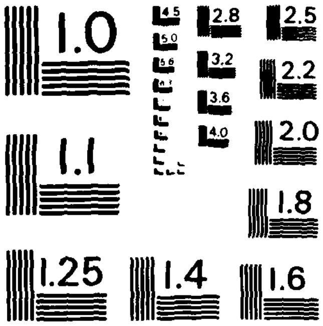
AD-A160 836 THE BAYESIAN APPROACH TO RECURSIVE STATE ESTIMATION: 1/2
IMPLEMENTATION AND APPLICATION(U) AIR FORCE INST OF
TECH WRIGHT-PATTERSON AFB OH S C KRAMER 1985
UNCLASSIFIED AFIT/CI/NR-85-139D F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(1)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 85-139D 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Bayesian Approach to Recursive State Estimation: Implementation and Application	5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Stuart Charles Kramer	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: University of California San Diego	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433 - 6583	12. REPORT DATE 1985 13. NUMBER OF PAGES 118
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASS 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver 25 Oct 85
LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT, Wright-Patterson AFB OH

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract of the Dissertation

The Bayesian Approach to Recursive State Estimation:

Implementation and Application

by

Stuart Charles Kramer

Doctor of Philosophy in Engineering Sciences (Systems Science)

University of California, San Diego, 1985

Professor Harold W. Sorenson, Chairperson

In Bayesian estimation, the objective is to calculate the complete density function for an unknown quantity conditioned on noisy observations of that quantity. This work considers recursive estimation of a nonlinear discrete-time system state using successive observations. The formal recursion for the density function is easily written, but generally there is *no closed form solution.* The numerical solution proposed here is obtained by modifying the recursion and using a simple piece-wise constant approximation to the density functions. The critical part of the algorithm then becomes a discrete linear convolution that can be realized using FFT's. Hence this approach requires only $O(N\ln(N))$ (where N is the number of grid points in the approximation) operations for the convolution instead of the $O(N^2)$ of previous solutions. The approach also allows detailed analysis of error propagation through the algorithm. This allows characterization of the situations leading to potentially large errors and derivation of the bound on the maximum error growth. The algorithm is shown to be quite stable by comparing its long-term performance to a Kalman filter for a linear system with Gaussian noises. Two potentially broad uses for the technique are then explored. First, the technique can be used to gain insight into stochastic system behavior by visualizing the propagation of the density through time. Second, it provides a benchmark for traditional point estimators, since the true optimal estimate for any given loss function can be calculated from the conditional density. To illustrate these uses, we consider two nonlinear applications: simultaneous state/parameter estimation and bearings-only tracking. In both cases the density behavior is analyzed, and then displayed graphically. The mean of the

[illegible] estimation

85-139D

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (AU). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: The Bayesian Approach to Recursive State Estimation: Implementation and Application

AUTHOR: Stuart Charles Kramer

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS __________ ( ) b. $__________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):

UNIVERSITY OF CALIFORNIA

SAN DIEGO

The Bayesian Approach to Recursive State Estimation: Implementation and Application

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Engineering Sciences (Systems Science)

by

Stuart Charles Kramer

Committee in charge:
Professor Harold W. Sorenson, Chairperson
Professor Anthony Sebald
Professor D. V. Bhaskar Rao
Professor Laurence Milstein
Professor Robert Engle

1985

The dissertation of Stuart Charles Kramer is approved,
and it is acceptable in quality
and form for pub'ication on microfilm.

Anthony V Sebald

B. V. Bhaskar Rao

Robert C.

L B Milstein

Harold W. Sorenson
Chairperson

University of California, San Diego

1985

RE: Classified Reference, Distribution
Unlimited
No change per Ms. June DeSouza, AFIT/NR

✓

A-1

Table of Contents

List of Figures and Tables

List of Figures and Tables (cont.)

Acknowledgements

This dissertation was started while I was a student under the Air Force Institute of Technology's Civilian Institution Program, and completed while I was teaching at the School of Engineering of the Air Force Institute of Technology. Therefore, I would like to gratefully acknowledge the assistance of the United States Air Force in allowing me the opportunity to pursue this degree, and in supporting me while I attended school. I wish to thank my fellow faculty at the Air Force Institute of Technology, particularly my department head, Dr. Peter Torvik, for their patience and assistance while I finished this dissertation.

I am also deeply indebted to the faculty of the Department of Applied Mechanics and Engineering Sciences at the University of California, San Diego, for their help and encouragement. I particularly want to thank my principle advisor Dr. Harold Sorenson for sticking with me while I concluded my dissertation via long-distance.

Finally, my deepest thanks go to my wife Ede and my daughters Olivia and Hilary, for their love, support, and assistance throughout this difficult endevour. I could not have done it alone.

Vita

May 20, 1954 — Born — Fort Belvoir, Virginia

1976	BSEE, Colorado State University
1978	MS in Systems Engineering, Air Force Institute of Technology
1978-1981	Systems Analyst, U.S. Air Force
1981-1984	Student, University of California, San Diego
1984-1985	Instructor, Air Force Institute of Technology, Dayton, Ohio
1985	Doctor of Philosophy, University of California, San Diego

Publications

S. Kramer, L. Boehmer, M. Gatti, "RV Fratricide Study - A Program and User's Guide," Air Force Weapons Laboratory, Kirtland AFB, NM, AFWL-TN-79-520, 1979.

R. Almassy, R. Fowler, D. Gleason, J. Jackson, S. Kramer, W. Krutzler, G. Machovina, D. Ray, T. Stumpf, P. Warika, *Preliminary Design and Analysis of an Aircushion Vehicle as an Alternate Aircraft Takeoff System, Vol. 1 (Secret), Vol. 2 (Secret), Vol. 3, (Unclass), Vol. 4 (Unclass)*, DTIC Doc. AD-C016363L, AD-C016364L, AD-B032273L, AD-B032274L, March 1978.

S. Kramer, "A Character-Generating Interface," *Slide Rule,* Vol 19, 16-17, Jan-Feb 1976.

1. INTRODUCTION

1.1. The Bayesian Approach to Recursive State Estimation

The Bayesian approach to state estimation is based in the Bayesian philosophy of statistics. which is a subject unto itself. For the sake of brevity, we will only briefly discuss some facets of the Bayesian approach which have direct bearing on this work. For more detailed discussions of and justifications for the approach, the reader must refer to the considerable literature on the subject. References [1]-[3] and [35]-[37] are a sampling of applicable works. The interested reader will undoubtedly find many additional useful references with minimal effort.

For our purposes it is enough to note that there are two distinguishing features of the Bayesian approach. The first, and less controversial, is the recognition that the ultimate use of estimation (or more correctly, statistical inference) is to provide a rational basis for decision making under uncertainty, and that this function is the first stage of the decision process. The second stage is picking a policy to maximize an expected value or minimize an expected loss. Bayesian statistics provides the framework for the calculation, manipulation, and interpretation of probability densities as a basis for decision making. Decision theory provides the utility of loss function, calculates expectations, and maximizes or minimizes in the process of making a decision. As an example, consider the optimal stochastic control problem. The decision is what control policy to adopt in order to minimize the expected value of some cost functional. Calculating the density function to be used in taking the expectation is independent of the choice of cost functional. The Bayesian viewpoint merely makes this separation explicit.

The second feature of this approach amounts to the replacement of the word 'random' with the more general term 'uncertain' in our probabilistic discussions. It does not matter whether the uncertainty arises from a truly random process or from a deterministic process that is sufficiently complicated or poorly understood that we are uncertain of its state. All types of uncertainty equate to randomness. Broadening our definition of random in this way is not as far-fetched as it may seem; even classical statisticians do it at times. As an example, consider the over-used coin flip. It is not really a random event in the classical sense. That is, if I could in fact exactly duplicate the conditions of a toss, all the applied forces and the timing of the catch and all the other factors, then I would get the same result each time. Taken the other way, if I had sufficient knowledge of the physical factors affecting the coin, I could predict the outcome with complete certainty. We are only willing to accept it as random because this is such a complicated system (person and coin together) that we generally have insufficient knowledge to predict its outcome. As another example closer to an engineer's heart, look at computer generated pseudo-random numbers. For most practical purposes we accept these as random, when in fact they are generated by a (very complicated) purely deterministic procedure. A more extreme example is the problem of parameter identification. For a given system, the unknown parameter has a definite value. If we could measure it, it would become a known constant. It is clearly not a random variable in the classical sense, yet the standard engineering practice is to treat it as such. If anything, this broader interpretation of random is probably closer to most engineers intuitive understanding than the classical definition.

Complementing the broadening of the definition of random is a reinterpretation of the probability density associated with an unknown quantity. In the Bayesian view, the density represents the observer's judgement of the relative likelihoods of the possible outcomes. *Put another way, the density function is a measure of the observer's knowledge about the unknown quantity.* This interpretation raises several points. First, the density function is a feature of the system/observer pair, and is not an intrinsic feature of the system. Furthermore, it may be legitimately different for different observers of the same system, since different observers may have differing amounts of information about the system. Second, the observer must be able (willing?) to quantify his knowledge in the form of a density function. While we will take the position that this is a given, we point out that it is not a trivial problem. Third, this interpretation need not be at odds with the classical relative frequency interpretation. Faced with a system for which we have historical information, it is reasonable to assume (without information to the contrary) that the same relative frequencies of outcomes will continue. Lastly, despite the broadening of our interpretation, all the familiar rules for manipulating densities still apply.

Now look at these two ideas in the context of an estimation problem. Since we want eventually to use our information in a decision process, and since the density describes our information, we must need to calculate the entire density function for the unknown quantity. This will describe the probability (likelihood) of a particular outcome conditioned on our knowledge. If we are fortunate, we may have subsequent observations of the system. The additional information can be incorporated into our density using standard probability theory, resulting in a new density conditioned on our increased knowledge. Again, the entire density is required to describe the information we have about the unknown quantity.

The main point is that the Bayesian objective is to calculate the entire probability density function of the unknown quantity, based on the available information. This work focuses on the discrete-time state estimation problem, where we want information about the state of a system given imperfect observations of the system. Moreover, we want to do so recursively, so that we update our information step by step as we get new observations. We will see shortly that this problem naturally yields a recursive solution.

1.2. The General Solution

In the present case, the unknown quantity is the state of a discrete-time system which propagates through time according to

$$x_{k+1} = f_k(x_k, u_k, w_k)$$

where $x_k \in R^n$ is the system state at time $k \geqslant 0$, $u_k \in R^m$ is the known control applied to the system, $w_k \in R^l$ is an unknown disturbance, and $f_k : R^n \times R^m \times R^l \to R^n$ is the system transition function. The only assumption on the times represented by k is that they are sequential; in particular, it is not necessary that they are equally spaced in time. Note that the system is memoryless, in the sense that the the state at time $k+1$ depends only on the state and inputs at time k, and not on the past history of the state or inputs. The information available to us is contained in a set of measurements, augmented by the known inputs. The measurements $z_k \in R^o$ are given by

$$z_k = h_k(x_k, v_k)$$

where $v_k \in R^p$ is an unknown disturbance, and $h_k : R^n \times R^p \to R^o$ is the measurement function. For convenience define the information vector

$$Z_k = \{z_i, u_i \; ; \; i = 0, k\}$$

The vector Z_k represents all the information we have gathered about the system during its operation.

Suppose that we wish to solve the prediction problem. In other words, we want to know at each time k the state of the system at the next time, $k+1$. In view of the unknown disturbances in the system and the observations, it is clear that we cannot know this precisely. We can however make an assessment of the likelihood of the system state taking on any particular value based on the information available to us. In the Bayesian view, this means we want to calculate the conditional prediction density $p(x_{k+1}|Z_k)$. Applying elementary rules for manipulating densities, we can write

$$p(x_{k+1}|Z_k) = \int \frac{p(x_k|Z_{k-1},z_k)\, p(x_{k+1}|x_k,Z_k) p(u_k|x_k,z_k,Z_{k-1})}{p(u_k|z_k,Z_{k-1})}\, dx_k$$

If we define the time update density t_k as

$$t_k(x_{k+1},x_k) \equiv \frac{p(x_{k+1}|x_k,Z_k) p(u_k|x_k,z_k,Z_{k-1})}{p(u_k|z_k,Z_{k-1})}$$

then the above becomes

$$p(x_{k+1}|Z_k) = \int p(x_k|Z_{k-1},z_k)\, t_k(x_{k+1},x_k)\, dx_k \tag{1.1}$$

The function t_k represents the likelihood of the system state transitioning from a given state at time k to another at $k+1$, and is calculable from the model description and the information vector. The density $p(x_k|Z_{k-1},z_k)$ is the conditional density for the state at time k given observations through time k, and is known as the filtering density. We see from (1.1) that the conditional prediction density is the nonlinear convolution of the current filtering density and the time update density.

Let us look now at the filtering density. Applying Bayes' theorem, we get

$$\begin{aligned} p(x_k|Z_{k-1},z_k) &= \frac{p(x_k|Z_{k-1})\, p(z_k|x_k,Z_{k-1})}{p(z_k|Z_{k-1})} \\ &= \frac{p(x_k|Z_{k-1})\, p(z_k|x_k,Z_{k-1})}{\int p(x_k|Z_{k-1})\, p(z_k|x_k,Z_{k-1})\, dx_k} \end{aligned}$$

Define the measurement update density m_k as

$$m_k(x_k,z_k) \quad p(z_k|x_k,Z_{k-1})$$

so that the above becomes

$$p(x_k|Z_{k-1},z_k) = \frac{p(x_k|Z_{k-1})\, m_k(x_k,z_k)}{\int p(x_k|Z_{k-1})\, m_k(x_k,z_k)\, dx_k} \tag{1.2}$$

The measurement update density describes the likelihood of getting the particular observation z_k from different possible state values, and is calculable from the model description and the data. Equation (1.2) shows that the filtering density is obtained by multiplying the last prediction density by the measurement density and then normalizing.

We can continue to apply (1.1) and (1.2) alternately until we reach $k=0$. At that point, we must assume an initial density $p(x_0) = p(x_0|Z_{-1})$, the prediction density for the state at time zero before any observations are made. The three elements that we need to compute $p(x_{k+1}|Z_k)$ then are $p(x_0)$, and m_i and t_i for $i=1,k$. These densities are known as the priors for the problem and summarize our a priori knowledge about the behavior and structure of the system. Although the time and measurement update densities may in general depend parametrically on the particular measurement realization, they are structurally determined by the system model.

Viewed from the opposite direction in time (from $t=0$ forward), equations (1.1) and (1.2) form a natural recursion. Given the priors, we alternately calculate the conditional prediction density and conditional filtering density as we accumulate observations. These two equations represent the complete formal solution to the recursive Bayesian state estimation problem. Although the solution is easy to obtain in this general form, its practical solution is not so simple. As in many cases in engineering, approximate methods of solution must be devised.

1.3. Structure of this Dissertation

This dissertation is basically divided into two parts. First, we develop a new approach to the implementation of equations (1.1) and (1.2). Then, we use that approach as a tool in exploring some estimation problems.

Beginning the first part, chapter 2 discusses previous work in developing approximate ways of implementing the recursion, and relates this work to those efforts. Basically what has been done is to approximate the densities in various ways, so that the convolution in (1.1) and the integration in (1.2) can be done numerically. The approach here is to use a simple approximation (piecewise constant) that results in a computationally efficient algorithm. The algorithm itself is presented in Chapter 3. First the basic equations are derived for a somewhat more restrictive model than the one given in the previous section. Next, implementation of these equations is discussed, since the computational advantage of this method obviously depends on the implementation details. Finally, since this is an approximate method and there is some error associated with it, an analysis of the error and its propagation is presented.

Chapter 4 covers some simple examples of the application of the algorithm to demonstrate its performance. A scalar linear system with Gaussian noise is considered first. This allows the approximation to be compared against the exact solution, since the exact solution for this case is given by the Kalman filter. This simple example is studied quite thoroughly to demonstrate the performance of this technique. Finally, a simple, nonlinear, non-Gaussian case is considered to further demonstrate the method.

Using this algorithm as a tool, we then look at some broader issues associated with the Bayesian approach. Even with the computational improvement of this method, the Bayes solution is fairly expensive. The logical question is what do we gain by using this approach. The answer to that, of course, is highly problem dependent. We can describe, however, several general ways that the algorithm can be of use to us. First, simply working with the equations and thinking of the process in terms of operations on densities provides a deeper understanding of the mechanics of the process. For more detail, we can use the close approximation of the posterior density that the algorithm generates to plot the prediction density or develop other useful descriptions of it. This can provide additional insight into the behavior of the system, and is useful in much the same way as traditional simulation is for deterministic systems. Furthermore, as is done with deterministic simulation, we can use the technique to examine how the behavior of the system varies as we change features of the system. Finally, again since we have a representation of the posterior density, we can actually calculate the optimal estimator for any given loss function. Thus we can compare the performance of traditional point estimators directly to that of the optimal. In this way we can judge, for instance, whether apparently poor performance is a fault in the estimator, or is actually just the best that one can expect. This could be particularly useful in assessing transient performance, where there are virtually no theoretical results. In Chapter 5 we consider these uses in a concrete situation, in the context of the problem of simultaneous state and parameter estimation for a linear system. In Chapter 6 we continue by looking in some detail at a passive tracking problem, where the measurements consist only of bearings to a target. Lastly, Chapter 7 contains a summary of the major conclusions of this work.

2. REVIEW OF PREVIOUS WORK AND RELATION TO THIS EFFORT

Equations (1.1) and (1.2) represent the most general solution to the recursive Bayesian state estimation problem. The difficulty is that there is generally not a closed form solution to this recursion. The notable exception is when the system is described by the linear equations

$$x_{k+1} = F_k x_k + G_k w_k$$

$$z_k = H_k x_k + v_k$$

where w_k and v_k are white Gaussian sequences uncorrelated with the state. If the initial density is Gaussian, the conditional densities remain Gaussian, and the mean and variance of these densities propagate according to the well-known Kalman filter equations. Since a Gaussian is completely described by its mean and variance, the Kalman filter is in fact the Bayes solution for this model. This fact apparently was first noted by Ho and Lee [33].

Ideally, we would like the densities to maintain the same functional form, as in the linear Gaussian case, so that the Bayesian recursion could be accomplished by algebraically updating a finite number of parameters. Densities with this property are known as reproducing densities. Spragins [34] has shown for the special case of recursively estimating a fixed parameter that a reproducing density exists if and only if there exists a sufficient statistic for the data which is expressible as a vector of fixed length. This is not true in the general case, however, because of the presence of the time update due to the system dynamics and input noise. Reproducing densities have been shown to exist for a few particular combinations of system dynamic functions and noise densities [4,5], but they are not generally useful. It is difficult to find reproducing densities for this recursion because of the two very different operations involved: multiplication for the measurement update, and non-linear convolution for the time update. Closure with respect to both operations is rare.

When the model is not linear Gaussian, we must look for approximate means of evaluating the recursion (1.1)-(1.2). The earliest, and certainly the most used approximation, is the extended Kalman filter or EKF. In the EKF, the system model is linearized about the mean of the last prediction density and the Kalman filter equations applied directly to the linearized system. In the Bayesian context, this amounts to approximating the measurement and time update densities by Gaussians. This in turn implies that the conditional densities are approximated by Gaussians. Clearly, this is a crude approximation, highly dependent on the quality of the linearization and the actual form of the noise densities. In practice, however, this approximation has been shown to be reasonably good in a wide variety of cases. (For a number of good examples of applications of the EKF, see [32].) Its principal advantage is that it is by far the computationally simplest of all the approximation techniques. On the other hand, the EKF is subject to divergence, where the actual error in the estimate exceeds the error covariance approximation provided by the filter. The EKF also has poor transient response in many cases, even though the steady state performance may be acceptable. The difficulties arise since the relatively smooth approximating Gaussian may miss important features of the true densities. Considerable work has been done on improving the EKF by various means, but we will not discuss those modifications here. Since virtually all of these techniques are derived as point estimators, and not as true approximations to the conditional densities, they are not Bayes estimators.

Series expansions such as Gram-Charlier and Edgeworth expansions were the next attempt to improve the fidelity of the density approximations [6-11]. In this technique, the densities are approximated by an infinite sum of polynomials which are orthogonal with respect to the Gaussian density and can be used to represent a large class of other densities. The convolution of (1.1) can then be represented as the sum of convolutions of the simpler polynomials. In theory, this can provide arbitrarily good approximation to the actual densities. In practice, however, it was found that a large number of terms are needed to get reasonable accuracy for distinctly non-Gaussian densities. In addition, when the series are truncated, as they must be, the resulting approximation can be negative over parts of the state space. Hence the approximation may not be a density itself. This can cause significant disturbances in subsequent approximate densities, as well as in any expectations taken in the decision process.

Both these previous techniques are essentially local techniques, since they are designed to be most accurate in a restricted central region. The remaining techniques we will discuss are global, in that they attempt to provide a uniform degree of approximation over the entire densities. (This distinction is due to Sorenson [31].) There are basically two global approaches: direct approximation of the densities, and approximation of the integrals in (1.1) and (1.2). Both approaches are based on sampling the densities at points distributed through the region containing non-negligible probability. In function approximation, the points and the approximate density values at those points are chosen on the basis of matching some interpolating function to the density. The integration (or functional) approximation approach picks the points based on a particular numerical integration scheme. Of course, the two approaches are usually dual in the sense that a particular interpolation method results in the integration being equivalent to some numerical integration approximation, and similarly, a numerical integration method implies a particular interpolation scheme. Nevertheless, this distinction is useful for classifying the approximate Bayes methods.

The simplest of the global function approximation methods is the point-mass method introduced by Bucy [12] and elaborated by Bucy and Senne [13]. In this method, the densities are approximated by point masses located on a regular grid. To keep the grid compatible with the evolving density while reducing storage requirements, Bucy and Senne proposed a floating rotating grid. The grid at each time is centered on the mean of the prediction density and based on the eigenvectors of the covariance matrix. Bucy and Senne also reduced storage by retaining only those points in an ellipsoid based on the covariance matrix. The size of the ellipsoid is chosen so that only a fixed fraction of the total probability mass is discarded. The final refinement was the use a separate grid for each mode if the densities were multimodal. Given an a priori density approximated in this way, the filtering density resulting from (1.2) is also a collection of point masses. The convolution of (1.1) can then be written as a summation. This summation is then evaluated at the new grid points to provide the approximation to the new prediction density.

An alternative approach was introduced at about the same time by Alspach and Sorenson [14-17]. In this method, the densities are approximated by a weighted sum of Gaussians, each centered on a different point in state space. The recursion can then be written as a collection of EKF's operating in parallel. Since each Gaussian in the sum has comparatively small variance, it is more likely that the linearization in the EKF will remain a valid approximation. The choice of grid point locations, variances, and weights is best done by minimizing the L_2 norm of the error, although approximate methods can be used to reduce computation.

Both these methods, particularly the point-mass method, have considerable computational and storage requirements, although they can provide reasonably accurate density approximations. These requirements were considered excessive in light of the computational resources available at the time they were developed. Both speed and storage were quite limited compared to that available today. Of the two limitations, storage was actually the more restrictive. This tended to make it more profitable to trade off storage for computational complexity. For a given desired accuracy, it was clear that you could get by with fewer grid points if you used a more sophisticated interpolation scheme. A theoretical framework for this work was provided by Center [18] who considered the problem in terms of generalized least squares. His approach allowed for the development of an essentially unlimited number of different approximations, of which the point-mass and Gaussian sum approaches are examples. Most of the remaining work in this direction focused on different spline approximations [19-22].

The most recent of the function approximations is the p-vector approach of Sorenson [23]. This method is derived from the Gaussian sum method by taking the limit as the variances of the individual Gaussians go to zero. It is also related to the point-mass method, but provides a computational advantage over both. This method also reduces the convolution (1.1) to a summation.

The alternative approach of approximating the integral followed much the same path. All such methods amount to replacing the integral with a weighted sum of samples of the integrand. The particular quadrature formula used determines the locations of the samples in state space and the weights used in the summation. The basic issue again is trading off computational complexity against the number of sample points required to achieve a desired accuracy. Klein [24-27] in particular has investigated a number of different quadrature formulas.

All these global techniques share two related elements:

1. The method must provide a procedure for defining the initial and subsequent grids. For computational efficiency, the grid should be redefined at each time to track the conditional densities. The elements of the grid which must be specified are the boundaries of the grid in state space, and the number and distribution of points within the boundary. The distribution of the points will usually depend on the approximation technique that will be used. In most of these techniques, setting up the grid is a non-trivial problem.

2. Given the grid, the means of performing the Bayesian update must be specified. This will depend on the particular approximation (either function or integral) that is chosen. In almost all cases (except Gaussian sum) this results in replacing the convolution (1.2) with a discrete nonlinear convolution. It is this operation which tends to be the most expensive computationally since it requires on the order of N^2 operations,'where N is the number of grid points.

As discussed above, the direction of previous research has been determined primarily by the limitations in computing power that existed in the early seventies. Recent advances in computer hardware have loosened those limitations. It is now reasonable to reduce computational complexity at the expense of storage requirements. As memory costs continue to decline and computer speeds increase, this trade-off becomes even more favorable. This change in the computing environment motivated the work presented here. The basic approach is to use a simple approximation to the conditional densities in order to obtain a fairly simple representation of the recursion (1.1)-(1.2).

The method that will be presented in this dissertation is based on approximating the conditional densities by functions piecewise constant on regions defined by a regular grid. This has several advantages:

1. The grid can be described in a particularly simple way; the region of interest is filled with a number of identical multidimensional polyhedrons. This results in an easy grid update.
2. The density approximation has a simple form that makes such operations as taking expectations easy. This is important since we expect the density to be used in some decision making process which usually involves taking the expected value of some cost functional.
3. The convolution (1.1) can be written as a discrete linear convolution instead of a discrete nonlinear convolution. This allows the convolution to be calculated using FFTs at a cost proportional to $N \ln(N)$ instead of N^2.
4. The error in approximating the conditional densities and its propagation through the recursion can be analyzed in a straightforward way. Although the error bounds turn out to not be extremely tight, the analysis does point out the situations in which lead to poor performance.

Although the idea of using piecewise constant approximations has been mentioned briefly before [28], it was not pursued. The idea of manipulating the recursion to achieve a linear convolution appears to be new. The use of FFTs for evaluating convolutions is, of course, not new, but application to this problem is. Finally, the propagation of the error through the recursion has not been addressed in any depth before.

3. THE ALGORITHM

3.1. Assumptions, Notation, and Conventions

Before we derive the proposed algorithm, we must specialize the very general model introduced in Chapter 1, and introduce some notation that will make the following presentation clearer. We will restrict our attention to systems modeled by

$$x_{k+1} = f_k(x_k, u_k) + w_k$$

$$z_k = h_k(x_k, v_k) \tag{3.1}$$

The noise sequences $\{w_k\}$ and $\{v_k\}$ are assumed to be white and zero-mean, to be uncorrelated with the state and with each other, and to have densities with finite support. The control sequence $\{u_k\}$ is either a priori known or calculable at each time from only the available observations, which implies that $p(u_k \mid z_k, Z_{k-1}, x_k) = p(u_k \mid z_k, Z_{k-1})$. The system dynamics function f_k and the observation function h_k are at least measurable functions.

While this is a more restrictive model than that of Chapter 1, it is still quite general. In fact, many systems which at first might not appear to fit these assumptions can be modeled by (3.1). For instance, colored noises can be included by appending a subsystem driven by white noise, whose output is the desired colored noise. In other cases, this model can provide an adequate approximation. As an example, consider Gaussian noise. It has a density with infinite support. An adequate approximation might be to truncate the support at, say, plus and minus three standard deviations. (Is anything really Gaussian anyway? Given the experimental observation that most 'real' noises are thinner in the tails than the Gaussian, the above might actually be a more appropriate model in many cases than assuming Gaussian noise.)

Restricting ourselves to the model above allows us to simplify the expressions for the densities of interest. Applying elementary rules for manipulating densities, as we did in Chapter 1, we can write the equations for the conditional prediction and filtering densities for the model (3.1) as

$$p(x_k \mid Z_k) = C^{-1}\, p(x_k \mid Z_{k-1})\, p(z_k \mid x_k)$$

$$C = \int p(x_k \mid Z_{k-1})\, p(z_k \mid x_k)\, dx_k \tag{3.2}$$

$$p(x_{k+1}|Z_k) = \int p(x_k|Z_k)\, p(x_{k+1}|x_k,u_k)\, dx_k$$

Following standard practice, multidimensional integrals as above are understood to mean

$$\int dx\ g(x) = \int dx_{(1)} \int dx_{(2)} \cdots \int dx_{(n)}\, g(x_{(1)},\ldots,x_{(n)})$$

where $x_{(i)}$ is the ith component of the vector x. Unless otherwise noted, integrals are assumed to be over the entire support of the integrand.

From the model we note that

$$p(x_{k+1}|\,x_k,u_k) = p_{w,k}(x_{k+1}-f_k(x_k,u_k))$$

where $p_{w,k}$ is the density function for w_k. Now, for notational convenience, we define

$$\pi(x) = \pi_k(x_k) = p(x_k|\,Z_{k-1})$$

$$\mu(x,z) = \mu_k(x_k,z_k) = p(z_k|\,x_k)$$

$$\phi(x) = \phi_k(x_k) = p(x_k|\,Z_k) \tag{3.3}$$

$$r(x) = r_k(x) = p_{w,k}(x)$$

As implied in the above definitions, we will drop the explicit dependence on the time k whenever there is no ambiguity. To help minimize confusion, the variable k will be used only to denote the time. With these definitions, the recursion (3.2) becomes

$$\phi(x) = C^{-1}\,\pi_k(x)\,\mu(x,z) \tag{3.4}$$

$$\pi_{k+1}(x) = \int \phi(y)\, r(x - f(y,u))\, dy \tag{3.5}$$

where the normalizing constant C is chosen so that $\int\phi(x)dx = 1$.

In actually computing the recursion, we will use approximations to the above densities. A hat will be used to signify an approximate density, and a tilde to signify the associated error function. Hence, we would write $\tilde{\pi}(x) = \pi(x) - \hat{\pi}(x)$. We will be approximating the densities on a regular multidimensional grid in state space defined by the points $x_j = Aj + b$, where j is an n-vector with integer elements, each in the range 1 to M. There are, therefore, $N = M^n$ total grid points. The $n \times n$ matrix A and the n-vector b are chosen so that the grid covers some region of interest. Each point x_j of the grid is the center of a cell in R^n defined by

$$I_j = \{z = y + x_j\ ;\ y \in I_0\}$$

where I_0 is the generic cell given by

$$I_0 = \{z = A\gamma \; ; \; \gamma_{(1)} \in (-\frac{1}{2}, \frac{1}{2}), \cdots, \gamma_{(n)} \in (-\frac{1}{2}, \frac{1}{2})\}$$

The volume of each cell, which we will call α, is given by $|\det(A)|$. Note that the only restriction on A is that $\det(A) \neq 0$. Thus we are not restricted to rectangular grids, and so have increased flexibility in tailoring the grid to the densities. Figure 3.1 shows two examples for n=2 and M=3. Observe also that $\bigcup_j I_j$ covers the entire region of interest (by definition of the grid), and that $I_j \cap I_i = \emptyset$ for $i \neq j$.

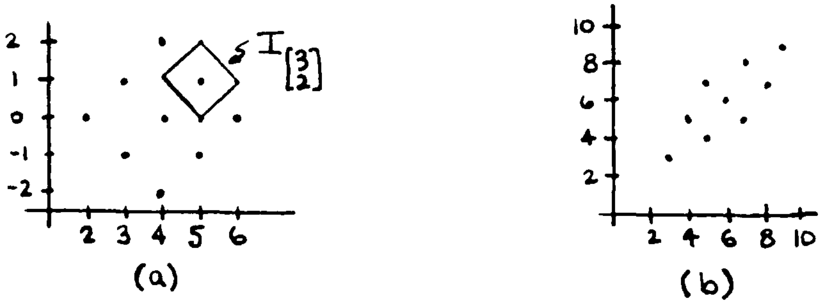


Figure 3.1 – Example grids using (a) $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and (b) $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Finally, we note that we will be using multidimensional summations in the work ahead. They are exactly analogous to integrals, so

$$\sum_j y_j = \sum_{j_{(1)}=1}^{M} \cdots \sum_{j_{(n)}=1}^{M} y_{j_{(1)}, \cdots, j_{(n)}}$$

Like integrals, the sum is assumed to be over the entire grid when no limits are specified.

3.2. Derivation of the Algorithm

As discussed in the introduction, the difficulty with the recursion we have derived for calculating the conditional prediction density is that there is generally no closed form solution. This being the case, we are forced to look for some means of approximating the recursion. The approach here is to approximate the prediction density π_k by a piecewise constant function, and then apply (3.4) and (3.5) to recursively update the approximate prediction density. In deriving the algorithm, however, we will start with π_{k+1} and work backward.

Begin by considering a slightly modified version of equation (3.5). If we define $y=f(x,u)$, then the state update equation becomes

$$x_{k+1} = y + w_k$$

and then (3.5) becomes

$$\pi(x) = \int_{S(\psi)} \psi(y)\, r(x-y)\, dy \tag{3.6}$$

where $\psi(y)$ is the conditional density function $p(y|\,Z_k)$ and $S(\psi)$ is the support of ψ. Notice that (3.6) is a linear convolution. Now assume that we have defined a grid that covers the support of π. (We will discuss how to get such a grid a bit later.) The best L_1 or L_2 constant approximation to a function over a region is the average of the function over that region. Hence, the best piecewise constant approximation to π on the given grid is $\hat{\pi}(x)=\pi_i$, $x \in I_i$, where

$$\pi_i = \frac{1}{\alpha}\int_{I_i} \pi(x)\, dx = \frac{1}{\alpha}\int_{I_i} dx \int_{S(\psi)} dy\, \psi(y)\, r(x-y) \tag{3.7}$$

with the last equality due to (3.6). Since w_k is a zero-mean sequence, we know that the origin is contained in $S(r)$, the support of r. This implies that $S(\psi) \subset S(\pi)$, so the assumed grid also covers the support of ψ. We can therefore use the covering property of the grid to break up the inner integral in (3.7) to get

$$\pi_i = \frac{1}{\alpha}\int_{I_i} dx \sum_j \int_{I_j} dy\, \psi(y)\, r(x-y) = \sum_j \frac{1}{\alpha}\int_{I_i} dx \int_{I_j} dy\, \psi(y)\, r(x-y)$$

The interchange of summation and integration is justified since it is a finite sum. If we now now replace $\psi(y)$ with a piecewise approximation $\hat{\psi}(y)=\psi_j$, $y \in I_j$, we get

$$\pi_i \approx \sum_j \psi_j \frac{1}{\alpha}\int_{I_i} dx \int_{I_j} dy\, r(x-y) \tag{3.8}$$

Consider the double integral above; using the definition of the grid, and with a change of variable, we get

$$\frac{1}{\alpha}\int_{I_i} dx \int_{I_j} dy\, r(x-y) = \alpha \int_{I_0} d\delta_1 \int_{I_0} d\delta_2\, r(A(i-j)+\delta_1-\delta_2) \tag{3.9}$$

Note that the result is a function only of the difference $(i-j)$; this is crucial to obtaining a discrete linear convolution. If we call the result of (3.9) $r_{i,j}$, then we get from (3.8)

$$\pi_i \approx \sum_j \psi_j r_{i-j} \tag{3.10}$$

We note this is a discrete linear convolution. Equation (3.9) is also the expression for α times the best piecewise approximation to r over the region $I_i \times I_j$, so in a sense we have replaced both ψ and r with their piecewise constant approximations in getting to equation (3.10).

To get to equation (3.8) we introduced the piecewise constant approximation to ψ. As with π we define $\hat{\psi}(y) = \psi_i$, $y \in I_i$, where

$$\psi_i = \frac{1}{\alpha} \int_{I_i} \psi(y)\, dy$$

Recall that $\psi(y)$ is the conditional density of $y = f(x,u)$. Using this, and suppressing the conditioning in the notation, we can rewrite the integral as

$$\int_{I_i} \psi(y)\, dy = \mathrm{Prob}(y \in I_i) = \mathrm{Prob}(x \in H_i\,; H_i = \{x : f(x,u) \in I_i\}) = \int_{H_i} \phi(x)\, dx \tag{3.11}$$

As long as f is measurable, H_i is a measurable set (since it is the inverse image of an open set), and so the integral is well defined. Now suppose that we have the piecewise constant approximation $\hat{\pi}_k(x)$ from the last time step, defined by $\hat{\pi}_k(x) = \pi_{k,i}$, $x \in I_{k,i}$, with $I_{k,i}$ based on the grid from the last iteration. Using this, and the definition of ϕ from (3.4), in (3.11), we get

$$\int_{H_i} \phi(x)\, dx = \int_{H_i} C^{-1} \pi_k(x) \mu(x,z)\, dx \approx C^{-1} \sum_j \int_{H_i \cap I_{k,j}} \hat{\pi}_k(x) \mu(x,z)\, dx$$

$$= C^{-1} \sum_j \pi_{k,j} \int_{H_i \cap I_{k,j}} \mu(x,z)\, dx$$

In the above we used the fact that the old grid covers the support of $\hat{\pi}_k$, just as the current grid covers the support of the current $\hat{\pi}$. This now allows us to write ψ_i in terms of the previous prediction density and the measurement update density as

$$\psi_i \approx C^{-1} \sum_j \pi_{k,j} \frac{1}{\alpha} \int_{H_i \cap I_{k,j}} \mu(x,z)\, dx \tag{3.12}$$

The integral term above is the constant approximation of μ over the region $H_i \cap I_{k,j}$. Hence, as with (3.10), we have effectively replaced both densities with their approximations.

The constant C in (3.12) is to be chosen so that the densities are properly normalized, that is, so that they integrate to one. Since we are primarily interested in the prediction density approximation $\hat{\pi}$, it is convenient to move the constant from (3.12) to (3.10) and leave $\hat{\psi}$ unnormalized. Applying the normalization requirement to $\hat{\pi}$ and using the modified form of (3.10) gives

$$\int \hat{\pi}(x)\, dx = \alpha \sum_i \pi_i = \alpha\, C^{-1} \sum_i \sum_j \psi_j\, r_{i-j} = 1$$

Therefore, we must have

$$C = \alpha \sum_i \sum_j \psi_j\, r_{i-j} = \alpha \sum_j \psi_j \sum_i r_{i-j} = \alpha \sum_j \psi_j \tag{3.13}$$

We now have the basic relations for recursively calculating the approximation to the conditional prediction density. Combining (3.9), (3.13), and the modified versions of (3.10) and (3.12), we get

$$\mathrm{H}_j = \{x : f_k(x,u) \in \mathrm{I}_j\} \tag{3.14a}$$

$$\psi_j = \frac{1}{\alpha} \sum_i \pi_{k,i} \int_{\mathrm{H}_j \cap \mathrm{I}_{k,i}} \mu_k(x,z)\, dx \tag{3.14b}$$

$$r_{i-j} = \alpha \int_{\mathrm{I}_0} d\delta_1 \int_{\mathrm{I}_0} d\delta_2\, r_k(A\,(i-j)+\delta_1-\delta_2) \tag{3.14c}$$

$$C = \alpha \sum_j \psi_j \tag{3.14d}$$

$$\pi_{k+1,i} = C^{-1} \sum_j \psi_j\, r_{i-j} \tag{3.14e}$$

It should be pointed out that the filtering density ϕ is not calculated explicitly in the recursion above. If for some reason ϕ is needed, we can readily calculate the piecewise constant approximation $\hat{\phi}(x)=\phi_i$, $x \in \mathrm{I}_{k,i}$, by

$$\phi_i = \frac{1}{\alpha_k} \int_{\mathrm{I}_{k,i}} C^{-1}\, \hat{\pi}(x)\, \mu(x,z)\, dx$$

$$= C^{-1}\, \pi_{k,i}\, \frac{1}{\alpha_k} \int_{\mathrm{I}_{k,i}} \mu(x,z)\, dx \equiv C^{-1}\, \pi_{k,i}\, \mu_i$$

C is chosen so that $\hat{\phi}$ will be a normalized density, so

$$C = \alpha_k \sum_i \pi_{k,i}\, \mu_i$$

If we have calculated $\hat{\varphi}$, we can modify (3.14b) to save some computation (but with some loss in accuracy) by substituting $\hat{\phi}$ for φ in (3.11). Since $\hat{\phi}$ is piecewise constant on the old grid, we get

$$\psi_j = \frac{1}{\alpha} \sum_i \phi_i \int_{H_j \cap I_{k,i}} dx$$

If we choose to do this, $\hat{\psi}$ will be normalized, so we can drop (3.14d) and remove C^{-1} from (3.14e).

