EVALUATION OF MULTI TARGET TRACKING ALGORITHMS

IN THE PRESENCE OF CLUTTER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR GÜNER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Mustafa Kuzuoğlu
Supervisor

Examining Committee Members

Prof. Dr. Mübeccel Demirekler (METU, EE) ______________________

Prof. Dr. Mustafa Kuzuoğlu (METU, EE) ______________________

Prof. Dr. Kemal Leblebicioğlu (METU, EE) ______________________

Prof. Dr. Gönül Turhan Sayan (METU, EE) ______________________

Ms. Elif Yavuztürk (ASELSAN, MST) ______________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name	: Onur GÜNER
Signature	:

ABSTRACT

EVALUATION OF MULTI TARGET TRACKING ALGORITHMS IN THE PRESENCE OF CLUTTER

Güner, Onur

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Mustafa Kuzuoğlu

August 2005, 88 Pages

This thesis describes the theoretical bases, implementation and testing of a multi target tracking approach in radar applications. The main concern in this thesis is the evaluation of the performance of tracking algorithms in the presence of false alarms due to clutter. Multi target tracking algorithms are composed of three main parts: track initiation, data association and estimation. Two methods are proposed for track initiation in this work. First one is the track score function followed by a threshold comparison and the second one is the 2/2 & M/N method which is based on the number of detections. For data association problem, several algorithms are developed according to the environment and number of tracks that are of interest. The simplest method for data association is the nearest-neighbor data association technique. In addition, the methods that use multiple hypotheses like probabilistic data association and joint probabilistic data association are introduced and investigated. Moreover, in the observation to track assignment, gating is an important issue since it reduces the complexity of the computations. Generally, ellipsoidal gates are used for this purpose. For estimation, Kalman filters are used for state prediction and measurement update. In filtering, target kinematics is an important point for the modeling. Therefore, Kalman filters based on different target kinematic models are run in parallel and the outputs of filters are combined to yield a single solution. This method is developed for maneuvering targets and is called interactive multiple modeling (IMM).

All these algorithms are integrated to form a multi target tracker that works in the presence (or absence) of clutter. Track score function, joint probabilistic data association (JPDAF) and interactive multiple model filtering are used for this purpose.

Keywords: clutter, false alarms, track initiation, data association, gating, target kinematics, IMM, JPDAF

ÖZ

PARAZİTLİ ORTAMLARDA ÇOKLU HEDEF TAKİP ALGORİTMALARININ DEĞERLENDİRİLMESİ

Güner, Onur

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof.Dr. Mustafa Kuzuoğlu

Ağustos 2005, 88 Sayfa

Bu tez, radar uygulamalarında çoklu hedef takibi yaklaşımlarının kuramsal taban, uygulama ve test etme aşamalarını anlatmaktadır. Bu tezdeki ana kavram parazitli ortamlardan kaynaklanan sahte hedeflerin bulunduğu durumlarda takip algoritmalarının performanslarının değerlendirilmesidir. Çoklu hedef takip algoritmaları üç ana başlıktan oluşmaktadır: iz başlatma, veri ilişkilendirme ve tahmin. İz başlatma için iki yöntem sunulmuştur. Birincisi, belirli bir seviye ile karşılaştırmanın yapıldığı iz skor fonksiyonu, ikincisi ise tespit sayısına dayanan 2/2 & M/N yöntemidir. Veri ilişkilendirme problemi için çevre koşullarına ve hedef sayısına göre birtakım algoritmalar geliştirilmiştir. Veri ilişkilendirme yöntemleri içinde en basit olanı en yakın komşu veri ilişkilendirme yöntemidir. Bunun yanında, çoklu hipotez kullanan olasılıksal veri ilişkilendirme ve ortak olasılıksal veri ilişkilendirme yöntemleri sunulmuş ve incelenmiştir. Bunlara ek olarak ölçümlerin izlerle ilişkilendirilmesinde, hesaplamalardaki zorlukları azaltması dolayısıyla kapı yöntemi önemli bir konudur. Genellikle, bunun için eliptik kapılar kullanılmaktadır. Tahminler için ise durum kestirimi ve ölçüm güncellemesi yapan Kalman filtreleri

kullanılmıştır. Filtrelemede, hedef dinamikleri filtrenin doğru modellenmesi için önemli bir noktadır. Bundan dolayı, çeşitli hedef dinamik modellerine dayanan Kalman filtreleri paralel çalıştırılmış ve filtrelerin çıktıları tek bir sonuca ulaşmak için birleştirilmiştir. Bu yöntem manevra yapan hedefler için geliştirilmiştir ve etkileşimli çoklu model (IMM) olarak adlandırılır.

Tüm bu algoritmalar parazitli veya parazitsiz ortamlarda çoklu hedef takibi yapabilmek için birleştirilmiştir. Bu amaç doğrultusunda iz skor fonksiyonu, ortak olasılıksal veri ilişkilendirme (JPDAF) ve etkileşimli çoklu model algoritması kullanılmıştır.

Anahtar Kelimeler: parazit, sahte hedef, iz başlatma, veri ilişkilendirme, kapı, hedef dinamikleri, IMM, JPDAF

To My Parents

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Prof. Dr. Mustafa Kuzuoğlu for his continued guidance, also patience and support throughout my research. He provided counsel, and assistance that greatly enhanced my studies and know - how.

I am also grateful to Bahar Şahin for her support, patience and belief in me. She consistently motivated me during the whole master's period.

Thanks go to my manager Dr. Murat Eren for his patience and support from beginning to end of my M.Sc. program. I am also grateful to several of my colleagues, especially those who shared my office and thereby my problems. These include Naci Orhan, Emre Özkan, Salim Sırtkaya, Alper Öztürk, Burak Seymen and Mehmet Altuntaş.

And my parents and brother, I want to thank them for everything.

ASELSAN A.Ş. who supported this work is greatly acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CA	Constant Acceleration
CT	Coordinated Turn
CV	Constant Velocity
EKF	Extended Kalman Filter
IMM	Interactive Multiple Modeling
IMMPDAF	Interactive Multiple Model Probabilistic Data Association Filter
IMMJPDAF	Interactive Multiple Model Joint Probabilistic Data Association Filter
JPDA	Joint Probabilistic Data Association
KF	Kalman Filter
NN	Nearest Neighbor
ned	North East Down
PDA	Probabilistic Data Association
P_D	Probability of Detection
P_{FA}	Probability of False Alarm

CHAPTER 1

INTRODUCTION

1.1 Multi Target Tracking

In many applications, from air traffic control to ballistic missile defense there is the problem of tracking a multitude of targets while they are moving in space. Multi target tracking deals with the state estimation of an unknown number of moving targets seen from one or more sensors such as radar, that generates measurements of their position in space at every time step. The problem is difficult because the dynamics of the targets are uncertain and the sensors do not provide any identification of the source of the measurements. Because of the difficulties involved in the multi target tracking problem, a wide range of methods have been developed to handle the tasks stated below.

- Track initiation,
- Data association,
- Improved estimation methods

Methods for handling track initiation and data association often have much in common. Historically, approaches to this problem can be classified as Bayesian and non-Bayesian.

Initially, the attention is given to the non-Bayesian methods. Following the pioneering work of Sittler in 1964, these methods include tracking via data association, track split filtering and the maximum likelihood method. In tracking via data association Sittler [18] proposed a method where the current target was split when a measurement is observed in the neighborhood of a predicted measurement from more than one sensor. Trajectories whose maximum likelihood functions fall below a certain threshold are dropped from further consideration. This method handles both track initiation and track termination, as well as false alarms and missing measurements.

In 1975, Smith and Buechler [19] expanded Sittler's approach including Kalman filtering which was not common in use in 1964. The concept of neighborhood in Sittler's work is now a validation region, which is derived from the innovation covariance matrix obtained from the Kalman filter. On the other hand, these two methods are impractical because the growth in the number of trajectories will saturate the memory of even the largest computational capability computers.

Stein and Blackman [20] further modernized Sittler's work in the development of the maximum likelihood method. They select only the most likely assignments of targets and measurements from each data set or scan, thereby avoiding trajectory growth problem. Morefield [21] extended this approach by dividing the data into mutually exclusive and exhaustive sets of feasible tracks. While this work put track initiation and maintenance on a more solid theoretical basis, it still has large computational and memory requirements in a dense target environment.

Initial work with Bayesian approaches begin with the nearest neighbor filter. Sea [22], Singer and Stein [23], Singer and Sea [24] used the nearest neighbor of a predicted measurement and modified Kalman filter for a priori probability that this measurement might be originated from the target of interest [25]. However, it was understood that this filter could easily lose the target in a cluttered environment. The suboptimal Bayesian approach, the probabilistic data association filter (PDAF) is developed by Bar-Shalom [27], Bar-Shalom and Tse [26][28] and Bar-Shalom and Birmiwal [29]. The PDAF approach incorporates groups of measurements into a track by giving to each group an a posteriori probability of being correct. The estimates and covariances in the PDAF account for the measurement origin uncertainty. The primary limitation of the PDAF method is that it only tracks a single target in a multiple target or cluttered environment. More recently, in order to track multiple targets in a cluttered environment joint probabilistic data association filter

(JPDAF) technique is proposed by Fortmann, Bar-Shalom and Scheffé [30], Chang and Bar-Shalom [14], and Chang, Chong and Bar-Shalom [31]. This method is used to jointly compute the probabilities for all targets and measurements that form a cluster. In the JPDAF approach, posterior probabilities of the joint probability distribution function are conditioned on all measurements up to the present, allowing multiple targets to be tracked in a cluttered environment.

Another concern in the multitarget tracking problem is the estimation techniques. In order to estimate the state of a target, a target motion model and sensors are used. Sensor data and motion model are fused in a Kalman filter into a state estimate. In addition, for a good estimation the process noise of the Kalman filter have to be well adjusted. Since the motion of a target switches from one to other during a time interval, it is difficult to have good results with a Kalman filter. For this reason, algorithms for hybrid state estimate consists of a bank of model matched filter modules. The modules can be Kalman filters. Although non switching multiple model estimation algorithms have been known since the mid 1960s, practical algorithms for switching case have only recently become available [4]. One of the significant approaches is the Generalized Pseudo Bayesian (GPB) method [2], [32]. This algorithm consists of a bank of filters for the state cooperating with a filter for the parameters [4]. Another approach is the Interactive Multiple Modeling (IMM) [33]. The IMM is similar to the GPB method but it has notably less computations. Its complexity is nearly linear as a function of the size of the problem. Another advantage is its modularity since it can be set up using KF or EKF. In this thesis, methods are examined and algorithms are implemented for solving multi target tracking and data association problems in the presence of clutter.

1.2 Objectives

In this thesis, two methods are proposed for data association and target tracking in a cluttered environment. The first algorithm is based on interactive multiple model probabilistic data association filter (IMMPDAF). Since the sensors can generate measurements supposedly interpreted as target positions in the presence of clutter, it is not trivial to associate measurements to targets. The nearest neighbor method fails in the case of high clutter density, when there is more than one target and when the targets intersect. On the other hand, the PDAF yields a kind of average target trajectory taking into consideration all the validated data points and updates the filter. This method also suffers when the validation regions overlap but can track one target in a cluttered environment. This data association technique is combined with the IMM method since the target dynamics can vary along its trajectory. In the IMM method, the target kinematics that are considered, are namely the constant velocity and constant acceleration models. These models are implemented in the Kalman filters, which are the modules of the IMM algorithm to improve the state predictions. This algorithm is run under several scenarios and as expected, it proves to be successful in the tracking of a single target in the presence of false alarms, while it does not exhibit the same success when two targets intersect in a cluttered environment.

The second method implemented in this thesis is the interactive multiple model joint probabilistic data association filter (IMMJPDAF). Again, the IMM algorithm is used for state prediction of the targets, which have multiple dynamics. The data association technique JPDAF is an extension of the probabilistic data association method. The JPDAF evaluates the probabilities of all possible associations and combines them accordingly in order to compute the updated estimate of the target position. This algorithm is also tested under several scenarios to see how it differs in performance from the IMMPDAF algorithm.

In this thesis, the implemented algorithms mentioned above are used in several scenarios to examine the reliability and robustness of these methods in the problem of multi target tracking in a cluttered environment.

1.3 Thesis Outline

The thesis is concerned with the problem of multi target tracking in cluttered environments.

In **Chapter 2**, the necessary theoretical bases for track initiation, gating and data association are provided. Using the information extracted from these approaches, the decision whether the target is a confirmed track is made and the computational complexity of tracking algorithms is reduced.

The necessary theoretical bases for multi target tracking are explained in **Chapter 3**. The tracking coordinates are described for making the measurements and maintaining the target tracks. In addition, the target kinematic models such as constant velocity, constant acceleration, and coordinated turn models are explained. Using these kinematic models, one can design the tracking filters and determine the filter parameters. Finally, for the estimation purposes Kalman filter, PDAF, JPDAF are discussed to give an understanding of the methods used in the algorithms of multi target tracking.

After the theoretical derivations given in Chapters 2 and 3, **Chapter 4** provides simulation results for multi target tracking algorithms used to track various numbers of targets in a cluttered environment. The algorithms are applied to scenarios in order to see if all of the targets can be tracked without a loss. The algorithms are tested for different cases and for different environment conditions (clutter, false alarms) to analyze reliability and robustness.

Chapter 5 provides the summary for the overall study as well as some concluding remarks and some open points for possible future studies.

CHAPTER 2

GATING AND DATA ASSOCIATION

2.1 INTRODUCTION

In a multi-target situation, the measurements may have originated from one of several targets and the number of such measurements is generally unknown. The number of targets and their trajectories are generally a priori unknown. In order to start a tracking algorithm one has to determine the target, which has the possibility of being a true target. This is usually done with a track initiation algorithm. The track initiation creates true tracks from target measurements and false tracks from clutter.

In the course of track evolution and maintenance, true tracks may become false due to lack of detections, clutter measurements, target maneuvers which the track filter could not follow, etc. Equivalently, false tracks may become true tracks if true target measurements begin to accumulate. For practical purposes, target tracking requires a mechanism to distinguish between true and false tracks, without a-priori knowledge of target existence and trajectory [8]. When a track is considered as a true track, it is confirmed and presented to the operator or next processing level. When a track is considered as a false track, it is terminated.

Data association algorithms deal with situations where there are measurements of uncertain origin. In many radar and sonar applications, measurements (detections) originate from both targets and non-targets, i.e. from various objects such as terrain, clouds etc and from thermal noise [8]. Unwanted measurements are usually referred to as clutter. In addition, target measurements are present at each measurement scan with only a certain probability of detection.

As mentioned before, track initiation methods try to calculate if the target is worth tracking. In this chapter following [2] and [1] cascaded logic 2/2 & M/N and track score function approaches are proposed for track initiation algorithms. In addition, gating is introduced which is required to reduce further unnecessary computations for unlikely observation to track assignments. As a possible data association technique, nearest neighbor (NN) approach is proposed in section 2.5.1. In this approach, the goal is to find the single most likely hypotheses for the assignment of observations to existing tracks and to new track initiation. Implementation of NN requires the solution of the assignment problem [1]. In Sections 2.5.2 and 2.5.3, probabilistic data association and joint probabilistic data association methods are introduced. These approaches form multiple hypotheses after each scan of data and then these hypotheses are combined before the next scan of data is processed.

2.2 TRACK INITIATION

Track initiation is one of the most important issues for target tracking. The problem becomes complex due to the existence of false alarms caused by the clutter or other kinds of effects. Because of these false alarms, there occurs wrong tracks and this causes unwanted situations. In this section, two track initiation methods are going to be explained. First, the 2/2 & M/N approach proposed by Bar – Shalom is considered. Second, the track score method proposed by Blackman and Popoli is given. In the tracking algorithms for the simulations in this thesis, second method is used for track initiation, determination and deletion. In addition, for state prediction and measurement update, the Kalman filter is used for the track initiation algorithms and the initialization of the filter is given in Section 2.3.

2.2.1 2/2 & M/N Track Initiation Method

In this method, in the first two measurements there must be detection of the target. Following the first two scans, from out of N scans there must be M detections. At the end of the track initiation method, the probability of starting a true target tracking decreases. However, at the same time the possibility of starting a wrong target track due to false alarms decreases. The numbers M and N are chosen according to the radar parameters. For tracking of targets that are close to the radar system, big numbers should not be selected. Probability calculation of the track initiation for M=2 and N=3 is given in Table 2-1.

Table 2-1 Track Initiation Probability Calculation

<table>
<tr><th>State</th><th>Sequence</th><th>Mode</th><th>Remark</th><th>Prob.</th><td rowspan="7">Track Initiation Probability

(S8a+S8b+S8c)</td></tr>
<tr><td>S1</td><td>Initial State</td><td>Search</td><td></td><td></td></tr>
<tr><td>S2</td><td>1</td><td>Track</td><td>Candidate Track</td><td>$P_d = 0.9$</td></tr>
<tr><td>S3</td><td>1 1</td><td>Track</td><td>Tentative Track</td><td>$P_d^{\,2} = 0.81$</td></tr>
<tr><td>S4</td><td>1 1 0</td><td>Track</td><td></td><td>$P_d^{\,2}(1-P_d) = 0.081$</td></tr>
<tr><td>S5</td><td>1 1 1</td><td>Track</td><td></td><td>$P_d^{\,3} = 0.729$</td></tr>
<tr><td>S6</td><td>1 1 1 0</td><td>Track</td><td></td><td>$P_d^{\,3}(1-P_d) = 0.0729$</td></tr>
<tr><td>S7</td><td>1 1 0 1</td><td>Track</td><td></td><td>$P_d^{\,3}(1-P_d) = 0.0729$</td><td rowspan="4">0.7873</td></tr>
<tr><td>S8a</td><td>1 1 1 1</td><td>Track</td><td>Confirmed Track</td><td>$P_d^{\,4} = 0.6561$</td></tr>
<tr><td>S8b</td><td>1 1 1 0 1</td><td>Track</td><td>Confirmed Track</td><td>$P_d^{\,4}(1-P_d) = 0.0656$</td></tr>
<tr><td>S8c</td><td>1 1 0 1 1</td><td>Track</td><td>Confirmed Track</td><td>$P_d^{\,4}(1-P_d) = 0.0656$</td></tr>
</table>

As seen above in the first scan the track initiation probability is 0.9 but after the algorithm is proceeded, probability becomes approximately 0.79. With the decrease in the detection probability, the track initiation probability according to the false alarms also decreases. For instance, for a false alarm of probability $P_{FA} = 10^{-6}$ the detection probability becomes approximately 10^{-12}. This is greater than the detection probability decrease of the true targets.

When one of the conditions in state eight are satisfied, the algorithms used for multi target tracking are taken into account and tracking continues with those algorithms. There can be situations that the measurements from the target can not be obtained. In these conditions, the algorithm proceeds according to the predictions of the Kalman filter. This is done for several scans and the target is deleted afterwards.

2.2.2 Track Score Function

A true target is most generally defined to be an object that will persist in the tracking volume for at least several scans. Thus, the persistent clutter that is not the interest of the tracking system should be tracked in order to minimize their interference with the targets of interest [1].

In the track score function method, initially every candidate track has a score and at each scan, the scores of the candidate tracks are updated according to a recursive equation. The track score (L) for the confirmation and deletion of a track is tested versus upper and lower thresholds T_2 and T_1 respectively. The alternatives for track confirmation, track deletion, or continuing the test are defined as follows

$$\begin{aligned} L \geq T_2\,; &\quad \textit{declare track confirmation} \\ T_1 < L < T_2\,; &\quad \textit{continue test} \\ L \leq T_1\,; &\quad \textit{delete track} \end{aligned} \tag{2-1}$$

The thresholds are defined as

$$T_2 = \ln\left[\frac{1-\beta}{\alpha}\right], \qquad T_1 = \ln\left[\frac{\beta}{1-\alpha}\right] \tag{2-2}$$

Here α is defined as the false track confirmation probability and β as the true track deletion probability. The false track confirmation probability can be calculated from system requirements. Assume that the system produces N_{FA} false alarms per second and there are N_{FC} false track confirmations allowed per hour.

Then, α , the allowable probability that any one of these false alarms will produce a false track, is

$$\alpha = \frac{N_{FC}}{3600\, N_{FA}} \tag{2-3}$$

Because β has less effect on the track confirmation threshold, its choice is less important but a small value, $\beta \leq 0.1$, can be used for computation of T_2 [1].

The recursive track score function is given as

$$L(k) = L(k-1) + \Delta L(k) \tag{2-4}$$

where

$$\Delta L(k) = \begin{cases} \ln[1 - P_D]; & \textit{no track update on scan } k \\ \Delta L_U(k); & \textit{track update on scan } k \end{cases} \tag{2-5}$$

The increment, ΔL_U , that occurs upon update, is the sum of the kinematics and signal related terms so that

$$\Delta L_U = \Delta L_K + \Delta L_S \tag{2-6}$$

$$\Delta L_K = \ln\left[\frac{V_C}{\sqrt{|S|}}\right] - \frac{\left[M \ln[2\pi] + d^2\right]}{2} \tag{2-7}$$

$$\Delta L_S = \ln[P_D / P_{FA}] \tag{2-8}$$

Note that in the equations above, M is the measurement dimension, S is the measurement residual covariance matrix, V_C is the validation gate volume and d^2 is the normalized distance for the measurement defined in terms of the measurement residual vector and covariance.

Another concern is the initial track score. The initial track score is entirely based on the first observation in the track [1]. Thus, there is no kinematics contribution to the initial track score and it is given by

$$L(1) = \ln[P_D / P_{FA}] \tag{2-9}$$

Note that the threshold values are chosen on the assumption that the initial track score is zero.

2.3 INITIALIZATION OF THE ESTIMATORS

The initial state and covariance of an estimation filter has to be determined in order to start the prediction and measurement update. At initialization, it is just as important that the covariance associated with the initial estimate reflect its accuracy. A state estimation filter is called consistent if its estimation errors are compatible with the filter calculated covariances [15]. The filter initialization derivations are denoted here by following [15].

According to the Bayesian model the true initial state is a random variable with a known mean – the initial estimate – and a given covariance matrix

$$x(0) \sim N[x(0/0), P(0/0)] \tag{2-10}$$

In simulations, generally the Bayesian model is used. However, since the initial state is a random variable, it differs in each simulation. Thus, more appealing method is used. First, the initial true state is chosen and than the initial estimate is generated according to

$$x(0/0) \sim N[x(0), P(0/0)] \tag{2-11}$$

The scenario becomes fixed and the initial condition of the filter becomes random by applying this model to filter initialization.

