From Receiver Operating Characteristic to System Operating Characteristic: Evaluation of a Track Formation System

YAAKOV BAR-SHALOM, FELLOW, IEEE, LEON J. CAMPO, AND PETER B. LUH, MEMBER, IEEE

Abstract—**This paper deals with the application of the multitarget tracking techniques in combination with a Markov chain model to the evaluation of a track formation system based on a composite logic. The goal is to evaluate target track detection probability and false track probability, and to determine the system's target capacity. The novelty of this work is that it computes target track detection probability *in the presence* of false alarms. The key to this is to explicitly account for the effective target detection probability. This probability is reduced by the presence of false alarms, which "mislead" the logic into using measurement association gates smaller than it should have used.**

The *system operating characteristic* (SOC) is introduced. The SOC is the plot of the target track detection probability versus the false track probability, where the values of target detection probability and the false alarm probability are varied to conform to the *receiver operating characteristic* (ROC) corresponding to the system's operating signal-to-noise ratio. This allows choice of the detector's operating point such as to satisfy the overall system requirements.

A procedure to infer the overall system capacity in terms of the minimum spacing of targets that ensures tracking with a certain reliability is also presented. The entire methodology is illustrated with a case study dealing with an actual system.

Nomenclature

P_D Per look target detection probability.

P_{FA} False alarm probability per look per (resolution) cell.

P_{D_t} Target track detection probability (for a single target in the absence of false alarms).

P_{F_t} False track probability (due to false alarms in the absence of targets).

$\tilde{P}_{D_t}$ Target track detection probability (for a single target in the presence of false alarms).

$\tilde{P}_{F_t}$ False track probability (due to false alarms in the neighborhood of a single target).

I. Introduction

THIS paper deals with the application of the multitarget tracking techniques [1]–[3] in combination with a Markov chain model to the design and evaluation of an actual track formation system operating in an environment with a high false alarm rate.

A track initiation procedure using a composite logic, suitable for a high data rate sensor, is considered. A high data rate sensor (relative to the target motion) is one in which target detections that occur in consecutive looks are at most a few resolution cells apart.

The goal is to evaluate a class of track formation logics based upon system requirements, and to determine system target capacity. The system requirements are:

1) the target track detection probability;
2) the false track probability.

An M out of N, denoted as M/N, logic is a test which stipulates that an event (association of a measurement from a "validation gate" [1] to a track) must occur at least M times in N consecutive sampling times. An n block composite logic is a functional composition of n M_i/N_i logics. The two-block logic, denoted as M_1/N_1, M_2/N_2, is the composition $M_2(M_1/N_1)/N_2$.

Blackman [3] presented an analytical technique for correlation (association) performance evaluation that yields the probability of a correct update. However, as pointed out in Reid [9], one cannot make an inference from one update to system performance over several updates due to the dependence of the associations across time (since one bad thing leads to another...). A score function was also considered in [3] for performance evaluation over time using a Markov chain model with fixed target detection probabilities. A Markov chain based technique was presented in [4], [6] to evaluate the probabilities of a target track initiation in the absence of false alarms. The effect of false alarms arising in the validation (association) gate at a given sampling time, when the target is not detected, is to reduce the probability of correctly associating subsequent target-originated measurements [2]. This is equivalent to reducing the target detection probability. The same technique was used in [4], [6] to evaluate the false track probability but under the rather restrictive assumption of fixed size gates.

In contrast to these approaches, the one presented here answers the question of what happens over several sampling times by explicitly accounting for the effect of false alarms on the association of measurements for maneuvering targets.

Holmes [7] described a Bayesian approach consisting of a single block logic called the adaptive track promotion logic. This logic relies on the ratio of the probability of a track being valid to its probability of being invalid given a sequence of detections. In Nagarajan [8] a two-stage composite scheme is used within the framework of the probabilistic data association filter [1]. In stage 1 a decision feature (number of times the signal in a cell exceeded all the reference cells' intensities) is used to determine whether or not a given cell has a target detection. This assumes that the target did not move much during the antenna dwell time. In the second stage, if there is more than one consecutive missed detection, then the track is dropped. For both algorithms [7], [8] performance evaluation at a system level can be done only via simulations.

The purpose of this paper is to present a performance evaluation scheme that does not require simulations and can account in a realistic way for the effect of the clutter. For the simplest tracking scenario of a single target in the absence of false alarms, standard Bernoulli sums can be used to get:

1) P_{D_t}—the target track detection probability in the absence of false alarms, in terms of P_D, the *per look target detection probability*. The other quantities of interest are:

2) $\tilde{P}_{D_t}$—the target track detection probability *in the presence of false alarms*, which is a function of P_D and P_{FA}, the *per*

Manuscript received October 27, 1987; revised October 23, 1988. Paper recommended by Past Associate Editor, C. Y. Chong. This work was supported by Norden Systems under Contract 45-00E6486 and by the Office of Naval Research under Contract N00014-87-K0057.

The authors are with the Department of Electrical and Systems Engineering, University of Connecticut, Storrs, CT 06268.

IEEE Log Number 8932852.

0018-9286/90/0200-0172$01.00 © 1990 IEEE

look per cell false alarm probability;

3) P_{F_t}—the false track probability in the absence of targets, a function of P_{FA};

4) $\tilde{P}_{F_t}$—the false track probability *in the presence of a target*, a function of P_{FA} and P_D.

To obtain the last three quantities above, the so-called common gate-history algorithm (CGH), has been devised. The CGH algorithm greatly reduces computational and storage requirements, and avoids the need for simulations, which would be costly due to low-probability events.

A plot of $\tilde{P}_{D_t}$ versus $\tilde{P}_{F_t}$, called a system operating characteristic (SOC), is made, where values of P_D and P_{FA} are varied to conform to the receiver operating characteristic (ROC) corresponding to the system's operating signal-to-noise ratio. This allows choice of the detector's operating point such as to satisfy the overall system requirements.

