AN N-SCAN BACK ALGORITHM FOR DATA ASSOCIATION IN MULTITARGET TRACKING

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

BY

ÖNDER HALUK TEKBAŞ

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

SEPTEMBER 1996

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur ÖZTÜRK
Director

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

Prof. Dr. Fatih CANATAN
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. Mete SEVERCAN

Prof. Dr. Yalçın TANIK

Prof. Dr. Kerim DEMİRBAŞ

Assoc. Prof. Dr. Kemal LEBLEBİCİOĞLU

Dr. Murat EREN

ABSTRACT

AN N-SCAN BACK ALGORITHM FOR DATA ASSOCIATION IN MULTITARGET TRACKING

Tekbaş, Önder Haluk

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kerim Demirbaş

September 1996, 113 Pages

In this thesis, basics of multitarget tracking systems and existing multitarget tracking methods are surveyed. A new N-scan back algorithm for data association in multitarget tracking systems is presented: Possible paths of each target are presented by trees. Expected measurements corresponding to the nodes of these trees are estimated. The distances of real measurements to these estimated measurements are calculated.The sums of these distances for nodes, each taken from different level, are found. The measurements are assigned to the target which yields the minimum sum of distances.

Simulation results are included and the performance of the new algorithm is compared with "Track Splitting Approach". Simulation results show that the new algorithm's performance is better than that of "Track Splitting Approach"

Keywords: Multitarget Tracking, Multiple-Target Tracking, Tracking, Data Association, Correlation

ÖZ

ÇOKLU HEDEF TAKİBİNDE KARŞILAŞILAN VERİ EŞLEME PROBLEMLERİ İÇİN N-TARAMA PERİYODU GERİ GİDEBİLEN BİR ALGORİTMA

Tekbaş, Önder Haluk
Yüksek Lisans., Elektrik-Elektronik Mühendisliği Bölümü
Danışman: Prof. Dr. Kerim Demirbaş

Eylül 1996, 113 Sayfa

Bu tezde, çoklu hedef takip sistemlerinin temelleri ve mevcut hedef takip metodları incelenmiştir. Çoklu hedef takip sistemlerindeki veri eşleme problemleri için yeni bir N-Tarama periyodu geri gidebilen bir algoritma geliştirilmiştir: Her hedefin olası yolları bir ağaç yapısıyla ele alınmış; bu ağacın düğümlerine karşılık gelen, beklenen gözlem ölçüm değerleri bulunmuş; gerçek ölçüm değerleri ile beklenen ölçüm değerleri arasındaki uzaklıklar hesaplanmış; herbiri farklı tarama perioduna ait olan düğümler için bu uzaklıklar toplanmış ve ölçüm değerleri, en küçük toplam uzaklık değerini sağlayan hedefe tahsis edilmiştir. Ayrıca, bu algoritmanın bilgisayar uygulaması ilgili simülasyon sonuçları ve performansının "Track Splitting Approach" algoritması ile karşılaştırılması da sunulmuştur. Simülasyon sonuçları yeni algoritmanın performansının "Track Splitting Approach" algoritmasından daha iyi olduğunu göstermiştir.

Anahtar kelimeler: Çoklu hedef takibi, Hedef izleme, Veri eşleme

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Prof.Dr. Kerim Demirbaş for his guidance and insight throughout the research. And the technical assistance of Ahmet Şenol is greatefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In a surveillance system using multitarget tracking (MTT) algorithms, the aim is to track several targets using one or more sensors. For that purpose, after deleting background noise sources (such as radar ground clutter) and internal noise sources (such as thermal noise), sensor measurements are partitioned into sets of observations, called tracks, each of which is produced by a target.

In multitarget tracking (MTT) systems, first, all measurements taken from sensors are analysed and extraneous measurements (background noise sources and internal noise sources) are eliminated. Then, remaining data are classified into subgroups, each belonging to one track (one target). This classification is named as "data association" or "correlation". One can find various methods, in the literature, (e.g., in [1], [2], [3], [4], [5], [6], [7] and [8]) for data association.

In this thesis, basics of multitarget tracking systems, existing tracking methods in the literature (e.g., in [1], [2], [3] and [9]) and their data association algorithms are surveyed. A new data association algorithm is developed and simulation results of the implementation of this algorithm are presented.

The organisation of this thesis is as below:

Chapter 1 is an introduction. Chapter 2 presents basics of multitarget tracking systems. In chapter 3, multitarget tracking methods, documented in the literature (e.g., in [1], [2], [3] and [9]), are reviewed and their differences are stated. In chapter 4, data association in multitarget tracking systems is specifically analysed, an algorithm is developed for data association and simulation results are presented. Chapter 5 concludes this thesis.

CHAPTER 2

THE BASICS OF MULTITARGET TRACKING

2.1. INTRODUCTION

In a surveillance system employing one or more sensors(*) together with computer subsystems, sensor measurements (reports) could have come from following sources:

.targets of interest,

.background noise sources such as radar ground clutter,

.internal sources such as thermal noise.

We should be able to track interested targets under this condition using some multitarget tracking algorithms (e.g., algorithms in [1], [2], [3] and [9])

The multitarget tracking objective is to reduce measurements related to background and internal noise sources and then to partition remaining sensor data into sets of observations, called tracks, produced by the same target. Once tracks are formed, relevant quantities (such as target velocity or future predicted position) can be computed for each track.

(*) Typical sensors are:
.radar
.sonar
.some optical sensors

Although we are able to use several sensors in a tracking system, in this thesis, we will concentrate on "Single-Sensor Multitarget Tracking".

2.2. ELEMENTS OF A BASIC MULTITARGET TRACKING (MTT) SYSTEM [1]

We will introduce basic elements of an MTT system by considering a simple recursive processing system (Figure 2.1)

Figure 2.1 gives a representation of the elements of a simple MTT system. 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. Our purpose is to show how the elements interrelate.

In a surveillance system, sensors generate reports (including observations or set of data) at regular or irregular intervals of time, called scans. As a notational convenience, in this thesis, the lower case "j" will be used as a running index for these consecutive scans and lower case "k" will be used for the last scan. Assume that data for scan time k, are received from the sensor, and the processing loop described in Figure 2.1 is to be performed. Incoming data (observations) are first considered for the update of existing tracks. Gating which will be explained in Section 2.2.3., determines a rough association of data to existing tracks and then, a more refined data association is realized by a correlation (or data association) algorithm. (Data association algorithms will be explained conceptually in Section 2.2.4. And Chapter 2 will give much more detailed information about existing data association (or correlation) algorithms.)

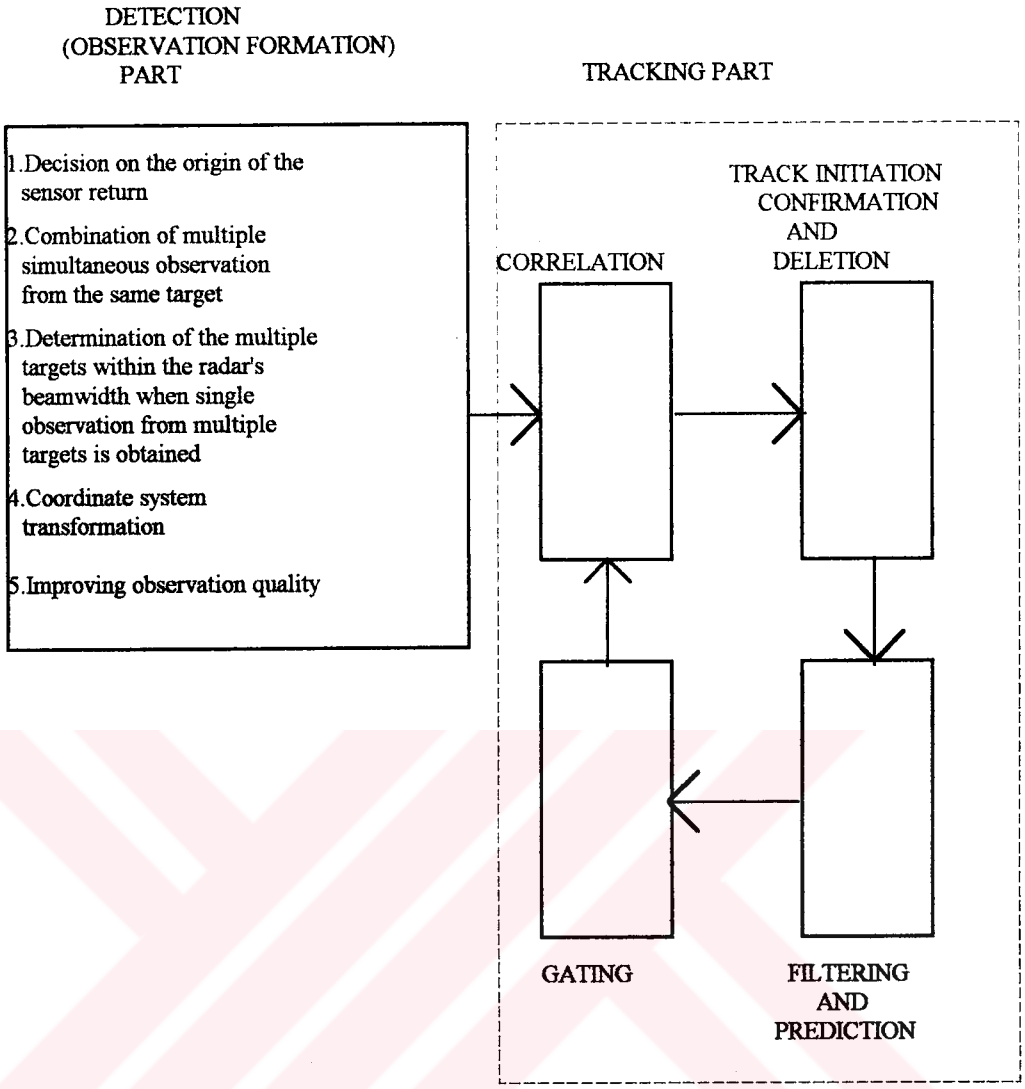


Figure 2.1: A Simple MTT System (A Simple Processing Loop)

Observations 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

(*) Concepts related to confirmation criteria will be give in Section 2.2.5

quality tracks (i.e. tracks that we can not assign any observations 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 (positions or other relevant quantities) 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.

2.2.1. Detection (Observation Formation) Part of The MTT System

The functions in detection part of Figure 2.1 are described in this section:

2.2.1.1. Decision on the origin of the sensor return

In detection (observation formation) part of an MTT system, decision on the origin of the sensor returns (i.e. whether they come from interested targets or result from extraneous sources, such as potential false alarms produced by noise and radar clutter) is performed. For that purpose we simply compare the incoming measurement signal power to a threshold level. Signals whose power levels are below that threshold are discarded, and those of higher levels are fed to the tracking part of MTT system. But, here we should ask the following question: "How should we determine that threshold level?". For this purpose and for some other requirements, one can perform feedback operation between tracking and detection parts of MTT system (Figure 2.2). In other words, results

from tracking part of an MTT system can be used as feedback to affect the detection (observation formation) part. And, probably the most important use of this feedback is "adaptive threshold control".

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

Figure 2.2 shows how information (feedback) from the tracking part can be fed into the detection part.

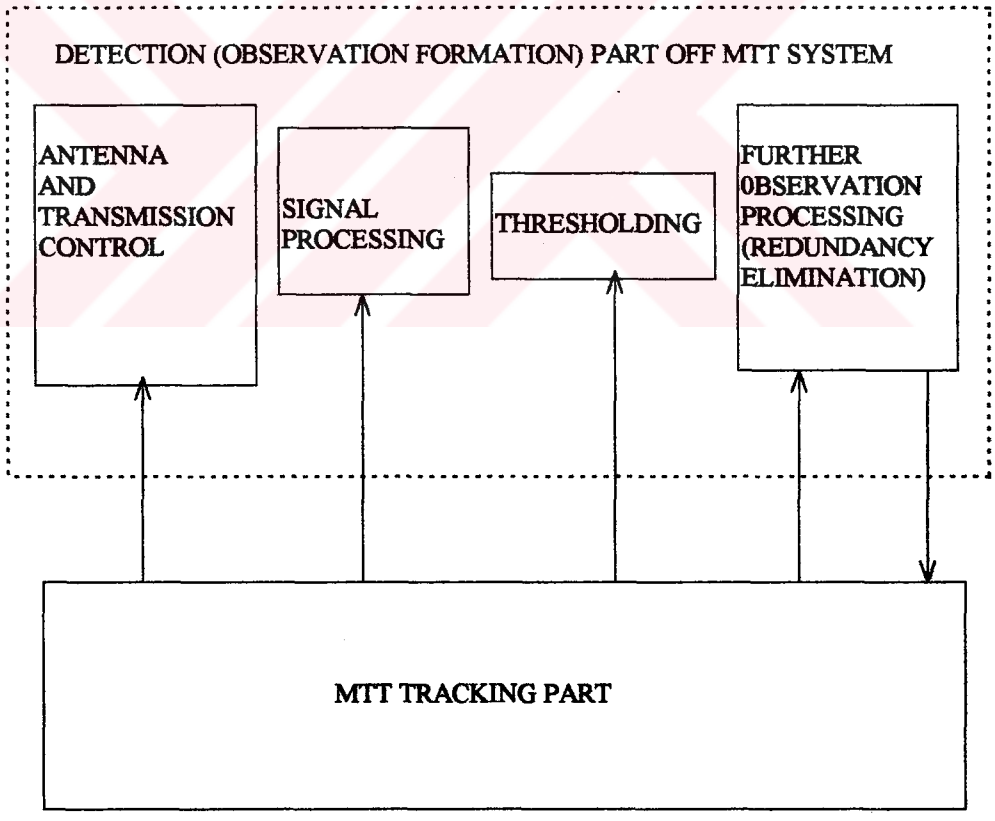


Figure 2.2: Feedback Between Detection and Tracking Parts of an MTT system

Feedback can first be applied to determine antenna positioning, resource allocation, and transmission control. Basically, the approach is to use a higher sampling rate in regions where established tracks exist. So, we increase the time on target in order to improve the detection probability.

In addition, in some tactical situations, our transmitted radar signal power should be controlled so that we don't miss the target while minimizing the probability that the radar signal is detected by hostile aircraft or groundbased tracking systems. This may be realized by limiting the system to intermittent search and adjusting the transmitted power according to the target range while illuminating the target.

Feedback is also used to affect signal processing. In the signal processing unit, special processing that can not feasibly be done everywhere may be performed in regions of expected target returns.(*)

In the use of feedback for threshold control, it is aimed to reduce the threshold in region of expected target return (this leads higher probabilities of detection and false alarm) and to increase it in region of greater than average background (clutter) power (which yields reduced probability of false alarm but also reduced probability of detection). Thus, by selective choice of threshold it may be possible to obtain a tolerable false alarm rate without loss of tracking performance.

(*)one application in this area is to form finer Doppler (range rate) filters in the region of expected target return. Another application would be to apply special processing to recognize and reduce the effects of JEM (Jet Engine Modulation) returns in the vicinity of the expected target range rate.

2.2.1.2. Combination of multiple simultaneous observations from the same target

In order to simplify the design of MTT elements we should be sure that sensor does not produce multiple simultaneous observations from the same source. But, in practice, we have 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 we encounter with multiple simultaneous observations received on different bars from the same target.) These observations should be combined and the composite observation should be 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. Trunk, [10], presents an algorithm for combining multiple simultaneous detections from the same source.

2.2.1.3. Determination of multiple targets within the radar's beamwidth when single observation from multiple targets is obtained

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. But, 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 is another function of the "Detection (Observation

Formation)" part of MTT system. Reference [11] proposes a method for determining target multiplicity within the radar's beamwidth.

