IEEE AEROSPACE and ELECTRONIC SYSTEMS MAGAZINE
January 2004
ISSN 0885-8985
Volume Nineteen
Number One
Part 2 of 2

Tutorials

IEEE

IEEE Aerospace & Electronic Systems Magazine Volume 19 Number 1 January 2004 0885-8985 Part Two of Two

Periodicals Postage paid at New York, NY, and at additional mailing offices.

IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE® (ISSN 0885-8985; USPS 212-660) is published monthly by the Institute of Electrical and Electronics Engineers, Inc. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society/Council, or its members. **IEEE Corporate Office:** Three Park Ave., 17th Floor, New York, NY 10016-5997, USA. **IEEE Operations Center:** 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA. **NJ Telephone:** (732) 981-0060. **Price/Publication Information:** Individual copies: IEEE Members $10.00 (first copy only), nonmembers $20.00 per copy. (Note: add $4.00 postage and handling charge to any order from $1.00 to $50.00, including prepaid orders.) Member and nonmember subscription prices available on request. Available in microfiche and microfilm. Periodicals postage paid at New York, NY, and at additional mailing offices. **Postmaster:** Send address changes to *IEEE Aerospace and Electronic Systems Magazine*, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA. GST Registration No. 125634188. Printed in United States of America.

Why this Publication?

IEEE Aerospace & Electronic Systems Magazine articles apprise readers of new developments, new applications of old technology, and news of society members, meetings, and related items; ***IEEE Transactions on Aerospace and Electronic Systems*** publishes novel, previously unpublished material of a technical nature.

What about those items that fall between these two diametric criteria?

This publication is our answer.

Future ***Tutorials*** will be forthcoming if they are useful and well received—as determined by *your* feedback.

We want to hear from you! Please direct comments—pro and con—to Peter Willett as this publication is his idea; he assumed the task of obtaining and managing the refereeing of these initial contributions. Peter explains what a tutorial is—and is not—on the following page.

Thanks are extended to Russ Lefevre, President, Ed Reedy, Vice-President of Publications, and the editorial and production staffs of both ***Transactions*** and ***Systems*** for their outstanding cooperation, knowledge, and assistance in bringing this initial issue to you.

Joel F. Walker
Assistant Vice President–
Publications, IEEE/AESS

What is an AESS Tutorial?

It is a rare article from which nothing is to be learnt, so what do we mean here by tutorial? For me, there are two kinds of tutorial articles: Those that provide a primer on an established topic, and those that let us in on the ground floor of something of emerging importance. In this, our initial issue, we have exemplars of each.

The first sort is excellently represented by the articles from Samuel Blackman and William Melvin. In both cases we have a noted expert who has been gracious (and brave) enough to have written us a field guide, respectively, to the MHT and to STAP. If you want to program an MHT you had better buy Sam's book (coauthored by Robert Popoli); but if you've heard the acronym and just want to know what it's all about, read what he writes here. New results on STAP appear almost to quickly to digest; but for an orientation, take a look at what Bill has given us in this issue.

The articles by Paul Baggenstoss and Ronald Mahler epitomize the other sort of tutorial. Here we have two very strong researchers who have each been laboring on a topic for some years. I have seen many of their presentations and followed much of their progress over that time. Interest in their respective areas is growing markedly, but I expect that many readers will not yet be aware of them. For many readers you will see it here first.

I like both sorts of tutorials very much. But as Associate Editor for both our transactions and our magazine, I know that there has been no logical place for them in the IEEE AES society until now. We hope, with these tutorials, that we can give them a home, a welcome, and provide a service to our membership.

We do not intend to publish ***Tutorials*** on a regular basis, but we hope to deliver them once or twice per year. We need good, useful tutorial articles (both kinds!) in relevant AESS areas. If you, the reader, can offer a topic of interest and an author to write about it, please contact me. Self-nominations are welcome, and even more ideal is a suggestion of an article that the editor(s) can solicit. All articles will be reviewed in detail. Criteria on which they will be judged include their clarity of presentation, their relevance and likely audience, and, of course, their correctness and scientific merit. As to the mathematical level, the articles in this issue are a good guide; In each case the author has striven to explain complicated topics in simple (well, tutorial) terms. There should be no (or very little) novel material. The home for archival science is our ***Transactions***, and submissions that need to be properly peer-reviewed will be rerouted there. Likewise, articles that are interesting and descriptive, but lack significant tutorial content, ought more properly be submitted to ***Systems Magazine***.

Despite Joel's kind words, these ***Tutorials*** are an idea for which credit ought to be spread to many quarters: Russ Lefevre, Ed Reedy, Ron Schroer, Dale Blair, and especially to Dave Dobson and Joel Walker himself. I hope these ***Tutorials*** turn out to be successful and useful, and, if they do, it is these individuals we should thank. I would like to echo Joel's request for inputs. We welcome your suggestions for topics and authors for future issues, as well as your thoughts and criticisms on the way that this issue has been structured. This is a new initiative, we are open to ideas. My contact information is on the inside front cover.

Peter K. Willett
Associate Editor,
IEEE Aerospace & Electronic Systems Magazine
Editor for Target Tracking and Multisensor Systems,
IEEE Transactions on Aerospace and Electronic Systems

Multiple Hypothesis Tracking For Multiple Target Tracking

SAMUEL S. BLACKMAN
Raytheon

Multiple hypothesis tracking (MHT) is generally accepted as the preferred method for solving the data association problem in modern multiple target tracking (MTT) systems. This paper summarizes the motivations for MHT, the basic principles behind MHT and the alternative implementations in common use. It discusses the manner in which the multiple data association hypotheses formed by MHT can be combined with multiple filter models, such as used by the interacting multiple model (IMM) method. An overview of the studies that show the advantages of MHT over the conventional single hypothesis approach is given. Important current applications and areas of future research and development for MHT are discussed.

Manuscript received February 22, 2003; revised April 27, 2003.

Author's current address: Raytheon, RE/R07/P572, PO Box 920, El Segundo, CA 90245, E-mail: (ssblackman@raytheon.com).

I. INTRODUCTION

Target tracking is an essential requirement for surveillance systems employing one or more sensors, together with computer subsystems, to interpret the environment. Typical sensor systems, such as radar, infrared (IR), and sonar, report measurements from diverse sources: targets of interest, physical background objects such as clutter, or internal error sources such as thermal noise. The target tracking objective is to collect sensor data from a field of view (FOV) containing one or more potential targets of interest and to then partition the sensor data into sets of observations, or tracks that are produced by the same object (or target). Note that the term target is used in a general sense. Once tracks are formed and confirmed (so that background and other false targets are reduced), the number of targets of interest can be estimated and quantities, such as target velocity, future predicted position, and target classification characteristics, can be computed for each track.

Since most surveillance systems must track multiple targets, multiple target tracking (MTT) is the most important tracking application. Fig. 1, taken from [1], shows the basic elements of a typical MTT system. Assume that tracks have been formed from previous data and a new set of input observations becomes available. In general observations can be received at regular intervals of time (scans or data frames) or they can occur irregularly in time. Here, we will use the general term scan to refer to any set of input measurements that were all produced at the same time. Then, the input observations are considered for inclusion in existing tracks and for initiation of new tracks. First, a gate, based upon the maximum acceptable measurement plus tracking prediction error magnitudes, is placed around the predicted track. Only those observations that are within the track gate are considered for update of the track. When closely spaced targets produce closely spaced observations there will be conflicts such that there may be multiple observations within a track's gate and an observation may be within the gates of multiple tracks. This is handled by the Observation-to-Track Association and Track Maintenance functions.

Fig. 2, also taken from [1], shows a typical conflict situation in which track gates are placed around the predicted positions (P1, P2) of two tracks, and three observations (O1, O2, O3) satisfy the gates of either (or both) of the tracks. The conventional data association method is denoted the global nearest neighbor (GNN) approach. It finds the best (most likely) assignment of input observations to existing tracks, which for example, would probably be O1 to track 1 and O2 to track 2. The term global is used to refer to the fact that the assignment is made considering all possible (within gates) associations

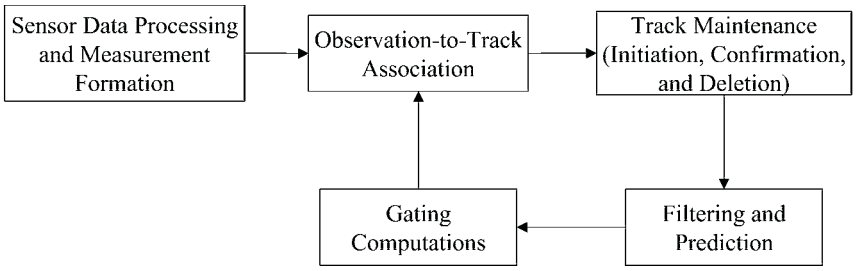


Fig. 1. Basic elements of a conventional MTT system.

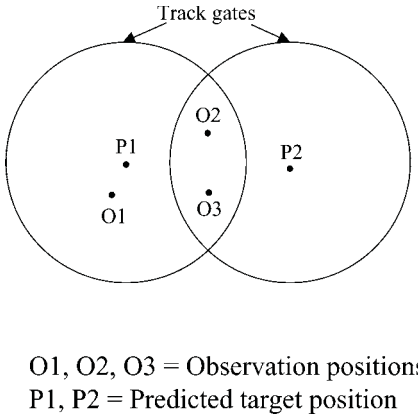


Fig. 2. Example of typical data association conflict situation.

under the constraint that an observation can be associated with at most one track. This distinguishes GNN from the archaic (but apparently still used in some systems) nearest neighbor (NN) approach in which a track is updated with the closest observation even if that observation may also be used by another track.

Only those tracks that are included in the best assignment are kept. Unassigned observations, in this case O3, initiate new tracks. Track confirmation and deletion are typically determined by rules, such as 3 detections in 4 frames of data for confirmation and N consecutive misses (typically $N = 4$ to 7) for track deletion.

Inherent in the standard GNN assignment is the assumption that an observation was produced by a single target. Tracks that do not share any common observations will be defined to be compatible. Thus, only compatible tracks can appear in the same assignment solution. Relaxation of this constraint to allow for the provision of unresolved targets that produce a single measurement will be discussed later.

Once observations are assigned to tracks, these tracks are updated during the filtering process. Conventional systems typically use a single Kalman filter. However, as discussed below, modern systems should use the interacting multiple model (IMM) approach in which several Kalman filters, tuned to different types of target maneuver, are run in parallel [1, 2]. Finally, all tracks are predicted to the time of the next set of measurements. The Kalman filter prediction covariances provide the uncertainty, in the predicted state estimate, that is required for the gating and association processes.

The GNN approach, which only considers the single most likely hypothesis for track update and new track initiation, only works well in the case of widely spaced targets, accurate measurements, and few false alarms in the track gates. For example, from results given in [1], even if the true target return is present, a single uniformly distributed false alarm in a three dimensional radar measurement space (typically range and 2 angles) reduces the probability of correct association to about 0.85. Thus, in about one out of 6 track update attempts a false alarm will be chosen rather than the correct target return. For the more usual case of multiple closely spaced targets and where missed true target detections occur, the probability of false track update is much worse. Experience indicates that often a single false update will lead to track loss and two consecutive false updates will usually lead to track loss.

The fact that misassociation represents an additional error source for a Kalman filter tracker was recognized in the very early stages of tracker development [3–5]. One approach that was proposed to improve GNN performance was to increase the Kalman filter covariance matrix to reflect this additional source of uncertainty [3, 4]. A similar approach, based upon work by R. Fitzgerald, also reduces the gain for uncertain association conditions, Sec. 6.12.1 of [1].

A second approach, which has become the Joint Probabilistic Data Association (JPDA) method, "hedges" for uncertain association conditions by allowing a track to be updated by a weighted (by probability) sum of all observations in its gate [2, 5]. This also means that an observation may contribute to the update of more than one track. Thus, for the example of Fig. 2, observations O1, O2, and O3 would all contribute to the update of track 1 and observations O2 and O3 would contribute to the update of both tracks.

Both the augmented GNN approach and the JPDA method increase the Kalman filter track covariance matrix to account for the association uncertainty. However, as illustrated in [6], increasing the Kalman

filter covariance matrix to account for uncertain association can exacerbate the problem whereby an increased covariance matrix leads to even more false observations in the track gate, etc. Also, the JPDA method suffers from a coalescence problem whereby tracks on closely spaced targets will tend to come together [7]. For example, from Fig. 2, since observations O2 and O3 will contribute to the updates of tracks 1 and 2, these tracks will be drawn together.

The problems that result from relatively simple upgrades to the GNN method and the recent dramatic increases in computational capabilities have led to a near universal acceptance of the multiple hypothesis tracking (MHT) approach as the preferred data association method for modern systems. MHT is a deferred decision logic in which alternative data association hypotheses are formed whenever observation-to-track conflict situations, such as shown in Fig. 2, occur. Then, rather than choosing the best hypothesis or, in effect, combining the hypotheses as in the JPDA method, the hypotheses are propagated into the future in anticipation that subsequent data will resolve the uncertainty.

Sections II and III will discuss the basic principles and commonly used implementations of MHT. Section IV discusses how modern filtering techniques (in particular IMM) can be combined with MHT. Section V outlines some important current applications of MHT and Section VI gives areas of development and extension.

II. MHT BASICS

The manner in which MHT forms multiple hypotheses and manages these hypotheses is illustrated by again referring to the example given in Fig. 2 and by referring to the overall structure shown in Fig. 3. As an example, assume that tracks T1 and T2 with predicted positions P1 and P2, represent a hypothesis (H_1) prior to the receipt of the three observations (O1, O2, O3) on the current scan. Then, there are 10 feasible hypotheses that can be generated from the initial single hypothesis. For example, the two most likely hypotheses would both update T1 with O1 but would update T2 with either O2 or O3. Another, unlikely but feasible, hypothesis would be that all observations represent new sources (false alarms or other previously undetected targets) so that neither T1 nor T2 would be updated and all observations would start new tracks.

Reid's Algorithm

Although Singer, Sea, and Housewright [8] introduced the basic idea of propagating multiple hypotheses for a single target in a false alarm background, Reid [9] first developed a complete algorithmic approach. Reid's algorithm defines a systematic way in which multiple data (observation-to-track) association hypotheses can be formed and evaluated for the problem of multiple targets in a false alarm (and/or clutter) background. Again using the example of Fig. 2, Reid's algorithm is illustrated by defining H_1 to be the hypothesis containing T1 and T2 before the receipt of the three observations. Next, define a newly formed track

$$\text{T3 (T1, O1)} = \text{track 3 formed from the association of T1 with O1}$$

with similar definitions for T4 (T2, O2) and T5 (T2, O3). Also define NT1, NT2, and NT3 to be the new tracks initiated from O1, O2, and O3. Then, 3 of the feasible 10 hypotheses that can be formed are

$$\begin{aligned} H_1&: \text{ T1, T2, NT1, NT2, NT3} \\ H_2&: \text{ T3, T4, NT3} \\ H_3&: \text{ T3, T5, NT2} \\ &\vdots \end{aligned} \tag{1}$$

Tracks are defined to be compatible if they have no observations in common. As illustrated by the example above, assuming T1 and T2 share no observations, MHT hypotheses are composed of sets of compatible tracks. Again note, as discussed in more detail later, the formulation can ideally be expanded in order to address the problem of closely-spaced unresolved targets that may produce a single measurement that should be assigned to the multiple tracks that may have been formed on these unresolved targets. Using Reid's algorithm approach, hypotheses are carried over from the previous scan. Then, on the receipt of new data, each hypothesis is expanded into a set of new hypotheses by considering all observation-to-track assignments for the tracks within the hypothesis. Again, as new hypotheses are formed, the compatibility constraint for tracks within a hypothesis is maintained.

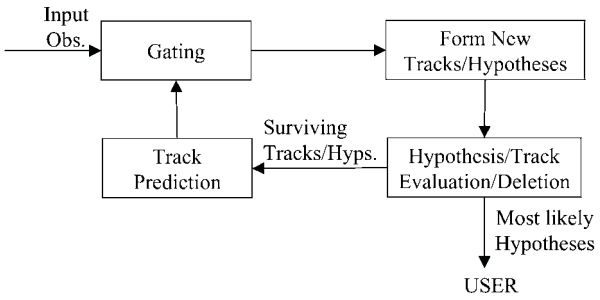


Fig. 3. MHT logic overview.

Track and Hypothesis Evaluation

The evaluation of alternative track formation hypotheses requires a probabilistic expression that

includes all aspects of the data association problem. These aspects include the prior probability of target presence, the false alarm density, the detection sequences and the dynamic (kinematic) consistency of the observations contained in the tracks. Reid [9] presents such a probabilistic expression. A mathematically equivalent, but computationally preferable, approach is the log-likelihood ratio, LLR (or track score) first proposed in the pioneering paper by Sittler [10], later detailed in [11] and summarized below.

A likelihood ratio (LR) for the formation of a given combination of data (including a priori probability data) into a track can be defined using a recursive relationship that follows directly from Bayes' rule

$$\text{LR} = \frac{p(D \mid H_1)P_0(H_1)}{p(D \mid H_0)P_0(H_0)} \triangleq \frac{P_T}{P_F} \tag{2}$$

Hypotheses H_1 and H_0 are the true target and false alarm hypotheses with probabilities P_T and P_F, respectively, and D is the data, so that

$p(D \mid H_i)$ = probability density function evaluated with the received data under the assumption that H_i is correct

$P_0(H_i)$ = a priori probability of H_i (such as expected density of true targets in a given area for H_1)

Note that the inclusion of a priori probabilities in (2) means that LR might formally be defined to be a probability ratio. However, following the original formulation of [10], we will refer to it as a likelihood ratio.

A true target is most generally defined to be an object that will persist in the tracking volume for at least several scans. Thus, this definition includes objects, such as persistent clutter, that may not be of interest to the tracking system but that should be tracked in order to minimize their interference with tracks on targets of interest. False alarms (or false targets) refer to erroneous detection events (such as those caused by random noise or clutter) that do not persist over several scans.

It is convenient to use the log likelihood ratio (LLR) or track score [10, 11] such that

$$\text{LLR} = \ln[P_T \mid P_F] \tag{3}$$

Then, LLR can be directly converted to the probability of a true target through

$$P_T/P_F = \frac{P_T}{1 - P_T} = e^{\text{LLR}}$$
$$P_T = e^{\text{LLR}}/[1 + e^{\text{LLR}}] \tag{4}$$

Thus, the LLR (track score) is all that needs to be computed (and maintained) in order to assess the validity of a track. Finally, as discussed further below, the track score can be used directly for track confirmation as an application of the classical sequential probability ratio test (SPRT).

The track score, $L(k)$, at scan k, can be placed in a convenient recursive form [1, 11]

$$L(k) = L(k-1) + \Delta L(k)$$
$$\Delta L(k) = \begin{cases} \ln(1 - \hat{P}_D); & \text{no update on scan } k \\ \Delta L_u(k); & \text{track update on scan } k \end{cases} \tag{5}$$

The loss in track score when a detection opportunity is missed is a function of the expected probability of detection ($\hat{P}_D$). As discussed in more detail in [1, 11], the gain, ΔL_u, in track score upon update is a function of the residual error (the difference between the measurement and the prediction) and its covariance matrix, the expected density of false returns, as well as $\hat{P}_D$. In addition, if signal intensity (such as signal-to-noise ratio, SNR) is measured, it may also be used in the track score.

Given the individual track scores, the hypothesis score is the sum of scores of all tracks contained in that hypothesis. Then, given hypothesis scores, the hypothesis probabilities can be computed [1, 11]. Finally, a track may be contained in multiple hypotheses so that its probability is the sum of probabilities of all hypotheses which contain it. For the example of (1), the probability of T3 would be the sum of probabilities for hypotheses H2, H3 and all other hypotheses that contain it.

To summarize, relatively simple computations can be performed to determine hypothesis and track probabilities. A theoretical objection that may be raised is that in order to compute these probabilities, such as through the track score as defined above, it is typical to assume very approximate Gaussian models for target dynamics and measurement error statistics, uniform distributions for false alarms (clutter and noise) and new targets and a nominal $\hat{P}_D$. However, all developers of practical MHT systems make essentially the same assumptions and, as discussed further below, results show that MHT with these assumptions performs substantially better than any other developed approach.

Practical Issues

As illustrated by the simple example given above, there is clearly a potential combination explosion in the number of hypotheses (and tracks within those hypotheses) that an MHT system can generate. Thus, a number of techniques have been developed to keep this potential growth in check. These techniques,

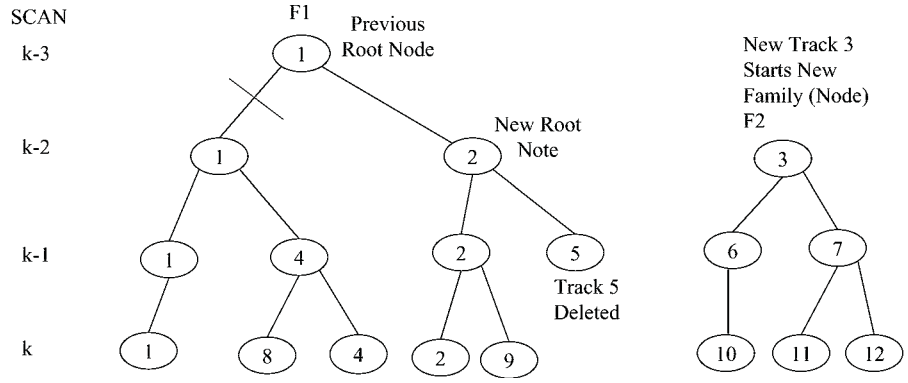


Fig. 4. Family (node) structure with N-scan pruning.

outlined next, include clustering, hypothesis and track pruning (deletion), and track merging.

The operation of clustering is performed to reduce the number of hypotheses that must be generated and evaluated. Clusters are collections of tracks that are linked by common observations. A cluster can include tracks that do not share observations directly. Thus, if track 1 shares an observation with track 2 and track 2 shares an observation with track 3, all three tracks are in the same cluster.

Clustering, in effect, decomposes a large problem into a set of smaller problems. Once clustering has been performed, the processing within each cluster can be done independently from other clusters. Thus, processing efficiencies can be achieved using a parallel processing structure whereby the processing for each cluster can be assigned to a separate processor. Then, within each cluster, hypotheses are evaluated and low probability hypotheses and tracks are deleted.

The key principle of the MHT method is that difficult data association decisions are deferred until more data are received. Thus, an important implementation feature used by all MHT developers is the family (or node) structure illustrated in Fig. 4. This structure provides a convenient mechanism for implementing a deferred decision logic and for presenting a coherent output from the MHT tracker to the user.

Fig. 4 shows how MHT track branches are formed and illustrates how a convenient structure for track pruning can be defined. Using this structure, a family is defined as a set of tracks with a common root node. Alternatively, what we define to be a family (of tracks all emanating from a single ancestor, or root node) can also be considered to be a target tree. Each branch represents a different data association hypothesis for the target and nodes are defined to be points where one track forms two or more branches. Because each branch track within the family (target tree) has at least one common node (the root node), these tracks are all incompatible with each other and can represent at most one target.

Based upon current data (including scan k), irrevocable decisions are made in the past (for the example this is scan $k-2$). Specifically, one approach finds the tracks from families F1 and F2 that are in the best current (scan k) hypothesis and goes back N scans (in this case $N = 2$) to establish a new root node. For example, if track 2 of F1 is in the best hypothesis, the new root node is track 2 at scan $k-2$. Subject to other tests, beyond the scope of this paper, if F2 does not have a track in the best hypothesis, the entire family would be deleted.

Note that the entire branch of F1 leading to tracks 1, 4, and 8 has been deleted. However, track 9 has been maintained even though track 2 was in the best hypothesis. This method is denoted N-scan pruning (or can be defined as an N-scan sliding window) and we have, for convenience of presentation, chosen $N = 2$ for the example. In practice, our experience is that N should generally be chosen to be at least 5. Also, rather than scans in the past, the decision is probably best made using N observations in the past but the basic principle is the same. Firm decisions are made in the past based upon later data.

Fig. 5, adapted from [12], shows the relationship between the families (track hypotheses for a given target) and the global (multiple track) hypotheses that are formed as collections of compatible tracks. A global hypothesis is formed by choosing at most a single track from each family.

The family representation of Fig. 4 also provides a convenient way to present MHT data to a user who typically wants one track per target, not a set of alternative tracks with probabilities. The tracks in the output trackfile are linked to the families and, at any given time, the most likely track in the family is presented to the user. This can lead to some apparent inconsistencies in the output as MHT branch probabilities change with the receipt of more data. For example, it may be that track 1 of F1 was the most likely track at scan $k-1$ but track 2 is the most likely track at scan k. Thus, a possible alternative is to provide an average state estimate, computed using the branch track probabilities, along with a

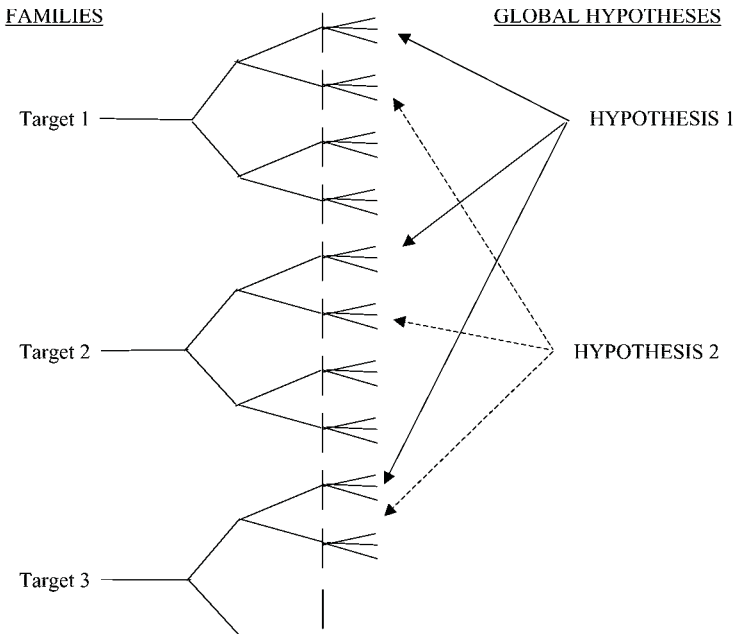


Fig. 5. Formation of hypotheses from tracks in families.

covariance that reflects the spread in the branch track state estimates. This approach is particularly useful for an agile beam radar system, discussed below, for which data association uncertainty should be used in the resource allocation logic.

III. ALTERNATIVE MHT IMPLEMENTATIONS

Although the same basic principles and mathematical models apply to all, there are several different approaches to MHT implementation. The first (hypothesis-oriented) approach follows the original work of Reid, outlined above. The computational feasibility of this approach has been greatly enhanced by the use of Murty's algorithm [13] to more efficiently generate hypotheses [14]. An alternative, track-oriented approach [1, 12] does not maintain hypotheses from scan to scan. As tracks are updated on each scan they are reformed into hypotheses. An innovative implementation of the track-oriented approach is the multidimensional (or multiple frame) assignment method [15, 16]. Finally, a Bayesian MHT approach has been proposed by van Keuk and Koch and associates [6, 17, 18]. The methods are briefly summarized below.

m-Best Implementation of Reid's Algorithm

As illustrated above, Reid's algorithm forms a large number of hypotheses that are collections of compatible tracks. These hypotheses are carried from one scan to the next where newly received observations are used to update the tracks in different ways. Thus, each hypothesis carried from the previous scan may give rise to many new hypotheses (most of which will later be discarded based upon low probability) as the tracks contained within the hypothesis are updated in different ways. This potential explosion of new hypotheses that may result from an indiscriminate expansion of the old hypotheses has been a barrier to the practical implementation of Reid's algorithm. Thus, a method to only generate "good" hypotheses is required and has been provided by the work of Cox et al. [14].

As discussed in [14], an efficient implementation of Reid's algorithm can be achieved using Murty's method for finding the m-best solutions to the assignment problem. Using this approach, given $m_p(k-1)$ hypotheses from the previous scan, the number of hypotheses formed on the current scan can be limited to $m(k)$ when m is an input parameter that could be set a priori or, presumably, could be chosen adaptively. The important principle is that the generation of many unconsequential, low probability hypotheses, that resulted from earlier implementations of Reid's algorithm, is avoided.

Track-Oriented MHT

The track-oriented approach recomputes the hypotheses using the newly updated tracks after each scan of data are received. Rather than maintaining, and expanding, hypotheses from scan to scan, the track-oriented approach discards the hypotheses formed on scan $k-1$. The tracks that survive pruning are predicted to the next scan k where new tracks are formed, using the new observations, and reformed into hypotheses. Except for the necessity to delete some tracks based upon low probability or N-scan pruning described above, no information is lost

because the track scores, that are maintained, contain all the relevant statistical data. The basic, currently unresolved, issue is whether it is more efficient to expand the old hypotheses using Murty's method or to reform the hypotheses using the updated tracks and their compatibilities with other tracks.

A strong argument for the track-oriented approach to MHT can be made by noting that the combinatories of hypothesis formation are such that there are typically many more hypotheses formed than tracks. Typically, for difficult scenarios, there may be several thousand comparable hypotheses formed from several hundred tracks in a cluster. Then, the process of maintaining a thousand (or more) hypotheses and expanding these hypotheses using Murty's method to find the best thousand new hypotheses may be prohibitive. On the other hand, our experience with track-oriented MHT has shown that several hundred tracks can easily be maintained and expanded into new hypotheses for difficult scenarios. Typical computational results for a difficult scenario with 100 closely spaced targets and a high radar update rate indicate the feasibility of real-time operation for a track-oriented MHT [19]. This study was performed using a single 866 Mhz Pentium III computer. Newer computers and/or parallel processing with several computers would allow real-time tracking for even more difficult scenarios.

Our implementation uses a relatively simple set of heuristic search methods, based upon a breadth-first method described in [1] and the A* search method described in [20]. The multiframe assignment (MFA) method, outlined next, represents a potentially more accurate and efficient implementation of track-oriented MHT.

Multi Dimensional (Multiframe) Assignment

Deb [15] and Poore [16] and their associates independently recognized that the MTT data association problem can be placed in a form where a multi dimensional assignment approach that uses the Lagrangian relaxation method is directly applicable. Like track-oriented MHT, this approach forms and maintains tracks from scan (frame) to scan and reforms tracks into hypotheses after each new scan of data are received. It also uses a sliding window approach which is similar to the N-scan pruning method used in conventional MHT and illustrated in Fig. 4. The unique feature of this method is the manner in which a Lagrangian relaxation method is used to find the most likely hypothesis or a set of the m-best hypotheses [21].

The input is a set of tracks with their scores and their compatibilities with other tracks. Again, two tracks are defined to be incompatible, and thus cannot be in the same hypothesis, if they share one or more observations. The process of arranging these tracks into hypotheses can be formulated as an optimization problem with the goal of maximizing the hypothesis score (sum of all track scores in hypothesis) with the constraints that no tracks in the hypothesis share observations.

The basic principle of the Langrangian relaxation approach is to replace constraints (in this case that an observation can be used by at most a single track) by Lagrange multipliers in the objective function (in this case the sum of track scores) used in the maximization. The "art" of this method involves the proper choice of Lagrange multipliers so that the solution formed from maximizing the objective function approaches the best feasible solution, in which each observation is used by at most a single track.

This optimization is very complex and requires sophisticated mathematics but we will (at least attempt to) summarize the basic principles. Two solutions to the hypothesis formation problem are obtained with cost defined to be the negative of score. The first solution, defined to be the relaxed or dual solution, may not satisfy the constraints (that an observation should be used once and only once). However, Lagrange multipliers are introduced into this solution and are chosen so that constraint violations are, effectively, given high costs. Thus, the number of constraint violations should be reduced over time with successive iterations of the method.

A second solution, denoted the recovered or primal solution, is obtained from the dual solution by enforcing the constraints. For example, one method for obtaining this solution starts with the assignment of the first two scans of data that was obtained by the dual solution. Then, it adds observations from the later scans by solving an assignment matrix, that enforces the constraints, on each later scan. Thus, a feasible, but likely suboptimal, solution is obtained.

The costs of the dual solution, $q(\underline{u})$, where $\underline{u}$ represents the Lagrangian multipliers, and the primal solution, $v(\bar{z})$, represent bounds on the cost, $v(z)$, of the true, but unknown, solution

$$q(\underline{u}) \leq v(z) \leq v(\bar{z})$$

where z and $\bar{z}$ are the set of binary variables that define which tracks are included in the true and the primal solutions, respectively [1, 15, 16].

Successive iterations are performed by using updated Lagrange multipliers in an attempt to increase $q(\underline{u})$ and decrease $v(\bar{z})$ and a stopping rule is defined so that the feasible primary solution is accepted when $q(\underline{u})$ and $v(\bar{z})$ are "close enough," or when time runs out and a solution is required.

The multiscan assignment method outlined above can be used to implement the N-scan pruning method used for track-oriented MHT, as illustrated in Fig. 4. Performing N-scan pruning requires a solution to the $N + 2$ scan assignment problem.

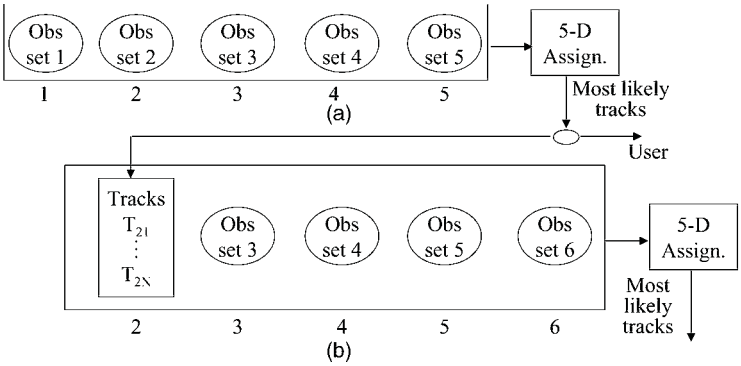


Fig. 6. Implementation of $N = 3$ scan pruning using 5D assignment. (a) First five scans of observations. (b) Tracks formed from first two scans and four scans of observations. (Adapted from notes by A. B. Poore.)

Fig. 6 illustrates the process for $N = 3$ (five-scan assignment). Referring to Fig. 6(a), initially five scans of data are collected with the observations on scan 1 effectively being the initial root nodes. The output of the 5D assignment problem will be a set of tracks in the most likely (solution) hypothesis. These tracks are traced back $N = 3$ scans to their root nodes (tracks) on scan 2. Then, all tracks that were in existence on scan 2 and that do not have one of these root node tracks as their ancestor on scan 2 are deleted.