A practical implementation in tracking can be done as follows. Use the last measurement as initial position estimate, obtain the velocity estimate by differencing the measurements and calculate the corresponding covariance matrix.

Consider two state components, say, position ξ and velocity $\dot{\xi}$ in a given coordinate system. If only position measurements $z(k)$ are available, then for true values $\xi(k), k = -1, 0,$ one generates the corresponding measurement noises

$$w(k) \sim N[0, R] \tag{2-12}$$

Then denoting by T the sampling interval one has

$$\xi(0/0) = z(0) \tag{2-13}$$

$$\dot{\xi}(0/0) = \frac{z(0) - z(-1)}{T} \tag{2-14}$$

And the corresponding 2x2 block of the initial covariance matrix is then

$$P(0/0) = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix} \tag{2-15}$$

This method guarantees consistency of the initialization of the filter, which starts updating the state at k=1 [15].

2.4 GATING

Gating is a technique for eliminating unlikely observations to track assignments. A gate is formed about the predicted measurement and all observations that satisfy the gating relationship are considered for track update [1]. The observations that are used for the track update are determined according to the data association method but most data association methods use gating in order to reduce the computational complexity. There are several gating techniques but in this section, the rectangular and ellipsoidal gates are going to be examined.

2.4.1 Rectangular Gates

Probably the simplest gating technique is to define rectangular gates. Observations are said to satisfy the gates of a given track if the residual vector satisfy the relationship below.

$$|y - \hat{y}| \leq K_G \sigma_r \qquad (2\text{-}16)$$

where σ_r is the residual standard deviation as defined in terms of the measurement ($\sigma_o{}^2$) and the prediction ($\sigma_p{}^2$) variances:

$$\sigma_r = \sqrt{\sigma_o{}^2 + \sigma_p{}^2} \qquad (2\text{-}17)$$

Choice of the gating coefficients K_G in practice typically will be $K_G \geq 3.0$. The large choice of the gating coefficient is made in order to compensate for the approximations involved in modeling the target dynamics through the Kalman filter covariance matrix [1].

2.4.2 Ellipsoidal Gates

Ellipsoidal gates are defined according to the residual vector and covariance that is obtained from the Kalman filter. At each scan, the normalized distance is calculated and compared with a gate. The gate is defined such that the association is allowed if the following relationship is satisfied by the normalized distance.

$$d^2 = \hat{y}' S^{-1} \hat{y} \leq \gamma \qquad (2\text{-}18)$$

Usually the d^2 is assumed to have chi-square distributed with number of degrees of freedom equal to the dimension of the measurements [2]. Thus, the threshold γ is obtained from the tables of the chi-square distribution.

Finally the volume within the ellipsoidal gate is

$$V = c_{n_z} |\gamma S|^{1/2} \qquad (2\text{-}19)$$

Note that n_z is the dimension of measurement and c_{n_z} is the volume of the unit sphere of this dimension.

2.5 DATA ASSOCIATION

In multi target tracking, data association is a fundamental issue. Data association deals with the uncertainty of which observation goes with a track or observation from a prior frame of data [9]. Data association updates tracks with new observations in some optimal manner so that accurate track estimation can be performed [10]. In this section the most important data association techniques namely nearest neighbor (NN) data association, probabilistic data association (PDA) and joint probabilistic data association (JPDA) are introduced.

In the NN method, a track can be updated by one observation at most, and non-conflict observation-to-track pairing is performed by minimizing the sum of distances from all the observations to the tracks to which they are assigned.' This is essentially an assignment problem. The PDA method, on the other hand, updates a track with a weighted average of all the observations within the validation gate. To account for the possibility of two or more targets in the same gate, the joint probabilistic data association was developed [10].

2.5.1 Nearest Neighbor (NN)

The NN method is the simplest and probably the most widely applied method for data association. This method may also be referred to as single hypotheses tracking or sequential most probable hypotheses tracking [1].

In the NN data association, a track can be updated by one observation at most and an observation can be assigned to one track at most. Typically, the first step for the NN data association is to form an assignment matrix [10]. These assignment matrix elements are the normalized distance function that is defined by the measurement residual vector and covariance.

In order to assign observations to tracks the method based on searching the assignment matrix for the observation-to-track pair with the minimum distance and making the assignment. The next step is removing the row and column that corresponds to the observation-to-track pair assigned from the assignment matrix and repeating searching for the reduced matrix.

2.5.2 Probabilistic Data Association (PDA)

The probabilistic data association algorithm calculates the association probabilities for each validated measurement at the current time to the target of interest [2]. This information is then used for the tracking filter.

Following [10], given N observations, $\{y_j(k), j = 1,...,N\}$, within the validation gate of track i at scan k, the probabilistic data association forms $N+1$ hypotheses H_j, where H_0, is the case in which none of the N observations is originated from track i, and H_j $j > 0$ denotes the hypotheses that j[th] observation is originated from track i. Assuming a Poisson distribution for the clutter with spatial density λ, the probability of hypotheses H_j is

$$p_{ij} = \frac{a_{ij}}{a_{i0} + \sum_{j=1}^{N} a_{ij}} \tag{2-20}$$

where

$$a_{i0} = (2\pi)^{M/2} \lambda \sqrt{|S_i(k)|}(1 - P_D) \tag{2-21}$$

$$a_{ij} = P_D \exp\left\{-\frac{1}{2} v_{ij}{}'(k) S_i^{-1}(k) v_{ij}(k)\right\} \tag{2-22}$$

These probabilities are used for innovation combination in order to manipulate in the filter state and covariance update equations.

2.5.3 Joint Probabilistic Data Association (JPDA)

In the probabilistic data association method, it is assumed that all the tracks are isolated and the observations that fall into the gate come from the target of interest. On the other hand, some observations may come from other targets. For this reason, the performance of the PDA may degrade significantly in the presence of multiple targets [10].

The joint probabilistic data association is concerned with this problem and includes the possibility of multiple target returns in a gate or one target return in multiple gates. The first step in JPDA is to construct all feasible joint events and the equations that give these considerations can be found in [10].

A feasible joint event at scan k denoted as $\theta(k)$ is a non-conflict association of N_T existing tracks with the N valid observations. Next, the conditional probability of each $\theta(k)$ is calculated. Assuming a Poisson density for clutter, the probability of the event $\theta(k)$ given the set of all the observations received up to current scan $k\,(Y^k)$ is given by

$$P\left[\theta(k)\middle|Y^k\right] = \frac{\lambda^n}{c} \prod_{j=1}^{N} \left(N\left[v_{i\,j}(k)\right]\right)^{\tau_j} \prod_{t=1}^{N_T} (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \qquad \text{(2-23)}$$

where

$$\tau_j = \begin{cases} 1, & observation\ j\ assigned\ to\ track\ i\ in\ the\ event\ \theta(k) \\ 0, & otherwise \end{cases} \qquad \text{(2-24)}$$

$$\delta_t = \begin{cases} 1, & t^{th}\ track\ has\ been\ assigned\ an\ observation\ in\ \theta(k) \\ 0, & otherwise \end{cases} \qquad \text{(2-25)}$$

Note that c is a normalization constant, n is the number of observations determined to be clutter in $\theta(k)$ and $N\left[v_{i\,j}(k)\right]$ denotes the normal probability density function with zero-mean and a covariance matrix equal to that of $v_{i\,j}(k)$.

The sum of the all probabilities of all joint events in which track t is associated with observation j is the probability of track t being associated with the observation j and is given by

$$p_{t\,j}(k) = \sum_{\theta(k)} P\left[\theta(k)\middle|Y^k\right] w_{t\,j}\left[\theta(k)\right] \qquad \text{(2-26)}$$

Here, $w_{tj}\left[\theta(k)\right]=1$ track t is associated with observation j in the event $\theta(k)$ and equals to zero otherwise.

This method provides an optimal data association solution. However, as a function of the number of targets and clutter points the hypotheses associating different returns to targets increases exponentially. Therefore, JPDA requires a prohibitive amount of processor time in order to calculate the probabilities [10].

CHAPTER 3

MULTI TARGET TRACKING

3.1 INTRODUCTION

This chapter discusses the choice of tracking coordinate and a class of widely used target models derived from simple equations of motion.

The tracking process requires a frame of reference in which to make measurements and to maintain target tracks. There are two natural ways to choose which coordinate system to use in a radar application [1]. The first possibility is to use the spherical coordinate system, with range, azimuth and elevation and the other alternative is a slightly modified Cartesian coordinate system, the north-east-down (ned) coordinate system. Both approaches have pros and cons, and which one of the two to use is not obvious. The non-linearities that appear in both the spherical and ned coordinate system do not provide an easy answer to what coordinate system to use. The deciding factor, in this case, turned out to be the easier motion description; hence, the ned coordinate system is used throughout this thesis.

Most tracking algorithms are model based because a good model-based tracking algorithm will greatly outperform any model-free tracking algorithm if the underlying model turns out to be a good one. Various mathematical models of target motion have been developed over the past three decades. Many of these models are useful for target detection and target tracking [5]. In this chapter, the underlying ideas and assumptions of these models are also given in order to help the reader understand how these models work and their pros and cons.

The most common target motion models in the Cartesian coordinate system are constant velocity, constant acceleration and coordinated turn. In order to track a target, models must match to the actual target dynamics. Therefore, a finite set of models are used in this work to handle a range of potential target dynamics.

The uncertainty in state estimation due to random target kinematics or mismodeling of target kinematics is represented by the process noise covariance matrix Q. In this work, target position (x), velocity (v_x) and acceleration (a_x) are used as states.

In addition, for target tracking, state estimation is another important issue. The state estimation problem consists of filtering and prediction of such quantities as target position, velocity and (if used) acceleration [1]. Target tracking is often conducted by the use of Kalman filters. This section will therefore introduce the Kalman filter, and the more general extended Kalman filter (EKF) which is used on non-linear models. Furthermore, a method to combine several filters is called interacting multiple models (IMM), and more powerful methods for multi target tracking such as probabilistic data association filter (PDAF) and joint probabilistic data association filter (JPDAF) are introduced.

3.2 COORDINATE SYSTEMS

3.2.1 Spherical coordinate system

The radar-based spherical coordinate system has the tempting feature that measurements returned from the radar are linear, avoiding to add non-linearities. On the other hand, very few target motions are described with linear functions in such a coordinate system, i.e., as long as the target is not using the radar as a pivot point for its maneuvers [1].

This approach also facilitates decoupling so that separate angle and range filters can be used. Finally, if a measured range rate is available it can be used directly in the range filter. There will be a range filter and, provided that azimuth and elevation angles are used there will also be two angle filters. The range filter is defined in a straightforward manner, but the choice of angle filters is more difficult.

3.2.2 Cartesian coordinate system

Tracking in Cartesian coordinates has the advantage of allowing the use of linear target dynamic models for extrapolation. However, since sensors measure angles a nonlinear transformation is required to relate the measurement to the Cartesian state variables. Therefore, there are two approaches for using these measurements to track in Cartesian coordinates.

Following the approach in [1], one method is to first transform the measurements to the Cartesian coordinates before feeding them to the tracking filter. The other method is to leave the measurements in the original form and input them through the use of a nonlinear measurement form of the Extended Kalman filter discussed later.

3.2.2.1 Measurement Transformation

The transformation between target range (R), azimuth angle (ϕ), and elevation angle (θ) and Cartesian position (x, y, z) is defined to be

$$x = R\cos\theta\cos\phi, y = R\cos\theta\sin\phi, z = R\sin\theta \tag{3-1}$$

Use of measurement transformation to Cartesian coordinates has several implications. First, an inaccurate range measurement may corrupt an accurate angle measurement or vice versa. Second, the transformed measurement errors are correlated. Thus, ideally a fully coupled state vector with x, y, z component position, velocity and (if used) acceleration, should be defined.

3.2.2.2 Nonlinear Measurement Input to EKF

Consider fully coupled filter that has all three position (x, y, z) and velocity (v_x, v_y, v_z) states with possible acceleration states as well. Then the measured range and angles can be input to tracking filter through the use of the nonlinear measurement

$$\bar{y} = h(\bar{x}) + \upsilon \tag{3-2}$$

Note that

$$\bar{x} = (x, y, z, v_x, v_y, v_z)^T, \; h(\bar{x}) = \begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} = \begin{bmatrix} R \\ \tan^{-1}\left[(y - y_s)/(x - x_s)\right] \\ \tan^{-1}\left[(z - z_s)/R\right] \end{bmatrix}$$

$$\text{and } R = \left[(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2\right]^{\frac{1}{2}}$$

In the above equations (x_s, y_s, z_s) is sensor position and taken as zero for simplicity. Then the elements of the measurement matrix are computed in the standard manner:

$$h_{ij} = \left.\frac{\partial h_i}{\partial x_j}\right|_{x=\hat{x}} \tag{3-3}$$

3.3 TARGET KINEMATICS

There are various target kinematics defined and examined in the literature [1], [5], [6], [7], [11]. Among these target kinematic models, constant velocity, constant acceleration and coordinated turn models are introduced in this section and the equations that describe the target kinematic models can be found in the related literature.

3.3.1 Constant Velocity (CV) Model

The most common definition of the constant velocity model is as follows. A constant velocity target moving in a coordinate ζ is described by the equation

$$\ddot{\zeta}(t) = 0 \tag{3-4}$$

In practice velocity can change slightly and can be modeled by zero mean white noise $\upsilon(t)$ as

$$\ddot{\zeta} = \upsilon(t) \tag{3-5}$$

Note that

$$E[\upsilon(t)] = 0, E[\upsilon(t)\upsilon(\tau)] = q(t)\delta(t-\tau)$$

The state vector for this model is two dimensional per coordinate and is given as

$$X = \begin{bmatrix} \zeta & \dot{\zeta} \end{bmatrix}'$$

The continuous time state equation is

$$\dot{X}(t) = AX(t) + D\upsilon(t) \tag{3-6}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ and } D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The corresponding discrete time state equation with sampling period T is

$$X(k+1) = FX(k) + \upsilon(k) \tag{3-7}$$

Here

$$F = e^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} S_\upsilon$$

Note that the S_υ is the power spectral density of the continuous time process noise.

A second version of this model is the direct discrete time model, which has a state equation given by

$$X(k+1) = FX(k) + \Gamma\upsilon(k) \tag{3-8}$$

where

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \text{ and } \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

From the above, the covariance of the discrete time process noise is

$$Q = E[\Gamma \upsilon(k) \upsilon(k)' \Gamma'] = \Gamma \sigma_\upsilon^2 \Gamma' = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_\upsilon^2$$

For this model, σ_υ should be of the order of the maximum acceleration magnitude a_M . A practical range is $0.5a_M \le \sigma_\upsilon \le a_M$. A constant velocity model is obtained by the small choice of intensity in the following sense: The changes in the velocity over a sampling time have to be small compared to the actual velocity.

The matrices F and Q are for one coordinate. In air traffic, three coordinates (x, y, z) must be considered. For this reason, the F and Q matrices that are used in the filter equations have a diagonal form, which is given by

$$F^1 = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & F \end{bmatrix} \text{ and } Q^1 = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix}$$

3.3.2 Constant Acceleration (CA) Model

A constant acceleration target moving in a coordinate ζ is described by the equation

$$\dddot{\zeta} = 0 \tag{3-9}$$

In practice, similarly to the constant velocity model, acceleration can change slightly and can be modeled by zero mean white noise $\upsilon(t)$ as

$$\dddot{\zeta} = \upsilon(t) \tag{3-10}$$

Note that

$$E[\upsilon(t)] = 0, E[\upsilon(t)\upsilon(\tau)] = q(t)\delta(t-\tau)$$

The state vector for this model is three dimensional per coordinate and given as

$$X = \begin{bmatrix} \zeta & \dot{\zeta} & \ddot{\zeta} \end{bmatrix}'$$

This model has two commonly used versions [5]. The first one, referred to as white-noise jerk model, assumes that the acceleration derivative is an independent (white noise) process.

The continuous time state equation is

$$\dot{X}(t) = AX(t) + D\upsilon(t) \tag{3-11}$$

Here

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The equivalent discrete time state equation with sampling period T is

$$X(k+1) = FX(k) + \upsilon(k) \tag{3-12}$$

where

$$F = e^{AT} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} S_\upsilon$$

Note that S_υ is the power spectral density, not the variance, of the continuous-time white noise.

The second version is called Wiener-sequence acceleration model. It assumes that the acceleration increment is an independent (white noise) process. The corresponding state-space representation is given below.

$$X(k+1) = FX(k) + \Gamma \upsilon(k) \tag{3-13}$$

Here

$$\Gamma = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix}$$

The covariance of the process noise is given by

$$Q = \begin{bmatrix} T^4/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2 & T \\ T^2/2 & T & 1 \end{bmatrix} \sigma_\upsilon^2$$

For this model, σ_υ should be of the order of the magnitude of the maximum acceleration increment over a sampling period, Δa_M. A practical range is $0.5\Delta a_M \le \sigma_\upsilon \le \Delta a_M$.

Here also the matrices used for the filter equations have a diagonal form as

$$F^1 = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & F \end{bmatrix} \text{ and } Q^1 = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix}$$

3.3.3 Coordinated Turn (CT) Model

In this model, motion is correlated across the tracking directions. Therefore, motion in the x, y and z directions are not independent. In the coordinated turn model the turn rate is assumed to be constant and given by

$$\omega = \frac{\|vxa\|}{v^2} \tag{3-14}$$

In addition, for a target with constant speed, the velocity and acceleration vectors are orthogonal and then the above equation for the magnitude of the turn rate simplifies to

$$\omega = \frac{\|a\|}{\|v\|} \tag{3-15}$$

The state-space form of this model for each Cartesian coordinate with state $X = \begin{bmatrix} \zeta & \dot{\zeta} & \ddot{\zeta} \end{bmatrix}'$ is given by

$$\dot{X}(t) = AX(t) + \upsilon(t) \tag{3-16}$$

In the equation above

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega^2 & 0 \end{bmatrix}$$

The corresponding discrete time model for each Cartesian coordinate is

$$X(k+1) = FX(k) + \Gamma\upsilon(k) \tag{3-17}$$

Here

$$F = \begin{bmatrix} 1 & \sin\omega T/\omega & (1-\cos\omega T)/\omega^2 \\ 0 & \cos\omega T & \sin\omega T/\omega \\ 0 & -\omega\sin\omega T & \cos\omega T \end{bmatrix} \text{ and } \Gamma = \begin{bmatrix} (\omega T - \sin\omega T)/\omega^3 \\ (1-\cos\omega T)/\omega^2 \\ \sin\omega T/\omega \end{bmatrix}$$

Considering the three Cartesian coordinates, the diagonal form used in the filters is as follows

$$F^{1} = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & F \end{bmatrix}$$

In order to use this model for state estimation, an estimate of ω is needed. This can be obtained from the latest velocity and acceleration estimates.

3.4 ESTIMATION

Estimation is the process of inferring the value of a quantity of interest from indirect, inaccurate and uncertain observations. When tracking is considered, the purpose of the estimation is the determination of the position, velocity and acceleration of a target.

Tracking is the estimation of the state of a moving object (target) based on remote measurements. This is done using one or more sensors at fixed locations or on moving platforms. At first sight, tracking can seem to be a special case of estimation. However, it is wider in scope. It uses all the tools from estimation and also requires use of statistical decision theory when some of the problems (data association) are considered. On the other hand, filtering is the estimation of the current state of dynamical system. In this manner, various filtering techniques with regard to multi target tracking are introduced. The state prediction and measurement update equations of the filters and also the probability calculations are based on the analysis in [2].

3.4.1 Kalman Filtering

The Kalman filter is described as a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method [12]. Kalman filtering allows the provision to model random target kinematics. Kalman filter will minimize the mean squared error as long as the target kinematics and the measurement noise are accurately modeled. In addition, Kalman filter has a number of other advantages for application to multi target tracking [1].

- The gain sequence is chosen automatically based on the assumed target maneuver measurement noise models. This means that the same filter can be used for varying target and measurement environments by changing a few parameters.

- The Kalman gain sequence automatically adapts to changing detection histories. This includes a varying sampling interval as well as missed detections.

- The Kalman filter provides a convenient measure of the estimation accuracy through the covariance matrix which is required to perform data association. Also having a measure of the expected prediction error variance is useful for maneuver detection and upon maneuver detection Kalman filter provides a way to adjust for varying target kinematics.

- Through use of the Kalman filter, it is possible to compensate for the effects of misassociation in the dense multi target tracking environment.

The Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x(k+1) = F(k)x(k) + \Gamma(k)v(k) \tag{3-18}$$

Note that $v(k)$ is the sequence of zero-mean white Gaussian process noise with covariance

$$E[v(k)v(k)'] = Q(k) \tag{3-19}$$

Then the covariance matrix ($Q(k)$) of the disturbance in the state equation is replaced by $\Gamma(k)Q(k)\Gamma(k)'$.

The measurement equation is

$$z(k) = H(k)x(k) + w(k) \tag{3-20}$$

Here $w(k)$ is the sequence of zero-mean white Gaussian measurement noise with covariance

$$E[w(k)w(k)'] = R(k) \tag{3-21}$$

The matrices F, Γ, H, Q, and R are assumed to be known and possibly time varying. The initial state, in general unknown, is modeled as a random variable, Gaussian-distributed with known mean and covariance. The two noise sequences and the initial state are assumed mutually independent.

The estimation algorithm starts with the initial estimate $x(0/0)$ of $x(0)$ and the associated initial covariance $P(0/0)$ assumed to be available. The Kalman filter algorithm consists of time-update and measurement-update equations. The time-update equations are one-step prediction of the state and its covariance.

3.4.1.1 Time Update Equations

$$x(k+1/k) = F(k)x(k/k) \tag{3-22}$$

$$P(k+1/k) = F(k)P(k/k)F(k)' + Q(k) \tag{3-23}$$

The predicted state and covariance is followed similarly by the prediction of the measurement

$$z(k+1/k) = H(k+1)x(k+1/k) \tag{3-24}$$

The covariance of the predicted measurement is

$$S(k+1) = H(k+1)P(k+1/k)H(k+1)' + R(k+1) \tag{3-25}$$

3.4.1.2 Measurement Update Equations

The gain of the filter is calculated as follows:

$$W(k+1) = P(k+1/k)H(k+1)'S(k+1)^{-1} \tag{3-26}$$

Thus, the updated state estimate can be written according to

$$x(k+1/k+1) = x(k+1/k) + W(k+1)\nu(k+1) \tag{3-27}$$

where

$$\nu(k+1) = z(k+1) - z(k+1/k) \tag{3-28}$$

is called the innovation or measurement residual. Also in the same sense, measurement prediction covariance can also be called innovation covariance.

