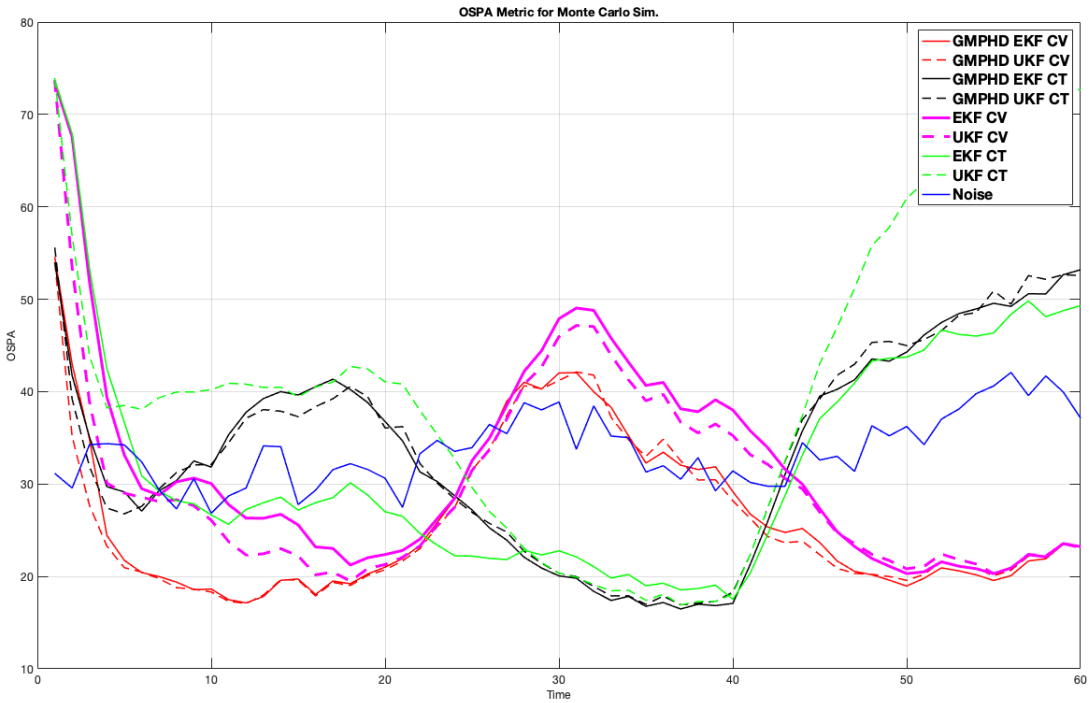

Figure 7.19. OSPA metric results of GM-PHD, EKF and UKF with two different motion models.

7.2.7. The GM-PHDF Experiments for Estimation of Number of Targets

For estimating the number of targets, the following methods were used in this experiments.

- Using a weight threshold, (indicated as WT at the figures below)
- Using centers of gravity method (indicated as COG at the figures below)
- Combination of two methods (indicated as WT&COG at the figures below)
- Using DB-SCAN algorithm (indicated as DBSCAN at the figures below, DB-SCAN parameters: eps = 0.15, MinPts = 1)
- Using several K-Means clustering methods as individually and as combined with other methods (indicated with their names at the figures below)

