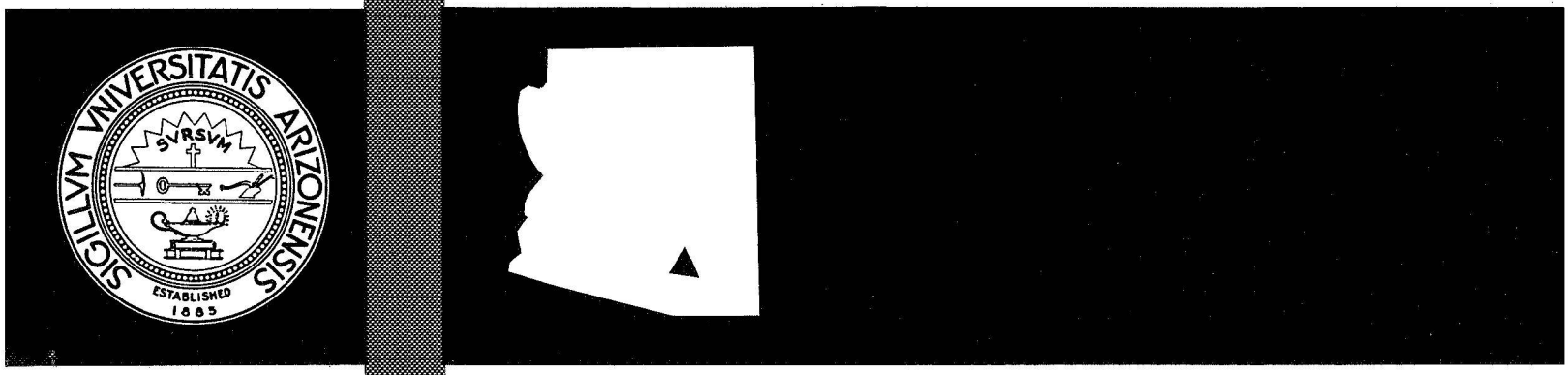
STOCHASTIC ALGORITHMS

FOR SELF-ADAPTIVE FILTERING AND PREDICTION

NASA 03-002-006

Semi-Annual Report - Part A

January 1971



ENGINEERING EXPERIMENT STATION
COLLEGE OF ENGINEERING
THE UNIVERSITY OF ARIZONA
TUCSON, ARIZONA

STOCHASTIC ALGORITHMS FOR SELF-ADAPTIVE FILTERING AND PREDICTION

by

Robert Lee Thomas Hampton

NOL -

NASA 03-002-006

Semi-Annual Report - Part A

January 1971

Donald G. Schultz
Principal Investigator

TABLE OF CONTENTS

Table of Contents (Continued)

Table of Contents (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

The contribution of this dissertation is the derivation of adaptive stochastic algorithms for learning directly the stationary discrete time optimal Kalman filter in an unknown random environment. To accomplish this derivation, an unsupervised learning criterion is formulated. It is shown that this criterion is a necessary and sufficient condition for optimal filtering. The criterion is then used in conjunction with the theory of Stochastic Approximation to develop the algorithms and to prove that they converge with probability one.

In addition, a method for accelerating the adaptation rate is given. Techniques for estimating other useful quantities such as the error covariance are also presented.

In the latter part of this study the stochastic algorithms are are modified to accommodate slowly varying statistics and special classes of nonlinear systems. Numerical simulations are performed to verify experimentally the theoretical expectations and predictions for both linear and nonlinear systems. The adaptive technique is then applied to learning the optimal filter for a thermionic point reactor using the "linearized prompt jump approximation" to the kinetics model of the reactor.

CHAPTER 1

INTRODUCTION

1.1 Prologue

Recently considerable attention has been directed toward self-adaptive (or self-learning) techniques for optimizing system performance. The basic idea is quite simple: one wishes to design a system to operate efficiently in an unknown or changing environment without the necessity of direct human intervention. Such systems are extremely important in the context of control and communication theory where it is often impractical or impossible to obtain the a priori information required to specify the optimum system.

The goal of this report is to provide a self-adaptive solution to the problem of optimal filtering, prediction, and smoothing of stochastic signals imbedded in random noise. However, before discussing the specific problem in more detail, a historical survey of this topic is appropriate.

1.2 Background and Historical Survey

One of the most important topics in control theory is the stochastic control problem. Here one is required to determine the optimum controller for a given plant without precise knowledge of the state $\underset{\sim}{x}(t)$ of the plant. The stochastic approach to optimum control is motivated by the fact that in general:

1) some of the state variables are not available for measurement,

2) the measurements contain noise,

3) the plant is subject to random input disturbances.

By using the familiar state variable representation, a discrete time linear dynamic system model of the plant can be described by

$$\underset{\sim}{x}(n+1) = \Phi(n+1,n)\ \underset{\sim}{x}(n) + D(n)\ \underset{\sim}{u}(n) + \underset{\sim}{w}(n) \qquad (1.1)$$

and the measurement of the state $\underset{\sim}{x}(n)$ by

$$\underset{\sim}{z}(n) = H(n)\ \underset{\sim}{x}(n) + \underset{\sim}{v}(n) \qquad (1.2)$$

where

$\underset{\sim}{x}(n)$ is the m x 1 system state vector

$\underset{\sim}{u}(n)$ is the ℓ x 1 control vector

$\underset{\sim}{w}(n)$ is the m x 1 white perturbation noise input vector

$\Phi(n+1,n)$ is the one-step m x m state transition matrix

$D(n)$ is the m x ℓ control matrix

$\underset{\sim}{z}(n)$ is the p x 1 measurement vector

$\underset{\sim}{v}(n)$ is the p x 1 white measurement noise vector

$H(n)$ is the p x m observation matrix.

The properties of the noise vectors $\underset{\sim}{v}(n)$ and $\underset{\sim}{w}(n)$ are defined by

$$E\{\underset{\sim}{w}(n)\ \underset{\sim}{w}^T(n)\} = Q(n)\ \delta(n-s)$$

$$E\{\underset{\sim}{v}(n)\ \underset{\sim}{v}^T(n)\} = R(n)\ \delta(n-s)$$

$$E\{\underset{\sim}{v}(n)\ \underset{\sim}{w}^T(s)\} = [0] \text{ for all } n,s$$

where

$Q(n)$ is the m x m perturbation noise covariance matrix

$R(n)$ is the p x p observation noise covariance matrix

and

$\delta(\cdot)$ is the Kronecker delta function.

The approach [Lee, 1964] generally used to attack this problem is to first estimate the state $\underset{\sim}{x}(n)$. Then this estimate $\hat{\underset{\sim}{x}}(n)$ is used as if it were the actual state to calculate the optimum control employing deterministic methods, such as the maximum principle. In other words, the stochastic control problem is separated into two phases, referred to as estimation and control. It has been proved that for linear systems with a quadratic performance index and subjected to white Gaussian perturbation and observation noise inputs, the optimal stochastic controller consists of an optimal estimator (filter) in cascade with an optimal deterministic controller [Joseph and Tou, 1961]. This result is known as the Separation Theorem. In this paper, only the estimation phase is considered because the deterministic control solution is well known [Schultz and Melsa, 1967 or Sage, 1968].

In communication theory an equally important topic is the stochastic detection problem. Here it is required to determine the optimal receiver for detecting the presence of a stochastic signal $\underset{\sim}{x}(t)$ imbedded in additive random Gaussian noise $\underset{\sim}{v}(t)$. If it is assumed that the signal $\underset{\sim}{x}(t)$ has a rational spectrum, it is possible to represent it as the state of a linear dynamic system with a white Gaussian noise input [Kalman, 1960]. The linear dynamic system is called the signal generating process,

and in discrete time state variable representation is described by

$$\underset{\sim}{x}(n+1) = \Phi(n)\ \underset{\sim}{x}(n) + \underset{\sim}{w}(n) \tag{1.3}$$

and the measurement of the signal x(n) by

$$\underset{\sim}{z}(n) = H(n)\ \underset{\sim}{x}(n) + \underset{\sim}{v}(n) \tag{1.4}$$

The notation is the same as that of Eqns. (1.1) and (1.2) except $\Phi(n+1,n)$ for r = 1 is simplified to $\Phi(n)$. The signal generating process (1.3) is identical to the control plant process (1.1), without the control input $\underset{\sim}{u}(n)$. However, the Separation Theorem states that the control term can be disregarded in the estimation phase. Therefore, the estimation problem and its solution are identical for both control and communication theory.

There also exists an analogous Separation Theorem solution to the stochastic detection problem [Kailath, 1963] which states that for a Gaussian signal with rational spectrum observed in white additive Gaussian noise, the optimal stochastic detector consists of an optimal estimator (filter) in cascade with the optimal detector for a completely known signal, i.e., the output of the filter is considered to be the actual signal. Again, only the estimation phase is considered, since the optimum detection solution is well known [Hancock and Wintz, 1966 or Van Trees, 1968].

Because of the identical mathematical framework of estimation in a control or communication context, no distinction between the two areas is made in the text that follows.

Wiener [1949] and Kolmogorov [1941] are credited with the solution for a single input-single output system. Wiener formulated the problem in terms of finding the optimum (in a minimum mean-square error sense) linear filter. He showed that a necessary and sufficient condition for optimality is that the filter satisfy the Wiener-Hopf equation, and he developed a method (spectral-factorization) for solving this equation for signals with a known stationary rational spectrum and for noise with a known stationary white spectrum.

Following Wiener's pioneering work, there developed an extensive literature which interpreted, simplified, modified, and extended his results. Detailed bibliographies may be found in Stumper [1955] and Balakrishnan [1963].

However, the case of a non-stationary multidimensional signal in non-stationary multidimensional noise remained unsolved in an engineering sense until 1960-1961 when Kalman [1960] and Kalman and Bucy [1961] published their fundamental papers. Instead of seeking a solution to the Wiener-Hopf equation in the frequency domain with the attendant problem of spectral factorization, Kalman combined the concept of state variable representation of dynamic systems with the orthogonal projection in a Hilbert space formulation of linear filtering to obtain a direct solution in the time domain. In contrast to Wiener's method, Kalman's results are in recursive form, and therefore ideally suited to real-time sequential digital computation. However, both the Wiener and Kalman theories require complete knowledge of the message generating

and observation noise covariance matrices, denoted by Q and R respectively. Kalman's method also requires the initial value of the error covariance matrix.

In the real world such extensive a priori information is generally not available. The consequence of not knowing R and/or Q is a suboptimal filter, i.e., an increase in the error covariance matrix. In some cases the increase is unbounded [Sorenson, 1966]. Detailed investigations of the suboptimal performance caused by insufficient a priori information have been widely reported in the literature, e.g., Soong [1965], Heffes [1966], and Nishimura [1966, 1967].

In addition, the inverse of R is often required to exist to avoid ill-conditioned matrices in the Kalman filter computation. The presence of either noiseless measurements or correlated observation noise can render R singular. In practice R itself is often ill-conditioned simply because one measurement is an order of magnitude more accurate than the others. Thus the Kalman filter formulation often generates application difficulties. Bryson and Johansen [1965] and Bryson and Mehra [1968] have modified the Kalman framework to handle aspects of this particular problem, but their technique necessitates state augmentation, which increases the dimension of the filter, increases the computation time, and requires taking derivatives.

1.3 Statement of the Problem and Previous Results

The inadequacy or absence of a priori knowledge leads naturally to the consideration of adaptive or learning approaches to optimum

estimation. Specifically, a self-adaptive solution to the discrete time, stationary optimum filtering and prediction problem is sought which does not require a priori specification of R and Q, and which retains the recursive features of Kalman's formulation.

Previous adaptive techniques can be divided into two types. The first, due to Magill [1965], assumes that the parameters of R and Q belong to a finite ensemble of a priori known possibilities. An optimum Bayesian pattern recognition algorithm for Gaussian distributions is used to learn which sampled data process is being observed. With this knowledge, Q and R are uniquely specified. Magill's method is valid only for a scalar observation process and is cumbersome to apply. For example, given N unknown elements of Q and the unknown scalar element R, with M possible values for each variance, there are $(N+1)^M$ combinations. Each combination requires the implementation of the corresponding Kalman filter equations. Hilborn and Lainiotis [1969] extended Magill's technique to a vector observation process and prove mean square and probability one convergence. In their case, there exist $(N+P)^M$ combinations where P is the number of unknown elements in the matrix R.

The second approach is to estimate directly the components of R and Q. Shellenbarger [1966,1967] devised a scheme using the likelihood principle to accomplish this estimation under the assumption of Gaussian distributions and other more restrictive requirements which limit its utility. One of these restrictions is fundamental. It implies that Q cannot be learned if it contains more than $m \times p$ elements. Proof of convergence was not considered. It is important to note that

these two approaches require the estimation of both the R and Q matrices. Then the entire set of Kalman's equations must be solved for the filter matrix each time the estimates of R and Q are updated.

1.4 Approach to the Problem

In this thesis an unsupervised learning criterion is formulated from which self-adaptive algorithms are derived. These algorithms learn the optimum discrete time stationary Kalman filter directly. This eliminates both the necessity of estimating R and Q as an intermediate step and the need to solve the entire set of filtering equations. The number of parameters to be determined and the computation time is thereby reduced. In addition, the problem associated with the occurrence of an ill-conditioned estimate of the R matrix is avoided. Satisfaction of the learning criterion is shown to be a necessary and sufficient condition for optimal filtering. The stochastic algorithms developed for estimating the optimum filter converge in mean-square and with probability one. The results are valid for scalar and vector valued signal and noise processes.

The learning criterion and the associated stochastic algorithms and their application constitute the principal contribution of this research paper.

1.5 Organization of the Dissertation

The second chapter presents and compares Wiener and Kalman filter theory. These results provide the foundation for the adaptive learning criterion.

Chapter 3 formulates the learning criterion and proves that it is necessary and sufficient for optimal filtering. A Markov sequence model of the learning criterion is given.

The unsupervised stochastic algorithms required for performing the adaptation indicated by the learning criterion are derived in Chapter 4. The theory of Stochastic Approximation is invoked to prove convergence of the algorithms to the optimum Kalman filter matrix. In addition, a method for optimizing the adaptative process is developed.

Chapter 5 contains additions and extensions to the previous results. Algorithms are derived for estimating Q, R, the error covariance Γ and the one-step prediction error covariance Σ. These quantities are required for optimum smoothing. The rather restrictive assumption, needed in Chapter 4, that $(H)^{-1}$ exist is removed. The algorithms of Chapter 4 are also modified to accommodate slowly time-varying signal and noise statistics and altered to include special classes of nonlinear dynamics.

The experimental results obtained by simulating the learning algorithms on the CDC 6400 digital computer are presented in Chapter 6. Many different systems are examined in detail. These investigations serve to confirm experimentally the conclusions of Chapters 4 and 5.

Chapter 7 summarizes the results of this study along with recommendations for further research.

The contribution of this report is the development of an unsupervised learning criterion which is a necessary and sufficient condition for optimal filtering, the derivation from this criterion of adaptive

stochastic algorithms for learning the optimum filter matrix directly, and the application of the algorithm to specific problems of communication and control theory.

CHAPTER 2

OPTIMAL FILTER THEORY

2.1 Introduction and Organization of the Chapter

The subject of this chapter is estimation theory, and its application to the engineering problem of extracting a stochastic signal from noise or estimating the state of a control system from noisy measurements. Section 2.2 defines the terms filtering, prediction, and smoothing which are used throughout the remainder of the chapter. Section 2.3 is devoted to some of the more important aspects of estimation theory. The application of this theory to the field of filtering and prediction leads to the Wiener and Kalman methodologies which are developed in Sections 2.4 and 2.5, respectively, and compared in Section 2.6. This comparison unifies the more general recursive structure of Kalman's solution to the time-varying state estimation problem and the non-recursive structure of Wiener's earlier solution to the stationary signal extraction problem.

The two theories are thus related as a prelude to developing a self-adaptive method of solving the stationary state estimation (or signal extraction) problem which retains the recursive features of Kalman's approach. Specifically, the Wiener-Hopf equation of Section 2.4 provides the theoretical foundation for the adaptive learning criterion, but its proof and application utilizes the recursive filter of Kalman presented

in Section 2.5. Prediction and smoothing are treated briefly in Sections 2.7 and 2.8.

2.2 Filtering, Prediction, and Smoothing: A Definition

At this point it is convenient to specify precisely what is meant by the terms filtering, prediction and smoothing of a stochastic signal $\underset{\sim}{x}(t)$ observed in additive noise $\underset{\sim}{v}(t)$.

Definition: Given the measurement time series[1] $\{\underset{\sim}{z}(kT) = H\underset{\sim}{x}(kT) + \underset{\sim}{v}(kT): k \in n-r, n-r+1, \ldots, n\}$ which is the sum of the random processes $\underset{\sim}{x}(t)$ and $\underset{\sim}{v}(t)$, observed over the discrete time interval n-r,n-r+1, ..., n where n, r are positive integers such that $n > r$,

(1) Filtering is the estimation of $\underset{\sim}{x}(\tau)$ at $\tau = nT$.

(2) Prediction is the estimation of $\underset{\sim}{x}(\tau)$ for $\tau > nT$.

(3) Smoothing is the estimation of $\underset{\sim}{x}(\tau)$ for $\tau < nT$.

All three cases are considered in the succeeding pages, but the emphasis is on filtering because it is the key operation.

Note that even though $\underset{\sim}{x}(t)$ and $\underset{\sim}{v}(t)$ may be continuous functions of time, the data $\underset{\sim}{z}(nT)$ is observed only at discrete times. That is, in this work only sampled data is considered.

2.3 Optimal Estimation: Bayesian Approach

To discuss optimality, a criterion of optimality must be defined. Suppose that a random variable x is to be estimated from the data set $Z = \{z(1), \ldots, z(n)\}$. Then $\hat{x}$ is called the optimal estimate of x given

[1] The capital T in the argument is generally suppressed.

Z if and only if the average loss

$$E\{\ell(x-\hat{x})\} = E_z\{E_x[\ell(x-\hat{x})|Z]\} = E_z\{L(\hat{x}|Z)\} \qquad (2.1)$$

is a minimum, where $\ell(x-\hat{x})$ is an appropriately defined loss function. In Eq. (2.1) the expectation with respect to Z is not dependent upon $\hat{x}$; therefore it suffices to choose $\hat{x}$ such that

$$L(\hat{x}|Z) = E_x\{\ell(x-\hat{x})|Z\} \qquad (2.2)$$

is minimized. A solution based on minimizing (2.1) or, equivalently, Eq. (2.2) is called a Bayes estimator. It has been shown [Sherman, 1955, 1958] that for a rather general class of loss function $\ell(\cdot)$ and a posterior densities that the Bayesian estimate is the conditional mean

$$\hat{x} = E\{x|Z\} \qquad (2.3)$$

THEOREM 2-1. Let $S = \{\ell(\cdot)$: ℓ is even and convex$\}$. If the a posteriori density $p(x|Z)$ is even about its conditional mean $E\{x|Z\}$, then the conditional mean $E\{x|Z\}$ is the optimum estimate of x given Z in the sense that it minimizes Eq. (2.2) for all $\ell \in S$.

Proof:

$\ell(e) = \ell(-e)$ evenness (a)

$\ell(ae_1 + be_2) \leq a\ell(e_1) + b\ell(e_2)$ e_1, e_2 convexity (b)

where $a + b = 1$, $a \in (0,1)$

and $p(y|Z) = p(-y|Z)$ evenness (c)

where $y = x - E\{x|Z\}$

$$
\begin{aligned}
L(\hat{x}|Z) &= E_x\{\ell(x-\hat{x})|Z\} \\
&= E_x\{\ell(\hat{x}-x)|Z\} && \text{by (a)} \\
&= E_y\{\ell(\hat{x} - E[x|Z] - y)|Z\} \\
&= E_y\{\ell(\hat{x} - E[x|Z] + y)|Z\} && \text{by (c)} \\
&= E_y\{\ell(E[x|Z] - \hat{x} - y)|Z\} && \text{by (a)} \\
&= E_y\{\ell(E[x|Z] - \hat{x} + y)|Z\} && \text{by (c)} \\
&= E_y\{\ell(y - \{E[x|Z] - \hat{x}\})|z\} \\
&= \frac{1}{2}E_y\{\ell(y + \{E[x|Z] - \hat{x}\}) \\
&\quad + \ell(y - \{E[x|Z] - \hat{x}\})|Z\} && \text{by (a)} \\
&\geq E_y\{\ell(\tfrac{1}{2}[y + \{E[x|Z] - \hat{x}\}] \\
&\quad + \tfrac{1}{2}[y - \{E[x|Z] - \hat{x}\}])|Z\} && \text{by (b)} \\
&= E_y\{\ell(y)|Z\} \text{ with equality iff } \hat{x} = E\{x|Z\}
\end{aligned}
$$

Q.E.D.

The class S can be greatly extended if two restrictions are added to the conditional density function.

THEOREM 2-2. Let $S = \{\ell(\cdot)$: ℓ is even and $\ell(e_2) \geq \ell(e_1) \geq 0$ for $e_2 \geq e_1 \geq 0$, $\ell(0) = 0\}$. If the a posteriori density $p(x|Z)$ is

(1) even and monotone non-increasing about its conditional mean and

(2) decreases rapidly enough so that $\lim_{y \to \infty} [\ell(y)p(y|Z)] = 0$, where $y = x - E\{x|Z\}$

then $E\{x|Z\}$ is the optimum Bayes estimate.

Proof: [Viterbi, 1966, p. 308].

Some examples of the $\ell(\cdot) \in S$ are

$$\ell(e) = \begin{cases} C, & |e| > c \\ 0, & |e| \leq c \end{cases}$$

$$\ell(e) = C\,|e|$$

$$\ell(e) = C\,[1 - \exp(-e^2)] .$$

Note that under the conditions of Theorem 2 the conditional mean $E\{x|Z\}$ coincides with the maximum a posteriori estimate.

In general this study deals with vector-valued random processes $\underset{\sim}{x}(n)$. Equation 2.2 then becomes

$$L(\hat{\underset{\sim}{x}}(n)|Z(n)) = E_{\underset{\sim}{x}}\{\ell[\,\|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|\,]|Z(n)\} \qquad (2.4)$$

where $Z(n) = \{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\}$ and $\|\cdot\|$ denotes the norm of the vector. Theorems 2-1 and 2-2 extend readily to include this case [Kalman, 1960].

If the error squared is chosen as the loss function, then restrictions (1) and (2) of Theorem 2-2 on the a posteriori density are unnecessary.

THEOREM 2-3. Let $\ell[\,\|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|\,] = \|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|^2$

$$= [\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)],$$

then $\hat{\underset{\sim}{x}}(n) = E\{\underset{\sim}{x}(n)|Z(n)\}$ minimizes (2.4) without the restraints of Theorem 2-2 on $p(\underset{\sim}{x}(n)|Z(n))$.

Proof:

$$E_{\underset{\sim}{x}}\{[\underset{\sim}{x} - \hat{\underset{\sim}{x}}]^T[\underset{\sim}{x} - \hat{\underset{\sim}{x}}]|Z\} = \hat{\underset{\sim}{x}}^T\hat{\underset{\sim}{x}} - 2\hat{\underset{\sim}{x}}^T E\{\underset{\sim}{x}|Z\} + E\{\underset{\sim}{x}^T\underset{\sim}{x}|Z\}$$

$$= (\hat{\underset{\sim}{x}} - E\{\underset{\sim}{x}|Z\})^T (\hat{\underset{\sim}{x}} - E\{\underset{\sim}{x}|Z\}) + E\{\underset{\sim}{x}^T\underset{\sim}{x}|Z\} - [E\{\underset{\sim}{x}|Z\}]^T E\{\underset{\sim}{x}|Z\}$$

$$\geq E\{\underset{\sim}{x}^T\underset{\sim}{x}|Z\} - [E\{\underset{\sim}{x}|Z\}]^T E\{\underset{\sim}{x}|Z\}$$

with equality iff $\hat{\underset{\sim}{x}}(n) = E\{\underset{\sim}{x}(n)|Z(n)\}$ Q.E.D.

The contents of theorems 2-1, 2-2, 2-3 give the "in principle" solution of the Bayesian estimation problem for a wide class of loss functions and probability structures. However, the explicit computation of the optimum filtering estimate $E\{\underset{\sim}{x}(n)|Z(n)\}$ is formidable except in the important case when $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ are Gaussian. Here the well-known result that $E\{\underset{\sim}{x}(n)|Z(n)\}$ is a linear function of the data $Z(n)$ may be exploited.

THEOREM 2-4: Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be Gaussian random vector sequences,

then
$$E\{\underset{\sim}{x}(n)|Z(n)\} = \sum_{j=i}^{n} A(n,j)z(j) = \hat{\underset{\sim}{x}}(n) \qquad (2.5)$$

where the $A(n,j)$ are m x p matrices and represent the response of the optimum filter matrix at time n to an impulse applied at time j.

Proof: [Deutsch, 1965, p. 32].

As mentioned briefly in Chapter 1, the classical method of specifying the optimal linear transformations $A(n,j)$ is through the Wiener-Hopf equation. This important theory is the subject of the next section and provides the theoretical basis for the adaptive learning

criterion developed in Chapter 3. Kalman's more recent contribution, which utilizes the orthogonal projection theorem to determine the optimal A(n,j), is presented in Section 2.5. His results provide the recursive solution to the optimum filter problem used in proving the learning criterion.

2.4 Wiener Filter Theory

The treatment here of optimal filter theory is in the spirit of Wiener's original work on single input-single output continuous time systems, but the formulation is for multiple input-multiple output sampled data systems.

THEOREM 2-5: The Wiener-Hopf Equation

Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be zero mean random sequences.

If either

(i) the random sequences are Gaussian or

(ii) the estimator $\hat{\underset{\sim}{x}}(n)$ is restricted to be a linear function of the data

$Z(n) = \{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\}$ and

$$\ell(\|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|) = \|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|^2$$

$$= [\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]$$

a necessary and sufficient condition for

$$\hat{\underset{\sim}{x}}(n) = \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j) \qquad (2.5)$$

to be the optimum estimate of $\underset{\sim}{x}(n)$ is that the filtering transformations $A(n,j)$ be chosen such that $\hat{\underset{\sim}{x}}(n)$ satisfies the Wiener-Hopf equation

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(\nu)\} = [0] \quad \text{for all } \underset{\sim}{z}(\nu) \in Z(n) \tag{2.6}$$

Proof: See Theorem 2-8 where it is shown that the orthogonal projection theorem implies the Wiener-Hopf equation and conversely.

The discrete multidimensional Wiener-Hopf equation may be put in a more familiar form by substituting Eq. (2.5) into Eq. (2.6) yielding

$$\begin{aligned} E\{[\underset{\sim}{x}(n) - \sum_{j=i}^{n} A(n,j)\ \underset{\sim}{z}(j)]\ \underset{\sim}{z}^T(\nu)\} \\ = E\{\underset{\sim}{x}(n)\underset{\sim}{z}^T(\nu)\} - \sum_{j=i}^{n} A(n,j)E\{\underset{\sim}{z}(j)\ \underset{\sim}{z}^T(\nu)\} \\ = R_{xz}(n,\nu) - \sum_{j=i}^{n} A(n,j)\ R_{zz}(j,\nu) \qquad (2.7) \\ = [0] \end{aligned}$$

where

$$R_{xz}(n,\nu) = E\{\underset{\sim}{x}(n)\ \underset{\sim}{z}^T(\nu)\}$$

and

$$R_{zz}(j,\nu) = E\{\underset{\sim}{z}(j)\ \underset{\sim}{z}^T(\nu)\}$$

If $i = -\infty$, Eq. (2.7) is assumed to satisfy the conditions of Theorem B.1 in the Appendix so that interchanging the order of expectation and summation is valid.

Wiener's approach[1] to explicitly determining the optimum filter consists of transforming Eq. (2.6) into the frequency domain[2] and using spectral-factorization. The solution is given in terms of the z-transform A(z) of the m x p filter matrix A(n-j) as illustrated in Fig. 2.1. The problem of synthesizing the filter remains. Employment of the frequency domain requires the following restrictions on the system defined by Eqns. (1.3) and (1.4)

(1) The state transition matrix Φ and observation matrix H must be time invariant.

(2) The statistics of $\underset{\sim}{w}(n)$ and $\underset{\sim}{y}(n)$ must be stationary.

(3) The data z(n) must be known for all past time, i.e., $i = -\infty$ in Eq. (2.7).

Under these conditions, Eqns. (2.5) and (2.7) become

$$\hat{\underset{\sim}{x}}(n) = \sum_{j=-\infty}^{n} A(n-j)\ \underset{\sim}{z}(j) \tag{2.8}$$

and

$$E\{\underset{\sim}{x}(n)\ \underset{\sim}{z}^T(\nu)\} - \sum_{j=-\infty}^{n} A(n-j)\ E\{\underset{\sim}{z}(j)\ \underset{\sim}{z}^T(\nu)\}$$

$$= R_{xz}(n-\nu) - \sum_{j=-\infty}^{n} A(n-j)\ R_{zz}(j-\nu)$$

$$= [0], \quad \nu \in (-\infty, \ldots, n) \tag{2.9}$$

---

1. The first general technique for determining the optimum multiple input-multiple output discrete filter using Wiener's method was given by Motyka and Cadzow [1967]. The results presented in this section follow their work.

2. Since only sampled-data is considered, frequency domain means the Z-transform domain.

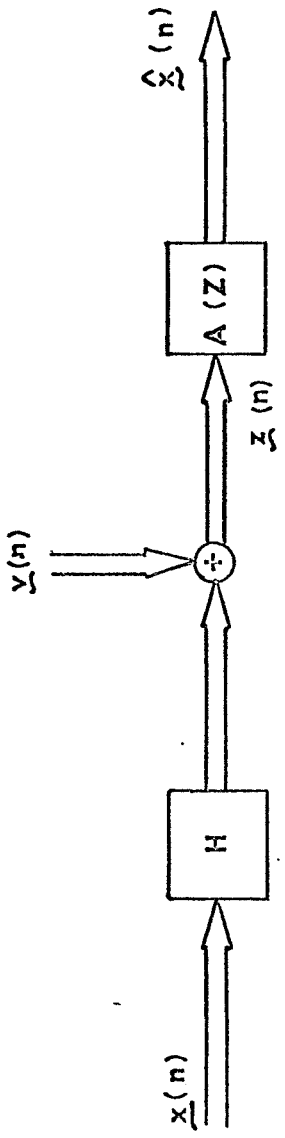


Fig. 2.1 Wiener Filter Model

By a change of variable (2.9) can be rewritten

$$R_{xz}(\alpha) - \sum_{\tau=0}^{\infty} A(\tau)\, R_{zz}(\alpha-\tau) = [0],\ \alpha\ \varepsilon(0, \ldots, \infty) \qquad (2.10)$$

which is just the discrete time stationary Wiener-Hopf equation.

The cross-spectral (generating function) matrix representation of (2.10) is

$$\phi_{xz}(Z) - A(Z)\ \phi_{zz}(Z) = [0] \qquad (2.11)$$

Since each element of (2.10) is zero for each α, each element of (2.11) is a polynomial in Z only. Thus each polynomial element converges for all Z inside the unit circule. Assuming $\phi_{zz}(Z)$ has a spectral factorization of the form

$$\phi_{zz}(Z) = \Delta(Z^{-1})\ \Delta^{T}(Z)$$

where $\Delta(Z)$ is a p x p matrix whose elements represent the Z-transforms of stable, linear, causal systems (i.e., polynomials in Z) and have no poles outside the unit circle, then a physically realizable Wiener filter exists. The frequency domain expression for this optimum m x p filter matrix is

$$A(Z) = [\{\Delta^{-1}(Z^{-1})\ \phi_{xz}(Z)\}_{c} + \{\Delta^{-1}(Z^{-1})\ \phi_{xz}(Z)\}_{+}]^{T}[\Delta(Z)]^{-1} \qquad (2.12)$$

where $\{\ \}_{c}$ is a matrix whose elements are constant and $\{\ \}_{+}$ is a matrix whose elements contain only poles inside the unit circle. The "in principle" solution given by (2.12) is cumbersome, not easy to synthesize, and not suited to machine computation.

Kalman's time domain approach not only eliminates these two difficulties, but also the three restrictions listed previously. He used the orthogonal projection theorem to obtain a recursive set of equations for optimum filtering and prediction.

2.5 Kalman Filter Theory

By applying the orthogonal projection theorem a different specification for the optimum linear transformations A(n,j) is obtained.

THEOREM 2-6: Orthogonal Projection

Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be zero mean random sequences. Let Y(n) be the linear manifold generated by the data:

$$Y(n) = \{\underset{\sim}{y}(n): \underset{\sim}{y}(n) = \sum_{j=i}^{n} B(n,j)\underset{\sim}{z}(j)\}$$

where the B(n,j) are arbitrary m x p matrices whose elements are real. Note that

$$\hat{\underset{\sim}{x}}(n) = \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j) \qquad (2.5)$$

must belong to Y(n).

If either

(i) the random sequences $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ are Gaussian or

(ii) the estimator $\hat{\underset{\sim}{x}}(n)$ is specified to be a linear function of the data $\{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\}$ and the loss function is the error squared

$$\ell(\underset{\sim}{e}(n)) = \underset{\sim}{e}^{T}(n)\underset{\sim}{e}(n) = \| \underset{\sim}{e}(n) \|^{2}$$

where the error $\underset{\sim}{e}(n) = \underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)$, then the filtering estimate $\hat{\underset{\sim}{x}}(n)$ of $\underset{\sim}{x}(n)$ is optimum if and only if the $A(n,j)$ are chosen such that the error

$$\underset{\sim}{e}(n) = \underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n) = \underset{\sim}{x}(n) - \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j)$$

is orthogonal to $Y(n)$, i.e.,

$$(\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n), \underset{\sim}{y}(n)) = 0 \quad \text{for all } \underset{\sim}{y}(n) \in Y(n) \tag{2.13}$$

where $(\cdot,\cdot)$ is the inner product induced on $Y(n)$ by $E\{(\cdot)^T(\cdot)\}$.

Proof: Note that from Theorem 2-4 the optimum estimate is the conditional mean and it takes the form of Eq. (2.5) for a large class of loss functions, including the error squared criterion. Thus, it suffices to prove only part (ii).

Let $\Theta_n = [A(n,i), \ldots, A(n,n)]$ and $\underset{\sim}{Z}^T(n) = [\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)]$, where Θ_n is a $m \times (n-i)p$ matrix and $\underset{\sim}{Z}(n)$ is a $(n-i)p \times 1$ vector.

Substituting Θ_n and $\underset{\sim}{Z}(n)$ into (2.5) gives

$$\hat{\underset{\sim}{x}}(n) = \Theta_n \underset{\sim}{Z}(n) \tag{2.14}$$

The linear transformations $A(n,j)$ embodied in Θ_n minimize $E\{\underset{\sim}{e}^T(n)\underset{\sim}{e}(n)\}$ iff for all $m \times (n-i)p$ matrices Δ_n,

$$E\{[\underset{\sim}{x}(n) - (\Theta_n + \Delta_n)\underset{\sim}{Z}(n)]^T[\underset{\sim}{x}(n) - (\Theta_n + \Delta_n)\underset{\sim}{Z}(n)]$$

$$- [\underset{\sim}{x}(n) - \Theta_n\underset{\sim}{Z}(n)]^T[\underset{\sim}{x}(n) - \Theta_n\underset{\sim}{Z}_n]\} \geq 0 \tag{2.15}$$

Expanding the left side of Ineq. (2.15) leads to

$$E\{[\underset{\sim}{x}(n) - \Theta_{n\sim}\underset{\sim}{Z}(n)]^T \Delta_{n\sim}\underset{\sim}{Z}(n) + [\Delta_{n\sim}\underset{\sim}{Z}(n)]^T[\underset{\sim}{x}(n) - \Theta_{n\sim}\underset{\sim}{Z}(n)] + [\Delta_{n\sim}\underset{\sim}{Z}(n)]^T\Delta_{n\sim}\underset{\sim}{Z}(n)\}$$

$$= 2E\{[\underset{\sim}{x}(n) - \Theta_{n\sim}\underset{\sim}{Z}(n)]^T \Delta_{n\sim}\underset{\sim}{Z}(n)\} + E\{[\Delta_{n\sim}\underset{\sim}{Z}(n)]^T \Delta_{n\sim}\underset{\sim}{Z}(n)\} \geq 0 \tag{2.16}$$

Now $E\{[\Delta_{n\sim}\underset{\sim}{Z}(n)]^T\Delta_{n\sim}\underset{\sim}{Z}(n)\} \geq 0$, and since it can be made arbitrarily small with respect to the term linear in Δ_n, the sign of the left side of Ineq. (2.16) is determined by this linear term. It, therefore, follows that

$$E\{[\underset{\sim}{x}(n) - \Theta_{n\sim}\underset{\sim}{Z}(n)]^T \Delta_{n\sim}\underset{\sim}{Z}(n)\} = 0 \text{ for all } \Delta_n \tag{2.17}$$

If Eq. (2.17) was not true, the sign of Eq. (2.16) would depend on Δ_n which would imply a contradiction because Δ_n is arbitrary.

Since $\underset{\sim}{y}(n) = \Delta_{n\sim}\underset{\sim}{Z}(n)$, Eq. (2.17) may be written

$$E\{[\underset{\sim}{x}(n) - \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j)]^T\underset{\sim}{y}(n)\} = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T\underset{\sim}{y}(n)\}$$

$$= (\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n), \underset{\sim}{y}(n))$$

$$= 0 \text{ for all } \underset{\sim}{y}(n) \in Y(n) \tag{2.13}$$

The converse follows easily. If Eq. (2.13) is true, then Eq. (2.16) reduces to

$$E\{[\Delta_{n\sim}\underset{\sim}{Z}(n)]^T\Delta_{n\sim}\underset{\sim}{Z}(n)\} \geq 0 \quad \text{for all } \Delta_n$$

Hence satisfaction of Eq. (2.13) minimizes the $E\{\underset{\sim}{e}_n^T\underset{\sim}{e}_n\}$.

Q.E.D.

Under conditions (i) of the theorem, the orthogonal projection of $\underset{\sim}{x}(n)$ on Y(n) is identical to the conditional mean $E\{\underset{\sim}{x}(n)|Z(n)\}$. Thus this theorem implies that the optimal linear estimator cannot be improved upon unless the random phenomenon are non-Gaussian and, even then, only if knowledge of at least third order joint probability distribution functions is available.[1] Kalman [1960] employed Theorem 2-6 to solve the optimum filtering problem, but his approach and proof differ from that given here.

COROLLARY 2-7:

$$(\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n), \hat{\underset{\sim}{x}}(n)) = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \hat{\underset{\sim}{x}}(n)\} = 0 \tag{2.18}$$

where

$$\hat{\underset{\sim}{x}}(n) = \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j) \tag{2.5}$$

By combining the state variable representation of dynamic systems with the orthogonal projection theorem, Kalman [1960] and Kalman and Bucy [1961] obtained the following recursive solution to the non-stationary multidimensional state estimation problem. Given the dynamic system of Eq. (1.3) and the measurement system of Eq. (1.4), then

$$\hat{\underset{\sim}{x}}(n) = \Phi(n)\hat{\underset{\sim}{x}}(n-1) + K(n)[\underset{\sim}{z}(n) - H(n)\Phi(n)\hat{\underset{\sim}{x}}(n-1)] \tag{2.19}$$

$$= E\{\underset{\sim}{x}(n)|Z(n)\} \quad \text{for Gaussian noise}$$

with the initial condition $\hat{\underset{\sim}{x}}(0) = \underset{\sim}{0}$, and

---

1. Given any random sequence, there exists a unique Gaussian sequence with the same mean and covariance kernel.

$$K(n) = \Gamma(n)H^T(n)R^{-1}(n)$$

$$= \Sigma(n|n-1)\ H^T(n)[H(n)\Sigma(n|n-1)\ H^T(n) + R(n)]^{-1} \qquad (2.20)$$

$$= \text{Kalman filter matrix}$$

$$\Gamma(n) = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)][\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T\}$$

$$= \Sigma(n|n-1) - \Sigma(n|n-1)H^T(n)[H(n)\Sigma(n|n-1)H^T(n) + R(n)]^{-1}H(n)\Sigma(n|n-1)$$

$$= [I - K(n)H(n)]\Sigma(n|n-1) \qquad (2.21)$$

$$= \text{Error covariance matrix}$$

with the initial condition $\Gamma(0) = \Gamma_o$

$$\Sigma(n+1|n) = E\{[\underset{\sim}{x}(n+1) - \Phi(n)\hat{\underset{\sim}{x}}(n)][\underset{\sim}{x}(n+1) - \Phi(n)\hat{\underset{\sim}{x}}(n)]^T\}$$

$$= \Phi(n)\Gamma(n)\Phi(n)^T + Q(n+1) \qquad (2.22)$$

$$= \text{One-step prediction error covariance matrix}$$

The block diagram for the Kalman filter is shown in Fig. 2.2. Note that it is in the form of a non-autonomous feedback system. The recursive nature of the Kalman filter equations makes it ideally suited for digital computation and sequential processing of new data in real time. These attributes are just as outstanding in the stationary system, where Φ, H, R, Q and therefore, K, Γ, ϕ, Σ, are time invariant. In this case, once the optimum Kalman filter matrix K_{opt} is determined, Eq. (2.19) is the only real time calculation required. When the covariance matrices Q of the plant perturbation noise and R of the observation noise are known a priori, K_{opt} can be determined off-line. In this report Q and R are not assumed to be known. The absence of this information necessitates

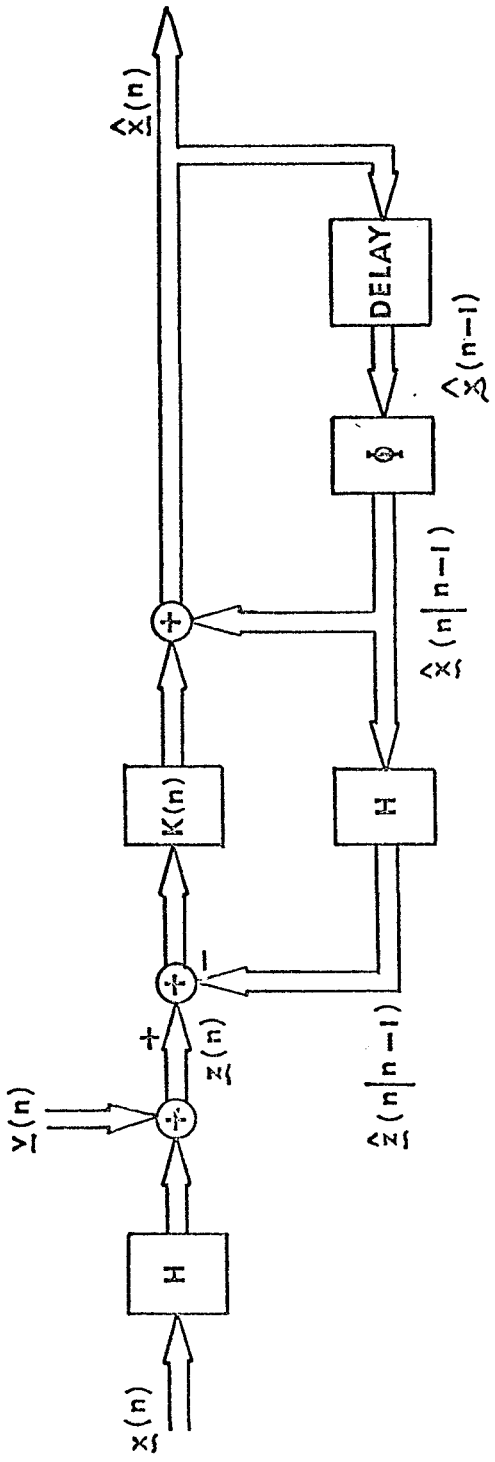


Fig. 2.2 Kalman Filter Model

two on-line recursions: one for estimating K_{opt} and one for estimating $\underset{\sim}{x}(n)$ with K_{opt} replaced by its estimate.