Section II presents the association gates for the case study on which the performance evaluation technique is illustrated throughout the rest of the paper. The performance evaluation for the situation of a single target in the absence of false alarms is described in Section III. The difficulties of performance evaluation in the presence of false alarms are discussed in Section IV. The CGH algorithm, which automatically generates the Markov chain states modeling the track formation logic and follows their evolution, is introduced in Section V. This algorithm is then used to evaluate the false track probability in the absence of targets (Section VI), the probability of detecting the track from a single target in the presence of false alarms (Section VII), and the probability of a false track in the neighborhood of a target (also in Section VII). The multiple target situation is considered in Section VIII, where the CGH algorithm is used to infer the overall system capacity in terms of the minimum separation of targets that ensures tracking with a certain reliability.

II. The Association Gates

Detections in the sensor's field of view, assumed here to be a radar, are assumed to be resolved into N_R range and N_F Doppler cells. A measurement association (acceptance) gate [1]–[3] consists of the range and Doppler cells centered at the predicted location of the target. A similar technique can be used with other types of measurements.

The following two assumptions are made.

1) The predicted location of the target in the measurement space (the center of the gate) is assumed here to be the last target detection.

2) The size of the acceptance gate is determined by the target dynamics and depends only on the elapsed time since the last detection.

Denote by $g(\omega_l)$ the size of an acceptance gate that has grown for ω_l looks since the *last* association. The size of the gate is determined by using the target's maximum velocity and acceleration. Thus, any range-Doppler cell that the target could be in (given the cell of the latest detection) will be part of the gate. Therefore, if the target is detected, it will be in its gate.

For the case study considered, Table I gives the postulated gate growth sequence as a function of the gate index ω_l that reflects the targets' maneuverability. These are dimensionless quantities obtained from the application that motivated the present study.

If at any look (sample time) a detection falls in the gate, then it is associated with the detection that established the gate. Unassociated detections are those detections that occur outside acceptance gates. Unassociated detections are used to initiate new track files.

III. Single Target in the Absence of False Alarms

The composite logic track initiation scheme lends itself to a quick and easy performance evaluation when considering a single target in the absence of false alarms. One can use Bernoulli sums to obtain the track detection probability P_{D_t}, in terms of the target detection probability for a single look P_D, for a "window period" T_w, over which the initiation takes place.

TABLE I
Growth of Gates

Number of looks since last detection (gate index)	Range cells on each side	Total range cells	Doppler cells on each side	Total Doppler cells	Gate size
1	1	3	3	7	21
2	1	3	6	13	39
3	2	5	9	19	95
4	2	5	12	25	125
5	2	5	15	31	155
6	3	7	18	37	259
7	3	7	21	43	301
8	4	9	24	49	411
9	4	9	27	55	495
10	4	9	30	61	549
11	5	11	33	67	737

For an M_1/N_1, M_2/N_2 composite logic the track detection probability is given by the following equations:

$$P_{M_1/N_1} = \sum_{i=M_1}^{N_1} C(N_1, i) P_D^i (1 - P_D)^{N_1 - i} \tag{3.1}$$

$$P_{D_t} = P_{M_2/N_2} = \sum_{j=M_2}^{N_2} C(N_2, j)(P_{M_1/N_1})^j (1 - P_{M_1/N_1})^{N_2 - j} \tag{3.2}$$

where

$$C(N, i) = \frac{N!}{(N-i)!i!}. \tag{3.3}$$

The window period in this case is $N_1 N_2 T$ where T is the basic sampling period.

For a 2/8, 4/4 logic (32 looks), Fig. 1 gives a plot of P_{D_t} versus P_D. For example, to obtain $P_{D_t} = 0.8$, one needs $P_D \approx 0.46$. Note the "S-shape" of the curve: medium values on the x-axis are mapped into large values on the y-axis, while small values are mapped into very small ones—this reflects the "discrimination power" of the logic.

Synchronous Versus Asynchronous Logic

If the counting of the composite logic is synchronized to start at a predetermined initial time and if the target is illuminated by the radar beam for the duration of the counting, then the above equations yield an exact relationship between P_D and P_{D_t}. If, however, the logic is triggered by an unassociated detection (initiator), i.e., it is a detection-triggered logic, then the relationship between P_D and P_{D_t} obtained from (3.2) is only approximate. Nevertheless, the difference is of the order of a few percent only. All the remaining sections are for detection-triggered logics.

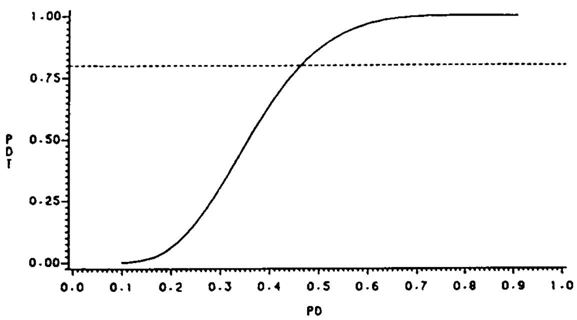


Fig. 1. Track detection probability versus single-look target detection probability for a 2/8, 4/4 composite logic.

Effect of False Alarms

The values obtained from (3.2) are good in a tracking environment where the target tracks are not affected significantly by false alarms (i.e., low probability of false alarms per look per cell, P_{FA}). In this case one would expect that the P_D required to give a desired P_{D_t}, as per (3.2), would be close to the P_D necessary in the actual tracking environment. Thus, (3.2) can be used for preliminary selection of which composite logic to use in more realistic and more complicated analyses.

