A DISTRIBUTED MULTIOBJECT TRACKING ALGORITHM

FOR

PASSIVE SENSOR NETWORKS

by

RICHARD P. HUGHES

B.S., United States Military Academy
(1979)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1980

Signature of Author_______________________________________
Department of Electrical Engineering and
Computer Science June 23, 1980

Certified by_______________________________________
Robert R. Tenney
Thesis Supervisor

Accepted by_______________________________________

Chairman, Departmental Graduate Committee

A DISTRIBUTED MULTIOBJECT TRACKING ALGORITHM

FOR

PASSIVE SENSOR NETWORKS

by

Richard P. Hughes

Submitted to the Department of Electrical Engineering and
Computer Science on June 23, 1980 in partial fulfillment
of the requirements for the Degree of Master of Science
in Electrical Engineering and Computer Science

ABSTRACT

The multiobject tracking problem is one that has gained much attention in recent years. The primary difficulty is the subproblem of associating measurements among a several number (perhaps unknown) of targets. In the present work, a recently proposed algorithm is modified and extended for application in distributed processing passive networks. This is done by distributing the measurement-target association problem to the individual sites in the network and then combining each site's resulting associations to form the desired target tracks. Computer simulations were run to demonstrate the capabilities of the procedure.

Thesis Supervisor: Robert R. Tenney

Title: Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I would like to thank my advisor, Bob Tenney, whose calm, collected wisdom got me through my rough spots.

Thanks also go to Dick Lacoss and Peter Green for many fruitful (and lengthy) discussions.

Thanks also must go to the Hertz Foundation who supported me during my stay at MIT.

And, of course, to Melodee, who was always behind me and whose love never failed me. I dedicate this work to her.

CHAPTER 1

INTRODUCTION

The need for ever more sophisticated multi-target algorithms has increased greatly in recent years. In military applications, especially, the need for these techniques is evident. Recently, there has been increased interest in distributed tracking systems, in which the tracking problem is broken down into several smaller problems and distributed to trackers at several different sites. Such problems, of course, present more challenges than the more traditional central processing tracking systems.

The basic problem in multi-target tracking is data association. If there are several (perhaps an unknown number of) targets in an area, measurements are received from each of them. There exists an uncertainty in the true origin of each measurement. Essentially, any multi-target tracking procedure should partition the measurements into sets associated with each target before the tracking of that target can be done.

Several important papers that have appeared in recent years come to grips with this problem in a variety of ways. Bar-Shalom [3] gives an excellent survey of recent work in the field. The most important paper for this work is the one by Donald Reid [8]. Reid formulates hypotheses of the origin of measurements sequentially, and organizes them in the form of a hypothesis tree. For each hypothesis constructed, a set of Kalman filters is constructed to track the targets implied by that hypothesis. The strength of the hypothesis is then evaluated by comparing the actual measurements with predicted values obtained from the state estimates of the filters. A specific formula for this is developed by Reid, and incorporates false alarms and new targets.

Reid assumes in his work a central processing system - that is, a system in which all tracking computations are done at a single site. The goal of this work is to apply Reid's ideas to a distributed processing system. As we shall see, the data association and target tracking can be separated quite naturally to fit in the mold of such a system. Our primary example will be a two node passive system that receives bearings to targets as measurements. The information signal is assumed to be acoustic in nature, which in combination with a desire to track fast moving targets improves the problems of propagation delay.

Chapter 2 develops the theoretical structure of the proposed tracking algorithm in great detail. In Chapter 3, we discuss the concept of a Distributed Sensor Network (DSN), which motivates the choice of our primary example. Chapters 4 and 5 develop the mathematical formulations necessary for the implementation of the tracking algorithm. Finally, in Chapter 6, we present some demonstrations of the performance of the algorithm.

CHAPTER 2

THE ALGORITHM

In this chapter, we will model the solution of a generalized tracking problem, and see how previous work in multi-target tracking fits into this model.

2.1 SOME DEFINITIONS

Consider a generalized tracking system T. The task of T is to determine the existence of and track certain objects located in a specified environment E. The objects are called targets. The system tracks a target if it can estimate its position and velocity at any time while it is located in E. A tracking system may also determine acceleration and other higher positional derivatives; however, for our purposes, position and velocity will be sufficient. This information is collectively called the target state.

To perform its task, the tracking system must obtain information from the environment. Usually, this information is in a form termed signals. For example, a radar system might use electromagnetic sinusoids, whereas in sonar, the signals are acoustic. Any place within T that receives signals is called a sensor. If the received signals originated in T and were reflected back, T is called an active system. If the signals originated in the environment, T is a passive system. It may happen that the system receives both types of signals, in which case T is referred to as a mixed system. Active systems generally obtain more information from the environment than passive systems; however, they are also more susceptible to detection by the targets themselves. A tracking system can also be classified by the manner in which it collects signals. Our attention will focus on time-sampling systems, which sample signals from all directions at specified instants of time.

Usually, the received signals in their original form are not very useful for tracking. A signal processor in T extracts data from the signals which can be used in tracking. In this work, data obtained in this manner from signals are termed measurements. Measurements may be scalar or vector in nature. For example, a measurement from a radar system might consist of three position coordinates. A measurement from a passive system, on the other hand, might simply be a bearing. In time-sampling systems, a scan is defined as a set of measurements obtained at the same time-sampling instant. In many systems, a signal processer requires signals from several different sensors to produce a single measurement. Normally, this set of sensors is fixed and is referred to collectively as a node. Since our attention is focused on the tracking problem, we will not go into detail the highly nontrivial problem of signal processing. Hence, all of our work will refer to measurements and nodes at the lowest level.

2.2 - THE TRACKING PROBLEM

In general, the tracking system must be able to track several targets simultaneously. In multi-target tracking, there are really two problems that must be solved. They are the data association problem and the track association problem. Logically, before a tracker can estimate the state of a particular target, it must know which measurements to use. Given a time-sequenced set of scans, the tracker must partition the set of all measurements into a set of mutually exclusive subsets. One subset may correspond to an assignment to no target at all. The measurements in this set are called false alarms. The other subsets will correspond to hypothesized targets. The process of forming these subsets is data association, and each subset of measurements is called a data track. There will be many different ways to perform data association on a given set. Each such configuration is called a data hypothesis.

We will asssume in our work that scans are to be processed sequentially. This is reasonable, since in a real-time system, it is usually desirable to incorporate new information as soon as possible. Therefore, at any given time, the number of possible data hypotheses depends on the number of a priori hypotheses and the number of measurements in the present scan. The sequential formation of hypotheses can be organized conceptually through the use of trees.

The structure of hypothesis trees is best illustrated through a specific example. Assume a priori there are no data tracks. At time t=0, the tracker receives two measurements. We make the simplifying assumption that a particular target cannot be the source of more than one measurement in a given scan, although the more general case can still be displayed in tree form. Thus, each measurement could be from a different legitimate target or a false alarm. This results in four different data hypotheses, as shown in Fig. 2-1.

m1 and m2 are the two measurements. As can be seen, each corresponds to a different level of the tree. The locations within the tree corresponding to the assignment of measurements to a data track are called nodes. (The use of this terms is unfortunate; however, it will always be clear from context whether we are talking about groups of sensors or tree structure.) A sequence of connected nodes is called a branch of the tree. Each branch of the tree corresponds to a different data hypothesis, as shown. Thus, hypothesis 1 assigns m1 to data track 1 and m2 to data track 2, while hypothesis 3 assigns m1 as a false alarm (represented by 0) and m2 to data track 2. We have made the convention of identifying a data track by the measurement number of its first measurement.

Assume now that we have a second scan with two measurements (m3 and m4). The number of assignments of m3 and m4 will depend on which prior hypothesis is assumed. For instance, if hypothesis 1 is assumed, the possible assignments fo- each are data track 1, data track 2, a new data track, or a false alarm, with the condition that both cannot be assigned to the same legitimate data track. If hypothesis 4 is assumed, the only possible assignments are a new data track or a false alarm. If we expand each branch in this manner, we obtain Fig. 2-2, which represents the total number of ways the measurements in the two scans can be associated, given our assumptions. It is easy to see that the tree expands at an exponential rate.

The above procedure for the construction of the hypothesis tree is known as a measurement-oriented approach, because every possible data track is listed for each measurement. A target-oriented approach would list

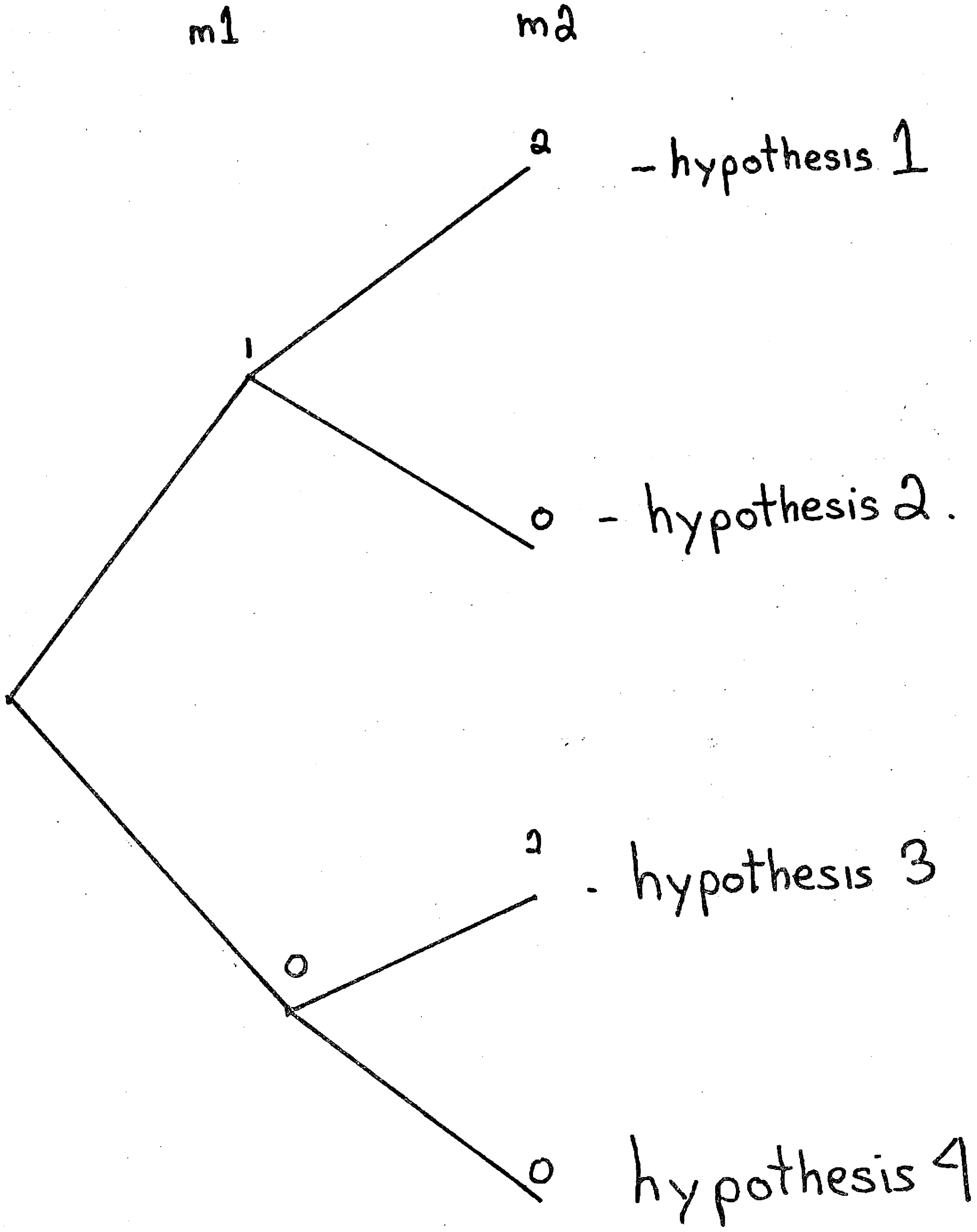

m1
m2
2
- hypothesis 1
1
0
- hypothesis 2.
2
. hypothesis 3
0
0
hypothesis 4

Figure 2-1

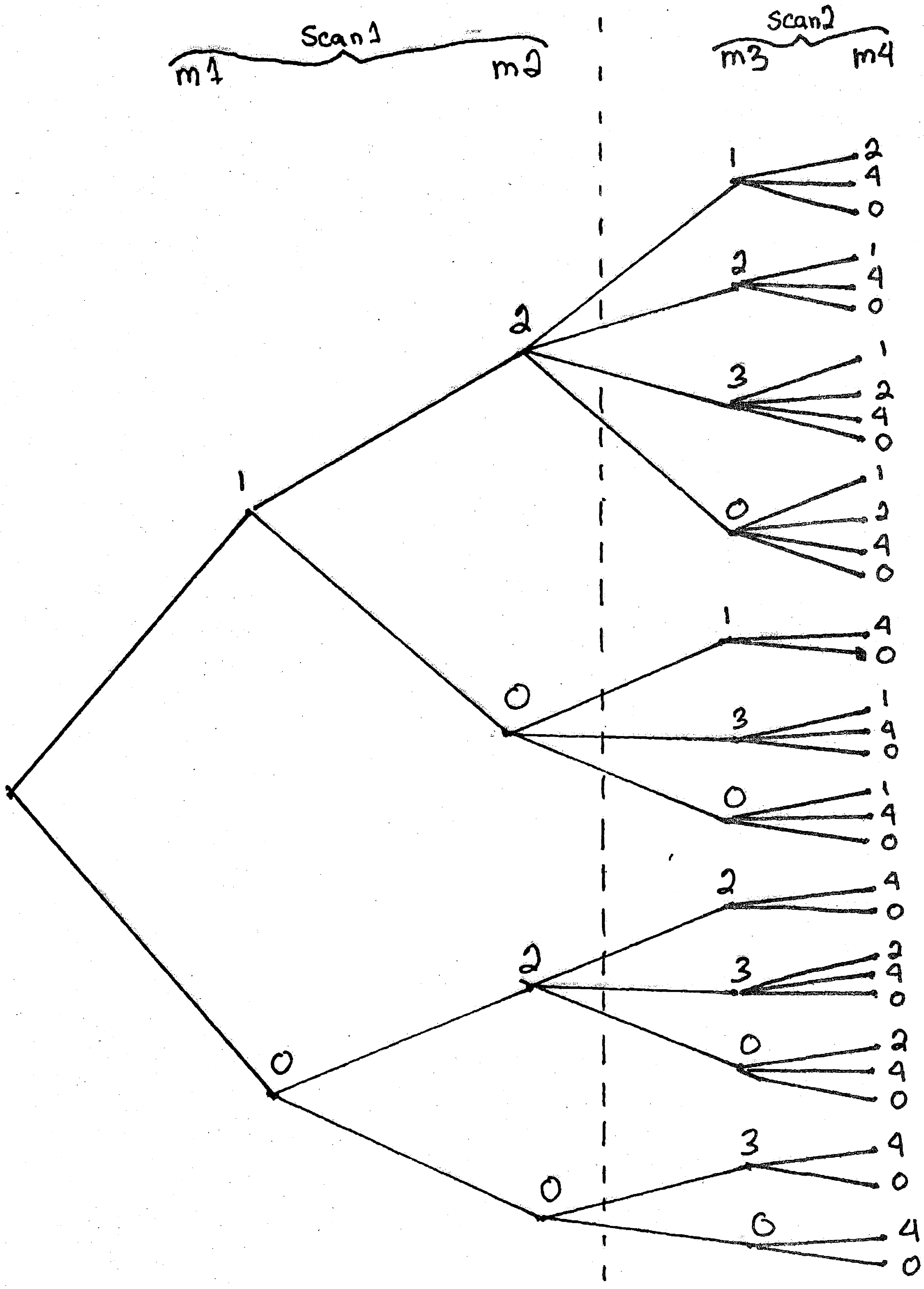

Scan1
m1
m2
Scan2
m3
m4
1
2
0
2
0
1
2
3
0
1
3
0
2
3
0
3
0
2
4
0
1
4
0
1
2
4
0
1
2
4
0
4
0
1
4
0
1
4
0
4
0
2
4
0
2
4
0
4
0
4
0

Figure 2-2

every possible measurement for each data track. However, in taking the latter approach, it is conceptually difficult to decide when a new data track should be hypothesized, whereas in the former case, new data tracks appear naturally as a part of tree expansion. For this reason, we will use the measurement-oriented approach.

If there are multiple nodes in the tracking system, we can perform the data association in two ways. A central processing system would construct one hypothesis tree, incorporating all measurements from every node in the system. A distributed processing system would construct a tree for each node or a subset of nodes. We can think of nesting the various trees in a distributed processing system in one another to produce an overall data association hypothesis tree that is equivalent to the tree constructed in a central processing system, which contains all possible data tracks that can be formed given the received measurements. With a distributed structure, we encounter the track association problem.

The track association problem can also be modeled as a hypothesis tree. Whereas data hypotheses associate measurements, track hypotheses associate the data tracks of different nodes. Given a track hypothesis and the data tracks it is conditioned on, we finally can combine the data to form sets of estimated target states, or target tracks. (This assumes, of course, that there are a sufficient number of data tracks to make such an estimation possible; i.e. the system must be observable). The production of these target tracks is the desired goal of the tracking system. Since the track association trees are each conditioned on data hypotheses from the various nodes of the system, we can nest the former into the overall data association tree of the tracking system. The resulting structure organizes, in a systematic way, all possible solutions of the tracking problem.

The prospect of wading through such an imposing structure is horrifying, at best. In the next sections we will discuss ways of making the solution more tractable. These methods, of course, produce suboptimal results, in the sense that the correct solution may accidentally be discarded. However, they are quite necessary for any practical implementation.

2.3 - DATA HYPOTHESIS REDUCTION - TREE PRUNING

The use of trees as a problem-solving technique is well known in artificial intelligence. They are used to systematically model the step-by-step solution of very general problems. As we have seen, they fit naturally into the tracking problem. However, in most applications, there is a well-defined "goal state", which will be reached eventually by one of the branches of the tree. In the tracking problem, there is no "goal state"; the trees are open-ended. Since the trees expand exponentially, we must impose constraints on tree growth in order to keep the problem manageable. Methods used for this purpose are called tree pruning techniques.

There are two ways to limit tree expansion. The breadth of the tree can be limited by retaining no more than a specified maximum number of branches each time the tree expands. This "pruning" of branches of the tree may result in one branch of an older scan being singled out. This is illustrated in Fig. 2-3. In this case, the scan is said to be identified. Scans will not always be identified by breadth constraints. Therefore, we must also limit the depth of a tree. When the number of scans in a tree attains a certain maximum allowable number, one branch in the oldest scan is singled out and the others are pruned. This scan is thus forcibly identified. In limiting the depth of a tree, redundant hypotheses may appear in the tree, depending on the mechanism used. A set of redundant hypotheses in a depth limited tree assign measurements in the same manner. They may assign measurements to exactly the same targets (in which case they are termed identical) or there may exist a 1-1 mapping of targets between the hypotheses. A set of redundant hypotheses may be combined under the assumption that any past differences which have been dropped off of the tree are insignificant. This is called hypothesis merging. (A more thorough treatment of the foregoing concepts may be found in [6].)