2.2.1.4. Coordinate system transformation

Another function performed during the process of observation formation is the transformation of the measured kinematic quantities (e.g., position) to a coordinate system which is more convenient for performing the functions of other MTT elements.

2.2.1.5. Improving observation quality

A fifth function of the "Detection (Observation Formation)" part of MTT system is that of improving observation quality. For example, certain known characteristics of the radar signal return may be used to remove the undesired components caused by jet engine modulation (JEM).

2.2.2. Filtering and Prediction

Actually target tracking is a problem of estimation. 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 [12]. 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 (which will be given later) 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, we will concentrate on linear estimation problems 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:

$$Z(j)=h(j,x(j),w(j)) \quad j=1,2,3,.....,k \qquad (2.1)$$

where w(j) is the measurement noise at the j^{th} scan.

The aim is to find a function

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

(*) Actually, there are two commonly used approaches to filtering and prediction problem:

1. Filters using fixed tracking coefficients for the statistics of measurement noise and target dynamics. For example, α-β filter.
2. Kalman filtering which generates time-variable tracking coefficients However, Kalman filter will be analyzed in this thesis.

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

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

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

$$\tilde{x}(k) \triangleq x(k)-\hat{x}(k) \tag{2.4}$$

Parameter ,x, (state of a target) can be time-invariant or time varying, as stated above. In Section 2.2.2.1, we will analyze time invariant case (i.e., static estimation problem). Dynamic estimation problem will be analyzed in Section 2.2.2.2.

2.2.2.1. Linear Estimation in Static Systems (I.e., Parameter, x, is Time-invariant)

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 or Fisher 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 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:

$$p(x|Z)= \frac{P(Z|x)p(x)}{p(Z)} = \frac{1}{c} p(Z|x)p(x) \qquad (2.5)$$

where c is a normalization constant (doesn't 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)

The Non-Bayesian (Likelihood Function) Approach

In contrast to the above case, in the non-Bayesian approach there is no prior pdf associated with the parameter and thus one can not define a posterior pdf 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)

If in a given set of measurements, errors are additive, zero-mean, Gaussian and independent, then ML estimate coincides with the LS estimate and MAP estimate coincides with MMSE estimate.

In Appendix A, estimation of Gaussian random vectors is given as an example.

Now, dynamic estimation problem will be explained:

2.2.2.2. Linear Estimation in Dynamic Systems (I.e., Parameter, x, is Time-varying)

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

$$x(k+1)=F(k)x(k)+G(k)u(k)+v(k) \qquad k=0,1,2,.... \qquad (2.8)$$

where x(k) is the n_x-dimensional random state vector (e.g., position, velocity, ..., etc), 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(*) :

$$E[v(k)v(k)']=Q(k) \qquad (2.9)$$

The measurement equation is

$$z(k)=H(k)x(k)+w(k) \quad k=1,2,.... \qquad (2.10)$$

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

$$E[w(k)w(k)']=R(k) \qquad (2.11)$$

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 can be nonstationary.

(*)vectors which are primed with symbol (') are the transpose of the corresponding vectors.

The initial state x(0), in general known, can be modeled as a random vector, Gaussian distributed with known mean and covariance. The two noise sequences and the initial state are assumed mutually independent.

Equations (2.8) through (2.11) and the condition stated in the previous paragraph constitute the linear Gaussian (LG) assumption.

In the dynamic equation ,Equation (2.8), 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_x 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

$$E\left[[\Gamma(k)v(k)][\Gamma(k)v(k)]'\right]= \Gamma(k)Q(k)\Gamma(k)' \qquad (2.12)$$

The linearity of Equation (2.8) and Equation(2.10) leads to the preservation of the Gaussian property of the state and measurements - this is a Gauss-Markov process.

The following notations will be used:

The conditional mean

$$\hat{x}(j|k) \overset{\Delta}{=} E[x(j)|Z^k] \qquad (2.13)$$

where

$$Z^k \triangleq \{z(j),\ j \le k\} \tag{2.14}$$

denotes the sequence of observations available at time k.

It can be proven that the estimate of a random vector x(k) in terms of z(k), according to the minimum mean square error (MMSE) criterion, is the conditional mean of x(k) given z(k) (proof is available at page 98 of [12]). Considering the above fact, the conditional mean given in Equation (2.13) will be defined as 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

$$\tilde{x}(j|k) = x(j) - \hat{x}(j|k) \tag{2.15}$$

The conditional covariance matrix of x(j) given the data Z^k or the covariance associated with the estimate (Equation (2.13))is:

$$P(j|k) = E\left[[x(j) - \hat{x}(j|k)][x(j) - \hat{x}(j|k)]' | Z^k\right] = E[\tilde{x}(j|k)\,\tilde{x}(j|k)' | Z^k] \tag{2.16}$$

2.2.2.3. Kalman Filtering

In the previous section we stated some basics of estimation concept. 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

Algorithm, [12]

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

$$\hat{x}(k|k)=E[x(k)|Z^k] \qquad (2.17)$$

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

$$P(k|k)=E\left[[x(k)-\hat{x}(k|k)][x(k)-\hat{x}(k|k)]'|Z^k\right] \qquad (2.18)$$

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

The recursion that yields the state estimate at k+1 and its covariance can be obtained from the static estimation equations. Detailed derivation of the dynamic estimation algorithm can be found in Appendix B. However, as an overview, we can say 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 2.3. Note that at every stage k the entire past is summarized by the sufficient statistic $\hat{x}(k|k)$ 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.

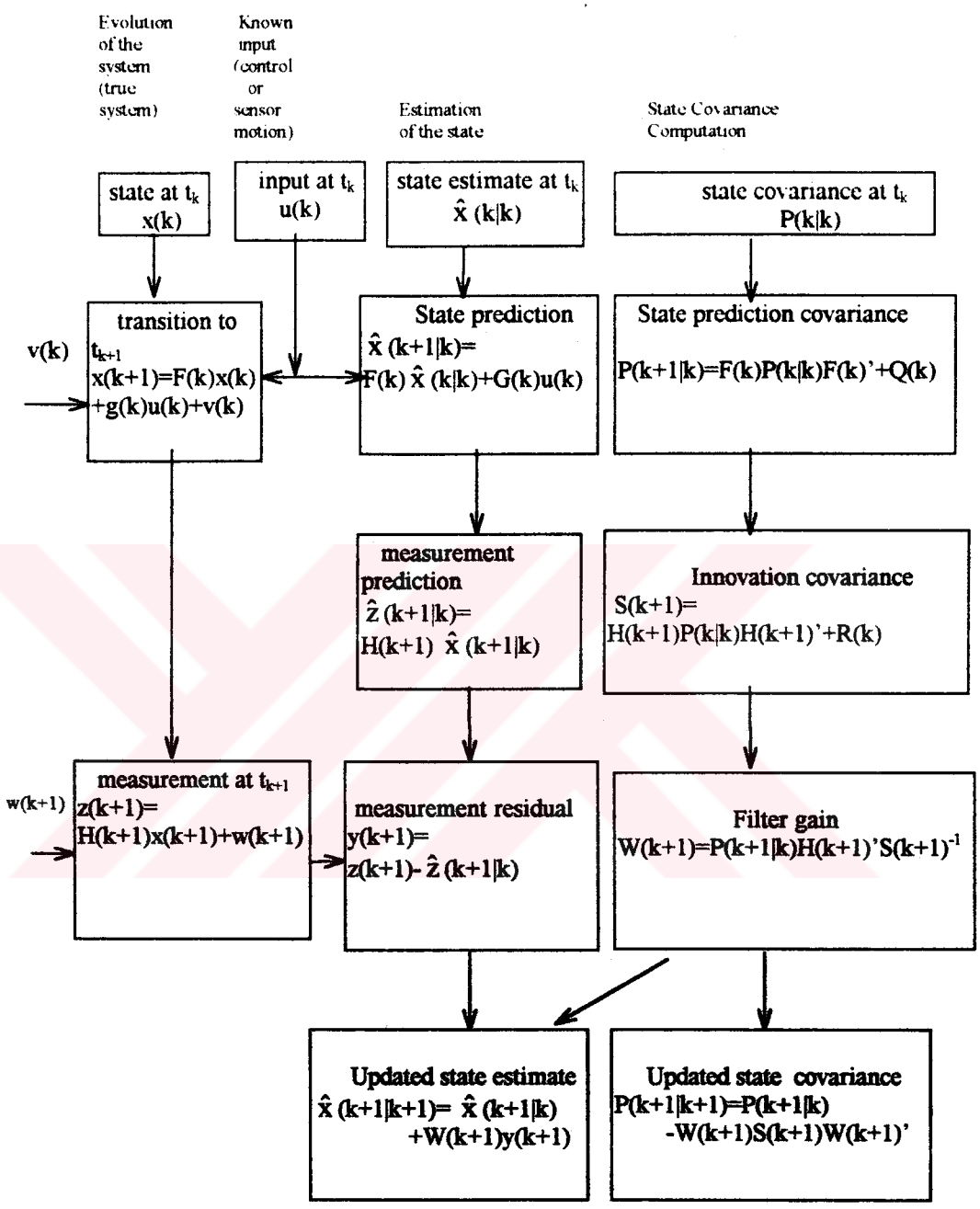


Figure 2.3: The Flowchart of one Cycle of Kalman Filtering

The state estimate cycle consists of

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.

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.

2.2.3. Gating, [1]

Gating is the first part of the correlation algorithm (data association algorithm) used to decide if an observation belongs to a previously established target track or to a new track. In other words, gating is a coarse test that classifies an observation into one of the following two categories:.

1. Candidate for track update

2. Initial observation for new tentative track

Gates are established, and gating is performed in the following general manner:

Assume that, we are at scan time "k+1". The measurement at this time is:

$$z = z(k+1) = H(k+1)\, x(k+1) + w(k+1) \qquad (2.19)$$

where

H(k): The measurement matrix, and

w(k): Zero-mean, white Gaussian measurement noise with covariance matrix R(k).

The vector difference between measured and predicted quantities,

$$y = y(k+1) = z(k+1) - H(k+1)\hat{x}(k+1 \mid k) \qquad (2.20)$$

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

$$S = S(k+1) = H(k+1)P(k+1 \mid k)H(k+1)' + R(k) \qquad (2.21)$$

where $P(k+1 \mid k)$ is the prediction covariance matrix.. Assume that the measurement is M dimensional. The norm of the residual vector is defined as :

$$d^2 = y'S^{-1}y \qquad (2.21)$$

and the M-dimensional Gaussian probability density, related with the residual is:

where |S| is the determinant of S

If the probability of detection is unity or no extraneous returns are expected, considering the Gaussian density above, we can say that the optimal gate size should be infinite in order to obtain an optimal correlation performance. But, even for the above case , we should determine a finite gate size in order to reduce the number of observation to track pairings.

But, for the most part of the tracking problems the probability of detection is not unity or there exist extraneous returns in the surveillance volume. So, the determination of a finite gate size is more meaningful.

In general, three types of gating method is used in MTT systems. They are:

1. Rectangular gates
2. Ellipsoidal gates
3. Gates with more than one level.

Rectangular Gates

While we are using this method, an observation is said to satisfy the gate of a given track if all elements, y_l, of the residual vector y satisfy the relationship

$$|z_l - \hat{z}_l| = |y_l| \leq K_{Gl}\,\sigma_r \qquad (2.23)$$

where σ_r is the residual standard deviation as defined in terms of the measurement (σ_0) and the prediction (σ_p) variances:

$$\sigma_r = \sqrt{\sigma_o^2 + \sigma_p^2} \qquad (2.24)$$

Now, we will concentrate on how to chose K_{Gl} Assume that for all dimensions $K_{Gl}=K_G$ We are going to make use of the probability (P_G) of a valid observation satisfying the gating relationship which is defined as

$$P_G = [1-Pr(|t| \geq K_G)]^M = 1 - MPr(|t| \geq K_G) + \text{higher order terms} \qquad (2.25)$$

where $Pr(|t| \geq K_G)$ is the probability of the magnitude of a standard, normalized Gaussian random variable ($t \sim N(0,1)$) exceeding K_G . If we assume that $Pr(|t| \geq K_G)$ is small enough we can neglect higher order terms and we get

$$P_G = [1-Pr(|t| \geq K_G)]^M \cong 1 - MPr(|t| \geq K_G) \qquad (2.26)$$

For example, if K_G is chosen to be 3,

$$Pr(|t| \geq 3) = 2Pr(t \geq 3) = 2(0.00135) = 0.0027$$

and, for 4-dimensional radar measurement (azimuth, elevation, range, range rate), the theoretical probability of a valid observation satisfying the gating test for K_G 3 is .

$$P_G = [1-Pr(|t| \geq 3)]^4 \cong 1-4Pr(|t| \geq 3) = 1-4(0.0027) \cong 0.99$$

So the choice of $K_G=3$ is quite meaningful. But in the presence of target maneuver this value may be chosen around (3.5). Or, even in the above case, a one percent probability of rejecting a valid return in every scanning period, can lead to a much larger cumulative value (>20 percent) over the life of a typical track. So, larger values of K_G (K_G >3) may be used.

Ellipsoidal Gates

In this method, the gate size is determined by the following relation;

$$d^2 = y'S^{-1}y \leq G \qquad (2.27)$$

where

d^2 is the norm of the residual vector

S is the residual covariance matrix (defined in Equation (2.21))

y is the measurement residual vector (defined in Equation (2.20))

References [4] and [5] proposes a maximum likelihood gate G_0 which is defined as

$$G_0 = 2 \ln \left[\frac{P_D}{(1-P_D)\beta(2\pi)^{M/2}\sqrt{|S|}} \right] \qquad (2.28)$$

References [4] and [5] proposes a maximum likelihood gate G_0 which is defined as

$$G_0=2 \ln \left[\frac{P_D}{(1-P_D)\beta(2\pi)^{M/2}\sqrt{|S|}} \right] \quad (2.28)$$

where

P_D is the probability of detection.

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

M is the measurement dimension.

The probability (P_G) that a valid observation will fall within the gate G can be found by

$$P_G= \int \int_{\text{over ellipsoidal volume } V_G} f(y)dy_1,\ldots,dy_M \quad (2.29)$$

Comparison of Gating Methods [1]

This comparison is realized by considering gating volumes of the two methods mentioned above.

The volume of an M-dimensional rectangular gate normalized with respect to residual variances is:

$$V_{G1}(M)=(2K_G)^M \quad (2.30)$$

The normalized volume of an M-dimensional ellipsoidal gate is:

$$V_{G1}(M)=C_M G^{M/2} \tag{2.31}$$

where

$$C_M = \frac{\pi^{M/2}}{G((M/2)+1)} = \begin{cases} \dfrac{\pi^{M/2}}{(\frac{M}{2})!} & , \text{M even} \\ \dfrac{2^{M+1}(\frac{M+1}{2})!\pi^{\frac{M-1}{2}}}{(M+1)!} & , \text{M odd} \end{cases} \tag{2.32}$$

In Equation.(2.32) G(.) stands for Gamma function

Now, we define the ratio of the two volume expressions above as

$$r(M)=\frac{V_{G1}(M)}{V_{G2}(M)} \tag{2.33}$$

Table 2.1, taken from [1], give appropriate values of K_G and G for different measurement dimension (M). Referring to this table, we can say that the relative volume required by the rectangular gating method increases rapidly as the measurement dimension increases. This means that the probability of accepting an extraneous return for the rectangular gating method increases as compared to that of ellipsoidal gate. That is, the ellipsoidal gating decreases the probability of an extraneous observation falling within the gates. However, use

of the ellipsoidal gate requires somewhat more computation because the normalized distance function must be computed using Equation.(2.27) An alternative way of gating is to do a coarse gating test first, using relatively large (K_G~ 4.0) rectangular gates, and then to use ellipsoidal gates for those observation-to-track pairings that satisfy the coarse gating. This process has been found to decrease significantly the number of required calculations.

