FOREWORD

This report was prepared by the Department of Engineering at the University of California, Los Angeles, on Air Force Contract AF33(657)-11477 under Task No. 822501 of Project No. 8225, *Research in Advanced Applications of Sampled Data and Other Non-Linear Control Systems Theory.* The work was administered under the direction of AF Flight Dynamics Laboratory, Research and Technology Division. Charles Harmon was project engineer for the laboratory.

The studies presented represent the efforts from May 1963 to May 1964. The authors are members of the research department at the University of California, Los Angeles. Professor C. T. Leondes served as principal investigator.

This report is based on a dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Engineering at the University of California, Los Angeles.

This is the final report and it concludes the work on AF33(657)-11477. The contractor's report number is FDL-TDR-64-48.

This document is unclassified.

ABSTRACT

In an effort to bridge the gap between theory and practice, the re-entry problem area is chosen to apply advanced control techniques. This problem area affords many separate areas where theory could be applied.

This volume is separated into two parts: Part 1 describes digital computer controlled adaptive processes which can be employed for the flight path control problem; i.e., control of a vehicle about a predetermined trajectory (optimal or otherwise). In Part 2 optimal re-entry trajectories are determined.

This technical documentary report has been reviewed and is approved.

W. A. SLOAN, Jr.
Colonel, USAF
Actg. Chief, Flight Control Division
Air Force Flight Dynamics Laboratory

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

ILLUSTRATIONS (Continued)

TABLES

SYMBOLS

$*$	transpose
$\dot{\ }$, d/dt	derivative
-1	inverse
$\dagger$	generalized pseudo inverse
t	time, independent variable
T	sampling period
T^1	observation interval, optimization interval
T_1	operation interval
$\lvert\quad\rvert$	absolute value
$\lVert\quad\rVert$	Euclidean norm
$<\ ,\ >$	scalar product
E	expectation
$\hat{\ }$	optimum estimate
$\underset{-M,\ M}{\text{sat}(a)}$	$\begin{cases} M & \text{if } M < a \\ a & -M < a < M \\ -M & a < -M \end{cases}$
$\underline{\ }$	vector
$\sim$	vector formed by points sampled at discrete intervals
z	z-transform variable
s	Laplace transform variable
$\lVert x\rVert_Q^2$	quadratic form, $x^* Qx$
I	identity matrix
ϵ	"is a member of"
$\triangleq$	equal by definition
$\underset{b}{\text{Min}}(a)$	choose b to minimize a
$\rightarrow$	approaches
$\Rightarrow$	implies
∇_x	gradient with respect to x

SYMBOLS (Continued)

$a \oplus b$	product space of a and b
$\doteq$	left-hand side is to be chosen so that it best approximates the right-hand side in the least squares sense (or, vice versa)
Δ	$\frac{1}{2}$ the confidence interval
k	present time
$\Big\vert_o$	evaluated at nominal or optimal point

CHAPTER 1
INTRODUCTION

1.1 General Statement

Part 1 of this volume is concerned with the problem of controlling processes under the condition of uncertain changes in the process to be controlled. Of course, the feedback principle solves this problem to some extent. Larger process variations and increased accuracy requirements dictate, however, more sophistication in the control. Control systems designed specifically to consider these problems have been called "adaptive" control systems.

Independent of the study on adaptive controls, engineers and mathematicians have been concerned with optimal controls, i.e., the computation of controlling forces for a known process which minimizes some performance criterion. One can use results from optimal controls for the adaptive control problem if first identification is made on the process. Such a philosophy has been taken by Kalman, Merriam, Braun, Meditch, and Hsieh (References 1-5). This problem area will also be the concern of our investigations.

The remainder of this chapter will be divided into four parts. First, a definition of adaptive controls will be given to set the general framework for our discussions. Second, background material will be given which is pertinent to the present study. Third, the purpose and goal of our endeavor will be stated. Finally, the organization of Part 1 will be given.

1.2 Definitions

In order to effectively treat the subject of adaptive control systems it is desirable to give a definition for this category of control systems. The term "adaptive" has been attached to a wide variety of control systems. It is the intent here to set forth a definition which will encompass the various adaptive systems given in the past.

Before stating what we mean by adaptive control systems, let us define the expression "acceptably performing system".† This term will describe the external manifestations of the system under investigation. The goal for a control system design should be clear to the designer as the first step in his design process. The definitive delineation of "acceptably performing system" is an attempt to express quantitatively whether the designer has attained this goal. Let us first describe some terminology. With reference to Figure 1, let

† Zadeh (Reference 6) uses "adaptivity" for "acceptably performing systems". His definition for adaptive system is used to define acceptably performing systems.

Manuscript released by the author May 1, 1964 for publication as an FDL Technical Documentary Report.

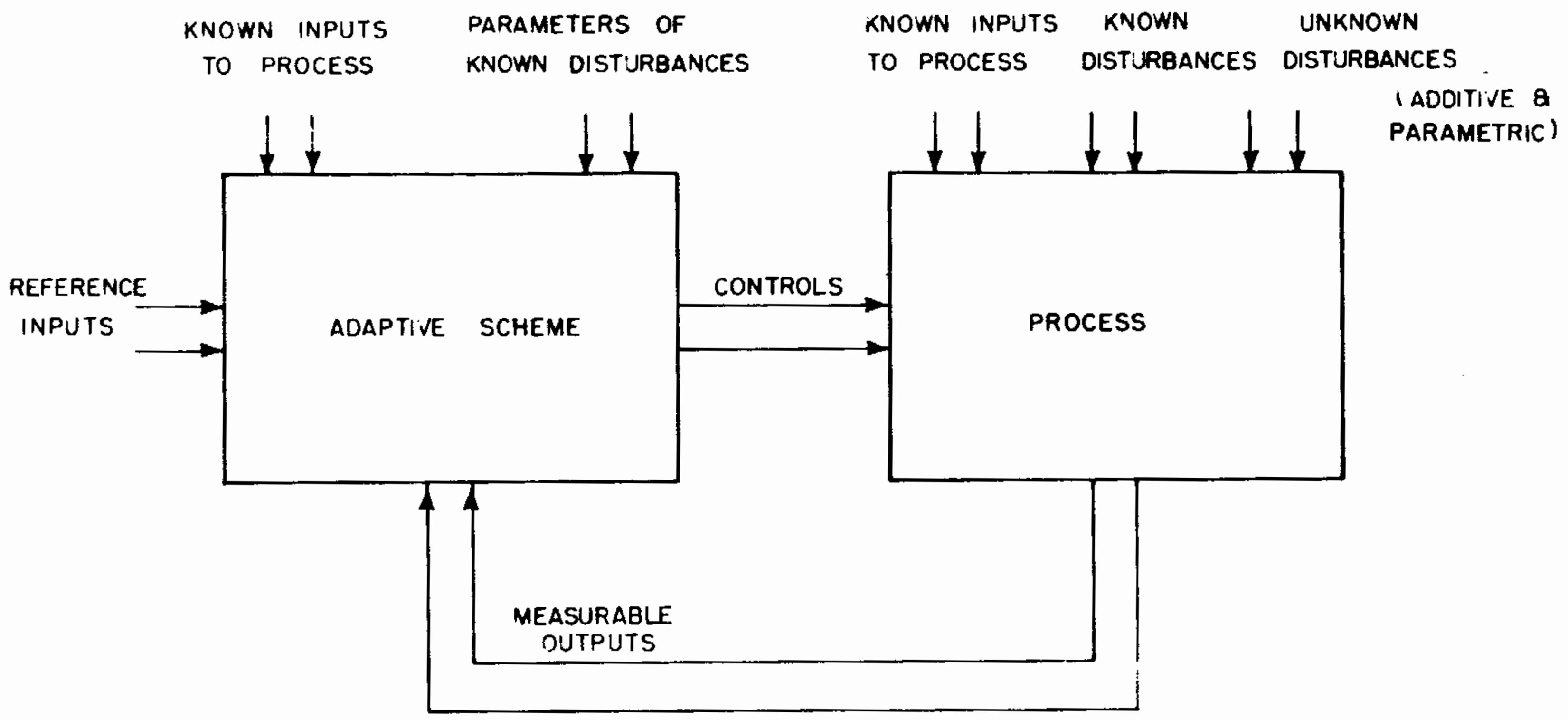


Figure 1. Adaptive Control Systems

$\mathcal{A} \equiv$ system including process and controller

$v(t) \equiv$ vector function defined for the interval of operation $0 \leq t \leq T$, and composed of (some can be missing): (a) reference inputs, (b) known inputs to the process, (c) disturbances to the process, (d) measurable outputs

$\gamma \equiv$ parameter vector belonging to some set Γ which determines the portion of the set $v(t)$ -- items (a), (b), and (c) above -- to be impressed upon the system

$S_\gamma \equiv$ set of $v(t)$ which is generated for the particular γ

$P(\gamma) \equiv$ performance criterion which can take on a range of values for a given γ

$W \equiv$ set to which it is desired to restrict $P(\gamma)$

In the above terminology, we define a criterion of acceptability in the following manner. If $P(\gamma)$ is maintained in the set W, then we satisfy the criterion of acceptability. This notion leads to a definition.

Definition 1 -- A system, $\mathcal{A}$, is an acceptably performing system (APS) with respect to S_γ and W if it satisfies the criterion of acceptability with every source in the family S_γ, γ belonging to Γ.

To elaborate, we have an APS if it is possible to design a mechanism in $\mathcal{A}$ which can provide a control to maintain the performance criterion within acceptable limits as given by W. This acceptable performance is to be maintained for a class of inputs as represented by S_γ.

Even open-loop systems can be APS as long as the criterion of acceptability is maintained. The problem arises, however, when $P(\gamma)$ cannot be maintained in W. Here, it becomes necessary to consider more complex mechanisms within $\mathcal{A}$ to satisfy the criterion of acceptability. Therefore, one is led to many possible alternatives for the construction of the control mechanism, each with an attempt to satisfy the criterion of acceptability.

To illustrate the above notions let us, for example, describe the pitch response of an aircraft attitude control system. Given a command or reference input, signals are sent to the controlling surfaces which, in turn, deflect and, through interaction with the air stream, create a torque on the aircraft. As an example, the performance criterion, $P(\gamma)$ can be selected as the percent overshoot to a step command. The γ could be the altitude. For a particular altitude, γ, a set or ensemble of air-density variations, can be experienced. This set corresponds to part of the $v(t)$ just described. For the set of air-density variations a range of $P(\gamma)$ can be experienced, say from 0% to 30%. If this range of $P(\gamma)$ is within the allowable set as given by W then we have an APS.

In a given application it may be difficult to tie down Γ, W, and P. However, the definition gives us a starting point from which we can describe various mechanizations which have as an intended goal the maintenance of some performance criterion within prescribed limits.

Now, we are prepared to describe or define various forms of mechanization of $\mathcal{Q}$. It is in this connection that we can describe what we mean by an adaptive control system.[†] Of course, open-loop systems and feedback control systems are familiar descriptions of control mechanizations. In contrast to these forms we desire the distinctive characteristics of adaptive control systems. Let us present a definition. Along with the definition for adaptive control systems we give definitions for the other two forms in order to point out the distinctive characteristics.

Definition 2 -- A system is an open-loop system if control action as a function of time is impressed upon the process on the basis of a priori knowledge of the process.

Definition 3 -- A system is a feedback control system if a means is provided to monitor the variables depending upon the control action (state or controlled variables) in order to accordingly modify the subsequent control action in an attempt to be an acceptably performing system.

Definition 4 -- A system is an adaptive control system if a means is provided to monitor, in addition to the state variables, its performance and/or process (internal and/or external) characteristics in order to accordingly modify the control action in an attempt to be an acceptably performing system.

Admittedly, Definition 3 is influenced by other definitions given in the past, perhaps most strongly by that given by Cooper and Gibson (Reference 7). An important point, however, is to make a fairly general definition so that it encompasses the many adaptive systems described in the literature while at the same time making a distinction from the other two forms of mechanization. To elaborate, monitoring performance and/or process characteristics makes adaptive control systems different from feedback control systems. It should be noted that adaptive control systems are feedback control systems but the converse is not necessarily the case; therefore, it is expected that better performance can be achieved by adaptive control systems. And it is for this reason that we study adaptive control systems. It is noted, however, that even an adaptive control system may not be an APS.

From the definition, it is observed that commonly described controllers which make modifications depending upon environmental measurements (e.g., air-data measurements) fall into the class of adaptive control systems.

[†] Zadeh (Reference 6) used the term adaptive for the external manifestation while we choose to use adaptive for the internal mechanization.

Adaptive control systems have appeared in many forms. No attempt will be made in this section to survey all the different schemes devised in the past because several good survey articles are available (References 1, 7, 17, 22, 24). However, three categories which appear to encompass a large proportion of adaptive control systems are (1) the high-gain schemes, (2) the model-referenced schemes, and (3) optimum-adaptive schemes. The order of the listing is in the direction of increased complexity. It is expected that the range of performance will vary with the different schemes, and complexity should be added only if improved performance is obtainable and mandatory. Presently, the selection of a particular scheme appears to depend on loosely defined qualitative judgment and constitutes the "art" of engineering.

1.3 Background

Although the definition given above was stated recently, engineers in the past knew intuitively what was desired, i.e., achieve acceptable control in the presence of large variations in the process. Even before the term adaptive was attached to control systems, engineers used, for example, air data measurements to vary the controller. This situation can certainly be adaptive by the definition given above. No attempt will be made in this section to survey all the different schemes devised in the past because several good survey materials are available (References 7, 8, 9). It will be more the intent to delineate the three categories into which the different schemes seem to fall. These are 1) the high-gain scheme, 2) the model-referenced scheme, and 3) the optimum-adaptive scheme.

From the practical standpoint, the high-gain scheme, first proposed by Minneapolis-Honeywell Company (Reference 10), has been widely discussed and tested. It has been proven to be of wide applicability. The gain in the feedback loop around the changing process is kept as high as possible in order that the input-output transference is close to unity. Because stability problems arise at high gain, the signal in the loop is monitored to check for oscillations. With this information the loop gain is adjusted to keep the system on the verge of instability. A response close to that of a particular model is obtained regardless of the process parameters by placing a model in front of the feedback loop. A schematic diagram of the high-gain scheme is shown in Figure 2. One of the objections to this approach is that the designer must have considerable a priori information about the process, i.e., he must know the general vicinity where the roots of the system go into the right half plane. Of course, a frequency insensitive unity gain can only be approached implying that the output response will differ to some extent from the model response. Also, small oscillations are always present in the loop. (This oscillation has been reported to be unobjectionable in aerospace applications.) A variation of the same philosophy has been recently given by Horton (Reference 11).

If one is willing to accept more complexity, the model-referenced scheme can provide better response. This scheme has been tested successfully in experimental flight tests by a group at MIT (Reference 12). Stability

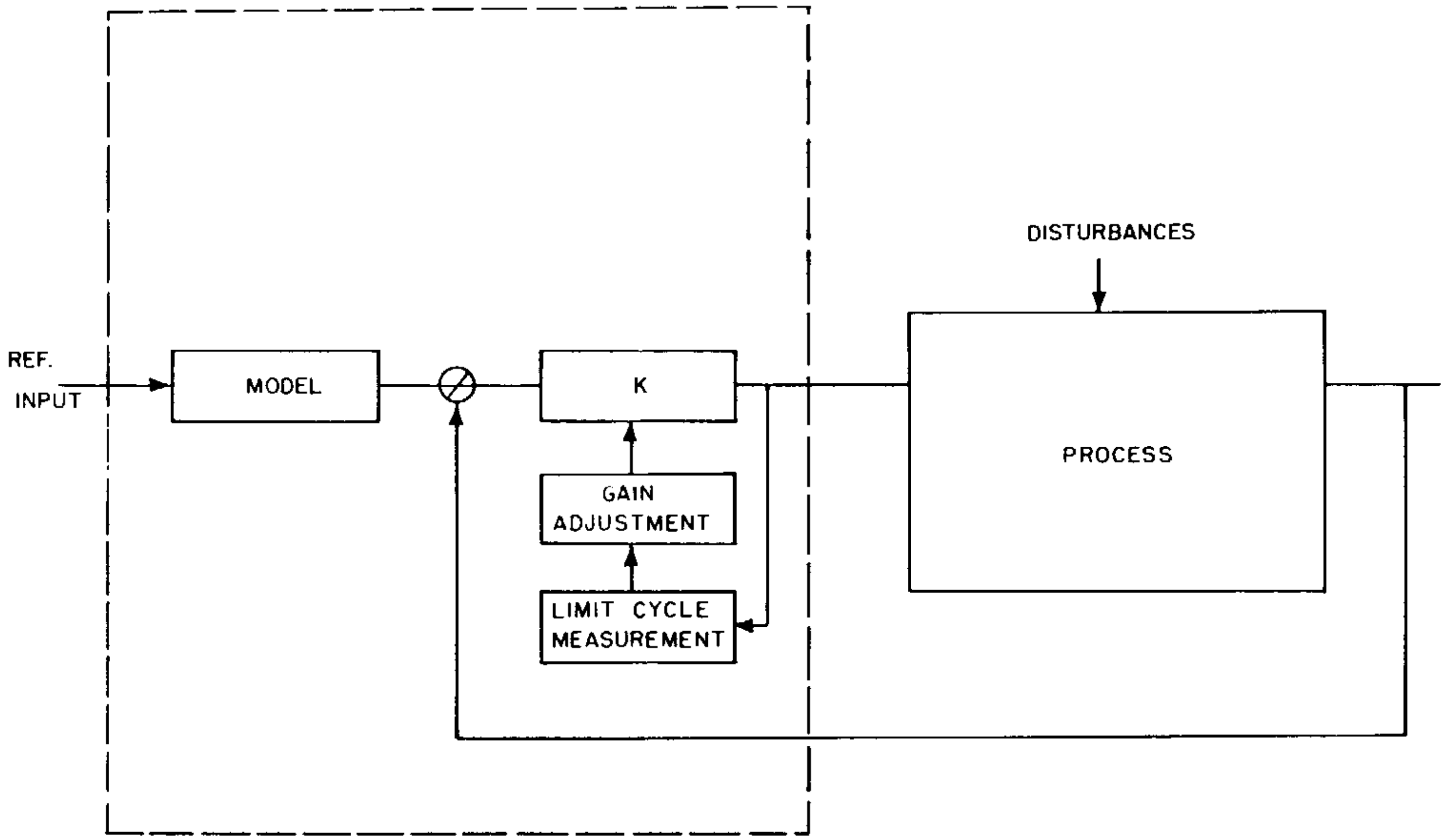


Figure 2. High Gain Scheme

problems associated with this type of scheme have been performed by Donalson (Reference 13). A schematic diagram of this method is shown in Figure 3. The method simply adjusts the controller parameters so that the process output is kept close to the model output. The stability problem ensues in the parameter adjustment loop. With this scheme unstable processes and nonlinear processes can be handled. The method, however, requires a good knowledge of the form of the process.

As the state of the computer art advances, one asks if there are still better methods which can improve upon the accuracy of the system. In regards to this, optimum-adaptive schemes are investigated. This area is still primarily in the exploratory stage with no applications reported. Experimental-verification has been made to a limited extent via analog and digital simulation. Some of the contributors in this area are Kalman (Reference 1), Merriam (Reference 2), Braun (Reference 3), Meditch (Reference 4), and Hsieh (Reference 5). A schematic diagram for this scheme is given in Figure 4. Basically the technique solves an optimization problem on the assumption that the process and the states are known. Since the process state and parameters are unknown to some extent in an adaptive task, both state estimation and process identification must be performed.

The identification problem has been investigated by many investigators independent of the adaptation scheme. Some background material on identification will be given in Chapters 4 and 5.

1.4 Objectives of the Study

The major objective is to investigate unexplored areas of the optimum-adaptive scheme to adaptation. We will look into both the area of synthesis of optimum controls and the area of identification. Application will then be sought in the area of re-entry of aerospace vehicles.

An extreme amount of background material is available for the optimum control problem. In fact, several alternative approaches are available. These are 1) maximum principle, 2) dynamic programming, 3) functional analysis, and 4) steepest descent methods. The on-line computation of optimal controls, however, is not in a satisfactory state of affairs except possibly for the quadratic criterion-linear process case. The previous investigators for the most part have remained in this latter case. In our investigation we impose an added constraint of bounds on the control force. We will also stay, however, in the quadratic criterion-linear process case. The nonlinear (quadratic) programming approach is used as it seems to be the most suitable method when we have this additional constraint. In our problem we postulate a digital computer to compute the control forces. This postulation reduces the problem to the discrete case.

In the area of identification, we will investigate two principal areas. First, the statistical aspects of the estimated parameters will be studied. Here, we study the concept of confidence interval primarily for the case

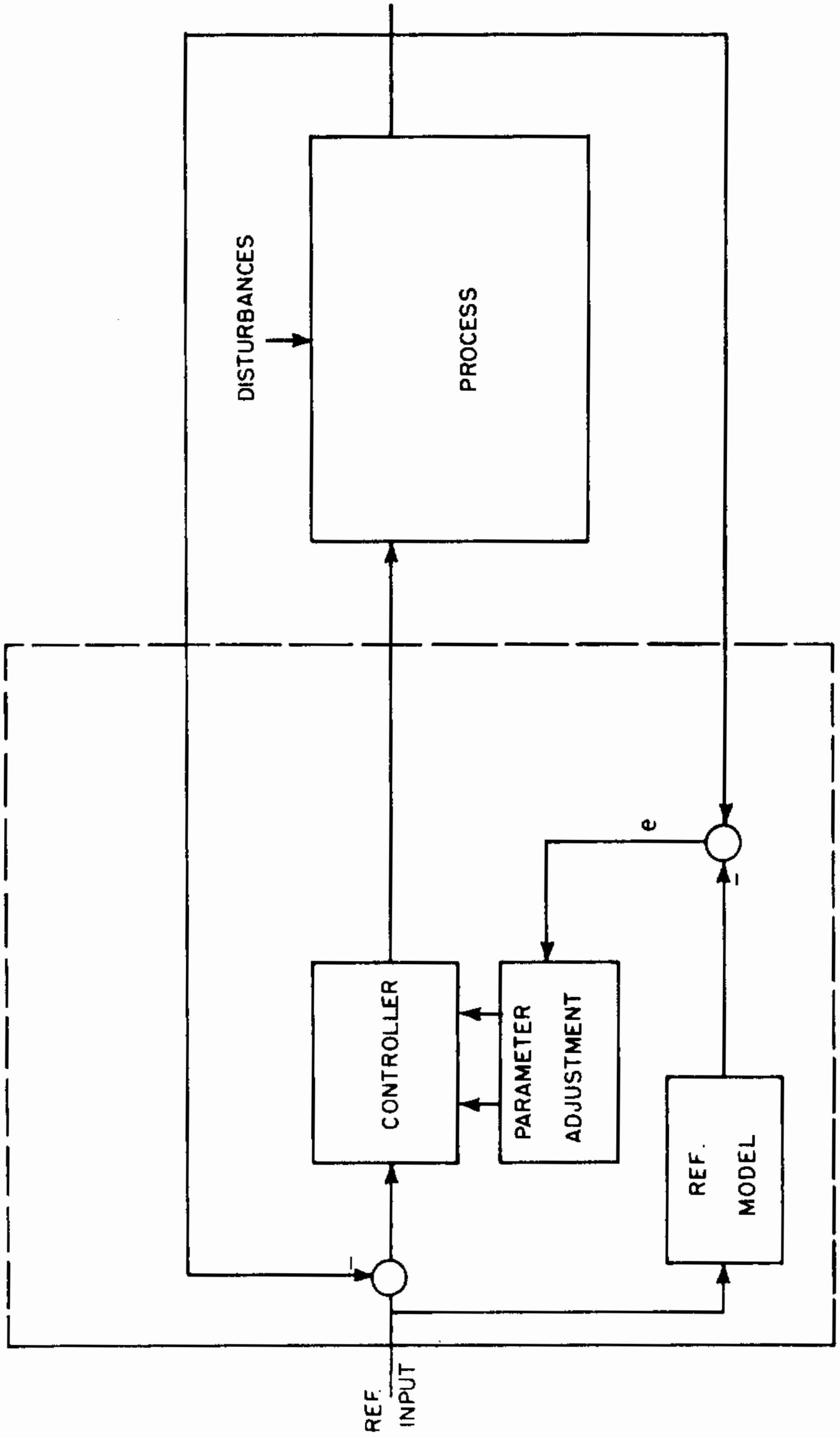


Figure 3. Model Referenced Scheme

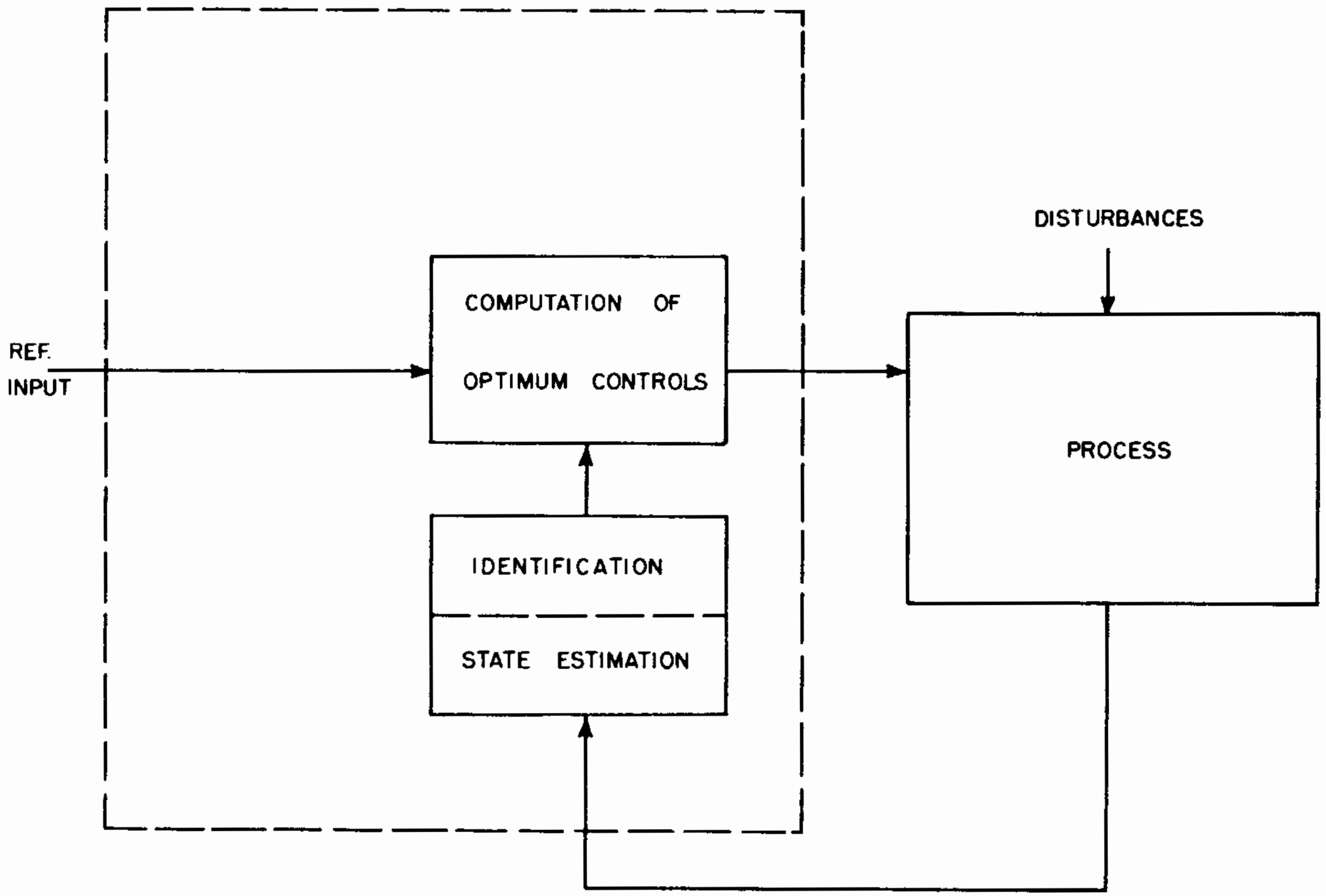


Figure 4. Optimum-Adaptive Scheme

with unknown variances. Secondly, we re-examine the learning model approach of Margolis (Reference 14) from another viewpoint; i.e., we will study an integral error-square criterion previously unexplored by Margolis. A modified Newton's procedure will be employed.

Generally, the identification problem is coupled with the state estimation problem. Identification of process parameters can be made if the states are known, or estimation of the states can be made if the process is known. We will attack this problem by investigating identification methods which depend only on partial knowledge of the states. Then, estimation of the states will be made with the identified parameters.

1.5 Organization of Part 1

This report is organized into eight chapters with five appendices. This first chapter provides introduction to the subject via definitions, background materials, and objectives.

Chapter 2 gives algorithms for on-line discrete control of linear processes with a quadratic criterion without inequality constraints on the control force. This is mainly review material and is included primarily for setting the stage for Chapter 3.

Chapter 3 gives algorithms for on-line discrete control of linear processes with a quadratic criterion with inequality constraints on the control force. Here, quadratic programming methods will be applied and suitability of on-line computation will be verified by experimentation through digital simulation.

Chapter 4 explores the statistical aspects of the explicit mathematical relation method of identification. Also, the recursive method of Greville (Reference 15) and Kalman (Reference 16) will be applied to identification.

Chapter 5 explores the learning model approach with an integral error-square criterion. The application of Newton's method to the learning model approach will be verified through experimentation via digital simulation.

Chapter 6 gives a method for state variable estimation. This is again primarily review material but it is an integral part of the overall adaptive system.

Chapter 7 explores possible application areas for the proposed method of adaptation. The area of re-entry of aerospace vehicles is chosen.

Chapter 8 concludes Part 1 by suggestions for future studies.

Appendix 1 describes the notation used in the control system and it states concisely the problems attacked in Part 1.

Appendix 2 gives a brute force method to solve the quadratic programming problem of Chapter 3. Although the method is cumbersome, it is included because it gives added insight into the problem.

Appendix 3 reviews the pertinent quadratic programming theorems. Several routines described in Chapter 3 will draw heavily from these theorems based on the Kuhn and Tucker (Reference 17) theorems.

In Appendix 4 the recursive method of Greville is adopted for the identification problem. The algorithms are re-derived from the postulates given by Penrose (References 18, 19).

Appendix 5 gives the correspondence between Greville and Kalman's recursive procedures.

CHAPTER 2

OPTIMUM LINEAR DISCRETE CONTROL

2.1 Introduction

This chapter gives algorithms for optimum linear discrete controls with a quadratic performance criterion. No inequality constraints will be considered here. This is mainly review material and is included primarily to set the stage for the next chapter. As previously stated, we confine ourselves to the discrete control case as we postulate a digital computer to perform the synthesis.

This chapter first gives a philosophical basis for our adaptive control before proceeding to give algorithms.

2.2 General Philosophy

We envision using the optimum-adaptive control to keep the process output close to some desired trajectory. This operation is to be maintained over some time interval which we will designate as the operation interval. In other words, we desire to minimize the performance criterion

$$P = \sum_{k=0}^{N_1} ||\underline{y}_d(k) - \underline{y}(k)||_Q^2 \qquad (1)$$

where $\underline{y}_d(k)$ - desired trajectory

$\underline{y}(k)$ - actual trajectory

N_1 - number of sampling intervals in operation interval

$k=0$ - beginning of operation interval

Q - a non-negative weighting matrix

The desired trajectory will be assumed known throughout the operation interval. A controller designed to minimize p is termed a follower and an example will be given in Chapter 7.

The optimization of (1) is not practical primarily for three reasons. First of all, open-loop control ensues and it is more desirable to recompute periodically the optimal control. Secondly, the process is uncertain for time into the future. Thirdly, the on-line numerical computation required may be too large. As a result, it is more practical to perform periodically the following optimization. We choose a fixed time interval into the future from the present time, designated optimization interval, and perform a minimization over this interval. Therefore, instead of (1) we minimize periodically

$$J = \sum_{j=k+1}^{k+N} ||\underline{y}_d(j) - y(j)||_Q^2 \qquad (2)$$

where N - number of sampling intervals in the optimization interval

k - present time

The time relation of the intervals under consideration is given in Figure 5.

The idea of adaptive controls originated from a desire to emulate the desirable human characteristics. Therefore, as the general philosophy, we give a human analogy discussion. A similar discussion was first presented by Merriam (Reference 2).

A human faced with a control problem, such as driving an automobile, has the problem of selecting optimally the next decision in a multi-stage decision process. This decision will be based on the present state and the knowledge (maybe intuitive) of the process response (automobile behavior). A human will decide on a particular control on the basis of considerations given over a relatively short time into the future. For example, the road conditions may change and the human will not apply the same control on a rough road as on an icy road. With knowledge of the desired path over a short time into the future (optimization interval) and the knowledge of the vehicle response, the human can apply proper control effort on the steering wheel. The criterion given by (2) will then replace the subjective evaluation performed by a human.

Repeating ourselves to some extent, although (2) may lead to sub-optimal policies it may be the only proper criterion to apply in any given circumstance. Inherent in the above discussion were state estimation and process identification. These functions are performed by the human through observation and testing vehicle response. As a human could adapt to different vehicles (different responses) and also changes in the same vehicle (road conditions, tire blow-out, etc.), an adaptive control must be able to perform these tasks if it is to have the finer human capabilities.

2.3 Control of Continuous Linear Process by Digital Computer

We will attempt to control processes which are describable by linear ordinary differential equations. We immediately make the following assumption.

Assumption: Changes occurring in the process during an optimization interval will be assumed to be small.

This allows us to use constant coefficient differential equations which, in turn, will relieve the computational requirements. With more complexity, considerations can be carried over to the variable coefficient case.

The process is then described by

$$\underline{\dot{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \tag{3}$$

where A - n x n matrix

B - n x r matrix

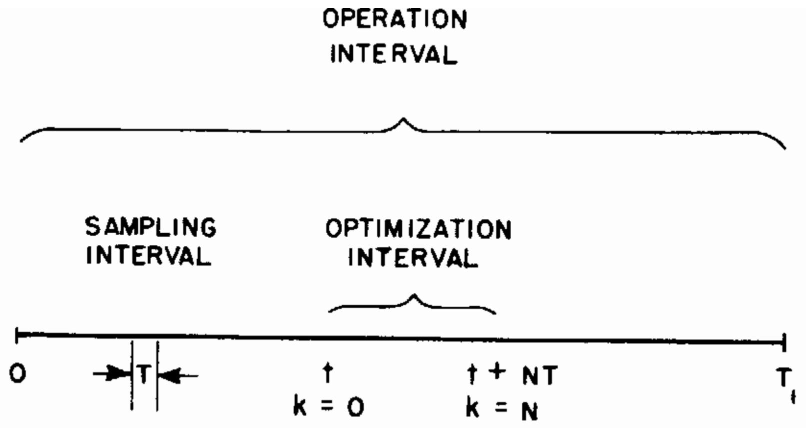


Figure 5. Time Relation of Intervals Under Consideration

The solution of (3) is given by

$$\underline{x}(t) = X(t)\left[\underline{x}(o) + \int_o^t X^{-1}(\tau)\, B\, \underline{u}(\tau)\, d\tau\right] \tag{4}$$

where $X(t)$ is the matrix solution of

$$\dot{X}(t) = AX(t) \qquad \text{with } X(0) = I \text{ (identity matrix)}$$

When digital computers are employed as controllers, the control signal will have the appearance of a staircase signal shown in Figure 6. Mathematically, it is formed by a sample-hold combination. In mathematical notation,

$$\underline{u}(\tau) = \underline{u}(k) \qquad (k-1)T \le \tau < kT \qquad k = 1, 2, \ldots N \tag{5}$$

For this staircase situation (4) can be solved at discrete instants of time.

$$\underline{x}(k) = X(k)\left[\underline{x}(o) + \int_o^{kT} X^{-1}(\tau)\, B\, \underline{u}(\tau)\, d\tau\right]$$

$$= X(k)\left[\underline{x}(0) + \sum_{i=1}^{k} \int_{(i-1)T}^{iT} X^{-1}(\tau)\, B\, \underline{u}(\tau)\, d\tau\right] \tag{6}$$

Since

$$\underline{x}(k-1) = X(k-1)\left[\underline{x}(o) + \sum_{i=1}^{k-1} \int_{(i-1)T}^{iT} X^{-1}(\tau)\, B\, \underline{u}(\tau)\, d\tau\right]$$

we can write $\underline{x}(k)$ in terms of $\underline{x}(k-1)$.

$$\underline{x}(k) = X(k)\, X(k-1)^{-1}\, \underline{x}(k-1) + X(k)\int_{(k-1)T}^{kT} X^{-1}(\tau)\, B\, d\tau\, \underline{u}(k)$$

Let

$$\Phi = X(k)\, X(k-1)^{-1}$$

$$\Gamma = X(k)\int_{(k-1)T}^{kT} X^{-1}(\tau)\, B\, d\tau$$

We note that $X(k)$ is the solution of

$$X(k) = \Phi\, X(k-1) \qquad \text{with } X(0) = I$$

Therefore,

$$\underline{x}(k) = \Phi\, \underline{x}(k-1) + \Gamma\, \underline{u}(k) \tag{7}$$

In terms of Φ and Γ, (6) becomes

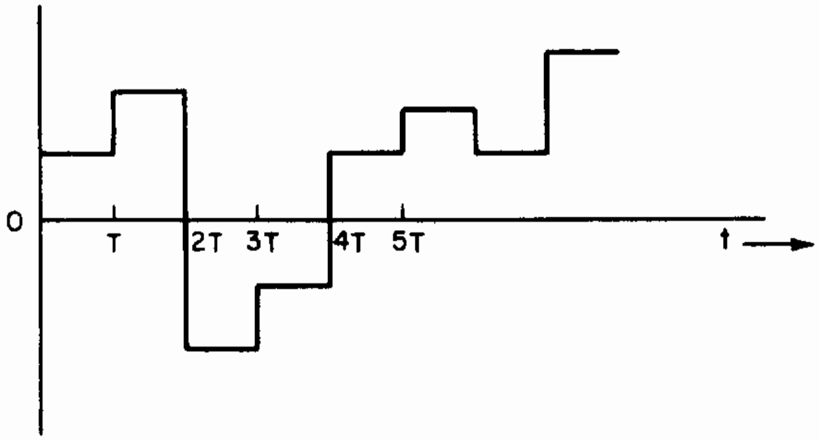


Figure 6. Staircase Signal

$$\underline{x}(k) = \Phi^k \underline{x}(o) + \sum_{i=1}^{k} \Phi^{k-i} \Gamma \underline{u}(i) \tag{8}$$

It may not be possible to measure all the state variables. The measured output, $\underline{y}(k)$ is usually some linear combination of the state variables.

$$\underline{y}(k) = H \underline{x}(k) \tag{9}$$

The basic deterministic model is shown in Figure 7.

To the deterministic model we can add stochastic disturbances: 1) load disturbances and 2) measurement errors. The distinction should be carefully noted. Load disturbances generally cause the state variables to become stochastic, and these can be incorporated into the deterministic model by including in addition to the control forces, $\underline{u}(k)$, other inputs, $\underline{w}(k)$, which are white noise. Measurement error can, without too much loss of generality, be considered as additive white noise on the output variable. The model with stochastic disturbances is shown in Figure 8.

In the discussion on optimization, it is desired to restrict the amplitude of the control force. This is accomplished in this chapter indirectly by adding terms to the performance criterion, (2).

$$J = \sum_{j=k+1}^{k+N} \left|\left|\underline{y}_d(j) - \underline{y}(j)\right|\right|_Q^2 + \left|\left|\underline{u}(j)\right|\right|_R^2 \tag{10}$$

where R is a non-negative weighting matrix.

2.4 Discussion of the Maximum Principle and the Calculus of Variations Approach

The maximum principle and the calculus of variations approach can be applied to the discrete version of the linear-process, quadratic-criterion case. Chang (Reference 20) and Katz (Reference 21) investigated the maximum principle for the discrete case giving necessary conditions. As the calculus of variations approach yields the same algorithm, this latter point of view will be discussed in this section. This approach was taken by Kipiniak .(Reference 22). It will be observed that this approach leads to a feedback control law; i.e., the control is given as a function of the state variables.

It should be pointed out that although only necessary conditions are satisfied, the solution to the necessary condition should be necessary and sufficient. That is, if we know the existence and uniqueness of the minimum to the problem and if only a unique solution is provided by the necessary condition, that solution is the minimum. From the arguments given in Appendix 3 we can show existence and uniqueness of the minimum. It is noted that the infinite domain is a convex set.

For the linear process

$$\underline{x}(j) = \Phi \underline{x}(j-1) + \Gamma \underline{u}(j) \tag{7}$$

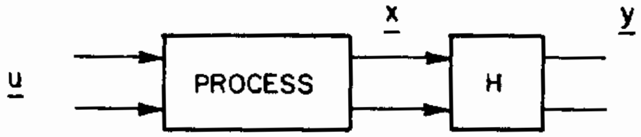


Figure 7. Deterministic Model

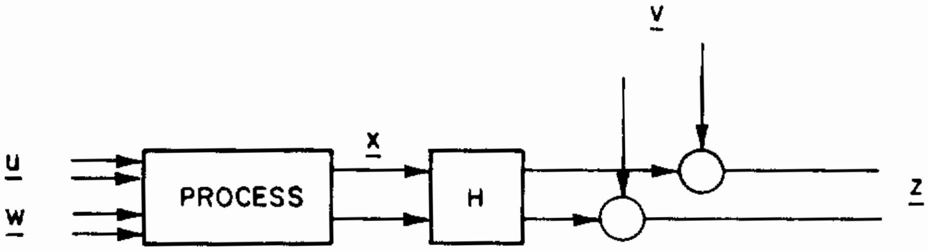


Figure 8. Model with Stochastic Disturbances

with $\underline{x}(k) = \underline{x}^o$, and the criterion

$$J = \sum_{j=k+1}^{k+N} \frac{1}{2} \|\underline{x}(j)\|_Q^2 + \frac{1}{2} \|\underline{u}(j)\|_R^2 \qquad (11)^\dagger$$

Although (11) differs from (10), the derivation follows the same lines. Equation (11) is applicable directly to the regulator problem which is important in itself. Using Lagrange multipliers, the constrained functional to be minimized becomes

$$J_1 = \sum_{j=k+1}^{k+N} \frac{1}{2} \|\underline{x}(j)\|_Q^2 + \frac{1}{2} \|\underline{u}(j)\|_R^2$$
$$+ < \underline{p}(j), \underline{x}(j) - \Phi\, \underline{x}(j-1) - \Gamma\, \underline{u}(j) >$$

The necessary condition states that the total differential of J_1 vanishes for independent differentials of $\underline{x}(j)$, $\underline{u}(j)$, and $\underline{p}(j)$. Taking the differential, we get

$$dJ_1 = \sum_{j=k+1}^{k+N-1} d\, \underline{x}(j)^* \left\{ Q\, \underline{x}(j) + \underline{p}(j) - \Phi^*\, \underline{p}(j+1) \right\}$$
$$+ d\, \underline{x}(k+N)^* \left\{ Q\, \underline{x}(k+N) - \underline{p}(k+N) \right\}$$
$$+ \sum_{j=k+1}^{k+N} d\, \underline{u}(j)^* \left\{ R\, \underline{u}(j) - \Gamma^*\, \underline{p}(j) \right\}$$
$$+ d\, \underline{p}(j)^* \left\{ \underline{x}(j) - \Phi\, \underline{x}(j-1) - \Gamma\, \underline{u}(j) \right\} = 0$$

Therefore, the following relations must be satisfied.

$$\underline{x}(j) = \Phi\, \underline{x}(j-1) + \Gamma\, \underline{u}(j) \qquad (7)$$

$$\underline{p}(j) = (\Phi^*)^{-1}\, \underline{p}(j-1) + (\Phi^*)^{-1}\, Q\, \underline{x}(j-1) \qquad (12)^\ddagger$$

$$\underline{u}(j) = R^{-1}\, \Gamma^*\, \underline{p}(j) \qquad (13)$$

with transversality condition

$$\underline{p}(k+N) = -\, Q\, \underline{x}(k+N) \qquad (14)$$

or,

[†] The use of x instead of y implies H = I. If the criterion does not contain every state, x_i, then Q can appropriately be chosen with zero elements.