IV. Evaluation in a Realistic Environment

To analyze tracking environments which contain false alarms one cannot use the Bernoulli sums as was the case for a single target in the absence of false alarms. The false alarms occur on a per look per cell basis. Thus, the probability of having false returns in a target gate depends upon the number of cells in the gate. But the number of cells in a gate depends upon how long ago the detection that established the gate occurred. Thus, the probability of a track with M detections in N looks depends upon the order in which the detections occurred because of the gate growth. Since (3.2) will give the same probability for all tracks with M detections in N looks regardless of the order in which the detections occurred, it cannot be used to analyze the target-in-clutter situation.

One approach in evaluating situations with false returns is to enumerate every possible track sequence that will satisfy the composite logic requirements. Each possible track has an occurrence probability depending upon the track makeup as pertaining to targets, false returns, and no returns. For the case of false alarms in the absence of targets, this approach is not too burdensome computationally. However, when considering the more complicated scenario of a single target in the presence of false returns, the computational load of this (brute force) approach increases tremendously.

Because of the excessive computation required to evaluate the performance in an environment with false alarms, an approximate method, which greatly reduces the computational load, is described in the sequel.

V. The Common Gate-History Algorithm

For the scenario of a single target in the presence of false alarms, a track can have any one of the following makeups:

1) detections from the target only;
2) detections from the target and from false alarms;
3) detections from false alarms only.

Given these possible track makeups, the evaluation of a two-stage composite logic of the type M_1/N_1, N_2/N_2 can be done based on a sufficient statistic of a track, which consists of:

1) the number of time steps since the *last detection* [target (*TGT*) or false alarm (*FA*)], denoted as ω_l;

2) the number of time steps since the *last target detection*, denoted as ω_{lt}; and

3) the count of detections within stage 1 of the logic (logic status), denoted as λ (this holds for the 2/8, 4/4 logic where there is only one element for logic status; for other composite logics more than one logic status element might be required).

Under assumptions AS1 and AS2 to be introduced later, the *gate-history vector*

$$\omega = [\omega_l, \omega_{lt}, \lambda] \tag{5.1}$$

is the sufficient statistic for the 2/8, 4/4 logic. This vector evolves according to a Markov chain. The states and transition probabilities can be set up manually (and then the chain evolution follows from standard equations) or, in an alternative, automated manner to be discussed next.

Using this sufficient statistic the algorithm sequentially generates the feasible values of ω and lumps all tracks according to a *common* ω. In view of this, the procedure is called the *common gate-history* (CGH) algorithm. This procedure and the accompanied recursive calculation of the probabilities of the corresponding gate histories is significantly simpler than the manual setup of the corresponding Markov chain state-space and transition matrix. Thus, the CGH *generates automatically* the states of the Markov chain and follows their evolution.

Once a track is started, it is continued by observing the events that occur within its acceptance gates. For the purpose of evaluation the following events are considered.

A_1 No detection.
A_2 Target detection.
A_3 False alarm.
A_4 Target detection and a false alarm.

The events with more than one false alarm in a gate are assumed to occur with low probabilities assuming P_{FA} is low enough (10^{-3} or less for the gates presented in Table I). If one has to work with a larger value of P_{FA}, then the algorithm should be augmented to include more events. This is summarized by the following assumption.

AS1: Events $A_1, \cdots, A_4$ are exhaustive.

The CGH evaluation algorithm works by continuing each track at each look according to the four possible events: the track probability at look i is multiplied by the probability of the event that continues this track at look $i+1$ (to give the updated track probability at look $i+1$) and then lump (add the track probabilities of) all the tracks with a common ω. This leads to a Markov chain with a manageable number of states, which are generated automatically. The Appendix illustrates how the lumping of tracks is accomplished.

Probabilities of the Gate Events

The probabilities of the events $A_1, \cdots, A_4$ are given next. The gate size (number of range and Doppler cells) for an ω_l-step prediction is denoted as $g(\omega_l)$.

The "no detection" probability in gate $g(\omega_l)$ is

$$P\{A_1\} = \left(1 - \frac{g(\omega_l)}{g(\omega_{lt})} P_D\right)(1 - P_{FA})^{g(\omega_l)} \tag{5.2}$$

where the ratio of actual gate $g(\omega_l)$ to the size it should have been, $g(\omega_{lt})$, is used to reduce P_D to a lower "effective" value. This assumes a uniform spatial distribution of the location of a target in a gate.

The probability of the target only detected is

$$P\{A_2\} = \frac{g(\omega_l)}{g(\omega_{lt})} P_D (1 - P_{FA})^{g(\omega_l)}. \tag{5.3}$$

The probability of one false alarm and no target detection in gate $g(\omega_l)$ is (assuming the probability of more than one false alarm to be negligible)

$$P\{A_3\} = (1 - (1 - P_{FA})^{g(\omega_l)}) \left(1 - \frac{g(\omega_l)}{g(\omega_{lt})} P_D\right). \tag{5.4}$$

Finally, the probability of one false alarm and target detection in gate $g(\omega_l)$ is

$$P\{A_4\} = (1 - (1 - P_{FA})^{g(\omega_l)}) \left(\frac{g(\omega_l)}{g(\omega_{lt})} P_D \right). \quad (5.5)$$

Description of the Algorithm

For event A_1 the updating of ω is done by incrementing ω_l and ω_{lt} by 1 (the last detection and last target detection are one more look further back in time) and by not changing λ. For event A_2, since there has just been a target detection, then $\omega_l = \omega_{lt} = 1$ and λ is incremented by 1. For event A_3, since there has just been a false detection, $\omega_l = 1$; however, ω_{lt} gets incremented by 1 because the detection was not from the target and λ is incremented by 1.

For event A_4 there is a simultaneous occurrence of a false alarm and the target detection. Assuming that a track splitting algorithm is used, then event A_4 yields a true track as well as a false track. In view of this, assumptions AS2a and AS2b are used.