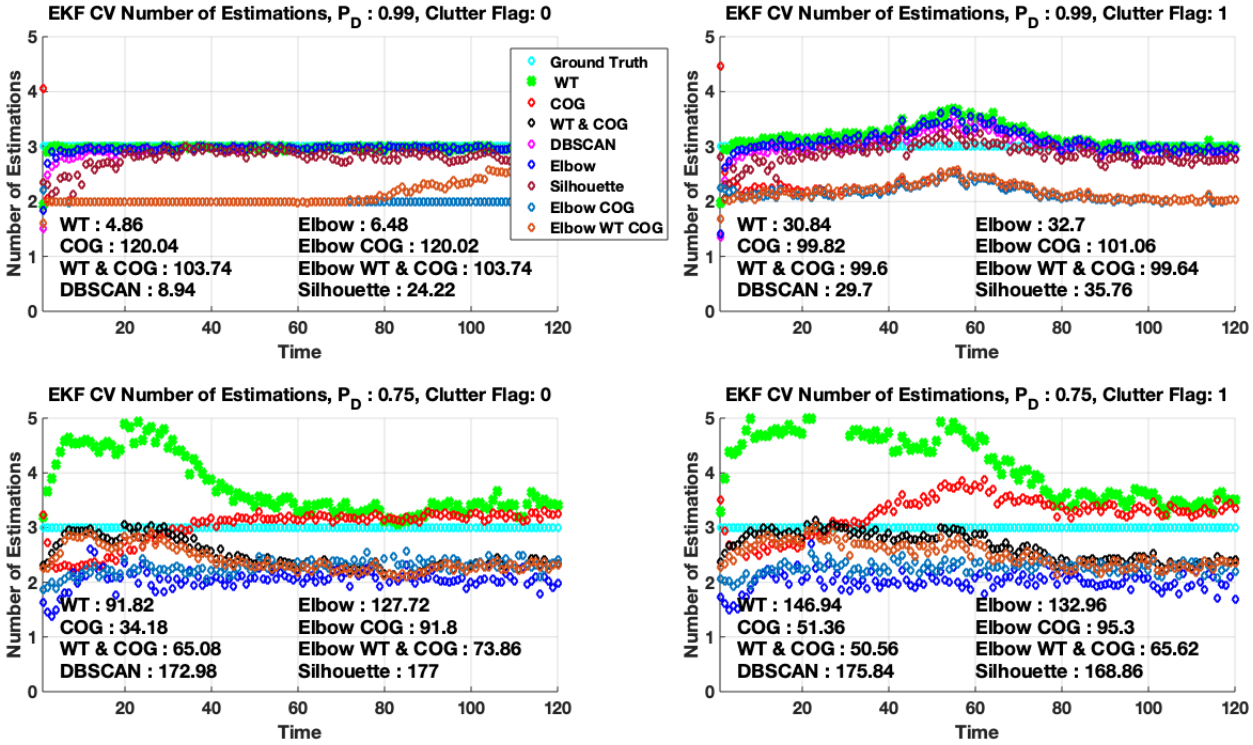

Figure 7.20. Number of Estimation Errors for 3 Targets.

