

Neural Network Directed Bayes Decision Rule for Moving Target Classification

XI YU

MAHMOOD R. AZIMI-SADJADI, Senior Member, IEEE
Colorado State University

In this paper, a new neural network directed Bayes decision rule is developed for target classification exploiting the dynamic behavior of the target. The system consists of a feature extractor, a neural network directed conditional probability generator and a novel sequential Bayes classifier. The velocity and curvature sequences extracted from each track are used as the primary features. Similar to hidden Markov model (HMM) scheme, several hidden states are used to train the neural network, the output of which is the conditional probability of occurring the hidden states given the observations. These conditional probabilities are then used as the inputs to the sequential Bayes classifier to make the classification. The classification results are updated recursively whenever a new scan of data is received. Simulation results on multiscan images containing heavy clutter are presented to demonstrate the effectiveness of the proposed methods.

Manuscript received December 26, 1998; revised October 9, 1999; released for publication October 18, 1999.

Refereeing of this contribution was handled by T. Busch and V. Skormin.

IEEE Log No. T-AES/36/1/02494.

This work was funded by the Optoelectronic Computing Systems (OCS) Center at Colorado State University, under NSF/REC Grant 9485502.

Authors' address: Signal/Image Processing Laboratory, Dept. of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, E-mail: (azimi@engr.colostate.edu).

0018-9251/00/\$10.00 © 2000 IEEE

I. INTRODUCTION

The problem of detection and classification of moving targets from IR, radar and sonar platforms has attracted a lot of attention in recent years [1–25]. This involves discrimination among different types of moving targets as well as the characterization of background clutter. Several factors contribute to make the detection and classification processes a very complex problem. These include nonrepeatability and variation of the moving target signature for different targets, extremely low signal-to-noise ratio, competing clutter, and lack of any *a priori* knowledge about the shape, geometry, and dynamics of the targets. Consequently, efficient and robust detection and classification schemes are needed to solve this complex problem.

Several different schemes [3–13] for moving target detection have been developed and applied to IR, radar and sonar imagery data. These include spatio-temporal filtering [3, 4], maximum likelihood (ML) estimation [5, 12], recursive Kalman filtering [6–11] and neural network-based methods [13]. In [4], a 3-D spatio-temporal filtering scheme is developed. The operations are done in the frequency domain by passing the image through a bank of directional filters each tuned to extract line features of the tracks with certain orientations. Bar-Shalom, *et al.* [9, 10] used two different centroid-based measurements to track targets in forward looking IR images. In a more recent paper [11], Shertukde and Bar-Shalom extended this approach by using the joint probabilistic data association (JPDA) in conjunction with a Kalman state estimator. The scheme in [12] uses a track-before-detect procedure to detect multiple targets in presence of false measurements. The methodology is based upon ML estimation as the measurement model is nonlinear. Roth [13] developed a neural network-based scheme for detection of straight line tracks in background clutter. A Hopfield network was trained to implement the optimum post-detection target track receiver operation.

Over the years, several feature extraction schemes have been developed for target classification. In [14], polarimetric target signatures are derived and plotted for simple objects using the complex radar cross-section matrix. The target classification is made based on the polarimetric signatures and the characterization of the parameters derived from them. Walton and Jouny [15] showed that the scattering characteristics of a radar target can be specified by its impulse response and there is a clear relationship between the actual location of specific scattering mechanisms and the time they appear in the impulse response. However, complex targets often have multiple scattering interactions. The bispectrum displays the specifics of the interactions of both single and multiple reflection mechanisms.

Based on the bispectral signatures the classification of scale models of commercial aircraft was then demonstrated. In [16], the frequency-selective fading channel model was employed to characterize the effects of the multipath propagation medium and evaluate the performance of radar target classification systems. The relative amplitude, relative phase, and absolute amplitude measurements were selected as the features to make the classification. In [17], the radar backscatter signal measurements are first represented as a set of down range scattering centers parameterized by the Prony modeling technique. The relative range, size and possibly polarimetric shape of the resulting parametric scattering centers are then used to form a description of the “structure” of the target. Finally, these “structural descriptions” of the measured backscatter signal are used to classify the targets. Zyweck and Bogner [18] used high resolution range profiles to classify commercial aircrafts. A large bandwidth is used to provide high resolution in range for better target discrimination. On the other hand, coherence makes cross-range resolution and radar image possible. In [19], a multidimensional sensor suite consisting of a laser radar and a passive IR sensor is developed and evaluated for detecting and identifying ships at long ranges from an airborne platform. The passive IR sensor detects targets by taking advantage of the high target-to-background contrast and the ability of the sensor to track over a wide field of view. The information content of the range profile of the ship is then exploited for target identification. In [20], the classification of high-range resolution (HRR) radar signatures using multiscale features is employed. The method in [21] employs neural networks namely the self-organizing map (SOM) and learning vector quantization (LVQ) for automatic target recognition (ATR) from HRR target signatures. Hauter, *et al.* [22] considered the problem of target classification using synthetic aperture radar (SAR) polarizations from a Bayesian decision point of view. They investigated the optimum design of a data fusion structure given that each classifier makes a target classification decision for each polarization channel. In [23], both neural networks and conventional classification schemes were used to determine the class of a target from the inverse SAR (ISAR) imagery acquired during reconnaissance missions. In [24, 25] a 2-D pattern-matching algorithm rejects cultural clutter false alarms (i.e., nontargets) and classifies the remaining detections by target type e.g., tank, armored personnel carrier, or howitzer using a SAR target recognition system.

The above target classification schemes generally assume some *a priori* knowledge of signatures of the targets to perform the classification. However, when only a multiscan radar, sonar, or IR image is available, no *a priori* information about the target signature

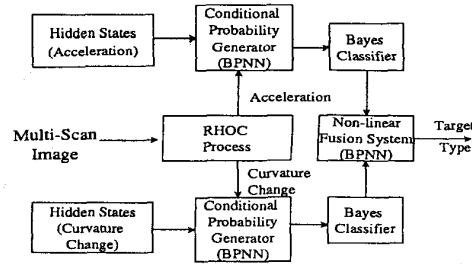


Fig. 1. Block diagram of classification scheme for two-target case.

can be assumed. A new method referred to as neural network directed Bayes classifier is developed here. Fig. 1 shows the overall block diagram of the proposed classification scheme for two-target case. Since a moving target builds a spatial-temporal track in the 3-D space and there exists spatial-temporal correlation between consecutive target observation points, the recursive high order correlation (RHOC) scheme [1, 2] is first used to detect the targets in an extremely cluttered environment. After detecting the target tracks, two sets of features namely acceleration and curvature changes are extracted from each track. Based on each feature type, several hidden states are generated [26] to train the neural network which in turn generates the conditional probability of occurring the hidden states given the observations. The output of this network is used as the input to a new sequential Bayes classifier. A nonlinear fusion system is then employed to combine both of the classifiers results to make the final classification decisions. Simulated target tracks for three types of targets were generated using first order Markov models. The classification performance under different conditions demonstrate the effectiveness and robustness of the proposed scheme.

This paper is organized as follow. Section II briefly reviews the RHOC process [1, 2]. Sections III and IV give the details on the conditional probability generator and the sequential Bayes classifier subsystems, respectively. Simulation results are presented in Section V.

II. TARGET DETECTION AND CLUTTER REJECTION USING RHOC SCHEME

A multiscan image is actually obtained by stacking several 2-D images collected at some time intervals by a sensor whose field of view is fixed with respect to the background. This yields a 3-D image in the coordinates (x, y, t) where (x, y) are the spatial variables and t represents the time. A moving target in several consecutive scans forms a track or a signature in this 3-D image.