AS2a: In computing the false track probability $\tilde{P}_{F_t}$, tracks continued with event A_4 (one FA and target detected) are lumped with the tracks that continue with event A_3 (one FA only).

AS2b: In computing the target track probability $\tilde{P}_{D_t}$, tracks continued with event A_4 are lumped with the tracks that continue with event A_2 (target detection only).

With these assumptions the CGH algorithm under AS2a will give realistic false track probabilities while with AS2b the target track probability will be realistic.

For the 2/8, 4/4 scheme, λ is used to check for the satisfaction of the first stage in the logic (2/8) requirement. The 2/8 requirement covers eight looks which are made in a time span denoted by T_d. A track file (sequence of detections) is initialized with a new (unassociated) detection for a detection triggered logic. Upon initialization, λ is set to 1. A track file with no detections within the *current* T_d will have $\lambda = 0$. A track file with one detection within the current T_d will have $\lambda = 1$. A track file with two or more detections within the current T_d will have $\lambda = 2$, at which point the counter is "saturated." This is because track files with $\lambda = 2$ have satisfied the 2/8 requirement for the current T_d. At the end of the current T_d only those track files with $\lambda = 2$ will be allowed to continue to the subsequent T_d. At the start of the next T_d period, λ is reset to zero.

The 4/4 requirement is satisfied by track files which have satisfied four consecutive 2/8 requirements. For logics other than the 2/8, 4/4 composition, different schemes may be required to check for logic acceptance. With suitable modifications to the scheme used for the above two-stage composition, any M_1/N_1, M_2/N_2 composite logic can be evaluated. This methodology can be extended to logics with more than two stages at the cost of increasing the dimension of the sufficient statistic.

Summary

The CGH algorithm generates automatically the states of the Markov chain modeling the track formation. The inputs P_D and P_{FA} are used to calculate the probabilities of track continuation events A_1 through A_4. A track probability is found by multiplying the probabilities of the events that make up the track.

At each look the tracks are lumped by adding the track probabilities of the track files with a common gate history vector ω. This keeps the number of possible states of the Markov chain within reasonable computation limits—the algorithm has been run on an IBMPC-AT.

VI. False Tracks in the Absence of Targets

The common gate-history algorithm can be used to evaluate the relationship between the probability of a false alarm per look per cell P_{FA} and the false track probability P_{F_t} for composite logic track initiation schemes. This can be done by setting P_D equal to zero.

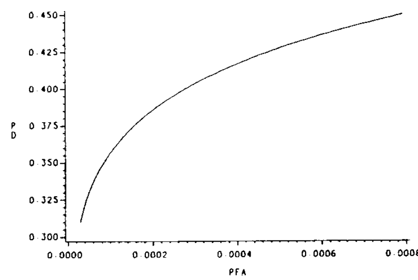


Fig. 2. The ROC curve for SNR=9 dB.

In this case only pure false tracks (tracks with only false detections) exist with nonzero probability. Note that the numerical relationship between P_{FA} and P_{F_t} obtained in this way incorporates the approximation that the low probability events of two or more false returns in a gate can be neglected. This is satisfactory if P_{FA} is 10^{-3} or less.

After the preliminary relationship between P_D and P_{D_t} is obtained for a particular logic, as shown in Section III, use is made of the ROC curve with P_D determined from the desired value of P_{D_t}, to yield P_{FA}.

For the sake of illustration, consider the ROC curve generated according to the Swerling I detection model [5]

$$\text{SNR} = 10 \log \left(\frac{\log P_{FA}}{\log P_D} - 1 \right) \text{dB}. \quad (6.1)$$

Fig. 2 shows a plot of the ROC with SNR = 9 dB. The P_{FA} value obtained from the ROC is used to obtain P_{F_t}, as discussed above, using the CGH algorithm. If an acceptable value of P_{F_t} is found, then the analysis can proceed to the more complicated case of a single target in the presence of false alarms. Several candidate logic schemes should be considered if possible. If an acceptable value of P_{F_t} does not exist for this particular logic, then another logic scheme is to be evaluated.

This type of analysis requires caution because there may not exist a logic which yields an acceptable P_{D_t} and P_{F_t} for a given SNR. It is because of this that the CGH algorithm is valuable since it is quick and requires no simulations. Simulations would be very costly because of the large number of Monte Carlo runs needed to evaluate small probability events. Several logic schemes can be analyzed in a short time and, if every scheme tried is unacceptable, then something else must be done such as relaxing the requirements on P_{D_t} and P_{F_t}, or a higher SNR is necessary.

For the 2/8, 4/4 logic, Fig. 3 gives a plot of P_{F_t} versus P_{FA} obtained from the CGH algorithm with $P_D = 0$, with the gate growth sequence of Table I, and the ROC curve (6.1) with SNR = 9 dB.

For example, as can be seen from Fig. 2, the detection threshold setting corresponding to $P_D = 0.46$ (obtained in Section III) yields $P_{FA} = 0.9 \cdot 10^{-3}$ which, in turn, gives, from Fig. 3, $P_{F_t} = 10^{-6}$. This latter figure is the false track probability (starting from a resolution cell) in a region of the surveillance space where there is no target, with the above detection threshold setting.

VII. Single Target in the Presence of False Alarms

Once a few candidate logic schemes have been found one has to consider the more complicated scenario of a single target in the presence of false alarms. With a range of P_D and P_{FA} selected from the ROC as the inputs to the CGH algorithm, one can obtain corresponding values of:

- $\tilde{P}_{D_t}$, the *target track probability in the presence of false alarms*; and

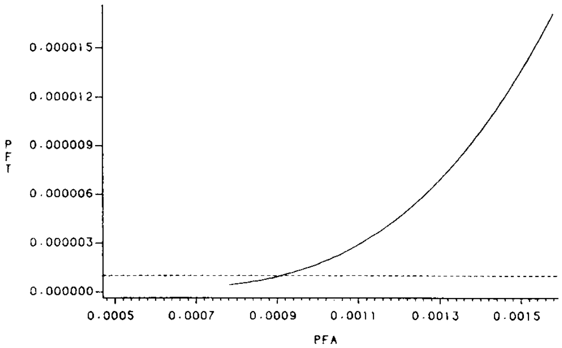


Fig. 3. P_{F_t} versus P_{FA} for the 2/8, 4/4 composite logic.

