Track-to-Track Association In Distributed Sensor Networks With Redundancies

By
TALHA IRSHAD
Fall 2017-MS SYSE-5
00000206022

Supervisor:
DR. MIAN ILYAS AHMAD

Department Of Computational Engineering

Research Center For Modelling And Simulation
National University Of Sciences And Technology (NUST),
Islamabad, Pakistan.

June, 2021

Track-to-Track Association In Distributed Sensor Networks With Redundancies

By
TALHA IRSHAD
Fall 2017-MS SYSE-5
00000206022

Supervisor
DR. MIAN ILYAS AHMAD

Department Of Computational Engineering

A Thesis Submitted in Partial Fulfillment Of The Requirements
For The Degree OF Masters In Systems Engineering (MS SYSE)

In

Research Center For Modelling And Simulation
National University Of Sciences And Technology (NUST),
Islamabad, Pakistan.

June, 2021

DECLARATION

I, Talha Irshad declare that this thesis titled "track to track association In Distributed Sensor Networks With Redundancies" and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST.

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated.

3. Where I have consulted the published work of others, this is always clearly attributed.

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

5. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Talha Irshad
00000206022

Copyright Notice

Dedication

This thesis is dedicated to my beloved parents and siblings

Acknowledgements

"The function of education is to teach one to think intensively and to think critically. Intelligence plus character – that is the goal of true education"– Martin Luther King.

Alhamdulillah, all the praises belong to the Almighty for giving me strength and perseverance during this thesis. Foremost, I would like to commend my supervisor, Dr Mian Ilyas Ahmad, for his guidance and valuable support. His encouragement and idea sharing was great during this thesis work which paved way for smooth implementation of the research work.
I would also like to appreciate the support offered by my general evaluation committee members Dr. Muhammad Tariq Saeed, Engr. Sikander Hayat Mirza and Dr. Salma Sherbaz. Lastly, I would also like to commend the assistance extended by my peers Engr. Rafi Uz Zaman, Engr. Muhammad Salsabeel Ahmad and Engr. Khaula Ayesha during this research.

TALHA IRSHAD

Contents

List of Abbreviations

STT	Single Target Tracking
TWS	Track While Scan
SAM	Situational Awareness Mode
MSMT	Multi Sensor Multi Target
TDMA	Time Division Multiple Access
FCM	Fuzzy Clustering Means
T2TA	track to track association
M2TA	Measurement To Track Association
M2MA	Measurement To Measurement Association
3D	Three Dimensional
KF	Kalman Filter
PF	Particle Filter
EKF	Extended Kalman Filter
TDL	Tactical Data Link
DA	Data Association
DF	Data Fusion

ECI	Earth Centered Inertial
ECEF	Earth Centered Earth Fixed
GCS	Geodetic Coordinate System
PDAF	Probability Data Association Filter
MHT	Multi Hypothesis Tracker
2D	Two Dimensional
SSPC	Sequential Sum Of Pairwise Costs
SAPC	Sum Of All Pair-Wise Costs
SMNDNN	Sequential Minimum Normalized Distance Nearest Neighbour
IFC	Intrusionistic Fuzzy Clustering

List of Figures

Abstract

Distributed sensor networks that are tracking the movement of multiple objects often sense redundant objects due to measurement noises and network errors. This can cause confusion as the actual number of objects and their actual locations are difficult to identify. track to track association algorithms can be used to overcome this problem. Many algorithms have been proposed in literature and they can be largely classified into two main categories i.e statistical algorithms and clustering based algorithms. One important clustering based algorithm is fuzzy track to track association algorithm which is the focus of this thesis. A variant of the fuzzy track to track association algorithm is tested on a set of data generated through a model that represents a multi sensor multi target scenario environment. In actual sensors the error in tracks is usually induced in azimuth, elevation and range values hence an error model based on azimuth, elevation and range system is proposed in this thesis. The resolutions for the association algorithm are also based on this realistic error model. In addition, time synchronization of tracks is also essential before performing track association. A linear predictor that synchronizes the tracks prior to their association is employed in this thesis and the performance of algorithm is analyzed under time synchronization of tracks. Also a recent technique based on batch processing is studied which improves the performance of fuzzy track to track association algorithm in certain cases. The technique in particular is beneficial for the real-time implementation of the algorithm. The results of our proposed method are compared under various multi sensor multi tar-

get scenarios and it is observed that the proposed method is more than 90 percent accurately associating the given tracks in scenarios where the noise in tracks is low.

Keywords: T2TA,Data Association, Data Fusion, Fuzzy Clustering

Chapter 1

Introduction

1.1 Background

Airborne radars are used for reconnaissance and interception of aerial targets. These radars usually work at very high frequencies and have various modes of operation like Track while Scan (TWS), Situational Awareness mode (SAM) and Single Target Tracking (STT). Due to various design limitations of the airborne radars they have limitation on their ranges. To enhance the performance of aerial missions there is a need to enhance the ranges of these airborne radars.
To overcome the design limitations of the airborne radars multiple radars are connected to each other with a wireless network.This network between a group of air crafts is known as Tactical data link (TDL). The advantage of this approach is that it reduces the information uncertainty and enhances the range of airborne sensor network. But sharing information over a network brings a problem with it, a single target could get represented multiple times. To get rid of this problem a track to track association (T2TA) algorithm followed by a Track fusion (TF) algorithm must be implemented. Consider a scenario shown in Figure 1.1 as following: There are two aircrafts in the scenario moving in the same direction. They are tracking a single target moving away from them. These aircrafts are working in an overlapping coverage region. The radars on aircrafts have inaccuracies due to which they report somewhat different position of the target. When these aircrafts

share their data over the data link, due to sensor inaccuracies, different sampling rates of sensors and communication delays, it is possible that the target gets depicted twice on the individual displays of the aircrafts.

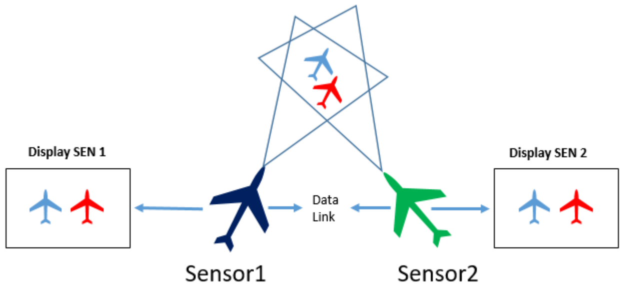

Figure 1.1: Two sensors tracking one target and sharing data over TDL

A robust track to track association algorithm is needed to eliminate this problem of duplicity of targets while sharing data over Tactical data link. The complexity of the problem increases even further when the number of targets and sensor increases as shown in Figure 1.2:

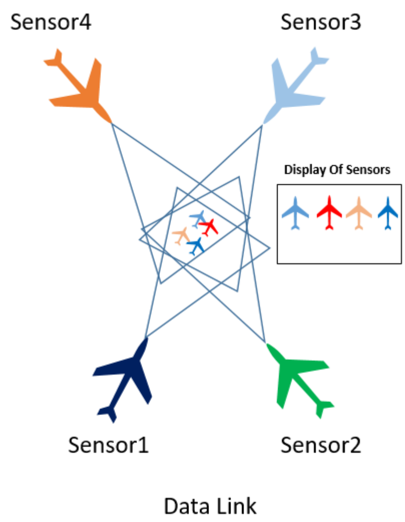

Figure 1.2: Multiple sensors tracking multiple targets and sharing data over TDL

In the aforementioned scenario there are four sensors sharing their data over a TDL. The sensors are operating in an overlapping coverage region, giving rise to the ambiguity that which of the sensors are tracking same targets and which of them are tracking different targets. The following sensor-target combinations are possible of the scenario in figure 1.2.

(A) There is only one target and all the sensors are tracking the same target.
(B) There are four different targets and all the sensors are locking different targets.
(C) There are two different targets, each of sensors is tracking one of the two targets. Another combination is to have one sensor tracking a target and the other three sensors tracking the second target.
(D) There are three different targets, a pair of sensor is tracking the same target while the other two sensors are tracking two different targets.
Multi target tracking systems(MTT) are used to detect, track and counter aerial threats. These systems are composed of various heterogeneous sensors sharing information with each other over a network. Each sensor that is part of this network generates sensor level tracks using its own processed data. System level tracks are generated after fusing the sensor level tracks and the tracks that are

received over the network.
Since sensors can operate in an overlapping coverage region, it is important to identify which tracks belong to same targets and which of them belong to different targets. This process of correlating the tracks prior to fusion is known as track to track association.
Distributed sensor networks (DSN) is an architecture in which multiple sensors are connected to each other via a network. Each sensor in this network is called a node which is generating some piece of information. The network in this architecture could be working on a wired or wireless communication protocol. The nodes part of this architecture could be few large sensors or a huge number of small micro sensors. The nodes could be mobile or stationary in nature. The topology of this architecture could be a stationary or a dynamic one.

1.2 Problem Statement

In modern surveillance systems there are various heterogeneous sensors that are physically distributed in space to gather information on targets in overlapping coverage regions. To generate tracks and finding the actual position of targets there is a need to fuse these tracks. This fusion process can be carried out in two ways, the first being the centralized fusion, in which all the tracks are sent to a central node (fusion center) for fusion which fuses the tracks together and then sends the remaining tracks after fusion to individual nodes. The other being the distributed fusion in which each individual node on the network shares track data with every other node on the network, and every local node performs its own fusion [18][23].
In practical distributed sensor environments with multi target tracking, there is a need to implement a robust track to track association algorithm before performing track fusion. There are various constraints on application of track to track association in real environments as tracks are asynchronous in nature with varying accuracy's of tracks, hence there is a need for implementation of a suitable track to track association algorithm that performs optimally in real environments.Therefore, an algorithm based on fuzzy clustering means algorithm is studied to address the problem of track to track association in this thesis.

1.3 Research Objectives

This thesis concentrates on studying methodologies for track to track association and suggests a new framework for track to track association that is suitable for real time multi-target environments. The main objectives of this thesis are as following:

- Creating a simulation environment for testing the two configurations i.e all at one and pair-wise, of fuzzy track to track association algorithm
- Test the performance of fuzzy track association algorithm under various sensor-target scenarios.
- Test the performance of the algorithm after applying time-synchronization to track data.
- Test the performance of fuzzy algorithm after applying batch processing.
- Test the performance of fuzzy track association after applying altitude and heading filters.

1.4 Motivation

The main aim of undertaking this research topic was its wide ranging applications, some of which include:

- Computer vision.
- Self driving cars.
- Sea, air and surface surveillance.
- Missile defense.

1.5 Thesis Layout

Chapter 2 of this thesis gives a brief overview of the research already conducted in this field, firstly in this chapter we give an overview about working of the radar technology, after that we discuss how targets are tracked using radars. We also discuss the various frame of references that are used in airborne target tracking, track to track association and fuzzy track to track association algorithms described in literature are also discussed in this chapter. Chapter 3 of this thesis describes the methodology we adopted for data generation and testing our algorithm. Chapter 4 presents the results and related discussion on our proposed algorithm for various sensor-target scenarios. Chapter 5 briefly discusses the conclusion of this research work and considers some future directions of this research.

Chapter 2

Literature Review

2.1 Radar Basics

Radar stands for radio detection and ranging, it is a device that enhances one's perception for observation of the environment. Radars are capable of gathering the information about one's environment in adverse conditions like fog, haze, darkness etc. Radars also have an advantage to see objects at very long ranges which makes them a very useful tool for a plethora of applications.
At the very basic level a radar is made up of two antennas for transmission and reception (Bi-static Radar) of electromagnetic waves,an oscillator that can generate electromagnetic radiation and a receiver that can perceive fluctuations in received energy are also part of the radar. Radar works by emitting electromagnetic radiations in the environment, these radiations strike the objects in the environment and return to the radar receiving antenna. On the basis of these reflections from various objects radar finds out its desired objects called as targets.
The reflections from targets give information about their range, azimuth, elevation and velocity; only 3D radars are capable of gathering elevation information.The range of target from the radar is computed by using the information on the time elapsed between emission and reception of the radar signal after hitting the target. The information on the angles of the target are computed using the angle of arrival of the radar signal after striking the target. Radars are capable of detecting both stationary and mobile targets, to distinguish between stationary and mobile

targets radars use the information on shift in the relative frequency of radar and target, hence in this way targets can be discerned from unwanted objects [27]. A basic radar block diagram is given in Figure 2.1:

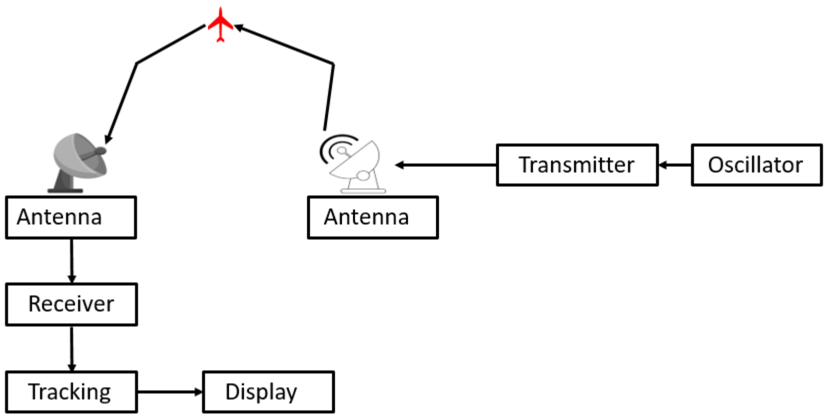

Figure 2.1: Radar Block diagram

There are many types of radars, if distinguished on the basis of physical relationship between transmitting and receiving antenna, radars can be of two types; mono-static and bi-static radars. Mono-static radars use same antenna for transmission and reception of radar signals while in bi-static radars there are different antennas for transmission and reception of radar signals separated by a sizeable distance. Radar classification is also done on the basis of the kinds of signals they transmit. A continuous wave (CW) radar is a radar whose transmitter continuously emit radar signal, on the other hand a pulsed radar is a radar which transmits a relatively short burst of radar signal called as pulses and after every pulse the receiver is turned on to receive the echo. Radars can also be classified on the basis of the type of missions they perform, search and tracking radars are the classifications of radars if differentiated on the basis of mission they perform. Search radars are the radars which continuously scan a designated volume without staying at a specific location, their primary responsibility is to detect the range and direction of the targets. Tracking radars are the radars which continuously follow the targets and are responsible for giving information about the range, azimuth, elevation and doppler of the targets [10].

2.2 Modes of a Radar

On the basis of their functionality, radars can be classified into various modes of operation. Some of the radar modes are given as following:

2.2.1 Pulse Search

This is a traditional technique that uses pulses to accurately find range, angle and speed of a target. This technique is prone to errors as it can easily be deceived by jamming and it also has smaller range as compared to other modes.

2.2.2 Velocity Search

This mode of a radar uses a high pulse repetition frequency (PRF) doppler waveform to determine the azimuth and velocity of an adversary aircraft. This mode performs better when the targets are moving towards the radar in nose direction, its performance degrades in case of tail-on targets.

2.2.3 Track while Scan (TWS)

In this mode the radar maintains tracks on several targets while still searching for other targets. The accuracy of this mode is less as compared to single target track mode as radar is distributing its computational resources among various different targets.

2.2.4 Single Target Track (STT)

This is the most accurate mode of a radar in which the radar continuously directs its energy on a single target. In this mode the radar antenna is continuously following the target where ever it moves.

2.2.5 Dual Target Track (DTT)

In this mode the radar tracks two different targets, the radar antenna automatically predicts the coordinates of the second target and moves in that direction

2.2.6 Situational Awareness Mode (SAM)

In this mode the radar doesn't track any targets, it just keep operator updated about the situation of the surrounding environment.

2.2.7 Raid Assessment

In this mode the radar only maintains a track of a single target but also routinely checks whether other adversary aircrafts are present in its immediate surroundings.

2.3 Radar Tracking

Tracking is a process by which a radar measures essential data on target and uses this data for determining the path of a target and estimating the future positions of a target. Radar usually measures the data on range, azimuth angle, elevation angle and doppler shift or a combination of these to track its targets. Generally any radar can perform as a tracking radar provided its data is handled properly but specifically a tracking radar is a radar that is able to perform angle tracking tasks.

Tracking radars can be further classified into categories based on the method by which they track targets. There are two major types of tracking radars, continuous tracking radar and track while scan (TWS) radar. A Continuous tracking radar provides uninterrupted data on specific target. Whereas track while scan radar provides sampling data on one or more targets. These radars also differ from each other on the basis of their equipment usage.

Airborne tracking radars use three major techniques for target tracking, they are sequential lobing, conical scan and simultaneous lobing. In sequential lobing we use two different radar beams for alternate instants to track the target. Conical scan technique uses a continuously rotating beam for tracking the target. While, the simultaneous lobing technique uses simultaneous beams from two different radar antennas placed at a certain offset to track the targets, this technique is the most accurate of all aforementioned technique's [27]. Simultaneous lobing technique is used in modern mono-pulse radars, which is briefly described below.

2.3.1 Components of a Tracking Radar

Generally a tracking radar can be divided into two main components i.e hardware and software components [12]. These components will be briefly described as following:

2.3.1.1 Hardware Components

A tracking radar has eight major components which work in tandem to perform the tracking function of a radar. These components are as following:
(A) Synchronizing System
(B) Transmitter System
(C) RF (monopulse duplexer) and antenna system.
(D) Receiving system.
(E) Ranging system.
(F) Antenna positioning system.
(G) Presentation system.
(H) IF and RF testing systems.

2.3.1.2 Software Components

(A) Angle tracking

2.4 Filtering for Target Tracking

Real time tracking is an essential feature of a tracking radar. Filtering is the process in which radar data is processed with the help of adaptive processing techniques to track targets i.e to find their range, course, heading ,flight level, speed etc. Filtering also helps in minimizing the measurement errors by utilizing appropriate methods to accurately calculate the speed, position and acceleration of a desired target. The filtered data from several filter banks in the radar processor is the basis of the target tracking process. In tracking real time statuses of targets are updated based on the filtered data obtained from filters in the radar processor [25]. Apart from tracking process filtering also helps in observation and display of the targets of interest. Many filters have been proposed in the literature to accurately track objects based on the types of targets and the application area in which the filters are to be deployed, some of the filters are discussed briefly as following:

2.4.1 Alpha-Beta-Gamma Filter

An alpha beta gamma filter works on a prediction correction model. The filter is initialized after a certain number of measurement data is available to the filter after that using this available measurement data we predict the position and velocity at next time interval and hence in this fashion a target of interest is tracked using an alpha beta gamma filter. The equation of an alpha beta gamma filter are presented as following:

$$X_p(k+1) = X_s(k) + TV_s(k) + \frac{T^2}{2}a_s(k) \tag{2.1}$$

$$V_p(k+1) = V_s(k) + Ta_s(k) \tag{2.2}$$

$$X_s(k) = X_p(k) + \alpha(X_0(k) - Xp(k)) \tag{2.3}$$

$$V_s(k) = V_p(k) + \frac{\beta}{T}(X_0(k) - Xp(k)) \tag{2.4}$$

$$a_s(k) = a_p(k) + \frac{\gamma}{T^2}(X_0(k) - Xp(k)) \tag{2.5}$$

where X, V and a are position, velocity and acceleration respectively. X is the measurement. α, β and γ are the tuning parameters of the filter. These can be disturbed to enhance the performance of the filter [26].

2.4.2 Kalman Filter (KF)

A kalman filter is a more robust and optimal filter as compared to alpha beta gamma filter. It also works on a prediction correction model. The filter uses the currently known information about the target dynamics and projects it forward and tells what the next state of the target will be. The correction part of the filter compares our predicted state with the measurement reported by the radar and compensates for the error. A kalman filter can have many configuration depending on the type of problem at hand but the simplest form of a kalman filter is called a linear kalman filter and its equations are given as following [24]:
Firstly we predict the state vector using the provided state dynamic equations,

$$\hat{X}_{k|k-1} = F\hat{X}_{k-1} + G_{k-1}u_{k-1} \tag{2.6}$$

Where $\hat{X}_{k|k-1}$ is the vector of the predicted state, $\hat{X}_{k-1}$ is the estimated vector of the previous state, u is the input vector, and F and G are the matrices that define the dynamics of the system.
Similarly the covariance associated with the state can also be predicted in a similar fashion as:

$$\hat{P}_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^{T} + Q_{k-1} \tag{2.7}$$

Where $\hat{P}_{k|k-1}$ is the prediction of the error covariance matrix, $\hat{P}_{k-1}$ is the previous estimation of the error covariance matrix and Q is the process noise associated with the process.
After acquiring the predicted values of the state estimate and covariance matrix we can compute the kalman gain of the filter by using following equation:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{2.8}$$

Where H matrix is the matrix defining output equation and R matrix is the measurement noise covariance matrix.
After computation of the kalman gain of the filter we update the state using following equation:

$$\hat{X}_k = \hat{X}_{k|k-1} + K_k(z_k - H_k \hat{X}_{k|k-1}) \tag{2.9}$$

Where z_k is measurement of output equation and K is the kalman gain.
Similarly the update of error covariance matrix is obtained by:

$$P_k = (I - K_k H_k) P_{k|k-1} \tag{2.10}$$

Where I is the identity matrix.

2.4.3 Extended Kalman Filter (EKF)

Linear kalman filter is usually used when the noise in the system of interest is Gaussian. In systems where non-linearity is involved an extended version of the kalman filter known as extended kalman filter is used. The non-linearity of system in extended kalman filter is usually approximated using derivatives of first or second order [29]. An extended kalman filter can be realized with the help of following equations:
The model and observation equation of an extended kalman filter can be given as:

$$\hat{X}_k = f(\hat{X}_{k-1}) + w_{k-1} \tag{2.11}$$

$$Z_k = h(\hat{X}_k) + v_k \tag{2.12}$$

Where $\hat{X}_k$ is the state vector, w_k is the process noise vector, Z_k is the observation vector, v_k is the measurement noise vector, f(.) is the non-linear vector function of the process and h(.) is the non-linear vector function of the measurement.
The initialization of the filter can be done with the following equation:

$$\hat{X}_0^a = \mu_0 \text{ with error covariance } P_0 \tag{2.13}$$

The prediction step of an extended kalman filter works in following fashion:

$$\hat{X}_k^p \approx f(\hat{X}_{k-1}^a) \tag{2.14}$$

$$P_k^p = J_f(\hat{X}_{k-1}^a)P_{k-1}J_f^T(\hat{X}_{k-1}^a) + Q_{k-1} \tag{2.15}$$

Where $\hat{X}_k^p$ is the predicted state vector, f(.) is the non linear vector function of the process, P_k^p is the error covariance matrix, J_f is the jacobian of the non linear vector function of the process and Q is the co-varinace vector of the noise related to the process.
The equations for the correction process of an extended kalman filter are given as following:

$$\hat{X}_k^a \approx \hat{X}_k^p + K_k(z_k - h(\hat{X}_k^p)) \tag{2.16}$$

$$K_k = P_k^p J_h^T(\hat{X}_k^p)(J_h(\hat{X}_k^p)P_k^p J_h^T(\hat{X}_k^p) + R_k)^{-1} \tag{2.17}$$

$$P_k = (I - K_k J_k(\hat{X}_k^p))P_k^p \tag{2.18}$$

Where h(.) is the non linear vector function of the observation, Q_k is the covariance vector of the noise related to the process, R_k is the covariance matrix of the noise related to the measurement, K_k is the filter gain, superscript p denote the predicted quantities and I is the identity matrix.

2.4.4 Particle Filter

Particle filters can be used when the system is non linear and the noise related to the system is non Gaussian. It works under similar conditions as EKF but differs from it in its implementation as it doesn't make linearity assumptions while filtering the data. Kalman filters achieve filtering goals by using linear projections but particle filter instead uses sequential monte-carlo method. Particle filters have shown superiority over extended kalman filters in many real world applications and hence are a very important filtering tool. An algorithmic version of a particle filter is described as following.