3.3. Interpretation as a Probability Mass Filter

It is interesting to note that the recursion described by (3.14) can also be derived from a rather different point of view. Based on this alternate approach, the recursion can be interpreted as updating the probability mass associated with each element of the grid, as opposed to updating the probability density.

We begin as in section 3.2 by letting $y = f(x,u)$, and assuming that we have a grid that covers the region of interest in state space. The conditional probability that x_{k+1} is in I_i is given by

$$\text{Prob}(x_{k+1} \in I_i) = \sum_j \text{Prob}(y \in I_j)\ \text{Prob}(x_{k+1} \in I_i \mid y \in I_j) \tag{3.15}$$

The conditioning on the measurements has been omitted from the notation for clarity. Recalling equations (3.11) and (3.12),

$$\text{Prob}(y \in I_j) = \int_{H_j} C^{-1}\, \pi_k(x)\, \mu(x,z)\, dx \tag{3.16}$$

where H_j is the inverse image of I_j as before. Suppose that we have the set of probabilities $\text{Prob}(x_k \in I_{k,i})$ from the last time step. If we make the approximation that the mass is uniformly distributed within each region, then

$$\pi_k(x) = \frac{1}{\alpha_k} \text{Prob}(x_k \in I_{k,i})$$

Making this substitution in (3.16) gives

$$\text{Prob}(y \in I_j) = C^{-1} \sum_i \frac{1}{\alpha_k}\ \text{Prob}(x_k \in I_{k,i}) \int_{H_j \cap I_{k,i}} \mu(x,z)\, dx \tag{3.17}$$

The second factor in the sum in (3.15) can be written as

$$\mathrm{Prob}(x_{k+1} \in \mathrm{I}_i \mid y \in \mathrm{I}_j) = \frac{\int_{\mathrm{I}_i} dx \int_{\mathrm{I}_j} dy \; p(x_{k+1} | y) \; p(y)}{\int_{\mathrm{I}_j} p(y) \; dy}$$

This time we make the approximation that the mass for y is uniformly distributed over each region, so $p(y)$ is constant over I_j. This, combined with the definition of r, gives

$$\mathrm{Prob}(x_{k+1} \in \mathrm{I}_i \mid y \in \mathrm{I}_j) = \frac{1}{\alpha} \int_{\mathrm{I}_i} dx \int_{\mathrm{I}_j} dy \; r(x-y) \tag{3.18}$$

Equations (3.15), (3.17), and (3.18) give an approximate recursion for calculating the probability masses on each interval.

The equivalence between this recursion and (3.14) is obvious given the following observations:

$$\mathrm{Prob}(x_k \in \mathrm{I}_{k,i}) = \alpha_k \pi_{k,i}$$

$$\mathrm{Prob}(y \in \mathrm{I}_j) = \alpha \; C^{-1} \psi_j$$

$$\mathrm{Prob}(x \in \mathrm{I}_i \mid y \in \mathrm{I}_j) = r_{i-j}$$

Making these substitutions in (3.15), (3.17), and (3.18) produces

$$\psi_j = \frac{1}{\alpha} \sum_i \pi_{k,i} \int_{\mathrm{H}_j \cap \mathrm{I}_{k,i}} \mu(x,z) \; dx$$

$$\pi_{k+1,i} = C^{-1} \sum_j \psi_j \; r_{i-j} \tag{3.20}$$

The constant C is obtained by applying the normalization constraint, which gives $C = \sum_i \psi_i$.

The recursion derived in this section (equations (3.20)) is identical to that of (3.14). We conclude that the recursion can be interpreted either as an update of the approximate probability density functions, or as an approximate update of the probability masses for each region in the grid.

3.4. Implementation

Several aspects of the implementation of the recursion defined by (3.14) need to be discussed before we consider any specific examples.

As the basic equations stand, there is little to recommend them computationally over any other approach, particularly since they are based on a fairly crude approximation, and therefore may require many points to get a given accuracy. The main culprit is (3.14e) which is a discrete convolution of the two sequences $\{\psi_i\}$ and $\{\tau_i\}$. Direct computation of this convolution requires on the order of N^2 operations, where N is the total number of points in the grid. It is not necessary, however, to compute it directly. Since it is a linear convolution, we can use transform techniques. Specifically, we can take the product of the discrete Fourier transforms (DFT) of the sequences, and take the inverse transform of the result. Using standard FFT techniques, a DFT requires on the order of $N\,ln(N)$ operations. The FFT approach, then, requires on the order of $(3N\,ln(N) + N)$ operations (two forward DFTs, one multiplication, and one inverse DFT). It does not take a very large N to get substantial savings. We can also use the considerable work that has already been done in developing highly efficient FFT algorithms ([29] and [30] are recent examples) to further increase the advantage. Since FFT algorithms tend to be highly parallel, it is also possible to take advantage of various computer hardware such as parallel and array processors.

There is potential danger in the FFT approach, though, since the FFT convolution is circular, and hence may result in aliasing. To avoid this, the grid must be large enough to completely cover the support of π. We assumed this at the beginning of section 3.2, so this requirement does not affect the algorithm directly. But since both $\{\psi_i\}$ and $\{\tau_i\}$ are defined on this same grid, the above implies that portions of both sequences will necessarily be zero. This in turn indicates there is a certain amount of waste in both storing and performing computations on the zero elements. It may be advisable to use dynamic storage techniques to only store the non-zero elements, and customize the FFT algorithm to take advantage of the fact that only the middle portion of these sequences is non-zero.

The FFT approach also provides the normalizing constant C given by (3.14d). In many cases, this normalization is not really necessary, since we lose no information if we choose not to normalize. Dropping C would save computation at a slight cost of ease in interpreting the results. Some care must be taken to insure that the unnormalized density does not exceed the numerical limits of the computer, though. It is quite possible for it to go to zero everywhere as far as the computer is concerned. In any case, we get C automatically, and can then normalize at the cost of only N multiplies.

We get C as follows. Let $\{\psi^{*}\}$ be the transform sequence of $\{\psi\}$. From the definition of the DFT,

$$\psi_i^{*} = \sum_{l} \psi_l \exp(\frac{-2\pi\, j(i-1)(l-1)}{N})$$

so $\psi_1^{*} = \sum_{l} \psi_l$. Using this in (3.14d) gives $C = \alpha \psi_1^{*}$. Since we have calculated the transform of ψ, C is readily available. The normalizing constant can be included when the transform sequences are multiplied together, or π can be normalized after the inverse transform, or it can be left unnormalized and C carried along to be used if needed.

Next, note that (3.14b) and (3.14c) require evaluating integrals. Two factors make this less troublesome than it seems. First, the integrands are a priori given functions, not involving the approximated prediction density. Equation (3.14c) is particularly straight-forward since it is over a single well-defined region. Second, as alluded to in section 3.3, the integrals can generally be evaluated in terms of the noise distribution functions. Thus we are trading evaluations of the density function for evaluations of the distribution function. For the many cases where the distribution function is known in closed form, this is no additional work. In the cases where the noise characteristics have been found experimentally, the distribution function is generally more accurately estimated than the density. In some cases then, this method may actually be easier and more accurate. If the distribution functions are not available in closed form and numerical integration is used, however, these equations may represent a large computational burden.

Equation (3.14b) has another nasty feature in addition to the integral. As written, it is a convolution which would require on the order of N^2 operations in addition to the integrals. In general, this could be true. In practice, the system dynamics will probably be well-behaved enough that most of the sets $H_j \cap I_i$ will be empty for any given j. Since we only have to sum over the non-empty intersections, this can be a significant reduction. The regularly shaped grid regions I_i make it fairly easy to both determine which regions will intersect, and describe the intersection, even though the inverse image sets H_j may be strangely shaped.

Specification of the inverse image set defined by (3.14a) can be the most algebraically difficult part of implementing this algorithm. H_i is usually described by giving the equations of its sides, each side being the inverse image of a side of the original grid region. Depending on the form of f, these can be awkward equations. If f is one to one, H_i will be a single region. If f is many to one though, it will likely be several disconnected regions, and so the integral in (3.14b) will be the sum of integrals over those regions. Fortunately, the problems with specifying H_i typically occur while coding the algorithm for a particular system, and are not a major computational burden when running it. We will see some examples of calculating H_i and $H_i \cap I_i$ in the applications sections ahead.

The recursion of (3.14) does not specify how to select the new grid at each time step, other than to require that it covers the support of the new prediction density. The support can be calculated before π itself is as follows. Let $S(\pi_k)$ be the support π_k, and $S(\mu)$ be the support of μ. Then the support of ϕ is $S(\phi) = S(\pi_k) \cap S(\mu)$, since ϕ is the product of π_k and μ. The support of ψ, $S(\psi)$, is the smallest region containing the image of $S(\phi)$ under the system dynamics function f. Finally, the support of π_{k+1} is

$$S(\pi_{k+1}) = \{x = y + z \ ; \ y \in S(\psi), \ z \in S(\tau)\}$$

The grid can then be chosen in any convenient way, as long as it completely covers $S(\pi_{k+1})$. It is worth pointing out that the grids at each time may be completely independent, even to containing different numbers of points.

Note that the preceding depends on the assumption made earlier that the noise densities have finite support. There are a number of commonly used noise densities which have infinite support. In those cases it is necessary to truncate the densities at some point to achieve finite support. Done judiciously, this truncation will have negligible effect.

To initialize the recursion, we need to specify an initial grid and prediction density approximation. The initial grid is chosen to cover the support of the given initial prediction density. The density approximation is then $\hat{\pi}_0(x) = \pi_{0,i}$, $x \in I_{0,i}$, $\pi_{0,i} = \frac{1}{\alpha} \int_{I_{0,i}} \pi_0(x) \, dx$, where $\pi_0(x)$ is the initial prediction density.

The last point to be discussed is the selection of the number of grid points along each axis of the grid system. This number must be large enough to provide sufficient accuracy of the approximations, but should be as small as possible to minimize the computational burden. There may also be some restrictions on the number of grid points due to the FFT algorithm chosen, although these are usually minor. Unfortunately, there are no firm guidelines to help here. Probably the best approach is to start with a small number and increase it until you get reasonable accuracy for the particular application. As we will see in the numerical examples in chapter 4, we can get surprisingly good results with relatively few points.

3.5. Analysis of Error Propagation

For classical point estimation techniques the major concern is whether the point estimate converges to the true value, and if so, how fast. Here, since we are actually calculating an approximation to the posterior density, the question is instead whether the density approximation diverges from the true density. In other words, we are concerned with bounding the maximum growth of the error instead of finding the minimum shrinkage.

We turn now to characterizing the error that accumulates in the prediction density approximation as we execute the recursion (3.14). As we will show, there is an upper bound on the growth of the error from step to step. This bound is the best possible, in the sense that there are pathological situations in which the bound is achieved. It is not, however, what one would call reassuringly tight, since it allows fairly rapid error growth. Fortunately, in deriving the bound we are also able to characterize the situations which lead to large error growth, so that we can say that generally the error growth will be much less than its bound.

For this section we will make two simplifying assumptions to help decrease the notational clutter. All the conclusions also apply in the general case. First, we restrict our attention to the scalar case. Among other things, this allows us to use the representation $x=\alpha(i+\delta)+\beta$, $\delta\in(-,)$, for $x\in I_i$, where the scalars α and β define the current grid. Second, we assume the system dynamics function f is invertible. With this assumption, we can write

$$\psi(y) = \pi_k(f^{-1}(y))\,\mu(f^{-1}(y),z)\,g(y)$$

where $g(y)=|df^{-1}(y)/dy|$. Note that ψ is not normalized, so that it corresponds to the earlier derivation.

We begin by assuming that we have $\hat{\pi}_k(x)$, the prediction density approximation at time k. The error is $\tilde{\pi}=\pi-\hat{\pi}$, and in practice is unknown since we would not know π. Note, however, that

$$\int \tilde{\pi}(x)\,dx = \int \pi(x)\,dx - \int \hat{\pi}(x)\,dx = 1 - 1 = 0$$

Hence, we do know that $\tilde{\pi}$ oscillates about zero, since its average over its support is zero. Let $\tilde{\pi}$ be bounded by $|\tilde{\pi}_k(x)| \leq \epsilon_{\pi_k} \pi_k(x)$.

Now consider ψ. It is approximated by $\hat{\psi}(y) = \psi_j$, $y \in I_j$, ψ_j defined by (3.14b). For $y \in I_j$ we get

$$\tilde{\psi}(y) = \psi(y) - \psi_j = (\hat{\pi} + \tilde{\pi})(f^{-1}(y))\mu(f^{-1}(y),z)g(y) - \psi_j$$

$$= \tilde{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y)$$

$$+ \hat{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y) - \frac{1}{\alpha}\int_{I_j} \hat{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y)dy$$

Define

$$\tilde{\psi}_c(y) = \tilde{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y) \tag{3.21}$$

$$\tilde{\psi}_a(y) = \hat{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y) - \frac{1}{\alpha}\int_{I_j} \hat{\pi}(f^{-1}(y))\mu(f^{-1}(y),z)g(y)dy \tag{3.22}$$

The term $\tilde{\psi}_c$ is the 'carryover' error. It is the error in $\hat{\psi}$ from using the density approximation $\hat{\pi}$ instead of the actual density π. From the assumed bound on $\tilde{\pi}$, we get

$$|\tilde{\psi}_c(y)| = |\tilde{\pi}\ \mu\ g| \leq \epsilon_{\pi_k} \pi_k\ \mu\ g = \epsilon_{\pi_k} \psi(y) \tag{3.23}$$

The other term, $\tilde{\psi}_a$, is the 'approximation' error. It is the error which would result from approximating ψ by $\hat{\psi}$ if $\hat{\pi}$ were the true density. This error is (conceptually at least) calculable, since $\hat{\pi}$ and μ are known. Suppose, then, that

$$|\tilde{\psi}_a(y)| \leq \epsilon_{\psi_a} \psi(y) \tag{3.24}$$

Since we are approximating with a piecewise constant function, we expect that $\tilde{\psi}$ will be a worse approximation (ϵ_{ψ_a} will be greater) as ψ becomes steeper.

It is worth emphasizing that $\tilde{\psi}_c$ and $\tilde{\psi}_a$ arise from distinct sources. If $\pi_k = \hat{\pi}_k$, so that we had no error in the last prediction density, $\tilde{\psi}_a$ would be zero, but $\tilde{\psi}_c$ would be unaffected. Likewise, $\tilde{\psi}_c$ is unaffected by the quality of the approximation of ψ as reflected by $\tilde{\psi}_a$. Two other features of these error terms should be noted: first, $\int_{I_j} \tilde{\psi}_a(y)dy = 0$, and second, even though $\int \tilde{\pi}(x)dx = 0$, it is very likely that $\int \tilde{\psi}_c(y)dy \neq 0$. Hence, $\tilde{\psi}_a$ locally averages to zero, while $\tilde{\psi}_c$ is not guaranteed to average to zero even over its entire support.

Next, substitute $\psi = \hat{\psi} + \tilde{\psi}_a + \tilde{\psi}_c$ in the convolution for π_{k+1}. We get

$$\pi_{k+1}(x) = C^{-1} \int (\hat{\psi}(y) + \tilde{\psi}_a(y) + \tilde{\psi}_c(y))\, \tau(x-y)\, dy$$

$$= C^{-1}\left\{ \sum_j \psi_j \int_{I_j} \tau(x-y)dy + \int \tilde{\psi}_a(y)\tau(x-y)dy + \int \tilde{\psi}_c(y)\tau(x-y)dy \right\} \tag{3.25}$$

Note the integral in the first term above. Its piecewise constant approximation for $x \in I_i$ is τ_{i-j} given by (3.14c). Therefore define

$$\tilde{\tau}_{i-j}(\delta) = \int_{I_j} \tau((\alpha(i+\delta)+\beta)-y)\, dy \;-\; \tau_{i-j}$$

As with $\tilde{\psi}_a$, this error is calculable since τ is a known function. Accordingly, assume

$$|\, \tilde{\tau}_{i-j}(\delta)| \leqslant \epsilon_\tau \int_{I_j} \tau((\alpha(i+\delta)+\beta)-y)\, dy$$

Notice that $\int \tilde{\tau}_i(\delta) d\delta = 0$, so $\tilde{\tau}$ also locally averages to zero. Substituting the above into (3.25) yields

$$\pi_{k+1}(x) = C^{-1}\left\{ \sum_j \psi_j \tau_{i-j} + \sum_j \psi_j \tilde{\tau}_{i-j}(\delta) + \int \tilde{\psi}_a(y)\tau(x-y)dy \right.$$

$$\left. + \int \tilde{\psi}_c(y)\tau(x-y)dy \right\} \tag{3.26}$$

The normalizing constant C is obtained by solving $\int \pi(x)dx = 1$, which results in

$$C = \int dx \left\{ \sum_j \psi_j \tau_{i-j} + \sum_j \psi_j \tilde{\tau}_{i-j}(\delta) + \int \tilde{\psi}_a(y)\tau(x-y)dy + \int \tilde{\psi}_c(y)\tau(x-y)dy \right\}$$

$$= \alpha \sum_j \psi_j + \int \tilde{\pi}_k(x)\, \mu(x,z)\, dx \tag{3.27}$$

where we have used the fact that $\tilde{\tau}$ and $\tilde{\psi}_a$ integrate to zero. Refer back to equations (3.14d) and (3.14e). If we let $\hat{C}$ be defined by (3.14d) and substitute this and equation (3.14e) into (3.26) and (3.27), we get

$$C = \hat{C} + \int \tilde{\pi}_k(x)\, \mu(x,z)\, dx$$

$$\pi_{k+1}(x) = \frac{\hat{C}}{C}\pi_i + \frac{1}{C}\left\{ \sum_j \psi_j \tilde{\tau}_{i-j}(\delta) + \int \tilde{\psi}_a(y)\tau(x-y)dy + \int \tilde{\psi}_c(y)\tau(x-y)dy \right\}$$

From this it is obvious that the error functions are

$$\tilde{C} = \int \tilde{\pi}_k(x)\ \mu(x,z)\ dx \tag{3.28}$$

$$\tilde{\pi}_{k+1} = -\frac{\tilde{C}}{\hat{C}}\pi_{k+1}(x) + \frac{1}{\hat{C}}\left\{\sum_j \psi_j \tilde{r}_{i-j}(\delta) + \int \tilde{\psi}_a(y) r(x-y)dy \right.$$

$$\left. + \int \tilde{\psi}_\epsilon(y) r(x-y)dy \right\} \tag{3.29}$$

We will examine each term in turn.

First consider $\tilde{C}$ from (3.28). Using the assumed bound on $\tilde{\pi}_k$ we get

$$|\tilde{C}| \leqslant \int |\tilde{\pi}_k(x)|\,\mu(x,z)dx \leqslant \epsilon_{\pi_k}\int \pi_k(x)\mu(x,z)dx = \epsilon_{\pi_k}\int \psi(y)dy = \epsilon_{\pi_k}C$$

The bound on this error is only attained in the unlikely event that μ is concentrated on a set where $\tilde{\pi}_k$ attains one of its extremes. At the other extreme, if μ is constant over the support of $\tilde{\pi}_k$, $\tilde{C}=0$, since $\int \tilde{\pi}_k dx = 0$. Generally, we would expect $\tilde{C}$ to be small when μ is spread out, while a more concentrated μ would increase the likelihood of a larger error.

The first term of (3.29) is the normalization error in $\hat{\pi}$ due to the error in the normalizing constant $\hat{C}$. If $\tilde{C}$ is small, this term will likewise be small. In any case, this is a fairly benign error, since it is a simple scaling error and affects all the intervals equally.

Next consider the second term of (3.29). Using the assumed bounds on $\tilde{r}$, $\tilde{\psi}_a$, and $\tilde{\psi}_\epsilon$, this term can be bounded by

$$|\sum_j \psi_j \tilde{r}_{i-j}(\delta)| \leqslant \epsilon_r \sum_j \psi_j \int_{I_j} r((\alpha(i+\delta)+\beta)-y)\ dy$$

$$= \epsilon_r \int (\psi(y) - \tilde{\psi}_a(y) - \tilde{\psi}_\epsilon(y))\ r((\alpha(i+\delta)+\beta)-y)\ dy$$

$$\leqslant \epsilon_r(1+\epsilon_{\psi_a}+\epsilon_{\pi_k})\int \psi(y)\ r((\alpha(i+\delta)+\beta)-y)\ dy = \epsilon_r(1+\epsilon_{\psi_a}+\epsilon_{\pi_k})\pi_{k+1}(x)$$

The first inequality is achieved only if $|\tilde{r}|$ reaches a maximum at the same δ for all i; otherwise, the error is strictly less than its bound. The second inequality depends on the behavior of the convolution of $\tilde{\psi}_a$ and $\tilde{\psi}_\epsilon$ with r, and will be discussed when we get to the third and fourth terms of the error equation. Note that this term is analogous to $\tilde{\psi}_a$. It is the error which would result from approximating π with $\hat{\pi}$ if $\hat{\psi}$ were the true density ($\psi = \hat{\psi}$). Hence we would expect it to depend on how steep π is. We can investigate this further by making the rough approximation that $\int_{I_j} r(x-y)dy = r_{i-j}$ at $x = \alpha i + \beta$, and is linear between. This implies

$$\tilde{\tau}_{i-j}(\delta) = \delta(\tau_{i+1-j} - \tau_{i-j}) \; ; \; \delta \in (0, \frac{1}{2})$$

Using this approximation gives

$$\sum_j \psi_j \tilde{\tau}_{i-j}(\delta) \approx \sum_j \psi_j \delta(\tau_{i+1-j} - \tau_{i-j}) = \hat{C}\delta(\pi_{i+1} - \pi_i)$$

So, roughly speaking, this term depends directly on the rate of change of $\hat{\pi}$ (and hence π), as we would intuitively expect.

Now look at the third term. This is the propagation of the approximation error in $\hat{\psi}$ through the time update convolution. With the bound on $\tilde{\psi}_a$ from (3.24), we have

$$| \int \tilde{\psi}_a(y)\tau(x-y)dy | \leqslant \epsilon_{\psi_a} \int \psi(y)\tau(x-y)dy = \epsilon_{\psi_a} C\pi_{k+1}(x)$$

Recall that $\int_{I_j} \tilde{\psi}_a(y)dy = 0$. Hence if τ is reasonably constant over intervals of length α, then this term will be near zero. Another way to look at this is to recognize that the convolution is an averaging operation. Since $\tilde{\psi}_a$ locally averages to zero, we would expect the convolution to be near zero as long as τ is locally nearly constant.

Lastly, consider the fourth term of (3.29). This term results from the propagation through the current cycle of the recursion of the error in the prediction density from the previous time step. Using the assumed bound on the previous prediction density error $\tilde{\pi}_k$, and the same steps as in the above paragraph, we obtain

$$| \int \tilde{\psi}_c(y)\tau(x-y)dy | \leqslant \int | \tilde{\pi}_k(f^{-1}(y)) | \mu(f^{-1}(y),z)g(y)\tau(x-y)dy$$

$$\leqslant \epsilon_{\pi_k} \int \pi_k(f^{-1}(y))\mu(f^{-1}(y),z)g(y)\tau(x-y)dy = \epsilon_{\pi_k} C\pi_{k+1}(x)$$

The behavior of this term is much the same as the last, except that we require the product $\mu g \tau$ to be reasonably constant over the whole support of π_k, instead of just locally. This is because we can only assert that the entire integral of $\tilde{\pi}_k$ is zero, rather than the stronger statement that the integral over intervals of the grid is zero. Clearly, this is more difficult requirement to meet. Even so, if μ and τ are not excessively concentrated, we can expect this term to be significantly less than its bound. This behavior is important, since it implies that we can expect past errors to be forgotten to some extent. This in turn indicates there is a possibility of reaching a steady state error level, where the error introduced at each iteration is roughly balanced by the attenuation of past errors.

The total error bound is obtained by combining all the bounds for the individual terms in (3.29). This produces

$$\tilde{\pi}_{k+1}(x) \leq \frac{1}{1-\epsilon_{\pi_k}} \{\epsilon_{\pi_k} + \epsilon_r(1 + \epsilon_{\psi_a} + \epsilon_{\pi_k}) + \epsilon_{\psi_a} + \epsilon_{\pi_k}) \, \pi_{k+1}(x)$$

Thus, the growth of the error for each iteration is bounded in this sense. Unfortunately, this is not a particularly restrictive bound. For example, a relatively small 5% error in each of the constituent terms allows nearly 22% error for the total.

The above covers the behavior of the error in the density approximation. In light of the interpretation of the recursion as a mass filter, it is also useful to consider the behavior of the error in the mass assigned to each interval. Let $P_i = \text{Prob}(x_{k+1} \in \mathrm{I}_i \mid Z_k)$. Then

$$P_i = \int_{\mathrm{I}_i} \pi_{k+1}(x) \, dx$$

The approximate probability $\hat{P}_i$ generated by the algorithm is given by

$$\hat{P}_i = \int_{\mathrm{I}_i} \hat{\pi}_{k+1}(x) \, dx = \alpha\pi_{k+1,i}$$

Hence the error is

$$\tilde{P}_i = P_i \quad \hat{P}_i = \int_{\mathrm{I}_i} (\pi_{k+1}(x) - \hat{\pi}_{k+1}(x)) \, dx = \int_{\mathrm{I}_i} \tilde{\pi}_{k+1}(x) \, dx$$

Using (3.29) to substitute for $\tilde{\pi}_{k+1}$ for $x \in \mathrm{I}_i$, we get

$$\tilde{P}_i = \int_{\mathrm{I}_i} \left\{ -\frac{\tilde{C}}{\hat{C}} \pi_{k+1}(x) + \frac{1}{\hat{C}} \left(\sum_j \psi_j \tilde{r}_{i-j}(\delta) + \int \tilde{\psi}_a(y) r(x-y) dy \right. \right.$$

$$\left. \left. + \int \tilde{\psi}_c(y) r(x-y) dy \right) \right\} dx \qquad (3.30)$$

The integral of course distributes over the sum. The first term is simply a constant times P_i. The second term is zero since the integral over any interval of $\tilde{r}_{i-j}$ is zero. The third term can be bounded by

$$\left| \int_{\mathrm{I}_i} dx \int dy \, \tilde{\psi}_a(y) r(x-y) \right| \leq \epsilon_{\psi_a} \int_{\mathrm{I}_i} dx \int dy \, \psi(y) r(x-y)$$

$$= \epsilon_{\psi_a} \int_{I_i} C \pi_{k+1}(x) dx = \epsilon_{\psi_a} C P_i$$

Similarly, the fourth term is bounded by

$$| \int_{I_i} dx \int dy \; \tilde{\psi}_a(y) r(x-y) | \leqslant \epsilon_{\pi_k} C P_i$$

Hence the total error in the approximation to the probability mass is bounded by

$$| \tilde{P}_i | \leqslant \frac{2\epsilon_{\pi_k} + \epsilon_{\psi_a}}{1 - \epsilon_{\pi_k}} P_i$$

Note that this bound depends on the error in the density approximation from the previous step. Hence, although the error in the mass approximation is less than the error in the density approximation at each step, it can grow as rapidly.

In addition to the errors in the density and mass approximations, we may also be concerned with the error in the ultimate decision from using the approximate density instead of the true density. In general, the decision will depend on the expected value V of some cost functional v. Following the approach used above, we see immediately that the error in using the approximation to V is given by

$$\tilde{V} = \int \tilde{\pi}_{k+1}(x) \; v(x) \; dx$$

If $v \geqslant 0$, the error can be bounded by

$$| \tilde{V} | \leqslant \int | \tilde{\pi}_{k+1}(x) | \; v(x) \; dx \leqslant \epsilon_\pi \int \pi_{k+1}(x) \; v(x) \; dx = \epsilon_\pi \; V$$

where ϵ_π is the bound for the error in the current density approximation. Recall though that $\int \tilde{\pi}_{k+1}(x) dx = 0$, so if the cost functional is not overly concentrated, we can expect the error to be significantly less than the bound. Clearly if v happens to be concentrated in the wrong places, then the error could approach the bound, but this would be unlikely. Once again we arrive at the intuitively reasonable characterization that the errors will be aggravated by sharply spiked functions. If v takes on both positive and negative values, we cannot bound the error in this way.

As we have seen, the error bound for the recursion can grow at an uncomfortable rate. Fortunately, the conditions when the errors reach their bounds are fairly unusual. In general terms, as long as the densities remain reasonably evenly spread out, the errors should be substantially below the bounds. This matches with our intuitive expectation, since the piecewise constant approximations that are being used are best when the densities are reasonably flat in the scale of the grid. This in turn implies that we can reduce the error by increasing the number of grid points. In fact, the piecewise constant approximations converge uniformly for the class of continuous densities as $N \rightarrow \infty$, so the recursion would be exact for densities in that class. We can still use this technique for discontinuous densities, but we cannot guarantee convergence as N increases.

In the next chapter we will look at a number of numerical examples to show the typical error behavior of this algorithm.

4. NUMERICAL EXAMPLES OF THE ALGORITHM

4.1. The Linear Gaussian Case

In this chapter we consider some examples of the algorithm in use. The first is the linear Gaussian case. This case is important as a benchmark since, as we pointed out earlier, the Kalman filter is its Bayes' solution. Hence we can compare the approximate solution provided by the algorithm with the exact analytical solution. As we will see, the algorithm performs quite well.

Implementing the recursion of (3.14) for a scalar system with linear dynamics is very straightforward. To begin, we will use a time invariant system. The model is

$$x_{k+1} = Fx_k + w_k$$

$$z_k = Hx_k + v_k$$

where w_k and v_k are zero-mean Gaussian with covariances Q and R respectively. Hence

$$\mu(x,z) = (2\pi R)^{-} \exp(\frac{-(Hx-z)^2}{2R})$$

$$r(x) = (2\pi Q)^{-} \exp(\frac{-x^2}{2Q}) \quad .$$

Since the algorithm demands that μ and r have finite support, we truncate the Gaussian densities at plus and minus three standard deviations. The approximation grid is simply a collection of intervals on the real line, defined by the two scalars α_k and β_k. With this information, we can now implement the equations in (3.14). The inverse image sets H_i (equation (3.14a)) are single intervals, given by $((\alpha_{k+1}(i-)+\beta_{k+1})/F,(\alpha_{k+1}(i+)+\beta_{k+1})/F)$. It is a simple matter to calculate the intervals in the old grid that contain the endpoints of each H_i, and from that identify the nonempty intersections in equation (3.14b). Generally, only two or three intervals in the old grid are involved. The remaining equations are implemented directly, and do not require further discussion.

As discussed in the previous section, we are most concerned with the possibility that the accuracy of the density approximation may degrade as time progresses. We can begin by visually comparing the approximate density with the exact density obtained from the Kalman filter. Since we are going to look at the density after considerable time has passed. we will use $F=1$ so that the system has reasonable long-term behavior. Figures 4.1-3 show the comparison after 500 iterations for a 32 point grid and three combinations of noise covariances. Visually, at least, the approximate density is a good representation of the actual density. Reviewing the density comparisons over the complete history reveals that the approximation maintains nearly the same degree of fidelity throughout. This tends to support the observation from the previous section that the error may reach a steady state condition instead of constantly increasing.

Visually, the approximation seems stable, but what can we say numerically? The obvious choice of error to calculate is the ratio error bound defined in the previous chapter. Unfortunately, calculation of that error measure is complicated by the presence of two additional sources of error: first, the error introduced by the finite word length FFT, and second, the error resulting form the truncation of the Gaussian densities. Both these contributions are of the same magnitude as the density itself near the tails. This can result in arbitrarily large error ratios as one moves away from the center. Thus, the error criterion of the last section tends to give inflated results. Restricting attention to the central intervals minimizes these contributions but is not rigorously justifiable. Even so, it can provide information on the general trends of the error, so we will use the ratio error restricted to the interval ± 1.5σ about the mean of the actual density. In other words,

$$\mathrm{RatioError} = \max_{|x-\mu| \leqslant 1.5\sigma} \frac{|\pi(x) - \hat{\pi}(x)|}{\pi(x)}$$

where μ and σ are those of the actual density. Figures 4.4-6 show this measure versus time for the same situations as above, averaged over 20 runs. Note that the error remains more or less constant over the entire run. This again supports the proposition that the algorithm achieves a steady state error. Another useful numerical comparison is between the moments of the approximate density and those of the true density. Figures 4.4-6 also show the averaged error in the first two central moments. The error in the mean is given in percent of the actual standard deviation. The error in the variance is in percent of the actual variance. Once again, the errors are seen to be reasonably constant.

How do these results depend on the particular parameters used for these first examples? There are four different parameters to consider: the system dynamic

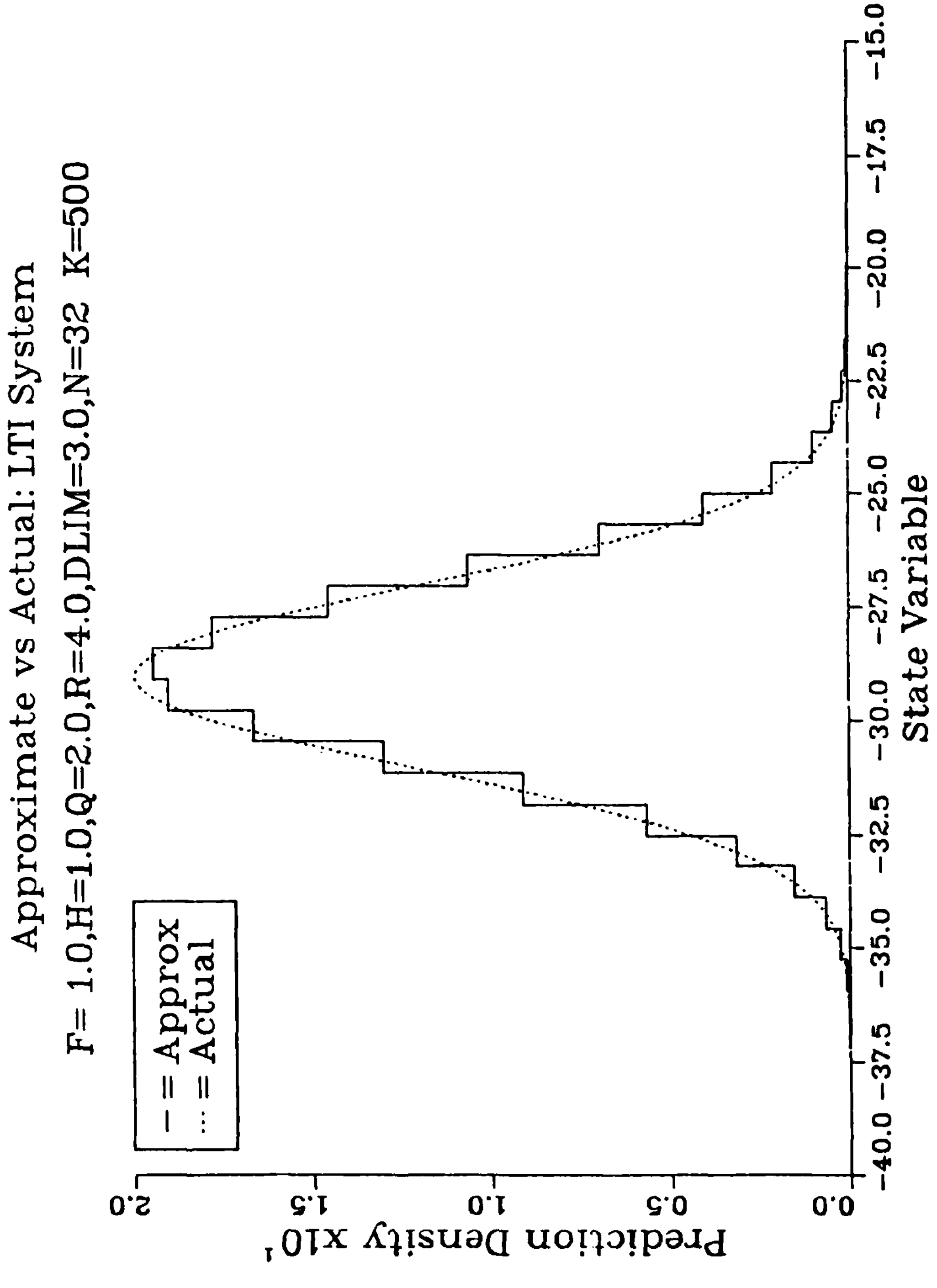


Figure 4.1 - Actual versus approximate density for linear time invariant system with Gaussian noise, after 500 time steps.

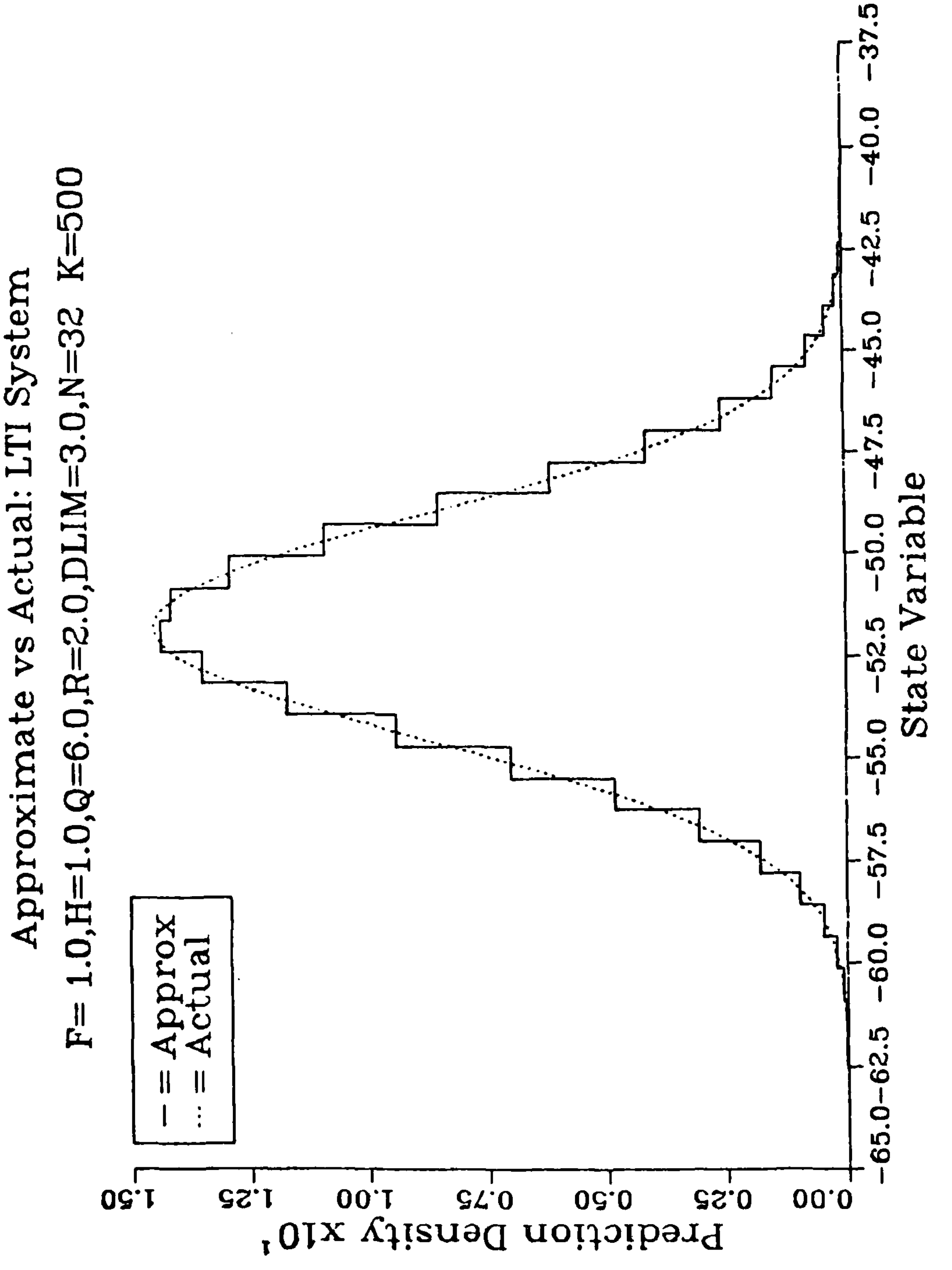


Figure 4.2 - Actual versus approximate density for linear time invariant system with Gaussian noise, after 500 time steps.

