Purdue University
Purdue e-Pubs

ECE Technical Reports | Electrical and Computer Engineering

6-3-1996

Suboptimal Target Tracking in Clutter Using a Generalized Probabilistic Data Association Algorithm

Wai Ying Kan
Purdue University School of Electrical and Computer Engineering

James V. Krogmeier
Purdue University School of Electrical and Computer Engineering

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr

Kan, Wai Ying and Krogmeier, James V., "Suboptimal Target Tracking in Clutter Using a Generalized Probabilistic Data Association Algorithm" (1996). *ECE Technical Reports.* Paper 99.
http://docs.lib.purdue.edu/ecetr/99

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

Suboptimal Target Tracking in Clutter Using A Generalized Probabilistic Data Association Algorithm

Wai Ying Kan
James V. Krogmeier

TR-ECE 96-9
June 1996

School of Electrical
and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285

Suboptimal Target Tracking in Clutter Using a Generalized Probabilistic Data Association Algorithm[†]

Wai Ying Kan and James V. Krogmeier
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285

June **3,** 1996

[†]This research was supported by the NEC Faculty Fellows Program.

Contents

List of Tables

List of Figures

Abstract

Simple tracking algorithms based upon nearest neighbor filtering do not correctly consider measurement origin uncertainty and, therefore, fail to perform well in situations of high target density and clutter. The optimal tracking algorithm for commonly used target–clutter models computes the posterior density of the target state conditioned on the past history of observations. This posterior density is a Gaussian mixture with the number of terms equal to the number of possible ways to associate observations and targets. Though a recursive algorithm may be developed for the optimal estimator, it requires exponentially growing memory and computation and is, therefore, unimplementable. In this paper a new suboptimal algorithm is proposed where approximation is done by naturally partitioning and grouping the target state estimates into a set of approximate sufficient statistics. A new criterion function is introduced in this approximation process. The well–known Probabilistic Data Association filter (PDAF) turns out to be a special case of the new algorithm. Comparisons are made for the proposed estimator versus the PDAF.

1 Introduction

Target tracking is an old problem with origins going back as far as eighteenth century astronomers who first attempted to determine the orbits of the visible planets. More modern work can trace its ancestry to the early 1960s where the problem was driven by applications in ballistic missile defense, orbital vehicle tracking, and air traffic control. Certain applications of target tracking have become relatively more important in the last few years. These include air traffic control [1] fueled by large growth in civilian aviation and the resulting traffic congestion near major airports, and highway vehicle surveillance [2], motivated by current interest in intelligent transportation systems.

A major issue in the design of target tracking systems is the uncertainty associated with the origin of measurements. Such uncertainty arises due to the presence of clutter, receiver related false alarms, and other nearby targets. In many situations the measurement origin uncertainty is a far more important impairment to tracking performance than is the noise associated with the measurements themselves. The use of standard trajectory estimation algorithms where the measurement nearest (in some metric) to the predicted measurement is chosen to update a track can lead to very poor performance when the density of spurious measurements is high. Such an algorithm (the nearest. neighbor filter) does not properly account for the fact that the measurement chosen for track update may be unrelated to the target. The recent introduction of advanced sensors which capture new types of target information (target signatures, images, etc.) in addition to position and velocity have made advances in tracking algorithms possible. At the same time these new data collection possibilities complicate implementation by adding to the huge amount of data that must be processed. Processing power will therefore continue to be a bottleneck in the implementation of sophisticated tracking algorithms.

Some of the earliest tracking research done in the modern spirit can be traced to the 1964 paper by Sittler [3]. In order to account for measurement origin uncertainty, he proposed splitting the track whenever more than one observation was made in the vicinity of a predicted measurement. A likelihood function for each trajectory was computed and those falling below a threshold were dropped in order to avoid an exponential growth in complexity. Sittler's work was done before the Kalman filter became popular in trajectory estimation problems. Similar approaches were taken by Stein and Blackman [4] and Smith and Buechler [5] who modernized Sittler's approach and incorporated Kalman filtering and

dynamic .modeling of the targets. In the same spirit Morefield [6] solved for the maximum likelihood data association hypothesis by formulating the problem using integer linear programming. Though the algorithm is a batch processing method, it can be reformulated in a recursive way with a loss of global optimality.

The tracking algorithms mentioned above are all based upon finding the maximum likelihood. data association. Once measurements have been grouped into individual tracks, state estimates and error covariances are computed from a standard set of Kalman filters. In this sense they are basically non–Bayesian methods because the estimated trajectories are computed assuming that the maximum likelihood data association is correct. The resulting state estimates and covariances do not account for the possibility that the data association is incorrect [7].

Bayesian approaches to target tracking were first developed using the nearest neighbor filter and a modification to the Kalman filter which accounted for the ***a priori*** probability that the measurement used for state update was spurious [8, 9]. Later, Jaffer and Bar-Shalom [10] modified the filter to use posterior statistics on the correctness of the measurement used in the nearest neighbor filter. More recent work has focussed on the incorporation of all validated measurements into the target trajectory estimate using a so–called "all neighbors" filter. The optimal algorithm in this Bayesian setup was derived by Singer, Sea and Housewright [11] and involves a geometrically expanding tree of trajectories which account for all possible data associations. For a single target in clutter, the minimum mean squared error estimate of the state is a linear combination of all trajectory estimates in the tree weighted by the posterior probabilities of each corresponding data association. Since this procedure involves an exponentially growing memory, suboptimal algorithms are required for implementation. One suboptimal algorithm, known as the N–scan filter [11], involves splitting tracks back only N scans in time. This prevents memory requirements from growing. In the examples included in [11] it was noted that near optimal performance was obtained for N as small as one.

The most successful algorithm in the class of Bayesian all–neighbors filters is the probabilistic data association (PDA) filter of Bar-Shalom and his collaborators [12, 13]. It corresponds to an $N = 0$ scan filter in the terminology of Singer et al. The algorithm updates the target state estimate using all validated measurements and posterior probability weightings. The PDA algorithm is derived by making the assumption that the one step

prediction density of the state given all past observations is a Gaussian density (strictly speaking it is a Gaussian mixture). From this assumption the PDA algorithm is derived. It achieves good performance with complexity only moderately greater than that of the standard nearest neighbor filter.

The outline of the paper is as follows. The basic modeling assumptions used are given in Section 2 followed by a brief discussion of the optimal algorithm. Section **3** develops the hypothesis clustering approach and summarizes the new tracking algorithm for the case of one target in clutter. Simulation results are presented in Section 4 and the paper ends with a short conclusion.

2 Model for Target and Clutter

In the following we outline our model for tracking a single target with noisy observations of the state trajectory embedded in clutter. The model we use here will apply to a variety of tracking algorithms including: the optimal linear filter, the nearest neighbor filter, the optimal nonlinear filter, and the probabilistic data association filter. For simplicity, we will consider only linear dynamic models for the state. Nonlinear models arise naturally from physical considerations in tracking problems but: (1) the fundamental problem in target tracking in clutter is the nonlinearity due to unknown data association rather than nonlinearities due to state dynamics, and (2) the usual approach to nonlinear dynamics is linearization about a known trajectory anyway.

2.1 Target dynamic **model**

The state of the target at discrete time k is represented by an n_x x 1 vector x_k which is assumed to satisfy a dynamic model of the form

$$x_{k+1} = Fx_k + w_k \tag{1}$$

where the initial state x_0 has a Gaussian distribution with mean μ_0 and covariance P_0 and the process noise w is an independent sequence of zero mean Gaussian random vectors with covariance matrix Q. As usual, we assume that the initial state and the noise are statistically independent. Note also, that we may allow F and Q to depend upon the time index k. This is not explicitly indicated in the interest of a simpler notation.