The system model which is non linear in case of particle filtering is presented in a following manner:

$$x_k = f(x_{k-1}, u_{k-1}) + \omega_k \tag{2.19}$$

$$y_k = g(x_k) + v_k \tag{2.20}$$

Where x_k is the state vector, f(.) is the system equation and is non linear, g(.) is equation for measurement and y_k is the vector for measurement. ω_k is the system noise that contains all un-modeled dynamics involved, and any mismatches between process and the system, v_k is the measurement noise that models inaccuracies involved during the measurement process, u_k is the control input to the system [14].

The state estimation task of a particle filter involves estimation of the given equation:

$$E_{p(x_{1:n}|y_{1:n})}[x_n] = \int_{x_n} x_n p(x_{1:n}|y_{1:n}) dx_n \tag{2.21}$$

There are two main hindrances to compute the aforementioned equation, (i) sampling directly from $p(x_{1:n}|y_{1:n})$ is not possible and (ii) estimation of the above equation is only feasible in an online recursive manner. Overcoming the problems mentioned before is only possible if we employ an importance function of the form $q(x_{1:n}|y_{1:n})$. This allows the recursive estimation possible and to compute importance weights in time.

$$E_{p(x_{1:n}|y_{1:n})}[x_n] = \int_{x_n} x_n \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} q(x_{1:n}|y_{1:n}) dx_n \tag{2.22}$$

Where $W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})}$ is the recursive weight. The proposal distribution $q(x_{1:n}|y_{1:n})$ can be rewritten as following:

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{1:n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{2.23}$$

Usually the states follow Markov processes $q(x_n|x_{1:n-1}) = q(x_n|x_{n-1})$ so the above equation can be further simplified as:

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{2.24}$$

The target distribution can be simplified in a following manner:

$$p(x_{1:n}|y_{1:n}) = p(x_n|x_{1:n-1}, y_{1:n}) p(x_{1:n-1}|y_{1:n}) \tag{2.25}$$

$$= p(x_n|y_{1:n}) p(x_{1:n-1}|y_{1:n}) \tag{2.26}$$

$$= \frac{p(y_n|x_n) p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} p(x_{1:n-1}|y_{1:n-1}) \tag{2.27}$$

We can recursively update the importance weight using Equations (2.32) and (2.29) as given below:

$$W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} \tag{2.28}$$

$$\propto \frac{p(y_n|x_n)p(x_n|x_{n-1})p(x_{1:n-1}|y_{1:n-1})}{q(x_n|x_{1:n-1}, y_{1:n})q(x_{1:n-1}|y_{1:n-1})} \tag{2.29}$$

$$\propto \frac{p(y_n|x_n)p(x_n|x_{n-1}))}{q(x_n|x_{1:n-1}, y_{1:n}))}W_{n-1} \tag{2.30}$$

$$\propto p(y_n|x_n)W_{n-1} \tag{2.31}$$

The particle filters have following two steps:

- **Prediction:** Every sample is propagated across the system model to get samples from the prior at step $k : x_k(i)^* = f(x_{k-1}(i)^*) + \omega_{k-1}$ where $\omega_{k-1} \sim p(\omega_{k-1})$
- **Update:** In the update step we essentially implements Bayes rule $p(x_k|x_{k-1}, y_k) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$

Although the equations are in the form of density functions but in practical scenarios the filter is run based on discrete samples.

2.5 Tactical Data Link (TDL)

To increase the field of view of a standalone sensor or a group of sensors and to increase the number of inferences that could be drawn from the information, as

more inferences can be drawn from the information from multiple sensors, a concept of tactical data link has evolved under the knowledge area of C4SIR systems. Tactical data links provide a continuous stream of data, gathered from multiple sensors over a wireless network, to command center and also to individual actors involved in the mission. The main advantage of the tactical data link is that it provides the data gathered from multiple sources in near real time, the data may be composed of precise location of the object of interest, its imagery etc [28].
The main purpose of the tactical data link is to provide enhanced monitoring capability at the command center on the mission under progress. It also provides the capability of sending mission related data to the participant nodes from the command center. Standards have also been developed on sharing of different kinds of data that could be shared using tactical data link as depicted in Figure 2.2.

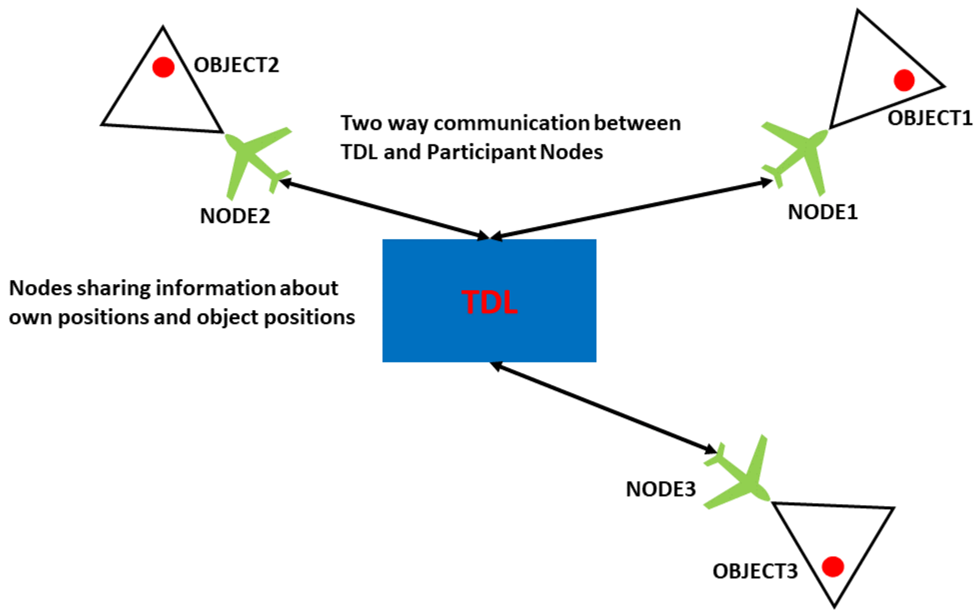

Figure 2.2: A basic concept diagram showing TDL working

2.6 Frames of Reference

In object tracking, reconnaissance and surveillance missions there are multiple disparate sensors involved which are making a complete picture of a mission. These sensors are working on different kinds of frame of reference measuring various parameters required to track and search objects of interest. Some of the most commonly used frames of reference in airborne systems are briefly mentioned as following:

2.6.1 Earth Centered Inertial (ECI)

A frame of reference that is either stationary in nature or is mobile with a constant velocity is said to be an inertial frame of reference. The measurements of an inertial sensor are obtained in reference to an inertial frame that is resolved along the sensitive axis of the instrument. For an inertial frame, the center of mass of earth acts as an origin, the z-axis is along earth's rotation passing through the conventional terrestrial pole (CTP), a right handed system is completed by y-axis and z-axis points towards vernal equinox in an equatorial plane.

2.6.2 Body Frame

In a body frame the origin of the frame lies at the center of the body, x-axis of the frame points in the longitudinal direction, y-axis in the lateral direction while the z-axis is perpendicular to the x-y plane pointing in a downward direction. There is a rigid attachment between a body frame and the body of interest. The geographical representation of this frame can be described with the help of three angles known as Euler angles (ϕ, θ, Ψ). Where ϕ is the roll angle of the body, θ is the pitch angle and Ψ is the yaw angle of the body. Generally this frame is used to represent the outputs of inertial sensors. When the body of interest rotates this frame rotates with the body and its origin moves in the direction of the body. Axes of body frame are shown as following in Figure 2.3:

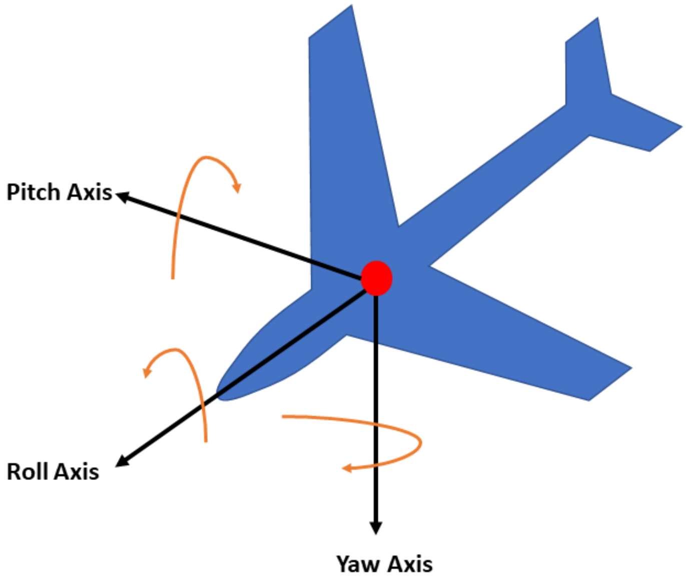

Figure 2.3: Body Frame

2.6.3 Geographic Frame

In navigation the most commonly used frame is the geographic frame, origin of this frame lies at the center of the earth. This frame considers an ellipsoidal nature of the earth and then assigns the point of interest P in a 3-tuple manner as (λ, ϕ, h). Where λ is the latitude, ϕ is the longitude and h is the elevation of the point of interest. Latitude is an angle defined in the meridian plane, it runs from equatorial plane to ellipsoidal normal at point P, longitude is an angle in the equatorial plane running from prime meridian to the projection of point P on the equatorial plane. A geographic frame is depicted in Figure 2.4:

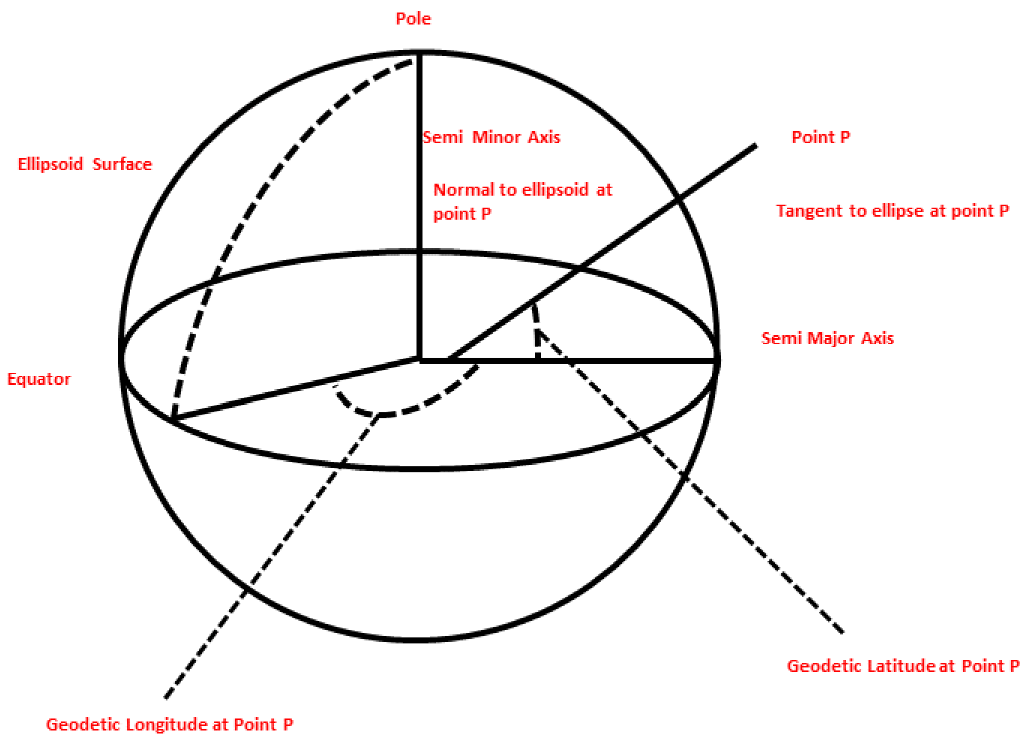

Figure 2.4: Geographic Frame

2.6.4 Earth Centered Earth Fixed Frame (ECEF)

This frame is like geographic frame on the basis of position of origin but ECEF frame is represented in terms of Cartesian coordinates. The x-axes of this frame points in the direction of the north pole and x-y plane is at the equator. Literature shown its better to perform tracking related tasks in this frame. A basis diagram of ECEF frame is shown in Figure 2.5 below:

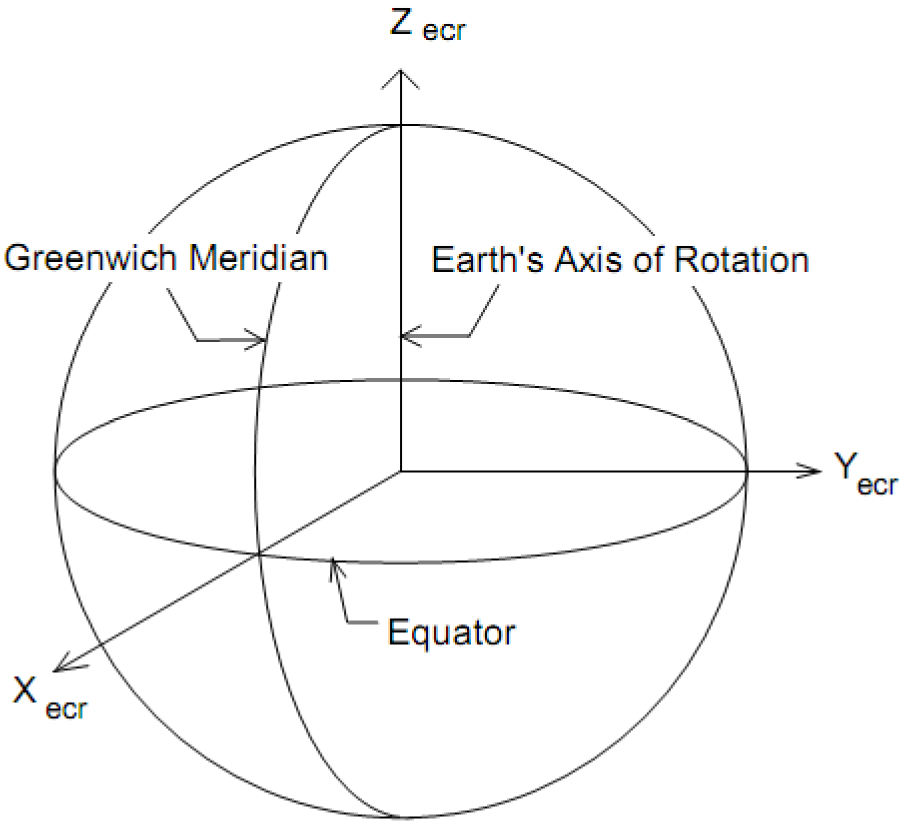

Figure 2.5: Earth Centered Earth Fixed Frame

2.6.5 Local Geodetic Frame

This frame of reference is also known as East-North-Up (ENU) reference system. This frame of reference is determined by fitting a tangent plane to the earth's reference geoid at point of interest P. The x-axis of this frame points toward east, y-axis towards the true north and z-axis points in the upward direction from the surface of earth hence completing the rotation. A diagram of local geodetic frame is shown in Figure 2.6 as following:

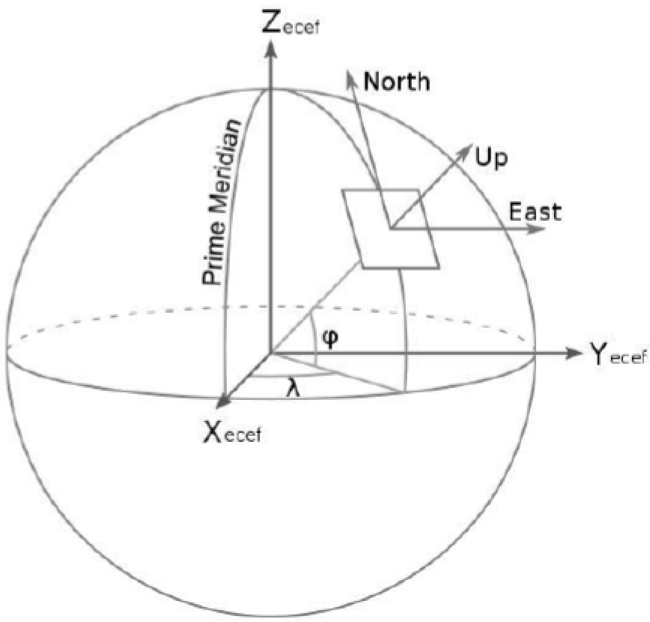

Figure 2.6: Local Geodetic Frame

2.7 Data Association

Data association is one of the most important task in Multi target tracking (MTT) problems. Data association could be simply defined as a hypothesis testing method for comparison of two pieces of information, the results of this hypothesis testing could be either the pieces of information are same or they are different. Data association problem can be broadly categorized into three categories, (a) Measurement to Measurement association (M2MA), (b) Measurement to Track association (M2TA) and (c) track to track association (T2TA). The concept of data association is simply explained with the help of Figure 2.7:

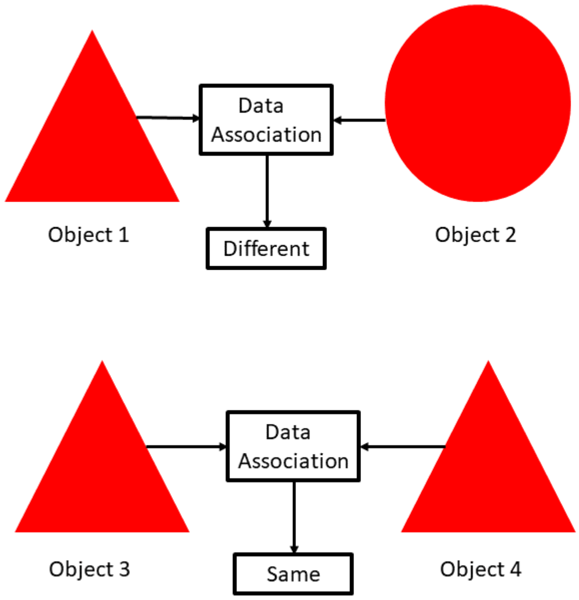

Figure 2.7: Data Association

2.7.1 Measurement to Measurement Association

To initialize a track a radar or sensor obtains a sequence of measurements from consecutive scans, these measurements are then compared to each other on the basis of various geometrical relationships between them. These relationships could be on the basis of distance between measurements, velocity of measurements, angle of measurements etc. If a comparison between these measurements satisfies a predetermined criteria known as threshold then the measurements are associated and a track is initiated on the basis of these associated measurements, this process of track initiation is called as measurement to measurement association [11]. Some of the commonly used track association algorithms are briefly discussed as following:

2.7.1.1 Rule Based Track Initiation

This algorithms initiates a track based on following two conditions:
1- The measured or estimated speed shall be greater than a predefined minimum speed and less than a predefined maximum spreed ($V_{max} < V_{est} < V_{min}$). If the initiation of sensor is performed on the basis of N-scans than the criteria given above is: $V_{min} < ||r_{i+1} - r_i||/t_s < V_{max}$ for i=1,2,...,N-1, Where r_i is the position vector of the target at scan i and t_s is the interval time of measurements.
2- Estimated or measured acceleration shall be less than a maximum predefined acceleration ($a_{max} > a_{est}$), for i=1,2,...,N-1, $||(r_{i+1} - ri)/t_s - (r_i - r_{i-1})/t_s|| < a_{max}t_s$

2.7.1.2 Logic Based Track Initiation

Consider r_k^i is the position of measurement k at scan i. Following steps produce a logic-based initiate track:
1- The process of initiation starts when the first two scans report their measurements, a velocity is estimated based on these two scans i.e., $V^{(e)} = [r_j^2 - r_k^1]/t_s$. Where $r_{k,j}$ are positional data at different time stamps and t_s is the time interval between k and j time stamps. If $V^{(e)}$ satisfies a gating criterion based on speed as, $V_{min} < ||V^{(e)}|| < V_{max}$ then a likely track is posted and the prediction for position is made into the third scan as $r^{(3)} = r_j^2 + V^{(e)}t_s$. 2- For the third scan we set up an acceptance gate having radius r_0 around each potential track that was formed during the second scan. Any measurement falling in the gate $||r_k^3 - r^{(3)}|| < r_0$ will augment the potential tracks of the second scan. We will then compute the velocity $V^{(2e)}$ and acceleration $a^{(2e)}$, where $a^{(2e)} = [V^{(2e)} - V^{(e)}]/t_s$ and using the information we make position prediction as $r^{(4)} = r_i^3 + t_s V^{(2e)} + 1/2 t_s^2 a^{(2e)}$. Where $a^{(2e)}$ is the acceleration at third scan and r_i^3 is the measured position at third scan. If there are multiple measurements satisfying the above criterion the track will be split. If no measurement satisfy the criterion the track will get terminated. If the measurements are not associated with any tracks at any scan together with the unused measurements at the last scan are used to search for new potential tracks by method given in first step.
3- The method explained in second step is repeated for a predefined N number of

scans. Every potential track that remains at the end of the process is a new track.

2.7.1.3 Hough Transform Track Initiation Technique

The points in Cartesian coordinates are mapped using hough transform in $\theta - \rho$ plane as:

$$\rho = x\cos\theta + y\sin\theta \tag{2.32}$$

Where ρ is the distance from the line passing through (x,y) to the origin and θ is the angle with the x-axis to the normal. The range of θ is $0 - 180$ degrees and ρ can take positive and negative values. A curve is generated in $\theta - \rho$ plane for each point in x-y plane and if the points are collinear they will generate a family of curves intersecting at point (ρ_0, θ_0). Hence a straight line track that is initiated in x-y plane is like searching an intersection point in $\theta - \rho$ plane.
We divide the parameter θ into N_θ equal portions, the length of each portion being $\Delta\theta = \pi/N_\theta$ and each portion centered at:

$$\theta_k = (k - 1/2)\Delta_\theta \quad for \quad k = 1, 2, ..., N_\theta \tag{2.33}$$

We calculate lengths for each observation (x_i, y_i) for all θ_n points. We make computations for N measurements for N continuous scans and obtain a set of ρ values given by : $\rho_i(\theta_k) = x_i \cos\theta_k + y_i \sin\theta_k$ for i=1,2,..,N and n=1,2,...,N_θ. We calculate an average of ρ over all the scans for every θ_k and measure deviation of ρ from this average and get the maximum deviation and then another search is performed for finding a minimum deviation over all θ_n. The minimum value of the search is compared with a predefined threshold to initiate a track.