The scan data is converted to binary images by mapping any sensor return with intensity greater than

zero to one. This is done to prevent the possibility of losing dim targets. Now stacking several such binary images for different time scans gives a 3-D image which can be modeled [1, 2] by

$$F(x, y, t_n) = S(x, y, t_n) + [1 - S(x, y, t_n)]N(x, y, t_n) \quad (1)$$

where $F(\cdot)$ is the observed binary image consisting of two disjoint parts, i.e., target image $S(\cdot)$ and clutter image $N(\cdot)$ which are also binary.

Since a target moves in certain directions and builds a time-dependent track in the 3-D space, there exists spatial-temporal correlations between consecutive scan points. To find this dependency, we can use the following correlation equation:

$$Y(x, y, t_n) = g \left[\sum_{i=-|v|}^{|v|} \sum_{j=-|v|}^{|v|} F(x, y, t_n) F(x+i, y+j, t_{n+1}) \right] \quad (2)$$

where v is the maximum allowable target movement from one scan to the next and $Y(\cdot)$ represents how $F(x, y, t_n)$ at scan t_n are correlated to their neighboring points, within a window of size $(2v+1) \times (2v+1)$, at scan t_{n+1} . The function $g(\cdot)$ is a hard limiter threshold function with $g(c) = 1$ for $c \geq 0$ and $g(c) = 0$ for $c < 0$. Obviously, $Y(x, y, t_n) = 1$, implies that there is a two-point spatial-temporal sequence initiated at location (x, y) at scan t_n to location $(x+i, y+j)$ at scan t_{n+1} .

To determine the spatial-temporal correlation of more than two data points, and also impose both velocity and curvature limitations for the moving target, we can get a three consecutive scan RHOC equation [2] as

$$Y^{(k)}(x, y, t_n) = g \left[Y^{(k-1)}(x, y, t_n) \sum_{i_1} \sum_{j_1} Y^{(k-1)}(x+i_1, y+j_1, t_{n+1}) \times \sum_{i_2} \sum_{j_2} Y^{(k-1)}(x+i_1+i_2, y+j_1+j_2, t_{n+2}) \right] \quad (3)$$

where (i_1, j_1) and (i_2, j_2) represent the possible movements from scans t_n to t_{n+1} and t_{n+1} to t_{n+2} , respectively, and k is the order of the RHOC process. Consequently, $Y^{(k)}(x, y, t_n)$ gives correlational information among k consecutive scans, i.e., if $Y^{(k)}(x, y, t_n) = 1$, then there may exist a track extending from location (x, y, t_n) at scan t_n to a point at scan t_{n+k} ; otherwise these data points do not lie on a possible track and can subsequently be removed.

The relation between (i_1, j_1) and (i_2, j_2) can be shown as

$$\theta = \arctan \frac{j_1}{i_1} \quad (4)$$

where θ is the target bearing angle from scan t_n to scan t_{n+1} . Now if the maximum target moving curvature is σ then the turning angle from scan t_{n+1} to scan t_{n+2} should be within a bounded region, i.e.,

$$\theta - \sigma \leq \arctan \frac{j_2}{i_2} \leq \theta + \sigma. \quad (5)$$

Having specified the curvature constraint, the acceptable range of the values of (i_2, j_2) can be determined using (5). Thus, the range of movements from scan t_{n+1} to t_{n+2} is limited according to the assumed maximum moving curvature [1].

After applying the RHOC process to detect the target tracks, in some cases there may still be some noise points that remain in the processed image together with the real target track points. This occurs when the noise points are spatially and temporally close to the target points, hence satisfying the RHOC condition. A consistency test was adopted here to reject such points. The main idea of this algorithm is that the dynamic behavior of the real moving target always has a consistent characteristic, i.e., it may not change too frequently. Based on this idea, for scan k , we first select the data points left after the RHOC process from scan $k-4$ to scan $k+4$, i.e., in totally 9 scans. In each scan, there can be one or more points. All the possible target tracks based on these points are then considered. The choice of nine consecutive scans is empirically determined to be optimum since increasing the scan numbers will tremendously increase the number of candidate tracks while reducing the scan number will reduce the accuracy in estimation. For every possible track, the acceleration and curvature changes are calculated at each point. The variance of the each sequence is then calculated. To combine the variances of both acceleration and curvature change, we divide each variance by its mean and then add the two results for each candidate track yielding

$$S(k) = \frac{\sigma_a^2(k)}{\mu_a(k)} + \frac{\sigma_d^2(k)}{\mu_d(k)} \quad (6)$$

where σ^2 is the variance, μ is the mean, subscripts a and d represent acceleration and curvature changes, respectively, and S provides a measure of consistency. A track with the smallest S value is chosen among all the nine-point candidate tracks and the fifth point (in the middle) within this 9-points sequence is kept to be the real target point for scan k in the whole detected track. This process is then repeated to estimate the target point at scan $k+1$ and so on.

To see the effectiveness of the RHOC and consistency test, consider the multiscan (23 scans) image in Fig. 2(a) which contains one target track and clutter/noise with density 1%. The processed image after the RHOC process is shown in Fig. 2(b). As can be seen, most of the noise and clutter points