[‡] It is noted that $(\Phi^*)^{-1}$ exists since Φ is a fundamental matrix.

$$\underline{x}(j) = (\Phi \Gamma R^{-1} \Gamma^* \Phi^{*-1} Q + \Phi)\, \underline{x}(j-1) + \Gamma R^{-1} \Gamma^* \Phi^{*-1} \underline{p}(j-1)$$

$$\underline{p}(j) = \Phi^{*-1} Q\, \underline{x}(j-1) + \Phi^{*-1} \underline{p}(j-1) \qquad (15)$$

with

$$\underline{x}(k) = \underline{x}^o, \qquad \underline{p}(k+N) = -Q\, \underline{x}(k+N)$$

Or,

$$\begin{bmatrix} \underline{x}(j) \\ \underline{p}(j) \end{bmatrix} = \theta \begin{bmatrix} \underline{x}(j-1) \\ \underline{p}(j-1) \end{bmatrix} \qquad (16)$$

where

$$\theta = \begin{bmatrix} \Phi \Gamma R^{-1} \Gamma^* \Phi^{*-1} Q + \Phi & \Gamma R^{-1} \Gamma^* \Phi^{*-1} \\ \Phi^{*-1} Q & \Phi^{*-1} \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}$$

Thus,

$$\begin{bmatrix} \underline{x}(k+N) \\ \underline{p}(k+N) \end{bmatrix} = \Psi \begin{bmatrix} \underline{x}(k) \\ \underline{p}(k) \end{bmatrix} \qquad (17)$$

where

$$\Psi = \theta^N \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix}$$

Since $\underline{p}(k+N) = -Q\, \underline{x}(k+N)$, we can eliminate $\underline{p}(k+N)$ and $\underline{x}(k+N)$ from (17). Thus

$$\underline{p}(k) = -\left(\Psi_{22} + Q\Psi_{12}\right)^{-1} \left(\Psi_{21} + Q\,\Psi_{11}\right) \underline{x}(k) \qquad (18)$$

and

$$\underline{p}(k+1) = \left[\theta_{21} - \theta_{22}\left(\Psi_{22} + Q\,\Psi_{12}\right)^{-1}\left(\Psi_{21} + Q\,\Psi_{11}\right)\right] \underline{x}(k)$$

Thus, the feedback solution is given by (13). The inverse here is assumed to exist. It seems that Q should be chosen so that the inverse exists even for large parameter variations. It is noted that existence is required if the problem is to be a necessary and sufficient condition.

$$\underline{u}(k+1) = -\Lambda\, \underline{x}(k) \qquad (19)$$

where

$$\Lambda = -R^{-1}\Gamma^* \left(\theta_{21} - \theta_{22}\left(\Psi_{22} + Q\,\Psi_{12}\right)^{-1}\left(\Psi_{21} + Q\,\Psi_{11}\right)\right)$$

If the process does not change, Λ will be a constant and the feedback problem is easy. In an adaptive task, the Γ, Ψ_{ij}, and θ_{ij} must be updated as Φ and Γ change.

2.5 Dynamic Programming Approach

The derivation and algorithm given in this section are due to Kalman (Reference 16). Again, necessary conditions are used to arrive at the solution. As before if a unique solution is provided the solution is necessary and sufficient.

We start with the process

$$\underline{x}(j) = \Phi\, \underline{x}(j-1) + \Gamma\, \underline{u}(j) \qquad (7)$$

with $\underline{x}(o) = \underline{x}^{o}$, and the criterion

$$J_{\underline{N}} = \sum_{j=1}^{N} \frac{1}{2}\, \|\underline{x}(j)\|_{Q}^{2} + \frac{1}{2}\, \|\underline{u}(j)\|_{R}^{2} \qquad (11)$$[†]

Let

$$f_{\underline{N}} = \underset{\underline{u}(1)}{\text{Min}}\; J_{\underline{N}}$$ (bar underneath the time index N indicates time-to-go)

or,

$$f_{\underline{N}} = \underset{\underline{u}(1)}{\text{Min}} \left\{ J_{\underline{1}}\left(\underline{x}(o),\ \underline{u}(1)\right) + f_{\underline{N-1}}\left(\underline{x}(1)\right) \right\} \qquad (20)$$

$$f_{\underline{1}} = \underset{\underline{u}(N)}{\text{Min}} \left\{ J_{\underline{1}}\left(\underline{x}(N-1),\ \underline{u}(N)\right) \right\}$$

For the problem we are considering, it can be shown by induction that $f_{\underline{N}}$ is a quadratic form, or

$$f_{\underline{m}} = \underline{x}^{*}(j)\, M(\underline{m})\, \underline{x}(j) \qquad (21)$$

where

$\underline{m} + j = N$

$\underline{m}$ - time-to-go

j - running time

It is noted that

$$f_{\underline{0}} = 0$$

and therefore,

$$M(\underline{0}) = 0.$$

[†] To simplify the notation in this section and Section 2.6 time index k has been dropped. ($j=1 \Rightarrow j=k+1$, $j=N \Rightarrow j=k+N$).

Also,

$$J_{\underline{1}} = ||\underline{x}(1)||^2_Q + ||\underline{u}(1)||^2_R \tag{22}$$

Upon substituting (21) and (22) into (20),

$$f_{\underline{N}} = \underset{\underline{u}(1)}{\text{Min}} \left\{ ||\underline{x}(1)||^2_Q + ||\underline{u}(1)||^2_R + ||\underline{x}(1)||^2_{M(N-1)} \right\}$$

Since $\underline{x}(1)$ is a function of $\underline{u}(1)$ and $\underline{x}(0)$,

$$f_{\underline{N}} = \underset{\underline{u}(1)}{\text{Min}} \left\{ ||\,\Phi\underline{x}(0) + \Gamma\,\underline{u}(1)||^2_{Q+M(N-1)} + ||\underline{u}(1)||^2_R \right\}$$

Or

$$f_{\underline{N}} = \underset{\underline{u}(1)}{\text{Min}} \left\{ ||\underline{x}(0)||^2_{\Phi^*\left(Q+M(\underline{N-1})\right)\Phi} + ||\underline{u}(1)||^2_{\Gamma^*\left(Q+M(N-1)\right)\Gamma+R} + 2\,\underline{u}^*(1)\,\Gamma^*\left(Q+M(\underline{N-1})\right)\Phi\,\underline{x}(0) \right\} \tag{23}$$

Differentiating the quantity in the bracket with respect to $\underline{u}(1)$ and setting the derivative equal to zero, we get[†]

$$\underline{u}(1) = -\left(\Gamma^*\left(Q+M(\underline{N-1})\right)\Gamma+R\right)^{-1}\Gamma^*\left(Q+M(\underline{N-1})\right)\Phi\,\underline{x}(0)$$

or, the feedback solution is given by (equivalent to Equation 19)

$$\underline{u}(1) = -\left(\Gamma^*\,P(\underline{N-1})\,\Gamma + R\right)^{-1}\Gamma^*\,P(\underline{N-1})\;\Phi\,\underline{x}(0) \tag{24}$$

where

$$P(\underline{N-1}) = Q + M(\underline{N-1})$$

A recursive relation can be derived for $P(\underline{N-1})$. We note that

$$f_{\underline{N}} = ||\underline{x}(0)||^2_{M(\underline{N})} = ||\underline{x}(0)||^2_{P(\underline{N})-Q}$$

Upon substituting (24) into (23), we also have

[†] It is easily seen that the inverse here exists since the first term in the parentheses is positive semi-definite and the second term is positive definite.

$$f_{\underline{N}} = \left\|\underline{x}(0)\right\|^2_{\Phi^* P(\underline{N-1})\Phi}$$

$$+ \left\|\underline{x}(0)\right\|^2_{\Phi^* P(\underline{N-1})\,\Gamma\left(\Gamma^* P(\underline{N-1})\Gamma + R\right)^{-1} \Gamma^* P(\underline{N-1})\Phi}$$

$$+ \left\|\underline{x}(0)\right\|^2_{-2\Phi^* P(\underline{N-1})\,\Gamma\left(\Gamma^* P(\underline{N-1})\,\Gamma+R\right)^{-1} \Gamma^* P(\underline{N-1})\Phi}$$

Therefore,

$$P(\underline{N}) = \Phi^* \left\{ P(\underline{N-1}) - P(\underline{N-1})\,\Gamma\,\left(\Gamma^* P(\underline{N-1})\,\Gamma + R\right)^{-1} \Gamma^* P(\underline{N-1}) \right\} \Phi$$
$$+ Q \qquad (25)$$

with

$$P(\underline{0}) = 0.$$

This is a nonlinear Ricatti equation. Equations (25) and (24) give the optimal control force. At each sampling interval these equations are re-used. The quantities Φ and Γ can be changed as new information is available. Whether the algorithm given in this section is better than that given in the previous section is debatable. The two may well be computationally equivalent. One difference which is evident is that for the second algorithm we are assured of a unique solution. In the first algorithm an inverse was assumed to exist.

2.6 An Extension to the Stochastic Case

With stochastic disturbances the algorithms derived for the deterministic case can still be applied if a particular criterion function is chosen. This situation was first shown for the white noise case by Joseph and Tou (Reference 23). Extensions to the more general case were given by Gunckel and Franklin (Reference 24), Florentin (Reference 25), and Schultz (Reference 26). Apparently, this situation was known previously to statisticians under the name "Uncertainty Equivalence Principle". A result of their studies is presented in this section.

The stochastic model is given by

$$\underline{x}(k) = \Phi\, \underline{x}(k-1) + \Gamma\, \underline{u}(k) + \Xi\, \underline{w}(k)$$

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

where $\underline{w}(k)$ and $\underline{v}(k)$ are sequences of independent Gaussian noise. We choose the following performance criterion.

$$J = E\left\{ \sum_{j=1}^{N} \left\|\underline{x}(j)\right\|^2_Q + \left\|\underline{u}(j)\right\|^2_R \right\}$$

The optimal control is then given by

$$\underline{u}(1) = -\left(\Gamma^* P(\underline{N-1})\,\Gamma + R\right)^{-1} \Gamma^* P(\underline{N-1})\;\Phi\,\hat{\underline{x}}(0)$$

$$P(\underline{N}) = \Phi^*\left\{ P(\underline{N-1}) - P(\underline{N-1})\,\Gamma\left(\Gamma^* P(\underline{N-1})\,\Gamma + R\right)^{-1} \Gamma^* P(\underline{N-1})\right\}\Phi + Q$$

with $P(\underline{0}) = 0$. The equations are exactly the same except $\underline{x}(0)$ is replaced by the best least squares estimate, $\hat{\underline{x}}(0)$.

Of course, the results in this section do not reflect changes in Φ and Γ which can occur in an adaptive problem. At least, the above results give assurance that proper action is being taken in a stationary situation.

2.7 Stability of the Closed-Loop System

There may be some question whether the implementation of the optimal on-line controller in a closed-loop manner gives a stable system. For the case discussed in this chapter we can give sufficient conditions for stability. We employ the discrete version of Lyapunov's direct method. Let us state first Lyapunov's theorem (Reference 27):

Stability Theorem: If for the process

$$\underline{x}(k+1) = \underline{f}\left(\underline{x}(k)\right)$$

there exists a scalar function of the state variables, $V\left(\underline{x}(k)\right)$, such that $V(0) = 0$, and

i) $V(\underline{x}) > 0$ when $x \neq 0$

ii) $V\left(\underline{x}(k+1)\right) < V\left(\underline{x}(k)\right)$ for $k > K$, K finite

iii) $V(\underline{x})$ is continuous in $\underline{x}$

iv) $V(\underline{x}) \to \infty$ when $\underline{x} \to \infty$,

then the equilibrium solution $\underline{x} = 0$ is globally stable and $V(\underline{x})$ is a Lyapunov function for the system.

For the application of this theorem, let us choose the following criterion function.

$$V\left(\underline{x}(k)\right) = \left|\left|\underline{x}(k)\right|\right|^2 \tag{26}$$

The problem is to determine $\underline{x}(k+1)$ when the optimal controller is used. Let us consider the formulation of Section 2.4. From (17) we have

$$\underline{x}(k) = \psi_{11}\,\underline{x}(k+N) - \psi_{12}\,Q\,\underline{x}(k+N)$$

$$\underline{p}(k) = \psi_{21}\,\underline{x}(k+N) - \psi_{22}\,Q\,\underline{x}(k+N)$$

Eliminating $\underline{x}(k+N)$, we have

$$\underline{p}(k) = \left(\psi_{21} - \psi_{22}\, Q\right) \left(\psi_{11} - \psi_{12}\, Q\right)^{-1} x(k)$$

Therefore,

$$x(k+1) = \left(\theta_{11} + \theta_{12}\left(\psi_{21} - \psi_{22}\, Q\right) \left(\psi_{11} - \psi_{12}\, Q\right)^{-1} \right) x(k)$$

From this follows a sufficient condition for the stability of the optimal on-line controller.

Theorem 2.1: If

$$\left(\theta_{11}+\theta_{12}\left(\psi_{21}-\psi_{22}\, Q\right)\left(\psi_{11}-\psi_{12}\, Q\right)^{-1}\right)^{*}\left(\theta_{11}+\theta_{12}\left(\psi_{21}-\psi_{22}\, Q\right)\left(\psi_{11}-\psi_{12}\, Q\right)^{-1}\right) - I$$

is negative definite, then the system employing the on-line controller (without inequality constraints) is stable.

Of course, the choice of the Lyapunov function, (26), may be overly restrictive. In this case some other choice will have to be investigated.

It seems that the stability problem will become more severe as the optimization interval is shortened. Other problem areas may include time lag in computation and process parameter errors. These problems will be left as future research topics. Let us look at an example to demonstrate the theorem given above.

Example 2.1: For the process

$$x(k) = .9\, x(k-1) + u(k) \qquad x(0) = 1$$

we will use a $u(k)$ which minimizes

$$J = \sum_{j=k}^{k+4} \frac{1}{2}\, x(j)^2 + \frac{1}{2}\, u(j)^2$$

Equations (15) and (16) become

$$\begin{bmatrix} x(k) \\ p(k) \end{bmatrix} = \begin{bmatrix} 2.01111 & 1.11111 \\ 1.11111 & 1.11111 \end{bmatrix} \begin{bmatrix} x(k-1) \\ p(k-1) \end{bmatrix}$$

and (17) becomes

$$\begin{bmatrix} x(k+4) \\ -x(k+4) \end{bmatrix} = \begin{bmatrix} 39.90408 & 26.87973 \\ 26.87973 & 18.13147 \end{bmatrix} \begin{bmatrix} x(k) \\ p(k) \end{bmatrix}$$

Eliminating $x(k+4)$ we get

$$p(k) = -\, 1.48371\, x(k)$$

Therefore,

$$x(k+1) = .36255\ x(k)$$

Applying the theorem, $(.36225)^2 - 1 < 0$. Therefore we have stability.

2.8 How Good is Suboptimal?

For the general philosophy, the controller was based on performing optimization at every sampling instant over a finite interval into the future. Several reasons were given for doing this. One of the reasons was the uncertainty in the process into the future. The question then arises: How good is the controller based on a fixed optimization interval, if the process is known into the future? (We reiterate again that the controller based on a fixed optimization interval may be the best one could do in the face of uncertainty.) As a comparison, we can make the following two computations. First, we will solve the problem which minimizes

$$\sum_{0}^{\infty} \frac{1}{2}\ \|x(k)\|_Q^2 + \frac{1}{2}\ \|u(k)\|_R^2 \qquad \text{(Situation 1)}$$

with the process and initial conditions given. This solution is strictly open-loop. Secondly, we solve the problem which minimizes at every sampling instant the following criterion.

$$\sum_{j=k}^{k+N} \frac{1}{2}\ \|x(j)\|_Q^2 + \frac{1}{2}\ \|u(j)\|_R^2 \qquad \text{(Situation 2)}$$

The second philosophy is the basis for our on-line controller. For the comparison we will assume that the process is known for all times. We will illustrate the comparison with two examples.

Example 2.2: Let us consider the scalar process

$$x(k) = .9\ x(k-1) + u(k) \qquad x(0) = 1$$

For Situation 1, we use

$$\sum_{0}^{\infty} \frac{1}{2}\, x(k)^2 + \frac{1}{2}\, u(k)^2$$

For Situation 2, we use

$$\sum_{j=k}^{k+4} \frac{1}{2}\, x(j)^2 + \frac{1}{2}\, u(j)^2$$

We will use the calculus of variations approach. The Euler equations are

$$x(k) = .9\ x(k-1) + u(k)$$

$$p(k+1) = 1.11111\ p(k) + 1.11111\ x(k)$$

$$u(k) = p(k)$$

Eliminating u(k), we get

$$\begin{bmatrix} x(k+1) \\ p(k+1) \end{bmatrix} = \begin{bmatrix} 2.01111 & 1.11111 \\ 1.11111 & 1.11111 \end{bmatrix} \begin{bmatrix} x(k) \\ p(k) \end{bmatrix}$$

For Situation 1, we can eliminate p(k) and obtain

$$x(k+2) - 3.12222\, x(k+1) + x(k) = 0$$

This has the general solution

$$x(k) = A\,(.36234)^k + B\,(2.75988)^k$$

To satisfy the initial conditions: $x(0) = 1$ and $x(\infty) = 0$, we obtain

$$x(k) = (.36234)^k \qquad \text{(Situation 1)}$$

For situation 2, the solution is given in the example of the previous section, or

$$x(k) = (.36255)^k \qquad \text{(Situation 2)}$$

When 8T was considered as the optimization interval the response was

$$x(k) = (.36235)^k$$

Example 2.3: The conditions are the same as the last example except we take an unstable process given by

$$x(k) = 1.11111\, x(k-1) + u(k)$$

The Euler equations after eliminating u(k) become

$$\begin{bmatrix} x(k+1) \\ p(k+1) \end{bmatrix} = \begin{bmatrix} 2.01111 & .9 \\ .9 & .9 \end{bmatrix} \begin{bmatrix} x(k) \\ p(k) \end{bmatrix}$$

For Situation 1, the solution is

$$x(k) = .39789^k$$

For Situation 2 with 4T as the optimization interval, the solution is

$$x(k) = .39858^k$$

For Situation 2 with 8T as the optimization interval, the solution is

$$x(k) = .39791^k$$

The amazing revelation of these examples is that only a short finite time into the future is required for the optimization interval. Of course, more complicated processes may require a longer optimization interval. Example 2.3 reveals that unstable processes can be controlled using the above procedure.

2.9 Additional Remarks

This chapter provides background for the extension given in Part 1. It provides review material for the discrete version of the linear process and quadratic criterion case. No inequality constraints are considered in this chapter. The extension in the next chapter considers inequality constraints on the control variable.

Two algorithms were presented for computing the optimal control based on two different approaches to the optimal control problem. A third possible approach is the use of the steepest descent method. It is not discussed here because it is presented in the dissertation by Hsieh (Reference 28).

A philosophy for the adaptive scheme (perform optimization over a fixed interval into the future) is given in this chapter. This approach will be verified in Chapter 3 for the case with inequality constraints through experimentation.

CHAPTER 3

SYNTHESIS OF CONTROL FORCES WITH INEQUALITY CONSTRAINTS

3.1 Introduction

In this chapter, we extend considerations given in Chapter 2 to the case when we impose inequality constraints on the control variable.

The problem of on-line synthesis of control forces is no different from the optimization problem. The difficult requirement is that it must be rapidly performed. Also for the adaptive task, it must be performed in terms of easily measured parameters.

Horing (Reference 29) has considered an on-line controller calling it a predictive controller. He solves the same problem by using concepts from pattern recognition and he synthesizes the controller by adders and logical elements. Complexity arises in his method if he wishes to lengthen the optimization interval.

Ho and Brentani (Reference 30) have extensively studied quadratic programming methods applied to the control problem. The problem of minimizing the quadratic error over an optimization interval falls in their nonlinear class requiring additional calculations. It is shown subsequently that the quadratic error problem can be attacked directly using the formulation by Ho (Reference 31) from an earlier paper. Ho and Brentani explore a method which projects the gradient on the feasible region, R. Although this method can also be applied to our problem an alternate method used by Hildreth (Reference 32) called a coordinatewise gradient method will be explored. Both methods can be applied to the particular control problem with ease (in comparison to some general quadratic programming problem). It is to be emphasized that this report is exploring a follower type controller in comparison to the more difficult (computationally) trajectory optimization problem.

In this chapter, the coordinatewise gradient method will be described. Secondly, some simulation results will be presented showing responses to several different inputs. A comparison is made with responses of conventional sampled-data systems. Effects of parameter errors on the responses are experimentally observed. Extensions are then made to bounds on the rate of change of the control variable. One extension gives a hybrid computational procedure.

In Appendix 2, a brute-force method is described. The dimensionality problem of this method is indicated, thus recommending the gradient method. Although of little practical value, a study of the brute-force method is important in that it gives geometrical insight into the problem.

3.2 Problem Formulation

The philosopy for the determination of the control force was stated in Chapter 2. In addition to the consideration given to the formulation of the problem posed in Chapter 2, we require the control variables to be bounded, i.e.,

$$|u(k)| \leq M \tag{27}$$

For the sake of ease in presentation, the single control force and single output case will be considered. Generalization can be made to the multi-pole case (see Ho -- Reference 30). The input-output relationship of the process is given by

$$y(\ell) = \sum_{j=1}^{\ell} g(\ell+1-j)\, u(j) + y_o(\ell) \qquad (28)^{\dagger}$$

where

$g(\ell)$ - response to a unit pulse of width T at ℓT seconds from the initiation of pulse

$y_o(\ell)$ - initial condition response

The $g(\ell)$ are to be estimated by methods in Chapters 4 and 5. Equation (27) is rewritten in matrix form.

$$\underset{\sim}{y} = G\, \underset{\sim}{u} + \underset{\sim}{y}_o \tag{29}$$

where

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \cdot \\ \cdot \\ \cdot \\ y(N) \end{bmatrix} \qquad u = \begin{bmatrix} u(1) \\ u(2) \\ \cdot \\ \cdot \\ \cdot \\ u(N) \end{bmatrix} \qquad y_o = \begin{bmatrix} y_o(1) \\ y_o(2) \\ \cdot \\ \cdot \\ \cdot \\ y_o(N) \end{bmatrix}$$

$$G = \begin{bmatrix} g(1) & 0 & 0 & \cdots & 0 \\ g(2) & g(1) & 0 & & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ g(N) & g(N-1) & & & g(1) \end{bmatrix} = \left[\underset{\sim}{g}_1 \;\middle|\; \underset{\sim}{g}_2 \;\middle|\; \cdots \;\middle|\; \underset{\sim}{g}_N \right]$$

[†] As was done in Section 2.5, the index k has been dropped. $j = 1 \Rightarrow j = k+1$, $j = N \Rightarrow j = k + N$.

This G matrix is triangular because of physical realizability. Also, the $\underset{\sim}{g}_i$ are linearly independent if $g(1) \neq 0$. For this discrete case the G matrix has rank N if and only if the process is controllable (Reference 31). It is observed that if $g(1) \neq 0$ the system is controllable.

In terms of the above notation the criterion becomes

$$J = \frac{1}{2} \left|\left|\underset{\sim}{y}_d - \underset{\sim}{y}\right|\right|^2 \tag{30}$$

where $\underset{\sim}{y}_d$ - desired trajectory. Let

$$\underset{\sim}{d}' = \underset{\sim}{y}_d - \underset{\sim}{y}_o, \tag{31}$$

and

$$\underset{\sim}{d} = \sum_{i=1}^{N} u(i)\, \underset{\sim}{g}_i \tag{32}$$

The $\underset{\sim}{d}$ will not in general be made equal to $\underset{\sim}{d}'$ because of (27).

The problem which can now be stated is: Determine u(i) which minimizes

$$J = \frac{1}{2} \left|\left|\underset{\sim}{d}' - \underset{\sim}{d}\right|\right|^2 \tag{33}$$

Each column vector, $\underset{\sim}{g}_i$, can be viewed as a basis which collectively spans a linear manifold of E_N (output-space). Without bounds the problem can be solved readily[†] because the G matrix is triangular. With bounds the problem is to determine a point in a closed convex region which is nearest to the desired point, $\underset{\sim}{d}'$. The closed convex region is in particular a parallelotope in E_N.

3.3 Coordinatewise Gradient Method

In this section we look at a gradient method to iteratively approach the optimum point. We modify the method of steepest descent to consider limitations on the movement of the trial point. Because of the simplicity of the boundaries (parallelotope) compared to some general quadratic programming problem we anticipate some easy gradient method to apply. Ho (Reference 30) also utilizes the simpleness of the boundaries in his method. An obvious method is to adjust each component one at a time. In this way the bounds on the components can easily be applied.

Let us look at a two-dimensional problem as shown in Figure 9. In the method we can start from any point in R. For the sake of discussion, let us begin at the origin, O. First, we move in the u(1) direction. In the u(1) direction, we seek the minimum which is located at point a'.

[†] This statement is not true for other criterions which include, for example, a penalty for control energy.

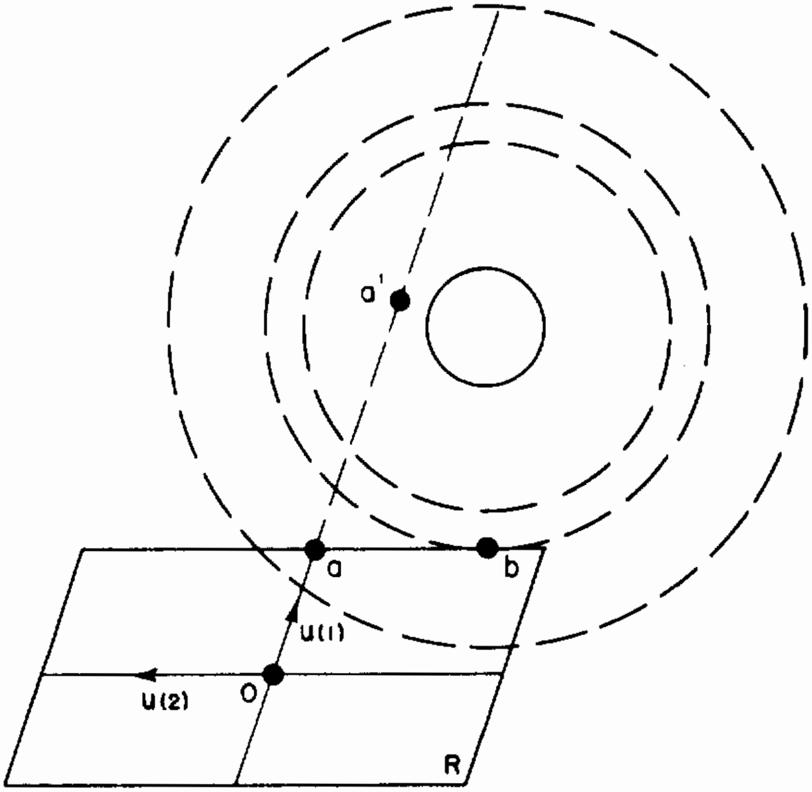


Figure 9. Path of Descent

Since we cannot reach that point we stop at point a. Next, we seek a minimum in the u(2) direction starting from point a. The minimum in the u(2) direction is found at point b. Since this is the optimum point in R, we have reached the optimum in two iterations. (For higher dimensions the optimum will usually not be reached so rapidly.)

Next, the equations which will be programmed will be derived. The point in R is given by

$$\underset{\sim}{d} = \sum_{j=1}^{N} u(j)\, \underset{\sim}{g}_i = G\, \underset{\sim}{u} \tag{32}$$

We seek the minimum of

$$J = \frac{1}{2} \left|\left|\underset{\sim}{d}' - G\, \underset{\sim}{u}\right|\right|^2$$

The gradient along a component is

$$\frac{\partial J}{\partial u(j)} = \underset{\sim}{g}_j^{*}\, G\, \underset{\sim}{u} - \underset{\sim}{g}_j^{*}\, \underset{\sim}{d}' = \nabla_j \tag{34}$$

It is noted that the gradient along a component is a scalar. The corrected value for the u(j) component is

$$u(j)^{(n+1)} = u(j)^{(n)} + \epsilon_n\, \nabla_j^{(n)}$$

The ϵ_n is found by seeking the minimum along the direction of the j^{th} component $\underset{\sim}{g}_j$. Expanding J,

$$2J = <G^{*}\, G\, \underset{\sim}{u},\ \underset{\sim}{u}> - 2<G^{*}\, \underset{\sim}{d}',\ \underset{\sim}{u}> + \left|\left|\underset{\sim}{d}'\right|\right|^2$$

Let us work with the terms which depend on $\underset{\sim}{u}$.

$$Q(\underset{\sim}{u}) \triangleq <G^{*}\, G\, \underset{\sim}{u}, \underset{\sim}{u}> - 2<G^{*}\, \underset{\sim}{d}',\ \underset{\sim}{u}> \tag{35}$$

Also,

$$\underset{\sim}{u}^{(n+1)} = \underset{\sim}{u}^{(n)} + \epsilon_n\, \underset{\sim}{m}^{(n)}$$

where $\underset{\sim}{m}^{(n)}$ is zero except for the j^{th} element, which is equal to $\nabla_j^{(n)}$. Substituting $u^{(n+1)}$ into (35) we obtain

$$Q\left(\underset{\sim}{u}^{(n)} + \epsilon_n\, \underset{\sim}{m}^{(n)}\right) = Q\left(\underset{\sim}{u}^{(n)}\right) + 2\,\epsilon_n < G^{*}\, G\, \underset{\sim}{u}^{(n)} - G^{*}\, \underset{\sim}{d}',\ \underset{\sim}{m}^{(n)} >$$

$$+ \epsilon_n^2 < G^{*}\, G\, \underset{\sim}{m}^{(n)},\ \underset{\sim}{m}^{(n)} >$$

The minimum along a particular direction is then given by

$$\frac{d}{d\,\epsilon_n} Q(\quad) = 2 < G^* G\, \underset{\sim}{u}^{(n)} - G^* \underset{\sim}{d}',\ \underset{\sim}{m}^{(n)} >$$

$$+ 2\,\epsilon_n < G^* G\, \underset{\sim}{m}^{(n)},\ \underset{\sim}{m}^{(n)} > = 0$$

Or,

$$\epsilon_n = \frac{< G^* G\, \underset{\sim}{u}^{(n)} - G^* \underset{\sim}{d}',\ \underset{\sim}{m}^{(n)} >}{< G^* G\, \underset{\sim}{m}^{(n)},\ \underset{\sim}{m}^{(n)} >}$$

The vector $\underset{\sim}{m}$ is zero except for the j^{th} element. Therefore, ϵ_n in the j^{th} direction is

$$\epsilon_{n_j} = \frac{-\nabla_j}{\underset{\sim}{g}_j^* \underset{\sim}{g}_j \nabla_j} = \frac{-1}{< \underset{\sim}{g}_j, \underset{\sim}{g}_j >}$$

Therefore, at the n^{th} step we get the $n+1$ approximation by

$$u(j)^{(n+1)} = u(j)^{(n)} - \frac{\nabla_j^{(n)}}{||\underset{\sim}{g}_j||^2} \tag{36}$$

As $u(j)^{(n+1)}$ could possibly exceed a bound we must limit its amplitude, or

$$u(j)^{(n+1)} = \underset{M}{\text{sat}} \left[u(j)^{(n+1)} \right] \tag{37}$$

The quantity on the left is used for the next iteration. Therefore, the vital equations are (34), (36), and (37). The simplicity of the equations to be solved is noted. Every iteration requires only

$N^2/2 + 5N/2 - 1$ additions, $N^2/2 + 5N/2$ multiplications, and 1 division.

An iteration for the coordinatewise gradient method should not be compared with one iteration for Ho's method. The computation time for N iterations of the coordinatewise gradient method should more closely correspond with one iteration of the other method.

The procedure described above can be modified to possibly improve the rate of convergence. Before each iteration, the gradient in each coordinate direction is evaluated. The direction of the largest gradient is then chosen for the descent. If no motion is possible in that direction the direction for the next highest gradient is chosen, etc. Of course, such a procedure will demand more from the computer; however, it may still be much simpler than other methods.

3.4 Remarks on Convergence

Comments in this section will be largely heuristic, appealing to the geometrical picture. A discussion on the existence and uniqueness is given in Appendix 3.

The proof of convergence has been given by Hildreth (Reference 32) and D'Esopo (Reference 33) for the parallelotope region that we have (rectangular in u-space). It should be emphasized that convergence of the coordinatewise gradient method is assured only for this particular type of constraint. Geometrically, the convergence can be visualized for the two-dimensional case. The criterion function, J, defines a surface in d-space which is a circular paraboloid (non-circular paraboloid in u-space). In the parallelotope region, R, we are to converge upon the lowest point on this surface. At each iteration (although we select the direction of the coordinates) we measure the slope and we choose to go in the negative slope direction. Along any direction the slope is either positive, negative, or zero. If zero, we temporarily do nothing because if the point is non-minimal some other coordinate will have nonzero slope. The procedure stops when either we arrive at a point where all the gradient components are zero (min in R), or motion of the trial point is restrained by the boundaries of R. If restrained by a single boundary, the gradient will be normal to that boundary.

3.5 A Remark on the Initial Trial

Of course, the success of the gradient method will depend upon the closeness of the initial guess or trial to the answer. This section describes a technique whereby a good initial guess can be obtained. As previously described we envision repeating the same optimization procedure every T seconds. Although the optimization yields the control force for the entire optimization interval, NT, only the first component is ever used. However, the other components can be used as an initial approximation for the following interval of consideration. If the changes caused by disturbances and process and input variations are small during T, one should be able to compute the optimal controls rapidly since the initial approximations will be very close to the optimal point. In Figure 10, u(2) in interval 1 becomes the first guess for u(1) in interval 2. Only an initial approximation for the last T seconds is missing. For this reason, the iteration is initiated from the last T interval, working forward, and repeating this process. In this way the first iteration will not disturb the initial good approximation of the other intervals. For the reason that only one component may be initially indeterminate, it is felt that the coordinatewise gradient method may be the most suitable in this application.

If no initial approximation is available, the unbounded solution can be computed. By simply passing the unbounded solution through a limiter operation we have a possible initial guess.

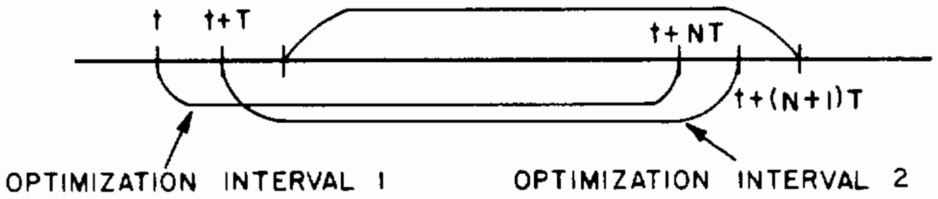


Figure 10. Translation of Optimization Intervals

3.6 Example for One Optimization Interval

Before proceeding to the simulation of the controller in a closed-loop, let us examine in detail the iteration procedure for one optimization interval. We take a four-dimensional example. Let us consider the following linear process described in terms of the Laplace transfer function

$$\frac{Y(s)}{U(s)} = \frac{.5}{s(s+.5)}$$

with a sampling period in the controller of T = 1 sec. The unit pulse response is given by the succession of the following values

$$g_j = (.21306,\ .52270,\ .71050,\ .82442)$$

The G matrix is

$$G = \begin{bmatrix} .21306 & 0 & 0 & 0 \\ .52270 & .21306 & 0 & 0 \\ .71050 & .52270 & .21306 & 0 \\ .82442 & .71050 & .52270 & .21306 \end{bmatrix}$$

We will let $y_d(j) = 1$ and assume that the initial condition is equal to zero. Therefore, $d'(j) = 1$. We restrict $u(j)$ such that $|u(j)| \leq 5.5$.

Using the gradient method we assume as a first approximation the set u(j) obtained by limiting the unbounded solution. The unbounded solution for the problem is

$$u(j)' = (4.69,\ -6.82,\ 5.77,\ -4.89)$$

Therefore the first approximation is

$$u(j)^{(1)} = (4.69,\ -5.5,\ 5.5,\ -4.89)$$

Figure 11 shows the optimum bounded-control sequence obtained from the gradient method and the brute-force method as described in Appendix 2. The non-gradient solution was possible because the example chosen was one of the special cases (see Appendix 2). (The brute-force method was not programmed in general terms.) Also, Figure 11 shows the unbounded solution. It is noted that although the unbounded solution exceeds the bounds twice, the bounded solution has only one component at the boundary.

Figure 12 shows the corresponding output of the linear process. Actually, the output will be continuous rather than the staircase signal shown. The staircase response is plotted for convenience and the response at the sampling instants will correspond exactly with the actual response.

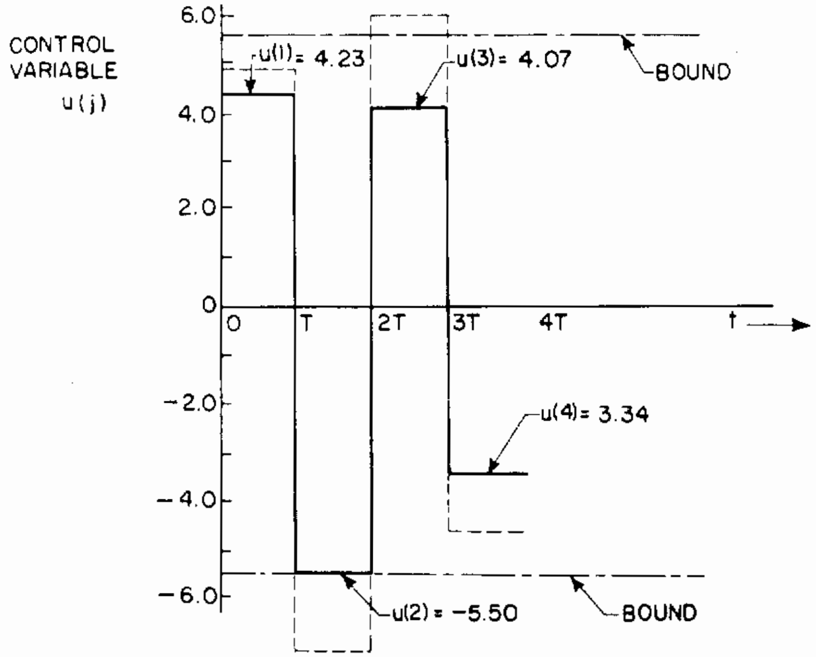


Figure 11. Bounded Control Sequence

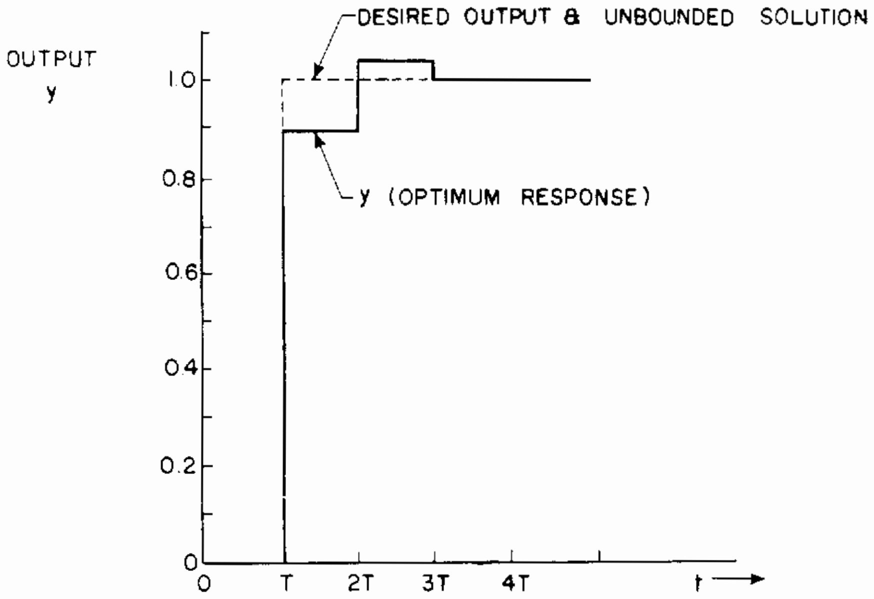


Figure 12. Output of Process for Optimum Control Sequence

Table I shows how the optimum point is approached by the gradient method. The gradient method has the characteristic that errors are initially rapidly reduced and the finer accuracy is obtainable only after many iterations. Table 1 shows that good approximations are obtained after 16 iterations. In an adaptive control task the solutions should be approached even sooner because as discussed in the previous section we generally have a good initial approximation.

3.7 Simulation

A digital simulation was performed on an IBM 7090 to operate the controller in a feedback loop. The flow chart is shown in Figure 13.

First, the controller was required to cause the process to follow a triangular wave. The process used previously (an example) was again considered. Optimization intervals of 4T and 8T were considered with T = 1 sec. A comparison is made with a conventional controller shown in Figure 14 for which the K was chosen so that the damping ratio was 0.5. For a comparison, the bounds on the on-line controller were selected from the maximum and minimum control forces experienced by the conventional controller. The numbers of iterations per sampling interval were respectively 20 and 40 for the 4T and 8T cases. (This means that each component was iterated 5 times.) Simulation was performed over 100 sampling intervals.

A portion of the results is shown in Figure 15. A marked improvement in the response is noted. The conventional controller response shows the characteristic lag which is not present for the on-line controller response. For the example chosen it is seen that no appreciable difference is seen in the responses of the 4T and 8T cases. The number of iterations was increased by a factor of two with no appreciable difference in the response.

The control forces for the conventional controller and the on-line controller are shown respectively in Figures 16 and 17. The on-line controller's controls are more jumpy but such constraints on the rate of change were not considered in the optimization.

An estimate can be made of the computation time per sampling interval using the formulas previously stated. (No. (Add) = $N^2/2+5N/2-1$, No. (Multiply) = $N^2/2+5N/2$, No. (Division) = 1 per iteration.) Let us assume that we have an on-line digital computer with an add-time of 35 μ sec. (The add-time for the IBM 7090 is 2 μ sec.) Considering that we have 10 digit multiplication and that the transfer time is 1/2 the add time, the estimate is .016 sec. per sampling interval for 20 iterations and 4T case (.001 sec. for IBM 7090). Therefore, compared to the 1 sec. sampling period the computation time is only a fraction.

