

# Study of Data-Association Algorithms for Object Tracking by Automotive Radar

by

Farah Jabin

A thesis submitted in partial fulfillment for the  
degree of Master of Technology

under supervision of

Dr. Shobha Sundar Ram

Department of Electronics and Communication Engineering  
Indraprastha Institute of Information Technology, Delhi

August 2018



# Study of Data-Association Algorithms for Object Tracking by Automotive Radar

by

Farah Jabin

A thesis submitted in partial fulfillment for the  
degree of Master of Technology

to

Indraprastha Institute of Information Technology, Delhi

August 2018

# Certificate

This is to certify that the thesis titled "Study of Data-Association Algorithm for Object Tracking by Automotive Radar" being submitted by Farah Jabin (Roll No.- MT16091) to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original work carried out by her under my supervision. In my opinion, thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. The results contained in the thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Date: \_\_\_\_\_

Dr. Shobha Sundar Ram  
Assistant Professor  
Department of Electronics and Communication Engineering  
Indraprastha Institute of Information Technology, Delhi  
New Delhi 110020

Dr. Anirban Roy  
Lead Technical Architect  
ADAS development  
Continental AG  
Bengaluru 560100

# *Abstract*

Multiple-target tracking (MTT) is one of the key aspects of driver assistance systems, and has been subject to considerable research. One critical part of an MTT system is solving the data association problem, which is associating correct observations to the existing tracks. Several data association algorithms have been proposed to solve this problem. The multiple hypothesis tracker (MHT) is a well-studied and currently the preferred method of data association in MTT application. MHT maintains several possible data association hypotheses or solutions and uses new observations to eliminate unlikely hypotheses over time. Despite being regarded as the most prominent data association method, MHT implementations remains a challenge because of its computational complexity. In this thesis, we have described an efficient method for reducing the computational requirement of MHT in automotive applications.

This work also presents four most commonly used association algorithms. First, the two types of single hypothesis trackers are introduced: sub-optimal nearest neighbour or SNN and global nearest neighbour or GNN. Then the all-neighbour data association methods: probabilistic data association (PDA) and joint probabilistic data association (JPDA), and their extensions are presented.

The algorithms are implemented and tested for four different real case road scenarios, in presence of clutter and missed detection, to check which tracker is more suitable to track which traffic situation. The optimal sub-pattern assignment (OSPA) metric is used to quantify and compare the performance of the association algorithms.

## *Acknowledgements*

I would like to express my sincere gratitude to my supervisor Dr. Anirban Roy and Dr. Shobha Sundar Ram for their guidance, coaching and support for this thesis.

I would also like to thank Dr. Anshu Gupta for the opportunity to join Continental as intern and for his insightful comments and encouragements. Special thanks to all the other interns and team members of the ADAS team, without their passionate participation and input, this work could not have been successfully conducted.

This work is dedicated to my parents for their teachings and values passed on to me. I thank their understanding even in those moments in which my immaturity prevailed. To my sisters for all their support and friendship throughout these years.

# Contents

<b>Certificate</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Objective . . . . .	3
1.3 Thesis Outline . . . . .	3
<b>2 Kalman Filtering</b>	<b>5</b>
2.1 Linear Kalman Filter . . . . .	6
2.2 Extended Kalman Filter . . . . .	8
2.3 Target Motion and Measurement Model . . . . .	10
2.3.1 Constant Velocity Model . . . . .	10
2.3.2 Coordinated Turn Rate and Velocity Model . . . . .	11
2.4 Clutter Model . . . . .	12
<b>3 Data Association</b>	<b>14</b>
3.1 Track Score . . . . .	16
3.2 Gating . . . . .	17
3.3 Single Hypothesis Tracking . . . . .	18
3.4 All Neighbour Data Association Approach . . . . .	21
3.4.1 Probabilistic Data Association . . . . .	21
3.4.2 Joint Probabilistic Data Association . . . . .	22
3.4.3 PDA Extensions . . . . .	24
<b>4 Multiple Hypothesis Tracker</b>	<b>25</b>
4.1 MHT Algorithm Elements . . . . .	27
4.1.1 Track Formation . . . . .	27

---

4.1.2	Track-Level Pruning and Confirmation . . . . .	27
4.1.3	Confirmation Matrix . . . . .	29
4.1.4	Hypothesis Generation and Pruning . . . . .	29
4.1.5	Global Level Pruning . . . . .	30
4.1.6	N-Scan Pruning . . . . .	31
4.2	Presentation of MHT Data . . . . .	32
4.2.1	Primary Track File . . . . .	32
4.2.2	Global Track File . . . . .	33
<b>5</b>	<b>Simulation Results</b>	<b>34</b>
5.1	OSPA Performance Metric . . . . .	34
5.2	Results . . . . .	35
5.2.1	Scenario 1: Three Sparse Targets . . . . .	35
5.2.2	Scenario 2: Three Crossing Targets . . . . .	37
5.2.3	Scenario 3: Three Parallel Targets . . . . .	39
5.2.4	Scenario 4: Two Maneuvering Targets . . . . .	40
<b>6</b>	<b>Conclusion</b>	<b>44</b>
6.1	Future Work . . . . .	45
	<b>Bibliography</b>	<b>46</b>

# List of Figures

1.1	Top view of an Advanced Driver Assistance System. [1]	1
2.1	Application of the Kalman Filter	5
2.2	Prediction and filtering cycle of the Kalman Filter	8
2.3	Prediction and filtering cycle of Extended Kalman Filter.	10
3.1	Flow of a conventional Data Association algorithm.	14
3.2	Types of data association algorithms.	15
3.3	Illustration of the gating process.	18
3.4	Association conflict situation.	20
3.5	Example of association conflict for JPDA.	23
4.1	Association example for TOMHT track tree formation.	25
4.2	Tree structure for example in figure 4.1.	26
4.3	Flowchart of track-oriented MHT logic.	28
4.4	Example of N-Scan pruning.	31
5.1	True and estimated trajectories of scenario 1.	36
5.2	OSPA distance for scenario 1.	36
5.3	OSPA localization component for scenario 1.	37
5.4	OSPA cardinality component for scenario 1.	37
5.5	True and estimated trajectories of scenario 2.	38
5.6	OSPA distance for scenario 2.	38
5.7	OSPA localization component for scenario 2.	38
5.8	OSPA cardinality component for scenario 2.	39
5.9	True and estimated trajectories of scenario 3 obtained by PDA.	39
5.10	True and estimated trajectories of scenario 3 obtained by MHT.	40
5.11	OSPA distance for scenario 3.	40
5.12	OSPA localization component for scenario 3.	40
5.13	OSPA cardinality component for scenario 3.	41
5.14	True and estimated trajectories of scenario 4 obtained by PDA.	41
5.15	True and estimated trajectories of scenario 4 obtained by MHT.	42
5.16	OSPA distance for scenario 4.	42
5.17	OSPA localization component for scenario 4.	42
5.18	OSPA cardinality component for scenario 4.	43

# List of Tables

3.1	TMCR matrix for example shown in figure 3.3 . . . . .	18
3.2	TMCR matrix for gating situation shown in figure 3.4 . . . . .	19
3.3	Hypothesis matrix for example shown in figure 3.5 . . . . .	24
4.1	Compatibility matrix for tracks at scan $(k + 1)$ in figure 4.2 . . . . .	29
4.2	Primary track file structure. . . . .	33
4.3	Global track file structure. . . . .	33
5.1	Computation time of MHT for scenario 1. . . . .	37

*Dedicated to my beloved family...*

# Chapter 1

## Introduction

Advanced Driver Assistance System or ADAS is a collection of safety systems (sensors and signal processing algorithms) that work together to increase road safety and help the driver as shown in figure 1.1. Some of the features include: adaptive cruise control (ACC), collision avoidance system, proximity monitor, automatic parking, pedestrian monitor and lane keep assist. Many of these features heavily rely on accurate estimation of the car's surroundings. Target tracking allows the system to detect obstacles, other vehicles and pedestrians in the car's surroundings and estimate their position and speed. The objective of target tracking is to collect data from sensor's field of view (FOV) and then to segregate that data into sets of observations originating from targets of interest and those originating from background clutter. There are two target tracking frameworks - single target tracking (STT) and multiple target tracking (MTT) [2]. In STT, it is assumed that there is only one target of interest amid clutter. The tracker only has to decide whether an observation has originated from the target or the background noise

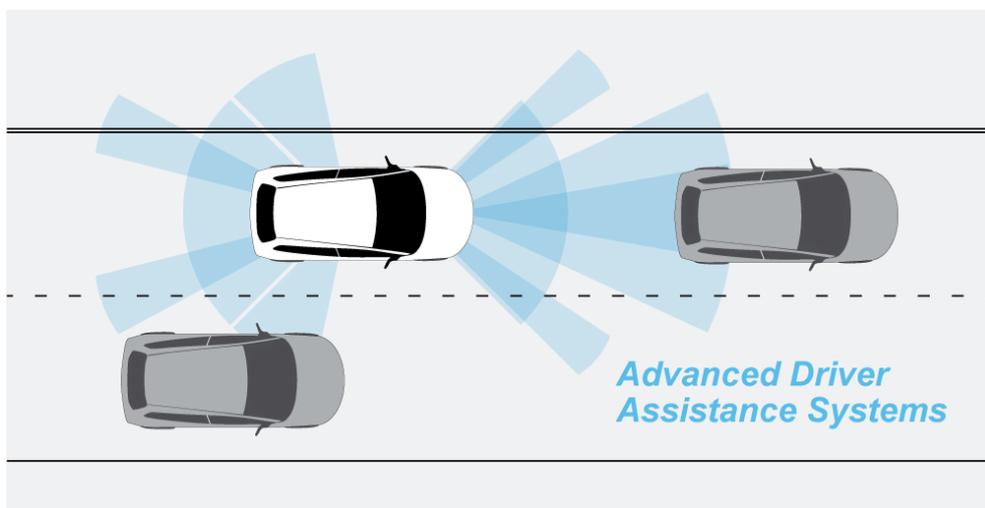


FIGURE 1.1: Top view of an Advanced Driver Assistance System. [1]

or clutter. While an MTT system involves a data association logic in order to decide from which target, if any, a particular observation originated [3]. Once a list of potential tracks is formed by estimating the number of targets, the kinematic parameters for each track can be computed [4]. MTT is more complex due to the following reasons:

1. **Unknown number of targets:** The number of true targets in the vicinity is not known in an MTT system. This ambiguity is also due to the presence of clutter, false alarms and measurement loss in the sensor.
2. **Association uncertainty:** The identity of the origin of each observation is unknown, i.e it is not clear which measurement is to be associated with which target.
3. **Overlapping ambiguity:** In case of multiple targets in close vicinity there could be an overlap of ambiguities with common measurements shared by more than one track.

There are many data association algorithms available in literature. The simplest approach is the nearest neighbour method, also known as the single hypothesis tracker. It selects and propagates the single most likely association solution at each step. It has two variants: sub optimal nearest neighbor (SNN) and global nearest neighbor (GNN). Another set of association algorithms consider the probability of the data association of the concerned observation with all the neighbours - the probabilistic data association (PDA), joint probabilistic data association (JPDA) and their extensions [5], [6]. They form multiple hypotheses at each scan and combine them into one, weighted by their association probabilities [7]. Multiple hypothesis tracker (MHT) is a deferred decision logic that maintains and propagates several data association (track-to-measurement association) solutions before making a firm decision, assuming that the track-to-measurement conflict will resolve as new data is received.

## 1.1 Motivation

A critical part of target tracking in ADAS is associating the measurements to the correct targets when vehicles are closely spaced and rapidly changing their states. Commonly used association algorithms like GNN and PDA do not perform well in case of heavy clutter and close targets and suffer from the problems of track loss and track coalescence. To address these problems, several extensions of GNN and PDA were developed like JPDA, NN-PDA, kNN-JPDA and MHT. Among these methods, MHT is shown to provide improved performance in MTT systems. There are two basic types of MHT frameworks: hypothesis-oriented MHT or HOMHT [8] and track-oriented MHT or TOMHT. Out of

the two, TOMHT is more favoured currently. It typically maintains a set of potential tracks instead of hypotheses and propagates most likely tracks using a track tree structure.

Although TOMHT is the most preferred multiple-target tracking algorithm, it is also the most complex association algorithm to implement and suffers from potential combinatoric explosion of number of tracks and hypotheses that can be formed at each time step. Hence, MHT requires extensive computational resources and is rarely used in ADAS applications.

## 1.2 Objective

The objective of this thesis is to implement a computationally feasible TOMHT that uses a concept called inner gating to limit the number of tracks formed and reduce its computation time. The performance of the algorithm is compared with the performance of different data association algorithms for multiple object tracking in automotive applications. The tracking algorithms that were implemented are: SNN, GNN, PDA, NN-PDA and TOMHT. The trackers were tested for different MTT road scenarios in presence of measurement origin uncertainty and missed detections. The trackers are compared to decide which algorithm is most effective in tracking different road scenarios and traffic conditions. A performance metric called the optimal sub-pattern assignment or OSPA metric is used to evaluate the performance of the tracking methods.

## 1.3 Thesis Outline

This thesis is divided into 6 chapters and an overview of each chapter is given below:

- **Chapter 2: Kalman Filtering**

This chapter presents the kinematic state estimation problem in target tracking, which involves filtering and estimation of quantities like target position and velocity. Filtering is the step that incorporates the assigned measurements into the predicted tracks to update their state estimates.

- **Chapter 3: Data Association**

This chapter discusses different types of data association methods and their basic functions that include gating, track-to-measurement correlation matrix and track validity score and probability computation.

- **Chapter 4: Multiple Hypothesis Tracker**

Implementation of the proposed track-oriented MHT is presented in this chapter. The elements of this method are discussed in detail, that include track handling, hypothesis generation, track confirmation and deletion and user presentation logic.

- **Chapter 5: Simulation Result**

This chapter briefly describes the OSPA performance metric and shows the results of our comparative analysis.

- **Chapter 6: Conclusion**

This chapter concludes all the findings of this work and discusses about possible future work.

## Chapter 2

# Kalman Filtering

This chapter covers the state estimation problem using Kalman filters in tracking systems. State estimation refers to the continual prediction and estimation of kinematic states (position, velocity, acceleration etc.) of a moving target,  $x(t)$ . The job of an estimator is to estimate  $x(t)$ , by combining a set of noisy measurements  $y(t)$  with predictions from a previous time instant,  $x(t - 1)$ , such that the estimation error is minimized in some respect. The problem of state estimation has its roots in the concepts of least square estimation (LSE). LSE uses a batch of data to estimate the parameters that minimizes the sum of squared error [9]. Kalman filter can be derived from a recursive form of LSE [10]. The main difference between recursive Kalman filter and LSE is that the former includes process noise into the target motion model. Figure 2.1 depicts the application of Kalman filter in the state estimation problem. The output of a physical system (such as a moving target) is measured using sensors that convey some useful information about the kinematic behavior of the system. There is inherently some process noise associated with this mapping. The sensor observations also include uncertainties or measurement noise. The Kalman filter then estimates the unknown system states by optimizing a given condition.