In order to apply the above methods, we must have some means of measuring the strength of the various hypotheses. In other words, we must define a probability measure on the set of branches of the tree. The particular definition and evaluation of a probability measure depends upon the nature of the tracking system and the types of measurements. One observation can be made here, however. For data association trees, a natural definition of probability is one that is monotonically related to the "closeness" of a measurement to the predicted value of a data track. The closer a measurement is to a particular prediction, the higher the probability that the measurement is associated with the corresponding data track. This use of predictors, and the fact that we are using sequential processing,

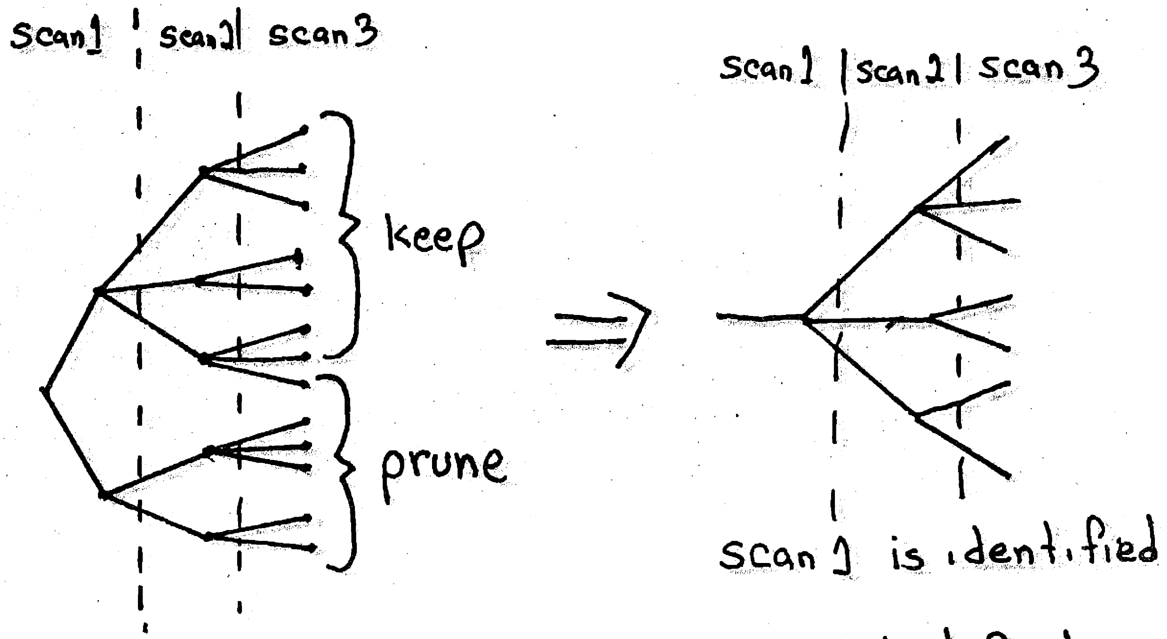


Figure 2-3 - Scan Identification

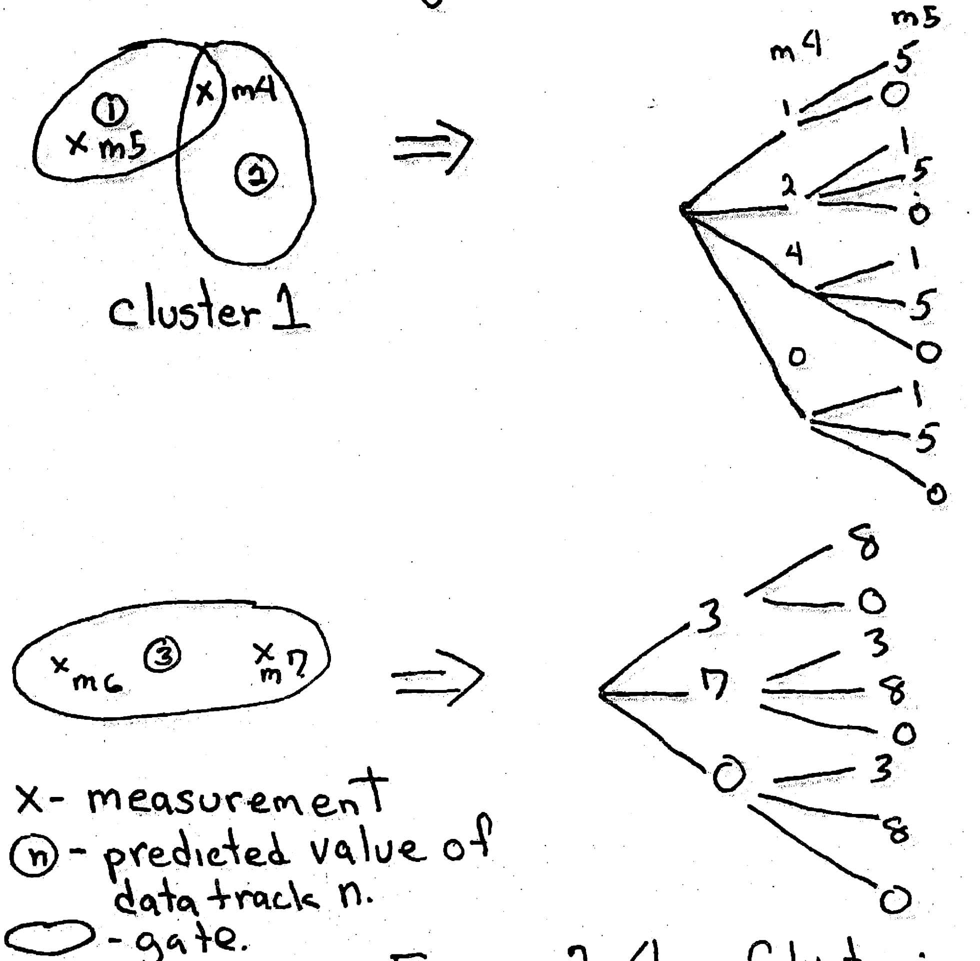


Figure 2-4 Clustering

suggests the use of Kalman filters. A detailed application of this idea appears in Chapter 4.

The definition of this "closeness probability" leads to the concept of clustering. (See Fig 2-4). Suppose a threshold is defined such that, during tree expansion, any branch whose probability lies below this threshold is pruned. We can think of drawing a "gate" around the predicted value of each data track. The probability of a hypothesis below the threshold is then equivalent to the present measurement falling outside the gate. By considering only measurements inside the gate, we have effectively pruned the tree of hypotheses with probabilities below a certain level. Suppose now that the gates of various data tracks do not overlap. The scan can then be partitioned into a number of subsets, each subset containing the measurements falling inside the gate of a different data track. Because the possible data associations of each of these subsets are mutually exclusive, we can break up the data association tree into a number of smaller, independent trees. This is an enormous simplification, for the sum of all of the hypotheses in the smaller trees will be much less than the number of hypotheses in the tree which spawned them. For fixed resources, this amounts to an increase in the number of hypotheses that can be considered simultaneously. If some of the gates do overlap, we can form one tree for the cluster of the corresponding data tracks. There is still a simplification in this case, although not as great. The concept of clustering is important for our later work.

2.4 - TRACK HYPOTHESIS REDUCTION - THE DELAYED N-SCAN ALGORITHM

The data association trees of different tracking systems are usually quite similar in structure. The construction of track association trees will vary widely with the tracking system. In some cases, the track association trees are degenerate, and target state estimates are a trivial consequence of data association. For instance, take the case of a single node, single target tracker that uses position measurements. Initial target states can be estimated using only two measurements. In fact, the target state can be used in a Kalman filter to evaluate probabilities in the tree. This formulation is precisely the "N-scan algorithm" developed by Singer, Sea, and Housewright [9]. The term "N-scan" refers to the fact that the hypothesis tree is depth-limited to N-scans, and hypothesis merging is done over the past N-scans. A generalization to the multi-target case was presented by Reid [8]. Here again, target states are used for probability calculations. Reid uses clustering in his algorithm as well as an N-scan approach.

The generalization of these algorithms to a multi-node active system, of course, requires the correlation of state estimates from different nodes, and hence requires trees. However, assuming that node estimates are independent of one another, and that noise and target density are sufficiently low, ambiguities will resolve themselves fairly rapidly. The case is different for passive systems, which are inherently multi-node in nature. Even if it is possible to estimate target state from the measurements of a single, passive node, the covariance matrix of such an estimate is usually so large that estimate is virtually useless. Correlation between data tracks of different nodes in the system is thus necessary to obtain good estimates.

Correlation of data tracks requires time, and so reduction of track association trees is a rather drawn out process. It would be desirable to do as few of these operations as possible. This is the motivation for the following scheme, called the "delayed N-scan approach." Instead of constructing track association trees for each data association hypothesis under consideration, we delay the construction until data tracks are determined in an N-scan algorithm on the data association tree. In other words, a measurement is not used in updating probabilities of track hypotheses until the scan to which it belongs becomes identified in the data association tree. This technique effectively separates the data association and track association processes. The resulting benefits are the same as in employing clustering, since the track association trees were originally nested in the data association tree. The disadvantage, of course, is that the incorporation of measurements into state estimates is delayed by the N-scan algorithm on the data association tree. It is the price paid for simplification of the problem. This procedure will be used in our application in Chapter 3.

2.5 - OTHER WORK IN MULTITARGET TRACKING

Before leaving this chapter, we should summarize previous work in multitarget tracking and compare it to the approach taken here. An excellent survey of multitarget tracking methods can be found in a paper by Bar-

Shalom [3].

All of the work in the literature focuses primarily on developing techniques to resolve the uncertainty in the origin of received measurements; i.e., the data association problem. The applications of these techniques generally assume that target state estimates can be obtained directly from given sequences of measurements.

The approaches to the problem can be classified as Bayesian and non-Bayesian. Non-Bayesian approaches do not take into account a priori information. The early work of Sittler [10] typifies this approach. In his algorithm, data association trees are formed in a similar manner as in our algorithm, although this is not explicitly stated. Kalman filters estimating target states are initiated and updated directly by the given measurements. The innovations of the filter of each branch are used to sequentially compute a likelihood function. Target tracks whose likelihoods are below a certain threshhold value are then rejected. A somewhat different technique developed by Morefield [7] minimizes the likelihood function by transforming the problem into an integer programming problem. This produces the most likely set of target tracks given all of the data, but is a batch processing technique.

Two observations can be made. The first is that the state estimates and covariances are conditioned on the corresponding data hypotheses being true. However, no probability value is obtained for the data hypotheses themselves. This is essentially due to the philosophy of the non-Bayesian approach. The second observation is that, without a priori information, it is difficult to apply these algorithms in a distributed processing system, because the data association and track association are intertwined.

The Bayesian approaches to the multitarget tracking problem attach to each measurement a probability of being correct, based on a priori information. The resulting target state estimates and covariances reflect the uncertainty in the origin of the measurements. In [1], Bar-Shalom and Tse deal with the single target case. Assuming a target state has already been initiated, a gate can be formed around the estimated target state. For each measurement that falls within the gate, a probability of being correct is computed based on how close the measurement is to the estimate. These measurements, weighted by their probabilities, are then used to update the estimate. The resulting filter is called the probabilistic data association filter (PDAF). This approach is target oriented in nature. As such, it is difficult to incorporate initialization of target tracks into the scheme. The PDAF is extended to the multitarget case in [2]. However, the scheme for computing probabilities is very complicated. In addition, the application of these ideas to a distributed processing system would increase enormously the amount of computation required. Because of this, and our desire to include track initiation, we have rejected this approach.

The work of Singer et. al. [9] and Reid [8] has already been mentioned. We have described the delayed N-scan algorithm as a generalization of Reid. It is probably more accurate to characterize it as a combination of the Bayesian and non-Bayesian approaches. In the sequel, we correlate data tracks in the track association trees by computing a liklihood function for each track hypothesis. The target state estimation is conditioned on a particular data hypothesis (or hypotheses) being true. This is exactly the non-Bayesian approach. However, the data hypotheses upon which these target tracks are conditioned are obtained by a "pre-processor" which employs a Bayesian N-scan algorithm similar to Reid's. Thus, although the target state estimates do not reflect measurement origin uncertainty, they are based on hypotheses that have been singled out in a Bayesian weeding process.

2.6 - SUMMARY

In this chapter, we have defined some basic terms and constructed a model for the solution of the tracking problem. We have considered, in a general setting, some useful techniques for rendering the model amenable to practical implementation, and we have seen how some previous work in tracking fits into this model. In the next chapter, we present an application of these ideas to a specific tracking system.

CHAPTER 3

DISTRIBUTED SENSOR NETWORKS

The procedures developed in the last chapter are quite general in nature. In order to demonstrate their utility, we need to have a specific application. In this chapter, we will describe a general tracking system called a Distributed Sensor Network. This tracking system will provide the motivation for the example to which we shall apply our algorithm.

In principle, the more information a tracking system can obtain from the environment, the better it will be able to track targets. Active systems are almost always used in situations where maximum target information is the only criteria or has absolute priority. Most active systems, such as radar, send a signal into the environment and receive reflected versions of those signals, along with clutter from the environment. This operation of reflection allows an estimation of the time delay between transmission and reception. Since the speed of propagation of the signal is assumed known, this is equivalent to a range estimate. Passive systems, on the other hand, do not have this information, and hence must contain more sensors and nodes than active systems to obtain equivalent information.

In many cases, however, maximum target information is not the sole criteria, nor does it always have the highest priority. Active systems have the property that potential targets in the environment may be able to detect the radiated signals. In many cases, especially in military applications, this detectability is a major drawback, and the design of passive systems is desirable.

DSN (Distributed Sensor Network) is a research project of MIT Lincoln Laboratory. DSN is a multi-node surveillance system designed to detect, locate, track, and identify low-flying aircraft. Although in actual practice the system may be a mixed one, with both active and passive sensors, it turns out to be more fruitful to study a strictly passive system, since the distributed processing and control problems of large active systems occur in much smaller passive systems. In addition, in reference to detectability, it would be desirable for the system to be capable of carrying out its functions using only passive sensors. Hence, we will view the DSN as having a solely passive capability.

The new information input to the DSN are acoustic signals. Each node of the DSN contains an array of acoustic sensors (probably high quality microphones). Each sensor samples the incoming signals at a specified rate to obtain a digital sample set. After predetermined intervals of time, the sampled signals from all sensors are input to the signal processing component of the node. Essentially, this component uses high resolution frequency wavenumber analysis to detect phase differences in the signals, thereby obtaining a direction. The output can be viewed as a curve plotting received signal power vs azimuth. An example plot is shown in Figure 3-1.

These power-azimuth curves, produced at given instants of time, are the input data to the tracking system. To keep things simple for our models to come, we will use only the azimuthal information. The power in-

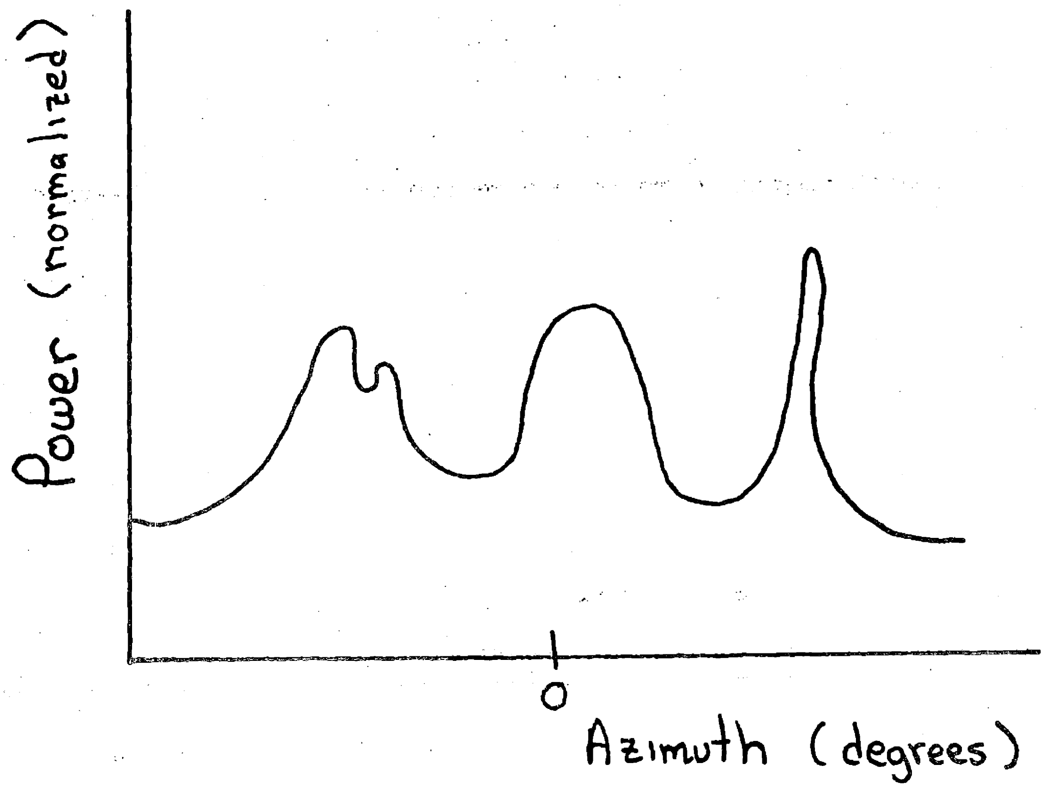


Figure 3-1 Power-Azimuth plot

formation could be useful in resolving ambiguities; however, we shall not consider this role.

We are thus assuming that each node of the DSN has available a set of azimuths and variances of those azimuths at discrete instants of time. The tracking problem is to combine the azimuths from the nodes to produce target tracks. We should keep in mind that the nodes are geographically dispersed and that the velocity of signal propagation is comparable to the speed of potential targets. In addition, the measurement times at different nodes are not necessarily synchronous. However, for simplicity, we shall ignore this last observation.

In a large scale DSN, it is obvious that a distributed processing scheme is much more desirable than a central processing one. First of all, the large amounts of information dictate that it be handled in pieces in order to sort it out. Second, communication is generally much slower than computation, and so for time efficiency, the computational load should be distributed as much as possible. Communications also radiate power, which is undesirable if detectability is an issue.

Thus, it seems logical to do as much of the processing at individual nodes as possible. There is not enough information gathered by the node to produce independently a reliable target track. However, it is possible for the node to perform its data association independently of other nodes. This is why we separated out data association and track association in our algorithm. It fits very naturally into the problem of distributing computational load in a tracking system.

To produce the actual target tracks, we can correlate data tracks of pairs of nodes to produce target tracks. Target tracks can then be refined at a higher level by comparing various two-node results. This scheme sets up a hierarchy similar to multi-site radar, with the exception that the set of two-node target tracks are not always independent.

The two-node tracking problem is what we intend to study. We will assume that our tracking system consists of two nodes each independently receiving azimuth measurements. For simplicity, we will assume that the environment is two dimensional and that nodes and targets are dimensionless points. All noise in the measurements is assumed to be Guassian and white.

With the specification of our example, we are now ready to explore mathematical details. The first order of business is to discuss a definition of probability for our hypothesis trees. This is the subject of the next chapter.

CHAPTER 4

THE CALCULATION OF HYPOTHESIS PROBABILITIES

In order to apply our algorithm to the two-node system, we first need to define a probability measure on both the data association trees and the track association trees. With this we can evaluate the strength of various hypotheses and eliminate unlikely ones. In this chapter, we develop a theoretical basis for calculating the probabilities.

4.1 - DATA ASSOCIATION TREES

We first consider the data association tree at a single node. Suppose, for the moment, that an estimate of the target state is available at a measurement time t. We can then form a prediction of the incoming measurement based on this target state estimate. Intuitively, we would expect that the closer the measurement is to the predicted value, the more likely the measurement is associated with the target. As is well known, we can sequentially form target state estimates as data arrives through the use of Kalman filters.