As illustrated in Fig. 6(b), the root nodes are taken to be the tracks on scan 2 that were the ancestors of the tracks in the most likely hypothesis. The next scan of data is used to update the tracks that survived pruning on the previous scan. The process continues with, in general, new observations received on scan $k + 1$, a sliding window of observations received on scans $k, k-1, \ldots, k-N+1$ and the root node tracks on scan $k-N$. This process is illustrated in Fig. 6(b) for $k = 5$ and $N = 3$. See [22, 23] for more details on efficient implementation.

Bayesian MHT

The technique denoted Bayesian MHT [6, 17, 18] is designed to more closely represent the probability density functions (PDF) of alternative data association hypotheses. The PDF is represented as a Gaussian mixture that represents the joint distribution of the targets under track. Thus, the method effectively requires knowledge, or assumption, of the number of targets in track. Reference [18] addresses the problem of estimating this number.

IV. MHT AND MULTIPLE MODEL FILTERING

It is widely accepted that accurate tracking of dynamic targets requires the use of multiple Kalman filter models. The basic idea of all multiple model approaches, as applied to tracking maneuvering targets, is that maneuvers are typically abrupt deviations from basically straight-line target motion. Because this process is very difficult to represent with a single maneuver model, multiple models, representing different potential target maneuver states, are run in parallel and continuously evaluated using filter residual histories. Bayes' rule and the residuals are used to determine the probabilities of validity of the models. The output is then typically a probability-weighted composite of the individual filters.

There are two basic approaches that can be used to combine MHT with multiple model filtering. The first, outlined in [12], is to add a set of maneuver hypotheses to the MHT data association hypotheses. Thus, an additional set of hypotheses which differ in target dynamics history will be formed. Use of interacting multiple model (IMM) filtering appears to be difficult for this approach.

IMM filtering has become generally accepted as the best method for using multiple filter models [2]. The unique feature of the IMM approach is the manner in which the state estimates and the covariance matrices are combined via the process defined to be mixing. The basic principle is that the currently more accurate (as determined by the computed model probabilities) models transfer their state estimates to the less accurate models. For example, in the case of a maneuvering target, the state estimates from the maneuver models, that should follow the target motion fairly well, are transferred to the nonmaneuver filter that otherwise would develop a large lag.

In order to conveniently do IMM filtering within an MHT framework, we believe that it is most convenient to define tracks according to their data association history. A similar approach is presented in [24]. Then, the track score (or probability) is computed using all component IMM filter models. Thus, hypothesis formation and pruning are done on the composite tracks, containing contributions from all IMM filter models, rather than on the IMM model tracks independently. Using this approach the mixing process is conveniently done for each track, rather than requiring mixing across tracks.

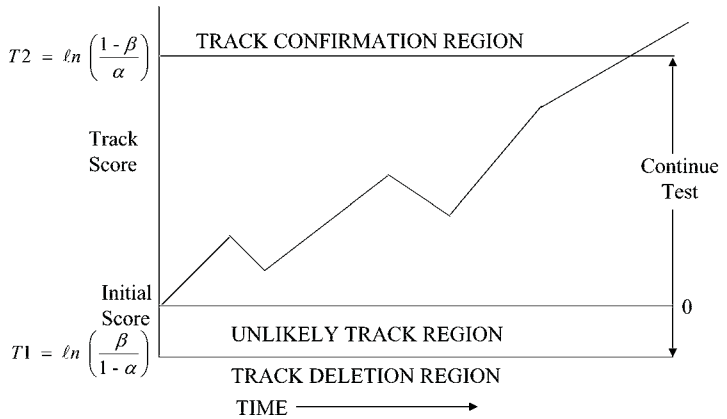


Fig. 7. Score function approach as an application of SPRT.

Given that hypothesis formation and pruning are performed using all the IMM filter models for each track, the next issue is how to perform gating and how to update the combined track score. One approach is to form a composite track state estimate and covariance matrix before gating and to perform gating using the composite quantities.

The composite state estimate and covariance matrix are formed from weighted (by the filter model probabilities) sums of the state estimates and covariance matrices of the individual filters. Alternatively, using a second approach, each filter model can be used separately for gating. In this case, there will be separate state estimates (and corresponding covariance matrices) that will be individually compared with the candidate observations. The observation-to-track gating test will then be satisfied if the gating test is satisfied for any filter model. Similarly, the track score can be computed from the composite track residual (and residual statistics) or a combined track score can be computed using the individual residual data from the different IMM filter models.

It has been our experience that the second approach is preferred. During times of nonmaneuver, the composite state and covariance matrix (and resulting gate) may become so heavily weighted towards the nonmaneuver models that an abrupt target maneuver can lead to track loss. Finally, the extension to the track score required when multiple filter models are used is straight forward [1].

V. MHT APPLICATIONS

The actual practical implementation of MHT has been impeded by the, currently incorrect [19], perception that its complexity precludes real-time application. Also, the security restrictions that surround technologies, such as tracking, being developed for current military applications and company proprietary policies have greatly restricted the ability of MHT tracker developers to publish and compare their results, and to share ideas. Another problem is that very little comparative study of MHT performance, versus that of alternative tracking methods, has been reported in the tracking literature. However, the brief summary of reported comparative studies, such as [25], given in [1] and the growing acceptance of MHT among those in the tracking community clearly indicate that MHT is the currently preferred method for difficult tracking problems. We next summarize some important applications with which the author is familiar.

Track Confirmation and Maintenance for Dim Targets in Clutter

As illustrated by Fig. 7, and discussed further in [1, 18, 26], a confirmation test that uses the track score (LLR) is essentially an application of the classical sequential probability ratio test (SPRT). Then, as detailed in [1], the choice of confirmation and deletion thresholds (T1 and T2, respectively, shown in Fig. 7) can be related to tracking requirements (such as the number of false tracks allowed per hour) through the parameters α = false track confirmation, and β = true track deletion probability. This approach also provides a convenient analysis tool for preliminary system design [1, 26].

The application of SPRT theory to MHT track confirmation assumes that false alarms are uncorrelated in time. In practice, such as for tracking targets against a background of ground clutter, clutter returns tend to be correlated in time. In this case, it is best to maintain tracks on the stationary sources of ground clutter that produce the returns. Thus, special logic using motion or signal characteristics is developed to inhibit the output of these tracks to the user [27].

A number of studies, discussed further in [1], have indicated that an MHT tracker will provide performance that is comparable to the conventional, single hypothesis (GNN) method at 10 to 100 times

the false alarm density of the GNN method. This allows a system using MHT to operate at a lower detection threshold, in order to detect and track dim targets [28]. However, the comparative study given in [29] showed that a well-designed track-before-detect (TBD) approach that, in effect, combines the detection and tracking functions, will confirm tracks on nonmaneuvering dim targets at much lower SNR (about 4–5 dB lower for the cases considered in [29]).

Agile Beam Radar

Efficient allocation of radar resources is one of the major issues in the design of an agile beam (or electronically scanned) radar tracking system. Moreover, following [1, 30–32] use of an MHT tracker can greatly enhance the effectiveness of an allocation scheme. Specifically, the combined use of MHT data association and IMM filtering and prediction methods provides the most accurate estimates of tracking error that are required for efficient sensor allocation. The IMM filter model probabilities and covariance matrices provide estimates of the error due to target maneuver and the potential error due to data association is computed from alternative MHT hypotheses. Further discussions of the radar benchmark study that demonstrated the effectiveness of an IMM/MHT solution to the agile beam radar resource allocation problem are given in [30] and the Introduction of [33].

Missile Defense Systems

Post boost tracking scenarios for missile defense systems are characterized by a large number (potentially hundreds or even thousands) of closely spaced objects. These objects are deployed over time by the post boost vehicle (PBV or bus) and very accurate tracks are required for impact point prediction. In addition, track purity (defined to be the proportion of observations in a track that were produced by the same source) must be high so that discrimination can be successfully performed. Discrimination methods employ Bayesian or Dempster-Shafer reasoning to determine the target type using the characteristics (such as intensity profile) of the measurements in the track as examined over time. For example, it is very important to discriminate between the lethal reentry vehicle (RV) and decoys that are employed to "trick" the tracking and discrimination algorithms.

Both radar and space-based infrared (SBIR) tracking systems are being developed. Given the stringent tracking requirements, it is generally accepted that MHT should be used for both types of sensors and there are several special features, outlined next, that must be addressed for these applications.

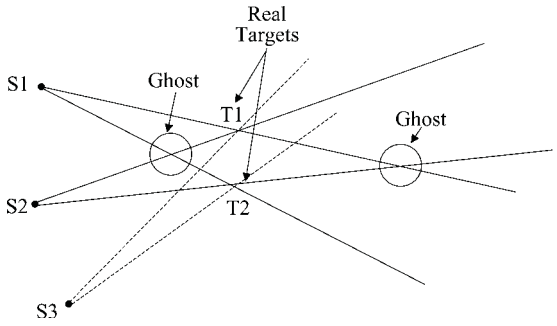


Fig. 8. Triangulation with angle tracks leads to false intersections that can be resolved with MHT and later data.

First, objects (RVs and various types of decoys) are deployed from the PBV with basically the same velocity as the PBV. Thus, a "warm start" track initiation (or spawning) procedure is used in order to quickly obtain the required tracking accuracy.

Referring to Fig. 2, observations O2 and O3 would be candidates for "warm start" new track initiation (in addition to the hypotheses that they update T2 or possibly T1 also). Thus, the new tracks would be given a position estimate based upon the measurement and a velocity estimate based upon the velocity of the parent tracks (either T1 or T2 or an average of the two) for which they satisfied a gating relationship. For the SBIR system, in which only angle measurements are available, the range from the sensor platform, as well as the platform position, will be used along with the measured angles to form the initial position estimate. The initial "warm start" track filter covariance matrices are defined using the measurement error variances and the parent track range (for SBIR) and velocity error covariance matrices. Also, terms to account for the potential differences in velocity of the newly detected (resolved) object and the parent track object are added. Finally, once spawning occurs, MHT processing will take over to determine, using later data, which observations (in our example O2 or O3) should start new tracks and which should update existing tracks (or possibly be discarded).

An additional source of data association uncertainty occurs for the angle-only measurements of an SBIR system. Tracks on targets that are separated from existing tracks (so that spawning cannot be accurately used) must be initiated by the triangulation process, illustrated in Fig. 8 and discussed further in [1]. Using this procedure, the intersections (or near intersections) of mono (angle-only) tracks from two platforms (S1, S2) are used to initiate stereo (3D position and velocity) tracks. The problem, shown in Fig. 8, is that, for closely spaced targets, there may be false intersections that form ghost tracks, as well as the correct intersection where the targets actually exist. Thus, an MHT approach is required so that all feasible tracks are maintained until either the evolving

TABLE I
Comparative Conventional and MHT Tracking Errors Referenced to an Idealized System

	Early Time		Intermediate		Late	
System	Position	Velocity	Position	Velocity	Position	Velocity
Conventional	3.3	2.9	2.8	5.0	3.0	3.3
MHT	1.7	2.1	1.4	2.0	1.5	1.2
Idealized	1.0	1.0	1.0	1.0	1.0	1.0

geometry or data from additional sensors (S3) allows the system to sort out the ghosts from the true target tracks.

In order to illustrate the advantages of MHT over conventional single hypothesis (GNN) tracking, Table I gives recent comparative results for a difficult SBIR application. Table I gives comparative 97 percent Monte Carlo simulation derived position and velocity errors for three tracking systems. The 97 percent level values were defined such that only 3 percent of the tracking error (averaged over multiple targets and multiple Monte Carlo runs) exceeded these values at the sampling times. A highly optimistic reference for Table I was an idealized system for which perfect observation-to-track association was performed. The observations were assigned target truth tags, which were used for the association, but the effects of unresolved targets and missed detections were included.

The MHT and conventional (GNN) tracking system RMS position and velocity tracking errors are normalized with respect to the idealized system errors. Results are presented at three times. The initial (early) time is when targets are beginning to become resolved so that by the last (late) time nearly all targets were resolved. Of course, the tracking errors decreased for all systems (even though some ratios increased) with time but the comparative advantage of the MHT system is clearly apparent over the entire scenario. Finally, note that the MHT errors closely approach those of the idealized system towards the end of the scenario while the conventional tracker errors remain at about 3 times the values for the idealized system.

Ground Target Tracking

Probably the most important, and challenging, current tracking application uses data from airborne (or spaced-based) sensors to track ground targets. Difficult target dynamics include move-stop-move and on and off-road target motion as well as closely spaced targets moving in groups (convoys). Sensor difficulties result from potentially long revisit times (greater than 10 sec.), obscured (by mountains or building) sensor line-of-sight, unresolved targets, out of sequence measurements in multiple sensor systems, and the fact that a radar operating in the standard ground moving target detection (GMTI) mode will not detect stopped (or slow moving) targets that cannot be distinguished from the ground clutter.

The difficulty of tracking ground targets has led to the consensus that multiple filter models, for on and off-road tracking, and MHT data association are required. For example, see [34–38] and Chapt. 6 of [33].

An example of the interesting challenges of the ground target tracking problem are targets that use move-stop-move motion in order to evade detection by GMTI radar. This necessitates the development of a special stopping target filter model and the inclusion of the hypothesis that a missing detection results from a stopped target, rather than a random miss or an incorrect track prediction [36–38]. In particular, the lack of detection can actually be used to infer target position by forming the hypothesis that a missed detection results from the fact that the target has stopped [37].

VI. MHT RESEARCH AND DEVELOPMENT AREAS

As stated by Daum several years ago [39], a major, mostly ignored, tracking problem is the presence of unresolved, or partially resolved, measurements produced by closely spaced targets. In closely-spaced target scenarios, such as aircraft flying in formation, an observation will often be produced by two, or more, targets. Thus, for these conditions, the standard MHT assumption that an observation was produced by a single target, and thus can only be assigned to a single track, must be modified. This issue becomes particularly important when tracking with sensors of different resolution capability, such as radar and IR. References [1, 40–42] and [33, ch. 4] present methods that are applicable to the extension of MHT to include hypotheses that allow a potentially merged observation to update more than one track.

Another important area of research is the combination of MHT with group tracking. An example where combined group and MHT tracking will be required is the missile defense problem where large numbers of objects may be deployed from the PBV (bus) in a short time period [43, 44]. As discussed in [44], there may be time intervals, as the targets are first deployed, when the proliferation of closely spaced targets may cause the number of MHT hypotheses formed to become prohibitive. The

proposed solution [44] uses group tracking until the targets separate sufficiently to allow feasible MHT tracking of individual targets. The determination of when (and how) to make the transition from a group track to MHT tracking of individual targets is the major issue. Other applications where combined MHT and group tracking will be required for optimal performance include tracking formations of aircraft [18] and convoys of ground moving targets [45].

As shown in [28], the track-before-detect (TBD) approach may significantly outperform MHT for the task of track confirmation of dim nonmaneuvering targets. However, the TBD approach, which essentially integrates signal intensity along a set of potential, nearly straight line paths, is not applicable to highly maneuvering targets and has questionable applicability in dense target environments. Thus, the goal is to combine use of the powerful TBD methods, such as the dynamic programming algorithm, DPA [1, 46] and Bayesian tracking [28, 47], for detecting and tracking widely-spaced dim targets with IMM/MHT techniques that are most applicable to maneuvering targets in dense environments. Reference [46] discusses a combined DPA/MHT tracking system.

Standard track and hypothesis evaluation methods currently only use metric (measured position, range rate, etc.) and possibly intensity (measured SNR, etc.) data. The increased capability of sensors to measure other feature data, such as high range resolution (HRR) and jet engine modulation (JEM) radar measurements, and the development of multiple sensor tracking systems dictate that features, attributes and target classification/ID should be used to improve data association. This is particularly true for the problem of maintaining tracks on high priority targets for the ground target tracking problem [48].

A basic issue is how to weight attribute/ID data versus metric measurements. For example, a radar return might contain JEM information regarding engine type that is consistent with other target type information contained in the track, but the measured range rate may differ significantly from the track's predicted range rate. How should the observation-to-track score reflect these two different, and possibly inconsistent, data sources? As outlined in [1, 49] a mapping to likelihood (or LLR) is required. However, to the author's knowledge this approach has not yet been implemented for a practical system.

As discussed further in [33, ch. 1], the multisensor distributed tracking problem is of great practical importance. One basic issue/goal for a distributed platform system is to attempt to ensure that all platforms have a Single Integrated Air Picture (SIAP) so that, for example, track 1 on platform 1 represents the same target as track 1 on platform 2, etc. Methods for maintaining SIAP for conventional (single hypothesis) tracking use an associated measurement report (AMR) that is sent from the platform that receives a measurement. The AMR contains the association decision, made by the platform that produced the measurement, which is broadcast to all other platforms in the network who update their tracks accordingly, without any further association logic being performed.

Maintaining SIAP for an MHT system is much more difficult because multiple current association hypotheses are maintained so that, as shown in Fig. 4, final irrevocable decisions are delayed. In the meanwhile, as the result of imperfect communication (missing and out-of-sequence data), the family structures on the different platforms may diverge. Also, track initiation and confirmation decisions may differ as different platforms use different sequences of measurements to initiate duplicate tracks on the same target. This is an important area of current research with approaches discussed in [50–52] and [33, ch. 6].

REFERENCES

[1] Blackman, S., and R. Popoli (1999)
Design and Analysis of Modern Tracking Systems.
Norwood, MA: Artech House, 1999.

[2] Bar-Shalom, Y., and X-R. Li (1995)
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.

[3] Nahi, N. (1969)
Optimal recursive estimation with uncertain observation.
IEEE Transactions on Information Theory, **IT-15**, 4 (July 1969), 457–462.

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

[5] Bar-Shalom, Y., and Tse, E. (1975)
Tracking in a cluttered environment with probabilistic data association.
Automatica, **11** (Sept. 1975), 451–460.

[6] Fleskes, W., and van Keuk, G. (1987)
On single target tracking in dense clutter environment-quantitative results.
In *Proceedings of 1987 International Radar Conference*, 130–134.

[7] Fitzgerald, R. J. (1985)
Track biases and coalescence with probabilistic data association.
IEEE Transactions on Aerospace and Electronic Systems, **21**, 6 (Nov. 1985), 822–825.

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

[9] Reid, D. B. (1976)
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, **21**, 1 (Feb. 1976), 101–104.

[10] Sittler, R. W. (1964)
An optimal data association problem in surveillance theory.
IEEE Transactions on Military Electronics, **8** Apr. 1964, 125–139.

[11] Blackman, S. (1986)
Multiple Target Tracking with Radar Applications.
Norwood, MA: Artech House, 1986.

[12] Kurien, T. (1990)
Issues in the design of practical multitarget tracking algorithms.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Norwood, MA: Artech House, 1990, ch. 3.

[13] Murty, K. G. (1968)
An algorithm for ranking all the assignments in order of increasing cost.
Operations Research, **16** (1968), 682–687.

[14] Cox, I. J., and Hingorani, S. L. (1996)
An efficient implementation of Reid's multiple hypotheses tracking algorithm and its evaluation for the purposes of visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **18**, 2 (Feb. 1996), 138–150.

[15] Deb. S., et al. (1997)
A generalized s-d assignment algorithm for multisensor-multitarget state estimation.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 2 (Apr. 1997), 523–538.

[16] Poore, A. B., and Robertson, A. J. (1997)
A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems.
Computational Optimization and Applications, **8**, 2 (Sept. 1997), 129–150.

[17] Koch, W. (1996)
Retrodiction for Bayesian multiple hypothesis/multiple target tracking in densely cluttered environment.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets*, SPIE Proceedings, **2759** (1996), 429–440.

[18] van Keuk, G. (2002)
MHT extraction and track maintenance of a target formation.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 1 (Jan. 2002), 288–295.

[19] Blackman, S., Dempster, R., and Reed, R. (2001)
Demonstration of multiple hypothesis tracking (MHT) practical real-time implementation feasibility.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2001*, SPIE Proceedings, **4473** (2001), 470–475.

[20] Pearl, J. (1984)
Heuristics: Intelligent Search Strategies for Computer Problem Solving.
Addison Wesley, 1984.

[21] Popp, R. L., Pattipati, K. R., and Bar-Shalom, Y. (2001)
An m-best s-d assignment for multiple target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001), 22–39.

[22] Poore, A. B. (1997)
Hot starts for track maintenance in multisensor-multitarget tracking.
Signal and Data Processing of Small Targets, SPIE Proceedings, **3163** (July 1997), 341–450.

[23] Poore, A. B., and Drummond, O. E. (1997)
Track initiation and maintenance using multidimensional assignment problems.
In P. M. Pardalos, D. Hearn, and W. Hager (Eds.), *Network Optimization*, New York: Springer-Verlag, 1997, 407–422.

[24] Torelli, R., Graziano, A., and Farina, A. (1999)
IM3HT algorithm: A joint formulation of IMM and MHT for multi-target tracking.
European Journal of Control, **5** (1999), 46–53.

[25] de Feo, M., et al. (1997)
IMMJPDA versus MHT and Kalman filter with nn correlation: Performance comparison.
IEE Proceedings, Radar, Sonar, Navigation, **144**, 2 (Apr. 1997), 49–56.

[26] Blackman, S., Dempster, R., and Broida, T. (1993)
Multiple hypothesis track confirmation for infrared surveillance systems.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 3 (July 1993), 810–823.

[27] Attili, J. B., et al. (1996)
False track discrimination in a 3-d signal/track processor.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1996*, SPIE Proceedings, **2759** (1996), 205–217.

[28] Chang, K. C., Mori, S., and Chong, C. Y. (1994)
Evaluating a multiple-hypothesis multitarget tracking algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **20**, 2 (Apr. 1994), 578–590.

[29] Barlow, C. A., and Blackman, S. S. (1998)
A new Bayesian track-before-detect design and comparative performance study.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1998*, SPIE Proceedings, **3377** (1998), 181–191.

[30] Blackman, S. S., et al. (1999)
IMM/MHT solution to radar benchmark tracking problem.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 2 (Apr. 1999), 730–738.

[31] van Keuk, G. (1995)
Multiplehypothesis tracking with electronically scanned radar.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 3 (July 1995), 916–927.

[32] Koch, W. (1999)
On adaptive parameter control for phased-array tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1999*, SPIE Proceedings, **3809** (1999), 444–455.

[33] Bar-Shalom, Y., and Blair, W. D. (Eds.)(2000)
Multitarget-Multisensor Tracking: Applications and Advances, Vol. 3, Norwood, MA: Artech House, 2000.

[34] Kurien, T., et al. (2000)
Discoverer-II GMTI tracker.
Prepared for Discoverer-II Joint Program Office, Alphatech Technical Report TR-984, Nov. 15, 2000.

[35] Kirubarajan, T., et al. (2000)
Ground target tracking with variable structure IMM estimator.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 1 (Jan. 2000), 26–46.

[36] Shea, P. J., et al. (2000)
Improved state estimation through use of roads in ground tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2000*, SPIE Proceedings, **4048** (2000), 321–332.

[37] Koch, W. (2001)
GMTI-tracking and information fusion for ground surveillance.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2001*, SPIE Proceedings, **4473** (2001), 381–392.

[38] Lin, L., Kirubarajan, T., and Bar-Shalom, Y. (2002)
New assignment-based data association for tracking move-stop-move targets.
Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, July 2002.

[39] Daum, F. E. (1994)
The importance of resolution in multiple target tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1994*, SPIE Proceedings, **2235** (1994), 329–338.

[40] Koch W., and van Keuk, G. (1997)
Multiple hypotheses track maintenance with possibly unresolved measurements.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 3 (July 1997), 883–891.

[41] Blair, W. D., and Brandt-Pearce, M. (1999)
NNJPDA for tracking closely-spaced Rayleigh targets with possibly merged measurements.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1999*, SPIE Proceedings, **3809** (1999), 396–408.

[42] Sinha, A., Kirubarajan, T., and Bar-Shalom, Y. (2002)
Maximum likelihood angle extractor for two closely spaced targets.
IEEE Transactioins on Aerospace and Electronic Systems, **38**, 1 (Jan. 2002), 182–203.

[43] Kovacich, M., et al. (1991)
An application of MHT to group to object tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 1991*, SPIE Proceedings, **1481** (1991), 357–370.

[44] Gadaleta, S., et al. (2002)
Multiple frame cluster tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2002*, SPIE Proceedings, **4728** (2002), 275–289.

[45] Koch, W. (2002)
On expectation maximization applied to GMTI convoy tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2002*, SPIE Proceedings, **4728** (2002), 450–460.

[46] Shaw, S., and Arnold, J. F. (1995)
Design and implementation of a fully automated OTH radar tracking system.
IEEE Proceedings of the 1995 International Radar Conference, May 1995, 294–298.

[47] Stone, L. D., Barlow, L. A., and Corwin, T. L. (1999)
Bayesian Multiple Target Tracking.
Norwood, MA: Artech House, 1999.

[48] Nguyen, D. H., et al. (2002)
Feature-aided tracking of moving ground vehicles.
In E. G. Zelnio (Ed.), *Algorithms for Synthetic Aperture Radar Imagery IX*, SPIE Proceedings, **4727** (2002), 234–245.

[49] Drummond, O. E. (2001)
Feature, attribute, and classification aided target tracking.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2001*, SPIE Proceedings, **4473** (2001), 542–558.

[50] Lu, S., Poore, A. B., and Suchomel, B. J. (2001)
Network MFA tracking architectures.
In O. E. Drummond (Ed.), *Signal and Data Processing of Small Targets 2001*, SPIE Proceedings, **4473** (2001), 447–457.

[51] Lu, S. (2001)
Network Multiple Frame Assignment Architectures.
Ph.D. thesis, Colorado State University, 2001.

[52] Dunham, D., Blackman, S., and Dempster, R. (2004)
Multiple hypothesis tracking (MHT) for a distributed platform system.
To appear *Signal and Data Processing of Small Targets 2004*, SPIE Proceedings, Apr. 2004.

Samuel S. Blackman received the B.A. degree in mathematics from the University of Hawaii, Honolulu, in 1960 and the M.S. degree in physics from the University of New Mexico, Albuquerque in 1963.

He has been employed by Hughes Aircraft Company (now Raytheon) since 1963 and has worked principally in the development of tracking systems. His current research interests include the applications of tracking and estimation techniques to ground targets.

Mr. Blackman is the author of *Multiple Target Tracking with Radar Applications*, Artech House (1986), *Design and Analysis of Modern Tracking Systems*, Artech House (1999), and numerous papers related to estimation and tracking. He is a member of Phi Beta Kappa.

A STAP Overview

WILLIAM L. MELVIN, Senior Member, IEEE

This tutorial provides a brief overview of space-time adaptive processing (STAP) for radar applications. We discuss space-time signal diversity and various forms of the adaptive processor, including reduced-dimension and reduced-rank STAP approaches. Additionally, we describe the space-time properties of ground clutter and noise-jamming, as well as essential STAP performance metrics. We conclude this tutorial with an overview of some current STAP topics: space-based radar, bistatic STAP, knowledge-aided STAP, multi-channel synthetic aperture radar and non-sidelooking array configurations.

Manuscript received March 17, 2003; revised June 23, 2003.

Refereeing of this contribution was handled by P. K. Willett.

Author's address: Georgia Tech Research Institute, Sensors & Electromagnetic Applications Laboratory, 7220 Richardson Rd., Smyrna, GA 30080, E-mail: (bill.melvin@gtri.gatech.edu).

I. INTRODUCTION

Moving target indication (MTI) is a common radar mission involving the detection of airborne or surface moving targets. The signal-to-noise ratio (SNR)—a characterization of the noise-limited performance of the radar against a target with radar cross section (RCS) σ_T at range r—is approximated as

$$\mathrm{SNR}(\phi,\theta) = \left(\frac{P_t G_t(\phi,\theta)}{4\pi r^2}\right)\left(\frac{\sigma_T}{4\pi r^2}\right)\left(\frac{A_e G_{sp}}{N_{in} F_n L_{rf}}\right) \tag{1}$$

where P_t is peak transmit power, $G_t(\phi,\theta)$ is antenna gain for direction (ϕ,θ), A_e is the effective receive aperture area, G_{sp} represents processing gains, N_{in} is the input noise power, F_n is the receiver noise figure and L_{rf} represents radio frequency (RF) system losses [1]. Assuming the noise is uncorrelated (white) and Gaussian, the probability of detection (P_D) is a one-to-one, monotonic function of both SNR and the probability of false alarm (P_{FA}). By maximizing SNR, the processor maximizes the probability of detection for a fixed probability of false alarm. In light of (1), the radar designer ensures detection of targets with diminishing radar cross section at farther range by increasing power-aperture $P_t A_e$. System constraints and cost limit the deployable power-aperture product.

Radar mounted on aerospace platforms must also mitigate the otherwise deleterious impact of ground clutter returns and jamming on moving target detection. We collectively refer to clutter and jamming as interference. Assuming Gaussian-distributed interference, P_D depends on both signal-to-interference-plus-noise ratio (SINR) and the specified value of P_{FA} in a manner analogous to the white noise detection scenario. Since SINR $\leq$ SNR, interference always degrades detection performance in comparison with the noise-limited case. Fig. 1

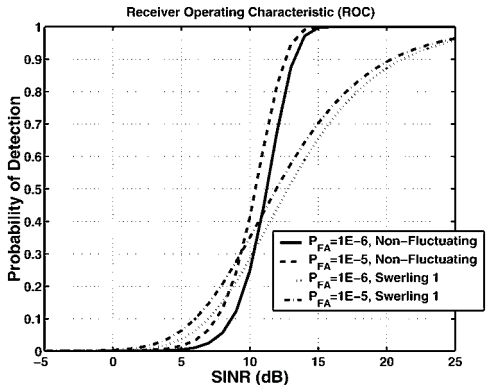


Fig. 1. Receiver operating characteristics for non-fluctuating and fluctuating targets.

shows the receiver operating characteristic (ROC) for non-fluctuating and Swerling 1 targets; the abscissa corresponds to output SINR. This figure clarifies the

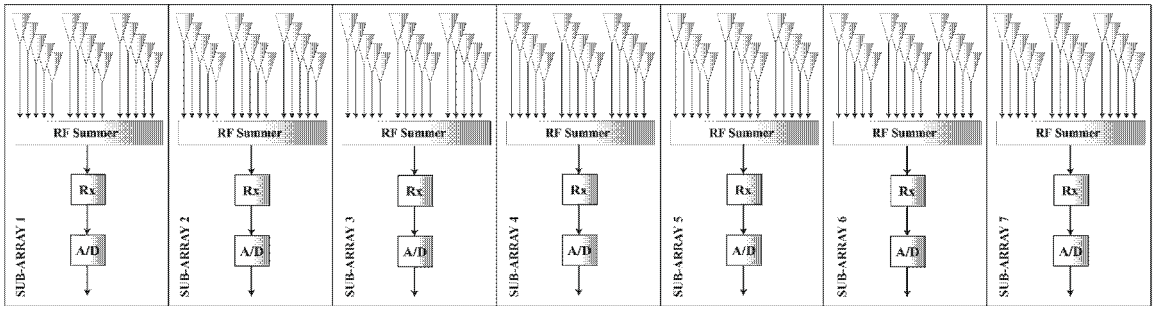


Fig. 2. Multi-channel ESA configured as a uniform linear array.

monotonic relationship between SINR, P_D and P_{FA}; for a fixed-value of P_{FA}, maximizing SINR is tantamount to maximizing P_D.

Spatial and temporal signal diversity, or degrees of freedom (DoF), greatly enhances radar detection in the presence of certain types of interference. Specifically, the appropriate application of space-time DoFs efficiently maximizes SINR when the target competes with ground clutter and barrage noise jamming. Ground clutter returns exhibit correlation in both spatial and temporal dimensions, while jamming is predominantly correlated in angle for modest bandwidth. Space-time adaptive processing (STAP) involves adaptively (or dynamically) adjusting the two-dimensional space-time filter response in an attempt at maximizing output SINR, and consequently, improving radar detection performance.

The objective of this paper is to develop the basic theory of space-time adaptive processing (STAP) as it relates to aerospace radar detection of moving targets in clutter-limited environments, and also consider some current trends in STAP research. Following groundbreaking adaptive array development by Howells [2], Applebaum [3] and Widrow et al. [4], Brennan and Reed introduced STAP to the airborne radar community in a seminal 1973 paper [5]. In years since the Brennan and Reed paper, STAP has been vigorously researched [6–67]. The recent advancement of high speed, high performance, digital signal processors makes fielding STAP-based radar systems possible on manned and unmanned airborne platforms and spaceborne satellites.

We organize this paper as follows. In Section II we describe the space-time properties of ground clutter and noise jamming. Section III formulates the STAP weight vector by considering the maximum SINR filter, while Section IV describes important STAP performance metrics. We then consider reduced-dimension (RD) and reduced-rank (RR) STAP formulations in Section V; RD/RR-STAP represent practical approaches for improving the statistical convergence of the adaptive filter, while RD-STAP also has the added advantage of reducing computational burden. Section VI concludes the paper by highlighting current trends in STAP research, including STAP application to space-based radar, bistatic radar, non-sidelooking arrays, and multi-channel synthetic aperture radar (SAR). We also overview knowledge-aided STAP in this final section.

II. PROPERTIES OF GROUND CLUTTER AND NOISE JAMMING

In this section we discuss the space-time characteristics of ground clutter and noise jamming. Toward this end, we first briefly describe spatial and temporal sampling and then consider two-dimensional space-time signals.

STAP systems generally employ an electronically scanned antenna (ESA) divided into multiple receive channels. A collection of antenna elements, known as a sub-array; the RF manifold, including RF summer; a receiver; and, an analog-to-digital converter (A/D) constitute the spatial channel. Fig. 2 depicts a multi-channel ESA configured as a uniform linear array (ULA). The multi-channel array spatially samples a propagating plane wave by effectively measuring the (nominally linear) phase difference among channels. Different phase variation corresponds to different signal direction of arrival.