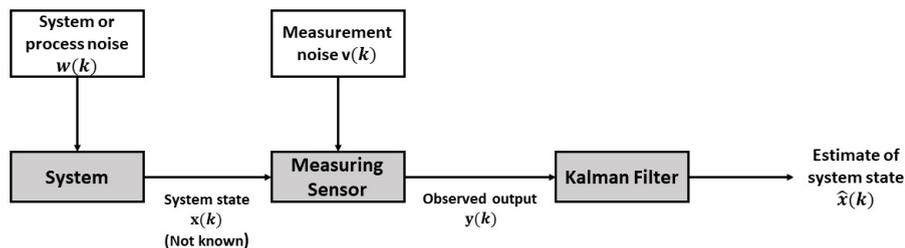


FIGURE 2.1: Application of the Kalman Filter

The general filtering problem can be formulated as follows [11]

$$x(k+1) = f(x(k), u(k), w(k)) \quad (2.1a)$$

$$y(k) = h(x(k), v(k)) \quad (2.1b)$$

Equation (2.1a) defines the system's dynamics, where  $x(k+1)$  is the state vector at time  $k+1$ , which is a function of the previous system states  $x(k)$ , control vector  $u$  and process noise  $w$ . Equation (2.1b) is the measurement model where  $y$  is the observation vector and  $v$  represents the sensor noise. The state transition and measurement functions,  $f$  and  $h$  respectively, along with noise characteristics are known to the filter. The job of Kalman filter is to obtain the best estimate of  $x(k+1)$  given a set of measurements,  $\{y(1), y(2), \dots, y(k)\}$ . The linear Kalman filter operates under linearity and Gaussian conditions on system dynamics, discussed in section 2.1. If either of the process model or measurement model is non-linear, extended Kalman filter (EKF) is used for state estimation, which is discussed in section 2.2. Section 2.3 presents the system design and problem formulation for different target tracking models.

## 2.1 Linear Kalman Filter

If we consider the system dynamics to be linear time-variant, equation (2.1a)-(2.1b) changes to

$$x_{k+1} = Fx_k + Gu_k + w_k \quad k > 0 \quad (2.2a)$$

$$y_k = Hx_k + v_k \quad (2.2b)$$

Let  $x$  be the  $N$ -dimensional state vector that is to be estimated,  $F$  is the known  $N \times N$  state transition matrix and  $H$  is the  $M \times N$  measurement matrix where  $M$  is the measurement dimension.  $w_k$  and  $v_k$  are uncorrelated, zero-mean, white, Gaussian noise sequences with known symmetric positive semi-definite covariance matrices;

$$E[w_k w_k^T] = Q_k \quad (2.3)$$

$$E[v_k v_k^T] = R_k \quad (2.4)$$

$$E[w_k v_k^T] = 0 \quad (2.5)$$

Let  $x_0$  be the initial state of the system, which is a Gaussian random vector with known mean and covariance matrix

$$E[x_0] = \hat{x}_{0|0} \quad \text{and} \quad E[(x_0 - \hat{x}_{0|0})(x_0 - \hat{x}_{0|0})^T] = P_{0|0} \quad (2.6)$$

Then to obtain the state estimate of system given by (2.2a)-(2.2b), the Kalman filter propagates the conditional probability density function  $p(x_k|Y^k)$ , which is Gaussian for all  $k$

$$p(x_k|Y^k) \sim \mathcal{N}(\hat{x}_{k|k}, P_{k|k}) \quad (2.7)$$

where  $Y^k = \{y_1, y_2, \dots, y_k\}$  denotes the sequence of measurements till time  $k$ . Conditional mean of the pdf (2.7),  $\hat{x}_{k|k}$  is the estimate of  $x_k$  and  $P_{k|k}$  is the covariance matrix that quantifies the estimation uncertainty. Therefore, Kalman filter evaluates the conditional pdf by propagating only its first and second moments [11]. This process is recursive which means that to evaluate the current state estimate  $\hat{x}_{k|k}$ , the filter only requires the new measurement  $y_k$  and the previous estimate  $\hat{x}_{k-1|k-1}$ . The transition from  $p(x_k|Y^k)$ , which is the previous estimate to  $p(x_{k+1}|Y^{k+1})$  new estimate is implemented in two steps:

- **Prediction or time update:** This step evaluates the pdf  $p(x_{k+1}|Y^k)$  which is the a-priori estimate of mean and covariance for the next step i.e. measurement update. This predicted estimate represents the best knowledge about the system at time  $k+1$  before the observation at that time instant is made [12].
- **Filtering or measurement update:** The filtering cycle calculates the posteriori estimate  $p(x_{k+1}|Y^{k+1})$ , using the a-priori estimate from the previous step and the new observation  $y(k+1)$ . It corrects the predicted state based on the new information at time  $k+1$ .

Given the system dynamics and initial conditions, the Kalman equations are given below. The first step is to compute the predicted state and error covariance matrix using equations given below

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k} + u_k \quad (2.8)$$

$$P_{k+1|k} = FP_{k|k}F^T + Q \quad (2.9)$$

Next,  $\nu_{k+1}$ , known as the innovation, is calculated as the difference between the new measurement  $y_{k+1}$  and predicted measurement  $H\hat{x}_{k|k}$  in (2.10). The innovation quantifies the new information contributed by observation at time  $k+1$ . Along with innovation, the Kalman gain  $K_{k+1}$ , given by equation (2.11) is computed. This is the relative weight given to predicted state and is automatically tuned by the filter. If measurement error

is high, Kalman gain reduces and the filter places more weight on predicted state. On the other hand, with a high gain filter gives more weightage to recent observation [13].

$$\nu_{k+1} = y_{k+1} - H\hat{x}_{k+1|k} \quad (2.10)$$

$$K_{k+1} = P_{k+1|k}H^T(HP_{k+1|k}H^T + R)^{-1} \quad (2.11)$$

The final step is to compute the current estimates using equation (2.12)-(2.13).

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\nu_{k+1} \quad (2.12)$$

$$P_{k+1|k+1} = (I - K_{k+1}H)P_{k+1|k} \quad (2.13)$$

After each prediction and filtering update, the cycle is repeated with the previous estimates used to project the new a-priori estimates. Figure 2.2 summarizes the complete sequential, recursive Kalman filter algorithm (2.8)-(2.9).

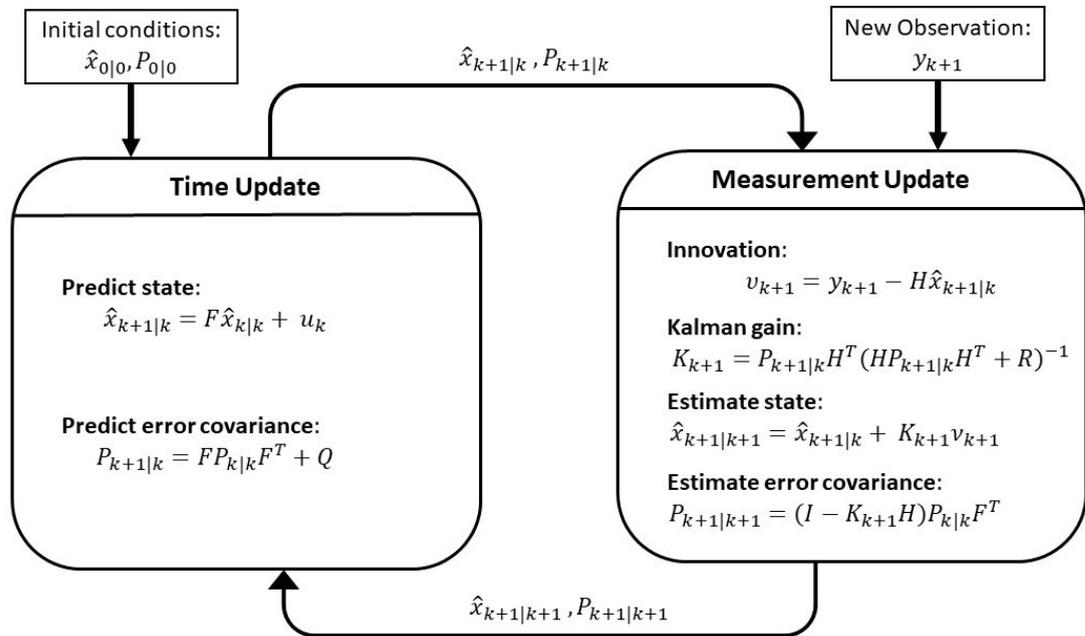


FIGURE 2.2: Prediction and filtering cycle of the Kalman Filter

## 2.2 Extended Kalman Filter

This section discusses the modified Kalman filter problem for the case when the target dynamics and/or measurement process is nonlinear. An extension of linear Kalman filter, known as the extended Kalman filter or EKF is used under these conditions.

Consider the following nonlinear state space model:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (2.14)$$

$$y_k = h(x_k) + v_k \quad (2.15)$$

where  $w_k$  and  $v_k$  are zero-mean, uncorrelated white Gaussian random noise sequences with known covariance  $Q$  and  $R$  respectively.  $f(x_k, u_k)$  is the nonlinear state transition matrix and  $h(x_k)$  is the nonlinear measurement matrix. Due to the nonlinear dynamics (2.14)-(2.15), the conditional pdf  $p(x_{k+1}|Y^k)$  and  $p(x_{k+1}|Y^{k+1})$  are no longer Gaussian. An optimal nonlinear filter in this case has to propagate the entire conditional pdf to compute its mean and covariance [11]. To make this task simpler, EKF approximates the nonlinear system dynamics to a linearized version. The linear approximation is done by taking the Jacobian of the state transition matrix  $f(x_k, u_k)$  and measurement matrix  $h(x_k)$  around the last estimate. One cycle of EKF is composed of the following step:

1. **Linearize the system model:** Calculate the Jacobian of state transition matrix around the previous estimate:

$$F_{k+1} = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k|k}} \quad (2.16)$$

2. **Prediction or time update:** Use the linearized system dynamic to get the predicted state and covariance:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) \quad (2.17)$$

$$P_{k+1|k} = F_{k+1}P_{k|k}F_{k+1}^T + Q \quad (2.18)$$

3. **Linearize the measurement model:** Calculate the Jacobian of measurement matrix around predicted estimate:

$$H_{k+1} = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k+1|k}} \quad (2.19)$$

4. **Filtering or measurement update:** Compute the Kalman gain, state and error covariance:

$$K_{k+1} = P_{k+1|k}H_{k+1}^T(H_{k+1}P_{k+1|k}H_{k+1}^T + R)^{-1} \quad (2.20)$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}[y_{k+1} - h(\hat{x}_{k+1|k})] \quad (2.21)$$

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k} \quad (2.22)$$

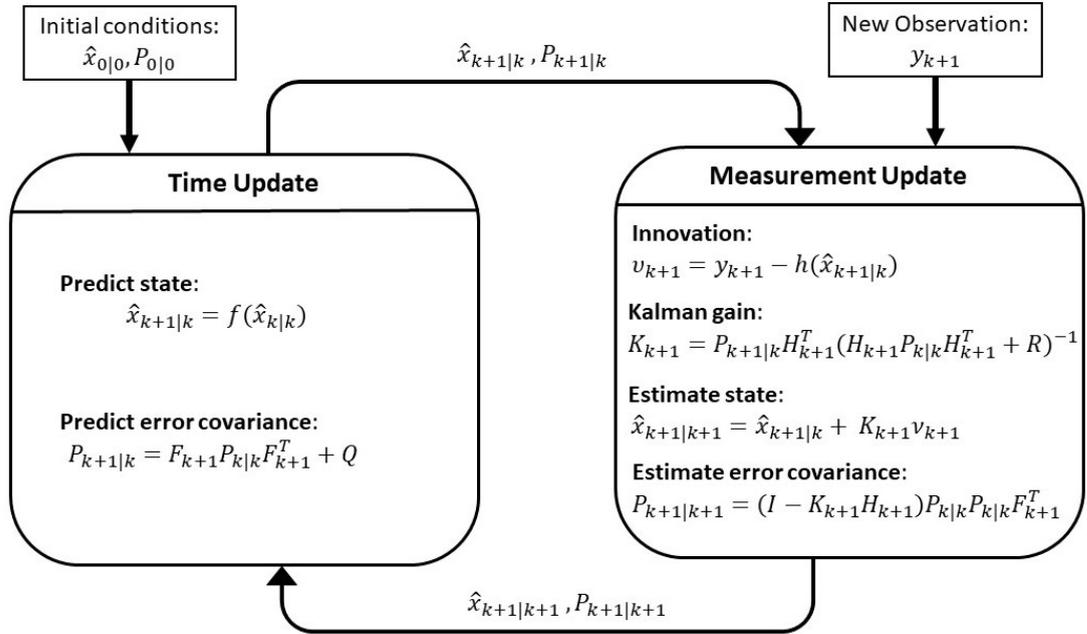


FIGURE 2.3: Prediction and filtering cycle of Extended Kalman Filter.

As seen from the above equations, the basic workings of EKF is like linear Kalman filter but with linear approximation. However, EKF is computationally more complex than its linear counterpart. Since it is based on approximations, the error covariance matrices  $P_{k+1|k}$  and  $P_{k+1|k+1}$  do not represent the true error in estimation. It also suffers from convergence and stability problem, which means that if the approximation is not good enough the filter may diverge and result in poor tracking performance. The operation of EKF is summarized in figure 2.3.

## 2.3 Target Motion and Measurement Model

Two target models are used in this thesis to generate different trajectory scenarios. First is a constant-velocity (CV) model which models the linear trajectories of targets moving with constant velocity. Naturally, linear Kalman filter is used for tracking this model. The second model considers maneuvering targets with a constant turn rate and is tracked using extended Kalman filter. We discuss both models below.