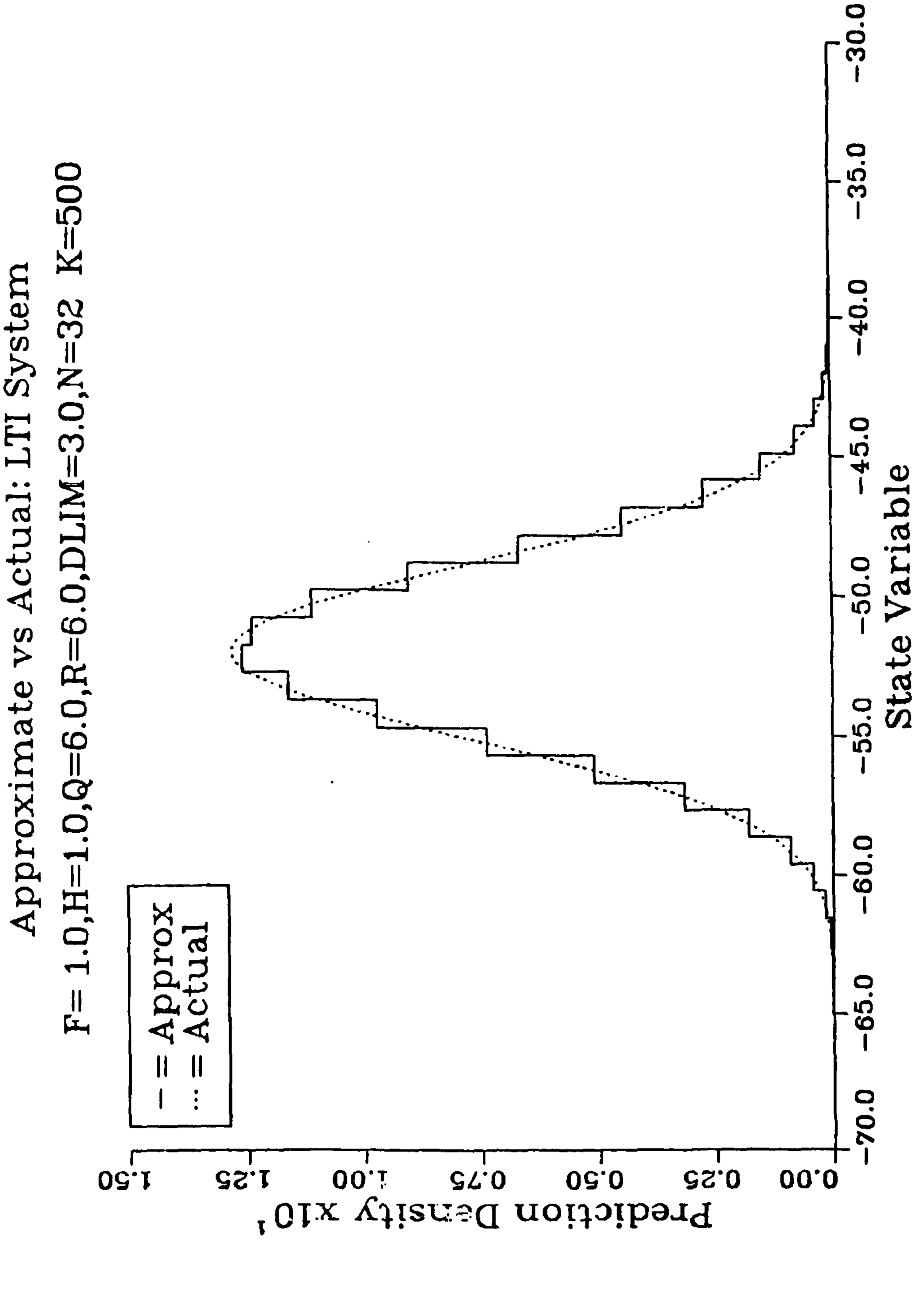


Figure 4.3 - Actual versus approximate density for linear time invariant system with Gaussian noise, after 500 time steps.

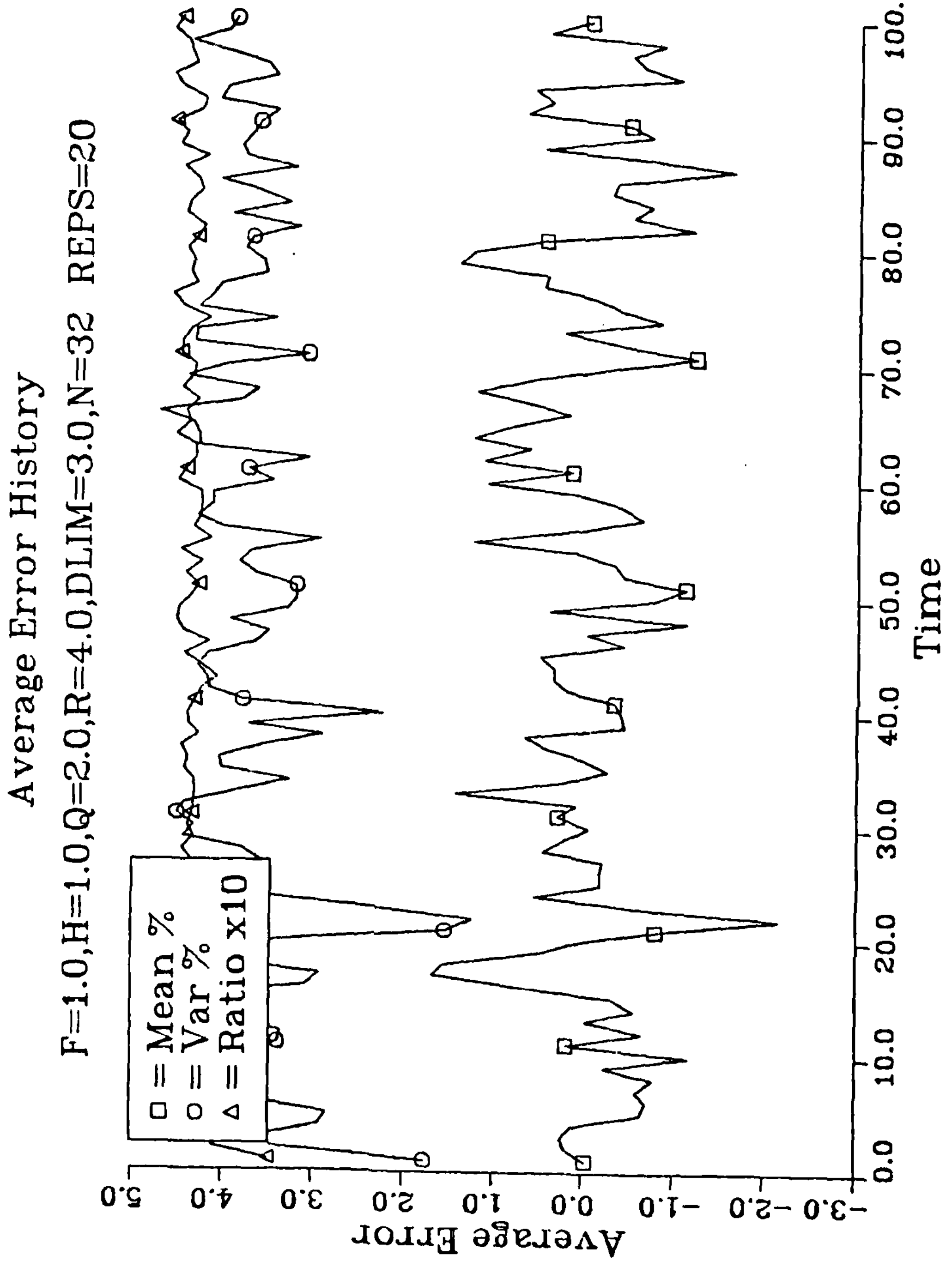


Figure 4.4 - Average error history of approximate density for linear time invariant Gaussian system.

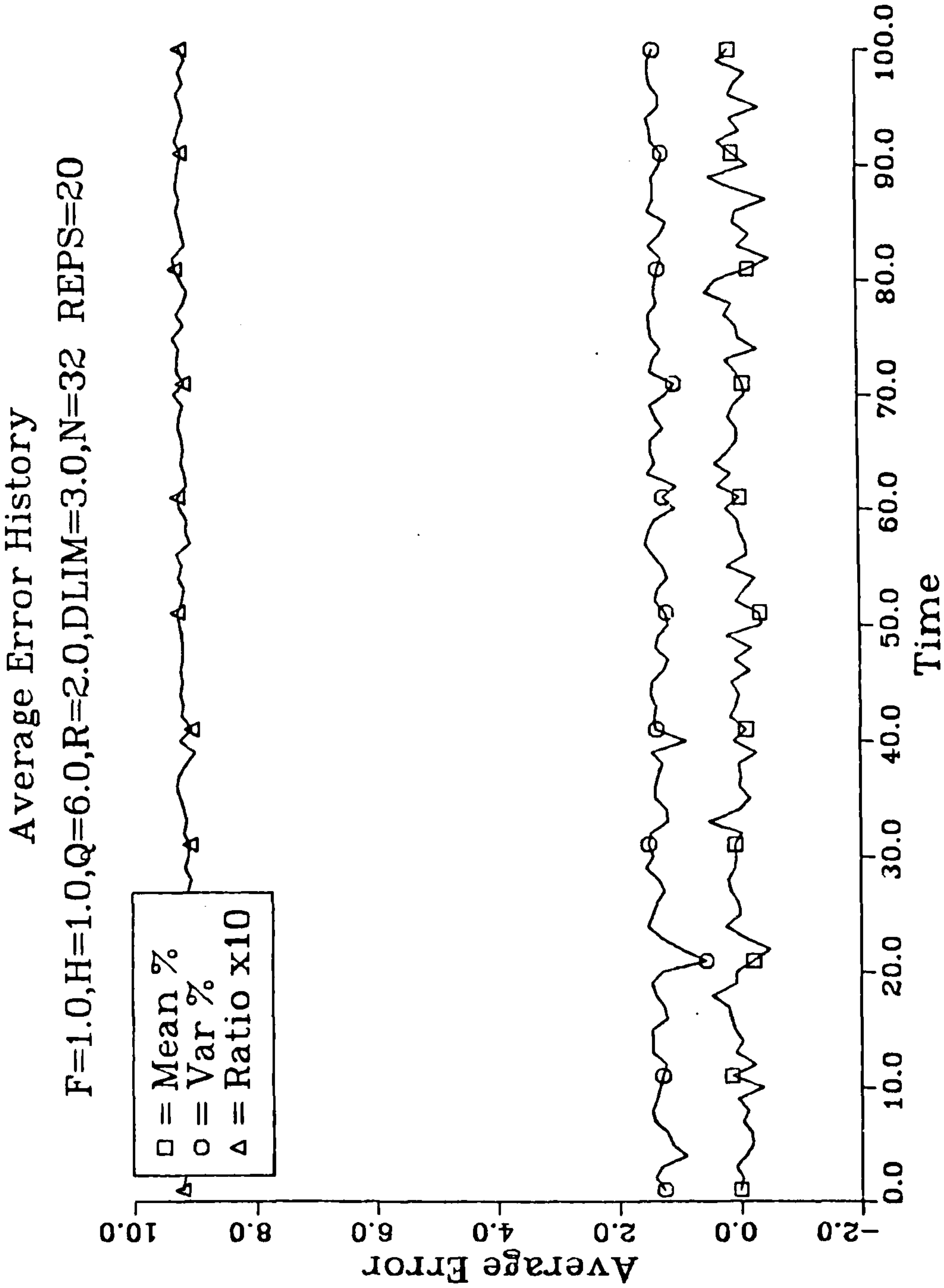


Figure 4.5 - Average error history of approximate density for linear time invariant Gaussian system.

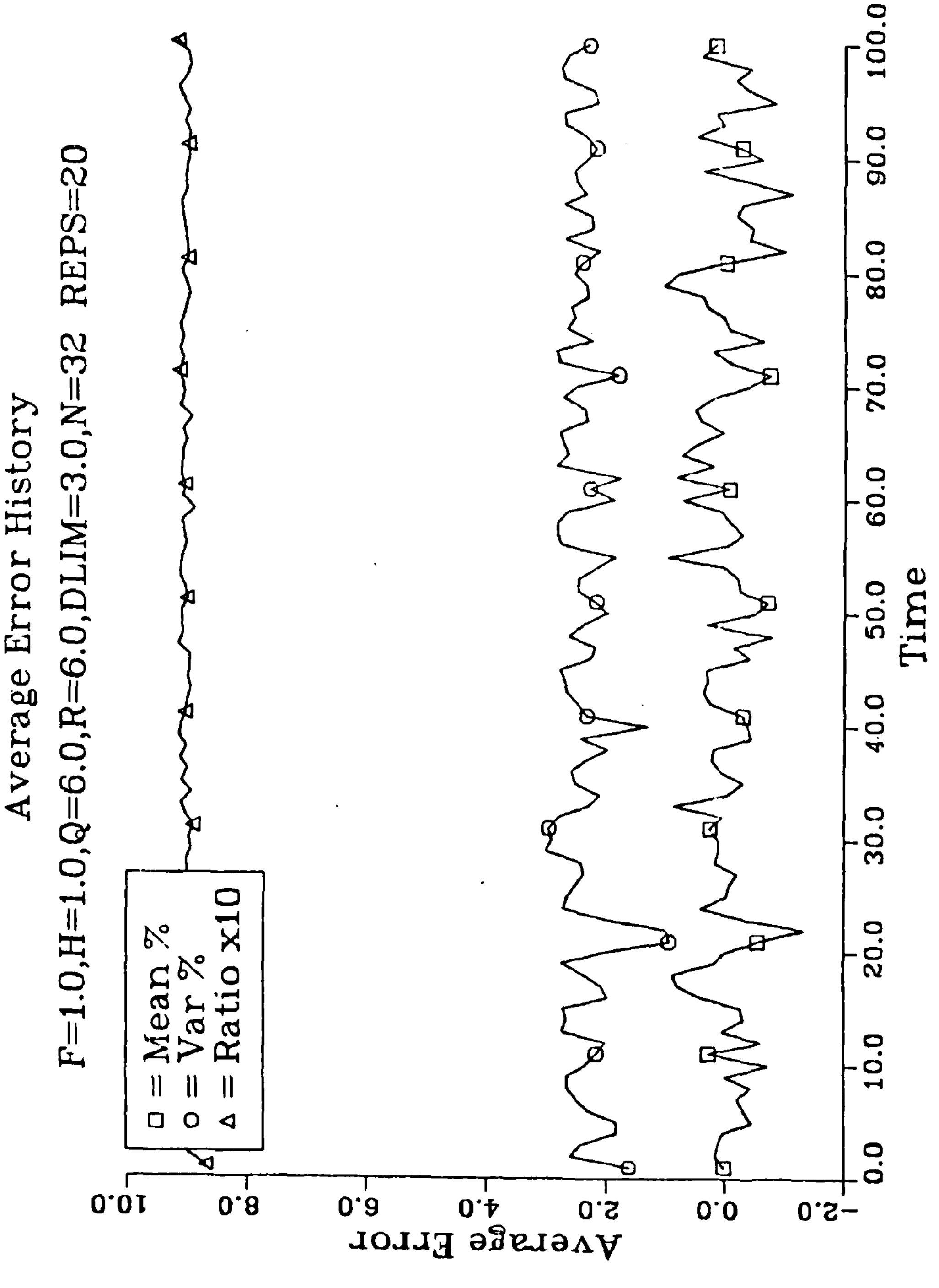


Figure 4.6 - Average error history of approximate density for linear time invariant Gaussian system.

constant F, the noise variances, the number of grid points, and the point at which the Gaussian densities are truncated. The first two are parameters of the system, and the second two are parameters of the approximation algorithm. We will use the time-averaged mean and variance of the numerical error measures defined above to look at the effects these parameters have. Table 4.1 summarizes the results of the various cases to be considered.

Look first at changing the system constant F. Figures 4.7-9 show the resulting density approximations after 20 time steps for three different cases: stable, unstable, and oscillatory systems. We would expect these changes to have little effect on the performance of the algorithm, and the figures and the data from Table 4.1 bear this out. The variations in the error statistics are very small.

Next consider different system noise variances. Figures 4.1-3 showed three situations different only in the noise variances, and visually there was little difference. However, looking at the numerical data in Table 4.1, we notice a pronounced change as the ratio of system input noise variance to observation noise variance changes. Note, though, that it is the ratio that is important, not the individual magnitudes. The statistics for the base case ($Q=2$, $R=4$, so $\frac{Q}{R}=0.5$) are virtually identical to those for $Q=6$, $R=12$ ($\frac{Q}{R}=0.5$ again). As the ratio of the variances gets smaller, the errors increase in all categories. This is because, as the ratio gets larger, the data in the current measurement is weighted more relative to the data in the old prediction density. This effectively reduces the dependence on the past data, and hence reduces the carryover of old errors. In terms of the densities, the smaller the ratio becomes, the more sharply spiked μ (and hence ψ) becomes compared to r. Thus $\hat{\pi}$, the result of the convolution of $\hat{r}$ and $\hat{\psi}$, looks increasingly like a shifted version of $\hat{r}$, with only the error associated with approximation of r and no carryover from past iterations. So the algorithm performance does depend to some degree on the system parameters, but in a predictable and reasonable way.

We turn now to varying the algorithm parameters. Figures 4.10-12 show the density approximations for 8, 12, and 16 grid points. As we would expect, the fidelity of the approximation improves as we increase the number of grid points. Similarly, the data in Table 4.1 shows decreasing errors as the number of grid points increases. Interestingly, even with as few as 8 points, we get a fair representation of the density.

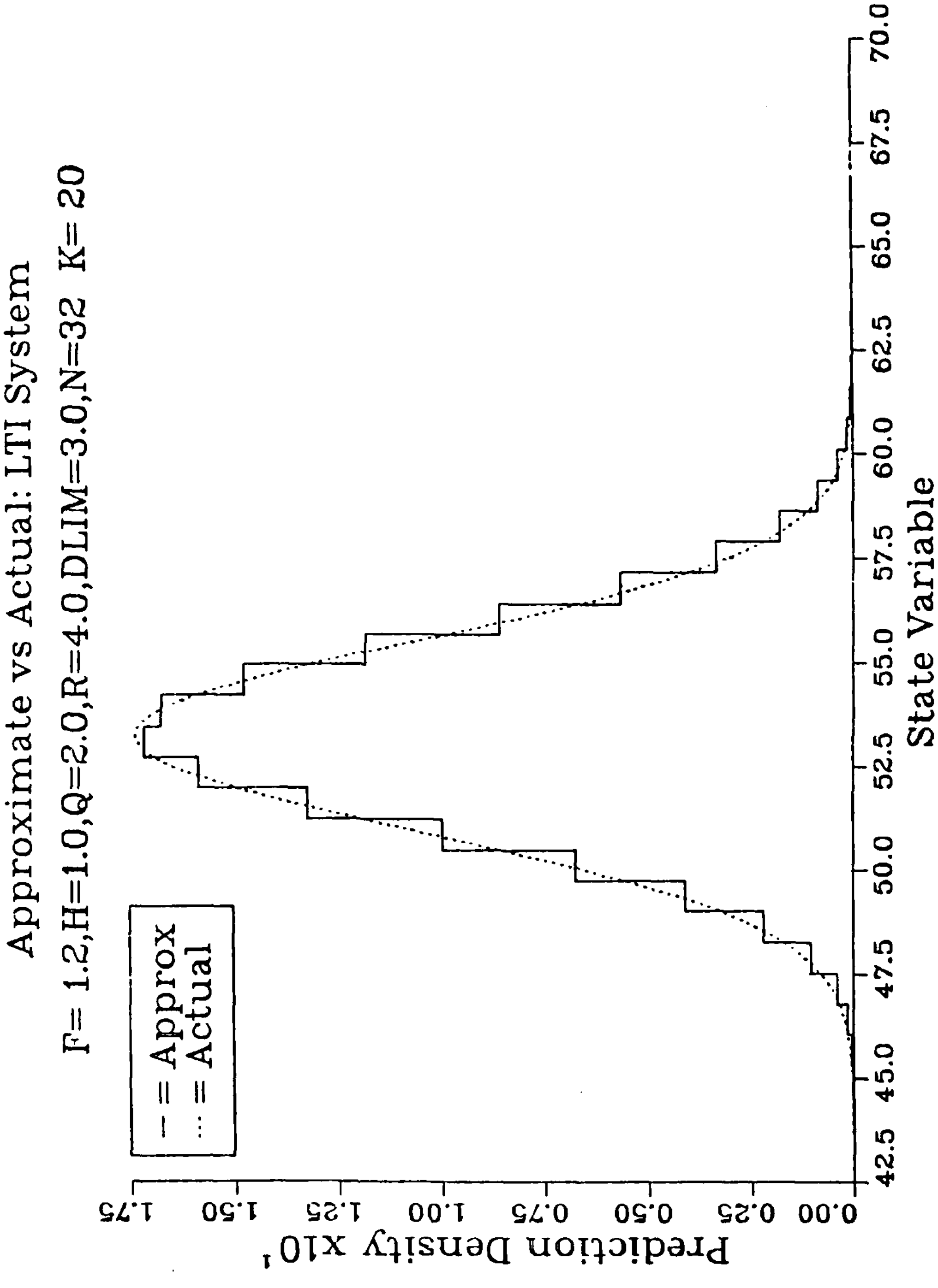


Figure 4.7 - Actual versus approximate density with $F = 1.2$.

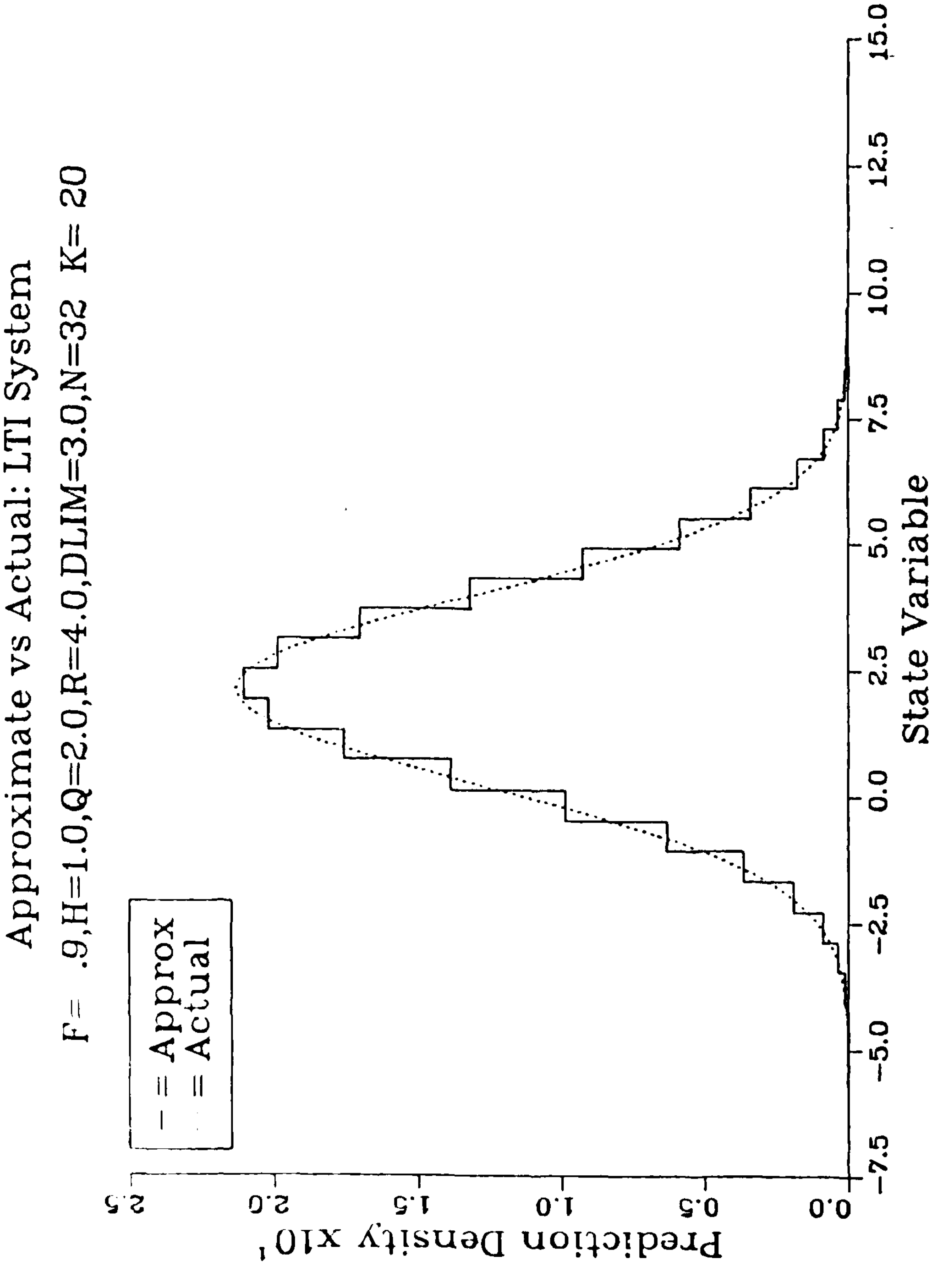


Figure 4.8 - Actual versus approximate density with $F = 0.9$.

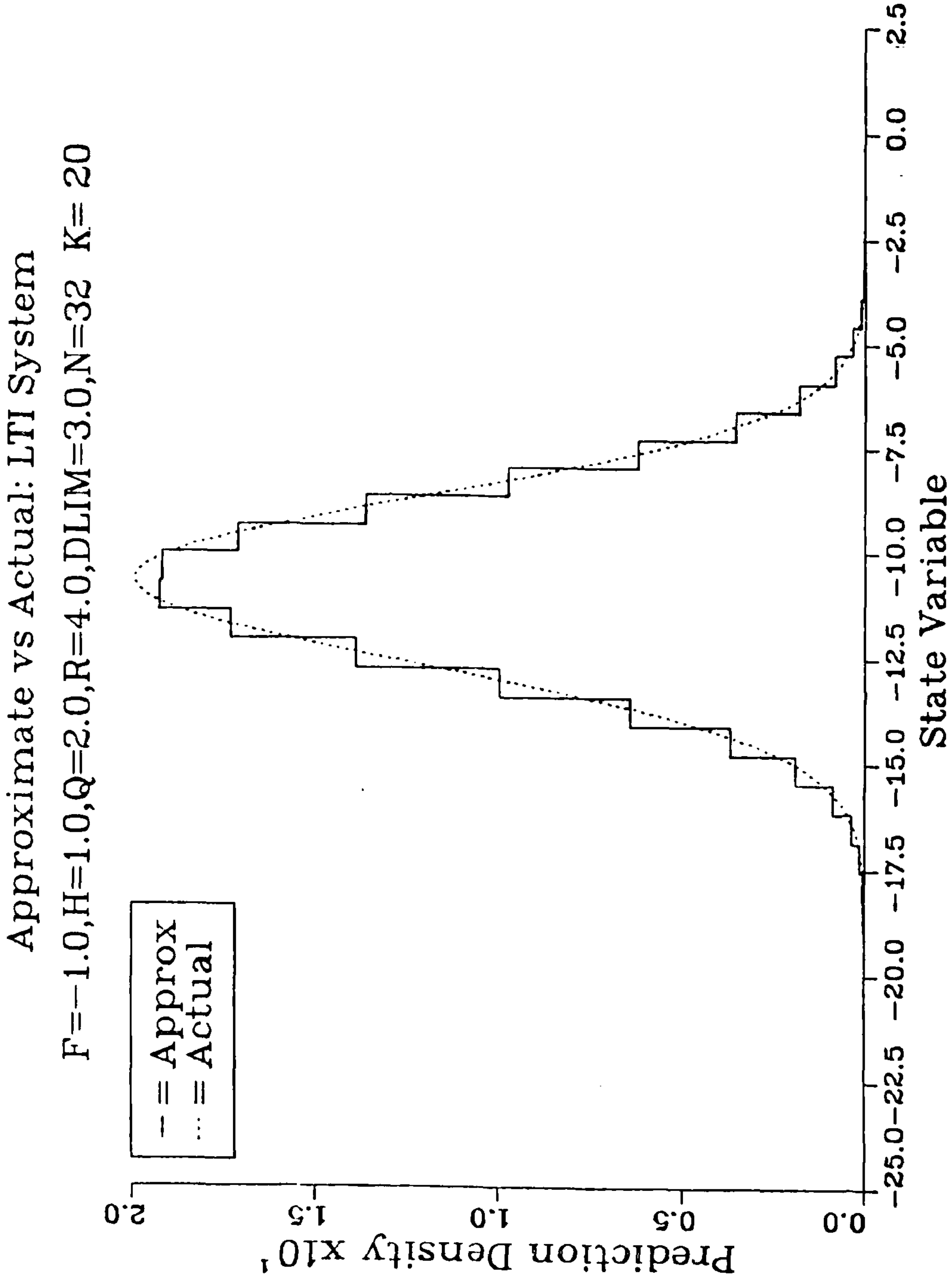


Figure 4.9 - Actual versus approximate density with $F = -1.0$.

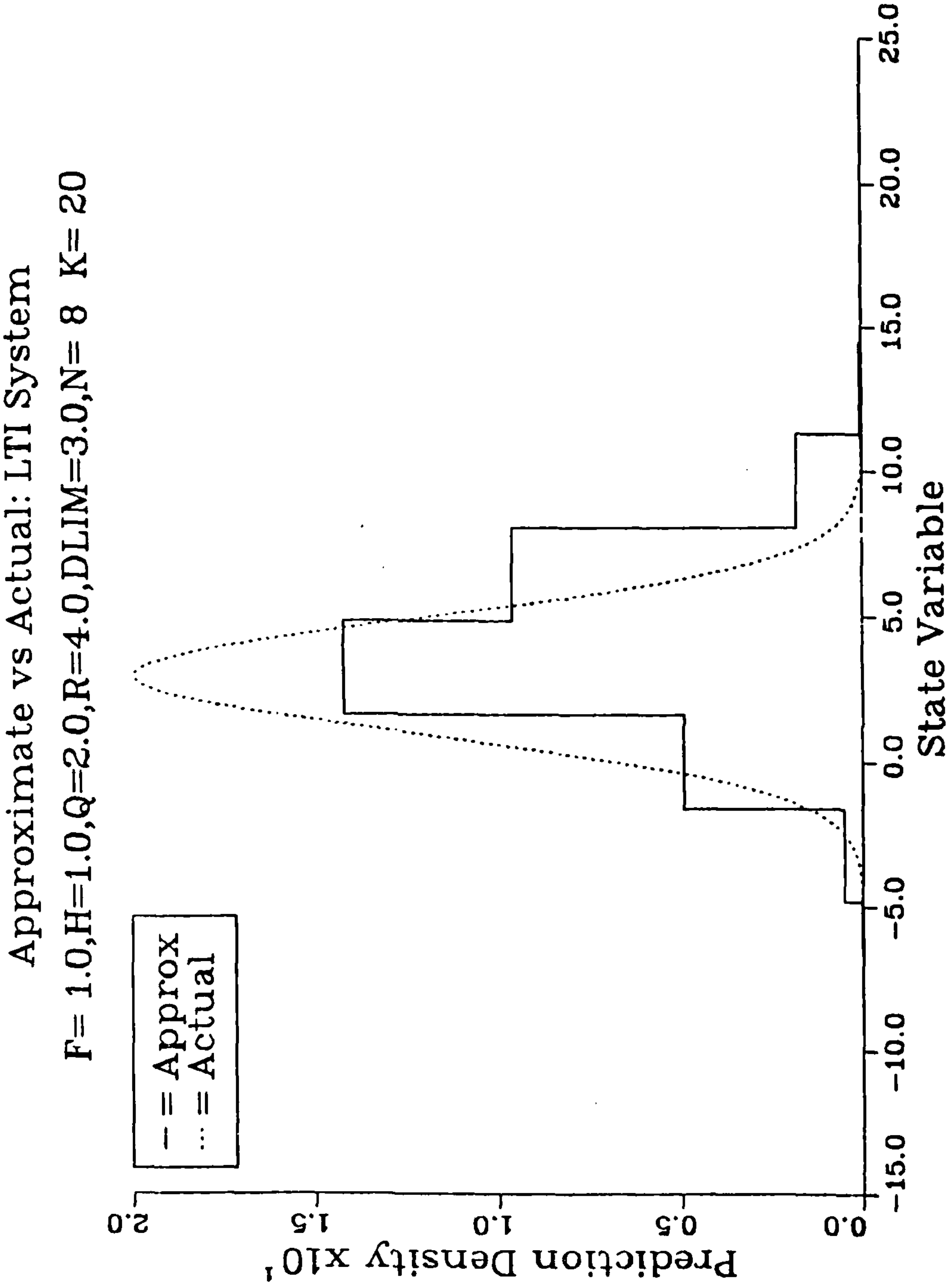


Figure 4.10 - Actual versus approximate density with 8 grid intervals.

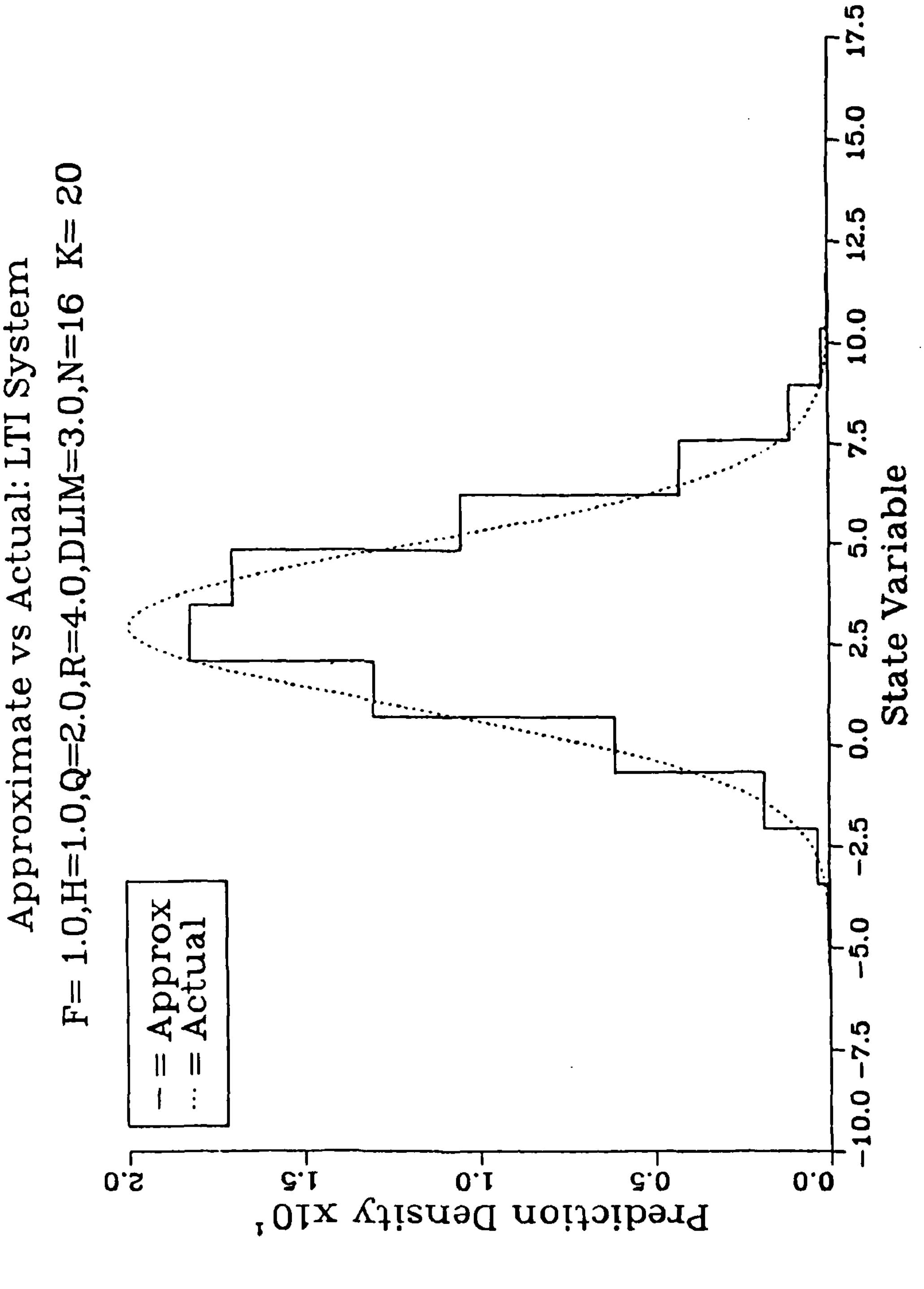


Figure 4.11 - Actual versus approximate density with 16 grid intervals.