2.7.2 Measurement to Track Association

Once the tracks are initialized, the next task is to maintain those tracks. Measurement to track association algorithms are used for track maintenance process, these algorithms keep a record of initialized tracks and when the new measurements are reported they compare those initialized tracks with the new reported

measurements and find out whether the new reported measurements belong to an already existing track or not, and if it belongs to an already existing track then to which existing track it belongs and extend that track. These association algorithms define a gating region on the basis of a predicted state of each track and if a certain measurement falls inside that gate they get associated [20]. Measurement to track association can be mathematically described as following:
Consider at time step k+1 the set of tracks which were predicted at prior time step can be given as:

$$X = \{(x_1, P_1), ..., (x_n, P_n)\} \tag{2.34}$$

Where $x_1, ..., x_n$ are the states and $P_1, ..., Pn$ are the covariance matrices of errors linked to the states. The track density related to the tracks can be modelled as a Gaussian distribution as:

$$f(x|i) = N_{p_i}(x - x_i) \tag{2.35}$$

And the likelihood function for the sensor at time k+1 is as:

$$f_i(z|x) = N_R(z - Hx_i) \tag{2.36}$$

The set of measurements obtained at time set k+1 is as:

$$Z = \{z_1, ..., z_m\}, |Z| = m \tag{2.37}$$

The measurement to track association can be then defined with a function as following:

$$\Theta : \{1, ..., n\} \to \{0, 1, ..., m\} \tag{2.38}$$

It has the following property: $\Theta(i) = \Theta(i') > 0$ implies that i=i'. The above

function can be explained using following points:
1- For every i=1,...,n, if any $\Theta(i) > 0$ then the measurement $Z_{\Theta(i)}$ associates uniquely with track x_i.
2- There is no track detection as no measurement associates with x_i in case $\Theta(i) = 0$.
3- If any measurements don't meet the criteria laid down in step 1 they are deemed as false detection's.
Many algorithms have been studied in literature, such as, Nearest Neighbor (NN), Global Nearest Neighbor (GNN), Probability Data Association Filter (PDAF), Joint Probability Data Association Filter (JPDAF), Multi Hypothesis Tracker (MHT) etc. to solve the measurement to track association problem. Some of the most commonly used algorithms are discussed briefly as following:

2.7.2.1 Probability Data Association Filter

Consider that we have established a single target track under the influence of noise (clutter). Let P_D be the probability of detection and P_G be the probability that accurate target falls inside the track gate. Let β be the density of extraneous returns and is poisson in nature ($\beta = \beta_{NT} + \beta_{FT}$, where NT is new target and FT is false target). Consider we have M number of observations within the region gate of track j, so we can form M+1 hypotheses on the basis of this data for track j. The first hypothesis is H_0 and represents that none of observations is valid. The probability of H_0 cane be given as:

$$p_{j0} = \beta^M (1 - P_D P_G) \tag{2.39}$$

Correspondingly $H_k (k = 1, 2, ..., M)$ is the hypothesis that observation k is valid and its probability is given by:

$$p_{jk} = \beta^{M-1} P_D P_G \frac{e^{\frac{-d_{jk}}{2}}}{P_G (2\pi)^{\frac{N}{2}} \sqrt{|S_j|}} = \beta^{M-1} P_D \frac{e^{\frac{-d_{jk}}{2}}}{(2\pi)^{\frac{N}{2}} \sqrt{|S_j|}} \tag{2.40}$$

Where S_j is the innovation noise covariance and d_{jk} is the euclidean distance. Ultimately the normalization equation to compute the probability for all M+1 hypotheses is given as:

$$p_{jk} = \frac{p_{jk}}{\sum_{l=0}^{M} p_{jl}} \tag{2.41}$$

A simplified form of p_{jk} after normalization process, as factor β^{M-1} cancels during normalization, can be given as:

$$p_{jk} = \begin{cases} \frac{b}{b+\sum_{l=1}^{M} \alpha_{jl}}, & j = 0 \text{(No valid observation)} \\ \frac{\alpha_{jk}}{b+\sum_{l=1}^{M} \alpha_{jl}}, & 1 \leq \text{ k} \leq \text{ M} \end{cases} \tag{2.42}$$

where

$$b = (1 - P_D P_G)\beta(2\pi)^{\frac{N}{2}}\sqrt{|S_j|}$$

$$\alpha_{jk} = P_D e^{\frac{-d_{jk}^2}{2}}$$

Once the probabilities are computed the next step is to merge the hypotheses. First we compute the residuals associated with M observations that are to be used in a kalman filter update as:

$$y_j(l) = \sum_{k=1}^{N} p_{jk} y_{jk}(l) \tag{2.43}$$

Where

$$y_{jk}(l) = y_k(l) - Hx_i(k|k-1)$$

$$y_k(l) = \text{observation k received at scan l}$$

Then we use standard kalman filter update equation as:

$$x(k|k) = x(k|k-1) + K(k)y(k) \tag{2.44}$$

Where $K(k)$ is the gain. The covariance at scan k is given as:

$$P(k|k) = P^0(k|k) + dP(k) \tag{2.45}$$

Where $P^0(k|k)$ is the covariance of the kalman filter in case if there is only single return and that makes the correct association, and $dP(k)$ is an increment added in case if there are uncertain associations.
Where

$$P^0(k|k) = p_{j0}P(k|k-1) + (1-p_{j0})P^{'}(k|k)$$

$$dP(k) = K(k)[\textstyle\sum_{l=1}^{N} p_{jl}y_{jl}y_{jl}^{'} - y_j y_j^{'}]K(k)^{'}$$

And $P^{'}(k|k)$ is the kalman filter's standard covariance matrix, it is computed as following:

$$P^{'}(k|k) = [I - K(k)H]P(k|k-1) \tag{2.46}$$

The aforementioned equations define a probability data association filter for a single target in clutter, for cases of multiple targets many variants of probability data association filter have been studied in literature [5].
For track initiation and deletion process of a probability data association filter, Interactive multiple model probability data association filter (IMM-PDAF) and Integrated probabilistic data association (IPDA) filter have been realized.

2.7.2.2 Multi Hypothesis Tracker

Multiple hypothesis tracker is best suitable for scenarios when an observation-track conflict situation arises. It is based on a deferred decision logic and creates alternative hypothesis in case of a conflict situation. Consider the scenario in Figure 2.8:

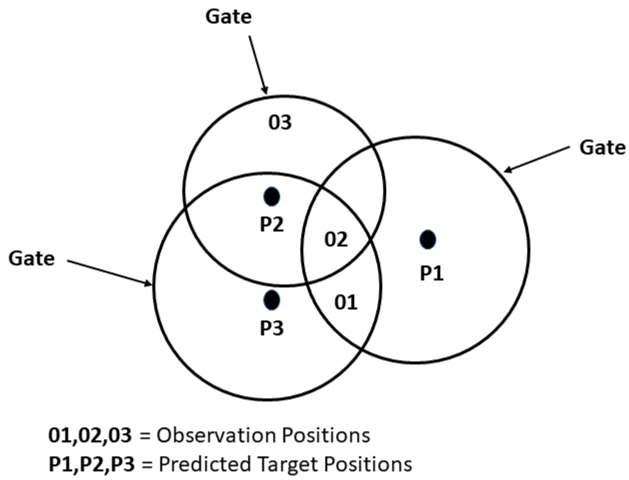

Figure 2.8: Example of an association conflict situation

A flow chart diagram depicting the high level working of MHT is given in Figure 2.9:

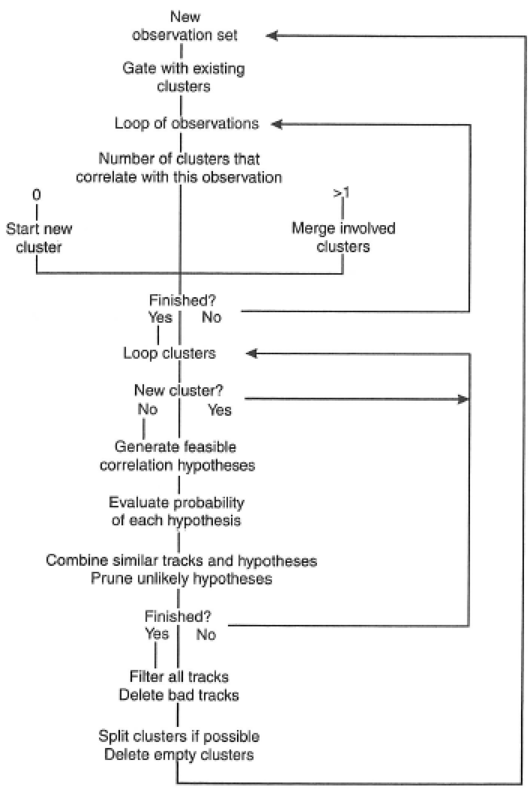

Figure 2.9: Flow chart of MHT

2.7.3 Track-to-Track Association

The aforementioned algorithms are useful at a sensor level only. When multiple sensors are sharing information with each other or sending their information to a centralized location then there is a need to perform an association among this information, the association algorithms which perform a correlation test between such kind of information are known as track to track association algorithms. Track to track association algorithms are used in scenarios when there are multiple sensors involved and are reporting well established tracks. Track to track association presumes that measurement to measurement and measurement to track association has already been performed and sensors are reporting their time-stamped data along with their covariance matrices.

2.7.3.1 Approaches for Track-to-Track Association

The idea of track-to-track association was first propounded by Kaynuck and Singer in their seminal paper on the topic [15]. This idea was further extended by many researchers afterwards, many algorithms have been proposed on track-to-track association [5]. The algorithms employ a cost function that tests the likelihood that tracks reported from two distinct sensors characterize a same or different target. An assignment algorithm could be employed on the costs obtained from each sensor to get the maximum likelihood (ML) pairing of tracks for each sensor. There is a similarity between some track-to-track association algorithms and measurement to track association in a sense that both use cost and assignment algorithms to find out the associations. Track to track association algorithms could be divided into several categories depending on the number of sensors and tracks involved i.e algorithms for two sensors and one track each, algorithms for two sensors and multiple tracks each and algorithms for multiple sensors and multiple tracks each.

Algorithms For Two Sensors and Two Tracks

The most easiest and simplest way of checking whether two tracks reported from two different sensors have originated from a same object or from different objects is pairwise association test. Other algorithms such as clustering algorithms could be used to associate the two tracks but they are not cost effective for such a simple problem. The test is given as following:

$$\tilde{\Delta^{ij}}(k)^T[T^{ij}(k)]^{-1}\tilde{\Delta^{ij}}(k) \leq D_a \tag{2.47}$$

Where,

$$\tilde{\Delta^{ij}}(k) = \tilde{x^i}(k) - \tilde{x^j}(k) \tag{2.48}$$

$$T^{ij}(k) = R^i(k) - R^j(k) \tag{2.49}$$

D_a is a threshold computed based on prior data, $\tilde{x}^i(k)$ is a state vector originating from sensor i and $R^i(k)$ is a covariance matrix of that state from sensor i.
The track state vectors and their related covariance matrices should be in a common frame of reference. The state vectors could be composed of 2D positions, 3D positions , 2D positions and velocities so on and so forth. The complexity of the test will grow by adding more parameters to the vectors. State vectors and covariance matrices are usually coming from the radar tracker and to apply this test successfully they should be time aligned which is generally not possible so predictions are made about the state vectors to apply this test.
Another parameter given in aforementioned Equation 2.52 is D_a which is a threshold used to test the hypothesis that whether the tracks are similar or different. It is usually calculated based on the prior data available on the scenarios under consideration.

Algorithms For Two Sensors and Multiple Tracks

Usually sensors are tracking more than one targets so a simple pairwise association test can not be applied in this case. Many algorithms have been proposed in literature for scenarios in which there are two sensors having multiple targets, some of them are discussed as below:

2D Assignment Problem For Two Sensors

If there are two sensors having more than two tracks each then the 2D assignment algorithm first computes a 2D matrix of costs using Equation 2.52. In this 2D matrix there is a cost computed for every track pair, each cost is then verified with a threshold and all the unlikely track correlations are disregarded. After that the track to track association is a solution of a 2D assignment problem [17]. An example of a 2D matrix is given in Figure 2.10. The constrains to an assignment problem are as following:
(a) The overall cost of summation should be minimum.
(b) There should be one assignment in each row and column at most.

	1	2	3	4	5
1	0.95	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	0.94	0.35
3	0.61	0.02	0.92	0.92	0.81
4	0.49	0.82	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	0.14

Figure 2.10: 2D matrix for two sensors with 5 tracks each

The 2D assignment algorithms have a complexity of $O(n!)$ but it can be reduced to a complexity of $O(n^3)$ by the use of algorithms such as Munkres algorithm, Auction algorithm and Hungarian method. The Figure 2.11 depicted below show the complexity reduction procedure using Hungarian method:

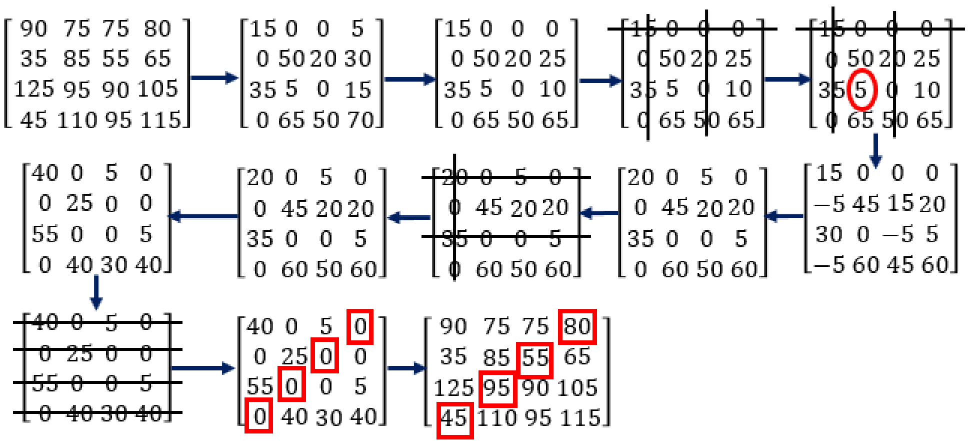

Figure 2.11: Complexity reduction of 2D assignment problem using Hungarian method

Track-to-Track Association based on Weighted State Correlation Similarity

The above mentioned assignment method technique is a single scan technique and requires covariance matrices for associating the tracks. In some cases these covariance matrices are difficult to obtain, the association performance could also be improved by applying batch processing. A track-to-track association algorithm based on weighted state correlation similarity could prove to a better alternative to algorithm mentioned above in cases where accuracy is preferred over time [31]. This method derives a hybrid distance matrix from the correlation coefficients of the covariance matrix which is computed using the sequential states of individual tracks and distance between those target states. The position components of state vector of track l of sensor k are given as:

$$X_l^k(n) = [\hat{x}, \hat{y}, \hat{z}]^T \tag{2.50}$$

The mean of track l of sensor k is given as:

$$\mu_l^k = \frac{1}{N+1} \sum_{n=n0}^{n0+N} X_l^k(n) \tag{2.51}$$

Where, N is the window size. The covariance matrix can then be computed as following:

$$\Sigma_l^k = \frac{1}{N+1} \sum_{n=n0}^{n0+N} [X_l^k(n) - \mu_l^k][X_l^k(n) - \mu_l^k]^T \tag{2.52}$$

We will then obtain the ρ coefficients from the covariance matrix given above, then we will obtain a vector V_l^k which will have the implicit track direction as correlation coefficients have the strength of relative movements between the two random variables.

$$V_l^k = [\rho_{xy}^{l,k}, \rho_{xz}^{l,k}, \rho_{yz}^{l,k}]^T \tag{2.53}$$

Now, the distance between the means and correlations of l^{th} track of sensor-1 and m^{th} track of sensor-2 are given as following:

$$\Delta\mu_{lm}^{12} = (\mu_l^1 - \mu_m^2)^T(\mu_l^1 - \mu_m^2) \tag{2.54}$$

$$\Delta V_{lm}^{12} = (V_l^1 - V_m^2)^T(V_l^1 - V_m^2) + 1 \tag{2.55}$$

When the two targets are same the cost of means and correlations are smaller as compared to when they are different. The cost of associating track l of sensor-1 with track m of sensor-2 is given as:

$$J_{12}(l, m) = \Delta\mu_{lm}^{12}\Delta V_{lm}^{12} \tag{2.56}$$

The proposed algorithm is given as following:

Algorithm 1 Proposed methodology

Require: $X_l^1, X_m^2, L, \in \quad l = 1, 2, ..., M, m = 1, 2, ..., N$

Ensure: $A_{M \times N}$

$L_k = (k-1)L + 1...kL, \quad k = 1, 2, 3, ...$

for k=1,2,...,$\frac{T}{L}$ **do**

  for l=1,2,...,M **do**

    for m=1,2,...,N **do**

      $Calculate : \mu_l^1, \Sigma_l^1, \mu_m^2, \Sigma_m^2$

      $\Delta\mu_{lm}^{12} = f(\mu_l^1, \mu_m^2)$

      $\Delta V_{lm}^{12} = g(\Sigma_l^1, \Sigma_m^2)$

      $J_{12}(l, m) = \Delta\mu_{lm}^{12} \Delta V_{lm}^{12}$

    end for

  end for

  Apply Hungarian algorithm: $J_{12}(l, m) \rightarrow A_k(l, m)$

end for

if $(l \rightarrow m) > \in$ **then**

  $A[l, m] = 1$

else

  $A[l, m] = 0$

end if

return $A[l, m]$

Where, f(.) and g(.) functions represent the Equations presented in (2.56), (2.57) and (2.58) respectively.

Track-to-Track Association based on Sequence Processing of States

Another algorithm for two sensors and multiple tracks is track-to-track association based on sequence processing of states. This algorithms take a batch of estimated states within a predefined time interval, the argument behind this algorithm is that association results obtained on the basis of a batch processed data will be more accurate in contrast to single scan data [30].

The prerequisite for performing association through this algorithm is that the mean vector of the states and covariance matrices related to the states are readily available at every time instant. Let, X_t^{ml} be a state vector with a mean u_t^{ml} and covariance matrix P_t^{ml}, where t denote the time instance, l denotes a track and m is the sensor from which the data is being obtained. Now, we obtain a new random vector by averaging the state vectors across multiple time instants as following:

$$Z^{ml}_{n_{t1}:n_{tn}} = \frac{1}{n_{tn} - n_{t1} + 1} \sum_{t=n_{t1}}^{n_{tn}} X^{ml}_t \tag{2.57}$$

Where, X_t is a state vector, n_{t1} and n_{tn} are the initial and the final time indices of each time interval respectively.
Now we calculate the expected value of the averaged random vector which is described above as following:

$$\mu^{ml} = E[Z^{ml}_{n_{t1}:n_{tn}}] = \frac{1}{n_{tn} - n_{t1} + 1} \sum_{t=n_{t1}}^{n_{tn}} \mu^{ml}_t \tag{2.58}$$

Now, the approximate covariance of the averaged random vector described above, under the assumptions that states are uncorrelated, is given as following:

$$\Sigma^{ml} = \frac{1}{(W')^2} \sum_{t=n_{t1}}^{n_{tn}} P^{ml}_t \tag{2.59}$$

Where, $(W' = n_{tn} - n_{t1} + 1)$ is the window length. Now, usually the states are correlated so we compensate the covariance matrix as following:

$$\hat{\Sigma^{ml}} = \alpha \sqrt{\Sigma^{ml}} \sqrt{\Sigma^{ml}}^T \tag{2.60}$$

Where α is the scaling factor and the roots of the covariance matrices can be obtained by the use of Cholesky factorization. The matrix of costs for associating tracks can be found out easily if μ^{ml} and Σ^{ml} are available for all m and l. The costs for a two sensor case can be found as following:

$$\Pi(a, b)^{12} = (\mu^{1a} - \mu^{2b})^T (\Sigma^{2b})^{-1} (\mu^{1a} - \mu^{2b}) \tag{2.61}$$

Where, $\Pi(a, b)^{12}$ represents the cost of association between track a of sensor1 and track b of sensor2. μ^{1a}, μ^{2b} represent the time averaged means of track a and b obtained from sensor 1 and 2 respectively, and Σ^{2b} is the estimated covariance matrix of track b of sensor2. We then apply the Munkre's algorithm after finding the cost association matrix [21]. Now, the final associations are made on the basis of following expression:

$$A(t)_{0:T} = mode[A(t)_0, A(t)_1, ..., A(t)_T] \tag{2.62}$$

The above expression suggests that $A(t)_k$ is the association result of track t of sensor1 in time interval k, whereas $A(t)_{0:T}$ the track number of sensor2 that track t of sensor1 is most associated in time interval T. This is then selected as the final correspondence of sensor 1.

Algorithms for Multiple Sensors having Multiple Tracks (N-D Assignment Problem)

When there are multiple sensors with each having multiple tracks with them, then a methodology similar to 2D assignment problem can be employed to perform track-to-track association. However, this now becomes an N-D assignment problem having every possible combination of track pairs. This N-D assignment problem is not polynomially bounded and hence is a NP hard problem. So, to solve the N-D assignment problem there are some methods available in literature with their own pros and cons. Some of these approaches are discussed as following:

- Sequential Minimum Normalized Distance Nearest Neighbour (SMNDNN)
- Extension of Pairwise Cost Methods.
- Clustering Based Algorithms.

Sequential Minimum Normalized Distance Nearest Neighbour

For a distributed tracking system coordinates of the local tracks must be synchronized to a common reference time [4]. Once this time synchronization of tracks is achieved then we perform track-to-track association and fusion tasks. The sequential minimum normalized distance nearest neighbour approach uses a distributed

tracker setup and is based on a MDM/OR logic to cater the multi-radar track correlation to fuse the track pairs with minimum normalized distance based on minimum mean squared error (MMSE) methodology. This algorithm is advantageous in a sense that it works sequentially both in time and space. In this technique the fused tracks are generated at each scan of sensor without prior knowledge from the last or past scans, the fused tracks could also be generated by making sensor pairs in space. Essentially, the sequential structure of the algorithm makes it into a one-to-one assignment architecture algorithm [8].
The algorithm works in following manner:

- To initialize this algorithm there must be at least two sensors.
- Apply 2D track association algorithm for the two sensors.
- Fuse the likely track states and their related covariance matrices with the help of following relationship:

$$\tilde{X_{ab}} = R_a(R_a + R_b)^{-1}\tilde{X}_a + R_b(R_a + R_b)^{-1}\tilde{X}_b \tag{2.63}$$

$$R_{ab} = R_a(R_a + R_b)^{-1}R_b \tag{2.64}$$

  Where, X and R are respective state and covariance matrices.
- Once the fused tracks from a pair of sensors are obtained,initialize a synthetic sensor and assign these fused tracks to it. Follow the same technique for all the remaining sensors till there is no sensor left for association [33]. The complete methodology is explained via following Figure 2.12:

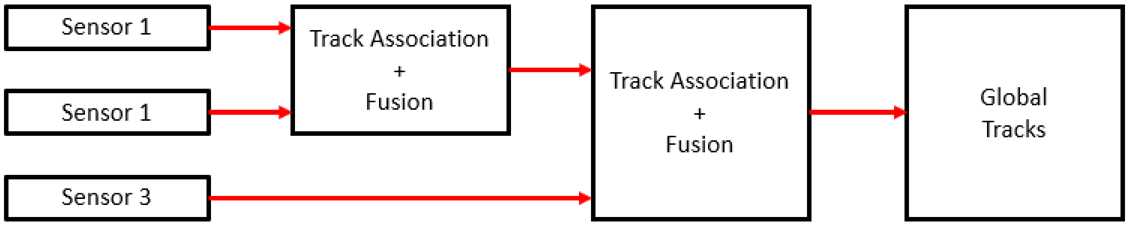

Figure 2.12: Sequential Minimum Normalized Distance Nearest Neighbour

Extension of Pairwise Association Test Methods

In the previous section we considered a pairwise association test which is only applicable when there are only two sensors present. However, by making some alterations pairwise association test can be modified to cater for a multi sensor multi target environment. Extension of pairwise association test methods are realized to extend the pairwise test to multi sensor multi target environment, these methods compute costs which are then compared with a threshold or thresholds to test the hypotheses [16]. Generally, the costs are smaller if the targets are same and they are larger if they are different. Many algorithms can be classified as extension of pairwise association test algorithms, some of them are mentioned as following:

Sequential Sum Cost Method (SSCM)

The mathematical form of this algorithm is presented as following:

$$C(S_1, ..., S_N) = \sum_{i=1}^{N-1}[(S_{i+1} - S_i)^T(P_i + P_{i+1})^{-1}(S_{i+1} - S_i)] \qquad (2.65)$$

Where, C is the cost, S is the state vector and P is the covariance matrix. This method is dependent on the order of the sensors.

Generalized Likelihood Ratio Test (GLRT)

The algorithm is mathematically represented as following:

$$C(S_1, ..., S_N) = \sum_{i=1}^{N} [(S_i - S_{f,1:N})^T (P_i)^{-1} (S_i - S_{f,1:N})] \tag{2.66}$$

Where,

$$S_{f,1:N} = P_{f,1:N} \sum_{i=1}^{N} P_i^{-1} S_i \tag{2.67}$$

$$P_{f,1:N} = (\sum_{i=1}^{N} P_i^{-1})^{-1} \tag{2.68}$$

Where, subscript f represents the fused quantities, C is the cost, S is the state vector and P is the covariance matrix. The performance of this algorithm is quite good, but its drawback is that it has to compute fused states every time before getting to any decision.