### 2.3.1 Constant Velocity Model

Assume that the target is moving in a two-dimensional surveillance area with constant velocity. Ideally the acceleration of the target is zero i.e.  $\ddot{x} = 0$ . But in real world, the

target velocity will undergo some variation and this is modelled by the random process noise  $w_k$ . The state and observation vectors are:

$$\mathbf{x}_k = \begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} \quad \mathbf{y}_k = \begin{bmatrix} x_m(k) \\ y_m(k) \end{bmatrix} \quad (2.23)$$

where  $x(k)$  and  $y(k)$  are  $x$  and  $y$  position coordinates of the target and  $\dot{x}(k)$  and  $\dot{y}(k)$  are the velocities. Considering that the system has no external input, equations (2.2a)-(2.2b) reduce to:

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k \quad (2.24)$$

$$\mathbf{y}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (2.25)$$

The state transition and measurement matrix are given as:

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.26)$$

where  $\Delta T$  is the sampling time. Let the power spectral density of the process noise be  $q$ , then the covariance matrix is:

$$Q_k = q \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} & 0 & 0 \\ \frac{\Delta T^2}{2} & \Delta T & 0 & 0 \\ 0 & 0 & \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ 0 & 0 & \frac{\Delta T^2}{2} & \Delta T \end{bmatrix} \quad (2.27)$$

### 2.3.2 Coordinated Turn Rate and Velocity Model

A coordinated turn motion is a turn with a constant angular rate and velocity along a curved road of constant radius [14]. Since in reality the curvature of roads are not constant, a noise term is added to include the variation in curvature. The state vector for this model includes turn rate  $\omega$  in addition to the other states. Thus the state vector at scan  $k$  is  $\mathbf{x}_k = [x(k), \dot{x}(k), y(k), \dot{y}(k), \omega(k)]^T$ . The observation vector is same as in CV

model. The system state model is defined as follows:

$$f(\mathbf{x}_k) = \begin{bmatrix} x + \frac{\dot{x}}{\omega_k} \sin(\omega_k \Delta T) - \frac{\dot{y}}{\omega_k} (1 - \cos(\omega_k \Delta T)) \\ y + \frac{\dot{x}}{\omega_k} (1 - \cos(\omega_k \Delta T)) + \frac{\dot{y}}{\omega_k} \sin(\omega_k \Delta T) \\ \dot{x} \cos \omega_k \Delta T - \dot{y} \sin \omega_k \Delta T \\ \dot{x} \sin \omega_k \Delta T + \dot{y} \cos \omega_k \Delta T \\ \omega_k \end{bmatrix} \quad (2.28)$$

Since the transition matrix is a nonlinear function of  $\mathbf{x}_k$ , EKF is used instead of linear Kalman filter to track this model. Let  $\sigma_\omega$  be the standard deviation of turn rate, then the process noise covariance is given by:

$$Q_k = q \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} & 0 & 0 & 0 \\ \frac{\Delta T^2}{2} & \Delta T & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} & 0 \\ 0 & 0 & \frac{\Delta T^2}{2} & \Delta T & 0 \\ 0 & 0 & 0 & 0 & \frac{\sigma_\omega^2}{q} \end{bmatrix} \quad (2.29)$$

## 2.4 Clutter Model

In a tracking environment, there will be random interference in terms of undesirable measurements. These false measurements are called the *clutter*.

Consider that the sensor FOV is divided into  $N$  resolution cell or pixels. Then the clutter can be modelled based on the following assumptions [15]:

- the probability of false alarm detection in each cell is  $P_{FA} = p$ .
- The event of such detection in each cell is independent across time and of each other.
- they are uniformly spatially distributed.

Then the number of false alarms in these  $N$  cells follows the binomial distribution ( $m$  success in  $N$  trials):

$$P\{n_{FA} = m\} = \binom{N}{m} p^m (1-p)^{N-m} \quad (2.30)$$

Following (2.30), the average number of false alarms will be  $E[n_{FA}] = Np$ , and the spatial density as:

$$\lambda = \frac{E[n_{FA}]}{V_c} = \frac{Np}{V_c} \quad (2.31)$$

where  $V_c$  is the volume of  $N$  cells.

Equation (2.30) can be approximated with the Poisson distribution if  $N$  is large enough, such that  $Np$  is of order 1 or more:

$$P_{FA}(m) = e^{-Np} \frac{(Np)^m}{m!} \quad (2.32)$$

$$= e^{-\lambda V_c} \frac{(\lambda V_c)^m}{m!} \quad (2.33)$$

Hence, the number of false alarms in a certain FOV volume can be sampled from the Poisson distribution.

Then the spatial distribution of these false alarms follows the uniform distribution, which means that the clutter is uniformly distributed over the sensor FOV volume  $V_c$ :

$$p(y|y \text{ is a false alarm}) = \frac{1}{V_c} \quad (2.34)$$

where  $y$  denotes the measured detection.

## Chapter 3

# Data Association

This chapter discusses some of the common data association (DA) methods. Figure 3.2 illustrates the basic recursive flow of a simple data association algorithm. Lets assume that time is  $k > 0$  and the tracks has been initiated in the previous scan. As new data is received from the sensor, gating is performed to sort the reasonable track-to-observation pairs from the unlikely ones. This is done to reduce the number of hypotheses computations. Then data association is performed to select one association hypothesis from the list of all possible hypotheses. The tentative tracks are then tested for confirmation and deletion, where tracks that fulfill a certain confirmation criteria are said to represent a true target and low-quality tracks are assumed to represent false tracks and are deleted. This is known as track maintenance. Finally, filtering is performed to update the track state estimates using the assigned observations. A statistical score is maintained for each tentative track, which allows a probabilistic method for track deletion and confirmation. The track score function is introduced in section 3.1. The score is also directly convertible to probabilities that can be used to evaluate alternative association hypotheses. The process of gating is discussed in section 3.2. Section 3.3 discusses

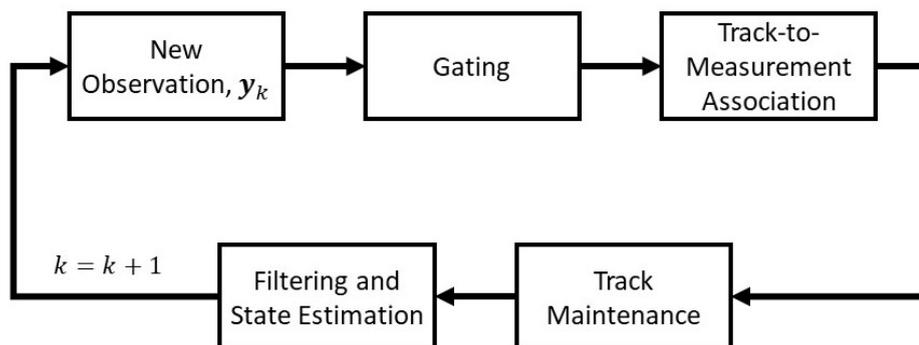


FIGURE 3.1: Flow of a conventional Data Association algorithm.

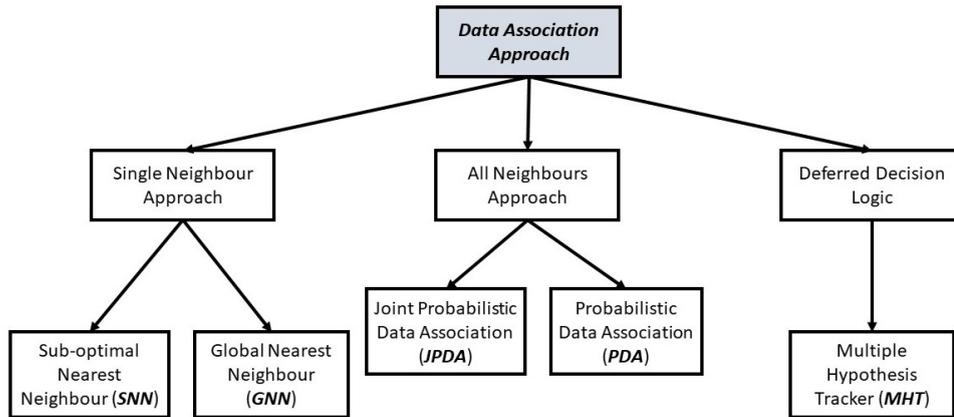


FIGURE 3.2: Types of data association algorithms.

the conventional single hypothesis, single neighbour data association approaches: sub-optimal nearest neighbour (SNN) and global nearest neighbour (GNN). The nearest neighbour approach (both SNN and GNN) attempts to select and propagate the single most likely hypothesis at each time step. It assigns a unique track-to-measurement pairing, so that at most one observation is used to update an already established track. Another data association approach is the all-neighbours approach, discussed in section 3.4, in which a track is updated using all neighbouring measurements that fall within the gated region. Probabilistic data association (PDA) is an all-neighbour approach which assumes that there is only one target of interest in clutter and it has been initialized [3]. But PDA does not perform well in multiple-target scenarios. Hence it was modified into the joint probabilistic data association (JPDA) for tracking multiple targets [15]. Multiple hypothesis tracker [4] is a deferred decision logic data association approach. It uses all gated observations to form multiple alternative tracks and then uses successive data to resolve the uncertainty in identifying the true track among the multiple possibilities. There are two basic implementations of MHT:

1. **Hypothesis oriented MHT (HOMHT)**: A hypothesis is a set of one or more tracks that are all compatible with each other. At each scan, as new data is received, the algorithm adds a new set of hypotheses to the old set (carried over from the previous scan) by considering all possible combinations of track-to-measurement assignments. Then the set is pruned by eliminating low probability hypotheses.
2. **Track oriented MHT (TOMHT)**: Instead of maintaining hypotheses from scan to scan, TOMHT propagates a set of potential tracks. These tracks are updated and pruned as new observations are received and then regrouped into hypotheses. A track score is used to delete unlikely or low score tracks before they are formed

into hypotheses and the surviving tracks are then passed on to the next scan. The implementation of track oriented MHT is presented in chapter 4.

### 3.1 Track Score

The track score function is used to evaluate the conditional probability of the alternative tracks. It defines the likelihood that the track corresponds to a true target as opposed to the likelihood that all detections are false alarms. True targets are generally assumed to be objects that persist in the tracking volume for several scans. False alarms, on the other hand, refer to the erroneous and unwanted detections that do not persist over multiple scans [4]. For maintaining the likelihood score, it is more convenient to use log-likelihood ratio (LLR) instead of likelihoods, as LLR can be added recursively whereas probabilities would have to be multiplied. The recursive formula for track score  $L_i(k)$ , for  $i^{th}$  track at scan  $k$  is [4]

$$L_i(k) = L_i(k-1) + \Delta L(k) \quad (3.1)$$

Here  $\Delta L(k)$ , the track score increment, is dependent on the availability of detection for track update, and is defined as,

$$\Delta L(k) = \begin{cases} \ln(1 - P_D) & ; \text{no track update on } k^{th} \text{ scan} \\ \Delta L_u(k) & ; \text{track update on } k^{th} \text{ scan} \end{cases} \quad (3.2)$$

Since  $P_D$  is always less than unity,  $\ln(1 - P_D) \leq 0$ . Therefore, the track score,  $L_i(k)$  will decrease whenever the track is not updated. On the contrary, the track will be incremented by a positive amount  $\Delta L_u(k)$  when a measurement is used to update the track. The increment,  $\Delta L_u$  is given by

$$\Delta L_u = \ln \left[ \frac{P_D V_C}{P_{FA} \sqrt{|S|}} \right] - \frac{M \ln(2\pi) + d^2}{2} \quad (3.3)$$

where,

$P_D$  = Probability of detection

$P_{FA}$  = Probability of false alarm

$M$  = Measurement dimension

$V_C$  = Volume of FOV (field of view)

$S$  = Residual covariance matrix

$d^2$  = Mahalanobis distance between observation and predicted target position  
 $= \nu^T S^{-1} \nu$

Defining the false alarm density as  $\beta_{FT} = P_{FA}/V_C$  and placing it in equation (3.3) results in

$$\Delta L_u = \ln \left[ \frac{P_D}{\beta_{FT} \sqrt{|S|} (2\pi)^{M/2}} \right] - \frac{d^2}{2} \quad (3.4)$$

The initial score for new tracks is based on the first measurement used to initiate the track and is calculated using (3.5), where  $\beta_{NT}$  is the new track density.

$$L_i(1) = \ln \left[ \frac{P_D \beta_{NT}}{\beta_{FT}} \right] \quad (3.5)$$

Track score is used to confirm potential tracks and delete unlikely ones. A track is deleted if its track score is below a certain fixed deletion threshold. Similarly, a track will be confirmed if its track score is above the confirmation threshold. The log-likelihood score defined by (3.1), can directly be converted to track validity probability through

$$p_i(k) = \frac{\exp(L_i(k))}{1 + \exp(L_i(k))} \quad (3.6)$$

## 3.2 Gating

Gating is the technique used to eliminate unlikely track-to-observation pairs. A gate is formed around the predicted track position and all observations that falls within the gate area (satisfy the gating condition) are considered to be viable options for track update. The process of gating is illustrated in figure 3.3. As defined in (2.10), innovation is the difference between the actual measurement and predicted measurement whose covariance matrix is  $S = (HP_{k+1|k}H^T + R)$ . Dropping the time index  $k$  for convenience, the norm of innovation is calculated as

$$d^2 = \nu^T S^{-1} \nu \quad (3.7)$$

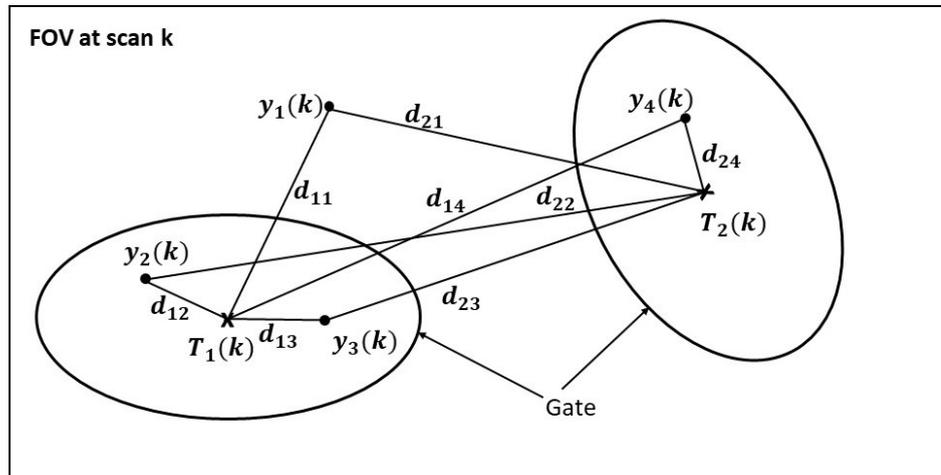


FIGURE 3.3: Illustration of the gating process.

Tracks	Observation			
	$y_1$	$y_2$	$y_3$	$y_4$
$T_1$	1000	$d_{12}^2$	$d_{13}^2$	1000
$T_2$	1000	1000	1000	$d_{24}^2$

TABLE 3.1: TMCR matrix for example shown in figure 3.3

$d^2$ , commonly known as the Mahalanobis distance, represents the statistical distance between tracks and observations. Association between a track and an observation is allowed if this distance is less than a gating threshold  $G$  i.e.  $d^2 = \nu^T S^{-1} \nu \leq G$ . The threshold  $G$  is taken from the chi-squared distribution where the degree of freedom is equal to the measurement dimension  $M$  [15]. For example, let there be two tracks at scan  $k - 1$ . If four measurements are received in the next scan  $k$ , as shown in figure 3.3, then the Mahalanobis distance for each of the measurement and track pair is computed using (3.7). The distance  $d_{ij}^2$  is then compared to the gating threshold  $G$  to eliminate the unlikely pairs. Given the distance function, a track to measurement correlation (TMCR) matrix is formed. Table.3.1 shows the assignment matrix for the example shown in figure ?? For gated measurements, the statistical distance is entered in the matrix. And a large number (say 1000) is entered for measurements that fail the gating test. The process of selecting the optimal assignment solution is done by solving this TMCR matrix and it depends on the data association approach used.