As the pulse response of the previous example did not tail off (because of the integrator) another process was selected with

$$\frac{Y(s)}{U(s)} = \frac{.25}{(s+.5)^2}$$

TABLE 1

CONTROL SEQUENCE AND OUTPUT VS. NUMBER OF ITERATIONS

Iteration	u(1)	u(2)	u(3)	u(4)	y(1)	y(2)	y(3)	y(4)	J
0	4.693	-5.50	5.50	-4.89	1.000	1.281	1.632	1.795	.5549
1	4.693	-5.50	5.50	-5.50	1.000	1.281	1.632	1.665	.4600
2	4.693	-5.50	3.987	-5.5	1.000	1.281	1.309	.874	.0954
3	4.693	-5.50	3.987	-5.5	1.000	1.281	1.309	.874	.0954
4	4.519	-5.50	3.987	-5.5	.963	1.190	1.185	.730	.0724
8	4.418	-5.50	3.863	-4.231	.941	1.138	1.087	.853	.0259
12	4.360	-5.50	3.805	-3.539	.929	1.107	1.034	.921	.0119
16	4.326	-5.50	3.782	-3.171	.922	1.089	1.004	.960	.0079
20	4.305	-5.50	3.779	-2.983	.917	1.078	.989	.981	.0067
24	4.292	-5.50	3.787	-2.895	.914	1.071	.981	.993	.0064
28	4.283	-5.50	3.799	-2.862	.913	1.067	.978	.999	.0063
32	4.277	-5.50	3.814	-2.859	.911	1.064	.977	1.003	.0062
36	4.272	-5.50	3.830	-2.872	.910	1.061	.977	1.004	.0062
44	4.266	-5.50	3.861	-2.917	.909	1.058	.978	1.005	.0061
52	4.261	-5.50	3.889	-2.968	.908	1.055	.981	1.005	.0060
60	4.257	-5.50	3.913	-3.015	.907	1.053	.983	1.005	.0059
68	4.253	-5.50	3.935	-3.058	.906	1.051	.986	1.004	.0058
79	4.251	-5.50	3.962	-3.111	.906	1.050	.989	1.004	.0058
Ans.	4.231	-5.50	4.075	-3.337	.902	1.040	1.000	1.000	.0056

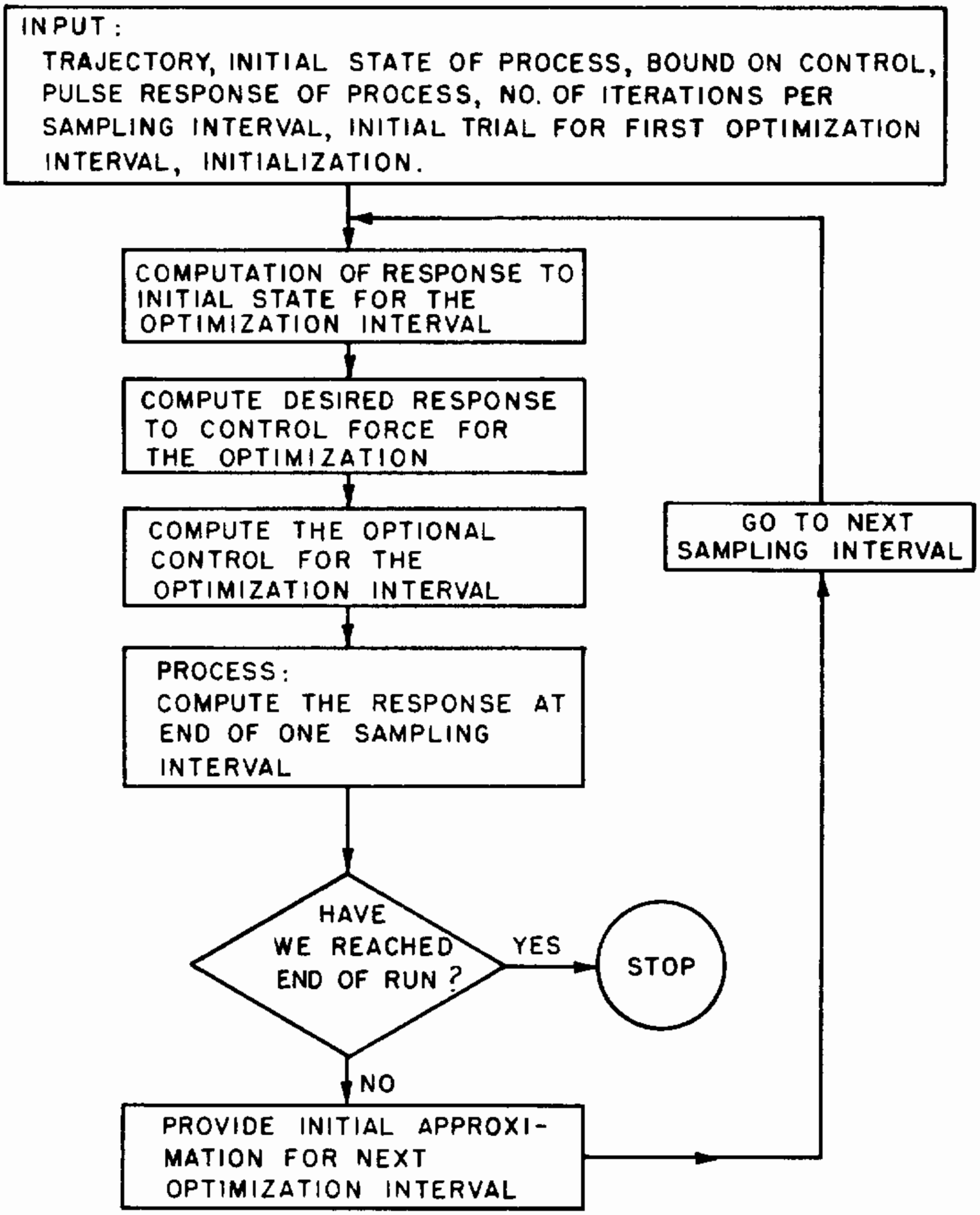

INPUT:
TRAJECTORY, INITIAL STATE OF PROCESS, BOUND ON CONTROL, PULSE RESPONSE OF PROCESS, NO. OF ITERATIONS PER SAMPLING INTERVAL, INITIAL TRIAL FOR FIRST OPTIMIZATION INTERVAL, INITIALIZATION.
COMPUTATION OF RESPONSE TO INITIAL STATE FOR THE OPTIMIZATION INTERVAL
COMPUTE DESIRED RESPONSE TO CONTROL FORCE FOR THE OPTIMIZATION
COMPUTE THE OPTIONAL CONTROL FOR THE OPTIMIZATION INTERVAL
PROCESS:
COMPUTE THE RESPONSE AT END OF ONE SAMPLING INTERVAL
HAVE WE REACHED END OF RUN ?
YES
STOP
NO
PROVIDE INITIAL APPROXIMATION FOR NEXT OPTIMIZATION INTERVAL
GO TO NEXT SAMPLING INTERVAL

Figure 13. Simulation Flow Chart

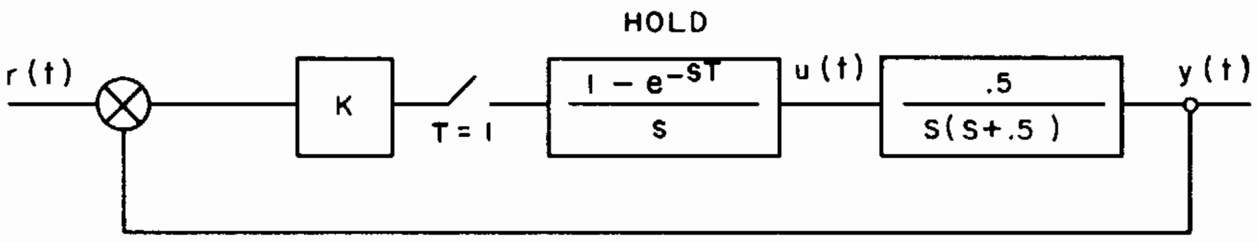


Figure 14. Conventional Controller

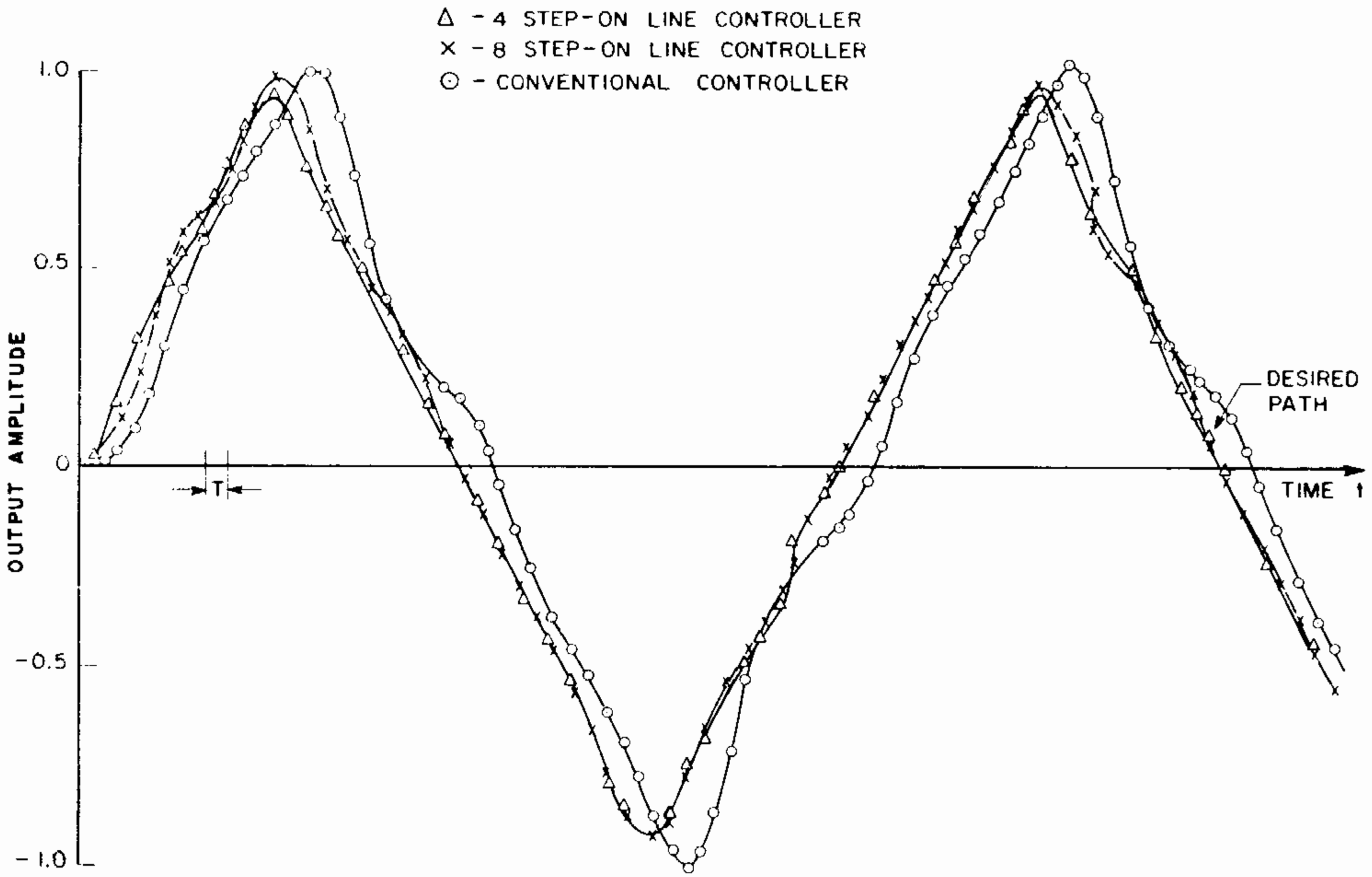


Figure 15. Comparison of On-Line vs. Conventional, for $\frac{y}{u} = \frac{.5}{s(s+.5)}$, Triangular Input

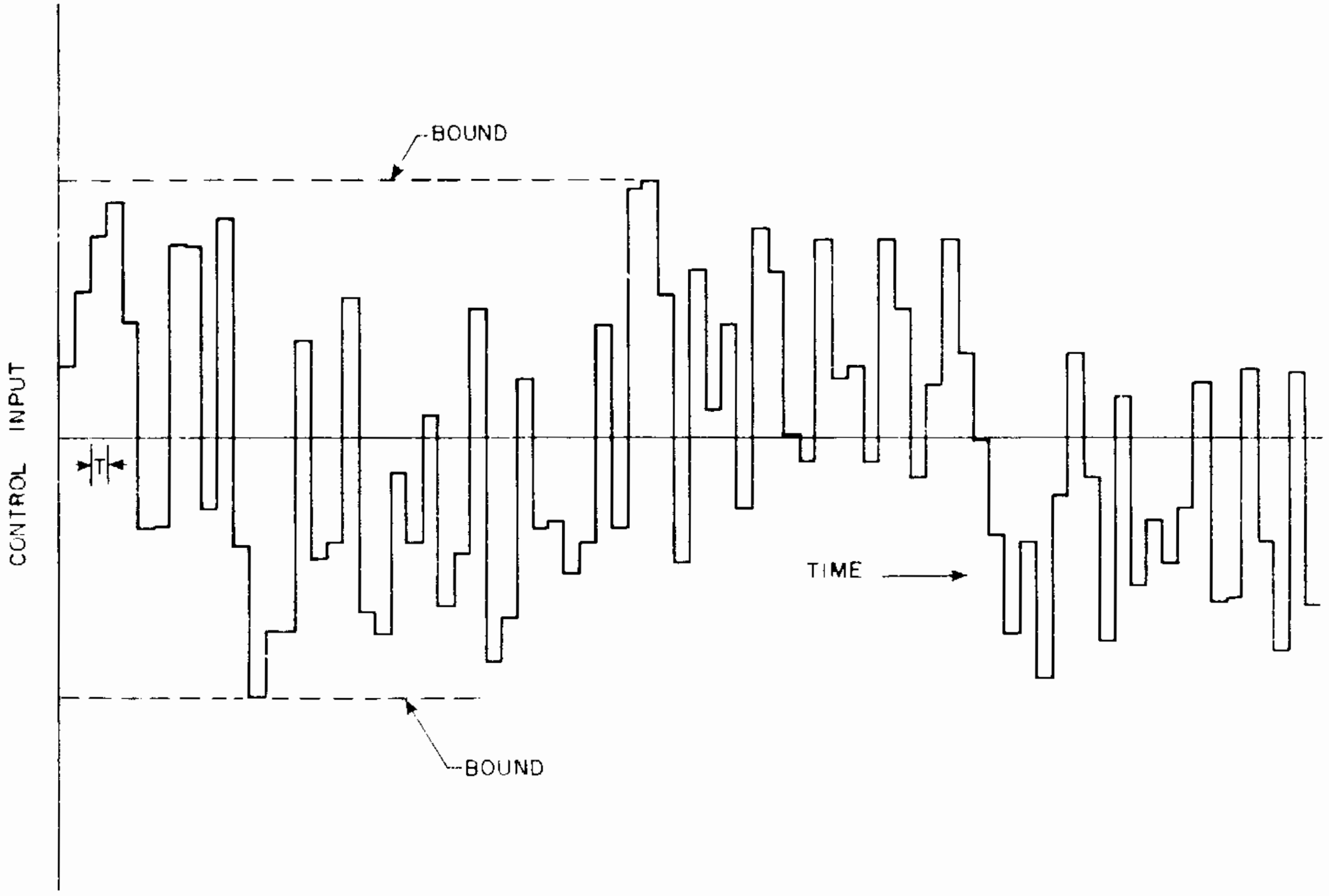


Figure 16. Control Force for On-Line Controller

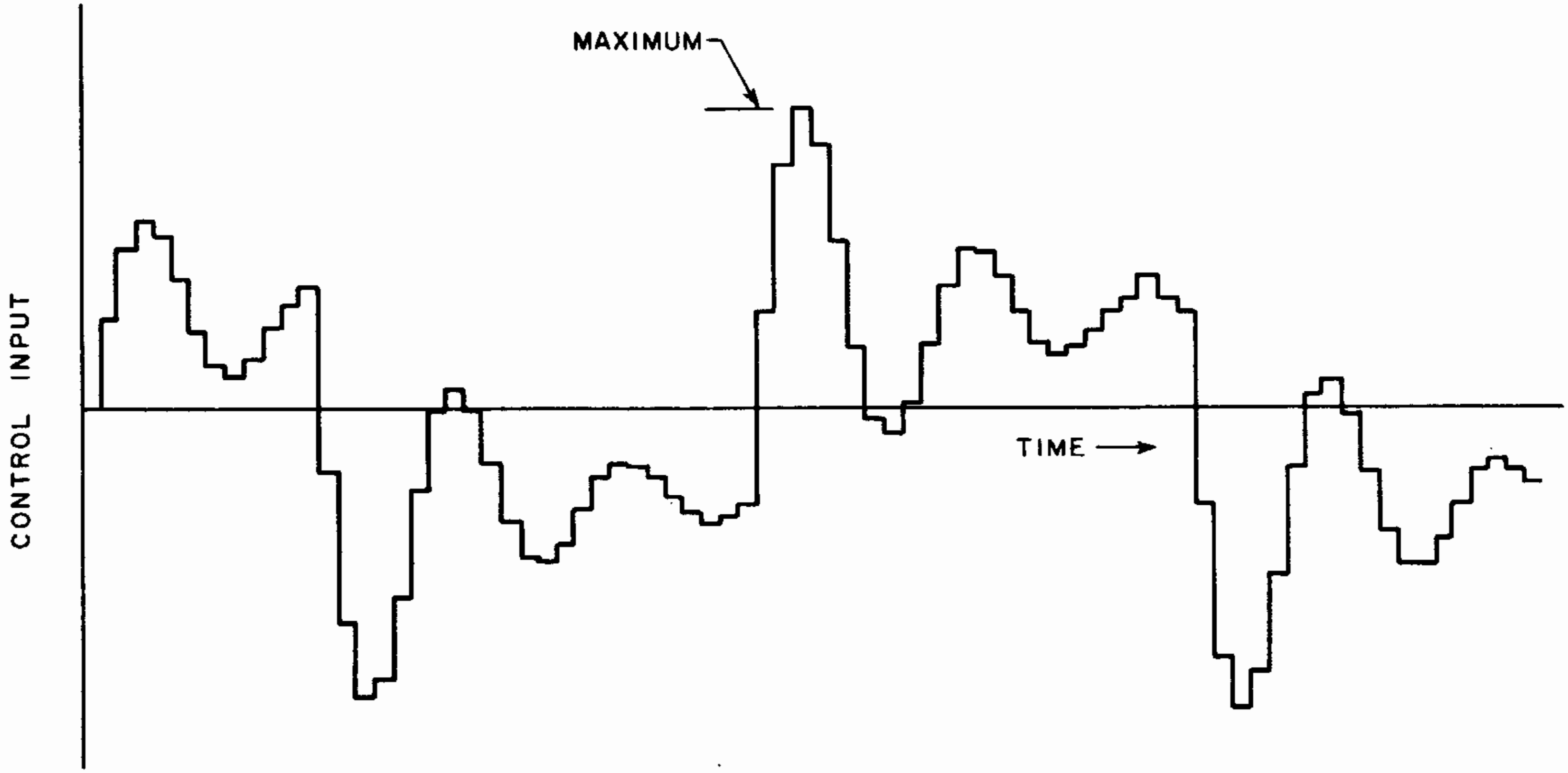


Figure 17. Control Force for Conventional Controller

and with T = 1 sec. The results of the simulation are shown in Figure 18 for the 8T case. Again, an improvement is noticed over the conventional controller. No appreciable improvement was noticed when the number of iterations was increased by a factor of two. A sine wave was also tried and the results are shown in Figure 19.

It is felt that the results reveal that some new types of responses can be obtained by using an on-line controller. It should be noted that if the conventional controller must operate in the linear range a simulation must be performed with all the possible inputs that the feedback process will encounter. On the other hand, the on-line controller at all times can do its best with the available control forces.

3.8 Effect of Uncertainties in Process Parameters

The optimal controls are computed assuming that the process is known accurately. In an adaptive task, one is not so fortunate as to have accurate knowledge of the process. It is very desirable then to know whether suitable control action is obtained even with inaccuracies in the process parameters, say 10%. If we have this condition, then assurance is given that if the process parameters are known to within 10%, then the overall system will behave satisfactorily. Therefore, optimal controls and trajectories should be experimentally studied with errors in process parameters. A few experimental results are reported in this section.

The situation of Figure 18 was studied further. The process was

$$\frac{Y(s)}{U(s)} = \frac{.25}{(s+.5)^2}$$

The time constant of 0.5 was uncertain to the controller and values of 0.45, 0.5, and 0.55 were respectively used. Optimization intervals of 8T and 4T iterations per sampling period were used. The differences in the responses were hardly noticeable to plot on a graph. Therefore, the initial part of the runs are tabulated in Table 2 for the 8T case for comparison purposes. The output of the conventional controller is also tabulated.

One should not draw sweeping conclusions from a single example. However, the results indicate that possibilities are present and any individual problem should be analyzed by simulation. The close tracking capability in spite of errors in the process information can possibly be attributed to the feedback which is present in the on-line controller.

3.9 Bounds on the Rate of Change of the Control Variable

Instead of having bounds on the amplitude, we can place bounds on the rate of change of the control variable. Let us look at the four-dimensional case as an example.

$$\underset{\sim}{d} = u(1)\,\underset{\sim}{g}_1 + u(2)\,\underset{\sim}{g}_2 + u(3)\,\underset{\sim}{g}_3 + u(4)\,\underset{\sim}{g}_4 \qquad (38)$$

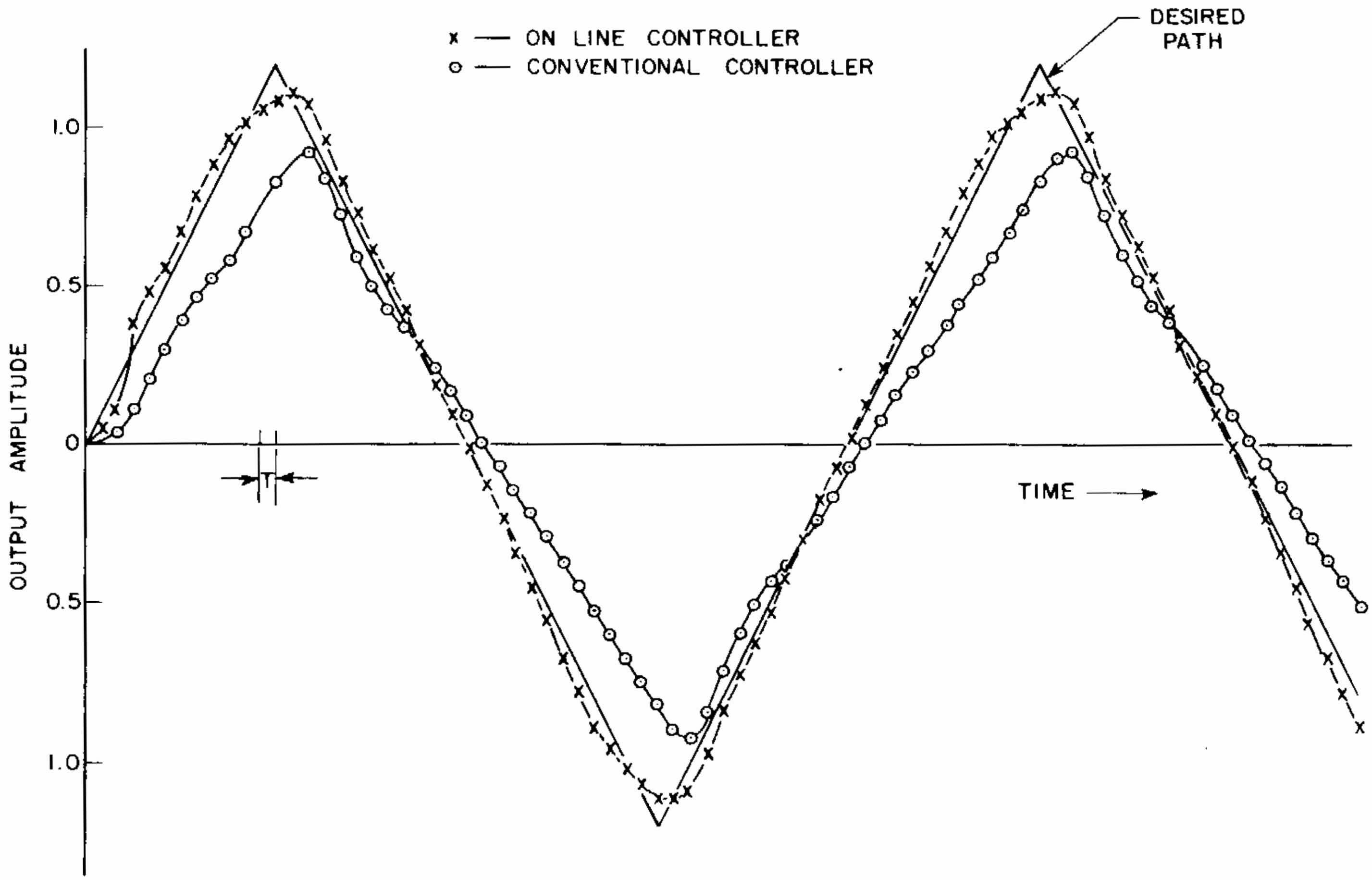


Figure 18. Comparison of On-Line vs. Conventional, $\frac{y}{u} = \frac{.25}{(s+.5)^2}$, Triangular Input

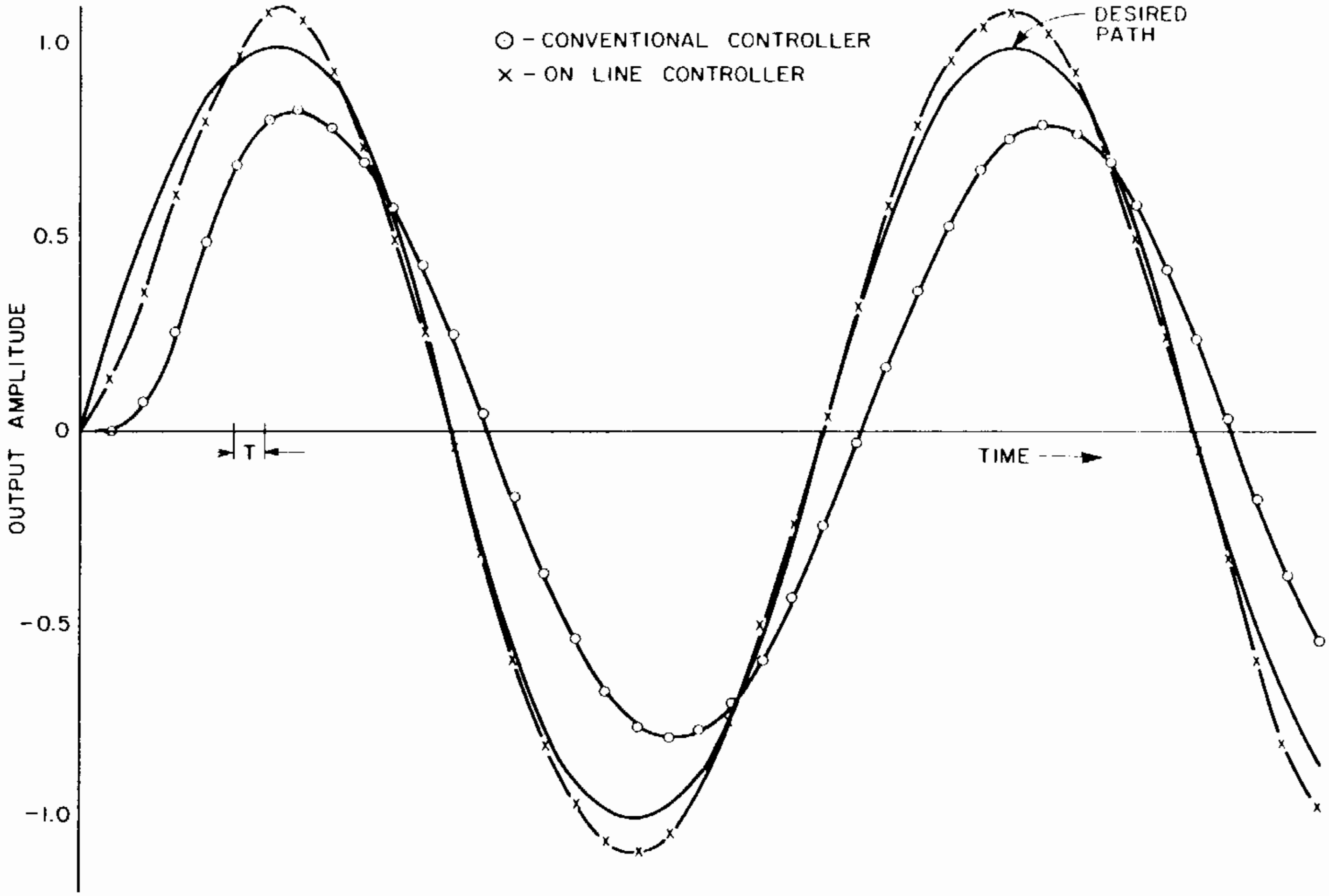


Figure 19. Comparison of On-Line vs. Conventional,
$\frac{y}{u} = \frac{.25}{(s+.5)^2}$, Sine Wave Input

TABLE 2

EFFECT OF UNCERTAINTIES IN PARAMETERS

k	Desired Path	a=.45	a=.5	a=.55	Conv.
1	1.0	.546	.559	.521	.0
2	2.0	1.986	1.838	1.771	.271
3	3.0	3.733	3.471	3.302	.990
4	4.0	4.847	4.783	4.628	1.981
5	5.0	5.628	5.590	5.568	2.974
6	6.0	6.665	6.517	6.500	3.819
7	7.0	7.801	7.635	7.573	4.521
8	8.0	8.798	8.651	8.582	5.173
9	9.0	9.589	9.472	9.411	5.858
10	10.0	10.182	10.094	10.046	6.600
11	11.0	10.611	10.547	10.510	7.381
12	12.0	10.912	10.867	10.841	8.163
13	11.0	11.120	11.089	11.071	8.927
14	10.0	10.499	10.640	10.763	9.131
15	9.0	9.440	9.760	10.004	8.429
16	8.0	8.181	8.460	8.738	7.189
17	7.0	7.093	7.266	7.473	5.950
18	6.0	6.180	6.263	6.386	5.014
19	5.0	5.170	5.274	5.369	4.364
20	4.0	4.033	4.193	4.320	3.812
21	3.0	2.935	3.108	3.247	3.193
22	2.0	1.789	1.938	2.097	2.455
23	1.0	.898	.966	1.056	1.643
24	0.0	- .306	- .205	- .098	.828
25	-1.0	-1.359	-1.190	-1.096	.051

We wish to bound the difference between succeeding control forces.

$$|u(k) - u(k-1)| \leq M_2$$

We put no constraints on the range of $u(i)$ itself. Rewriting (38) we get

$$\underset{\sim}{d} = u(1)\left(\underset{\sim}{g}_1 + \underset{\sim}{g}_2 + \underset{\sim}{g}_3 + \underset{\sim}{g}_4\right) + \left(u(2) - u(1)\right)\left(\underset{\sim}{g}_2 + \underset{\sim}{g}_3 + \underset{\sim}{g}_4\right)$$
$$+ \left(u(3) - u(2)\right)\left(\underset{\sim}{g}_3 + \underset{\sim}{g}_4\right) + \left(u(4) - u(3)\right)\ (\underset{\sim}{g}_4)$$

Let

$$\underset{\sim}{h}_i = \sum_{j=0}^{4-i} \underset{\sim}{g}_{4-j} \qquad \ell(i) = u(i) - u(i-1)$$

Then,

$$\underset{\sim}{d} = \ell(1)\,\underset{\sim}{h}_1 + \ell(2)\,\underset{\sim}{h}_2 + \ell(3)\,\underset{\sim}{h}_3 + \ell(4)\,\underset{\sim}{h}_4 \qquad (39)$$

where

$$|\ell(i)| \leq M_2$$

Now, we can use the same method as discussed previously and solve for $\ell(i)$ which in turn can be solved for $u(i)$.

3.10 Weighting Between Error and Control Energy

In place of (30) it may be desirable to use instead the following criterion which also penalizes control energy.

$$J = \frac{1}{2}\,||\underset{\sim}{y}_d - \underset{\sim}{y}||^2 + \frac{1}{2}\,||\underset{\sim}{u}||^2$$

Now, distances in state-space or y-space have no longer the same significance as before. With less geometrical significance, however, the problem can be viewed as done by Ho in the solution space or control space. If we still desire to limit the control force, a point is then desired in a hypercube, R. The two-dimensional problem is shown in Figure 20.

In Figure 20, the lines of constant J are no longer circular, but the J hyper-surface defined at every point of the solution space can be shown to be convex. It can be assumed here that J is continuous with bounded second partial derivatives with respect to $u(k)$. Then, J is a convex function of $u(k)$ if the symmetric matrix of the second partial derivatives is positive semi-definite at all points of R (see Eggleston, Reference 32, page 51). It is noted in passing that the sum of convex functions is convex. This follows simply from the fact that the sum of semi-definite matrices is semi-definite. Writing J in terms of u, we have

$$J = \frac{1}{2}\,||\underset{\sim}{d}' - G\,\underset{\sim}{u}||^2 + \frac{1}{2}\,||\underset{\sim}{u}||^2 \qquad (40)$$

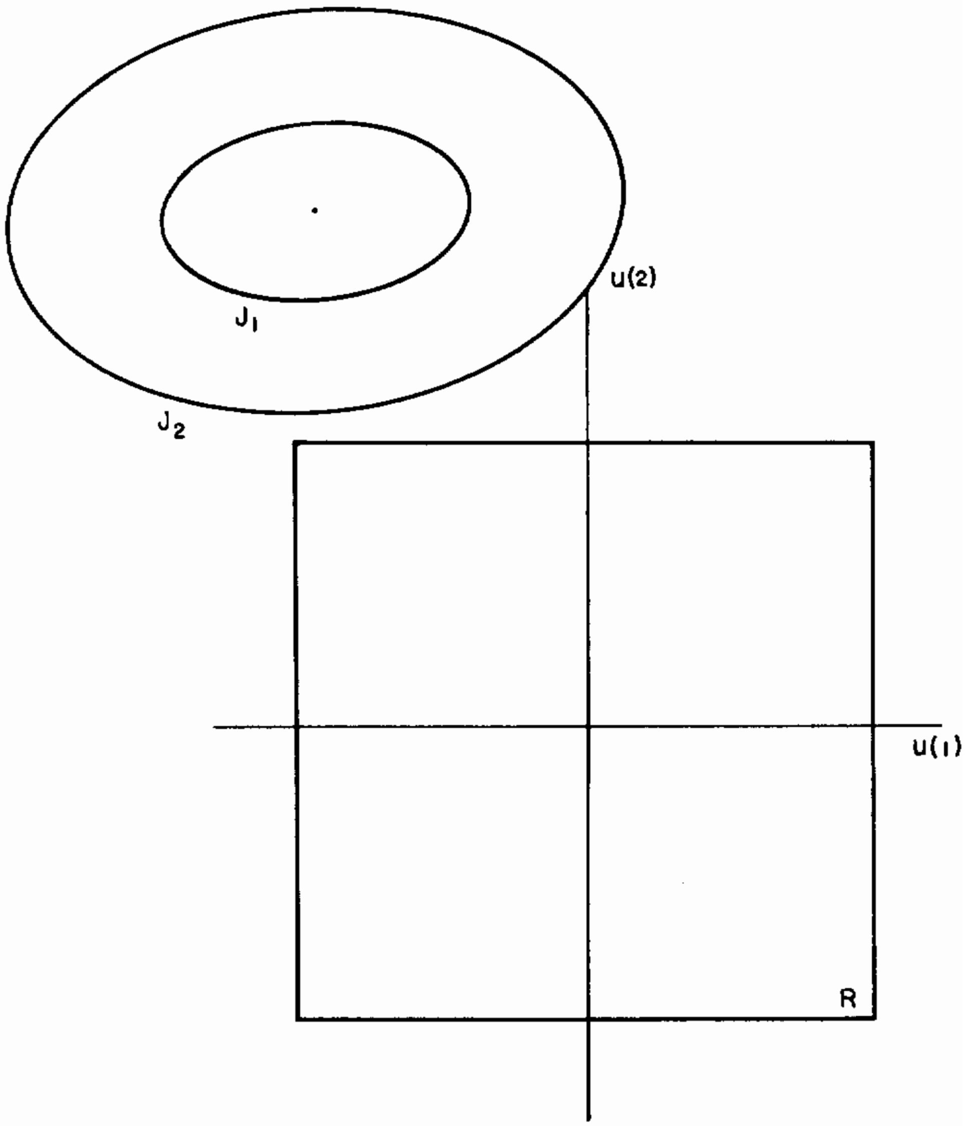


Figure 20. Solution Space (u-Space)

The second partial derivative matrix of the first term is $G^* G$ which is symmetric and positive definite (columns of G are linearly independent). The second partial derivative matrix of the second term is simply 2I. Therefore, the coordinatewise gradient method is still applicable for this case.

3.11 Bounds on Both Control Force and the Rate of Change of Control Force

Most practical systems have limitations both on the magnitude of the control force and on the rate of change of control force. Let us restate the problem with the added constraint.

Problem: Given a) Process:

$$y(\ell) = \sum_{j=1}^{\ell} g(\ell + 1 - j)\, u(j) + y_o(\ell)$$

b) Constraints:

$$|u(j)| \leq M_1$$

$$|u(j) - u(j-1)| \leq M_2$$

Determine: $u(j)$ $j = 1, 2, \ldots, N$ which minimizes

$$J = \frac{1}{2} \sum_{j=1}^{N} \left(y_d(j) - y(j)\right)^2$$

The problem is again a quadratic programming problem but with more constraints. The region from which a solution is to be chosen will no longer be a parallellotope. The region for the two-dimensional case is shown in Figure 21 in u-space.

The problem is to find a point in u-space and in the shaded region which has the smallest J. For such regions, the coordinatewise gradient method or Ho's simplified gradient projection method is not directly applicable. Therefore, a more involved method is required. Rosen's (Reference 35) gradient projection method is applicable but the use of such a scheme on-line is questionable. Thus, we look for a simpler scheme to apply to our particular problem.

The procedure to be described will transform the above problem so that the constraints will be rectangular. Such a scheme has been described by Hildreth (Reference 32). The constraints being rectangular, we can apply the coordinatewise gradient method or Ho's simplified gradient projection method. It should be noted that the following procedure can also be used for control problems with state variable constraints by converting to equivalent statements on u.

As before,

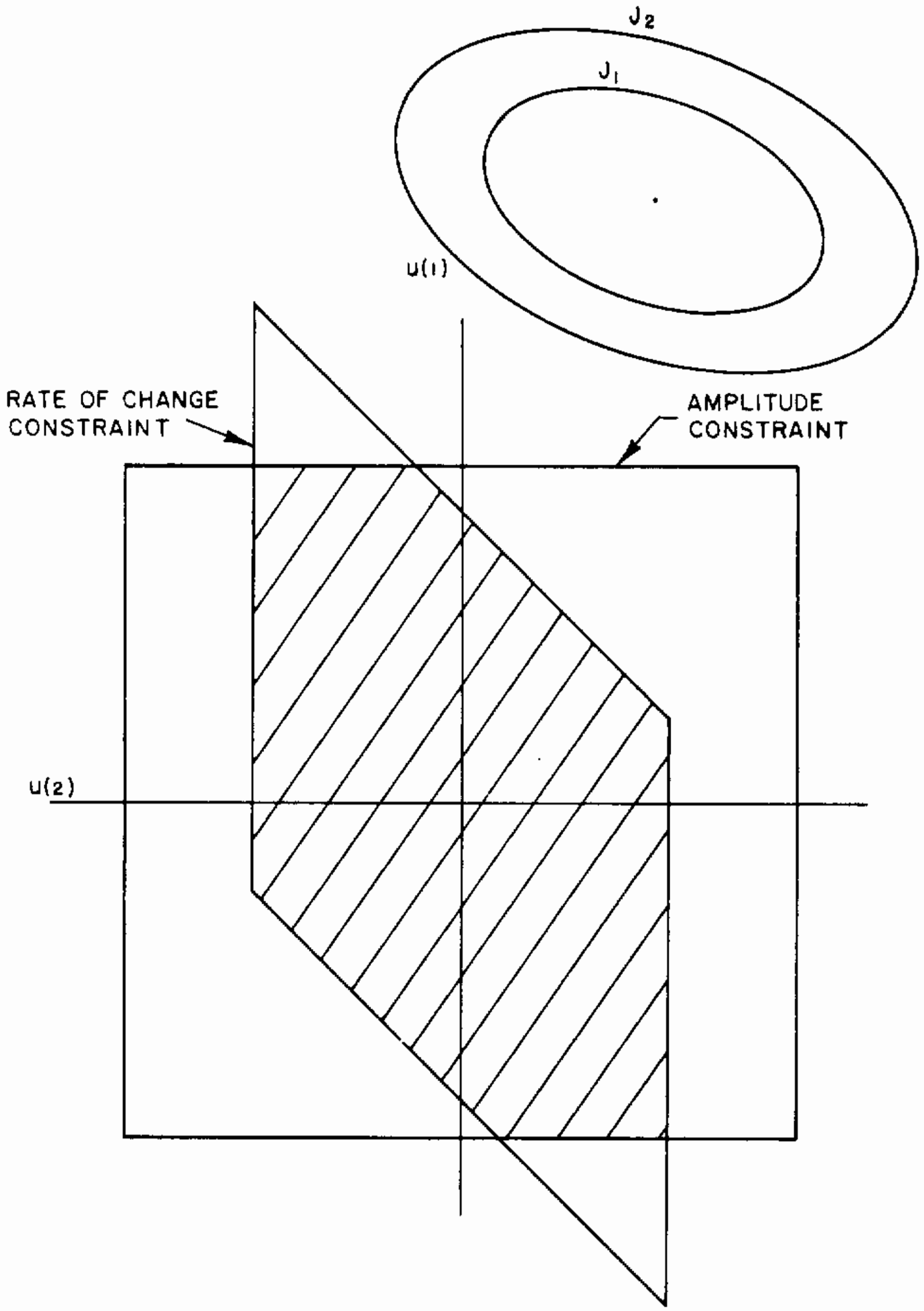


Figure 21. Two-Dimensional Case with Multiple Constraints

$$J(\underset{\sim}{u}) = \frac{1}{2} \left|\left|\underset{\sim}{d}' - G \underset{\sim}{u}\right|\right|^2$$

Taking the parts of $J(\underset{\sim}{u})$ which depend on $\underset{\sim}{u}$,

$$Q(\underset{\sim}{u}) = \frac{1}{2} \underset{\sim}{u}^* C \underset{\sim}{u} + \underline{h}^* \underset{\sim}{u} \qquad (41)$$

where

$$C = G^* G \qquad N \times N \text{ matrix}$$

$$\underline{h} = - G^* \underset{\sim}{d}' \quad N \times 1 \text{ vector}$$

The constraints can be placed in the form

$$D \underset{\sim}{u} - \underline{b} \geq 0$$

To illustrate that problems with amplitude and rate-of-change constraints can be put into this form, let us look at the two-dimensional example. In this case,

$$D = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} -M_1 \\ -M_1 \\ -M_1 \\ -M_1 \\ -M_2 - u(0) \\ -M_2 + u(0) \\ -M_2 \\ -M_2 \end{bmatrix}$$

Returning to the general formulation, we form the Lagrangian

$$\phi(\underset{\sim}{u}, \underline{\lambda}) = Q(\underset{\sim}{u}) - \underline{\lambda}^* (D \underset{\sim}{u} - \underline{b})$$

From the theorems given in Appendix III (Kuhn-Tucker theorems, Reference 17), the task is to find the saddle point of $\phi(\underset{\sim}{u}, \underline{\lambda})$, or solve the following max-min problem.

$$\underset{\underline{\lambda} \geq 0}{\text{Max}} \; \underset{\underset{\sim}{u}}{\text{Min}} \left(\frac{1}{2} \underline{u}^* C \underset{\sim}{u} + \underline{h}^* \underset{\sim}{u} - \underline{\lambda}^* (D \underset{\sim}{u} - \underline{b})\right) \qquad (42)$$

The following is an equivalent problem.

$$\underset{\underline{\lambda} \geq 0}{\text{Min}} - \left[\underset{\underset{\sim}{u}}{\text{Min}} \left(\frac{1}{2} \underset{\sim}{u}^* C \underset{\sim}{u} + \underline{h}^* \underset{\sim}{u} - \underline{\lambda}^* (D \underset{\sim}{u} - \underline{b})\right)\right] \qquad (43)$$

We can differentiate $\phi(u, \lambda)$ with respect to u to solve the first minimum.

$$\underset{\sim}{u} = C^{-1}(D^* \underline{\lambda} - \underline{h}) \tag{44}$$

Upon substituting (44) into (43), we have the following problem. The terms which do not depend on $\underline{\lambda}$ have been left out.

$$\underset{\lambda \geq 0}{\text{Min}} \; [\underline{\lambda}^* \Lambda \underline{\lambda} + Y \underline{\lambda}] \tag{45}$$

where

$$\Lambda = \frac{1}{2} D C^{-1} D^*$$

$$Y = \underline{h}^* C^{-1} D^* - \underline{b}^*$$

Now, the coordinatewise gradient method or Ho's simplified gradient projection method can be used to solve this new problem. Upon determining $\underline{\lambda}$, (44) yields the optimum $\underset{\sim}{u}$. We note that the $\underline{\lambda}$ obtained need not be unique.

