MULTI-SCAN DATA ASSOCIATION ALGORITHM FOR MULTITARGET TRACKING

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

BY

EMRE AĞIRNAS

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

DECEMBER 2004

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. Kerim DEMİRBAŞ
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU (METU, EE) _____

Prof. Dr. Kerim DEMİRBAŞ (METU, EE) _____

Asst. Prof. Dr. Ali Özgür YILMAZ (METU, EE) _____

Assoc. Prof. Dr. Bilgehan GÜVEN (METU, STAT) _____

M. Sc. Sevda ERDOĞDU (ASELSAN) _____

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 :

Signature :

ABSTRACT

MULTI-SCAN DATA ASSOCIATION ALGORITHM FOR MULTITARGET TRACKING

AĞIRNAS, Emre

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kerim DEMİRBAŞ

December 2004, 111 pages

Data association problem for multitarget tracking is determination of the relationship between targets and the incoming measurements from sensors of the target tracking system. Performance of a multitarget tracking system is strongly related to the chosen method for data association and target tracking algorithm. Incorrect data association effects state estimation of targets.

In this thesis, we propose a new multi-scan data association algorithm for multitarget tracking systems. This algorithm was implemented by using MATLAB programming tool. Performances of the new algorithm and JPDA method for multiple targets tracking are compared. During simulations linear models are used and the uncertainties in the sensor and motion models are modeled by Gaussian density. Simulation results are presented. Results show that the new algorithm's performance is better than that of JPDA method.

Moreover, a survey over target tracking literature is presented including basics of multitarget tracking systems and existing data association methods.

Keywords: Multitarget Tracking, Target Tracking, Data Association

ÖZ

ÇOKLU HEDEF TAKİBİ İÇİN ÇOK TARAMALI VERİ EŞLEME ALGORİTMASI

AĞIRNAS, Emre

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kerim DEMİRBAŞ

Aralık 2004, 111 sayfa

Çoklu hedef takibi için veri eşleme problemi hedefler ile hedef takip sisteminin algılayıcılarından gelen ölçümler arasındaki ilişkinin belirlenmesidir. Çoklu hedef takip sistemlerinin performansı seçilen veri eşleme ve hedef izleme algoritmalarına sıkıca bağlıdır. Doğru yapılmamış veri eşleme hedeflerin olası durumlarının tahmin edilmesini etkiler.

Bu tezde, çoklu hedef takip sistemleri için yeni bir çok taramalı veri eşleme algoritması önerilmiştir. Bu algoritma MATLAB programlama aracı kullanılarak gerçeklenmiştir. Çoklu hedef takibi için yeni algoritma ve JPDA metodunun performansları karşılaştırılmıştır. Simülasyonlar sırasında lineer modeller kullanılmış; algılayıcı ve hareket modellerindeki belirsizlikler Gaussian dağılımı ile modellenmiştir. Simülasyon sonuçları verilmiştir. Sonuçlar yeni algoritmanın performansının JPDA metodundan daha iyi olduğunu göstermiştir.

Bunlardan başka, çoklu hedef takip sistemleri ve mevcut veri eşleme metodlarını içeren hedef takip literatürü incelenmiştir.

Anahtar Kelimeler: Çoklu Hedef Takibi, Hedef Takibi, Veri Eşleme

To my parents Fatma and Cemal and sister Esin

ACKNOWLEDGMENTS

First of all, I wish to express my sincere gratitude to Prof. Dr. Kerim DEMİRBAŞ for his supervision, valuable guidance and helpful suggestions.

I am also grateful to ASELSAN Inc. for the facilities provided for the completion of this thesis.

I have to thank my dear friends Emre ÇAKIR, Özkan ÜNVER, Mehmet AKANGÖL, Timuçin BÜK and Özgür ERALP who were with me during thesis study period. I am also grateful to them for their understandings and support.

In addition, I want to give my thanks to Hasan MERCAN and again to Emre ÇAKIR. Small conversations, which I have made with them while I was studying my thesis out of working hours in ASELSAN, motivated me to work harder.

Lastly, and most importantly, I wish to thank my parents, Fatma AĞIRNAS and Cemal AĞIRNAS and my sister Esin AĞIRNAS for providing a loving environment for me. I am forever indebted to them for their understanding, endless patience, support and encouragement when it was most required. Without their help, finishing this thesis would have been a lot harder. It is to my parents and sister that I dedicate this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FOV : Field Of View

GNN : Global Nearest Neighbor Method

IR : Infra Red

JPDA : Joint Probabilistic Data Association

LS : Least Square

MAP : Maximum A Posteriori

MHT : Multiple Hypothesis Tracking

ML : Maximum Likelihood

MMSE: Minimum Mean Square Error

MTT : Multi Target Tracking

PDF : Probabilistic Distribution Function

TBD : Track Before Detect

CHAPTER 1

INTRODUCTION

In recent years, there has been increasing interest in the problem of multitarget tracking (MTT). Typical applications are Air Traffic Control (ATC), space surveillance, radar tracking, intelligent information retrieval, and video-based traffic surveillance [1]. The aim of a multitarget tracking system is to collect data of one or more sensors from the surveillance region containing potential targets of interest and to keep the field of view of the sensor (or sensors) looking at those targets. Typical sensors report measurements from targets of interest, background noise sources such as clutter, or internal noise sources such as thermal noise.

The fundamental problem of the multitarget tracking system is the process of data association. Data association is determination of the relationship between the measurements and tracks after elimination of unwanted measurements. A track is a state trajectory estimated from the measurements that have been associated with the same target [2].

There are various methods in literature (e.g., in [3], [4], [5]) for data association. Most data association methods require a measure of probability in order to evaluate alternative hypotheses [6]. The aim of basic Global Nearest Neighbor (GNN) approach is to find the single most likely hypothesis at each scan. In a cluttered environment the measurements coming from sensors may be from clutter, false alarms and targets of interest. As a result, the measurements other than real targets make it difficult to assign received measurements to existing targets. The accuracy of data association is critically important since any misassociation of the data to the targets will often lead to tracking failures such as the cross-over of tracks generated by parallel targets, the loss of tracks, as well as false track

initiations [7]. Moreover, if the clutter density is sufficiently large, false tracks can be produced. The resulting number of false tracks can overload the available computational resources and degrade the performance of the MTT systems. For these reasons, techniques dealing with data association have received much attention in MTT research [5, 6].

There are many data association techniques used in MTT systems ranging from the simpler nearest-neighbor approaches to the very complex multiple hypothesis tracking (MHT). The simpler techniques are commonly used in MTT systems, but their performance degrades in clutter. The more complex MHT provides improved performance, but it is difficult to implement and in clutter environments a large number of hypotheses may have to be maintained, which requires extensive computational resources. Because of these difficulties, some other algorithms having smaller computational requirements were developed [8, 9].

The goal of this thesis is to propose a new data association algorithm and to compare the performance of new algorithm with an existing multitarget tracking data association algorithm in literature.

Organization of the thesis is as below:

Chapter 2 is devoted to basics of multitarget tracking. The general structure of a multitarget tracking system is given.

In Chapter 3, existing multitarget tracking methods in literature are surveyed.

In Chapter 4, a new data association algorithm is developed.

In Chapter 5, simulation results and performance comparisons are given.

Chapter 6 is devoted to the conclusions of the results obtained from simulations made.

CHAPTER 2

THE BASICS OF MULTITARGET TRACKING

2.1 Introduction

Target tracking is an essential requirement for surveillance systems employing one or more sensors, together with computer subsystems, to interpret the environment. Typical sensor systems, such as radar, infrared (IR), and sonar, report measurements from diverse sources: targets of interest, background noise sources such as clutter, and internal error sources such as thermal noise [6].

The target tracking objective is to collect sensor data from a field of view (FOV) containing one or more potential targets of interest and to partition the sensor data into sets of observations, or tracks, that are produced by the same sources. Once tracks are formed and confirmed (so that background and other false targets are reduced), the number of targets can be estimated and quantities, such as target velocity, future predicted position, and target classification characteristics, can be computed for each track [5].

In the following section we introduce the basic terminology of target tracking.

2.2 Basic Target Tracking Terminology

The following terminology is adopted in multitarget tracking [8]:

State of a target: a vector of the smallest dimension that summarizes the past history of a target sufficiently in order to predict its future. It typically consists of kinematic components (position, velocity, etc.) as well as feature components (e.g. spectral characteristics). Often the state of a target cannot be measured directly.

Measurement: a noise-corrupted observation related to the state of a target, such as position and signal strength.

Track: a set of measurements over time that are associated with the same target.

Sensor: a device that observes the environment by reception of some signals.

Scan: a snapshot of a region of the environment obtained by the sensor at a point in time, called the sampling time.

Maneuver: unpredictable changes in target motion.

2.3 Elements of a Basic Multitarget Tracking System [5]

In this section, we introduce the basic elements of a basic MTT system discussed as in [5].

The representation of the elements of a simple MTT system is given in Figure 2.1. There is considerable overlap of the functions of these elements, but this representation provides a convenient partitioning which will be used to introduce the typical functions required for an MTT system. In this section the overview of the MTT problem and the interrelation of the elements will be discussed.

To describe the MTT system, it is assumed that the tracks have been formed on the previous scan and recursive processing is used. Incoming data (observations) are first considered for the update of existing tracks. Gating which will be explained in Section 2.3.3, tests determine which possible observation-to-track pairings are "reasonable", and a more refined data association algorithm is used to determine final pairings. (Data association will be explained conceptually in Section 2.3.4 and Chapter 3 will give much more detailed information about existing data association (or correlation) algorithms.)

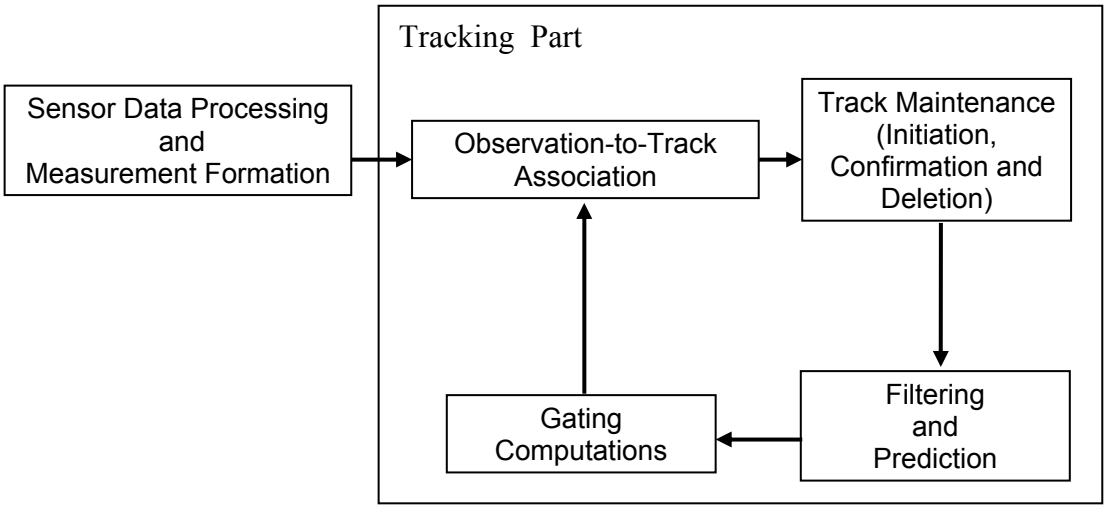


Figure 2-1 Basic elements of Multitarget Tracking System

Observations which are not assigned to existing tracks can initiate new tentative tracks. A tentative track becomes confirmed when the number of the observations included in the track satisfy confirmation criteria. Similarly, low quality tracks (i.e. tracks that any observations can not be assigned within a predetermined finite number of consecutive scans or within a finite total elapsed time since the last track update), are deleted. Finally, the future state of each established track and corresponding measurements at next scanning step are predicted. Gates, which can be defined as validation regions around these predicted quantities where the probability of correct observations falling inside is quite high, are placed and the processing cycle repeats. Now, we will discuss functions of the elements, block by block, in more detail as discussed in [6].

2.3.1 Sensor Data Processing and Measurement Formation Part of the MTT System

One of the important function performed by the Sensor Data Processing and Measurement Formation part is the determination of a decision rule on the return of sensor, which is sensed during the time on target, so as to discriminate between returns from targets of interest and returns from extraneous sources, such as potential false alarms produced by sensor noise and background clutter. The

simplest approach to the decision process is to compare the incoming signal power or intensity to a threshold level. Signals whose power or intensity levels are below that threshold are discarded, and those of higher levels are fed to the tracking part of MTT system. It is important to determine the appropriate threshold level that does not change the overall MTT system performance. For this purpose and for some other requirements, feedback may be provided between tracking and detection (Measurement Formation) parts of MTT system. In other words, results from tracking part of an MTT system can be used as feedback to affect the detection part. And, probably the most important use of this feedback is adaptive threshold control.

Detection (Measurement Formation) and tracking parts are often designed independently for MTT systems as shown in Figure 2.1. However these should be interrelated.

In order to simplify the design of MTT elements it must be known that sensor does not produce multiple simultaneous observations from the same source. But, in practice, there are multiple simultaneous observations from the same source due to the scanning requirements. For example, a radar may achieve a required elevation angle coverage by scanning two or more bars at different elevation angles. So the problem of receiving multiple simultaneous observations on different bars from the same target is encountered. These observations received on different bars from the same target can be combined and the composite observation is input to the tracking part of the MTT system. For this purpose a "Preprocessing Redundancy-Elimination" process is realized by "Detection (Observation Formation)" part of MTT system.

In addition to the problem of multiple observations from a single target, the problem of single observation from multiple targets may be produced too. For example, radar measurement techniques might not be able to resolve several closely spaced targets that are within the radar's beamwidth. However, some data processing techniques can be performed to determine whether multiple targets

exist within the radar's beamwidth, even if distinct measurements from all targets cannot be obtained. This information, as well as the results from previous data association (giving the expected number of targets within a given region), can be used to modify the current data association. This is another function of the "Detection (Measurement Formation)" part of MTT system.

2.3.2 Filtering and Prediction [10]

The filtering step incorporates the assigned observations into the updated track parameter estimates. For those tracks that were not assigned an observation, the previous predicted estimates become the filtered estimates. Then, predictions are made to the time when the next data scan is to be received. Thus, prediction quantities such as target position and velocity are of great importance because they define the center of the region representing the uncertainty of the prediction. [6]

Estimation is another important problem of target tracking. In this section of thesis we give an introduction to the basic estimation problem. The problem of non-random and random parameter estimation is introduced and additionally some common estimators are discussed.

Estimation can be defined as the operation of assigning a value to an unknown system state or parameter (e.g. position of a target, in our case) based on noise corrupted observations involving some function of the state or the parameter [2]. According to the choice of mathematical models used, we are able to classify an estimation problem into two basic groups:

1. Linear estimation

2. Nonlinear estimation

A linear estimation problem may be static (time-invariant) or dynamic (time-varying) and same classification is also true for nonlinear estimation problems.

Practical problems, including target tracking, are generally nonlinear dynamic estimation problems because either state equation or measurement equation or both are nonlinear. But, for most practical purposes, the overall MTT system design problems are the same, regardless whether a linear or a nonlinear model is used. For that reason, in this section linear estimation problems are described by considering an MTT problem before giving a brief summary of Kalman filtering which is the state estimator for discrete time linear dynamic systems driven by white noise. First of all, some general definitions will be given:

For the case of an MTT problem, assuming we are at scan time "k".

The measurement for each j^{th} scan can be denoted by:

$$Z(j) = h(j, x(j), w(j)) \qquad \text{j =1, 2, 3, ……, k} \qquad (2.1)$$

where $x(j)$ is the state of target and $w(j)$ is the measurement noise at the j^{th} scan.

The aim is to find a function

$$\hat{x}(k) \stackrel{\Delta}{=} x(k, Z^k) \qquad (2.2)$$

where Z^k denotes all previous observations, i.e.,

$$Z^k = \{z(j)\}_{j=1}^{k} \qquad (2.3)$$

Equation (2.2) is called the estimator and the value of this function for the k^{th} scan time is the estimate of $x(k)$. The estimation error corresponding to the estimate $\hat{x}(k)$ is

$$\tilde{x}(k) \stackrel{\Delta}{=} x(k) - \hat{x}(k) \qquad (2.4)$$

Parameter, x, (state of a target) can be time-invariant or time varying, as stated above. In the following section, linear estimation in time-invariant systems is discussed. In Section 2.3.2.2, time varying case will be analyzed.

2.3.2.1 Linear Estimation in Static Systems

There are two approaches one can use in the estimation of a time invariant parameter:

1. Non random (unknown constant): There is an unknown true value x. This is also called the Non-bayesian approach.

2. Random: The parameter is a random variable (or vector) with a prior (a priori) pdf p(x) -a realization of x according to p(x) is assumed to have occurred- this value then stays constant during the measurement process. This is also called the Bayesian approach.

Here, these two approaches will be analyzed in more detail:

The Non-Bayesian (Likelihood Function) Approach

In the non-Bayesian approach there is no prior pdf associated with the parameter and thus a posterior pdf can not be defined for it. In this case one has the pdf of the measurements conditioned on the parameter, called the likelihood function (LF) of the parameter, which is denoted by:

$$\lambda_Z(x) = p(Z|x) \tag{2.6}$$

or

$$\lambda_k(x) = p(Z^k|x) \tag{2.7}$$

It is a measurement of how "likely" a parameter value is for the observations that are made.

The likelihood function can be used in several ways to estimate x. Two of them are:

1. Maximum Likelihood Estimation (ML Estimation)

2. Least Square Estimation (LS Estimation)

Maximum Likelihood Estimator

A widely used decision rule for estimation of non-random parameters is the ML method. It simply aims to find the value of the parameter x which maximizes the likelihood function $\lambda(x)$. The ML-estimate is thus given by:

$$\hat{x}^{ML} = \arg\max_x \lambda(x) = \arg\max_x p(Z \mid x) \qquad (2.8)$$

Note that, while x is an unknown constant, $\hat{x}^{ML}$, being a function of the set of random observations Z, is a random variable.

Least Squares Estimator

Another estimator for non-random parameters is the LS method. Assuming that the sequence of measurements z is known to be related through the deterministic function h in the form

$$z_j = h(x, j) + w_j \qquad (2.9)$$

The least squares estimate of x after taking k measurements is the value which minimizes the sum of the squared errors:

$$\hat{x}^{LS} = \arg\min_{x} \sum_{j=1}^{k} \left[z_j - h(x, j) \right]^T \left[z_j - h(x, j) \right] \qquad (2.10)$$

There is no assumption made about the noise w_j. In case these are i.i.d zero-mean Gaussian random variables, then the LS-estimate coincides with the ML-estimate.

The Bayesian Approach

In this approach, one starts with the prior pdf of the parameter from which one can obtain its posterior (or a posterior pdf) using Bayes' formula:

$$p(x|Z) = \frac{p(Z|x)\,p(x)}{p(Z)} = \frac{1}{c}\,p(Z|x)\,p(x) \qquad (2.11)$$

where c is a normalization constant (does not depend on x).

The posterior pdf can be used in several ways to estimate x. Two of them are

1. Maximum A Posterior Estimation (MAP Estimation)

2. Minimum Mean Square Error Estimation (MMSE Estimation)

Maximum A Posteriori Estimator

In the case where the parameter x is considered to be random, a realization of x according to a pdf p(x) is assumed to have occurred. Bayes' rule is used to find the new a posterior pdf:

$$p(x \mid z) = \frac{p(z \mid x)\,p(x)}{p(z)} \qquad (2.12)$$

The a posteriori pdf p(x | z) is a measure of how likely a parameter value is, given the observations up to time step k and the a priori information p(x). Thus for a

random parameter, the MAP-estimator is the equivalent to the ML-estimator for a non-random parameter:

$$\hat{x}^{MAP} = \arg\max_x p(x \mid z) \tag{2.13}$$

$$= \arg\max_x \left[p(z \mid x)\, p(x) \right] \tag{2.14}$$

Obviously, if the a priori pdf p(x) is a uniform distribution (also called diffuse prior), then the ML-estimate and the MAP-estimate are going to be the same.