State observations for such a model are written

$$z_k = Hx_k + v_k \tag{2}$$

where H is n_z x n_x and the measurement noise v is a sequence of independent Gaussian random vectors with zero mean and covariance R. The matrices H and R may be allowed to depend upon the time index k. Finally! it is assumed that x_0, w, and v are statistically independent of each other.

While there is no need to restrict the structure of the matrix F above, the canonical tracking example is the constant velocity model where (for one dimensional motion with uniform sampling of a continuous state model with rate $1/T$), $x_k = \begin{bmatrix} \xi_k & \dot{\xi}_k \end{bmatrix}^T$,

$$F = \begin{bmatrix} 1 & \\ 0 & \end{bmatrix}, \quad \text{and} \quad Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} \text{q.}$$

If the measurement consists of position only, then one would have $z_k = \xi_k + v_k$, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and R would be a scalar. In simulations we will usually use a normalization of the state and output equations which amounts to using T for the unit of time and $\sqrt{R}$ for the unit of distance. The new state is $\bar{x}_k = \begin{bmatrix} \xi_k/\sqrt{R} & \dot{\xi}_k T/\sqrt{R} \end{bmatrix}^T$, the new observation is $\bar{z}_k = z_k/\sqrt{R}$, and the model matrices become

$$\bar{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \bar{Q} = \begin{bmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{bmatrix} (qT^3/R),$$

H = H, and R = 1. The advantage of the normalized model in simulations is that it contains but a single parameter q = qT^3/R controlling the "tracking difficulty [13]." By adding more components to the model above, we can consider target motion in two or three dimensions.

2.2 Remark on notation

To denote the probability of an event E we will write $P(\mathcal{E})$. Probability mass functions will be denoted by p and continuous probability density functions by f. For joint density–mass functions we will also use f. If δ is a discrete random variable, we may use either of the following to indicate its mass function: $p_\delta(\cdot)$ or $p(\delta)$. In addition, we may write $p_\delta(0)$ or $p(\delta = 0)$ to indicate the same thing. Similar notation is used for continuous random variables.

2.3 Measurement model for clutter

Many different models for clutter in target tracking problems have been proposed. It seems clear that a reasonable model for clutter should (1) be independent of the target state, and (2) be dependent upon the volume observed by the sensors which produce target state observations. If we wish to model clutter which arises (as in a radar tracking problem) from fixed objects on the ground (trees, tall buildings, etc.) or from thermal noise in the radar receiver, then a reasonable approach would say that clutter locations are a realization of a spatial Poisson process taking values in the n_z x 1 observation space. Associated with the process is a density λ_c with the interpretation that if a region in the observation space has volume V, then the number of clutter centers located in that region is a Poisson random variable with parameter $\lambda_c V$. Furthermore, conditioned upon the number of clutter centers in a region, the locations are independently and uniformly distributed.

Most tracking algorithms contain an observation preprocessing stage where candidate measurements are validated before use in track update. Thus, a tracking algorithm would recursively generate a sequence of validation gates $\mathcal{G}_k$, k $\geq$ 0, which are subsets of the n_z–dimensional observation space. The idea being that a measurement received in scan k is considered for track update only if it falls inside $\mathcal{G}_k$. Since the tracking algorithm must recursively generate the gates, $\mathcal{G}_k$ must be a function of all past (i.e., up to scan k – 1) validated measurements. We will suppose that an observation vector

$$y_k = \left[\begin{array}{cccc} y_{k,1}^T & y_{k,2}^T & \cdots & y_{k,m_k}^T \end{array} \right]^T$$

of size $n_z m_k \times 1$ ($m_k \geq 0$ is the number of validated measurements) is produced by the sensor at time k. If the target measurement is detected, then it is one of the measurements $y_{k,i}$ and any others are due to clutter. Note that the ordering is assumed to be random.

Let V_k be the volume of $\mathcal{G}_k$ and let n_k be the number of validated clutter measurements in scan k ($0 \leq n_k \leq m_k$). Using the spatial Poisson model for clutter generation, it follows that the conditional distribution of n_k given V_k is Poisson with parameter $\lambda_c V_k$. Furthermore, if (assuming $n_k > 0$)

$$c_k = \left[\begin{array}{cccc} c_{k,1}^T & c_{k,2}^T & \cdots & c_{k,n_k}^T \end{array} \right]^T$$

denotes the vector of clutter measurements validated in scan k, then the probability density of c_k given n_k and $\mathcal{G}_k$ is specified by noting that the $c_{k,i}$ are independently and uniformly distributed over the gate $\mathcal{G}_k$.

At a :particular scan time k, the target related measurement z_k will be present in the observation set only if the target is actually detected. To model the detection process let $\{d_k : k \geq 0\}$ be an i.i.d. sequence taking values 0 and 1 with the interpretation that the target is detected at time k if and only if $d_k = 1$. Define the detection probability to be $P_D = p_d(1)$. We include the possibility that z_k will fall outside of the gate in the detection probability. Of course, the validation gates are designed to contain the target related measurement with high probability.

The observation process is of varying dimension depending upon the detection of the target and the number of clutter measurements. The observation at scan time k has the form (m_k, y_k) where m_k is a non–negative integer indicating the number of measurements and y_k is a real vector of dimension $m_k n_z$. This notation is used to emphasize the fact the observation contains the number of measurements information.

To completely specify the measurements in an observation scan we need to know if the scan contains a target related measurement, and, if so, where that measurement is located in the observation set. To keep track of the actual measurement situation. we hypothesize a data association random process $\{\pi_k : k \geq 0)$. Define π_k as a random function of the random variables d_k and n_k in the sense that

$$\pi_k = (j_k, n_k + d_k).$$

The first component j_k indicates the position of the target related measurement in the vector y_k; it may take values between 0 and $m_k = n_k + d_k$. If $d_k = 0$, then $j_k = 0$ and (for $n_k \geq 1$)

$$y_k = [c_{k,1}^T, c_{k,2}^T, \ldots, c_{k,n_k}^T]^T.$$

If $d_k = 1$, then j_k takes a value between 1 and $m_k = n_k + 1$ and

$$y_k = [c_{k,1}^T, \ldots, c_{k,j_k-1}^T, z_k^T, c_{k,j_k}^T, \ldots, c_{k,n_k}^T]^T;$$

in otherwords, z_k appears a.s the element y_{k,j_k} in the vector y_k of dimension $m_k n_z = (n_k + d_k) n_z$. If $n_k = 0$, then y_k consists of the single measurement z_k when $d_k = 1$, or is empty when $d_k = 0$.

The marginal distributions of the j_k will be specified in such a way that they are identically distributed, and, in addition, we will assume that they are independent. The distribution of j_k is specified by conditioning on d_k and n_k. If no target detection is made

$p(j_k = 0|d_k = 0, n_k) = 1$ regardless of the value of n_k while if the target is detected

$$p(j_k|d_k = 1, n_k) = \begin{cases} 1/(n_k + 1) & \text{for } 1 \leq j_k \leq n_k + 1 \\ 0 & \text{otherwise} \end{cases}$$

As specified above the data association process π_k is iid. Each must take values of the form $\pi_k = (j_k, m_k)$ where $m_k \geq 0$ and $0 \leq j_k \leq m_k$. The unconditional distribution of π_k is found from the work given above as

$$P\{\pi_k = (j_k, m_k)\} = \begin{cases} (1 - P_D)p_n(m_k) & m_k \geq 0,\ j_k = 0 \\ \frac{P_D p_n(m_k - 1)}{m_k} & m_k \geq 1,\ 1 \leq j_k \leq m_k \end{cases} . \tag{3}$$

2.4 The Optimal Single Target Tracking Algorithm

For tracking a single target in clutter it is natural to choose the mean-squared error *(MSE)* as the criterion for optimality. In this section we summarize the MMSE estimator and illustrate a recursive algorithm for its computation. The goal is the computation of the estimate of the state at time k given the sequence $\Gamma_k = \{\gamma_0, \ldots, \gamma_k\}$ of observations up to time k. The algorithm which results is not practical because it requires exponentially growing memory and computation. By conditioning on the past data association sequence we can write

$$\mathrm{E}\{x_k|\Gamma_k\} = \sum_{\Pi_k} E\{x_k|\Gamma_k, \Pi_k\} p(\Pi_k|\Gamma_k) \tag{4}$$

where the sum is over all Π_k compatible with Γ_k. Recall that Γ_k contains the information on past numbers of validated measurements $M_k = \{m_0, \ldots, m_k\}$. Given this we may just as well index the data association Π_k using $J_k = \{j_0, \ldots, j_k\}$ the sequence of locations of the target related measurements.