Table 2.1:Comparative Probabilities Of Accepting Extraneous Observations
(P_G is assumed to be 0.99)

M	K_G	G	r(M)
2	2.81	9.21	1.09
3	2.94	11.34	1.28
4	3.04	13.28	1.59
5	3.09	15.09	2.0

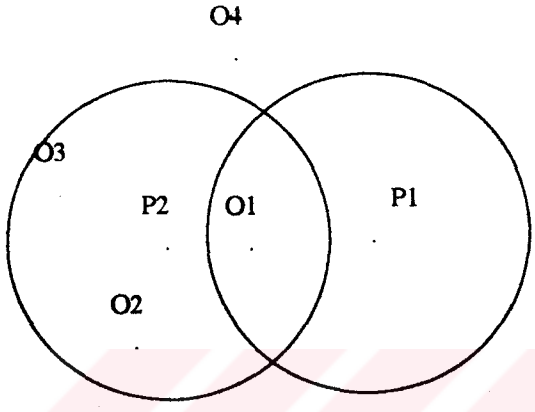
2.2.4. Data Association (Correlation)

The correlation algorithm takes the output of the gating function and makes final observation-to-track assignments. In the case where a single observation is within the gate of a single track, the assignment can be immediately made. However, for closely spaced targets, it is more likely that conflict situations, such as those shown in Figure 2.4 will arise. Basically there exist two types of conflicts situations:

1. An observation may fall within the gates of multiple target tracks (such as O1 of Figure 2.4.)

2. Multiple observations may fall within the gate of a target track. (such as O2, O3 of Figure 2.4.)

In chapter 2, We will introduce DATA ASSOCIATION algorithms in more detail.



O1,O2,O3 : MEASURED OBSERVATION LOCATIONS
P1,P2 : PREDICTED OBSERVATION LOCATIONS

Figure 2.4: An Example of a Correlation Conflict Situation

2.2.5. Track Initiation, confirmation, and Deletion

Depending on the correlation algorithm used, some observations might not be assigned to existing tracks. Those observations are candidate for initiation of new tentative tracks or they are deleted. In order to maintain accurate tracking it is best to initiate new tracks whenever initiation may be questionable, but then to make confirmation requirements more stringent.

Once a tentative track is formed, a confirmation logic is usually required because the probability of a single observation being from an extraneous source

confirmation criteria are three-out-of-four or three-out-of-five. However, for a radar system in which Doppler (range rate) information is available, two observations may suffice so that two-out-of-three or two-out-of-four criteria may be used. If the required confirming observations are not received, the previous observations are dismissed as false alarms.

If we don't have no correlating observation to update a track within a sufficiently long time period or within a finite number of consecutive scans, that track must be deleted. It might be best to delete a track just because of its low quality.

CHAPTER 3

EXISTING MULTITARGET TRACKING (MTT) METHODS

3.1. INTRODUCTION

As we mentioned in the previous chapter, the purpose of the multitarget tracking (MTT) methods is to associate new observations with one of the existing tracks or to initiate a new tarck. In the litterature, (e.g., in [1], [2], [3] and [9]), one can find several MTT methods developed for the above purpose.

There exist two main distinctions when comparing multitarget tracking(MTT) methods:

The first distinction:

Methods are of **batch** or **recursive** (sequential) type.

The second distinction:

Methods use **"nearest-neighbor"** or **"all-neighbor"** approaches.

3.1.1. Differences between batch and recursive (sequential) methods?

Batch processing techniques represent the ideal situation (optimal solution to MTT) where no information is lost due to preprocessing because all observations are processed together. If we use lower case "j" as a running

index, and the lower case "k" as the index for the last scan of a set of observations then, using batch processing, at scan k, the data received on that scan as well as the data received on all previous scans j=1,2,3,....,k-1 would be processed together to form target tracks and state (positions, velocity, et cetera.) estimates. On the other hand, using **recursive (or sequential)** methods, processing is done at each scan using only data received on that scan to update the results of previous processing. For example, in the simplest recursive processing, observations received at scan k would be compared for correlation with tracks formed on the previous (k-1) scan, but would not affect the previous correlation decisions.

Using recursive methods, we have an advantage when computational capacities are considered because we process only data in a particular scan. But, as mentioned above, using this approach, data association decisions are irrevocable once made.

The computations associated with batch processing rapidly begin to exceed computational capacities as more and more data are received. Thus, **batch methods are usually ideal solutions which are not computationally feasible.**

A modified version of batch processing, known as **deferred decision approach**, allows the final decision on difficult data association situations to be postponed until more information, such as the next frame of data is received. Alternative hypotheses are formed and evaluated when later data are received. This approach clearly has the potential for ultimately achieving a much higher correct decision probability then the sequential decision method. However, to

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

3.1.2. Differences between "nearest-neighbor" and "all-neighbor" approaches?

A second distinction between multitarget tracking algorithms arises in solving correlation conflict situations in which multiple observations fall within the same gate (or gates) or observations fall within the gates of more than one track. There are two basic approaches to this problem. The first approach, called **"nearest-neighboor,"** looks for a unique pairing so that at most one observation can be used to update a given track. Using this approach, the optimal solution is obtained by assigning observations to tracks in order to minimize the total summed distance from all observations to the observation prediction positions of existing tracks.

To illustrate one suboptimal solution, the example shown in Figure 2.4 will be solved.

In Figure 2.4. O1, O2, O3 and O4 are new observation positions received at last scan k. P1 and P2 are predicted observation positions for track 1 (T1) and track 2 (T2) respectively. Circles show the gated regions of the two tracks.

The suboptimal solution is reached using the following rules:

1. O1 is assigned to T1 because O1 is the only observation within the gates of T1 while T2 has other observations (O2,O3) within its gate.

2. O3 is assigned to T2 because the distance between O3 and P2, is smaller than the distance between O2 and P2.

3. O4 can, without question, be used to initiate a new track, but new track initiation using O2 may be restricted. This restriction is based upon the practical consideration that multiple observations within the gate of a single established track are often the result of a failure in the observation redundancy-elimination logic, discussed in Section 2.2.1.2. Thus, this restriction serves to prevent initiation of extraneous tracks.

In Section 3.2, two different suboptimal and an optimal solution are given for an other conflict situation which is presented in Figure.3.1.

The alternative to nearest-neighbor correlation is the **"all-neighbor"** approach, which incorporates all observations within the neighborhood, as defined by the gate around the predicted target position. The position update is then based on a weighted sum of all observations, with the weighting calculated using probability theory. For the example shown in Figure 2.4, this means that O1, O2, and O3 would all be used to update the position estimate of T2, but the weighting of O1 would be much smaller than the weightings for O2 and O3, since O1 is also within the gate of T1.

Considering two main distinctions between MTT methods discussed above, we are able to classify MTT methods in litterature into one of the following methods:

1. Method using sequential nearest neighbor correlation (standard (sequential) nearest neighbor method), [2]

2. Methods using probabilistic (non-bayesian) approaches, [2], [9]. Two of them are:

.The track splitting approach

The maximum likelihood method

3. Methods using bayesian approaches, [9]

a-) Nearest-neighboor, single scan (or N scan back), a posteriori algorithm. Basically this is a sequential method. But difficult decisions may be deferred until more data are received. This method is generally called "Multiple Hypothesis Tracking" (MHT) approach.

b-) All-neighboor, single scan, a posteriori algorithms. There are essentially two types of this approach.

i-) Probabilistic Data Association (PDA) which is suitable for tracking only one single target.

ii-) Joint Probabilistic Data Association (JPDA) which is suitable for tracking multiple targets.

c-) Multiple-scans, all neighbors, a posteriori algorithm. This method is essentially suitable for tracking only one single target but may also be used in tracking several targets. There are essentially two types of this approach.

i-) "Batch processing approach" where number of scans is infinite. This is the OPTIMAL SOLUTION between all of the solutions using Bayesian rule (i.e. algorithms using a posteriori probabilities)

ii-) "Deferred decision approach" where number of scans is finite.

The methods mentioned above will be discussed in following sections.

3.2. STANDARD (SEQUENTIAL) NEAREST NEIGHBOR METHOD, [2]

Assumptions:

1 It is assumed that N_i observations are taken for each scan.

2. N_i observations taken for each scan, are associated with the existing tracks at same time.

3. Number of targets are not known. Therefore, new tracks can be initiated after each scan (after N_i observations are taken).

4. It is assumed that only one observation is received for each target at each scan.

5. It is assumed that any observation belonging to at least one of the existing gates, will be used to update one of the established tracks, i.e., they can not initiate a new track. Candidate observations for track initiation are those which belong to none of the existing gates.

In this method, each track is defined by its state vector . Existing established tracks, if any, first are extrapolated to the current sample time and validation gates are computed. An assignment matrix, which will be explained below, is computed. The nearest neighbor association rule is invoked. Established tracks are updated, using some filters such as Kalman Filter

Unassigned detections are processed by the track initiator. New tracks satisfying the initialization criteria are promoted to established tracks (i.e. , tentative tracks become confirmed) and their filter states and validation gates are initialized.

Established tracks that have exceeded the minimum number of sample times without an assigned detection are terminated.

The critical point of this method lies on the determination of the assignment matrix and the solution of this matrix will give the assignment of observations to existing tracks. For that reason, we will concentrate on how to form the assignment matrix and how to solve it:

An assignment matrix consists of elements which give a value related with the measurement residual (Equation (2.20)). These values are calculated as follows:

Assume a Gaussian distribution for the residual. So, the likelihood function associated with the assignment of observation j to track i (for M-dimensional measurement vector) is:

$$g_{ij}=\frac{e^{\frac{-d_{ij}^2}{2}}}{(2\pi)^{M/2}\sqrt{|S_i|}} \qquad (3.1)$$

where S_i is the residual covariance matrix for track i, and

$$d_{ij}^2 = y_{ij}^{'} S_i^{-1} y_{ij} \qquad (3.2)$$

y_{ij} = residual vector between new observation Oj and the predicted observation value, P_i ,of track "i" (T_i) defined at Equation (2.20)

The basic goal is to choose assignments that maximizes the sum of corresponding g_{ij} terms. Or alternatively, taking the logarithm of Equation (3.1), it is seen that maximization of the g_{ij} is equivalent to minimization of the quantity:

$$d^2_{G_{ij}} = d^2_{ij} + \ln|S_i| \qquad (3.3)$$

Since, using "nearest-neighbor" approach, each observation can be assigned only one track and a track can be updated using only one observation, we can only chose one element from each row and one element from each column of the assignment matrix. So we will have M assignments if we deal with an M by M assignment matrix. The choice of these M elements sould be such that the sum of corresponding $d^2_{G_{ij}}$ terms should be minimum of all possible alternatives.

This concept is explained considering the hypothetical distance values of the assignment matrix of Figure 3.2, which is related to the situation given in Figure 3.1.

In Figure 3.1., O1, O2, O3, and O4 are new observation positions received at last scan; P1, P2, and P3 are predicted observations of track 1 (T1), track 2(T2), and track 3(T3) respectively; and circles with centers located at P1, P2 and P3 show the gated regions of corresponding tracks. In Figure 3.2., same notation is used for observations, prediction observations and tracks. In addition, since we don't need to calculate the residual between a predicted

observation position and a new observation which is outside the gate of this track, these type of assignments are denoted by the symbol "X". The circled entries in Figure 3.2., show the elements of the optimal solution.

The optimal solution for this example is obtained by choosing the set of observation to track pairings which have minimum summed total distance. This set is determined by considering all possible set of observation to track pairings and is shown by circled entries in Figure 3.2. O4 which is located outside the gated regions of all tracks, can be used to initiate a new track. But, this example is a simple one. For more complex situations, some algorithms should be used to obtain optimal solution. Some of them are:

. Ford-Fulkerson Algorithm [13]

. Hungarian Algorithm [14]

. Munkres' Algorithm [15]

In some MTT problems trying to get the optimal solution may be too time consuming and so, not feasible. For that purpose a number of approximate solutions to the assignment problem have been developed. These solutions are simple to implement . But, they are not guaranteed to give the optimal assignments for all conditions. In addition, they may not be able to assign one or several observations into existing tracks which is the case occuring in "Suboptimal solution 2" which will be given in this section. These are drawbacks of the suboptimal solutions. Two of these suboptimal solutions are given below:

.Suboptimal Solution 1

1 If the number of non-zero elements is unity in a row of the assignment matrix the corresponding observation-to-track pairing is accepted , and related row and column of this pair is removed from the matrix.

2. If the number of non-zero elements is unity in a column of the assignment matrix the corresponding observation-to-track pairing is accepted, and related row and column of this pair is removed from the matrix.

3. After each application of rule 1, the reduced matrix is again subject to the rule 1.

4. After each application of rule 2, the reduced matrix is again subject to the rule 2.

5. If the number of non-zero elements in a row is still greater than one chose the observation with minimum distance.

6. If the number of non-zero elements in a column is still greater than one chose the track with minimum distance.

. Suboptimal Solution 2

1. Search the assignment matrix for the closest (minimum distance) observation-to-track pair and make the indicated assignment.

2. Remove the row and column of the observation-to-track pair identified above from the assignment matrix and repeat rule 1 for the reduced matrix.

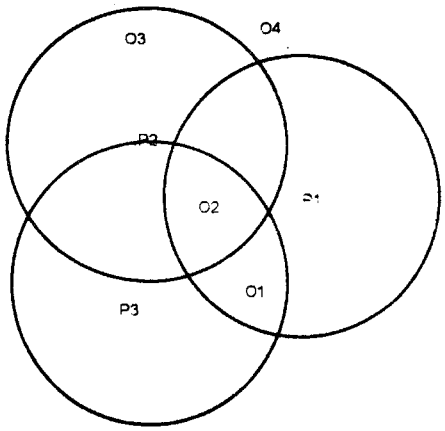
To compare the solutions obtained by the optimal and two suboptimal methods discussed above the total distance values of these solutions for the assignment matrix shown in Figure 3.2 are computed:

The total distance for the optimal assignment is 23. Using the first suboptimal method, the initial assignment is O3 to T2 because O3 is singly validated which means that there exists only one non-zero element in the column of the O3. Then, processing T1 first will lead to assignment of O2 to T1 because the corresponding distance is 6, which is less than the distance, 9, from O1 to T2. Finally the remaining assignment is O1 to T3. This leads to three assignments, but the total distance is 24, which exceeds that of the optimal solution.

The second suboptimal solution begins by making the minimum distance (3) of O2 to T2. Then, O1 is assigned to T3., but O3 is left without an assignment. Thus, the second suboptimal solution leads to less than the maximum number of possible assignments.

For, most simpler cases, these three solutions will agree. But here, the example was chosen to emphasize differences.

As a summary, we can say that the application of standard nearest neighbor algorithm uses the measurement nearest in some sense to the predicted measurement ("nearest neighbor filter"). But, it can lead to very poor results in an environment where spurious measurements occur frequently. This is because such an algorithm does not account for the fact that the measurement used in the filter might have originated from a source different from the target of interest.