As described in [11], the non-Bayesian ML-estimator is nothing but the Bayesian MAP-estimator with complete a priori ignorance, reflected by the uniform a priori distribution. This provides a philosophically unifying view of the Bayesian and non-Bayesian approaches to estimation.

Note that the MAP-estimator is a particular case of Bayesian estimation, where the expected value of the cost function $K(\hat{x}, x)$

$$K(\hat{x}, x) = \left| \hat{x} - x \right|^2 = \begin{cases} 0, & \hat{x} = x \\ 1, & \hat{x} \neq x \end{cases} \tag{2.15}$$

is to be minimized. The expected value of the cost function is equal to the error probability given the observations z :

$$E\left[K(\hat{x}, x) \mid z \right] = P(X \neq x \mid Z = z) \tag{2.16}$$

Thus the MAP-estimator is given by:

$$\hat{x}^{MAP} = \arg\min_x P(X \neq x \mid Z = z) \tag{2.17}$$

$$= \arg\max_x P(X = x \mid Z = z) \tag{2.18}$$

$$= \arg\max_x P(X = x) \frac{p(z \mid x)}{p(z)} \tag{2.19}$$

$$= \arg\max_x P(X = x)\, p(z \mid x) \tag{2.20}$$

Minimum Mean-Squared Error Estimator

The equivalent to the LS-estimator for random parameters is obtained by minimizing the mean-squared error, the MMSE-estimator:

$$\hat{x}^{MMSE} = \arg\min_x E\left[(\hat{x} - x)^T (\hat{x} - x) \mid z\right] \tag{2.21}$$

where w_k is a sequence of random variables with known pdf. The solution to this problem is obtained by differentiating the expected value with respect to the estimate $\hat{x}$:

$$\frac{\partial}{\partial \hat{x}}\left[E\left[(\hat{x} - x)^T (\hat{x} - x) \mid z\right]\right] = \frac{\partial}{\partial \hat{x}}\left[\int_{-\infty}^{\infty} (\hat{x} - x)^T (\hat{x} - x)\, p(x \mid z) dx\right] \tag{2.22}$$

$$= 2\int_{-\infty}^{\infty} (\hat{x} - x) p(x \mid z) dx = 0 \tag{2.23}$$

Thus, the equation below is obtained

$$\hat{x}^{MMSE} = \int_{-\infty}^{\infty} x p(x \mid z) dx = E\left[x \mid z\right] \tag{2.24}$$

which means, that the MMSE-estimate is equal to the conditional mean of x given all observations z.

Note that the MMSE-estimation is a particular case of Bayesian estimation, where the expected value of the cost function $K(\hat{x}, x) = \left|\hat{x} - x\right|^2$ is to be minimized:

$$\hat{x}^{MMSE} = \arg\min_x E\left[K(\hat{x}, x) \mid z\right] = E\left[\left|\hat{x} - x\right|^2 \mid z\right] \tag{2.25}$$

Now, dynamic estimation problem will be explained:

2.3.2.2 Linear Estimation in Dynamic Systems [10]

Consider a discrete-time linear dynamic system described by a vector difference equation with additive white Gaussian noise that models "unpredictable disturbances." The dynamic (plant) equation is

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \qquad k = 0,1,.... \qquad (2.26)$$

where x(k) is the n_x-dimensional state vector, u(k) is an n_u-dimensional known input vector (e.g., control or sensor platform motion), and v(k), k = 0, 1, . . . , is the sequence of zero-mean white Gaussian process noise (also n_x vectors) with covariance

$$E[v(k)v(k)'] = Q(k) \qquad (2.27)$$

The measurement equation is

$$z(k) = H(k)x(k) + w(k) \qquad k = 1,.... \qquad (2.28)$$

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

$$E[w(k)w(k)'] = R(k) \qquad (2.29)$$

The matrices F, G, H, Q, and R are assumed known and possibly time-varying. In other words, the system can be time-varying and the noises non-stationary. The initial state x(0), 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 above constitutes the linear Gaussian (LG) assumption.

In the dynamic equation (2.26), the process noise term v(k) is sometimes taken as $\Gamma(k)v(k)$ with v(k) an n_v-vector and $\Gamma(k)$ a known n_x by n_v matrix. Then the covariance matrix of the disturbance in the state equation, which is Q(k) if v(k) enters directly, is to be replaced by

$$E\left[\left[\Gamma(k)v(k)\right]\left[\Gamma(k)v(k)\right]'\right] = \Gamma(k)Q(k)\Gamma(k)' \tag{2.30}$$

The linearity of (2.26) and (2.28) leads to the preservation of the Gaussian property of the state and measurements - this is a Gauss-Markov process.

The following notation will be used:

The conditional mean

$$\hat{x}(j \mid k) \overset{\Delta}{=} E\left[x(j) \mid Z^k\right] \tag{2.31}$$

where

$$Z^k \overset{\Delta}{=} \left\{z(i), i \le k\right\} \tag{2.32}$$

denotes the sequence of observations available at time k, is the

- Estimate of the state if j = k (also called filtered value)
- Smoothed value of the state if j < k
- Predicted value of the state if j > k

The estimation error is defined as

$$\tilde{x}(j \mid k) \overset{\Delta}{=} x(j) - \hat{x}(j \mid k) \tag{2.33}$$

The conditional covariance matrix of x(j) given the data Z^k or the covariance associated with the estimate (2.31) is

$$P(j \mid k) \stackrel{\Delta}{=} E\left[\left[x(j) - \hat{x}(j \mid k)\right]\left[x(j) - \hat{x}(j \mid k)' \mid Z^k\right]\right] = E\left[\tilde{x}(j \mid k)\tilde{x}(j \mid k)' \mid Z^k\right] \quad (2.34)$$

It was shown earlier that the MMSE criterion for estimation leads to the conditional mean as the optimal estimate.

Estimation Algorithm

In the previous section some basics of estimation concept are given. In this section we will analyze an algorithm related with Kalman filtering which is the state estimator for a linear discrete time time-varying (dynamic) system.

Estimation Algorithm, [2]

The algorithm start with the initial estimate x(0|0) of x(0) and the associated initial covariance P(0|0), assumed to be available. The second (conditioning) index 0 stands for Z^0, the initial information.

One cycle of dynamic estimation algorithm - the Kalman filter (KF)- will thus consist of mapping the estimate

$$\hat{x}(k \mid k) \stackrel{\Delta}{=} E\left[x(k) \mid Z^k\right] \quad (2.35)$$

which is the conditional mean of the state at time k (the current stage) given the observations up to and including time k, and the associated covariance matrix

$$P(k \mid k) \stackrel{\Delta}{=} E\left[\left[x(k) - \hat{x}(k \mid k)\right]\left[x(k) - \hat{x}(k \mid k)' \mid Z^k\right]\right] \quad (2.36)$$

into the corresponding variables at the next stage, namely, $\hat{x}(k+1 \mid k+1)$ and $P(k+1 \mid k+1)$.

This follows from the fact that a Gaussian random variable is fully characterized by its first two moments.

The values of past known inputs are subsumed in the conditioning, but (most of the time) will not be shown explicitly.

Detailed derivation of the dynamic estimation problem can be found in Appendix A. As an overview, it can be said that:

Under the Gaussian assumption for the initial state (or initial state error) and all the noises entering into the system, the Kalman filter is the optimal MMSE state estimator. If these random variables are not Gaussian and one has only their first two moments, then the Kalman filter algorithm is the best linear state estimator, that is, the LMMSE state estimator.

The flowchart of one cycle of the Kalman filter is presented in Figure 2.2. Note that at every stage (cycle) k the entire past is summarized by the sufficient statistic x (k|k) and the associated covariance P(k|k).

The left-side column represents the true system's evolution from the state at time k to the state at time k+1 with the input u(k) and the process noise v(k). The measurement follows from the new state and the noise w(k+1). The known input (e.g., control, platform motion, or sensor pointing) enters (usually) the system with the knowledge of the latest state estimate and is used by the state estimate to obtain the predicted value for the state at the next time.

The state estimation cycle consists of the following:

1. State and measurement prediction (also called time update) and

2. State update (also called measurement update)

The state update requires the filter gain, obtained in the course of the covariance calculations. The covariance calculations are independent of the state (and control-assumed to be known) and can, therefore, be performed off-line.

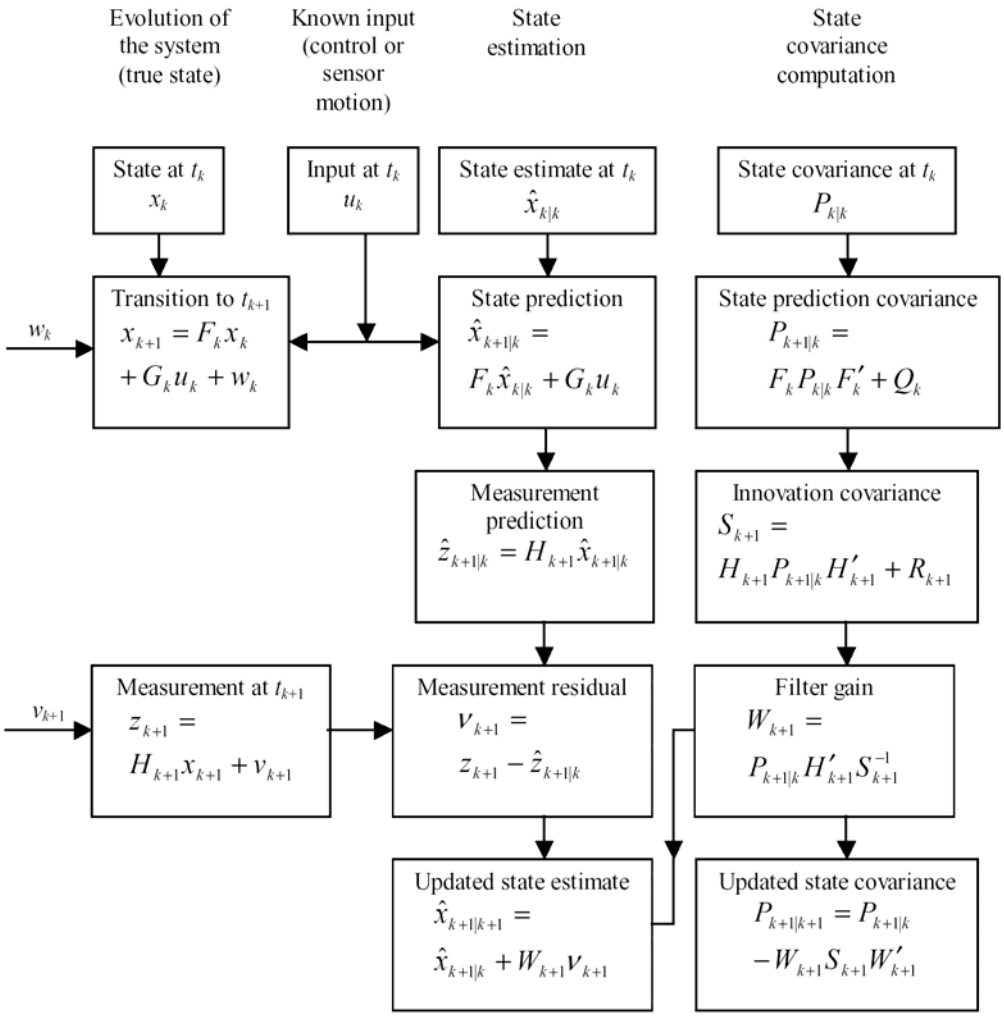


Figure 2-2 The Flow Chart of one cycle of Kalman Filtering

As a summary, we can say that filtering and prediction are used to estimate present and future target kinematic quantities such as position, velocity, and acceleration.

Fundamental (not sophisticated) techniques which are used for filtering and prediction purposes, assume widely separated targets in a sparse false-alarm background so that the errors introduced by uncertain observation to-track correlation can be ignored.

2.3.3 Gating, [5]

Gating is a technique for eliminating unlikely observation-to-track pairings. A gate is formed about the predicted measurement and all observations that satisfy the gating relationship (fall within the gate) are considered for track update. The manner in which the observations are actually chosen to update the track depends on the data association method but most data association methods utilize gating in order to reduce later computation. The actual gating process is typically a progression of operations with increasing complexity. One computationally efficient approach is to first use a coarse form of gating, such as binning. Using the binning approach, the measurement space is divided into a grid of cells (or bins). Then, a track is only compared with observations in its bin and in adjacent bins. Bin size can be chosen adaptively based on track quality and available computational resources. Efficient techniques for the implementation of coarse gating are presented in [12-14].

Define $\tilde{y}$ to be the residual (or innovation) vector, which is the difference between the actual y and the expected $\hat{y}$ measurement vectors. In general, at scan k, the residual vector is

$$\tilde{y} = y(k) - h(\hat{x}(k \mid k-1)) \tag{2.37}$$

$$y(k) = h(x(k)) + w(k) \tag{2.38}$$

where the measurement h(x) is a nonlinear function of the target state x and v is zero-mean, white Gaussian measurement noise with covariance matrix R. In the case of a linear measurement

$$y(k) = Hx(k) + w(k) \tag{2.39}$$

In either the linear or nonlinear measurement case, the residual covariance matrix is defined to be S = HPH' + R where P is the one-step prediction covariance matrix. Assume that the measurement is of dimension M. Then, defining d^2 to be the norm of the residual (or innovation) vector, $d^2 \stackrel{\Delta}{=} \tilde{y}' S^{-1} \tilde{y}$, the M-dimensional Gaussian probability density for the residual is

$$f(\tilde{y}) = \frac{e^{-d^2/2}}{(2\pi)^{M/2}\sqrt{|S|}} \tag{2.39}$$

where |S| = the determinant of S.

Rectangular Gates

Probably the simplest gating technique is to define rectangular regions such that an observation (**y**, with elements y_l) is said to satisfy the gates of a given track if all elements, $\hat{y}_l$, of the residual vector $\hat{y}$ satisfy the relationship:

$$\left|y_l - \hat{y}_l\right| = \left|\tilde{y}_l\right| \le K_{Gl}\sigma_r \tag{2.40}$$

where σ_r is the residual standard deviation as defined in terms of the measurement (σ_o^2) and the prediction (σ_p^2) variances:

$$\sigma_r = \sqrt{\sigma_o^2 + \sigma_p^2} \tag{2.41}$$

Finally, note that the prediction variance (σ_p^2) is the appropriate diagonal element taken from the Kalman filter covariance matrix.

Choice of the rectangular gating coefficients K_{Gl} has been discussed in more detail in [5]. However, in practice, a typical choice will be K_{Gl} > 3.0. This large choice of gating coefficient is typically made in order to compensate for the approximations involved in modeling the target dynamics through the Kalman filter covariance matrix. Finally, due to gate volume considerations discussed in [5], rectangular gating should be followed by the more precise ellipsoid gating.

Ellipsoidal Gates

Define a gate (G) such that association is allowed if the following relationship is satisfied by the norm (d^2) of the residual vector:

$$d^2 = \tilde{y}' S^{-1} \tilde{y} \leq G \tag{2.42}$$

where

d^2 is the norm of the residual vector

S is the residual covariance matrix

$\tilde{y}$ is the measurement residual vector

A maximum likelihood gate, proposed in [6], is defined as

$$d^2 \leq G_0 = 2 \ln \left[\frac{P_D}{(1 - P_D)(2\pi)^{M/2} \beta \sqrt{|S|}} \right] \tag{2.43}$$

where

P_D is the probability of detection.

β is the new source density which is expected value of the new sources (true targets and false alarms) that arise per unit volume per unit scan time.

M is the measurement dimension.

As discussed further in [5], d^2 is typically assumed to have the x_M^2 distribution for correct observation-to-track pairing. Using these properties, Table 2.1 presents expressions for the probability, $P_G(M)$, of a valid observation of dimension M satisfying the gate G.

Table 2-1 Probability, Pa, of Valid Observation Falling Within M-Dimensional Gate of Size G

M	$P_G(M)$
1	$2gc(\sqrt{G})$
2	$1-\exp(-G/2)$
3	$2gc(\sqrt{G})-\sqrt{2G/\pi}\exp(-G/2)$
4	$1-(1+G/2)\exp(-G/2)$
5	$2gc(\sqrt{G})-(1+G/3)\sqrt{2G/\pi}\exp(-G/2)$
6	$1-1/2(G^2/4+G+2)\exp(-G/2)$

where $gc(x)=\frac{1}{\sqrt{2\pi}}\int_0^x \exp(-u^2/2)du$ is standard Gaussian probability integral

Finally, the volume within the ellipsoidal gate is

$$V_G(M) = C_M \sqrt{|S|} G^{M/2} \tag{2.44}$$

Thus, the normalized volume is

$$V_{G2}(M) = C_M G^{M/2} \tag{2.45}$$

where

$$C_M = \frac{\pi^{M/2}}{\Gamma(\frac{M}{2}+1)} = \begin{cases} \dfrac{\pi^{M/2}}{(\frac{M}{2})!}, & M \ \ even \\ \dfrac{2^{M+1}(\frac{M}{2}+1)!\pi^{(M-1)/2}}{(M+1)!}, & M \ \ odd \end{cases} \tag{2.46}$$

In particular,

$C_1 = 2$, $C_2 = \pi$, $C_3 = 4\pi/3$, $C_4 = \pi^2/2$

Maneuver Gating

The rectangular and ellipsoidal gates just discussed are based on the Kalman filter covariance matrix, which represents an idealized statistical target dynamics model. Thus, practical systems frequently allow for a "last resort" gate that may, under certain conditions, be applied to potential observation-to-track pairings that fail the conventional Kalman filter covariance matrix-based gate tests. These tests may, for example, be applied if no observation is found to satisfy the conventional gate tests for a given track. Maneuver gates are defined to more closely model the most severe potential maneuvers that targets can perform.

2.3.4 Observation-to-Track Association

The gating, observation-to-track association and track maintenance functions shown in Figure 2.1 are part of the overall data association function. First, gating is

used to determine which observations are valid candidates to update existing tracks. Gating is performed primarily to reduce unnecessary computations by the association and maintenance functions that follow.

Figure 2.3 illustrates gating for two closely spaced targets and four observations. Note that the gates may overlap for closely spaced targets. Gates are established and gating is performed in the following general manner.

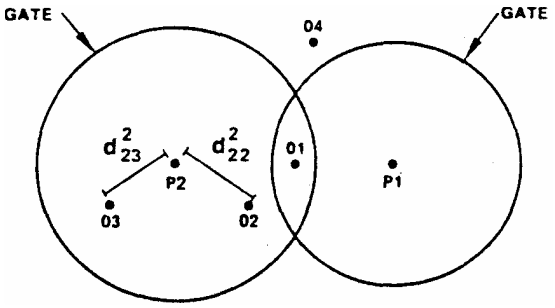
1. Estimates are made of what the measured quantity should be at the time of the next observation. This usually means kinematic quantities, such as position as estimated by tracking filters. This process is called prediction, and statistics describing the accuracy of these estimates are required. Also, an estimate of measurement accuracy is required.

2. The difference between each measurement and its corresponding estimate is formed. It is often useful to form a total distance d_{ij}^{2*} from track i to observation j. Thus, a normalization process is required whereby the differences in each of the component measurements are squared, divided by the variances of the expected differences, and summed to form a total normalized (or statistical) distance. For example, if range (R) and angle (θ) are measured, the normalized distance is

$$d^2 = \frac{(R_p - R_o)^2}{\sigma_R^2} + \frac{(\theta_p - \theta_o)^2}{\sigma_\theta^2} \qquad (2.48)$$

where (R_p, θ_p) is the predicted position, (R_o, θ_o) is the measured position, σ_R^2 is the variance of $R_p - R_o$, and σ_θ^2 is the variance of $\theta_p - \theta_o$,

* The quantity d_{ij} is actually a squared distance but, for convenience, it will be referred to simply as a distance.



01,02,03,04 = OBSERVATION POSITIONS
P1, P2 = PREDICTED TARGET POSITIONS
d_{22}^2 = DISTANCE FROM P2 TO 02

Figure 2-3 Gating Example

3. A maximum error between estimate and measurement is formed for all measured quantities by using the estimate and measurement accuracy statistics. The computed differences are compared to the computed maximum allowable error. Normally, these allowable errors are set to at least the three-standard-deviation level for assumed-zero-mean Gaussian statistics. If the differences are less than the maximum allowable errors, the observation satisfies the gate. Alternatively, the gating can be carried out by using the normalized distance, but computations can usually be saved if the measurement components are first examined individually.