The optimal algorithm [11, 14] computes the estimate in Equation *(4)* by setting up recursive equations based upon the Kalman filter for the computation of the terms $\mathrm{E}\{x_k|\Gamma_k, \Pi_k\}$ and $p(\Pi_k|\Gamma_k)$. At the end of the processing of the measurements in observation scan k, there is a term in the sum (4) for every possible data association hypothesis Π_k given M_k. In all, there are $\prod_{l=0}^{k}(m_l + 1)$ terms.

3 Hypothesis Clustering for Tracking

3.1 Background on Hypothesis Clustering

A variety. of techniques have been proposed in the literature for dealing with the central problem in target tracking in clutter: the exponentially growing memory and computational requirements of the optimal algorithm [7]. All such methods involve a combination of validation gates, pruning of extremely unlikely hypotheses, and the combination of hypotheses with similar trajectory estimates. These operations may be viewed as a type of clustering algorithm [15] applied to the individual trajectory estimates. Though such notions are central to a tracking algorithm and its performance, the details of individual algorithms are often not precisely laid out in the literature.

Since hypotheses may be viewed as branches on a tree, hypothesis reduction techniques are often viewed a.s either branch pruning or branch combination. Many algorithms use a combination of both techniques. A typical approach [14] uses a threshold on the probability of each data association hypothesis, only those with sufficiently large probability are retained. The same pruning technique has been proposed for the N–scan filters. Pruning with a fixed threshold is not enough by itself to eliminate all complexity problems. Neither is pruning enough to ensure good performance. Hypotheses must also be combined either using the method of Singer [11] to combine data association hypotheses having the last N scans in common or by combining hypotheses which have "similar effects" as in Reid's paper [14]. The former is really a method based on clustering. This is the approach taken in this paper.

3.2 A Tracking Based Clustering Criterion

The structure of computation for the proposed tracking algorithm (one target case) is shown in Figure 1. In the same spirit as was done for the PDA filter [12], we derive the new algorithm based upon the assumption that the conditional density of the state x_k given the observations Γ_{k-1} up to scan time k – 1 is an L component Gaussian mixture. In this paper the number of terms L in the mixture is considered to be fixed from scan to scan. Varying L allows a tradeoff between performance and complexity; the case $\mathrm{L} = 1$ actually corresponds to the PDA filter. With the notation $\mathcal{N}(x|\mu, \mathrm{P})$ to denote the multivariate Gaussian density with inean μ and covariance P, the assumption on the posterior density

of the state x_k can be written

$$f(x_k|\Gamma_{k-1}) = \sum_{i=1}^{L} \beta_k(i)\mathcal{N}(x_k|\hat{x}_{k|k-1}(i), \Sigma_{k|k-1}(i)). \quad (5)$$

Let the statistics parameterizing the terms in the mixture above be denoted by

$$U_{k|k-1}(i) = \left(\hat{x}_{k|k-1}(i),\ \Sigma_{k|k-1}(i),\ \beta_k(i)\right)$$

for $1 \leq i \leq$ **L.** Then the algorithm shown in Figure 1 processes the input statistics $\{U_{k|k-1}(i)\}_{i=1}^{L}$ and the scan k observations $\{y_{k,l}\}_{l=1}^{m_k}$ to produce:

1. The filtered state estimate $\hat{x}_{k|k}$ and the error covariance $\Sigma_{k|k}$.

2. The statistics needed to propagate the solution to the next scan $\{U_{k+1|k}(i)\}_{i=1}^{L}$, where $U_{k+1|k}(i) = \left(\hat{x}_{k+1|k}(i),\ \Sigma_{k+1|k}(i),\ \beta_{k+1}(i)\right)$.

The details concerning each block in Figure 1 are presented in the following sections.

3.3 Hypothesis Tree Construction

Given the validity of the assumption in Equation (5), the one step prediction density $f(x_k|\Gamma_{k-1})$ is updated by the processing of the scan k observations $\{y_{k,l}\}_{l=1}^{m_k}$ to give the posterior density $f(x_k|\Gamma_k)$ as an $L(m_k + 1)$ component Gaussian mixture. The details, which art: by now quite standard [11, **14,** 7], are illustrated by the diagram of Figure 2.

Each component i in the mixture **(5)** is the root of a tree formed from the scan k observation γ_k and the possible data association hypotheses. These are indexed by $0 \leq j \leq m_k$ where $j = 0$ corresponds to the hypothesis that the target was not detected in scan k and $j > 0$ corresponds to the hypothesis that the target related measurernent is $y_{k,j}$. The Kalman filter is applied yielding $L(m_k + 1)$ trajectory estimates

$$\begin{aligned} \hat{\mu}_{k|k}(i,j) &= \hat{x}_{k|k-1}(i) + G(i,j)(y_{k,j} - H\hat{x}_{k|k-1}(i)) \\ S_{k|k} &= \Sigma_{k|k-1}(i) - G(i,j)H\Sigma_{k|k-1}(i) \end{aligned} \quad (6)$$

where $1 \leq i \leq L$, $0 \leq j \leq m_k$, and the gain term is given by

$$G(i,j) = \begin{cases} 0 & \text{if } j = 0 \\ \Sigma_{k|k-1}(i)H^T(H\Sigma_{k|k-1}(i)H^T + R)^{-1} & \text{if } j \neq 0 \end{cases}.$$

A probability $\alpha_k(i,j)$ is associated with each branch in the tree. These are calculated from the measurement model and the data association hypothesis by defining

$$\tilde{\alpha}_k(i,0) = V_k^{-m_k}(1-P_D)p_n(m_k)\beta_k(i)$$

and

$$\tilde{\alpha}_k(i,j) = \mathcal{N}(y_{k,j}|H\hat{x}_{k|k-1}(i), H\Sigma_{k|k-1}(i)H^T + R)V_k^{-(m_k-1)}P_D m_k^{-1}p_n(m_k-1)\beta_k(i)$$

for $j > 0$. The parameters above are as defined in Section 2 on the measurement model. The probability calculation is finished upon normalization

$$\alpha_k(i,j) = \frac{\tilde{\alpha}_k(i,j)}{\sum_{i',j'}\tilde{\alpha}_k(i',j')}. \tag{7}$$

For simplicity in referring to the above statistics we define

$$T_{k|k}(i,j) = \left(\hat{\mu}_{k|k}(i,j),\ S_{k|k}(i,j),\ \alpha_k(i,j)\right)$$

for $1 \leq i \leq$ L and $0 \leq j \leq m_k$.