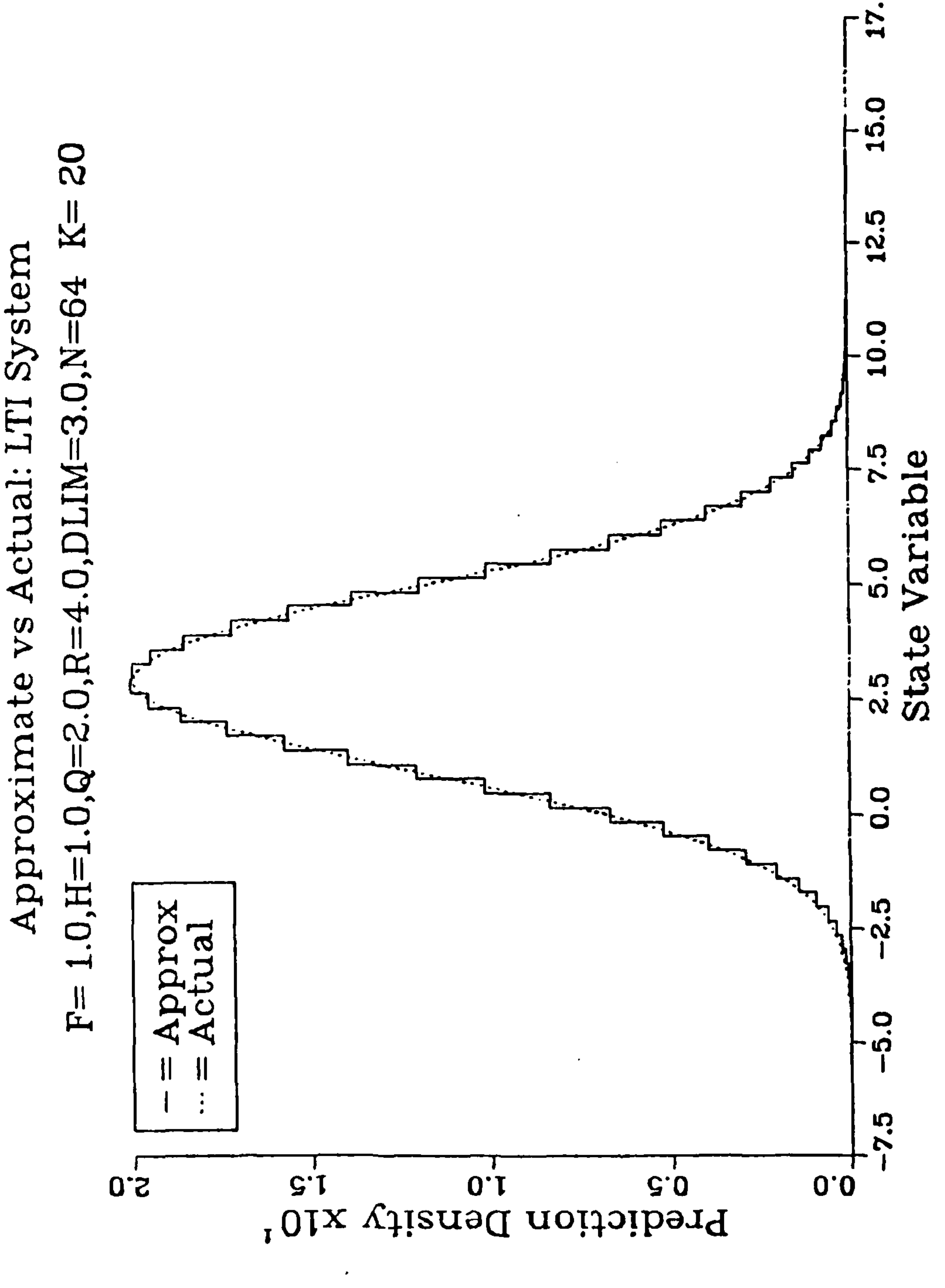


Figure 4.12 - Actual versus approximate density with 64 grid intervals.

Case	Error in Mean (%)		Error in Variance(%)		Maximum Ratio Error (%)	
	Mean	Var	Mean	Var	Mean	Var
Base	-.116	8.95	3.71	5.58	43.5	25.9
$F=1.2$	.0213	8.22	3.60	5.12	42.7	18.6
$F=0.9$	-.0576	7.60	3.45	4.58	42.8	21.7
$F=-1.0$	-.0240	7.23	3.88	3.70	43.6	24.5
$Q=6$, $R=2$	-.0236	.735	1.31	.413	31.9	3.27
$Q=6$, $R=6$	.0436	3.37	2.39	2.51	37.5	9.98
$Q=6$, $R=12$	-.115	8.94	3.71	5.57	43.5	26.3
N=8	-.644	792	111	108	2.66e+3	1.05e+7
N=16	-.293	74.4	20.8	13.0	149	1.73e+3
N=64	-.0589	2.73	-.897	5.09	17.0	5.16
± 4σ truncation	-.0361	12.0	8.95	.788	66.8	88.23
± 2.5σ truncation	-8.42e-3	19.6	-1.74	21.8	32.8	30.3

System: $x_{k+1}=Fx_k+w_k$; $z_k=x_k+v_k$; w_k Gaussian $(0,Q)$, v_k Gaussian $(0,R)$
Base case: $F=1$, $Q=2.0$, $R=4.0$, Gaussians truncated at ± 3.0σ,
N=32 grid points
Average over 1000 time steps

Table 4.1 - Summary of Errors in Approximate Density as Parameters are Varied

Despite the rather large ratio errors for that case, the moments are the approximate density are not too far off. The average error in the mean is less than 1% and 95% of the time the approximate mean is within $.56\sigma$ of the actual (based on a Gaussian error distribution and a 2σ bound). It is also worth noting that we begin to get decreasing returns as we go from 16 to 32 and then to 64 grid points. This points out the importance of balancing desire for numerical accuracy and computational burden. In practice, it may be necessary to experiment at the beginning to select a reasonable number of grid points for the particular application.

Finally, look at changing the truncation point for the Gaussian densities. Figure 4.13 shows the resulting approximate density when the Gaussians are truncated at $\pm 4\sigma$, and Figure 4.14 shows the result for $\pm 2.5\sigma$. Visually, there appears to be little difference. Numerically, however, we note a degradation for both cases in comparison to the base case. As we increase the truncation point, we include only tiny additional probability mass, but force the grid to cover a larger interval with the same number of points. As a result, the densities appear more sharply peaked, and so, as predicted by the error analysis of the previous chapter, the ratio error increases, as well as the errors in the mean and variance. On the other hand, reducing the truncation point discards an increasingly significant amount of mass, even though it improves the grid coverage in the central region of the density. This causes the observed reduction in ratio error with increased variability of the moment errors. The increased variance of the moment errors for the 2.5σ cutoff also occurs because reducing the truncation point aggravates a problem that any truncation of a infinite density can generate. As you can see in Figure 4.14, the approximate density for this case has a fairly obvious skew to the right relative to the actual density. This is caused by a noise sample near the truncation point. The multiplication for the measurement update will now emphasize the region near the truncation, resulting in a distortion on one side of the density. The more severely the density is truncated, the worse the problem will be. Care must be taken in selecting the truncation point for a particular application, and, as with selection of the number of grid points, some experimentation may be required.

Two more features of the algorithm are worth noting. First, the error in the variance appears to have a definite positive bias. In fact, from looking at the average of the absolute value of the error, it seems that the variance of the approximate density is almost always greater than the actual variance for most cases. In other words, the approximate density almost always overestimates the variance. This is intuitively reasonable when you consider the form of the approximation. Over any interval, the approximation is usually greater than the actual density on the side away from the mean, and smaller on the side closer to the mean. Hence the approximation will show a

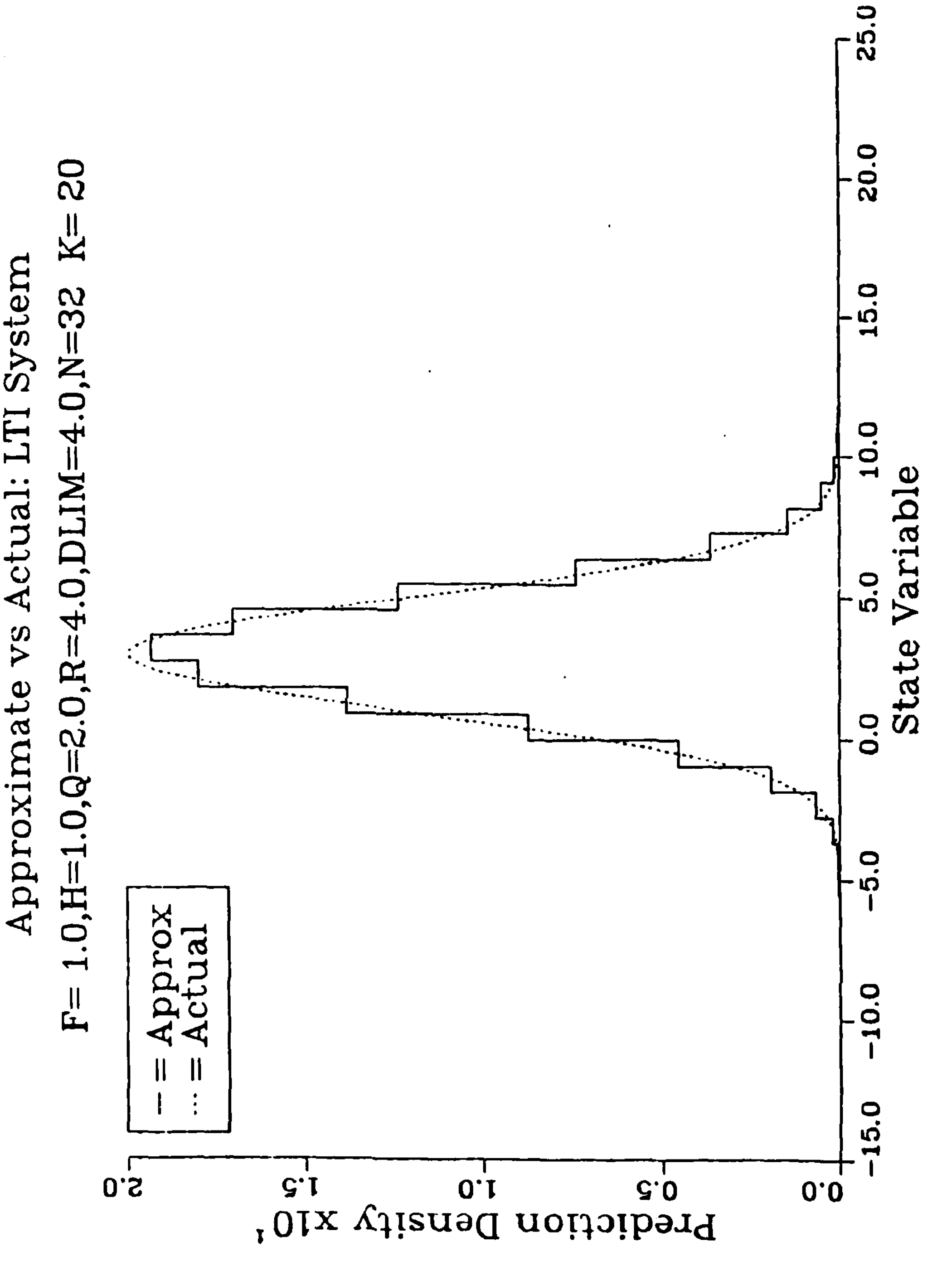


Figure 4.13 - Actual versus approximate density with Gaussians truncated at ± 4.0σ.

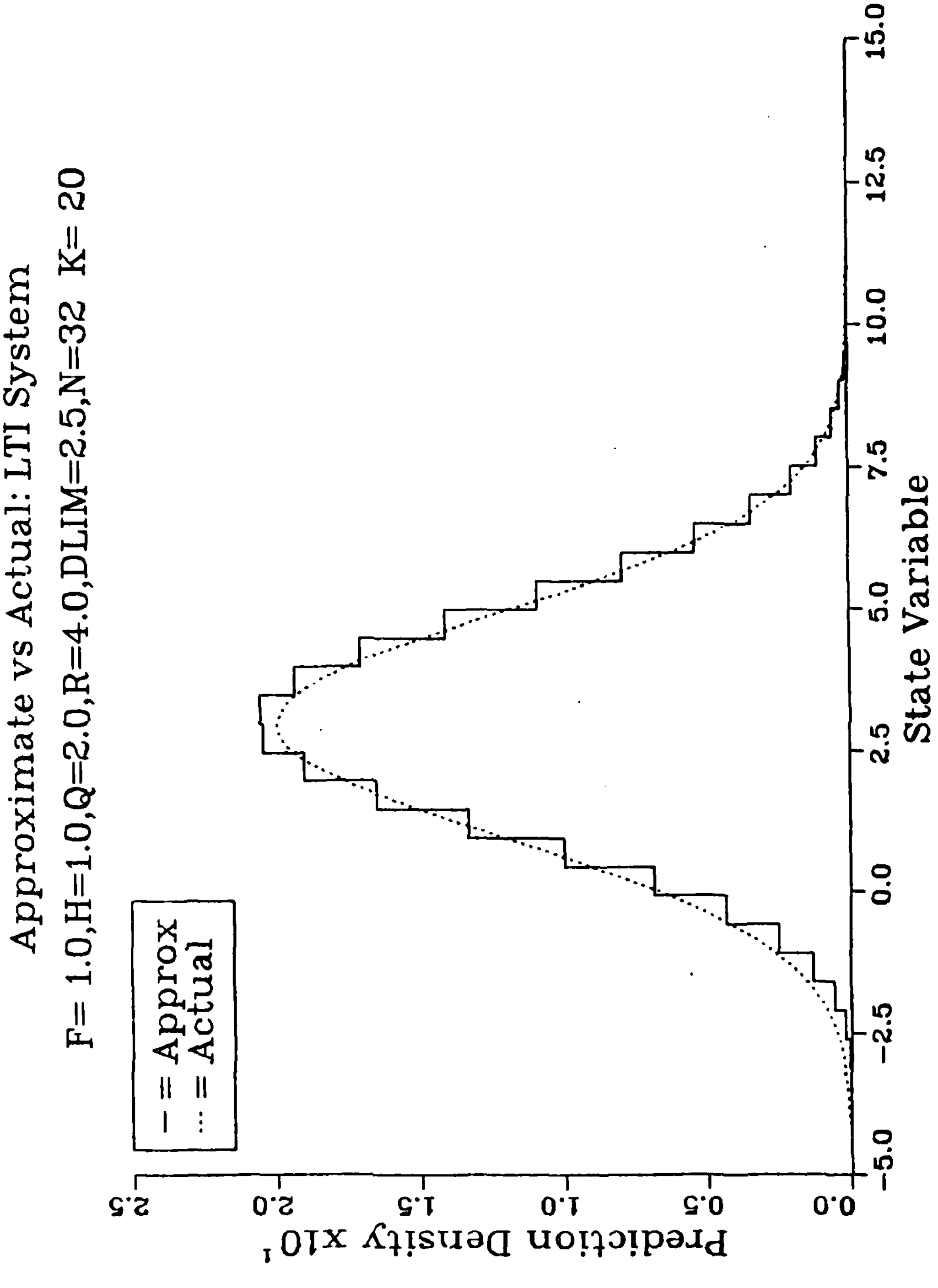


Figure 4.14 - Actual versus approximate density with Gaussians truncated at ± 2.5σ.

greater spread than the actual. For the mean, however, the differences will tend to balance each other, so we do not see a bias in the mean of the approximate density. This is a very nice property, since it says that the algorithm generates a consistently conservative approximation to the actual density. The two exceptions to this general observation are the cases for N=64 and truncation at ± 2.5σ. In both cases the effect of lost mass beyond the truncation point cancels the effect described above.

Second, you may have noted from the graphs that the interval which is the peak of the approximate density often contains the peak of the actual density. In the runs made to generate the data in Table 4.1, this occurred about 95 percent of the time, consistently for all cases. This effect is related to the interpretation of the algorithm as a probability mass filter. The algorithm almost always assigns greatest mass to the appropriate interval.

To close this section, let's now relax the time-invariant restriction. Figure 4.15 shows the density comparison for a system with Gaussian noises with time-varying variances given by

$$Q_k = Q_0 + \sin(\frac{2\pi k}{13}) \; ; \quad R_k = R_0 + \sin(\frac{2\pi k}{9})$$

As you can see, the algorithm handles this case without any difficulty. Numerically, the errors are comparable to the base time-invariant case.

4.2. Sign-only Observations

We have seen in the last section that the approximate algorithm generates a well-behaved, stable approximation to the actual prediction density. There is little use in continued application of the algorithm to the linear Gaussian case of the last section though, because we already have an exact solution for that case. More interesting is to look at nonlinear non-Gaussian systems, and use the algorithm to explore their behavior. For example, having a representation of the complete posterior density allows us to actually calculate the optimal estimate for any given cost function. Thus we can compare the performance of any approximate point estimator to the theoretical optimum. Alternatively, we can use the algorithm to look at the effects of different system structures on the information in the observations. Many applications are possible.

As a simple illustrative example, consider a system with linear dynamics and Gaussian input noise, but where we observe only the sign of the output. The system is

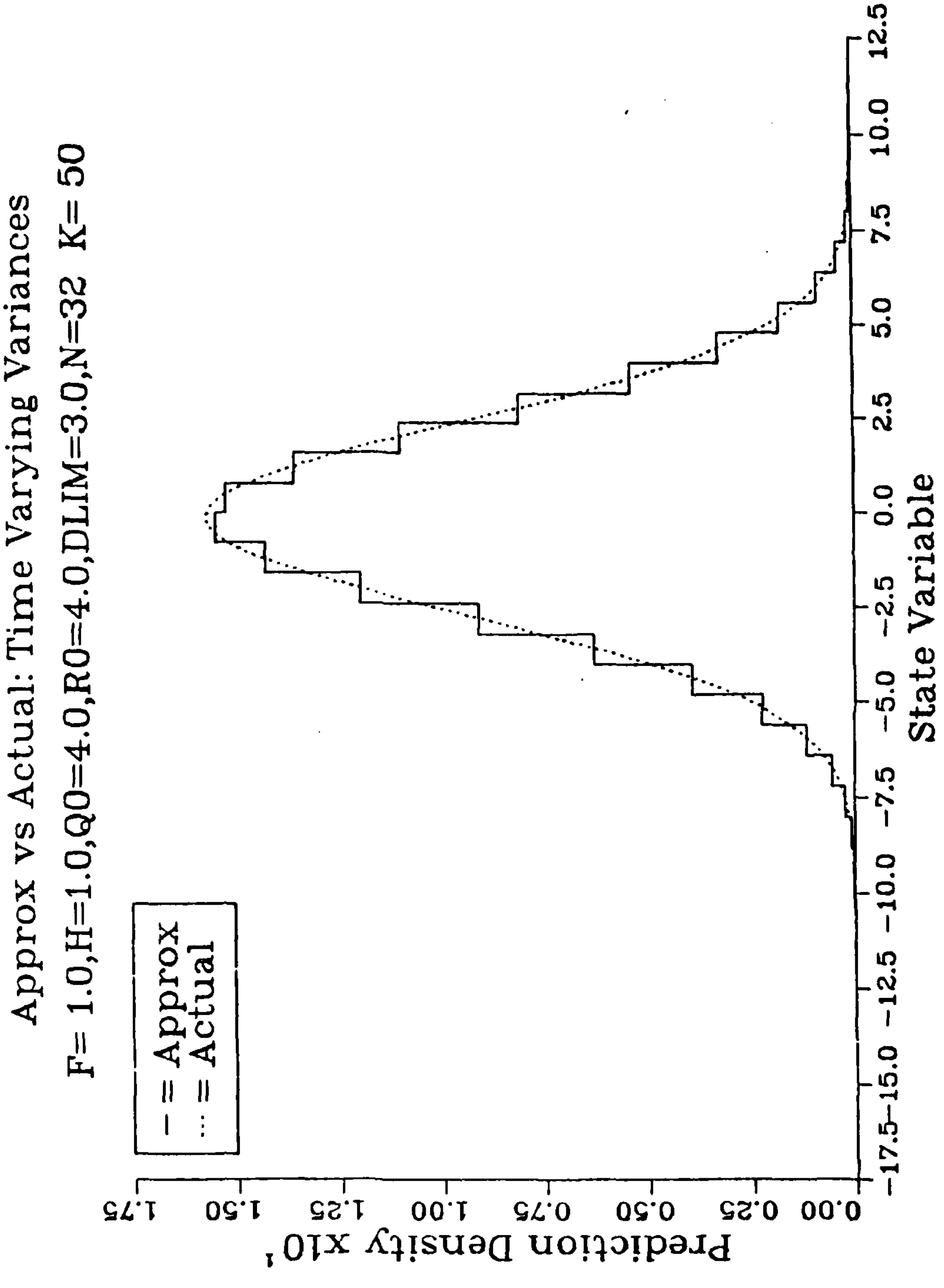


Figure 4.15 - Actual versus approximate density for linear system with time-varying noise variances.

described by

$$x_{k+1} = Fx_k + w_k$$

$$z_k = \mathrm{sgn}(x_k + v_k)$$

where $\mathrm{sgn}(x)$ is the sign function defined by

$$\mathrm{sgn}(x) \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

It is interesting to note that although this is a simple nonlinearity, there is no EKF for this system, since we cannot linearize the sgn function.

It is a simple modification to apply the algorithm to the above system, since the algorithm is nicely modular. The only change in the system is in the observation equation, so the only change in the algorithm is in the measurement density μ. For this case we have

$$\mu(x,z) = c_1(x)\delta(z-1) + c_{-1}(x)\delta(z+1)$$

where δ is the Dirac delta or 'impulse' function, and c_1 and c_{-1} are given by

$$c_1(x) = \mathrm{Prob}(x+v \geqslant 0) = \int_{-x}^{\infty} (2\pi R)^{-\frac{1}{2}} \exp(\frac{-y^2}{2R})\, dy$$

$$c_{-1}(x) = \mathrm{Prob}(x+v < 0) = \int_{-\infty}^{-x} (2\pi R)^{-\frac{1}{2}} \exp(\frac{-y^2}{2R})\, dy = 1 - c_1(x)$$

Other than this, the algorithm is implemented exactly as it was for the first section. Also as in the first section, we will use a 32-interval grid, and truncate the Gaussians at $\pm\, 3\sigma$.

One thing that would be nice to know for this system is how much we lose only having the sign for the observation. Or conversely, how much could we gain if we had full observations (sign plus magnitude). To at least start to answer this question, we can compare the approximate prediction density for the sign-only system to the output from a Kalman filter operating on the full measurements. After a little experimenting with the system it becomes apparent that there are two basic regimes to consider. First, where the state remains in the vicinity of zero, and second, where it does not. In the first case, we get reasonably frequent switches in the sign of the output. As might be expected, this leads to a roughly Gaussian shaped density. Figure 4.16 shows such a

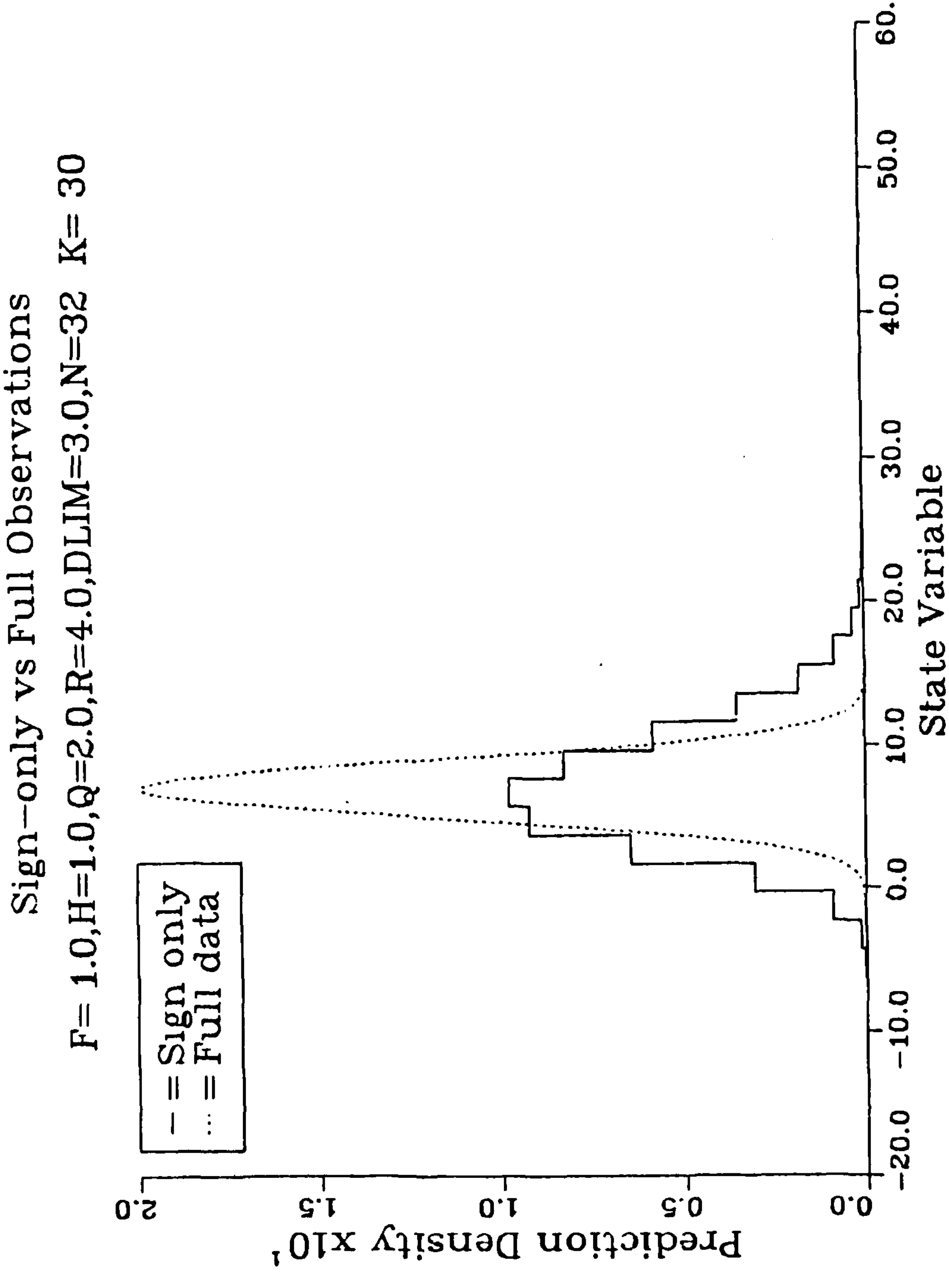


Figure 4.16 - Sign-only versus full observations for $F = 1.0$.

situation. Note that the means of the two densities are quite close. The density for the sign only case, however, has a noticeably larger variance, indicating the reduced information content of the sign-only observations. In the second case, where the state drifts away from zero, the observations are consistently the same sign. Now we know only which side of zero the state is on, but not how far away it is. We would expect, then, that the density would begin to smear out, displaying a large degree of uncertainty about the state. Figure 4.17 shows the two densities for this case. Again, the means of the two densities are not grossly different, but the variance of the sign-only density is considerably larger. The sign-only density is also clearly skewed away from zero, showing the larger degree of uncertainty in that direction.

Looking at the time history of the density gives additional information. For simplicity, we will just consider the mean of the two densities, since these provide a reasonable measure of the relative locations of the densities. Figure 4.18 shows this for the same system parameters as in Figure 4.16. Note the extreme response of the sign-only density to changes in the sign of the observation. As noted before, this is because the data does not contain information on how far the state has moved. The latter half of the graph shows the effect of the sign-only density spreading out in the positve direction.

At this point we can make some general observations. First, even at best, any state estimate based only on the sign of the output will have poor performance compared to having full observations. Second, a sign-only estimator will perform best when the state is near zero so that there frequent changes in the sign of the data. These are hardly surprising, and might even be intuitively obvious, but the above is the only reasonable way to actually demonstrate them.

We could continue by extracting numerical data, and quantifying the behavior of this system, but since this example is only intended to demonstrate the application of the approximate density algorithm, we will not. It is sufficient to point out that analyses of this type can only be accomplished by calculating the complete posterior density, and that this algorithm provides a powerful means of doing so.

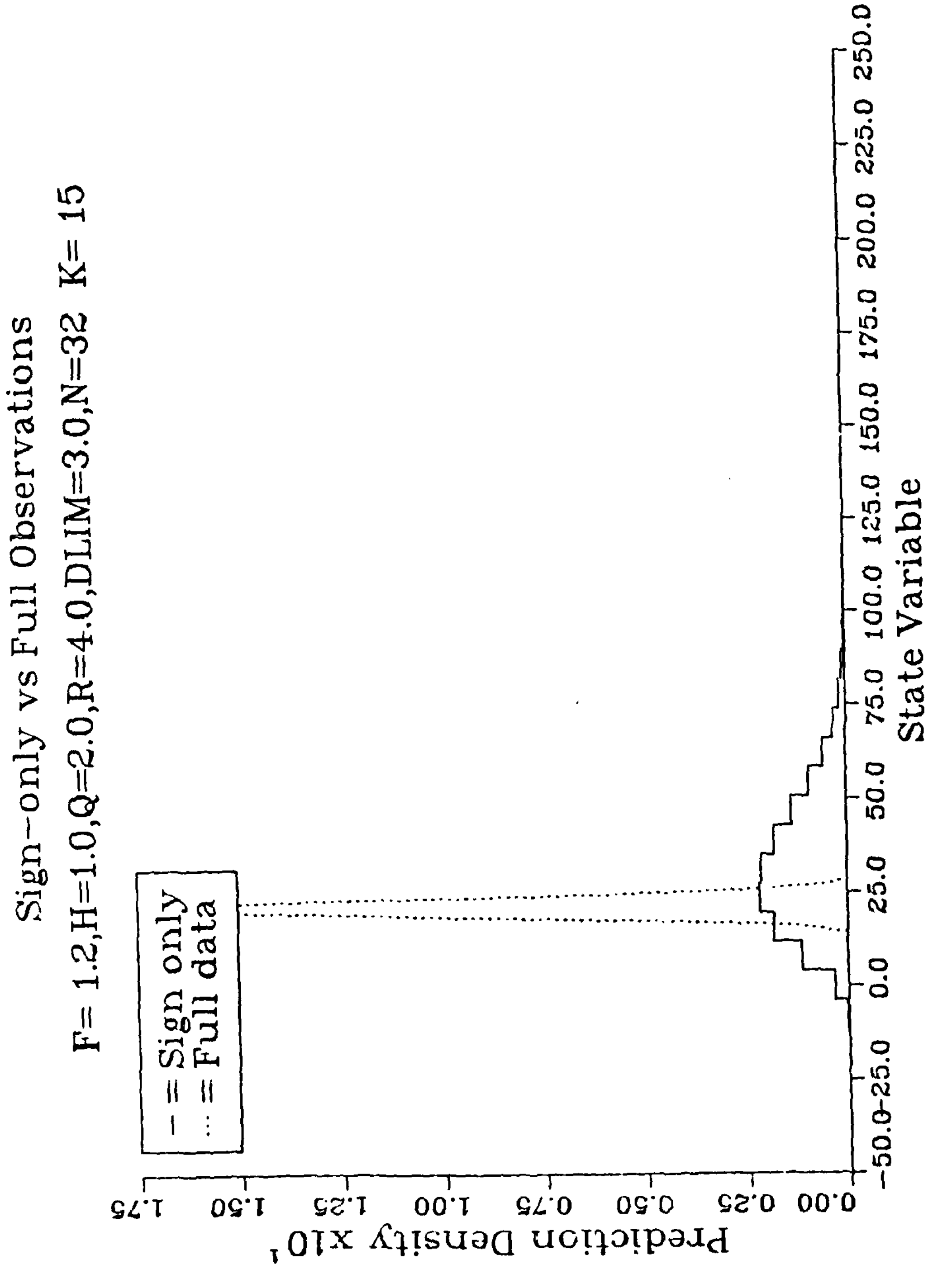


Figure 4.17 - Sign-only versus full observations for $F = 1.2$.

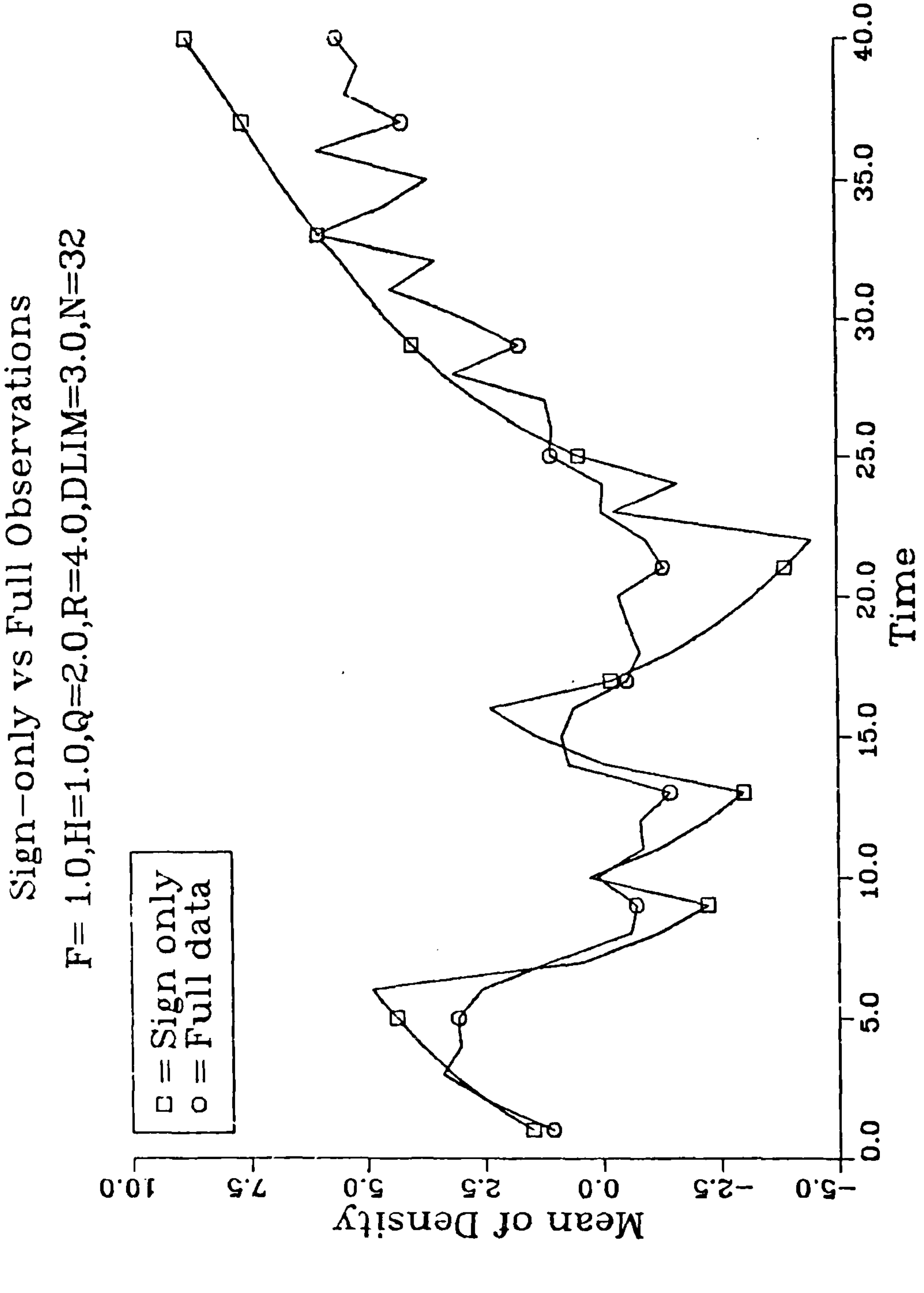


Figure 4.18 - Time history of mean of sign-only and mean of full observation densities for $F = 1.0$.

5. APPLICATION TO PARAMETER IDENTIFICATION

5.1. Introduction

Well, what good does this algorithm do us? The answer to that is highly problem dependent. Clearly, for many combinations of systems and objective functions, the traditional point estimators provide an acceptable answer. Furthermore, since this technique can be fairly expensive computationally, even a poor point estimator may be better for some applications. Rather than futilely try to find gross generalizations, probably the best approach is to identify particular applications, and explore them using this technique.

We can describe, however, several general ways that the algorithm can be of use to us. First, simply working with the equations and thinking of the process in terms of operations on densities provides a deeper understanding of the mechanics of the process. For more detail, we can use the close approximation of the posterior density that the algorithm generates to plot the prediction density or develop other useful descriptions of it. This can provide additional insight into the behavior of the system, and would be useful in much the same way as traditional simulation is for deterministic systems. Furthermore, as is done with deterministic simulation, we can use the technique to examine how the behavior of the system varies as we change features of the system. Finally, again since we have a representation of the posterior density, we can actually calculate the optimal estimator for any given loss function. Thus we can compare the performance of traditional point estimators directly to that of the optimal. In this way we can judge, for instance, whether apparently poor performance is a fault in the estimator, or is actually just the best that one can expect. This could be particularly useful in assessing transient performance, where there are virtually no theoretical results. To consider these uses in a concrete situation, we will look at them in the context of the *problem of simultaneous state and parameter estimation*. We will look first at using this approach to analyze the propagation of the prediction density, and then we will use it to evaluate the performance of a point estimator.

What is the parameter estimation (or system identification) problem? It is where we know (or assume we know) the system model except for a finite number of parameters. The object is to be able to estimate both the system state and the system parameters using the observations. Generally, the system can be nonlinear, and the parameters time varying. In practice, however, virtually all the work has been done for linear systems with time-invariant parameters. For that case, it has been possible to take advantage of the linear structure of the system to obtain reasonably effective point estimators. A large body of work exists concerning these estimators, which we will not try to review here. These approaches generally arise from considering the states and parameters as distinctly different entities. On reflection, though, there is actually little to distinguish the unknown parameters from the unknown states of the system. When we recall the Bayesian definition of randomness, we see there is really no distinction at all. From the Bayesian viewpoint, the unknown parameters are simply additional system states in a nonlinear system model. As Peterka [3] and Ljung [38] have pointed out, this is the only logically consistent approach. With this view of the problem, it is straightforward (although possibly algebraically involved) to apply the algorithm to the parameter identification problem.

5.2. Applying the Algorithm

For our first analysis, we will consider a scalar linear identification problem, where we have a model with linear dynamics with an uncertain transition parameter. The model is

$$x_{k+1} = Fx_k + u_k + w_k$$

$$z_k = x_k + v_k$$

where the transition parameter F is unknown, and where u_k is a known input sequence and w_k and v_k are uncorrelated, white, zero-mean, Gaussian noise sequences with variances Q and R respectively. The equivalent nonlinear vector system that we will use is given by[1]

$$x_{k+1} = \begin{bmatrix} x_{(1),k+1} \\ x_{(2),k+1} \end{bmatrix} = \begin{bmatrix} x_{(1),k} \\ x_{(1),k}x_{(2),k} + u_k \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} = f(x_k, u_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$z_k = x_{(2),k} + v_k = h(x_k \, , v_k)$$

where u_k, w_k and v_k are as above. We are now in a position to apply the algorithm

[1] Recall that the notation $x_{(i)}$ refers to the ith component of the vector x.

developed in this paper.

First, we must define the grid. For simplicity, we will use a rectangular grid, defined by the cells

$$\mathrm{I}_{k,j} = \Big\{x_{(1)} \in [a_{(1),k}(j_{(1)} - \frac{1}{2}) + b_{(1),k} \ , \ a_{(1),k}(j_{(1)} + \frac{1}{2}) + b_{(1),k}] \ ;$$

$$x_{(2)} \in [a_{(2),k}(j_{(2)} - \frac{1}{2}) + b_{(2),k} \ , \ a_{(2),k}(j_{(2)} + \frac{1}{2}) + b_{(2),k}]\Big\}$$

so in the definition given in chapter 3, the matrix A will be diagonal. The volume of each cell is therefore given by $\alpha_k = a_{(1),k} a_{(2),k}$. At each time step, we will define the new grid based on the smallest rectangle containing the support of the new prediction density. Note that since there is no noise on $x_{(1)}$ and no direct observation of that state, the support of the density along that dimension will not change. Hence, it will remain the same as the support of the initial density, and the grid parameters in that dimension will be constant.

Now consider each of the equations of the recursion (3.14) in turn. First is equation (3.14a), which is the computation of the inverse image of a grid cell. Since the mapping given by f above is one-to-one, the inverse image is a closed figure like the original cell, bounded by the inverse images of the sides of the original cell. The inverse image set is given by

$$\mathrm{H}_j = \Big\{x_{(1)} \in [a_{(1),k+1}(j_{(1)} - \frac{1}{2}) + b_{(1),k+1} \ , \ a_{(1),k+1}(j_{(1)} + \frac{1}{2}) + b_{(1),k+1}] \ ;$$

$$x_{(2)} \in [\frac{a_{(2),k+1}(j_{(2)} - \frac{1}{2}) + b_{(2),k+1} - u_k}{x_{(1)}} \ , \ \frac{a_{(2),k+1}(j_{(2)} + \frac{1}{2}) + b_{(2),k+1} - u_k}{x_{(1)}}]\Big\}$$

Figure 5.1 shows a typical H_j in relation to the grid at the previous time step.