O1, O2, O3, O4 = OBSERVATION POSITIONS

P1, P2, P3 = PREDICTED OBSERVATION POSITIONS

Figure 3.1: Example of a Conflict Situation

PREDICTED OBSERVATIONS \ OBSERVATIONS	O1	O2	O3
P1	(9)	6	X
P2	X	3	(10)
P3	8	(4)	X

X = OBSERVATION OUTSIDE GATE

◯ = OPTIMAL SOLUTION

Figure 3.2: Assignment Matrix for Example of Figure 2.1.

3.3. METHODS USING PROBABILISTIC (NON-BAYESIAN) APPROACHES (SEQUENTIAL), [9]

Those methods were motivated by the need to find a reasonable way of incorporating measurements of uncertain origin into existing tracks. For example, the pioneering work of Sittler [6] in this area was based on an algorithm which was of the type used before Kalman filter became popular. The method consisted of splitting the track whenever more than one return (detection) was observed in the neighborhood of the predicted measurement (i.e., in the gate of the corresponding track). Then the likelihood function of each trajectory was computed and those whose likelihood was below a threshold were dropped. Similar approaches were developed within the framework of Kalman filtering by Fraser and Meier [16] for active sonar tracking and by Smith and Buechler [17] for radar tracking. We will explain track splitting (or branching) algorithm in the following section in more detail

Another probabilistic (non-bayesian) method which is particularly suitable for track formation for several targets in the same neighborhood was developed by Morefield [18]. This method, discussed later , is based upon likelihood functions and converts the association of measurements to form tracks into an integer programming problem. .

3.3.1. The Track Splitting Approach (Sequential)

Assumptions:

1. It is assumed that N_i observations are taken for each scan.

2. N_i observations taken for each scan, are associated with the existing tracks at same time.

3. Number of targets are not known. therefore, new tracks can be initiated after each scan (after N_i observations are taken).

4. As mentioned previously, it is assumed that only one observation is received for each target at each scan.

5. It is assumed that any observation belonging to at least one of the existing gates, will be used to update one of the established tracks. I.e., They can not initiate a new track. Candidate observations for track initiation are those which belong to none of the existing gates.

6. The probability of detection is assumed to be unity. In other words, missing detections are not included.

7. False alarms are not included.

In this method, initialization of a track is usually based upon one or two correlating measurement. And, for every subsequent observation that falls in a "window" (gate) around the location the new measurement for an existing track is expected, the track is split. A Kalman filter is used to estimate the state of each split track. Since the number of branches can grow exponentially the likelihood function of each split track is computed and the unlikely ones are discarded. In this algorithm, the state of a target is assumed to evolve in time according to the equation:

$$x_{k+1} = F x_k + v_k \qquad (3.4)$$

and the corresponding measurement is given by

$$z_k = H x_k + w_k \tag{3.5}$$

with v_k and w_k being zero-mean mutually independent white Gaussian noises with known covariances Q_k and R_k. respectively. (Equation (3.4) and Equation (3.5) are the new representations of Equation (2.8) and Equation (2.10) respectively. This kind of notation is chosen for the sake of simplicity)

A particular (say, the l^{th}) sequence of measurements up to scan time k is:

$$Q^{k,l} = \{z_{1,i}, z_{2,i}, \ldots\ldots, z_{k,i}\} \tag{3.6}$$

where $z_{i,j}$ is the i^{th} measurement at time j.

The likelihood function of this sequence of measurements conditioned upon their having originated from the same target, say target i, can be written in terms of normalized residual distances as follows:

$$\Lambda(Q^k) = p(z_1, z_2, \ldots\ldots, z_k \mid Q^k \text{ is a correct track})$$

$$= p(y_1, \ldots, y_k) = \prod_{j=1}^{k} p(y_j)$$

$$= c_k \exp\left\{-\frac{1}{2}\sum_{j=1}^{k} y_j' S_i^{-1} y_j\right\} \tag{3.7}$$

where c_k is a constant related with the $1/2\pi$ term of the gaussian assumption for the density function of the residual denoted as y_i in the above equation. The S_i term in the above equation is the residual covariance matrix obtained from the standard Kalman filter. Taking the negative logarithm of the Equation (3.7) and omitting the constant term, we obtain the following equation:

$$\lambda(Q^k) = -\log \Lambda(Q^k) = \lambda(Q^{k-1}) + \frac{1}{2} y_k' S_i^{-1} y_k \qquad (3.8)$$

Then a support function $C_{ij}(k)$ associated with the assignment of observation j to track i has been obtained and using the generalized distance equation (Equation (3.3)), the recursive relationship defining this support function can be written as:

$$C_{ij}(k) = C_i(k-1) - \frac{1}{2} d^2_{G_{ij}}(k) \qquad (3.9)$$

where $C_i(k-1)$ is the support associated with track i from the previous scan, and $C_i(0)$ is the logarithm of the a priori probability associated with track i (a more complete score function reducing the effects of the assumptions made at the beginning of this section is given and solved in chapter 9 (p. 260) of [1]). Thus, a support function can be computed recursively for each track using the residual information of Equation (3.3) and Equation (3.9).

The track splitting (or branching) algorithm is defined by the Equation (3.9) and the use of three rules:

1. A gating relationship, such as that defined by Equation (2.27), is used to eliminate unlikely observation-to-track pairings.

2. If two or more tracks have similar state estimates, only the most likely, as measured by the support function, is maintained. This reduces the number of multiple tracks on the same target and the total number of tracks is also limited.

3. Tracks are eliminated if the support function falls below a threshold value.

A major drawback of the track splitting (or branching) algorithm is the fact that it does not take into account the association constraint(*) For that reason some hypothesis which are not consistent with each other may exist because an observation may belong to several tracks.

This algorithm is of sequential type and can be used in an environment where the number of targets is arbitrary and unknown. The state estimates and covariances are obtained by a standard Kalman filter based on an assumed sequences of measurements; however no probability that a sequence is correct can be obtained. The main problem with this algorithm is that, its computational and memory requirements can grow with time and saturate even large computing systems. Another disadvantage of this algorithm is that, using this approach, data association decisions are irrevocable once made.

(*) The association constraint is the fact that an observation can not truly belong to several different targets.

3.3.2. The Maximum Likelihood Method

Assumptions:

1. It is assumed that N_i observations are taken for each scan.

2. N_i observations taken for each scan, are associated with the existing tracks at same time.

3. Number of targets are not known. therefore, new tracks can be initiated after each scan (after N_i observations are taken).

4. As mentioned previously, it is assumed that only one observation is received for each target at each scan.

5. It is assumed that any observation belonging to at least one of the existing gates, will be used to update one of the established tracks, i.e., they can not initiate a new track. Candidate observations for track initiation are those which belong to none of the existing gates.

6. The probability of detection is assumed to be unity. In other words, missing detections are not included.

7. False alarms are not included.

In this approach it is aimed to find the most likely set of trajectories while "The Track Splitting Approach" deals with all "reasonably likely" tracks. In other words, the purpose is to reduce the number of tracks to be considered in the search for the most likely set of tracks. Again, each target's state is assumed to evolve according to the following equation

$$x_{k+1} = F x_k + w_k$$

and the corresponding measurement is given by

$$z_k = H x_k + v_k$$

with w_k and v_k being zero-mean mutually independent white Gaussian noises with known covariances. However, when a set of measurements is obtained, one does not know with certainty which measurement originated from which target, if any.

Assume m measurements are obtained at scan time j (j=1,2,3,..k). and from all particular sequences of measurements up to scan time k, each of which consisting of N observations, we are able to construct *I* feasible ("not too unlikely") tracks by associating sequentially returns by a simple gating procedure that carries out a coarse test.

Let us define set of these feasible tracks as a "partition" (denoted as r) of the set of measurements up to and including scan time k(denotes as Z). A partition r is "feasible" if the union of all elements of r equals to Z, and, the intersection of any two individual elements of r is a null set.

The posterior probability of one such partition being correct is

$$P\{r|z\} = \frac{p(Z|r)\, P\{r\}}{p(Z)} \qquad (3.10)$$

where P{r} is the prior probability of the partition r. In the absence of target signature or other prior information one usually assumes the priors to be equal

for all feasible partitions. In this case searching for the most likely trajectory based upon Equation (3.10) coincides with maximizing the likelihood function for the measurements over all the feasible partitions, i.e.,

$$\max p(Z|r).$$

Once the most likely set of trajectories has been chosen, the state estimates and covariances are computed in this method from a corresponding set of standard Kalman filters.

3.4. METHODS USING BAYESIAN APPROACHES, [9]

3.4.1. MHT Approach (This approach is essentially sequential but sometimes the final decision may be deferred for several scans in order to receive more data so that difficult situations are easily solved using these data.)

Assumptions:

1. It is assumed that N_i observations are taken for each scan.

2. N_i observations taken for each scan, are associated with the existing tracks at same time.

3. Number of targets are not known. therefore, new tracks can be initiated after each scan (after N_i observations are taken).

4. As mentioned previously, it is assumed that only one observation is received for each target at each scan.

5. It is assumed that any observation belonging to at least one of the existing gates, will be used to update one of the established tracks, i.e., they can not initiate a new track. Candidate observations for track initiation are those which belong to none of the existing gates.

6. The probability of detection is not assumed to be unity. In other words, missing detections are assumed to be present.

7. False alarms are included.

In this method, we are going to make use of Bayes' rule which is :

$$P(H_l|Z_k) = \frac{P(Z_k|H_l)P(H_l)}{P(Z_k)} \qquad (3.11)$$

where H_l is identified as a hypothesis concerning the origin of received measurement data. This hypothesis will, in general, contain some groupings of observations into tracks and the identification of other observations to be false alarms. Also Z_k is defined to be the most recently received data set. Then,

$P(H_l)$ = a priori (before the reception of data set Z_k) probability that hypothesis H_l is correct.

$P(Z_k | H_l)$ = probability of receiving Z_k given H_l

$P(H_l | Z_k)$ = a posteriori (after the reception of data set Z_k) probability of H_l

$P(Z_k) = \Sigma P(Z_k | H_l) P(H_l)$ =probability of receiving data set Z_k

In this method, first the probability of a given data association is obtained from $P(H_l)$. Then, after each data set is received, the a posteriori probabilities, $P(H_l | Z_k)$, are computed from $P(H_l)$ and the probability, $P(Z_k | H_l)$ Finally the a posteriori probability, $P(H_l | Z_k)$ becomes a priori probability, $P(H_l)$, when the next set of data is received.

Whenever difficult correlation decisions arise, a final decision is postponed and alternative hypothesis, H_l, are formed and re-evaluated when later data are received. Hypotheses whose association probabilities are high enough are maintained and processing cycle repeats.

In the literature this method is first used for tracking a single target and the problem of track initiation is not considered [19] because dealing with multiple targets and new track initiation were leading to more complex computations, but, computational capacities were not allowing to do that. Fortunately, after the increase in computational capacities, Reid has presented, [20, 21] a structure (denoted Reid's algorithm) which made MHT appear feasible in the case tracking multiple targets together with initiation of new tracks.

MHT is essentially a method using nearest-neighboor correlation technique. In other words, at most, a single observation is assigned to a given track and observations are assigned, at most, to one track. But, it may hold in abeyance the final decision until more data are received

The need to process all observations in updating established tracks caused the use of all-neighboors algorithms which will be explained in the following sections

3.4.2. Single Scan (Sequential), All Neighbors, A Posteriori Algorithms. A Suboptimal Bayesian Approach: The Probabilistic Data Association (PDA). And Extension to The Joint Probabilistic Data Association (JPDA)

Assumptions:

1. There exists only one target.

2. It is assumed that N_i observations are taken for each scan.

3. N_i observations taken for each scan, are associated with the existing track at same time.

3. New track inintiation is not allowed.

4. All observations will be used to update the existing track.

5. The probability of detection is not assumed to be unity. In other words, missing detections are assumed to be present.

6. False alarms may be present.

This algorithm, presented in [7] and [22], incorporates sequentially into a track clusters of measurements by attaching to each track a probability of being correct. Such an approach yields estimates and covariances that account for the measurement origin uncertainty.

It is assumed that there is only one(*) target of interest, whose track has been already initiated, which is observed in the presence of additional measurements that can originate from clutter or other targets. The incorrect measurements are assumed to be i.i.d. random variables with uniform spatial distribution. The state and the measurements of the target of interest are described by Equation (3.4) and Equation (3.5), respectively.

The set of measurements (obtained at scan time k) lying in a neighborhood of the predicted location of the observation from the target is

$$Z_k$$

and the accumulated set of these measurements (up to and including scan time k) is denoted as

$$Z^k$$

The best estimate of the target's state is the conditional mean based upon all observations that with some nonzero probability originated from the target, i.e.,

$$\hat{x}_{k|k} = E\left\{x_k \middle| Z^k\right\} = \sum_{i=1}^{m_k} \beta_{k,i} E\left\{x_k \middle| H_{k,i}, Z^k\right\} = \sum_{i=1}^{m_k} \beta_{k,i} \hat{x}_{k|k,i} \qquad (3.12)$$

where $H_{k,i}$ denotes the event that the i^{th} measurement, $z_{k,i}$ is correct ($i=1,2,.....,m_k$). $H_{k,0}$ is the event that none of them is correct. In the Equation

(*)Number of targets of interest has been extended to arbitrary in [8]

(3.12) $\beta_{k,i}$ is the probability of $H_{k,i}$ being correct given all set of observations up to and including time k, and it is denoted as:

$$\beta_{k,i}=P\{H_{k,i}\,|\,Z^k\} \qquad i=0,1,\ldots,m_k \tag{3.13}$$

Now, we will derive the required probabilities $\beta_{k,i}$

Assume the probability density of the state at time k given past observations to be Gaussian with mean $\hat{x}_{k|k-1}$ and covariance $P_{k|k-1}$, i.e.,

$$p(x_k\,|\,Z^{k-1})=N(\hat{x}_{k|k-1}, P_{k|k-1}) \tag{3.14}$$

Using Bayes' rule, the probabilities $\beta_{k,i}$ can be written as:

$$\begin{aligned}\beta_{k,i}=P\{H_{k,i}\,|\,Z^k\}&=\!=P\{H_{k,,i}\,|\,Z_k,Z^{k-1}\}\\&=c_k^{-1}p(Z_k\,|\,H_{k,i},Z^{k-1})P\{H_{k,i}\,|\,Z^{k-1}\}\end{aligned} \tag{3.15}$$

where c_k is a normalization constant.

For i=0, i.e., when all returns satisfying the gate are incorrect ones, their joint density is:

$$p(Z_k\,|\,H_{k,0},Z^{k-1}) = \prod_{i=1}^{m_k} p(z_{k,i}\,|\,H_{k,0},Z^{k-1}) = V_k^{-m_k} \tag{3.16}$$

where V_k is the volume of the gated region.

In the above equation we assumed that incorrect measurements are uniformly distributed.

The probability of $H_{k,,0}$ based on the past data is:

$$P\{H_{k,,0} \mid Z^{k-1}\} = \alpha_1 + (1-\alpha_1)\alpha_2 \qquad (3.17)$$

where α_1 is the probability that the correct return will not lie in the gated region and α_2 is the probability that the correct return will not be detected.

For i=1,2,.....,m_k the density on the r.h.s. of Equation (3.15) is:

$$p(Z_k \mid H_{k,,i}, Z^{k-1}) = f_k(z_{k,,i})\ V_k^{-m_k+1} \qquad (3.18)$$

where

$$f_k(z_{k,,i}) = (1-\alpha_1)^{-1}\, N(\hat{z}_{k|k-1}, S_k) \qquad (3.19)$$

is a truncated normal distance which is zero outside the gated region.

