COMPLETE

APPLICATION OF EMPIRICAL BAYES DECISION PROCEDURES TO DISCRETE TIME LINEAR FILTERING

BY

N71-19915

S. J. KAMAT

H. F. MARTZ, JR.

FINAL PROJECT REPORT NO. QT-70-001

NASA GRANT NGR 44-011-031

"INVESTIGATION OF AN EMPIRICAL BAYES FILTER FOR USE IN TRAJECTORY ESTIMATION"

DEPARTMENT OF INDUSTRIAL ENGINEERING

DECEMBER, 1970

Texas Tech University

LUBBOCK

ACKNOWLEDGEMENTS

It is with sincere appreciation that the authors wish to thank the National Aeronautics and Space Administration for the support provided under NASA Grant NGR 44-011-031 which has led to the preparation of this final project report. The authors also wish to thank Dr. J. M. Lewallen, Manned Spacecraft Center, Houston, Texas, and Dr. G. H. Born, Jet Propulsion Laboratory, Pasadena, California, for their cooperation and support of this research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

The theory of filtering concerns the estimation of an underlying physical process from observations of it which may be corrupted by "noise". The physical process is characterized as a random process. One example of such a process is the transmission of a signal. During the transmission of a signal process the transmission channel will perturb the signal by introducing an unwanted random process (noise). A transformation of the received signal is required that will remove as much of the noise as possible in order that a better estimate of the information-bearing signal may be obtained. Another example is the tracking of a space vehicle using radar observations from tracking stations. The calculated trajectory of the space vehicle will be perturbed by effects of the planets whose gravitational constants could not be considered in the derivation of the trajectory. The position of the space vehicle at any given time must be estimated from observations of it from the tracking stations.

The model of the physical process is often random differential equation in case of a continuous-time process or a random difference equation in case of a discrete-time process. In case of a continuous-time model the physical process is assumed to evolve continually while in case of a discrete-time model the physical process is assumed to evolve in discrete steps of time. If the random equation is linear in terms of the state of the process the model is referred to as a linear model. A filtering problem with a linear model is defined as a linear filtering problem. This dissertation is concerned with discrete-time linear filtering problems.

Statement of the Problem

A discrete-time linear system is defined by

$$x(t_n) = \Phi(t_n,t_{n-1})x(t_{n-1}) + u(n-1) \qquad (1.1)$$

where x and u are rx1 vectors and Φ is an rxr matrix. The vector x defines the state of the system. In case of the transmission problem the components of x could be the frequency, amplitude and phase of the signal. For the space vehicle tracking problem the first three components of x may be the position coordinates and the next three the velocity components. The vector u is called the state disturbance error. The initial state $x(t_0)$ and the state disturbance error u are assumed to be random vectors. Thus $x(t_n)$ becomes a random vector.

It is normally assumed that

$$E[u(i)] = 0 \tag{1.2}$$

$$E[x(t_0)] = c \tag{1.3}$$

$$\mathrm{cov}[x(t_0)] = P_0 \tag{1.4}$$

$$\mathrm{cov}[u(i)] = Q_i \tag{1.5}$$

where c is the mean vector of $x(t_0)$, and P_0 and Q_i are the covariance matrices of $x(t_0)$ and u(i) respectively. It is also assumed that u_i is independent of u_j, for all $i \neq j$.

The fundamental matrix Φ is the state transformation matrix and satisfies the following relations.

$$\Phi(t_n, t_n) = I \tag{1.6}$$

$$\Phi(t_n, t_k) = \Phi^{-1}(t_k, t_n) \tag{1.7}$$

$$\Phi(t_i, t_n) = \Phi(t_i, t_k)\Phi(t_k, t_n) \ . \tag{1.8}$$

A set of observations linear in components of the state vector of this system is defined by

$$y(n) = H(n)x(t_n) + v(n) \tag{1.9}$$

where y is a px1 observation vector, v is a px1 observation error vector, and H is a pxr matrix relating the state of the system to the observations. The observation error v is assumed to be a random vector. The vector y(n) thus becomes a random vector, since $x(t_n)$ and v(n) are random. It is normally assumed that

$$E[v(i)] = 0 \tag{1.10}$$

$$E[v(i)v^T(j)] = \delta_{ij}R_i \tag{1.11}$$

where δ_{ij} is the Kronecker delta, and R_i is the covariance matrix of v(i). It is also assumed that u_i is independent of v_j, for all i and j.

The problem is to estimate $x(t_k)$, given the observations y(1), y(2), ..., y(n). This estimate is denoted by $\hat{x}(t_k/n)$. Since the vector y(n) is the observation of the system at time t_n, y(1), ..., y(n-1) will be referred to as past data. If it is desired to estimate $x(t_n)$ at the present time t_n, the problem is referred to as a filtering problem. If at time t_n, it is required to predict $x(t_k)$ for some future time $t_k > t_n$, the situation is termed as predication. If it is desired to estimate $x(t_k)$ for some time $t_k < t_n$, the problem is called a smoothing problem. These three situations can be combined and all three problems referred as "estimation" problems [18].

Linearization of a Non-Linear Problem

Generally in space navigation the set of equations describing the system and the observations are non-linear. Following is a brief discussion of transformation of such a problem to a linear one. Consider a non-linear problem of the form

$$x(t_n) = g(x(t_{n-1}),n-1) + u(n-1) \qquad (1.12)$$

$$z(n) = f(x(t_n),n) + v(n) \qquad (1.13)$$

where x, u and v are interpreted as before and z is the observation vector. By expanding g and f in a Taylor series about a nominal $x(t_n)$ the above problem is transformed to

$$x_s(t_n) = G(t_n)x_p(t_{n-1}) + u(n-1) \qquad (1.14)$$

$$y_p(n) = F(n)x_p(t_n) + v(n) \qquad (1.15)$$

where

$$x_p(t_n) = x(t_n) - x_{nom}(t_n) \qquad (1.16)$$

$$x_s(t_n) = x(t_n) - g(x_{nom}(t_{n-1}),n-1) \qquad (1.17)$$

$$y_p(n) = y(n) - f(x_{nom}(t_n),n) \qquad (1.18)$$

$$G(t_n) = \frac{\partial}{\partial x} g(x,n-1)\Big|_{x=x_{nom}(t_{n-1})} \qquad (1.19)$$

$$F(n) = \frac{\partial}{\partial x} f(x,n)\Big|_{x=x_{nom}(t_n)} \tag{1.20}$$

The terms of order higher than one in the expansions of g and f have been ignored.

Survey of Previous Literature

Wiener [33] and Kolmogorov [21] solved the problem of providing the estimate of a random signal process on the basis of observation of it additively corrupted by noise. This solution was dependent on the assumption of stationarity, ergodicity, and knowledge of the entire past of the observed process. The end result was the specification of the weighting function of the optimal estimator as a solution of the Wiener-Hopf equation. Wiener used spectral factorization to determine the transfer function of the optimal filter in the finite-dimensional case.

Kalman [18], and Bucy and Kalman [12] obtained a solution as an algorithm to produce the numerical estimate from numerical observations, under weaker assumptions than those of Wiener. They relaxed the assumption of stationarity and the necessity of the knowledge of the entire past of the observed process. This theory is known as Kalman-Bucy filtering.

Smith, Schmidt, and McGee [30] were first to recognize the possibility of applying this theory to space navigation and their work was first published in book form by Battin [3]. Bucy, Englar, and Kalman [9], and Kalman and Englar [20] developed an automatic synthesis digital computer program for realization of the estimator in the Kalman-Bucy filter.

Bass, Norum, and Schwartz [2], and Bucy [6] obtained results in non-linear filtering. Bryson and Johansen [5] and Bucy [8] have done research into problems with correlated noise. Bucy [7] and Bucy and Follin [10] consider finite-time filtering problems.

Joseph [17] has done research into sub-optimal filtering using the method of subsystem partitioning. Meditch [25] considers sub-optimal filtering for continuous dynamic processes. Starich [31] used theoretical considerations of the effects of both diagonal variances and off-diagonal covariances of states on the estimation process to stipulate some sub-optimal configurations.

Erzberger [16] studied the application of Kalman-Bucy filtering to aircraft navigation. Smith [29], and Wing and Joseph [34] studied the evolution and estimation of error covariances. Recently Abramson [1] and Shellenbarger [28] have considered some inference problems associated with linear dynamic systems.

Four existing methods of solving the estimation problem are presented below. The "estimation problem" was presented in the first section of this chapter. The basic assumptions for this problem were listed under equations (1.2) to (1.5), (1.10) and (1.11).

<u>Least Squares Method A</u> [11]

Here it is assumed that the state disturbance error is nonexistent; this is the same as annexing the following equation to the basic assumptions.

$$Q_i = 0, \quad i = 0, 1, 2, \ldots \tag{1.21}$$

Using equations (1.6) to (1.8), the state $x(t_i)$ can be written as

$$x(t_i) = \Phi(t_i, t_k)x(t_k). \tag{1.22}$$

Equation (1.9) can be now transformed to

$$y(i) = H(i)\Phi(t_i, t_k)x(t_k) + v(i). \tag{1.23}$$

At time t_n, y(1), y(2), ..., y(n) will be the available observations. Define the following matrix notations

$$\eta_n = \begin{bmatrix} y(1) \\ y(2) \\ \cdot \\ \cdot \\ \cdot \\ y(n) \end{bmatrix}_{np\times 1}$$

$$B_{n,k} = \begin{bmatrix} H(1)\Phi(t_1,t_k) \\ H(2)\Phi(t_2,t_k) \\ \cdot \\ \cdot \\ \cdot \\ H(n)\Phi(t_n,t_k) \end{bmatrix}_{npxr} \qquad (1.25)$$

and

$$\gamma_n = \begin{bmatrix} v(1) \\ v(2) \\ \cdot \\ \cdot \\ \cdot \\ v(n) \end{bmatrix}_{npx1} \qquad (1.26)$$

Using these notations and equation (1.23), following matrix equation is obtained

$$\eta_n = B_{n,k}x(t_k) + \gamma_n. \qquad (1.27)$$

Using the assumptions in equations (1.10) to (1.11), the least squares estimate of $x(t_k)$ is the quantity $\hat{x}(t_k/n)$ which minimizes the scalar quantity

$$[\eta_n - B_{n,k}\hat{x}(t_k/n)]^T[\eta_n - B_{n,k}\hat{x}(t_k/n)]. \qquad (1.28)$$

Thus

$$\hat{x}(t_k/n) = [B^T_{n,k}B_{n,k}]^{-1}B^T_{n,k}\eta_n. \qquad (1.29)$$

The rank of $B_{n,k}$ must equal r for this solution to be unique. Hence a sufficient number of independent observations must have been made so that the rank of $B_{n,k}$ will equal r.

Least Squares Method B [11]

The assumption in equation (1.21) is relaxed under this method. The state $x(t_i)$ can be written as

$$x(t_i) = \Phi(t_i,t_k)x(t_k) - \sum_{j=i}^{k-1} \Phi(t_i,t_{j+1})u(j). \qquad (1.30)$$

Hence the observations $y(i)$ become

$$y(i) = H(i)\{\Phi(t_i,t_k)x(t_k) - \sum_{j=i}^{k-1} \Phi(t_i,t_{j+1})u(j)\} + v(i). \qquad (1.31)$$

Let η_n and $B_{n,k}$ be the same as in equations (1.24) and (1.25) respectively, and let

$$\gamma_n = \begin{bmatrix} v(1) - H(1)\sum_{j=1}^{k-1} \Phi(t\ ,t_{j+1})u(j) \\ v(2) - H(2)\sum_{j=2}^{k-1} \Phi(t_2,t_{j+1})u(j) \\ \vdots \\ v(n) - H(n)\sum_{j=n}^{k-1} \Phi(t_n,t_{j+1})u(j) \end{bmatrix}_{npx1} \qquad (1.32)$$

The matrix equation thus obtained is

$$\eta_n = B_{n,k}x(t_k) + \gamma_n \qquad (1.33)$$

which is of the same form as in equation (1.27). Thus the least squares estimator of $x(t_k)$ is given by

$$\hat{x}(t_k/n) = [B_{n,k}^T B_{n,k}]^{-1} B_{n,k}^T \eta_n \qquad (1.34)$$

which is identical to the one obtained under the previous method. Thus the previous method can be treated as a special case of this method. Sufficient number of independent observations must have been made for this estimator to be unique.

Maximum Likelihood Method

Assume that the state disturbance error is non-existent and that the observation error is Gaussian. The joint density of v(k) is given by

$$p(v(k)) = \frac{1}{(2\pi)^{p/2}|R_k|^{1/2}} \exp\{-\tfrac{1}{2} v(k)^T R_k^{-1} v(k)\} \qquad (1.35)$$

where R_k is the covariance matrix of v(k). Assuming that the v(k)'s are uncorrelated as in equation (1.11), and using γ_n as defined in equation (1.26), the joint density of γ_n is given by

$$p(\gamma_n) = \frac{1}{(2\pi)^{np/2}|R_N|^{1/2}} \exp \{- \tfrac{1}{2} \gamma_n^T R_N^{-1} \gamma_n\} \quad (1.36)$$

where R_N is a block-diagonal matrix consisting of R_1, R_2, ..., R_n. Since $\eta_n = B_{n,k}x(t_k) + \gamma_n$,

$$p(\gamma_n;x(t_k)) = \frac{1}{(2\pi)^{np/2}|R_N|^{1/2}} \exp \{- \tfrac{1}{2} (\eta_n - B_{n,k}x(t_k))^T R_N^{-1} (\eta_n - B_{n,k}x(t_k))\}. \quad (1.37)$$

The maximum likelihood technique selects the quantity $\hat{x}(t_k/n)$ which maximuzes $p(\gamma_n;x(t_k))$. This quantity will also maximize the exponent in equation (1.37). Thus the maximum likelihood estimator is given by

$$\hat{x}(t_k/n) = (B_{n,k}^T R_N^{-1} B_{n,k})^{-1} B_{n,k}^T R_N^{-1} \eta_n. \quad (1.38)$$

<u>Kalman-Bucy Filter</u> [12]

This is essentially a Bayesian approach to the problem. To simplify notation, the time index is used as a subscript instead of as an argument of the different vectors involved. Using squared error loss the Bayes estimator for x_k given the observations y_1, y_2, ..., y_k is given by the mean of the posterior density

$$p(x_k/y_1,\ldots y_k) = p(x_k/Y_k) = \frac{p(x_k,Y_k)}{p(Y_k)} . \qquad (1.39)$$

Under the assumptions of independence given by equations (1.4), (1.5), (1.10) and (1.11), equation (1.39) may be simplified as

$$p(x_k/Y_k) = \frac{p(y_k/x_k,Y_{k-1})p(x_k,Y_{k-1})}{p(Y_k)} . \qquad (1.40)$$

Since y_k does not depend on Y_{k-1}, if x_k is given

$$p(y_k/x_k,Y_{k-1}) = p(y_k/x_k). \qquad (1.41)$$

Using equation (1.41), equation (1.40) becomes

$$p(x_k/Y_k) = \frac{p(y_k/x_k)p(x_k,Y_{k-1})}{p(Y_k)} . \qquad (1.42)$$

Using conditional probabilities

$$p(x_k,Y_{k-1}) = p(x_k/Y_{k-1})p(Y_{k-1}) \qquad (1.43)$$

and

$$p(Y_k) = p(y_k/Y_{k-1})p(Y_{k-1}), \qquad (1.44)$$

equation (1.42) can be rewritten as

$$p(x_k/Y_k) = \frac{p(y_k/x_k)p(x_k/Y_{k-1})}{p(y_k/Y_{k-1})} . \qquad (1.45)$$

The three density functions on the right-hand side of equation (1.45) are assumed to be known a priori.

The state disturbance error u_k and the observation error v_k are assumed to be Gaussian. Thus the conditional density function $p(x_k/Y_k)$ is of the form

$$p(x_k/Y_k) = \frac{1}{(2\pi)^{r/2}|U_k|^{1/2}} \exp\{-\frac{1}{2}(x_k-\hat{x}_{k/k})^T U_k^{-1}(x_k-\hat{x}_{k/k})\} \tag{1.46}$$

where

$$\hat{x}_{k/k} = E[x_k/Y_k] \tag{1.47}$$

and

$$U_k = \text{cov}[x_k/Y_k]. \tag{1.48}$$

Define the estimation error by

$$\tilde{x}_{k-1/k-1} = x_{k-1} - \hat{x}_{k-1/k-1} \tag{1.49}$$

$$\tilde{x}_{k/k-1} = x_k - \hat{x}_{k/k-1}. \tag{1.50}$$

It has been shown by Kalman [18] that

$$E[\tilde{x}_{k-1/k-1}] = 0 \tag{1.51}$$

$$E[\tilde{x}_{k/k-1}] = 0. \tag{1.52}$$

Let
$$P_{k-1} = \text{cov}[\tilde{x}_{k-1/k-1}] \tag{1.53}$$

$$\bar{P}_k = \text{cov}[\tilde{x}_{k/k-1}] \tag{1.54}$$

and

$$E[x_k/Y_{k-1}] = \hat{x}_{k/k-1} = \bar{x}_k. \tag{1.55}$$

Since $x_k = \Phi_{k,k-1}x_{k-1} + u_{k-1}$, $\hat{x}_{k/k-1}$ can be obtained as

$$\hat{x}_{k/k-1} = \Phi_{k,k-1}\hat{x}_{k-1/k-1}. \tag{1.56}$$

Equation (1.54) can be rewritten as

$$\bar{P}_k = E[(x_k-\hat{x}_{k/k-1})(x_k-\hat{x}_{k/k-1})^T] \tag{1.57}$$

or,

$$\bar{P}_k = E[(\Phi_{k,k-1}\tilde{x}_{k-1/k-1}+u_{k-1})(\Phi_{k,k-1}\tilde{x}_{k-1/k-1}+u_{k-1})^T]. \tag{1.58}$$

Since $\tilde{x}_{k-1/k-1}$ is independent of u_{k-1}, equation (1.58) simplifies to

$$\bar{P}_k = \Phi_{k,k-1}P_{k-1}\Phi^T_{k,k-1} + Q_{k-1}. \tag{1.59}$$

Now

$$E[y_k/x_k] = H_k x_k \tag{1.60}$$

and

$$\text{cov}[y_k/x_k] = E[(y_k-H_kx_k)(y_k-H_kx_k)^T/x_k] = E[v_kv_k^T/x_k] = R_k. \tag{1.61}$$

Also

$$E[y_k/Y_{k-1}] = H_k\bar{x}_k \tag{1.62}$$

and

$$\text{cov}[y_k/Y_{k-1}] = E[(y_k-H_k\bar{x}_k)(y_k-H_k\bar{x}_k)^T/Y_{k-1}]$$

$$= E[(H_k\tilde{x}_{k/k-1}+v_k)(H_k\tilde{x}_{k/k-1}+v_k)^T].$$

(1.63)

Since $\tilde{x}_{k/k-1}$ is independent of v_k, equation (1.63) reduces to

$$\text{cov}[y_k/Y_{k-1}] = H_k\bar{P}_kH_k^T + R_k. \quad (1.64)$$

Using the above equations, equation (1.45) can now be rewritten as

$$p(x_k/Y_k) = \frac{\left|H_k\bar{P}_kH_k^T+R_k\right|^{1/2}}{(2\pi)^{r/2}\left|R_k\right|^{1/2}\left|P_k\right|^{1/2}} \exp\{-\frac{1}{2}[(y_k-H_kx_k)^T$$

$$R_k^{-1}(y_k-H_kx_k) + (x_k-\bar{x}_k)^T\bar{P}_k^{-1}(x_k-\bar{x}_k) -$$

$$(y_k-H_k\bar{x}_k)^T(H_k\bar{P}_kH^T+R_k)^{-1}(y_k-H_k\bar{x}_k)]\} \quad (1.65)$$

Equation (1.65) can be simplified to the same form as equation (1.46), and upon comparison of the two, the following equations are obtained.

$$\hat{x}_k = \bar{x}_k + \bar{P}_kH_k^T(H_k\bar{P}_kH_k^T+R_k)^{-1}(y_k-H_k\bar{x}_k) \quad (1.66)$$

$$P_k = \bar{P}_k - \bar{P}_kH_k^T(H_k\bar{P}_kH_k^T+R_k)^{-1}H_k\bar{P}_k. \quad (1.67)$$

Equations (1.56), (1.59), (1.66), and (1.67) are known as the Kalman-Bucy filter (or Kalman filter). To start the estimation process the values x_0 and P_0 are assumed known a priori. The solution for the prediction problem is given by

$$\hat{x}_{k/n} = \Phi_{k,n}\hat{x}_{n/n}, \quad k>n \qquad (1.68)$$

whereas for the smoothing problem

$$\hat{x}_{k/n} = \hat{x}_{k/k}, \quad k<n. \qquad (1.69)$$

Discussion of these Methods

Of the four methods presented above for the solution of the linear filtering problem, the Kalman filter is usually the best one. The least squares method does not weigh the observations according to their relative accuracy, but treats them equally. The maximum likelihood method ignores the state disturbance error. Since the state disturbance error gets propagated into the system with the passage of time, it may seriously impair the performance of the maximum likelihood method. Hence the least squares and the maximum likelihood methods are usually inferior to the Kalman filter.

Although Kalman [18] justifies the assumption of Gaussian state disturbance error, the advantage of this assumption is seen in the simplification of the mathematical manipulation of density functions. If a real physical process can be exactly described by a linear system, the justification of Gaussian noise is reasonable. However, if the linear system is more of an approximation to a real physical process it is possible that the state disturbance error will not be Gaussian. For example, if a non-linear system was the exact representation of the real physical process, then the linear system under consideration has been obtained by linearization as discussed in an earlier section. The state disturbance error then consists of the random error as well as the higher order terms which were ignored. In this case there is no physical or logical justification for assuming this error to be Gaussian.

This dissertation relaxes the assumption of any form of distribution for the state disturbance error. A filter is developed which does not depend on the form of the distribution. The observation error is assumed to be Gaussian as in the Kalman filter. This may be justified by recognizing that the observation error is caused by an accumulation of microscopic errors introduced by different components of the instruments used for observation. By the Central Limit Theorem, the observation error will tend to be distributed normally.

Purpose and Outline of Subsequent Chapters

In Chapter II a smooth empirical Bayes estimator will be developed for the mean of a multivariate normal distribution. The estimates obtained by using this estimator will be compared against the maximum likelihood estimates, by using Monte Carlo simulation. Average squared error of the extimates will be used as the measure of performance of the corresponding estimator. In Chapter III this smooth empirical Bayes estimator will be used to estimate the state vector of the linear system with linear observations. The performance of this empirical Bayes filter will be compared with the performance of the Least Squares filter and the performance of the Kalman filter, by using Monte Carlo simulation. Conclusions will be summarized in Chapter IV about the properties and limitations of the smooth empirical Bayes estimator and areas for future research will be discussed.

CHAPTER II

A SMOOTH EMPIRICAL BAYES ESTIMATOR FOR THE MEAN OF A MULTIVARIATE NORMAL DISTRIBUTION

In this chapter a smooth empirical Bayes estimator will be obtained for the mean of a multivariate normal random process. This estimator will then be used in Chapter III to solve the estimation problem discussed in Chapter I.

Let x be a random p-vector which is normally distributed with unknown mean vector θ and known covariance matrix R. Let θ have an unknown and unspecified density function $g(\theta)$ which is time-invariant. This density function is usually referred to as the prior density. Consider that at time n, nature chooses a value θ_n according to $g(\theta)$ and using this value as the mean of the normal distribution, then chooses the vector x_n according to $f(x|\theta_n,R)$. Consider $x_1, \ldots, x_n$ to be sequence of such vectors over time. Thus x_1 is distributed normally with mean θ_1, x_2 is distributed normally with mean θ_2, etc. Using a squared error loss function, the Bayes estimator for θ_n is given by $E[\theta|x_n]$ which can be expressed as

$$E[\theta|x_n] = \frac{\int \theta f(x_n|\theta)g(\theta)d\theta}{\int f(x_n|\theta)g(\theta)d\theta} . \tag{2.1}$$

The integration in equation (2.1) is carried out over the entire range of θ. Since $g(\theta)$ is assumed to be unknown, a method to estimate $g(\theta)$ is desired.

Cacoullos [13] considered a class of estimates $g_n(\theta)$ of $g(\theta)$ of the form

$$g_n(\theta) = \frac{1}{nh^P(n)} \sum_{i=1}^{n} K\left[\frac{\theta-\theta_i}{h(n)}\right] \tag{2.2}$$

where $K(y)$ is a kernel which is chosen to satisfy suitable conditions and $h(n)$ is a sequence of positive constants which satisfy

$$\lim_{n\to\infty} h(n) = 0. \tag{2.3}$$

He showed that this class of density estimators is asymptotically unbiased and consistent in quadratic mean if the following conditions are satisfied.

$$\sup_y |K(y)| < \infty \tag{2.4}$$

$$\int |K(y)|dy < \infty \tag{2.5}$$

$$\lim_{||y||\to\infty} ||y||^P K(y) = 0 \tag{2.6}$$

$$\int K(y)dy = 1 \tag{2.7}$$

$$\lim_{n\to\infty} nh^P = 0. \tag{2.8}$$

The kernel

$$K(y) = \frac{1}{(2\pi)^{p/2}} \exp\left[-\frac{1}{2} y^T y\right] \tag{2.9}$$

and the constant

$$h(n) = n^{-1/25} \tag{2.10}$$

satisfy the conditions in equations (2.3) to (2.9). Usually the argument y of the kernel may have components with different units. In this case a different type of kernel can be defined such that it will allow each element of the difference vector to be divided by a different h in equation (2.2). The constant h_j for the j-th element as defined below was suggested by Bennett [4] for a univariate case.

$$h_j = n^{-1/25} \{\text{Std. Dev. } (\theta_{ij})\}.$$

The density function $g(\theta)$ can then be estimated by $g_n(\theta)$, where

$$g_n(\theta) = \frac{1}{nh^P(2\pi)^{p/2}} \sum_{i=1}^{n} \exp\left\{-\frac{1}{2h^2}(\theta-\theta_i)^T(\theta-\theta_i)\right\}, \quad -\infty<\theta<\infty \tag{2.11}$$

provided $\theta_1, \ldots, \theta_n$ are known exactly.

Replacing $g(\theta)$ by $g_n(\theta)$ equation (2.1) becomes

$$E_n[\theta|x_n] = \frac{\int \theta f(x_n|\theta) g_n(\theta) d\theta}{\int f(x_n|\theta) g_n(\theta) d\theta} . \qquad (2.12)$$

Since the density function $f(x_n|\theta)$ is given by

$$f(x_n|\theta) = \frac{1}{(2\pi)^{p/2}|R|^{1/2}} \exp \{- \frac{1}{2} (x_n-\theta)^T R^{-1}(x_n-\theta)\}, \qquad (2.13)$$

$f(x_n|\theta)g_n(\theta)$ can be written as

$$f(x_n|\theta)g_n(\theta) = \frac{1}{nh^p(2\pi)^p|R|^{1/2}} \sum_{i=1}^{n} \exp \{-\frac{1}{2h^2}(\theta-\theta_i)^T(\theta-\theta_i) -\frac{1}{2}(x_n-\theta)^T R^{-1}(x_n-\theta)\}. \qquad (2.14)$$

Now

$$\frac{1}{h^2} (\theta-\theta_i)^T(\theta-\theta_i) + (x_n-\theta)^T R^{-1}(x_n-\theta)$$

$$= \frac{1}{h^2} \theta^T\theta - \frac{2}{h^2} \theta^T\theta_i + \frac{1}{h^2} \theta_i^T\theta_i + x_n^T R^{-1} x_n - 2x_n^T R^{-1}\theta + \theta^T R^{-1}\theta$$

$$= \theta^T(R^{-1}+\frac{1}{h^2} I)\theta - 2\theta^T(R^{-1}x_n+\frac{1}{h^2}\theta_i) + x_n^T R^{-1} x_n + \frac{1}{h^2}\theta_i^T\theta_i . \qquad (2.15)$$

Let

$$P = R^{-1} + \frac{1}{h^2} I \qquad (2.16)$$

and

$$q_i = R^{-1}x_n + \frac{1}{h^2}\theta_i \qquad (2.17)$$

Then

$$f(x_n|\theta)g_n(\theta) = \frac{1}{nh^p(2\pi)^p|R|^{1/2}} \sum_{i=1}^{n} \exp -\frac{1}{2}[(\theta-p^{-1}q_i)^T$$

$$P(\theta-P^{-1}q_i) - q_i^T P^{-1}q_i + x_n^T R^{-1}x_n +$$

$$\frac{1}{h^2}\theta_i^T\theta_i] \quad . \qquad (2.18)$$

Now

$$r(\theta) = \frac{|P|^{1/2}}{(2\pi)^{p/2}} \exp\{-\frac{1}{2}(\theta-P^{-1}q_i)^T P(\theta-P^{-1}q_i)\} \qquad (2.19)$$

is a normal density function with the mean vector $P^{-1}q_i$ and the covariance matrix P^{-1}. Hence

$$\int\theta f(x_n|\theta)g_n(\theta)d = \frac{P^{-1}}{nh^p(2\pi)^{p/2}|R|^{1/2}|P|^{1/2}} \sum_{i=1}^{n} q_i \exp$$

$$[\frac{1}{2} q_i^T P^{-1}q_i - \frac{1}{2} x_n^T R^{-1}x_n - \frac{1}{2h^2}\theta_i^T\theta_i],$$

$$(2.20)$$

and

$$\int f(x_n|\theta)g_n(\theta)d\theta = \frac{1}{nh^p(2\pi)^{p/2}|R|^{1/2}|P|^{1/2}} \sum_{i=1}^{n} \exp$$

$$[\tfrac{1}{2}\, q_i^T P^{-1} q_i - \tfrac{1}{2}\, x_n^T R^{-1} x_n - \frac{1}{2h^2}\, \theta_i^T \theta_i]. \tag{2.21}$$

$$E_n(\theta|x_n) = \frac{P^{-1} \sum_{i=1}^{n} q_i \exp\, [\tfrac{1}{2}\, q_i^T P^{-1} q_i - \frac{1}{2h^2}\, \theta_i^T \theta_i]}{\sum_{i=1}^{n} \exp\, [\tfrac{1}{2}\, q_i^T P^{-1} q_i - \frac{1}{2h^2}\, \theta_i^T \theta_i]} \tag{2.22}$$

Equations (2.10), (2.16), (2.17) and (2.22) would give $E_n(\theta|x_n)$ if $\theta_1, \ldots, \theta_n$ were known exactly. In this case $E_n(\theta|x_n)$ would be a consistent estimator for $E(\theta|x)$. However, the values $\theta_1, \ldots, \theta_n$ remain unknown and estimates of these values need to be used. If $\tilde{\theta}_i$ is a consistent estimate of θ_i, $E_n(\theta|x_n)$ will be consistent. Since the distribution of θ's is unknown, also, as a first step $\tilde{\theta}_i$ will be taken as x_i, $i=1, \ldots, n$. In the next section a slightly modified approach will be used for obtaining the values of $\tilde{\theta}_i$'s. Using $\tilde{\theta}_i = x_i$ equation (2.17) is replaced by

$$q_i^{(1)} = R^{-1} x_n + \frac{1}{h^2}\, x_i. \tag{2.23}$$

The estimator for θ_n is then given by

$$E_n^{(1)}(\theta|x_n) = P^{-1}\sum_{i=1}^{n} q_i^{(1)} \exp\left[\frac{1}{2} q_i^{(1)T}P^{-1}q_i^{(1)} - \frac{1}{2h^2} x_i^T x_i\right] \Big/ \sum_{i=1}^{n} \exp\left[\frac{1}{2} q_i^{(1)T}P^{-1}q_i^{(1)} - \frac{1}{2h^2} x_i^T x_i\right]. \quad (2.24)$$

The estimates $E_i^{(1)}(\theta|x_i)$ obtained as a result of using $\tilde{\theta}_i=x_i$, can now be used in place of $\tilde{\theta}_i$ to obtain $E_i^{(2)}(\theta|x_i)$. This process of using the estimates could be iterated and $E_i^{(k-1)}(\theta|x_i)$ could be used as the value of $\tilde{\theta}_i$ to obtain the extimates $E_i^{(k)}(\theta|x_i)$. Using a smooth estimator with different kernel, Bennett [4] found that the first iteration normally results in an improved estimate; however, further iteration produces estimates not as good as the one obtained with one iteration. In a later section, the results of Monte Carlo simulation are presented which includes the comparison of the performances of $E_i^{(1)}(\theta|x_i)$ and $E_i^{(2)}(\theta|x_i)$.