2.6 Theoretical Equivalence of Wiener and Kalman Filter Theory

In order to provide a common framework for comparing the approaches of Wiener and Kalman, it is necessary to show that the orthogonality of the error $\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)$ and the linear manifold $Y(n)$ implies the Wiener-Hopf equation and conversely. Note that Eq. (2.6) may be viewed as an outer product induced by $E\{(\cdot)(\cdot)^T\}$. With this observation the generalized form of the multidimensional Wiener-Hopf equation is given by the outer product

$$[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)])(\underset{\sim}{z}(\nu) = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\underset{\sim}{z}^T(\nu)\}$$
$$= [0] \quad \text{for all } \underset{\sim}{z}(\nu) \in Z(n)$$

The purpose of writing the Wiener-Hopf equation as an outer product is to place it in contradistinction to the principle of orthogonality as given by the inner product Eq. (2.13). However, they are equivalent as proved in the next theorem.

THEOREM 2-8: The Wiener-Hopf Equation

$$[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)])(\underset{\sim}{z}(\nu) = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\underset{\sim}{z}^T(\nu)\}$$
$$= [0] \text{ for all } \underset{\sim}{z}(\nu) \in Z(n) \qquad (2.6)$$

is satisfied if and only if the orthogonal projection theorem

$$([\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)], \underset{\sim}{y}(n)) = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \underset{\sim}{y}(n)\}$$
$$= 0 \quad \text{for all } \underset{\sim}{y}(n) \in Y(n) \qquad (2.13)$$

holds, where

$$\underset{\sim}{y}(n) = \sum_{k=i}^{n} B(n,k)\underset{\sim}{z}(k) \tag{2.16}$$

the B(n,k) are arbitrary m x p matrices whose elements are real, and

$$\underset{\sim}{\hat{x}}(n) = \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j) \tag{2.5}$$

Proof: Assume Eq. (2.13) holds. By substituting Eqns. (2.16) and (2.5) into Eq. (2.13) becomes

$$E\{[\underset{\sim}{x}(n) - \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j)]^T \sum_{k=i}^{n} B(n,k)\underset{\sim}{z}(k)\} = 0 \text{ for all } B(n,k) \tag{2.13}$$

To prove that Eq. (2.13) implies the Wiener-Hopf equation, it is first expanded and rewritten as a sum of its scalar components. The inner product of Eq. (2.13) is equivalent to the following summations

$$= E\{\sum_{a=1}^{m} [\underset{\sim}{x}(n) - \sum_{j} A(n,j)\underset{\sim}{z}(j)]_a [\sum_{k} B(n,k)\underset{\sim}{z}(k)]_a\}$$

$$= E\{\sum_{a=1}^{m}\sum_{b=1}^{p} \{[\underset{\sim}{x}(n) - \sum_{j} A(n,j)\underset{\sim}{z}(j)]_a \sum_{k} B_{ab}(n,k)z_b(k)\}\}$$

$$= E\{\sum_{a,b}\sum_{k} [\underset{\sim}{x}(n) - \sum_{j} A(n,j)\underset{\sim}{z}(j)]_a B_{ab}(n,k)z_b(k)\}$$

$$= E\{\sum_{k}\sum_{a,b} B_{ab}(n,k)[\underset{\sim}{x}(n) - \sum_{j} A(n,j)\underset{\sim}{z}(j)]_a z_b(k)\}$$

Now by interchanging the linear operations of summation and expectation, the above equation becomes

$$\sum_k \sum_{a,b} B_{ab}(n,k)\, E\Big\{ \sum_{c=1}^{m} [\underset{\sim}{x}_a(n) - \sum_j A_{ac}(n,j)\underset{\sim}{z}_c(j)[\underset{\sim}{z}_b(k)\Big\}$$

$$= \sum_k \sum_{a,b} B_{ab}(n,k) \sum_c [E\{\underset{\sim}{x}_a(n)\underset{\sim}{z}_b(k)\} - \sum_j A_{ac}(n,j)E\{\underset{\sim}{z}_c(j)\underset{\sim}{z}_b(k)\}]$$

Performing the sum over c gives

$$\sum_k \sum_{a,b} B_{ab}(n,k)[R_{xz}(n,k) - \sum_j A(n,j)R_{zz}(j,k)]_{ab} = 0 \qquad (2.23)$$

Since (2.23) is zero for arbitrary B(n,k), the term in brackets must be identically zero for all a, b, and k. Otherwise, a contradiction exists. Thus, it has been shown that each element of Eq. (2.6) is zero.

$$\therefore \quad R_{xz}(n,k) - \sum_{j=i}^{n} A(n,j)R_{zz}(j,k)$$

$$= E\{[\underset{\sim}{x}(n) - \sum_{j=i}^{n} A(n,j)\underset{\sim}{z}(j)]\underset{\sim}{z}^T(k)\}$$

$$= E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\underset{\sim}{z}^T(k)\}$$

$$= [0] \text{ for all } \underset{\sim}{z}(k) \in Z(n)$$

By reversing the argument, the converse follows readily. Q.E.D.

Corollary 2-9:

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\hat{\underset{\sim}{x}}^T(n)\} = [0]$$

$$\text{iff} \quad E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \hat{\underset{\sim}{x}}(n)\} = 0$$

The previous theorem and the accompanying corollary provide a common foundation for the filter theory of Wiener and Kalman.

Even though the theoretical equivalence of Wiener and Kalman filter theory has been demonstrated, the explicit computational relations for the optimum filters of Wiener and Kalman represented by Eqns. (2.12) and (2.20) respectively bear no external resemblance. The solution for the Wiener filter $A(n-\nu)$ is given in the frequency domain as the Z-transform of the impulse response of the optimum filter matrix. The Kalman filter matrix $K(n)$ is given in the time domain as a function of the error covariance matrix, the observation matrix, and the measurement noise covariance matrix. In the filtering Eq. (2.19), $K(n)$ acts simply as a weighting for the correction term. However, when the restrictions required for the Wiener approach are satisfied, the two filters must be identical. For this case Φ, H, R, and Q are time invariant. Therefore, the optimum Kalman filter, denoted by K_{opt}, is also constant. With these observations the relationship of $A(n-\nu)$ to K_{opt} is derived. Since the estimate $\hat{\underset{\sim}{x}}(n)$ is the same regardless of the filter used, it is seen that

$$\hat{\underset{\sim}{x}}(n) = \Phi\hat{\underset{\sim}{x}}(n-1) + K_{opt}[\underset{\sim}{z}(n) - H\,\Phi\,\hat{\underset{\sim}{x}}(n-1)] = \sum_{\alpha=-\infty}^{n} A(n-\alpha)\underset{\sim}{z}(\alpha)$$

From Theorem 2-5

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\underset{\sim}{z}^T(n)\} = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)][\underset{\sim}{x}^T(n)\,H^T + \underset{\sim}{v}^T(n)]\} = [0]$$

or

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{x}^T(n)\}\ H^T = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{v}^T(n)\} \qquad (2.24)$$

The left hand side of (2.24) is equal to $\Gamma(n)H^T$ from Corollary 2-9, and the right hand side is $E\{\hat{\underset{\sim}{x}}(n)\ \underset{\sim}{v}^T(n)\}$ since $\underset{\sim}{x}(n)$ is independent of $\underset{\sim}{v}(n)$. Therefore, (2.24) becomes

$$\Gamma(n)\ H^T = E\{\hat{\underset{\sim}{x}}(n)\ \underset{\sim}{v}^T(n)\}$$

$$= E\{\sum_{\alpha=-\infty}^{n} A(n-\alpha)\ \underset{\sim}{z}(\alpha)\ \underset{\sim}{v}^T(n)\}$$

$$= E\{\sum_{\alpha=-\infty}^{n} A(n-\alpha)[H\underset{\sim}{x}(\alpha) + \underset{\sim}{v}(\alpha)]\ \underset{\sim}{v}^T(n)\}$$

$$= \sum_{\alpha=-\infty}^{n} A(n-\alpha)E\{\underset{\sim}{v}(\alpha)\ \underset{\sim}{v}^T(n)\}$$

$$= A(0)R$$

which implies

$$A(0) = \Gamma(n)\ H^T\ R^{-1}$$

The optimum Kalman filter is by definition (2.15) equal to $\Gamma(n)\ H^T\ R^{-1}$. Therefore,

$$A(0) = K_{opt}$$

Thus the impulse response of the optimum Wiener Filter matrix evaluated at time equal to zero gives the optimum Kalman filter matrix. This means that if a method can be devised to learn A(0) when Q and R are unknown,

then the adaptive filter solution does not require estimation of the entire impulse response of the optimum filter matrix in the Wiener formulation or estimation of the elements of Q and R, and the subsequent necessity of calculating Σ in the Kalman formulation. Such a method represents a potentially great reduction in computation time, memory space, and algorithmic complexity. An iterative scheme for realizing this potential is developed in the next two chapters. First, however, optimal prediction and smoothing are introduced.

2.7 Prediction Theory

The previous sections contain a complete theory for optimal filtering as required in this report. The necessary theorems and their proofs for optimal prediction are directly analogous to those for optimal filtering.

THEOREM 2-10: Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be Gaussian random vector sequences, then

$$E\{\underset{\sim}{x}(n+r)\,|\,Z(n)\} = \sum_{j=i}^{n} A(r,j)\underset{\sim}{z}(j) \qquad (2.25)$$

$$= \hat{\underset{\sim}{x}}(n+r|n) \ , \ r > 0$$

where the $A(r,j)$ are m x p matrices. $\hat{\underset{\sim}{x}}(n+r|n)$ is the estimate of $x(n)$ given the data up to time r. The optimal linear transformations $A(r,j)$ can be determined from the orthogonal projection theorem.

THEOREM 2-11: Let $\{x(n)\}$ and $\{z(n)\}$ be zero mean random sequences. Let $Y(r)$ be the linear manifold generated by the data,

$$Y(r) = \{\underset{\sim}{y}(n) : \underset{\sim}{y}(n) = \sum_{j=i}^{n} B(r,j)\underset{\sim}{z}(j)\}$$

where $\underset{\sim}{y}(n)$ is any m x 1 vector belonging to $Y(n)$ and the $B(n,j)$ are m x p real matrices. If either

(i) the random sequences $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ are Gaussian or

(ii) the predictor $\hat{\underset{\sim}{x}}(n+r|n)$ is specified to be a linear function of the data $Z(n) = \{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\}$ and

$$\ell\{[\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)]\} = ||\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)||^2$$

Then the prediction $\hat{\underset{\sim}{x}}(n+r|n)$ is optimum if and only if the $A(r,j)$ are chosen such that the error $\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)$ is orthogonal to $Y(n)$,

$$E\{[\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)]^T \underset{\sim}{y}(n)\} = 0 \text{ for all } \underset{\sim}{y}(n) \in Y(r) \qquad (2.26)$$

The equivalence of the orthogonal projection theorem and the Wiener-Hopf equation for prediction is given by Theorem 2-12.

THEOREM 2-12: A necessary and sufficient condition for

$$E\{[\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)]\ \underset{\sim}{z}^T(k)\} = [0] \text{ for all } \underset{\sim}{z}(k) \in Z(n)$$

is that

$$E\{[\underset{\sim}{x}(n+r) - \hat{\underset{\sim}{x}}(n+r|n)]^T\ \underset{\sim}{y}(n)\} = 0 \text{ for all } \underset{\sim}{y}(n) \in Y(n)$$

Kalman's solution for predicting $\underset{\sim}{x}(n+r)$ is directly dependent on the optimum filtering equations.

$$\hat{\underset{\sim}{x}}(n+r|n) = \Phi(n+r,n)\ \hat{\underset{\sim}{x}}(n) \tag{2.27}$$

$$= \text{the prediction of } \underset{\sim}{x}(n+r) \text{ given the data}$$

$$\{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\} = Z(n)$$

$$= E\{\underset{\sim}{x}(n+r)|Z(n)\} \quad \text{for Gaussian noise}$$

where $\hat{\underset{\sim}{x}}(n)$ is the optimum filtering estimate of $\underset{\sim}{x}(n)$.

In the stationary case Eq. (2.27) reduces to

$$\hat{\underset{\sim}{x}}(n+r|n) = \Phi^r\ \hat{\underset{\sim}{x}}(n) \tag{2.28}$$

Regardless, once the optimum filter is determined, so is the optimum predictor.

2.8 Smoothing Theory

Recall from Section 2.1 that smoothing is the estimate $\hat{\underset{\sim}{x}}(m|n)$ of $\underset{\sim}{x}(m)$ at time n, when $m < n$. The optimum smoothing estimate for the class of loss functions of Section 2.2 is given by Theorem 2-13.

THEOREM 2-13: Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be Gaussian random vector sequences, then

$$E\{\underset{\sim}{x}(m)|Z(n)\} = \sum_{j=i}^{n} A(n,j)\,z(j) = \hat{\underset{\sim}{x}}(m|n) \tag{2.29}$$

Proof: [Meditch 1967, p. 162]

In practice it is convenient to dichotomize smoothing into three classes

(1) Fixed-Interval Smoothing: $\hat{\underset{\sim}{x}}(m|N)$ where $m \in [i, \ldots, N-1]$ and $n = N$, a fixed integer.

(2) Fixed-Point Smoothing: $\hat{\underset{\sim}{x}}(M|n)$ where m = M, a fixed integer and $n \in [M+1, M+2, \ldots]$

(3) Fixed-Lag Smoothing: $\hat{\underset{\sim}{x}}(m|m+N)$ where $m \in [0,1,2,\ldots]$ and N is a fixed integer.

The results presented here for these three classifications essentially follow Meditch [1969]. The interested reader is referred to his work for details and proofs.

THEOREM 2-14: The stationary optimal fixed-interval smoothed estimate $\hat{\underset{\sim}{x}}(m|N)$ of $\underset{\sim}{x}(m)$ is determined from

$$\hat{\underset{\sim}{x}}(m|N) = \hat{\underset{\sim}{x}}(m) + F_{opt}\,[\hat{\underset{\sim}{x}}(m+1|N) - \Phi\hat{\underset{\sim}{x}}(m)] \tag{2.30}$$

and the smoothing gain matrix by

$$\begin{aligned} F_{opt} &= \Gamma\,\Phi^{T}\,\Sigma^{-1} \\ &= [I - K_{opt}\,H]\,\Sigma\,\Phi^{T}\,\Sigma^{-1} \end{aligned} \tag{2.31}$$

As in prediction the fixed-interval smoothing estimate is directly dependent on the optimum filter K_{opt}, but the one-step prediction error covariance Σ must also be known. Alternately, knowledge of the error covariance matrix Γ can be substituted for K_{opt}.

THEOREM 2-15: The stationary optimum fixed-point smoothing estimate $\hat{\underset{\sim}{x}}(M|n)$ or $\underset{\sim}{x}(M)$ is determined from

$$\hat{\underset{\sim}{x}}(M|n) = \hat{\underset{\sim}{x}}(M|n-1) + S(n)H^{T}R^{-1}[\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)] \tag{2.32}$$

and the smoothing gain matrix by

$$S(n) = S(n-1)\ \Phi^T[I - H^T R^{-1} H\Gamma]$$

$$= S(n-1)\ \Phi^T[I - H^T R^{-1} H(I - K_{opt} H)\ \Sigma] \qquad (2.33)$$

with the initial condition

$$S(M) = \Gamma = [I - K_{opt} H]\Sigma$$

Again the fixed-point optimal smoothing matrix is a function of K_{opt}. However, the observation noise covariance matrix R is also required to compute S(n).

THEOREM 2-16: The stationary optimum fixed-lag smoothing estimate $\hat{\underset{\sim}{x}}(m|m+N)$ of $\hat{\underset{\sim}{x}}(m)$ is determined from

$$\hat{\underset{\sim}{x}}(m|m+N) = \Phi\hat{\underset{\sim}{x}}(m-1|m-1+N) + U_{opt}\ [\hat{\underset{\sim}{x}}(m-1|m-1+n) - \hat{\underset{\sim}{x}}(m-1)]$$

$$+ J(m+N)\ K_{opt}\ [\underset{\sim}{z}(m+N) - H\Phi\ \hat{\underset{\sim}{x}}(m-1+N)] \qquad (2.34)$$

and the smoothing gain matrices by

$$U_{opt} = Q[\Phi^{-1}]^T\ \Gamma^{-1}$$

$$= Q[\Phi^{-1}]^T\ [\Sigma - K_{opt}\ H\Sigma]^{-1} \qquad (2.35)$$

$$J(m+N) = F_{opt}^{-1}\ J(m-1+N)F_{opt}$$

$$= \Sigma[\Sigma\Phi^T - K_{opt}\ H\Sigma\Phi^T]^{-1}\ J(m-1+N)\ [I - K_{opt}\ H]\ \Sigma\Phi^T\Sigma^{-1} \qquad (2.36)$$

with the initial condition

$$J(N) = [F_{opt}]^N$$

This is the most complex smoothing algorithm from both a computational and informational standpoint. It requires knowledge of K_{opt}, Σ, and the plant perturbation noise covariance matrix Q. Numerous matrix inversions must also be calculated.

Sequential methods of learning the Q, R, and Σ are derived in Chapter 6. This permits an adaptive solution to the problem of optimal smoothing without a priori knowledge of Q and R. Since Σ is also being estimated directly, the necessity of solving the discrete matrix Riccati equation each time the estimates of Q and R are updated is eliminated.

2.9 Summary

This chapter provides the theoretical foundation for the rest of this study. It has reviewed the fundamental concepts of estimation theory and their application. It was shown that for Gaussian random sequences, the optimum estimator is linear. For a given system this important fact yields the filter theory of Wiener and Kalman, which are proved to be theoretically equivalent. However, the assumptions required to obtain an explicit solution using the Kalman method are much less restrictive. The Kalman filter equations are also in recursive form which provides a great computational advantage, even in the stationary case. The theory developed here is used extensively in the next chapter to derive the learning criterion for adaptive learning of the optimal Kalman filter.

CHAPTER 3

THE LEARNING CRITERION

3.1 Introduction and Organization of the Chapter

In Chapter 2 some of the important concepts of estimation theory were reviewed, and the results of Wiener and Kalman filter theory were presented and compared. There it was shown that for optimum filtering the estimator must satisfy the Wiener-Hopf equation. This equation is also the fundament of the learning criterion developed in Section 3.2. The learning criterion (Theorem 3-3) provides the theoretical foundation for learning the optimum Kalman filter matrix. Section 3.3 reduces the learning criterion to a Markov sequence model. This Markov model is used in Chapter 4 to motivate the adaptive technique.

3.2 The Unsupervised Learning Criterion

The purpose of the unsupervised learning criterion is to provide a necessary and sufficient condition for an adaptive solution to the stationary optimum filtering and prediction problem, where the signal and noise covariance matrices Q and R are unknown. In addition, this criterion must have two other characteristics.

1. It must be a function of measurable and/or calculable quantities only.
2. It must provide information from which convergent algorithms can be derived.

Otherwise, the criterion is meaningless from an engineering viewpoint.

As observed in Chapter 2 optimal filtering is a necessary condition for optimal smoothing. As a consequence, the unsupervised learning criterion is a necessary condition for adaptively solving the optimal smoothing problem.

The learning criterion developed in this chapter is said to be unsupervised because it operates on-line using only the noisy information generated by the actual filtering process. Specifically, it utilizes the residual time series $\{\underset{\sim}{\delta}_n\}$ which is the difference between the measurement process $\{\underset{\sim}{z}(n)\}$ and the predicted measurement process $\{H\Phi\underset{\sim}{\hat{x}}(n-1)\}$. By contrast a supervised learning criterion operates off-line using noise free externally supplied information. For example, in the context of this paper the learning process would be classified as supervised if, in addition to the noisy measurements $\underset{\sim}{z}(n)$, knowledge of the actual states $\underset{\sim}{x}(n)$ were available over a required training period.

In the search for an appropriate learning criterion the following elementary lemma is useful. It is included for completeness.

LEMMA 3-1:

Let the vector $\underset{\sim}{\gamma}$ be orthogonal to the data set $\{\underset{\sim}{z}(1), \ldots, \underset{\sim}{z}(n)\}$. Then $\underset{\sim}{\gamma}$ is orthogonal to any linear combination of the data.

Since the only information available is the residual time series $\{\underset{\sim}{\delta}_n\}$, it is natural to determine what properties it has when the filter is optimum and suboptimum. The independence and zero mean characteristics of $\underset{\sim}{v}(n)$ and $\underset{\sim}{w}(n)$ are exploited freely in the following theorem proofs without comment.

THEOREM 3-2:

Given the dynamic system

$$\underset{\sim}{x}(n+1) = \Phi\underset{\sim}{x}(n) + \underset{\sim}{w}(n) \tag{1.3}$$

the observation process

$$\underset{\sim}{z}(n) = H\underset{\sim}{x}(n) + \underset{\sim}{v}(n) \tag{1.4}$$

where the inverse of H exists, and the filtering equations

$$\hat{\underset{\sim}{x}}(i) = \underset{\sim}{0}, \text{where i is the initial time}$$

$$\hat{\underset{\sim}{x}}(n) = \Phi\hat{\underset{\sim}{x}}(n-1) + K[\underset{\sim}{z}(n) - H\Phi\ \hat{\underset{\sim}{x}}(n-1)] \tag{2.19}$$

If $K = K_{opt}$

$$= \Sigma H^T[H\Sigma H^T + R]^{-1}$$

i.e., K_{opt} is the optimum filter matrix chosen such that Theorem 2-6 is satisfied, then

$$E\{[\underset{\sim}{\Delta}_{j+1} - H\Phi K_{opt}\underset{\sim}{\delta}_j]^T\underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n \tag{3.1}$$

where

$$\underset{\sim}{\delta}_j = \underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)$$

$$\underset{\sim}{\Delta}_{j+1} = \underset{\sim}{z}(j+1) - H\Phi\Phi\hat{\underset{\sim}{x}}(j-1)$$

and conversely.

Proof: First examine the bracketed term in the above expectation

$$\underset{\sim}{\Delta}_{j+1} - H\Phi K_{opt}\underset{\sim}{\delta}_j = \underset{\sim}{z}(j+1) - H\Phi\Phi\hat{\underset{\sim}{x}}(j-1) - H\Phi K_{opt}\underset{\sim}{\delta}_j$$

$$= \underset{\sim}{z}(j+1) - H\Phi\Phi\hat{\underset{\sim}{x}}(j-1) - H\Phi K_{opt}[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]$$

$$= \underset{\sim}{z}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)$$

$$= \underset{\sim}{\delta}_{j+1}$$

Therefore, Eq. (3.1) may be rewritten as

$$E\{\underset{\sim}{\delta}^T_{j+1}\underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n \tag{3.2}$$

Assume $K = K_{opt}$, which from Theorem (2-6) implies that

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T B(n,j)\underset{\sim}{z}(j)\} = 0 \qquad \text{for all } n,j$$

where $j \leq n$, and $B(n,j)$ is an arbitrary m x p matrix whose elements are real. Now expanding Eq. (3.2) results in

$$E\{[\underset{\sim}{z}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\}$$

$$= E\{[H\Phi\underset{\sim}{x}(j+1) + \underset{\sim}{v}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\}$$

$$= E\{[\underset{\sim}{x}(j+1) - \Phi\hat{\underset{\sim}{x}}(j)]^T H^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)] +$$

$$E\{\underset{\sim}{v}^T(j+1)[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\}$$

$$= E\{[\Phi\underset{\sim}{x}(j) + \underset{\sim}{w}(j) - \Phi\hat{\underset{\sim}{x}}(j)]^T H^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\} +$$

$$E\{\underset{\sim}{v}^T(j+1)\}\ E\{\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)\}$$

$$= E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]^T \Phi^T H^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\} +$$

$$E\{\underset{\sim}{w}^T(j) H^T[\underset{\sim}{z}(j) - H\Phi\underset{\sim}{x}(j-1)]\} + 0$$

$$= E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]^T \Phi^T H^T \underset{\sim}{z}(j)\} -$$

$$E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]^T \Phi^T H^T H\Phi\hat{\underset{\sim}{x}}(j-1)\} + 0$$

The first term in the last equation is zero from the projection theorem. The second term is zero since $\hat{\underset{\sim}{x}}(j-1)$ is a linear combination of the $\underset{\sim}{z}(k)$, $k \leq j-1$ and, therefore, $\Phi^T H^T H\Phi\hat{\underset{\sim}{x}}(j-1) \in Y(n)$.

Consequently,

$$E\{\underset{\sim}{\delta}_{j+1}^T \underset{\sim}{\delta}_j\} = 0 \text{ for all } j \leq n.$$

Converse: Assume $E\{\underset{\sim}{\delta}_{j+1}^T \underset{\sim}{\delta}_j\} = 0$ for all $j \leq n$.

As a consequence of this assumption, the following preliminary result is obtained

$$E\{\underset{\sim}{\delta}_{j+1}^T \underset{\sim}{\delta}_j\} = E\{[\underset{\sim}{z}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]^T [\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\}$$

$$= E\{[H\underset{\sim}{x}(j+1) + \underset{\sim}{v}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]^T[\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]\}$$

$$= E\{[\underset{\sim}{x}(j+1) - \Phi\hat{\underset{\sim}{x}}(j)]^T H^T \underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n \qquad (3.3)$$

By hypothesis it is known that the lag-one residuals are orthogonal. It must now be shown that

$$E\{\underset{\sim}{\delta}_{n+1}^T \underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n$$

This is accomplished by iterating backward in time, starting with $j = n-1$. The state, observation and filtering equations are substituted freely in the following text.

$$E\{\underset{\sim}{\delta}^T_{n+1}\underset{\sim}{\delta}_{n-1}\} = E\{[\underset{\sim}{z}(n+1) - H\Phi\hat{\underset{\sim}{x}}(n)]^T[\underset{\sim}{z}(n-1) - H\Phi\hat{\underset{\sim}{x}}(n-1)]\}$$

$$= E\{[H\,\underset{\sim}{x}(n+1) + \underset{\sim}{v}(n+1) - H\Phi\hat{\underset{\sim}{x}}(n)]^T\,\underset{\sim}{\delta}_{n-1}\}$$

$$= E\{[\underset{\sim}{x}(n+1) = \Phi\hat{\underset{\sim}{x}}(n)]^T H^T\,\underset{\sim}{\delta}_{n-1}\}$$

$$= E\{[\Phi\,\hat{\underset{\sim}{x}}(n) + \underset{\sim}{w}(n) - \Phi\Phi\,\hat{\underset{\sim}{x}}(n-1) - K[\underset{\sim}{z}(n) - H\Phi\,\hat{\underset{\sim}{x}}(n-1)]]^T H^T\underset{\sim}{\delta}_{n-1}\}$$

$$= E\{[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)]^T\,\Phi^T\,H^T\,\underset{\sim}{\delta}_{n-1}\} -$$

$$E\{[\underset{\sim}{z}(n) - H\Phi\,\hat{\underset{\sim}{x}}(n-1)]^T\,K^T\,\Phi^T\,H^T\,\underset{\sim}{\delta}_{n-1}\}$$

The first term in the above equation is zero from Lemma 3-1 and Eq. (3.3). The second term is zero from Lemma 3-1 and Eq. (3.2). Thus

$$E\{\underset{\sim}{\delta}^T_{n+1}\,\underset{\sim}{\delta}_{n-1}\} = 0$$

Continuing the iteration, it is concluded that

$$E\{\underset{\sim}{\delta}^T_{n+1}\underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n \tag{3.4}$$

A more illustrated form of Eq. (3.4) is

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T \underset{\sim}{\delta}_j\} = 0 \qquad \text{for all } j \leq n \qquad (3.5)$$

In particular for j = i+1, Eq. (3.4) becomes

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T[\underset{\sim}{z}(i+1) - H\Phi\hat{\underset{\sim}{x}}(i)]\} = 0$$

Since $\hat{\underset{\sim}{x}}(i) = \underset{\sim}{0}$,

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T \underset{\sim}{z}(i+1)\} = 0$$

or

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T B(n,i+1)\ \underset{\sim}{z}(i+1)\} = 0$$

Proceeding by induction, assume

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T B(n,k)\underset{\sim}{z}(k)\} = 0 \qquad \text{for all } k \leq n-1$$

Now for k = n, Eq. (3.5) gives

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T \underset{\sim}{\delta}_n\} = 0$$

$$= E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T \underset{\sim}{z}(n)\}$$

$$- E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T \Phi^T H^T H\Phi\ \hat{\underset{\sim}{x}}(n-1)\}$$

Note that $\hat{\underset{\sim}{x}}(n-1)$ is a linear combination of $\{\underset{\sim}{z}(i), \underset{\sim}{z}(i+1), \ldots, \underset{\sim}{z}(n-1)\}$. Thus, from Lemma 3-1, the second term in the above equation is zero. Consequently,

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T B(n,k)\ \underset{\sim}{z}(k)\} = 0 \qquad \text{for all } k \leq n$$

or equivalently

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T\ \underset{\sim}{y}(n)\} = 0 \qquad \text{for all } \underset{\sim}{y}(n) \in Y(n)$$

where $\underset{\sim}{y}(n) = \sum_{k=i}^{n} B(n,k)\underset{\sim}{z}(k)$ Q.E.D.

This theorem is important because it states that the filter is optimal if and only if the residual time series $\{\underset{\sim}{\delta}_n\}$ is an orthogonal series. Since the residual process is observable, this theorem provides a method for determining whether or not the filter matrix is optimum. But if it is suboptimum, Theorem 3-2 does not indicate how it should be altered to approach K_{opt}. Consequently, it fails to meet the second characteristic required of a learning criterion. This deficiency motivates Theorem 3-3 which is used in Chapter 4 to develop algorithms that learn K_{opt}.

THEOREM 3-3: The Learning Criterion

Given the system of Theorem 3-2, a necessary and sufficient condition for the satisfaction of the Wiener-Hopf equation

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(j)\} = [0] \qquad \text{for all } n, j. \quad (2.6)$$

where $j \leq n$

is

$$E\{[\underset{\sim}{\Delta}_{j+1} - H\Phi\ K_{opt}\ \underset{\sim}{\delta}_j]\ \underset{\sim}{\delta}_j^T\} = [0] \qquad \text{for all } j \leq n \qquad (3.6)$$

Proof: From the development of Eq. (3.2) from Eq. (3.1), Eq. (3.6) may be written

$$E\{[\underset{\sim}{\Delta}_{j+1} - H\Phi\, K_{opt}\underset{\sim}{\delta}_j]\ \underset{\sim}{\delta}_j^T\} = E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\}$$

$$= [0] \qquad \text{for all } j \leq n \qquad (3.6)$$

Now assume the Wiener-Hopf equation (2.6) holds, then

$$E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = E\{[\underset{\sim}{z}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)][\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]^T\}$$

$$= E\{[H\ \underset{\sim}{x}(j+1) + \underset{\sim}{v}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]\ \delta_n^T\}$$

$$= E\{[H\Phi\ \underset{\sim}{x}(j) + \underset{\sim}{w}(j) - H\Phi\hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T\}$$

$$= H\Phi\ E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{z}^T(j)\}$$

$$- H\Phi\ E\{[\underset{\sim}{x}(j) - \underset{\sim}{x}(j)]\ \hat{\underset{\sim}{x}}^T(j-1)\}\ \Phi^T\ H^T$$

The first term is zero directly from (2.6). The second term is also zero from (2.6) because $\hat{\underset{\sim}{x}}(j-1)$ is a linear combination of $\underset{\sim}{z}(k)$, $k \leq j-1$. Thus,

$$E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = [0] \qquad \text{for all } j \leq n$$

Converse: Assume (3.6) is true.

As a consequence of this assumption, the following preliminary result is obtained

$$E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = E\{[\underset{\sim}{z}(j+1) - H\Phi\ \hat{\underset{\sim}{x}}(j)][\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]^T\}$$

$$= E\{[H\underset{\sim}{x}(j+1) + \underset{\sim}{v}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T\}$$

$$= HE\{[\underset{\sim}{x}(j+1) - \Phi\hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T\}$$

$$= H\Phi E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T = [0] \qquad \text{for all } j \leq n$$

Since the inverse of H exists,

$$E\{[\underset{\sim}{x}(j) - \hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T\} = [0] \qquad \text{for all } j \leq n \tag{3.7}$$

It must now be shown that

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_j^T\} = [0] \qquad \text{for all } j \leq n$$

This is accomplished by iterating backward in time, starting with j = n-1. Proceeding as in Theorem 3-2,

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_{n-1}^T\} = E\{[\underset{\sim}{z}(n+1) - H\Phi\hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}_{n-1}^T\}$$

$$= E\{[H\underset{\sim}{x}(n+1) + \underset{\sim}{v}(n+1) - H\Phi\hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}_{n-1}^T\}$$

$$= HE\{[\Phi\underset{\sim}{x}(n) + \underset{\sim}{w}(n) - \Phi\hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}_{n-1}^T\}$$

$$= H\Phi\ E\{[\Phi\underset{\sim}{x}(n-1) + \underset{\sim}{w}(n-1) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}^T_{n-1}\}$$

$$= H\Phi\ E\{[\Phi\underset{\sim}{x}(n-1) - \Phi\hat{\underset{\sim}{x}}(n-1) - K[\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)]]\ \underset{\sim}{\delta}^T_{n-1}\}$$

$$= H\Phi\Phi\ E\{[\underset{\sim}{x}(n-1) - \hat{\underset{\sim}{x}}(n-1)]\ \underset{\sim}{\delta}^T_{n-1}\} -$$

$$H\Phi K\ E\{[\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)]\ \underset{\sim}{\delta}^T_{n-1}\}$$

The first term is zero from Eq. (3.7) and the second term is zero from Eq. (3.6). Therefore,

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}^T_{n-1}\} = [0] \quad .$$

Using an identical argument and continuing the iteration, it can be shown that

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}^T_{j}\} = [0] \qquad \text{for all } j \leq n \tag{3.8}$$

A more useful form of the above equation is obtained by substituting the observation equation for $\underset{\sim}{z}(n+1)$ and the state equation for $\underset{\sim}{x}(n+1)$. It is

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}^T_{j}\} = [0] \qquad \text{for all } j \leq n \tag{3.9}$$

Proceeding inductively, it can now be shown that the Wiener-Hopf equation follows from Eq. (3.9). In particular, for $j = i+1$, it becomes

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}^T_{i+1}\} = [0]$$

and since

$$\hat{\underset{\sim}{x}}(i) = \underset{\sim}{0}$$

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)][\underset{\sim}{z}(i+1) - H\Phi\hat{\underset{\sim}{x}}(i)]^T\}$$

$$= E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(i+1)\} = [0]$$

Again proceeding by induction, assume that

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}_k^T\} = [0] \qquad \text{for all } k \leq n-1$$

or equivalently

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(k)\} = [0] \qquad \text{for all } k \leq n-1$$

Now for k = n, Eq. (3.9) becomes

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{\delta}_n^T\} = [0]$$

$$= E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(n)\} - E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \hat{\underset{\sim}{x}}(n-1)\}\Phi^T H^T$$

Because $\hat{\underset{\sim}{x}}(n-1)$ is just a linear combination of $\{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n-1)\}$, the second term in the last equation is null. Therefore, it follows that

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\ \underset{\sim}{z}^T(k)\} = [0] \qquad \text{for all } k \leq n$$

Q.E.D.

This theorem states that the lack of orthogonality when K is not equal to the optimal filter K_{opt} is reflected in a non-null correlation matrix and conversely. That is, $K = K_{opt}$ if and only if

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n\} \triangleq \Omega(n+1,n)$$

$$= [0] \qquad (3.10)$$

$\Omega(n+1,n)$ can be used as a basis for learning K_{opt}, as shown in Chapter 4, by devising a scheme to utilize this correlation to adjust K_n[1] such that it converges to K_{opt} as n approaches infinity.

The following corollary is a direct consequence of Theorem 3-3.

Corollary 3-4

Equation (3.6), $E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = [0]$ for all j implies $K = \Sigma H^T[H \Sigma H^T + R]^{-1}$ which is the optimum Kalman filter matrix.

Proof:

$$E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = E\{[\underset{\sim}{z}(j+1) - H\Phi\hat{\underset{\sim}{x}}(j)][\underset{\sim}{z}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]^T\}$$

$$= H\, E\{[\underset{\sim}{x}(j+1) - \Phi\hat{\underset{\sim}{x}}(j)]\ \underset{\sim}{\delta}_j^T\}$$

Substituting Eq. (2.19) for $\hat{\underset{\sim}{x}}(j)$ gives

$$= H\Phi\, E\{[\underset{\sim}{x}(j) - \Phi\hat{\underset{\sim}{x}}(j-1) - K[H\underset{\sim}{x}(j) + \underset{\sim}{v}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]]\ \underset{\sim}{\delta}_j^T\}$$

$$= H\Phi\, E\{[\underset{\sim}{x}(j) - \Phi\hat{\underset{\sim}{x}}(j-1)][\underset{\sim}{x}(j) - \Phi\hat{\underset{\sim}{x}}(j-1)]^T\}\ H^T$$

$$- H\Phi K\, E\{[H\underset{\sim}{x}(j) + \underset{\sim}{v}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)][H\underset{\sim}{x}(j) + \underset{\sim}{v}(j) - H\Phi\hat{\underset{\sim}{x}}(j-1)]^T\}$$

$$= H\Phi\Sigma H^T - H\Phi KH\, E\{[\underset{\sim}{x}(j) - \Phi\hat{\underset{\sim}{x}}(j-1)][\underset{\sim}{x}(j) - \Phi\hat{\underset{\sim}{x}}(j-1)]\}\ H^T$$

$$- H\Phi K\, E\{\underset{\sim}{v}(j)\ \underset{\sim}{v}^T(j)\}$$

---

1. Since in the adaptive process the filter matrix is being changed as n increases, its value at time n is denoted K_n.

$$= H\Phi\Sigma H^T - H\Phi K\,[H\Sigma\, H^T + R] = [0]$$

Solving for K yields

$$K = \Sigma H^T\,[H\Sigma\, H^T + R]^{-1}$$

which is the optimal Kalman filter.

Q.E.D.

3.3 A Linear Regression Model of the Learning Criterion

Since Theorem 3-3 indicates through the correlation between $\underset{\sim}{\delta}_{n+1}$ and $\underset{\sim}{\delta}_n$ whether or not the filter matrix is optimal, it satisfies the first requirement for a learning criterion. To show that it permits the development of a method for satisfying the second requirement and, therefore, qualifies as an unsupervised learning criterion, it is convenient to represent the correlation between the residuals as a linear regression

$$\underset{\sim}{\delta}_{n+1} = P_n \underset{\sim}{\delta}_n + \underset{\sim}{e}_n \tag{3.11}$$

where P_n is the regression matrix and $\underset{\sim}{e}_n$ is the error term. Post-multiplying both sides of Eq. (3.11) by $\underset{\sim}{\delta}_n^T$ and taking the expected value gives

$$E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n^T\} = P_n\, E\{\underset{\sim}{\delta}_n\underset{\sim}{\delta}_n^T\} + E\{\underset{\sim}{e}_n\underset{\sim}{\delta}_n^T\}$$

or

$$\Omega(n+1,n) = P_n\ \Omega(n,n) + E\{\underset{\sim}{e}_n \underset{\sim}{\delta}_n^T\} \tag{3.12}$$

Selecting the regression matrix to be

$$P_n = \Omega(n+1,n)\ \Omega^{-1}(n,n) \tag{3.13}$$

forces

$$E\{\underset{\sim}{e}_n \underset{\sim}{\delta}_n^T\} = [0]$$

P_n then represents the correlation between $\underset{\sim}{\delta}_{n+1}$ and $\underset{\sim}{\delta}_n$. It is easy to see that when $P_n = [0]$ for all n, the learning criterion is satisfied and conversely. Thus, an equivalent statement of Theorem 3-3 can be made in terms of P_n.