Given an appropriate reference point, the respective phase at the mth spatial channel due to a propagating plane wave with a specific direction of arrival is

$$\gamma_{s/m} = \text{time delay} \times \text{radian frequency} = \tau_m \omega \quad (2)$$

where τ_m is the time-delay between reception of the plane wave at the reference point and the mth channel and ω is radian frequency. We calculate τ_m from the geometry given in Fig. 3 as follows:

$$\begin{aligned} \tau_m &= \frac{\mathbf{k}(\phi,\theta)\cdot\mathbf{d_m}}{c} \\ \mathbf{d_m} &= d_{x/m}\hat{\mathbf{x}} + d_{y/m}\hat{\mathbf{y}} + d_{z/m}\hat{\mathbf{z}} \\ \mathbf{k}(\phi,\theta) &= \cos\theta\sin\phi\hat{\mathbf{x}} + \cos\theta\cos\phi\hat{\mathbf{y}} + \sin\theta\hat{\mathbf{z}} \end{aligned} \quad (3)$$

where $\mathbf{d_m}$ is the position vector corresponding to the phase center of the mth channel, $\mathbf{k}(\phi,\theta)$ is a unit vector pointing normal to the plane wave, ϕ and θ represent azimuth and elevation angles, c is the

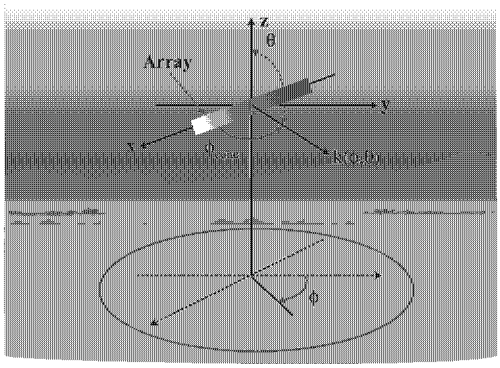


Fig. 3. Array geometry.

velocity of the propagating wave, $\{d_{x/m}, d_{y/m}, d_{z/m}\}$ are the Cartesian coordinates of the mth channel phase center, and $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ are unit vectors along the Cartesian axes. Considering a sidelooking ULA, and substituting (3) into (2), gives

$$\gamma_{s/m} = \frac{\omega}{c} d_{x/m} \cos\theta \sin\phi = \frac{2\pi}{\lambda} d_{x/m} \cos\phi_{\text{cone}}. \quad (4)$$

In this case we use the relationships $\omega = 2\pi f$ and $\lambda = c/f$, where λ is wavelength and f is frequency in hertz. Also, $\cos\phi_{\text{cone}}$ is the direction cosine between the x-axis and the unit vector $\mathbf{k}(\phi,\theta)$; ϕ_{cone} is known as the cone angle and defines a conical ambiguity surface for the ULA's angle measurement.

Let d equal the uniform sub-array spacing of an M channel ULA. Also, place the first channel at the origin and designate it the phase reference. The received spatial signal vector is $\mathbf{x_s} = a_s \mathbf{s_s}(f_{sp})$, where a_s is a random, complex voltage and

$$\mathbf{s_s}(f_{sp}) = [1 \quad \exp(j2\pi \cdot f_{sp}) \quad \exp(j2\pi \cdot 2f_{sp}) \quad \cdots \quad \exp(j2\pi \cdot (M-1) f_{sp})]^T \quad (5)$$

is the spatial steering vector. The variable f_{sp} is known as spatial frequency and is given by

$$f_{sp} = \frac{d}{\lambda_o} \cos\phi_{\text{cone}}. \quad (6)$$

We assume a narrow signal bandwidth, thus replacing λ by its center value λ_o, and an error-free array manifold.

A length N periodic pulse train, with pulse repetition interval (PRI) T, comprises the transmit waveform. The radar system uses the pulse train to temporally sample the signal environment. Consider a point scatterer initially at range r_o from the antenna phase center reference point. Assume motion of either the scatterer, radar platform, or both radar and scatterer, leading to a pulse-to-pulse change in range of Δr. The fast-time (range) delay due to motion for receive pulse n is

$$\tau_n = \text{time delay} = \frac{\text{distance}}{c} = \frac{2r_o + n2\Delta r}{c}. \quad (7)$$

The pulse-to-pulse phase is given by

$$\phi_n = \text{time delay} \times \text{radian frequency} = \tau_n \omega = 4\pi \left[\frac{r_o + n\Delta r}{\lambda}\right]. \quad (8)$$

Since radian frequency is the time derivative of phase, the corresponding Doppler frequency is

$$\frac{1}{2\pi}\frac{d\phi_n}{dt} = f_d = \frac{2\Delta r}{\lambda \Delta t} = \frac{2v_r}{\lambda}. \quad (9)$$

Δt represents change in the time variable and v_r denotes the radial velocity component, or line-of-sight velocity.

The resulting temporal signal vector corresponding to a point scatterer with normalized Doppler $\tilde{f}_d = f_d T$ is $\mathbf{x_t} = a_t \mathbf{s_t}(\tilde{f}_d)$, where a_t is a random complex voltage and

$$\mathbf{s_t}(\tilde{f}_d) = [1 \quad \exp(j2\pi \cdot \tilde{f}_d) \quad \exp(j2\pi \cdot 2\tilde{f}_d) \quad \cdots \quad \exp(j2\pi \cdot (N-1)\tilde{f}_d)]^T \quad (10)$$

is known as the temporal steering vector. Comparing (10) and (5), we find a mathematical similarity between spatial and temporal responses. For this reason, the collection of N receive pulses is sometimes called the temporal aperture.

The space-time signal vector corresponding to the return from a point scatterer with spatial frequency f_{sp} and normalized Doppler frequency $\tilde{f}_d$ is $\mathbf{x_{s-t}} = a_{s-t}\mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d)$, where a_{s-t} is a random, complex voltage and $\mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d)$ is the space-time steering vector. The space-time steering vector is written as the Kronecker product of the temporal and spatial steering vectors:

$$\begin{aligned}\mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d) &= \mathbf{s_t}(\tilde{f}_d) \otimes \mathbf{s_s}(f_{sp}) \\ &= [1 \cdot \mathbf{s_s}^T(f_{sp}) \quad e^{j2\pi\cdot\tilde{f}_d} \cdot \mathbf{s_s}^T(f_{sp}) \quad e^{j2\pi\cdot 2\tilde{f}_d} \cdot \mathbf{s_s}^T(f_{sp}) \quad \cdots \quad e^{j2\pi\cdot(N-1)\tilde{f}_d} \cdot \mathbf{s_s}^T(f_{sp})]^T. \end{aligned} \quad (11)$$

Fig. 4 shows the two-dimensional power spectral density (PSD) of two unity amplitude space-time signals, one at an angle of 20 deg and Doppler frequency of 200 Hz, the second at −30 deg and Doppler of −100 Hz. Each signal appears as a two-dimensional sinc function.

MTI signal processing operates on the radar data cube, shown in Fig. 5. While the radar processor does not store data in the form shown in Fig. 5, the data cube is a convenient means of visualizing subsequent space-time processing. Each data cube corresponds to a single coherent processing interval (CPI). Pre-processing steps convert the RF signals at the multiple receiver (Rx) channels to complex baseband space-time and range samples; the A/D clock rate is at least as high as the waveform bandwidth for complex sampling. It is common to refer to the range

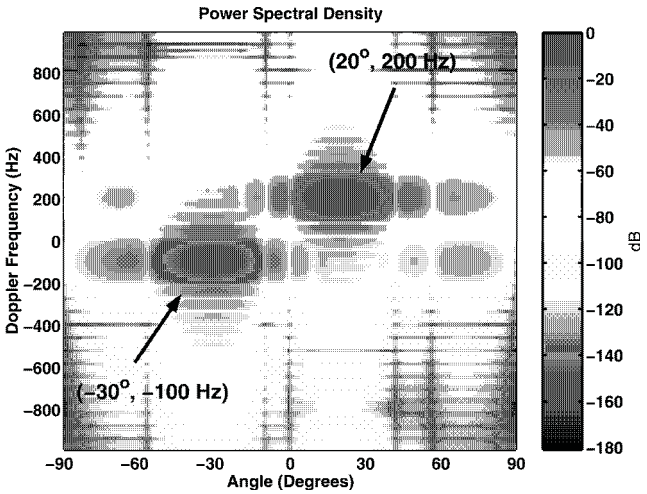


Fig. 4. Angle-Doppler PSD of two space-time signals.

dimension as fast-time and the pulse dimension as slow-time; the processor collects the fast-time samples at the A/D rate. Each row of the data cube corresponds to a spatial sample and each column to a slow-time sample, while the L unambiguous range samples extend in the third dimension. The page of the data cube corresponding to the kth range cell is

$$\mathbf{X_k} = \begin{pmatrix} [\mathbf{X_k}]_{1,1} & [\mathbf{X_k}]_{1,2} & \cdots & [\mathbf{X_k}]_{1,N} \\ [\mathbf{X_k}]_{2,1} & [\mathbf{X_k}]_{2,2} & \cdots & [\mathbf{X_k}]_{2,N} \\ \vdots & \vdots & \cdots & \vdots \\ [\mathbf{X_k}]_{M,1} & [\mathbf{X_k}]_{M,2} & \cdots & [\mathbf{X_k}]_{M,N} \end{pmatrix}. \quad (12)$$

In this format, it is possible for the processor to spatially beamform across the rows and Doppler process across columns. Vectorizing (12) by stacking each succeeding column one beneath the other yields the space-time snapshot for the kth range, i.e. $\mathbf{x_k} = \mathbf{x_{s-t/k}} = \mathbf{X_k}(:)$.

The ground clutter return corresponding to the kth range results from the coherent summation of the many scattering centers within the bounds of each iso-range, including range ambiguities. A simple, yet effective model for the clutter space-time snapshot takes the form

$$\mathbf{c_k} = \sum_{m=1}^{N_a} \sum_{n=1}^{N_c} \mathbf{a_{s-t}}(m,n;k) \odot \mathbf{s_{s-t}}(f_{sp/m,n}, \tilde{f}_{d/m,n};k) \quad (13)$$

where we assume each iso-range consists of N_c statistically independent clutter patches, N_a indicates the number of ambiguous ranges, $f_{sp/m,n}$ and $\tilde{f}_{d/m,n}$ represent the spatial and normalized Doppler frequencies of the $m-n$th patch, and $\mathbf{a_{s-t}}(m,n;k)$ is the length-NM vector containing the space-time voltages for each channel-pulse-range sample and is proportional to the square-root of the clutter patch RCS. Also, $\odot$ represents the Hadamard (element-wise) product. Given the platform velocity vector, $\mathbf{v_p} = v_{p,x}\hat{\mathbf{x}} + v_{p,y}\hat{\mathbf{y}} + v_{p,z}\hat{\mathbf{z}}$, the normalized clutter patch Doppler is

$$\tilde{f}_{d/m,n} = \frac{2v_{r/m,n}T}{\lambda} = \frac{2T}{\lambda}(\mathbf{v_p} \cdot \mathbf{k}(\phi_{m,n}, \theta_{m,n})). \quad (14)$$

Similarly, $v_{r/m,n}$ is the corresponding radial velocity for the $m-n$th clutter patch. From (14) we see that ground clutter Doppler has a distinct dependence on angle. Fig. 6 pictorially characterizes the calculation in (13).

The Multi-Channel Airborne Radar Measurements (MCARM) Program collected data to examine the performance potential of STAP [15]. To justify the basic form of (13), we compare the minimum variance distortionless response (MVDR) spectra [11, 16] for measured and simulated data in Fig. 7. From this figure we find acceptable correspondence between the overall characteristics of the measured and synthetic data; some of the angular spreading of the measured clutter interference is likely due to radome multipath reflections and near-field scattering effects not included in the simulation model.

Narrowband noise jamming signals are the result of the intentional introduction of a noise-like waveform into the receive aperture. A commonly

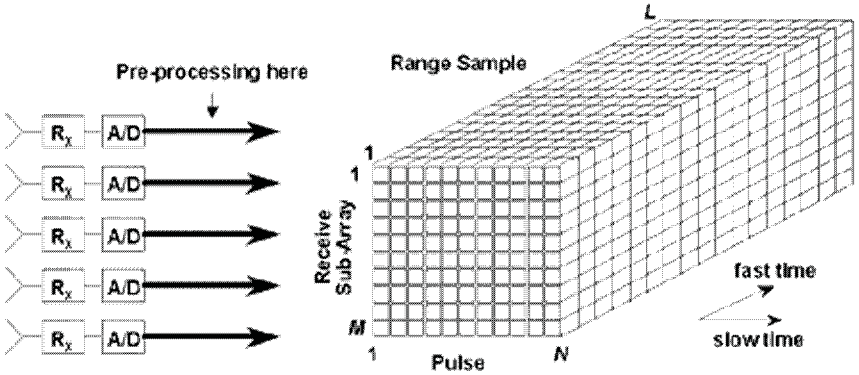


Fig. 5. Radar data cube.

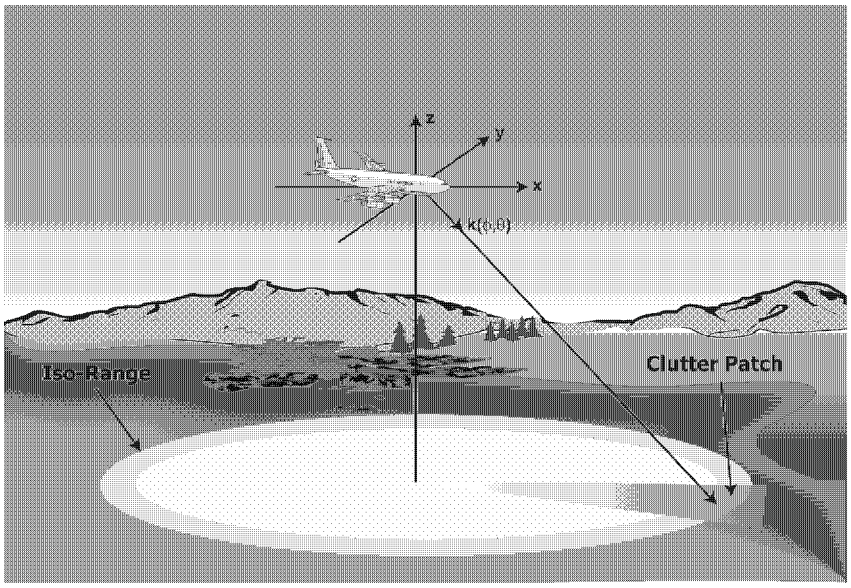


Fig. 6. Geometry for space-time clutter patch calculation.

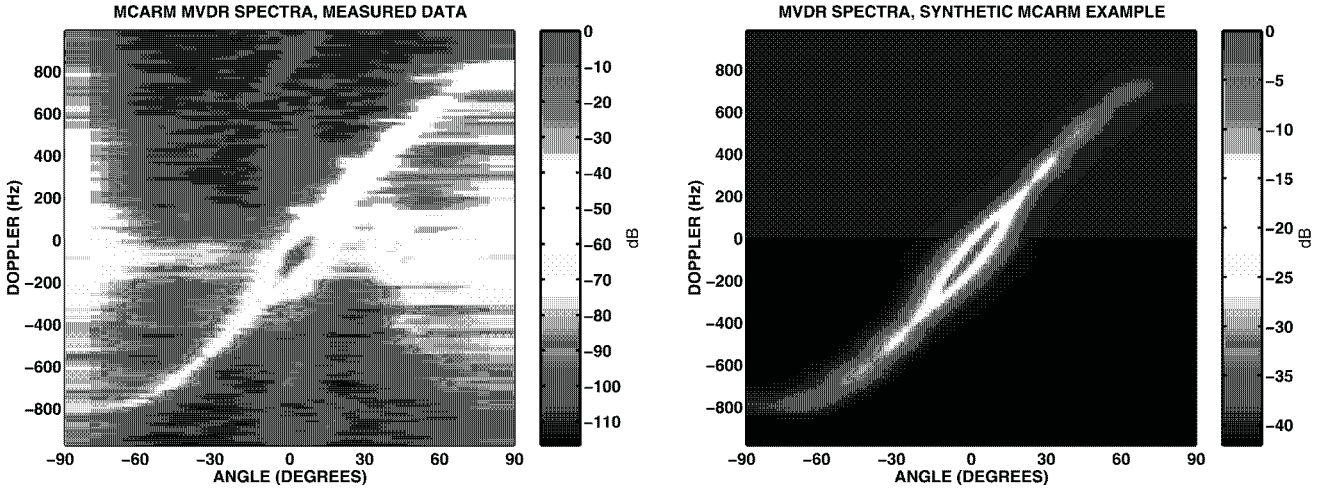


Fig. 7. Comparison of MVDR spectra for measured MCARM data (left) and simulation (right).

employed model for such N_j jamming signals is

$$\mathbf{j_k} = \sum_{m=1}^{N_j} \mathbf{z_m} \otimes \mathbf{s_s}(f_{sp/m}) \tag{15}$$

where $\mathbf{z_m}$ contains voltage samples of the mth jammer waveform taken at the PRI. The different jammer waveforms are uncorrelated with each other, with each waveform being uncorrelated over the PRI:

$$E[[\mathbf{z_m}]_n \cdot [\mathbf{z_m}]_m^*] = \sigma_{J/m}^2 \delta((n-m)T) \tag{16}$$

with "$*$" indicating conjugation. $\sigma_{J/m}^2$ is the single-channel power of the mth jammer. The simple model of (16) neglects jammer correlation in fast-time.

The complete interference space-time snapshot is

$$\mathbf{x_k} = \mathbf{c_k} + \mathbf{j_k} + \mathbf{n_k}. \tag{17}$$

$\mathbf{n_k}$ represents the uncorrelated component due to thermal receiver noise or sky noise; the corresponding single-channel noise power is σ_n^2. Equation (17) presumes the additive nature of clutter, jamming and noise signals. Fig. 8 provides an example of the PSD for ground clutter, jamming and noise signals; the radar array is sidelooking and configured as a ULA. A single jammer signal is present at $-39°$.

III. SPACE-TIME ADAPTIVE FILTER FORMULATION

The space-time processor linearly combines the elements of the data snapshot, yielding the scalar output

$$y_k = \sum_{m=1}^{NM} [\mathbf{w_k}]_m^* [\mathbf{x_k}]_m = \mathbf{w_k}^H \mathbf{x_k} \tag{18}$$

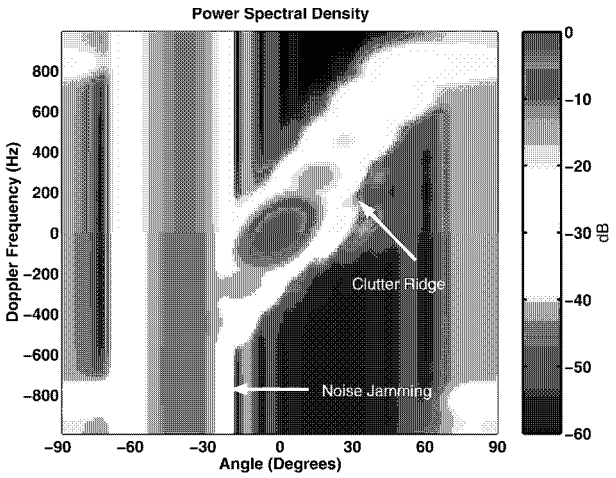


Fig. 8. Simulated PSD for clutter, jamming and noise signals.

where the superscript "H" denotes conjugate transposition and $\mathbf{w_k}$ is the NM-length weight vector. The finite impulse response (FIR) filter of Fig. 9 corresponds to the complex inner product operation of (18); the digital memory, z^{-1}, correspond to time delays at the PRI. A threshold is set to discriminate between one of two hypotheses,

$$\begin{aligned} H_0: &\quad \mathbf{x_k} = \mathbf{c_k} + \mathbf{j_k} + \mathbf{n_k} \\ H_1: &\quad \mathbf{x_k} = \mathbf{s} + \mathbf{c_k} + \mathbf{j_k} + \mathbf{n_k} \end{aligned} \tag{19}$$

while maintaining a specified P_{FA}. $\mathbf{s}$ is the target space-time snapshot. The condition H_0 is the null-hypothesis, or case of target absence, while H_1 is the alternative hypothesis indicating target presence.

Maximizing the output SINR is a key objective of the space-time processor. From (18)–(19), the output SINR is the ratio of signal-to-interference-plus-noise power at the filter output:

$$\text{SINR} = \frac{E[\mathbf{w_k^H s s^H w_k}]}{E[\mathbf{w_k^H x_{k/H_0} x_{k/H_0}^H w_k}]} = \frac{\sigma_s^2 |\mathbf{w_k^H s_{s-t}}(f_{sp}, \tilde{f}_d)|^2}{\mathbf{w_k^H R_k w_k}}. \tag{20}$$

$E[\cdot]$ denotes the expectation operator and $\mathbf{R_k} = E[\mathbf{x_{k/H_0} x_{k/H_0}^H}]$ is the interference covariance matrix. We further assume the target snapshot takes the form $\mathbf{s} = \alpha_{s-t}\mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d)$; α_{s-t} is a complex RMS voltage and $\sigma_s^2 = E[|\alpha_{s-t}|^2]$ is the single-channel, single-pulse signal power. As pointed out in Section I, maximizing SINR equivalently maximizes P_D for a fixed P_{FA} in the multivariate Gaussian case.

The optimal weight vector maximizes the output SINR and takes the form $\mathbf{w_k} = \beta \mathbf{R_k^{-1}} \mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d)$, for arbitrary scalar β [5, 17]. To see this, express (20) as

$$\text{SINR} = \sigma_s^2 \frac{|\breve{\mathbf{w}}_\mathbf{k}^H \breve{\mathbf{s}}|^2}{\breve{\mathbf{w}}_\mathbf{k}^H \breve{\mathbf{w}}_\mathbf{k}} \leq \sigma_s^2 \frac{\breve{\mathbf{w}}_\mathbf{k}^H \breve{\mathbf{w}}_\mathbf{k} \breve{\mathbf{s}}^H \breve{\mathbf{s}}}{\breve{\mathbf{w}}_\mathbf{k}^H \breve{\mathbf{w}}_\mathbf{k}} \tag{21}$$

where $\mathbf{w_k} = \mathbf{R_k^{-1/2}} \breve{\mathbf{w}}_\mathbf{k}$ and $\mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d) = \mathbf{R_k^{1/2}} \breve{\mathbf{s}}$. For the covariance matrices of interest, $\mathbf{R_k} = \mathbf{R_k^{1/2}} \mathbf{R_k^{1/2}}$. By choosing $\breve{\mathbf{w}}_\mathbf{k} = \breve{\mathbf{s}}$, (21) achieves the upper bound. Substituting the prior expressions gives

$$\begin{aligned} \breve{\mathbf{w}}_\mathbf{k} = \mathbf{R_k^{1/2}} \mathbf{w_k} = \breve{\mathbf{s}} = \mathbf{R_k^{-1/2}} \mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d) \\ \Rightarrow \mathbf{w_k} = \mathbf{R_k^{-1}} \mathbf{s_{s-t}}(f_{sp}, \tilde{f}_d). \end{aligned} \tag{22}$$

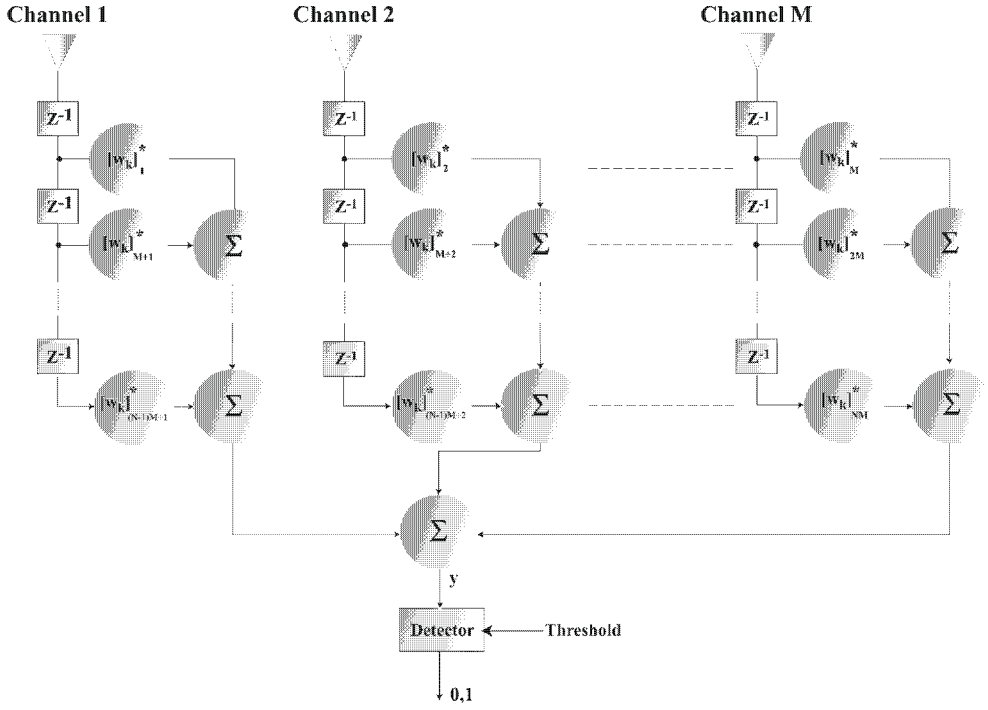


Fig. 9. Space-time filtering operation and thresholding.

Scaling the weight vector by β does not alter the output SINR.

STAP is a data domain implementation of the optimal filter with weight vector given by (22). In practice, both $\mathbf{R_k}$ and $\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$ are unknown. The processor substitutes an estimate for each quantity to arrive at the adaptive weight vector

$$\hat{\mathbf{w}}_\mathbf{k} = \hat{\beta}\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{v_{s-t}} \tag{23}$$

where $\hat{\beta}$ is a scalar, $\mathbf{v_{s-t}}$ is a surrogate for $\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$ and $\hat{\mathbf{R}}_\mathbf{k}$ is an estimate of $\mathbf{R_k}$. This approach is known as sample matrix inversion (SMI). Alternate adaptive weight calculation methods are given in [5, 12, 13, 16], for example.

It is most common to compute the covariance matrix estimate as [17]

$$\hat{\mathbf{R}}_\mathbf{k} = \frac{1}{P}\sum_{m=1}^{P}\mathbf{x_m}\mathbf{x_m}^H. \tag{24}$$

$\{\mathbf{x_m}\}_{m=1}^{P}$ are known as secondary or training data. If all training data are independent and identically distributed (iid) with respect to the null-hypothesis condition of the test cell, choosing $P \approx 2NM$ yields an average performance loss of roughly 3 dB [17]. To avoid target self-whitening, the processor excludes the cell under test, as well as several adjacent cells (known as "guard cells"), from the training data set. The performance loss results from the difference between actual and estimated covariance matrices.

The processor tests for targets at a series of discrete points over the spatial and Doppler frequencies of interest. Given knowledge of the sub-array locations and the PRI, the processor generates the hypothesized space-time steering vector $\mathbf{v_{s-t}}$ via a calculation identical in form to (11). Error between $\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$ and $\mathbf{v_{s-t}}$, known as steering vector mismatch, also leads to some performance loss. Leading causes of steering vector mismatch include array errors and straddling (straddle loss occurs since the processor commonly tests for $\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$ by choosing various guesses $\mathbf{v_{s-t}}$ equally spaced over a range of spatial and Doppler frequencies). Further discussion on the impact of steering vector mismatch is given in [18].

Certain choices of β or $\hat{\beta}$ can prove advantageous. For example, forming the square-law output and setting

$$\hat{\beta} = \frac{1}{\sqrt{\mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{v_{s-t}}}} \tag{25}$$

yields the test statistic

$$\eta = |\hat{\mathbf{w}}_\mathbf{k}^H\mathbf{x_k}|^2 = \frac{|\mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{x_k}|^2}{\mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{v_{s-t}}}. \tag{26}$$

Equation (26), known as the adaptive matched filter (AMF) test statistic, exhibits constant false alarm rate (CFAR) properties [19, 20]. Comparing η to a fixed threshold v_T', and recognizing $\mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{v_{s-t}}$ as an estimate of the output noise power, provides further insight:

$$\eta \underset{H_0}{\overset{H_1}{\gtrless}} v_T' \rightarrow |\mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{x_k}|^2 \underset{H_0}{\overset{H_1}{\gtrless}} v_T' \cdot \mathbf{v}_\mathbf{s-t}^H\hat{\mathbf{R}}_\mathbf{k}^{-1}\mathbf{v_{s-t}}. \tag{27}$$

Hence, we can view (27) as a comparison of the filter output power to a fixed threshold times a multiplier corresponding to an estimate of the noise power. Alternately, in light of (26), the normalization serves to modulate the filter output by the inverse of the noise power estimate so that a fixed threshold provides CFAR performance.

In addition to the AMF, other extensions of the SMI beamformer yielding CFAR behavior include the generalized likelihood ratio test (GLRT) of [21] and the adaptive coherence estimator (ACE) given in [22].

IV. PERFORMANCE METRICS

An optimum detection statistic for (18) under the Gaussian assumption $\mathbf{x}_{\mathbf{k}/\mathbf{H}_0} \sim CN(0,\mathbf{R_k})$, follows from the likelihood ratio test and appears as [1, 23, 24]

$$|y_k| \underset{H_0}{\overset{H_1}{\gtrless}} v_T. \tag{28}$$

where v_T is the detection threshold.

The performance of (28) is given by

$$\begin{aligned} P_{FA} &= \exp\left(\frac{-\beta_T^2}{2}\right) \\ P_D &= \int_{\beta_T}^{\infty} u\exp\left(\frac{-(u^2+\alpha^2)}{2}\right)I_0(\alpha u)du \end{aligned} \tag{29}$$

where P_{FA} is the probability of false alarm, P_D is the probability of detection, $\beta_T = v_T/\sqrt{\mathbf{w_k^H R_k w_k}}$ is a normalized detection threshold, $I_0(\cdot)$ is the modified zero-order Bessel function of the first kind, and α equals the square-root of the peak output SINR. In light of (20), and accounting for average signal power, we find

$$\alpha^2 = 2\times \text{SINR} = 2\times\frac{\sigma_s^2|\mathbf{w_k^H s}_{s-t}(f_{sp},\tilde{f}_d)|^2}{\mathbf{w_k^H R_k w_k}}. \tag{30}$$

Equation (29) is a monotonic function of α, and hence α^2. Thus, maximizing SINR likewise maximizes P_D for a fixed value of P_{FA}, a point clarified by Fig. 1 and thoroughly discussed in [5].

Due to the critical importance of SINR, STAP researchers commonly employ SINR loss factors to assess detection performance potential [6, 7]. Two commonly used SINR loss factors are $L_{s,1}(f_{sp},\tilde{f}_d)$ and $L_{s,2}(f_{sp},\tilde{f}_d)$, where each loss term is bound between zero and unity. $L_{s,1}(f_{sp},\tilde{f}_d)$ compares

interference-limited performance to noise-limited capability, assuming all quantities are known:

$$L_{s,1}(f_{sp},\tilde{f}_d) = \frac{\text{SINR}|_{\mathbf{w_k}}}{\text{SNR}} = \left(\frac{\mathbf{w_k}^H\mathbf{R_s}\mathbf{w_k}}{\mathbf{w_k}^H\mathbf{R_k}\mathbf{w_k}}\right)\Big/\left(\frac{\sigma_s^2}{\sigma_n^2}NM\right) \tag{31}$$

where $\mathbf{R_s} = E[\mathbf{ss}^H]$ is the signal correlation matrix. Substituting the optimal weight vector, $\mathbf{w_{k/opt}} = \beta\mathbf{R_k}^{-1}\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$, specifies the upper bound on performance in the maximum SINR sense. Since the optimal weight vector calculation requires the known covariance matrix, $L_{s,1}(f_{sp},\tilde{f}_d)$ is sometimes called the clairvoyant SINR loss.

$L_{s,2}(f_{sp},\tilde{f}_d)$ determines the loss between an implementation requiring estimated statistics and the clairvoyant case (e.g., adaptive versus optimum):

$$\begin{aligned} L_{s,2}(f_{sp},\tilde{f}_d) &= \frac{\text{SINR}|_{\mathbf{w_k}=\hat{\mathbf{w}}_\mathbf{k}}}{\text{SINR}|_{\mathbf{w_k}=\mathbf{w_{k/opt}}}} \\ &= \left(\frac{\hat{\mathbf{w}}_\mathbf{k}^H\mathbf{R_s}\hat{\mathbf{w}}_\mathbf{k}}{\hat{\mathbf{w}}_\mathbf{k}^H\mathbf{R_k}\hat{\mathbf{w}}_\mathbf{k}}\right)\Big/\left(\frac{\mathbf{w}_\mathbf{k/opt}^H\mathbf{R_s}\mathbf{w_{k/opt}}}{\mathbf{w}_\mathbf{k/opt}^H\mathbf{R_k}\mathbf{w_{k/opt}}}\right). \end{aligned} \tag{32}$$

If the training data are iid and there is no steering vector mismatch (i.e., $\mathbf{v_{s-t}} \triangleq \mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)$), Reed, Mallett and Brennan showed $L_{s,2}(f_{sp},\tilde{f}_d)$ is beta-distributed with mean [17]

$$E[L_{s,2}(f_{sp},\tilde{f}_d)] = \frac{(P+2-NM)}{(P+1)}. \tag{33}$$

P is the number of training data vectors. Equation (33) suggests a nominal training requirement of $P \approx 2NM$ training data vectors to achieve an average loss of 3 dB between adaptive and optimal filters; this result is known as the RMB Rule after its originators. Interestingly, convergence depends only on the number of samples, not the particular characteristics of the interference environment. Additional discussion on SMI detection loss is given in [25].

Given the loss factor terms, SINR can be written

$$\text{SINR}(f_{sp},\tilde{f}_d) = \text{SNR}(f_{sp}) \times L_{s,1}(f_{sp},\tilde{f}_d) \times L_{s,2}(f_{sp},\tilde{f}_d) \tag{34}$$

where $\text{SNR}(f_{sp})$ is the angle-dependent signal-to-noise ratio. Those target velocities closest to the dominant clutter component, and exhibiting SINR loss above some acceptable value, viz. $L_{s,1}(f_{sp},\tilde{f}_d) \cdot L_{s,2}(f_{sp},\tilde{f}_d) \geq \varepsilon$, determine the minimum detectable velocity (MDV). For example, suppose we calculate SNR to be 13 dB, thereby yielding $P_D = 0.87$ for $P_{FA} = 1E-6$ according to (29) and Fig. 1 for a non-fluctuating target. If our minimum detection requirement is $P_D = 0.5$ for this same false alarm rate, then SINR must be greater than or equal to 11.25 dB. This indicates a tolerable combined SINR loss of 1.75 dB, or $\varepsilon = 0.668$.

Improvement factor (IF) is another common metric, given as the ratio of output SINR to the input SINR measured at a given space-time element:

$$\text{IF} = \frac{\text{SINR}_{\text{out}}}{\text{SINR}_{\text{element}}} = \frac{|\mathbf{w_k}^H\mathbf{s_{s-t}}(f_{sp},\tilde{f}_d)|^2(\sigma_c^2+\sigma_n^2)}{\mathbf{w_k}^H\mathbf{R_k}\mathbf{w_k}} \tag{35}$$

where σ_c^2 is the total clutter power received by a single sub-array on a single pulse [6]. In the noise-limited case, (35) defaults to the space-time integration gain (nominally, NM). IF closely relates to the preceding SINR loss definitions.