Covariance Correction

If $\tilde{\theta}_i$'s had the following two properties

$$E[\tilde{\theta}_i] = E[\theta_i] \quad (2.25)$$

$$\mathrm{cov}[\tilde{\theta}_i] = \mathrm{cov}[\theta_i] \qquad (2.26)$$

Then the distributions of $\tilde{\theta}_i$'s and θ_i's will have matching first two moments. This will give a second order match between the two distributions, which should bring the estimator closer to the unknown Bayes estimator.

Now

$$\mathrm{cov}[\tilde{\theta}_i] = \mathrm{cov}_\theta(E[\tilde{\theta}_i|\theta_i]) + E_\theta[\mathrm{cov}(\tilde{\theta}_i|\theta_i)]. \qquad (2.27)$$

For $\tilde{\theta}_i = x_i$,

$$E[\tilde{\theta}_i] = E_\theta(E[x_i|\theta_i]) = E[\theta_i] \qquad (2.28)$$

and

$$\mathrm{cov}[\tilde{\theta}_i] = \mathrm{cov}[\theta_i] + \mathrm{cov}[x_i|\theta_i] \qquad (2.29)$$

since $E[\tilde{\theta}_i|\theta_i] = \theta_i$, and since $\mathrm{cov}[x_i|\theta_i]$ is not a function of θ_i. Recall that

$$\mathrm{cov}[x_i|\theta_i] = R \qquad (2.30)$$

which is assumed to be known.

Equations (2.28) and (2.29) show that $\tilde{\theta}_i = x_i$ has the property in equation (2.25), but not the one in equation (2.26). Hence a linear transformation of the form $\theta_i^* = A\tilde{\theta}_i + b$ is desired so that θ_i^* will possess both the properties given in equations (2.25) and (2.26), where A is a pxp matrix and b is a px1 vector. Now

$$E[\theta_i^*] = AE[\tilde{\theta}_i] + b = AE[\theta_i] + b. \qquad (2.31)$$

Hence

$$b = (I-A)E[\theta_i] \qquad (2.32)$$

will guarantee that θ_i^* will possess the first property for any matrix A. Also

$$\mathrm{cov}[\theta_i^*] = A\ \mathrm{cov}[\tilde{\theta}_i]\ A^T$$

$$= A\ [\mathrm{cov}[\theta_i] + R]\ A^T. \qquad (2.33)$$

Since the covariance matrices involved in equation (2.33) are symmetric, the matrix A is required to be a symmetric matrix so that all the elements of A can be uniquely determined from equation (2.33). Even with this restriction the resulting equations are complex quadratic equations which cannot be solved analytically without additional restrictions. One possible solution is obtained by requiring the matrix A to be of diagonal form. The elements of A are then found by using the relation between the diagonal elements of matrices on the two sides of equation (2.33); namely

$$A_{jj} = \{[\mathrm{cov}(\theta_i)]_{jj}/[\mathrm{cov}(\theta_i) + R]_{jj}\}^{1/2}. \qquad (2.34)$$

where $[\ ]_{jj}$ refers to the j-th diagonal element of the indicated matrix.

Since $cov(\theta_i)$ is unknown, it must be estimated from the observations. The matrix $cov(\tilde{\theta}_i)$ can be estimated from the observed values x_i, i=1, ..., n. Since R is known, $cov(\theta_i)$ can be estimated by using equation (2.29). Sampling error may cause some diagonal element of A to be negative when n is small (less than five). The particular element will then be set equal to one.

Time-Dependent Covariance Matrix

Suppose now that the restriction that R remain constant over time is relaxed. Let x_n be normally distributed with mean θ_n and known covariance matrix R_n. Equation (2.1) will still hold if $f(x_n|\theta)$ is replaced by $f_n(x_n|\theta)$ to account for changing R_n. Mathematical manipulation similar to that in the first section of this chapter yields the following equations for the estimator for θ_n.

$$P_n = R_n^{-1} + \frac{1}{h^2} I \qquad (2.35)$$

$$q_{in} = R_n^{-1} x_n + \frac{1}{h^2} \theta_i \qquad (2.36)$$

$$E_n(\theta|x_n) = P_n^{-1} \sum_{i=1}^{n} q_{in} \exp\left[\frac{1}{2} q_{in}^T P_n^{-1} q_{in} - \frac{1}{2h^2} \theta_i^T \theta_i\right] \Big/$$

$$\sum_{i=1}^{n} \exp\left[\frac{1}{2} q_{in}^T P_n^{-1} q_{in} - \frac{1}{2h^2} \theta_i^T \theta_i\right] . \tag{2.37}$$

The values $\tilde{\theta}_i = x_i$, i=1, ..., n must be used in place of θ_i since θ_i's are unknown. The covariance correction developed in the previous section can be used in this case also, with the conditional covariance matrix in equation (2.30) being

$$\text{cov}[x_i|\theta_i] = R_i . \tag{2.38}$$

Discussion of the Estimator

Three different estimators have been presented in the previous sections, namely $E_n^{(1)}(\theta|x_n)$, $E_n^{(2)}(\theta|x_n)$, and $E_n^{(1)*}(\theta|x_n)$, the third one being obtained using the covariance correction. These estimators can be used for either the case with a constant covariance matrix R, or the case with a changing covariance matrix R_n. It would be desirable to consider the distributional properties of these estimators analytically and obtain the expected values and covariance matrices of their estimation errors. The estimation error of an estimator is defined by the difference between the true vector and the estimated vector. This type of analysis would also help in establishing confidence intervals about these estimators. The prior distribution $g(\theta)$ is required to be of a known form to perform this analysis. But, this would defeat the purpose of the development of the smooth empirical Bayes estimator, which was to relax the distributional assumptions on the prior distribution. This prevents any further analytical computation of the expected values and covariance matrices of the estimation errors for these estimators.

An alternative method is to assume the prior distribution to be of a given form. It may be possible to obtain the distributional properties of the estimators with this assumption; however, a different analysis would need to be performed for each distribution, since the results obtained by this analysis can be expected to change with a change in the assumed prior distribution. Also, due to the presence of exponential terms in the summations in the expressions of the estimators, with the exponents being differences of quadratic forms, the computations can be very time consuming and complicated.

Another alternative method of analysis is Monte Carlo simulation. This is done by using a different shaped prior distribution for each run of the simulation. By using a number of replications in the simulation, a measure of the expected value and the covariance matrix of the estimation error can be obtained from the averaged results.

The first method of analysis is to be preferred over the other two methods, if it can be performed. Since in the present case it is not possible to perform such an analysis due to the lack of knowledge of the prior distribution, the second method of analysis would be the next choice. However, this second method being a very time consuming process, the third method of Monte Carlo simulation is used in this chapter to study the distributional properties of the estimation error.

Results of Monte Carlo Simulation

Appendix A describes the computer program which is used to compare the performance of the estimators developed in this chapter with the performance of the maximum likelihood estimator, since it is an unbiased and consistent estimator for the mean of a multivariate normal distribution with known distributional properties. For convenience the individual elements of θ_i were separately generated as independently distributed random variables from a member of the Pearson family of distributions. At observation stage i, ten individual observations were generated. The mean of these ten observations is the maximum likelihood estimator and is designated as

x_i, the single observation at stage i required for the smooth empirical Bayes estimator. This process is used to simulate fifty stages of observations.

The expected value of the squared error matrix for the maximum likelihood estimator will be $R = R_1/10$ where R_1 is the covariance matrix used for generating the ten observations at each stage. For the smooth empirical Bayes estimator, the average squared error matrix is obtained by averaging over one hundred replications and this is used as an estimate of the expected covariance matrix of the estimation error. Martz and Krutchkoff [24] developed an empirical Bayes estimator for the mean of a multivariate normal distribution. They found that the performance of that estimator depended on a summary quantity defined by

$$z_j = \frac{[R - (\phi^{-1} + R^{-1})^{-1}]_{jj}}{[(\phi^{-1} + R^{-1})^{-1}]_{jj}} \qquad (2.39)$$

where ϕ is the covariance matrix of θ used in the simulation and R is the conditional covariance matrix of the sample mean. This summary quantity is used here to index the different sets of parameters used in the simulation. The dimension of x is chosen to be $p = 6$ for the simulation, since in the next chapter the filter developed using this estimator will be applied to a trajectory estimation problem, which as mentioned in Chapter I usually has a six-dimensional state vector. An average z quantity is defined to index the different sets of data simulated, as

$$z = \frac{1}{6} \sum_{j=1}^{6} z_j. \qquad (2.40)$$

The performance ratio is defined as the ratio of the trace of the average squared error matrix of the smooth empirical Bayes estimator to the trace of R, which is the expected squared error matrix of the maximum likelihood estimator. The estimator $E_n^{(1)}(\theta|x_n)$ is referred to as the first iteration. When the observations used for this estimator are corrected using the covariance correction, the estimator is termed as the first iteration with covariance correction. The estimator $E_n^{(2)}(\theta|x_n)$ is called the second iteration.

Figures 1 to 11 present the performance ratios as functions of the number of observation stages. Figures 1 and 2 give the performance of the first iteration, first iteration with covariance correction, and the second iteration, for z = 1.167 and z = 2.217 respectively. In both these figures it can be seen that the second iteration is better than the first iteration and that the first iteration with covariance correction is significantly better than the first and second iterations.

Figure 3 presents the performance of the first iteration for five different values of z, namely, 0.6, 1.167, 2.217, 2.904 and 4.267 and the performance of this first iteration can be seen to improve with increase in z. Figures 4 and 5 give the performances of the first iteration with covariance correction and the second iteration respectively for the same five values of z. They show that the performances of the first iteration with covariance correction and the second iteration also improve with increase in z.

To observe the effect of a different functional value for the positive constant h, the three estimates are obtained using h = 1. Figure 6 compares the performances of the first iteration for h=1 and $h=n^{-1/25}$. Figures 7 and 8 give the same comparison for the first iteration with covariance correction and the second iteration respectively. From these three figures it can be seen that all three estimators are rather insensitive to small changes in h.

The results presented in Figures 1 to 8 are obtained using a J-shaped distribution for the simulation of the individual elements of θ. Figure 9 presents the performances of the first iteration with covariance correction for six different shaped distributions used for the individual elements of θ. The performance is seen to be distinctly better for a U-shaped distribution, while being almost identical for the remaining five shaped distributions. A U-shaped distribution on the elements of θ makes the prior $g(\theta)$ be non-unimodal, while for the other five shaped distributions, $g(\theta)$ becomes unimodal.

Since the smooth empirical Bayes estimator uses the observations for all past stages, the process of using these estimators may become considerably time consuming and may require substantially large memory, as the number of stages increases. A finite memory estimator that uses only k past stages of observations can be defined as

$$E_n(\theta|x_n) = P^{-1} \sum_{i=n-k+1}^{n} q_i \exp\left\{\frac{1}{2} q_i^T P^{-1} q_i - \frac{1}{2h^2} \theta_i^T \theta_i\right\} / \sum_{i=n-k+1}^{n} \exp\left\{\frac{1}{2} q_i^T P^{-1} q_i - \frac{1}{2h^2} \theta_i^T \theta_i\right\} . \tag{2.41}$$

Figures 10 and 11 present the performance of the first iteration with covariance correction with finite memory for five different values of k and for z=1.167, and z=2.217 respectively. It can be seen that the performance improves with increase in k. But, this improvement shows a saturation effect as k increases. Hence depending on the memory restriction and the cost of additional computation time an optimum value of k can be chosen so that the improvement in performance is a maximum subject to the above restrictions. Once a particular value of k is chosen, the finite memory estimator can be used for $n>k$ while using the total memory estimator for $n\leq k$.

Conclusions

A smooth empirical Bayes estimator has been obtained for the mean of a multivariate normal random process in three different forms; namely, the first iteration, the first iteration with covariance correction and the second iteration. Using Monte Carlo simulation the performance of these three forms of the estimator is compared with the performance of the maximum likelihood estimator. All these three forms of the estimator are found to perform better than the maximum likelihood estimator for the particular data used in the simulation. The first iteration with covariance correction is observed to be better than the other two forms of the estimator, and will be used in the next chapter for application to the problem of estimating the state vector of a discrete time linear system with a linear set of observations. No confidence intervals could be obtained for any of the three forms of the estimator since a direct method of analysis could not be used to investigate the statistical properties of the estimator.

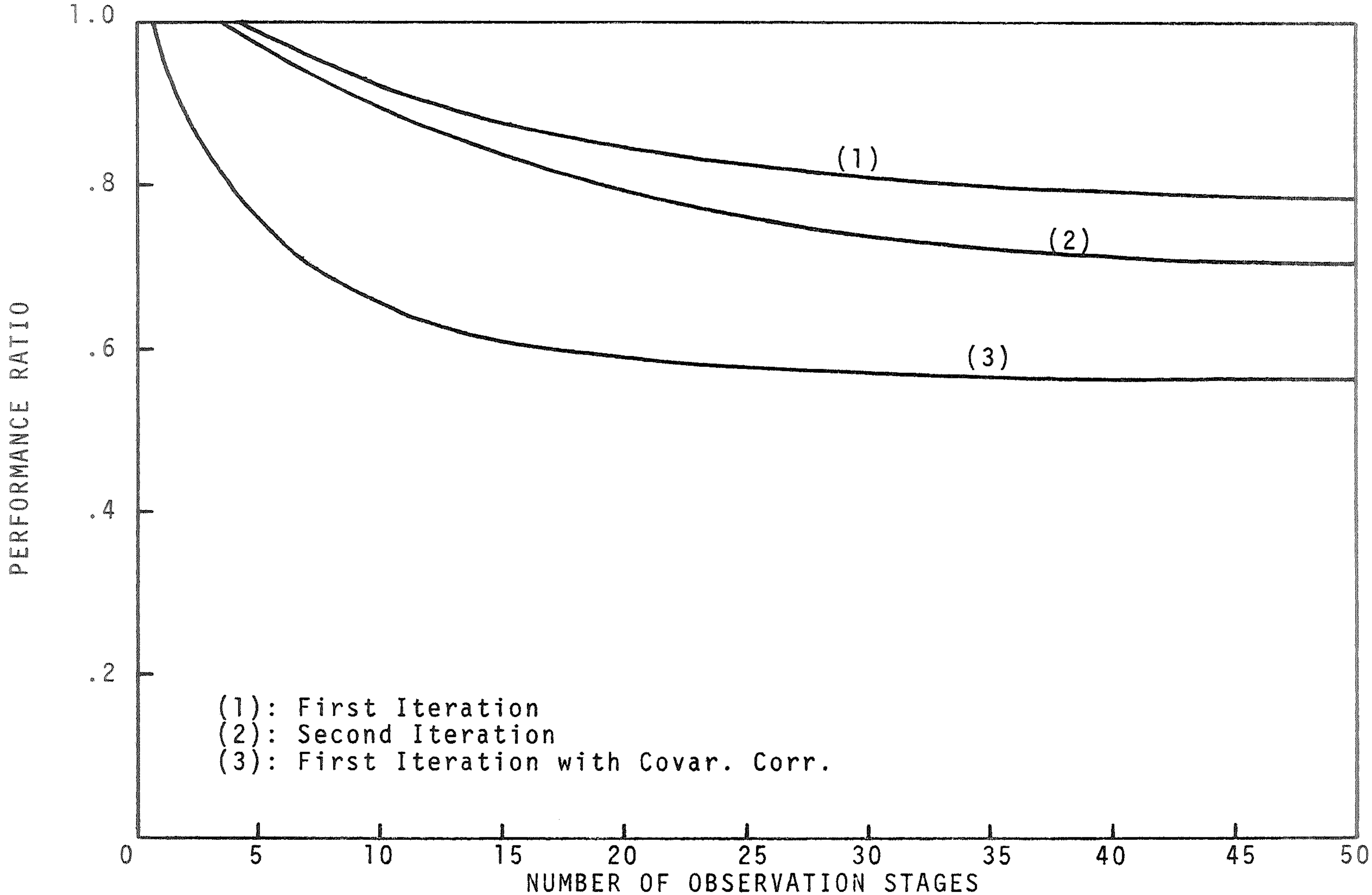


Figure 1. Performance Ratios for First Iteration, First Iteration with Covariance Correction, and Second Iteration, for z=1.167.

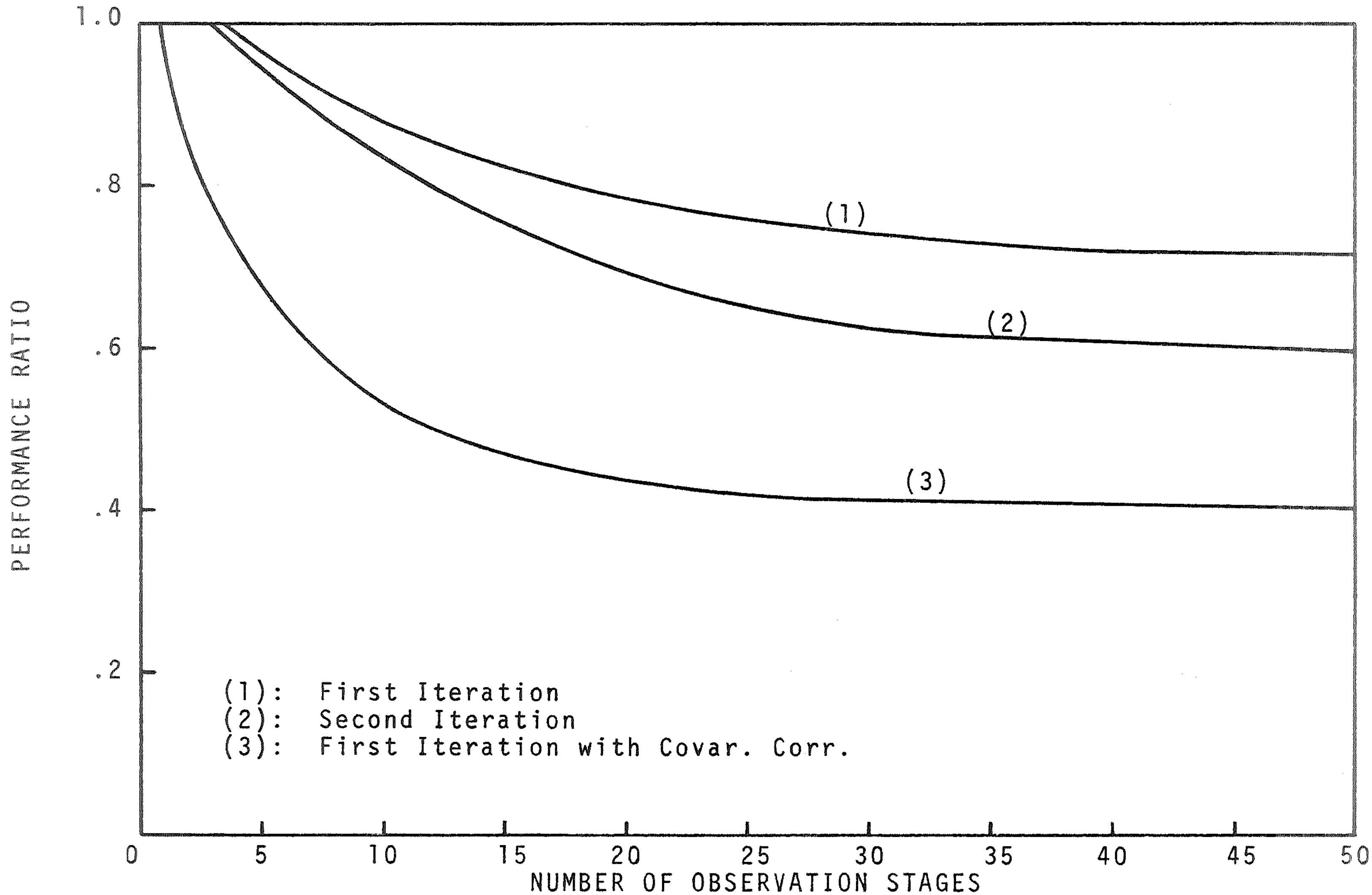


Figure 2. Performance Ratios for First Iteration, First Iteration with Covariance Correction, and Second Iteration, for $z=2.217$.

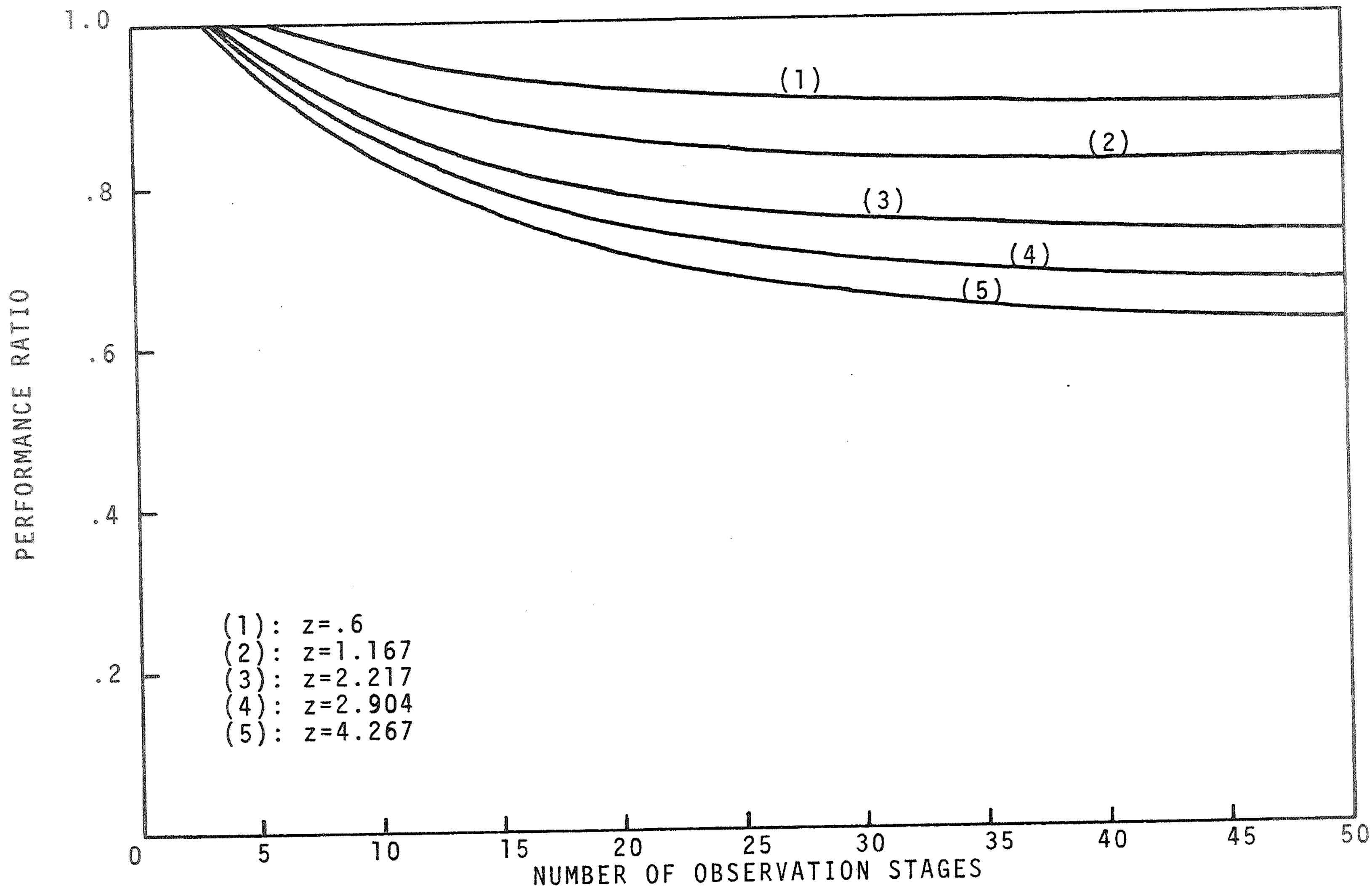


Figure 3. Performance Ratios for First Iteration for z=.6, 1.167, 2.217, 2.904 and 4.267.

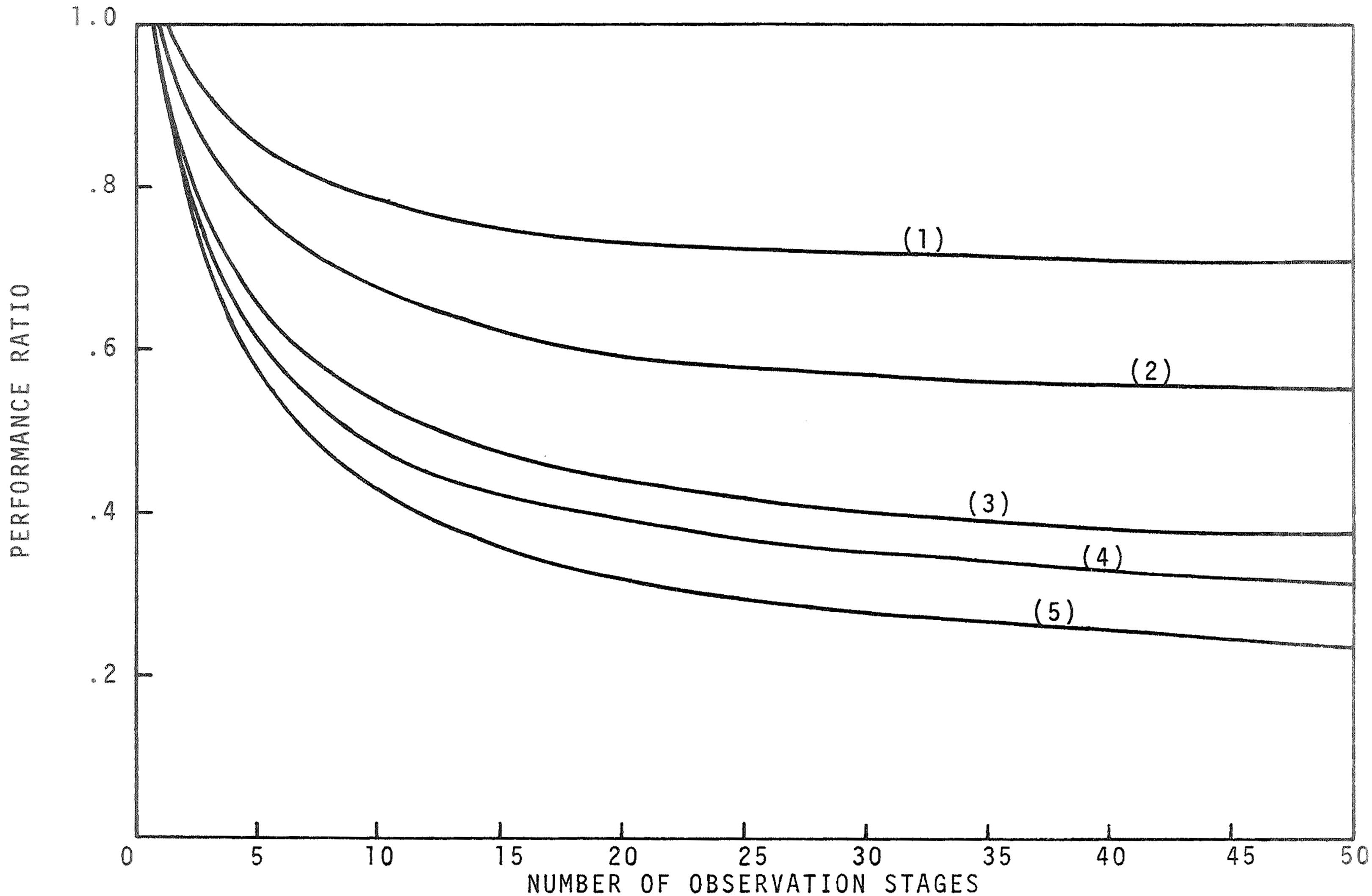


Figure 4. Performance Ratios for First Iteration with Covariance Correction for z=.6, 1.167, 2.217, 2.904 and 4.267.