THEOREM 3-4:

Given the system of Theorem 3-2, the filter matrix K is optimal if and only if

$$P_n = [0] \qquad \text{for all n.}$$

Thus, an algorithm is required which utilizes P_n to adjust K_n in a way that forces P_n to converge to the null matrix as n approaches infinity. Equivalently, the adjustment must force $\underset{\sim}{\delta}_{n+1}$ to approach $\underset{\sim}{e}_n$ as n increases without bound. The method for achieving this goal is deferred to Chapter 4, where it is shown that P_n is proportional to the difference between the optimum filter matrix K_{opt} and the actual filter matrix K_n.

3.4 Summary

This chapter has developed a learning criterion which is a necessary and sufficient condition for optimality of the filter matrix. Furthermore, it is a function of measurable quantities only and it specifies what must be accomplished to learn K_{opt}. The learning process was reduced to a regression equation such that when the correlation matrix P_n is null, Theorem 3-3 is satisfied. In the next chapter, the regression equation is used to indicate how the adaptation should be performed. Stochastic algorithms are derived from a mean-square error performance index to estimate the optimum filter matrix. The theory of stochastic approximation is invoked to prove that the adaptation algorithms converge.

CHAPTER 4

DERIVATION OF THE ADAPTIVE ALGORITHMS

4.1 Introduction and Organization of the Chapter

This chapter is concerned with the derivation of stochastic algorithms which converge to the optimum Kalman filter matrix. This development is based on the unsupervised learning criterion of Theorem 3-3. Section 4.2 utilizes the regression model of the learning procedure to motivate the adaptation process. Section 4.3 introduces the theory of stochastic approximation which provides the theoretical foundation for this chapter. In Section 4.4 a performance index is introduced from which the adaptive learning algorithms are derived. Section 4.5 uses the theory of stochastic approximation and the learning criterion (Theorem 3-3) to prove probability one convergence of the algorithms. A method for accelerating the rate of convergence is given in Section 4.6. Section 4.7 summarizes the chapter.

4.2 Motivation for the Adaptive Process

At the end of Chapter 3 the learning criterion was reduced to a linear regression

$$\underset{\sim}{\delta}_{n+1} = \underset{\sim}{\Delta}_{n+1} - H\Phi K_n \underset{\sim}{\delta}_n = P_n \underset{\sim}{\delta}_n + \underset{\sim}{e}_n \tag{3.11}$$

such that the learning criterion is satisfied if and only if $P_n = [0]$. From Theorem 3-4 this condition implies that the filter matrix is optimum and conversely.

Consider now the correlation P_n that exists when the filter matrix is suboptimum. The goal is to use this correlation matrix to force K_n to approach K_{opt} as n approaches infinity. As a consequence, P_n must converge to the null matrix or equivalently $\underset{\sim}{\delta}_{n+1}$ must approach $\underset{\sim}{e}_n$ in the limit. The role of P_n in accomplishing the above goal is made evident by first rewriting Eq. (3.11) as

$$\underset{\sim}{e}_n = \underset{\sim}{\delta}_{n+1} - P_n \underset{\sim}{\delta}_n \tag{4.1}$$

and setting

$$P_n = H\Phi\Delta K_n \tag{4.2}$$

where ΔK_n at this point is an arbitrary matrix chosen to satisfy Eq. (4.2). The following convention is observed in the ensuing development: a suboptimal estimate obtained using the structure of the filtering Eq. (2.19), i.e., the filter used is not optimal, is denoted $\hat{\underset{\sim}{x}}_n$ to distinguish it from the optimal estimate denoted $\hat{\underset{\sim}{x}}(n)$. Assume that optimum estimation has occurred up to, but not including, time n. Now substituting for $\underset{\sim}{\delta}_{n+1}$, $\underset{\sim}{\delta}_n$ and P_n in Eq. (4.1) gives

$$\underset{\sim}{e}_n = z(n+1) - H\Phi\hat{\underset{\sim}{x}}_n - H\Phi\Delta K_n [\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)]$$

By replacing $\hat{\underset{\sim}{x}}_n$ with the filtering equation, the above expression becomes

$$\underset{\sim}{e}_n = \underset{\sim}{z}(n+1) - H\Phi \{\Phi\hat{\underset{\sim}{x}}(n-1) + K_n[\underset{\sim}{z}(n) - H\Phi\, \hat{\underset{\sim}{x}}(n-1)]\}$$

$$- H\Phi\Delta K_n [\underset{\sim}{z}(n) - H\Phi\, \hat{\underset{\sim}{x}}(n-1)]$$

$$= \underset{\sim}{z}(n+1) - H\Phi \{\Phi\underset{\sim}{\hat{x}}(n-1) + (K_n + \Delta K_n)[\underset{\sim}{z}(n) - H\Phi\ \underset{\sim}{\hat{x}}(n-1)]\}$$

Observe that ΔK_n appears as a correction to the filtering matrix K_n employed at time n. If $K_n + \Delta K_n$ had been used instead, the correlation P_n would have been zero. This means that $K_n + \Delta K_n = K(n)$, thus the equation for $\underset{\sim}{e}_n$ reduces to

$$\underset{\sim}{e}_n = \underset{\sim}{z}(n+1) - H\Phi \{\Phi\underset{\sim}{\hat{x}}(n-1) + K(n)[\underset{\sim}{z}(n) - H\Phi\ \underset{\sim}{\hat{x}}(n-1)]\}$$

$$= \underset{\sim}{z}(n+1) - H\Phi\ \underset{\sim}{\hat{x}}(n)$$

$$= \underset{\sim}{\delta}_{n+1} \qquad (4.3)$$

Equation (4.3) implies that the filter matrix has been adjusted so that the new correlation matrix is null. Thus from Theorem 3-4, the new filter matrix $K(n) = K_n + \Delta K_n$ is the optimum Kalman filter. The value of ΔK_n as determined from (4.2) is

$$\Delta K_n = (H\Phi)^{-1} P_n \qquad (4.4)$$

under the assumption that H inverse exists. However, since the distribution function necessary to determine P_n is unknown, ΔK_n cannot be calculated. But P_n and, therefore ΔK_n, can be estimated by using the theory of stochastic approximation. In the next section a brief introduction to the subject is presented.

4.3 Stochastic Approximation

Basically, stochastic approximation is an algorithmic search technique for optimization in the presence of uncertainty. This

uncertainty or noise may arise from an ignorance of the underlying phenomena, experimental errors or inherent random fluctuations. The nomenclature stochastic approximation emphasizes the stochastic nature of the errors in, say, the process measurements and the use of these measurements to approximate the location of the goal.

Such stochastic search algorithms are naturally more complex than their deterministic counterparts. However, the fundamental considerations are the same:

(1) selecting a direction in which to search, then

(2) selecting the distance to move (choosing the step size).

The effect of random error on an algorithm may cause it to converge to some non-optimum value or even to diverge. Therefore, correct convergence takes priority over speed of convergence. In stochastic approximation this consideration is reflected in the choice of step sizes. The direction to move is selected as if the process was deterministic. This is reminiscent of the Separation Theorem of control theory. The neglect of the stochastic nature of the problem means that some step directions may be incorrect, but such setbacks are swamped-out in the long run by additional data if the step sizes are properly selected. This is the crux of stochastic approximation and is directly related to the law of large numbers.

A powerful theorem due to Gladyshev [1965] serves to lay the mathematical groundwork for stochastic approximation and is used later in conjunction with the Learning Criterion (Theorem 3.3) to prove convergence of the algorithm for learning the optimum Kalman filter matrix.

THEOREM 4.1:

Let $\{y_n: n=1, 2, \ldots\}$ be an observable random sequence which depends on a real parameter α such that

$$E\{y_n(\alpha)\} = f_n(\alpha) \tag{4.5}$$

and for the real constant C,

$$f_n(\alpha_{opt}) = C \tag{4.6}$$

$$f_n(\alpha) < C \quad \text{for} \quad \alpha < \alpha_{opt}$$

$$f_n(\alpha) > C \quad \text{for} \quad \alpha > \alpha_{opt}$$

The value of α which satisfies the regression equation (4.6) is estimated by observing y_n and using the recursive relation

$$\alpha_{n+1} = \alpha_n - a_n [y_n(\alpha_n) - C] \tag{4.7}$$

If $\{a_n\}$ is a sequence of positive numbers such that

(i) $$\sum_{n=1}^{\infty} a_n = \infty$$

(ii) $$\sum_{n=1}^{\infty} a_n^2 < \infty$$

and

(iii) $$\text{g.l.b.}_{|\alpha_n - \alpha_{opt}| < \varepsilon} (\alpha_n - \alpha_{opt}) [f_n(\alpha_n) - C] \leq 0 \quad \text{for all } \varepsilon > 0$$

(iv) There exists a positive number r such that for all $\underset{\sim}{\alpha}$

$$E\{y_n^2(\alpha)\} < r(1 + \alpha^2)$$

then the sequence $\{\alpha_n\}$ defined by (4.7) converges with probability one to α_{opt}, i.e., $P\{\lim_{n \to \infty} \alpha_n = \alpha_{opt}\} = 1$.

If C = 0, then the stochastic approximation sequence (4.7) becomes an algorithm for finding the unique zero of $f_n(\alpha)$. Historically, this application was the motivation for the original work performed by the statisticians Robbins and Monro [1951]. Dvoretzky [1956] extended and unified their work and that of other investigators. However, the viewpoint is strictly that of mathematical statistics. A comprehensive survey of stochastic approximation and its engineering applications is contained in Hampton [1967]. For an in-depth study of the subject, the reader is referred to the interesting monographs by Albert and Gardner [1967] and Fu [1968].

4.4 Stochastic Algorithms for Learning the Optimum Filter Matrix

To facilitate the application of stochastic approximation, the problem of estimating P_n is rewritten in the structure of a performance index. Let $L(P_n)$ be the expected value of the convex performance index to be minimized,

$$L(P_n) = E\{\ell(\underset{\sim}{\delta}_{n+1} - \underset{\sim}{\hat{\delta}}_{n+1})\} \tag{4.8}$$

where $\ell(\cdot)$ is the loss function and $\underset{\sim}{\hat{\delta}}_{n+1} = P_n \underset{\sim}{\delta}_n$. When $L(P_n)$ is known (the deterministic case), Eq. (4.8) can be minimized by solving Eq. (4.9) iteratively.

$$\nabla_p L(P_{opt}) = E\{\nabla_p \ell(\underset{\sim}{\delta}_{n+1} - \hat{\underset{\sim}{\delta}}_{n+1})\} = [0] \tag{4.9}$$

Using standard computational techniques such as the gradient method, the recursive relation for finding P_{opt} is

$$\hat{P}_{n+1} = \hat{P}_n + \nabla_p L(\hat{P}_n) W_{n+1} \tag{4.10}$$

where $\{W_{n+1}\}$ is a weighting sequence which determines the magnitude of the correction term. Under appropriate convergence conditions, $\lim_{n \to \infty} \hat{P}_n = P_{opt}$. A block diagram, Fig. 4.1, of the process generated by the deterministic algorithm (4.10) reveals it to be a nonlinear feedback system. It has no input which implies that all the information needed to determine P_{opt} is included in the system a priori.

In an adaptive stochastic setting the probability distribution function required to perform the expectation indicated by Eq. (4.8) is not available. Thus, the performance index $L(P_n)$ and its gradient are unknown. This condition is precisely the motivation for stochastic approximation. What is needed is a method to utilize the available information, specifically the gradient of the loss function, to solve Eq. (4.9). By identifying $\nabla_p \ell(\underset{\sim}{\delta}_{n+1} - \hat{\underset{\sim}{\delta}}_{n+1})$ with the random sequence $y_n(\alpha)$ of Eq. (4.5) and using the stochastic approximation algorithm (4.7) with $C = 0$, the following recursive relation is obtained

$$\hat{P}_{n+1} = \hat{P}_n - a_{n+1} \nabla_p \ell(\underset{\sim}{\delta}_{n+1} - \hat{P}_n \underset{\sim}{\delta}_n) W_{n+1} \tag{4.11}$$

where $\{a_n\}$ satisfies conditions (i) and (ii) of Theorem 4.1 and $\{W_{n+1}\}$

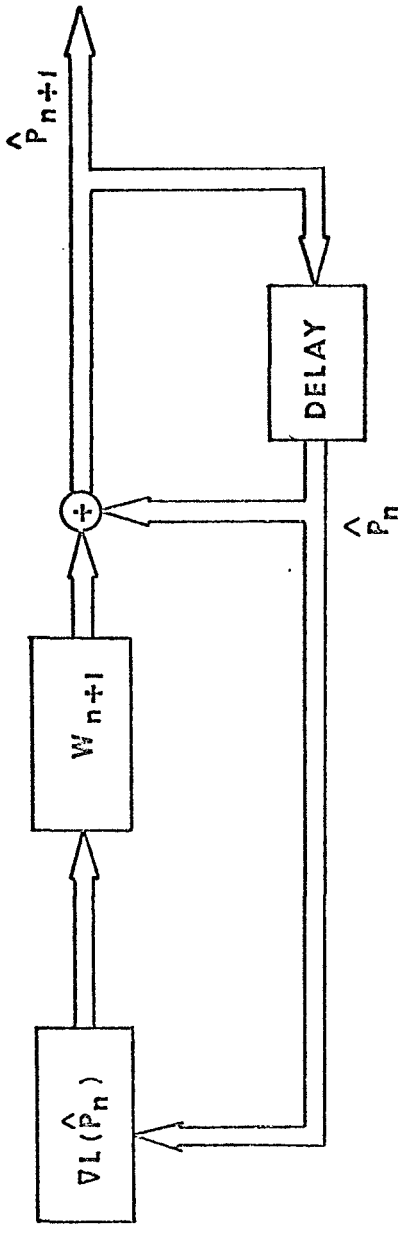


Fig. 4.1 Block Diagram of the Feedback System for the Deterministic Algorithm (4.10)

is an appropriate set of weighting terms. If W_{n+1} is chosen to be the identity matrix for all n, then Eq. (4.11) reduces to a standard stochastic approximation recursion. In general, W_{n+1} is a matrix chosen to accelerate the rate of convergence. By defining

$$G_n \triangleq a_n W_n \tag{4.12}$$

Eq. (4.11) becomes

$$\hat{P}_{n+1} = \hat{P}_n - \nabla_p \, \ell(\delta_{n+1} - \hat{P}_n \delta_n) G_{n+1} \tag{4.13}$$

This gradient descent algorithm is the direct stochastic analogue of the deterministic algorithm (4.10). Since Eq. (4.13) is a random sequence, its convergence must be established in a probabilistic sense. A block diagram realization of this process is illustrated in Fig. 4.2. It represents a nonlinear feedback system similar to that of Fig. 4.1, but it is forced. This important difference implies that a continuous input of external information is necessary to solve Eq. (4.9) iteratively in an unknown random environment. The concept of viewing the algorithm as a feedback system means that algorithmic convergence may be interpreted in terms of system stability.

To proceed with the development of Eq. (4.11), a differentiable and convex loss function must be specified. A quadratic loss function

$$\ell(\underset{\sim}{\delta}_{n+1} - \underset{\sim}{\hat{\delta}}_{n+1}) = \frac{1}{2} (\underset{\sim}{\delta}_{n+1} - P_n \underset{\sim}{\delta}_n)^T (\underset{\sim}{\delta}_{n+1} - P_n \underset{\sim}{\delta}_n) \tag{4.14}$$

is selected because it permits an optimum selection of the weighting matrices W_{n+1} and a systematic proof that the associated algorithm

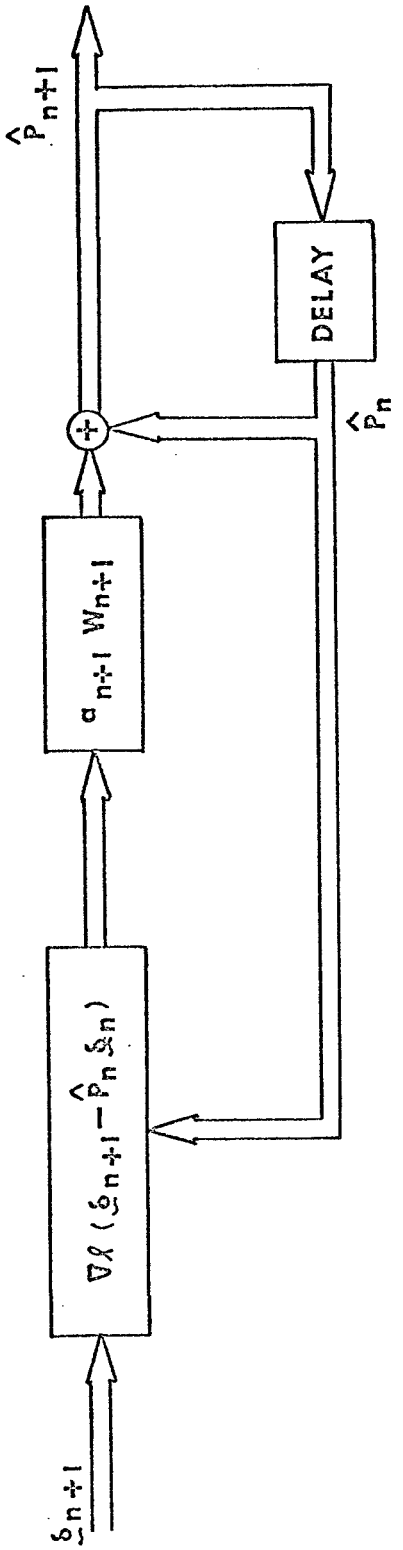


Fig. 4.2 Block Diagram of the Feedback System for the Adaptive Stochastic Algorithm (4.13)

converges to the Kalman filter matrix. This choice of loss function results in the familiar mean square performance index. Substituting the gradient of $\ell(\cdot)$ with respect to P_n into Eq. (4.11) gives

$$\hat{P}_{n+1} = \hat{P}_n + a_{n+1}[(\underset{\sim}{\delta}_{n+1} - \hat{P}_n \underset{\sim}{\delta}_n)\underset{\sim}{\delta}_n^T] \; W_{n+1} \tag{4.15}$$

It is readily verified that conditions (i) and (ii) of Theorem 4.1 are satisfied by any sequence of the form $(n)^{-b}$, where $1/2 < b \leq 1$. Therefore, letting $a_n = n^{-1}$, the harmonic series, gives the fastest rate of asymptotic convergence. Using this sequence, Eq. (4.15) becomes

$$\hat{P}_{n+1} = \hat{P}_n + [(\underset{\sim}{\delta}_{n+1} - \hat{P}_n \underset{\sim}{\delta}_n)\underset{\sim}{\delta}_n^T] \; \frac{W_{n+1}}{n+1}$$

$$= \hat{P}_n + \{[\underset{\sim}{z}(n+1) - H\Phi\hat{\underset{\sim}{x}}_n] - \hat{P}_n[\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}_{n-1}]\} \; \underset{\sim}{\delta}_n^T \; \frac{W_{n+1}}{n+1}$$

Substituting $\hat{P}_n = H\Phi \; \Delta\hat{K}_n$ from (4.2) into the above equation, results in

$$\hat{P}_{n+1} = \hat{P}_n + \{\underset{\sim}{z}(n+1) -$$

$$H\Phi[\hat{\underset{\sim}{x}}_n + \Delta\hat{K}_n(\underset{\sim}{z}(n) - H\Phi \; \hat{\underset{\sim}{x}}_{n-1})]\} \; \underset{\sim}{\delta}_n^T \; \frac{W_{n+1}}{n+1}$$

Using the filtering equation for $\hat{\underset{\sim}{x}}_n$, it is seen that

$$\hat{P}_{n+1} = \hat{P}_n + \{\underset{\sim}{z}(n+1) -$$

$$H\Phi \; [\Phi\hat{\underset{\sim}{x}}_{n-1} + (\hat{K}_n + \Delta\hat{K}_n)(\underset{\sim}{z}(n) - H\Phi \; \hat{\underset{\sim}{x}}_{n-1})]\} \; \underset{\sim}{\delta}_n^T \; \frac{W_{n+1}}{n+1}$$

The expression in brackets is just an updated estimate of $\underset{\sim}{x}(n)$, therefore,

$$\hat{P}_{n+1} = \hat{P}_n + \{\underset{\sim}{z}(n+1) - H\Phi\ \hat{\underset{\sim}{x}}_n\}\ \underset{\sim}{\delta}_n^T\ \frac{W_{n+1}}{n+1}$$

$$= \hat{P}_n + \underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n^T\ \frac{W_{n+1}}{n+1} \qquad (4.16)$$

Assuming the inverse of H exists,

$$\Delta\hat{K}_{n+1} = (H\Phi)^{-1}\ \hat{P}_{n+1} \qquad (4.17)$$

Instead of estimating P_{opt} and then calculating $\Delta\hat{K}_{n+1}$ as required when using the stochastic algorithm (4.16), it is preferable to estimate K_{opt} directly. This is accomplished by re-defining the loss function $\ell(\cdot)$. Let

$$\Delta_{n+1} = (H\Phi)^{-1}[\underset{\sim}{z}(n+1) - H\Phi\Phi\ \hat{\underset{\sim}{x}}(n-1)]$$

with $\underset{\sim}{\delta}_n = \underset{\sim}{z}(n) - H\Phi\ \hat{\underset{\sim}{x}}(n-1)$ as before, and

$$\ell(\underset{\sim}{\Delta}_{n+1} - \hat{\underset{\sim}{\Delta}}_{n+1}) = \frac{1}{2}\ (\underset{\sim}{\Delta}_{n+1} - K_n\underset{\sim}{\delta}_n)^T(\underset{\sim}{\Delta}_{n+1} - K_n\underset{\sim}{\delta}_n) \qquad (4.18)$$

Then, the iterative solution to

$$\nabla_K L(K_{opt}) = \nabla_K E\{\ell(\underset{\sim}{\Delta}_{n+1} - K_{opt}\underset{\sim}{\delta}_n)\}$$

$$= E\{\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - K_{opt}\underset{\sim}{\delta}_n)\}$$

$$= [0] \qquad (4.19)$$

is given by the stochastic approximation algorithm

$$\hat{K}_{n+1} = \hat{K}_n + a_{n+1}\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n)\, W_{n+1} \tag{4.20}$$

$$= \hat{K}_n + a_{n+1}\{[\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n]\}\, \underset{\sim}{\delta}_n^T W_{n+1}$$

$$= \hat{K}_n + a_{n+1}\, (H\Phi)^{-1}[z(n+1) - H\,\Phi\Phi x(n-1)] -$$

$$\hat{K}_n[\underset{\sim}{z}(n) - H\,\Phi\,\underset{\sim}{\hat{x}}(n-1)]\}\, \underset{\sim}{\delta}_n^T W_{n+1}$$

$$= \hat{K}_n + a_{n+1}\, (H\Phi)^{-1}\, \underset{\sim}{\delta}_{n+1} \underset{\sim}{\delta}_n^T W_{n+1} \tag{4.21}$$

where $\left\{a_n = \frac{1}{n}\right\}$.

The recursive relation of Eq. (4.21) is the desired stochastic algorithm for learning the Kalman filter matrix K_{opt}. In the following section it is shown that Eq. (4.20) converges with probability one.

4.5 Convergence of the Stochastic Learning Algorithms

Proving that Eq. (4.20) converges to the solution of Eq. (4.19) only indicates that the performance index

$$\frac{1}{2} E\{(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)^T (\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\}$$

is minimized. It does not demonstrate that this minimum is also the Kalman filter. This is the first task at hand.

THEOREM 4-2:

The value of the filter that minimizes

$$L(K_n) = \frac{1}{2} E\{(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)^T (\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\} \tag{4.22}$$

coincides with the optimum Kalman filter K_{opt}.

Proof: Consider Eq. (4.19)

$$E\{\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\} = - E\{(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\ \underset{\sim}{\delta}_n^T\}$$

Using the definition of $\underset{\sim}{\Delta}_{n+1}$ and $\underset{\sim}{\delta}_n$ in the above relation gives

$$\begin{aligned} & E\{\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\} \\ & = - E\{(H\Phi)^{-1}\{[\underset{\sim}{z}(n+1) - H\Phi\Phi\ \underset{\sim}{\hat{x}}(n-1)] - K_n[\underset{\sim}{z}(n) - H\Phi\ \underset{\sim}{\hat{x}}(n-1)]\}\ \underset{\sim}{\delta}_n^T\} \\ & = - (H\Phi)^{-1} E\{\underset{\sim}{\delta}_{n+1} \underset{\sim}{\delta}_n^T\} \end{aligned} \tag{4.23}$$

Now taking the limit of Eq. (4.19) as K_n approaches K_{opt} and invoking the results of Theorem 3-3 (the Learning Criterion) yields

$$\begin{aligned} \lim_{K_n \to K_{opt}} E\{\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\} &= - (H\Phi)^{-1} \lim_{K_n \to K_{opt}} E\{\underset{\sim}{\delta}_{n+1} \underset{\sim}{\delta}_n^T\} \\ &= [0] \end{aligned} \tag{4.24}$$

Since the solution to Eq. (4.24) minimizes the performance index $L(K_n)$, the proof of the theorem is complete.

Q.E.D.

Now since $L(K_n)$ is a quadratic function of K_n, proving that algorithm (4.20) converges with probability one to the minimum of $L(K_n)$ implies that it converges to K_{opt} with probability one. The scalar case is considered first.

THEOREM 4-3:

Given the performance index

$$L(K_n) = E\{\ell(\Delta_{n+1} - K_n\delta_n)\}$$

where

$$\ell(\Delta_{n+1} - \hat{\Delta}_{n+1}) = \frac{1}{2}(\Delta_{n+1} - K_n\delta_n)(\Delta_{n+1} - K_n\delta_n) \tag{4.18}$$

and the stochastic algorithm

$$\hat{K}_{n+1} = \hat{K}_n - a_{n+1} \nabla_K \ell(\Delta_{n+1} - \hat{K}_n\delta_n) W_{n+1} \tag{4.20}$$

$$= \hat{K}_n + a_{n+1}(H\Phi)^{-1} \delta_{n+1} \delta_n W_{n+1} \tag{4.21}$$

where $\{a_n\}$ satisfies conditions (i) and (ii) and $\{W_{n+1}\}$ is a uniformly bounded sequence, then Eq. (4.21) converges with probability one to the minimum of the performance index which from Theorem 4-2 is equal to the optimum Kalman filter.

Proof:

By comparing Eqns. (4.20) and (4.22) with Eqns. (4.5) and (4.7), one observes that the gradient of the loss function corresponds to the random sequence $y_n(\alpha_n)$ and the expected value of the gradient of the loss function corresponds to $f_n(\alpha_n)$. Therefore, conditions (i) - (iv) of Theorem 4-1 become

(i)
$$\sum_{n=1}^{\infty} a_n W_n = \infty$$

(ii)
$$\sum_{n=1}^{\infty} (a_n W_n)^2 < \infty$$

(iii)
$$\text{g.l.b.}_{|K_n - K_{opt}| < \varepsilon} (K_n - K_{opt})\, E\{\nabla_K \ell(\Delta_{n+1} - K_n \delta_n)\} \geq 0 \text{ for all } \varepsilon > 0$$

(iv)
$$E\{[\nabla_K \ell(\Delta_{n+1} - K_n \delta_n)]^2\} \leq r(1 + K_n)^2$$

Since W_n is uniformly bounded for all n, conditions (i) and (ii) are immediately satisfied. Condition (iv) simply requires that the loss function not increase more rapidly than a quadratic in K_n. The loss function of Eq. (4.18) meets this restriction.

The only condition that remains to be verified is (iii). It is now considered.

The left hand side of inequality (iii) may be written

$$E\{(K_n - K_{opt})\,[\nabla_K \ell(\Delta_{n+1} - K_n \delta_n)]\}$$

$$= - E\{(K_n - K_{opt})\delta_n [\ell'(\Delta_{n+1} - K_n \delta_n)]\}^1$$

$$= - E\{(K_n - K_{opt})\delta_n [\ell'(\Delta_{n+1} - (K_n - K_{opt})\,\delta_n - K_{opt}\delta_n)]\}$$

$$= - E\{\Lambda_n \ell'(\psi_{n+1} - \Lambda_n)\} \geq 0$$

---

1. The notation $\ell'(\cdot)$ denotes the derivative of ℓ with respect to its argument.

or equivalently

$$E\{\Lambda_n \ell'(\psi_{n+1} - \Lambda_n)\} \leq 0$$

where

$$\Lambda_n \triangleq (K_n - K_{opt})\ \delta_n$$

and

$$\psi_{n+1} \triangleq \Delta_{n+1} - K_{opt}\ \delta_n$$

For the optimum filter matrix, Eq. (4.23) gives

$$E\{\nabla_K \ell(\Delta_{n+1} - K_{opt}\delta_n)\} = - E\{\ell'(\Delta_{n+1} - K_{opt}\delta_n)\delta_n\}$$

$$= - E\{\delta_n \ell'(\psi_{n+1})\} = 0 \qquad (4.25)$$

Therefore,

$$(K_n - K_{opt})\ E\{\delta_n\ \ell'(\psi_{n+1})\} = E\{(K_n - K_{opt})\ \delta_n\ \ell'(\psi_{n+1})\}$$

$$= E\{\Lambda_n\ \ell'(\psi_{n+1})\}$$

$$= 0 \qquad (4.26)$$

By virtue of the convexity of $\ell(\cdot)$, the following property holds

$$\ell'(\psi_m) - \ell'(\psi_p) \leq 0 \text{ for all } \psi_m \leq \psi_p$$

Consequently,

$$\ell'(\psi_{n+1} - \Lambda_n) - \ell'(\psi_{n+1}) \leq 0 \quad \text{for all } \Lambda_n > 0$$

Multiplying both sides of the above inequality by Λ_n gives

$$\Lambda_n \ell'(\psi_{n+1} - \Lambda_n) - \Lambda_n \ell'(\psi_{n+1}) \leq 0 \quad \text{for all } \Lambda_n > 0 \qquad (4.27)$$

Taking the expectation of (4.27) yields

$$E\{\Lambda_n \ell'(\psi_{n+1} - \Lambda_n)\} - E\{\Lambda_n \ell'(\psi_{n+1})\} \leq 0$$

The second term in this inequality is zero from Eq. (4.26). Thus,

$$E\{\Lambda_n \ell'(\psi_{n+1} - \Lambda_n)\} \leq 0$$

which confirms that condition (iii) is satisfied.

Q.E.D.

In general, algorithm (4.21) is a random matrix sequence. To use Theorem 4-1 in this case, the argument of Theorem 4-3 must be applied to each element of the matrix $E\{\nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - K_n \underset{\sim}{\delta}_n)\}$ separately.

4.6 Acceleration of Convergence

In Section (4.4) it was pointed out that if W_{n+1} is a constant, algorithms Eqns. (4.11) and (4.20) are basically stochastic gradient schemes. Letting W_{n+1} equal the identity matrix and focusing attention on Eq. (4.20) results in

$$\hat{K}_{n+1} = \hat{K}_n - \nabla_K \ell(\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n) \frac{I}{n+1}$$

$$= \hat{K}_n + \frac{1}{n+1} [\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n] \underset{\sim}{\delta}_n^T \qquad (4.28)$$

For this choice of W_{n+1} the correction term in Eq. (4.28) simply proceeds in the direction of the local gradient of the loss function, with its magnitude weighted by the factor $\frac{1}{n+1}$. This indicates that even though large corrections are possible at the beginning of the estimation process when $\hat{K}_n$ is far from K_{opt}, no information from past estimation stages is utilized. Note that the $\frac{1}{n+1}$ factor forces the correction term to approach zero as the estimation progresses, since it becomes more probable that the observed gradient is due to noise instead of estimation error.

To accelerate convergence a second order stochastic algorithm is developed. This is accomplished by applying Newton's method to the loss function. For the scalar case this procedure gives

$$\hat{K}_{n+1} = \hat{K}_n - \frac{1}{n+1} \nabla_K \ell(\Delta_{n+1} - \hat{K}_n \delta_n) \{\nabla_K [\nabla_K \ell(\Delta_{n+1} - \hat{K}_n \delta_n)]\}^{-1} \tag{4.29}$$

$$= \hat{K}_n + \frac{1}{n+1} (\Delta_{n+1} - \hat{K}_n \delta_n) \delta_n \{\delta_n \delta_n\}^{-1} \tag{4.30}$$

$$= \hat{K}_n + \frac{1}{n+1} (\Delta_{n+1} - \hat{K}_n \delta_n) \delta_n W_{n+1}$$

where

$$W_{n+1} = \{\delta_n \delta_n\}^{-1}$$

Even though Eq. (4.30) should converge more rapidly than Eq. (4.28), it does not utilize the information from past stages to select the step-size. One solution to this problem is to define an intermediate loss function

$$\ell_n(K_n) = \frac{1}{2n} \sum_{j=o}^{n-1} (\underset{\sim}{\Delta}_{j+1} - K_n \underset{\sim}{\delta}_j)^T (\underset{\sim}{\Delta}_{j+1} - K_n \underset{\sim}{\delta}_j) \tag{4.31}$$

The filter matrix $\hat{K}_n$ which minimizes Eq. (4.31) is a least squares solution to

$$\nabla_K \ell_n(\hat{K}_n) = \frac{1}{n} \sum_{j=o}^{n-1} (\underset{\sim}{\Delta}_{j+1} - \hat{K}_n \underset{\sim}{\delta}_j) \underset{\sim}{\delta}_j^T = [0] \tag{4.32}$$

Solving for $\hat{K}_n$ yields

$$\hat{K}_n = \sum_{j=o}^{n-1} \underset{\sim}{\Delta}_{j+1} \underset{\sim}{\delta}_j^T \left[\sum_{j=o}^{n-1} \underset{\sim}{\delta}_j \underset{\sim}{\delta}_j^T \right]^{-1} \tag{4.33}$$

and

$$\hat{K}_{n+1} = \sum_{j=o}^{n} \underset{\sim}{\Delta}_{j+1} \underset{\sim}{\delta}_j^T \left[\sum_{j=o}^{n} \underset{\sim}{\delta}_j \underset{\sim}{\delta}_j^T \right]^{-1} \tag{4.34}$$

It is now necessary to put Eq. (4.34) in recursive form, which also eliminates the matrix inversion. Using Eq. (4.33), Eq. (4.34) becomes

$$\hat{K}_{n+1} = \left[\hat{K}_n \sum_{j=o}^{n-1} \underset{\sim}{\delta}_j \underset{\sim}{\delta}_j^T + \underset{\sim}{\Delta}_{n+1} \underset{\sim}{\delta}_n^T \right] \left[\sum_{j=o}^{n-1} \underset{\sim}{\delta}_j \underset{\sim}{\delta}_j^T + \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T \right]^{-1} \tag{4.35}$$

By defining W_n^{-1} as

$$W_n^{-1} \triangleq \frac{1}{n} \sum_{j=o}^{n-1} \underset{\sim}{\delta}_j \underset{\sim}{\delta}_j^T \tag{4.36}$$

then W_{n+1}^{-1} is determined by

$$W_{n+1}^{-1} = (1 - \frac{1}{n+1}) W_n^{-1} + \frac{1}{n+1} \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T$$

$$= \frac{n}{n+1} W_n^{-1} + \frac{1}{n+1} \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T$$

$$= \frac{1}{n+1} (nW_n^{-1} + \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T) \tag{4.37}$$

or equivalently

$$W_{n+1} = (n+1)(nW_n^{-1} + \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T)^{-1} \tag{4.38}$$

Using the well-known "inside-out" lemma (see e.g., Deutsch, 1965, p. 119) of numerical analysis, Eq. (4.38) may be rewritten

$$W_{n+1} = \frac{n+1}{n} \{W_n - W_n \underset{\sim}{\delta}_n [n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n]^{-1} \underset{\sim}{\delta}_n^T W_n\} \tag{4.39}$$

Now substitute Eq. (4.36) into Eq. (4.35). The result is

$$\hat{K}_{n+1} = [\hat{K}_n \; nW_n^{-1} + \underset{\sim}{\Delta}_{n+1} \underset{\sim}{\delta}_n^T](nW_n^{-1} + \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n)^{-1}$$

which can be rewritten using the "inside-out" lemma as

$$\hat{K}_{n+1} = [\hat{K}_n \; nW_n^{-1} + \underset{\sim}{\Delta}_{n+1} \underset{\sim}{\delta}_n^T] \; W_n - \left[\frac{W_n \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T W_n}{(n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)}\right] \frac{1}{n} \tag{4.40}$$

Performing the indicated multiplication and observing that $\underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n$ is a scalar reduces Eq. (4.40) to

$$\hat{K}_{n+1} = \hat{K}_n + \frac{\underset{\sim}{\Delta}_{n+1}\underset{\sim}{\delta}_n^T W_n - \hat{K}_n \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T W_n}{(n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)}$$

$$= \hat{K}_n + [\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n] \frac{\underset{\sim}{\delta}_n^T W_n}{(n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)} \qquad (4.41)$$

Equation (4.41) is recursive and the matrix inversion has been avoided. To put this algorithm in the form of Eq. (4.20) it is necessary to prove that

$$\frac{\underset{\sim}{\delta}_n^T W_{n+1}}{n+1} = \frac{\underset{\sim}{\delta}_n^T W_n}{(n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)}$$

Pre-multiplying Eq. (4.37) by W_n and post-multiplying by W_{n+1} gives

$$W_n = \frac{1}{n+1} (nI + W_n \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T) W_{n+1}$$

Again pre-multiplying by $\underset{\sim}{\delta}_n^T$

$$\underset{\sim}{\delta}_n^T W_n = \frac{1}{n+1} (n \underset{\sim}{\delta}_n^T + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n \underset{\sim}{\delta}_n^T) W_{n+1}$$

$$= \frac{1}{n+1} (n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n) \underset{\sim}{\delta}_n^T W_{n+1}$$

or

$$\frac{\underset{\sim}{\delta}_n^T W_{n+1}}{n+1} = \frac{\underset{\sim}{\delta}_n^T W_n}{(n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)} \qquad (4.42)$$

Substituting this equation into the estimation scheme of Eq. (4.41) gives the desired stochastic relation

$$\hat{K}_{n+1} = \hat{K}_n + [\underset{\sim}{\Delta}_{n+1} - \hat{K}_n \underset{\sim}{\delta}_n] \, \underset{\sim}{\delta}_n^T \, \frac{W_{n+1}}{n+1} \tag{4.43}$$

where

$$W_{n+1} = \frac{n+1}{n} [W_n - W_n \underset{\sim}{\delta}_n (n + \underset{\sim}{\delta}_n^T W_n \underset{\sim}{\delta}_n)^{-1} \underset{\sim}{\delta}_n^T W_n] \tag{4.39}$$

This stochastic approximation algorithm for learning the Kalman filter matrix is an optimal one in the sense that

(a) it utilizes the information from all past estimation stages rather than one stage,

(b) it minimizes the intermediate loss function $\ell_n(K_n)$ at each stage of iteration instead of just following the local gradient at each stage, and

(c) it does not affect the asymptotic convergence of $\hat{K}_n$ to K_{opt}.

In addition, by invoking the "strong law of large numbers" it can be shown that $\ell_n(K)$ converges to $L(K)$ with probability one [Ho and Blaydon, 1966], where $L(K)$ is given by Eq. (4.22).

The increase in statistical efficiency attained when using the weight matrix defined in Eq. (4.39) is paid for in computational complexity. If W_{n+1} is selected to be a constant for all n, only one iteration is required per estimation stage, as opposed to two when W_{n+1} is specified by Eq. (4.39). It is worth noting that the asymptotic behavior of both algorithms is governed by the "law of large numbers",

i.e., the convergence rate for large n is proportional to 1/n. However, from an engineering viewpoint the adaptive properties of the algorithm for small and intermediate values of n is crucial. In this region the accelerated version is far superior to the non-accelerated one, as is demonstrated experimentally in Chapter 6.

4.7 Summary

In this Chapter adaptive stochastic algorithms were derived for learning the optimal Kalman filter. The theory of stochastic approximation was briefly reviewed. It was employed in conjunction with the Learning Criterion of Chapter 3 to prove probability one convergence of the algorithms. A scheme for accelerating algorithmic convergence was presented. The final results of this acceleration technique are given by Eqns. (4.39) and (4.43). These are the two algorithms used to obtain the principal simulation results of Chapter 6.

However, before the experimental data are presented, the results obtained in this chapter are used in Chapter 5 to estimate other useful quantities such as the error covariance Γ, the predicted error covariance Σ, as well as Q and R. These estimates are needed to adaptively solve the optimal smoothing problem.

CHAPTER 5

ADDITIONS AND EXTENSIONS

5.1 Introduction and Organization of the Chapter

In Chapter 4 adaptive algorithms are derived for solving the optimal filtering. Adaptive prediction is included in this solution as a necessary condition. For optimal smoothing, however, knowledge of the optimum filter must be augmented by additional information. Specifically, the fixed-interval smoothing also requires the one-step error covariance matrix Σ (see Eq. (2.31)); fixed point smoothing requires R^{-1} as well as K_{opt} and Σ (see Eq. (2.33)); fixed-lag smoothing requires K_{opt}, Σ, and Q (see Eqns. (2.35) and (2.36)). In addition, if the error covariance matrix Γ is known, the formulas for optimal smoothing may be significantly simplified. Thus, Section 5.2 uses the results of Chapter 4 to estimate Σ, Γ, R, and Q. Techniques for lifting the rather restrictive assumption that the inverse of H exists are introduced in Section 5.3. Section 5.4 suggests simple modification to the algorithm of Chapter 4 to accommodate slowly time-varying signal and noise statistics. These same algorithms are heuristically altered in Section 5.5 a certain class of nonlinear systems. A summary of the chapter is contained in Section 5.6.