3.12 A Compromise Procedure for the Multiple Constraint Case

If the procedure outlined in Section 3.11 is not computationally feasible, then the following compromising procedure can be tried.

A method is proposed which attacks directly the magnitude of the control force and which indirectly constrains the rate-of-change by using a penalty function.

We attack the problem in u-space with the criterion,

$$J = \frac{1}{2} \left|\left| \underset{\sim}{y}_d - \underset{\sim}{y} \right|\right|^2 + \lambda \sum_{j=1}^{N} \left(\frac{u(j) - u(j-1)}{M_2} \right)^{\alpha}$$

where $u(0) = 0$ (or the control used in the previous interval),

α is an even integer (2, 4 etc.)

The larger the value of α the closer will the solution approximate the solution to the original problem. For $\alpha > 2$, the problem is slightly more complicated by the fact that J is no longer quadratic.

With this formulation, the coordinatewise gradiant method or Ho's simplified gradient projection method will apply directly in u-space.

3.13 Between Sample Considerations

Besides the errors at the sampling instants, considerations can be given to the output of the process between sampling instants. Instead of (29), we use

$$\underset{\sim}{\bar{y}} = \bar{G} \underset{\sim}{u} + \underset{\sim}{\bar{y}}^{o}$$

where

$$\overline{G} = \begin{bmatrix} g(1) & 0 & 0 & \cdots & 0 \\ \overline{g}(1) & 0 & 0 & & \\ g(2) & g(1) & 0 & & \\ \overline{g}(2) & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ g(N) & & & & g(1) \\ \overline{g}(N) & \overline{g}(N-1) & \cdots\cdots & & \overline{g}(1) \end{bmatrix} \qquad \overline{\underset{\sim}{y}} = \begin{bmatrix} \underline{y}(1) \\ y(1) \\ y(2) \\ \overline{y}(2) \\ \cdot \\ \cdot \\ \cdot \\ y(N) \\ \overline{y}(N) \end{bmatrix}$$

where

$\overline{y}(j)$ - the output T/2 sec. after $y(j)$

$\overline{g}(j)$ - response to a unit pulse of width T at $(j + \frac{1}{2})$ sec. from the initiation of pulse.

The criterion becomes

$$J = \frac{1}{2} \left|\left|\overline{\underset{\sim}{y}}_d - \overline{\underset{\sim}{y}}\right|\right|^2$$

From here, the procedure is exactly the same as before. If desired, the procedure can be extended to more in-between points.

3.14 A Hybrid Computational Procedure

In this section a method will be proposed which exploits the particular features of the analog and digital computers. As shown in Appendix 3, $\lambda_j > 0$ can be used as a test to determine whether the minimum is on a particular bounding hyperplane. Upon determination of the hyperplanes upon which the minimum lies we can determine the minimum point by projection. The analog computer will be employed for the zero-nonzero determination; while the digital computer will be employed for the projection operation.

To each constraint

$$\sum_i d_{ji} u_i - b_j \geq 0$$

is associated a λ_j. For those inequalities satisfied by the equality we have $\lambda_j > 0$. For those in equalities satisfied by a strict inequality we have $\lambda_j = 0$. We are interested in determining those λ_j which are positive. From Theorem 3 of Appendix 3 we have to satisfy

$$\underset{\sim}{u} = C^{-1} (D^* \underline{\lambda} - \underline{h}) \qquad (46)$$

$$D \underset{\sim}{u} - \underline{b} \geq 0 \qquad (47)$$

$$\underline{\lambda}^{*}\,(D\,\underset{\sim}{u} - \underline{b}) = 0 \tag{48}$$

$$\underline{\lambda} \geq 0 \tag{49}$$

Let us substitute (46) into (47) and (48) eliminating $\underset{\sim}{u}$. We obtain the set

$$D\,C^{-1}\,(D^{*}\,\underline{\lambda} - \underline{h}) - \underline{b} \geq 0$$

$$\underline{\lambda}^{*}\left(D\,C^{-1}\,D^{*}\,\underline{\lambda} - (D\,C^{-1}\,\underline{h} + \underline{b})\right) = 0$$

$$\underline{\lambda} \geq 0$$

Let

$$\underline{w} = D\,C^{-1}\,(D^{*}\,\underline{\lambda} - \underline{h}) - \underline{b} \tag{50}$$

Then, we have the symmetrical set of relations to satisfy.

$$\underline{w} \geq 0$$

$$\underline{\lambda} \geq 0$$

$$< \underline{w}, \underline{\lambda} > = 0 \tag{51}$$

The equation in (51) requires that $w_j = 0$ when $\lambda_j > o$ and $\lambda_j = 0$ when $w_j > 0$.

Instead of using λ_j, we can use w_j to determine whether the optimum point is on a particular hyperplane. The magnitude of w_j gives the distance from the optimum point to the hyperplane, H_j. Therefore, we are interested in those w_j which are zero. As we are interested only in the zero-nonzero aspect, an analog computer with limited accuracy can be employed. If a w_j is close to zero there will be little harm in calling it zero.

Upon determining those w_j's which are zero we collect the corresponding inequalities which are to be satisfied by equalities

$$H_j \;:\; \sum_i d_{ji}\,u_i - b_j = 0 \qquad j = 1, \ldots, q \tag{52}$$

The equations may not necessarily be linearly independent. There is no loss of generality in assuming that $\underline{d}_j$ vectors, which are normal to the hyperplanes, H_j, have unit norm.

To perform the projection it will be convenient to find a point which is common to all of the hyperplanes. Let us write (52) in vector form

$$< \underline{d}_j, \underset{\sim}{u} > - b_j = 0$$

or

$$\bar{D}\,\underset{\sim}{u} - \underline{\bar{b}} = 0$$

where

$$\bar{D} = q \times n \text{ matrix}$$

$$\bar{\underline{b}} = q \times 1 \text{ vector}$$

A point $\underset{\sim}{u}^{\ddagger}$ which is common to all of the hyperplanes in (52) is given by the pseudo-inverse.

$$\underset{\sim}{u}^{\ddagger} = \bar{D}^{\dagger}\, \underline{b} \qquad (53)$$

Before proceeding, we describe the projection operator as described by Rosen (Reference 35). (We extend Rosen's work by employing the pseudo-inverse.) Let us consider the linear subspaces (includes origin) corresponding to the hyperplanes, H_j.

$$\bar{D}\, \underset{\sim}{u} = 0$$

The normals to the subspaces $(\underline{d}_j)$ span the q-dimensional subspace $\tilde{Q}$. The subspace obtained by the intersection of the hyperplanes translated to the origin we designate as Q. Now, the total space consists of the product space of $\tilde{Q}$ and Q, or $E_n = \tilde{Q} \oplus Q$.

Now, the projection of a vector in E_n onto $\tilde{Q}$ is given by

$$\tilde{D}_q = \bar{D}\, \bar{D}^{\dagger}$$

The projection of a vector in E_n onto Q is given by the n x n matrix.

$$D_q = I - \bar{D}\, \bar{D}^{\dagger}$$

Since we are interested in the intersection of hyperplanes translated from the origin, we form the vector from $\underset{\sim}{u}^{\ddagger}$ to the desired point $\underset{\sim}{u}'$, or $\underset{\sim}{u}' - \underset{\sim}{u}^{\ddagger}$. Performing the projection we obtain

$$(I - \bar{D}\, \bar{D}^{\dagger})\, (\underset{\sim}{u}' - \underset{\sim}{u}^{\ddagger})$$

Now, the optimum point is obtained by

$$\underset{\sim}{u}^{o} = (I - \bar{D}\, \bar{D}^{\dagger})\, (\underset{\sim}{u}' - \underset{\sim}{u}^{\ddagger}) + \underset{\sim}{u}^{\ddagger} \qquad (54)$$

The computation of (54) will be performed on a digital computer with the pseudo-inverse subroutine described in Appendix 4. It should be pointed out that the technique is directly applicable when the criterion is given by (33). Otherwise, the gradient vector must be projected in an iterative manner.

The reasons for employing analog computation are: 1) speed of response and 2) minimal accuracy requirements. The implicit function technique does not seem to have a counterpart in digital computation except by using analogous techniques such as DDA. Let us describe the analog circuit requirements by looking at a simple example. Although simple constraints are considered in the example, multiple constraints can be considered without modification of the method. It is not difficult to envision special purpose computers for on-line application.

Example: Find u(1) and u(2) which minimizes

$$J = \left\| \underset{\sim}{d}' - G\, \underset{\sim}{u} \right\|^2 = \underset{\sim}{u}^* G^* G\, \underset{\sim}{u} - 2\, \underset{\sim}{d}'^{\,*} G\, \underset{\sim}{u} = \underset{\sim}{d}'^{\,*} \underset{\sim}{d}'$$

where

$$G = \begin{bmatrix} g(1) & 0 \\ g(2) & g(1) \end{bmatrix}$$

subject to $|u(i)| \leq M \quad i = 1, 2$

The constraints in vector form are

$$D\, \underset{\sim}{u} - \underline{b} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \end{bmatrix} + \begin{bmatrix} M \\ M \\ M \\ M \end{bmatrix} \geq 0$$

In (50)

$$C = G^* G \qquad (2 \times 2)$$

$$\underline{h} = -\, 2\, \underset{\sim}{d}'^{\,*} G \qquad (2 \times 1)$$

Let

$$D\, C^{-1} D^* = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{14} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \sigma_{41} & \cdots & \sigma_{44} \end{bmatrix}$$

$$D\, C^{-1}\, \underline{h} - \underline{b} = \underline{\eta} \qquad (4 \times 1)$$

A schematic for the implicit function method for solving (51) is shown in Figure 22. Only 1 channel is shown. For the two-dimensional example there will be 4 similar channels. In general, a channel is required per constraint. The circuit employs integrators, summers, diodes, and relays.

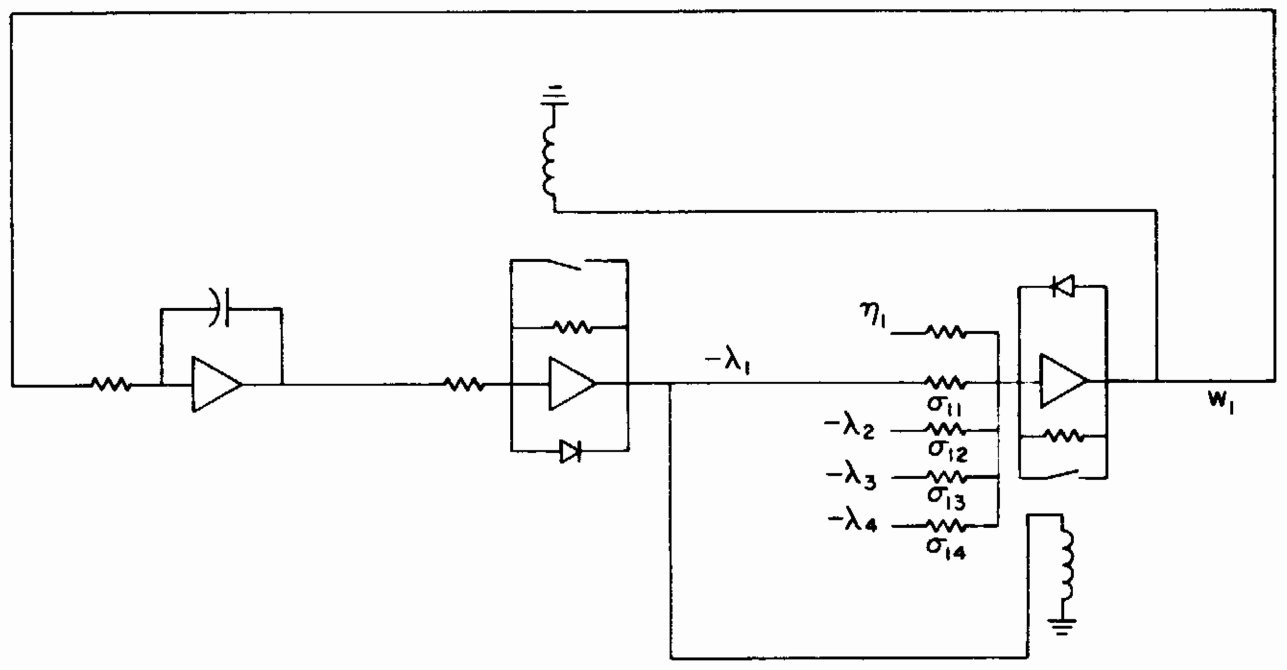


Figure 22. Zero-Non-Zero Determination of w_i on Analog Computer

CHAPTER 4

IDENTIFICATION OF PROCESS PARAMETERS - EXPLICIT MATHEMATICAL RELATION METHOD

4.1 Introduction

Many methods have been proposed for identification (more precisely, parameter estimation) of physical processes. The method to be used in a particular application may depend upon, among other conditions: 1) the manner in which the estimated information is used, and 2) the amount of a priori information available. The methods sought then must fit the control signal synthesis method discussed in Chapter 3. As the identification is to be performed on-line there are requirements on the speed and amount of computation. If a priori information is available the simpler is the identification problem. To have methods which can be readily performed on-line we usually require a certain amount of knowledge about the process.

Our discussion will be restricted to those methods which have the following characteristics. First of all, the process is assumed linear and stationary. The stationarity is assumed for the time interval of the data from which an identification is made. Secondly, the identification should be performed without inserting externally generated test signals. It should depend only on the normal signals present in the system. Lastly, because noise is inevitable in the systems, smoothing should be provided.

For linear processes either the weighting frunction or the coefficients of the difference equation (discrete case) are identified. We confine ourselves to the determination of the coefficients. Discussions on the determination of the weighting function are given by Levin (Reference 36), Kerr and Surber (Reference 37), Balakrishnan (Reference 38), and Hsieh (Reference 28).

Restricting ourselves to the determination of the coefficients of the difference equation, essentially two different approaches are available: 1) the explicit mathematical relation method, and 2) the learning model method. The explicit mathematical relation method requires knowledge of the exact form of the difference equation. This restriction is somewhat relaxed for the learning model method in the sense that a lower order model can be made to approximate a higher order process. This chapter will discuss the explicit mathematical relation method. Chapter 5 will discuss the learning model method.

The explicit mathematical relation method was used by Kalman (Reference 1) but the basic philosophy dates as far back as 1951 when Greenberg (Reference 39) discussed methods for determining stability derivatives of an airplane. Subsequent work on this method was performed by Bigelow and Ruge (Reference 40). The method will be generalized by bringing in the concept of the pseudo-inverse. Furthermore, statistical analysis has been lacking in the previous studies on this particular method. Therefore, statistical considerations will be given in terms of the confidence interval.

In accordance with considerations given in Chapter 2, the explicit mathematical relation method does not rely on the exact knowledge of the state variables.

A thorough survey of identification methods is provided in a report by Eykhoff (Reference 41).

4.2 Description of the Mathematical Relation Method

Briefly, the method reconstructs the equation of the process by measuring the output and input, and their previous values (sufficiently enough so that all of the terms in the equation are accounted for). By taking redundant measurements filtering is provided. Additional filtering can also be obtained by inserting filters (this can be done without sacrificing the identification process).

The method can best be described by taking an example. Let us determine the coefficients of the following difference equation.

$$y(k) = \alpha_1 \, y(k-1) + \alpha_2 \, u\,(k) \tag{55}$$

The problem is to determine α_1 and α_2. These parameters can be constant but unknown or changing due to changes in environment. Usually, $y(k)$ will not be directly observed but with a contaminating noise quantity as depicted in Figure 49. Thus,

$$z(k) = y(k) + v(k) \tag{56}$$

The values of $z(k)$ and $u(k)$ will be stored for some interval of time into the past; and throughout this interval the parameters α_1 and α_2 are assumed to be constant. Since $y(k)$ cannot be directly measured, (55) is rewritten in terms of $z(k)$.

$$z(k) - v(k) = \alpha_1 \left[z(k-1) - v(k-1) \right] + \alpha_2 \, u(k) \tag{57}$$

or

$$z(k) = \alpha_1 \, z(k-1) + \alpha_2 \, u(k) + v_1(k) \tag{58}$$

where

$$v_1(k) = v(k) - \alpha_1 \, v(k-1)$$

Taking a set of measurements, (58) can be rewritten in vector form.

$$\underset{\sim}{z}_k = \alpha_1 \underset{\sim}{z}_{k-1} + \alpha_2 \underset{\sim}{u}_k + \underset{\sim}{v}_{1k} \tag{59}$$ †

where

† The k signifies that N data points into the past from time k are considered.

$$\underset{\sim}{z}_k = \begin{bmatrix} z(k-N+1) \\ \cdot \\ \cdot \\ \cdot \\ z(k) \end{bmatrix}, \text{ etc.}$$

In matrix form

$$\underset{\sim}{z}_k = A\,\underline{\alpha} + \underset{\sim}{v}_{1k} \tag{60}$$

where

$$A = \left[\, \underset{\sim}{z}(k-1) \;\vdots\; \underset{\sim}{u}(k) \,\right]$$

Let

$$\check{\underset{\sim}{z}}_k = A\,\underline{\alpha} \tag{61}$$

The $\check{\underset{\sim}{z}}_k$ is in the manifold of $\underset{\sim}{z}_{k-1}$ and $\underset{\sim}{u}_k$. The quantity $\underset{\sim}{z}_k$ is not necessarily in the linear manifold because of $\underset{\sim}{v}_{1k}$. Since $\underset{\sim}{v}_{1k}$ is unknown, a reasonable estimate of the parameters would be those values which result from the projection of $\underset{\sim}{z}_k$ on the manifold of $\underset{\sim}{z}_{k-1}$ and $\underset{\sim}{u}_k$. The projection yields

$$\begin{aligned} \langle \underset{\sim}{z}_k - \check{\underset{\sim}{z}}_k, \underset{\sim}{z}_{k-1} \rangle &= 0 \\ \langle \underset{\sim}{z}_k - \check{\underset{\sim}{z}}_k, \underset{\sim}{u}_k \rangle &= 0 \end{aligned}$$

or,

$$\begin{aligned} \alpha_1 \langle \underset{\sim}{z}_{k-1}, \underset{\sim}{z}_{k-1} \rangle + \alpha_2 \langle \underset{\sim}{u}_k, \underset{\sim}{z}_{k-1} \rangle &= \langle \underset{\sim}{z}_k, \underset{\sim}{z}_{k-1} \rangle \\ \alpha_1 \langle \underset{\sim}{z}_{k-1}, \underset{\sim}{u}_k \rangle + \alpha_2 \langle \underset{\sim}{u}_k, \underset{\sim}{u}_k \rangle &= \langle \underset{\sim}{z}_k, \underset{\sim}{u}_k \rangle \end{aligned} \tag{62}$$

In terms of the matrix equation, (62) is

$$A^* A\,\underline{\alpha} = A^* \underset{\sim}{z}_k \tag{63}$$

Equations (62) and (63) are known as normal equations, and if $\underset{\sim}{z}_{k-1}$ and $\underset{\sim}{u}_k$ are linearly independent, then the solution is given by

$$\underline{\hat{\alpha}} = (A^* A)^{-1} A^* \underset{\sim}{z}_k \tag{64}$$

If $\underset{\sim}{z}_{k-1}$ and $\underset{\sim}{u}_k$ are not necessarily linearly independent, (64) can be generalized to

$$\underline{\hat{\alpha}} = A^{\dagger} \underset{\sim}{z}_k \tag{65}$$

The pseudo-inverse, extensively discussed by Penrose (References 18, 19) provides a unique solution even if the inverse in (64) does not exist. It provides the solution with Min $||\underline{\alpha}||$. It should be noted that the minimum norm solution may not be the actual values of the process parameters.

However, a solution is provided to the problem formulation instead of some nonsensical solution. A recursive method of evaluating the pseudo-inverse is presented in Appendix 3 essentially following the derivation given by Greville (Reference 15). It is rederived starting with the axioms given by Penrose. The relation of Greville's routine with Kalman's recursive filtering technique (Reference 16) is given in Appendix 4.

During the first few steps of the recursive procedure we always have a singular situation. The advantage of Greville's procedure is that a unique solution is provided even for these first few steps; and eventually as the nonsingular situation is reached the solution is obtained without error.

4.3 Additional Filtering

In conjunction with the use of redundant data, it is possible to incorporate additional filtering. This filtering should be provided without compromising the identification process. Let us describe this filtering process on the same example. We designate F( ) as a linear discrete filter and operate on both sides of (58).

$$F\Big(z(k)\Big) = \alpha_1\, F\Big(z(k-1)\Big) + \alpha_2\, F\Big(u(k)\Big) + v_2(k) \qquad (66)$$

Now, the quantities $F\Big(z(k)\Big)$ and $F\Big(z(k-1)\Big)$ are respectively closer to y(k) and y(k-1). Therefore, we have in vector form

$$\underset{\sim}{f}_k = \alpha_1 \underset{\sim}{f}_{k-1} + \alpha_2\, \underset{\sim}{fu}_k + \underset{\sim}{v}_{2k} \qquad (67)$$

where

$$\underset{\sim}{f}_k = \begin{bmatrix} F\Big(z(k-N+1)\Big) \\ \cdot \\ \cdot \\ \cdot \\ F\Big(z(k)\Big) \end{bmatrix}, \text{ etc.}$$

The identification configuration will appear as in Figure 23.

4.4 Block Processing of Data

The Greville-Kalman recursive method can process the data as it arrives. However, there is one difficulty. In an adaptive task in which the process is changing it becomes necessary to lop off the effect of old data. Of course, in an adaptive task in which the process is unknown but constant, there is no problem because the recursive method can start at time t = 0 and continue up to the present time. A possible solution to the former case is block processing depicted in Figure 24. The recursive method is initiated at the start of each observation interval.

This, of course, is simple minded. The estimated values are changed every NT seconds. If parameters are changing continually this procedure may not be satisfactory.

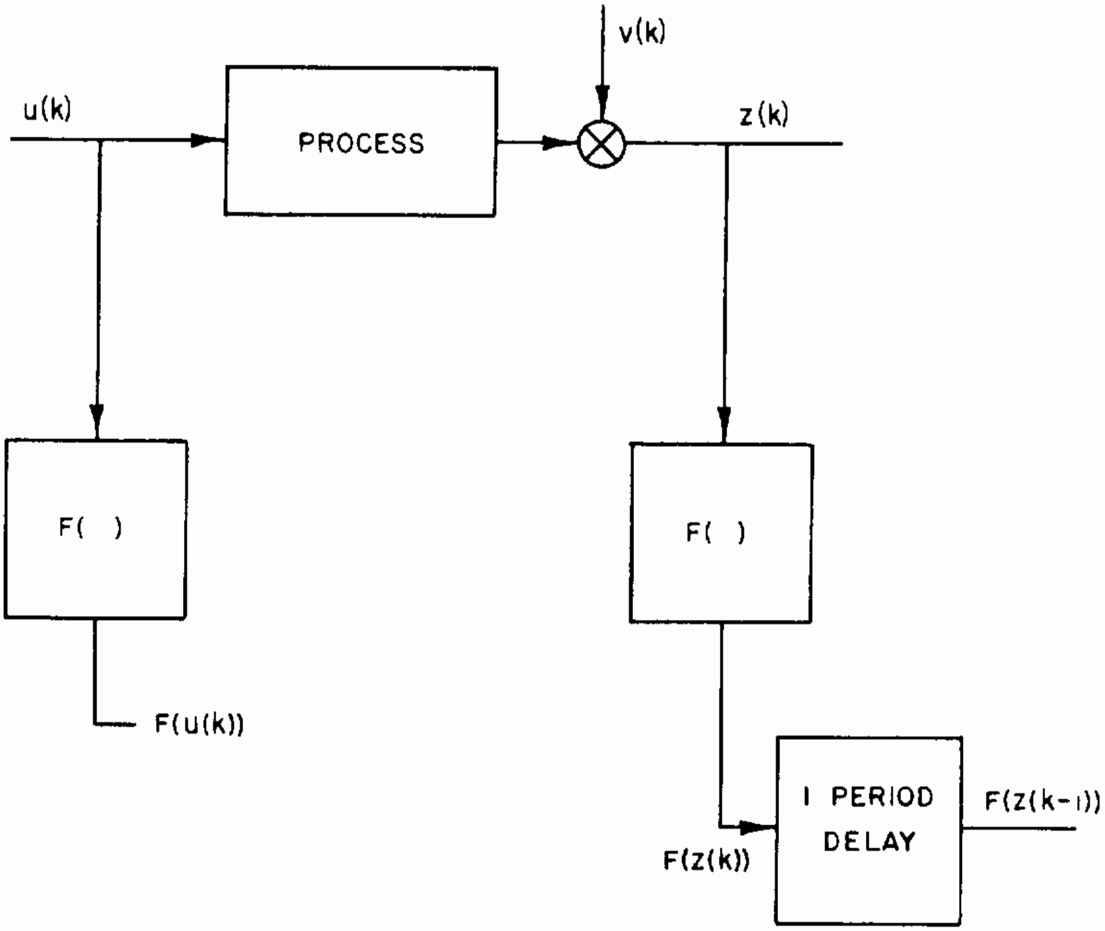


Figure 23. Configuration for Additional Filtering

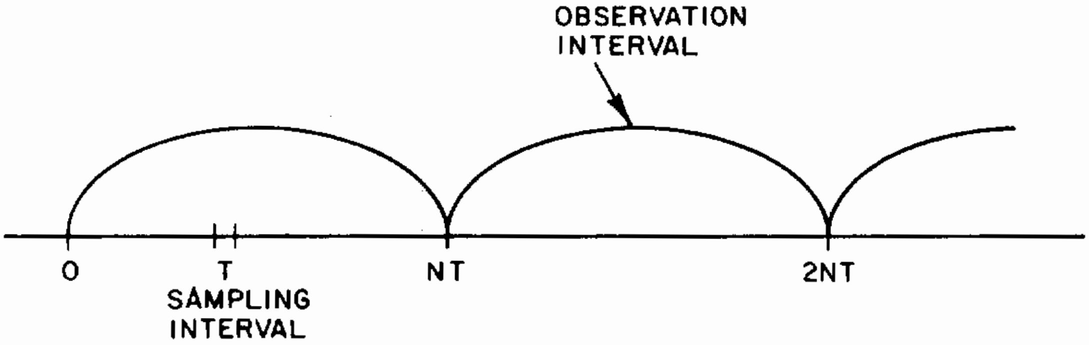


Figure 24. Block Processing

4.5 Exponential Weighting

The lopping off of old data can be provided by exponential weighting. This weighting can be incorporated into the recursive method previously described by determining the solution to

$$W_k A_k \underline{\alpha}_k \doteq W_k \underset{\sim}{z}_k \tag{68}$$ [§]

The dot above the equal sign signifies that the α's are to be chosen so that the left-hand side best approximates the right-hand side in the sense that we have

$$\text{Min} \left\| W_k A_k \underline{\alpha}_k - W_k \underset{\sim}{z}_k \right\|^2$$

The W_k is equal to

$$W_k = W_k^* = \begin{bmatrix} \sqrt{w^k} & 0 & & 0 \\ 0 & \cdot & & \\ & \cdot & & \\ & \cdot & \sqrt{w^2} & 0 \\ 0 & & 0 & \sqrt{w} \end{bmatrix}$$

with $0 \leq w \leq 1$ and w is the staleness factor.

The solution is given by

$$\hat{\underline{\alpha}}_k = A_k^\dagger W_k \underset{\sim}{z}_k \tag{69}$$

where $A_k^\dagger$ is the pseudo-inverse of $W_k A_k$.

It is observed that

$$\left[\begin{array}{c} \sqrt{w}\ W_{k-1} A_{k-1} \\ \hline \sqrt{w}\ \underline{a}_k^* \end{array}\right] \underline{\alpha}_k = \left[\begin{array}{c|c} \sqrt{w}\ W_{k-1} & 0 \\ \hline 0 & \sqrt{w} \end{array}\right] \underset{\sim}{z}_k \tag{70}$$

The recursive equations can be derived in the same way as in Appendix 3. The important equations are

$$\hat{\underline{\alpha}}_k = \hat{\underline{\alpha}}_{k-1} - \sqrt{w}\, \underline{b}_k \underline{a}_k^* \hat{\underline{\alpha}}_{k-1} + \sqrt{w}\, \underline{b}_k\ z(k) \tag{71}$$

$$\underline{c}_k^* = \sqrt{w}\, \underline{a}_k^* - \sqrt{w}\, \underline{a}_k^* A_{k-1}^\dagger W_{k-1} A_{k-1} \tag{72}$$

[§] The subscript k again refers to the present time.

Case 1: $c_k \neq 0$

$$\underline{b}_k = \left(\underline{c}_k^* \underline{c}_k\right)^{-1} \underline{c}_k \tag{73}$$

Case 2: $c_k = 0$

$$\underline{b}_k = \left(\sqrt{w} + (\sqrt{w})\, \underline{a}_k^* A_{k-1}^{\dagger} W_{k-1} A_{k-1}^{\dagger^*} \underline{a}_k\right)^{-1} A_{k-1}^{\dagger} W_{k-1} A_{k-1}^{\dagger^*} \underline{a}_k \tag{74}$$

$$A_k^{\dagger} W_k A_k = A_{k-1}^{\dagger} W_{k-1} A_{k-1} + \underline{b}_k \underline{c}_k^* \tag{75}$$

$$A_k^{\dagger} W_k A_k^{\dagger^*} = \left[\sqrt{w^{-1}} - \underline{b}_k \underline{a}_k^*\right] A_{k-1}^{\dagger} W_{k-1} A_{k-1}^{\dagger^*} \left[\sqrt{w^{-1}} - \underline{a}_k \underline{b}_k^*\right] + \sqrt{w}\, \underline{b}_k \underline{b}_k^* \tag{76}$$

The exponential weighting is depicted in Figure 25. The point k represents the present time. Recent data are given larger weights than older data. As $w \rightarrow 1$, these equations will revert to the growing memory case of Appendix 4.

4.6 Uniform Weighting - Observable Case Only

If adequate computer storage space is provided, at any sampling instant a finite amount of data into the past can be analyzed. The recursive equations for this uniform weighting have been worked out by Gainer (Reference 42) for the observable case $\left((A^* A)^{-1} \text{ exists}\right)$. The procedure will be outlined in this section. Pictorially, the uniform weighting slides forward in time as depicted in Figure 26.

For adding the effect of new data, $\underline{\hat{\alpha}}_{k,N}$ was determined in terms of $\underline{\hat{\alpha}}_{k-1,N-1}$ and the new set of data $\underline{a}_k$, $z(k)$.[§] This time, the set of data to be deleted $\left(\underline{a}_{k-N}, z(k-N)\right)$ and $\underline{\hat{\alpha}}_{k,N}$ are given, and it is desired to determine $\underline{\hat{\alpha}}_{k,N-1}$. From (64)

$$\underline{\hat{\alpha}}_{k,N} = P_{k,N} A_{k,N}^* \underline{z}_{k,N} \tag{77}$$

where

$$P_{k,N} = \left(A_{k,N}^* A_{k,N}\right)^{-1} \tag{78}$$

or, (the subscript k is dropped when unambiguous)

[§] In $\underline{\alpha}_{k,N}$ the N signifies that N data points are taken and k signifies time. This notation is adopted primarily for this section.

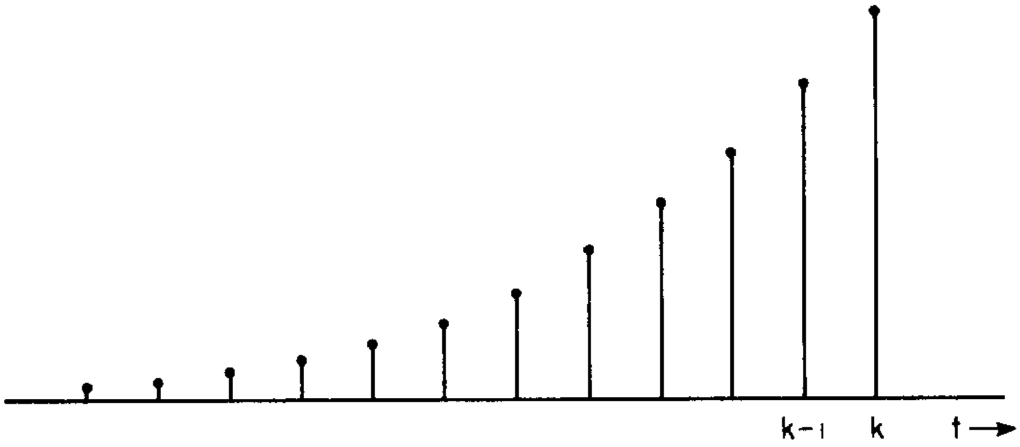


Figure 25. Exponential Weighting

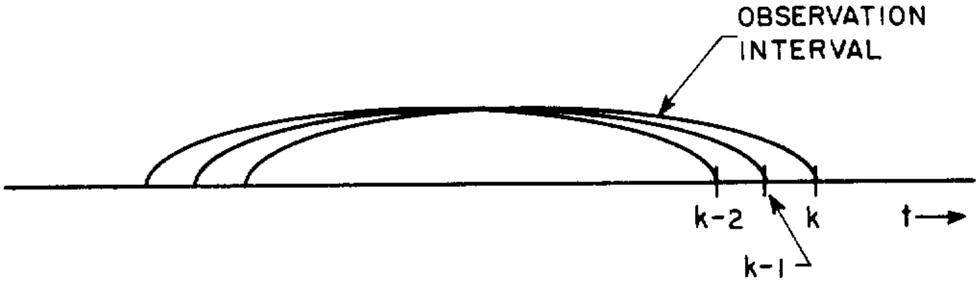


Figure 26. Uniform Weighting

$$\underline{\hat{\alpha}}_N = P_N \left[\underline{a}_{k-N} \;\vdots\; A^*_{N-1} \right] \begin{bmatrix} z(k-N) \\ - - - \\ \underset{\sim}{z}_{N-1} \end{bmatrix}$$

$$= P_N \underline{a}_{k-N} z(k-N) + P_N A^*_{N-1} \underset{\sim}{z}_{N-1} \tag{79}$$

Also,

$$\underline{\hat{\alpha}}_{N-1} = P_{N-1} A^*_{N-1} \underset{\sim}{z}_{N-1} \tag{80}$$

We note that

$$P_N^{-1} = \underline{a}_{k-N} \underline{a}^*_{k-N} + A^*_{N-1} A_{N-1} = \underline{a}_{k-N} \underline{a}^*_{k-N} + P_{N-1}^{-1} \tag{81}$$

Therefore

$$\underline{\hat{\alpha}}_{N-1} = \left[P_N^{-1} - \underline{a}_{k-N} \underline{a}_{k-N} \right]^{-1} A^*_{N-1} \underset{\sim}{z}_{N-1} \tag{82}$$

or,

$$A^*_{N-1} \underset{\sim}{z}_{N-1} = \left[P_N^{-1} - \underline{a}_{k-N} \underline{a}^*_{k-N} \right] \underline{\hat{\alpha}}_{N-1} \tag{83}$$

Substituting (83) into (79), we have

$$\underline{\hat{\alpha}}_N = \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right] \underline{\hat{\alpha}}_{N-1} + P_N \underline{a}_{k-N} z(k-N)$$

or

$$\underline{\hat{\alpha}}_{N-1} = \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} \left(\underline{\hat{\alpha}}_N - P_N \underline{a}_{k-N} z(k-N) \right) \tag{84}$$

We can eliminate the inverse by noting the following

$$\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right] = I$$

$$\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} - \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} P_N \underline{a}_{k-N} \underline{a}^*_{k-N} = I \tag{85}$$

Post multiply by $P_N \underline{a}_{k-N}$

$$\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} P_N \underline{a}_{k-N} - \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \right]^{-1} P_N \underline{a}_{k-N} \beta$$

$$= P_N \underline{a}_{k-N}$$

where

$$\beta = \underline{a}^*_{k-N} P_N \underline{a}_{k-N} \qquad \text{(scalar)}$$

Therefore,

$$\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N}\right]^{-1} P_N \underline{a}_{k-N} = \frac{1}{1-\beta} P_N \underline{a}_{k-N} \tag{86}$$

Post multiply (85) by $\hat{\underline{\alpha}}_N$

$$\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N}\right]^{-1} \hat{\underline{\alpha}}_N = \hat{\underline{\alpha}}_N + \frac{1}{1+\beta} P_N \underline{a}_{k-N} \underline{a}^*_{k-N} \hat{\underline{\alpha}}_N \tag{87}$$

Substituting (86) and (87) into (84), we have

$$\hat{\underline{\alpha}}_{N-1} = \hat{\underline{\alpha}}_N + \frac{1}{1+\beta} P_N \underline{a}_{k-N} \left(\underline{a}^*_{k-N} \hat{\underline{\alpha}}_N - z(k-N)\right) \tag{88}$$

Now, P_{N-1} will be derived in terms of P_N. From (81)

$$P_{N-1}^{-1} = P_N^{-1}\left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N}\right]$$

or,

$$P_{N-1} = \left[I - P_N \underline{a}_{k-N} \underline{a}^*_{k-N}\right]^{-1} P_N$$

To eliminate the inverse, we post multiply (85) by P_N. Therefore,

$$P_{N-1} = P_N + \frac{1}{1-\beta} P_N \underline{a}_{k-N} \underline{a}^*_{k-N} P_N \tag{89}$$

Equations (88) and (89) are to be used with the recursive equations of Appendix 4 to perform uniform weighting. A sequence of add, delete, add, delete, add, ... alternatingly using the above equations for the oldest data and equations of Appendix 4 for the new data is required. It is noted that $1-\beta$ may be equal to zero. When such a situation arises, the elimination of that particular row of data can be deferred, of course, with attendant increase in programming complexity.

4.7 Confidence Interval

The determination of the accuracy with which parameters can be estimated requires statistical analysis. An extensive study in the area of least squares has been made by Linnik (Reference 43). The particular results which are useful for our purposes will be presented here.

Let us refer to Figure 49 and consider the case when $v(k)$ is a sequence of uncorrelated Gaussian random variables. As $v_1(k)$ is a function of $v(k)$ and $v(k-1)$ in (58) and if we consider the data points at every other sampling interval, $v_1(k)$ would be an uncorrelated sequence of noise. Therefore, our samples are taken so that we consider the white noise case. Of course, one would do better to consider every data point even if they are correlated. However, the white noise case is more convenient for the determination of confidence intervals and it will provide a conservative determination.

We will consider the case when the variance (σ^2) of $v_1(k)$ is unknown. It is observed that even if the variance of $v(k)$ is known, the variance of $v_1(k)$ is unknown because $v_1(k)$ is a function of the parameters to be determined.

First, let us discuss the properties of the optimum estimate, $\hat{\underline{\alpha}}$. We state the significant properties as lemmas. The proofs can be found in Linnik (Reference 43).

Lemma 4.1: The estimators from the least squares analysis are unbiased, i.e.,

$$E\,\hat{\underline{\alpha}} = \underline{\alpha}$$

Lemma 4.2: The unbiased estimators, $\hat{\underline{\alpha}}$, form a Gaussian, n-dimensional vector with the correlation matrix.

$$R_{\hat{\underline{\alpha}}} = \sigma^2 (A^* A)^{-1}$$

or

$$\mathrm{Var}\, \hat{\alpha}_i = \sigma^2 \left\{ (A^* A)^{-1} \right\}_{ii}$$

Therefore,

$$\frac{\hat{\alpha}_i - \alpha_i}{\sigma \sqrt{\left\{ (A^* A)^{-1} \right\}_{ii}}} \in N(0, 1)$$

where N(0, 1) represents Gaussian distribution with zero mean and standard deviation of one.

Next, we consider the properties of $\hat{\underline{v}}$, given by

$$\hat{\underline{v}} = A\, \hat{\underline{\alpha}} - \underset{\sim}{z}$$

Lemma 4.3: The minimum variance unbiased estimator also satisfies the condition

$$\| \hat{\underline{v}} \|^2 = \min$$

Lemma 4.4: The error vector, $\hat{\underline{v}}$, is an (N-n) dimensional Gaussian vector and it is independent of $\hat{\underline{\alpha}}$.

Lemma 4.5: The random variable $\hat{\underline{v}}^* \hat{\underline{v}}$ is distributed as χ^2 with N-n degrees of freedom and it is independent of $\hat{\underline{\alpha}}$.

Now, we have the quantities which can form the t-distribution. If ξ and $\Sigma\, \xi_i^2$ are statistically independent Gaussian random variables with the latter having n' degrees of freedom, the t-distribution is formed by the following ratio.

$$t = \frac{\xi}{\sqrt{\frac{1}{n'} \Sigma\, \xi_i^2}} = \frac{\xi}{\sqrt{\frac{\chi^2}{n'}}}$$

Let

$$\xi = \frac{\hat{\alpha}_i - \alpha_i}{\sigma \sqrt{\left\{(A^* A)^{-1}\right\}_{ii}}}$$

$$\chi^2 = \frac{1}{\sigma^2} \underline{\hat{v}}^* \underline{\hat{v}}$$

$$n' = N-n$$

then,

$$t_{N-n} = \frac{\hat{\alpha}_i - \alpha_i}{\sqrt{\left\{(A^* A)^{-1}\right\}_{ii} \frac{\underline{\hat{v}}^* \underline{\hat{v}}}{N-n}}}$$

It is observed that the unknown variance cancels when the ratio is formed.

