Interim Report

RESEARCH IN AND APPLICATION OF MODERN AUTOMATIC CONTROL THEORY TO NUCLEAR ROCKET DYNAMICS AND CONTROL

Prepared under Grant ~~NsG-490~~ NGL-03-002-006
National Aeronautics and Space Administration

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

Interim Report

RESEARCH IN AND APPLICATION OF
MODERN AUTOMATIC CONTROL THEORY
TO NUCLEAR ROCKET DYNAMICS AND CONTROL

Prepared under Grant ~~NsG-490~~
National Aeronautics and Space Administration
(Donald G. Schultz, Principal Investigator)

by

Robert L. T. Hampton
Cliff A. Harris
C. H. Lewis

Nuclear and Electrical Engineering Departments
The University of Arizona
Tucson, Arizona 85721

January 1970

Engineering Experiment Station
The University of Arizona
College of Engineering
Tucson, Arizona

Table of Contents

INTRODUCTION AND OUTLINE OF REPORT

This report summarizes the work that has been done on the NASA Research Grant NsG-490 since the last reporting date of July, 1969. The report is divided into four separate parts titled:

Part I: Stochastic Approximation and its Engineering Applications

Part II: Stochastic Algorithms for Self-Adaptive Filtering and Prediction

Part III: The Control of Nonlinear Stochastic Control Systems Under Discounted Performance Criteria

Part IV: Linear Stability of a Nuclear Rocket Engine With Two Reactivity Feedbacks

Each of the separate parts includes its own index, bibliography, and pagination, and the content of each is discussed briefly in this introduction. The report has been labeled an interim report because work in three of the four areas is not complete, as discussed below.

Of the four parts, Part I on "Stochastic Approximation and Its Engineering Applications," is the only portion of this report that may be considered complete. This is a tutorial treatment of the subject of stochastic approximation that emphasizes the algorithmic approach to optimization in the presence of uncertainty or noise. The uncertainty or noise may arise from basic ignorance of the underlying phenomena, experimental errors, or inherent random fluctuations. The nomenclature Stochastic Approximation is used to emphasize the stochastic nature of the errors in, say, the process measurements, and the use of these measurements (past and present) to calculate the approximate location of the optimum or goal. Of particular importance is the fact that the use of the stochastic approximation algorithms assumes no a priori knowledge of the noise statistics that are involved in the optimization problem at hand. This is an important practical consideration.

The stochastic approximation theory that is described in Part I is the basis for Part II,"Stochastic Algorithms for Self Adaptive Filtering and Prediction." The basic goal of this research is to develop a self-adaptive solution to the problem of optimal filtering, prediction, and detection of stochastic signals imbedded in random noise. In particular, the random noise is considered to be unknown. This is in contrast to theories of Weiner and Kalman which require a complete knowledge of the covariance matrices of both the plant and observation noises. Rarely are such complete descriptions available, and, in fact, the requirement that the noise covariance matrix be nonsingular has often resulted in unwarranted assumptions as to the nature of the components of the noise involved. In this report, 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 the plant and noise covariance matrices as an intermediate step and the need to solve the entire set of filtering equations. The problem associated with the need for the nonsingular measurement noise covariance matrix is thus elminated or rather bypassed by using this alternate approach. It is shown that the stochastic algorithms developed for estimating the optimum filter converge in a mean square sense with probability one. The results are valid for scalar and vector values for signal and noise processes. It is expected that the research described in Part II of this report will be completed by July 1, 1970, and a more complete and final report will be issued at that time.

Part III on "The Control of Nonlinear Stochastic Control Systems under Discounted Performance Criteria," is similar to Part II in that it presents the theoretical basis for a Ph.D. dissertation. As in Part II, the system dynamics are modeled with difference equations, and the goal is to obtain a practical algorithmic approach. Here, however, the problem is one of determining the optimal control, rather than one of obtaining optimum estimates, as above. The approach is through the use of dynamic programming in a partitioned state space, where the advantage to be gained over a conventional dynamic programming approach is largely a computational one. The discounting factor in the performance criteria, β, is required to insure convergence.

The format of this presentation is largely one of theorem, lemma, and proof, with only two relatively simple examples. There seems to be real hope, however, that the optimal control methods developed in Part III may well be applicable to the control of the restartable nuclear rocket engine, where the discounting factor β may be related to failure probability, and the noise or variation in the system may be considered as due to changing parameters within the system, in particular the degeneration of the core. More will be said on this and other applications in the annual report due in July.

The last portion of this report, Part IV, is concerned with "Linear Stability of a Nuclear Rocket Engine with Two Reactivity Feedbacks." As the title implies, this research applies directly to the nuclear rocket work. The object of this investigation is to define stability boundaries for the nonlinear reactor in a number of parameter space. Much of the work was done by simulation on an analogue computer for the linear system in preparation for attacking the nonlinear model. An attempt to simulate the nonlinear equations on the analogue computer proved unreliable, due to complexities involved. Work will continue in this area, with simulation to be done on the Electrical Engineering Department's hybrid facility.

PART I

STOCHASTIC APPROXIMATION AND ITS ENGINEERING APPLICATIONS

by

R. L. T. Hampton

December 1969

TABLE OF CONTENTS

I. INTRODUCTION

The purpose of this paper is to provide an up-to-date investigation of the methods of Stochastic Approximation and its application to the Information Sciences. The discussion will attempt to give physical and intuitive meaning to the mathematical conditions of Stochastic Approximation rather than to reproduce rigorous proofs, which can be found in the referenced literature.

1.1 Definition of Stochastic Approximation

Stochastic Approximation is essentially an algorithmic technique for optimization in the presence of uncertainty. This uncertainty or noise may arise from basic ignorance of the underlying phenomena, experimental errors or inherent random fluctuations. The nomenclature Stochastic Approximation is used to emphasize the stochastic nature of the errors in, say, the process measurements, and the use of these measurements (past and present) to calculate the approximate location of the optimum or goal. In addition, no a priori knowledge of the noise statistics is required in Stochastic Approximation methods. Such stochastic problems are naturally more difficult than deterministic problems. However, algorithmic search techniques, whether concerned with random errors or not, involve two fundamental considerations:

(1) selecting a direction in which to move,

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

1.2 Effect of Random Error on Convergence[1,2]

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 (stability) takes priority over speed of convergence optimization in a stochastic environment. In Stochastic Approximation, this effect is reflected in the choice of step sizes, consideration (2) above. The direction to move, consideration (1), is selected as if the process were deterministic. That is the experimental observations are assumed to be error free. This means that some step directions may be incorrect, but such set-backs are swamped-out in the long run by additional data if the step sizes are properly selected. Note this is nothing more than a modified statement of the law of large numbers.

1.3 Intuitive Selection of the Step Size

The following statement by Poisson of the empirical law of large numbers sheds commonsense insight on the method of Stochastic Approximation.

> In many different fields, empirical phenomena appear to obey a certain general law, which can be called the Law of Large Numbers. This law states that the ratios of numbers derived from the observation of a very large number of similar events remain practically constant, provided that these events are governed partly by constant factors and partly by variable factors whose variations are irregular and do not cause a systematic change in a definite direction. Certain values of these relations are characteristic of each given kind of event. With the increase in length of the series of observations, the ratios derived from such observations come closer and closer to these characteristic constants. They could be expected to reproduce them exactly if it were possible to make series of observations of an infinite length.[3]

It is upon this ~~experientail~~ experiential truth that Stochastic Approximation methods, as well as all applications of probability theory, are based. Intuitively, then, one knows that if the present estimate (method) is backed by extensive observations (experience), then it should not be significantly altered by new data. The converse is true for an estimate based on relatively few noisy observations. A simple illustration of this is the sample mean. For example, let

$$y = M + \epsilon$$

where y = measurable information

ϵ = experimental error (unbiased)

and M = desired constant.

Then after n observations $(y_1, \ldots, y_n)$, the best estimate of M is

$$M_n = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$$= (1 - \frac{1}{n})\, M_{n-1} + \frac{1}{n}\, y_n$$

$$= M_{n-1} + \frac{1}{n}\, (y_n - M_{n-1}) \qquad (1)$$

where M_{n-1} = "old" estimate

and y_n = new datum point

Note that M_{n-1} is weighted by $(1 - \frac{1}{n})$, whereas, y_n is weighted in inverse proportion to the number of observations, which approaches zero as n approaches infinity. However, this defense of the status quo is no longer valid if there exist changes in the process. In such a case, an adaptive weighting technique must be devised. This will be further discussed in a later section. Regardless, the method of weighting new data in proportion to 1/n is of fundamental importance in determining the step size in Stochastic Approximation. This is because

$$\lim_{n\to\infty} \frac{1}{n} = 0$$

Therefore, if the step sizes are decreased according to the harmonic sequence $\{\frac{1}{n}\}$, the corrections approach zero in the limit. This property is necessary for convergence. Second,

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty$$

This property of the harmonic sequence guarantees that the correction process will not stop short of the optimum point regardless of the initial estimate, i.e., the sequence has an infinite amount of corrective effort. Third,

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

or, equivalently

$$\sum_{n=N}^{\infty} (\frac{1}{n})^2 \rightarrow 0 \text{ as } N \rightarrow \infty.$$

This property ensures that the cumulative effect of the noise error variance remains finite.[2] Why this is so will be explained in section 2.4.

In Chapter II, specific Stochastic Approximation methods will be reviewed, and it will be shown that equation (1) is actually a Stochastic Approximation algorithm.

II. METHODS OF STOCHASTIC APPROXIMATION

Historically, two basic types of Stochastic Approximation were developed. The first was the Robbins-Monro (R-M) procedure[4] for finding the unique root of an unknown regression function and the second was the Kiefer-Wolfowitz (K-W) procedure for finding the maximum of an unknown unimodal regression function.[5] Dvoretzky unified and generalized these earlier studies.[6] Detailed reviews of the above results and their variations may be found in Derman,[7] Schmetterer,[8] and Venter.[21]

2.1 Robbins-Monro Method

The R-M algorithm is the exact stochastic analog of a simple deterministic algorithm for solving

$$M(x) = k \tag{2}$$

where $M:R^1 \to R^1$

and k is any real number.

The deterministic algorithm is

$$x_{n+1} = x_n + a_n \,[k - M(x_n)] \tag{3}$$

where a_n is sequence of real numbers which must satisfy certain conditions to ensure convergence (see Ref. 9).

When there is random error present, M(x) cannot be measured, but a noisy observation y(x) of M(x) can be made. Now, however, y(x) is a random variable with a distribution function $F(y|x)$ defined such that

$$E\{y|x\} = \int_{-\infty}^{\infty} y(x)\ d\ F(y|x) = M(x) \qquad \text{for all } x \qquad (4)$$

Thus M(x) is the regression function of y on x. The problem is still to iteratively solve equation (2), but equation (3) is no longer meaningful, regardless of whether $F(y|x)$ is known or not, since M(x) is not observable. Under these circumstances, a stochastic version of equation (3) is defined

$$x_{n+1} = x_n + a_n\ [k - y(x_n)] \qquad (5)$$

where $\{x_n\}$ is now a sequence of nonstationary random variables which converges in some stochastic sense to the solution of (2).

2.1.1 Convergence

Robbins and Monro proved that the algorithm (5) converges in mean-square to the correct solution, say $\hat{x}$, of (2) if the sequence $\{a_n\}$ satisfies the three conditions

$$\text{(a) } \lim_{n\to\infty} a_n = 0 \qquad \text{(b) } \sum_{n=1}^{\infty} a_n = \infty \qquad \text{(c) } \sum_{n=1}^{\infty} a_n^{\ 2} < \infty$$

and the regression function M(x) can be bounded on either side of the solution $\hat{x}$ by a straight line.[2]

(d) $\left|M(x)\right| \leq a\left|x - \hat{x}\right| + b \qquad (a, b > 0)$

(e) $E\{\left|M(x) - y(x)\right|^2\} = \sigma^2(x) < \sigma^2 < \infty \qquad \text{for all } x$

The physical meaning of the conditions on $\{a_n\}$ is exactly the same as stated in section 1.3 where a_n is interpreted as the step size. Note that the harmonic sequence $\{1/n\}$ not only satisfies (a), (b), and (c), but also gives the fastest possible reduction of the step size without violating any of the conditions; that is, for any sequence $\{\frac{1}{n^\alpha}\}$

$$\sum_{n=1}^{\infty} \frac{1}{n^\alpha} < \infty \qquad \text{for } \alpha > 1$$

Condition (d) is necessary to prevent an overshoot of $\hat{x}$ that cannot be corrected by a sequence $\{a_n\}$ satisfying (a), (b), and (c). Condition (e) is required for the obvious reason that if the variance of the measurements is not finite for all values of x, then it would be impossible to guarantee conversion of the algorithm in general.

Blum[10] and Kallianpur[11] established independently that the above conditions are sufficient for convergence with probability 1 of algorithm (5). As in Ref. 2, the statement is often made in the literature on Stochastic Approximation that probability one convergence implies mean square convergence. This is not so. However, mean-square convergence does imply convergence with probability one under certain conditions (see Ref. 36), but not in general.

2.1.2 Root Finding and Extremum

To use the R-M technique for finding the unique zero of $M(x)$, one simply lets $k = 0$ in equation (4) giving

$$x_{n+1} = x_n - a_n \, y(x_n) \quad . \qquad (6)$$

If $M(x)$ has multiple roots, then there is no a priori way to know to which one equation (6) will converge. Starting from the same initial estimate x_o, (6) may converge to a different zero of $M(x)$ each time the iteration process is run. This effect is a result of the noise in $y(x_n)$.

The R-M method can be made to search for the unique extremum of $M(x)$ with no inflection points, by simply searching for the root of $M'(x)$. If feasible, this is the most effective Stochastic Approximation procedure for extremum searching, i.e., it gives a faster convergence rate than the K-W method. This approach is difficult because it must be assumed that

1. $M(x)$ is everywhere differentiable.

2. $M'(x) = \frac{d}{dx} E\{y(x) \big| x\} = E\{\frac{d}{dx} y(x) \big| x\}$.[12]

and the measurements of $\frac{d}{dx} y(x)$ will generally be extremely noisy. These problems lead naturally to the K-W procedure, which will be discussed in section 2.2 after a comparison of the rate of step-size reduction for the R-M algorithm and a deterministic algorithm.

2.1.3 Stochastic Vs. Deterministic Step-Size Reduction

Such a comparison is informative because it will make salient the effect of noise on the rate of step-size reduction. For simplicity, the deterministic Bolzaro procedure will be used. It successively halves the step sizes,

$$\frac{|x_{n+1} - x_n|}{|x_n - x_{n-1}|} = \frac{1}{2}$$

For comparison with the R-M algorithm, it is necessary to use expected values since

$$|x_{n+1} - x_n| = |a_n\, y\, (x_n)|$$

depends on the noise in the particular measurement $y(x_n)$. Therefore

$$\frac{E\,|x_{n+1} - x_n|}{E\,|x_n - x_{n-1}|} = \frac{a_n}{a_{n-1}}\;\frac{E\,|y(x_n)|}{E\,|y(x_{n-1})|}$$

$$= \frac{a_n\, M(x_n)}{a_{n-1}\, M(x_{n-1})}$$

$$= \frac{n-1}{n}\;\frac{M(x_n)}{M(x_{n-1})}$$

where the harmonic sequence has been used for $\{a_n\}$. Still, the form of $M(x)$ must be known. First, let it be constant for all x, then

$$\frac{E|x_{n+1} - x_n|}{E|x_n - x_{n-1}|} = \frac{n-1}{n} = 1 - \frac{1}{n} \qquad (7)$$

Now use the other extreme $M(x) = Ax$ where A is large. Then

$$\frac{E|x_{n+1} - x_n|}{E|x_n - x_{n-1}|} = \frac{(n-1)\ \cancel{A}x_n}{n\ \cancel{A}x_{n-1}} = \frac{(n-1)\ x_n}{n\ x_{n-1}}$$

but now $x_n = x_{n-1} - a_{n-1}\ M(x_n) = x_{n-1} - \frac{Ax_{n-1}}{n-1}$.

Therefore, the ratio reduces to

$$\frac{E|x_{n+1} - x_n|}{E|x_n - x_{n-1}|} = \frac{n-1}{n}\ [1 - \frac{A}{n-1}] = 1 - \frac{A+1}{n} \ (8)$$

In both cases,

$$\lim_{n\to\infty} \frac{E|x_{n+1} - x_n|}{E|x_n - x_{n-1}|} = 1$$

Thus the noise makes it impossible to decrease the steps as rapidly as in the deterministic case, especially late in the search.[2] Equation (7) can be obtained from (8) by letting $A = 0$. Equation (8) is incorrect in Ref. 2.

2.2 Kiefer-Wolfowitz Method

Following the idea of Robbins and Monro, Kiefer and Wolfowitz constructed an algorithm for finding the extremum (maximum or minimum) of an unknown unimodal regression function M(x). Their process is the exact stochastic analog of the deterministic iteration procedure first formulated by Germansky;[13] his procedure was essentially a form of steepest descent,

$$x_{n+1} = x_n - a_n M'(x_n) \qquad (8)$$

where a_n is chosen ∋ $M(x_{n+1}) < M(x_n)$.

When $M'(x_n)$ is unknown and/or noisy, it is necessary to approximate it in some way. The K-W technique uses two measurements of the observable function y(x) at $(x_n + c_n)$ and $(x_n - c_n)$ to obtain an average slope

$$\frac{y(x_n + c_n) - y(x_n - c_n)}{2\ c_n}$$

which is used as the approximation to $M'(x_n)$. This gives the K-W algorithm analogous to (8)

$$x_{n+1} = x_n - a_n \left[\frac{y(x_n + c_n) - y(x_n - c_n)}{2\ c_n}\right] \qquad (9)$$

This iteration process converges in mean-square and with probability 1 to the minimum of M(x), say $\hat{x}$, if

(a) both the step size a_n and the distance between measurements approach zero

$$\lim_{n\to\infty} a_n = 0 \qquad \lim_{n\to\infty} c_n = 0$$

(b) As in the R-M method, $\sum_{n=1}^{\infty} a_n < \infty$ to assure that the correction does not stop short of the minimum $\hat{x}$.

(c) $\sum_{n=1}^{\infty} (\frac{a_n}{c_n})^2 < \infty$ so that the random effects will tend to offset one another in the long run.

(d) To prevent excessive over-correction, a restriction similar to condition (d) of the R-M process is required. It is

$$\left|M(x_2) - M(x_1)\right| < a\left|x_2 - \hat{x}\right| + b < \infty$$

i.e., the average slope of M(x) for any pair of measurements can be bounded by a straight line.

Since even the function $M(x) = \exp(x^2)$ satisfies requirement (d), it is not a severe restriction.[2]

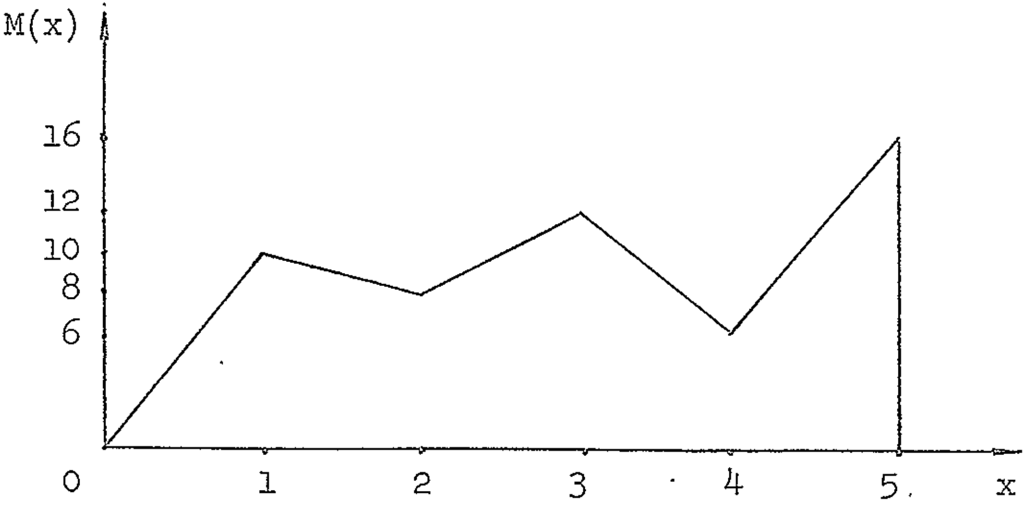
Even though the K-W algorithm is designed for unimodal functions, it is interesting to examine its behavior on a multiple peak M(x).[14] Such an example will also illustrate how to use the algorithm. The example is shown in Fig. 1. Here the maximum of $M(x)$ will be sought so the minus sign in equation (9) must be changed to a plus sign giving

$$x_{n+1} = x_n + \frac{1}{n} \left[\frac{y(x_n + \sqrt[3]{1/n}) - y(x_n - \sqrt[3]{1/n}}{\sqrt[3]{1/n}} \right] \quad (10)$$

where the sequences $a_n = 1/n$ and $c_n = \sqrt[3]{1/n}$ have been chosen to satisfy conditions (a), (b), and (c). The noise in this problem is additive Gaussian with zero mean and unity variance. Therefore, $y(x) = M(x) + e$. The process is started by arbitrarily selecting $x_1 \epsilon [0,5]$. Measurements of $y(x)$ are then made at $(x_1 + 1)$ and $(x_1 - 1)$ as shown in Fig. 1, where the results for three different starting points are illustrated. The convergence to the absolute maximum is rapid for an initial x of 0.25 and 2.50, but for an initial $x = 3.00$, the iteration converges to the local extremum $M(x) = 12.00$. Because of the noise, however, it is not possible to predict to which extremum of a multipeak regression function that the algorithm will converge, even if the same initial point is used each time.

2.3 Generalized Process of Dvoretzky

The basic notion of Stochastic Approximation is that for any deterministic algorithm, there exists a stochastic counterpart, i.e., an algorithm where uncertainty



n	x_n	c_n	$M(x_n+c_n)$	$M(x_n-c_n)$	$y(x_n+c_n)$	$y(x_n-c_n)$
1	.25	1	9.5	0	9.22	0
2	5.00	.79	16.0	8.1	15.74	6.99
3	5.00	.69	16.0	9.1	16.67	6.53
4	5.00	etc.				
End						
1	2.5	1	9.00	9.00	8.72	8.44
2	2.78	.79	10.28	8.02	10.02	6.91
3	4.02	.69	13.10	10.00	13.74	7.43
4	5.00	etc.				
End						
1	3.00	1	6	8.00	5.72	7.44
2	1.72	.79	10.02	9.30	9.76	8.19
3	2.35	.69	11.76	8.68	12.40	6.11
4	3.79	.63	10.20	11.04	10.57	11.07
5	3.58	.58	8.5	11.88	11.25	11.63
6.	3.39	etc.				
.						
.						
.						
50	3.00	Converges to the local peak @ x = 3				

Figure 1

is present in some form. Following this idea, Dvoretzky formulated a generalized Stochastic Approximation method consisting of a deterministic algorithm T with a superimposed random component e,

$$x_{n+1} = T_n(x_1, \ldots, x_n) + e_n \qquad (11)$$

where T_n, $n \geq$ is a sequence of Borel-measurable mappings from R^n (n-dimensional Euclidean space) into R^1. Dvoretzky proved the following theorem for this process.[6]

Theorem 1. Let α_n, β_n, γ_n be non-negative functions from R^n into R^1 $\ni$

$\lim_{n\to\infty} \alpha_n(x_1, \ldots, x_n) = 0$ uniformly $\forall$ sequences $x_1, x_2, \ldots$

$\sum_{n=1}^{\infty} \beta_n(x_1, \ldots, x_n)$ converges uniformly $\forall$ sequences $x_1, x_2, \ldots$

$\sum_{n=1}^{\infty} \gamma_n(x_1, \ldots, x_n)$ diverges to ∞ uniformly $\forall$ sequences $x_1, x_2, \ldots$

Further, let $\hat{x}$ be a real number $\ni$

$$\left|T_n(x_1, \ldots, x_n) - \hat{x}\right| \leq \max\{\alpha_n, [(1 + \beta_n)\left|x_n - \hat{x}\right| - \gamma_n]\} \qquad (12)$$

$\forall (x_1, \ldots, x_n) \epsilon R^n$. Also require $\sum_{n=1}^{\infty} E\{e_n{}^2\} < \infty$ (13)

and $E\{e_n | x_1, \ldots, x_n\} = 0$ with probability one. Then x_n as defined in equation (11) converges in mean square and with probability 1 to $\hat{x}$,

$$\lim_{n\to\infty} E\left|x_n - \hat{x}\right|^2 = 0, \; P\{\lim_{n\to\infty} x_n = \hat{x}\} = 1.$$

It is important to point out two features of this powerful theorem.

(a) Since T_n may be a function of all the observations $(x_1, \ldots, x_n)$, the correction may be based on all past measurements, instead of just on the latest measurement x_n as in the R-M and K-W methods.

(b) The sequences $\{\alpha_n\}$, $\{\beta_n\}$, and $\{\gamma_n\}$ can depend on the measurements $(x_1, \ldots, x_n)$.

For example, these properties make it possible to devise a stochastic Newton-Rapson method or a sequential least squares estimator based on the last m observations, $m \leq n$. The resulting accelerated convergence is obtained at the expense of computational simplicity.

As another illustration of the versatility of this theorem, consider the sample mean given by equation (1),

$$M_n = M_{n-1} + \frac{1}{n}(y_n - M_{n-1}) \tag{1}$$

where M is the unknown mean. By defining the noise-free algorithm to be

$$T_n = (1 - a_n)\, M_{n-1} + a_n M, \; \gamma_n = a_n = \frac{1}{n}$$

and the superimposed random component to be

$$e_n = a_n(y_n - M),$$

then $$M_n = T_n + e_n \tag{1a}$$

and $\left|T_n - M\right| = (1 - a_n) \cdot \left|M_{n-1} - M\right|$ is a special case of (12) and e_n satisfies (13). Therefore, (1) is a special case of Stochastic Approximation, which implies

$$\lim_{n\to\infty} E\,\{M_n - M)^2\} = 0 \quad \text{and} \quad P\{\lim_{n\to\infty} M_n - M\} = 1.$$

Even though simple, this example is important because it contains the idea of estimation of an unknown, but constant system parameter. If the parameter is also time varying, then γ_n can be made dependent on the last $m \leq n$ measurements. The result is an adaptive parameter estimator, e.g., see section 4.5.

2.4 The K-W and R-M Methods as Special Cases of Dvoretzky's Process

By defining T_n and e_n as

$$T_n = x_n + \frac{a_n}{2\, c_n}\,[M(x_n + c_n) - M(x_n - c_n)] \tag{9a}$$

$$e_n = \frac{a_n}{2\, c_n}\,[y(x_n + c_n) - M(x_n + c_n) - y(x_n - c_n) + M(x_n - c_n)], \tag{9b}$$

then $x_{n+1} = T_n + e_n$ gives the K-W algorithm, equation (9).

Similarly, choosing T_n and e_n as

$$T_n = x_n + a_n \, [k - M(x_n)] \tag{5a}$$

$$e_n = a_n \, [M(x_n) - y(x_n)], \tag{5b}$$

then $x_{n+1} = T_n + e_n$ gives the R-M algorithm, equation (5).

At this point, the necessity of requiring that the sum of the observation variances be finite (equation (13) of Dvoretzky's Theorem) will be explained. For simplicity, let e_n be unbiased so that the total measurement variance is $\sum_{n=1}^{\infty} E\{e_n^2\} = \sigma_e^2$.

If $\sigma_e^2 < \infty$, $\sum_{n=N+1}^{\infty} E\{e_n^2\} = \sigma_e^2 - \sum_{n=1}^{N} E\{e_n^2\}$ which is just the variance for the measurements remaining after N trials. Since $\lim_{N\to\infty} \sum_{n=1}^{N} E\{e_n^2\} = \sigma_e^2$, $\lim_{N\to\infty} \sum_{n=N+1}^{\infty} \{e_n^2\} = 0$.

Therefore, the variance of the error approaches zero as the number of observations increases to infinity. This property is obviously necessary for convergence and holds only if σ_e^2 is finite. It is the selection of the step sizes that ensures this condition; for example, in the R-M process

$$\sigma_e^2 = \sum_{n=1}^{\infty} E\{e_n^2\} = \sum_{n=1}^{\infty} E\{a_n^2[M(x_n) - y(x_n)]^2\}$$

$$= \sum_{n=1}^{\infty} a_n^{\ 2} \ E\{[M(x_n) - y(x_n)]^2\} = \sum_{n=1}^{\infty} a_n^{\ 2} \ \sigma^2(x_n)$$

$$\leq \sigma^2 \sum_{n=1}^{\infty} a_n^{\ 2}, \text{ where } \sigma^2 \geq \sigma^2 \geq \sigma^2 \ (x_n), \ \forall_n$$

$$< \infty \text{ iffi } \sum_{n=1}^{\infty} a_n^{\ 2} < \infty.$$

Hence the requirement that $\sum_{n=1}^{\infty} a_n^{\ 2} < \infty$ in the R-M process and similarly for $\sum_{n=1}^{\infty} (\frac{a_n}{c_n})^2 < \infty$ in the K-W process.

III. OTHER PROPERTIES AND EXTENSIONS OF STOCHASTIC APPROXIMATION METHODS

In this chapter the following topics will be discussed:

1. multidimensional Stochastic Approximation algorithms
2. generalized regression functions
3. asymptotic distribution of the estimates

3.1 Multidimensional Stochastic Approximation Algorithms

The multidimensional R-M and K-W methods were introduced first by Blum, who used a Lyapunov type approach to prove convergence with probability one of the two methods. For the R-M process the algorithm is

$$\underline{x}_{n+1} = \underline{x}_n + a_n\ [\underline{K} - \underline{y}(\underline{x}_n)] \qquad (14)$$

where $\forall\ \underline{K},\ \underline{x} \epsilon R^m\ \exists$ an m-dimensional random variable $\underline{y}(\underline{x})\ \ni$

$$E\{\underline{y}|\underline{x}\} = E\{\underline{y}(\underline{x})\} = \underline{M}(\underline{x}) \text{ exists.}$$

Thus, the problem is to find the solution to $\underline{M}(\underline{x}) = \underline{K}$. To guarantee convergence, the sequence $\{a_n\}$ must satisfy the conditions previously specified, and there must exist a Lyapunov type function $V(\underline{x})$ such that

$$V(\underline{x}) > 0, \; \forall \; \underline{x}$$

and

$$(\nabla_{\underline{x}} V(\underline{x}), \; \underline{M}(x)) = \sum_{i=1}^{m} \frac{\partial V(\underline{x})}{\partial x_i} M_i(\underline{x}) \leq 0, \quad \forall x$$

Fortunately, this convergence can also be established by an extension of Dvoretzky's Theorem by simply replacing absolute values $|x_n - \hat{x}|$ with the norms $||\underline{x}_n - \hat{\underline{x}}||$. This avoids the search for a suitable Lyapunov function. If a positive definite m x m matrix R^{-1} is inserted in equation (14) giving

$$\underline{x}_n = \underline{x}_{n-1} + a_n \; R \; [\underline{K} - \underline{y}(\underline{x}(n))] \tag{15}$$

the convergence is not affected. And, if one knows $E\{\underline{y} \; \underline{y}^T\}$, then choosing $R = E\{\underline{y} \; \underline{y}^T\}^{-1}$ decreases the variance of the estimates. Note that there is still no assumption on the noise structure. Later a recursive scheme for calculating R will be developed.

The multidimensional K-W algorithm can take several forms, but in each case difference approximations are needed for every component of $\nabla_{\underline{x}} \underline{M}(\underline{x})$. Blum's method requires m+1 observations at the points

$$\begin{array}{lcl} \underline{x}_n^o = \underline{x}_n & & \underline{y}(\underline{x}_n) = y(x_n^o) \\ \underline{x}_n^1 = \underline{x}_n + c_n \underline{e}_1 & \Longrightarrow & \underline{y}(\underline{x}_n + c_n \underline{e}_1) = y(x_n^1) \\ \vdots & & \vdots \qquad\qquad \vdots \\ \underline{x}_n^m = \underline{x}_n + c_n \underline{e}_m & & \underline{y}(x_n + c_n \underline{e}_m) = y(x_n^m) \end{array}$$

where $\underline{e}_j$ denotes a unit vector whose j element is 1. The measurements $\underline{y}(\underline{x}_n^j)$ determine the <u>one</u>-<u>sided</u> derivative approximations $\Delta\, \underline{y}_n$ where

$$\Delta\, \underline{y}_n = \frac{[\underline{y}(\underline{x}_n^1) - y(\underline{x}_n^o),\ \ldots,\ \underline{y}(\underline{x}_n^m) - \underline{y}(\underline{x}_n^o)]}{c_n}$$

Then the recursive relation

$$\underline{x}_{n+1} = \underline{x}_n - a_n\, \Delta\, \underline{y}_n \tag{16}$$

converges with probability one to the minimum of $M(\underline{x})$. However, Sacks has shown that the asymmetric observations about $\underline{x}_n$ cause slow convergence to the correct $\underline{\hat{x}}$.[16]

Based on the extension of Dvoretsky's Theorem, Gray proved that the symmetric version of (16) converges with probability one and in mean square.[17] It is

$$\underline{x}_{n+1} = \underline{x}_n - a_n\, \Delta\, \underline{y}_n \tag{17}$$

where

$$\Delta\, \underline{y}_n = [\underline{y}(\underline{x}_n^1) - \underline{y}(\underline{x}_n^{-1}),\ \ldots,\ \underline{y}(\underline{x}_n^m) - \underline{y}(\underline{x}_n^{-m})]/2\ c_n$$

and $\underline{x}_n^j = \underline{x}_n + c_n\underline{e}_j$ and $\underline{x}_n^{-j} = \underline{x}_n - c_n\underline{e}_j$

$j = 1, \ldots, m$

This algorithm converges faster than the previous one, but requires 2 m observations (2 for each dimension) instead of m+1.

3.2 Generalized Regression Functions

The purpose of this section is to lay the foundation for the interpretation of the regression function as a performance index so that the methods of Stochastic Approximation can be applied to communication and control systems. Since Stochastic Approximation methods are applicable to any problem that can be formulated as one of regression, the extension is not difficult.

First, assume the observations are from a random process y(t) and there exists a function $\ell(.,.)$ which depends on y(t) and a parameter vector $\underline{k}$. The performance function $\ell(y,\underline{k})$ determines the performance index $L(\underline{k})$ defined as a regression function

$$L(\underline{k}) = E_y\{\ell(y,\underline{k})\} \qquad (18)$$

$$= \int_{-\infty}^{\infty} \ell(y,\underline{k})d\ F(y)$$

It is desired to minimize $L(\underline{k})$ by selecting the optimum $\underline{k} = \hat{\underline{k}}$. If $L(\underline{k})$ is a convex function of $\underline{k}$, then $\hat{\underline{k}}$ is given by

$$\nabla_{\underline{k}} L(\underline{k}) = \underline{0} \qquad (19)$$

If $L(\underline{k})$ is known, equation (19) can be solved iteratively by the gradient method giving[18]

$$\underline{k}_{n+1} = \underline{k}_n - a_n \nabla_{\underline{k}} L(\underline{k}_n) \qquad (20)$$

where $\underline{k}_{n+1}$ converges to $\hat{\underline{k}}$.

Note that deterministic problems can be put in this format by letting the density function f(y) be a delta function. However, when the distribution function F(y) is not given a priori, L($\underline{k}$) is not known. This condition is precisely the motivation for Stochastic Approximation techniques. Thus if $\ell(y,\underline{k})$ is differentiable, the R-M method gives

$$\underline{k}_{n+1} = k_n - a_n \nabla_{\underline{k}} \ell(y_{n+1}, \underline{k}_n) \tag{21}$$

as an iterative solution to equation (19). When $\ell(y,\underline{k})$ is not differentiable, the K-W method gives

$$\underline{k}_{n+1} = \underline{k}_n - a_n \Delta \ell_n \tag{22}$$

where $\Delta \underline{\ell}_n$ is the vector whose jth component is

$$\Delta \ell_n^j = \frac{[\ell(y_{n+1}, \underline{k}_n+c_n\underline{e}_j) - \ell(y_{n+1}, \underline{k}_n-c_n\underline{e}_j)]}{2\ c_n}$$

$j = 1, \ldots, m$

Algorithms (21) and (22) can be shown to converge in mean square and with probability one (see Refs. 6, 12, 15, and 17) for most problems in engineering application. The most restrictive requirement is that $\ell(\cdot,\underline{k})$ have a unique extremum.

Note the similarity between these stochastic algorithms and the deterministic algorithm (20). However, since $\nabla_{\underline{k}}\ell(y,\underline{k})$ in equation (21) or its approximation in equation (22) depends on a single realization of the random process y(t) which may contain noise, $\underline{k}_n$ is a non-stationary random vector.

In the important special case where the performance index is the mean-square error, algorithm (22) reduces to algorithm (21).[12] For example, let

(1) x(t) be a noise corrupted signal (the noise is not required to be additive),

(2) S(t) be the desired signal,

(3) $\sum_{i=1}^{m} k_i F_i(t) = \hat{S}(t,\underline{k})$ be the estimate of S(t) where the k_j are the adjustable parameters that weight the outputs of the filters $F_j(t)$ such that the minimum mean-square error is obtained. This form is general since if a sufficiently large number of $F_j(t)$ are used, the overall filter can approximate arbitrarily closely any nonlinear operator.[19]

(4) The error $e(t,\underline{k}) = S(t) - S(t,\underline{k})$

(5) The performance function $\ell(e(t),\underline{k})$ $= \ell(e(t,\underline{k})) = e^2(t,\underline{k})$

(6) Thus $L(\underline{k}) = E\{e^2(t,\underline{k})\}$

For discrete values of t, algorithm (22) can be used to minimize $L(\underline{k})$, where the jth component of $\Delta\, \underline{\ell}_n$ becomes

$$\Delta\, \ell_n^j = (2\ c_n)^{-1}\ \{\ell[S(n) - \sum_{i=1}^{m} k_i F_i(n) - c(n)F_j(n)]$$

$$- \ell[S(n) - \sum_{i=1}^{m} k_i F_i(n) + c(n)F_j(n)]$$

Since $\ell(e) = e^2$, this reduces to

$$\Delta\, \ell_n^j = (2\ c_n)^{-1}\{4[S(n) - \sum_{i=1}^{m} k_i F_i(n)]\ c_n F_j(n)\}$$

$$= 2[S(n) - \sum_{i=1}^{m} k_i F_i(n)]\ F_j(n) \qquad (23)$$

$$= \frac{\partial}{\partial k_j}\ \ell(e(t,\underline{k}))$$

$j = 1, \ldots, m$

Hence, $\Delta\, \underline{\ell}_n = \nabla_{\underline{k}}\ell(e(t,\underline{k}))$ which means the K-W procedure reduces to the R-M procedure for the mean-square error criteria. This is important because the R-M algorithm is computationally simpler and converges faster.

3.3 Asymptotic Distribution of Stochastic Approximation Estimates

Even though Stochastic Approximation methods are nonparametric (no assumption regarding the form of the distribution function of the noise is necessary), it can be shown that under rather general conditions the estimates are asymptotically normal.[16]

Considering first the one-dimensional R-M algorithm equation (5), Sacks proved that for $a_n = A/n$ the random variable

$$\sqrt{n}\,(x_n - \hat{x}) \text{ is } N\,[0, \frac{A^2\sigma^2}{(2A\,\zeta-1)}]$$

where $\sigma^2 = \sup_x E\{|y(x) - M(x)|^2\} < \infty$

and $\zeta = M'(\hat{x})$ = slope of the regression function at $x = \hat{x}$. In the multidimensional case, the random vector

$\sqrt{n}(\underline{x}_n - \hat{\underline{x}})$ is also asymptotically normal $N[0,PQP^{-1}]$, where

PQP^{-1} is the covariance matrix

Q has entries $q_{ij} = A^2(ab_i + Ab_j - 1)^{-1}\,\pi^*_{ij}$

$$\pi^* = P^{-1}\pi P$$

$$\pi = \lim_{\underline{x}\to\hat{\underline{x}}} [\underline{y}(\underline{x}) - \underline{M}(\underline{x})][\underline{y}(\underline{x}) - \underline{M}(\underline{x})]^T$$

$$\nabla_{\underline{x}}\,\underline{M}\,(\hat{\underline{x}}) = B = PDP^{-1}$$

D = diagonal matrix of eigenvalues (b_j, j=1, ..., m) of B

P = orthogonal matrix $\ni$ $B = PDP^{-1}$

For the uni-dimensional K-W algorithm of equation (9), the random variable $\sqrt{n}(x_n - \hat{x})$ is again asymptotically normal

$$N[0, \frac{A^2\sigma^2}{(8\zeta\,A-1)}]$$

where

$$\sigma^2 = \sup_x E\{|y(x) - M(x)|^2\} < \infty$$

and

$$\zeta = -\frac{1}{2} M''(\hat{x})$$

In the multidimensional K-W process the random vector $\sqrt{n}\, c_n\, (\underline{x}_n - \hat{\underline{x}})$ is also asymptotically normal $N[0, PQP^{-1}]$, where Q has entries $q_{ij} = A^2(4Ab_i + 4Ab_j - 1)^{-1}\, \pi_{ij}^*$

$$\pi^* = P^{-1}\pi P$$

$$\pi = \lim_{\underline{x}\to\hat{\underline{x}}} [\underline{y}(\underline{x}) - \underline{M}(\underline{x})]\ [\underline{y}(\underline{x}) - M(\underline{x})]^T$$

$$B = PDP^{-1}$$

D = diagonal matrix of eigenvalues of B

P = orthogonal matrix $\ni$ $B = PDP^{-1}$

For details and proofs see Ref. 16.

In the next chapter, techniques for accelerating the convergence and increasing the efficiency (in the statistical sense) of the estimators will be considered. The discussion will be limited to the R-M process, since analogous results hold for the K-W process.