The normalized distance just defined is the mathematical result of a likelihood expression. In general, there is also a term to penalize poor-quality tracks (with large prediction errors).

The association takes the observation-to-track pairings that satisfied gating and determines which observation-to-track assignments will actually be made. The simplest approach that is commonly applied will be denoted global nearest neighbor (GNN). This approach determines a unique assignment so that at most one observation can be used to update a given track, and an observation can be used to update at most a single track. This assignment is typically made in order to minimize cost (such as total summed distance) or to maximize likelihood. The

solution can be placed in the form of the well-known assignment problem for which optimal, as well as computationally less demanding suboptimal, solutions have been developed.

2.3.5 Track Maintenance (Initiation, Confirmation, and Deletion)

Track maintenance (Figure 2.1) refers to the functions of track initiation, confirmation, and deletion. A simple approach to track initiation is to start new tracks on those observations that are not assigned to existing tracks. However, a more preferable method will start tentative tracks on all observations and use subsequent data to determine which of these newly initiated tracks are valid.

Once a tentative track is formed, a confirmation logic is usually required because the probability of a single observation being from an extraneous source is too high for immediate confirmation. Thus, it is usually required that at least one other observation be assigned to a tentative track before the track is considered to be confirmed. The gate size and the length of time allowed for that confirming observation can be chosen as functions of the confidence in the validity of the original observation. A typical simple rule for track confirmation is that M correlating observations should be received within N scans. However, a much better approach is to define a track score function and compare this score with an appropriately chosen track confirmation threshold.

A track that is not updated becomes degraded, and it must be deleted if not updated within some reasonable interval. If a sufficiently long time elapses without detection, the target will probably no longer be within the scan volume. Also, even if the lack of detections is consistent with an assumed low probability of detection, it might be best to delete a track just because of its low quality. A typical simple rule is to delete a track after N_D consecutive scans have produced no updating observations. Again, however, the use of a track score function is more general and is more readily applied to varying detection capabilities (such as with multiple sensor systems). Also, the track score reflects the quality of the update so that updates that barely satisfy the gate may actually decrease the score.

CHAPTER 3

DATA ASSOCIATION METHODS

3.1 Introduction

In this chapter we describe the data association problem that is how we combine measurements and existing tracks or to initiate new track. The observation-to-track correlation or data association is a problem of great importance for multiple target tracking applications. Several methods have been discussed in estimation and tracking literature [3], [4], [5]. In general multitarget tracking deals with state estimation of an unknown number of targets. Some methods are special cases which assume that the number of targets is constant or known. The observations are considered to originate from targets if detected and also from clutter. The term clutter is used for returns from the objects other than targets, producing false alarms. In many applications only one measurement is assumed from each target object.

Five basic characteristics distinguish the various methodologies that have been developed for data association. The successive subsections describe these characteristics, the corresponding methods, and their applicability and expected performance as given in [6].

3.1.1 Unique-Neighbor versus All-Neighbors Data Association

The unique-neighbor approach to data association determines a pairing so that, at most, one observation can be paired with a previously established track. Also, a given observation can be used only once, either to update an existing track or to start a new track. This method is based on likelihood theory, and the goal is to minimize an overall distance function that considers all observation-to-track

pairings that satisfy a preliminary gating test. However, because for a dense target environment many sets of pairings may be computed to have similar likelihoods, the probability of error typically is large for this approach, unless multiple hypotheses are allowed.

The alternative all-neighbors approach incorporates all observations within a gated region about the predicted track position into the update of that track. Also, a given observation can be used to update multiple tracks. Track update then is based on a probabilistically determined weighted sum of all observations within the gated region.

The all-neighbors approach performs an averaging over observation-to-track data association hypotheses that have roughly comparable likelihoods. This approach is most applicable for track maintenance of widely spaced targets in a clutter background.

3.1.2 Sequential versus Deferred Decision Logic

An ideal approach that would process all observations (from all time) together is typically not feasible computationally. Thus, a more standard approach has been to perform processing in a recursive (or sequential) manner as data are received. For example, a set of observations may be collected from a time scan of a radar and observation-to-track data association performed on these observations during the period when the sensor is collecting the next scan of data. Using this approach, data association decisions are irrevocable.

A deferred decision approach to data association, allows the final decision on difficult data association situations to be postponed until more information is received. In effect, this approach allows a modified version of batch processing. Alternative hypotheses are formed and evaluated when later data are received. This approach clearly has the potential for achieving a much higher correct decision probability than the sequential decision method. However, to maintain

computational feasibility, an intricate logic to delete (prune) unlikely hypotheses and to combine similar hypotheses is required.

The all-neighbors approaches can also be implemented using a deferred decision approach. For example, a given track can still be updated with a probabilistically weighted set of observations that were received on a previous scan. However, the computation of weightings can use data received on subsequent scans.

3.1.3 Use of Group Information

Many targets of interest, particularly for military applications, tend to travel in groups (or formations). Intuitively, this information should be directly applicable for data association and track filtering. For example, even if individual tracks are maintained on aircraft in a formation, the fact that the aircraft are flying together should provide additional smoothing on the velocity estimates. Similarly, if a newly resolved target first appears close to a group, the new track formed on that target can be initiated with the group velocity.

3.1.4 Use of Thresholding

As discussed earlier, the measurement process, shown in Figure 2.1, will include detection thresholding so that a set of observations can be produced. This observation set would include only the output from detection bins, such as pixels in an IR sensor or range/range rate bins in a radar, in which the measured signal intensity, amplitude, or SNR exceeded the detection threshold.

Although there may be some feedback from the tracker to the signal processing system, the use of detection thresholding clearly separates the functions of detection and tracking. Also, thresholding leads to an irretrievable loss of information whereby, for example, a signal just below the threshold level is completely ignored. Thus, as an alternative to thresholding, an approach has been proposed that considers the detection and track confirmation processes to occur, in effect, simultaneously [16]. As stated in [16], for this approach "the detection of a target and the estimation of its state are an inseparable part of the same decision

process." This approach can be contrasted with the traditional sequential approach whereby a detection device is used to generate observations and then track initiation and confirmation logic operates on these observations to form tracks. This alternative approach is frequently denoted track before detect (TBD), although, in effect, the tracking and detection processes occur simultaneously.

3.1.5 State Estimation without Explicit Data Association

Standard MTT methods require some type of observation-to-track data association before target state estimation. Recently, several approaches have been proposed whereby target state estimation is performed directly without an explicit observation-to-track assignment being performed. These approaches typically involve the propagation of the PDF associated with the presence of targets in various bins in target state space [17, 18]. The TBD approach is an example of state estimation without explicit data association.

3.2 Data Association Methods [6]

The existing data association methods presented in [6] will be discussed in the following three sections.

3.2.1 Global Nearest Neighbor Method

The Global Nearest Neighbor (GNN) method (which just maintains the single most likely hypothesis) is the simplest, and probably the most widely applied, method for data association. This method may also be referred to as single hypothesis tracking or sequential most probable hypothesis tracking. In this method the input data is handled in a purely sequential manner. For each new data set, the goal is to identify the most likely assignment of observations to existing tracks and to the new source hypotheses. Using GNN, track scores can be computed and used for track confirmation and deletion. However, conventional GNN systems have typically used a simple M/N rule for confirmation and N_D consecutive misses for track deletion [5].

A major problem in GNN method system design is the definition of a solution for observation-to-track association. Association conflict situations, as illustrated in Figure 3.1, occur when there is more than one observation in a track gate or an observation is in the gate of more than one track. This problem is typically addressed through the formation and the solution of an assignment matrix.

It is necessary to define the elements of this matrix, before discussing solutions to the observation-to-track assignment matrix. As discussed in [5], one approach is to define a generalized statistical distance for the assignment of observation of j track i:

$$d_{G_{ij}}^{2} = d_{ij}^{2} + \ln\left[\left|S_{ij}\right|\right] \tag{3.1}$$

where d_{ij}^{2} is as defined in (2.42) and $\ln\left[\left|S_{ij}\right|\right]$ is the logarithm of the determinant of the residual covariance matrix. Then, $d_{G_{ij}}^{2}$ is used as a cost for those observation-to-track assignments that satisfy the gate. Either an arbitrarily large cost can be assigned to pairings that do not satisfy the gate or the logic can be designed to ignore these pairings.

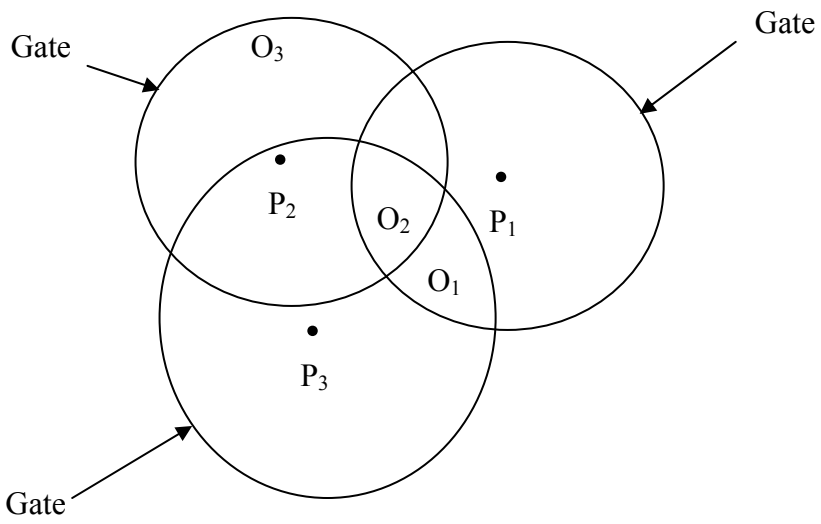


Figure 3-1 Example of an association conflict situation

The distance function given in (3.1) contains the term $\ln\left[\left|S_{ij}\right|\right]$, which results from the Gaussian likelihood calculations [5]. This term has the effect of penalizing tracks with greater prediction uncertainty. Thus, it ensures that tracks with missing observations, and resulting larger covariance matrices, do not "steal" observation from higher quality tracks.

Given a cost (or distance) function, such as the one obtained from (3.1), the typical approach is to form an observation-to-track assignment matrix. Table 3.1 gives a hypothetical assignment matrix based on the example given in Figure 3.1. The non allowed assignments, which failed the gate test, are denoted by X. Then, one approach is to find the solution that gives the maximum number of possible assignments and, given the maximum number of assignments, minimizes the total cost. This "optimal" solution can be found by enumeration for simple cases, such as shown in Figure 3.1 and Table 3.1, but the solution generally requires more efficient as assignment solution methods.

A more general association solution can be defined using the score function approach. Using this method, the "optimal" assignment may not maximize the possible number of observation-to-track assignments. For example, one good assignment may be preferable to two lower quality assignments, in which the gate tests were barely satisfied.

The more general association technique has been given in [5]. The resulting assignment matrix is of the form shown in Table 3.2. From this table, note that this more general assignment matrix approach includes the assignment of observations to the new tracks, which represent the new source hypotheses. A new source could either be a false alarm (or clutter point) or a newly detected target.

In the case of no measured signal or attribute terms, the elements of the general assignment matrix are defined in terms of a score gain given by

$$a_{ij} = G_{ij} - d_{ij}^2 \tag{3.2}$$

Table 3-1 Assignment Matrix for Example of Figure 3.1

Tracks	Observations Q1	Q2	Q3
T1	(9)	6	X
T2	X	3	(10)
T3	8	(4)	X

X = observations outside gate; ◯ = optimal solution

Table 3-2 Example of Generalized Assignment Matrix

	Observations		
Tracks	**01**	**02**	**03**
T1	a_{11}	a_{12}	a_{13}
T2	a_{21}	a_{22}	a_{23}
T3	a_{31}	a_{32}	a_{33}
New track 1	0	X	X
New track 2	X	0	X
New track 3	X	X	0

The term a_{ij} is the margin by which the statistical distance, d_{ij}^2 passed the gate, G_{ij}. The gate G_{ij} for the association of observation j to track i is given by (2.43). The elements corresponding to the assignment of an observation to a new track are zero.

The elements of the assignment matrix given in Table 3.2 represent the score gains associated with the various observation-to-track assignments. Thus, the optimal solution will now be that set of assignments that produces the maximum score gain. This is in contrast to the minimum cost (distance) solution required for the method illustrated by Table 3.1. However, an assignment method designed to produce minimum cost can be applied to find the maximum score gain by simple algebraic operations on the elements. For example, costs can be defined to be the negative of the score gains. Then, the desired solution again becomes the minimum cost solution.

As discussed earlier, the standard GNN method considers the current scan observation-to-track assignment. Thus, the cost (or alternatively the score gain) used in the assignment matrix is computed using only the current scan's data and the predicted track position. An alternative approach, uses retrodicted probabilities such that final observation-to-track assignment on scan k is made using data on subsequent scans k + 1, . . . , k + n; n ≤ 1, as well as the current scan's data. This

method also requires solution of the assignment matrix but the elements of the assignment matrix are computed using data from several scans.

Various ad hoc techniques, such as branching, discussed in [5], can be used to improve GNN data association, but modern computational capabilities have made the application of more accurate methods, such as JPDA and MHT, feasible. Thus, the GNN method is becoming obsolete except for nondemanding applications. However, solutions to the assignment problem are required for many tracking applications, as well as GNN association.

3.2.2 Probabilistic Data Association Method

Consider the case of a single established target track in the presence of extraneous returns, such as clutter. Define P_D to be the probability of detection and P_G to be the probability that a correct (target) return will fall within the track gate. Typically $P_G \approx 1.0$, but for generality P_G will be retained. Assume the extraneous return density to be Poisson with density β (note that β now includes new targets and false returns, $\beta = \beta_{NT} + \beta_{FT}$).

Given N observations within the gate of track i, there are N + 1 hypotheses that can be formed for that track. The first (denoted H_0) is the case in which none of the observations is valid. Using the results of [5], the probability of H_0 is proportional to p'_{i0} where

$$p'_{i0} = \beta^N \left(1 - P_D P_G\right) \tag{3.3}$$

Similarly, the probability of the hypothesis H_j (j = 1, 2, . . ., N) that observation j is the valid return is proportional to

$$p'_{ij} = \beta^{N-1} P_D P_G \frac{e^{-\frac{d_{ij}^2}{2}}}{P_G (2\pi)^{\frac{M}{2}} \sqrt{|S_i|}} = \frac{\beta^{N-1} P_D e^{-\frac{d_{ij}^2}{2}}}{(2\pi)^{\frac{M}{2}} \sqrt{|S_i|}} \tag{3.4}$$

Finally, the probabilities p_{ij} associated with the N + 1 hypotheses that assign observation j to track i are computed through the normalization equation:

$$p_{ij} = \frac{p'_{ij}}{\sum_{l=0}^{N} p'_{il}} \tag{3.5}$$

The factor β^{N-1} cancels during the normalization process, and thus is not required in the computation of p_{ij}. Therefore, a convenient simplified form becomes

$$p_{ij} = \begin{cases} \dfrac{b}{b + \sum_{l=1}^{N} \alpha_{il}}, & j = 0 \quad (no \quad valid \quad observation) \\ \dfrac{\alpha_{il}}{b + \sum_{l=1}^{N} \alpha_{il}}, & 1 \le j \le N \end{cases} \tag{3.6}$$

where

$$\begin{aligned} b &= (1 - P_D P_G)\beta(2\pi)^{\frac{M}{2}} \sqrt{|S_i|} \\ \alpha_{ij} &= P_D e^{-\frac{d_{ij}^2}{2}} \end{aligned} \tag{3.7}$$

The next step, after computation of the probabilities using (3.3), (3.4) and (3.5), is to merge hypotheses. The resulting method is taken from Bar-Shalom and Tse [19]. First, the residual, for use in the Kalman filter update equation, is a weighted sum of the residuals associated with the N observations:

$$\tilde{y}_i(k) = \sum_{j=1}^{N} p_{ij} \tilde{y}_{ij}(k) \tag{3.8}$$

where

$$\tilde{y}_{ij}(k) = y_j(k) - H\hat{x}_i(k \mid k-1)$$

$$y_j(k) = \quad observation \quad j \quad received \quad at \quad scan \quad k$$

For notational convenience, the subscript i, denoting track i, is omitted whenever there ought to be no ambiguity in referring to the Kalman filter quantities. Then, the standard Kalman filter update equation is used:

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + K(k)\tilde{y}(k) \tag{3.8}$$

with the gain, K(k), computed using Kalman filter equations. However, the covariance, at scan k, is modified according to the equation:

$$P(k \mid k) = P^0(k \mid k) + dP(k) \tag{3.9}$$

where $P^0(k \mid k)$ is the Kalman covariance that would be computed if a single return were present and known to be the correct association, and $dP(k)$ is an increment added to reflect the effect of uncertain association. The equations defining $P^0(k \mid k)$ and $dP(k)$ are

$$\begin{aligned} P^0(k \mid k) &= p_{i0}P(k \mid k-1) + (1 - p_{i0})P^*(k \mid k) \\ dP(k) &= K(k)\left[\sum p_{ij}\tilde{y}_{ij}\tilde{y}_{ij}^{'} - \tilde{y}_i\tilde{y}_i^{'}\right]K^{'}(k) \end{aligned} \tag{3.10}$$

and $P^*(k \mid k)$ is the standard Kalman filter covariance matrix:

$$P^*(k \mid k) = \left[I - K(k)H\right]P(k \mid k-1) \tag{3.11}$$

The term $dP(k)$ increases the covariance according to both the a posteriori probabilities (or uncertainties) and the spread of the observations found within the

track gate. For example, for a single observation within the gate, using (3.8) and (3.10) and dropping subscript i, for track i, the obtained equations are

$$\begin{aligned} \tilde{y} &= p_1 \tilde{y}_1 \\ dP(k) &= K\left[p_1 \tilde{y}_1 \tilde{y}_1' - \tilde{y}\tilde{y}'\right]K' = p_1(1-p_1)K\tilde{y}_1\tilde{y}_1'K' \end{aligned} \tag{3.12}$$

which gives a maximum correction for uncertainty when the probability that the observation is valid (P_1) is 0.5. Similarly, if there are two measurements in the gate and each has probability 0.5 of validity ($P_1 = P_2 = 0.5$. $p_0 = 0$), the covariance correction term becomes

$$dP = 0.25K(\tilde{y}_1 - \tilde{y}_2)(\tilde{y}_1 - \tilde{y}_2)'K' \tag{3.13}$$

The probability computations defined in (3.3), (3.4) and (3.6) require an assumed value, β, for the extraneous return density. A non parametric version of the PDA uses the number of returns, N, in the track gate to estimate β for each set of observations [3] from

$$\beta = N/V_G \tag{3.14}$$

where V_G is the gate volume computed using (2.44, 2.45 and 2.46).

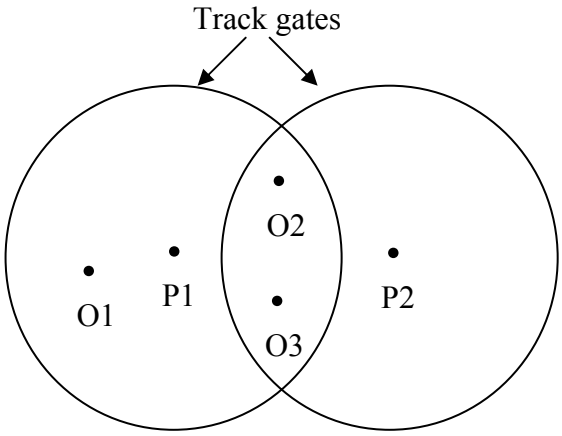
Equations (3.6) through (3.10) define the PDA. Results for a single target in clutter [19] have shown a significant decrease in the number of lost tracks when the PDA method is compared with a standard GNN association method. However, an extension, discussed next, is required if multiple target tracks are considered.

3.2.3 Joint Probabilistic Data Association Method

The JPDA method is identical to the PDA except that the association probabilities are now computed using all observations and all tracks. The state estimation gain

and covariance are computed as before, using (3.8) through (3.10). The probability computation of (3.6) must be extended, however, to include multiple tracks.