Using the t-distribution we can determine the interval about $\hat{\alpha}_i$ which will include α_i with a certain probability. For example, let us use Pr. = .90; then,

$$\Pr \left\{ \left| t_{N-n} \right| \leq \gamma \right\} = .90$$

The γ is found from well-tabulated tables. Therefore,

$$\left| \hat{\alpha}_i - \alpha_i \right| = \gamma \sqrt{\left\{(A^* A)^{-1}\right\}_{ii} \frac{\underline{\hat{v}}^* \underline{\hat{v}}}{N-n}}$$

Thus, the range 2Δ of the .90 confidence interval is

$$2\Delta = \left[\hat{\alpha}_i \pm \gamma \sqrt{(A^* A)^{-1}_{\ ii} \frac{\underline{\hat{v}}^* \underline{\hat{v}}}{N-n}} \right] \tag{90}$$

The difficulty in the use of the confidence interval lies in the fact that $A^* A$ and $\underline{\hat{v}}^* \underline{\hat{v}}$ change as the interval of consideration changes. Possibly one could use the conservative (larger) estimate of these quantities to get an estimate of 2Δ. The important point to observe is that to decrease 2Δ, N-n must be increased.

The above results can be extended to the case when exponential weighting is used. The range 2Δ is then given by

$$2\Delta = \left[\hat{\alpha}_i \pm \gamma \sqrt{\left\{\left(A^* W^2 A\right)^{-1}\right\}_{ii} \frac{\underline{\hat{v}}^* W^2 \underline{\hat{v}}}{N-n}} \right] \tag{91}$$

4.8 Determination of Pulse Response

In the type of adaptive controller studied in Chapter 3, the elements of the pulse response are desired along with the coefficients of the difference equation. However, the pulse response and the coefficients of the difference equations are closely related; and two methods are available for determining the pulse response.

First, there is the well-known method of deriving the pulse response from the coefficient via long division. Although it is relatively simple to perform the calculations, there may be uncertainty in the propagation of errors through the division process.

In the other method, the pulse response coefficients can be measured directly. Let us first look at difference equations which have only a single forcing term. In this case, the states can simply be chosen as x(k), x(k-1), x(k-2), etc. The second order example has the form

$$\underline{x}(k) = \Phi\, \underline{x}(k-1) + \underline{\gamma}\, u(k) \qquad (92)$$

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

where

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

$$\underline{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_1(k-1) \end{bmatrix}$$

The equations are

$$z(k) - v(k) = \phi_{11}\Big(z(k-1) - v(k-1)\Big) + \phi_{21}\Big(z(k-2) - v(k-2)\Big) + g(1)\, u(k)$$

or

$$\underset{\sim}{z}_k = \phi_{11}\, \underset{\sim}{z}_{k-1} + \phi_{21}\, \underset{\sim}{z}_{k-2} + g(1)\, \underset{\sim}{u}_k + \underset{\sim}{v}_{1k}$$

where

$$\underset{\sim}{z}_k^{*} = \Big(z(k-N+1), \ldots, z(k)\Big) \qquad \text{(N samples)}$$

For the state variables chosen, the above equations apply to the case when there is only a single forcing term. From this expression for $\underset{\sim}{z}_k$, the pulse response at the end of one sampling interval, g(1), can be determined along with estimates of ϕ_{11} and ϕ_{12}. The least-squares procedure is again used.

In order to obtain g(2), we need an equation for $\underline{x}(k)$ in terms of x(k-2). From

$$\underline{x}(k-1) = \Phi\, \underline{x}(k-2) + \underline{\gamma}\, u(k-1)$$

we obtain

$$\underline{x}(k) = \Phi^2\, \underline{x}(k-2) + \Phi\, \underline{\gamma}\, u(k-1) + \underline{\gamma}\, u(k) \qquad (93)$$

Therefore,

$$\underset{\sim}{z}_N = \phi_{11}^{(2)}\, \underset{\sim}{z}_{N-2} + \phi_{12}^{(2)}\, \underset{\sim}{z}_{N-3} + g(2)\, \underset{\sim}{u}_{N-1} + g(1)\, \underset{\sim}{u}_N + \underset{\sim}{v}_{2N}$$

where

$$\Phi^{2} = \Phi\ \Phi = \begin{bmatrix} \phi_{11}^{(2)} & \phi_{12}^{(2)} \\ \\ \phi_{21}^{(2)} & \phi_{22}^{(2)} \end{bmatrix}$$

From the above expression, g(1) and g(2) can be formed along with $\phi_{11}^{(2)}$ and $\phi_{12}^{(2)}$. If more elements of the pulse response are desired, the above procedure is repeated. The pattern is now, however, familiar. For example, if g(1) to g(4) are desired, the following equation would be used.

$$\underset{\sim}{z}_N = \phi_{11}^{(4)} \underset{\sim}{z}_{N-4} + \phi_{12}^{(4)} \underset{\sim}{z}_{N-5} + g(4)\, \underset{\sim}{u}_{N-3} + g(3)\, \underset{\sim}{u}_{N-2}$$

$$+ g(2)\, \underset{\sim}{u}_{N-1} + g(1)\, \underset{\sim}{u}_N + \underset{\sim}{v}_{4N}$$

Although the procedure requires larger equations, the advantage in using this method is that the unknown coefficients are determined directly.

In the case where there is more than one forcing term, the above procedure can be used but with a little more difficulty. There are two alternatives. First, if $x_1(k)$, $x_1(k-1)$, etc., are used as state variables the problem can be treated as a multiple control input problem. The second approach is to use a different set of state variables so that the single difference equation can be put into the form of (92). The procedure will be briefly illustrated.

Let us look at the example given by

$$x(k) + \alpha_1 x(k-1) + \alpha_2 x(k-2) = \beta_1 u(k) + \beta_2 u(k-1)$$

Let

$$x_1(k) = x(k)$$

$$x_2(k-1) = \alpha_2\ x_1(k-2) - \beta_2\ u(k-1)$$

then

$$x_1(k) = -\alpha_1\ x_1(k-1) - x_2(k-1) + \beta_1\ u(k)$$

$$x_2(k) = \quad \alpha_2\ x_1(k-1) \qquad\qquad - \beta_2\ u(k)$$

The equations are now in the form of (92). Let us see what is involved if we desire g(1) and g(2). The top row of the vector equation, (93), is

$$x_1(k) = \phi_{11}^{(2)}\ x_1(k-2) + \phi_{12}^{(2)}\ x_2(k-2) + g(2)\ u(k-1) + g(1)\ u(k)$$

In terms of the measured quantities we have

$$\underset{\sim}{z}_k = \phi_{11}^{(2)} \underset{\sim}{z}_{k-2} + \phi_{12}^{(2)} \alpha_2 \underset{\sim}{z}_{k-3} - \phi_{12}^{(2)} \beta_2 \underset{\sim}{u}_{k-2}$$

$$+ g(2) \underset{\sim}{u}_{k-1} + g(1) \underset{\sim}{u}_k + \text{noise}$$

Along with g(1) and g(2) other coefficients are determined.

Although this method requires more manipulations, it gives the required coefficients directly.

CHAPTER 5

IDENTIFICATION OF PROCESS PARAMETERS - LEARNING MODEL METHOD

5.1 Introduction

The other approach available for estimation of coefficients of a difference equation is the learning model method. It is felt that if some a priori estimate of the unknown parameters is available then we should be able to use this information to advantage. This is probably the motivation for the learning model method. This method was originally studied by Margolis (Reference 44) using the sensitivity function. The sensitivity function is also used by Staffanson (Reference 45) who was concerned with parameter determination from flight test data. Several characteristics are apparent in Margolis' work.

1. One is constantly worried about the stability problem.
2. Noise considerations were not given.
3. One must choose the gain in the steepest descent procedure.
4. The use of sensitivity functions is generally valid for small regions about a trial point.

To overcome some of the above problem areas, this chapter will give an alternative procedure primarily patterned after Newton's method but with the extensive use of the digital computer to give assurance of monotone convergence. Newton's method is chosen because it is known for its rapid rate of convergence. By considering blocks of data at a time, smoothing is performed. We will first briefly describe Margolis' approach through an example so that it will provide a basis for comparison. Again, we restrict ourselves to the discrete case.

Two other possibilities for performing the learning model method should be mentioned. The first is the quasi-linearization approach described by Bellman, et al (Reference 46). This method was found to be very cumbersome for the discrete case. The other method is the orthogonal function approach used by Elkind, et al (Reference 47). Fixing the model time constants a priori seems to be a crude method.

5.2 Margolis' Sensitivity Function Approach

Margolis' learning model approach is shown in Figure 27. Margolis used the error-squared as the criterion. Integrals of error-squared led to stability problems. Even though Margolis may have had success in many situations for the continuous case, the discrete case may lead to other conclusions. Therefore, we will look at the discrete case. The procedure will be described here with the results given later.

Let us choose to discuss the first order process with two unknown parameters α_1 and α_2.

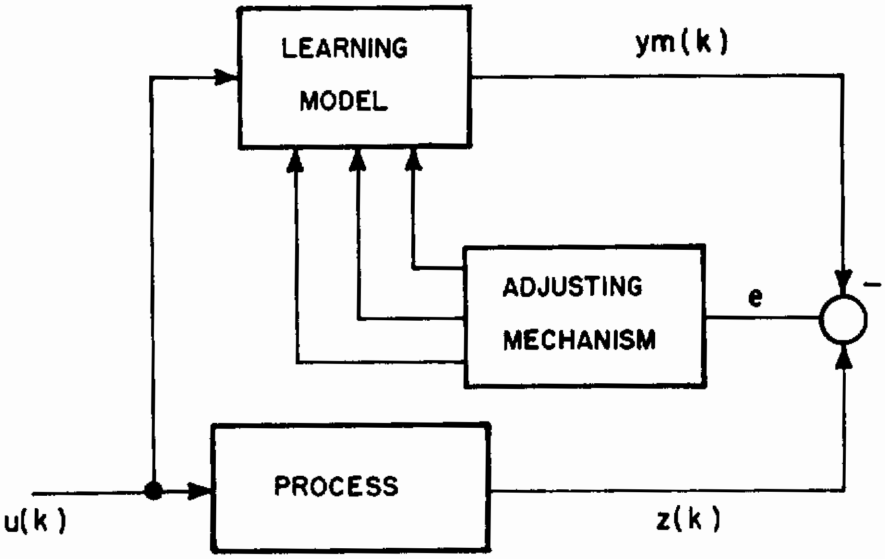


Figure 27. Margolis' Learning Model Approach

$$y(k) = \alpha_1\, y(k-1) + \alpha_2\, u(k) \tag{94}$$

$$z(k) = y(k) + v(k)$$

The equation for the model is given by

$$ym(k) = a_1\, ym(k-1) + a_2\, u(k) \tag{95}$$

The coefficients a_1 and a_2 are to be adjusted to minimize

$$J = \Big(z(k) - ym(k)\Big)^2 \tag{96}$$

We take the gradient of J with respect to a_1 and a_2.

$$\frac{\partial J}{\partial a_1} = -\,2\,\Big(z(k) - ym(k)\Big)\, u_1(k) \tag{97}$$

$$\frac{\partial J}{\partial a_2} = -\,2\,\Big(z(k) - ym(k)\Big)\, u_2(k) \tag{98}$$

where

$$u_1(k) = \frac{\partial ym(k)}{\partial a_1}$$

$$u_2(k) = \frac{\partial ym(k)}{\partial a_2}$$

The $u_1(k)$ and $u_2(k)$ are called sensitivity functions and they are determined from equations obtained by differentiating (95) with respect to the parameters. Therefore,

$$u_1(k) = a_1\, u_1(k-1) + ym(k-1) \tag{99}$$

$$u_2(k) = a_1\, u_2(k-1) + u(k) \tag{100}$$

The corrections on the parameters a_1 and a_2 are taken in the direction of steepest descent.

$$a_1(k+1) = a_1(k) - 2K\Big(z(k) - ym(k)\Big)\, u_1(k) \tag{101}$$

$$a_2(k+1) = a_2(k) - 2K\Big(z(k) - ym(k)\Big)\, u_2(k) \tag{102}$$

where K is the gain in the steepest descent procedure. The K is to be chosen from stability and noise considerations.

5.3 Modified Newton's Approach

We next describe a method which will be extensively studied in this chapter. Again we will use an example to illustrate the procedure.

Instead of operating on the error as shown in Figure 27, the stability problem can possibly be alleviated by solving the following problem:

Problem 5.1: Find the parameters (a_i) of the model which minimizes

$$J = \sum_{j=1}^{N} \Big(z(j) - ym(j) \Big)^2 \qquad (103)$$

where $ym(j)$ is subject to the dynamical constraint

$$ym(j) = a_1\ ym(j-1) + a_2\ u(j) \qquad (104)$$

The time indices are shown in Figure 28.[§] In our case, the model (104) could be of lower order than the actual process (model fitting problem). We start from an initial trial or estimate of the parameters, $a_i^{(1)}$, and the initial conditions for the interval of observation, $ym(0)^{(1)}$. With these initial trials (104) is solved to obtain a nominal solution, $ym(j)^{(1)}$, $j = 0, 1, \ldots, N$. Next, the perturbation equations of (104) are written, evaluated along the nominal $ym(j)^{(1)}$.

$$\delta ym(j) = a_1^{(1)}\ \delta ym(j-1) + ym^{(1)}\ (j-1)\ \delta a_1(j-1) + u(k)\ \delta a_2(j-1) \qquad (105)$$

We adjoin to (105) other equations which maintain the parameters constant. This trick was used by Bellman, et al (Reference 46).

$$\begin{aligned} \delta a_1(j) &= \delta a_1(j-1) \\ \delta a_2(j) &= \delta a_2(j-1) \end{aligned} \qquad (106)$$

Let

$$\underline{\xi}(j(= \begin{bmatrix} \delta ym(j) \\ \delta a_1(j) \\ \delta a_2(j) \end{bmatrix} \qquad (107)$$

Then

$$\underline{\xi}(j) = \Phi(j-1)\ \xi(j-1) \qquad (108)$$

where

$$\Phi(j-1) = \begin{bmatrix} a_1^{(1)} & ym^{(1)}(j-1) & u(j) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (109)$$

At this stage, instead of solving the optimization problem stated in (103), the following problem is solved.

Problem 5.2: Find the initial conditions of (108) which minimizes

[§] To simplify the notation, the index k is dropped. Thus, at the time of computation, $J=0 \Rightarrow j=k-N$ and $j=N \Rightarrow j=k$.

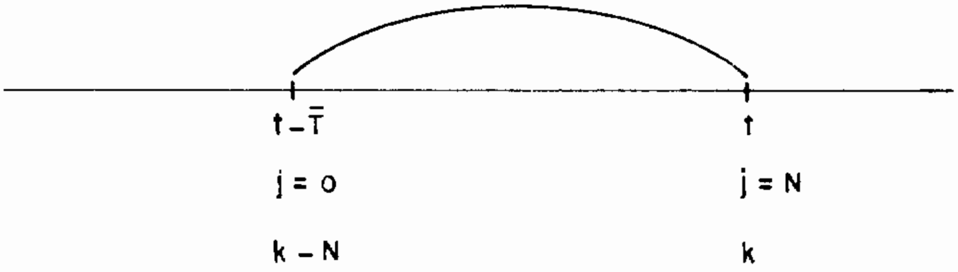


Figure 28. Observation Interval

$$J = \sum_{j=1}^{N} \left(z(j) - ym^{(1)}(j) - \delta ym^{(1)}(j) \right)^2 \tag{110}$$

where $\delta ym^{(1)}(k)$ is subject to the constraint (108). We have converted a nonlinear problem into a linear problem. By repeatedly solving this last problem we hope to approach the solution to the first problem.

Problem 5.2 is solved by using the least-squares curve fitting procedure. It is noted that

$$ym^{(1)}(j) + \delta ym^{(1)}(j) = z(j) + n(j) \tag{111}$$

where $n(j)$ is the discrepancy caused by noise and error in the parameter adjustment. Let

$$\delta ym^{(1)}(j) \doteq z(j) - ym^{(1)}(j) \tag{112}$$

The right-hand side of (112) is known and it is desired to determine $\delta ym^{(1)}(j)$, subject to (108), which best approximates $z(j) - ym(j)^{(1)}$. Equation (112) can be rewritten as

$$\underline{h}^* \underline{\zeta}(j) \doteq z(j) - ym^{(1)}(j) \tag{113}$$

where

$$\underline{h}^* = (1 \quad 0 \quad 0).$$

The N equations represented by (113) can all be rewritten in terms of $\zeta(0)$ by using (109).

$$\begin{aligned}
\underline{h}^* \underline{\zeta}(0) &= z(0) - ym^{(1)}(0) \\
\underline{h}^* \phi(1,0)\, \underline{\zeta}(0) &= z(1) - ym^{(1)}(1) \\
&\;\;\vdots \\
\underline{h}^* \phi(N,0)\, \underline{\zeta}(0) &= z(N) - ym^{(1)}(N)
\end{aligned}$$

Or, in matrix form

$$A\, \underline{\zeta}(0) = \underline{\xi} \tag{114}$$

where

$$A = \begin{bmatrix} h^* \\ h^* \phi(1,0) \\ h^* \phi(N,0) \end{bmatrix} \qquad N + 1 \text{ x } 3 \text{ matrix}$$

$$\underline{\xi} = \begin{bmatrix} z(0) - ym^{(1)}(0) \\ \vdots \\ z(N) - ym^{(1)}(N) \end{bmatrix} \qquad N + 1 \text{ x } 1 \text{ vector}$$

The pseudo-inverse routine is used to solve (114).

$$\underline{\xi}(0)^{(1)} = A^{\dagger}\, \underline{\xi}^{(1)} \tag{115}$$

From (115) we can make corrections to the initial trial of the parameters and initial conditions.

$$\begin{aligned} a_i^{(2)} &= a_i^{(1)} + \delta a_i^{(1)}(0) \\ ym(0)^{(2)} &= ym(0)^{(1)} + \delta ym^{(1)}(0) \end{aligned} \tag{116}$$

The procedure can now be repeated.

5.4 Algorithm and Convergence

The procedure outlined in the last section may well be divergent. Procedures using the digital computer can, however, be used to give monotone convergence. This section will give the algorithm which assures this important property.

From the initial trial and solution we can compute the error index.

$$J_1 = \sum \left(z(j) - ym^{(1)}(j) \right)^2 = \left|\left| \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} \right|\right|^2$$

The problem is to find a $\delta ym(k)$ such that J_2 given by

$$J_2 = \sum \left(z(j) - ym^{(1)}(j) - \delta ym(j) \right)^2$$

is less than J_1.

The difference $J_1 - J_2$ must be greater than zero.

$$\begin{aligned} J_1 - J_2 = {} & \cancel{\left|\left| \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} \right|\right|^2} - \cancel{\left|\left| \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} \right|\right|^2} \\ & + 2 < \delta \underset{\sim}{ym},\ \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} > \\ & - \left|\left| \delta ym \right|\right|^2 \geq 0 \end{aligned}$$

or

$$2 < \delta \underset{\sim}{ym},\ \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} > - \left|\left| \delta \underset{\sim}{ym} \right|\right|^2 \geq 0 \tag{117}$$

Equation (117) is the condition for convergence. If

$$< \delta \underset{\sim}{ym},\ \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} > \neq 0$$

Then for $\delta \underset{\sim}{ym}$ sufficiently small (117) can be satisfied since the first term is linear in $\delta \underset{\sim}{ym}$ while the second term is quadratic. It is noted that the first term in (117) is positive since it is the scalar product between the error and the projection of the error on the linear manifold.

The condition

$$< \delta \underset{\sim}{ym}, \underset{\sim}{z} - \underset{\sim}{ym}^{(1)} > = 0 \tag{118}$$

requires that $\underset{\sim}{ym}^{(1)}$ is closer to $\underset{\sim}{z}$ than any nearby point obtained through linear perturbation. In other words, the gradient is zero and we have a local minimum.

The situation is shown in Figure 29. The first linear correction is 1'. Upon solving (104) point 1 is obtained which may well give a J which is greater than J_1. If $J_2 > J_1$, then we cut the correction, $\delta ym(k)$, by a half yielding point 1. If the J at point 1 is less than J_1 then we keep the correction given by $\delta ym(k)^{(1)}/2$. If not we cut $\delta ym(k)^{(1)}/2$ by a half and repeat this process. By using this cutting procedure we have monotone convergence until condition (118) is reached.

In an on-line task, we are limited in the number of iterations we can make at a given time. The requirement is not as stringent, however, as the control synthesis problem because the estimation can be made at wider time intervals for slowly varying processes. If we limit the number of cutting procedures described in the last paragraph, we may never find the correction which will give a smaller J. In this case no corrections will be made and we go on to the next interval. Here again, no interval may give corrections, in which case the method fails. It is felt, however, that for a class of problems in which the estimates are within a certain range from the true values the routine will be applicable. This problem seems no worse than the instability problem associated with Margolis' procedure.

5.5 Simulation

A digital simulation of the modified Newton's procedure was made on an IBM 7090. As a comparison, the discrete version of Margolis' procedure was also simulated. The experimental set-up and results will be discussed in this section.

Let us first describe the experimental set-up for the modified Newton's procedure. The first-order process with two unknown coefficients was taken as an example. This process has the form

$$y(k) = \alpha_1 \, y(k-1) + \alpha_2 \, u(k)$$
$$z(k) = y(k) + v(k)$$

The noise, v(k) was an uncorrelated noise with a uniform distribution because it was readily available. It is believed that this distribution is more severe than the usual Gaussian noise if the variances of the two are the same. Many runs were made, however, without noise.

The flow chart for the simulation is shown in Figure 30. Over 100 points of u(k) were inserted. Either a triangular wave with a period of 24 sampling instants or a square wave with a period of 20 sampling instants was used. The observation interval was taken as 10

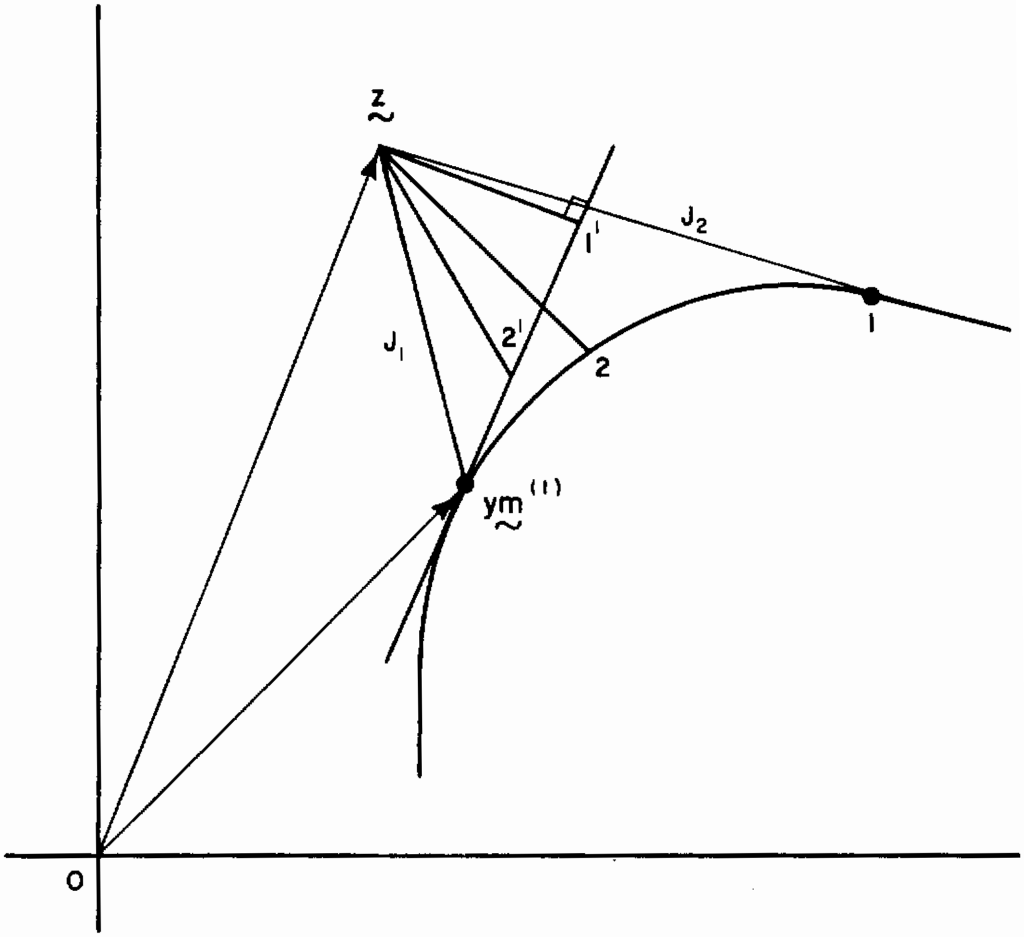


Figure 29. Two-Dimensional Picture of Correction Scheme

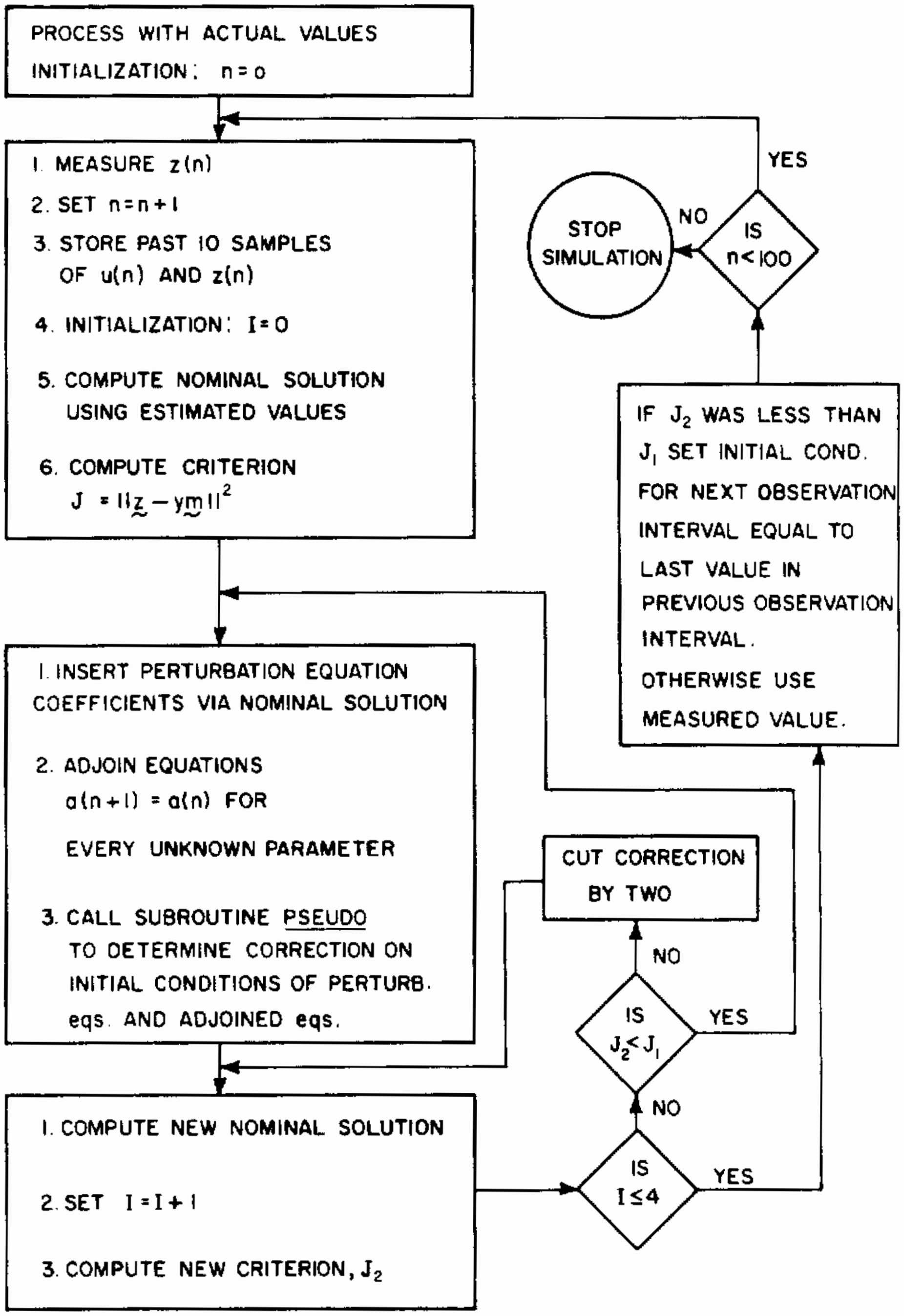

PROCESS WITH ACTUAL VALUES
INITIALIZATION: n = o
1. MEASURE z(n)
2. SET n=n+1
3. STORE PAST 10 SAMPLES OF u(n) AND z(n)
4. INITIALIZATION: I = 0
5. COMPUTE NOMINAL SOLUTION USING ESTIMATED VALUES
6. COMPUTE CRITERION $J = \|\underset{\sim}{z} - \underset{\sim}{ym}\|^2$
YES
NO
STOP SIMULATION
IS n<100
IF J_2 WAS LESS THAN J_1 SET INITIAL COND. FOR NEXT OBSERVATION INTERVAL EQUAL TO LAST VALUE IN PREVIOUS OBSERVATION INTERVAL.
OTHERWISE USE MEASURED VALUE.
1. INSERT PERTURBATION EQUATION COEFFICIENTS VIA NOMINAL SOLUTION
2. ADJOIN EQUATIONS a(n+1) = a(n) FOR EVERY UNKNOWN PARAMETER
3. CALL SUBROUTINE PSEUDO TO DETERMINE CORRECTION ON INITIAL CONDITIONS OF PERTURB. eqs. AND ADJOINED eqs.
CUT CORRECTION BY TWO
NO
IS $J_2 < J_1$
YES
NO
1. COMPUTE NEW NOMINAL SOLUTION
2. SET I = I + 1
3. COMPUTE NEW CRITERION, J_2
IS $I \leq 4$
YES

Figure 30. Flow Chart for Modified Newton's Procedure

sampling instants and the intervals were taken in a block processing manner. (In an actual application probably more points will be taken.) Four iterations were taken per observation interval. If needed, the cutting-by-two procedure was counted as an iteration. The method requires initial conditions for the model equations at the beginning of every observation interval. These were supplied by either of two ways. First, if the previous interval revealed an improvement in the criterion J, then the state values at the last sampling instant of the nominal solution of the previous interval were used as the initial conditions. Otherwise, the measured outputs were used as the initial conditions.

For Margolis' procedure essentially the same conditions prevailed to permit a comparison. The procedure provides adjustment after every sampling instant as described in Section 5.2. This procedure requires insertion of a gain, K, for the steepest descent procedure.

For the first series of runs, the process parameters were taken as constant but unknown. The estimates were initially displaced from the true value. A representative no-noise case is shown in Figure 31. After three observation intervals the true values are obtained. It was found that large displacements of the initial estimates can still provide convergence. Even unstable roots were identified. From this series of runs, it is felt that any root near and within the unit circle can be identified for the first-order process regardless of the initial uncertainty.

For the second series of runs, noise was added to the output of the process. Noises with 5% and 10% of the peak output were inserted along the initial displacements of the estimated values. Several results are shown in Figures 32, 33, and 34. The results shown convergence from the displacements but an error in the estimated values. The 10% noise case reveals that possibly more than 10 points are required for the averaging. Essentially, there is no significant difference between the triangular and square wave inputs.

For the third series of runs, the true values were continually changed as a ramp. Both noise and no-noise cases were taken. Some results are shown in Figures 35, 36, 37, and 38. First, even without noise the tracking capability is rather poor if the parameters are changing as much as 0.0025 per sampling instant. With 5% noise, the situation is even worse. Close analysis of Figure 36 showed that up to 50T the signal-to-noise ratio was much worse than 5%. As the signal portion increased, the tracking capability improved. Figures 37 and 38 show that the method is able to track changes in parameter of 0.00125 per sampling instant even with noise. Essentially, there is no significant difference between the triangular and square wave inputs.

Results using the discrete version of Margolis' procedure are shown in Figures 39, 40, 41, 42, 43, and 44. The adjustment of K is very critical. Many runs were made before a satisfactory K was obtained. (This adjustment was very troublesome on the digital computer.)