IV. METHODS OF INCREASING THE RATE OF CONVERGENCE AND EFFICIENCY OF THE ESTIMATORS

In algorithmic techniques, one wants large step-sizes when the goal is far away and rapidly decreasing step-sizes as the goal is approached. Historically, Kester was the first to present such a procedure for Stochastic Approximation methods.[20]

4.1 Kesten's Acceleration Method

For the R-M algorithm

$$x_{n+1} = x_n - a_n\, y(x_n) \tag{24}$$

this procedure simply keeps the value of a_n constant until the sign of the observation $y(x_n)$ changes, then a_n is decreased in a manner that satisfies Dvoretsky's Theorem. The motivation being that when the zero of equation (24) is not near at hand, then the measurements of $y(x_n)$ will in general be of the same sign. However, as the goal is approached, overshoot will occur causing the estimates to oscillate about the zero of $M(x_n)$. In the latter case, the step sizes should be decreased rapidly. The table below illustrates the technique.[2]

Measurement #	1	2	3	4	5	6	7	8	Total movement
Sign of Measurement	+	+	+	-	-	+	-	+	
Unaccelerated values of a_n	1	1/2	1/2	-1/4	-1/5	1/6	-1/7	1/8	$1 \frac{149}{180}$
Accelerated values of a_n	1	1	1	-1/2	-1/2	1/3	-1/4	1/5	$2 \frac{17}{60}$

Cruz-Diaz has suggested a normalized R-M method

$$x_{n+1} = x_n - a_n \operatorname{sgn} [y(x_n)] \qquad (25)$$

which converges under the same conditions as the regular R-M algorithm.[2] This approach greatly accelerates convergence for regression functions such as $M(x) = xe^{-x}$ whose amplitude is very small for values of x much greater and much less than the actual zero $\hat{x}$.

4.2 Dvoretzky's Optimum Sequence

In Ref. 6, Dvoretzky proved the following minimax result for the sequence $\{a_n\}$.

Theorem 2. If the random variable y(x) satisfies $E\{y^2(x)\} \leq \sigma^2 < \infty$ and whose regression function

$M(x) = E\{y(x)\}$ is bounded by

$$0 < A\left|x - \hat{x}\right| \leq M(x) \leq B\left|x - \hat{x}\right| < \infty \tag{26}$$

and it is known that

$$\left|x_1 - \hat{x}\right| \leq C = \sqrt{\frac{2\sigma^2}{A(B-A)}} \tag{27}$$

then the sequence of

$$a_n = \frac{AC^2}{\sigma^2 + nA^2C^2} = \frac{A\ V_n}{2 + A\ V_n}\ \frac{2}{A+B} \tag{28}$$

yields the upper bound

$$\max_{x_n} V_n = \max_{x_n} E\{x_n - \hat{x})^2\} = \frac{\sigma^2 C^2}{\sigma^2 + (n-1)\ A^2C^2} = \frac{2\ V_n}{2 + A^2 V_n} \tag{29}$$

The sequence defined by (28) is optimum for the R-M process of equation (24) in the sense that for any other sequence, the upper bound given by (29) is violated. The result is minimax since a_n is chosen so that the maximum possible value of equation (29) is minimum. A heuristic proof of this theorem is given in Ref. 2. By using equation (27), the constant C can be eliminated from (28) and (29) giving

$$a_n = \frac{1}{A[n + \frac{(B-A)}{2A}]} \tag{30}$$

$$\max_{x_n} V_n = \max_{x_n} E\{(x_n - \hat{x})^2\} = \frac{\sigma^2}{A[\frac{B-A}{2} + (n-1)A]} \tag{31}$$

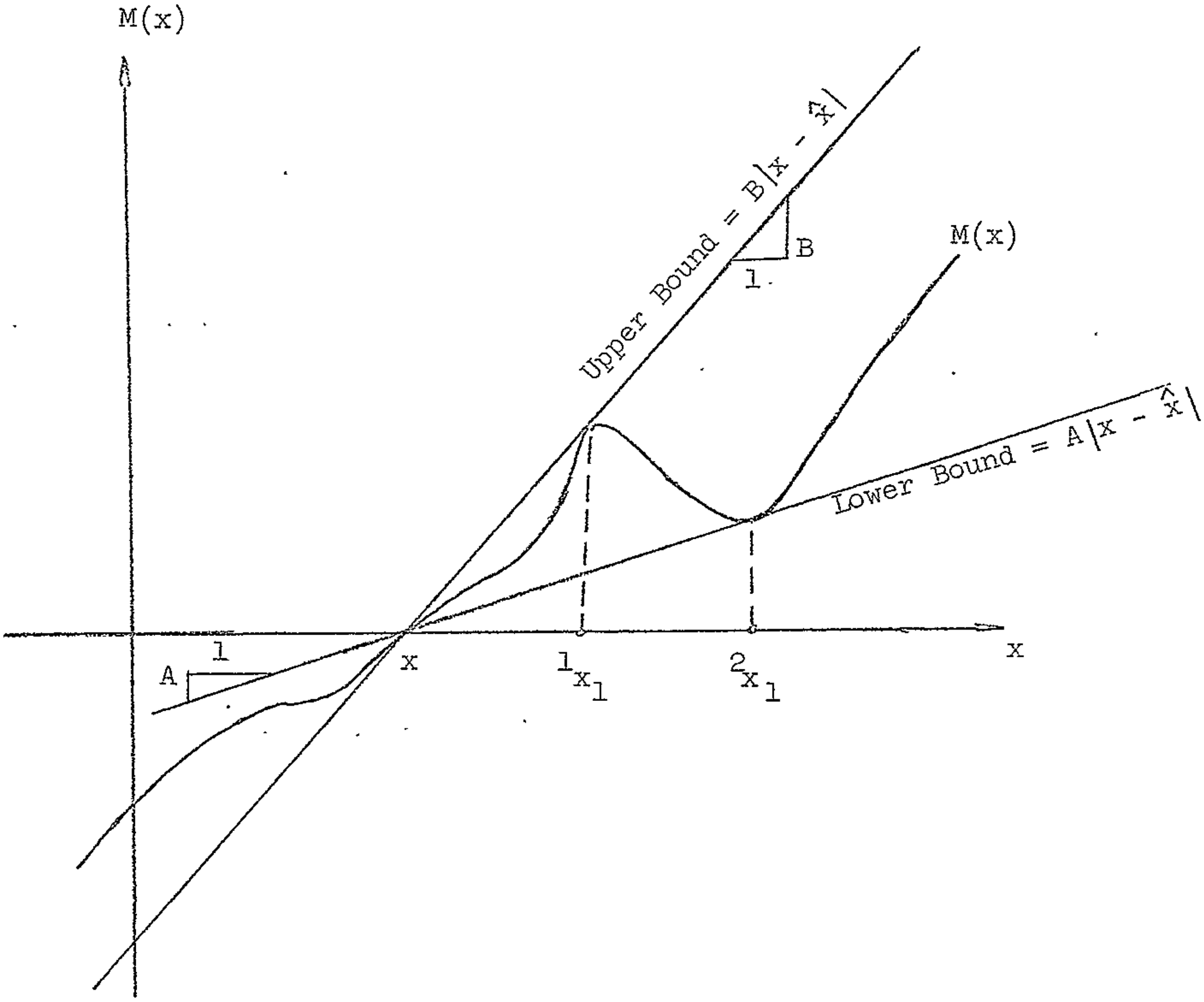
Expression (30) indicates that the optimum sequence in the minimax sense is not harmonic and does not depend on the noise. The one case where a_n reduces to a harmonic sequence is when $M(x)$ is a straight line, i.e., $A = B$. Then $a_n = \frac{1}{An}$. The effect of the noise shows up only in the variance of x_n or uncertainty in the location of $\hat{x}$ given by equation (31). Note that it is A, the slope of the lower bounding line, that determines the size of the interval of uncertainty for large n. Thus, if A is small, the interval of uncertainty remains significant for a much larger time.

The entire discussion of the last paragraph has been predicated on the assumption that $|x_1 - \hat{x}| < C$, where C is a known constant. If this is not true, then equation (30) is not the optimum sequence. However, a minimax solution can still be found. There are two worst cases; one where the expected value of the measurement at $x_1 = {}^1x_1$ falls on the upper bound $B|x - \hat{x}|$ and the other where $x_1 = {}^2x_1$ falls on the lower bound $A|x - \hat{x}|$ (see Fig. 2). In the first case,

$$M({}^1x_1) = B({}^1x_1 - \hat{x})$$

Therefore, since $x_2 = x_1 - a_1y(x_1)$ and

$$E\{x_2\} = x_1 - a_1M(x_1),$$



Regression Curve M(x)

Fig. 2

$$
\begin{aligned}
E\{{}^1x_2\} - \hat{x} &= {}^1x_1 - a_1M({}^1x_1) - \hat{x} \\
&= {}^1x_1 - a_1B({}^1x_1 - \hat{x}) - \hat{x} \\
&= (1 - a_1B)({}^1x_1 - \hat{x}) \qquad (32)
\end{aligned}
$$

Similarly, in the second case,

$$E\{{}^2x_2\} - \hat{x} = (1 - a_1A)({}^2x_1 - \hat{x}) . \qquad (33)$$

For any given x_1, equations (32) and (33) become the inequalities

$$E\{x_2\} - \hat{x} \leq (1 - a_1B)(x_1 - \hat{x}) \qquad (34)$$

$$E\{x_2\} - \hat{x} \geq (1 - a_1A)(x_1 - \hat{x})$$

or

$$E\{x_2\} - \hat{x} \leq (a_1A - 1)(x_1 - \hat{x}) \qquad (35)$$

The largest possible error in $E\{x_2\}$ is the greater value of (34) and (35),

$$\max\ [E\{x_2\} - \hat{x}] = \max\ [(1 - a_1B)(x_1 - \hat{x}),\ (a_1A - 1)(x_1 - \hat{x})] \qquad (36)$$

For the minimax solution a_1 is selected so that (36) is minimum. This obviously occurs when both terms on the right-hand side of (36) are equal. Therefore,

$$1 - a_1B = (a_1A - 1) \implies a_1 = \frac{2}{A + B}$$

Then

$$\min \max [E\{x_2 - \hat{x}\}] = \frac{B - A}{B + A} (x_1 - \hat{x})$$

Thus the minimax choice of a_1 behaves as if the regression function M(x) were a straight line of slope (A+B)/2. So in general if $\left|x_n - \hat{x}\right| \geq C$, all terms of $\{a_n\}$ are set equal to $\frac{2}{A + B}$. Then as soon as $\left|x_n - \hat{x}\right| < C$, the terms of $\{a_n\}$ are reduced in accordance with equation (30).[2]

Since the asymptotic distribution of the R-M estimator

$$\sqrt{n}\,(x_n - \hat{x}) \text{ is } N[0, \frac{A^2\sigma^2}{(2A\zeta - 1)}],$$

the asymptotic distribution of x_n is

$$N[\hat{x}, \frac{A^2\sigma^2}{n(2A\zeta - 1)}].$$

Choosing A to minimize the variance of x_{n+1} gives $A = \zeta^{-1}$. So

$$\text{Var } \{x_n\} = \frac{\sigma^2}{n\zeta^2} \qquad (37)$$

and

$$a_n = \frac{1}{n\zeta} \qquad (38)$$

is the sequence that gives the lowest asymptotic variance. The conclusion is that Dvoretzky's minimax sequence sacrifices long-term efficiency for short-term efficiency. Note that in the vicinity of the zero $\hat{x}$, the regression function may be closely approximated by the straight line $M(x_n) \approx \zeta(x_n - \hat{x})$; therefore, $y(x_n) \approx \zeta(x_n - \hat{x}) + e$.

If the experimental error (which has mean 0 and variance σ^2) is also normally distributed, then $y(x_n)$ is $N[\zeta(x_n - \hat{x}), \sigma^2]$. The Rao-Cramer lower bound on the variance of unbiased estimators for $\hat{x}$ is given by

$$1/n\ E\{[\frac{\partial \ell n\ p(y;\ \hat{x})}{\partial \hat{x}}]^2\} = V_{min}$$

$$E\{[\frac{\partial \ell n\ p(y;\hat{x})}{\partial \hat{x}}]^2\} = E\{[\frac{\zeta(y - \zeta(x_n - \hat{x}))}{\sigma^2}]^2\}$$

$$= \frac{\zeta^2}{\sigma^4} E\{[y - \zeta(x_n - \hat{x})]^2\}$$

$$= \frac{\zeta^2}{\sigma^4} \sigma^2 = \frac{\zeta^2}{\sigma^2}$$

Thus $V_{min} = \frac{\sigma^2}{n\zeta^2}$, which is exactly the asymptotic variance of x_n, given by equation (37). Consequently, for the case of Gaussian noise, the R-M algorithm gives an unbiased asymptotically efficient estimate of $\hat{x}$.

4.3 Summary of Section 4.2

In the previous section one sees three stages of the algorithmic search, in each of which the selection of the coefficients a_n differs.[2] The first stage is when the goal $\hat{x}$ is far away. Here, the coefficients should be largest and such that

$$a_n = \frac{2}{A + B},\ n = 1,\ \ldots,\ m$$

Secondly, when x_n is close enough to $\hat{x}$ to satisfy equation (27), then the coefficients are set equal to

$$a_n = \frac{1}{A[n + \frac{(B - A)}{2A}]}, \quad n = m + 1, \ldots, p$$

Finally, when x_n is near enough to $\hat{x}$ for $M(x)$ to be linear, the coefficients should be

$$a_n = \frac{1}{n\zeta}, \quad n = p + 1, \ldots$$

where $\zeta = M(x)$.

In practice, it is impossible to exactly carry out this procedure because

(1) the bounds A and B on the regression function $M(x)$ must be estimated in general,

(2) the constant C in equation (26) is known since the experimenter selects it, but $\hat{x}$ is unknown so it is not possible to determine precisely when $|x_n - \hat{x}| < C$,

(3) the slope of the regression function at $\hat{x}$ must also be estimated since both $M(x)$ and $\hat{x}$ are unknown.

4.4 Another Minimax Method

One rather obvious method of accelerating the convergence of x_n to $\hat{x}$ is to simply average, say N, observations of $y(x)$ and use this smoothed measurement $z(x)$ in place of the $y(x)$, i.e.,

let
$$z_1 = \frac{1}{N} [y(x_1) + \dots + y(x_n)]$$
$$z_2 = \frac{1}{N} [y (x_{N+1}) + \dots + y(x_{2N})]$$
$$\vdots$$
$$z_n = \frac{1}{N} [y(x_{N(n-1)+1}) + \dots + y(x_{Nn})]$$

and the R-M algorithm of equation (6) becomes

$$x_{n+1} = x_n - a_n z_n \qquad (39)$$

and assuming the random error is stationary,

$$\mathrm{Var}\ [y(x)] = \sigma^2(x) = \sigma^2$$

Therefore,

$$\mathrm{Var}\ [Z_n] = \frac{\sigma^2}{N}$$

and

$$E\{Z_n\} = \frac{1}{N} \sum_{i=1}^{N} M(x) = M(x)$$

so the bounds of equation (26) hold for $E\{Z_n\}$. Using

$$C = \sqrt{\frac{2\sigma^2/N}{A(B-A)}}, \qquad a_n = \frac{AC^2}{\frac{\sigma^2}{N} + nA^2C^2}$$

and $\max\limits_{x_n} E\{(x_n - \hat{x})\} = \max \mathrm{Var}\ \{x_n\} = \dfrac{C^2\sigma^2/N}{\sigma^2/N + (n-1)A^2C^2}$

which are the equations of algorithm (39) analogous to equations (27), (28), and (29) of algorithm (24). With these analogous forms, one obtains the following minimax result for algorithm (39)

$$a_n = \frac{1}{A[n + \frac{(B-A)}{2A}]} \tag{40}$$

$$\max_{x_n} E\{x_n - \hat{x}\} = \max_{x_n} \operatorname{Var}\{x_n\} = \frac{\sigma^2/N}{A[\frac{B-A}{2} + (n-1)A]} \tag{41}$$

This result indicates, as would be expected, that a_n is unchanged and max $\operatorname{Var}\{x_n\}$ is reduced proportionally to the number N of observations used in the smoothing process. What this process has achieved, for some given max $\operatorname{Var}\{x_n\}$, is a reduction by the factor N of the number of algorithmic iterations required, but the number of observations of $y(x)$ is not reduced.

At first glance it appears that the same reduction in variance would accrue with an associated reduction in observations required if we let

$$\begin{aligned} z_1 &= y(x_1) \\ z_2 &= \frac{1}{2}[y(x_1) + y(x_2)] \\ &\vdots \\ z_n &= \frac{1}{n}[y(x_1) + \dots + y(x_n)] \end{aligned} \tag{42}$$

However, I tried this approach using Dvoretzky's method of attack (see Ref. 6), and ended up with equations (30) and (31), which indicates no reduction of the variance in x_n. The reason for this is because the z_n defined by (42) contains no more information about $M(x)$, excepting the new observation $y(x_n)$, than does z_{n-1}.

4.5 Acceleration and the Method of Least Squares

A standard problem in optimal filtering is: Given n measurements of y where

$$y_i = \underline{c}_i x + e_i, \; i = 1, 2, \ldots, n \tag{43}$$

$\underline{c}$ is a known 1 x k row vector which may change with i, $\underline{x}$ is an unknown, but constant k vector, and e is an unbiased random variable: find the estimate $\hat{\underline{x}}_n$ of $\underline{x}$ such that $J_n(x)$ is minimized. For a least squares solution, J_n is

$$J_n(\underline{x}) = E_n^T E_n = (Y_n - C_n\underline{x})^T(Y_n - C_n\underline{x}) \tag{44}$$

where $E^T = [e_i, \ldots, e_n]$

$Y_n^T = [y_1, \ldots, y_n]$ $\qquad \underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}$

$$C_n = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nk} \end{bmatrix} = \begin{bmatrix} \underline{c}_1 \\ \underline{c}_2 \\ \vdots \\ \underline{c}_n \end{bmatrix}$$

The solution is

$$\nabla_{\underline{x}} J_n(\underline{x}) = -2C_n^T (Y_n - C_n \hat{\underline{x}}_n) = 0$$

yielding

$$\hat{\underline{x}}_n = [C_n^T C_n]^{-1} C_n^T Y_n \tag{45}$$

The subscript n represents , number of observations of y from which the least sqr :s estimate $\hat{\underline{x}}_n$ of $\underline{x}$ is made. What is now needed is a r ..rsive version of (45) so that new data can be incorporated iteratively as it is received. This is achieved as follows: assume another observation $y_{n+1} = \underline{c}_{n+1}\,\underline{x} + e_{n+1}$ is made, then

$$[C^T_{n+1}\; C_{n+1}]\; \hat{\underline{x}}_{n+1} = C^T_{n+1}\; Y_{n+1}$$

where

$$C_{n+1} = \left[\begin{array}{c} C_n \\ \hline \underline{c}_{n+1} \end{array}\right] \quad \text{and} \quad Y_{n+1} = \left[\begin{array}{c} Y_n \\ \hline y_{n+1} \end{array}\right]$$

Thus

$$[C^T_n\; C_n + \underline{c}^T_{n+1}\; \underline{c}_{n+1}]\; \hat{\underline{x}}_{n+1} = C^T_n\; Y_n + \underline{c}^T_{n+1}\; y_{n+1}$$

$$= [C^T_n\; C_n]\; \hat{x}_n + \underline{c}^T_{n+1}\; y_{n+1}$$

Subtracting $[\underline{c}^T_{n+1}\; \underline{c}_{n+1}]\; \hat{x}_n$ from both sides gives

$$[C^T_n\; C_n + \underline{c}^T_{n+1}\; \underline{c}_{n+1}]\; (\hat{\underline{x}}_{n+1} - \hat{\underline{x}}_n) = \underline{c}^T_{n+1}\; y_{n+1} - [c^T_{n+1}\; \underline{c}_{n+1}]\; \hat{x}_n$$

$$= \underline{c}^T_{n+1}\; [y_{n+1} - \underline{c}_{n+1}\; \hat{x}_n]$$

or

$$\hat{\underline{x}}_{n+1} = \hat{\underline{x}}_n + [C^T_n\; C_n + \underline{c}^T_{n+1}\; \underline{c}_{n+1}]^{-1}\; \underline{c}^T_{n+1}\; [y_{n+1} - \underline{c}_{n+1}\; \hat{\underline{x}}_n] \tag{46}$$

which is the desired recursive relation.

However, equation (46) still requires a matrix inversion every time a new observation is made. This difficulty can be removed by using the "inside out" lemma of numerical analysis. First let $P_n = C_n^T C_n$ (47)
so that

$$P_{n+1}^{-1} = C_{n+1}^T C_{n+1} = C_n^T C_n + \underline{c}_{n+1}^T \underline{c}_{n+1} = P_n^{-1} + \underline{c}_{n+1}^T \underline{c}_{n+1} \tag{48}$$

Then by the lemma,

$$P_{n+1} = P_n - P_n \underline{c}_{n+1}^T [\underline{c}_{n+1} P_n \underline{c}_{n+1}^T + 1]^{-1} \underline{c}_{n+1} P_n \tag{49}$$

Since $\underline{c}_{n+1} P_n \underline{c}_{n+1}$ is a scalar, the problem has been significantly simplified. Substituting (47) into (46) gives[22]

$$\begin{aligned}\underline{\hat{x}}_{n+1} &= \underline{\hat{x}}_n + [P_n^{-1} + \underline{c}_{n+1}^T \underline{c}_{n+1}]^{-1} \underline{c}_{n+1}^T [y_{n+1} - \underline{c}_{n+1} \underline{\hat{x}}_n] \\ &= \underline{\hat{x}}_n + P_{n+1} \underline{c}_{n+1}^T [y_{n+1} - \underline{c}_{n+1} \underline{\hat{x}}_n] \\ &= \underline{\hat{x}}_n + P_{n+1} \underline{c}_{n+1}^T [y_{n+1} - \hat{y}_{n+1}] \end{aligned} \tag{50}$$

Throughout the above discussion, the existence of the inverse of $C_{n+1}^T C_{n+1}$ has been assumed. This is analogous to the observability condition discussed by Kalman in the case of state estimation of dynamic systems.[23]

To relate the recursive formula (50) to Stochastic Approximation, premultiply equation (48) by P_{n+1} and postmultiply by P_n obtaining

$$P_n = P_{n+1} [I \quad \underline{c}^T_{n+1} \, \underline{c}_{n+1} \, P_n]$$

Postmultiplying again by $\underline{c}^T_{n+1}$ gives

$$P_n \, \underline{c}^T_{n+1} = P_{n+1} [\underline{c}^T_{n+1} + c^T_{n+1} \, \underline{c}_{n+1} \, P_n \, \underline{c}^T_{n+1}]$$

or

$$P_{n+1} \, \underline{c}^T_{n+1} = P_n \underline{c}^T_{n+1} [\underline{c}_{n+1} \, P_n \, \underline{c}^T_{n+1} + I]^{-1} \qquad (51)$$

When $\underline{c}_{n+1}$ is time invariant, $\underline{c}_{n+1} = \underline{c}$ and equation (51) becomes

$$P_{n+1} \, \underline{c}^T = P_n \, \underline{c}^T [\underline{c} \, P_n \, \underline{c}^T + I]^{-1} \qquad (52)$$

$$= P_{n-1} \, \underline{c}^T [2 \, \underline{c} \, P_{n-1} \, \underline{c}^T + I]^{-1}$$

$$= P_{n-2} \, \underline{c}^T [3 \, \underline{c} \, P_{n-2} \, \underline{c}^T + I]^{-1}$$

$$\vdots$$

$$= P_o \, \underline{c}^T [(n+1) \, \underline{c} \, P_o \, \underline{c}^T + I]^{-1} \qquad (53)$$

Equation (53) was obtained by repeated application of (52) to itself.[23] For a large number of iterations, the asymptotic versions of (53) and (50) are

$$P_{n+1} \, \underline{c}^T = \frac{P_o \, \underline{c}^T}{n+1} [\underline{c} \, P_o \, \underline{c}^T]^{-1}, \qquad n \text{ large} \qquad (54)$$

$$\hat{\underline{x}}_{n+1} = \hat{\underline{x}}_n + \frac{P_o \underline{c}^T}{n+1} [\underline{c} \, P_o \, \underline{c}^T]^{-1} [y_{n+1} - \underline{c} \, \hat{x}_n] \qquad (55)$$

Since $P_o \, \underline{c}^T [\underline{c} \, P_o \, \underline{c}^T]^{-1}$ is simply some constant k x 1 vector, each element of $P_{n+1} \, \underline{c}^T$ is just a constant divided by n+1, for - large.

Therefore,

$$\lim_{n\to\infty} P_n \underline{c}^T \quad \underline{0}, \text{ k x 1 vector}$$

$$\sum_{n=1}^{\infty} P_n \underline{c}^T = \underline{\infty}$$

and $$\sum_{n=1}^{\infty} [P_n \underline{c}^T]^2 < \underline{\infty}$$

which are the vector equivalents of the properties required of the sequence $\{a_n\}$ in Stochastic Approximation methods, and $P_n \underline{c}^T$ plays exactly the same role in algorithm (50) as does $a\underline{R}_n^{-1} = \underline{R}_n^{-1}$ in the R-M algorithm of equation (1 . Denoting $P_n \underline{c}^T$ by $\underline{Y}$ $(y_1, \ldots, y_n)$, (50) becomes

$$\underline{\hat{x}}_n = \underline{\hat{x}}_{n-1} + \underline{Y}(y_1, \ldots, y_n) [y_n - \underline{c}_{n+1} \underline{\hat{x}}_{n-1}]$$

where $\underline{Y}$ depends on all past measurements. The presence of P_o allows one to use any available a priori knowledge. For example, if the confidence in the initial estimate $\underline{\hat{x}}_o$ is low, choose $P_o = I$. It has been shown experimentally that Stochastic Approximation algorithm (50) converges much more rapidly than any of the previously mentioned acceleration techniques.[24] However, two iterative computations, equations (50) and (51), are now required; thereby paying for the increased acceleration with computational time and complexity. It should be emphasized that algorithm (50) is valid only when the parameter vector $\underline{x}$ is time invariant.

Similar results hold in the time varying case. For

$$y_i = \underline{c}_i \, \underline{x}_i + e_i$$

$\underline{x}_{i+1} = \Phi \, \underline{x}_i$ and Φ is a known transition matrix, these results are

$$\hat{\underline{x}}_{n+1} = \Phi \, \hat{\underline{x}}_n + N_{n+1} \, \underline{c}_n^T \, [\underline{c}_n \, N_{n+1} \, \underline{c}_n^T + 1]^{-1} \, (y_{n+1} - \underline{c}_{n+1} \Phi \hat{\underline{x}}_n) \tag{56}$$

$$P_{n+1} = N_{n+1} - N_{n+1} \, \underline{c}_n^T \, [\underline{c}_n \, N_{n+1} \, \underline{c}_n + 1]^{-1} \, \underline{c}_n \, N_{n+1} \tag{57}$$

$$N_{n+1} = \Phi \, P_n \, \Phi^T \tag{58}$$

These three equations define the adaptive estimation procedure alluded to at the end of section 2.3. They are valid when $\underline{c}$ and $\underline{x}$ are time-varying and can easily be adapted to the case where Φ is also time-varying.[25] The similarity between this estimator and Kalman's estimator is striking.

V. CONTINUOUS TIME STOCHASTIC APPROXIMATION METHODS

The purpose of developing continuous time Stochastic Approximation algorithms is to provide differential equations analogous to the difference equations (6) and (9). These differential equations can then be implemented on an analog computer.

5.1 The Continuous R-M Algorithm

By writing equation (5) as

$$x_{n+1} - x_n = - a_n\, y(x_n)$$

and considering the limiting case, we obtain the differential equation[26]

$$\frac{dx(t)}{dt} = - a(t)\, y(x(t)) \qquad (59)$$

in which a(t) must satisfy

$$\lim_{t\to\infty} a(t) = 0, \qquad \int_0^\infty a(t)\, dt = \infty \quad \text{and} \quad \int_0^\infty a^2(t)dt < \infty$$

The multidimensional version of (59) is just

$$\dot{\underline{x}}(t) = - a(t)\, \underline{y}(\underline{x}(t)) \qquad (60)$$

with the same restrictions on a(t).

For example, a suitable choice of a(t) is

$$a(t) = \frac{1}{t + 1} \quad t \geq 0 \ .$$

Both of the above algorithms converge in mean square and with probability one under slightly more restrictive conditions than the discrete time analogs.

5.2 The Continuous Time K-W Algorithm

By writing equation (9) as

$$x_{n+1} - x_n = - a_n \left[\frac{y(x_n + c_n) - y(x_n - c_n)}{2\ c_n}\right]$$

and again considering the limiting case, we obtain the one-dimensional K-W differential equation

$$\frac{dx(t)}{dt} = - a(t) \ [\frac{y(x(t) + c(t)) - y(x(t) - c(t))}{2\ c(t)} \tag{61}$$

in which a(t) and c(t) must satisfy

$$\lim_{n\to\infty} a(t) = 0, \qquad \int_0^\infty a(t)dt = \infty$$

$$\lim_{n\to\infty} c(t) = 0, \qquad \int_0^\infty \left[\frac{a(t)}{c(t)}\right]^2 dt < \infty$$

The multidimensional version of algorithm (17) is

$$\dot{\underline{x}}(t) = - a(t)\ \Delta\ \underline{y}(t) \tag{62}$$

where

$$\Delta \underline{y}(t) = [\underline{y}(\underline{x}^{1}(t)) - \underline{y}(\underline{x}^{-1}(t)), \ldots, \underline{y}(\underline{x}^{m}(t)) - \underline{y}(\underline{x}^{-m}(t)]/2c(t)$$

where

$$\underline{x}^{j}(t) = \underline{x}(t) + c(t)\, \underline{e}_j \text{ and } \underline{x}^{-j}(t) = \underline{x}(t) - c(t)\, \underline{e}_j$$

$j = 1, \ldots, m$

These two algorithms also converge in mean square and with probability one, but again under more restrictive conditions than the discrete time schemes.[12]

In the next chapter, application of Stochastic Approximation methods will be made to various engineering problems.

VI. ENGINEERING APPLICATIONS

The basis for application of Stochastic Approximation methods to engineering problems was laid in section 3.2, where the minimization of a performance index was formulated as a regression problem. However, the presence of constraint equations was not considered, but can be easily included using Lagrange multipliers. Assume it is desired to minimize

$$L(\underline{k}) = E_y \{\ell(y,\underline{k})\}$$

subject to the constraints

$$F_i(\underline{k}) = E_y\{f_i(y,\underline{k})\} = 0 \qquad i = 1, \ldots, M < m$$

where m is the dimension of $\underline{k}$. Then by defining the auxiliary loss function

$$\ell^1 = \ell + \underline{\lambda}^T \underline{f}$$

$\underline{\lambda}$ is a M x 1 vector

$\underline{f}$ is a M x 1 vector of constraints

and using this new loss function in equation (21), we obtain

$$\underline{k}_{n+1} = \underline{k}_n - a_n \nabla_k \ell^1(y_{n+1}, \underline{k}_n, \underline{\lambda}_n) \qquad (63a)$$

$$= \underline{k}_n - a_n \nabla_k \ell(y_{n+1},\underline{k}_n) - a_n \underline{\lambda}_n^T \nabla_k \underline{f}(y_{n+1},\underline{k}_n)$$

Equation (63) is a function of $\underline{\lambda}_n$ which is given by the companion algorithm

$$\underset{mx1}{\underline{\lambda}_{n+1}} = \underline{\lambda}_n + b_n \, k\underline{f}(y_{n+1}, \underline{k}_n) \tag{63b}$$

where b_n must be a Stochastic Approximation sequence and $\nabla_k \underline{f}$ is a M x m matrix.

Inequality constraints can also be handled, but require the introduction of an additional vector variable that converts the inequality to an equality constraint. The result is three interdependent algorithms.

With this foundation, some application of Stochastic Approximation methods will be presented.

6.1 Coding Theory

Schalkwijk and Kailath[27] considered the problem of transmitting one of M possible signals, where each signal takes T seconds to transmit, over a noisy channel without memory with the availability of a noiseless feedback link (such a situation is typical of a satellite-to-ground transmission). It is important to remember that the feedback path can not increase the channel capacity as was first shown by Shannon,[28] but does simplify the complexity of the coding and decoding required to achieve a given performance.

To begin with, the communication is assumed to be over a forward channel with white Gaussian noise of spectral density $\frac{N_o}{2}$ and a noiseless feedback channel. The message information is transmitted by modulating the amplitude of N orthonormal waveforms $\varphi_i(t)$,

$$\int_o^T \varphi_i(t)\varphi_j(t)dt = \delta_{ij} \qquad i, j = 1, 2, \ldots, N$$

Since the time allowed to transmit the information signal is T seconds, these waveforms might represent N successive and non-overlapping pulses of duration T/N. Thus, the information signal transmitted has the form

$$S(t) = \sum_{i=1}^{N} \alpha_i \varphi_j(t)$$

and the received signal is

$$Y(t) = S(t) + N(t)$$

Reception is then achieved by using filters matched to the waveforms $\varphi_i(t)$, giving as outputs

$$Y_i = \alpha_i + N_i$$

where, due to the assumed structure of N(t), the N_i are zero mean stochastically independent random variables with variance $N_{o/2}$.

This procedure is valid even if the original channel is a continuous time channel because the matched filter for white Gaussian noise computes the likelihood ratio, which gives a sufficient statistic, and therefore preserves all the relevant information in the received waveform required for the decision process.

The coding method for sending one of the M possible messages consists of dividing the unit interval [0, 1] into M disjoint equal-length message intervals. Then select as the "message point" φ_k, the mid-point of the kth message interval, i.e.,

$$\varphi_k = \frac{2k - 1}{2M}, \qquad k = 1, \ldots, M.$$

Now by transmitting the code point φ_k via successive signals $\mathcal{s}_i\varphi_i(t)$, $j = 1, \ldots N$. At the receiver, an estimate of φ_k is formed from $Y_i = \mathcal{s}_i + N_i$. Letting α_n denote the estimate of φ_k after receiving n values of Y_i, the mean square error is

$$E\{(\alpha_n - \varphi_k)^2\} \qquad n = 1, 2, \ldots, N$$

which decreases as n increases. At the conclusion of the Nth transmission a decision is made as to which code message φ_k was transmitted by choosing the coding point closest to α_N. The error probability P_e is then given by

$$P\{\left|\alpha_n - \varphi_k\right| \geq 1/2\ M\}.$$

The goal is to invent a coding scheme such that for any given $\epsilon > 0$, we have $P_e < \epsilon$ for a transmission rate R less than the channel capacity of the Gaussian noise channel, which here is assumed to have an infinite bandwidth and with the usual constraint on the average transmission power P_{avg}. For this channel, the capacity is

$$C = \frac{P_{avg}}{2N_o \ell n2} \text{ bits/sec,}^{29}$$

and the transmission rate $R = \frac{\log_2 M}{T} \frac{\text{bits}}{\text{sec}}$.

It is not possible to achieve the above-stated goal by simply transmitting φ_k, with $s_i = \varphi_k$, $i = 1, \ldots, N$ and using a fixed rate R less than the channel capacity. However, since a noiseless feedback link is available, the receiver can re-transmit α_n, its current estimate of φ_k back to the transmitter. Thus the transmitter can simply transmit a correction term to the receiver. Then since α_n approaches φ_k as n increases, the average power (in a statistically considered sense) needed to transmit the correction decreases as n increases from 1 to N. This saving of average power is sufficient to achieve a transmission rate arbitrarily close to channel capacity while keeping P_e as small as desired by increasing T. This is the idea behind the method of Schalkwijk and Kailath.

Specifically, they begin by taking α_1, the first estimate of φ_k as 0.5. The receiver feeds back this estimate without error to the transmitter, which then generates an error signal s_1 such that

$$\mathcal{A}_1 = \beta(\alpha_1 - \varphi_k) = \beta(0.5 - \varphi_k), \ \beta > 0$$

The signal $\mathcal{A}_1$ is then transmitted and observed at the receiver as

$$Y_1 = \mathcal{A}_1 + N_1 = \beta(0.5 - \varphi_k) + N_1$$

The receiver now computes the second estimate

$$\alpha_2 = \alpha_1 - \frac{\zeta^{-1}}{1} Y_1$$

where ζ is a constant that is chosen to minimize the variance of the estimates α_n. From section 4.2 equations (37) and (38), this minimum is achieved by choosing $\zeta^{-1} = \beta$ Therefore, $\alpha_2 = \alpha_1 - \frac{\beta}{1} Y_1$ which is now re-transmitted to the transmitter, where the correction is made and sent as $\mathcal{A}_2 = \beta(\alpha_2 - \varphi_k)$. Again the signal received is

$$Y_2 = \mathcal{A}_2 + N_2 = \beta(\alpha_2 - \varphi_k) + N_2$$

and the receiver computes α_3

$$\alpha_3 = \alpha_2 - \frac{\beta}{2} Y_2$$

In general, then, one receives

$$Y_n = \mathcal{A}_n + N_n$$

and computes

$$\alpha_{n+1} = \alpha_n - \frac{\beta}{n} Y_n. \tag{64}$$

The estimate α_{n+1} is then sent back to the transmitter which will transmit

$$\mathcal{A}_{n+1} = \beta(\alpha_{n+1} - \varphi_k)$$

This coding scheme is diagramed in Fig. 3. Note that equation (64) is just a R-M algorithm with

$$M(\alpha) = \beta(\alpha - \varphi_k) \qquad \text{and} \qquad Y_n(\alpha) = M(\alpha) + N_n$$

Therefore, we know α_{n+1} converges to φ_k in mean square and with probability one.

Without going into further detail, the results of this coding scheme will be summarized:

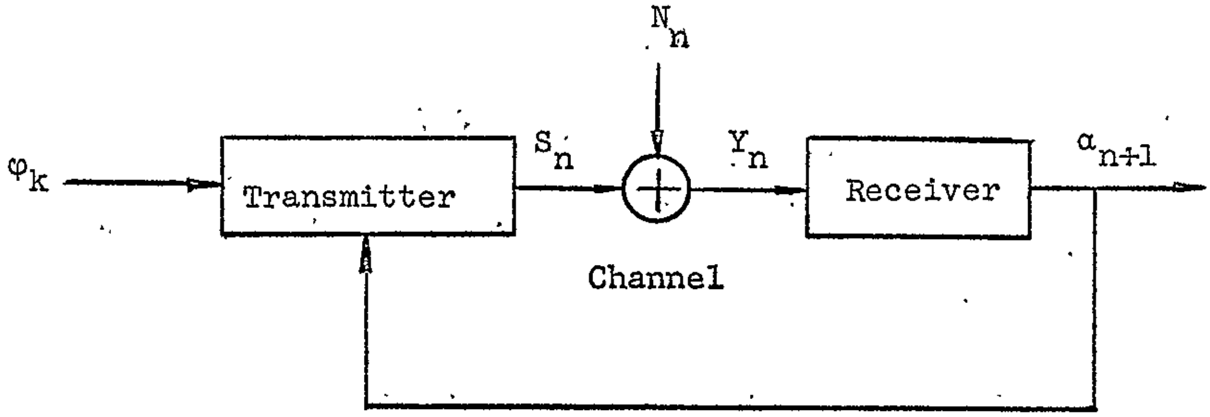
1. For any rate R less than C,

$$P_e = 2 \text{ erfc } \left\{\frac{\sqrt{3}e^{(C-R)T}}{e^{1.577}}\right\}$$

2. This coding scheme achieves a given P_e for a rate R with a transmission time T approximately one-tenth as long as required for the same P_e and rate R with orthogonal coding and no feedback.

3. If the delay τ in the feedback path is included, the performance deteriorates negligibly so long as $\tau << T$.

In the previous application, the channel was assumed to have no bandwidth constraint. For the same problem, except where the channel is bandlimited,



Communication System with Noiseless Feedback Path

Fig. 3

Schalkwijk and Kailath's R-M coding scheme gave <u>the first deterministic procedure to achieve rates up to channel capacity</u>.

6.2 Filtering and Predictio

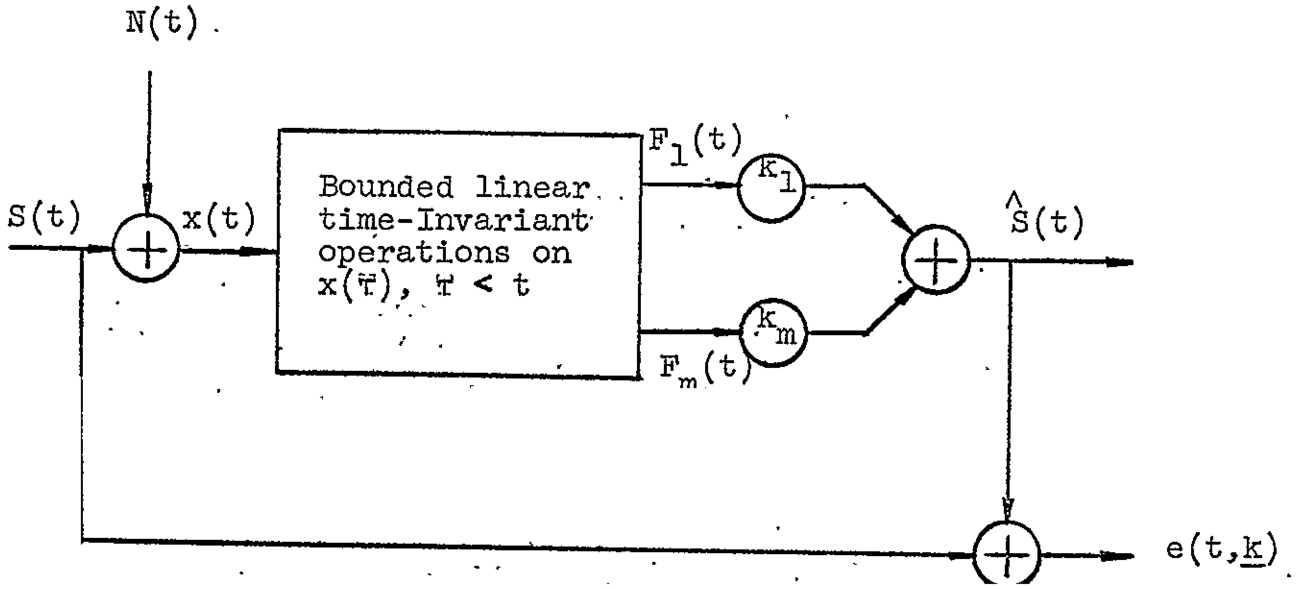
The filtering and prediction problem is essentially one in system optimization. Here the attention will be primarily devoted to filtering. This problem reduces to finding a matched filter for a noise corrupted deterministic signal and a Wiener filter for a Gaussian signal in a noisy Gaussian background. The foundation for this application was laid in section 3.2, where it was noted that the only restriction on the loss function $\ell(\cdot)$ was that it be strictly convex. For simplicity, the old standby error squared criteria will be used here, $\ell(e) = e^2(t,\underline{k})$.