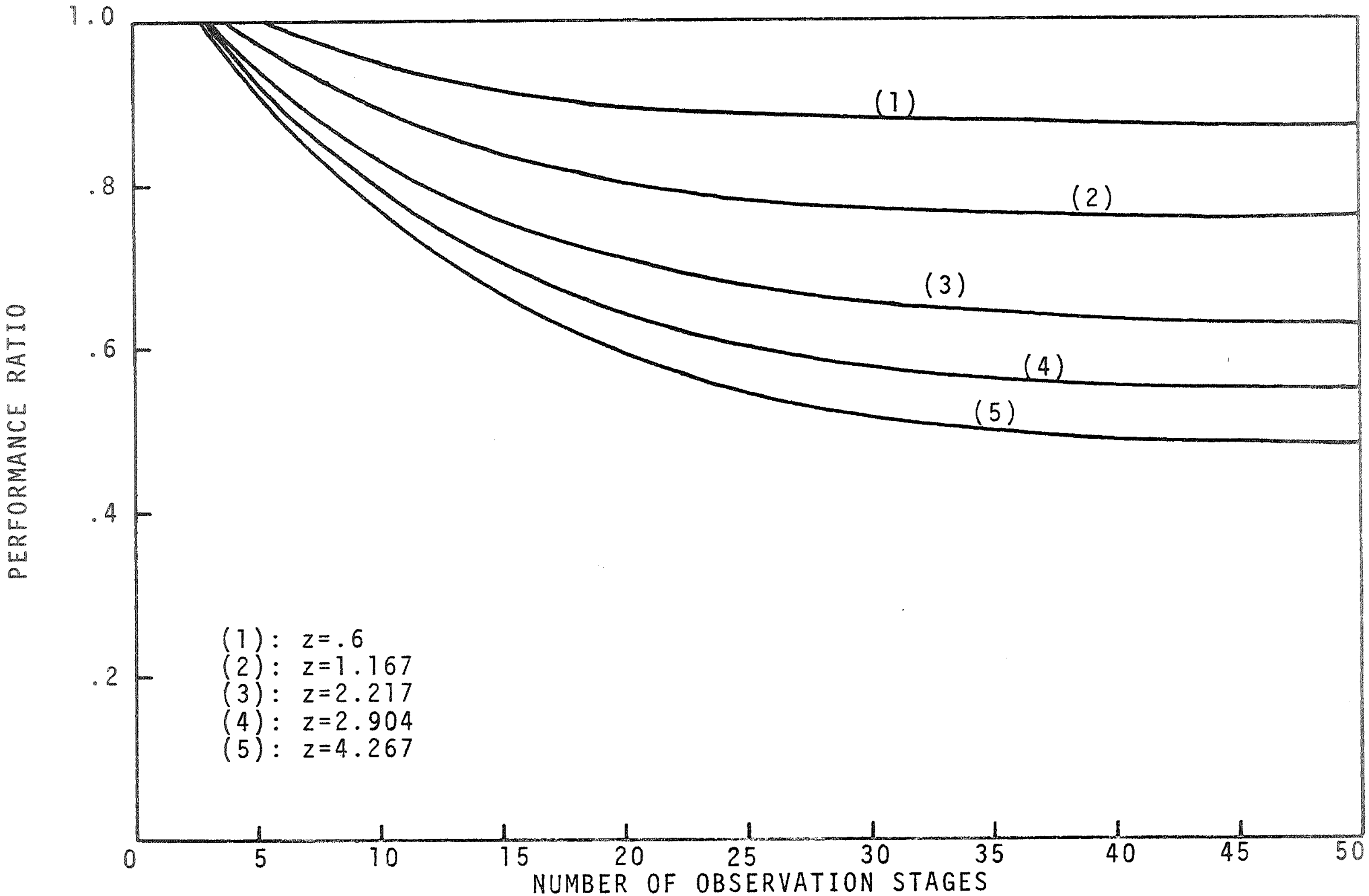


Figure 5. Performance Ratios for Second Iteration for z=.6, 1.167, 2.217, 2.904 and 4.267.

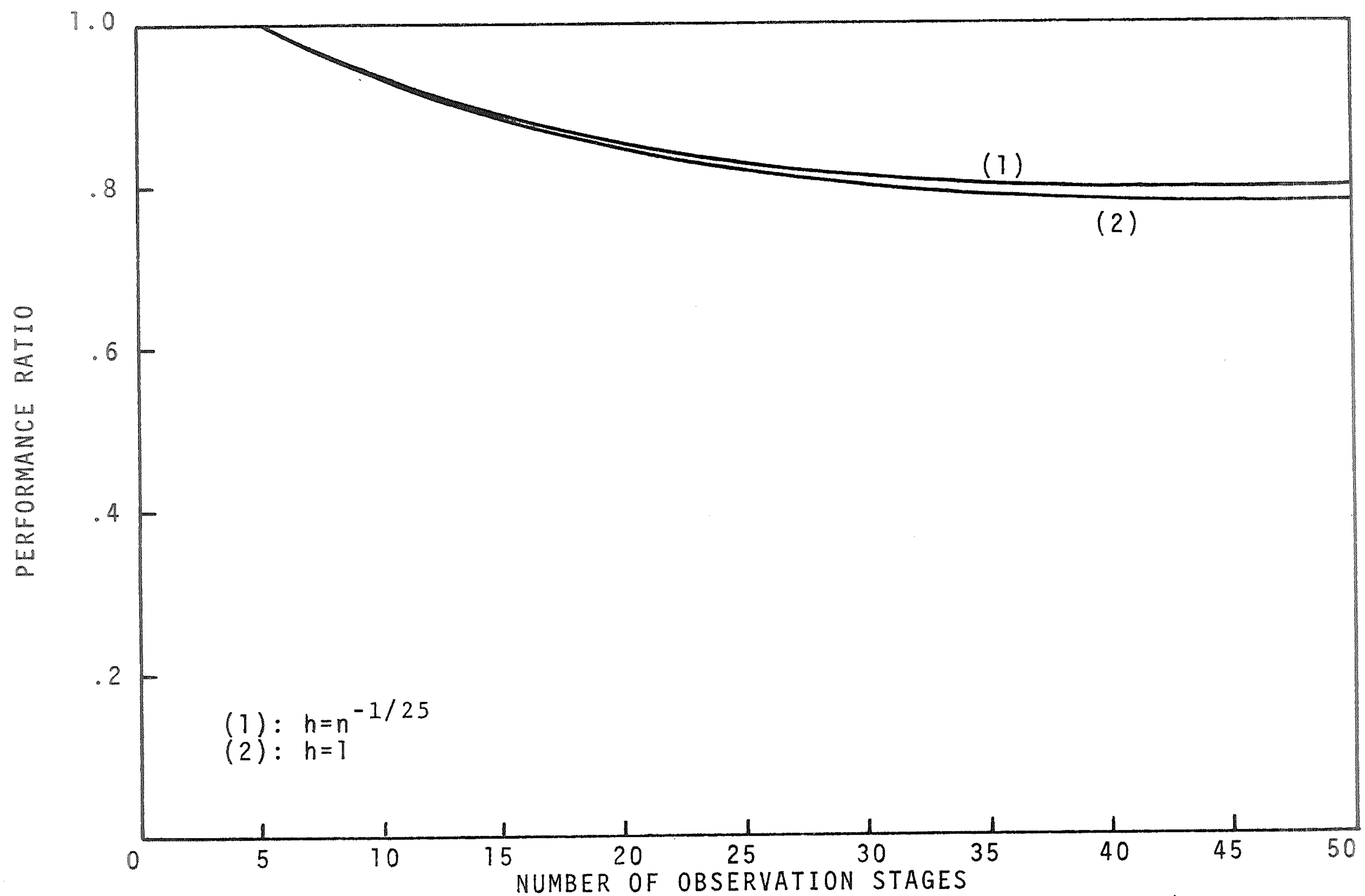


Figure 6. Performance Ratios for First Iteration for $h=1$ and $h=n^{-1/25}$ with $z=1.167$.

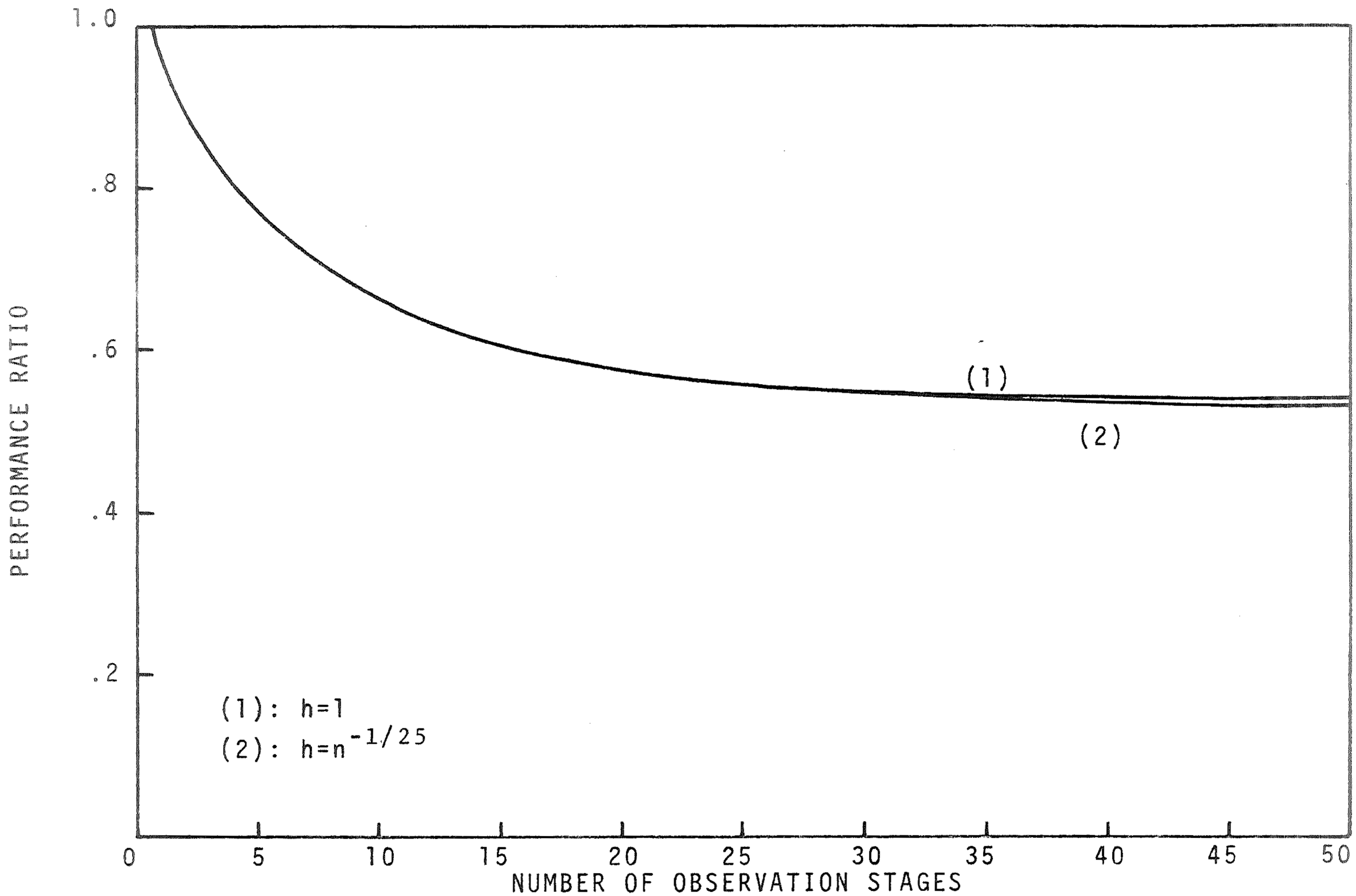


Figure 7. Performance Ratios for First Iteration with Covariance Correction for $h=1$ and $h=n^{-1/25}$ with $z=1.167$.

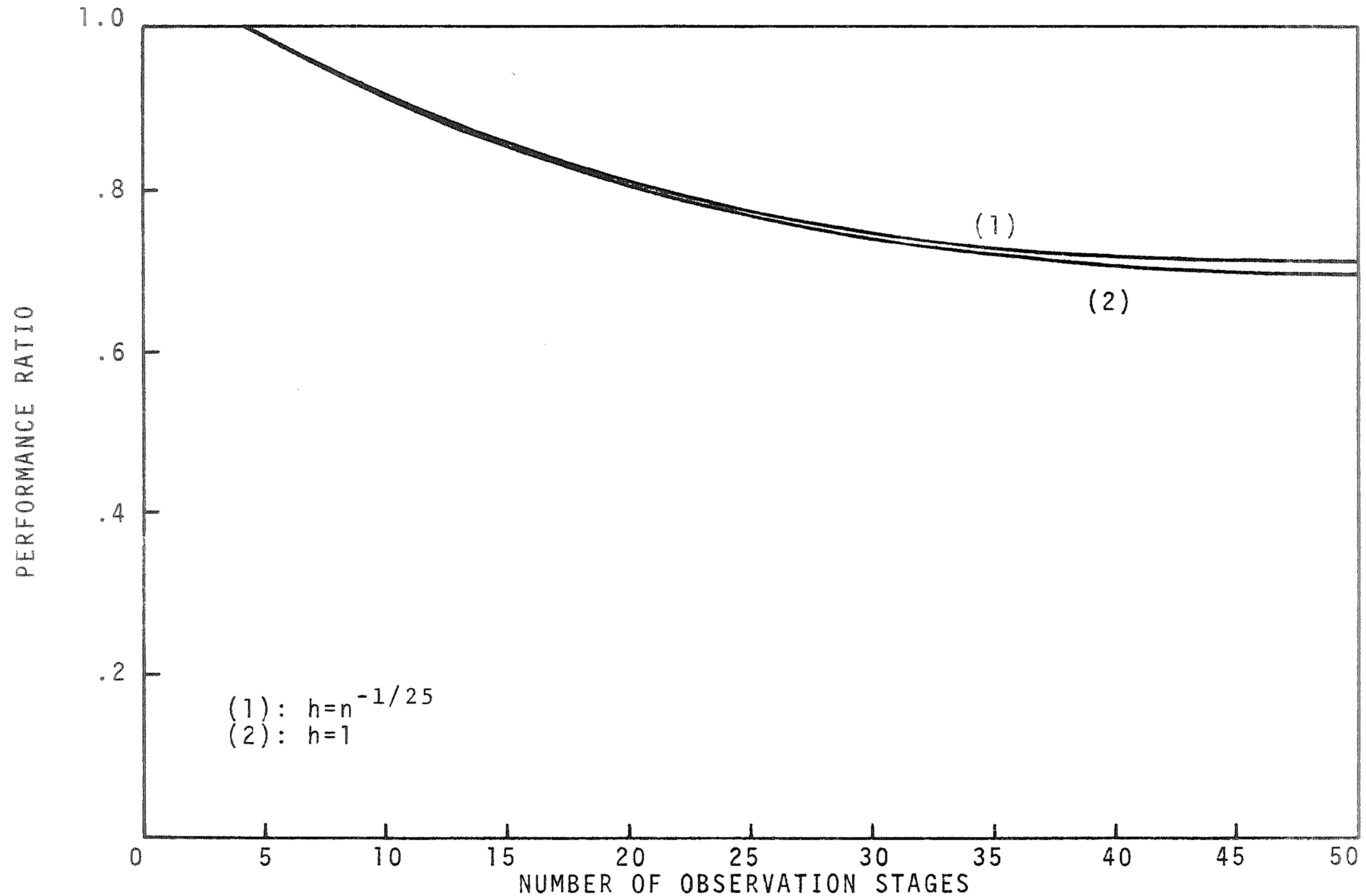


Figure 8. Performance Ratios for Second Iteration for $h=1$ and $h=n^{-1/25}$ with $z=1.167$.

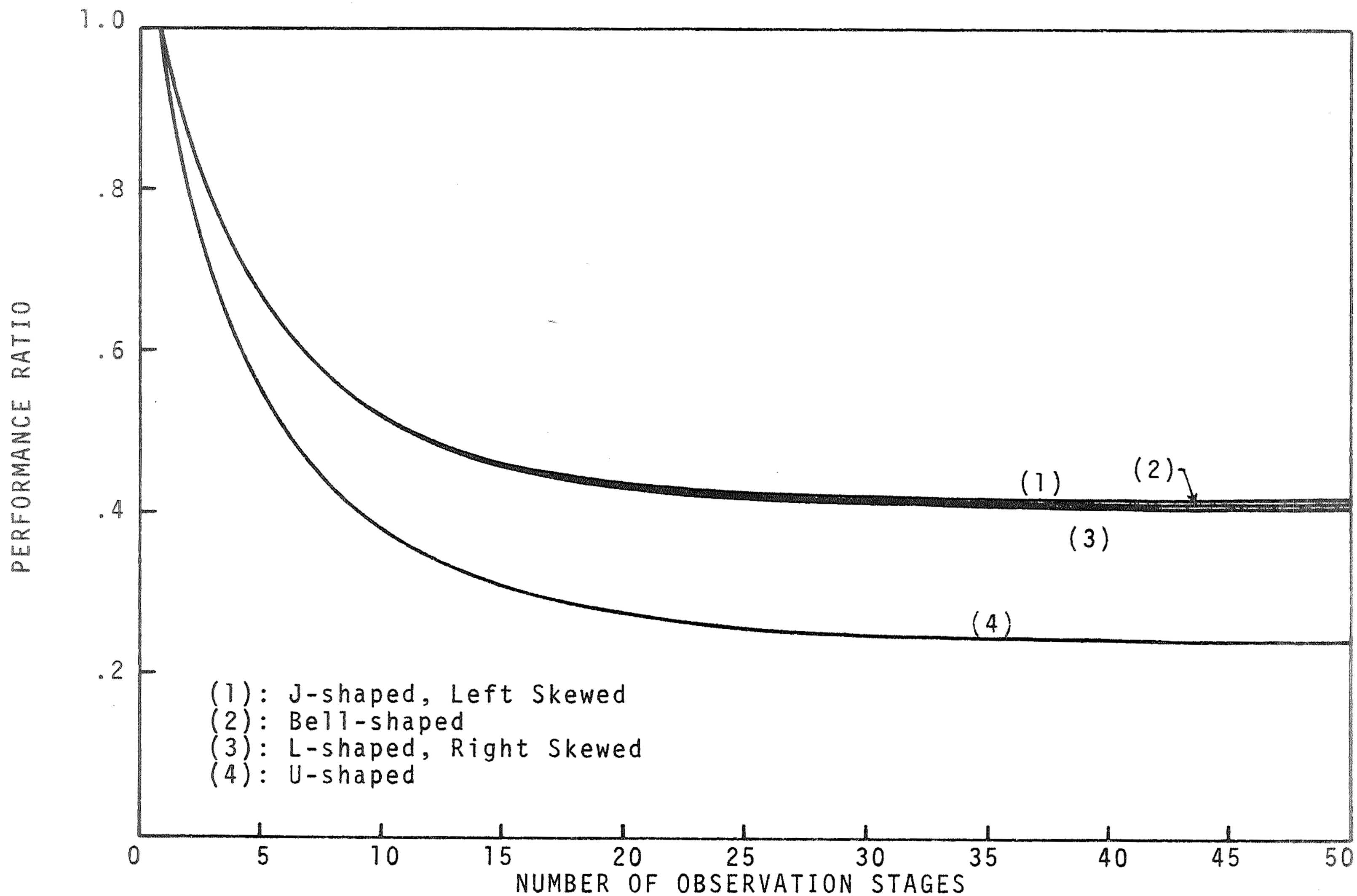


Figure 9. Performance Ratios for First Iteration with Covariance Correction for Six Different Shaped Distributions.

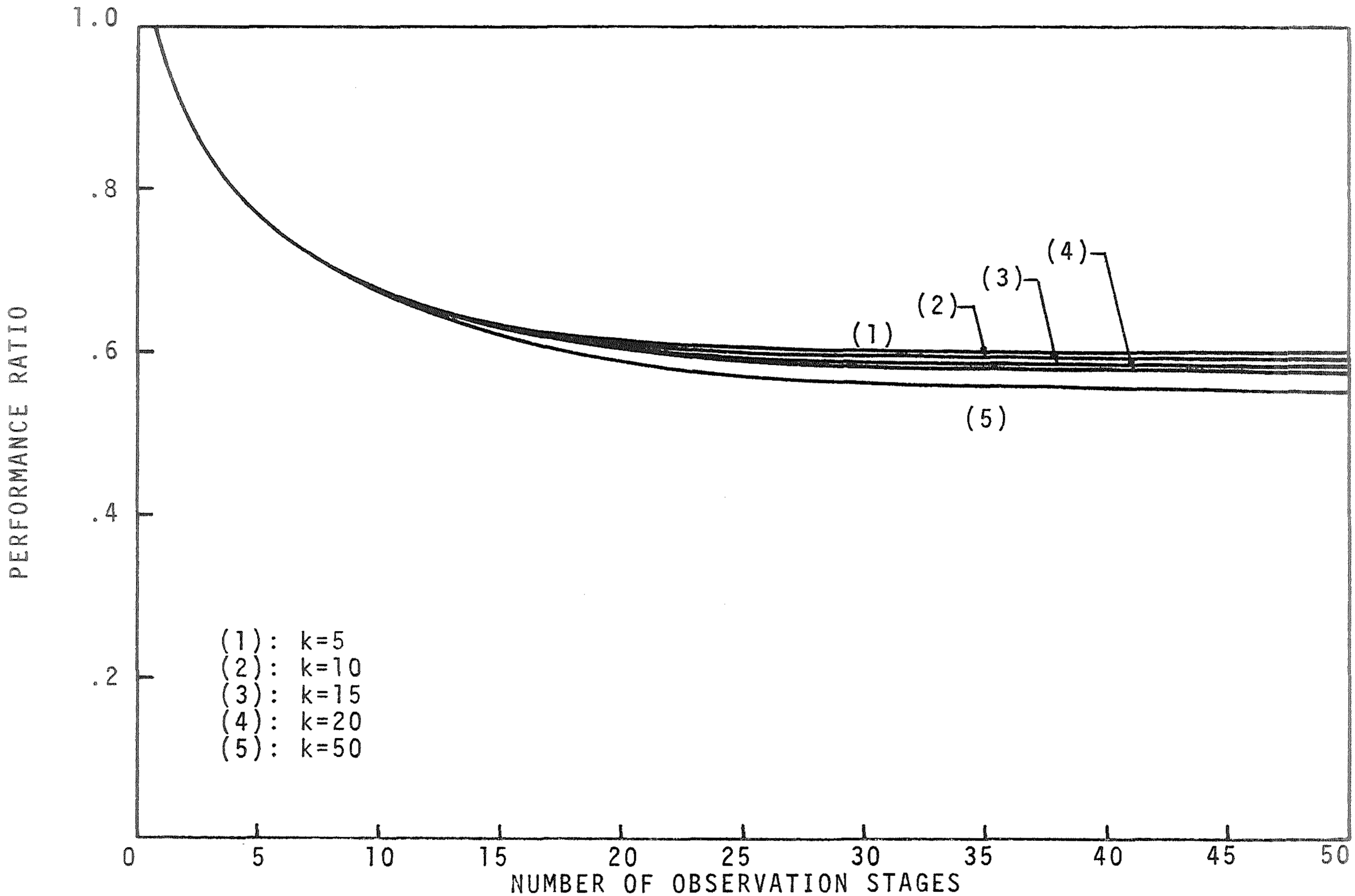


Figure 10. Performance Ratios for First Iteration with Covariance Correction with Finite Memory for z=1.167.

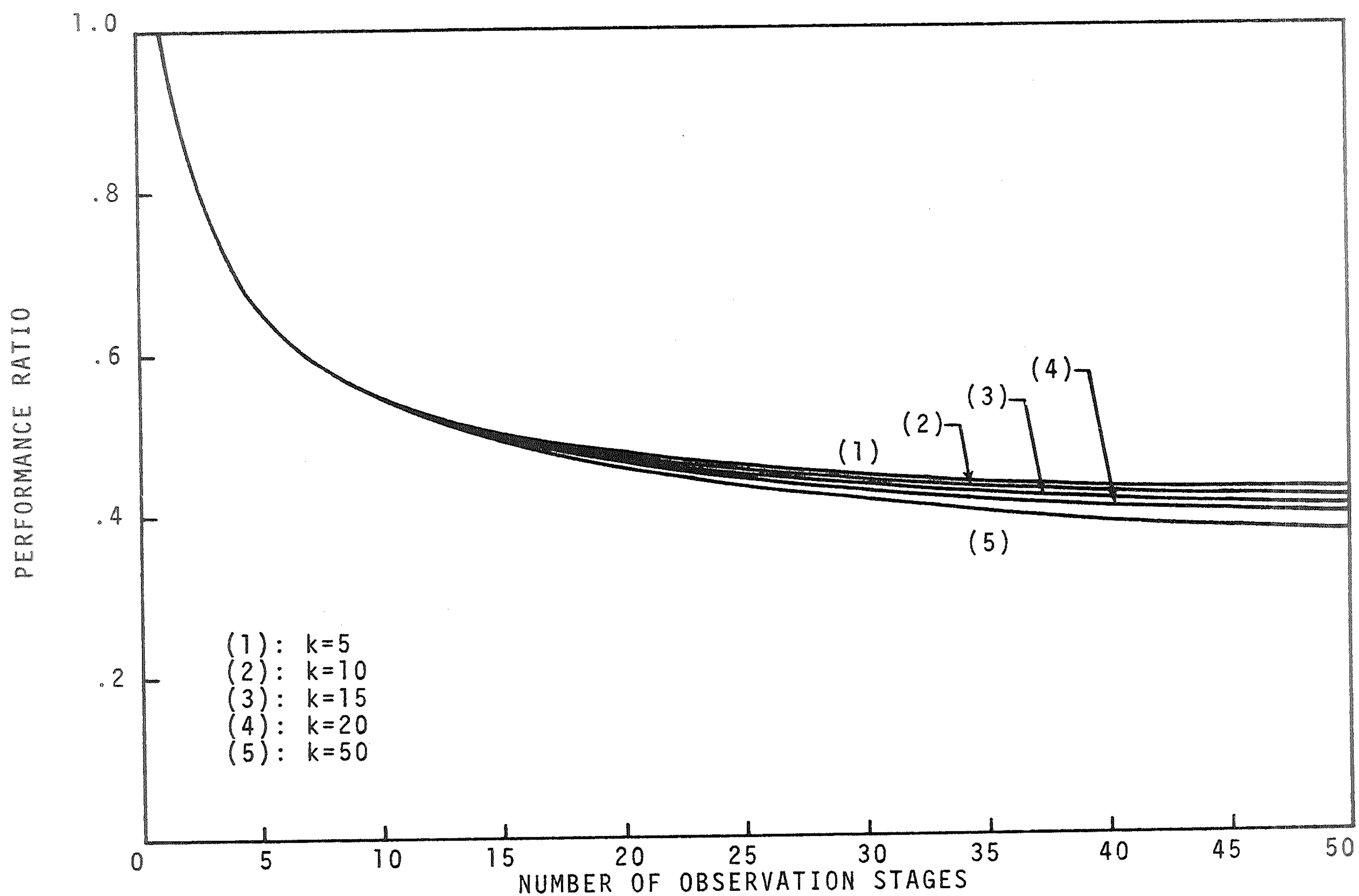


Figure 11. Performance Ratios for First Iteration with Covariance Correction with Finite Memory for z=2.217.

CHAPTER III

SMOOTH EMPIRICAL BAYES ESTIMATION IN DISCRETE TIME LINEAR SYSTEMS

Recall that a discrete time linear system is defined by

$$x_n = \Phi_{n,n-1} x_{n-1} + u_{n-1} \tag{3.1}$$

and a linear set of observations on this system is defined by

$$y_n = H_n x_n + v_n, \tag{3.2}$$

where x and u are rx1 vectors, y and v are px1 vectors, Φ is an rxr matrix and H is pxr matrix. The vector x is the state of the system, and the vector u is called the state disturbance error. The matrix Φ is called the state transition matrix and the matrix H is the matrix relating the observations to the state vector. The vector y is the observation vector and the vector v is called the observation error.

Let

$$E[x_0] = c \tag{3.3}$$

$$E[v_i] = 0 \tag{3.4}$$

$$E[v_i v_j^T] = \delta_{ij} R_i. \tag{3.5}$$

Assume that u_i is independent of v_j, for all i and j, and that u_i is independent of u_j, for all $i \neq j$. Also assume that v_i is normally distributed with a mean vector and covariance matrix as stated in equations (3.4) and (3.5) respectively. Further assume that u_{n-1} has an unknown and unspecified distribution which remains stationary over time. Assume that $c, \Phi_{i,i-1}$, H_i, and R_i are known.

Consider equation (3.1) in the form

$$u_{n-1} = x_n - \Phi_{n,n-1} x_{n-1}. \qquad (3.6)$$

Let

$$q_n = y_n - H_n \Phi_{n,n-1} x_{n-1}. \qquad (3.7)$$

Using equation (3.2), q_n can be restated as

$$q_n = H_n x_n - H_n \Phi_{n,n-1} x_{n-1} + v_n$$

$$= H_n u_{n-1} + v_n. \qquad (3.8)$$

Define

$$r_n = (H_n^T H_n)^{-1} H_n^T q_n \qquad (3.9)$$

or

$$r_n = u_{n-1} + (H_n^T H_n)^{-1} H_n^T v_n \qquad (3.10)$$

where the transformation in equation (3.9) requires the matrix $(H_n^T H_n)$ to be a full rank matrix. If, however, $(H_n^T H_n)$ is not of full rank then the use of a suitable generalized inverse should be considered. (See [15]). Recall that v_n is assumed to be normally distributed with a mean vector zero and covariance matrix R_n. Thus, given u_{n-1}, r_n will be conditionally normally distributed with mean vector u_{n-1} and covariance matrix S_n, where

$$S_n = (H_n^T H_n)^{-1} H_n^T R_n H_n (H_n^T H_n)^{-1}. \tag{3.11}$$

Assume temporarily at time t_n, that $x_0, x_1, \ldots, x_{n-1}$ are known. In this case, $r_1, \ldots, r_n$ will be known exactly by virtue of equations (3.7) and (3.9).

Recall that in Chapter II a random p-vector x_n was assumed to be conditionally normally distributed with mean vector θ_n and covariance matrix R_n, where θ_n was the realization of a random vector θ at time t_n according to an unknown and unspecified time-invariant density function $g(\theta)$. Here the random vector r_n has the same characteristics as the vector x_n in Chapter II. Hence, the smooth empirical Bayes estimator of Chapter II can be used here to estimate the state disturbance error u_{n-1}, which is the conditional mean of r_n. Recall that the "best" estimator in Chapter II was found to be that estimator using the observations transformed by a linear transformation to obtain a second order match between the prior distribution $g(\theta)$ and the estimated prior $g_n(\theta)$.

A similar estimator will be used here and its application is described below.

Define the estimates

$$\hat{E}[u_{n-1}] = \frac{\sum_{i=1}^{n} r_i}{n} = \bar{r}_n \qquad (3.12)$$

and

$$\hat{cov}[u_{n-1}] = \frac{\sum_{i=1}^{n} (r_i-\bar{r}_n)(r_i-\bar{r}_n)^T}{n-1} - S_n = C_n. \qquad (3.13)$$

Let A be a diagonal matrix with

$$A_{jj} = \{[C_n]_{jj}/[C_n + S_n]_{jj}\}^{1/2}$$

and (3.14)

$$b = (I - A)\bar{r}_n.$$

If a diagonal element of C_n is negative due to sampling error, which can happen only for small values of n, then the corresponding element of A is taken to be equal to one.