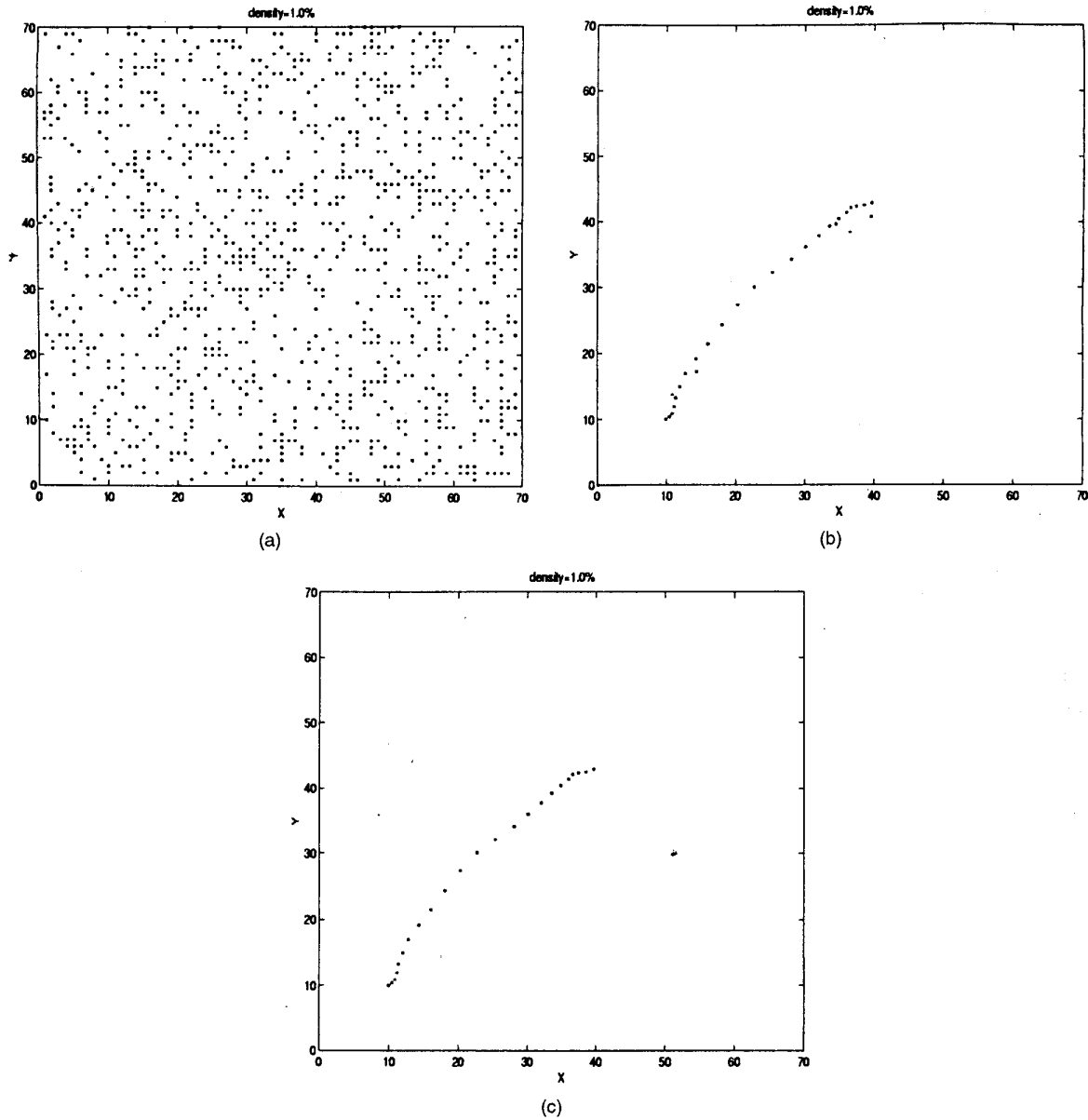


Fig. 2. Results of RHOC process on multiscan image. (a) Original multiscan image containing target track and noise/clutter. (b) detected target track after RHOC process. (c) Final result after consistency testing process. Noise and clutter points in vicinity of target points are removed.

are removed and all the target points are retained in the image. Nonetheless, there are few noise points in the proximity of the target points that are also retained after the RHOC process. Fig. 2(c) is the resultant image after the above-mentioned consistency process is applied. In this particular case, all the target points are retained while the noise points are rejected.

Having removed the competing clutter and noise and detected the target tracks using the RHOC process, we need to extract appropriate features from these tracks in order to perform target classification. If we define v_i and c_i as the velocity and curvature

of the moving target at time (or scan) t_i , then the acceleration value at this time is $a_i = v_{i+1} - v_i$, and the change in curvature is $d_i = c_{i+1} - c_i$ where $i \in [1, N]$ is the scan number and N is the total number of scans used. Note that a_i and d_i are evaluated at each target scan point. The information on the dynamic behavior of the target, i.e., the velocity and curvature can be employed to classify certain kinds of targets. This is due to the fact that these two features and their time history provide the maneuver characteristics of the moving targets. These characteristics that are different from target to target can be used to make the classification.

III. CONDITIONAL PROBABILITY GENERATOR

The classification using the Bayes decision rule requires the knowledge of the conditional probabilities based on the extracted features. In our scheme these are provided by the conditional probability generator (CPG) sub-system. The principle behind the CPG is somewhat similar to the hidden Markov model (HMM) [26] where we define several kinds of state patterns. These patterns are then used to train the neural network-based CPG system. During the testing phase, real observation patterns that may not be the same as the defined patterns are applied to the CPG system. The output of the CPG then provide the conditional probabilities of occurring certain kind of state pattern given the observation pattern.

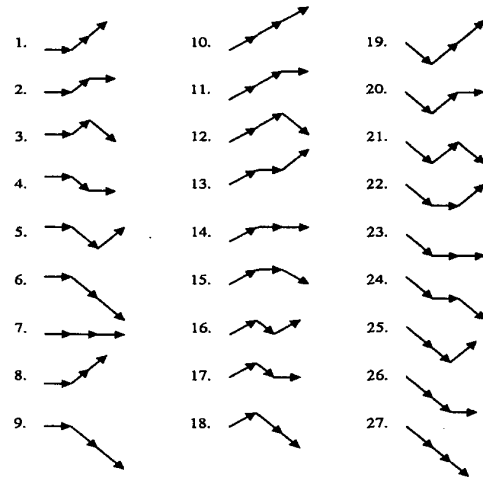


Fig. 3. All possible shapes for hidden states model.

A. Determination of Hidden States

To determine the conditional probabilities needed for subsequent Bayes classification process we need to define several hidden states for each feature type. These state vectors represent all the possible typical patterns of feature changes in three consecutive scans. Each hidden state vector contains three elements and each element can accept only three possible values namely A , 0 , or $-A$. For the acceleration feature, these values correspond to acceleration, constant speed, or deceleration cases, respectively. For the curvature changes feature, these correspond to left turn, no turn and right turn, respectively. The reason behind choosing three components in each hidden state vector is that choosing two components does not adequately represent some complicated patterns while four components would have tremendously increased the computational complexity. Therefore, we have a total of 27 hidden state vectors for each feature type as shown in Fig. 3. Note that in this figure, the arrow with 45° , 0° , and -45° deg orientation represents the values A , 0 , and $-A$, respectively.

The distribution of the amplitude of the target features is obviously different for different types of targets. Thus, when we define the hidden state vectors to classify two types of targets, the choice of the amplitude value A definitely affects the classification performance. The reason being for different amplitude A of the hidden state vectors, the conditional probabilities that CPG generates are different. If A is chosen to be too large, most of the real observation pattern components will be far less than A and as a result these patterns will be taken as state pattern $0,0,0$ whereas if the value of A is too small, they will be taken as A,A,A . In this case, we cannot discriminate the target types efficiently. The optimum value of A makes the observation patterns distribute more evenly across all the hidden state patterns hence leading to a more efficient classification.

This optimum value is determined empirically by varying this value, retraining the network for the new hidden states for every new A and then testing on a subset of the whole testing data. More specifically, the amplitude A is varied from possible small to large values while generating the subtest results from a small part of the testing samples. The peak in the performance plot is then determined and the value A which corresponds to this best performance point is obtained. This amplitude A is then kept fixed to perform the classification on the whole testing data.

B. Generating the Conditional Probabilities by Neural Networks

Based on the feature patterns and the changing amplitude, the target dynamic behavior can be determined by the conditional probabilities of occurring the hidden states given the observation vectors (patterns). The observation vectors are extracted from the actual target tracks and forming vectors similar to feature vectors with three components representing acceleration and curvature change sequences. Every two consecutive vectors contain two overlapping elements in order to consider all the dynamical information of the targets. The conditional probability can then be obtained using a back-propagation neural network (BPNN).

Let us assume that V is the input vector and M is the desired vector. The goal is to find a function $d(V)$ which is able to accurately recover M from V [7]. The most general way of selecting an estimate of the function $d(V)$ for functional fitting is to take a family of arbitrary functions of V and search among them (using the training process) for an individual function that best complies with the least mean squared (LMS) criterion, i.e.,

$$\epsilon^2 = \min_{d(V)} = E[|d(V) - M|^2]. \quad (7)$$

The optimum function is the so-called regression function. In [8], it is shown that the solution to this optimization problem is the conditional mean estimates, i.e.,

$$d(V) = \sum_M MP(M/V) = E(M/V). \quad (8)$$

Once this mapping function is captured by the neural network, for a given observation pattern O_j the output vector of the neural network is given [29] by

$$d(O) = \sum_M MP(M/O_j) = \begin{bmatrix} P(M_1/O_j) \\ P(M_2/O_j) \\ \vdots \\ P(M_K/O_j) \end{bmatrix} \quad (9)$$

where M is a K -dimensional vector with only one of its components equal to 1 and all the others equal to 0.