Next look at equation (3.14b). Recalling that $\mu = p(z_k \mid x_k)$, we get

$$\mu(x, z_k) = (2\pi R)^{-\frac{1}{2}} \exp(-\frac{(z_k - x_{(2)})^2}{2R})$$

Now consider the limits on the integral of μ. Referring to figure 5.1, it is clear that since the grid in the $x_{(1)}$ direction does not change over time, we have $\mathrm{H}_j \cap \mathrm{I}_{k,i} = \emptyset$ if $j_{(1)} \neq i_{(1)}$. If $j_{(1)} = i_{(1)}$, we have

$$\mathrm{H}_j \cap \mathrm{I}_{k,i} = \Big\{x_{(1)} \in [a_{(1),k+1}(j_{(1)} - \frac{1}{2}) + b_{(1),k+1} \ , \ a_{(1),k+1}(j_{(1)} + \frac{1}{2}) + b_{(1),k+1}] \ ;$$

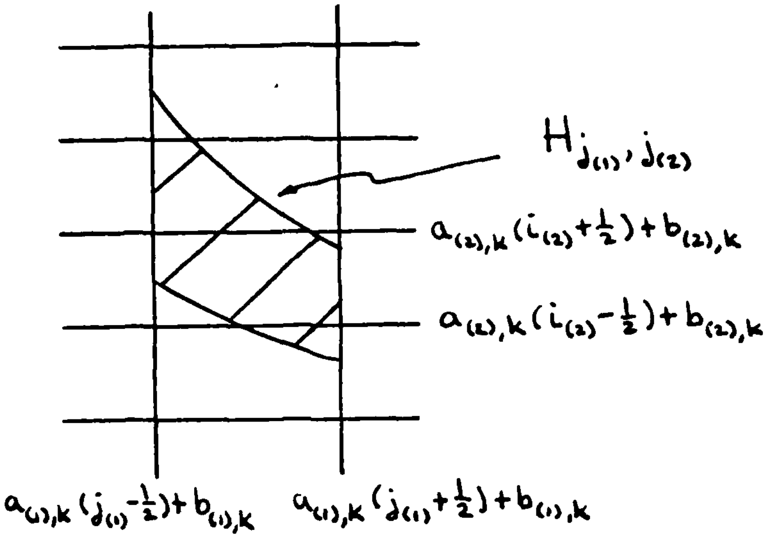

$H_{j_{(1)},j_{(2)}}$
$a_{(2),k}(i_{(2)}+\frac{1}{2})+b_{(2),k}$
$a_{(2),k}(i_{(2)}-\frac{1}{2})+b_{(2),k}$
$a_{(1),k}(j_{(1)}-\frac{1}{2})+b_{(1),k}$
$a_{(1),k}(j_{(1)}+\frac{1}{2})+b_{(1),k}$

Figure 5.1 - Typical H$_j$ in relation to old grid.

$$x_{(2)} \in \left| \frac{a_{(2),k+1}(j_{(2)}-\frac{1}{2})+b_{(2),k+1}-u_k}{x_{(1)}} , \frac{a_{(2),k+1}(j_{(2)}+\frac{1}{2})+b_{(2),k+1}-u_k}{x_{(1)}} \right|$$

$$\cap \ \left| a_{(2),k}(i_{(2)}-\frac{1}{2})+b_{(2),k} \ , \ a_{(2),k}(i_{(2)}+\frac{1}{2})+b_{(2),k} \right| \Big\}$$

$$= \Big\{ x_{(1)} \in |\xi_1(i,j) \ , \ \xi_2(i,j)| \ ; \ x_{(2)} \in |\varsigma_1(x_{(1)},i,j) \ , \ \varsigma_2(x_{(1)},i,j)| \Big\}$$

Calculating the limits for $i_{(2)}$ over which the intersection is non-empty is straightforward. Given those limits, (l_1,l_2), equation (3.14b) becomes

$$\psi_{j_{(1)},j_{(2)}} = \frac{1}{a_{(1),k+1}a_{(2),k+1}} \sum_{i_{(2)}=l_1}^{l_2} \pi_{k,j_{(1)},i_{(2)}}$$

$$\int_{\xi_1(i,j)}^{\xi_2(i,j)} dx_{(1)} \int_{\varsigma_1(x_{(1)},i,j)}^{\varsigma_2(x_{(1)},i,j)} dx_{(2)} \ (2\pi R)^{-\frac{1}{2}} \exp(-\frac{(z_k-x_{(2)})^2}{2R}) \qquad (5.1)$$

The inner integral above can be rewritten as

$$G(z_k-\varsigma_1(x_{(1)},i,j),R) - G(z_k-\varsigma_2(x_{(1)},i,j),R)$$

where $G(x,R)$ is the cumulative distribution function for a Gaussian with mean zero and variance R. We now have

$$\psi_{j_{(1)},j_{(2)}} = \frac{1}{a_{(1),k+1}a_{(2),k+1}} \sum_{i_{(2)}=l_1}^{l_2} \pi_{k,j_{(1)},i_{(2)}}$$

$$\int_{\xi_1(i,j)}^{\xi_2(i,j)} dx_{(1)}(G(z_k-\varsigma_1(x_{(1)},i,j),R) - G(z_k-\varsigma_2(x_{(1)},i,j),R)) \tag{5.2}$$

Now consider equation (3.14c). The system time update density r is given by

$$r_k(x) = \delta(x_{(1)})\ (2\pi Q)^{-\frac{1}{2}}\exp(-\frac{{x_{(2)}}^2}{2Q})$$

where $\delta(x)$ is the Dirac delta or 'impulse' function. Hence we obtain

$$r_i = a_{(1),k+1}a_{(2),k+1} \int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_{(1),1} \int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_{(1),2}\ \delta(a_{(1),k+1}(i_{(1)}+\epsilon_{(1),1}-\epsilon_{(1),2})$$

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_{(2),1} \int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_{(2),2}\ (2\pi Q)^{-\frac{1}{2}}\exp(-\frac{(a_{(2),k+1}(i_{(2)}+\epsilon_{(2),1}-\epsilon_{(2),2}))^2}{2Q})$$

$$= a_{(1),k+1}a_{(2),k+1}\delta_{i_{(1)}} \int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_1 \int_{-\frac{1}{2}}^{\frac{1}{2}} d\epsilon_2\ (2\pi Q)^{-\frac{1}{2}}\exp(-\frac{(a_{(2),k+1}(i_{(2)}+\epsilon_1-\epsilon_2))^2}{2Q}) \tag{5.3}$$

where δ_i is the Kroneker delta. As with the equation for ψ, the above can be rewritten in simpler form in terms of the cumulative distribution function. The equations are now detailed enough to implement in a computer program.

5.3. Analysis of Density Behavior

The first step in analyzing the density behavior is to develop the specific realizations of the equations in the recursion for the problem at hand, as we did in the last section. In doing this, we are forced to begin thinking of the process in terms of operations on entire densities, instead of on scalars, as we might in other schemes. As we do, we often gain a deeper understanding of the mechanics of the density propagation just by visualizing the general behavior. In doing so, it is useful to think in terms of three basic operations: first, the multiplication of the old prediction density and the measurement update that yields the filtering density ϕ; second, the nonlinear transformation due to the system dynamics that gives the intermediate density ψ; and third, the linear convolution of ψ and the input noise density that gives the new prediction density. The first operation tends to trim the density, the second skews it, and the third smears it out again. Let's look at these operations in the context of this problem. The first thing to note is that the measurement density is independent of the parameter $x_{(1)}$. In other words, the density looks like a ridge running parallel to the $x_{(1)}$ axis. For convenience, imagine that the prior density is completely uniform over the grid area. The result of the multiplication operation then would be a duplication of the measurement density, a ridge parallel to the $x_{(1)}$ axis. The second operation rearranges the density according to the system dynamics. In this case, we map the density at a point (a,b) to the point (a,ab). This results in the ridge line of the density running along a straight line through (0,0). The spread of the density in the $x_{(2)}$ direction is also affected, going to zero toward $x_{(1)}=0$ and increasing with increasing $x_{(1)}$. Next, the convolution spreads the density out along the $x_{(2)}$ direction. The result is a somewhat wedge-shaped ridge running at an angle to both axes. Now when we process the next measurement, we again multiply by a ridge parallel to the $x_{(1)}$ axis. It is easy to visualize this and see that the result is a hump shaped density roughly at the intersection of the two original ridges. As the process is repeated, we see that the density will converge in both dimensions.

At this point, it is worth digressing to discuss what we mean by convergence of the density. As we discussed earlier, the density function is interpreted as a measure of the information that we have about some state of nature. The more spread out the density, the less precise our knowledge. Conversely, the more sharply peaked the density is, the more sure we are. Hence we are interested not only in the location of the density but also in its shape; we want it to become increasingly peaked as we process measurements.

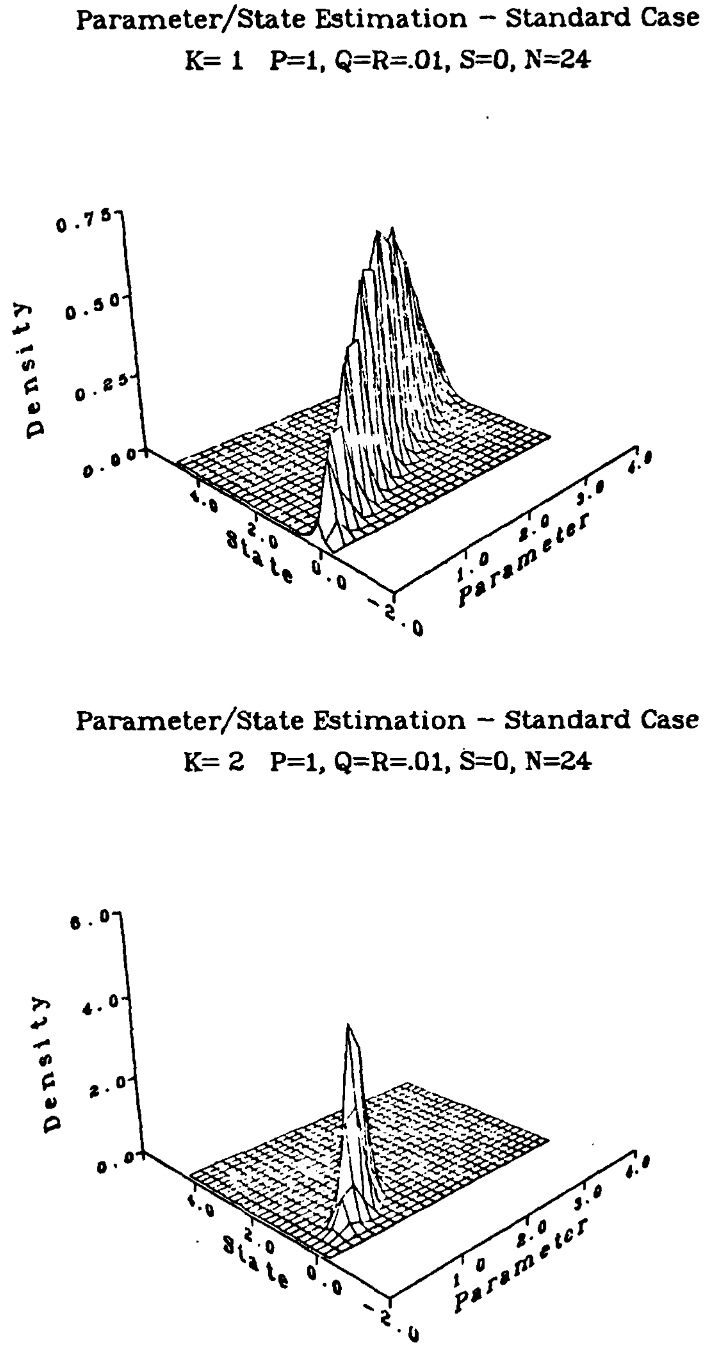
This type of intuitive analysis can provide new insights. For instance, it makes it clear that the crucial operation is the twisting of the density due to the dynamics update. Hence the effectiveness of an estimator rests on its ability to account for this effect. As another example, it points out the desirability of having persistently exciting inputs. Returning to the simplified visualization of multiplying two ridge shaped densities, it is clear that convergence in $x_{(1)}$ depends on the angle between the two ridge lines.

We want to avoid the two being nearly parallel. The angle is determined primarily by the rotation from the dynamics update, and the degree of rotation is in turn determined by how far away from zero the original ridge was. Therefore, we want to avoid operating the system with small state values. To do so requires persistent inputs, particularly for stable systems. Finally, we can assess the impact of the true value of the transition parameter $x_{(1)}$ on the density behavior. Based on the last point, we might think that a larger true $x_{(1)}$ would be easier to identify since it leads to larger state values. But note that the spreading of the density in the dynamics update increases with $x_{(1)}$, and this in turn will degrade the convergence of the density in $x_{(1)}$. It is not clear which effect is dominant, although I suspect that they roughly balance each other. In this instance, we would need to go further than the intuitive analysis for a definitive answer. Note, though, that whether the plant is stable or unstable is not an issue; here there is nothing magic about the stability boundary.

Having developed an intuitive feel for the propagation of the prediction density, let us now look at the density. Actually plotting the density is probably the best way to visualize it, but may not be really practical for greater than two dimensions. In some cases, it may be reasonable to plot two dimensional slices of the density, or two dimensional marginal densities, which can be done using conventional plotting. Alternatively, information such as the first few moments, or location of the peak or peaks can provide a good feel for the shape of the density in multiple dimensions. The entire problem of visualizing multi-dimensional surfaces is little explored, and would be worth pursuing.

For the case at hand, there is no trouble plotting the density. Figure 5.2a shows the prediction density[2] for a low noise case, $Q=R=0.1$. The true parameter value was 1, and the actual initial state was 2. The initial density was Gaussian centered at (2,2). It is obvious that the ridge line lies along a line as the earlier analysis predicted. Figure 5.2b shows the density at the next time step. As you would expect in a low noise case, there has been tremendous convergence. The convergence is also helped by the large initial state value, which resulted in a relatively large amount of rotation on the first iteration. The rotation of the ridge line is still discernible in this plot.

[2] Due to the graphics software used to produce the plots in this chapter, the densities are displayed as continuous functions rather than the piecewise constant approximations they really are. Keep in mind the actual functional form.



Parameter/State Estimation – Standard Case
K= 1 P=1, Q=R=.01, S=0, N=24

Parameter/State Estimation – Standard Case
K= 2 P=1, Q=R=.01, S=0, N=24

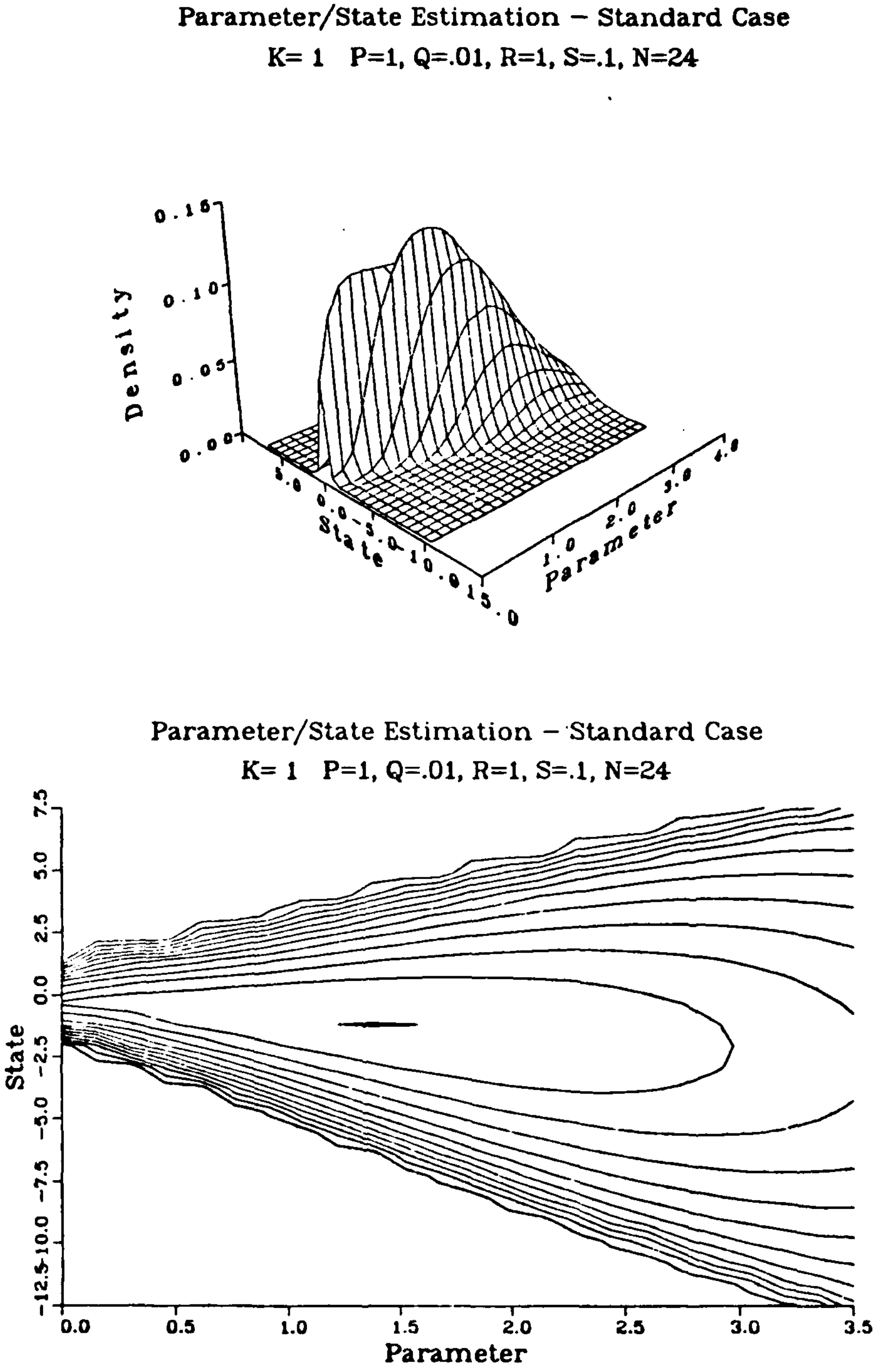
Figure 5.2 - Parameter/state prediction density for $Q = R = 0.01$

Figure 5.3 shows the prediction density for another case. Here a large measurement noise variance, $R = 1$, was chosen to get relatively slow convergence of the density. For this case, the parameter was 1, the initial state -0.5, and the initial density Gaussian centered at (1,0). Figure 5.3a is a mesh plot showing the density as a three dimensional surface. Although it is not as obvious as in the last plot, it is still apparent that the ridge line runs at an angle as expected. The shallow angle is to be expected due to the small initial state. Figure 5.3b shows a contour plot of the log of the same density. In this plot the increasing spread of the density as $x_{(1)}$ increases is clear. Figure 5.4 shows the prediction density for the same run two time steps later. Despite the high noise, there has been considerable convergence. The ridge line skew is less noticeable on the mesh plot of 5.4a than the last time. Looking at the contour plot of 5.4b, though, we see that the ridge line has developed a definite curve. This is easily explained by thinking again of the intuitive analysis. Since the measurement variance is large, and the input variance small, for a first approximation we can neglect the first and third steps, and the influence of the known input. Now start with a ridge running parallel to the $x_{(1)}$ axis. After the first dynamics update the ridge follows a straight line through the origin. After the second, it follows a parabolic arc, since we mapped (a,b) into (a,ab) then into (a,a^2b). After the third, it will follow a cubic. Including the neglected effects disrupts this nice progression, but the basic mechanism is there.

Any number of parametric studies are possible at this point. It is possible to look at the effects of varying the noise variances, the initial state and parameter values, or the initial state and parameter estimates. We can explore different noise and initial state densities, such as uniform instead of Gaussian, and look at their effects. The possibilities are virtually limitless. They are, however, not very meaningful without a specific application in mind. Therefore, we will not pursue this, and instead encourage users to apply the technique to their particular problem.

5.4. Analysis for Other Unknown Parameters

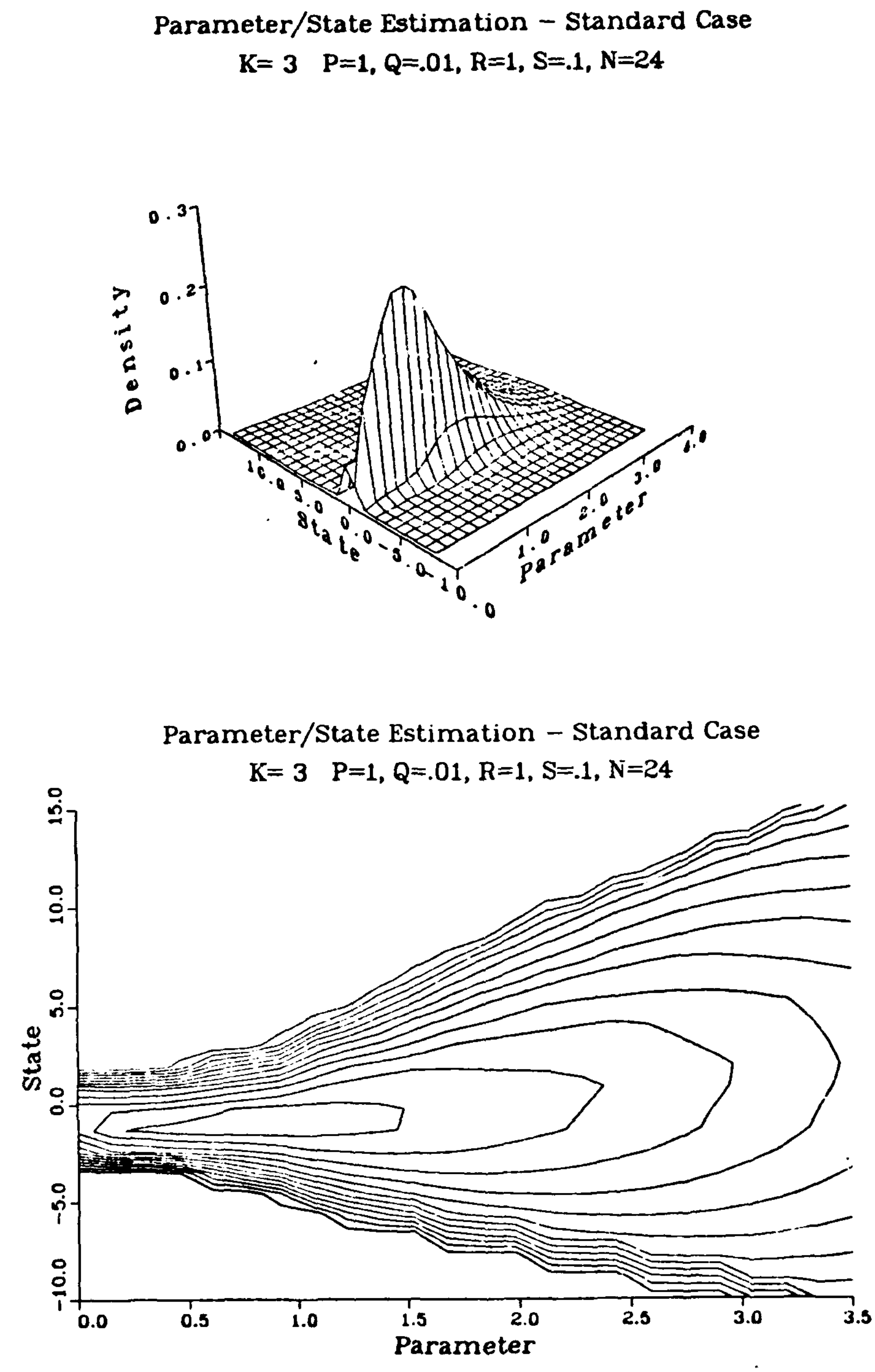
Having gone through the analysis above for the case of an unknown transition parameter, it is quite easy to extend the intuitive analysis to other situations. In this section we will consider four other possible unknown parameters, and do a brief parallel analysis of the estimation problem for each.



Parameter/State Estimation – Standard Case
K= 1 P=1, Q=.01, R=1, S=.1, N=24

Parameter/State Estimation – Standard Case
K= 1 P=1, Q=.01, R=1, S=.1, N=24

Figure 5.3 - Parameter/state prediction density for Q=0.01, R=1, k=1.



Parameter/State Estimation – Standard Case
K= 3 P=1, Q=.01, R=1, S=.1, N=24

Parameter/State Estimation – Standard Case
K= 3 P=1, Q=.01, R=1, S=.1, N=24

Figure 5.4 - Parameter/state prediction density for $Q=0.01$, $R=1$, $k=3$.

First, consider the case of an unknown scaling on the input. This is modeled by

$$x_{k+1} = Fx_k + Eu_k + w_k$$

$$z_k = x_k + v_k$$

where the parameter E is unknown. The nonlinear state model is

$$x_{k+1} = \begin{bmatrix} x_{(1),k+1} \\ x_{(2),k+1} \end{bmatrix} = \begin{bmatrix} x_{(1),k} \\ Fx_{(2),k} + x_{(1),k}u_k \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} = f(x_k, u_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$z_k = x_{(2),k} + v_k = h(x_k\ ,\ v_k)$$

The grid will be defined as in the previous section. The inverse image of a grid cell is now given by

$$\mathrm{H}_j = \Big\{ x_{(1)} \in \big[a_{(1),k+1}(j_{(1)} - \frac{1}{2}) + b_{(1),k+1}\ ,\ a_{(1),k+1}(j_{(1)} + \frac{1}{2}) + b_{(1),k+1}\big]\ ;$$

$$x_{(2)} \in \Big[\frac{a_{(2),k+1}(j_{(2)} - \frac{1}{2}) + b_{(2),k+1} - x_{(1)}u_k}{F}\ ,\ \frac{a_{(2),k+1}(j_{(2)} + \frac{1}{2}) + b_{(2),k+1} - x_{(1)}u_k}{F}\Big]\Big\}$$

so the inverse image is a parallelogram. This obviously modifies the definitions of the limit functions ς and ξ in (5.1) and (5.2) but does not otherwise affect the equations.

The density behavior for this case is somewhat similar to that for the first case. Again imagine a uniform prior. As before, the measurement density is a straight ridge parallel to the $x_{(1)}$ axis and located at $x_{(2)}=z$, and it is reproduced by the first measurement update. The system dynamics map the point (a,b) into $(a, Fa+bu)$, so the dynamics update results in a ridge running along the line $x_{(2)}=Fz+x_{(1)}u$. Note that the spread of the density in the $x_{(2)}$ direction does not depend on $x_{(1)}$. The convolution with the input noise density spreads the density out in the $x_{(2)}$ direction, resulting in a straight ridge with its $x_{(2)}$ spread independent of $x_{(1)}$. Note the contrast with the last case, where the $x_{(2)}$ spread increases with $x_{(1)}$. Processing the next measurement results in hump at the intersection of the two densities. As before, the degree of convergence depends on the angle between the two ridge lines, but in this case, the angle depends directly on the input, not on the state. Zero input results in no convergence along $x_{(1)}$, as would be expected. In the previous case we wanted to have inputs to keep the system state from getting small; here we must have inputs to get any convergence at all. Also note that the spread of the densities does not depend on $x_{(1)}$, so there is no variation in convergence with true value of the parameter.

Next consider an unknown scaling in the output equation. The model is

$$x_{k+1} = Fx_k + u_k + w_k$$

$$z_k = Hx_k + v_k$$

where the parameter H is unknown. The nonlinear state model is

$$x_{k+1} = \begin{bmatrix} x_{(1),k+1} \\ x_{(2),k+1} \end{bmatrix} = \begin{bmatrix} x_{(1),k} \\ Fx_{(2),k} + u_k \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} = f(x_k, u_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$z_k = x_{(1),k} x_{(2),k} + v_k = h(x_k\ ,\ v_k)$$

Note that this model has linear dynamics. Using the same grid as before, a simple computation yields the inverse image set

$$\mathbf{H}_j = \Big\{ x_{(1)} \in |a_{(1),k+1}(j_{(1)} - \frac{1}{2}) + b_{(1),k+1}\ ,\ a_{(1),k+1}(j_{(1)} + \frac{1}{2}) + b_{(1),k+1}|\ ;$$

$$x_{(2)} \in |\frac{a_{(2),k+1}(j_{(2)} - \frac{1}{2}) + b_{(2),k+1} - u_k}{F}\ ,\ \frac{a_{(2),k+1}(j_{(2)} + \frac{1}{2}) + b_{(2),k+1} - u_k}{F}| \Big\}$$

which is just a rectangle. Again, this leads to modification of the limit functions in (5.1). More importantly, we must also change μ because of the change in the measurement equation. Equation (5.1) becomes

$$\psi_{j_{(1)},j_{(2)}} = \frac{1}{\alpha_{k+1}} \sum_{i_{(2)}=l_1}^{l_2} \pi_{k,j_{(1)},i_{(2)}}$$

$$\int_{\xi_1(i,j)}^{\xi_2(i,j)} dx_{(1)} \int_{\zeta_1(x_{(1)},i,j)}^{\zeta_2(x_{(1)},i,j)} dx_{(2)}\ (2\pi R)^{-\frac{1}{2}} \exp(-\frac{(z_k - x_{(1)}x_{(2)})^2}{2R})$$

The density behavior for this case is quite different than the first two. Again, we will start with a uniform prior. The measurement density defined above is a somewhat banana-shaped ridge, with the ridge running along the hyperbola $x_{(2)} = z/x_{(1)}$. The isocontours follow hyperbolas in all four quadrants. The dynamics update scales the density and shifts it in the $x_{(2)}$ direction, and then the convolution with the input noise density spreads it out in the $x_{(2)}$ direction independent of $x_{(1)}$. The result is essentially a shifted version of the measurement density. The next measurement update can now be roughly visualized as the multiplication of two of these densities offset with respect to each other. If there is little offset, there will be relatively little convergence. On the other hand, if there is a large offset, the isocontours of the densities will cross at large angles, resulting in convergence in both dimensions. In contrast to the first two cases, it is the nonlinear nature of the measurement update that is important, rather than the dynamics. A few moments sketching the densities for different situations shows that the best situation is one with large oscillating known inputs. At the extreme, if $u=0$, then there is no convergence in $x_{(1)}$. There is also a fairly strong dependence on the true value of the parameter. Larger true $x_{(1)}$ tends to amplify the effect of the inputs, and so the density will converge faster.

Now look at unknown scaling on the measurement noise. The model for this is

$$x_{k+1} = Fx_k + u_k + w_k$$

$$z_k = x_k + Dv_k$$

where the parameter D is unknown. The nonlinear state model is

$$x_{k+1} = \begin{bmatrix} x_{(1),k+1} \\ x_{(2),k+1} \end{bmatrix} = \begin{bmatrix} x_{(1),k} \\ Fx_{(2),k} + u_k \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} = f(x_k,u_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$z_k = x_{(2),k} + x_{(1),k}v_k = h(x_k\,,\,v_k)$$

This model has the same dynamics as the last, so the grid and inverse image set are the same. The measurement noise density μ is changed, however, making (5.1)

$$\psi_{j_{(1)},j_{(2)}} = \frac{1}{\alpha_{k+1}} \sum_{i_{(2)}=l_1}^{l_2} \pi_{k,j_{(1)},i_{(2)}}$$

$$\int_{\xi_1(i,j)}^{\xi_2(i,j)} dx_{(1)} \int_{\varsigma_1(x_{(1)},i,j)}^{\varsigma_2(x_{(1)},i,j)} dx_{(2)}\,(2\pi R)^{-\frac{1}{2}} \exp(-\frac{(z_k - x_{(2)})^2}{2Rx_{(1)}{}^2})$$

The measurement density for this case is a fan-shaped ridge, with the apex at $x=(0,z)$, and the ridge line running along $x_{(2)}=z$. Starting with the ubiquitous uniform prior, we get a copy of this density after the first measurement update. The system dynamics map (a,b) into $(a,Fb+u)$, resulting in a scaling and shift in the $x_{(2)}$ direction. The convolution then spreads the density out in the $x_{(2)}$ direction. The resulting density is still fan-shaped. Now when the next measurement is processed, we multiply by another fan-shaped ridge with a parallel ridge line. The result is yet another fan-shaped ridge, with no convergence in the $x_{(1)}$ direction. To take an extreme example, suppose that the true value of $x_{(1)}$ is 0. Then at each measurement update the ridge lines of the densities would line up. It is easy to see that the density will not converge in the $x_{(1)}$

direction. For non-zero $x_{(1)}$, the ridge lines will be offset. This will tend to reduce the density for smaller values of $x_{(1)}$, but will not affect the higher values. In this case, the parameter is not identifiable, and this is easily shown by considering the actual density behavior as we have done.

5.5. Comparisons of Point and Bayes' Estimates

Let's now look at the final use mentioned in Section 5.1. We will use for our candidate point estimator a minimum variance estimator derived by Liang and Christensen [39]. Liang has published the results of several simulation studies comparing the filter (referred to hereafter as the MVF) to the EKF for a parameter identification problem [40]. In his paper Liang points out the theoretical difficulties of comparing suboptimal nonlinear filters, and concludes that only detailed numerical simulations can provide the basis for meaningful comparisons. This is undoubtedly true, but Liang mistakenly discounts the possibility of comparing the suboptimal filters to the true optimal solution. The true optimal solution is available for this case as the mean of the conditional density computed via the algorithm derived in this paper (for simplicity, we will refer to the conditional mean for this case as the Bayes' estimate). What we will do, then, is extend the analysis of [40] to include the true optimum.

In his paper, Liang explores the behavior of the MVF for three slightly different discrete-time system models. For this work we will concentrate on the second of the systems, which is described by the model

$$x_{k+1} = \begin{bmatrix} x_{(1),k+1} \\ x_{(2),k+1} \end{bmatrix} = \begin{bmatrix} x_{(1),k} \\ x_{(1),k}\, x_{(2),k} \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} = f(x_k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

$$z_k = x_{(1),k}\, x_{(2),k} + v_k = h(x_k\ ,\ v_k)$$

Note that the unknown parameter $x_{(1)}$ now appears as a scaling factor in the observation equation as well as in the dynamics, and the model does not include a known input. This model does not seem to have a clear physical interpretation, and may have been chosen simply because the EKF does not handle it well.

The equations developed in Section 5.2 can be applied almost directly to this model. The main change is in the equations containing μ, which now is given by

$$\mu(x,z_k) = (2\pi R)^{-\frac{1}{2}} \exp\left(-\frac{(z_k - x_{(1)}x_{(2)})^2}{2R}\right)$$

This changes (5.2) to

$$\psi_{j_{(1)},j_{(2)}} = \frac{1}{a_{(1),k+1}a_{(2),k+1}} \sum_{i_{(2)}=l_1}^{l_2} \pi_{k,j_{(1)},i_{(2)}}$$

$$\int_{\xi_1(i,j)}^{\xi_2(i,j)} dx_{(1)} \frac{(G(z_k - x_{(1)}\varsigma_1(x_{(1)},i,j),R) - G(z_k - x_{(1)}\varsigma_2(x_{(1)},i,j),R))}{x_{(1)}}$$

Except for the removal of the input u_k, the remaining equations in the section do not change.

For the above model, Liang looks at eight different combinations of initial conditions and noise variances. We will consider four of those cases, the ones for which the performance of the MVF and the EKF differ. For all cases, the true parameter value $x_{(1)}$ is 1.0, the initial value of the state $x_{(2)}$ is chosen randomly from a zero-mean Gaussian distribution, and the initial variances for the state and parameter are 1.0. The initial density for the Bayes' algorithm is bivariate Gaussian, centered at the initial estimate. The results for each case are based on the average of 25 independent Monte Carlo runs of the first 30 time steps (one observation per time step).

