Tieman, Larry Robert

A REAL-TIME MULTITARGET TRACKER BY ADAPTIVE HYPOTHESIS TESTING FOR AIRBORNE SURVEILLANCE SYSTEMS

The University of Oklahoma PH.D. 1982

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A REAL-TIME MULTITARGET TRACKER BY ADAPTIVE HYPOTHESIS TESTING FOR AIRBORNE SURVEILLANCE SYSTEMS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

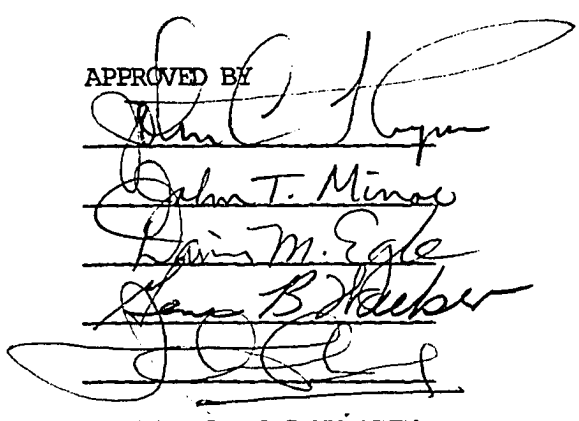
BY

LARRY ROBERT TIEMAN

Norman, Oklahoma

1982

A REAL-TIME MULTITARGET TRACKER BY ADAPTIVE HYPOTHESIS TESTING FOR AIRBORNE SURVEILLANCE SYSTEMS



APPROVED BY

DISSERTATION COMMITTEE

ACKNOWLEDGEMENTS

Sincere appreciation is extended:

To Dr. John Thompson for his training, inspiration and support.

To Dr. Gene Walker, Dr. Davis Egle, Dr. John Cheung and Dr. John Minor for serving on the committee.

To Dr. James Payne whose early sage advice was invaluable.

To my friends at the 552 AWAC Wing especially James Davidson, Ralph Hollenbeck, Dan Holder and Roy Lingo who helped in countless ways.

To Willis Janssen who did the graphics and editing.

To Ms. Glenda Thompson who prepared the manuscript and Southwest Business Systems who graciously supplied the quality equipment.

And, to my wife, Charlotte, who has suffered a rather lonely year.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

A REAL-TIME MULTITARGET TRACKER BY ADAPTIVE HYPOTHESIS TESTING FOR AIRBORNE SURVEILLANCE SYSTEMS

CHAPTER I

INTRODUCTION

In the last decade a considerable volume of literature has developed on recursive algorithms. Because of the variety of military and civilian applications, ranging from ballistic missile defense to harbor traffic control, and because of the diversity of sensors, extending from radar to optical scanners, numerous algorithms have been developed for specific purposes. These algorithms are generally based on Kalman filter theory with a vast body of research in the various extensions and suboptimizations necessary for real time applications. These algorithms are generally successful for a specific application which is often a single-target, single sensor system or a multiple-target, single sensor system.

In airborne surveillance systems, tracking multiple targets with a variety of sensors is a requirement. In target tracking there is often uncertainty associated with the origin of an observation as well as measurement inaccuracy, usually modeled as additive white noise. This uncertainty can result from clutter, false targets, or track densities which preclude a positive target/observation pairing. Selection of the wrong measurement as input to the tracking algorithms

can have disastrous effects on track continuity, maneuver response, and smoothed parameters.

One means, in theory, of reducing the observation uncertainty is to equip aircraft with transponders which reply to interrogation with range, azimuth, and a track signature that can be unique. This track signature, which can be changed by the pilot upon request, offers positive proof of the observation's origin and can be accepted or rejected as input to the tracking algorithm depending upon whether or not it matches the code assigned in the database.

Airborne surveillance systems employ transponder technology with less success than ground based systems. As early as 1977, operators aboard the Airborne Warning and Control System's E-3As reported significant difficulty in maintaining track continuity for tracks with beacon reports as the primary measurement. This was in direct conflict with the experience of ground based control systems and underscored the change in environment and the role of the airborne platform. New environmental and operational demands present unique tracking problems.

This dissertation presents the framework for a new multitarget tracker specifically designed for an airborne surveillance platform. This Real-time Adaptive Hypothesis Testing Tracker (RAHTT) is influenced by the sensors available, several previous tracker algorithms, the environment of an airborne system, and the tactics of modern high performance fighters. This first chapter will examine each of these factors in order to provide a clear, consolidated starting point for the development of the RAHTT algorithm.

The pertinent mathematics are reviewed in the second chapter. Unlike more ad hoc algorithms, RAHTT is based on a solid mathematical foundation. As will be discussed in chapter III, this mathematical basis allows an initial simplification of the problem without concern for the eventual extensions.

In chapter III the concepts and important details of RAHTT are examined. Because the algorithmn is rather complex, considerable effort is made to explain the concept of RAHTT.

The results of RAHTT are evaluated in chapter IV. As will be discussed more thoroughly then, it is difficult to evaluate RAHTT and other existing algorithms due to their philosophical differences.

Finally, chapter V will give the conclusions of this work and the remaining extensions necessary to implement RAHTT in an airborne system.

Sensor Fundamentals

Radar

Development of background information logically begins with the sensor systems. Sensors are divided into two classes, those that provide number-of-targets type information as well as the location of each target and those that provide some form of identification about the originating target. Radar is typical of the former and IFF, of the later. The track of a target can be determined by discrete measurements taken from scan to scan. The quality of such a track is dependent on the time between observations, the accuracy of the

sensor, and the number of other targets and reports in the vicinity [Skolnik,1980:183].

Radar provides azimuth, range, and, depending upon radar type, range rate (radial velocity) and altitude. Associated with these parameters, there are measurement errors, generally assumed as additive white noise, which must be reduced by some estimation technique. Some form of Kalman filter is the preferred estimation technique due to its low memory requirement and recursiveness.

In addition to these parameters for true reports, radar also returns false alarms. A false alarm, which has a Poisson distribution [Skolnik,1980:123-129], results when noise exceeds the minimum detectable signal. To a tracking algorithm, a false alarm is indistinguishable from a true measurement.

Potentially, the most powerful tool for discriminating between targets in radar tracking is range rate. When coupled with range, range rate provides exceptionally accurate maneuver and heading information. Discounting engine modulation effects, which may be accurate enough for aircraft type identification [Allen,1980], the range rate of an aircraft in straight-line, unaccelerated flight has very little error. Range rate can provide information on acceleration or heading changes. Because of its sensitivity to heading, range rate provides an additional dimension for discriminating among dense traffic.

Range rate also provides an effective means of rejecting clutter. Reports that have a range rate below a threshold are likely to be noise and are not processed as true reports. Complicating the

use of range rate, certain track geometries can produce a low radial velocity report that is indistinguishable from clutter.

The nature of long range radar requires slow rotation rates (commonly six or ten RPM). With a single receiver and unity detection probability, the sample rate is only every six or ten seconds. Reid [1979] noted that tracker performance is bound by the accuracy of the data and the data rate. This is particularly true in the maneuvering, multitarget problem. When targets maneuver, poor measurement resolution will disguise the maneuver. If the maneuver is not large, the filter will treat it as a particularly noisy report and will suffer degradation in the target estimates. If the maneuver is large but the data rate is too low, the target can become hopelessly lost among other reports.

One proposal [Allen,1980] is to increase the number of radars within the present rotodome or vary the rotation so that the radar repeatedly scans only a sector of the sky. While the feasibility of this proposal, in terms of aircraft structure, power, and space, is unstudied, it clearly would have positive effects on tracker performance.

Radar has no inherent identification characteristics. When the report proximity exceeds the radar's resolution, the lack of identification data can have serious consequences for tracking algorithms. In the multitarget problem, many target geometries cause targets to exchange tracks (tracks switches). Once a switch has occurred, there is no automatic means available to reestablish the true track on its correct trajectory.

Identification Friend of Foe (IFF)

Identification data are gathered on targets suitably equipped with transponders, such as the U.S. standardized Mark XII IFF/SIF (selective identification feature). IFF provides range, azimuth, and, usually code and altitude information. This transmitted code can be compared with the code assigned to specific tracks and stored in the database to decrease the observation's uncertainty.

A limitation to transponder technology is that an aircraft must be equipped with the appropriate hardware and must receive the interrogation before a report is generated. Obviously, those aircraft without transponders cannot be tracked by IFF.

Years of experience with ground based systems show that IFF measurement error is white and that the probability of receiving a correct code is approximately 0.95. Yet, when the first operational missions with the Airborne Warning and Control System's E-3As were flown, periods of erratic azimuthal measurements were observed. In addition, during these periods of azimuthal excursions, the probability of decoding the track's code dropped dramatically. This perturbation of azimuth is known as jitter.

Jitter is related to target density, distance from the airborne receiver, and the reflectivity of the surface over which the targets are flying. These factors cause multipath returns to the transponder receiver which garble the code and defeat the hardware's ability to determine report azimuth.

All existing systems have some multipath effect. The mitigating circumstance for ground systems is that the multipath report is so close to the direct reply that no detectable distortion occurs. For airborne systems relying on IFF, the multipath effects can be disastrous.

IFF was designed as a means of identification, but because of its discrete capability, many systems rely on the measurement for tracking. As will be discussed in the Environment section, the potential problems of IFF make it highly unreliable as a primary tracking source. The unexpected problems of jitter and code reliability degrade the capabilities of transponder tracking by increasing the uncertainty of the report's origin. There is, however, considerable information to be gleaned from an IFF report other than as a primary input to the filter.

Since 1977, the Air Force, the MITRE Corporation, and The Boeing Company have studied the effects and causes of jitter, and the potential software and hardware solutions for the E-3A. As of now, no completely suitable software fix has been proposed. While studies continue at Tinker AFB and at MITRE on software suitable for integration in present tracking algorithms, the research presented in this dissertation extends previous work to a new tracker designed more suitable for the environment of the E-3A and other airborne sensor systems and more capable of dealing with jitter. Before examining this new design, it is instructive to understand the fundamentals of tracking targets and the environment that makes an airborne receiver unique.

Tracking Fundamentals

Terminology

Three common elements in all the tracking algorithms to be examined are a predictive phase, an association/correlation phase, and a smoothing phase.

Track prediction is the estimation of the target's position to the time of the next observation. This is done in conjunction with the state extrapolation of the Kalman filter and is the prime input to the second phase, association/correlation.

Association is a coarse screening process which selects for further consideration only those tracks that have a reasonable probability of being related to a specific report. Correlation is a fine screening process which determines report/track pairs that have a high probability of being related to each other (correlated pairs). Not all algorithms require a correlation but all require some form of association. Typically, an association window is conceptually drawn around the predicted report position. This window, which may be a circle, an ellipse, or a simple range and azimuth test, represents a statistical area that should contain the next observation. The association window is based upon the extrapolated track states and performance characteristics (the track's ability to maneuver, accelerate, or decelerate) weighted by some statistical multiplier to raise the confidence level.

It should be clear that in dense traffic, the association windows for different tracks will overlap. For those algorithms that require a one-to-one track/report pair, a correlation process breaks the multiple associations. A track signature plays a very important role for these algorithms because it reduces the likelihood of a wrong pairing. If there is no signature and there are multiple associations, the probability of an incorrect association increases. Various algorithms have been proposed to contend with the multiple association problem. Though the approaches vary greatly, these methods all utilize an association window for report selection and a Kalman filter for track smoothing.

The peril of multitarget tracking is the association of the observation with the wrong track. The recent literature in multitarget tracking is motivated by the need to find a reasonable means of incorporating the uncertainty of the measurement's origin into the tracking algorithms Y@ar-Shalom,1978]. Those algorithms having a direct impact on this dissertation are presented in historical order.

Major Algorithms

Nearest Neighbor. This is the standard and most often implemented correlation procedure. The observation selected is the one that most nearly matches the predicted position.

In most implementations, "nearest" is based upon some positional test value and IFF code matches. A positional test value

is based on the sum of the standardized errors of those parameters appropriate for the report (i.e. range, azimuth, and range rate error).

Nearest neighbor can lead to very poor results in areas where there are false targets, multiple targets, or jitter because the tracking algorithm does not account for the possibility that the measurement used might have originated from a source other than the track of interest [Bar-Shalom,1978].

Track Splitting. Smith and Buechler [1975] presented a multiple-object branching algorithm. Tracks are initialized based upon one or two measurements; then for each measurement falling within the association window, the track is split. From the innovations property of the Kalman filter, the likelihood of each split is calculated.

Since the optimal branching solution consists of splitting each preceding trajectory into alternate trajectories at each scan and then evaluating alternatives, for N scans the number of alternatives is sufficient to overwhelm any real-time machine. In order to make a practical algorithm three pruning rules are used:

1. If the measurement is further than a specified maximum distance from the predicated position of a track, it is considered unlikely to have originated from this track and is discarded (association).
2. If two hypothesized trajectories are similar they likely represent the same track. Trajectories within a specified distance of a more likely

trajectory are dropped. Up to L_{MAX} of the best trajectories are kept.

3. Trajectories that fall below a given likelihood threshold are dropped.

Reid[1979] pointed out several flaws with this work. In calculating the likelihoods, Smith and Buechler assumed that each report was present (P_d =1.0), and they did not account for false alarms. Moreover, apparently a target can be associated with every measurement within it's association window. If the track density if sufficient, measurements can be in several windows leading to data-association hypotheses that are not mutually exclusive.

In spite of these errors, track splitting is a fundamental concept of many modern trackers. The work of Smith and Buechler extended the theory of branching algorithms by noting that if unknown process parameters are constant and assume only a limited number of values, then the optimal nonlinear filter can be separated into parallel linear filters.

Maximum Likelihood. The problems of false returns, missing reports, and mutually exclusive data associations were addressed by Morefield [1977]. This approach yields the most likely data association hypotheses by approaching the issue as an unsupervised pattern recognition problem.

Feasible track trajectories are reduced by a coarse window association. The set of feasible tracks, F, is considered as clusters of measurements Z that are reasonable to incorporate into hypotheses. The Bayesian decision process is restricted to hypotheses formed using

F, so that any hypotheses formed using F is simply a subset of Z. In other words, a measurement cannot belong to more than one track.

It is natural to compute the likelihood function of the Kalman filter state estimates, basing the hypothesis test on the innovations sequence. In this problem, finding the most likely trajectory coincides with maximizing the likelihood function of the feasible measurements over all feasible tracks. The negative log-likelihood function is modified to include the number of points in each track and the hypervolume in which the observations are made.

Morefield formulates multitarget tracking in terms of an integer linear programming problem. His algorithm is basically a batch-processing technique allowing N data points to accumulate before backtracking through the data. Although a recursive version is included, it does not guarantee optimality over time as does the batch-processing version [Reid,1979].

Both track splitting and maximum likelihood are essentially nonBayesian approaches that attempt to make decisions upon the likelihood of a trajectory and then estimate the states of the trajectory. The state estimates and the covariances are conditioned upon accepted tracks being true.