The parametric filter form is shown in Fig. 4, where

$$F_i(t) = \int_0^t h_i(\tau)\ x\ (t-\tau)d\tau, \qquad i = 1, \ldots, m$$

are fixed optimum filters for a given set of m different conditions on the signal and noise. The goal is to recursively adjust the variable parameter set $\underline{k}$ as some environmental or system condition changes, say, the noise power level or noise distribution function, so that

$$L(\underline{k}) = E\{\ell[e(t,\underline{k})]\} = E\{e^2(t,\underline{k})\}$$



Filter Structure

Fig. 4

is minimized. Thus we seek the solution to $\nabla_{\underline{k}}\ L(\underline{k}) = 0$, but this is impossible since the distribution function of the error is not assumed to be known. Using Stochastic Approximation, we iteratively solve $\nabla_k\ \ell[e(t,k)]$, but this requires the availability of $S(t)$. However, the problem can be simplified so that it is not necessary to observe $S(t)$ to select the optimum $\underline{k}$ as is indicated in Fig. 4. Assuming the signal and noise are uncorrelated,

$$L(\underline{k}) = E\{[S - \hat{S}]^2\} = E\{[x - \hat{S} - N]^2\}$$

$$= E\{[x - \hat{S}]^2\} - 2E\{N(S + N)\} + 2E\{N\hat{S}\} + E\{N^2\}$$

$$= E\{[x - \underline{k}^T F]^2\} - E\{N^2\} + 2\underline{k}^T \int_0^t \underline{h}(\tau) R_{NN}(\tau) d\tau$$

Therefore,

$$\nabla_{\underline{k}}\ L\ (\underline{k}) = \nabla_{\underline{k}}\ E\{[x - \underline{k}^T F]^2 + 2 \int^t \underline{h}(\tau) R_{NN}(\tau) d\tau$$

However,

$\nabla_{\underline{k}}\ E\{[x - \underline{k}^T F]^2\}$ still can not be computed because the probability distribution function of $x(t)$ is not assumed to be known, even though the filters $F_i(t)$ were designed for Gaussian noise. But the reason for having a parameter vector to adjust is because $N(t)$ may not be Gaussian. Regardless, we do not assume a knowledge of the structure of $N(t)$. Therefore, to solve

$$\nabla_k\ E\{[x - \underline{k}^T F]^2\} + 2 \int_0^t \underline{h}(\tau) R_{NN}(\tau) d\tau = 0$$

we must iteratively solve

$$\nabla_k[x - \underline{k}^T F]^2 + 2 \int_0^t \underline{h}(\tau) R_{NN}(\tau) d\tau = 0$$

or

$$- [x - \underline{k}^T F]F + \int_0^t \underline{h}(\tau) R_{NN}(\tau) d\tau = 0 \qquad (65)$$

where the autocorrelation function $R_{NN}(\tau)$ of the noise is assumed to be determinable. The R-M algorithm for finding the optimum $\underline{k}$ is then

$$\underline{k}_{n+1} = \underline{k}_n + a_n \{[x(t) - \underline{k}_n^T F(t)] F(t) + \underline{K}\} \qquad (66)$$

where

$$\underline{K} = \int_0^t \underline{h}_n(\tau) R_{NN}(\tau) d\tau$$

In the case of detecting deterministic signals, the matched filter $h_j(t)$ is approximated by a linear combination of known functions $\varphi_i(t)$

$$h_j(t) = \sum_{i=1}^{m} k_{ij} \varphi_i(t)$$

where the subscript j corresponds to the filter matched to the jth deterministic signal. An analysis similar to that above then gives the optimum $\underline{k}_j$.

By using the continuous Stochastic Approximation differential equation corresponding to the difference equation (66), the optimization of $\underline{k}$ may be simply implemented on an analog computer.

6.3 Estimating Probability Densities and Correlation Functions[30]

The estimation of an unknown function $y = f(x)$ from a finite number of randomly observed points of the input data $x(t)$ which may also be noise corrupted can be solved using Stochastic Approximation by assuming that $f(x)$ may be represented or approximated by a sum of arbitrary independent functions $\varphi_i(x)$, so that

$$\hat{f}(x) = \underline{k}^T \underline{\varphi}(x) = \sum_{i=1}^{m} k_i \varphi_i(x) \tag{67}$$

where $\underline{k}$ is our variable parameter vector. For simplicity, let the $\varphi_i(x)$ be orthonormal and choose $\underline{k}$ to minimize

$$L(\underline{k}) = \int_X [f(x) - \underline{k}^T \underline{\varphi}(x)]^2 \, dx$$

by again solving

$$\nabla_k L(\underline{k}) = 2\int_X [f(x) - \underline{k}^T \underline{\varphi}(x)] \underline{\varphi}(x) \, dx = 0$$

$$= 2\int_X f(x)\underline{\varphi}(x)dx - 2\underline{k} = 0$$

because the $\varphi_i(x)$ are orthonormal. Therefore, $L(\underline{k})$ is minimized at

$$\hat{\underline{k}} = \int_X f(x)\underline{\varphi}(x)dx = E\{\underline{\varphi}(x)\}$$

but $f(x)$ is unknown, so use the Stochastic Approximation algorithm to solve

$$E\{\underline{\varphi}(x) - \hat{\underline{k}}\} = 0$$

The necessary recursive relation is simply

$$\underline{k}_{n+1} = \underline{k}_n + a_n [\underline{\varphi}(x_n) - \underline{k}_n]$$

or its continuous analogue

$$\frac{d\underline{k}}{dt} = a(t) [\underline{\varphi}[x(t)] - \underline{k}(t)] \qquad (69)$$

To estimate a correlation function $R(\tau) = E\{x(t+\tau)x(t)\}$

of the random process $x(t)$, when $f(x(t))$ is unknown, one applies the algorithm

$$R_{n+1}(\tau) = R_n(\tau) + a_n [x(n+\tau)x(n) - R_n(\tau)]$$

or

$$\frac{dR_t(\tau)}{dt} = a(t) [x(t+\tau)x(t) - R_t(\tau)] \qquad (70)$$

6.4 Identification

There are many examples where Stochastic Approximation methods can be applied to system identification.[26,35] [38] Here the elementary case of identifying a causal time invariant discrete system will be considered. If the input is applied at $n = 0$, the output $x(n)$ may be written using the convolution summation as

$$x(m) = \sum_{i=o}^{m} k_i u(m-i) = \underline{k}^T \underline{u}$$

The identification procedure consists of determining the weighting sequence $k_1, \ldots, k_m$ denoted by $\underline{k}^T$ by observing the output x(t) which may be noise corrupted and minimizing some convex error criteria $L(\underline{k}) = E\{\ell(e(t))\}$. Invoking the methods of Stochastic Approximation, one obtains

$$\underline{k}_{n+1} = \underline{k}_n + a_n \ell'[x[n] - \underline{k}_n^T \underline{u}[n]]\, \underline{u}\,[n]$$

and we know

$$\underset{n\to\infty}{\ell.i.m}\ \underline{k}_n = \underline{k} = [k_o, \ldots, k_m]^T$$

6.5 Dual Control[30]

This is one of the most difficult problems in control theory and was essentially defined by Fel'baum[31] using the decision theory approach. The goal is to control a plant with unknown parameters and external disturbances. Fel'baum's approach is almost impossible to apply, even if the a priori distribution of the plant parameter and the external influences are given, except in simple cases.

A more general approach that is somewhat less complicated than Fel'baum's and requires less a priori knowledge is a Stochastic Approximation formulation.

Given the linear discrete system

$$\underline{x}(n) = \sum_{i=1}^{M} C_i \underline{x}(n-i) + \sum_{i=1}^{N} d_i u(n-i)$$

$$= \underline{C}^T \underline{x} + \underline{d}^T\, \underline{u}$$

where $\underline{C}$ and $\underline{d}$ are unknown.

Define $\underline{k} = (C_1, \ldots, C_M, d_1, \ldots, d_N)$

and $\underline{Z}(n) = [\underline{x}(n-1), \ldots, \underline{x}(n-M), u(n-1), \ldots, u(n-N)]$

Therefore, $x(n) = \underline{k}^T \underline{Z}(n)$ and choose the loss function $\ell(\cdot)$ to be a convex function

$$\ell = \ell[\underline{x}(n) - \underline{k}^T \underline{Z}(n)]$$

We want to find the solution to $\nabla_{\underline{k}} E\{\ell\} = 0$ by iteratively solving $\nabla_{\underline{k}} \ell(\cdot) = 0$.

Using the R-M method, the convergent identification algorithm is

$$\underline{k}_n = \underline{k}_{n-1} + a_n \nabla_k \ell[x(n+1) - \underline{k}^T(n-1)\ \underline{Z}(n)]$$

$$= \underline{k}_{n-1}\ a_n\ \ell'[x(n+1) - \underline{k}^T(n-1)\ \underline{Z}(n)]\ \underline{Z}(n) \qquad (71)$$

where ℓ' denotes the derivative of ℓ with respect to its argument.

The controller is designed to generate a control law of the form

$$u(n) = \underline{\beta}^T\ \underline{m}[x(n)] = \sum_{i=1}^{P} \beta_i\ m_i[\underline{x}(n)]$$

where the m_i are linearly independent functions. The control performance index $I(\underline{k}_{opt}, \underline{\beta})$ is

$$I(\underline{k}_{opt}, \underline{\beta}) = E\{J[\underline{x} = \underline{k}^T_{opt}\ \underline{Z},\ u(\underline{x}, \underline{\beta})]\}$$

where J is a convex loss function. Now we wish to find the $\underline{\beta}_{opt}$ that solves $\nabla_{\underline{\beta}} I(k_{opt}, \underline{\beta}) = 0$ using only knowledge of

$\nabla_{\underline{\beta}} J[\underline{k}^T_{opt}\ \underline{Z},\ u\ (\underline{k}^T_{opt}\ \underline{Z},\ \underline{\beta})]$, but $\underline{k}_{opt}$ is not known so the best one can do is use $\nabla_{\underline{\beta}}\ J[\underline{k}^T_{n-1}\ \underline{Z}(n),\ u(k_{n-1}\underline{Z},\ \underline{\beta})]$

or equivalently

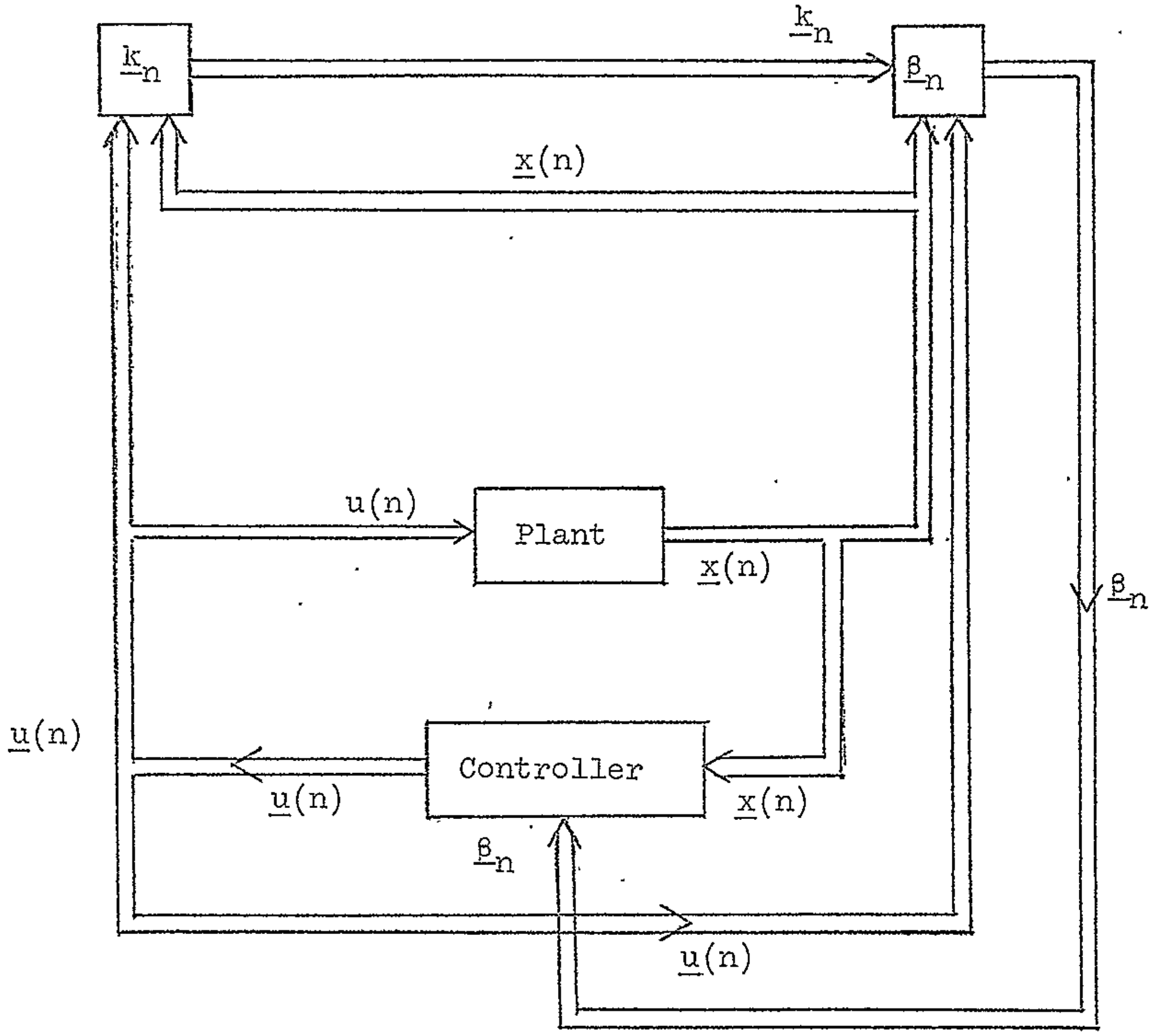
$$\nabla_{\underline{\beta}} J\{\underline{k}^T_{n-1}\ \underline{Z}(n),\ \underline{\beta}^T \underline{m}[\underline{x}(n)]\}$$

Thus the algorithm for finding the optimum $\underline{\beta}$ is

$$\underline{\beta}_n = \underline{\beta}_{n-1} + b_n\ \nabla_{\underline{\beta}} J\{\underline{k}^T \underline{Z}(n),\ \underline{\beta}^T_{n-1}\ \underline{m}[\underline{k}^T_{n-1}\ \underline{Z}(n)]\}$$

$$= \underline{\beta}_{n-1} + b_n\ \nabla_{\underline{\beta}} J\{\underline{x}(n-1) + u(n-1)\} \qquad (72)$$

which gives the convergent control algorithm. Note that the equation of identification (71) and that of control (72) are interdependent. Their block diagram representation is shown in Fig. 5. Analogous continuous algorithms can easily be derived for analog simulation.



Realizations of Algorithms (71) and (72)

Fig. 5

6.6 Controllable Parameters[30]

A common problem in control systems and in mass production of, say, missile components is to adjust a set of controllable parameters $\underline{k}$ to minimize the influence of uncontrollable changes in a set of parameters $\underline{c}$ on desired system performance. For example, $\underline{c}$ may be the pole and zero locations or gain and $\underline{k}$ may be the state variable feedback coefficients. Or $\underline{c}$ may be the mean values and variances of a set of variables and $\underline{k}$ may be the adjustable means and variances of a set of controllable parameters.

Thus, we define a performance criteria $I(\underline{c},\underline{k})$ where the variations in $\underline{c}$ may be random, but stationary. The attempt is to find the value of $\underline{k}$ that minimizes

$$E\{I(\underline{c},\underline{k})\} = \int_{\underline{c}} I(\underline{c},k)dF(\underline{c}) = J(k)$$

This problem can be solved in general even if $F(\underline{c})$ is unknown by applying Stochastic Approximation, obtaining the algorithm, where c_{n-1} is obtained by continuously monitoring it.

$$\underline{k}_n = \underline{k}_{n-1} + a_n \nabla_{\underline{k}} \, I[\underline{c}_{n-1},\underline{k}_{n-1}] \tag{73}$$

6.7 Allocation of Limited Resources[30,32]

This last application deals with an Operation Research problem in reliability or allocation of limited resources. It is desired to find the optimum method

$p(x) = \underline{k}^T\underline{\varphi}(x)$ of distributing a limited quantity of resources x in which we wish to maximize the expected gain G

$$G = E\{g[p(x),x]\}$$

under the constraint on the resources

$$\int_X W(x)p(x)dx = c$$

where W(x) is a weighting function, say, 1 in this example.

When the probability density function f(g) or equivalently f(x) is not known, it is common practice to seek a minimax solution. By applying Stochastic Approximation, we can avoid this conservative approach.

It is first necessary to guarantee that the constraint

$$\int_X p(x)dx - \tilde{c} = \underline{k}^T\int_X \underline{\varphi}(x)dx - c$$

$$= \underline{k}^T\ \underline{B} - c = 0$$

is satisfied. This is easily accomplished by using Lagrange multipliers, giving

$$I = G(\underline{k}) + \lambda(\underline{k}^T\underline{B} - c)$$

and seeking the solution to

$$\nabla_{\underline{k}} I = 0$$

by recursively computing

$$\underline{k}_n = \underline{k}_{n-1} + a_n\{\nabla_{\underline{k}} g[\underline{k}_{n-1}^T \varphi(x_n), x_n] + \lambda_n \underline{B}\} \qquad (74)$$

$$\lambda_n = \lambda_{n-1} + b_n[\underline{k}_{n-1}^T \underline{B} - c) \qquad (75)$$

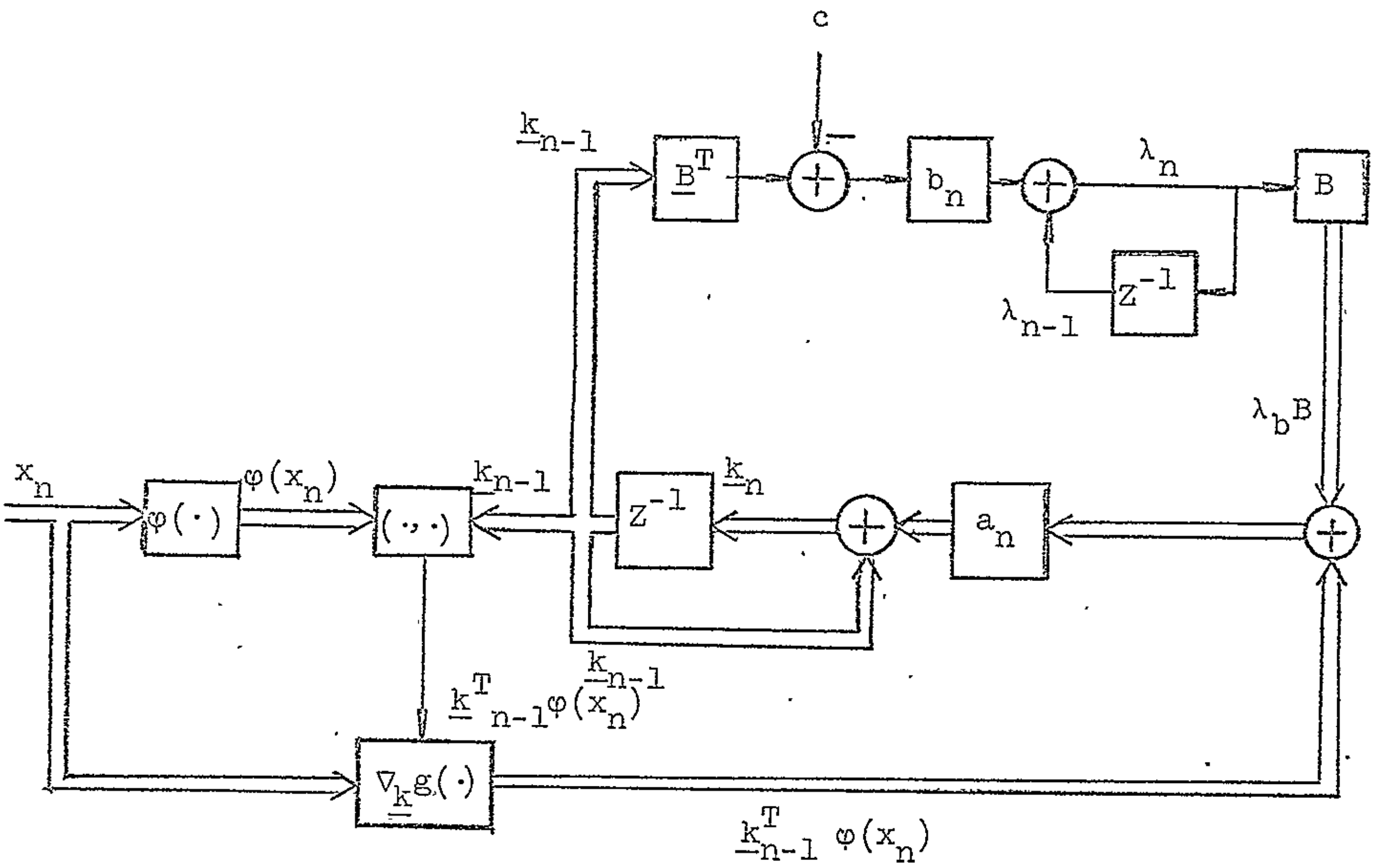
or using their continuous counterparts

$$\frac{d\underline{k}}{dt} = a(t)\{\nabla_k g[\underline{k}^T(t)\varphi(x(t)), x(t)] + \lambda(t)b\}$$

$$\frac{d\lambda}{dt} = b(t)[\underline{k}^T(t)\underline{B} - c]$$

The block diagram for equations (74) and (75) is shown in Fig. 6. The unusual characteristic of the schematic is that it is in essence a perceptron, a device originally devised by Rosenblatt[33] in his work on artificial intelligence.[34] Here, however, Rosenblatt's threshold functions have been replaced with the linearly independent functions $\underline{\varphi}$.

In concluding this chapter, it should be noted that the techniques discussed in Chapter IV on accelerating Stochastic Approximation schemes may be used in all the applications considered.



Circuit Realizing Equations (74) and (75)

Fig. 6.

CONCLUSION

This paper has been in essence an attempt to deal with many topics in optimization theory from an algorithmic viewpoint suitable for computer solution. Such an approach is especially useful in complicated engineering systems where the only analytically feasible solution requires simplifications that make the results meaningless.

It should be pointed out that many other research topics which are appropriate for Stochastic Approximation methods have not been presented. Some of these subjects are pattern recognition,random-rounding computer errors, quantal response in biological systems, learning control systems, inertial and non-inertial non-linear system identification and control, process control, estimation in radar and radio astronomy, trainable threshold logic and probabilistic automata. In addition, the Stochastic Approximation algorithms considered contain the Potential Function method of Aizerman, Braverman, and Rozonoer as a special case.[37]

In closing, areas of future research will be cited. A few of these are: development of a

(1) Stochastic Approximation Newton-Rapson Method

(2) Stochastic Approximation Conjugate Gradient Method

(3) and extension of Stochastic Approximation Methods to function spaces as has been done for steepest ascent methods.

A forthcoming paper on self-adaptive filtering and prediction will describe original results which are a direct consequence of this study.

REFERENCES

1. Chang, S. L., Optimum Control Systems, 1961.

2. Wilde, D., Optimum Seeking Methods, 1964.

3. Poisson, S. D., Recherches sur le Probabilité des Jugements, 1837.

4. Robbins, H., and S. Monro, "A Stochastic Approximation Method," Ann. Math. Stat., Vol. 22, No. 1, 1951.

5. Kiefer, J., and J. Wolfowitz, "Statistical Estimation of the Maximum of a Regression Function," Ann. Math. Stat., Vol. 23, No. 3, 1952.

6. Dvoretzky, A., "On Stochastic Approximation," Third Berkeley Symposium on Math. Statistics and Probability, 1956.

7. Derman, C., "Stochastic Approximation," Ann. Math. Stat., Vol. 27, 1956.

8. Schmetterer, L., "Stochastic Approximation," Fourth Berkeley Symposium on Math. Statistics and Probability, 1961.

9. Geiringer, H., and R. Von Mises, "Praktische Verfahren der Gleichungsauflösung," Z. Angew. Math. Mech., Vol. 9, 1929.

10. Blum, J. R., "Approximation Methods Which Converge with Probability One," Ann. Math. Stat., Vol. 25, No. 2, 1954.

11. Kallianpur, G., "A Note on the Robbins-Monro Stochastic Approximation Method," Ann. Math. Stat., Vol. 25, No. 2, 1954.

12. Sakrison, D. J., "Stochastic Approximation," Advances in Communication Systems, 1966.

13. Germansky, B., "Notiz über die Lösung von Extremaufgaben Mittels Iteration," Z. Angew. Math. Mech., Vol. 14, 1933.

14. Machol, R. E., Information and Decision Processes, 1960.

15. Blum, J., "Multidimensional Stochastic Approximation Procedures," Ann. Math. Stat., Vol. 25, No. 4, 1954.

16. Sacks, J., "Asymptotic Distribution of Stochastic Approximation Procedures," Ann. Math. Stat., Vol. 29, No. 2, 1958.

17. Gray, K. B., "Application of Stochastic Approximation to Random Circuits," XVI Symposium in Applied Mathematics, 1964.

18. Deutsch, R., Estimation Theory, Prentice Hall, 1965.

19. Wiener, N., Nonlinear Problems in Random Theory, M.I.T. Press, 1958.

20. Kesten, H., "Accelerated Stochastic Approximation," Ann. Math. Stat., Vol. 29, No. 1, 1958.

21. Venter, J., "On Stochastic Approximation Methods," Ph.D. Dissertation, Univ. of Chicago, 1964.

22. Lee, R. C. K., Optimal Estimation, Identification and Control, M.I.T. Press, 1964.

23. Ho, Y. C., "Stochastic Approximation Method and Optimal Filter Theory," Journal of Math. Anal. and Applications, Vol. 6, 1962.

24. Ho, Y. C., and C. C. Blaydon, "On the Abstraction Problem in Pattern Recognition," Fifth Symposium on Discrete Adaptive Processes, Chicago, 1966.

25. Ho, Y. C., and R. C. K. Lee, "Identification of Linear Dynamic Systems," Proceedings of the 3rd Discrete Adaptive Control Symposium, 1964.

26. Sakrison, D. J., "The Use of Stochastic Approximation to Solve the System Identification Problem," IEEE Transactions on A-C, October 1967.

27. Schalkwijk, J., and T. Kailath, "A Coding Scheme for Additive Noise Channels with Feedback - Part I: No Bandwidth Constraint, Part II: Band-Limited Signals," IEEE Transactions on Information Theory, April 1966.

28. Shannon, C. E., "The Zero-Error Capacity of a Noisy Channel," IRE Transactions on Information Theory, September 1956.

29. Fano, R. M., Transmission of Information 1961.

30. Tsypkin, Y. Z., "Adaptation, Learning, and Self-learning in Automatic Control Systems," 3rd IFAC Conference, London, 1966.

31. Fel'baum, A. A., "Dual Control Theory, I-III," Automation and Remote Control, Sept. 1960, Jan. and Feb. 1961.

32. Klipman, B. O., "The Theory of Search," Operations Research, Vol. 5, No. 5, 1957.

33. Rosenblatt, F., Principles of Neurodynamics 1962.

34. Nilsson, N. J., Learning Machines, 1965.

35. Kwatny, H., and C. Shen, "Identification of Non-Linear Systems Using a Method of Stochastic Approximation," 1967 JACC.

36. Loeve, M., Probability Theory, 1960.

37. Aizerman, M., Braverman, E., and Rozonoer, L., Automation and Remote Control, pp. 821-837, 1175-1190 and 1546-1556, Vol. 25, 1964.

38. Saridis and Ricker, Simulation, pp. 241-248, Nov. 1968.

PART II

STOCHASTIC ALGORITHMS FOR SELF-ADAPTIVE FILTERING AND PREDICTION

by

Robert L. T. Hampton

January 1970

TABLE OF CONTENTS

ABSTRACT

In this report the problem of self-adaptive optimal estimation of a sampled stochastic signal observed in random noise is formulated and an engineering solution is developed. Chapter I introduces the topic and reviews the results of recent research. Chapter II gives the necessary background material from estimation theory. Chapter III develops the learning criterion and derives the adaptive stochastic algorithms from it. The learning criterion is based on the principle of orthogonality of Chapter II. Chapter IV presents the experimental results obtained by applying the learning criterion and associated algorithms to specific systems.

CHAPTER I

INTRODUCTION

1.1 Prologue

Recently considerable attention has been directed toward self-adaptive (or self-learning) optimum systems. The basic idea is quite simple: one wishes to design a system to perform 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.

In this report the goal is to provide a self-adaptive solution to the problem of optimal filtering, prediction, and detection of stochastic signals imbedded in random noise. However, before discussing the principal results, 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 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 state transition representation, a linear dynamic system model of the plant can be described by

$$\underset{\sim}{x}(n+1) = \underset{\sim}{\Phi}\underset{\sim}{x}(n) + D\underset{\sim}{\mu}(n) + \underset{\sim}{w}(n) \tag{1.1}$$

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

$$\underset{\sim}{y}(n) = H\underset{\sim}{x}(n) + \underset{\sim}{v}(n) \tag{1.2}$$

where

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

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

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

Φ is the one-step m x m state transition matrix

D is the m x ℓ control matrix

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

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

H is the p x m observation matrix.

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 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 thesis, only the estimation phase is considered because the deterministic control solution is well known (Shultz and Melsa, 1967 or Sage, 1968). In communication theory an equally important topic is the stochastic detection problem. Here one is required to determine the optimal receiver for detecting the presence of a stochastic signal $\underset{\sim}{x}(t)$ imbedded in additive random noise $\underset{\sim}{v}(t)$. Assuming 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 noise input (Kalman, 1960). The linear dynamic system is called the signal generating process, and in state transition representation is described by

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

and the measurements of the signal $\underset{\sim}{x}(n)$ by

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

The notation is the same as that of equations (1.1) and (1.2). The signal generating process (1.3) is identical to the control plant process (1.1), except for the control input $\underset{\sim}{\mu}(n)$. However, the Separation Theorem states 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.

Also, there 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 deterministic signal, i.e., the output of the filter is considered to be the actual signal. Again, only the estimation phase is considered since the deterministic 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 was that the filter satisfy the Wiener-Hopf equation, and 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 Balakrishman (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 representation of linear filtering to obtain a direct solution in the time domain. In contrast to the 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, 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 must exist to perform the Kalman filter computations. The presence of either noiseless measurements or correlated observation noise can render R singular. In practice R is often ill-conditional simply because one measurement is an order of magnitude more accurate than the others. Thus the Kalman filter formulation can generate application difficulties. Bryson and Johansen (1965) and Bryson and Mehra (1968) have modified the Kalman framework to handle this particular problem, but their technique necessitates state augmentation which increases the dimension of the filter and the computation time.

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 sampled data, stationary optimum filtering and prediction problem is sought which does not require a priori specification of R and Q and 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 with the single unknown element R, and 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.

The second approach is to estimate directly the components of R and Q. Shellenbarger (1966) showed how to use the likelihood principle to accomplish this estimation under the assumption of Gaussian distributions and other more restrictive requirements which limit its utility. As a result, Shellenbarger (1967) developed a more general least-squares learning method to determine R and Q. Proof of convergence is not considered. It is important to note that both of these approaches require the determination of both the R and Q matrices, and the existence of the inverse of the estimated R matrix. Then the entire set of Kalman's equations must be solved for the estimated optimum filter each time the estimates of R and Q are updated.

1.4 Approach to the Problem

In this report, 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 also reduced. In addition, the problem associated with the existence and computation of

R^{-1} 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 a mean-square and with probability one. The results are valid for scalar and vector valued signal and noise processes.

1.5 Organization of the Report

The second chapter presents a comparison of Wiener and Kalman filter theory which serves also to introduce the notation to be used. The review of Kalman's theory lays the foundation for the motivation of the learning criterion.

Chapter III formulates the learning criterion and proves its necessity and sufficiency for optimum filtering. The stochastic algorithms required for performing the adaptation indicated by the learning criterion are then presented. The theory of Stochastic Approximation is invoked to prove the convergence of the algorithms. An extension to time-varying signal and noise statistics is suggested.

Chapter IV applies the theory of Chapter III to specific problems and presents the results of simulations which illustrate the success of this self-adaptive method for (1) different initial values of the filter matrix with R and Q held constant and (2) different values of R and Q with the initial choice of the filter matrix fixed.

Chapter V contains conclusions along with recommendations for further research.

CHAPTER II

OPTIMUM FILTERING

2.1 Introduction and Organization

The objective of this chapter is to present several of the more important results from the theory of optimal estimation. The application of these results to the engineering problem of extracting a stochastic signal from noisy observations or estimating the state of a control system leads to the Wiener and Kalman theories which are developed and compared.

At this point it is necessary to specify exactly what is meant by filtering, and prediction, of a stochastic signal $x(t)$ observed in additive noise $v(t)$.

Definition: Observe the sum $z(nT) = x(nT)$ of the two random processes $x(t)$ and $v(t)$, representing the signal and noise respectively, over the discrete time interval $((n-m)T, nT)$, $n > m$. Filtering is the estimate of $x(\tau)$ at $\tau = nT$.

Prediction is the estimate of $x(\tau)$ for $\tau > nT$.

Both cases will be dealt with in the succeeding pages, but the greatest emphasis is placed on filtering because it is the key operation. Note that even though $x(t)$ and $v(t)$ may be continuous functions of time, the data $z(nT)$ is observed only at discrete times. That is, in this thesis only sampled data is considered.

2.2 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 set of data $Z = \{z(1), \ldots, z(n)\}$. Then $\hat{x}$ will be called the optimal estimate of x given Z if and only if the average loss

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

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

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

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

$$\hat{x} = E\ \{x|Z\} \tag{2.3}$$

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

proof:

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

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

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

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

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

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

$= E_x\{\ell(\hat{x}-x)|Z\}$ by (a)

$= E_y\{\ell(\hat{x} - E \ x|Z - y)|Z\}$

$= E_y\{\ell(\hat{x} - E \ x|Z + y)|Z\}$ by (c)

$= E_y\{\ell(E \ x|Z - \hat{x} - y|Z\}$ by (a)

$= E_y\{\ell(E \ x|Z - \hat{x} + y) \ |Z$ by (c)

$= E_y\{\ell(y - \{E \ x|Z - \hat{x}\})|Z\}$

$= \frac{1}{2} E\{\ell(y + \{E \ x|Z - \hat{x}\} + 1(y$

$-\{E \ x|Z - \hat{x}\})|Z\}$ by (a)

$\geq E\{\ell(\frac{1}{2} \ y + \{E \ x|Z - \hat{x}\} + \frac{1}{2} \ y$

$- \{E \ x|Z - \hat{x}\})|Z\}$ by (b)

$= E\{\ell(y)|Z\}$ with equality iff $\hat{x} = E\{x|Z\}$

Q.E.D.

The class S can be greatly extended if we add two restrictions to the conditional density.

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

(1) symmetric and monotone nonincreasing about its conditional mean.

(2) decreasing rapidly enough so that lim $\ell(y)\ p(y|Z) = 0$, where $y = x - E\{\ x|Z\}$

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

proof: see Viterbi (1966).

Some examples of the $\ell(\cdot) \varepsilon S_1$ are

$$\ell(e) = \begin{cases} K, & |e| \geq k \\ \\ 0, & |e| \geq k \end{cases}$$

$$\ell(e) = K\ |e|$$

$$\ell(e) = K\left[1 - \exp\ (-e^2)\right] .$$

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

In general what follows will concern vector-valued random processes $\underset{\sim}{x}(n)$. Equation 2.2 then becomes

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

where $Z(n) = \{\underset{\sim}{z}(1), \ldots, \underset{\sim}{z}(n)\}$ 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 $\ell\left[\|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|\right] = \|\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\|^2 = \left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]^T \left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]$, then $E\{\hat{\underset{\sim}{x}}(n) \mid Z(n)\}$ minimizes (2.4) without any restraints on $p(\underset{\sim}{x}(n) \mid Z(n))$.

Proof:

$$\begin{aligned} E_{\underset{\sim}{x}}\{[\underset{\sim}{x} - \hat{\underset{\sim}{x}}]^T[\underset{\sim}{x} - \hat{\underset{\sim}{x}}] \mid Z\} &= \hat{\underset{\sim}{x}}^T \hat{\underset{\sim}{x}} - 2\hat{\underset{\sim}{x}}^T E\{\underset{\sim}{x} \mid Z\} + E\{\underset{\sim}{x}^T \underset{\sim}{x} \mid Z\} \\ &= (\hat{\underset{\sim}{x}} - E\{\underset{\sim}{x} \mid Z\})^T (\hat{\underset{\sim}{x}} - E\{\underset{\sim}{x} \mid Z\}) + E\{\underset{\sim}{x}^T\underset{\sim}{x} \mid Z\} - [E\{\underset{\sim}{x} \mid Z\}]^T E\{\underset{\sim}{x} \mid Z\} \\ &\geq E\{\underset{\sim}{x}^T\underset{\sim}{x} \mid Z\} - [E\{\underset{\sim}{x} \mid Z\}]^T E\{\underset{\sim}{x} \mid Z\} \end{aligned}$$

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

2.3 Principle of Orthogonality and the Wiener-Hopf Equation

The contents of theorems 2-1, 2-2, and 2-3 give the "in principal" solution of the Bayes estimation problem for a wide class of loss functions and probability structures. However, the explicit computation of this optimum 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 we have the well known result that $E\{\underset{\sim}{x}(n) \mid Z(n)\}$ is a linear function $T\left[Z(n)\right]$ of the observations $\underset{\sim}{z}(\cdot)$, e.g., see Deutsch (1965). The optimal linear operator $T\left[Z(n)\right]$ can be determined using the orthogonal projection theorem

THEOREM 2-4. Let $\{\underset{\sim}{x}(n)\}$ and $\{\underset{\sim}{z}(n)\}$ be zero mean random sequences. Let $Z(n)$ represent the closed linear manifold[1] generated by the data $\{B\ \underset{\sim}{z}(i) \ldots, \underset{\sim}{z}(n)\}$ where B is the general m x p generator matrix for Z(n)

If either

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

(ii) the estimator $\underset{\sim}{\hat{x}}(n)$ is required to be a linear function $T\left[Z(n)\right]$ of the data $\{\underset{\sim}{z}(i), \ldots, \underset{\sim}{z}(n)\}$ and $\ell\ ||\underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n)|| = \ ||\underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n)||^2$. Then the optimal estimate $\underset{\sim}{\hat{x}}(n)$ of $\underset{\sim}{x}(n)$ is such that the error $\underset{\sim}{e}(n) - \underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n)$ is orthogonal to $Z(n)$, I.e.,

$$\begin{aligned}(\underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n), B\underset{\sim}{z}(j)) &= E\{\ \underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n)\ ^T\ B\underset{\sim}{z}\ (j)\} \\ &= 0 \quad B\underset{\sim}{z}(j) \varepsilon Z(n)\end{aligned} \tag{2.5}$$

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

Proof: see Kalman (1960)

COROLLARY 2-5 $(\underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n), \underset{\sim}{\hat{x}}(n)) = E\{\left[\underset{\sim}{x}(n) - \underset{\sim}{\hat{x}}(n)\right]^T \underset{\sim}{\hat{x}}(n)\} = 0$
where $\hat{x}(n) = T\left[Z(n)\right]$

Under condition (i) of the theorem, the orthogonal projection of $\underset{\sim}{x}(n)$ on $Z(n)$ is identical to the conditional mean $E\{\underset{\sim}{x}(n) \mid Z(n)\}$. Thus, this theorem implies that the optimum linear estimator can not be improved upon unless the random phenomenon are non-Gaussian and, even then, only by assuming knowledge of at least third order probability distribution functions.[2] Consequently we know the general form of $\hat{x}(n)$ in the sampled-data case is

$$\underset{\sim}{\hat{x}}(n) = T\left[Z(n)\right] = \sum_{\nu=i}^{n} A(n,\nu)\underset{\sim}{z}(\nu) \tag{2.6}$$

[2] Given any random sequence, there exists a unique Gaussian random sequence with the same mean and covariance.

where A(n,ν) is an m x p filter matrix. If the data acquisition rate is high enough to be considered continuous, then (2.6) becomes an integral equation. Regardless, A(n,ν) is chosen to satisfy the orthogonal projection theorem. This is the method employed by Kalman (1960) to solve the optimal filter problem.

Kalman and Bucy (1961) used this theorem, i.e., the orthogonality of e(n) and Z(n), to derive the multidimensional Wiener-Hopf equation. The Wiener-Hopf equation is given by the outer product

$$\begin{aligned} \left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]\ )\ (\underset{\sim}{z}(j) &= E\{\left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right] \underset{\sim}{z}^T(j)\} \\ &= \left[0\right],\ \forall\, \underset{\sim}{z}(j) \ \varepsilon\ Z(n) \end{aligned} \qquad (2.7)$$

Since $\hat{\underset{\sim}{x}}(n)$ is given by (2.6), (2.7) can be written

$$\begin{aligned} E\{\left[\underset{\sim}{x}(n) - \sum_{\nu=i}^{n} A(n,\nu)\right] \underset{\sim}{z}(\nu).\ \underset{\sim}{z}^T(j)\} &= E\{\underset{\sim}{x}(n)\ \underset{\sim}{z}^T(j)\} \\ - \sum_{\nu=i}^{n} A(n,\nu)\ E\ \underset{\sim}{z}(\nu)\ \underset{\sim}{z}^T(j)\} &= \left[0\right],\ \forall\, \underset{\sim}{z}(j)\ \varepsilon\ Z(n) \end{aligned} \qquad (2.8)$$

If i = -∞ in (2.8), the sum is assumed to be uniformly convergent so the order of summation and integration may be interchanged. For the scalar case it is obvious that the result given by orthogonal projection theorem equation (2.5) and the Wiener-Hopf equation (2.7) are identical. To show this equivalence in the random vector case, a dual space approach was used. The result is summarized in Theorem 2-6.

THEOREM 2-6: A necessary and sufficient condition for

$$E\left\{\left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right] \underset{\sim}{z}^T(j)\right\} = \left[0\right] \forall \underset{\sim}{z}(j) \varepsilon Z(n) \tag{2.7}$$

where $\hat{\underset{\sim}{x}}(n) = \sum_{\nu = i}^{n} A(n,\nu)\ \underset{\sim}{z}(\nu)$

is that $A(n,\nu)$ be chosen such that

$$E\left\{\left[\underset{\sim}{x}(n) - \underset{\sim\sim}{x}(n)\right]^T B\underset{\sim}{z}(j)\right\} = 0 \quad \forall B\underset{\sim}{z}(j) \varepsilon Z_1(n) \tag{2.5}$$

Proof: see Kalman (1960) and Kalman and Bucy (1961)

COROLLARY 2-7: $E\left\{\left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right] \hat{\underset{\sim}{x}}^T(n)\right\} = \left[0\right]$

This thoorem and the accompany corollary provides a common framework for the filter theory of Wiener and Kalman.