Since the BPNN uses the LMS criterion with desired outputs of 1 and 0, it can therefore be employed to compute $P(M/O_j)$ s. However, the hidden states are used in place of the desired vector M to train the BPNN. To increase the robustness of the BPNN, the training patterns should also include some noisy state patterns. The observation vectors O_j contains acceleration and curvature change components a_i or d_i . The output of the BPNN provides the conditional probability of occurring the hidden states given the observed features.

IV. SEQUENTIAL BAYES CLASSIFICATION SCHEME

Having computed the conditional probabilities for each observation vector using the CPG, one can simply use these probabilities as the input to another BPNN to perform the classification. Although this approach is fairly simple and can lead to reasonable results (see Section V), the size of the input vector to the classifier (i.e., acceleration and curvature changes sequence) must be fixed. Obviously, this major drawback precludes the use of this simple scheme for real-time target classification and tracking. The goal of this section is to develop a sequential classification scheme that can operate in a continuous real-time mode and update the classification results whenever a new data point is arrived.

Clearly, if "enough" observations are available, they can fully represent the statistical model of the target for perfect classification. In such an idea case, no more observation is needed to update the decision. However, in most of the real-life scenarios one cannot afford to wait making decisions until all of the needed observations are obtained. Consequently, decision has to be made based on only partial or even incomplete observation set.

Let us define the whole observation space as $\bar{O}(N) = \{O_1, O_2, \dots, O_j, \dots, O_N\}$ where O_j , $j \in [1, N]$ is the j th observation vector containing either the acceleration or curvature changes extracted from a track. These observations are arrived sequentially, from t_1 to t_N . Experience indicates that the probability density function of these observations is uniform and hence for a given N , one can use an *a priori* probability $P(O_j)$ for all.

At time or scan k , O_1 to O_k are observed and we would like to use these k observations to make a decision based on this partial observation set. Let $O(k) = \bigcup_{j=1}^k O_j$ be the collection of such observations. Note that union operation eliminates the redundancies among the observation vectors due to the overlapping elements. The goal of the Bayes classification is to maximize the *a posteriori* conditional probability $P[O(k) | T_n]$ of the observed sequence $\bar{O}(k)$, given the target model T_n which represents the n th target class.

The following developments are done assuming only two-target scenarios. Nevertheless, when the number of targets is more than two, one can classify each pair of targets first, and then use the likelihood ratio test to make the final decision for all the targets. Using this algorithm, if for targets j and k ,

$$P(O(k)/T_j) > P(O(k)/T_k) \quad (10)$$

then reject T_k from consideration and test T_{k+1} and continue. Otherwise, reject T_j , replace T_j with T_k , and continue. When getting the final T_j , we can make the decision that the target belongs to class j [30].

Here the *a priori* conditional probability $P(O(k) | T_n)$ is generated using a sequential scheme. Let us define

$$R_n(k) = P\left(\bigcup_{j=1}^k O_j/T_n\right) \quad (11)$$

and

$$S_n[m(k)] = P\left(M_{m(k)}, \bigcup_{j=1}^k O_j/T_n\right) \quad (12)$$

where $R_n(k)$ is the conditional probability of occurring the partial observation sequence $\bigcup_{j=1}^k O_j$ given the target type T_n , and $S_n[m(k)]$ is the joint probability of occurring the observation sequence $\bigcup_{j=1}^k O_j$ and the hidden state $M_{m(k)}$ at time k , given the target type T_n . Here $m(k)$ is the m th hidden state at time k . Obviously, we are interested in computing $R_n(k)$ in a recursive fashion using the following initial conditions:

$$R_n(1) = P(O_1/T_n) \quad (13)$$

and

$$S_n[m(1)] = P(M_{m(1)}, O_1/T_n) = P(M_{m(1)}/O_1, T_n)P(O_1/T_n) \quad (14)$$

where $M_{m(1)}$ is the m th hidden state occurring at time 1, $P(M_{m(1)}/O_1, T_n)$ is generated by using the CPG and we explain how to get $P(O_1/T_n)$ in the later part of this section.

Now incrementing k to $k + 1$, yields the following equation for $S_n[m(k + 1)]$, i.e.,

$$\begin{aligned} S_n[m(k + 1)] &= P\left(M_{m(k+1)}, \bigcup_{j=1}^{k+1} O_j/T_n\right) \\ &= P(M_{m(k+1)}/O_{k+1}/T_n) \\ &\quad + P\left(M_{m(k+1)}, \bigcup_{j=1}^k O_j/T_n\right) \\ &\quad - P\left(M_{m(k+1)}, O_{k+1}, \bigcup_{j=1}^k O_j/T_n\right). \end{aligned} \quad (15)$$

In the above equation, the first term can be expanded using

$$P(M_{m(k+1)}/O_{k+1}/T_n) = P(M_{m(k+1)}/O_{k+1}, T_n)P(O_{k+1}/T_n) \quad (16)$$

where $M_{m(k+1)}$ is the m th hidden state occurring at time $k + 1$ and $P(M_{m(k+1)}/O_{k+1}, T_n)$ can be obtained by the CPG. The computation of $P(O_{k+1}/T_n)$ is also explained in the later part of this section. The second term in (15) can be expanded by the total probability property, i.e.,

$$\begin{aligned} &P\left(M_{m(k+1)}, \bigcup_{j=1}^k O_j/T_n\right) \\ &= \sum_{m1(k)} P\left(M_{m(k+1)}/M_{m1(k)}, \bigcup_{j=1}^k O_j, T_n\right) \\ &\quad \times P\left(M_{m1(k)}, \bigcup_{j=1}^k O_j/T_n\right). \end{aligned} \quad (17)$$

Since the information of $\bigcup_{j=1}^k O_j$ is embedded in $M_{m1(k)}$, we can alternatively write

$$\begin{aligned} &P\left(M_{m(k+1)}, \bigcup_{j=1}^k O_j/T_n\right) \\ &= \sum_{m1(k)} P(M_{m(k+1)}/M_{m1(k)}, T_n)S_n[m1(k)] \end{aligned} \quad (18)$$

where $P(M_{m(k+1)}/M_{m1(k)}, T_n)$ is the transitional probability of the m th hidden state occurring at time $k + 1$ given that the $m1$ th hidden state has occurred at time k for target T_n . The computation of this transitional probability is also further explained in the later part of this section.