- $\tilde{P}_{F_t}$, the *false track probability in the neighborhood of a target.*

For the purpose of evaluation, it is necessary to define a target track. For the 2/8, 4/4 logic, a *target track* is defined as any sequence of detections that satisfies the logic conditions and at least one detection per eight looks is from the target. Any other sequence of detections that satisfies the logic requirements will be called a false track. The requirement to have the target at least once in eight looks has been stipulated so that a tracking filter will receive at least this much target information. Other target track definitions might be also appropriate.

The value of $\tilde{P}_{D_t}$ will differ from P_{D_t} obtained using Bernoulli sums in at least two respects. As defined, the target track may have the target detection only once during eight looks, whereas for the case of a single target in the absence of false alarms a target track must have the target at least twice during an eight look span (2/8 requirement). Also the target track may contain false detections which occur at a different rate than the target detections. However, since the false alarm rate is low, the effects of false alarm contamination on target tracks should be negligible.

The value of $\tilde{P}_{F_t}$ obtained in this evaluation of a single target in the presence of false alarms can differ significantly from P_{F_t} obtained in the absence of targets. This is due to the following phenomenon: in the neighborhood of target tracks, false tracks can be "spawned" from a target track. Therefore, false tracks can occur with a higher probability due to target detections.

Caution must be used when interpreting the false track results: P_{F_t} obtained from the analysis of Section VI will be valid only in regions of the tracking environment where there are no targets. The value of $\tilde{P}_{F_t}$, which pertains only to the vicinity of targets, can be significantly different.

Evaluation of a Composite Logic for a Single Target in the Presence of False Alarms

A range of values for P_D and P_{FA} are selected from the ROC and used as inputs to the CGH algorithm.

To obtain $\tilde{P}_{D_t}$, the CGH algorithm:

1) computes the probability of all detection sequences that satisfy the logic using assumption AS2b ($TGT\&FA \rightarrow TGT$);

2) separately stores the probability of target tracks according to the definition adopted.

To obtain $\tilde{P}_{F_t}$, the CGH algorithm:

1) computes $\tilde{P}_{D_t}$ under assumption AS2a ($TGT\&FA \rightarrow FA$);

2) subtracts it from the total probability of all detection sequences that satisfy the logic.

A set of $\tilde{P}_{D_t}$ and $\tilde{P}_{F_t}$ values are plotted against each other to form the *system operating characteristic* curve (SOC). The resulting SOC can then be used for deciding:

1) which logic scheme will best meet the system requirements for target and false track probabilities;

2) which point on the ROC curve (detection threshold) should be used; and

3) whether the system SNR can meet the system requirements.

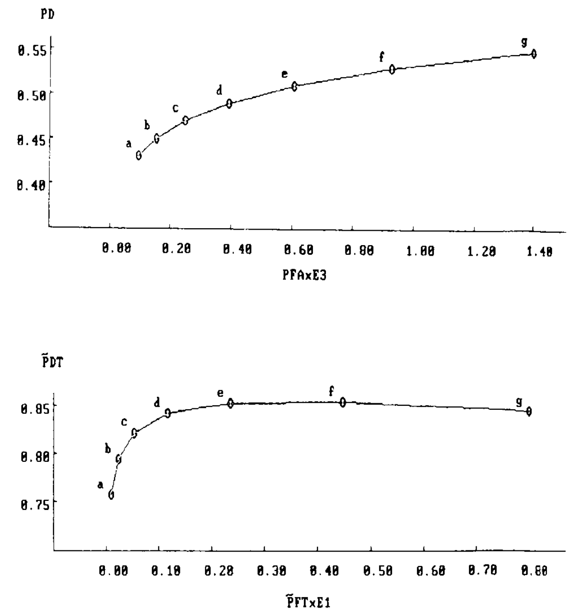


Fig. 4. The ROC for SNR = 10 dB and the SOC for the 2/8, 4/4 logic.

A plot of the ROC for SNR = 10 dB and the corresponding SOC for the 2/8, 4/4 logic are given in Fig. 4. Fig. 5 presents the ROC and SOC plots for 12 dB SNR, and the same logic.

The data points are alphabetically labeled to indicate corresponding points on the two plots. Notice that, in the SOC, as P_D and P_{FA} increase, both $\tilde{P}_{D_t}$ and $\tilde{P}_{F_t}$ increase. However, as P_{FA} becomes greater than 10^{-3}, Fig. 4 indicates that $\tilde{P}_{D_t}$ starts to decrease.

This decrease in $\tilde{P}_{D_t}$ is explained as follows. From (5-3) one can see that the *effective* P_D for event A_2 (target only) depends upon the ratio of $g(\omega_l)$ to $g(\omega_{lt})$. An increase in P_{FA} decreases $g(\omega_l)$ because false returns are occurring more often. Although an increase in P_D decreases $g(\omega_{lt})$ as well, however, since the rate of increase of P_{FA} is much larger than the rate of increase of P_D (from point b onwards, as seen from the ROC of Fig. 4), the ratio of $g(\omega_l)$ to $g(\omega_{lt})$ decreases. Thus, the *effective* P_D goes down and the target track probability decreases.

If one wants $\tilde{P}_{D_t}$ to be 0.8 or greater, then, as seen from Figs. 4 and 5, a minimum P_D of 0.45 is needed. However, since there is a sharp knee in the SOC, it would be better to have a P_D of 0.5 or greater.