The first two cases, shown in Figures 5.5 and 5.6, are for small noise variances, $Q = R = 0.01$. Figure 5.5 shows the results for an initial parameter estimate of 2.0. For this case, Liang showed that the EKF tended to diverge, while the MVF converged to the true parameter value after a transient phase. The results here show the MVF converging very slowly, if at all. The Bayes' estimate, though, quickly converges to the true parameter value. The Bayes estimate of the state is also markedly superior to the MVF estimate. Clearly, there is a wide margin between the performance of the suboptimal filter and the true optimal. Figure 5.6 shows the result of making the initial parameter estimate 0.1. Liang actually used an initial parameter estimate of 0 for this second case, which causes the EKF to not respond at all, and adversely affects the transient response of the MVF. Using 0.1 for the initial parameter estimate seems more reasonable, and can only improve the MVF response. Liang's results show the MVF converging on average to a parameter estimate of approximately 0.6. Figure 5.6 shows essentially that behavior for the MVF, perhaps improved slightly since it appears to be moving slowly toward the true value. The Bayes' estimate, on the other hand, converges quickly to near the true value. Note that the Bayes estimate jumps from the initial estimate to near the true value after the first observation. This rapid convergence

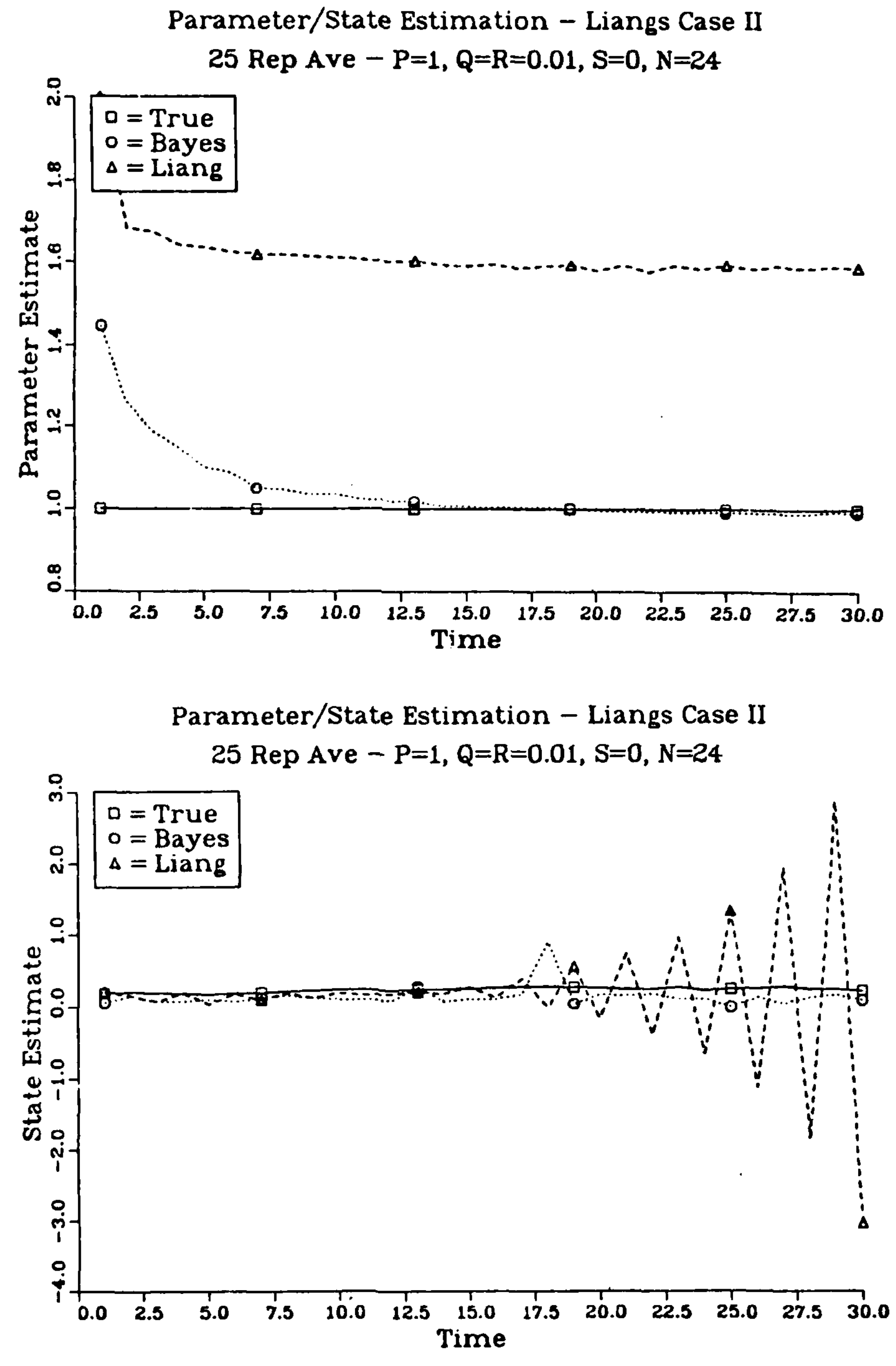
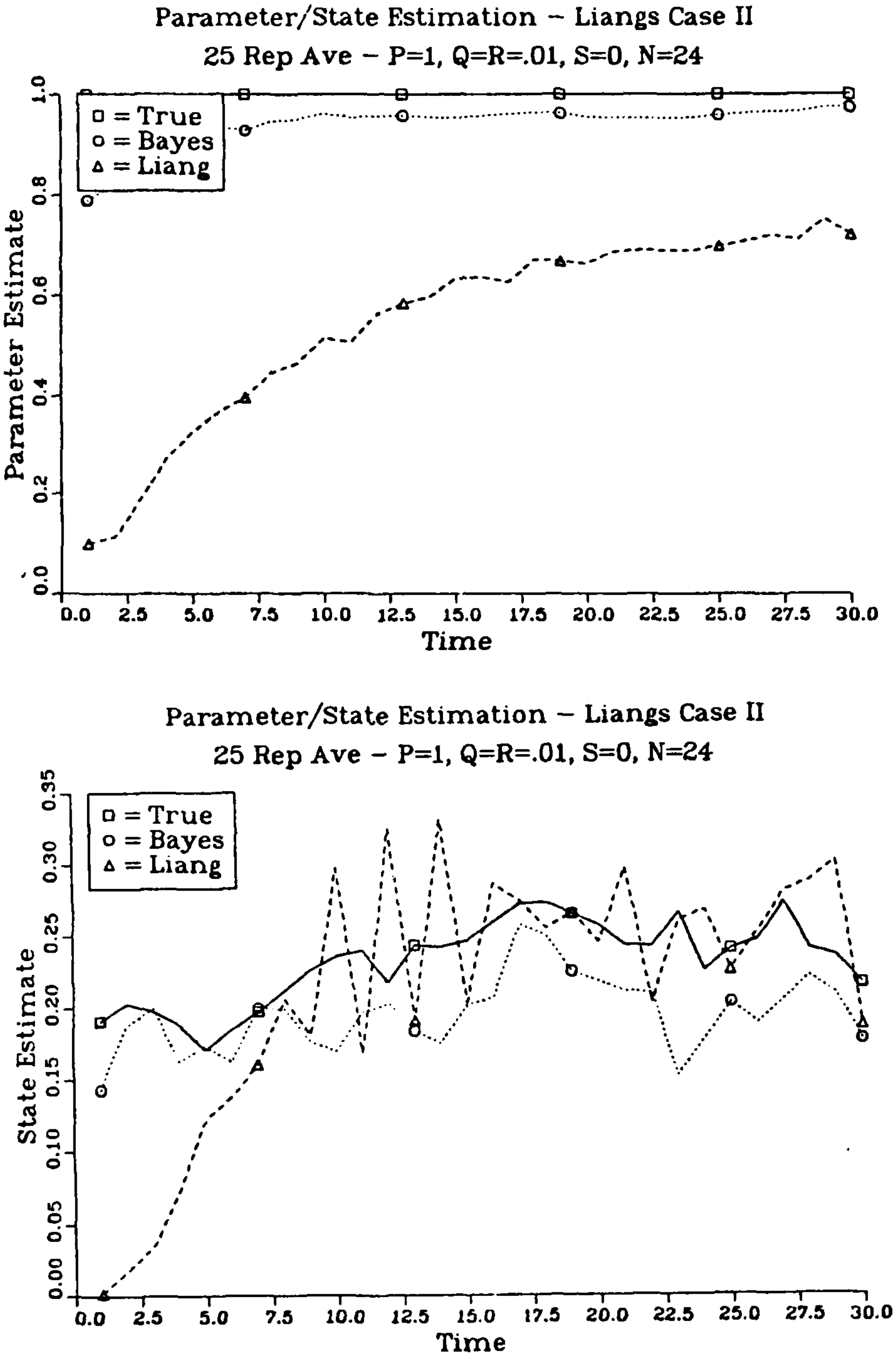


Figure 5.5 - Averaged parameter/state estimates for Liang's Case II, $Q = R = 0.01$, initial parameter estimate 2.0.


Parameter/State Estimation – Liangs Case II
25 Rep Ave – P=1, Q=R=.01, S=0, N=24

Parameter/State Estimation – Liangs Case II
25 Rep Ave – P=1, Q=R=.01, S=0, N=24

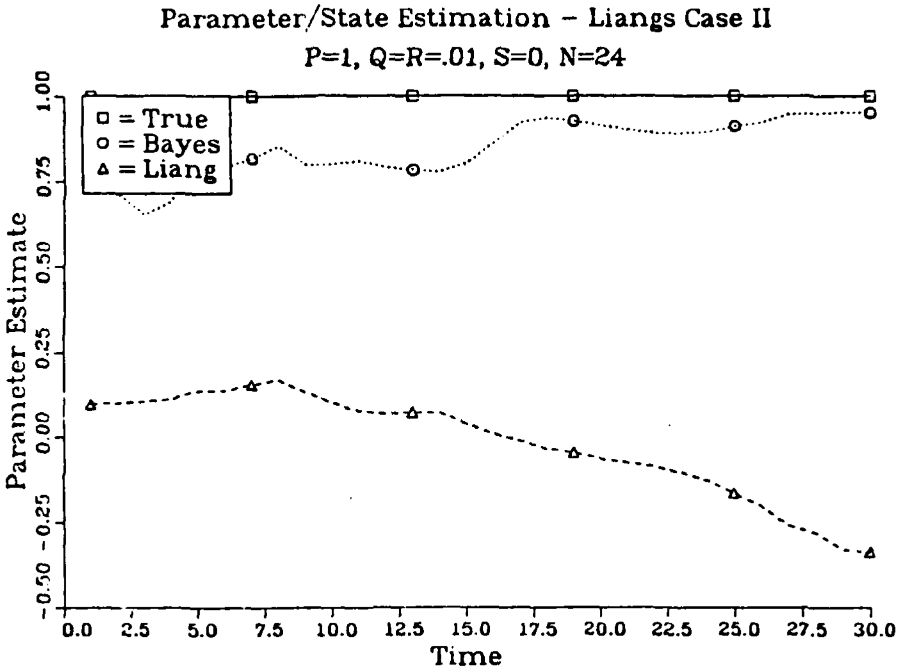
Figure 5.6 - Averaged parameter/state estimates for Liang's Case II, $Q = R = 0.01$, initial parameter estimate 0.1.

appears to be typical of the low noise cases, as would be expected. Even though the average behavior of the MVF is not too bad, it is rather inconsistent. As an example, the results shown in Figure 5.7 turned up on one of the test runs. This figure shows the filter performance for a single Monte Carlo run. Here the MVF actually diverges, while the optimal estimate performs essentially as the average shown in Figure 5.6. Throughout the individual runs that were reviewed, the Bayes' estimate performed very consistently.

In addition to being concerned with the convergence of the conditional mean to the true value, we are interested in the convergence of the density as discussed in Section 5.3. Even if the mean of the density were close to the true value, we would have significantly less confidence in the estimate if the density were spread out than if it were a sharp spike. For an example, consider the model with the noise levels and initial conditions used for Figure 5.5. Figure 5.8 shows the conditional density at time steps 1, 4, and 8 for a single realization. As we would expect for a low noise case, there is very rapid convergence in the state after a single measurement. A more measurements are processed, the density converges in the parameter direction also. In general, the density is clearly converging as we would like it to. This behavior was also seen consistently in the other individual runs that were reviewed.

The next two cases show the results of larger noise variances, $Q=R=1.0$, with the two initial conditions used above. Figure 5.9 results from the initial parameter estimate 2.0. For this case, Liang shows both the MVF and the EKF converging fairly rapidly to the true parameter value, with the EKF slightly superior. Here the MVF and the Bayes' estimates show comparable performance, although the Bayes' estimate tends to display somewhat smoother convergence. The fairly large oscillations in the MVF estimates (particularly the state estimate) seen in this figure appear commonly in the simulations. Figure 5.10 shows the result for an initial estimate of 0.1. Liang's results for this case show the MVF converging to an estimate of about 0.5, then drifting very slowly toward the true value. The results here show that the Bayes' estimate converges more quickly and closer to the true value than the MVF. Again, note that the Bayes' estimate is smoother than the MVF.

These results allow us to generalize Liang's conclusions as follows.



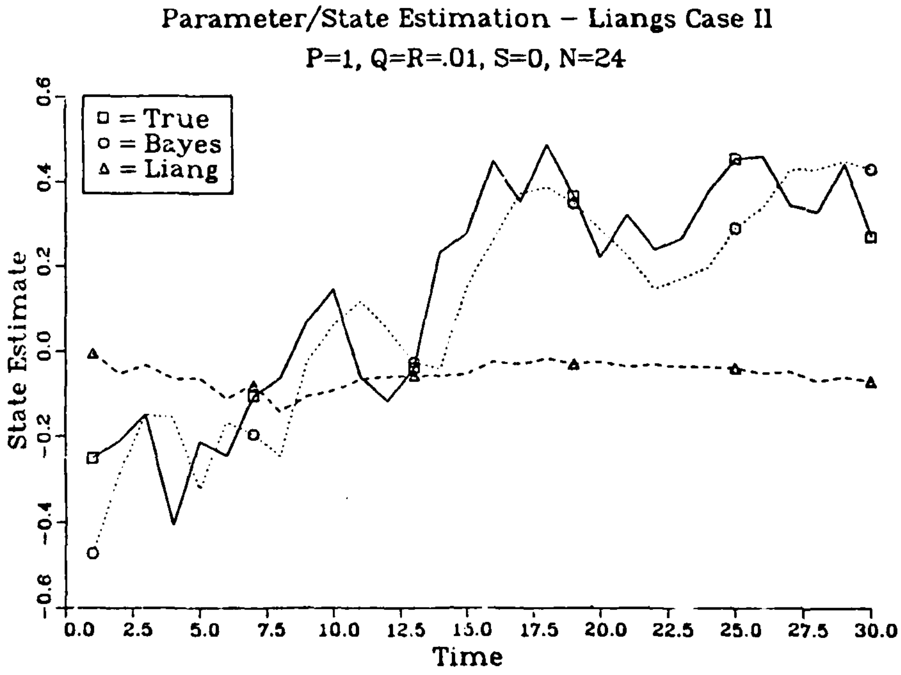


Figure 5.7 - Sample run parameter/state estimates for Liang's Case II, $Q = R = 0.01$, initial parameter estimate 0.1.

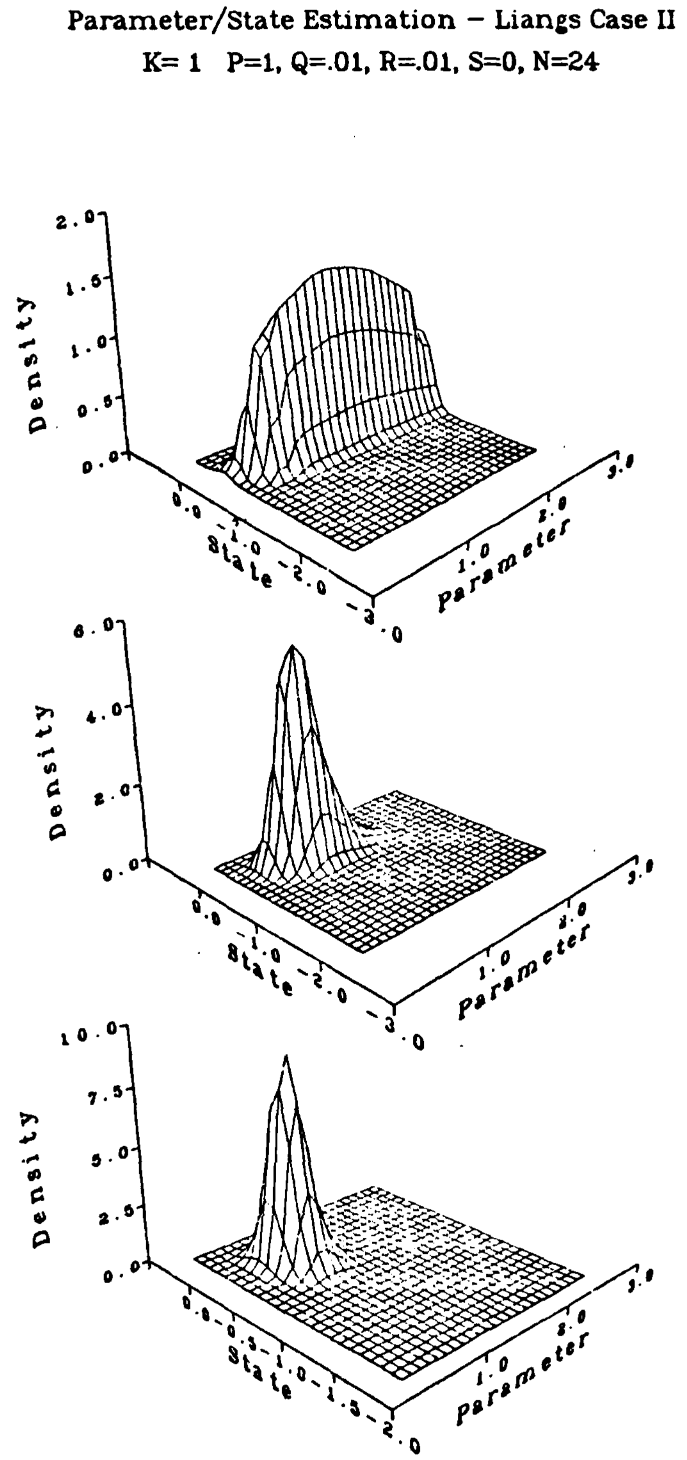
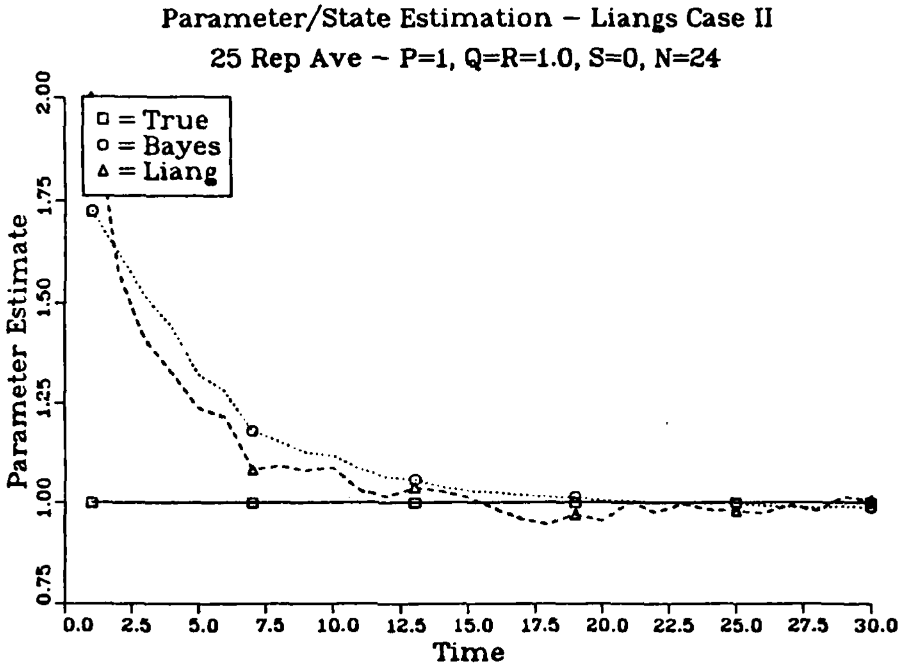


Figure 5.8 - Representative densities for time steps 1, 4, and 8, for Liang's Case II, $Q = R = 0.01$, initial parameter estimate 2.0.



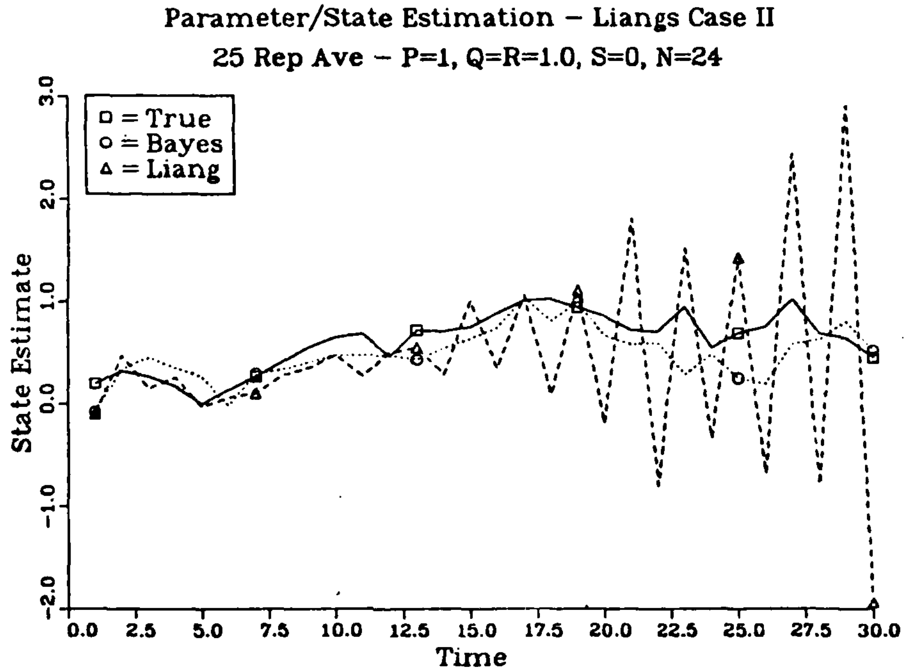
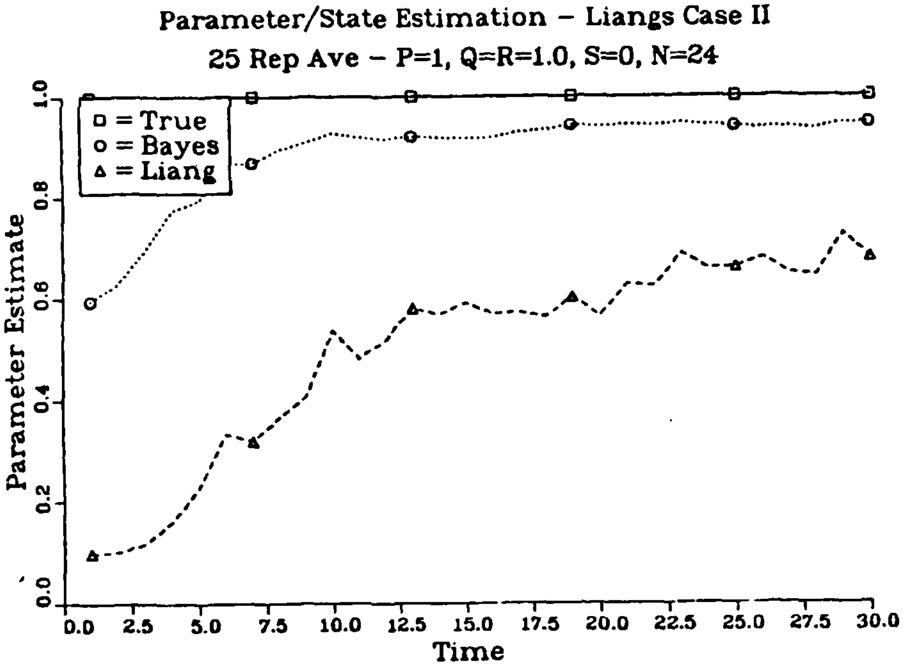


Figure 5.9 - Averaged parameter/state estimates for Liang's Case II, $Q=R=1.0$, initial parameter estimate 2.0



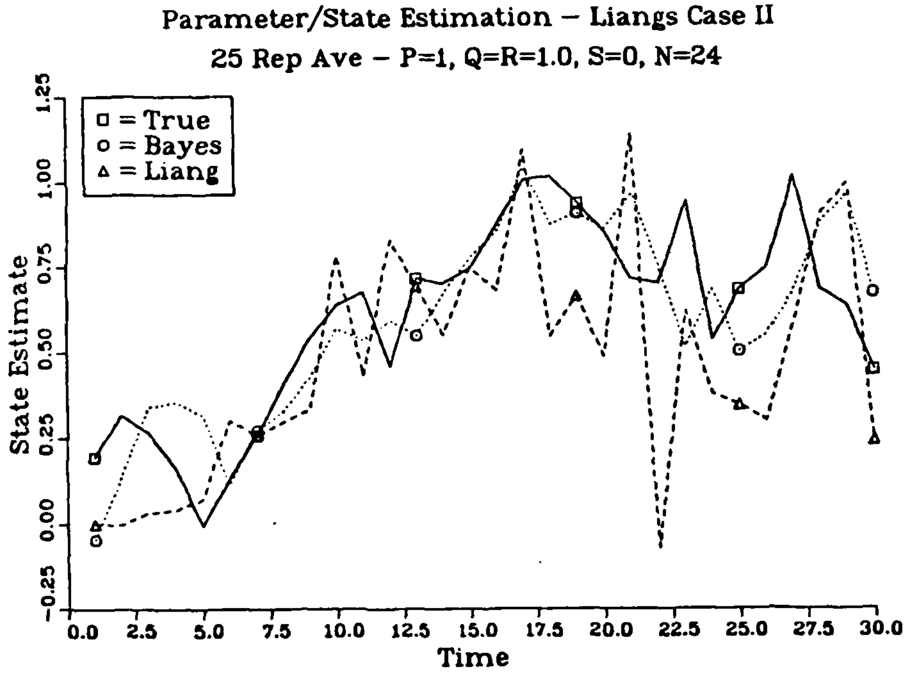
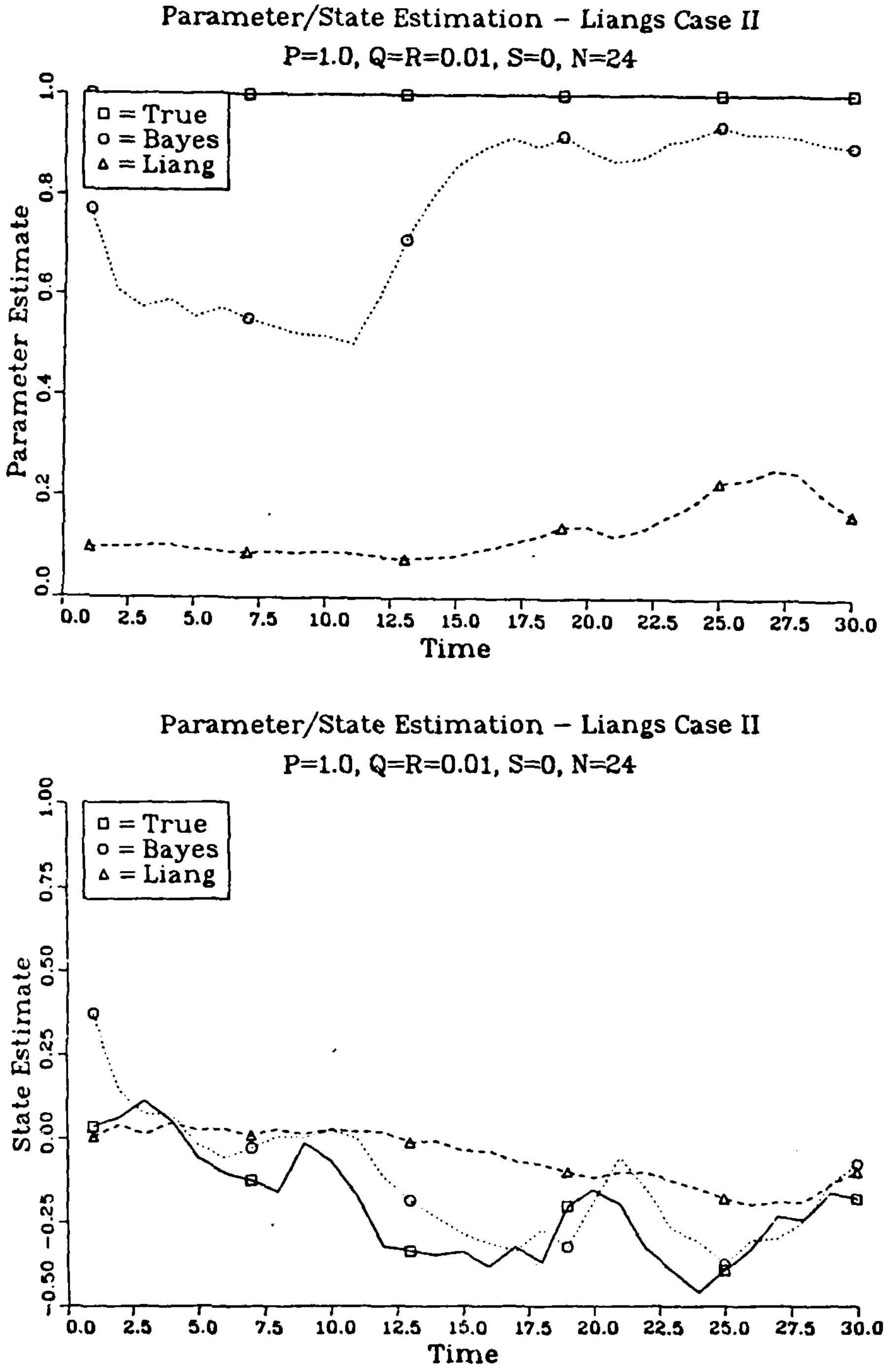


Figure 5.10 - Averaged parameter/state estimates for Liang's Case II, $Q = R = 1.0$, initial parameter estimate 0.1.

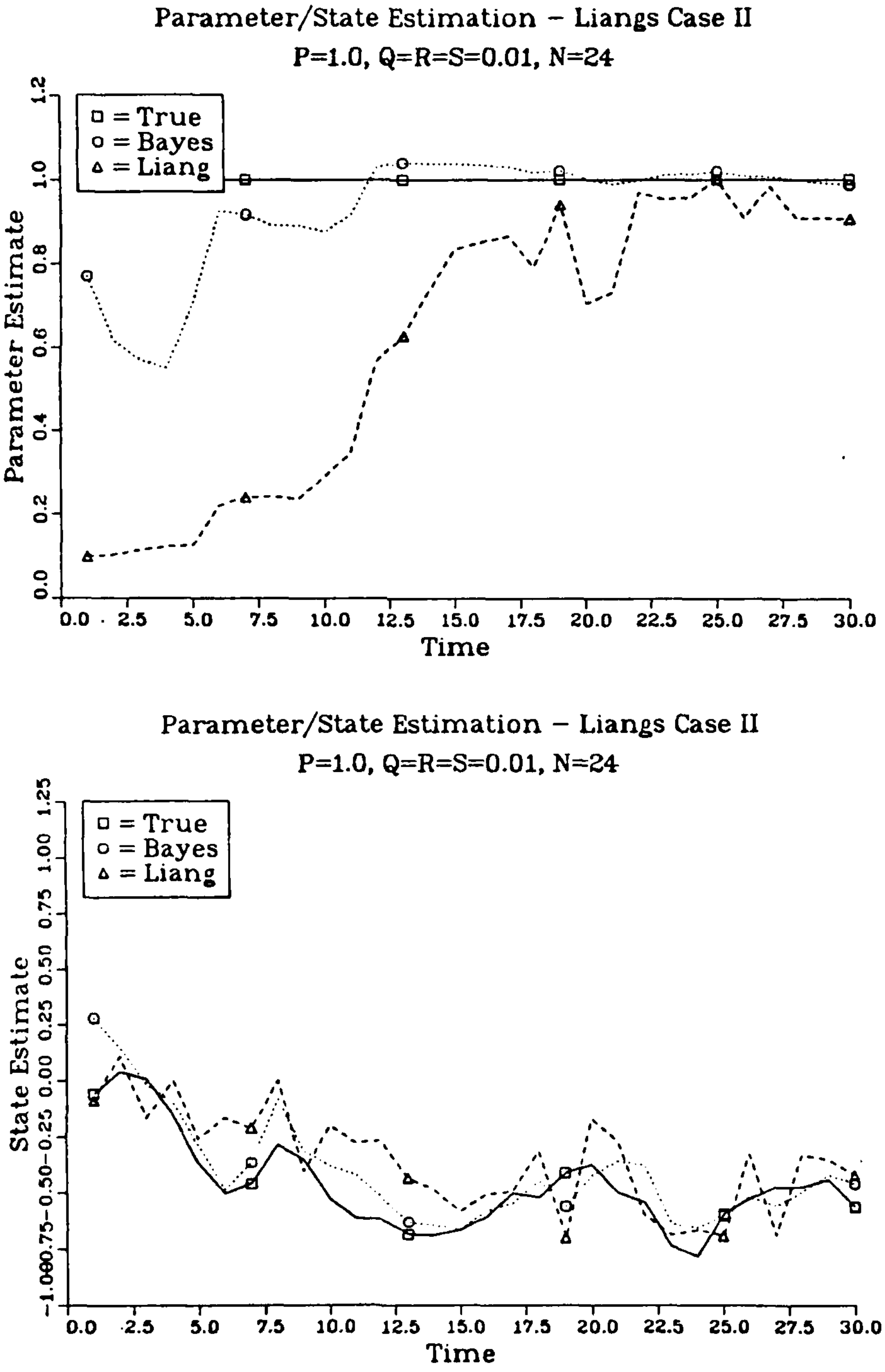
1. When the level of noise inputs is large enough to mask the effects of nonlinearities, point estimators such as the MVF and EKF tend to perform fairly close to the optimum, although they may be quite sensitive to initial conditions.
2. Although the MVF is claimed to be 'far superior' to the EKF when the noises are small (so that the effects of the nonlinearities are not negligible), it is still far from the optimal. This indicates that the MVF still fails to capture important features of the posterior densities.

As a final point in this section, let's look at the effect of including a known input sequence. As noted above, Liang did not include this factor in his analysis. It is well-known, however, that the input sequence can have a large impact on the performance of the estimator. Hence, we would expect based on theoretical considerations that including it would improve the performance of the estimators. Figure 5.11 shows a simulation result for the second case considered above, without an input sequence. Figure 5.12 shows the result with u_k a zero-mean Gaussian sequence with variance 0.01, so that the input signal to noise ratio is 1. The same noise sequence realizations were used in both simulations, so the differences are due solely to the inclusion of the known input. The improvement in the performance of both estimators is quite remarkable. The MVF estimate goes from essentially nonconvergent to convergent. The Bayes' estimate converges much faster and much closer to the true value. Both state estimates show the same type of improvement. Of course, this is not the optimal input sequence, so it may be possible to get even more improvement. Even so, this is a good example of the performance improvement possible when an input sequence is available.


Parameter/State Estimation – Liangs Case II
P=1.0, Q=R=0.01, S=0, N=24

Parameter/State Estimation – Liangs Case II
P=1.0, Q=R=0.01, S=0, N=24

Figure 5.11 - Parameter/state estimates for Liang's Case II, without known input sequence.



Parameter/State Estimation – Liangs Case II
P=1.0, Q=R=S=0.01, N=24

Parameter/State Estimation – Liangs Case II
P=1.0, Q=R=S=0.01, N=24

Figure 5.12 - Parameter/state estimates for Liang's Case II, with known input sequence, signal to noise ratio 1.0.

6. APPLICATION TO BEARINGS-ONLY TRACKING

6.1. Problem Definition

Lets turn now to a more specific problem. The bearings-only tracking problem is a very important practical problem that arises in a number of situations. The prototypical situation is a sea-based sonar tracking problem, where a moving ship (the receiver) is listening to the acoustic output of another moving ship (the target or source). The objective is to estimate the speed and location of the source based on the noisy measurements of the bearing of the sound source relative to the receiver. The geometry of this problem is shown in Figure 6.1. The source is located at the point $P_s = (p_{s,x}, p_{s,y})$, with velocity V_s. The receiver is located at P_r, and has velocity V_r. The angle Θ is the bearing of the source relative to the receiver.

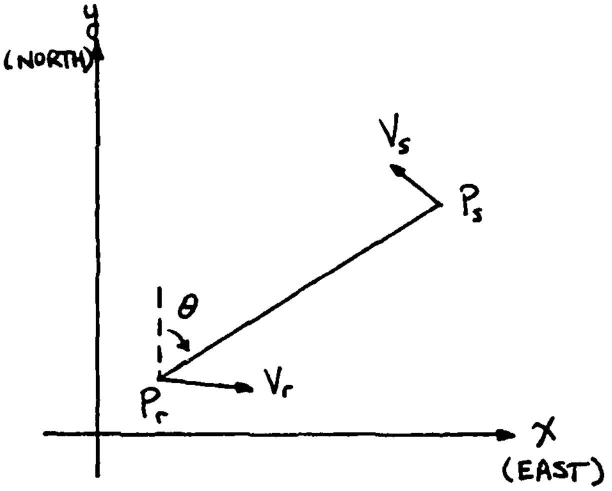


Figure 6.1 - *Bearings-only tracking geometry.*

It is important to point out that the system is only observable when the target has constant velocity and the receiver maneuvers satisfy certain constraints [41]. In particular, the system is not observable if the receiver follows a constant velocity path. This makes at least some sort of receiver maneuver mandatory if we wish to completely characterize the target position and velocity.

There are two other particular points to made about this problem. First, the problem is intrinsically nonlinear, and hence is not addressable by conventional linear estimation techniques. Second, in realistic situations, there are relatively few measurements available due to the usually short contact times. Hence the transient performance of an estimator is of critical importance. For these reasons, we would expect the Bayesian approach to be desirable.

A number of different point estimation approaches have been tried. The extended Kalman filter is the first obvious choice. It has reasonable performance in many cases, but is prone to premature covariance collapse and solution divergence, due the interaction and feedback of filter errors [42]. Pseudo-linear formulations have been suggested to avoid this difficulty, but the resulting filters produce biased estimates when used with noisy data [42]. Petridis has developed a different approach based on a partition algorithm [43]. This algorithm seems to avoid the EKF's divergence problems, but since it consists of a number of filters operating in parallel, it incurs a large computational cost. Yet another approach is to work in other than the 'natural' cartesian coordinates. Aidala and Hammel [44] reformulated the problem in terms of modified polar (MP) coordinates and then applied the EKF. This resulted in a decoupling of certain terms in the covariance update, so that covariance collapse would not occur. The problem has also been approached from the Bayesian viewpoint, by calculating a representation of the posterior density. Bucy [45] used the point mass approximation to look at the density propagation for an aircraft based problem. Sorenson [23] used his p-vector approach in exploring a ship based problem, and used the peak of the approximate posterior density to define the maximum a posteriori estimator.

In this section we will apply the algorithm developed earlier to the bearings-only tracking problem. First, we will analyze the density propagation from a qualitative point of view first to try to get some additional insight into the problem. Then we will compare the performance of the conditional mean of the estimated density to the MP filter for several example scenarios culled from the literature.

6.2. Algorithm Development

The first step is to define the system equations which we will use. As noted above, there are several possible formulations, each leading to different equations. The choice of reference frame is extremely important in designing point estimators and has

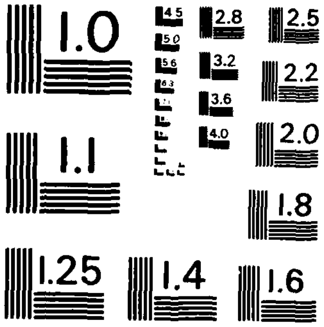
AD-A160 836 THE BAYESIAN APPROACH TO RECURSIVE STATE ESTIMATION: IMPLEMENTATION AND APPLICATION(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH S C KRAMER 1985 2/2

UNCLASSIFIED AFIT/CI/NR-85-139D F/G 12/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

been demonstrated to greatly affect the ultimate filter performance. This is because the errors from the approximations used in the filters depends heavily on the particular form of the system equations. These choices are rather less important for the Bayesian formulation, since there are no approximations being made. It is of some importance in applying the algorithm of this paper, since the choice of state will affect the computational complexity of the algorithm.

The most commonly used, and conceptually simplest, form is based on cartesian coordinates. The main advantages of this formulation are that it results in linear dynamics and that it is easy to visualize. The state vecto. : defined as the position and velocity of the source at time step k, so[1]

$$x_k = [\,p_{s,x}(k\delta) \;\; p_{s,y}(k\delta) \;\; v_{s,x} \;\; v_{s,y}\,]^T$$

where δ is the time between steps. Note the velocity does not depend on time, since we are assuming a constant velocity source. The system equations are then

$$x_{k+1} = \begin{bmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k = Fx_k$$

$$z_k = \tan^{-1}\left(\frac{x_{(1),k} - p_{r,x}(k\delta)}{x_{(2),k} - p_{r,y}(k\delta)} \right) + \eta_k = h_k(x_k, \eta_k)$$

where $(p_{r,x}(k\delta), p_{r,y}(k\delta))$ is the position of the receiver at time step k, that is, $t = k\delta$. The observation noise η_k is assumed to be zero-mean white Gaussian noise with variance R. Note that there is no noise term in the system dynamics. This is for consistency with previous work, and is not a necessary assumption for the Bayesian approach. Because there is no plant input noise, the above can be rewritten as an equivalent identification problem in terms of the initial position of the source. This results in the alternative formulation

$$x' = [\,p_{s,x}(0) \;\; p_{s,y}(0) \;\; v_{s,x} \;\; v_{s,y}\,]^T$$

$$z_k = \tan^{-1}\left(\frac{x'_{(1)} + k\delta x'_{(3)} - p_{r,x}(k\delta)}{x'_{(2)} + k\delta x'_{(4)} - p_{r,y}(k\delta)} \right) + \eta_k = h_k(x', \eta_k)$$

[1] I apologize for the clash between the convention of using x as the state variable, and that of using x as the horizontal (east) coordinate. The context should make it clear which meaning is intended.

where $p_{r,x}$, $p_{r,y}$, and η_k are as above. This formulation has even simpler dynamics than the one above.

It is also possible to reformulate the problem so that the model has nonlinear dynamics, but is linear in the observation. This is the approach used by Aidala and Hammel for their MP filter. In terms of the algorithm in this paper, however, this tradeoff of observation nonlinearity for system nonlinearity is not appealing. Nonlinearities in the system dynamics can make the inverse image set very unwieldy, and make the calculation of equation (3.14b) considerably more difficult. Unless the change also results in a corresponding reduction in the complexity of the integrand, it would not be worthwhile. In this case it does not, so we will concentrate on the cartesian formulation. Of the two above formulations, the second has the simpler dynamics (basically none), and since they are otherwise equivalent, we will use it.

Having selected the model, we can now develop the equations to implement the algorithm. The first step is to define the grid. The easiest grid to use is based on hyper-rectangles defined by

$$\mathrm{I}_j = \left\{ x_{(i)} \in \left(a_{(i)}(j - \frac{1}{2}) + b_{(i)} \,,\, a_{(i)}(j + \frac{1}{2}) + b_{(i)} \right) \; ; \; i = 1..4 \right\}$$

This is definitely not to imply that this is the optimum choice for the grid. It is certain that other grid definitions could conform more closely to the density shapes and hence increase the efficiency of the algorithm by minimizing the number of grid cells with non-negligible mass. For clarity of presentation, however, this simple definition is probably best.