Consider the typical type of conflict situation, such as that shown in Figure 3.2, in which multiple observations fall within the validation gates of multiple target tracks. In this example, three observations (01, 02, 03) are within the gate of the predicted position (P1) of the first track (T1), while only 02 and 03 are within the gate of the second track (T2). The JPDA will compute a weighted residual for T1 based on the three observations, but the weights for 02 and 03 will be reduced to account for their presence within the gate of T2. The residual for T2 will be formed using 02 and 03.



01, 02, 03 = Observation positions
P1, P2 = Predicted target position

Figure 3-2 Example of correlation conflict condition suitable for JPDA

A hypothesis matrix approach will be used to illustrate the required probability computations from an example based on Figure 3.2. Table 3.3 gives the hypothesis matrix (consisting of all sets of feasible observation-to-track assignments) and the associated hypothesis probabilities for this example. The numbers represent the observations assigned to the tracks, and 0 represents the assignment of no

observation to a given track. Also, g_{ij} is the Gaussian likelihood function associated with the assignment of observation j to track i, as defined:

$$g_{ij} = \frac{e^{-\frac{d_{ij}^2}{2}}}{(2\pi)^{\frac{M}{2}}\sqrt{|S_{ij}|}} \tag{3.16}$$

where d_{ij}^2 and S_{ij} are the normalized distance and residual covariance matrix, respectively, associated with the assignment of observation j to track i.

Table 3.3 shows the structure for computing the hypothesis probabilities $P'(H_l)$. Defining N_0 and N_T to be the numbers of observations and tracks, respectively, note that certain common factors may appear in all $P'(H_l)$. If $N_0 > N_T$, as in our example, the common factor is $\beta^{(N_0-N_T)}$ while if $N_T > N_0$, the common factor will be $(1 - P_D)^{(N_T-N_0)}$. The same probability of detection (P_D) has been assumed for all tracks, but the extension to track-dependent P_D is direct. Finally, the normalized probabilities $P(H_l)$ are computed in the standard manner (where N_H is the total number of hypotheses):

$$P(H_l) = \frac{P'(H_l)}{\sum_{j=1}^{N_H} P'(H_j)} \tag{3.17}$$

Table 3-3 Hypothesis Matrix for Example of Figure 3-2

Hypothesis Number	Track Number		Hypothesis Likelihood p(Hj)	Likelihood (Normalized Probability) for Example
	1	2		
1	0	0	$(1 - P_D)^2\beta^3$	2.4 x 10^{-6} (0.011)
2	1	0	$g_{11}\, P_D\, (1 - P_D)\beta^2$	1.82 x 10^{-5} (0.086)
3	2	0	$g_{12}\, P_D\, (1 - P_D)\beta^2$	1.11 x 10^{-5} (0.053)
4	3	0	$g_{13}\, P_D\, (1 - P_D)\beta^2$	4.1 x 10^{-6} (0.019)
5	0	2	$g_{22}\, P_D\, (1 - P_D)\beta^2$	8.6 x 10^{-6} (0.041)
6	1	2	$g_{11} g_{22}\, P_D^2\, \beta$	6.47 x 10^{-5} (0.306)
7	3	2	$g_{13}\, g_{22}\, P_D^2\, \beta$	1.44 x 10^{-5} (0.068)
8	0	3	$g_{23}\, P_D\, (1 - P_D)\beta^2$	6.7 X 10^{-6} (0.032)
9	1	3	$g_{11}\, g_{23}\, P_D^2\, \beta$	5.04 x 10^{-5} (0.239)
10	2	3	$g_{12}\, g_{23}\, P_D^2\, \beta$	3.06 x 10^{-5} (0.145)

As an example, consider the condition shown in Figure 3.2 with two-dimensional measurement, in which

$$M = 2, \qquad \sqrt{|S_i|} = \sigma_{\tilde{y}}\sigma_{\tilde{x}}$$

and with numerical values

$$\sigma_{\tilde{y}} = \sigma_{\tilde{x}} = 1, \qquad \beta = 0.03 \qquad P_D = 0.7$$

$$d_{11}^2 = 1, \qquad d_{12}^2 = 2, \qquad d_{13}^2 = 4$$

$$d_{22}^2 = 2.5, \qquad d_{23}^2 = 3$$

Table 3.3 gives the probabilities associated with the various hypotheses.

To compute the probability p_{ij} that observation j should be assigned to track i, a sum is taken over the probabilities from those hypotheses in which this assignment

occurs. For example, for track 1 (noting that j = 0 refers to no assignment), the probabilities are

$$p_{10} = P(H_1) + P(H_5) + P(H_8) = 0.011 + 0.041 + 0.032 = 0.084$$
$$p_{11} = P(H_2) + P(H_6) + P(H_9) = 0.086 + 0.306 + 0.239 = 0.631$$
$$p_{12} = P(H_3) + P(H_{10}) = 0.053 + 0.145 = 0.198$$
$$p_{13} = P(H_4) + P(H_7) = 0.019 + 0.068 = 0.087$$

Similarly, for track 2,

$$p_{20} = P(H_1) + P(H_2) + P(H_3) + P(H_4) = 0.169$$
$$p_{21} = 0$$
$$p_{22} = P(H_5) + P(H_6) + P(H_7) = 0.415$$
$$p_{23} = P(H_8) + P(H_9) + P(H_{10}) = 0.416$$

Thus, as expected, the most heavily weighted events are computed to be the assignment of 01 to T1 and either 02 or 03 to T2.

If an observation does not fall within the gate of a given track, the associated probability is, of course, taken to be zero (p_{21} = 0 for our example). The remaining calculations are then performed using (3.8) through (3.10) to update filter estimates and covariance terms.

Table 3.3 essentially enumerates the hypotheses in order to compute the observation-to-track assignment a posteriori probabilities. This process of forming hypotheses (and resultant probabilities) could potentially be done more efficiently using the method for computing the N-best solutions to the assignment problem discussed in section 6.5.2 of reference [6].

3.2.4 Multiple Hypothesis Tracking Method

Multiple Hypotheses Tracking (MHT) is a deferred decision logic in which alternative data association hypotheses are formed whenever there are observation-

to-track conflict situations such as shown in Figures 3.1 and 3.2. Then, rather than combining these hypotheses, as in the JPDA method, the hypotheses are propagated in anticipation that subsequent data will resolve the uncertainty. The full MHT approach is, in effect, a formal implementation of previously developed methods such as those presented in [15].

The original MHT method, denoted Reid's algorithm, was first presented in [20]. A detailed discussion of an implementation of Reid's algorithm has been given in Chapters 10 and 14 of [5]. In this section we will give an overview of MHT methods.

The manner in which MHT forms multiple hypotheses and manages these hypotheses is illustrated by again referring to the example given in Figure 3.2 and by referring to the overall structure shown in Figure 3.3. As an example, assume that tracks T1 and T2 with predicted positions P1 and P2, represent a hypothesis (H_1) prior to the receipt of the three observations (O1, O2, O3) on scan k. Then, as shown in Table 3.3, there are 10 feasible hypotheses that can be generated from the initial single hypothesis. The relative likelihoods (and resultant probabilities) of each of these hypotheses can be computed in a direct manner, such as shown in Table 3.3, or through the use of the updated track scores. The latter, score-based method is more computationally convenient because scores are added whereas probabilities would be multiplied.

As discussed in the previous section, the JPDA method uses the hypotheses probabilities to compute observation-to-track assignment probabilities and then updates each track with a probabilistically weighted composite of all observations within the track gates. Thus, the 10 hypotheses would be effectively combined and a single two-track hypothesis maintained. On the other hand, MHT will choose to maintain a subset of the 10 hypotheses and defer the decision so that subsequent scans of data can be used to resolve the uncertainty at the current time. Note that each of the 10 hypotheses can be further expanded to include the new target tracks that can be formed by the observations that are not used by the hypotheses for

track update. Finally, note that each of the 10 hypotheses defined in Table 3.3 may spawn another set of hypotheses upon the receipt of more data on the next scan $k+1$ and so on for subsequent scans.

There is clearly a potential combinatory explosion in the number of hypotheses (and tracks within those hypotheses) that an MHT system can generate. Thus, a number of techniques have been developed to keep this potential growth in check. These techniques, include clustering, hypothesis and track pruning (deletion), and track merging.

There are two basic approaches to MHT implementation. The first (hypothesis-oriented) approach follows the original work of Reid [20]. It maintains the hypothesis, structure from scan to scan and continually expands and cuts back (prunes) the hypotheses as new data are received. At each scan, a set of hypotheses will be carried, over from the previous scan and composed of one or more tracks that are compatible with all other tracks in the hypothesis. Compatible tracks are defined to be tracks that do not share any common observations. Then, on the receipt of new data, each hypothesis is expanded into a set of new hypotheses by considering all observation-to-track assignments for the tracks within the hypothesis. Again, as new hypotheses are formed, the compatibility constraint for tracks within a hypothesis is maintained. Direct, but inefficient, hypothesis expansion methods are given in [5, 20] but the recently proposed use of Murty's algorithm [21, 22] will reduce the number of low probability hypotheses that are unnecessarily formed and later deleted.

An alternative (track-oriented) approach [23-26] does not maintain hypotheses from scan to scan. The tracks formed on each scan are reformed into hypotheses and the tracks that survive pruning are predicted to the next scan where the process continues.

Next, following Figures 3.3 and 3.4, the operations of MHT, which are required by both implementation methods, are given. Then, an overview of the differences

between the hypothesis-oriented and track-oriented approaches is given. A detailed implementation of the hypothesis-oriented method is given in Chapters 10 and 14 of [5]. Cox and Hingorani [21] discuss how an efficient implementation of Reid's algorithm can be achieved using Murty's method for finding the N-best solutions to the assignment problem.

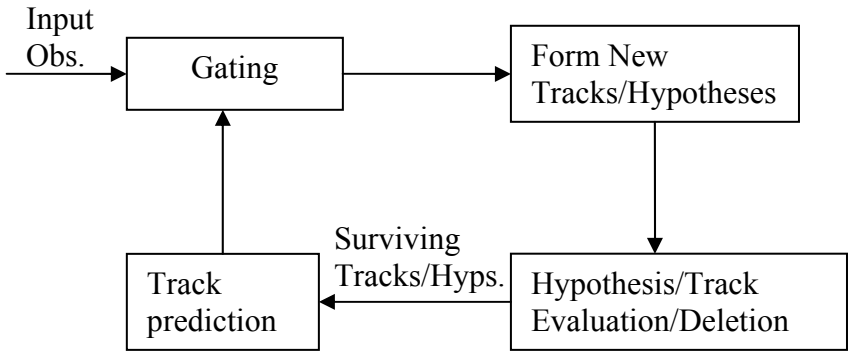


Figure 3-3 MHT logic overview.

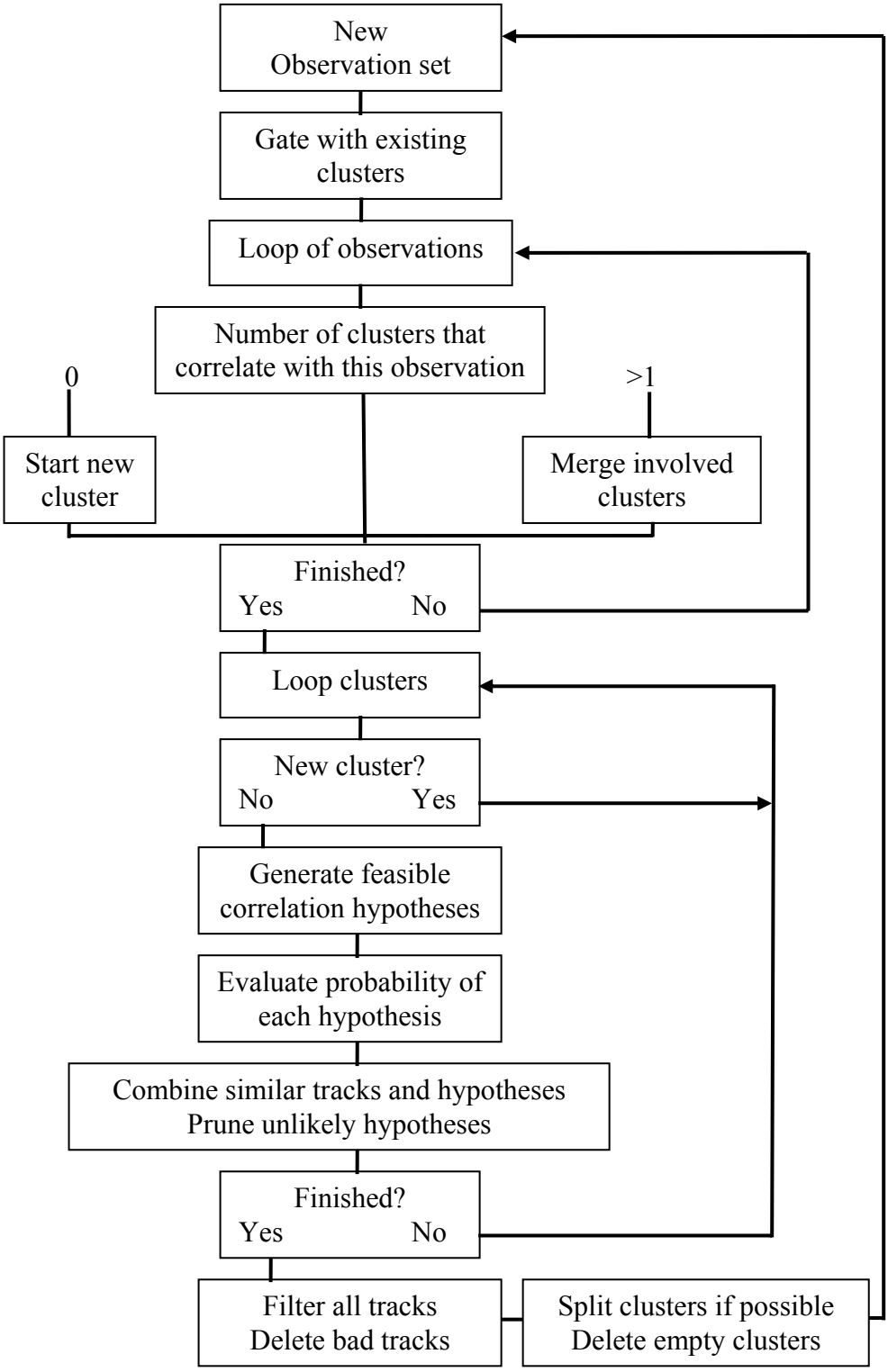


Figure 3-4 High-level flow chart of MHT algorithm

MHT Tracker Elements

Hypotheses are collections of compatible tracks. Tracks are defined to be incompatible if they share one or more common observations. This definition may be relaxed for certain cases in which it is suspected that a single observation was produced by two or more unresolved targets. Track quality is described by a track score. Using either approach, hypotheses can be evaluated by computing hypothesis scores, which are formed by summing the track scores of all tracks contained within a given hypothesis. Thus, referring to Figure 3.3, hypotheses are evaluated (by converting hypothesis scores to probabilities), unlikely hypotheses are deleted, and track probabilities are computed from the surviving hypotheses. Low score tracks are deleted and the surviving tracks are filtered and predicted to the next scan where, after gating tests, new tracks are formed. The mathematical relationships by which hypothesis scores are converted to probabilities so that low probability hypotheses and tracks can be deleted have been presented in [5].

The operation of clustering is performed to reduce the number of hypotheses that must be generated and evaluated. Clusters are collections of incompatible tracks that are linked by common observations. A cluster can include tracks that do not share observations directly. Thus, if track 1 shares an observation with track 2 and track 2 shares an observation with track 3, all three tracks are in the same cluster. It is most efficient to maintain cluster information from scan to scan, but as shown in Figure 3.4, reclustering and the formation of new clusters must be performed as each new set of data is received.

Once clustering has been performed, the processing within each cluster can be done independently from other clusters. Thus, using a parallel processing implementation, the processing for each cluster can be assigned to a separate processor. Within each cluster, the next step is to form and evaluate new hypotheses and to delete unlikely hypotheses and tracks. Then, the surviving tracks are filtered. Note that the track scores can be computed and the subsequent logic leading to track deletion can be performed before Kalman filtering. Thus, the

computationally intensive Kalman filtering operations should only be performed on those tracks that survive pruning.

An additional operation that is typically performed in all MHT algorithms is the merging of similar tracks. Track similarity can be defined in terms of common observations and/or similar state estimates. The merging operation may be performed, using common observation history, before hypothesis formation and evaluation so that fewer tracks (and resultant hypotheses) need be considered. A typical rule would be to merge tracks that have the last N observations (where $N > 3$) in common. Also, track merging based on track state estimation similarity may be performed after Kalman filtering.

Alternative MHT implementations

Table 3.4 summarizes the major differences between the conventional (hypothesis-oriented) and the track-oriented approach to MHT implementation. As indicated in Table 3.4, the main difference between these approaches is that the conventional approach maintains hypotheses from scan to scan and uses these hypotheses to directly spawn new hypotheses. On the other hand, the track oriented approach reforms the hypotheses from the tracks on each scan.

Table 3-4 Summary of Differences Between Alternative MHT Implementation Approaches

Conventional MHT (Reid's Algorithm [20])	**Track-Oriented MHT (Kurien [23])**
Maintains high probability hypotheses from scan to scan	Only high probability tracks maintained
Current hypotheses spawn new hypotheses	Hypotheses reformed after tracks formed
Low probability hypotheses deleted	Low probability tracks deleted

A major potential advantage of the track-oriented approach is that hypothesis formation can be restricted to higher quality tracks. Low score tracks are deleted before hypotheses are formed. Also, following the bi-level approach of [25, 26], the hypothesis formation method only uses positive tracks in the initial hypothesis formation. This greatly reduces the search space with no loss of completeness since the most significant hypotheses will contain only positive score tracks. Only the higher score (probability) hypotheses are retained. Then, the probabilities of the negative score tracks are computed from their compatibility with the hypotheses that have been retained.

CHAPTER 4

MULTI-SCAN DATA ASSOCIATION METHOD

4.1 Introduction

Existing data association methods were surveyed in previous chapter. All of these methods require a measure of probability in order to evaluate alternative hypotheses. The goal of the GNN method is to find the single most likely hypothesis for the assignment of observations to existing tracks and to new track initiations. The JPDA method forms multiple hypotheses after each scan of data and then these hypotheses are combined before the next scan of data is processed. MHT method forms multiple hypotheses and propagates them from scan to scan.

These methods are essentially sequential (recursive) methods, but only in multiple hypothesis testing method, difficult decisions may be deferred until more data are received. Another characteristic of these methods is that, at each scan time, for every track one must determine gates which can be defined as validation regions around predicted observation positions where the probability of correct (real) observations falling inside is quite high. In addition, in all these methods, different data association approaches are used for assigning measured observations to existing tracks.

In this chapter, a new data association algorithm is developed. This algorithm is a multi-scan, nearest-neighbor algorithm.

In this algorithm, the probable states for each target are calculated. The probable states for consecutive scan times are connected and a tree structure is obtained. The branches of this tree represent the possible paths for the target. By using the node values (states) of the tree, the expected measurements for the target are

calculated. Expected measurements for a target also form a tree structure. For multiple scans, distances between the expected measurements constituting each branch and measurements coming from sensor are calculated. Sum of distances for each scan time are found. The measurements, which give the minimum sum of distances, are assigned to the target.

The multi-scan data association algorithm has the following assumptions:

1. The state of each target is assumed to evolve in time according to the following equation:

   $$x(k+1)=F(k)x(k) + v(k) \qquad (4.1)$$

   and the corresponding measurement is given by:

   $$z(k)=H(k)x(k) + w(k) \qquad (4.2)$$

   where

   x(k+1) is an n-dimensional state vector of the target at next scan time “k+1", assuming we are at scan time "k".

   z(k) is an n-dimensional measurement vector corresponding to the state vector, x(k), at scan time "k".

   F(k) and H(k) are n by n matrices which are known and possibly time varying.

   v(k) is an n-dimensional zero-mean white Gaussian process noise vector.

   w(k) is an n-dimensional zero-mean white Gaussian measurement noise vector.