### 3.3 Single Hypothesis Tracking

The nearest neighbour (NN) approach is the simplest type of data association method. The algorithm maintains and propagates the single most likely hypothesis from scan to

Tracks	Observation					
	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
$T_1$	1000	1000	$d_{13}^2$	$d_{14}^2$	$d_{15}^2$	1000
$T_2$	1000	$d_{22}^2$	1000	1000	$d_{25}^2$	1000
$T_3$	$d_{31}^2$	$d_{32}^2$	1000	1000	1000	1000
$T_4$	1000	1000	1000	1000	1000	1000

TABLE 3.2: TMCR matrix for gating situation shown in figure 3.4

scan. As new set of data is received, the gates are formed around the existing tracks and the gated observations are sorted into the assignment matrix or TMCR matrix. This matrix is then solved to identify the most likely observation assignment for the existing tracks. Association conflicts arises when more than one measurement falls within the gate of a track and/or when there is an overlap of gates and one or more measurements are inside the gate of multiple tracks. One such situation is illustrated in figure 3.4. Table 3.2 shows the assignment matrix for a given example.

Based on the method used to solve the TMCR, NN approach can be of two type: Sub-optimal nearest neighbour (SNN) and global nearest neighbour (GNN):

- **Sub-optimal nearest neighbour:** In SNN [7], the TMCR matrix is searched for the minimum statistical distance  $d_{ij}^2$  and the  $j^{th}$  observation is assigned to  $i^{th}$  track. The assigned track-observation pair is then removed from the TMCR matrix and the process is repeated for the reduced matrix. This recursive process of search and remove is continued for all the tracks. For example, the minimum distance in table 3.2 is  $d_{25}^2$ . So track  $T_2$  will be updated using observation  $y_5$ . Next, the second row (corresponding to  $T_2$ ) and fifth column (corresponding to  $y_5$ ) will be removed from the matrix before searching for the next minimum distance.

A problem with SNN is that once an incorrect assignment is made, it is very unlikely that the track will recover and this results in track loss. Also, as the track-to-measurement assignment process is not optimal, SNN suffers from high rate of false association in presence of closely spaced targets and heavy clutter.

- **Global nearest neighbour:** GNN [15] uses optimal assignment algorithms like Munkres or Hungarian to solve the assignment problem. It solves the TMCR matrix by minimizing the summed total distance of the assigned track-to-observation pair. Since the association is optimal, GNN suffers from less number of track loss and false association as compared to SNN.

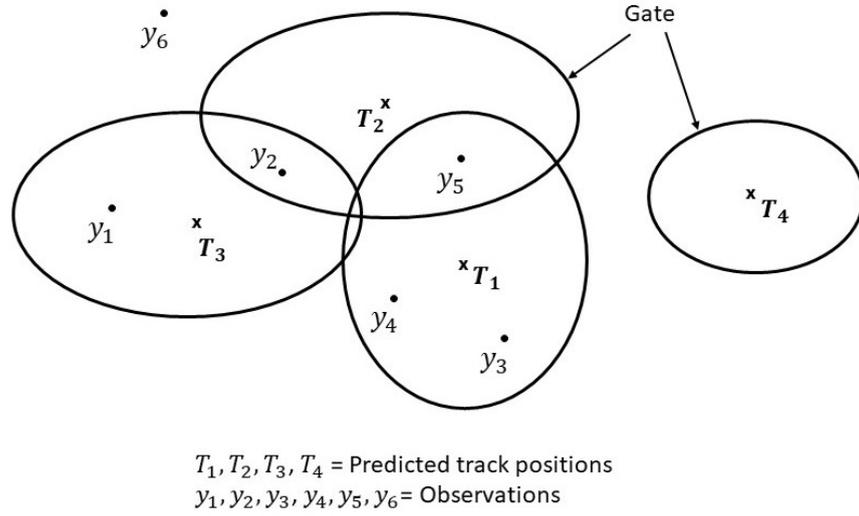


FIGURE 3.4: Association conflict situation.

Measurements that do not fall inside any gate are used to initiate new tracks. And tracks that has no gated or assigned measurement are simply extrapolated in time,

$$\hat{x}_{k|k} = \begin{cases} \hat{x}_{k|k-1} + K\nu_k & ; \text{if measurement is assigned for track update} \\ \hat{x}_{k|k-1} & ; \text{if no valid measurement for track update} \end{cases} \quad (3.8)$$

Track management is done to eliminate redundant and false tracks. An  $N_D$ -scan logic is used for track deletion, wherein tracks with  $N_D$  consecutive missed detections are assumed to be false tracks and are deleted [4]. Tracks that are very similar are combined together to avoid redundant tracks. Consider two tracks whose estimates at scan  $k$  are given as

$$\text{Track 1 : } \hat{x}_{k|k}^1 \text{ and } P_{k|k}^1 \quad (3.9)$$

$$\text{Track 2 : } \hat{x}_{k|k}^2 \text{ and } P_{k|k}^2 \quad (3.10)$$

If  $D_{th}$  is the merging threshold, then the two tracks will be fused if they satisfy the following condition:

$$(\hat{x}_{k|k}^1 - \hat{x}_{k|k}^2)^T (P_{k|k}^1 + P_{k|k}^2)^{-1} (\hat{x}_{k|k}^1 - \hat{x}_{k|k}^2) \leq D_{th} \quad (3.11)$$

And the combined state and covariance is given by [3]

$$\hat{x}_{k|k}^c = \hat{x}_{k|k}^1 + P_{k|k}^1 (P_{k|k}^1 + P_{k|k}^2)^{-1} (\hat{x}_{k|k}^1 - \hat{x}_{k|k}^2) \quad (3.12)$$

$$P_{k|k}^c = P_{k|k}^1 - P_{k|k}^1 (P_{k|k}^1 + P_{k|k}^2)^{-1} P_{k|k}^1 \quad (3.13)$$

### 3.4 All Neighbour Data Association Approach

The all neighbours approach uses all gated observations, weighted by their association probabilities to update the track of interest.

#### 3.4.1 Probabilistic Data Association

The probabilistic data association method assumes that there is only one target of interest in clutter and it has been initialized [3]. In presence of multiple targets, PDA evaluates each track individually in a sequential manner, like multiple single-target tracking. If there are  $m$  gated measurements for the  $i^{\text{th}}$  track, then the number of possible association hypotheses will be  $m + 1$ . The first hypothesis,  $H_0$  is the case that none of the measurements is target originated. Similarly, hypothesis  $H_j (j = 1, 2, \dots, m)$  is the event that the  $j^{\text{th}}$  measurement is target originated.

$$H_j = \begin{cases} \text{none of the measurement is valid} & ; j = 0 \\ y_j \text{ is the valid measurement} & ; j = (1, 2, \dots, m) \end{cases} \quad (3.14)$$

The subscript  $i$ , denoting track  $i$  is dropped for notational convenience. The association probability  $p_j$ , which is the probability of hypothesis  $H_j$  being true can be calculated as [15]

$$p_j = \begin{cases} \frac{b}{b + \sum_{k=1}^m a_k} & ; j = 0 \\ \frac{a_j}{b + \sum_{k=1}^m a_k} & ; 1 \leq j \leq m \end{cases} \quad (3.15)$$

where

$$b = P_{FA} \sqrt{|2\pi S|} \frac{1 - P_D}{P_D} \quad (3.16)$$

$$a_j = e^{-0.5 \nu_j^T S^{-1} \nu_j} \quad (3.17)$$

After computing the probabilities, the hypotheses are merged to get the current state estimates. The standard Kalman filter state update equation becomes [3]:

$$\hat{x}(k|k) = \sum_{j=0}^m \hat{x}_j(k|k) p_j \quad (3.18)$$

Where  $\hat{x}_j(k|k)$  is the estimated state conditioned on the  $j^{\text{th}}$  measurement being correct with a probability of  $p_j$ . This conditional estimate is given by

$$\hat{x}_j(k|k) = \begin{cases} \hat{x}(k|k-1) + K\nu_j & ; j = (1, 2, \dots, m) \\ \hat{x}(k|k-1) & ; j = 0 \end{cases} \quad (3.19)$$

where innovation  $\nu_j$  is

$$\nu_j = y_j - H\hat{x}(k|k-1) \quad (3.20)$$

The Kalman gain  $K$  and predicted state  $\hat{x}(k|k-1)$  are same as in equation (2.11) and (2.8) respectively. The combined covariance is

$$P(k|k) = p_0P(k|k-1) + (1-p_0)P_c(K|k) + P_s(k|k) \quad (3.21)$$

where

$$P_c(k|k) = P(k|k-1) - KSK^T \quad (3.22)$$

$$P_s(K|K) = K\left(\sum_{i=1}^m p_i\nu_i\nu_i^T - \nu\nu^T\right)K^T \quad (3.23)$$

$$\nu = \sum_{i=1}^m p_i\nu_i \quad (3.24)$$

The covariance given by (3.21) is an extension of equation (2.13) for PDAF. For the case that none of the gated measurements are correct, the state estimates will not be updated and estimated covariance will be  $P(k|k-1)$  weighted by the probability of this event  $p_0$ . For the event that correct measurement is available, the covariance update  $P_c(k|k)$  is weighted by probability  $(1-p_0)$ . But, since it is not known which of the  $m$  measurements is correct, the term  $P_s(k|k)$  is added to compensate for this error.

### 3.4.2 Joint Probabilistic Data Association

JPDA is an extension of PDA which uses all tracks and observations in the vicinity. The state estimation and covariance computations are same as PDA, using equations (3.18) to (3.24). Unlike PDA, JPDA assumes that there can be multiple targets with shared measurements in clutter [16]. Consider an association conflict situation shown in figure 3.5. There are three measurements, of which  $y_1$  falls within the gate of track  $T_2$ , observation  $y_2$  is inside the gates of both tracks  $T_1$  and  $T_2$  and  $y_3$  falls inside the gate of track  $T_1$ . While computing the association probabilities for track  $T_1$ , JPDA will give less weightage to observation  $y_2$  because of its presence inside the gate of  $T_2$ . A list of all possible association hypotheses is generated along with their corresponding hypothesis

probabilities. The hypothesis likelihoods are computed as [4]

$$p(H_k) = \prod_{\substack{\text{Track } i \text{ assigned} \\ \text{to measurement } j}} g_{ij} P_D \prod_{\substack{\text{Tracks assigned} \\ \text{to no measurement}}} (1 - P_D) \prod_{\substack{\text{Unassigned} \\ \text{measurements}}} P_{FA} \quad (3.25)$$

Where  $g_{ij}$  is the Gaussian likelihood function of the event that measurement  $j$  is assigned to track  $i$ , and is defined as,

$$g_{ij} = \frac{e^{-d_{ij}^2/2}}{(2\pi)^{M/2} \sqrt{|S_{ij}|}} \quad (3.26)$$

where  $d_{ij}^2$  is the Mahalanobis distance and  $S_{ij}$  is the residual covariance of the association of observation  $j$  to track  $i$ . Table.3.3 gives the hypothesis matrix for the example shown in figure 3.5, where  $j = 0$  refers to no observation assignment. Now to compute the association probability  $p_{ij}$  of assigning measurement  $i$  to track  $j$ , the hypothesis probabilities of those hypotheses are summed which includes this assignment. For example,

$$p_{10} = p(H_1) + p(H_4) + p(H_6) \quad (3.27)$$

$$p_{11} = p(H_2) + p(H_5) + p(H_7) \quad (3.28)$$

$$p_{20} = p(H_1) + p(H_2) + p(H_3) \quad (3.29)$$

$$p_{21} = 0 \quad (3.30)$$

The association probability of a non-gated observation is zero. The probability for observation that are not shared by any other track is computed to be more compared to an observation that is gated by multiple tracks. For instance,  $p_{21}$  will be more heavily weighted than  $p_{22}$  and  $p_{13}$  will be more than  $p_{12}$ . The rest of the calculation is done using (3.18) to (3.24).

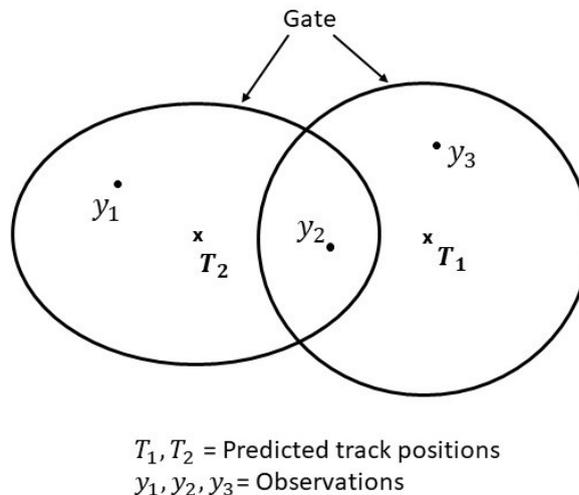


FIGURE 3.5: Example of association conflict for JPDA.

Hypothesis	Tracks		Hypothesis Probability
$H_k$	$T_1$	$T_2$	$p(H_k)$
1	0	0	$(1 - P_D)^2 P_{FA}^3$
2	1	0	$g_{11} P_D (1 - P_D) P_{FA}^2$
3	2	0	$g_{12} P_D (1 - P_D) P_{FA}^2$
4	0	2	$g_{22} P_D (1 - P_D) P_{FA}^2$
5	1	2	$g_{11} g_{22} P_D^2 P_{FA}$
6	0	3	$g_{23} P_D (1 - P_D) P_{FA}^2$
7	1	3	$g_{11} g_{23} P_D^2 P_{FA}$
8	2	3	$g_{12} g_{23} P_D^2 P_{FA}$

TABLE 3.3: Hypothesis matrix for example shown in figure 3.5

### 3.4.3 PDA Extensions

In both PDA and JPDA, the use of weighted-average method for track update leads to biased tracks in case of closely spaced targets. Which means that if the targets are in close proximity, their gates will overlap and they will be updated with the same sets of measurements and move closer than they actually are. In worst case, this often results in the two tracks to coalesce to a position midway between the two targets. A number of extensions of PDA and JPDA has been proposed to address the problem of track coalescence. The nearest-neighbour PDA or NNPDPA is one such extension, first proposed in [6]. NNPDPA is like a combination of GNN and PDA. It uses association probabilities to associate measurements to tracks on a one-to-one basis by solving the TMCR matrix. It is much like nearest neighbor approach, except that the measure of nearness is the association probability, rather than commonly used statistical distance [6]. A similar extension of JPDA was proposed in [17] called k-NNJPDA. k-NNJPDA forms all possible hypotheses, like in JPDA, and selects  $k$  most likely hypotheses for association probability calculation and track update. The value of  $k$  should be high enough to avoid track loss and low enough to prevent track coalescence [5]. When  $k = 1$ , k-NNJPDA will be equivalent to NNJPDA, so that each track will be updated with only one observation.