Probabilistic Data Association. Probabilistic data association (PDA) algorithms are Bayesian approaches that yield estimates and covariances and account for measurement origin uncertainty [Bar-Shalom and Tse, 1975].

In the suboptimal form, the best estimate of a target's state is the conditioned mean based upon the measurements that have some

probability of originating from that track. These observations are selected by constructing an association window. Those reports falling within the window are assumed to have some probability of originating from the track of interest. This "all neighbors" approach does not allow track splits but combines all the latest measurements into the state estimates.

While the suboptimal approach used only the last observation, in the optimal form, the state estimate is a combination of all measurements from initial to present. The exponentially increasing memory and computational requirements are eased by combining all tracks which have identical histories for the past N observations. This modification is referred to as the "N-scan-back" filter.

The Reid Algorithm. A potentially more powerful algorithm is the nonprobabilistic data association, Baynesian, multiple-scan algorithm of David B. Reid[1979].

The basic approach is to generate a set of data-association hypotheses that account for the origin of all observations. The probabilities of these associations are calculated recursively using all known information, including the report density and sensor characteristics. A tree structure is constructed with each branch representing a data association. This technique allows a track/report pairing based upon previous and subsequent measurements. To minimize the computation and space requirements, unlikely hypotheses are eliminated and like target estimates are combined. All targets and measurements are divided into independent cluster. Clustering gives

the algorithm a parallel characteristic since each cluster can be processed independently.

Multiple-scan associations give the algorithm the capability to use later measurements to aid correlation. Hence, the algorithm is not committed to a unique pairing until sufficient data is available to make the correlation with a degree of certainty.

Adaptive Hypothesis Testing. Closely related to the work of Reid is that of Keverian and Sandell[1979]. Their algorithm is nonBayesian and differs in hypothesis deletion strategies. Keverian and Sandell have been influenced by work in hypothesis testing for dynamic systems, especially the multiple-model adaptive estimation (MMAE) of Magill[1965].

A MMAE algorithm provides an optimal nonlinear filter for the estimation problem with observations coming from one of a finite set of linear systems [Keverian and Sandell,1979:5]. In its Bayesian form the algorithm recursively calculates the probabilities of the hypotheses that one of the possible linear systems is the actual system and produces the optimal estimate of the system state. In general, it can be shown that the MMAE algorithm identifies the linear model closest, where the measure of closeness is some information distance [Baram and Sandell, 1968], to the true state, which may be nonlinear and high order.

In the multitarget problem, the set of possible linear systems generating the measurements corresponds to the set of objects being tracked. By adapting the hypotheses deletion and creation to the evolving situation, a data-driven algorithm adaptive on a more

abstract level is created. Such an algorithm is termed adaptive hypothesis testing.

Multiple-scan correlations and hypothesis testing offer a framework for a more advanced multitarget tracker appropriate for the environment encountered by airborne surveillance systems such as AWACS and NIMROD. To understand why existing tracking systems are inadequate, it is important to understand the environment these systems encounter.

Environment

Surveillance Volume

By virtue of being at altitude, the surveillance volume for airborne systems is considerably larger than more powerful ground based systems. With this increased coverage comes far more sensor measurements than can possibly be tracked. For example, a single E-3A off the coast of Virginia can monitor nearly the entire East Coast. Such a vast number of observations would swamp any real-time computing system, especially when it is noted that the surveillance function is only one of many Command and Control tasks competing for processing time.

Many proposed algorithms assume that all targets will be tracked. With the Reid algorithm, for example, a report is either the track desired, a new track, or a false report. This simplifies the problem of resolving correlation conflicts since all observations are either tracks or false reports. Since the statistics of false reports are well known, there is a fixed statistic available for assigning

probabilities to various types of origins. Reid also assumes that the density of known to unknown targets is available. While this data could be adaptive, the point remains that an airborne system cannot be designed to track all targets in all environments.

Target Density

During engagements track densities become exceptionally high. Considering the range from the battle at which airborne systems operate, correct report/track pairing becomes exceptionally difficult. Algorithms, based on nearest neighbor criteria, that force a decision on correlated pairs in this environment are often wrong.

IFF will not provide any information in the battle area. Most tacticians agree that fighter aircraft will turn off their transponders when they reach the forward edge of battle to preclude the enemy from identifying and tracking them. Any algorithm that relies on IFF as the primary sensor for tracking will find itself seriously degraded in a major engagement.

Machine-Machine, Man-Machine Interface

Multiple-scan correlation algorithms are limited by the interfaces taking place in a Command and Control network. The periodic reporting of target information is utilized throughout a battle area to make time dependent decisions. Other computer based systems are often tracking identical targets that, if not updated properly, will result in considerable confusion. Frequently, therefore, a multiple-scan algorithm must reach a decision more rapidly than the data will allow.

A more serious problem is the man-machine interface. The operator of the system must have timely track updates upon which to make decision; a track cannot "hang in space" while ambiguities are resolved by the tracker. The operator of the system can resolve tracking conflicts far better than any software, but his attention is often focused on more critical areas of the engagement making intervention in software decisions unlikely. When an area of the battle has his attention, the position update must come with the regularity he expects. By using, at worst, the extrapolated movement of the target, the past history of the observations, and his knowledge of tactics, he can make decisions about track trajectories that software could not reach. This limits the time a tracker can wait before forcing a decision on ambiguous data. A careful balance must be developed that keeps the man-machine interface foremost in perspective.

Environmental Factors Not Considered

There are additional complications that are not examined in this study. The most difficult, and not fully resolved problem, is jamming. The advances in electronic counter measures are primarily dealt with in hardware. In this work all tracking is done "in the clear."

A second area, the problems and errors induced in ground stabilizing the sensor measurements is not included. It is assumed that this conversion is error free. While error free conversion is not possible, carefully derived equations can guarantee maximum errors of .2 NM. Finally, the position of the surveillance platform is

considered stationary and known. This is done primarily to increase the readability of the equations. Navigation equipment has progressed to the point that positional error of the platform is not significant.

Summary

In this first chapter, the literature and language of multiple correlation algorithms and tracking were introduced. A brief discussion was presented on the environment and the tracking problems of airborne sensors.

The key mathematics of RAHTT are examined in the next chapter. The algorithm developed in the sequel closely follows the work of Reid and of Kervian and Sandell except that it is designed for an airborne system. Since a major problem for current trackers using IFF is jitter and mode/code reliability, an adaptive scheme is developed to maximize the information from an IFF report. The result is an algorithm that is adaptive on several levels. Since a primary goal of this research is to extend previous work to include IFF jitter, the tracks are assumed to be straight-line, nonmaneuvering. This assumption does not appear to be overly restrictive and follows the examples of previous authors in validating new concepts. Further, this restriction eliminates only a few data points as interceptors tend to fly straight-line, unaccelerated flight paths, except during combat, with only periodic course corrections.

Finally, as mentioned and as will be more fully developed in the next chapter, the underlying theory of the algorithm provides a straight forward, though not trivial, extension to nonlinear motion.

CHAPTER II

KALMAN FILTERS AND MULTIPLE MODEL ESTIMATION

Introduction

The key mathematics of the algorithm presented in the next chapter come from two closely related fields—Kalman filter theory and multiple model estimation. It has been over a decade since Kalman[1960] and Kalman and Bucy[1961] extended Wiener's work. During that time the papers and books written explaining, extending, and modifying Kalman's work number in the thousands. In a similiar way, the work of Magill[1965] in multiple model estimation has been studied, extended, and formalized.

The theoretical foundation of this dissertation is the Multiple Model Estimation Algorithm (MMEA) and its variations. Since MMEA is dependent on Kalman filter theory, it is appropriate to begin there.

Discrete Kalman Filter

Introduction

The purpose of this section is to introduce the basics of discrete time Kalman filter and derive the equations needed in the

next chapter. A more rigorous treatment of the field can be found in numerous texts.[1]

The Kalman filter is generally accepted as the best method of providing the motion analysis of the steady state parameters of a moving target. With benefit of foresight, this discussion of Kalman filtering is limited to the discrete time linear Kalman filter and linear equations of target motion.

Generally, the motion, measurement, and Kalman filter equations are partitioned into X and Y components. This uncoupling yields independent X and Y tracking algorithms which must be solved each iteration.

There are important benefits from this decoupling. Sensor measurements are generally in terms of range and azimuth. With the exception of radial or constant speed circular flight, an aircraft flying at a constant velocity in polar coordinates causes a nonlinear change in range and azimuth. Thus polar coordinates are generally inconsistent with aircraft tracking algorithms [Burke,1972].[2]

The reduction in size of the covariance and state matrices that accompany partitioned components significantly reduces the amount of computer time needed. Kalata[1976] states that for a six-state tracking problem, partitioning the filter into a three dimensional system reduced computation time from 5000 to 375 microseconds.

---

[1] For a basic introduction to Kalman filtering with applications to aircraft tracking see Burke[1972]. An advanced mathematically oriented work is Jazwinski[1970].

[2] Various authors have proposed spherical tracking systems. See [Moose, Vanlandingham, and McCabe, 1979].

Further, the loss in performance due to the partitioning can be bounded by comparing the off diagonal terms of the measurement uncertainty matrix R with the diagonal terms. These are generally very small percentages.

Equations of State and Measurement

For the remainder of this work, targets are assumed to evolve according to the general equation

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{w}(k) \qquad 2.1$$

where

Φ is the state transition matrix

Γ is the disturbance matrix

w is a white noise sequence with zero mean and covariance Q

The state variables are related to measurements z by

$$\underline{z}(k+1) = H\underline{x}(k+1) + \underline{v}(k+1) \qquad 2.2$$

where

H is a measurement matrix

v is a white noise sequence with zero mean and covariance R.

Φ and H are represented as time invariant only for notational ease. The term $\Gamma\underline{w}(k)$ is used to model random disturbances in the state vectors; often representing aircraft maneuvers or inaccuracies or unknowns in the state model. It is ignored for the remainder of this work.

For straight line, unaccelerated target, the equations of motion are

$$x(k+1) = x(k) + \dot{x}(k)\,\Delta t \qquad 2.3$$

$$x(k+1) = \dot{x}(k) \qquad 2.4$$

$$y(k+1) = y(k) + \dot{y}(k)\,\Delta t \qquad 2.5$$

$$y(k+1) = \dot{y}(k) \qquad 2.6$$

where

Δt is the time interval between updates

$\dot{x}$ and $\dot{y}$ components are independent components of velocity

k is the time index ($k\,\Delta t$ is the total elapsed time).

Placing equations 2.3 through 2.6 in the form of 2.1 and dropping the time index yields

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & 0 \\ \dot{x} & 0 \\ 0 & y \\ 0 & \dot{y} \end{bmatrix} \qquad 2.7$$

Because measurements are received in range and azimuth but the equations of motion and the Kalman filter equations are in terms of x and y it is necessary to calculate the variance of the measurement in terms of x and y (see appendix B).

$$V_{xm}(k+1) = (\sigma_R \operatorname{Cos}\psi)^2 + (\sigma_\theta R \operatorname{Sin}\psi)^2 \qquad 2.8$$

$$V_{ym}(k+1) = (\sigma_R \operatorname{Sin}\psi)^2 + (\sigma_\theta R \operatorname{Cos}\psi)^2 \qquad 2.9$$

where

R is the range of the report

Ψ is the azimuth of the report

σ_R is the standard deviation of sensor range

σ_θ is the standard deviation of sensor azimuth

Thus the measurement equations are

$$x_m(k+1) = x(k+1) + V_{xm}(k+1)$$

$$y_m(k+1) = y(k+1) + V_{ym}(k+1) \qquad 2.10$$

and, in the form of equation 2.2,

$$\underline{Z} = \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} V_{xm} \\ V_{ym} \end{bmatrix} \qquad 2.11$$

The estimation of the state variables is the objective of the Kalman filter. There are two phases to the estimation problem that provide a minimum variance estimate of the states and that give the filter its recursive feature. These two phases are examined in order.

Prediction or Extrapolation

The general Kalman filter equations for state and covariance prediction are

$$\hat{\underline{X}}(k+1|k) = \Phi \hat{\underline{X}}(k|k) \qquad 2.12$$

$$\hat{P}(k+1|k) = \Phi \hat{P}(k|k) + Q \qquad 2.13$$

where

$\hat{\underline{x}}$ is the state estimate

$\hat{\underline{P}}$ is the estimated covariance matrix of the states

Q is the covariance of the disturbance matrix and may be dependent on Γ. Under the assumptions already introduced, Q is dropped.

The notation (k+1|k) refers to the estimation problem when the time of interest occurs after the last available measurement. In tracking problems, this is the estimation of the target's next position based upon the filter's estimate of the target's last position. The notation (k|k) is used to indicate that an estimate is desired to coincide with the last measurement point. In estimation theory this is called filtering but more often, in aircraft tracking, is referred to as smoothing. Smoothing, in estimation terms, occurs when the time of interest falls within the span of available data[Gelb,1979:2]

With these definitions in mind, the prediction equations for the state variable are

$$\begin{aligned}
\hat{x}(k+1|k) &= \hat{x}(k|k) + \hat{\dot{x}}(k|k)\,\Delta t \\
\hat{\dot{x}}(k+1|k) &= \hat{\dot{x}}(k|k) \\
\hat{y}(k+1|k) &= \hat{y}(k|k) = \hat{\dot{y}}(k|k)\,\Delta t \\
\hat{\dot{y}}(k+1|k) &= \hat{\dot{y}}(k|k)
\end{aligned} \qquad 2.14$$

where

$\hat{}$ denotes estimated terms.

For the partitioned constant speed straight-line cartesian coordinate formulation of aircraft motion, the covariance matrix P for the x component of state variables is

$$P_x = \begin{bmatrix} V_x & C_{x\dot{x}} \\ C_{x\dot{x}} & V_{\dot{x}} \end{bmatrix} \qquad 2.15$$

where

V_x is the variance in x

$C_{x\dot{x}}$ is the covariance of x and $\dot{x}$

and the time indicies are dropped for convenience.

V_x and V_y are initialized to V_{xm} and V_{ym} respectively.