Sum of All Pairwise Costs (SAPC)
The mathematical representation of SAPC algorithm is given as following:

$$C(S_1, ..., S_N) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} [(S_i - S_j)^T (P_i + P_j)^{-1} (S_i - S_j)] \tag{2.69}$$

Where, C is the cost, S is the state vector and P is the covariance matrix. As compared to SSPC and GLRT, SAPC algorithm is most suitable of all as it has less computational complexity and is also independent of the ordering of sensors.

Algorithms based on Clustering

Clustering is defined as sorting the data into groups in a manner that similar kinds of data is placed in a same group, and there should be minimum inter-group similarity. Track to track Association tasks limits the choice of clustering algorithms that could be employed to accurately solve the problem. Popular algorithms such as k-means clustering are infeasible for this application, as the data in this case is

in the form of tracks with varying dimensionality, and k-means algorithm requires computation of means of the data which is difficult in this application.
For track-to-track association problem in multi sensor multi target environment, an alternative approach to means based algorithms are distance based clustering algorithms. Distance based algorithms define a distance function between the tracks and based on the output of the distance function cluster the tracks into different groups [13]. Some of distance based clustering algorithms are discussed as following:

Density-Based Clustering

These are a large family of algorithms depending upon the gating of distance parameters for clustering data. These algorithms juxtapose the distance between a a pair of tracks with a distance gate or threshold to determine that if the tracks are close or not. A simple gated clustering and a density based spectral clustering with applications to noise (DBSCAN) are categorized as density based algorithms. A simple version of density based algorithms is presented as following:

Algorithm 2 Clustering Based on Gating

```
for every track do
    if a track is not part of any cluster then
        Put the track in a cluster;
    end if
    for every track with unknown distance to initial track do
        Compute distance;
        if inter track distance is below a certain threshold then
            if second track is part of cluster a cluster then
                Merge the cluster of the two tracks;
            else
                Insert the second track into the cluster of the initial track
            end if
        end if
    end for
end for
```

Spectral Clustering

This is another class of clustering algorithms which is also based on distance based clustering. This class of clustering algorithms is a part of larger class known as dimensionality reduction methods. The main aim of these algorithms is to reduce the dimensionality of the data but retaining the accuracy of the information. Once these algorithms achieve their data compression task, the problem is then solved with any simple clustering algorithm. A simple spectral clustering algorithm is presented as following:

Algorithm 3 Spectral Clustering

1: Compute a distance matrix D based on the given tracks;
2: Compute adjacency matrix A using the distance matrix D;
3: Compute Laplacian matrix L (un-normalized) using adjacency matrix A;
4: Compute the e eigenvectors of L having the smallest magnitude;
5: Use respective elements of each eigenvector to create objects and group them into k clusters using any simple clustering algorithm.

Algorithms based on Fuzzy Clustering

Fuzzy clustering is a subclass of large class of clustering algorithms known as partition clustering algorithms. These algorithms allow a single piece of data to be part of more than one cluster, in contrast to normal clustering in which a single piece of data can be part of only one cluster [22]. If a certain data is to be classified into a certain number of clusters, the normal clustering algorithms assign a probability of either 0 or 1 to individual data points to be a part a of certain cluster. In contrast, the fuzzy clustering assigns probability between 0 and 1 to a certain data point showing its linkage to every possible cluster.

Approaches based on fuzzy clustering have been extensively studied for solving track-to-track association problems, some of these approaches are discussed as following:

Fuzzy Track-to-Track Association in distributed multi-sensor environment

This algorithm is based on a fuzzy clustering means (FCM) algorithm. A fuzzy clustering means algorithm computes a membership matrix M which presents the probability of any data point s_n to be a part of a fuzzy cluster l (with cluster center c_l). The probability of association of data points to a certain fuzzy cluster are called as degrees of membership, they are computed by minimizing the sum of squared errors which are weighted by their related i^{th} power of the degree of membership [2]. The expressions for degree of membership and cluster centers are given as following:

$$m_{ln} = \frac{1}{[\sum_{m=1}^{a}(\frac{d_{ln}}{d_{mn}})^{\frac{2}{i-1}}]} \quad \forall \text{ l,n}, \tag{2.70}$$

$$c_l = \frac{\sum_{n=1}^{b}(m_{ln})^i x_n}{\sum_{n=1}^{b}(m_{ln})^i} \tag{2.71}$$

Where m is the degree of membership, c is the cluster centroid, i is the fuzziness parameter and its value is between 1 and ∞ but the algorithm works optimally between 1 and 2, a is the number of clusters, b is the total number of measurements and X_n is the state vector.
Now to calculate the degree of membership there is parameter d which is the distance between different reports, it is computed as following:

$$d_{ln} = \begin{cases} ||R_n - R_l|| & \text{if } l \neq n \\ ||\Delta_l||^2 & \text{if } l = n \end{cases} \tag{2.72}$$

Where, R is the track report which can have various attributes of targets such as positional data, Δ is the resolution of the sensor from which the report is originating, l is the target and n is the sensor.
By using Equations 2.75 and 2.77 we get two matrices, the first matrix is a distance matrix and the second matrix is a membership matrix, the order of these matrices depend upon the number of reports that are to be associated. Then, the association decision becomes a hypothesis testing problem which is carried out as following:

$$H = \begin{cases} \text{The tracks are same} \quad m_{nl} > m_{ll} \\ \text{The tracks are different} \quad m_{nl} < m_{ll} \end{cases}$$

Intrusionistic Fuzzy Clustering Based Multi-Sensor Track-to-Track Association

To formulate a track-to-track association problem in a multi-sensor environment, consider the following scenario.

Suppose we have N sensors in an overlapping coverage region, they are denoted by $S_1, S_2, ..., S_N$, every one of them is sensing a number of targets T which is unknown at present. Usually the track association is dealt by associating a complete track at once but this algorithm only takes into account single frame of track at time t, which renders the track association problem into a state association one. Assume that sensor S_1 estimates M_1 target states, sensor S_2 estimates M_2 target states and S_N sensor estimates M_N target states. These target states can be represented as following:

$$x_1, ..., x_{M_1}, ..., x_{M_1+M_2}, ..., x_{M_1+M_2+M_3}, ..., x_{M_1+M_2+M_3,...,M_N} \tag{2.73}$$

Where, each target has A attributes of its own as:

$$x_j = (x_{j1}, ..., x_{ji}, ..., x_{jA}) \tag{2.74}$$

Where, x_{ji} is a scalar representing the i^{th} attribute of track state x_j, $1 \leq i \leq A$

The main aim of this technique is to split the $M_1 + M_2 +, ..., M_N$ target states into various groups and make sure that same target represents the states in one group.

The intrusionistic fuzzy sets (IFS) are a generalized form of normal fuzzy sets, the main difference between them is the extension of membership degrees, non membership degrees and the fuzzy set itself with an uncertainty degree and describing that uncertainty degree as following:

$$B = \{< x, \mu_B(x), v_B(x) | x \epsilon X >\} \tag{2.75}$$

Where, X is called as a universe of discourse, μ is the membership degree and v is the non-membership degree. These must follow the conditions as described below:

$$\mu_B(x)\epsilon[0,1], v_B(x)\epsilon[0,1], \mu_B(x) + v_B(x)\epsilon[0,1] \tag{2.76}$$

Now, we define the hesitation degree being mapped from x to B as:

$$\Pi_B(x) = 1 - \mu_B(x) - v_B(x) \tag{2.77}$$

The advantage of this technique over normal fuzzy sets is due to its enhanced ability of dealing with uncertainty [19].
Now, the next step in this technique is to transform the target states and their attributes to intrinsic fuzzy sets attributes. We can express the target states using following matrix:

$$X_s = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1A} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M_1 1} & x_{M_1 2} & \cdots & x_{M_1 A} \\ \vdots & \vdots & \vdots & \vdots \\ x_{(M_1+\ldots+M_N)1} & x_{(M_1+\ldots+M_N)2} & \cdots & x_{(M_1+\ldots+M_N)A} \end{bmatrix}$$

The significance of the above matrix is that each row of the matrix represents one target and the first M_1 rows are for sensor S_1, in a similar fashion all sensors and their relative targets are represented.
Another problem in converting this problem to IFS is, IFS cannot have

negative elements but the states may be negative. So, to eliminate the negative elements from our matrix we will map the negative elements to their positive counterparts in a following manner:

$$\hat{x_{ji}} = x_{ji} - x_{imin} \tag{2.78}$$

where,

$$x_{imin} = min(x_{1i}, x_{2i}, ..., x_{(M_1+...+M_N)i}) \tag{2.79}$$

This transforms the above state matrix as following:

$$\hat{X_s} = \begin{bmatrix} \hat{x_{11}} & \hat{x_{12}} & ... & \hat{x_{1A}} \\ \vdots & \vdots & \vdots & \vdots \\ x_{\hat{M_1 1}} & x_{\hat{M_1 2}} & ... & x_{\hat{M_1 A}} \\ \vdots & \vdots & \vdots & \vdots \\ x_{(M_1+\hat{...}+M_N)1} & x_{(M_1+\hat{...}+M_N)2} & ... & x_{(M_1+\hat{...}+M_N)A} \end{bmatrix}$$

We will now compute the IFS of each corresponding attribute in $\hat{X_s}$ in following manner:

* For every element $\hat{x_{ji}}$ in every column, select two thresholds ϕ_1 and ϕ_2; $\phi_1 < \phi_2$.
* If $|\hat{x_{ji}} - \hat{x_{ki}}| \leq \phi_1, k = 1, 2, ..., M_1 + ... + M_N$, $\hat{x_{ki}}$ can be considered as a membership of $\hat{x_{ji}}$. Suppose $x_{support} = \sum \hat{x_{ki}}$, then the degree of membership of $\hat{x_{ji}}$ can be computed as:

$$\mu(\hat{x_{ji}}) = \frac{x_{support}}{\sum_{m=1}^{n1+n2} \hat{x_{mi}}} \tag{2.80}$$

* If $|\hat{x_{ji}} - \hat{x_{ki}}| \geq \phi_2, l = 1, 2, ..., M_1 + ... + M_N$, then $\hat{x_{li}}$ is considered as a non-membership of $\hat{x_{ji}}$. Suppose $x_{object} = \sum \hat{x_{li}}$ and the non-membership can then be computes as follwoing:

$$v(\hat{x_{ji}}) = \frac{x_{object}}{\sum_{m=1}^{n1+n2} \hat{x_{mi}}} \tag{2.81}$$

Hence we have obtained an IFS matrix as following:

$$X_{ILF} = \begin{bmatrix} [\mu(\hat{x_{11}}), v(\hat{x_{11}})] & ... & [\mu(\hat{x_{1A}}), v(\hat{x_{1A}})] \\ \vdots & \vdots & \vdots \\ [\mu(x_{\hat{M_1 1}}), v(x_{\hat{M_1 1}})] & ... & [\mu(x_{\hat{M_1 A}}), v(x_{\hat{M_1 A}})] \\ \vdots & \vdots & \vdots \\ [\mu(x_{(M_1+\hat{...}+M_N)1}), v(x_{(M_1+\hat{...}+M_N)1})] & ... & [\mu(x_{(M_1+\hat{...}+M_N)A}), v(x_{(M_1+\hat{...}+M_N)A})] \end{bmatrix}$$

Where, $\hat{x_{ji}}$ can be used to calculate the distances which in turn are used to calculate the thresholds; $d_{ji}^k = |\hat{x_{ji}} - \hat{x_{ki}}|$, $\phi_1 = d_{[(M_1+\hat{..}+M_N)]/3}$ and $\phi_2 = d_{[2(M_1+\hat{..}+M_N])/3}$.

We then calculate the association coefficients of the IFS as following:

$$\mathrm{C}(\mathrm{T}_1, T_2) = \frac{\sum_{i=1}^m w_i(\mu_{T1}(x_i)\mu_{T2}(x_i)+v_{T1}(x_i)v_{T2}(x_i)+\Pi_{T1}(x_i)\Pi_{T2}(x_i))}{max(\sum_{i=1}^m w_i(\mu_{T1}^2(x_i)+v_{T1}^2(x_i)+\Pi_{T1}^2(x_i)),\sum_{i=1}^m w_i(\mu_{T2}^2(x_i)+v_{T2}^2(x_i)+\Pi_{T2}^2(x_i)))}$$

Where $w = (w_1, w_2, ..., w_m)$ is the weight vector for states $x_j(j = 1, 2, ..., m)$ with $w_j \geq 0$ and $\sum_{j=1}^m w_j = 1$.

Now, we formulate a matrix for the $(M_1 + M_2 + ... + M_N)^2$ association coefficients. The resultant matrix has $(M_1 + M_2 + ... + M_N)$ rows and $(M_1 + M_2 + ... + M_N)$ columns as following:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1(M_1,\ldots,M_N)} \\ c_{21} & c_{22} & \cdots & c_{2(M_1,\ldots,M_N)} \\ \vdots & \vdots & \vdots & \vdots \\ c_{(M_1,\ldots,M_N)1} & c_{(M_1,\ldots,M_N)2} & \cdots & c_{(M_1,\ldots,M_N)(M_1,\ldots,M_N)} \end{bmatrix}$$

Where, c_{ij} denotes the coefficients for target i and j respectively.
Matrix C is not transitive in nature, firstly we will make this matrix C transitive in nature and after that will perform association. Suppose we have satisfied the transitivity for matrix C then the matrix can be represented as following:

$$S = \begin{bmatrix} S_{11} & S_{12} & \ldots & S_{1N} \\ S_{21} & S_{22} & \ldots & S_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ S_{N1} & S_{N2} & \ldots & S_{NN} \end{bmatrix}$$

Now, we can compute the association matrix from the transitive C matrix as following:

$$S_{ij} = \begin{bmatrix} \hat{c_{11}} & \hat{c_{12}} & \cdots & \hat{c_{1M_j}} \\ \hat{c_{21}} & \hat{c_{22}} & \cdots & \hat{c_{2M_j}} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{c_{M_i1}} & \hat{c_{M_i2}} & \cdots & \hat{c_{M_iM_j}} \end{bmatrix}$$

Where, hat denotes the coefficients of the C matrix after applying transitivity.
The track association rules for two sensors can be stated as following:

* Find the largest element $\hat{c_{ij}^{\phi}}$ in $S_{12}, 1 \leq i \leq M_1$ *and* $1 \leq j \leq M_2$;
* Associate the i^{th} target from S_1 and j^{th} target from S_2 and then make all elements in i^{th} row and j^{th} column zero.

* iterate the algorithm until all the elements in S_{12} are zero.

Fuzzy Double-Threshold Track Association Algorithm

Consider a scenario in which we have two nodes S_1 and S_2 operating in a multi-sensor multi-target environment. The two nodes have their local processing systems and have their own set of target tracks.

Consider $T_1 = (1, 2, ..., k, ..., m1)$ are the tracks from S_1 and $T_2 = (1, 2, ..., l, ..., m2)$ are the tracks from S_2.

The membership functions for this technique can be computed as following:

$$\mu_i = \mu(\mu_i) = e^{(-\tau_i(\frac{\mu_i^2}{\sigma_i^2}))}; \quad i = 1, 2, ..., n \tag{2.82}$$

Where, τ_i is the degree of adjustment, μ is the fuzzy element set for i^{th} parameter and σ_i is the degree of latitude for the i^{th} parameter.

A basic flow diagram of the algorithm is presented in following Figure 2.13:

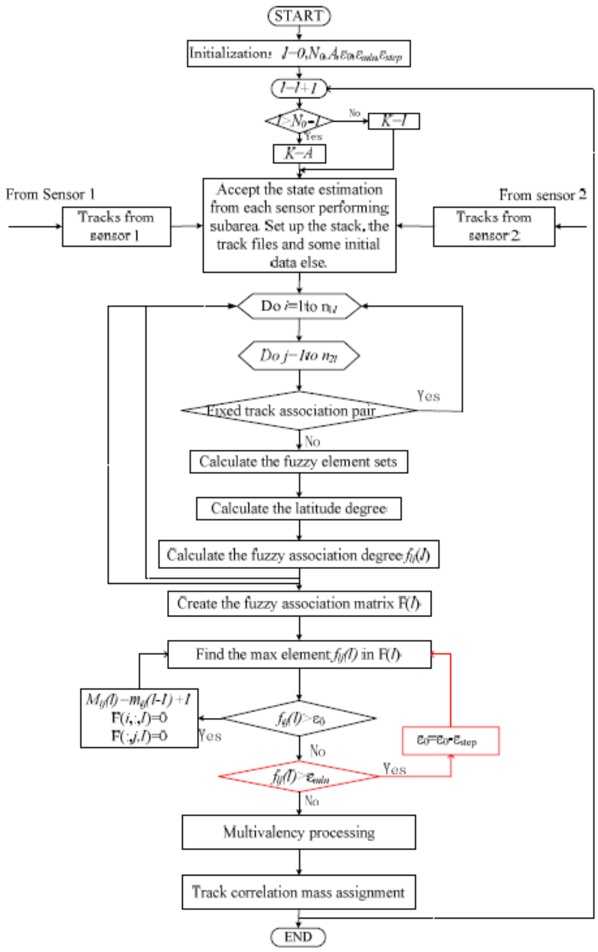

Figure 2.13: Flow diagram of fuzzy double threshold algorithm

The aforementioned algorithm has following steps [6]:

* First of all we have to initialize the parameters. The parameters are the initial moment 1, the threshold N_0, the threshold A, the threshold ϵ_0, the threshold ϵ_{min} where $(\epsilon_0 > \epsilon_{min})$ and the step ϵ_{step}.
* After initialization is done then we have to find out which track pairs are a fixed track association pair. If any two tracks are fixed

association pair then they don't have to qualify the association test anymore. Consider we have two tracks l and m, to find out whether they are fixed association pair we will check the condition in case $l > N_0$ that if the total number of correct association pair during a time period $(l - N_0 + 1 \quad to \quad l) > K$, then track l and m are a fixed association pair. In case $l < N_0$, we have to compute the total number of correct association pair for whole time K.

* Now we will calculate the comprehensive similarity using following expression:

$$f_{lm}(l) = \sum_{b=1}^{m} a_b(l)\mu_b; \quad l\epsilon T_1; m\epsilon T_2 \tag{2.83}$$

  Where a_b is the the assignment weight at moment l.

* Now build an association matrix of the tracks from sensor l and sensor m as below:

$$F(l) = \begin{bmatrix} f_{11}(l) & f_{12}(l) & ... & f_{1T_2}(l) \\ f_{21}(l) & f_{22}(l) & ... & f_{2T_2}(l) \\ \vdots & \vdots & \vdots & \vdots \\ S_{T_1 1}(l) & S_{T_1 2}(l) & ... & S_{T_1 T_2}(l) \end{bmatrix}$$

* Now, if $f_{lm} > \epsilon_0$ track l and m are associated and assign zeros to row l and column m, iterate this process until the max parameter in F(l) becomes less than ϵ_0.
* The next step in this algorithm is to select the correct gate, if the max element in F(l) is greater than ϵ_{min} then decrease the threshold ϵ_0 by order of ϵ_{step} to achieve better track association. Repeat this process until max in F(l) is not less than ϵ_{min}.
* If track l from S_1 correlates with more than one tracks from S_2 then initiate the multi valency processing for track l.

* In this manner associate the tracks via track pair assignment.

Fuzzy Binary Track Correlation Algorithm

This algorithm firstly computes the membership function between a pair of tracks as following:

$$\mu_i(\mu_i) = e^{(-\tau_i(\frac{\mu_i^2}{\sigma_i^2}))} \tag{2.84}$$

After calculating the membership of each pair we apply a weighted average method to compute the following fuzzy association matrix:

$$f_{lm}(l) = \sum_{b=1}^{m} a_b(l)\mu_b \tag{2.85}$$

$$F(l) = \begin{bmatrix} f_{11}(l) & f_{12}(l) & ... & f_{1T_2}(l) \\ f_{21}(l) & f_{22}(l) & ... & f_{2T_2}(l) \\ \vdots & \vdots & \vdots & \vdots \\ S_{T_11}(l) & S_{T_12}(l) & ... & S_{T_1T_2}(l) \end{bmatrix}$$

By aforementioned procedure we can compute a fuzzy correlation matrix between T_1 tracks of sensor S_1 and T_2 tracks of sensor S_2.
Now the next step is to associate the tracks, if $f_{lm} > \epsilon$ then track l is regarded as being correlated with track m. After achieving a correlation we delete the row and column in which association was achieved in $F(l)$. The process is repeated until all the elements remaining in $F(l)$ are less than ϵ, where ϵ is a threshold and is set a $\epsilon \geq 0.5$. In this manner the association between the tracks from two sensors at time l is processed. By using the double threshold method of the automatic radar detection theory, we can select two positive integers Y and Z, $\forall\ l = 1, 2, ..., Z$ in the association test, where $m_{lm}(l) = m_{lm}(l-1) + 1$ and $m_{lm}(0) = 0$, if tracks l and m are associated, otherwise $D_{lm}(l) = D_{lm}(l-1) + 1$.
Where m is an association mass and D is a separation mass between the two tracks at time l. The correlation between tracks is approximately

confirmed when the following condition is achieved:

$$m_{lm}(l-1) \geq L \tag{2.86}$$

The association test between two tracks is terminated when only one pair satisfy the above equation, otherwise if more than one pairs satisfy the above equation multi valency processing is initiated [32]. Similarly the dis-association is tested with the help of following expression:

$$D_{lm}(l-1) \geq Z - L \tag{2.87}$$

The association test is terminated if the above expression is true.
The method for hypothesis testing for association is described as following:

$$g_{lm}(l) = -f_{lm}(l) = -\sum_{i=1}^{m} a_i(l)\mu_i \tag{2.88}$$

We define a binary hypothesis testing as following:

$$\tau_{lm} = \begin{cases} 1, & H_0 \\ 0, & H_1 \end{cases}$$

Where, H_0 means that both the tracks are same and H_1 means the tracks are different. By doing this we have decomposed the association problem into a fuzzy classic assignment problem as following:

$$min \sum_{l=1}^{T_1} \sum_{m=1}^{T_2} \tau_{lm} g_{lm}(l) \tag{2.89}$$

subject to:

$$\text{If} \quad T_1 > T_2, \sum_{l=1}^{T_2} \tau_{lm} \leq 1, \sum_{m=1}^{T_1} \tau_{lm} \leq 1 \tag{2.90}$$

$$\text{If} \quad T_1 < T_2, \sum_{l=1}^{T_2} \tau_{lm} \leq 1, \sum_{m=1}^{T_1} \tau_{lm} \leq 1 \tag{2.91}$$

And,

$$\sum_{l=1}^{T_1} \sum_{m=1}^{T_2} \tau_{lm} = min(T_1, T_2) \tag{2.92}$$

A complete flow chart of the algorithm is depicted as following 2.14:

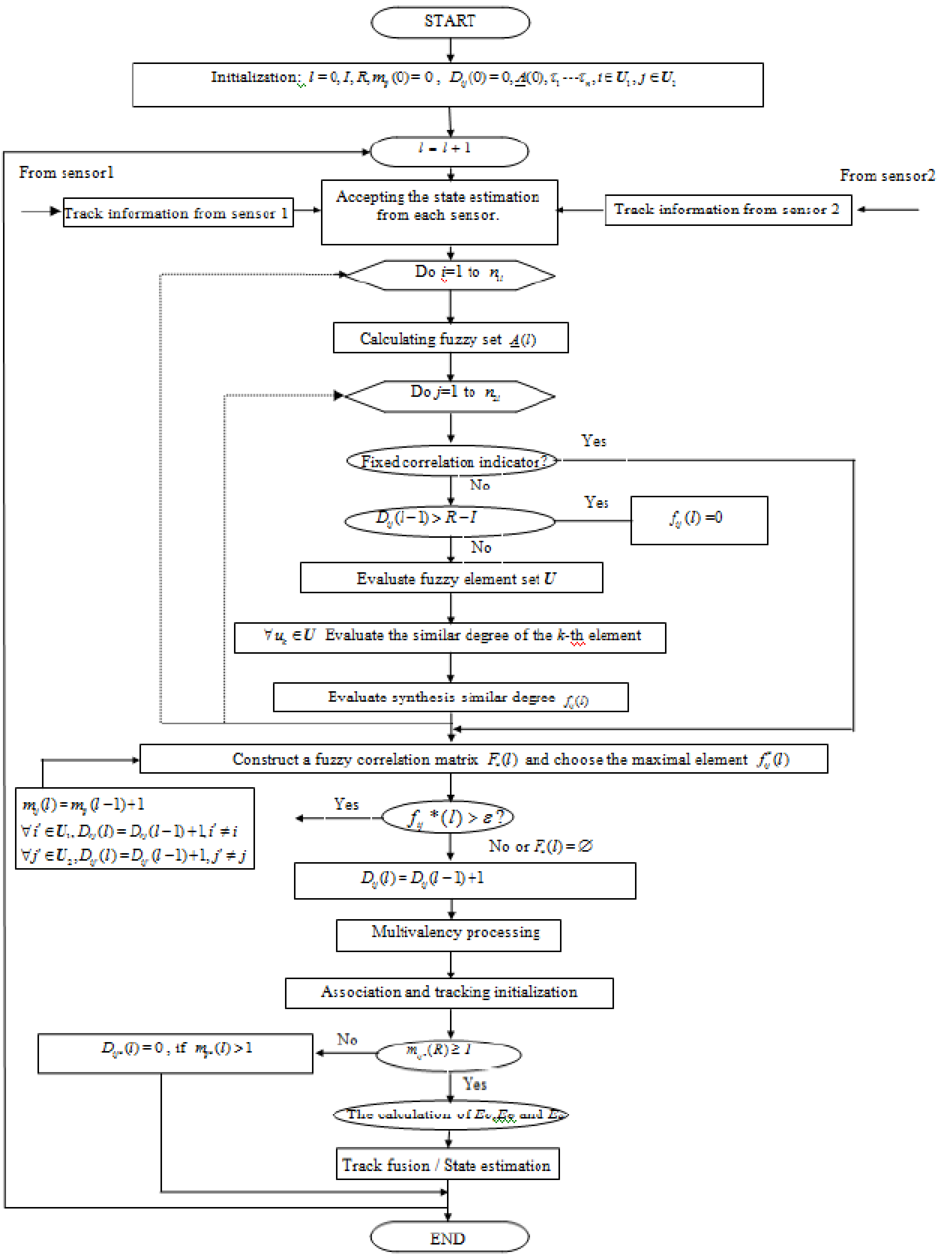

Figure 2.14: Flow diagram of fuzzy binary track correlation algorithm

Fuzzy Track-to-Track Association using Membership Function and Clustering

Consider a scenario in which each sensor is reporting N_R reports to the fusion center. The interpretation, that whether a pair of tracks has originated from same target or a different target, can be computed using fuzzy membership function.
If we want to correlate two tracks l and m, coming from sensors S_1 and S_2 then we have to select the fuzzy attributes at first. These attributes can be the position of the tracks, the velocities of the tracks etc. The membership function based on various attributes are computed as following:

$$\begin{cases} \mu_1(t) = |x_l^{T_1}(t) - x_m^{T_2}(t)| \\ \mu_2(t) = |y_l^{T_1}(t) - y_m^{T_2}(t)| \\ \mu_3(t) = |\hat{x}_l^{T_1}(t) - \hat{x}_m^{T_2}(t)| \\ \mu_4(t) = |\hat{y}_l^{T_1}(t) - \hat{y}_m^{T_2}(t)| \end{cases}$$

Where, x and y are positional estimates of the tracks and x-hat and y-hat are velocity estimates of the tracks.
The degree of membership d_{lm} between two tracks can then be computed as following:

$$\mu_{lm}(\mu_t) = e^{(-\tau(\frac{\mu_t^2}{\sigma^2}))} \tag{2.93}$$

$$d_{lm}(i) = \sum_{i=1} a_i \mu_{lm}(\mu_t) \tag{2.94}$$

Where, $\mu_t(t = 1, 2, 3, 4)$ are the fuzzy attributes as explained above, σ and τ are empirically compute parameters and can be computed by performing a large number of simulations. a is the fuzzy weight of attributes and depends upon the accuracy of various attributes.
The range of the fuzzy membership degrees d_{lm} lies between 0 and 1 with 0 signifying no association between the tracks and 1 signifying complete association between the tracks. The greater the value of d the more strong correlation between the tracks.

The next step in this technique is to perform clustering operation on the N_R track reports from various sensors. Consider we have a fuzzy membership matrix for the N_R tracks being reported as following:

$$F(l) = \begin{bmatrix} d_{11} & d_{12} & ... & d_{1N_R} \\ d_{21} & d_{22} & ... & d_{2N_R} \\ \vdots & \vdots & \vdots & \vdots \\ d_{N_R1} & d_{N_R2} & ... & d_{N_RN_R} \\ \end{bmatrix}$$

The next step is to find the tracks which satisfy the clustering condition, which is $d_{lm} > J$ where J is a threshold which depends on the type of scenario under consideration and is empirical in nature. The d_{lm}'s which satisfy our condition are placed in the same cluster and they signify that track l and n tracks are tracking the same target [9].

Chapter 3

Methodology

The main aim of this chapter is to present a system model that is used to test the performance of the studied algorithm i.e fuzzy track-to-track association algorithm under close to real multi-target tracking conditions. To achieve our goal, firstly we define sensor and target trajectories under consideration that they are being reported in Geodetic coordinate system (GCS). We have also defined a noise model that ensures to make our scenarios as real as possible, for that we have considered standard deviations of sensors in Azimuth Elevation Range (AER) coordinate system and transform these into a Gaussian noise in GCS system. After that we convert our sensor and target trajectories from GCS system to Earth Centered Earth Fixed (ECEF) coordinate system and apply a time synchronization predictor on these sensor and target tracks to align these to a common time reference, once the tracks are time aligned we then apply a fuzzy track to track association algorithm. The chapter ends with a discussion on advantages of batch processing and single scan data for testing of the algorithm. figure 3.1 pictorially explains the system model employed for testing the performance of the track association algorithm.

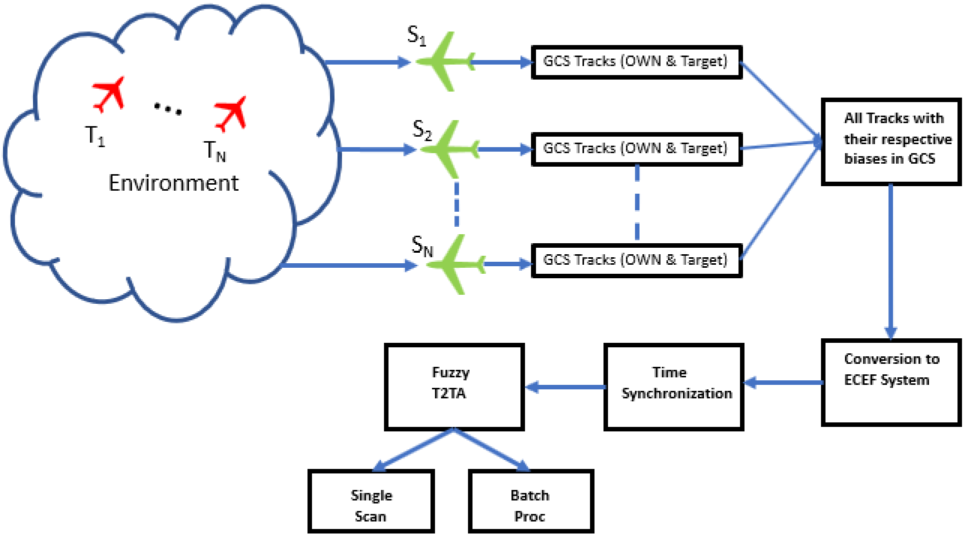

Figure 3.1: System Description

3.1 Multi-Sensor Multi-Target Scenario Generation

To test the performance of our algorithm there is a need to create a multi-sensor multi-target environment that imitates the real world scenarios. To model the system correctly there is a need to create a simulation environment that is able to generate P number of sensor-platforms that are tracking T number of unknown targets. There are two possible ways through which our goal can be achieved, these are mentioned as following:

- Scenario Generator
- Way Point Trajectory Method

3.1.1 Scenario Generator

To model multi-target multi-sensor scenarios, a scenario generator was designed and developed using Matlab app designer in Matlab version 2020a by Mr.Rafi-ul-zaman (NUST-SEECS-MS-EE-16); a peer working on a project with the author of this thesis. The scenario generator has the user interface as shown in the following Figure 3.2.

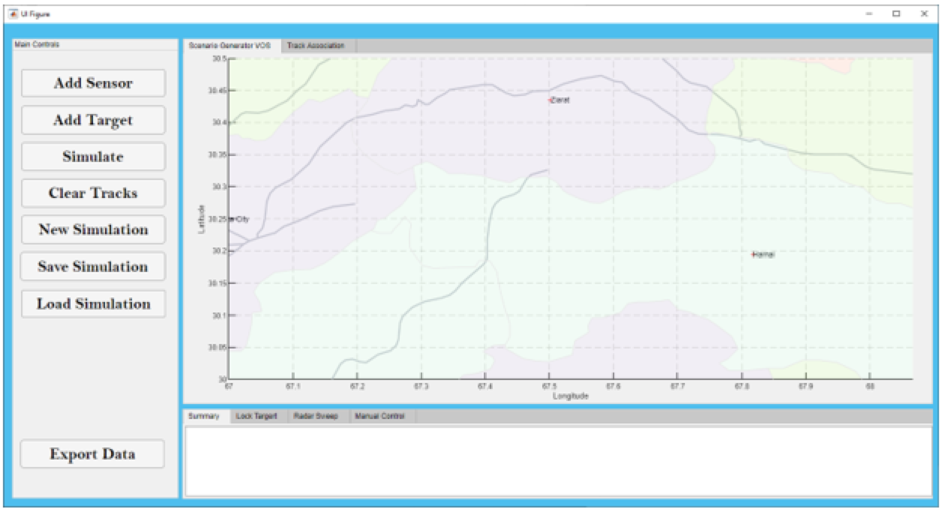

Figure 3.2: Graphical User Interface of Scenario Generator

The scenario generator has following salient features:

- A maximum of 8 sensor-platforms can be added to a certain scenario.
- A maximum of 8 targets can be added to a certain scenario.
- Real-time scenario display on the window at the top-right portion of the interface.
- Saving a certain scenario and using it in the future.
- The data generated on sensor and targets can be ported to a CSV file.
- Provision of unique ID's for sensors and targets.

Consider we have two sensor-platform that are tracking two targets in their environment. To model this scenario using scenario generator first of all we will push the add sensor button twice to add two sensors and set their properties using the options provided by the scenario generator; the properties include giving a particular trajectory to a sensor, setting its speed, heading etc. Once the sensors are added then we add targets in a similar manner as the sensors were added. After adding sensors and targets we push the simulate button and the scenario starts running; the display window on the top-right corner of the interface starts displaying the scenario as shown in the Figure 3.3.

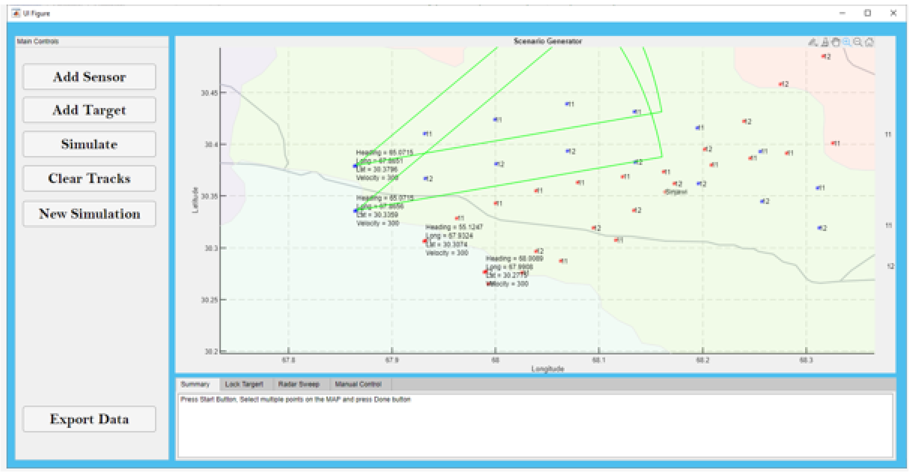

Figure 3.3: Generated Scenario on Two Sensors and Two Targets

In the figure above, the blue points represent the trajectory of the sensor-platforms and the red points represent the trajectory of the targets. The green cone shows the range of the sensors on the sensor-platforms. When a certain target enters into a cone of a certain sensor its data starts to get reported; the data includes the latitude, longitude, altitude, heading and speed of the target in a sensor cone. The data of the sensor-platforms is reported all the times which includes the similar kind of data as targets along with the specific IDs of the sensor platforms. When a target is not in a sensor cone it reports empty data.

3.1.2 Way Point Trajectory Method

The second method to generate the multi-sensor multi-target data for testing and validating our algorithm is way point trajectory method using Matlab. To generate sensor and target tracks using this method there is Matlab navigation toolbox which provides a command for trajectory generation. To generate a track using this command the user needs to provide the way-points which are followed by target and sensors, a sampling rate at which the data is gathered, orientation of the platform and the reference frame in which data is to be generated. For the purpose of this thesis we are gathering our data in geodetic coordinate system. Following Figure 3.4 shows Matlab commands for generating tracks using way points trajectory method:

```
waypoints = [0,0,0; ... % Initial position
             0,1,0; ...
             1,1,0; ...
             1,0,0; ...
             0,0,0];    % Final position

toa = 0:4; % time of arrival

orientation = quaternion([0,0,0; ...
                          45,0,0; ...
                          135,0,0; ...
                          225,0,0; ...
                          0,0,0], ...
                          'eulerd','ZYX','frame');

trajectory = waypointTrajectory(waypoints, ...
    'TimeOfArrival',toa, ...
    'Orientation',orientation, ...
    'SampleRate',1);
```

Figure 3.4: Generate Tracks using Way Points

3.2 Coordinates Conversion

To effectively perform track-to-track association, there is a need to have tracks in same global coordinates. Usually the positions of airborne objects are defined in geodetic frame which has coordinates as Latitude, Longitude and Altitude (λ, ϕ, h).

This coordinate system is also called as East-North-UP (ENU) or North-East-Down (NED). For the purpose of this thesis we are also generating our trajectories in this coordinate system. But usually standard deviations and noise measures which are the bedrock of association algorithms are usually not available in this coordinate system, another problem directly feeding the data in this coordinate system to association algorithm is that a small difference in this coordinate system is a very large real distance and in networked environments where the data is encoded into binary bits there is a chance of data loss. Hence for these reasons firstly we transform the track data from this coordinate system to a Cartesian Earth-Centered Earth-Fixed (ECEF) coordinate system. The advantages of this system are, that it has very well defined distance measures which makes it easy to apply track-to-track association, and it also works well in networked environments. Following paragraph explains a conversion from ENU to ECEF system:

3.2.1 ENU To ECEF

Consider an aerial object whose position in ENU frame of reference is given as $[P_e; P_n; P_u]$. To transform this position into an ECEF position $[x_{ecef}; y_{ecef}; z_{ecef}]$ following equations are used:

$$\begin{bmatrix} x_{ecef} \\ y_{ecef} \\ z_{ecef} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + C \begin{bmatrix} P_e \\ P_n \\ P_u \end{bmatrix}$$

Where, $[x_0; y_0; z_0]$ is the origin of the ECEF coordinates, C is the transformation matrix and is given as following:

$$C = \begin{bmatrix} -\sin(\lambda_0) & -\sin(\phi_0)\cos(\lambda_0) & \cos(\phi_0)\cos(\lambda_0) \\ \cos(\lambda_0) & -\sin(\phi_0)\sin(\lambda_0) & \sin(\phi_0)\cos(\lambda_0) \\ 0 & \cos(\phi_0) & \sin(\lambda_0) \end{bmatrix}$$

Where, ϕ_0 and λ_0 are the longitude and latitude of the origin of the local tangent plane. This conversion can be easily done by using a Matlab command lla2ecef(

3.3 Measurement Inaccuracies

In 3D radars the position of a target of interest are measured in the form of range, azimuth and elevation of that target. There are multiple source which impact these measurement such as clutter, glint, propagation errors, jamming from the foes etc., however,only following errors are considered in the literature [27, 7]:

- Random measurement error dependent on Signal to Noise Ratio (S/N)
- Small errors due to noise (as in certain receptors)
- Bias errors of the radars linked to their calibration.

3.3.1 Accuracy in Range Measurement

The range of a radar is computed in a following manner:

$$R = c\Delta t/2 \tag{3.1}$$

Or as,

$$R = c/2B \tag{3.2}$$

Where, c is the speed of light and B is the bandwidth of the radar channel. The measurement accuracy in the range of a certain radar can be computed using following equation:

$$\sigma_R = \sqrt{\sigma^2_{R_{S/N}} + \sigma^2_{R_{bias}} + \sigma^2_{R_{rnd}}} \approx \sigma_{R_{S/N}} \tag{3.3}$$

The dominant range inaccuracy in range comes from the signal to noise ratio and is given as following:

$$\sigma_{R_{S/N}} = \frac{\Delta R}{\sqrt{\frac{2S}{N}}} \tag{3.4}$$

The equation mentioned above is derived from the Cramer-Rao bound for the time interval. If a radar has a bandwidth of 1 Megahertz then typically its range-resolution is 150m. For a signal to noise ratio of 15db, the resolution $\sigma_{R_{S/N}}$ is 27.5m, if the bias error is 9.5m and the resolution $\sigma_{R_{rnd}}$ is 0.02 times the range resolution which comes out to be 3m. This makes the overall range measurement inaccuracy to be 40m which is the result of higher signal to noise ratio. The internal noise of the radar is usually 1.25 percent to 4 percent of the range resolution, this measures to be 6m in our case.

3.3.2 Accuracy in Angular Measurement

The angular accuracy of a monopulse radar is given by following expression:

$$\sigma_A = \sqrt{\sigma^2_{A_{S/N}} + \sigma^2_{A_{bias}} + \sigma^2_{A_{rnd}}} \tag{3.5}$$

Where,

$$\sigma_{A_{S/N}} = \frac{\theta}{k_m\sqrt{\frac{2S}{N}}} \tag{3.6}$$

Where, the beam width of the radar is θ, and k_m is the slope of the monopulse difference with a typical value of 1.6. The angular noise produced by the internal components of the radar typically ranges from 0.8 percent to 4 percent of the angular resolution, the random angular errors are generally very small if the beam width is kept at 1 degree and signal to noise ratio maintained at 12db. The error $\sigma_{A_{S/N}}$ is usually 0.12 degrees or 2.098mrad. If we consider $\sigma_{A_{bias}}$ to be 0.5mrad and $\sigma_{A_{rnd}}$ to be 0.2mrad then the typical azimuth error becomes 2.7mrad. The table below, Figure 3.5, presents the values for measurement errors for typical radar scenarios:

S.No	Parameter	Value of S/N @5db	Value of S/N @15db	Value of S/N @20db
1	Bandwidth	1MHz	1MHz	1MHz
2	Beamwidth	1 degree	1 degree	1 degree
3	ΔR	150m	150m	150m
4	k_m	1.6	1.6	1.6
5	σ_R @5db($\sigma_{R_{S/N}}$ =47.5m, $\sigma_{R_{bias}}$ =3m, $\sigma_{R_{rnd}}$ =1.5m)	47.62m	27.6m	24m
6	σ_{azm}	0.19deg/3.32 mrad	0.11deg/1.91 mrad	0.10deg/1.74 mrad
7	σ_{elev}	0.19deg/3.32 mrad	0.11deg/1.91 mrad	0.10deg/1.74 mrad
8	Band	X-Band		

Figure 3.5: Typical Values of Radar Inaccuracies

3.3.3 A Simple Error Model for Range-Angle Accuracy

For keeping things simple, a range-angle inaccuracy model is assumed in this thesis. In this model we have assumed that range and angle errors are only dependent on the range of the targets from the sensors. The table below, Figure 3.6, presents the values for measurement errors for certain ranges [1]:

Range (R) / Standard Deviation (σ)	R<13.89Km	R<46.3Km	R>46.3km
σ_R	818.584m	820.436	1804.4036m
σ_θ	0.225deg	0.855deg	0.22296deg
σ_ϕ	0.758deg	0.935deg	0.55508deg

Figure 3.6: Range Based Inaccuracies

3.4 Error Model for GCS and ECEF Tracks

In this thesis we have assumed that the sensor-platforms are working in a distributed manner, this means that every sensor-platform has its own processing unit hence each platform will add its own systematic biases. The tracks generated in this thesis are in geodetic coordinate system having latitude, longitude and altitude as coordinates. These are transformed into ,Cartesian Earth-Centered Earth-Fixed system having x,y,z coordinates, prior to applying track-to-track association. The systematic biases added to the tracks deviate them from their original positions hence their is a need to model them at both ends of the system i.e track generation end and track reception end which is consequently fed into a track-to-track association algorithm.

A technique is proposed in this thesis which computes the error in GCS coordinates and ECEF coordinates if the radar standard deviation in AER coordinates is known, the position of respective sensor in GCS coordinates is known and the position of respective target in GCS coordinates is known.

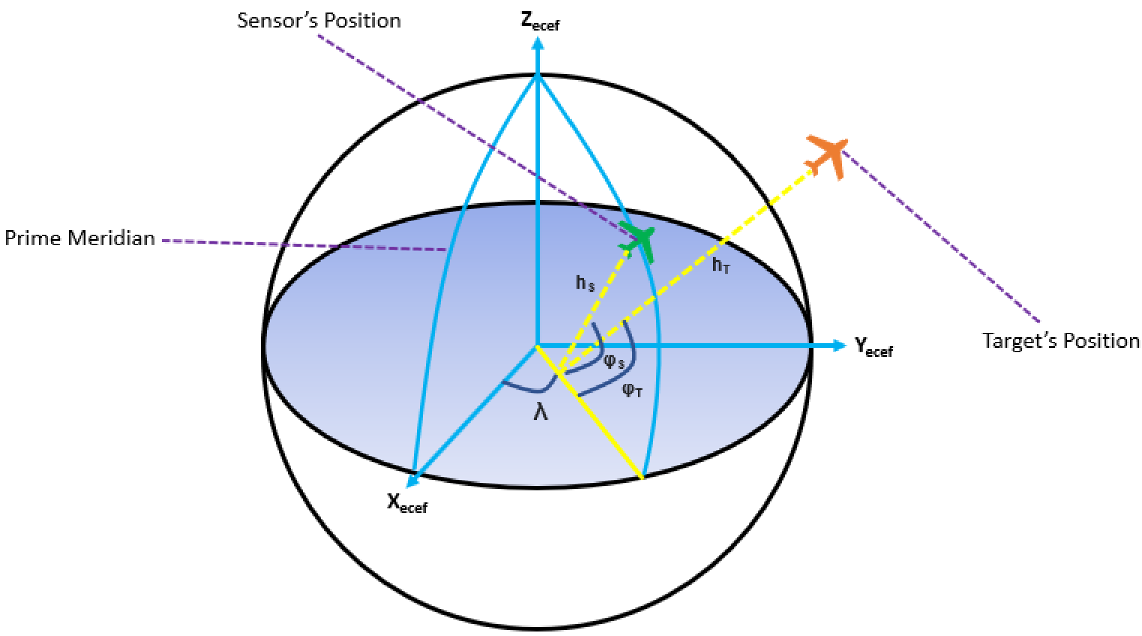

Figure 3.7: Relative Positions of Sensor and Target in Geodetic Coordinate System

In the figure above the sensor is located at geodetic coordinates (ϕ_s, λ_s) having ϕ_s as latitude and λ_s as longitude of the sensor. This position can be translated from geodetic coordinates to Cartesian coordinates with sensor position as the origin in following manner:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} (N(\phi_s) + h_s)\cos\phi_s \cos\lambda_s \\ (N(\phi_s) + h_s)\cos\phi_s \sin\lambda_s \\ (N(\phi_s)(1 - e^2) + h_s)\sin\lambda_s \end{bmatrix}$$

Where the values of parameters are given as:

- Length of semi major axis = a = 6378.137km
- Length of semi minor axis = b = 6356.7523142km
- Flattening of the earth = f = $\frac{1}{298.257223563}$

- Square of first eccentricity = $e^2 = 2f - f^2$
- Length of the normal at sensor's latitude to the z-axis = N = $\frac{a}{\sqrt{1-e^2\sin^2(\phi)}}$
- $\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$, is the Cartesian position of the sensor.