V. REDUCED-DIMENSION/REDUCED-RANK STAP FORMULATIONS

In the previous sections we described STAP as a two-dimensional, adaptive, linear filter operating on M spatial channels and N pulses. This direct formulation is known as the joint-domain STAP. Critical joint-domain STAP limitations include minimal training sample support and substantial computational burden. For example, given $N = 128$ and $M = 22$ as in the MCARM data collection [15], the nominal training support of $2NM = 5632$ far exceeds the roughly 630 range bins comprising the unambiguous range interval. Additionally, computational burden associated with the SMI approach is $O(N^3M^3)$. Certainly reducing either N or M is possible, but either option adversely affects performance by degrading the available space-time aperture. The purpose of this section is to bring to the reader's attention several alternate STAP formulations, based on reducing the processor's dimensionality and/or applying a low-rank interference covariance matrix approximation, to circumvent joint-domain STAP limitations.

Observe that a two-dimensional (angle-Doppler) frequency domain implementation of STAP is possible; barring apodization loss, the frequency domain approach yields detection performance identical to its space-time counterpart. Adaptively combining a sub-set of the frequency domain observations (e.g., select angle and/or Doppler filter outputs) is possible, and intuitively should yield acceptable performance since the frequency domain transform tends to compress the ground clutter response into distinct angle-Doppler bins (see Fig. 8).

It is common to refer to STAP as any member of the class of linear, adaptive filtering algorithms operating on space-time observations to enhance certain characteristics, viz. target detection. Such alternate STAP algorithms generally fall into one of two groups based on the particular transformations applied to the data. Reduced-dimension (RD) STAP

methods apply data independent transformations to pre-filter the data and reduce the number of adaptive DoFs [6–8, 26–28]; reduced computational burden and improved statistical convergence (i.e., reduced training sample support) are the primary benefits. On the other hand, reduced-rank (RR) STAP methods employ data dependent transformations to construct the space-time adaptive filter [29–35]; improved statistical convergence is the objective of RR-STAP. While RD and RR-STAP generally provide good performance, some concerns include reduced interference cancellation capability, degraded MDV and potential impact on ancillary functions, such as bearing estimation, as well as computational burden in the RR-STAP case. Performance metrics given in Section IV apply to the RD or RR-STAP architectures.

In RD-STAP, a linear transformation projects the space-time data vector $\mathbf{x_k}$ into a lower dimensional subspace. The transformed data vector is

$$\tilde{\mathbf{x}}_{\mathbf{k}} = \mathbf{T}^H \mathbf{x}_{\mathbf{k}}, \qquad \mathbf{T} \in C^{NMxJ} \tag{36}$$

where $J \ll NM$ and $\tilde{\mathbf{x}}_{\mathbf{k}}$ has dimension $J \times 1$. Computational burden associated with matrix inversion drops from $O(N^3M^3)$ to $O(J^3)$, and nominal sample support decreases from $2NM$ to $2J$ in accord with the RMB rule.

The $J \times J$ null-hypothesis covariance matrix corresponding to (36) is

$$\tilde{\mathbf{R}}_{\mathbf{k}} = E[\tilde{\mathbf{x}}_{\mathbf{k/H_0}} \tilde{\mathbf{x}}^H_{\mathbf{k/H_0}}] = \mathbf{T}^H \mathbf{R}_{\mathbf{k}} \mathbf{T}. \tag{37}$$

Applying the same transformation to the space-time steering vector gives $\tilde{\mathbf{s}} = \mathbf{T}^H \mathbf{s}_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d)$. The corresponding optimal weight vector is $\tilde{\mathbf{w}}_{\mathbf{k}} = \tilde{\beta} \tilde{\mathbf{R}}_{\mathbf{k}}^{-1} \tilde{\mathbf{s}}$, for arbitrary scalar $\tilde{\beta}$. The adaptive solution involves calculating $\hat{\tilde{\mathbf{R}}}_{\mathbf{k}}$ from (24) using the transformed training data set $\{\mathbf{T}^H \mathbf{x}_{\mathbf{m}}\}_{m=1}^{P}$, replacing $\tilde{\mathbf{s}}$ with the hypothesized steering vector $\tilde{\mathbf{v}}$, and then forming and applying the adaptive weight vector $\hat{\tilde{\mathbf{w}}}_{\mathbf{k}} = \hat{\tilde{\mathbf{R}}}_{\mathbf{k}}^{-1} \tilde{\mathbf{v}}$.

A variety of choices for $\mathbf{T}$ are possible. Naturally, the best choices provide an effective combination of DoFs to mitigate interference while minimizing computational burden and the requisite number of training samples for covariance estimation. Common selections for $\mathbf{T}$ include: post-Doppler transformation with selection of multiple adjacent bins [26]; post-Doppler, beamspace transformation, with selection of adjacent Doppler filters and spatial beams [27]; and, pre-Doppler, adaptive filtering of multiple, adjacent pulses, followed by traditional Doppler processing to achieve integration gain [28]. Fig. 10 depicts the processing flow for the former two methods. The common radar processing building blocks—Doppler processing and beamforming—serve as the foundation for these approaches. Three to five adjacent bins are typical for the multi-bin, post-Doppler approach with full spatial DoFs available, as shown in the top of Fig. 10. In contrast, the post-Doppler, beamspace technique appearing on the bottom of Fig. 10 provides a different complement of spatial (angle) and temporal (Doppler) DoFs; three to five adjacent angle-Doppler bins (nine to twenty-five total DoFs) is typical.

Reduced-rank STAP takes advantage of the low rank nature of clutter and jamming observations [6, 7, 29–35]. Transformations applied to the data are necessarily data dependent. To illustrate the basic concept, we consider two cases taken from [31, 32].

The eigen-decomposition of the space-time covariance matrix is

$$\mathbf{R}_{\mathbf{k}} = \sum_{m=1}^{NM} \lambda_{k/m} \mathbf{q}_{\mathbf{k/m}} \mathbf{q}^H_{\mathbf{k/m}} \tag{38}$$

where $\mathbf{q}_{\mathbf{k/m}}$ is an eigenvector corresponding to eigenvalue $\lambda_{k/m}$ [16]. The optimal weight vector in (22) can be written

$$\mathbf{w}_{\mathbf{k}} = \frac{1}{\lambda_0} \left[\mathbf{s}_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d) - \sum_{m=1}^{NM} \frac{\lambda_{k/m} - \lambda_0}{\lambda_{k/m}} \alpha_{k/m} \mathbf{q}_{\mathbf{k/m}} \right] \tag{39}$$

where $\lambda_0 = \min(\lambda_{k/m})$ and $\alpha_{k/m} = \text{proj}(\mathbf{q}_{\mathbf{k/m}}, \mathbf{s}_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d))$. This result was first given in [32] and suggests one can view STAP as an adaptive pattern synthesis: the filter places notches in the quiescent response, given by $\mathbf{s}_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d)$, at locations given by $\mathbf{q}_{\mathbf{k/m}}$ and where $\lambda_{k/m}$ controls the null-depth. The adaptive filter output is

$$y_k = \frac{\mathbf{s}^H_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d)}{\lambda_0} \left[\mathbf{I}_{\mathbf{NM}} - \sum_{m=1}^{NM} \frac{\lambda_{k/m} - \lambda_0}{\lambda_{k/m}} \mathbf{q}_{\mathbf{k/m}} \mathbf{q}^H_{\mathbf{k/m}} \right] \mathbf{x}_{\mathbf{k}} \tag{40}$$

thereby suggesting the subtraction of weighted eigen-components from the space-time data vector $\mathbf{x_k}$ followed by matched filtering. (Noting that $a_{k/m} = \mathbf{q}^H_{\mathbf{k/m}} \mathbf{x}_{\mathbf{k}}$ is a Karhunen-Loeve coefficient [16] clarifies this point.) When inserting the covariance estimate into (39), the source of potentially poor adaptive filter response is evident: the perturbed noise floor estimate leads to the subtraction of random, noise-like eigenvectors from the quiescent response, resulting in elevated filter sidelobes. The reduced-rank version improves on this situation by only incorporating those eigenvectors corresponding to the $J \ll NM$ largest eigenvalues. Hence, the resulting filter is called a principal components, reduced-rank filter.

The principal components inverse (PCI) solution applies when $\lambda_{k/m} \gg \lambda_0$ for all signal subspaces [29, 31, 33–35]. Assuming J colored-noise components,

$$y_k = \mathbf{s}^H_{\mathbf{s-t}}(f_{sp}, \tilde{f}_d) \left[\mathbf{I}_{\mathbf{NM}} - \sum_{m=1}^{J} \mathbf{q}_{\mathbf{m}} \mathbf{q}^H_{\mathbf{m}} \right] \mathbf{x}_{\mathbf{k}}. \tag{41}$$

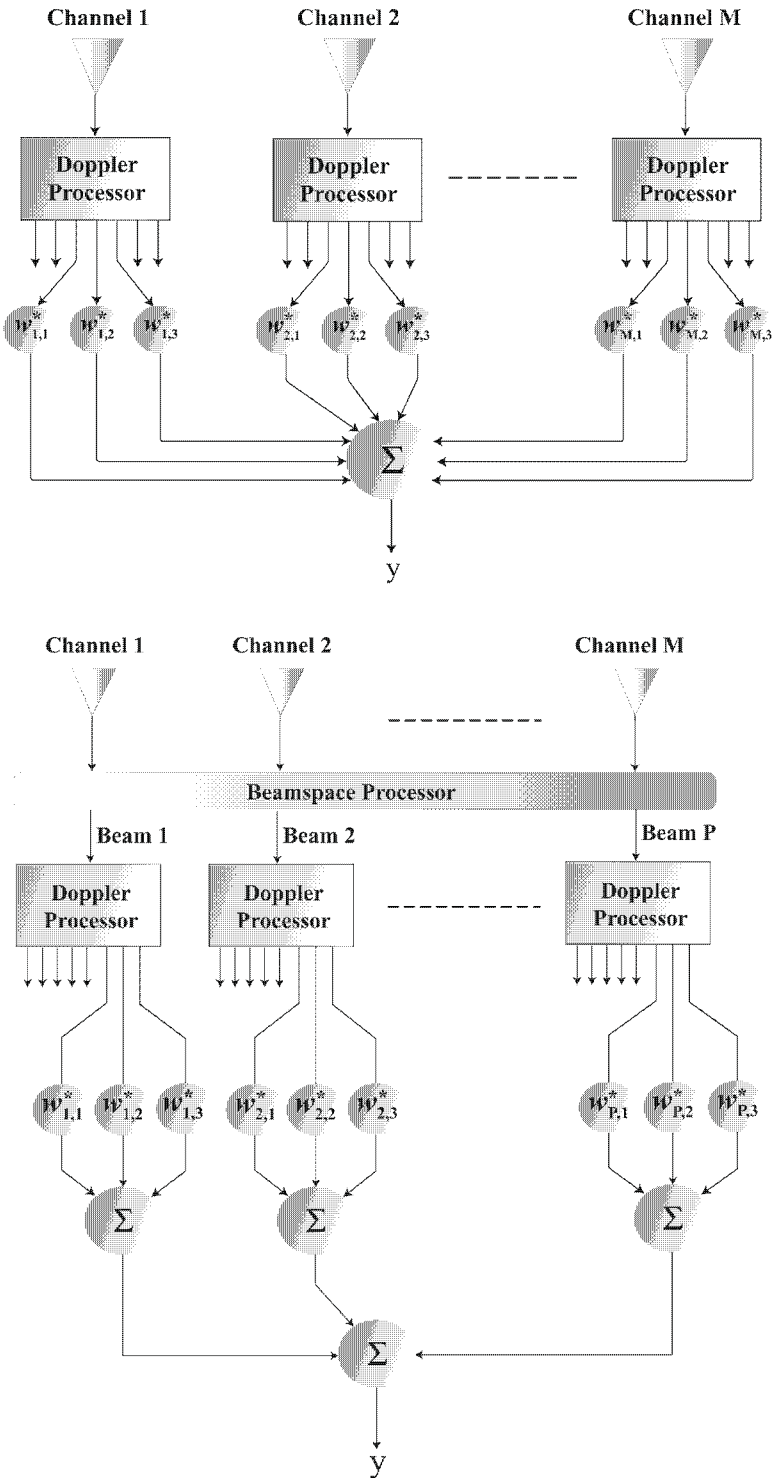


Fig. 10. Comparison of element-space, post-Doppler (top), and beamspace, post-Doppler (bottom) reduced-dimension STAP.

Equation (41) follows from (40) by setting the eigenvalue weighting in the summation to unity and removing the λ_0 normalization of the matched filter. The filter given by (41) implements orthogonal projection to cancel interference and then applies a matched filter. The size of the training set reduces to $2J$, or twice the interference rank, for performance roughly equivalent to the full DoF STAP case employing $2NM$ training data. Further development of the PCI method, including discussion of statistical convergence improvements in contrast to the RMB rule, is given in [33–35]. Because of its enhanced convergence, RR-STAP can provide better performance than the SMI formulation operating with limited training data; the corresponding cost of RR-STAP includes the use of the computationally

demanding singular value decomposition (SVD) and a mechanism for rank determination.

The recently introduced parametric adaptive matched filter (PAMF) [36] employs a low order, multi-channel autoregressive model to characterize the interference environment; this method enhances covariance matrix estimation when employing low sample support, exhibiting rapid convergence to the performance bound set by the optimum filter. By utilizing all NM space-time DoFs, the PAMF suffers no performance loss due to reduced dimensionality. Since the method does not explicitly employ rank-reduction, the PAMF does not neatly fit into either reduced-dimension or reduced-rank categories previously discussed. Extensions of the PAMF to non-Gaussian clutter mitigation have also been considered and are extensively documented in [37–39].

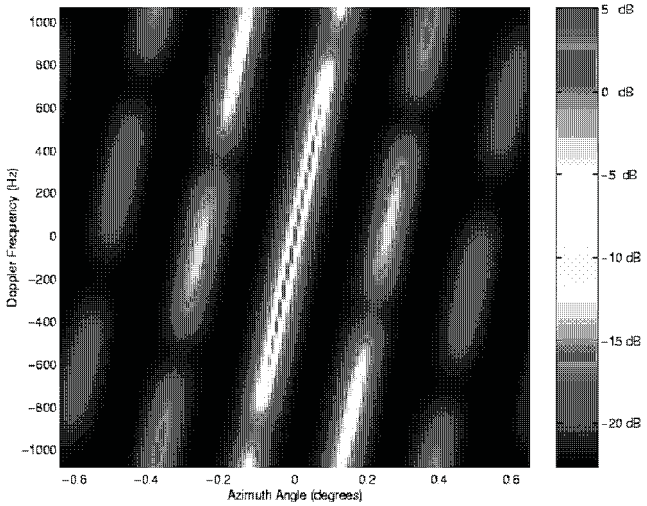


Fig. 11. MVDR spectra using SBR parameters defined in [41].

VI. OVERVIEW OF SOME CURRENT TOPICS

Previous sections of this paper discuss fundamental aspects of STAP. In this next section of the paper we briefly highlight current STAP trends. While our treatment of each topic is cursory, making the reader aware of recent STAP initiatives and discussing those areas where STAP is impacting radar system development serves as our primary motivation. This list of topics is not exhaustive, but does cover a fairly broad range of STAP activity.

A. STAP Application to Space-Based Radar

Space-based radar (SBR) provides the potential for near-continuous surveillance coverage of the Earth's surface. Since the SBR is down-looking, clutter and jammer mitigation techniques are integral parts of the MTI mode design. An overview of SBR is given in [40].

Major distinctions between spaceborne and airborne platforms include the very high satellite platform velocity (at lower orbits), much steeper operational grazing angles, profound influence of the antenna pattern, the potential for dramatic variation in clutter statistics, size of the antenna footprint on the ground, and the deterministic nature of the satellite orbit. Additionally, the launch vehicle limits the size, power and weight of the SBR system. In low earth orbit, the satellite travels at approximately 7 km/s; this contrasts with the 120–220 m/s velocity typical of airborne surveillance radar. At higher grazing angles, the clutter is more specular, thereby increasing the amount of clutter power competing with the target signal. The field of regard for SBR is very large. Hence, the system can survey large areas, but clutter cultural features can change dramatically. Also, in SBR MTI, the pulse repetition frequency (PRF) is set to avoid range ambiguities in the radar footprint. The limited PRF leads to substantial Doppler aliasing. Perhaps, most importantly, the azimuth dimension of the aperture strongly influences mainbeam clutter spread. STAP plays an important role in overcoming some of the diffraction-limited characteristics of deployable space-based arrays whose size is limited by launch vehicle constraints. Also, by maximizing SINR, STAP provides the best detection performance potential for a fixed dwell; hence, STAP reduces the dwell time necessary to achieve a specified P_D, hence supporting high area coverage rates.

Using the clutter model of (13), with additional modeling to incorporate orbital mechanics, we arrive at the clutter-plus-noise MVDR spectra of Fig. 11; parameters for the simulation come from [41]. Sixty-four pulses comprise the coherent dwell and the array is linear with twelve spatial channels. Array dimensions are 16 m in azimuth by 2.5 m in elevation. A key observation concerning Fig. 11 is the high degree of aliasing, mainly in Doppler, but also in angle. (Angle ambiguity occurs because the separation between the twelve spatial channels far exceeds one-half of a wavelength; the pattern shown in Fig. 11 repeats itself in angle.) Fig. 11 makes it clear that SBR MTI is an endo-clutter detection problem: the target signal directly competes with mainbeam clutter over virtually the entire unambiguous Doppler space.

Fig. 12 shows the SINR loss curve using the optimal space-time processor; the results provide an acceptable match to those in [41]. Observe the poor performance of the 8 m azimuth by 5 m elevation array (40 m^2 total). The shorter azimuth dimension leads to increased beamwidth. The mainbeam clutter spread across this increased azimuth beamwidth is very large; in combination with signal aliasing, clutter affects all Doppler frequencies. The 16 m azimuth by 2.5 m elevation antenna maintains the imposed 40 m^2 aperture stow size, yet provides significantly

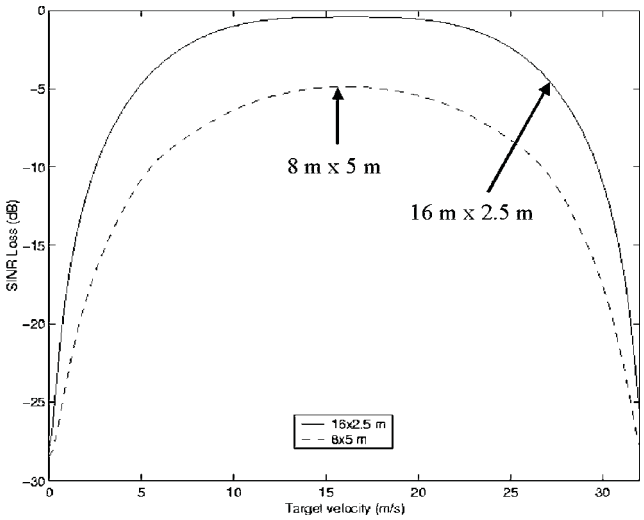


Fig. 12. SBR SINR loss for varying antenna size and optimal space-time processing.

better performance. Since the azimuth beamwidth is much narrower for the 16 m dimension, mainbeam clutter spread is minimized. The reduced mainbeam clutter spread translates into much better SINR loss performance. Hence, STAP for SBR is a full system design task, coupling algorithm selection with the appropriate choice of radar system parameters. While not shown herein, the performance of the non-adaptive processor is unacceptable.

A notional signal processing architecture for a spaceborne ground moving target indication (GMTI) radar is given in [41]. A multi-channel array is broken into sub-arrays; the sub-arrays are then used for adaptive jammer and clutter cancellation. Sub-band filtering—sub-dividing the received signal into smaller width frequency bins—is used to compensate for signal bandwidth. Without sub-banding, dispersion across the array degrades cancellation performance. Essentially, the processor applies narrowband STAP within each sub-band. Jammer cancellation occurs in a separate step from clutter mitigation to reduce computational burden and training data requirements. The jammer cancellation step requires a special training interval and operates in beamspace. Jammer canceled beams are then pulse compressed and fed into a beamspace STAP used to mitigate clutter. After clutter cancellation, the processor re-stitches the waveform in the sub-band combiner to achieve the original range resolution. Scalar data then proceed to a detector, such as a cell averaging constant false alarm rate (CA-CFAR) circuit. The post-processor accomplishes target tracking. Adaptive array processing plays a key role in this architecture: a one-dimensional adaptive canceler suppresses the jammer, while the two-dimensional STAP minimizes the impact of clutter on detection performance.

References [42, 43] describe the application of STAP to sparse, distributed aperture SBR. Non-uniform spatial sampling—to disturb grating lobes—in combination with a non-sidelooking antenna configuration, leads to non-stationary clutter conditions; [44] describes this SBR challenge and considers ameliorating solutions.

B. Bistatic STAP

Bistatic radar systems offer several advantages over their monostatic counterparts, including reduced space loss, silent operation, reduced susceptibility to jamming, and synergistic coherent operation with existing systems. Among its drawbacks, bistatic aerospace radar systems must effectively cope with severe, spectrally diverse ground clutter returns. For this reason, effective bistatic clutter cancellation techniques are crucial to look-down bistatic system deployment.

The class of adaptive clutter filtering techniques—viz., STAP and its variants—developed for monostatic airborne radar offer a logical starting framework in bistatics. However, the non-stationary nature of bistatic ground clutter, resulting from the complex influence of sensor geometry and motion, directly violates intrinsic adaptive algorithm assumptions. Non-stationarity leads to covariance estimation errors and hence degrades adaptive filter performance. For this reason, STAP techniques developed for monostatic radar require modification in the bistatic STAP case [45–52].

Table I summarizes some recently developed bistatic STAP techniques given in [46–52]. The performance of the different methods can vary greatly. A progression from the most simplistic approach—localized training—to more elaborate methods is evident in the table.

C. Knowledge-Aided STAP

Typical STAP operating environments are heterogeneous due to a variety of factors. As described previously, STAP relies on a covariance matrix estimate as part of its implementation; the processor selects training data over range to arrive at this estimate via the calculation in (24). When the statistics of the training data vary with respect to the properties of the test cell, the asymptotic covariance matrix estimate is erroneous. The impact of this erroneous estimate has been considered, for example, in [53–56]. Table II provides a taxonomy of clutter heterogeneity.

The judicious application of a priori knowledge has shown potential to enhance performance in complex, heterogeneous clutter environments [57–59]. For example, target signals in the training data substantially degrade detection performance [56]. By using database information, such as the knowledge sources listed in [59], to identify

TABLE I
Summary of Bistatic STAP Techniques

Method	Rationale	References
Localized processing	Local training selection minimizes non-stationarity.	[46, 47, 49, 51]
Time-varying weights	Approximate time-varying nature of unknown, optimal weight vector by allowing linear variation over range.	[46, 48]
Localized time-varying weights	Impose linear variation in weight vector by restricting training range interval.	[49]
Doppler Warping	Apply deterministic, complex taper to align clutter Doppler for specified cone angle, thereby enhancing STAP training set.	[48]
Angle-Doppler compensation	Deterministically align maximum angle-Doppler response ("spectral centers") over range to enhance STAP training set.	[50, 51]
Higher-order Doppler warping	Deterministically align clutter Doppler over various cone angles.	[52]

TABLE II
Taxonomy of Clutter Heterogeneity and Related Effects

Heterogeneity Type	Causes	Impact on Adaptive Radar
Amplitude	Shadowing and obscuration, range-angle dependent change in clutter reflectivity, strong stationary discretes, sea spikes, urban centers, land-sea interfaces, etc.	Null depth depends on eigenvalue ratio—MLE "averaging" leads to underestimated eigenvalue magnitude, and consequently, uncancelled clutter and increased false alarm rate.
Spectral	Intrinsic clutter motion due to soft scatterers (trees, windblown fields, etc.), ocean waves, weather effects.	Null width set to mean spread—too narrow for some range cells, too wide for others—thereby leading to either increased clutter residue or signal cancellation. Degrades MDV.
CNR-dependent spectral mismatch	Modulation of principal components and other low power signal terms rise above the noise floor with increases in CNR.	Same impact as spectral mismatch.
Moving Scatterers	Ground traffic, weather, insects and birds, air vehicles.	Mainlobe nulling, false sidelobe target declarations, and distorted beam patterns exhaust DoF.
Some Other Effects	Chaff, multi-bounce/ multi-path, impact of platform geometry (e.g., non-sidelooking or bistatic) on angle-Doppler behavior over range.	Combination of above effects.

and screen roadways from the training set, or by employing data-derived knowledge, tremendous performance gains (approximately 15 dB for the example in [56]) are possible. In [60], Farina et al. describe a nonlinear STAP processing scheme relying on a priori knowledge and data-derived characteristics to enhance detection performance in heterogeneous clutter environments. Both [56] and [60] use actual measured data as part of the analysis.

The development of knowledge-aided STAP is a major objective of the Defense Advanced Research Projects Agency's (DARPA's) Knowledge-Aided Sensor Signal Processing and Expert Reasoning (KASSPER) Program, commenced in 2002 [61].

D. Multi-Channel Synthetic Aperture Radar (MSAR)

SAR processing is tantamount to matched filtering; in general, each stationary scatterer has a unique phase history exploitable by the processor to discriminate and estimate electromagnetic reflectivity [1, 62]. The phase history strictly depends on the time-varying range between a single antenna phase center (APC) and the scatterer as the platform flies a particular path. The long-dwell nature of SAR translates to resolution enhancements in the cross-range (slow-time) dimension, nominally as a result of an azimuth chirp signal at a Doppler offset corresponding to the clutter patch's cross range position.

Inherently, SAR makes no provision for moving target detection. Depending on the particular target motion, the corresponding image of the target typically appears displaced in cross-range or blurred. Despite these challenges, SAR plays a key role when high cross-range resolution is necessary to identify the particular target class. Additionally, to enhance discrimination, the cancellation of stationary ground clutter is desirable. Incorporating multiple SAR receive channels is the best option for target imaging

with simultaneous cancellation of stationary clutter returns. Original multi-channel schemes relied on the displaced phase center antenna (DPCA) principle [1, 6, 7]. DPCA attempts to arrest platform motion; hence, coupling DPCA with SAR processing on each receive channel is often called arrested SAR processing.

Timing errors, non-ideal sensor geometry, beampattern mismatch and receiver channel errors limit DPCA's capability [6, 62]. To provide effective clutter suppression, STAP is applied to complex, multi-channel SAR data. Approaches for applying STAP to multi-channel SAR are given in [62], and for some architectures, appear reminiscent of the post-Doppler STAP methods described in Section V; actual measured data from the German AER-II multi-channel SAR system are used to evaluate performance.

E. Non-Sidelooking Configurations, Canted Arrays and Nonlinear Arrays

Much of the basic STAP theory is developed for side-looking array radar (SLAR) configurations [6–8]. This configuration is most benign, since ground clutter Doppler and spatial frequencies are proportional (see (6) and (14)), thereby suggesting a common angle-Doppler response over range. The stationary nature of the clutter angle-Doppler response facilitates covariance matrix estimation.

In forward-looking array radar (FLAR), an angle-Doppler dependence of the ground clutter returns exists for slant ranges less than five times the platform altitude. Reference [6] describes this angle-Doppler dependence for the FLAR configuration. The consequent covariance matrix estimation errors resulting from the sensor geometry-induced non-stationary nature of the training data set leads to degraded detection performance. To improve performance, a Doppler compensation method is given in [63]. Similar clutter non-stationarity occurs for canted arrays, which likewise can be compensated through processing [64].

Non-linear arrays are under consideration for effective sensor airframe mounting. The performance of a circular, curved array is described in [65]. The nonlinear nature of the array induces clutter nonstationarity—as a result of the same mechanism leading to non-stationarity in FLAR and canted arrays—which consequently degrades the adaptive filter performance. Application of a time-varying filter function [66, 67] is used in [65] to improve performance.

VII. SUMMARY

This paper provides a tutorial overview of STAP for aerospace moving target indication radar applications. STAP is a data-domain implementation of an optimum filter applied to space-time samples for a given range.

Our introductory comments mentioned the monotonic relationship between SINR and probability of detection for a fixed false alarm rate. In clutter-limited environments, STAP efficiently maximizes output SINR to maximize the probability of detection. To understand the properties of ground clutter and narrowband noise jamming, we described spatial, temporal and space-time sampling using a multi-channel, multi-pulse radar system. We then described space-time clutter and jammer models; the clutter model was validated with measured, airborne radar data. Subsequently, we developed the STAP weight vector formulation and then discussed critical performance metrics (probability of detection, SINR loss and improvement factor). We formulated reduced-dimension and reduced-rank STAP methods as approaches to improve statistical convergence and, in the reduced-dimension STAP case, to mitigate computational burden. The paper culminated with a brief overview of some current topics: STAP for space-based radar, bistatic STAP, knowledge-aided STAP, STAP application to multi-channel SAR, and the use of STAP in non-sidelooking array configurations.

ACKNOWLEDGMENT

The author gratefully acknowledges his collaboration with Dr. Daniel Leatherwood, of the Georgia Tech Research Institute, on space based radar system modeling and signal processing algorithm development, and is thankful to Dr. Leatherwood for providing Figs. 11–12. Furthermore, the author thanks the anonymous reviewers for their constructive comments.

REFERENCES

[1] Skolnik, M. I. (1980)
Introduction to Radar Systems (2nd ed.).
New York: McGraw Hill, 1980.

[2] Howells, P. W. (1965)
Intermediate frequency sidelobe canceller.
U.S. Patent 3202990, Aug. 24, 1965.

[3] Applebaum, S. P. (1966)
Adaptive Arrays.
Syracuse University Research Corporation, Rept. SPL TR 66-1, Aug. 1966.

[4] Widrow, B., Mantey, P. E., Griffiths, L. J., and Goode, B. B. (1967)
Adaptive antenna systems.
IEEE Proceedings, **55** (Dec. 1967), 2143–2159.

[5] Brennan, L. E., and Reed, I. S. (1973)
Theory of adaptive radar.
IEEE Transactions on Aerospace and Electronic Systems, **AES-9**, 2 (Mar. 1973), 237–252.

[6] Klemm, R. (1998)
Space-Time Adaptive Processing: Principles and Applications.
IEE Radar, Sonar, Navigation and Avionics, **9** (1998).

[7] Ward, J. (1994)
Space-Time Adaptive Processing for Airborne Radar.
Lincoln Laboratory Tech. Rept. ESC-TR-94-109, Dec. 1994.

[8] Jaffer, A. G., Baker, M. H., Ballance, W. P., and Staub, J. R. (1991)
Adaptive Space-Time Processing Techniques For Airborne Radars.
Rome Laboratory Technical Rept. TR-91-162, July 1991.

[9] Klemm, R. (Ed.) (1999)
Special issue on STAP.
IEE Electronics & Comm. Engineering Journal, Feb. 1999.

[10] Melvin, W. L. (Ed.) (2000)
Special section on STAP.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000).

[11] Johnson, D. H., and Dudgeon, D. E. (1993)
Array Signal Processing: Concepts and Techniques.
Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] Monzingo, R. A., and Miller, T. W. (1980)
Introduction to Adaptive Arrays.
New York: Wiley, 1980.

[13] Compton, R. T. (1988)
Adaptive Antennas: Concepts and Performance.
Englewood Cliffs, NJ: Prentice-Hall, 1988.

[14] Hudson, J. E. (1981)
Adaptive Array Principles.
IEE Press, 1981.

[15] Fenner, D. K., and Hoover, W. F. (1996)
Test results of a space-time adaptive processing system for airborne early warning radar.
In *Proceedings of 1996 IEEE National Radar Conference*, Ann Arbor, MI, May 13–16, 1996, 88–93.

[16] Haykin, S. (1996)
Adaptive Filter Theory (3rd ed.).
Upper Saddle River, NJ: Prentice-Hall, 1996.

[17] Reed, I. S., Mallett, J. D., and Brennan, L. E. (1974)
Rapid convergence rate in adaptive arrays.
IEEE Transactions on Aerospace and Electronic Systems, **AES-10**, 6 (Nov. 1974), 853–863.

[18] Blum, R. S., and McDonald, K. F. (2000)
Analysis of STAP algorithms for cases with mismatched steering and clutter statistics.
IEEE Transactions on Signal Processing, **48**, 2 (Feb. 2000), 301–310.

[19] Robey, F. C., Fuhrman, D. R., Kelly, E. J., and Nitzberg, R. (1992)
A CFAR adaptive matched filter detector.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 1 (Jan. 1992), 208–216.

[20] Chen, W. S., and Reed, I. S. (1991)
A new CFAR detection test for radar.
Digital Signal Processing, Vol. 1, Academic Press, 1991, 198–214.

[21] Kelly, E. J. (1986)
An adaptive detection algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **AES-22**, 1 (Mar. 1986), 115–127.

[22] Kraut, S., Scharf, L. L., and McWhorter, L. T. (2001)
Adaptive subspace detectors.
IEEE Transactions on Signal Processing, **49** (Jan. 2001), 1–16.

[23] Kay, S. M. (1998)
Fundamentals of Statistical Signal Processing: Detection Theory.
Upper Saddle River, NJ: Prentice-Hall, 1998.

[24] DiFranco, J. V., and Rubin, W. L. (1980)
Radar Detection, Dedham, MA: Artech-House, 1980.

[25] Nitzberg, R. (1984)
Detection loss of the sample matrix inversion technique.
IEEE Transactions on Aerospace and Electronic Systems, **AES-20**, 6 (Nov. 1984), 824–827.

[26] DiPietro, R. C. (1992)
Extended factored space-time processing for airborne radar.
In *Proceedings of 26th Asilomar Conference*, Pacific Grove, CA, Oct. 1992, 425–430.

[27] Wang, H., and Cai, L. (1994)
On adaptive spatial-temporal processing for airborne surveillance radar systems.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 3 (July 1994), 660–670.

[28] Blum, R., Melvin, W., and Wicks, M. (1996)
An analysis of adaptive DPCA.
In *Proceedings of 1996 IEEE National Radar Conference*, Ann Arbor, MI, May 13–16, 1996, 303–308.

[29] Haimovich, A. (1996)
The Eigencanceler: Adaptive radar by eigenanalysis methods.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 2 (Apr. 1996), 532–542.

[30] Guerci, J. R., Goldstein, J. S., and Reed, I. S. (2000)
Optimal and adaptive reduced-rank STAP.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 647–661.

[31] Tufts, D. W., Kirsteins, I., and Kumaresan, R. (1983)
Data-adaptive detection of a weak signal.
IEEE Transactions on Aerospace and Electronic Systems, **AES-19**, 2 (Mar. 1983), 313–316.

[32] Gabriel, W. F. (1986)
Using spectral estimation techniques in adaptive processing antenna systems.
IEEE Transactions on Antennas and Propagation, **AP-34**, 3 (Mar. 1986), 291–300.

[33] Kirsteins, I. P., and Tufts, D. W. (1994)
Adaptive detection using low rank approximation to a data matrix.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 1 (Jan. 1994), 55–67.