Using equations (3.12) to (3.14),

$$r_i^* = Ar_i + b \qquad (3.15)$$

is the transformed observation required for the application of the smooth empirical Bayes estimator with covariance correction. The estimate of u_{n-1} is then given by

$$\hat{u}_{n-1} = B_n^{-1} \sum_{i=1}^{n} p_{in} \exp\left[\frac{1}{2} p_{in}^T B_n^{-1} p_{in} - \frac{1}{2h^2} r_i^{*T} r_i^*\right] / \sum_{i=1}^{n} \exp\left[\frac{1}{2} p_{in}^T B_n^{-1} p_{in} - \frac{1}{2h^2} r_i^{*T} r_i^*\right] \quad (3.16)$$

where

$$B_n = S_n^{-1} + \frac{1}{h^2} I \quad (3.17)$$

$$p_{in} = S_n^{-1} r_n + \frac{1}{h^2} r_i^* \quad (3.18)$$

$$h = n^{-1/25} \quad (3.19)$$

and S_n is given by equation (3.11). Using equation (3.16) the state vector can be estimated as

$$\hat{x}_n = \Phi_{n,n-1} x_{n-1} + \hat{u}_{n-1}. \quad (3.20)$$

However, since x_i's usually remain unknown, suitable estimates are required for these unknown quantities. To use the smooth empirical Bayes filter equations at time t_n, suitable estimates of $x_0, \ldots, x_{n-1}$ are required. Since x_0 has a mean vector equal to c, $\hat{x}_0 = c$ is a suitable extimate to start the smooth empirical Bayes filter, as no observations are available at time t_0. The (n-1) estimates of x_i obtained at times $t_1, t_2, \ldots, t_{n-1}$ by means of the smooth empirical Bayes filter can then be used in forming the smooth empirical Bayes estimate of x_n. Thus the estimate of x_n becomes

$$\hat{x}_n = \Phi_{n,n-1}\hat{x}_{n-1} + u_{n-1}. \tag{3.21}$$

Let

$$\bar{x}_n = \Phi_{n,n-1}\hat{x}_{n-1}. \tag{3.22}$$

The equations for estimating x_n can then be rewritten as

$$r_{n1} = (H_n^T H_n)^{-1} H_n^T (y_n - H_n \bar{x}_n) \tag{3.23}$$

$$r_{n1}^* = A r_{n1} + b \tag{3.24}$$

$$z_{in} = S_n^{-1} r_{n1} + \frac{1}{h^2} r_{i1}^* \tag{3.25}$$

$$\hat{u}_{n-1} = B_n^{-1} \sum_{i=1}^{n} z_{in} \exp\left[\frac{1}{2} z_{in}^T B_n^{-1} z_{in} - \frac{1}{2h^2} r_{i1}^{*T} r_{i1}^*\right] \Big/ \sum_{i=1}^{n} \exp\left[\frac{1}{2} z_{in}^T B_n^{-1} z_{in} - \frac{1}{2h^2} r_{i1}^{*T} r_{i1}^*\right] \tag{3.26}$$

where S_n, B_n, and h are given by equations (3.11), (3.17), and (3.19) respectively, and the estimate of x_n is given by

$$\hat{x}_n = \bar{x}_n + \hat{u}_{n-1}. \tag{3.27}$$

The diagonal matrix A and the vector b used for the linear transformation for the covariance correction as given in equation (3.24) are given by

$$\bar{r}_n = \hat{E}[u_{n-1}] = \frac{\sum_{i=1}^{n} r_{i1}}{n}$$

$$C_{n1} = \hat{\text{cov}}[u_{n-1}]$$

$$= \frac{\sum_{i=1}^{n} (r_{i1}-\bar{r}_{n1})(r_{i1}-\bar{r}_{n1})^T}{n-1} - S_n \tag{3.29}$$

$$A_{jj} = \{[C_{n1}]_{jj}/[C_{n1} + S_n]_{jj}\}^{1/2} \tag{3.30}$$

and

$$b = (I - A)\bar{r}_{n1}. \tag{3.31}$$

Method of Analysis

Since no assumption is made about the distribution of the state disturbance error except that it be uncorrelated and stationary over time, a direct method of analysis as described in Chapter II cannot be used to obtain the statistical properties of the smooth empirical Bayes filter. The statistical properties to be studied are the expected value and covariance matrix of the estimation error, where the estimation error is the vector difference of the true state and its estimate. Thus, the following two methods have to be excluded from consideration. The first one uses analytical methods to compute the required statistical properties, which depend on the parameters of the problem. As a second analytical method, for a given state disturbance error distribution it may be possible to obtain the statistical properties desired; however, a different analysis would have to be performed for each distribution.

The third method of analysis is Monte Carlo simulation. This is basically an alternative to the second method mentioned above. The second method can be very time consuming due to the presence of exponential terms in the summations involved in the filter, with exponents being differences of quadratic forms. This method of analysis will be used in comparing the filter performance with some standard methods.

A Sample Problem

Consider a space vehicle which is assumed to be in a circular orbit around the earth at a distance of 4500 miles from the earth's center. Consider only the effect of the gravitation of the earth on the space vehicle. Define a spherical coordinate system with the center of the earth as the origin as shown in Figure 12. The effect of the earth's gravity will be assumed to be constant at points equidistant from the earth's center. Thus, the orientation of the coordinate system is insignificant. Also define a superimposed rectangular coordinate system with the same origin as the spherical coordinate system, as shown in Figure 12. The coordinates of the space vehicle in the two coordinate systems are related by the following equations.

$$z_1 = r \cos\theta \sin\phi \tag{3.32}$$

$$z_2 = r \sin\theta \sin\phi \tag{3.33}$$

and

$$z_3 = r \cos\phi . \tag{3.34}$$

Since it is assumed that the space vehicle is in a circular orbit, r remains constant. Assume that this orbit is in a plane containing the z_3-axis. Thus, θ is a constant. Assume also that the space vehicle is moving at a constant speed. Thus $d\phi/dt$ remains constant. The acceleration components along the three rectangular axes can be obtained by differentiating equations (3.32), (3.33), and (3.34) twice with respect to the time t and by keeping r, θ, $d\phi/dt$ constant and hence

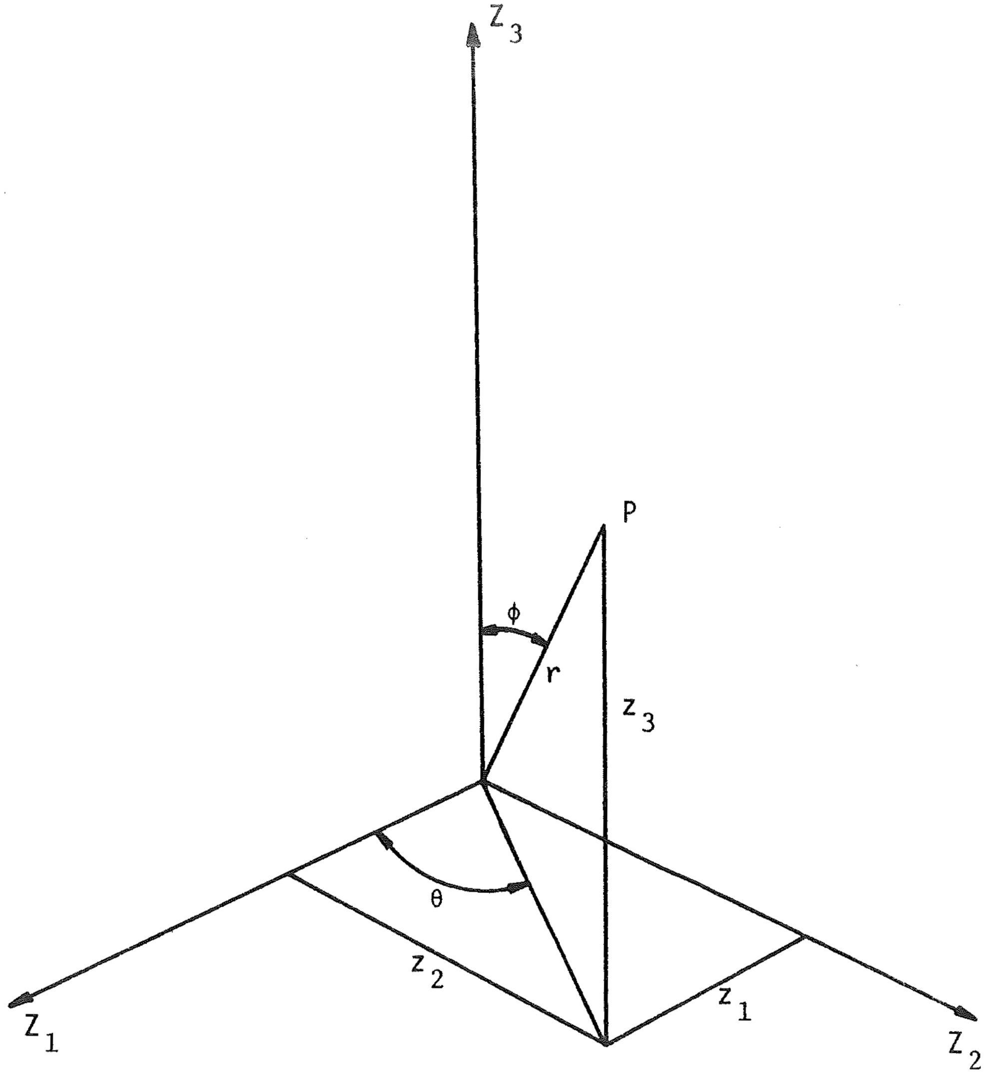


Figure 12. Superimposed Spherical and Rectangular Coordinate Systems.

$$a_1 = \frac{d^2 z_1}{dt} = -r\left[\frac{d\phi}{dt}\right]^2 \cos\theta \ \sin\left[\frac{d\phi}{dt}\ t\right] \tag{3.35}$$

$$a_2 = \frac{d^2 z_2}{dt} = -r\left[\frac{d\phi}{dt}\right]^2 \sin\theta \ \sin\left[\frac{d\phi}{dt}\ t\right] \tag{3.36}$$

$$a_3 = \frac{d^2 z_3}{dt} = -r\left[\frac{d\phi}{dt}\right]^2 \cos\left[\frac{d\phi}{dt}\ t\right]. \tag{3.37}$$

The magnitude of the resultant acceleration vector is then given by

$$a = a_1^{\ 2} + a_2^{\ 2} + a_3^{\ 2}$$

$$= r\left[\frac{d\phi}{dt}\right]^2 \tag{3.38}$$

By the laws of dynamics a must be the same as the acceleration caused by earth's gravity. Assuming an approximate value of 32.2 ft./sec.^{2} for this acceleration, the unknown constant $d\phi/dt$ becomes

$$\frac{d\phi}{dt} = \left[\frac{32.2}{r}\right]^{1/2}$$

$$= \left[\frac{32.2}{4500 \times 5280}\right]^{1/2}$$

$$= .001163. \tag{3.39}$$

Consider a six-dimensional state vector whose first three components are the coordinates z_1, z_2, z_3 and whose last three components are the velocity components along the corresponding three axes. Let the constant θ be $\pi/4$. At time t_0, let $\phi_0=0$. Then the mean c of x_0 is given by

$$c = \begin{bmatrix} 0 \\ 0 \\ 23{,}760{,}000 \\ 19{,}573.086 \\ 19{,}573.086 \\ 0 \end{bmatrix}. \tag{3.40}$$

Let Δt be such that

$$\Delta\phi = \left[\frac{d\phi}{dt}\right]\Delta t = .1 \text{ radian.}$$

Then

$$\Delta t = \frac{.1}{d\phi/dt} = \frac{.1}{.001163} = 859.8 \text{ sec.}$$

Now, at time t_n,

$$x_n = \begin{bmatrix} r\cos\theta\,\sin\phi \\ r\sin\theta\,\sin\phi \\ r\cos\phi \\ r\frac{d\phi}{dt}\cos\theta\,\cos\phi \\ r\frac{d\phi}{dt}\sin\theta\,\cos\phi \\ -r\frac{d\phi}{dt}\sin\phi \end{bmatrix} \tag{3.41}$$

and at time t_{n+1},

$$x_{n+1} = \begin{bmatrix} r\cos\theta\,(\sin\phi\cos\Delta\phi + \cos\phi\sin\Delta\phi) \\ r\sin\theta\,(\sin\phi\cos\Delta\phi + \cos\phi\sin\Delta\phi) \\ r(\cos\phi\cos\Delta\phi - \sin\phi\sin\Delta\phi) \\ r\frac{d\phi}{dt}\cos\theta\,(\cos\phi\cos\Delta\phi - \sin\phi\sin\Delta\phi) \\ r\frac{d\phi}{dt}\sin\theta\,(\cos\phi\cos\Delta\phi - \sin\phi\sin\Delta\phi) \\ -r\frac{d\phi}{dt}(\sin\phi\cos\Delta\phi + \cos\phi\sin\Delta\phi) \end{bmatrix}. \quad (3.42)$$

Relating x_n to x_{n+1}, and using the known values of r, θ, dϕ/dt, and Δϕ, the following state transition matrix is obtained.

$$\Phi = \begin{bmatrix} a & 0 & b & 0 & 0 & 0 \\ 0 & a & b & 0 & 0 & 0 \\ c & 0 & a & 0 & 0 & 0 \\ d & 0 & 0 & a & 0 & 0 \\ 0 & d & 0 & 0 & a & 0 \\ 0 & 0 & d & 0 & 0 & a \end{bmatrix} \quad (3.43)$$

where

$$a = .995$$
$$b = .07059073$$
$$c = -.14118153$$
$$d = -.0001163.$$

Further assume that H is a constant matrix equal to the identity matrix. This is equivalent to observing the coordinates and the velocity components of the space vehicle directly. Let P_0, and Q be the covariance matrices of the initial state vector and the state disturbance error. These matrices are required for the simulation. Assume that the following covariance matrices are known.

$$P_0 = \begin{bmatrix} 25000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 250 & 0 & 0 \\ 0 & 0 & 0 & 0 & 250 & 0 \\ 0 & 0 & 0 & 0 & 0 & 250 \end{bmatrix}$$

$$Q = \begin{bmatrix} 25000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 250 & 0 & 0 \\ 0 & 0 & 0 & 0 & 250 & 0 \\ 0 & 0 & 0 & 0 & 0 & 250 \end{bmatrix}$$

$$R = \begin{bmatrix} 20000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 200 \end{bmatrix}$$

For convenience the matrix R is assumed to be time independent. It can be seen from these matrices that the standard deviation of each of the first three components of x_0, u_i and v_i is equal to 158.11, 158.11, and 141.42 feet respectively, and the standard deviation of each of the last three components of x_0, u_i and v_i is 15.81, 15.81, and 14.14 ft./sec. respectively.

Results of Monte Carlo Simulation

The program used to simulate the above problem is described in Appendix B. One hundred replications are generated and averaged to obtain estimates of the mean squared error matrices. The initial state vector x_0 is generated from a multivariate normal distribution with mean vector c and covariance matrix P_0. The elements of the state disturbance error vector are generated from a member of the Pearson family of distributions so that the mean vector is zero and the covariance matrix is Q. The observation error vector v is generated from a multivariate normal distribution with zero mean and covariance matrix R.

For each set of parameters three different filters are used to simulate the estimation process. They are the smooth empirical Bayes filter, the Kalman filter, and the least squares filter. Recall that the Kalman filter is defined by the following equations.

$$\bar{x}_n = \Phi_{n,n-1}\hat{x}_{n-1} \qquad (3.44)$$

$$\overline{P}_n = \Phi_{n,n-1} P_{n-1} \Phi^T_{n,n-1} + Q_{n-1}$$

$$\hat{x}_n = \overline{x}_n + \overline{P}_n H_n^T (H_n \overline{P}_n H_n^T + R_n)^{-1} (y_n - H_n \overline{x}_n) \qquad (3.46)$$

$$P_n = \overline{P}_n - \overline{P}_n H_n^T (H_n \overline{P}_n H_n^T + R_n)^{-1} H_n \overline{P}_n \qquad (3.47)$$

The least squares filter can be restated as

$$\hat{x}_n = \left[\sum_{i=1}^{n} \psi^T_{n,i} H_i^T H_i \psi_{n,i} \right]^{-1} \sum_{i=1}^{n} \psi^T_{n,i} H_i^T y_i \qquad (3.48)$$

where

$$\psi_{i,i} = I \qquad (3.49)$$

$$\psi_{n,i} = \Phi_{i,n} = \prod_{j=i+1}^{n} \Phi^{-1}_{j,j-1}, \quad n>i. \qquad (3.50)$$

The ratio of the trace of the average squared error matrix at each stage of the process to the trace of the matrix S at that stage is used as a scalar index of the performance of the particular filter. Due to the particular choice of the matrix H, the matrix S is the same as the matrix R, the observation error covariance matrix. Figure 13 presents the performance ratio for the least squares filter as a function of n, the stage of the process, when an L-shaped distribution is used to generate the components of the state disturbance error vector.

Figure 14 gives the performance ratio for the same filter, but for a U-shaped distribution on the components of the state disturbance error vector.

From these two figures, the performance ratio, and hence the trace of the average squared error matrix which is the numerator of the performance ratio, can be seen to diverge rapidly with n. The divergence is even more pronounced for the U-shaped distribution. This agrees with the findings reported by Bucy and Joseph [11]. The presence of the state disturbance error makes the components of the combined error vector γ_n non-independent, where γ_n is defined as

$$\gamma_n = \begin{bmatrix} v_1 - H_1 \sum_{j=1}^{n-1} \Phi_{1,j+1} u_j \\ v_2 - H_2 \sum_{j=2}^{n-1} \Phi_{2,j+1} u_j \\ \vdots \\ v_{n-1} - H_n u_{n-1} \\ v_n \end{bmatrix} .$$

The estimates obtained by the use of the least squares filter are thus not minimum variance estimates. Since the performance of the least squares filter is generally poor in the presence of the state disturbance error, it will not be considered further.

Figure 15 presents the performance ratios for the smooth empirical Bayes filter and the Kalman filter for an L-shaped distribution for the elements of the state disturbance error vector. The performance of the Kalman filter in this case is seen to be better than the performance of the smooth empirical Bayes filter. Figure 16 gives the performance ratios for the above two filters for a U-shaped distribution on the components of the state disturbance error vector. In this case, the smooth empirical Bayes filter outperforms the Kalman filter.

Figure 17 presents the performance ratios for the smooth empirical Bayes filter when used with six different shaped distributions on the elements of the state disturbance error vector. The performance of this filter can be seen to be about the same for all the distributions, except for the U-shaped distribution for which it gives slightly better results. Figure 18 gives the performance ratios for the Kalman filter for the same six distributions on the components of the state disturbance error. The Kalman filter can be seen to perform almost the same for all the distributions, except for the U-shaped distribution for which it gives relatively poor results.

A summary quantity z is defined as in Chapter II by

$$z = \frac{1}{6} \sum_{j=1}^{6} z_j$$

where

$$z_j = \frac{[S_n - (Q_n^{-1} + S_n^{-1})^{-1}]_{jj}}{[(Q_n^{-1} + S_n^{-1})^{-1}]_{jj}} . \qquad (3.51)$$

For the parameters used in simulating the results presented in Figures 13 to 18, the average summary quantity was z=.8. To observe the effect of larger covariance matrices on the performances of the smooth empirical Bayes filter and the Kalman filter, the following matrices were used, which gave the summary quantity of z=1.2.

$$P_0 = Q = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 & 0 & b \end{bmatrix}$$

$$R = \begin{bmatrix} c & 0 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & 0 & d \end{bmatrix}$$

where

$$a = 625{,}000{,}000$$

$$b = 62{,}500$$

$$c = 750{,}000{,}000$$

$$d = 75{,}000.$$

It can be observed from these matrices that the standard deviation of each of the first three components of x_0, u_i and v_i is equal to 25,000, 25,000, and 27,386 feet respectively, and the standard deviation of each of the last three components of x_0, u_i and v_i is 250, 250, and 273.9 ft./sec. respectively.

Figure 19 presents the performance ratios for the smooth empirical Bayes filter and the Kalman filter when used with an L-shaped distribution on the components of the state disturbance error and with covariance matrices defined above for z=1.2. The Kalman filter can be observed to perform better than the smooth empirical Bayes filter. Figure 20 gives the performance ratios for the same two filters used with the same covariance matrices for z=1.2 but with a U-shaped distribution on the components of the state disturbance error vector. In this case, the smooth empirical Bayes filter outperforms the Kalman filter.

The observation error covariance matrix is then changed to the following matrix to obtain a summary quantity z=2.0.

$$R = \begin{bmatrix} e & 0 & 0 & 0 & 0 & 0 \\ 0 & e & 0 & 0 & 0 & 0 \\ 0 & 0 & e & 0 & 0 & 0 \\ 0 & 0 & 0 & f & 0 & 0 \\ 0 & 0 & 0 & 0 & f & 0 \\ 0 & 0 & 0 & 0 & 0 & f \end{bmatrix}$$

where

$$e = 1{,}250{,}000{,}000$$

$$f = 125{,}000.$$

The covariance matrices P_0 and Q are the same as for z=1.2.

Figure 21 presents the performance ratios for the smooth empirical Bayes filter and the Kalman filter used with z=2.0 and with an L-shaped distribution on the components of the state disturbance error. The Kalman filter is seen to outperform the smooth empirical Bayes filter. Figure 22 presents the performance ratios for the same two filters when used with the same covariance matrices; but, with a U-shaped distribution on the components of the state disturbance error. The smooth empirical Bayes filter outperforms the Kalman filter in this case.

Since the development of the Kalman filter assumes a multivariate normal distribution for the state disturbance error, it will be a minimum variance estimator of the state vector, provided this assumption holds. (See Kalman [18]). The smooth empirical Bayes estimator in this case cannot be as good as the Kalman filter. Figure 23 presents the performance ratios for the two filters when the state disturbance error is generated from a multivariate normal distribution for z=2.0. As stated above, the Kalman filter does outperform the smooth empirical Bayes filter.

Figure 24 presents the performance ratios for the smooth empirical Bayes filter when used with different sets of covariance matrices; that is, for z=.8, 1.2, and 2.0, and with an L-shaped distribution on the components of the state disturbance error. It can be observed that the performance of this filter improves with increase in z. Figure 25 presents the performance ratios for the same filter for the same z values but with a U-shaped distribution on the components of the state disturbance error. In this case also, the performance of this filter is observed to improve with increase in z.

The covariance matrix Q_i of the state disturbance error vector was assumed to be a constant matrix in the development of the smooth empirical Bayes filter. This would restrict the application of this filter to a class of problems where the above assumption holds true. It is thus desired to observe the sensitivity of the filter when applied to a problem where Q_i is different for each i. For this purpose the constant matrix Q is multiplied by a random number which is generated from a uniform distribution on the interval zero to one, to obtain the matrix Q_i. Since the mean of the random numbers used will be 0.5, the z value on the average will be twice the z value obtained by using the matrix Q. Figure 26 gives the performance ratios for the smooth empirical Bayes filter and the Kalman filter when used with changing Q matrices and with an L-shaped distribution on the components of the state disturbance error. The Kalman filter outperforms the smooth empirical Bayes filter; however, the performance of the smooth empirical Bayes filter is not very poor. Figure 27 presents the performance ratios for the same two filters under the same conditions as above but with a U-shaped distribution on the components of the state disturbance error. The smooth empirical Bayes filter is observed to outperform the Kalman filter in this case.

Conclusions

A smooth empirical Bayes filter has been developed for estimating the state vector of a discrete time linear system with linear sets of observations. The distributional assumptions on the state disturbance error as well as some of the distributional assumptions on the initial state vector have been relaxed. The performance of this filter is examined by means of Monte Carlo simulation of a realistic problem in trajectory estimation. It is found that the performance of this filter does not depend significantly on the form of the state disturbance error distribution used in simulation. Comparisons of this filter's performance with the performance of the Kalman filter for different sets of parameters showed that the Kalman filter outperforms the smooth empirical Bayes filter for all unimodal distributions on the components of the state disturbance error, whereas the smooth empirical Bayes filter had a better performance than the Kalman filter when used with a U-shaped distribution on the components of the state disturbance error. It has also been observed that the performance of the smooth empirical Bayes filter depends on the relative magnitudes of the covariance matrices of the state disturbance error and the observation error. The summary quantity z contains the information on these relative magnitudes, and the performance of the filter is found to improve with increase in z. Since the performance of the least squares filter is found to be very poor for the problem simulated, it is not used in any comparisons.

The Kalman filter requires approximately 0.66 seconds to process one observation. The time to process one observation using the smooth empirical Bayes filter depends on the number of past observations used. For twenty-five past observations the time required was 0.82 seconds. Hence, if a finite memory filter with a maximum of twenty-five past observations is used, the smooth empirical Bayes filter is compatible with the Kalman filter for real time estimation. The smooth empirical Bayes filter can be used as a batch processor just as the Kalman filter.

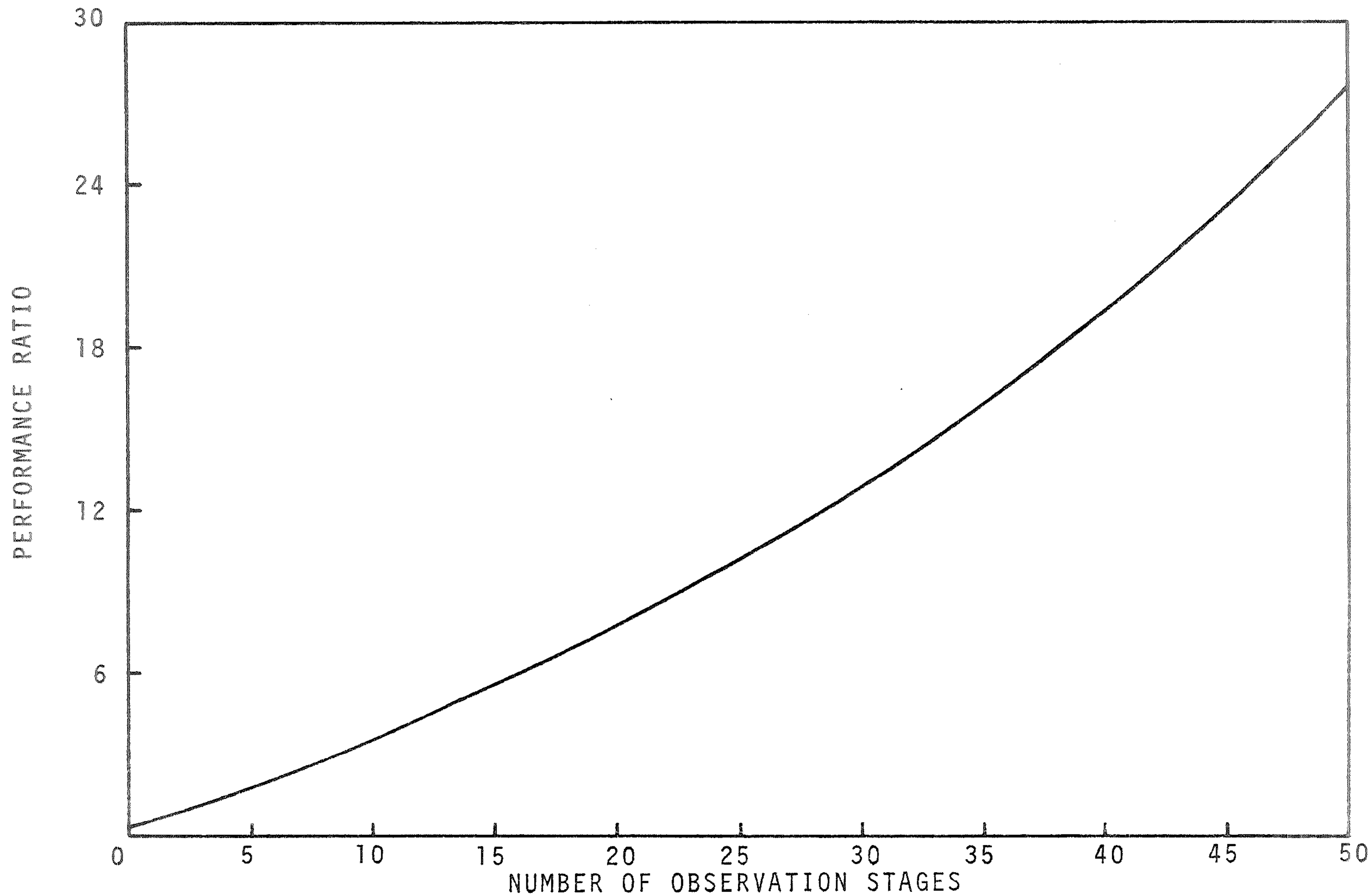


Figure 13. Performance Ratio for Least Squares Filter for z=0.8 with L-shaped Distribution on Components of u.

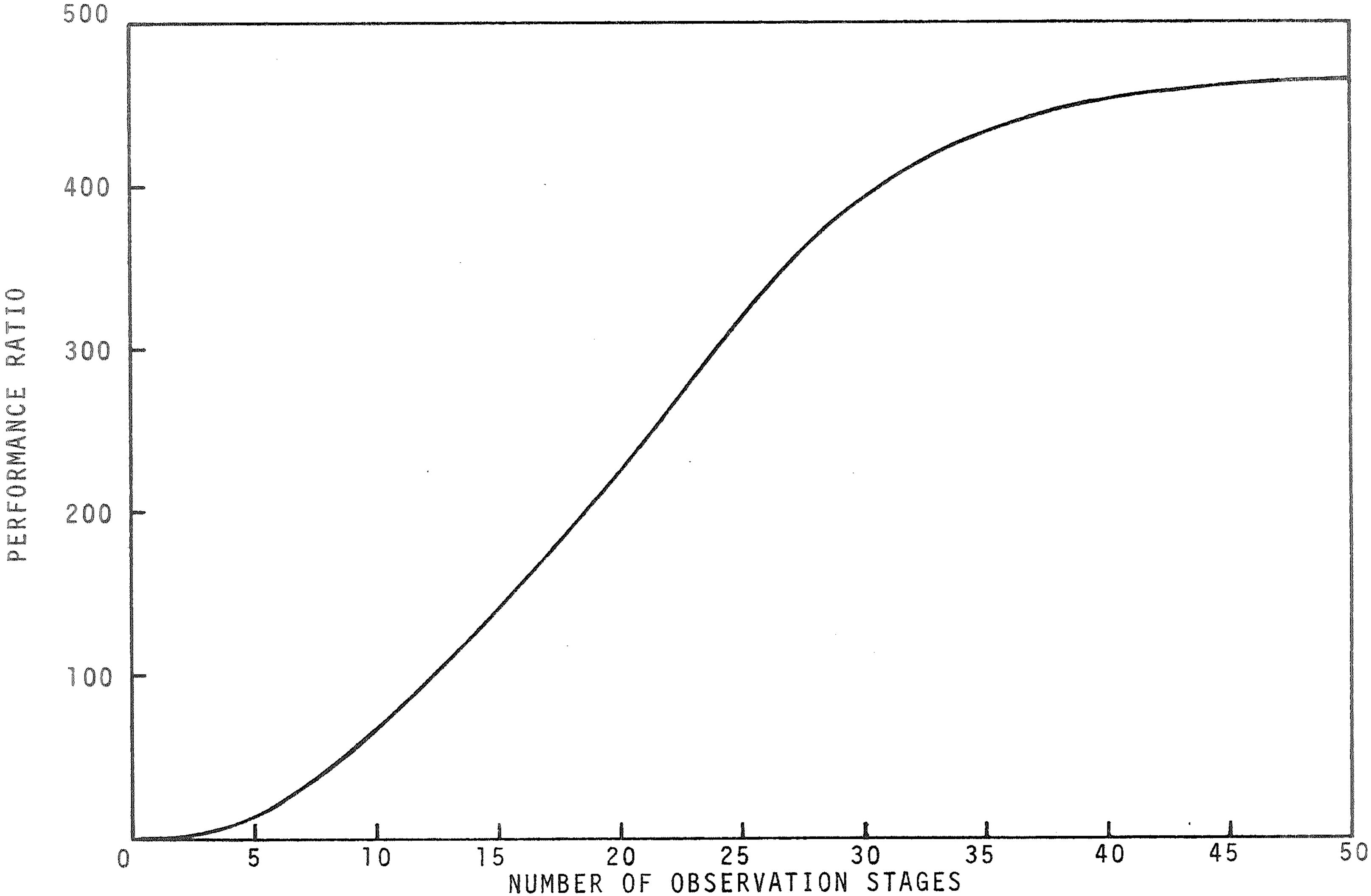


Figure 14. Performance Ratio for Least Squares Filter for z=0.8 with U-shaped Distribution on Components of u.