Finally, the updated covariance of the state is as follows

$$P(k+1/k+1) = P(k+1/k) - W(k+1)S(k+1)W(k+1)' \tag{3-29}$$

Note from equation (3-26) that the filter gain is proportional to state prediction variance and inversely proportional to the innovation variance. Thus, the gain is large if the state prediction is inaccurate where the measurement is accurate. On the other hand, the gain is small if the state prediction is small and the measurement is inaccurate. A large gain indicates rapid response to the measurement in updating the state, while a small gain yields a slower response to the measurement.

The flow chart of one cycle of the Kalman filter is presented in Figure 3-1.

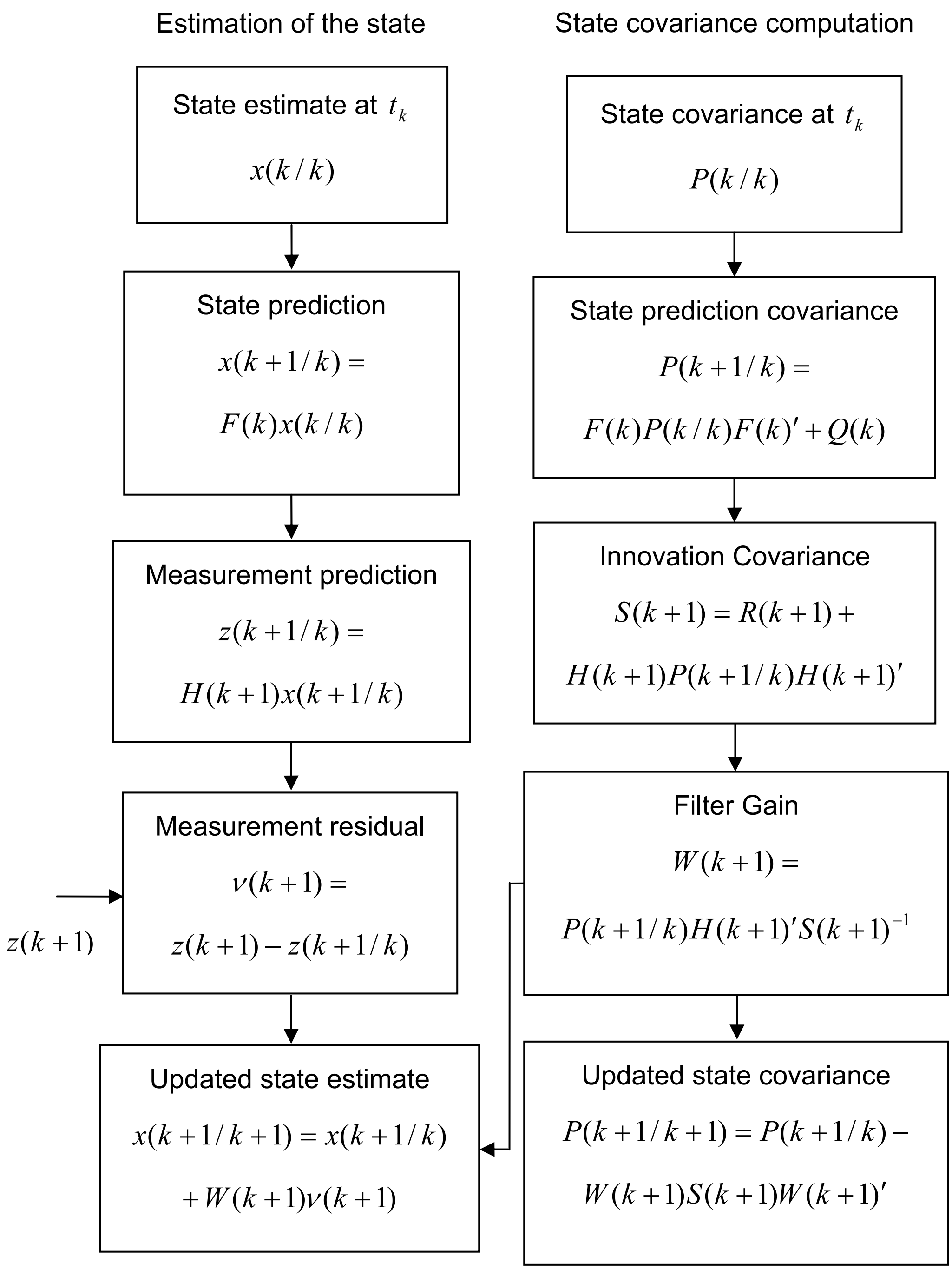

Figure 3-1 One Cycle of Kalman Filter Algorithm

3.4.2 Extended Kalman Filtering (EKF)

As described above, the Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete time controlled process that is governed by the linear stochastic difference equation. A Kalman filter that is obtained by a linearization about the current mean and covariance is referred to as an extended Kalman filter or EKF. The EKF is applicable under the conditions of a nonlinear measurement process and/or nonlinear target kinematics.

Consider the system with dynamics

$$x(k+1) = f[k, x(k)] + v(k) \tag{3-30}$$

where the noise is assumed to be additive, zero-mean and white

$$E[v(k)] = 0 \tag{3-31}$$

$$E[v(k)v(j)'] = Q(k)\delta_{kj} \tag{3-32}$$

The measurement is

$$z(k) = h[k, x(k)] + w(k) \tag{3-33}$$

where the measurement noise is also additive, zero-mean and white

$$E[w(k)] = 0 \tag{3-34}$$

$$E[w(k)w(j)'] = R(k)\delta_{kj} \tag{3-35}$$

Similar to the linear case, it will be assumed that one has the estimate at time k. The prediction of the state and covariance are same as the linear Kalman filter case. On the contrary, to obtain the predicted state the nonlinear function in equation (3-30) is expanded in a Taylor series around the latest estimate $x(k/k)$ and to obtain the predicted covariance the nonlinear function in equation (3-33) is expanded in Taylor series around the latest predicted state $x(k+1/k)$. Thus, the equations of the extended Kalman filter are as follows.

$$x(k+1/k) = F(k)x(k/k) \quad (3\text{-}36)$$

$$P(k+1/k) = F(k)P(k/k)F(k)' + Q(k) \quad (3\text{-}37)$$

$$z(k+1/k) = H(k+1)x(k+1/k) \quad (3\text{-}38)$$

$$S(k+1) = H(k+1)P(k+1/k)H(k+1)' + R(k+1) \quad (3\text{-}39)$$

$$W(k+1) = P(k+1/k)H(k+1)'S(k+1)^{-1} \quad (3\text{-}40)$$

$$x(k+1/k+1) = x(k+1/k) + W(k+1)\nu(k+1) \quad (3\text{-}41)$$

$$\nu(k+1) = z(k+1) - z(k+1/k) \quad (3\text{-}42)$$

$$P(k+1/k+1) = P(k+1/k) - W(k+1)S(k+1)W(k+1)' \quad (3\text{-}43)$$

Note that in the above equations the matrices F and H are the Jacobian of the functions f and h in (3-30) and (3-33) respectively.

$$F(k) = \left.\frac{\partial f(k)}{\partial x}\right|_{x=x(k/k)} \quad (3\text{-}44)$$

$$H(k+1) = \left.\frac{\partial h(k+1)}{\partial x}\right|_{x=x(k+1/k)} \quad (3\text{-}45)$$

As seen from the above equations, the main difference of the EKF from the Kalman filter is the evaluation of the Jacobians of the state transition and the measurement equations. Due to this, the covariance computations are not decoupled anymore from the state estimate calculations and can not be done offline.

The flow chart of one cycle of the Extended Kalman filter is presented in Figure 3-2.

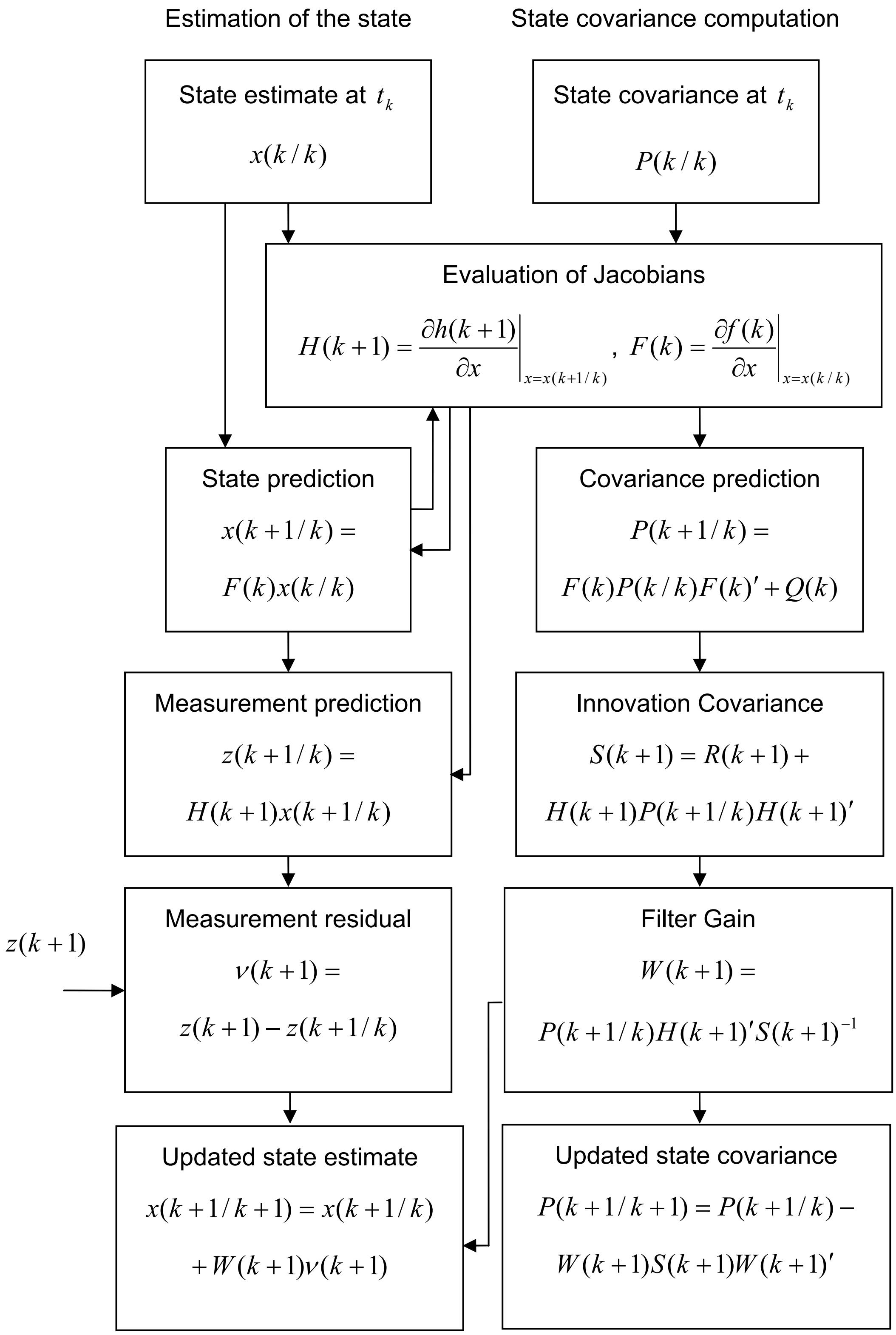

Figure 3-2 One Cycle of Extended Kalman Filter Algorithm

3.4.3 Probabilistic Data Association Filter (PDAF)

The PDA filter is based on the use of Probabilistic Data Association algorithm, which is the calculation of the association probabilities for each validated measurement at the current time to the target of interest [2]. Among the possibly several validated measurements, at most one of them can be target originated if the target has been detected and the corresponding measurement fell into the validation region. At each time, validation region is set up. The measurements that do not fall into the gate are assumed to be due to false alarm or clutter and are modeled as independent identically distributed with uniform spatial distribution.

The target detections occur independently over time and with known detection probability P_D. Also the probability that a correct target return will fall within the track gate is P_G.

Following [2] the PDAF uses a decomposition of the estimation with respect to the origin of each element of the latest set of measurements, denoted as

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \tag{3-46}$$

where $z_i(k)$ is the i[th] validated measurement and $m(k)$ is the number of measurements in the validation region at time k.

The cumulative set of measurements is

$$Z^k = \{Z(j)\}_{j=1}^{k} \tag{3-47}$$

The validation region is the elliptical region

$$V(k,\gamma) = \left\{ z : \left[z - z(k/k-1)\right]' S(k)^{-1} \left[z - z(k/k-1)\right] \leq \gamma \right\} \tag{3-48}$$

where γ is the gate threshold and

$$S(k) = H(k)P(k/k-1)H(k)' + R(k) \tag{3-49}$$

is the covariance of the innovation corresponding to the true measurement.

The volume of the validation region is

$$V(k) = c_{n_z} \left|\gamma S(k)\right|^{1/2} = c_{n_z} \gamma^{n_z/2} \left|S(k)\right|^{1/2} \tag{3-50}$$

Here the coefficient c_{n_z} depends on the dimension of the measurement (it is the volume of n_z dimensional unit sphere: $c_1 = 2, c_2 = \pi, c_3 = 4\pi/3, etc.$).

3.4.3.1 The State Estimation

The prediction equations are standard filter equations. The prediction of the state and covariance to $k+1$ is done as follows

$$x(k+1/k) = F(k)x(k/k) \tag{3-51}$$

$$z(k+1/k) = H(k+1)x(k+1/k) \tag{3-52}$$

Similarly the covariance of the predicted state is

$$P(k+1/k) = F(k)P(k/k)F(k)' + Q(k) \tag{3-53}$$

The innovation covariance for the correct measurement is, again as in the standard filter

$$S(k+1) = H(k+1)P(k+1/k)H(k+1)' + R(k+1) \tag{3-54}$$

The update equations are calculated according to the association probabilities. The association probability is due to the association events

$$\theta_i(k) = \begin{cases} z_i(k)\, is\, the\ t\arg et\ originated\ measurement,\ i = 1,\ldots,m(k) \\ none\, of\ the\, measurements\ is\ t\arg et\ originated,\ i = 0 \end{cases} \tag{3-55}$$

The conditional mean of the state at time k can be written as

$$x(k/k) = \sum_{i=0}^{m(k)} x_i(k/k)\beta_i(k) \tag{3-56}$$

where $x_i(k/k)$ is the updated state conditioned on the event that the i^th validated measurement is correct and

$$\beta_i(k) \cong P\{\theta_i(k) | Z^k\} \tag{3-57}$$

is the conditional probability of this event – the association probability, obtained from the PDA procedure as will be presented in the next subsection.

The estimate conditioned on measurement i being correct is

$$x_i(k/k) = x(k/k-1) + W(k)v_i(k) \qquad i = 1,\ldots,m(k+1) \tag{3-58}$$

Note that the corresponding innovation is

$$v_i(k) = z_i(k) - z(k/k-1) \tag{3-59}$$

The gain is the same as in the standard filter

$$W(k) = P(k/k-1)H(k)'S(k)^{-1} \tag{3-60}$$

For $i = 0$ if none of the measurements is correct, or, if there is no validated measurement one has

$$x_0(k/k) = x(k/k-1) \tag{3-61}$$

3.4.3.2 The State and Covariance Update

Combining (3-54) and (3-57) into (3-53) yields the state update equation of the PDAF as

$$x(k/k) = x(k/k-1) + W(k)v(k) \tag{3-62}$$

where the combined innovation is

$$v(k) = \sum_{i=1}^{m(k)} \beta_i(k)v_i(k) \tag{3-63}$$

The covariance associated with the updated state is

$$P(k/k) = \beta_0(k)P(k/k-1) + \left[1-\beta_0(k)\right]P^c(k/k) + \widetilde{P}(k) \tag{3-64}$$

where the covariance of the state updated with the correct measurement is

$$P^c(k/k) = P(k/k-1) - W(k)S(k)W(k)' \tag{3-65}$$

and the spread of the innovation term is

$$\widetilde{P}(k) = W(k)\left[\sum_{i=1}^{m(k)} \beta_i(k)v_i(k)v_i(k)' - v(k)v(k)'\right]W(k)' \tag{3-66}$$

3.4.3.3 Calculation of the Association Probability

To evaluate the association probabilities, the conditioning is broken down into the past data Z^{k-1} and the latest data $Z(k)$.

Since a probabilistic inference can be made on both the number of measurements in the validation region as well as on their location, this is written out explicitly as follows:

$$\beta_i(k) = P\left\{\theta_i(k) \middle| Z^k\right\} = P\left\{\theta_i(k) \middle| Z(k), m(k), Z^{k-1}\right\} \tag{3-67}$$

Using Bayes' formula, the above is rewritten as

$$\beta_i(k) = \frac{1}{c} p\left[Z(k) \middle| \theta_i(k), m(k), Z^{k-1}\right] P\left\{\theta_i(k) \middle| m(k), Z^{k-1}\right\} \qquad i = 0,1,\ldots,m(k) \tag{3-68}$$

The joint density of the validated measurements conditioned on $\theta_i(k),\ i \neq 0$, is the product of the Gaussian pdf of the correct measurements and the pdf of the incorrect measurements, assumed uniform in the validation region whose volume $V(k)$ is given by equation (3-50).

The pdf of the correct measurement with the P_G factor that accounts for restricting the normal density to the validation gate is

$$p\left[Z(k)\middle|\theta_i(k), m(k), Z^{k-1}\right] = P_G^{-1} N\left[z_i(k); z(k/k-1), S(k)\right] = P_G^{-1} N\left[v_i(k); 0, S(k)\right] \tag{3-69}$$

The pdf from (3-68) is then

$$p\left[Z(k)\middle|\theta_i(k), m(k), Z^{k-1}\right] = \begin{cases} V(k)^{-m(k)+1} P_G^{-1} N\left[v_i(k); 0, S(k)\right] & i = 1, \ldots, m(k) \\ V(k)^{-m(k)} & i = 0 \end{cases} \tag{3-70}$$

The probabilities of the association events conditioned only on the number of validated measurements are

$$\begin{aligned} \gamma_i\left[m(k)\right] &= P\left\{\theta_i(k)\middle|m(k), Z^{k-1}\right\} = P\left\{\theta_i(k)\middle|m(k)\right\} \\ &= \begin{cases} \dfrac{1}{m(k)} P_D P_G \left[P_D P_G + (1 - P_D P_G)\dfrac{\mu_F\left[m(k)\right]}{\mu_F\left[m(k)-1\right]}\right]^{-1} & i = 1, \ldots, m(k) \\ (1 - P_D P_G)\dfrac{\mu_F\left[m(k)\right]}{\mu_F\left[m(k)-1\right]}\left[P_D P_G + (1 - P_D P_G)\dfrac{\mu_F\left[m(k)\right]}{\mu_F\left[m(k)-1\right]}\right]^{-1} & i = 0 \end{cases} \end{aligned} \tag{3-71}$$

where $\mu_F(m)$ is the probability mass function of the number of false measurements in the validation region.

Two models can be used for the $\mu_F(m)$ in a volume of interest V:

- a Poisson model with a certain spatial density λ

$$\mu_F(m) = e^{-\lambda V} \frac{(\lambda V)^m}{m!} \tag{3-72}$$

- a diffuse prior model

$$\mu_F(m) = \mu_F(m-1) \tag{3-73}$$

Using the parametric Poisson model in (3-71) yields

$$\gamma_i[m(k)] = \begin{cases} P_D P_G \left[P_D P_G m(k) + (1 - P_D P_G)\lambda V(k)\right]^{-1} & i = 1,\ldots,m(k) \\ (1 - P_D P_G)\lambda V(k)\left[P_D P_G m(k) + (1 - P_D P_G)\lambda V(k)\right]^{-1} & i = 0 \end{cases} \tag{3-74}$$

The nonparametric diffuse prior (3-73) yields

$$\mu_F(k) = \begin{cases} \dfrac{1}{m(k)} P_D P_G & i = 1,\ldots,m(k) \\ 1 - P_D P_G & i = 0 \end{cases} \tag{3-75}$$

Using (3-74) and (3-70) with the explicit expression of the Gaussian pdf into (3-68) yields, after some cancellations, the final equation of the parametric PDA with the Poisson clutter model

$$\beta_i(k) = \begin{cases} \dfrac{e_i}{b + \sum_{j=1}^{m(k)} e_j} & i = 1,\ldots,m(k) \\ \dfrac{b}{b + \sum_{j=1}^{m(k)} e_j} & i = 0 \end{cases} \tag{3-76}$$

where

$$e_i = e^{-\frac{1}{2} v_i(k)' S(k)^{-1} v_i(k)} \tag{3-77}$$

$$b = \left(\frac{2\pi}{\gamma}\right)^{\frac{n_z}{2}} \lambda V(k) c_{n_z}^{-1} \frac{1 - P_D P_G}{P_D} \tag{3-78}$$

The nonparametric PDA is the same as above except for replacing $\lambda V(k)$ in (3-78) by $m(k)$ - this obviates the need to know λ.

The flow chart of one cycle of the Probabilistic Data Association Filter (PDAF) is presented in Figure 3-3.