The third term in (15) can be expanded as

$$\begin{aligned} &P\left(M_{m(k+1)}, O_{k+1}, \bigcup_{j=1}^k O_j/T_n\right) \\ &= P\left(M_{m(k+1)}/O_{k+1}, \bigcup_{j=1}^k O_j, T_n\right)P\left(O_{k+1}, \bigcup_{j=1}^k O_j/T_n\right). \end{aligned} \quad (19)$$

The first term in this equation can be simplified to $P(M_{m(k+1)}/O_{k+1})$ since when O_{k+1} is given, the conditioning of $M_{m(k+1)}$ on $\bigcup_{j=1}^k O_j$ can be ignored. The second term can be expanded using the same method as in (17) and (18), i.e.,

$$\begin{aligned} P\left(O_{k+1}, \bigcup_{j=1}^k O_j/T_n\right) &= \sum_{m2(k)} \sum_{m3(k)} P(O_{k+1}/M_{m3(k+1)}) \\ &\quad \times P(M_{m3(k+1)}/M_{m2(k)}, T_n)S_n[m2(k)]. \end{aligned} \quad (20)$$

Thus, we can get the final recursive equation for $S_n[m(k + 1)]$ as

$$\begin{aligned} S_n[m(k + 1)] &= P(M_{m(k+1)}/O_{k+1}, T_n)P(O_{k+1}/T_n) \\ &\quad + \sum_{m1(k)} P(M_{m(k+1)}/M_{m1(k)}, T_n)S_n[m1(k)] \\ &\quad - P(M_{m(k+1)}/T_n) \\ &\quad \times \left\{ \sum_{m2(k)} \sum_{m3(k)} P(O_{k+1}/M_{m3(k+1)}) \right. \\ &\quad \left. \times P(M_{m3(k+1)}/M_{m2(k)}, T_n)S_n[m2(k)] \right\}. \end{aligned} \quad (21)$$

In the similar fashion for $R_n(k + 1)$ we get

$$\begin{aligned} R_n(k + 1) &= P\left(\bigcup_{j=1}^{k+1} O_j/T_n\right) = P\left(\bigcup_{j=1}^k O_j \cup O_{k+1}/T_n\right) \\ &= P\left(\bigcup_{j=1}^k O_j/T_n\right) + P(O_{k+1}/T_n) \\ &\quad - P\left(\bigcup_{j=1}^k O_j, O_{k+1}/T_n\right) \\ &= R_n(k) + P(O_{k+1}/T_n) \\ &\quad - \left\{ \sum_{m1(k)} \sum_{m2(k)} P(O_{k+1}/M_{m1(k+1)}) \right. \\ &\quad \left. \times P(M_{m2(k+1)}/M_{m1(k)}, T_n)S_n[m1(k)] \right\}. \end{aligned} \quad (22)$$

To find $P(O_k/T_n)$ s, we can use the total probability, i.e.,

$$\begin{aligned} P(O_k/T_n) &= \frac{P(T_n/O_k)P(O_k)}{P(T_n)} \\ &= \frac{\sum_{m(k)} P(T_n/M_{m(k)})P(M_{m(k)}/O_k)P(O_k)}{P(T_n)} \end{aligned} \quad (23)$$

where $P(M_{m(k)}/O_k)$ s are obtained from the outputs of the CPG, and $P(T_n/M_{m(k)})$ can be calculated by CPG over all the target tracks in the training sets.

The transitional probabilities can be estimated by using the neural network CPG on the training data. Here we have used two two-layer BPNNs, one for acceleration and one for curvature patterns. The architecture of each network is 3-36-27 since based upon the three extracted features, the network generates all (i.e., 27) the necessary conditional probabilities. For a certain type of target, the initial values of all the components the 27×27 transitional matrix are set to zero. Then, the observation vectors from the training data set for that type of target are used as the input to determine these probabilities. For every pair of consecutive observations e.g., O_k, O_{k+1} , we can get a pair of conditional probability vectors, $P(M/O_k), P(M/O_{k+1})$, (where M is a vector containing from M_1 to M_{27}), from the output of the CPG. For each vector, $P(M/O_k)$ or $P(M/O_{k+1})$, the maximum component of the conditional probability output (say $P(M_i/O_k), P(M_j/O_{k+1})$) is chosen and its hidden state number (i, j) is recorded. This process is repeated for every O_k and O_{k+1} pairs in the training data and the relative frequency of occurrence of such (i, j) pair is used as the (i, j) th element of the transitional matrix.

Once the outputs of the Bayes classifiers for both acceleration and curvature change features are determined recursively using (21) and (22), a nonlinear fusion scheme is used to combine these two outputs and get the final classification result. A two-layer BPNN was designed to perform this operation. The network had two inputs corresponding to the classification outputs of the two feature vectors, four hidden layer nodes, and two output nodes for the final decision. This network is also trained using the same training sets used to find $P(M_i/T_n)$

V. IMPLEMENTATION AND RESULTS

This section provides the models which were used to simulate the target tracks and presents the results of the proposed neural network directed Bayes classifier for different operating conditions. These include classification of three types of targets using likelihood ratio test, and for different noise level classification of two types of targets for different model parameters

and the performance comparison with the scoring (neural network) classifier and Bayes classifier.

A. Simulated Target Track Models

The target under consideration normally moves at a constant velocity. Turns, evasive maneuvers, and accelerations due to the atmosphere turbulence may be viewed as perturbations upon the constant velocity trajectory. The acceleration of the moving target, since it accounts for the target deviations from a straight line trajectory, will henceforth form the target maneuver variable. The (single dimension) maneuver capability can be satisfactorily specified by two quantities: the variance, or magnitude, of the target maneuver and the time constant or duration of the target maneuver. The target acceleration, and hence the target maneuver, is correlated in time, i.e., if a target is accelerating at time k , it is likely to be accelerating at time $k + \tau$ for sufficiently small τ as well [6].

Based on the above assumption, target tracks were generated based on a first-order Markov model [5, 6] with different sets of parameters. Along a single direction in a 2-D space, the model is given by

$$a_{k+1} = \rho_m a_k + \sqrt{1 - \rho_m^2} \sigma_m r_k \quad (24)$$

where a_k is the moving target acceleration value at time k and r_k is a white noise process with zero mean and unit standard deviation. The first-order correlation parameter is $\rho_m = e^{-\alpha T}$. From [5], we know that α is also called the reciprocal of the maneuver (acceleration) time constant. Varying α from small values close to zero to 1 corresponds to slow to quick maneuvers. The deviation of acceleration σ_m is given by

$$\sigma_m = A_{\max} \sqrt{\frac{[1 + 4P_{\max} - P_0]}{3}} \quad (25)$$

where it is assumed that the target can accelerate at a maximum rate A_{\max} (or $-A_{\max}$) with a probability P_{\max} , or the target undergoes no acceleration with a probability P_0 , or will accelerate between the limits $-A_{\max}$ and A_{\max} according to a uniform distribution [28].

Note that the above model is represented only in one dimension. When combined with the models in the other dimension, we can fully represent the tracks.

B. Classification Performance for Three Types of Targets

Three sets of model parameters were used to generate three different target types. These are given in Table I. The likelihood ratio test was then used to process the output of the sequential Bayes classifier and get the final result.