5.2 Estimation of Σ, Γ, R and Q

After the optimal filter has been learned, estimation schemes for Σ, Γ, R, and Q can be derived. To accomplish this, additional use

must be made of the information contained in the residual time series $\{\underset{\sim}{\delta}_n\}$. The algorithm for estimating Σ is considered first. By performing the expectation indicated in Eq. (3.10), as in Corollary 3-4, it is seen that[1]

$$\Omega(1) = E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n^T\} \tag{3.10}$$

$$= H\Phi\{\Sigma H^T - K[H\Sigma H^T + R]\} \tag{5.1}$$

Similarly, $\Omega(0)$ can be shown to be

$$\Omega(0) = E\{\underset{\sim}{\delta}_n\underset{\sim}{\delta}_n^T\}$$

$$= H\Sigma H^T + R \tag{5.2}$$

Solving Eq. (5.1) for Σ produces

$$\Sigma = (H\Phi)^{-1}\Omega(1)(H^T)^{-1} + K[H\Sigma H^T + R](H^T)^{-1} \tag{5.3}$$

Substituting for $[H\Sigma H^T + R]$ by using Eq. (5.2) gives

$$\Sigma = [(H\Phi)^{-1}\ \Omega(1) + K\Omega(0)](H^T)^{-1} \tag{5.4}$$

As noted earlier $\Omega(1)$ and $\Omega(0)$ cannot actually be calculated since the statistics of the distribution functions are unknown. However, they may be estimated recursively using standard stochastic approximation methods, which yield

---

1. The notation $\Omega(n+1,n)$ and $\Omega(n,n)$ has been simplified to $\Omega(1)$ and $\Omega(0)$ respectively denoting the difference between the time arguments.

$$\hat{\Omega}_n(1) = \hat{\Omega}_{n-1}(1) + a_n[\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n^T - \hat{\Omega}_{n-1}(1)] \tag{5.5}$$

$$\hat{\Omega}_n(0) = \hat{\Omega}_{n-1}(0) + b_n[\underset{\sim}{\delta}_n\underset{\sim}{\delta}_n^T - \hat{\Omega}_{n-1}(0)] \tag{5.6}$$

where $\{a_n\}$ and $\{b_n\}$ satisfy the conditions of Theorem 4.1. Using this theorem, Eqns. (5.5) and (5.6) can be shown to converge to (5.1) and (5.2) respectively. Substituting these estimates into (5.4) along with the estimate of K_{opt} provides the desired relation for estimating Σ.

$$\hat{\Sigma}_n = [(H\Phi)^{-1}\ \hat{\Omega}_n(1) + \hat{K}_n\ \hat{\Omega}_n(0)]\ (H^T)^{-1} \tag{5.7}$$

To derive an expression for estimating the error covariance matrix Γ, a general expression for Γ must be developed. The relation given by Eq. (2.21) is deficient because it is valid only for optimal filtering. By definition

$$\Gamma = E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)][\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T\}$$

Substituting the filtering equation for $\hat{\underset{\sim}{x}}(n)$ yields

$$\Gamma = E\{[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1) + K(\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1))]$$

$$[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1) + K(\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1))]^T\}$$

By exploiting the independence of $\underset{\sim}{v}(n)$ and $w(n)$, the above equation may be written

$$\Gamma = E\{[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)][\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)]^T\}$$

$$- E\{[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)][\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)]^T\} H^T K^T$$

$$- KH\ E\{[\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)][\underset{\sim}{x}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)]^T\}$$

$$+ K\ E\{[\underset{\sim}{z}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)][\underset{\sim}{z}(n) - \Phi\hat{\underset{\sim}{x}}(n-1)]^T\} K^T$$

$$= \Sigma - \Sigma\ H^T K^T - KH\Sigma + K[H\Sigma H^T + R]\ K^T$$

$$= [I - KH]\Sigma + \{K[H\Sigma H^T + R] - \Sigma H^T\}\ K^T \qquad (5.8)$$

When K is equal to the optimal filter as given by Eq. (2.20), the term in braces is zero, and Eq. (5.8) reduces to the optimum expression for the error covariance matrix. Observe from Eq. (5.1) that Eq. (5.8) may be rewritten

$$\Gamma = [I - KH]\ \Sigma - (H\Phi)^{-1}\ \Omega(1)\ K^T \qquad (5.9)$$

Replacing the unknown quantities with their respective estimates, the algorithm for $\hat{\Gamma}_n$ is obtained

$$\hat{\Gamma}_n = [I - \hat{K}_n H]\ \hat{\Sigma}_n - (H\Phi)^{-1}\ \hat{\Omega}_n(1)\ \hat{K}_n^T \qquad (5.10)$$

To estimate R, the estimator for Σ and $\Omega(0)$ is substituted into Eq. (5.2), giving

$$\hat{R}_n = \hat{\Omega}_n(0) - H\ \hat{\Sigma}_n\ H^T \qquad (5.11)$$

$$= \hat{K}_n^{-1}\ \hat{\Gamma}_n\ H^T \qquad (5.12)$$

Equation (5.11) is generally preferable to Eq. (5.12) for estimating R because it does not require the existence or calculation of the inverse of $\hat{K}_n$. If R^{-1} is required as it is in the smoothing Eq. (2.32), it may be obtained directly from Eq. (2.20)

$$R^{-1} = (H^T)^{-1} \Gamma^{-1} K$$

from which the estimate of R inverse is

$$\hat{R}_n^{-1} = (H^T)^{-1} \hat{\Gamma}_n^{-1} \hat{K}_n \tag{5.13}$$

This result does require the inverse of $\hat{\Gamma}_n$, but it is usually better conditioned than $\hat{R}_n$.

The last quantity to be estimated is Q. From Eq. (2.22)

$$Q = \Sigma - \Phi \Gamma \Phi^T \tag{5.14}$$

$$= \Sigma - \Phi[I - KH] \Sigma \Phi^T \tag{5.15}$$

Substituting the estimates of the unknown quantities into Eqns. (5.14) and (5.15) respectively results in

$$\hat{Q}_n = \hat{\Sigma}_n - \Phi \hat{\Gamma}_n \Phi^T \tag{5.16}$$

$$= \hat{\Sigma}_n - \Phi[I - \hat{K}_n H] \hat{\Sigma}_n \Phi^T \tag{5.17}$$

Equations (5.5), (5.6), (5.7), (5.10), (5.11), (5.12), (5.13), (5.16) and (5.17) provide a complete set of algorithms for estimating Σ, Γ, R^{-1}, R, and Q. Where more than one method is available for estimating the same quantity, the algorithm used is dictated by computational considerations. However, these algorithms may be significantly

simplified by noting from the Learning Criterion Theorem 3-3 that

$$\Omega(1) = E\{\underset{\sim}{\delta}_{n+1}\underset{\sim}{\delta}_n^T\}$$

approaches the null matrix as $\hat{K}_n$ approaches K_{opt}. Thus for large n, algorithm (5.7) reduces to

$$\hat{\Sigma}_n = \hat{K}_n \hat{\Omega}_n(0)(H^T)^{-1} \tag{5.18}$$

This remains a close approximation for small and intermediate values of n if the accelerated algorithm defined by Eqns. (4.43) and (4.37) for estimating K_{opt} is employed. Another benefit accrued by using Eq. (4.43) and (4.37) is made obvious by rewriting it as a stochastic approximation

$$W_{n+1}^{-1} = W_n^{-1} + \frac{1}{n+1}(\underset{\sim}{\delta}_n\underset{\sim}{\delta}_n^T - W_n^{-1}) \tag{5.19}$$

Comparing this equation with Eq. (5.6) reveals that W_n^{-1} is just $\hat{\Omega}_n(0)$. Therefore, algorithm (5.18) is simply

$$\hat{\Sigma}_n = \hat{K}_n W_n^{-1}(H^T)^{-1} \tag{5.20}$$

Thus, the algorithm has not only been greatly simplified, but the computation of the two recursions Eqns. (5.5) and (5.6) have been eliminated. The algorithm (5.10) for estimating Γ becomes

$$\hat{\Gamma}_n = [I - \hat{K}_n H]\ \hat{\Sigma}_n \tag{5.21}$$

The expression for estimating R and Q remain unchanged because they were written in terms of $\hat{\Gamma}_n$ and $\hat{\Sigma}_n$.

5.3 Non-Existence of H Inverse

When the inverse of the observation matrix H does not exist, the stochastic algorithms derived in Chapter 4 for learning the optimum filter must be modified or a new invertible observation matrix must be created. Two methods for forming a new observation matrix and one method for modifying the algorithms are presented. They are

1. measurement stacking with time delay,
2. measurement stacking with state augmentation, and
3. residual stacking.

For purposes of discussion the following typical third order, single-input, single-output system is used. In phase variable representation, the system is described by Eqns. (1.3) and (1.4)

$$\underset{\sim}{x}(n+1) = \Phi\underset{\sim}{x}(n) + \underset{\sim}{w}(n) \tag{1.3}$$

$$\underset{\sim}{z}(n) = H\underset{\sim}{x}(n) + \underset{\sim}{v}(n) \tag{1.4}$$

where

$$\Phi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{bmatrix}$$

$$H = [1 \quad 0 \quad 0]$$

and $$\underset{\sim}{w}^T(n) = [0 \quad 0 \quad w(n)]$$

The results to be presented are valid for general Φ and H matrices.

5.3.1 Measurement Stacking with Time Delay

This is a straight forward, brute-force technique. It consists simply of accumulating m observations where m is the order of the system, and creating an equivalent m x m observation matrix, say H_N, whose inverse does exist.

For illustration, consider the example third order system. Since H is 1 x 3, three measurements must be stacked at every iteration. Making use of the deterministic relationships between the state variables and starting at time n, the measurements are

$$z(n) = x_1(n) + v(n)$$

$$z(n+1) = x_1(n+1) + v(n+1)$$

$$= x_2(n) + v(n+1)$$

$$z(n+2) = x_1(n+2) + v(n+2)$$

$$= x_3(n) + v(n+2)$$

Therefore,

$$\underset{\sim}{z}(n) = H_N \underset{\sim}{x}(n) + \underset{\sim}{v}(n)$$

where $H_N = I$

and $\underset{\sim}{v}^T(n) = [v(n), v(n+1), v(n+2)]$

Note that to avoid violating the assumption that the measurement noise $\underset{\sim}{v}(n)$ be uncorrelated from stage to stage, three new observations must be made at each iteration. This means that the filtering estimate of $\underset{\sim}{x}(n)$ is being performed at time n+2. To obtain the estimate of $\underset{\sim}{x}(n+2)$, $\hat{\underset{\sim}{x}}(n)$ must be predicted ahead two stages. This is the disadvantage of the technique. However, it is not as deleterious as it appears, because

the message generating noise directly affects only the prediction of $x_3(n)$. Still this method is not recommended unless the number of measurements required to form H_N is small compared to m.

5.3.2 Measurement Stacking With State Augmentation

The prediction problem associated with the previous technique can be avoided by using the standard technique of augmenting the state vector $\underset{\sim}{x}(n)$ with the measurement noise $\underset{\sim}{v}(n)$ [Meditch, 1969, p. 195]. This forms a new m x p state vector $\underset{\sim}{y}(n)$ such that

$$\underset{\sim}{y}(n) = \begin{bmatrix} \underset{\sim}{x}(n) \\ \underset{\sim}{v}(n) \end{bmatrix}$$

$$\underset{\sim}{y}(n+1) = \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} \underset{\sim}{y}(n) + \begin{bmatrix} \underset{\sim}{w}(n) \\ \underset{\sim}{v}(n+1) \end{bmatrix}$$

$$= \Phi_N \underset{\sim}{y}(n) + \begin{bmatrix} \underset{\sim}{w}(n) \\ \underset{\sim}{v}(n+1) \end{bmatrix} \qquad (5.22)$$

where the 0's are null matrices of appropriate dimension, and

$$\underset{\sim}{z}(n) = [H, I]\, \underset{\sim}{y}(n)$$

$$= H\underset{\sim}{x}(n) + \underset{\sim}{v}(n) \qquad (5.23)$$

This augmented structure requires only one new measurement per estimation stage regardless of the H matrix. The new (m+p) x (m+p) invertible observation matrix H_N is formed by stacking the new measurement with predicted measurements from the previous iteration.

This technique is made clearer by considering the example system given by Eqns. (5.22) and (5.23). The four required measurements are

$$\begin{aligned} z(n) &= [H,I]\ \underset{\sim}{y}(n) \\ &= [1,0,0,1]\ \underset{\sim}{y}(n) \\ &= x_1(n) + v(n) \end{aligned} \tag{5.24}$$

$$\begin{aligned} z(n)_1 &= [H\Phi_N,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= [1,0,0,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= x_1(n) + e_1(n) \end{aligned} \tag{5.25}$$

$$\begin{aligned} z(n)_2 &= [H\Phi_N\Phi_N,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= [0,1,0,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= x_2(n) + e_2(n) \end{aligned} \tag{5.26}$$

$$\begin{aligned} z(n)_3 &= [H\Phi_N\Phi_N\Phi_N,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= [0,0,1,0]\ \hat{\underset{\sim}{y}}(n-1) \\ &= x_3(n) + e_3(n) \end{aligned} \tag{5.27}$$

where $e_i(n)$, $i=1,2,3$ is the difference between the value of $x_i(n)$ and $\hat{x}_i(n|n-1)$. From Eqns. (5.24) - (5.27), it is obvious that H_N is

$$H_N = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

This technique is recommended when the dimension of $\underset{\sim}{z}(n)$ is small compared to the dimension of $\underset{\sim}{x}(n)$. Note that both techniques for

forming an invertible observation matrix inherently change the original Kalman filter structure. The standard Kalman filter is an m x p matrix, where p is the dimension of the $\underset{\sim}{z}(n)$ and satisfies the inequality

$$1 \leq p \leq m$$

The new observation matrix H_N produces an m dimensional measurement. Thus, the new Kalman filter is always an m x m matrix. A technique which preserves the original structure of the problem is presented in Section 5.3.3.

5.3.3 Predicted Residual Stacking

The development of this section is based on the common case where H is an 1 x m matrix, i.e., only the output state variable can be observed. This implies that the Kalman filter is an m x 1 vector. A similar development can be performed for any given H matrix. Here the predicted residual is defined by

$$z(n+r) - H\Phi^{r+1} \hat{\underset{\sim}{x}}(n-1) \triangleq \Delta_{n+r}$$

Now let

$$\underset{\sim}{\Delta}(n+m,n) \triangleq \begin{bmatrix} H\Phi^m \\ \vdots \\ H\Phi^2 \\ H\Phi \end{bmatrix}^{-1} \begin{bmatrix} \Delta_{n+m} \\ \vdots \\ \Delta_{n+2} \\ \Delta_{n+1} \end{bmatrix}$$

The loss function of Eq. (4.18) is now redefined as

$$\ell(\underset{\sim}{\Delta}(n+m,n) - \underset{\sim}{\Delta}(n+m,n))$$

$$= [\underset{\sim}{\Delta}(n+m,n) - \underset{\sim n}{K}\delta_n]^T[\underset{\sim}{\Delta}(n+m,n) - \underset{\sim n}{K}\delta_n] \qquad (5.29)$$

where δ_n is as defined in Chapter 3. To see that Eq. (5.29) is a meaningful definition for the loss function, form the linear regression equation

$$\underset{\sim}{\Delta}(n+m,n) = \underset{\sim n}{K}\delta_n + \underset{\sim}{e}_n \qquad (5.30)$$

Now the prediction residual transition vector is

$$\underset{\sim n}{K} = E\{\underset{\sim}{\Delta}(n+m,n)\ \delta_n^T\}(E\{\delta_n\delta_n^T\})^{-1} \qquad (5.31)$$

After some computation, it is found that

$$E\{\underset{\sim}{\Delta}(n+m,n)\ \delta_n^T\} = \Sigma H^T$$

and

$$E\{\delta_n\delta_n^T\} = (H\Sigma H^T + R)$$

Therefore,

$$\underset{\sim n}{K} = \Sigma H^T(H\Sigma H^T + R)^{-1}$$

$$= \underset{\sim opt}{K}$$

Consequently, $\underset{\sim n}{K}$ represents the optimal Kalman filter matrix. Also, $\underset{\sim n\sim n}{K}\delta$ may be considered to represent the predicted value of $\underset{\sim}{\Delta}(m+m,n)$. Since $\underset{\sim n}{K}$ cannot be calculated, $\underset{\sim opt}{K}$ is estimated as in Chapter 4 using the theory of stochastic approximation. Applying this theory to the loss function of Eq. (5.29) gives

$$\hat{K}_{n+1} = \hat{K}_{n} - a_{n+1} \nabla_{K} \ell(\Delta(n+m,m) - \hat{K}_{n} \delta_{n}) \; W_{n+1}$$

$$= \hat{K}_{n} + a_{n+1} [\Delta(n+m,n) - \hat{K}_{n} \delta_{n}] \; \delta_{n}^{T} W_{n+1} \qquad (5.32)$$

where $a_n = 1/n$ and W_{n+1} is a uniformly bounded sequence. The algorithm of Eq. (5.32) is analogous to Eq. (4.20) and can be shown to converge to K_{opt} with probability one. In general the method of this section is recommended over those of Section 5.3.1 and 5.3.2. However, the author conjectures that the mean-square error associated with the state estimation procedure of 5.3.2 is at least as small as that of this section.

5.4 Time-Varying Signal and Noise Statistics

In practice the stationary assumption of Chapters 3 and 4 is valid only for a finite time interval. If this time interval is less than the essential adaptation time of the learning algorithms, then the results of Chapter 4 may still be employed. Even in this case, it is important to determine when the time variations have changed the optimal filter significantly.

A statistical test for making such a determination can be devised from either Theorem 3-2 or Theorem 3-3 (the Learning Criterion). Theorem 3-2 states that $E\{\delta_{n+1}^{T} \delta_{n}\} = 0$ if and only if the filter is optimum, and Theorem 3-3 states that $E\{\delta_{n+1} \delta_{n}^{T}\} = [0]$ if and only if the filter is optimum. Either of these two facts may be used to develop a confidence interval test for the time average of $\delta_{n+1}^{T} \delta_{n}$ or $\delta_{n+1} \delta_{n}^{T}$. Alternately, the chi-square method can be used to test whether or not the times average of $\delta_{n+1}^{T} \delta_{n}$ or $\delta_{n+1} \delta_{n}^{T}$ is significantly different from

zero or the null matrix respectively. Since these techniques are standard in the statistical literature, the details are not presented here.

Returning now to the problem of a time-varying environment, two different approaches are considered. The first is useful when the time variation is slow and its exact nature is unknown. The second is valid only when the time variation of the filter is known.

5.4.1 Measurement Discounting

The algorithms of Chapter 4 weight all measurements equally over the adaptation interval. Thus as the interval increases, or equivalently as the number of iterations n increases, the effect of each measurement becomes vanishingly small. This effect may be eliminated by attenuating past observations. Such a procedure essentially shapes the memory of the algorithm. One simple way of limiting the memory is to weight the data equally over a finite number of observations under the assumption that the process is stationary over this time interval.

A different approach is to exponentially attenuate past measurements [Korn, 1966, p. 45]. To modify the accelerated algorithm of Eqns. (4.39) and (4.43) for an exponentially weighted memory requires that the intermediate loss function of Eq. (4.31) be redefined as

$$L_n(K_n) = \frac{1}{2n} \sum_{j=o}^{n-1} (\underset{\sim}{\Delta}_{j+1} - K_n \underset{\sim}{\delta}_j)^T \lambda^{n-1-j} (\underset{\sim}{\Delta}_{j+1} - K_n \underset{\sim}{\delta}_j) \tag{5.33}$$

where $\lambda \in (0,1)$.

Now following the development which led to Eqns. (4.39) and (4.43), the recursive relation for learning K_{opt} is found to be

$$\hat{K}_{n+1} = \hat{K}_n + [\Delta_{n+1} - \hat{K}_n \delta_n] \, \delta_n^T \, \frac{W_{n+1}}{n+1} \tag{5.34}$$

and

$$W_{n+1} = \frac{n+1}{n\lambda} \{W_n - W_n \underset{\sim}{\delta}_n \, [n\lambda + \delta_n^T \, W_n \underset{\sim}{\delta}_n]^{-1} \, \underset{\sim}{\delta}_n^T \, W_n\} \tag{5.35}$$

The term λ which provides the exponential damping also determines the effective sample size of the estimation process. Both the ability of the algorithm to refine the estimates and to follow time variations are determined by λ. Unfortunately the two requirements are conflicting. The first implies a large value (near unity) of λ, and the second implies a small value (near zero) of λ. Thus, a compromise is necessary in selecting λ. Usually λ must be greater than 0.95 to ensure stability.

When the time evolution of $\underset{\sim}{K}(n)$ is known with Q(n) and R(n) unknown, the procedure of Section 5.4.2 is recommended.

5.4.2 Known Time Evolution of K(n)

Let $\underset{\sim}{K}(n)$ evolve in time in a manner described by Eq. (5.36)

$$\underset{\sim}{K}(n+1) = F[\underset{\sim}{K}(n)] + \underset{\sim}{\omega}(n) \tag{5.36}$$

where F is the m x m filter transition matrix and $\underset{\sim}{\omega}(n)$ is an m x 1 random perturbation vector. In this case Eq. (4.39) is modified in a manner that is reminiscent of Kalman's filtering equation. This modification yields

$$\hat{\underset{\sim}{K}}_{n+1} = F(\hat{\underset{\sim}{K}}_n) + [\underset{\sim}{\Delta}_{n+1} - F(\hat{\underset{\sim}{K}}_n)\ \underset{\sim}{\delta}_n]\ \underset{\sim}{\delta}_n^T\ \frac{W_{n+1}}{n+1} \tag{5.37}$$

5.5 Nonlinear Dynamics

When the plant dynamics are nonlinear, the filter that is optimal in the sense that it minimizes the mean square error is unknown. In this section a different definition for optimality of the filter is given and adaptive algorithms which learn this optimal filter are presented. The Learning Criterion as given by Eq. (3.6)

$$E\{\underset{\sim}{\delta}_{j+1}\underset{\sim}{\delta}_j^T\} = [0] \qquad \text{for all } j \leq n \tag{3.6}$$

of Theorem 3-3 is used as the definition for optimal filtering. The definition is selected because it is equivalent to optimal Kalman filtering when the dynamics are linear and because it forces the residual time series to be white. This latter fact implies that all the information possible has been extracted from this series. The system model considered is given by the equations

$$x(n+1) = f(x_n) + w(n) \tag{5.38}$$

$$z(n) = x(n) + v(n) \tag{5.39}$$

and the form of the filtering equation is specified to be

$$\hat{x}(n+1) = f(\hat{x}(n)) + K_n\ [z(n+1) - f(\hat{x}(n))] \tag{5.40}$$

Note that each of these equations is the same as their linear counterparts with the linear dynamics replaced with the nonlinear dynamics.

Equation (5.38) corresponds to a stochastic signal with a nonrational spectral density.

From Eq. (5.40) it is obvious that the residual time series is

$$\delta_n = z(n+1) - f(\hat{x}(n)) \tag{5.41}$$

Thus from the definition of optimal filtering given by Eq. (3.6), the filter is optimum if and only if it is chosen such that

$$E\{[z(j+1)-f(\hat{x}(j))][z(j) - f(\hat{x}(j-1))]\} = 0 \quad \text{for all } j \leq n \tag{5.42}$$

Two different classes of nonlinearities are now considered. The first type satisfies a global Lipschitz condition

$$|f(x_n) - f(x_m)| \leq \phi |x_n - x_m| \quad \text{for all } x_n, x_m$$

For this class of nonlinearities the adaptive algorithm for learning the filter that satisfies Eq. (5.42) is given by Eq. (4.21) where H = 1 and Φ is the Lipschitz constant.

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n} \phi^{-1} \delta_{n+1} \delta_n W_{n+1} \tag{5.43}$$

The second type of nonlinearity has an inverse. The dynamics $f(x_n)$ must be such that

$$f^{-1}[f(x_n)] = x_n \quad \text{for all } x_n \tag{5.44}$$

For this case the stochastic approximation algorithm is heuristically modified as follows

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n} f^{-1}(\delta_{n+1}\delta_n)\, W_{n+1} \qquad (5.45)$$

The engineering justification for the nonlinear algorithms of Eqns. (5.43) and (5.45) is that they "work". That is, they allow one to make "useful" predictions. Experimental confirmation of this statement is presented in the next chapter.

5.6 Summary

This chapter has derived algorithms for estimating useful quantities such as Σ, Γ, R and Q. This knowledge is necessary to solve the optimal smoothing problem. The restrictive condition of Chapter 4 requiring that H^{-1} exist has been lifted. The algorithms of Chapter 4 were then heuristically modified to accommodate special classes of time-varying and nonlinear dynamics. This chapter also concludes the theoretical aspects of the report. The next chapter contains a presentation of the experimental results of this study.

CHAPTER 6

NUMERICAL SIMULATION OF THE ADAPTIVE ALGORITHMS

6.1 Introduction and Organization of the Chapter

In Chapter 4 adaptive stochastic algorithms were derived for directly learning the optimal Kalman filter K_{opt}. Chapter 5 extended these results to include estimation of Σ, Γ, Q and R. These quantities are required for adaptive optimal smoothing. The restrictive assumption that H^{-1} exist was also lifted and the algorithms of Chapter 4 were modified to accommodate slowly time-varying systems statistics and certain classes of nonlinear dynamics.

In this chapter the adaptive algorithms of Chapter 4 and Sections 5.2 and 5.5 are applied to specific systems. The experimental results were obtained by simulating the different systems and the associated algorithms on a CDC 6400 computer. In interpreting these simulations, it is emphasized that double precision arithmetic was not used.

The first part of the chapter considers linear systems. Section 6.2 presents the experimental data for first order through fourth order dynamic systems. Nonlinear plants are discussed in Section 6.3. Section 6.4 applies the theory to the problem of estimating the states of a thermionic reactor.

6.2 Linear System Simulations

For convenience the dynamic system equations and adaptive algorithms to be studied are repeated here. The system dynamics and observations are defined by

$$\underset{\sim}{x}(n+1) = \Phi\underset{\sim}{x}(n) + \underset{\sim}{w}(n) \tag{1.3}$$

$$\underset{\sim}{z}(n) = H\underset{\sim}{x}(n) + \underset{\sim}{v}(n) \tag{1.4}$$

The filtering equation is

$$\hat{\underset{\sim}{x}}(n) = \Phi\hat{\underset{\sim}{x}}(n-1) + K[\underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)] \tag{2.19}$$

where the estimate $\hat{\underset{\sim}{x}}(n)$ is optimal if and only if K is determined according to Eq. (2.20). The non-accelerated adaptive algorithm for learning the optimal filter K_{opt} when R and Q are unknown is given by

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n+1}\,[\underset{\sim n+1}{\Delta} - \hat{K}_n \underset{\sim n}{\delta}]\ \underset{\sim n}{\delta}^T \tag{4.28}$$

where

$$\underset{\sim n+1}{\Delta} = (H\Phi)^{-1}[\underset{\sim}{z}(n+1) - H\Phi\Phi\hat{\underset{\sim}{x}}(n-1)]$$

and

$$\underset{\sim n}{\delta} = \underset{\sim}{z}(n) - H\Phi\hat{\underset{\sim}{x}}(n-1)$$

The accelerated version of Eq. (4.28) is

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n+1}\,[\underset{\sim n+1}{\Delta} - \hat{K}_n \underset{\sim n}{\delta}]\ \underset{\sim n}{\delta}^T W_{n+1} \tag{4.43}$$

where

$$W_{n+1} = \frac{n+1}{n}\,[W_n + W_n \underset{\sim n}{\delta}\ (n + \underset{\sim n}{\delta}^T W_n \underset{\sim n}{\delta})^{-1}\ \underset{\sim n}{\delta}^T W_n] \tag{4.39}$$

The equations for estimating Σ, Γ, R and Q are

$$\hat{\Sigma}_n = \hat{K}_n W_n^{-1} (H^T)^{-1} \tag{5.19}$$

$$\hat{\Gamma}_n = [I - \hat{K}_n H]\ \hat{\Sigma}_n \tag{5.20}$$

$$\hat{R}_n = W_n^{-1} - H\ \hat{\Sigma}_n\ H^T \tag{5.21}$$

$$\hat{Q}_n = \hat{\Sigma}_n - \Phi \hat{\Gamma}_n\ \Phi^T \tag{5.16}$$

A listing of the computer program developed to perform these calculations may be found in the Appendix. Unless it is specified otherwise, both the plant perturbation noise $\underset{\sim}{v}(n)$ and observation noise $\underset{\sim}{w}(n)$ have Gaussian distributions.

6.2.1 First Order Dynamics

In this case Eqns. (2.20) and (2.21) can be solved explicitly for K_{opt}

$$K_{opt} = \frac{-(1 + Q/R - \Phi^2)}{2\Phi^2} + \left\{ \frac{(1 + Q/R - \Phi^2)^2}{4\Phi^4} + \frac{Q/R}{\Phi^2} \right\}^{1/2} \tag{6.1}$$

EXAMPLE 6.1:

Given: $\Phi = 0.5$, $H = 1.0$, $Q = R = 1.0$

For these parameters $K_{opt} = 0.53$. To begin the adaptive process, an initial value K_o for the filter must be selected. In the scalar case $K_{opt} \in [0,1]$. Choosing $K_o = 0.0$ corresponds to the one extreme of assuming the measurements are just noise, i.e., they contain no information. For this case the experimental results for both the non-accelerated

adaptive algorithms are plotted in Fig. 6.1. The solid line represents the optimum value of the filter. The filter initial condition of zero is denoted by a small circle on the 0 iteration line. The accelerated algorithm has converged to within 10% of its optimal value in 80 iterations. The non-accelerated algorithm requires 500 iterations to attain this accuracy. Both algorithms converge asymptotically to K_{opt}. This serves to verify experimentally the observations made in Chapter 4.

From Eqns. (5.16), (5.19), (5.20), and (5.21), Σ, Γ, R, and Q may be estimated. Using these quantities, estimates of M, S, U, and J, which are required for optimal smoothing, may be obtained. These estimates after 1000 iterations of algorithm (4.43) are compared with their optimum values in Table 6.1.

TABLE 6.1

COMPARISON OF THE ESTIMATES AND THE OPTIMAL VALUES FOR EXAMPLE 6.1

Quantity	K	Σ	Γ	R	Q	M	S	U	J, N=1
Optimal Value	.531	1.13	.531	1.0	1.0	.235	.1250	1.888	0.235
Estimate	.528	1.15	.542	1.04	.994	.236	.1245	1.834	0.236

To illustrate the insensitivity of the adaptive algorithms to the initial guess for the filter, the convergence properties are examined for $K_o = 1.0$. This corresponds to the other extreme of assuming the measurements are perfect, i.e., they contain no noise. The experimental results for the accelerated and non-accelerated algorithms are shown in Fig. 6.2. Again the accelerated algorithm converges much more rapidly.

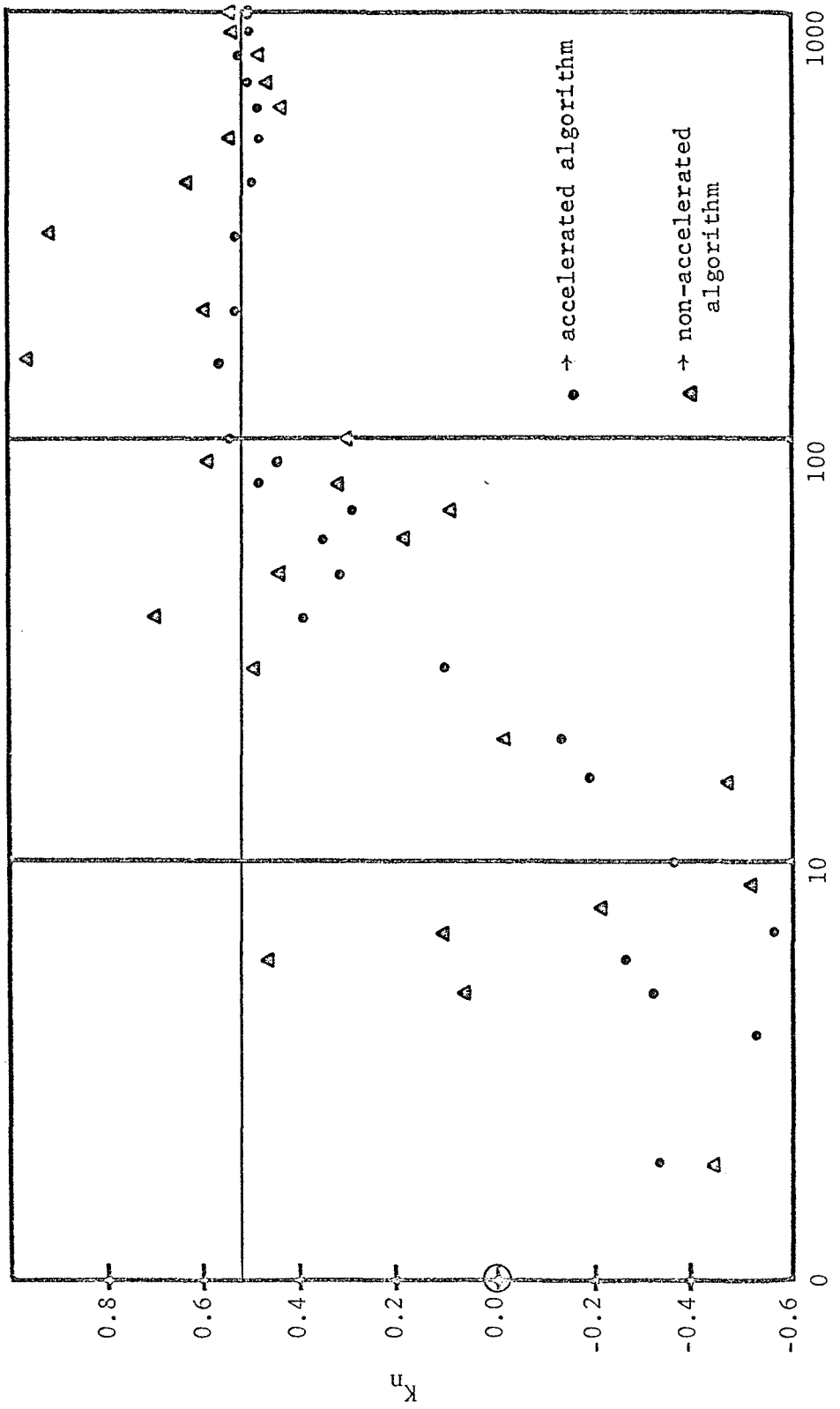


Fig. 6.1 Time Evolution of K_n for Example 6.1: $K_o = 0.0$

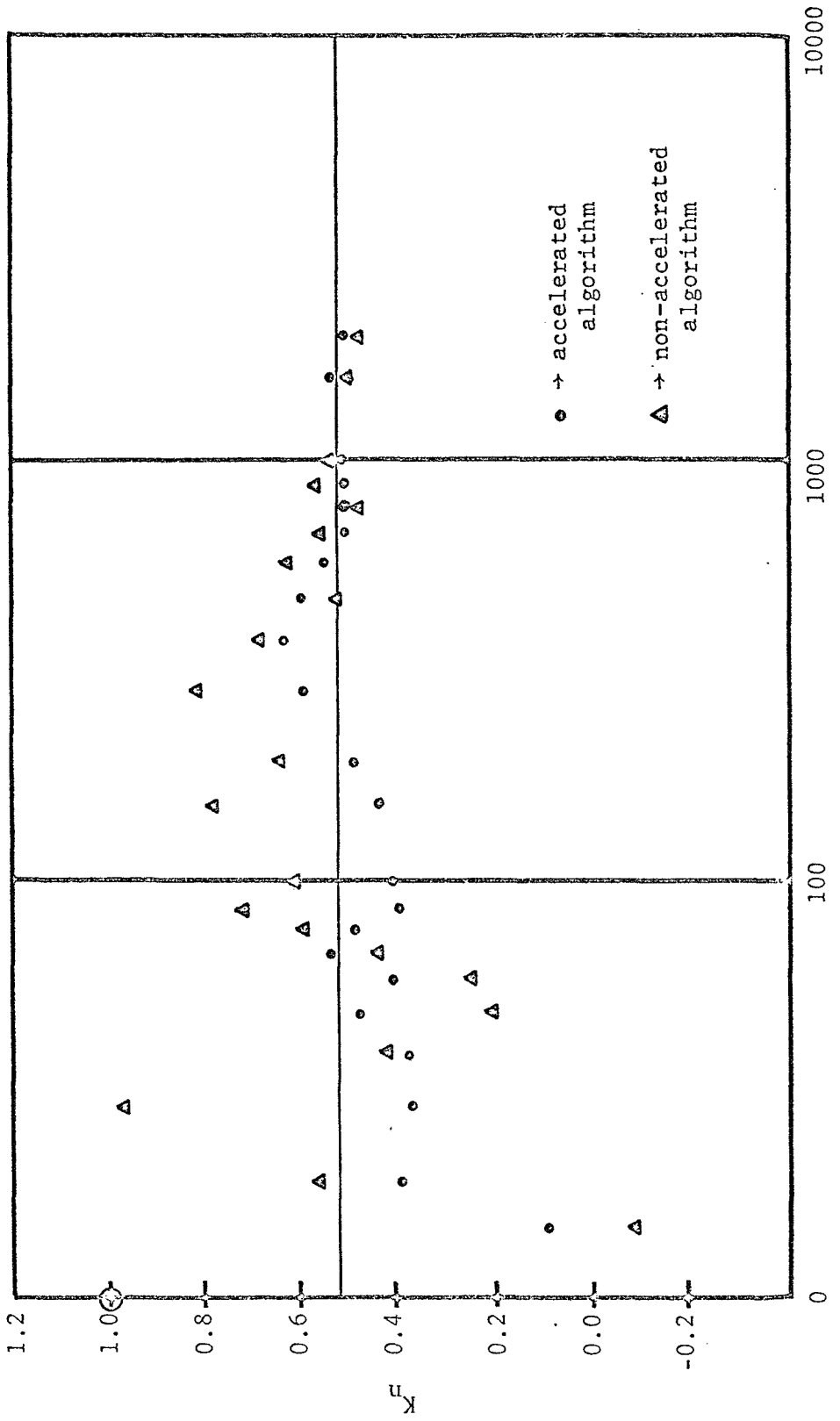


Fig. 6.2 Time Evolution of K_n for Example 6.1: $K_o = 1.0$

EXAMPLE 6.2:

Given: $\Phi = 0.5$, $H = 1.0$, $Q = 1.0$, $R = 4Q$

The purpose of this example is to demonstrate the power of the self-adaptive algorithm of Eqns. (4.43) and (4.39) to learn the optimum filter in a very noisy environment. From a communication theory viewpoint, Eq. (1.3) is simply the signal generating process and $\underset{\sim}{x}(n)$ is the signal. The signal to noise ratio for this problem is given by the ratio

$$\frac{\phi_x(o)}{\phi_v(o)}$$

where $\phi_x(\tau)$ denotes the autocorrelation function of x. $\phi_x(o)$ is

$$\phi_x(o) = \frac{\phi_w(o)}{(1-\Phi^2)} \qquad (6.2)$$

$$= \frac{Q}{(1-\Phi^2)}$$

Therefore, the signal to noise ratio is

$$\frac{\phi_x(o)}{\phi_v(o)} = \frac{Q}{(1-\Phi^2)R} \qquad (6.3)$$

For the parameters of this example, Eq. (6.3) has the numerical value of

$$\frac{\phi_x(o)}{\phi_v(o)} = \frac{1}{3}$$

or

$$\left.\frac{\phi_x(o)}{\phi_v(o)}\right|_{db} = -9.55 \text{ db}.$$

A signal to noise ratio of -9.55 db is extremely low. Still, the adaptive algorithm learns the optimal filter quite well. The results after 1,000 iterations are summarized in Table 6.2. The initial value of the filter was 1.0. Similar results were obtained for $K_o = 0.0$.

TABLE 6.2

COMPARISON OF THE ESTIMATES AND THE OPTIMAL VALUES FOR EXAMPLE 6.2

Quantity	K	Σ	Γ	R	Q
Optimal Value	.236	1.24	.944	4.00	1.00
Estimate	.240	1.29	.961	4.08	1.04

Other first order problems were considered with Φ being varied from 0.2 to 1.0 (the threshold of instability) and with different distributions functions for $\underset{\sim}{w}(n)$ and $\underset{\sim}{v}(n)$. In all cases K_n converged to K_{opt} within 1,000 iterations.