The probability of $H_{k,,i}$ conditioned upon past data is assumed the same for all i. Then,

$$P\left\{H_{k,i}\middle|Z^{k-1}\right\}=\frac{1-P\left\{H_{k,0}\middle|Z^{k-1}\right\}}{m_k}=\frac{(1-\alpha_1)(1-\alpha_2)}{m_k} \qquad (3.20)$$

Inserting Equation (3.16) thorough Equation (3.19) into Equation (3.15) yields the equations of the PDA (probabilistic data association) method:

$$\beta_{k,i}=f_k(z_{k,,i})[\,b_k+\sum_{i=1}^{m_k} f_k(z_{k,,i})]^{-1} \qquad (3.21)$$

where f_k is defined in Equation (3.19) and

$$b_k=m_k(\alpha_1+\alpha_2-\alpha_1\alpha_2)[(1-\alpha_1)(1-\alpha_2)V_k]^{-1} \qquad (3.22)$$

Considering the Gaussian assumption and referring to the Section .2.2.2,.we can obtain $\hat{x}_{k|k,i}$ of Equation (3.12) by using Kalman filter:

$$\hat{x}_{k|k,i}=\hat{x}(k\,|\,k\text{-}1)+W_k v_{k,i} \qquad (3.23)$$

$$\hat{x}_{k|k}=\sum_{i=0}^{m_k}\beta_{k,i}\hat{x}_{k|k,i}=\hat{x}_{k|k\text{-}1}+W_k v_k \qquad (3.24)$$

where

$$v_k = \sum_{i=1}^{m_k} \beta_{k,i} v_{k,i} \tag{3.25}$$

is the weighted residual which uses all measurements satisfying the gate, and

$$W_k = P_{k|k-1} H_k' S_k^{-1} \tag{3.26}$$

is the weighting matrix

The covariance associated with the estimate (Equation (3.24)) is:

$$P_{k|k} = \beta_{k,i} P_{k|k-1} + (1-\beta_{k,0}) P_{k|k}^{*} + W_k \left[\sum_{i=1}^{m_k} \beta_{k,i} v_{k,i} v_{k,i}' v_k v_k' \right] W_k' \tag{3.27}$$

where $P_{k|k}^{*}$ is the covariance of the update if we have only one return. The last term above is a positive semidefinite matrix which shows the effect of the incorrect measurements by increasing the covariance of the update. The algebraic details related to Equation (3.27) can be found in [22].

Equation (3.24) through Equation (3.27) constitute the probabilistic data association filter (PDAF)

An important feature of the PDAF is that its computational requirements are the same as the standard filters when only one return falls in the gate and increase only slightly when the need of processing multiple returns arises.

As mentioned above, this algorithm deals with tracking only one single target. For tracking multiple targets PDA is extended to JPDA (Joint Probabilistic Data Association) The JPDA method is identical to the PDA except that the association probabilities are computed using all observations and all tracks. In this thesis, equations related to the JPDA method (which is given in [8]) will not given. But a more general and an optimal solution to multitarget tracking will be explain in the following section.

3.4.3. Multiple-scans (Batch), All Neighbors, A Posteriori Algorithm

An Optimal Bayesian Approach

As in the previous sections, we are going to assume that there is only one target and the number of observations with uncertain origin is arbitrary. The target and measurements models are the same. The main difference between this method and the one presented in the previous section is the following: the decomposition of the state estimate (according to the total probability theorem) is done in terms of all combinations of measurements from initial to present time rather than only in terms of the latest measurements This is equivalent to splitting the track from the initial time to the current and recombining this into a single estimate.

While memory and computational requirements are increasing with time, this approach is of interest because it requires no approximations. Yet, this method can not be considered as a feasible method. Instead, suboptimal versions that are intermediate between this and the one from the previous section can be obtained.

The computational details of this algorithm can be found in [9] and [23]. But we are not going to analyze this method in this thesis

3.5. CONCLUDING REMARKS

The MHT and PDA methods originated around the same time, and development has proceeded in parallel. The PDA and the later JPDA are, however, basically a special case of MHT. The advantage of the PDA (or JPDA) is that it is a relatively simple recursive method which does not require the storage of past observation data nor multiple hypotheses.

An apparent disadvantage associated with the PDA (or JPDA) is the lack of an explicit mechanism for track initiation. So, it might be suitable to employ some other algorithms for track initiation and then to use PDA (or JPDA) for track maintenance. Similarly, there is no well-defined technique for track deletion associated with PDA (or JPDA), but as with track initiation, acceptable algorithms should be used for that purpose.

The most important factor in the choice of MHT versus JPDA methods is probably the false alarm (or false target) density. For high false target densities, MHT is probably not feasible, so JPDA is favored. However, MHT becomes feasible for the lower false target densities.

The following general guideline to the choice of data association methods is offered. For sparse environment (where the probability of multiple targets within the same gate is expected to be very low), the standard nearest-neighbor method (suboptimal or optimal solutions to assignment matrix) can be used. If some difficult decision situations occur, one may switch to branching algorithm

method (suboptimal or optimal solutions to assignment matrix) can be used. If some difficult decision situations occur, one may switch to branching algorithm (track splitting) and the final decision may be deferred until more data are received. Alternatively, if the number of targets of interest is unity or perfect gating may be realized for multiple targets (i.e., gates of different tracks don't intersect) the PDA method can be used to reach a higher performance because it incorporates all neighboring measurements and accounts for uncertainty in the origin of the measurements. Switching to the track splitting algorithm may again be required in some correlation conflict situations.

As the target or the false return density increases, application of MHT techniques becomes most appropriate. Ultimately, as the density further increases, JPDA becomes an attractive alternative because then, MHT requires extremly high computational capacities and time so that the use of a sequential method is inevitable.

CHAPTER 4

A NEW N-SCAN BACK DATA ASSOCIATION ALGORITHM

In Chapter 3, existing tracking methods were surveyed and methods using nearest-neighbor approaches were classified into :

- Standard nearest-neighbor method [2]
- Track splitting method [9]
- Maximum Likelihood Method [9]
- Multiple Hypothesis Testing (MHT) Method [2][9]

These methods are essentially sequential (recursive) methods, but only in multiple hypothesis testing method, difficult decisions may be deferred until more data are received (i.e., MHT method can be viewed as an N-scan back algorithm). 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 an N-scan back, nearest-neighbor algorithm.

In this algorithm, possible paths of each target are represented by trees. Expected measurements corresponding to the nodes of these trees are estimated. The distances of measurements to these estimated measurements are calculated. The sums of these distances for nodes each taken from different levels, are found. The measurements are assigned to the target which yields the minimum sum of distances.

The proposed data association algorithm assumes followings:

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_x - dimensional state vector of the target at next scan time "k+1" , assuming we are at scan time "k" .

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

$F(k)$ and $H(k)$ are $n_x X n_x$ matrices which are known and possibly time varying.

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

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

2. The number, M, 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" possible values where L is a given finite positive integer. This approximation is explained in Appendix C in detail and discrete random variables approximating the Gaussian random variable with zero mean and unit variance is given in Table C.1.)
4. Each component of process noise v(k), is a discrete random variable with "L_2" possible values.
5. There are no false alarms. In other words, If there are M targets to be tracked, the number of observations taken at each scan will also be "M".
6. The decision for observation to target pairings will be made after each N-scan time intervals.

For the case of a single target there does not exist a data association problem. Because we know that all observations belong to this target since no false alarms are allowed.

In the case of multiple targets, however, we are not sure which observation belong to which target. In order to solve this problem, we proceed as follows:

Consider a target. The number, B, of possible initial states for this target will be $L_1^{n_x}$. Consider one of these states. Using Equation (4.1), compute possible states of the target at next scan. The number, D, of these possible states will be $L_2^{n_x}$. In this way, continue to computing subsequent possible state values for following scan times. Perform these computations for other possible initial states and for all other targets. At the end, for each target, you will be end up with a tree structure, Figure 4.1, which is a tree representation of possible paths of the target for N scan time periods. In Figure 4.1, each level correspond to a scan time period and X_{ij} denotes a node of the tree such that X_{ij} is the j^{th} possible state location at level i. (Since we develop an N-scan back algorithm maximum value of i may be N.)

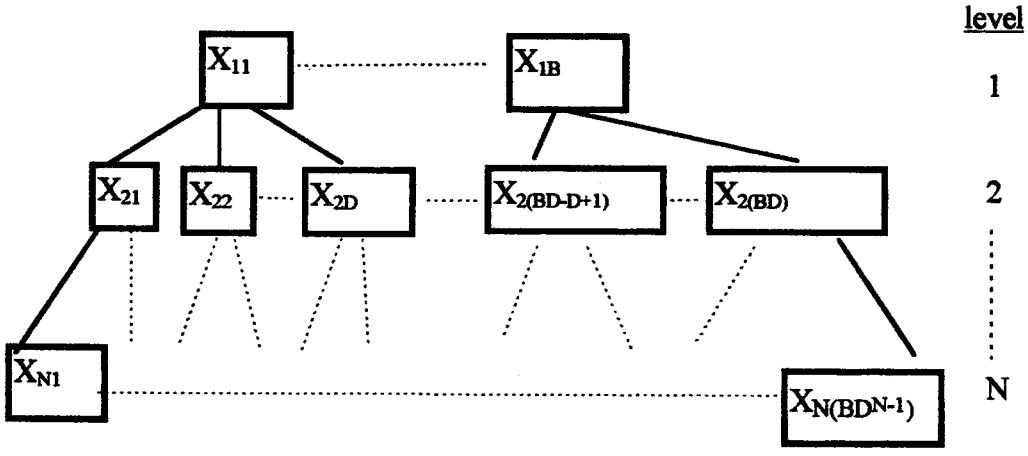


Figure 4.1: A Tree Representation of Possible Paths of A Target for N Scan Time Periods.

Similarly, another tree structure will be obtained for estimated observation values for M targets and for N scan time periods, using Equation (4.2) and possible states of corresponding targets obtained previously. 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. (Since we develop an N-scan back algorithm maximum value of i may be N.)

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

Each level correspond to a single scanning period.

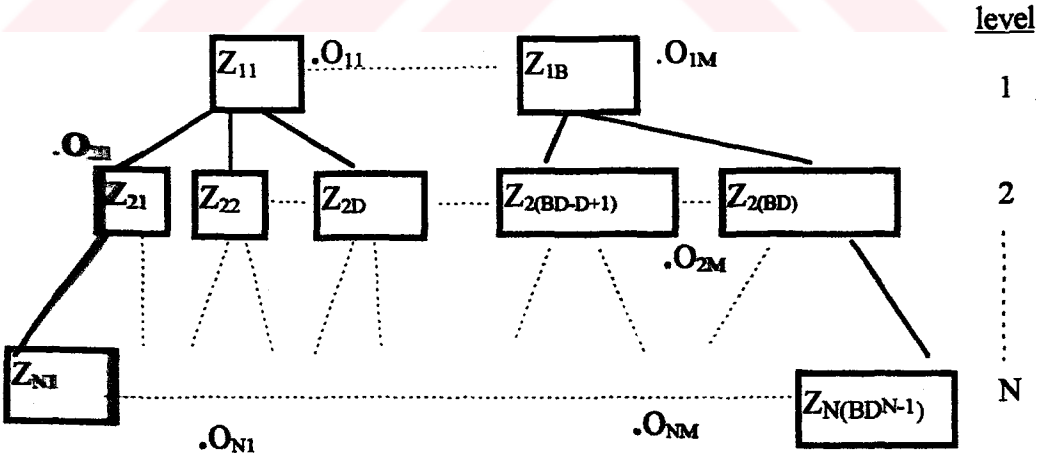


Figure 4.2: A Tree Representation of Estimated Observations of A Target for N Scan Time Periods.

Assuming that the tree structure shown in Figure 4.2. is available for every target and measured observation values taken from the sensor (radar or any other sensors) for the first N scan time periods (for the first N levels) are ready, the problem which is to be solved is to decide which observation set belongs to which target. For the solution of this problem, first, we will suggest to compute any possible distance values between measured observation locations and estimated observation locations. Then, the sums of these distances for nodes each taken from different levels, are found. The measurements are assigned to the target which yields the minimum sum of distances.

For individual distance measurements we will use following general distance equation:

$$d^2_{imj} = y'_{imj} P_{ij} y_{imj} \qquad (4.3)$$

where

- d^2_{imj} is a normalized squared distance value between m^{th} measured observation location vector O_{im}, and j^{th} estimated observation location vector Z_{ij}, at level i.
- y_{imj} is the residual vector between m^{th} measured observation location vector O_{im}, and j^{th} estimated observation location vector Z_{ij}, at level i. In other words,

$$y_{imj} = O_{im} - Z_{ij} \qquad i=1,2,\ldots\ldots,N \qquad \text{(levels)}$$

$$m=1,2,\ldots\ldots,M$$

$$j=1,2,\ldots\ldots ML'$$

and

$$O_{im}=\begin{bmatrix} O_{im_x} \\ O_{im_y} \\ O_{im_z} \end{bmatrix} \qquad Z_{ij}=\begin{bmatrix} Z_{ij_x} \\ Z_{ij_y} \\ Z_{ij_z} \end{bmatrix}$$

- $$P_{ij}=\begin{bmatrix} \frac{1}{\alpha} & 0 & 0 \\ 0 & \frac{1}{\beta} & \\ 0 & 0 & \frac{1}{\gamma} \end{bmatrix}$$

where,

$$\alpha = Prob\left(Z_{ij_x}\right)$$

$$\beta = Prob\left(Z_{ij_y}\right)$$

$$\gamma = Prob\left(Z_{ij_z}\right)$$

These probabilities can be found in a tabular form in Table C.1. The inverses of these probabilities, are included in P_{ij}, 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.

For the data association problem described above, we propose following algorithm. Computer implementation of this algorithm is realized in C programming language. Simulation results of this implementation is given in Section 4.3.

4.1. ALGORITHM (*)

- STEP 1: Construct the tree structure for possible paths (possible states) related with one of the targets using Equation (4.1).

- STEP 2: Obtain estimated observations tree structure related with tree structure constructed at STEP 1 using Equation (4.2).

- STEP 3: Form an observation set by taking one observation from each "level". (e.g., for M=3, L_1=3, L_2=3, N=3, using estimated observations tree structure shown in Figure 4.2, an example of observation set will be $\{O_{11}, O_{21}, O_{31}\}$.

- STEP 4: Compute and record all possible distance values, d^2_{imj}. related with any measured observation included in the observation set formed previously.

(*)In this algorithm, in addition to the assumptions made at the beginning of this chapter, it is assumed that radar measurements (real observation values) are ready. In the implementation of the algorithm a random number generator is used to cope with this problem.

- STEP 5: Form a special distance set by taking one pairing (i.e. one d^2_{imj} term) from each level. (e.g., in Figure 4.2, for observation set $\{O_{11}, O_{21}, O_{31}\}$, an example of distance set will be $\{d^2_{111}, d^2_{211} d^2_{311}\}$). (Some distance sets are not allowed. For example, in Figure 4.2., if set of observations is assumed to be $\{O_{11}, O_{21}, O_{31}\}$, and if, at level 1, d^2_{111} is taken as the first item of special set formed, one can not accept $d^2_{21(28)}$ as a second item. In other words, only one of d^2_{21j}, (j=1,2,.....27), terms must be accepted as a second item.)

- STEP 6: Sum all d^2_{imj} terms in the previous distance set and record this summation..

- STEP 7: Replace the distance set at STEP 5 with a different possible one. IF there doesn't exist any different distance set GO TO STEP 8, OTHERWISE, GO TO STEP 6.

- STEP 8: Replace the observation set at STEP 3 with a different observation set. IF there doesn't exist any different observation set GO TO STEP 9. OTHERWISE, GO TO STEP 4.