The next step is to determine the inverse image set defined by equation (3.14a). Since there are no system dynamics in this case, the inverse image set H_j is simply I_j itself. This makes the limits for the integral in equation (3.14b) extremely simple. If the grid has not been changed from the last time step, the two grid systems coincide exactly and (3.14b) becomes

$$\psi_j = \frac{1}{\alpha} \pi_{k,j} \int_{\mathrm{I}_j} \mu_k(x,z) \, dx$$

If we wish to improve the efficiency of the algorithm, we can make the grid dynamic so that it can conform to the current density. If this is done, the intersection will have to be calculated in order to evaluate (3.14b). If the new grid is also based on hyper-rectangles, then the intersection of two cells is also a hyper-rectangle with easily calculated edges. Turning now to the integrand, we recall that $\mu = p(z \mid x)$, so that for this problem

$$\mu_k(x,z) = (2\pi R)^{-\frac{1}{2}} \exp\left\{\frac{-1}{2R}(z - \tan^{-1}\left(\frac{x_{(1)} + k\delta x_{(3)} - p_{r,x}(k\delta)}{x_{(2)} + k\delta x_{(4)} - p_{r,y}(k\delta)}\right))^2\right\}$$

You should note that this makes the integral above rather complicated, even for the simple case of a fixed grid. Evaluating this integral is the primary computational burden in implementing the algorithm for this problem, but little effort was put into optimizing the coding for its evaluation.

As pointed out earlier, there is no system input noise for this problem. This eliminates the need for equation (3.14c). Note that this also negates one of the benefits of the algorithm presented here; since the convolution does not enter the problem, we get *no* benefit from being able to use the FFT technique to evaluate it. Despite this, it is still worthwhile to apply this algorithm with the expectation of getting a reasonable approximation to the actual posterior density.

Finally, equations (3.14d) and (3.14e) are implemented directly to normalize the updated density.

6.3. Qualitative Density Analysis

Often we can learn a great deal by looking qualitatively at the density behavior with time, motivated by an enlightened consideration of the problem from the Bayesian viewpoint. Qualitative analysis is largely visualization of the density, and imagining the effects in different situations. Visualization of the density was fairly easy in the last chapter, where the density was a simple surface in three-space. Visualization for this problem is much harder, since most people aren't equipped to imagine surfaces in five dimensional space. Two possible alternatives to help with the visualization come to mind: we can consider 'slices' of the density, or look at the marginal densities. For slices, we fix two of the four state variables and look at the resulting surface in three-space. For the marginal densities, we usually consider the joint density of two of the states, so we again have a surface in three space. The problem with looking at the marginals is that they tend to obscure the fine detail of the density. Marginals are like projections of the density onto subspaces, so it is somewhat like trying to guess the shape of an object from its shadows. Dealing with only slices, though, still leaves the problem of integrating the individual visualizations to provide an understanding of the entire density. Neither approach is entirely satisfactory. so both may be necessary to describe the density behavior.

We begin by reviewing the measurement update function μ. The peak of the function (where the argument of the exponential is zero) lies along the hyperplane defined by

$$\tan(z) = \frac{x_{(1)} + k\delta x_{(3)} - p_{r,x}(k\delta)}{x_{(2)} + k\delta x_{(4)} - p_{r,y}(k\delta)}$$

and contours of the density lie along the related hyperplanes

$$\tan(z+c) = \frac{x_{(1)} + k\delta x_{(3)} - p_{r,x}(k\delta)}{x_{(2)} + k\delta x_{(4)} - p_{r,y}(k\delta)}$$

where c determines the density value on the contour. For a particular velocity (fixed $x_{(3)}$ and $x_{(4)}$), the image of the density in position space ($x_{(1)} \times x_{(2)}$) is a ridge which gets wider as it gets further away from the origin. The peak is along the line

$$x_{(2)} = \frac{1}{\tan(z)} x_{(1)} + C$$

where C depends on the fixed velocities, the measurement, and the position of the receiver. So for each fixed velocity slice, the angle of the ridge with respect to the $x_{(1)}$ axis is the same, but the intercept changes from slice to slice. Roughly speaking, the value of μ at a point represents the likelihood that a target starting from that point with the fixed velocity could be responsible for the given observation.

With this in mind, lets look at the propagation of the density in position space for a fixed velocity. As usual, start with a uniform initial density. At the first time step ($k=0$), $C=0$, so the μ ridge line runs through (0,0) along the first observed bearing line. Since the prior is uniform, the result of the measurement update is a duplicate of μ. Note that μ will run along only the positive range half of the bearing line. For the second measurement, C is nonzero and presumably z is different than the first, so the ridge of the prior and the ridge of μ intersect. The multiplication of the measurement update results in a hump at the intersection. The spread will depend on the width of the ridges at the intersection and on the angle of the intersection. The location of the hump identifies the appropriate starting point for a target with the given velocity to meet the observations. With the third measurement, we get yet another C, z, and μ. This time, the μ ridge may or may not align with the peak in the prior. If it does not, the result is attenuation of the peak. This indicates that fixed velocity is not compatible with the observations. The total probability associated with this slice will decline. If μ does align with the peak in the prior, the peak is accentuated. In this way, certain velocity slices will build up peaks at particular locations. The height of the peak and its spread is indicative of the likelihood that the velocity associated with that slice could be associated with the observations. The location of the peak gives the required initial location for that velocity to have given those measurements. Taken together, the peaks for all the velocity slices form a line in the state space, giving a set of likely initial position/velocity pairs.

When the receiver maneuvers, it changes the progression of C and z. If we were starting from a uniform prior, this would build up a different line through the state space. Instead, we are starting with the density resulting from all the observations before the maneuver. In some velocity slices, the peaks from the before and after maneuver observations will be far apart. This indicates that there is not likely that a single starting point could explain both sets of observations. In those slices, both peaks will be attenuated. In others, the peaks will be close together, resulting in an accentuated peak. In this way the most likely of the position/velocity pairs are picked out.

In terms of the entire density, we can imagine the first μ as centered on a vertical plane. Successive μs are rotated with respect to the first, but all intersecting along a line. The receiver maneuver changes the rotation so that the intersection of succeeding μs is along a different line. The intersection of these two lines determines the eventual peak of the full density. Actually, this is the idealized behavior. The presence of noise perturbs the observations from the natural progression, so that the hyperplanes no longer intersect along a single line. For some noise sequences, this can mimic the effect of a small receiver maneuver by appearing to produce two distinct intersecting lines. This can result in apparent convergence, or false observability. These effects usually average out, but can give temporary spurious peaks in the density, particularly before the maneuver.

The behavior in fixed position slices is the same. For a fixed position, the density in velocity space converges to the velocity most likely to have produced the given observations starting from the fixed position. If the given position is not compatible with the measurements, the density in that slice flattens out and goes to zero.

It is now fairly easy to see what the marginal densities will look like. After the first measurement, the marginal position density will be a replica of the first μ, since each fixed velocity slice is identical. The marginal velocity density, though, will still be uniform. As measurements are processed, the marginal densities will converge to ridges, with peaks along the projections of the line that the peak of the entire density is on. After the maneuver, the marginal densities will begin to converge to peaks. Note that it is not appropriate to imagine the marginal density as the product of the before maneuver and after maneuver marginal densities as we did with the entire density.

What conclusions can we draw from this analysis? Two come quickly to mind. First, we see the perhaps obvious relationship between distance and convergence. Looking again at a fixed velocity slice, we note that the spread of μ in the vicinity of the peak is proportional to the range to the target for the current measurement for a target with that initial location and velocity. Thus, we will get better convergence in the true velocity slice if we can maneuver to keep the range to the target at a minimum for all the measurements. As a corollary, this says that convergence will be better for a target that happens to be closer, than one which is far away. Second, it provides us a potential criterion for picking better maneuvers. Recall the second measurement update described above. We noted that the convergence of the peak in the fixed velocity slice depended on the angle of intersection of the two ridge lines. If the two are nearly parallel, convergence will be mostly perpendicular to the original ridge line. If they are at right angles, the convergence will be mostly along the original ridge line. This suggests that we should try to maneuver so as to get a wide range of bearings, to ensure convergence in all directions. The other effect that a maneuver has is in shifting the mutual intersection line of the succeeding μs. For best convergence, we would like the new line to be skewed away from the old line as much as possible. Although it has not been explored in this paper, it is possible that strategies could be developed with this in mind.

6.4. Comparison of the Conditional Mean to a Point Estimator

As we discussed in the last chapter, one use of the Bayesian approach is in describing the behavior of the true optimal estimate for comparison to suboptimal or approximate estimators. In this section we compare the mean of the conditional density to Aidala and Hammel's MP filter. As in the last chapter, the point estimator is intended to approximate the minimum variance estimate of the state. The mean of the posterior density, on the other hand, is the minimum variance estimate.

The first issue in using our approach is how many points to use in the grid. Some preliminary experiments were run using between six and ten points on each dimension. All the runs used the same geometry and random number seed to insure comparability. The runs showed continued change as the number of points increased, although the size of the change decreased and was fairly small between the nine and ten point runs. This coupled with the rapidly increasing computational and storage requirements suggested that nine or ten points on each dimension would be satisfactory. Since we are not using a particularly fine grid, it is worth emphasizing that this section is really a comparison of two approximate techniques. The approximate density is a close approximation to the true density, but it is not an exact representation.

In this comparison, we will look at four different scenarios (target/receiver geometries) gathered from other papers on the bearings-only problem. For each scenario we will present typical individual runs. Each run covers 16 minutes, with one sample per minute. The receiver always starts at coordinates (0,0) and proceeds at constant velocity for the first 8 time steps. Immediately after the eighth observation, the receiver executes an instantaneous maneuver to a new velocity, and maintains that velocity until the end of the run. The target starts at a specified position, and proceeds at constant velocity for the entire run. All distances are in meters, and speeds in meters per minute. The first plot of each set shows the target and receiver tracks for the entire run, and the estimated target tracks for time steps nine through sixteen. The second figure of each set shows the estimator errors versus time. The errors are plotted as percentage error in range to target and speed of target, and degree error in bearing to target and target heading.

Filter initialization is always a difficulty. The MP filter was initialized in accord with the recommendations in Aidala and Hammel's paper, with the exception that the initial range estimate was adjusted to near the true initial range instead of being fixed at 10 km. The initial density for the Bayes' algorithm was uniform over the initial grid, which in turn was chosen to include the true position. Aside from the requirement that the initial grid must contain the true initial conditions, it is clear that we want to pick a small grid to increase the resolution in the main region of the density. What may not be clear is that we also need the grid to be large enough to cover all the density, so that the mean is accurate. With an adjustable grid, it is usually better to err on the large side and let the grid adapt to the density.

The first two examples are from Aidala and Hammel's paper that introduced the MP filter [44]. The first is their long range scenario. The target is initially at $P_t(0)=(24688,0)$ with velocity $V_t=(617,0)$. The receiver begins with velocity $V_r=(617,617)$ and has $V_r=(617,-617)$ after the maneuver. The noise variance is $R=1.4$ degrees squared. The initial grid for the Bayes' algorithm is defined by

$$\left\{x_{(1)}\in(15000,40000)\ ;\ x_{(2)}\in(-2000,2000)\ ;\ x_{(3)}\in(100,1000)\ ;\ x_{(4)}\in(-400,400)\right\}$$

This is a somewhat unrealistic example, since the receiver is managing to follow the target at same v_x velocity. Figure 6.2 shows the tracks for this example. The two estimates show similar performance. Turning to the error plots in Figure 6.3, we see that the Bayes' mean has somewhat better range and speed estimates, and essentially equal bearing and heading errors. This is not surprising, since the bearings to the target a fairly small (note the different x and y scales) making the linearization of the EKF more accurate.

The second example is Aidala and Hammel's short range scenario. It is the same as above except that the target is initially at $P_t(0)=(2468,0)$, and the initial Bayes grid is defined by

$$\left\{x_{(1)}\in(1500,3000)\ ;\ x_{(2)}\in(-300,300)\ ;\ x_{(3)}\in(100,1000)\ ;\ x_{(4)}\in(-300,300)\right\}$$

We have two examples for this scenario, differing only in the specific noise sequences used. In the first (Figures 6.4 and 6.5), the MP filter manages to maintain a good estimate of the bearing to the target, but otherwise does quite poorly. The Bayes' mean on the other hand does very well. The data for time step eight (not graphed), shows that both estimators had comparable estimates at that time, with the MP filter having a somewhat larger range estimate. After the ninth measurement, both reduced the estimated range, but the MP filter overcompensated and could not recover. In the second example (Figures 6.6 and 6.7), we see a more extreme case. Here, the MP filter cannot even maintain lock on the target bearing. The Bayes' mean begins with a worse estimate than the previous case, but converges anyway.

The behavior of the Bayes' mean is clearer if we look at the approximate density. Figure 6.8 shows the marginal initial position density at time steps eight, ten, and twelve for the second of the above cases. At time step eight, the density has converged along the original bearing line, but not in range. According to our discussion in the previous section, though, the height of the ridge should be nearly constant, but it is not.

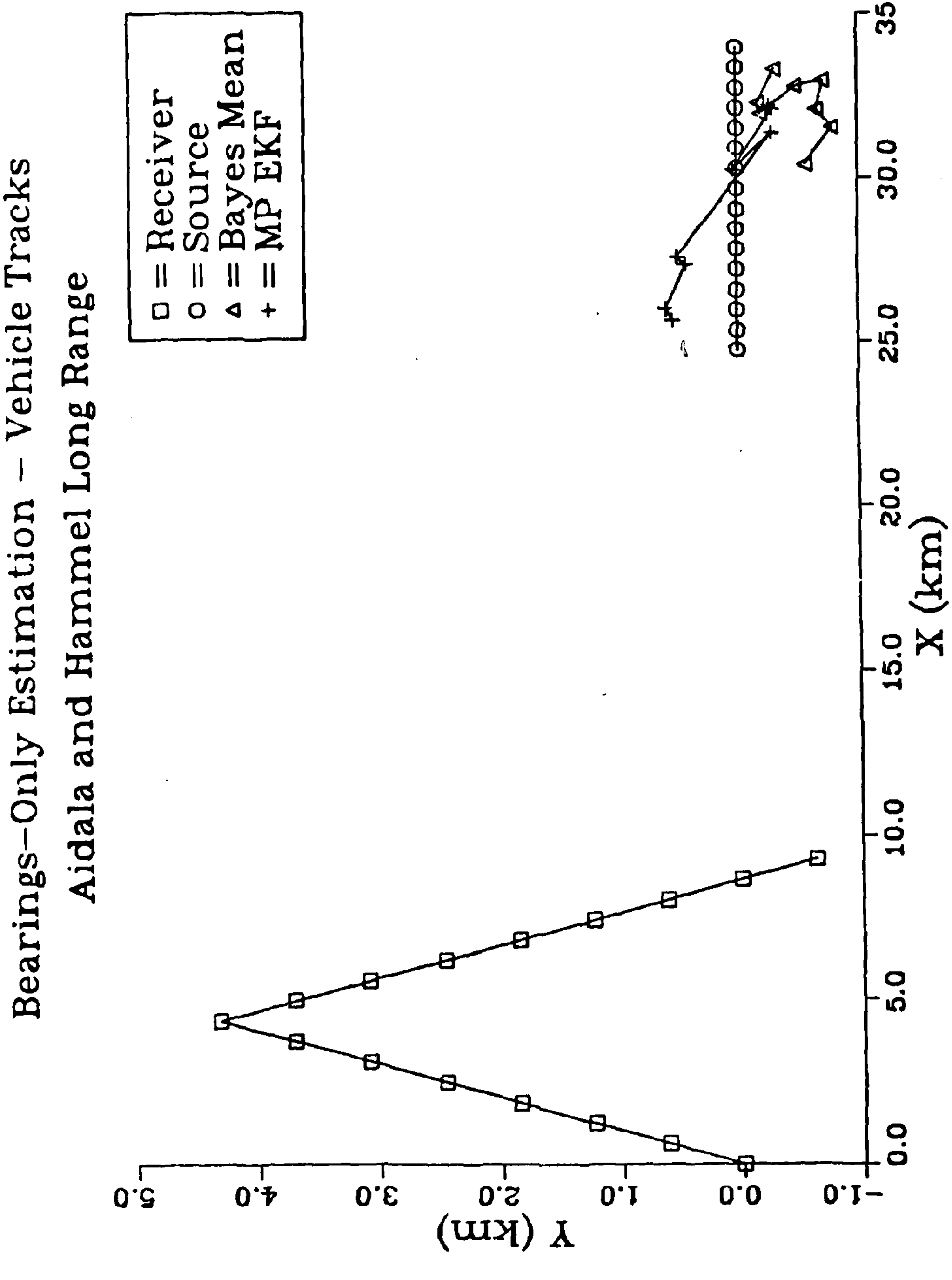
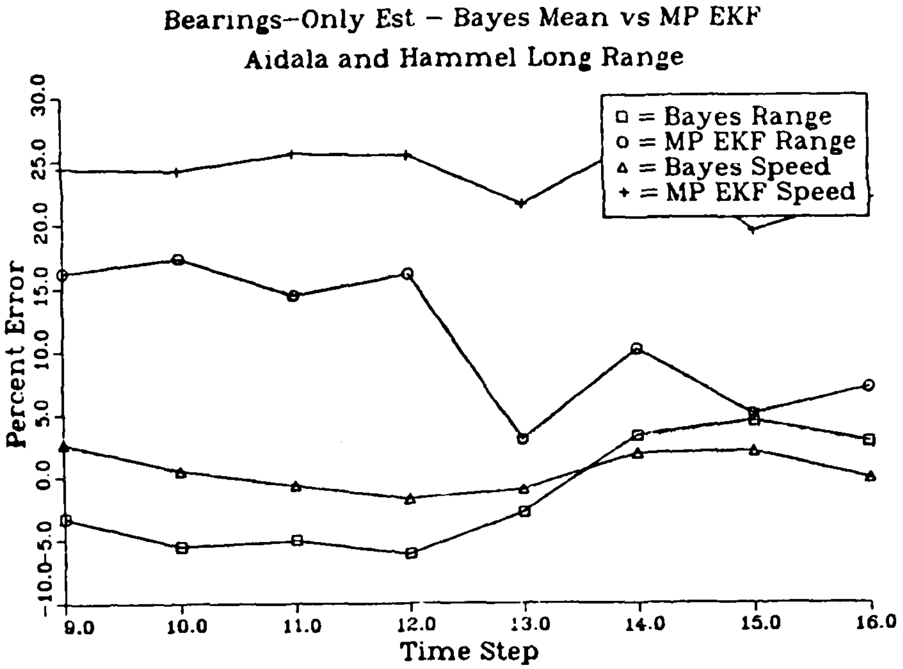


Figure 6.2 - Vehicle tracks for Aidala and Hammel's long range scenario.



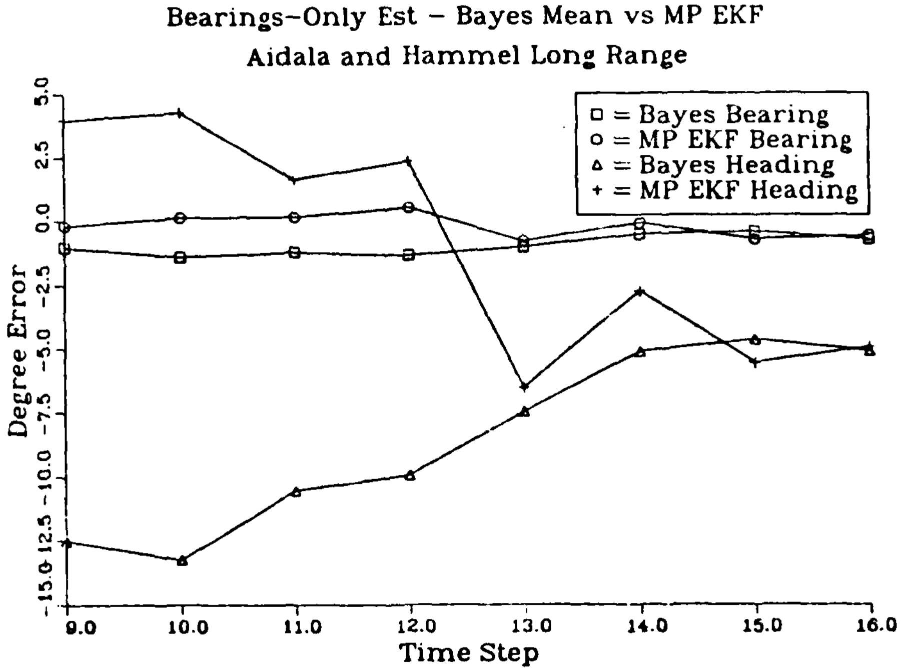


Figure 6.3 - Comparison of estimator errors for Aidala and Hammel's long range scenario.

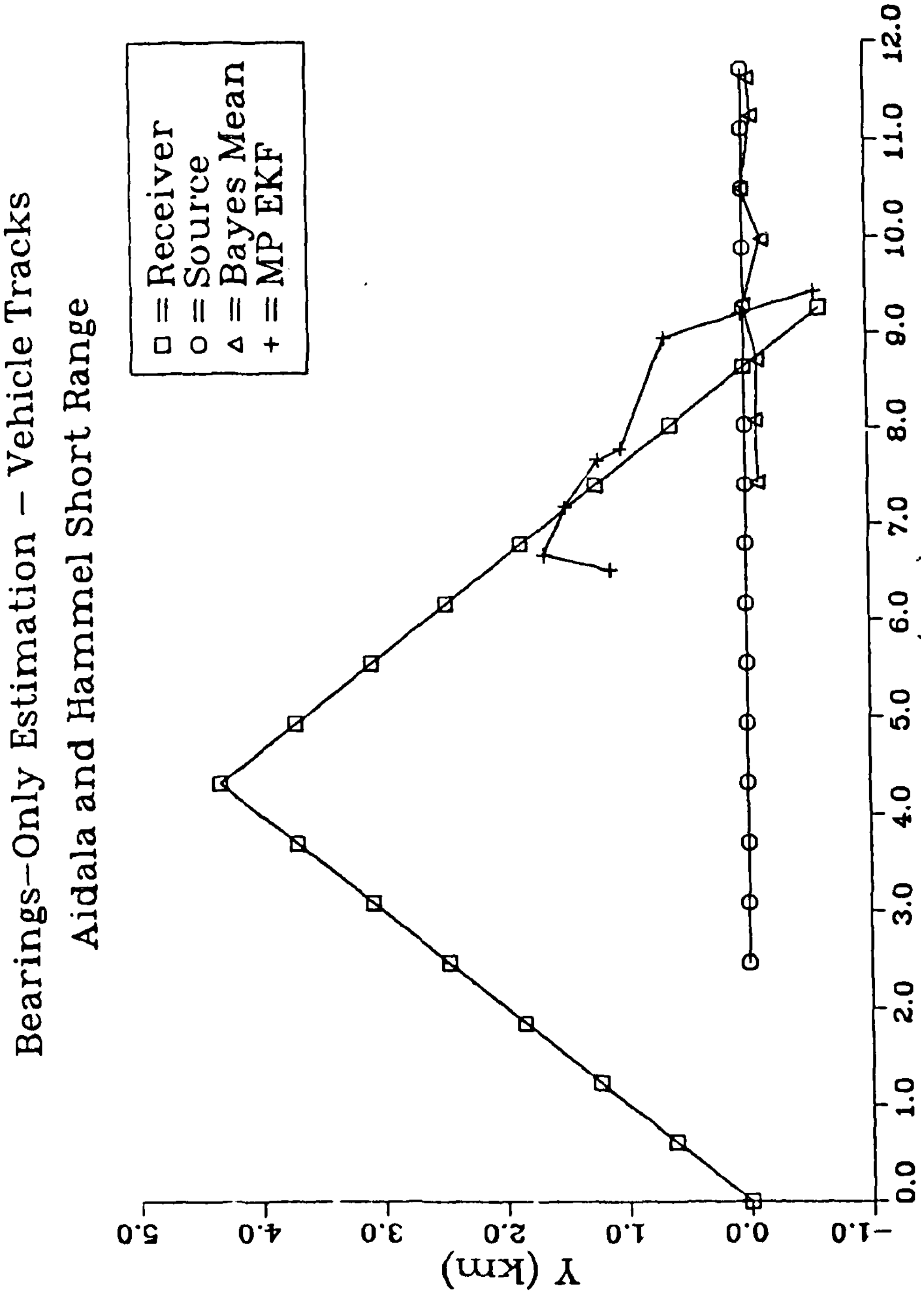
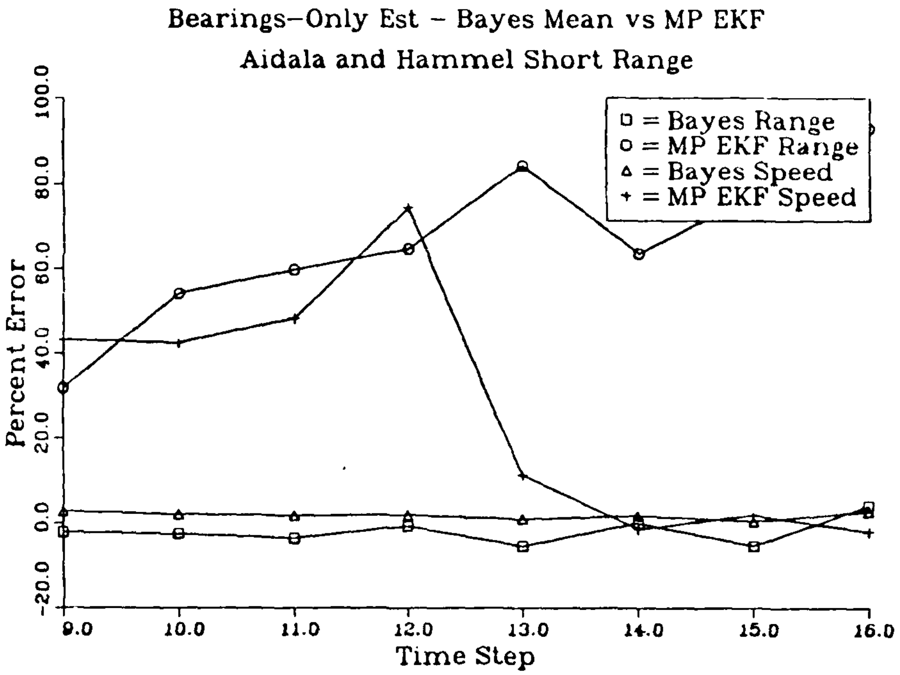


Figure 6.4 - Vehicle tracks for Aidala and Hammel's short range scenario, first case.



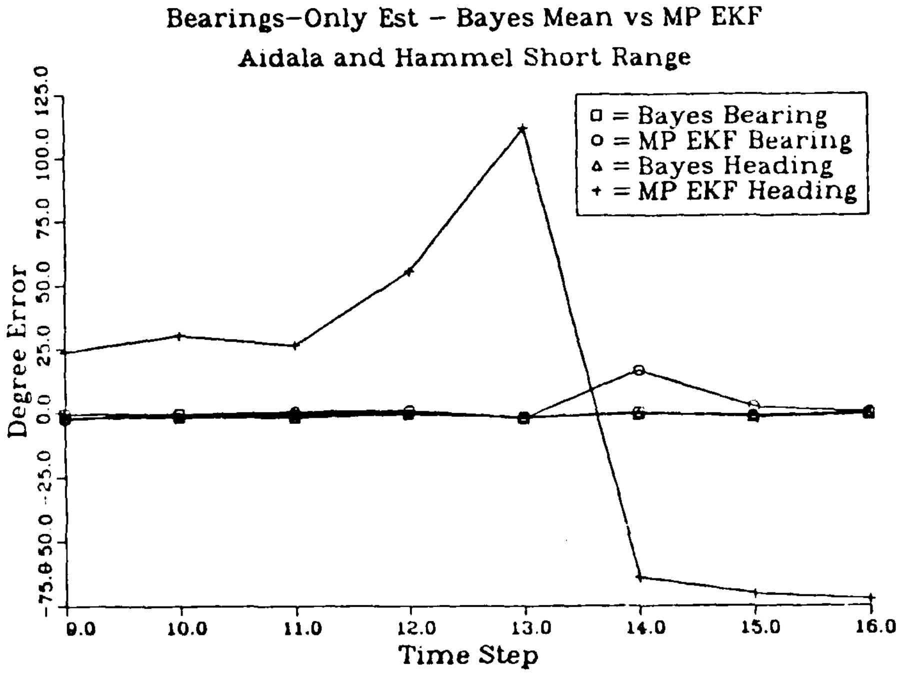


Figure 6.5 - Comparison of estimator errors for Aidala and Hammel's short range scenario, first case.

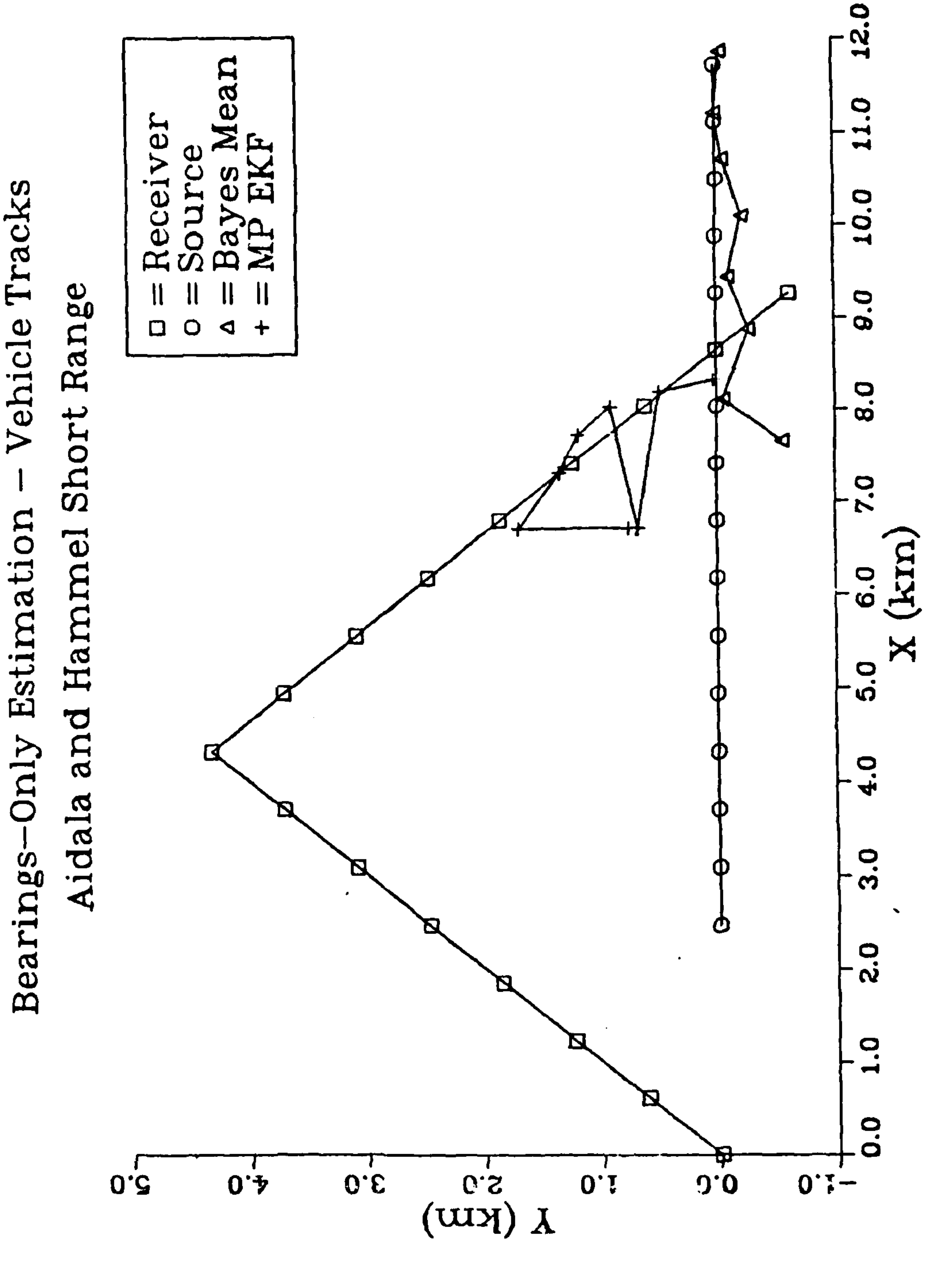
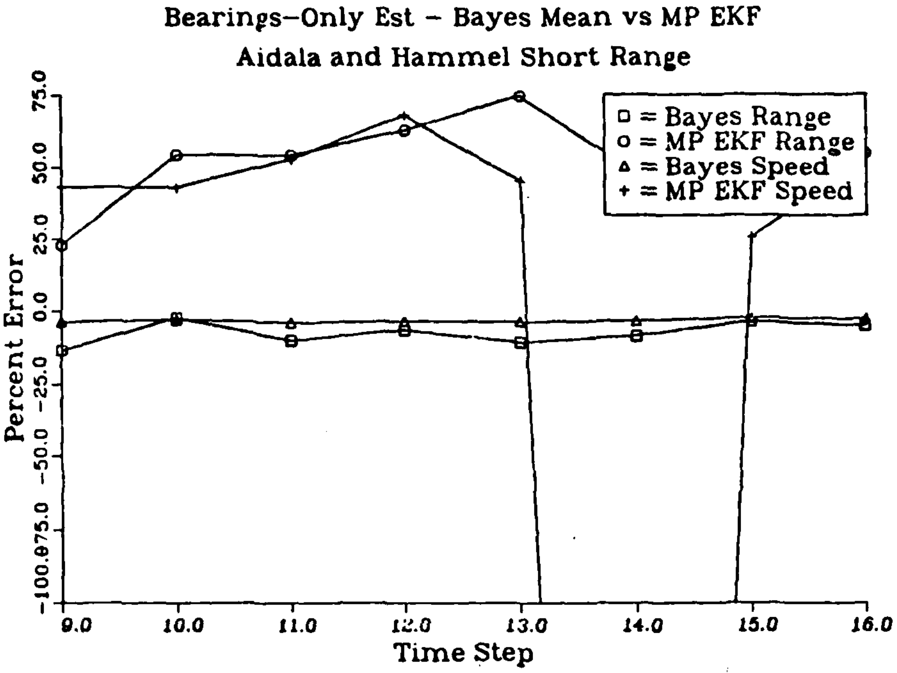


Figure 6.6 - Vehicle tracks for Aidala and Hammel's short range scenario, second case.



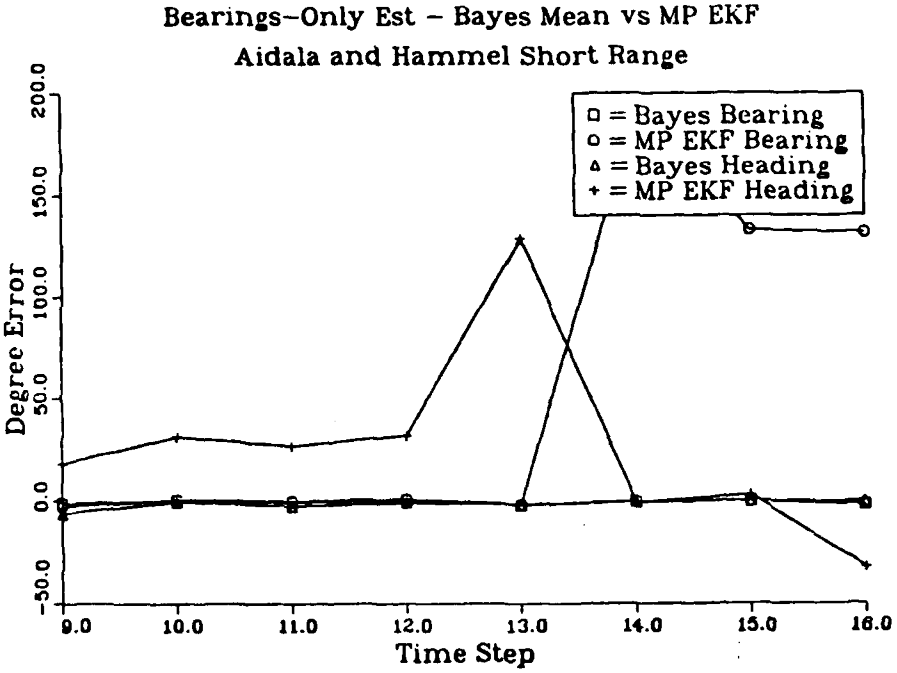


Figure 6.7 - Comparison of estimator errors for Aidala and Hammel's short range scenario, second case.

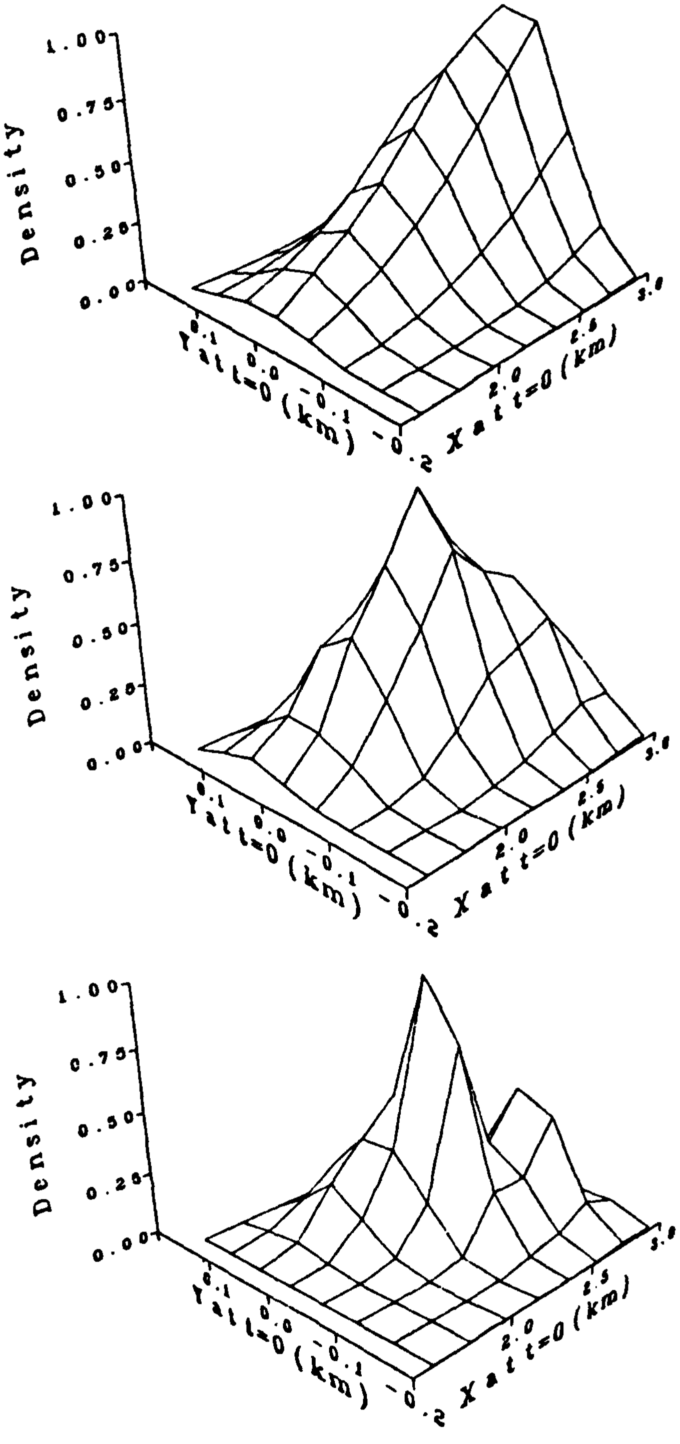


Figure 6.8 - Marginal initial position densities for Aidala and Hammel's short range scenario, second case, time steps eight, ten, and twelve.