6.2.2 Second Order Dynamics

EXAMPLE 6.3:

Given:

$$\Phi = \begin{bmatrix} 0.966 & 0.0 \\ 0.155 & 0.894 \end{bmatrix}, \qquad H = I,\ Q = \begin{bmatrix} 1.441 & 0.738 \\ 0.738 & 0.610 \end{bmatrix},\ R = I$$

For this simulation K_o was chosen to be the null matrix [0.0]. The data for the adaptation process are shown in Fig. 6.3. Again the process has essentially converged within a small number of iterations (approximately

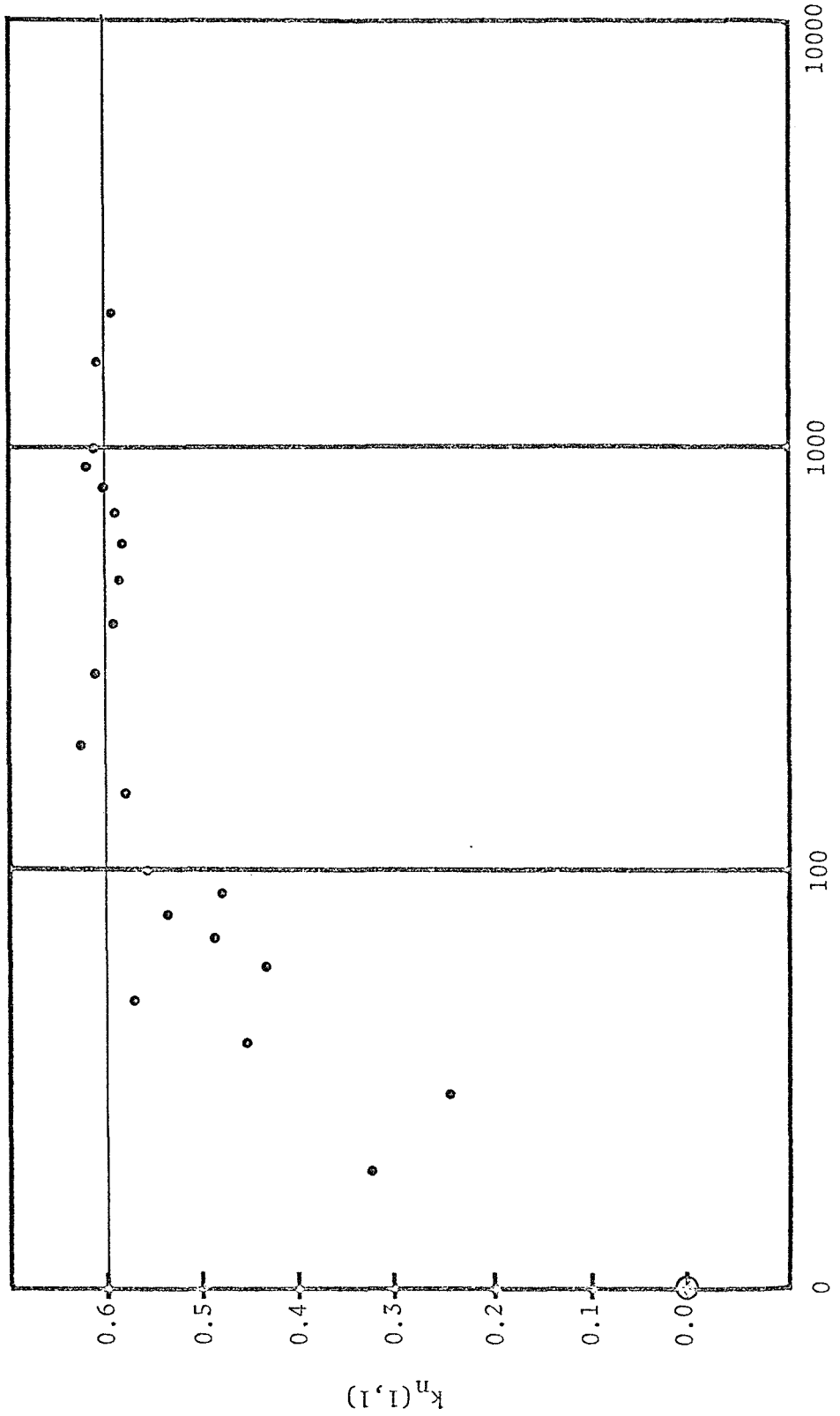


Fig. 6.3 Time Evolution of K_n for Example 6.3: Symmetry Forced

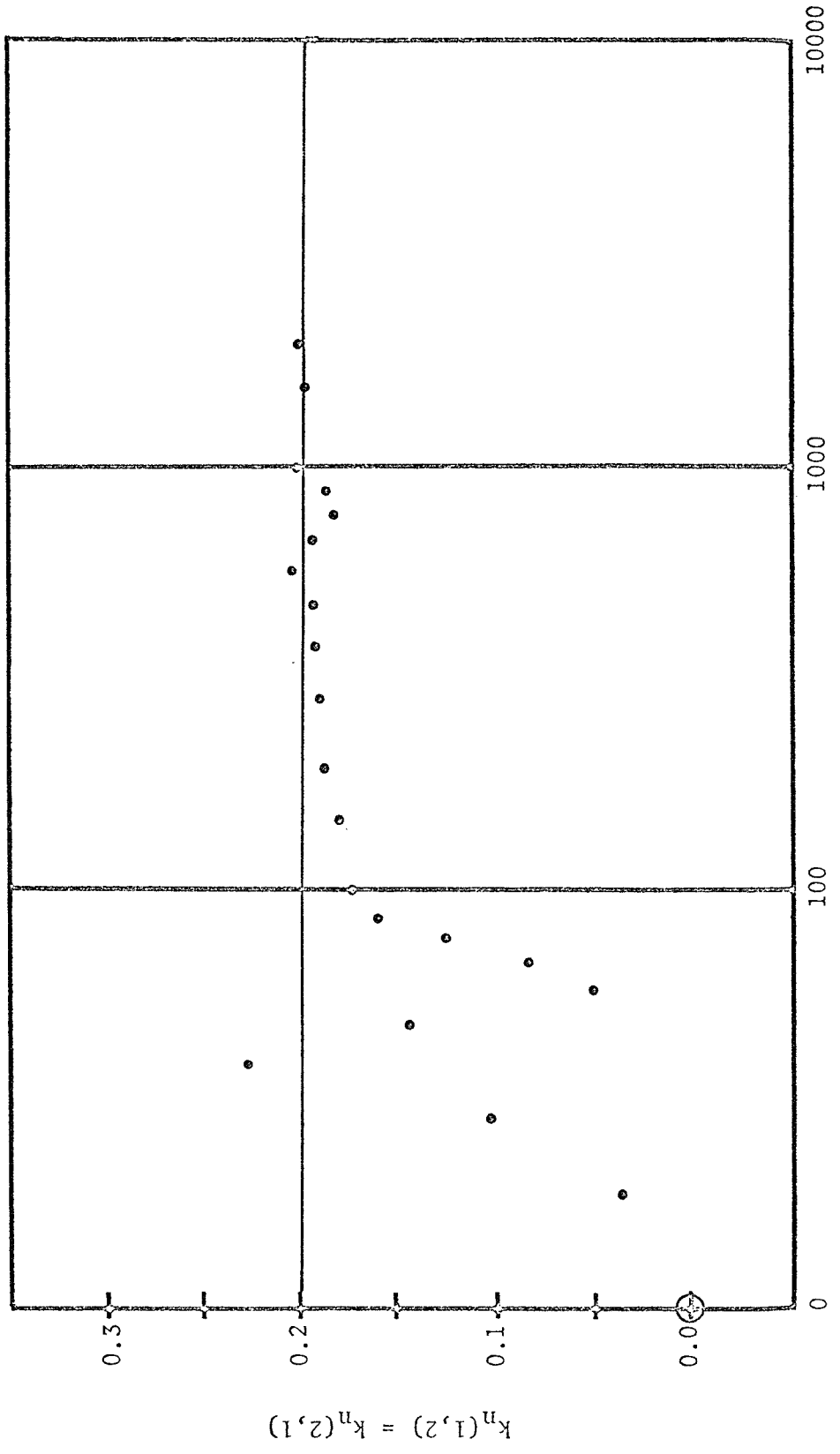


Fig. 6.3 Time Evolution of K_n for Example 6.3: Symmetry Forced (continued)

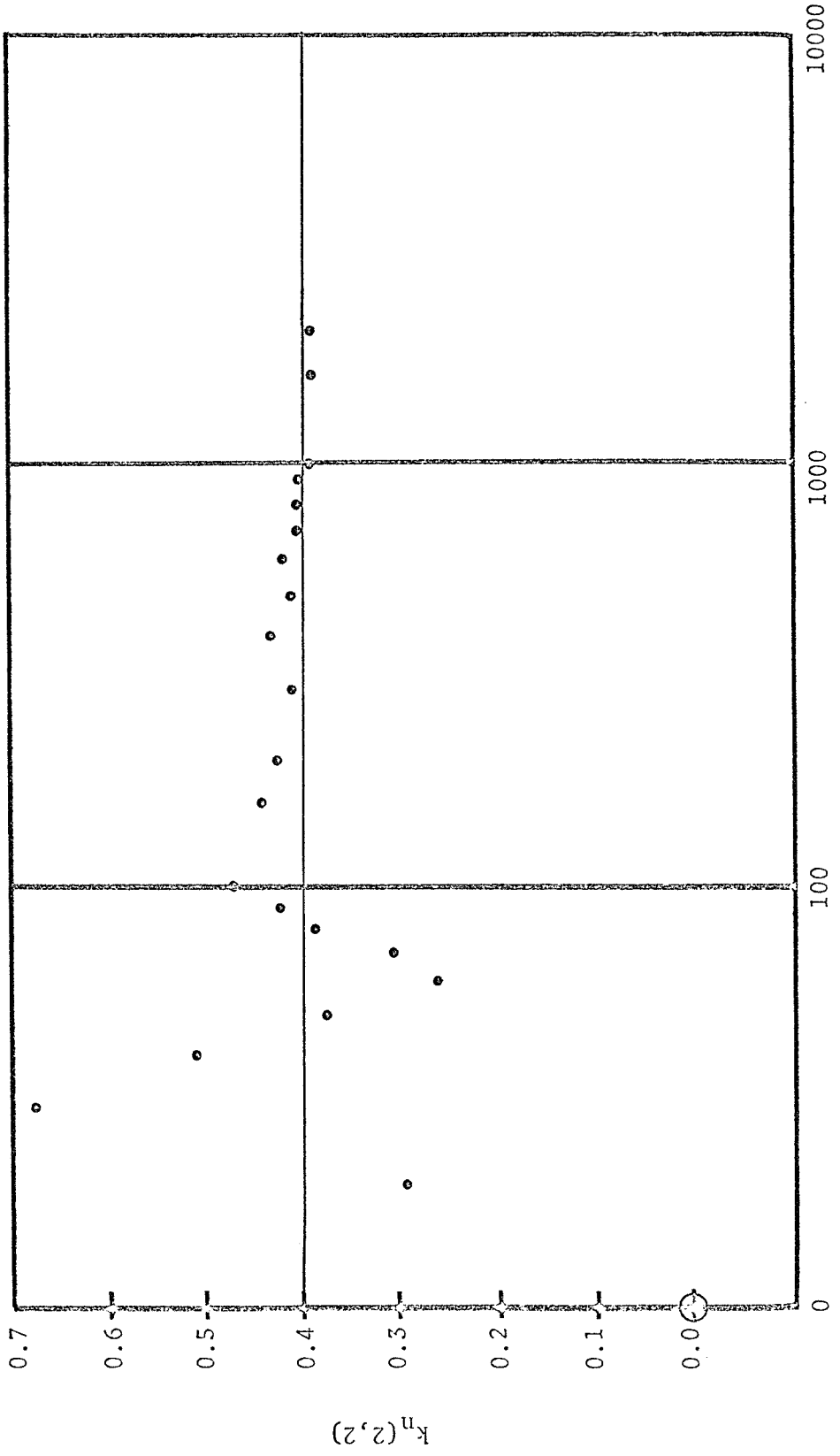


Fig. 6.3 Time Evolution of K_n for Example 6.3: Symmetry Forced (continued)

150). The estimates after 1000 iterations are compared with their optimal values below

$$\hat{K} = \begin{bmatrix} 0.591 & 0.202 \\ 0.202 & 0.397 \end{bmatrix} \qquad K_{opt} = \begin{bmatrix} 0.600 & 0.200 \\ 0.200 & 0.400 \end{bmatrix}$$

$$\hat{\Sigma} = \begin{bmatrix} 1.893 & .960 \\ .960 & 1.009 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$$

$$\hat{\Gamma} = \begin{bmatrix} .635 & .203 \\ .203 & .408 \end{bmatrix} \qquad \Gamma = \begin{bmatrix} .60 & .20 \\ .20 & .40 \end{bmatrix}$$

$$\hat{R} = \begin{bmatrix} 1.164 & -.032 \\ -.032 & .985 \end{bmatrix} \qquad R = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\hat{Q} = \begin{bmatrix} 1.301 & .688 \\ .688 & .611 \end{bmatrix} \qquad Q = \begin{bmatrix} 1.441 & 0.738 \\ 0.738 & 0.610 \end{bmatrix}$$

Note from Fig. 6.3 that symmetry has been forced on $k_n(1,2)$ and $k_n(2,1)$. The adaptation process for the same system and K_o, but where symmetry is not forced, is illustrated in Fig. 6.4. As can be seen, $k_n(1,2)$ and $k_n(2,1)$ converge more rapidly when symmetry is forced.

6.2.3 Third Order Dynamics

EXAMPLE 6.4:

Given:

$$\Phi = \begin{bmatrix} 0.88 & 0.112 & 0.05 \\ 0.0 & 0.76 & 0.0 \\ 0.06 & 0.0 & 0.97 \end{bmatrix} \qquad H = I$$

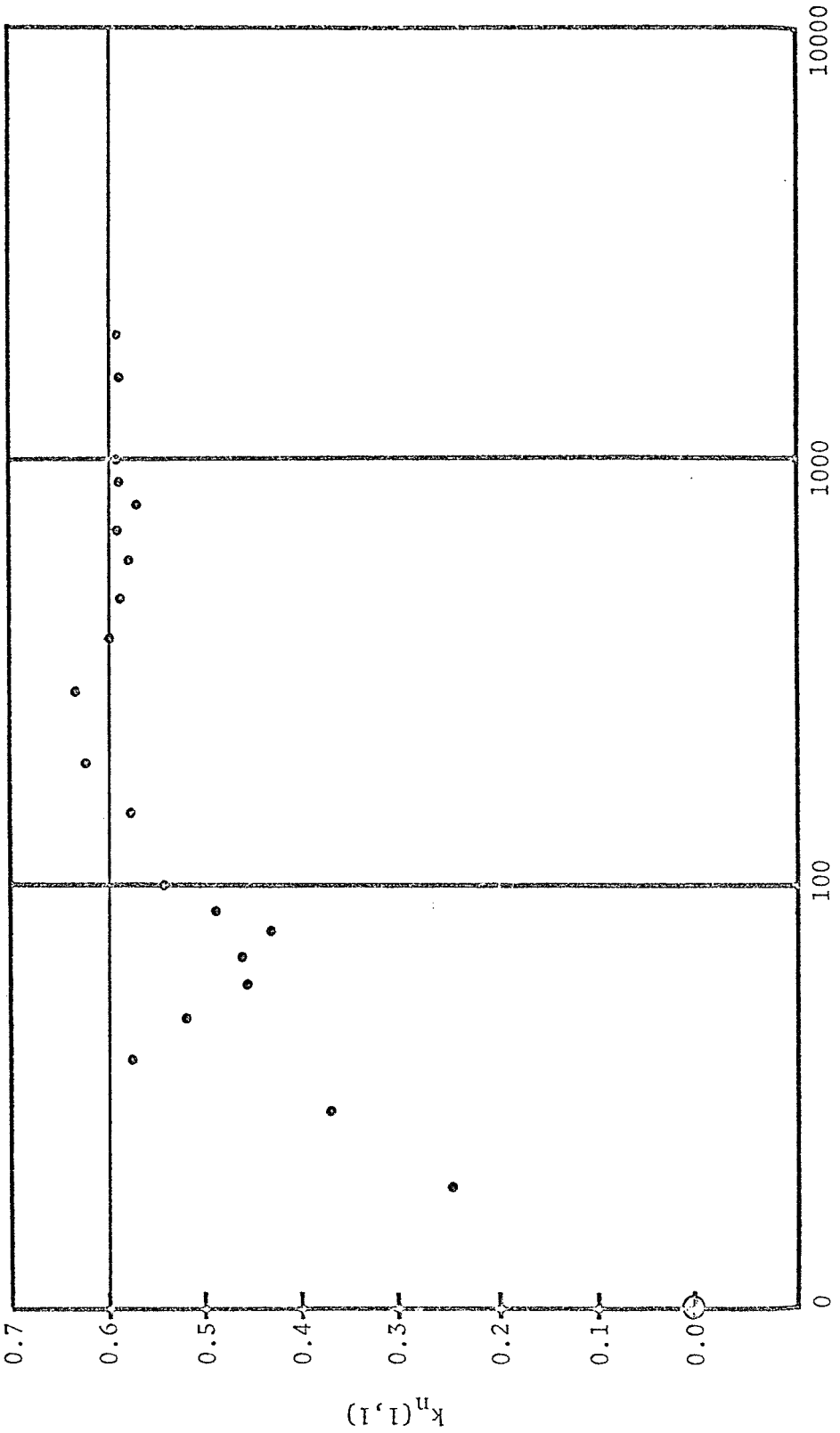


Fig. 6.4 Time Evolution of K_n for Example 6.3: Symmetry Not Forced

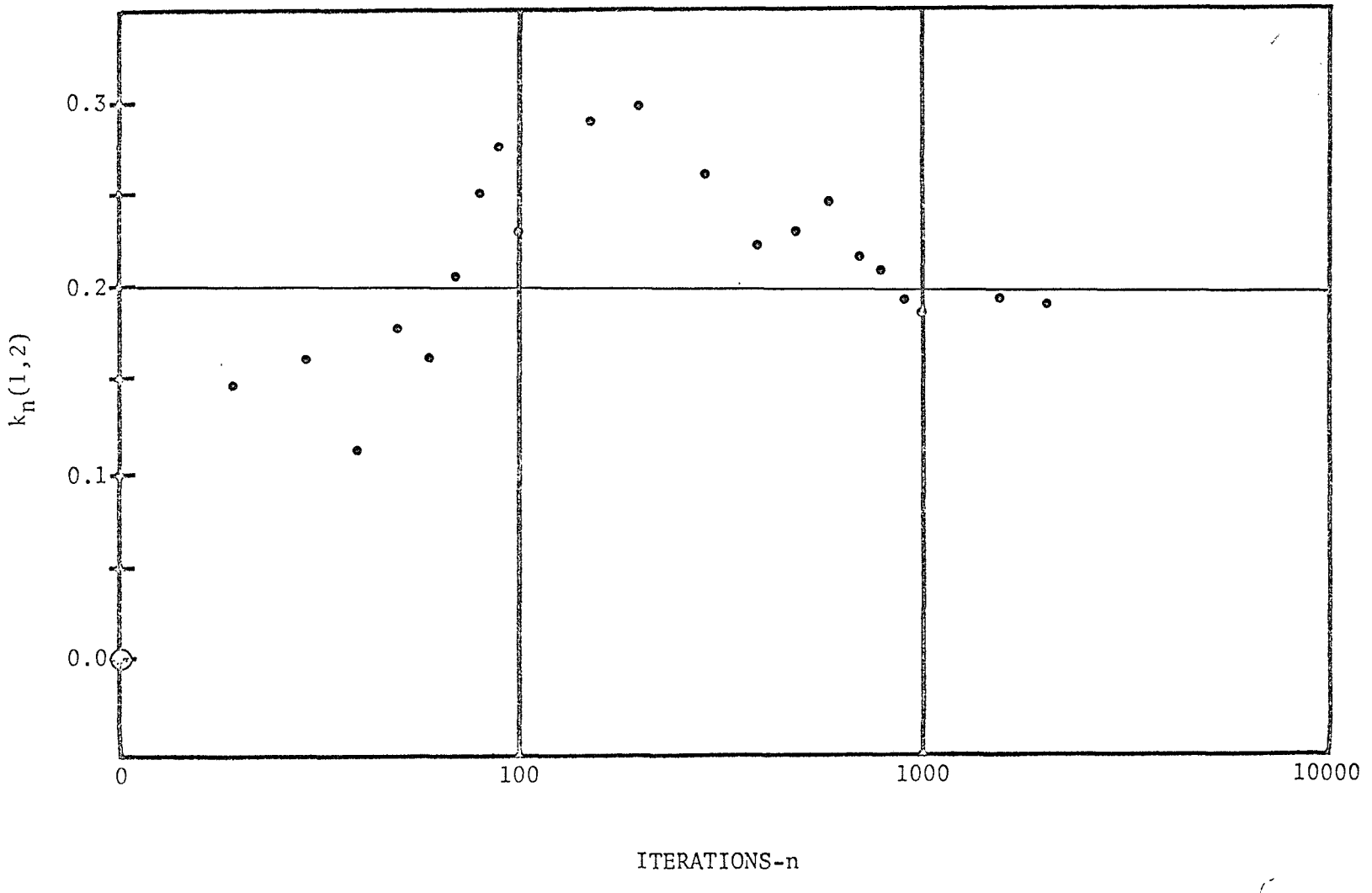


Fig. 6.4 Time Evolution of K_n for Example 6.3: Symmetry Not Forced (continued)

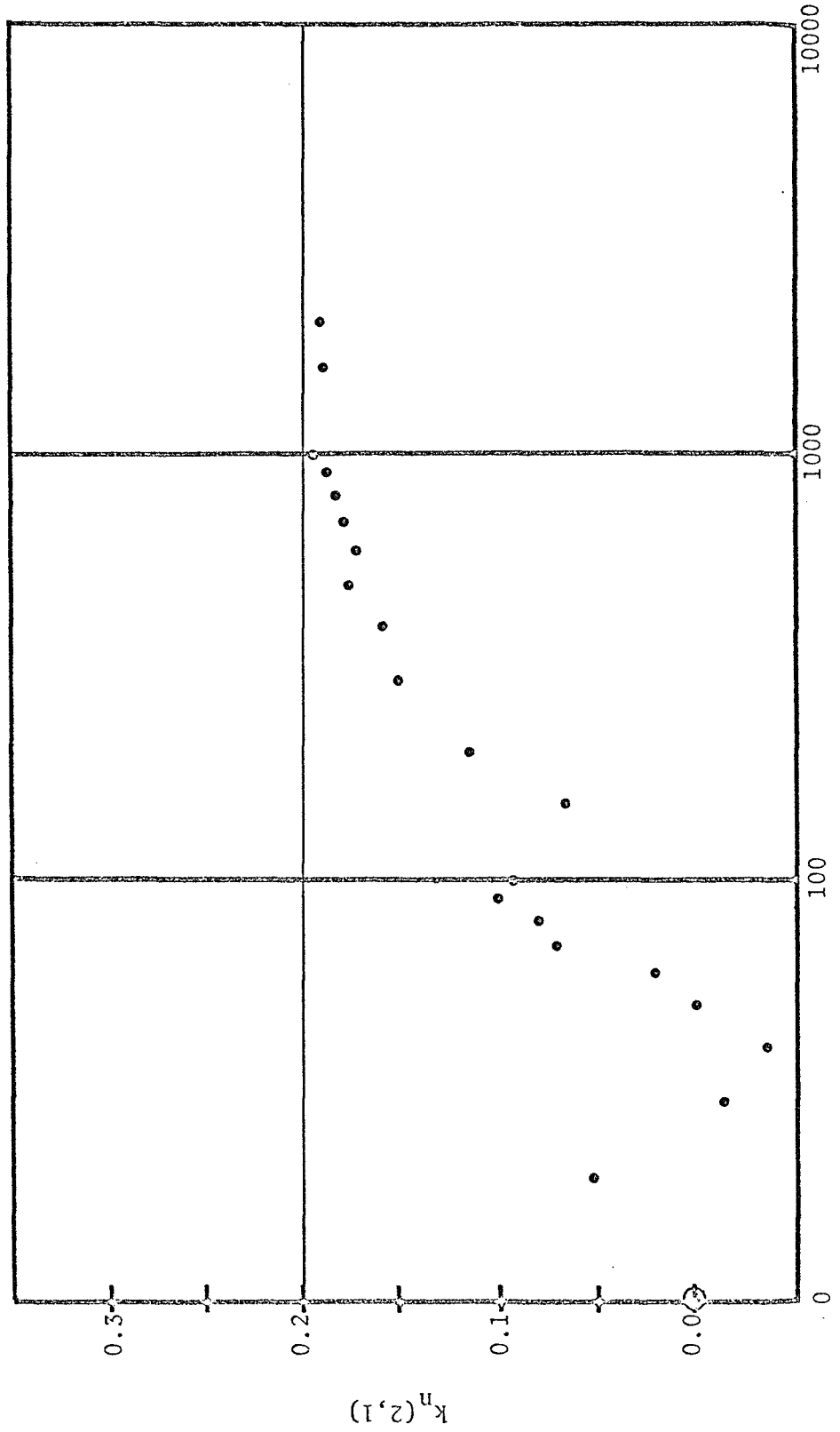


Fig. 6.4 Time Evolution of K_n for Example 6.3: Symmetry Not Forced (continued)

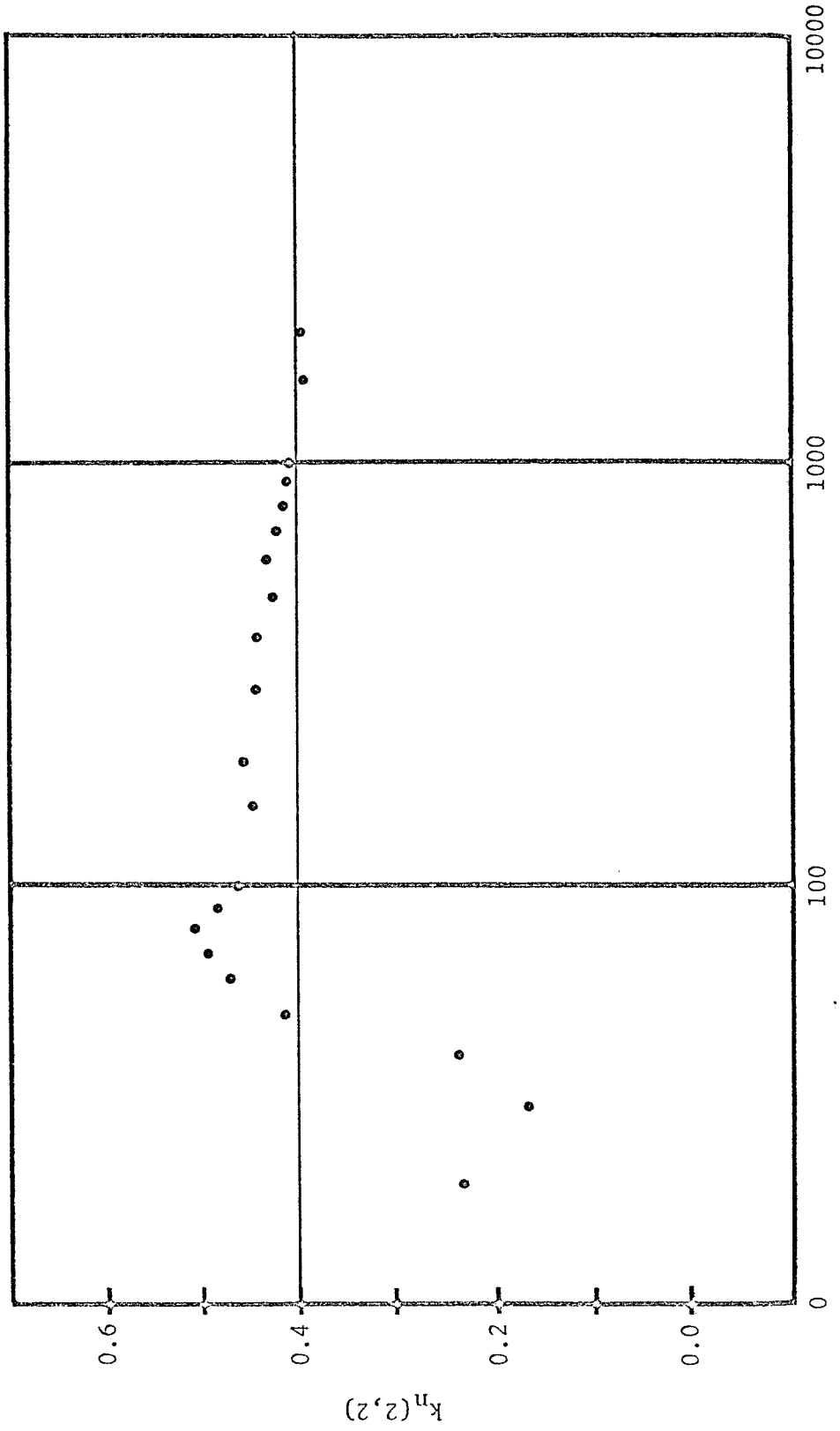


Fig. 6.4 Time Evolution of K_n for Example 6.3: Symmetry Not Forced (continued)

$$Q = \begin{bmatrix} 2.54 & 1.46 & 1.14 \\ 1.46 & 2.73 & 1.25 \\ 1.14 & 1.25 & 1.70 \end{bmatrix} \qquad R = \begin{bmatrix} 1.11 & 1.83 & -.865 \\ 1.83 & 7.25 & 0.30 \\ -.864 & 0.30 & 2.00 \end{bmatrix}$$

The system model specified above was simulated for Gaussian, triangular and uniformly distributed noise processes $\underset{\sim}{w}(n)$ and $\underset{\sim}{v}(n)$. For Gaussian processes and with $K_o = [0.0]$ the time evolution of the adaptive algorithm is shown in Fig. 6.5. One thousand iterations are now required for convergence. The results after 3,000 iterations are summarized below.

$$\hat{K} = \begin{bmatrix} 0.861 & -.162 & 0.269 \\ 0.256 & 0.220 & 0.241 \\ 0.281 & -.047 & 0.570 \end{bmatrix}, \qquad K_{opt} = \begin{bmatrix} 0.871 & -.160 & 0.260 \\ 0.267 & 0.227 & 0.234 \\ 0.271 & -.049 & 0.567 \end{bmatrix}$$

$$\hat{\Sigma} = \begin{bmatrix} 2.93 & 2.13 & 1.02 \\ 2.13 & 4.26 & 1.55 \\ 1.02 & 1.55 & 2.49 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 3.0 & 2.0 & 1.0 \\ 2.0 & 4.0 & 1.5 \\ 1.0 & 1.5 & 2.5 \end{bmatrix}$$

$$\hat{\Gamma} = \begin{bmatrix} .478 & .561 & -.269 \\ .561 & 2.27 & .295 \\ -.269 & .295 & .868 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} .458 & .511 & -.282 \\ .511 & 2.21 & .309 \\ -.282 & .307 & .885 \end{bmatrix}$$

$$\hat{R} = \begin{bmatrix} 1.24 & 1.82 & -.915 \\ 1.82 & 7.03 & .270 \\ -.915 & .270 & 2.00 \end{bmatrix}, \qquad R = \begin{bmatrix} 1.11 & 1.83 & -.864 \\ 1.83 & 7.25 & .300 \\ -.864 & .300 & 2.00 \end{bmatrix}$$