$V_{\dot{x}}$ is found by noting that

$$\hat{\dot{x}} = \frac{x_m(k+1) - x_m(k)}{\Delta t}$$

$$\tilde{\dot{x}} \triangleq \dot{x} - \hat{\dot{x}} \qquad \tilde{x}_m \triangleq x - x_m$$

$$\text{then} \quad \tilde{\dot{x}} = \frac{\tilde{x}_m(k+1) - x_m(k)}{\Delta t}$$

$$\tilde{\dot{x}}^2 = \frac{\tilde{x}_m(k+1)^2 + \tilde{x}_m(k)^2 - 2\tilde{x}_m(k+1)\,\tilde{x}_m(k)}{\Delta t^2}$$

but since $\tilde{x}_m(k+1)$ and $\tilde{x}_m(k)$ are independent

$$V_{\dot{x}} = E[\tilde{\dot{x}}^2] = \frac{2V_{xm}}{\Delta t^2} \qquad 2.16$$

where

$\hat{\dot{x}}$ is the estimate of $\dot{x}$

$\tilde{\dot{x}}$ is the error in the estimate

In a similar way

$$\hat{\hat{x}}\dot{\hat{x}} = x_m(k+1)[x_m(k+1) - x_m(k)]/\Delta t$$

$$\tilde{x}\dot{\tilde{x}} = \tilde{x}_m(k+1)[\tilde{x}_m(k+1) - \tilde{x}_m(k)]/ t$$

$$= \frac{(\tilde{x}_m(k+1))^2 - \tilde{x}_m(k+1)\tilde{x}_m(k)}{\Delta t}$$

$$C_{x\dot{x}} = E[\tilde{x}\dot{\tilde{x}}] = \frac{V_{xm}}{\Delta t}$$

The derivation for P_y is identical. The predicted covariance equations are

$$V_x(k+1|k) = V_x(k|k) + 2\,\Delta t C_{x\dot{x}}(k|k) + \Delta t^2 V_{\dot{x}}(k|k)$$

$$V_y(k+1|k) = V_y(k|k) + 2\,\Delta t C_{y\dot{y}}(k|k) + \Delta t^2 V_{\dot{y}}(k|k)$$

$$V_x(k+1|k) = V_{\dot{x}}(k|k) \qquad 2.17$$

$$V_y(k+1|k) = V_{\dot{y}}(k|k)$$

$$C_{xx}(k+1|k) = C_{x\dot{x}}(k|k) + \Delta t V_x(k|k)$$

$$C_{yy}(k+1|k) = C_{y\dot{y}}(k|k) + \Delta t V_y(k|k)$$

In this section the state and covariance prediction equations were derived for partitioned, straight line, constant velocity aircraft motion. In the next section the smoothed state and covariance equations are developed.

Update and Gain Equations

The general Kalman filter equations for state and covariance update are

$$\hat{\underline{x}}(k+1|k+1) = \hat{\underline{x}}(k+1|k) + K[\underline{z}(k+1) - H\hat{\underline{x}}(k+1|k)] \qquad 2.18$$

$$K = P(k+1|k)H^T[HP(k+1|k)H^T + R]^{-1} \qquad 2.19$$

$$P(k+1|k+1) = [I-KH]P(k+1|k) \qquad 2.20$$

where

K is the Kalman gain matrix

I is the identity matrix

$(Z(k) - H\hat{X}(k|k))$ is the innovation

The criterion for choosing a Kalman gain matrix K is to minimize a weighted scalar sum of the diagonal elements of the error covariance matrix P. This is equivalent to minimizing the length ofthe estimation error [Gelb,1979:109].

The gain matrix for x, $K_x(k+1)$ is formulated by letting

$$S(k+1|k) = [HP(k+1|k)\ H^T + R] \qquad 2.21$$

then

$$K_x(k+1) = P_x(k+1|k)\ H^T S_x(k+1|k)^{-1}$$

$$= \begin{bmatrix} V_x(k+1|k) & C_{xx}(k+1|k) \\ C_{xx}(k+1|k) & V_x(k+1|k) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} S_x(k+1|k)^{-1}$$

$$= \begin{bmatrix} V_x(k+1|k) \\ C_{xx}(k+1|k) \end{bmatrix} S_x(k+1|1)^{-1}$$

$$S_x(k+1|k)^{-1} = [HP(k+1|k)H^T + R]^{-1}$$

which simplifies to

$$S_x(k+1|k)^{-1} = \begin{bmatrix} \dfrac{1}{V_x(k+1|k) + V_{xm}} \\ 0 \end{bmatrix}$$

thus

$$K_X = \begin{bmatrix} V_x \\ C_{xx} \end{bmatrix} \begin{bmatrix} \frac{1}{V_x + V_{xm}} \\ 0 \end{bmatrix} * \begin{bmatrix} \frac{V_x}{V_x + V_{xm}} \\ \frac{C_{xx}}{V_x + V_{xm}} \end{bmatrix}$$

In the notation adopted in this paper, the gains are

$$K_{\alpha x}(k+1) = \frac{V_x(k+1|k)}{V_x(k+1|k) + V_{xm}(k+1)}$$

$$K_{\beta x}(k+1) = \frac{C_{xx}(k+1|k)}{V_x(k+1|k) + V_{xm}(k+1)}$$

$$K_{\alpha y}(k+1) = \frac{V_y(k+1|k)}{V_y(k+1|k)+V_{ym}(k+1)}$$

$$K_{\beta y}(k+1) = \frac{C_{yy}(k+1|k)}{V_y(k+1|k)+V_{ym}(k+1)}$$

2.23

The updated state and covariance equations are now available by substitution and matrix operations. The updated state equations are

$$
\begin{aligned}
\hat{\dot{x}}(k+1|k+1) &= \hat{\dot{x}}(k+1|k) + K_{\alpha x}\,[\dot{x}_m(k+1) - \hat{\dot{x}}(k+1|k)] \\
\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K_{\beta x}\,[x_m(k+1) - \hat{x}(k+1|k)] \\
\hat{\dot{y}}(k+1|k+1) &= \hat{\dot{y}}(k+1|k) + K_{\alpha y}\,[\dot{y}_m(k+1) - \hat{\dot{y}}(k+1|k)] \\
\hat{y}(k+1|k+1) &= \hat{y}(k+1|k) + K_{\beta y}\,[y_m(k+1) - \hat{y}(k+1|k)]
\end{aligned}
\qquad 2.24
$$

and the updated covariance equations are

$$
\begin{aligned}
V_x(k+1|k+1) &= V_x(k+1|k)\,[1-K_{\alpha x}(k+1)] \\
C_{x\dot{x}}(k+1|k+1) &= C_{x\dot{x}}(k+1|k)\,[1-K_{\alpha x}(k+1)] \\
V_{\dot{y}}(k+1|k+1) &= V_{\dot{y}}(k+1|k)\,[1-K_{\alpha\dot{y}}(k+1)] \\
C_{y\dot{y}}(k+1|k+1) &= C_{y\dot{y}}(k+1|k)\,[1-K_{\beta\dot{y}}(k+1)] \\
V_{\dot{x}}(k+1|k+1) &= V_{\dot{x}}(k+1|k)-K_{\beta x}(k+1)\,C_{x\dot{x}}(k+1|k) \\
V_{\dot{y}}(k+1|k+1) &= V_{\dot{y}}(k+1|k)-K_{\beta y}(k+1)\,C_{y\dot{y}}(k+1|k)
\end{aligned}
\qquad 2.25
$$

To complete the filter equations it is necessary to examine how the tracker is initialized. As previously noted, the elements of the covariance matrix are initialized to

$$
P_x(0) = \begin{bmatrix} V_{xm} & \dfrac{V_{xm}}{\Delta} \\ \dfrac{V_{xm}}{\Delta} & \dfrac{2V_{xm}}{\Delta^2} \end{bmatrix}
$$

This is reasonable since the only values of any real certainty are the statistics of the sensor data. For the same reason the target position is initialized to the report position.

Velocity initialization depends upon whether a two or one point initialization is used. For a two point initialization the

initial track velocity components and heading are determined from the two measurements. If a one point initialization is used, then prior information about target heading and velocity must be assumed. The heading and velocity estimates are independent in a one point initialization therefore the off-diagonal terms of the covariance matrix are set to zero [Kalata,1976].

The Kalman filter provides the minimum variance estimate of the target's states. As such, it is the optimal estimator. Unfortunately, the Kalman filter has no capacity to select, from a set of reports, that measurement caused by the target of interest.

In the first chapter, the various approaches were introduced that form the historical and theoretical foundations for RAHTT. The most significant of these, multiple model estimation, must be examined further before discussing the actual algorithm.

Multiple Model Estimation

Introduction

The stated purpose of this dissertation is to develop the algorithmic framework for a tactical tracker suitable for the environment described in the previous chapter. Such an algorithm must be capable of utilizing numerous sensors, responsive to both linear and nonlinear flight paths, and consistent with airborne computing systems.

The theoretical foundation for the algorithm developed in the sequel is MMEA, introduced in chapter I. The mathematical basis for multiple model estimation and the transition to adaptive hypothesis

testing are presented in this section. The mathematical details of MMEA are quite intricate and a full presentation is beyond the scope of this work.

Before discussing MMEA more formally, it is useful to give the reader an idea of where the presentation is going. Consider a target whose motion is described by one of a set of possible linear systems. The problem is to decide which of the possible systems most nearly represents the true motion of the target.

If a series of measurements is taken of the target motion and filtered using a Kalman filter appropriately developed for each of the possible linear systems, then, as the number of measurements increases, the Kalman filter matched to the correct system should be closer to the data than the mismatched systems.

In other words, by using several Kalman filters, each matched to one of a set of possible linear systems and each filter driven by identical measurements, it is possible to select that system that most nearly describes the true system.

Consider a similiar case where the target motion may be nonlinear and high order. By considering each of the components of motion, it is possible to derive linear equations describing each component. Kalman filters can be used to estimate each component by driving the filters with the identical measurement sequence and calculating online the probabilities of each component. In this way a nonlinear system can be estimated by a bank of linear filter.

With this basic introduction complete it is possible to give a more formal discussion. The development in the next section closely

follows the tutorial work of Athans and Change[1976] supplemented by the dissertation of Yoram Baram[1976].

State Estimation With Unknown Parameters

The basic concept of MMEA is to construct a bank of Kalman filters with each filter matched to a possible parameter vector value. The Kalman filters generate state estimates that are combined using the posteriori hypothesis probabilities as weighting factors. If one of the selected parameter vectors coincides with the true parameter vector, MMEA gives the minimum variance estimate of both the state and parameter vector.

MMEA is concerned with the problem of selecting, from a specified set of models, the "best" model by using a set of observations to mathematically describe a physical phenomenon. The relationship between the model and the observations is uncertain and must be expressed in a probabilistic framework. The model set can be specified in terms of a parameter set such that to each parameter there corresponds a model and vice versa. The problem of model selection can then be defined as a parameter estimation problem. It should be noted that the true parameter cannot, in general, be assumed to belong to the specified parameter set.

If a parameter vector is denoted by $\underline{\nu}$ then the standard state equation can be rewritten as a stochastic dynamic system whose dynamics depend on $\underline{\nu}$

$$\hat{\underline{x}}(k+1) = \Phi(\underline{\nu})\hat{\underline{x}}(k) + \Gamma(\underline{\nu})\underline{w}(k) \qquad 2.27$$

Similiarly, the measurement equation can be written as

$$\underline{Z}(k+1) = H(\Upsilon)\,\underline{x}(k+1) + \underline{y}(k+1) \qquad 2.28$$

$\underline{\Upsilon}$ is a vector whose elements represent the key parameters. These elements are, in general, only approximately known; however, in any practical application, reasonable information about the nominal value(Υ_0) and the degree of uncertainty is available from engineering studies, simulation, etc.

$\underline{\Upsilon}$ can be viewed as a random vector with all prior information captured in its prior density function $p(\Upsilon)$. The confidence in the estimation of $\underline{\Upsilon}_0$ is communicated to the mathematics by the prior covariance matrix

$$Cov[\underline{\Upsilon}:\underline{\Upsilon}] = E\{(\underline{\Upsilon} - \underline{\Upsilon}_0)(\underline{\Upsilon} - \underline{\Upsilon}_0)^T\}$$

In filtering measurements Z(k) the objective is to obtain, in real time, estimates of the actual true state $\underline{x}(k)$. The state estimate is denote

$$\hat{\underline{x}}(k|k) \qquad 2.29$$

and the state estimation error is denoted by

$$\tilde{\underline{x}}(k|k) \triangleq \underline{x}(k) - \hat{\underline{x}}(k|k)$$

Accurate state estimation is affected by the uncertainty in modeling the true values of the parameter vector $\underline{\Upsilon}$ by its nominal $\underline{\Upsilon}_0$. As these errors become more significant, the performance of the Kalman filter begins to deteriorate. If the major parameter uncertainty is in the state dynamics rather than the measurement equation, then the

increased parameter uncertainty is reflected in the calculation of the one-step prediction estimate $\hat{\underline{X}}(k+1|k)$. It is possible to overcome this uncertainty by relying more on the data, especially when measurement ambiguity is minimized. However, if the observations are seriously misleading, as with jitter and code reliability, the resultant state estimate can be seriously in error with filterdivergence probable.

As the uncertainty in $\underline{Y}$ increases, even with well behaved data, using an extended or adaptive Kalman filter will give unsatisfactory performance[Athans and Chang,1979:7-17].

The effects of large parameter uncertainty on the state estimation algorithm can be studied by subdividing the parameter space into regions.

Subdivision of the Parameter Space

A major concern of this dissertation is the state estimation of targets when uncertainty arises as to the proper report/target pairing. The proper framework for studying this uncertainty consists of an underlying probability space and a separate parameter space, of which the true parameter may not be a member. Likelihood ratios and maximum likelihood estimates are naturally defined in this framework. In a Bayesian framework, the inherent assumption is that the true parameter is a member of a given parameter space, i.e., the parameter space is part of the underlying sample space. Thus, while the Bayesian hypothesis is assumed in the definition of Bayesian estimates, the analysis of these estimates, as well as the maximum

likelihood estimate, is performed using the underlying nonBayesian estimate [Baram,1979:11-12].

Since in most physical problems some prior knowledge is available about the ranges of the parameter vector elements, it is possible to find a subset of the parameter space representing all reasonable values that $\underline{Y}$ can attain. A finite set of parameter values is denoted $\underline{Y}_0, \underline{Y}_1, \ldots, \underline{Y}_n$. For each $\underline{Y}_i$, redefine

$$\begin{aligned} \Phi(\underline{Y}_i) &\triangleq \Phi_i \\ \Gamma(\underline{Y}_i) &\triangleq \Gamma_i \\ H(\underline{Y}_i) &\triangleq H_i \end{aligned} \qquad 2.31$$

with the understanding that all matrices in 2.31 can be time varying.

It is then possible to rewrite 2.27 and 2.28 as

$$\hat{\underline{x}}(k+1) = \Phi_i \hat{\underline{x}}(k) + \Gamma_i \underline{w}(k) \qquad 2.32$$

$$\underline{z}(k+1) = H_i \underline{x}(k+1) + \underline{v}(k+1) \qquad 2.33$$

resulting in a class of N distinct linear stochastic dynamic systems.

$\underline{Y}$ is a discrete random vector which can be modeled by a set of hypotheses $\{H_1, H_2, \ldots, H_n\}$ denoting a set of events and H, a scalar variable, representing a hypotheses variable. The interpretation attached to the event $H = H_j$ is $\underline{Y} = \underline{Y}_j$ [Athans and Chang, 1976:23]. Then if

$$Z(k) = \{\underline{z}(1), \underline{z}(2), \ldots, \underline{z}(k)\}$$

$$\begin{aligned} A_i(k) &\triangleq P_r(H = H_i \mid Z(k)) \\ &= P_r(\underline{Y} = \underline{Y}_i \mid Z(k)) \end{aligned}$$

Given the measurement set $Z(k)$, $A_i(k)$ is interpreted as the probability that the i^{th} hypotheses (the i^{th} model) is the correct one.