In a more general setting, we assume that the measurements are the output of a linear system plus an additive noise term. As is well known, we can formulate a state variable description of the system in many different ways by choosing different definitions of the state variables of the system. For the moment, let us assume we have settled on a particular definition for the state of the system. We may write the (linear) state equation and measurement equation as

$$\begin{aligned} \underline{x}(k+1) &= F(k)\underline{x}(k) + G(k)\underline{w}(k) \\ \underline{z}(k) &= H(k)\underline{x}(k) + \underline{v}(k) \end{aligned} \tag{4.1}$$

$\underline{x}(k)$ is the state of the system at time k and $\underline{z}(k)$ is the measurement at time k. We assume that w and v are independent Gaussian white noise sequences with

$$E[\underline{w}(k)\underline{w}^T(k)] = Q(k)$$

$$E[\underline{v}(k)\underline{v}^T(k)] = R(k) \tag{4.2}$$

Define $\hat{\underline{x}}(k|l)$ to be the linear least squares estimates of $\underline{x}(k)$ given data up to time l and $P(k|l)$ to be the covariance matrix of this estimate. We can obtain $\hat{\underline{x}}(k|k)$ and $P(k|k)$ through the use of the discrete time Kalman filter equations as follows:

$$\begin{aligned} \hat{\underline{x}}(k|k) &= \hat{\underline{x}}(k|k-1) + K(k)[\underline{z}(k) - H(k)\hat{\underline{x}}(k|k-1)] \\ \hat{\underline{x}}(k+1|k) &= F(k)\hat{\underline{x}}(k|k) \\ K(k) &= P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1} \\ P(k|k) &= [I - K(k)H(k)]P(k|k-1)[I - K(k)H(k)] + K(k)R(k)K^T(k) \\ P(k+1|k) &= F(k)P(k|k)F^T(k) + G(k)Q(k)G^T(k) \end{aligned} \tag{4.3}$$

Using these equations, we can sequentially update the state estimate as measurements arrive. One quantity that will be of interest to us is the so-called innovations sequence

$$\nu_k = \underline{z}(k) - H(k)\hat{\underline{x}}(k|k-1)$$

It can be shown (e.g. [11]) that $\mathcal{V}_k$ is a Guassian white noise sequence given our previous assumptions with

$$E[\mathcal{V}_k \mathcal{V}_k^T] = B_k = H(k)P(k|k-1)H^T(k) + R(k) \qquad (4.4)$$

We are now ready to derive our result concerning the calculation of probabilities of hypotheses. Our deviation follows closely the work of Reid [8]. Although Reid assumes that $\underline{x}(k)$ is the actual target state, the results are valid for our more general definition of $\underline{x}(k)$.

Let

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

denote the measurements received at time k and

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

denote the cumulative set of measurements through time k. Also, define

$$\Gamma^k = \{\Gamma^k_i \ , \ i=1,2,\ldots,I_k\}$$

to be the cumulative set of hypotheses just after time k. Each Γ^k_i corresponds to a branch on the hypothesis tree.

Now, define

$$P^k_i = P(\Gamma^k_i \mid Z^k)$$

that is, P^k_i is the probability at time k of the branch Γ^k_i of the hypothesis tree given the data through time k (Z^k). In actuality, this is equivalent to the conditioned joint probability of the prior hypothesis Γ^{k-1}_g and the data hypothesis for the current measurement set ψ_h. Dropping the dependence on past data for notational simplicity, we can use Bayes' equation to write the relationship

$$P_i^k = P(\Gamma_g^{k-1}, \psi_h \mid Z(k)) = \frac{1}{\eta} P(Z(k) \mid \Gamma_g^{k-1}, \psi_h) \times P(\psi_h \mid \Gamma_g^{k-1}) \tag{4.5}$$

The term η is a normalizing factor given by

$$\eta = \sum_g \sum_h P(Z(k) \mid \Gamma_g^{k-1}, \psi_h) \times P(\psi_h \mid \Gamma_g^{k-1}) \times P(\Gamma_g^{k-1})$$

If we can find expressions for the first two terms on the right hand side of (4.5), we will have a recursive relationship for calculating probabilities.

The first term is the probability density function of the current set of measurements Z(k) given the prior hypothesis Γ_g^{k-1}, and the current data hypothesis ψ_h. Assuming that each measurement $\underline{z}_m(k)$ in Z(k) is conditionally independent, we have

$$P(Z(k) \mid \Gamma_g^{k-1}, \psi_h) = \prod_{m=1}^{M_k} P(\underline{z}_m(k) \mid \Gamma_g^{k-1}, \psi_h) \tag{4.6}$$

Suppose that ψ_h assigns $\underline{z}_m(k)$ as either a false alarm or a new data track. In either case, there is no a priori information to determine whether one set of possible measurements is more likely than another. Hence, in these two cases, we will assume that it appears as a uniform distribution

$$P(\underline{z}_m(k) \mid \Gamma_g^{k-1}) = \frac{1}{V} \tag{4.7}$$

V is the volume (or area) of the part of the environment covered by the node.

If neither case above holds, then ψ_h assigns $z_m(k)$ to either a previously established data track (confirmed track) or to a data track whose existence is implied by Γ_g^{k-1} (<u>tentative track</u>). Tentative tracks are initiated when a new data track is hypothesized in the course of expansion of the data association tree. A tentative track becomes confirmed if it still exists when the scan in which it is initiated is identified.

If a Kalman filter is running on the assigned data track, we have available a current estimate $\underline{\hat{x}}(k|k)=\underline{\hat{x}}_k$. Since this estimate does not depend on the current measurement, we have

$$P(\underline{z}_m(k) \mid \Gamma_g^{k-1}) = P\{\underline{z}_m(k)-H(k)\underline{\hat{x}}_k \mid \Gamma_g^{k-1},\psi_h\} \tag{4.8}$$

The last is the conditional density function of the innovations of the filter, which has a normal distribution. Hence

$$P(\underline{z}_m(k) \mid \Gamma_g^{k-1},\psi_h) = N[\underline{z}_m(k)-H(k)\underline{\hat{x}}_k,B^k] \tag{4.9}$$

with

$$B^k = H(k)P(k|k)H^T(k) + R(k)$$

$$N(\nu,P) = \frac{\exp(-\frac{1}{2}\nu^T P^{-1}\nu)}{\sqrt{(2\pi)^n|P|}} \tag{4.10}$$

where n is the dimension of the innovation vector x.

The second term on the right hand side of (4.5) is the probability of the current data association hypothesis ψ_h given the prior hypothesis Γ_g^{k-1}. ψ_h has three items of information:

a) Number - the number of measurements associated with prior data tracks ($N_{DT}(h)$), false alarms ($N_{FT}(h)$) and new data tracks ($N_{NT}(h)$).

b) Configuration - the partition of the set Z(k) into three subsets corresponding to prior targets, false targets, and new targets.

c) Assignment - the assignment of each measurement associated with prior data tracks to the specific source.

In addition, the prior hypothesis gives $N_{TGT}(g)$, the total number of data tracks, confirmed and tentative, implied by that hypothesis. Thus if $N_{DT} \neq N_{TGT}$, ψ_h implicitly assumes that measurements were not received from some of the prior data tracks.

To find the probability of the numbers N_{DT}, N_{FT} and N_{NT} given Γ_g^{k-1}, we make the following assumptions. First, the probability that a target will generate a measurement which is actually received by the node is a constant P_D (called the probability of detection). In other words, the reception of a measurement can be described in probabilistic terms as a Bernoulli trial. If we further assume that the detectability of each target is independent of the others, and that each target can only generate one measurement in a given scan, then we see that N_{DT} is a Bernoulli process and its distribution is binomial.

Second, we assume that the number of false alarms follows a Poisson distribution. This is an assumption often made in tracking problems. It makes intuitive sense, because while the appearance of false alarms is random, in many cases they have a constant average rate of appearance over reasonably lengthy periods of time. We will also assume the number of new targets follows a Poisson distribution. The assumption is much harder to justify. In order to make it reasonable in actual practice, the average rate of new target appearances must be adjusted much more often than the rate for false alarms.

With these assumptions, we have

$$P(N_{DT},N_{NT},N_{FT} \mid \Gamma_g^{k-1}) = \binom{N_{TGT}}{N_{DT}} \quad (1-P_D)^{(N_{TGT}-N_{DT})}$$

$$\times\ F_{N_{FT}}(\beta_{FT}V)F_{N_{NT}}(\beta_{NT}V) \tag{4.11}$$

where

P_D = probability of detection

β_{FT} = density of false alarms

β_{NT} = density of previously unknown targets that have been detected

$$F_n(\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$$

Now, the total number of measurements is

$$M_k = N_{DT} + N_{FT} + N_{NT}$$

The number of different partitions of the M_k measurements, given the numbers N_{DT}, N_{FT}, and N_{NT} is

$$\binom{M_k}{N_{DT}}\binom{M_k-N_{DT}}{N_{FT}}$$

Assuming that each configuration is equally likely, the conditional probability of a specific configuration is

$$P(\text{Configuration} \mid N_{DT},N_{FT},N_{NT}) =$$

$$\frac{1}{\binom{M_K}{N_{DT}}\binom{M_k-N_{DT}}{N_{DT}}} \tag{4.12}$$

Given the configuration, the number of possible assignments of the N_{DT} measurements to the N_{TGT} prior data tracks is

$$\cancel{\frac{N_{TGT}!}{(N_{TGT}-N_{DT})!}} \qquad \frac{N_{TGT}!}{(N_{TGT}-N_{DT})!}$$

Assuming that each such assignment is equally likely, we have

$$P(\text{Assignment} \mid \text{Configuration}) = \frac{(N_{TGT}-N_{DT})!}{N_{TGT}!)} \qquad (4.13)$$

The joint conditional probability of N_{DT}, N_{FT}, N_{NT}, the configuration, and the assignment is the product of (4.11), (4.12), (4.13), and is also the conditional probability of ψ_h. Thus we get

$$P(\psi_h \mid \Gamma_g^{k-1}) = \frac{N_{FT}!\ N_{NT}!}{M_k!} \times P_D^{N_{DT}}(1-P_D)^{(N_{TGT}-N_{DT})}$$

$$\times\ F_{N_{FT}}(\beta_{FT}V)F_{N_{NT}}(\beta_{NT}V) \qquad (4.14)$$

If we now substitute (4.6), (4.7), (4.9) and (4.14) into (4.5) we obtain

$$P_i^k = \frac{1}{\eta}\,\frac{N_{FT}!N_{NT}!}{M_k!} \times P_D^{N_{DT}}(1-P_D)^{(N_{TGT}-N_{DT})}$$

$$\times\ F_{N_{FT}}(\beta_{FT}V)F_{N_{NT}}(\beta_{NT}V)$$

$$\times \prod_{m=1}^{N_{DT}} N[\underline{z}_m(k)-H(k)\hat{\underline{x}}_m^k\ ,\ B_m^k]\ \frac{1}{V^{N_{FT}+N_{NT}}}\ P_g^{k-1} \qquad (4.15)$$

The measurements have been implicitly ordered in the above so that the first N_{DT} of them correspond to those assigned to prior data tracks by ψ_h. Substituting into (4.15), simplifying, and incorporating constants in η, we finally get

$$P_i^k = \frac{1}{\eta}P_D^{N_{DT}}(1-P_D)^{(N_{TGT}-N_{DT})}\beta_{FT}^{N_{FT}}\beta_{NT}^{N_{NT}}$$

$$\times \prod_{m=1}^{N_{DT}} N[\underline{z}_m(k)-H(k)\hat{\underline{x}}_m^k\ ,\ B_m^k]\quad P_g^{k-1} \qquad (4.16)$$

Note that this is independent of V.

As mentioned by Reid, this equation is easily implemented in the framework of tree expansion. Prior to expansion, all prior probabilities are multiplied by $(1-P_D)^{N_{TGT}}$. Then, as each branch is created for a specific measurement assignment, we multiply the probability of the prior hypothesis by either β_{FT}, β_{NT}, or $P_D N[\underline{z}_m(k)-H(k)\hat{\underline{x}}_m^k\ ,\ B_m^k]/(1-P_D)$, depending on the assignment. The probabilities can be normalized after tree expansion, although this is not strictly necessary since only the relative probabilities are important.

We can take the negative logarithm of (4.16) to obtain a recursive likelihood equation which is additive rather than multiplicative. We can write this recursion as follows:

1) To each prior hypothesis Π_g^{k-1}, add to its likelihood

$$-N_{TGT}(g)\ln(1-P_D)$$

2) As each measurement is assigned we add Δ, where

$$\Delta = \begin{cases} -\ln\beta_{FT} & \text{for false alarms} \\ -\ln\beta_{NT} & \text{for new data tracks} \\ \frac{1}{2}V^T B^{-1} V + \frac{n}{2}\ln 2\pi + \frac{1}{2}\ln|B| - \ln P_D + \ln(1-P_D) & \end{cases}$$

for prior data tracks

We see, then, that we need to specify three items of information in order to use this definition on the data association trees:

a) The state variable representation (4.1)

b) The false alarm density β_{FT}

c) The new target density β_{NT}

We next consider track association trees.

4.2 - TRACK ASSOCIATION TREES

Track association trees behave somewhat differently than data association trees. Here we hypothesize various combinations of data tracks from different nodes to produce possible target tracks. Assuming that each such combination produces a unique target track, we see that these trees have constant depth. When a new data track appears, new nodes are created by correlating the new track with existing tracks from other nodes. The process is a breadth expansion, rather than depth because the new hypotheses are not conditioned on the hypotheses already in existence. There is no need to consider false alarms or new targets at this level; these have already been determined at the data association level. We thus need only a method of distinguishing those combinations of data tracks that correspond to real targets. Those that do not correspond to real targets are termed ghosts.

The problem in setting up a general definition of probability here is that the probability of each node must be evaluated over a time interval. This differs considerably from the nodes in data association trees, whose probabilities are evaluated at points in time. If new nodes are added to an existing track association tree, then the probabilities of the new nodes will be evaluated over different time intervals than the older ones. The question then becomes one of comparing these probabilities.

Instead of computing probabilities, we will compute a likelihood function for each track hypothesis. Suppose we are using the data tracks from n nodes to form a track hypothesis. Define $\underline{z}(k)$ to be the augmented measurement vector, consisting of the measurement vectors of the n nodes. We can model the problem as follows:

$$\underline{x}(k+1) = A(k)\underline{x}(k) + B(k)\underline{w}(k)$$

$$\underline{z}(k,\theta) = C(k)\underline{x}(k) + \underline{v}(k)$$

$\underline{x}(k)$ here is the target state. The parameter θ denotes the dependence of $\underline{z}$ on the track hypotheses. To find the most likely hypothesis, we need to maximize the likelihood function for θ.

The problem in this form resembles the multiple model identification problem (see Van Trees [11]). Using the model, we can obtain a recursive relationship for the log likelihood function of θ. The result is

$$L(\theta,k+1) = L(\theta,k) + \frac{1}{2}\underline{\nu}(k,\theta)B^{-1}(k,\theta)\underline{\nu}^{T}(k,\theta)$$

where $\underline{\nu}(k,\theta)$ is the innovations at time k obtained from a Kalman filter associated with the track hypothesis identified with θ, and $B(k,\theta)$ is its covariance. With this relationship, we have a means of choosing the most likely track hypothesis.

The initialization of a track hypothesis occurs when a new data track appears out of the data association process at some node. The initialization becomes somewhat complex when the measurements are time delayed. In some cases, it may not be possible to initialize the correct track hypothesis. For instance, consider the situation in which a target is closer to a node A than another node B. Because of the signal propagation delay, the measurements at A will be more recent than those at B. Presumably, node A will detect the target first. If we attempt to correlate the new data track with existing data tracks at B, we see that the correct correlation will not be formed, since B has not yet detected the target.

With this in mind, we present the following scheme for the construction of track association trees. When a node initiates a new data track through its data association process, a new track association tree is also created. We attempt to initialize correlations with existing data tracks from the other nodes for a given length of time. After this period, no more attempts at initialization are made. In this manner, the correct track hypothesis should be formulated with the last node to detect the target. In order to compare hypotheses on different track association trees, we shall use time averaged values of likelihoods computed in the manner described above; i.e. we shall use $L(k,\theta)/\Delta t$, where Δt is the window length.

The utility of the above method will, of course, depend on the tracking system. In some cases, it may turn out that differentiating between ghosts and targets is virtually impossible. We then must either bring to bear other information in the system to distinguish the real targets, or we must continue to track the ghosts as targets. These issues will appear in our discussion in the next chapter.

4.3 - SUMMARY

In this chapter, we have presented a formulation for the calculation of probabilities of hypotheses on the data association trees and the track association trees. Armed with this knowledge, we can now proceed with the application of the algorithm to the two-node system. The necessary mathematics are developed in the next chapter.

CHAPTER 5

STATE SPACE MODELS FOR THE TWO NODE SYSTEM

In order to apply the formulation presented in Chapter 4, we need to develop state variable representations to implement the necessary Kalman filters. Our goal in this chapter is to develop the appropriate state space models for both the data association process and the track association process. Although we wish to obtain filters that perform reasonably well, we do not undertake a thorough examination of these state space models. As a consequence, some important issues are left unexplored. However, our main purpose is to demonstrate the tracking algorithm developed earlier; we do not propose to derive the optimum tracking filters for the specific tracking system under consideration. Hence, we content ourselves with a less detailed, but adequate, analysis focused on the data association aspect of the problem.

5.1 - STATE SPACE MODEL : THE DATA ASSOCIATION FILTER

The most natural way to define a state variable for this problem is the target state as defined in chapter 2. It is easy to see how the measurements are related to the target state. In Figure 5-1, ϕ is the noiseless measurement (the acoustic azimuth) at time k. Because the speed of sound is finite, this means that ϕ corresponds to a target position at time t sometime in the past. (This is called the acoustic target position to distinguish it from the true target position at time k). The relationship

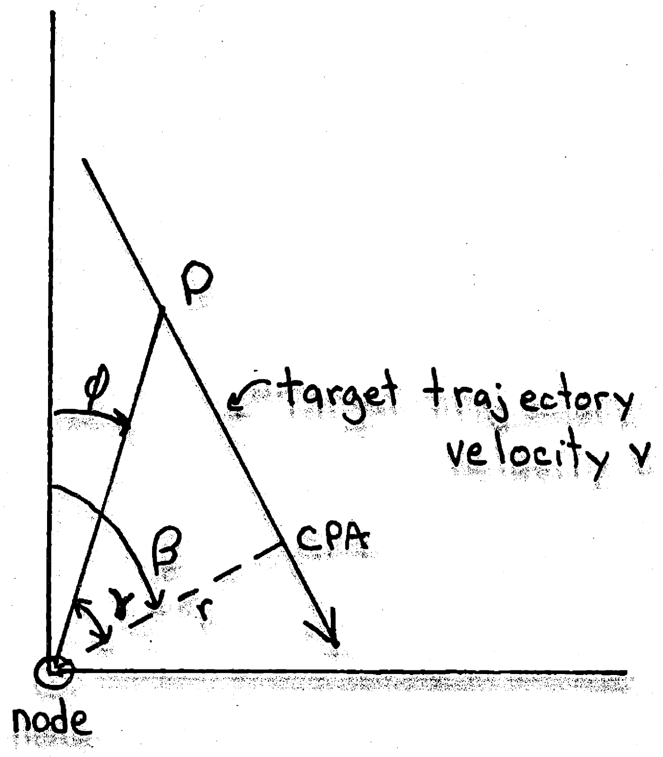


Figure 5-1.

between k and t is

$$k = t + \frac{R}{c} \tag{5.1}$$

where R is the range to the acoustic target position (the <u>acoustic range</u>) and c is the speed of sound. Now, let β be the acoustic azimuth (at time k_0) when the acoustic target position is at the closest point of approach (CPA). The time the target is actually at CPA is t_0. Let γ be the angle around CPA - that is the angle between the line to the apparent target position P and the line to the CPA. We see that

$$\phi = \beta + \gamma , \qquad -\frac{\pi}{2} < \gamma < \frac{\pi}{2} \tag{5.2}$$

Let us assume that the velocity v and the heading is constant. The distance between P and CPA is

$$d = v(t-t_0) \tag{5.3}$$

A negative distane implies the target is approaching CPA. We also have

$$d = r\tan\gamma \tag{5.4}$$

where r is the distance to CPA. Thus

$$v(t-t_0) = r\tan\gamma \tag{5.5}$$

or, substituting for t and t_0,

$$v(k - \frac{R}{c} - k_0 + \frac{r}{c}) = r\tan\gamma$$

$$\tan\gamma + \frac{vR}{cr} = \frac{v}{r}(k-k_0) + \frac{v}{c}$$

But $R = r\sec\gamma$, and so

$$\tan\gamma + \frac{v}{c}\sec\gamma = \frac{v}{r}(k-k_0) + \frac{v}{c}$$

Solving for γ, and substituting into (5.2), we finally obtain

$$\phi = \beta + \frac{\frac{v}{r}(k-k_0) + \frac{v}{c} \quad 1 + \sqrt{1 + \frac{v^2}{r^2}(k-k_0)^2 + \frac{2v^2}{rc}(k-k_0)}}{1 - \frac{v^2}{c^2}} \qquad (5.6)$$

The quantities k_0, $\frac{v}{r}$, $\frac{v}{c}$, and β provide full knowledge of the target state. Hence, we could choose these to be the state variables for the system model.