3.4 Calculation of the Filtered Estimate

Given the assumption in (5), the filtered density $\mathrm{f}(x_k|\Gamma_k)$ is an $L(m_k+1)$ component Gaussian mixture. The means, covariances, and weighting probabilities are computed as in Equations (6) and (7). For the filtered state estimate and error covariance we simply take the mean and covariance of the density $f(x_k|\Gamma_k)$. This is the natural generalization of the PDA [7]. Thus, the filtered estimate at time k is given by

$$\hat{x}_{k|k} = \sum_{i,j}\alpha_k(i,j)\hat{\mu}_{k|k}(i,j) \tag{8}$$

and the error covariance is computed from

$$\Sigma_{k|k} = \sum_{i,j}\alpha_k(i,j)\left(S_{k|k}(i,j) + \hat{\mu}_{k|k}(i,j)\hat{\mu}_{k|k}^T(i,j)\right) - \hat{x}_{k|k}\hat{x}_{k|k}^T. \tag{9}$$

Statistics at $k\|k-1$	$U_{k\|k-1}(i) = \left(\hat{x}_{k\|k-1}, \Sigma_{k\|k-1}(i), \beta_k(i)\right)$	$1 \leq i \leq L$
Intermediate statistics (after filtering)	$T_{k\|k} = \left(\hat{\mu}_{k\|k}(i,j), S_{k\|k}(i,j), \alpha_k(i,j)\right)$	$1 \leq i \leq L$ $0 \leq j \leq m_k$
Intermediate statistics (after model propagation)	$T_{k+1\|k} = \left(\hat{\mu}_{k+1\|k}(i,j), S_{k+1\|k}(i,j), \alpha_k(i,j)\right)$	$1 \leq i \leq L$ $0 \leq j \leq m_k$
Statistics at $k+1\|k$	$U_{k+1\|k}(i) = \left(\hat{x}_{k+1\|k}, \Sigma_{k+1\|k}(i), \beta_{k+1}(i)\right)$	$1 \leq i \leq L$

Table 1: Summary of Statistics

3.5 Computation of New Statistics

In order that the tracking algorithm continues to operate with a fixed memory, it is necessary to reduce the $L(m_k+1)$ terms in the density $f(x_k|\Gamma_k)$ back to L components. In this paper we take the viewpoint that the approximation to reduce the collection of statistics is a type of clustering problem. This has two facets: the first is the criterion used to determine the best grouping of statistics, the second is the choice of representatives for each cluster.

Before application of the clustering algorithm the model is used to propagate the statistics according to

$$\begin{aligned} \hat{\mu}_{k+1|k}(i,j) &= F\hat{\mu}_{k|k}(i,j) \\ S_{k+1|k}(i,j) &= FS_{k|k}(i,j)F^T + Q \end{aligned}$$

for $1 \leq i \leq L$ and $0 \leq j \leq m_k$. The branch probabilities remain unchanged during the model propagation operation. See Figure 2. At this point the set of statistics which summarizes tracking information is

$$\{T_{k+1|k}(i,j) : 1 \leq i \leq L, 0 \leq j \leq m_k\} \tag{10}$$

where $T_{k+1|k}(i,j) = \left(\hat{\mu}_{k+1|k}(i,j), S_{k+1|k}(i,j), \alpha_k(i,j)\right)$.

In order to reduce complexity it is necessary to replace the set of $L(m_k+1)$ statistics above by a set $\{U_{k+1|k}(i)\}_{i=1}^{L}$ of L statistics. This is done by introducing a clustering criterion in the next section. The situation is summarized in Table 1.

3.5.1 Clustering Criterion

Many target tracking algorithms employ some form of hypothesis reduction techniques for complexity reduction. These methods often amount to hypothesis combination where the

idea is to combine hypotheses having similar effects. According to Reid [14] the metric used is that the means and variances of the estimates corresponding to combined hypotheses be "sufficiently close." No further details regarding this suggestion were given in [14]. In this paper we interpret this idea as a clustering problem applied to the hypotheses themselves and then investigate the metrics used to define closeness.

Let $\mathcal{I}_k$ denote the index set $\{(i,j) : 1 \leq i \leq L, 0 \leq j \leq m_k\}$ of the statistics (hypotheses) in Equation (10). Let

$$\mathcal{I}_k = \bigcup_{l=1}^{L} I_k(l).$$

be a partition of the index set. Each such partition corresponds to a proposed grouping (clustering) of the hypotheses represented by the statistics $T_{k+1|k}(i,j)$. The quality of a clustering is measured by a function of the form (i.e., good clusterings minimize this measure)

$$\sum_{l=1}^{L} \sum_{(i,j)\in I_k(l)} D(T_{k+1|k}(i,j), U_{k+1|k}(l)) \tag{11}$$

where two things need be specified:

1. The metric $D(\cdot,\cdot)$
2. The calculation of the cluster representative $U_{k+1|k}(l)$ which is specified to be a function of $T_{k+1|k}(i,j)$, $(i,j) \in I_k(l)$.

Since we are concentrating on a particular scan time in Equation (11) we will drop the subscripts k + 1|k, k + 1, and k in the statistics $T_{k+1|k}(i,j)$, $U_{k+1|k}(l)$, and in their components. This will considerably simplify the notation. Then some possible clustering criteria might be:

(A) $D_A(T(i,j),U(l)) = \|\hat{\mu}(i,j) - \hat{x}(l)\|^2 + \|S(i,j) - \Sigma(l)\|_F^2$ where $U(l)$ is given by

$$\begin{aligned}
\hat{x}(l) &= \frac{1}{|I(l)|} \sum_{(i,j)\in I(l)} \hat{\mu}(i,j) \\
\Sigma(l) &= \frac{1}{|I(l)|} \sum_{(i,j)\in I(l)} S(i,j) \\
\beta(l) &= \sum_{(i,j)\in I(l)} \alpha(i,j).
\end{aligned}$$

(B) D_B the same as D_A but with $U(l)$ defined by

$$\begin{aligned}\hat{x}(l) &= \frac{1}{\beta(l)} \sum_{(i,j)\in I(l)} \alpha(i,j)\hat{\mu}(i,j) \\ \Sigma(l) &= \frac{1}{\beta(l)} \sum_{(i,j)\in I(l)} \alpha(i,j)S(i,j) \\ \beta(l) &= \sum_{(i,j)\in I(l)} \alpha(i,j).\end{aligned}$$

Note that in case A the probabilities will not influence the desired optimal grouping while in case B they will.

One might expect better results if a tracking performance–based clustering criterion is used. In order to evaluate the cost of placing index $(\mathrm{i},j) \in \mathcal{I}$ into the group represented by $I(l) \subset \mathcal{I}$ we consider the effect of filtering the target related measurement in the next scan (i.e. scan k + 1) starting from statistic $T(i,j)$ as opposed to starting from statistic $U(l)$. Let y denote the unobservable target related measurement which will be available in the next scan. Starting from $T(i,j)$ the filtered estimate would be

$$\hat{\mu}(i,j) + S(i,j)H^T(HS(i,j)H^T + R)^{-1}(y - H\hat{\mu}(i,\mathrm{j})) \tag{12}$$

and starting from $U(l)$ the filtered estimate would be

$$\hat{x}(l) + \Sigma(l)H^T(H\Sigma(l)H^T + R)^{-1}(y - H\hat{x}(l)). \tag{13}$$

Our goal is to minimize the mean square difference between the filtered estimates in (12) and (13). Since the next scan observation is not yet available, the expectation is over the (approxirnate) joint distribution of $(\mathrm{i},j,\mathrm{y})$ given Γ_k and given that y is target related. Since the joint distribution of (i,j,y) is given by $\mathrm{f}(i,j,y|\Gamma_k) = \mathrm{f}(y|i,j,\Gamma_k)\alpha(i,j)$, the metric in the new clustering criterion is specified by

$$\sum_{l=1}^{L} \sum_{(i,j)\in I(l)} \alpha(i,j) \int \mathcal{N}(y|H\hat{\mu}(i,j), HS(i,j)H^T + R) \cdot \tag{14}$$
$$\cdot \|\hat{\mu}(i,j) + G_1(i,j)(y - H\hat{\mu}(i,j)) - \hat{x}(l) - G_2(l)(y - H\hat{x}(l))\|^2 dy$$

where the gains are given by

$$\begin{aligned}G_1(i,j) &= S(i,j)H^T(HS(i,j)H^T + R)^{-1} \\ G_2(l) &= \Sigma(l)H^T(H\Sigma(l)H^T + R)^{-1}.\end{aligned}$$