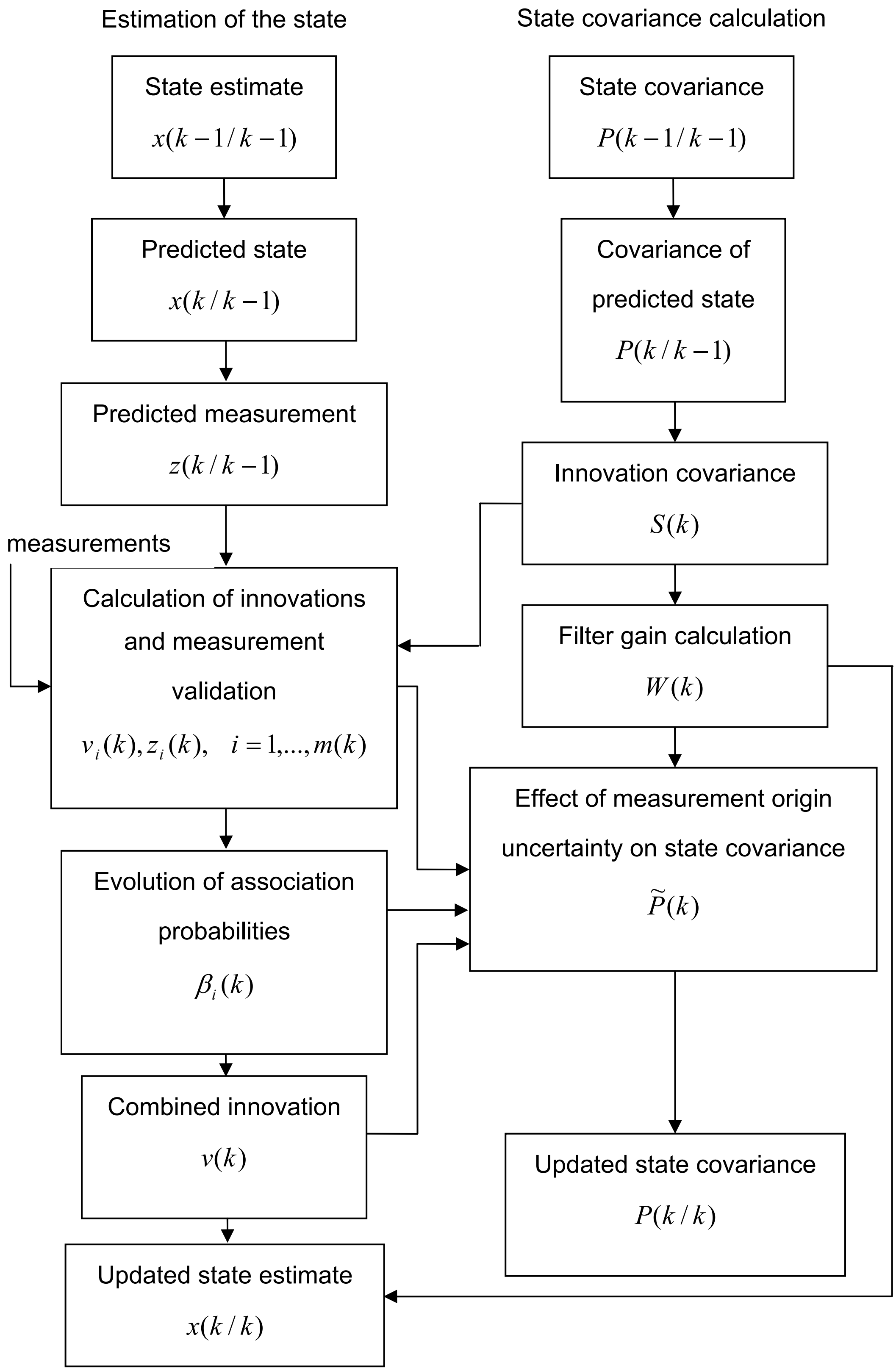

Figure 3-3 One Cycle of the PDAF Algorithm

3.4.4 Joint Probabilistic Data Association Filter (JPDAF)

The Joint Probabilistic Data Association (JPDA) approach is the extension of the PDA method. The PDAF models all the incorrect measurements as random interference, with uniform spatial distribution. The performance of the PDAF degrades significantly when a neighboring target gives rise to persistent interference [2]. The JPDA algorithm assumes that, in a multi target environment, a measurement could have originated from either one of the targets or none of them.

In the JPDAF algorithm the measurement to target association probabilities are computed jointly across the targets and the association probabilities are computed only for the latest set of measurements. In addition, the state estimation is done separately for each target as in the PDAF [2].

The state estimation equations are exactly same as in the PDAF case in subsection 3.4.3.1. The key feature of the JPDA is the joint association events and is given by

$$\theta = \bigcap_{j=1}^{m} \theta_{jt_j} \qquad j = 1,...,m;\ t = 0,1,...,N_T \tag{3-79}$$

where θ_{jt} is the event that measurement j originated from target t and t_j is the index of the target to which measurement j is associated in the event under consideration [2].

3.4.4.1 The Joint Events

In JPDAF, validation gates are not used for the probability calculation of the association events but helps for selection of the joint events. Define validation matrix as

$$\Omega = \left\lfloor w_{jt} \right\rfloor \qquad j = 1,...,m; t = 0,1,...,N_T \tag{3-80}$$

Validation matrix has binary elements that indicate measurement j lies in the validation gate of target t. A joint association event θ is represented by the event matrix

$$\hat{\Omega}(\theta) = \left[\hat{w}_{jt}(\theta)\right] \tag{3-81}$$

consisting of units in Ω corresponding to the association in θ.

$$\hat{w}_{jt}(\theta) = \begin{cases} 1 & if\ \theta_{jt} \in \theta \\ 0 & otherwise \end{cases} \tag{3-82}$$

A feasible association event is one where

- a measurement can have only one source

$$\sum_{t=0}^{N_T} \hat{w}_{jt}(\theta) = 1 \qquad \forall j \tag{3-83}$$

- at most one measurement can originate from a target

$$\delta_t(\theta) = \sum_{j=1}^{m} \hat{w}_{jt}(\theta) \le 1 \qquad t = 1, \ldots, N_T \tag{3-84}$$

The binary variable defined in (3-84) is called target detection indicator because it indicates whether a measurement is associated with target t in event θ. In addition, another binary variable called measurement association indicator is defined in order to indicate if measurement j is associated with a target in event θ.

$$\tau_j(\theta) = \sum_{t=1}^{N_T} \hat{w}_{jt}(\theta) \tag{3-85}$$

From the above definitions the number of false measurements in event θ is

$$\phi(\theta) = \sum_{j=1}^{m} \left[1 - \tau_j(\theta)\right] \tag{3-86}$$

3.4.4.2 Evaluation of the Joint Probabilities

By using Bayes' formula the joint association event probabilities are

$$P\{\theta(k)|Z^k\} = P\{\theta(k)|Z^k, m(k), Z^{k-1}\} = \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|m(k)\} \tag{3-87}$$

where c is the normalization constant.

The first term of the equation (3-87) is the likelihood function of the measurements where $m(k)$ is the number of measurements in the union of the validation regions at time k. Measurements not associated with a target are assumed to be uniformly distributed in the surveillance region of volume V. The likelihood function can be written as follows

$$p[Z(k)|\theta(k), m(k), Z^{k-1}] = V^{-\phi} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \tag{3-88}$$

The second term is the prior probability of a joint association event and is given by

$$\begin{aligned} P\{\theta(k)|m(k)\} &= P\{\theta(k), \delta(\theta), \phi(\theta)|m(k)\} \\ &= \frac{\phi!}{m(k)!} \mu_F(\phi) \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \end{aligned} \tag{3-89}$$

Substituting (3-88) and (3-89) into (3-87) yields the posterior probability of a joint association event as

$$P\{\theta(k)|Z^k\} = \frac{1}{c} \frac{\phi!}{m(k)!} \mu_F(\phi) V^{-\phi} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \tag{3-90}$$

The model used for the $\mu_F(\phi)$ in equation (3-90) determines the parametric or non-parametric version of the JPDAF.

The parametric JPDAF uses the Poisson pmf

$$\mu_F(\phi) = e^{-\lambda V} \frac{(\lambda V)^{\phi}}{\phi!} \tag{3-91}$$

which, requires to know the spatial density λ of the false measurements.

When we insert equation (3-91) into the (3-90), we get the joint association probabilities of the parametric JPDAF

$$\begin{aligned} P\{\theta(k) | Z^k\} &= \frac{\lambda^{\phi}}{c_1} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \\ &= \frac{1}{c_2} \prod_j \{\lambda^{-1} f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \end{aligned} \tag{3-92}$$

Here c_1 is the appropriate normalization constant and since $m(k)$ is a fixed number, c_2 is a new normalization constant defined as

$$c_2 = c_1 \lambda^{-m(k)} \tag{3-93}$$

The non-parametric JPDAF uses the diffuse prior

$$\mu_F(\phi) = \varepsilon \qquad \forall \phi \tag{3-94}$$

which does not require λ.

Using nonparametric version equation 3-90 becomes

$$\begin{aligned} P\{\theta(k) | Z^k\} &= \frac{1}{c_3} \frac{\phi!}{V^{\phi}} \prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \\ &= \frac{1}{c_4} \phi! \prod_j \{V f_{t_j}[z_j(k)]\}^{\tau_j} \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} \end{aligned} \tag{3-95}$$

Again, c_3 is the appropriate normalization constant and c_4 is defined as

$$c_4 = c_3 V^{-m(k)} \tag{3-96}$$

The flow chart of one cycle of the JPDAF is presented in Figure 3-4.

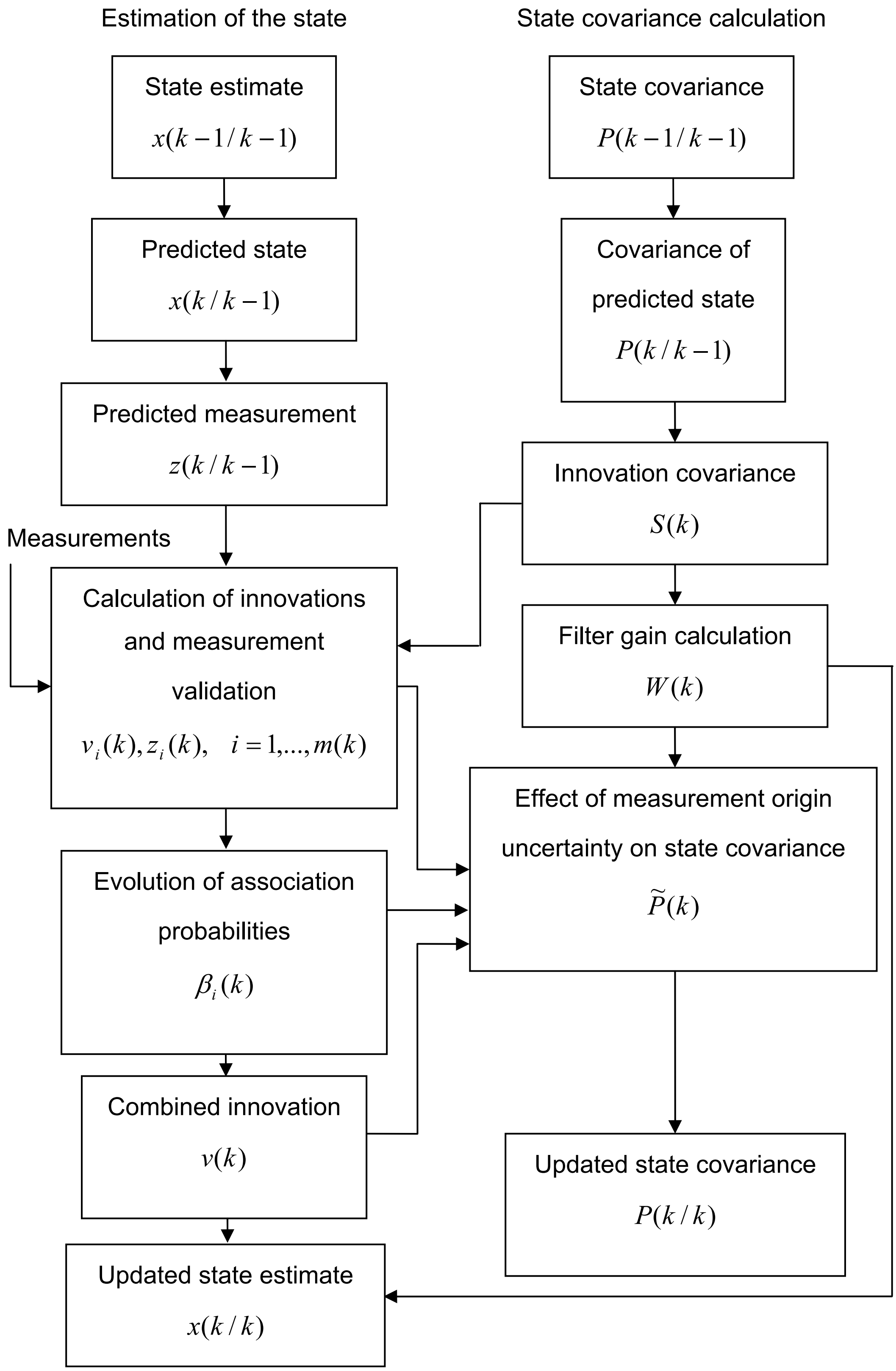

Figure 3-4 One Cycle of the JPDAF Algorithm

3.4.5 Interactive Multiple Modeling

The Interacting Multiple Model (IMM) estimator is a suboptimal hybrid filter that has been shown to be one of the most cost-effective hybrid state estimation schemes. The main feature of this algorithm is its ability to estimate a state of a dynamic system with several behavior modes, which can switch from one to another [4]. The importance of this approach is that the computational requirements are nearly linear. On the other hand, its performance is same as an algorithm that has quadratic complexity.

Another advantage of the IMM filter is that it can be constructed by using a variety of filters such as the Kalman filter (or EKF for modeling nonlinearities) or even probabilistic data association (PDA) and joint probabilistic data association (JPDA) filters for data association in cluttered environments.

In the IMM approach, more than one model is used to describe the motion of the target. The probability of each model being true is found using a likelihood function for the model, and switching between models is considered using a transition probability. Finally, the combined estimate is obtained using a weighted sum of the estimates from the sub filters according to a posteriori probabilities of different models as the weighting factors. The IMM algorithm has shown better results than the switching schemes, because a smooth transition is achieved from one model to another [3].

One cycle of the IMM algorithm consists of interaction (mixing), filtering and combination. At each time, initial estimate of each filter belonging to a certain mode is computed by mixing the state estimates of all filters. This is followed by filtering (i.e. prediction and update steps) for each mode. Finally, the weighted sum of the updated state estimates of all filters is combined to yield the state estimate. The equations that summarize the topics above are given in the paragraphs below.

The simplest hybrid system is described by a set of linear models with the mode transition governed by a first order homogenous Markov chain [4].

$$p_{ij} = P\{m_j(k+1) \mid m_i(k)\} \tag{3-97}$$

where $m_j(k)$ describes the distinct segment of the trajectory. In other words, it denotes the model that is in effect during the motion.

Following Bar-Shalom and Li [2], Simeonova and Semerdjiev [13], one cycle of the IMM algorithm is as follows.

3.4.5.1 Interaction (Mixing)

State estimates and covariances of all filters are mixed to obtain initial state for the filter matched to mode $m_j(k)$. In order to obtain the initial state, first the mixing probabilities are calculated.

$$\mu_{i/j}(k-1/k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1) \tag{3-98}$$

Here $\bar{c}_j$ is the normalization factor and given by

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij} \mu_{i/j}(k-1/k-1) \quad r = possible\ \mathrm{mod}\,es \tag{3-99}$$

The mixed states ($x^{0j}(k-1/k-1)$) and covariances ($P^{0j}(k-1/k-1)$) then are calculated as follows

$$x^{0j}(k-1/k-1) = \sum_{i=1}^{r} x^i(k-1/k-1)\mu_{i/j}(k-1/k-1) \tag{3-100}$$

$$P^{0j}(k-1/k-1) = \sum_{i=1}^{r} \mu_{i/j}(k-1/k-1)\Big\{P^i(k-1/k-1) + \\ \left[x^i(k-1/k-1) - x^{oj}(k-1/k-1)\right]\left[x^i(k-1/k-1) - x^{oj}(k-1/k-1)\right]'\Big\} \tag{3-101}$$

Here $x^i(k-1/k-1)$ is the state estimate and $P^i(k-1/k-1)$ is the covariance of the state estimate of the ith filter.

3.4.5.2 Filtering

The estimate (3-100) and covariance (3-101) are used as input to the filter match to $m_j(k)$ in order to obtain state estimate $x^j(k/k)$ and covariance $P^j(k/k)$, and also the mode likelihood $\Lambda_j(k)$. The state estimation, covariance and mode likelihood calculation of each filter is done in parallel as follows

$$x^j(k/k-1) = F^j(k)x^{0j}(k-1/k-1) \tag{3-102}$$

$$P^j(k/k-1) = F^j(k)P^{0j}(k-1/k-1)F^j(k)' + Q^j(k) \tag{3-103}$$

$$S^j(k) = H^j(k)P^j(k/k-1)H^j(k)' + R^j(k) \quad (residual\ \mathrm{cov}ariance) \tag{3-104}$$

$$W^j(k) = P^j(k/k-1)H^j(k)'S^j(k)^{-1} \quad (filter\ gain) \tag{3-105}$$

$$z^j(k/k-1) = H^j(k)x^j(k/k-1) \qquad (measurement\ prediction) \tag{3-106}$$

$$v^j(k) = z(k) - z^j(k/k-1) \qquad (residual) \tag{3-107}$$

$$x^j(k/k) = x^j(k/k-1) + W^j(k)v^j(k) \tag{3-108}$$

$$P^j(k/k) = P^j(k/k-1) - W^j(k)S^j(k)W^j(k)' \tag{3-109}$$

$$\Lambda_j(k) = N(v^j(k);0,S^j(k)) = \left|S^j(k)^{-1/2}\right| \exp\left\{-\frac{1}{2}v^j(k)'S^j(k)^{-1}v^j(k)\right\} \tag{3-110}$$

The mode likelihood is used to update the mode probabilities for the next time scan. This is done as follows

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_{i=1}^{r} p_{ij}\mu_i(k-1) = \frac{1}{c}\Lambda_j(k)\bar{c}_j \tag{3-111}$$

Here c is the normalization factor and is given by

$$c = \sum_{j=1}^{r} \Lambda_j(k)\bar{c}_j \tag{3-112}$$

3.4.5.3 Combination

The combination of the updated state estimates and covariances produces the output estimates.

$$x(k/k) = \sum_{j=1}^{r} x^j(k/k)\mu_j(k) \tag{3-113}$$

$$P(k/k) = \sum_{j=1}^{r} \left\{ P^j(k/k) + \left[x^j(k/k) - x(k/k)\right]\left[x^j(k/k) - x(k/k)\right]' \right\} \mu_j(k) \tag{3-114}$$

The IMM algorithm has three important properties: it is recursive, modular and has fixed computational requirements per cycle. The one cycle algorithm of the IMM is given in Figure 3-5.

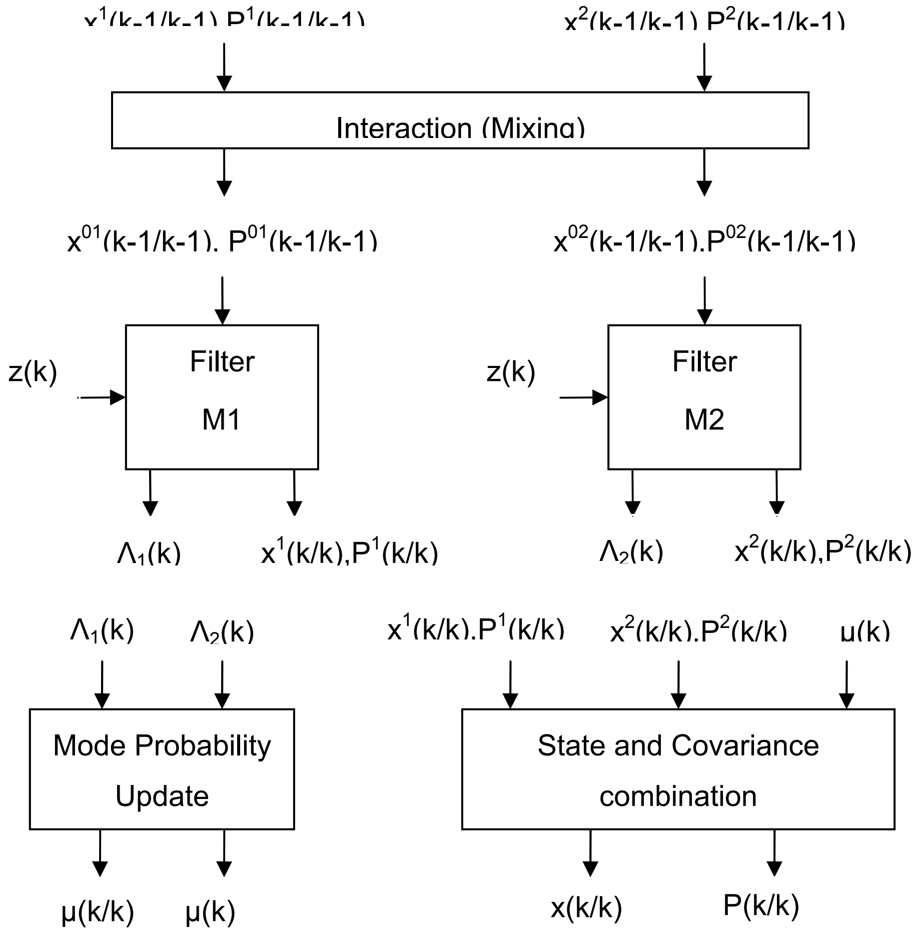

Figure 3-5 One Cycle of the IMM Algorithm

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 INTRODUCTION

In this chapter, the simulations that show the implementation results of the theory given in the previous chapters are explained in detail. In order to demonstrate the differences between the methods for data association and tracking, different scenarios are constructed and the algorithms are applied to these scenarios.

In the implementation of the theory for multi target tracking, the algorithms that are used can be stated as follows: track score function for track initiation, probabilistic data association and joint probabilistic data association for observation to track assignments and interactive multiple modeling for estimation (filtering).

All the scenarios in this chapter are carried out offline (not real time) with artificial data sequences. In the scenarios, noise is added to the real data in order to construct measurements. In addition, the multi target tracking problem is made more realistic by including false alarms due to clutter. For this reason, while generating false alarm data, the number of false alarms is stated according to the equation below and these data are distributed uniformly to the noise data to get the measurements, which includes false alarm data.

$$FA\_count = floor(abs(randn)) \tag{4-1}$$

These data sequences are generated in MATLAB and two methods are implemented to process this data off-line. The methods are implemented as m-files and all the m-files are generalized for future use. The artificial data is used to validate the implementation of the algorithms. According to the results of the simulations, we can easily see that IMMPDAF technique is successful in single target tracking and in tracking of multiple targets that are far away from each other. On the other hand, IMMJPDAF technique is successful in multi target tracking in all aspects. In the scenarios, IMMJPADF is run for tracking up to ten targets at a time. In the next section of this chapter first the simulations that are done for IMMPDAF technique is shown and the simulations for the IMMJPDAF technique are given afterwards. In the figures, the positions are given in kilometers and the time is given in seconds.

4.2 IMMPDAF Results

In order to examine the performance of the IMMPDAF algorithm three scenarios are created. First, there is a maneuvering target, which moves approximately for 165 seconds. Second, there are two maneuvering targets with non-intersecting trajectories. Finally, in the last scenario, again there are two targets, which have constant accelerations, and intersecting trajectories. The results for these scenarios are given below.

In the first scenario, the target makes a maneuver in the x-y plane while its z-coordinate is constant. Blue lines give the actual position of the target and the measurements are shown as red dotted points in Figure 4-1 and Figure 4-2. The IMMPDAF algorithm is applied to this scenario and it successfully tracks the target. The estimation results of the filter are given in Figure 4-3.