The target position in Cartesian coordinated can be computed in a similar manner as following:

$$\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} = \begin{bmatrix} (N(\phi_T) + h_T)\cos\phi_T\cos\lambda_T \\ (N(\phi_T) + h_T)\cos\phi_T\sin\lambda_T \\ (N(\phi_T)(1-e^2) + h_T)\sin\lambda_T \end{bmatrix}$$

Once we have obtained the sensor and target positions in Cartesian coordinates we then obtain the target position in local coordinates of the sensor by using a rotation matrix which aligns ECEF to ENU coordinates of the sensor as following:

$$\begin{bmatrix} x_{TL} \\ y_{TL} \\ z_{TL} \end{bmatrix} = \begin{bmatrix} -\sin\lambda_s & \cos\lambda_s & 0 \\ -\sin\phi_s\cos\lambda_s & -\sin\phi_s\sin\lambda_s & \cos\phi_s \\ \cos\phi_s\cos\lambda_s & \cos\phi_s\sin\lambda_s & \sin\phi_s \end{bmatrix} \begin{bmatrix} x_T - x_s \\ y_T - y_s \\ z_T - z_s \end{bmatrix}$$

The azimuth angle of the target with respect to radar in local (ENU) coordinates of the sensor can be computed as:

$$\tan\alpha = \frac{x_{TL}}{y_{TL}} \tag{3.7}$$

Where, α is the azimuth angle.
The elevation angle of the target with respect to radar in local (ENU) coordinates of the sensor can be computed as:

$$\tan \beta = \frac{z_{TL}}{\sqrt{x_{TL}^2 + y_{TL}^2}} \tag{3.8}$$

Where, β is the elevation angle.
The slant range between sensor and target can be computed as following:

$$R = \sqrt{(x_s - x_T)^2 + (y_s - y_T)^2 + (z_s - z_T)^2} \tag{3.9}$$

After the coordinates in Azimuth-Elevation-Range coordinate system are available we incorporate the sensor inaccuracies in these coordinates as given below. The typical inaccuracies are available in table form in Figures 3.6 and in table 3.5.

$$\alpha_{new} = \alpha + \sigma_\theta \tag{3.10}$$

Where, α_{new} is the azimuth angle after incorporating the azimuth resolution of the sensor to the measured azimuth of the target in Equation 3.7.

$$\beta_{new} = \beta + \sigma_\phi \tag{3.11}$$

Where, β_{new} is the elevation angle after incorporating the elevation resolution of the sensor to the measured elevation of the target in Equation 3.8.

$$R_{new} = R + \sigma_R \tag{3.12}$$

Where, R_{new} is the slant range after incorporating the range resolution of the sensor to the measured slant range of the target in Equation 3.9.
Once the error resolutions are incorporated, the position of the target in local coordinates of the sensor also changes. The new local coordinate position is computed using following equations:

$$x_{TL_{new}} = R_{new} \cos(\beta_{new}) \sin(\alpha_{new}) \tag{3.13}$$

$$y_{TL_{new}} = R_{new} \cos(\beta_{new}) \cos(\alpha_{new}) \tag{3.14}$$

$$z_{TL_{new}} = R_{new} \sin(\beta_{new}) \tag{3.15}$$

After obtaining the new position of target in local coordinates of the sensor, we can find the new position of target with inaccuracies in Cartesian coordinates by applying following transformation:

$$\begin{bmatrix} x_{T_{new}} \\ y_{T_{new}} \\ z_{T_{new}} \end{bmatrix} = \begin{bmatrix} -\sin\lambda_s & -\sin\phi_s\cos\lambda_s & \cos\phi_s\cos\lambda_s \\ \cos\lambda_s & -\sin\phi_s\sin\lambda_s & \cos\phi_s\sin\lambda_s \\ 0 & \cos\phi_s & \sin\phi_s \end{bmatrix} \begin{bmatrix} x_{TL_{new}} \\ y_{TL_{new}} \\ z_{TL_{new}} \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

After obtaining the new estimate the error in Cartesian coordinates can be obtained as following:

$$\sigma_{xerror} = x_{T_{new}} - x_T \tag{3.16}$$

$$\sigma_{yerror} = y_{T_{new}} - y_T \tag{3.17}$$

$$\sigma_{zerror} = z_{T_{new}} - z_T \tag{3.18}$$

Similarly, the error in geodetic coordinates can be obtained by transforming the new target Cartesian coordinates into new geodetic coordinates using the Matlab command ecef2lla() or by using WGS-84 model. Once the new geodetic coordinates are obtained, the error can simply be found by:

$$\sigma_{lat_{error}} = \Phi_{T_{new}} - \Phi_T \tag{3.19}$$

$$\sigma_{long_{error}} = \lambda_{T_{new}} - \lambda_T \tag{3.20}$$

$$\sigma_{alt_{error}} = h_{T_{new}} - h_T \tag{3.21}$$

3.5 Fuzzy Track-to-Track Association Algorithm

We take help from a simple scenario for formulation of a track-to-track association problem in multi-sensor multi-target (MSMT) setting as shown in Figure 3.8. The scenario has two sensors in an overlapping coverage region observing three distinct targets. A total of four track reports will be generated for these three targets due to overlapping nature of the scenario. Let's consider the target reports are denoted as, T_{op}, $o = 1, 2, 3$ and $p = 1, 2$, have two attributes i.e. the positions in Cartesian coordinates as x and y coordinates of the tracks. In track report T_{op}, o represents the target number and p represents the sensor number.

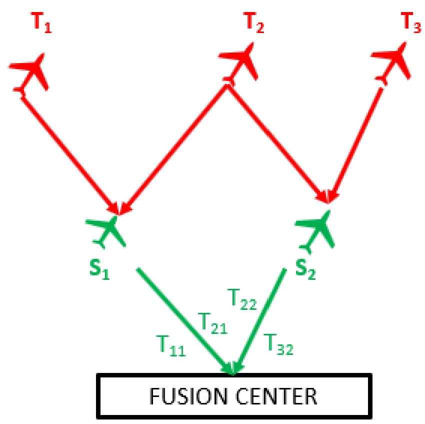

Figure 3.8: Two Sensors and Three Targets in a MSMT Setting

The data reported from the tracks as mentioned in figure above can be presented in the form of a data matrix as presented in table in Figure 3.9. The data matrix contains the information on the tracks being reported to the fusion center at every single scan. The columns of the matrix represent the tracks which are under consideration for correlation, and the rows present the peculiar features of the tracks. The main aim of the fuzzy track-to-track association algorithm or any track-to-track association algorithm,for that matter, is to identify the similar (same target) and dissimilar tracks (different targets) among multiple reported tracks.

Track / Attributes	T_{11}	T_{21}	T_{22}	T_{32}
Position-x (m)	**50**	**75**	**77**	**95**
Position-y (m)	**60**	**85**	**86**	**155**

Figure 3.9: Data Matrix

The data matrix presented in the table above is based on a very simple scenario having very simple data which doesn't require analysis based on clustering. The results obtained from the above matrix concludes that tracks T_{21} and T_{22} represent same target while the other two tracks T_{11} and T_{32} are reporting different targets. But in very large data sets, say for example; having 15 attributes and 100 tracks, the conclusion made on visual inspection fails and requires more advanced technique to perform track-to-track association such as, clustering. Fuzzy clustering algorithm is a very useful technique to solve track association problems, it is explained in detail in the following paragraphs:

The fuzzy track-to-track association algorithm is based on fuzzy clustering means algorithm (FCM), this algorithm formulates a membership matrix M having members m_{oq}, these members represent the membership degrees of the data point x_q within a fuzzy cluster o (having a center of the cluster c_o). The membership degrees are computed by minimizing the sum of squared errors which are weighted by their corresponding a^{th} power of the membership degrees, a is an iterative parameter found after a large number of simulations and works well with a value between 1 and 2 for the problem under consideration. The expressions for membership degrees and cluster centers are given as following:

$$m_{oq} = \frac{1}{[\sum_{p=1}^{n_c} (\frac{d_{oq}}{d_{pq}})^{\frac{2}{a-1}}]} \forall \text{ o,q,} \tag{3.22}$$

$$c_o = \frac{\sum_{q=1}^{n_m} (m_{oq})^a x_q}{\sum_{q=1}^{n_m} (m_{oq})^a} \forall \text{ o,} \tag{3.23}$$

Where n_c represents the number of clusters, and n_m represents the number of measurements in total.

Now, consider we have two tracks so we will initiate two distinct clusters for these. We can then find the optimal membership degrees using the following matrix:

$$DC_{fcm} = \begin{bmatrix} ||x_1 - c_1||^2 & ||x_2 - c_1||^2 \\ ||x_1 - c_2||^2 & ||x_2 - c_2||^2 \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

The membership degrees define the extent of similarity between the members of the matrix DC_{fcm}. We require to match the matrix described above to solve our problem, track-to-track association, effectively. Consider T_o is a column vector having A attributes, these attributes could be range, bearing, speed etc. Every attribute in the column vector will have a related resolution to it, $\Sigma_0, o = 1, 2,$ that depicts the accuracy of the sensor for that attribute. Let's assume that there are two sensors and first sensor is more accurate as compared to the second one i.e $\Sigma_1(p) < \Sigma_2(p) \forall p = 1, 2, ...A$, where A is the attribute number. For performing track-to-track association our main goal is decide whether the two track reports characterize same target or different targets. The scheme of the fuzzy track-to-track association algorithm is to transform all attribute differences between two tracks into a single membership degree (cost) and then compare this single membership degree to another membership degree (threshold), which is computed using the known attribute resolutions of the sensors which are reporting the tracks [2]. Once the single membership degree and threshold between a pair of tracks is available the problem simplifies into a binary hypothesis testing problem as following:

$$H = \begin{cases} 1, & \text{Reported Tracks are Identical} \\ 0, & \text{Reported Tracks are Non-Identical} \end{cases} \tag{3.24}$$

When we are comparing a pair of tracks the track-to-track association decision can be taken in two manners i.e (1) compare resolution of sensor 1 with the distance between tracks from sensor 1 and 2, (2) compare resolution of sensor 2 with the distance between tracks from sensor 1 and 2. The idea is explained by presenting matrix DC_{fcm} for two sensors as following:

$$DC = \begin{bmatrix} ||\Sigma_1||^2 & ||T_2 - T_1||^2 \\ ||T_1 - T_2||^2 & ||\Sigma_2||^2 \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

Where,

$$d_{oq} = \begin{cases} ||T_q - T_o||^2, & \text{if} \quad o \neq q \\ ||\Sigma_o||^2, & \text{if} \quad o = q \end{cases} \tag{3.25}$$

We can then use Equations 3.22 and 3.23 to compute the optimum membership degrees for a scenario of two sensors having a track each as following:

$$m_{11} = \frac{(\Sigma_1' \Sigma_1)^{\frac{1}{1-a}}}{(\Sigma_1' \Sigma_1)^{\frac{1}{1-a}} + ((T_1 - T_2)'(T_1 - T_2))^{\frac{1}{1-a}}} \tag{3.26}$$

$$m_{12} = \frac{((T_1 - T_2)'(T_1 - T_2))^{\frac{1}{1-a}}}{(\Sigma_2' \Sigma_2)^{\frac{1}{1-a}} + ((T_2 - T_1)'(T_2 - T_1))^{\frac{1}{1-a}}} \tag{3.27}$$

$$m_{21} = \frac{((T_2 - T_1)'(T_2 - T_1))^{\frac{1}{1-a}}}{(\Sigma_2' \Sigma_2)^{\frac{1}{1-a}} + ((T_1 - T_2)'(T_1 - T_2))^{\frac{1}{1-a}}} \tag{3.28}$$

$$m_{22} = \frac{(\Sigma_2' \Sigma_2)^{\frac{1}{1-a}}}{(\Sigma_2' \Sigma_2)^{\frac{1}{1-a}} + ((T_2 - T_1)'(T_2 - T_1))^{\frac{1}{1-a}}} \tag{3.29}$$

Hence, a similarity matrix is obtained as given below:

$$S = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

In the matrix above the diagonal elements represent the membership degrees of the sensor thresholds of sensors 1 and 2 respectively, and the off diagonal elements represent the membership degree of the difference between two reported tracks according to respective sensors. The fuzzy association decision now can be taken in two manners i.e (1) on the basis of more accurate sensor and (2) on the basis of less accurate sensor as following:
The fuzzy decision on the basis of more accurate sensor is given as:

$$FD_1 = \begin{cases} 1, & \text{if} \quad m_{21} > m_{11}, \\ 0, & \text{if} \quad m_{21} < m_{11}, \end{cases} \tag{3.30}$$

The fuzzy decision based on the less accurate sensor is given as:

$$FD_2 = \begin{cases} 1, & \text{if} \quad m_{12} > m_{22}, \\ 0, & \text{if} \quad m_{12} < m_{22}, \end{cases} \tag{3.31}$$

Sensors usually have varying resolutions and noises, so to cater the impact of noises on association decisions and make the algorithm more robust, we take the global decisions on the basis of less accurate as following:

$$FD_g = FD_2 \tag{3.32}$$

Hence, the correlation between two reported tracks T_1 and T_2 is given as:

$$CORR(T_1, T_2) = \begin{cases} 1, & \text{if} \quad FD_g = 1 \quad \text{(Tracks are same)}, \\ 0, & \text{if} \quad FD_g = 0 \quad \text{(Tracks are different)}, \end{cases} \tag{3.33}$$

A logical improvement to the method described above is to extend it to a multi-sensor multi-target scenario. This can be achieved very easily by defining a matrix as following:

$$
= \begin{bmatrix} ||\Sigma_1||^2 & ||T_1 - T_2||^2 & ... & ||T_1 - T_{n_T}||^2 \\ ||\Sigma_2||^2 & ||T_2 - T_1||^2 & ... & ||T_2 - T_{n_T}||^2 \\ \vdots & \vdots & ... & \vdots \\ ||\Sigma_{n_T}||^2 & ||T_{n_T} - T_2||^2 & ... & ||T_{n_T} - T_{n_T - 1}||^2 \end{bmatrix}
$$

Where, n_T is the total number of track reports.
The resolution elements can be diagonalized to obtain a matrix similar to matrix DC as following:

$$
DC = \begin{bmatrix} ||\Sigma_1||^2 & ||T_1 - T_2||^2 & ... & ||T_1 - T_{n_T}||^2 \\ ||T_2 - T_1||^2 & ||\Sigma_2||^2 & ... & ||T_2 - T_{n_T}||^2 \\ \vdots & \vdots & ... & \vdots \\ ||T_{n_T} - T_1||^2 & ||T_{n_T} - T_2||^2 & ... & ||\Sigma_{n_T}||^2 \end{bmatrix}
$$

The elements of the above matrix are obtained using:

$$
d_{oq} = \begin{cases} ||T_q - T_o||^2, & \text{if} \quad o \neq q \\ ||\Sigma_o||^2, & \text{if} \quad o = q \quad \text{Where} \quad o, q = 1, 2..., n_T \end{cases} \tag{3.34}
$$

Consequently, the distance matrix is obtained as given below:

$$DC = \begin{bmatrix} d_{11} & d_{12} & ... & d_{1n_T} \\ d_{21} & d_{22} & ... & d_{2n_T} \\ \vdots & \vdots & ... & \vdots \\ d_{n_T1} & d_{n_T2} & ... & d_{n_Tn_T} \\ \end{bmatrix}$$

After obtaining the distance matrix we can then find the similarity matrix using Equations 3.22 and 3.23 as following:

$$S = \begin{bmatrix} m_{11} & m_{12} & ... & m_{1n_T} \\ m_{21} & m_{22} & ... & m_{2n_T} \\ \vdots & \vdots & ... & \vdots \\ m_{n_T1} & m_{n_T2} & ... & m_{n_Tn_T} \\ \end{bmatrix}$$

After obtaining the similarity matrix we can easily get the association decision between any two tracks T_a and T_b, T_b from less accurate sensor, as:

$$CORR(T_a, T_b) = \begin{cases} 1, & \text{if} \quad m_{T_aTb} > m_{T_bT_b} \quad \text{(Tracks are same)}, \\ 0, & \text{if} \quad m_{T_aTb} < m_{T_bT_b} \quad \text{(Tracks are different)}, \end{cases} \tag{3.35}$$

3.6 Track Synchronization

The tracks used in this thesis are of a certain format, Every reported track has a sensor identity from which it is being reported, it has a unique time tag at which it is being reported, it has the positional coordinates of sensor along with the sensor speed and heading, and similarly the positional coordinates of the tracked targets along with their speeds and headings. An example of data packets used in this thesis is shown in Figure 3.10 as following:

SEN-ID	Time-Tag	SEN-LAT	SEN-LONG	SEN-ALT	SEN-SPEED	SEN-HEADING	TAR-LAT	TAR-LONG	TAR-ALT	TAR-SPEED	TAR-HEADING

Figure 3.10: Data Packet

The track data described in preceding paragraph has a time tag assigned to every track, so in order to perform correct associations first of all we have to align the tracks to a common time instance and then apply the association algorithm. Consider a MSMT scenario having two sensors each reporting one track, both sensors having an update rate of 100ms. Each sensor will have an allocated time-slot for reporting its data, consider the scenario starts, sensor 1 reports its data at 100ms then sensor 2 will report its data at 200 ms then sensor 1 will again report its data at 300ms and so on and so forth. Until both the sensor report their respective tracks the association will not be initiated, the data of both sensors will be placed in a data buffer until the association time is reached. The association time will be chosen in a manner in which both sensors have reported at-least two updates of their respective tracks. A concept diagram depicting a time synchronization scenario for two sensors case is presented in Figure 3.11 as following:

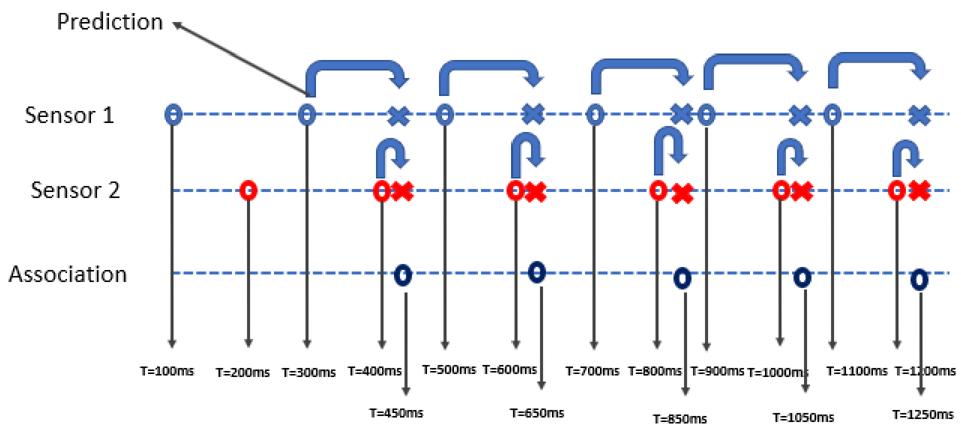

Figure 3.11: Time Synchronization For Two Sensors Scenario

3.6.1 Time Synchronization Filter

For the sake of keeping things simple in this thesis, we have simply used a kalman filter's prediction equation to achieve synchronization between tracks. The inputs to our synchronization filter are the geodetic positions of the sensor and its related target-track with at-least two samples of positional data, the time instants related to the two samples and the time at which prediction is to be made. Once the filter has the required inputs it converts the geodetic positional data into ECEF positional data for the provided two instants. The time difference between the two samples and change in position between elapsed time are computed which are further utilized to compute velocity in Cartesian coordinates. Using the computed velocities, the positional data of latest sample, the time at which prediction is to be made and a constant velocity motion model equation we predict the positions at the desired time.

Let's assume we have converted the geodetic positions to Cartesian positions, and let (x_1, xt_1),(y_1, yt_1) and (z_1, zt_1) be sensor and track positions respectively, at previous instant and (x_2, xt_2),(y_2, yt_2) and (z_2, zt_2) be sensor and target positions respectively, at current instant. Let t_1 be the time for previous instant and t_2 be the time for current instant for sensor and target-track data. Now, the Cartesian velocities for sensor and target can be computed as following:

$$(v_x, v_{xt}) = (\frac{x_2 - x_1}{t_2 - t_1}, \frac{xt_2 - xt_1}{t_2 - t_1}) \tag{3.36}$$

$$(v_y, v_{yt}) = (\frac{y_2 - y_1}{t_2 - t_1}, \frac{yt_2 - yt_1}{t_2 - t_1}) \tag{3.37}$$

$$(v_z, v_{zt}) = (\frac{z_2 - z_1}{t_2 - t_1}, \frac{zt_2 - zt_1}{t_2 - t_1}) \tag{3.38}$$

Let, 50ms after t_2, be the time at which we have to make prediction, so we can use following equation to make prediction:

$$\begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \\ v_{xnew} \\ v_{ynew} \\ v_{znew} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ v_x \\ v_y \\ v_z \end{bmatrix}$$

Where, T=50ms.

In a similar manner we can predict the velocities and positions for target-track. After the prediction the positions can be obtained using following equation:

$$\begin{bmatrix} x_{pred} \\ y_{pred} \\ z_{pred} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \\ v_{xnew} \\ v_{ynew} \\ v_{znew} \end{bmatrix}$$

Once the predicted sensor and target positions are obtained, we pass these predicted positions to the error model presented in section 3.4 to make the model more realistic.

3.7 Fuzzy Algorithms Used in this Thesis

Three different algorithmic versions of fuzzy track-to-track association algorithms were studied in this thesis, these are mentioned as following:

- Converted Measurement Fuzzy Track-to-Track Association Algorithm.

Algorithm 4 CMF Algorithm

1: Initialize with Sensor and Target tracks in geodetic coordinates.
2: Add noise using proposed model to geodetic coordinates of tracks.
3: Convert positions of Sensors and Targets into ECEF coordinates.
4: Apply time synchronization to tracks.
5: Apply noise model to synchronized data to obtain Cartesian resolution of tracks.
6: Apply fuzzy track-to-track association algorithms.
7: Obtain the results.

- Fuzzy Track-to-Track Association Algorithm With Speed and Heading Filters

Algorithm 5 HSF Algorithm

1: Initialize with Sensor and Target tracks in geodetic coordinates time synchronized data.
2: Convert positions of Sensors and Targets into ECEF coordinates.
3: Apply fuzzy track-to-track association algorithm based on positional data.
4: Apply fuzzy track-to-track association algorithm based on Heading.
5: Apply fuzzy track-to-track association algorithm based on Speed.
6: Give weightage to fuzzy algorithms in order: positional > heading > speed.
7: Obtain the results.