As shown in Appendix A the clustering criterion above may be equivalently written as

$$\sum_{l=1}^{L} \sum_{(i,j)\in I(l)} \alpha(i,j)\left(L_1(i,j,l) + L_2(i,j,l)\right) \tag{15}$$

where

$$\begin{aligned} L_1(i,j,l) &= \|(I - G_2(l)H)(\hat{\mu}(i,j) - \hat{x}(l))\|^2 \\ L_2(i,j,l) &= \mathrm{Tr}\left\{(G_2(l) - G_1(i,j))(HS(i,j)H^T + R)(G_2(l) - G_1(i,j))\right\}. \end{aligned}$$

Given the clustering criterion of Equation (15) we also need to specify the calculation of the cluster representatives $U(l) = (\hat{x}(l), \Sigma(l), \beta(l))$. For any fixed partition, it is shown in Appendix B that the set of $U(l)$ minimizing the criterion has

$$\begin{aligned} \hat{x}(l) &= \frac{\sum_{(i,j)\in I(l)} \alpha(i,j)\hat{\mu}(i,j)}{\sum_{(i,j)\in I(l)} \alpha(i,j)} \\ \Sigma(l) &= \frac{\sum_{(i,j)\in I(l)} \alpha(i,j)\left(S(i,j) + (\hat{\mu}(i,j) - \hat{x}(l))(\hat{\mu}(i,j) - \hat{x}(l))^T\right)}{\sum_{(i,j)\in I(l)} \alpha(i,j)} \end{aligned} \tag{16}$$

Note that the probability terms in $U(l)$ play no role in the clustering at this stage (they would, however, play a role in subsequent stages of the algorithm). Therefore, we pick them such that

$$\beta(l) = \sum_{(i,j)\in I(l)} \alpha(i,j).$$

3.5.2 Iterative Optimization

Given the work above, the clustering criterion is completely specified, and all that remains is the method to compute the optimal partition of the hypotheses. Let $c(i,j)$ denote the label ($1 \leq c(i,j) \leq$ L) assigned to the hypothesis indexed by $(i,j) \in \mathcal{I}$ and rewrite the criterion in (15) as

$$\mathcal{F}(c) = \sum_{(i,j)\in\mathcal{I}} \alpha(i,j)\left(L_1(i,j,c(i,j)) + L_2(i,j,c(i,j))\right). \tag{17}$$

In this way the clustering process becomes a well–defined problem in discrete optimization: find c such that (17) attains its global minimum among all possible partition functions c. In theory, the optimal way of solving this discrete combinatorial problem is to compute the

cost for all possible groupings. However, this is not feasible because there are approximately $L^N/L!$ ways of partitioning a set of $N = (m_k + 1)L$ elements into L subsets. Therefore, we result to a suboptimal algorithm (coordinate descent [15]). The basic idea is to find an initial partition and to "move" estimates from one group to another if such a move will improve the value of the criterion function. In general, the procedure only guarantees a local minimum of the criterion function.

The criterion function (17) can be viewed as a function of the N variables $\{c(i,\mathbf{j}): 1 \leq i \leq L, 0 \leq \mathbf{j} \leq m_k\}$ which take values in the set $\{1,2,\ldots,L)$. In applying this method, an estimate statistic or "point" is "moved" from one group to another if the value of the criterion function decreases. As the value decreases after each iteration and there are only a finite number of "points", at least a local minimum can be located.

Let's consider the function c as a vector in the space Z^N where $Z = \{1,2,\ldots,L)$. The iterative optimization algorithm is then:

1. Choose a starting point $c = (c_1,\ldots,c_N)$.
2. Set $i = 1$.
3. Find x_i such that
$$x_i = \text{argmin}_{c_i \in Z} f(c_1,\ldots,c_i,\ldots,c_N)$$
4. Replace c by $c = (c_1,\ldots,x_i,\ldots,c_N)$.
5. Set $i = i + 1 \pmod N$.
6. Repeat from **3** until done.

4 Simulation Results

For the computer simulations reported here we used a state vector consisting of four components (position and velocity in the x- and y-directions) which was normalized as described in Section 2.1. The parameters of the simulations below were the target maneuvering index and the clutter density. Two quantities were computed in order to compare the performance of the various algorithms: the position RMS error and the lost track probability. In all of the simulations, the target detection probability was set equal to 0.99 and the gate constant

was equal to 4. In otherwords, provided that a track was not lost, the probability of having a target related measurement in the gate was nearly equal to one. Clutter was generated according; to the method described in [12]. Each run consisted of 100 scan:; and the results were averaged over 200 independent trials.

4.1 Comparison of Position RMS Error

Let the first and third components of the state vector be denoted by $x(k)$ and $y(k)$, respectively (the position components). The RMS position error for each tracking algorithm was computed from

$$\mathcal{E}_{RMS} = \left[\frac{1}{200} \sum_{j=1}^{200} (x^{(j)}(100) - \hat{x}^{(j)}(100))^2 + \frac{1}{200} \sum_{j=1}^{200} (y^{(j)}(100) - \hat{y}^{(j)}(100))^2 \right]^{1/2}.$$

This was computed as a function of the maneuvering index for two different values of the clutter density. Three tracking algorithms were compared: the PDA, the 1-Scan Back filter of [11], and the algorithm of this paper with $L = 2$ (2-GPDA). The result is shown in Figure **3.**

4.2 Comparison of Lost Track Probability

In this study we use the following definition of a lost track [16]: a track is considered lost when the correct measurement is not in the validation region of at least the last 20 scans. The simulation parameters and the algorithms compared are the same as in the previous section. Simulations were done comparing lost track probability as a function of maneuvering index for two different values of the clutter density (Figure 4), and as a function of clutter density for two different values of the maneuvering index (Figure *5).*

5 Conclusion

A new algorithm for tracking a single target in clutter has been presented vvhich generalizes the well-known PDA method. A new criterion function for specifying the distortion measure was derived in the approximation process. Estimate points were merged together in the fashion that the criterion function was minimized. This process was done by a clustering technique which naturally groups the estimates and choses the centroid as the representative

for those estimate points in that group. It was shown that the PDAF is a special case of this proposed algorithm. The new algorithm exhibited better performance at the expense of a small increase in complexity compared with the PDAF.