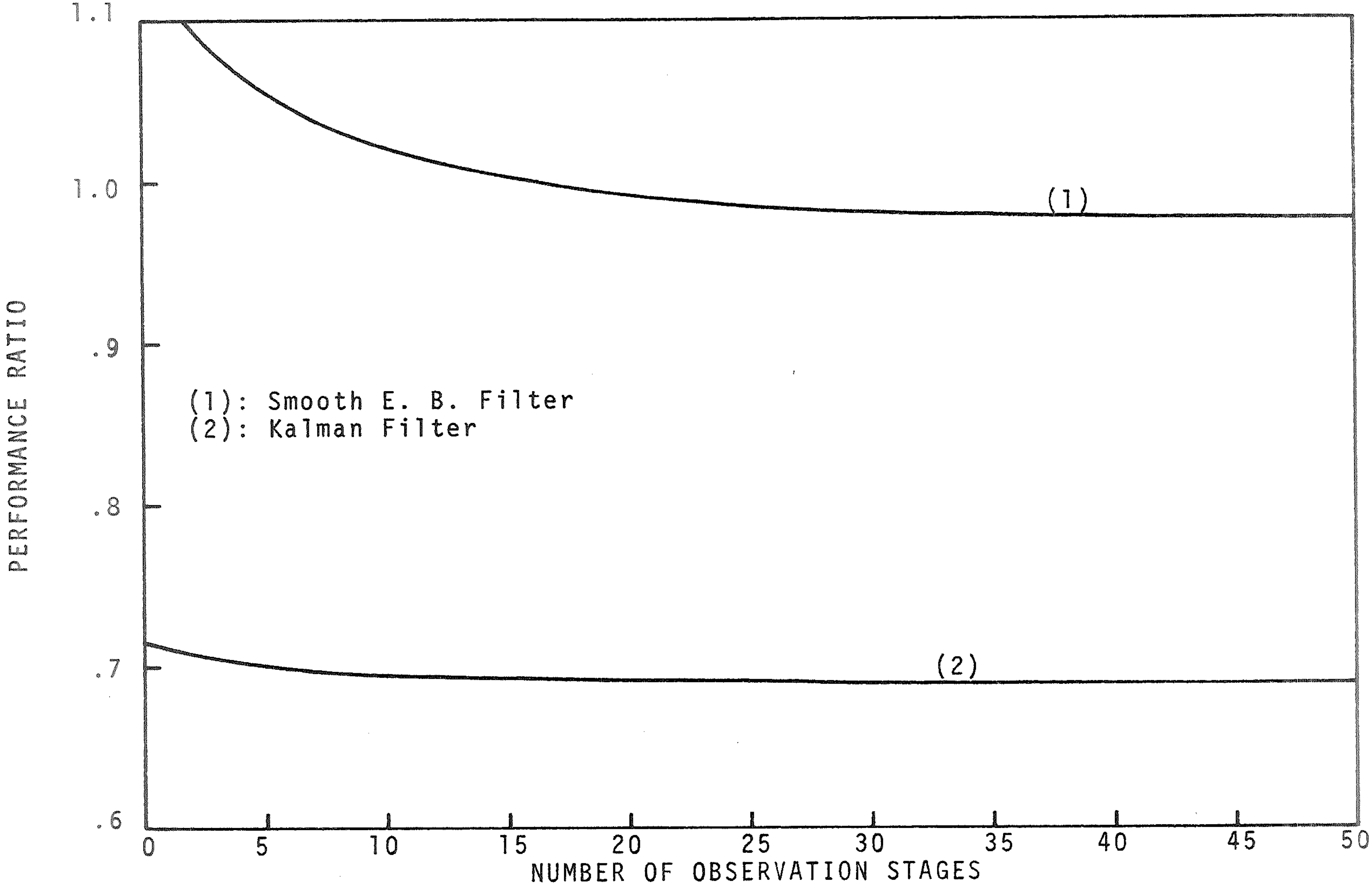


Figure 15. Performance Ratios for Smooth E. B. Filter and Kalman Filter for $z=0.8$ with L-shaped distribution on Components of u.

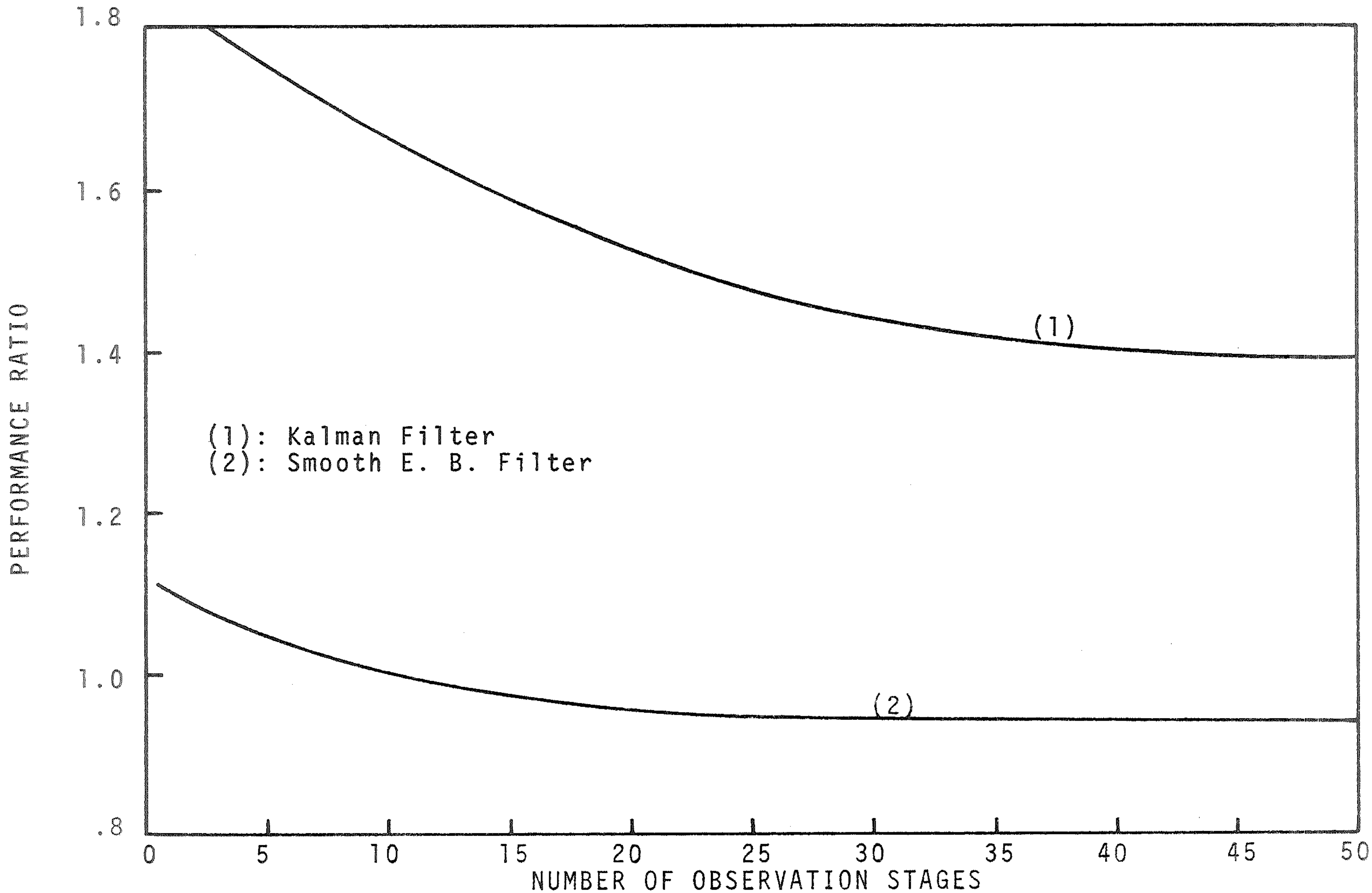


Figure 16. Performance Ratios for Smooth E. B. Filter and Kalman Filter for $z=0.8$ with U-shaped Distribution on Components of u.

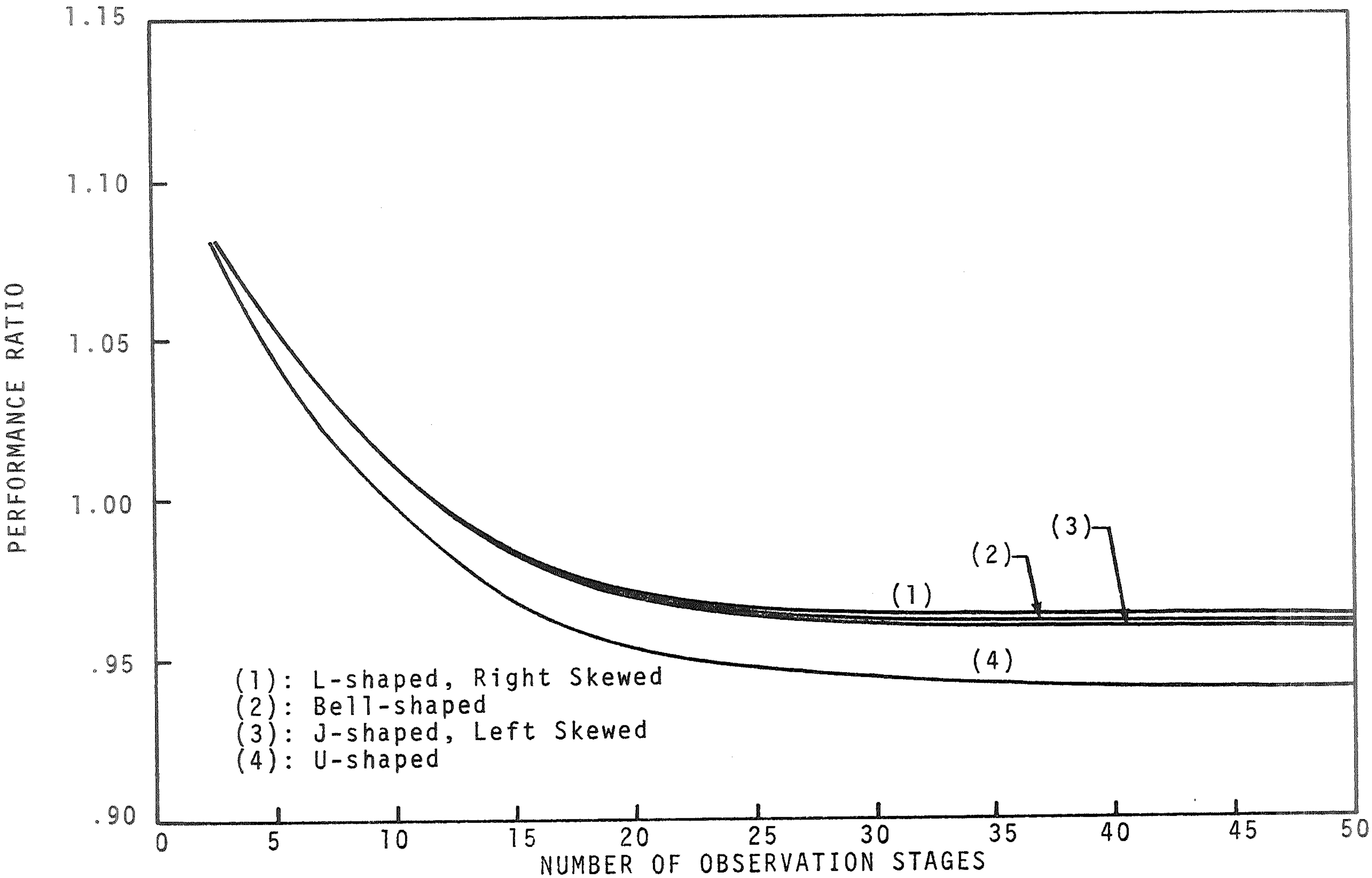


Figure 17. Performance Ratios for Smooth E. B. Filter for z=0.8 with Six Different Shaped Distributions on Components of u.

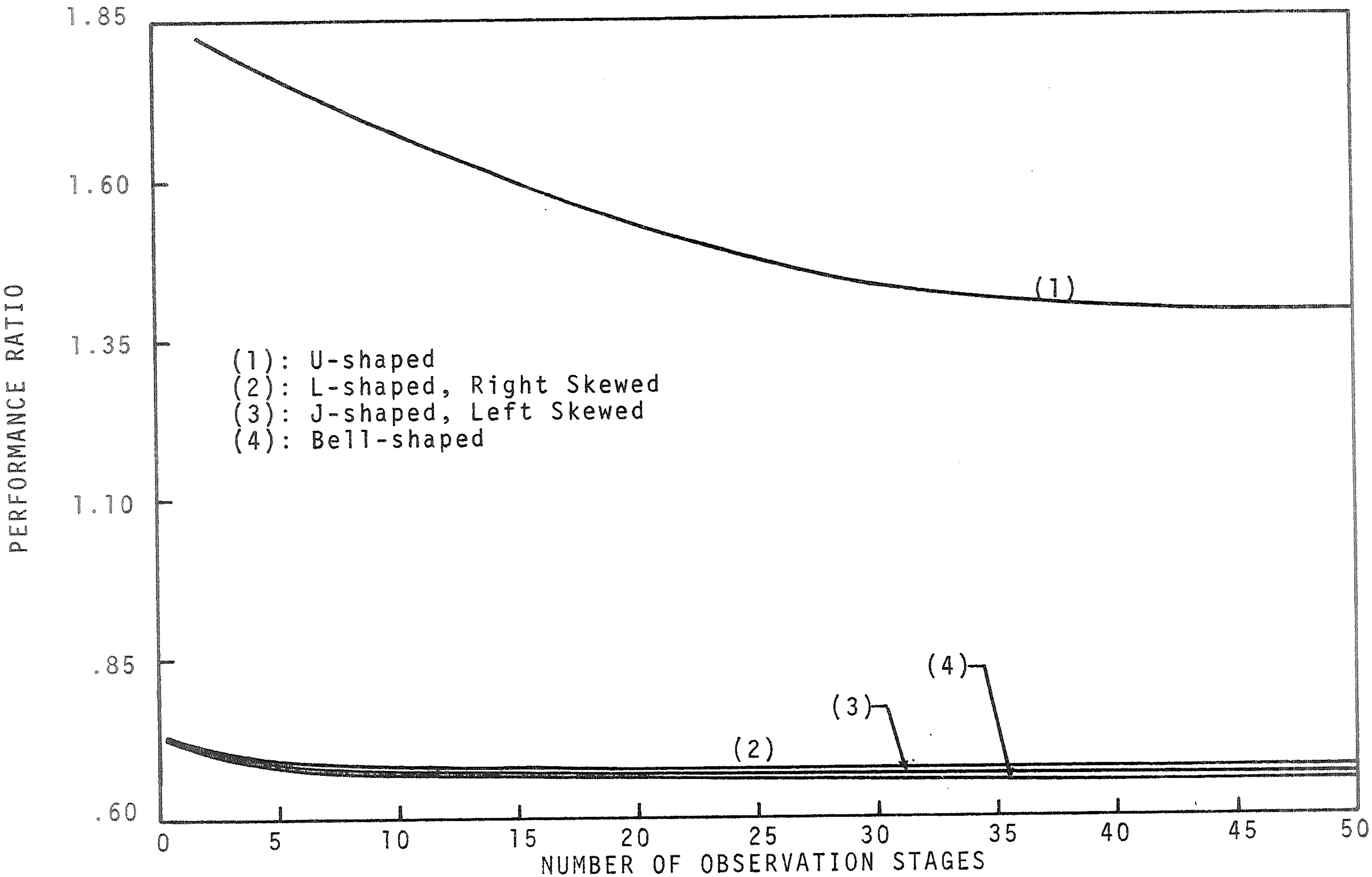


Figure 18. Performance Ratios for Kalman Filter for z=0.8 with Six Different Shaped Distribution on Components of u.

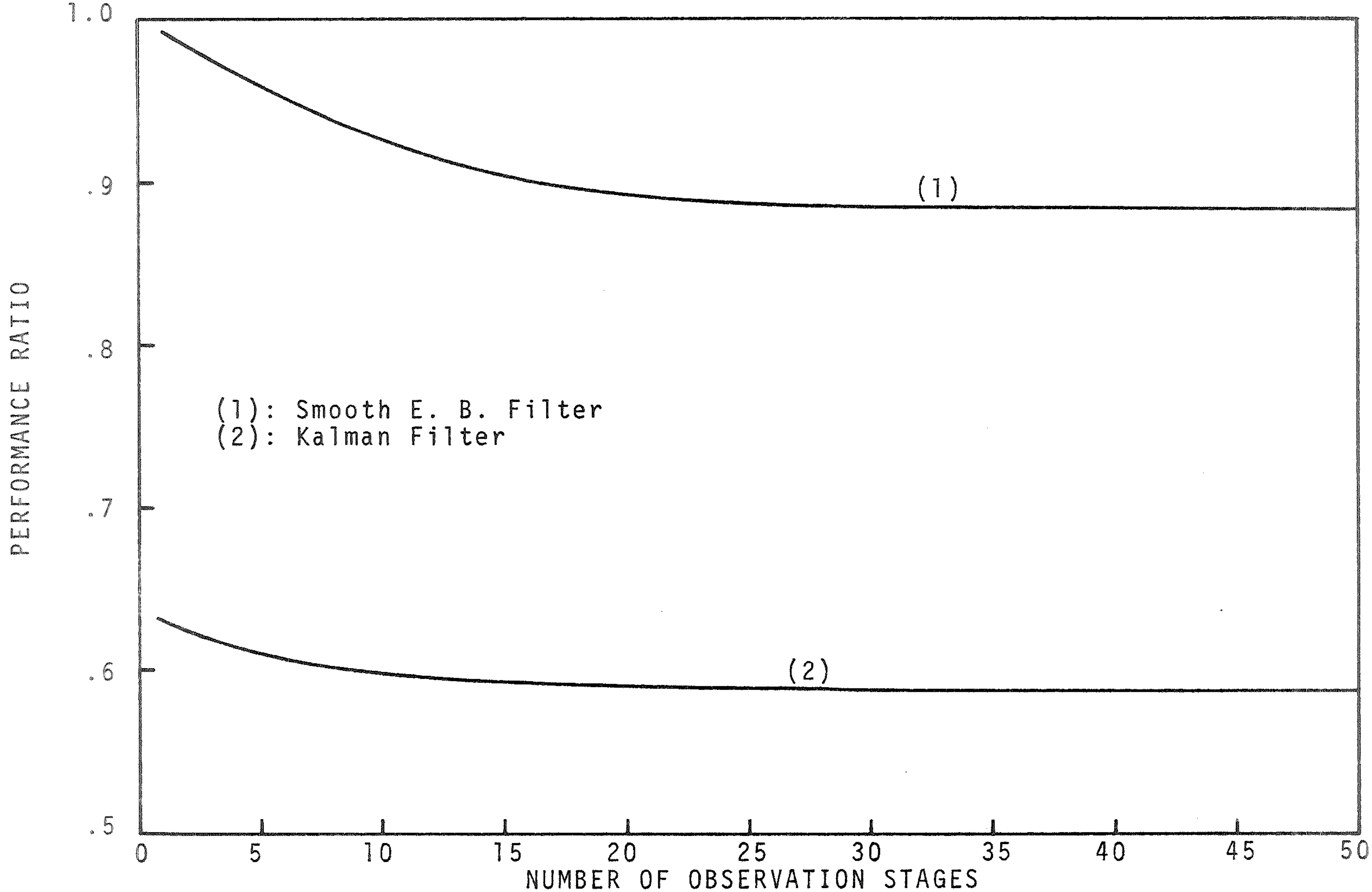


Figure 19. Performance Ratios for Smooth E. B. Filter and Kalman Filter for z=1.2 with L-shaped Distribution on Components of u.

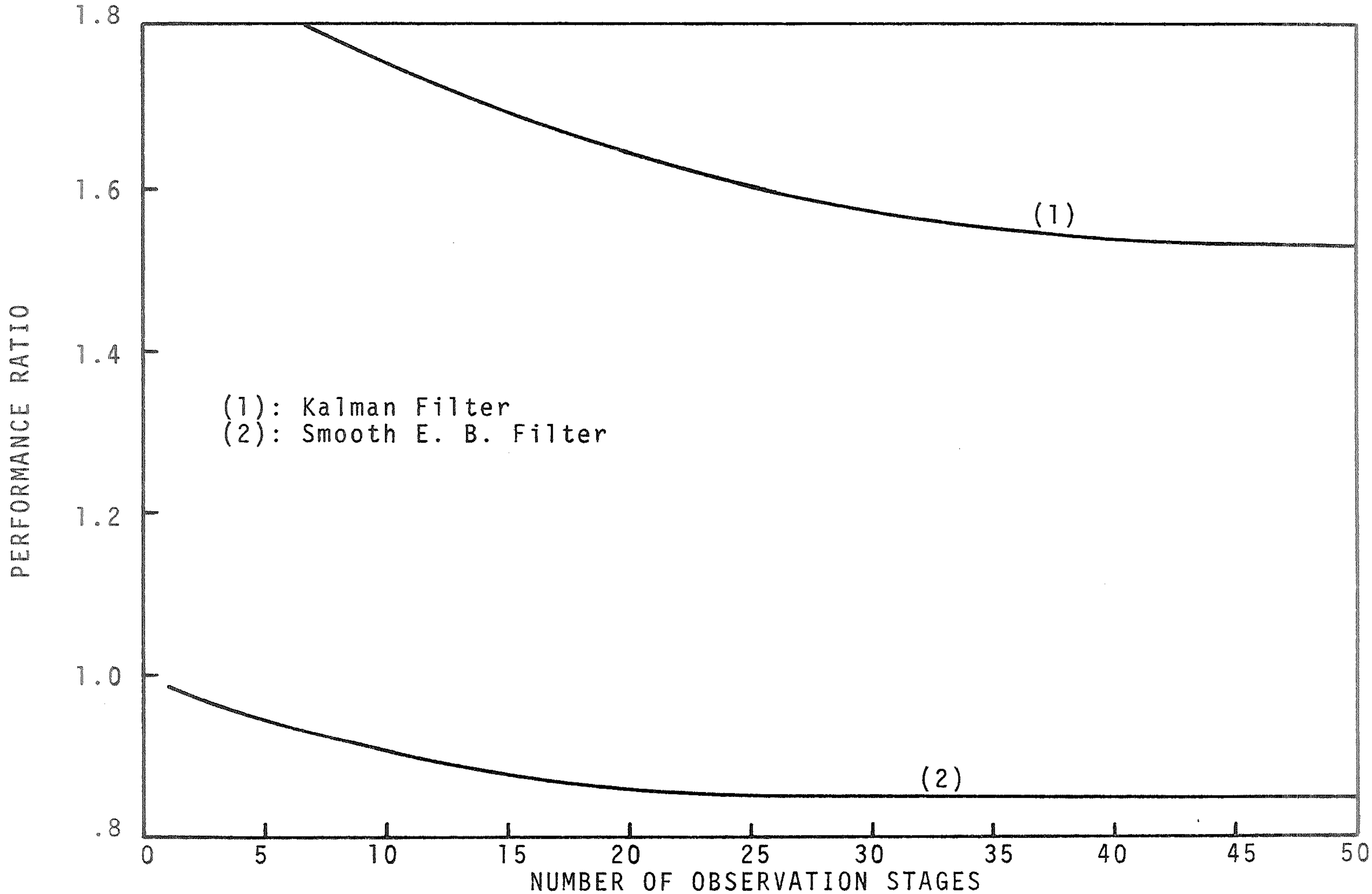


Figure 20. Performance Ratios for Smooth E. B. Filter and Kalman Filter for z=1.2 with U-shaped Distribution on Components of u.

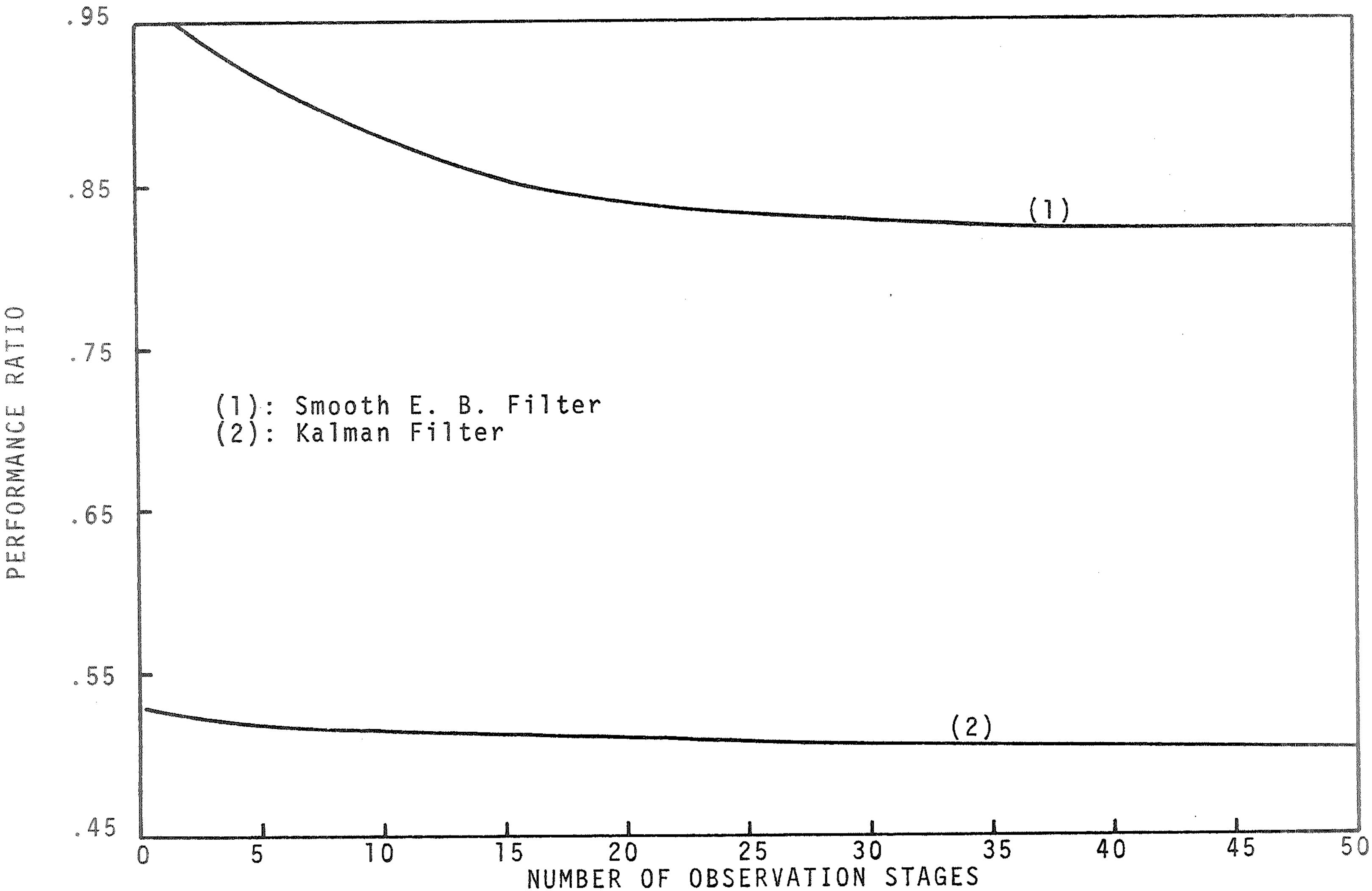


Figure 21. Performance Ratios for Smooth E. B. Filter and Kalman Filter for z=2.0 with L-shaped Distribution on Components of u.