2.4 Wiener and Kalman Filter Theory

The purpose of this section is to present the results of Wiener and Kalman for comparison. The reader is referred to the appropriate references in section 1.2 for a complete derivation. The model used is given by equations (1.3) and (1.4)

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

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

with $$E\{\underset{\sim}{w}(n)\ \underset{\sim}{w}^T(n)\} = Q(n) \tag{1.4}$$

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

Wiener's approach[3] uses the frequency domain[4] and the solution is given in terms of the Z-transform $\hat{A}(Z)$ of the m x p filter matrix A(n-) as illustrated in FIGURE 1. The problem of synthesizing the filter remains. Employment of the frequency domain requires the following restriction.

(1) The system Φ and observation matrix H are time invariant.

(2) The statistics of $\underset{\sim}{w}(n)$ and $\underset{\sim}{\nu}(n)$ are stationary.

(3) The data $\underset{\sim}{z}(n)$ are known for past time, i.e., $i = -\infty$ in equation (2.8).

Under these conditions, equations (2.6) and (2.9) become

$$\hat{x}(n) = \sum_{\nu=-\infty}^{n} A(n-\nu)\ \underset{\sim}{z}(\nu) \qquad (2.9)$$

and

[3] To the author's knowledge, 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).

[4] Since only sampled data is considered, frequency domain means the Z-transform domain.

and

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

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

$$= \left[0\right],\ j\varepsilon\ -\infty, \ldots, n \tag{2.10}$$

By a change of variable (2.10) can be rewritten

$$R_{xz}(\alpha) - \sum_{\tau=0}^{\infty} A(\tau)\ R_{zz}(\alpha-\tau) = \left[0\right],\quad \alpha\varepsilon\{0, \ldots,\infty\} \tag{2.11}$$

The cross-spectral (generating function) matrix representation of (2.11) is

$$\phi_{xz}(Z) - \phi_{zz}(Z)\ A^T(Z) = \left[0\right] \tag{2.12}$$

Since each element of (2.11) is zero, each element of (2.12) is a polynomial in positive powers of Z only. Thus each polynomial element must converge for all Z inside the unit circle. Assuming $\phi_{zz}(Z)$ has a spectral factorization of the form

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

where $\Delta(Z^{-1})$ is a p x p matrix whose elements represent the Z-transform of stable, linear, casual systems (i.e., polynomials containing a constant and positive powers of Z) and have no poles inside the unit circle, then a physically realizable Wiener filter exists. The frequency domain expression for this optimum filter is -

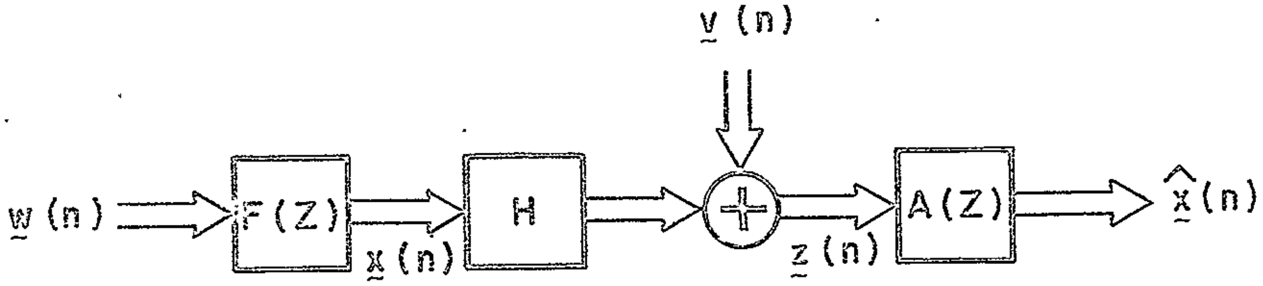


FIG. 2.1 WIENER FILTER MODEL

$$A^T(Z) = \left[\Delta^T(Z)\right]^{-1} \left[\{\Delta^{-1}(Z^{-1})\ \phi_{xz}(Z)\}_c + \Delta^{-1}(Z^{-1})\ \phi_{xz}(Z)\}_+ \right] \quad (2.13)$$

where $\{\ \}_c$ is a matrix whose elements are constants and $\{\ \}_+$ is a matrix whose elements contain only poles inside the unit circle. The "in principle" solution given by (2.13) is not easy to synthesize, and is not suited to machine computation.

Kalman"s time domain approach not only eliminates these two difficulties, but also the three restrictions listed on page 9 . He used the orthogonal projection theorem to obtain the following recursive set of equations for optimum filtering and prediction:

$$\hat{\underset{\sim}{x}}(n) = \Phi\ \hat{\underset{\sim}{x}}(n-1) + K(n) \left[\underset{\sim}{z}(n) - H\ \Phi\ \hat{\underset{\sim}{x}}(n-1) \right] \quad (2.14)$$

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

$$K(n) = \Sigma(n)\ H^T R^{-1}(n) \quad (2.15)$$

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

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

$$\Sigma(n) = \text{Cov}\ \{\underset{\sim\sim}{x}(n)\ \mid\ Z(n)\} \quad (2.16)$$

$$= E\ \{ \left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right] \left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]^T \mid Z(n)\}$$

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

$$= \text{Error Covariance matrix} -$$

$$\Sigma(n+1|n) = E\ \{ \left[\underset{\sim}{x}(n+1) - \Phi\ \hat{\underset{\sim}{x}}(n)\right] \left[\underset{\sim}{x}(n+1) - \Phi\ \hat{\underset{\sim}{x}}(n)\right]^T \mid Z(n)\} \quad (2.17)$$

$$= \Phi\ \Sigma(n)\ \Phi^T + Q(n+1)$$

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

$$\hat{\underset{\sim}{x}}(n+1|n) = \Phi\ \hat{\underset{\sim}{x}}(n)$$

$$= \text{One-step prediction of } \underset{\sim}{x}(n+1)$$

$$= E\ \{\underset{\sim}{x}(n+1)\ \ |\ \ Z(n)\} \text{ for Gaussian Noise}$$

The block diagram for the Kalman filter is shown in Figure 2.2. Note that it is in the form of a closed-loop feedback system. The necessity of knowing the covariance matrices Q(n) of the white plant perturbation (signal generating) noise and R(n) of the white observation noise is obvious from inspection of (2.15) - (2.17).

When restrictions (1) - (3) required for the Wiener approach are satisfied, the Kalman filter is equivalent to the Wiener filter. This must be true from Theorem 2-6. However, the computational superiority of Kalman's method is still evident. It is interesting to determine exactly how the Kalman filter matrix K is related to the optimum Wiener filter A(n-ν). For this case

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

From Theorem 2-6

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

or

$$E\ \left\{\left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]\ \underset{\sim}{x}^T(n)\right\}\ H^T = E\left\{\left[\underset{\sim}{x}(n) - \hat{\underset{\sim}{x}}(n)\right]\ \underset{\sim}{\nu}^T(n)\right\} \qquad (2.19)$$

The left hand side of (2.19) is $\Sigma(n)\ H^T$ from Corollary 2-7, and the right hand side is $E\ \{\hat{\underset{\sim}{x}}(n)\ \underset{\sim}{\nu}^T(n)\}$ since $\underset{\sim}{\nu}(n)$ is independent of $\underset{\sim}{x}(n)$. Therefore,

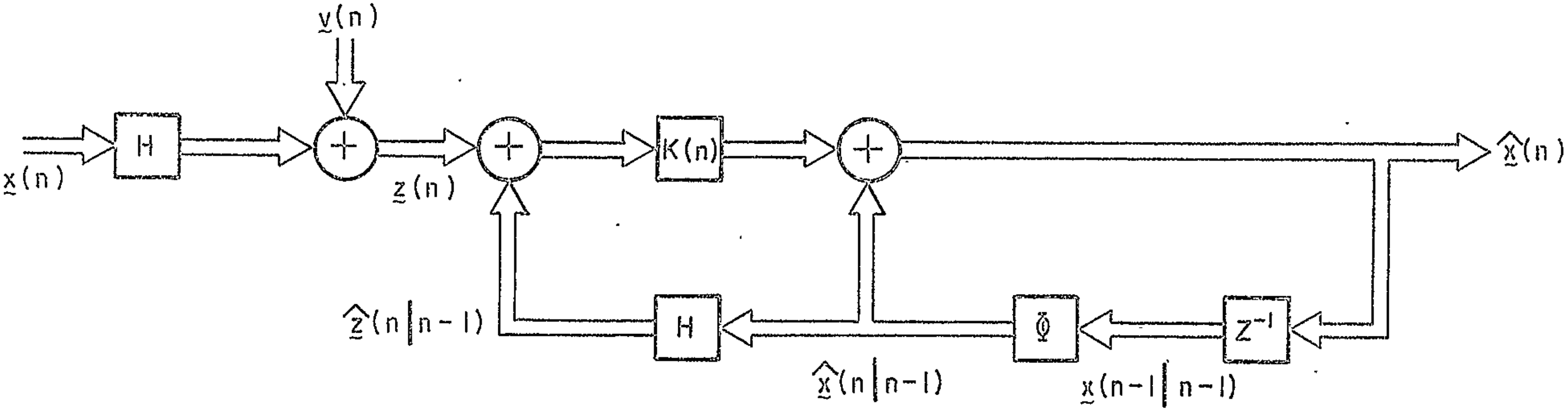

$\underset{\sim}{x}(n)$
H
$\underset{\sim}{v}(n)$
$\underset{\sim}{z}(n)$
K(n)
$\hat{\underset{\sim}{x}}(n)$
$\hat{\underset{\sim}{z}}(n|n-1)$
H
$\hat{\underset{\sim}{x}}(n|n-1)$
Φ
Z^{-1}
$\underset{\sim}{x}(n-1|n-1)$

FIG. 2.2 KALMAN FILTER MODEL

$$\begin{aligned}\Sigma(n)\,H^T &= E\ \{\underset{\sim}{\hat{x}}(n)\ \underset{\sim}{\nu}^T(n)\}\\ &= E\ \{\sum_{\alpha=-\infty}^{n} A(n-\alpha)\ \underset{\sim}{z}(\alpha)\underset{\sim}{\nu}^T(n)\}\\ &= E\ \{\sum_{\alpha=-\infty}^{n} A(n-\alpha)\ \left[H\ \underset{\sim}{x}(\alpha) + \underset{\sim}{\nu}(\alpha)\right]\underset{\sim}{\nu}^T(n)\}\\ &= \sum_{\alpha=-\infty}^{n} A(n-\alpha)\ E\ \{\underset{\sim}{\nu}(\alpha)\ \underset{\sim}{\nu}^T(n)\\ &= A(0)R\end{aligned}$$

which implies

$$\begin{aligned}A(0) &= \Sigma(n)\ H^T\ R^{-1}\\ &\triangleq K\end{aligned}$$

Thus the impulse response of the optimum Wiener filter matrix evaluated at time equal to zero gives the optimum Kalman filter matrix.

2.5 Summary

This chapter has reviewed some of the fundamental concepts of estimation theory and its application. It was shown that for Gaussian noise the optimum estimator is linear. For a given system this important result yields the filter theory of Wiener and Kalman which was reviewed and compared.

CHAPTER III

FORMULATION OF THE LEARNING CRITERION AND THE ASSOCIATED STOCHASTIC ALGORITHMS

3.1 Introduction and Organization of the Chapter

In Chapter II 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 to be developed in this chapter. Stochastic algorithms, based on this criterion, are derived which asymptotically converge to the optimum filter. Stochastic Approximation techniques are invoked to prove this convergence.

3.2 The Learning Criterion

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

(1) It must be a function of measurable and/or calculable quantities.

(2) It must provide information from which convergent algorithms can be derived.

Otherwise, the criterion is meaningless from an engineering point of view.

THEOREM 3-1.

Given the dynamic system -

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

the observation process

$$\underset{\sim}{z}(n) = H\,\underset{\sim}{x}(n) + \underset{\sim}{\nu}(n) \qquad (1.4)$$

and the filtering equation

$$\hat{\underset{\sim}{x}}(n) = \Phi\hat{\underset{\sim}{x}}(n-1) + K_n \left[z(n) - H\,\Phi\,\hat{\underset{\sim}{x}}(n-1)\right] \qquad (2.14)$$

If $K_n = K_{opt}$

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

That is, K_{opt} is the optimum Kalman filter matrix.

Then,

$$E\,\{\delta^T_{n+1}\ \delta j\} \triangleq (\delta_{n+1},\ \delta j) = 0 \ \forall\, j \leq n \qquad (3.1)$$

where,

$\delta j = \underset{\sim}{z}(j) - H\,\Phi\,\hat{\underset{\sim}{x}}(j)$, and conversely.

This theorem is important because it implies that when K_n is not the optimum filter matrix, the residual process $\{\delta_n\}$ is not orthogonal.

THEOREM 3-2. The Learning Criterion

If $E\,\{\delta^T_{n+1}\ \delta j\} = 0 \quad \forall\, j \leq n$ and $E\{Xo\} = 0$,

then,

$$E\,\{\delta_{n+1}\ \delta j^T\} = \left[0\right] \ \forall\, j \leq n,$$

and conversely.

Thus, the lack of orthogonality, when K_n is not equal to K_{opt} is reflected in a non-null correlation matrix between the residuals,

$$E\ \{\ \delta_{n+1}\ \delta j^T\} \triangleq C(n+1-j)\ \neq\ \begin{bmatrix}0\end{bmatrix} \qquad j \leq n \tag{3.2}$$

C(n+1.j) could be used as a basis for learning K_{opt} if a technique can be devised to utilize this correlation between δ_{n+1} and δ_n, to adjust K_n such that $K_n \rightarrow K_{opt}$ as $n \rightarrow \infty$.

Note first, the fact that the correlation between δ_{n+1} and δ_n can be represented by the stationary Markoff sequence

$$\delta_{n+1} = P\ \delta_n + e_n \tag{3.3}$$

where $\{e_n\}$ is a zero mean random sequence. Post-multiplying both sides of (3.3) by δ_n^T and taking the expected value gives

$$E\ \{\delta_{n+1}\ \delta_n^T\} = P_{n+1} E\{\delta_n \delta_n^T\} + E\ \{\ e_n\ \delta_n^T\ \}$$

or

$$C(1) = P\ C(0) + E\{\ e_n\ \ \delta_n^T\}$$

Choosing the state transition matrix $P = C(1)\ C^{-1}(0)$

forces $E\ \{\delta_n\ \delta_n^T\} = \begin{bmatrix}0\end{bmatrix}$. P represents the correlation between δ_n and δ_{n+1}. If P can be forced to approach $\begin{bmatrix}0\end{bmatrix}$ as n approaches ∞, then from theorems 3-1 and 3-2, K_{n+1} approaches K_{opt}. Thus an algorithm is required which uses P to adjust K_{n+1} such that $P \rightarrow \begin{bmatrix}0\end{bmatrix}$ as $n \rightarrow \infty$. Equivalently, the adjustment must force $\delta_{n+1} \rightarrow e_n$ as $n \rightarrow \infty$.

3.2 Development of the Adaptive Algorithms

In the derivation of this algorithm the measurement matrix H is assumed to be invertible so that P can be written in the form

$$P = H \Phi \Delta K_{n+1} \tag{3.4}$$

where ΔK_{n+1} is an arbitrary matrix chosen to satisfy equation (3.4). Let the initial value of the filter matrix be K_n^o, then

$$\hat{x}^o(n) = \Phi \hat{x}^o(n-1) + K_n^o \left[z(n) - H \Phi \hat{x}^o(n-1) \right] \tag{3.5}$$

Rewriting eq (3.3)

$$e_n = \delta_{n+1} \cdot P_o \, \delta n^o$$

$$= z(n+1) - H\Phi\hat{\chi}^o(n-1) - H \Phi \Delta K^o_{n+1} \left[z(n) - H \Phi \hat{x}^o(n-1) \right] \tag{3.6}$$

Substituting (3.5) into (3.6) gives

$$e_n = z(n+1) - H \Phi \left\{ \Phi \hat{x}^o(n-1) + K_n^o \left[z(n) - H \Phi \hat{x}^o(n-1) \right] \right\}$$

$$- H \Phi \Delta K^o_{n+1} \left[z(n) - H \Phi \hat{x}^o(n-1) \right]$$

$$= z(n+1) - H \Phi \left\{ \Phi \hat{x}^o(n-1) + (K_n^o + \Delta K^o_{n+1}) \left[z(n) - H \Phi \hat{x}^o(n-1) \right] \right\}$$

$$= z(n+1) - H \Phi \left\{ \Phi \hat{x}^o(n-1) + K^1_{n+1} \left[z(n) - H \Phi \hat{x}^o(n-1) \right] \right\}$$

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

$$= \delta^1_{n+1} \tag{3.7}$$

Equation (3.7) implies that $P^1 = 0$ in the equation

$$\delta^1_{n+1} = P_1 \delta^o_n + e_n = e_n$$

and since

$$E \left\{ e_n \left[z(n) - H \Phi \hat{x}^o(n-1) \right]^T \right\} = \left[0 \right]$$

$$E \left\{ \left[z(n+1) - H \Phi \hat{x}^1(n) \right] \left[z(n) - H \Phi \hat{x}^o(n-1) \right]^T \right\} = \left[0 \right] \tag{3.8}$$

From Theorem 3-1 and 3-2, ΔK_{n+1} is the correction to K_n^o required to satisfy eqn (3.8). Therefore, under steady-state conditions

$$K_n^o + \Delta K_{n+1} = K_{opt} .$$

However, since the $E\{\delta_{n+1}\delta_n^T\}$ and $E\{\delta_n \delta_n^T\}$ are unknown, ΔK_{n+1} can not be calculated.

But P_{n+1} and, therefore K_{n+1}, can be estimated by using the method of stochastic approximation (Dvoretzky 1956). A detailed survey of stochastic approximation is contained in Hampton 1969.

To provide insight into the derivation of the stochastic algorithm, the problem of determining P is reformulated in a performance index framework. Let L(P) be the expected value of the performance index to be minimized,

$$L(P) = E\{\ell(\delta_{n+1} - \hat{\delta}_{n+1})\} \tag{3.9}$$

where $\ell(\delta_{n+1} - \hat{\delta}_{n+1}) = (\delta_{n+1} - P\ \delta_n)^T (\delta_{n+1} - P\ \delta_n)$, is the performance index.

When L(P) is known (the deterministic case), equation (3.9) can be minimized by solving

$$\nabla_P\ L(P_{opt}) = 0 \tag{3.10}$$

iteratively

$$\hat{P}_{n+1} = \hat{P}_n + \nabla_P\ L(\hat{P}_n)\ B_{n+1} \tag{3.11}$$

under appropriate convergence conditions.

In the case at hand L(P) is not known. This condition is precisely the motivation for stochastic approximation which states that (3.11) may be replaced by the random matrix sequence -

$$\hat{P}_{n+1} = \hat{P}_n + \nabla_P \, \ell(\delta_{n+1} - \hat{\delta}_{n+1}) \, B_{n+1} \tag{3.12}$$

where $B_{n+1} = \alpha_{n+1} W_{n+1}$

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

$$a_n \geq 0 \ , \ \sum_{n=o}^{\infty} a_n = \infty \ \text{ and } \ \sum_{n=o}^{\infty} a_n^2 < \infty$$

and $\{W_n\}$ is a sequence of uniformly bounded linear matrix operators. Under the above conditions the random sequence generated by (3.12) converges to P_{opt} in mean square and with probability one.

Choosing $a_n = 1/n$ and remembering $\hat{P}_n = H \, \Phi \, \Delta \, \hat{K}_n$, then (3.12) becomes

$$\hat{P}_{n+1} = \hat{P}_n + \{(\delta^{o}_{n+1} - \hat{P}_n \, \delta_n^{o}) \, \delta_n^{oT}\} \ \frac{W_{n+1}}{n+1} \tag{3.13}$$

$$= \hat{P}_n + \{ \left[z(n+1) - H \, \Phi \, \hat{x}(\overset{o}{n}) \right] - \hat{P}_n \left[z(n) - H \, \Phi \, \hat{x}(\overset{o}{n}-1) \right] \} \ \delta_n^{o\,T} \ \frac{W_{n+1}}{n+1}$$

$$= \hat{P}_n + \{ z(n+1) - H \left[\Phi \, \hat{x}(\overset{o}{n}) + \Delta \, \hat{K}_n \, (z_n - H \, \Phi \hat{x}(\overset{o}{n}-1)) \right] \} \delta_n^{o\,T} \frac{W_{n+1}}{n+1}$$

$$= \hat{P}_n + \{ z(n+1) - H \, \Phi \left[\Phi \, \hat{x}(\overset{o}{n}-1) + (\hat{K}_n + \Delta\hat{K}_n) \, (z(n) - H \, \Phi \, \hat{x}(\overset{o}{n}-1)) \right] \} \delta_n^{o\,T} \frac{Wn+1}{n+1}$$

$$= \hat{P}_n + \{ z(n+1) - H \, \Phi \, \hat{x}(\overset{1}{n}) \} \ \left[z(n) - H \, \Phi \, \hat{x}(\overset{o}{n}-1) \right]^T \frac{Wn+1}{n+1}$$

$$= H \, \Phi \, \Delta\hat{K}_n + \ \delta^{1}_{n+1} \ (\delta^{o}_n)^T \ \frac{Wn+1}{n+1} \tag{3.14}$$

and

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

In equation (3.13) the expression in braces is the gradient of the performance index and determines the direction of the correction term. W_{n+1} is a weighting matrix and determines the magnitude of the correction term. The choice of W_{n+1} is vitally important since it determines the rate of convergence of the algorithm. From a computational viewpoint it is more efficient to let W_{n+1} be a constant matrix for all n. For this choice the correction term merely follows the local gradient at each stage of iteration. From a statistical viewpoint it is more efficient to let

$$W^{-1}_{n+1} = \frac{1}{n+1} \{nW^{-1}_n + \delta^o_n (\delta^o_n)^T\} \tag{3.15}$$

$$W_{n+1} = \frac{n+1}{n} \{W_n - W_n \delta^o_n \left[n + (\delta^o_n)^T W_n \delta^o_n \right] (\delta^o_n)^T W_n\}$$

For this choice the correction term is such that the performance index is minimized at each iteration stage. Thus, intuitively (3.15) should converge more rapidly, but at the expense of computation time.

Instead of estimating P_{opt} and then calculating ΔK as is required in using the stochastic algorithm of (3.14), it would be desirable to estimate K_{opt} directly. This can be accomplished by re-defining the performance index $\ell(\cdot)$. Let

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

$$\delta_n = z(n) - \Phi \hat{x}(n) \quad \text{and}$$

$$\ell(\Delta_{n+1} - K_{n+1}\,\delta_n) = (\Delta_{n+1} - K_{n+1}\,\delta_n)^T\,(\Delta_{n+1} - K_{n+1}\,\delta_n)$$

Then

$$\hat{K}_{n+1} = \hat{K}_n + \nabla_K\, L(\hat{K}_n)\; B_{n+1} \tag{3.16}$$

$$= \hat{K}_n + \{\, \Delta_{n+1} - \hat{K}_n\,\delta_n\,\}\delta_n^T\; W_{\frac{n+1}{n+1}}$$

If H^{-1} exists (3.16) becomes

$$\hat{K}_{n+1} = \hat{K}_n + \{\, \Phi^{-1}\left[z(n+1) - \Phi\Phi\,\hat{x}(n-1)\right] - \hat{K}_n\left[z(n) - \Phi\,\hat{x}(n)\right]\}\delta_n^T\; W_{\frac{n+1}{n+1}}$$

$$= \hat{K}_n + \{\, \Phi^{-1}\, z(n+1) - \Phi\, x(n-1) - \hat{K}_n\left[z(n) - \Phi\,\hat{x}(n-1)\right]\}\delta_n^T\; W_{\frac{n+1}{n+1}}$$

$$= \hat{K}_n + \Phi^{-1}\left[z_{n+1} - \Phi\,\hat{x}(n)\right]\delta_n^T\; W_{\frac{n+1}{n+1}}$$

$$= \hat{K}_n + \Phi^{-1}\,\delta_{n+1}\,\delta_n^T\; W_{\frac{n+1}{n+1}} \tag{3.17}$$

where

$$W_{n+1}^{-1} = \frac{1}{n+1}\left[nW_n^{-1} + \delta_n\,\delta_n^T\right] \tag{3.18}$$

Equation (3.17) with W_n defined as in (3.18) satisfies the convergence conditions of stochastic approximation. Therefore, $\hat{K}_{n+1}$ converges to K_{opt}, the optimum Kalman filter matrix, in mean square and with probability one.

3.3 Estimation of Γ, Σ, R, and Q

By including an estimate of C(0) in the computational routine, convergent algorithms for the error covariance Γ, the one-step prediction error covariance Σ, the plant perturbation noise covariance Q, and the observation noise covariance R, are easily derived. These algorithms are

$$\hat{C}_n(0) = \hat{C}_{n-1}(0) + \alpha_n \left[\delta_n \delta_n^T - \hat{C}_{n-1}(0) \right] \qquad (3.19)$$

where α_n is chosen to satisfy the requirements of stochastic approximation,

$$\hat{\Sigma}_n = \hat{K}_n \hat{C}_n(0) \qquad (3.20)$$

$$\hat{\Gamma}_n = (I - \hat{K}_n) \hat{\Sigma}_n \qquad (3.21)$$

$$\hat{R}_n = \hat{K}_n^{-1} \hat{\Gamma}_n \qquad (3.22)$$

$$\hat{Q}_n = \hat{\Sigma}_n - \Phi \hat{\Gamma} \Phi^T \qquad (3.23)$$

Equations (3.20) through (3.23) are valid for H = I. Analogous results hold for H ≠ I.

In Chapter IV the experimental success of the algorithms derived in this Chapter are presented.

CHAPTER IV

NUMERICAL SIMULATION OF THE SELF-ADAPTIVE FILTERING ALGORITHMS

4.1 Introduction

In Chapter III, the learning criterion for self-adaptive filtering was formulated and several convergent stochastic algorithms for performing this adaptation were derived. In this Chapter these algorithms will be applied to specific systems. The experimental results were obtained on The University of Arizona's CDC 6400, using the FORTRAN IV language. In interpreting these results it should be pointed out that double precision arithmetic was not used.

4.2 Experimental Results

Given the system defined by

$$x_{n+1} = \Phi\, x_n + W_n \tag{1.3}$$

$$z_n = H\, x_n + \nu_n \tag{1.4}$$

with Q and R unknown, the optimum estimate of x_{n+1} is computed with a Kalman filter of the form

$$\hat{x}_{n+1} = \Phi\, \hat{x}_n + K \left[z_{n+1} - H\Phi\, \hat{x}_n \right] \tag{2.14}$$

Since Q and R are not known the optimum value of K can not be calculated. However, K_{opt} can be learned using the algorithm

$$\hat{K}_{n+1} = \hat{K}_n + \Phi^{-1}\, \delta_{n+1}\, \delta_n^T\, W_{\frac{n+1}{n+1}} \tag{3.17}$$

with

$$W^{-1}_{n+1} = \frac{1}{n+1}\left[nW^{-1}_{n} + \delta_{n}\,\delta_{n}^{T}\right] \tag{3.18}$$

EXAMPLE 1: 1st Order Plant

$\Phi = 1/2,\ H = 1,\ Q = 1,\ R = 1$

$x_{n+1} = 1/2\ x_n + w_n$

$z_n = x_n + \nu_n$

For this 1st order case, it is easily shown that

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

$$= 0.53$$

Figure 4.1 illustrates the adaptive process for an initial value of the Kalman filter

$K_o = 0.0$

which corresponds to the one extreme of assuming the measurements are just noise, i.e., they contain no information. As can be seen K_n as determined by (3.17) has essentially converged to K_{opt} within 200-iterations, with $K_n \rightarrow K_{opt}$ as $n \rightarrow \infty$. Γ, Σ, Q, and R were also estimated using equations (3.19) through (3.24). These results for 1,000 iterations are compared with their actual values in FIG. 4.1

FIGURE 4.2 illustrates the adaptive process for an initial value of K equal to

$K_o = 1.0$

which represent the other extreme of assuming the measurements contain no noise,

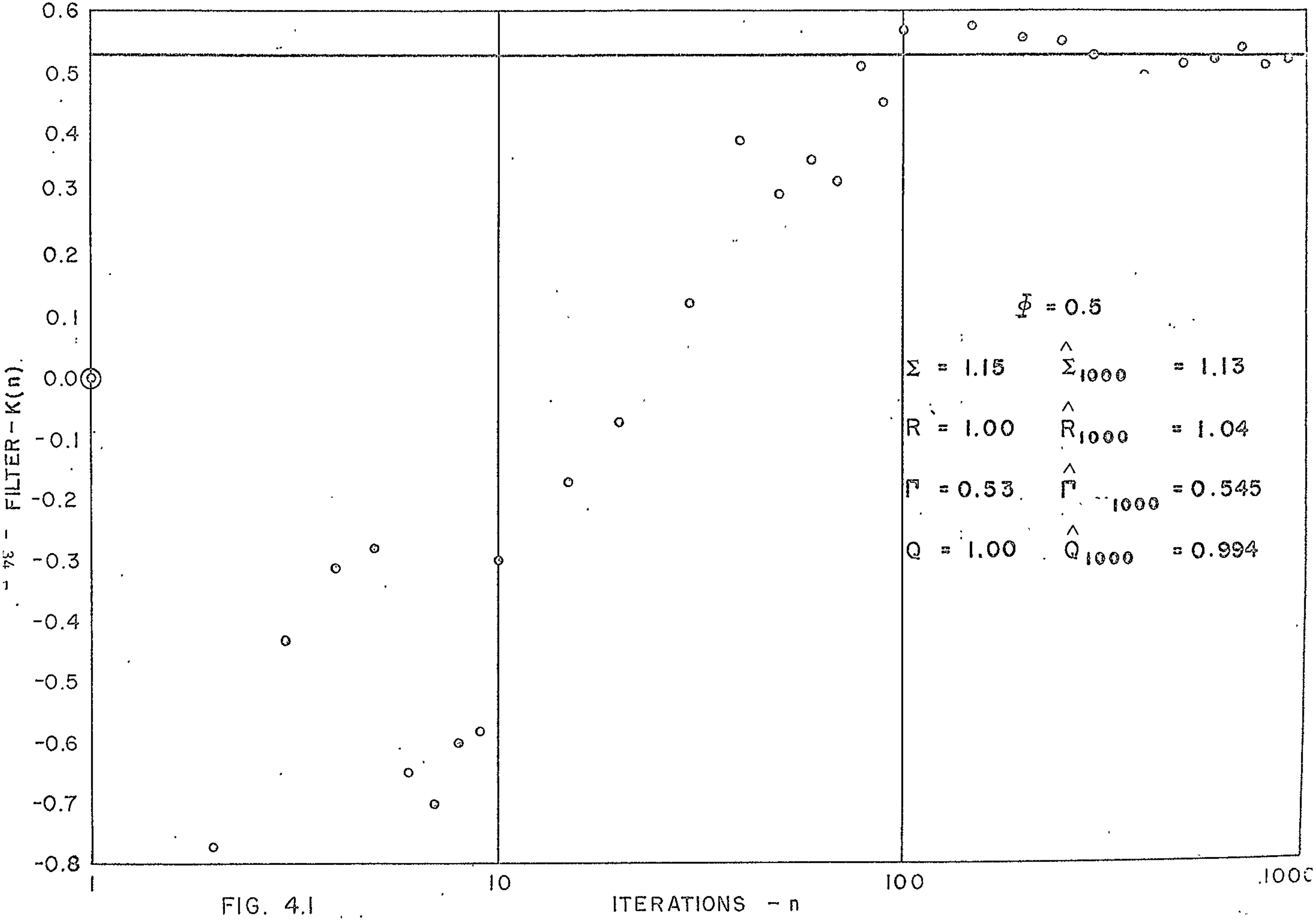

0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
FILTER - K(n)
1
10
100
1000
ITERATIONS - n
$\Phi = 0.5$
$\Sigma = 1.15$ $\hat{\Sigma}_{1000} = 1.13$
$R = 1.00$ $\hat{R}_{1000} = 1.04$
$\Gamma = 0.53$ $\hat{\Gamma}_{1000} = 0.545$
$Q = 1.00$ $\hat{Q}_{1000} = 0.994$

FIG. 4.1

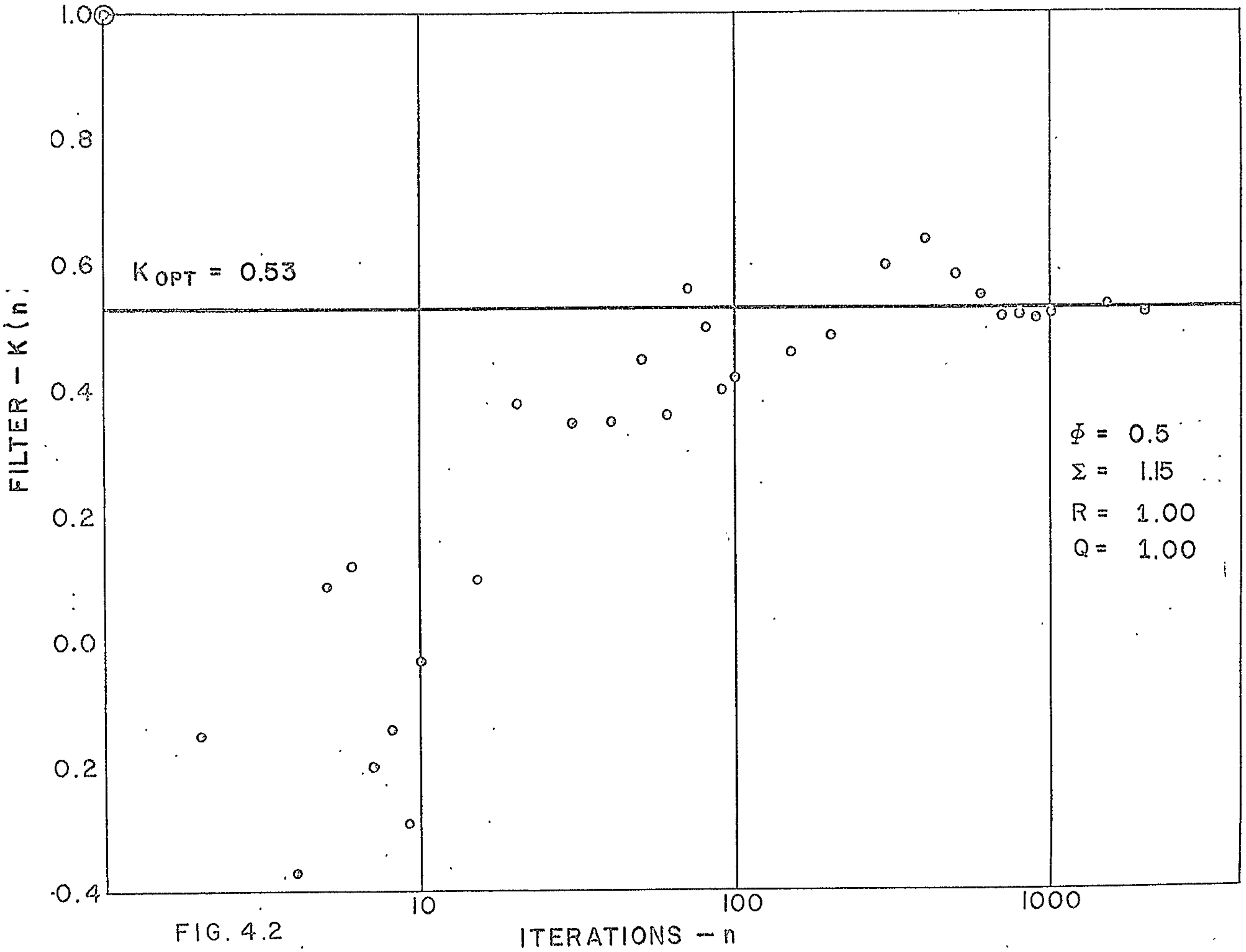

1.00
0.8
0.6
0.4
0.2
0.0
0.2
-0.4
10
100
1000
FILTER — K(n)
ITERATIONS — n
K OPT = 0.53
Φ = 0.5
Σ = 1.15
R = 1.00
Q = 1.00

FIG. 4.2

i.e., they are perfect. Again K_n has essentially converged to K_{opt} within 600-iterations. This indicates that algorithm (3.17) is not sensitive to the initial value of K_o.

Other 1st order problems were considered with Φ being varied from 0.2 to 1.0 (the threshold of instability), and with the values Q and R also being varied. In all cases considered K_n converged to K_{opt} within 2,000--iterations.

EXAMPLE 2: 2nd Order Plant

$$\Phi = \begin{bmatrix} 0.966 & 0.000 \\ 0.155 & 0.894 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1.441 & 0.738 \\ 0.738 & 0.610 \end{bmatrix}$$

$$H = I \qquad , \qquad R = I$$

For this case

$$\Sigma = \begin{bmatrix} 2.00 & 1.00 \\ 1.00 & 1.00 \end{bmatrix}$$

and

$$K_{opt} = \begin{bmatrix} 0.600 & 0.200 \\ 0.200 & 0.400 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}.$$

For this example K_o was chosen to be

$$K_o = \begin{bmatrix} 0.40 & 0.00 \\ 0.00 & 0.60 \end{bmatrix}$$

The results of the adaptation process is shown in FIG. 4.3 (a) through FIG. 4.3 (c). Again the process has essentially converged in 1,000 iterations. Note that symmetry was forced on $k_{12}(n)$ and $k_{21}(n)$. The adaption process for the same system and K_o without forcing symmetry is illustrated in FIG. 4.4 (a) through 4.4 (d).