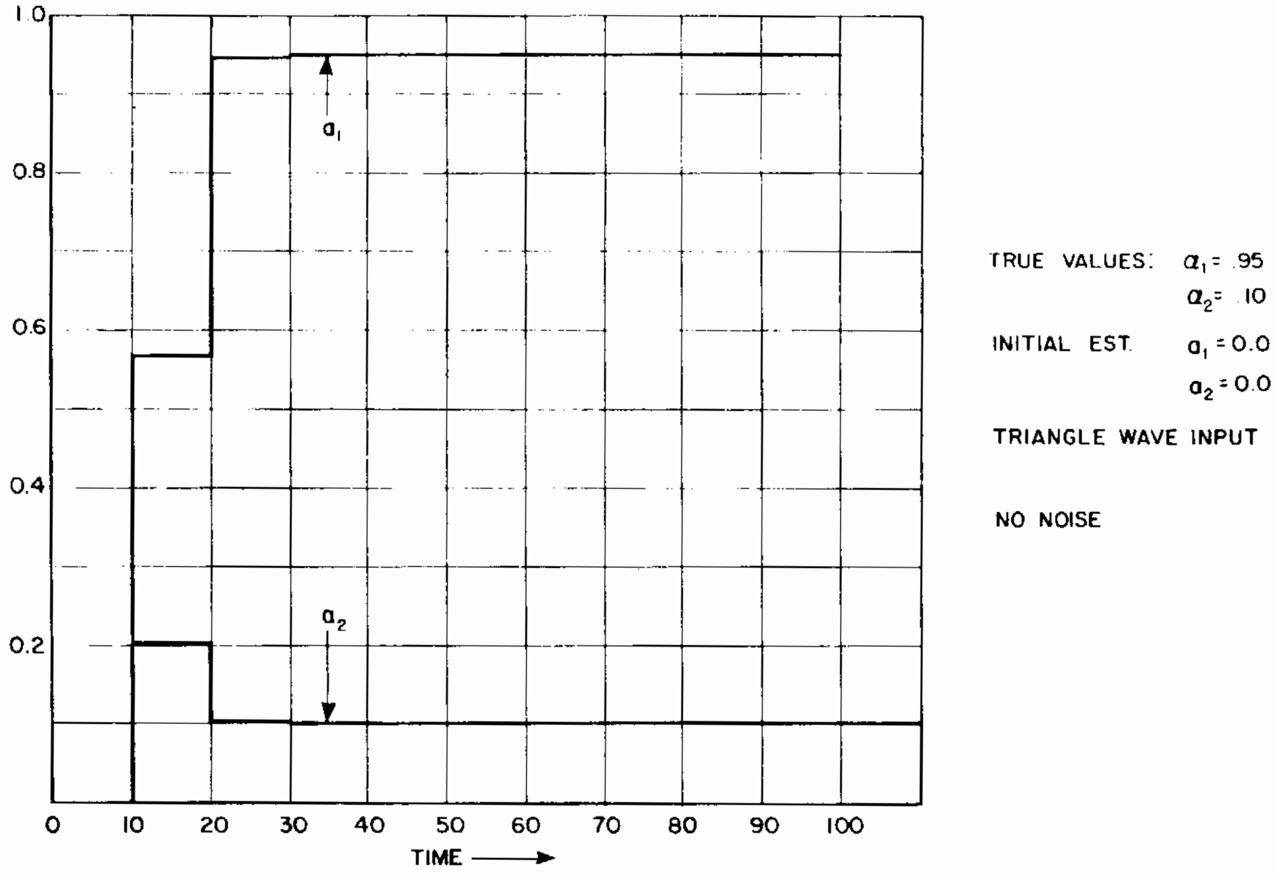


Figure 31. Modified Newton's Procedure, Constant Unknown Parameter

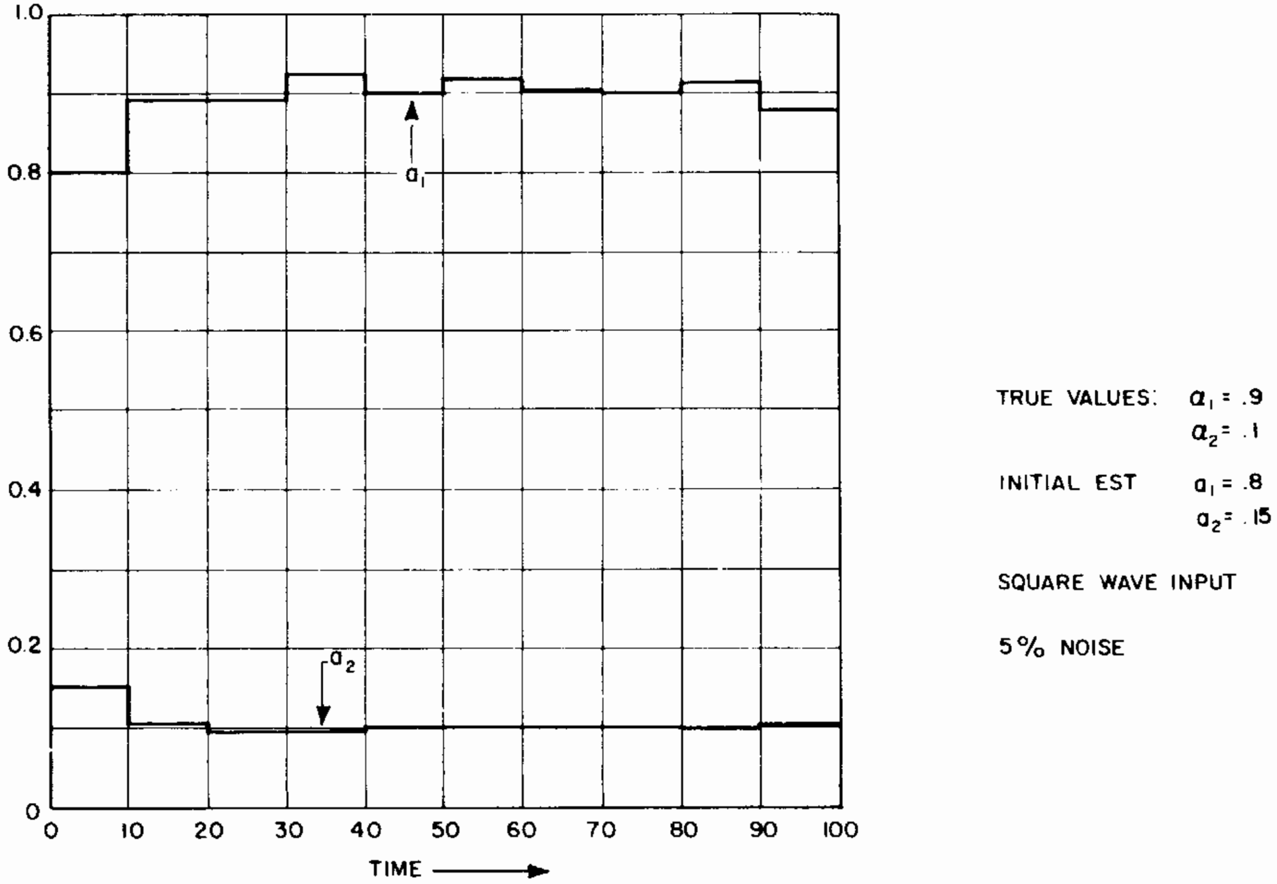


Figure 32. Modified Newton's Procedure, 5% noise, Square Wave

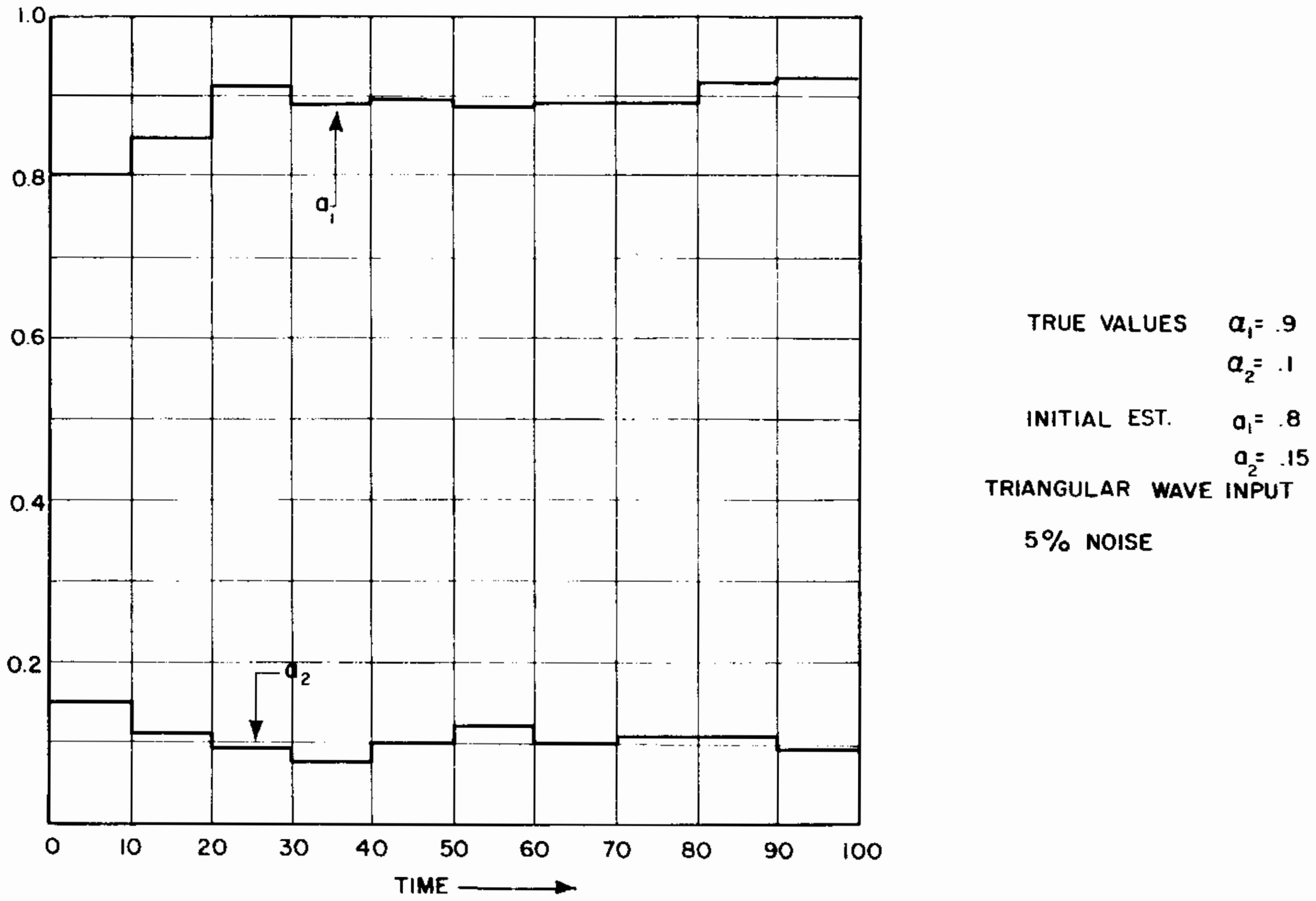


Figure 33. Modified Newton's Procedure, 5% noise, Triangular Wave

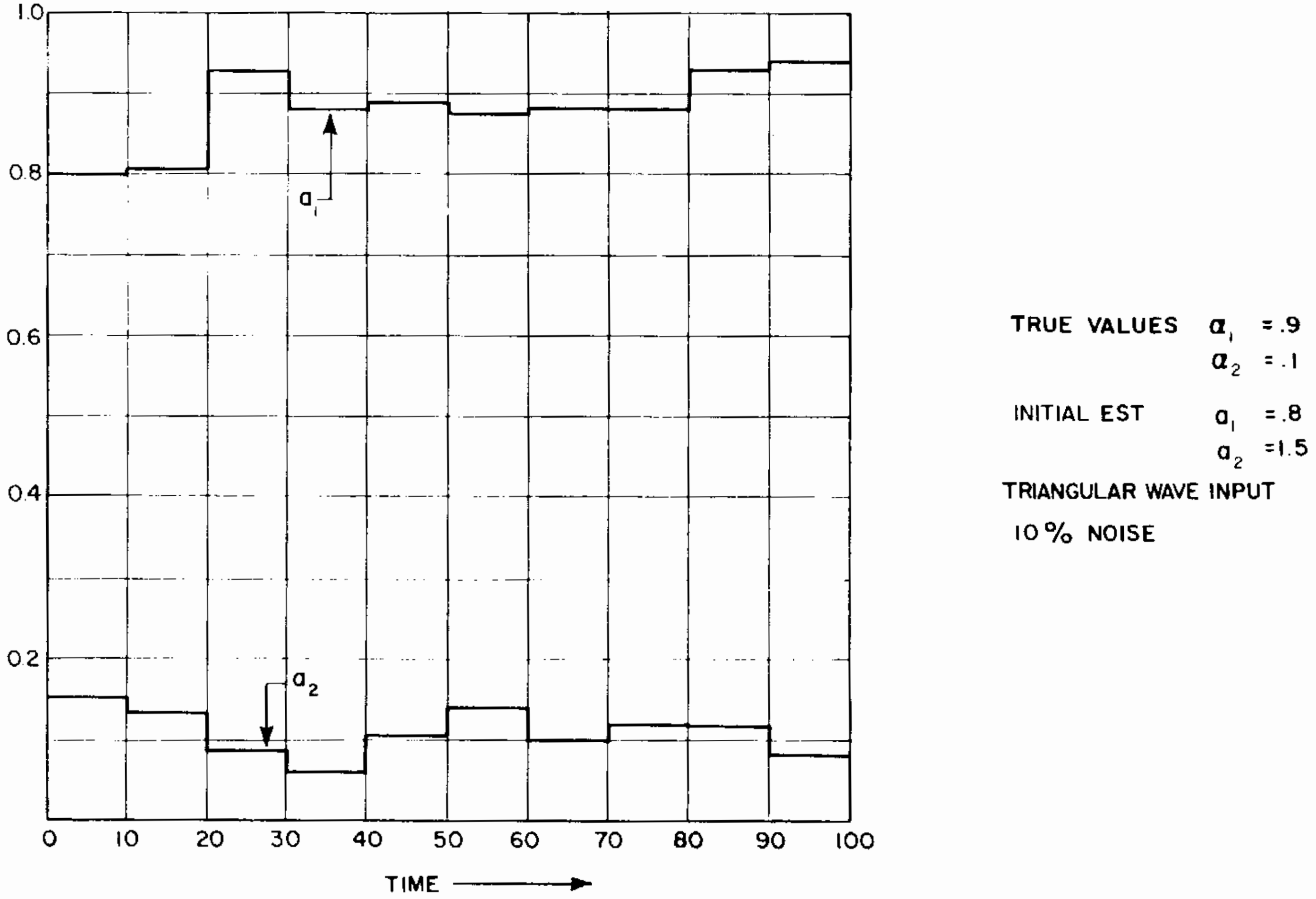


Figure 34. Modified Newton's Procedure, 10% noise, Triangular Wave

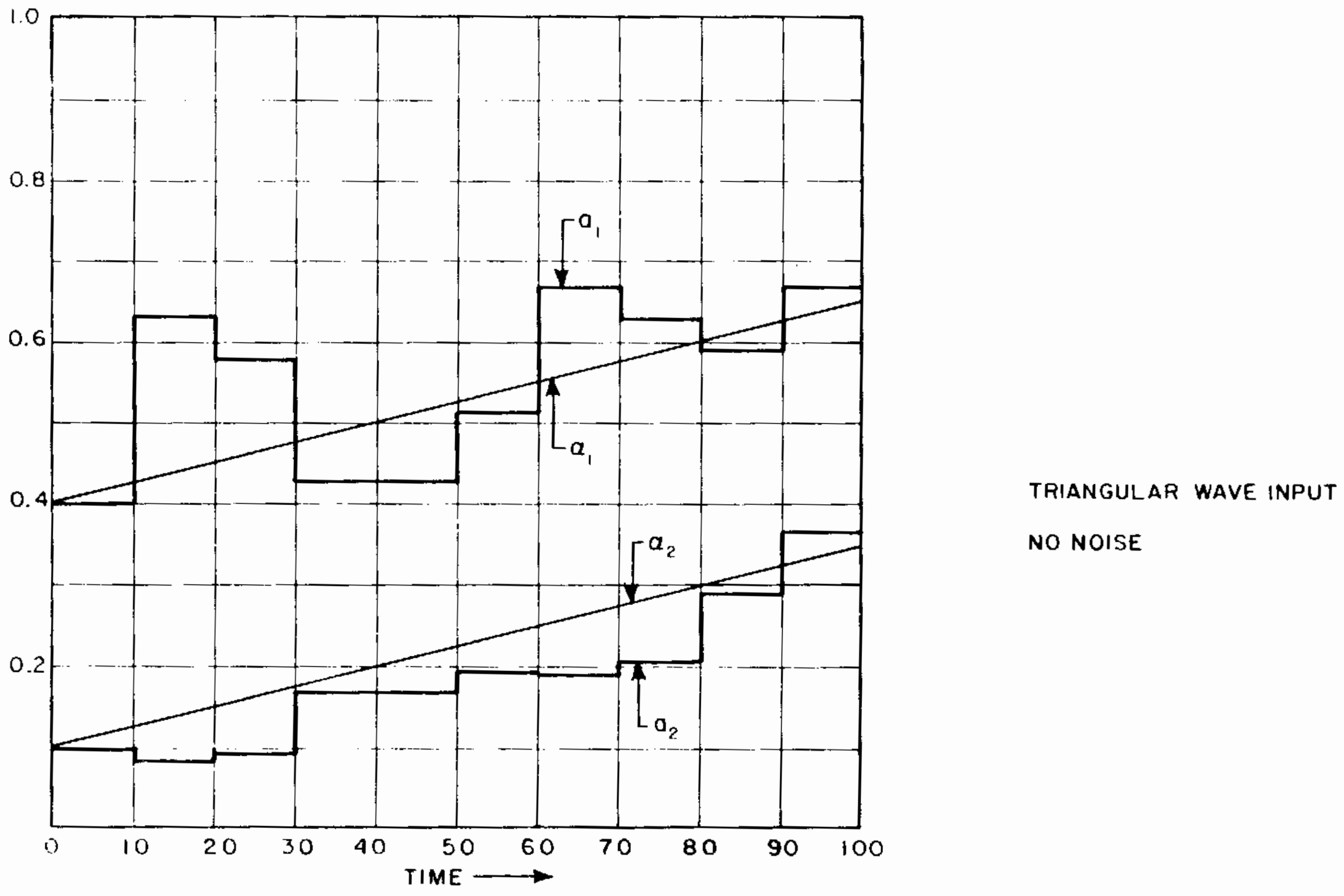
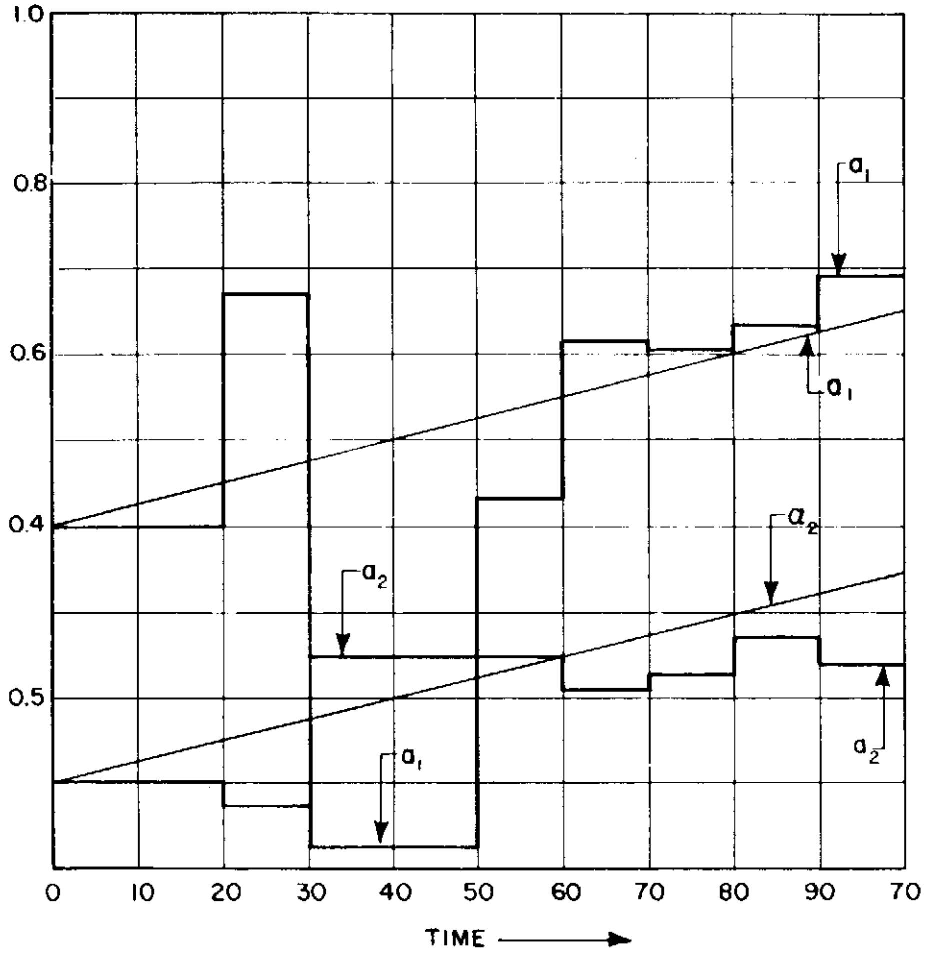


Figure 35. Modified Newton's Procedure, Changing Parameters (.0025/T), No Noise



TRIANGULAR WAVE INPUT

5% NOISE

Figure 36. Modified Newton's Procedure, Changing Parameters (.0025/T), 5% Noise

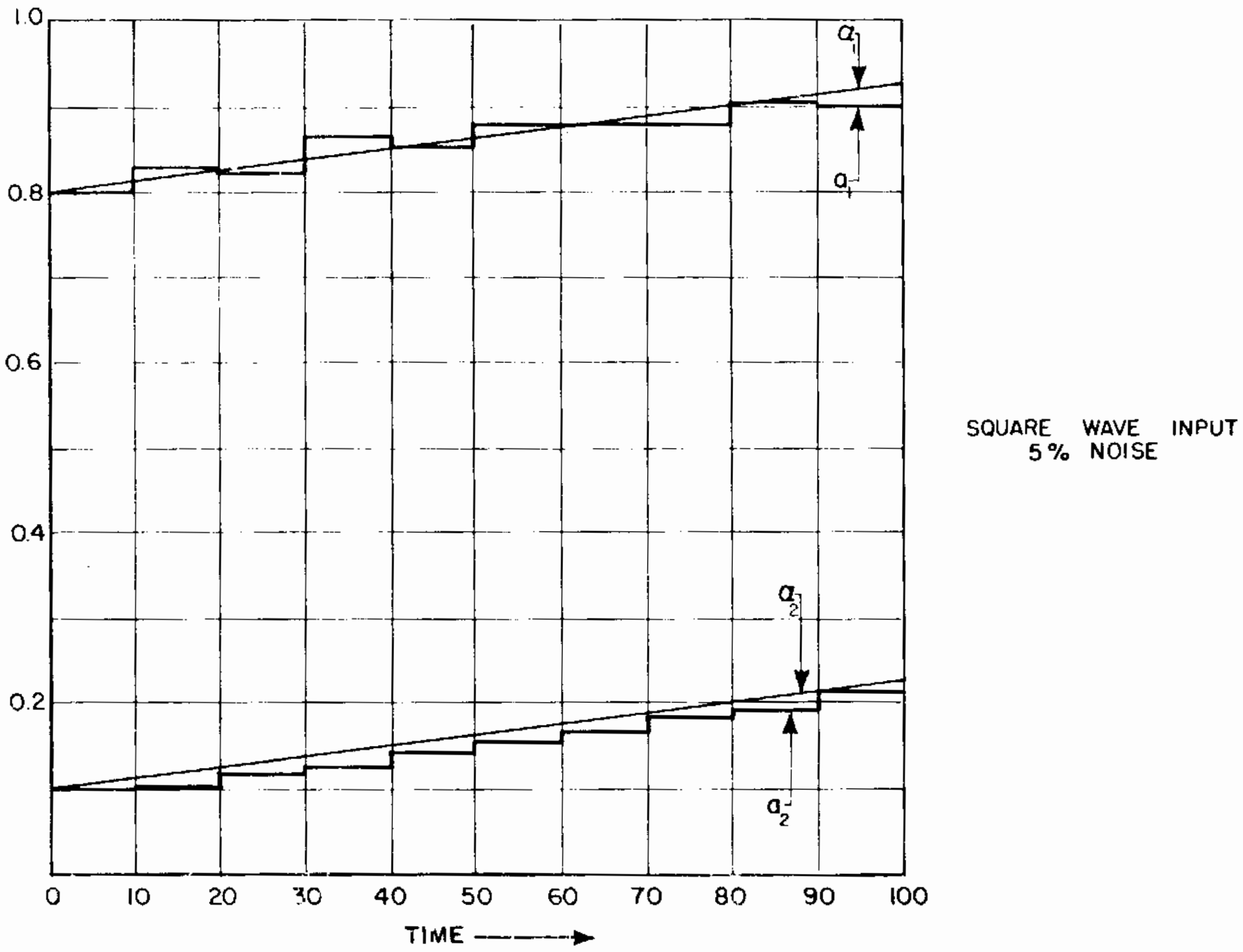


Figure 37. Modified Newton's Procedure, Changing Parameters (.00125/T), Square Wave

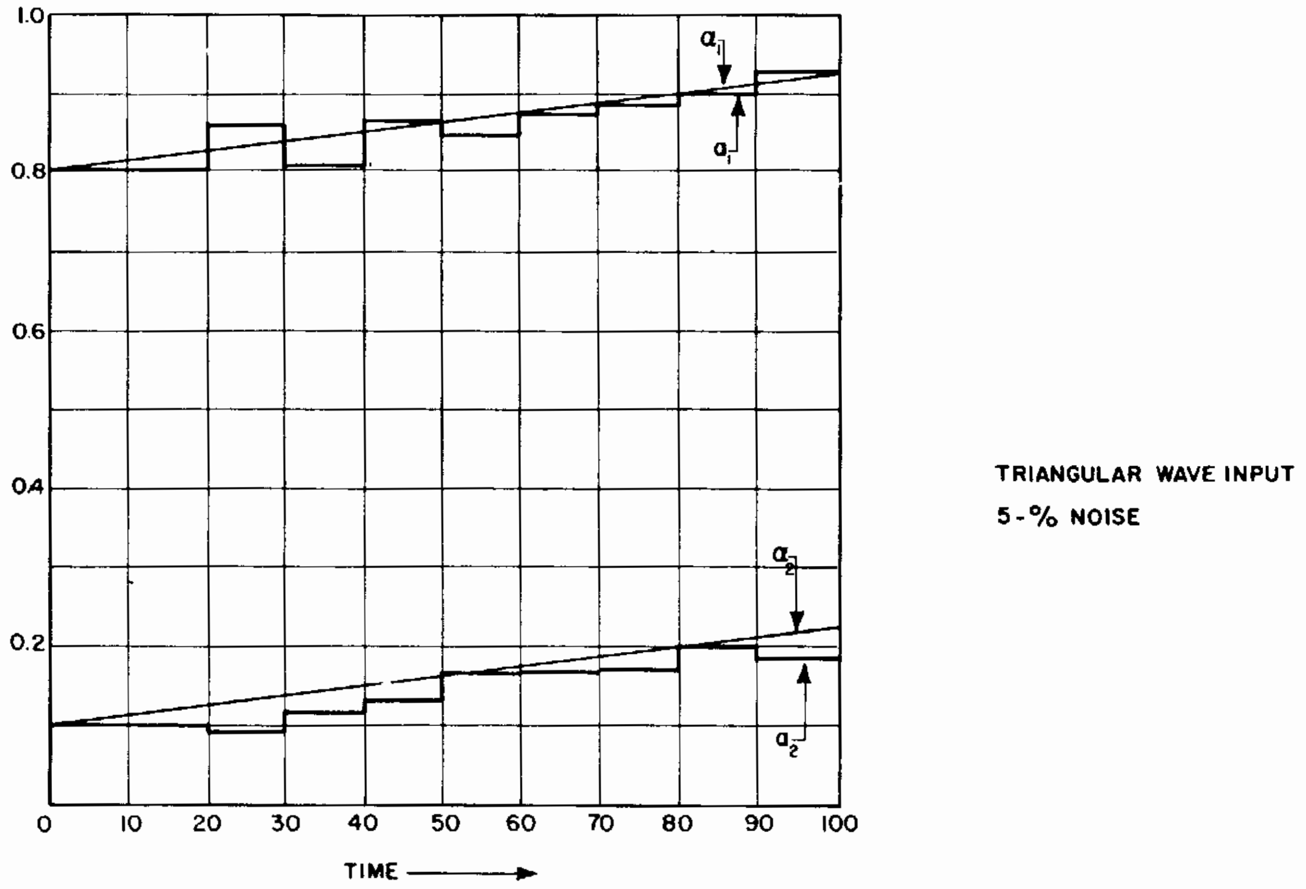


Figure 38. Modified Newton's Procedure, Changing Parameters (.00125/T) Triangular Wave

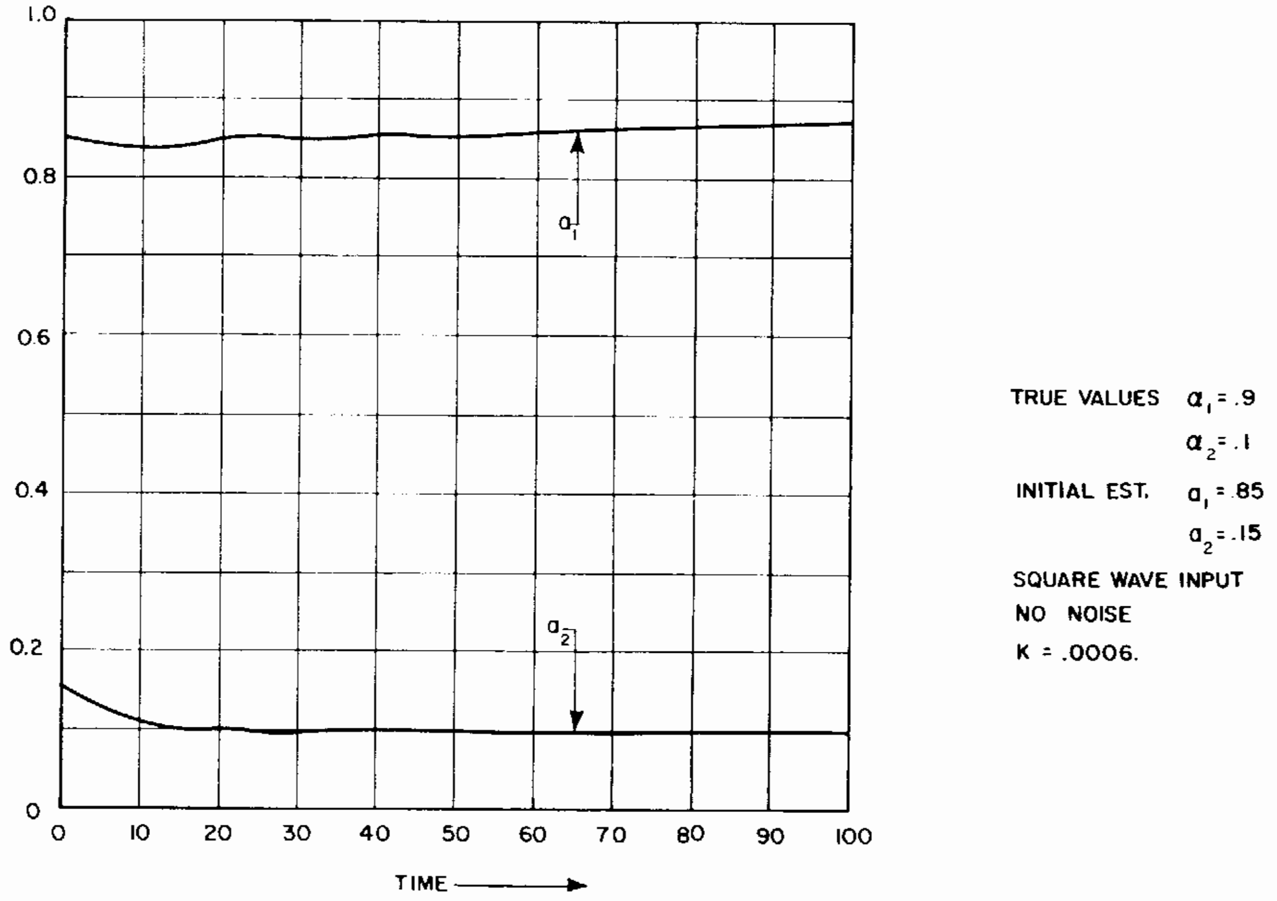


Figure 39. Margolis' Procedure, Constant Unknown Parameters, Square Wave

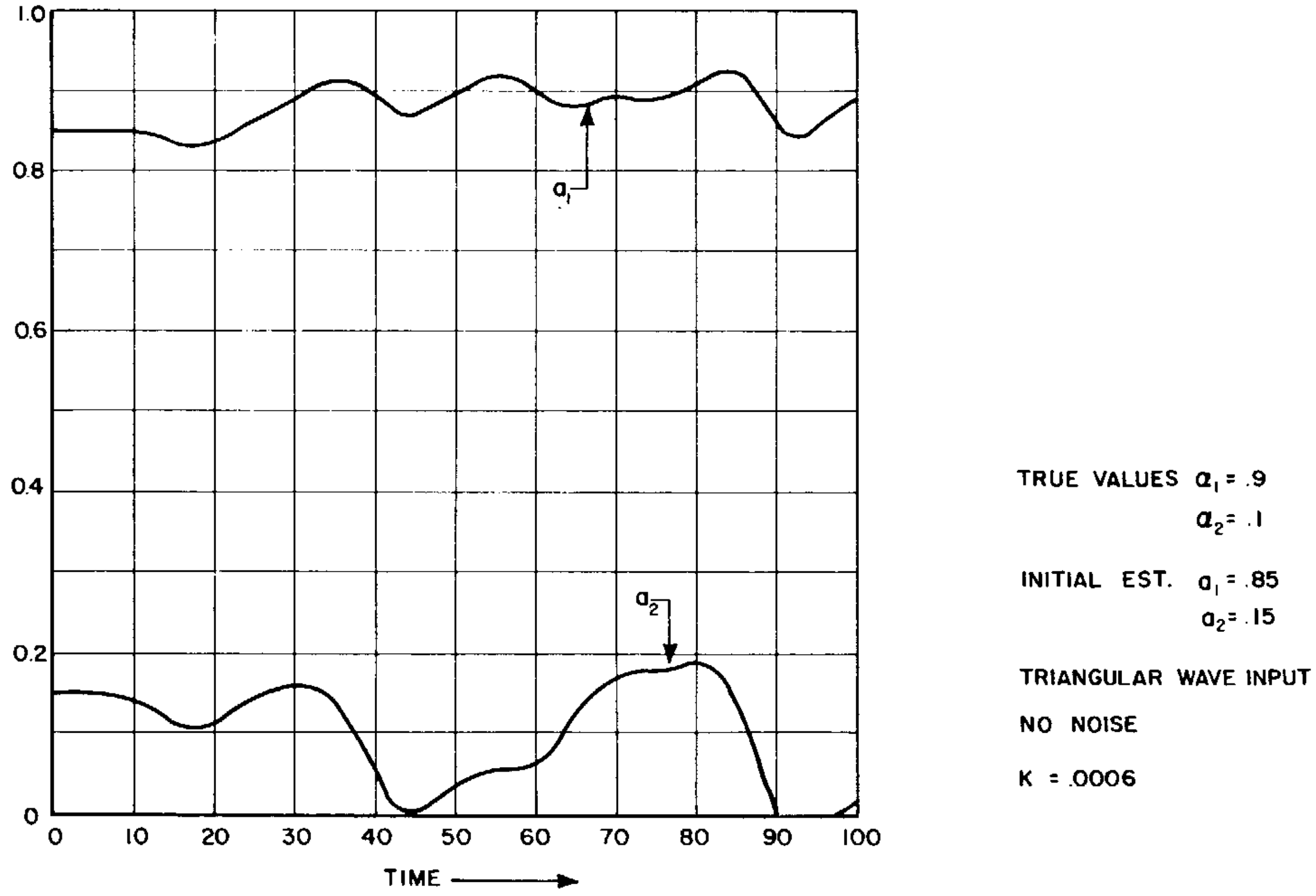
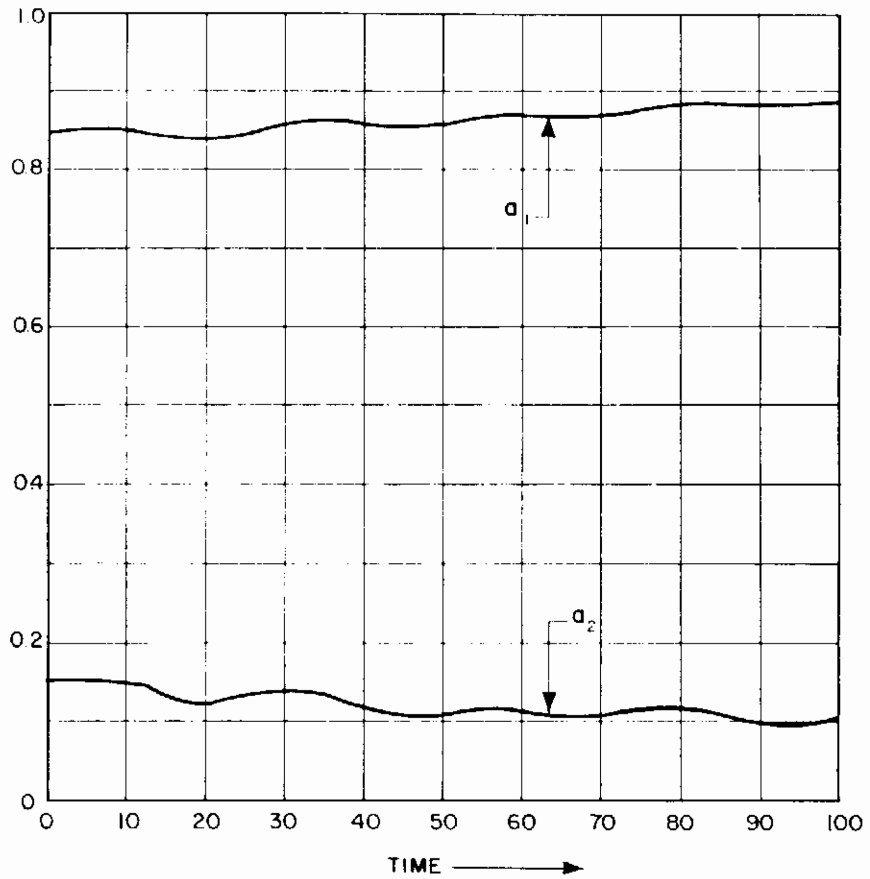


Figure 40. Margolis' Procedure, Constant Unknown Parameters, Triangular Wave



TRUE VALUES $a_1 = .9$
$a_2 = .1$

INITIAL EST. $a_1 = .85$
$a_2 = .15$

TRIANGULAR WAVE INPUT

NO NOISE

K = .00024

Figure 41. Margolis' Procedure, Constant Unknown Parameters, $a_1 = .85$

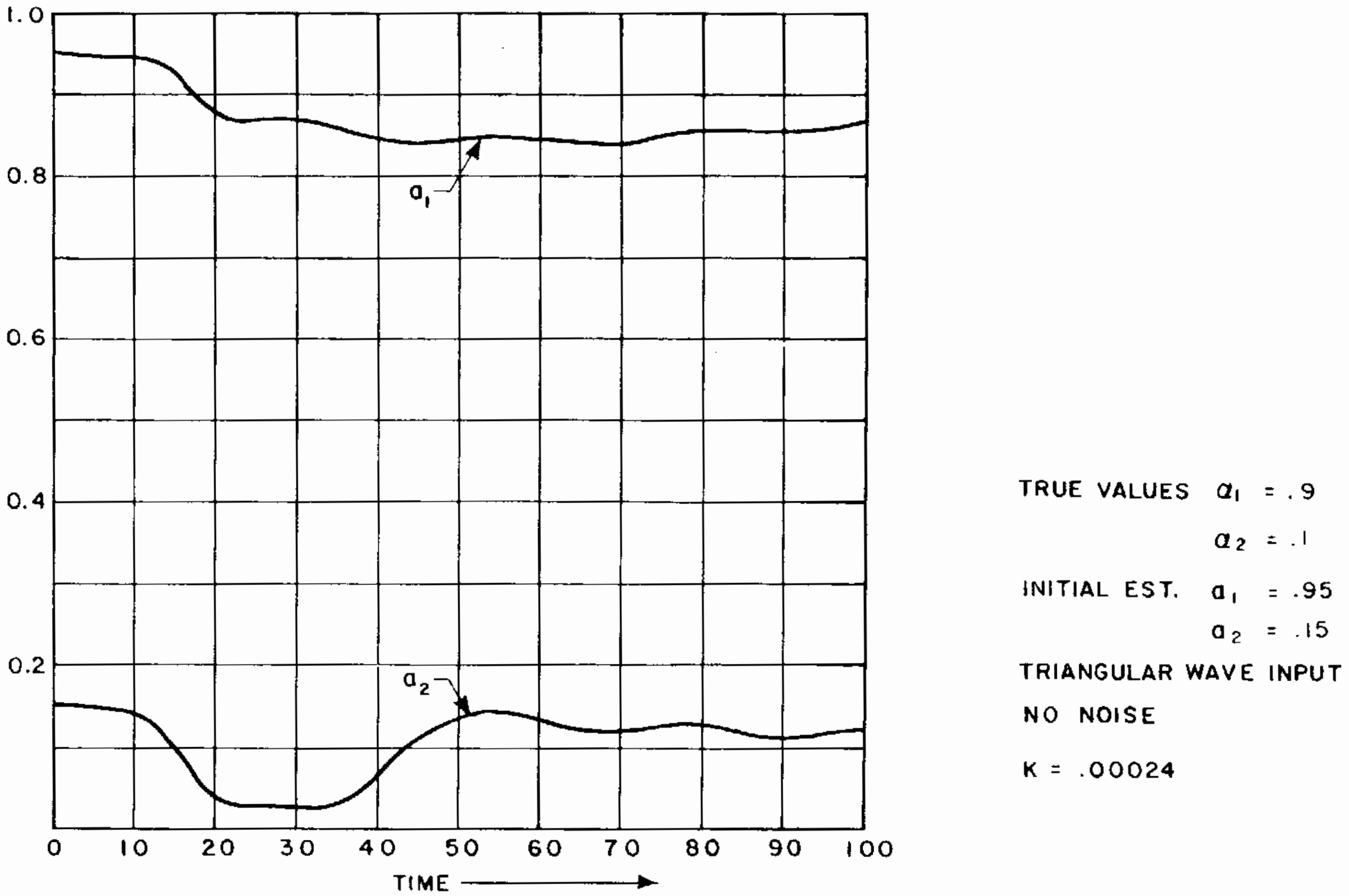
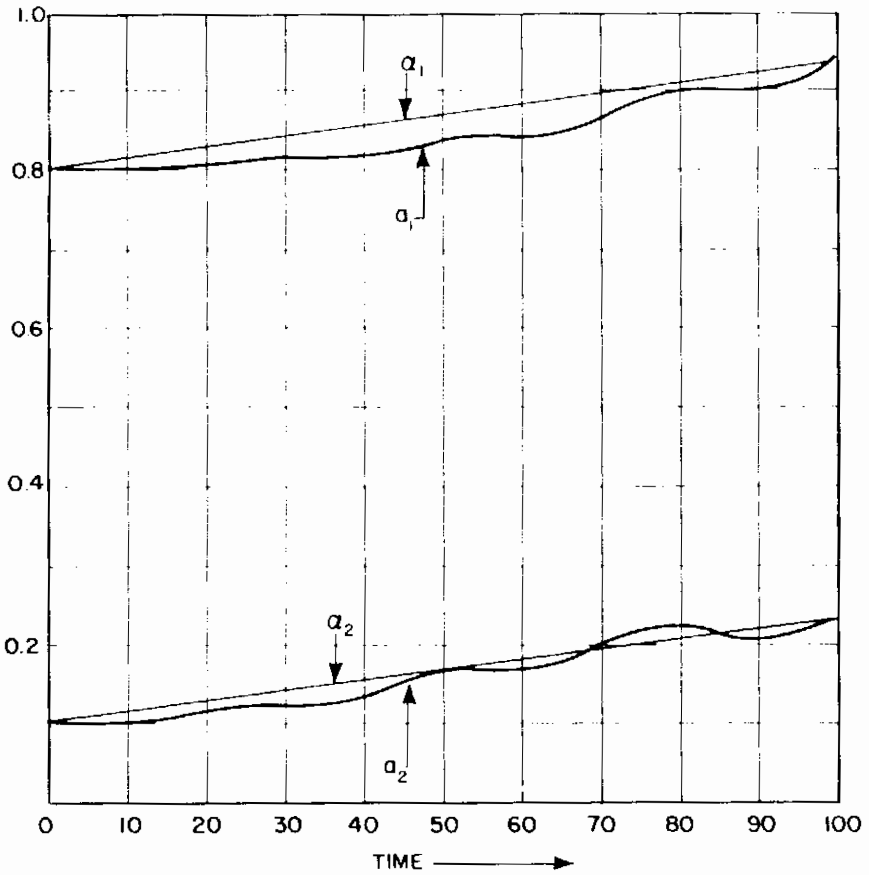


Figure 42. Margolis' Procedure, Constant Unknown Parameters, a_1 = .95



TRIANGLE WAVE INPUT

NO NOISE

K = .00024

Figure 43. Margolis' Procedure, Changing Parameters

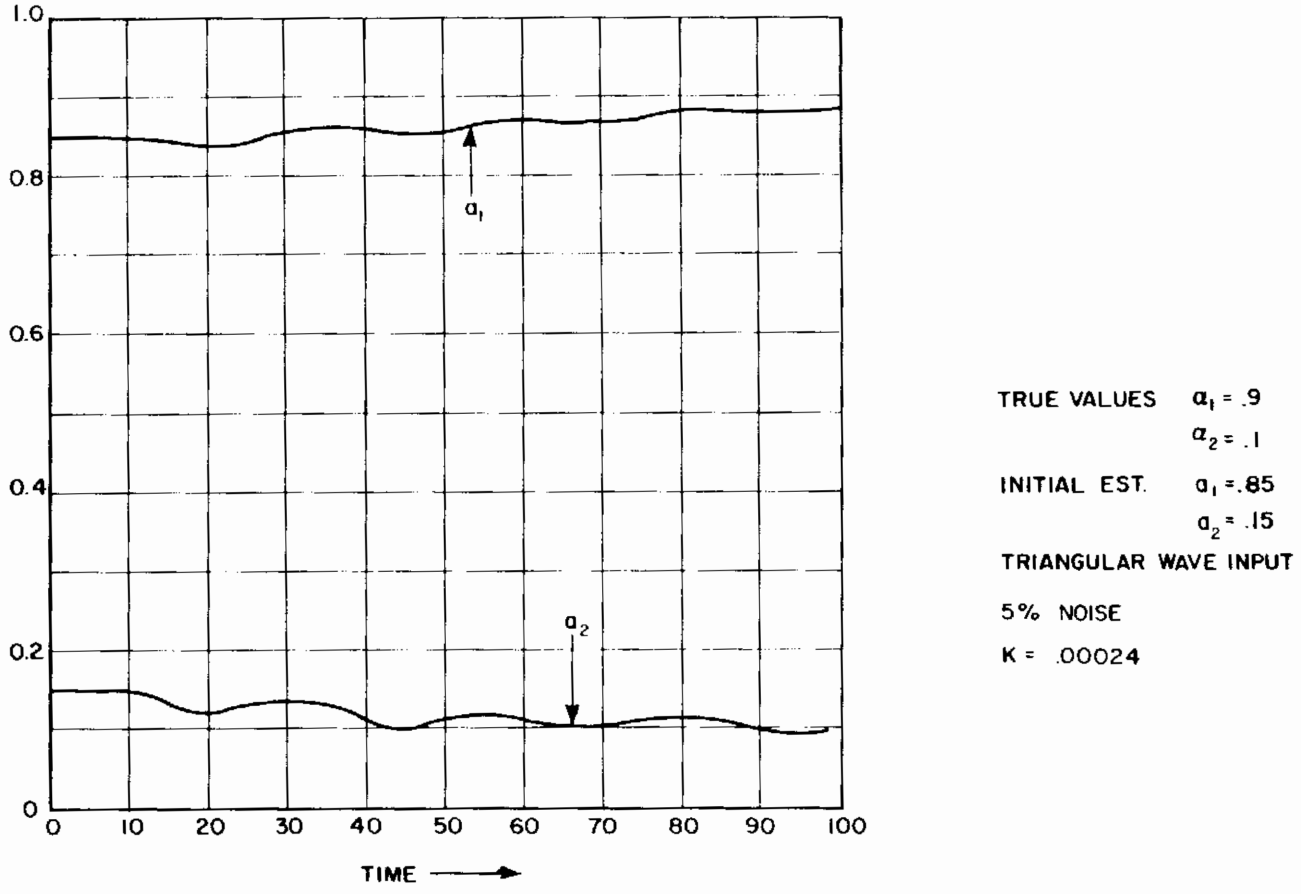


Figure 44. Margolis' Procedure, 5% Noise

This gain was dependent upon the input signal. When the gain was adjusted to give a satisfactory response to square waves, it was unsatisfactory for the triangular wave (Figures 39 and 40). Different types of behavior were obtained depending upon the direction of the initial offset. It seems that the best adjustment for K is when the behavior is slightly overdamped. Otherwise, oscillations appear to persist for a long time. With K adjusted to this seemingly suitable value, then it takes a long time before the true values are obtained. The method is also not applicable for large displacements of the initial guesses and the K seems to depend upon the values of the parameters which are being estimated. With the gain set so that the behavior is slightly overdamped, noise did not affect appreciably the response. (This fact was conjectured by Margolis.) In fact, with noise the gain should be even smaller.

Let us summarize the difficulties of the discrete version of Margolis' procedure.

1) The gain depends upon the input signal.
2) The response is slow, when K is properly adjusted.
3) The behavior differs depending upon the direction of the initial offset.
4) The method is applicable for small initial displacements between the estimate and true values.
5) The gain depends upon the true parameter values of the process.

Because of the criticalness of K, the modified Newton's procedure appears to be more practical even with the added complexity in computation. Even for the well-monitored experiments the adjustment of K was difficult. In an on-line application where the parameters are uncertain, the problems would be almost insurmountable.

5.6 A Possible Alternative

If the pseudo-inverse routine is computationally demanding, an alternative would be to use the steepest-descent method to perform the inversion of the rectangular matrix, (114). Choosing the criterion

$$J = \left|\left|A\ \underline{\zeta}(0) - \underline{\xi}\right|\right|^2 \qquad (119)$$

Or equivalently, minimizing

$$Q\left(\zeta(0)\right) = \underline{\zeta}(0^*)\ A^*A\ \underline{\zeta}(0) - 2\ \underline{\xi}^*\ A\ \underline{\zeta}(0)$$

We assume here that sufficient data points are processed so that A^*A is positive definite.

The gradient is given by

$$\nabla_{\zeta(0)}\ Q = A^*A\ \underline{\zeta}(0) - A^*\ \underline{\xi}$$

The next approximation is given by

$$\underline{\zeta}(0)^{(n+1)} = \underline{\zeta}(0)^{(n)} - \epsilon_n \nabla_{\zeta(0)} Q^{(n)}$$

where ϵ_n is determined so that the minimum point in the direction of the gradient is obtained, or

$$\epsilon_n = \frac{\left|\left|\nabla_{\zeta(0)} Q^{(n)}\right|\right|^2}{< A^*A \nabla_{\zeta(0)} Q^{(n)}, \nabla_{\zeta(0)} Q^{(n)} >}$$

As before, one can check to see whether (103) is actually decreasing, and if not perform the cutting by two procedures. It is noted here that even if J in (119) is continually decreasing it does not imply that (103) is decreasing.

CHAPTER 6

STATE VARIABLE ESTIMATION

6.1 Introduction

To use the adaptive controller discussed in Chapter 3, we must know the state variable at every sampling instant. This chapter will discuss a method of estimating these variables. The content of this chapter draws heavily from the work of Kalman (Reference 16). Joseph and Tou (Reference 23) have also made studies along this line.

The state variables can be estimated if the process is known. Also, it is known that the process can be determined if the state variables are known accurately. The task in adaptive controls is one step more difficult because neither the state variables nor the process is known accurately at any time. However, we can employ the following philosophy. If identification methods are available which can operate with inaccurate knowledge of the state variables, then the identified process can be used in the state variable estimation. A possible reason for taking this route is that the state variables generally change faster than the process parameters. As the identification methods of Chapters 4 and 5 were applicable even with unprecise knowledge of the state variables, those results can be used to update the process parameters in the state variable estimation. Therefore, the state variable estimation part can employ Kalman's recursive technique. The procedure will be outlined mainly to complete the total picture.

6.2 Outline of Estimation Problem

Let us refer to the process configuration shown in Figure 45. From the knowledge of $\underline{z}(k)$ and $\underline{u}(k)$, it is required to estimate the state, $\underline{x}(k)$, at the present time. The past values of $\underline{z}(k)$ and $\underline{u}(k)$ are known from some initial start time. The process characteristics, G_1 and G_2, are known, the former through identification. In an adaptive task the transfer characteristics are time varying. As new parameter values are obtained, the corresponding values used in the estimation will be changed. The covariance matrices of $\underline{v}(k)$ and $\underline{w}(k)$ are also known. These noise sources can be taken to be white noise. It is noted that because of G_2 the load disturbances can have a non-white spectra.