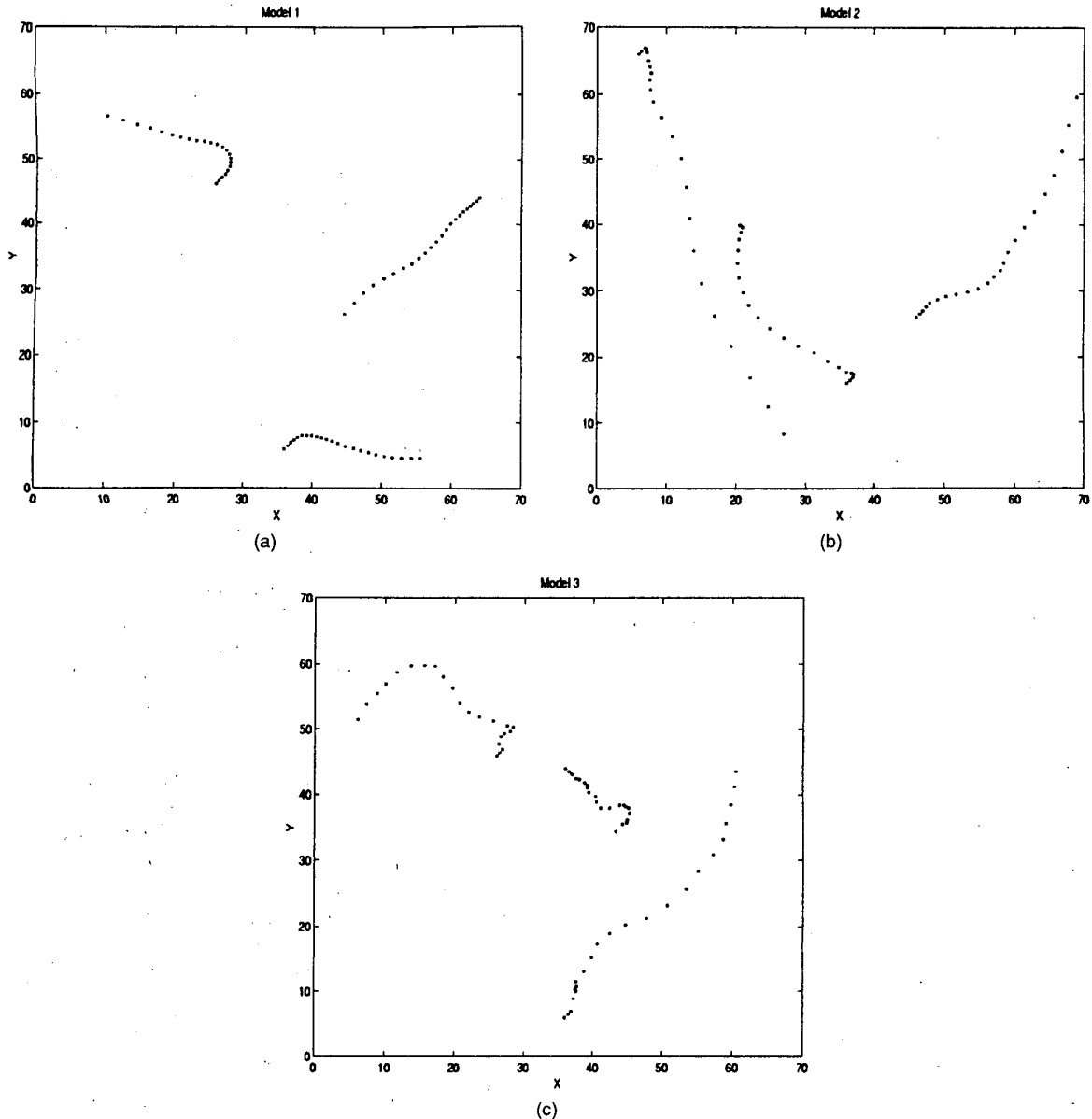


Fig. 4. Three types of targets based on different model parameters. (a) Typical tracks for target 1. (b) Typical tracks for target 2. (c) Typical tracks for target 3.

TABLE I
Model Parameters for Three Types of Targets

| Parameter | Model 1 | Model 2 | Model 3 |
|------------|---------|---------|---------|
| α | 1/60 | 1/10 | 1/2.5 |
| A_{\max} | 2g | 4g | 6g |
| P_{\max} | 0.1 | 0.2 | 0.3 |
| P_0 | 0.7 | 0.5 | 0.3 |

Among these three models, target 1, (T_1), is the one with the most steady dynamic behavior as α for this target is very small. The maximum allowable acceleration A_{\max} for a target with steady

behavior is always less than those for the other types. Additionally, it is obvious that steady moving targets have smaller probability associated with A_{\max} and larger probability associated with A_0 . Thus, we choose a small value for P_{\max} and a large value for P_0 for this target. In contrast, target 3, (T_3), is the one with the most erratic dynamic behavior. As a result, its α value is closer to one and A_{\max} is the largest. For the same reason its P_{\max} is the largest and P_0 is the smallest. The dynamic behavior for target 2, (T_2), is in between T_1 and T_3 , hence all its model parameters are also between those of T_1 and T_3 . Figs. 4(a), (b), and (c) show some typical examples of tracks for T_1 , T_2 , and T_3 , respectively.

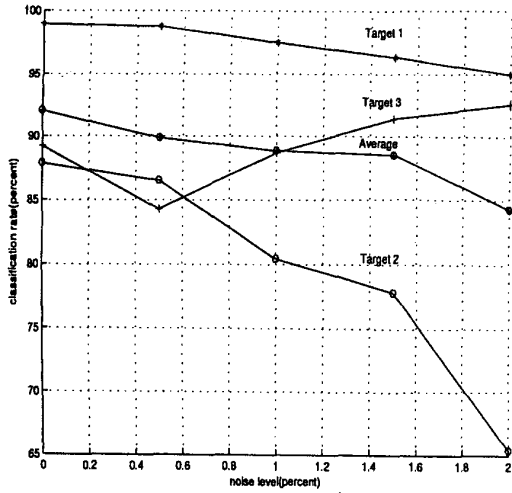


Fig. 5. Classification performance for three types of targets.

For each type of target, 1000 tracks were generated based on its corresponding dynamic model. These were then divided into two parts: training and testing samples, each containing 500 tracks. The neural network CPGs, with 3 inputs, 36 hidden layer neurons and 27 output neurons, were trained using the standard back-propagation learning. The maximum number of epochs was 80,000, the learning rate was 0.001 and the error goal was chosen to be 0.001. The classification performance was evaluated on both the noise-free cases as well as heavily cluttered multiscan images for different noise/clutter levels. Fig. 5 shows the classification performance plots for all three types of targets and at several noise levels namely, 0.5%, 1.0%, 1.5%, and 2.0%. Note that the noise level, $n\%$, used here implies that in a 70×70 window there are $4900 \times n\%$ noise points in the background image in each scan. Thus, the 2.0% noise level represents extremely noisy or cluttered cases.

Since the model parameters of T_1 are widely apart from those of T_2 and T_3 , the performance degrades only slightly when increasing the noise level. However, the performance on T_2 degrades substantially. The reason being the model parameters for T_2 and T_3 are very close together. In addition, T_3 classification rate drastically decreased at noise level 0.5%. The reason for this sharp decline is that among all three targets, T_3 has more erratic behavior. As a result, after the RHOC process and because of the consistency condition in this scheme, at this low level noise, the extracted tracks associated with this target exhibit similar dynamic behavior to that of T_2 . However, when the noise level increases due to the erratic behavior of T_3 , the presence of noise close to the tracks leads to a superficial improvement in the classification rate while significantly degrading that of T_2 which has more consistent behavior. The overall classification rate for all three types of targets degrades only

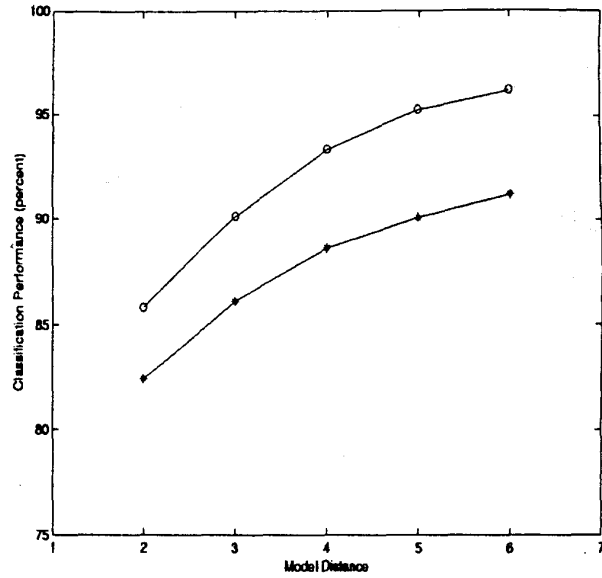


Fig. 6. Classification performance for different model distance values ($\alpha_3 = 1/5$ for the upper plot).

moderately when increasing the noise level as shown in Fig. 5.