2. The number of targets is arbitrary but known.

3. Each component of initial states of targets is random and approximated by a discrete random variable with "L_1" possible values. (Behind this assumption, lies the fact that an absolutely continuous random variable can be approximated by a discrete random variable. This discrete random variable will be a variable with "L_1" possible values where L_1 is a given finite positive integer. This approximation is explained in Appendix B in detail and discrete random variables approximating the Gaussian random variable with zero mean and unit variance is given in Table B.1.)

4. Each component of process noise v(k), is a discrete random variable with "L_2" possible values.

5. The false alarms and track misses are also considered.

6. The decision for observation to target pairings will be made after each multi-scan (N-scan) time intervals.

Data association is needed when there is a target and also false alarms or when there are more than one target. If there is only one target and no false alarms then data association problem does not exist. Solution to data association problem is discussed beginning from the next paragraph.

First of all, we think about a target. The number of possible initial states for this target will be given. Let this number be L_1. These L_1 possible states are calculated by using the discrete approximation of the continuous random variable. Then, using Equation (4.1) from the first assumption of algorithm and calculated initial states, possible states of the target at next scan are computed. The number of these possible states will be given as L_2. The state computation according to Equation (4.1) continues until we reach the maximum number of scan time required. The same computations are made for all targets. Then we obtain a branched tree

structure for each target as shown in Figure 4.1, which is a tree representation of possible paths of the target for multi-scan (N-scan) time periods. In Figure 4.1, each level correspond to a scan time and X_{ij} denotes a node of the tree such that X_{ij} is the j^{th} possible state location at level i.

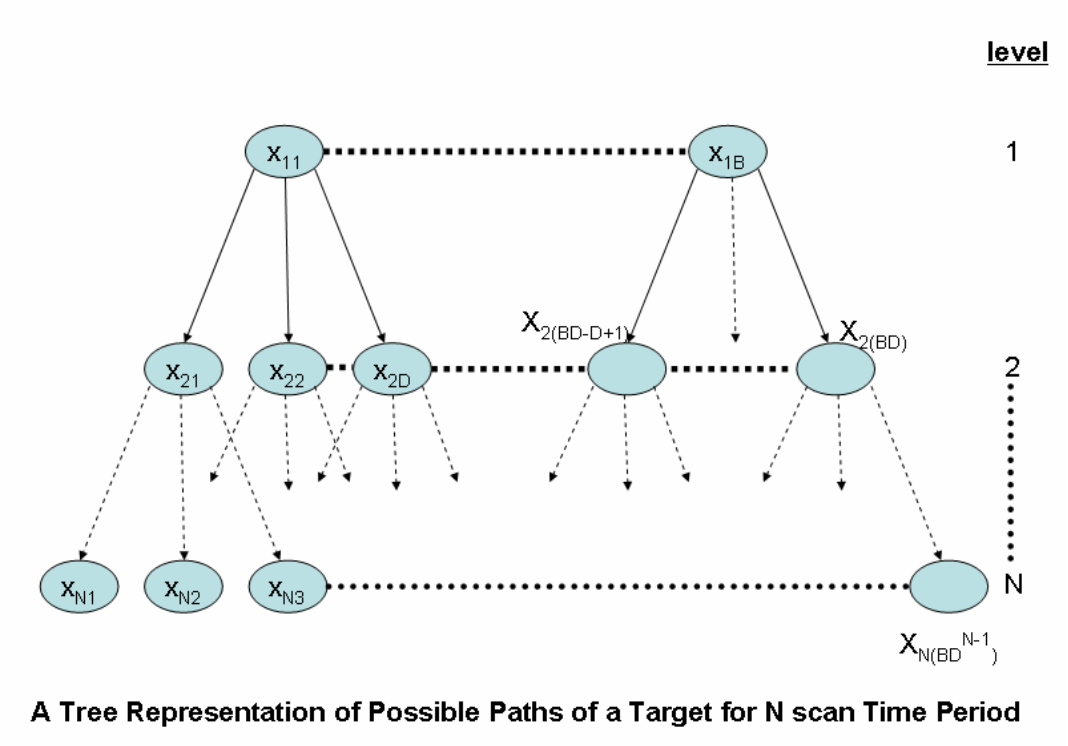


Figure 4-1 A Tree Representation of Possible Paths of a Target for N Scan Time Periods

By using the possible states for a target and the Equation (4.2) from the first assumption, the estimated observation values will be calculated. These values also form a tree structure. An example of tree representation of estimated observation locations is given in Figure 4.2, where:

Z_{ij}: the j^{th} estimated observation location vector at level i.

O_{im}: the m^{th} measured observation location vector at level i. (m= 1, 2,…,M where M is number of targets,)

Each level corresponds to a single scan time.

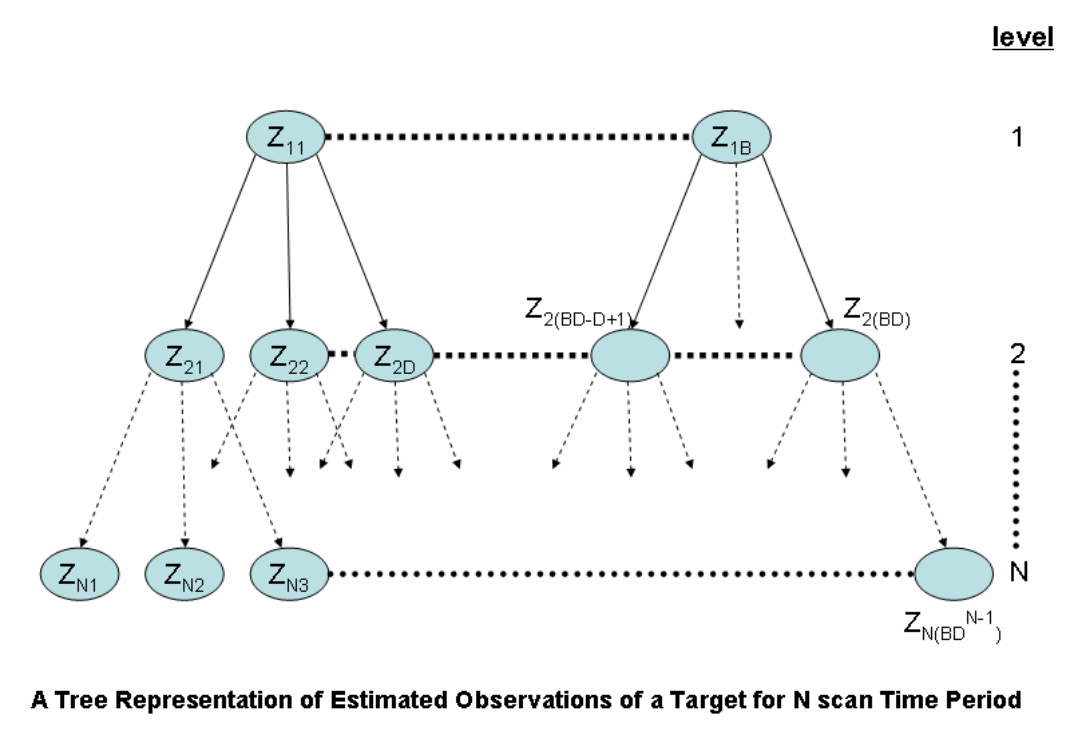


Figure 4-2 A Tree Representation of Estimated Observations of a Target for N Scan Time Periods

The estimated observation trees for all targets are calculated. It is assumed that the measurements coming from a sensor (radar) are available for first N scan time periods. Data association algorithm must solve the problem of matching the measurements to a target. To solve this problem, we compute the distances between the measured observations and the estimated observations. Distance calculation process is done in three steps. First, all permutations of the measured observations are calculated. Secondly, for each permutation the distances between the measurements that are assumed to come from a target and corresponding nodes

(estimated observation) on each branch of the estimated observation tree of that target are calculated. Same calculation is done for each target. Lastly, sum of the distance values for all targets are calculated. Distance calculation process is done for all permutations. The permutation, that has the minimum sum of distances, is chosen for association. The measurements in this permutation for the last N-scan time are assigned to the targets.

We will use following general distance equation for calculation of the distance between the measured observations and a node on the estimated observation tree for each dimension:

$$d^2_{imjd} = y^2_{imjd} / P_{ijd} \qquad (4.3)$$

where

d^2_{imjd} is a normalized squared distance value between m^{th} measured observation location O_{imd}, and j^{th} estimated observation location $Z_{ijd,}$ at level i in dimension d.

y_{imjd} is the residual between m^{th} measured observation O_{imd}, and j^{th} estimated observation Z_{ijd}, at level i in dimension d.

d stands for a component such as x, y, or z component of the target dynamic process.

In other words,

$y_{imjd} = O_{imd} - Z_{ijd}$ i = 1, 2, ……, N (levels)

m = 1, 2, ……, M

j = 1, 2, ……, ML

and

$P_{ijd} = \text{Prob}(\text{X}_{\text{ijd}})$ is the probability of the j^{th} predicted state location $X_{ijd,}$ at level i in dimension d.

The following total distance equation is used for calculation of the distance between the measured observations and a node on the estimated observation tree:

$$d_{imj}^2 = d_{imjx}^2 + d_{imjy}^2 + d_{imjz}^2 \qquad (4.4)$$

The P_{ijd} probabilities can be found in a tabular form in Table B.1. The inverses of these probabilities are used in Equation (4.3) because cost of accepting a lower valued residual distance between any measured observation location and an estimated observation location having a low probability, should be high as compared to an estimated observation location having a higher probability.

Our algorithm for the data association of multiple targets will be explained next section. Computer implementation of this algorithm is realized in MATLAB programming language. Simulation results of this implementation are given in Chapter 5.

4.2 Algorithm

1. The tree structure for possible states of targets with given initial states is constructed by using Equation (4.1).

2. The probabilities for possible state values for targets are calculated according to the table in Appendix B.

3. Estimated observation values for each target are calculated using motion tree and Equation (4.2).

4. Using random number generator the measured observation values are generated for each target according to the equations (4.1), (4.2) and given mean and variances.

5. All of the permutations of the radar measurements for N-scan are calculated.

6. For each permutation, distances between the measurements that are assumed to come from a target and corresponding nodes (estimated observation) on each branch of the estimated observation tree of that target are calculated.

7. Same calculation must be done for each target. If there are not any other targets then go to next step else go to step 6.

8. Sum of the distance values for all targets are calculated.

9. Distance calculation process must be done for all permutations. If there are not any other permutations left then go to next step, else go to step 6.

10. The permutation, that has the minimum sum of distances, is chosen for association.

11. Then for the next scan all the steps above are done for the N previous values of the current scan.

4.3 Computer Implementation of Data Association Algorithm Using MATLAB

In the implementation of data association algorithm, a recursive tree structure creation method is used to obtain required tree structure showing estimated measurement locations. To obtain the real measurements Gaussian random number generator of the MATLAB programming tool is used. Implementation can not be extended for large values of L_1, L_2 and N, maximum value of L_1, L_2 and N are assumed to be 4 because of memory limitations.

CHAPTER 5

SIMULATION RESULTS AND PERFORMANCE COMPARISON

5.1 Introduction

In this chapter, simulation results of the algorithm discussed in previous chapter are given. In these simulations effects of different parameters to the performance of the new data association algorithm are analyzed and the performance of the new algorithm is compared with JPDA method. In the explanation part of each simulation, number of the targets, number of discrete values approximating any component of v(k) and w(k), and the number, N, of levels (i.e. the number of scans to decide association) are given. F, H, matrices, mean and covariance of v(k) and w(k) and initial state vectors(start points) related with each target are given in section 5.2.

In all simulations targets evolve for 30 scan time periods. First data association is made after the N^{th} scan time where N is the chosen scan time period to decide association. A tree structure which is explained in section 4.1, starting with the initial state of target is constructed and data associations are made for first N scans. Also to make data association for the scans between N and 2N-2, a tree starting with initial state values and having the number of levels equal to number of scan time is constructed. Using this tree our algorithm makes data association for that scan time. For 2N-1 and greater scan times data association is made by using a tree structure starting from the node (state) having smallest distance to the associated measurement at the time N scan before. This tree has N levels and is used to make decision. Data association can be summarized as follows:

when $k < N$ ➔ data association is deferred until first N scans come.

when $k = N$ ➔ data association is made using an N-level tree structure whose initial nodes are the initial states of targets.

when $N < k \leq 2N - 2$ ➔ data association is made using an k-level tree structure whose initial nodes are the initial states of targets.

when $2N - 2 < k < \infty$ ➔ data association is made using an N-level tree structure starting from the node (state) having smallest distance to the associated measurements at scan k-2.

where k is the current scan and N is the number of levels to decide data association.

For the case that N is equal to 3, data association decision method is explained as follows. The first data association decision is made after the three scans. After the measurements of the third scan time come, a 3-level tree structure starting with the initial states of targets is constructed as explained in the multi-scan data association algorithm. Data association decisions for the first three scans are made using this tree. When the measurements of the fourth scan come, the data association decisions must be made using a 3-level tree structure starting with the nodes having minimum distances to the measurements for the second scan. But the nodes having minimum distances to the measurements for the second scan are not sufficiently trustworthy because 3-level tree structure was not used when the decision for the second scan made. Because of that, decisions for the fourth scan are made using a tree structure starting the initial states of targets and having four-levels. When the measurements of the fifth scan come, the data association decisions must be made using a 3-level tree structure starting from the node (state) having smallest distance to the associated measurements at third scan. Decisions for the third scan are made using 3-level tree structure so they are sufficiently trustworthy. A 3-level tree structure is constructed using our algorithm with initial points being the nodes having minimum distances between the associated measurements on the third scan. Using this tree, data association decisions for the

fifth scan are made. Data association decisions for the following scans are made as the same way for the fifth scan. In this method after the third scan, data association decisions for each scan are made for three times. When we are at scan k, data association decisions which are made immediately after k^{th} scan measurements come are accepted by our algorithm.

5.2 Implementation of JPDA Algorithm using MATLAB programming tool

JPDA algorithm is implemented as described in [6]. Starting point of the implementation is the calculation of predicted measurements for targets. In JPDA algorithm predicted states of targets are calculated by using Kalman filter. Using predicted states and the measurement prediction equation shown in Figure 2-2, predicted measurements are calculated. Kalman filter implementation is given starting with the next paragraph.

Firstly, it is assumed that the target dynamic process model is in the form:

$$x(k+1) = Fx(k) + v(k) \qquad (5.1)$$

where x is the target state vector, F is the state transition matrix, and v is white Gaussian process noise with a known covariance Q. The measurements are assumed to be given by:

$$y(k) = Hx(k) + w(k) \qquad (5.2)$$

where H is the measurement matrix and w is zero mean white Gaussian measurement noise with covariance R. The initial state prediction value for a target is the mean of the initial state of target. The initial value for the error covariance (P) is process noise covariance Q. Using initial conditions and the following equations predicted states can be calculated for all scans.

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + K(k)\left[y(k) - H\hat{x}(k \mid k-1)\right] \qquad (5.3)$$

$$K(k) = P(k \mid k-1)H'\left[HP(k \mid k-1)H'+R\right]^{-1} \quad (5.4)$$

$$P(k \mid k) = \left[I - K(k)H\right]P(k \mid k-1) \quad (5.5)$$

$$\hat{x}(k+1 \mid k) = F\hat{x}(k \mid k) \quad (5.6)$$

$$P(k+1 \mid k) = FP(k \mid k)F'+Q \quad (5.7)$$

where I is identity matrix and K(k) is Kalman gain matrix.

Predicted measurement values can also be calculated by using the following equation:

$$\hat{y}(k+1 \mid k) = H\hat{x}(k \mid k) \quad (5.8)$$

Real measurements, that are assumed to be coming from radar, are generated by using Gaussian random number generator function of MATLAB programming tool.

At this point, we have predicted measurements and the real measurements. The vector difference between the real and predicted measurements,

$$\tilde{y}(k) = y(k) - H\hat{x}(k \mid k-1) \quad (5.9)$$

is defined to be the residual (or innovations) vector with residual covariance matrix,

$$S = HP(k \mid k-1)H'+R \quad (5.10).$$

The normalized distance between the real and predicted measurements is defined as:

$$d_{ij}^{\ 2} = \tilde{y}_{ij}^{\ '} S_{ij}^{\ -1} \tilde{y}_{ij} \qquad (5.11)$$

where $\tilde{y}_{ij}$ is the residual and S_{ij} is the residual covariance vectors.

Next, we define validation regions for each target. Ellipsoidal gating is used to eliminate unlikely measurements. Normalized distances between real and predicted measurements are calculated. The real measurement is in the validation gate of the predicted measurement if the normalized distance satisfies the condition given in Equation 2.43. In this equation P_D and β values are taken 0.7 and 0.03 as given in [6], respectively. The value of M (dimension number) is taken to be 3.

Then, all probable hypotheses are formed. Hypotheses are the joint events which assign the valid measurements to the targets. The hypothesis formation for an example situation shown in Figure 3-2 is defined in section 3.4 of this thesis. Then, for all of these hypotheses, hypothesis likelihood values are calculated. The structure for computing the hypothesis probabilities is shown in Table 3-3. In order to calculate hypothesis likelihood, we need to compute Gaussian likelihood for all prediction-measurement pairings. Gaussian likelihood is the g_{ij} term. The calculation of Gaussian likelihood is given in Equation 3.16 where the terms $d_{ij}^{\ 2}$ and S_{ij} are the normalized distance and residual covariance matrix, respectively. $d_{ij}^{\ 2}$ can be calculated using Equation 5.11 and S_{ij} is the residual covariance matrix of the Kalman filter. The values for M, P_D, and β values are taken same as given above.

After calculation of hypothesis likelihood values for all hypotheses, the normalized probabilities are computed using Equation 3.17. In this Equation the $P^{'}(H_j)$ is the hypothesis likelihood value and N_H is the number of total hypotheses.

Lastly, to compute the probability p_{ij} that measurement j should be assigned to target i, a sum is taken over the probabilities from those hypotheses in which this assignment occurs. The measurement j which has the highest probability p_{ij}, is assigned to target i. There is an example (taken from reference [6]) in Section 3.4 showing the steps of the JPDA algorithm in detail.

5.3 Simulations

The parameters of targets that are used simulations are shown below.

Target 1:

$$F = \begin{bmatrix} 1.2 & 0.1 & 0.5 \\ 0.1 & 1.4 & 0.1 \\ 0.5 & 0.2 & 1.2 \end{bmatrix} \qquad H = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 1 & 3 & 2 \end{bmatrix}$$

$$\text{mean of v(k)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \text{covariance of v(k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of w(k)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{covariance of w(k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of initial state} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{covariance of initial state} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Target 2:

$$F = \begin{bmatrix} 1.2 & 0 & 0.5 \\ 0.1 & 1.15 & 0.1 \\ 0.5 & 0.5 & 1.2 \end{bmatrix} \qquad H = \begin{bmatrix} 2 & 4 & 5 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\text{mean of } v(k) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{covariance of } v(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of } w(k) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad \text{covariance of } w(k) = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2.25 \end{bmatrix}$$

$$\text{mean of initial state} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{covariance of initial state} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Target 3:

$$F = \begin{bmatrix} 1.4 & 0.1 & 0.8 \\ 0.1 & 1.4 & 0.1 \\ 0.5 & 0.2 & 1.2 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

$$\text{mean of } v(k) = \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix} \qquad \text{covariance of } v(k) = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

$$\text{mean of } w(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{covariance of } w(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of initial state} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \qquad \text{covariance of initial state} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

Target 4:

$$F = \begin{bmatrix} 1.15 & 0.1 & 0.4 \\ 0.1 & 1.25 & 0.15 \\ 0.5 & 0.3 & 1.2 \end{bmatrix} \qquad H = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 1 \\ 5 & 1 & 1 \end{bmatrix}$$

$$\text{mean of v(k)} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix} \qquad \text{covariance of v(k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of w(k)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{covariance of w(k)} = \begin{bmatrix} 6.25 & 0 & 0 \\ 0 & 2.25 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\text{mean of initial state} = \begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix} \qquad \text{covariance of initial state} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Target 5:

$$F = \begin{bmatrix} 1.15 & 0.1 & 0.4 \\ 0.1 & 1.3 & 0.15 \\ 0.5 & 0.3 & 1.1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

$$\text{mean of v(k)} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \qquad \text{covariance of v(k)} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

$$\text{mean of w(k)} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \text{covariance of w(k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{mean of initial state} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \qquad \text{covariance of initial state} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

5.3.1 Effect of the increase in number of targets

In these simulations, our aim is to show the effect of the increase in number of targets by considering targets which parameters are given above. Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

In simulations, when the number of targets is 2, then the parameters of the first 2 targets (Target 1, and Target 2) are used, when the number of targets is 3, then the parameters of the first 3 targets (Target 1, Target 2, and Target 3) are used and so on.