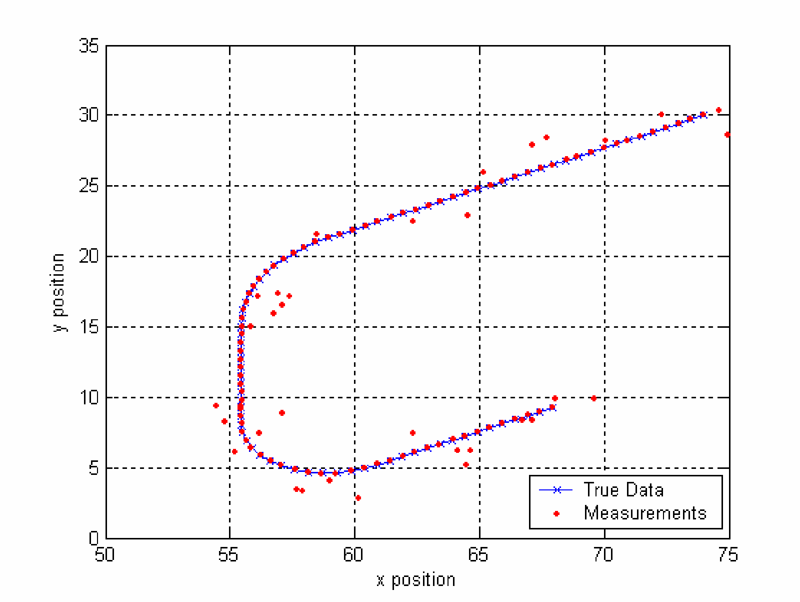

Figure 4-1 X and Y positions of the target (Scenario 1)

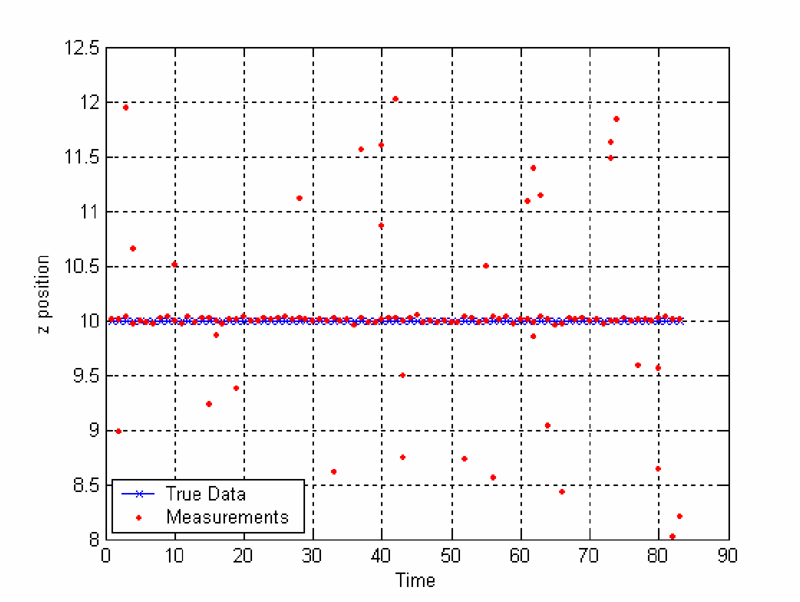

Figure 4-2 Z position of the target (Scenario 1)

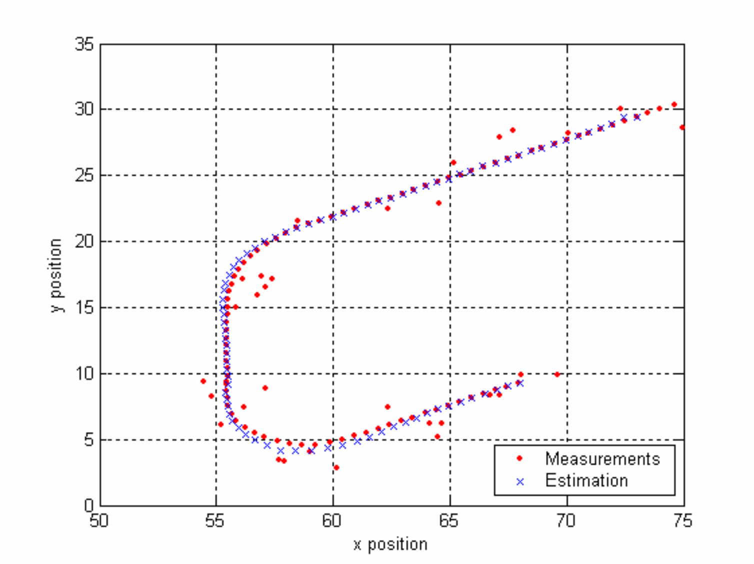

Figure 4-3 Output of the IMMPDAF Algorithm (Scenario 1)

In the second scenario, there are two targets, which make maneuvers in the x- y plane at different altitude (i.e. z-coordinates of the trajectories are not equal but constant). Blue lines indicate the actual position of the targets and the measurements are shown as red dotted points in Figure 4-4 and Figure 4-5. The IMMPDAF algorithm is still successful for this scenario since the targets are far away from each other. The result of the simulation is shown in Figure 4-6.

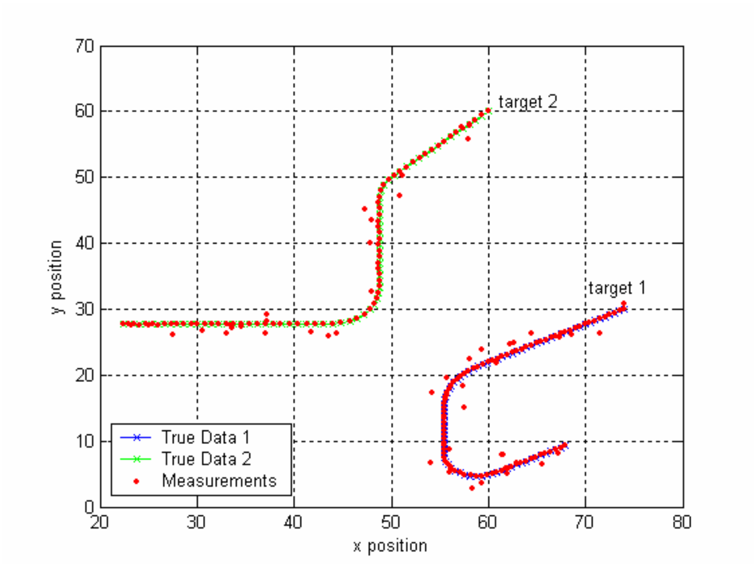

Figure 4-4 X and Y positions of the targets (Scenario 2)

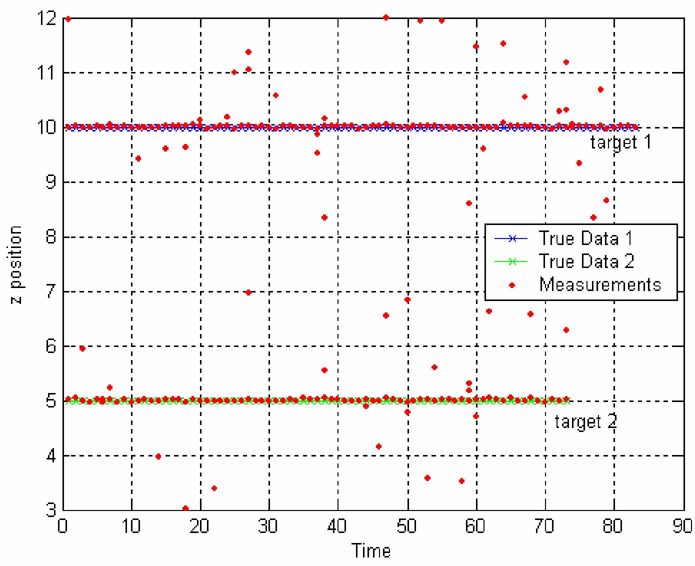

Figure 4-5 Z positions of the targets (Scenario 2)

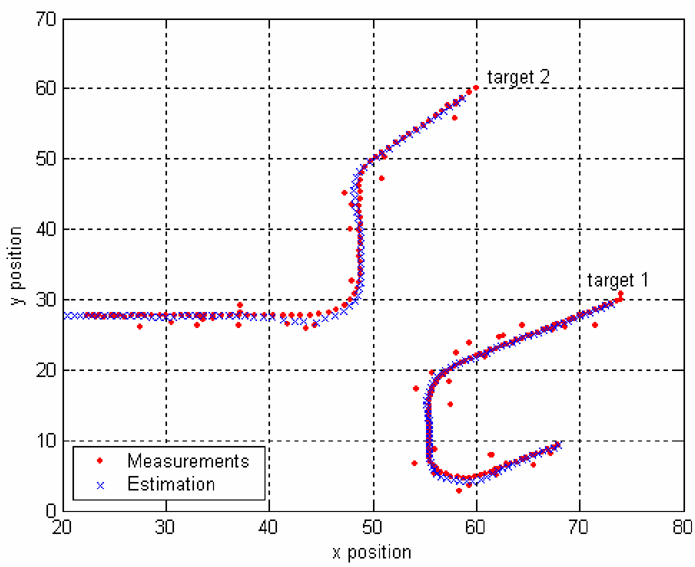

Figure 4-6 Output of the IMMPDAF Algorithm (Scenario 2)

In the last scenario for the IMMPDAF algorithm, there are again two targets of interest. On the other hand, the trajectories of these targets intersect on the x-y plane. During intersection, the z-coordinate difference between the targets falls within the gate. The actual positions of the targets are given in Figure 4-7 and

Figure 4-8. In this case, the IMMPDAF algorithm can not track the targets due to the fact that the targets get very close in the intersection region. The simulation result is shown in Figure 4-9.

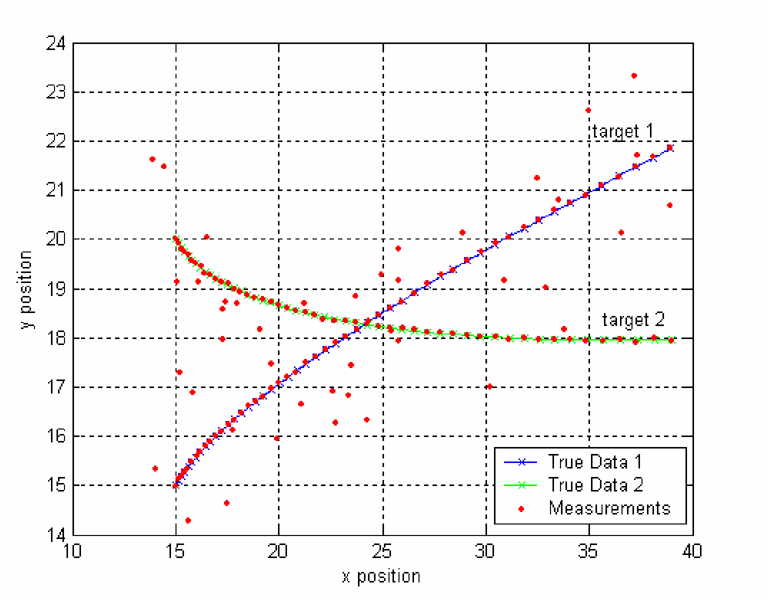

Figure 4-7 X and Y positions of the targets (Scenario 3)

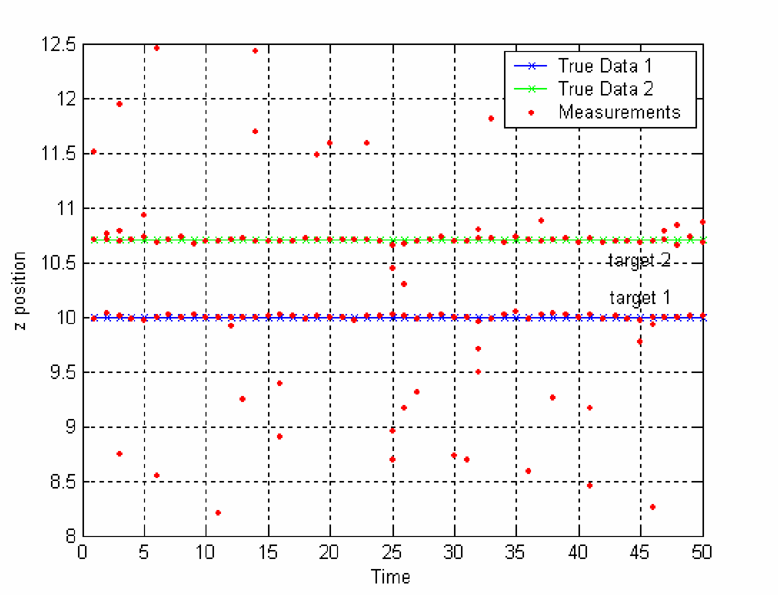

Figure 4-8 Z positions of the targets (Scenario 3)

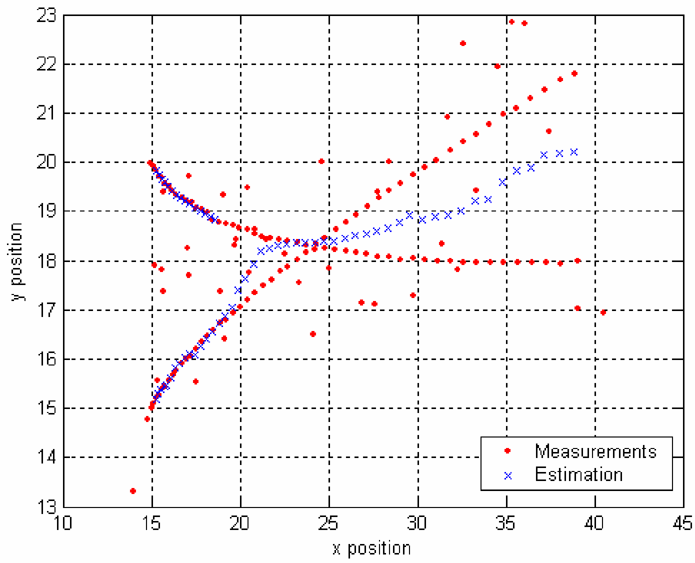

Figure 4-9 Output of the IMMPDAF Algorithm (Scenario 3)

As seen from the figure above, IMMPDAF can not track the targets. This is because in the intersection region, the algorithm assigns the measurements to only one of the targets and after calculating the probabilities, updates only that target's position. Therefore, after a while the IMMPDAF algorithm stops updating the other target's position and deletes it from the track file.

4.3 IMMJPDAF Results

The IMMJPDAF algorithm is first applied to the scenario where the IMMPDAF algorithm fails to track intersecting targets. Then, the algorithm is run at various scenarios where the number of targets increases three to ten. Some of the paths of the targets are constructed according to the scenarios in the literature "Benchmark for Radar Allocation and Tracking in ECM" [16].

The simulation result of the third scenario is given in Figure 4-10.

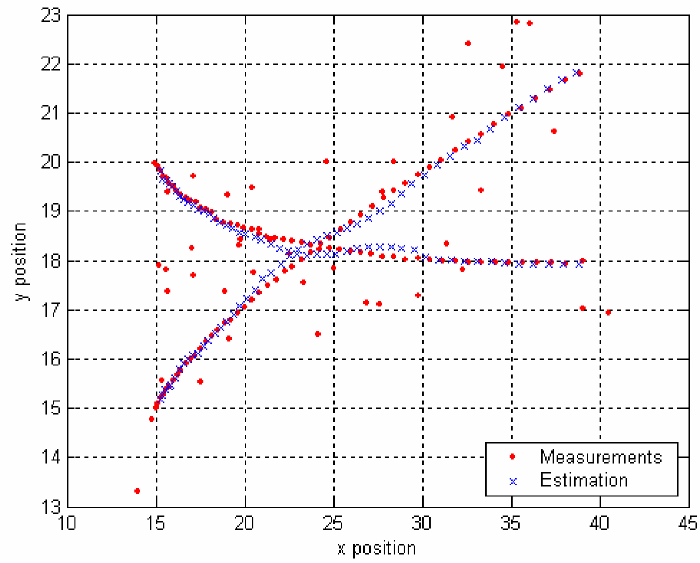

Figure 4-10 Output of the IMMJPDAF Algorithm (Scenario 3)

Here the algorithm considers all the measurements and calculate the probabilities of the hypotheses that the measurements can come from any of the targets. As seen from the above figure, IMMJPDAF algorithm successfully tracks the targets.

In the following scenarios, the number of targets is increased gradually and the IMMJPDAF results are given.

In the fourth scenario, there are three targets. The second target has a constant acceleration in the x- y plane while the first and the third targets have a constant velocity along their paths. At the intersection points the z coordinate difference of the targets fall within the gates. The second target passes simultaneously with the first and the third target at the intersection points. The positions of the targets are shown in Figure 4-11 and Figure 4-12 and the filter output is shown in Figure 4-13.

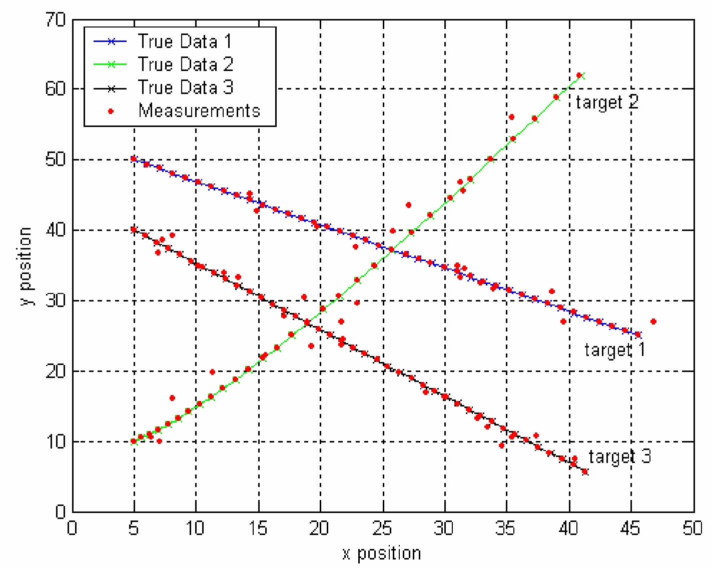

Figure 4-11 X and Y positions of the targets (Scenario 4)

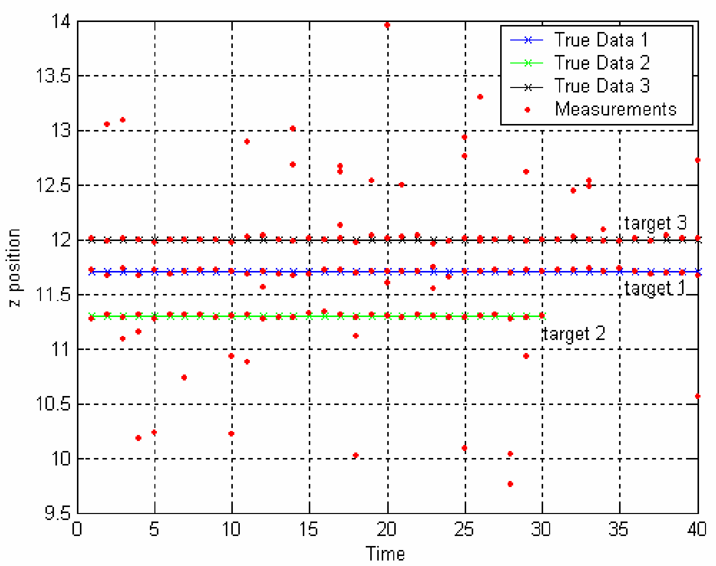

Figure 4-12 Z positions of the targets (Scenario 4)

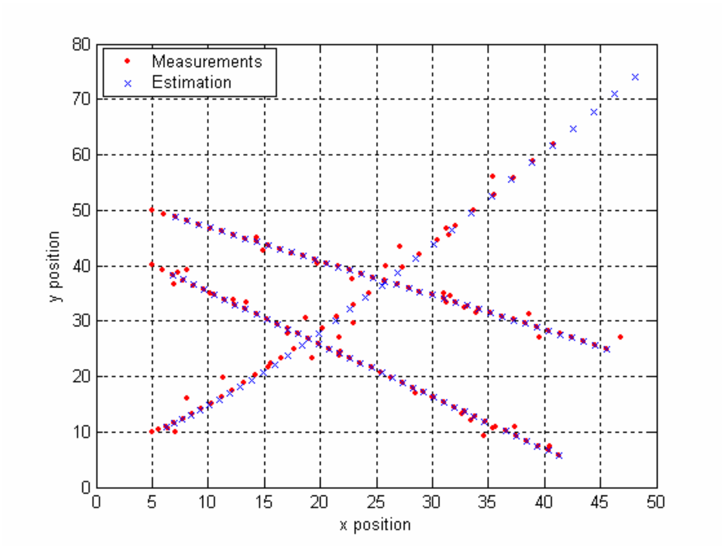

Figure 4-13 Output of the IMMJPDAF Algorithm (Scenario 4)

The fifth scenario consists of four targets. The second and the fourth target have a constant acceleration in the x-y plane while the first and the third targets have a constant velocity along their paths. The z coordinate differences of the targets fall within the gates at the intersection points. The second target passes simultaneously with the first and the third target at the intersection points. The positions of the targets are shown in Figure 4-14 and Figure 4-15 and the filter output is shown in Figure 4-16.