There are two major difficulties in using a state model in this form. First of all, it is difficult to obtain an estimate of β until past CPA. Because the governing equation (5.6) is nonlinear, we must resort to linearized version of the Kalman filter (called the extended Kalman filter or EKF). The quantity β is crucial for an accurate linearization. The best way to obtain an estimate of β is to set up a bank of filters, each conditioned on a different value of β. Based on the performance of these filters, we would then apply some sort of decision criteria to selsect the best filter. This has been done by Hebbert [5] for the case of a single target, no system noise, and an infinite signal propagation speed. In our case, there will already be several Kalman filters running, and replacing each of these with a bank of filters increases the memory and computation enormously.

The second major difficulty in using the full target state in our model is the resulting weak observability of the system. Theoretically, we can obtain information about every state variable from any given measurement. However, we get a relatively large amount of information about a particular state variable only at the expense of the other state variables.

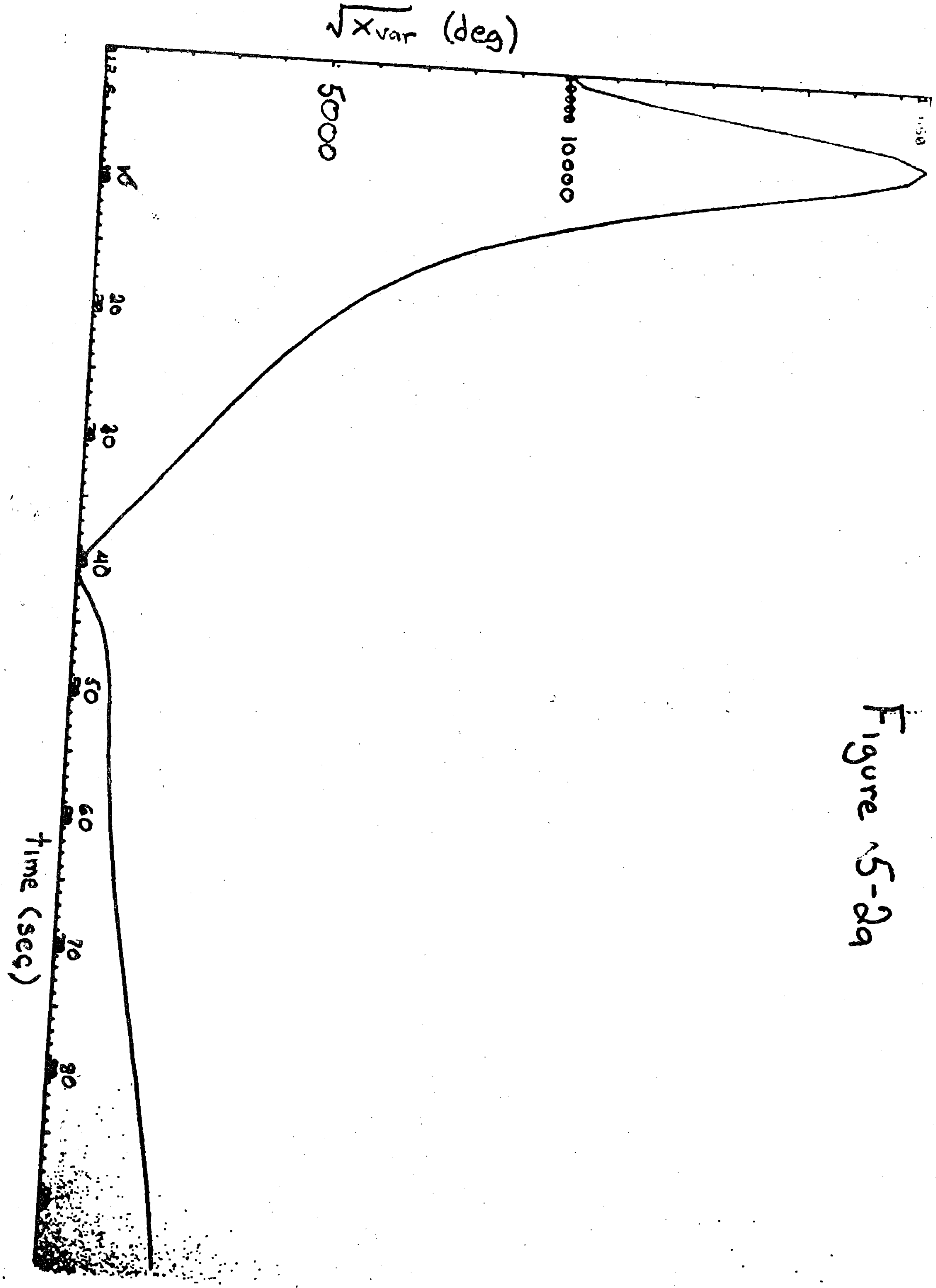


Figure 5-2a

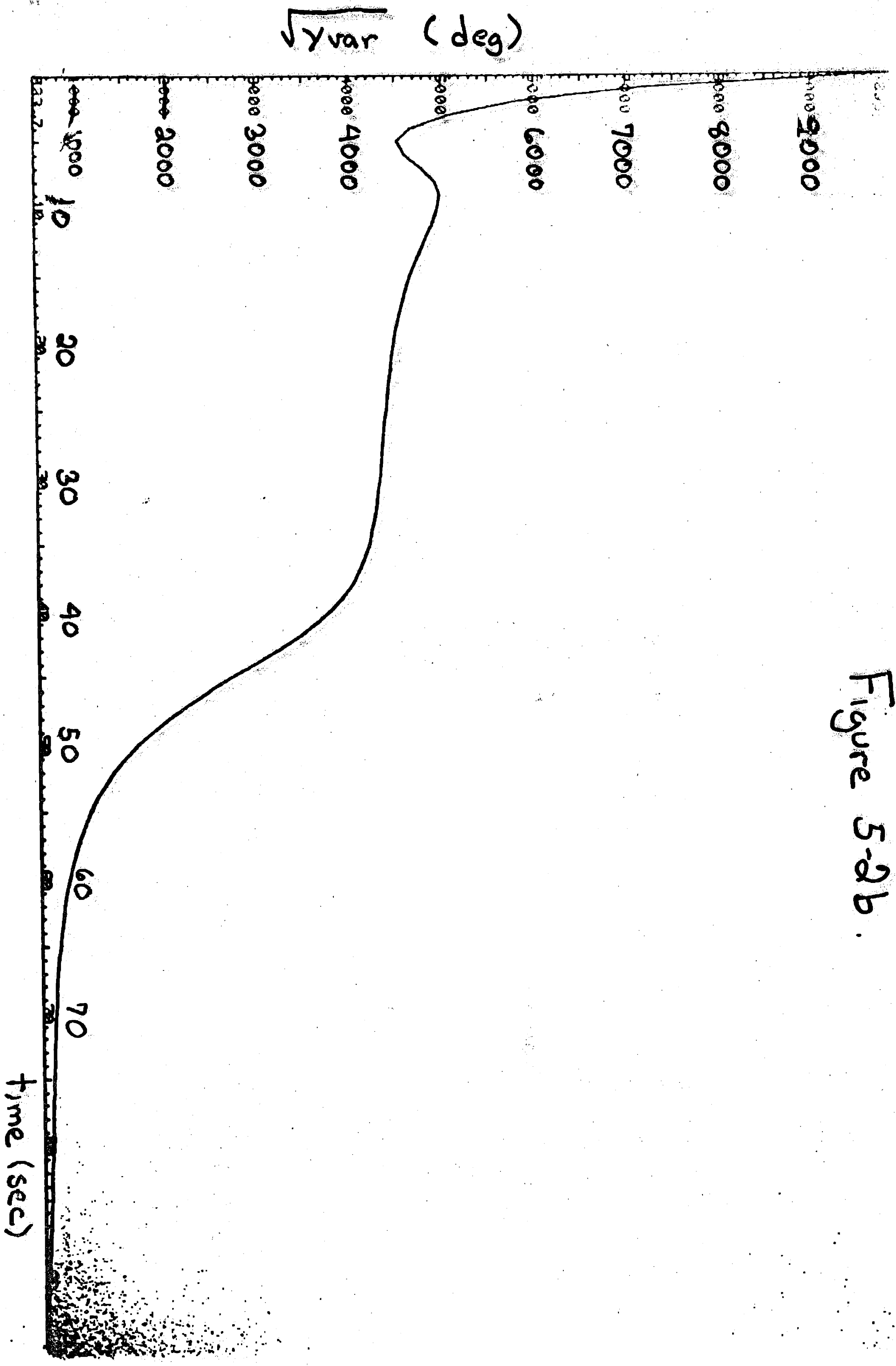

$\sqrt{Yvar}$ (deg)
1000
2000
3000
4000
5000
6000
7000
8000
9000
10
20
30
40
50
60
70
time (sec)

Figure 5-2b.

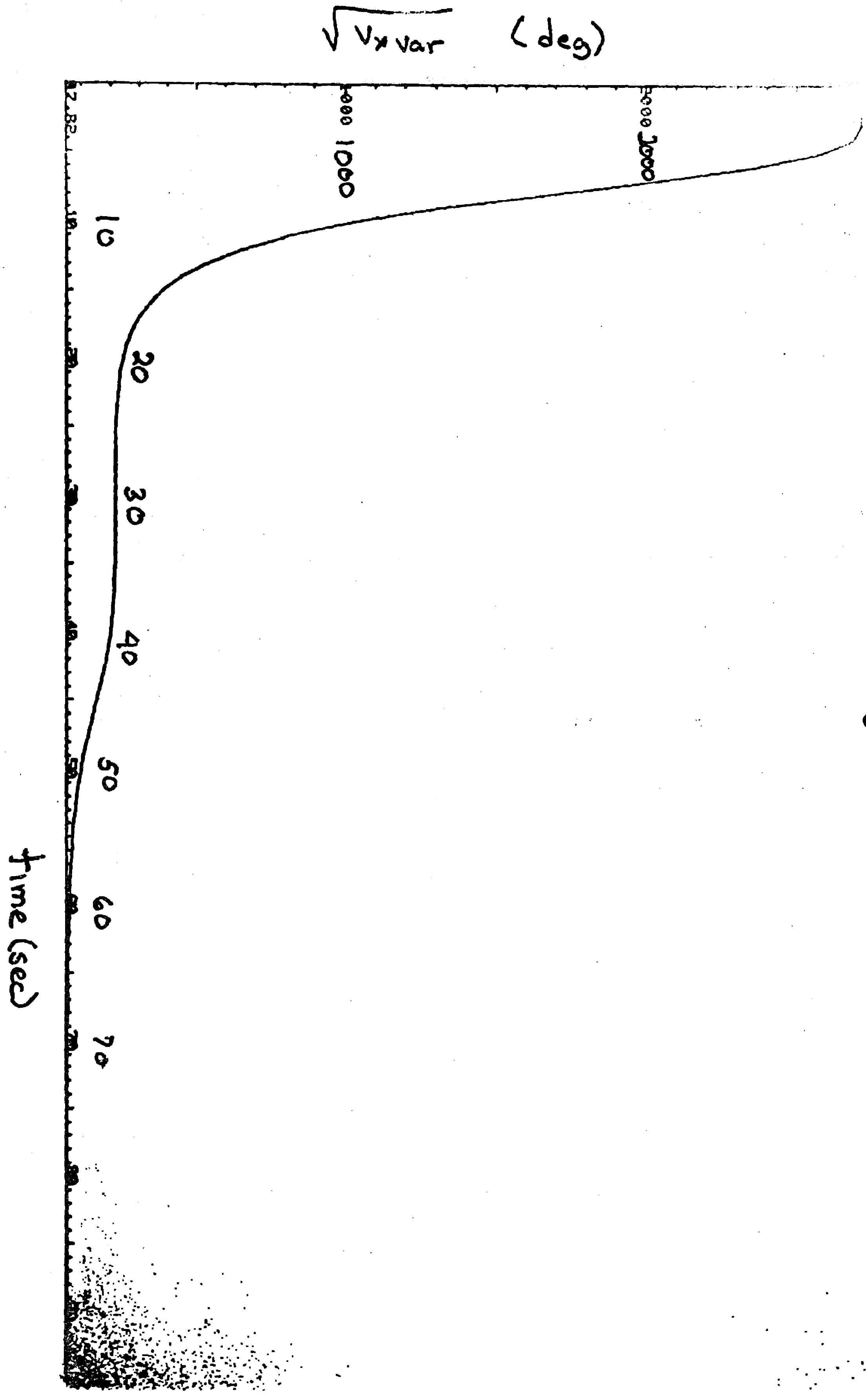
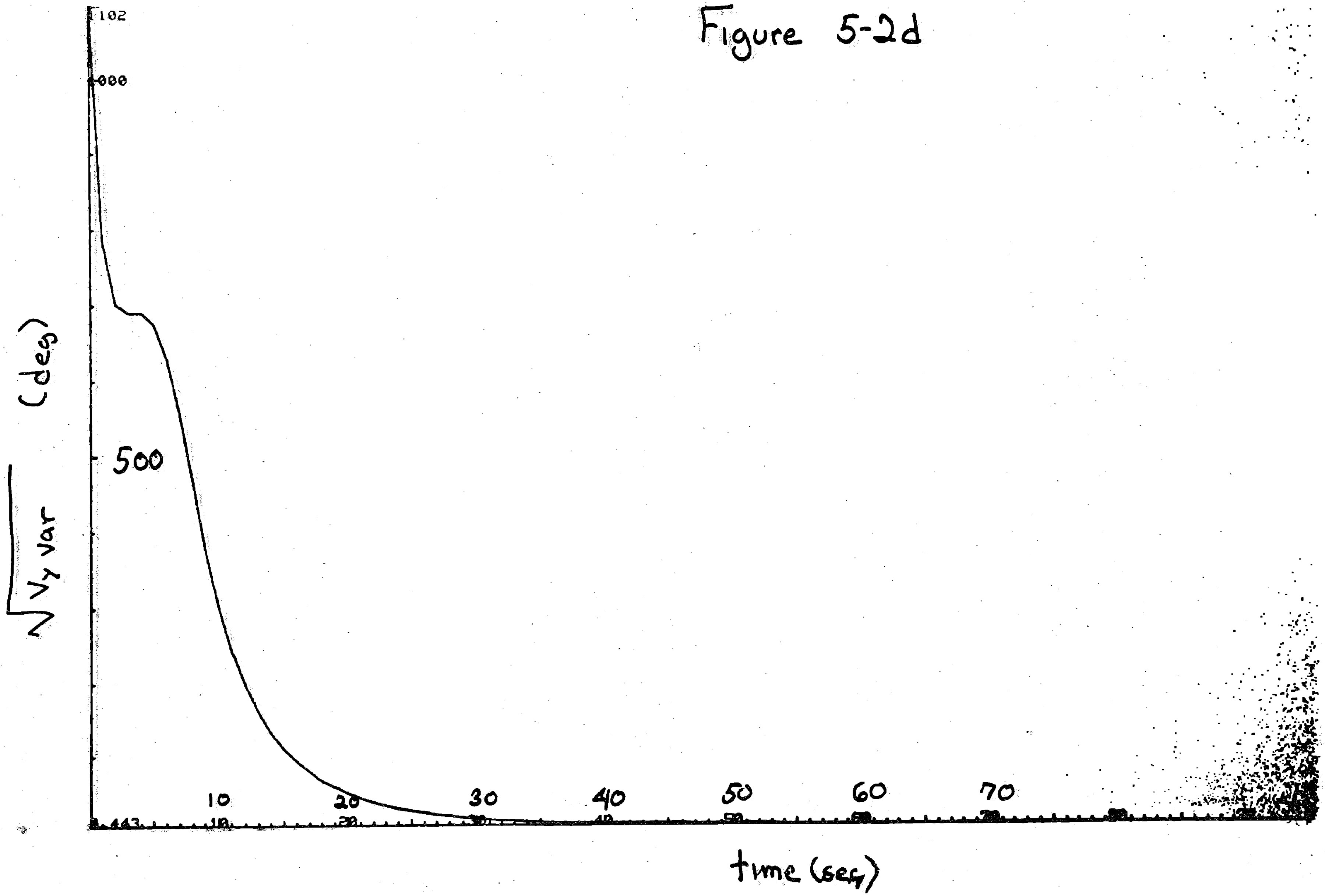


Figure 5-2c

Figure 5-2d


$\sqrt{V_y \text{ var}}$ (deg)
1102
1000
500
0.443
10
20
30
40
50
60
70
time (sec)

Essentially, the azimuth ϕ gives one position coordinate in a two dimensional system. We would expect, then, that we could not obtain substantial reductions in the uncertainty of both position coordinates is shown clearly in Figure 5-2(a-d). x is defined as the position coordinate parallel to a given target's trajectory, and y to be position coordinate perpendicular to the trajectory. v_x and v_y are the velocities in the x and y directions respectively. Figures 5-2(a-d) display the square root of the variances of x, y, v_x, and v_y as a function of time for a typical target trajectory. The variances are the diagonal elements of the error covariance matrix of an EKF tracking x, y, v_x, and v_y. The filter was always linearized about the exact trajectory. The measurement noise and the system noise in this filter were set to 0 (i.e., Q=R=0), and the initial covariance was set to a very large value. The filter thus starts off with essentially no a priori information, and all reductions of the covariance are solely due to the incoming measurements. The covariance matrix obtained is known as the Cramer-Rao lower bound, which implies that no tracking scheme can attain mean square errors less than those shown here. As can be seen, there is a rapid reduction in the variance of v_y while the target is still far from CPA (which occurs at t=45 seconds). The variance of v_x stays relatively high until the target approaches CPA. Thus, in the region approaching CPA, the full state tracker is at best weakly observable. The large uncertainty in at least one coordinate in this region could result in a poor linearization and filter divergence. Since we wish to perform data association in this area, we must reject the full target state as a model.