The distance between the mean of initial states for neighboring targets is 4 and 1000 in simulations whose results are shown in Table 5.1, and Table 5.2, respectively.

In a simulation targets evolve for 30 scan time periods. In each case, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made in one case. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in number of targets, probability of error, Pe, is computed as number of targets increase from 2 to 5.

Results of these simulations for the new algorithm and for JPDA method are tabulated in Table 5.1 and Table 5.2.

Table 5-1 Effect of the Increase in Number of Targets (when distance between mean of initial states of targets are 4)

Number of targets	**New Algorithm**		**JPDA Approach**	
	Error	**Pe**	**Error**	**Pe**
2	14	0,046	15	0,050
3	55	0,183	217	0,723
4	242	0,806	250	0,833
5	260	0,866	290	0,966

Table 5-2 Effect of the Increase in Number of Targets (when distance between mean of initial states of targets are 1000)

Number of targets	**New Algorithm**		**JPDA Approach**	
	Error	**Pe**	**Error**	**Pe**
2	0	0	0	0
3	0	0	24	0,080
4	0	0	27	0,090
5	210	0,700	240	0,800

Results: As can be seen from Tables 5-1 and 5-2, increasing number of targets degrades the performance of the new algorithm and JPDA method. We can also see that the probability of error for the new algorithm is always less than the

probability error for JPDA method. It is obvious that the increasing distance increases the performance of both algorithms.

5.3.2 Effect of the increase in number, L_2, of discrete values approximating any component of v(k) in Equation (4.1)

In these simulations, our aim is to show the effect of the increase in number, L_2, of discrete values approximating any component of v(k) in Equation (4.1) by considering targets which parameters are given previously. Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Distance between the mean of initial states for neighboring targets: 3.4.

The parameters of the first 2 targets (Target 1, and Target 2) and the first 3 targets (Target 1, Target 2, and Target 3) are used in simulations whose results are shown in Table 5.3, and Table 5.4, respectively.

In a simulation targets evolve for 30 scan time periods. In each case, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made in one case. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in number, L_2, of discrete values approximating any component of v(k) in Equation (4.1), probability of error, Pe, is computed as L_2 increases from 2 to 4.

Results of these simulations are tabulated in Table 5.3 and Table 5.4.

Table 5-3 Effect of the Increase in Number, L_2, of Discrete values Approximating v(k) (Simulation is made using Target 1, and 2.)

L_2	Error	Probability of Error
2	36	0.012
3	18	0.060
4	12	0.040

Table 5-4 Effect of the Increase in Number, L_2, of Discrete values Approximating v(k) (Simulation is made using Target 1, 2, and 3.)

L_2	Error	Probability of Error
2	71	0.236
3	58	0.193
4	40	0.133

Result: From Tables 5-3 and 5-4, we see that increasing discrete values for approximating the v(k) improves the performance of the new algorithm.

5.3.3 Effect of the increase in distance between mean of initial states of targets when there are no false alarms and measurement misses

In these simulations, our aim is to show the effect of the increase in distance between mean of initial states of targets by considering the targets whose parameters are given previously.

Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

These simulations are made for four different numbers of targets. The varying parameters for these simulations are given below.

Case 1:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 2 targets (Target 1, and 2).

Case 2:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 3 targets (Target 1, 2 and 3).

Case 3:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 4 targets (Target 1, 2, 3 and 4).

Case 4:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 5 targets (Target 1, 2, 3, 4 and 5).

In a simulation, targets evolve for 30 scan time periods. For each different parameter set, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made for one parameter set. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

Pe= E/300

To show the effect of the increase in distance between mean of initial states of neighboring targets probability of error, Pe, is computed for eighteen different choices of distances.

Results of these simulations for the new algorithm and for JPDA method are tabulated in Tables 5-5, 5-6, 5-7, and 5-8 and probability of error. Pe versus Ln distance is given in Figures 5-1, 5-2, 5-3, and 5-4.

Table 5-5 Effect of the Increase in Distance between Mean of initial States of 2 Targets

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	27	0,090	55	0,183
2.4	26	0,086	44	0,146
2.8	24	0,080	34	0,113
3.0	21	0,070	27	0,090
3.4	18	0,060	21	0,070
4.0	14	0,046	15	0,050
10	0	0	0	0
20	0	0	0	0
30	0	0	0	0
50	0	0	0	0
60	0	0	0	0
80	0	0	0	0
100	0	0	0	0
150	0	0	0	0
200	0	0	0	0
300	0	0	0	0
500	0	0	0	0
1000	0	0	0	0

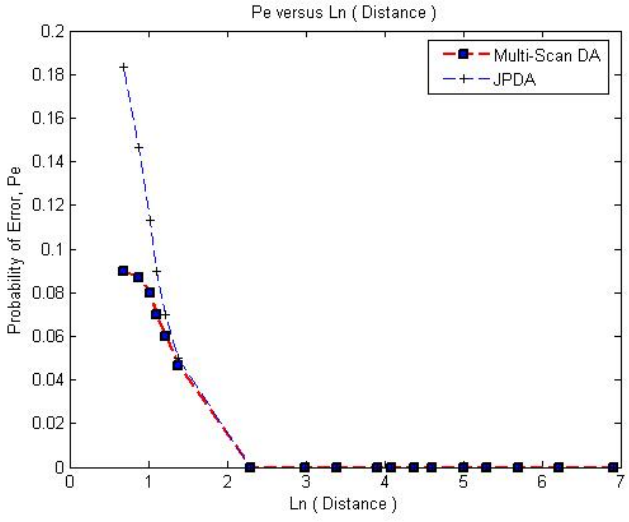


Figure 5-1 Pe versus Ln (Distance)

Table 5-6 Effect of the Increase in Distance between Mean of initial States of 3 Targets

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	102	0,340	253	0,843
2.4	73	0,243	249	0,830
2.8	64	0,213	229	0,763
3.0	58	0,193	222	0,740
3.4	56	0,186	218	0,726
4.0	55	0,183	217	0,723
10	52	0,173	216	0,720
20	33	0,110	215	0,716
30	29	0,096	210	0,700
50	27	0,090	202	0,673
60	25	0,083	170	0,566
80	24	0,080	161	0,536
100	19	0,063	127	0,423
150	13	0,043	110	0,366
200	0	0	94	0,313
300	0	0	65	0,216
500	0	0	24	0,080
1000	0	0	2	0,006

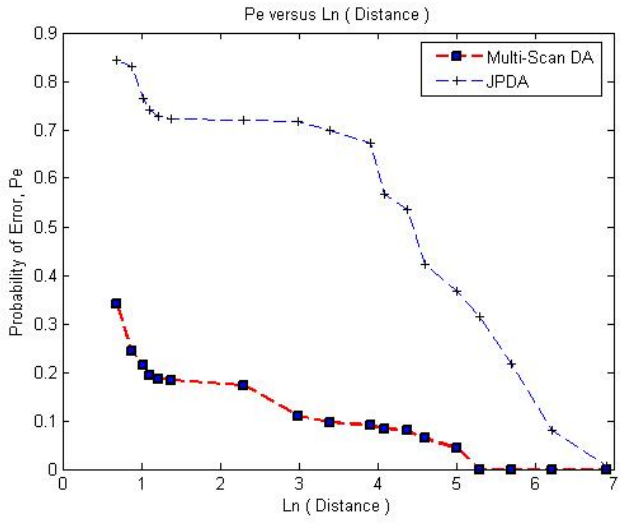


Figure 5-2 Pe versus Ln (Distance)

Table 5-7 Effect of the Increase in Distance between Mean of initial States of 4 Targets

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	290	0,966	300	1
2.4	280	0,933	300	1
2.8	270	0,900	290	0,966
3.0	265	0,883	290	0,966
3.4	260	0,866	290	0,966
4.0	242	0,806	250	0,833
10	88	0,293	241	0,803
20	55	0,183	234	0,780
30	42	0,140	221	0,736
50	35	0,116	196	0,653
60	28	0,093	189	0,630
80	20	0,066	168	0,560
100	17	0,056	143	0,476
150	7	0,023	115	0,383
200	1	0,003	62	0,206
300	0	0	59	0,196
500	0	0	27	0,090
1000	0	0	3	0,010

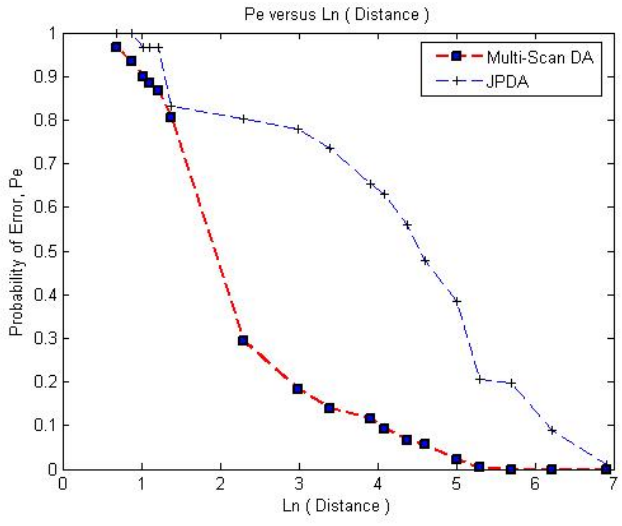


Figure 5-3 Pe versus Ln (Distance)

Table 5-8 Effect of the Increase in Distance between Mean of initial States of 5 Targets

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	300	1	300	1
2.4	300	1	300	1
2.8	290	0,966	300	1
3.0	280	0,933	290	0,966
3.4	270	0,900	290	0,966
4.0	260	0,866	290	0,966
10	257	0,856	280	0,933
20	248	0,826	280	0,933
30	247	0,823	277	0,923
50	238	0,793	275	0,916
60	237	0,790	275	0,916
80	232	0,773	275	0,916
100	231	0,770	273	0,910
150	220	0,733	272	0,906
200	220	0,733	271	0,903
300	218	0,726	246	0,820
500	210	0,700	240	0,800
1000	206	0,686	219	0,730

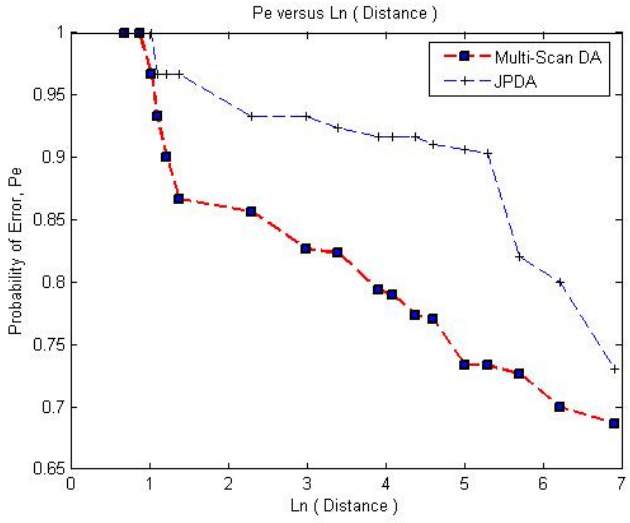


Figure 5-4 Pe versus Ln (Distance)

Results: From Figures 5-1, 5-2, 5-3, 5-4, we see that increase of the distance between targets increases the performance of the new algorithm. It can also be seen that increasing number of targets decreases the performance. In all simulations the new algorithm has better performance than JPDA method.

5.3.4 Effect of the increase in distance between mean of initial states of targets when there are false alarms

In these simulations, our aim is to show the effect of the increase in distance between mean of initial states of targets when there are false alarms by considering the targets whose parameters are given previously. The false alarms are created using the parameters other than the targets which are tracked. In all scan times there is one more measurement than the number of targets.

Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

These simulations are made for three different numbers of targets. The varying parameters for these simulations are given below.

Case 1:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 2 targets (Target 1, and 2) and 1 false alarm (generated using model of Target 3) for each scan.

Case 2:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 3 targets (Target 1, 2 and 3) and 1 false alarm (generated using model of Target 4) for each scan.

Case 3:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 4 targets (Target 1, 2, 3 and 4) and 1 false alarm (generated using model of Target 5) for each scan.

In a simulation, targets evolve for 30 scan time periods. For each different parameter set, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made for one parameter set. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in distance between mean of initial states of targets when there are false alarms, probability of error, Pe, is computed for eighteen different choices of distances.

Results of this example for the new algorithm and for JPDA method are tabulated in Tables 5-9, 5-10, and 5-11 and probability of error. Pe versus ln distance is given in Figures 5-5, 5-6, and 5-7.

Table 5-9 Effect of the Increase in Distance between Mean of initial States of 2 Targets (Target 1, and 2) and 1 false alarm

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	268	0,893	290	0,966
2.4	265	0,883	276	0,920
2.8	263	0,876	271	0,903
3.0	256	0,853	257	0,856
3.4	232	0,773	254	0,846
4.0	231	0,770	253	0,843
10	211	0,703	225	0,750
20	145	0,483	223	0,743
30	121	0,403	218	0,726
50	96	0,320	214	0,713
60	64	0,213	184	0,613
80	55	0,183	176	0,586
100	32	0,106	143	0,476
150	6	0,020	107	0,356
200	4	0,013	100	0,333
300	0	0	67	0,223
500	0	0	27	0,090
1000	0	0	3	0,010

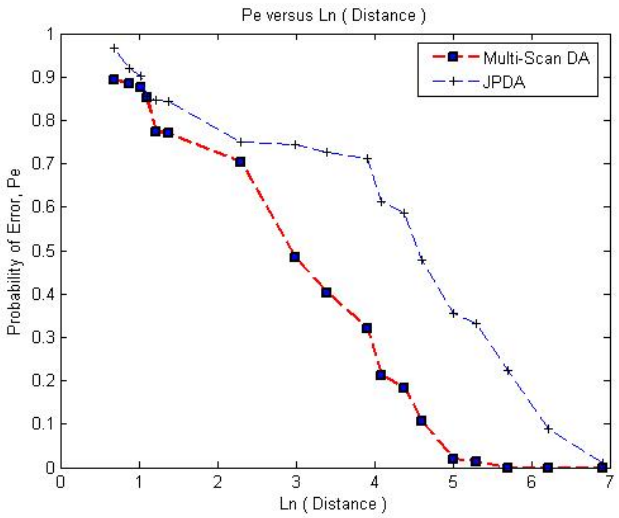


Figure 5-5 Pe versus Ln (Distance)

Table 5-10 Effect of the Increase in Distance between Mean of initial States of 3 Targets (Target 1, 2, and 3) and 1 false alarm

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	290	0,966	300	1
2.4	280	0,933	300	1
2.8	275	0,916	300	1
3.0	275	0,916	290	0,966
3.4	270	0,900	290	0,966
4.0	251	0,836	265	0,883
10	103	0,343	242	0,806
20	100	0,333	241	0,803
30	63	0,210	234	0,780
50	36	0,120	209	0,696
60	28	0,093	195	0,650
80	27	0,090	168	0,560
100	23	0,076	146	0,486
150	7	0,023	115	0,383
200	1	0,003	104	0,346
300	0	0	75	0,250
500	0	0	27	0,090
1000	0	0	4	0,013

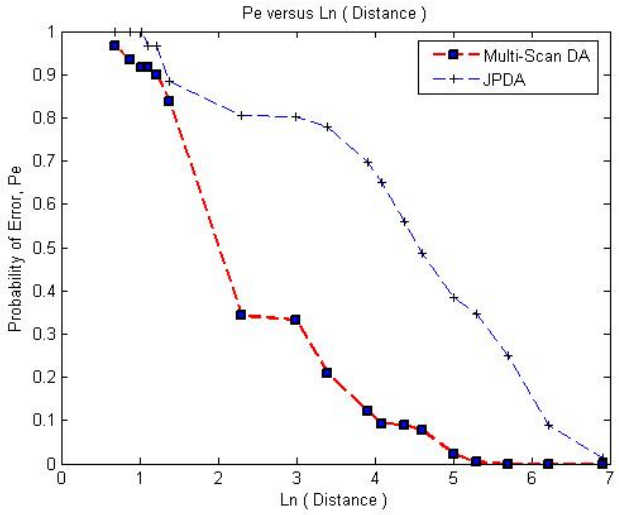


Figure 5-6 Pe versus Ln (Distance)

Table 5-11 Effect of the Increase in Distance between Mean of initial States of 4 Targets (Target 1, 2, 3 and 4) and 1 false alarm

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	300	1	300	1
2.4	300	1	300	1
2.8	300	1	300	1
3.0	290	0,966	300	1
3.4	280	0,933	290	0,966
4.0	280	0,933	290	0,966
10	270	0,900	290	0,966
20	265	0,883	288	0,960
30	260	0,866	280	0,933
50	258	0,860	280	0,933
60	247	0,823	280	0,933
80	236	0,786	279	0,930
100	235	0,783	274	0,913
150	234	0,780	273	0,910
200	234	0,780	268	0,893
300	231	0,770	262	0,873
500	220	0,733	242	0,806
1000	205	0,683	217	0,723

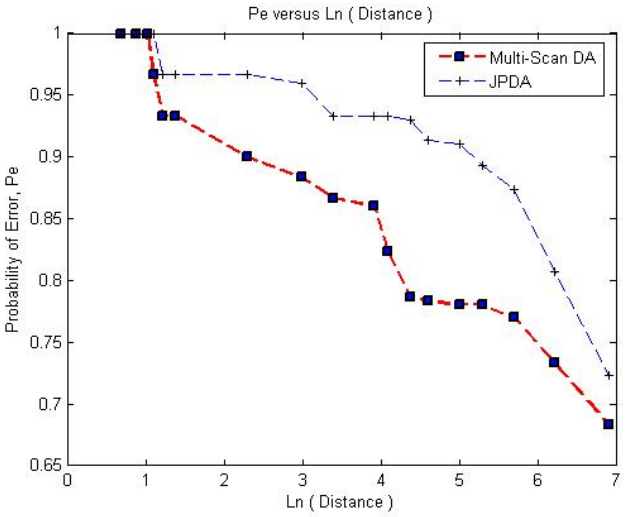


Figure 5-7 Pe versus Ln (Distance)

Results: By comparing Figures 5-2, 5-3, 5-4 with Figures 5-5, 5-6, and 5-7, we see that the false alarm affects the performance of both algorithms. Performance of both algorithms degreases when radar measurements contain false alarm. From Figures 5-5, 5-6, and 5-7, we see that increase of the distance between targets decreases the probability of error for both algorithms. It can also be seen that increasing number of targets decreases the performance. In all simulations the new algorithm has better performance than JPDA method.

5.3.5 Effect of the increase in distance between mean of initial states of targets when there are measurement misses

In these simulations, our aim is to show the effect of the increase in distance between mean of initial states of targets when there are measurement misses by considering the targets whose parameters are given previously. The measurement misses are obtained by deleting an observation from randomly chosen 5 scan times from 30 scan period of the randomly chosen target.

Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

These simulations are made for three different numbers of targets. The varying parameters for these simulations are given below.

Case 1:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 3 targets (Target 1, 2 and 3) and 1 measurement miss.

Case 2:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 4 targets (Target 1, 2, 3 and 4) and 1 measurement miss.

Case 3:

Distance between mean of initial states of neighboring targets is increasing from 2 to 1000 and there are 5 targets (Target 1, 2, 3, 4 and 5) and 1 measurement miss.

In a simulation, targets evolve for 30 scan time periods. For each different parameter set, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made for one parameter set. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in distance between mean of initial states of targets when there are measurement misses, probability of error, Pe, is computed for eighteen different choices of distances.

Results of these simulations for the new algorithm and for JPDA method are tabulated in Tables 5-12, 5-13, and 5-14 and probability of error. Pe versus Ln distance is given in Figures 5-8, 5-9 and 5-10.