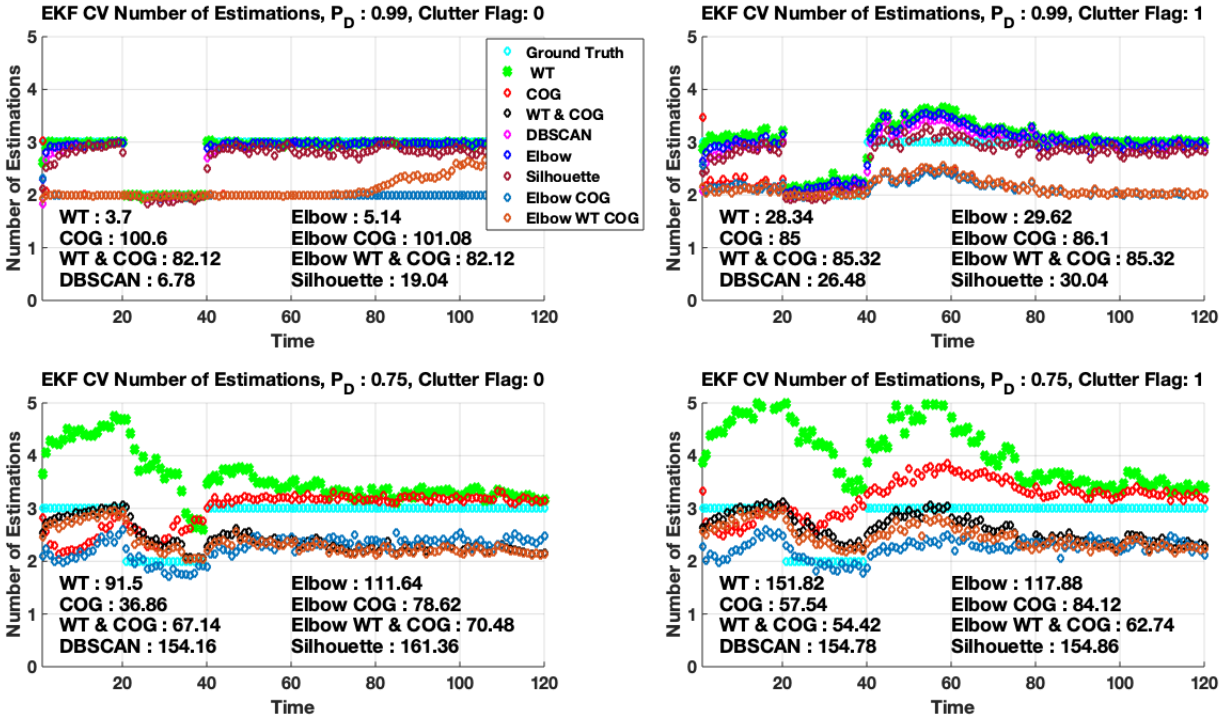

Figure 7.21. Number of Estimation Errors When There is a Change of Number of Targets in Time (3 targets between time steps(0-20), 2 targets between time steps(20-40), 3 targets between time steps(40-120)).