We thus must define our state variables in another way. One method is make a Taylor series approximation for ϕ, i.e.,

$$\phi(k+T) = \phi(k) + T\phi'(k) + \frac{T^2}{2}\phi''(k) + \frac{T^3}{6}\phi'''(k) + \ldots \qquad (5.7)$$

Our problem is to determine the appropriate number of terms to retain in the expansion. A typical plot of the acoustic azimuth vs. time is shown in Figure 5-3. Obviously, a linear approximation is not very good over the entire range, so the term ϕ'' should at least be retained. On the other hand, retention of too many terms can lead to a rather sluggish filter with a relatively high average error covariance. In addition, computation and memory requirements increase with the number of state variables. Therefore, we restrict our attention to two possibilities: the three-state vector

$$\underline{x}_A(k) = \begin{bmatrix}\phi(k)\\ \phi'(k)\\ \phi''(k)\end{bmatrix}$$

and the four-state vector

$$\underline{x}_B(k) = \begin{bmatrix}\phi(k)\\ \phi'(k)\\ \phi''(k)\\ \phi'''(k)\end{bmatrix}$$

Our system model is

$$\underline{x}(k+T) = F\underline{x}(k) + \begin{bmatrix}0\\0\\0\\1\end{bmatrix} w(k)$$

$$w(k) \qquad (5.8a)$$

where

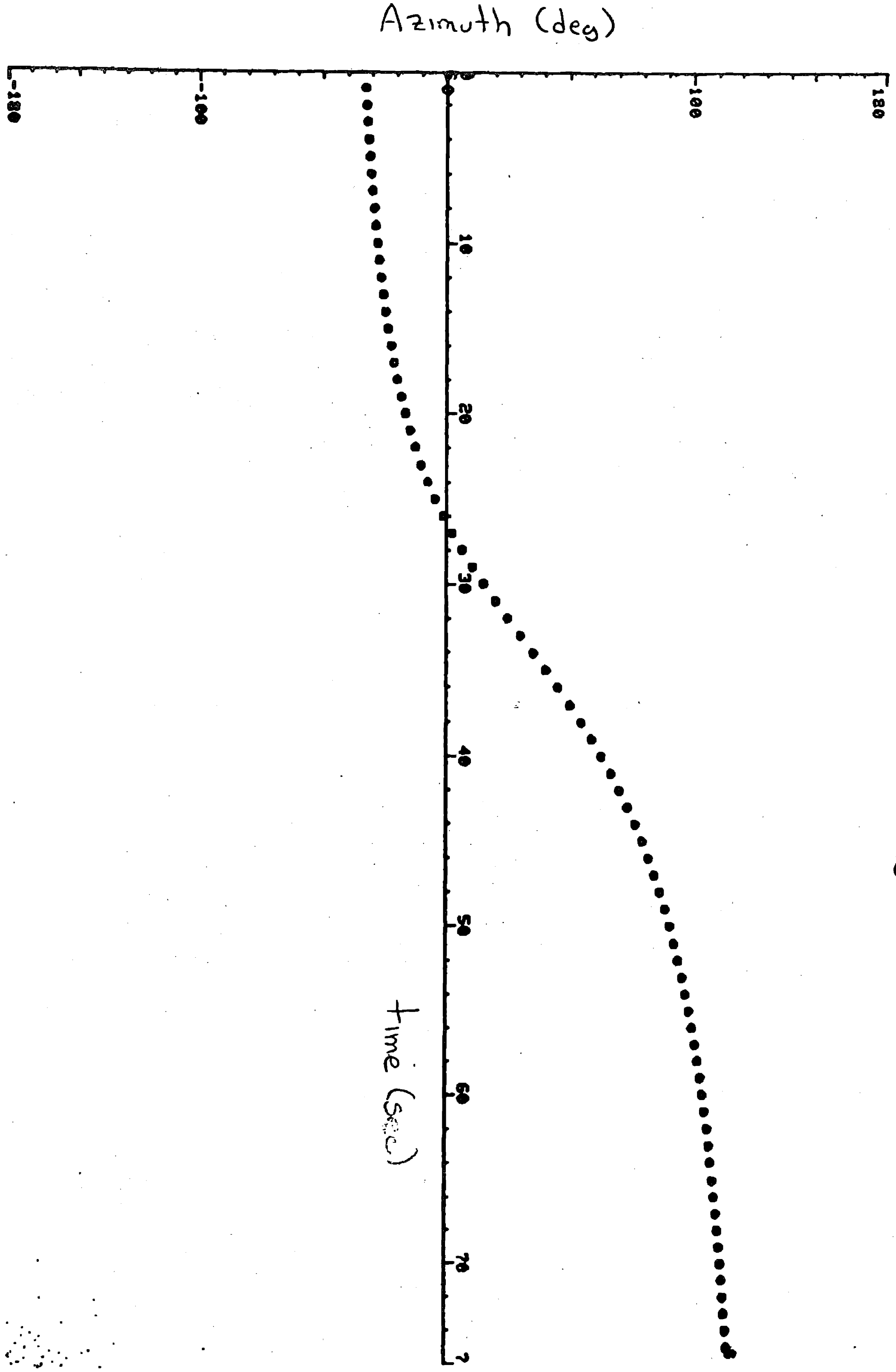


Figure 5-3

$$F = \begin{cases} \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, & \underline{x}(k) = \underline{x}_A(k) \\ \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & \frac{1}{6}T^3 \\ 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \underline{x}(k) = \underline{x}_B(k) \end{cases}$$

(5.8b)

We assume that w(k) is a scalar, zero-mean, white Guassian noise process with variance Q. Noise is entered only through the highest derivative for simplicity. The measurement equation for the system is

$$z(k) = H\underline{x}(k) + v(k) \qquad (5.9a)$$

with

$$H = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, & \underline{x}(k) = \underline{x}_A(k) \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, & \underline{x}(k) = \underline{x}_B(k) \end{cases} \qquad (5.9b)$$

v(k) is a scalar, zero -mean,white Guassian noise process independent of w(k) with variance R.

The above model does have a serious drawback, however. When the ratio $\frac{v}{r}$ is large, corresponding to small range and high velocity, the actual ø-curve approaches a step function, and its derivatives approach singularity functions. The model is very poor in these situations. In fact, the filter will not react as fast as the changes in these derivatives, and it may well lose the data track. Our solution to this difficulty is to bypass it, pointing out that it is easily detected from an azimuth history, and hence special mechanisms, which will not be developed here, can be invoked to deal with it. Thus, we will consider only target trajectories with reasonably small values of $\frac{v}{r}$.

In order to properly choose a good model, we test a number of possible values of Q in both the 3-state anu the 4-state filter over a range of typical trajectories. The trajecories were produced by varying the velocity and the distance to CPA, the two controlling parameters. Figure 5-3 shows the various combinations that were used. For each hypothesized model, a set of data from each target was produced by adding measurement noise of a standard deviation of 3^{o} to the exact acoustic azimuths. The Kalman filter corresponding to the model was run separately on each data set, and an average squared error (ase) between the true acoustic azimuth and the filter estimate was computed for each track. An overall ase was computed for each filter.

The results of the above Monte Carlo simulation are shown in Figure 5-4. It turned out that filter runs from two of the trajectories produced average squared errors that biased those computed from other trajectories. The first table shows the results if these runs were retained. The second run gives the computations if these runs are eliminated from consideration. As can be seen, the model with the minimum squared error is different in each table. The errors are close, however, indicating that there are several models that will give nearly the same performance. We choose the model indicated by the second table, because all of our example target trajectories will have $\frac{v}{r} < .1$. Thus, we choose the three state filter with Q=0.001 as our data association filter.

r(m) \ V (m/sec)	50	100	150	200
1000	x	x	bias2	bias1
2000	x	x	x	x
3000	x	x	x	x
4000	x	x	x	x

	Q	All data	Throw out bias1	Throw out bias1 & bias2
3 state	.03	5.82	4.58	4.54
	.01	5.23	3.94	3.61
	.003	4.95	3.58	3.35
	.001	6.03	3.79	3.23
	.0003	9.08	6.49	4.35
	.0001	10.64	8.44	5.88
4 state	.0001	6.72	4.37	4.06
	.00003	4.85	3.90	3.88
	.00001	6.30	4.85	4.42
	.000003	6.65	5.71	4.12
	.000001	8.86	6.58	5.58
	.0000003	7.97	6.48	5.51

Figure 5-4

One last item to consider is the initialization procedure for the Kalman filter based on (5.8) and (5.9). Assuming the model is reasonably observable, the filter will be relatively insensitive to the initial state. However, we can suggest a natural procedure. Far from CPA, the acoustic azimith will change very slowly. Therefore, we initialize the first component of the state to the first measurement received. The other components, which are derivatives of the first, are set to 0. The initial error covariance can be chosen somewhat more arbitrarily, since error covariance approaches a steady state value relatively quickly, with small values of Q and R. For the model as determined above (R=9.0,Q=0.001), it turns out the covariance reaches steady state after approximately n measurement updates.

With the above system model and initialization procedure, we can construct the Kalman filters necessary to implement the recursive probability formula developed in the chapter 4. This, in turn, gives us an implementation for the data association trees. We now turn to the problem of constructing system models for use in the track association filters.

5.2 - STATE SPACE MODEL : TARGET TRACKING FILTER

Developing a system model that produces a good filter to estimate the full target state from the data tracks of the two nodes is a much more difficult task. Conceptually, we can track the target state in two ways. We can track the acoustic position and velocity, or we can track the true position and velocity. We shall investigate the former method first.

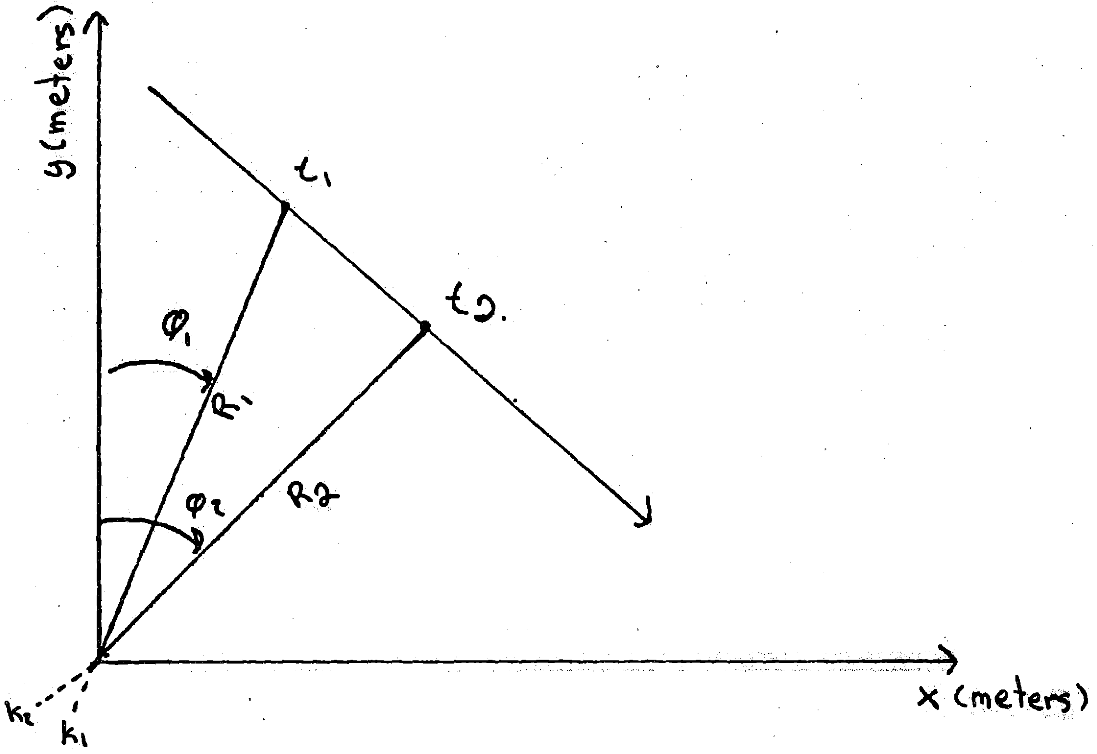


Figure 5-5

Consider Figure 5-5. Assume that the target has a constant velocity and constant heading. The node receives measurement ϕ_1, at time k_1, corresponding to a true target position at time t_1. The relationship between k_1 and t_1 is given by (5.1):

$$k_1 = t_1 + \frac{R_1}{c}$$

R_1 is the range to the acoustic target position. At time k_2, the node receives the measurement ϕ_2, and we have

$$k_2 = t_2 + \frac{R_2}{c}$$

Thus,

$$k_2-k_1 = t_2-t_1 + \frac{R_2-R_1}{c} \tag{5.10}$$

Even if we choose k_2-k_1, the sampling period, to be constant, the time difference between consecutive acoustic positions is constantly changing in a nonlinear fashion. Thus the system equation is nonlinear, even though the target trajectory is perfectly linear.

It turns out that the equations of the EKF for this model are very complicated. They turn out to be very sensitive to linearization. We can get an idea of this by reasoning as follows. While the target is approaching CPA (during which, it is hoped, the target will be acquired), the acoustic range decreases. According to (5.10), this means the time difference between acoustic positions is greater than the time difference between the corresponding measurements. Now, since the system equation describes the time evolution of the state, its nonlinearities will affect the time update equations of the EKF. One would expect, then, that the effects of these nonlinearities would be worse than expected for a constant (known) update period, at least while the target is approaching CPA. Indeed, it turns out that the filter is not only very sensitive to linearization, but that enhanced time difference between acoustic positions causes the error covariance matrix to shrink prematurely, reducing the influence of the current measurements.

Another difficulty in this approach is in the incorporation of the measurements into the filter. As discussed later, it turns out that the best method for measurement update enters measurements from each node independently into a single filter. At any given sampling time, however, the measurement from one node will not correspond to the same acoustic position as the measurement from the other node. We thus have the difficult task of performing two time updates, corresponding to each pair of measurements.

Our second approach is to track the true target state. In this case, the system model for a linear trajectory is also linear:

$$\underline{x}(k+T) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underline{x}(k) + G\underline{w}(k) \qquad (5.11)$$

T is now exactly equal to the sampling period. Because of this and the fact the equation is linear, we do not have the sensitivity in the time update as we did in the previous model.

We should point out that the derivation of the current measurements from the current target state does involve a degree of approximation, as the actual signal received at time k was produced by the target at time $k-\frac{R}{c}$. In the case of no process noise, (i.e., no target deviations from a constant course), this is not a formal difficulty as the state variables contain enough information to allow a position to be extrapolated backwards in time. Formal difficulties arise when the process noise is assumed (as it must be in this case for practical reasons), but the approximation that current measurements can be derived from the current state is minor compared with other approximations; e.g., ignoring altitude, multipath signal effects, etc.

We now turn to consideration of the measurement equation for the system (5.11). There are three ways to incorporate the measurements into the filter. One way would be to combine the azimuths in some fashion to form position measurements. This procedure is called crossfixing, and we would have a linear measurement equation. As discussed in the next section, however, the crossfixing is not always successful, and we could thus lose information for updating target state estimates.

The other two methods use the given measurements from the nodes independently. The first method would identically initialize two filters, and then run each filter on a different node's measurements. The estimates of each filter could then be combined, taking into account the common initialization. The second method would be run one filter and incorporate both measurements. Theoretically, both approaches should produce equivalent results. As pointed out in the last section, however, we do not obtain much of a decrease in the variance of the position coordinate perpendicular to the trajectory. This implies that this coordinate is relatively unobservable. We wil thus obtain large errors in the estimation of this component, which will in turn give poor linearization. Without a good linearizations, the EKF becomes unstable. For these reasons, we will use a single tracking filter that incorporates measurements from both nodes independently.

We now derive the measurement equation for the system (5.11). Consider the Figure 5-6. U is the true target position with coordinates (x, y) and P is the acoustic position with coordinates (x_P, y_P). ψ is the true target azimuth and ϕ is the acoustic azimuth from node S. Node S has coordinates (x_S, y_S). Let t be the time difference between P and U. Then the distance between these two points is vt, where v is the velocity of the target. In addition, t must also be the travel time of the signal from P to S. Hence, the distance between these points is ct. The angle θ is as shown, and is easily seen to equal $\alpha-\psi$, where α is the heading of the target. By the law of cosines we see that

$$(ct)^2 = (vt)^2+\rho^2-2vt\rho\cos\theta$$

where ρ is the true target range. Solving for t we get

$$t = \frac{-\rho v\cos\theta+\sqrt{\rho^2c^2-\rho^2\sin^2\theta}}{c^2-v^2} \tag{5.12}$$

Now, let v_x be the velocity in the x direction, and v_y be the velocity in the y direction. Then we have

$$\psi = \tan^{-1}\frac{x-x_S}{y-y_S}$$

(5.13)

$$\alpha = \tan^{-1}\frac{v_x}{v_y} \tag{5.14}$$

Since $\theta=\psi-\alpha$, we can derive the following using trigonometric identities:

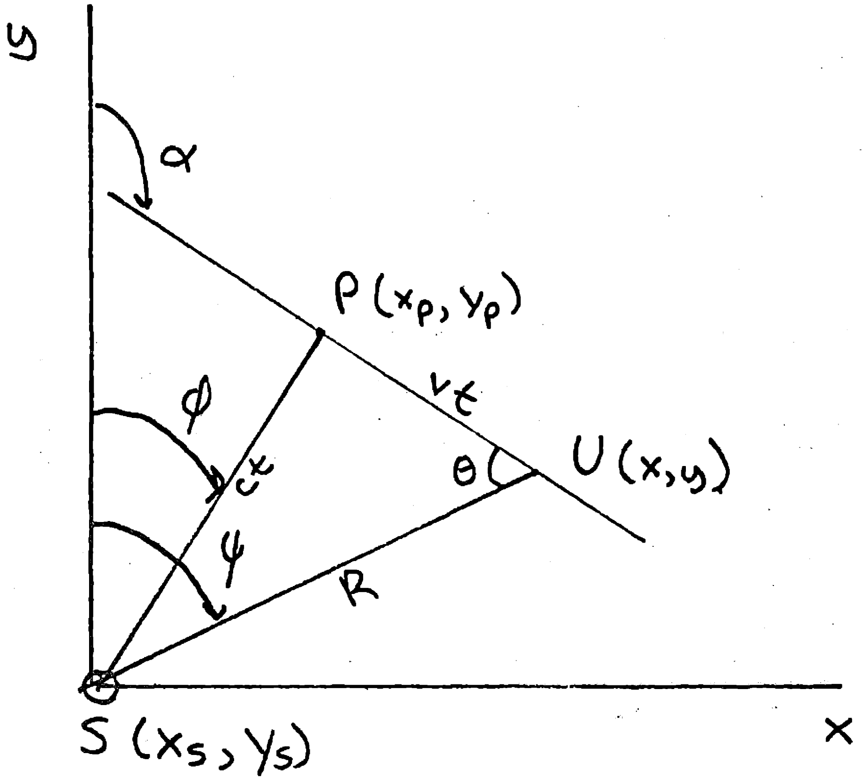


Figure 5-6

$$\cos\theta = \frac{xv_x+yv_y}{\rho v} \qquad (5.16)$$

$$\sin\theta = \frac{xv_y-yv_x}{\rho v} \qquad (5.16)$$

Substituting into (5.12) and simplifying, we get

$$t = \frac{-(xv_x+yv_y)+\sqrt{\rho^2c^2-(xv_y-yv_x)^2}}{c^2-v^2} \qquad (5.17)$$

Since ρ and v are functions of x, y, v_x, and v_y, (5.17) gives t completely in terms of the true target state. We now have

$$x_P = x-tv_x$$

$$y_P = y-tv_y$$

and

$$\phi = \tan^{-1}\frac{x_P}{y_P} = \tan^{-1}\frac{x-tv_x}{y-tv_y} \qquad (5.18)$$

Thus, our measurement equation is

$$z(k) = \tan^{-1}\frac{x(k)-t(k)v_x(k)}{y(k)-t(k)v_y(k)} + w(k) \qquad (5.19)$$

where w is the noise term and t is given by (5.12).