C. Effects of Parameter α

In the simulated target model (24), there are three parameters which control the maneuvering behavior of the targets. Parameter α controls the time constant of the target dynamics, i.e., how long the target will keep the same maneuvering behavior. This parameter determines the correlation parameter ρ in the model (24). On the other hand, r_k represents the driving noise in the model and σ_m controls the amplitude of the maneuvering change of the target. Among these parameters, α is the most important one as it primarily controls the dynamic behavior of the target. Thus, in this subsection, we study the effects of varying the model parameter α on the classification performance.

Since the discrimination of T_1 does not present a challenge to the classification system, here we primarily focus our attention on the classification of T_2 and T_3 . The values of A_{\max} , P_{\max} , and P_0 were kept the same as models 2 and 3 as in the last experiment. Thus, the parameter σ_m is fixed for each model. The parameter α of model 3 was kept at $1/2.5$ while that of models 2 varied with values $1/5$, $1/7.5$, $1/10$, $1/12.5$, and $1/15$. Let us define a model distance $\delta = \alpha_3/\alpha_2$ where the subscript 2 and 3 correspond to the model 2 and 3, respectively.

Fig. 6 shows the correct classification rate for the two targets for different values of δ . As can be observed, when we increase the model distance δ the rate of improvement at the beginning is substantial. It is interesting to see that when $\delta = 2$, i.e., the model

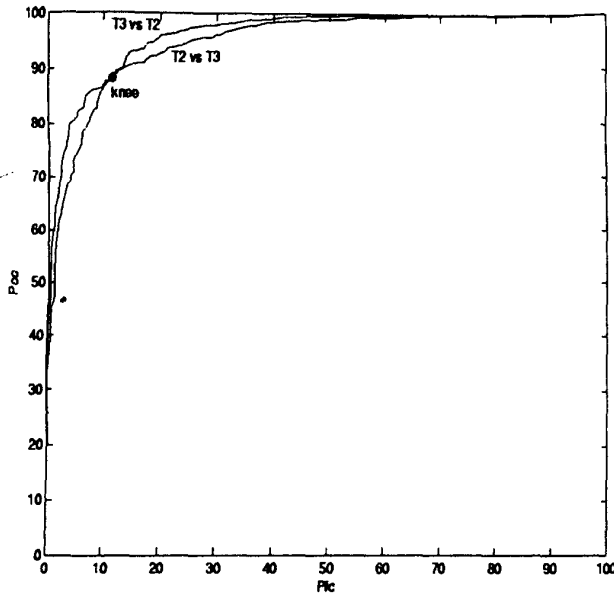


Fig. 7. ROC curve for two-target classification.

distance is very close, we still can get around 82% classification rate. This demonstrates the robustness of the classification system. As we further increase the distance δ , the correct classification rate will approach a certain final value, which may not be 100%. This is due to the fact that the choice of $\alpha_3 = 1/2.5$, leads to some erraticness in the behavior and hence more misclassifications. If we decrease α_3 to $1/5$, the improvement in the classification rate at most model distances will be more than 4% as shown in Fig. 6. Consequently, parameter α greatly affects the classification performance.

Fig. 7 shows the receiver operating characteristic (ROC) curve obtained on 1000 testing tracks for T_2 and T_3 for $\alpha_2 = 1/10$ and $\alpha_3 = 1/2.5$ case. Note that this ROC curve for "T3 vs T2" case presents the classification rate (P_{cc}) associated with T_3 only, whereas the false alarm corresponds to misclassifications (P_{fc}) of T_2 as T_3 , and vice versa for the "T2 vs T3" case. These curves are generated by varying the threshold for classification decision so that the false alarms and correct classifications for various samples of the testing set can be counted. Thus, it is possible to generate a probability of correct classification and a probability of false alarm for a certain threshold value. These two probabilities describe one point on the ROC curve for the testing data. As can be observed from this ROC curve, the classifier provides very good classification results although the model distance ($\delta=4$) is relatively close for these cases. From Fig. 7, one can find that for both T_2 vs T_3 and T_3 vs T_2 cases, the correct classification rate and the false alarm rate at the knee of the ROC curve are 88.3% and 11.7%, respectively. For each threshold value, we can generate a confusion

TABLE II
Confusion Matrix for Sequential Bayes Classifier

| True Class | Chosen Class | |
|------------|--------------|----------|
| | Target 2 | Target 3 |
| Target 2 | 87.8% | 12.2% |
| Target 3 | 10.6% | 89.4% |

Note: Model distance $\delta = 2$.

TABLE III
Confusion Matrix for Neural Network Classifier

| True Class | Chosen Class | |
|------------|--------------|----------|
| | Target 2 | Target 3 |
| Target 2 | 90.57% | 9.43% |
| Target 3 | 17.4% | 82.6% |

Note: Model distance $\delta = 2$.

matrix. In each matrix, for example in Table II, the summation of the components (1,1) and (1,2) should be equal to one. And the summation of (2,1) and (2,2) should also be one. For "T2 vs T3" case, we use the components (1,1) and (2,1) as the classification and false alarm rates, respectively. For the T_3 vs T_2 case, we use (2,2) and (1,2). So if in T_2 vs T_3 case, we find the knee point at which $(1,1) + (2,1) = 1$, then using the same confusion matrix, $(2,2) + (1,2) = 1$, at the knee point for T_3 vs T_2 case. That's why we get the same knee points for both cases. The difference between these two cases is that at the left side of the knee point, when increasing (P_{cc}), (P_{fc}) for the T_2 vs T_3 case will increase faster than that for the T_3 vs T_2 case. This is the opposite in the right of the knee point. Since these two curves are generated based on the same data, the area below each curve should be the same. Thus, if on the left side of the knee point, one curve is higher than the other, on the right side, this curve should be lower.

D. Performance Comparison

If the data length of the tracks is fixed, both a neural network and the proposed sequential Bayes classifier schemes can be used to make classification after the CPG process. Tables II and III show the comparison of these two classification methods based on the cases presented in the previous section. As can be seen, the sequential Bayes classifier in this paper has a better overall classification performance.

In real-life applications, the choice of the detection/classification method depends on the actual operating conditions. If the goal is to continuously process the multiscan image data scan by scan and in real-time, then the proposed Bayes classifier is more suitable owing to its sequential nature. In this scheme, when a new data point is received, the results can be updated. However, if a fixed-length track is given,

the neural network classifier may be an easier and less complicated choice. In addition, this method offers perhaps a faster decision and a simpler structure.

VI. CONCLUSION

This paper is concerned with the development of a new neural network directed Bayes classifier for moving target classification. The information on acceleration and curvature changes was extracted as the dynamic features and subsequently used for classification. Using these features, the conditional probabilities were generated using a neural network trained on several hidden states. A novel sequential Bayes classification scheme was developed that provides the classification results based upon the output of the neural network CPG. Using this classifier, the decision can be updated as new scans of data are becoming available. Finally, for each target pair, a nonlinear fusion system was implemented to combine the results of the classifier for each feature type. The likelihood ratio test algorithm was then employed for multiple types of targets. Simulation results on both noise-free and noisy cases for three kinds of targets were provided which showed the effectiveness of our scheme for classifying the targets based on their dynamical behavior. The classification results for a pair of targets with different model distance conditions were also presented. These results demonstrated the robustness of the algorithm even when the model distance is very small. Finally, the performance of the proposed sequential Bayes classifier scheme was compared with a neural network-based classifier.