## Chapter 4

# Multiple Hypothesis Tracker

Multiple hypothesis tracker (MHT) is a deferred decision logic that maintains and propagates several data association (track-to-measurement association) solutions before making a firm decision, assuming that the track-to-measurement conflict will resolve as new data is received [4]. As discussed in chapter 3, based on how the association hypotheses are propagated in time, MHT can be of two types. HOMHT which maintains a set of most likely hypotheses from scan to scan and expands and prunes the list as new data is received. The other type of MHT is TOMHT which propagates a set of high quality potential tracks instead of hypotheses. Both track-oriented and hypothesis-oriented MHT suffer from the problem of combinational explosion of the number of hypotheses/tracks that can be formed at each scan. The number of possible tracks and hypotheses formed has to be kept under control to ensure the computational feasibility of MHT algorithm.

This thesis presents an implementation of TOMHT in an automotive tracking scenario with inner gating to make it more computationally feasible. We have used several techniques to limit the number of tracks in MHT. To represent and store all possible tracks and hypotheses in TOMHT, we have used a data structure called track tree (shown in figure 4.2 ). A family in this structure is defined as a set of tracks that have a common

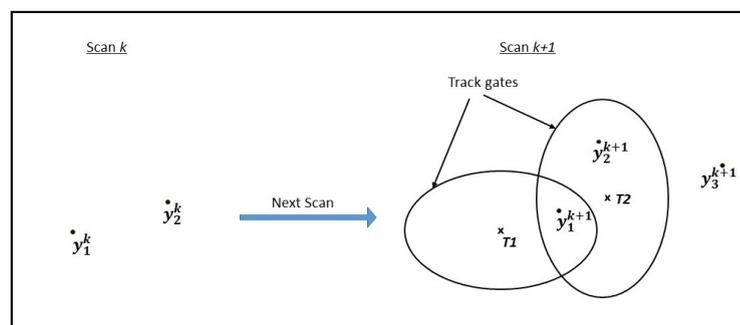


FIGURE 4.1: Association example for TOMHT track tree formation.

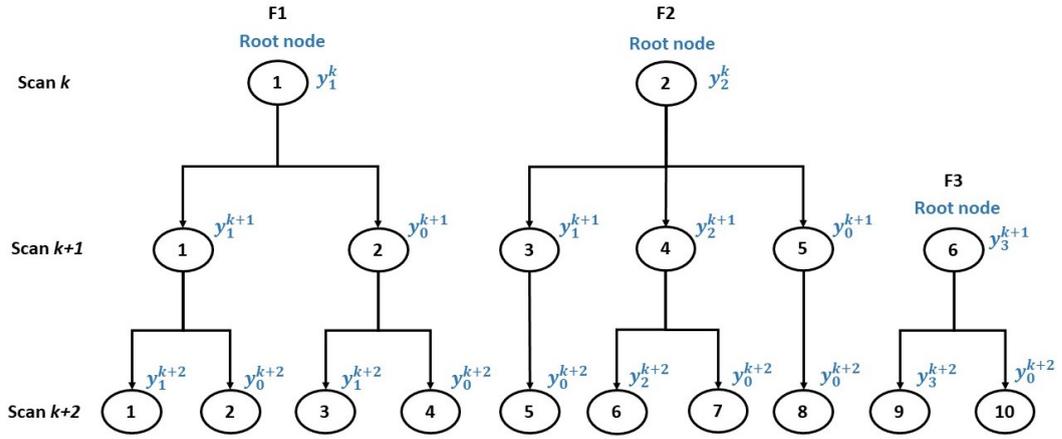


FIGURE 4.2: Tree structure for example in figure 4.1.

root node and corresponds to a single target [4]. Each branch emanating from a node in the family represents a different association hypothesis for that target. And all tracks in a family are incompatible with each other, as they share at least one common node. As a new set of observations is received, the tracks are updated using gated observations. The track tree is extended to include these new data association hypotheses and new trees are created that represents new potential targets. Pseudo-measurements are included for each track to represent the  $H_0$  hypothesis or missed detection.

Consider the gating scenario shown in figure 4.1. Let there be two detections at  $k^{th}$  scan,  $y^k = \{y_1^k, y_2^k\}$ . These observations are used to initiate two new tracks  $T_1^k$  (starting family  $F_1$ ) and  $T_2^k$  (starting family  $F_2$ ). Again at scan  $k+1$ , three observations are received and gating is performed around the predicted positions of the two tracks. Observation  $y_1^{k+1}$  falls inside the gates of both tracks  $T_1^k$  and  $T_2^k$ ;  $y_2^{k+1}$  is gated by  $T_2^k$ , as shown in figure 4.1. Figure 4.2 illustrates the track tree structure for this given example. As observation  $y_3^{k+1}$  does not fall inside any gate, it is used to initiate a new family,  $F_3$ . The other two tracks are updated with all viable measurements and missed detection hypothesis. As new data is received at each scan, the track tree is expanded to accommodate all possible association hypotheses. Thus, a great many tracks can potentially be formed in MHT, with many of tracks being incompatible with each other.

## 4.1 MHT Algorithm Elements

Figure 4.3 shows the flow of track-oriented MHT algorithm.

### 4.1.1 Track Formation

The first element of TOMHT is track formation which represents the operation and maintenance performed on the surviving tracks from the previous scan. At every scan, when a new set of observations is received, standard gating is performed on tracks that were carried over from the previous scan. Gating helps in determining viable measurement-to-track pairings. Each track is updated using all gated measurement and extrapolated (track update with no current measurement) to form potential tracks. Observations that does not fall within the gates of any track are used to initiate a new track. As there are no restrictions in MHT as to how many tracks can be updated using one observation, multiple tracks can share one or more common measurements. This can also lead to formation of too many tracks and hence to excessive computational requirements. We used a technique called **inner gating** to limit the number of tracks formed. Inner gating is when extrapolated tracks are only formed for those tracks that does not have any observation within its inner gate. It avoids the formation of redundant extrapolated tracks when the observations are close enough to the predicted track position. Although this greatly reduces the computational burden of TOMHT, it can also degrade the overall tracking performance when targets are in close proximity. Another technique is to not let the number of tracks formed by the system to exceed a maximum limit,  $N_{max}$ . When the number of tracks reaches the limit  $N_{max}$ , a pruning process is initiated and no new tracks are allowed to be formed.

### 4.1.2 Track-Level Pruning and Confirmation

Track score and track validity probability is computed for each track in the track tree structure, using equations (3.1) and (3.6) respectively. In track-level pruning, these track scores are compared with a deletion threshold. Tracks that have a score less than this threshold are deleted. Surviving tracks are then checked for confirmation, that is, track will be confirmed only if its score is above a fixed confirmation threshold. This confirmation status is used to determine the eligibility of a track for user representation (explained in section 4.2).

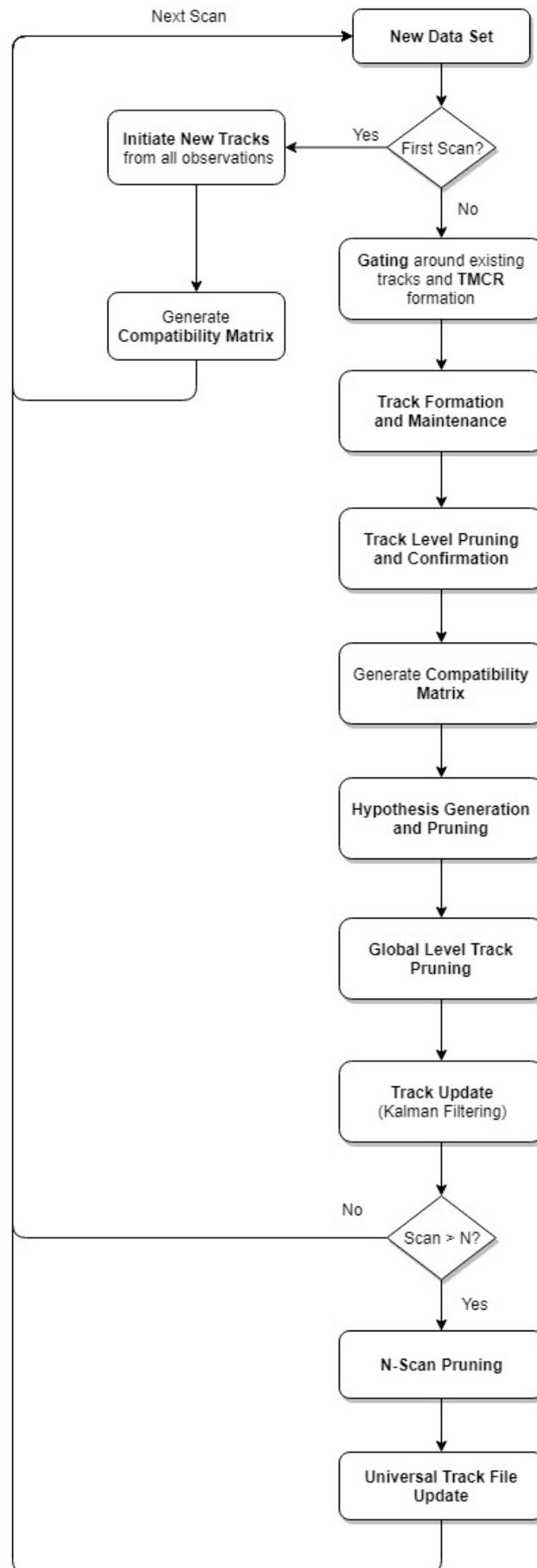


FIGURE 4.3: Flowchart of track-oriented MHT logic.

### 4.1.3 Confirmation Matrix

Tracks are considered to be compatible if they have no observation in common. Incompatibility between two tracks are passed on to their descendant tracks as well. For each track a compatibility list (list of tracks which are compatible to that track) is required for hypothesis formation. A compatibility matrix is maintained and stored at each scan. If number of existing tracks is  $n_T$ , then compatibility matrix will be an  $(n_T \times n_T)$  defined as follows

$$C_{ij} = \begin{cases} 1 & ; \text{if } i^{\text{th}} \text{ and } j^{\text{th}} \text{ tracks are compatible} \\ 0 & ; \text{if } i^{\text{th}} \text{ and } j^{\text{th}} \text{ tracks are incompatible} \end{cases} \quad (4.1)$$

This matrix is formed by comparing all tracks with each other for compatibility. Two tracks are considered to be incompatible if

1. Both tracks belong to the same family, which signifies that they spawned from the same parent track.
2. They share common observation in current scan.
3. They were incompatible in the previous scan, which signifies that they shared a common observation sometime in the past.

For example, consider tracks at scan  $(k + 1)$  in figure 4.2. The compatibility matrix for this set of tracks is given in table 4.1.

### 4.1.4 Hypothesis Generation and Pruning

A hypothesis is a set of compatible tracks that together represents the multiple targets being tracked. Number of tracks in a hypothesis is not fixed and can range from one to all tracks in the track file. Multiple such hypothesis are formed to account for all possible combinations of compatible tracks. This can lead to combinational explosion

Tracks	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$
$T_1$	0	0	0	1	1	1
$T_2$	0	0	1	1	1	1
$T_3$	0	1	0	0	0	1
$T_4$	1	1	0	0	0	1
$T_5$	1	1	0	0	0	1
$T_6$	1	1	1	1	1	0

TABLE 4.1: Compatibility matrix for tracks at scan  $(k + 1)$  in figure 4.2

in number of hypotheses as the track tree grows. A breadth-first approach is used for hypothesis formation in this implementation. It starts by defining one track hypotheses and then expanding them by adding in more tracks. As tracks are added to a hypothesis, their track score are also summed to compute the hypothesis score. The score of a given hypothesis  $H_j$  can be written as

$$L_{H_j} = \sum_{T_i \in H_j} L_i \quad (4.2)$$

Compatibility constraint of tracks in a hypothesis should be maintained while adding more tracks to it, this can be achieved easily by using the compatibility matrix. The expansion of hypotheses is done using only positive score tracks first. After this process, there will be a list of hypothesis with only positively scored tracks in them. Then tracks with negative scores are added, but this can only be done till the hypothesis score is above a set threshold. By the end of this process, there could be multiple hypotheses with same set of tracks in them; such repeated hypotheses are deleted. Also hypotheses with very low hypothesis score are pruned. The hypothesis with the highest score is called the most likely hypothesis and is used for updating the global track file for user representation.

#### 4.1.5 Global Level Pruning

As a given track can be included in more than one hypotheses, the global track probability,  $p_{T_i}$ , is computed as the sum of probabilities of all hypotheses that contain that track.

$$p_{T_i} = \sum_{\forall j | T_i \in H_j} p_{H_j} \quad (4.3)$$

where  $p_{H_j}$  is the probability of hypothesis  $j$  and is computed using hypothesis score  $H_j$  and scores of all  $J$  hypotheses

$$p_{H_j} = \frac{\exp(L_{H_j})}{1 + \sum_{k=1}^J \exp(L_{H_k})} \quad (4.4)$$

The global level pruning is done by comparing the track probability given by equation (4.3). Low score tracks that are not part of any surviving hypotheses will have zero track probability. Tracks with probability below a fixed deletion threshold are deleted. But this pruning is only done on unconfirmed tracks. If the track is confirmed, then it must have five consecutive miss detection along with a low track probability for it to be deleted. Finally Kalman state estimation is done on tracks that survive this pruning.

As Kalman filtering step is computationally heavy, this step is performed only when all poor quality tracks have been pruned.