From the SOC in both figures one can see that, in order to have a higher target track probability, one must tolerate a higher false track probability. Thus, if target accuracy is important and there are enough computer resources available so that extra false tracks are not a problem, then $\tilde{P}_{D_t}$ can be made large. Otherwise, $\tilde{P}_{D_t}$ will have to be assigned a value so that $\tilde{P}_{F_t}$ is not too large.

Overall System Load

Note that $\tilde{P}_{F_t}$ is the probability of a false track in the vicinity of a true track. The quantity P_{F_t} is the probability of a false track starting from a resolution cell without "help" from a neighboring target. For a practical situation one can have, say, $\tilde{P}_{F_t} \approx 10^{-2}$ and $P_{F_t} \approx 10^{-6}$.

The expected number of false tracks in the surveillance region is then

$$E(F_t) = P_{F_t} N_c + \tilde{P}_{F_t} N_t \quad (7.1)$$

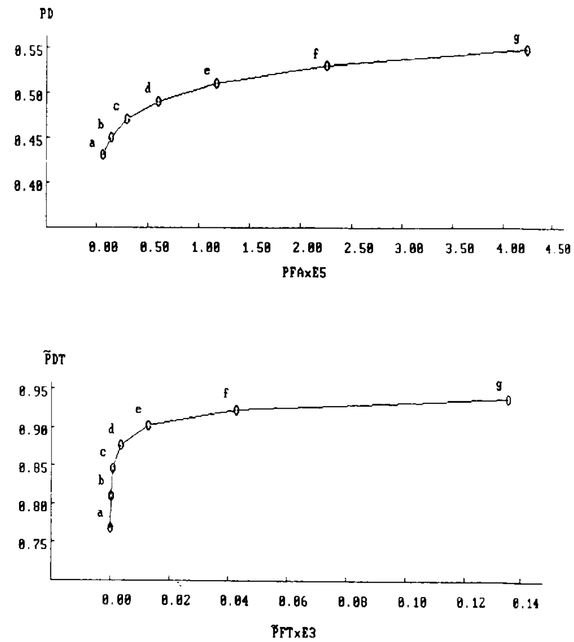


Fig. 5. The ROC for SNR = 12 dB and the SOC for the 2/8, 4/4 logic.

where:

N_c—number of resolution cells in the field of view;

N_t—number of targets.

The above can be used to obtain the computational load for the environment under consideration.

VIII. MULTIPLE TARGETS AND FALSE ALARMS

The final goal is to evaluate overall system capacity—the number of targets per unit area that can be handled. This is done by setting a "noninterference condition" for neighboring targets with a certain (high) probability. This can then be used to infer a target density for which, in the presence of false alarms, there will be no overlap with a certain (different, but still high) probability. This target density can then be seen as a system capacity subject to a noninterference condition between neighboring targets.

The approach consists of:

1) evaluation of the probability mass function (pmf) of the gate index ω_l (number of looks since the last detection, which determines the gate size) for a target track in the presence of false alarms; and

2) setting a "nonoverlap condition" for the validation gates of neighboring targets, which guarantees no interference.

Using the CGH algorithm, the pmf of the gate index for a single target with false alarms is computed. The probability of a gate index being equal to a given number (for a particular look) is found by summing the track probabilities of all tracks (both target and false tracks) with ω_l (defined in Section V) equal to this number, then dividing this sum by the total probability of all tracks.

Fig. 6 gives the pmf versus gate index for the 2/8, 4/4 logic for $P_D = 0.45$ and $P_{FA} = 10^{-3}$. Fig. 6(a) shows it for look 8 and Fig. 6(b) for look 32. The pmf for looks 16 and 24 are essentially the same as for look 32 and are not shown. For these particular looks (8, 16, 24, 32) the largest gate index is seven because tracks with gate index larger than seven would not have passed the 2/8 requirement, and thus are discarded.

In these figures one can see that the pmf goes down by a factor of approximately 2 as the gate index increases by one. This decrease is mainly due to the P_D value, which is close to

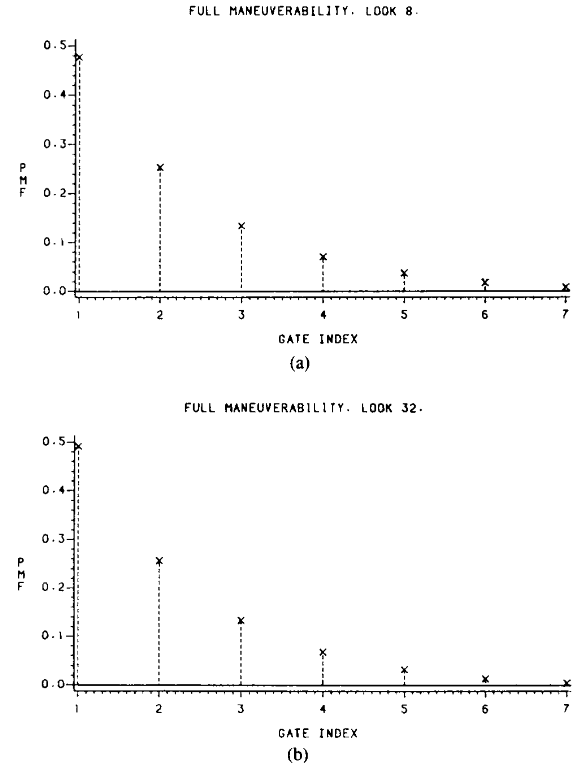


Fig. 6. Gate index pmf. (a) Full maneuverability—Look 8. (b) Full maneuverability—Look 32.

1/2. For a single target in the absence of false alarms one would get the pmf values decreasing by exactly a factor of $1/(1 - P_D)$ for an increase of gate index by 1.