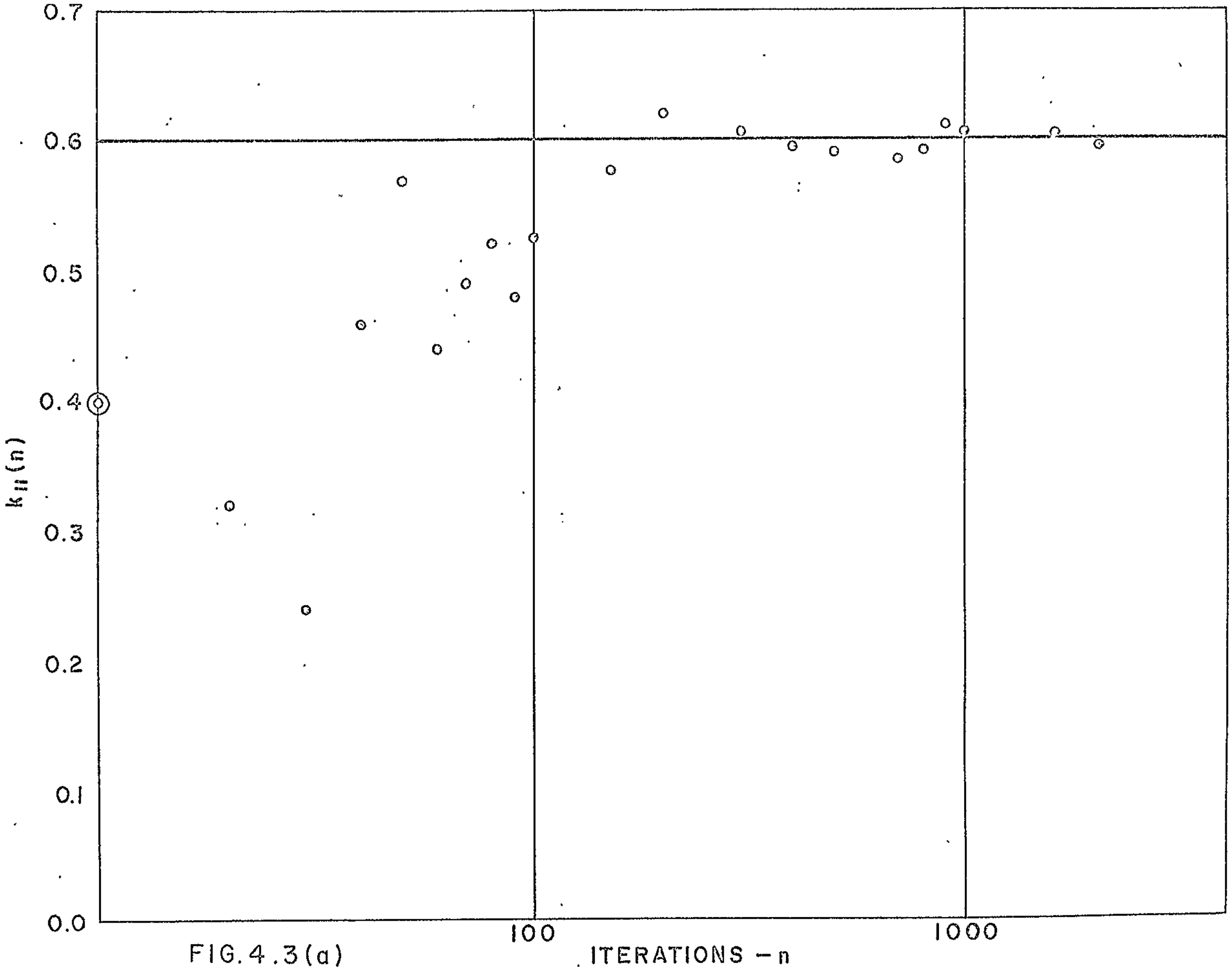


FIG. 4.3(a)

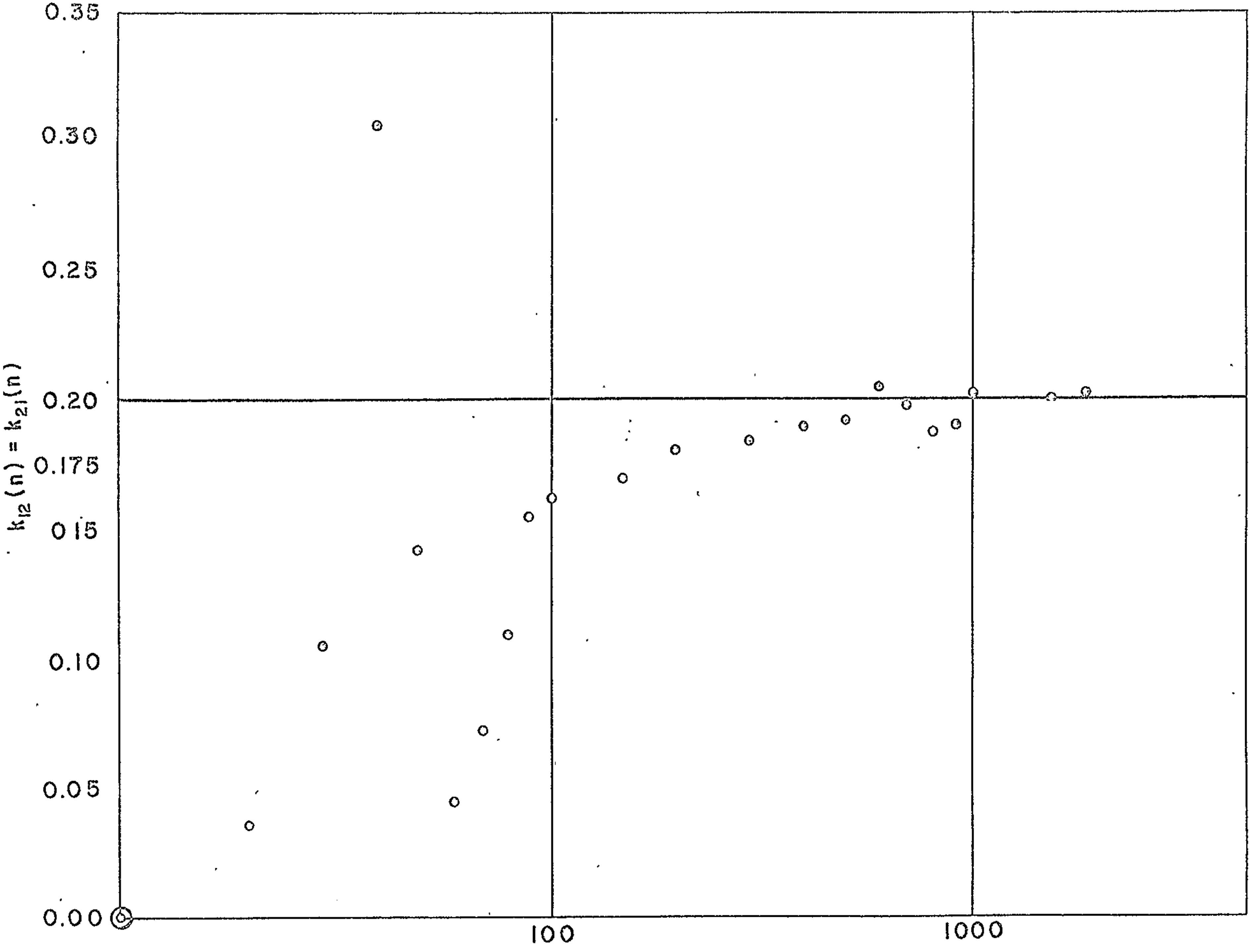

0.35
0.30
0.25
0.20
0.175
0.15
0.10
0.05
0.00
$k_{12}(n) = k_{21}(n)$
100
1000

FIG. 4.3 (b)

ITERATIONS - n

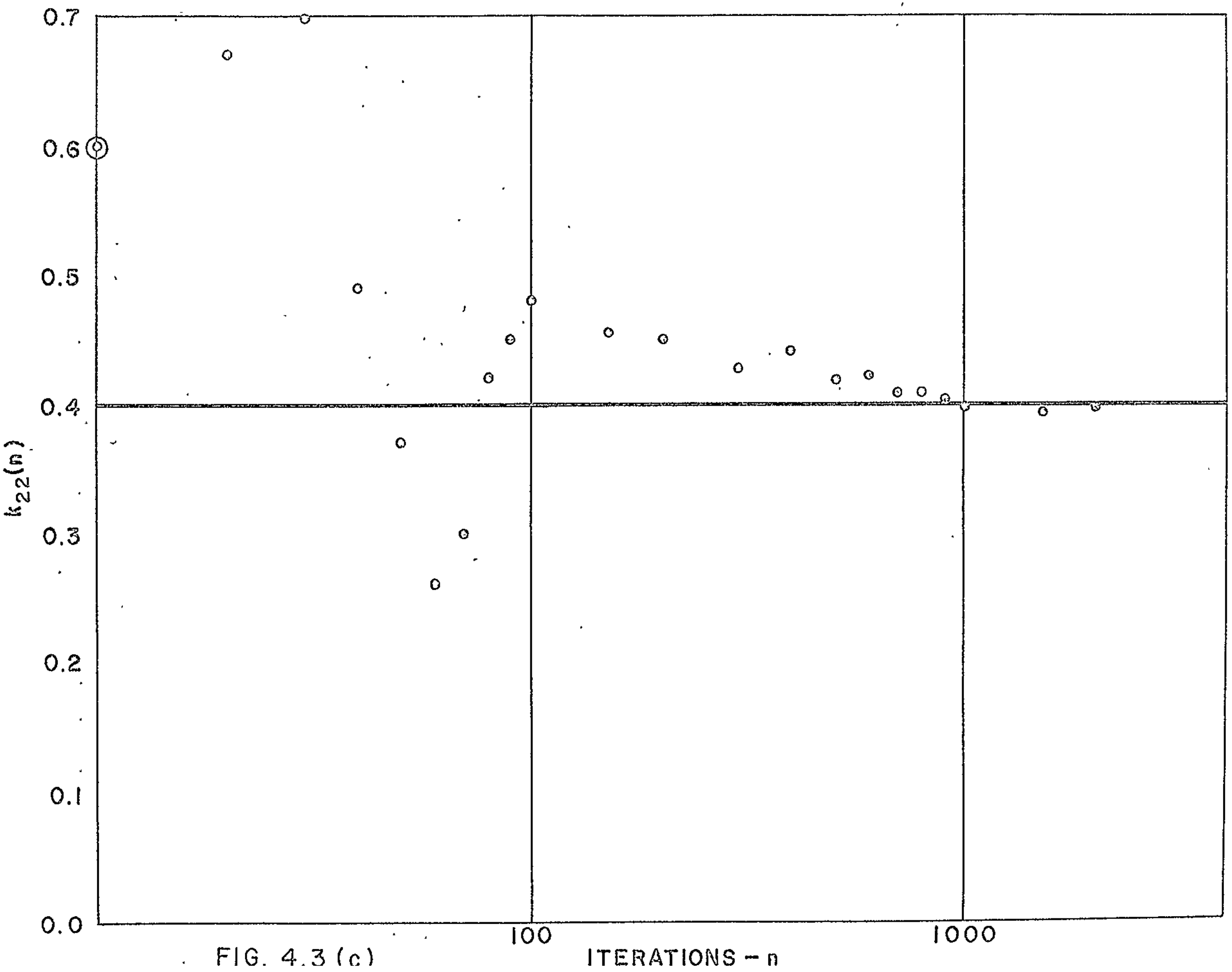


FIG. 4.3 (c)

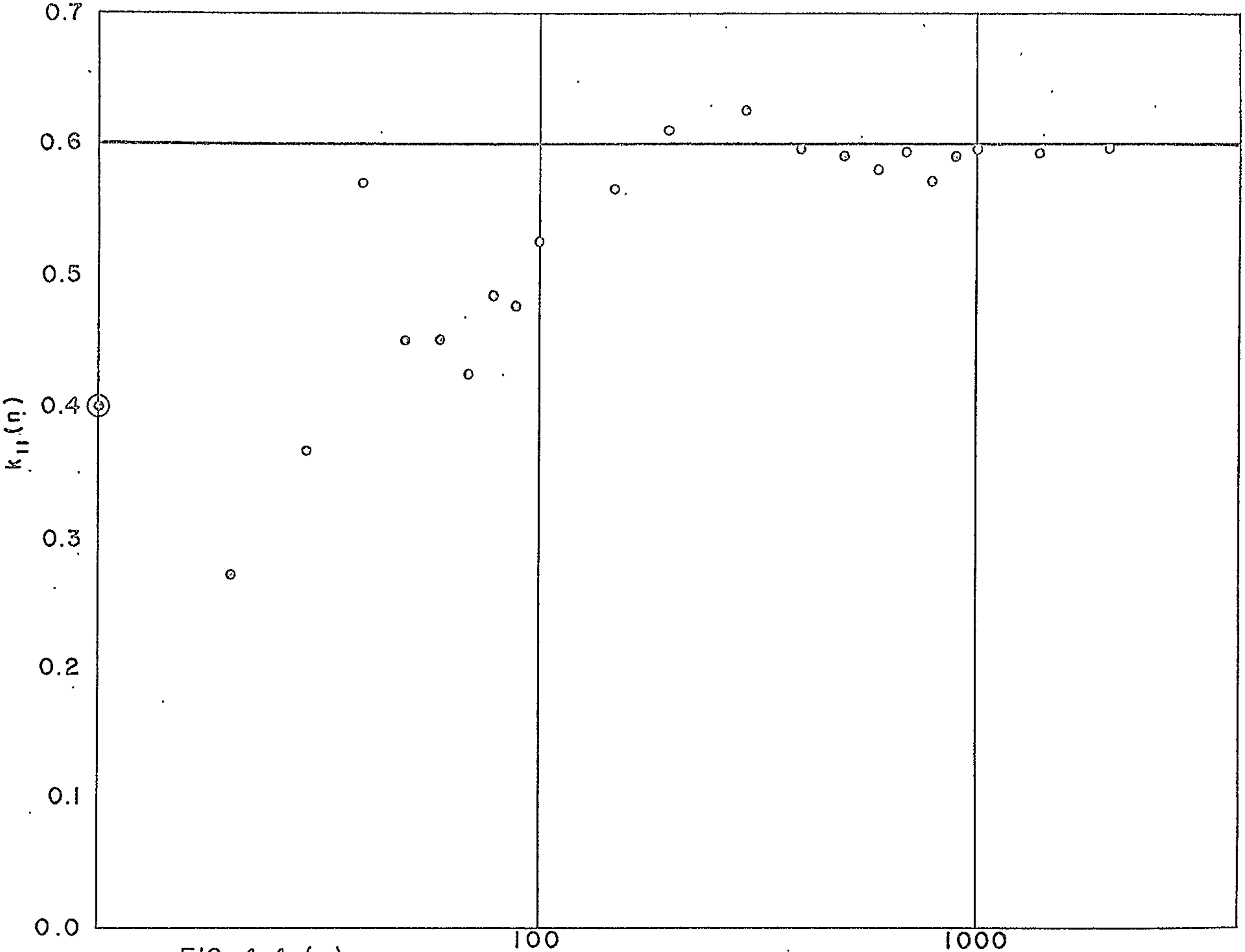


FIG. 4.4 (a)

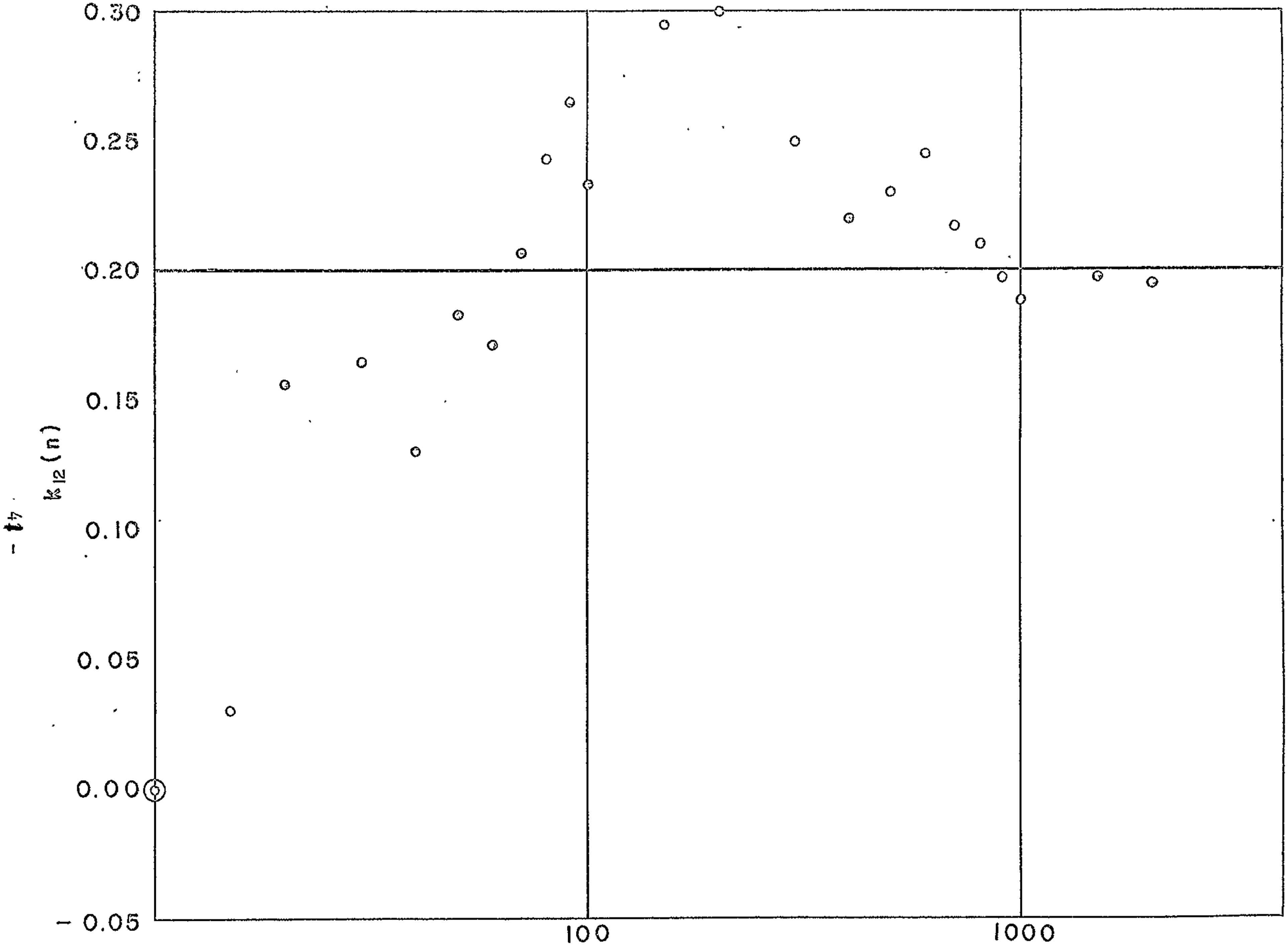


FIG. 4.4 (b)

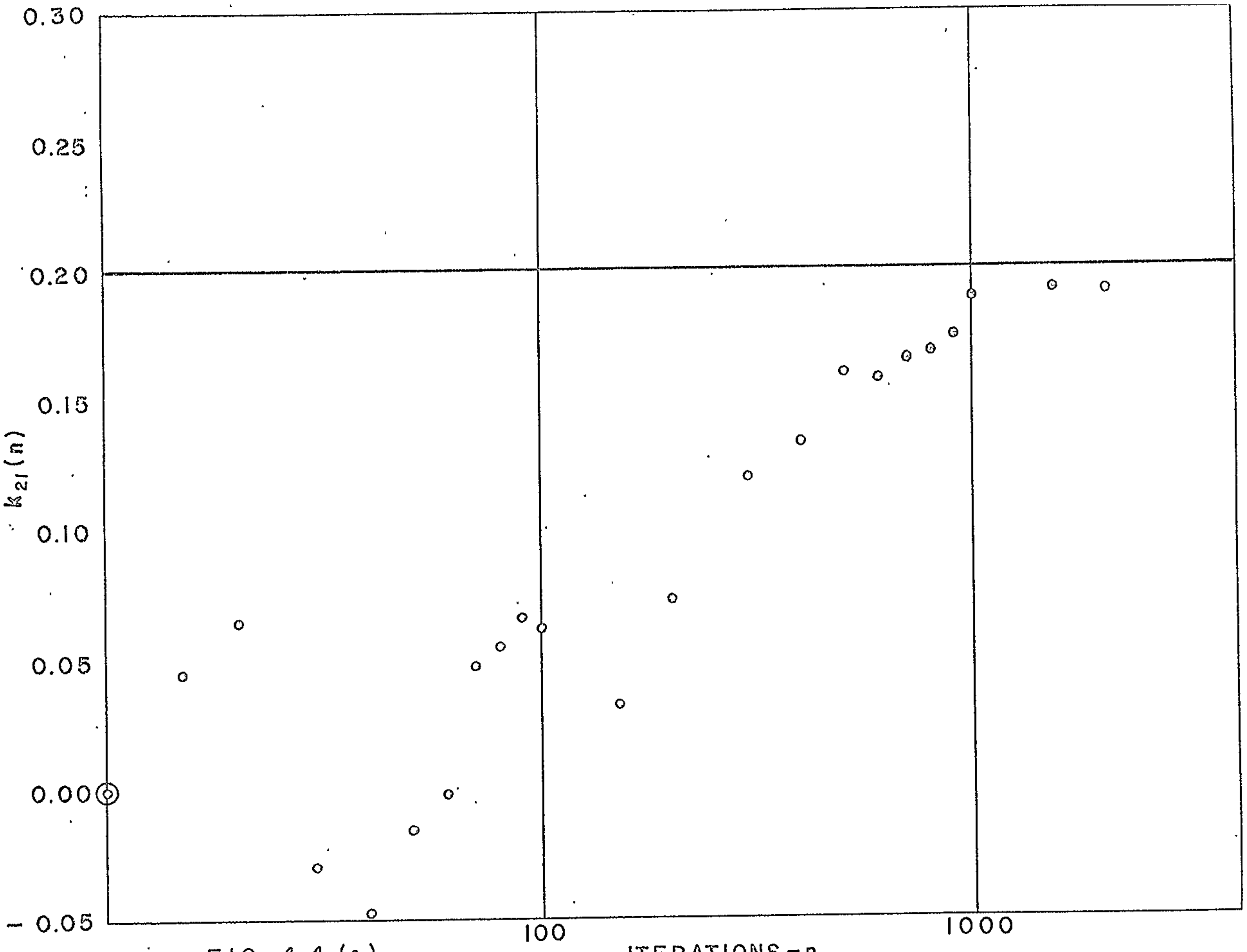


FIG. 4.4 (c)

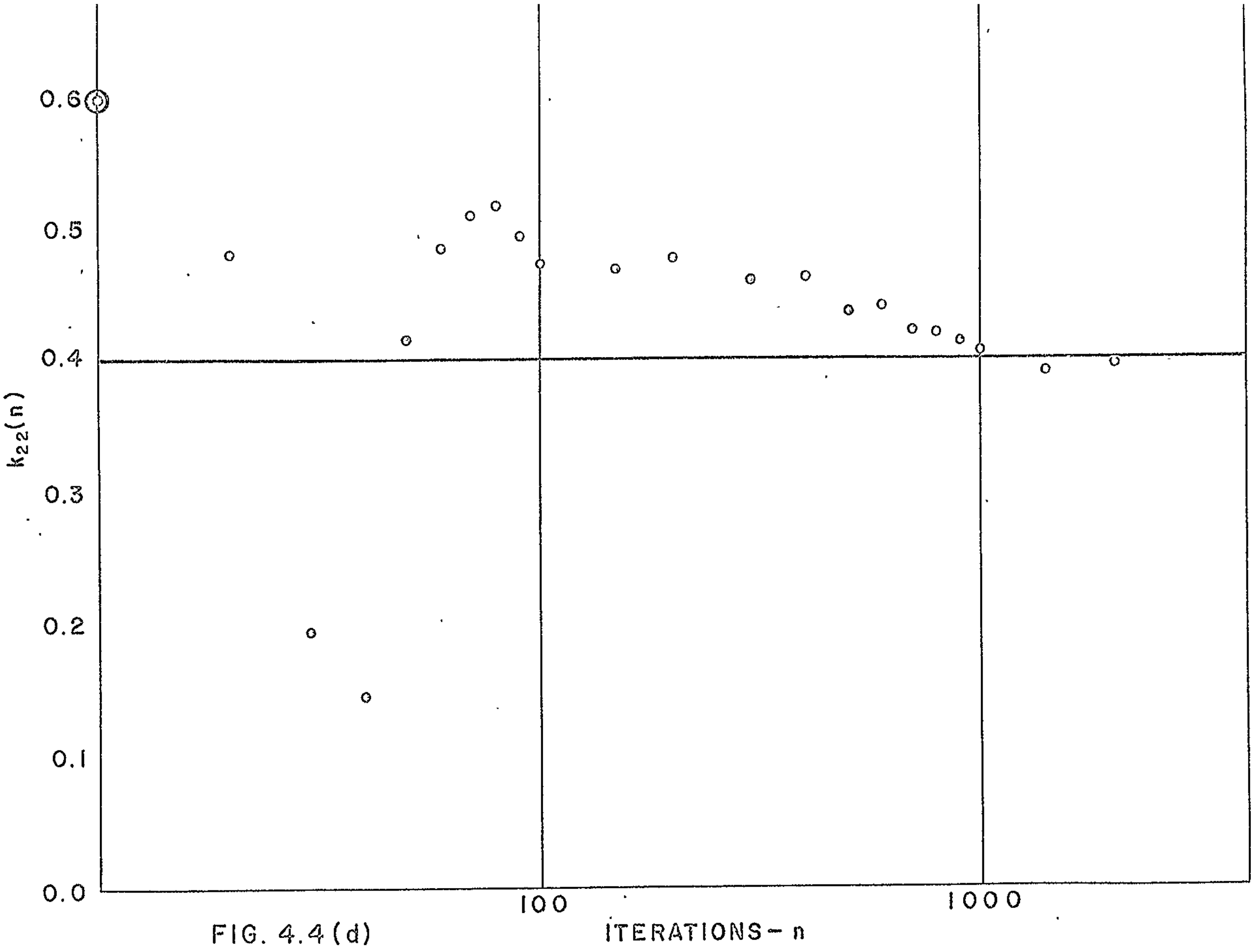


FIG. 4.4(d)

The random sequences $\{w_n\}$ and $\{\nu_n\}$ used for simulating the adaptive process were Gaussian in all examples presented. Analogous results were obtained for uniformly and triangularly distributed sequences.

CHAPTER V

CONCLUSION

This report has presented a self-adaptive technique for learning the optimum Kalman filter matrix in an environment where the covariance matrices of the plant and observation noise are unknown a priori. A future paper will describe this technique in greater depth and extend its application to nonlinear systems and present the experimental results for higher order systems.

REFERENCES

Balakrishran, 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 (Oct. 1963), pp. 237-239.

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.

And R. K. Mehra, "Linear Smoothing Using Measurements Containing Correlated Moise 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 & Sons, New York, 1963.

Dvoretzky, A., "On Stochastic Approximation," Third Berkeley Symposium on Math. Statistics and Probability, 1956.

Hampton, R.L.T., Stochastic Approximation and Its Engineering Application.

Heffes, H., "The Effect of Erroneous Models on the Kalman Filter Response," IEEE Trans. on Automatic Control, AC-11 (July 1966), pp. 541-543.

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.

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.

Lee, R.C.K., Optimal Estimation, Identification and Control, M.I.T. Press, Cambridge, Mass., 1964.

Motyka, P. R. and J. A. Cadyow, "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.

Papoulis, A., Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, 1965.

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.

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 (Sept. 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 (Sept. 1955), pp. 35-37.

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 Inc., New York, 1949.

PART III

THE CONTROL OF NONLINEAR STOCHASTIC CONTROL SYSTEMS UNDER DISCOUNTED PERFORMANCE CRITERIA

by

Cliff A. Harris

January 1970

TABLE OF CONTENTS

LIST OF FIGURES

THE CONTROL OF NONLINEAR STOCHASTIC CONTROL SYSTEMS UNDER DISCOUNTED PERFORMANCE CRITERIA

Introduction

Systems described by difference equations (state equations) and subject to uncertainty as to how they will evolve are of interest in many fields including engineering and economics. The optimal control of such systems was first formulated by Bellman in 1958, and major contributions were subsequently made by Howard (1960), Derman (1964), Blackwell (1962, 1965), and Veinott (1969). However, while much attention has been given to the existence of a solution under various conditions, little work has been directed toward the development of a practical algorithm. It is the purpose of this report and the author's dissertation to develop such an algorithm for discounted performance criteria. A fundamental study of stochastic control systems is made in Chapter II, which establishes the basis for the development of the algorithm in Chapter III. This report consists of these chapters and a summary of the example problems worked to date with a brief explanation of a proposed nuclear rocket control study.

CHAPTER 2

THE CONTROL OF FINITE MARKOV CHAINS

1.1 Introduction

A meaningful analytical examination of the stochastic control problem is found in considering the control of finite Markov chains. Dynamic plant equations and plant noise are modeled by a set of transition probabilities over a finite state space. Each control law is associated with a set of transition probabilities,and a cost function is defined. It is found that the cost function may be minimized by either dynamic programming or Howard's policy iteration. This chapter examines both these methods and the propertie of the cost function under various control laws.

2.2 Finite Markov Chains

Let $(\Omega,\mathcal{F}$, Prob) be a probability triple with Ω the set of elementary events, ω, $\mathcal{F}$, the σ-algebra of subsets of Ω and Prob the probability measure on $\mathcal{F}$. The finite set of real numbers, $\mathcal{X} = \{ {}^1x, {}^2x, \cdots {}^Jx \}$ is called the <u>state</u> <u>space</u> and constitutes the range of the random variable $\underset{\sim}{x}$ mapping Ω onto $\mathcal{X}$. A stochastic process is a sequence

$$\underset{\sim}{X} = \{ \underset{\sim}{x}_n | n = 0,1,2,\cdots \}$$

of random variables.

The stochastic process $\underset{\sim}{X}$ is said to be a Markov chain if for

$$E_n \in \mathcal{F} \quad \text{with } E_n = \{ \omega | \underset{\sim}{x}_n(\omega) = {}^ix, {}^ix \in \mathcal{X} \} ,$$

$$\text{Prob}[E_n | E_0 \cap E_1 \cap E_2 \cdots \cap E_{n-1}] = \text{Prob}[E_n | E_{n-1}] ,$$

whenever $\text{Prob}[E_0 \cap E \cap E_2 \cdots \cap E_{n-1}] \neq 0$. That is ,

$$\text{Prob}[\underset{\sim}{x}_n = {}^j x \mid \underset{\sim}{x}_0 = {}^a x, \underset{\sim}{x}_1 = {}^b x, \ldots, \underset{\sim}{x}_{n-1} = {}^i x] = \text{Prob}[\underset{\sim}{x}_n = {}^j x \mid \underset{\sim}{x}_{n-1} = {}^i x]$$

$$\triangleq p_{ij}(n),$$

where the pij(n) are the transition probabilities defining the chain. The transition (stochastic) matrix for the chain is

$$P(n) = [p_{ij}(n)] \qquad n = 0, 1, \ldots$$

The transition probabilities are related by the Chapman-Kolmogorov equation

$$p_{ij}(m,n) = \sum_{k=1}^{J} p_{ik}(m,r) p_{kj}(r,n) \quad , m \leq r \leq n \tag{2.1}$$

where $p_{ij}(m,n) \triangleq \text{Prob}[\underset{\sim}{x}_n = {}^j x \mid \underset{\sim}{x}_m = {}^i x] \quad , m \leq n .$

A chain is said to be homogeneous if

$$P(n) = P = \text{constant}$$

Then $P(0,n) = [p_{ij}(0,n)] = P^n.$

Let $\mu_j(n) = \text{Prob}[\underset{\sim}{x}_n = {}^j x]$ be the a priori probability that the chain is at state ${}^j x$ at time n, and let

$$\mu(n) = (\mu_1(n), \ldots, \mu_J(n))$$

be the row vector of all a priori probabilities at time n, then

$$\mu(n) = \mu(0) P(0,n)$$

and, for homogeneous chains,

$$\mu(n) = \mu(0) P^n.$$

The states are classified as

(a) ${}^{j}x$ is persistent if Prob $[\, \underset{\sim}{x}_n = {}^{j}x \text{ for some } n \,] = 1$,

(b) ${}^{j}x$ is transient if Prob $[\, \underset{\sim}{x}_n = {}^{j}x \text{ for some } n \,] < 1$

(c) ${}^{j}x$ is aperiodic if

$$\text{g.c.d.}\ \{ n \mid p_{jj}(n) > 0 \} = 1 ,$$

and

(d) ${}^{j}x$ is ergodic if it is persistent and aperiodic (for finite chains).

A chain is said to be ergodic if all states are ergodic. Examples of state classification are given in Figure 2.1, where the transition probabilities are represented by arrows.

The following theorem will be useful in examining the control of Markov chains.

Theorem 1: For a finite homogeneous ergodic Markov chain with transition matrix P, there exists a unique stationary probability distribution μ, and

$$p_{ij}(n) \to \mu_j \text{ as } n \to \infty \qquad \text{geometrically fast.}$$

Or in matrix form,

$$P^n \to 1\mu = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} (\mu_1, \mu_2, \ldots, \mu_J) \qquad \text{geometrically fast}$$

(Doob (1953), Ch. 5 §2).

Thus,

$$\mu(n) = \mu(0)\, P^n \longrightarrow \mu(0)\, 1\mu ,$$

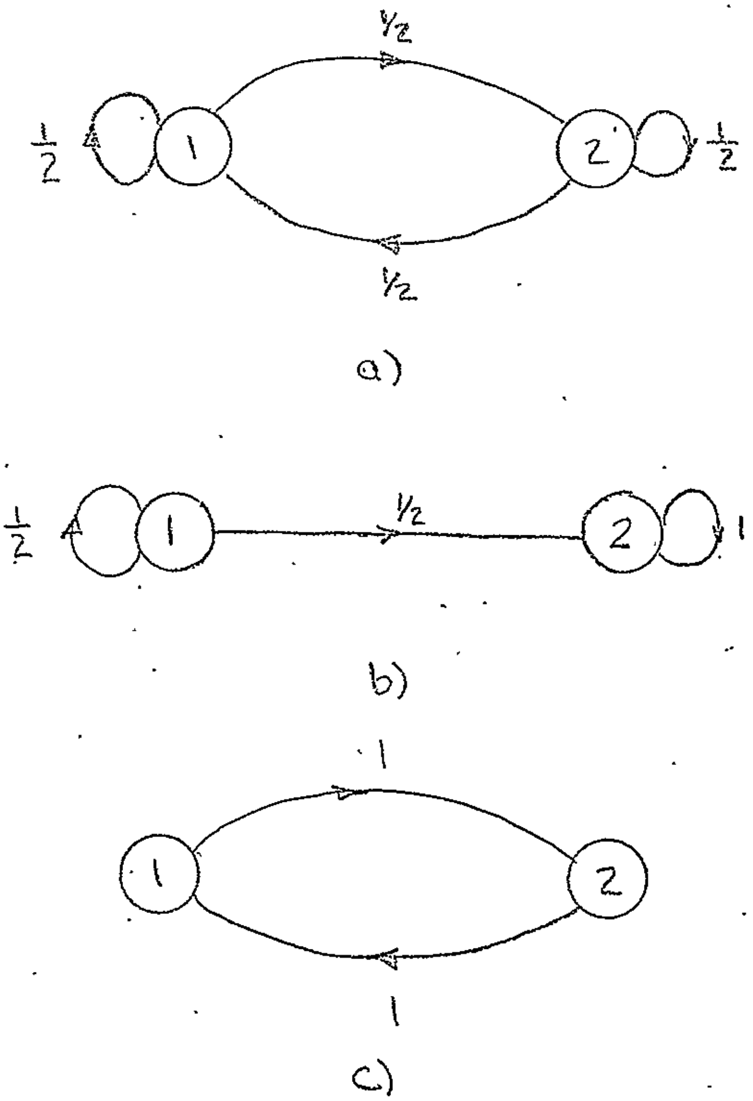


Figure 2.1 a) ergodic chain, b) non-ergodic chain, State 1 is transient, and State 2 is persistent, c) non-ergodic chain, States 1 and 2 are persistent but chain is not aperiodic.

or

$$\mu(n) \rightarrow \mu \quad as \quad n \rightarrow \infty ,$$

and $$\mu = \mu P.$$

2.3 Controlled finite Markov chains

The dynamic system to be controlled has a finite state space

$$X = \{ {}^1x, {}^2x, \ldots, {}^Jx \}$$

and is observed periodically (at every discrete time period). At each time period a control, α , which influences the behavior of the system is applied from a set of possible controls A. As a result of the application of the control $\alpha_k \in A$ with the system in state $x_k \in X$ at time k. there is a time independent,

(1) stage cost $0 < \ell(x_k, \alpha_k) < \infty$ incurred, and

(2) transition of the system from x_k at time $t = k$ to $x_{k+1} \in X$ at time $t = k+1$ with

$$p_{ij}(\alpha_k) = \mathrm{Prob}\left[\underset{\sim}{x}_{k+1} = {}^jx \,\middle|\, x_k = {}^ix , \alpha_k \right] .$$

There is also a discount factor, β , $0 \leq \beta < 1$; whereby, the cost $\ell(x,\alpha)$ for being in state x and applying a control α n periods into the future has a discounted cost of $\beta^n \ell(x,u)$ at the present.

Let U denote the set of control functions u from X into A (i.e., $u \in U$ implies $u(x) \in A$ for all $x \in X$). A policy, π, specifies a sequence of control functions for all time; $\pi = \{ u_0, u_1, \ldots \}$. Thus, at time k , with the system in state x_k, the control $u_k \in A$

is applied. A stationary policy is a policy for which $u_n = u$, $n=0,1,\cdots$ i.e., $\pi = \{u, u, \cdots\} \triangleq u^{\infty}$

Let
$$L(u) = [\ell({}^1x, u({}^1x)), \cdots, \ell({}^Jx, u({}^Jx))]^T$$
$$= [\ell_1(u), \ell_2(u), \cdots, \ell_J(u)]^T$$

be the column vector representation of the stage cost for all states under the control $u \in U$. Let $P(u)$ be the $J \times J$ Markov transition matrix for the control u in the Markov chain established by the policy π.

$$P(u) = [\,p_{ij}(u)]$$

Thus, by the Chapman-Kolmogorov equation, the transition matrix from time $t=0$ to $t=n$ is

$$P(n,\pi) = P(u_0)\,P(u_1)\cdots P(u_n).$$

For the policy π and the initial state x_0 the total expected cost vector is
$$V(\pi) = (v(1,\pi), v(2,\pi), \cdots, v(J,\pi))^T$$

where

$$v(i,\pi) = E\left\{ \sum_{n=0}^{\infty} \beta^n \ell(\underset{\sim}{x}_n, u_n(\underset{\sim}{x}_n)) \,\middle|\, \underset{\sim}{x}_0 = {}^ix, \pi \right\} \qquad (2.3)$$

or

$$= \ell_i(u_0) + \beta E\left\{ \sum \beta^{n-1}(\underset{\sim}{x}_n, u_n(\underset{\sim}{x}_n)) \,\middle|\, \underset{\sim}{x}_0 = {}^ix, \pi \right\}$$

$$= \ell_i(u_0) + \beta E\left\{ E\left\{ \sum_{n=1}^{\infty} \beta^{n-1} \ell(\underset{\sim}{x}_n, u_n(x_n)) \,\middle|\, \underset{\sim}{x}_1 \right\} \,\middle|\, \underset{\sim}{x}_0 = {}^ix, \pi \right\}$$

$$= \ell_i(u_0) + \beta E\left\{ v(\underset{\sim}{x}_1, \bar{\pi}) \,\middle|\, \underset{\sim}{x}_0 = {}^ix, u_0 \right\}$$

where $\bar{\pi} = \{u_1, u_2, \cdots\}$.

Thus,

$$v(i,\pi) = \ell(i,u_0) + \beta \sum_{j=1}^{N} p_{ij}(u_0)\, v(j,\bar{\pi}) , \qquad (2.4)$$

or, in vector form,

$$\begin{aligned} V(\pi) &= L(u_0) + \beta P(u_0) V(\bar{\pi}) \\ &= \sum_{n=0}^{\infty} \beta^n P(n,\pi) L(u_n) , \quad P^0(\pi) = I . \end{aligned}$$

It is desired, if possible, to obtain a policy π^* with the lowest cost,

$$V(\pi^*) \leqslant V(\pi) \qquad \text{for all } \pi$$

where $\leqslant$ means

$v(i,\pi^*) \leqslant v(i,\pi)$ for all π and $i = 1,\ldots,N$. Such a policy π^* is defined as optimal.

2.4 Effect of general policies

The theorems in this section are due to Blackwell (1962) and Howard (1960) and cast light on the classes of policies that it is profitable to study. The proofs which follow Blackwell almost verbatim are included for the insight they provide into the behavior of the expected cost function, $V(\pi)$, under different types of policies.

Lemma. There exists a $f \in F$ such that for an arbitrary $V = (v_1, v_2, \ldots, v_N)'$ and any $u \in F$

$$L(f) + \beta P(f) V \leqslant L(u) + \beta P(u) V .$$

Proof. Consider the i^{th} element

Let , $\varphi_i(\alpha) = \ell({}^i x, \alpha) + \beta \sum_j p_{ij}(\alpha)\, v_j$ for any $\alpha \in A$.

Obviously, $\varphi_i(\alpha) > 0$. Thus, the set $\{\varphi_i(\alpha) \mid \alpha \in A\}$ has a lower bound of zero and hence a greatest lower bound for say $\alpha_i \in A$. The control function f such that $f({}^i x) = \alpha_i$ satisfies the lemma.

Theorem 2 (Blackwell) If there is an optimal policy $\pi^* = \{u_0, u_1, \cdots\}$, there is an optimal policy which is stationary.

Proof. By hypothesis,

$$V(\pi^*) \leq V(\pi) \qquad \text{for all } \pi.$$

$$V(\pi^*) = L(u_0) + \beta P(u_0) V(\pi_1),$$

where $\pi_1 = \{u_1, u_2, \cdots\}$.

Also, $V(\pi^*) \leq V(\pi_1)$,

thus, $V(\pi^*) \geq L(u_0) + \beta P(u_0) V(\pi^*)$.

By the lemma, there exists a $f \in U$ such that

$$V(\pi^*) \geq L(f) + \beta P(f) V(\pi^*) \quad ,$$

and , again,

$$\begin{aligned} V(\pi^*) &\geq L(f) + \beta P(f)\left[L(u_0) + \beta P(u_0) V(\pi_1)\right], \\ &\geq L(f) + \beta P(f)\left[L(u_0) + \beta P(u_0) V(\pi^*)\right] \\ &\geq L(f) + \beta P(f)\left[L(f) + \beta P(f) V(\pi^*)\right] \\ &\geq L(f) + \beta P(f) L(f) + \beta^2 P(f) V(\pi^*) . \end{aligned}$$

By continuing this process,

$$V(\pi^*) \geqslant \sum_{n=0}^{n-1} \beta^n P^n(f) L(f) + \beta^N P^N(f) V(\pi^*) .$$

As $N \to \infty$, $\beta^N P^N(f) V(\pi^*) \to 0$ since $\beta < 1$ and $P^N(f)$ is a stochastic matrix.

Thus, as $N \to \infty$

$$V(\pi^*) \geqslant V(f^\infty)$$

but since π^* is optimal

$$V(\pi^*) = V(f^\infty)$$

and f^∞ is a optimal stationary policy.

Theorem 3 (Blackwell) Let $\pi = \{u_0, u_1, \ldots\}$ and $\pi' = \{f, u_0, u_1, \ldots\}$. If

$$V(\pi) \leqslant V(\pi') \quad \text{for all} \quad f \in U ,$$

then π is optimal.

Proof. By hypothesis,

$$L(f) + \beta P(f) V(\pi) \geqslant V(\pi) \qquad \text{for all} \quad f \in U$$

or, in particular,

$$L(f_N) + \beta P(f_N) V(\pi) \geqslant V(\pi) \qquad \text{for all} \quad f_N \in U$$

$$L(f_{N-1}) + \beta P(f_{N-1}) V(\pi) \geqslant V(\pi) \qquad \text{for all} \quad f_{N-1} \in U$$

$$\vdots$$

$$L(f_1) + \beta P(f_1) V(\pi) \geqslant V(\pi) \qquad \text{for all} \quad f_1 \in U$$

Thus,

$$L(f_1) + \beta P(f_1)\left[L(f_2) + \beta P(f_2) V(\pi)\right] \geqslant V(\pi) ,$$

or

$$L(f_1) + \beta P(f_1) L(f_2) + \beta^2 P(f_1) P(f_2) V(\pi) \geqslant V(\pi) .$$

Continuing this substitution process for the policy $\pi'' = \{f_1, f_2, \ldots, f_N, u_0, u_1, \ldots\}$

$$L(f_1) + \beta P(f_1) L(f_2) + \cdots + \beta^{N-1} P(f_1) P(f_2) \cdots P(f_{N-1}) L(f_N)$$
$$+ \beta^N P(f_1) \cdots P(f_N) V(\pi) \geqslant V\pi ,$$

or

$$V(\pi'') \geqslant V(\pi) .$$

Again as $N \to \infty$, $\beta^N P(f_1)P(f_2)\cdots P(f_N)V(\pi) \to 0$.

Each f_i is an arbitrary element of U ; thus as $N \to \infty$ $V(\pi'')$ becomes the cost of any policy. That is .

$$V(\pi) \leqslant V(\hat{\pi}) \text{ for any } \hat{\pi} .$$

Thus π is optimal.

Theorem 4 (Blackwell) Let $\pi = \{u_0, u_1, \cdots\}$ and $\pi' = \{f, u_0, u_1, \cdots\}$. If $V(\pi') < V(\pi)$, then for the stationary policy f^∞, $V(f^\infty) < V(\pi)$. ($<$ means $\leqslant$ for all elements with $<$ for some element)

Proof. By hypothesis,

$$L(f) + \beta P(f)V(\pi) < V(\pi) ,$$

$$L(f) + \beta P(f)[L(f) + \beta P(f)V(\pi)] < V(\pi),$$

thus,

$$L(f) + \beta P(f)L(f) + \beta^2 P^2(f)V(\pi) < V(\pi) .$$

Continuing this substitution process,

$$\sum_{n=0}^{N-1} \beta^n P^n(f)L(f) + \beta^N P^N(f)V(\pi) < V(\pi) .$$

Once again, as $N \to \infty$, $\beta^N P^N(f)V(\pi) \to 0$, and

$$V(f^\infty) = \sum_{n=0}^{\infty} \beta^n P^n(f)L(f) < V(\pi),$$

completing the proof.

Theorem 5 (Howard) If A is finite, then there is an optimal stationary policy.