Since this equation is nonlinear, we will have to use an EKF. For this, we need the vector derivative

$$\frac{d\phi}{d\underline{x}} = \begin{bmatrix}\frac{\partial\phi}{\partial x} & \frac{\partial\phi}{\partial y} & \frac{\partial\phi}{\partial v_x} & \frac{\partial\phi}{\partial v_y}\end{bmatrix} \qquad (5.20)$$

The equations for the above partial derivatives can be derived by straightforward though tedious, calculus. The results are

$$\frac{\partial\phi}{\partial x} = \frac{y_p+(-yv_x+xv_y)\frac{\partial t}{\partial x}}{x_p^2+y_p^2}$$

$$\frac{\partial\phi}{\partial y} = \frac{-x_p+(-yv_x+xv_y)\frac{\partial t}{\partial y}}{x_p^2+y_p^2}$$

$$\frac{\partial\phi}{\partial v_x} = \frac{-ty_p+(-yv_x+xv_y)\frac{\partial t}{\partial v_x}}{x_p^2+y_p^2}$$

$$\frac{\partial\phi}{\partial v_y} = \frac{tx_p+(-yv_x+xv_y)\frac{\partial t}{\partial v_y}}{x_p^2+y_p^2} \qquad (5.21)$$

where

$$\frac{\partial t}{\partial x} = \frac{-v_x + \frac{xc^2+v_y(yv_x-xv_y)}{D}}{c^2-v^2}$$

$$\frac{\partial t}{\partial y} = \frac{-v_y + \frac{yc^2+v_x(xv_y-yv_x)}{D}}{c^2-v^2}$$

$$\frac{\partial t}{\partial v_x} \quad \frac{-x + \frac{y(xv_y-yv_x)}{D} + 2v_xt}{c^2-v^2}$$

$$\frac{\partial t}{\partial v_y} \quad \frac{-y + \frac{x(yv_x-xv_y)}{D} + 2v_yt}{c^2-v^2}$$

$$D = \sqrt{\rho^2c^2-(yv_x-xv_y)^2} \qquad ((5.22)$$

The equations for the EKF are of exactly the same form as (4.3), except that here the measurement matrix is

$$H(k) = \left.\frac{d\phi}{d\underline{x}}\right|_{\underline{x}=\hat{\underline{x}}(k|k-1)}$$

The only thing left to specify here is the initialization procedure. We will initialize the position and velocity of the filter at one point in time. This will be done using the smoothed estimates of the azimuths and their derivatives at each node. The procedure involves crossfixing, which is the topic of the next section. A poor initialization can cause problems in two ways. If the filter is initialized when the target is far from CPA of both nodes, the trajectory will be unaffected by the measurements for a relatively long period of time. The errors in initialization will thus grow with time. In addition, if the estimated target position is much closer to one of the nodes than the true target position, the filter will give greater weight to the measurements from that node than it should. These effects cascade, resulting in filter divergence.

With these points in mind, we formulate the following restrictions on the initialization procedure. First of all, we use two azimuths to calculate a position only if their difference is greater than 60°. This ensures that we do not initialize when the acoustic position is too far away. Second, we artificially adjust the initial heading so that it points toward the midpoint of the line segment joining the two nodes. This helps keep the initial position away from either node. This procedure, of course, works well only for trajectories that pass between the nodes. However, this is not much of a limitation if we consider the two nodes as part of a larger network. In such a system, it is reasonable to assume that the interesting targets will eventually pass between some pair of nodes. In addition, a node pair may receive an initial state for a target that is being tracked by other nodes. This procedure, called target handoff, is important in a system with a large number of nodes. In these systems, initialization is required only if data tracks cannot be associated with a priori targets. This reduces the number of initializations a node pair must perform.

5.3 - CROSSFIXING AND NON-CLASSICAL GHOSTS

As mentioned in the last section, crossfixing is the procedure for obtaining position measurements from the azimuth measurements of the two nodes. For time-delayed measurements, the process is more complex than simple triangulation. Suppose that time target position is further away from node A than node B. The signal will thus reach B first. We cannot obtain a position using this measurement at B because the corresponding measurement at A has not yet arrived. On the other hand, if we use the current measurement of A, the corresponding measurement of B is a past measurement. Thus, crossfixing can only produce acoustic positions corresponding to measurements from the node farthest away from the target. Given the current measurement from one node, we can divide the crossfixing operation into two steps. The first step determines the measurement and its time of reception at the second node that corresponds to the same acoustic position as the given measurement. The second step is standard triangulation. Since the first step is the real problem here, we will investigate it here. The derivation follows [4].

Consider Figure 5-7. Suppose that we receive, at time t_B the azimuth ϕ_B at node B. We wish to find the measurement ϕ_A received by node A at time t_A that corresponds to the same acoustic position as ϕ_B. Let

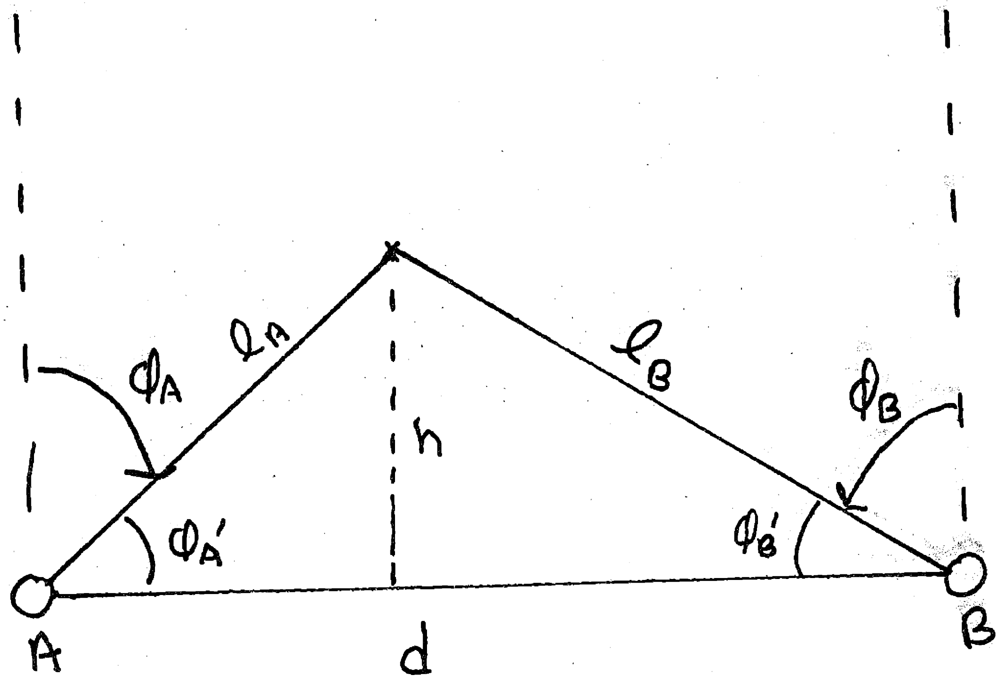


Figure 5-7

$$\delta t = t_B - t_A \tag{5.23}$$

and define t_P to be the time at which the true target position was the same as the desired acoustic position P. Then

$$\delta t = (t_B - t_P) - (t_A - t_P) \tag{5.24}$$

Now, $t_B - t_P$ is the travel time of the signal between P and B, while $t_A - t_P$ is the travel time of the signal between P and A. From the geometry of Figure 5-7, the following is evident:

$$l_A = \frac{h}{\sin\phi_A'} = \frac{h}{\cos\phi_A} \tag{5.25}$$

$$l_B = \frac{h}{\sin\phi_B'} = \frac{h}{\cos\phi_B} \tag{5.26}$$

(Remember ϕ_B is a negative quantity in this figure). Also,

$$t_A - t_P = \frac{l_A}{c} \tag{5.28}$$

$$t_B - t_P = \frac{l_B}{c} \tag{5.29}$$

From (5.25) and (5.26) we get

$$l_B = l_A \frac{\cos\phi_A}{\sin(\phi_A - \phi_B)} \tag{5.30}$$

Substituting this into (5.27) and solving for l_A, we get

$$l_A = \frac{d \cos\phi_B}{\sin(\phi_A - \phi_B)} \tag{5.31}$$

Also,

$$l_B = \frac{d \cos\phi_A}{\sin(\phi_A - \phi_B)} \tag{5.32}$$

Using (5.24), (5.28), (5.29), (5.31), and (5.32), we finally get

$$\delta t = \frac{d}{c} \cdot \frac{\cos\phi_A - \cos\phi_B}{\sin(\phi_A - \phi_B)}$$

(5.33)

Although we have used a restricted geometry here, the result is more general. The equation is valid, of course, only in the range where the measurements ϕ_A and ϕ_B intersect; namely,

$$\text{I.} \quad -\frac{\pi}{2} < \phi_B < \frac{\pi}{2} \quad , \quad \phi_B < \phi_A < \frac{\pi}{2}$$

$$\text{II.} \quad -\frac{\pi}{2} < \phi_B \leq -\pi \quad , \quad \frac{\pi}{2} < \phi_A < 2\pi+\phi_B$$

$$\text{III.} \quad \frac{\pi}{2} < \phi_B \leq \pi \quad , \quad \frac{\pi}{2} < \phi_A < \phi_B$$

For $\phi_B = \pm\frac{\pi}{2}$, the acoustic position is indeterminate.

Now, suppose that we have the past history of received measurements at node A (i.e., the azimuth vs time curve for A) as in Figure 5-8. Using (5.32) we can convert the single observation ϕ_B at node B into a curve of possible ϕ_A vs δt and plot it on the same scale (Figure 5-9). This curve is called the reflected observation curve. The intersection of these two curves gives the desired measurement ϕ_A and its reception time t_A.

It turns out that multiple intersections of the two curves is possible. This occurs when the target trajectory passes between the two nodes. Figure 5-10 shows an example. At time instants one second apart we have taken the noiseless measurements at B, and obtain all possible acoustic positions assuming a full past history of noiseless measurements at A. The target velocity was 150 meters per second at a heading of 135 degrees from north, and the distance between A and B is 5000 meters. In the figure, the real acoustic trajectory is obvious. The false positions form a track that flies from the perpendicular bisector of the line segment joining A and B

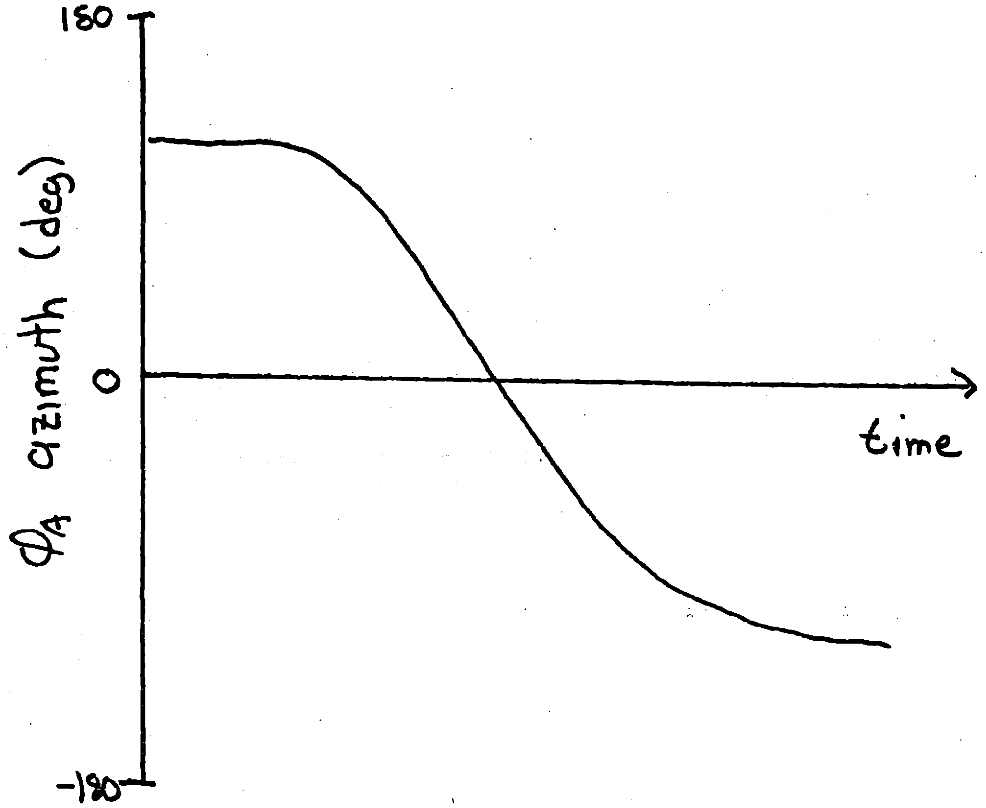
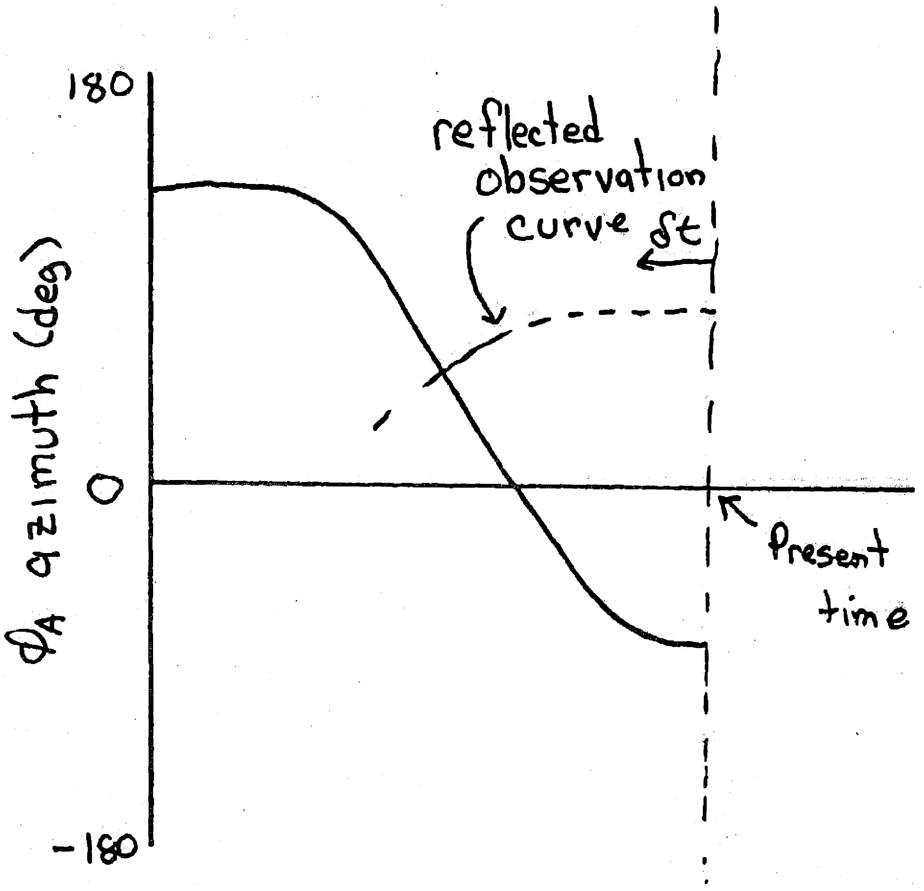