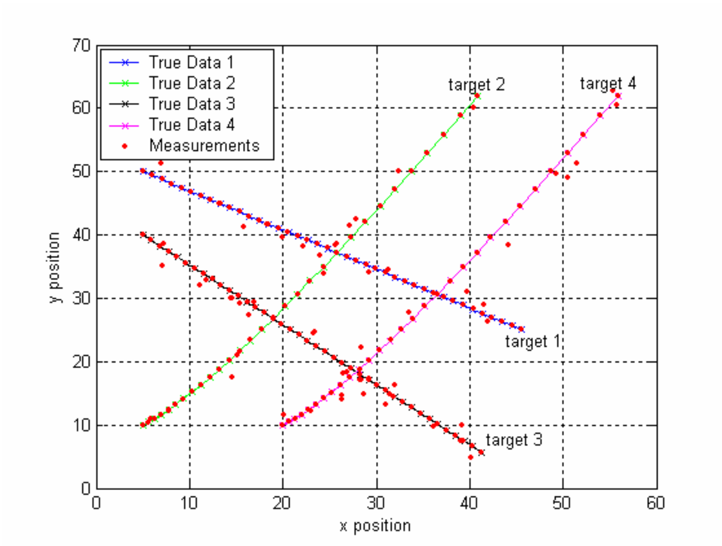

Figure 4-14 X and Y positions of the targets (Scenario 5)

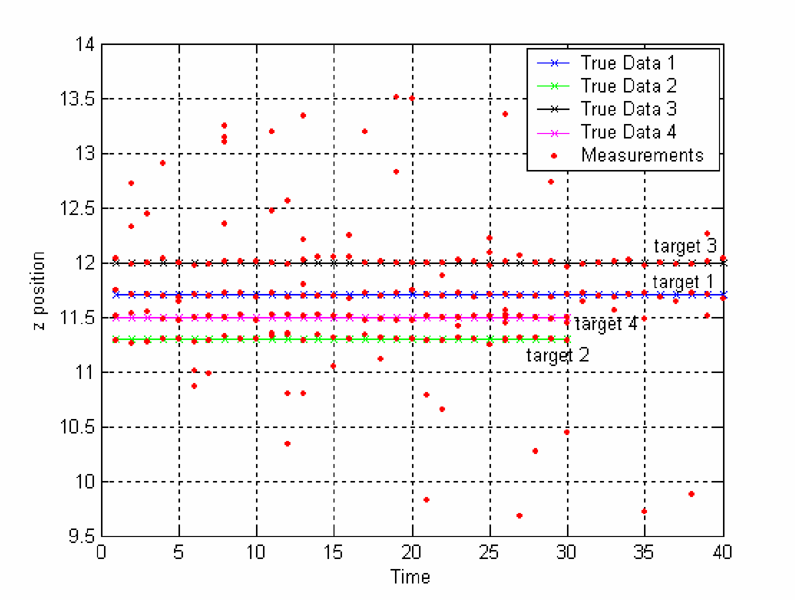

Figure 4-15 Z positions of the targets (Scenario 5)

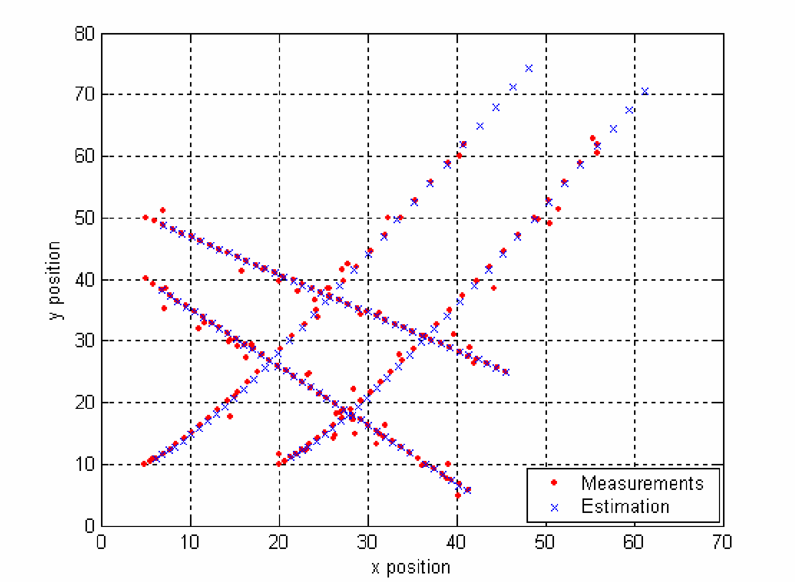

Figure 4-16 Output of the IMMJPDAF Algorithm (Scenario 5)

In the sixth scenario there are again four targets all of which doing a maneuver in x-y plane. Except the maneuvering time intervals, targets have a constant velocity along their paths. The z coordinate differences of the targets fall within the gates at the intersection points. The second and the third target meet simultaneously at the intersection point (x=23km, y=38km). The positions of the targets are shown in Figure 4-17 and Figure 4-18 and the filter output is shown in Figure 4-19.

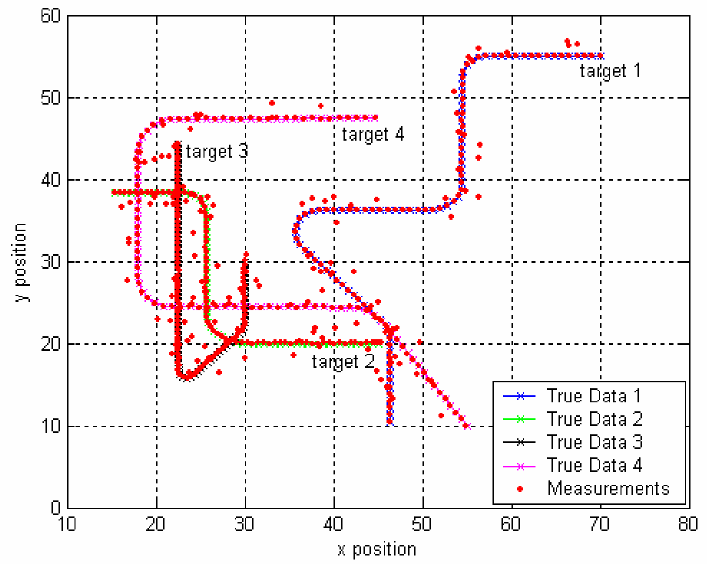

Figure 4-17 X and Y positions of the targets (Scenario 6)

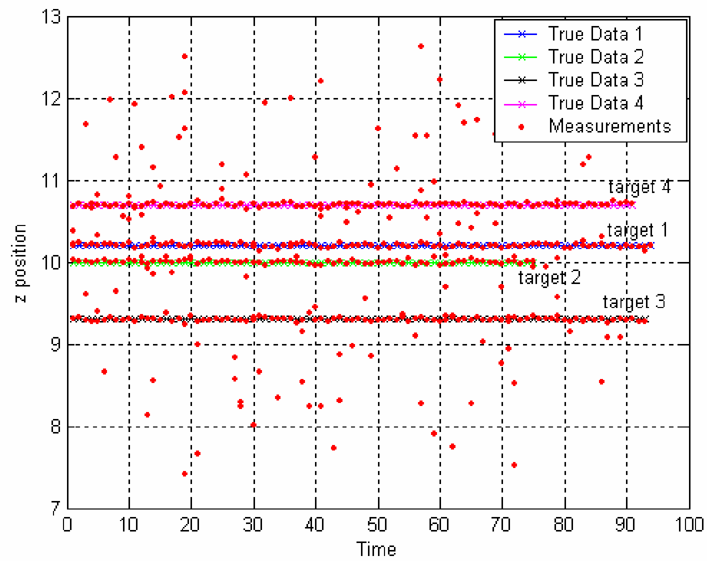

Figure 4-18 Z positions of the targets (Scenario 6)

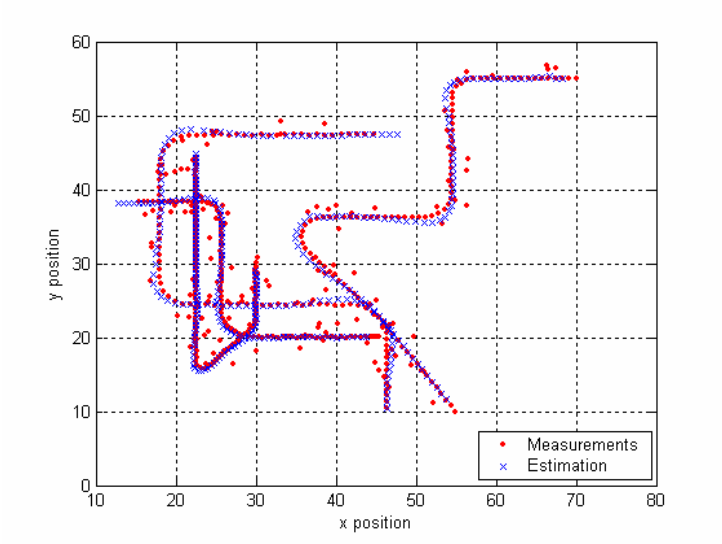

Figure 4-19 Output of the IMMJPDAF Algorithm (Scenario 6)

In the seventh and eighth scenarios, number of the targets is increased to five. In addition to the targets of the fifth scenario, one more maneuvering target is included. The target that is added in the seventh scenario, is far away from those four targets while the target in the eighth scenario, is closer. The positions of the targets in scenarios seven and eight are shown in Figure 4-20, Figure 4-21, Figure 4-23 and Figure 4-24 and the filter outputs are shown in Figure 4-22 and Figure 4-25.

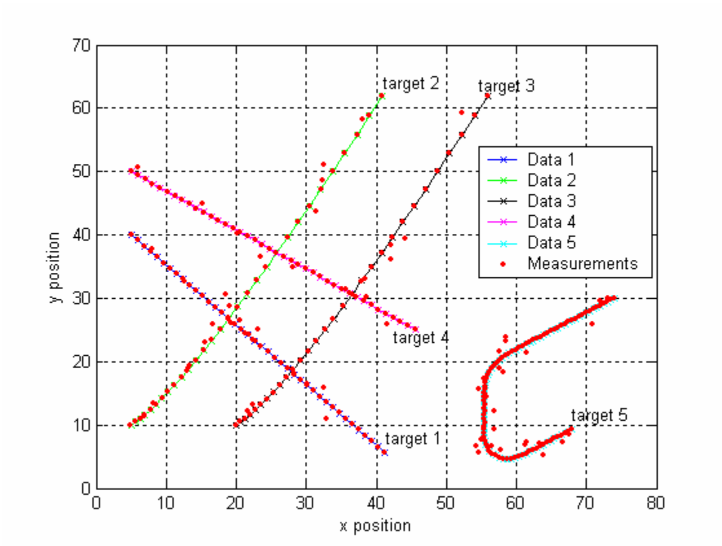

Figure 4-20 X and Y positions of the targets (Scenario 7)

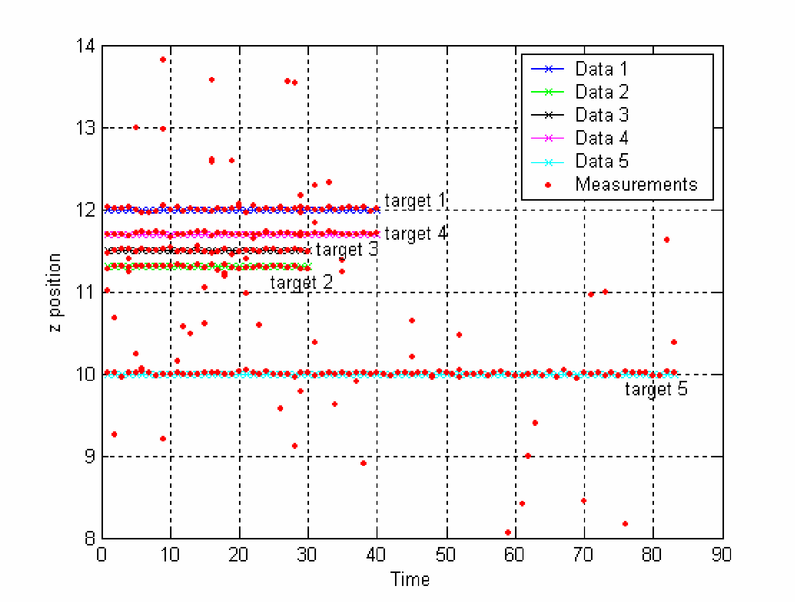

Figure 4-21 Z positions of the targets (Scenario 7)

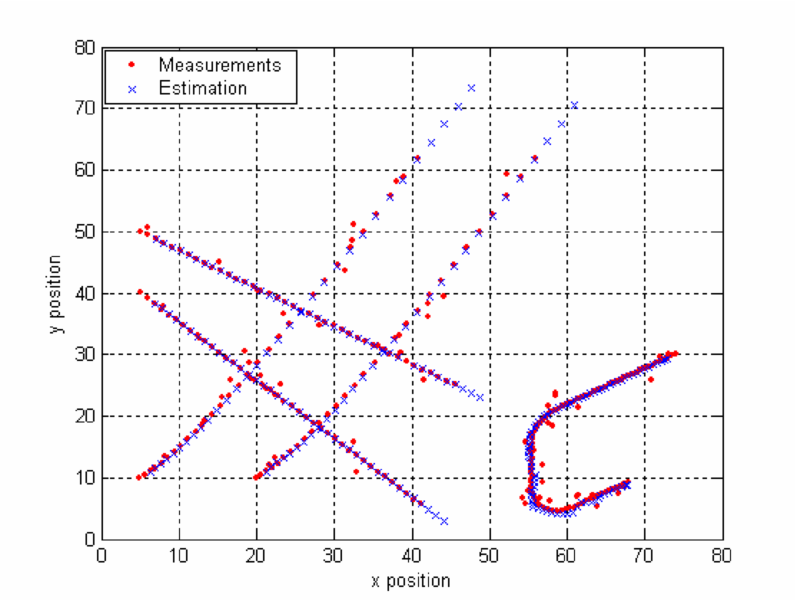

Figure 4-22 Output of the IMMJPDAF Algorithm (Scenario 7)

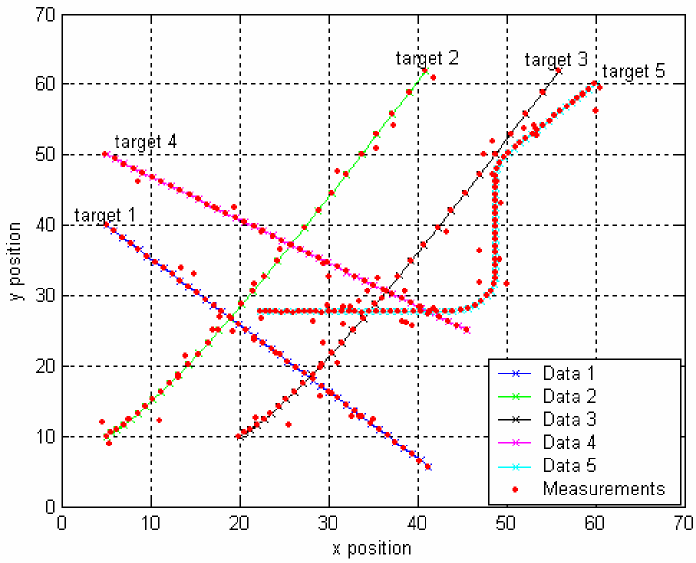

Figure 4-23 X and Y positions of the targets (Scenario 8)

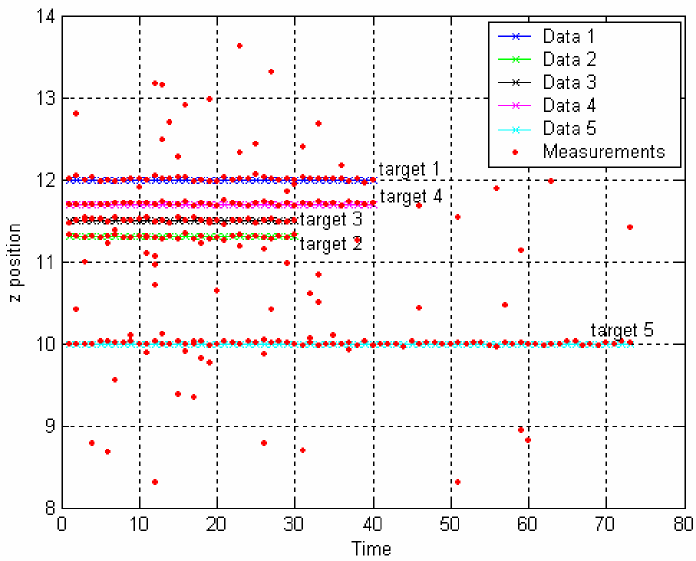

Figure 4-24 Z positions of the targets (Scenario 8)

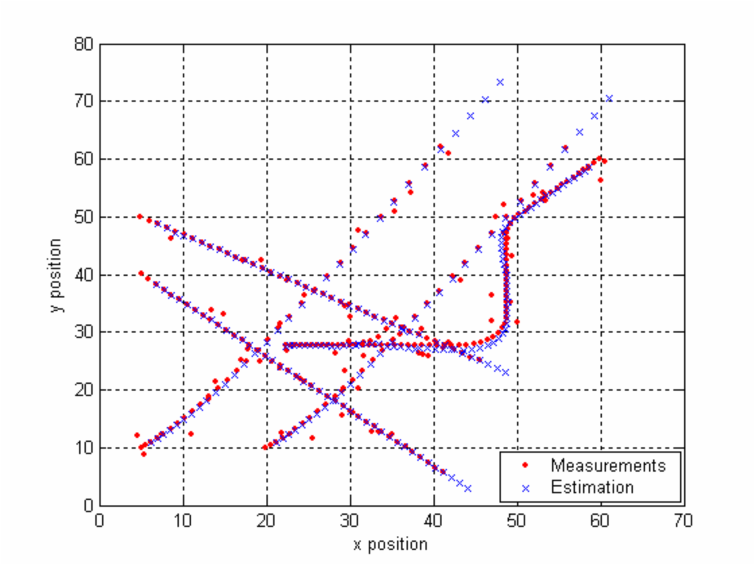

Figure 4-25 Output of the IMMJPDAF Algorithm (Scenario 8)

In the next two scenarios 9 and 10, the number of the targets is increased again by one. One more target is added to scenarios seven and eight to form these scenarios. The positions of the targets and the filter outputs are given below.

In Figure 4-26, Figure 4-27, Figure 4-29 and Figure 4-30 the positions of the targets and in Figure 4-28 and Figure 4-31 the filter outputs are shown.

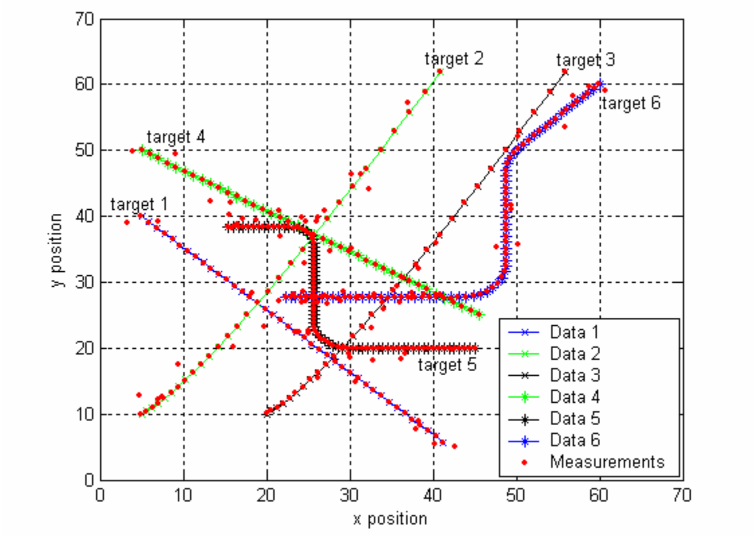

Figure 4-26 X and Y positions of the targets (Scenario 9)

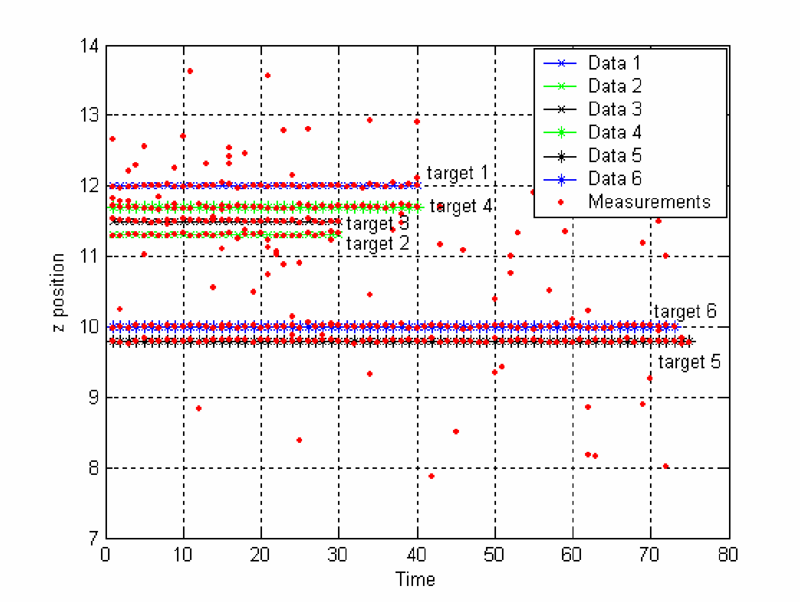

Figure 4-27 Z positions of the targets (Scenario 9)

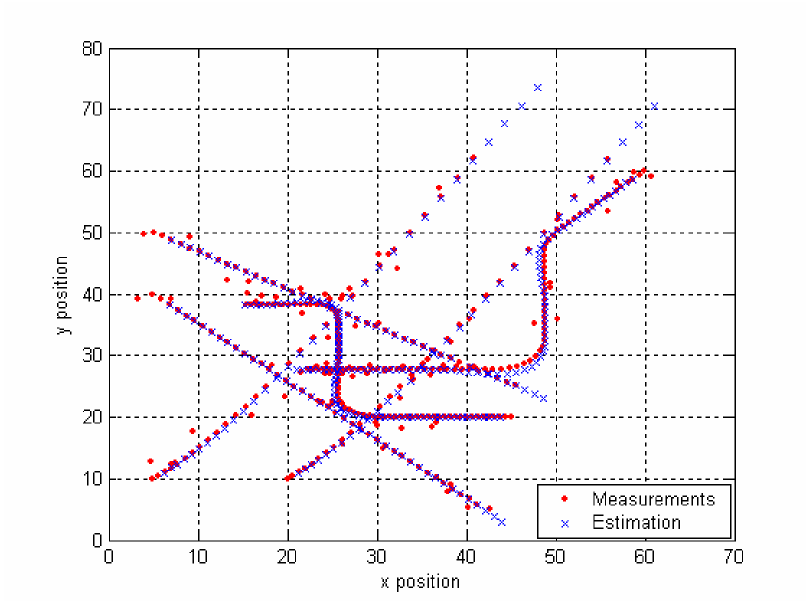

Figure 4-28 Output of the IMMJPDAF Algorithm (Scenario 9)