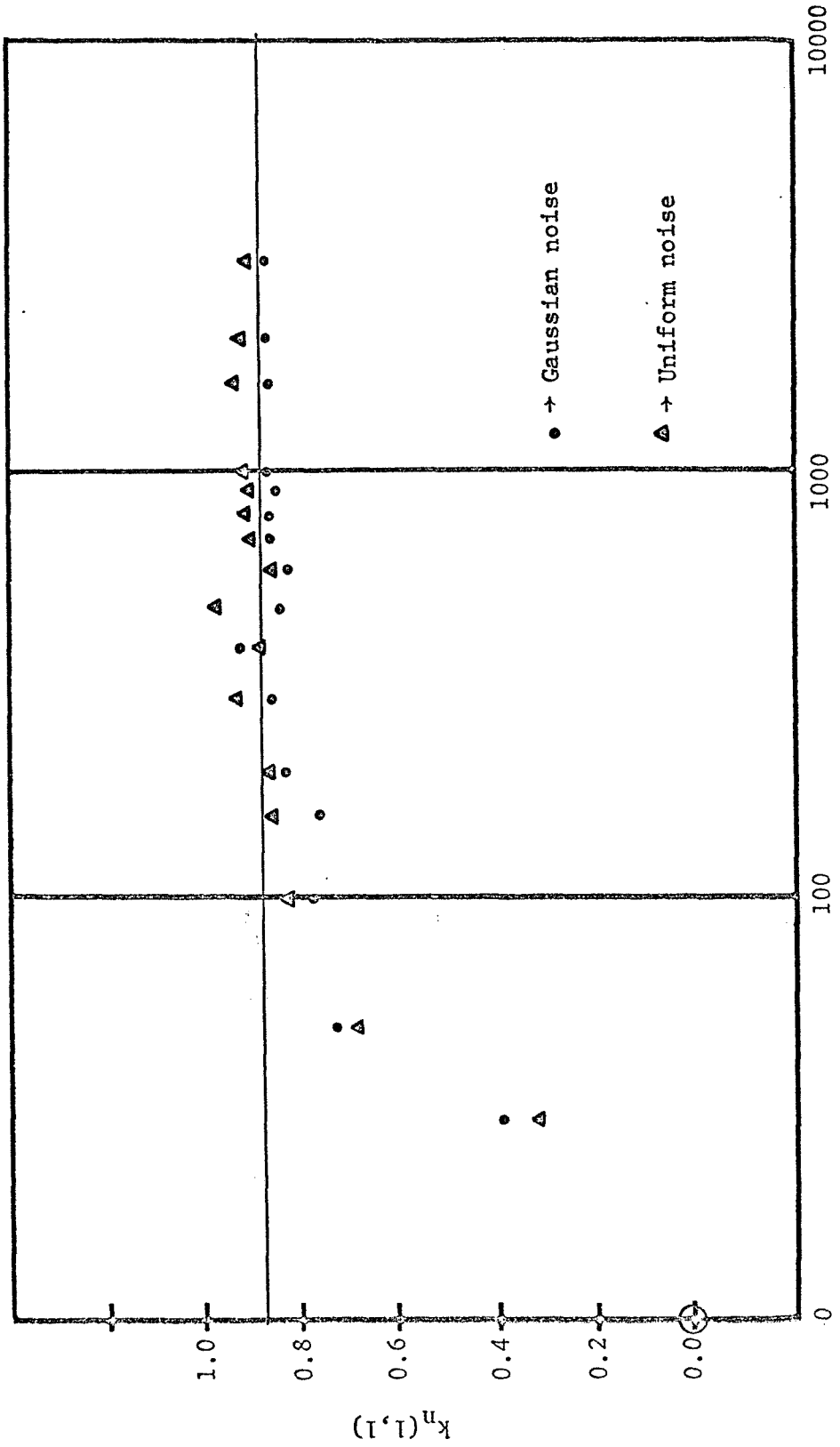


Fig. 6.5 Time Evolution of K_n for Example 6.4

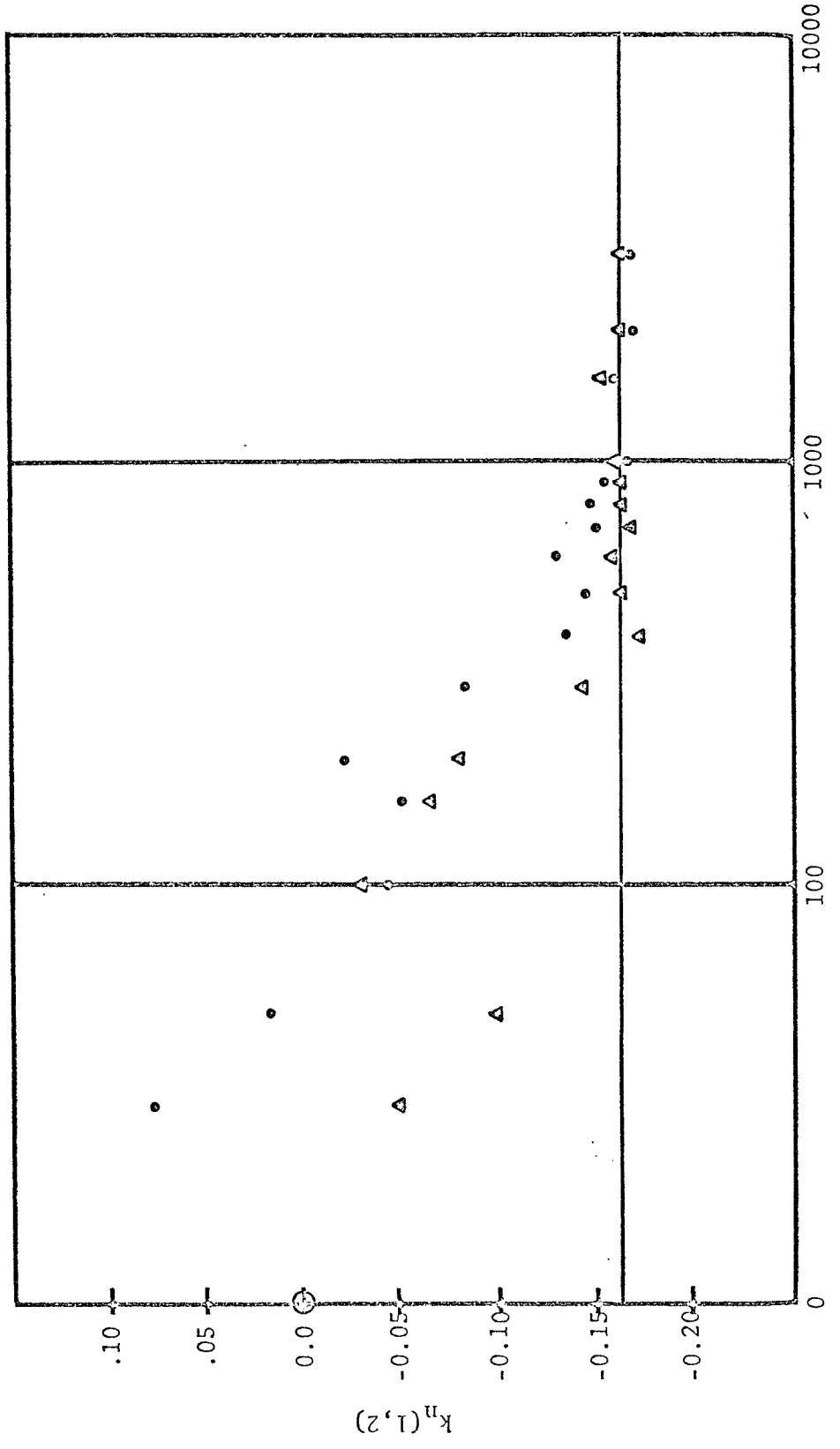


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

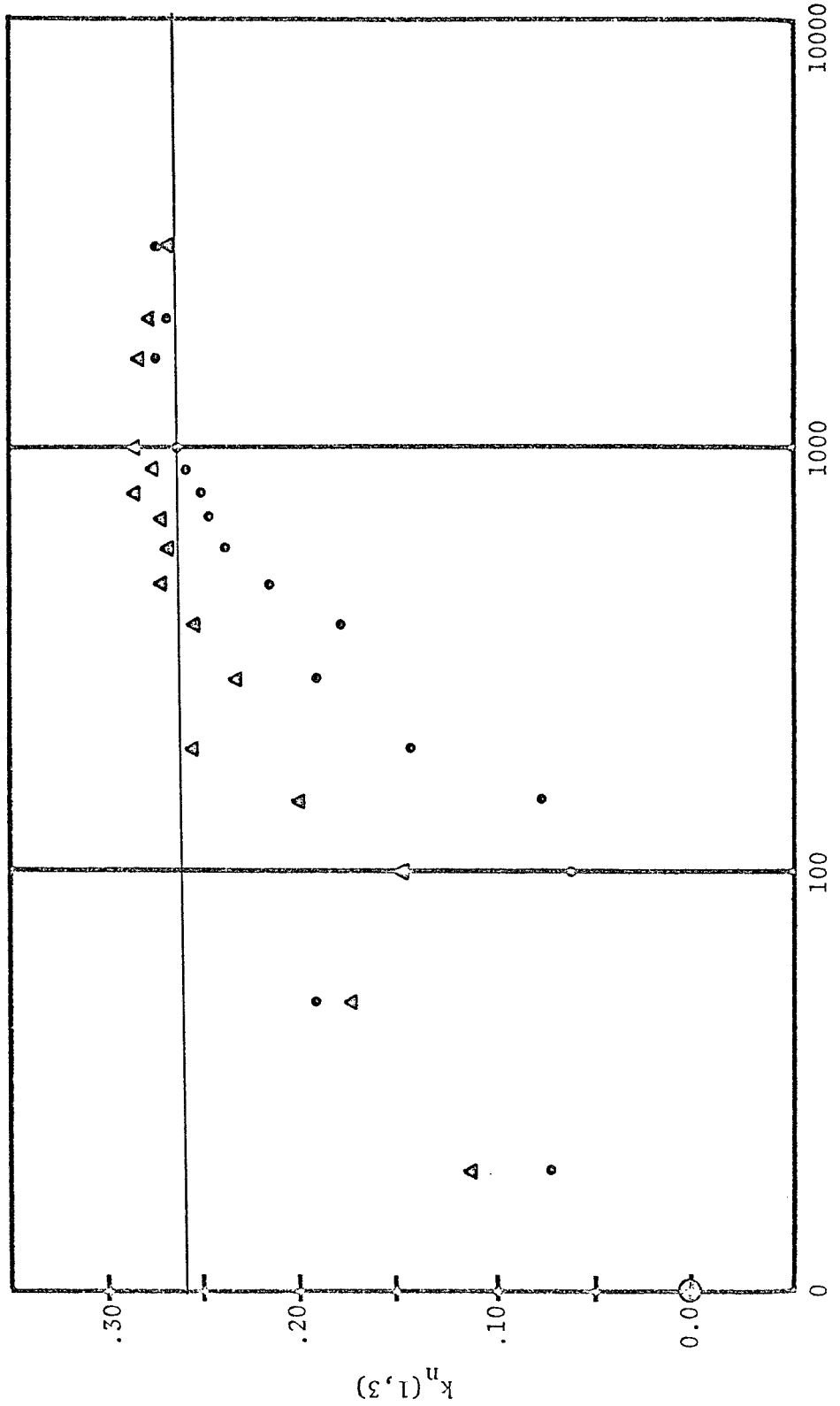


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

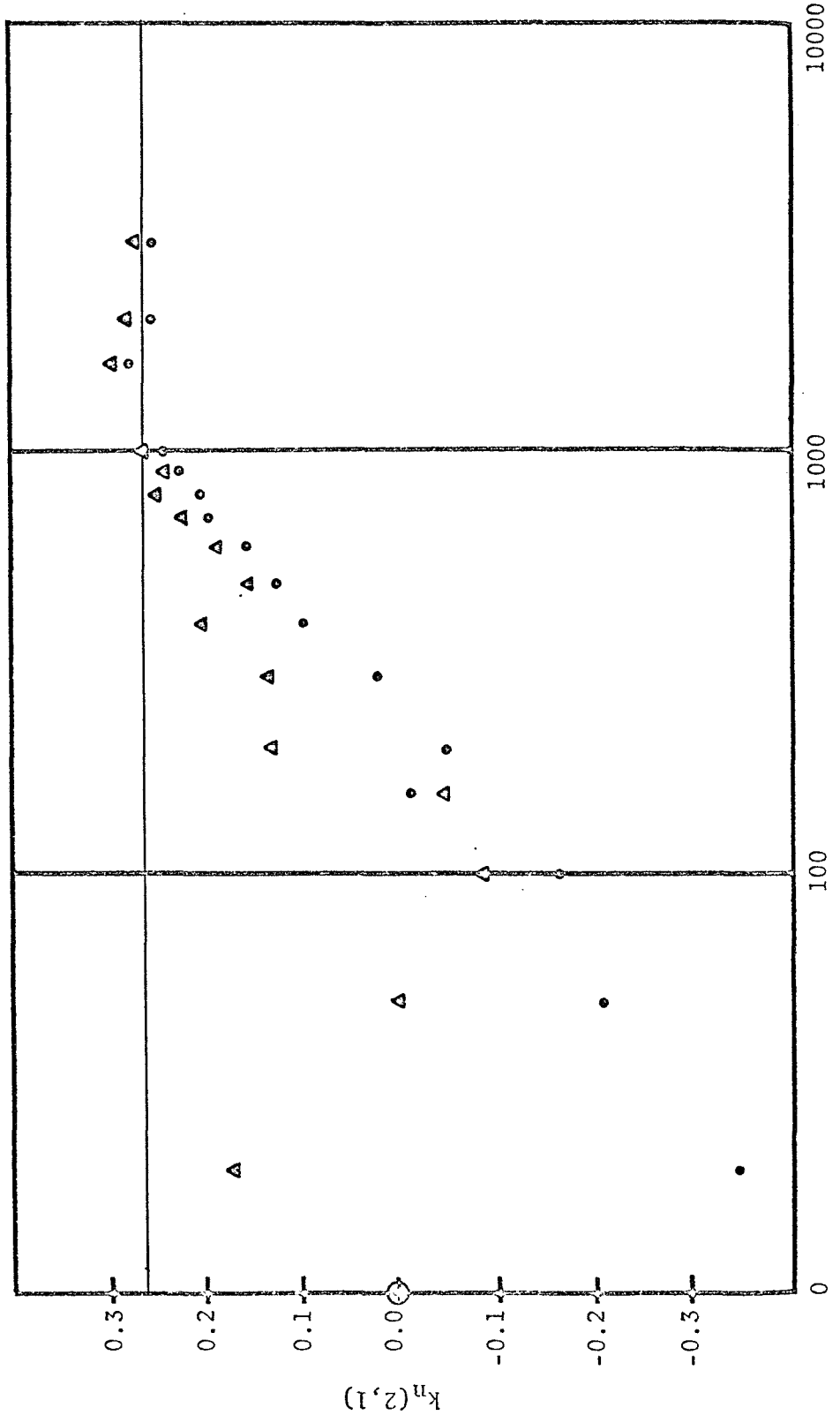


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

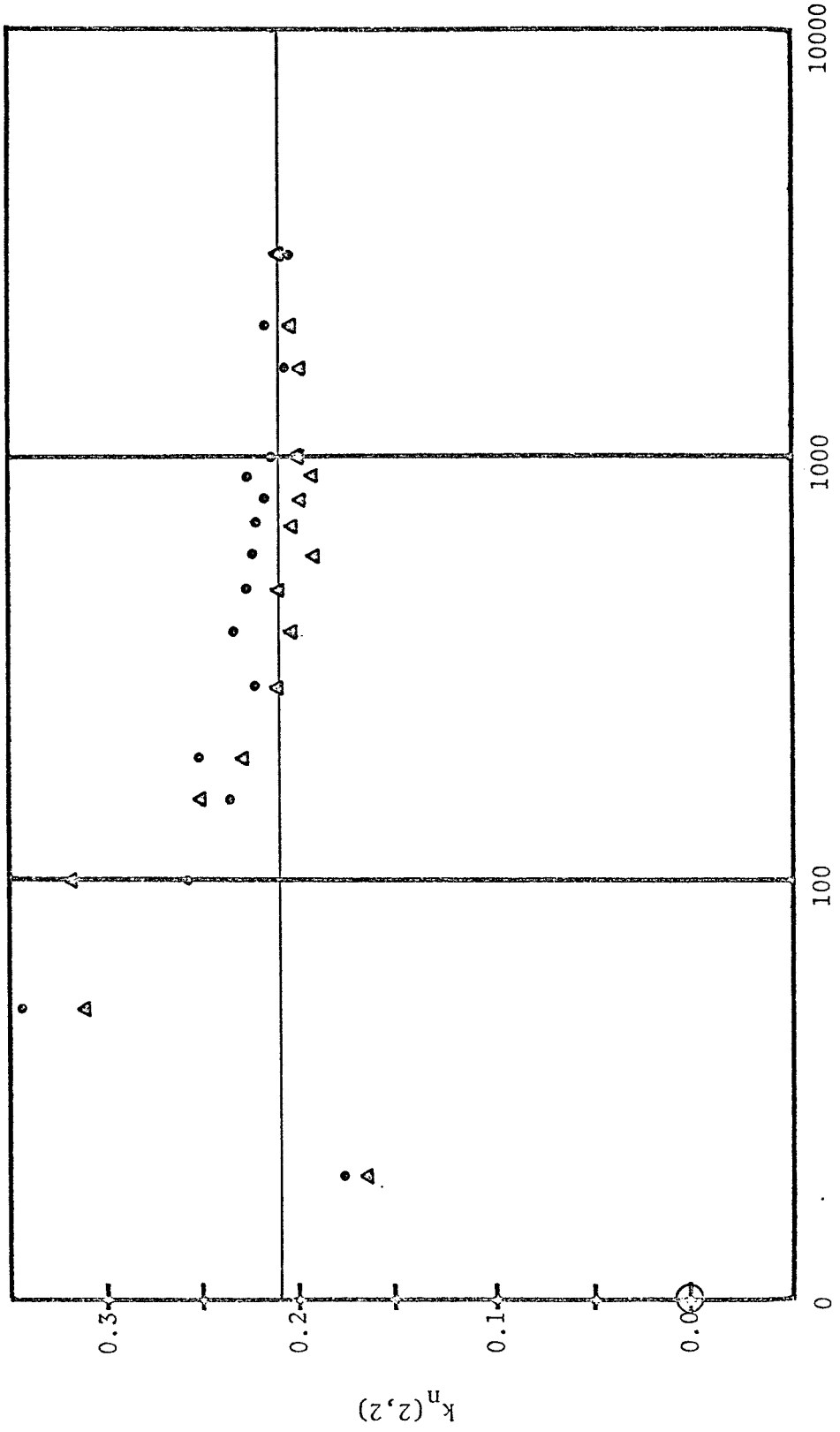


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

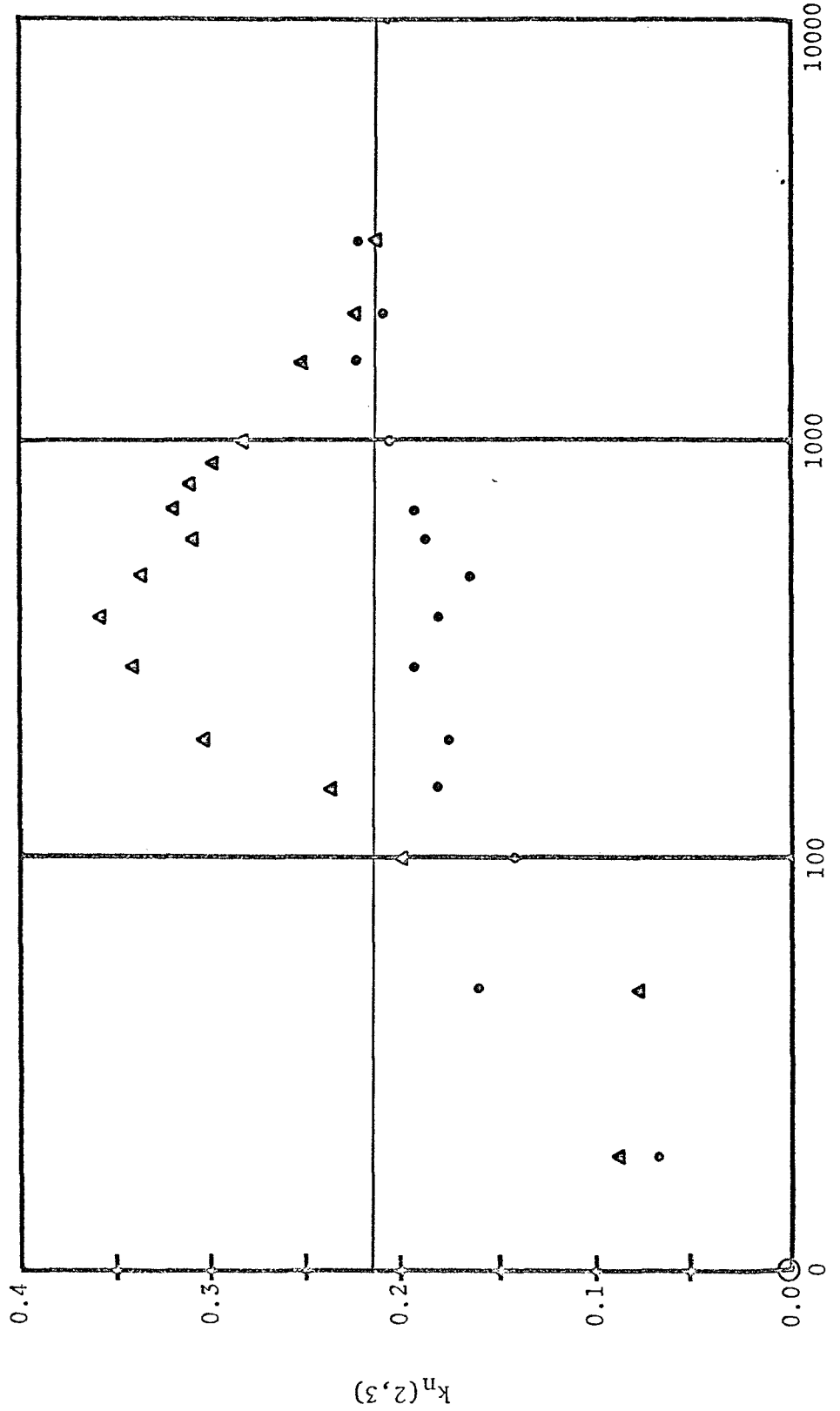


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

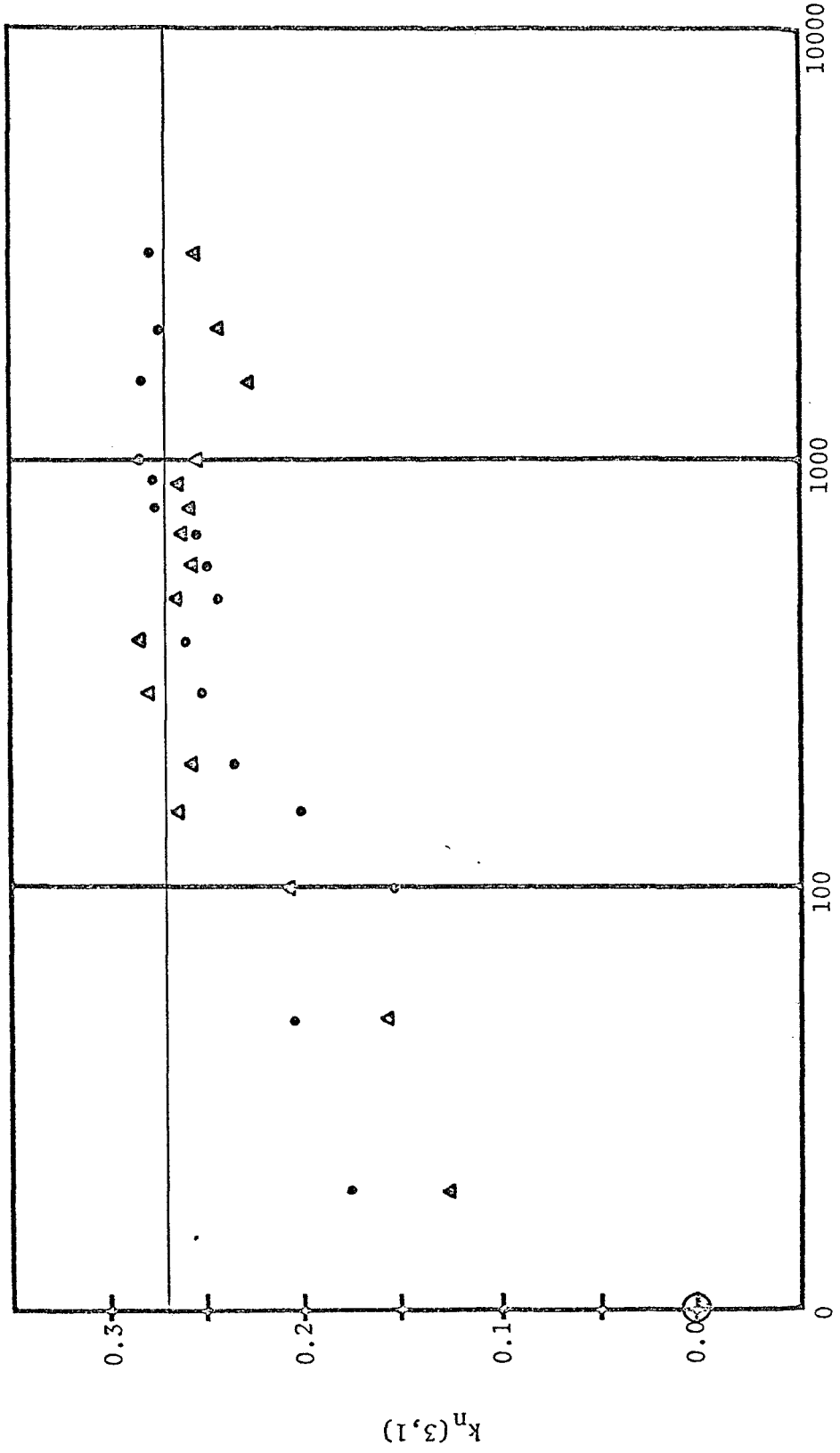


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

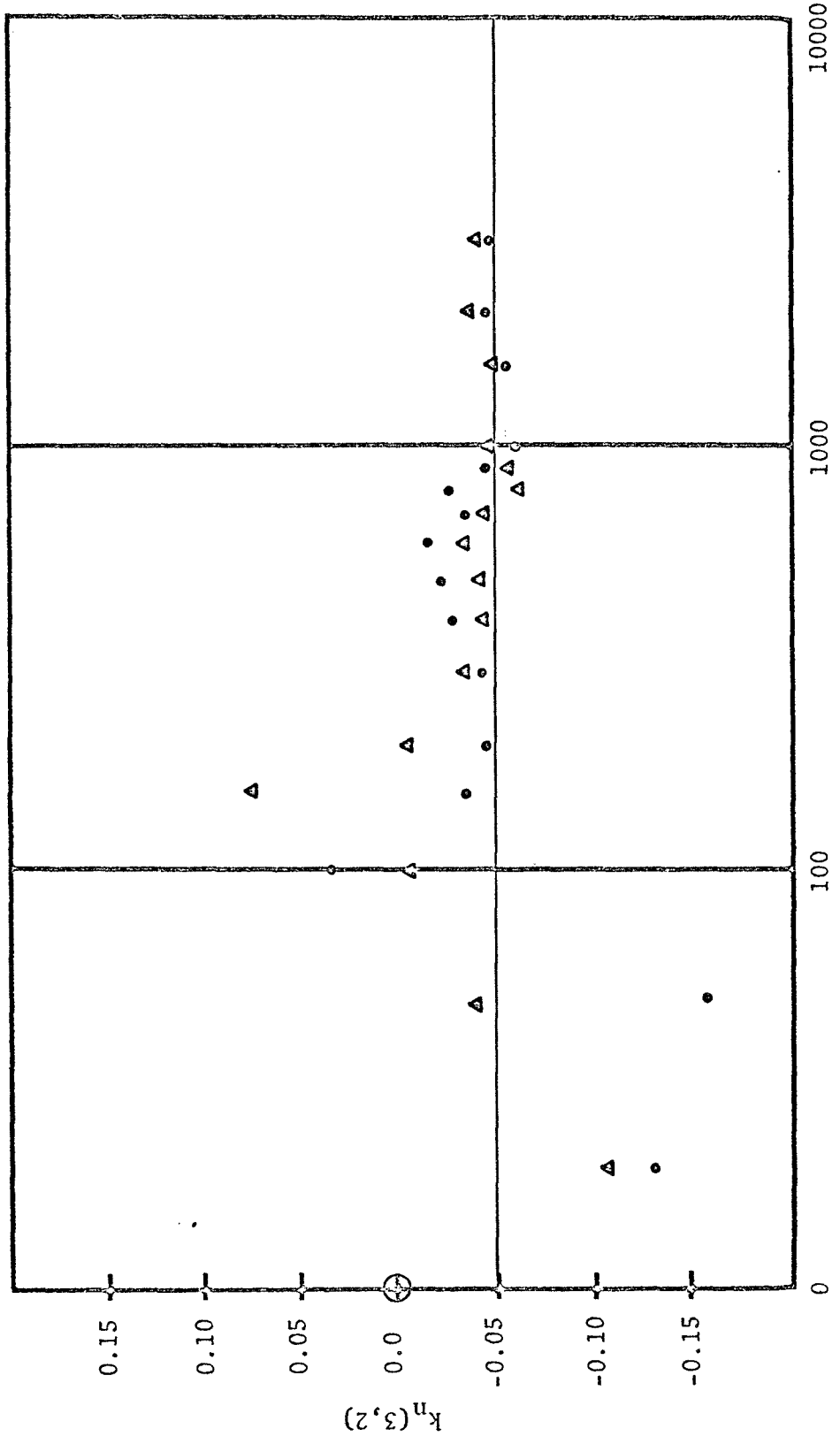


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

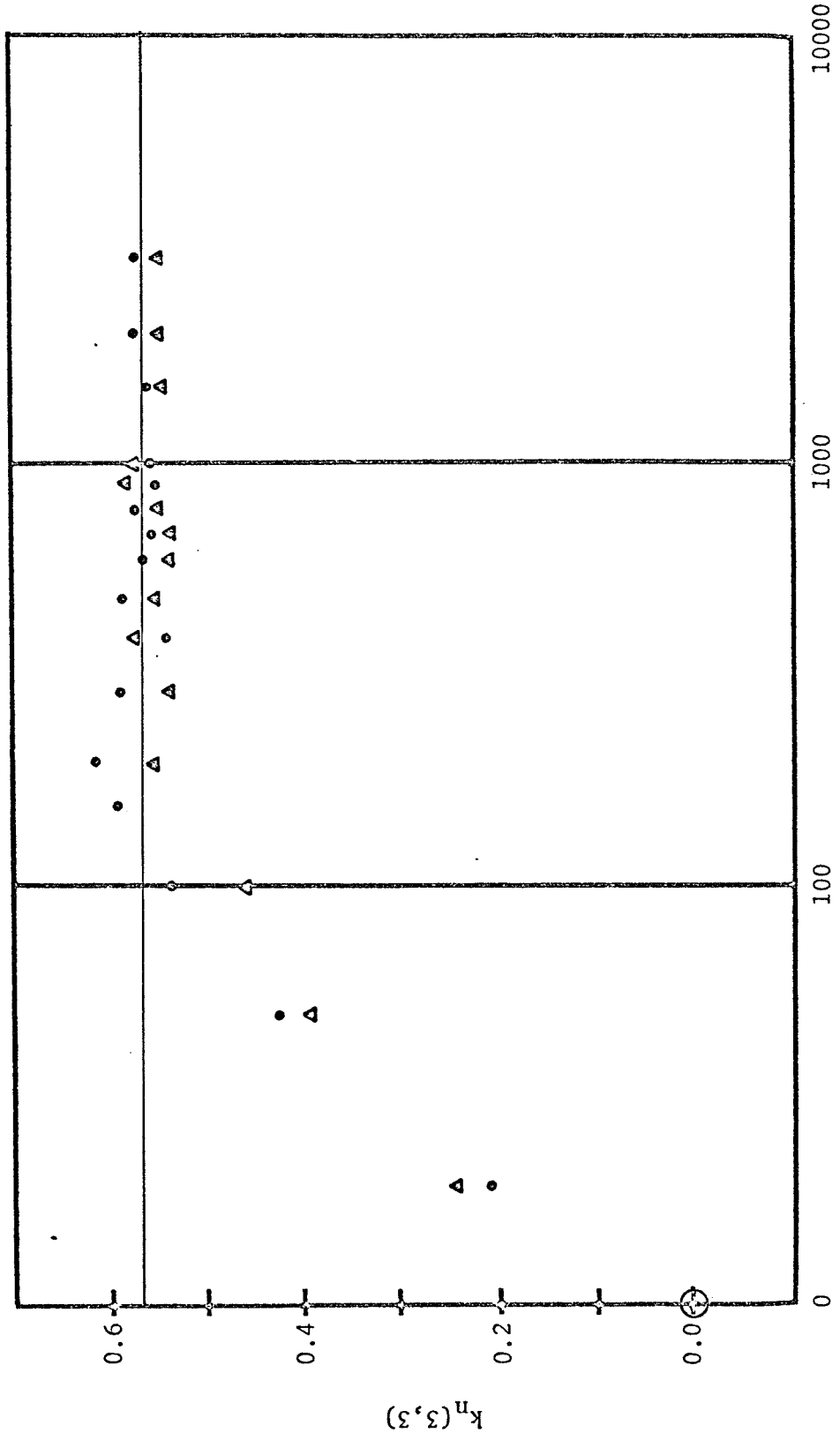


Fig. 6.5 Time Evolution of K_n for Example 6.4 (continued)

$$\hat{Q} = \begin{bmatrix} 2.40 & 1.55 & 1.15 \\ 1.55 & 2.95 & 1.30 \\ 1.15 & 1.30 & 1.70 \end{bmatrix}, \qquad Q = \begin{bmatrix} 2.54 & 1.46 & 1.14 \\ 1.46 & 2.73 & 1.25 \\ 1.14 & 1.25 & 1.70 \end{bmatrix}$$

As can be seen the estimates are in excellent agreement with the optimal values. The mean square error (MSE) of the filtering process is defined as

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]\}$$

Since the error covariance matrix Γ is defined by Eq. (2.21) as

$$E\{[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)][\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)]^T\}$$

Therefore, the mean square error is the sum of the major diagonal elements of Γ. Performing this calculation, the optimum and estimated mean square errors are

$$\hat{MSE} = 3.66, \qquad MSE = 3.55$$

The actual M.S.E. for the filter given by $\hat{K}$ can be determined by calculating the actual Γ using Eq. (5.8). It is

$$(MSE)_{act} = 3.62$$

This indicates that the "learned" filter $\hat{K}$ produces a MSE only 1.97% larger than the optimal filter. The corresponding one-step prediction mean square errors (PMSE) are

$$\hat{PMSE} = 9.68, \quad (PMSE)_{act} = 9.55, \quad PMSE = 9.50$$

To demonstrate the insensitivity of the algorithm of Eq. (4.43) to the assumed initial value of the filter K_o was chosen to be the identity matrix. After 1,000 iterations,

$$\hat{K} = \begin{bmatrix} 0.883 & -.171 & .254 \\ 0.253 & 0.239 & .258 \\ 0.260 & -.052 & .555 \end{bmatrix}$$

Theoretically, the adaptive algorithm is independent of the amplitude distributions of $\underset{\sim}{w}(n)$ and $\underset{\sim}{v}(n)$. Consequently, the final value of the filter should be the same. To test this experimentally, uniform and triangular distributions were used. The simulation results for 3,000 iterations with $K = [0.0]$ are

$$(\hat{K})_{\text{triangle}} = \begin{bmatrix} .881 & -.155 & .275 \\ .261 & .233 & .240 \\ .263 & -.047 & .554 \end{bmatrix}$$

$$(\hat{K})_{\text{uniform}} = \begin{bmatrix} .876 & -.163 & .267 \\ .270 & .223 & .230 \\ .249 & -.041 & .549 \end{bmatrix}$$

The case for uniformly distributed noise is also plotted in Fig. 6.5 for comparison with the Gaussian distributed case.

6.2.4 Fourth Order Dynamics

EXAMPLE 6.5:

Given:

$$\Phi = \begin{bmatrix} 0.87 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.69 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.94 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.75 \end{bmatrix}, \qquad H = I$$

$$Q = \begin{bmatrix} 1.60 & 0.32 & 0.88 & 0.85 \\ 0.32 & 1.09 & 0.78 & 0.96 \\ 0.88 & 0.78 & 2.38 & 1.43 \\ 0.85 & 0.96 & 1.43 & 2.15 \end{bmatrix}$$

$$R = \begin{bmatrix} 6.89 & 3.42 & -.55 & 1.41 \\ 3.42 & 6.17 & -1.10 & -0.34 \\ -.55 & -1.10 & 1.21 & 0.33 \\ 1.41 & -0.34 & 0.33 & 1.25 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1.60 & 0.32 & 0.88 & 0.85 \\ 0.32 & 1.09 & 0.78 & 0.96 \\ 0.88 & 0.78 & 2.38 & 1.43 \\ 0.85 & 0.96 & 1.43 & 2.15 \end{bmatrix}$$

To initialize the simulation process for this system, K_o was again chosen to be the null matrix. The time evolution of the filter adaptation for Gaussian distributions is shown in Fig. 6.6. As before, in addition to learning K_{opt}, Σ, Γ, R, and Q were also estimated. The results after 4,000 iterations are given below