A Simplification of the Clustering Criterion

First, we rewrite the integral in the criterion of *(14)* with a minimum of indices for simplicity

$$\int \mathcal{N}(y|H\hat{\mu}, HSH^T + R)\|\hat{\mu} + G_1(y - H\hat{\mu}) - \hat{x} - G_2(y - H\hat{x})\|^2 dy \tag{18}$$

where the gains are given by

$$\begin{aligned} G_1 &= SH^T(HSH^T + R)^{-1} \\ G_2 &= \Sigma H^T(H\Sigma H^T + R)^{-1}. \end{aligned}$$

Then, the norm in the integral can be written as

$$\|A + By\|^2 = A^T A + 2A^T By + y^T B^T By$$

where $A = (I - G_1 H)\hat{\mu} - (I - G_2 H)\hat{x}$ is not a function of y and $B = G_1 - G_2$. Substituting this form into the integral *(18)* results in

$$A^T A + 2A^T BH\hat{\mu} + \int y^T B^T By\mathcal{N}(y|H\hat{\mu}, HSH^T + R)dy. \tag{19}$$

Now the last integral above is simply $\mathrm{E}\{\|By\|^2\}$ given the indicated distribution for y. We have for this distribution

$$\begin{aligned} \mathrm{E}\{\|By\|^2\} &= \mathrm{Tr}\mathrm{E}\{(By)(By)^T\} \\ &= \mathrm{Tr}\{\mathrm{Cov}(By) + \mathrm{E}(By)\mathrm{E}(By)^T\} \\ &= \mathrm{Tr}\{B(HSH^T + R)B^T\} + \hat{\mu}^T H^T B^T BH\hat{\mu}. \end{aligned}$$

With this result, the integral in (18) becomes

$$A^T A + 2A^T B\hat{\mu} + \hat{\mu}^T H^T B^T BH\hat{\mu} + \mathrm{Tr}\{B(HSH^T + R)B^T\}$$

where the trace term above is precisely that in Equation *(15)*. Thus, it only remains to show that $A^T A + 2A^T B\hat{\mu} + \hat{\mu}^T H^T B^T BH\hat{\mu} = \|(I - G_2 H)(\hat{\mu} - \hat{x})\|^2$. Expancling by replacing A and B we obtain

$$\begin{aligned} A^T A &= \hat{\mu}^T(I - G_1 H)^T(I - G_1 H)\hat{\mu} - 2\hat{\mu}^T(I - G_1 H)^T(I - G_2 H)\hat{x} \\ &\quad + \hat{x}^T(I - G_2 H)^T(I - G_2 H)\hat{x} \\ A^T BH\hat{\mu} &= \hat{\mu}^T(I - G_1 H)^T G_1 H\hat{\mu} - \hat{\mu}^T(I - G_1 H)^T G_2 H\hat{\mu} \\ &\quad - \hat{x}^T(I - G_2 H)^T G_1 H\hat{\mu} + \hat{x}^T(I - G_2 H)^T G_2 H\hat{\mu} \\ \hat{\mu}^T H^T B^T BH\hat{\mu} &= \hat{\mu}^T H^T G_1^T G_1 H\hat{\mu} - 2ji^T H^T G_2^T G_1 H\hat{\mu} \\ &\quad + \hat{\mu}^T H^T G_2^T G_2 H\hat{\mu}. \end{aligned}$$

Substituting the above and cancelling terms results in the criterion of Equation (15).

B Choice of Minimizing Cluster Representatives

Consider a scalar version of our clustering problem for simplicity. We then write the clustering criterion as a function of the partition of the index set $\mathcal{I}$ as a union of the disjoint sets $\{I(l)\}_{l=1}^{L}$, and as a function of the cluster representatives[1] $\{U(l)\}_{l=1}^{L}$ according to

$$C(\{U(l)\}_{l=1}^{L}, \{I(l)\}_{l=1}^{L}) = \sum_{i=1}^{L} \sum_{(i,j)\in I(l)} \alpha(i,j)\,(L_1(i,j,l) + L_2(i,j,l))$$

where

$$\begin{aligned} L_1(i,j,l) &= (1 - G_2(l))^2(\hat{\mu}(i,j) - \hat{x}(l))^2 \\ L_2(i,j,l) &= (G_2(l) - G_1(i,j))^2(S(i,j) + R) \end{aligned}$$

and the gains are given by

$$\begin{aligned} G_1(i,j) &= S(i,j)/(S(i,j) + R) \\ G_2(l) &= \Sigma(l)/(\Sigma(l) + R). \end{aligned}$$

Note that in the scalar problem we may set $\mathrm{H} = 1$ without loss of generality.

For a fixed partition we wish to solve for the representative $U(l)$ which minimizes the criterion C. Note first that solving for the best $U(l')$ only involves the term $1 = l'$ in the sum above. Thus, the individual terms are decoupled and we may consider them separately in the minimization. For simplicity, we will drop reference to the index 1 in all of the following equations. Secondly, we note that the component β in $U = (\hat{x}, \Sigma, \beta)$ does not enter into the criterion C and for the moment we will therefore not consider the proper choice of β.

The problem we consider is reduced to the following

$$(\hat{x}^*, \mathrm{C}^*) = \arg\min_{\hat{x},\Sigma} C'(\hat{x}, \mathrm{C})$$

where

$$C'(\hat{x}, \Sigma) = \sum_{(i,j)\in I} \alpha(i,j)\left[(1 - G_2)^2(\hat{\mu}(i,j) - \hat{x})^2 + (G_2 - G_1(i,j))^2(S(i,j) + R)\right],$$

$G_1(i,j) = S(i,j)/(S(i,j) + \mathrm{R})$, and $G_2 = \Sigma/(\Sigma + \mathrm{R})$. Because Σ enters the criterion only through G_2 we will actually consider C' as a function of $\hat{x}$ and G_2 when we perform the

[1] $U(l) = (\hat{x}(l), \Sigma(l), \beta(l))$.

minimization. To get the necessary conditions for a critical point, we take the partials of the form C' with respect to the variables $\hat{x}$ and G_2 and set them equal to zero.

For $\hat{x}$ we obtain

$$\frac{\partial C'}{\partial \hat{x}} = -2 \sum_{(i,j)\in I} \alpha(i,j)(1-G_2)^2(\hat{\mu}(i,j) - \hat{x}).$$

Setting the above equal to zero and noting that the factor $-2(1-G_2)^2$ may be canceled (providecl that $G_2 \# 1$) we obtain the following candidate solution

$$\hat{x}^* = \frac{\sum_{(i,j)\in I} \alpha(i,j)\hat{\mu}(i,j)}{\sum_{(i,j)\in I} \alpha(i,j)}$$

For G_2 we have

$$\frac{\partial C'}{\partial G_2} = 2 \sum_{(i,j)\in I} \alpha(i,j) \left[(G_2 - G_1(i,j))(S(i,j)+R) - (\hat{\mu}(i,j) - \hat{x})^2(1-G_2)\right].$$

Setting the above equal to zero, solving, and substituting $\hat{x} = \hat{x}^*$ we obtain the candidate solution

$$\begin{aligned} G_2^* &= \frac{\sum_{(i,j)\in I} \alpha(i,j)\left[G_1(i,j)(S(i,j)+R) + (\hat{\mu}(i,j) - \hat{x}^*)^2\right]}{\sum_{(i,j)\in I} \alpha(i,j)\left[S(i,j)+R+(\hat{\mu}(i,j)-\hat{x}^*)^2\right]} \\ &= \frac{\sum_{(i,j)\in I} \alpha(i,j)\left[S(i,j)+(\hat{\mu}(i,j)-\hat{x}^*)^2\right]}{\sum_{(i,j)\in I} \alpha(i,j)\left[S(i,j)+(\hat{\mu}(i,j)-\hat{x}^*)^2+R\right]}. \end{aligned}$$

To verify that the critical point $(\hat{x}^*, G_2^*)$ is a minimum of the form C' we need to show that the Hessian is positive definite when evaluated at the critical point. To do this we compute the matrix of second partials:

$$\begin{aligned} \frac{\partial^2 C'}{\partial \hat{x}^2} &= 2(1-G_2)^2 \sum_{(i,j)\in I} \alpha(i,j) \\ \frac{\partial^2 C'}{\partial G_2 \partial \hat{x}} &= 4(1-G_2) \sum_{(i,j)\in I} \alpha(i,j)(\hat{\mu}(i,j) - \hat{x}) \\ \frac{\partial^2 C'}{\partial G_2^2} &= 2 \sum_{(i,j)\in I} \alpha(i,j)\left[S(i,j)+R+(\hat{\mu}(i,j)-\hat{x})^2\right]. \end{aligned}$$

Since the diagonal terms in the Hession above are strictly positive (assuming that $G_2 \neq 1$) and since the off diagonal terms are equal to zero at $(\hat{x}^*, G_2^*)$, we see that the Hession is indeed positive definite.