Table 5-12 Effect of the Increase in Distance between Mean of initial States of 3 Targets (Target 1, 2 and 3) and when there is a measurement miss in 5 scan times from 30 scans period

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	118	0,393	261	0,870
2.4	91	0,303	252	0,840
2.8	89	0,296	242	0,806
3.0	88	0,293	228	0,760
3.4	86	0,286	220	0,733
4.0	79	0,263	219	0,730
10	73	0,243	219	0,730
20	60	0,200	219	0,730
30	55	0,183	215	0,716
50	51	0,170	197	0,656
60	50	0,166	183	0,610
80	49	0,163	169	0,563
100	38	0,126	147	0,490
150	34	0,113	123	0,410
200	26	0,086	101	0,336
300	25	0,083	74	0,246
500	19	0,063	46	0,153
1000	16	0,053	4	0,013

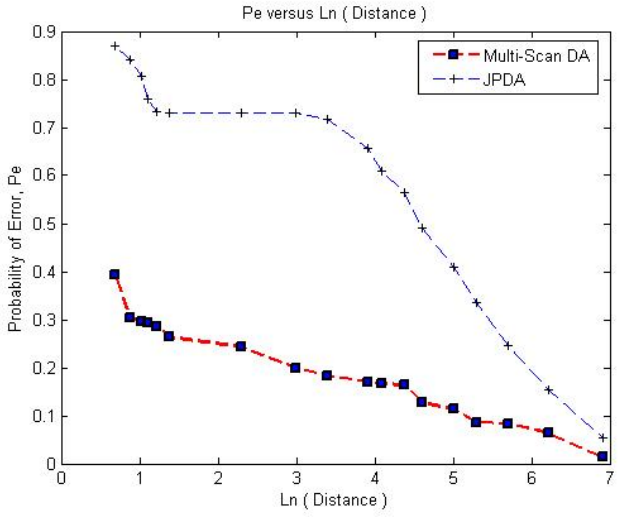


Figure 5-8 Pe versus Ln (Distance)

Table 5-13 Effect of the Increase in Distance between Mean of initial States of 4 Targets (Target 1, 2, 3 and 4) and when there is a measurement miss in 5 scan times of 30 scans period

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	300	1	300	1
2.4	290	0,966	300	1
2.8	280	0,933	300	1
3.0	270	0,900	290	0,966
3.4	270	0,900	290	0,966
4.0	267	0,890	283	0,943
10	100	0,333	248	0,826
20	89	0,296	239	0,796
30	76	0,253	228	0,760
50	76	0,253	222	0,740
60	53	0,176	190	0,633
80	45	0,150	170	0,566
100	40	0,133	150	0,500
150	36	0,120	144	0,480
200	31	0,103	102	0,340
300	30	0,100	60	0,200
500	27	0,090	53	0,176
1000	26	0,086	34	0,113

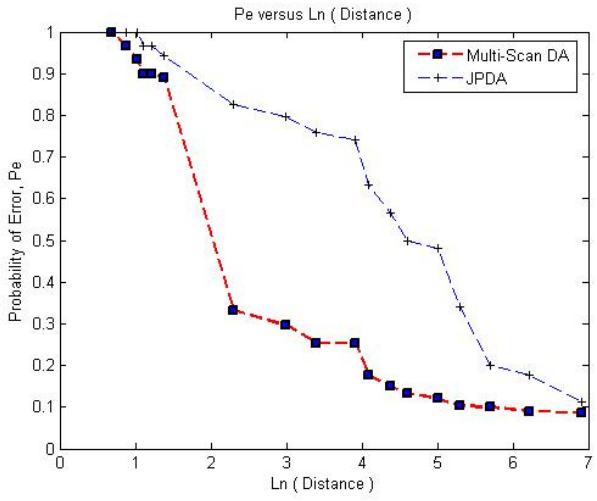


Figure 5-9 Pe versus Ln (Distance)

Table 5-14 Effect of the Increase in Distance between Mean of initial States of 5 Targets (Target 1, 2, 3, 4 and 5) and when there is a measurement miss in 5 scan times from 30 scans period

Distance	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
2	300	1	300	1
2.4	300	1	300	1
2.8	300	1	300	1
3.0	290	0,966	300	1
3.4	290	0,966	290	0,966
4.0	280	0,933	290	0,966
10	270	0,900	290	0,966
20	260	0,866	280	0,933
30	248	0,826	280	0,933
50	245	0,816	277	0,923
60	242	0,806	276	0,920
80	242	0,806	275	0,916
100	241	0,803	275	0,916
150	238	0,793	274	0,913
200	234	0,780	273	0,910
300	230	0,766	268	0,893
500	227	0,756	242	0,806
1000	222	0,740	228	0,760

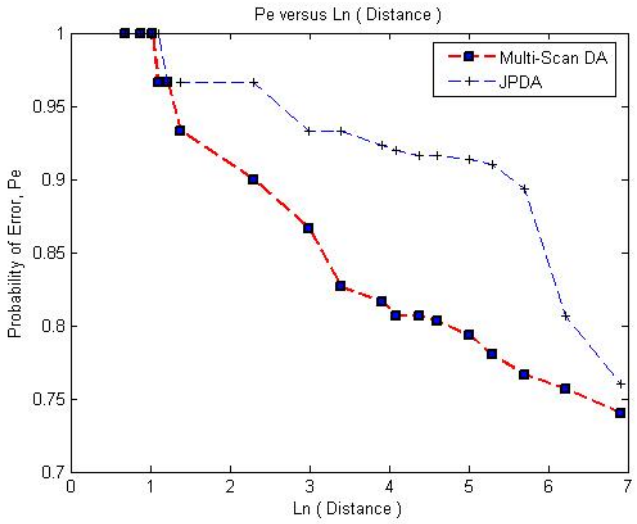


Figure 5-10 Pe versus Ln (Distance)

Results: By comparing Figures 5-2, 5-3, 5-4 with Figures 5-8, 5-9, and 5-10, we see that the measurement misses degreases the performance of both algorithms. From Figures 5-8, 5-9, and 5-10, we see that increase of the distance between targets decreases the probability of error for both algorithms. It can also be seen that increasing number of targets decreases the performance. In all simulations the new algorithm has better performance than JPDA method.

5.3.6 Effect of the increase in variance of each component of v(k) in Equation (4.1)

In these simulations, our aim is to show the effect of the increase in variance of each component of v(k) in Equation (4.1) by considering first 3 targets (Target 1, Target 2, Target 3) which parameters are given previously.

Parameters which are fixed during all cases of these simulations are as follows:

Number of scans needed to make association: 3,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

These simulations are made for six different situations. The varying parameters for these simulations are given below.

Case 1:

Variance of the x component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of y and z components are 1. The distance between mean of initial states of neighboring targets is 50 and there are 3 targets (Target 1, 2 and 3).

Case 2:

Variance of the x component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of y and z components are 1. The distance between mean of

initial states of neighboring targets is 150 and there are 3 targets (Target 1, 2 and 3).

Case 3:

Variance of the y component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of x and z components are 1. The distance between mean of initial states of neighboring targets is 50 and there are 3 targets (Target 1, 2 and 3).

Case 4:

Variance of the y component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of x and z components are 1. The distance between mean of initial states of neighboring targets is 150 and there are 3 targets (Target 1, 2 and 3).

Case 5:

Variance of the z component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of x and y components are 1. The distance between mean of initial states of neighboring targets is 50 and there are 3 targets (Target 1, 2 and 3).

Case 6:

Variance of the z component of v(k) is increasing from 1 to $5x10^{12}$ while the variances of x and y components are 1. The distance between mean of initial states of neighboring targets is 150 and there are 3 targets (Target 1, 2 and 3).

In a simulation, targets evolve for 30 scan time periods. For each different parameter set, simulations are made for 10 times in order to get an average value.

Totally 300 data association decisions are made for one parameter set. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in variance of each component of v(k) in Equation (4.1), probability of error, Pe, is computed for fourteen different choices of distances.

Results of these simulations for the new algorithm and for JPDA Method are tabulated in Table 5-15, 5-16, 5-17, 5-18, 5-19, 5-20 and probability of error Pe versus Ln variance is given in Figure 5-11, 5-12, 5-13, 5-14, 5-15, and 5-16.

Table 5-15 Effect of the Increase in Variance of x component of v(k) in Equation (4.1) (when variances of y and z components are 1 and distance between mean of initial states of targets are 50)

Variance of x	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	24	0.080	39	0.130
2	24	0.080	40	0.133
4	25	0.083	40	0.133
10	32	0.106	42	0.140
20	33	0.110	46	0.153
50	42	0.140	50	0.166
100	52	0.173	69	0.23
250	57	0.190	73	0.243
400	62	0.206	77	0.256
500	66	0.220	91	0.303
1000	84	0.280	99	0.330
$5x10^6$	92	0.306	108	0.360
$5x10^9$	100	0.333	109	0.363
$5x10^{12}$	129	0.43	129	0.43

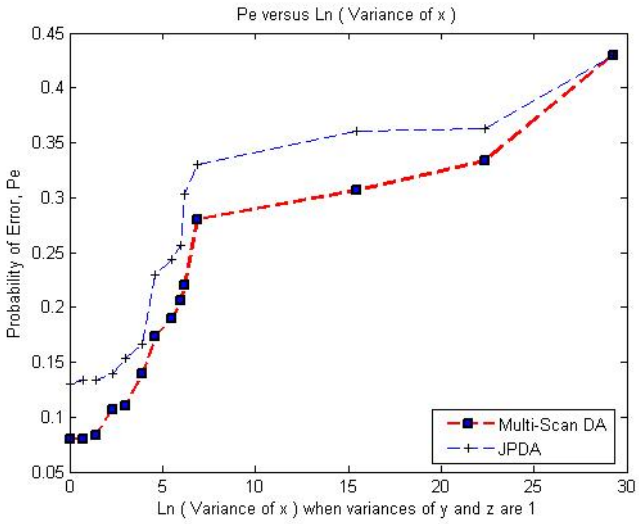


Figure 5-11 Pe versus Ln (Variance of x)

Table 5-16 Effect of the Increase in Variance of x component of v(k) in Equation (4.1) (when variances of y and z components are 1 and distance between mean of initial states of targets are 150)

Variance of x	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	12	0.040	28	0.093
2	12	0.040	39	0.130
4	14	0.046	44	0.146
10	15	0.050	50	0.166
20	17	0.056	53	0.176
50	18	0.060	60	0.200
100	28	0.093	61	0.203
250	50	0.166	61	0.203
400	56	0.186	62	0.206
500	58	0.193	62	0.206
1000	58	0.193	67	0.223
$5x10^6$	73	0.243	75	0.250
$5x10^9$	82	0.273	83	0.276
$5x10^{12}$	84	0.280	84	0.280

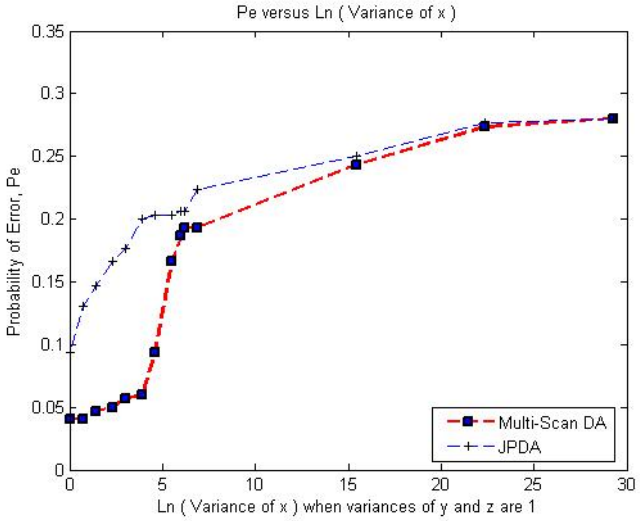


Figure 5-12 Pe versus Ln (Variance of x)

Table 5-17 Effect of the Increase in Variance of y component of v(k) in Equation (4.1) (when variances of x and z components are 1 and distance between mean of initial states of targets are 50)

Variance of y	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	16	0.053	49	0.163
2	19	0.063	50	0.166
4	23	0.076	68	0.226
10	28	0.093	69	0.230
20	32	0.106	77	0.256
50	34	0.113	85	0.283
100	34	0.113	101	0.336
250	39	0.130	103	0.343
400	43	0.143	103	0.343
500	44	0.146	106	0.353
1000	52	0.173	109	0.363
$5x10^6$	103	0.343	112	0.373
$5x10^9$	105	0.35	122	0.406
$5x10^{12}$	123	0.41	126	0.420

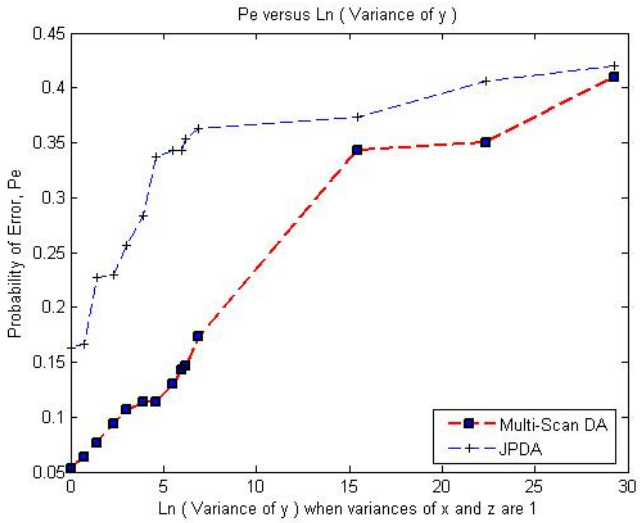


Figure 5-13 Pe versus Ln (Variance of y)

Table 5-18 Effect of the Increase in Variance of y component of v(k) in Equation (4.1) (when variances of x and z components are 1 and distance between mean of initial states of targets are 150)

Variance of y	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	4	0.013	34	0.113
2	6	0.020	37	0.123
4	6	0.020	40	0.133
10	13	0.043	45	0.150
20	14	0.046	46	0.153
50	18	0.060	51	0.170
100	21	0.070	57	0.190
250	29	0.096	59	0.196
400	32	0.106	60	0.200
500	37	0.123	63	0.210
1000	40	0.133	69	0.230
$5x10^6$	45	0.150	74	0.246
$5x10^9$	69	0.230	79	0.263
$5x10^{12}$	88	0.293	90	0.300

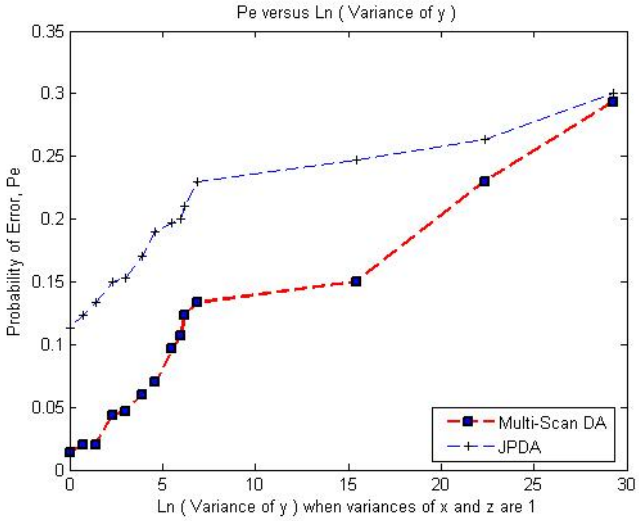


Figure 5-14 Pe versus Ln (Variance of y)

Table 5-19 Effect of the Increase in Variance of z component of v(k) in Equation (4.1) (when variances of x and y components are 1 and distance between mean of initial states of targets are 50)

Variance of z	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	25	0.083	33	0.110
2	26	0.086	33	0.110
4	30	0.100	55	0.183
10	30	0.100	63	0.210
20	32	0.106	79	0.263
50	37	0.123	81	0.270
100	48	0.160	82	0.273
250	59	0.196	87	0.290
400	63	0.210	88	0.293
500	64	0.213	88	0.293
1000	69	0.230	96	0.320
$5x10^6$	70	0.233	110	0.366
$5x10^9$	102	0.340	121	0.403
$5x10^{12}$	118	0.393	122	0.406

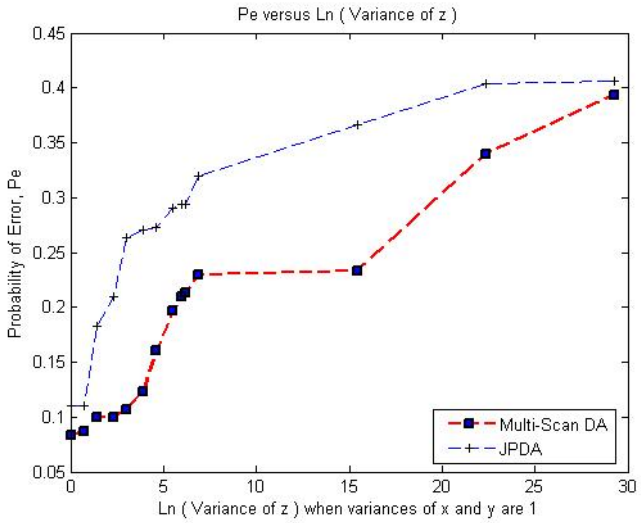


Figure 5-15 Pe versus Ln (Variance of z)

Table 5-20 Effect of the Increase in Variance of z component of v(k) in Equation (4.1) (when variances of x and y components are 1 and distance between mean of initial states of targets are 150)

Variance of z	New Algorithm		JPDA Approach	
	Error	Pe	Error	Pe
1	3	0.010	25	0.083
2	4	0.013	29	0.096
4	10	0.033	30	0.100
10	19	0.063	33	0.110
20	22	0.073	48	0.160
50	24	0.080	50	0.166
100	31	0.103	53	0.176
250	35	0.116	66	0.220
400	45	0.150	68	0.226
500	64	0.213	69	0.230
1000	73	0.243	73	0.243
$5x10^6$	78	0.260	79	0.263
$5x10^9$	83	0.276	85	0.283
$5x10^{12}$	85	0.283	88	0.293

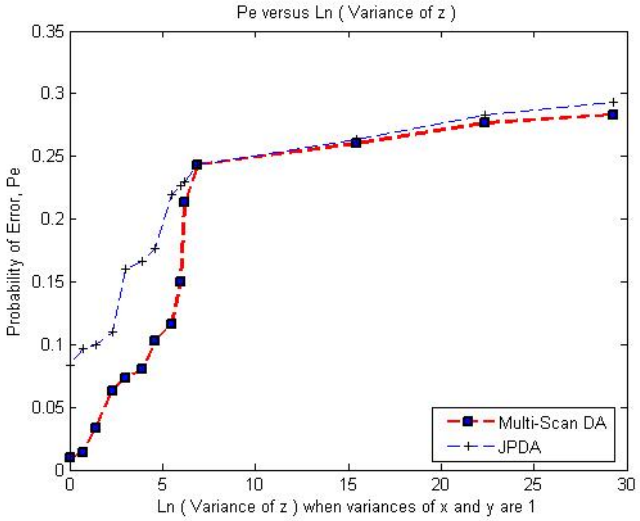


Figure 5-16 Pe versus Ln (Variance of z)

Results: From Figures 5-11, 5-12, 5-13, 5-14, 5-15, and 5-16, we see that increasing variance decreases the performance of the new algorithm and JPDA method. But it is easily seen that in all simulations the new algorithm has better performance than JPDA method.

5.3.7 Effect of the increase in number of scans to decide association

In these simulations, our aim is to show the effect of the increase in number of scans to decide association by considering targets which parameters are given above. Parameters which are fixed during all cases of these simulations are as follows:

Number of targets : 3,

Distance between mean of initial states of neighboring targets : 150,

Number of discrete values approximating the components of v(k) in Eq. (4.1) : 3.

In simulations, the parameters of the first 3 targets (Target 1, Target 2, and Target 3) are used.

In a simulation targets evolve for 30 scan time periods. In each case, simulations are made for 10 times in order to get an average value. Totally 300 data association decisions are made in one case. The numbers of incorrect data associations, E, are determined and the probability of error, Pe, is computed. The following formula is used to compute Pe.

$$Pe = E/300$$

To show the effect of the increase in number of scans to decide association, probability of error, Pe, is computed as number of scans increase from 2 to 4.