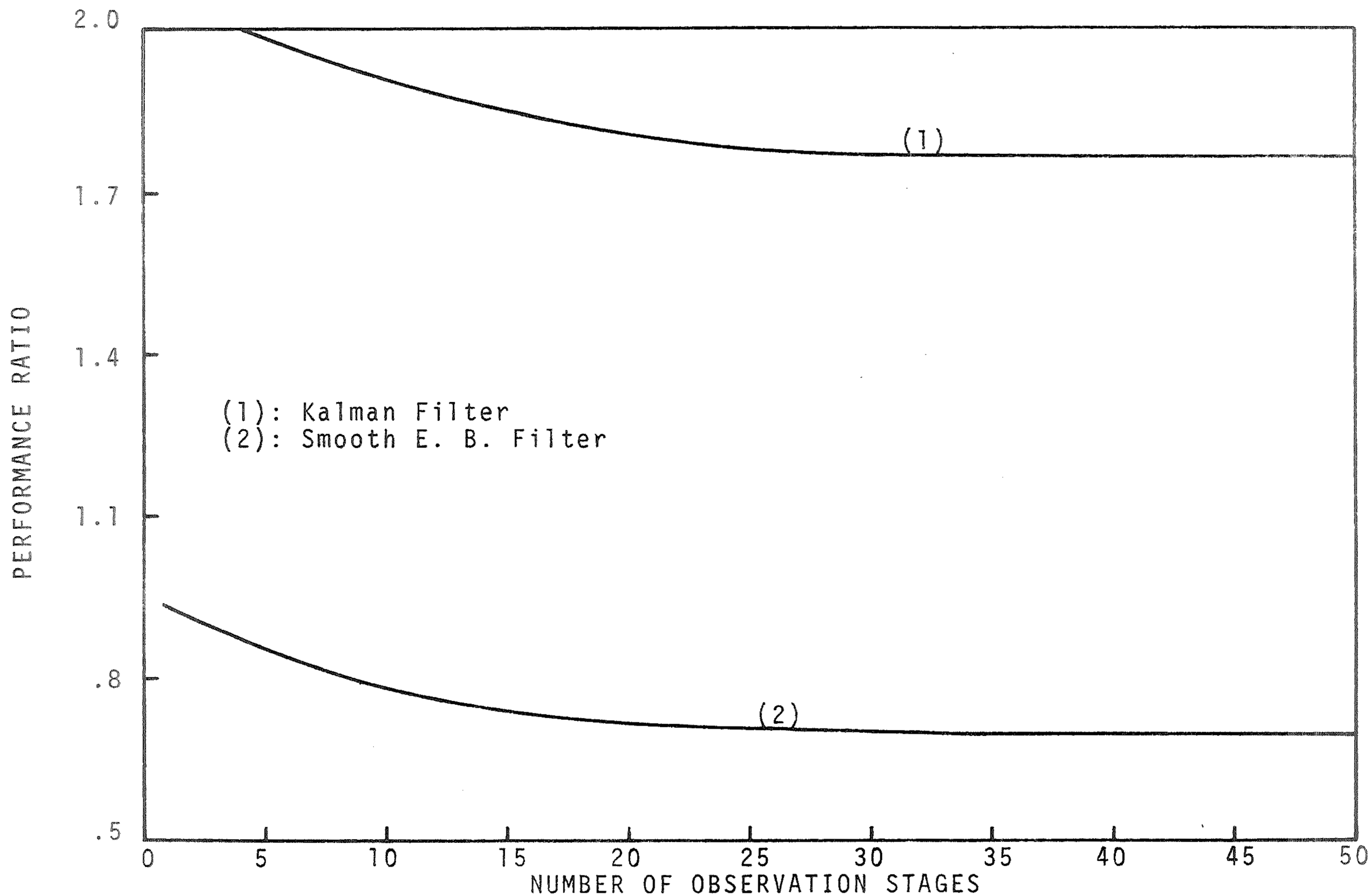


Figure 22. Performance Ratios for Smooth E. B. Filter and Kalman Filter for $z=2.0$ with U-shaped Distribution on Components of u.

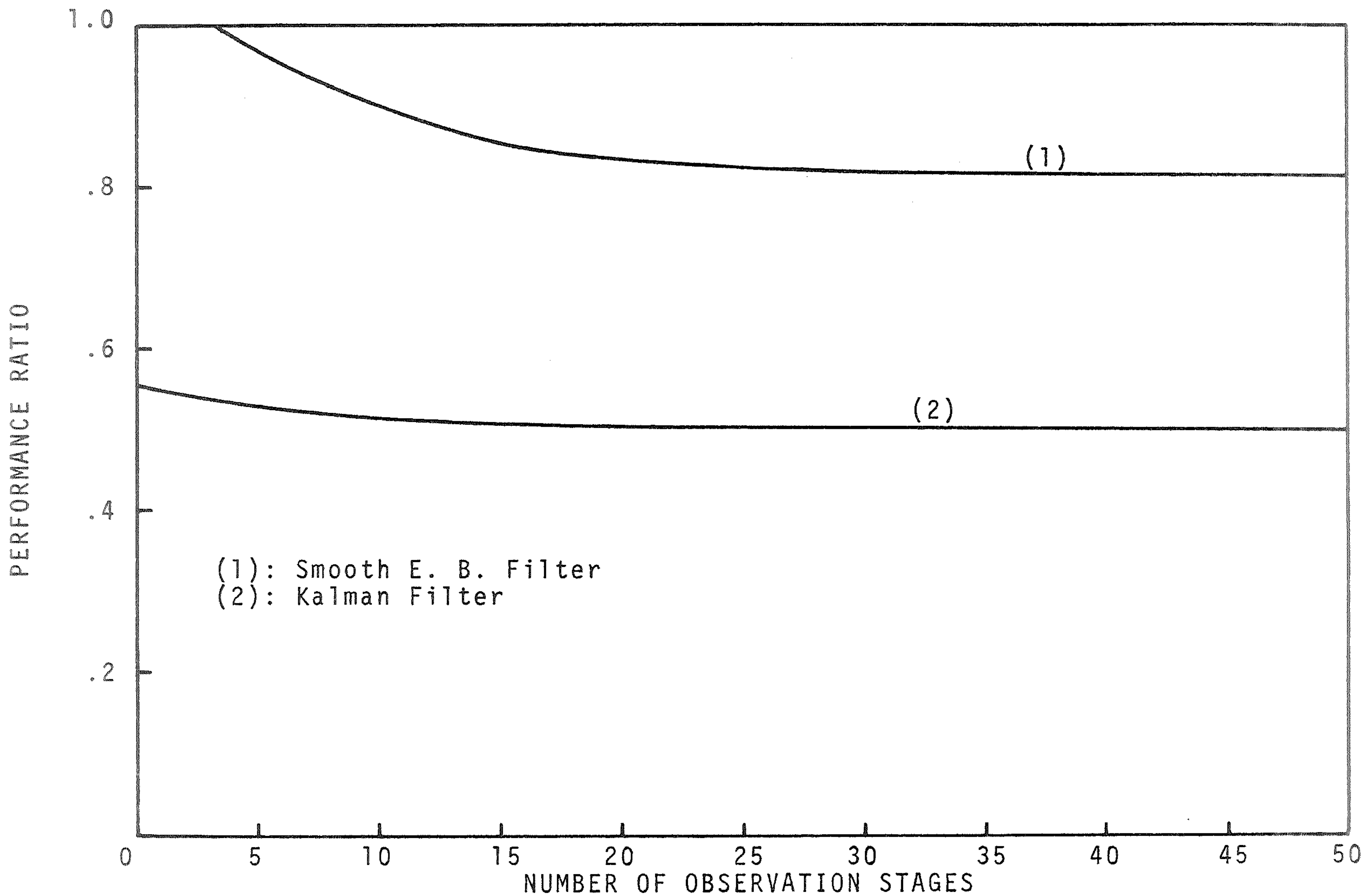


Figure 23. Performance Ratios for Smooth E. B. Filter and Kalman Filter for z=2.0 with Multivariate Normal Distribution on u.

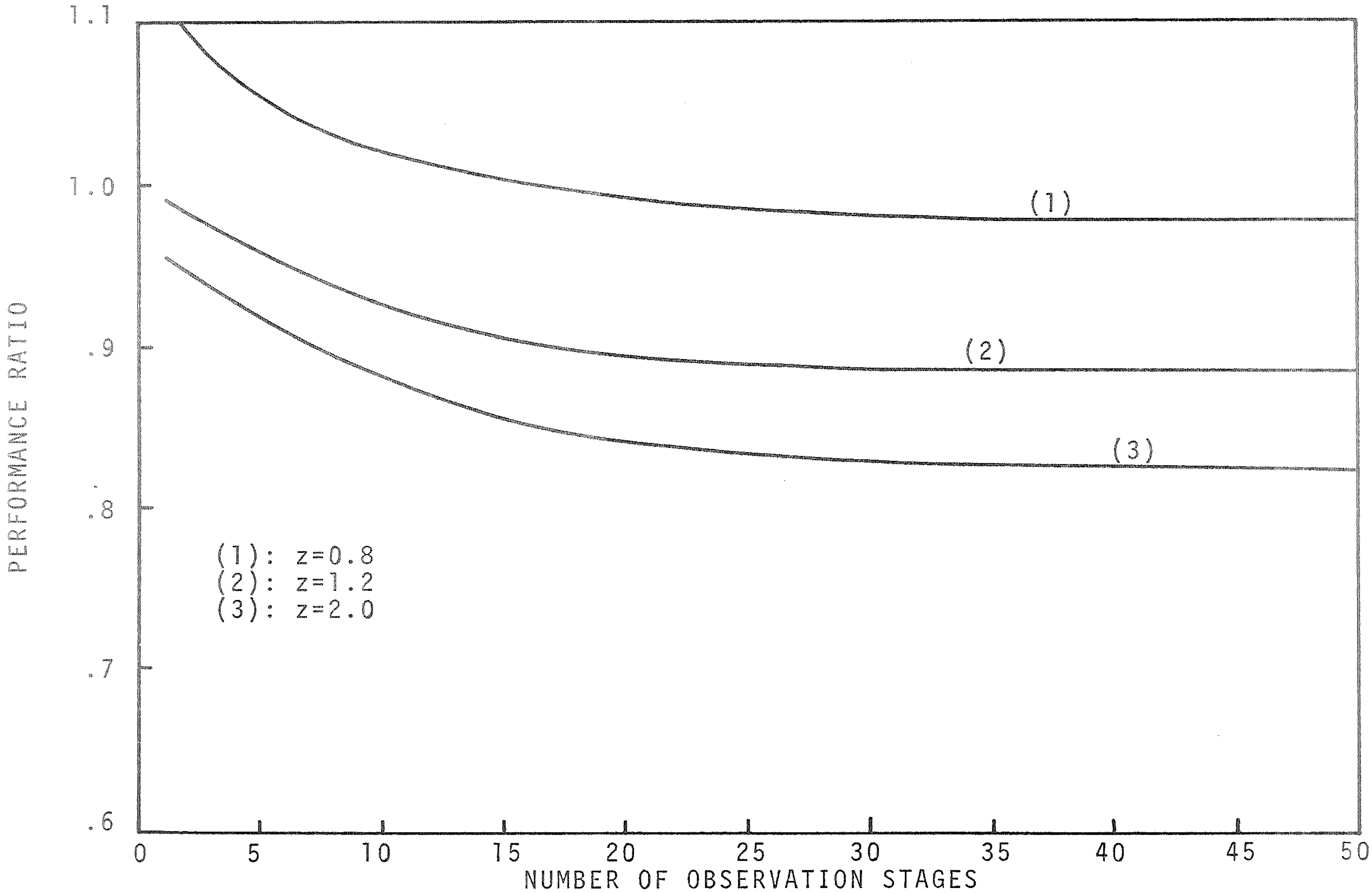


Figure 24. Performance Ratios for Smooth E. B. Filter for z=0.8, 1.2 and 2.0 with L-shaped Distribution on Components of u.

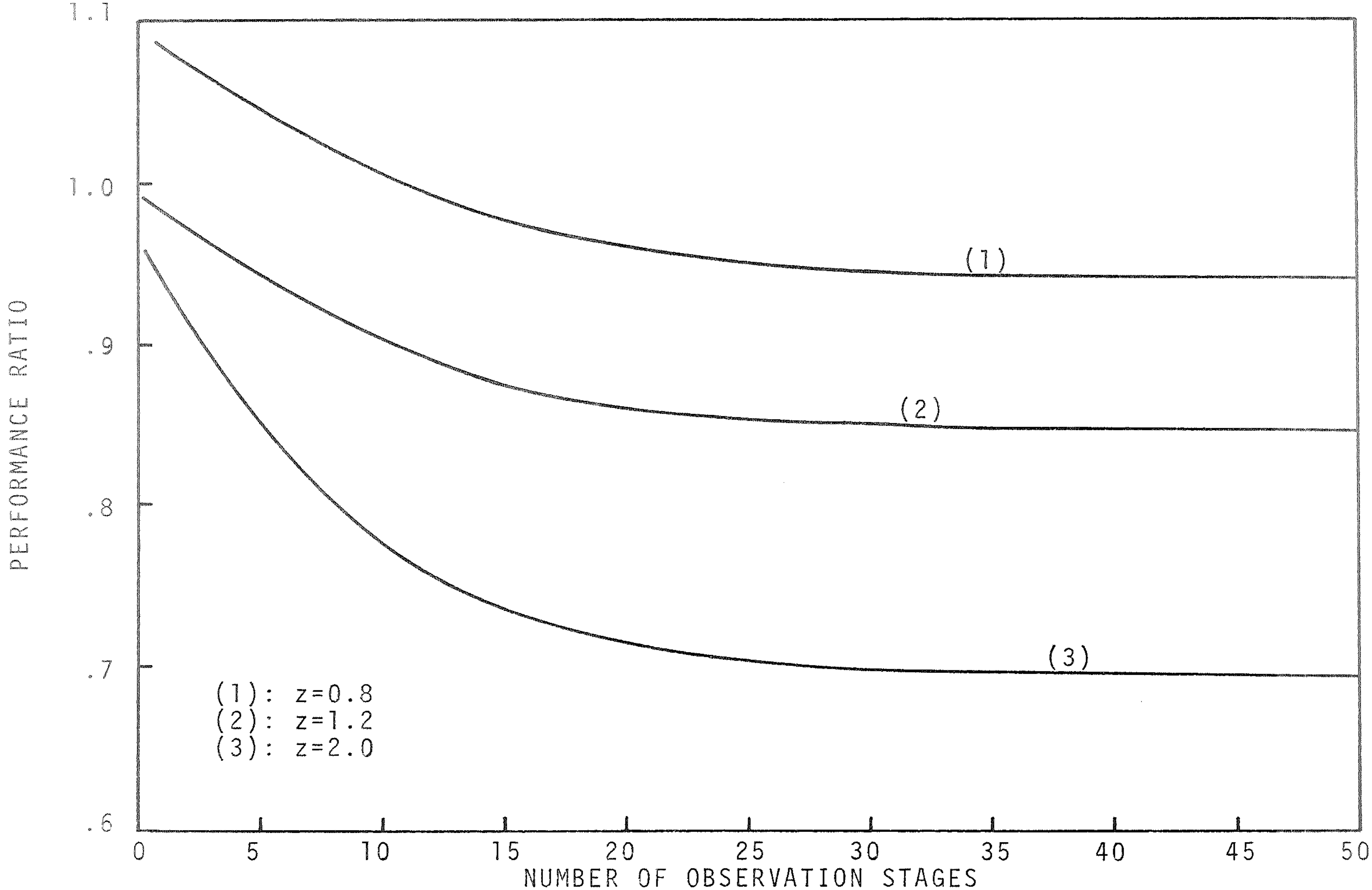


Figure 25. Performance Ratios for Smooth E. B. Filter for z=0.8, 1.2 and 2.0 with U-shaped Distribution on Components of u.

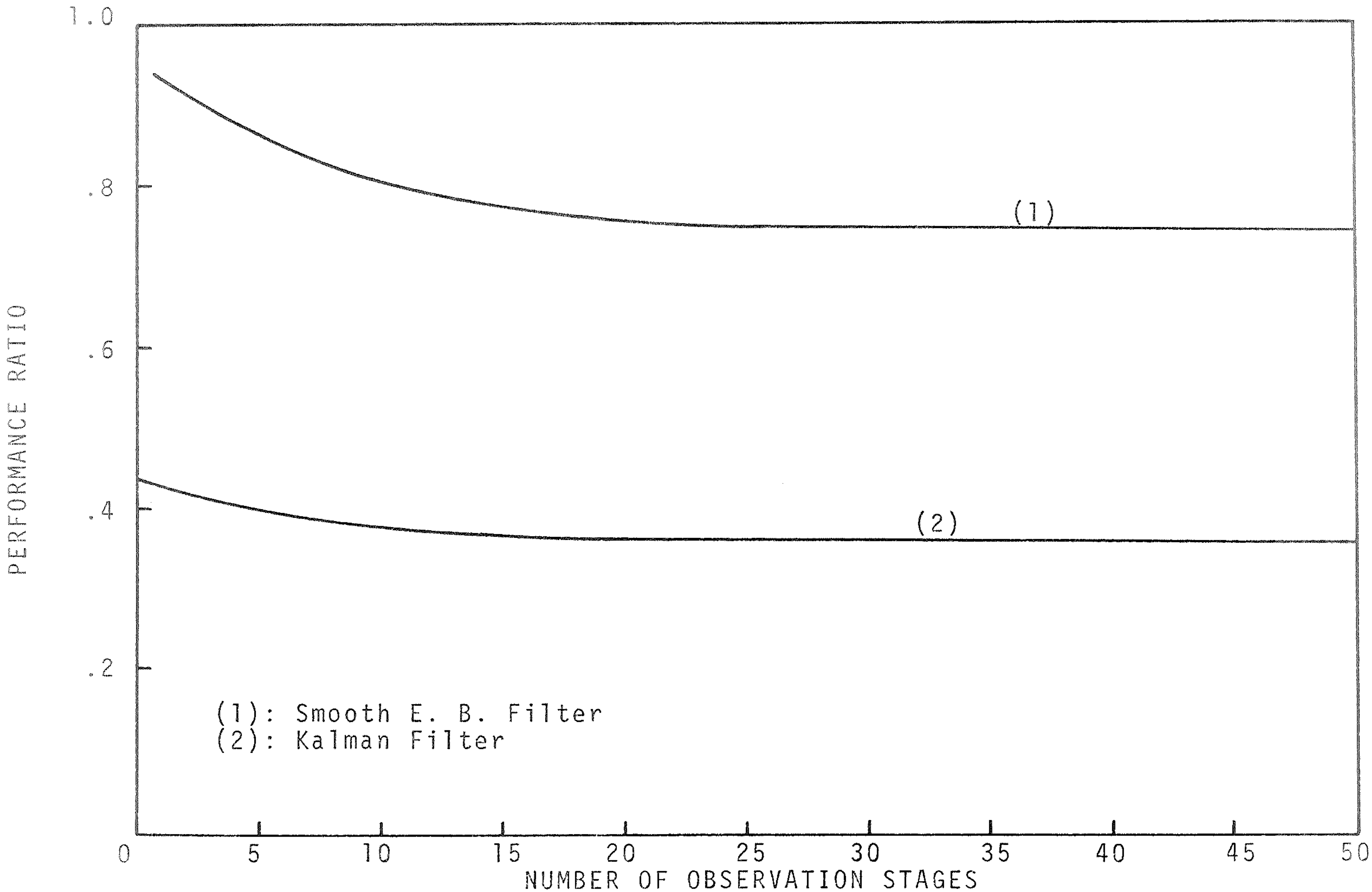


Figure 26. Performance Ratios for Smooth E. B. Filter and Kalman Filter for Changing Q, mean z=2.0 with L-shaped Distribution on Components of u.

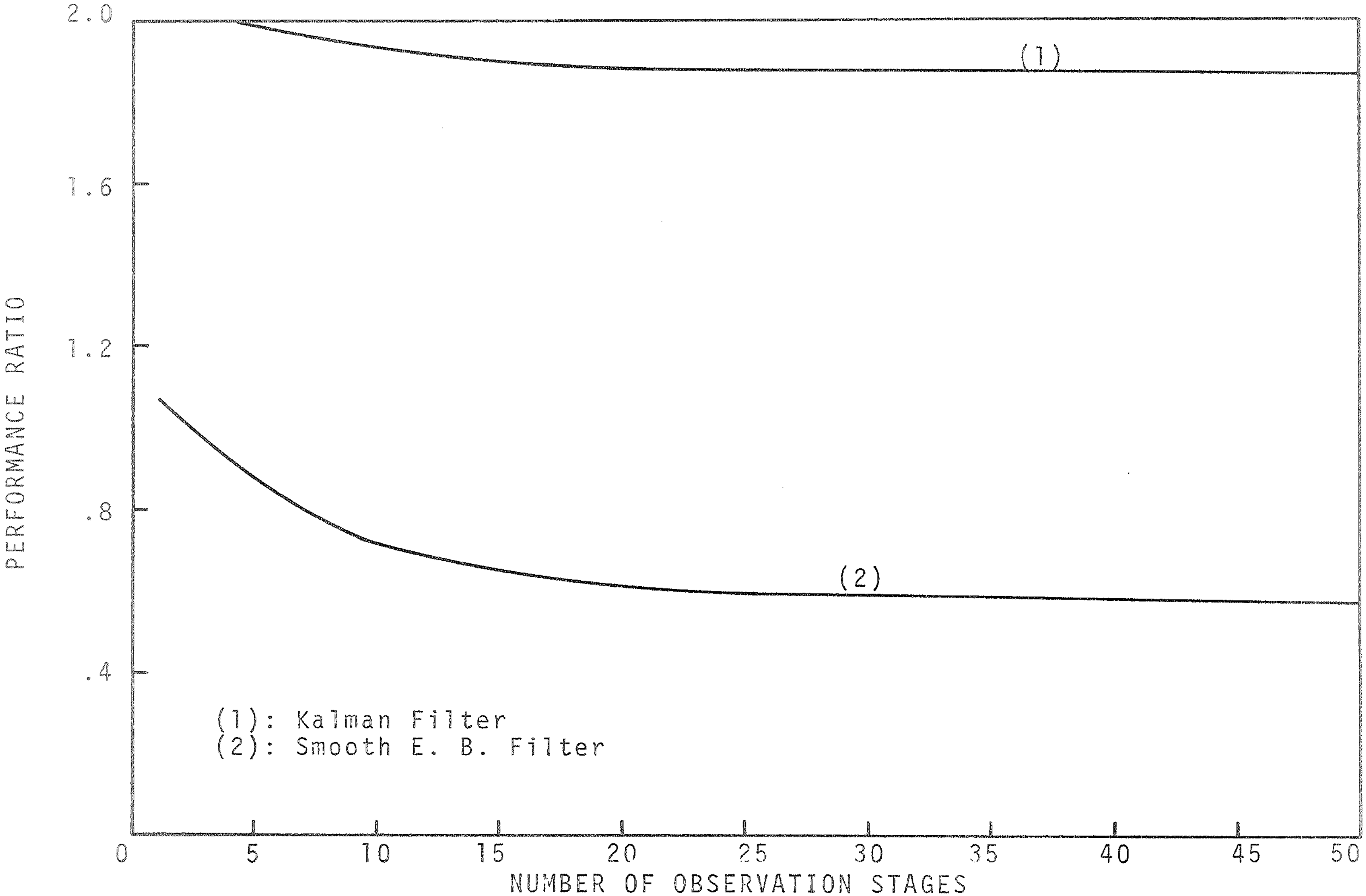


Figure 27. Performance Ratios for Smooth E. B. Filter and Kalman Filter for Changing Q, mean z=2.0 with U-shaped Distribution on Components of u.

CHAPTER IV

GENERAL CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

This chapter presents a summary of the conclusions of this research and suggests directions for future research.

General Conclusions

The primary objective of this research was to solve the problem of estimating the state vector of a discrete time linear system using linear sets of observations and without requiring any distributional assumptions whatsoever on the state disturbance error vector. Empirical Bayes decision procedures are used to obtain an estimator for the state vector of such a system. The distributional assumptions on the state disturbance error that were relaxed are: (i) the shape or form of the distribution, (ii) the necessity of any knowledge about the covariance matrix of this error, and (iii) the assumption of a known mean vector which is usually zero.

The relaxation of the first assumption is useful in trajectory estimation problems where usually no valid physical basis exist for assuming a specified form for the distribution on the state disturbance error vector. It is also significant in the case of physical processes that are truly characterized by a nonlinear system, which has been linearized to arrive at the linear system under consideration. The relaxation of the covariance assumption is significant in the computation stage since storage need not be allocated for these matrices. Also, in using the Kalman filter, these matrices must be either assumed a priori or estimated from the observation data. A poor estimator or invalid assumption may have a significant effect on the results obtained. Not requiring knowledge of these matrices thus removes one possible source of error. A similar argument can be put forth regarding the assumption of a known state disturbance error vector.

The only assumptions that were required in developing the empirical Bayes filter equations are: (i) the state disturbance error be uncorrelated over time and be independent of the observation error, and (ii) the distribution on the state disturbance error be time-invariant and thus remain stationary.

Some distributional assumptions on the initial state vector are also relaxed. The mean of this initial state vector is required for starting the estimation process, but knowledge of the covariance matrix and the form of the the distribution is not necessary. Although the sensitivity of the filter to the assumed initial state has not been investigated in this research, it is conjectured that using an arbitrary starting value for the initial state vector will hamper the performance of the filter until sufficient observations have been obtained to dampen the effect of this starting value.

The results of Monte Carlo simulation show that for the problem investigated the performance of the smooth empirical Bayes filter did not depend significantly on the true, but unknown, distribution on the state disturbance error vector. On the other hand, the Kalman filter performed poorly with a non-unimodal state disturbance error distribution compared to its performance for the unimodal distributions. Also, the Kalman filter required accurate knowledge of the covariance matrices of the state disturbance error vector and the initial state vector. Although the smooth empirical Bayes filter requires storage for all the past observation data, some savings are obtained by not having to store the covariances matrices. It was found in Chapter II, that the performance of the estimator for the mean of a multivariate normal random process was not greatly affected by using a curtailed set of past observations. It is conjectured here that the same will be true for the smooth empirical Bayes filter. The ideal number of past observations to use will depend on the loss of accuracy as compared to the savings in storage. In the case of the smooth estimator for the mean of a multivariate normal random process it was observed that the improvement in the performance of the estimator was almost insignificant as the number of past observations used increased beyond fifteen. Fifteen past observations gave nearly as good a performance as the use of all past observations. The same is true for the smooth empirical Bayes filter when used in a linear system.

The smooth empirical Bayes estimator for the mean of a multivariate normal random process was developed as a tool for application to the problem of estimating the state vector of the linear system. However, this by itself presents an original contribution to the existing empirical Bayes literature for estimation in the multivariate normal distribution. A detailed investigation of the properties of this estimator was not undertaken in this research since it was not a primary objective. It was observed by Martz and Krutchkoff [24] that the performance of their empirical Bayes estimator depended on the dimension of the vector being estimated. As the dimension increased beyond three the performance dropped off sharply. Since the smooth empirical Bayes estimator developed in this research was found to perform quite satisfactorily for a six-dimensional vector process, it can be expected to perform at least as good for a smaller dimensional process. The dimension of the process was chosen to be six with a foresight to latter application in trajectory estimation.

This smooth estimator did require more knowledge than the usual maximum likelihood estimator against which the comparisons were made. The usual maximum likelihood estimator cannot use past observations, if the randomness of the mean vector is assumed. Although the existence of the covariance matrix is required to form the likelihood function, the actual value of this conditional covariance matrix is not necessary for using the maximum likelihood estimator. In many practical situations past observations are available and the conditional covariance matrix can be estimated from the observations obtained at each time point, if it is unknown. Hence, the additional information required for the use of the smooth empirical Bayes estimator is not unduly restrictive in many practical situations.

A summary quantity z which summarized the information in the relative values of the conditional covariance matrix of the observable random vector and the covariance matrix of the prior distribution of the parameter, is used to index different sets of data used in the Monte Carlo simulation. Observations made from the results of the simulation justified the use of this summary value as a suitable index of performance. In a practical situation, the prior covariance matrix of the parameter remains unknown and at best can be estimated by using all available past information. The summary quantity z may thus be estimated using the two estimated covariance matrices, and this estimate of z can be revised at each time point as more data become available. These estimated z values can then be used as an indication of the improvement to be expected over the performance of the usual maximum likelihood estimator.

For the case of the smooth empirical Bayes filter for use in a linear system, an expression is obtained for the summary quantity z. However, the computation of this quantity in a practical situation requires knowledge of the covariance matrix of the state disturbance error vector. Since this quantity will be used only as an indication of the performance of the filter, an estimate of this covariance matrix can be used if a suitable estimator is available.

A performance measure of the filter is defined as the ratio of the trace of the mean squared error matrix to the trace of the observation error covariance matrix. A decrease in the performance measure is the same as an improvement in the performance of the filter. It is observed that the performance of the smooth empirical Bayes filter improves as the index z increases, providing all other parameters remain unchanged.

Additional Areas for Application of the Techniques Developed in this Research

The estimator for the mean of a multivariate normal random process has many applications. For an example, consider a clinic with a number of patients, each of whom undergoes a series of tests at certain intervals of time. The tests may determine the counts red blood cells, white blood cells or some specific bacteria and the concentrations of certain salts and sugar in the urine. Since these are all observations, the true values of the counts and the concentrations are unknown. If it is assumed that the true values have a time invariant distribution and that the observations are conditionally normally distributed with these true values as the means, then the situation in this case will be identical to the one described in Chapter II. Univariate analysis should not be used in this case since the counts and the concentrations may be correlated and thus a univariate analysis would ignore valuable related information. A number of observations will be required at each time point so that the conditional covariance matrix can be estimated if it is unknown.

For another example consider a machine which processes batches of different types of components. Suppose that each type of component has certain significant dimensions which are measured for a sample of fixed size from each batch. These measured dimensions will vary from sample to sample and also within a sample from one unit to another. The mean dimensions for a batch can be used for quality control purposes. Assuming that the measured dimensions for a sample are normally distributed about the mean dimensions for that batch, the estimator for the mean of a multivariate normal random process can be used to estimate the mean dimensions. The significant dimensions for all types of components should be put together in a vector form since the deviations of the measured dimensions for different components may be correlated due to the wear, tear and deterioration of the machine with each batch being processed.

The smooth empirical Bayes filter for estimating the state vector in a linear dynamic system can be used in forecasting. Consider for example forecasting of the prices of different stocks in the stock market. It is possible that the prices for the different stocks being considered are correlated. By modelling the time series of the prices as a linear dynamic system and considering forecasts obtained from other sources as the observations the smooth empirical Bayes filter can be used for forecasting the future prices. The state transition and the state-to-observations relationship matrix will be required to be estimated from the past observations.

Directions for Future Research

As an immediate extension of this research, investigation could be undertaken to modify the smooth empirical Bayes filter in order to solve the problem of estimating the state vector of a linear system when the state disturbance error is correlated in time. A random process correlated in time is usually referred to as being "colored". A completely arbitrary dependency structure is difficult to handle and certain restrictions in the dependency structure may need to be introduced. (See Martz [23] for a possible type of dependency structure).

As another extension of this research, a new filter could be developed which would relax the distributional assumptions on the observation error as well as the state disturbance error. The assumption of Gaussian observation error used in this research was justified in Chapter I using the most common form of the "Central Limit Theorem". For an example, assume that two observation devices were available and that their inherent characteristics were significantly different so that the observation error introduced with the use of one was significantly different from the observation error introduced with the use of the other. If it is also assumed that only one device can be used at a time and that the choice depends on certain characteristics of the system, then the observation error in this case could conceivably be non-unimodal and the assumption of Gaussian error could not hold.