[34] Kirsteins, I. P., and Tufts, D. W. (1991)
Rapidly adaptive nulling of interference.
In M. Bouvet and Bienvenu (Eds.), *High Resolution Methods in Underwater Acoustics*, New York, NY: Springer-Verlag, 1991.

[35] Kirsteins, I. P., and Tufts, D. W. (1985)
On the probability density of signal-to-noise-ratio in an improved adaptive detector.
In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Tampa, FL, 1985, 572–575.

[36] Roman, J. R., Rangaswamy, M., Davis, D. W., Zhang, Q., Himed, B., and Michels, J. H. (2000)
Parametric adaptive matched filter for airborne radar applications.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 677–692.

[37] Rangaswamy, M., and Michels, J. H. (2001)
A parametric detection algorithm for space-time adaptive processing in non-Gaussian clutter.
In D. Cochran, L. White, and B. Moran (Eds.), *Defence Applications of Signal Processing*, Elsevier Science B.V., Amsterdam, Netherlands, 2001.

[38] Michels, J. H., Himed, B., and Rangaswamy, M. (2000)
Performance of STAP tests in Gaussian and compound-Gaussian clutter.
Digital Signal Processing, **10**, 4 (Oct. 2000), 309–324.

[39] Michels, J. H., Rangaswamy, M., and Himed, B. (2002)
Performance of parametric and covariance based STAP tests in compound-Gaussian clutter.
Digital Signal Processing, **12**, 2/3 (Apr./July 2002), 307–328.

[40] Cantafio, L. (1989)
Space-Based Radar Handbook.
Norwood, MA: Artech House, 1989.

[41] Rabideau, D., and Kogon, S. (1999)
A signal processing architecture for space-based GMTI radar.
In *Proceedings of 1999 IEEE Radar Conference*, Waltham, MA, 96–101.

[42] Leatherwood, D. A., Melvin, W. L., and Garnham, J. (2000)
High-fidelity MTI modeling and analysis of a distributed aperture spaceborne concept.
In *Proceedings of AIAA Space 2000 Conference and Expo*, Paper AIAA-2000-5187, Long Beach, CA, Sept. 19–21, 2000.

[43] Leatherwood, D. A., Melvin, W. L., and Berger, S. D. (2001)
Adaptive signal processing for a spaceborne distributed aperture.
In *Proceedings of 2001 AIAA Conference*, Paper AIAA-2000-4725, Albuquerque, NM, Aug. 2001.

[44] Leatherwood, D. A., and Melvin, W. L. (2003)
Adaptive processing in a nonstationary spaceborne environment.
In *Proceedings of 2003 IEEE Aerospace Conference*, Big Sky, MT, Mar. 8–15, 2003.

[45] Klemm, R. (2000)
Comparison between monostatic and bistatic antenna configurations for STAP.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 596–608.

[46] Melvin, W. L., Callahan, M. J., and Wicks, M. C. (2000)
Adaptive clutter cancellation in bistatic radar.
In *Proceedings of 34th Asilomar Conference*, Pacific Grove, CA, Oct. 29–31, 2000, 1125–1130.

[47] Himed, B., Michels, J. H., and Zhang, Y. (2001)
Bistatic STAP performance analysis in radar applications.
In *Proceedings of 2001 IEEE Radar Conference*, Atlanta, GA, May 1–3, 2001, 198–203.

[48] Kogon, S. M., and Zatman, M. A. (2000)
Bistatic STAP for airborne radar systems.
In *Proceedings of IEEE SAM 2000*, Lexington, MA, Mar. 2000.

[49] Melvin, W. L., Callahan, M. J., and Wicks, M. C. (2002)
Bistatic STAP: Application to airborne radar.
In *Proceedings of 2002 IEEE Radar Conference*, Long Beach, CA, Apr. 22–25, 2002, ISBN 0-7803-7358-8.

[50] Himed, B., Zhang, Y., and Hajjari, A. (2002)
STAP with angle-Doppler compensation for bistatic airborne radars.
In *Proceedings of 2002 IEEE Radar Conference*, Long Beach, CA, Apr. 22–25, 2002, ISBN 0-7803-7358-8.

[51] Himed, B. (2002)
Effects of bistatic clutter dispersion on STAP systems.
In *Proceedings of 2002 IEE International Radar Conference*, Edinburgh, UK, Oct. 15–17, 2002, 360–364.

[52] Pearson, F., and Borsari, G. (2001)
Simulation and analysis of adaptive interference suppression for bistatic surveillance radars.
In *Proceedings of 2001 ASAP Symposium*, Lexington, MA, Mar. 13, 2001.

[53] Nitzberg, R. (1990)
An effect of range-heterogeneous clutter on adaptive Doppler filters.
IEEE Transactions on Aerospace and Electronic Systems, **26**, 3 (May 1990), 475–480.

[54] Armstrong, B. C., Griffiths, H. D., Baker, C. J., and White, R. G. (1995)
Performance of adaptive optimal Doppler processors in heterogeneous clutter.
IEE Proceedings on Radar, Sonar, Navigation, **142**, 4 (Aug. 1995), 179–190.

[55] Melvin, W. L. (2000)
Space-time adaptive radar performance in heterogeneous clutter.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 621–633.

[56] Bergin, J., Techau, P., Melvin, W. L., and Guerci, J. R. (2002)
GMTI STAP in target-rich environments: site-specific analysis.
In *Proceedings of 2002 IEEE Radar Conference*, Long Beach, CA, Apr. 22–25, 2002, ISBN 0-7803-7358-8.

[57] Melvin, W., Wicks, M., Antonik, P., Salama, Y., Li, P., and Schuman, H. (1998)
Knowledge-based space-time adaptive processing for AEW radar.
IEEE AES Systems Magazine, **13**, 4 (Apr. 1998), 37–42.

[58] Antonik, P., Schuman, H. K., Melvin, W. L., and Wicks, M. C. (1997)
Implementation of knowledge-based control for space-time adaptive processing.
In *Proceedings 1997 IEE International Radar Conference*, Edinburgh, Scotland, Oct. 14–16, 1997, 478–482.

[59] Weiner, D. D., Capraro, G. T., Capraro, C. T., Berdan, G. B., and Wicks, M. C. (1998)
An approach for utilizing known terrain and land feature data in estimation of the clutter covariance matrix.
In *Proceedings of 1998 IEEE National Radar Conference*, Dallas, TX, May 12–13, 1998, 381–386.

[60] Farina, A., Lombardo, P., and Pirri, M. (1999)
Nonlinear STAP processing.
IEE Electronics & Comm. Engineering Journal, Feb. 1999, pp. 41–48.

[61] Guerci, J. R. (2002)
Knowledge-aided sensor signal processing and expert reasoning.
In *Proceedings of 2002 Knowledge-Aided Sensor Signal Processing and Expert Reasoning (KASSPER) Workshop*, Washington, DC, Apr. 3, 2002, CD ROM.

[62] Ender, J. H. G. (1999)
Space-time processing for multichannel synthetic aperture radar.
IEE Electronics & Comm. Engineering Journal, Feb. 1999, 29–37.

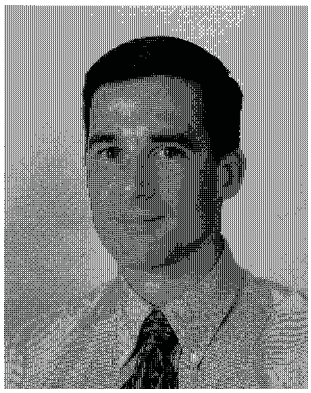
[63] Kreyenkamp, O., and Klemm, R. (2001)
Doppler compensation in forward-looking STAP radar.
IEE Proceedings of Radar, Sonar Navigation, **148**, 5 (Oct. 2001), 252–258.

[64] Borsari, G. (1998)
Mitigating effects on STAP processing caused by an inclined array.
In *Proceeding of 1998 IEEE Radar Conference*, Dallas, TX, May 1998, 135–140.

[65] Zatman, M. (2000)
Circular array STAP.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Apr. 2000), 510–517.

[66] Hayward, S. D. (1996)
Adaptive beamforming for rapidly moving arrays.
In *Proceedings of CIE International Conference of Radar* (IEEE Press), Beijing, China, Oct. 8–10, 1996, 480–483.

[67] Zatman, M. (2000)
The properties of adaptive algorithms with time varying weights.
In *Proceedings of IEEE Sensor & Multi-Channel Signal Processing Workshop* (SAM2000), Lexington, MA, Mar. 2000.



William L. Melvin (M'90—SM'99) received the B.S. degree (Magna Cum Laude), M.S., and Ph.D. degrees in electrical engineering from Lehigh University, Bethlehem, PA, in 1989, 1992, and 1994.

He received a commission as an officer in the United States Air Force on June 3, 1989, postponing his service commitment until completion of the Ph.D. degree. From Jan. 1994 until Jan. 1998 he was assigned to the United States Air Force Rome Laboratory, Signal Processing Branch, where he conducted research on improved detection of weak targets by airborne radar surveillance platforms. Upon leaving active duty service, he joined the Georgia Tech Research Institute, Sensors and Electromagnetic Applications Laboratory, where his current work involves multidimensional adaptive filtering, detection theory, and array signal processing. He is also a lecturer in several Georgia Tech Short Courses on sensor signal processing. He remains a U.S. Air Force Reserve Officer.

Dr. Melvin was awarded the Maj. General John J. Toomay Award for Excellence in Military Engineering. He is a member of Eta Kappa Nu and Tau Beta Pi, and a faculty associate of the Space Technology Advanced Research Center. He holds three U.S. patents.

Class-Specific Classifier: Avoiding the Curse of Dimensionality

PAUL M. BAGGENSTOSS, Member, IEEE
U.S. Naval Undersea Warfare Center

This article describes a new probabilistic method called the "class-specific method" (CSM). CSM has the potential to avoid the "curse of dimensionality" which plagues most classifiers which attempt to determine the decision boundaries in a high-dimensional featue space. In contrast, in CSM, it is possible to build classifiers without a common feature space. Separate low-dimensional features sets may be defined for each class, while the decision functions are projected back to the common raw data space. CSM effectively extends the classical classification theory to handle multiple feature spaces. It is completely general, and requires no simplifying assumption such as Gaussianity or that data lies in linear subspaces.

Manuscript received September 26, 2002; revised February 12, 2003.

This work was supported by the Office of Naval Research.

Author's address: U.S. Naval Undersea Warfare Center, Newport RI, 02841, E-mail: (p.m.baggenstoss@ieee.org).

I. INTRODUCTION AND BACKGROUND

The purpose of this article is to introduce the reader to the basic principles of classification with class-specific features. It is written both for readers interested in only the basic concepts as well as those interested in getting started in applying the method. For in-depth coverage, the reader is referred to a more detailed article [1].

Classification is the process of assigning data to one of a set of pre-determined class labels [2]. Classification is a fundamental problem that has to be solved if machines are to approximate the human functions of recognizing sounds, images, or other sensory inputs. This is why classification is so important for automation in today's commercial and military arenas.

Many of us have first-hand knowledge of successful automated recognition systems from cameras that recognize faces in airports to computers that can scan and read printed and handwritten text, or systems that can recognize human speech. These systems are becoming more and more reliable and accurate. Given reasonably clean input data, the performance is often quite good if not perfect. But many of these systems fail in applications where clean, uncorrupted data is not available or if the problem is complicated by variability of conditions or by proliferation of inputs from unknown sources. In military environments, the targets to be recognized are often uncooperative and hidden in clutter and interference. In short, military uses of such systems still fall far short of what a well-trained alert human operator can achieve.

We are often perplexed by the wide gap of performance between humans and automated systems. Allow a human listener to hear two or three examples of a sound—such as a car door slamming. From these few examples, the human can recognize the sound again and not confuse it with similar interfering sounds. But try the same experiment with general-purpose classifiers using neural networks and the story is quite different. Depending on the problem, the automated system may require hundreds, thousands, even millions of examples for training before it becomes both robust and reliable.

Why? The answer lies in what is known as the "curse of dimensionality." General-purpose classifiers need to extract a large number of measurements, or features, from the data to account for all the different possibilities of data types. The large collection of features form a high-dimensional space that the classifier has to sub-divide into decision boundaries. It is well-known that the complexity of a high-dimensional space increases exponentially with the number of measurements [3]—and so does the difficulty of finding the best decision boundaries from a fixed amount of training data. Unless a lot

is known about the data, allowing the features to be suitably conditioned so that the data samples fall in nicely organized patterns in the feature space, finding the "optimum" decision boundaries in a feature space above about 5 dimensions is futile [4]. Optimum decision boundaries require finding the probability distributions (probability density functions or PDFs) of each class in the feature space [2]. Sub-optimal decision boundaries, that is based on simplified probability models, or simple search procedures such as "nearest neighbor," can achieve very good performance if the data from the various classes are well separated in the feature space, but fail dramatically if there is any degradation of training data or input data quality.

These problems can potentially be avoided if we avoid working in a high-dimensional space. But how can we avoid working in high dimensions if all the measurements (features) carry pertinent information? One way is to keep a large number of features, but divide up the features according to their relevance to a particular class (class-specific features) and process them separately. Many schemes have been invented to try to find suitable rules for combining the processors [5, 6, 7, 8, 9, 10]. While they are on the right track, the problem with these classifiers is that they generally are unable to combine the results of the individual decisions in a way that is both theoretically optimal and completely general at the same time. What is needed is an extension to the classical theory of hypothesis testing that can account for class-specific features.

In answer to this need, the author proposed the class-specific method (CSM) in 1998 [11, 12, 13]. The initial formulation of the method suffered from several difficulties which were solved with the publication of the PDF projection theorem in 2000 [14, 15]. Further enhancements of the theory have resulted from the chain-rule [16, 1] a recursive application of the PDF projection theorem. The resulting classifier architecture, called the chain-rule processor [16, 1], blends the best aspects of signal processing and classification. CSM is completely general and makes no assumptions about the data such as that it yields to linear subspace decomposition. Nor does it require any special topology such as a binary tree of decisions. In fact, the classical feature classifier is a special case CSM that occurs when all classes are represented by a common feature set. But, unlike the classical classifier, CSM can circumvent the curse of dimensionality if each class can be represented (statistically described) using a separate low-dimensional feature set.

II. CLASSICAL APPROACH

The classical Bayesian classifier selects the most likely class hypothesis given the data according to

$$j^* = \arg\max_{j=1}^{M} p(H_j \mid \mathbf{x})$$

where $\mathbf{x}$ is the data and $\{H_1, H_2, \ldots, H_M\}$ are the M class hypotheses. Using Bayes rule, this may be written

$$j^* = \arg\max_{j} p(\mathbf{x} \mid H_j) p(H_j), \tag{1}$$

where $p(H_j)$ is the class prior probability for class H_j and $(\mathbf{x} \mid H_j)$ is the probability density functions (PDF) of the data assuming class H_j is true (otherwise known as the likelihood function). This classifier has the lowest expected cost (or lowest probability of error for equal class prior probabilities) of all classifiers [2, 17]. Unfortunately, the PDFs are unknown and need to be estimated from training data. Because the dimension of the raw data is too high, $\mathbf{x}$ has to be reduced to a set of information-bearing features using a feature transformation $\mathbf{z} = T(\mathbf{x})$. If it is possible to find a low-dimensional feature set that contains most or all of the necessary information, the problem can then be re-formulated in terms of $\mathbf{z}$. By regarding $\mathbf{z}$ as the data, the Bayesian feature classifier becomes

$$j^* = \arg\max_{j} \hat{p}(\mathbf{z} \mid H_j) p(H_j)$$

where $\hat{p}(\mathbf{z} \mid H_j)$ are the feature PDFs estimated from training data.

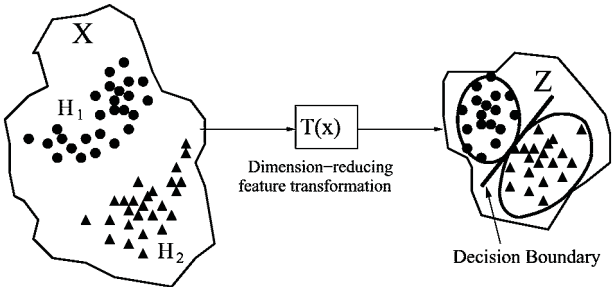


Fig. 1. Illustration of classical approach to classification. Original raw data space ($\mathbf{X}$) is mapped to a feature space ($\mathbf{Z}$) where the PDFs are approximated (ellipses) and decision boundaries (line) are constructed. Because of potential information loss, the classes can become overlapped in $\mathbf{Z}$, causing classification errors.

The classical approach to classification is summarized graphically in Fig. 1 for two data classes. The original raw data space ($\mathbf{X}$) is mapped to a feature space ($\mathbf{Z}$) where the PDFs are estimated and the decision boundaries are constructed. The curse of dimensionality forces the following trade-off: If the feature dimension is too high, there are severe errors in PDF estimation causing classification errors. If the feature dimension is too low, the loss of information causes the classes to become overlapped in $\mathbf{Z}$, also causing classification errors. There may be no feature dimension where the performance is acceptable. In short, the curse of dimensionality cannot be overcome. Once the raw data is discarded in favor of a common set of features, all hope is lost for achieving the best possible performance.

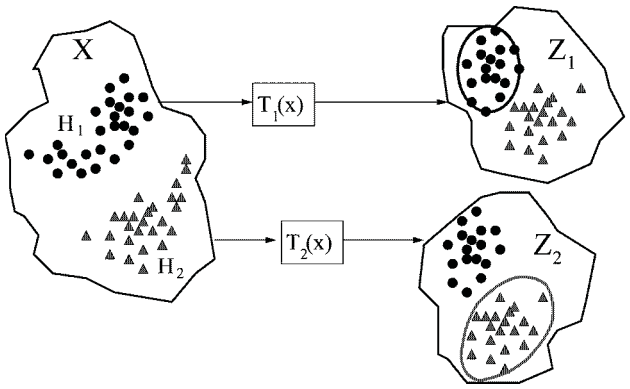


Fig. 2. Illustration of the first step of CSM. A separate feature transformation is designed for each class. Feature PDFs for each class are estimated on the corresponding feature space (ellipses).

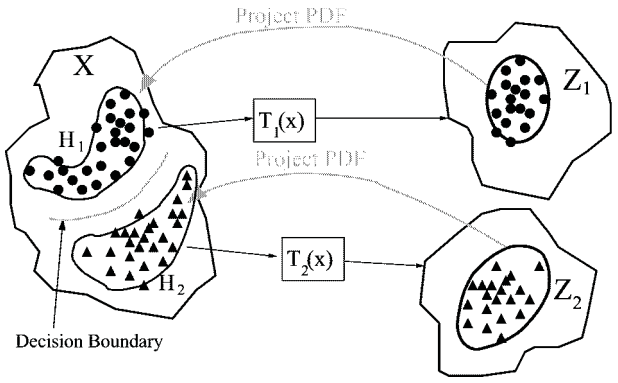


Fig. 3. Illustration of the second step of CSM. Feature PDFs are projected back to the raw data space where the decision boundaries are constructed.

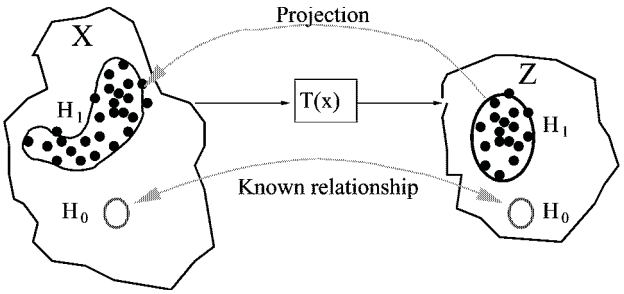


Fig. 4. Illustration of the PDF projection operation. Projection can be accomplished only if is possible to know both the raw data PDF and feature PDF for some reference hypothesis H_0.

III. CLASS-SPECIFIC METHOD (CSM)

Because this is a tutorial paper, we present only the most basic mathematical concepts of CSM. For further reading, the reader is referred to the most recent publications [1].

The classical approach loses the fight against the curse of dimensionality because it puts "all of its eggs in one basket." It requires a low-dimensional feature set that contains all of the necessary information—an impossible request. Instead of discarding the raw data, CSM actually operates in the raw data domain—but it estimates the PDFs in low-dimensional feature spaces. This requires a two-step procedure.

A. Step 1: Feature Transformation and PDF Estimation

First, the raw data is transformed into class-specific low-dimensional feature spaces. Let

$$\begin{aligned}\mathbf{z}_1 &= T_1(\mathbf{x})\\ \mathbf{z}_2 &= T_2(\mathbf{x})\\ &\vdots\\ \mathbf{z}_M &= T_M(\mathbf{x})\end{aligned}$$

be the M different feature sets and feature transformations. The PDFs $p(\mathbf{z}_m \mid H_m)$, $1 \le m \le M$, are then estimated from training data. This first step is illustrated in Fig. 2.

B. Step 2: PDF Projection Back to Raw Data Domain

Next, CSM converts the feature PDFs into raw-data PDFs. It projects the PDFs back to the raw data domain where the decision boundaries are constructed. CSM avoids the complexity issues of the raw-data space because the projection operators (functions that transform the PDFs to the raw data domain) are known functions that can be determined exactly from the feature transformations. This last step is illustrated in Fig. 3.

C. How the Projection Works

Projecting the PDF from the feature domain back to the raw data domain is made possible by the PDF projection theorem [15, 14]. This theorem may be thought of as a generalization of the well-known change-of-variables theorem which relates the PDF of $\mathbf{y}$ to the PDF of $\mathbf{x}$ when related by the $1:1$ transformation $\mathbf{y} = f(\mathbf{x})$. For continuous invertible transformations, it is a simple matter to recover the PDF of $\mathbf{x}$ from the PDF of $\mathbf{y}$ using the formula

$$p_x(\mathbf{x}) = \left|\frac{\partial y}{\partial x}\right| p_y(\mathbf{y}). \tag{2}$$

The PDF projection theorem (PPT) is a generalization of (2) for many-to-one transformations. Under certain conditions (having to do with sufficient statistics) it is possible to actually recover $p_x(\mathbf{x})$ from the feature PDF. In general, however, the PPT can only find a particular one of the many possible PDFs of $\mathbf{x}$ that could have produced the given feature PDF. This particular choice has some nice properties. The projection operation is illustrated in Fig. 4. Projection can only be accomplished if it is possible to know both the raw data PDF and feature PDF under some

reference hypothesis H_0. In general, it is impossible to determine the raw data PDF if all we know is the feature PDF. The projection operation finds only an approximation to the true raw data PDF of the given class hypothesis. The projected PDF defined on the raw data domain is given mathematically by

$$p_p(\mathbf{x} \mid H_j) \triangleq \left[\frac{p(\mathbf{x} \mid H_{0,j})}{p(\mathbf{z}_j \mid H_{0,j})} \right] \hat{p}(\mathbf{z}_j \mid H_j) \tag{3}$$

where $H_{0,j}$ are the class-specific reference hypotheses. Thus, the partial derivative (which generalizes to the determinant of the Jacobian matrix for multi-dimensions) in (2) is replaced by a ratio of PDFs. As expected, the PDF projection theorem simplifies to (2) for continuous invertible transformations. It may be proved [15, 14] that $p_p(\mathbf{x} \mid H_j)$ is a PDF, so it integrates to 1 on the raw data space, and that it is a member of the class of PDFs which generate the original feature PDF $\hat{p}(\mathbf{z} \mid H_j)$. This means that if a random variable $\mathbf{x}$ is drawn from the PDF in (3), and the result is transformed by the feature transformation $\mathbf{z}_j = T_j(\mathbf{x})$, then the PDF of $\mathbf{z}_j$ will be precisely $\hat{p}(\mathbf{z}_j \mid H_j)$, i.e. the projection process comes full circle.

Various interpretations of the projection theorem can be suggested. One interpretation is that since there are an infinite number of raw data PDFs that generate the feature PDF, it is necessary to invoke a constraint so that one unique raw data PDF can be found. The applicable constraint is that the likelihood ratio with respect to the reference hypothesis remains constant in either domain:

$$\frac{p_p(\mathbf{x} \mid H_j)}{p(\mathbf{x} \mid H_{0,j})} = \frac{\hat{p}(\mathbf{z}_j \mid H_j)}{p(\mathbf{z}_j \mid H_{0,j})}.$$

Another interpretation is possible if we reverse the usual thinking. Normally we start with two statistical hypothesis, then seek a sufficient statistic for differentiating between them. But we could also start with just one hypothesis ($H_{0,j}$) and a statistic ($\mathbf{z}_j$) and ask "what would be a second hypothesis for which $\mathbf{z}_j$ is sufficient against $H_{0,j}$?". The PDF constructed according to (3) is the second hypothesis we seek. Thus, $\mathbf{z}_j$ is sufficient for $H_{0,j}$ versus the hypothesis that $p_p(\mathbf{x} \mid H_j)$ is true.

D. How to Choose the Reference Hypothesis

A detailed mathematical treatment of the issues surrounding the reference hypothesis are given in [1]. Briefly, the conditions that H_0 must satisfy for the projection (3) to result in a valid PDF are that $p(\mathbf{z} \mid H_0)$ must never be precisely zero at any place where sample data can lie—otherwise, the term in the brackets in (3) cannot be evaluated. This is a rather mild constraint, easily satisfied by the most common PDFs such as Gaussian and exponential (assuming negative values are illegal). As long as this condition holds, (3) will be a PDF and will be among the class of PDFs which give rise to the specified feature PDF $\hat{p}(\mathbf{z}_j \mid H_j)$ when transformed by the given feature transformation. That being said, there are good and bad choices for $H_{0,j}$. A good choice of $H_{0,j}$ (one that will result in a good approximation to $p(\mathbf{x} \mid H_j)$) is one for which the features $\mathbf{z}_j = T_j(\mathbf{x})$ are approximately sufficient statistics for testing H_j versus $H_{0,j}$. Sufficiency is meant in the statistical sense, and does not mean "just good enough." It means that all information necessary to separate $H_{0,j}$ from H_j is present. Remember, though, that this condition is a goal, not a requirement and should not discourage anyone from trying a particular feature set. The closer the sufficiency condition can be approximated, the better the projected PDF will approximate $p(\mathbf{x} \mid H_j)$. It is also advisable that $H_{0,j}$ be such that $p(\mathbf{x} \mid H_{0,j})$ and $p(\mathbf{z} \mid H_{0,j})$ can both be determined either in closed form, or else to a good approximation, even in the (far) tails.

E. How to Build a CSM Classifier

By substitution of (3) into the Bayesian classifier (1), the CSM classifier results:

$$j^* = \arg\max_{j=1}^{M} \left[\frac{p(\mathbf{x} \mid H_{0,j})}{p(\mathbf{z}_j \mid H_{0,j})} \right] \hat{p}(\mathbf{z}_j \mid H_j) p(H_j). \tag{4}$$

The ratio

$$\mathrm{J}(\mathbf{x}, T_j, H_{0,j}) \triangleq \left[\frac{p(\mathbf{x} \mid H_{0,j})}{p(\mathbf{z}_j \mid H_{0,j})} \right] \tag{5}$$

we call the "J-function" and may be considered generalized Jacobian or correction term necessary to create the optimal Bayes classifier from the various feature PDFs.

F. When is CSM Optimal?

Clearly if the projected PDFs (3) are valid PDFs, no matter if they are accurate approximations to the desired PDFs $p(\mathbf{x} \mid H_j)$, the classifier (4) is a valid probabilistic classifier. Optimality occurs when the projected PDFs are equal to the desired PDFs. This happens when (1) the estimated feature PDFs, $\hat{p}(\mathbf{z}_j \mid H_j)$, are equal to the true feature PDFs, and (2) when the class-specific features, $\mathbf{z}_j = T_j(\mathbf{x})$ are sufficient statistics for deciding between the given class H_j and the chosen reference hypotheses $H_{0,j}$. Because the designer can choose both $T_j(\,)$ and $H_{0,j}$, it is to the designer's benefit to choose them jointly to approximate this condition. Note also that $H_{0,j}$ must be chosen from those hypotheses for which it is possible to solve for both $p(\mathbf{x} \mid H_{0,j})$ and $p(\mathbf{z}_j \mid H_{0,j})$. It is not always easy, but great strides have been made in recent years in being able to solve for the feature PDFs for many useful types of features [18, 19].

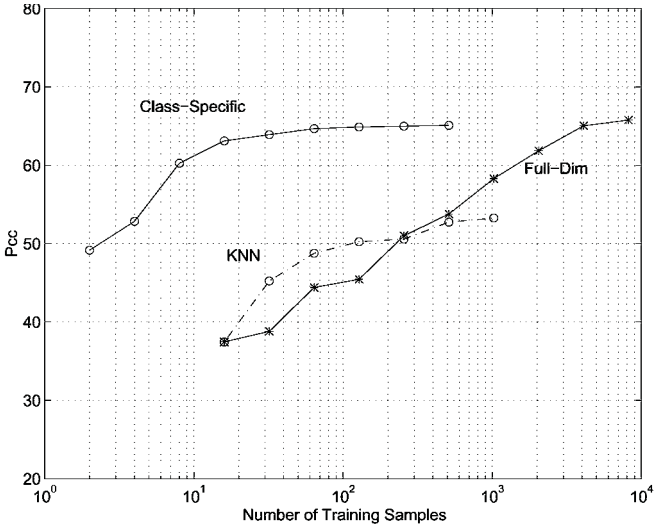


Fig. 5. Classification performance of CSM compared with classical approach in a 9-class synthetic data experiment. Classical approach used an 11-dimensional feature set composed of the union of all class-specific features.

G. Why is CSM Better Than the Classical Approach?

Both CSM and the classical approach have the same theoretical performance because they are both based on the optimal Bayesian classifier (1). Indeed, this is demonstrated in an experiment where a class-specific classifier was compared to a classical classifier using exactly the same features [11]. In the 9-class synthetic data experiment, the class-specific classifier used feature sets of dimension between 1 and 2, while the classical (full-dimensional) classifier operated on an 11-dimensional feature set (the union of the class-specific features). The performance was plotted as a function of the number of training samples and is repeated in Fig. 5. It shows that although the maximum performance of the classical classifier was the same, it required more than two orders of magnitude more training data to achieve it. Now imagine that only about 100 samples were available—observe on the graph the gap in performance that would exist. But the classical approach can never attain the promised performance because it needs to form a common feature set where the PDFs are estimated. The curse of dimensionality exacts a heavy toll on performance. For a given maximum feature dimension, CSM can collect much more information from the raw data because it can divide the information up according to class.

H. Paradigm Shift

Those that have worked with the classical approach have difficulty changing over to CSM which is an entirely new paradigm. Someone trained to view features as carrying information to distinguish one class from another may have a difficult time

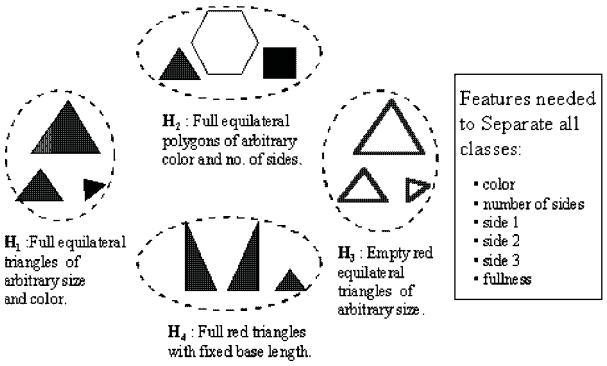


Fig. 6. Classical paradigm for a notional 4-class classification problem. Box on the right lists six measurements or "features" useful for classifying the four classes.

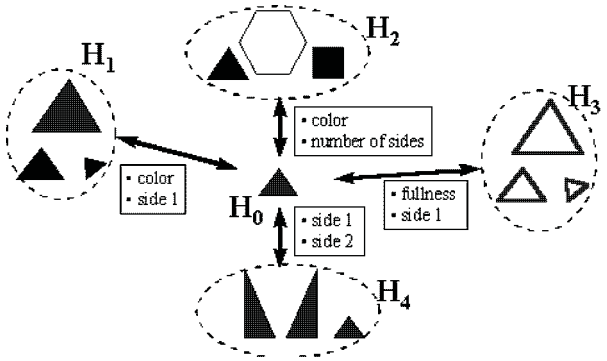


Fig. 7. Class-specific paradigm for a fixed reference hypothesis. Features (in box) are required to discriminate each class from the common reference hypothesis. Note that fewer features are required for the simpler binary problems.

viewing features in a way that ignores the other classes. A simple geometric example can illustrate the paradigm shift. Fig. 6 shows a notional 4-class problem involving sets of geometric shapes. The classical paradigm involves finding features that are able to discriminate among the four classes. A list of six measurements or "features" are provided in the yellow box on the right side of the figure. These six features are adequate for discrimination among the four classes. The mathematical implementation of the classical feature paradigm involves the maximization of the feature PDF:

$$j^* = \arg\max_j p(\mathbf{z} \mid H_j)$$

where $\mathbf{z} = T(\mathbf{x})$ is a common feature set.

Fig. 7 illustrates the class-specific paradigm using a fixed reference hypothesis. The features are required to discriminate each class from the common reference hypothesis. This is the original formulation of CSM but has a number of difficulties arising from the use of a common fixed reference hypothesis. The mathematical implementation of the fixed-reference class-specific paradigm involves the maximization of

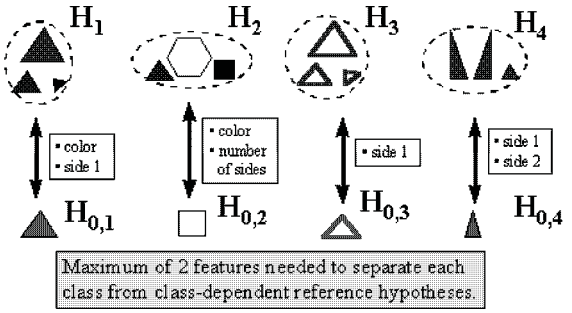


Fig. 8. Class-specific paradigm for class-specific reference hypotheses. Features are required to discriminate each class from the corresponding class-specific reference hypothesis.

the likelihood ratios

$$j^* = \arg\max_j \frac{p(\mathbf{z}_j \mid H_j)}{p(\mathbf{z}_j \mid H_0)}$$

where $\mathbf{z}_j = T_j(\mathbf{x})$, for $j = 1,\ldots,M$, are class-specific feature sets.