A recursive relationship describing the dynamic evolution of the posterior probabilities $A_i(k)$ is derived by Athans and Chang [1976:26-35] based upon the innovations property of the Kalman filter, standard statistical relationships, and estimation theory.

A bank of N Kalman filters is constructed with each filter using a specific set of matrices associated with $\underline{Y}_i$. Each filter is driven by the same measurement sequence. From each Kalman filter mean $\hat{\underline{x}}_i(k|k)$ and covariance matrix $P_i(k|k)$, the Gaussian density function $p(\underline{x}(k)|H_i,Z(k))$ is calculated. The overall state estimate is the probabilistically weighted average, by the posterior (hypotheses) probabilities of $A_i(k)$ of the state estimates generated by all of the N Kalman filters.

Conclusions

If there exists some form of statistically consistent measurement error, then the residuals of the correct Kalman filter model will be less than those of the mismatched model. As measurements are processed, the correct probability $A_i(k)$ will increase, while the mismatched model probabilities will decrease.

MMEA provides an optimal nonlinear filter for an estimation problem with measurements coming from one of a set of possible linear systems. It asymtotically identifies which of the possible linear systems is the actual system and converges to the optimal Kalman filter for that system[Kervian and Sandell,1979:5].

MMEA provides a sound theoretical foundation for a multitarget tracker and, could, in fact, be formally extended to the multiobject/multiple target case. As a practical matter, it is

necessary to develop an algorithm less rigorous mathematically but more intelligent; one capable of adding and deleting hypotheses in reaction to the evolving situation.

Multiple Target Hypothesis Testing

A key point of multiple model estimation is that a nonlinear system can be described by a bank of linear Kalman filters. In the multiobject tracking problem, the set of linear systems generating the observations is the set of targets to be tracked. Because the mathematics of MMEA is theoretically sound and, further, is capable of resolving nonlinear as well as linear tracks, this work is limited to the linear multitarget problem.

The problem of multitarget tracking by adaptive hypothesis testing revolves around two issues:

1. Prudent report/track selection (association)
2. Mathematical selection of the most likely target trajectory.

The mathematics of these two issues is examined in this section with algorithm details delayed until chapter III.

Association

As previously defined, association is the selection of reports likely to have been caused by the target of interest. In associating reports and tracks, the set of report/track pairs selected for further consideration must be the smallest set possible while maintaining a high probability of including the correct pair.

Association is accomplished via test on report range, azimuth, and range rate (if available). A report is associated with a track if all applicable tests are passed. Appendix B shows the derivation of the association equations.

It is necessary to arrive at a unique report/track pair (correlation) for reports and tracks having multiple associations. This is accomplished by postulating target motion to account for the associating reports and applying a maximum likelihood test.

Maximum Likelihood

The Kalman filter equations already specified allow a likelihood function to be calculated for each data association. Under the maximum likelihood approach, the hypothesized target motion is evaluated based on how well it fits the date.

From Kalman filter theory, the likelihood of the innovations

$$\underline{V}_j(k) \triangleq \underline{Z}_j(k) - \hat{\underline{X}}_j(k+1|k) \qquad 2.34$$

for target j is

$$P(\underline{V}_j(k),\underline{V}_j(k-1),\ldots\underline{V}_j(1)|\hat{\Omega}_n) = 2\pi^{-k/2}\Pi^k|S_j(k)|^{1/2}$$
$$\exp\{-1/2\sum_{i=1}^{k}\underline{V}_j^T(k)S_j(k)^{-1}\underline{V}_j(k)\} \qquad 2.35$$

$\hat{\Omega}_n$ = as a particular measurement sequence

$L_j(k)$ is defined as the negative log-likelihood function of 2.28. It is calculated recursively by [Smith and Buechlar,1975]

$$L_j(k) = L_j(k-1) + \frac{\ln|S_j(k)|}{2} + \frac{\underline{V}_j^T(k)S_j(k)^{-1}\underline{V}_j(k)}{2}$$
$$+\ln 2\pi \qquad 2.36$$

Simplifying

$$L_j(k)=L_j(k-1)+\frac{\ln|S_j(k)|}{2}+\frac{V_j^T(k)S_j(k)^{-1}V_j(k)}{2}+$$

$$\ln 2\pi$$

$$=L_j(k-1)+\frac{\ln}{2}\left\{\frac{1}{(V_x+V_{xm})(V_y+V_{ym})}\right\}+$$

$$\frac{\ddot{x}_I^2}{2(V_y+V_{ym})}+\frac{Y_I^2}{2(V_x+V_{xm})}+\ln 2\pi \qquad 2.37$$

where

$$x_I=x_m(k+1)-\hat{\ddot{x}}(k+1|k)$$

$$y_I=y_m(k+1)-\hat{\ddot{y}}(k+1|k)$$

and the time indexes are dropped in 2.37 for notational ease.

This chapter has studied the mathematics of the RAHTT algorithm to be developed in the sequal. As demonstrated, RAHTT has, as its basis, solid mathematical theory. There is, however, considerably more intelligence to this algorithm than implied by the mathematics. The concepts and algorithm details are presented in the next chapter.

CHAPTER III

DESCRIPTION OF THE RAHTT ALGORITHM

Introduction

The details of the Real-time Adaptive Hypothesis Testing Tracker (RAHTT) are presented in this chapter. The intent of this chapter is to discuss informally the concepts of each of the algorithm's functions (RAHTT is formally presented in appendix A).

The overall concept is to split each cluster of data associations into hypothesized tracks. A likelihood function is calculated for each postulated track based on the innovation sequence. After a low number of samples (three or less) the most likely data correlations are determined. Based upon the type and history of radar correlations and the frequency of correlating, code matching IFF returns, if available, the most likely track for each of the known targets is selected.

The algorithm divides naturally into three main blocks: i) report/track correlation, ii) hypothesis generation and deletion, and iii) trajectory verification. A key development in this paper is the three tier correlation process. It is imperative, for efficient tracking, that a report associate with as few reports as possible while maintaining a high probability that the correct report is

included since a large number of data associations can overwhelm the computing system.

Correlation

Correlation in RAHTT is defined as an associated one-to-one report/track pair. It serves as the algorithm's best estimate of the origin of the report and is the key to successful multitarget tracking. Often the targets are widely distributed and it is unnecessary to resort to such an expensive correlation process as a maximum likelihood selection. Even in "dense" traffic it is often possible to find unique report/track associations.

In the suboptimal data correlation used in this algorithm, correlations occur as the result of i) a unique radar report/track association, ii) the mostly likely data association as the result of the hypothesis testing framework, or iii) the most likely pairing resulting from hypothesis construction rules for the terminal level. No effort is made to modify the covariances to account for the probability of an incorrect correlation.

The unique association of a radar report/track is highly dependent upon the density of nearby traffic, false reports, and the geometry of other tracks. The association tests described in chapter II are very discriminating and will produce a high percentage of the total correlations. Because the association windows provide the report that most nearly fits the hypothesized trajectories, whenever a unique report/track pair is found, that pair is correlated and the

report, "fully identified", a term to be explained in the next section.

Radar report/track associations that are not unique must be resolved via the hypothesis testing framework of the next section. Each data association within the cluster is tested for likelihood against the postulated tracks. The most likely set of data associations based upon the measurement innovation sequence are termed correlated pairs.

IFF reports are associated in a similiar manner. Functionally, IFF reports attempt to verify target trajectories by confirming or denying the presence of the correct code matching IFF reply within a hypothesized target's association window. This is discussed in more detail in a later section.

Unique radar report/track associations provide a substantial number of the total correlations. By utilizing these correlations outside the more computationally demanding hypotheses testing scheme of the next section, RAHTT avoids generating unnecessary tracks and, more significantly, is able to accelerate the overall track selection process.

Hypothesis Generation and Deletion

Informal Discussion

The basic approach used in this algorithm is to generate a set of data-association hypotheses to account for all associated pairs that are not unique. These data-associations generate hypothesized tracks in a measurement oriented tree [Reid, 1979] contrasted with the

target-oriented hypotheses of Bar-Shalom[1975]. This allows more efficient space utilization, pruning techniques, and track initiation (see Keverian and Sandell, 1979:11] and [Reid, 1979]. The hypothesized tracks, generated tracks, represent the most likely target paths based upon prior knowledge of the number and states of the established tracks.

Data associations are mutually exclusive. Before a hypothesis is formed for track j, a check is made to ensure that track j is not associated with more than one measurement in the current dataset. If j is associated with a measurement but does not exist on a particular hypothesis then a new track is generated representing a trajectory from the last hypothesized data association.

Experience has shown that the orientation of the measurement hypothesis tree is difficult to understand. Since the effectiveness of RAHTT depends directly on the algorithm's ability to manipulate the date-associations in the tree, an example is warranted.

Consider figure 3.1a where established tracks T1 and T2 associate with reports A, B, and C. There are six feasible tracks represented by the lines connecting the targets and the reports. The date-associations and the hypothesized trajectories are represented by the hypothesis tree figure 3.1b.

In a measurement oriented hypothesis tree each level represents a report and the nodes of the tree, the tracks that associate with that report. Level A indicates that tracks T1 and T2 associated with report A plus a false alarm hypothesis.

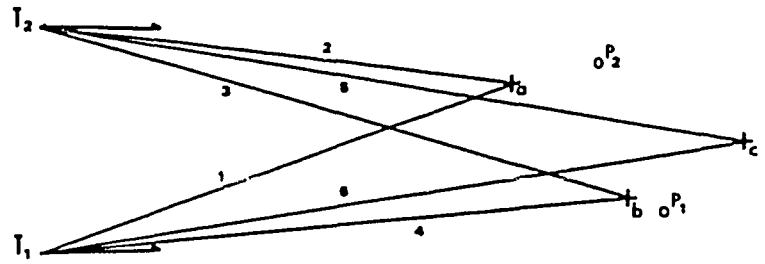


FIGURE 3.1a Feasible track example

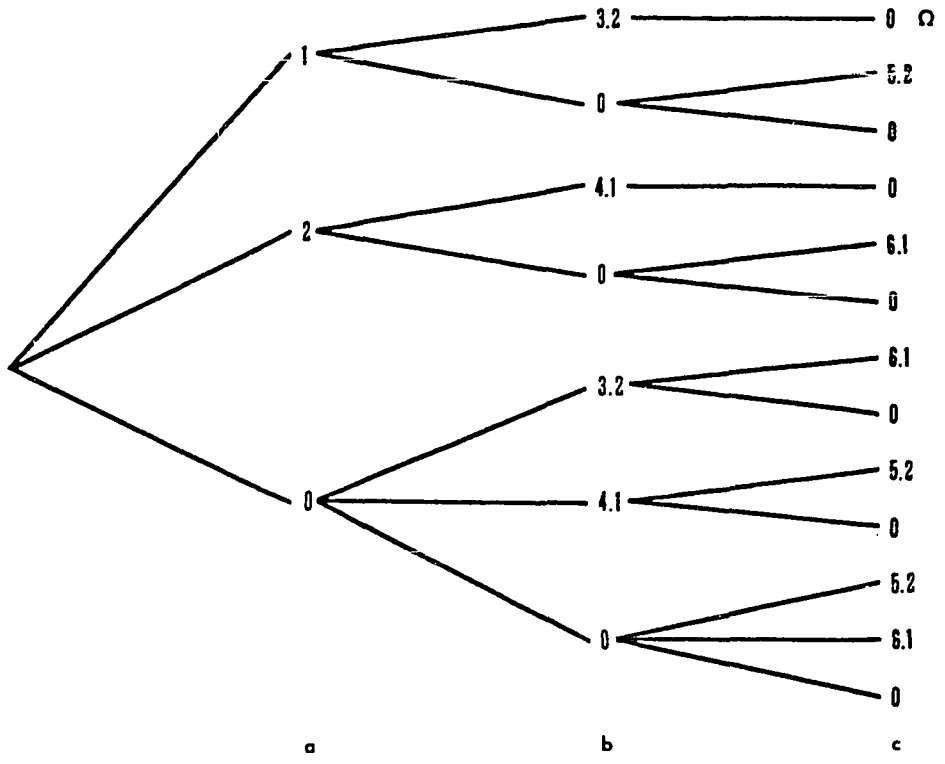


FIGURE 3.1b Example hypothesis tree

NOTE: Generated tracks are indicated by X.Y.Z where Z is the parent track and Y and X are successive generations.

Level B illustrates the mutually exclusive rule fordata-associations. Note that tracks T1 and T2 plus the false alarm hypothesis still associate with report B however new tracks have been generated to account for the hypothesized trajectory (see note on figure 3.2) and that not all associating tracks appear on all hypotheses.

The generated tracks reflect the physical reality that if a track was hypothesized to account for a data-association on one level of the tree then the same track cannot have subsequent associations in the same measurement dataset. It is important to keep in mind that the hypotheses are tracks for some known target. Hence 1, 4.1, and 6.1 reflect different possible tracks for target 1.

In a similiar way, once a track is assumed as the origin of a report on a hypothesis, then that track cannot be the assumed origin of another report in the same observation dataset. For example, Ω , represents the fact that at level A, T1 was the assumed origin; at level B, only track T2 (or a "spun" track from 2) can be hypothesized; and level C represents the fact that there are no tracks left to hypothesize as the origin of report C. To do otherwise would indicate that a target can, at the same instant, fly multiple paths.

It should be obvious that the number of generated tracks and the number of trajectories can become excessive if allowed to grow unbounded. In an airborne system, efficiently limiting the growth of hypothetical tracks is a serious concern. The failure of most multiple-scan correlation algorithms is their inability to deal with

the rapidly growing number of track trajectories, especially when missing reports and false alarms complicate the association process.

Tree Maintenance

In most N-scan correlation algorithms, a measurement is identified after a constant number of scans (N). It is clear that some measurement are readily identified after only one additional sample while others can never be resolved. At a high level, an adaptive hypothesis tree is constructed that identifies report/track pairs consistent with the severity of the initial and subsequent ambiguous report/track pairs. The tree is bound, first, by the breadth of the search and second, by the depth of the search.

The hypothesis tree functions primarily as a FIFO buffer with the early reports toward the root. As a report is resolved it is pushed out of the tree. A measurement remains in the tree until i) it is identified (correlated), ii) a maximum number of sampling periods have passed or iii) room is needed for new measurements.

Initially the tree is empty. When reports are entered into the tree, the FIFO is allowed to grow until it either identifies the first report or encounters one of the bounds. Selection of the bounds is critical to algorithm effectiveness and to the control of the tree's overall size. In selecting the bounds there are important tradeoffs in computational demands, optimal hypothesis selection, space consideration, and the man/machine interface.