Note that at look 8 the gate size pmf versus gate index has a slightly heavier tail than at look 32. This shift of probability mass from the tail to the front end, at looks 16, 24, and 32, is due to the false tracks dying out after the first period T_d (8 looks).

Inference on Minimum Target Separation

For a given target separation one can compute the probability of gate overlapping using the gate size pmf as shown next.

To consider the probability of overlap in range and Doppler, first note that if there is overlap in one coordinate but not in the other, then there is no gate overlap. Thus, there must be overlap in both coordinates for a gate overlap to occur.

The following notations will be used in the sequel.

i $\triangleq$ gate index (number of looks since last detection).

$\rho(i)$ $\triangleq$ number of range cells on each side of a gate, detailed in Table I. For example, with $i_A = 3$, one has $\rho(i_A) = 2$, i.e., target A has a validation gate spread in range of $2\rho + 1 = 5$ cells.

$\delta(i)$ $\triangleq$ number of Doppler cells on each side of a gate. For example, with $i_B = 2$, one has $\delta(i_B) = 6$, i.e., target B has a gate spread in Doppler of $2\delta + 1 = 13$ cells.

$x \vee y$ $\triangleq$ max (x, y).

s_r $\triangleq$ range separation (number of range cells between the two targets, i.e., if one is in cell j, the other is in cell $j + s_r$).

s_d $\triangleq$ Doppler separation.

With s_r and s_d given, the probability of two targets, A and B, overlapping in range and Doppler for one look is

$$P_{OV} = \sum_{i_A=1}^{i_{max}} \sum_{i_B=i_0}^{i_{max}} P(i_A)P(i_B) \tag{8.1}$$

TABLE II
PROBABILITY OF NO OVERLAP OVER LOOKS 9–16

s_d \ s_r	0	2	4	6
36	0.95	0.95	0.95	0.96
24	0.56	0.56	0.56	0.95
12	0.01	0.01	0.30	0.95
0	0.00	0.00	0.30	0.95

where $P(i)$ is the probability of gate index equal to i, and

$$i_0(s_d, s_r, i_A) = i_r(s_r, i_A) \vee i_d(s_d, i_A) \quad (8.2)$$

$$i_r(s_r, i_A) = \arg\min_i [\rho(i_A) + \rho(i) \geq s_r] \quad (8.3)$$

$$i_d(s_d, i_A) = \arg\min_i [\delta(i_A) + \delta(i) \geq s_d]. \quad (8.4)$$

In the above i_0 is the minimum gate index for target B such that, for given separation (s_d, s_r) and gate index i_A of target A, the gates overlap.

Once P_{OV} is found, then $1 - P_{OV}$ gives the probability of *no overlap* for one look. Thus, the product of $1 - P_{OV}$ over several looks gives the no-overlap probability of two adjacent targets over these looks.

Table II gives the probability of no overlap over looks 9–16. Thus, for a Doppler separation of 36 cells or more, regardless of the range separation, there will be a high no-overlap probability between adjacent targets over looks 9–16.

Notice also that for a range cell separation of 6 or more, there will be a high no-overlap probability regardless of the Doppler separation. Thus, once there is enough separation in either range or Doppler, then a high no-overlap probability is guaranteed.

IX. SUMMARY AND CONCLUSION

The preliminary evaluation of the track initiation based on a composite two-stage logic in a surveillance system consisted of the evaluation of:

1) the target track detection probability in the absence of false alarms, in terms of the per look target detection probability P_D;

2) the false track probability in the absence of targets, a function of the per look per cell false alarm rate P_{FA}.

This was followed by evaluation of the situation with target and false alarms, with the quantities of interest being:

3) the target track detection probability in the presence of false alarms, which is a function of P_D and P_{FA};

4) the false track probability in the presence of a target, a function of P_{FA} and P_D.

The first quantity, the target track detection probability in the absence of false alarms, could be obtained using standard Bernoulli sums. To obtain the remaining three quantities, the so-called common gate-history algorithm (CGH) was used. The CGH algorithm generates automatically the states of the Markov chain modeling the track formation process. It greatly reduces computational and storage requirements and avoids the need for simulations, which would be costly due to low-probability events. The assumptions of the CGH algorithm are as follows.

1) The validation gate size (number of resolution cells) depends on the target maneuverability and the elapsed time since the last detection.

2) The probability of more than one false alarm in a gate is negligible. The CGH algorithm has the feature of accounting for the reduced effective P_D due to the false alarms.

The *system operating characteristic* (SOC) was introduced as the plot of the target track detection probability in the presence of false alarms versus the false track probability, where values of P_D and P_{FA} are varied to conform to the ROC corresponding to the system's operating SNR. This allows choice of the detector's operating point such as to satisfy the overall system requirements.

The CGH algorithm can be used to infer the *overall system capacity* in terms of the maximum number of targets that can be tracked with a certain reliability—this was done by setting a "nonoverlap" condition between the gates of neighboring targets.

Future work can be done to account for more false alarms in the gate and incorporate the effect of polynomial smoothing on the data points to reduce the gate sizes.

t_1	t_2^-	t_2	t_3^- before lumping	t_3^- after lumping
TGT →	ω=[1,1,1]	A_1 (No detection):	ω=[2,2,1]→(1)	(1): ω=[2,2,1]
		A_2 (TGT):	ω=[1,1,2]→(3)	
		A_3 (FA):	ω=[1,2,2]→(2)	(2): ω=[1,2,2]
		A_4 (FA and TGT):	ω=[1,2,2]→(2)	(3): ω=[1,1,2]
FA →	ω=[1,F,1]	A_1 (No detection):	ω=[2,F,1]→(4)	(4): ω=[2,F,1]
		A_2 (TGT):	ω=[1,1,2]→(3)	
		A_3 (FA):	ω=[1,F,2]→(5)	
		A_4 (FA and TGT):	ω=[1,F,2]→(5)	(5): ω=[1,F,2]

Fig. 7. CGH algorithm—illustration of track continuation and lumping by sufficient statistics (common ω) for evaluation of $\tilde{P}_{F_t}$.