Fig. 8 illustrates the class-specific paradigm using class-specific reference hypotheses. Class-specific features are not chosen to discriminate a class from other classes, they are chosen to discriminate each class from the corresponding class-specific reference hypothesis which may be regarded as a special member of the class. In effect, this means the features are chosen to describe the class. It is important to remember that discrimination happens automatically if each class is well described. In effect, choosing features for description results in the same (or more) feature information content as discrimination but it assigns the information only to those classes for which it is relevant. In spite of this, it is a difficult paradigm shift to make for many people who have been taught to choose features for discriminatory power. The mathematical implementation of the class-specific paradigm involves the maximization of the projected PDFs

$$j^* = \arg\max_j p_p(\mathbf{x} \mid H_j)$$

where

$$p_p(\mathbf{x} \mid H_j) = \frac{p(\mathbf{x} \mid H_{0,j})}{p(\mathbf{z}_j \mid H_{0,j})} p(\mathbf{z}_j \mid H_j)$$

where $\mathbf{z}_j = T_j(\mathbf{x})$, for $j = 1,\ldots,M$, are class-specific feature sets and $H_{0,j}$, for $j = 1,\ldots,M$, are class-specific reference hypotheses.

I. Working in the Raw Data Domain

CSM creates decision boundaries in the raw data domain instead of in a common feature domain. This sounds troublesome at first. After all, the raw data dimension can be very large and we are interested in reducing the dimension! But remember that the dimension is only a problem for PDF estimation which happens on the low-dimensional feature space, not in the raw data space. Projecting to the raw data domain is done for us by the J-function (5) which does not suffer from the dimensionality curse because it does not need to be found empirically. The J-function can be determined exactly by analysis of the feature transformations.

There is a clear advantage to working in the raw data domain because it is a common ground where everything can be compared fairly. Interestingly, CSM is not the first attempt to work in the raw data domain. For example, Bishop [20] creates raw data PDFs, but his approach requires linear transformations to be tractable and amounts to something akin to principal component analysis (PCA). CSM, on the other hand, is completely general. The "ace in the hole" is the fact that the projected PDF is indeed a PDF and it depends only upon a few parameters—the parameters of the feature PDF and of the feature transformation and reference hypothesis. All of these parameters are "fair game" in a maximum likelihood maximization. Have you an idea for a better feature set? Compare it to the existing feature set based on the maximum likelihood principle. Have you an idea for a better reference hypothesis? Compare it to the existing reference hypothesis based on the maximum likelihood principle. This idea can is represented mathematically as

$$L(\mathbf{x}_1,\ldots,\mathbf{x}_K,H_0,T,\theta) = \max_{H_0,T,\theta} \sum_{k=1}^{K} \log\left\{\left[\frac{p(\mathbf{x}_k \mid H_0)}{p_z(T(\mathbf{x}_k) \mid H_0)}\right] p_z(T(\mathbf{x}_k);\boldsymbol{\theta})\right\} \quad (6)$$

K is the number of independent data samples and the subscript z is a reminder that the PDF $p_z(\,)$ is a function of the features $\mathbf{z} = T(\mathbf{x})$. To avoid "over-training," when implementing (6) in practice, it is recommended that the data be partitioned into separate training and testing sets for cross-validation.

J. Classifying Without Training

The PPT (3) is a decomposition of the raw data PDF into a trained and an untrained factors. The trained factor is the feature PDF $\hat{p}(\mathbf{z}_j \mid H_j)$ which needs to be estimated from training data of the corresponding class. The untrained factor $[p(\mathbf{x} \mid H_{0,j})/p(\mathbf{z}_j \mid H_{0,j})]$ is a known function of the input raw data $\mathbf{x}$, feature transformation $T_j(\,)$, and reference hypothesis $H_{0,j}$. But, for a fixed $\mathbf{x}$, it also can be viewed as a function of j. Thus, it contributes, sometimes in a dominant role, to the classification

decision. While the trained component asks "how does this sample compare with trained patterns?", the untrained component asks "how well does this feature set represent this raw data sample?". The untrained component can be seen as a generalization of a matched filter.

To see this, we consider a bank of linear matched filters as a set of class-specific feature extractors

$$z_j = T_j(\mathbf{x}) = |\mathbf{w}_j'\mathbf{x}|^2$$

where $\mathbf{w}_j$ is a signal template. Let $\mathbf{w}_j$ be normalized such that $\mathbf{w}_j'\mathbf{w}_j = 1$. The simple matched filter bank classifier is given by

$$j^* = \arg\max_j z_j.$$

Let us now design a class-specific classifier for these features. Under the reference hypothesis of independent Gaussian noise of variance 1, z_j is distributed $\chi^2(1)$

$$p(z_j \mid H_0) = \frac{1}{\sqrt{2\pi z_j}} \exp\left\{-\frac{z_j}{2}\right\}.$$

The PDF of $\mathbf{x}$ under H_0 is the Gaussian PDF

$$p(\mathbf{x} \mid H_0) = (2\pi)^{-N/2} \exp\left\{-\frac{1}{2}\sum_{n=1}^{N} x_n^2\right\}.$$

The log-J-function is easily shown to be

$$\begin{aligned}\log \mathrm{J}_j(\mathbf{x}, z_j) &= \log p(\mathbf{x} \mid H_0) - \log p(z_j \mid H_0) \\ &= \tfrac{1}{2}(\log z_j + z_j) + C(\mathbf{x})\end{aligned}$$

where $C(\mathbf{x})$ does not depend on j. The complete class-specific classifier is:

$$\arg\max_j \log \mathrm{J}_j(\mathbf{x}, z_j) + \log p(\mathbf{z}_j \mid H_j). \tag{7}$$

Since $\log \mathrm{J}_j$ is a monotonic increasing function of z_j, using only $\log \mathrm{J}_j(\mathbf{x}, z_j)$ as a classifier is equivalent to the matched filter bank classifier. Note that by ignoring the last term in (7) effectively assumes that each class has the same expected amplitude distribution.

It is clear that classification is quite possible without training as long as each class requires a distinctly different feature set for representation. This idea should not be taken literally without some care. Generalizing the "J-function-only" classifier to cases where the features are not matched filters, requires that some kind of a priori feature PDF should be used to account for differences in feature dimension and scaling. Note that this requirement is relaxed if the J-function is highly dominant.

IV. CHAIN RULE AND THE CHAIN-RULE PROCESSOR

As part of the paradigm shift from the classical architecture, we recommend looking at a sophisticated general-purpose classifier as a bank of signal processors. Each signal processor may be thought of as an optimal detector for differentiating the given class from the corresponding reference hypothesis. Each signal processor may be composed of multiple processing stages. If we regard the feature transformation $\mathbf{z} = T(\mathbf{x})$ as a single step, we write the projected PDF as

$$p_p(\mathbf{x} \mid H_1) = \left[\frac{p(\mathbf{x} \mid H_0)}{p(\mathbf{z} \mid H_0)}\right] \hat{p}(\mathbf{z} \mid H_1).$$

However, if we regard the feature transformation as three separate stages, $\mathbf{y} = T'(\mathbf{x})$, $\mathbf{w} = T''(\mathbf{y})$, then $\mathbf{z} = T'''(\mathbf{w})$, we may apply the PDF projection theorem recursively. For the first stage, we have

$$p_p(\mathbf{x} \mid H_1) = \left[\frac{p(\mathbf{x} \mid H_0)}{p(\mathbf{y} \mid H_0)}\right] p_p(\mathbf{y} \mid H_1).$$

Applying the same concept to $p_p(\mathbf{y} \mid H_1)$, we have

$$p_p(\mathbf{y} \mid H_1) = \left[\frac{p(\mathbf{y} \mid H_0')}{p(\mathbf{w} \mid H_0')}\right] p_p(\mathbf{w} \mid H_1)$$

and so on. The complete break-down is written

$$p_p(\mathbf{x} \mid H_1) = \left[\frac{p(\mathbf{x} \mid H_0)}{p(\mathbf{y} \mid H_0)}\right] \left[\frac{p(\mathbf{y} \mid H_0')}{p(\mathbf{w} \mid H_0')}\right] \left[\frac{p(\mathbf{w} \mid H_0'')}{p(\mathbf{z} \mid H_0'')}\right] \hat{p}(\mathbf{z} \mid H_1) \tag{8}$$

where H_0, H_0', H_0'' are reference hypotheses suited to each stage in the processing chain. The advantage of this approach is first that many processing chains may share the same first stages of processing, thus saving processing. Furthermore, analyzing just one stage at a time simplifies the analysis. Finally, there is great advantage in software modularity because each stage of processing can be encapsulated as a module.

A. Feature Modules

Feature modules are pre-packaged software modules that contain both feature calculation and J-function calculation. The three modules necessary for implementing the above three-stage chain would be

$$\text{Module 1:} \quad \mathbf{y} = T'(\mathbf{x}), \qquad j_1 = \log \frac{p(\mathbf{x} \mid H_0)}{p(\mathbf{y} \mid H_0)}$$

$$\text{Module 2:} \quad \mathbf{w} = T''(\mathbf{y}), \qquad j_2 = \log \frac{p(\mathbf{y} \mid H_0')}{p(\mathbf{w} \mid H_0')}$$

$$\text{Module 3:} \quad \mathbf{z} = T'''(\mathbf{w}), \qquad j_3 = \log \frac{p(\mathbf{w} \mid H_0'')}{p(\mathbf{z} \mid H_0'')}.$$

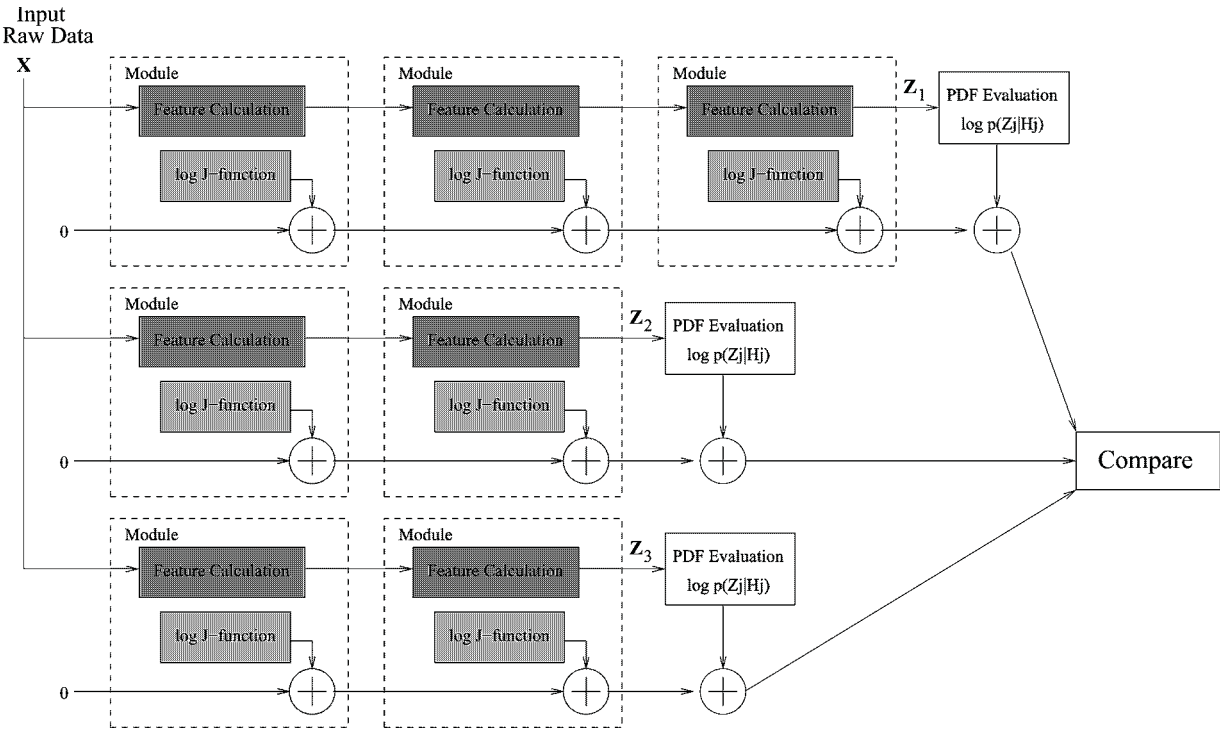


Fig. 9. Block diagram of a class-specific classifier using chain rule processors.

Completion of the processing chain is accomplished by accumulating the "correction terms"

$$\log p_p(\mathbf{x} \mid H_1) = j_1 + j_2 + j_3 + \log \hat{p}(\mathbf{z} \mid H_1). \qquad (9)$$

Class-specific classifiers can be rapidly designed by stringing together chains of pre-designed modules and accumulating the log J-function values.

B. Classifier Architecture

Implementation of a classifier is illustrated in Fig. 9. Each horizontal chain corresponds to one class. The chains are made up of series of modules. In accordance with (9), each module adds the corresponding correction term (J-function) to the stream. At the end, the aggregate J-function is added to the log feature PDF to arrive at the class output value.

V. BUILDING A CLASSIFIER

Because CSM is new, there is a large learning curve for those being introduced to it. There are many difficulties and pitfalls associated with building a classifier that should be mentioned.

A. Common Problems

The following is a list of problems and difficulties that are often encountered in designing and implementing a class-specific classifier.

1) *Sufficiency.* Recall that the designer should, as a goal, strive for a feature set/reference hypothesis combination where the features are approximately sufficient to discriminate the class of interest from the reference hypothesis. Sufficiency is does not mean "just enough," i.e. sufficient to get the job done. Sufficiency means all of the information has been extracted for discrimination. But this is a goal, not a requirement. It should not discourage anyone from using a set of features that is reasonable. A common mistake is to leave out a significant amount of information relating to the discrimination of a given class from the fixed reference hypothesis simply because it is not necessary to discriminate the data most if not all of the time. Here's an example. Consider discriminating a sinewave in additive correlated noise from a reference hypothesis of independent noise. While it may be adequate to concentrate on the sinewave, do not lose sight of the fact that the background noise also is different from H_0 and can significantly contribute to discrimination. It would be better in this case to use the correlated noise as the reference hypothesis.

2) *Using "all classes" as a reference hypothesis.* The suggestion that the reference hypothesis H_0 can be defined as a combination of "all classes" has been made several times. While in principle, H_0 can be any hypothesis, even this one, it defeats the purpose of class-specific features. This is because all the features are needed for discrimination of one class from all the others (see above item). Furthermore, the "all class" does not yield to

mathematical analysis, needed to compute the J-function.

3) *Tail probability errors*. A common misconception is that the denominator PDF in the J-function, $p(\mathbf{z} \mid H_0)$, can be estimated from training data. This is only true if all possible realizations of input data will be within the central part of the distribution and not highly unlikely. This could work, for example, with low-SNR signals. But such a system would perform poorly against high-SNR signals. It may be possible to position the reference hypothesis "close" to the data sample, then attempt to estimate the PDF of the features by random trials. Note that this would need to be re-done for each sample to be tested. It also must meet the requirements for a variable reference hypothesis.

4) *Segmentation*. Segmentation is the practice of carving up data into fixed-sized segments, then extracting features from each segment. This is an important first step in processing. The choice of segmentation size is often a difficult choice in traditional classifiers, because it is necessary to choose the segment size that is "good enough" for all classes. The class-specific method affords us the luxury of using different segmentation sizes for each class. This is because the likelihood comparisons are made on the raw data, which is always the same. A common error people make is that because of the different segment sizes used across different classes, the amount of raw data varies slightly due to the fact that the input data size is not divisible by all segment sizes. This can be a fatal error. It is necessary to only use an input data record size that can be divisible perfectly by each considered segment size.

5) *Failure to validate analysis*. Some form of absolute validation is necessary before using a module. In Section VII, a method of validating the J-function analysis is provided. There is no obvious way to locate errors except with this approach.

B. Module Design

There are more than one method of module design. The designer should not give up on using a good set of features because one module design approach fails—there may be another that works.

1) *Fixed reference hypothesis*. In this approach, a fixed reference hypothesis, such as independent Gaussian noise of a fixed variance is chosen. Then, the numerator and denominator densities of the J-function must be known exactly or approximated with the saddlepoint approximation [18] to insure accurate tail values.

2) *Floating reference hypothesis*. Floating the reference hypothesis by positioning it "close" to the data sample to be tested is a means of avoiding the tails. In general, a reference hypothesis cannot be made dependent on the data—this violates the concept of a statistical hypothesis. But under certain conditions, the dependence of the numerator and denominator of the J-function on changes in the reference hypothesis cancel out making the approach feasible [1]. The reference hypothesis may be floated as a function of the data as long as the features are sufficient statistics to distinguish all the possible hypotheses that may result. Floating the hypothesis may be a simple as adjusting the variance of the Gaussian assumption to agree with the sample variance of the data. Or, it may be as sophisticated as controlling the noise spectrum of an autoregressive model to agree with the observed autocorrelation function. The designer must insure that the features are sufficient or approximately sufficient to discriminate between the various reference hypotheses. For example, any feature set that contains the sample variance explicitly as a component or where the sample variance can be inferred from the features is fully sufficient to discriminate between any pair of variance hypotheses. Therefore, the variance of the reference hypothesis can be "floated."

3) *On-the-fly analysis*. It is possible to make a rapid Montecarlo-type analysis of the feature PDFs under a floating reference hypothesis.[1] This is useful when the PDF of the features defies analysis.

C. NUWC Module Library

The class-specific module is the building block of a class-specific classifier. It can be a source of frustration if a classifier designer wishes to use a feature set and cannot because no analysis is available. This is why a library of pre-tested class-specific modules is useful. A central repository of class-specific modules is being collected at a web-site at NUWC:

http://www.npt.nuwc.navy.mil/csf/index.html

To date, this collection includes the following feature transformations:

1) various invertible transformations;
2) spectrogram;
3) arbitrary linear functions of exponential RVs;
4) autocorrelation function (contiguous and non-contiguous);
5) autoregressive parameters (Reflection coefficients);
6) cepstrum (including MEL Cepstrum);
7) order statistics of independent RVs;
8) sets of quadratic forms.

New feature modules may be designed using the analysis tools of CR bound analysis (for maximum likelihood features) Readers are encouraged to use the library and submit their own contributed modules.

[1]The author wishes to thank Mario Fritz for this suggestion.

VI. EXAMPLES OF FEATURE CHAINS

Examples are necessary to make clear the important point thus far discussed. Each example shows how a feature transformation chain can be analyzed to obtain the correction term for PDF projection. When the feature transformation occurs in more than one step, the examples are broken down into separate modules. For each module, we provide the following information (all enclosed in boxes for clarity)

> **Feature Calculation:** The mathematical expression of the feature calculation.

> $\mathbf{H}_0$: A description of the reference hypothesis.

The class-specific correction term (J-function) is given by

$$\mathrm{J}(\mathbf{x},T,H_0) = \frac{p(\mathbf{x} \mid H_0)}{p(\mathbf{z} \mid H_0)}.$$

We separately provide the numerator and denominators:

> **Numerator PDF:** The numerator PDF of the J-function.

> **Denominator PDF:** The denominator PDF of the J-function.

The simplest kind of feature transformation is an invertible transformation. While these are not useful for dimension reduction, they are important for feature conditioning. For invertible transformations, the J-function is just the absolute value of the determinant of the Jacobian matrix of the transformation. Thus

$$\mathrm{J}(\mathbf{x},T) = |\det(\mathbf{J})|$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \frac{\partial z_1}{\partial x_3} & \cdots \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & \frac{\partial z_2}{\partial x_3} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

For invertible transformations, we provide the complete J-function only:

> **J-Function (Jacobian):** The log of the determinant of the Jacobian matrix.

A. Log Transformation

An example of an invertible transformation is the log function. Consider the transformation

> **Feature Calculation:** $z_i = \log(x_i), \quad 1 \le i \le M.$

We have $dz/dx = 1/x$, thus $\log \mathrm{J} = \log(1/x) = -\log x = -z$. For taking the logarithm of a vector of length M, we have

> **J-Function (Jacobian):** $\log \mathrm{J} = -\sum_{i=1}^{M} z_i.$

B. Variance Estimate

A very simple example of a class-specific module is the sample variance. Let $\mathbf{x}$ be a time-series of length N and let $\mathbf{z}$ be the variance estimate

> **Feature Calculation:** $z = T(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} x_i^2.$

Let the reference hypothesis be

> $\mathbf{H}_0$: Independent zero-mean Gaussian noise of variance 1.

Then the numerator of the J-function is

> **Numerator PDF:**
>
> $$\log p(\mathbf{x} \mid H_0) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\left(\sum_{i=1}^{N} x_i^2\right).$$

Since $\mathbf{z}$ has the Chi-squared distribution with N degrees of freedom (scaled by $1/N$), the denominator of the J-function is

> **Denominator PDF:**
>
> $$\log p(\mathbf{z} \mid H_0) = \log N - \log\Gamma\left(\frac{N}{2}\right) - \frac{N}{2}\log 2 + \left(\frac{N}{2} - 1\right)\log(Nz) - \frac{Nz}{2}.$$

C. Autocorrelation Function

A very useful feature set in stationary time-series analysis is the autocorrelation function (ACF). The ACF coefficient of lag τ is an estimate of the mean or expected value of the product $x_t x_{t-\tau}$, which for

stationary time-series, is independent of t. The ACF is the fundamental feature extraction behind many spectral estimation techniques with varying names such as linear predictive coding (LPC), autoregressive (AR) modeling, and reflection coefficients (RC). All of these methods are related and begin by estimating the ACF using a variety of methods. The benefit of AR modeling is that the spectral information can be boiled down to but a few coefficients which can hold spectral information with high resolution. The first $P+1$ ACF lags ($\tau = 0,1,\ldots,P$) are required for a Pth order AR model [21]. These ACF lags can then be transformed to RCs or AR coefficients using invertible transformations, thus they are equivalent from a modeling point of view. A good source of information on the topic is the book by Kay [21].

It may also be useful to use arbitrary ACF lags, rather than only the first $P+1$ lags. This is especially true when dealing with periodic time-series such as human voice, where the lag value at the pitch period is also of interest. Let $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ be a time-series of length N. We define the M-dimensional feature set $\mathbf{z}$ as the arbitrary ACF lags $k_1, k_2, \ldots, k_M$. Thus, the feature calculation is

Feature Calculation: $\mathbf{z} = [r_{k_1}, r_{k_1}, \ldots, r_{k_M}]$,

where $$r_k = \frac{1}{N}\sum_{i=1}^{N} x_i x_{[i+k]_N}$$

where the braces $[i-k]_N$ indicates modulo-N. These are known as the circular ACF estimates because of the modulo indexing. We choose this form of the ACF because it simplifies the analysis. A solution is available for arbitrary forms of the ACF based on quadratic forms [19], but is more complicated. As before, let the reference hypothesis be

$\mathbf{H}_0$: Independent zero-mean Gaussian noise of variance 1.

Then, as before, the numerator of the J-function is

Numerator PDF:

$$\log p(\mathbf{x} \mid H_0) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\left(\sum_{i=1}^{N} x_i^2\right).$$

There is no known closed-form expression for the joint PDF of $\mathbf{z}$ under H_0, although a cumbersome but exact expression is available for the normalized statistics $\tilde{r}_k = r_k / r_0$ (See [18] Section IIB). However, an approximation based on the saddlepoint approximation (SPA) [22] that is valid in the tails has been published. Specifically, in [18], Section IVB, the SPA for the scaled ACF estimates $\tilde{\mathbf{z}} = 2N\mathbf{z}$ are derived. The J-function denominator is thus,

Denominator PDF:

$$\log p(\mathbf{z} \mid H_0) = M \log(2N) + \log p(\tilde{\mathbf{z}} \mid H_0)$$

where $p(\tilde{\mathbf{z}} \mid H_0)$ is from [18], Section IVB.

D. Contiguous ACF and Reflection Coefficients

Reflection coefficients (RCs) are an alternate way of representing the information in an AR model. The RCs can be more convenient and easier to statistically model. Reflection coefficients (RCs) may be calculated from ACF estimates [21], and therefore we may use the results of Section VIC followed by a conversion to RCs. However, Section VIC is more general since it describes an approach that can handle arbitrary ACF lags; whereas the RCs are computed from a contiguous set of ACF lags (lags 0 through P). The use of contiguous ACF samples allows a different approach to analysis of the ACF features which is both instructive and useful for comparison purposes. If we use the circular ACF estimates as before, we can calculate the ACF samples by first computing the magnitude-squared DFT, then the inverse DFT. A third stage is necessary to convert to RCs and a fourth stage is used for further conditioning. The complete chain provided below has been found to be extremely versatile in modeling time-series. By segmenting the time-series, signals can be converted into sequences of feature vectors that can be statistically modeled using the a hidden Markov model (HMM). These feature sequences can also be converted back into time-series to validate the fidelity of the representation. As an additional check of model fidelity, the trained HMM can be used to generate random feature sequences, then converted into time-series for listening. Because of the versatility of CSM, each signal type can be represented using a particular choice of segment size and AR model order.

1. *Stage 1: Magnitude-Squared DFT*

In the first stage, we let $\mathbf{y} = [y_0, y_1, \ldots, y_{N/2}]$ be the magnitude-squared DFT of $\mathbf{x}$,

Feature Calculation:

$$y_k = \left|\sum_{i=1}^{N} x_i \exp\left\{-\frac{j2\pi(i-1)k}{N}\right\}\right|^2, \quad k = 0,1,\ldots,N/2.$$

As before, we let the reference hypothesis be

$\mathbf{H}_0$:	Independent zero-mean Gaussian noise of variance 1.

Also, as before, the numerator of the J-function is

Numerator PDF:

$$\log p(\mathbf{x} \mid H_0) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\left(\sum_{i=1}^{N} x_i^2\right).$$

The DFT bins are independent under H_0, but not identically distributed. DFT bins 0 and $N/2$ are real-valued so y_k have the Chi-squared distribution with 1 degree of freedom scaled by N, which we denote by $p_0(y)$:

$$p_0(y) = \frac{1}{N\sqrt{2\pi}}\frac{y_i}{N}^{-1/2}\exp\left\{-\frac{y_i}{2N}\right\}.$$

DFT bins 1 through $N/2-1$ are complex so y_k have the Chi-squared distribution with 2 degrees of freedom scaled by $N/2$, which we denote by $p_1(y)$:

$$p_1(y) = \frac{1}{N}\exp\left\{-\frac{y_i}{N}\right\}.$$

The complete denominator PDF is

Denominator PDF:

$$\log p(\mathbf{y} \mid H_0) = \log p_0(y_0) + \sum_{k=1}^{N/2}\log p_1(y_k) + \log p_0(y_{N/2}).$$

2. Stage 2: Inverse DFT

In the second step, let $\mathbf{r} = [r_0, r_1, \ldots, r_P]$ be the first $P+1$ ACF lags, which can be computed from $1/N$ times the first $P+1$ samples of the real part of the inverse DFT of $\mathbf{y}$. This may be written as

$$r_k = \frac{1}{N^2}\sum_{i=0}^{N/2}\epsilon_i\, y_i \cos\left\{\frac{2\pi ik}{N}\right\}, \qquad k = 0,1,\ldots,P$$

where $\epsilon_i = 1$ for $i = 0, N/2$, and $\epsilon_i = 2$ for $i = 1,2,\ldots,N/2-1$. This may be written in the matrix form

Feature Calculation: $\mathbf{r} = \mathbf{C}'\mathbf{y}$

where matrix $\mathbf{C}$ is defined accordingly.

Now, for the first time, we use a reference hypothesis other than independent Gaussian noise. In fact, we use a floating reference hypothesis—one that depends upon the data sample. The use of a floating reference hypothesis and the constraints on how it may vary are discussed elsewhere [1]. The floating reference hypothesis is the AR spectrum corresponding to the ACF $\mathbf{r}$. Using the Levinson-Durbin recursion [21], we may transform $\mathbf{r}$ into the AR coefficients $\{a_1, a_2, \ldots, a_P, \sigma^2\}$. The corresponding AR spectrum is written

$$y_k^r = \sigma^2\left|\sum_{k=0}^{P} a_k \exp\left\{-\frac{j2\pi ik}{N}\right\}\right|^2$$

where the superscript "r" is a reminder that the AR spectrum depends on $\mathbf{r}$. We let our reference hypothesis, denoted by $H_0(\mathbf{r})$, be that the mean of $\mathbf{y}$ equals the AR spectrum $\mathbf{y}^r = [y_0^r, y_1^r, \ldots, y_{N/2}^r]$. For simplicity, we assume the elements of $\mathbf{y}$ are independent.

$\mathbf{H}_0(\mathbf{r})$:	That $\mathbf{y}$ has independent elements with mean $\mathrm{E}(\mathbf{y}) = \mathbf{y}^r$.

Under $\mathbf{H}_0(\mathbf{r})$, the elements of $\mathbf{y}$ are independent and Chi-squared with 1 or 2 degrees of freedom with mean y_k^r. Bins $k = 0, N/2$ are distributed according to

$$p_0(y, y^r) = \frac{1}{y^r\sqrt{2\pi}}(y_i/y^r)^{-1/2}\exp\left\{-\frac{y_i}{2y^r}\right\}$$

while bins 1 through $N/2-1$ are exponentially distributed according to

$$p_1(y, y^r) = \frac{1}{y^r}\exp\left\{-\frac{y_i}{y^r}\right\}.$$

In summary,

Numerator PDF:

$$\log p(\mathbf{y} \mid H_0(\mathbf{r})) = \log p_0(y_0, y_0^r) + \sum_{k=1}^{N/2}\log p_1(y_k, y_k^r) + \log p_0(y_{N/2}, y_{N/2}^r).$$

We may use the central limit theorem (CLT) to approximate the PDF of $\mathbf{r}$ under $H_0(\mathbf{r})$ because the mean of $\mathbf{r}$ under $H_0(\mathbf{r})$ is very nearly $\mathbf{r}$ itself. Under $H_0(\mathbf{r})$, the elements of $\mathbf{y}$ are independent with mean $\mathbf{y}^r$ and diagonal covariance $\mathbf{\Sigma}_y^r$ given by

$$\mathbf{\Sigma}_y^r(i,i) \triangleq \mathcal{E}((y_i - y_i^r)^2 \mid H_0(\mathbf{r})) = \begin{cases} 2(y_i^r)^2, & i = 0, N/2 \\ (y_i^r)^2, & 1 \le i \le N/2-1. \end{cases}$$

Under $H_0(\mathbf{r})$, $\mathbf{r}$ has mean

$$\mathbf{r}^r \triangleq \mathrm{E}(\mathbf{r} \mid H_0(\mathbf{r})) = \mathbf{C}'\mathbf{y}^r$$

and covariance

$$\Sigma_r^r = \mathbf{C}'\Sigma_y^r\mathbf{C}.$$

$$\log p(\mathbf{r} \mid H_0(\mathbf{r})) = -\frac{(P+1)}{2}\log(2\pi) - \tfrac{1}{2}\log|\det(\Sigma_r^r)| - \tfrac{1}{2}(\mathbf{r}-\mathbf{r}^r)'(\Sigma_r^r)^{-1}(\mathbf{r}-\mathbf{r}^r). \quad (10)$$

If we we make the approximation $\mathbf{r}^r \simeq \mathbf{r}$, we obtain

Denominator PDF:

$$\log p(\mathbf{r} \mid H_0(\mathbf{r})) = -\frac{(P+1)}{2}\log(2\pi) - \tfrac{1}{2}\log|\det(\Sigma_r^r)|.$$

3. Stage 3: Conversion to RCs

The conversion from ACF to RCs is an invertible transformation that is characterized by a Jacobian matrix. The determinant of this matrix is the J-function of the transformation.

Feature Calculation: $\mathbf{r} \rightarrow$ (Levinson recursion for reflection coefficients) $\rightarrow \mathbf{k}$

where $\mathbf{r}$ is the ACF vector, $\mathbf{r} \triangleq [r_0, r_1, \ldots, r_P]$, and $\mathbf{z}$ is the vector of reflection coefficients augmented by the variance (zero-th lag ACF sample),

$$\mathbf{k} \triangleq [r_0, k_1, \ldots, k_P].$$

Note that we use r_0 and not the AR prediction error variance σ_0^2. This transformation is invertible and is characterized by the Jacobian

J-function (Jacobian):

$$\log \mathrm{J} = -P\log(r_0) + \sum_{i=1}^{P-1}(P-i)\log(1-k_i^2).$$

4. Stage 4: Log-Bilinear Transformation

Although the RCs have desirable properties as features, they are subject to the limit $|k_i| < 1$ which produces a discontinuity in the PDF. As a result, the PDF can be difficult to estimate using so-called non-parametric PDF estimators such as Gaussian mixtures. To obtain more Gaussian behavior, the log-bilinear transformation is recommended (thanks to S. Kay for recommending this).

Feature Calculation: $k_i' = \dfrac{\log(1-k_i)}{\log(1+k_i)}$, $1 \le i \le P$, $r_0' = \log(r_0)$.

This transformation is invertible and is characterized by the Jacobian

J-function (Jacobian):

$$\log \mathrm{J} = r_0' - \sum_{i=1}^{P}\log\left(\frac{2}{1-k_i^2}\right).$$

VII. EXPERIMENTAL VALIDATION

A very important question in developing a class-specific classifier is how to validate the analysis of a feature transformation. Because the numerator and denominator PDFs of the J-function are often evaluated in the far tails, we can never know if these PDF values are correct by histogram techniques. In Section VIC and VID (up to stage 2), two methods are presented for calculating the J-function for ACF samples. It may be verified that for contiguous ACF samples, the two approaches produce exactly the same features. The J-function values produced by the two methods are very close, but not exactly the same. Such comparisons are reassuring but are not a complete test and cannot be made for all problems. The following approach is a complete end-to-end test that has proved to be very useful.

Validation of the feature modules amounts to validating the PDF projection theorem (3). To validate (3), we design a hypothesis H_v for which we know the PDF $p(\mathbf{x} \mid H_v)$ exactly and for which we can create a large amount of synthetic raw data samples. We convert the synthetic data to features which we use to obtain the PDF estimate $\hat{p}(\mathbf{z} \mid H_v)$. Using this estimate in (3), we obtain an estimate $p_p(\mathbf{x} \mid H_v)$. To validate the result, we plot the projected PDF values $p_p(\mathbf{x} \mid H_v)$ on one axis and the exact values $p(\mathbf{x} \mid H_v)$ on the other axis for each sample of synthetic data. The points should lie near the $y = x$ line. An example is shown in Fig. 10 where we tested the chain of four feature modules in Section VID. The synthetic data used in the experiment were 100 time-series of independent Gaussian noise of variance 100 and length 4096. The features were computed using an AR model order of 4 with segmentation to 64-sample segments, thus producing 64 independent feature vectors of dimension 5 per sample. A Gaussian mixture model was used to statistically model the features.

VIII. CLASS-SPECIFIC TIME-SERIES CLASSIFIER USING REFLECTION COEFFICIENTS AND HMM

We can put to use the material thus-far discussed to arrive at a fully modular, extremely versatile class-specific classifier. A functional block-diagram of this classifier is provided in Fig. 11. A given time-series is processed by each class-model to arrive at a raw-data log-likelihood for the class. Each block labeled "RC(P)" computes the reflection coefficients of order P from the associated time-series segment.