With the exception of the points $G_2 = \mathbf{1}$ and $\Sigma = -\mathbf{R}$, we have a one–to–one correspondence between G_2 and C. Thus we may solve $C = G_2R/(1 - G_2)$, and write down the solution

$$\Sigma^* = \frac{\sum_{(i,j)\in I} \alpha(i,j)\,[S(i,j) + (\hat{\mu}(i,j) - \hat{x})^2]}{\sum_{(i,j)\in I} \alpha(i,j)}$$

Note that $\Sigma \geq 0$ corresponds to $0 \leq G_2 < \mathbf{1}$ and note that this is always satisfied by the gains here.

Finally we pick

$$\beta^* = \sum_{(i,j)\in I} \alpha(i,j)$$

which completes the specification of the minimizing cluster representatives

$$U^* = (\hat{x}^*, \mathbf{C}^*\beta^*).$$

References

[1] X. R.. Li and Y. Bar-Shalom. Design of an interacting multiple model algorithm for air traffic control tracking. IEEE Transactions on Control *Systems* Technology, 1(5):186–194, September 1993.

[2] M. D. Jolly, S. Lakshmanan, and A. K. Jain. Vehicle segmentation and classification using deformable templates. IEEE Transactions on Pattern *Analysis* and Machine *Intelligence*, 18(3):293–308, March 1996.

[3] R. W. Sittler. An optimal data association problem in surveillance theory. IEEE *Transactions* on Military Electronics, MIL-€2125–139, April 1964.

[4] J. J. Stein and S. S. Blackman. Generalized correlation of multi-target track data. IEEE Transactions on Aerospace and Electronic Systems, AES-I 1(6):1207–1217, November 1975.

[5] P. Smith and G. Buechler. A branching algorithm for discriminating and tracking multiple targets. IEEE Transactions on Automatic Control, AC-20:101–104, February 1975.

[6] C. L. Morefield. Application of 0-1 integer programming to multitarget tracking problems. IEEE Transactions on Automatic Control, AC-22:302–311, June 1977.

[7] Y. Bar-Shalom. Tracking methods in a inultitarget environment. *IEEE* Transactions on *Automatic* Control, AC-23(4):618–626, August 1978.

[8] R. A. Singer and J. J. Stein. An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems. In Proc. *of* the 1971 IEEE *Conference* on Decision and Control, pages 171–175, Miami Beach, FL, 1971.

[9] R. A. Singer and R. G. Sea. New results in optimizing surveillance system tracking and data correlation performance in dense multitarget environments. *IEEE* Transactions on *Automatic* Control, AC-18(6):571–582, December 1973.

[10] A. G. Jaffer and Y. Bar-Shalom. On optimal tracking in multiple-target environments. In Proc. of the Third Symposium on Nonlinear Estimation Theory and its Applications, pages 112–117, San Diego, CA, September 1972.

[11] R. A. Singer, R. G. Sea, and K. B. Housewright. Derivation and evaluation of improved tracking filters for use in dense multitarget environments. IEEE Transactions on *Information* Tlzeory, IT-20(4):423–432, July 1974.

[12] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. Automatica, 11:451–460, 1975.

[13] Y. Bar-Shalom. Tracking and Data Association. Academic Press, Boston, 1988.

[14] D. B. Reid. An algorithm for tracking multiple targets. IEEE Transactions on *Automatic* Control, AC-24(6):843–854, December 1979.

[15] R. O. Duda and P. E. Hart. Pattern Classification and ,Scene Analysis. Wiley-Intel-science, New York, NY, 1973.

[16] X. R. Li and Y. Bar-Shalom. Stability evaluation and track life of the PDAF for tracking in clutter. IEEE *Transactions* on Automatic Control, 36(5):588–601, May 1991.

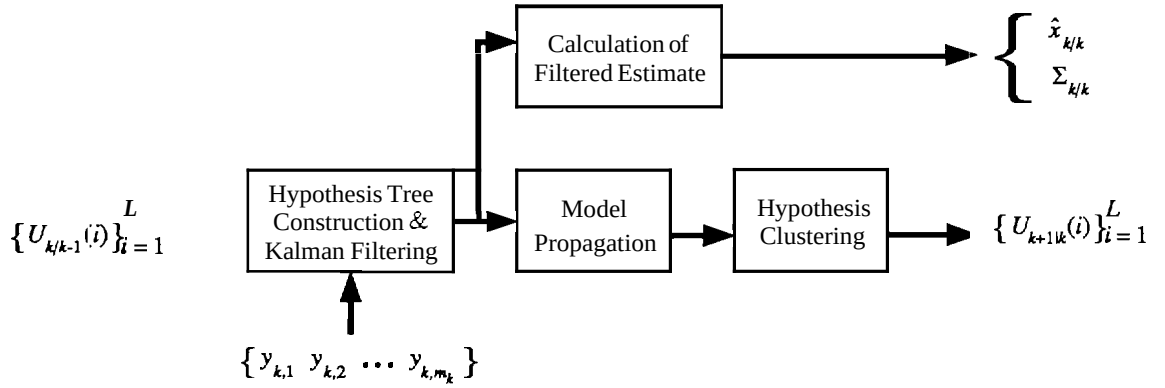


Figure 1: Summary of computation in the generalized PDA algorithm of this paper.