REFERENCES

- [1] Liou, R. J., and Azimi-Sadjadi, M. R. (1993)
Dim target detection using high order correlation.
IEEE Transactions on Aerospace and Electronic Systems, **29** (July 1993), 841–856.
- [2] Liou, R. J., and Azimi-Sadjadi, M. R. (1998)
Multiple target detection using modified high order correlation.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 2 (Apr. 1998), 553–567.
- [3] Reed, I. S., Gagliardi, R. M., and Shao, H. M. (1983)
Application of three-dimensional filtering to moving target detection.
IEEE Transactions on Aerospace Electronic Systems, **AES-19**, 6 (Nov. 1983), 898–905.
- [4] Porat, B., and Friedlander, B. (1990)
A frequency domain algorithm for multiframe detection and estimation of dim targets.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **12** (Apr. 1990), 398–401.
- [5] Mohanty, N. C. (1981)
Computer tracking of moving targets in space.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **PAMI-3**, 5 (Sept. 1981), 606–611.
- [6] Reid, D. B. (1979)
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, **AC-24**, 12 (Dec. 1979), 843–854.
- [7] Blackman, S. S. (1986)
Multiple-Target Tracking with Radar Applications.
Dedham, MA: Artech House, 1986.
- [8] Bar-Shalom, Y., and Fortman, T. E. (1988)
Tracking and Data Association.
New York: Academic Press, 1988.
- [9] Bar-Shalom, Y., Shertukde, H. M., and Pattipati, K. P. (1989)
Use of measurements from an imaging sensor for precision target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 6 (Nov. 1989), 863–872.
- [10] Bar-Shalom, Y., Shertukde, H. M., and Pattipati, K. P. (1990)
Precision target tracking for small extended objects.
Optical Engineering, **29**, 2 (Feb. 1990), 121–126.
- [11] Shertukde, H. M., and Bar-Shalom, Y. (1991)
Tracking of crossing targets with imaging sensors.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 4 (July 1991), 582–592.
- [12] Shertukde, H. M., and Bar-Shalom, Y. (1990)
Detection and estimation for multiple targets with two omnidirectional sensors in the presence of false measurements.
IEEE Transactions on Acoustics, Speech, and Signal Processing, **38**, 5 (May 1990), 749–763.
- [13] Roth, M. (1989)
Neural networks for extraction of weak targets in high clutter environments.
IEEE Transactions on Systems, Man, and Cybernetics, **19**, 10 (Oct. 1989), 1210–1217.
- [14] Riegger, S., and Wiesbeck, W. (1989)
Wide-band polarimetry and complex radar cross section signatures.
Proceedings of the IEEE, **77**, 5 (May 1989), 649–658.
- [15] Walton, E. K., and Jouny, I. (1990)
Bispectrum of radar signatures and application to target classification.
Radio Science, **25**, 2 (Mar.–Apr. 1990), 101–113.
- [16] Strausberger, D. J., Garber, E. D., Chamberlain, N. E., and Walton, E. K. (1992)
Modeling and performance of HF/OTH radar target classification systems.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 2 (Apr. 1992), 396–402.
- [17] Silverstein, P. B., Sands, O. S., and Garber, F. D. (1991)
Radar target classification and interpretation by means of structural descriptions of backscatter signals.
IEEE AES Systems Magazine (May 1991), 3–7.
- [18] Zyweck, A., and Bogner, R. (1996)
Radar target classification of commercial aircraft.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 2 (Apr. 1996), 589–606.
- [19] Baum, J., Tung, E., and Rak, S. (1994)
Non-cooperative identification of ships with electrooptical data.
The Lincoln Laboratory Journal, **7**, 1 (1994).
- [20] Eom, K. B., and Chellappa, R. (1997)
Noncooperative target classification using hierarchical modeling of high-range resolution radar signatures.
IEEE Transactions on Signal Processing, **45**, 9 (Sept. 1997), 2318–2327.

- [21] Stewary, C., Lu, Y., and Larson, V. (1994)
A neural clustering approach for high resolution radar target classification.
Pattern Recognition, **27**, 4 (1994), 503–513.
- [22] Hauter, A., Chang, K. C., and Karp, S. (1997)
Polarimetric fusion for synthetic aperture radar target classification.
Pattern Recognition, **30**, 5 (1997), 769–775.
- [23] Menon, M. M., Boudreau, E. R., and Kolodzy, P. J. (1993)
An automatic ship classification system for ISAR imagery.
The Lincoln Laboratory Journal, **6**, 2 (1993).
- [24] Novak, L. M., Owirka, G. J., and Netishen, C. M. (1993)
Performance of a high-resolution polarimetric SAR automatic target recognition system.
The Lincoln Laboratory Journal, **6**, 1 (1993).
- [25] Verbout, S. M., Irving, W. W., and Hanes, A. S. (1993)
Improving a template-based classifier in a SAR automatic target recognition system by using 3-D target information.
The Lincoln Laboratory Journal, **6**, 1 (1993).
- [26] Rabiner, L. R. (1989)
A tutorial on hidden Markov models and selected applications in speech recognition.
Proceedings of the IEEE, **77**, 2 (Feb. 1989), 257–285.
- [27] Ricquebourg, Y. (1997)
Tracking of articulated structures exploiting spatio-temporal image slices.
In *Proceedings of 1997 IEEE International Conference on Image Processing*, Santa Barbara, CA, Oct. 1997.
- [28] Singer, R. (1970)
Estimating optimal tracking filter performance for manned maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **6**, 4 (July 1970), 473–483.
- [29] Schurmann, J. (1996)
Pattern Classification, A Unified View of Statistical and Neural Approaches.
New York: Wiley, 1996.
- [30] Stark, H. (1994)
Probability, Random Processes, and Estimation Theory for Engineering.
Englewood Cliffs, NJ: Prentice Hall, 1994.



Xi Yu received the B.Sc. degree in electronics engineering from Tainjin University, China in 1995 and M.Sc. degree in electrical and computer engineering from Colorado State University in 1999.

From 1997 to 1998, he was a graduate research assistant at Digital Signal and Image Processing Laboratory, Colorado State University, where he was engaged in research in the areas of target classification and neural networks. He is currently working at Cisco Systems, Inc.



M. R. Azimi-Sadjadi (S'81—M'81—SM'89) received his M.S. and Ph.D. degrees from the Imperial College of Science and Technology, University of London, England in 1978 and 1982, respectively, both in electrical engineering with specialization in digital signal/image processing.

He is currently a full professor at the Electrical and Computer Engineering Department at Colorado State University (CSU). He is also serving as the Director of the Multi-sensory Computing Laboratory (MUSCL) at CSU. His main areas of interest include digital signal and image processing, target detection, classification and tracking, adaptive filtering and system identification, and neural networks.

Dr. Azimi-Sadjadi has authored over one hundred journal and refereed conference publications. He is the co-author of the book *Digital Filtering in One and Two Dimensions* (Plenum Press, 1989). He is the recipient of the 1999 ABELL Teaching Award, 1993 ASEE-Navy Senior Faculty Fellowship Award, 1991 CSU Dean's Council Award, and 1984 DOW Chemical Outstanding Young Faculty Award. He has served as an Associate Editor of the *IEEE Transactions on Signal Processing*.