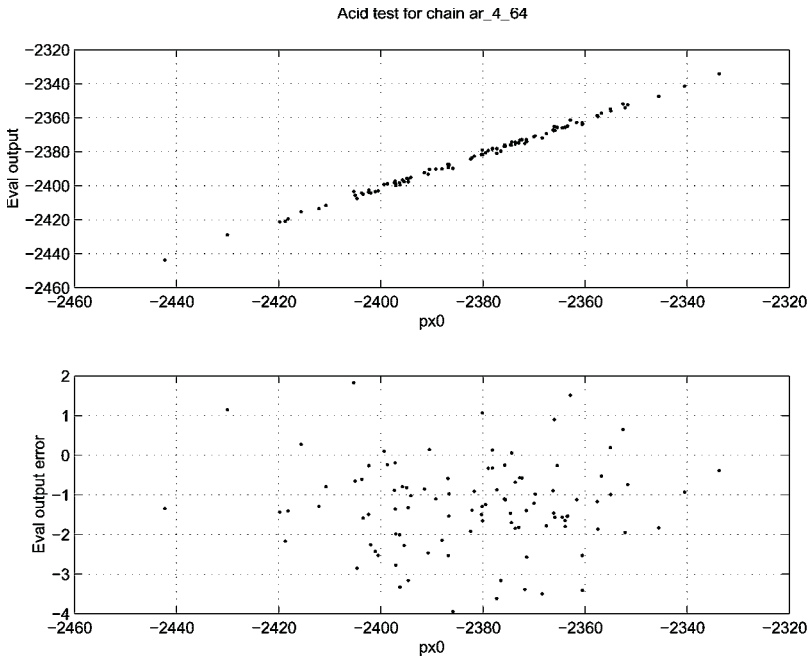


Fig. 10. Example of validation test results for 4th order autoregressive features (Section VID stages 1–4). Upper graph shows theoretical log-PDF values on x-axis and PDF projection theorem values on y-axis for 100 synthetic events. Lower graph shows the errors.

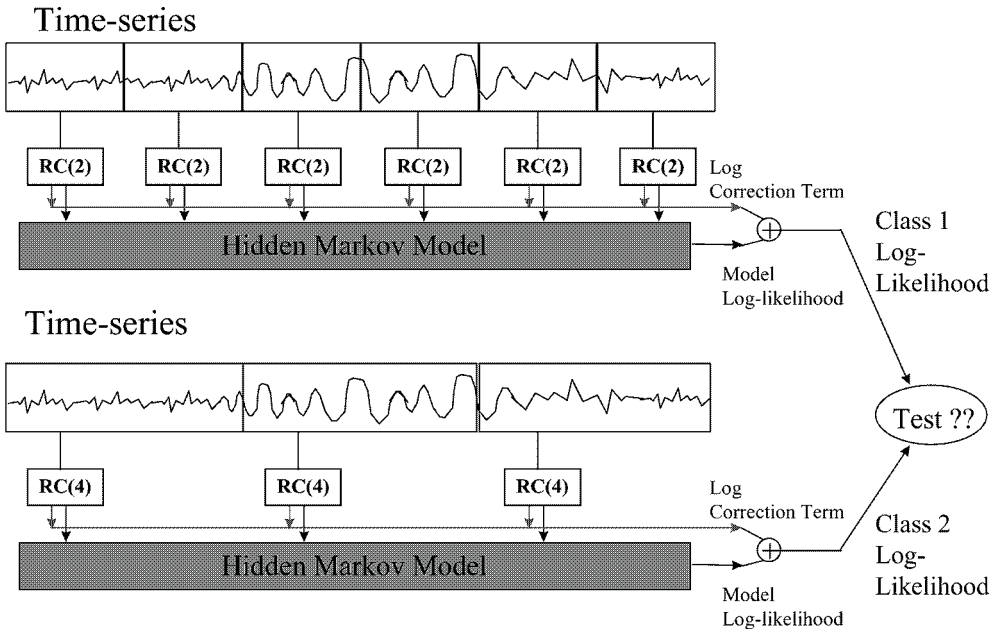


Fig. 11. Block diagram of an HMM and RC-based class-specific classifier. A given time-series is processed by each class-model to arrive at a raw-data log-likelihood for the class. Each block labeled "RC(P)" computes the *P*th order reflection coefficients from the corresponding time-series segment and is implemented by a series of modules (see text).

The figure shows two class-models employing different segmentation lengths as well as different model orders. The log-correction terms of all the segments are added together and the aggregate correction term is added to the HMM log-likelihood (from the forward procedure [23]) to arrive at the final raw data log-likelihood for the class. The segmentation sizes and model orders are optimized for each class individually, eliminating the need to "compromise."

Each "RC(P)" block is composed of a series of modules implementing ACF calculation followed by conversion to RCs, and ending with feature conditioning by the log-bilinear transformation. This may be implemented by the three modules described in Sections VIC, VID3, and VID4. Alternatively, the

four modules of Sections VID1, VID2, VID3, and VID4 will produce virtually identical features and J-function values. This classifier has the added benefit that the models may be validated by re-synthesis of time-series from features (either computed from actual data or generated at random by the HMM).

It should be stressed that we are not limited to using RC features and HMM PDF models. As long as care is taken in computing the correction terms, any feature set and any statistical model may be employed. Straight DFT features may be preferable to RC features for sinusoidal signals. Wavelet features may be preferable for certain other types of signals. A particularly good set of features for DFT (or wavelet processing) is to save the largest M bins and residual energy. The correction term for this feature set has been worked out by Nuttall. [24]. Nuttall has also derived the correction term for features that may be written as a set of inter-dependent quadratic forms, [25].

IX. CONCLUSION

Previous to the class-specific method, practitioners in image or signal classification had no guidance from classical theory in dealing with complex problems. The incomplete theory forced practitioners to think of feature extraction from the point of view of class separability. This flawed paradigm led the practitioner down the slippery slope of high dimensionality. Now that the reader has been introduced to the fundamental concepts of classification theory using class-specific features, he or she has the tools necessary to attack classification problems one class at a time, capturing all the necessary information in the features and not being forced to "make-do" with features that are general enough for all classes, but not sufficient for any class. The examples provided are enough to build a simple, yet effective class-specific time-series classifier.

REFERENCES

[1] Baggenstoss, P. M. (2002)
The PDF projection theorem and the class-specific method.
IEEE Transactions on Signal Processing, **51**, 3 (Mar. 2003), 672–685.

[2] Duda and Hart
Pattern Classification and Scene Analysis.
Wiley, 1973.

[3] Stone, C. J. (1980)
Optimal rates of convergence for nonparametric estimators.
Annals of Statistics, **8**, 6 (1980), 1348–1360.

[4] Scott, D. W. (1992)
Multivariate Density Estimation.
Wiley, 1992.

[5] Balachander, T., and Kothari, R. (1999)
Oriented soft localized subspace classification.
International Conference on Audio, Speech, & Signal Processing, **2** (1999), 1017–1020.

[6] Frimpong-Ansah, Pearce, K., Holmes, D., and Dixon, W. (1989)
A stochastic/feature based recogniser and its training algorithm.
International Conference on Audio, Speech, & Signal Processing, **1** (1989), 401–404.

[7] Kumar, S., Ghosh, J., and Crawford, M. (1999)
A versatile framework for labeling imagery with large number of classes.
Proceedings of the International Joint Conference on Neural Networks, Washington, D.C., 1999, 2829–2833.

[8] Kumar, S., Ghosh, J., and Crawford, M. (2000)
A hierarchical multiclassifier system for hyperspectral data analysis.
In J. Kittler and F. Roli, (Eds.), *Multiple Classifier Systems*, Springer, 2000, 270–279.

[9] Watanabe, H., Yamaguchi, T., and Katagiri, S. (1997)
Discriminative metric design for robust pattern recognition.
IEEE Transactions on Signal Processing, **45**, 11 (1997), 2655–2661.

[10] Belhumeur, P., Hespanha, J., and Kriegman, D. (1997)
Eigenfaces versus Fisherfaces: recognition using class specific linear projection.
IEEE Transactions on Pattern & Machine Intelligence, **19** (July 1997), 711–720.

[11] Baggenstoss, P. M. (1998)
Class-specific feature sets in classification.
In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC)*, National Institute of Standards and Technology, 1998, 413–416.

[12] Baggenstoss, P. M. (1999)
Class-specific features in classification.
IEEE Transactions on Signal Processing, **47** (Dec. 1999), 3428–3432.

[13] Kay, S. (2000)
Sufficiency, classification, and the class-specific feature theorem.
IEEE Transactions on Information Theory, **46** (July 2000), 1654–1658.

[14] Baggenstoss, P. M. (2000)
A theoretically optimum approach to classification using class-specific features.
In *Proceedings of International Conference on Pattern Recognition*, Barcelona, 2000.

[15] Baggenstoss, P. M. (2001)
A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces.
IEEE Transactions on Speech and Audio, **9**, 4 (May 2001), 411–416.

[16] Baggenstoss, P. M. (2002)
The chain-rule processor: Optimal classification through signal processing.
In *Proceedings of International Conference on Pattern Recognition*, Quebec, Aug. 2002.

[17] Lehmann, E. H. (1959)
Testing Statistical Hypotheses.
New York: Wiley, 1959.

[18] Kay, S. M., Nuttall, A. H., and Baggenstoss, P. M. (2001)
Multidimensional probability density function approximation for detection, classification and model order selection.
IEEE Transactions on Signal Processing, **49** (Oct. 2001), 2240–2252.

[19] Nuttall, A. H., and Baggenstoss, P. M. (2002)
The joint distributions for two useful classes of statistics with applications to classification and hypothesis testing.
Submitted to *IEEE Transactions on Signal Processing*, 2002.

[20] Bishop, C. M., Svensen, M., and Williams, C. K. I. (1998)
GTM: The generative topographic mapping.
Neural Computation, **10**, 1 (1998), 215–234.

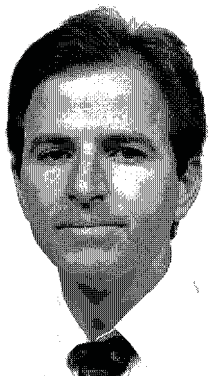
[21] Kay, S. (1998)
Modern Spectral Estimation: Theory and Applications.
Prentice Hall, 1988.

[22] Barndorff-Nielsen, O. E., and Cox, D. R. (1989)
Asymptotic Techniques for Use in Statistics.
Chapman and Hall, 1989.

[23] Rabiner, L. R. (1989)
A tutorial on hidden Markov models and selected applications in speech recognition.
Proceedings of the IEEE, **77** (Feb. 1989), 257–286.

[24] Nuttall, A. H. (2001)
Joint probability density function of selected order statistics and the sum of the remaining random variables.
NUWC Technical Report 11345, Oct. 2001.

[25] Nuttall, A. H. (2000)
Saddlepoint approximation and first-order correction term to the joint probability density function of M quadratic and linear forms in K Gaussian random variables with arbitrary means and covariances.
NUWC Technical Report 11262, Dec. 2000.



Paul M. Baggenstoss (S'82—M'90) received his Ph.D. in electrical engineering (statistical signal processing) at the University of Rhode Island (URI) in 1990.

Since then he has been applying signal processing and hypothesis testing (classification) to sonar problems. From 1979 to 1996, he was with Raytheon Co., Portsmouth, RI. He joined the Naval Undersea Warfare Center, Newport, RI, in 1996 where he is today.

He is the author of numerous conference and journal papers in the field of signal processing and classification and has taught part-time as an adjunct professor of Electrical Engineering at the University of Connecticut, Storrs. He is the recipient of the 2002 URI Excellence Award in Science and Technology.

"Statistics 101" for Multisensor, Multitarget Data Fusion

RONALD P. S. MAHLER
Lockheed Martin NE&SS Tactical Systems

This tutorial summarizes the motivations, concepts, techniques, and applications of finite-set statistics (FISST), a system-level, "top-down" direct generalization of ordinary single-sensor, single-target engineering statistics to the multisensor, multitarget realm. FISST provides powerful new conceptual and computational methods for dealing with multisensor, multitarget, and multi-evidence data fusion problems. The paper begins with a broad-brush overview of the basic concepts of FISST. It describes how conventional single-sensor, single-target formal Bayesian modeling is carefully extended to general data fusion problems. We examine a simple example: joint detection and tracking of a possibly non-existent maneuvering target in heavy clutter. The tutorial concludes with a commentary on certain criticisms of FISST.

Manuscript received December 20, 2002; revised June 23, 2003.

Refereeing of this manuscript was handled by P. K. Willett.

Author's current address: MS U2S26, 3333 Pilot Knob Road, Eagan MN 55121, E-mail: (ronald.p.mahler@lmco.com).

I. INTRODUCTION

Broadly speaking, data fusion is the process of directing the right data sources on the right platforms to the right targets at the right times, with the goal of detecting, localizing, identifying, and determining the threat potential of as many targets of interest as possible, whether these targets be individual entities such as tanks or jet fighters, or group entities such as infantry battalions or jet fighter sorties.

Progress in single-sensor, single-target data fusion (e.g., tracking) has been greatly facilitated by the existence of a systematic, mathematically rigorous, and yet practical engineering statistics that has supported the development of new concepts in the field. Like all engineering mathematics, engineering statistics is a tool and not an end in itself. It must have two qualities:

Trustworthiness: Constructed upon systematic, reliable mathematical foundations, to which we can appeal when the going gets rough.

Fire and forget: These foundations can be safely neglected in most applications, leaving a serviceable mathematical machinery in their place.

These two qualities are inherently in conflict. If foundations are so mathematically complex that they cannot be taken for granted in most engineering situations, then they are shackles and not foundations. But if they are so simple that they repeatedly lead us into engineering blunders, then they are simplistic and not simple!

The dividing line between the serviceable and the simplistic is what might be called the "Bar-Shalom test." Y. Bar-Shalom is probably the world's most well-known and influential researcher in data fusion applications. Quoting Einstein, he has often said: "Things should be as simple as possible—but no simpler!"

This is one of the defining characteristics of the "Statistics 101" concepts and techniques that most signal processing engineers learn as undergraduates: (1) random vectors, (2) probability-mass and probability-density functions, (3) differential and integral calculus, (4) statistical moments such as expected value and covariance, (5) optimal state estimators, (6) signal and signature modeling, and (7) optimal signal-processing filters such as the Kalman filter, etc.

These concepts and techniques are central to a particular R&D philosophy about how data fusion algorithms should be devised: formal Bayesian statistical modeling. Algorithms can be always cobbled together using catch-as-catch-can techniques—for example, by immediately discretizing a problem and applying simple, brute force computational methods. However, algorithm behavior may be difficult to diagnose because of hidden

assumptions and ad hoc design choices. Also, brute force approximation often leads to computational intractability, numerical instability, poor convergence, and other problems. In formal statistical modeling, one instead begins with a careful Bayes-statistical specification (a model) of the problem. Then, from it, one derives a mathematically optimal solution and theoretically principled approximations of this solution. Algorithm behavior is usually more explicable because assumptions and important design decisions in both the model and the approximations have been carefully parsed into a systematic, disciplined chain of reasoning.

Given the importance of this engineering statistics in the single-sensor, single-target realm, one might have expected that multisensor, multitarget data fusion would already rest upon a similarly systematic, rigorous, and yet practical statistical foundation. Surprisingly, until recently this has not been the case. The major reason is that multisensor, multitarget systems introduce a major complication absent from single-sensor, single-target problems. Such systems are comprised of randomly varying numbers of randomly varying objects of various kinds: randomly varying collections of targets, randomly varying collections of sensors and sensor-carrying platforms, and randomly varying observation-scans collected by those sensors. A rigorous mathematical foundation for stochastic multi-object problems—point process theory [5, 30]—has been in existence for decades. Unfortunately, this theory has traditionally been formulated with the requirements of mathematicians rather than engineers in mind.

In 1994 we introduced an "engineering friendly" version of point process theory called finite-set statistics (FISST) [7, 15, 18]. The purpose of this paper is to provide a high-level overview of FISST that requires minimal familiarity with concepts of probability. FISST is engineering-friendly in that it is geometric (i.e., treats multitarget systems as visualizable images); and directly generalizes the Bayes "Statistics 101" formalism that most signal processing engineers already understand—including formal Bayes-statistical modeling methods.

However, these methods do not generalize in a straightforward manner [15]. The following are examples of how multisensor-multitarget statistics differs from single-sensor, single-target statistics. The standard Bayes-optimal state estimators are not defined in general, and neither are such familiar concepts as expected value, least-squares optimization, and Shannon entropy. Other concepts, such as miss distance, require major reworking. Also, no explicit, general, and systematic techniques exist for modeling multisensor-multitarget problems and then transforming these models into Bayesian form. FISST specifically addresses such gaps.

FISST results in a systematic Bayesian unification of detection, classification, tracking, decision-making, sensor management, group-target processing, expert-systems theory, and performance evaluation in multi-platform, multi-source, multi-evidence, multi-target, multi-group problems. Highlights are:

1) multisource-multitarget information theory [7], resulting in a scientific basis for performance evaluation of multisensor-multitarget algorithms [33];

2) a unified, probabilistic foundation for many aspects of expert systems theory (fuzzy logic, the Dempster-Shafer theory, rule-based evidence, Bayesian statistics) [15, 18, 20, 29];

3) robust target identification, when underlying sources or sensors are imperfectly understood [8, 20, 29];

4) simultaneous optimal estimation of the numbers, identities, and geokinematics of targets [7, 15, 18];

5) a systematic approach for detection, tracking, and ID of multiple group targets [13];

6) a unified, control-theoretic approach to multisensor-multitarget sensor management [14, 16, 17, 19];

7) potentially powerful new approximation techniques such as multitarget statistical analogs of constant-gain Kalman filters [13, 21] or MHT approximations for sensor management [16, 17, 19].

FISST has attracted great interest since 1994. FISST-based algorithms are being or have been investigated under R&D contracts from U.S. Department of Defense agencies such as the Army Research Office, the Air Force Office of Scientific Research, SPAWAR Systems Center, the Missile Defense Agency, the Army Missile Research and Defense Command, and three different sites of the Air Force Research Laboratory. The Australian Defence Science and Technology Organisation is funding several activities in this area, and several research teams around the world are investigating FISST methods.

Applications at the applied-R&D level include:

1) scientific performance evaluation [33];

2) robust automatic target recognition using Synthetic Aperture Radar (SAR) data [8];

3) adaptive sensor and platform management [16, 17, 19];

4) multi-target detection and tracking in high-density environments [21];

5) robust passive-acoustic classification; and more.

We begin by providing a broad-brush overview of finite-set statistics. In two succeeding sections we summarize the specifics of the approach. A simple illustration—tracking a possibly non-existent maneuvering target in heavy clutter—is described and we conclude with comments regarding certain criticisms of FISST.

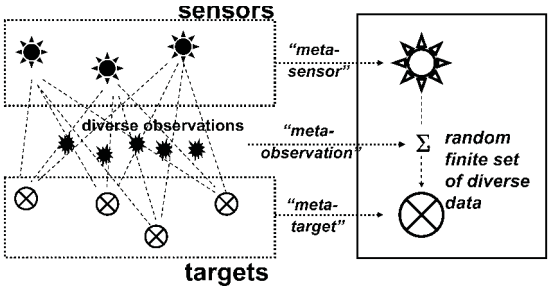


Fig. 1. Basic concept of finite-set statistics. Multisensor-multitarget problems are mathematically transformed into single-sensor, single-target problems by bundling all sensors into a single "meta-sensor" and all targets into a single "meta-target."

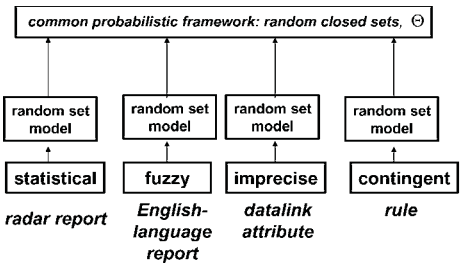


Fig. 2. Common representation of diverse data. Data of diverse types—radar, natural-language statements, features, rules from rulebases—are transformed into a common framework: the random subset.

II. FINITE-SET STATISTICS (FISST) IN A NUTSHELL

The basic concepts of FISST are summarized in Figs. 1–5. At left in Fig. 1, one or more sensors or other data sources of arbitrary types collect multiple observations from a group of targets. None of these targets have necessarily been detected yet.

The basic idea underlying FISST is to transform this multisource-multitarget problem into a mathematically equivalent single-sensor, single-target problem. All of the sensors are mathematically bundled into a single "meta-sensor" that retains all of the characteristics of the original sensors: their probabilities of detection, measurement models, and clutter or false alarm models. The targets are likewise bundled into a single "meta-target" that retains all of the characteristics of the individual targets.

A major barrier to data fusion has been the disparate forms that data can have. Data supplied by tracking radars can be accurately described in statistical form. But it is unclear how English-language evidence might be represented mathematically. Features (such as those transmitted on datalink) and rules from knowledge-bases exhibit varying degrees of ambiguity. As shown in Fig. 2, FISST deals with all such data by transforming it into a common mathematical framework—a random subset—that makes common processing possible.

Fig. 3 illustrates a specific, intuitive example of this. In the natural-language observation 'Gustav is

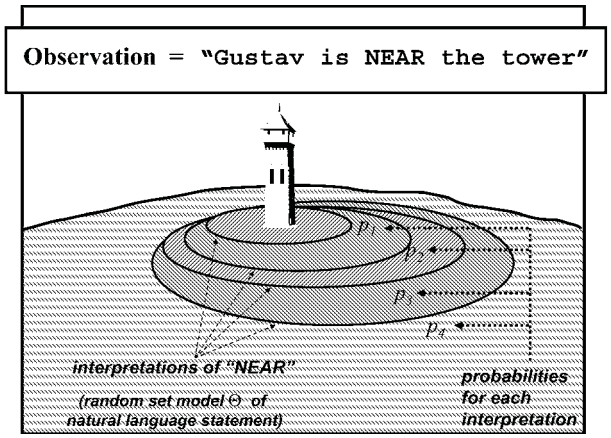


Fig. 3. Probabilistic representation of English language statement. Ambiguous observation 'NEAR the tower' is modeled as a series of ellipses surrounding the tower, each of which has a certain probability of being the correct interpretation of the concept 'NEAR'.

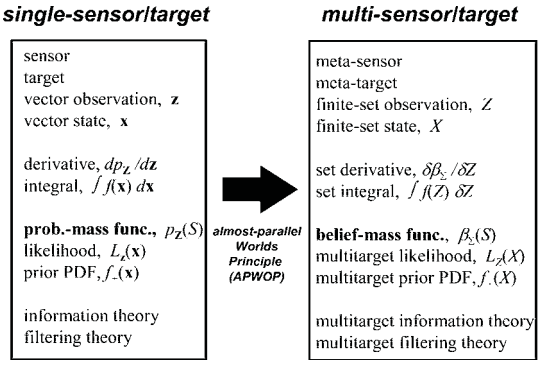


Fig. 4. Multisensor-multitarget statistics. Single-sensor, single-target statistics is directly generalized to multisensor-multitarget statistics. The "almost-parallel worlds principle" permits direct generalization of many single-sensor, single-target solution and approximation techniques.

NEAR the tower', the ambiguous concept 'NEAR' is modeled as a series of nested ellipses surrounding the tower. Each ellipse is assigned a subjective probability that it is the correct interpretation of 'NEAR'. The resulting randomly varying ellipse (a random subset of the plane) is a probabilistic model of 'NEAR'. One novel consequence of this approach is that many familiar expert-system methodologies (fuzzy logic, the Dempster-Shafer theory, rule-based evidence) can be unified under a single probabilistic umbrella. (This aspect of FISST is heavily indebted to random set methods for data fusion pioneered by I. R. Goodman and H. T. Nguyen [7].)

Fig. 4 illustrates the mathematical core of finite-set statistics. Single-sensor observations and single-target states are generalized to multisensor-multitarget observation-sets and multitarget state-sets. The integral and derivative of undergraduate calculus are generalized to multisensor-multitarget "set derivatives" and

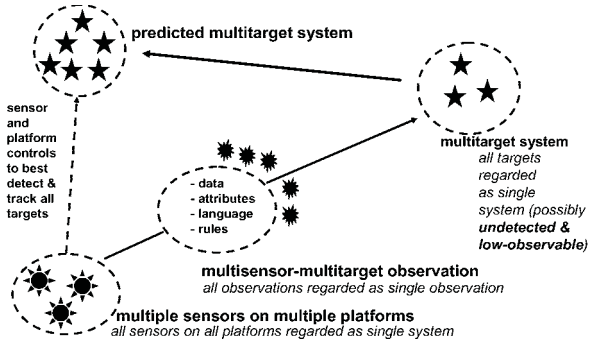


Fig. 5. Unified Bayesian multisource-multitarget data fusion. General data fusion problem, including sensor management, can be formulated as a system-level tracking-and-control problem involving arbitrary forms of evidence.

"set integrals." Single-sensor, single-target probability-mass functions and likelihood functions are generalized to multisensor-multitarget belief-mass functions and likelihood functions. And so on. The "almost-parallel worlds principle" states that multisensor-multitarget problems can be attacked through direct generalization of solution techniques for the analogous single-sensor, single-target problems. Using these techniques, one can devise principled new approximation strategies for applications such as cluster tracking, group target tracking, and sensor management [12, 16, 17, 19, 21].

Fig. 5 illustrates a consequence of this system-level perspective. The detection, tracking, and identification of multiple targets using multiple evidence types provided by multiple re-allocatable sources can be thought of as being, in effect, no different than a missile-tracking camera problem. Sensors must be redirected to maximize knowledge about the multitarget system.

III. SINGLE-SENSOR, SINGLE-TARGET STATISTICS

Bayes statistics has become the most widely accepted engineering mathematics for single-sensor, single-target target detection, tracking, identification, and data fusion applications. This is due in large part to the fact that it leads to provably optimal algorithms within an (often deceptively) simple mathematical framework. This simplicity is a great strength—but also a great weakness.

In recent years, the temptation has arisen of trying to appear deeply authoritative about any engineering R&D problem—and at the same time avoid careful thinking—by citing Bayes' rule, declaring victory, and then portraying complacency towards unexpected difficulties as a sign of intellectual superiority. However, in the words of the statisticians J. C. Naylor and A. F. M. Smith, "The implementation of Bayesian inference procedures can be made to appear deceptively simple" [25, p. 214]—and indeed has been. Also often forgotten is the fact that the optimality and the simplicity of Bayesian methods can be taken for granted only in standard applications addressed by standard textbooks.

In this section we summarize the elements of formal Bayesian statistical modeling: target states, sensor observations, the Bayes rule data-update, likelihood functions, motion updates, and Markov transition probabilities. We describe some basic modeling issues associated with real-time Bayesian approaches, and how they are resolved.

Target States and Sensor Measurements. Suppose that a single sensor collects data from a single, moving target. Thinking like a physicist, we first precisely model the physical "states" that our "particle"—the target—could be in. The state-model is typically a vector, such as $\mathbf{x} = (x, y, z, v_x, v_y, v_z, a_x, a_y, a_z, c)$, that captures pertinent target descriptors such as position x, y, z, velocity v_x, v_y, v_z, acceleration a_x, a_y, a_z, and target identity/label c. Second, we precisely model the observations the sensor collects. For example, the sensor may observe only target position in spherical coordinates, in which case observation-models will be vectors such as $\mathbf{z} = (r, \theta, \phi)$.

Bayesian Approach to Solving Single-Sensor, Single-Target Problems. The primary question to be answered is this: What state $\mathbf{x}$ did the target have to be in to best explain the generation of the observation $\mathbf{z}$, given everything else already known about the target? In Bayes statistics, one does not say definitively that any particular $\mathbf{x}$ is the correct state, either before or after the collection of $\mathbf{z}$. Instead, we can say only that before and after the collection there were probabilities $p_+(\mathbf{x})$ and $p(\mathbf{x} \mid \mathbf{z})$, respectively, that any given $\mathbf{x}$ is the correct state. Here $p_+(\mathbf{x})$ is the prior probability of $\mathbf{x}$, which encapsulates all of our knowledge about each $\mathbf{x}$ before the collection. The posterior probability of $\mathbf{x}$, $p(\mathbf{x} \mid \mathbf{z})$, encapsulates all of our knowledge afterwards. The sums $\int p_+(\mathbf{x})d\mathbf{x}$ and $\int p(\mathbf{x} \mid \mathbf{z})d\mathbf{x}$ of all the probabilities over all the states must equal one. This is because probability is a zero-sum game: some states cannot be highly probable unless all other states are highly improbable.

Using Bayes' Rule to Incorporate New Single-Sensor Data. To get the posterior probability we must "adjust" the values of the prior probability, increasing $p_+(\mathbf{x})$ if $\mathbf{z}$ favors $\mathbf{x}$ and decreasing $p_+(\mathbf{x})$ if otherwise. Stated differently, for any $\mathbf{z}$ we must multiply $p_+(\mathbf{x})$ by a factor $L_{\mathbf{z}}(\mathbf{x})$ that is large if $\mathbf{z}$ favors $\mathbf{x}$ and small if otherwise. The value $L_{\mathbf{z}}(\mathbf{x})$ is called the likelihood that a target with state $\mathbf{x}$ generated the observation $\mathbf{z}$. Posterior probability $p(\mathbf{x} \mid \mathbf{z})$ must sum to one but in general $L_{\mathbf{z}}(\mathbf{x})p_+(\mathbf{x})$ does not. So, we must divide $L_{\mathbf{z}}(\mathbf{x})p_+(\mathbf{x})$ by the sum $K = \int L_{\mathbf{z}}(\mathbf{x})p_+(\mathbf{x})d\mathbf{x}$. The resulting deceptively simple-looking formula

$$p(\mathbf{x} \mid \mathbf{z}) = K^{-1} L_{\mathbf{z}}(\mathbf{x}) p_+(\mathbf{x}) \tag{1}$$

for the posterior probability is called Bayes' rule.

Estimating the Target State. The posterior probability $p(\mathbf{x} \mid \mathbf{z})$ encapsulates everything that we know about the target state, based on current evidence. It is not useful to us as engineers unless we have a "mathematical can opener" that allows us to extract the information that we really want: the position, velocity, identity, etc. of the target. One method is to choose the most probable state—i.e., find the $\mathbf{x}$ that makes $p(\mathbf{x} \mid \mathbf{z})$ largest. This procedure is an example of a Bayes-optimal state estimator—i.e., one that minimizes the central objective function of the Bayesian approach, the Bayes risk [31, pp. 54–59].

Engineering Issues, I. As engineers, we should be very troubled at this point. We have solved our problem by conjuring up a seemingly magical quantity, the likelihood $L_{\mathbf{z}}(\mathbf{x})$. Theoretically speaking, $L_{\mathbf{z}}(\mathbf{x}) = p(\mathbf{z} \mid \mathbf{x})$ is a so-called "conditional probability." But this bare fact merely transforms magic into equally unhelpful "mathe-magic." The real questions that must be answered are these: What explicit, general procedure might allow us to derive a concrete formula for $L_{\mathbf{z}}(\mathbf{x})$ in any specific situation? How do we know that this formula is "true"—i.e., faithfully reflects the actual behavior of the sensor? If the likelihood is not true, then any claim of "optimality" is hollow because it applies only to whatever sensor is actually modeled by the incorrect likelihood. In particular, if $L_{\mathbf{z}}(\mathbf{x})$ is too imperfectly modeled then an algorithm will "waste" data trying (and perhaps failing) to overcome the mismatch with reality.

More generally, citing Bayes' rule and declaring victory only passes the buck to the data simulation community and dodges the real algorithm-design issue: what to do when $L_{\mathbf{z}}(\mathbf{x})$ cannot be well-characterized. For example, it is unclear that sufficiently high-fidelity likelihoods $L_{\mathbf{z}}(\mathbf{x})$ can ever be implemented in real time for certain data sources such as High Range Resolution Radar (HRRR) and Synthetic Aperture Radar (SAR). In the case of other information sources—English language statements, rules, attributes—it is unclear how to mathematically represent them as "data" $\mathbf{z}$, let alone how one might construct $L_{\mathbf{z}}(\mathbf{x})$ given that fact.

Another difficulty arises from the fact that a state estimator must be selected with care. If it has unrecognized inefficiencies, then data will be unnecessarily "wasted" in trying to overcome them—though not necessarily with success. We do not have a Bayes-optimal solution unless we have a Bayes-optimal state estimator. It should have other desirable properties, e.g. rapid and stable convergence to the actual target state.

Sensor Measurement Models. How do we construct the true likelihood $L_{\mathbf{z}}(\mathbf{x})$? Usual engineering practice is to begin with an equation such as $\mathbf{z} = (r, \theta, \phi) = h(\mathbf{x})$, which states that the observation $\mathbf{z} = h(\mathbf{x})$ will be collected if the target has state $\mathbf{x}$. The function $h(\mathbf{x})$ captures the fact that the sensor usually cannot observe the entire target state $\mathbf{x}$ but rather only some incomplete and/or transformed view of it. In addition, because of internal noise the sensor will actually collect not $\mathbf{z}$ but some random perturbation $\mathbf{z} + \Delta\mathbf{z}$ of $\mathbf{z}$. This leads to a measurement model

$$\mathbf{z} = h(\mathbf{x}) + \Delta\mathbf{z}.$$

Since $\Delta\mathbf{z}$ is random there is a probability $p_{\Delta\mathbf{z}}(\mathbf{w})$ that $\Delta\mathbf{z}$ will take any given value $\mathbf{w}$. Using undergraduate calculus, one can show that the true likelihood is:

$$L_{\mathbf{z}}(\mathbf{x}) = p_{\Delta\mathbf{z}}(\mathbf{z} - h(\mathbf{x})). \tag{2}$$

In practice, this formula can be looked up in a textbook and so we need never actually bother with its formal derivation. But, as we shall see, in the multitarget case no textbook yet exists that allows us such an easy escape from mathematics and careful thinking.

Accounting for Interim Target Motion. What if the target is moving? Let $p_0(\mathbf{x})$ be the probability that $\mathbf{x}$ is the correct target state, based on all evidence collected up to the previous data-collection time t_{k-1}. If the target were not in motion, $p_+(\mathbf{x})$ would equal $p_0(\mathbf{x})$. Because the target is moving we cannot actually know $p_+(\mathbf{x})$, so how do we get it? We assume that we know the probability $p_+(\mathbf{x} \mid \mathbf{x}_0)$ that the target will move to state $\mathbf{x}$ from state $\mathbf{x}_0$—the Markov transition probability. Since the target had a probability $p_0(\mathbf{x}_0)$ of being in state $\mathbf{x}_0$ previously, $p_+(\mathbf{x} \mid \mathbf{x}_0)p_0(\mathbf{x}_0)$ is the probability that the new state will be $\mathbf{x}$ given that possibility. Summing over all prior states $\mathbf{x}_0$, we get the total probability that the target will be in state $\mathbf{x}$ at the next data collection:

$$p_+(\mathbf{x}) = \int p_+(\mathbf{x} \mid \mathbf{x}_0)p_0(\mathbf{x}_0)d\mathbf{x}_0. \tag{3}$$

Engineering Issues, II. The more accurately that $p_+(\mathbf{x} \mid \mathbf{x}_0)$ models target motion, the more effectively Bayes' rule will do its job. Otherwise, a certain amount of data must be expended in overcoming poor motion-model selection. But where does $p_+(\mathbf{x} \mid \mathbf{x}_0)$ come from? How do we ensure that it faithfully reflects the motion of the target if we are lucky enough to have modeled it correctly? Once again, we appear to have sidestepped difficult issues by conjuring up a magical quantity $p_+(\mathbf{x} \mid \mathbf{x}_0)$. And once again, the fact that $p_+(\mathbf{x} \mid \mathbf{x}_0)$ is a conditional probability is unhelpful "mathe-magic."