Another area for future research is the area of non-linear filtering. An initial approach could be to define the system in a manner that would permit the separation of the state disturbance error so that it may be estimated using empirical Bayes decision procedures. This may require the functions involved in the system to be single-valued and to possess unique inverses. The inverse functions may be explicitly required, thus limiting the classes of functions that can be considered, although, theoretical stipulation of an inverse function is possible for every function which has a unique inverse. Continuity may also be required to be assumed for these functions.

The area of non-linear filtering should provide a tremendous opportunity for further research. Once the problem mentioned above is solved, further relaxations of the assumptions required to obtain this solution could open up more areas of research.

Conclusions

Two original contributions have been made by this research. A smooth empirical Bayes estimator has been developed for estimating the mean of a multivariate normal random process. This estimator was demonstrated to perform quite satisfactorily by means of Monte Carlo simulation of sample problems. This estimator was then used to develop a smooth empirical Bayes filter for estimating the state vector of a discrete time linear system with linear sets of observations. By means of a realistic example in trajectory estimation the performance of this filter has been examined using Monte Carlo simulation. In addition, the results of this research should serve as a foundation for future research in the areas indicated in the previous section.

LIST OF REFERENCES

1. Abramson, P. D. "Simultaneous Estimation of the State and Noise Statistics in Linear Dynamical Systems," Ph. D. Thesis, Massachussetts Institute of Technology, Cambridge, Massachussetts (1968).

2. Bass, R. W., Norum, V. D., and Schwarts, L., "Optimal Multichannel Nonlinear Filtering, "Journal of Mathematical Analysis and Applications, (1966).

3. Battin, R. H., Astronautical Guidance, McGraw-Hill (1964).

4. Bennett, G. K., "Smooth Empirical Bayes Estimation with Application to the Weibull Distribution," Ph. D. Thesis, Texas Tech University, Lubbock, Texas (1970).

5. Bryson, A. E. and Johansen, D. E., "Linear Filtering for Time-varying Systems Using Measurements Containing Colored Noise," IEEE Transactions, Automatic Control (1965).

6. Bucy, R. S., "Nonlinear Filtering," IEEE Transactions, Automatic Control (1965).

7. Bucy, R. S., "Optimum Finite Time Filters for a Special Non-stationary Class of Inputs," Internal Report, BBD 600, Johns Hopkins University/Applied Physics Laboratory (1959).

8. Bucy R. S., "Optimal Filtering with Correlated Noise," Journal of Mathematical Analysis and Applications (1967).

9. Bucy, R. S., Englar, T. S., and Kalman, R. E., "Fundamental Study of Adaptive Systems," Technical Report ASD-TR-61-27, Vol. I and II, (1962 and 1964).

10. Bucy, R. S. and Follin, J. W., "Adaptive Finite Time Filtering," IRE Transactions, Automatic Control (1962).

11. Bucy, R. S. and Joseph, P. D., Filtering for Stochastic Processes with Applications to Guidance, Interscience Publishers (1968).

12. Bucy, R. S. and Kalman, R. E., "New Results in Linear Filtering and Prediction Theory," Journal of Basic Engineering, ASME, SER. D (1961).

13. Cacoullos, T., "Estimation of a Multivariate Density," Annals of Institute of Statistical Mathematics (Tokyo), 18 (1966).

14. Cramer, H., Mathematical Methods of Statistics, Princeton University Press (1964).

15. Deutsch, R., Estimation Theory, Prentice Hall (1965).

16. Erzberger, H., "Application of Kalman Filtering to Error Correlation of Inertial Navigators," NASA TN D-3874.

17. Joseph, P. D., "Sub-optimal Linear Filtering," TRW-Space Technology Laboratories IOC 9321.4-653 (1963).

18. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, ASME (1960).

19. Kalman, R. E., "New Methods and Results in Linear Prediction and Estimation Theory," Technical Report 61-1, Research Institute of Advance Study, Baltimore, Maryland (1961).

20 Kalman, R. E. and Englar, T. S., "A User's Manual for ASP-C," NASA CR-475 (1965).

21. Kolmogorov, A. N., "Interpolation und Extrapolation von Stationaren Zufalligen Folgen," Bulletin of Academic Sciences, USSR, SER. MATH. (1941).

22. Layton, D. and Sheffield, P. S., "Kalman Filter Study," TRW Systems Report 99900-6099-R000 (1966).

23. Martz, H. F., Jr., "Multivariate Extended Empirical Bayes Estimation," Proceedings of the Symposium on Empirical Bayes Estimation and Computing in Statistics, Texas Tech University, Lubbock, Texas (1969).

24. Martz, H. F., Jr. and Krutchkoff, R. G., "Empirical Bayes Estimators in a Multiple Linear Regression Model," Biometrica, 56 (1969).

25. Meditch, J. S., "Sub-optimal Filtering for Continuous Dynamic Processes," Aerospace Report, TDR-469 (5107-35) (1964).

26. Robbins, H., "An Empirical Bayes Approach to Statistics," _Proceedings of Third Berkeley Symposium on Mathematical Statistics and Probability_, University of California (1955).

27. Robbins, H., "The Empirical Bayes Approach to Statistical Decision Problems," _Annals of Mathematical Statistics_ (1964).

28. Shellenbarger, J. C., "Estimation of Covariance Parameters for an Adaptive Kalman Filter," _Ph. D. Thesis_, Iowa State University, Ames, Iowa (1966).

29. Smith, G. L., "Sequential Estimation of Observation Error Covariances in a Trajectory Estimation Problem," AIAA Journal (1967).

30. Smith, G. L., Schmidt, S. F., and McGee, L. A., "Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity On-Board a Circumlunar Vehicle," NASA TR R-135 (1962).

31. Starich, D. V., "New Techniques in Sub-optimal Linear Filtering," _Ph. D. Dissertation_, Northwestern University, Evanston, Illinois (1967).

32. Thomas, D. G., "Computer Methods for Generating Pseudo-Ramdon Numbers from Pearson Distributions and Mixtures of Pearson and Uniform Distribution," _M. S. Thesis_, Virginia Polytechnic Institute, Blacksburg, Virginia (1966)

33. Wiener, N., _Extrapolation, Interpolation and Smoothing of Stationary Time Series_, Tech. Press of M.I.T. and John Wiley (1949).

34. Wing, J. and Joseph, P. D., "Some Errors Occuring when Propagating Covariance Matrices," Space Technology Laboratories IOC 9321.4-679 (1964).

APPENDIX A

MONTE CARLO SIMULATION COMPUTER PROGRAM OF CHAPTER II

A. Title:

A Monte Carlo simulation program for the smooth empirical Bayes estimator for the mean of a multivariate normal random process developed in Chapter II.

1. Programmer--Satish J. Kamat.
2. Machine--IBM 360/50.
3. Language--Fortran IV.
4. Date Completed--Spring 1970.
5. Compiler--OS/360 (HASP II System).
6. Approximate Compile Time--0.73 minutes.
7. Approximate Lines of Output--600 (excluding program listing).
8. Approximate Core-space Used--33,250 bytes.

B. Purpose:

This program simulates one hundred replications of a multivariate normal random process as described in Chapter II and obtains the estimate of the mean squared error matrix at each observation stage of the smooth empirical Bayes estimator for the mean of this process.

C. Restrictions:

This program has been written for a six-dimensional process.

D. Definitions:

1. Subroutines:

   a. MAIN--The main program acts as a monitor in simulating the random process and calls various subroutines. It also performs the computations necessary to obtain the mean squared error matrices.

   b. INPUT--This subroutine reads the input information.

   c. MTRXOP--This subroutine performs various matrix operations depending on an input parameter.

   d. GJR--This subroutine inverts the given matrix using the Gauss-Jordan-Rutishauser method with double pivoting.

   e. SUMRZ--This subroutine obtains the summary quantities defined in Chapter II.

   f. SIMLTE--This subroutine generates the random numbers for the process.

   g. EMPEST--This subroutine calculates the estimate of the mean vector of the process.

h. MTF--This subroutine performs triangular factorization on the given matrix using Kraut factorization.

i. VARCOR--This subroutine obtains the estimates of the observations using covariance correction.

j. RANDU--This subroutine generates random numbers in the range of 0.0 to 1.0.

k. GAUSS--This subroutine generates random numbers from a normal distribution.

m. PURGE2--This subroutine generates random numbers from a member of the Pearson family of distributions.

2. Subroutines Called By:

   a. MAIN--INPUT, MTF, MTRXOP, GJR, SUMRZ, PURGE2, SIMLTE, EMPEST.

   b. SUMRZ--MTRXOP, GJR.

   c. SIMLTE--GAUSS, MTRXOP.

   d. EMPEST--MTRXOP, GJR, VARCOR.

   e. GAUSS--RANDU.

3. Subscripted Variables in MAIN:

   a. RS(100)--Stores the random numbers generated by PURGE2.

   b. RDS(300)--Stores the random numbers for the elements of the mean for fifty stages.

c. R(6,6)--Covariance matrix of x.

d. R1(6,6)--Triangular factor matrix of R.

e. RINV(6,6)--Inverse of R.

f. X(6,50)--Matrix which stores the vector x for fifty stages.

g. XLS(6)--Maximum likelihood estimate of the mean.

h. ENXN(6)--First iteration estimate of the mean using smooth E. B. estimator.

i. ENXN1(6,50)--Matrix which stores the estimate ENXN for fifty stages.

j. ENXN2(6)--Second iteration estimate of the mean using smooth E. B. estimator.

k. T(6)--The mean of the normal distribution.

m. COV1(6,6,50)--Stores the mean squared error matrices for the first iteration.

n. COV2(6,6,50)--Stores the mean squared error matrices for the second iteration.

4. Important Non-Subscripted Variables in MAIN:

a. IIIX--The seed for PURGE2.

b. NREP--Number of replications.

c. VAR--Variance of numbers generated by PURGE2.

d. LN--Index of replication.

E. Input and Output:

1. Input:

Card 1--(I4)--ISEED, the seed for GAUSS.

Card 2--(I4)--K, the number of observations generated at each stage.

Cards 3 to 8--(6F10.6)--R, one row of the matrix on each card in sequence.

Cards 9 and 10--Specification cards for PURGE2.

2. Output:

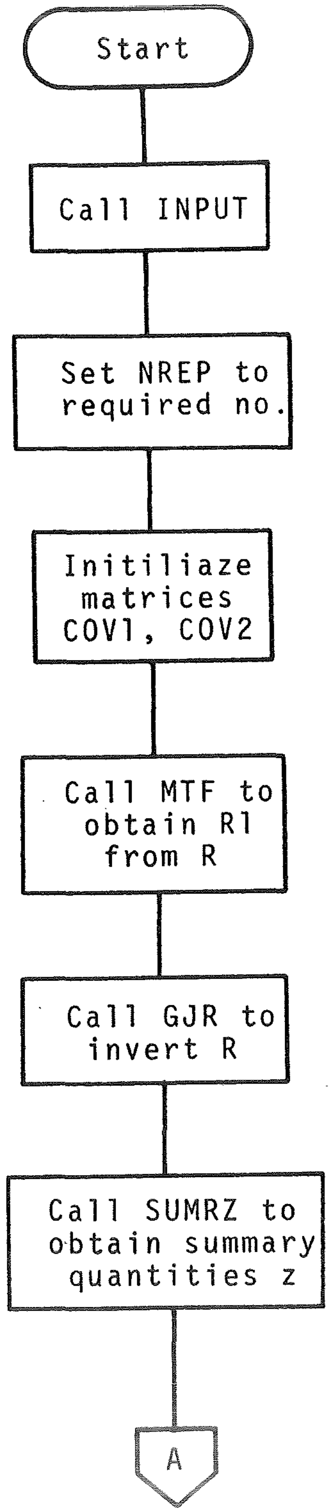
First Page--R, R1, RINV.

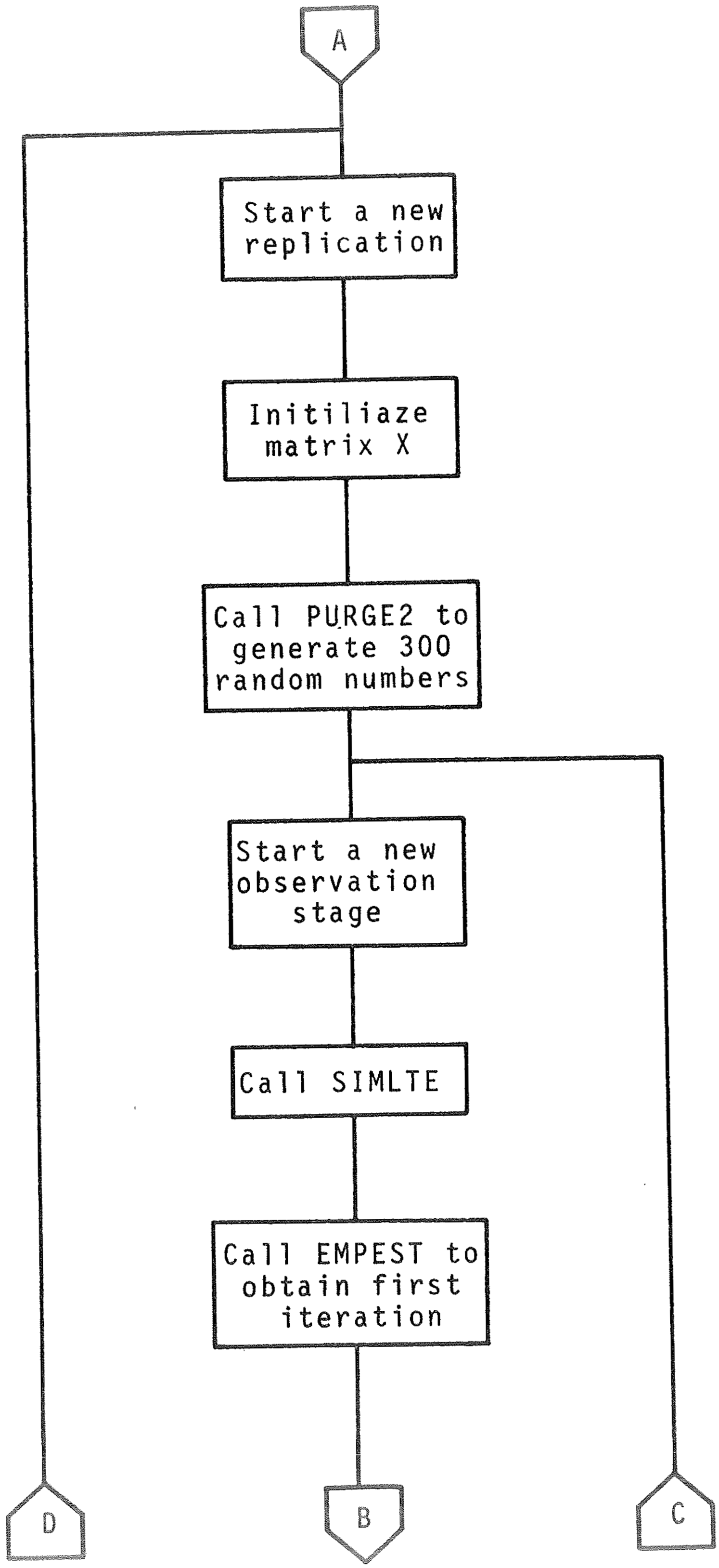
Second Page--Z, the summary quantities.

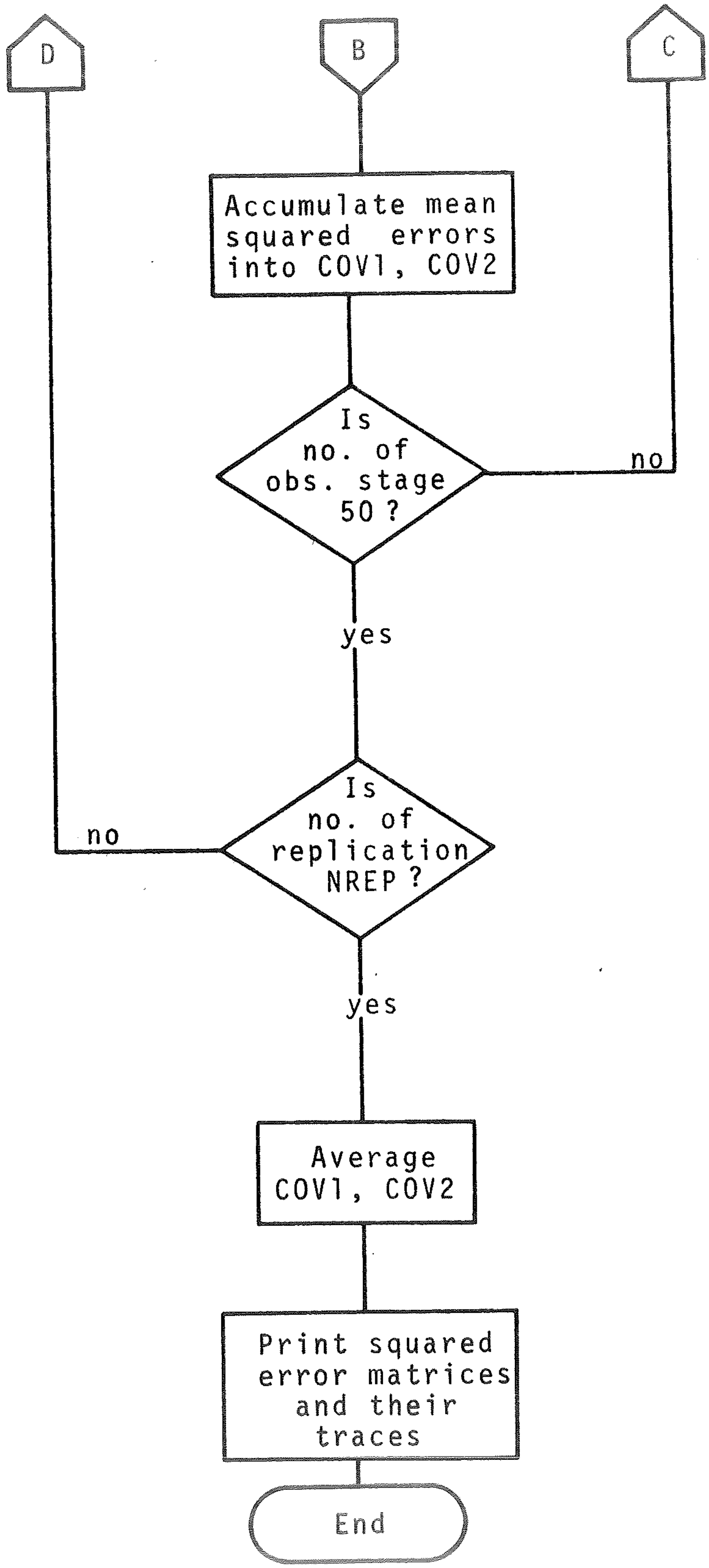
Third Page--Parameters in PURGE2.

Subsequent Pages--The mean squared error matrices, described in detail in the output.

F. Flow Chart of MAIN:




A
Start a new replication
Initiliaze matrix X
Call PURGE2 to generate 300 random numbers
Start a new observation stage
Call SIMLTE
Call EMPEST to obtain first iteration
D
B
C


D
B
C
Accumulate mean squared errors into COV1, COV2
Is no. of obs. stage 50 ?
no
yes
Is no. of replication NREP ?
no
yes
Average COV1, COV2
Print squared error matrices and their traces
End

```
      COMMON/BLK1/R/BLK2/ISEED,KI/BLK3/R1/BLK4/RDS/BLK5/X
      COMMON/Z2/RS/Z11/IIIX
      DIMENSION RS(100),RDS(300),R(6,6),R1(6,6),RINV(6,6),
     1X(6,50),XLS(6),ENXN(6),ENX1(6,50),ENXN2(6),T(6)
      DIMENSION COV1(6,6,50),COV2(6,6,50)
      CALL INPUT
    1 FORMAT (6F14.8)
      PRINT 1, R
      IIIX = 519
      NREP = 100
      DO 16 I=1,6
      DO 16 J=1,6
      DO 16 K=1,50
      COV1(I,J,K) = 0.
   16 COV2(I,J,K) = 0.
      CALL MTF (R1,R,6)
      PRINT 1, R1
      DO 5 I=1,6
      DO 5 J=1,6
    5 RINV(I,J) = R(I,J)/FLOAT(KI)
      CALL MTRXOP (RINV,R1,R,6,6,6,6,5)
      ZERO = 0.
      CALL GJR (RINV,6,ZERO,MSING,6,6)
      IF (MSING.EQ.2)   GO TO 200
      PRINT 1, RINV
      VAR = 0.5
      CALL SUMRZ (R,RINV,VAR)
      DO 95 LN=1,NREP
      DO 15 I=1,6
      DO 15 J=1,50
   15 X(I,J) = 0.
      DO 10 I=1,3
      IF (LN.GT.1)   GO TO 17
      IF (I.EQ.1)    CALL PURGE2(1,5)
      IF (I.GT.1)    CALL PURGE2(2,5)
      GO TO 18
   17 CALL PURGE2(2,5)
   18 CONTINUE
      IJ = (I-1)*100
      DO 10 J=1,100
   10 RDS(IJ+J) = RS(J)
      DO 90 K=1,50
      KN = (K-1)*6
      CALL SIMLTE (K)
      DO 20 I=1,6
      T(I) = RDS(KN+I)
   20 XLS(I) = X(I,K)
      CALL EMPEST (K,RINV,X,ENXN,1,XLS)
      DO 30 I=1,6
```

```
 30   ENX1(I,K) = ENXN(I)
      CALL EMPEST (K,RINV,ENX1,ENXN2,2,XLS)
 35   FORMAT (6F10.6,/)
      DO 40 I=1,6
      DO 40 J=1,6
      COV1(I,J,K) = COV1(I,J,K)+(ENXN(I)-RDS(KN+I))*(ENXN(
     1J)-RDS(KN+J))
 40   COV2(I,J,K) = COV2(I,J,K)+(ENXN2(I)-RDS(KN+I))*(ENXN
     12(J)-RDS(KN+J))
 90   CONTINUE
 95   CONTINUE
      DO 100 I=1,6
      DO 100 J=1,6
      DO 100 K=1,50
      COV1(I,J,K) = COV1(I,J,K)/FLOAT(NREP)
100   COV2(I,J,K) = COV2(I,J,K)/FLOAT(NREP)
      DO 150 K=1,50
      PRINT 102, K
      PRINT 104, ((COV1(I,J,K),J=1,6),I=1,6)
      TR = 0.
      DO 120 I=1,6
120   TR = TR+COV1(I,I,K)
      PRINT 105, TR
      PRINT 103, K
      PRINT 104, ((COV2(I,J,K),J=1,6),I=1,6)
      TR = 0.
      DO 130 I=1,6
130   TR = TR+COV2(I,I,K)
      PRINT 105, TR
150   CONTINUE
102   FORMAT (1H0,78H COVARIANCE MATRIX FOR E.B. ESTIMATOR
     1 (FIRST APPROXIMATION) FOR EXPERIMENT NO.,I3,/)
103   FORMAT (1H0,79H COVARIANCE MATRIX FOR E.B. ESTIMATOR
     1 (SECOND APPROXIMATION) FOR EXPERIMENT NO.,I3,/)
104   FORMAT (1X,6F18.8)
105   FORMAT (1H),29H TRACE OF THE ABOVE MATRIX IS,F18.8)
200   CALL EXIT
      END


      SUBROUTINE INPUT
      COMMON/BLK1/R/BLK2/ISEED,K
      DIMENSION R(6,6)
  5   FORMAT (I4)
 15   FORMAT (6F10.6)
      READ 5, ISEED,K
      DO 20 I=1,6
 20   READ 15, (R(I,J),J=1,6)
      RETURN
      END
```

```
      SUBROUTINE MTRXOP (A,B,C,K,L,M,N,KOP)
C
C     KOP = 1 .... C = AXB
C         = 2 .... C = A+B
C         = 3 .... C = A-B
C         = 4 .... B = A TRANSPOSE
C         = 5 .... C = A
C
      DIMENSION A(K,L),B(M,N),C(K,N)
      GO TO (10,30,50,70,90), KOP
   10 DO 20 J=1,K
      DO 20 J=1,N
      C(I,J) = 0.
      DO 20 IJ=1,M
   20 C(I,J) = C(I,J)+A(I,IJ)*B(IJ,J)
      GO TO 120
   30 E = 1.
   35 CONTINUE
      DO 40 I=1,K
      DO 40 J=1,L
   40 C(I,J) = A(I,J)+E*B(I,J)
      GO TO 120
   50 E = -1.
      GO TO 35
   70 DO 80 I=1,L
      DO 80 J=1,K
   80 B(I,J) = A(J,I)
      GO TO 120
   90 E = 0.
      GO TO 35
  120 RETURN
      END



      SUBROUTINE GJR (A,NN,EPS,MSING,IS,IS1)
      INTEGER P,Q
      DIMENSION A(IS,IS1),B(125),C(125),P(125),Q(125)
      N = NN
      IF (EPS.LE.0.)   EPS = 1.E-26
      MSING = 1
      DO 10 K=1,N
      PIVOT = 0.
      DO 20 I=K,N
      DO 20 J=K,N
      IF (ABS(A(I,J))-ABS(PIVOT))  20,20,30
   30 PIVOT = A(I,J)
      P(K) = I
      Q(K) = J
```

```
 20   CONTINUE
      IF (ABS(PIVOT)-EPS)  40,40,50
 50   IF (P(K)-K) 60,80,60
 60   DO 70 J=1,N
      L = P(K)
      Z = A(L,J)
      A(L,J) = A(K,J)
 70   A(K,J) = Z
 80   IF (Q(K)-K)  85,90,85
 85   DO 100 I=1,N
      L = Q(K)
      Z = A(I,L)
      A(I,L) = A(I,K)
100   A(I,K) = Z
 90   CONTINUE
      DO 110 J=1,N
      IF (J-K)  130,120,130
120   B(J) = 1./PIVOT
      C(J) = 1.
      GO TO 140
130   B(J) = -A(K,J)/PIVOT
      C(J) = A(J,K)
140   A(K,J) = 0.
110   A(J,K) = 0.
      DO 10 I=1,N
      DO 10 J=1,N
 10   A(I,J) = A(I,J)+C(I)*B(J)
      DO 155 M=1,N
      K = N-M+1
      IF (P(K)-K)  160,170,160
160   DO 180 I=1,N
      L = P(K)
      Z = A(I,L)
      A(I,L) = A(I,K)
180   A(I,K) = Z
170   IF (Q(K)-K)  190,155,190
190   DO 150 J=1,N
      L = Q(K)
      Z = A(L,J)
      A(L,J) = A(K,J)
150   A(K,J) = Z
155   CONTINUE
151   RETURN
 40   PRINT 45, P(K),Q(K),PIVOT
 45   FORMAT (16H0SINGULAR MATRIX3H I=I3,3H J=I3,7H PIVOT
     1=E16.8/)
      MSING = 2
      GO TO 151
      END
```

```
      SUBROUTINE SUMRZ (R,RINV,VAR)
      DIMENSION R(6,6),RINV(6,6),P(6,6),Q(6,6),Z(6)
      DO 10 I=1,6
      DO 10 J=1,6
      P(I,J) = 0.
      IF ( I.EQ.J )   P(I,J) = 1./VAR
10    CONTINUE
      CALL MTRXOP (P,RINV,Q,6,6,6,6,2)
      ZERO = 0.
      CALL GJR (Q,6,ZERO, MSING,6,6)
      IF (MSING.EQ.2)   STOP
      CALL MTRXOP (R,Q,P,6,6,6,6,3)
      DO 20 I=1,6
20    Z(I) = P(I,I)/Q(I,I)
25    FORMAT (1H1,40H THE SUMMARY QUANTITIES Z ARE AS FOLL
     1OWS,/)
26    FORMAT (1X,6F10.3)
      PRINT 25
      PRINT 26, Z
      RETURN
      END


      SUBROUTINE SIMLTE (N)
      COMMON/BLK2/ISEED,K/BLK3/R1/BLK4/RDS/BLK5/X
      DIMENSION R1(6,6),RDS(300),X(6,50),Y(6),Z(6)
      IN = (N-1)*6
      DO 20 I=1,K
      DO 10 J=1,6
      CALL GAUSS (ISEED,1.,0.,RN)
10    Y(J) = RN
      CALL MTRXOP (R1,Y,Z,6,6,6,1,1)
      DO 15 J=1,6
15    X(J,N) = (Z(J)+RDS(IN+J))/FLOAT(K)+X(J,N)
20    CONTINUE
      RETURN
      END


      SUBROUTINE EMPEST (K,RINV,TX,ENX,IN,XN)
      COMMON/BLK1/R/BLK4/RDS
      DOUBLE PRECISION EB,Q2,DEN,QA
      DIMENSION RDS(300),T(6),RINV(6,6),P(6,6),XN(6),Q(6),
     1XJ(6),X(6,50),PQ(6),Q1(6),Q2(6),ENX(6),R(6,6)
      DIMENSION TX(6,50),Q3(6)
      IF(IN.GT.1)   GO TO 25
      AK = K
      A = AK**(-.04)
```

```
      KN = (K-1)*6
      DO 10 I=1,6
10    T(I) = RDS(KN+I)
      DO 20 I=1,6
      DO 20 J=1,6
      P(I,J) = RINV(I,J)
      IF (I.EQ.J)   P(I,J) = P(I,J+1./(A*A)
20    CONTINUE
      CALL MTRXOP (RINV,XN;Q,6,6,6,1,1)
      ZERO = 0.
      CALL GJR (P,6,ZERO,MSING,6,6)
      IF (MSING.EQ.2)   STOP
25    CONTINUE
      DO 26 I=1,6
      DO 26 J=1,K
26    X(I,J) = TX(I,J)
30    CONTINUE
      DO 35 I=1,6
35    Q2(I) = 0.
      DEN = 0.
      DO 70 J=1,K
      DO 40 I=1,6
      XJ(I) = X(I,J)
40    Q1(I) = Q(I)+XJ(I)/(A*A)
      CALL MTRXOP (P,Q1,PQ,6,6,6,1,1)
      B = 0.
      DO 50 I=1,6
50    B = B+.5*Q1(I)*PQ(I)-.5*XJ(I)*XJ(I)/(A*A)
      EB = EXP(B)
      DO 60 I=1,6
60    Q2(I) = Q2(I)+Q1(I)*EB
      DEN = DEN+EB
70    CONTINUE
      DO 80 I=1,6
80    Q2(I) = Q2(I)/DEN
      DO 75 I=1,6
      QA = Q2(I)
75    Q3(I) = SNGL(QA)
      CALL MTRXOP (P,Q3,ENX,6,6,6,1,1)
      RETURN
      END


      SUBROUTINE MTF (A,B,N)
      DIMENSION A(N,N),B(N,N)
      DO 10 I=1,N
      DO 10 J=1,N
10    A(I,J) = 0.
      DO 70 J=1,N
```

```
      DO 70 I=J,N
      C = 0.
      L = J-1
      IF (L)  40,40,20
20    DO 30 K=1,L
30    C = C+A(I,K)*A(J,K)
40    CONTINUE
      IF (I-J)  50,50,60
50    A(I,J) = SQRT(B(I,J)-C)
      GO TO 70
60    A(I,J) = (B(I,J)-C)/A(J,J)
70    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VARCOR (T,X,K)
      COMMON/BLK1/R
      DIMENSION R(6,6),T(6,50),X(6,50),SUMT(6),SQT(6),COV
     1T(6)
      IF (K-1)  10,10,25
10    DO 15 I=1,6
      SUMT(I) = T(I,1)
      SQT(I) = T(I,1)*T(I,1)
15    CONTINUE
      DO 20 I=1,6
20    X(I,1) = T(I,1)
      RETURN
25    CONTINUE
      DO 30 I=1,6
      SUMT(I) = SUMT(I)+T(I,K)
30    SQT(I) = SQT(I)+T(I,K)*T(I,K)
      DO 35 I=1,6
35    COVT(I) = (SQT(I)*K-SUMT(I)*SUMT(I))/(K*(K-1))
      DO 45 I=1,6
      B = (COVT(I)-R(I,I))/COVT(I)
      IF (B.LT.0)   B = 1.0
      A = SQRT(B)
      DO 40 J=1,K
40    X(I,J) = A*T(I,J)+(1.-A)*SUMT(I)/K
45    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RANDU (IX,IY,YFL)
      IY = IX*65539
      IF (IY)  5,6,6
```

```
5   IY = IY+2147483647+1
6   YFL = IY
    YFL = YFL*.4656613E-9
    RETURN
    END


    SUBROUTINE GAUSS (IX,S,AM,Y)
    A = 0.
    DO 50 I-1,12
    CALL RANDU (IX,IY,Y)
    IX = IY
50  A = A+Y
    V = (A-6.0)*S+AM
    RETURN
    END
```

APPENDIX B

MONTE CARLO SIMULATION COMPUTER PROGRAM OF CHAPTER III

A. Title:

A Monte Carlo simulation program to simulate the sample problem in trajectory estimation presented in Chapter III.