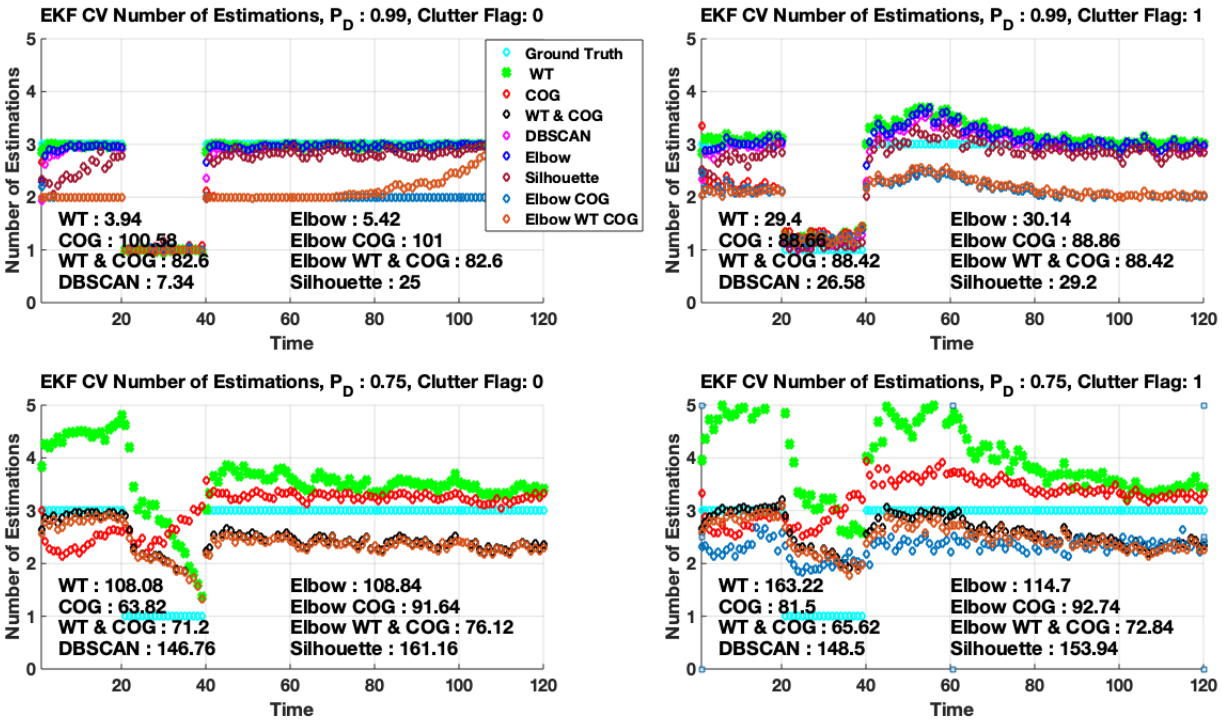

Figure 7.22. Number of Estimation Errors When There is a Change of Number of Targets in Time (3 targets between time steps(0-20), 1 target between time steps(20-40), 3 targets between time steps(40-120)).

It is clear from the experiments that when detection probability is low and the measurement set contains false measurements, WT and DBSCAN give similar poor results in comparison with the other methods (COG and Est WT-COG). This may stem from that measurement likelihood values which are multiplication factors of the Gaussian weights, can be extremely high or low due to the so many clusters in measurement set. So after normalization, some weights can vanish and others can rise seriously. In this case, it is hard to determine the correct threshold value for Est WT. Also DBSCAN algorithm fails to select correct weights using constant eps value owing to so much separation of the weight values. We can also say that performances of all methods show little changes in their OSPA scores against clutter existence in measurements. But their actual performances degrade due to the noise score reduction.

8. CONCLUSION & FUTURE WORK

In this thesis, we examined the tracking performances of various Kalman filters proposed in the literature for the target tracking problem, both in the case where the target has a single motion model and in the scenario where the target has more than one motion model. While evaluating the tracking performance of the filters, we took the low percentage of missing traces as a reference. It has been noticed that the Kalman Filter is an optimal filter for target tracking when the target motion model is linear, while the use of EKF and UKF is more appropriate in nonlinear systems. On the other hand, in the scenario where there is more than one motion model, it has been seen that the IMM method, in which more than one filter motion model is run in parallel, is the most suitable solution for targets with different motion models. The most common multi-target tracking algorithm in radar surveillance systems today uses a maneuver detector with a standard Kalman filter. In our thesis, we touched on the working principle of the CUSUM algorithm and presented the simulation results for the selection of the correct motion model with the CUSUM detector in the scenario where the target has different motion patterns. It is envisaged that it can be implemented as a complete multi-pattern target tracking algorithm that tracks a target with multiple motion patterns at the same time.

As a continuation of this work, our future work is primarily to run target tracking algorithms with measurements from real radar raw data. Full testing of target tracking filters and multi-model detectors could not be performed because the artificial data we used throughout this thesis did not contain clutter. Therefore, it is not known exactly how our study will perform in the case of real radar data. Therefore, target tracking behavior should be re-observed under different noise amounts and clutter models, and the points where the filters are missing or faulty should be determined exactly. Thus, it will be possible to make some improvements in tracking performance in line with the adjustments to be made in the parameters of the filters.

REFERENCES

1. Yang, S. and M. Baum, "Extended Kalman Filter for Extended Object Tracking", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4386–4390, 2017.

2. Shen, J., Y. Liu, S. Wang and Z. Sun, "Evaluation of Unscented Kalman Filter and Extended Kalman Filter for Radar Tracking Data Filtering", *2014 European Modelling Symposium*, pp. 190–194, 2014.

3. Mochnac, J., S. Marchevsky and P. Kocan, "Bayesian Filtering Techniques: Kalman and Extended Kalman Filter Basics", *2009 19th International Conference Radioelektronika*, pp. 119–122, 2009.

4. Zhao, J., M. Netto and L. Mili, "A Robust Iterated Extended Kalman Filter for Power System Dynamic State Estimation", *IEEE Transactions on Power Systems*, Vol. 32, No. 4, pp. 3205–3216, 2017.

5. Bishop, A. N., P. N. Pathirana and A. V. Savkin, "Target Tracking with Range and Bearing Measurements via Robust Linear Filtering", *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 131–135, 2007.