#### 4.1.6 N-Scan Pruning

To further restrict the number of branches in the track tree, an N-scan pruning [4] technique is used. For trees with depth more than  $N$ , a new root node is assigned going  $N$  scans back, based on tracks included in the most likely hypothesis. And all tracks that do not share this newly assigned root node are deleted. A family that does not have a track in the most likely hypothesis, will not have a new root node. Such families will be checked for any confirmed tracks, if none of its tracks are confirmed then the whole family can be deleted. But if there is even one confirmed track in that family then the new root node will be selected on the basis of highest sum track probabilities of leaf nodes. In N-Scan pruning, the value of depth ( $N$ ) limits the number of branches formed. Taking a large  $N$  will allow for more observations to be included before the decision of pruning is made. This will improve system performance, but will also allow a lot of branches to be formed resulting in excessive computational time requirement. On the other hand, too small a value of  $N$  will greatly compromise the performance. So  $N$  should be taken such that there is a balance between computational complexity and performance. In this implementation  $N$  is set at 3. For example, again consider the track tree illustrated in figure 4.2. Let at scan  $k + 2$ , the most likely hypothesis include tracks  $\{1, 7, 9\}$  and for this example  $N = 1$ . Then going one scan back, new root nodes are established, as illustrated in figure 4. Tracks that do not share this newly assigned root node are deleted. In this example, deleted tracks will be  $\{3, 4, 5, 6\}$ .

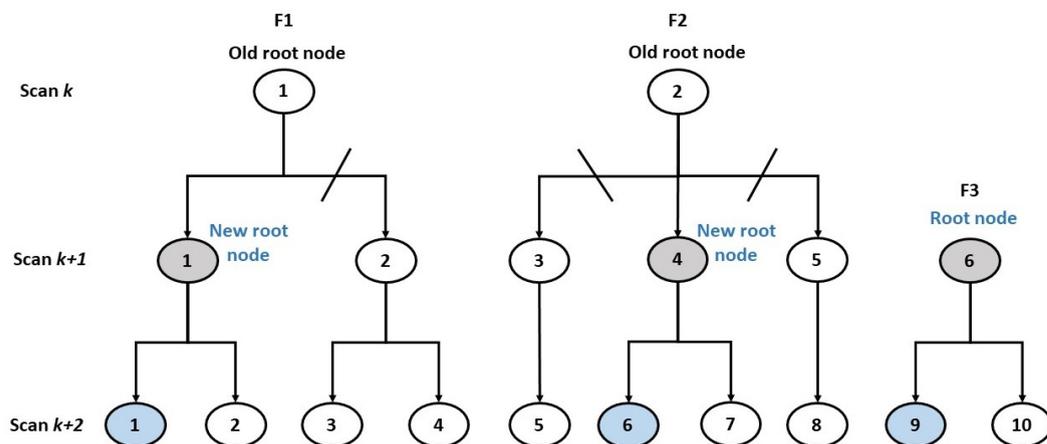


FIGURE 4.4: Example of N-Scan pruning.

## 4.2 Presentation of MHT Data

As seen above, a TOMHT system produces several potential tracks corresponding to the same target. Also, tracks are formed and deleted multiple times at every scan, which can lead to inconsistency in track identities associated with each track. Thus, a direct continuous output to user becomes difficult. To address this problem two set of tracks are maintained; **primary track file** and **secondary track file**.

### 4.2.1 Primary Track File

The first set of tracks is known as primary tracks, it contains all surviving tracks that are carried forward to the next scan. These tracks are arranged in decreasing order of their track score. Primary track file can contain more than one track for each target. Gating, track formation, track level pruning and global level pruning are performed on members of this track file. Table 4.2 presents the structure of primary track file. Table 4.2 is described in detail as follows:

- **Start Index** contains the measurement ID of the observation used to initiate the family to which that track belongs. It also contains the scan number at which its family was first formed. This is done to keep track of members belonging to one family. Tracks with same start index will have emanated from the same root node and hence belong to the same family. For example in figure 4.2, start index of track  $T_3^{k+3}$  will be  $[k, 1]$  and that of track  $T_9^{k+2}$  will be  $[k + 1, 3]$
- **TID** stands for track ID and holds an integer number that represents a track. This can change in each scan as tracks are deleted and new ones are added to the track file.
- **TID History** holds the list of all TIDs that was assigned to the track at each scan. This field signifies the parent node from which a track was spawned. In the example shown in figure 4.2, TID history of  $T_8^{k+2}$  will be  $[2, 5, 8]$ .
- **MID** contains the list of IDs of measurement that was used to update the track in every scan. For track  $T_3^{k+2}$ , MID will be  $[1, 0, 1]$ .
- **CID** can hold only two values, 1 or 0, and signifies if a track is confirmed or not respectively.
- **Track score, track validity probability and track probability** are calculated using equations (3.1), (3.6) and (4.3) respectively.

1	2	3	4	5	6	7	8	9	10
Start Index	TID	TID History	MID	Est. State	Est. Covar	CID	Track Score	Validity Prob	Track Prob
$[k, m_i^k]$	$T_i^k$	$\{T_i^k\}_{k=1}^K$	$\{m_i^k\}_{k=1}^K$	$\hat{x}(k k)$	$P(k k)$	0/1	$L_i(k)$	$p_i$	$p_{T_i}$

TABLE 4.2: Primary track file structure.

#### 4.2.2 Global Track File

This track file is used for user representation. There can be only one track representing a target in this set, that is, there can be only one track from each family. At each scan, after all data processing is done, most likely confirmed tracks are chosen from the primary track set. Then a primary-to-global track assignment is done to update the global tracks with primary tracks that best represents the target. This is done by using the tree structure shown in figure 4.2. At each scan, hypothesis formation is done using primary tracks. Once global track file is formed, confirmed primary tracks that are included in the most likely hypothesis (hypothesis with highest score) and belongs to same family as the target in global track, can be used to update it. Unassigned tracks from most likely hypothesis are used to create new global tracks. Global tracks that do not get a primary track assignment are extrapolated for several scans (typically four or five) before getting deleted. And since each hypothesis can have only one track from a family, there will be only one track corresponding to a target in the global track file. Table 4.3 presents the structure of global track file used.

1	2	3	4	5	6
Start Index	TID	MID	Est. State	Est. Covar	AID
$[k, m_i^k]$	$T_i^k$	$\{m_i^k\}_{k=1}^K$	$\hat{x}(k k)$	$P(k k)$	0/1

TABLE 4.3: Global track file structure.

## Chapter 5

# Simulation Results

### 5.1 OSPA Performance Metric

In a single-object system, the root mean square (RMS) error is used for performance evaluation. RMS is the difference between estimated state and true state. But a multiple-target tracker is different from a single-target system in the sense that both the target states and the number of targets are unknown. For example, consider a tracking situation where there are two true targets present in the FOV and the estimated number of targets are three. A simple RMS error will not work in this condition for two reasons:

1. It is not known which ground truth is to be compared with which estimated track
2. the error in estimating the number of targets also has to be captured.

The optimal sub-pattern assignment (OSPA) metric was first introduced in [18] as a consistent metric for evaluating the performance of multiple-target trackers. The OSPA metric is comprised of two components: localization error and cardinality error; the localization error signifies the inaccuracy in state estimation and cardinality error captures the error in estimating the number of targets.

Let at  $k^{th}$  scan, the set of estimated tracks be  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  and set of true targets be  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ , where  $m$  and  $n$  are the number of estimated and true tracks respectively. An  $m \times n$  cost matrix is then computed whose entries are:

$$C_{ij} = \min(c, d(\mathbf{x}_i, \mathbf{y}_j)) \quad (5.1)$$

where  $d(\mathbf{x}_i, \mathbf{y}_j)$  is the Euclidean distance between the two tracks and  $c$  is a positive unitless parameter known as the association cut-off radius. The cost matrix is solved for

optimal  $\mathbf{X}$ - $\mathbf{Y}$  assignment using optimal assignment algorithms like Munkres [19], this will give us the minimum summed  $d_{cost}$ .  $p$  is the order of OSPA metric and can be  $0 \leq p \leq \infty$ . The OSPA metric can be computed as:

$$d_{ospa} = \left[ \frac{1}{\max(m, n)} (d_{cost}^p + c^p |m - n|) \right]^{1/p} \quad (5.2)$$

$$d_{loc} = \left[ \frac{d_{cost}^p}{\max(m, n)} \right]^{1/p} \quad \text{and} \quad d_{card} = \left[ \frac{c^p |m - n|}{\max(m, n)} \right]^{1/p} \quad (5.3)$$

The order parameter  $p$  defines sensitivity to outliers or estimates that are not close to any ground truth target. The metric becomes more unforgiving to the outliers for greater values of  $p$ . On the other hand, the parameter  $c$  determines the importance of target number accuracy as compared to localization error. If  $c$  is small then localization error is more strongly weighted than cardinality error and vice versa. It also gives the cut-off distance for deciding whether two points in  $\mathbf{X}$  and  $\mathbf{Y}$  should be paired together.

## 5.2 Results

Track oriented MHT is implemented, both with and without inner gating in the presence of measurement origin uncertainty. The performance of TOMHT is compared with other data-association algorithms namely, sub-optimal nearest neighbor (SNN), global nearest neighbor (GNN), probabilistic data association (PDA) and nearest-neighbor PDA (NN-PDA). These data association algorithms are implemented for four different multiple-target tracking scenarios and compared for a fixed detection probability and false alarm density. OSPA metric is used for performance evaluation and comparison.

Out of the four scenarios implemented, three were modelled on the constant velocity (CV) model with linear trajectories, discussed in section 2.3.1. The tracker is executed for 10s with a sampling period ( $\Delta T$ ) of 0.1s, giving a total of 100 scans. The process noise intensity for CV model is taken to be  $q = 0.01m^2/s^3$  and measurement noise covariance matrix  $R = \text{diag}\{\sigma_x^2, \sigma_y^2\}$ , where  $\sigma_x^2 = \sigma_y^2 = 0.25m^2$ . The last scenario is modelled using the coordinated turn rate and velocity (CTRV) model for a period of 25s and 250 scans. The spectral density of process noise is set as  $q = 0.1m^2/s^3$  and standard deviation of turn rate is  $\sigma_\omega^2 = 0.01rad^2$ .

### 5.2.1 Scenario 1: Three Sparse Targets

Consider three sparsely distributed linearly moving targets with constant velocity. Figure 5.1 shows the ground truth vs estimated tracks obtained by the proposed TOMHT

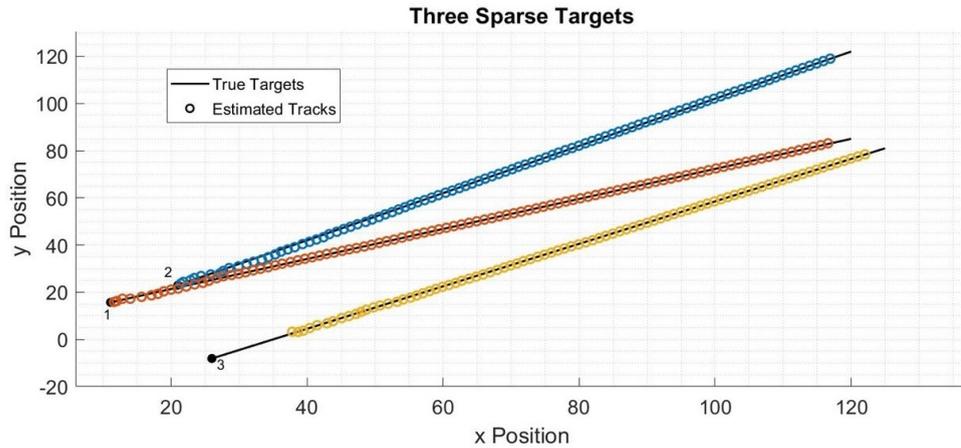


FIGURE 5.1: True and estimated trajectories of scenario 1.

with inner gating. Note that all targets are born at scan  $k = 1$  and dies at scan  $k = 100$ . For this scenario, probability of detection,  $P_d = 0.9$  and average number of false alarms per scan,  $N_{fa} = 3$ . The average performance is computed over 100 Monte Carlo runs.

Figure 5.2 shows the OSPA distance for MHT, MHT with inner gating, GNN, PDA and NN-PDA. The order parameters  $p$  and  $c$  were set at 1 and 20 respectively. Figures 5.3 and 5.4 shows the two components of OSPA: localization and cardinality. A comparison of single hypothesis trackers and multiple hypothesis tracker shows that after the initial settle-in phase, the localization error in all filters stabilizes to almost the same level. On the other hand, cardinality OSPA is minimum in case of MHT and worst in case of PDA. This indicates that most number of false tracks are generated in PDA, compared to other trackers. Table 5.1 compares the computation time of the two MHT algorithms. We can conclude that the use of inner gating significantly improves the running time of MHT without compromising its performance.

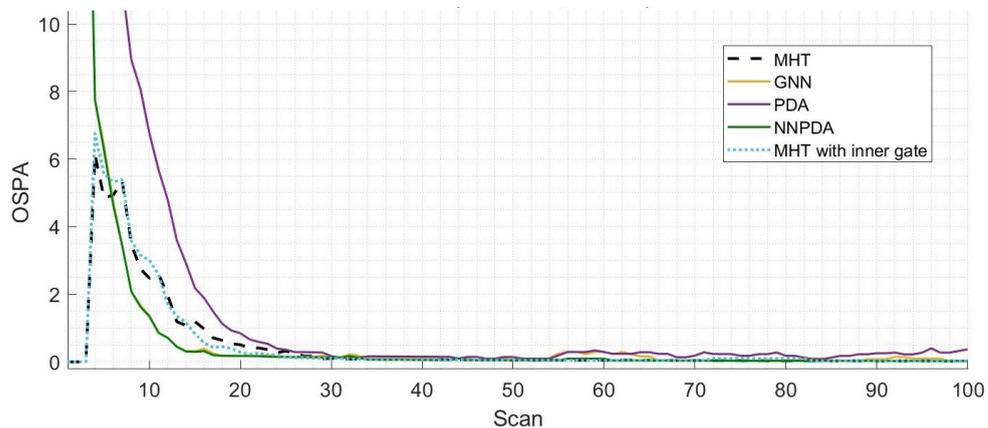


FIGURE 5.2: OSPA distance for scenario 1.

Tracking Algorithm	Running time (sec)
MHT	1890
MHT with inner gating	165

TABLE 5.1: Computation time of MHT for scenario 1.

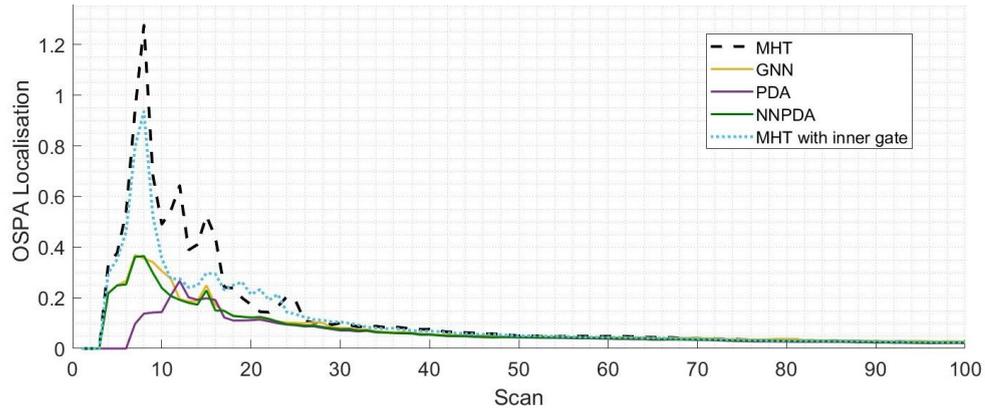


FIGURE 5.3: OSPA localization component for scenario 1.