- STEP 9: Identify the set of observations yielding minimum summation. This set belongs to the target under consideration.

- STEP 10: Delete observations which assignments are made until now from the original observation set. IF all observations are used and assigned, except last M observations, assign those last M observations to the target which is not analyzed yet., and GO TO STEP 11. OTHERWISE , replace the tree structure with one of the remaining targets' which is not analyzed yet and GO TO STEP 3.

- STEP 11: Now, observations belonging to each target are known (i.e., data associations are made) and remaining functions needed in an MTT system (such as filtering) may be realized.

4.2. COMPUTER IMPLEMENTATION OF DATA ASSOCIATION ALGORITHM USING C

In the implementation of the algorithm, creation of measured (real) observation values (e.g., O_{11}, O_{21},...., O_{NM}) is needed and, for that purpose, a random number generator is used to obtain a zero mean and unity variance Gaussian random variable, x. In order to obtain a Gaussian random variable, y, with mean being equal to m, and variance σ, $y=\sigma x+m$ transformation is made (proof is given at the end of Appendix C.)

In the implementation of data association algorithm, a tree structure creation method is used to obtain required tree structure showing estimated observation locations. Though implementation can 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 3 because of memory limitations. This limitation will be explained by a simple example showing a memory overflow situation where L_1=8, L_2=8 and N=3 :

A Memory Overflow Example

Assume we are at scan time "k", z(k) is a 3x1 random vector, and elements of z(k) correspond for x,y,z components of an estimated observation location. If L=8, each component of z(k+1), estimated observation location at time "k+1", may have 8 possible different locations yielding totally, 8x8x8 (=512) estimated observation locations at scan time "k+1" considering all possibilities related with all components. If we extend this calculation for 3 levels (i.e., assume N=3), total number, T, of nodes in a tree structure like that of Figure 4.2., will be:(*)

$T=1+512+512^2 = 262647$

At each node, we have to store x,y,z component values, each requiring 4 byte memory location. In addition, at any node, pointers which are special variable types holding memory addresses of 512 possible values of estimated observation locations must be present and number of these pointers, each

(*)In this calculation, initial value of z(k) for the first level is assumed to be known for the sake of simplicity.

Finally, for 262645 nodes, total memory requirement is :

262645x2 KB=525294 KB ≅ 525 MB

which is a memory overflow situation for our working conditions.

4.3. SIMULATION RESULTS AND PERFORMANCE COMPARISON WITH "TRACK SPLITTING APPROACH"

Example 1 (A Correct Data Association):

For this example,

- Number of targets, M, is assumed to be 3.
- Number of possible discrete values, L, of a Gaussian random variable approximating an absolutely continuous Gaussian random variable is assumed to be 3. (i.e., L_2=3)
- Number of scan time periods; N, between two consecutive decisions is assumed to be 3.
- Initial possible state values for targets are assumed to be:

  x axis component of initial state for target 0 = 1

  y axis component of initial state for target 0 = 1

  z axis component of initial state for target 0 = 1

  x axis component of initial state for target 1 = 5

  y axis component of initial state for target 1 = 5

  z axis component of initial state for target 1 = 5

  x axis component of initial state for target 2 = 10

  y axis component of initial state for target 2 = 10

z axis component of initial state for target 1 = 5

x axis component of initial state for target 2 = 10

y axis component of initial state for target 2 = 10

z axis component of initial state for target 2 = 10

- F matrix in Equation (4.1) is given as:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

- And finally, in generating measured observation values by random number generator, Gaussian process noise, v(k) in Equation (4.1), and Gaussian measurement noise, w(k) in Equation (4.2), are both assumed to be random vectors with zero-mean and unit covariance. But the program is capable of dealing with any random variable (i.e., mean≠0 and variance≠1). And H matrix in Equation (4.2) is assumed to be an identity matrix.

In simulation, real observation values are needed to be present and, as mentioned above, a random number generator is used for that purpose. Results are given in Table 4.1 where

- observation[LEVEL : i][j].x is x axis component of generated observation location of target j at level i
- observation[LEVEL : i][j].y is y axis component of generated observation location of target j at level i
- observation[LEVEL : i][j].z is z axis component of generated observation location of target j at level i

At the end, program made a classification between observation values and we saw that this classification is completely coherent with that of generated observation values given in Table 4.1. This classification is given in Table 4.2 where x, y, z components of assigned observations are tabulated for different levels and targets.

Example 2 (An Incorrect Data Association):

In this example, same assumptions used in Example 1 are considered. Different observations are generated by changing a seed value in the random number generator used in the program. It is seen that classification made by the program for this simulation is not coherent with that of generated observation values. In Table 4.3, generated observations for this example are tabulated. In Table 4.4, data associations made by the program are presented. Notations used in Tables 4.3. and 4.4 are similar to those of Tables 4.1 and 4.2 respectively.

Table 4.1: Generated Observation Values For The First Simulation

observation[LEVEL : 0][0].x : -0.656862
observation[LEVEL : 0][0].y : -0.147527
observation[LEVEL 0][0].z : 0.616303
observation[LEVEL : 1][0].x : 3.252998
observation[LEVEL : 1][0].y : 1.630326
observation[LEVEL : 1][0].z : 0.470085
observation[LEVEL : 2][0].x : 6.038915
observation[LEVEL : 2][0].y : 5.232207
observation[LEVEL : 2][0].z : 4.765645
observation[LEVEL : 0][1].x : 5.487755
observation[LEVEL : 0][1].y : 5.521997
observation[LEVEL : 0][1].z : 5.927278
observation[LEVEL : 1][1].x : 10.910786
observation[LEVEL : 1][1].y : 10.932306
observation[LEVEL : 1][1].z : 9.188408
observation[LEVEL : 2][1].x : 23.633509
observation[LEVEL : 2][1].y : 20.032001
observation[LEVEL : 2][1].z : 18.833771
observation[LEVEL : 0][2].x : 10.334901
observation[LEVEL : 0][2].y : 10.542736
observation[LEVEL : 0][2].z : 8.129009
observation[LEVEL : 1][2].x : 21.033613
observation[LEVEL : 1][2].y : 20.184494
observation[LEVEL : 1][2].z : 21.262737
observation[LEVEL : 2][2].x : 38.672684
observation[LEVEL : 2][2].y : 43.836178
observation[LEVEL : 2][2].z : 39.831375

Table 4.2: Correct Associations Made By The Program For The First Simulation

==OBSERVATIONS BELONGING TO TARGET 0==			level
X: -0.656862	Y: -0.147527	Z: 0.616303	0
X: 3.252998	Y: 1.630326	Z: 0.470085	1
X: 6.038915	Y: 5.232207	Z: 4.765645	2

==OBSERVATIONS BELONGING TO TARGET 1==			level
X: 5.487755	Y: 5.521997	Z: 5.927278	0
X: 10.910786	Y: 10.932306	Z: 9.188408	1
X: 23.633509	Y: 20.032001	Z: 18.833771	2

==OBSERVATIONS BELONGING TO TARGET 2==			level
X: 10.334901	Y: 10.542736	Z 8.129009	0
X: 21.033613	Y: 20.184494	Z 21.262737	1
X: 38.672684	Y: 43.836178	Z 39.831375	2

Table 4.3: Generated Observation Values For The Second Simulation

observation[LEVEL : 0][0].x : 1.049873
observation[LEVEL : 0][0].y : 0.906048
observation[LEVEL : 0][0].z : 2.000624
observation[LEVEL : 1][0].x : 3.404015
observation[LEVEL : 1][0].y : 2.515595
observation[LEVEL : 1][0].z : 1.184208
observation[LEVEL : 2][0].x : 4.322154
observation[LEVEL : 2][0].y : 2.937063
observation[LEVEL : 2][0].z : 1.149853
observation[LEVEL : 0][1].x : 6.728763
observation[LEVEL : 0][1].y : 6.139554
observation[LEVEL : 0][1].z : 3.947072
observation[LEVEL : 1][1].x : 10.134068
observation[LEVEL : 1][1].y : 8.490119
observation[LEVEL : 1][1].z : 9.968396
observation[LEVEL : 2][1].x : 19.159565
observation[LEVEL : 2][1].y : 14.002287
observation[LEVEL : 2][1].z : 17.713900
observation[LEVEL : 0][2].x : 10.132525
observation[LEVEL : 0][2].y : 10.938110
observation[LEVEL : 0][2].z : 12.051403
observation[LEVEL : 1][2].x : 12.051403
observation[LEVEL : 1][2].y : 18.950457
observation[LEVEL : 1][2].z : 17.561136
observation[LEVEL : 2][2].x : 41.731201
observation[LEVEL : 2][2].y : 35.130291
observation[LEVEL : 2][2].z : 39.458256

Table 4.4: Incorrect Associations Made By The Program For The Second Simulation

==OBSERVATIONS BELONGING TO TARGET 0==			level
X: 1.049873	Y: 0.906048	Z: 2.000624	0
X: 3.404015	Y: 2.515595	Z: 1.184208	1
X: 4.322154	Y: 2.937063	Z: 1.149853	2
==OBSERVATIONS BELONGING TO TARGET 1==			level
X: 6.728763	Y: 6.139554	Z: 3.947072	0
X: 12.051403	Y: 18.950457	Z: 17.561136	1
X: 41.731201	Y: 35.130291	Z: 39.458256	2
==OBSERVATIONS BELONGING TO TARGET 2==			level
X: 10.132525	Y: 10.938110	Z: 12.051403	0
X: 10.134068	Y: 8.490119	Z: 9.968396	1
X: 19.159565	Y: 14.002287	Z: 17.713900	2

In following subsections, effect of different parameters to the performance of the new data association algorithm is analyzed by examples and the performance of the new algorithm is compared with "Track Splitting Approach"[*]. In these examples, maximum number of targets is assumed to be 6; maximum number, L_2 ,of discrete values approximating any component of v(k) or w(k) which are Gaussian noise vectors in Equations (4.1) and (4.2) respectively, is assumed to be 3; and maximum number, N, of levels (i.e., number of scans between any two consecutive decisions) is assumed to be 3. F, H matrices, mean and covariance of v(k) and w(k) and initial state vectors (startpoints) related with each target are given below

Target 1:

$$F=\begin{bmatrix} 2 & 1.2 & 0.5 \\ 1 & 1.9 & 1 \\ 1.5 & 2 & 2 \end{bmatrix} \qquad H=\begin{bmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 1 & 3 & 2 \end{bmatrix}$$

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

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

[*] Implementation of "Track Splitting Approach" is based on the explanation given in section 3.3.1.

$$\text{initial state}=\begin{bmatrix}1\\0\\0\end{bmatrix}$$

Target 2:

$$F=\begin{bmatrix}2 & 1 & 0.5\\ 1.1 & 1.5 & 1\\ 1.5 & 3 & 2\end{bmatrix} \qquad H=\begin{bmatrix}2 & 4 & 5\\ 1 & 1 & 1\\ 3 & 2 & 1\end{bmatrix}$$

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

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

$$\text{initial state}=\begin{bmatrix}-1\\0\\0\end{bmatrix}$$

Target 3:

$$F=\begin{bmatrix}1.8 & 1 & 0.8\\ 1 & 1.9 & 1\\ 1.5 & 2 & 2\end{bmatrix} \qquad H=\begin{bmatrix}1 & 3 & 4\\ 2 & 1 & 2\\ 2 & 2 & 1\end{bmatrix}$$

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

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

initial state= $\begin{bmatrix} 0 \\ \sqrt{3} \\ 0 \end{bmatrix}$

Target 4:

$F=\begin{bmatrix} 1.75 & 2.5 & 3 \\ 1.5 & 2.25 & 4.5 \\ 1 & 2 & 3.5 \end{bmatrix}$ $H=\begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 1 \\ 5 & 1 & 1 \end{bmatrix}$

mean of v(k)= $\begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix}$ **Covariance of v(k)=** $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$

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

$$\text{initial state}=\begin{bmatrix} 0 \\ 1/\sqrt{3} \\ 2\sqrt{2/3} \end{bmatrix}$$

Target 5:

$$F=\begin{bmatrix} 2 & 4 & 1 \\ 1 & 2 & 1 \\ 5 & 1 & 3 \end{bmatrix} \qquad H=\begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 1 \\ 2 & 2 & 1 \end{bmatrix}$$

$$\text{mean of } v(k)=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{Covariance of } v(k)=\begin{bmatrix} 4 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 6.25 \end{bmatrix}$$

$$\text{mean of } w(k)=\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \qquad \text{Covariance of } w(k)=\begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\text{initial state}=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Target 6:

$$F=\begin{bmatrix}2 & 3 & 1\\ 2 & 2 & 1\\ 5 & 2 & 3\end{bmatrix} \qquad H=\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 1\\ 2 & 3 & 1\end{bmatrix}$$

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

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

$$\text{initial state}=\begin{bmatrix}0\\ 1/\sqrt{3}\\ -2\sqrt{2/3}\end{bmatrix}$$

Effect of the increase in number of targets:

In this example, it is aimed to show the effect of the increase in number of targets by considering targets which parameters are given above. In addition, number, N, of levels (i.e., number of scan time periods between two consecutive decisions) is 3; and number, L_2 ,of discrete values approximating any component of v(k) in Equation (4.1) is 3.

To show the effect of the incerase in number of targets, probability of error, Pe, is computed as number of targets increase from 2 to 6. For that

purpose, 30 simulations are performed for each case and the number, E, of incorrect data associations are determined. Probability of error, Pe, is computed as

$$Pe=\frac{E}{30}$$

Results of this example for the new algorithm and for "Track Splitting Approach" are tabulated in Table 4.5 and Table 4.6 and probability of error, Pe, versus number of targets is given in Figure 4.3 and Figure 4.4.

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

In this example, it is aimed 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 first 3 targets (Target 1, Target2, Target 3) which parameters are given previously. In addition, number, N, of levels (i.e., number of scan time periods between two consecutive decisions) is 3.

To show the effect of the incerase 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 1 to 3. For that purpose, 30 simulations are performed for each case and the number, E, of incorrect data associations are determined. Probability of error, Pe, is computed as:

$$Pe=\frac{E}{30}$$

Results of this example are tabulated in Table 4.7, and probability of error, Pe, versus L_2 is given in Figure 4.5.

Effect of the increase in distance between initial states of targets:

In this example, it is aimed to show the effect of the increase in distance between initial states of targets by considering first 3 targets (Target 1, Target2, Target 3) which parameters are given previously. In addition, number, N, of levels (i.e., number of scan time periods between two consecutive decisions) is 3; and number, L_2 ,of discrete values approximating any component of v(k) in Equation (4.1) is 3.

To show the effect of the increase in distance between initial states of targets, probability of error, Pe, is computed for five different choices of initial states. Where,

Case 1:

Distance between initial states of each target is 0.4

Case 2:

Distance between initial states of each target is 1

Case 3:

Distance between initial states of each target is 1.6

Case 4:

Distance between initial states of each target is 1.8

Case 5:

Distance between initial states of each target is 2

Case 6:

Distance between initial states of each target is 2.4

Case 7:

Distance between initial states of each target is 2.8

Case 8:

Distance between initial states of each target is 4

To determine probability of error, Pe, 30 simulations are performed for each above case and the number, E, of incorrect data associations are determined. Probability of error, Pe, is computed as:

$$Pe = \frac{E}{30}$$

Results of this example for the new algorithm and for "Track Splitting Approach" are tabulated in Table 4.8, and probability of error, Pe, versus squared distance is given in Figure 4.6

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

In this example, it is aimed 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, Target2, Target 3) which parameters are given previously. In addition, number, N, of levels (i.e., number of scan time periods between two consecutive decisions) is 3; and number, L_2 ,of discrete values approximating any component of v(k) in Equation (4.1) is 3.