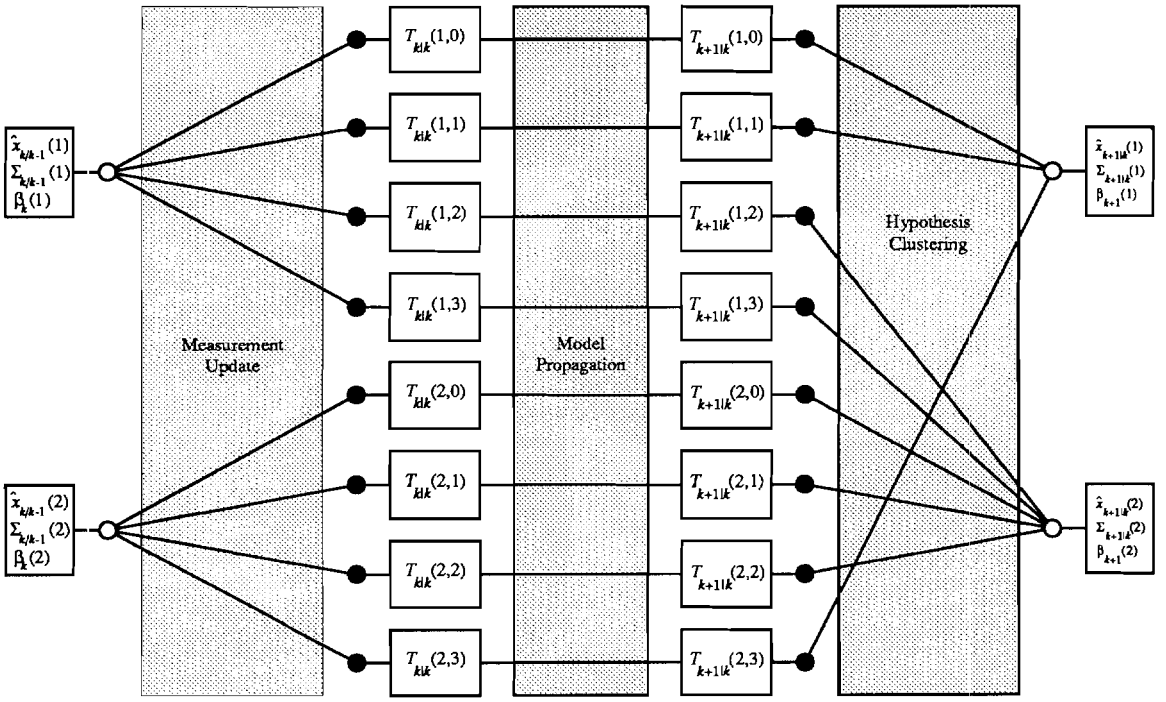
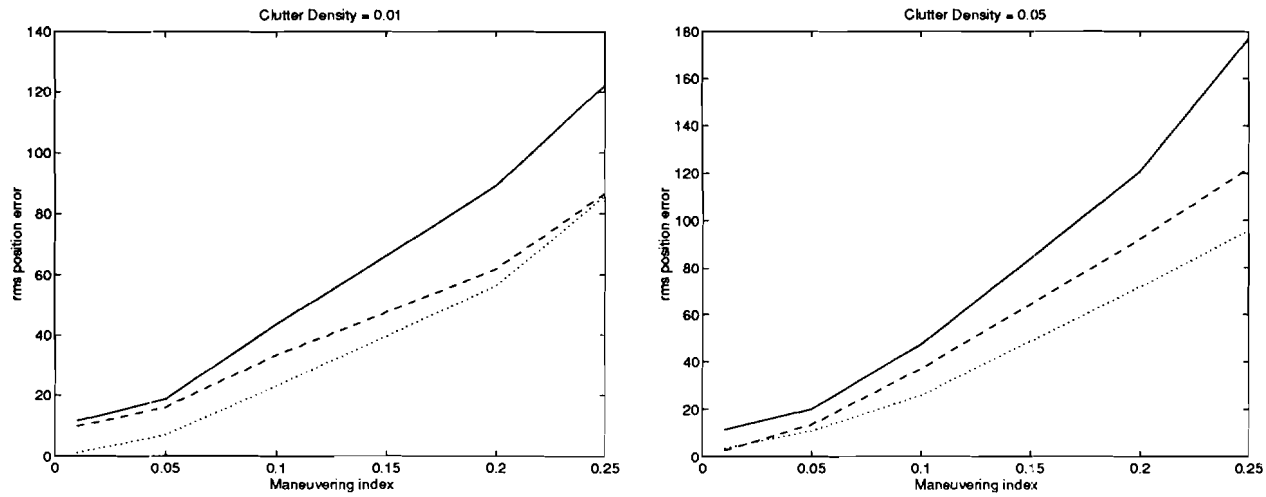
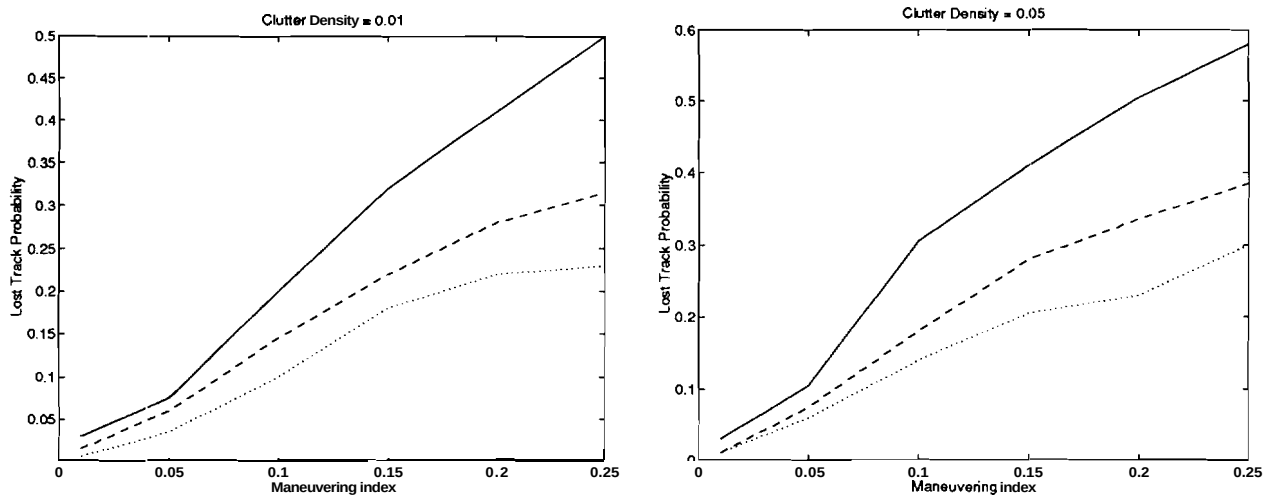


Figure 2: Structure of the Hypothesis Tree. For the case illustrated L = 2 and $m_k = 3$ observations are made at scan time k.



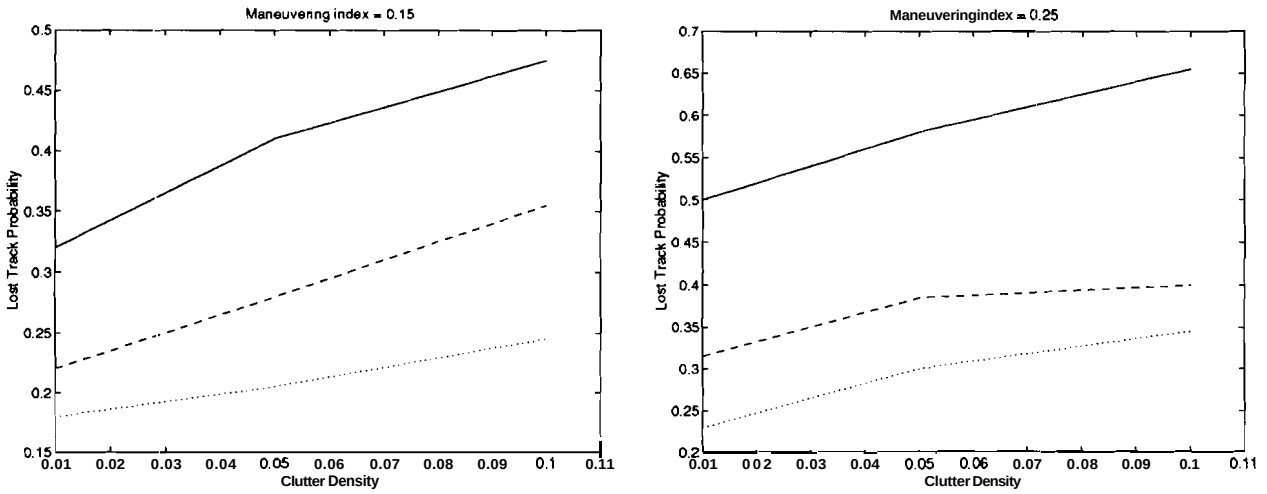
(a) (b)

Figure **3:** Comparison of position RMS error versus maneuvering index for three tracking algorithms (PDA – solid line, 1–Scan Back – dashed line: 2–GPDA – dotted line). In (a) $\lambda = 0.01$ and in (b) $\lambda = 0.05$.



(a) (b)

Figure 4: Comparison of lost track probability versus maaeuvering index for three tracking algorithms (PDA – solid line, 1–Scan Back – dashed line, 2–GPDA – dotted line). In (a) $\lambda = 0.01$ and in (b) $\lambda = 0.05$.



(a) (b)

Figure 5: Comparison of lost track probability versus clutter density for three tracking algorithms (PDA – solid line, 1–Scan Back – dashed line, 2–GPDA - dotted line). In (a) the maneuvering index is 0.15 and in (b) the maneuvering index is 0.25.