- Window Based Fuzzy Track-to-Track Association Algorithm

Algorithm 6 Windowing Based Fuzzy Algorithm

1: Initialize with Sensor and Target tracks in geodetic coordinates.
2: Add noise using proposed model to geodetic coordinates of tracks.
3: Convert positions of Sensors and Targets into ECEF coordinates.
4: Apply time synchronization to tracks.
5: Apply noise model to synchronized data to obtain Cartesian resolution of tracks.
6: Set the window size and apply fuzzy track-to-track association algorithms.
7: Average the fuzzy results over applied window length
8: Obtain the results with windowing.

Chapter 4

Results And Discussion

To test the performance of fuzzy track-to-track association algorithm in different settings, a scenario based methodology based on MSMT scenarios was employed in this thesis. Following sections will present various MSMT scenarios along with the performance of fuzzy track-to-track association algorithm in those scenarios.

4.1 Scenario: Two Sensors One Target

Consider a scenario in which there are two sensors working in an overlapping coverage region. Both the sensors are tracking the same target. The scenario is borrowed from [3], in this scenario there are two ground sensors situated at $(-2, 85)$km and $(4, 85)$km, there is a target with initial position of $(3, 86.6)$km. The sensors are depicted with red circles and the target trajectories from sensor-1 and sensor-2 are presented using green and blue asterisk trajectories in Figure 4.1. The target is moving at a speed of 300m/s. The target moves in a following manner, the target is initially moving towards south-east at an angle of $-135°$. At t=15s the target starts making a change in its course at a constant turn-rate of $4°/s$ until t=26s and moves toward east. At t=35s the target makes another change in its course at a constant turn-rate of $4°/s$ until the end of scenario at t=61s and starts moving toward north-east.

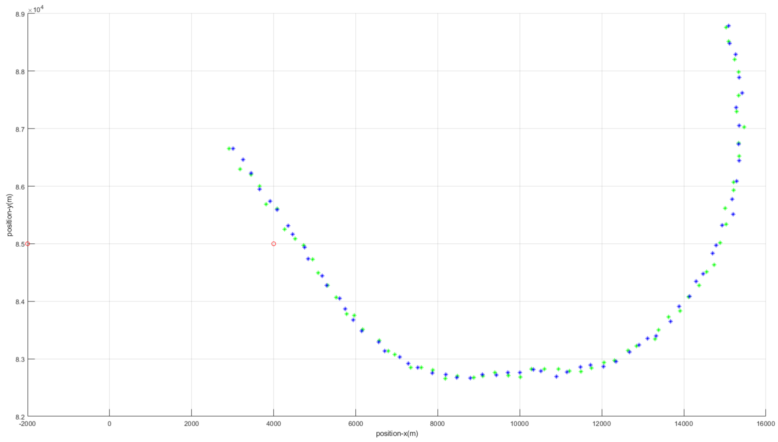

Figure 4.1: Sensor and Target Positions

Both the sensors have a range resolution σ_R of 10m and a bearing resolution of σ_b of $\frac{2}{3}mrad$.

4.1.1 Results

To obtain the results we firstly transformed the tracks from NEWS coordinate system to Cartesian coordinates, along with position transformation we also transformed the resolutions of sensors into a Gaussian noise in Cartesian coordinates along respective dimensions. Once the tracks were obtained with their respective error they were compared with each other using fuzzy track-to-track association algorithm, the results are presented as following:

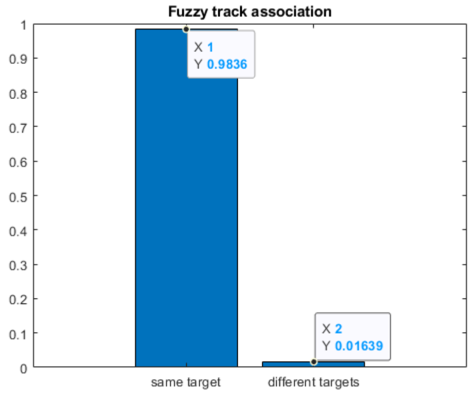

Figure 4.2: Results: Two Sensors Single Target

Figure 4.2 depicts that the algorithm was fairly able to associate the two tracks as a single target with an accuracy of 98.36 percent. The scenario was run for 1000 monte-carlo simulations and the accuracy results were averaged.

4.2 Scenario: Two Sensors Two Targets

Consider a scenario in which there are two sensors working in a distributed fusion environment, The sensors are sharing their tracks with each other. Consider there are two targets in the environment of the sensors, sensor-1 is tracking target-1 and sensor-2 is tracking target-2. The sensors are situated at $(2, 90)$km and $(2, 80)$km respectively. The initial positions of targets are $(10, 90)$ km and $(10, 90.3)$km respectively. Initially both targets move in parallel at a speed of 300m/s, they are moving towards south-east at an angle of -135°. At t=15s target-1 starts making a change in course with a constant turn-rate of $4°/s$ until t=26s and moves towards east. At t=15s target-2 starts making a change in course with a constant turn-rate of -$4°/s$ until t=26s and moves towards south. At t=35s target-1 makes another course change with a constant turn-rate of $4°/s$ until the end of scenario t=61s and starts moving towards north-east. At t=35s target-2 also makes another course change with a constant turn-rate of -$4°/s$ until the end of scenario t=61s and starts moving towards south-west. The scenario is depicted in Figure 4.3 as following:

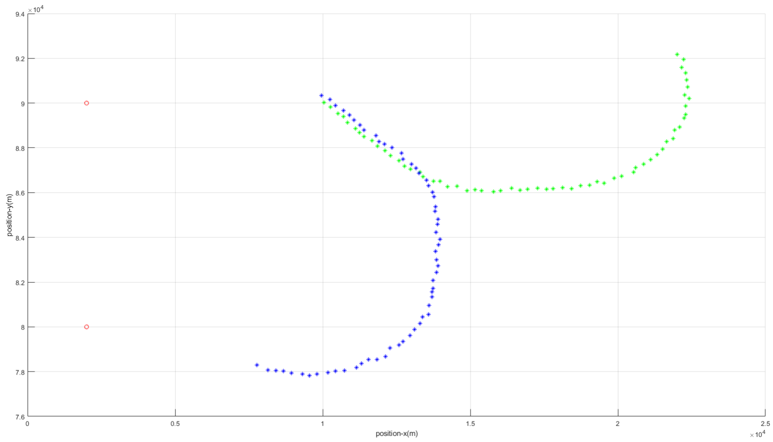

Figure 4.3: Results: Positions of Sensors and Targets

Both the sensors have a range resolution σ_R of 35m and a bearing resolution of σ_b of $2mrad$.

4.2.1 Results

To obtain the results we firstly transformed the tracks from NEWS coordinate system to Cartesian coordinates, along with position transformation we also transformed the resolutions of sensors into a Gaussian noise in Cartesian coordinates along respective dimensions. Once the tracks were obtained with their respective error they were compared with each other using fuzzy track-to-track association algorithm, the results are presented as following:

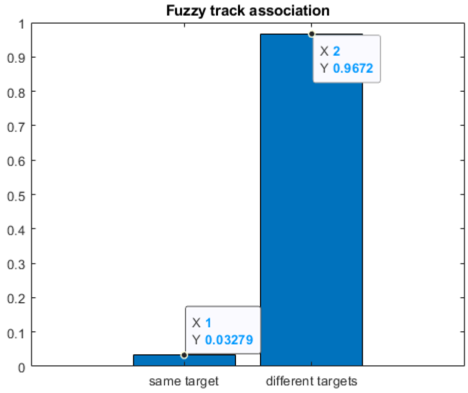

Figure 4.4: Results: Two Sensors Two Targets

Figure 4.4 depicts that the algorithm was fairly able to associate the two tracks as two different targets with an accuracy of 96.72 percent. The scenario was run for 1000 monte-carlo simulations and the accuracy results were averaged.

4.3 Scenario: 3D GCS Tracks with Two Sensors Two Targets

Previously we considered the performance of fuzzy track-to-track association algorithm with 2D tracks. A logical progression would be to test the performance of the algorithm with 3D tracks. Consider we have two targets reporting their in GCS coordinate system, the first target moves from GCS coordinates $(30.46°LAT, 68.79°LONG, 3048mALT)$ to $(33.23°LAT, 70.35°LONG, 3048mALT)$ with a speed of 300m/s, the second target move from $(30.46°LAT, 68.79°LONG, 3048mALT)$, the second target moves from GCS coordinates $(30.48°LAT, 68.74°LONG, 3109mALT)$ to $(33.25°LAT, 70.3°LONG, 3109mALT)$. Both the targets are following a straight trajectory and are moving in parallel. There are two sensors that are tracking these targets are both the targets are tracked by both the sensors. The sensors are working in an overlapping coverage region, the first track of sensor-1 has target-1 and the second track of sensor-1 is target-2, while the first track of sensor-2 is target-2 and the second track is target-1. The target tracks are depicted in Figure 4.5, the blue tracks are from sensor-1 and the red tracks are from sensor-2.

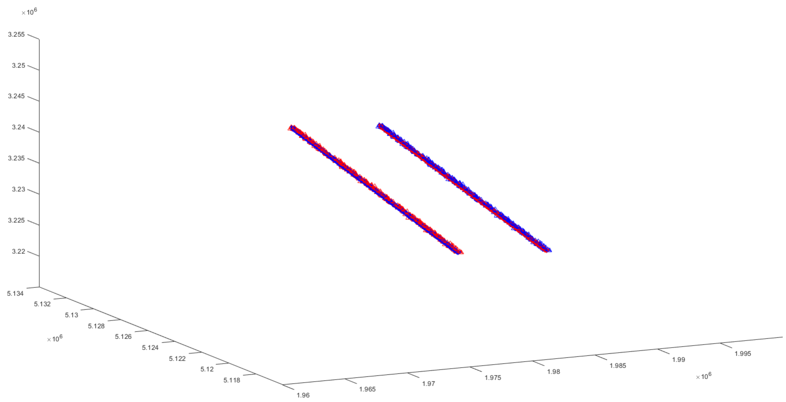

Figure 4.5: GCS Tracks of targets.

4.3.1 Results

Firstly the GCS tracks were transformed from GCS coordinates to ECEF coordinates. The resolutions of the sensors in ECEF coordinates were unavailable so we tried various resolutions in ECEF coordinates that correctly associated the targets. Histograms of various resolutions were plotted and the best performing resolutions were selected as the resolutions for algorithm. Afterwards the fuzzy track-to-track association algorithm was applied to the scenario to test the performance. The results are presented in following Figure 4.6. In the figure above S11 stands for sensor-1's first track S12 stand for sensor-1's second track and so on. The left most sub-plot in the figure shows a comparison between the first track from sensor-1 and first track from sensor-2 and depicts that they represent different targets. Similarly, other subplots show the different possible combinations.

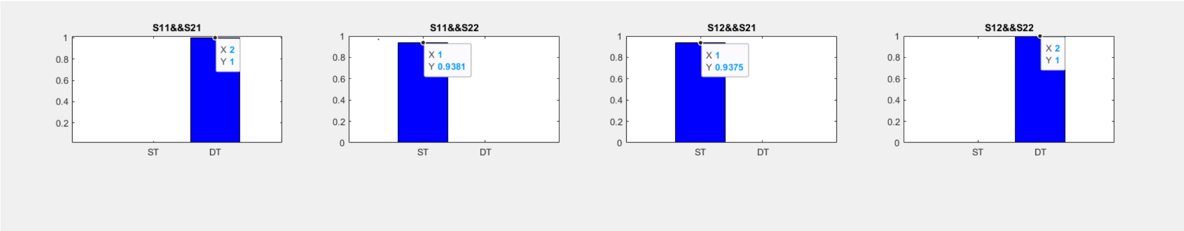

Figure 4.6: Two Sensors Four Tracks

figure 4.6 shows that the algorithm was able to associate the respective tracks of the two sensors with reasonable accuracy.

4.4 Scenario: 3D GCS Tracks with Eight Sensors Three Targets

In this scenario there are eight sensors in total working in an overlapping coverage region. There are three targets in the overlapping region of these sensors. Target-1 is being tracked by sensors-1,4,5,6,7 and 8, and moves from GCS coordinates $(30.46° LAT, 68.79° LONG, 3048 m ALT)$ to $(33.23° LAT, 70.35° LONG, 3048 m ALT)$ with a speed of 300m/s, this target is represented by multi-colored multiple-tracks (from various sensors) in Figure 4.7 below. Target-2 is being tracked by sensor-2 and moves from GCS coordinates $(30.48° LAT, 68.74° LONG, 3109 m ALT)$ to $(33.25° LAT, 70.3° LONG, 3109 m ALT)$ with a speed of 300m/s, the track of this target is presented with a blue track in Figure 4.7. Target-3 is being tracked by sensor-3 and moves from GCS coordinates $(30.47° LAT, 68.7° LONG, 3200 m ALT)$ to
$(33.24° LAT, 70.26° LONG, 3200 m ALT)$ with a speed of 300m/s, the track of this target is presented with a blue track in Figure 4.7. The GCS coordinates are converted to their ECEF equivalents for presentation purposes as following:

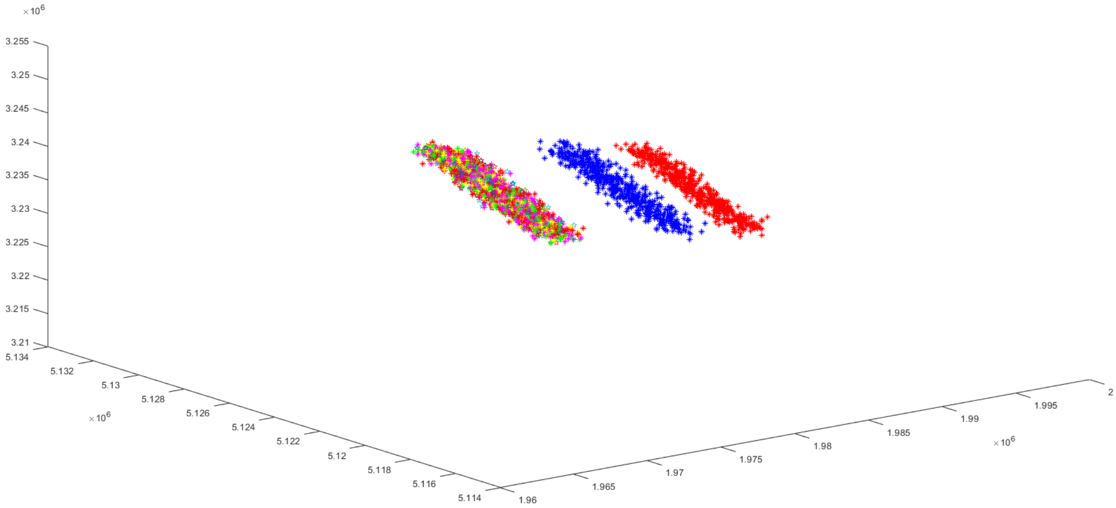

Figure 4.7: Tracks of the Three Targets

4.4.1 Results

The algorithm correctly confirms the ground truth scenario selected. The algorithm also provides the insights into the associations at a track-pair level; which track pair represents the same target and which track pair represents different targets. The algorithm also tells us about the the total number of targets present in the scenario along with the accuracy of the result. The results for the scenario discussed above are presented in Figures 4.8 and 4.9. In the figures above the Sen01Trk01 stands for the first track from sensor-1, Sen02Trk01 stands for the first track from sensor-2 and similarly the notation is followed for sensor track combinations. The first sub-plot on Figure 4.8 depicts that the first track reported from sensor-1 does not associate with first track from sensor-2, all the other sub-plots depicts various sensor-track association combinations. The last sub-plot on 4.9 depicts the cumulative number of targets in the scenario after verification from the association algorithm.

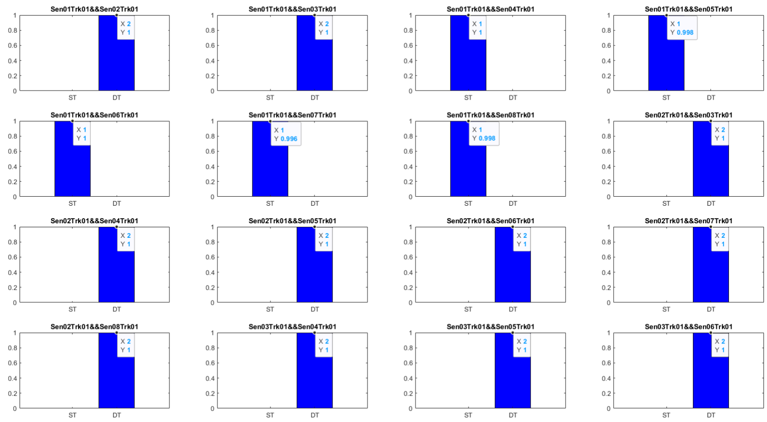

Figure 4.8: Eight Sensors Three Targets(1)

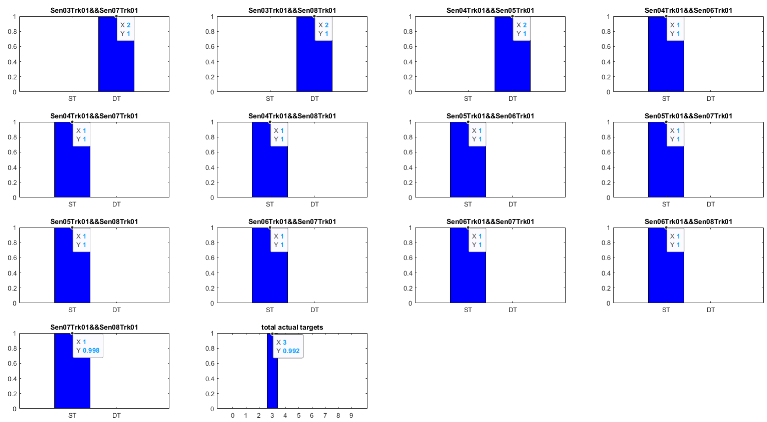

Figure 4.9: Eight Sensors Three Targets(2)

4.5 Scenario: Two Sensors with One Track Each in Gaussian uncertainty Environment

In previous two sections the tracks coming from sensors were considered to be noise-less. In real scenarios the track data is impacted due to various kinds of

uncertainties. To account for the impact of the uncertainties on fuzzy track-to-track association algorithm we introduce uncertainties in the tracks reported by the sensors.
Consider a scenario in which there are two air-borne sensors working in an overlapping coverage region. Sensor-1 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.35° LAT, 70.3° LONG, 8001 m ALT)$ to $(31.35° LAT, 74.24286° LONG, 7781 m ALT)$, the heading of the sensor is 90° from the north. Sensor-2 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.5° LAT, 70° LONG, 8003 m ALT)$ to $(31.5° LAT, 73.941421° LONG, 8354 m ALT)$, the heading of the sensor is 90° from the north. There is a single target in the overlapping region of the sensors which accelerates and decelerates during the scenario, with the minimum speed of 98m/s and a maximum of 114m/s. The target moves from GCS coordinates $(31.2° LAT, 72.37° LONG, 10298 m ALT)$ to $(31.2° LAT, 74.62228° LONG, 102943 m ALT)$, the heading of the target was 90° due-north. To make the scenario more realistic an error was induced in the target tracks coming from both sensors using the model presented in table of Figure 3.6. The tracks of the target from respective sensors are presented in Figures 4.10 and 4.11. The scenario is run for 2740 seconds with an update rate of 1s for each sensor.

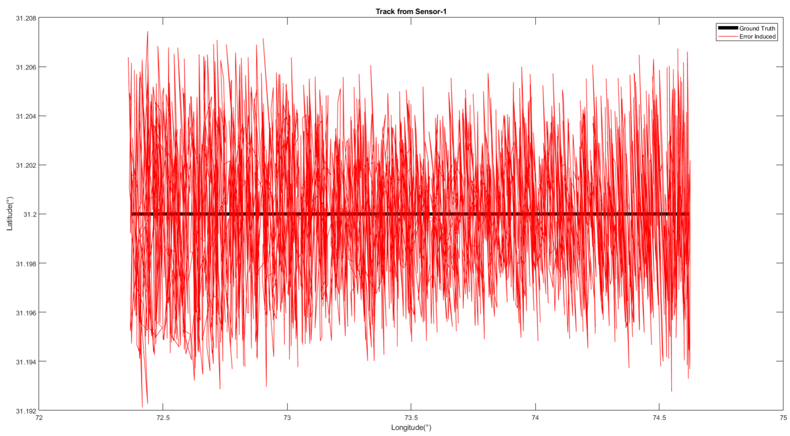

Figure 4.10: Target Track with Ground Truth and Noise from Sensor-1

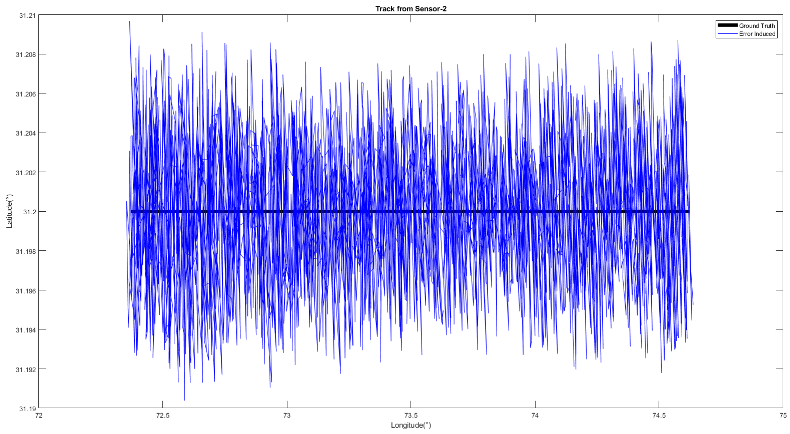

Figure 4.11: Target Track with Ground Truth and Noise from Sensor-2

4.5.1 Results

The resolutions for the fuzzy track-to-track association algorithm were computed based on the relative distance between the sensors and the target. The resolutions for both the sensors were kept identical and results were obtained as shown in the following figure 4.12.

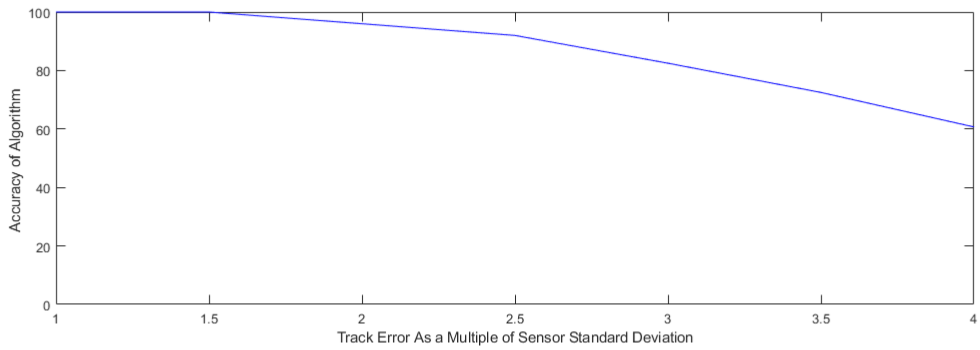

Figure 4.12: Accuracy of Algorithm under the influence of increasing Track noise

The aforementioned figure shows that the accuracy of the algorithm is very good when the noise standard deviation in the tracks is less than 1.5 times the respective standard deviations of the sensors. When the noise standard deviation in the tracks is increased the accuracy of the algorithm keeps on deteriorating.