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 tree different situations: First variance of x component of v(k) is changed by keeping y and z components constants. Then, variance of y component of v(k) is changed by keeping x and z components constants. And then, variance of z component of v(k) is changed by keeping x and y components constants. For each situation, variance values are assumed to be 1, 2, 5, 10, 100, 250, 400, 500, 1000, $5X10^6$, $5X10^9$, $5X10^{12}$. To determine probability of error, Pe, 30 simulations are performed for each alternative and the number, E, of incorrect data associations are determined. Probability of error, Pe, is computed as:

$$Pe=\frac{E}{30}$$

Results of this example for the new algorithm and for "Track Splitting Approach" are tabulated in Tables 4.9, 4.10, 4.11, 4.12, 4.13, 4.14 and probability of error, Pe, versus variances are given in Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12.

Table 4.5: Effect of the Increase in Number of Targets (when distance between initial states of targets are 2)

Number of Targets	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach)
2	2	0.07	21	0.7
3	15	0.50	30	1.00
4	24	0.80	30	1.00
5	28	0.93	30	1.00
6	30	1.00	30	1.00

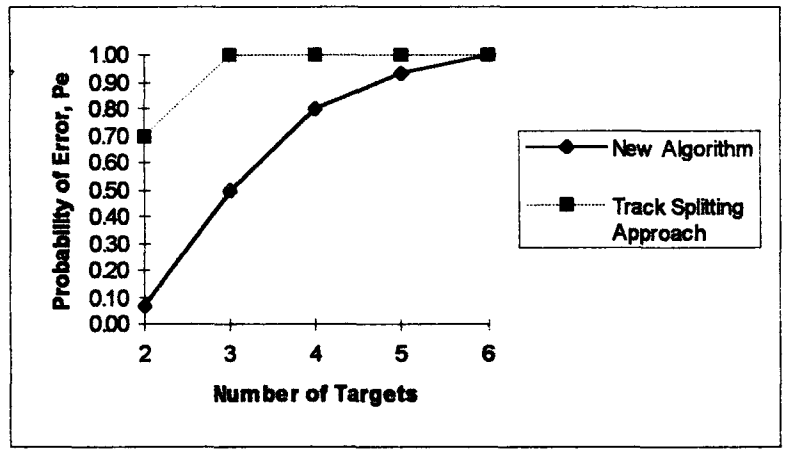


Figure 4.3: Pe versus Number of Targets

Table 4.6: Effect of the Increase in Number of Targets (when distance between initial states of targets are 500)

Number of Targets	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach)
2	0	0.00	1	0.03
3	0	0.00	2	0.07
4	0	0.00	15	0.50
5	0	0.00	23	0.77
6	0	0.00	25	0.83

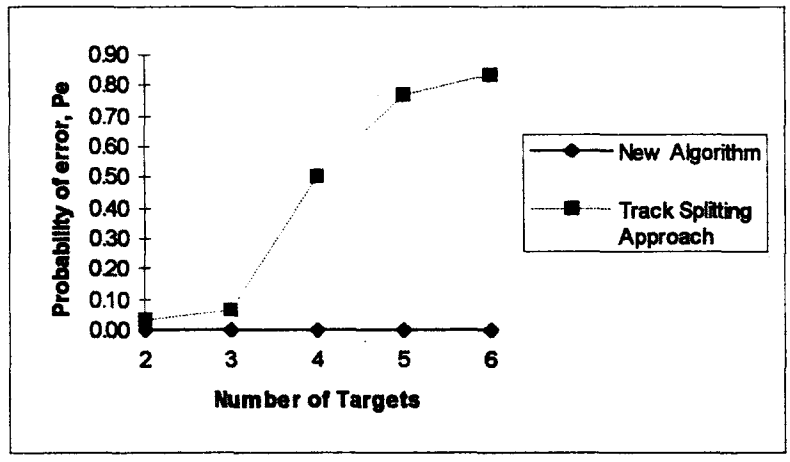


Figure 4.4: Pe versus Number of Targets

Table 4.7: Effect of the Increase in Number, L_2, of Discrete Values Approximating v(k) in Equation (4.1)

L2	E	Pe (=E/30)
1	20	0.67
2	19	0.63
3	15	0.50

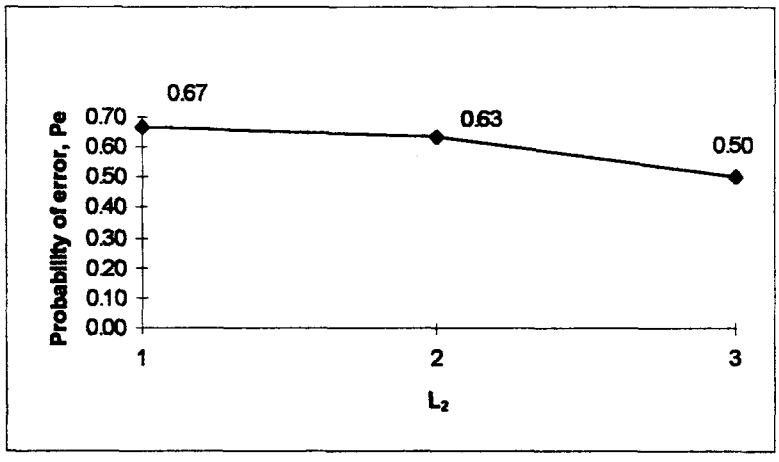


Figure 4.5: Pe versus L_2

Table 4.8: Effect of the Increase in Distance Between Initial States of Targets

Distance	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach)
0.4	30	1.00	30	1.00
1	30	1.00	30	1.00
1.6	22	0.73	30	1.00
1.8	20	0.67	30	1.00
2	15	0.50	30	1.00
2.4	10	0.33	30	1.00
2.8	5	0.17	30	1.00
4	0	0.00	30	1.00
10	0	0.00	30	1.00
100	0	0.00	25	0.83
200	0	0.00	18	0.60
300	0	0.00	12	0.40
500	0	0.00	2	0.07
1000	0	0.00	0	0.00

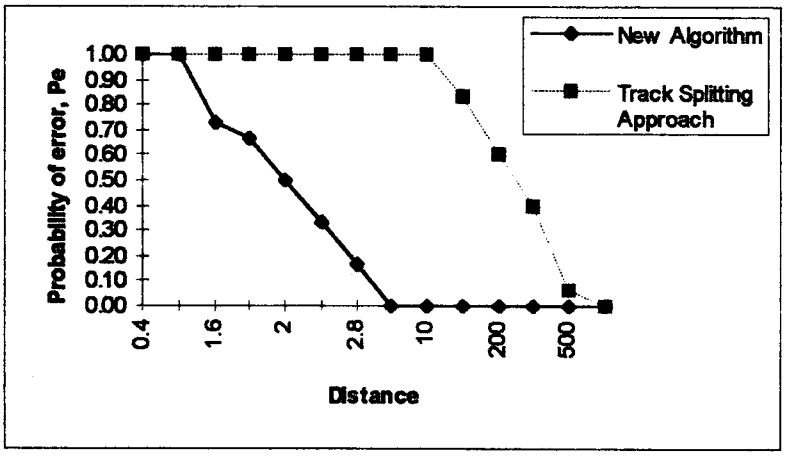


Figure 4.6: Pe versus Distance

Table 4.9: 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 initial states of targets are 2.8)

Variance of x	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	4	0.13	30	1
2	5	0.17	30	1
5	7	0.23	30	1
10	14	0.47	30	1
100	27	0.90	30	1
250	29	0.97	30	1
400	29	0.97	30	1
500	29	0.97	30	1
1000	29	0.97	30	1
$5X10^{6}$	29	0.97	30	1
$5X10^{9}$	30	1.00	30	1
$5X10^{12}$	30	1.00	30	1

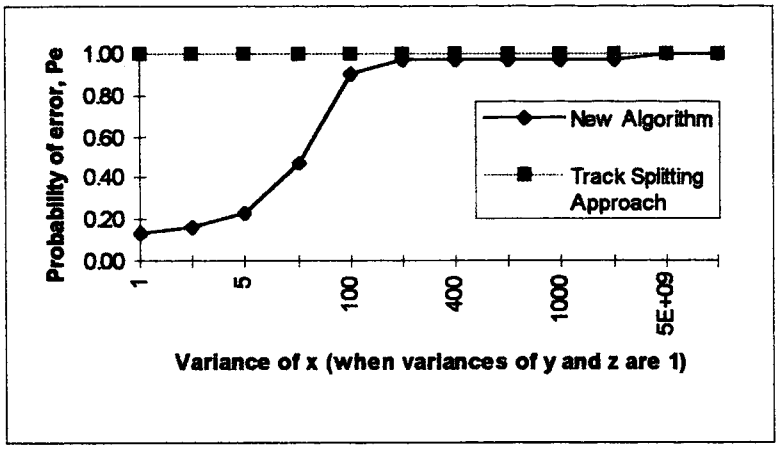


Figure 4.7: Pe versus Variance of x (when distance between initial states of targets are 2.8)

Table 4.10: 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 initial states of targets are 300)

Variance of x	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	0	0.00	2	0.07
2	0	0.00	3	0.10
5	0	0.00	6	0.20
10	0	0.00	13	0.43
20	0	0.00	20	0.67
50	0	0.00	28	0.93
100	0	0.00	29	0.97
250	0	0.00	30	1.00
400	5	0.17	30	1.00
500	9	0.30	30	1.00
1000	18	0.60	30	1.00
$5X10^6$	29	0.97	30	1.00
$5X10^9$	29	0.97	30	1.00
$5X10^{12}$	29	0.97	30	1.00

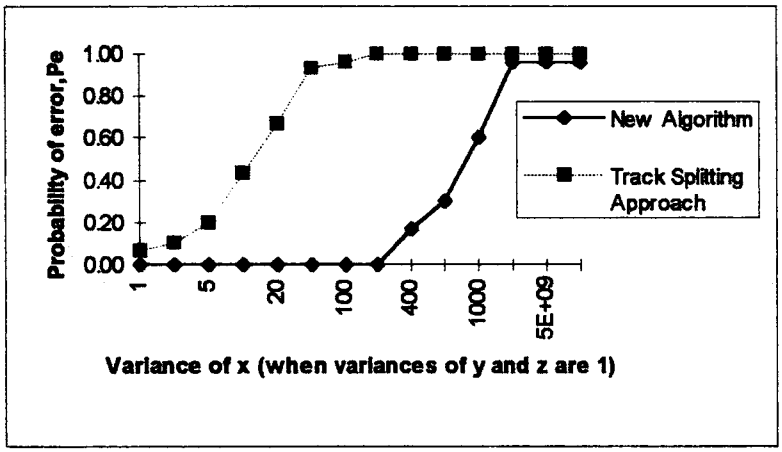


Figure 4.8: Pe versus Variance of x (when distance between initial states of targets are 300)

Table 4.11: 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 initial states of targets are 2.8)

Variance of y	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	4	0.13	30	1
2	5	0.17	30	1
5	10	0.33	30	1
10	20	0.67	30	1
100	28	0.93	30	1
250	28	0.93	30	1
400	29	0.97	30	1
500	29	0.97	30	1
1000	29	0.97	30	1
$5X10^6$	29	0.97	30	1
$5X10^9$	30	1.00	30	1
$5X10^{12}$	30	1.00	30	1

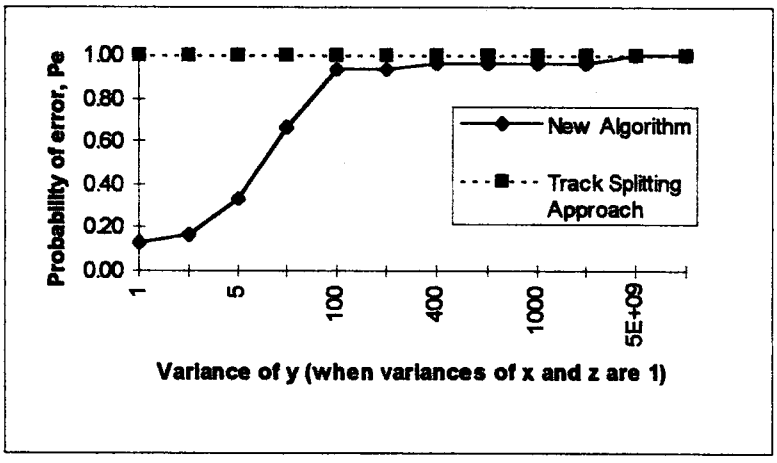


Figure 4.9: Pe versus Variance of y (when distance between initial states of targets are 2.8)

Table 4.12: 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 initial states of targets are 300)

Variance of y	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	0	0.00	2	0.07
2	0	0.00	9	0.30
5	0	0.00	27	0.90
10	0	0.00	29	0.97
20	0	0.00	30	1.00
50	0	0.00	30	1.00
100	0	0.00	30	1.00
250	3	0.10	30	1.00
400	11	0.37	30	1.00
500	13	0.43	30	1.00
1000	21	0.70	30	1.00
$5X10^6$	28	0.93	30	1.00
$5X10^9$	28	0.93	30	1.00
$5X10^{12}$	28	0.93	30	1.00

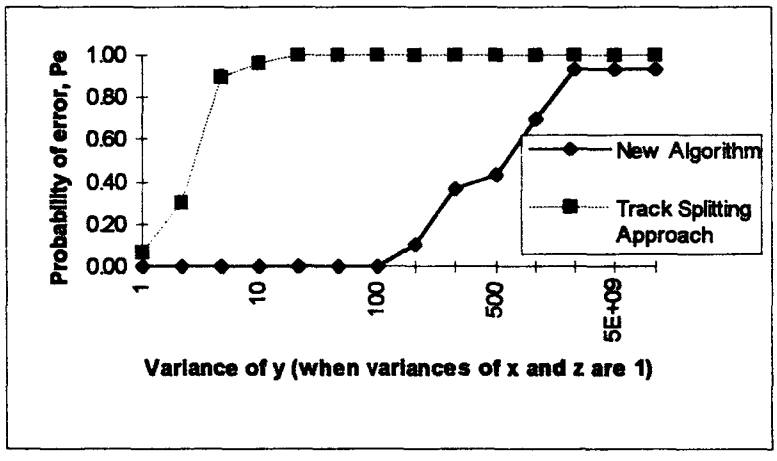


Figure 4.10: Pe versus Variance of y (when distance between initial states of targets are 300)

Table 4.13: 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 initial tates of targets are 2.8)

Variance of z	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	4	0.13	30	1
2	5	0.17	30	1
5	12	0.40	30	1
10	18	0.60	30	1
100	27	0.90	30	1
250	28	0.93	30	1
400	28	0.93	30	1
500	28	0.93	30	1
1000	28	0.93	30	1
$5X10^6$	28	0.93	30	1
$5X10^9$	30	1.00	30	1
$5X10^{12}$	30	1.00	30	1