Proof. Consider any stationary policy g^∞, then either

$$\text{(a)}\quad \ell({}^i x, \alpha_i) + \beta \sum_j p_{ij}(\alpha_i)\, v(j, g^\infty) \geq v(i, g^\infty) \qquad \text{for all } \alpha_i \in A,\ i = 1, 2, \ldots, J$$

or

$$\text{(b)}\quad \ell({}^i x, \alpha_i) + \beta \sum_j p_{ij}(\alpha_i)\, v(j, g^\infty) < v(i, g^\infty)$$

for some $\alpha_i \in A$

and some i

If (a) holds, then for any $f \in U$, $f({}^i x) = \alpha_i$, the policy $\pi' \doteq (f, g, g, \ldots)$ is more costly than the stationary policy g^∞, i.e.,

$$V(g^\infty) \leq V(\pi')$$

and by Theorem 3 g^∞ is optimal. On the other hand, if g^∞ is not optimal, i.e., there is some i for which (b) holds, then a new control function, u, is defined such that for all i,

$$u({}^i x) = \begin{cases} g({}^i x) & \text{, for case (a)} \\ \alpha_i & \text{, for case (b).} \end{cases}$$

Then by the construction of u, for the policy $\pi_u \doteq \{u, g, g, \ldots\}$

$$V(\pi_u) < V(g^\infty)$$

By Theorem 4,

$$V(u^\infty) < V(g^\infty).$$

Thus, we have a policy, u^∞, which improves upon g^∞. Since A is finite, there are only a finite number of stationary policies. Thus, there is one which has no improvement and is optimal.

The motivation for restricting the class of control laws studied to those that are stationary is contained in Theorem 2; it is seen that any optimal policy may be replaced by a stationary optimal policy. Theorem 5 lays the basis for a constructive method of finding this optimal stationary policy, Howard's iteration in policy space. In the next section this procedure is explained. The set of admissible control policies is taken to be stationary; thus, for notational convenience the policy $u^{\infty} = \{u, u, \ldots\}$ and the control function $u \in \mho$ are considered to be equivalent, and $V(u) = V(u^{\infty})$.

2.5 Howard's policy iteration for $0 \leq \beta < 1$

Before the method of policy improvement contained in the proof of Theorem 5 can be applied, there must be a means of obtaining the expected cost vector, $V(u)$, for any $u \in \mho$. Consider any stationary policy, u , over n stages, then let,

$$\begin{aligned} v_n(i,u) &\triangleq E\left\{\sum_{k=0}^{n} \beta^k \ell(\underset{\sim}{x}_k, u(\underset{\sim}{x}_k)) \Big| \underset{\sim}{x}_0 = {}^i x\right\} \\ &= \ell_i(u) + \beta E\left\{E\left\{\sum_{0}^{n} \beta^{k-1} \ell(\underset{\sim}{x}_k, u(x_k)) \Big| \underset{\sim}{x}_1\right\} \Big| \underset{\sim}{x}_0 = {}^i x\right\} \\ &= \ell_i(u) + \beta \sum_{j} p_{ij}(u)\, v_{n-1}(j,u), \end{aligned} \qquad (2.6)$$

or in matrix form

$$\begin{aligned} V_n(u) &= L(u) + \beta P(u) V_{n-1}(u) \\ &= L(u) + \beta P(u) L(u) + \cdots + \beta^n \overset{n}{P}(u) L(u) \end{aligned} \qquad (2.7)$$

The stage cost function $\ell({}^i x, u({}^i x))$ is bounded for all i by definition. Let this bound be M. Then

$$V_n(u) \leq M\mathbb{1} + \beta P(u) M\mathbb{1} + \cdots + \beta^n \overset{n}{P}(u) M\mathbb{1}$$

$$\mathbb{1} = (1, 1, \ldots, 1)^T$$

$$V_n(u) \leqslant (1+\beta+\cdots+\beta^n)M\mathbf{1}$$
$$< \frac{M}{1-\beta}\mathbf{1} \quad \text{for all } 0 \leqslant \beta < 1$$

It is apparent that the sequence

$$v_0(i,u),\, v_2(i,u),\, \dots,\, v_n(i,u);\, \dots$$

is monotonically increasing for all i. Since $v_n(i,u)$ is bounded, it follows that the limit exits.

Say, $\lim_{n \to \infty} v_n(i,u) = v(i,u)$.

This limit is

$$v(i,u) = E\left\{ \sum_{k=0}^{\infty} \beta^n \ell(\underset{\sim}{x}_n, u(\underset{\sim}{x}_n)) \,\middle|\, \underset{\sim}{x}_0 = x \right\},$$

or the total excepted cost of applying the policy $u \in U$ from (2.3).

Again taking the limit as $n \longrightarrow \infty$ from (2.6),

$$v(i,u) = \ell_i(u) + \beta \sum_j p_{ij}(u)\, v(j,u), \tag{2.8}$$

or $V(u) = L(u) + \beta P(u) V(u)$

Thus, $[I - \beta P(u)] V(u) = L(u)$;

and

$$V(u) = [I - \beta P(u)]^{-1} L(u), \tag{2.10}$$

if the inverse exists.

To establish the existence of the inverse, consider an arbitrary stochastic matrix P. $[I - \beta P]^{-1}$ exists if and only if $\det[I - \beta P] \neq 0$, or $\det[\lambda I - P] \neq 0$ where $\lambda = \frac{1}{\beta}$, $\beta \neq 0$. However, for a stochastic matrix P, all eigenvalues are of magnitude equal to or less than one. Thus, $\det[\lambda I - P] = 0$ only if $|\lambda| \leq 1$, but $0 \leq \beta < 1$ implies $|\lambda| > 1$. Therefore, $\det[\lambda I - P] \neq 0$, or equivalently $\det[I - \beta P] \neq 0$, and the inverse exists.

Another useful result follows immediately. For a fixed policy $u \in U$ the cost $V(u)$ is a continuous function of β. Consider

$$V(u) = [I - \beta P(u)]^{-1} L(u).$$

It is apparent that the elements of the inverse are rational functions of β with no singularities for $0 \leq \beta < 1$. Thus, $v(i,u)$ is a continuous function of β

Howard's policy iteration is a two-step iterative process as follows:

(1) for a given stationary policy $u \in U$ determine

$$V(u) = [I - \beta P(u)]^{-1} L(u),$$

and go to step 2 with $V = V(u)$;

(2) for the cost function $V = (v_1, v_2, \cdots, v_J)^T$ select $u \in U$ such that $u({}^{i}x) \in A$ minimizes

$$\ell({}^{i}x, u({}^{i}x)) + \beta \sum_j p_{ij}(u({}^{i}x)) v_j \quad , \quad i = 1, \cdots, J$$

and repeat step 1.

The process is terminated when step 2 yields no further improvement. The resulting u is the optimal stationary policy by Theorem 4 for a finite control set A. The last V generated by the process is the total expected cost vector for the optimal policy u. The policy iteration procedure can be started at either step 1 or step 2. If there is no convenient policy to assume for initiating the process, that is, if there is no policy suspected to be near the optimum, then it is attractive to let $\gamma = 0$ initially. This results in the first policy iteration improving upon the stage cost—a reasonable procedure if no additional knowledge is available about the optimum.

2.6 Direct dynamic programming

An alternative to considering the infinite duration process with a stationary control just solved by policy iteration is to examine a finite duration process. An optimal control sequence which minimizes the expected cost over n time periods is sought. The conventional dynamic programming functional equation results, and taking the limit as $n \to \infty$ the same control is obtained as by policy iteration. Consider,

$$v_n(i) \triangleq \min_{\{u_0, u_1, \ldots, u_n\}} \left\{ \varphi_i + E\left\{ \sum_{k=0}^{n} \beta^k \ell(\underset{\sim}{x}_k, u_k(\underset{\sim}{x}_k)) \,\middle|\, \underset{\sim}{x}_0 = {}'x \right\}\right.$$

$$= \min_{u_0} \left\{ \varphi_i + \ell_i(u_0) + \beta \sum_{j_1} p_{ij_1}(u_0) \min_{u_1} \left\{ \ell_{j_1}(u_1) + \cdots \right.\right.$$

$$\left.\left. \cdots + \min_{u_{n-1}} \left\{ \ell_{j_{n-1}}(u_{n-1}) + \beta \sum_{j_n} p_{j_{n-1} j_n}(u_{n-1}) \min_{u_n} \{ \ell_{j_n}(u_n) \} \cdots \right\}\right.\right.$$

$$v_n(i) = \min_{u_0} \left\{ \ell_i(u_0) + \beta \sum_j p_{ij}(u_0) v_{n-1}(j) \right\}, \qquad (2.11)$$

where $v_{-1}(i) = \varphi_i$ is an arbitrary terminal cost, $i = 1, \ldots, J$.

As before, the set of cost functions

$\{ v_0(i), v_1(i), \ldots, v_n(i) \}$ is bounded for all

$i = 1, \dots, J$ since

$$0 < v_n(i) \leq M\,(1+\beta+\cdots+\beta^n) < \frac{M}{1-\beta},$$

where $M = \max\limits_{i,\alpha\in A} \ell({}^i x, \alpha)$

Let $v_1({}^i x) = M/1-\beta$ for $i = 1, \dots, J$; for this terminal cost $v_n(i)$ decreases monotonically. To show this, observe that

$$v_0(i) \leq v_{n-1}(i) \quad \text{for all } i$$

Now assume

$$v_n(i) \leq v_{n-1}(i) \quad \text{for all}$$

and show

$$v_{n+1}(i) \leq v_n(i) \quad \text{for all } i$$

$$\begin{aligned} v_n(i) &= \min_u \Big\{ \ell_i(u) + \beta \sum_j p_{ij}(u)\, v_{n-1}(j) \Big\} \\ &= \ell_i(u^*) + \beta \sum_j p_{ij}(u^*)\, v_{n-1}(j) \quad \text{, say.} \\ v_{n+1}(i) &= \min_u \Big\{ \ell_i(u) + \beta \sum_j p_{ij}(u)\, v_n(j) \Big\} \\ &\leq \ell_i(u^*) + \beta \sum_j p_{ij}(u^*)\, v_n(j) \\ &\leq \ell_i(u^*) + \beta \sum_j p_{ij}(u^*)\, v_{n-1}(j) \end{aligned}$$

Thus $v_{n+1}(i) \leq v_n(i)$, and $v_n(i)$ is seen to decrease monotonically. Again, since the sequence $\{ v_0(i), v_1(i), \dots \}$ is monotonically decreasing and bounded below by zero, it has a limit as $n \to \infty$, say, $v(i)$. Taking this limit in (2.11)

$$v(i) = \min_{u}\left\{ l_i(u) + \beta \sum_{j} p_{ij}(u)\, v(j) \right\}. \qquad (2.12)$$

Thus, (2.12) defines the expected cost function for an optimal policy over an infinite duration. Furthermore, it can be established that the solution to the equation is unique. Assume to the contrary that two solutions, $v(i)$ and $y(i)$ exist with associated control functions u and S. Then

$$v(i) = l_i(u) + \beta \sum_{j} p_{ij}(u)\, v(j),$$

$$y(i) \leq l_i(u) + \beta \sum_{j} p_{ij}(u)\, y(j).$$

Subtracting yields,

$$[y(i) - v(i)] \leq \beta \sum_{j} p_{ij}(u)\, [y(j) - v(j)].$$

By successive substitution,

$$[y(i) - v(i)] \leq \beta^{n} \sum_{j} p_{ij}(n,u)\, [y(j) - v(j)].$$

Taking the limit as $n \to \infty$,

$y(i) - v(i) \leq 0$ for all i.

By a symmetrical argument,

$y(i) - v(i) \geq 0$ for all i.

Thus,

$$y(i) = v(i),$$

and the solution $V = (v(1), ---, v(J))^{T}$ to (2.12) is seen to be unique. Also, by letting $n \to \infty$ the control which results from dynamic programming is optimal for the original cost function (2.3). Since

$$v(i) = \min_u \left\{ l_i(u) + \beta \sum_j p_{ij}(u)\, v(j) \right\}$$

$$= l_i(u^*) + \beta \sum_j p_{ij}(u)\, v(j)\text{, say.}$$

There exists no $g \epsilon V$ such that

$$l_i(g) + \beta \sum_j p_{ij}(g) v(j) \lneq l_i(u^*) + \beta \sum_j p_{ij}(u^*) v(j).$$

It is seen that the solution to (2.12) is the same as the solution of Howard's policy iteration procedure. Thus, the solution of the dynamic programming iterative equation

$$V_n = \min_u \left\{ L_1(u) + \beta P(u) V_{n-1} \right\}$$

as $n \to \infty$ yields the same cost function as does policy iteration. It is also apparent that, if the limiting control function resulting from dynamic programming is used as a stationary policy, then this policy is the same as the one resulting from policy iteration.

One important question still remains unanswered. What is the rate of convergence of the dynamic programming solution to the stationary optimum? As before, the sequence

$$v_0(i),\ v_1(i),\ v_2(i)\ \ldots.$$

decreases monotonically to $v(i)$.

$$v(i) = \min_{u} \left\{ l_i(u) + \beta \sum_j p_{ij}(u) v(j) \right\}$$
$$= l_i(u^*) + \beta \sum_j p_{ij}(u^*) v(j) .$$
$$v_n(i) = \min_{u} \left\{ l_i(u) + \beta \sum_j p_{ij}(u) v_{n-1}(j) \right\}$$
$$\leq l_i(u^*) + \beta \sum_j p_{ij}(u^*) v_{n-1}(j)$$

Thus,

$$v_n(i) - v(i) \leq \beta \sum_j p_{ij}(u^*) [v_{n-1}(j) - v(i)]$$

Let,

$$\epsilon_n \triangleq \operatorname{Max}_i \left\{ v_n(i) - v(i) \right\} > 0 ,$$

then,

$$\epsilon_n \leq \beta \, \epsilon_{n-1} \quad . \tag{2.13}$$

The maximum deviation of V_n from V thus decreases at a rate of at least β. Practical experience shows that this estimate of the rate of improvement is quite close. It is seen that for β less than about .7 the rate of convergence is very rapid.

The maximum error, ϵ_n, is, of course, impossible to obtain during the dynamic programming algorithm since the final cost V is unknown. A bound on ϵ_n can however be found.

Let

$$\delta_n = \max_i \left\{ v_{n-1}(i) - v_n(i) \right\} > 0$$

As before, for

$$v_n(i) = l_i(u^*) + \beta \sum_j p_{ij}(u^*) v_{n-1}(j) \; ;$$
$$v_{n-1}(i) \leq l_i(u^*) + \beta \sum_j p_{ij}(u^*) v_{n-2}(j) .$$

Thus,

$$v_{n-1}(i) - v_n(i) \leq \beta \sum_j p_{ij}(u^*) \left[v_{n-1}(j) - v_{n-2}(j) \right],$$

and

$$\delta_n \leq \beta \delta_{n-1}.$$

However,

$$\begin{aligned} \epsilon_n &= \delta_{n+1} + \delta_{n+2} + \cdots \\ &\leq \delta_n \left[\beta + \beta^2 + \cdots \right] \end{aligned}$$

or

$$\epsilon_n \leq \delta_n \frac{\beta}{1-\beta}$$

Thus the error ϵ_n is bounded by the observable stage difference, δ_n. The dynamic programming algorithm can be terminated when δ_n gets sufficiently small.

2.7 Howard's policy iteration for $\beta = 1$

The control of finite Markov chains with $\beta = 1$ (i.e., no discounting) is somewhat more difficult to examine than the discounted, cost chains. It is convenient to assume not only a finite set of stationary control laws, but also to restrict A such that for any $u \in U$ the resulting Markov chain is ergodic. Before defining what optimal control means for the undiscounted costs, the behavior of the cost function is examined.

Let,

$$v_n(i,u) = E\left\{ \sum_{k=0}^{n} \ell(\underset{\sim}{x}_k, u(\underset{\sim}{x}_k)) \middle| \underset{\sim}{x}_0 = {}^i x \right\} \qquad (2.15)$$

be the undiscounted expected cost function for the stationary policy, u, applied to n stages. Then, as before,

$$v_n(i,u) = l_i(u) + \sum_j p_{ij}(u)\, v_{n-1}(j) \tag{2.16}$$

with $v_{-1}(i) = 0$,

or
$$v_n(i,u) = l_i(u) + \sum_j p_{ij}(u)\, l_j(u) + \cdots + \sum_j p_{ij}(n,u)\, l_j(u)$$

In matrix form

$$\begin{aligned} V_n(u) &= L(u) + P(u)V_{n-1}(u) \\ &= L(u) + P(u)L(u) + \cdots + P^n(u)L(u) \end{aligned} \tag{2.17}$$

By Theorem 1,

$$P^n(u) = 1\mu + Q_n(u)$$

where $Q_n(u) \to 0$ as $n \to \infty$, geometrically fast.

Consider,

$$\lim_{n\to\infty} n^{-1} V_n(u) = \lim_{n\to\infty} n^{-1} \sum_{m=0}^{n} P^m(u)L(u)$$, if the limit exists,

$$= \lim_{n\to\infty} n^{-1} \sum_{m=0}^{n} 1\mu L(\mu) + \lim_{n\to\infty} n^{-1} \sum_{m=0}^{n} Q_m(u)L(u)$$

However,

$$\lim_{n\to\infty} n^{-1} \sum Q_m(u)L(u) = 0 \qquad \text{since} \qquad \lim_{n\to\infty} Q_n(u) = 0 .$$

Thus,

$$\lim n^{-1} V_n(u) = [\mu L(u)]\, 1$$

and for large n,

$$V_n(n) \simeq n[\mu L(u)]1 + \text{constant}$$

$$V_n(u) \simeq n g(u) 1 + W_n(u) \text{ , say} \tag{2.18}$$

The scalar $g(u) = \mu L(\mu)$ is the stationary average cost of the policy u, and the vector $W_n(u)$ is called the potential of the policy. Substituting (2.18) into (2.17):

$$n g(u) \mathbf{1} + W_n(u) \cong L(u) + P(u)\left[(n-1) g(u) \mathbf{1} + W_{n-1}(u)\right]$$

or

$$W_n(u) + g(u) \mathbf{1} \cong L(u) + P(u) W_{n-1}(u) \qquad n \text{ . large.}$$

with $W_{-1}(u) = 0$

In the limit as $n \to \infty$,

$$W(u) + g(u) \mathbf{1} = L(u) + P(u) W(u) \tag{2.21}$$

A stationary policy $u^* \in U$ is said to be optimal if

$$g(u^*) \leqslant g(u) \quad \text{for all } u \in U .$$

That is, the optimal policy for $\beta = 1$ is the one which accrues the least average cost.

The question arises, does (2.21) determine $g(u)$ and $W(u)$ uniquely? To answer this, consider two solutions, W, g and Y, a for the same policy u. (2.21) immediately yields,

$$W - Y + (g-a) \mathbf{1} = P(u)[W - Y]$$

or

$$Z = P(u) Z + C$$

where

$$C = (a-g) \mathbf{1} \; , \; Z = W - Y$$

Thus,

$$Z = nC + P(u) Z$$

$$\longrightarrow nC + \mathbf{1}\mu Z \quad \text{as } n \longrightarrow \infty .$$

However, the elements of Z are bounded as $n \to \infty$, thus $C \equiv 0$ and $g = a$. Therefore, the stationary average cost is determined uniquely by (2.21).

Now, with $C \equiv 0$, in the limit as $n \to \infty$,

$$Z = \mathbf{1} \mu Z \quad , \quad \sum_{j=1}^{J} \mu_j = 1 \quad \text{with } \mu_j > 0$$

The only solution to this equation is

$Z_i =$ constant.

Therefore, the potential, $W(u)$, for a given policy, $u \epsilon \mathcal{U}$, is determined up to an additive constant.

Howard's policy iteration for undiscounted cost may now be specified as follows: .

(1) for a given stationary policy, $u \in \mathcal{U}$, determine $g(u)$ and from $W(u)$ from

$$W(u) + g(u) = L(u) + P(u) W(u)$$

and go to step (2) with $W = W(u)$;

(2) for the potential function, W, select u such that $u({}^i x)$ minimizes

$$\ell({}^i x, u({}^i x)) + \sum_{j} p_{ij}(u) W_j \qquad i = 1, 2, \cdots, J$$

and repeat step (1).

Again, the process is terminated when there is no further improvement in W, or equivalently when the policy u ceases to change in step 2. To show that the policy iteration indeed yields an optimum stationary policy, consider any policy $u \in \mathcal{U}$, then

$$w(i,u) + g(u) = \ell_i(u) + \sum_{j} p_{ij}(u) w(j,u) \qquad (2.21)$$

A new policy, $\hat{u} \in \mathcal{U}$ is generated by minimizing the right hand side of 2.21. It is apparent that the additive constant in W does not affect $\hat{u}$. Now,

where

$$W(i,u) + g(u) \geqslant l_i(\hat{u}) + \sum_j p_{ij}(\hat{u})\, W(j,u),$$

and $>$ applies for some i.

Thus,

$$W(i,u) - W(i,\hat{u}) + g(u) - g(\hat{u}) \geqslant \sum_j p_{ij}(\hat{u})[W(j,u) - W(j,\hat{u})] \quad (2.22)$$

Recalling that for the stationary probability distribution, $\hat{\mu}$, associated with $\hat{u}$,

$$\sum_{k=1}^{J} \hat{\mu}_k p_{ij}(\hat{u}) = \hat{\mu}_j$$

and multiplying, (2.22) by μ_i and summing yields,

$$\sum_i \hat{\mu}_i [W(i,u) - W(i,\hat{u})] + g(u) - g(\hat{u}) > \sum_j \hat{\mu}_j [W(j,u) - W(j,\hat{u})]$$

or $\quad g(u) - g(\hat{u}) > 0$

Therefore, $g(\hat{u}) < g(u)$ and the policy, $\hat{u}$, generated by policy iteration is superior to u, the policy which preceded it. Since there are only a finite number of policies eventually there occurs a policy which can not be improved upon in step (2). This policy is the optimal policy.

2.8 The optimal control as $\beta \to 1$

It is interesting to consider whether the control obtained for $\beta < 1$ but sufficiently close to one is the same as the control for $\beta = 1$. Let $\beta < 1$ be a fixed policy arbitrarily close to one with an associated optimal policy u^*. Then call the second best policy the one with minimum maximum deviation from $V\cdot(u^*,\beta)$, where the notation includes the dependence on β . Thus the second best policy is such that

$$\max_i \left\{ v(i,u,\beta) - v(i,u^*;\beta) \right\}$$

is minimized. Now, since for any fixed u ., $V(u,\beta)$ is continuous in β it follows that there exists a β_0 such that for all β , $\beta_0<\beta<1$ the cost of the policy u^* is less than that of the second best policy. Holding this policy fixed, a potential type function is defined for $\beta_0<\beta<1$ so that the optimal policy as $\beta\to 1$ may be examined. Let,

$$W_n(u^*,\beta) \triangleq V_n(u^*,\beta) - \sum_{k=0}^{n} \beta^k g(u^*)\,\mathbf{1} \qquad (2.24)$$

where $g(u^*)$ is the average stationary cost of u^* for $\beta=1$, and

$$V_n(u^*,\beta) \triangleq \sum_{k=0}^{n} \beta^k P^k(u^*) L(u^*) .$$

Since $\sum_{k=0}^{n} \beta^k g(u^*)\mathbf{1}$ is constant with respect to i, finding the control u such that $u(^ix)$ minimizes $\left\{ l_i(u) + \sum_j p_{ij}(u) W_n(j;u^*,\beta) \right\}$ is the same as the control which minimizes $\left\{ l_i(u) + \sum_j p_{ij}(u) V_n(j;u^*,\beta) \right\}$ Now examine the potential function as $\beta\to 1$,

$$W_n(u^*,\beta) = \sum_{k=0}^{n} \beta^k \left[\mathbf{1}\mu(u^*) + Q_k(u^*) \right] L(u^*) - g(u^*) \frac{1-\beta^{n+1}}{1-\beta}$$

$$= \sum_{k=0}^{n} \beta^k g(u^*) + \sum_{k=0}^{n} \beta^k Q_k(u^*) L(u^*) - g(u^*) \frac{1-\beta^{n+1}}{1-\beta}$$

$$= \sum_{k=0}^{n} \beta^k Q_k(u^*) L(u^*) .$$

In the last section

$$\lim_{n\to\infty} \sum_{k=0}^{n} Q_k(u^*) \cdot L(u^*) = W(u^*)$$

existed for undiscounted cost; thus,

$$\lim_{n\to\infty} \lim_{\beta\to1} W_n(u^*, \beta) = W(u^*)$$

and further it is apparent that,

$$\lim_{\beta\to1} \lim_{n\to\infty} W_n(u^*, \beta) = W(u^*)$$

Also,

$$\lim_{n\to\infty} V_n(u^*, \beta) = V(u^*, \beta)$$

Since the control policy obtained by applying step two of Howard's policy iteration to either $V_n(u^*, \beta)$ or $W_n(u^*, \beta)$ is the same, it follows that the optimum policy u^* is valid for $\beta_0 < \beta \leq 1$. Thus the undiscounted problem can be solved by solving the discounted problem for β sufficiently close to one.

CHAPTER III

A NUMERICAL ALGORITHM FOR OPTIMAL CONTROL

3.1 Introduction

In Chapter II the characteristics of the expected cost function were examined, and two methods, Howard's policy iteration and dynamic programming, were developed for obtaining the optimal control of finite Markov chains. In this chapter stochastic systems whose state space is defined on the continuum are considered. However, rather than view these systems rigorously as infinite state diffusion processes, they will be considered as finite Markov chains with the large but finite discrete state space. A numerical algorithm which employs a quadratic approximation to the expected cost function for a partitioned state space will be developed.

3.2 System description

The systems to be studied are defined by a set of difference equations

$$x(k+1) = f(x(k), \alpha(k)) + \xi(k) \qquad (3.1)$$

called the plant equation, where

k = time parameter

x = n-dimensional state vector

α = q-dimensional control vector

ξ = n-dimensional random vector, plant noise

f = n-dimensional vector function.

The state $x=(x_1, x_2, \cdots, x_n)^T$ is restricted to the state space

$\mathcal{X} = \{x \mid xmin_i \leq x_i \leq xmax_i, i=1,\cdots,n\}$ and any transition out of this region under (2.1) is not considered. The control $\alpha = (\alpha_1, \ldots, \alpha_q)^T$ is restricted to the control space A. The random variable ξ, called the plant noise, has a known probability density function, $p_\xi(\xi)$, which is time invariant, and ξ is independent from one time instant to another. If it is desired to model a system with correlation between plant noise from one time instant to the next, it is possible to define additional state variables and new random variables for which the plant noise is independent (Meier 1965). Also, with no loss of generality is considered to have zero mean.

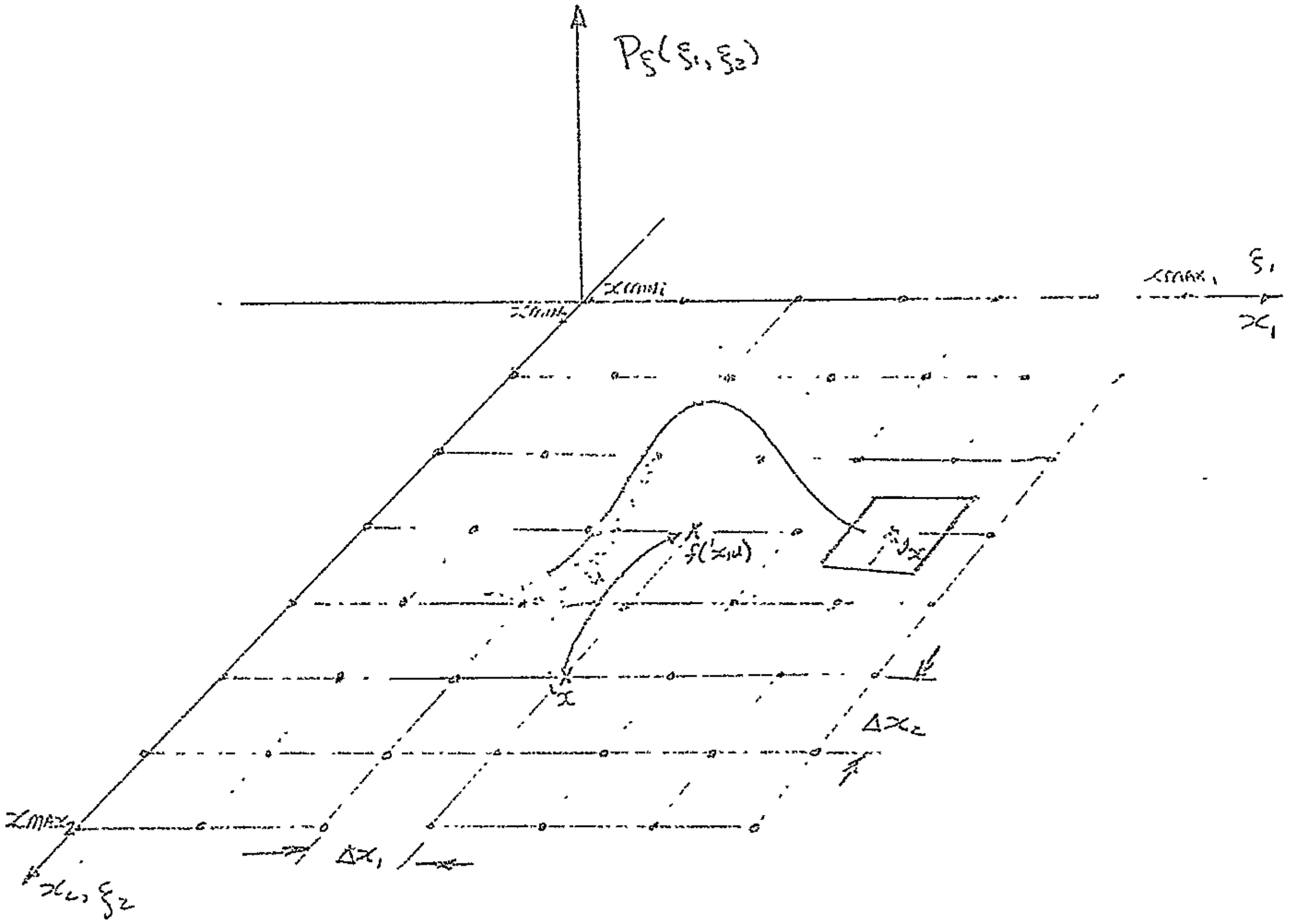
Stochastic constrol systems with continuous state space can be considered, as an approximation, to be finite Markov chains by establishing a grid on the state space $\mathcal{X}$. The grid points are states of the finite Markov chain and the transition probabilities, p_{ij}, defining the chain under a stationary control law, are obtained by determining the probability of a state transition from ${}^ix$ on the grid to a hypercube about ${}^jx$ on the grid. To better illustrate this, consider the second order system in Figure 2-1. The transition probability $p_{ij}(\alpha)$ under control α is defined as

$$p_{ij}(u) = \int_{{}^jx_1-\frac{1}{2}\Delta x_1 - f_1({}^ix,\alpha)}^{{}^jx_1+\frac{1}{2}\Delta x_1 - f_1({}^ix,\alpha)} \int_{{}^jx_2-\frac{1}{2}\Delta x_2 - f_2({}^ix,\alpha)}^{{}^jx_2+\frac{1}{2}\Delta x_2 - f_2({}^ix,\alpha)} p_\xi(\xi_1, \xi_2)\, d\xi_1 d\xi_2 \qquad (3.2)$$

The stage cost at time k is defined, as before,

$$0 < \ell(x(k), \alpha(k)) < M$$

The total expected cost function is, as in Chapter II, for a stationary control law, u,



'.;gure 3.1 Grid for discrete state approximation to a continuous stace system.

$$v(x) = E\left\{ \sum \beta^k l\left(\underset{\sim}{x}(k), u(\underset{\sim}{x}(k))\right) \middle| \underset{\sim}{x}(0) = x_0 \right\}$$

Again the control law u with $u(x(k)) \in A$ is sought which minimizes $v(x)$ for all $x \in X$, or, for the finite Markov chain representation, all x which are grid points.

As before,

$$v(x) = l(x_0, u(x_0)) + \beta E\left\{v\left(f(x_0, u(x_0)) + \xi_0\right)\right\} \quad (3.3)$$

3.3 Solution by Howard's policy iteration

To find the optimal control via policy iteration it is first necessary to model the system as a finite Markov chain. A grid must be established which is sufficiently fine to approximate the behavior of the system defined on the continuum. Dividing each coordinate x_i into N_i equal increments Δx_i wide accomplishes this for Δx_i small enough,

$$N_i = \frac{x_{max\,i} - x_{min\,i}}{\Delta x_i}, \quad i = 1, \cdots, n,$$

and defines $J = \prod_{i=1}^{n} N_i$ grid points.

Now to obtain transition probabilities $p_{ij}(u)$ under the stationary control u it will be necessary to perform the integration in (3.2) times. Then having attained the $J \times J$ transition matrix

$$P(u) = [p_{ij}(u)],$$

step one of the policy iteration procedure (Section 2.5) requires inverting

$$[I - \beta P]$$

also, a $J \times J$ matrix. In the minimization in step two, it will again be necessary to evaluate (3.2) J^2 times for each control law considered. The number of control laws considered will depend on the numerical minimization technique used, but it is evident that this number could be large even for limited control spaces. To see the prodigious labor necessary to employ Howard's policy iteration for systems with continuous state space, consider a second order example with

$$X = \{ 1 \le x_1 \le 100 \; , \; 1 \le x_2 \le 100 \}$$

and let $\Delta x_1 = \Delta x_2 = 1$. Then $N_1 = N_2 = 100$, and $J = 10^4$. Thus P has 10^8 elements as does $[I - \beta P]$. Already it is evident that while Howard's policy iteration is a valuable technique for finding the optimal control of finite Markov chains with very few states and a useful theoretical tool, it is impractical to employ it on the systems defined in this chapter. It would be necessary in the present example to store 100 million transition probabilities in computer storage and invert a $10^4 \times 10^4$ matrix to achieve only step one of the first iteration of Howard's method—clearly an overwhelming computational task. On the other hand, it will be shown in the next section that dynamic programming as developed in Section 2.6 offers a more palatable numerical solution.

3.4 Solution by dynamic programming

To employ dynamic programming, as before, a N-stage minimum expected cost function is defined,

$$v(x_0, N) = \min_{\{u_0, u_1, \cdots u_N\}} E\left\{ \sum_{k=0}^{N} \beta^k \ell(\underset{\sim}{x}(k), u_k(x(k))) \;\middle|\; \underset{\sim}{x}(0) = x_0 \right\}$$

or

$$v(x_0, N) = \min_{u_0} \left\{ \ell(x_0, u_0(x_0)) + \beta E\left\{ v(f(x_0, u_0(x_0)) + \xi_0, N-1) \;\middle|\; x_0 \right\} \right\} \qquad (3.3b)$$

with $v(x,0)=0$. Again a grid is imposed on the state space with N_i increments along the x_i axis and $J=\prod_{i=1}^{n} N_i$ total grid points. It would now be possible to employ (3.2) to define the $J\times J$ transition matrix P and (3.3) would become, as in the last chapter,

$$v({}^{i}x,N)=\min_{u}\left\{ \ell({}^{i}x,u({}^{i}x))+\beta\sum_{j=1}^{J}P_{ij}(u)\,v({}^{j}x,N-1)\right\}$$

for all the grid points. However, to avoid the difficulty of obtaining P, a more convenient approximation is to quantify the noise in a manner similar to imposing a grid on the state space. That is, the probability density function $p_\xi(\xi)$ is approximated by imposing a grid on the domain of p_ξ and attaching a probability to each grid point. Then the noise is described by the set of noise values $\{{}^{i}\xi,\ i=1,\dots,N_\xi\}$ and the associated probabilities, $\{p({}^{i}\xi),\ i=1,\dots,N_\xi\}$. Now equation (3.3b) becomes,

$$v({}^{i}x,N)=\min_{u}\left\{ \ell({}^{i}x,u({}^{i}x))+\beta\sum_{j=1}^{N_\xi}p({}^{j}\xi)\,v\left(f({}^{i}x,u)+{}^{j}\xi,N-1\right)\right\} \quad (3.4)$$

$v({}^{i}x,0)=0$ for $i=1,2,\cdots,J$. Equations (3.1) and (3.4) describe the dynamic programming numerical algorithm for the solution of the stochastic control problem with discounted cost. While the dynamic programming functional equation (3.4) offers a solution to a wide range of problems analytically, the computational requirements of high-speed computer memory and computing time can become excessive except for simple problems. The memory requirements are the same as for deterministic problems while the computation time is more severe. To better observe these difficulties and to see that Bellman's "curse of dimensionality" not only affects memory requirements but also computing time in the stochastic control problem a more detailed examination of the algorithm is in order.

Since it was shown in the previous chapter that $\lim_{n \to \infty} v({}^{i}x, N) = v({}^{i}x)$ there is no necessity to store all the cost functions and control functions generated as (3.4) is solved. Only the last cost function and the present cost function,and control function that is being generated, need be stored. Thus, $3 \cdot J = 3 \prod_{i=1}^{n} N_i$ memory locations are required to store the information vital to the iteration of (3.4). Further, for economy in computation time, these values should be stored in high-speed memory (Larson, 1968) which for most computers is limited to about 10^5 words. Thus for the second order example of Section 3.3 it would be necessary to have available $3 \cdot 10^4$ high-speed memory locations. For a three dimensional state space with $N_i = 100, i = 1,2,3$, $3 \cdot 10^6$ storage locations would be necessary, overwhelming the capacity of nearly any computer. This "curse of dimensionality" is a severe limitation to the problems solvable by dynamic programming. A first order problem is shown in Figure 3.2. To evaluate $v({}^{i}x, k)$ with the control $u_k({}^{i}x)$ applied, it is necessary to evaluate $v(x_{k+1}, k+1)$ by interpolation of the stored cost function at time $k+1$ N_ξ times where N_ξ is the number of discrete noise levels used to approximate the probability density function $p_\xi(\xi)$.

For a second order plant with

$$\begin{aligned} x_1(k+1) &= f_1(x(k), u) + \xi_1(k) \\ x_2(k+1) &= f_2(x(k), u) + \xi_2(k) , \end{aligned}$$

and ξ_1 independent of ξ_2 then both ξ_1 and ξ_2 could be quantified separately into say M_1 and M_2 levels. Thus, $N_\xi = M_1 M_2$ and in general

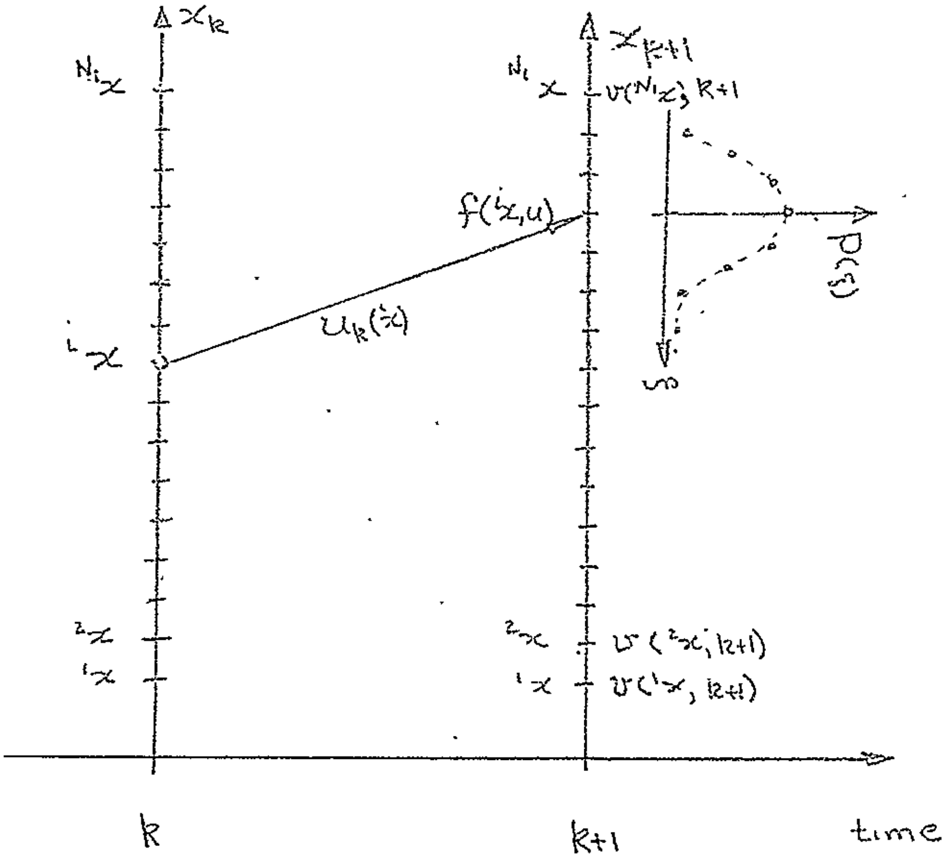


Figure 3.2a The dynamic programming numerical algorithm for first order problem.

for a n^{th} order plant $N_\xi = \prod_{i=1}^{n} M_i$, and the cost function must be evaluated $J\prod_{i=1}^{n} M_i$ times for each iteration of (3.4). Consider each noise element quantified into, say, five levels. The number of cost function evaluations necessary for the stochastic problem as opposed to the deterministic problem $(\text{Prob}[\xi=0]=1)$ increases by a factor of five for each increase in dimensionality. Thus the "curse of dimensionality" affects the computation time of the stochastic problem with respect to the quantization of the noise. It is the main purpose of this dissertation to develop an algorithm which alleviates the high-speed memory requirement and long computational time intrinsic to a straightforward application of dynamic programming to the stochastic control problem. The next section begins the development of this algorithm.

3.5 Dynamic programming with a partitioned state space

The problem of excessive high-speed storage which is attendant to the dynamic programming algorithm was attacked with considerable success by Larson (1964, 1968) for the case of a deterministic plant and continuous time, i.e.,

$$\dot{x}(t) = f(x(t), u(t), t) .$$

Larson's method, called state increment dynamic programming, took specific advantage of time being defined on the continuum. This restriction and the deterministic nature of his plant equation thwart a direct application of his technique to the discrete time stochastic problem under study. However, a basic concept of Larson's method will be employed for the problem at hand. State space will be partitioned into blocks, and these blocks will be treated individually in calculating the optimal control and cost function. The expected cost function, $U(x)$, over each of these blocks will be approximated by a quadratic surface. The effect of this partition and the quadratic surfaces is to substantially reduce the amount of high-speed memory necessary and also to reduce the computation time. The price paid for these advantages is a more approximate control law than that achieved by conventional dynamic programming. However, the classes of systems examined will be restricted such that this loss of accuracy is not substantial.