We note

$$\underline{x}(k) = \underline{x}_1(k) + \underline{x}_2(k)$$

where $\underline{x}_1(k)$ is known. Let

$$\underline{\nu}(k) = \underline{z}(k) - \underline{x}_1(k)$$

$$\underline{x}_2(k) = \underline{x}(k) - \underline{x}_1(k)$$

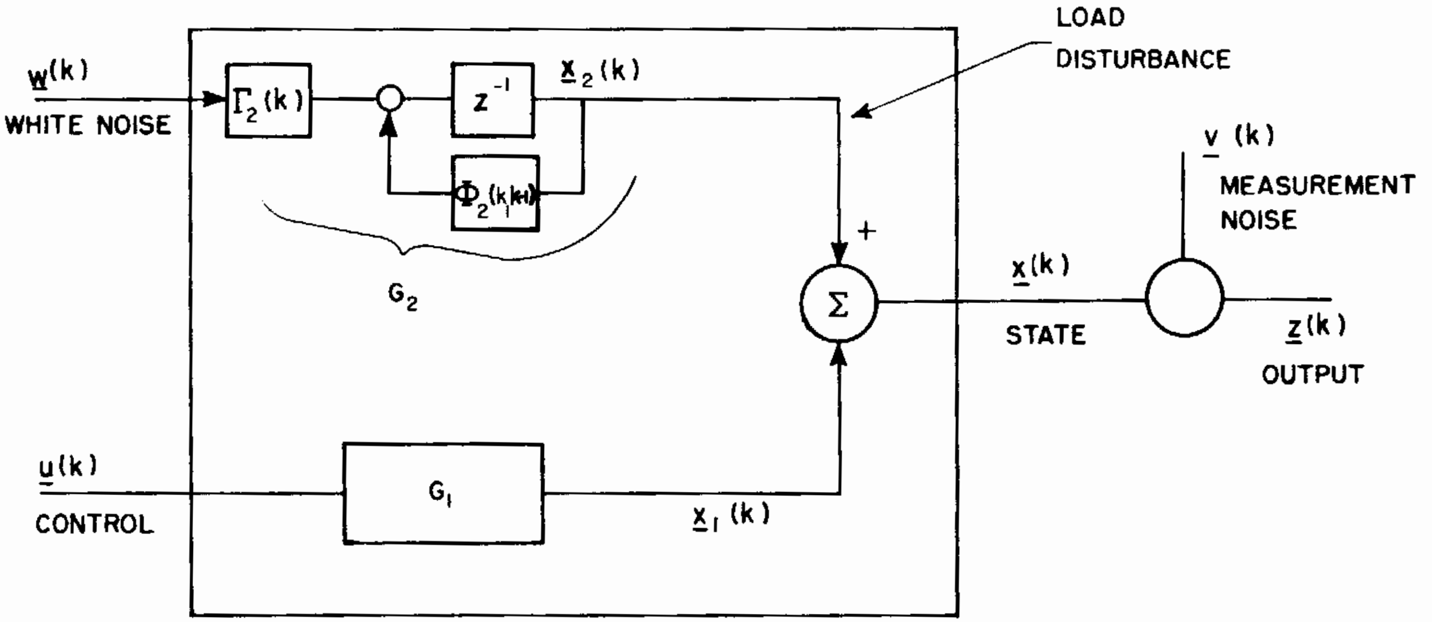


Figure 45. Process Configuration

The problem is now simply the determination of $\hat{\underline{x}}_2(k)$ which is the conditional expectation given $\underline{\nu}(k)$, $k = 0, 1, \ldots, k$. From $\hat{\underline{x}}_2(k)$ the estimate of the state is

$$\hat{\underline{x}}(k) = \hat{\underline{x}}_2(k) + \underline{x}_1(k)$$

Therefore, it can be seen that Kalman's filtering algorithm which can treat time-varying processes is applicable here.

CHAPTER 7

APPLICATION TO THE RE-ENTRY FLIGHT CONTROL PROBLEM

7.1 Introduction

The control of an aerospace vehicle entering the earth's atmosphere is one of the more challenging problems facing engineers at the present time (References 48, 49). Large variations and uncertainties in the process dynamics, primarily due to variations in air density, make feedback control mandatory. Furthermore, accuracy requirements may dictate using some sophisticated form of adaptive controls. This chapter will outline how the scheme discussed in the previous chapters can be applied to the re-entry problem.

7.2 Flight Path Control Problem

Probably the ideal method for the re-entry problem would compute optimum controls depending upon the present state and the desired destination. As time progresses the controls are recomputed. This task using the nonlinear equations of motion, however, is very difficult, requiring an enormous (IBM 7090) computer. Even if a computer is available the computation time will be an appreciable portion of the re-entry time. Therefore, some other procedure is required.

Several alternative schemes have been suggested in the literature (References 50, 51). One scheme performs re-entry by following a previously computed, stored optimal-trajectory. The adaptive control philosophy discussed in the previous chapters can be applied for such a scheme. Linear dynamical equations are obtained by writing perturbation equations evaluated along the nominal optimal trajectory.

Another scheme is to approximate the optimal path by segments of shorter paths which are easier to solve. This scheme is illustrated in Figure 46. As an example, the optimal path is approximated by three segments: 1) constant lift-to-drag ratio path, 2) constant altitude path, and 3) constant lift-to-drag ratio path. The adaptive control philosophy discussed in the previous chapters can be applied to each segment separately. The procedure will be illustrated for the constant altitude segment.

7.3 Constant Altitude Controller

First, the two-dimensional equations of motion will be derived. Let us refer to Figure 47. Summing the forces in the direction of V we obtain

$$\dot{V} = g \sin \gamma - \frac{D}{m} \qquad (120)$$

Summing the forces in the direction perpendicular to V we obtain

$$m V \dot{\theta} = g \cos \gamma - L$$

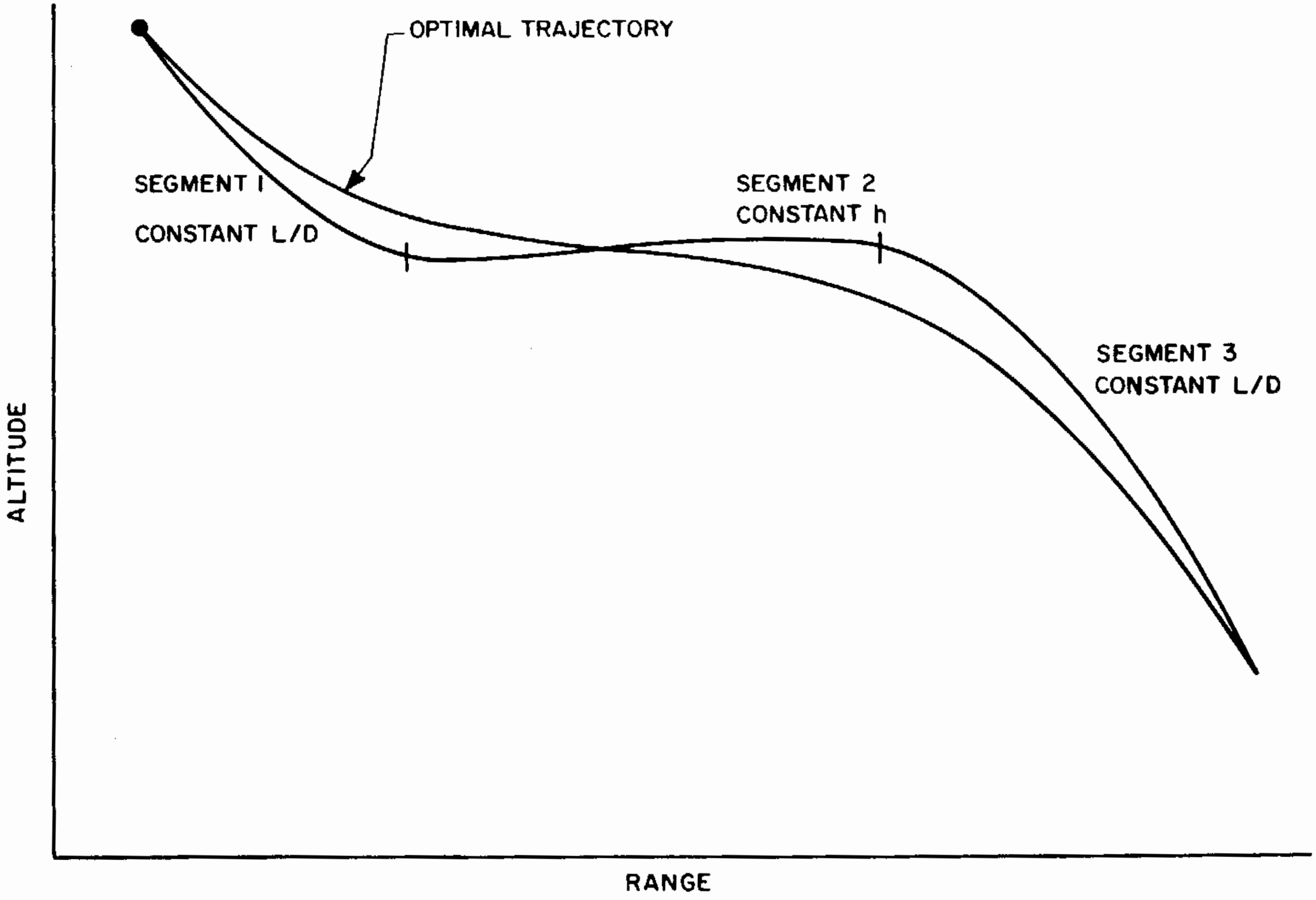


Figure 46. Approximation of Optimal Path

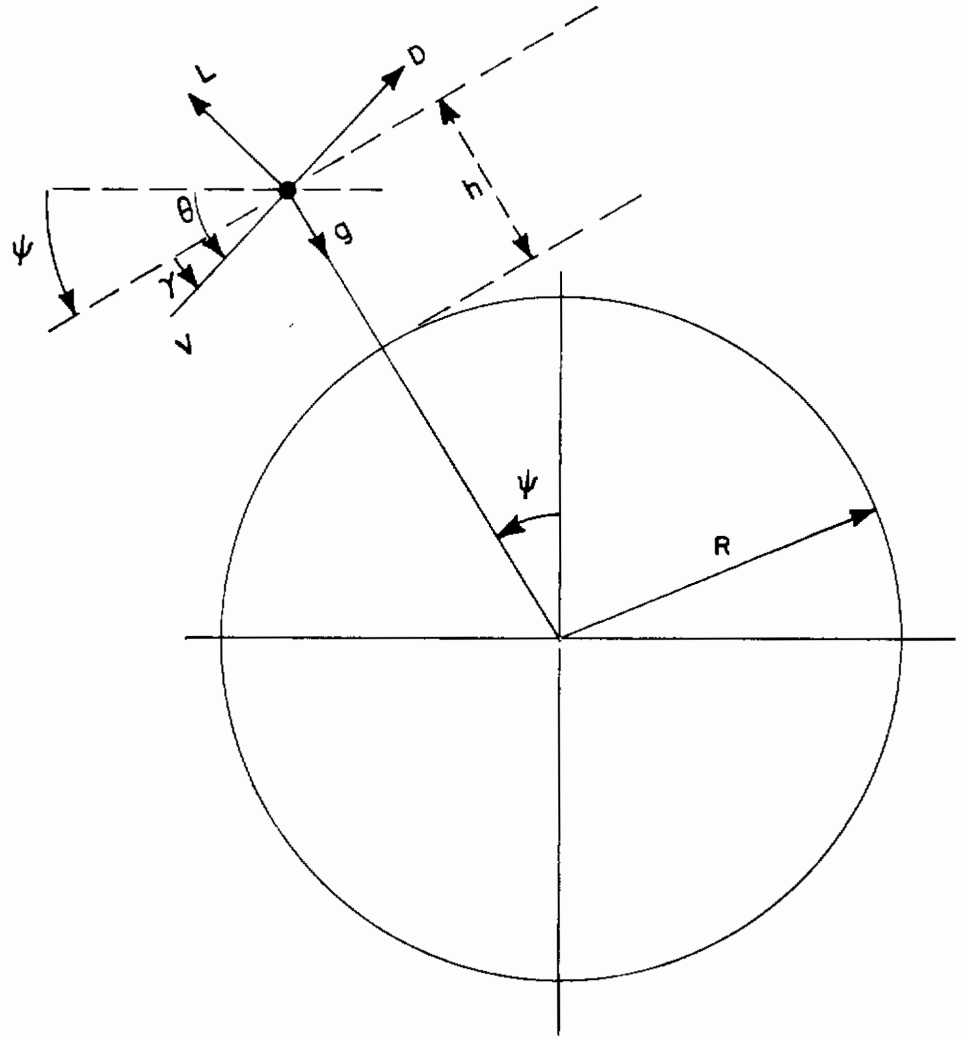


Figure 47. Geometry of Re-entry

Since

$$\dot{\psi} + \dot{\gamma} = \dot{\theta}$$

$$\dot{\psi} = \frac{V \cos \gamma}{R + h}$$

we obtain

$$\dot{\gamma} = -\frac{V \cos \gamma}{R + h} + \frac{g}{V} \cos \gamma - \frac{L}{mV} \tag{121}$$

In addition, the altitude rate is given by

$$\dot{h} = - V \sin \gamma \tag{122}$$

The names attached to the above symbols are:

γ - flight path angle measured from local horizontal
V - velocity
R - radius of Earth
h - altitude
L - lift force
D - drag force
m - vehicle mass
g - acceleration of gravity

In control terms, V, γ, and h are the state variables and L and D are control forces. The amount of lift and drag being applied at any time can be measured by accelerometers because

$$a_D = \frac{D}{m}$$

$$a_L = \frac{L}{m}$$

where

a_D - magnitude of deceleration measured by an accelerometer oriented along the velocity vector.

a_L - magnitude of deceleration measured by an accelerometer oriented perpendicular to the velocity vector.

Since independent control of lift and drag would be very difficult physically, we will assume that lift is a control force and D is a function of L.

$$D = f(L)$$

Next, we write perturbation equations of (120), (121), and (122) about the constant altitude condition. It is noted that $\gamma_o = 0$ and $\dot{h}_o = 0$. Therefore,

$$\dot{V}_o = - f\left(L_o(t)\right) \tag{123}$$

Or, the velocity must decrease along a constant altitude path. Also,

$$L_o(t) = -\frac{m\,V_o^2(t)}{R + h_o} + mg \qquad (124)$$

Along a constant altitude path, L_o, which is a function of time, must satisfy (123) and (124). Writing perturbation equations about V_o, $\gamma_o = 0$, h_o, L_o, we obtain

$$\delta\dot{V} = g\,\delta\,\gamma - 1/m \left.\frac{\partial f(L)}{\partial L}\right|_o \delta L \qquad (125)$$

$$\delta\dot{\gamma} = \frac{-mV_o^2 + (mg + L_o)(R + h_o)}{m(R + h_o)\,V_o^2}\,\delta V + \frac{V_o}{(R+h_o)^2}\,\delta h - \frac{1}{mV_o}\,\delta L \qquad (126)$$

$$\delta\dot{h} = -\,V_o\;\delta\;\gamma \qquad (127)$$

The uncertainties in g, m, R, and $\left.\frac{\partial f(L)}{\partial L}\right|_o$ require us to use an adaptive controller. Of course, if approximate values are known they should be used as an initial trial in any iterative identification process. In matrix form

$$\dot{\underline{x}} = A\,\underline{x} + \underline{b}\,u \qquad (128)$$

where

$$u = \delta\,L$$

$$\underline{x} = \begin{bmatrix} \delta\,V \\ \delta\,\gamma \\ \delta\,h \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & a_{12} & 0 \\ a_{21} & 0 & a_{23} \\ 0 & a_{32} & 0 \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix}$$

and

$$a_{12} = g$$

$$a_{21} = \frac{-mV_o^2 + (mg + L_o)(R+h_o)}{m(R+h_o)\,V_o^2}$$

$$a_{23} = \frac{V_o}{(R+h_o)^2}$$

$$a_{32} = -V_o$$

$$b_1 = -\frac{1}{m}\left.\frac{\partial f(L)}{\partial L}\right|_o$$

$$b_2 = -\frac{1}{mV_o}$$

At any time instant, a_{ij} and b_i are treated as constants over a short time interval. Such an assumption is valid if the coefficients are changing slowly. The signal flow graph for (128) is shown in Figure 48.

Reducing the signal flow graph we obtain

$$\frac{\delta h}{\delta L} = \frac{b_2\, a_{32}\, s + b_1\, a_{21}\, a_{32}}{s\left(s^2 - a_{12}\, s - a_{32}\, a_{23}\right)}$$

Making δL to be a staircase signal as appearing from a digital controller, we obtain the discrete input-output transfer function.

$$\frac{\delta h(z)}{\delta L(z)} = \frac{\beta_1 z^2 + \beta_2 z + \beta_3}{z^3 + \alpha_1 z^2 + \alpha_2 z + \alpha_3}$$

The coefficients α_i, β_i must be identified through the identification process.[§] It is noted here that if only $\left.\frac{\partial f(L)}{\partial L}\right|_o$ (a function of density) is uncertain, only the numerator coefficients will be uncertain. Thus, only the numerator coefficients need to be estimated and the identification procedure will be greatly simplified. Upon knowing these coefficients the controller scheme discussed in Chapter 3 can be applied. The bounds on δL are obtained from

$$L_{min} \le \delta L + L_o \le L_{max}$$

Or,

$$L_{min} - L_o \le \delta L \le L_{max} - L_o$$

[§] We know that the numerator is at most a quadratic since the initial value of the response is zero.

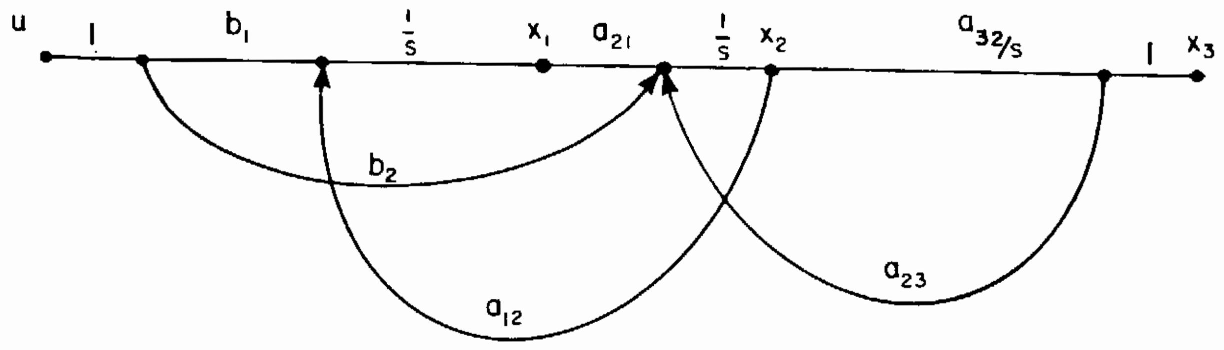


Figure 48. Signal Flow Graph for Linearized Process

The criterion function for this problem would be

$$J = \frac{1}{2} \sum_{j=k+1}^{k+N} \delta h(j)^2$$

This type of problem has been termed "tracking" problem. The desired path is known a priori, and the function of the controller is to keep the process close to this path. Besides the above illustration, one can envision many control problems that fall in this category.

CHAPTER 8

SUMMARY AND SUGGESTED EXTENSIONS

8.1 Summary

The major concern of Part 1 is the development of tools necessary to perform adaptation in a control problem with an unknown process. The approach taken to perform adaptive control was to measure the process through observation of the input-output data and to compute optimal controls on the basis of estimated parameter values and estimated state-variables. Therefore, there are three phases to this approach to adaptive controls.

1) Parameter Estimation
2) State-Variable Estimation
3) Computation of Optimal Controls

The three phases were studied separately indicating approaches which can accomplish these tasks.

In the area of optimal control computations, methods presently available were summarized. These methods are for the linear process case with quadratic performance criterion. Next, extension was made to the case with inequality constraints on the control variable. For this case quadratic programming methods using a gradient method were found to be suitable. The philosophy of employing an optimization interval for a finite time into the future was verified through computer simulation on an example. Because feedback was employed the technique showed experimentally that it can tolerate at least 10% error in the parameter values. Formulation was given also to handle constraints on both the amplitude and the rate of change of the control variable.

For the parameter estimation phase two approaches were studied: 1) the explicit mathematical relation method and 2) the learning model method. For the explicit mathematical relation method the recursive method of Greville was adopted to give estimated parameter values. Tools necessary for the statistical problem of assigning confidence intervals were given. For the learning model approach, a modified Newton's Method was presented and verified experimentally. Convergence considerations were given. Experimental comparison with Margolis' approach was made in terms of speed and noise-handling capabilities. With added computer complexity, experimental results verified the superior performance characteristics of the modified Newton's procedure.

For the state variable estimation phase, Kalman's recursive filtering technique was adopted.

Finally, an outline was given in Chapter 7 to apply the optimal-adaptive approach to a phase of the re-entry problem.

8.2 Suggestions for Further Research

Of course, a study of three phases of the problem does not imply completion. There is still the problem of tying all phases together to see whether the combination will work. Such a task would take an extensive programming effort. When application is eventually contemplated this task will have to be undertaken.

For the more immediate extensions, one can, for example, verify experimentally the case with bounds both on the magnitude and the rate-of-change of the control variable. From the practical point of view this case seems to be the most realistic.

For the on-line computer optimization, it seems that the coordinate-wise gradient method and Ho's simplified gradient projection method are the two feasible methods. As an extension a comparison of the two procedures can be made.

Analog methods suggested in Section 3.14 can be tried for the quadratic programming problem. The task here requires hybrid computational capability.

More extensive stability studies can be made for the on-line controller employed. Some considerations were given to the case without inequality constraints. No analytical methods were given for the case with inequality constraints. Although stability problems were not encountered in the experiments, possible situations may arise especially when the optimization interval is shortened. Other problem areas include computation time lag and error in parameter knowledge.

The simulation of the on-line controller (Section 3.7) revealed that the responses were slightly underdamped. Possibly one can choose different weighting in the criterion function to improve the response. Adding a term which weights the use of control energy is a definite possibility. These considerations can also be given to the example in Section 2.8 for the problem without inequality constraints.

For the explicit mathematical relation method, experimental studies can be made so that a direct comparison with the learning model approach can be made.

For the learning model approach, only block processing was employed experimentally. Analyzing intervals into the past from the present time at every sampling instant is another possibility. Iterations per observation interval could then possibly be reduced to a single iteration. The effectiveness of the identification is dependent upon the input signal employed. One could possibly attempt to use signals which more closely resemble signals present at the input of the process. If signals present at the input to the process do not give satisfactory results, then one should consider injection of suitable signals. Also, no analytical statistical considerations were given for the learning model approach.

For the general identification area, one can make a comparison between determining the weighting function and determining the coefficients of the difference equation. It is generally believed that the computer demand is less for the latter problem. But it would be of interest to determine the actual difference in the computer requirements of the two approaches.

The techniques outlined in this report require a digital computer for computations. Before these techniques can be applied the numerical computations must be translated into computer requirements (time and space). Some considerations were given in Chapter 3. Considerations could be extended to other chapters.

Finally, computer verification is needed for the application to the re-entry problem before serious consideration can be given.

REFERENCES

1. Kalman, R. E., "Design of a Self-Optimizing Control System", ASME Trans., 80:468-478, February 1958.

2. Merriam, C. W., "Use of a Mathematical Error Criterion in the Design of Adaptive Control Systems", AIEE Trans., Part II, 79:506-512, January 1960.

3. Braun, L., Jr., On Adaptive Control Systems, Doctoral Dissertation, Polytechnic Institute of Brooklyn, 1959.

4. Meditch, J. S. and Gibson, J. E., "On the Real-Time Control of Time Varying Linear Systems", IRE Trans. on Automatic Control, AC-7, No. 4:3-10, July 1962.

5. Hsieh, H. C., "On the Synthesis of Adaptive Controls by the Hilbert Space Approach", Report No. 62-19, Department of Engineering, University of California, Los Angeles, June 1962.

6. Zadeh, L. A., "On the Definition of Adaptivity", Proc. of the IEEE, 51:469-470, March 1963.

7. Cooper, G. R., et al, "Survey of the Philosophy and the State of the Art of Adaptive Systems", Tech. Report No. 1, Contract AF33(616)-6890, PRF 2358, School of Elec. Engr., Purdue University, Lafayette, Indiana, July 1960.

8. Aseltine, J. A., Mancini, A. R., and Sarture, C. W., "A Survey of Adaptive Control Systems", IRE Trans. on Automatic Control, AC-6:102-108, December 1958.

9. Mishkin, E. and Braun, L., Jr., Adaptive Control Systems, McGraw-Hill Book Co., New York, 1961.

10. Stear, E. B. and Gregory, P. C., "Capabilities and Limitations of Some Adaptive Techniques", Proc. of 1962 National Aerospace Electronics Conf., 644-660, Dayton, Ohio, 1962.

11. Horton, W. F. and Elsner, R. W., "An Adaptive Technique", Report ADR-556, Lear Siegler, Inc., Santa Monica, California, March 1963.

12. Whitaker, H. P., Yarnom, J., and Kezar, A., "Design of Model-Reference Adaptive Control Systems of Aircraft", Report R-164, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, September 1958.

13. Donalson, D. D., The Theory and Stability Analysis of a Model Referenced Parameter Tracking Technique for Adaptive Automatic Control Systems, Doctoral Dissertation, Department of Engineering, University of California, Los Angeles, California, 1961.

14. Margolis, M., On the Theory of Process Adaptive Control Systems, the Learning Model Approach, Doctoral Dissertation, Department of Engineering, University of California, Los Angeles, California, 1959.

15. Greville, T. N. E., "Some Applications of the Pseudo-Inverse of a Matrix" SIAM Review, 2:15-22, 1960.

16. Kalman, R. E., Englar, T. S., and Bucy, R. S., "Fundamental Study of Adaptive Control Systems", Tech. Report No. ASD-TR-61-27, Vol. 1, Aeronautical Systems Division, Air Force System Command, Wright-Patterson Air Force Base, Ohio, April 1962.

17. Kuhn, H. W. and Tucker, A. W., "Nonlinear Programming", Proc. Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 481-492, 1960.

18. Penrose, R., "A Generalized Inverse of Matrices", Proc. Cambridge Phil. Soc., 51:406-413, 1955.

19. Penrose, R., "On Best Approximate Solutions of Linear Matrix Equations", Proc. Cambridge Phil. Soc., 52:17-19, 1956.

20. Chang, S. S. L., Synthesis of Optimal Control Systems, McGraw-Hill Book Co., New York, 1961.

21. Katz, S., "A Discrete Version of Pontryagin's Maximum Principle", Journal of Electronics and Control, 13:179-184, August 1962.

22. Kipiniak, W., Dynamic Optimization and Control, MIT Press and John Wiley and Sons, Inc., New York, 1961.

23. Joseph, R. D., and Tou, J. D., "On Linear Control Theory", AIEE Trans. Part II, Applications and Industry, 80:193-196, September 1961.

24. Gunckel, T. L., II, and Franklin, G. F., "A General Solution for Linear, Sampled-Data Control", Proc. of 1962 Joint Automatic Control Conference, New York University, New York, June 1962.

25. Florentin, J. J., "Partial Observability and Optimal Control", Journal of Electronics and Control, 13:263-379, September 1962.

26. Schultz, P. R., Optimal Control in the Presence of Measurement Errors and Random Disturbances, Doctoral Dissertation, Department of Engineering, University of California, Los Angeles, California, 1963.

27. Bertram, J. E., "The Direct Method of Lyapunov in the Analysis and Design of Discrete-Time Control Systems", Work Session on Lyapunov's Second Method, edited by L. F. Kazda, University of Michigan, 79-104, Ann Arbor, 1960.

28. Hsieh, H. C., Synthesis of Adaptive Control Systems by the Function Space Methods, Doctoral Dissertation, Department of Engineering, University of California, Los Angeles, California, 1963.

29. Horing, S., "On the Optimum Design of Predictive Control Systems", paper presented at 1962 WESCON, Los Angeles, California, August 1962.

30. Ho, Y. C. and Brentani, P. B., "On Computing Optimal Control with Inequality Constraints", paper presented at the Symposium on Multivariable Control Systems, Boston, Massachusetts, November 1962.

31. Ho, Y. C., "Solution Space Approach to Optimal Control Problems", ASME, Journal of Basic Engr., 83:53-58, March 1961.

32. Hildreth, C., "Quadratic Programming Procedure", Naval Research Logistics Quarterly, 4:79-85, 1957.

33. D'Esopo, D. A., "A Convex Programming Procedure", Naval Research Logistics Quarterly, 1:33-42, 1959.

34. Eggleston, H. G., Convexity, Cambridge University Press, Cambridge, England, 1958.

35. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints", Journal of SIAM, 8:181-217, March 1960.

36. Levin, M. J., "Optimum Estimation of Impulse Response in the Presence of Noise", IRE Trans. on Circuit Theory, CT-7:50-56, March 1960.

37. Kerr, R. P. and Surber, W. H., "Precision of Impulse Response Identification Based on Short Normal Operating Records", IRE Trans. on Automatic Control, AC-6:173-182, May 1961.

38. Balakrishnan, A. V., "Determination of Nonlinear System from Input-Output Data", presented at the Conf. on Identification and Representation Problems, Princeton University, March 1963.

39. Greenberg, H., "A Survey of Methods for Determining Stability Parameters of an Airplane from Dynamic Flight Measurements", NACA, TN-2340, April 1951.

40. Bigelow, S. C. and Ruge, H., "An Adaptive System Using Periodic Estimation of the Pulse Transfer Function", IRE National Convention Record, Part 4, 25-38, 1961.

41. Eykhoff, P., "Process-Parameter Estimation", to be published as a chapter in the book: R. H. Macmillan (edit.), Progress in Control Engineering, Vol. 2, London, Heywood & Co., 1963.

42. Gainer, P. A., "A Method for Computing the Effect of an Additional Observation on a Previous Least-Squares Estimate", NASA, TN-D-1599, November 1962.

43. Linnik, Y. V., Method of Least Squares and the Principles of the Theory of Observation, Pergamon Press, New York, 1961.

44. Margolis, M., On the Theory of Process Adaptive Control Systems, the Learning Model Approach, Report No. 60-32, Department of Engineering, University of California, Los Angeles, May 1960.

45. Staffanson, Forrest L., "Determining Parameter Corrections According to System Performance - A Method and its Application to Real-Time Missile Testing", Army Missile Test Center, White Sands Missile Range, Lab. Res. Report 20, July 1960.

46. Bellman, R., Kagiwada, H., and Kalaba, R., "A Computational Procedure for Optimal System Design and Utilization", Proc. Nat. Acad. of Sci., 48:1524-1528, September 1962.

47. Elkind, J. I., Green, D. M., and Starr, E. A., "Application of Multiple Regression Analysis to Identification of Time Varying Linear Dynamic Systems", IRE Trans. on Automatic Control, AC-8:163-166, April 1963.

48. Becker, J. V., "Re-Entry from Space", Scientific American, 204:49-57, 1960.

49. Wingrove, R. C., "A Survey of Atmosphere Re-Entry Guidance and Control Methods", paper presented at IAS meeting, New York, January 1963.

50. Bryson, A. E. and Denham, W. F., "Multivariable Terminal Control for Minimum Mean-Square Deviation from a Nominal Path", Proc. IAS Symp. on Vehicle Systems Optimization, Garden City, New York, November 1961.

51. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation", Journal of SIAM on Control, 1:193-223, 1963.

52. Ho, Y. C., "The Method of Least Squares and Optimal Filtering Theory", Memo RM-3329-PR, The Rand Corp., Santa Monica, California, October 1962.

APPENDIX I

NOTATION AND CONCISE STATEMENT OF PROBLEMS

An attempt has been made to keep the notation consistent throughout Part 1.

I.1 Notation of Process Variables

The notation used for the single-input single-output process is given in Figure 49.

In terms of the state variables, the notation is given in Figure 50. In equation form

$$\begin{aligned} \underline{x}(k) &= \Phi(k, k-1)\, \underline{x}(k-1) + \Gamma(k)\, \underline{u}(k) + \Xi(k)\, \underline{w}(k) \\ \underline{z}(k) &= H(k)\, \underline{x}(k) + \underline{v}(k) = \underline{y}(k) + \underline{v}(k) \end{aligned} \qquad (129)$$

where

$\underline{x}(k)$ - n x 1 state vector

$\underline{y}(k)$ - m x 1 output vector

$\underline{z}(k)$ - m x 1 measured output vector

$\underline{u}(k)$ - r x 1 control vector

$\underline{w}(k)$ - q x 1 uncontrollable input vector

$\underline{v}(k)$ - m x 1 uncorrelated noise vector

Φ - n x n transition matrix

Γ - n x r matrix

H - m x n matrix

Ξ - n x q matrix

I. 2 Concise Statement of the Problems

In this section the problems treated in Part 1 will be stated in a concise form.

Problem I. 1: Given

i) process defined by (129).

ii) $\underline{z}(k)$

iii) $\underline{v}(k)$, $\underline{w}(k)$ - rough estimates of the variances can probably be given.

iv) some elements of Φ and Γ are known; for other elements possibly statistical characteristics can be given. (Changes usually occur slowly).

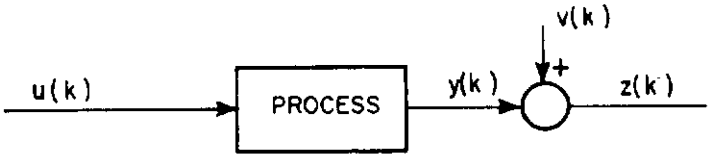


Figure 49. Single Input - Single Output Process

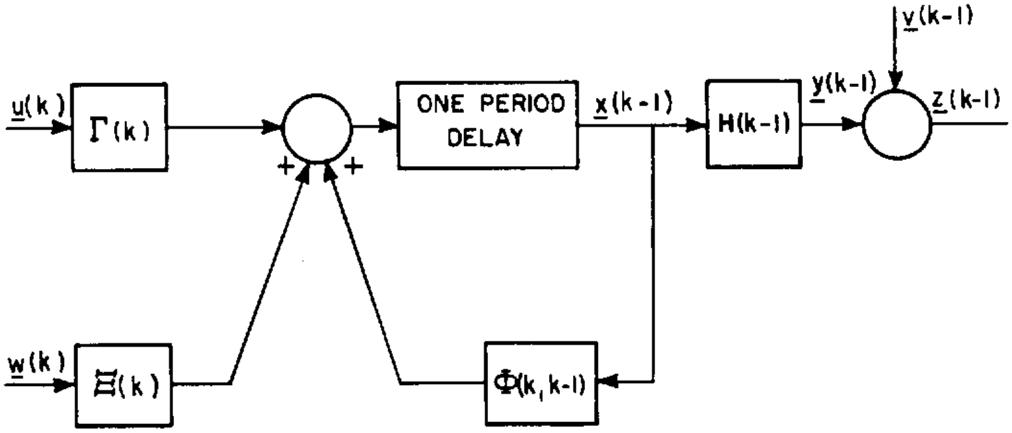


Figure 50. Process in Terms of State Variables

v) Ξ and H are known

vi) $\underline{y}_d(k)$ - s x 1 vector (s ≤ m); given for $k = 0, 1, 2, \ldots, N_1$

Determine the sequence $\underline{u}(k)$, $k = 1, 2, \ldots, N_1$ which minimizes

$$p = \sum_{k=1}^{N_1} \left|\left|\underline{y}_d(k) - Y\,\underline{y}(k)\right|\right|_Q^2$$

where Y - known s x m matrix

Q - known positive definite matrix

Subject to the constraints

$$|u_i(k)| \leq M \quad i\ 1, 2, \ldots, r;\ k=1, \ldots, N_1$$

Because Problem I.1 is difficult to solve we split the problem into several parts. At every sampling instant an optimization over a short interval of time into the future is performed (for simplicity of discussion we treat the single-input, single-output case).

Problem I.2 (Chapter 3): Given

i) process defined by

$\underline{x}(k) = \Phi\,\underline{x}(k-1) + \underline{\gamma}\,u(k)$

where Φ, $\underline{\gamma}$ are known

ii) initial conditions $\underline{x}(0)$ (known)

iii) $y_d(j)$ $j=k, k+1, \ldots, k+N$, $N < N_1$

Determine $u(j)$ $j = k+1, \ldots, k+N$ which minimizes

$$J = \sum_{j=k+1}^{k+N} \left(y_d(j) - Y\,\underline{y}(j)\right)^2$$

Y - 1 x n matrix

Subject to $|u(j)| \leq M$ $j = k+1, \ldots, k+N$

We can add other constraints.

Problem I.3 (Chapter 3): same as Problem I.2 except we add the constraint

$$|u(j) - u(j-1)| \leq M' \quad j = k+1, \ldots, k+N$$

In Problems I.2 and I.3 we assume that we know Φ, $\underline{\gamma}$, and $\underline{x}(k)$. The Φ and $\underline{\gamma}$ are estimated through identification; and $\underline{x}(k)$ is obtained through state estimation. Although the two problems are tied together we choose to separate them. For the identification problem we solve:

Problem I. 4 (Chapters 4 and 5): Given

i) the process form

$$\underline{x}_1(k) = \Phi_1 \; \underline{x}_1(k-1) + \underline{\gamma}_1 \; u(k)$$

$$\underline{z}(k) = H \; \underline{x}_1(k) \quad + \underline{\eta}(k)$$

where $\underline{\eta}(k)$ is correlated noise

$$\underline{x}_1(k), \; \Phi_1, \; \underline{\gamma}_1$$

as shown in Figure 45.

ii) $u(j)$ known $j = k, \; k-1, \ldots, k-N$

iii) $\underline{z}(j)$ known $j = k, \; k-1, \ldots, k-N$

Determine Φ_1 and $\underline{\gamma}_1$ (assuming they are constants over the observation interval)

The estimation problem can be stated in the following way.

Problem I. 5 (Chapter 6): Given

i) a random process defined by

$$\underline{x}_2(k) = \Phi_2(k, k-1) \; \underline{x}_2(k-1) + \Gamma_2(k) \; \underline{w}(k)$$

$$\underline{y}_2(k) = H_2(k) \; \underline{x}_2(k)$$

where Φ_2, Γ_2, H_2 are known (Figure 45)

$\underline{v}(k)$ and $\underline{w}(k)$ - uncorrelated Gaussian noise with known variances

$y_1(k)$ is a known deterministic sequence

ii) $\underline{z}(j)$ $j = 0, 1, \ldots, k$ (present time)

Determine an estimate $\hat{y}(k)$ which minimizes

$$E\left(y(k) - \hat{y}(k) \right)^2$$

APPENDIX II

A BRUTE-FORCE METHOD FOR THE QUADRATIC PROGRAMMING PROBLEM

This appendix outlines the non-iterative method of solving the quadratic programming problem of Chɛ ter 3. Only the two- and three-dimensional cases are considered.

II. 1 Two-Dimensional Case

For the two-dimensional case, (32) reduces to

$$\underset{\sim}{d} = u(1)\, \underset{\sim}{g}_1 + u(2)\, \underset{\sim}{g}_2 \tag{130}$$

with

$$|u(k)| \leq M$$

Equation (130) can be rewritten in terms of unit vectors.

$$\underset{\sim}{d} = \alpha_1 \underset{\sim}{i}_1 + \alpha_2 \underset{\sim}{i}_2$$

where

$$\underset{\sim}{i}_k = \frac{\underset{\sim}{g}_k}{||\underset{\sim}{g}_k||}$$

$$\alpha_k = u(k)\ ||\underset{\sim}{g}_k||$$

with

$$|\alpha_k| \leq M_k$$

$$M_k = M\ ||\underset{\sim}{g}_k||$$

Let us look at Figure 51. The total planar space shown in Figure 51 is the space spanned by the linear combination of $\underset{\sim}{i}_1$ and $\underset{\sim}{i}_2$. Since there are bounds on α_k we are restricted to operate over region R which is a parallelogram. The problem then is to approximate $\underset{\sim}{d}'$ as closely as possible by a point $\underset{\sim}{d}$ in R.

If $\underset{\sim}{d}'$ lies in R then we have the unbounded case and the solution is easy as we can invert a triangular matrix. Now, if $\underset{\sim}{d}'$ lies outside of R, two possibilities occur. If $\underset{\sim}{d}'$ lies in the unshaded region, then the optimum point is obtained by making a projection on one of the edges of the parallelogram. If $\underset{\sim}{d}'$ lies in the shaded region, the optimum point is at a vertex.

After these observations, let us see how we could solve the problem. The discussion will be restricted to the sector defined by $\alpha_1 > 0$, $\alpha_2 > 0$ shown in the top-left sector of Figure 51. The same considerations hold true for the other sectors; also, the technique should also apply whether the vertex is obtuse or acute.

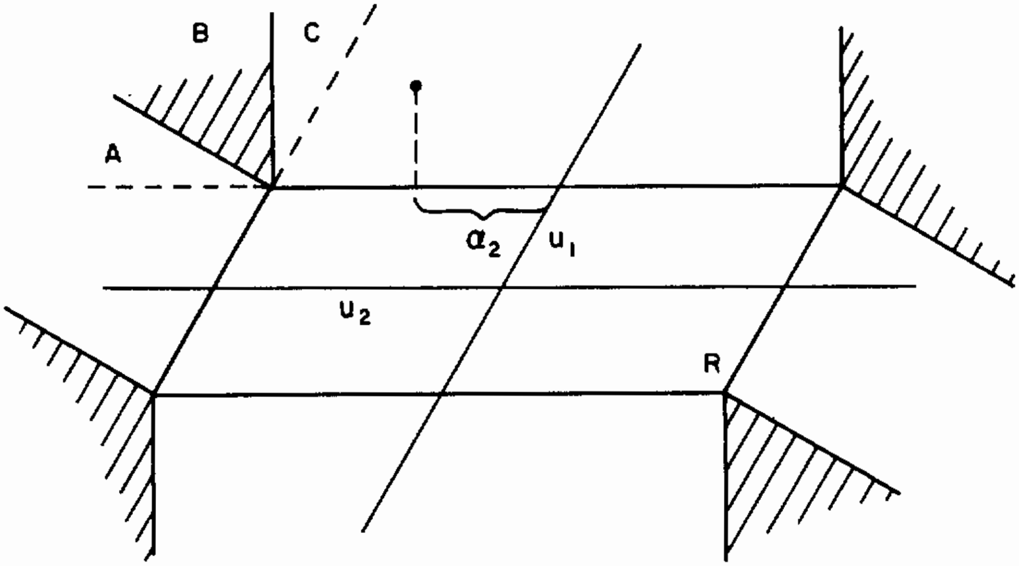


Figure 51. Two-Dimensional Case

First, let us compute the unbounded solutions, α_1' and α_2', i.e.,

$$\underset{\sim}{d}' = \alpha_1' \underset{\sim}{i}_1 + \alpha_2' \underset{\sim}{i}_2$$

Then, we can make tests via the digital computer to see whether any of these α_i' exceeds M_i. If neither exceeds their bounds we have no problem so we will not consider this case. We have three cases to consider.

Case 1: $\alpha_1' > M_1$ $\alpha_2' \leq M_2$

Case 2: $\alpha_1' \leq M_1$ $\alpha_2' > M_2$

Case 3: $\alpha_1' > M_1$ $\alpha_2' > M_2$

Cases 1 and 2 present no problem because we can project $\underset{\sim}{d}'$ on the edge which is exceeded. The projection may exceed the vertex of the parallelogram in which case we take that vertex as the solution. Now, for Case 3, we see that it can be in one of the sectors A, B, or C. If it is in A or C the optimum point is on one of the edges; while if it is in B the optimum point is at the vertex.

Because of these possibilities, if α_1' and α_2' exceed their bounds we must project onto both edges. If either of the projections lands on the edge we have the optimum point. If both projections exceed the vertex, the vertex is the optimum point.

At least for the two-dimensional case the above tests can be readily implemented on the computer.