Usually, the first bound encountered is the breadth of the tree. Unlike Kervian and Sandell[1979] or Reid[1979], the maximum

breadth of the tree is not fixed, except for an absolute upper limit, but is adaptive to the number of tracks not uniquely paired during a sample.

When the maximum breadth of the tree is reached and there are more tracks to enter, the most unlikely hypothesis on the new level is deleted unless it is the only remaining branch. Often, as a result of this trimming, a report will have all nodes assigned to the same track. Such a report is termed fully identified. A fully identified report is the algorithm's final decision on report/track pairing (correlation). When, in the process of building a terminal level, a report becomes fully identified, it is dropped from the tree and the appropriate track updated to reflect this correlation. This selection of the most likely association at the terminal level is often very accurate; especially when unforced. More generally, this type correlation is in reaction to a cluster of data associations that is too large for the hypothesis tree to accept and maintain its prescribed bounds. Hence, as the tree pushes closer and closer to its breadth bound, fewer different associations can be retained. This approach is quite consistent with the overall adaptive intent of RAHTT.

The breadth of the tree controls the number of hypotheses considered. To provide for a maximum number of useful hypotheses, there are two maximum breadths, both adaptive to the number of tracks and severity of the cross-associations entering the tree. The expansion bound, the larger bound, enables more hypotheses to be considered during the addition of levels to the tree. Once the

expansion of the tree is complete for all measurements in the sample, the likelihood of all remaining hypotheses is calculated, and the tree is reduced to the trim bound, the lower bound for breadth. This branch and bound method corresponds to a breadth first expansion of the tree resulting in the growth of the tree by set of hypothetical data associations. Hypotheses with low likelihood functions are discovered and pruned.

After all levels of the tree have been completed, the first level is examined to see if it is fully identified. If the report is fully identified, it is pushed from the tree and the new first level is examined. This procedure continues until all fully identified reports are pushed from the tree.

The maximum depth of the tree is a function of the maximum time delay permitted before forcing a pairing and a function of the number of reports having ambiguious associations. If the depth of the tree exceeds the maximum allowed depth, then the first level is identified by using the most likely hypothesis and is pushed from the tree.

The identification of a level of the hypothesis tree during the reduction from the expansion bound to the trim bound may mean that the set of hypothesized tracks can be reduced. Referring again to figures 3.1a and 3.1b, if report A is identified as track T2 then obviously the hypothesized trajectory for T1 to A is invalid. Moreover, all subsequent hypotheses based on the existence of a track T1 as initially defined, i.e. T1 to report A, must have been determined as unlikely (or less likely) and must have been dropped

from the tree. In place of the old T1, the next "spun" track of T1 is promoted to T1. In this example, T4.1 becomes T1. All generated tracks surviving this initial pruning are termed potential tracks.

The hypothesis tree determines the most likely correlation of ambiguious report/track associations. It does this by reacting to both the number and complexity of the data associations. The more ambiguious the data the wider the tree; the more voluminous the date, the deeper the tree. As the bounds are approached, the tree becomes more selective in those date associations it considers until it is forced to declare the most likely association the correlation.

The hypothesis tree makes only initial decisions about target trajectories. The ultimate decision about a target's true path is left to the trajectory verification function.

Trajectory Verification

Radar Verification

The three tier correlation scheme results in multiple tracks with correlations occurring at different times. In order to account for the probability that the correlated pair is incorrect, conjectural tracks surviving the initial pruning become potential tracks and are allowed to compete for future correlations. Consider figure 3.2 representing the hypothesis tree from figure 3.1b after the initial pruning and table 3.1, the corresponding track file.

A potential track must correlate within a prescribed time period, the potential track delay, or be dropped. Generally the

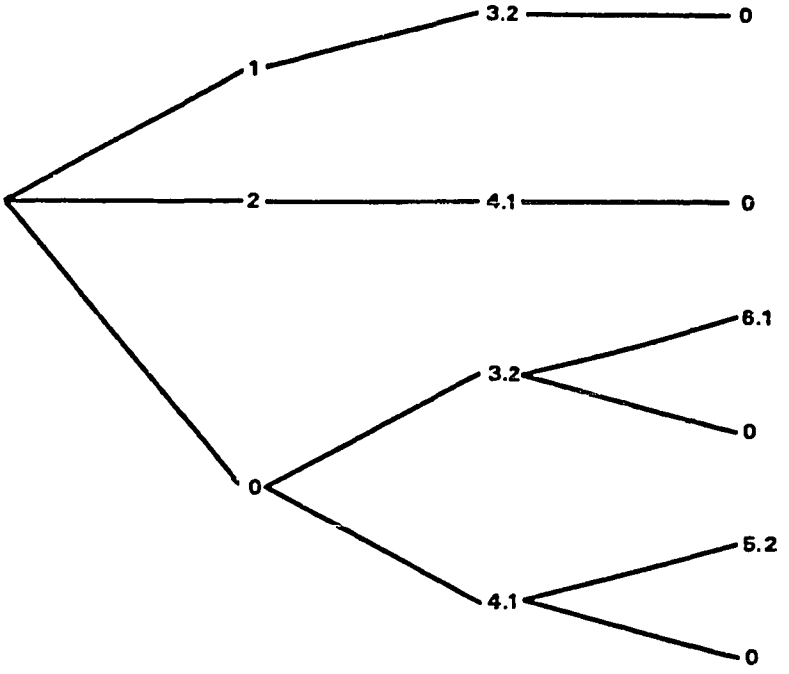


FIGURE 3.2 Example hypothesis tree after pruning

TRACK FILE

ESTABLISHED	POTENTIAL	TENTATIVE
1	3.2	
2	4.1	
	5.2	
	6.1	

TABLE 3.1

minimum is the time delay required by the hypothesis tree, as a function of maximum depth, before forcing a correlation.

A potential track that correlates with a report by any means is promoted to a tentative track. Tentative tracks are subjected to additional tests for i) convergence with the parent, ii) quality, and, iii) duplicate trajectories.

Tentative tracks will often converge to their parent track. Tests are made on the states of the tentative tracks and their parent tracks. Those tentative tracks found "close" to the parent track are dropped as duplicate trajectories.

Tentative tracks surviving the similiar trajectory test are then compared with their parent track for quality. Simulation studies show that each of the three correlations have a relatively stable success rate. Track selection is thus a probabilistic problem. Each of the three correlations is weighted in proportion to its mean success rate. This weighting is the track's quality.

The quality of the parent track and that of the tentative track are compared and if the quality of the tentative track exceeds that of the parent track by a predefined amount, the parent track assumes the states of the tentative track and the tentative track is dropped.

If the quality of the tentative track does not exceed the parent's by the required amount within a specified time, the tentative track delay, the parent track and the tentative track qualities are compared and the track with the higher quality becomes the established track.

Once the quality tests have been completed, the tentative tracks are compared for duplicate trajectories. Tracks with the lowest quality are dropped. When the quality is equal, the quality of the parent track is compared and the tentative track of the parent with the highest quality is dropped. If this test fails, neither of the tentative tracks is dropped.

Unfortunately, there are many track geometries that can cause track switch or a track convergence. Figures 3.3, 3.4, and 3.5 illustrate one simple example. Unlike some tracking systems, RAHTT has the added capability to use realistic IFF data to correct target trajectories.

IFF Verification

IFF verifies that the hypothesized track is in the proper location by attempting to associate the track with the proper code and the IFF report. If the report and track associate and no other tracks associate with the report, then the IFF report is used for smoothing. If the report associates with more than one track, at least one of which has the proper code, then the hypothesized track is assumed correct. If the report does not associate with a code matching track, then tests are performed to determine whether or not the report is jittered.

IFF jitter is manifested as an azimuth only distortion and can be detected by an association test for azimuth. The tests for jitter are necessary to prevent the algorithm from detecting a missed

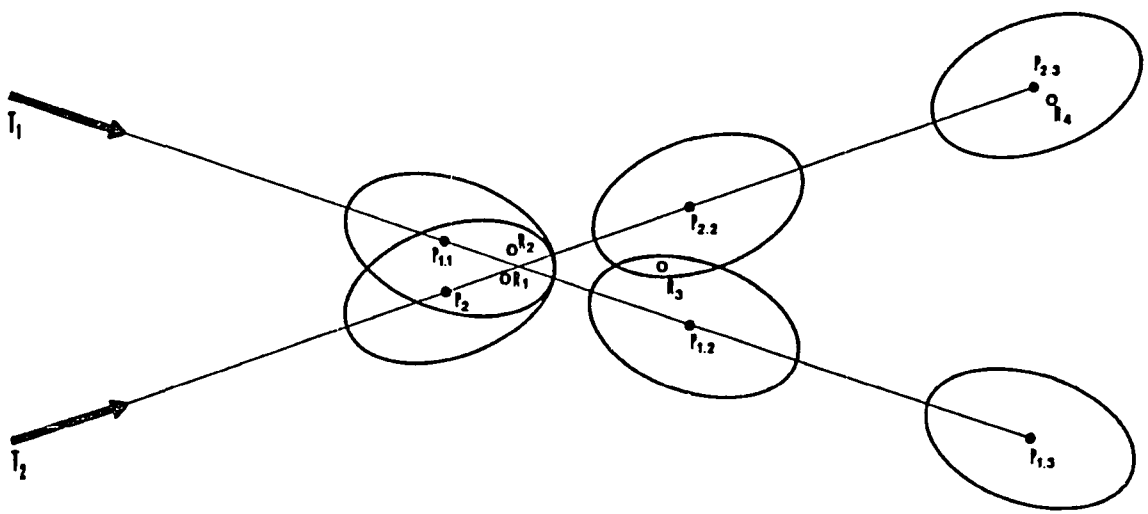


FIGURE 3.3 Example track convergence

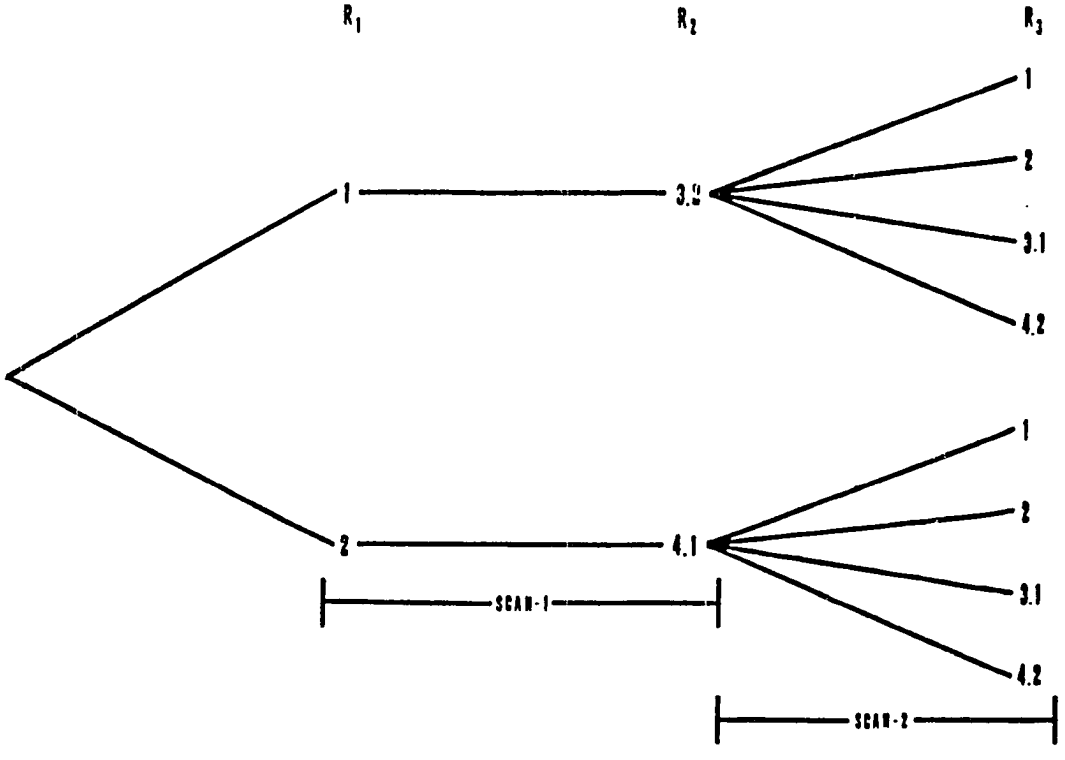


FIGURE 3.4 Example convergence hypothesis tree

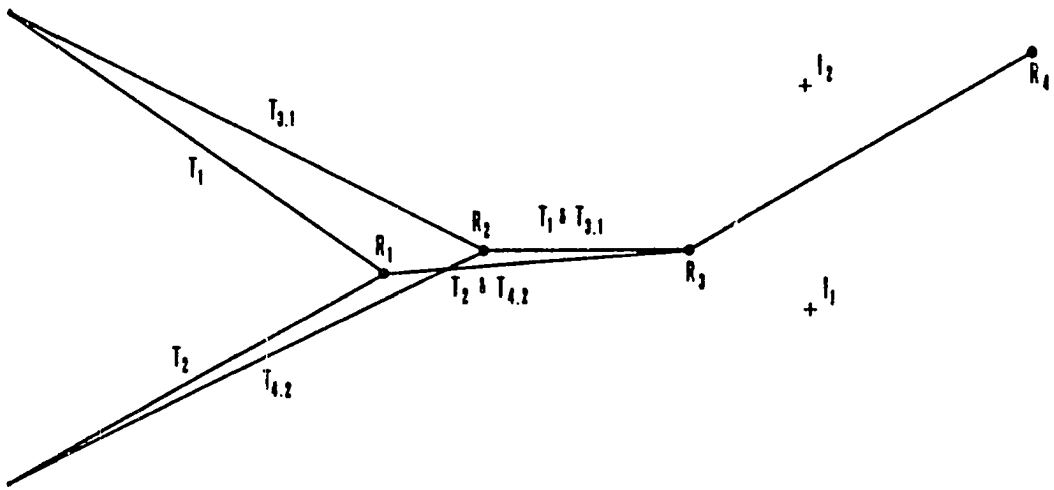


FIGURE 3.5 Converged feasible tracks

association and taking corrective action on a jittered report. When a report fails to associate in azimuth only, it is a probable jittered report.

It is possible for switched or converged tracks to be reflected as azimuth only association failures. To prevent these conditions from giving continuous false jitter indications, consecutive azimuth only association failures are tested for consistency. If the reports are found to be within a reasonable orientation, then a new track is generated to account for the IFF path. The new track is subject to the same upgrading rules as radar generated tracks.

Converged or switched tracks are often detected by IFF. When reports fail to associate in all dimensions, the report more accurately reflects the true track position than the hypothesized track. Under this condition, the report is used for smoothing.

It is important to remember that a target that is an in area where unique associations are not possible has probably generated several new tracks to account for the ambiguous data associations. When these tracks are tested against a code matching IFF report, each may fall into a different IFF test. The result is that the track closest to the true track will have the best IFF association and, consequently, will more accurately reflect the target's path. Tracks that do not have as strong an IFF association will eventually be dropped through the quality tests.