6. Li, T., J. M. Corchado, H. Chen and J. Bajo, "Track a smoothly maneuvering target based on trajectory estimation", *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8, 2017.

7. Lai, C.-P., Y.-J. Ren and C. Lin, "ADS-B Based Collision Avoidance Radar for Unmanned Aerial Vehicles", *2009 IEEE MTT-S International Microwave Symposium Digest*, pp. 85–88, 2009.

8. Gunjal, P. R., B. R. Gunjal, H. A. Shinde, S. M. Vanam and S. S. Aher, "Moving

Object Tracking Using Kalman Filter", *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, pp. 544–547, 2018.

9. Shantaiya, S., K. Verma and K. Mehta, "Multiple Object Tracking Using Kalman Filter and Optical Flow", *European Journal of Advances in Engineering and Technology*, Vol. 2, pp. 34–39, 01 2015.

10. Black, J., T. Ellis and P. Rosin, "Multi View Image Surveillance and Tracking", *IEEE International Workshop on Motion and Video Computing*, pp. 169–174, 2002.

11. Kang, J., I. Cohen and G. Medioni, "Tracking People in Crowded Scenes Across Multiple Cameras", *Asian Conference on Computer Vision*, Vol. 7, p. 15, 2004.

12. Efe, M., *Adaptive Approaches to Manuevering Target Tracking*, Ph.D. Thesis, The University of Sussex, 1998.

13. Feo, M., A. Graziano, R. Miglioli and A. Farina, "IMMJPDA versus MHT and Kalman Filter with NN Correlation: Performance Comparison", *IEE Proceedings on Radar, Sonar and Navigation (Part F)*, Vol. 144, pp. 49–56, 05 1997.

14. Chen, B. and J. Tugnait, "Tracking of Multiple Maneuvering Targets in Clutter Using IMM/JPDA Filtering and Fixed-lag Smoothing", *Automatica*, Vol. 37, pp. 239–249, 02 2001.

15. Cooper, D., "Multiple Target Tracking with Radar Applications", *Electronics and Power*, Vol. 33, No. 6, p. 407, 1987.

16. Kirubarajan, T., Y. Bar-Shalom, W. Blair and G. Watson, "IMMPDAF for Radar Management and Tracking Benchmark with ECM", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 4, pp. 1115–1134, 1998.

17. Basar, T., *A New Approach to Linear Filtering and Prediction Problems*, pp. 167–179, 2001.

18. Li, Q., R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application", *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pp. 74–77, 2015.

19. Wan, E. and R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pp. 153–158, 2000.

20. Mei, W. and Y. Bar-Shalom, "Unbiased Kalman Filter Using Converted Measurements: Revisit", *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference on Signal and Data Processing of Small Targets*, Vol. 7445, p. 74450U, 08 2009.

21. Longbin, M., S. Xiaoquan, Z. Yiyu, S. Z. Kang and Y. Bar-Shalom, "Unbiased Converted Measurements for Tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, pp. 1023–1027, 1998.

22. Yang, H., H. Liu, Z. Zhou and A. Xu, "A Practical Adaptive Nonlinear Tracking Algorithm With Range Rate Measurement", *International Journal of Distributed Sensor Networks*, Vol. 14, p. 155014771877686, 05 2018.

23. Vo, B.-N. and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter", *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, pp. 4091–4104, 2006.

24. Needham, C. and R. Boyle, "Performance Evaluation Metrics and Statistics for Positional Tracker Evaluation", *2003 3rd International Conference on Computer Vision Systems*, pp. 278–289, 01 2003.

25. Naidu, D. V., G. Gopalratnam and S. K. N., "Three Model IMM-EKF for Tracking

Targets Executing Evasive Maneuvers", *45th AIAA Aerospace Sciences Meeting and Exhibit*, 01 2007.

26. Beard, M., B. T. Vo and B.-N. Vo, "OSPA: Using the OSPA Metric to Evaluate Multi-target Tracking Performance", *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 86–91, 2017.