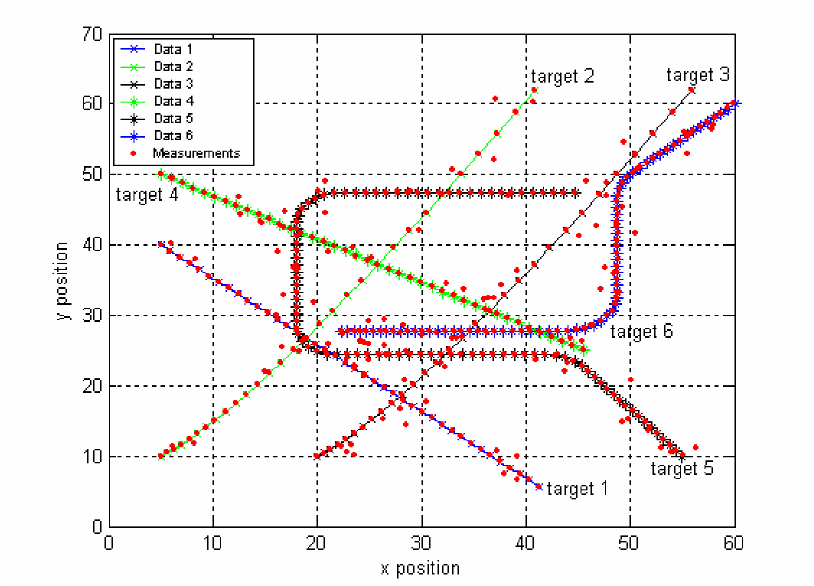

Figure 4-29 X and Y positions of the targets (Scenario 10)

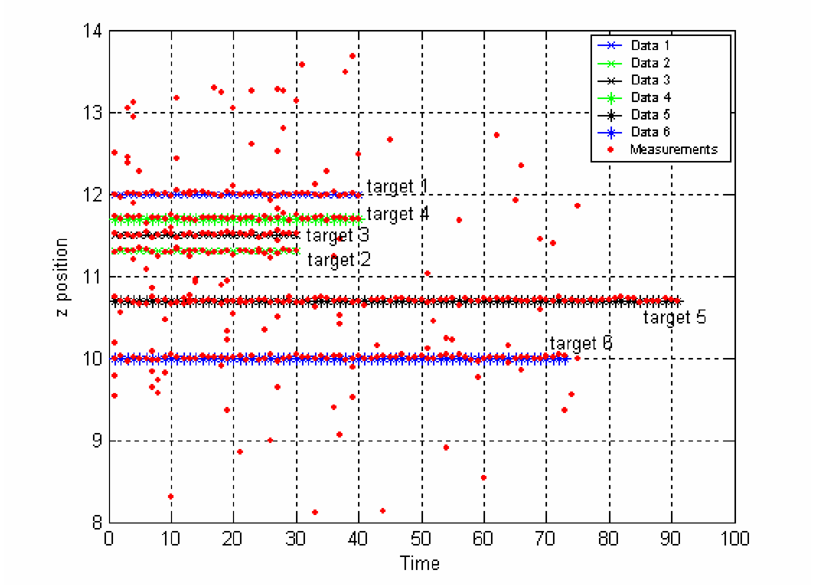

Figure 4-30 Z positions of the targets (Scenario 10)

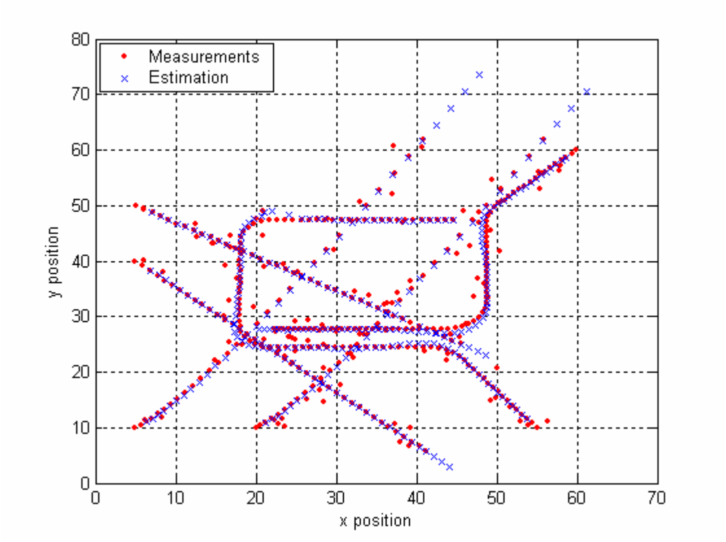

Figure 4-31 Output of the IMMJPDAF Algorithm (Scenario 10)

The number of targets becomes seven in the next two scenarios 11 and 12. Every target moves according to specific kinematic model, in order to asses the performance of the algorithm for all possible target trajectories. In the scenarios, targets have constant velocity, constant acceleration and coordinated turn dynamics. The positions of the targets are shown in Figure 4-32, Figure 4-33, Figure 4-35 and Figure 4-36. In addition, the outputs are shown in Figure 4-34 and Figure 4-37.

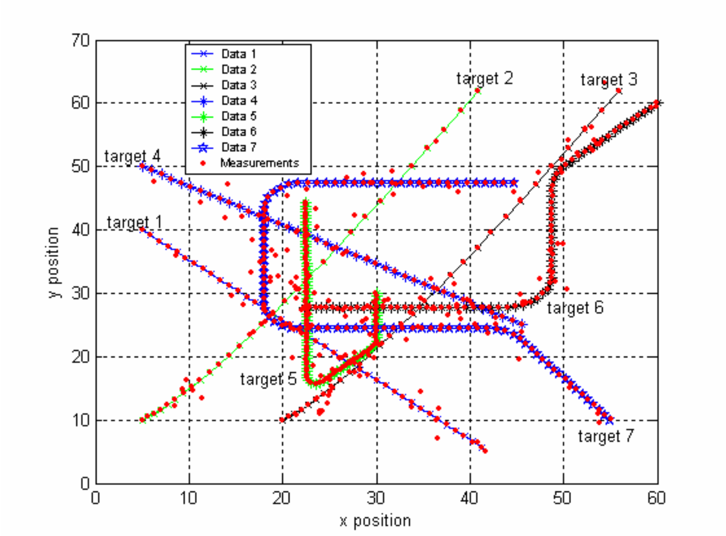

Figure 4-32 X and Y positions of the targets (Scenario 11)

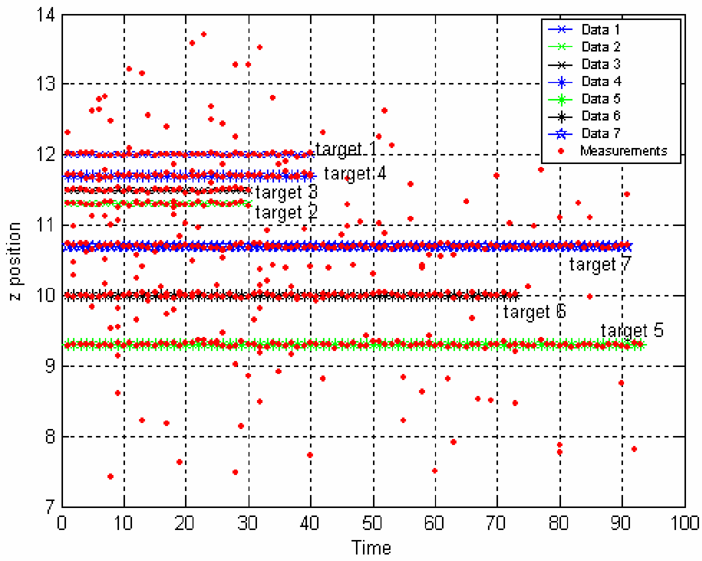

Figure 4-33 Z positions of the targets (Scenario 11)

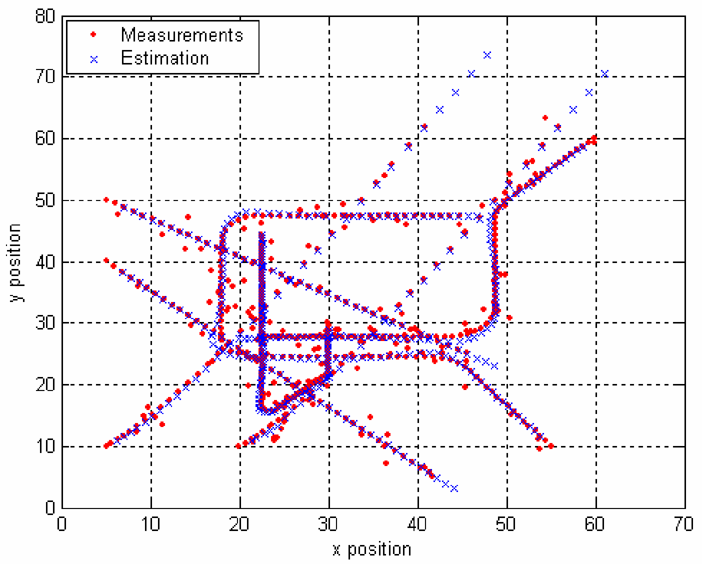

Figure 4-34 Output of the IMMJPDAF Algorithm (Scenario 11)

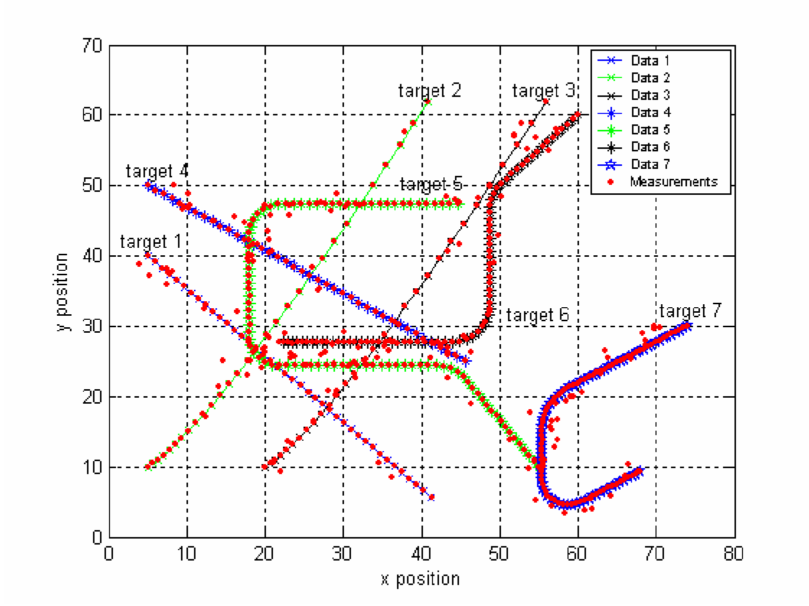

Figure 4-35 X and Y positions of the targets (Scenario 12)

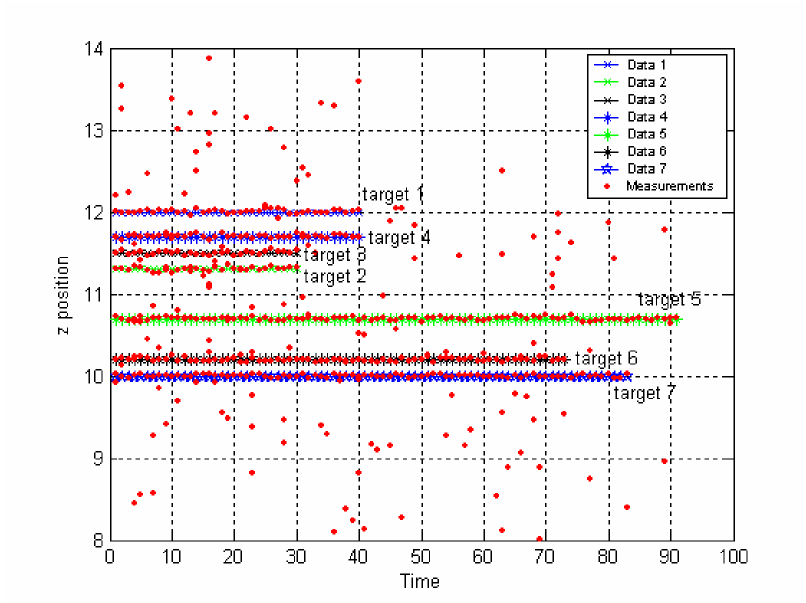

Figure 4-36 Z positions of the targets (Scenario 12)

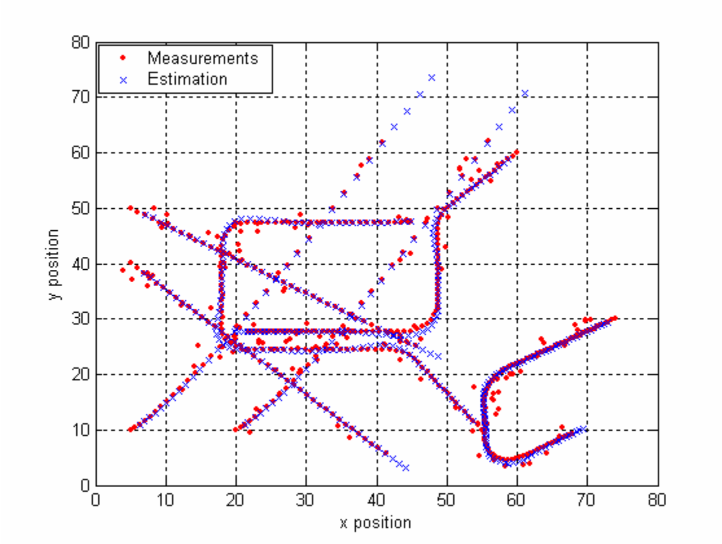

Figure 4-37 Output of the IMMJPDAF Algorithm (Scenario 12)

The next two scenarios (13 and 14) consist of eight targets. In the thirteenth scenario targets move more closely to each other. On the other hand, in the fourteenth scenario targets are grouped in four. Again, targets have constant velocity, constant acceleration and coordinated turn dynamics in these scenarios. The positions of the targets are shown in Figure 4-38, Figure 4-39, Figure 4-41 and Figure 4-42. In addition, the outputs are shown in Figure 4-40 and Figure 4-43.

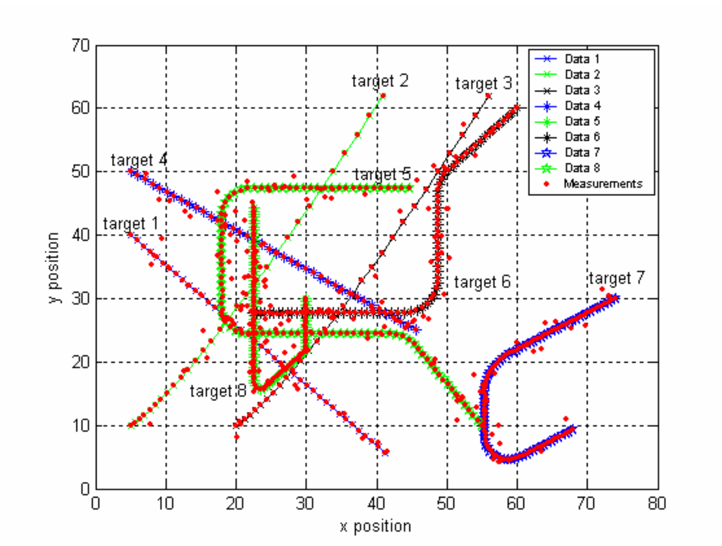

Figure 4-38 X and Y positions of the targets (Scenario 13)

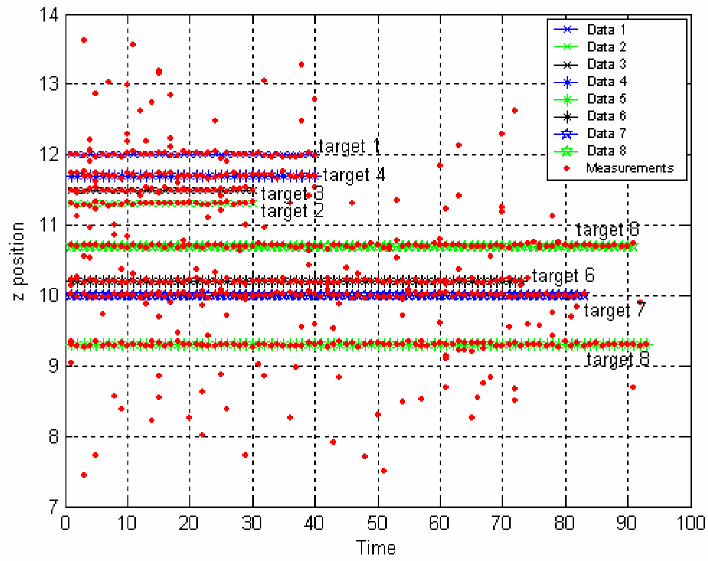

Figure 4-39 Z positions of the targets (Scenario 13)

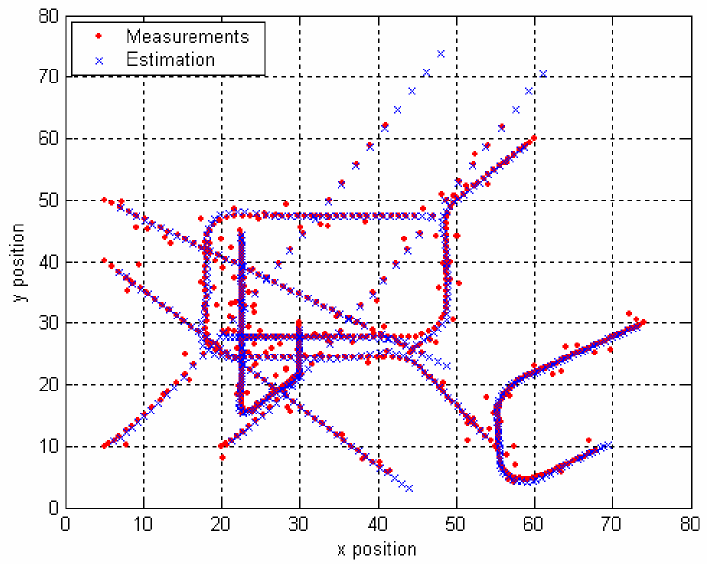

Figure 4-40 Output of the IMMJPDAF Algorithm (Scenario 13)

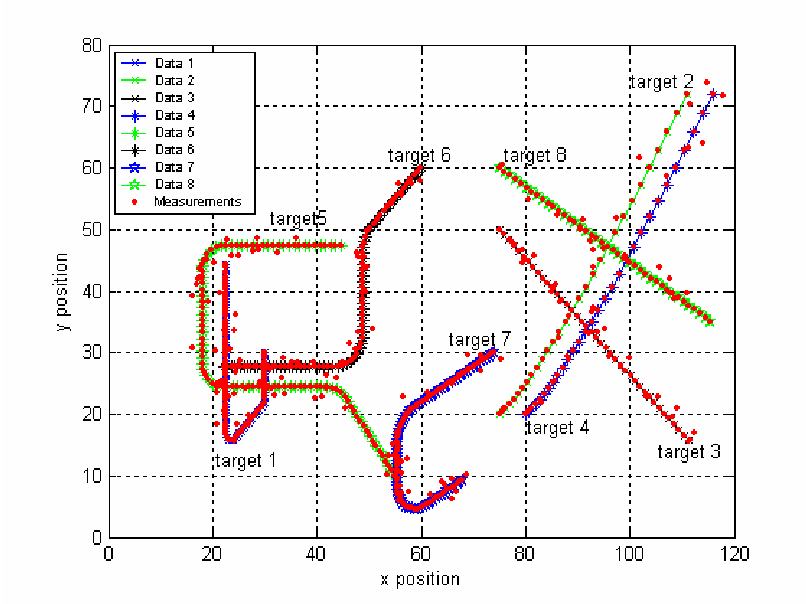

Figure 4-41 X and Y positions of the targets (Scenario 14)

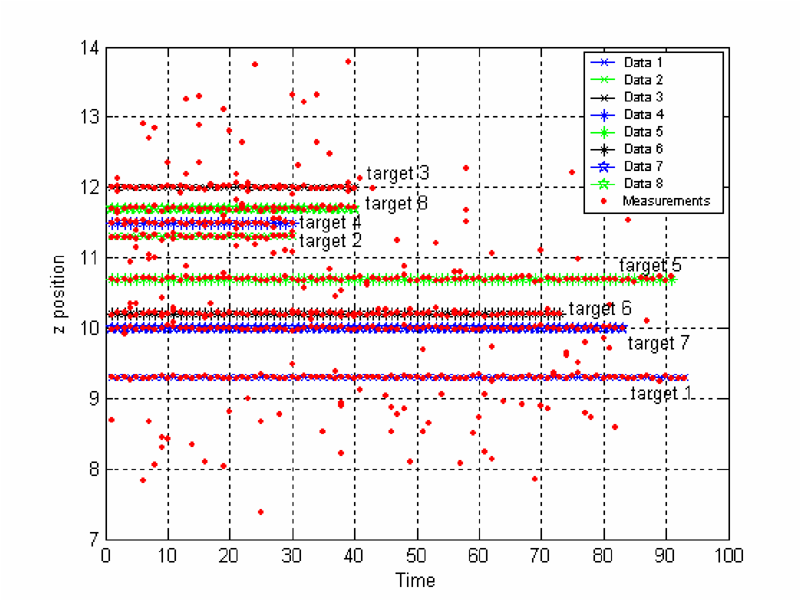

Figure 4-42 Z positions of the targets (Scenario 14)

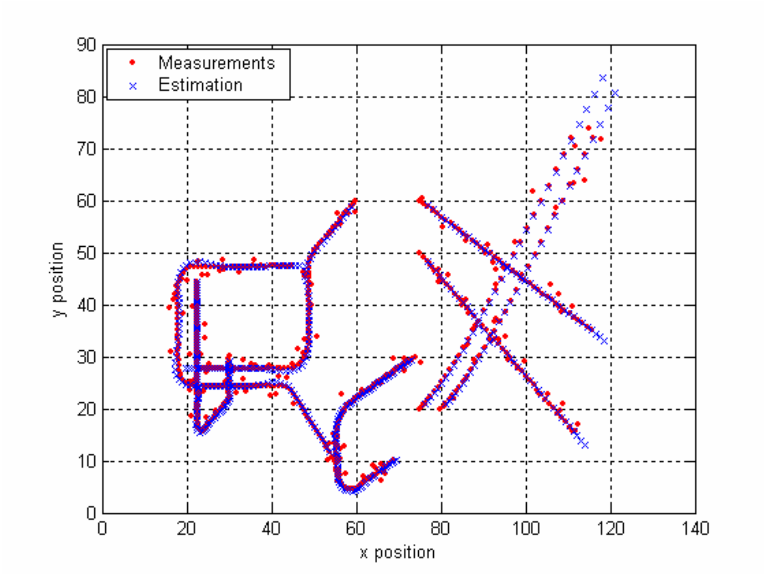

Figure 4-43 Output of the IMMJPDAF Algorithm (Scenario 14)

The number of targets is increased to nine in scenario 15. This is done by adding one target to the previous scenario. The positions of the targets are given in Figure 4-44 and Figure 4-45 and the results are shown in Figure 4-46.