IFF gives RAHTT the capability to recover from incorrect correlations. The tests devised are simple yet very effective.

Further, the way IFF is used is consistent with the overall adaptive testing framework employed for radar reports and reflects the algorithm's ability to adapt on a more abstract level.

High Level Adaptation

Adaptive hypothesis testing, as employed in RAHTT, results in an algorithm that adjusts to its environment on a high level. The hypothesis tree and the Kalman filter statistics carry with them knowledge of the past. These features have memory and hence decisions are made not simply on tne data available, but on data and decisions of the past.

The hypothesis tree reacts to more than just the positions of the radar reports; it reacts to the volume of data by adjusting the breadth of its search. When the limits of the tree's ability to compensate for report density is reached, the algorithm compensates by selecting the most likely data association of the terminal level.

The flow of data associations through the hypothesis tree, the quality tests, and the Kalman filter statistics give this algorithm memory. It makes decisions on hypothesized track trajectories by observing the flow of unique radar correlations, most likely radar correlations, terminal level correlations, and IFF code matches. It adjust its hypothesized tracks over time to reflect the best estimate of the target's position based on this often conflicting data.

Algorithm Computational Aspects

As noted several times, RAHTT is designed to work in an airborne system where computer resources are limited. It is appropriate to briefly examine the impact on such a system.

The RAHTT algorithm is formally presented in appendix A. Careful study will show that there are two main computational and space intensive sections.

The first such section performs the generation and deletion of hypotheses via the hypothesis tree. The algorithm used for the simulation study contained a true tree structure however in any realistic application, using one of the present command and control languages (JOVIAL and ADA), a matrix structure would be far more efficient.

The cost of the hypothesis framework, whether tree or matrix, is directly determined by the size of that structure. The size of the hypothesis tree used by this algorithm is minimized by the measurement orientation, rapid elimination of unlikely hypotheses, and the correlation process which effectively keep the tree very small.

The second primary computational savings is achieved by judicious track growth. Each track beginning a process sampling must be tested for association against each report. Computationally, the association test require more computer time than any other aspect of the processing cycle. Since any optimization of the association tests applies equally well to any tracker, the measure of efficiency for association is the number of tracks generated by the algorithm.

Interestingly, the Kalman filter is called exactly once for each association. By noting whether a smoothing has been accomplished for this report/track pair, it is possible to eliminate duplicate calls to the filter. This rather surprising result means that once again the measure of the increased computer resources is directly proportional to the number of generated tracks. Limiting the number of tracks and the size of the hypothesis tree are precisely the approaches used in developing this algorithm.

Extensive simulation studies have verified the effectiveness of the Real-time Adaptive Hypothesis Testing Tracker. These studies are summarized in the next chapter.

CHAPTER IV

EVALUATION OF THE ALGORITHM

Introduction

This chapter documents, via simulation studies, the initial evaluation of the Real-time Adaptive Hypothesis Testing Tracker (RAHTT). The goal is to determine whether RAHTT provides sufficient improvement over a Nearest Neighbor style Tracker (NNT) to warrant further research and testing.

Comparing trackers with such differing underlying philosophies, as have RAHTT and NNT, can be a somewhat demanding task. NNT has the decided advantage, in a linear problem, of tending to fail to the correct path. When muiltiple associations occur, NNT selects the association that most nearly fits the predicted track position. This correlation process leaves some tracks without a correlation when the probability of detection is less than unity. If those tracks missing correlations were previously well established then the lack of a correlation has no effect on tracker performance. i.e., NNT must correlate with the wrong report before track divergence or switches can occur. It follows from this reasoning that performance is optimistic for an NNT optimized for linear motion.

Likewise, jitter has little effect on track continuity for a linear NNT. The NNT selects the IFF report within the association

window with the proper code. Only rarely does the proper set of jittered reports occur that can defeat the association process.

RAHTT, on the other hand, may generate a false track in response to a jittered report. Without some means of making reasonable decisions about the existence of a jittered reply, RAHTT would generate numerous, IFF supported, tracks.

This brief philosophical discussion is not designed to prepare the reader to accept poor results. Clearly RAHTT must make a substantial improvement in performance over NNT to justify future study. The standard selected to measure RAHTT performance is the performance of the optimized NNT over identical data. The measure for this evaluation is track continuity which is defined as the percentage of tracks completing a scenario within two miles of the actual target position and correlating on the correct report. The performance of the RAHTT is studied in terms of i) track continuity as radar detection rates vary, ii) track continuity as clutter (false alarms) and report density increases, and iii) track continuity for RAHTT in the presence of jitter with and without jitter logic. The performance is expressed, where appropriate, as a percentage of tracks completing a scenario. Care should be taken to note only the relative performance of the two trackers considered since the absolute performance is appropriate only for those scenarios and conditions of the test.

RAHTT is a complex algorithm with a multitude of parameters to be optimized. Experimental conclusions about all the parameters available in the algorithm and their interactions are left for future

study. The intention of of the rest of this chapter is to develop a basic understanding of a few of the parameters. This is a reasonable approach. Ultimately this algorithm, to be of practical use, must be extended to include maneuvering, multiple targets. It is during that extention that full study of the parameters is most appropriate.

Clearly the results of a tracking test are conditioned by the scenarios. To eliminate biasing the test results, fifty randomly generated scenarios were used as the test cases. A test case consisted of ten targets. Each target within a scenario was randomly placed within a prescribed area and given a random heading and velocity. Each scenario was then executed three times using standard simulation techniques; the mean of the three runs being the performance of the tracker against that scenario. The test results reflected in figures and tables for the remainder of this chapter were determined from the fifty scenarios.

Test Results

As discussed in chapter III, RAHTT has several aspects to its memory. Like other multiple-scan algorithms, RAHTT generates tracks via a hypothesis testing framework (tree) that accounts for multiple associations. These hypothesized tracks represent the most likely target paths based upon those statistics generated by examining samples of measurement/track associations. Important questions to answer immediately are how large must the hypothesis tree be for efficient correlation and how many samples of data must be gathered to provide reasonable performance.

Figure 4.1 shows the effect on correlation as the breadth of the hypothesis tree is increased. The trim factor (bound) is set as multiples of the number of reports and tracks in the cluster with the number of established tracks in the cluster forming the lower bound. Not surprisingly, the more hypothesized data associations in the tree the more likely a correct correlation.

The depth of the tree partially defines the time delay in determining the set of potential tracks and, computationally, determines the likelihood of the track set. Tests were conducted with three tree delays: i) a zero-scan delay which resolves the cross associations based on the current scan of data, ii) a one-scan delay which uses the present and previous data associations, and iii) a two-scan delay. Additionally, a zero-scan tree makes all decisions on track trajectory in the current scan.

The results verify the work of Singer, Sea, and Housewright [1974]. Their work showed that a two-scan memory tree performed nearly as well as an optimal, all-scans, memory. An important conclusion from that work and verified by the tests in this dissertation is that it is unnecessary to retain more than two samples of data to approach optimal estimation of the likely tracks and good performance can be obtained with a one-scan memory.

The intention of the three level correlation scheme, described in the previous chapter, is to reserve for the hypothesis tree those track associations not unique thereby limiting the number of reports placed in the tree. To minimize the depth of the tree a second function, that of quality, examines the set of all tracks and makes

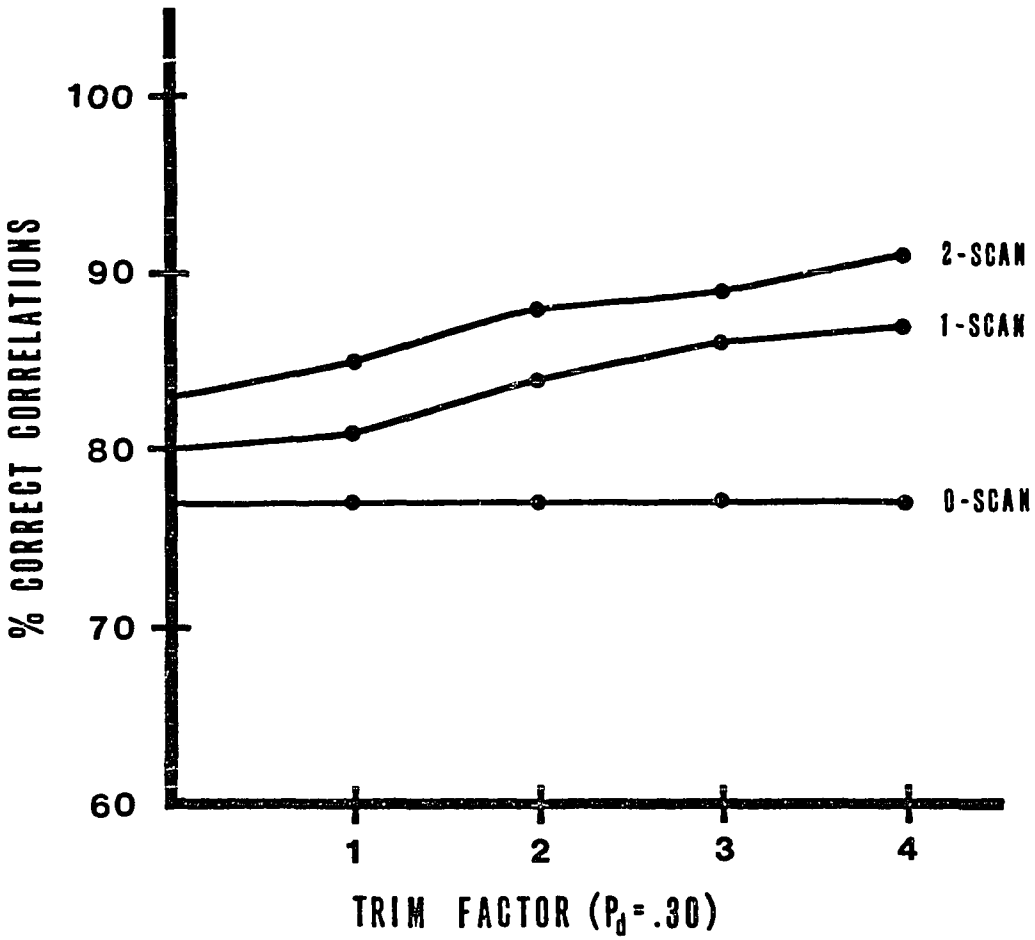


FIGURE 4.1 Correlation versus Trim Factor

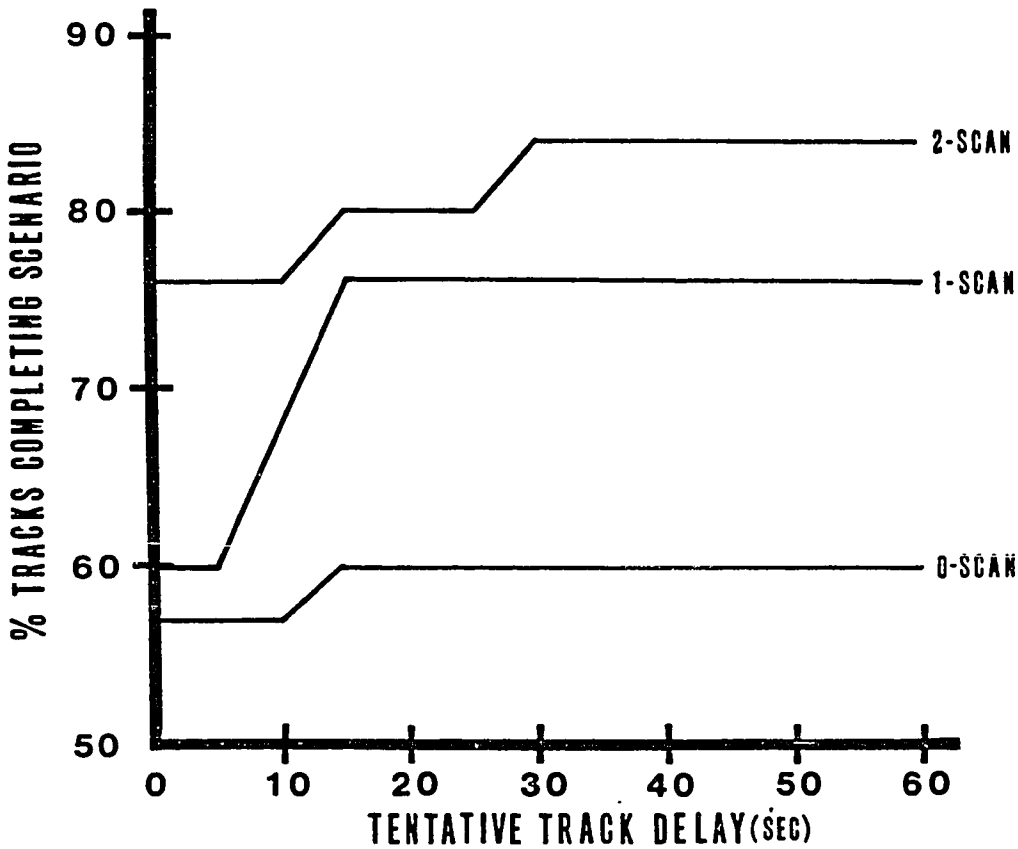


FIGURE 4.2 Success rate at varying tenative track delays

decisions based on the frequency and type correlation for a parent and its generations of tracks.

For the tests conducted in this dissertation the quality function proved a powerful tool for determining true target trajectory (see chapter III for a discussion of the quality function). Referring to figure 4.1, note that any two correlations for a one or two-scan tree will produce a probability of above 80%. Testing indicated that indeed a two correlation quality difference was sufficient to make most track decisions. This success rate verifies the results obtained by varying the tentative track delay. Extending the tentative track delay, the time allowed for the second correlation, beyond a certain point has minimal effect. The improvement over time reflected in figure 4.2 is a result of the few cases where two tracks were supported by correlations for a time before one lost correlation. More typically, the algorithm is able to make very rapid decisions on track quality keeping the number of tracks at a minimum.

The depth and breadth of the hypothesis tree and the quality function define the memory of RAHTT. This memory enables RAHTT to recover from wrong correlations where NNT fails. This approach demonstrates considerable improvement over NNT.

Performance Comparison

RAHTT substantially out performs NNT in all tests conducted. Figures 4.4 and 4.3 show the relative performance of the two trackers for varying radar detection rates with and without false alarms respectively. The curves for NNT compare favorably with that calculated by Bar-Shalom and Tse[1975].

Related to the performance is the computational burden required to achieve the improved performance. The reader needs to keep in mind that RAHTT resorts to the hypothesis tree only when there are cross associations. At any one time there are relatively few clusters of tracks utilizing independently constructed hypothesis trees [Trunk and Wilson, 1980]. Table 4.1 shows the increased track load at various points in the RAHTT algorithm as the number of reports per association window increases. Comparisons with the probabilistic data association filter (PDAF) of Bar-Shalom and Tse, [1975] are relatively meaningless. PDAF test results are for a single target with varying amounts of clutter while RAHTT is a multitarget tracker. Further, there are substantial differences in the way the statistics were gathered. Nevertheless, it is possible to get a <u>feel</u> for the computational requirements by examining table 4.1.