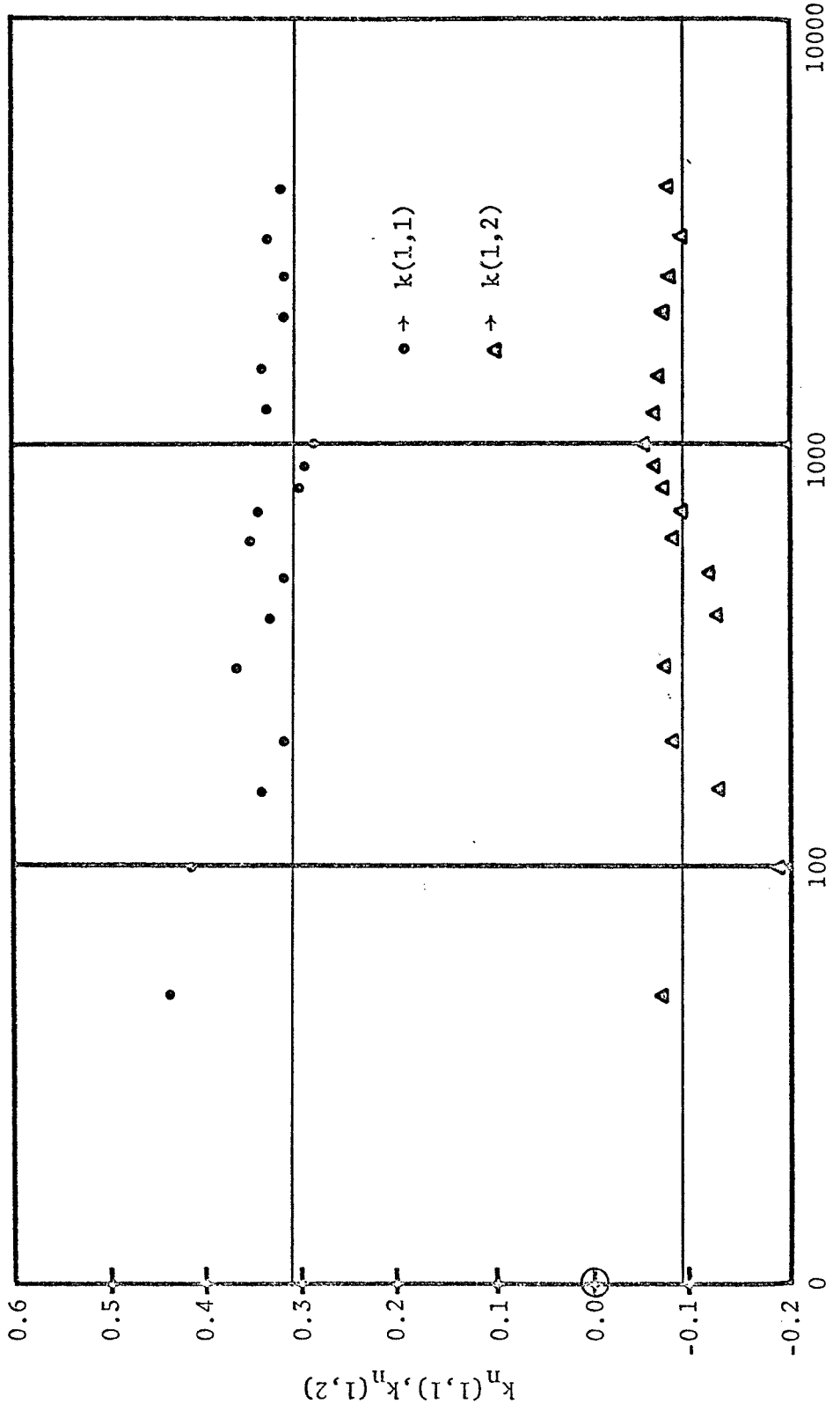


Fig. 6.6 Time Evolution of K_n for Example 6.5

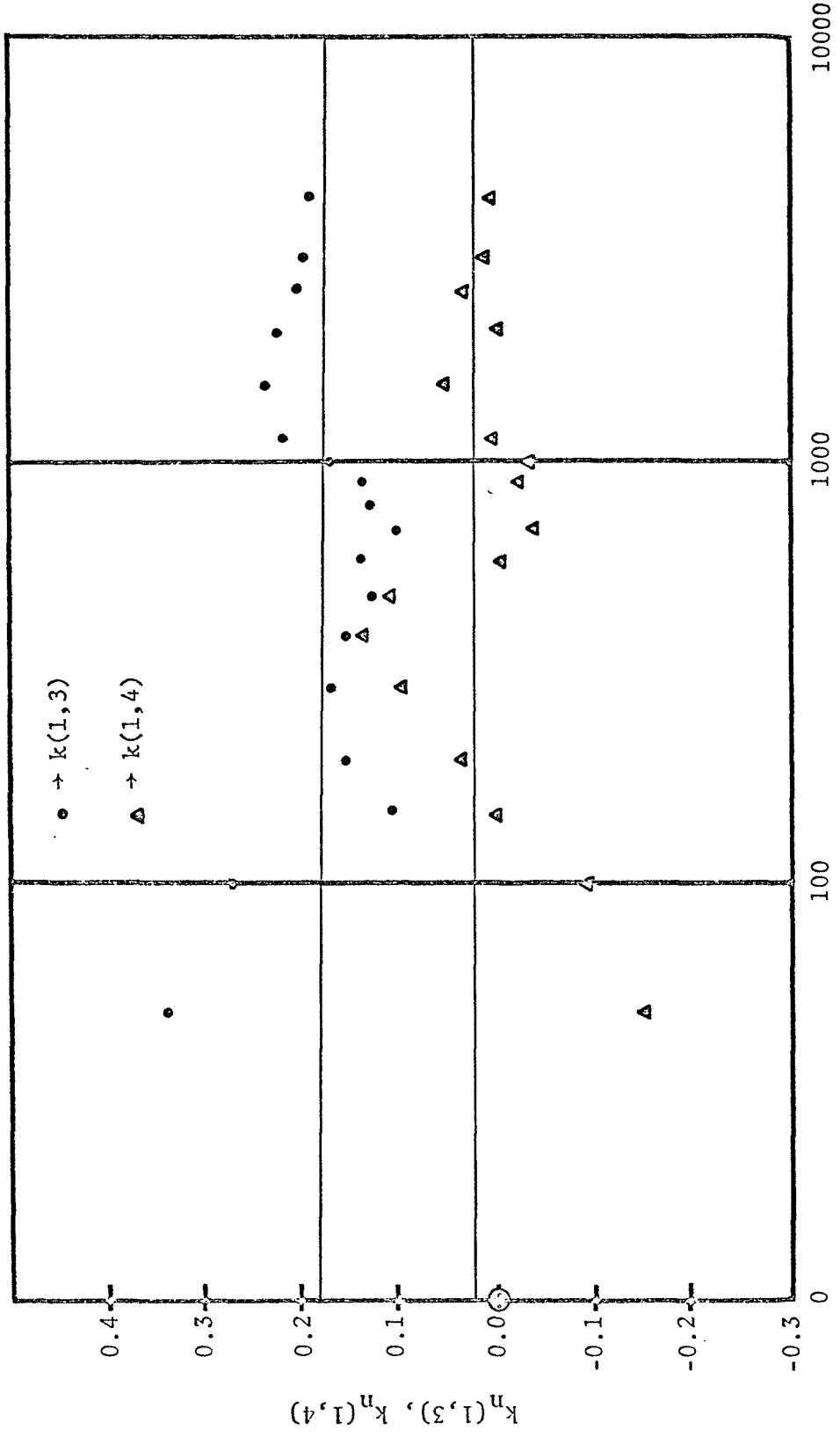


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

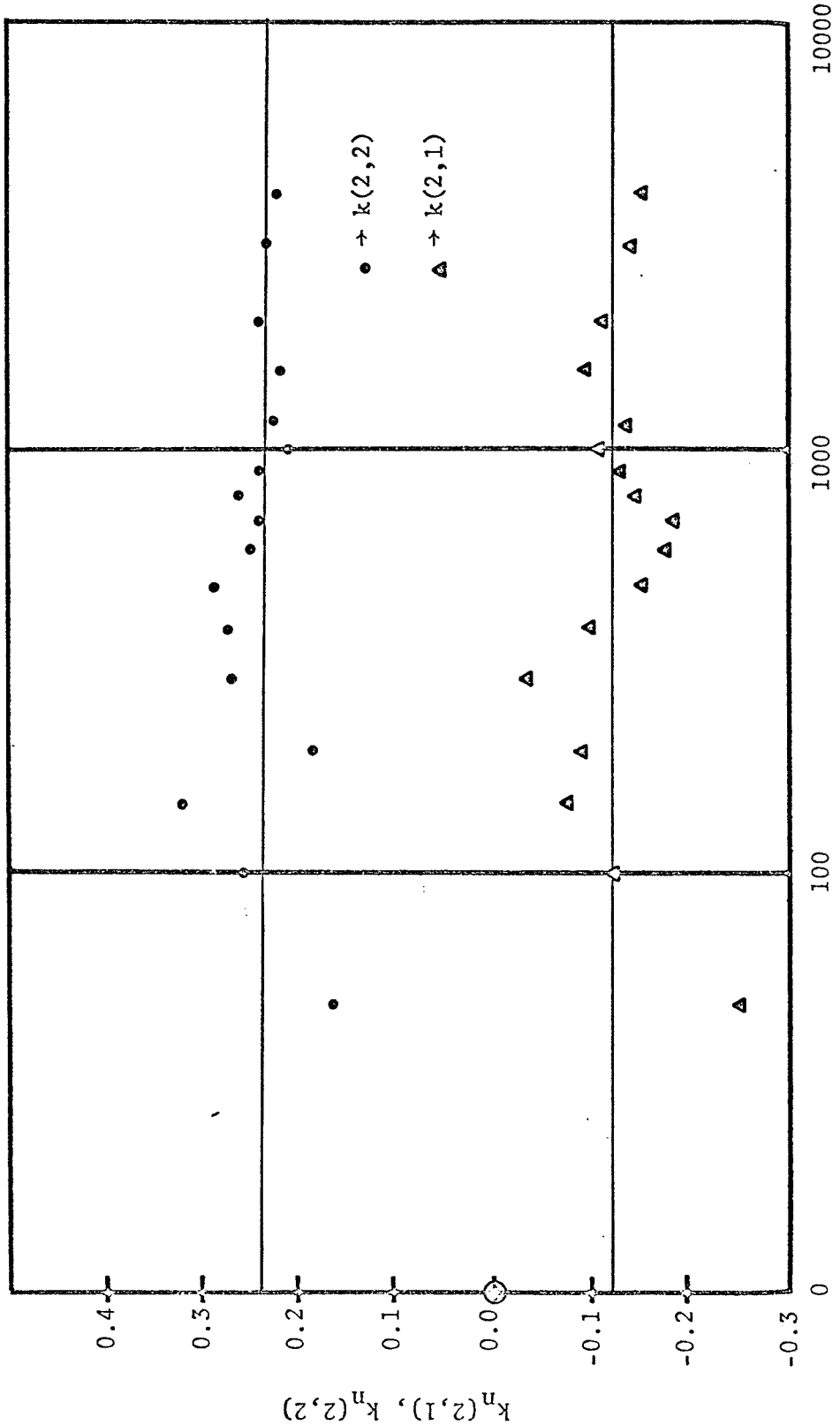


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

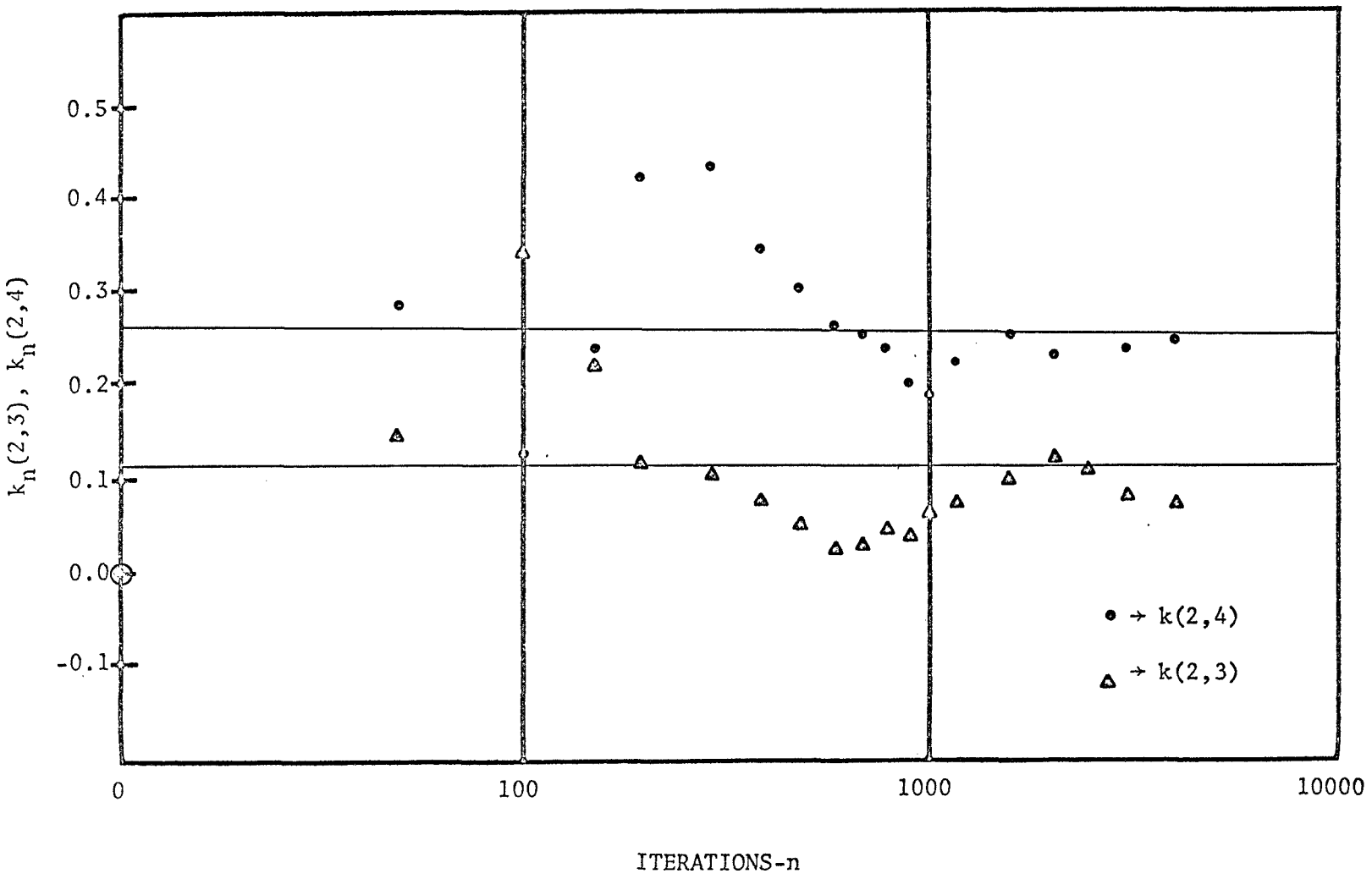


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

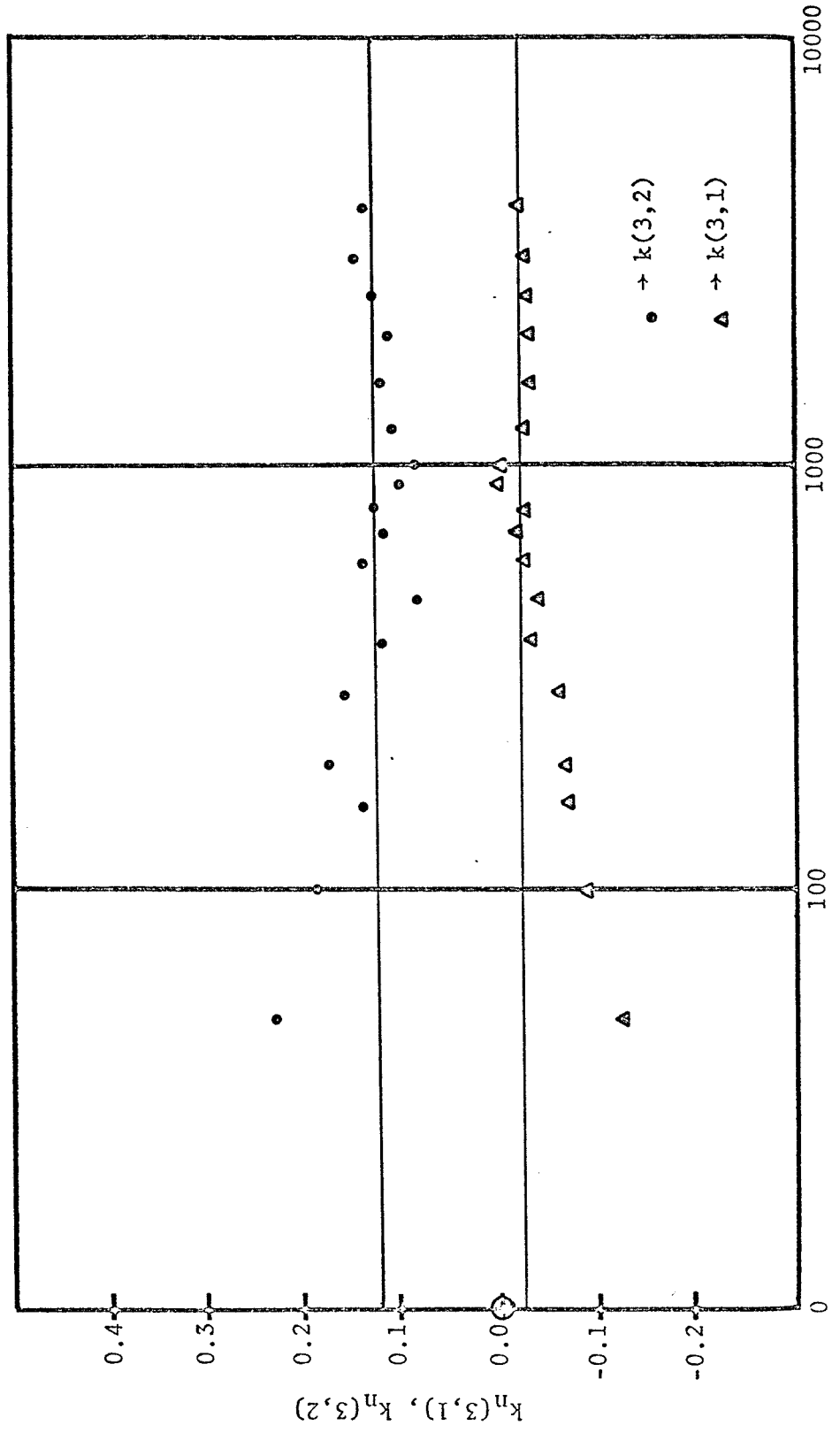


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

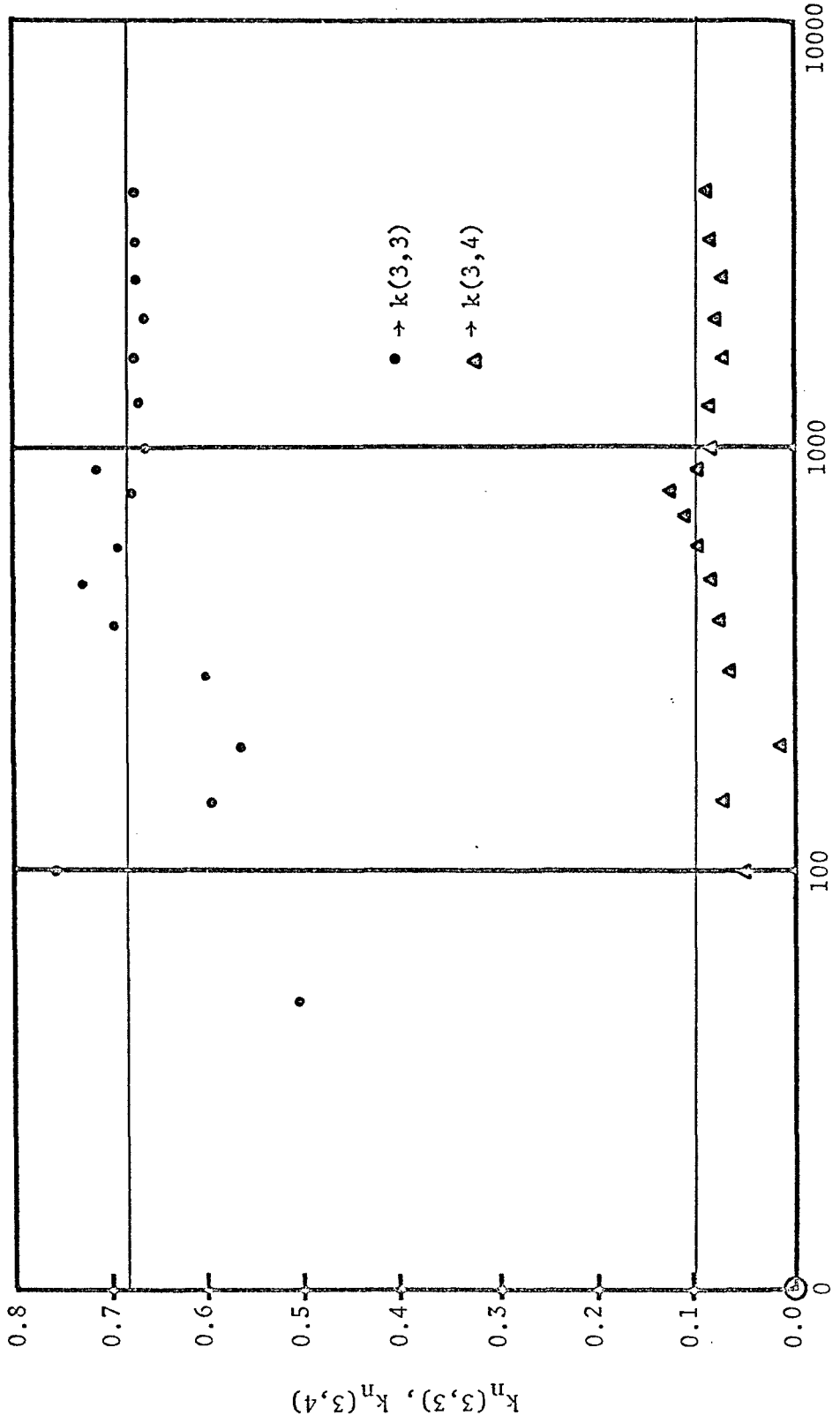


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

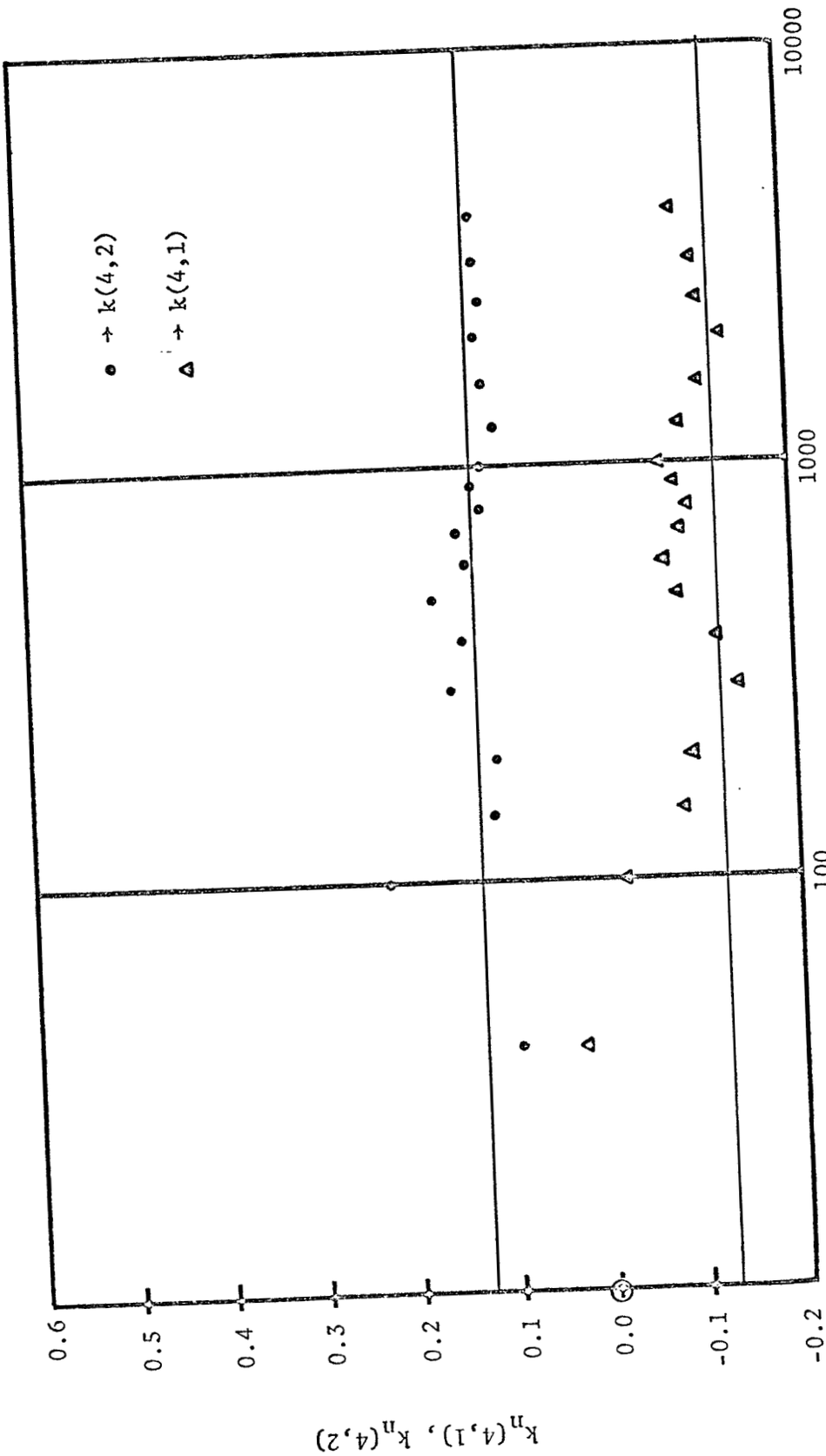


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

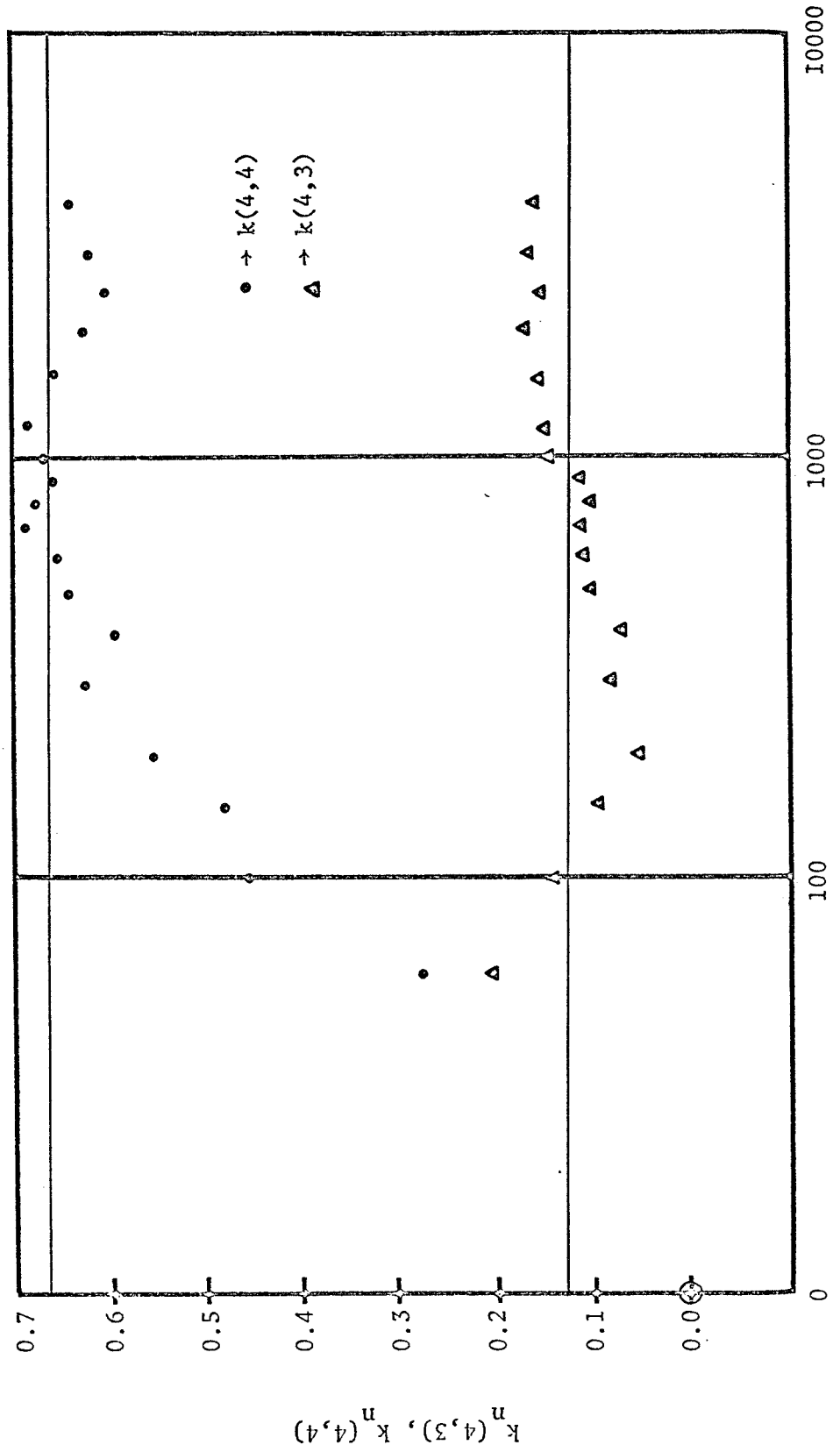


Fig. 6.6 Time Evolution of K_n for Example 6.5 (continued)

$$\hat{K} = \begin{bmatrix} .327 & -.086 & .208 & .030 \\ -.130 & .234 & .095 & .254 \\ -.004 & .130 & .679 & .096 \\ -.091 & .137 & .163 & .628 \end{bmatrix}, \quad K_{opt} = \begin{bmatrix} .323 & -.095 & .181 & .037 \\ -.118 & .239 & .116 & .259 \\ -.006 & .123 & .686 & .102 \\ -.116 & .138 & .137 & .667 \end{bmatrix}$$

$$\hat{\Sigma} = \begin{bmatrix} 3.18 & .744 & 1.06 & 1.31 \\ .744 & 1.71 & .854 & 1.29 \\ 1.06 & .854 & 3.02 & 1.46 \\ 1.31 & 1.29 & 1.46 & 2.51 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 3.0 & .5 & 1.0 & 1.2 \\ .5 & 1.5 & .75 & 1.0 \\ 1.0 & .75 & 3.0 & 1.5 \\ 1.2 & 1.0 & 1.5 & 2.5 \end{bmatrix}$$

$$\hat{\Gamma} = \begin{bmatrix} 1.94 & .399 & .154 & .560 \\ .399 & .835 & -.068 & .118 \\ .154 & -.068 & .700 & .060 \\ .560 & .118 & .060 & .579 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 1.85 & .308 & .148 & .542 \\ .308 & .854 & -.048 & .081 \\ .148 & -.048 & .701 & .100 \\ .542 & .081 & .100 & .628 \end{bmatrix}$$

$$\hat{R} = \begin{bmatrix} 6.7 & 3.3 & -.43 & 1.34 \\ 3.3 & 6.00 & -1.06 & -.524 \\ -.43 & -1.06 & 1.20 & .032 \\ 1.34 & -.524 & .032 & 1.24 \end{bmatrix}, \quad R = \begin{bmatrix} 6.89 & 3.42 & -.55 & 1.41 \\ 3.42 & 6.17 & -1.10 & -.335 \\ -.55 & -1.10 & 1.21 & .033 \\ 1.41 & -.335 & .033 & 1.25 \end{bmatrix}$$

$$\hat{Q} = \begin{bmatrix} 1.71 & .503 & .935 & .946 \\ .503 & 1.31 & .898 & 1.23 \\ .935 & .898 & 2.39 & 1.42 \\ .946 & 1.23 & 1.42 & 2.18 \end{bmatrix}, \quad Q = \begin{bmatrix} 1.60 & .315 & .879 & .847 \\ .315 & 1.09 & .781 & .958 \\ .879 & .781 & 2.38 & 1.43 \\ .847 & .958 & 1.43 & 2.15 \end{bmatrix}$$

These estimates are again in good agreement with the optimal values, but as the dimension of the problem increases so does the number of iterations required to attain this agreement. By summing the diagonal elements of

the appropriate error covariance matrix, the corresponding mean square errors are calculated to be

$$\hat{MSE} = 4.05 \qquad MSE = 4.04 \qquad (MSE)_{ACT} = 4.11$$

Therefore, the "learned" filter $\hat{K}$ produces a MSE only 1.74% larger than the optimal filter assuming perfect a priori information. This example represents the best results obtained for fourth order problems. Many other examples were simulated, and in every case the actual mean square error was less than 2.75% greater than its optimum value.

6.3 Nonlinear Dynamics

In this section the nonlinear system algorithms of Section 5.5 are investigated. The first class of nonlinearities considered satisfy a global Lipschitz condition. The second class have a unique inverse. In both cases the form of the filtering equation is given by Eq. (5.40).

6.3.1 Nonlinearities That Satisfy a Global Lipschitz Condition

Consider the class of nonlinear systems $\{f(x): |f(x_n) - f(x_m)| \leq \phi|x_n - x_m|$, for all $x_n, x_m\}$. Then the adaptive algorithm, which learns the filter that is optimal in the sense that it satisfies Eq. (5.42), is given by Eq. (5.43)

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n}\phi^{-1}\delta_{n+1}\delta_n W_{n+1} \tag{5.43}$$

Because the optimal filter is unknown in the nonlinear case, the results of this section are compared with those for a linear system with the same Φ and noise covariances Q and R. Also from the state equation (5.38), a little reflection reveals that the lower bound for the

prediction error covariance Σ is given by Q. Therefore, the plant perturbation covariance Q can be used as a reference for evaluating the performance of the adaptively "learned" filter.

EXAMPLE 6.6: Saturation Nonlinearity

Given: $\phi = 0.5$, $Q = R = 1$

$$f(x) = \begin{cases} \phi b & \text{for } x > b \\ \phi x & \text{for } |x| \leq b \\ -\phi b & \text{for } x < -b \end{cases} \tag{5.44}$$

For this problem it is obvious that if b is increased without bound, then the "learned" filter should approach the value obtained for the linear system. This does occur and is illustrated in Fig. 6.7 for b = 0.5, 1.0, and 2.0 respectively. For these three systems the sample mean square error Γ and the sample one-step predicted mean square error were calculated for both the filtered and non-filtered estimates of the state x(n). By non-filtered, it is meant that the measurements are used as if they are noise free. This corresponds to an implied filter magnitude of 1.0. The non-filtered one-step prediction of $\underset{\sim}{x}(n+1)$ given z(n) is then determined by f(z(n)). The data for 3,000 iterations are summarized in Table 6.3.

TABLE 6.3

COMPARISON OF THE ADAPTIVELY FILTERED AND NON-FILTERED RESULTS FOR EXAMPLE 6.6

	b = 0.5			b = 1.0			b = 2.0		
Quantity	K	Γ	Σ	K	Γ	Σ	K	Γ	Σ
Non-Filtered	1.0	.855	1.03	1.0	.72	1.12	1.0	.69	1.18
Adaptively Filtered	.387	.529	1.01	.43	.537	1.05	.526	.538	1.07
Optimal Linear	.531	.531	1.13	.531	.531	1.13	.531	.531	1.13

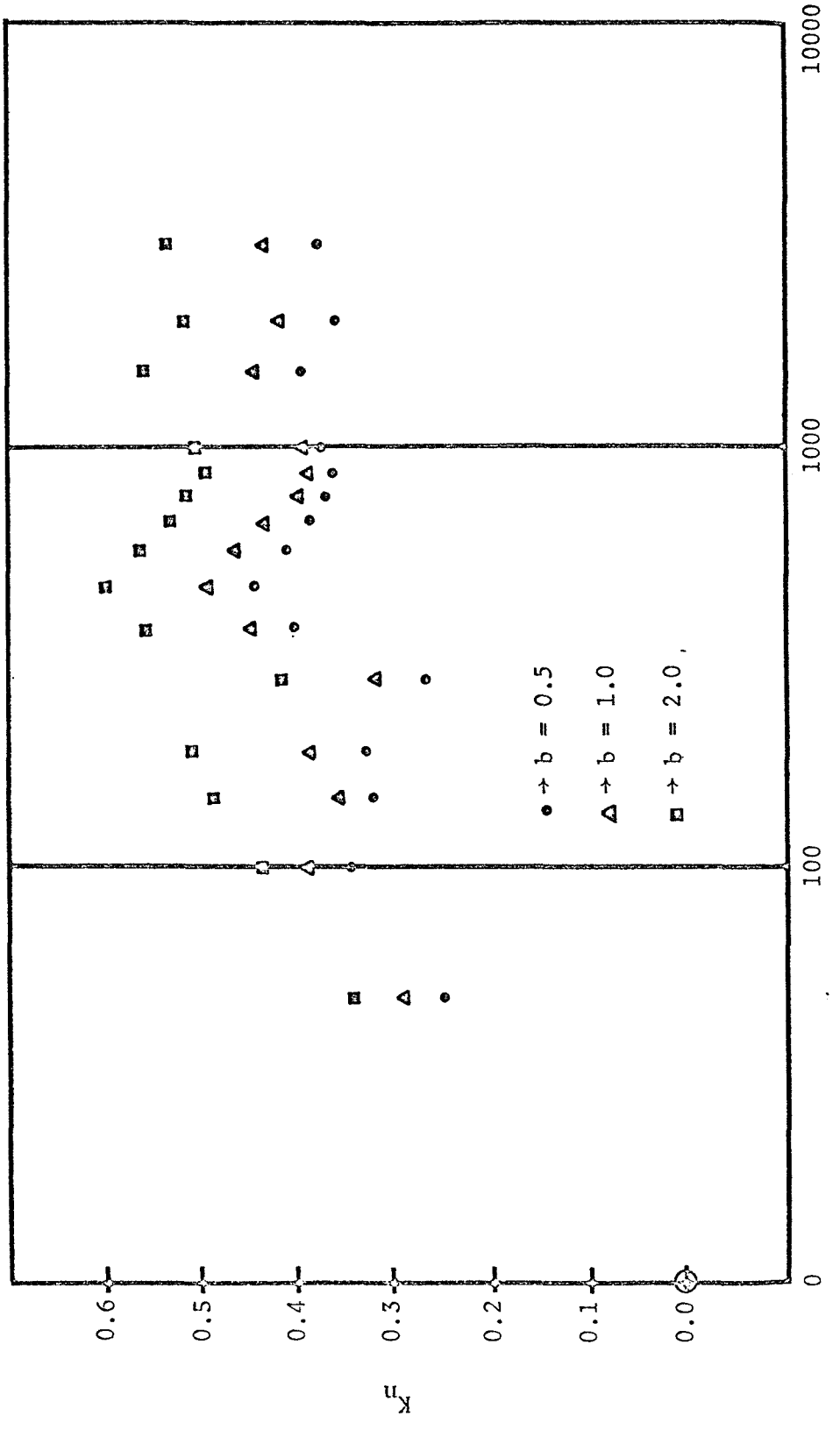


Fig. 6.7 Time Evolution of K_n for Example 6.6

Observe from Table 6.3 that as the saturation threshold b is increased, the value of the adaptively "learned" filter approaches the K_{opt} for the corresponding linear system, i.e., $b = \infty$. Note also that for $b = 0.5$, the mean square error 0.855 associated with the non-filtered estimate is 61.8% larger than that for the adaptively filtered estimate. The computed Γ and Σ for the adaptive filter are also smaller than those for the optimal linear filter case. This is rather interesting. And it is not unexpected since when the system is in saturation, it is not difficult to accurately predict its next state much of the time. The other point worthy of mention is the lower bound on Σ. Since $Q = 1.0$, the absolute lower bound on Σ is 1.0 which is very close to the calculated Σ.

EXAMPLE 6.7: Saturation Nonlinearity

Given: $\phi = 0.5$, $b = 1.0$

The purpose of this example is to examine the effect of changing Q and R while keeping their ratio constant. In a linear system such a change does not alter K_{opt}. In a nonlinear system this is no longer true. Figure 6.8 illustrates this effect for $R = 4Q$ with $Q = 1$ and $Q = 4$. As is to be expected, the larger Q creates a smaller "learned" filter. This follows since the system is operating in the saturated region more often. The experimental calculations after 5000 iterations are listed in Table 6.4. The increase in the non-filtered estimation mean square error Γ over that for adaptive filtering is 343%. By comparing the other quantities in Table 6, it seems that the non-linear adaptive algorithm developed in Section 5.5 does indeed "work" well.

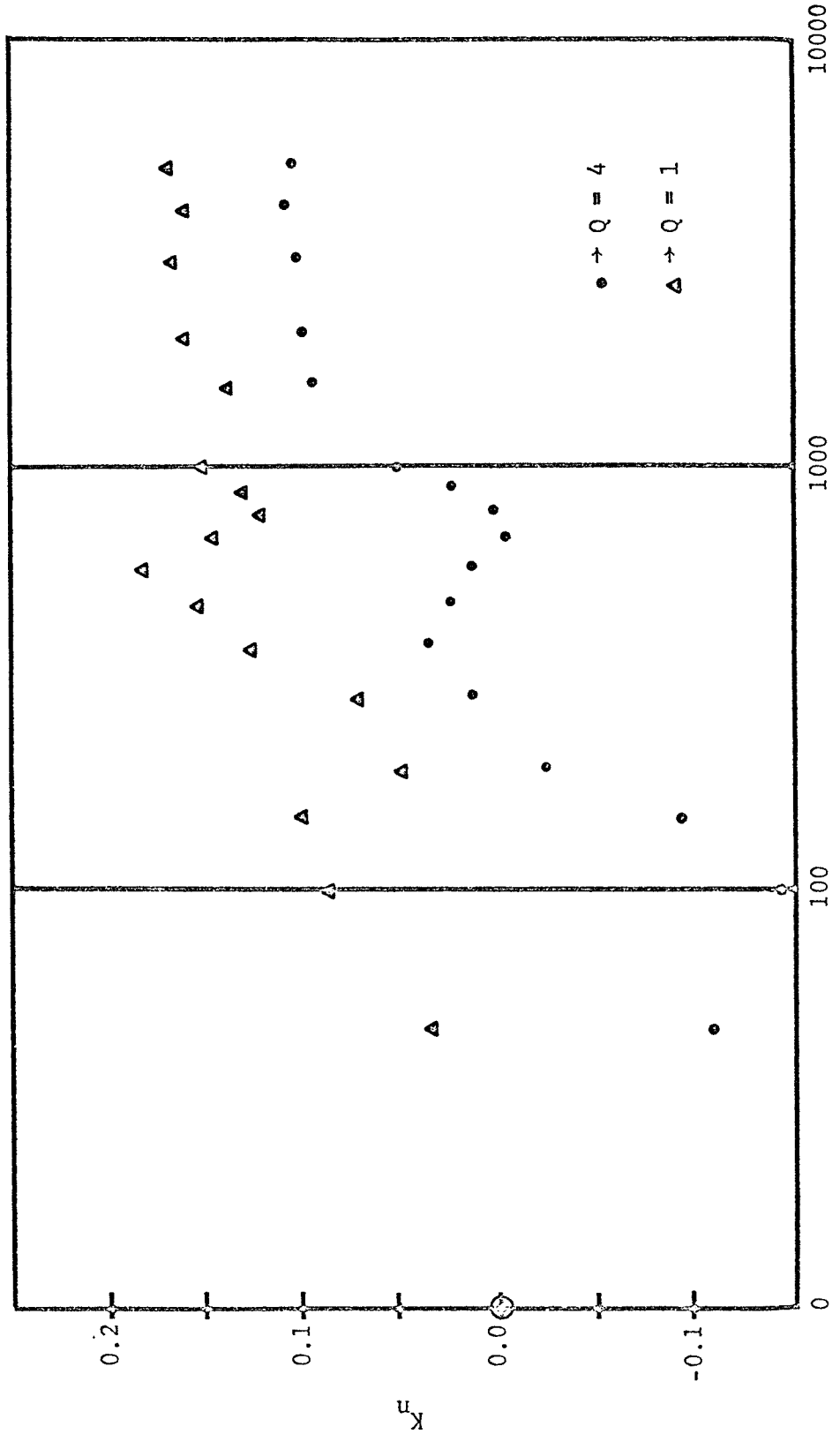


Fig. 6.8 Time Evolution of K_n for Example 6.7

TABLE 6.4

COMPARISON OF THE ADAPTIVELY FILTERED AND NON-FILTERED RESULTS FOR EXAMPLE 6.7

	Q = 1.0			Q = 4.0		
Quantity	K	Γ	Σ	K	Γ	Σ
Non-Filtered	1.0	3.76	1.06	1.0	20.0	4.23
Adaptively Filtered	.165	0.85	1.02	.103	4.52	4.13
Optimal Linear	.236	0.94	1.24	.236	3.76	4.94

EXAMPLE 6.8: Dead Zone Nonlinearity

Given: $\phi = 0.5$, $b = 0.25$

$$f(x) = \begin{cases} \phi(x-b) & \text{for } x > b \\ 0 & \text{for } |x| \leq b \\ -\phi(x-b) & \text{for } x < -b \end{cases} \qquad (5.45)$$

Algorithm (5.43) was programmed for the dead zone nonlinearity of Eq. (5.45). The results for $Q = R = 1.0$ are shown in Fig. 6.9. The sample mean square errors Γ and Σ after 3000 iterations are 0.534 and 1.1 respectively. The absolute lower bound on Σ is 1.0 and 0.531 is the optimal value for Γ in a linear system. Figure 6.9 also contains a plot for $Q = R = 2.0$.

EXAMPLE 6.9: Dead Zone Nonlinearity

Given: $\phi = 0.5$, $b = 0.25$, $Q = 1.0$, $R = 4.0$

The time evolution of K_n for these parameters is illustrated in Fig. 6.10 and a comparison of the adaptively filtered and non-filtered results is given in Table 6.5.

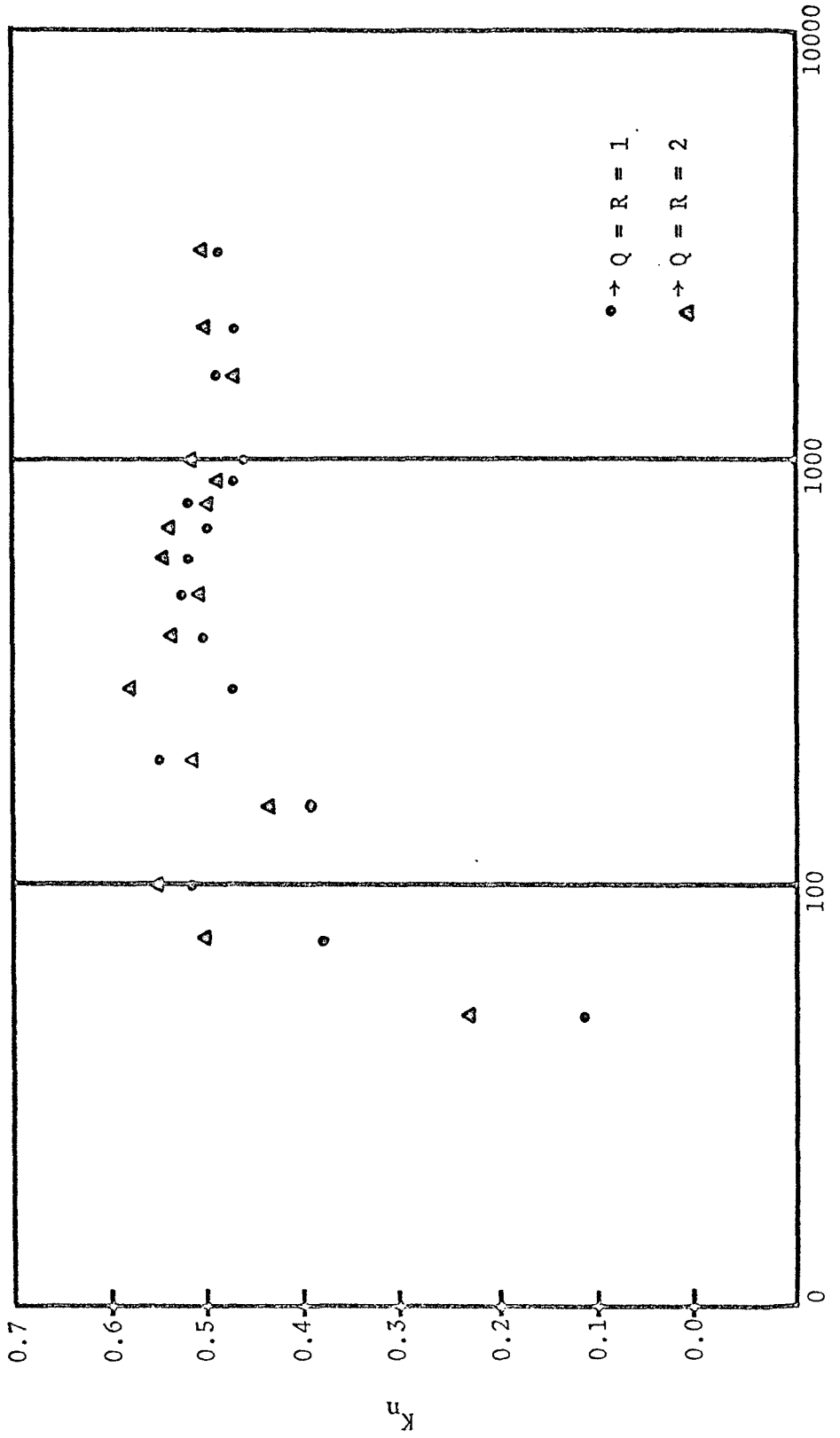


Fig. 6.9 Time Evolution of K_n for Example 6.8

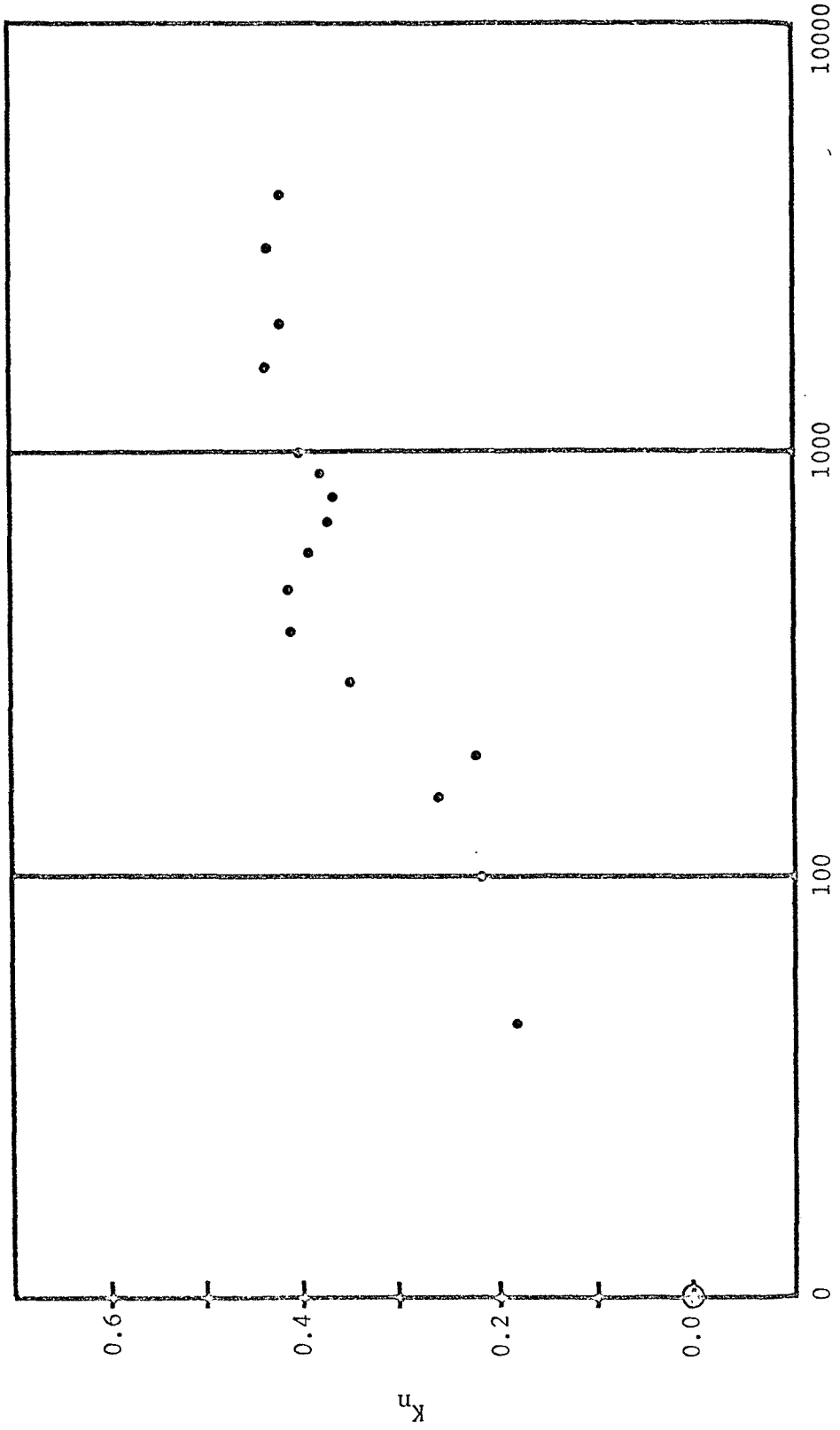


Fig. 6.10 Time Evolution of K_n for Example 6.9

TABLE 6.5

COMPARISON OF THE ADAPTIVELY FILTERED AND NON-FILTERED RESULTS FOR EXAMPLE 6.9.

Quantity	K	Γ	Σ
Non-Filtered	1.0	1.90	1.75
Adaptively Filtered	.41	1.1	1.18

As in the saturation case, the adaptively filtered Γ and Σ are far superior to the non-filtered Γ and Σ. The reduction in Γ is 72.7%. Again the filtered value of Σ approaches its absolute lower bound of 1.0.

6.3.2 Nonlinearities That Have Unique Inverses

Consider the set of nonlinear functions $\{f(x)=y: f^{-1}(y) = x\}$. Then the adaptive stochastic algorithm, which learns the filter that is optimal in the sense of Eq. (5.42), is given by Eq. (5.45).

$$\hat{K}_{n+1} = \hat{K}_n + \frac{1}{n} f^{-1} (\delta_{n+1}\delta_n) W_{n+1} \qquad (5.45)$$

EXAMPLE 6.10: Cubic Nonlinearity

Given: $\phi = 0.1$, $Q = 0.64$, $R = 4.0$

$f(x) = \phi x^3$

The behavior of the adaptive algorithm for $K_o = 0.0$ is shown in Fig. 6.11. The "learned" filter is 0.232. The error covariances after 3,000 iterations are presented in Table 6.6.

TABLE 6.6

COMPARISON OF THE ADAPTIVELY FILTERED AND NON-FILTERED RESULTS FOR EXAMPLE 6.10.

Quantity	K	Γ	Σ
Non-Filtered	1.0	2.70	.881
Adaptively Filtered	.232	.652	.681

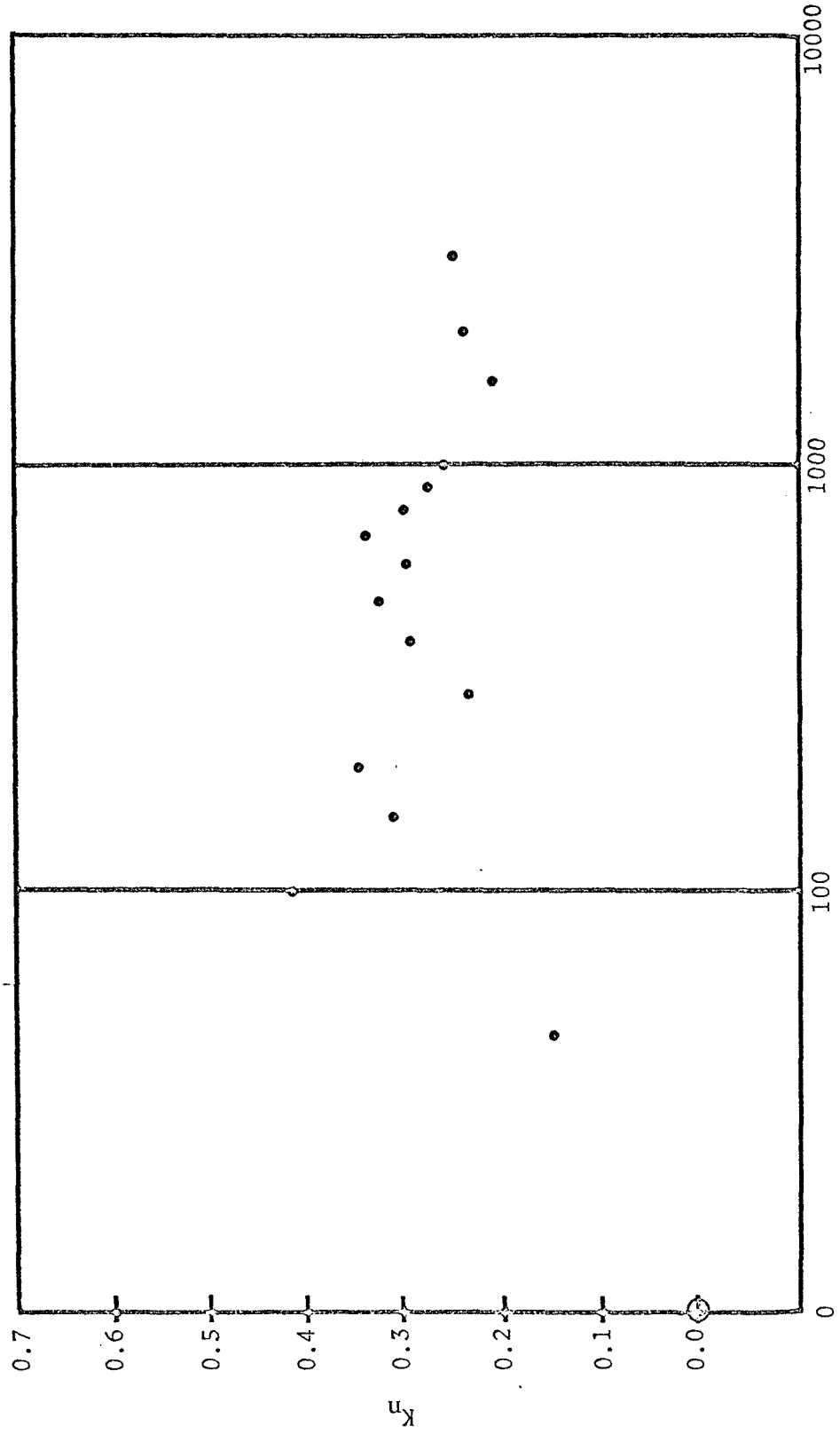


Fig. 6.11 Time Evolution of K_n for Example 6.10

Observe that the filtered Γ is over four times smaller than the non-filtered Γ.