APPENDIX

THE AUTOMATIC GENERATION OF THE MARKOV CHAIN

The generation of the chain states by lumping of the tracks with common gate-history is illustrated in the example of Fig. 7 for a detection-triggered logic. The status is shown at look 1 (time t_1), just prior to look 2 (time t_2^-), at t_2, and prior to look 3 (time t_3^- before and after lumping).

At look 1 the two possible events that could start a track file are a detection from the target or a false alarm. Notice that, for the track with the target detection, $\omega_l = 1$ because there is a detection at this look, and $\omega_{lt} = 1$ because the detection was from the target, and $\lambda = 1$ because there has been one detection so far. Note that $\omega_{lt} = F$ for the track started with an FA. Here F is a flag indicating that there has not been a target detection in this track file. This is later used to evaluate the probability that a track started by an FA will "pick up" a target.

The two track files which exist at look 1 are continued at look 2 by the occurrence of the events A_1 through A_4. The probabilities of events A_1 through A_4 are computed using P_D and P_{FA}.

From Fig. 7 one can see that after the second look there are eight different tracks. However, some of these tracks have the same sufficient statistics and, after lumping, there are only five track files left. At look 3 if the tracks are not lumped according to the sufficient statistics there will be 32 different track files, but after lumping, only 9 track files remain.

Thus, by lumping tracks according to a common ω, the number of track files is reduced far below the number of possible distinct tracks. For the 2/8, 4/4 scheme, over the first eight looks there are 6544 unique tracks that satisfy the 2/8 logic. However, with the CGH algorithm, each of these 6544 tracks will correspond to one of *only 35 possible* ω. This number of states was determined by the CGH algorithm when automatically generating the chain.

ACKNOWLEDGMENT

Stimulating interaction with M. Greenspan, R. Guarino, D. Ariel, and M. Tobin is gratefully acknowledged.

REFERENCES

[1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

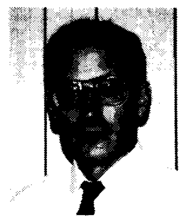
[2] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Principles and Techniques*, UCLA Extension and University of Maryland Short Course Notes, 1987/1988.

[3] S. S. Blackman, *Multiple-Target Tracking with Radar Applications.* Dedham, MA: Artech House, 1986.
[4] F. R. Castella, "Sliding window detection probabilities," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-12, pp. 815–819, Nov. 1976.
[5] J. V. DiFranco and W. L. Rubin, *Radar Detection.* Dedham, MA: Artech House, 1980.
[6] A. Farina and F. A. Studer, *Radar Data Processing* (Vol. I: Introduction and Tracking, Vol. II: Advanced Topics and Applications). Hertfordshire, England: Letchworth and New York: Wiley, Research Studies Press, 1985.
[7] J. E. Holmes, "Development of algorithms for the formation and updating of tracks," in *Proc. IEE 1977 Int. Radar Conf.*, London, Oct. 1977, pp. 81–85.
[8] V. Nagarajan, R. N. Sharma, and M. R. Chidambara, "An algorithm for tracking a maneuvering target in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, pp. 560–573, Sept. 1974.
[9] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 843–854, Dec. 1979.

Yaakov Bar-Shalom (S'63–M'66–SM'80–F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion—Israel Institute of Technology, Haifa, Israel, in 1963 and 1967, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently, he is Professor of Electrical and Systems Engineering at the University of Connecticut, Storrs. His research interests are in estimation theory and stochastic adaptive control and he has published over 100 papers in these areas. He has been consulting to numerous companies, and originated a series of multitarget-multisensor tracking short courses offered via UCLA Extension, University of Maryland, at government laboratories, private companies, and overseas. He has also developed the interactive software MULTIDAT for automatic track formation and tracking of maneuvering targets in clutter.

Dr. Bar-Shalom has been elected Fellow of the IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has coauthored the book *Tracking and Data Association* (New York: Academic, 1988). During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. From 1983 to 1987 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and is currently a member of the Board of Governors of the IEEE Control Systems Society. In 1987 he received the IEEE CSS Distinguished Member Award.



Leon J. Campo was born in Stafford Springs, CT, on March 6, 1957. He received the B.S. and M.S. degrees in electrical engineering from the University of Connecticut, Storrs, in 1984 and 1986, respectively.

Currently, he is working towards the Ph.D. degree in electrical engineering and is a Research Assistant in the Department of Electrical and Systems Engineering, University of Connecticut, Storrs. His research interests include multiple-target tracking, stochastic estimation, and stochastic control. He worked as a Coop engineer with IBM in Burlington, VT, and has done consulting for Norden Systems. From 1976 to 1980 he was in the U.S. Air Force.

Peter B. Luh (S'76–M'80) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, Republic of China, in 1973, the M.S. degree in aeronautics and astronautics engineering from the Massachusetts Institute of Technology, Cambridge, in 1977, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1980.

Since 1980 he has been with the University of Connecticut, Storrs, where he is currently an Associate Professor in the Department of Electrical and Systems Engineering. His major research interests include hierarchical planning and control of large scale systems, schedule generation and reconfiguration for manufacturing schedule generation and reconfiguration for manufacturing systems, distributed decisionmaking, game theory, and multitarget tracking. He has been a principal investigator and consultant to many industry and government funded projects in the above areas, and has published about 80 papers.

Dr. Luh is a member of Sigma Xi. He is an Associate Editor of the IEEE Transactions on Automatic Control, won the Best Paper Award of the 1987 Joint Command and Control Research Symposium, and has served on Program Committees and Operating Committees of several major national, international, and intersociety conferences. He is listed in *Who's Who in Engineering* and *Who's Who in the East*.