This is a result of the algorithm averaging μ over fixed size cells. Recall that in a fixed velocity slice μ is a wedge shaped ridge with a constant height along the ridge line. The average over a cell near the origin on the ridge line will be small, since μ rolls off quickly. On the other hand, the average over a cell away from the origin will be larger, since μ remains near its maximum over a larger area. This effect is predicted by the earlier error analysis, i.e., the error in the approximation increases when the density is sharply peaked in relation to the grid size. Also note that the peak of the density is at the extreme edge of the grid, implying that the grid does not cover the entire density. This is to be expected, since the actual density is infinite in the range direction at this point. The loss of mass is not a problem in this application, since we are not performing the convolution step (remember we have no plant noise). Hence, the prediction density has only local dependencies and is not affected by the loss of mass. Following the maneuver, however, the density begins to converge in range, and by time step ten we see marked convergence. Note that the grid almost entirely covers the density. There appears to be some mass left off at the far edge, so the calculated mean range is probably a bit on the short side. By time step twelve, the density has converged enough that the grid effectively covers it entirely, and it exhibits a strong central peak. There is also a noticeable secondary peak, but it is gone by the fourteenth time step.

The next scenario comes from Sorenson's p-vector paper [23]. The target is initially at $P_s(0)=(2164,-190)$ with velocity $V_s=(-280,310)$. The receiver begins with velocity $V_r=(63,94)$ and turns to $V_r=(-94,63)$ after the maneuver. The noise variance is $R=0.5$ degrees squared. The initial grid for the Bayes' algorithm is defined by

$$\left\{x_{(1)} \in (1500,3200)\ ;\ x_{(2)} \in (-800,800)\ ;\ x_{(3)} \in (-500,100)\ ;\ x_{(4)} \in (-100,500)\right\}$$

Figures 6.9 and 6.10 display the results for this case. The MP filter gets a good estimate of the true bearing to the target, but does very poorly in all other regards. The Bayes' mean does much better, but still not terribly well. Figure 6.11 shows the density for this case at time steps nine and sixteen. The density at step nine (first measurement after the maneuver), shows some convergence in range, but the very little over the following measurements. This is due in part to the fact that the point on closest approach occurs at about the sixth time step, well before the maneuver. Hence, the spread of the measurement density near the true range is getting wider with each measurement, severely curtailing the possible convergence.

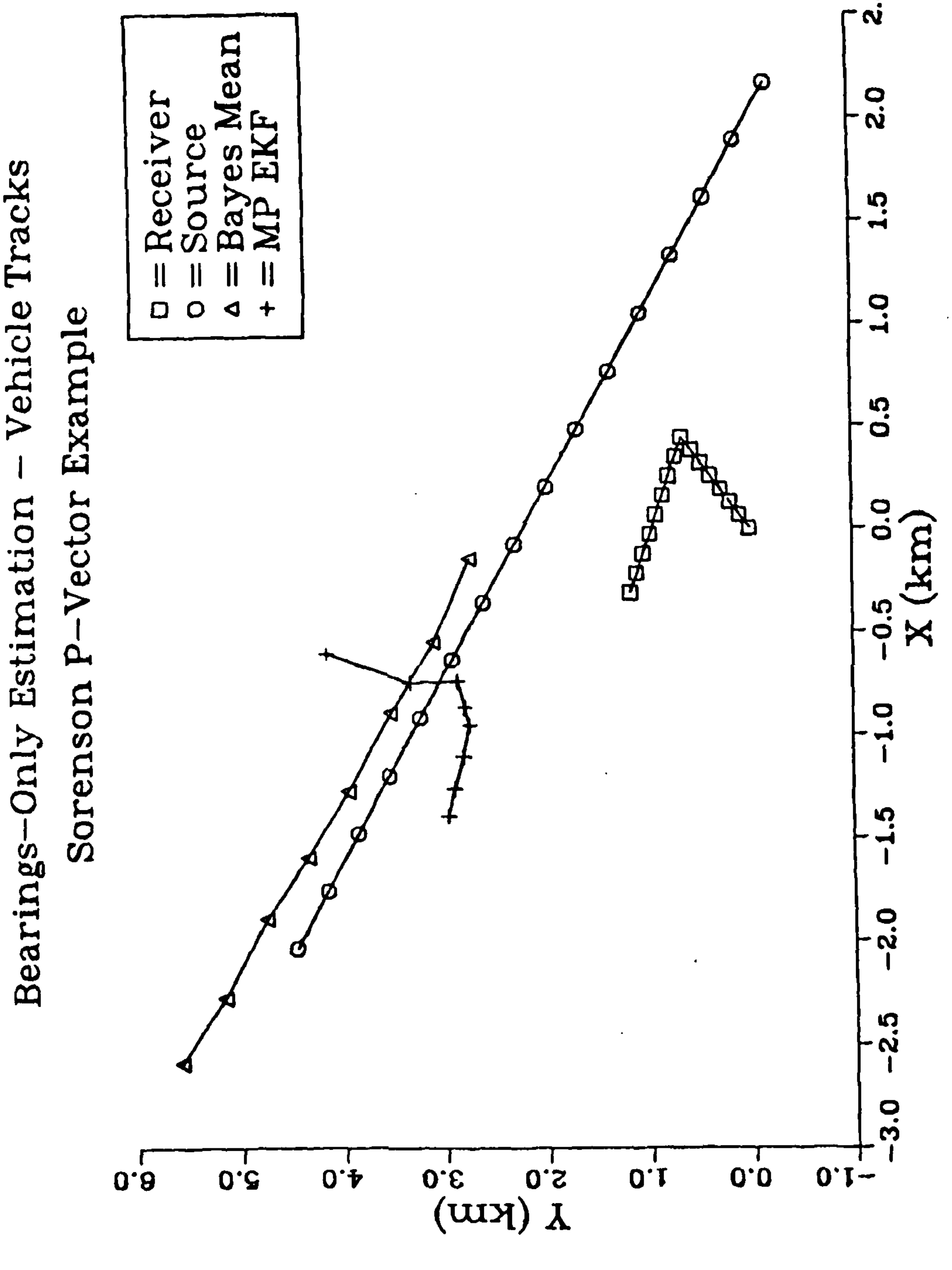


Figure 6.9 - Vehicle tracks for Sorenson's p-vector example.

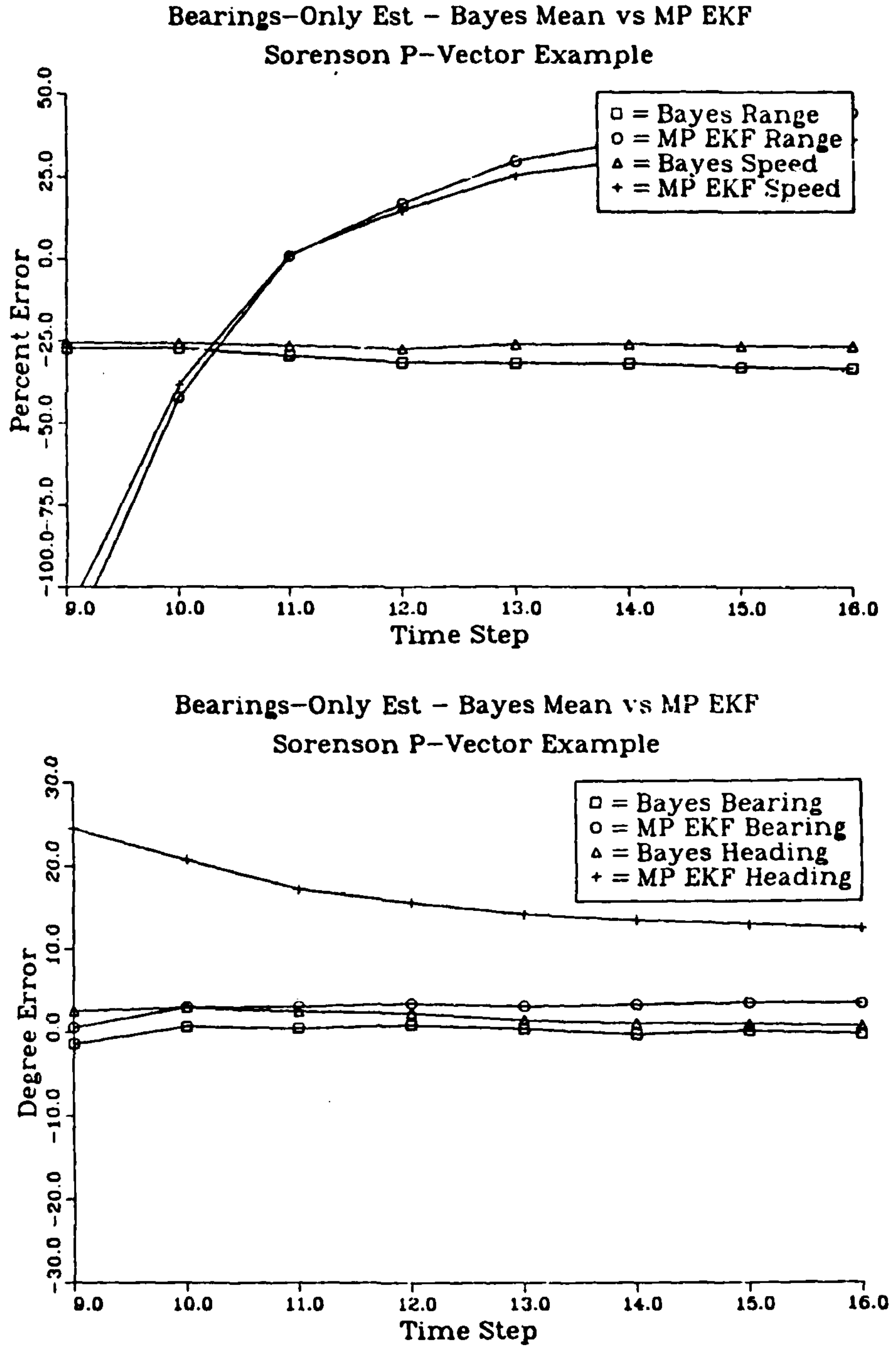
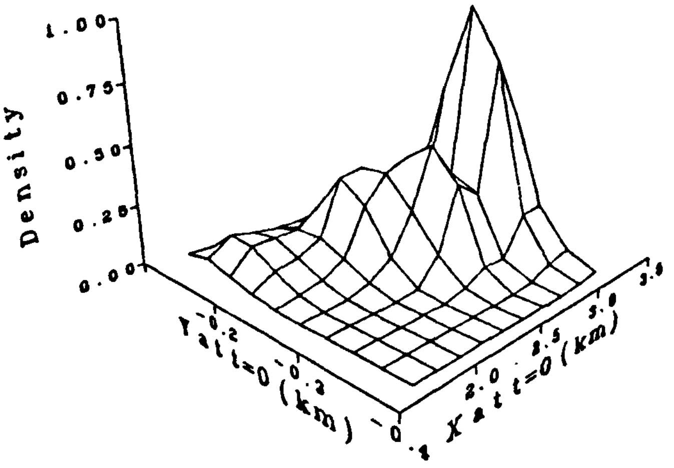


Figure 6.10 - Comparison of estimator errors for Sorenson's p-vector example.

Sorenson P–Vector Example
Marginal Initial Position Density – k=9



Sorenson P–Vector Example
Marginal Initial Position Density – k=16

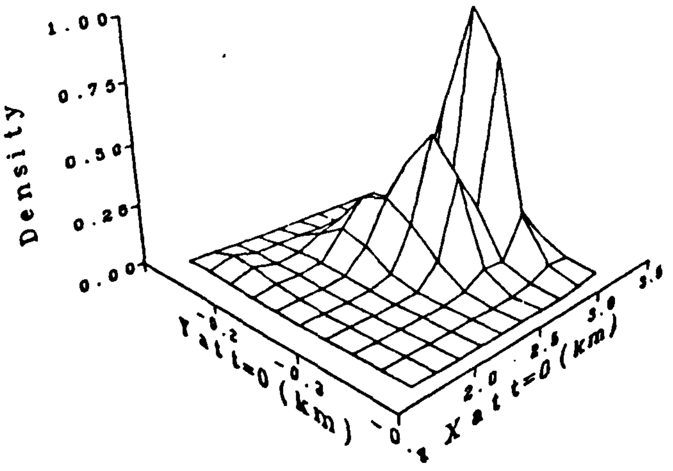


Figure 6.11 - Marginal initial position densities for Sorenson's p-vector example, time steps nine and sixteen.

The last example is from a paper by Aidala on Kalman filter behavior [42]. The target is initially at $P_t(0)=(4383,-383)$ with velocity $V_t=(-482,575)$. The receiver begins with velocity $V_r=(98,169)$ and turns to $V_r=(-177,82)$ after the maneuver. The noise variance is $R=9.0$ degrees squared. The initial grid for the Bayes' algorithm is defined by

$$\left\{x_{(1)} \in (3000,6000)\ ;\ x_{(2)} \in (-1000,100)\ ;\ x_{(3)} \in (-600,100)\ ;\ x_{(4)} \in (0,800)\right\}$$

This is a particularly difficult scenario due to the long range and high observation noise level. As can be seen in Figures 6.12 and 6.13, the MP estimate diverges for this case. At time step eight (not graphed), immediately before the maneuver, the MP filter estimate is fairly close to the true value, though not as close as the Bayes' mean. The first measurement after the maneuver causes the MP range estimate to get very large. The next measurement induces such a large residual that the range estimate goes negative, and the MP filter then locks onto this 'negative' track, giving a bearing error of roughly 180°. To check whether this behavior was due to the particular noise sequence, the MP filter was rerun ten times using different random number sequences. The same behavior was exhibited for all runs, indicating that it is a result of the geometry or the initialization of the MP filter. The Bayes' mean, on the other hand, does well, although not as well as the graph of the tracks would seem to indicate due to scale of plot.

These results confirm the advantages of using the conditional mean of the approximate density as an estimator. The consistently good performance of the mean in turn confirms a reasonable degree of accuracy in the approximate density generated by the algorithm. These results also indicate that the MP filter does not live up to theoretical limits, since it does not even match the mean of the approximate density. We also see that the MP filter has extremely poor performance for at least some geometries.

The good results from the approximate Bayes' mean come at a price, however. The algorithm took roughly 50 seconds of central processing time on a CDC CYBER computer per time step for a 10×10×10×10 grid. (Even so, this is roughly real time, since most sea-based applications have one minute time steps.) The most time consuming portion of of the algorithm is the integration of the measurement density. In fact, approximately 90% of the program run time is spent in the integration routine. Since no vigorous attempt was made to optimize this part of the program, it is reasonable to expect that it could be improved considerably. The program could also be speeded up by taking advantage of the considerable parallelism in the algorithm. Even if the

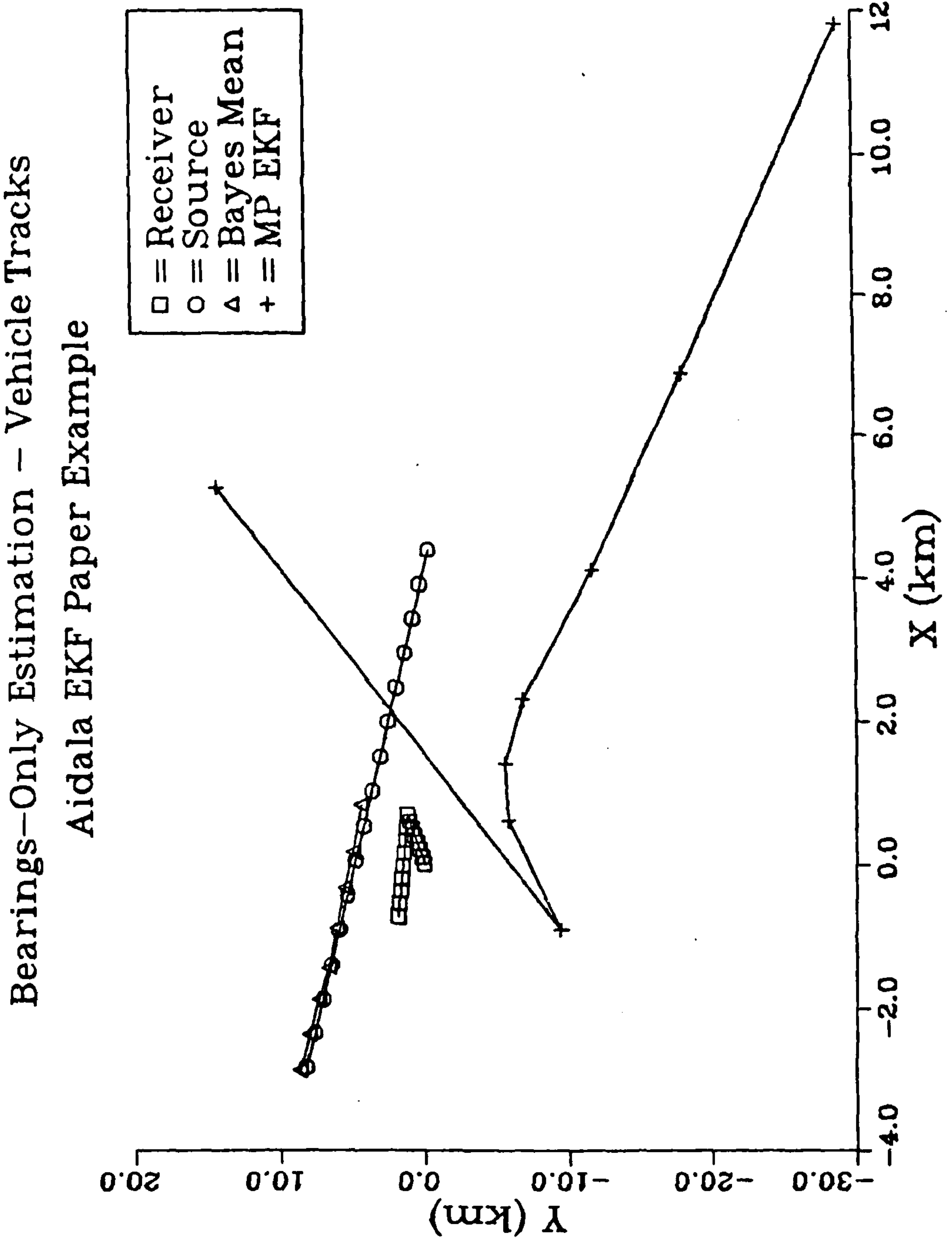


Figure 6.12 - Vehicle tracks for Aidala's EKF example.

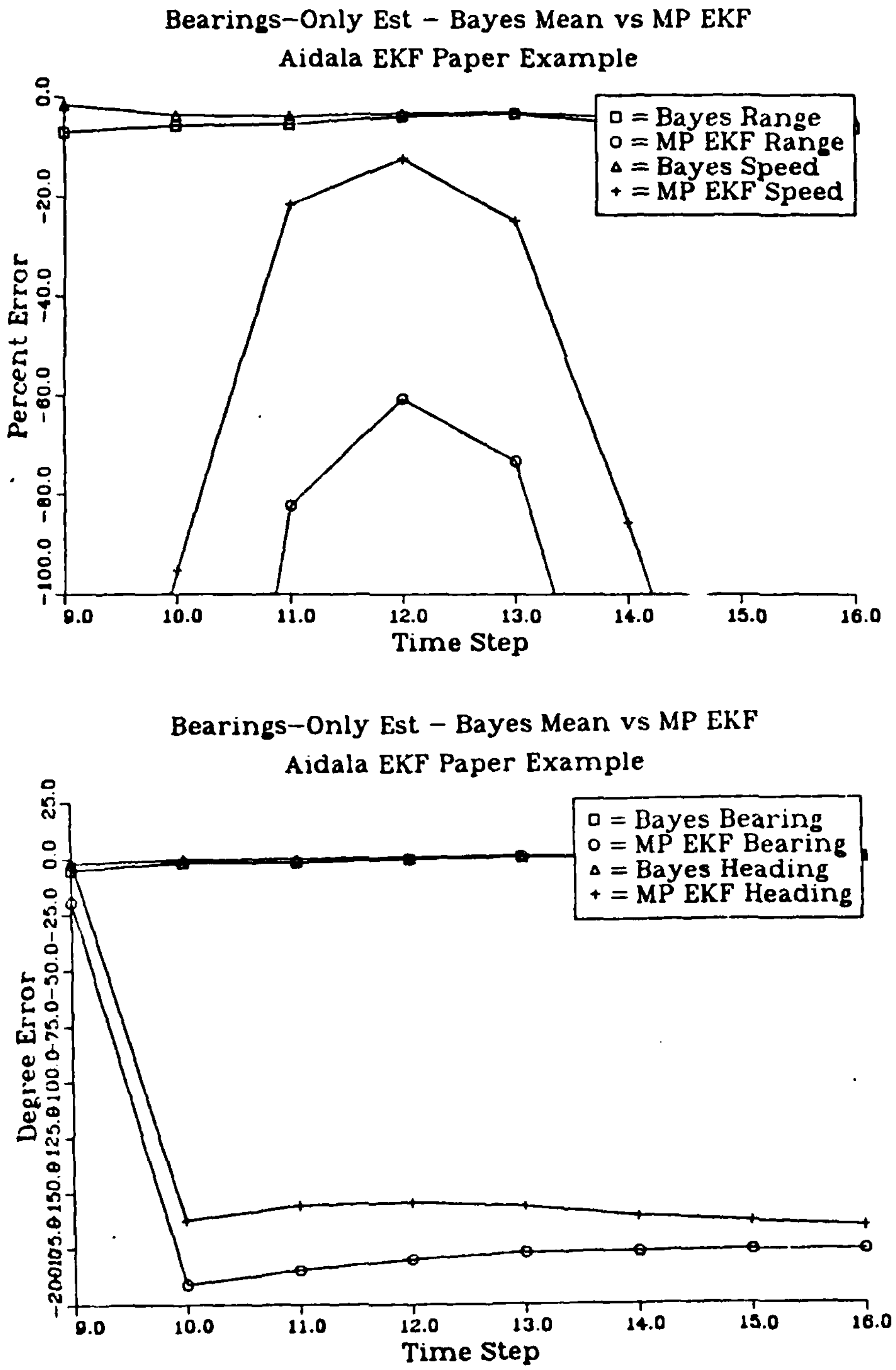


Figure 6.13 - Comparison of estimator errors for Aidala's EKF example.

computational burden remains too large for on-line implementation, the technique is still valuable as a benchmark for other approximate estimators.

7. SUMMARY AND CONCLUSIONS

This dissertation has presented the derivation and application of a new algorithm for recursively generating an approximation of the conditional state density for a nonlinear, discrete-time system.

The algorithm is derived by taking the formal recursive equations for the conditional density and consistently using piece-wise constant approximations for the various constituent densities. The resulting algorithm is given by equation set (3.14). As an interesting aside, the algorithm can also be derived by considering the recursive calculation of the probability mass associated with each grid cell. In other words, the algorithm can also be thought of as a probability mass filter.

There are two main advantages to this algorithm. First, the time update becomes a discrete linear convolution, instead of a nonlinear convolution as in other algorithms. This allows the convolution to be evaluated using FFT techniques, costing $O(3N\ln(N)+N)$ operations instead of $O(N^2)$ operations, where N is the number of grid points. Second, the algorithm allows a thorough analysis of the propagation of the error in the approximate density. The error analysis shows that the increase in the error at each iteration is bounded, although the bound is not very small. More importantly, the analysis yields a characterization of the situations where large error growth could be expected. This characterization indicates that we can expect the error growth to be far less than the bound in most cases of interest. The expected error stability was demonstrated by extensive comparisons of the approximate density to the true density for a linear time-invariant system with Gaussian noises. The results demonstrate the stability and accuracy of the algorithm over a wide range of conditions.

For the next two applications, the algorithm was considered as a tool in implementing a Bayesian approach. Two common but problematic nonlinear systems were considered. The first was the case of a scalar linear system with an unknown parameter. The objective was to estimate both the state and the parameter simultaneously. The second case was that of a moving receiver taking noisy bearing measurements of a moving target, and trying to estimate the target's position. In both cases the specific equations for implementing the algorithm were derived, and used to augment a qualitative analysis of the density behavior. The algorithm was then used to calculate an approximation to the conditional density, which in turn yielded the conditional mean, which is the minimum variance estimator. This approximation to the optimal estimate was then compared to appropriate point estimators for each case, showing the superiority of the mean of the approximate density.

We obtain several benefits from this combination of the Bayesian approach and this new algorithm. First, one can develop considerable insight into system behavior by considering the problem in terms of the probability densities. This provides a unique perspective not available otherwise. The algorithm augments this by providing a means of calculating the densities so that their behavior can be examined. Second, the combination provides a benchmark for all other estimation schemes. Since the actual density function is available, the true optimum estimate for any given loss function can be calculated. Thus, the performance of any other estimator can be compared to the best possible performance. Third, we could consider actually using this combination to provide an optimal estimator for online use. This would be particularly useful in cases where the transient performance is critical. The algorithm is currently fairly expensive computationally, but no attempt had been made to take advantage of its parallelism. Customized computer architectures would undoubtedly provide immense increases in speed.

In a broad sense, this dissertation shows the advantages of taking the Bayesian view, and provides an improved tool for implementing a Bayesian approach. Taking a Bayesian view can lead to a deeper understanding of the behavior of the system, and hence to a better appreciation of the potentials and limitations of approximate estimators. At best, the Bayesian approach can actually provide the optimal estimate for a given loss function. At worst, it provides a valuable benchmark for the performance of other estimators.

REFERENCES

[1] L. J. Savage, *The Foundations of Statistics.* New York: Dover, 1972.

[2] G. E. P. Box, G. C. Tiao, *Bayesian Inference in Statistical Analysis.* Reading, Mass.: Addison-Wesley, 1973.

[3] V. Peterka, "Bayesian System Identification," *Automatica,* 17, 41-53, 1981.

[4] C. Johnson, E. B. Stear, "Reproducing Probability Density Classes on Lie Groups with Application to Discrete Time Filtering," *IEEE Trans Inform Theory,* IT-26, 124-129, 1980.

[5] L. D. Servi, Y. C. Ho, "Recursive Estimation of Uniformly Distributed Measurement Noise," *IEEE Trans Automatic Control,* AC-26, 563-564, 1981.

[6] H. W. Sorenson, A. R. Stubberud, "Nonlinear Filtering by Approximation of the A Posteriori Density," *Int J Control,* 8, 33-51, 1968.

[7] K. Srinivasan, "State Estimation by Orthogonal Expansion of Probability Distributions," *IEEE Trans Automatic Control,* AC-15, 3-10, 1970.

[8] C. Hecht, "Digital Realization of Nonlinear Filters," in *Second Symp on Nonlinear Estimation,* 1971, 152-158.

[9] C. Hecht, "System Identification Using Bayesian Estimation," in *Fourth Symp on Nonlinear Estimation,* 1973, 107-113.

[10] S. R. McReynolds, "Multidimensional Hermite-Gaussian Quadrature Formulae and their Application to Nonlinear Estimation," in *Sixth Symp on Nonlinear Estimation,* 1975, 188-191.

[11] W. W. Willman, "Edgeworth Expansions in State Perturbation Estimation," *IEEE Trans Automatic Control,* AC-26, 493-498, 1981.

[12] R. S. Bucy, "Bayes' Theorem and Digital Realization for Nonlinear Filters," *J Astronaut Sci,* 17, 80-94, 1969.

[13] R. S. Bucy, K. D. Senne, "Digital Synthesis of Nonlinear Filters," *Automatica,* 7, 287-298, 1971.

[14] D. L. Alspach, H. W. Sorenson, "Approximation of Density Functions by a Sum of Gaussians for Nonlinear Bayesian Estimation," in *First Symp on Nonlinear Estimation,* 1970, 19-31.

[15] H. W. Sorenson, D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica,* 7, 465-479, 1971.

[16] D. L. Alspach, H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Trans Automatic Control,* AC-17, 439-448, 1972.

[17] D. L. Alspach, "Gaussian Sum Approximations in Nonlinear Filtering and Control," *Information Sciences,* 7, 271-290, 1974.

[18] J. L. Center, "Practical Nonlinear Filtering of Discrete Observations by Generalized Least-Squares Approximation of the Conditional Probability Distribution," in *Second Symp on Nonlinear Estimation,* 1971, 88-99.

[19] R. J. P. de Figueiredo, J. G. Jan, "Spline Filters," in *Second Symp on Nonlinear Estimation,* 1971, 127-138.

[20] R. S. Bucy, H. Youssef, "Nonlinear Filter Representation Via Spline Functions," in *Fifth Symp on Nonlinear Estimation,* 1974, 51-60.

[21] D. G. Lainiotis, J. G. Deshpande, "Parameter Estimation using Splines," *Information Sciences,* 7, 291-315, 1974.

[22] A. H. Wang, R. L. Klein, "Implementation of Nonlinear Estimators Using Monospline," in *13th IEEE Conf on Decision and Control,* 1976, 1305-1307.

[23] H. W. Sorenson, "The Bayesian Approach to Nonlinear Filtering and Plant Identification," presented at *6th IFAC Symp on Ident and Syst Param Est,* 1982.

[24] R. L. Klein, W. Kasemratanasunti, "Quadrature Formulae and the Implementation of Optimal Nonlinear Estimation," in *Fourth Symp on Nonlinear Estimation,* 1973, 151-160.

[25] A. H. C. Wang, R. L. Klein, "Optimal Discretization and Updating for Quadrature Estimators," in *Fifth Symp on Nonlinear Estimation,* 1974, 282-296.

[26] J. Y. Liang, R. L. Klein, "Nonlinear Estimation Implemented with Optimal Quadrature Formulae," in *Sixth Symp on Nonlinear Estimation,* 1975, 160-170.

[27] A. H. C. Wang, R. L. Klein, "Optimal Quadrature Formula Nonlinear Estimators," *Information Sciences,* 16, 169-184, 1978.

[28] E. Tse, R. E. Larson, S. McReynolds, "The Quantization Problem in the Parallel Implementation of the Optimum Nonlinear Filter," in *Fifth Symp on Nonlinear Estimation,* 1974, 252-255.

[29] R. M. Mesereau, T. C. Speake, "A Unified Treatment of Cooley-Tukey Algorithms for the Evaluation of the Multidimensional DFT," *IEEE Trans Acoustics, Speech, and Signal Proc,* ASSP-29, 1011-1017, 1981.

[30] L. Auslander, E. Feig, S. Winograd, "New Algorithms for the Multidimensional Discrete Fourier Transform," *IEEE Trans Acoustics, Speech, and Signal Proc,* ASSP-31, 388-403, 1983.

[31] H. W. Sorenson, "On the Development of Practical Nonlinear Filters," *Information Sciences,* 7, 253-270, 1974.

[32] "Special Issue on Applications of Kalman Filtering," *IEEE Trans Automatic Control,* AC-28, 254-434, 1983.

[33] Y. C. Ho, R. C. K. Lee, "A Bayesian Approach to Problems in Stochastic Estimation and Control," *IEEE Trans Automatic Control,* AC-9, 333-339, 1964.

[34] J. Spragins, "A Note on the Iterative Application of Bayes' Rule," *IEEE Trans Inform Theory,* IT-11, 544-549, 1965.

[35] H. Jeffreys, *Theory of Probability.* London: Oxford University Press, 1961.

[36] R. C. Jeffrey, *The Logic of Decision.* New York: McGraw-Hill Book Company, 1965.

[37] M. Tribus, *Rational Descriptions, Decisions, and Designs.* New York: Pergamon Press, 1969.

[38] L. Ljung, T. Soderstrom, *Theory and Practice of Recursive Identification.* Cambridge, Mass.: The MIT Press, 1983.

[39] D. F. Liang, G. S. Christensen, "New Estimation Algorithms for Discrete Nonlinear Systems and Observations with Multiple Time Delays," *Int J Control,* 23, 613-625, 1976.

[40] D. F. Liang, "Comparisons of Nonlinear Recursive Filters for Systems With Non-Negligible Nonlinearities," in *Control and Dynamic Systems, Vol 20,* C. T. Leondes, Ed. New York: Academic Press, 1983.

[41] S. C. Nardone, V. J. Aidala, "Observability Criteria for Bearings-Only Target Motion Analysis," *IEEE Trans Aero and Elect Syst,* AES-17, 162-166, 1981.

[42] V. J. Aidala, "Kalman Filtering Behavior in Bearings-Only Tracking Applications," *IEEE Trans Aero and Elect Syst,* AES-15, 29-39, 1979.

[43] V. Petridis, "A Method for Bearings-Only Velocity and Position Estimation," *IEEE Trans Automatic Control,* AC-26, 488-493, 1981.

[44] V. J. Aidala, S. E. Hammel, "Utilization of Modified Polar Coordinates for Bearings-Only Tracking," *IEEE Trans Automatic Control,* AC-28, 283-294, 1983.

[45] R. S. Bucy, C. Hecht, K. D. Senne, *An Engineer's Guide to Building Nonlinear Filters, Vol I,* DTIC Doc AD-746921, May 1972.

BIBLIOGRAPHY

1. B. D. O. Anderson, J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice Hall, 1979.

2. H. M. Beisner, "Recursive Bayesian Method for Estimating States of Nonlinear System from Sequential Indirect Observations," *IEEE Trans Syst Sci and Cyber,* SSC-3, 101-105, 1967.

3. W. M. Bowles, J. A. Cartelli, "Global Approximation for Nonlinear Filtering with Application to Spreading Spectrum Ranging," in *Control and Dynamic Systems, Vol 19,* C. T. Leondes, Ed. New York: Academic Press, 1983.

4. G. E. P. Box, G. C. Tiao, "A Further Look at Robustness via Bayes' Theorem," *Biometrika,* 49, 419-430, 1962.

5. M. A. Budin, "Parametric Estimation Using Least-Squares Polynomial Smoothing," *IEEE Trans Syst, Man, and Cyber,* SMC-3, 371-381, 1973.

6. C. S. Burrus, T. W. Parks, *DFT/FFT and Convolution Algorithms: Theory and Implementation.* New York: John Wiley & Sons, 1985.

7. Y. T. Chien, K. S. Fu, "On Bayesian Learning and Stochastic Approximation," *IEEE Trans Syst Sci Cyber,* SSC-3, 28-38, 1967.

8. B. DeFinetti, *Theory of Probability: A Critical Introductory Treatment, Vol 1.* New York: John Wiley & Sons, 1974.

9. I. R. Goodman, "Some Results on the Additive Nonlinear Discrete Filtering Problem," in *Sixth Symp Nonlinear Estimation,* 1975, 100-103.

10. G. C. Goodwin, R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis.* New York: Academic Press, 1977.

11. P. J. Huber, *Robust Statistics.* New York: John Wiley & Sons, 1981.

12. IEEE Acoustics, Speech, and Signal Processing Society, Digital Signal Processing Committee, *Programs for Digital Signal Processing.* New York: IEEE Press, 1979.

13. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.

14. W. Kasemratanasunti, R. L. Klein, "Implementation of Nonlinear Estimators Using Quadrature Formulae," *Int J Control*, 20, 835-848, 1974.

15. R. L. Klein, W. Kasemratanasunti, "Stochastic Control Using Quadrature Estimators," in *Fourth Symp Nonlinear Estimation*, 1973, 161-166.

16. S. Kullback, *Information Theory and Statistics*. New York: John Wiley & Sons, 1959.

17. D. G. Lainiotis, "Optimal Nonlinear Estimation," *Int J Control*, 14, 1137-1148, 1971.

18. D. F. Liang, "Exact and Approximate State Estimation Techniques For Nonlinear Dynamic Systems," in *Control and Dynamic Systems, Vol 19*, C. T. Leondes, Ed. New York: Academic Press, 1983.

19. D. F. Liang, G. S. Christensen, "Estimation for Discrete Nonlinear Time-Delayed Systems and Measurements with Correlated and Colored Noise Processes," *Int J Control*, 28, 1-10, 1978.

20. T. Lin, S. S. Yau, "Bayesian Approach to the Optimization of Adaptive Systems," *IEEE Trans Sys Sci Cyber*, SSC-3, 77-85, 1967.

21. A. G. Lindgren, K. F. Gong, "Properties of a Bearings-Only Motion Analysis Estimator: An Interesting Case Study in System Observability," in *Proc 12th Asilomar Conf Circuits, Syst, and Comput*, 1978, 265-272.

22. D. V. Lindley, *Introduction to Probability and Statistics from a Bayesian Viewpoint*. Cambridge: Cambridge University Press, 1965.

23. L. Ljung, "On the Consistency of Prediction Error Identification Methods," in *System Identification: Advances and Case Studies*, R. K. Mehra and D. G. Lainiotis, Ed. New York: Academic Press, 1976.

24. L. Ljung, T. Soderstrom, I. Gustavsson, "Counterexamples to General Convergence of a Commonly Used Recursive Identification Method," *IEEE Trans Automatic Control*, AC-20, 643-652, 1975.

25. L. Ljung, "Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems," *IEEE Trans Automatic Control,* AC-24, 36-50, 1979.

26. S. Marcus, "Optimal Nonlinear Estimation for a Class of Discrete-Time Stochastic Systems," *IEEE Trans Automatic Control,* AC-24, 297-302, 1979.

27. R. M. Mesereau, T. C. Speake, "The Processing of Periodically Sampled Multidimensional Signals," *IEEE Trans Acoustics, Speech, and Signal Proc,* ASSP-31, 188-194, 1983.

28. B. Morgan, *An Introduction to Bayesian Statistical Decision Processes.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1968.

29. S. C. Nardone, A. G. Lindgren, K. F. Gong, "Measurements, Information, and Target Motion Analysis," in *Proc 12th Asilomar Conf Circuits, Syst, and Comput,* 1984, 359-365.

30. S. C. Nardone, A. G. Lindgren, K. F. Gong, "Fundamental Properties and Performance of Conventional Bearings-Only Target Motion Analysis," *IEEE Trans Automatic Control,* AC-29, 775-787, 1984.

31. T. L. Oberlin, R. L. Kashyap, "Bayes Decision Rules Based on Objective Priors," *IEEE Trans Sys, Man, Cyber,* SMC-3, 359-364, 1973.

32. S. Ohmatsu, Y. Tomita, T. Soeda, "Optimal Filtering for Discrete-Time Nonlinear Systems," *IEEE Trans Automatic Control,* AC-21, 116-118, 1976.

33. A. V. Openheim, R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

34. K. D. Senne, "Bayes Law Computation: A Survey," in *Fourth Symp Nonlinear Estimation,* 1973, 267-270.

35. S. D. Silvey, *Statistical Inference.* London: Chapman and Hall, 1975.

36. H. W. Sorenson, "An Overview of Filtering and Stochastic Control in Dynamic Systems," in *Control in Dynamic Systems, Vol 12,* C. T. Leondes, Ed. New York: Academic Press, 1976.

37. H. W. Sorenson, "Approximate Solutions of the Nonlinear Filtering Problem," in *IEEE 1977 Conf on Decision and Control,* 1977, 620,625.

38. H. W. Sorenson, *Parameter Estimation: Principles and Problems.* New York: Marcel Dekker, 1980.

39. H. W. Sorenson, "Parameter and State Estimation: Introduction and Interrelation," in *Proc 6th IFAC Symp Ident and Syst Param Est*, 1982, 85-89.

40. F. Z. Unton, "A Method for Accelerating the First-Order Stochastic Approximation Algorithms," *IEEE Trans Automatic Control,* AC-26, 573-575, 1981.

41. K. Watanabe, "Application of Pseudolinear Partitioned Filter to Passive Vehicle Tracking," *Int J Syst Sci,* 15, 959-975, 1984.

42. H. Weiss, J. B. Moore, "Improved Extended Kalman Filter Design for Passive Tracking," *IEEE Trans Automatic Control,* AC-25, 807-811, 1980.

43. A. Zellner, *An Introduction to Bayesian Inference in Econometrics.* New York: John Wiley & Sons, 1971.