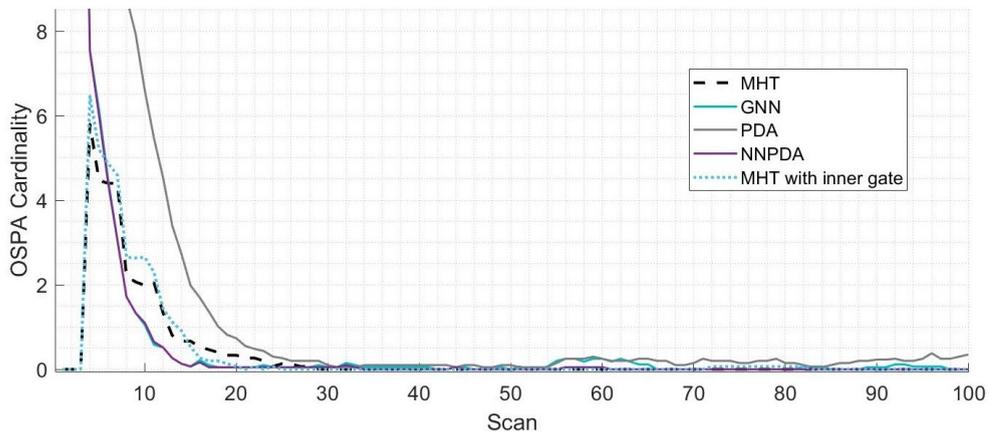


FIGURE 5.4: OSPA cardinality component for scenario 1.

### 5.2.2 Scenario 2: Three Crossing Targets

This scenario has three linear targets moving with a constant velocity. Target 1 and target 2 are crossing paths at  $k = 20$ , and target 1 and target 3 are crossing at  $k = 70$ . Figure 5.5 shows the true target trajectories versus estimated tracks as tracked by MHT (with inner gating). For comparison, the detection probability is taken to be 0.9 and average number of false alarm is 3. The OSPA comparison is shown in figure 5.6 for all five trackers obtained over 100 MC runs. Figure 5.7 and 5.8 shows the localization and cardinality errors. The occasional spikes in localization error occurs due to track loss and is worse in case of PDA and SNN. This is due to the problem of track coalescence in PDA when targets are in close proximity and high track loss rate in SNN. NN-PDA shows the best performance in this scenario, both in terms of cardinality and localization

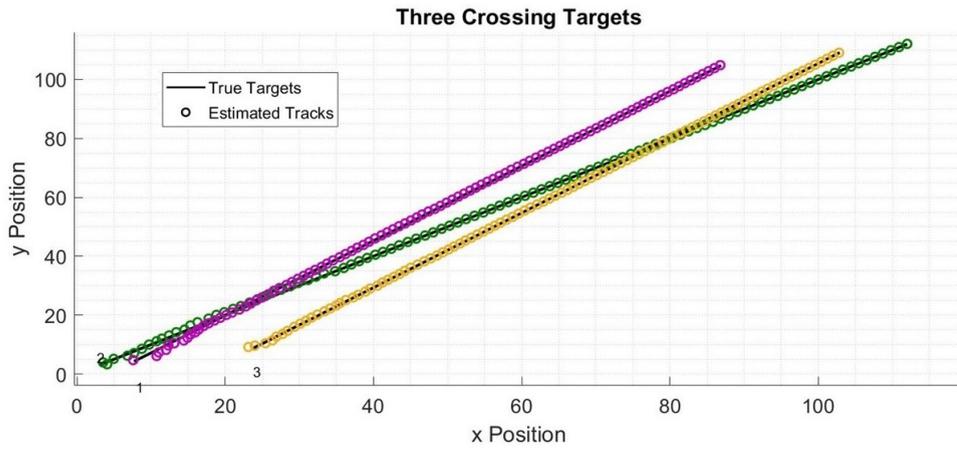


FIGURE 5.5: True and estimated trajectories of scenario 2.

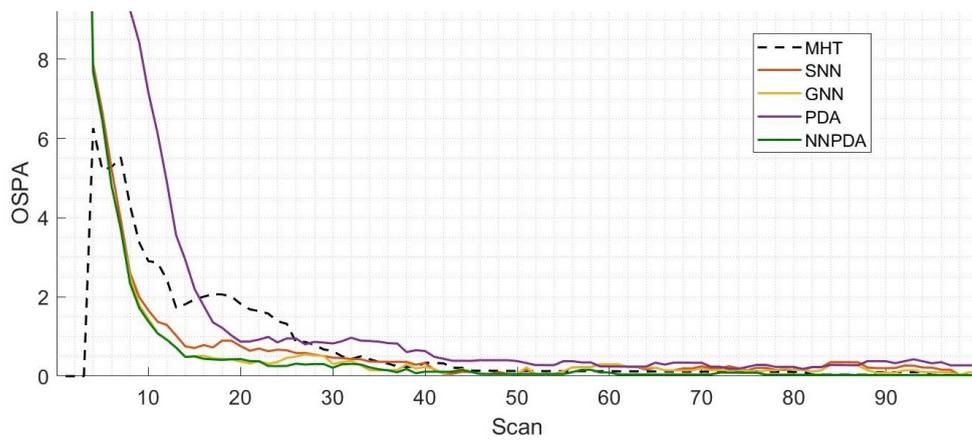


FIGURE 5.6: OSPA distance for scenario 2.

error. MHT also has almost the same performance as NN-PDA, except that it takes a slightly longer time to settle to an average value.

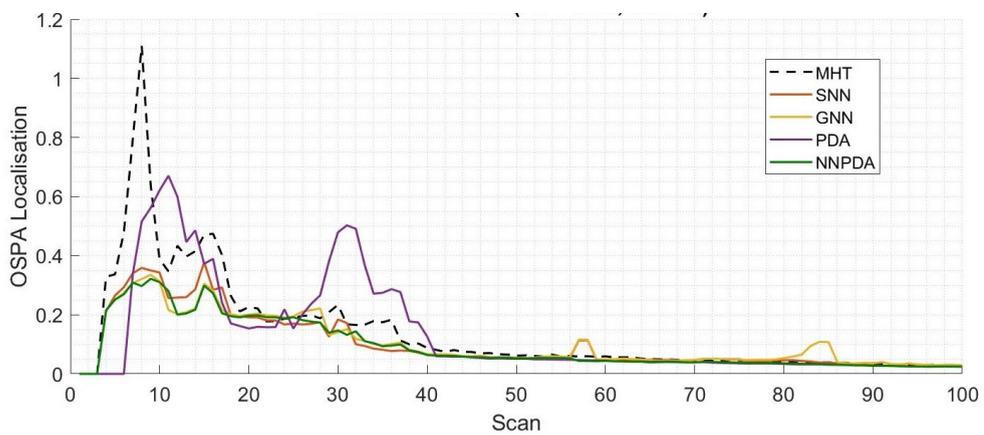


FIGURE 5.7: OSPA localization component for scenario 2.

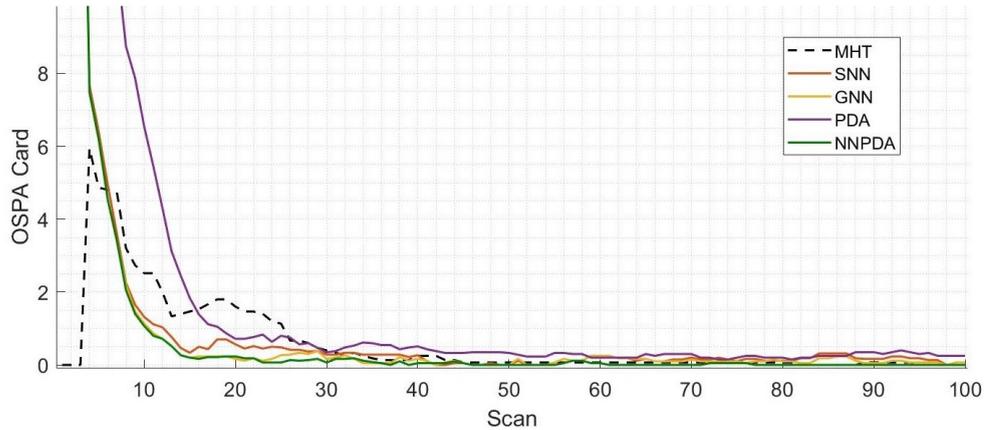


FIGURE 5.8: OSPA cardinality component for scenario 2.

### 5.2.3 Scenario 3: Three Parallel Targets

In this scenario, three closely spaced linear targets are considered. All three are moving in a straight line with constant velocity and parallel to each other. The distance between two adjacent targets is kept at 2 units. This multiple-object trajectory is tracked using all five trackers. Figure 5.9 and 5.10 shows the estimated trajectories tracked by PDA and MHT respectively. We can see that PDA results in a lot of track breakage and track coalescence, whereas MHT gives a much smoother result.

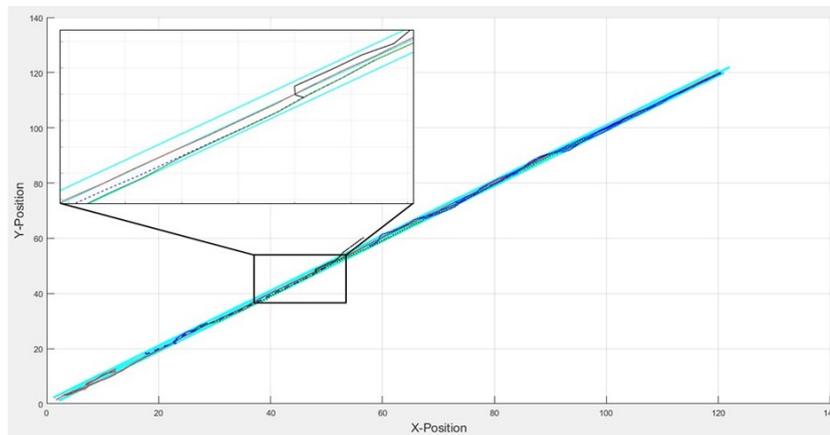


FIGURE 5.9: True and estimated trajectories of scenario 3 obtained by PDA.

Figures 5.11, 5.12 and 5.13 shows the OSPA distance for 100 MC trail runs. All filters are implemented for  $P_d = 0.9$  and  $N_{fa} = 3$ . It is clear that PDA has the worst performance both in terms of cardinality and localization error, closely followed by SNN. Whereas, GNN and NNPD perform the best with lowest localization error and close to zero OSPA cardinality. MHT when used with inner gating gives an OSPA cardinality which approaches SNN, which in turn increases its overall OSPA distance.

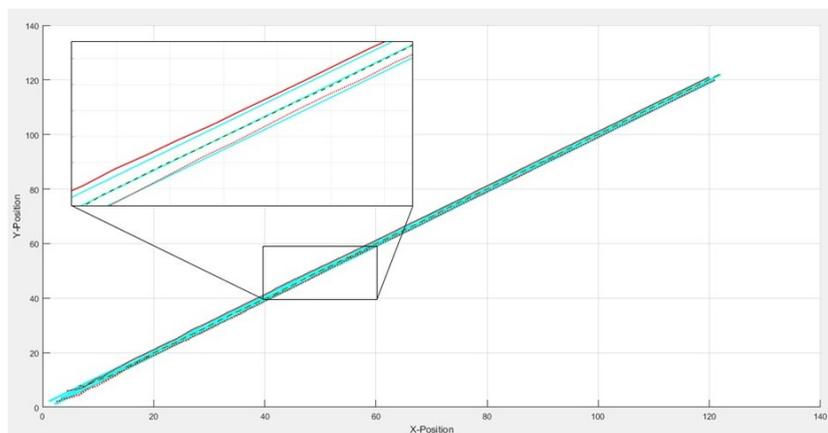


FIGURE 5.10: True and estimated trajectories of scenario 3 obtained by MHT.

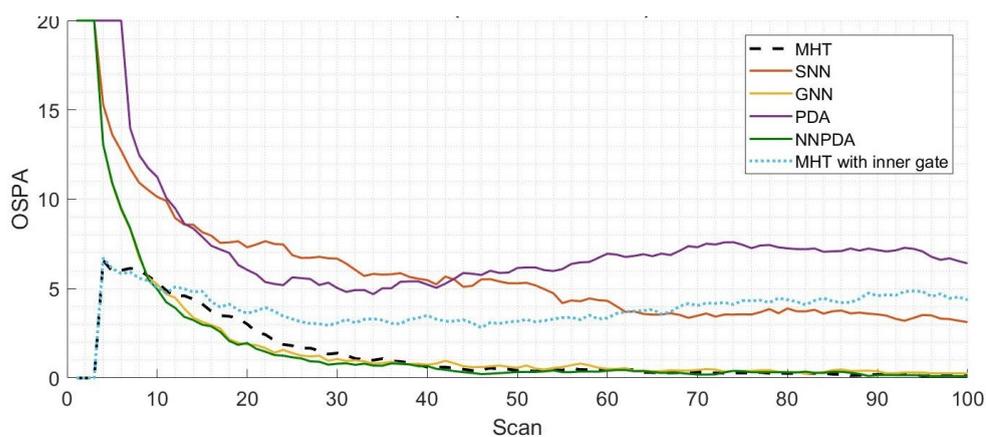


FIGURE 5.11: OSPA distance for scenario 3.

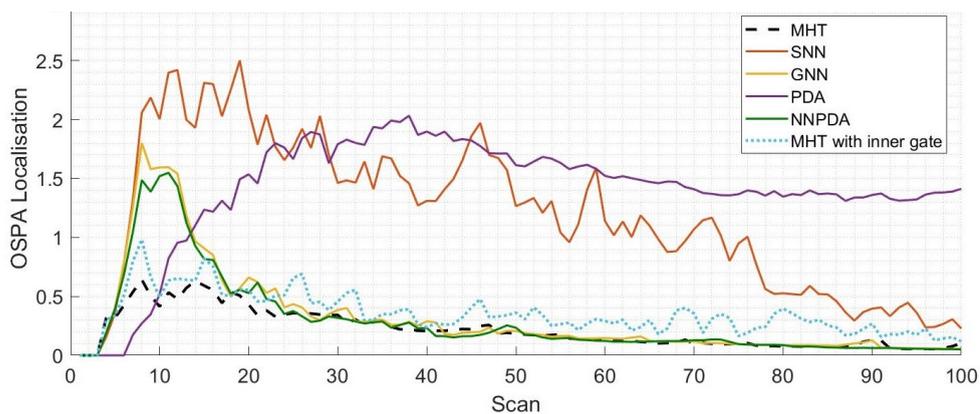


FIGURE 5.12: OSPA localization component for scenario 3.