Results of these simulations for the new algorithm and for JPDA method are tabulated in Table 5.1 and Table 5.2.

Table 5-21 Effect of the Increase in Number of Targets (Simulation is made using parameters of Target 1, 2, and 3. Distance between mean of initial states for neighboring targets is 150)

Number of scans	**New Algorithm**		**JPDA Approach**	
	Error	**Pe**	**Error**	**Pe**
2	100	0.333	127	0.423
3	17	0.056		
4	7	0.023		

Results: As can be seen from Table 5-21, increasing number of scans to decide association increases the performance of the new algorithm.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we have presented a multi-scan data association algorithm for multitarget tracking. The main difference of this algorithm from classical methods is the usage of discrete approximation of continuous random variable technique for the process noise. The predicted states for targets are found by using these approximated values. The classical data association algorithms generally use Kalman filter to predict the state of targets.

Our aim in this thesis is to develop a different approach for data association problems in multitarget tracking. We calculate all of the probable trajectories for each target according to approximation method. Probable trajectories of targets are formed with predicted measurement values. After that, the distances between real measurement values and predicted measurement locations are calculated. The elements of real measurement set which gives the minimum sum of distance, are assigned to relevant targets.

This algorithm can be named as multi-scan, nearest neighbor, recursive algorithm. Multi-scan is chosen because decision of the data association is made after a given number of scan times. We can use nearest neighbor because minimum sum of distance values are used to associate the real measurements with targets. It is recursive because recursive distance calculation is used to calculate the distance values.

Two different approaches are examined during thesis work. One of them is our multi-scan data association algorithm the other is classical JPDA algorithm. The

simulations are performed using scenarios that represent several conditions to compare these algorithms.

Multi-scan data association algorithm can be viewed as a subsystem of a general multitarget tracking (MTT) system.

Simulation results show that performance of the new algorithm is better than JPDA algorithm. For example, from Tables 5-1 and 5-2 it can be easily seen that the performance of new algorithm is better when number of targets decreases. In addition, in figures 5-1, 5-2, 5-3, and 5-4 we can conclude that for any initial distance value between targets, the new algorithm is desirable. From figures 5-5, 5-6 and 5-7, we can see that the addition of false alarm degrades the performance of the new algorithm. Also, the increase in probability of error can be seen from figures 5-8, 5-9, and 5-10 when there are observation misses. Moreover, effect of the increase in variance v(k) in Equation (4.1) is presented in Figures 5-11, 5-12, 5-13, 5-14, 5-15, and 5-16 and it is seen that, for any variance value, the new algorithm's performance is higher than JPDA algorithm. Lastly, from Table 5-21 one can easily see that increase in number of scans to decide data association increases the performance of the new algorithm.

As a result, the simulations show that computer implementation of this algorithm, cover the requirements for multiple target tracking.

As a future work, possibility for reduction of the computation complexity of the proposed multi-scan data association algorithm should be examined. Also, a multi-scan version of the JPDA would also be interesting research topic that would lead the performances of the JPDA algorithm closer to the multi-scan data association algorithm.

REFERENCES

[1] Pattipati, K.R., Deb, S., Bar-Shalom, Y., Washburn, R.B.Jr., "A New Relaxation Algorithm and Passive Sensor Data Association", IEEE Transactions on Automatic Control, Vol. 37, No 2, (Feb. 1992), pp. 198-213

[2] Bar-Shalom, Y., and Li, X.R., Estimation and Tracking: Principles, Techniques, and Software. Boston, MA, Artech House, 1993.

[3] Bar-Shalom, Y., and Fortmann, T.E., "Tracking and Data Association", Academic Press, 1988

[4] Bar-Shalom, Y. (Ed.), "Multitarget-Multisensor Tracking: Advanced Applications, Vol.1.", Artech House, 1990

[5] Blackman, S.S., "Multiple Target Tracking with Radar Applications", Artech House, 1986

[6] Blackman, S.S., and Popoli, R. "Design and Analysis of Modern Tracking Systems", Artech House, 1999

[7] Li, X., Luo, Z.Q., Wong, K.M., Bosse, E., "An Interior Point Linear Programming Approach to Two-Scan Data Association", IEEE Trans. On Aerospace and Electronic Systems, Vol. 35, April 1999, 474-490.

[8] Bar-Shalom, Y., and Li, Xiao-Rong, Multitarget-Multisensor Tracking: Principles and Techniques, New York, YBS, 1995.

[9] Leung H., Zhijian Hu, and M. Blanchette, Evaluation of multiple radar target trackers in stressful environments, IEEE Trans. On Aerospace and Electronic Systems, vol. 35, April 1999, 663-674.

[10] Tekbaş, Ö. H. , "An N-Scan Back Algorithm for Data Association in Multitarget Tracking" METU, M.S. Thesis, 1996

[11] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software, John Wiley & Sons, 2001.

[12] Uhlmann, J.K., et al., "Efficient Approaches for Report/Cluster Correlation in Multitarget Tracking Systems," NRL Report 9281, Naval Research Laboratory, Washington, D.C. 20375-5000, Dec. 31, 1990

[13] Uhlmann, J.K., "Algorithms for Multiple-Target Tracking," American Scientists, Vol. 80, Mar-Apr.1992, pp. 128-141

[14] Collins, J.B., and J.K. Uhlmann, "Efficient Gating in Data Association with Multivariate Gaussian Distributed States," IEEE Trans. On Aerospace and electronic Systems, Vol. AES-28, No.3, July 1992, pp. 909-916

[15] Sittler, R. W., "An Optimal Data Association Problem in Surveillance Theory," IEEE Trans. on Military Electronics, Vol. MIL-8, Apr. 1964. pp. 125-139.

[16] Gish, H., and R. Mucci, "Target State Estimation in a Multi-Target Environment," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-23. Jan. 1987, pp. 60-73.

[17] Bethel. R. E., and G. J. Paras, "A PDF Multi-Target Tracker: IEEE Trans. On Aerospace and Electronic Systems, Vol. AES-30, No.2, Apr. 1994, pp. 386-403.

[18] Barlow. C. A., et al., "Unified Data Fusion." Proc. 9th National Symp. Sensor Fusion, Monterey, CA, Mar. 1996, pp. 321-330.

[19] Bar-Shalom, Y., and Tse, E. “Tracking in cluttered environment with probabilistic data association.” Automatica, 11 (Sept. 1975), 451-460

[20] Reid, D. B., “An algorithm for tracking multiple targets.”, IEEE Transactions on Automatic Control, AC-24 (Dec. 1979), 843-854

[21] Cox. I. J., and S. L. Hingorani. "An Efficient Implementation of Reid's Multiple Hypotheses Tracking Algorithm and Its Evaluation for the Purposes of Visual Tracking," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-18, No.2, Feb. 1996, pp. 138-150.

[22] Cox, J. J., and M. L. Miller. "On Finding Ranked Assignments with Application to Multitarget Tracking and Motion Correspondence,' IEEE Trans. on Aerospace and Electronic Systems. Vol. AES-31. No. I. Jan. 1995, pp. 486-489.

[23] Kurien. T., "Issues in the Design of Practical Multitarget Tracking Algorithms," Multitarget-Multisensor Tracking: Advanced Applications. Y. Bar-Shalom (Ed.). Norwood, MA: Artech House. 1990. Chap. 3.

[24] Demos, G. C., et al., “Applications of MHT to Dim Moving Targets,”Signal and Data Processing of Small Targets, Proc. SPIE. Vol. 1305, Apr. 1990, pp. 297-309.

[25] Chan. D. S. K., et al., “Tracking in a High-Clutter Environment: Simulation Results Characterizing a BiLevel MHT Algorithm,” Signal and Data Processing of Small Targets 1993, Proc. SPIE, Vol. 1954, 1993, pp. 540-551.

[26] Chan. D. S. K., and D. A. Langan, “Performance Results of the Bilevel MHT Tracking Algorithm for Two Crossing Targets in a High Clutter

Environment," Signal and Data Processing of Small Targets 1994. Proc. SPIE. Vol. 2235. Apr. 1994. pp. 406-416.

[27] Singer, R.A. and Sea, R.G., "Derivation and Evaluation of Improved Tracking Filters for use in Dense Multitarget Environments," IEEE Transaction on Information Theory, Vol. IT-20, pp. 423-432, July 1974

[28] Demirbaş, K., "Information Theoric Smoothing Algorithms for Dynamic Systems with or without Interference", Advances in Control and Dynamic Systems,Vol XXI, Academic Press, pp. 175-295, 1984.

[29] Apostal, A.M., "Mathematical Analysis" Addison Wesley, Reading,Massachussets, 1958

APPENDIX A

ESTIMATION OF GAUSSIAN RANDOM VARIABLE, X, AS AN EXAMPLE FOR LINEAR ESTIMATION IN STATIC SYSTEMS, [2]

Let x be the random vector to be estimated and z the measurement or the observation.

The estimate of the random vector x in terms of z according to the minimum mean square error (MMSE) criterion† is the conditional mean of x given z. (Proof is available at page 98 of [2])

For x and z jointly Gaussian, the conditional mean is:

$$\hat{x} = E[x \mid z] = \bar{x} + P_{xz} P_{ZZ} (z - \bar{z}) \qquad \text{(A.1)}$$

and the corresponding conditional covariance matrix is:

$$P_{XX|Z} = E[(x - \hat{x})(x - \hat{x})' \mid z] = P_{XX} - P_{XZ} P_{ZZ}^{-1} P_{ZX} \qquad \text{(A.2)}$$

where

$$\bar{x} = E[x]$$

---

† Actually MMSE is a particular case of static estimation problems. It is a Bayesian approach. In the Bayesian approach, the aim is to minimize a cost function C(x-x). The MMSE cost function is a quadratic. The widespreed use of quadratic criterion is primarily due to (relative) ease of obtainins the solution.

$$\bar{z} = E[z]$$

$$P_{XX} = \text{cov}(x) = E\left[(x - \bar{x})(x - \bar{x})'\right]$$

$$P_{ZZ} = \text{cov}(z) = E\left[(z - \bar{z})(z - \bar{z})'\right]$$

$$P_{XZ} = \text{cov}(x, z) = E\left[(x - \bar{x})(z - \bar{z})'\right] = P_{ZX}$$

Proofs of equation (A.1) and equation (A.2) are available at page 43-44 of [27]

Remarks: The MMSE estimate -the conditional mean- of a Gaussian random vector in terms of another Gaussian random vector (the measurement) is a linear combination of

- The prior (unconditional) mean of the variable to be estimated.
- The difference between the measurement and its prior mean.

The conditional covariance of one Gaussian random vector given another Gaussian random vector (the measurement) is independent of the measurement. Both of the above properties hinge strictly on the assumption that the two random vectors under consideration are jointly Gaussian.

APPENDIX B

APPROXIMATION OF AN ABSOLUTELY CONTINUOUS RANDOM VECTOR BY A DISCRETE RANDOM VECTOR [28]

Let "L" be a given positive integer and D^m be the set of distribution functions of all mx1 discrete random vectors with "L" possible values, where superscript m stands for the dimensionality of random vectors. Then the problem of approximating an absolutely continuous mx1 random vector X^m with distribution function $F_X m(.)$ by an mx1 discrete random vector with "L" possible values is to find a distribution function $F_{y0^m}(.) \in D^m$, which minimizes the objective function J(.) over the set D^m :

$$J(F_{y0^m}(.)) = \min_{F_{y^m}(.) \in D^m} J(F_{y^m}(.)) \qquad \text{(B.1)}$$

where

$$J(F_{y^m}(.)) \stackrel{\Delta}{=} \int_{R^m} \left[F_{x^m}(a) - F_{y^m}(a)\right]^2 da, \qquad F_{y^m}(.) \in D^m \qquad \text{(B.2)}$$

note that the integration is performed over the m dimensional Euclidean space R^m. The discrete random vector defined by $F_{y_0{}^m}(.)$ is referred to as the optimum discrete random vector approximating the random vector X^m. Here, the approximation of an absolutely continuous random variable X with distribution necessary conditions that the optimum discrete random variable approximating X must satisfy are obtained. Finally, discrete random variables are obtained.

Let us now state two theorems and defined some symbols which are used. The proofs of the theorems are given in [29]

Theorem B.1 [29]

Let $f(y) \overset{\Delta}{=} f(y_1, y_2, ..., y_l)$ be a real-valued function on an open set Γ of R^l, and let f(y) have finite partial derivatives $\partial f(y) / \partial y_k$, $k = 1,2,...,l$ at each point of Γ. If f(y) has a local minimum at the point $y_0 \overset{\Delta}{=} (y_{1,0}, y_{2,0}, ..., y_{l,0},)$ in Γ, then $\partial f(y) / \partial y_k \Big|_{y=y_0} = 0$ for each, $k = 1,2,...,l$.

Theorem B.2 [29]

Let $f(y) \overset{\Delta}{=} f(y_1, y_2, ..., y_l)$ be a rea1-valed function on an open set Γ of R^l, and let f(y) have continuous second order partial on Γ. Let $y_0 \overset{\Delta}{=} (y_{1,0}, y_{2,0}, ..., y_{l,0},)$ be a point of Γ for which $\partial f(y) / \partial y_k \Big|_{y=y_0} = 0$ for each, $k = 1,2,...,l$. Assume that the determinant $G \overset{\Delta}{=} \det\left\{\left[\nabla^2 f(y)\right]_{y=y_0}\right\} \neq 0$, where

$$\left[\nabla^2 f(y)\right]_{ij} \overset{\Delta}{=} \frac{\partial^2}{\partial y_i \partial y_j} f(y)$$

Let G_{l-k} be the determinant obtained from G by deleting the last k rows and columns. If the l numbers $G_1, G_2, ..., G_l$ are all positive, then f(y) has a local minimum at y_0.

We now define D as the set of distribution functions of all discrete random variables with "L" possible values, where "L" is a given positive integer. We next

define S as the set of all stepfunctions with L steps. A stepfunction g(.) with L steps is, by definition, a function with "L+l" possible values in the real line R such that g(.) is zero at $-\infty$ and one at $+\infty$; the set of numbers which are mapped by g(.) to a chosen possible value is an interval in R. the intervals corresponding to the L+ 1 possible values are nonoverlapping, and the union of these intervals is the real line, that is,

$$S \overset{\Delta}{=} \{g(x): \quad g(x) = 0 \quad for \quad x < y_i;$$

$$g(x) = P_i \quad for \quad y_i \le x \le y_{i+1}, y_i \in (-\infty,+\infty);$$

$$g(x) = 1 \quad for \quad x \ge y_L; y_{i+1} \le y_i, y_i \in (-\infty,+\infty);$$

$$i = 1,2,...,L-1$$

In order to find the optimum discrete random variable with L possible values that approximates an absolutely continuous random variable X with distribution function $F_x(.)$, we must find a distribution function $F_{y0}(.)$ which minimizes the objective function J(.) over the set D:

$$J(F_{y_0}(.)) = \min_{F_y(.) \in D} J(F_y(.)) \qquad \text{(B.3)}$$

$$= \min_{g(.) \in S} J(g(.)) \qquad \text{(B.4)}$$

where ,

$$J(F_y()) \overset{\Delta}{=} \int_{-\infty}^{+\infty} \left[F_x(a) - F_y(a)\right]^2 da \qquad \text{(B.5)}$$

The equality in equation (B.4) follows from the following arguments. Let a stepfunction $g_0(\,) \in$ S minimize J(.) over the set S, since the distribution function $F_x(.)$ is nondecreasing, $g_0(.)$ must be nondecreasing; hence it is a nondecreasing step function whose range changes from zero to one; therefore $g_0(\,) \in$ D. Thus the aim is to find a step function $g_0(\,) \in$ S which minimizes the objective function J(.) over S. That is we would like to minimize the following function over $y_i \in (-\infty,+\infty)$ and $P_j \in (0,1)$ (where i = 1, 2, ..., L; j = 1, 2,..., L-1):

$$J(g().) = \int_{-\infty}^{y_1} F_x^2(a)da + \int_{y_1}^{y_2} [F_x(a) - P_1]^2 da$$

$$+ \int_{y_2}^{y_3} [F_x(a) - P_2]^2 da + ... + \int_{y_{L-1}}^{y_L} [F_x(a) - P_{L-1}]^2 da$$

$$+ \int_{y_L}^{\infty} [F_x(a) - 1]^2 da, \qquad \text{(B.6)}$$

It follows from Theorem B.l that if $g_0(x)$, which is defined by

$$g_0(x) = \begin{cases} 0, & x < y_{1,0}, \\ P_{i,0}, & \text{if } y_{i,0} \le x < y_{i+1,0}, \\ 1, & x \ge y_{L,0}, \end{cases} \qquad i = 1,2,...,L-1 \qquad \text{(B.7)}$$

is a step function which minimizes equation (B.6), this must satisfy the following set of equations:

$$P_{1,0} = 2F_x(y_{1,0});$$

$$P_{1,0} + P_{i+1,0} = 2F_x(y_{i+1,0}), \qquad i = 1,2,3,...,L-2;$$

$$1 + P_{L,0} = 2F_x(y_L);$$

$$P_{i,0}(y_{i+1,0} - y_{i,0}) = \int_{y_{i,0}}^{y_{i+1,0}} F_x(a)da \qquad i = 1,2,...,L-1 \qquad \text{(B.8)}$$

Using equation (B.8) and Theorem B.2, the discrete random variables (with L possible values where L = 1,2,..,8) which approximate the normal random variable with zero mean and unit variance have been numerically obtained and are tabulated in Table B.1.

Let y_0 be the optimum discrete random variable with L possible values $y_{1,0}, y_{2,0},..., y_{L,0}$ which approximate the normal random variable with zero mean and unit variance, and let $P_{i,0}$ be defined by $\Pr ob\{y_0 = y_{i,0}\}$. Let z_0 be the optimum discrete random variable with L possible values $z_{1,0}, z_{2,0},..., z_{L,0}$ which approximates the normal random variable with mean μ and variance σ^2; and let $P'_{i,0}$ be defined by $\Pr ob\{z_0 = z_{i,0}\}$. By equation (B.8) , it can easily be verified that

$$z_{i,0} = \sigma y_{i,0} + \mu, \qquad P'_{i,0} = P_{i,0}, \qquad i = 1,2,..., L \qquad \text{(B.9)}$$

Table B.1: Discrete Random Variables Approximating the Gaussian Random Variable with Zero Mean and Unit variance

Number of possible values of y_0	i	L possible values and corresponding probabilities of y_0							
		1	2	3	4	5	6	7	8
L=1	$y_{i,0}$	0.000							
	$P_{i,0}$	1.000							
L=2	$y_{i,0}$	-0.675	0.675						
	$P_{i,0}$	0.500	0.500						
L=3	$y_{i,0}$	-1.005	0.0	1.005					
	$P_{i,0}$	0.315	0.370	0.315					
L=4	$y_{i,0}$	-1.219	-0.355	0.355	1.219				
	$P_{i,0}$	0.223	0.227	0.227	0.223				
L=5	$y_{i,0}$	-1.376	-0.592	0.0	0.592	1.376			
	$P_{i,0}$	0.169	0.216	0.230	0.216	0.169			
L=6	$y_{i,0}$	-1.499	-0.767	-0.242	0.242	0.767	1.499		
	$P_{i,0}$	0.134	0.175	0.191	0.191	0.175	0.134		
L=7	$y_{i,0}$	-1.599	-0.905	-0.423	0.0	0.423	0.905	1.599	
	$P_{i,0}$	0.110	0.145	0.162	0.166	0.l-62	0.145	0.110	
L=8	$y_{i,0}$	-1.683	-1.018	-0.567	-0.183	0.183	0.567	1.018	1.683
	$P_{i,0}$	J.093	0.123	0.139	0.145	0.145	0.139	0.123	0.093

y_0, is the discrete random variable with L possible values $y_{1,0}, y_{2,0}, ..., y_{L,0}$ which approximates the normal random variable with zero mean and unit variance.

$y_{i,0}$; is the i[th] possible value of $y_0 : P_{i,0} = \Pr ob\{y_0 = y_{i,0}\}$.