To better illustrate these concepts, consider the second order problem and two dimensional state space in Figure 3.2. Here the state space has been partitioned into 25 blocks of equal dimension. There is no advantage in unequal dimensions, so for simplicity equal dimension blocks are used for the partition. The expected cost function is also partitioned into the surfaces above each block. In the figure the furface partitions above blocks 0 and 5 are illustrated. These surfaces are then to be approximated by a quadratic fit which in the two dimensional case will be, for block ℓ,

$$U(x) = \alpha(\ell) + \beta_1(\ell)x_1 + \beta_2(\ell)x_2 + \gamma_{11}(\ell)x_1^2 + \gamma_{12}(\ell)x_1x_2 + \gamma_{22}(\ell)x_2^2$$

and for the n^{th} order system,

$$U(x) \doteq \alpha(\ell) + \sum_{i=1}^{n} \beta_i(\ell)x_i + \sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\ell)x_1x_2 \qquad (3.5)$$

The block size is selected such that, as illustrated in Figure 3.3, when ${}^{i}x$ is under consideration and control u is applied

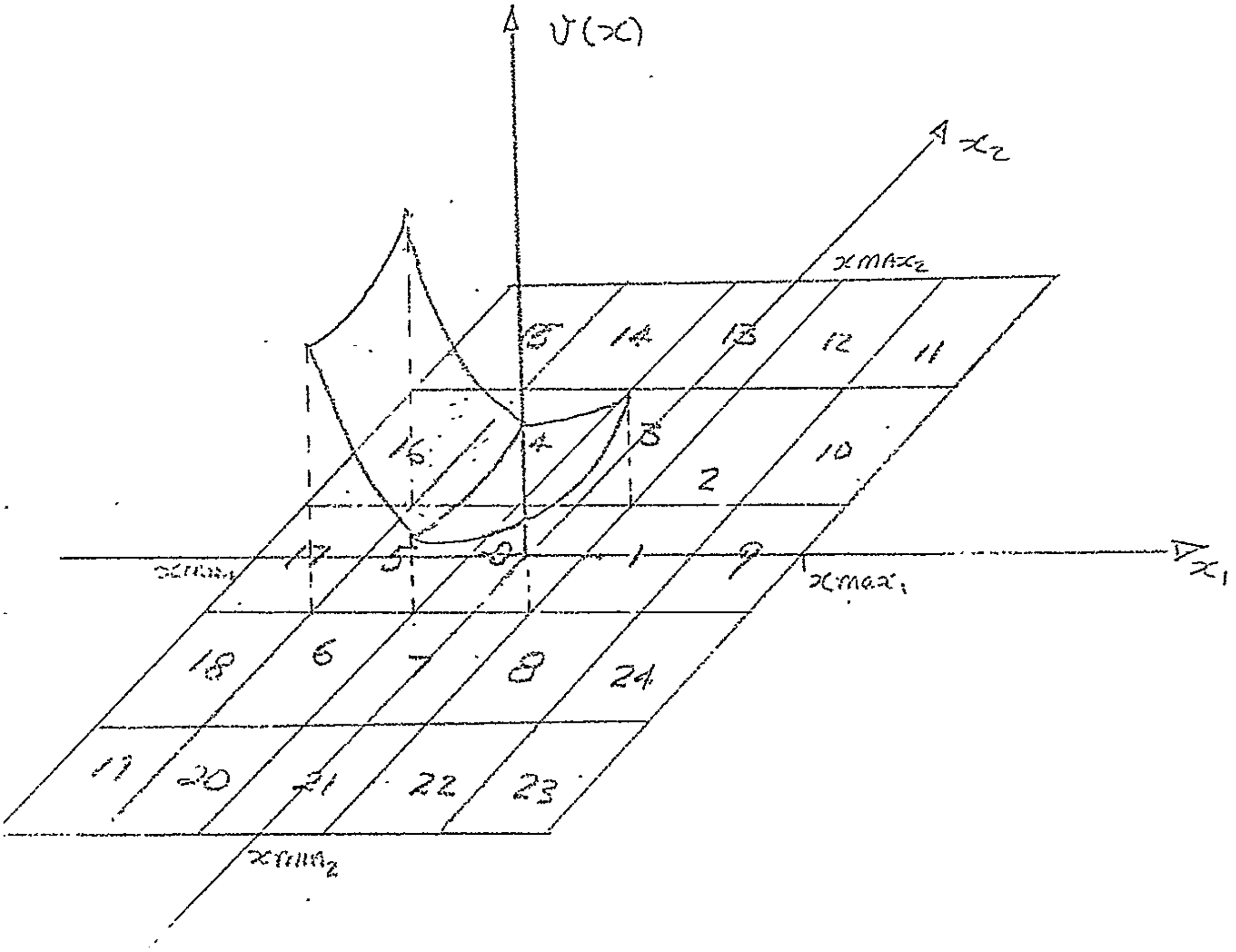


Figure 3.2 b Partitioned two dimensional state space.

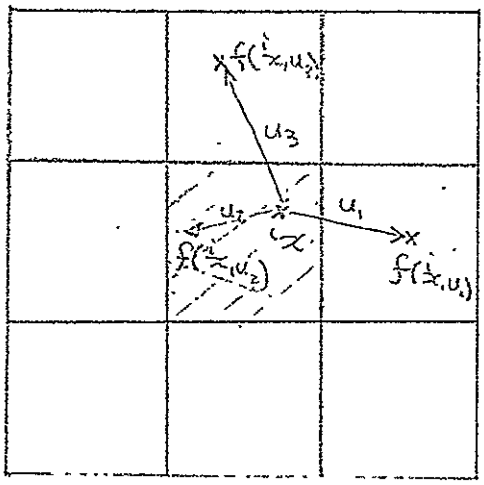


Figure 3.3 Transitions from the state ${}^{i}x$

$f({}^{i}x,u)$ lies in the block containing ${}^{i}x$ or an adjacent block. This condition can be met easily enough by making the block size very large. However, since the cost function or surface over each block is to be approximated by a quadratic surface, it is also desirable to have the blocks small in size. Thus, a compromise must be reached, and this compromise obviously depends upon the problem being solved. A reflective examination of the system equations is usually adequate to determine an appropriate block size.

Consider for example that the state space in Figure 3.2 is $x\min_1 = x\min_2 = -25$ and $x\max_1 = x\max_2 = 25$ and that $\Delta x_1 = \Delta x_2 = 1$. Thus, each block would have 100 points in it (including its boundaries) with 10 increments to a side. The cost surface above each block would be described by 6 numbers, α, β's, and γ's. Since for each x a member of block ℓ $(x \in B_\ell)$ $f(x,u)$ is restrained to be a member of either B_ℓ or a block adjacent, it is possible to evaluate (3.3) for all points in B_ℓ with only the parametric description of B_ℓ and its adjacent blocks in high-speed memory. Thus, recalling Figure 3.3, only $9 \cdot 6 = 36$ high-speed memory locations are necessary to store the cost surface for the partitioned state space algorithm. For conventional dynamic programming $50^2 = 2500$ high-speed memory locations would be necessary.

Obviously, even for conventional dynamic programming it would be possible to store the entire cost function in low-speed memory (tape, disc, or drum storage). However, then it would be necessary to go to low-speed memory for each cost function evaluation. This is a time-consuming process which would involve $K \cdot \prod_{i=1}^{n} N_i \prod_{i=1}^{n} M_i$ accesses to low-speed storage where K is the number of controls evaluated at each state point. With $K=10$ the example considered in Section 3.3 would require $10 \cdot 100 \cdot 25 = 25{,}000$ accesses to low-speed memory. For the partitioned state space (PSS) algorithm only accesses would have to be made to low-speed memory, where N_B is the number of blocks (25 accesses for the problem in Figure 3.2). In the next section the PSS algorithm is shown to reduce computation time as well as high-speed storage.

3.6 The quadratic approximation of the cost surface

The criteria for fitting the quadratic surface to the cost function over a given block is taken to be unweighted least squares regression. For block B_ℓ recall,

$$\bar{V}_\ell(x) = \alpha(\ell) + \sum_{i=1}^{n} \beta_i(\ell) x_i + \sum_{i=1}^{n} \sum_{j=1}^{i} \hat{\gamma}_{ij}(\ell) x_i x_j \tag{3.5}$$

and the functional to be minimized is,

$$J(\alpha, \beta_1, \ldots, \beta_n, \hat{\gamma}_{11}, \ldots, \hat{\gamma}_{nn}) = \sum_{x \in B_\ell} \left(V(x) - \bar{V}_\ell(x) \right)^2$$

for the $M = \frac{n^2+3n+2}{2}$ parameters of the quadratic surface. Thus

$$\frac{\partial J}{\partial \alpha} = \frac{\partial J}{\partial \beta_i} = \frac{\partial J}{\partial \hat{\gamma}_{ij}} = 0 \,,$$

which yields,

$$\sum_{x \in B_\ell} \left\{ \alpha + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij} x_i x_j \right\} = \sum_{x \in B_\ell} V(x) \,, \tag{3.6a}$$

$$\sum_{x \in B_\ell} \left\{ \alpha x_k + \sum_{i=1}^{n} \beta_i x_i x_k + \sum_{i=1}^{n} \sum_{j=1}^{i} \hat{\gamma}_{ij} x_i x_j x_k \right\} = \sum_{B_\ell} V(x) x_k \tag{3.6b}$$

and

$$k = 1, \ldots, n$$

$$\sum_{x \in B_\ell} \left\{ \alpha x_k x_m + \sum_{i=1}^{n} \beta x_i x_k x_m + \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij} x_i x_j x_k x_m \right\} = \sum \upsilon(x) x_k x_m \quad (3.6c)$$

$$k = 1, \ldots, n \; ; m = 1, \ldots, k$$

(3.6) may be summarized in matrix form as,

$$S\, Z(\ell) = T(\ell) \quad (3.7)$$

where $\quad Z(\ell) = (\alpha, \beta_1, \ldots, \beta_n, \gamma_{11}, \ldots, \gamma_{nn})^T$

and $\quad T(\ell) = \sum_{x \in B_\ell} (\upsilon(x), x_1 \upsilon(x), \ldots, x_n \upsilon(x), x_1^2 \upsilon(x), \ldots, x_n^2 \upsilon(x))^T$

are $1 \times M$ column vectors, and S is the $M \times M$ matrix described by (3.6) such that (3.7) holds. Thus, the column vector, Z, describing the quadratic surface is

$$Z(\ell) = S^{-1} T(\ell)$$

It is not necessary to invert a S matrix for each block; instead, since all blocks have the same dimensions, S, T may be calculated for a block with standard coordinates, and $\upsilon(x)$ transformed to this block. Thus the $M \times M$ matrix S need be inverted only once. Further, the storage for the surface for B_ℓ and adjacent blocks is $N_S = 3^n \cdot M$ locations. Thus,

$$n = 2 \implies N_S = 36$$
$$n = 3 \implies N_S = 270$$
$$n = 4 \implies N_S = 1215 \; ; \text{etc.}$$

To see that the quadratic approximation not only reduces high-speed storage requirements but also computation time, recall (3.3)

$$\upsilon(x, N) = \min_u \left\{ \ell(x, u) + \beta E\left\{ \upsilon(f(x, u) + \xi, N-1) \right\} \right\}$$

For the noise quantified into N_ξ values (3.3) becames (3.4),

$$v(x,N) = \min_u \left\{ \ell(x,u) + \beta \sum_{j=1}^{N_\xi} p({}^j\xi)\, v\left(f(x,u) + {}^j\xi, N-1\right) \right\}.$$

Therefore, it is necessary to evaluate $v(x,N-1)$ N_ξ times for each control considered, where N_ξ will have a tendency to increase geometrically with the dimension, n. On the other hand, for PSS dynamic programming with $f(x,u)$ lying in block ℓ and parameters $\bar{z}(\ell)$ describing $v(x,N-1)$. for $x \in B_\ell$,

$$v(x,N) = \min_u \left\{ \ell(x,u) + \beta E \left\{ v\left(f(x,u) + \xi, N-1\right) \right\} \right\},$$

or, approximately,

$$v(x,N) = \min_u \Big\{ \ell(x,u) + \beta E \Big\{ \alpha(\ell) + \sum_{i=1}^{n} \beta_i(\ell) \left(f_i(x,u) + \xi_i \right)$$
$$+ \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij}(\ell) \left(f_i(x,u) + \xi_i \right) \left(f_j(x,u) + \xi_j \right) \Big\} \Big\}$$

$$= \min_u \Big\{ \ell(x,u) + \beta \Big\{ \alpha(\ell) + \sum_{i=1}^{n} \beta_i(\ell) f_i(x,u) + \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij}(\ell) f_i(x,u) f_j(x,u)$$
$$+ \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij}(\ell) E\{\xi_i \xi_j\} \Big\}$$

$$v(x,N) = \min_u \left\{ \ell(x,u) + \beta \left\{ \bar{v}_\ell \left(f(x,u), N-1 \right) + \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij} E[\xi_i \xi_j] \right\} \right\} \qquad (3.8)$$

Thus, only one cost function evaluation must be made for each control and the additional term, $\sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij}(\ell) E[\xi_i \xi_j]$

calculated using known covariances, $E\{\xi_i \xi_j\}$. The cost function evaluation is of $\bar{V}(x,N-1)$ rather than $V(x,N-1)$; however, the computation time of the two evaluations is comparable. The quadratic approximation to the cost function, therefore, affords a significant savings in cost function evaluations and computation time.

3.7 PSS algorithm

Once the state space has been partitioned, the PSS dynamic programming algorithm can be applied. A flow diagram of the basic procedure is contained in Figure 3.4 while a more detailed flow diagram and Fortran program listing are to be found in Appendix A.

A particular block is designated as the origin block (for example, block 1 of Figure 3.2) and the cost surface associated with it is determined by techniques to be discussed in Section 3.9. The origin block is generally selected to contain the minimum of the cost function over all state space if possible. For many problems it is easy to define the origin block appropriately, such as the stochastic regulator problem where the system is to be driven to the origin of state space.

With the cost surface for the origin block obtained, another block, say B_1 , is considered for processing (Step 2). Both this block and all adjacent calculated blocks are brought into high-speed storage. The block being processed must have at least one calculated block next to it. This is not a significant restriction on the method, as, in general, the blocks are ordered in such a manner that they radiate out from the origin block as they are considered (Figure 3.2).

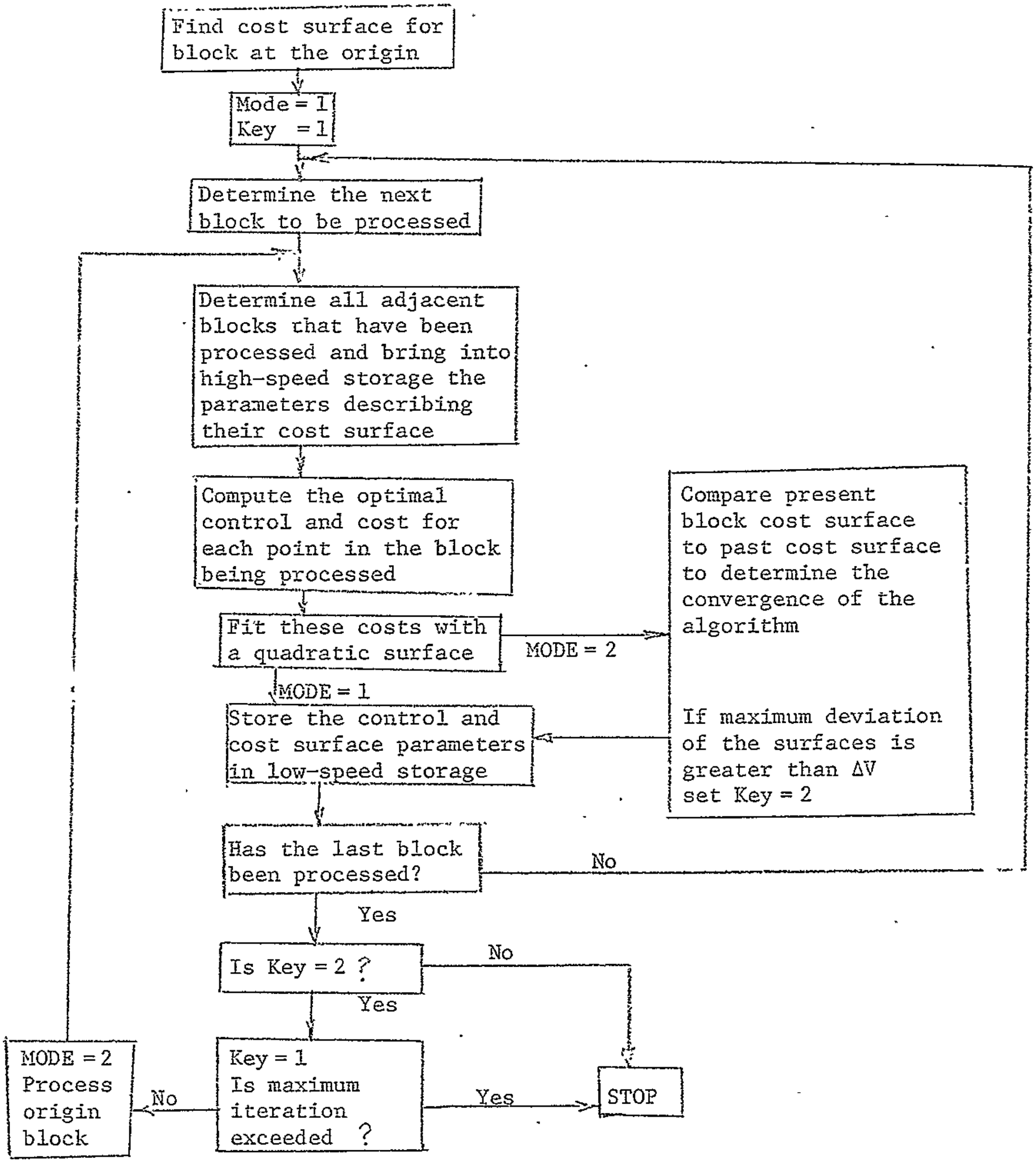

Find cost surface for block at the origin
Mode = 1
Key = 1
Determine the next block to be processed
Determine all adjacent blocks that have been processed and bring into high-speed storage the parameters describing their cost surface
Compute the optimal control and cost for each point in the block being processed
Fit these costs with a quadratic surface
MODE = 2
Compare present block cost surface to past cost surface to determine the convergence of the algorithm
If maximum deviation of the surfaces is greater than ΔV set Key = 2
MODE = 1
Store the control and cost surface parameters in low-speed storage
Has the last block been processed?
No
Yes
Is Key = 2 ?
No
Yes
MODE = 2
Process origin block
No
Key = 1
Is maximum iteration exceeded ?
Yes
STOP

Figure 3.4 Flow diagram for Dynamic Programming with Partitioned State Space.

The optimal control and cost of each point $x \in B_\ell$ is calculated (Step 4) by

$$v(x) = \min_u \{ \ell(x,u) + \beta \{ \bar{v}_m (f(x,u)) + \sum\sum \gamma_{ij}(m) E[\xi_i \xi_j] \} \quad (3.8)$$

where $f(x,u) \in B_m$ with $Z(m)$ known, or B_m is the closest block to the point $f(x,u)$ for which $Z(m)$ is calculated. The iteration variable N has been suppressed since the blocks will be stored back in the same location after they are processed. That is, the stage identity is destroyed. The set of costs, $\{v(x) | x \in B_\ell\}$, is then fitted (Step 5) with a quadratic surface, $Z(\ell)$. For the first pass through state space $(\mathrm{MODE} = 1)$, the control for B_ℓ and the parameters of adjacent blocks are then placed in low-speed storage (Step 7) and Step 2 is repeated.

After all of state space has been considered once, the algorithm goes into $\mathrm{MODE} = 2$ (Step 8). For all subsequent calculations $f(x,u)$ is assured of lying in a calculated block for the evaluation of (3.8). Also, a comparison of the present cost surface and the previous cost surface over the block is made (Step 6) to determine the convergence of the algorithm. Convergence is guaranteed for $\beta < 1$ by (2.12). The process is continued until convergence is attained over all of state space or until a maximum number of iterations is reached.

3.8 Block processing order

Before the algorithm described in the last section may be applied, the partition of state space must be ordered; i.e., an integer must be associated with each block which determines when it will be processed during a pass through state space. The only restriction upon this ordering is that each block be adjacent to a block previously processed during the current processing sequence. This restriction causes the blocks to tend to radiate out through state space from the origin block as they are considered. There is, however, reason to be more selective in the ordering. Namely, it would be ideal if the optimal control, u , at a point x always caused $f(x,u)$ to lie in a block which had already been processed during that pass through state space. This could be accomplished if the optimal control were already known. The block ordering could be taken opposite to the direction $[f(x,u)-x]$, that is, opposite to the direction of the expected transition from x under optimal control. Obviously, if the optimal control were known, the problem would be solved; however, in many problems although the optimal solution is not known, there is some knowledge as to the manner in which the system should be controlled.

This idea was made explicit by Larson with the concept of preferred direction of motion. The preferred direction of motion is, basically, the expected direction in which the trajectories of the system tend under optimal control. The information used in establishing the preferred direction is a priori and rests on an intuitive feeling for the system's behavior. The blocks are then processed opposite to the preferred direction.

If the preferred direction is not known, the algorithm still works and will converge, although more iterations over state space may be necessary. Thus a general technique for ordering the blocks in the absence of a preferred direction is desired. This objective can be achieved in the following way, again suggested by Larson. Let the blocks be designated as in Figure 3.5 where B_0 is the origin block and is defined to have coordinates $B_0 = (0,0)$. The blocks $B_1, \ldots, B_8$ are said to lie in layer one $(L=1)$, $B_9, \ldots, B_{24}$ in layer two, etc. These blocks have coordinates,

$$\begin{aligned} B_1 &= (0,1) \\ B_2 &= (0,-1) \\ &\vdots \\ B_8 &= (-1,1) \\ &\vdots \\ B_{24} &= (-2,2) \ . \end{aligned}$$

The ordering is achieved by counting with 2-digits modulo $M = 2L+1$ for the blocks in layer L. Take for example layer one; counting yields 00, 01, 02, 10, 11, 12, 20, 21, 22. These numbers MOD/3 are associated with the block coordinates (0,0), (0,1), (0,-1), (1,0), (1.1), (1,-1), (-1,0), (-1,1), (-1,-1), respectively, and the block ordering through the first layer is achieved. For the second layer counting MOD/5 yields 00, 01, 02, 03, 04, 10, 11, 12, 13, 14, 20, 21, 23, 24, 30, 31, 32, 33, 34, 40, 41, 42, 43, 44. The MOD/5 digits are associated with the block coordinate elements as follows:

$$\begin{aligned} 0 \text{ MOD}/5 &\Rightarrow 0 \\ 1 \text{ MOD}/5 &\Rightarrow 1 \\ 2 \text{ MOD}/5 &\Rightarrow -1 \\ 3 \text{ MOD}/5 &\Rightarrow 2 \\ 4 \text{ MOD}/5 &\Rightarrow -2 \end{aligned}$$

B_{23}	B_{13}	B_9	B_{11}	B_{18}
B_{21}	B_7	B_1	B_4	B_{16}
B_{20}	B_6	B_0	B_3	B_{15}
B_{22}	B_8	B_2	B_5	B_{17}
B_{24}	B_{14}	B_{10}	B_{12}	B_{19}

Figure 3.5 Block ordering for a second order system.

Thus, the sequence of MOD/5 numbers corresponds to the block coordinates (0,0), (0,1), (0,-1), (0,2), (0,-2), (1,0), (1,1), (1,-1), (1,2), (1,-2), (-1,0), (-1,1), (-1,-1), (-1,2), (-1,-2),(2,0), (2,1), (2,-1), (2,2), (2,-2), (-2,0), (-2,1), (-2,-1), (-2,2), (-2,-2). Deleting those coordinates in layers lower than layer two results in the sequence, (0,2), (0,-2), (1,2), (1,-2), (-1,2), (-1,-2), (2,0), (2,1), (2,-1), (2,2), (2,-2), (-1,0), (-2,1), (-2,-1), (-2,2), (-2,-2) with the associated blocks $B_9, B_{10}, \ldots, B_{24}$. This counting procedure can be carried out through an arbitrary number of layers and for a n^{th} order system. The n^{th} order system would require counting with n-digits MOD/M. A detailed flow diagram and program listing for the ordering of blocks is in Appendix C.

3.9 Calculating the origin block

To initiate the PSS algorithm it is necessary to calculate the quadratic cost surface associated with the origin block for the first pass through state space. This can be done either by dynamic programming using quadratic approximation over the origin block or by policy iteration also employing quadratic approximation.

Howard's policy iteration has application in finding the cost function of the origin block for the continuous state space stochastic control problem. Again, let the quadratic cost surface over the origin block be described by

$$\bar{U}(x) = \alpha + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{i} \gamma_{ij} x_i x_j \quad .$$

Then for a fixed policy $u_0 \in U$ defined for all grid points in the block, it is desired that

$$\bar{U}(x) = \ell(x,u_0) + \beta E\left\{ \bar{U}(f(x,u_0)+\xi \right\} \quad . \tag{3.9}$$

However, there are, in general, more than $(n^2+3n+2)/2$ points in a block for a n^{th} order system. Thus, a least square equation error criteria is used to determine the quadratic fit for the cost function. That is, letting

$$e_x = \ell(x,u_0) + \beta E\left\{ \bar{U}\left(f(x,u_0)+\xi\right)\right\} - \bar{U}(x) ,$$

the functional,

$$J = \sum_{x\in B_0} e_x^2$$

is minimized with respect to $\alpha, \beta_1, \ldots, \beta_n, \hat{\gamma}_{11}, \ldots, \hat{\gamma}_{nn}$. This minimization determines a set of linear equations which in turn define the quadratic surface, $\bar{U}_0(x)$, associated with the policy u_0 . This surface is then used in step two of Howard's policy iteration to determine a new policy $u_1 \in \bar{U}$. The policy iteration is carried out until convergence. It has been found numerically that while this procedure works well at the origin block (containing the minimum point of the cost surface) it does not converge well for other blocks. Thus, it can not be used to find the cost surface for blocks other than the origin.

A second technique to find the cost surface of the origin block is to employ dynamic programming. Assuming a terminal cost of zero, the dynamic programming algorithm can be applied to each point in B_0 , i.e.,

$$U(x) = \min_{u} \left\{ \ell(x,u) \right\} \quad \text{for all } x \in B_0 .$$

This cost function is fitted with a quadratic surface $\bar{v}(x)$, then,

$$v(x) = \min_{u} \left\{ \ell(x,u) + \beta E\left\{ \bar{v}(f(x,u)+\xi)\right\}\right\} \qquad (3.3)$$

is calculated for all $x \in B_0$. Again a quadratic surface $\bar{v}(x)$ is fitted to the cost function $v(x)$ and (3.3) applied. This procedure is carried out until convergence with the speed of convergence described in (2.13).

EXAMPLES

1) Scalar examples

(a)	plant equation -	$x(k+1) = x(k) + u(k) + \xi(k)$
	stage cost -	$\ell(x,u) = x^2 + u^2$
	noise -	$\xi(k) - N(0,1)$
	discount factor -	$\beta = .7$
	state space -	$-10 \leq x \leq 10$
	control	$-2.2 \leq u \leq 2.2$
(b)	plant equation	$x(k+1) = \frac{1}{2}x(k) + \frac{1}{20}x(k)^2 + u(k) + \xi(k)$
	stage cost	$\ell(x,u) = x^2 + u^2$
	noise	$\xi(k) - N(0,1)$
	discount factor	$\beta = .7$
	state space	$-10 \leq x \leq 10$
	control	$-2.2 \leq u \leq 2.2$

Problems (a) and (b) were solved by both dynamic programming and PSS dynamic programming. The state space was partitioned into $B_0 = \{-2 \leq x \leq 2\}$, $B_1 = \{2 \leq x \leq 6\}$, $B_2 = \{-6 \leq x \leq -2\}$, $B_3 = \{+6 \leq x \leq 10\}$ and $B_4 = \{-10 \leq x \leq -6\}$ with $\Delta x = \frac{1}{2}$.

The percentage difference by the two methods in the final cost functions was less than 3%, and the control functions were identical.

2) Second order example

plant equation $x_1(k+1) = x_1(k) + \frac{1}{10} x_2(k) + \xi_1(k)$

$x_2(k+1) = x_2(k) + u + \xi_2(k)$

noise $\xi_1(k)$ and $\xi_2(k)$ independent $N(0,1)$

discount factor $\beta = .7$

state space $-5 \leq x_1 \leq 5$

$-5 \leq x_2 \leq 5$

control $-2.2 \leq u \leq 2.2$

grid $\Delta x_1 = \Delta x_2 = .5$

partition - blocks are square with side 2 units long ,i.e., 25 grid-points per block

This problem was solved by both dynamic programming and PSS dynamic programming. The percentage difference by the two methods of the cost functions was less than 5.4%, while the control functions were identical (within the accuracy of the search). The PSS method took approximately 1/5 the computation time of standard dynamic programming. For the noise $N(0,2.5)$ the cost functions were within 10% of each other, while the accuracy of the control was unaffected.

3) The discount factor interpreted as a reliability probability

Let β = Prob [the system does not fail in one time period]

$1-\beta$ = Prob [the system fails in one time period],

and, let

$\ell(x,u)$ be the stage cost of operating, and

F be the cost of failure. Now the total expected cost is,

$$v(x) = \underset{\xi}{E}\left\{ \sum_{k=0}^{\infty} \beta^{k} \ell(\underset{\sim}{x}(k), u(k)) + \sum_{j=0}^{\infty} (1-\beta)\beta^{j} F \,\middle|\, \underset{\sim}{x}(0) = x_0 \right\}$$

$$= \underset{\xi}{E}\left\{ \sum_{k=0}^{\infty} \beta^{k} \ell(x(k), u(k)) \,\middle|\, \underset{\sim}{x}(0) = x_0 \right\} + F$$

Thus, the cost function to be minimized for the discount factor, β, interpreted as a reliability probability is the same as before except for an additive constant F which does not affect the minima. Therefore, the PSS algorithm can be applied to problems of this nature. In particular, say, to a nuclear rocket control system where the control, u, is applied briefly at the start of a control period and the rocket is allowed to coast for some time with the probability of a system failure being β. The study of a particular system of this nature is under way presently.

BIBLIOGRAPHY

1) Bellman, R. E., "Dynamic Programming Stochastic Control Processes", *Information and Control*, Vol. 1, 1958 pp 228-239

2) Blackwell, D., "Discrete Dynamic Programming", *Annuls of Mathematical Statistics*, 1962, Vol. 33, No. 2, pp 719-726.

3) Blackwell, D., "Discounted Dynamic Programming", *Annuls of Mathematical Statistics*, 1965, Vol. 36, No. 1, pp 226-235.

4) Derman, Cyrus, "On Sequential Control Processes", *Annuls of Mathematical Statistics*, 1964, Vol. 35, No. 1, pp 341-349.

5) Howard, R. A., *Dynamic Programming and Markov Processes*, M. I. T. Press, 1960.

6) Larson, R. E., *State Increment Dynamic Programming*, American Elsevier Publishing Co., 1968.

7) Veinott, A. F., "Discrete Dynamic Programming with Sensitive Discount Optimality Criteria", *Annuls of Mathematical Statistics*, 1969, Vol. 40, No. 5, pp 1635-1660.

PART IV

LINEAR STABILITY OF A NUCLEAR ROCKET ENGINE WITH TWO REACTIVITY FEEDBACKS

by

C. H. Lewis

January 1970

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

ABSTRACT

The nonlinear equations describing nuclear reactor behavior due to reactivity feedback from variation of temperature and moderator density in the core are analyzed. Parameter spaces are defined, and stability boundaries for the linearized system are determined. Analog computer solutions of the linearized equations are presented as verification of the stability of the system.

CHAPTER 1

INTRODUCTION

A large amount of work has been done in the past on the problem of analyzing a nuclear reactor with two reactivity feedback mechanisms for stability. In a recent doctoral dissertation,[1] Schmidt concentrated upon a system with feedback from two different temperature regions. In his work, a parameter space is defined, and stability boundaries are plotted. In the following paper, a method for determining coordinates for such parameter spaces and stability boundaries in general is demonstrated, by application to a system presented by Smith and Stenning.[2]

Analog computer solutions to the linearized system equations are presented in the form of state space plots of the system power vs. that system state which gives rise to the prompt reactivity feedback.

CHAPTER 2

OPEN LOOP SYSTEM EQUATIONS

The differential equations which describe the system to be studied are presented in this chapter. These include the point reactor model, the prompt jump approximation, and the reactivity feedback model corresponding to feedback due to a temperature variation and a gas density or pressure variation in the core.

The equations are normalized and linearized about an equilibrium operating point, because the normalized form is more convenient to use in the analog computer simulation of the system, and because the method of stability analysis to be used is applicable only to the linearized form. The feedback loop transfer function is determined, and its pole-zero plot is included as an aid in visualizing the system dynamics.

Neutron Kinetics Equations

The point reactor model is well known, and its derivation will not be repeated here. In the case of one delayed group of neutrons, the source free equations may be written

$$\frac{dN}{dt} = \frac{\rho - \beta}{\ell} N + \lambda \Gamma \qquad (2\text{-}1a)$$

$$\frac{d\Gamma}{dt} = \beta/\ell \; N - \lambda \Gamma \qquad (2\text{-}1b)$$

where N is the mean neutron density or power

Γ is the mean delayed neutron precursor concentration,

ρ is the reactivity of the system,

ℓ is the neutron generation time,

3 is the delayed neutron fraction, $\beta = \sum_{i=1}^{m} \beta_i$

and

λ is the mean weighted precursor decay constant,

$$\frac{1}{\lambda} = \frac{1}{\beta} \sum_{i=1}^{m} \frac{\beta_i}{\lambda_i}$$

Since the derivation of this lumped parameter model assumes that the delayed neutron precursors remain very close to the spot at which they were created, it is not entirely applicable to the case of a rocket engine in which the core is made of graphite, and the precursors are said to diffuse rapidly and may be swept out with the propellant before releasing a neutron. This difficulty may be partially circumvented by using a modified value of β, the delayed neutron fraction.

The equations are normalized about the equilibrium operating values N_0 and Γ_0:

$$\frac{dN'}{dt} = a \left(\rho' N' - N' + \Gamma' \right) \qquad (2-2a)$$

$$\frac{d\Gamma'}{dt} = b \left(N' - \Gamma' \right) \qquad (2-2b)$$

where $a = \beta/\ell$ $\quad b = \lambda$, $\quad \rho' = \frac{\rho}{\beta}$, $\quad N' = \frac{N}{N_0}$, an [illegible]

The linearization is accomplished by expanding these equations in a Taylor series about the equilibrium operating point X_0, neglecting higher order terms.

$$\frac{d\delta N'}{dt} = a(-\delta N' + \delta\Gamma' + \delta\rho') \qquad (2-3a)$$

$$\frac{d\delta\Gamma'}{dt} = b(\delta N' - \delta\Gamma') \qquad (2-3b)$$

In terms of the original system variables, $\delta X = \frac{X - X_0}{X_0}$

If $\frac{dN}{dt}$ is small compared to $|\frac{\rho - \beta}{\ell} N|$ or $\lambda\Gamma$ in equation 2-1a, it may be neglected. Equation 2-1a then yields $\Gamma = \frac{\beta - \rho}{\ell} N$.

Substituting this value of Γ into equation 2-1b and simplifying,

$$(\beta - \rho)\frac{dN}{dt} = \left(\lambda\rho + \frac{d\rho}{dt}\right)N \qquad (2-4)$$

This is the nonlinear prompt jump approximation to the one delayed group point reactor kinetics equation. Normalizing about the equilibrium point as was done previously,

$$\frac{dN'}{dt} = \frac{\left(\lambda\rho + \frac{d\rho}{dt}\right)}{\beta - \rho} N' \qquad (2-5)$$

Linearizing,

$$\frac{d\delta N'}{dt} = b\,\delta\rho' + \frac{d\delta\rho'}{dt} \qquad (2-6)$$

Feedback Equations

The feedback system to be studied represents a proposed nuclear rocket engine in which the reactor is used to impart high energy to hydrogen propellant, which moves through the reactor, acting as coolant and moderator, and is then expelled from the nozzle. The prompt reactivity feedback mechanism in this system is the temperature, which causes expansion of the graphite core. An increase in temperature leads to an increase in hydrogen moderator pressure, or a decrease in density. This is the delayed feedback mechanism.

The nonlinear equations for temperature-pressure feedback are given below:

$$\frac{dT}{dt} = \theta N - \phi_1 P T^{1/2} \qquad (2-7a)$$

$$\frac{dP}{dt} = \phi_2 P T^{1/2} - \phi_3 P^2 T^{-1/2} \qquad (2-7b)$$

$$\rho = -A_1 T - A_2 P T^{-1} + \rho_{ex} \qquad (2-8)$$

where θ and ϕ_i have replaced the somewhat more complicated coefficients of Smith and Stenning. The external reactivity term is necessary to insure that $\rho_o = 0$, since the temperature and pressure are absolute quantities and are always greater than zero.

Upon normalization,

$$\frac{dT'}{dt} = c\left(N' - P'T'^{1/2}\right) \qquad (2\text{-}9a)$$

$$\frac{dP'}{dt} = d\left(P'T'^{1/2} - P'^{2}T'^{-1/2}\right) \qquad (2\text{-}9b)$$

$$\rho' = -A_1'T' - A_2'P'T'^{-1} + \rho'_{ex} \qquad (2\text{-}10)$$

where c and d are defined as the normalizing constants for the equations describing the behavior of the system variables which control the prompt and delayed feedback mechanisms, respectively.

$$c = \frac{\omega n_0}{T_0} = \frac{\phi_1\phi_2}{\phi_3}T_0^{1/2}, \quad d = \phi_2 T_0^{1/2}, \quad \rho' = \rho/\beta$$

$$A_1' = \frac{A_1T_0}{\beta} \qquad A_2' = \frac{A_2P_0}{\beta T_0}$$

The useful relation $\phi_2 T_0 = \phi_3 P_0$ is discovered as an equilibrium condition of equation 2-9b.

The normalized equations are linearized to

$$\frac{d\,\delta T'}{dt} = c\left(\delta N' - \delta P' - \tfrac{1}{2}\delta T'\right) \qquad (2\text{-}11a)$$

$$\frac{d\,\delta P'}{dt} = d\left(\delta T' - \delta P'\right). \qquad (2\text{-}11b)$$

$$\delta\rho' = (A_2' - A_1')\,\delta T' - A_2'\,\delta P' \qquad (2-12)$$

In matrix form,

$$\begin{bmatrix} \frac{d\delta T'}{dt} \\ \frac{d\delta P'}{dt} \end{bmatrix} = \begin{bmatrix} -c/2 & -c \\ d & -d \end{bmatrix} \begin{bmatrix} \delta T' \\ \delta P' \end{bmatrix} + \begin{bmatrix} c \\ 0 \end{bmatrix} \delta N'$$

$$\delta\rho' = \begin{bmatrix} (A_2' - A_1') & -A_2' \end{bmatrix} \begin{bmatrix} \delta T' \\ \delta P' \end{bmatrix}$$

These equations are in the standard form $\underline{\dot{x}} = \underline{Ax} + \underline{b}u$, $y = \underline{c}^T\underline{x}$, with $\delta N'$ as the control input u, and the output $y = \delta\rho'$.

The feedback loop transfer function is given by

$$\frac{\delta\rho'}{\delta N'} = \underline{c}^T(s\underline{I} - \underline{A})^{-1}\underline{b} .$$

$$-\frac{\delta\rho'}{\delta v'} = \frac{c(A_1' - A_2')\left(s + \frac{dA_1'}{A_1' - A_2'}\right)}{(s+\eta_1)(s+\eta_2)},$$

$$\eta_{1,2} = \tfrac{1}{2}\left(\tfrac{c}{2} + d\right) \pm \tfrac{1}{2}d\sqrt{\frac{c^2}{4d^2} - \frac{5c}{d} + 1} \qquad (2-13)$$

For real roots, the discriminant in equation 2-13 must be greater than or equal to zero, or

$$\frac{c}{d} < .20 \quad \text{or} \quad \frac{c}{d} > 19.8 \qquad (2-14)$$

A pole-zero plot for the feedback system for c/d <.20 is shown in Figure 2.1.

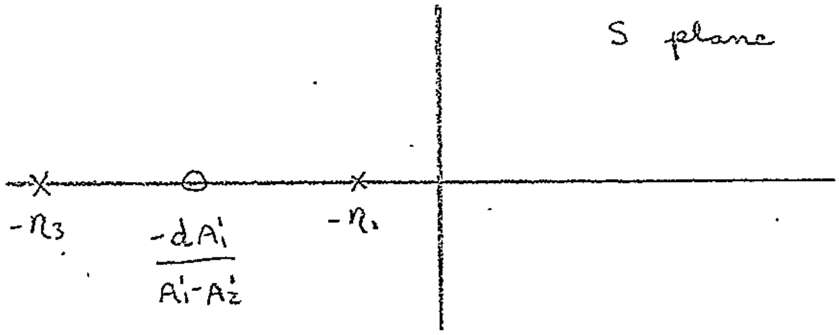


Figure 2.1 Pole-zero plot of -H(s) for Temperature-Pressure Feedback

Unfortunately, the position of the poles and zero is not independent of the equilibrium point about which we choose to linearize and normalize our equations. Had this position been determined only by θ and ϕi, we could choose any equilibrium point and derive stability criteria valid for every equilibrium point in the linearized system. If the system states varied at a reasonable rate, we could assume the system remained close to some equilibrium (not necessarily the starting equilibrium) at all time, and their position in the parameter space relative to the boundaries remained fixed during a short-term perturbation.

If the quantities which define the stability curves are functions of the particular equilibrium point, however, the effect will be to move the stability curves around the parameter space during an excursion. Whether this results in a larger or smaller region of stability than predicted remains to be seen.

CHAPTER 3

CLOSED LOOP SYSTEM

In this chapter, the neutronic behavior predicted by a) the prompt jump approximation, and b) the one delayed group point kinetic model is coupled with the feedback equations. A parameter space is defined, and the stable and unstable portions of it are determined. The results of analog computer solutions of the linearized equations corresponding to various points in the parameter space are presented.