#### 5.2.4 Scenario 4: Two Maneuvering Targets

Here two maneuvering targets are simulated using the CTRV model. The two targets takes a U-turn in close proximity without crossing each other and then moves apart. Figure 5.14 shows the tracked trajectory obtained by MHT, and figure 5.15 shows the same trajectory tracked by PDA. We can see that track coalescence is happening in

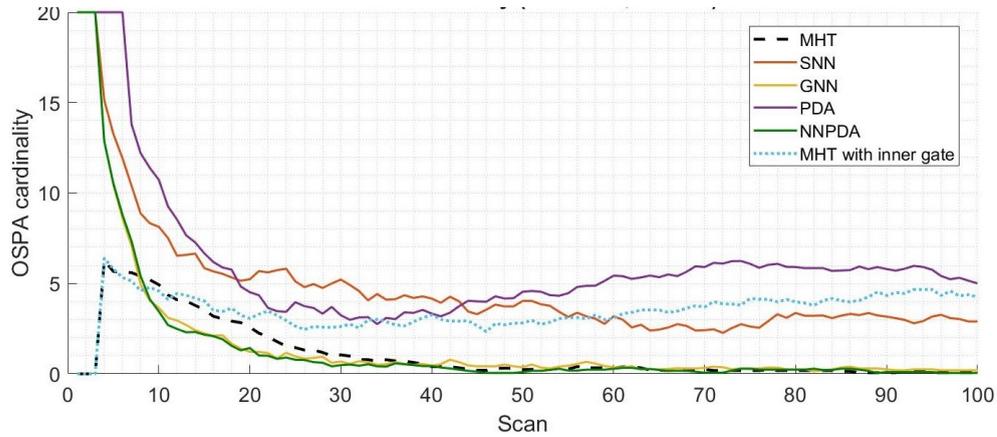


FIGURE 5.13: OSPA cardinality component for scenario 3.

figure 5.15, where the estimated tracks 1 starts following the wrong target when the two targets are close to each other. This trajectory was tracked by seven data association algorithms: GNN, PDA, JPDA, NN-PDA, kNN-JPDA, NN-JPDA and MHT. Figure 5.16 shows the OSPA distance comparison for this scenario. It is evident that the problem of track coalescence is worse in case of PDA and JPDA, resulting in a high OSPA peak. The performance improves a little with the two JPDA extensions: kNN-JPDA and NN-JPDA. MHT is shown to give one of the best results, along with GNN and NN-PDA.

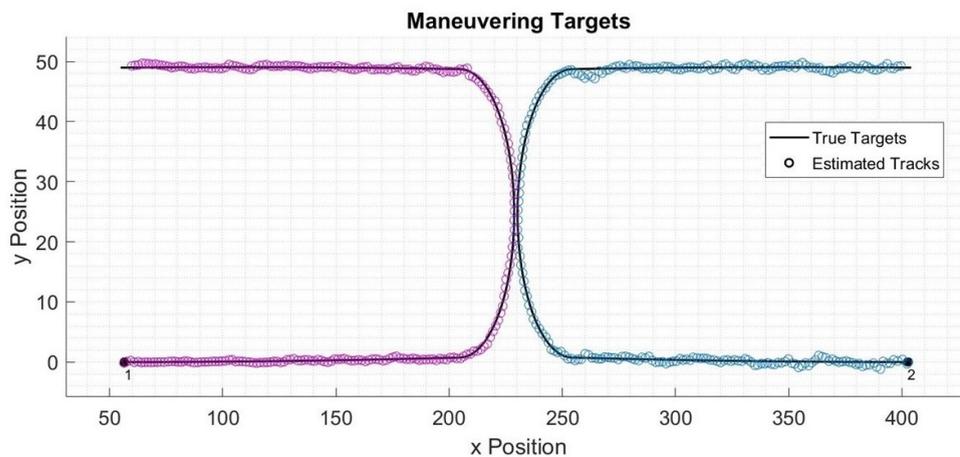


FIGURE 5.14: True and estimated trajectories of scenario 4 obtained by PDA.

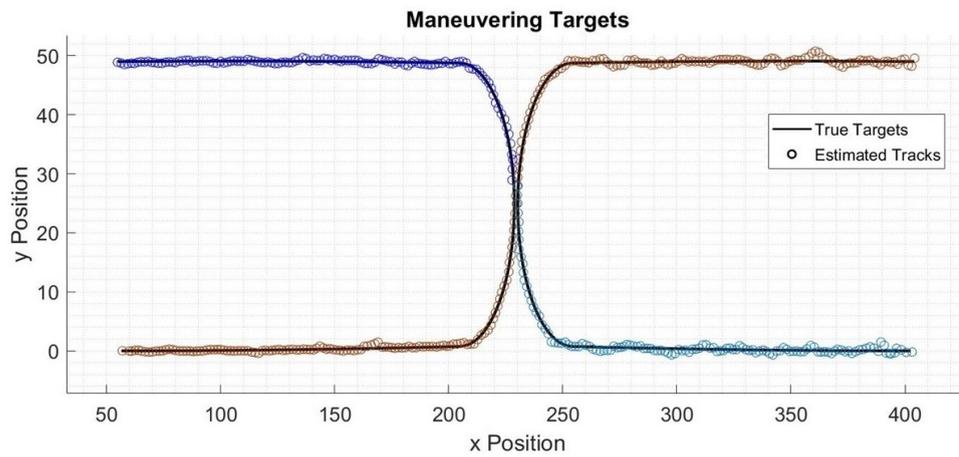


FIGURE 5.15: True and estimated trajectories of scenario 4 obtained by MHT.

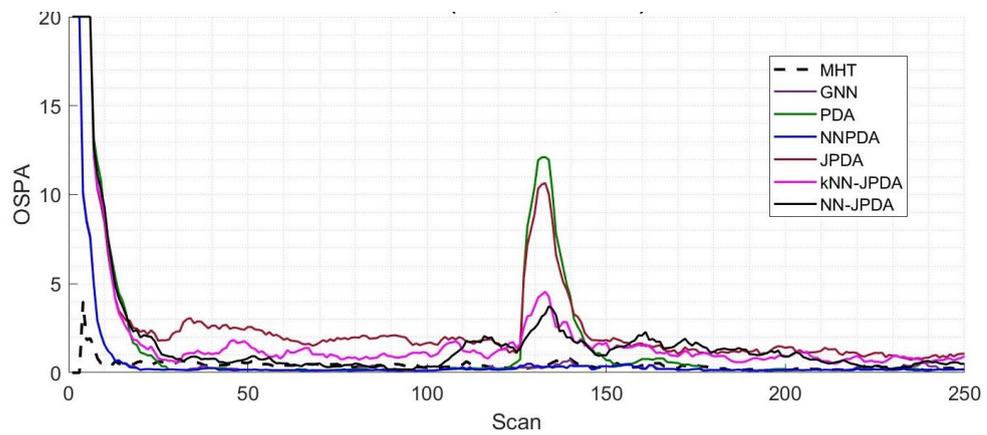


FIGURE 5.16: OSPA distance for scenario 4.

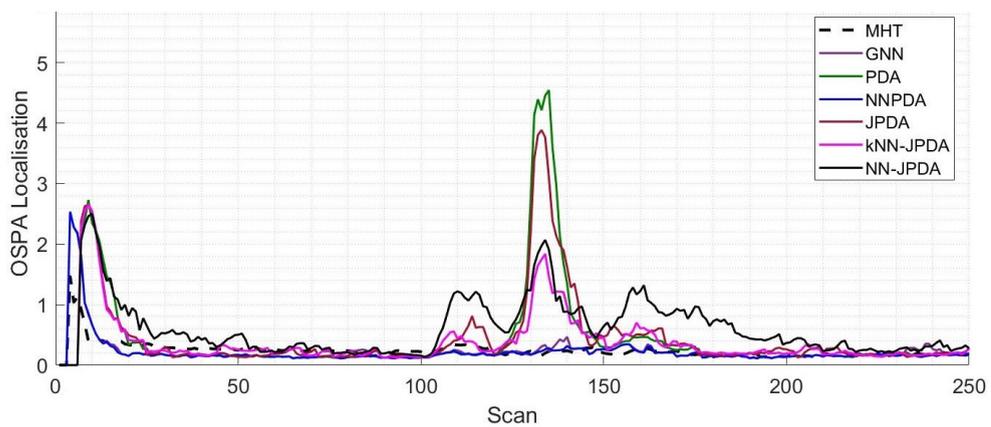


FIGURE 5.17: OSPA localization component for scenario 4.

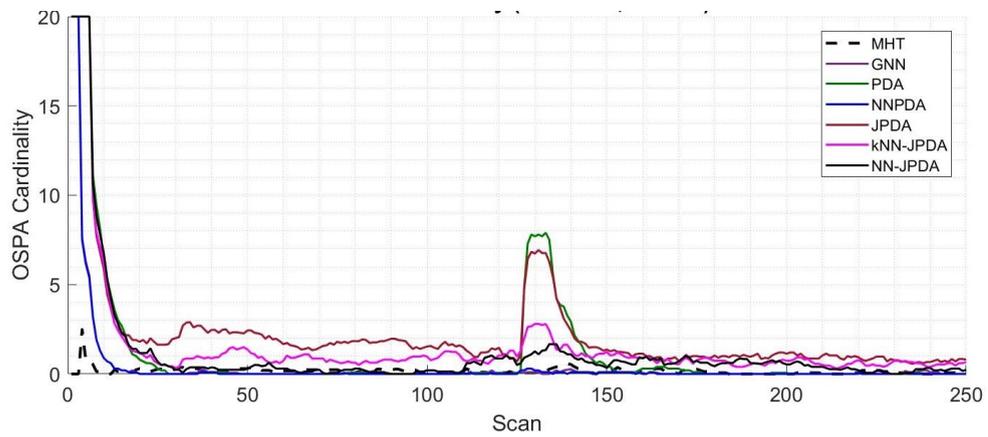


FIGURE 5.18: OSPA cardinality component for scenario 4.

## Chapter 6

# Conclusion

In this thesis, we investigated several methods of data association and target tracking for automotive applications. We proposed an efficient MHT method that used inner gating to reduce the computational complexity of track-oriented MHT. Inner gating limits the number of track branches that are formed and given to the hypothesis formation logic. We have described the implementation of the proposed track-oriented MHT along with a detailed comparison with other single hypothesis trackers, namely SNN, GNN, PDA, JPDA and NN-PDA.

The trackers were tested for four different MTT road scenarios in presence of random clutter and detection uncertainty. Three of the scenarios had linearly moving targets with different trajectories: sparse, crossing and closely moving targets, and were tracked using the constant velocity - linear Kalman filter motion model. The fourth trajectory was modelled on coordinated turn rate and velocity - extended Kalman filter model, and had two maneuvering targets with a close turn. Monte Carlo simulations were performed for each scenario to get the OSPA performance metric for all the trackers.

All of the data association algorithms were shown to have almost the same estimation accuracy when tracking the sparse targets. In the other three cases, PDA and SNN had the worst OSPA. This is because SNN is not good at resolving gating conflicts and is more likely to make incorrect assignment. Whereas PDA suffers from track coalescence, since it tends to update close targets with same set of measurements. NN-PDA and MHT are better at tracking crossing and closely spaced targets. But in case of close parallel targets, use of inner gating in MHT can degrade the overall performance. This can be improved by decreasing the inner gating threshold, which will in turn increase the computation time. Hence, there is a trade-off between run-time and accuracy in this case. For the other two scenarios, i.e crossing and maneuvering targets, the proposed MHT gives the best estimation accuracy.

Hence for near ideal automotive situations when the cars/vehicles are widely spaced, any of the conventional tracker can be used reliably. But in case of crossing cars, say at a road intersection or round-about, PDA, JPDA and SNN fails. In these situations MHT and NN-PDA is shown to work best. The most complex scenario to track is a densely populated road, when vehicles are moving in tight parallel lanes. For best results from MHT, the size of inner gating should be carefully selected.

## 6.1 Future Work

In this work, we have assumed that each target in the sensor FOV are point objects and can result in at-most one detection. The work can be extended to include a more practical extended target model, in which same target can result in multiple detections. An average ADAS system uses multiple on-board sensors to increase the detection probability and enlarge its FOV. The proposed MHT can be implemented for a multiple-sensor multiple-target (MSMT) scenario. Also, an interacting multiple model (IMM) algorithm can be used to efficiently model the maneuvering targets with changing state models.

# Bibliography

- [1] S. A. Glass. The features of adas, your driver assist system. [Online]. Available: <https://www.safelite.com/windshield-auto-glass-technology/adas>
- [2] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.
- [3] V. Naidu, G. Giriya, and J. Raol, "Evaluation of data association and fusion algorithms for tracking in the presence of measurement loss," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003, p. 5733.
- [4] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems (artech house radar library)," *Artech house*, 1999.
- [5] S.-l. Chen and Y.-b. Xu, "A new joint possibility data association algorithm avoiding track coalescence," *International Journal of Intelligent Systems and Applications*, vol. 3, no. 2, pp. 45–51, 2011.
- [6] R. J. Fitzgerald, "Development of practical pda logic for multitarget tracking," *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom (Ed.), Norwood. MA: Artech House, pp. 1–23, 1990.
- [7] S. Y. Bar and T. Fortmann, "Tracking and data association," Ph.D. dissertation, Academic Press Cambridge, 1988.
- [8] D. Reid *et al.*, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [9] H. W. Sorenson, "Least-squares estimation: from gauss to kalman," *IEEE spectrum*, vol. 7, no. 7, pp. 63–68, 1970.
- [10] A. Gelb, *Applied optimal estimation*. MIT press, 1974.
- [11] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, 2004.

- 
- [12] E. Brookner, *Tracking and Kalman filtering made easy*. Wiley New York, 1998.
- [13] R. G. Brown, P. Y. Hwang *et al.*, *Introduction to random signals and applied Kalman filtering*. Wiley New York, 1992, vol. 3.
- [14] M. Roth, G. Hendeby, and F. Gustafsson, “EKF/UKF maneuvering target tracking using coordinated turn models with polar/cartesian velocity,” in *Information Fusion (FUSION), 2014 17th International Conference on*. IEEE, 2014, pp. 1–8.
- [15] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. YBs London, UK:, 1995, vol. 19.
- [16] B.-S. Yaakov, D. Fred, and H. Jim, “The probabilistic data association filter: estimation in the presence of measurement origin un-certainty,” *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.
- [17] H. A. Blom and E. A. Bloem, “Probabilistic data association avoiding track coalescence,” *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 247–259, 2000.
- [18] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.
- [19] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.