A true track splitting algorithm (TSA) generates a new track for each multiple association. Over twenty scans, the length of the scenarios used to develope the date for table 4.1, a TSA can generate up to 10^4 tracks (see [Bar-Shalom and Tse, 1975]). Table 4.1 shows the increased track load a TSA would generate in a scan using the association techniques of RAHTT.

While the track loading and performance of RAHTT are quite satisfactory the overall effectiveness can be substantially improved by adding IFF sensor measurements.

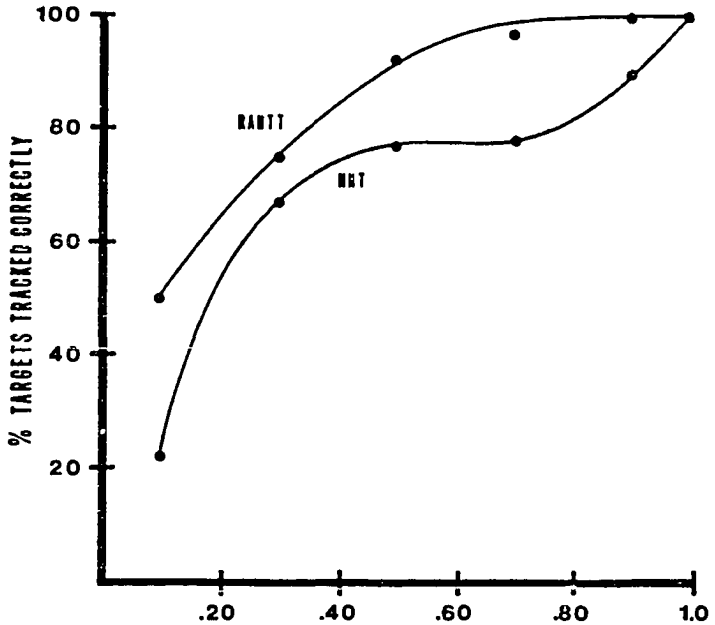


FIGURE 4.3 Comparison of RAHTT and NNT(uncluttered)

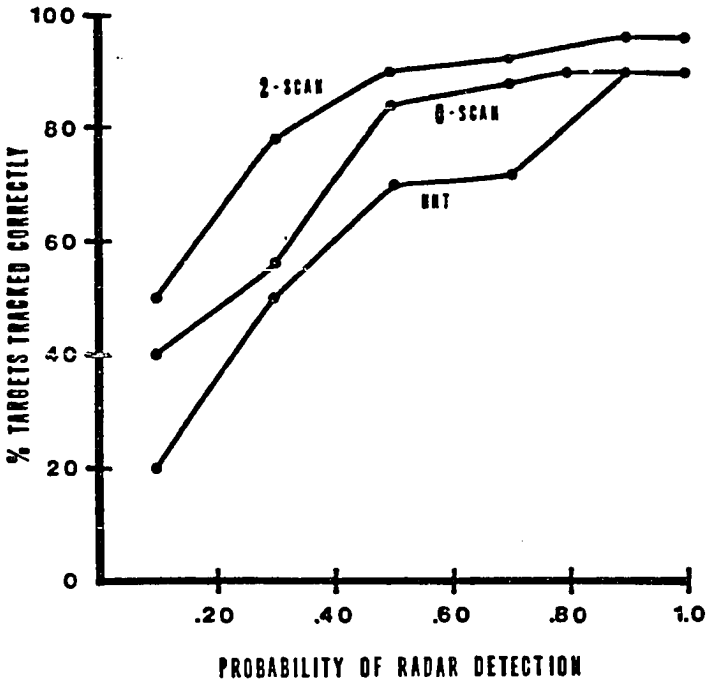


FIGURE 4.4 Relative Performance of Trackers
(in clutter)

APPROXIMATE PERCENTAGE INCREASE IN TRACK LOAD

Returns per Association Window	1	2	3	4
RAMTT				
Generated by hypothesis tree	—	110	280	500
Prior to pruning	—	14	76	128
After pruning	—	8	58	100
Generated by Track Splitting/scan	—	380	580	657
PDAF	—	400	600	800

TABLE 4.1 Increase in Track Load

IFF Performance

When unjittered IFF reports are processed by RAHTT the tracker performance approximates the probability of IFF detection. For the scenarios tested the performance is approximately 92% for all radar detection rates.

When jittered reports are included the overall RAHTT performance without the jitter detection and correction tests declines to approximately the radar performance. This happens because RAHTT only confirms track/IFF positional relationships without the jitter tests. Thus a jittered report cannot alter a track.

The jitter detection and correction tests improve the performance to approximately 88% regardless of radar detection. The success rate is lowered because there are always a few tracks still in the process of being corrected at the end of the simulation.

These IFF jitter tests and corrections are very simple and powerful tools for dealing with low to moderate jitter. Flight testing has been conducted on these jitter tests with excellent results.

Conclusions

RAHTT demonstrates a substantial improvement over NNT for tests on specific and random scenarios. The size of the hypothesis tree and the tracks generated are successfully limited by the three tier correlation process and the quality tests. Effects of IFF jitter are lessened by consistency checks on azimuth and orientation.

CHAPTER V

CONCLUSIONS AND AREAS FOR FUTURE RESEARCH

The Real-time Adaptive Hypothesis Testing Tracker developed in this dissertation is designed specifically for an airborne surveillance system. To meet the limited computational resources, the unique environmental demands, and the operational requirements, an original three level correlation strategy is used to reduce the size of the hypothesis testing tree. Probabilities of missed or incorrect correlations are accounted for by a quality system based upon the type and frequency of correlations. Various pruning criteria, most of which are found in the literature, are applied to further limit the growth of the hypothetical tracks.

IFF is used as a means of verifying and correcting incorrect trajectories. Consistency and orientation tests lessen the impact of light to moderate jitter.

RAHTT is a substantial improvement over a nearest neighbor tracker and preliminary results demonstrate its feasibility for potential operational implementation.

There remain several areas for study before actual implementation is possible.

The most obvious area is the extension to maneuvering flight. While the techniques and mathematics of Multiple Model Estimation insure that this extension is feasible, the actual implementation remains to be determined. Certainly the association process will have

to be modified to include a window modeled for aircraft maneuvers. Likewise, the Kalman filter will have to be changed to also model nonlinear motion. Neither of these problems is conceptually significant. It is important to note that there is no requirement that all tracks utilize the same filter or association mechanism. This would permit varying association and filtering approaches to differing data environments. One obvious approach would be to utilize a Kalman filter optimized for maneuvering flight only when there is a possibility the target is actually in a maneuver. At other times suitable performance could be obtained with a less extensive, and, consequently, less computationally demanding filter.

The Kalman filter is well suited for parallel computation. By dividing the track/report pairs into independent clusters, RAHTT has provided the foundation for a parallel or multiprocessor system. There remains a significant amount of work in this area, especially in the joining of previously independent clusters.

Finally, there remains the problems associated with an actual implementation. While great care was taken in the design and testing to account for known implementation problems, there are always unanticipated problems.

On a higher level, this dissertation demonstrates a need for a real time computational language that permits flexible, natural tree structures such as found in LISP. The addition of some of the artificial intelligence features of LISP would greatly enhance RAHTT's ability to deal with more discrete or unique information. IFF is one such attribute that could be of great value if it were more easily

bound to the hypothesized tracks. When the extensions are made to maneuvering flight, aircraft type and performance would also be immediate candidates. The potential benefits of adding more artificial intelligence to a computational language would permit RAHTT to be the truly flexible tracker desired.

The tracker developed in this dissertation represents a significant improvement in airborne, multitarget tracking over the present nearest neighbor algorithm. RAHTT provides improved track continuity in all areas of target and report density, probability of report detection, and susceptability to jittered reports. It accomplishes this improvement with modest increases in computational and space resources. Further, RAHTT was designed with potential software, computer hardware, and new sensor systems in mind to allow rapid and flexible upgrading.

REFERENCES CITED

Books

Allen, Robert W., Operational Benefit Analysis. The Boeing Company, 1980.

Athans, M. and C. B. Chang, Adaptive Estimation and Parameter Identification Using Multiple Model Estimation Algorithm. Techinical Note 1976-28, MIT, 1976.

Gelb, Authur, Applied Optimal Estimation, MIT Press, 1979.

Jazwinski, Andrew H., Stochastic Processing and Filtering Theory, New York, Academic Press, Inc., 1970.

Keverian, K. M. and Sandell, N.R., Multiobject Tracking by Adaptive Hypothesis Testing, LIDS-R-9559, MIT, Dec. 1979.

Skolnik, M. I., Introduction to Radar Systems, New York, McGraw Hill, 1980.

Articles

Baram, Yomar and N. R. Sandell, Jr. "An Information Theoretic Approach to Dynamic Systems Modeling and Identification," IEEE Trans. on Automatic Control, Vol. AC-23, Feb. 1968, pp. 61-66.

Bar-Shalom, Y., "Tracking Methods in a Multitarget Environment," IEEE Trans. Automatic Control, Vol. AC-23, Aug. 1978, pp. 618-626.

Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatic, Vol. 11, 1975, pp. 451-460.

Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering (ASME), Vol. 82D, March 1960, pp. 35-45.

Kalman, R. E. and Bucy, R., "New Results in Linear Filtering and Prediction," Journal of Basic Engineering (ASME), Vol. 83D, 1961, pp. 95-108.

Magill, D. T., "Optimal Adaptive Estimation of Sampled Processes," IEEE Trans. on Automatic Control, Vol. AC-10, Oct. 1965, pp. 434-439.

Morefield, C. L., "Applications of 0-1 Integer Programming to Multitarget Tracking Problems," IEEE Trans on Automatic Control, Vol. AC-22, June 1977, pp. 302-312.

Reid, D. B., "An Algorithm for Tracking Multiple Targets, " IEEE Trans. Automatic Control, Vol. AC-24, Dec. 1979, pp. 843-854.

Singer, R. A., Sea, R. G., and Housewright, K., "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments," IEEE Trans. Information Theory, Vol. IT-20, July 1974, pp. 423-432.

Smith, P. and Buechler G., "A Branching Algorithm for Discriminating and Tracking Multiple Objects," IEEE Trans. Automatic Control, Vol. AC-20, Feb. 1975, pp. 101-104.

Trunk G. V. and Wilson J. D., "Track Initiation of Occasionally Unresolved Radar Tracks," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-17, Jan. 1981, pp. 122-130.

APPENDIX A

THE RAHTT ALGORITHM

DEFINITIONS

k Time index

$Z_{k,i}$ A radar report

M_k Number of radar reports

$Z(k) = \{Z_{k,i}\}\ i = 1,2,\ldots,M_k$

$\{\tau_0\}$ All established tracks

$\{\tau_T\}$ All tentative tracks

$\{\tau_p\}$ All potential tracks

$\tau_k = \{\tau_0 \cup \tau_T \cup \tau_p\}$

$\{\theta_\ell^i\}$ Associations for report i. ℓ--track(s) associated with $i\ \ell \subset \tau_k$

$\{\theta_i^\ell\}$ Associations for track ℓ

λ Cluster index. A cluster is a grouping of associations that have either reports or tracks in common.

Z_λ^k Cluster

Ω_i^k All hypothetical tracks for i

$\overline{\Omega}_k^m$ Terminal hypothesis indexed by m

$\overline{\Omega}_k^{m,n}$ Individual terminal nodes

$\hat{\Omega}_n$ A hypothesis of data associations

$\hat{\Omega}_{n,k}$ That part of $\hat{\Omega}_n$ formed during k

$\overline{\Omega}^m$ A level of the tree

```
for k=1 to ∞ do;
            /*Form associations via equations Chapter II*/
   for i=1 to M_k do;
      for j = 1 to τ_{k-1} do;
         form associations {θ_ℓ^i} and {θ_i^ℓ};
         end j;
      end i;
      for i=1 to M_k do;  /*loop through reports              */
         if |{θ_ℓ^i}| = 1 and |{θ_i^ℓ}| = 1 then do;/* looking */
                              /*for unique associations       */
            SMOOTH(i,ℓ);
            Q_ℓ = Q_ℓ +Q_s;      /*Quality of Track + Quality of */
                                 /*Unique Correlation          */
            end unique;
         else if {θ_ℓ^i} = ∅ then next i; /*IF NO ASSOCIATIONS···*/
                                          /*NEXT REPORT       */
         else if λ=0 then do;  /* form first cluster   */
            λ=1;
            Z_λ = {θ_ℓ^i};
            end;
         else for each λ do;   /*IF A REPORT OR TRACK IN θ_ℓ^i
                                 OR θ_i^ℓ    */
            IF{Z_λ^k} ∩ {θ_ℓ^i} ≠ ∅ OR {Z_λ^k} ∩ {θ_i^ℓ}≠ ∅  /* ADD TO*/
                                                  /*CLUSTER*/
               then Z_λ^k = {Z_λ^k ∪ θ_ℓ^i};
            end;
```

```
        else do;      /* ADD NEW CLUSTER    */
             λ=λ+1;
             Z_λ^k = {θ_ℓ^i};
             end;
        end loop on i;
/* EACH CLUSTER IS FORMED INTO SUPERCLUSTERS BASED */
/* ON COMMON TRACKS AND EXISTING TREES ARE REFORMED */
/* process each Z_λ^k INDEPENDENT CLUSTER
for each z_λ^k do; /* each Z_λ^k can be solved by independent */
                   /* processors                                */
    for each i in Z_λ^k do;
       Ω_i^k = {θ_ℓ^i ∪ 0};  /*ADD false alarm hypothesis      */
       if m = 0 then do;  /* if first level this k          */
          m = 1;
          Ω̄_k^m = Ω_i^k;    /* add new terminal level */
          end;
       else do;  /*Add new level observing mutually       */
                 /*exclusive data                          */
                 /*ASSOCIATION···See Hypothesis Generation*/
                 /*and deletion Chapter III
          for each ℓεΩ_i^k & ℓ ε any Ω̄_k^m,n;
                 Create a new τ_p and Replace ℓ in Ω_i^k; with
                    new τ_p;
          for each Ω̄_k^m,n form new Ω̄_k^m+1 by;
              /*New Terminal sets Ω̄_k^m+1 are formed by       */
              /*setting Ω̄_k^m+1 to null then for each terminal */
              /*NODE ADD ALL TRACKS in Ω_i^k except those      */
```