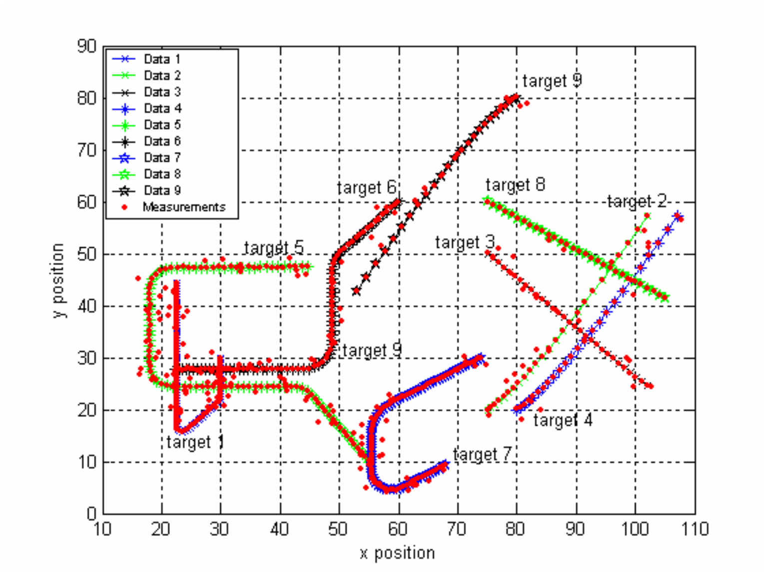

Figure 4-44 X and Y positions of the targets (Scenario 15)

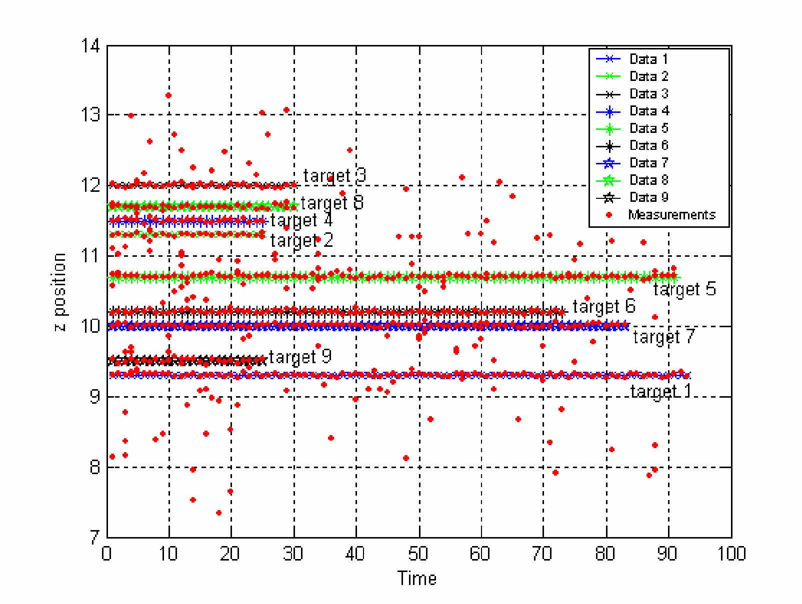

Figure 4-45 Z positions of the targets (Scenario 15)

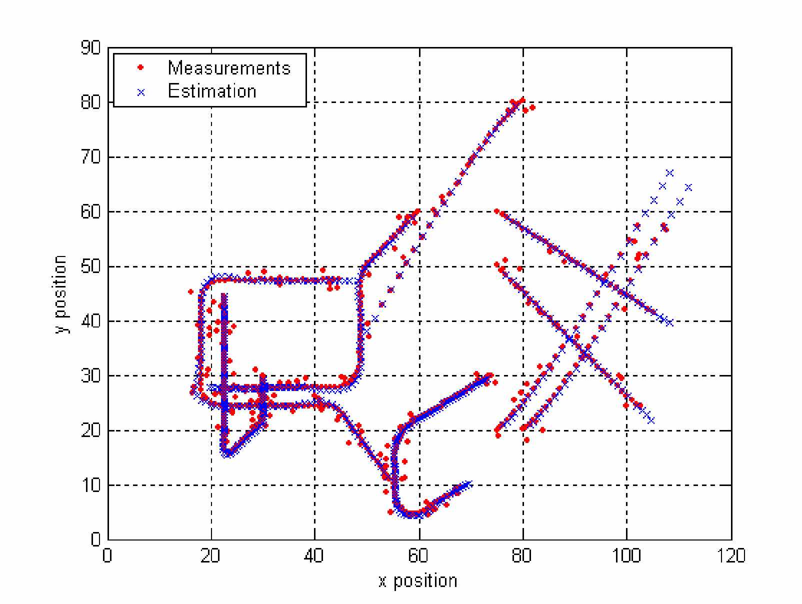

Figure 4-46 Output of the IMMJPDAF Algorithm (Scenario 15)

Finally, in the last scenario (scenario 16) ten targets are tracked at the same time. The positions of the targets are given in Figure 4-47 and Figure 4-48 and the results are shown in Figure 4-49.

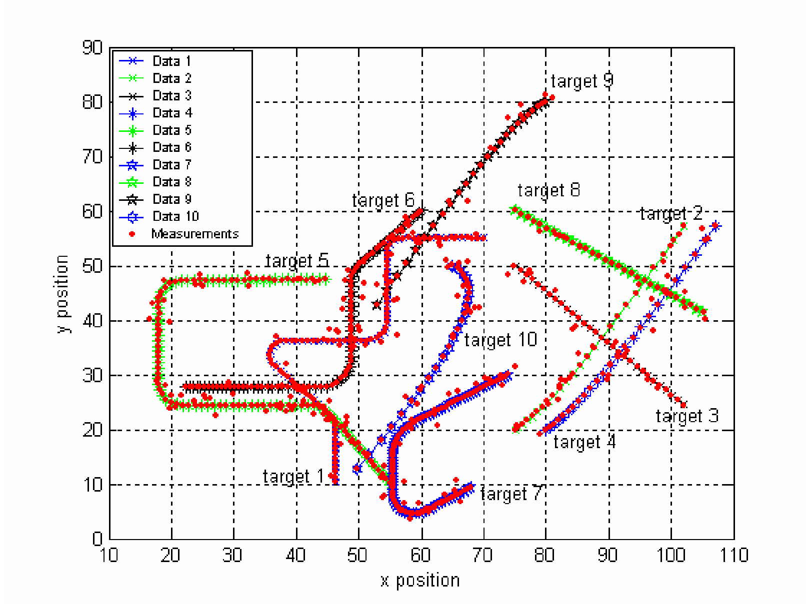

Figure 4-47 X and Y positions of the targets (Scenario 16)

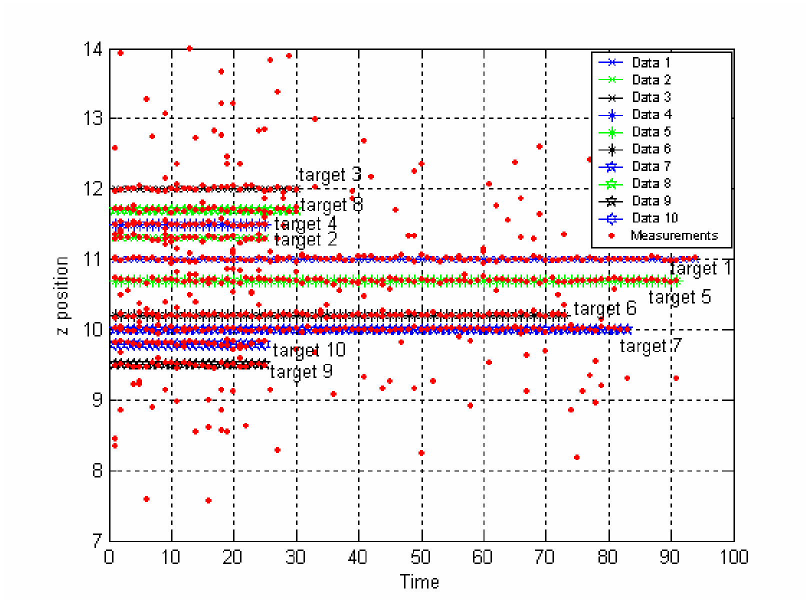

Figure 4-48 Z positions of the targets (Scenario 16)

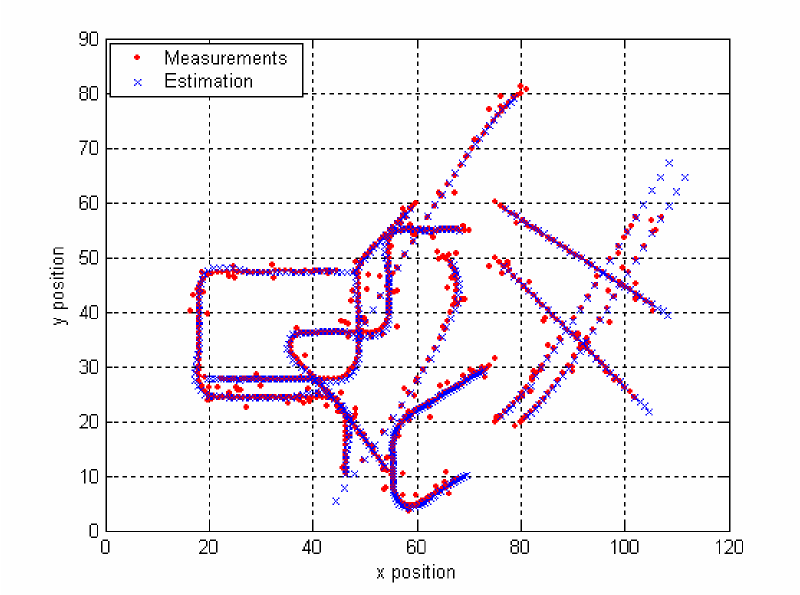

Figure 4-49 Output of the IMMJPDAF Algorithm (Scenario 16)

In all the scenarios above, the algorithms used for the implementation of the IMMPDAF and IMMJPDAF techniques have much in common. In both algorithms, the estimations for the target trajectories are held in track files. These track files are generated for every measurement obtained in the first scan. The track files record some variables such as state estimates, gate radius, Kalman parameters and scores. In the next time scan, the Kalman filters for the tracks are initialized and the scores are calculated for the tracks. This procedure is called track initiation and in the track initiation phase, it is seen that using nearest neighbor data association method with the Kalman filter is enough to determine the candidate targets for tracking. In this period, the score calculation is performed at each scan and a comparison is made with a track confirmation and deletion threshold. Those tracks whose scores greater than a confirmation threshold are marked and send to tracking algorithm. On the other hand, if the score of a track is below a deletion threshold than the track is dropped and its track file is deleted. By using this technique, it is seen that after two or three scans, the targets are determined and tracking begins.

In the tracking algorithm, measurements are associated with the targets according to the data association techniques, namely PDA and JPDA, used in the algorithm. The measurements, which are not assigned to tracks, are candidates of new targets and they are used to generate new track files or update the scores of the recent tracks. The scores of the targets are computed in the tracking algorithms as well in order to give a decision about the application of tracking algorithms. For some reason, it is possible not to get measurements from a target. In such a case, the score of the target is decreased and when it is below a certain threshold decided by the programmer, the target is dropped and tracking is finished. In the implementation of the algorithms, the targets, which are not assigned to any measurements, are continued to be tracked approximately for three radar scans. By setting the threshold, the scan number for tracking the targets, which have no measurement update can be set. This proves to be useful not to lose a target if we can not get information from the target for some reason.

The implemented tracking algorithms are compared with each other by running them in case of multiple intersecting targets. In the third scenario, there are two intersecting targets. When the first algorithm IMMPDAF is applied to this scenario, it is seen that one of the targets is dropped at the intersecting point. On the other hand, the second algorithm IMMJPDAF tracks both targets successfully. Therefore, we can say that IMMJPDAF method is capable of tracking intersecting multiple targets in case of false alarms while IMMPDAF method can not handle tracking multiple targets when there are overlapping validation gates. Therefore, use of the IMMJPDAF technique for multi target tracking has advantages in real life, since there can be situations where two or more targets may have overlapping validation gates at the same instant.

Another important issue is the computational time for real-time applications. In order to have the capability of tracking multiple targets, the computations should be done within a scan period, which is of the order of a few seconds. The computational time increases as the number of the targets increases with overlapping validation gates. This is due to the fact that the probability calculations of the joint events increase, when there is more than one target in the same validation gate. From the simulations carried out for the sixteen scenarios, it is seen that in the worst-case scenario the computations last for approximately 1 to 1.5 seconds. Therefore, we can say that the algorithms are suitable for real-time applications.

Finally, it is concluded that the implemented IMMJPDAF technique is suitable for tracking multiple targets in the presence of false alarms due to clutter in real-time applications.

CHAPTER 5

CONCLUSIONS

Multi target tracking in heavy clutter is a challenging task because each measurement must either be assigned to an existing track, or tentatively kept to initiate a new track. Multi target tracking differs from other state estimation problems by the fact that the measurement origin is also uncertain because of clutter or multiple targets. This uncertainty forces the tracking problem to be divided into two subtasks: data association and state estimation [35]. When new measurements are obtained the association between the measurements and the track list requires testing which measurement to track assignment is correct while estimating the target states.

Several approaches are proposed to solve this data association problem. The simplest method is the nearest neighbor approach. This method associates the measurement, which is located at a minimum distance with the track but unfortunately does not yield good results in the case of false alarms. In order to track a target in a cluttered environment, the probabilistic data association method, which uses a weighted average of all the measurements falling inside a target's validation region is considered and implemented in this thesis. It is seen from the simulations that the probabilistic data association (PDAF) approach is successful for a single target but can not track two intersecting targets whose validation regions overlap.

In order to handle this problem the JPDA algorithm, which is the extension of the probabilistic data association approach, is introduced and implemented. The JPDA generates all possible joint measurement to track assignments and calculates the a posteriori probability of each joint event. From these probabilities, the data association coefficients of each track are calculated and used to update the track estimates. In this thesis, the simulations, which include up to ten targets, are experimented and it is seen that the joint probabilistic data association filter (JPDAF) handles tracking of multiple targets in the cluttered environment successfully.

Generally, the targets being tracked are highly maneuverable. Therefore, it is important that the tracking algorithm must be capable of adapting rapidly to the maneuvers in order to achieve good performance [35]. In the state estimate of a target, usually Kalman filtering approach is used but a single filter lacks the ability to estimate the states of maneuvering targets. For this reason, a number of Kalman filters are gathered together where each filter is designed based on a different target kinematics. The interactive multiple model uses a finite number of filters to characterize the behavior of the target. In this thesis, the IMM method is used to estimate the targets' kinematics. Therefore, target trajectories are investigated in the thesis to design the filters that are going to be used in the IMM approach. In the implementation of this method, it is seen that two sub models, namely constant velocity and constant acceleration models are sufficient to achieve good estimation results.

In real time tracking, another important issue is the time constraints. The processor capacity that is going to be used for the implementations is crucial in the choice of the algorithms. In the thesis, the algorithms are optimized so that a single run of the algorithm to track multiple targets in a cluttered environment finishes in one scan time.

Finally, multi target tracking methods, which have been proposed since mid 1960s, have been thoroughly investigated and understood deeply in this thesis work.

REFERENCES

[1] Blackman, S. , Popoli, R. , “Design and Analysis of Modern Tracking Systems”, Artech House, 1999.

[2] Bar-Shalom, Y. , Li, X.R. , “Multitarget – Multisensor Tracking: Principles and Techniques”, YBS Publishing, 1995.

[3] Munir, A. , Atherton, D.P. , “Adaptive interacting multiple model algorithm for tracking a maneuvering target” IEEE Proc., Radar, Sonar, Navig., Vol. 142, No. 1, February 1995.

[4] Mazor, E. , Averbuch, A. , Bar-Shalom, Y. , Dayan, J. , “Interacting Multiple Model Methods in Target Tracking: A Survey” IEEE Trans. on Aerospace and Electronic Systems, Vol. 34, No. 1, January 1998.

[5] Li, X.R. , Jilkov, V.P. , “A Survey of Maneuvering Target Tracking: Dynamic Models”, Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, USA, April 2000.

[6] Slocumb, B.J. , West, P.D. , Shirey, T.N. , Kamen, E.W. , “Tracking A Maneuvering Target in the Presence of False Returns and ECM Using A Variable State Dimension Kalman Filter”, Proceedings of the American Control Conference, Seattle, Washington, June 1995.

[7] Bar-Shalom, Y. , Birmiwal, K. , “Variable Dimension Filter for Maneuvering Target Tracking”, IEEE Trans. On Aerospace and Electronic Systems, Vol.AES-18, No.5, September 1982.

[8] Musicki, D. , Suvorova, S., “Target Track Initiation Comparison and Optimization”, Proceedings of the Seventh International Conference on Fusion and Control, June 2004, pp. 24-28.

[9] Oliver E. D. , “Multiple Target Tracking with Multiple Frame, Probabilistic Data Association”, Signal and Data Processing of Small Targets 1993, Vol. 1954.

[10] Zhijian, H. , Leung, H. , Blanchette, M., “Evaluation of Data Association Techniques in a Real Multi-Target Radar Tracking Environment”, Signal and Data Processing of Small Targets 1995, Vol. 2561.

[11] Bar-Shalom, Y. , Li, R. , Kirubarajan, “Estimation with Applications to Tracking and Navigation”, John Wiley & Sons, 2001.

[12] Welch, G. , Bishop, G. , “An Introduction to the Kalman Filter”, UNC-Chapel Hill, TR 95-041, May 23, 2003.

[13] Simeonova, I. , Semerdjiev, T. , “Specific Features of IMM Tracking Filter Design”, Information & Security, An International Journal, Vol. 9, 2002, pp. 154-165.

[14] Chang, K. C. , Bar-Shalom, Y. , “Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers”, IEEE Transactions On Automatic Control, Vol. AC-29, No. 7, July 1984.

[15] Bar-Shalom, Y. , Li, X.R. , “Estimation and Tracking: Principles, Techniques, and Software”, Artech House, 1993.

[16] Blair, W.D. , Watson, G.D. , Kirubarajan, T. , Bar-Shalom, Y. , “Benchmark for Radar Allocation and Tracking in ECM”, IEEE Transactions On Aerospace and Electronic Systems, Vol. 34, No. 4, October 1998.

[17] Zhou, B. , Bose, N.K. , “An Efficient Algorithm for Data Association in Multitarget Tracking”, IEEE Transactions On Aerospace and Electronic Systems, Vol. 31, No. 1, January 1995.

[18] Sittler, R.W. , “An Optimal Data Association Problem in Surveillance Theory” , IEEE Transactions on Military Electronics, Vol. MIL-8, April 1964, pp. 125-139.

[19] Smith, P. , Buechler, G. , “A Branching Algorithm for Discriminating and Tracking Multiple Objects”, IEEE Transactions on Automatic Control, Vol. AC-20, No. 1, February 1975, pp. 101-104.

[20] Stein, J.J. , Blackman, S.S., “Generalized Correlation of Multi-Target Track Data”, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-11, No. 6, November 1975, pp. 1207-1217.

[21] Morefield, C.L. , “Application of 0-1 Integer Programming to Multitarget Tracking Problems”, IEEE Transactions on Automatic Control, Vol. AC-22, No. 6, June 1977, pp. 302-312.

[22] Sea, R.G. , “An Efficient Suboptimal Decision Procedure for Associating Sensor Data with Stored Tracks in Real-Time Surveillance Systems”, Proceedings of the 1971 IEEE Conference on Decision and Control, December 1971, pp. 33-37.

[23] Singer, R.A. , Stein, J.J. , “An Optimal Tracking Filter for Processing Sensor Data of Imprecisely Determined Origin in Surveillance Systems”, Proceedings of the 1971 IEEE Conference on Decision and Control, December 1971, pp. 171-175.

[24] Singer, R.A. , Sea, R.G. , “New Results in Optimizing Surveillance System Tracking and Data Correlation Performance in Dense Multitarget Environments”, IEEE Transactions on Automatic Control, Vol. AC-18, No. 6, December 1973, pp. 571-582.

[25] Bar-Shalom, Y. , “Tracking Methods in a Multitarget Environment”, IEEE Transactions on Automatic Control, Vol. AC-23, No. 4, August 1978, pp. 618-626.

[26] Bar-Shalom, Y. , Tse, E. , “Tracking in a Cluttered Environment with Probabilistic Data Association”, Proceedings of the Fourth Symposium on Nonlinear Estimation Theory and Its Applications, September 1973, pp. 13-22.

[27] Bar-Shalom, Y. , “Extension of the Probabilistic Data Association Filter to Multitarget Tracking”, Proceedings of the Fourth Symposium on Nonlinear Estimation Theory and Its Applications, September 1974, pp. 16-21.

[28] Bar-Shalom, Y. , Tse, E. , “Tracking in a Cluttered Environment with Probabilistic Data Association”, Automatica, Vol. 11, September 1975, pp. 451-460.

[29] Bar-Shalom, Y. , Birmiwal, K. , “Consistency and Robustness of PDAF for Target Tracking in Cluttered Environments”, Automatica, Vol. 19, No. 4, July 1983, pp. 431-437.

[30] Fortmann, T.E. , Bar-Shalom, Y. , Scheffé, M. , “Multitarget Tracking Using Joint Probabilistic Data Association”, Proceedings of the 19th IEEE Conference on Decision and Control, Vol. 2, December 1980, pp. 807-812.

[31] Chang, K.C. , Chong, C.Y. , Bar-Shalom, Y. , “Joint Probabilistic Data Association in Distributed Sensor Networks”, Proceedings of the 1985 American Control Conference, Vol. 2, June 1985, pp. 817-822.

[32] Chang, C.B. , Athans, M. , “State Estimation for Discrete System With Switching Parameters”, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-14, May 1978, pp. 418-425.

[33] Bloom, H.A.P. , Bar-Shalom, Y. , "The interacting multiple model algorithm for systems with Markovian switching coefficients", IEEE Transactions on Automatic Control, Vol. 33, No. 8, August 1988, pp. 780-783.

[34] Zhou, B. , Bose, N.K. , "An Efficient Algorithm for Data Association in Multitarget Tracking", IEEE Transactions on Aerospace and Electronic Systems, Vol. 31, No. 1, January 1995, pp. 458-468.

[35] Sarunic, P.W. , Evans, R.J. , "Adaptive update rate tracking using IMM nearest neighbor algorithm incorporating rapid re-looks", IEEE Proceedings Radar Sonar Navig., Vol. 144, No. 4, August 1997.