4.6 Scenario: Three Sensors with a Track Each in Gaussian uncertainty Environment

Consider a scenario in which there are three air-borne sensors working in an overlapping coverage region. Sensor-1 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.35° LAT, 70.3° LONG, 8001 mALT)$ to $(31.35° LAT, 74.24286° LONG, 7781 mALT)$, the heading of the sensor is 90° from the north. Sensor-2 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.5° LAT, 70° LONG, 8003 mALT)$ to $(31.5° LAT, 73.941421° LONG, 8354 mALT)$, the heading of the sensor is 90° from the north. Sensor-3 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.2° LAT, 70° LONG, 7996 mALT)$ to $(31.2° LAT, 74.94286° LONG, 7865 mALT)$, the heading of the sensor is 90° from the north. There is a single target in the overlapping region of the sensors 1 and 2 which accelerates and decelerates during the scenario, with the minimum speed of 98m/s and a maximum of 114m/s. The target moves from GCS coordinates $(31.2° LAT, 72.37° LONG, 10298 mALT)$ to $(31.2° LAT, 74.62228° LONG, 102943 mALT)$, the heading of the target was 90° due-north. There is another target in the coverage region of sensor-3 which accelerates and decelerates between 103m/s and 117m/s. The target moves from GCS coordinates $(30.90009° LAT, 72.370178° LONG, 10343 mALT)$ to $(30.900009° LAT, 74.622298° LONG, 10156 mALT)$, the heading of the target was 90° due-north. To make the scenario more realistic an error was induced in the target tracks coming from all the sensors using the model presented in table of Figure 3.6. The tracks of the target from respective sensors along with target ground-truths, with bold black lines in the background being ground-truths and the colored random lines being erroneous tracks from respective sensors, are presented in Figure 4.13. The scenario was simulated for a total of 2740s with the common update rate of 1s for each sensor.

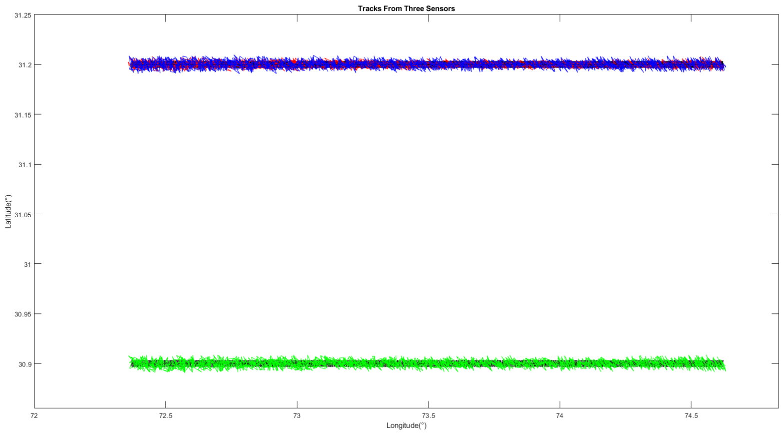

Figure 4.13: Ground Truths of Targets along with Sensor Tracks

4.6.1 Results

The resolutions for the fuzzy track-to-track association algorithm were computed based on the relative distance between the sensors and the targets. The resolutions for all the sensors were kept identical and results were obtained as shown in the following Figure 4.14.

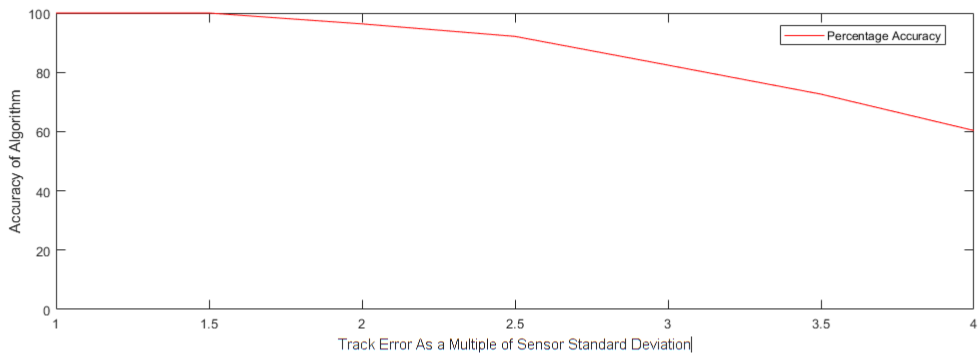

Figure 4.14: Accuracy of Algorithm under the influence of increasing Track noise

The aforementioned figure shows that the accuracy of the algorithm is very good when the noise standard deviation in the tracks is less than 1.5 times the respective standard deviations of the sensors. When the noise standard deviation in the tracks is increased the accuracy of the algorithm keeps on deteriorating.

4.7 Scenario: Four Sensors with a Track Each with different update rates in Gaussian uncertainty Environment

Consider a scenario in which there are four air-borne sensors working in an overlapping coverage region. Sensor-1 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.35°LAT, 70.3°LONG, 8001mALT)$ to $(31.35°LAT, 74.24286°LONG, 7781mALT)$, the heading of the sensor is 90° from the north. Sensor-2 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.5°LAT, 70°LONG, 8003mALT)$ to $(31.5°LAT, 73.941421°LONG, 8354mALT)$, the heading of the sensor is 90° from the north. Sensor-3 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.2°LAT, 70°LONG, 7996mALT)$ to $(31.2°LAT, 74.94286°LONG, 7865mALT)$, the heading of the sensor is 90° from the north. Sensor-4 is moving at a speed of 181m/s, it moves from GCS coordinates $(31.2°LAT, 70°LONG, 7996mALT)$ to $(31.2°LAT, 74.94286°LONG, 7275mALT)$, the heading of the sensor is 90° from the north. There is a single target in the overlapping region of the sensors 12 which accelerates and decelerates during the scenario, with the minimum speed of 98m/s and a maximum of 114m/s. The target moves from GCS coordinates $(31.2°LAT, 72.37°LONG, 10298mALT)$ to $(31.2°LAT, 74.62228°LONG, 102943mALT)$, the heading of the target was 90° due-north, the heading is changing slightly due to noise over the complete scenario. There is another target in the coverage region of sensors 34 which accelerates and decelerates between 103m/s and 117m/s. The target moves from GCS coordinates $(30.90009°LAT, 72.370178°LONG, 10343mALT)$ to $(30.900009°LAT, 74.622298°LONG, 10156mALT)$, the heading of the target was 90° due-north with heading changing slightly due to noise over the complete scenario. The scenario is working under the paradigm of time division multiple access, each sensor has its own designated time slot for sharing its track. The scenario starts with sensor-1 sharing its track data at 200ms after that every sensor shares its track with a difference of 200ms. The update rate of each sensor is 800ms. To make the scenario more realistic an error was induced in the target tracks coming from all the sensors using the model presented in table of Figure 3.6. The tracks of the target from respective sensors along with target ground-truths, With bold black lines in the background being ground-truths and the colored random lines being erroneous tracks from respective sensors, are presented in following Figure 4.15. The scenario was simulated for a total of 685s with the update rate of 800ms for each sensor.

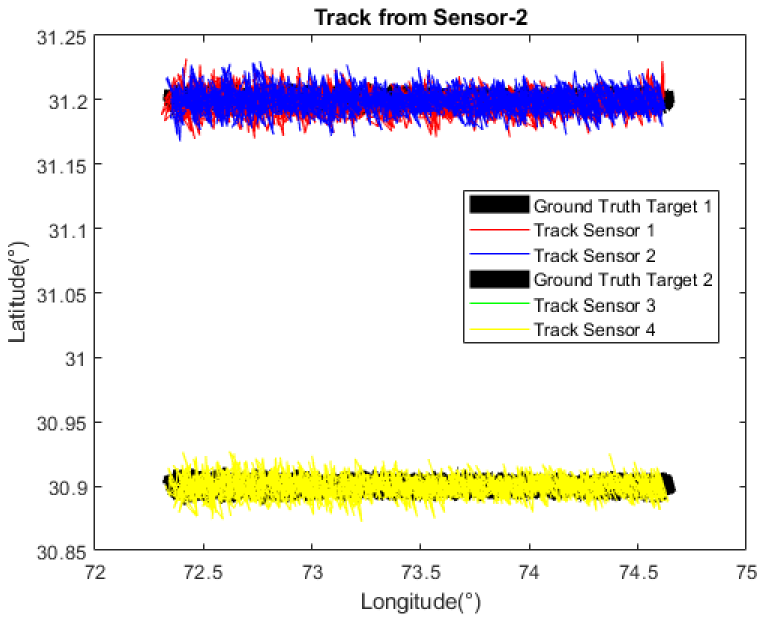

Figure 4.15: Target Tracks

4.7.0.1 Results: Applied linear prediction for Time Synchronization

To ascertain the performance of fuzzy track-to-track association in scenarios where the sensors have different update rates and their tracks are asynchronous a simple linear predictor was implemented as described in section 3.6. The results obtained are presented in following Figure 4.16.

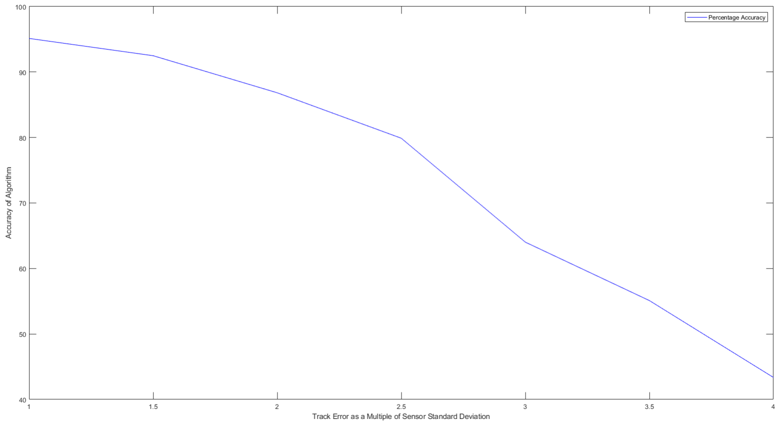

Figure 4.16: Accuracy of Algorithm under TDMA Paradigm

In the figure given above, one can clearly observe that the accuracy of the algorithm is significantly dropped when the tracks from sensors are asynchronous in contrast to the cases where the tracks were synchronized.

4.7.0.2 Results: Applied Batch Processing

In previous section we introduced time asynchronous tracks, they were synchronized using a linear time predictor which deteriorated the performance of the algorithm. The reason behind decrease in accuracy was that we were associating the tracks by only considering the track data of a single instant. The accuracy of the algorithm can be increased by applying batch processing technique to the algorithm. In this technique the association decision can be taken based on various instants of the data. Data windows of various sizes were employed to test the performance of the algorithm, the results are presented as following:

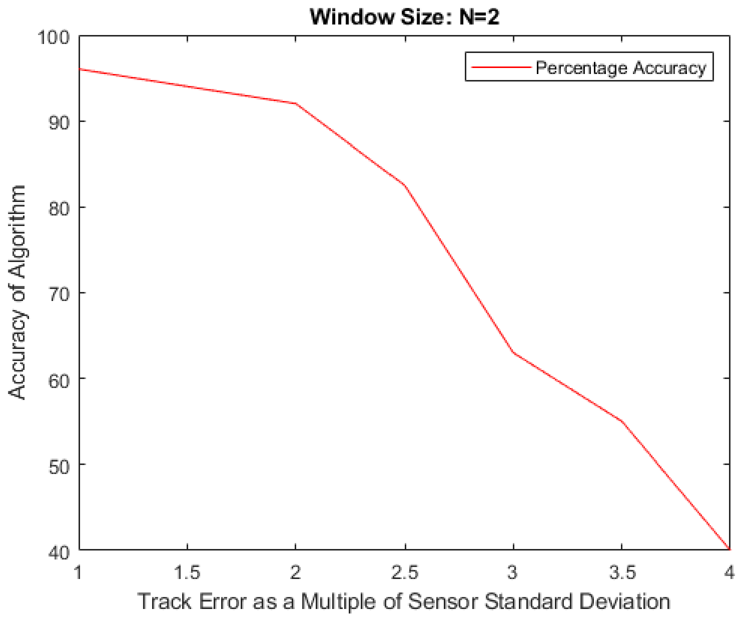

Figure 4.17: Accuracy of Algorithm with window size of two

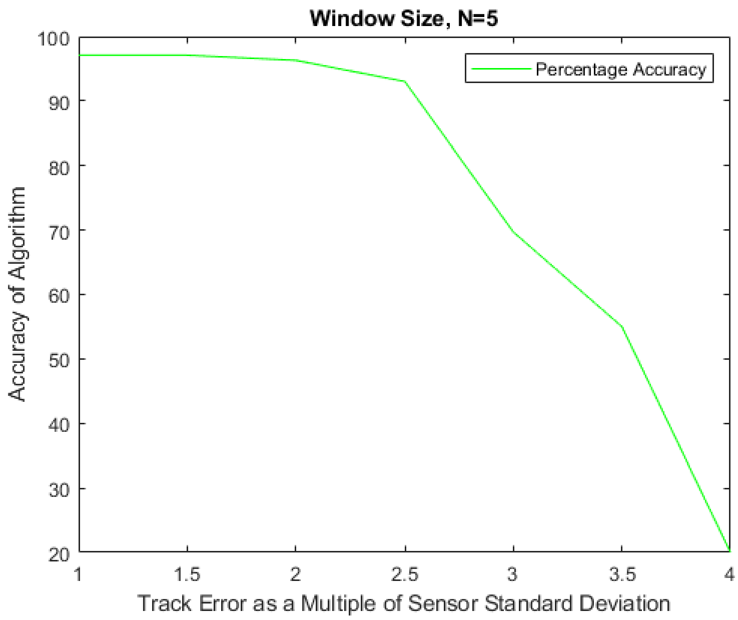

Figure 4.18: Accuracy of Algorithm with window size of five

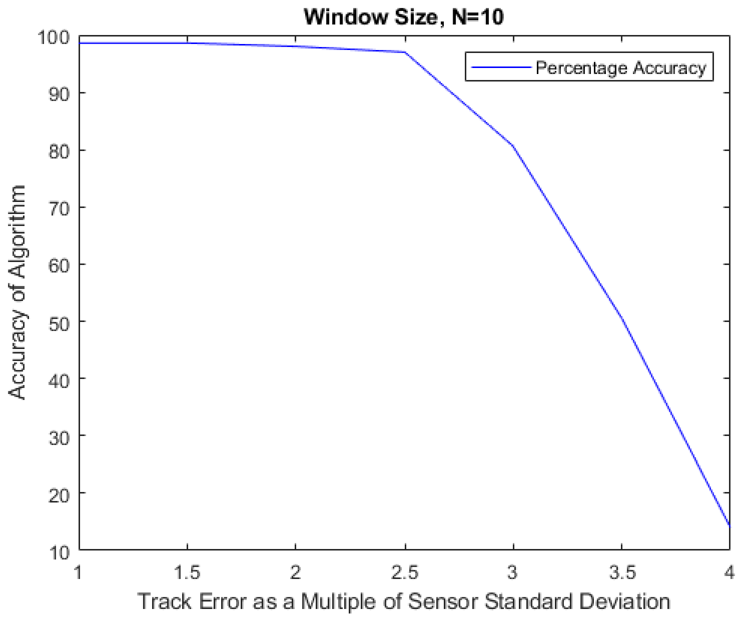

Figure 4.19: Accuracy of Algorithm with window size of ten

From the figures presented above one can observe that batch processing improves the performance of algorithm when the noise in tracks is low. When the noise in tracks is high the performance of batch processing is poor.

4.7.0.3 Results: Applied Speed and Heading Filters

Speed and heading filters were applied to increase the time efficiency of the algorithm. When the targets are different their speed and headings are also different, so we can use speed and heading filters to separate out different targets without going into the algorithm. This significantly improves the efficiency of the algorithm as the complexity of the scenarios grow. The time complexity for 1000 monte-carlo simulations of the scenario mentioned above is presented as following.

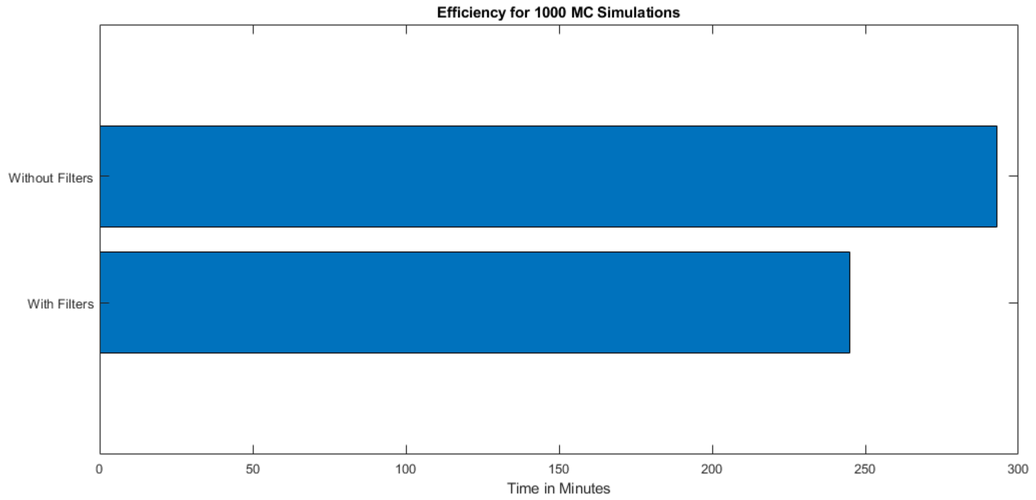

Figure 4.20: Efficiency of the Algorithms with Speed and Heading Filters

It is clearly visible that speed and heading filters make the algorithm more efficient. The simulations were performed on a desktop with specification as presented in Figure 4.21 below.

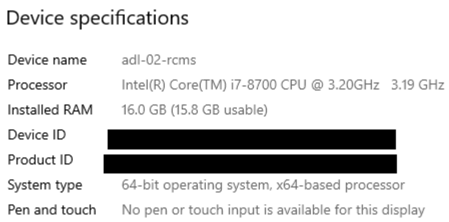

Figure 4.21: Machine Specifications

Chapter 5

Conclusion And Future Work

5.1 Conclusion

Fuzzy track to track association algorithm could prove to be a very beneficial algorithm for sensors that are working on geodetic coordinate system. It can also prove to be very efficient algorithm for sensor that are not identical. The performance of algorithm was very good after applying time alignment to the tracks, it performed significantly even under the error spread due linear prediction of track data. It was also shown that the performance of fuzzy track to track association algorithm with batch processing is excellent when the errors in the track data are low, the performance deteriorates when the errors in track data are high. The speed and heading filters applied along with the fuzzy track to track association algorithm proved to be a good choice as they improved the efficiency of the algorithm as compared to stand alone algorithm.
The system model presented in the thesis was a significant achievement as near real world testing of the algorithm was possible in simulation due to it. Due to the system it was also proven that fuzzy track to track association algorithm can perform significantly even the noise in the tracks is increased considerably.

5.2 Future Work

The improvements planned in future for this research work are as following:

- Implementation of the algorithm in a hardware compatible language.
- Testing the algorithm in real time distributed network.
- Implementation and analysis of the algorithm for a very large number of sensors and targets.

Bibliography

[1] Umair Ahmed. Modeling of uncertainty measurements of radar data in airborne applications for track-to-track data association algorithm. Master's thesis, 2017.

[2] Ashraf M Aziz. Fuzzy track-to-track association and track fusion approach in distributed multisensor–multitarget multiple-attribute environment. *Signal Processing*, 87(6):1474–1492, 2007.

[3] Yaakov Bar-Shalom and Huimin Chen. Multisensor track-to-track association for tracks with dependent errors. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 3, pages 2674–2679. IEEE, 2004.

[4] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs Storrs, CT, 1995.

[5] Samuel Blackman and Robert Popoli. Design and analysis of modern tracking systems(book). *Norwood, MA: Artech House, 1999.*, 1999.

[6] Wei Du, Huansheng Ning, Yuan Wei, and Jun Wang. Fuzzy double-threshold track association algorithm using adaptive threshold in distributed multisensor-multitarget tracking systems. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 1133–1137. IEEE, 2013.

[7] Johan Engblom. Air target measurement accuracy in the presence of ground clutter. Master's thesis, 2009.

[8] Ponciano Jorge Escamilla-Ambrosio and N Lieven. A multiple-sensor multiple-target tracking approach for the autotaxi system. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 601–606. IEEE, 2004.

[9] Shuaixin Hong, Dongliang Peng, and Yifang Shi. Track-to-track association using fuzzy membership function and clustering for distributed information fusion. In *2018 37th Chinese Control Conference (CCC)*, pages 4028–4032. IEEE, 2018.

[10] Hsueh-JyhLi and Yean-WoeiKiang. Radar and inverse scattering. https://doi.org/10.1016/B978-012170960-0/50047-5.

[11] Zhijian Hu, Henry Leung, and Martin Blanchette. Statistical performance analysis of track initiation techniques. *IEEE Transactions on Signal Processing*, 45(2):445–456, 1997.

[12] Mahmoud Yahia Hussein and Abdelrasoul Jabar Alzubaidi. Target tracking radar. *Target*, 4(05), 2014.

[13] Jonas Hyllengren. Clustering for multi-target tracking, 2017.

[14] Syed A Imtiaz, Kallol Roy, Biao Huang, Sirish L Shah, and Phanindra Jampana. Estimation of states of nonlinear systems using a particle filter. In *2006 IEEE International Conference on Industrial Technology*, pages 2432–2437. IEEE, 2006.

[15] Allen J Kanyuck and Robert A Singer. Correlation of multiple-site track data. *IEEE Transactions on aerospace and electronic systems*, (2):180–187, 1970.

[16] Lance M Kaplan, Yaakov Bar-Shalom, and William D Blair. Assignment costs for multiple sensor track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems*, 44(2):655–677, 2008.

[17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1):7–21, 2005.

[18] Martin E Liggins, Chee-Yee Chong, Ivan Kadar, Mark G Alford, Vincent Vannicola, and Stelios Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, 1997.

[19] Zhao Lingling, Dong Xianglei, Ma Peijun, Su Xiaohong, and Shi Chunmei. A new multi-sensor track association approach based on intuitionistic fuzzy clustering. In *International Conference on Advances in Information Technology*, pages 256–266. Springer, 2013.

[20] Ronald Mahler. Measurement-to-track association for nontraditional measurements. In *14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.

[21] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[22] NCSS. *NCSS Statistical Software*. URL: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Fuzzy_Clustering.pdf.

[23] Hairong Qi, S Sitharama Iyengar, and Krishnendu Chakrabarty. Distributed sensor networks—a review of recent research. *Journal of the Franklin Institute*, 338(6):655–668, 2001.

[24] Matthew B Rhudy, Roger A Salguero, and Keaton Holappa. A kalman filtering tutorial for undergraduate students. *International Journal of Computer Science & Engineering Survey*, 8(1):1–9, 2017.

[25] Peter Rohal and Jan Ochodnicky. Radar target tracking by kalman and particle filter. In *2017 Communication and Information Technologies (KIT)*, pages 1–4. IEEE, 2017.

[26] Kenshi Saho and Masao Masugi. α–β–γ tracking filters using acceleration measurements. *SpringerPlus*, 5(1):309, 2016.

[27] Merrill Ivan Skolnik et al. *Introduction to radar systems*, volume 3. McGraw-hill New York, 1980.

[28] Anca Stoica, Diana Militaru, Dan Moldoveanu, and Alina Popa. Tactical data link-from link-1 to link-22. *Sci. Bull. Mircea cel Batran Naval Acad*, 19(2):346, 2016.

[29] Gabriel A Terejanu et al. Extended kalman filter tutorial. *University at Buffalo*, 2008.

[30] Aybars Tokta and Ali Koksal Hocaoglu. A fast track to track association algorithm by sequence processing of target states. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.

[31] Aybars Tokta and Ali Koksal Hocaoglu. A track to track association algorithm based on weighted state correlation similarity. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–4. IEEE, 2018.

[32] Liu Yu, He You, Dong Kai, and Wang Haipeng. Fuzzy binary track correlation algorithms for multisensor information fusion. *Journal of Networks*, 9(3):672, 2014.

[33] Ying Zhang, Henry Leung, Martin Blanchette, Titus Lo, and John Litva. An efficient decentralized multiradar multitarget tracker for air surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1357–1363, 1997.