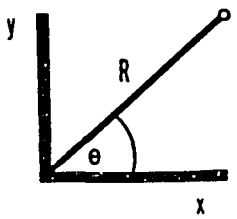
```
        /*tracks already on Ω̂_n formed during k, Ω̂_{n,k}*/
    then Ω̄_k^{m+1} = {{Ω̄_k^{m+1}} ∪ {Ω_i^k - Ω_i^k ∩ all Ω̂_{n,k}}
        /*If terminal level exceeds the expansion    */
        /*bound reduce the most unlikely Ω̄_k^{m+1,n}  */
        /*except if it is the last child of previous */
        /* Ω̄_k^{m,n}
    while |{Ω_k^{m+1}}| > EXP_B DROP(L min Ω̄_{k,n}^{m+1} that is not
                                     last child of Ω̄_k^{m,n});
    IF NOT SMOOTHED IN k SMOOTH(Ω_{k,n}^{m+1})
    m=m+1;
    end;  /* A level has been added to tree  */
/* ALL ASSOCIATIONS IN TREE.  REDUCE TO THE TRIM BOUND*/
while |{Ω_k^m}| > T_B  DROP(L_min Ω̄_k^{m,n});
/*IF it is time to identify level 1 or level 1 is  */
/*fully identified push it off tree                */
IF fully-identified(Ω̄^1) then
    increment Quality & PUSH from Tree;
    /*Select most likely hypothesis in tree and use */
    /*that hypothesis to select the level 1 correlation*/
else DO WHILE Tree to deep;
      Find L_max (Ω̂_n);
      Construct tree based on Ω̂_n;
      Push Lead level off Tree;
      Increment Track Quality;
      end;
end cluster;
```

```
    /*RADAR REPORTS PROCESSED.  τk REFLECTS NEW τp */
    /*and existing Tracks.  Process IFF            */
For All IFF Reports and all τk do;
    IF report and track codes match do;
          IF report AND Track Associate then Smooth;
          else do; /* Jitter Test */
            IF AZIMUTH ONLY ASSOCIATION Failure then do;
               IF Previous code matching report also
                  AZIMUTH ONLY FAILURE then do;
                  IF COMPATIBLE ORIENTATION AND DISTANCE
                     START New Track;
                  end previous AZIMUTH ONLY;
               end AZIMUTH FAILURE;
            IF RANGE only failure SMOOTH;
            STORE(IFF report);
    /* Process τk                */
    FOR each τ0 and each τT child of τ0;
          IF Quality of τT < τ0 by at least Upgrade
                                minimum then
             τ0 ← τT and Drop τT;
          IF Quality of τ0 > τT by at least Upgrade
              minimum then DROP τT;
```

```
                IF τT and τ0 have similar trajectors drop(τ)
                   with lower Quality τ0 ← τT if necessary;
                IF τT correlation time to old (tentative
                   Track delay) DROP (τT);
                end;
          FOR each τp do;
                IF τp has not had correlation with POTENTIAL
                      TRACK Delay then DROP (τp);
                end;
          FOR ALL τT do;
                IF any τT have similar paths then do;
                    DROP(τT) with lower Quality;
                    IF Quality equal then DROP τT with τ0
                       having higher Quality;
                    end;
                end;
end RAHTT;
```

APPENDIX B

For a variable A, $\sigma_A^2 = \text{Var } A = E(dA^2)$



$$x = R\cos\theta \quad dx = \cos\theta\, dR - R\sin\theta\, d\theta$$

$$y = R\sin\theta \quad dy = \sin\theta\, dR + R\cos\theta\, d\theta$$

$$\text{Var } x = E(dx^2) = E[(\cos\theta\, dR - R\sin\theta\, d\theta)^2]$$

$$= \sigma_R^2 \cos^2\theta + R^2\sin^2\theta\, \sigma_\theta^2$$

$$\text{Var } y = E[dy^2] = E[(\sin\theta\, dR + R\cos\theta\, d\theta)^2]$$

$$= \sigma_R^2 \sin^2\theta + \sigma_\theta^2 R^2\cos^2$$

Correlation Coefficient

$$P_{AB} = \sigma_{AB}/\sigma_A\sigma_B = \frac{E(AB)}{\sqrt{E(A^2)}\ \sqrt{E(B^2)}}$$

$\therefore$

$$\rho = \frac{E(dxdy)}{\sqrt{E(dx^2)}\ \sqrt{E(dy^2)}}$$

using predicted azimuth $\bar{\theta}$ and predicted range $\bar{R}$

$$E(dxdy) = \sin\bar{\theta}\cos\bar{\theta}\, E(d\bar{R}^2) - \sin\bar{\theta}\cos\bar{\theta}\, \bar{R}^2 E(d\bar{\theta}^2) + \bar{R}(\cos^2\bar{\theta} - \sin^2\bar{\theta})\, E(d\bar{R}d\bar{\theta})$$

Since $\bar{\theta}$ and $\bar{R}$ are independent last term = 0

$$E(dxdy) = (\sigma_R^2 - \bar{R}^2\sigma_\theta^2)\sin\bar{\theta}\cos\bar{\theta}$$

$$\rho = \frac{(\sigma_R^2 - \bar{R}^2\sigma_\theta^2)\sin\bar{\theta}\cos\bar{\theta}}{(\text{Var } x)^{\frac{1}{2}}\ (\text{Var } y)^{\frac{1}{2}}}$$

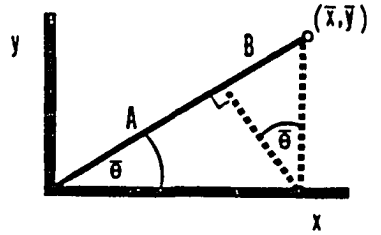
Using predicted asimuth $\bar{\theta}$, $\bar{x}$, and $\bar{y}$ the predicted range $\bar{R}$ is

$$\bar{R} = \bar{x}\cos\bar{\theta} + \bar{y}\sin\bar{\theta}$$

where

$$\bar{x} = \hat{x}\,\text{old} + \hat{x}\Delta t$$

$$\Delta t = t\,\text{report} - t\,\text{smoothing}$$



$$\text{Cos}\bar{\theta} = A/\bar{x} \qquad A = \bar{x}\cos\bar{\theta}$$

$$\text{Sin}\bar{\theta} = B/\bar{y} \qquad B = \bar{y}\,\text{Sin}\bar{\theta}$$

$$\bar{R} = A + B = \bar{x}\cos\bar{\theta} + \bar{y}\sin\bar{\theta}$$

then $d\bar{R} \cong \text{Sin}\bar{\theta}\, d\bar{y} + \text{Cos}\bar{\theta}\, dx$

relates the changes in $\bar{x}$ and $\bar{y}$ to changes in $\bar{R}$ while holding $\bar{\theta}$ constant.

Using predicted $\bar{R}$, $\bar{x}$, and $\bar{y}$

$$\bar{x} = \bar{R}\cos\bar{\theta}$$

$$\bar{y} = \bar{R}\sin\bar{\theta}$$

$$\bar{\theta} = \tan^{-1}\bar{y}/\bar{x}$$

$$d\bar{\theta} = \frac{(d\bar{y}/\bar{x} - \frac{\bar{y}d\bar{x}}{x^2})}{1 + \frac{\bar{y}^2}{x^2}} = \frac{x\,d\bar{y} - y\,d\bar{x}}{x^2} \cdot \frac{x^2}{x^2+y^2} = \frac{\bar{x}\,d\bar{y} - \bar{y}\,d\bar{x}}{x^2 + y^2}$$

$$= \frac{\bar{R}^*\cos\bar{\theta}\,dy - \bar{R}^*\sin\bar{\theta}\,d\bar{x}}{R^2}$$

$$d\bar{\theta} = \frac{\cos\bar{\theta}\,dy - \sin\bar{\theta}\,dx}{\bar{R}}$$

From

$$\bar{R} = \bar{x}\cos\bar{\theta} + \bar{y}\sin\bar{\theta}$$

$$\frac{d\bar{R}}{dt} = \frac{d\bar{x}}{dt}\cos\bar{\theta} + d\bar{y}/dt\ \sin\bar{\theta}$$

which relates changes in range to changes in velocity while holding $\bar{\theta}$ constant.

Calculate expected variance in predicted range $E(d\bar{R}^2)$ based on expected variance in prediction $\bar{x}$ and $\bar{y}$ $E(d\bar{x}^2)$&$E(d\bar{y}^2)$

$$d\bar{R} = \text{Cos}\theta\ d\bar{x} + \text{Sin}\theta\ dy$$

$$E(d\bar{R}^2) = E[\ (\text{Cos}\theta\ d\bar{x} + \text{Sin}\theta\ dy)^2]$$

$$= \text{Cos}\theta^2\ E(d\bar{x}^2) + \text{Sin}^2\theta E(d\bar{y}^2) + 2\ \text{Sin}\bar{\theta}\ \text{As}\ \bar{\theta}\ E(d\bar{x}d\bar{y})$$

where

$$E(dx^2) = E[X_{old} + x\Delta t)^2]$$

$$= E[x_{old}^2] + 2\Delta tE(x_{old}x) + \Delta t^2\ E(\dot{x}^2)$$

$$E(dy^2) = E[y_{old}^2\ ] + 2\Delta tE(y_{old}\dot{y}) + \Delta t^2E(\dot{y}^2)$$

$$E(dxdy) = \rho\sqrt{E(dx^2)}\ \ \sqrt{E(dy^2)}$$

$$= \frac{\sigma_R^2 - R^2\sigma_\theta^2\ \text{Sin}\bar{\theta}\ \text{Cos}\bar{\theta}}{(\text{var}\ x)^{\frac{1}{2}}\ (\text{var}\ y)^{\frac{1}{2}}}\ \sqrt{E(dx^2)}\ \ \sqrt{E(dy^2)}$$

Then the range window is

$$W_R = K_1(\sigma_R^2 + E(d\bar{R}^2))$$

Similarly

$$E(d\theta^2) = [\text{Sin}^2\bar{\theta}\,E(dy^2) + \text{Cos}^2\bar{\theta}\,E(dx^2) - 2\ \text{Sin}\bar{\theta}\text{Cos}\bar{\theta}E(d\bar{x}d\bar{y})]\bar{R}^2$$

$$\therefore\ W_\theta = K_2(\sigma_\theta^2 + E(d\bar{\theta}^2))$$

and

$$E\left(\frac{d\bar{R}^2}{dt}\right) = \mathrm{Sin}^2\bar{\theta}E\left(\frac{d\bar{y}^2}{dt}\right) + \mathrm{Cos}^2\bar{\theta}E\left(\frac{d\bar{x}^2}{dt}\right) + 2\ \mathrm{Sin}\bar{\theta}\mathrm{Cos}\bar{\theta}E\left(\frac{d\bar{x}}{dt}\ \frac{d\bar{y}}{dt}\right)$$

with

$$E\left(\frac{d\bar{x}}{dt}\ \frac{d\bar{y}}{dt}\right) = p\ \sqrt{E\left(\frac{d\bar{x}^2}{dt}\right)}\ \sqrt{E\left(\frac{d\bar{y}^2}{dt}\right)}$$

$$\dot{WR} = K_3\left(\sigma_{\dot{R}}^2 + E\left(\frac{d\dot{\bar{R}}^2}{dt}\right)\right)$$

Where K_1, K_2, and K_3 are sigma weighting factors.

APPENDIX C

SIMULATION ALGORITHM

```
/*  The simulation algorithm develops scenarios or reads scripted    */
/*  scenarios and generates report positions based on the true       */
/*  target position                                                  */

Procedure Main;  /*  Driver routine  */
For  I=1, forever do;
    Call SETUP;  /*  Read and create data  */
    For  J=1, number_of_simulations do;  /*  Simulations per scenario */
        For N=1, number_of_scans do;  /*  Scans per scenario          */
            If N>1 then do;
                Increment time by sample rate;
                Call True_target_generator;
                end;
            Call Apparent_target_generator;
            If N=1 do initialization; /* Kalman filter, housekeeping */
            Call Radar_report_generator;
            Call IFF_report_generator;
            Call RAHTT;
            end N;
        If J< number_of_simulations reset track positions to starting
            positions for this scenario;
            end J;
        end I;
```

```
Procedure SETUP;
/*  Setup is used to input data for control over the number, length, */
/*  and characteristics of scenarios.  To avoid excessive inputs     */
/*  FORTRAN namelists are used to modify only the desired data       */

If generating random scenarios Call Random_Scenario;
else do;
    Read inputs and modify data base;
    If input modifies scenario generation do;
        If random scenarios to generate Call Random_Scenario;
        else for each track Read X, Y, velocity, and heading;
        end;
    If input = STOP then STOP;
    If input = end_read then EXIT read loop;
    end read loop;
If tentative_track_delay = 0 then
    Tentative_track_delay = (3.3 * sample rate);
If Potential_track_delay = 0 then
    Potential_track_delay = Scan_delay * sample_rate;
end SETUP;
```

```
Procedure Random_Scenario; /* Generate a scenario */
/*  Upon input command a random scenario is created.  On          */
/*  subsequent calls to setup random scenarios are generated      */
/*  until the number of required scenarios have been created      */
/*  or until overriden                                            */

Calculate location and dimension of test area; /* Tests were conducted  */
                                               /* 140≦X≦150, 0≦Y≦10     */
Calculate scenario track start area; /* Target starting positions are  */
                                     /* uniformly spaced in a triangle */
                                     /* 145≦X≦150, 5≦Y≦10              */
For I=1, number_of_Tracks desired; /* usually 10                       */
    Calculate X and Y position of track;
    Calculate heading of track;  /*  random 105-180 degrees           */
    Calculate velocity of track; /*  random 200-500 KTs               */
    end I;

end Random_Scenario;
```

```
Procedure True_Target_Generator;     /* TTG */
/*  TTG moves the actual target position to the new true              */
/*  position for this sampling using standard equations of motion     */

end;

Procedure Apparent_Target_Generator;  /* ATG */
/*  ATG generates the apparent target position by corrupting          */
/*  the true target position                                          */
For each true track do;
    /* For the radar reports */
    Apply Guassian distributed error to true azimuth; /*μθ=0 σθ= .005 radians */
    Apply Guassian distributed error to true range; /*μR=0 σR= .1 NM          */
    Apply Guassian distributed error to true range rate; /*μṘ=0 σṘ= 10kt     */
    /* For IFF reports */
    If not jitter_allowed apply Guassian distributed error to true
        azimuth; /*μθ=0 σθ= .005 radians  */
    Apply Guassian distributed error to true range; /*μR=0 σR= .25 NM         */
    If jitter_allowed and this should be a jittered report;
        Apply jittered azimuth and azimuth bias;
    else if jitter_allowed apply Guassian distribution to
        true azimuth;

end  ATG;
```

```
Procedure Radar_Report_Generator;

Determine number and location of false alarms;
/* False alarms are Possion distributed with mean 15 for test and    */
/* uniformly distributed in test area                                */
For all reports detected and false alarms;
    Create a raw buffer of reports with the reports azimuth ordered;
    end;
end Radar_Report_Generator;

Procedure IFF_Report_Generator;
    For each apparent IFF report detected;
        Create a raw report buffer;
        end;
    end;
```