1. Programmer--Satish J. Kamat.
2. Machine--IBM 360/50.
3. Language--Fortran IV.
4. Date Completed--Summer 1970.
5. Compiler--OS/360 (HASP II System).
6. Approximate Compile Time--0.74 minutes.
7. Approximate Lines of Output--600 (excluding program listing).
8. Approximate Core-Space Used--42,500 bytes.

B. Purpose:

This program simulates one hundred replications of fifty observation stages for the sample problem in trajectory estimation. Different filters can be used for the estimation by changing the subroutine FILTER described later.

C. Restrictions:

This program has been written for sample problem and as such can handle constant state transition and observation-to-state relation matrices. The MAIN program will need modification for use in case of changing magnitudes of these matrices.

D. Definitions:

1. Subroutines:

   a. MAIN--The MAIN program acts as a monitor in simulating the problem and calls various subroutines. It also performs the calculations necessary to obtain mean squared error matrices.

   b. FILTER--This subroutine estimates the state vector using observations available.

   c. SIMLTE--This subroutine generates the random numbers for the simulation.

   d. EMPEST--This subroutine estimates the state disturbance error vector for the smooth empirical Bayes filter.

   e. MTF--This subroutine performs triangular factorization on the given positive definite symmetric matrix using Kraut factorization.

   f. VARCOR--This subroutine transforms the vectors used by EMPEST by using covariance correction.

g. MTRXOP--This subroutine performs various matrix operations depending upon an input parameter.

h. GJR--This subroutine inverts the given matrix using the Gauss-Jordan-Rutishauser method with double pivoting.

i. OUTPUT--This subroutine prints the mean squared error matrices for the given filter.

j. RANDU--This subroutine generates random numbers in the range of 0.0 to 1.0.

k. GAUSS--This subroutine generates random numbers from a normal distribution.

m. PURGE2--This subroutine generates random numbers from a member of the Pearson family of distributions.

2. Subroutines Called By:

   a. MAIN--MTF, MTRXOP, GJR, PURGE2, SIMLTE, FILTER, OUTPUT.

   b. FILTER--MTRXOP, EMPEST, GJR.

   c. SIMLTE--GAUSS, MTRXOP.

   d. EMPEST--MTRXOP, GJR, VARCOR.

   e. GAUSS--RANDU.

3. Subscripted Variables in MAIN:

   a. COV1(6,6,50)--Stores the mean squared error matrices for all observation stages.

b. Q(6,6)--The state disturbance error covariance matrix.

c. PO(6,6)--The initial state covariance matrix.

d. R(6,6)--The observation error covariance matrix.

e. QLT(6,6)--Lower triangular matrix factor of Q.

f. PLT(6,6)--Lower triangular matrix factor of PO.

g. RLT(6,6)--Lower triangular matrix factor of R.

h. XHAT(6)--Estimate of the state vector X.

i. XBAR(6)--Estimate of the last state vector mapped by the state transition matrix.

j. X(6)--The state vector.

k. XNEW(6)--The state vector before the addition of the state disturbance error.

m. RS(100)--Stores the random numbers generated by PURGE2.

n. RDS(300)--Stores the random numbers for the elements of the state disturbance error for fifty observation stages.

o. U(6)--The state disturbance error vector.

p. V(6)--The observation error vector.

q. F(6,6)--The state transition matrix.

r. H(6,6)--The observation-to-state relation matrix.

s. AY(6)--The observation vector before the addition of the observation error.

t. Y(6)--The observation vector.

u. YSTORE(6,50)--Stores the observations for all observation stages.

v. RINV(6,6)--Inverse of the matrix R.

w. XO(6)--The initial state vector.

4. Important Non-Subscripted Variables in MAIN:

a. IIIX--The seed for PURGE2.

b. NREP--Number of replications.

c. ISEED--The seed for GAUSS.

d. IN--Index for number of replication.

E. Input and Output:

1. Input:

Cards 1 to 6--(6F12.8)--F, one row of the matrix on each card in sequence.

Cards 7 to 12--(6F12.8)--H, one row of the matrix on each card in sequence.

Card 13--(6F13.3)--XO, the initial state vector.

Card 14--(I4)--NREP, the number of replications.

Card 15--(I4)--ISEED, the seed for GAUSS.

Cards 16 to 21--(6F13.3)--Q, one row of the matrix on each card in sequence.

Cards 22 to 27--(6F13.3)--PO, one row of the matrix on each card in sequence.

Cards 28 to 33-(6F13.3)--R, one row of the matrix on each card in sequence.

Cards 34 and 35--Specification cards for PURGE2.

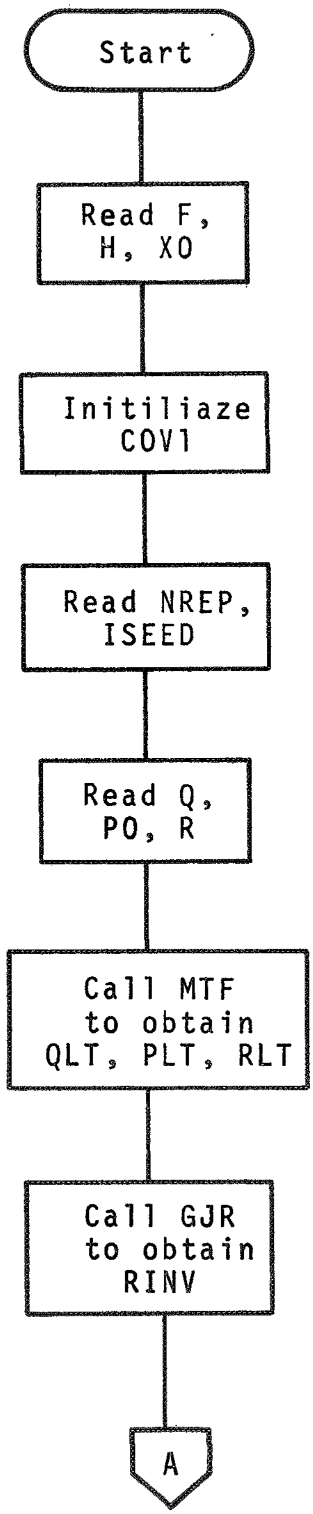
2. Output:

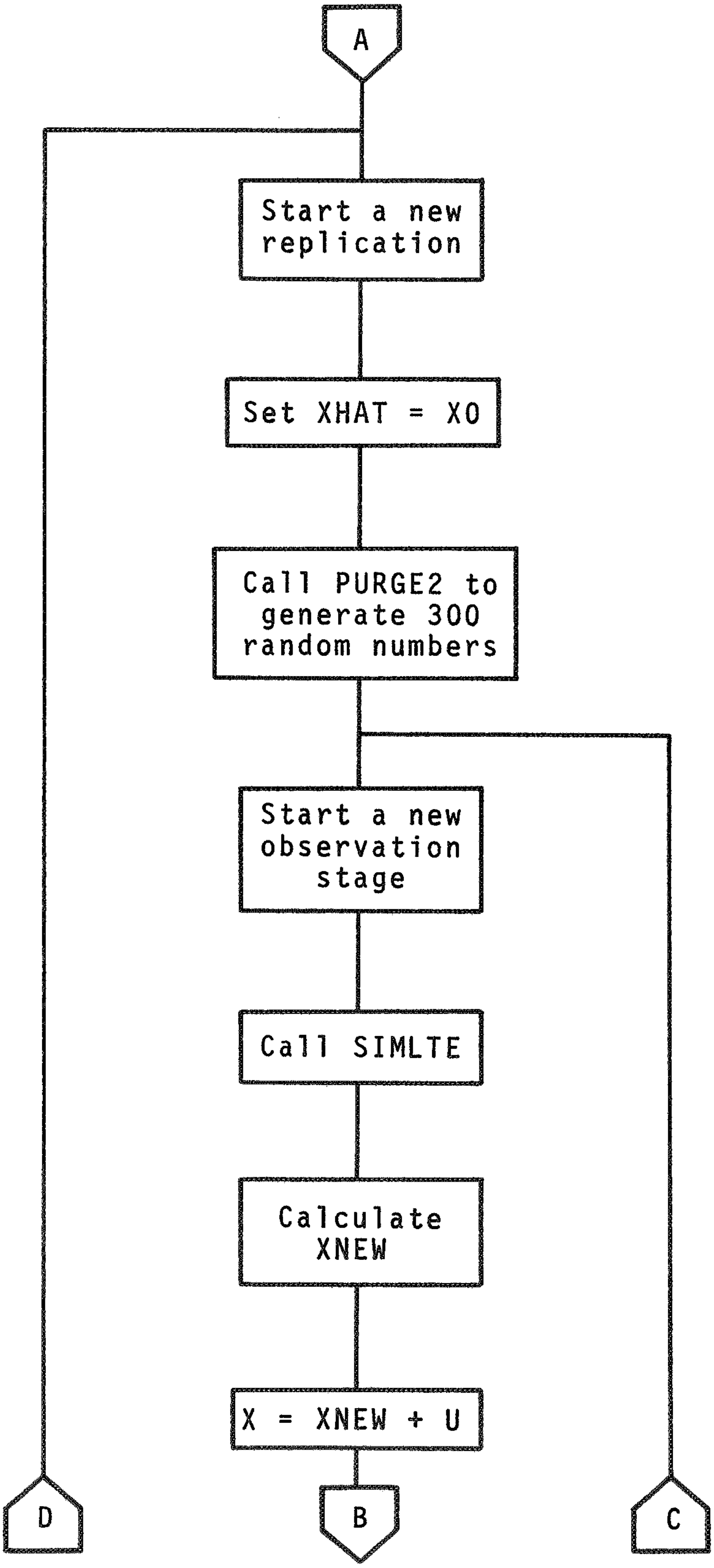
First Page--F, H, XO, NREP, ISEED, Q, PO, R, QLT, PLT, RLT, RINV.

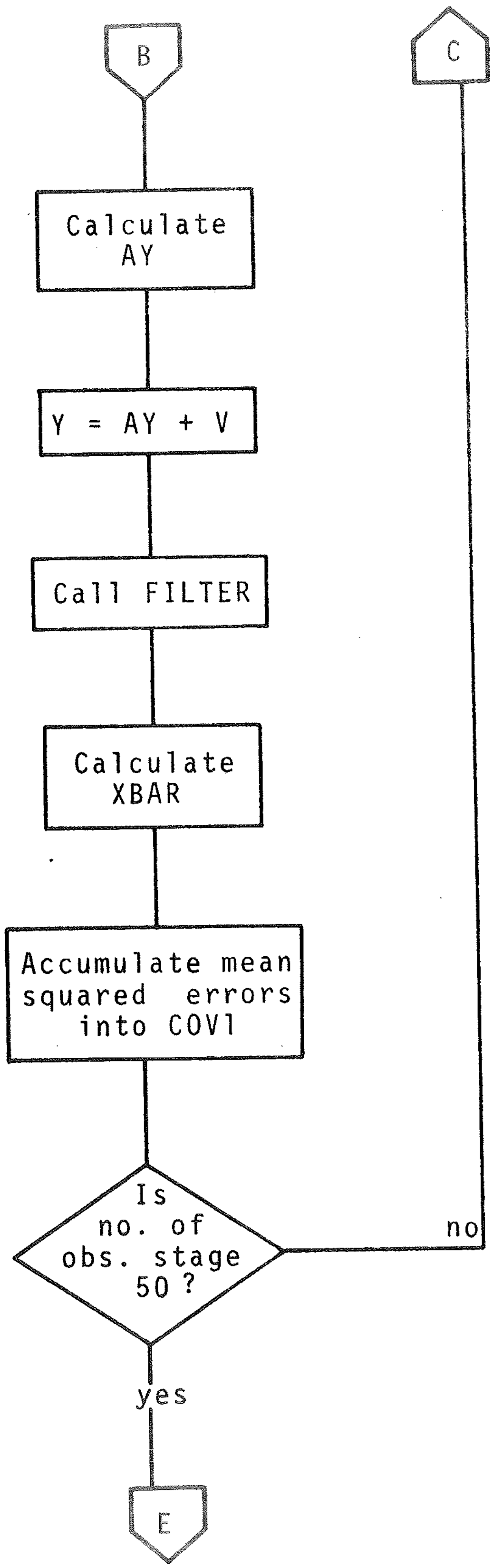
Second Page--Parameters in PURGE2.

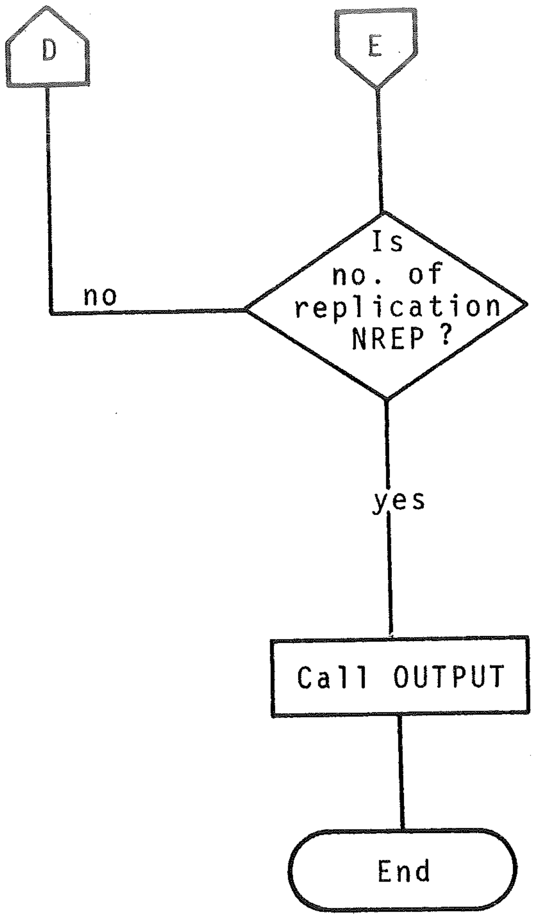
Subsequent Pages--The mean squared error matrices, described in detail in the output.

F. Flow Chart of MAIN:




A
Start a new replication
Set XHAT = XO
Call PURGE2 to generate 300 random numbers
Start a new observation stage
Call SIMLTE
Calculate XNEW
X = XNEW + U
D
B
C


B
C
Calculate
AY
Y = AY + V
Call FILTER
Calculate
XBAR
Accumulate mean
squared errors
into COV1
Is
no. of
obs. stage
50 ?
no
yes
E


D
E
Is no. of replication NREP ?
no
yes
Call OUTPUT
End

```
      COMMON/BLK1/RDS,PLT,QLT,RLT,X,U,V
      COMMON/BLK2/H,R,XBAR,XHAT
      COMMON/BLK3/K,ISEED
      COMMON/BLK4/RINV,IN
      COMMON/BLK6/XO
      COMMON/Z2/RS/Z11/IIIX
      DIMENSION COV1(6,6,50),Q(6,6),PO(6,6),R(6,6),QLT(6,6
     1),PLT(6,6),RLT(6,6),XHAT(6),XBAR(6),X(6),XNEW(6),RS(
     2100),RDS(300),U(6),V(6),F(6,6),H(6,6),AY(6),Y(6),YST
     3ORE(6,50),RINV(6,6),XO(6)
      IIIX = 519
      READ 5, ((F(I,J),J=1,6),I=1,6)
      READ 5, ((H(I,J),J=1,6),I=1,6)
      PRINT 5, F
      PRINT 5, H
    5 FORMAT (6F12.8)
      READ 6, XO
      PRINT 6, XO
    6 FORMAT (6F13.3)
    7 FORMAT (1X,6F14.3)
      DO 20 I=1,6
      DO 20 J=1,6
      DO 20 K=1,50
   20 COV1(I,J,K) = 0.
      READ 25, NREP,ISEED
      PRINT 25, NREP,ISEED
   25 FORMAT (I4)
      READ 6, ((Q(I,J),J=1,6),I=1,6)
      READ 6, ((PO(I,J),J=1,6),I=1,6)
      READ 6, ((R(I,J),J=1,6),I=1,6)
      PRINT 6, Q
      PRINT 6, PO
      PRINT 6, R
      CALL MTF (QLT,Q,6)
      CALL MTF (PLT,PO,6)
      CALL MTF (RLT,R,6)
      PRINT 6, QLT
      PRINT 6, PLT
      PRINT 6, RLT
      CALL MTRXOP (R,F,RINV,6,6,6,6,5)
      ZERO = 0.
      CALL GJR (RINV,6,ZERO,MSING,6,6)
      IF (MSING.EQ.2)   STOP
      PRINT 5, RINV
      DO 150 IN=1,NREP
      DO 30 I=1,6
   30 XHAT(I) = XO(I)
      CALL MTRXOP (F,XHAT,XBAR,6,6,6,1,1)
      DO 40 I=1,3
      IF (IN-1)  31,31,33
```

```
 31   IF(I-1)  32,32,33
 32   CALL PURGE2(1,5)
      GO TO 35
 33   CALL PURGE2(2,5)
 35   IJ = (I-1)*100
      DO 36 J=1,100
 36   RDS(IJ+J) = RS(J)
 40   CONTINUE
      PRINT 5, RDS
      DO 140 K=1,50
      CALL SIMLTE
      CALL MTRXOP (F,X,XNEW,6,6,6,1,1)
      CALL MTRXOP (XNEW,U,X,6,1,6,1,2)
      CALL MTRXOP (H,X,AY,6,6,6,1,1)
      DO 60 I=1,6
 60   YSTORE(I,K) = Y(I)
      CALL FILTER (Y,K)
      CALL MTRXOP (F,XHAT,XBAR,6,6,6,1,1)
      DO 70 I=1,6
      DO 70 J=1,6
 70   COV1(I,J,K) = COV1(I,J,K)+(XHAT(I)-X(I))*(XHAT(J)-X(
     1J))
140   CONTINUE
150   CONTINUE
      CALL OUTPUT (COV1,NREP)
      CALL EXIT
      END


      SUBROUTINE FILTER (Y,K)
      COMMON/BLK2/H,R,XBAR,XHAT
      COMMON/BLK5/Z,THETA
      DIMENSION H(6,6),HT(6,6),R(6,6),HHINV(6,6),XBAR(6),H
     1X(6),Z(6,50),HHINVH(6,6),THETA(6),UHAT(6),XHAT(6),Y(
     26)
      CALL MTRXOP (H,HT,R,6,6,6,6,4)
      CALL MTRXOP (HT,H,HHINV,6,6,6,6,1)
      CALL MTRXOP (H,XBAR,HX,6,6,6,1,1)
      DO 10 I=1,6
 10   Z(I,K) = Y(I)-HX(I)
      CALL EMPEST
      ZERO = 0.
      CALL GJR (HHINV,6,ZERO,MSING,6,6)
      IF (MSING.EQ.2)   STOP
      CALL MTRXOP (HHINV,HT,HHINVH,6,6,6,6,1)
      CALL MTRXOP (HHINVH,THETA,UHAT,6,6,6,1,1)
      CALL MTRXOP (UHAT,XBAR,XHAT,6,1,6,1,2)
      RETURN
      END
```

```
      SUBROUTINE SIMLTE
      COMMON/BLK1/RDS,PLT,QLT,RLT,X,U,V
      COMMON/BLK3/K,ISEED
      COMMON/BLK6/XO
      DIMENSION AX(6),PLT(6,6),X(6),AU(6),RDS(300),QLT(6,6
     1),U(6),RLT(6,6),AV(6),V(6),XO(6)
      IF (K-1)  10,10,30
   10 CONTINUE
      DO 20 I=1,6
      CALL GAUSS (ISEED,1.,0.,RN)
   20 AX(I) = RN
      CALL MTRXOP (PLT,AX,X,6,6,6,1,1)
      DO 25 I=1,-
   25 X(I) = X(I)+XO(I)
   30 CONTINUE
      KJ = (K-1)*6
      DO 40 J=1,6
   40 AU(J) = RDS(KJ+J)
      CALL MTRXOP (QLT,AU,U,6,6,6,1,1)
      DO 50 I=1,6
      CALL GAUSS (ISEED,1.,0.,RN)
   50 AV(I) = RN
      CALL MTRXOP (RLT,AV,V,6,6,6,1,1)
      RETURN
      END



      SUBROUTINE EMPEST
      COMMON/BLK3/K,ISEED
      COMMON/BLK4/RINV,IN
      COMMON/BLK5/TX,ENX
      DOUBLE PRECISION EB,Q2,DEN,QA
      DIMENSION RDS(300),T(6),RINV(6,6),P(6,6),XN(6),Q(6),
     1XJ(6),X(6,50),PQ(6),Q1(6),Q2(6),ENX(6),R(6,6),Q3(6),
     2TX(6,50)
      DO 5 I=1,6
    5 XN(I) = TX(I,K)
      AK = K
      A = AK**(-.04)
      KN = (K-1)*6
      DO 20 I=1,6
      DO 20 J=1,6
      P(I,J) = RINV(I,J)
      IF (I.EQ.J)    P(I,J) = P(I,J)+1./(A*A)
   20 CONTINUE
      CALL MTRXOP (RINV,XN,Q,6,6,6,1,1)
      ZERO = 0.
      CALL GJR (P,6,ZERO,MSING,6,6)
      IF (MSING.EQ.2)   STOP
      CALL VARCOR (TX,X,K)
```

```
25   CONTINUE
     DO 35 I=1,6
35   Q2(I) = 0.
     DEN = 0.
     DO 70 J=1,K
     DO 40 I=1,6
     XJ(I) = X(I,J)
40   Q1(I) = Q(I)+XJ(I)/(A*A)
     CALL MTRXOP (P.Q1,PQ,6,6,6,1,1)
     B = 0.
     DO 50 I=1,6
50   B = B+.5*Q1(I)*PQ(I)-.5*XJ(I)*XJ(I)/(A*A)
     EB = EXP(B)
     DO 60 I=1,6
60   Q2(I) = Q2(I)+Q1(I)*EB
     DEN = DEN+EB
70   CONTINUE
     DO 80 I=1,6
80   Q2(I) = Q2(I)/DEN
     DO 75 I=1,6
     QA = Q2(I)
75   Q3(I) = SNGL(QA)
     CALL MTRXOP (P,Q3,ENX,6,6,6,1,1)
     RETURN
     END


     SUBROUTINE MTF (A,B,N)
     DIMENSION A(N,N),B(N,N)
     DO 10 I=1,N
     DO 10 J=1,N
10   A(I,J) = 0.
     DO 70 J=1,N
     DO 70 I=J,N
     C = 0.
     L = J-1
     IF (L)  40,40,20
20   DO 30 K=1,L
30   C = C+A(I,K)*A(J,K)
40   CONTINUE
     IF (I-J)  50,50,60
50   A(I,J) = SQRT(B(I,J)-C)
     GO TO 70
60   A(I,J) = (B(I,J)-C)/A(J,J)
70   CONTINUE
     RETURN
     END
```

```
      SUBROUTINE VARCOR (T,X,K)
      COMMON/BLK1/R
      DIMENSION R(6,6),T(6,50),X(6,50),SUMT(6),SQT(6),COVT
     1(6)
      IF (K-1)  10,10,25
   10 DO 15 I=1,6
      SUMT(I) = T(I,1)
      SQT(I) = T(I,1)*T(I,1)
   15 CONTINUE
      DO 20 I=1,6
   20 X(I,1) = T(I,1)
      RETURN
   25 CONTINUE
      DO 30 I=1,6
      SUMT(I) = SUMT(I)+T(I,K)
   30 SQT(I) = SQT(I)+T(I,K)*T(I,K)
      DO 35 I-1,6
   35 COVT(I) = (SQT(I)*K-SUMT(I)*SUMT(I))/(K*(K-1))
      DO 45 I=1,6
      B = (COVT(I)-R(I,I))/COVT(I)
      IF (B.LT.O)   B = 1.
      A = SQRT(B)
      DO 40 X(I,J) = A*T(I,J)+(1.-A)*SUMT(I)/K
   45 CONTINUE
      RETURN
      END


      SUBROUTINE MTRXOP (A,B,C,K,L,M,N,KOP)
C
C     KOP = 1 .... C = AXB
C         = 2 .... C = A+B
C         = 3 .... C = A-B
C         = 4 .... B = A TRANSPOSE
C         = 5 .... C = A
C
      DIMENSION A(K,L),B(M,N),C(K,N)
      GO TO (10,30,50,70,90), KOP
   10 DO 20 I=1,K
      DO 20 J=1,N
      C(I,J) = 0.
      DO 20  IJ=1,M
   20 C(I,J) = C(I,J)+A(I,KJ)*B(IJ,J)
      GO TO 120
   30 E = 1.
   35 CONTINUE
      DO 40 I=1,K
      DO 40 J=1,L
```

```
 40   C(I,J) = A(I,J)+E*B(I,J)
      GO TO 120
 50   E = -1.
      GO TO 120
 70   DO 80 I=1,L
      DO 80 J=1,K
 80   B(I,J) = A(J,I)
      GO TO 120
 90   E = 0.
      GO TO 35
120   RETURN
      END


      SUBROUTINE GJR (A,NN.EPS,MSING,IS,IS1)
      INTEGER P,Q
      DIMENSION A(IS,IS1),B(125),C(125),P(125),Q(125)
      N = NN
      IF (EPS.LE.0.0)   EPS = 1.E-26
      MSING = 1
      DO 10 K=1,N
      PIVOT = 0.
      DO 20 I=K,N
      DO 20 J=K,N
      IF (ABS(A(I,J))-ABS(PIVOT))  20,20,30
 30   PIVOT = A(I,J)
      P(K) = I
      Q(K) = J
 20   CONTINUE
      IF (ABS(PIVOT)-EPS)  40,40,50
 50   IF (P(K)-K) 60,80,60
 60   DO 70 J=1,N
      L = P(K)
      Z = A(L,J)
      A(L,J) = A(K,J)
 70   A(K,J) = Z
 80   IF (Q(K)-K)  85,90,85
 85   DO 100 I=1,N
      L = Q(K)
      Z = A(I,L)
      A(I,L) = A(I,K)
100   A(I,K) = Z
 90   CONTINUE
      DO 110 J=1,N
      IF (J-K)  130,120,130
120   B(J) = 1./PIVOT
      C(J) = 1.
      GO TO 140
```

```
130   B(J) = -A(K,J)/PIVOT
      C(J) = A(J,K)
140   A(K,J) = 0.
110   A(J,K) = 0.
      DO 10 I=1,N
      DO 10 J=1,N
 10   A(I,J) = A(I,J)+C(I)*B(J)
      DO 155 M=1,N
      K = N-M+1
      IF (P(K)-K)  160,170,160
160   DO 180 I=1,N
      L = P(K)
      Z = A(I,L)
      A(I,L) = A(I,K)
180   A(I,K) = Z
170   IF (Q(K)-K)  190,155,190
190   DO 150 J=1,N
      L = Q(K)
      Z = A(L,J)
      A(L,J) = A(K,J)
150   A(K,J) = Z
155   CONTINUE
151   RETURN
 40   PRINT 45,P(K),Q(K),PIVOT
 45   FORMAT (1H0,15HSINGULAR MATRIX,3H I=,I3,3H J=,I3,7H P
     1IVOT=,E16.8,/)
      MSING = 2
      GO TO 151
      END


      SUBROUTINE OUTPUT (COV1,NREP)
      DIMENSION COV1(6,6,50)
  5   FORMAT (1H1,43H FOLLOWING ARE SQUARED ERROR MATRICES
     1AFTER,I4,13H REPLICATIONS)
 10   FORMAT (1H0,20H FOR OBSERVATION NO.,I3)
 12   FORMAT (1X,6F20.6)
 15   FORMAT (1H0,29H TRACE OF THE ABOVE MATRIX IS,F20.8)
      PRINT 5, NREP
      DO 20 K=1,50
      DO 20 I=1,6
      DO 20 J=1,6
 20   COV1(I,J,K) = COV1(I,J,K)/FLOAT(NREP)
      DO 30 K=1,50
      TR = 0.
      DO 25 I=1,6
 25   TR = TR+COV1(I,I,K)
      PRINT 10, K
      PRINT 12, ((COV1(I,J,K),J=1,6),I=1,6)
      PRINT 15, TR
```

```
30  CONTINUE
    RETURN
    END


    SUBROUTINE RANDU (IX,IY,YFL)
    IY = IX*65539
    IF (IY)  5,6,6
 5  IY = IY+2147483647+1
 6  YFL = IY
    YFL = YFL*.4656613E-9
    RETURN
    END


    SUBROUTINE GAUSS (IX,S,AM,Y)
    A = 0.0
    DO 50 I=1,12
    CALL RANDU (IX,IY,V)
    IX = IY
50  A = A+V
    Y = (A-6.0)*S+AM
    RETURN
    END
```