For the case shown in the figure the point of projection is determined by the condition for othogonality

$$< \underset{\sim}{d}' - M_1 \underset{\sim}{i}_1 - \alpha_2 \underset{\sim}{i}_2, \underset{\sim}{i}_2 > = 0$$

or,

$$\alpha_2 = < \underset{\sim}{d}', \underset{\sim}{i}_2 > - M_1 < \underset{\sim}{i}_1, \underset{\sim}{i}_2 >$$

and $\alpha_1 = M_1$ is the other component. The solutions are then

$$u(1) = \frac{\alpha_1}{||\underset{\sim}{g}_1||}$$

$$u(2) = \frac{\alpha_2}{||\underset{\sim}{g}_2||}$$

which are the control forces to be applied in succession if the optimization interval does not change during application.

II. 2 Three-Dimensional Case

For the three-dimensional case let us again find the unbounded solutions.

$$\underset{\sim}{d}' = \alpha_1' \underset{\sim}{i}_1 + \alpha_2' \underset{\sim}{i}_2 + \alpha_3' \underset{\sim}{i}_3$$

Again, we omit the case when the $|\alpha_i| \leq M$. We have the following cases. Also, as before, we consider the positive sector only.

Case 1:	$\alpha_1' > M_1$,	$\alpha_2' \leq M_2$,	$\alpha_3' \leq M_3$
Case 2:	$\alpha_1' \leq M_1$,	$\alpha_2' > M_2$,	$\alpha_3' \leq M_3$
Case 3:	$\alpha_1' \leq M_1$,	$\alpha_2' \leq M_2$,	$\alpha_3' > M_3$
Case 4:	$\alpha_1' > M_1$,	$\alpha_2' > M_2$,	$\alpha_3' \leq M_3$
Case 5:	$\alpha_1' > M_1$,	$\alpha_2' \leq M_2$,	$\alpha_3' > M_3$
Case 6:	$\alpha_1' \leq M_1$,	$\alpha_2' > M_2$,	$\alpha_3' > M_3$
Case 7:	$\alpha_1' > M_1$,	$\alpha_2' > M_2$,	$\alpha_3' > M_3$

The following observations were made after building a parallelopiped. Cases 1, 2, and 3 give no trouble as we can immediately conclude that respectively, $\alpha_1 = M_1$, $\alpha_2 = M_2$, $\alpha_3 = M_3$, and we can obtain the solution by projection on the sides (planes) which are exceeded. For cases 4, 5, and 6 we have two components that exceed the bounds. Here, we have to project $\underset{\sim}{d}'$ onto the sides of the parallelopiped which were exceeded by the two components. From the projections we can make conclusions as in the two-dimensional case. Case 7 is the most troublesome one. We first must project $\underset{\sim}{d}'$ onto each of three sides. We can draw some conclusions if any of the projections reveal that some projected components are less than the bounds. However, there is still the case when the three projections reveal that the projected components all exceed the bounds. In this latter case we have to take α_i two at-a-time and project $\underset{\sim}{d}'$ onto the edges (line) of the parallelopiped. If the projection on the edge exceeds the bound then we can conclude that the optimum point is at the vertex. Otherwise, the optimum point is on the edge obtained by projection on the edge.

We have seen how the problem has grown from the two-dimensional case. We can imagine how difficult the four-dimensional case will be. Because of these developments we are led to gradient methods.

APPENDIX III

QUADRATIC PROGRAMMING THEOREMS

Let us consider the following general quadratic programming problem.

Problem III. 1 Find the n-vector $\underline{u}$ which minimizes

$$j(\underline{u}) = \underline{u}^* C\underline{u} + \underline{h}^* \underline{u} \tag{131}$$

subject to a convex region defined by

$$D\underline{u} - \underline{b} \geq 0 \tag{132}$$

where

C known n x n positive definite matrix
D known m x n matrix
$\underline{h}$ known n vector
$\underline{b}$ known m vector

First, we show that a unique minimum exists for the problem.

Lemma III. 1 A unique solution to Problem III. 1 exists.

The existence is assured from the fact that $J(\underline{u})$ is bounded from below and the region of feasible solution is closed and non-empty.

For uniqueness, we first note that $J(\underline{u})$ is a positive definite quadratic form; therefore, it is a convex function. Let us assume non-uniqueness, and let $\underline{u}_1$ and $\underline{u}_2$ be two distinct minima. Because of convexity

$$J(\underline{u}) < J_{min} \text{ for } \underline{u} = \alpha \underline{u}_1 + (1-\alpha) \underline{u}_2 \text{ with } 0 < \alpha < 1$$

Point $\underline{u}$ is along a line between u_1 and u_2; and it is in the feasible region because of convexity of the region in $\underline{u}$. Therefore, by contradiction there is a unique solution.

The statements to follow are special cases of the more general theorems given by Kuhn and Tucker (Reference 17). It is rederived to fit more closely the problem we have. First, we give a sufficient condition for a minimum.

Theorem III. 1 Saddle Point Theorem (Sufficient Only)

If for the above problem we can find an n vector $\underline{u}^o$ and an m vector $\underline{\lambda}^o$ such that $\underline{u}^o$, $\underline{\lambda}^o$ forms a saddle point of the Lagrangian

$$\phi(\underline{u}, \underline{\lambda}) = J(\underline{u}) - \underline{\lambda}^* (D\underline{u} - b) \text{ for } \underline{\lambda} \geq 0$$

i.e., (133)

$$\phi(\underline{u}, \underline{\lambda}^o) \geq \phi(\underline{u}^o, \underline{\lambda}^o) \geq \phi(\underline{u}^o, \underline{\lambda})$$

then $\underline{u}^o$ is a minimum of $J(\underline{u})$ for $D\underline{u} - \underline{b} \geq 0$.

Proof: Since $\underline{u}^o$, $\underline{\lambda}^o$ is a saddle point,

$$J(\underline{u}) - \underline{\lambda}^{o*}(D\underline{u}-\underline{b}) \geq J(\underline{u}^o) - \underline{\lambda}^{o*}(D\underline{u}^o-\underline{b}) \geq J(\underline{u}^o) - \underline{\lambda}^{*}(D\underline{u}^o-\underline{b})$$

Since the right-hand inequality is true for $\lambda \geq 0$,

$$\underline{\lambda}^{o*}\,(D\underline{u}^o-\underline{b}) \leq 0$$

But,

$$\underline{\lambda}^{o*}\,(D\underline{u}^o-\underline{b}) \geq 0$$

Therefore,

$$\underline{\lambda}^{o*}\,(D\underline{u}^o-\underline{b}) = 0$$

The left-hand inequality becomes

$$J(\underline{u}) - \underline{\lambda}^{o*}\,(D\underline{u}-\underline{b}) \geq J(\underline{u}^o)$$

Since

$$\underline{\lambda}^{o*}\,(D\underline{u}-\underline{b}) \geq 0,$$

$$J(\underline{u}) \geq J(\underline{u}^o)$$

Thus, if we can find a saddle point then we are assured of a unique minimum. It is noted that the saddle point may or may not be a distinct point. This fact, however, is not important to us. Next, let us give sufficient conditions for a saddle point.

Lemma III. 2 The following conditions are sufficient for the existence of a saddle point. (Equations 134 and 135 are also necessary conditions but this fact is unimportant.)

1) $$\nabla_u\, \phi\Big|_o = 0 \tag{134}$$

2) $$\nabla_\lambda\, \phi\Big|_o \leq 0, \quad \nabla_\lambda\, \phi\Big|_o^* \underline{\lambda}^o = 0, \quad \underline{\lambda}^o \geq 0 \tag{135}$$

3) $$\phi(\underline{u}, \underline{\lambda}^o) \geq \phi(\underline{u}^o, \underline{\lambda}^o) + \nabla_u\, \phi\Big|_o^* (\underline{u}-\underline{u}^o) \tag{136}$$

4) $$\phi(\underline{u}^o, \underline{\lambda}) \leq \phi(\underline{u}^o, \underline{\lambda}^o) + \nabla_\lambda\, \phi\Big|_o^* (\underline{\lambda}-\underline{\lambda}^o) \tag{137}$$

Proof: Using (136) and (134)

$$\phi(\underline{u}, \underline{\lambda}^o) \geq \phi(\underline{u}^o, \underline{\lambda}^o)$$

Using (137) and (135)

$$\phi(\underline{u}^o, \underline{\lambda}) \leq \phi(\underline{u}^o, \underline{\lambda}^o) + \nabla_\lambda \cdot \phi\Big|_o^* \underline{\lambda}$$

but

$$\nabla_\lambda\, \phi\Big|_o \leq 0$$

Therefore,

$$\phi(\underline{u}^o, \underline{\lambda}) \leq \phi(\underline{u}^o, \underline{\lambda}^o)$$

We will prove another lemma which will be useful in the theorem to follow.

Lemma III. 3 If $\phi(\underline{x})$ is convex, then

$$\phi(\underline{x}) \ge \phi(\underline{x}^o) + \nabla_x \phi\Big|_o^* (\underline{x}-\underline{x}^o)$$

(If $\phi(\underline{x})$ is concave, the inequality is reversed.)

Proof: Using the definition for convex functions, i.e.,

$$(1-\theta)\ \phi(\underline{x}^o) + \theta\ \phi(\underline{x}) \ge \phi\Big((1-\theta)\ \underline{x}^o + \theta\, \underline{x}\Big)$$

with

$$0 \le \theta \le 1$$

$$\phi(\underline{x}) - \phi(\underline{x}^o) \ge \frac{\phi\Big((1-\theta)\ x^o + \theta\ x\Big) - \phi(x^o)}{\theta}$$

In the limit as $\theta \to 0$.

$$\phi(\underline{x}) \ge \phi(\underline{x}_o) + \nabla_x \phi\Big|_o^* (\underline{x}-\underline{x}^o)$$

The existence of a saddle point is assured by the following theorem.

Theorem III. 2 For Problem III. 1 the following are necessary and sufficient conditions:

1) $$\nabla_u \phi\Big|_o = 0 \tag{134}$$

or, $$\nabla_u J - D^* \underline{\lambda}^o = 0 \tag{138}$$

2) $$\nabla_\lambda \phi\Big|_o \le 0, \qquad \nabla_\lambda \phi\Big|_o^* \underline{\lambda}^o = 0, \qquad \underline{\lambda}^o \ge 0 \tag{135}$$

or, $$D\underline{u}^o - b \ge 0,\ \underline{\lambda}^{o*}(D\underline{u}^o - \underline{b}) = 0, \qquad \underline{\lambda}^o \ge 0 \tag{139}$$

3) $$\phi(\underline{u}, \underline{\lambda}^o) \ge \phi(\underline{u}^o, \underline{\lambda}^o) + \nabla_u \phi\Big|_o^* (\underline{u}-\underline{u}^o) \tag{136}$$

4) $$\phi(\underline{u}^o, \underline{\lambda}) = \phi(\underline{u}^o, \underline{\lambda}^o) + \nabla_\lambda \phi\Big|_o^* (\underline{\lambda}-\underline{\lambda}^o) \tag{140}$$

Proof: (Necessary Part)

We prove (136) and (140) first. Forming the Lagrangian

$$\phi(\underline{u}, \underline{\lambda}) = J(\underline{u}) - \underline{\lambda}^*(D\underline{u}-\underline{b})$$

We know that $J(\underline{u})$ is convex. For a given $\underline{\lambda}$, the second term is linear in $\underline{u}$. Therefore, $\phi(\underline{u}, \underline{\lambda}^o)$ is convex in $\underline{u}$. Thus (136) follows from Lemma III. 3. For a given $\underline{u}$, $\phi(\underline{u}, \underline{\lambda})$ is linear in $\underline{\lambda}$. Therefore, using Taylor's theorem (140) follows.

To show (134) and (135) let us note that the inequality can be replaced by

$$D\underline{u} - \underline{b} = \underline{s}^2 \tag{141}$$

Now, performing the usual optimization on

$$\psi(\underline{u}, \underline{\lambda}, \underline{s}) = J(\underline{u}) - \underline{\lambda}^*(D\underline{u}-\underline{b}-\underline{s}^2)$$

Taking the partials with respect to each of the variables,

$$\nabla_u \psi\Big|_o = \nabla_u \phi\Big|_o = 0$$

$$\nabla_\lambda \psi\Big|_o = -(D\underline{u}-\underline{b}-\underline{s}^2) = 0$$

or

$$\nabla_\lambda \phi\Big|_o = -\underline{s}^2 \qquad \text{or,} \qquad \nabla_\lambda \phi\Big| \le 0$$

$$\nabla_s \psi\Big|_o = 2\,\underline{s}^*\,\underline{\lambda}^o = 0 \qquad \text{or,} \qquad \underline{s}^*\,\underline{\lambda}^o = 0$$

Multiplying (141) by λ^o, we get

$$\nabla_\lambda \phi\Big|_o^* \underline{\lambda}^o = 0$$

There remains to show that $\underline{\lambda}^o \geq 0$. We are to satisfy m inequalities. There will be some inequalities which will be satisfied by equalities.

$$\sum_i d_{ji}\, u_i^o - b_j = 0 \qquad j = 1, \ldots, r. \tag{142}$$

There will be other inequalities which will be satisfied by strict inequalities.

$$\sum_i d_{ji}\, u_i^o - b_j > 0 \qquad j = r+1, \ldots, m. \tag{143}$$

For the strict inequality case, by (139) $\lambda_i = 0$ for $i = r+1, \ldots, m$.

Let us suppose that the λ_ν associated with one of (142) is non-positive, i.e., $\lambda_\nu \le 0$ where $0 \le \nu \le r$. Assuming that u_i^o is the minimum, let us take a point u_i slightly removed from u_i^o such that

$$\sum_i d_{ji}\, u_i - b_j = 0 \qquad j \neq \nu,\ j = 1, \ldots, r$$

$$\sum d_{ji}\, u_i - b_\nu > 0$$

We note that u_i is still in the constrained set. Multiplying (138) by $(\underline{u}-\underline{u}^o)$, we get

$$\nabla_u J\Big|_o^* (\underline{u}-\underline{u}^o) = \underline{\lambda}^*\, D^*(\underline{u}-\underline{u}^o)$$

$$= \lambda_\nu \Big(\sum d_{\nu i}\, u_i - b_\nu\Big)$$

Therefore, if $\lambda_\nu < 0$, then

$$\nabla_u J\Big|_o^* (\underline{u}-\underline{u}^o) < 0$$

If $\lambda_\nu = 0$, then

$$\nabla_u J \Big|_o^* (\underline{u}-\underline{u}) = 0$$

In both cases we have a contradiction, since we have a unique minimum. Therefore, $\lambda_\nu > 0$. Since ν is arbitrary, we see that, in general, the multipliers associated with the inequalities are non-negative. (Sufficiency) The conditions are also sufficient from Lemma III. 2 and Theorem III. 1.

APPENDIX IV

A RECURSIVE METHOD TO OBTAIN THE BEST ESTIMATE

In this appendix a recursive method is given to numerically determine the best estimate of parameters using the concept of the pseudo-inverse. The pseudo-inverse as defined by Penrose (References 18, 19) is used to solve a set of simultaneous algebraic equations when there are more equations than unknowns. Greville (Reference 15) gave a recursive method for the purpose of successively adding higher-order terms in the polynomial approximation problem. The question arose whether one could use Greville's method for the estimation problem when one desires to update the estimate as new data arrive. We show in this appendix that one can indeed use his method.

Some new lemmas are shown in this appendix which facilitate the derivation of the algorithm. We start directly with the axioms and lemmas given by Penrose. This route presents the derivation with less insight.

The pseudo-inverse is defined as that matrix, $A^{\dagger}$, which satisfies

$$A\,A^{\dagger}\,A = A \tag{144}$$

$$A^{\dagger}\,A\,A^{\dagger} = A^{\dagger} \tag{145}$$

$$(A\,A^{\dagger})^{*} = A\,A^{\dagger} \tag{146}$$

$$(A^{\dagger}\,A)^{*} = A^{\dagger}\,A \tag{147}$$

Several identities follow immediately as shown by Penrose. These identities are stated as lemmas.

Lemma IV. 1 a) $A^{*}\,A\,A^{\dagger} = A^{*}$ (148)

b) $A^{\dagger}\,A\,A^{*} = A^{*}$ (149)

Lemma IV. 2 a) $A^{*}\,A^{\dagger *}\,A^{\dagger} = A^{\dagger}$ (150)

b) $A^{\dagger}\,A^{\dagger *}\,A^{*} = A^{\dagger}$ (151)

Lemma IV. 3 $A^{\dagger\dagger} = A$ (152)

Lemma IV. 4 $A^{*}\,A = 0 \rightarrow A = 0$ (153)

It is noted that the inverse $[A^{*}\,A]^{-1}$ exists if and only if columns of A are linearly independent.

In the following discussion we will work with the equation

$$\underset{\sim}{y}_k = A_k\,\underset{-}{x}_k \tag{154}$$

where

$\underline{y}_k$ - k x 1 (given)

A_k - k x m (given) $\quad k > m$

$\underline{x}_k$ = m x 1 (unknown to be determined)

The problem is to find x_k by

$$\hat{\underline{x}}_k = A_k^{\dagger}\, \underline{y}_k \tag{155}$$

This represents the best-estimate after k instants of time. Each instant of time has a new set of data. Let us partition A_k in the following manner.

$$A_k = \left(\begin{array}{c} A_{k-1} \\ \hdashline \underline{a}_k^{*} \end{array}\right) \quad \begin{array}{c} k-1 \times m \\ \\ 1 \times m \end{array} \tag{156}$$

where $\underline{a}$ represents the new set of data. The pseudo-inverse, $A_k^{\dagger}$, can also be partitioned.

$$A_k^{\dagger} = \left(\begin{array}{c:c} B_k & \underline{b}_k \\ m \times k-1 & m \times 1 \end{array}\right) \tag{157}$$

Before we derive the algorithm for computation it is convenient to give some lemmas.

Lemma IV. 5 $\quad A_{k-1} A_k^{\dagger} A_k = A_{k-1}$ (158)

Proof: From (144)

$$A_k A_k^{\dagger} A_k = A_k$$

$$\left(\begin{array}{c} A_{k-1} \\ \hdashline \underline{a}_k^{*} \end{array}\right) A_k^{\dagger} A_k = \left(\begin{array}{c} A_{k-1} \\ \hdashline \underline{a}_k^{*} \end{array}\right)$$

$$\left(\begin{array}{c} A_{k-1} A_k^{\dagger} A_k \\ \hdashline \underline{a}_k^{*} A_k^{\dagger} A_k \end{array}\right) = \left(\begin{array}{c} A_{k-1} \\ \hdashline \underline{a}_k^{*} \end{array}\right)$$

Therefore,

$$A_{k-1} A_k^{\dagger} A_k = A_{k-1}$$

Lemma IV. 6 $\quad A_k^{\dagger} A_k A_{k-1}^{\dagger} = A_{k-1}^{\dagger}$ (159)

Proof: From (150) and (158) $A_k^{*} A_k^{\dagger *} A_{k-1}^{*} A_{k-1}^{\dagger *} A_{k-1}^{\dagger} = A_{k-1}^{\dagger}$

Using (147) $A_k^{\dagger} A_k A_{k-1}^{\dagger} A_{k-1} A_{k-1}^{\dagger} = A_{k-1}^{\dagger}$

From (145) $A_k^{\dagger} A_k A_{k-1}^{\dagger} = A_{k-1}^{\dagger}$

Lemma IV.7 $B_k A_{k-1} A_{k-1}^{\dagger} = B_k$ (160)

From (148) $A_k^{*} A_k B_k = A_{k-1}^{*}$ *

From (145) $A_k^{\dagger} A_k B_k = B_k$ **

Substituting * in (148) $A_k^{*} A_k B_k A_{k-1} A_{k-1}^{\dagger} = A_k^{*} A_k B_k$

Premultiply by $A_k^{\dagger} A_k^{\dagger *}$ and using (151)

$$A_k^{\dagger} A_k B_k A_{k-1} A_{k-1}^{\dagger} = A_k^{\dagger} A_k B_k$$

Using ** $B_k A_{k-1} A_{k-1}^{\dagger} = B_k$

Let us now derive the algorithm. First, multiply (157) and (156)

$$A_k^{\dagger} A_k = B_k A_{k-1} + \underline{b}_k \underline{a}_k^{*} \qquad (161)$$

Postmultiply by $A_{k-1}^{\dagger}$

$$A_k^{\dagger} A_k A_{k-1}^{\dagger} = B_k A_{k-1} A_{k-1}^{\dagger} + \underline{b}_k \underline{a}_k^{*} A_{k-1}^{\dagger}$$

Using Lemmas IV.6 and IV.7

$$A_{k-1}^{\dagger} = B_k + \underline{b}_k \underline{a}_k^{*} A_{k-1}^{\dagger} \qquad (162)$$

Therefore,

$$A_k^{\dagger} = \left(A_{k-1}^{\dagger} - \underline{b}_k \underline{a}_k^{*} A_{k-1}^{\dagger} \ \vdots \ \underline{b}_k \right) \qquad (163)$$

The task remains to find $\underline{b}_k$ Let us form $A_k^{\dagger} A_k$ from (163) and (156)

$$A_k^{\dagger} A_k = A_{k-1}^{\dagger} A_{k-1} - \underline{b}_k \underline{a}_k^{*} A_{k-1}^{\dagger} A_{k-1} + \underline{b}_k \underline{a}_k^{*}$$

Or,

$$A_k^{\dagger} A_k = A_{k-1}^{\dagger} A_{k-1} + \underline{b}_k \underline{c}_k^{*} \qquad (164)$$

where

$$\underline{c}_k^* = \underline{a}_k^* - \underline{a}_k^* A_{k-1}^\dagger A_{k-1} \qquad (165)$$

Again we divert from the main path to prove some more lemmas which will be useful later.

Lemma IV.8 $A_{k-1}\, \underline{c}_k^{\dagger *} = 0, \quad \text{if } \underline{c}_k \neq 0$ (166)

Proof:

First, it can be shown easily that $\underline{c}_k^* A_{k-1}^\dagger = 0$. From (165)

$$\underline{c}_k^* = \underline{a}_k^* - \underline{a}_k^* A_{k-1}^\dagger A_{k-1}$$

Post multiply by $A_{k-1}^\dagger$,

$$\underline{c}_k^* A_{k-1}^\dagger = \underline{a}_k^* A_{k-1}^\dagger - \underline{a}_k^* A_{k-1}^\dagger = 0$$

Post multiply by $A_{k-1} A_{k-1}^*$,

$$\underline{c}_k^* A_{k-1}^\dagger A_{k-1} A_{k-1}^* = 0$$

From (149)

$$\underline{c}_k^* A_{k-1}^* = 0 \qquad *$$

From (149)

$$\underline{c}_k^* \underline{c}_k \underline{c}_k^\dagger = \underline{c}_k^*$$

Substitute in *,

$$\underline{c}_k^* \underline{c}_k \underline{c}_k^\dagger A_{k-1}^* = 0$$

Since $\underline{c}_k \neq 0$, $\underline{c}_k^* \underline{c}_k \neq 0$

$$\underline{c}_k^\dagger A_{k-1}^* = 0$$

Taking transpose

$$A_{k-1}\, \underline{c}_k^{\dagger *} = 0$$

Lemma IV.9 $\underline{a}_k^* \underline{c}_k^{\dagger *} = 1, \quad \text{if } \underline{c}_k \neq 0$ (167)

Proof: From (148) $\underline{c}_k^* \underline{c}_k \underline{c}_k^\dagger = \underline{c}_k^*$

Post multiply by $\underline{c}_k$,

$$\underline{c}_k^* \underline{c}_k \underline{c}_k^\dagger \underline{c}_k = \underline{c}_k^* \underline{c}_k$$

Since $\underline{c}_k^* \, \underline{c}_k \neq 0$,

$$\underline{c}_k^\dagger \, \underline{c}_k = 1 = \underline{c}_k^* \, \underline{c}_k^{\dagger *}$$

Post multiply (165) by $c_k^{\dagger *}$

$$\underline{c}_k^* \, \underline{c}_k^{\dagger *} = \underline{a}_k^* \, \underline{c}_k^{\dagger *} - \underline{a}_k^* A_{k-1}^\dagger \, A_{k-1} \, \underline{c}_k^{\dagger *}$$

Using Lemma IV.8

$$\underline{a}_k^* \, \underline{c}_k^{\dagger *} = 1$$

Lemma IV.10

$$A_{k-1}^\dagger \, A_{k-1} \, \underline{b}_k = \underline{b}_k, \text{ if } |A_{k-1}^* \, A_{k-1}| > 0 \tag{168}$$

Proof: Let us start with an identity,

$$A_{k-1}^* \, A_{k-1} \, \underline{b}_k = A_{k-1}^* \, A_{k-1} \, \underline{b}_k$$

Substitute (148)

$$A_{k-1}^* \, A_{k-1} \, A_{k-1}^\dagger \, A_{k-1} \, \underline{b}_k = A_{k-1}^* \, A_{k-1} \, \underline{b}_k$$

if $|A_{k-1}^* \, A_{k-1}| > 0$,

$$A_{k-1}^\dagger \, A_{k-1} \, \underline{b}_k = \underline{b}_k$$

In the determination of $\underline{b}_k$ we have two cases to consider, 1) $\underline{c}_k \neq 0$ and 2) $\underline{c}_k = 0$.

Case 1)[§] $\underline{c}_k \neq 0$

Consider the matrix

$$P_k = A_{k-1}^\dagger \, A_{k-1} + \underline{c}_k^{\dagger *} \, \underline{c}_k^* \qquad\qquad *$$

Premultiply by $\underline{a}_k^*$

$$\underline{a}_k^* \, P_k = \underline{a}_k^* \, A_{k-1}^\dagger \, A_{k-1} + \underline{a}_k^* \, \underline{c}_k^{\dagger *} \, \underline{c}_k^*$$

Using Lemma IV.9

$$\underline{a}_k^* \, P_k = \underline{a}_k^* \, A_{k-1}^\dagger \, A_{k-1} + \underline{c}_k^*$$

Substitute (165),

$$\underline{a}_k^* \, P_k = \underline{a}_k^*$$

[§] This is the case when the columns of A are not linearly independent.

Premultiply * by A_{k-1}

$$A_{k-1} P_k = A_{k-1} A_{k-1}^{\dagger} A_{k-1} + A_{k-1} \underline{c}_k^{\dagger *} \underline{c}_k^{*}$$

Using (144) and Lemma IV.8

$$A_{k-1} P_k = A_{k-1}$$

Therefore,

$$\begin{pmatrix} A_{k-1} \\ - - - \\ \underline{a}_k^{*} \end{pmatrix} P_k = \begin{pmatrix} A_{k-1} \\ - - - \\ \underline{a}_k^{*} \end{pmatrix}$$

Or,

$$A_k P_k \quad A_k$$

Thus, P_k has the property of $A_k^{\dagger} A_k$. Or,

$$P_k = A_k^{\dagger} A_k$$

Let us consider * and (164)

$$A_k^{\dagger} A_k = A_{k-1}^{\dagger} A_{k-1} + \underline{b}_k \underline{c}_k^{*} \tag{164}$$

$$A_k^{\dagger} A_k = A_{k-1}^{\dagger} A_{k-1} + \underline{c}_k^{\dagger *} \underline{c}_k^{*} \qquad *$$

Or,

$$\underline{b}_k \underline{c}_k^{*} = \underline{c}_k^{\dagger *} \underline{c}_k^{*}$$

If we post multiply by $\underline{c}_k$, it is seen that

$$\underline{b}_k = \underline{c}_k^{\dagger *} = c_k \left(c_k^{*} c_k \right)^{-1} \tag{169}$$

The (-1) represents division in this situation.

Case 2)

$$\underline{c}_k = 0$$

From Equation (165),

$$\underline{a}_k^{*} = \underline{a}_k^{*} A_{k-1}^{\dagger} A_{k-1}$$

To simplify the writing let us define

$$\underline{d}_k^{*} = \underline{a}_k^{*} A_{k-1}^{\dagger} \tag{170}$$

Therefore, if $\underline{c}_k = 0$ from (165)

$$\underline{a}_k^{*} = \underline{d}_k^{*} A_{k-1} \tag{171}$$

Let us form the sub-matrix of $A_k A_k^{\dagger}$ obtained by deleting the last row and last column. From (156) and (163)

$$G_k = A_{k-1} A_{k-1}^{\dagger} - A_{k-1} \underline{b}_k \underline{d}_k^{*}$$

Since G_k and $A_{k-1} A_{k-1}^{\dagger}$ are symmetric (146), it follows that the last term is also symmetric. Since $A_{k-1} \underline{b}_k$ is a column matrix and $\underline{d}_k^{*}$ is a row matrix, it follows that

$$A_{k-1} \underline{b}_k = h \underline{d}_k \tag{172}$$

where h is some scalar to be determined. From (156) and (163)

$$A_k A_k^{\dagger} = \left(\begin{array}{c|c} A_{k-1} A_{k-1}^{\dagger} - h \underline{d}_k \underline{d}_k^{*} & A_{k-1} \underline{b}_k \\ \hline d_k^{*} - \underline{a}_k^{*} \underline{b}_k \underline{d}_k^{*} & \underline{a}_k^{*} \underline{b}_k \end{array}\right)$$

Using (172) and (171),

$$A_k A_k^{\dagger} = \left(\begin{array}{c|c} A_{k-1} A_{k-1}^{\dagger} - h \underline{d}_k \underline{d}_k^{*} & h \underline{d}_k \\ \hline \underline{d}_k^{*} - h \underline{d}_k^{*} \underline{d}_k \underline{d}_k^{*} & h \underline{d}_k^{*} \underline{d}_k \end{array}\right)$$

Because of symmetry and the fact that $\underline{d}_k^{*} \underline{d}_k$ is a scalar,

$$h \underline{d}_k = \underline{d}_k - h \underline{d}_k \underline{d}_k^{*} \underline{d}_k$$

Therefore,

$$h = \left(1 + \underline{d}_k^{*} \underline{d}_k\right)^{-1}$$

From (172),

$$A_{k-1}^{\dagger} A_{k-1} \underline{b}_k = h A_{k-1}^{\dagger} \underline{d}_k$$

Using Lemma IV.10

$$\underline{b}_k = \left(1 + \underline{d}_k^{*} \underline{d}_k\right)^{-1} A_{k-1}^{\dagger} \underline{d}_k$$

Since $\underline{d}_k^{*} = \underline{a}_k^{*} A_{k-1}^{\dagger}$, we have

$$\underline{b}_k = \left(1 + \underline{a}_k^{*} A_{k-1}^{\dagger} A_{k-1}^{\dagger *} \underline{a}_k\right)^{-1} A_{k-1}^{\dagger} A_{k-1}^{\dagger *} \underline{a}_k \tag{173}$$

It is noted again that (-1) signifies division.§

§ It is noted that this division always exists.

We are more interested in the solution, $\underline{x}_k$, instead of the pseudoinverse. It is desired to solve $\underline{x}_k$ in terms of $\underline{x}_{k-1}$. Let

$$\underline{y}_k = \left(\begin{array}{c} \underline{y}_{k-1} \\ \hdashline y_k \end{array}\right)$$

where y_k - last data

$\underline{y}_{k-1}$ - k - 1 x 1 vector

Then,

$$A^{\dagger}_{k-1} \underline{y}_{k-1} = \hat{x}_{k-1}$$

and,

$$\hat{\underline{x}}_k = A^{\dagger}_k \underline{y}_k = \left(A^{\dagger}_{k-1} - \underline{b}_k \underline{a}_k^{*} A^{\dagger}_{k-1} \;\vdots\; \underline{b}_k\right) \left(\begin{array}{c} \underline{y}_{k-1} \\ \hdashline y_k \end{array}\right)$$

$$\hat{\underline{x}}_k = \hat{\underline{x}}_{k-1} - \underline{b}_k \underline{a}_k^{*} \hat{\underline{x}}_{k-1} + \underline{b}_k y_k \tag{174}$$

It is noted that in no place A_k and $A^{\dagger}_{k-1}$ are required; but the quantities $A_k^{\dagger} A_k$ and $A_k^{\dagger} A_k^{\dagger *}$ are required. Therefore, a great savings of computer storage space can be made if we generate the latter quantities. From (164) and (163),

$$A^{\dagger}_k A_k = A^{\dagger}_{k-1} A_{k-1} + \underline{b}_k \underline{c}_k^{*} \tag{164}$$

$$\begin{aligned} A^{\dagger}_k A^{\dagger *}_k = {} & A^{\dagger}_{k-1} A^{\dagger *}_{k-1} - \underline{b}_k \underline{a}_k^{*} A^{\dagger}_{k-1} A^{\dagger *}_{k-1} \\ & - A^{\dagger}_{k-1} A^{\dagger *}_{k-1} (\underline{b}_k \underline{a}_k^{*})^{*} + \underline{b}_k \underline{a}_k^{*} A^{\dagger}_{k-1} A^{\dagger *}_{k-1} (\underline{b}_k \underline{a}_k^{*})^{*} \\ & + \underline{b}_k \underline{b}_k^{*} \end{aligned} \tag{175}$$

Let us summarize the important equations. The flow chart for the computation is shown in Figure 52.

$$\underline{c}_k = \underline{a}_k^{*} - \underline{a}_k^{*} A^{\dagger}_{k-1} A_{k-1} \tag{165}$$

Case 1: $\underline{c}_k \neq 0$

$$\underline{b}_k = \underline{c}_k \left(\underline{c}_k^{*} \underline{c}_k\right)^{-1} \tag{169}$$

Case 2: $\underline{c}_k = 0$

$$\underline{b}_k = \left(1 + \underline{a}_k^{*} A^{\dagger}_{k-1} A^{\dagger *}_{k-1} \underline{a}_k\right)^{-1} A^{\dagger}_{k-1} A^{\dagger *}_{k-1} \underline{a}_k \tag{173}$$

$$\hat{\underline{x}}_k = \hat{\underline{x}}_{k-1} - \underline{b}_k \underline{a}_k^{*} \hat{\underline{x}}_{k-1} + \underline{b}_k y_k \tag{174}$$

$$A^{\dagger}_k A_k = A^{\dagger}_{k-1} A_{k-1} + \underline{b}_k \underline{c}_k^{*} \tag{164}$$

$$A^{\dagger}_k A^{\dagger *}_k = (\underline{b}_k \underline{a}_k^{*} - I) A^{\dagger}_{k-1} A^{\dagger *}_{k-1} \left(I - (\underline{b}_k \underline{a}_k^{*})^{*}\right) + \underline{b}_k \underline{b}_k^{*} \tag{175}$$

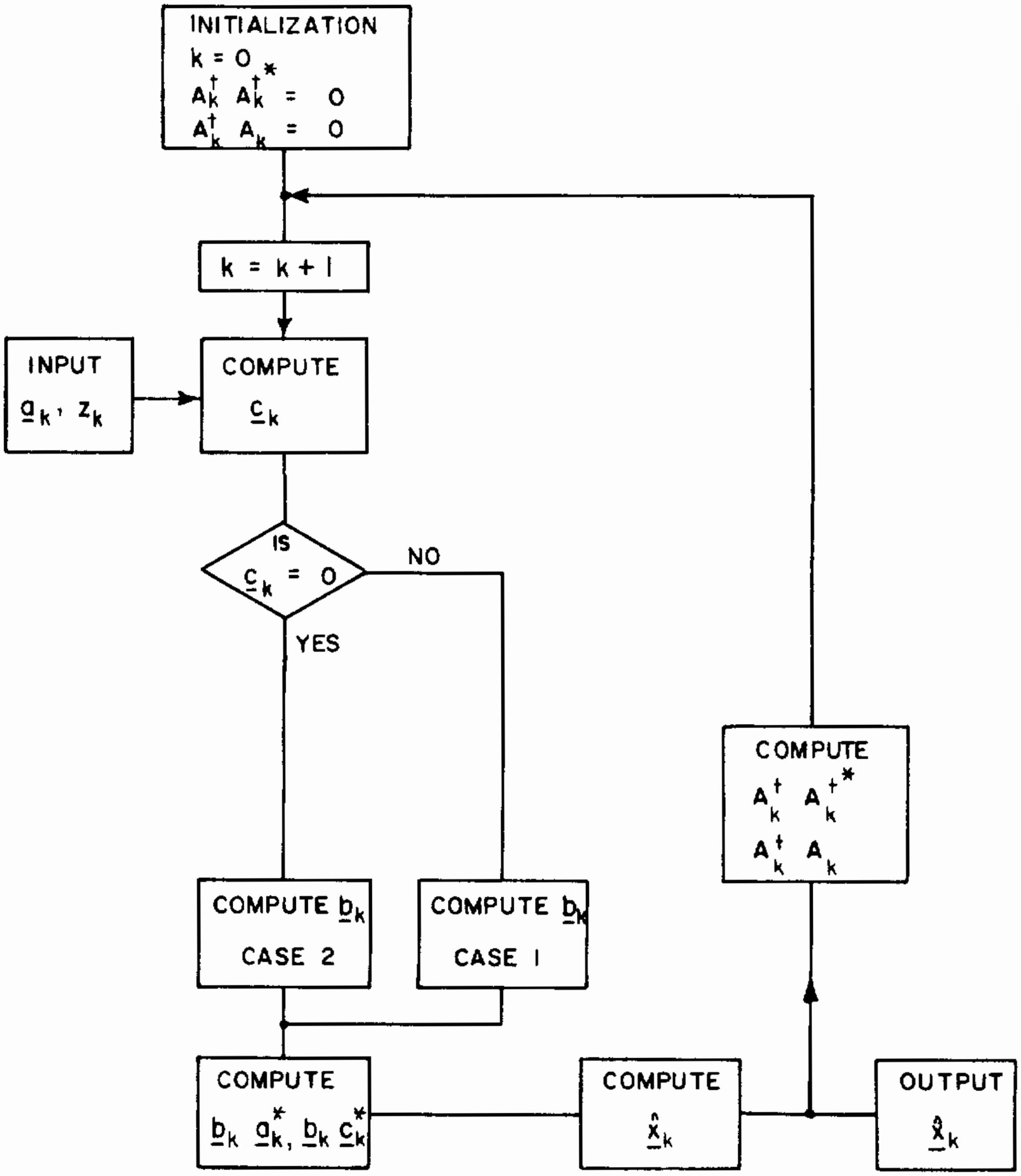


Figure 52. Flow Chart for Recursive Method

In conclusion, we remark that we have arrived at a method of computing the best estimate in terms of the previously calculated best estimate. This computation was also performed in a manner which saved computer storage and in a manner not requiring matrix inversion. It is also noted that the procedure will always give a solution since Penrose has shown the existence and uniqueness of the pseudo-inverse.

APPENDIX V

CORRESPONDENCE BETWEEN GREVILLE'S AND KALMAN'S RECURSIVE PROCEDURES

Although Greville's and Kalman's results were derived for seemingly different problem areas, the recursive procedures can be shown to be equivalent for certain conditions. Greville's routine given in Appendix IV of this report and Kalman's routine given in Reference 16 should be followed for the notation. The correspondence is not completely one-to-one in that Greville's routine is more general in one respect while Kalman's routine is more general in another respect.

Observability is the term used when A^*A is positive definite and $(A^*A)^{-1}$ exists. In this case the pseudo-inverse is given by

$$A^{\dagger} = (A^*A)^{-1} A^* \tag{176}$$

Greville's procedure is valid even for the unobservable case. It will be shown here that for the observable case Greville's procedure is equivalent to Kalman's procedure applied to the time-independent case. Kalman's routine is more general in the respect that for the observable situation the recursive routine can be extended to dynamic systems and the correlated case. Ho (Reference 52) has discussed the connection between least-squares theory and optimal filtering theory assuming that $(A^*A)^{-1}$ exists.

First, we show that for the observable case $\underline{c}_k = 0$.

Lemma V.1 If $(A^*A)^{-1}$ exists, then $\underline{c}_k = 0$. (Necessity)

From (165)

$$\underline{c}_k^{\,*} = \underline{a}_k^{\,*} - \underline{a}_k^{\,*} A_{k-1}^{\dagger} A_{k-1} \tag{165}$$

If $\left(A_{k-1}^{\;*} A_{k-1}\right)^{-1}$ exists then

$$A_{k-1}^{\dagger} = \left(A_{k-1}^{\;*} A_{k-1}\right)^{-1} A_{k-1}^{\;*}$$

The proof follows immediately upon substitution in (165).

For the case $\underline{c}_k = 0$ (Assuming Sufficiency) we show the equivalence of the recursive procedures. We have

$$\underline{b}_k = \left(1 + \underline{a}_k^{\,*} A_{k-1}^{\dagger} A_{k-1}^{\dagger *} \underline{a}_k\right)^{-1} A_{k-1} A_{k-1}^{\dagger *} \underline{a}_k^{\,*} \tag{173}$$

$$\begin{aligned} A_k^{\dagger} A_k^{\dagger *} = {} & A_{k-1}^{\dagger} A_{k-1}^{\dagger *} - \underline{b}_k \underline{a}_k^{\,*} A_{k-1}^{\dagger} A_{k-1}^{\dagger *} - A_{k-1}^{\dagger} A_{k-1}^{\dagger *} (\underline{b}_k \underline{a}_k^{\,*})^* \\ & \underline{b}_k \underline{a}_k^{\,*} A_{k-1}^{\dagger} A_{k-1}^{\dagger *} (\underline{b}_k \underline{a}_k^{\,*})^* + \underline{b}_k \underline{b}_k^{\,*} \end{aligned} \tag{175}$$

In Kalman's notation

$$K = \left(1 + H \sum (t/t-1)\, H^*\right)^{-1} \sum(t/t-1)\, H^* \tag{177}$$

$$\sum(t+1/t) = \sum(t/t-1) - KH \sum(t/t-1) - \sum(t/t-1)\, H^* K^*$$
$$+ KH \sum(t/t-1)\, H^* K^* + KK^* \tag{178}$$

The correspondence in notation is

Kalman:	Greville:
$\underline{\hat{x}}(t+1/t)$	$\underline{x}_k$
$K(t)$	b_k
$z(t)$	Y_k
$\underline{\hat{x}}(t/t-1)$	$\underline{\hat{x}}_{k-1}$
$H(t)$	$\underline{a}_k^*$
$\sum(t+1/t)$	$A_k^\dagger A_k^{\dagger *} = P_k$
$\sum(t/t-1)$	$A_{k-1}^\dagger A_{k-1}^{\dagger *}$

Looking at the last four terms in (178)

$$- \sum H^* \left(1 + H \sum H^*\right)^{-1} H \sum$$
$$- \sum H^* \left(1 + H \sum H^*\right)^{-1} H \sum$$
$$+ \sum H^* \left(1 + H \sum H^*\right)^{-1} H \sum H^* \left(1 + H \sum H^*\right)^{-1} H \sum$$
$$+ \sum H^* \left(1 + H \sum H^*\right)^{-1} \left(1 + H \sum H^*\right)^{-1} H \sum$$

$$= - \sum H^* \left[2 \left(1 + H \sum H^*\right)^{-1} - \frac{\left(1 + H \sum H^*\right)}{\left(1 + H \sum H^*\right)} \left(1 + H \sum H^*\right)^{-1} \right] H \sum$$

$$= - \sum H^* \left[\left(1 + H \sum H^*\right)^{-1} \right] H \sum$$

Therefore,

$$\sum(t+1/t) = \sum(t/t-1) - \sum(t/t-1)\, H^* \left[\left(H \sum(t/t-1)\, H^* + 1\right)^{-1} \right] H \sum(t/t-1) \tag{179}$$

Equation (179) corresponds with Equation III$_d$ of Kalman (Reference 16) (page 150a) for the case when $\Phi(t+1;t) = I$, $R(t) = I$, $Q(t) = C(t) = 0$.