Target Motion Models. The usual strategy for determining $p_+(\mathbf{x} \mid \mathbf{x}_0)$ is to first make a guess about the target's motion between data-collection times: straight-line motion (dead reckoning), 2 g horizontal turn, etc. This guess is expressed as an equation $\mathbf{x} = g(\mathbf{x}_0)$ which states that the target will have state $\mathbf{x}$ at the new collection time if it had state $\mathbf{x}_0$ at the old one. Since this equation is only a guess, the actual $\mathbf{x}$ will usually be not $\mathbf{x}$ but rather some perturbation

$\mathbf{x} + \Delta\mathbf{x}$ of $\mathbf{x}$. Assuming that $\Delta\mathbf{x}$ is a random vector, we get a motion model:

$$\mathbf{x} = g(\mathbf{x}_0) + \Delta\mathbf{x}.$$

The procedure for constructing the true Markov transition probability from such a model is exactly analogous to that for constructing a true likelihood function from a sensor measurement model. Let $p_{\Delta\mathbf{x}}(\mathbf{y})$ be the probability that the perturbation $\Delta\mathbf{x}$ will have the value $\mathbf{y}$. Then

$$p_+(\mathbf{x} \mid \mathbf{x}_0) = p_{\Delta\mathbf{x}}(\mathbf{x} - g(\mathbf{x}_0)). \tag{4}$$

Again, in practice this formula can just be looked up in a textbook. And also again, in the multitarget case no textbook yet exists that offers us such a painless exit.

Generalized Kalman Filtering: Bayes Recursive Filter. Suppose that the sensor collects a time-sequence $\mathbf{z}_1, \ldots, \mathbf{z}_k$ of observations from the target. Let $p_{k|k}(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{z}_k)$ be the posterior probability at the time of the collection of observation $\mathbf{z}_k$. Each time we collect a new observation $\mathbf{z}_{k+1}$, use (2) to account for interim target motion. Apply (1) to incorporate the new observation, and then use a Bayes-optimal state estimator to extract from $p_{k+1|k+1}(\mathbf{x})$ the information that we want. Repeating this process recursively we end up with the foundation for single-sensor, single-target problems, the Bayes nonlinear filter [3, pp. 373–377].

The familiar Kalman filter is a special case of this general filter. It results when we assume that, for any data collection time, h in (1) and g in (2) are matrices, and that the perturbations $\Delta\mathbf{z}$ and $\Delta\mathbf{x}$ are independent Gaussian white noise.

Engineering Issues, III. As we shall see in our simple example, the Bayes filter is a very powerful tool for problems in which conventional approaches, such as the extended Kalman filter (EKF), experience difficulty. But it is also much more computationally demanding than the EKF and related techniques. Consequently, more powerful real-time approximate methods have been the subject of extensive recent research.

Caution is in order here because in some of these efforts simple, inherently intractable brute force techniques have been successively hyped and abandoned in favor of equally untenable techniques that are hyped in their own turn. In reality, naïve ad hoc approximations result in an algorithm "wasting" a certain amount of data overcoming—or failing to overcome—accumulation of approximation error, numerical instability, etc. Credible approaches typically rely on principled approximations based on sophisticated mathematical techniques.

An approximate Bayes filter should rapidly and stably converge to the correct answer regardless of the data collected. So-called particle-system filters [6] appear promising in this regard for those niche applications, such as low-SNR tracking [2], that defy conventional approaches. But the computational tractability of even these filters is being promoted with excessive enthusiasm by some.

IV. MULTISENSOR-MULTITARGET STATISTICS

Finite-set statistics directly generalizes the single-sensor, single-target statistics of previous sections into a serviceable engineering statistics for multisensor-multitarget problems. In this section we introduce the basic elements of multisensor-multitarget Bayes statistics: multitarget state-sets, multisensor observation-sets, multisensor-multitarget likelihoods, multitarget Markov transition probabilities, and so on. And we show how FISST directly generalizes formal Bayesian modeling methods to the multisensor-multitarget realm.

Multitarget States and Multisensor-Multitarget Observations. Any single-target system is completely described by its state-vector $\mathbf{x}$. So, if each state-vector has the form $\mathbf{x} = (x, y, z, v_x, v_y, v_z, a_x, a_y, a_z, c)$ then exactly ten parameters (nine real numbers and one discrete value c) are required to specify any state of the target exactly. A multitarget system is considerably more complicated. Its complete description requires a unified state representation: a finite set of the form $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ where n is the number of targets and $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are the state vectors of the individual targets. This description must include the possibility $n = 0$ —i.e., no target at all is present, in which case we write $X = \emptyset$. Such a unified representation accounts for the fact that n is variable and that targets have no physically inherent order. Thus $\{\mathbf{x}_1, \mathbf{x}_2\} = \{\mathbf{x}_2, \mathbf{x}_1\}$ is a single unified state-model of two targets with state-vectors $\mathbf{x}_1, \mathbf{x}_2$. (Physical states should not have redundant state-models: for example, the vectors $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_2, \mathbf{x}_1)$ model two distinct non-physical states, whereas $\{\mathbf{x}_1, \mathbf{x}_2\}$ does not.)

So, if each single-target state vector has ten parameters, $1 + 10 + 20 = 31$ parameters are required to describe a system with up to two targets; $1 + 10 + 20 + 30 = 61$ parameters are required for a system with up to three targets; and so on.

Suppose now that our multitarget system is observed by s sensors, each of which may collect a single datum from each target. In general, any given sensor may not actually "see" any given target during any given data collection. Indeed, it is possible that no sensor will "see" any targets at all. So, if there are n targets present then the sensors could collect a set of observations that contains anywhere from 0 to $s \cdot n$ observations. Worse, one or more of the sensors may collect false observations (false alarms). So in general, the total observation collected by many sensors from

many targets is some finite but arbitrarily large set Z of ordinary observations.

Bayesian Approach to Solving Multisensor, Multitarget Problems. The primary question that must be answered is this: Given everything previously known about the targets, what number had to be present, and what states did they have to have, to explain the fact that we collected the multisensor observation-set Z? As before, we can say only that before and after the collection of Z there is, respectively, a multitarget prior probability $p_+(X)$ and a multitarget posterior probability $p(X \mid Z)$ that any given X is the correct state-set. The sum $\int p_+(X)\delta X$ or $\int p(X \mid Z)\delta X$ of all the probabilities over all the multitarget states X must equal one. This requirement introduces a new complication: the indicated integrals—called set integrals—must sum over not only all possible target states but also over all possible numbers of targets. A more subtle issue is this: in a careful Bayesian formulation, $p(X \mid Z)$ and $p_+(X)$ must be single functions defined on the unified multitarget state X. It is not correct to partition them by target number into a family of functions such as $p_+(\emptyset), p_+(\mathbf{x}_1), p_+(\mathbf{x}_1, \mathbf{x}_2), \ldots$.

Using Bayes' Rule to Incorporate New Multisensor Data. We construct the multitarget posterior probability $p(X \mid Z)$ from the multitarget prior probability $p_+(X)$ as before. Multiply $p_+(X)$ by a factor $L_Z(X)$—the multisensor-multitarget likelihood—that is large if the multisensor data-scan Z favors the multitarget state-set X and is small otherwise. The multisensor-multitarget version of Bayes' rule is:

$$p(X \mid Z) = K^{-1} L_Z(X) p_+(X) \tag{5}$$

where $K = \int L_Z(X) p_+(X) \delta X$ and where the integral is a set integral that sums over all possible numbers of targets.

Engineering Issues, IV. As engineers we should be even more troubled than before because the quantity $L_Z(X)$ is even more mathe-magical than before. The hard questions are unchanged: What general, explicit procedures allow us to derive concrete formulas for $L_Z(X)$? How do we ensure that these formulas are true—i.e., faithfully reflect the behaviors of the actual sensors? How do we know that they are not ad hoc contrivances or not erroneously constructed? If we shirk such issues we fail to grasp that there is a problem—any boast of "optimality" is hollow if $L_Z(X)$ models the wrong sensors. Alternatively, we could—as some have indeed done—play a shell game: purport that the likelihood is the correct one and stonewall those who want to see proof.

An unexpected difficulty arises when we try to extract the information we really want from the multitarget posterior probability $p(X \mid Z)$: the number, positions, velocities, and types of the targets. Unfortunately, the naïve generalizations of the standard single-target Bayes-optimal estimators do not exist in general [15, pp. 40–42].

For example, consider the naïve generalization of the single-target Bayes-optimal estimator described earlier: choose the X that makes $p(X \mid Z)$ largest. To keep things simple, suppose that targets are in the 1-D interval $[0,2]$ and distance is measured in meters. Assume that $p(X \mid Z)$ has the following simple form: $p(X \mid Z) = 0.5$ if $X = \emptyset$, $p(X \mid Z) = 0.25\ \mathrm{m}^{-1}$ for any $X = \{x\}$, and $p(X \mid Z) = 0$ otherwise. That is: according to current evidence, there is a 50-50 chance that no target exists and, if otherwise, it is a single target that is equally likely to be anywhere in $[0,2]$. Since $p(X \mid Z)$ has its largest value at $X = \emptyset$, the naïve estimator leads us to the conclusion that no target is present since $0.5 > 0.25$. However, change units of measurement from meters to kilometers. Then $p(X \mid Z) = 250\ \mathrm{km}^{-1}$ if $X = \{x\}$ and we now conclude that a target is present! The paradox arises because the naïve estimator prescribes an impossible procedure: comparing a unitless quantity $p(X \mid Z)$ (when $X = \emptyset$) to a quantity $p(X \mid Z)$ with units (when $X = \{x\}$).

Consequently, new estimators must be devised and shown to be Bayes-optimal, convergent, etc. [15, pp. 42–44]. One novel feature should be pointed out. These new estimators optimally unify into a single procedure two conflicting processes that are normally accomplished separately: target detection (determining whether or not targets exist and to what number) and target estimation (determining the states of the targets, if they exist).

Modeling the Multisensor Suite in a Multitarget Scenario. Finite-set statistics provides explicit, general, systematic tools for constructing $L_Z(X)$. A detailed discussion can be found in [15, pp. 33–34]. In analogy with the single-sensor, single-target case, we begin with a multisensor-multitarget measurement model [15, pp. 17–20]. For a single sensor it has the form

$$\overset{\text{all measurements}}{Z} = \overset{\text{target-generated measurements}}{h_1(X)} \cup \overset{\text{non-target generated measurements}}{\Delta Z_1} \tag{6}$$

Here, $h_1(X)$ models observations directly generated by targets, but taking missed detections into account; whereas ΔZ_1 models observations that are not target-generated (e.g., false alarms, clutter, Electronic Countermeasures, etc). The symbol '$\cup$' indicates merely that the set Z of all observations consists of both target-generated observations and non-target-generated observations.

Given a measurement model we must construct the true likelihood function. This is accomplished using the FISST multitarget integral and differential calculus—specifically, the inverse operation of the set integral known as the set derivative. This calculus strongly resembles ordinary undergraduate differential

and integral calculus, including the existence of "turn-the-crank" rules. A detailed discussion can be found in [15, pp. 27–32].

Accounting for Interim Multitarget Motion. In the multitarget case one must account not only for the fact that targets are moving, but also for the fact that their number can change. Targets enter or leave the scene, some are destroyed. Others—like missiles—spawn new targets. Imitating the single-sensor, single-target case, let $p_0(X)$ be the multitarget posterior probability at the previous data-collection time t_{k-1}. We assume that we know the multitarget Markov transition probability $p_+(X \mid X_0)$ that the targets will have state-set X if they originally had state-set X_0. The total probability that the targets will have state-set X at the next time-instant t_k is

$$p_+(X) = \int p_+(X \mid X_0) p_0(X_0) \delta X_0. \tag{7}$$

Because one must account for possible changes in target number, the indicated integral is a set integral.

Engineering Issues, V. Where does the mathe-magical quantity $p_+(X \mid X_0)$ come from? How do we ensure that it faithfully reflects the motions of the targets—including target appearances and disappearances—if we happen to have accurate information about such things? The naïve choice—$p_+(\mathbf{y}_1, \ldots, \mathbf{y}_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n) = p_+(\mathbf{y}_1 \mid \mathbf{x}_1) \cdots p_+(\mathbf{y}_n \mid \mathbf{x}_n)$—presumes that no targets appear or disappear and that target motions are independent. But multitarget filters based on such presumptions may perform poorly against dynamic multitarget environments, for the same reason that single-target trackers that assume straight-line motion may perform poorly against maneuvering targets.

Multitarget Motion Models. Finite-set statistics allows us to construct the true $p_+(X \mid X_0)$ by generalizing usual engineering practice. We first specify a multitarget motion model [15, pp. 21–23]:

$$\overset{\text{all targets}}{X} = \overset{\substack{\text{pre-existing targets} \\ \text{(including target} \\ \text{disappearance)}}}{g(X_0)} \cup \overset{\substack{\text{newly} \\ \text{appearing} \\ \text{targets}}}{\Delta X} \tag{8}$$

Here, $g(X_0)$ describes the current states of all targets that previously existed, but taking into account the probability that any given target may disappear. Also, ΔX describes the generation of new targets in the scene. The symbol '$\cup$' indicates that the total set of targets consists of both persisting targets and newly appearing ones.

The multitarget Markov transition probability $p_+(X \mid X_0)$ can be constructed from the multitarget motion model in the same way that the multisensor-multitarget likelihood function is constructed from the multisensor-multitarget measurement model—that is, via the set derivative [15, pp. 35–36].

Multisensor-Multitarget Bayes Filter. Suppose that the sensors collect a time-sequence $Z_1, \ldots, Z_k$ of observation-sets from the targets. Let $p_{k|k}(X) = p(X \mid Z_k)$ be the posterior probability at the time of the collection of observation-set Z_k. Each time that we collect a new observation Z_{k+1}, use (7) to account for interim multitarget motion, apply (5) to incorporate the new observations collected by the sensors, and then use a Bayes-optimal multitarget state estimator to extract from $p_{k+1|k+1}(X)$ the information we want: target number, target position, etc. Recursively repeating this process results in the multisensor-multitarget Bayes nonlinear filter—the foundation for multisensor-multitarget applications.

Engineering Issues, VI. The multisensor-multitarget Bayes filter is far more computationally challenging than its single-sensor, single-target special case and so even more powerful approximation strategies are required. Caution is again called for because, once again, simple brute force techniques have been successively hyped as "powerful and robust computational methods," only to be quietly abandoned in turn. Multitarget particle-system techniques appear promising [2, 9] but will likely be computationally tractable only for a handful of targets in those tracking environments where they are appropriate—i.e., where conventional multitarget tracking methods fail. Using FISST techniques, we have derived alternative, lower-fidelity approximation filtering strategies based on the concept of a multitarget first-order moment. Roughly speaking, such approaches are multitarget analogs of alpha-beta-gamma filters [13, 21].

A Short History of Bayes Multitarget Filtering. The general Bayes multitarget filter is a relatively new concept. The earliest research appears to be the sophisticated "Jump Diffusion" approach of Miller, Srivastava, Lanterman, et al. [11]. As Lanterman has noted, Jump Diffusion and FISST are complementary efforts and the former "…may provide a way to exploit the complicated multitarget posteriors arising from FISST formulations" [10]. Portenko et al. have used branching-process concepts to model target appearance and disappearance [26]. Kastella's "JMP [joint multitarget probabilities], and the conceptual apparatus surrounding it, are elements of…finite-set statistics (FISST)" [23, p. 27]. The "likelihood function approach" of Stone et al. has fundamental limitations [15, pp. 42, 91–93]. Challa et al. have shown that the IPDA tracking approach [24] arises directly from the FISST formal modeling methodology [4] and have established connections with its multitarget extension, JIPDA [22]. FISST techniques are being investigated and applied by Sidenbladh et al. [27, 28] and Vo, Doucet et al. [32].

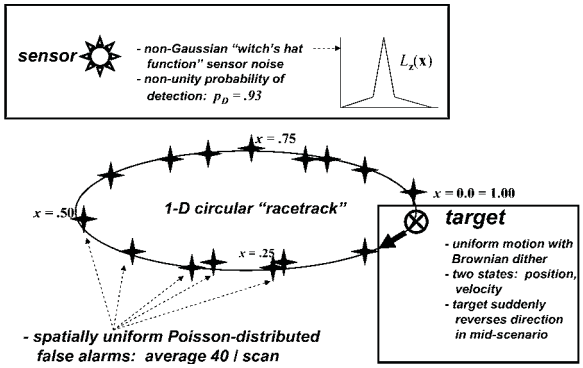


Fig. 6. Simple 1-D example. Recursive Bayes filter must detect and track a target moving around a 1-D racetrack, using a non-Gaussian sensor whose observations are corrupted by missed detections and false alarms.

V. A SIMPLE EXAMPLE

We illustrate some of the concepts described in this article using the simplest case of Bayes filtering with unknown target number: when target number can be zero or one. (More complex examples are too complicated to describe here but can be found elsewhere [21, 32].) The problem is depicted in Fig. 6. A particle moves around a circular "racetrack" of length one. Its motion is uniform but dithered by Brownian-motion perturbations. Its state is (x, v), where x is position and v is velocity. At mid-scenario (the time of the 250th observation-collection), the target abruptly reverses direction.

The target is interrogated by a position-observing sensor whose likelihood function is a non-Gaussian "witches hat" function (see Fig. 6). During each scan, the sensor collects an observation from the target 93% of the time, as well as an average of 40 false alarms. The false alarms are uniformly distributed spatially over the racetrack, and are Poisson-distributed in time.

Target number is assumed constant but unknown, so that the unified target state has the form $X = \emptyset$ or $X = \{(x, v)\}$. A measurement model is constructed from this information and the true likelihood function $f(Z \mid X)$ constructed from it using FISST techniques. The assumed motion model for the target is dead-reckoning: $g(x, v) = (x + \Delta t \cdot v, v)$. Fig. 7 shows a time-sequence of observations. The horizontal axis is observation number and the vertical axis is position on the racetrack. The target-generated observations are essentially invisible to the human eye. Nevertheless, the Bayes filter must determine if a target is present, find it, and then track it. (It is not provided with an estimate of initial target position, or any indication that the target actually exists.) After the abrupt maneuver at mid-scenario, the filter must re-acquire and re-lock the target. The filter is therefore a special case of the IPDA filter [4, 24] mentioned earlier.

Fig. 8 shows the target trajectory (solid line) and position estimates produced by the Bayes filter (dots). The filter finds and locks onto the target after about twenty scans, though becoming momentarily confused. After mid-scenario, about twenty observations are required before the filter detects the abrupt maneuver and another twenty before it re-acquires and re-locks. (The delays arise from the dead-reckoning motion model.)

The computational technique used in this particular example is a "spectral compression" filter of our devising that, like particle-system filters, has certain guaranteed-convergence properties. It has been described, briefly, only once in the open literature [1, pp. 212–213]. This is because, mindful of the rather blatant overselling prevalent in certain quarters, we wanted to first compare it to particle-system and other computational filters implemented by collaborators under our funding [2]. We have since found that particle-system filters are faster.

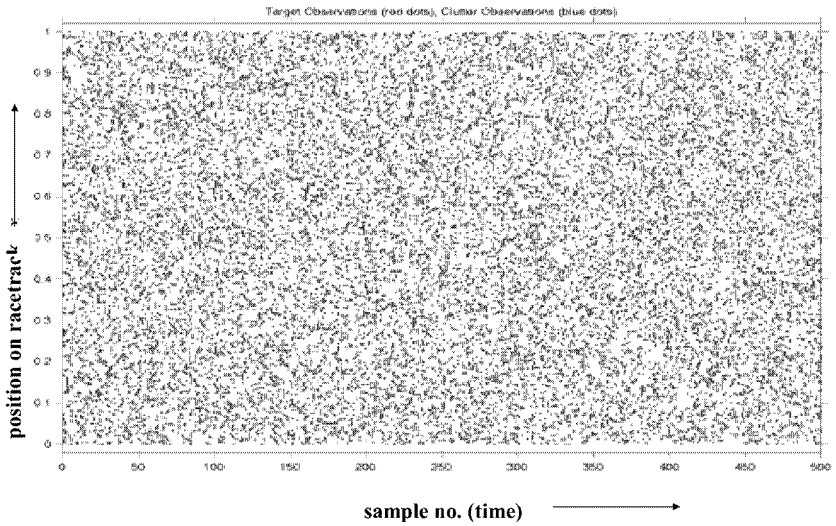


Fig. 7. Input data to recursive Bayes filter. Horizontal axis is time (observation-number) and vertical axis is target position on racetrack. Average of 40 false alarms per scan make observations of target's position nearly invisible to the human eye.

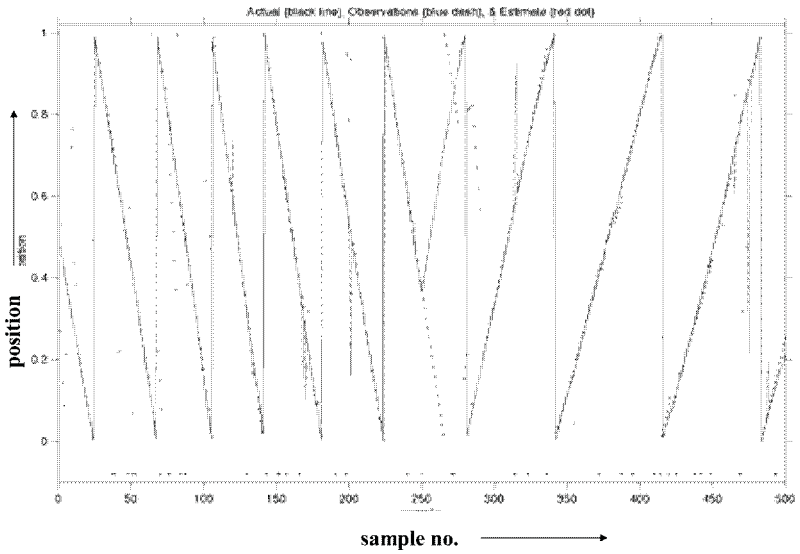


Fig. 8. Output of recursive Bayes filter. Solid line is the target trajectory while dots are the Bayes filter's estimates of target position. The filter successfully finds and tracks the target, and reacquires it after the sudden maneuver at mid-scenario.

VI. CRITICISMS OF FISST

A handful of partisans have claimed that a so-called "plain-vanilla Bayesian approach" suffices as down-to-earth, general "first principles" for Bayes multitarget filtering. Ergo, FISST is unnecessary or worse. However, our guide here should be the "Bar-Shalom test" cited earlier: "first principles" that lead to repeated blunders are simplistic, not simple.

The "plain-vanilla Bayesian approach" is so heedlessly formulated that it is not even Bayesian—and moreover, disparages random set concepts even while unwittingly assuming them! For example, one such partisan has: 1) boasted that his "plain-vanilla" approach addresses multitarget Bayes filtering with all necessary rigor and generality while being "straightforward"; 2) asserted that, therefore, FISST is pointless "obfuscation"; 3) subsequently introduced a complicated ur-theory for Bayes multitarget filtering; 4) failed to notice that this should be unnecessary since—as per his boast—the "plain-vanilla" approach already covers all of the bases; and 5) failed to comprehend that his ur-theory just unwittingly re-invents basic random set concepts in highly "obfuscated" notation! How can this be said to meet minimal standards of logic, let alone credibility?

Also, note the gamed yardsticks that are being applied. Anything more complicated than "plain-vanilla" is "obfuscation"—except when a "plain-vanilla" partisan is the obfuscator! Likewise: If FISST is illustrated using familiar observation models (e.g., post-detection reports), this proves that FISST is not "general." But explicit, general, rigorous methods for observation models (and many other things) prove only that FISST is not "simple"! Whereas the "plain-vanilla" approach is simple because it has no such methods—but yet, magically, is elastically all-subsuming! Similarly: A partisan avers that his approach addresses "the problem of search, track and identification, with the confounding issue that target count is unknown and must be estimated too"—whereas FISST addresses something else entirely: unified expert-systems theory. But FISST addresses both! When did misrepresentation and puffery come to suffice as "first principles"?

In asserting no significant difference between single-target and multitarget Bayes statistics, such partisans also fail to account for the actual, major differences—most seriously, by erroneously presuming that the naïve multitarget generalizations of the single-target Bayes-optimal state estimators exist. What is the credibility of "plain-vanilla Bayesian" when one of its central decision procedures is "not invariant under even a change of units"—especially given that these are the words used by one partisan in criticizing the same type of error in work that preceded his own?

The "plain-vanilla" stance essentially repudiates the formal statistical modeling standard that FISST directly extends to multisensor-multitarget problems. But "plain-vanilla" implementation has produced a succession of ad hoc, brute force algorithms afflicted by inherent—but less than candidly acknowledged—computational "logjams." When did "logjams" become exemplars of down-to-earth engineering practicality?

VII. SUMMARY

We have summarized the motivations, concepts, techniques, and applications of finite-set statistics (FISST). FISST is at root a careful, direct generalization of formal Bayes statistical modeling

to multisensor-multitarget problems. Ironically, this fact partly explains why many find it somewhat formidable. Data fusion engineers have typically been trained to think in bottom-up terms, rather than from the system-level viewpoint that direct generalization requires. As familiarity increases and as—hopefully—FISST continues to move closer to application, we expect that it will seem less novel. Because FISST closely emulates the familiar "Statistics 101" formalism, it is—with suitable pedagogic streamlining—potentially accessible to advanced undergraduates. For the interested reader, the best entry points into FISST are the technical monograph [15] or its condensed version [18]. References to more detailed aspects of FISST can be found in the body of the paper.

For lack of space, we were unable to describe the other major goal of FISST: extending formal Bayes modeling methods to data that is ambiguous, either because it is itself mathematically difficult to model or because the process by which it is generated (i.e., its likelihood function) is imperfectly understood. See the publications [8, 15, 20, 29] for further detail.

REFERENCES

[1] El-Fellah, A., Zajic, T., Lazja-Rooks, B., and Mehra, R. (2001)
Multitarget nonlinear filtering based on spectral compression and probability hypothesis density.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition X*, SPIE Proceedings, **4380** (2001), 207–216.

[2] Ballantyne, D. J., Chan, H. Y., and Kouritzin, M. A. (2001)
A branching particle-based nonlinear filter for multi-target tracking.
In *Proceedings of 2001 International Conference on Information Fusion*, Montreal, Aug. 7–10, 2001.

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

[4] Challa, S., Vo, B-N., and Wang, X. (2002)
Bayesian approaches to track existence—IPDA and random sets.
In *Proceedings of 5th International Conference on Information Fusion*, Vol. II, 2002, 1228–1235.

[5] Daley, D. J., and Vere-Jones, D. (1988)
An Introduction to the Theory of Point Processes.
New York: Springer-Verlag, 1988.

[6] Doucet, A., de Freitas, N., and Gordon, N. (Eds.) (2001)
Sequential Monte Carlo Methods in Practice.
Springer, 2001.

[7] Goodman, I. R., Mahler, R. P. S., and Nguyen, H. T. (1997)
Mathematics of Data Fusion.
Boston: Kluwer Academic Publishers, 1997.

[8] Hoffman, J. R., Mahler, R., Ravichandran, R., Mehra, R., and Musick, S. (2003)
Robust SAR ATR via set-valued classifiers: New results.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition XII*, SPIE Proceedings, **5096**, 139–150.

[9] Hue, C., Le Cadre, J-P., and Peréz, P. (2002)
Sequential Monte Carlo methods for multiple target tracking and data fusion.
IEEE Transactions on Signal Processing, **50**, 2 (2002), 309–325.

[10] Lanterman, A. D. (2003)
Sampling from multitarget Bayesian posteriors for random sets via jump-diffusion processes.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition XII*, SPIE Proceedings, **5096**, 300–311.

[11] Lanterman, A. D., Miller, M. I., Snyder, D. L., and Miceli, W. J. (1994)
Jump-diffusion processes for the automated understanding of FLIR scenes.
In F. A. Sadjadi (Ed.), *Automatic Target Recognition IV*, SPIE Proceedings, **2234** (1994), 416–427.

[12] Mahler, R. (2003)
Bayesian cluster tracking using a generalized Cheeseman approach.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition XII*, SPIE Proceedings, **5096**, 334–345.

[13] Mahler, R. (2002)
An extended first-order Bayes filter for force aggregation.
In O. Drummond (Ed.), *Signal and Data Processing of Small Targets 2002*, SPIE Proceedings, **4729** (2002), 196–207.

[14] Mahler, R. (1998)
Global posterior densities for sensor management.
In M. K. Kasten and L. A. Stockum (Eds.), *Acquisition, Tracking, and Pointing XII*, SPIE Proceedings, **3365** (1998), 252–263.

[15] Mahler, R. (2000)
An Introduction to Multisource-Multitarget Statistics and Its Applications.
Lockheed Martin Technical Monograph, Mar. 15, 2000.

[16] Mahler, R. (2003)
Multisensor-multitarget sensor management: A unified Bayesian approach.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition XII*, SPIE Proceedings, **5096**, 222–233.

[17] Mahler, R.
Objective functions for Bayesian control-theoretic sensor management, II: MHC-like approximation.
In S. Butenko, R. Murphey, and P. Paralos (Eds.), *New Developments in Cooperative Control and Optimization*, Boston: Kluwer Academic Publishers, to be published.

[18] Mahler, R. (2002)
Random set theory for target tracking and identification.
In D. L. Hall and J. Llinas (Eds.), *Handbook of Multisensor Data Fusion*, Boca Raton, FL: CRC Press, 2002, 14-1–14-133.

[19] Mahler, R.
Tractable multistep sensor management via MHT.
Proceedings of the Workshop on Multi-Hypothesis Tracking: A Tribute to Samuel Blackman, San Diego, CA, May 30, 2003, to be published.

[20] Mahler, R., Leavitt, P., Warner, J., and Myre, R. (1999)
Nonlinear filtering with really bad data.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition VIII*, SPIE Proceedings, **3720** (1999), 59–70.

[21] Mahler, R., and Zajic, T. (2002)
Bulk multitarget tracking using a first-order multitarget tracking filter.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition XI*, SPIE Proceedings, **4729** (2002), 175–186.

[22] Moreland, M., and Challa, S.
A multi-target tracking algorithm based on random sets.
In *Proceedings of 6th International Conference on Information Fusion*, Cairns, Australia, July 8–11, 2003.

[23] Musick, S., Kastella, K., and Mahler, R. (1998)
A practical implementation of joint multitarget probabilities.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition VII*, SPIE Proceedings, **3374** (1998), 26–37.

[24] Musicki, D., Evans, R., and Stankovic, S. (1994)
Integrated probabilistic data association.
IEEE Transactions on Automatic Control, **39**, 6 (1994), 1237–1241.

[25] Naylor, J. C. and Smith, A. F. M. (1982)
Application of a method for the efficient computation of posterior distributions.
Applied Statistics, **31**, 3 (1982).

[26] Portenko, N., Salehi, H., and Skorokhod, A. (1997)
On optimal filtering of multitarget tracking systems based on point processes observations.
Random Operators and Stochastic Equations, **1** (1997), 1–34.

[27] Sidenbladh, H.
Multi-target particle filtering for the probability hypothesis density.
In *Proceedings of 6th International Conference on Information Fusion*, Cairns Australia, July 8–11, 2003.

[28] Sidenbladh, H., and Wirkander, S-L.
Tracking random sets of vehicles in terrain.
In *Proceedings of 2003 IEEE Workshop on Multi-Object Tracking*, Madison WI, June 21, 2003.

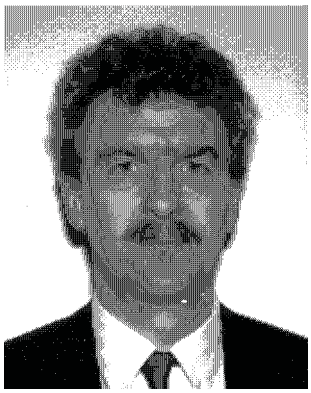
[29] Sorensen, E., Brundage, T., and Mahler. R. (2001)
None-of-the-above (NOTA) capability for INTELL-based NCTI.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition X*, SPIE Prodeedings, **4380** (2001), 281–287.

[30] Stoyan, D., Kendall, W. S., and Meche, J. (1995)
Stochastic Geometry and Its Applications, (2nd ed.).
New York: Wiley, 1995.

[31] van Trees, H. L. (1968)
Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory.
New York: Wiley, 1968.

[32] Vo, B., Singh, S., and Doucet, D.
Sequential implementation of the PHD filter for multi-target tracking.
In *Proceedings of 6th International Conference on Information Fusion*, Cairns, Australia, July 8–11, 2003.

[33] Zajic, T., Hoffman, J., and Mahler, R. (2000)
Scientific performance metrics for data fusion: new results.
In I. Kadar (Ed.), *Signal Processing, Sensor Fusion, and Target Recognition IX*, SPIE Proceedings, **4052** (2000), 172–182.



Ronald P. S. Mahler was born in Great Falls, MT, in 1948. He earned a B.A. in mathematics from the University of Chicago, Chicago, IL, in 1970, a Ph.D. in mathematics from Brandeis University, Waltham, MA, in 1974, and a B.E.E. in electrical engineering from the University of Minnesota, Minneapolis, in 1980.

He was an assistant professor of Mathematics at the University of Minnesota from 1974 to 1979. Since 1980 he has been employed as a research engineer at Lockheed Martin NE&SS Tactical Systems, Eagan, MN. His research interests include data fusion, expert systems theory, multitarget tracking, combat identification, sensor management, random set theory/point process theory, and conditional event algebra.

Dr. Mahler is the author, coauthor, or coeditor of over forty publications, including eleven articles in refereed journals, a book, a hardcover conference proceedings, and a monograph. He has been invited to present at many universities, U.S. government laboratories, and conferences including Harvard, Johns Hopkins, the University of Wisconsin, the University of Massachusetts, the Air Force Institute of Technology, SPAWAR Systems Center, the IEEE Conference on Decision and Control, the International Conference on Information, Decision, and Control, and the International Conference on Information Fusion.