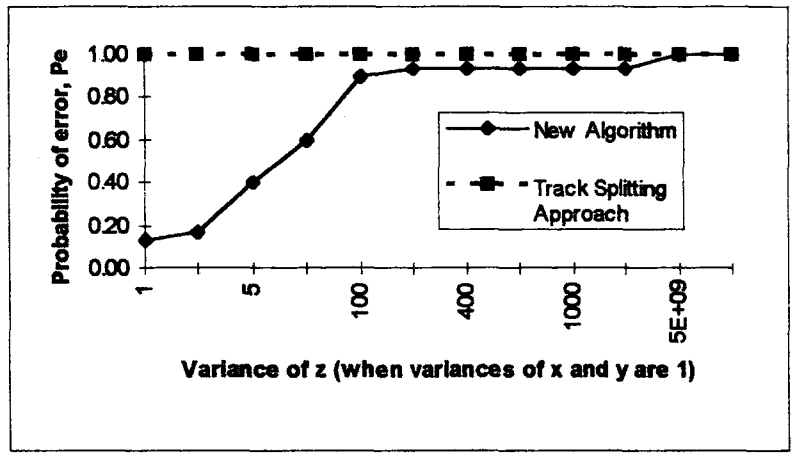


Figure 4.11: Pe versus Variance of z (when distance between initial states of targets are 2.8)

Table 4.14: 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 initial states of targets are 300)

Variance of z	E (New Algorithm)	Pe (=E/30) (New Algorithm)	E (Track Splitting Approach)	Pe (=E/30) (Track Splitting Approach
1	0	0.00	2	0.07
2	0	0.00	5	0.17
5	0	0.00	15	0.50
10	0	0.00	22	0.73
20	0	0.00	28	0.93
50	0	0.00	30	1.00
100	0	0.00	30	1.00
250	4	0.13	30	1.00
400	12	0.40	30	1.00
500	13	0.43	30	1.00
1000	18	0.60	30	1.00
$5X10^6$	27	0.90	30	1.00
$5X10^9$	27	0.90	30	1.00
$5X10^{12}$	27	0.90	30	1.00

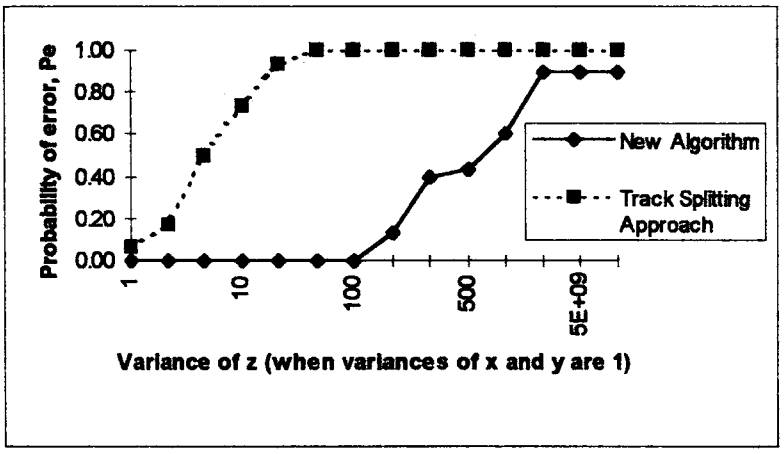


Figure 4.12: Pe versus Variance of z (when distance between initial states of targets are 300)

CHAPTER 5

CONCLUSION AND FUTURE WORK

In multitarget tracking systems, data association (correlation) is an important problem. Multitarget tracking systems use different algorithms to solve this problem. Some of them require gating procedure, but some algorithms don't require it.

In this thesis, it is aimed to develop a different approach to data association problems. For that purpose, surveillance region is initially partitioned into subregions, called predicted state nodes, and then, sensor measurements are classified into subgroups by a distance minimisation rule, which searches for measurements yielding a minimum summed distance between real observation values and predicted observation locations

This approach, or algorithm, can be named as an N-scan back, nearest neighbour, recursive algorithm. The term "recursive" is chosen because of the fact that a recursive distance calculation method is used in finding minimum summed distance between real observation values and predicted observation locations.

This algorithm, together with its software and hardware requirements, can be viewed as a subsystem of a general multitarget tracking (MTT) system.

As shown in Section 4.3, Performance of the new algorithm is better than that of "Track Splitting Approach". For example, in Figures 4.3 and 4.4, one can easily see that the new algorithm's performance is better when number of targets decreases. In addition, in figure 4.6, it can be deduced that for any initial distance value between targets, the new algorithm is preferable. Moreover, effect of the increase in variance v(k) in Equation (4.1) is presented in Figures 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 and it is seen that, for any variance value, the new algorithm's performance is higher than that of "Track Splitting Approach".

As a result, by looking at simulation results of computer implementation of this algorithm, one can conclude that it works properly under assumed conditions. Especially for targets which are not closely spaced, this algorithm can be viewed as an optimum data association algorithm.

As a future work, the implementation efficiency of the proposed N-scan back algorithm should be examined as far as computational requirements are concerned.

REFERENCES

[1]Blackman, S. S., Multiple-Target Tracking with Radar Applications, Artech House, INC., 1986

[2]Bar-Shalom, Y., Multitarget-Multisensor Tracking: Advanced Applications, Volume 1, University of Connecticut, Artech House, INC., 1990

[3]Bar-Shalom, Y., Multitarget-Multisensor Tracking: Advanced Applications, Volume 2, Artech House, INC., 1990

[4]Sea, R.G., " Optimal Correlation of Sensor Data with Tracks in Survaillance Systems," Proceedings of the Sixth International Conference on System Sciences, Jan. 9-11, 1973, Honolulu, HI, pp. 424-426

[5]Stein, J.J., and Blackman S.S., "Generalized Correlation of Multi-Target Track Data," IEEE Transactions on Aerospace and Electronics Systems, AES-11, Nov. 1975, pp. 497-508

[6]Sittler, R.W., "An Optimal Data Association Problem in Surveillance Theory," IEEE Transactions on Information Theory, vol. IT-20. pp. 423-432, July 1974.

[7]Bar-shalom Y. and Tse, E., "Tracking in a Cluttered Environment with Probabilistic Data Association," in Proc. 4th Symp. Nonlinear Estimation, Univ. Calif., San Diego, Sept. 1973, and Automatica, vol. 11, pp. 451-460, Sept. 1975.

[8]Bar-shalom, Y., "Extension of the Probabilistic Data Association Filter to Multitarget Environment," in Proc. 5th Symp. Nonlinear Estimation, Univ. Calif., San Diego, Sept. 1974.

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

[10]Trunk, G.V., "Range Resolution Of targets Using Autimatic detectors," IEEE Transactions on Aerospace and Electronic Systems, AES-14, Sept. 1978, pp. 750-755.

[11]Rickard, J.T. and Dillard, G.M., "Adaptive Detection Algorithms for Mulitple-target Situations," IEEE Transactions on Aerospace and Electronic Systems, AES-13, July 1977, pp. 338-343

[12]Bar-Shalom Y. and Xiao-Rong Li, Estimation and Tracking:Principles, Techniques, and Software, Artec House, Inc, 1993

[13]Ford, L.R., and Fulkerson, D.R., Flows in Networks, Princeton, NJ: Princeton University Press, 1962.

[14]Kuhn, H.W., "The Hungarian Method for the Assignment Problem," Naval Research Logistics Quarterly, No. 2, 1995, pp. 83-97.

[15]Burgeois, F., and Lassalle, J.C.,"An Extention of the Munkres Algorithm for the Assignment Problem to rectangular matrices," Communications of the ACM, vol. 14, Dec. 1971, pp. 802-806.

[16]Fraser, E.C. and Meier, L., "Mathematical Models and Optimum Computation for Computer-aided Active Sonar Systems," U.S. Navy Electronic Lab., SRI Final Rep. (First Year), San Diego, CA, Contract N123-(953)54486A, Mar. 1967.

[17]Smith, P. and Buechler, G., "A Branching Algorithm for Discriminating and Tracking Mulitple Objects," IEEE Transactions on Automat. Contr., vol. AC-20, pp. 101-104, Feb. 1975

[18]Morefield, C. L., "Application of 0-1 Integer Programming to Multitarget Tracking Problems," in Proc. IEEE Conf. Decision and Control, Dec. 1975 and IEEE Trans.Automat. Contr., vol. AC-22, pp. 302-312, June 1977

[19]Singer, R.A., Sea, R.G. and Housewright, R.B., "Derivation and Evaliation of Improved Tracking Filters for Use in Dense Multi-Target Environments," IEEE Transactions on Information theory, IT-20, July 1974, pp. 423-432.

[20]Reid, D. B., "An Algorithm for Tracking Multilple Targets," IEEE Transactions on Automatic Control, AC-24, Dec. 1979, pp. 843-854.

[21]Reid, D. B., "A Multiple Hypothesis Filter for Tracking Multiple Targets in a Cluttered Environment," Lockheed Missiles and Space Company Report, No. LMSC, D-560254, sept. 1977.

[22]Bar-Shalom, Y. and Jaffer, A., "Adaptive Nonlinear Filtering for Tracking with Measurements of Uncertain Origin," in Proc. 1972 IEEE Conf. Decision and Control, pp. 243-247, New Orleans, LA, Dec. 1972.

[23]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.

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

[25]Apostal, A.M., "Mathematical Analysis," Addison-Wesley, Reading, Massachussetts, 1958.

APPENDIX A

ESTIMATION OF GAUSSIAN RANDOM VARIABLE, X, AS AN EXAMPLE FOR LINEAR ESTIMATION IN STATIC SYSTEMS, [12]

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 [12])

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

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

and the corresponding conditional covariance matrix is:

$$P_{xx|z} = E[(x - \hat{x})(x - \hat{x})'|z] = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} \qquad (A.2)$$

(*) 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 uadratic.The widespreed use of quadratic criterion is primarily due to (relative) ease of obtaining the solution.

where

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

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

$P_{xx} = cov(x) = E[(x - \bar{x})(x - \bar{x})']$

$P_{zz} = cov(z) = E[(z - \bar{z})(z - \bar{z})']$

$P_{xz} = cov(x,z) = E[(x - \bar{x})(z - \bar{z})'] = P_{zx}$

Proofs of equation (A.1) and equation (A.2) are available at page 43-44 of [23]

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

DYNAMIC ESTIMATION AS A RECURSIVE STATIC ESTIMATION

The state estimate at k+1 and its covariance can be calculated by modifying static estimation equations, Equation (A.1) and Equation (A.2):

These modification can be realized by following substitutions which are indicated by "→".

$x \rightarrow x(k+1)$

$\bar{x} \rightarrow \hat{x}(k+1|k)=E[x(k+1)|Z^k]$

$z \rightarrow z(k+1)$

$\bar{z} \rightarrow \hat{z}(k+1|k)=E[z(k+1)|Z^k]$

$\hat{x} \rightarrow \hat{x}(k+1|k+1)=E[x(k+1)|Z^{k+1}]$

$P_{xx} \rightarrow P(k+1|k)=cov[x(k+1)|Z^k]=cov[\tilde{x}(k+1|k)|Z^k]$

$P_{zz} \rightarrow S(k+1)=cov[z(k+1)|Z^k]=cov[\tilde{z}(k+1|k)|Z^k]$

$P_{xz} \rightarrow cov[x(k+1),z(k+1)|Z^k]=cov[\tilde{x}(k+1),\ \tilde{z}(k+1)|Z^k]$

$P_{xx|z} \rightarrow P(k+1|k+1)=cov[x(k+1)|Z^{k+1}]=cov[\tilde{x}(k+1|k+1)|Z^{k+1}]$

$P_{xz}P_{zz} \rightarrow W(k+1)=cov[x(k+1),z(k+1)|Z^k]S(k+1)^{-1}$

where

$x(k+1)$: the variable which will be estimated. I.e., the state at $k+1$

$\hat{x}(k+1|k)$: mean of $x(k+1)$ prior to $k+1$. It is also called (one-step) predicted state

$z(k+1)$: observation at $k+1$

$\hat{z}(k+1|k)$: mean of $z(k+1)$ prior to $k+1$. It is also called the predicted measurement

$\tilde{x}(k+1)$: $x(k+1)- \hat{x}(k+1|k)$

$\tilde{z}(k+1)$: $z(k+1)- \hat{z}(k+1|k)$

$\hat{x}(k+1|k+1)$: the estimate posterior to $k+1$. It is also called the updated state estimate (or just the updated state)

$P(k+1|k)$: the prior covariance matrix of the state variable $x(k+1)$ to be estimated. It is also called the state prediction covariance or predicted state covariance

$S(k+1)$: the prior covariance of the observation $z(k+1)$. It is also called the measurement prediction covariance.

$cov[x(k+1),z(k+1)|Z^k]$: the covariance between the variable to be estimated $x(k+1)$ and the observation $z(k+1)$

$P(k+1|k+1)$: the posterior covariance of the state $x(k+1)$. It is also called the updated state covariance.

$W(k+1)$: filter gain

Using the above notation and applying dynamic and measurement equations (equation (2.8) and equation (2.10)) where necessary we are able to construct an algorithm which will give us the estimate of the updated state and the updated covariance at time k+1. The detailed derivation of this algorithm is given at page 214-215 of the [12] and the related flowchart is given in Figure 2.3

APPENDIX C

APPROXIMATION OF AN ABSOLUTELY CONTINUOUS RANDOM VECTOR BY A DISCRETE RANDOM VECTOR [24]

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 $F_{y_0^m}(.) \in D^m$, which minimizes the objective function J(.) over the set D^m :

$$J(F_{y_0^m}(.)) = \min_{F_{y^m}(.) \in D^m} J(F_{y^m}(.)). \qquad (C.1)$$

where

$$J(F_{y^m}(.)) \overset{\Delta}{=} \int_{R^m} [F_{x^m}(a) - F_{y^m}(a)]^2 \, da, \qquad F_{y^m}(.) \in D^m. \quad (C.2)$$

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 now state two theorems and defined some symbols which are used. The proofs of the theorems are given in [25]

Theorem C.1 [25]

Let $f(y) \overset{\Delta}{=} f(y_1, y_2,..., y_l)$ be a real-valed 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* .

Therom C.2 [25]

Let $f(y) \overset{\Delta}{=} f(y_1, y_2,..., y_l)$ be a real-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 $G \overset{\Delta}{=} \det\{[\nabla^2 f(y)]\big|_{y=y_0}\} \neq 0$, where

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

Let G_{l-k} be the determinant obtained from G by deleting the last k rows and columns. If th *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 witk "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+1" 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,

$$
\begin{aligned}
S \overset{\Delta}{=} \{g(x):\quad & g(x)=0 && \text{for } x<y_1; \\
& g(x)=P_i && \text{for } y_i \le x < y_{i+1},\ y_i \in (-\infty, +\infty); \\
& g(x)=1 && \text{for } x \ge y_L;\ y_{i+1} > y_i,\ y_i \in (-\infty, +\infty); \\
& i=1,2,\ldots,L-1\}
\end{aligned}
$$

In order to find the optimum discrete random variable witl L possible values that approximates an absolutely continuous random variable X with distribution function $F_X(.)$, we must find a distribution function $F_{y_0}(.)$ which minimizes the objective function J(.) over the set D:

$$J(F_{y_0}(.)) = \min_{F_y(.) \in D} J(F_y(.)).\,, \tag{C.3}$$

$$= \min_{g(.) \in S} J(g(.)). \tag{C.4}$$

where

$$J(F_y(.)) \overset{\Delta}{=} \int_{-\infty}^{+\infty} [F_x(a) - F_y(a)]^2 \, da \tag{C.5}$$

The equality in equation (C.4) follows from the following arguments. Let a step function $g_0(.)\in S$ 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):

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

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

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

It follows from Theorem C.1 that if $g_0(x)$, which is defined by

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

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

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

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

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

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

Using equation (C.8) and Teorem C.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 C.1.

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

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

Table C.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							
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	1.005				
	$P_{i,0}$	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	0.162	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
	$P_{i,0}$	0.093	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 th ith possible value of y_0: $P_{i,0} \overset{\Delta}{=} \text{Prob}\{ y_0 = y_{i,0}\}$.