EXAMPLE 6.11: Cubic Nonlinearity

Given: $\phi = 0.1$, $Q - 0.25$, $R = 1.0$

The time evolution of the filter learning process for this problem is illustrated in Fig. 6.12. Its value after 4000 iterations is 0.205. The mean square errors are compared in Table 6.7.

TABLE 6.7

COMPARISON OF THE ADAPTIVELY FILTERED AND NON-FILTERED RESULTS FOR EXAMPLE 6.11

Quantity	K	Γ	Σ
Non-Filtered	1.0	.742	.525
Adaptively Filtered	.205	.231	.252

The absolute lower bound on Σ is 0.25 and the Σ for the adaptively "learned" filter is only 0.252 which is twice as small as that for the non-filtered case. Also the filtered mean square error is 221% smaller than the non-filtered Γ. The conclusion is that the nonlinear algorithm of Eq. (5.45) gives very good results for the examples considered.

6.4 Application to a Thermionic Reactor Model

The thermionic reactor system considered in this section utilizes the linearized prompt-jump approximation to a simplified point reactor kinetics model. It is similar to the model treated in previous annual reports [Brehm, et al., 1968]. The reactor system is described by the following set of differential equations

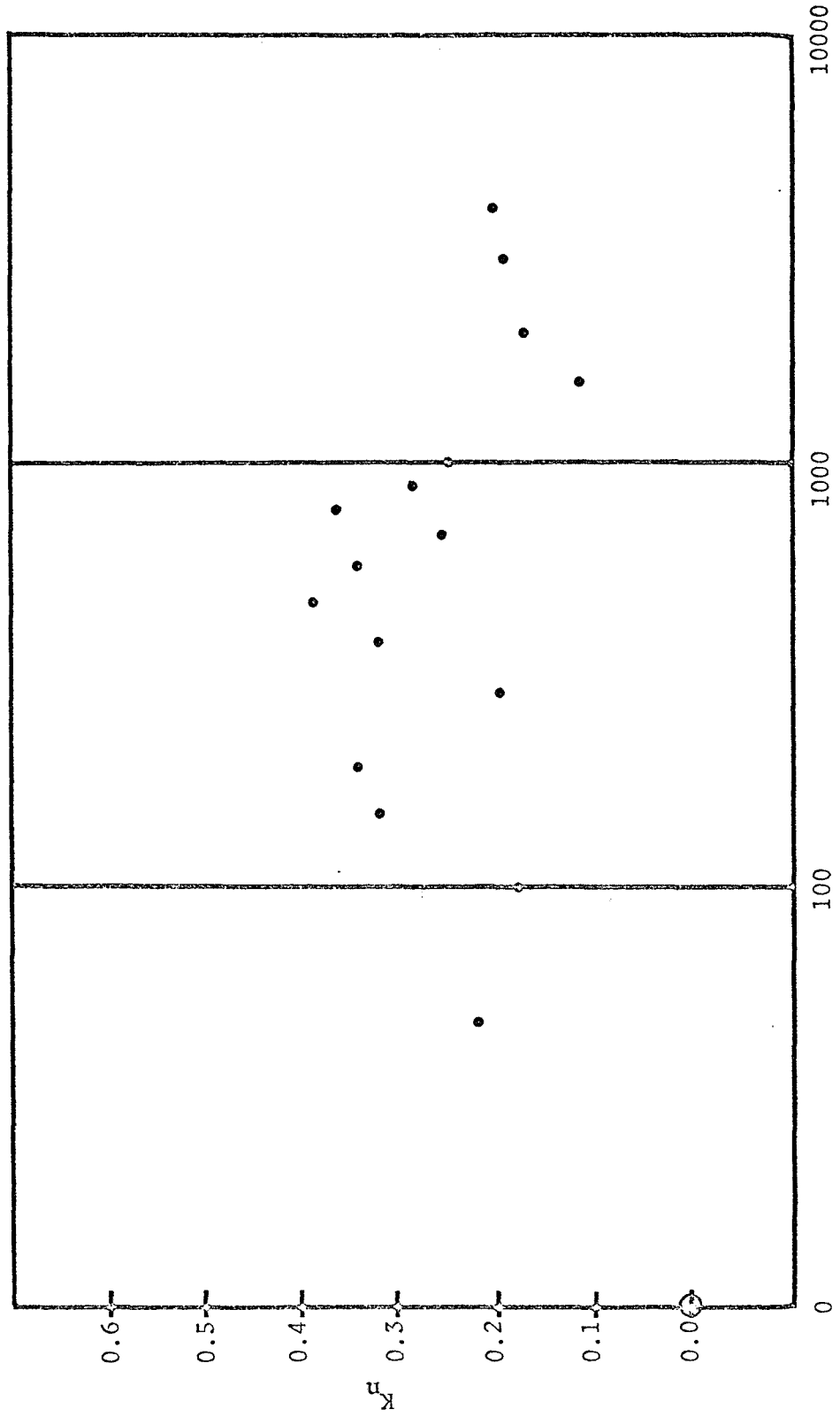


Fig. 6.12 Time Evolution of K_n for Example 6.11

$$\dot{x}_1 = -n_o\xi_1 x_2 - n_o\xi_2 x_3 + u_1(t) + \omega_1(t) \tag{6.4a}$$

$$\dot{x}_2 = bx_1 - \nu_1 x_2 + \omega_2(t) \tag{6.4b}$$

$$\dot{x}_3 = \nu_2 x_2 - \nu_3 x_3 + u_2(t) + \omega_3(t) \tag{6.4c}$$

where

x_1 is the neutron power density state variable

x_2 is the emitter surface temperature state variable

x_3 is the collector surface temperature state variable

$u_1(t)$ is the reactivity control input

$u_2(t)$ is the coolant control input

n_o is the equilibrium neutron power density

ξ_1 and ξ_2 are the reactivity temperature feedback coefficients

b, ν_1, ν_2, ν_3 are the reciprocal system time constants and are explained in detail by Brehm, et al. [1968]

ω_1, ω_2, and ω_3 represent the random plant disturbance inherent in the reactor.

The measurement system for the state variables is described by

$$y_1(t) = h_1 x_1(t) + \nu_1(t) \tag{6.5a}$$

$$y_2(t) = h_2 x_2(t) + \nu_2(t) \tag{6.5b}$$

$$y_3(t) = h_3 x_3(t) + \nu_3(t) \tag{6.5c}$$

where

ν_1, ν_2, and ν_3 represent the observation noise

h_1, h_2, and h_3 are the measurement transformation constants.

By multiplying each of the observation equations by the inverse of the appropriate measurement transformation, the following form is obtained.

$$z_1(t) = x_1(t) + v_1(t) \tag{6.6a}$$

$$z_2(t) = x_2(t) + v_2(t) \tag{6.6b}$$

$$z_3(t) = x_3(t) + v_3(t) \tag{6.6c}$$

Note that since the emitter surface temperature $x_2(t)$ can only be measured indirectly, it will be very noisy in general. By employing standard state variable notation Eqns. (6.4) and (6.6) may be compactly written

$$\dot{\underset{\sim}{x}}(t) = A\underset{\sim}{x}(t) + B\underset{\sim}{u}(t) + \underset{\sim}{\omega}(t) \tag{6.4}$$

$$\underset{\sim}{z}(t) = \underset{\sim}{x}(t) + \underset{\sim}{v}(t) \tag{6.6}$$

where the state, control, and output vectors are respectively

$$\underset{\sim}{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \qquad \underset{\sim}{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \qquad \underset{\sim}{z}(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{bmatrix}$$

and the associated system matrices are

$$A = \begin{bmatrix} 0 & -n_o\xi_1 & -n_o\xi_2 \\ b & -\nu_1 & 0 \\ 0 & -\nu_2 & -\nu_3 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The remaining vectors $\underset{\sim}{\omega}(t)$ and $\underset{\sim}{v}(t)$ represent the plant perturbation noise and measurement noise terms respectively.

By invoking the Separation Theorem, the control term $\underset{\sim}{u}(t)$ may be disregarded in learning the optimal Kalman filter. Thus, Eq. (6.4) becomes

$$\dot{\underset{\sim}{x}}(t) = A\underset{\sim}{x}(t) + \underset{\sim}{\omega}(t) \tag{6.7}$$

To proceed with the development, numerical values must be assigned to the elements of the A matrix. Nominal values for these parameters are

$$b = 0.25 \qquad \xi_1 = 0.5 \qquad \nu_1 = 0.05$$
$$n_o = 80 \qquad \xi_2 = 1.3 \qquad \nu_2 = 0.55$$
$$\nu_3 = 5.00$$

Using these values and a sampling time of 0.01 sec, the one-step state transition matrix equation describing the time evolution of (6.7) is

$$\underset{\sim}{x}(n+1) = \Phi\underset{\sim}{x}(n) + \underset{\sim}{w}(n) \tag{6.8}$$

where

$$\Phi = \begin{bmatrix} 0.998 & -0.39 & -1.02 \\ 0.0025 & 0.985 & -0.004 \\ 0.0 & -0.0052 & 0.95 \end{bmatrix}$$

and

$\underset{\sim}{w}(n)$ is zero mean random perturbation matrix whose covariance kernel is unknown.

The system defined by Eqns. (6.6) and (6.8) where the covariance matrices R and Q of $\underset{\sim}{v}(n)$ and $\underset{\sim}{w}(n)$ respectively are unknown, represents the simulation model of this section. The adaptive process Eqns. (4.39)

and (4.43) for learning the optimal filtering matrix for this system is initialized by setting K_o equal to the null matrix. The experimental behavior of the adaptive algorithms for different values of Q and R was studied. Its success is illustrated by the representative example presented below.

With

$$Q = \begin{bmatrix} 2.14 & 1.73 & 1.39 \\ 1.73 & 2.01 & 1.75 \\ 1.39 & 1.75 & 2.13 \end{bmatrix}$$

and

$$R = \begin{bmatrix} 7.97 & 1.84 & 2.42 \\ 1.84 & 6.06 & -1.08 \\ 2.42 & -1.08 & 1.46 \end{bmatrix}$$

the optimal filter matrix is

$$K_{opt} = \begin{bmatrix} .154 & .133 & .203 \\ -.134 & .384 & .509 \\ -.235 & .219 & .827 \end{bmatrix}$$

The corresponding estimates after 3,000 iterations are

$$\hat{Q} = \begin{bmatrix} 2.08 & 1.60 & 1.35 \\ 1.60 & 2.17 & 1.66 \\ 1.35 & 1.66 & 2.08 \end{bmatrix}, \quad \hat{R} = \begin{bmatrix} 7.86 & 1.87 & 2.30 \\ 1.87 & 6.01 & -1.01 \\ 2.30 & -1.01 & 1.41 \end{bmatrix}$$

$$\hat{K} = \begin{bmatrix} .158 & .115 & .194 \\ -.151 & .368 & .500 \\ -.220 & .211 & .818 \end{bmatrix}$$

The minimum MSE for this thermionic system is 3.90 while the actual MSE resulting from the use of the "learned" filter $\hat{K}$ is 3.98. This means that $\hat{K}$ produces a MSE only 2.05%.

The excellent results presented above demonstrate the success obtained from a formal application of the adaptive learning theory to the state estimation problem in a thermionic reactor. The model used was deliberately simplified because it allowed greater fiscal economy in simulation without seriously degrading the illustrative effects. The assumption that all the state variables are available as noisy measurements can be eliminated by using the algorithm of Section 5.3.3.

6.5 Conclusion

The conclusion drawn from the examples simulated in the chapter is that the adaptive stochastic algorithms derived in Chapter 4 for learning the optimum Kalman filter perform excellently in practice. Many other systems were simulated and the results were of similar quality. Also the predicted acceleration of convergence obtained by using the weighting sequence defined by Eq. (4.39) was verified experimentally.

These same algorithms were modified in Chapter 5 to handle certain classes of nonlinear systems. Again the experimental results were quite good. However, a well-defined reference for performance evaluation does not exist in the nonlinear case.

This chapter concludes the contribution of this dissertation. A short summary of the entire thesis and suggestions for further research are the topics of Chapter 7.

CHAPTER 7

EPILOGUE

7.1 Summary

Kalman filter theory is a complete and powerful solution to the linear stochastic state estimation problem. However, to apply the technique requires complete statistical knowledge of the random environment. Specifically, the plant perturbation covariance Q and the observation noise covariance R must be known *a priori*. In the real world such extensive information is generally not available. To alleviate this difficulty in applying Kalman filter theory, stochastic algorithms are developed which directly learn the optimal stationary discrete time Kalman filter without any knowledge of Q and R.

The structure of the report is as follows. After motivating the subject in Chapter 1, Wiener and Kalman filter theory are reviewed and compared in Chapter 2. This theory provides the foundation for the unsupervised learning criterion of Chapter 3. There the learning criterion is shown to be a necessary and sufficient condition for optimal filtering. In Chapter 4 the adaptive stochastic algorithms, required to accomplish the adaptation indicated by the learning criterion, are derived. The theory of Stochastic Approximation is invoked to prove probability one convergence of the algorithms to the optimum Kalman filter. Additions and extensions to these results are presented in Chapter 5. Methods for estimating the additional quantities necessary

for optimal smoothing are developed. The most important contribution of Chapter 5 is the lifting of the restriction of Chapter 4 requiring that the inverse of the observation matrix exist. The stochastic algorithms are also modified to accommodate slowly time varying statistics and certain classes of nonlinear systems. The experimental results obtained by simulating the algorithm of Chapter 4 and Section 5.5 on a CDC 6400 digital computer are given in Chapter 6. In particular, the theory is successfully applied to the adaptive state estimation problem in a thermionic reactor. This data confirms the theoretical expectations and predictions of the first part of the study.

7.2 Recommendations for Further Study

In this report only sampled data were considered. The author conjectures that analogous results exist for continuous data. This should be a fruitful research topic.

At a lower level it would be interesting to apply the results of Section 5.3.3 to examples where the inverse of the observation matrix does not exist.

Another topic worthy of investigation would be to imbed the nonlinear technique of Section 5.5 into a general control system containing a single nonlinearity such as a cubic or saturating amplifier.

APPENDIX A

LISTING OF THE SIMULATION PROGRAM

The following pages comprise a listing of the computer program developed to simulate the algorithms of Section 6.2.

```
      PROGRAM HAMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      INTEGER ORDER
      REAL    X(10,1),RNSEQW(10,1),RNSEQV(10,1),Z(10,1),LAMBDA,XHAT(10,1)
     1,DELTA1(10,1),DELTA2(10,1),FILTER(10,10),WEIGHT(10,10),V(10,10),
     2 W(10,10),DSTM(10,10),DSTMINV(10,10),DIVISOR ,STORE1(10,10),
     3 DELWGT(1,10)
      DIMENSION Q(10,10), R(10,10), STORE2(10,10), STORE3(10,10)
      EQUIVALENCE (DSTM,STM),(DSTMINV,STMINV)
      REAL STM(10,10),STMINV(10,10)
      DIMENSION TITLE (8)
      COMMON SUM1(10,10),SUM12(10,10),PDELTAK(10,10),DELTAK(10,10) ,
     1CHAT(10,10),SIGMA(10,10),RHAT(10,10),CERROR(10,10),QHAT(10,10)
      REAL IDENTY(10,10)
      DATA IDENTY / 1.0,10*0.0,1.0,10*0.0,1.0,10*0.0,1.0,10*0.0,1.0 ,
     1 10*0.0,1.0,10*0.0,1.0,10*0.0,1.0,10*0.0,1.0,10*0.0,1.0 /
C  READ AND PRINT INPUT DATA
   40 READ(5,50) TITLE,ITER,ORDER,IX,N,RANSET,LAMBDA,IS,ITOT
   50 FORMAT (8A10/4I10,2F10.0,A10,I10)
      IF(EOF,5) 60,70
   60 STOP
   70 WRITE(6,75) TITLE,ITER,ORDER,IX,RANSET,LAMBDA,N
   75 FORMAT(*1INPUT DATA FOR  * 8A10 /// * NUMBER OF  ITERATIONS* I9
     1 ,10X *ORDER OF PROBLEM* I9,15X *PRINT FREQUENCY* I9 //
     2 * RANDOM NUMBER GENERATOR STARTING POINT  *E10.3,10X *LAMBDA*
     3 E11.4,8X  *ITERATIONS OF GNOISE* I8)
      IF(IS.EQ.1H ) GO TO 80
      WRITE(6,78)
   78 FORMAT(/ * FORCE SYMMETRY    YES*)
      GO TO 95
   80 WRITE(6,85)
   85 FORMAT(/ * FORCE SYMMETRY    NO*)
   87 FORMAT(*0   * 10E13.4)
   90 FORMAT(8F10.0)
   95 WRITE(6,97)
   97 FORMAT(/// * STM MATRIX*)
      DO 100 I=1,ORDER
      READ(5,90) (STM(I,J),J=1,ORDER)
  100 WRITE(6,87) (STM(I,J),J=1,ORDER)
      READ(5,90) (X(J),J=1,ORDER)
      WRITE(6,105) (X(J),J=1,ORDER)
  105 FORMAT(/// * X VECTOR * // 3X,10E13.4)
      WRITE(6,110)
  110 FORMAT(/// * FILTER MATRIX*)
      DO 120 I=1,ORDER
      READ(5,90) (FILTER(I,J),J=1,ORDER)
  120 WRITE(6,87) (FILTER(I,J),J=1,ORDER)
      WRITE(6,125)
  125 FORMAT(/// * WEIGHT MATRIX*)
      DO 130 I=1,ORDER
      READ(5,90) (WEIGHT(I,J),J=1,ORDER)
  130 WRITE(6,87) (WEIGHT(I,J),J=1,ORDER)
      WRITE(6,135)
  135 FORMAT(/// * W MATRIX*)
      DO 140 I=1,ORDER
      READ(5,90) (W(I,J),J=1,ORDER)
  140 WRITE(6,87) (W(I,J),J=1,ORDER)
      WRITE(6,145)
```

```
  145 FORMAT(/// ' V MATRIX')
      DO 150 I=1,ORDER
      READ(5,90) (V(I,J),J=1,ORDER)
  150 WRITE(6,67) (V(I,J),J=1,ORDER)
C     DEFINE INITIAL CONDITIONS
      DUMMY=NAME(RANSET)
      DO 210 I=1,ORDER
  210 RNSEQV(I)=GNOISE(N)
      CALL MATRIX(20,ORDER,ORDER,1,V,10,RNSEQV,10,RNSEQV,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQV,10,XHAT,10)
      DO 230 I=1,ORDER
  230 RNSEQW(I)=GNOISE(N)
      CALL MATRIX(20,ORDER,ORDER,1,W,10,RNSEQW,10,RNSEQW,10)
      CALL MATRIX(20,ORDER,ORDER,1,DSTM,10,X,10,X,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQW,10,X,10)
      DO 240 I=1,ORDER
  240 RNSEQV(I)=GNOISE(N)
      CALL MATRIX(20,ORDER,ORDER,1,V,10,RNSEQV,10,RNSEQV,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQV,10,Z,10)
      CALL MATRIX(20,ORDER,ORDER,1,DSTM,10,XHAT,10,XHAT,10)
      CALL MATRIX(22,ORDER,1,0,Z,10,XHAT,10,DELTA1,10)
      CALL MATRIX(20,ORDER,ORDER,1,FILTER,10,DELTA1,10,STORE1,10)
      CALL MATRIX(21,ORDER,1,0,XHAT,10,STORE1,10,XHAT,10)
      DO 270 I=1,ORDER
      DO 270 J=1,ORDER
  270 STMINV(I,J)=STM(I,J)
      CALL MATRIX(10,ORDER,ORDER,0,STMINV,10,DUMMY)
      NOINTS=NOPRINT=0
      IF(ITOT.LE.0) GO TO 300
C  FILTER INCREMENT PROCESS
      DO 2700 I=1,200
 2700 SUM1(I,1)=0.0
      DO 2720 I=1,ITOT
      DO 2710 J=1,ORDER
      RNSEQV(J)=GNOISE(N)
 2710 RNSEQW(J)=GNOISE(N)
      CALL MATRIX(20,ORDER,ORDER,1,V,10,RNSEQV,10,RNSEQV,10)
      CALL MATRIX(20,ORDER,ORDER,1,W,10,RNSEQW,10,RNSEQW,10)
      CALL MATRIX(20,ORDER,ORDER,1,STM,10,X,10,STORE1,10)
      CALL MATRIX(21,ORDER,1,0,STORE1,10,RNSEQW,10,X,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQV,10,Z,10)
      CALL MATRIX(20,ORDER,ORDER,1,STM,10,XHAT,10,STORE1,10)
      CALL MATRIX(22,ORDER,1,0,Z,10,STORE1,10,DELTA2,10)
      CALL MATRIX(0,ORDER,1,0,DELTA1,10,DELWGT,1)
      CALL MATRIX(20,ORDER,1,ORDER,DELTA1,10,DELWGT,1,STORE1,10)
      CALL MATRIX(21,ORDER,ORDER,0,SUM1,10,STORE1,10,SUM1,10)
      CALL MATRIX(20,ORDER,1,ORDER,DELTA2,10,DELWGT,1,STORE1,10)
      CALL MATRIX(21,ORDER,ORDER,0,SUM12,10,STORE1,10,SUM12,10)
      CALL MATRIX(20,ORDER,ORDER,1,FILTER,10,DELTA2,10,STORE1,10)
      CALL MATRIX(20,ORDER,ORDER,1,STM,10,XHAT,10,XHAT,10)
      CALL MATRIX(21,ORDER,1,0,XHAT,10,STORE1,10,XHAT,10)
 2720 CALL MATRIX(1,ORDER,1,0,DELTA2,10,DELTA1,10)
      CALL MATRIX(1,ORDER,ORDER,0,SUM1,10,WEIGHT,10)
      CALL MATRIX(10,ORDER,ORDER,0,WEIGHT,10,DUMMY)
      CALL MATRIX(20,ORDER,ORDER,ORDER,SUM12,10,WEIGHT,10,FDELTAK,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,STMINV,10,FDELTAK,10,DELTAK,10)
      CALL MATRIX(21,ORDER,ORDER,0,FILTER,10,DELTAK,10,FILTER,10)
      WRITE(6,2730) ITOT
```

```
 2730 FORMAT(*1RESULTS OF OPTIONAL FILTER INCREMENT,* I7 * ITERATIONS*
     1 /// * SUM1 MATRIX* )
      DO 2740 I=1,ORDER
 2740 WRITE(6,87) (SUM1(I,J),J=1,ORDER)
      WRITE(6,2750)
 2750 FORMAT(// * SUM12 MATRIX*)
      DO 2760 I=1,ORDER
 2760 WRITE(6,87) (SUM12(I,J),J=1,ORDER)
      WRITE(6,480)
      DO 2770 I=1,ORDER
 2770 WRITE(6,87) (WEIGHT(I,J),J=1,ORDER)
      WRITE(6,2780)
 2780 FORMAT(// * DELTAK MATRIX*)
      DO 2790 I=1,ORDER
 2790 WRITE(6,87) (DELTAK(I,J),J=1,ORDER)
      WRITE(6,460)
      DO 3000 I=1,ORDER
 3000 WRITE(6,87) (FILTER(I,J),J=1,ORDER)
C  FILTER ADAPTATION PROCESS
  300 DO 320 I=1,ORDER
      RNSEQV(I)=GNOISE(N)
  320 RNSEQW(I)=GNOISE(N)
      CALL MATRIX(20,ORDER,ORDER,1,V,10,RNSEQV,10,RNSEQV,10)
      CALL MATRIX(20,ORDER,ORDER,1,W,10,RNSEQW,10,RNSEQW,10)
      CALL MATRIX(20,ORDER,ORDER,1,DSTM,10,X,10,X,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQW,10,X,10)
      CALL MATRIX(21,ORDER,1,0,X,10,RNSEQV,10,Z,10)
      CALL MATRIX(20,ORDER,ORDER,1,DSTM,10,XHAT,10,STORE1,10)
      CALL MATRIX(22,ORDER,1,0,Z,10,STORE1,10,DELTA2,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,DSTM,10,FILTER,10,FILTER,10)
      CALL MATRIX(23,ORDER,1,ORDER,DELTA1,10,WEIGHT,10,DELWGT,1)
      CALL MATRIX(20,ORDER,1,ORDER,DELTA2,10,DELWGT,1,STORE1,10)
      CALL MATRIX(21,ORDER,ORDER,0,FILTER,10,STORE1,10,FILTER,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,DSTMINV,10,FILTER,10,FILTER,10)
      IF(IS.EQ.1H ) GO TO 350
      CALL MATRIX(0,ORDER,ORDER,0,FILTER,10,STORE1,10)
      DO 340 I=1,ORDER
      DO 340 J=1,ORDER
  340 FILTER(I,J)=(FILTER(I,J)+STORE1(I,J))/2.0
  350 CALL MATRIX(20,1,ORDER,1,DELWGT,1,DELTA1,10,STORE1,10)
      DIVISOR=STORE1(1,1)+LAMBDA
      CALL MATRIX(20,ORDER,1,ORDER,DELTA1,10,DELWGT,1,STORE1,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,WEIGHT,10,STORE1,10,STORE1,10)
      DL=LAMBDA*DIVISOR
      DO 375 I=1,ORDER
      DO 370 J=1,ORDER
  370 WEIGHT(I,J)=WEIGHT(I,J)/LAMBDA-STORE1(I,J)/DL
  375 DELTA1(I)=DELTA2(I)
      CALL MATRIX(20,ORDER,ORDER,1,DSTM,10,XHAT,10,XHAT,10)
      CALL MATRIX(20,ORDER,ORDER,1,FILTER,10,DELTA2,10,STORE1,10)
      CALL MATRIX(21,ORDER,1,0,XHAT,10,STORE1,10,XHAT,10)
      NOINTS=NOINTS+1
      NOPRINT=NOPRINT+1
      IF(NOINTS.LT.ITER.AND.NOPRINT.LT.IX) GO TO 300
      WRITE(6,410) NOINTS,DIVISOR,TITLE,(DELTA1(I),I=1,ORDER)
  410 FORMAT(*1ITERATION* I8,8X *DIVISOR =* E12.4,8X,8A10 ///
     1 * DELTA1 VECTOR * // 10E13.4)
      WRITE(6,430) (X(I),I=1,ORDER)
```

```
  430 FORMAT(// * X VECTOR * // 10E13.4)
      WRITE(6,440) (Z(I),I=1,ORDER)
  440 FORMAT(// * Z VECTOR * // 10E13.4)
      WRITE(6,450) (XHAT(I),I=1,ORDER)
  450 FORMAT(// * XHAT VECTOR * // 10E13.4)
      WRITE(6,460)
  460 FORMAT(// * FILTER MATRIX *)
      DO 470 I=1,ORDER
  470 WRITE(6,87) (FILTER(I,J),J=1,ORDER)
      WRITE(6,480)
  480 FORMAT(// * WEIGHT MATRIX *)
      DO 490 I=1,ORDER
  490 WRITE(6,87) (WEIGHT(I,J),J=1,ORDER)
      NOPRINT=0
      IF(NOINTS.LT.ITER) GO TO 300
C FINAL CALCULATIONS
      DIV=ITOT+ITER
      CALL MATRIX(10,ORDER,ORDER,0,WEIGHT,10,DUMMY)
      DO 5000 I=1,ORDER
      DO 5000 J=1,ORDER
 5000 CHAT(I,J)=WEIGHT(I,J)/DIV
      CALL MATRIX(20,ORDER,ORDER,ORDER,FILTER,10,CHAT,10,SIGMA,10)
      CALL MATRIX(22,ORDER,ORDER,0,CHAT,10,SIGMA,10,RHAT,10)
      CALL MATRIX(22,ORDER,ORDER,0,IDENTY,10,FILTER,10,STORE1,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,STORE1,10,SIGMA,10,CERROR,10)
      CALL MATRIX(0,ORDER,ORDER,0,STM,10,STORE1,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,CERROR,10,STORE1,10,QHAT,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,STM,10,QHAT,10,STORE1,10)
      CALL MATRIX(22,ORDER,ORDER,0,SIGMA,10,STORE1,10,QHAT,10)
      WRITE(6,5010) TITLE
 5010 FORMAT(*1FINAL CALCULATIONS * 30X,8A10 /// * CHAT MATRIX * )
      DO 5020 I=1,ORDER
 5020 WRITE(6,87) (CHAT(I,J),J=1,ORDER)
      WRITE(6,5030)
 5030 FORMAT(// * SIGMA MATRIX *)
      CALL MATRIX(0,ORDER,ORDER,0,SIGMA,10,STORE1,10)
      DO 1001 I=1,ORDER
      DO 1001 J=1,ORDER
 1001 SIGMA(I,J) =(SIGMA(I,J) + STORE1(I,J)) / 2.0
      DO 5040 I=1,ORDER
 5040 WRITE(6,87) (SIGMA(I,J),J=1,ORDER)
      WRITE(6,5050)
 5050 FORMAT(// * CERROR MATRIX *)
      CALL MATRIX(0,ORDER,ORDER,0,CERROR,10,STORE1,10)
      DO 1002 I=1,ORDER
      DO 1002 J=1,ORDER
 1002 CERROR(I,J) = (CERROR(I,J)+STORE1(I,J)) /2.0
      DO 5060 I=1,ORDER
 5060 WRITE(6,87) (CERROR(I,J),J=1,ORDER)
      WRITE(6,5070)
 5070 FORMAT(// * RHAT MATRIX *)
      CALL MATRIX (0,ORDER,ORDER,0,RHAT,10,STORE1,10)
      DO 1003 I=1,ORDER
      DO 1003 J=1,ORDER
 1003 RHAT(I,J) = (RHAT(I,J) + STORE1(I,J)) / 2.0
      DO 5080 I=1,ORDER
 5080 WRITE(6,87) (RHAT(I,J),J=1,ORDER)
      WRITE(6,5090)
```

```
 5090 FORMAT(// * QHAT MATRIX *)
      CALL MATRIX (0,ORDER,ORDER,0,QHAT,10,STORE1,10)
      DO 1004 I=1,ORDER
      DO 1004 J=1,ORDER
 1004 QHAT(I,J) =(QHAT(I,J)+STORE1(I,J)) /2.0
      DO 5100 I=1,ORDER
 5100 WRITE(6,87) (QHAT(I,J),J=1,ORDER)
      DO 1010 I=1,ORDER
      DO 1010 J=1,ORDER
      STORE1(J,I) = W(I,J)
 1010 STORE2(J,I) = V(I,J)
      CALL MATRIX (20,ORDER,ORDER,ORDER,W,10,STORE1,10,Q,10)
      CALL MATRIX (20,ORDER,ORDER,ORDER,V,10,STORE2,10,R,10)
      PRINT 1011
 1011 FORMAT (// * Q MATRIX *)
      DO 1012 I=1,ORDER
 1012 WRITE (6,87) (Q(I,J),J=1,ORDER )
      PRINT 1013
 1013 FORMAT (// * R MATRIX * )
      DO 1014 I=1,ORDER
C
C         CALCULATE CLAMBDA
 1014 WRITE(6,87) (R(I,J),J=1,ORDER)
      DO 1020 I=1,ORDER
      DO 1020 J=1,ORDER
      STORE1(J,I) = 0.0
      IF (I.EQ.J) STORE1(J,I) = 1.0
 1020 CONTINUE
      DO 1021 I=1,ORDER
      DO 1021 J=1,ORDER
 1021 STORE2(I,J) = STORE1(I,J) - FILTER(I,J)
      CALL MATRIX (20,ORDER,ORDER,ORDER,STORE2,10,SIGMA,10,STORE1,10)
      DO   1023 I=1,ORDER
      DO   1023 J=1,ORDER
 1023 STORE3(J,I) =STORE2(I,J)
      CALL MATRIX(20,ORDER,ORDER,ORDER,STORE1,10,STORE3,10,STORE2,10)
      CALL MATRIX(20,ORDER,ORDER,ORDER,FILTER,10,R,10,STORE1,10)
      DO 1024 I=1,ORDER
      DO 1024 J=1,ORDER
 1024 STORE3(I,J) = FILTER(J,I)
      CALL MATRIX (20,ORDER,ORDER,ORDER,STORE1,10,STORE3,10,STORE1,10)
      DO 1025 I=1,ORDER
      DO 1025 J=1,ORDER
 1025 STORE2(J,I) = STORE2(J,I) + STORE1(J,1)
      PRINT 1026
 1026 FORMAT (// * CLAMBDA MATRIX *)
      DO 1027 I=1,ORDER
C
C  CALCULATE NSIGMA
 1027 WRITE(6,87) (STORE2(I,J),J=1,ORDER)
      CALL MATRIX (20,ORDER,ORDER,ORDER,STM,10,STORE2,10,STORE1,10)
      DO 1030 I=1,ORDER
      DO 1030 J=1,ORDER
 1030 STORE3(J,I) = STM(I,J)
      CALL MATRIX(20,ORDER,ORDER,ORDER,STORE1,10,STORE3,10,STORE2,10)
      DO 1035 I=1,ORDER
      DO 1035 J=1,ORDER
 1035 STORE2(J,I)  =  STORE2(J,I) + Q(J,I)
```

```
      WRITE(6,1036)
1036  FORMAT (// * HSIGMA MATRIX* )
      DO 1038 I=1,ORDER
1038  WRITE(6,87)   (STORE2(I,J),J=1,ORDER)
      GO TO 40
      END
```

APPENDIX B

TONELLI'S THEOREM

THEOREM B.1:

Given the relation

$$x(n,v) = E_w \{ \sum_{j=-\infty}^{n} y(n,j,v,w) \}$$

then the interchange of the expectation and summation

$$x(n,v) = \sum_{j=-\infty}^{n} E_w \{y(n,j,v,w)\}$$

is valid if either

(i) $E_w \{\sum_j |y(n,j,v,w)|\} < \infty$

(ii) $\sum_{j=-\infty}^{n} E_w \{|y(n,j,v,w)|\} < \infty$

This is a special case of Tonelli's Theorem [see for example Royden, 1967, p. 234].

LIST OF REFERENCES

Albert, A. A. and L. A. Gardner. Stochastic Approximation and Nonlinear Regression, M.I.T. Press, Cambridge, Massachusetts, 1967.

Balakrishnan, A. V. "Report on Progress in Information Theory in the U.S.A., 1960-1963: Prediction and Filtering," IEEE Trans. on Information Theory, IT-9 (October 1963), pp. 237-239.

Brehm, R. L., D. L. Hetrick, J. G. Guppy, K. D. Kerns, and T. R. Schmidt. "Dynamics of a Fast Reactor with In-Core Thermionic Converters, Second Annual Report," Engineering Experimental Station, University of Arizona, Tucson, Arizona, 1968.

Bryson, A. E., and D. E. Johansen. "Linear Filtering for Time-Varying Systems Using Measurements Containing Colored Noise," IEEE Trans. on Automatic Control, AC-10 (January 1965), pp. 4-10.

Bryson, A. E., and R. K. Mehra. "Linear Smoothing Using Measurements Containing Correlated Noise with an Application to Inertial Navigation," IEEE Trans. on Automatic Control, AC-13 (October 1968), pp. 496-503.

Deutsch, R. Estimation Theory, Prentice-Hall, Englewood Cliffs, N. J., 1965.

Doob, J. L. Stochastic Processes, John Wiley and Sons, New York, 1963.

Dvoretzky. A. "On Stochastic Approximation," Third Berkeley Symposium on Math. Statistics and Probability, 1956.

Fu, K. S. Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

Gladyshev, E. G. "On Stochastic Approximation," Theory of Probability and Its Applications, Vol. X, No. 2 (1965), pp. 275-278.

Hampton, R. L. T. Stochastic Approximation and Its Engineering Applications, Engineering Experiment Station, The University of Arizona, December, 1967.

Hancock, J. C. and P. A. Wintz. Signal Detection Theory, McGraw-Hill, New York, 1966.

Heffes, H. "The Effect of Erroneous Models on the Kalman Filter Response," IEEE trans. on Automatic Control, AC-11 (July 1966), pp. 541-543.

Hilborn, C. G., and D. G. Lainiotis. "Optimal Estimation in the Presence of Unknown Parameters," IEEE Trans. on System Science and Cybernetics, SSC-5 (January 1969) pp. 38-43.

Ho, Y. C. and C. Blaydon. "On the Abstraction Problem in Pattern Classification," Proceedings of the 1966 National Electronics Conference, (1966), pp. 857-862.

Joseph, P. D., and J. Tou. "On Linear Control Theory," AIEE Trans. on Applications and Industry, 80 (1961), pp. 193-196.

Kailath, T. "Adaptive Matched Filters," Chapter 6 in Mathematical Optimization Techniques, Richard Bellman, Ed., University of California Press, Los Angeles, 1963.

Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," J. Basic Engr., 82 D (1960), pp. 33-45.

Kalman, R. E., and R. S. Bucy. "New Results in Linear Filtering and Prediction Theory," J. Basic Engr., 83 D (1961), pp. 95-108.

Kolmogorov, A. N. "Interpolation and Extrapolation von Stationaren Zufalligne Holgen," Bull. Acad. Sci. (USSR), Ser. Math., 5 (1941), pp. 3-14.

Korn, G. A. Random-Process Simulation and Measurement, McGraw-Hill, New York, 1966.

Lee, R. C. K. Optimal Estimation, Identification and Control, M.I.T. Press, Cambridge, Mass., 1964.

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

Meditch, J. S. "Orthogonal Projection and Discrete Optimal Linear Smoothing," S.I.A.M. Journal on Control, Vol. 5, 1967.

_____. Stochastic Optimal Linear Estimation and Control, McGraw-Hill, New York, 1969.

Motyka, P. R. and J. A. Cadzow. "The Factorization of Discrete Process Spectral Matrices," Proceedings of the National Electronics Conference, 23 (1967), pp. 66-73.

Nishimura, T. "On the A Priori Information in Sequential Estimation Problems," IEEE Trans. on Automatic Control, AC-11 (April 1966), pp. 197-204.

_____. "Error Bounds of Continuous Kalman Filters and the Application to Orbit Determination Problems," IEEE Trans. on Automatic Control, AC-12 (June 1967), pp. 268-275.

Robbins, H., and S. Monro. "A Stochastic Approximation Method," Ann. Math. Stat., Vol. 22, No. 1, 1951.

Royden, H. L., Real Analysis, Macmillan, New York, 1963.

Sage, A. P. Optimum Systems Control, Prentice-Hall, Englewood Cliffs, N. J., 1968.

Schultz, D. G., and J. L. Melsa. State Functions and Linear Control Systems, McGraw-Hill, New York, 1967.

Shellenbarger, J. C. "Estimation of Covariance Parameters for an Adaptive Kalman Filter," Proceedings of the 1966 N.E.C., (1966) pp. 698-702.

_____. "A Multivariance Learning Technique for Improved Dynamic System Performance," Proceedings of the 1967 N.E.C., (1967), pp. 146-151.

Sherman, S. "A Theorem on Convex Sets With Applications," Ann. Math. Stat., 26 (1955), pp. 763-767.

_____. "Non-Mean-Square Error Criteria," IRE Trans. on Information Theory, IT-4 (September 1958), pp. 125-126.

Soong, T. T. "On A Priori Statistics in Minimum Variance Estimation Problems," J. Basic Engineering, 87 D (1965), pp. 109-112.

Sorenson, H. W. "Kalman Filtering Techniques," Chapter 6 in Advances in Control Systems, Vol. 3, C. T. Leondes, Ed., Academic Press, New York, 1966.

Stumper, F. L. "A Bibliography of Information Theory, Communication and Cybernetics," IRE Trans. on Information Theory, IT-1 (September 1955), pp. 35-37.

Van Trees, H. L. Detection, Estimation and Modulation Theory, John Wiley and Sons, New York, 1968.

Viterbi, A. J. Principles of Coherent Communication, McGraw-Hill, New York, 1966.

Wiener, N. The Extrapolation, Interpolation, and Smoothing of Stationary Time Series, John Wiley and Sons, New York, 1949.