Figure 5-8


180
0
-180
ϑ_A azimuth (deg)
reflected observation curve
δt
Present time

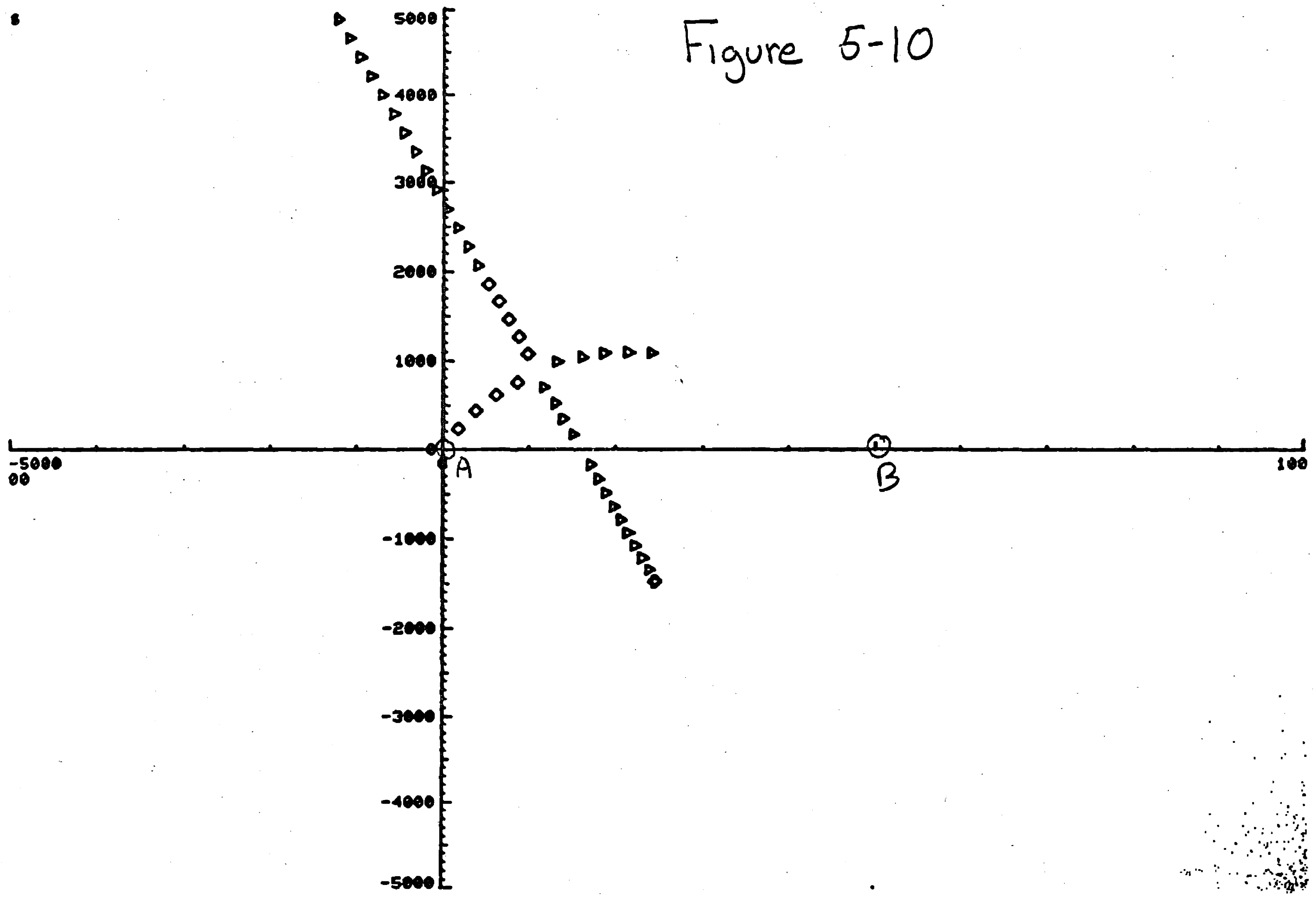

5000
4000
3000
2000
1000
0
-1000
-2000
-3000
-4000
-5000
-5000
00
100
A
B

Figure 5-10

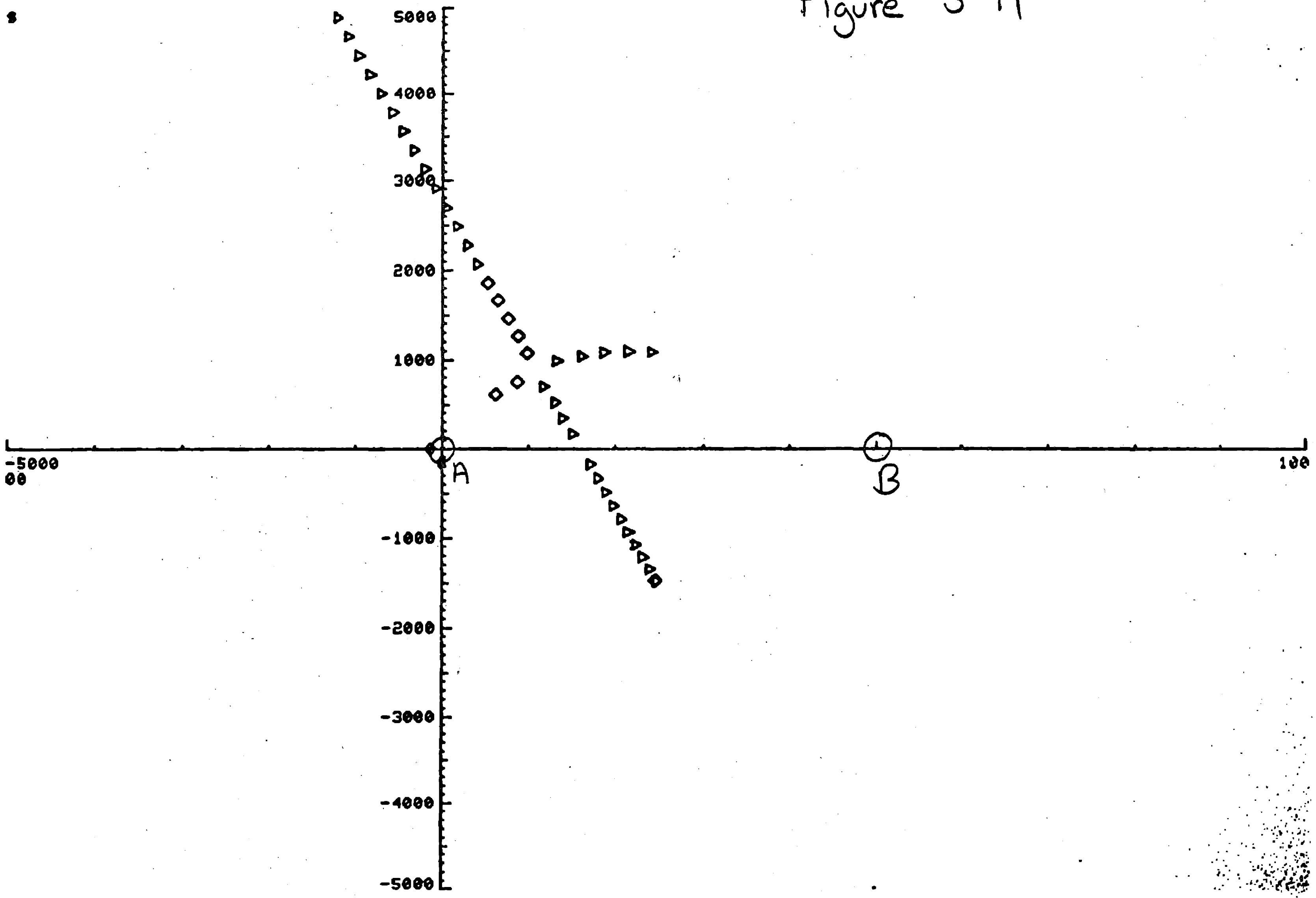
to node A. This behavior is exhibited by all such false tracks, and they occur only on the incoming path. We call these tracks non-classical ghosts, to distinguish them from the ghosts of Chapter 4.

A heurestic procedure can be developed which eliminates, for noiseless measurements, most traces of ghost tracks. Looking again at Figure 5-10, we see that the ghost track is concave downward. (The track begins at the bisector and heads toward A). Since the target heading is 135 degrees, the acoustic azimuth increases with time. However, succeeding positions along the ghost track requires the acoustic azimuth to decrease with time. Thus, the measurements of A that determine the ghost track forms a sequence that goes backwards in time. At each point in time, then, after crossfixing has been completed, we eliminate the past history at A that occurs prior to the earliest crossfix, then the rest of the ghost track will be eliminated. If it is not, the ghost track remains. Figure 5-11 shows the results for this example. As can be seen, only part of the ghost track is eliminated.

We can eliminate the rest of the ghost by observing that this portion of the ghost track must correspond to the later crossfix, else it would have been eliminated. We thus keep only the earliest crossfixes, which correspond to the real acoustic positions. The result, with a few minor adjustments that need not concern us here, are shown in Figure 5-12. Note that the procedure makes one expected error -- the first point at which the ghost is the earliest crossfix.

In the noisy case, deghosting is somewhat more difficult, primarily because crossfixing is not always successful. An important thing to notice

Figure 5-11


5000
4000
3000
2000
1000
-1000
-2000
-3000
-4000
-5000
-5000
00
100
A
B

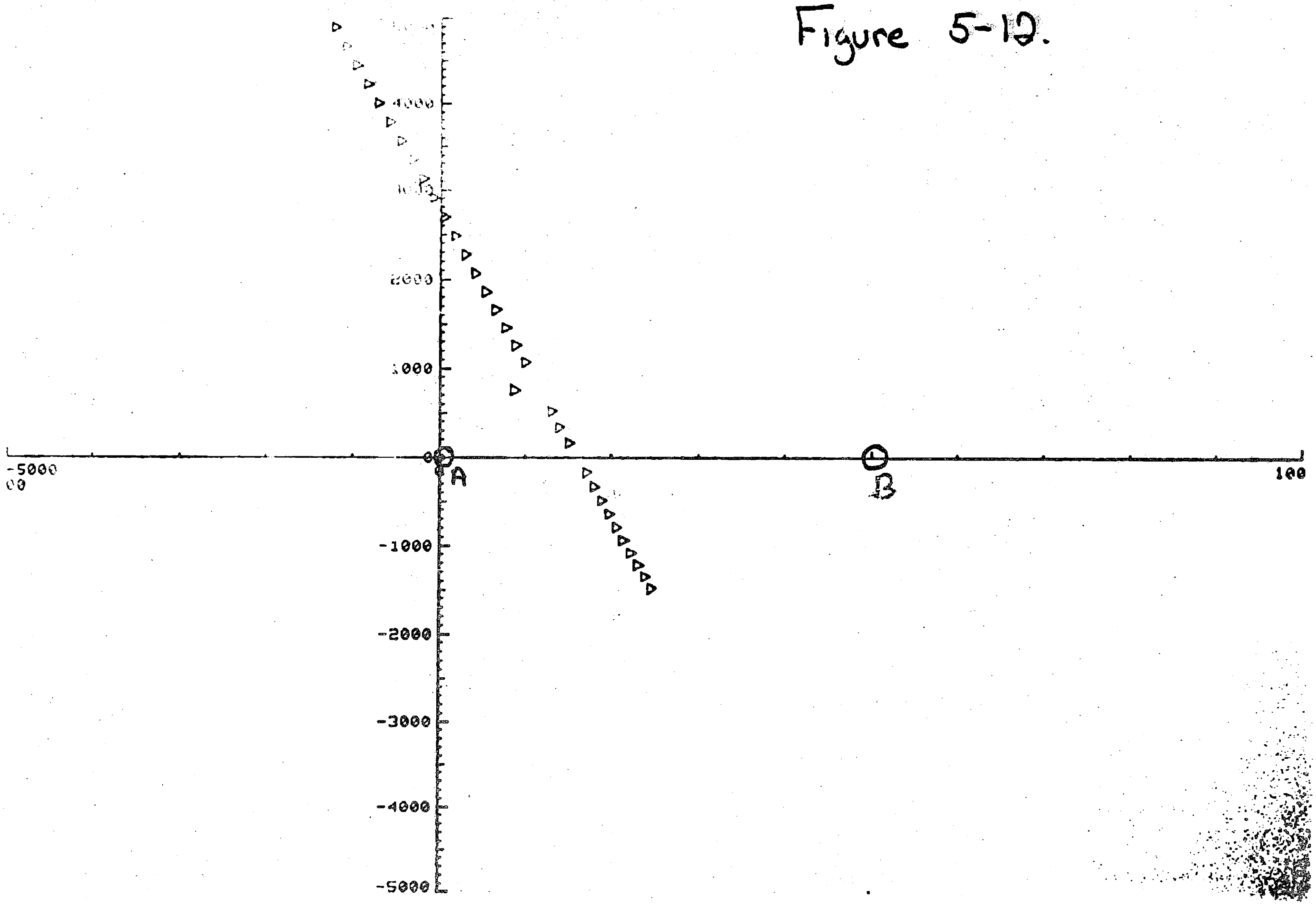


Figure 5-12.

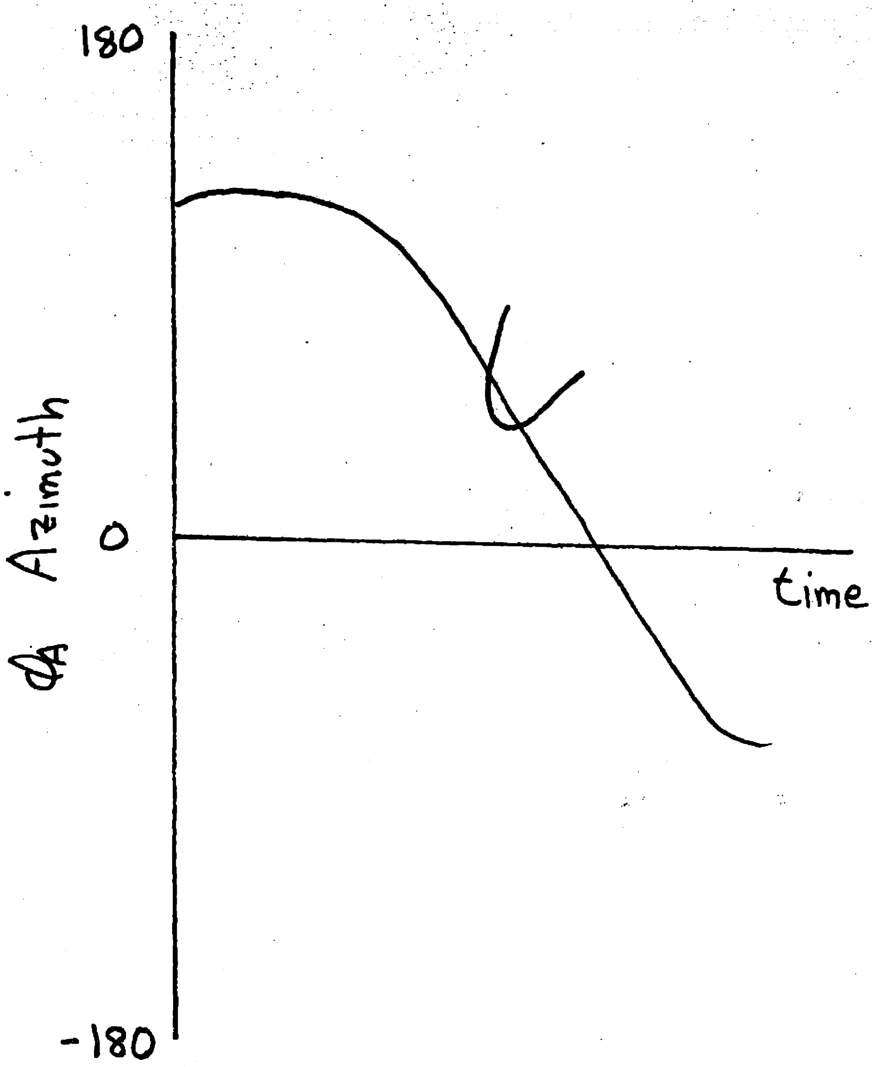


Figure 5-13

in Figure 5-12 is the gap that occurs in the region where the ghost track and the target track intersect. In this region, the reflected observation curve is almost tangential to the azimuth time curve; the intersection points are close together. This is shown in Figure 5-13 . In cases like these, it is very difficult to detect the intersection points. This is why the gap appears in Fig. 5.12; the curves were almost tangential and the crossfix failed. The situation is even worse when the measurements are noisy. The gaps can be many time-sampling periods long. In addition to this, the noisy data can cause random crosfixing failures anywhere along the path. Because of this loss of information, we choose not to use position measurements as updates to our filters. We will, however, use crossfixing in initialization.

5.4 - SUMMARY

In this chapter, we have developed state variable descriptions necessary for the construction of Kalman filters for both the data association trees and track association trees. We also discussed the operation of crossfixing and non-classical ghosts. In chapter 6, we shall present some results for this implementation of our tracking algorithm.

CHAPTER 6

RESULTS

In this chapter, we present demonstrations of the filters developed in Chapter 5 and their application to the tracking algorithm. It should be pointed out that these examples only highlight basic characteristics. In order to evaluate the strength of the filters, Monte Carlo simulations should be conducted over a wide range of target scenarios, and performance statistics must be defined and computed to facilitate the evaluation. Unfortunately, due to time constraints, we were unable to give a really thorough performance evaluation. We will, however, present some results that illustrate the characteristics of the filters and the algorithm.

All of the demonstrations in this chapter were done by computer simulation. The simulations, as well as the algorithm itself, was written in the C language and run on a PDP-11 computer.

In the first example, a Monte Carlo simulation was run on the data association filter to provide some information on its performance. The parameters of the example trajectory are those of Target 1 in Table 6-1. The node A at which the data association is carried out is positioned at the origin. A set of noiseless acoustic azimuths was generated for A, assuming a sampling period of one second. Only azimuths that corresponded to angles around CPA that were between -70° and 70° were retained. From this azimuthal set, 50 sets of noisy data were generated by adding white Guassian noise with a standard deviation of 3°. The data association filter was run separately on each set of data. For each point in time, an average mean square error was calculated over the 50 data sets. The results are shown in Table 6-2. Clearly, the filter performance is best far from CPA, when the azimuth-time curve is practically linear. As the acoustic data approaches the CPA angle, however, the azimuth derivatives are in flux, and performance degrades. Still, the overall average squared error was reasonable, about half the variance of the measurement noise.

The next example is an excellent demonstration of the capabilities of the data association process in a two-target environment. The trajectories are those of Target 1 and Target 2 in Table 6-1. Again, the reference node A is at the origin. Figure 6-1 shows the exact acoustical data generated by these trajectories. Note that the curves intersect twice, and that the angles of intersection are somewhat small. These regions are potential trouble spots for the data association process in the presence of noise. The node must be able to distinguish the noisy data so that they can be tracked. Otherwise, the target tracking process will not be good. Figure 6-2 shows the noisy data after the data association, while Figure 6-3 plots the filtered estimates. Although a few measurements around the curve intersections points are mixed up, the process manages to lock onto and track separately the two sets of data.

The next example demonstrates the target tracking filter. A second node B is placed at the point (5000, 0). Noisy sets of data are generated for both A and B from the trajectory of Target 1. The resulting target track, combining the two data tracks, is shown in Figure 6-4. Because of the restrictions imposed in Section 5.2, the target was well along the trajectory before the track was initialized. The plot is a bit misleading, in that the points of the track do not lie in sequential order following the true trajectory. This because the estimate of the current target state is updated by current measurements, which correspond to earlier states of the target. There always exists an uncertainty in the time of occurence of the earlier states; hence, one would expect consecutive estimates would not necessarily follow a sequentially follow the true trajectory.

Our last example demonstrates track association with the data tracks of the second example. Figure 6-5 shows the results. The ghost tracks (i.e., the incorrect track associations) turned out to be not much of a problem. One ghost did not even initialize a track. The other ghost in due course surpassed the speed of sound, and the algorithm automatically rejected it.

The peculiar behavior of the track of Target 2 is easily explained in terms of the initialization procedure outlined in chapter 5. The initial trajectory was set to head towards the point (2500,0). The track more or less held this direction until the target approached CPA. Then the measurements from both nodes began to have effect, and the track was subsequently pulled back around the actual trajectory.

As we mentioned earlier, the examples only highlight the major characteristics of the tracking algorithm in the two-node system. We did provide some statistics on the performance of the data association filter, pointing out its weaknesses, and demonstrated its success in resolving long term ambiguities. We also demonstrated the full target state filter and the track association process. However, much work still remains to be done in order to fully test the tracking algorithm. Measurements of performance of the full state tracker need to be evaluated. Also, the performance of the data association filter as a function of the rate of false alarms should be investigated to provide a measure of robustness.

The application of these ideas to larger distributed networks is also a further area of research that must be investigated. This would probably require a theoretical structue that combined various local target tracks into global tracks suitable for the users of the tracking system. The effect of communications, problems with maneuvering targets and target handoff procedures are other areas that will require careful research. Finally, we mention the importance in testing the algorithm in a real world system to investigate its real utility.

What we have attempted here is to show that our tracking algorithm does perform reasonably well in tracking multiple targets and resolving ambiguities in data associations. It is hoped that it will be a useful tool for future investigation into distributed processing tracking systems.