Derivation of the Stability Planes

The linearized prompt jump equation is repeated here for convenience:

$$\frac{d\delta N'}{dt} = b\,\delta\rho' + \frac{d\delta\rho'}{dt} \qquad (2-6)$$

From equation 2-12,

$$\frac{d\delta\rho'}{dt} = (A_2' - A_1')\frac{d\delta T'}{dt} - A_2'\frac{d\delta P'}{dt}$$

Substituting for

$\frac{d\delta T'}{dt}$, $\frac{d\delta P'}{dt}$ from equations 2-11a, 2-11b,

$$\frac{d\delta N'}{dt} = c\,(A_2' - A_1')\,\delta N' + \left[(c/2 - b)A_1' + (b - c/2 - d)A_2'\right]\delta T'$$

$$+ \left[cA_1' + (d - b - c)A_2'\right]\delta P'$$

$$\begin{bmatrix} \frac{d\delta N'}{dt} \\ \frac{d\delta T'}{dt} \\ \frac{d\delta P'}{dt} \end{bmatrix} = \begin{bmatrix} c(A_2' - A_1') & (\frac{c}{2} - b)A_1' + (b + c/2 - d)A_2' & cA_1' + (d - b - c)A_2' \\ c & -c/2 & -c \\ 0 & d & -d \end{bmatrix} \begin{bmatrix} \delta N' \\ \delta T' \\ \delta P' \end{bmatrix}$$

This is the closed loop system matrix. If its eigenvalues all have negative real parts, the system is asymptotically stable. A necessary condition for this is that the coefficients of the system characteristic equation not change sign. We will form the characteristic equation and compute the Hurwitz determinants. The conditions assuring their positiveness will lead to the stability boundaries in the parameter space.

The characteristic equation is

$$\mu^3 + \left[cA_1' - cA_2' + (c/2 + d)\right] u^2 + \left[(b+d)cA_1' - bcA_2' + \frac{3cd}{2}\right] \mu + bcdA_1' = 0.$$

The Hurwitz determinants are

$$H_0 = a_0 = 1$$

$$H_1 = a_1 = cA_1' - cA_2' + (c/2 + d)$$

$$H_2 = a_1 a_2 - a_3 = (b+d)(cA_1')^2 - (2b+d)(cA_1')(cA_2') + b(cA_2')^2$$

$$+ \left[c/2(b+3d) + d(c/2 + d)\right](cA_1') - \left[\frac{bc}{2} + bd + \frac{3cd}{2}\right](cA_2')$$

$$+ \frac{3cd}{2}\left(\frac{c}{2} + d\right)$$

$$H_3 = a_3 H_2 = (bcdA_1') \cdot H_2$$

If we let $X = CA_1' = \frac{\Theta N_0}{\beta} A_1$, $y = CA_2' = \frac{\Theta N_0 \phi_2}{\beta T_0 \phi_3} A_2$

and set the determinants equal to zero, the curves defined will be the boundaries between stable and unstable systems. H_3 is the static stability line; H_2 the resonance line. The space, as it might appear when typical operating parameters are substituted for b,c, and d, is shown in Fig. 3.1.

$$H_1: \quad y = x + (c/2 + d)$$

$$H_2: \quad (b+d)x^2 - (2b+d)xy + by^2 + \left[c/2(b+3d) + d(c/2+d)\right]x$$
$$- \left(\frac{bc}{2} + bd + \frac{3cd}{2}\right)y + \frac{3cd}{2}\left(\frac{c}{2}+d\right) = 0$$

$$H_3: \quad bdx = 0$$

The static stability line H_3 is in this case situated on the y-axis, which means that it is independent of the equilibrium value we choose to calculate c and d. Furthermore, the analog computer studies detailed in the next section show that the system behavior is quite insensitive to the value of y chosen if x is held constant. The effect of the "moving" stability boundaries referred to in the last chapter should therefore not be too great.

To determine the region of stability if the prompt jump model is replaced by the one delayed group point reactor model, we will repeat

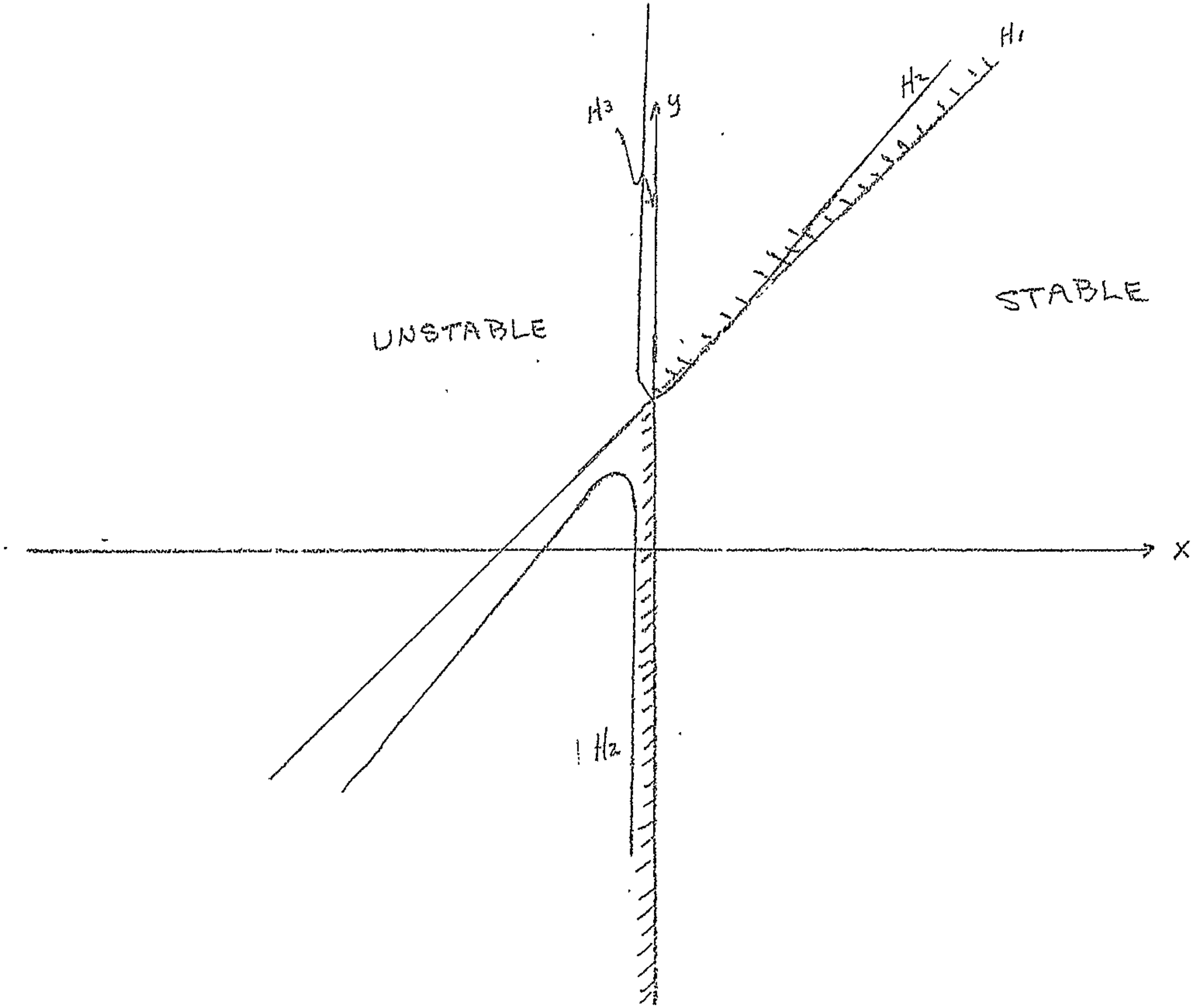


Fig. 3.1

Typical Parameter Space for c/D ≐ .2

the procedure outlined previously; i.e. find the linearized system closed loop matrix, determine the characteristic equation, form the Hurwitz determinants and set them equal to zero. The resulting equations are the stability boundaries in a parameter space whose coordinates are the product of the feedback coefficients and any set of constants by which they are consistently multiplied in the equations.

The system is defined by equations 2-3a, 2-3b, 2-11a, 2-11b, and 2-12. If the closed loop system matrix

$$\begin{bmatrix} \frac{d\delta N'}{dt} \\ \frac{d\delta \Gamma'}{dt} \\ \frac{d\delta T'}{dt} \\ \frac{d\delta P'}{dt} \end{bmatrix} = \begin{bmatrix} -a & a & a(A_2' - A_1' & -aA_1' \\ b & -b & 0 & 0 \\ c & 0 & -c/2 & -c \\ 0 & 0 & d & -d \end{bmatrix} \begin{bmatrix} \delta N' \\ \delta \Gamma' \\ \delta T' \\ \delta P' \end{bmatrix}$$

is manipulated as before, the stability boundaries are

$$H_2: \quad a(a+c/2)x - a(a+c/2+d)y + (c/2+d)\left[(a+b)(a+b+c/2+d) + \frac{3cd}{2}\right] = 0$$

$$H_2: \quad a^2(a+c/2)(b+d)x^2 - a^2\left[(2b+d)(a+c/2+d/2) + \frac{d^2}{2}\right]xy$$

$$+ a^2b(a+c/2+d)\,y^2 + a\Big\{\left[c/2(a+b)\right]\left[b(a+b+c/2+d) + 3d(a+c/2)\right]$$

$$+ \left[d(c/2+d)\right]\left[a(a+b+c/2+d) + \frac{3c}{2}(b+d)\right]\Big\}x$$

$$- a\left[(a+b+c/2+d)\left(b(a+b)(c/2+d) + \frac{3acd}{2}\right) + \frac{3bcd}{2}(c+d)\right]y$$

$$+ (a+b)\left(\frac{cd}{2}+d\right)\left(\frac{3cd}{2}\right)\left[(a+b)(a+b+c/2+d) + \frac{3cd}{2}\right] = 0$$

$$H_3: \quad abdx = 0$$

Analog Computer Simulation Results

The University of Arizona's Computer Systems, Inc. 5800 analog computer was used to simulate the linearized equations to verify the stability plane results. For these runs, two specific cases were chosen; one corresponds to the parameters used by Wiberg and Woyski,[3] for which the feedback loop poles are real, and the other is a fabricated case in which the feedback loop poles are complex.

Case A

A system with No = 2000 MW, T_o = 2000°K, $\theta = .2$, $\phi_1/\phi_3 = .06$, $\phi_2 = .0745$, and $\lambda = .1\ \text{sec}^{-1}$ would have normalized time constants $b = .1$, $c = .2$, $d = 3.33$. Since $c/d = \phi_1/\phi_3 = .06$, condition 2-14 for real poles is met.

The parameter space with these values substituted for the inverse time constants is shown in Figure 3.2. The values of A_1', A_2' corresponding to points A-L on Fig. 3.2 are listed in Table 3.1.

Table 3.1
Parameter Space Locations of
Points in Figs. 3.2, 3.4

Point	A_1'	A_2'
A	-1	-2
B	1	-2
C	2	-2
D	2	0
E	2	2
F	2	5
G	2	6
J	2	10
K	2	15
L	2	20

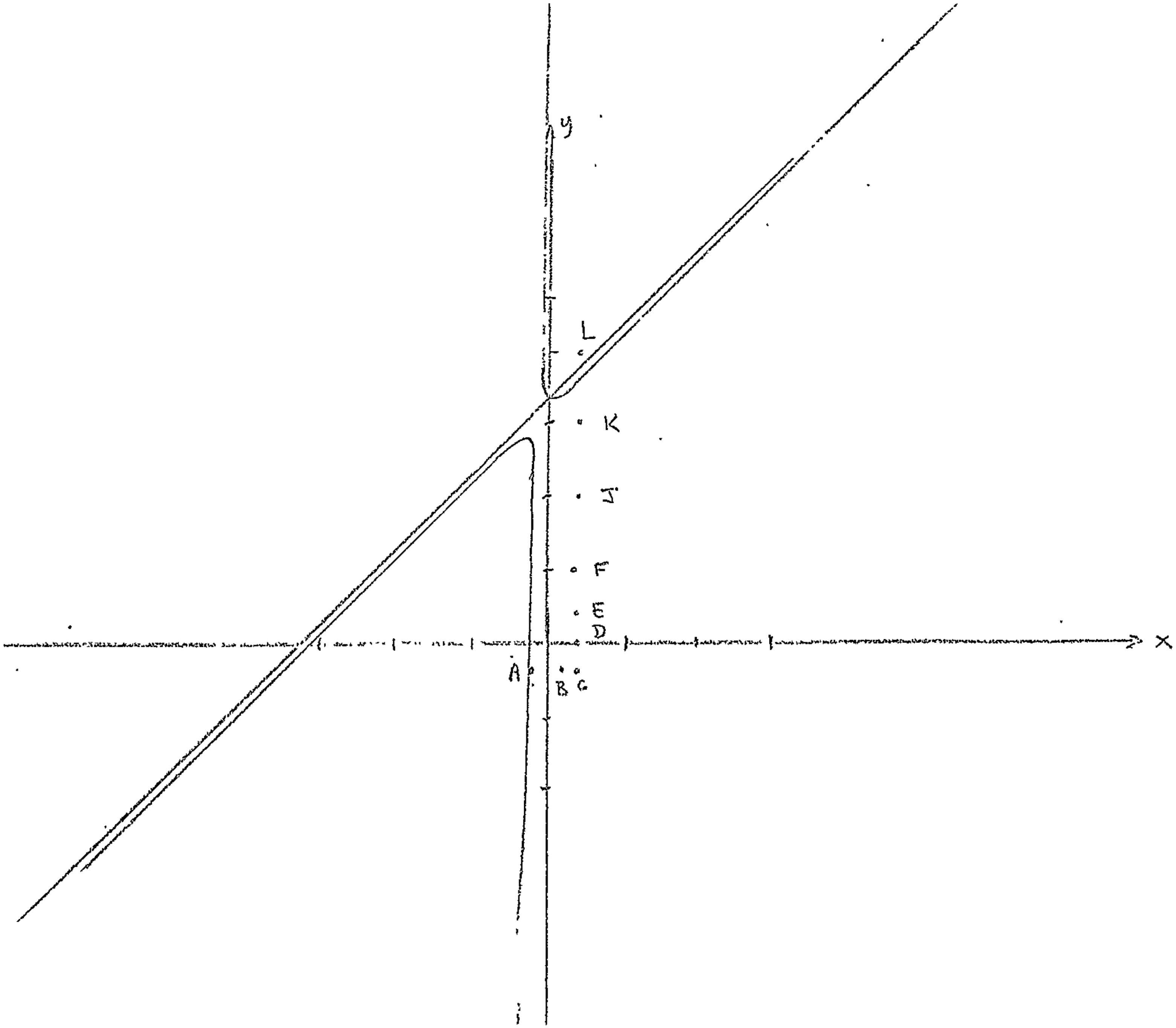


Fig. 3.2

Parameter Space for Case A

Response to a nonequilibrium N'(0) is shown in Fig. 3.3. As was predicted, the system is unstable for $x<0$. The behavior at points K and L indicates that the stability boundary is where it was predicted to be. (Note, however, that this space is valid only near the operating level, and we cannot draw any conclusions from it about very low power operation.) The system may be driven unstable, but A'_2 must be very large.

Use of the stability boundary equations determined for one delayed group neutronics results in a very slight shifting of the boundaries. The prompt jump equation is seen to be a good approximation to the point reactor kinetic equations in this case.

Case B

If $\phi_1/\phi_3 = .4$, with all the other parameters remaining the same as in Case A, c = .710, d = 1.775, and c/d = .4. The poles of the feedback loop transfer function are now complex. The new stability boundaries are shown in Fig. 3.4. Resonant behavior may be expected as the value of y is increased for $x>0$. State space plots for the linearized equations are shown in Fig. 3.5. Sustained oscillations are observed in the vicinity of point F, and the system is unstable at point G.

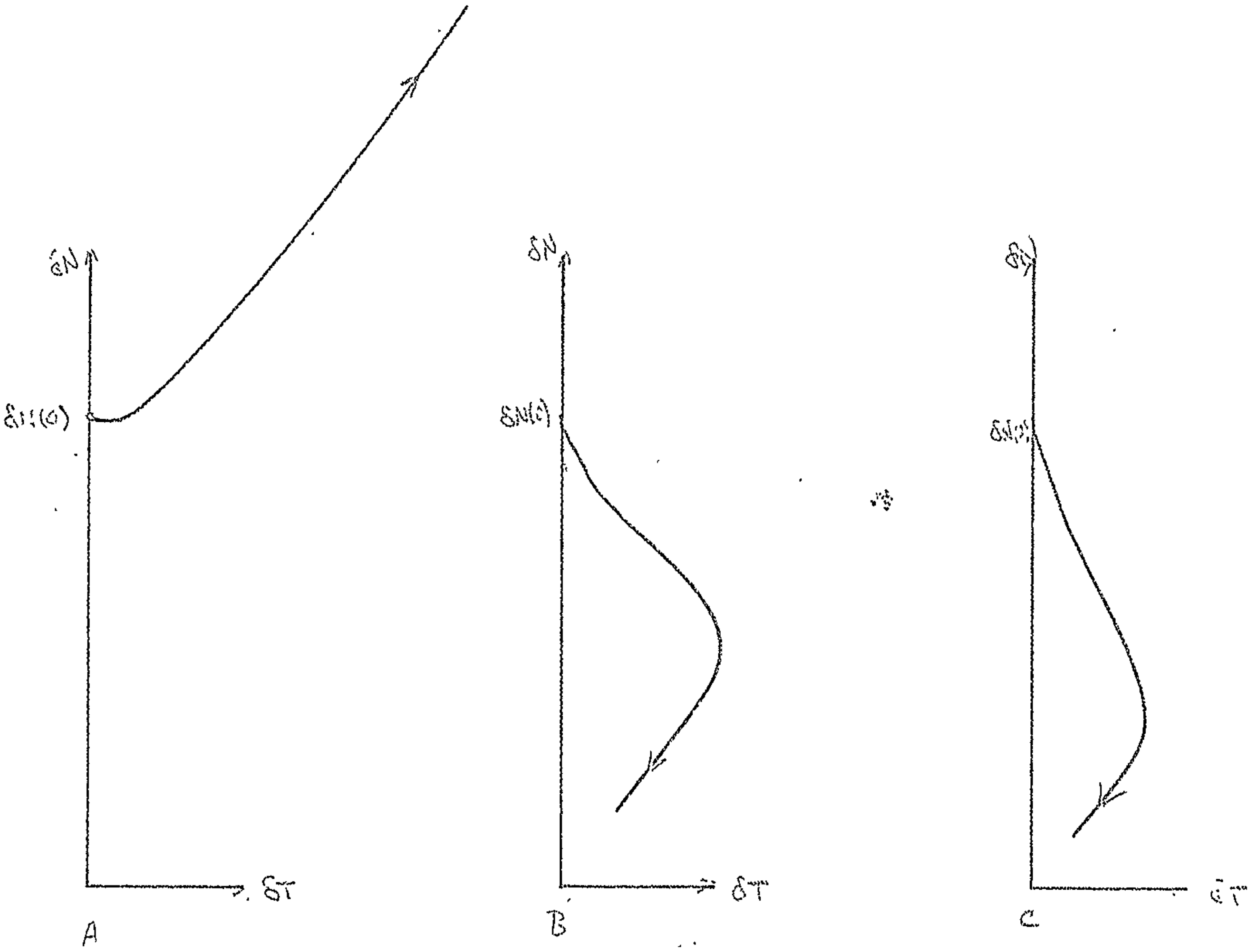


Fig 3.3
State Space Paths for Case A

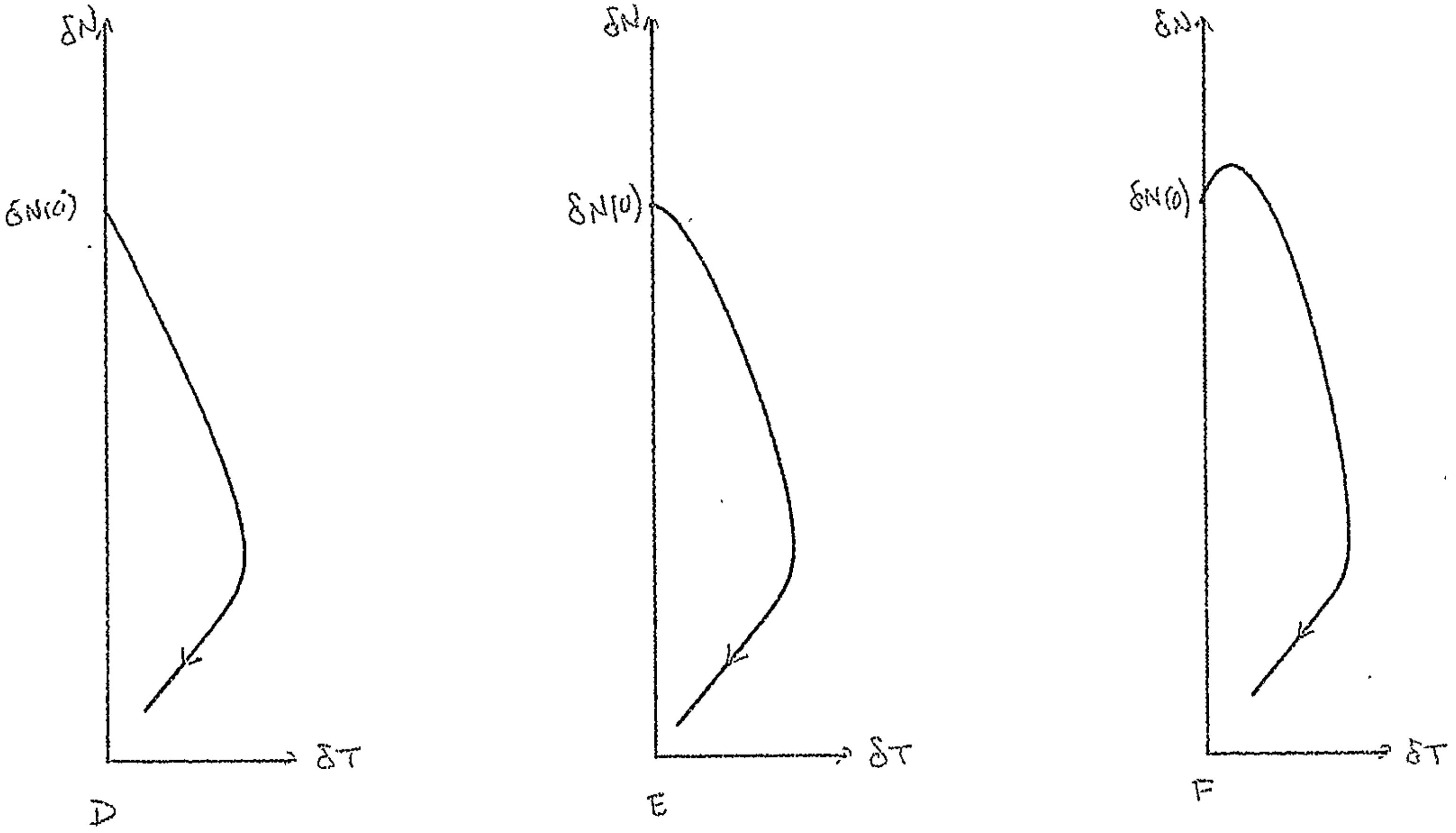


Fig 33 cont.

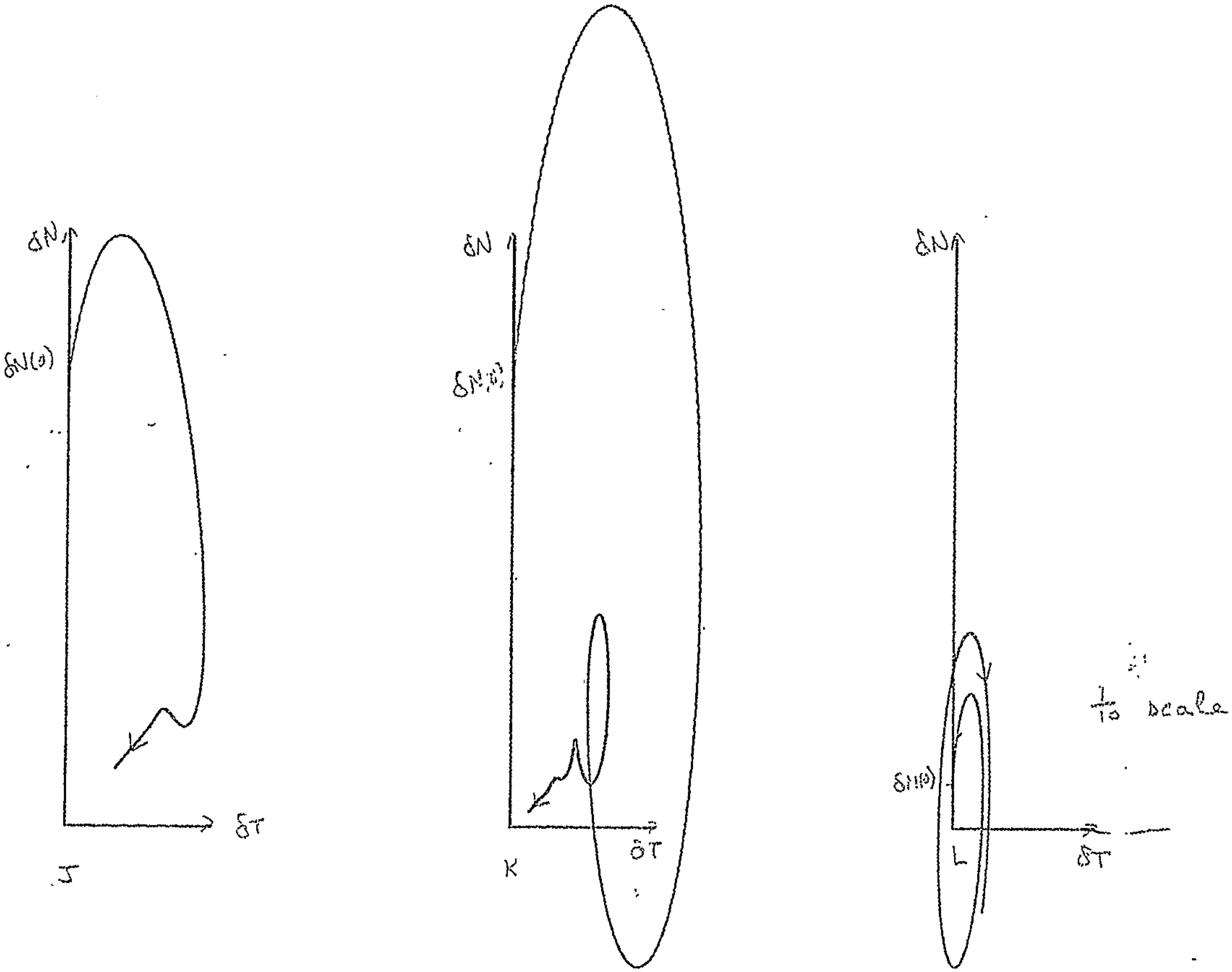


Fig. 3.3 cont.

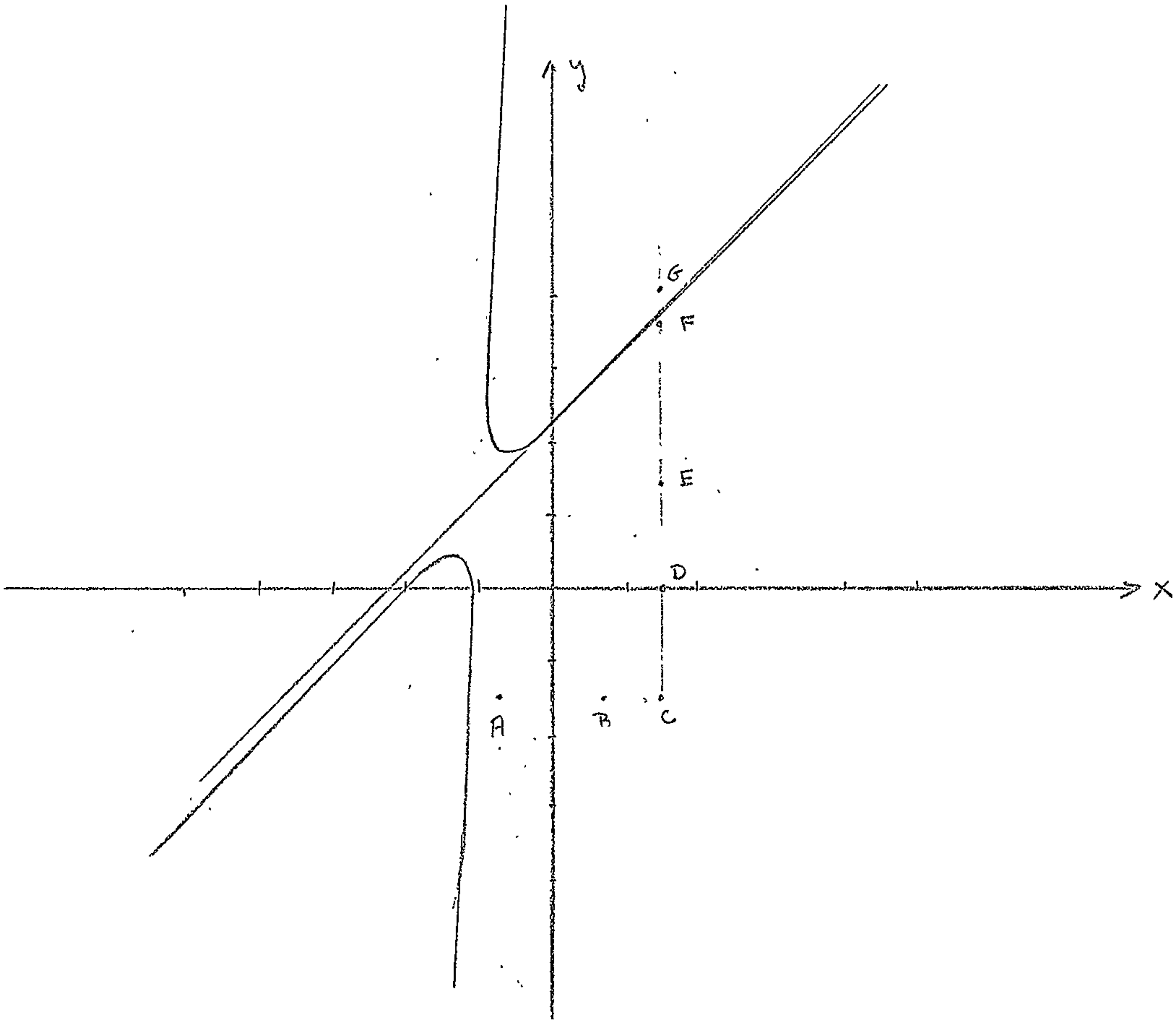


Fig. 7.

Parameter Space for Case B

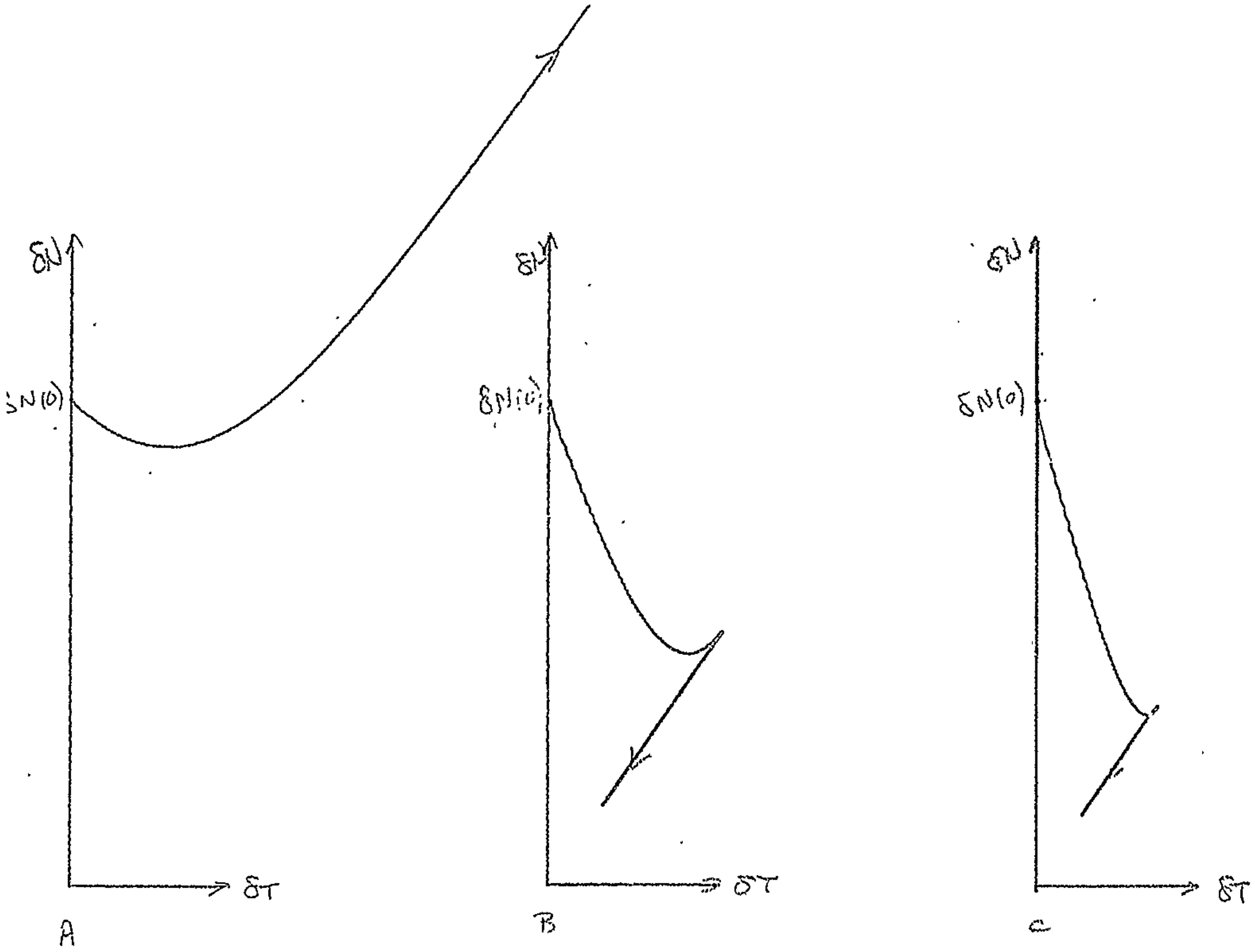


Fig. 3.5

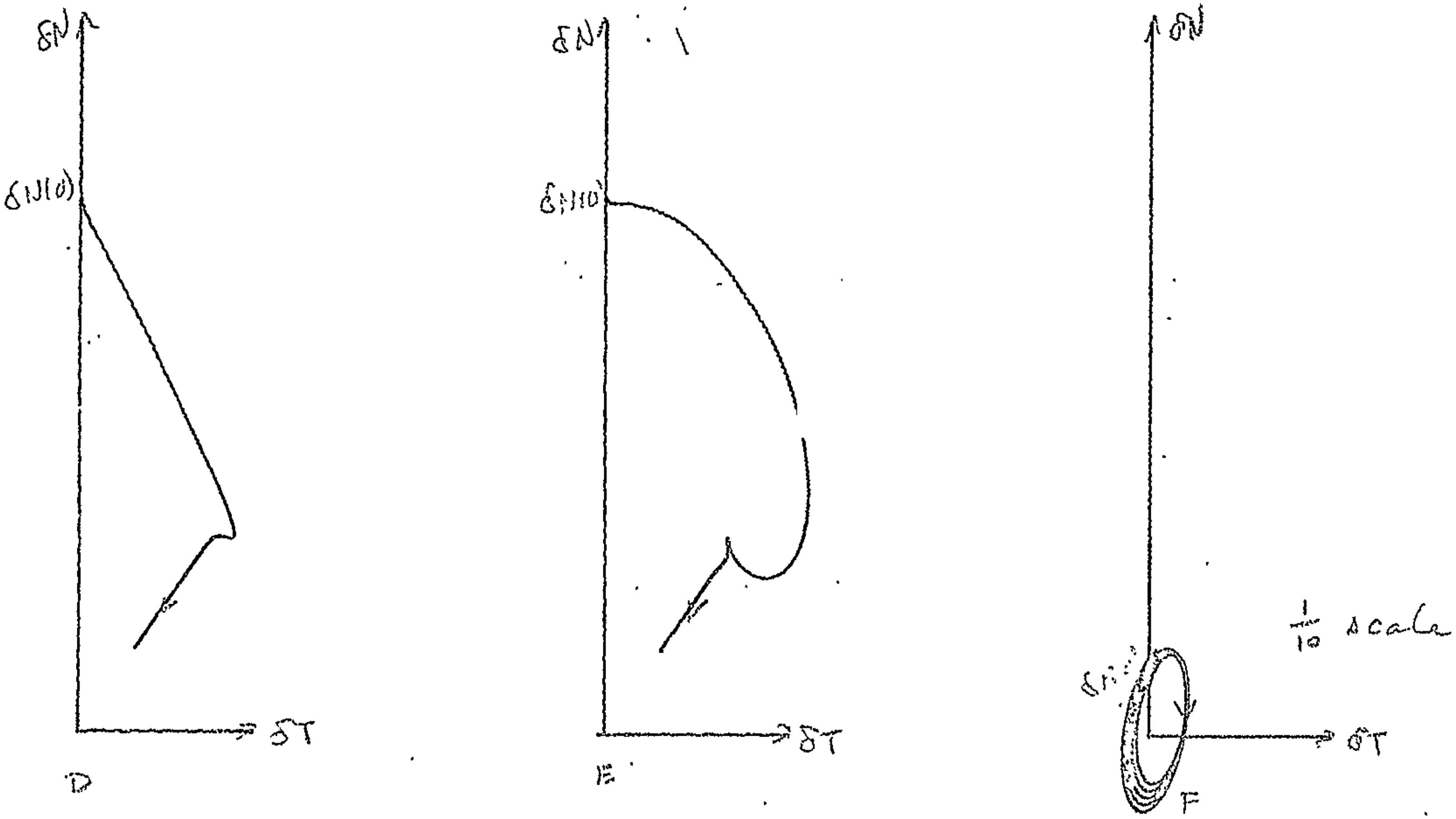
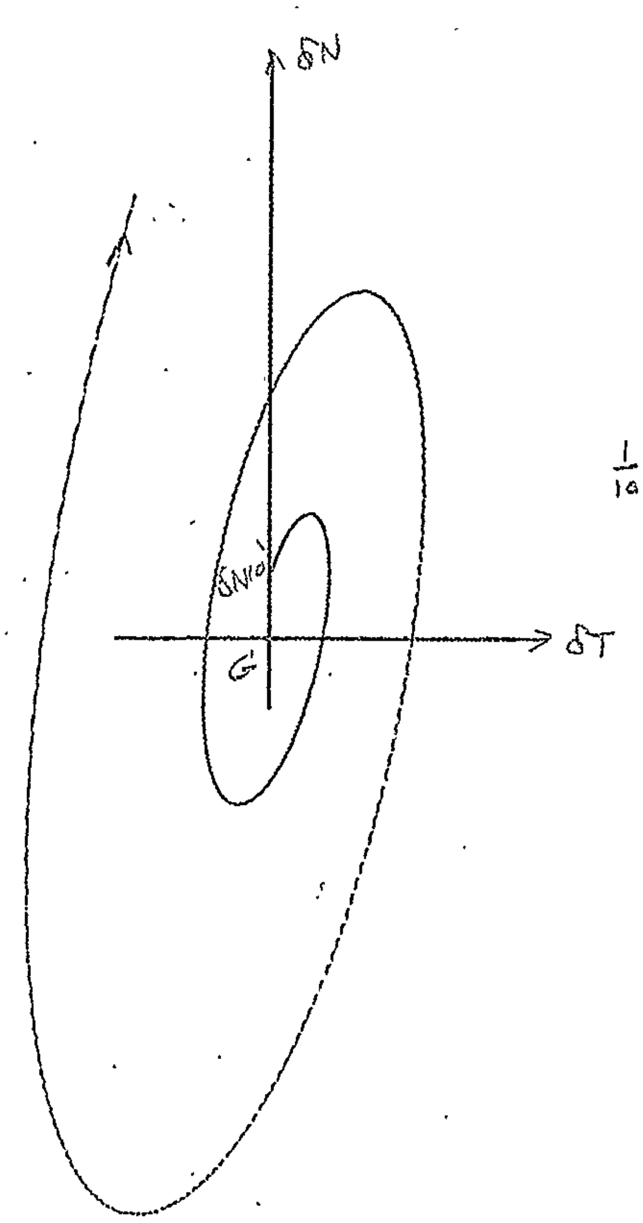


Fig 3,5 cont



$\frac{1}{10}$ scale

Fig 3.5, cont.

Chapter 4

CONCLUSION

The nonlinear equations describing the behavior of a proposed nuclear rocket engine were linearized and analyzed. Regions of linear stability and instability in a parameter space were delineated, and a general method for the determination of the coordinates of such parameter spaces was outlined.

Two particular systems were chosen for further investigation. They corresponded to feedback system transfer functions with and without complex poles. The system with complex poles exhibited oscillatory behavior for certain values of feedback coefficients, as was predicted.

Much work remains to be done on the problem of determination of the regions of stability for this system. A digital computer code has been written to help in the plotting of the stability boundaries. With its help, an investigation of the effect of changing only the equilibrium power on the stability boundaries may be carried out.

An attempt has been made to simulate the nonlinear equations on the analog computer, but due to the complexity of the problem and inherent inaccuracy of the nonlinear computing devices, the results were deemed unreliable. However, arrangements have been made with the Electrical Engineering Department for use of their PDP-9 digital computer. The problem will be coded in DARE, a new digital simulation language. It is hoped that with this tool, the nonlinear systems equations may be solved and used to verify the predicted system stability.

REFERENCES

1. Schmidt, T. R., Nonlinear Oscillations and Stability of a Nuclear Reactor with Two Reactivity Feedbacks, Doctoral dissertation, University of Arizona, June 1969.

2. Smith, H. P., and A. H. Stenning, "Open Loop Stability and Response of Nuclear Rocket Engines", NSE 11, 76-84 (1961).

3. Wiberg, D. M., and J. S. Woyski, "Stability of Nuclear Rocket Engine Dynamics", Nucl. Appl. 5, 35-45 (1968).