EXAMPLE TARGET TRAJECTORIES

	Initial Position	Velocity	Heading
1	(-2000,4000)	150	135°
2	(1500,5000)	100	180°

TABLE 6-1

Time	Ase
1.000000	5.786116
2.000000	5.155712
3.000000	4.166945
4.000000	3.396970
5.000000	4.236127
6.000000	5.138920
7.000000	3.986259
8.000000	3.498243
9.000000	3.422858
10.000000	3.055617
11.000000	3.936506
12.000000	3.323135
13.000000	3.183666
14.000000	3.118470
15.000000	3.567365
16.000000	3.879807
17.000000	3.274875
18.000000	3.597581
19.000000	2.931094
20.000000	2.566012
21.000000	3.238309
22.000000	4.230144
23.000000	5.340903
24.000000	6.974324
25.000000	8.431169
26.000000	10.123829
27.000000	12.403783
28.000000	14.933562
29.000000	14.682765
30.000000	16.610709
31.000000	14.077408
32.000000	12.717192
33.000000	12.456069
34.000000	9.190056
35.000000	6.202578
36.000000	3.948690
37.000000	3.596761
38.000000	2.955395
39.000000	2.396943
40.000000	2.298342
41.000000	2.669751
42.000000	3.128651
43.000000	3.242486
44.000000	3.337838
45.000000	2.704929
46.000000	3.012638
47.000000	3.110344
48.000000	3.165179
49.000000	3.049241
50.000000	2.737964
51.000000	2.581655
52.000000	2.593394
53.000000	2.477546
54.000000	2.695481
55.000000	3.742409
56.000000	3.483666
57.000000	2.838047
58.000000	1.960729
59.000000	2.285217
60.000000	2.821533
61.000000	2.750949
62.000000	2.671629
63.000000	3.251803
64.000000	2.034057

TABLE 6-2

Figure 6-1

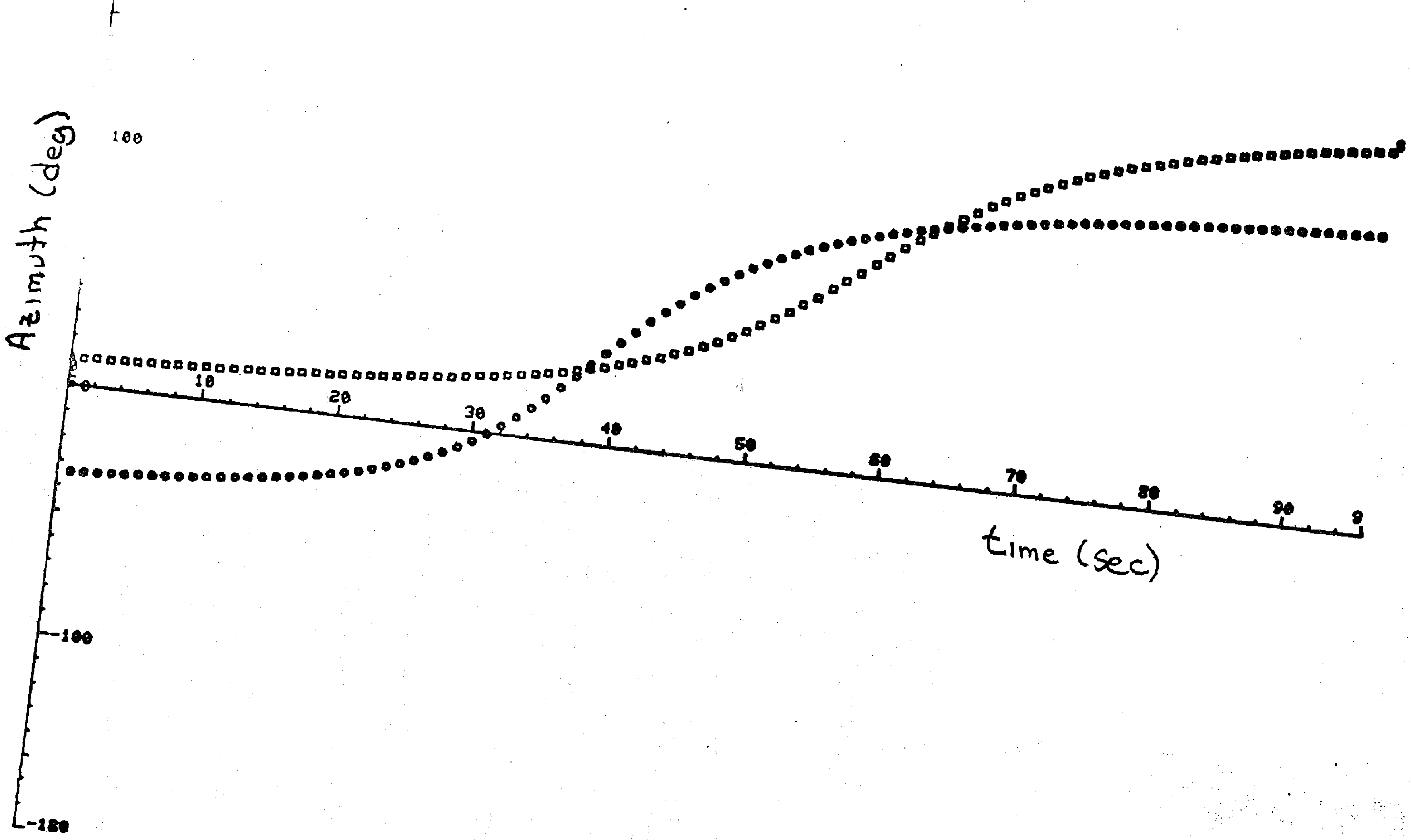


Figure 6-2

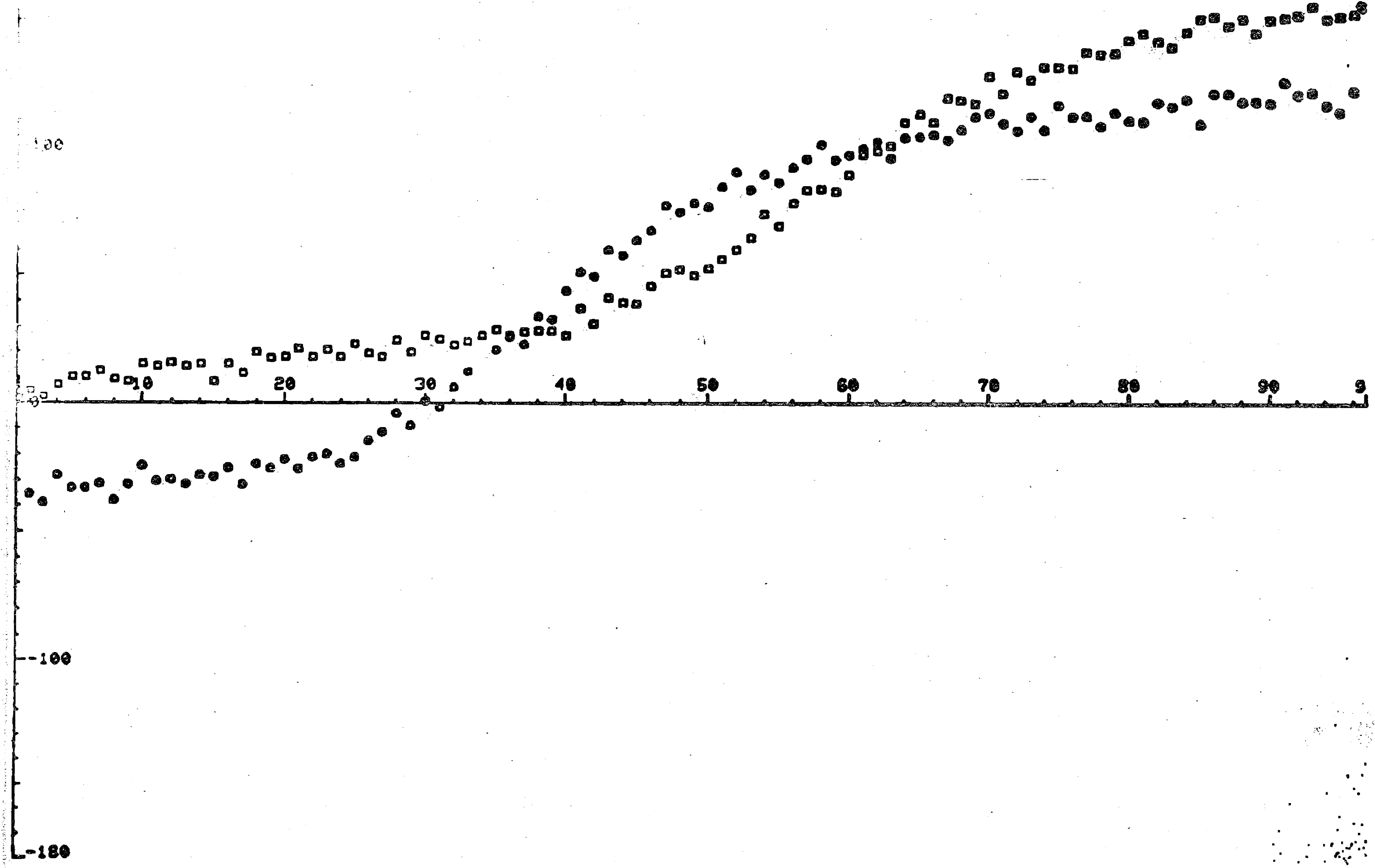
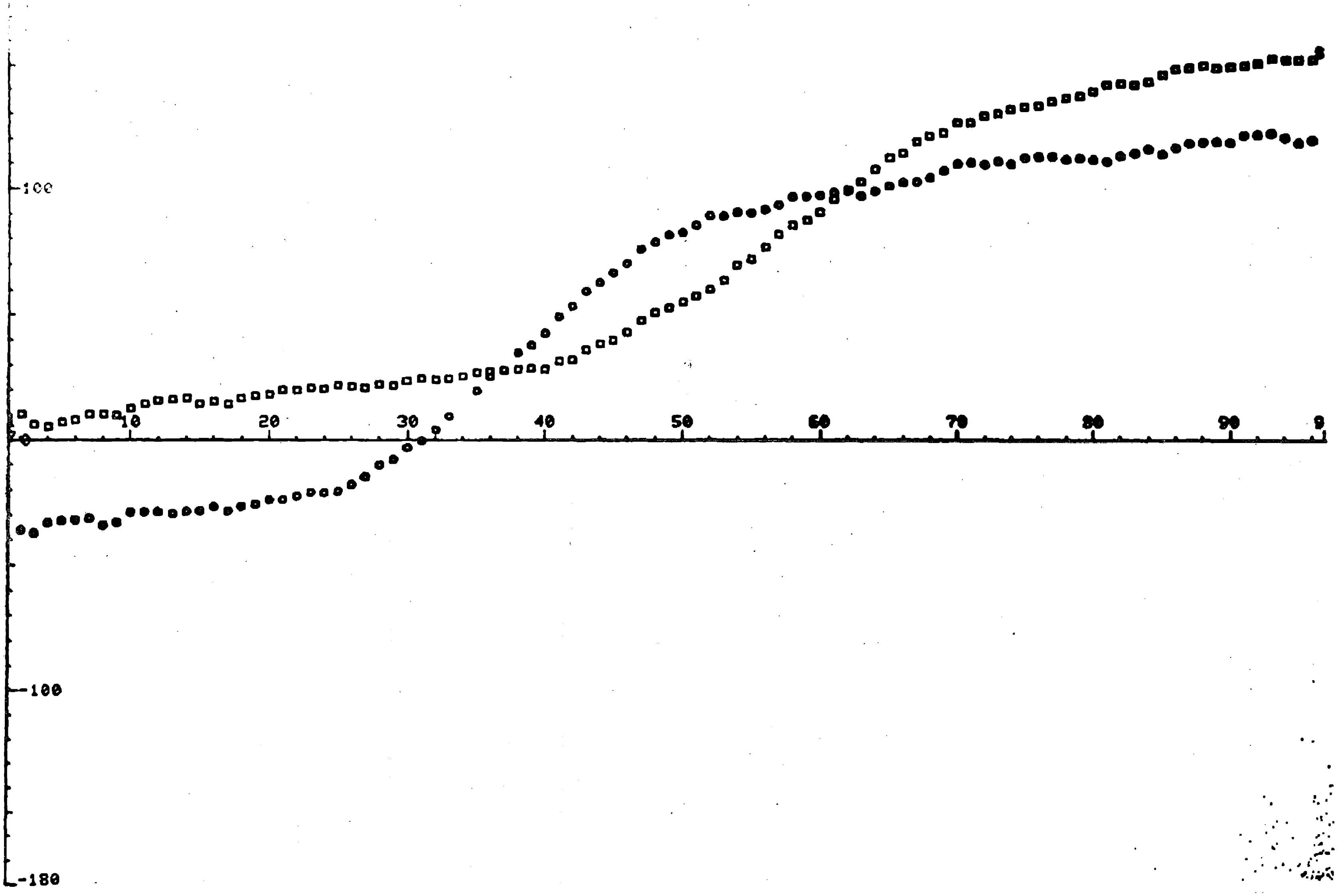


Figure 6-3



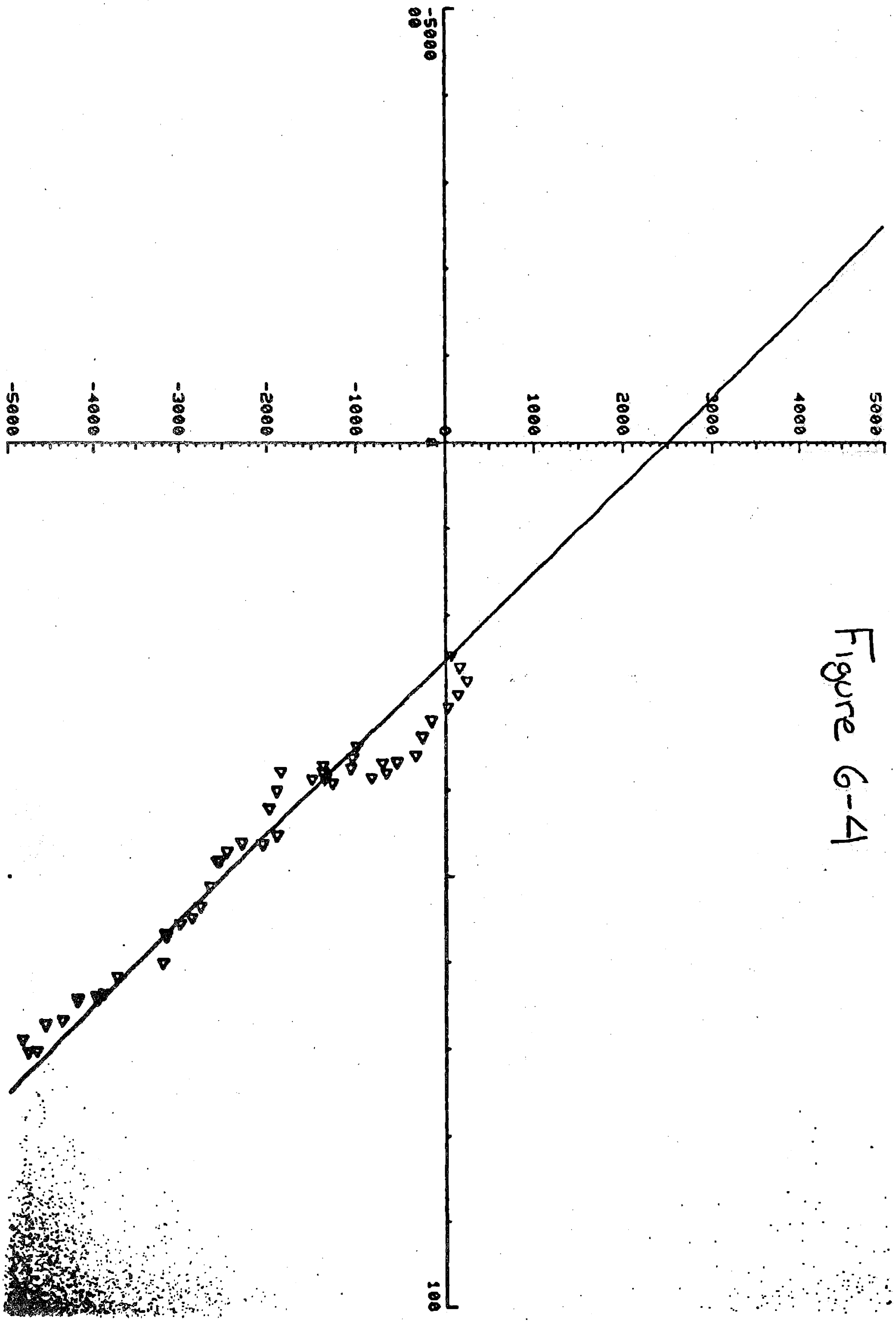

-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000
-5000
00
100

Figure 6-4

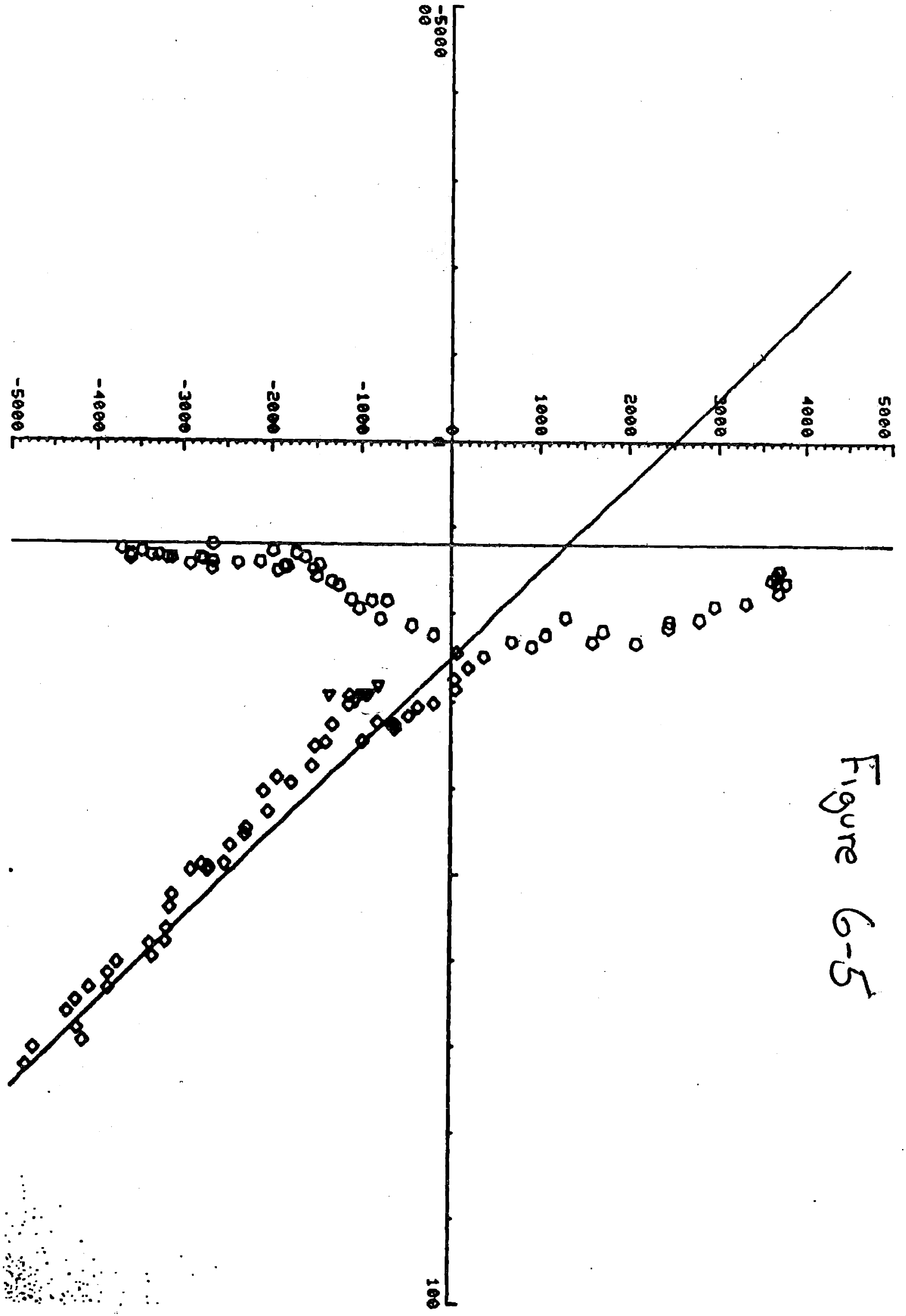

-5000
00
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000
100

Figure 6-5

REFERENCES

[1] Y.Bar-Shalom, and E.Tse,"Tracking in a Cluttered Environment with Probabilistic Data Association", in Proc. 4th Symposium Nonlinear Estimation, Univ. California, San Diego, Sept. 1973.

[2] Y. Bar-Shalom, "Extension of the Probabilistic Data Association Filter to Multitarget Environments", in Proc. 5th Symposium Nonlinear Estimation, Univ. California, San Diego, Sept. 1974.

[3] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment," IEEE Trans. Automatic Control, vol AC-23, no. 4, August 1978.

[4] P. Green, "An Alternate Method for Position Location," Unpublished Memo, Group 22, MIT Lincoln Laboratory, Lexington, Mass.

[5] R.S. Hebbert, and T.S. Ballard, "A Tracking Algorithm Using Bearing Only," Naval Surface Weapons Center, TR 75-150, White Oak, Silver Spring, MD 25 Oct 1975 (ADA024454).

[6] K. Kaverian, "Multiobject Tracking by Adaptive Hypothesis Testing," S.B. Thesis, Massachusetts Institute of Technology, May, 1979.

[7] C.L. Morefield, "Application of 0-1 Integer Programming to the Multitarget Tracking Problem," in Proc. IEEE Conf. Decision Control, December, 1979.

[8] D. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Automatic Control, vol AC-24, no. 12, December 1979.

[9] R.A. Singer, R.G. Sea, and K. Housewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments," IEE Trans. Information Theory, vol IT-20, pp. 423-432.

[10] R.W. Sittler, "An Optimal Data Association Problem in Surveillance Theory," IEEE Trans. Military Electronics, vol MIL-8, pp.125-139, April 1964.

[11] H. L. Van Trees, Detection, Estimation, and Modulation Theory, part I. Wiley: New York, 1968.