IDENTIFICATION OF PARAMETERS IN DISTRIBUTED PARAMETER SYSTEMS

by

NORMAN STEPHEN BORDAN

B. Eng., Sir George Williams University, 1968

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

We accept this thesis as conforming to the

required standard

Research Supervisor..................................

Members of Committee ....

....

Head of Department ....

Members of the Department of

Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

May, 1971

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date 25th May 1971

ABSTRACT

This thesis deals with the identification of parameters in distributed parameter systems. Two sensitivity methods, namely; Meissinger's method and the method of structural sensitivity are extended to obtain the sensitivity coefficients of discretized distributed parameter systems. The method of Bingulac and Kokotovic is extended to identify parameters in the one and the two dimensional parabolic differential equations.

CSMP (continuous system modeling programme) is used throughout to simulate the systems.

Results for both sensitivity schemes are obtained, and it is found that although structural sensitivity is advantageous for parameter identification in ordinary differential equations, this is not the case for partial differential equations.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

Figure		Page
1.1	Process Optimization	2
1.2	Model Building	3
1.3	Parameter Optimization	5
2.1	Temporal Discretization	11
3.1	Three Loop Kokotovic System	19
3.2	Spatially Discretized Distributed	20
3.3	Kokotovic Distributed Parameter System	21
3.4	Structural Sensitivity	23
4.1	The Method of Bingulac and Kokotovic	27
4.2	Mode Control	27
4.3	Example #1 (α_1 versus α_2)	31
4.4	Example #2 (α_2 versus α_3)	32
4.5	Example #3 (α_1 versus α_2)	35
4.6	Example #3 (α_2 versus α_3)	36
4.7	Example #4 (α_1, α_2 versus no. of iterations)	37
4.8	Example #4 (α_3, α_4 versus no. of iterations)	38
4.9	Example #4 (α_5 versus no. of iterations)	39
4.10	Example #5 (α_1 versus α_2)	43
4.11	Example #5 (α_2 versus α_3)	44
4.12	Example #5 (β_1 versus β_2)	45
4.13	Example #5 (β_2 versus β_3)	46
B.1		52
B.2	Analogue Computer Implementation #1 of the Method of Bingulac and Kokotovic	53
B.3		54

Figure Page

ACKNOWLEDGEMENT

This work was supported by a National Research Council Scholarship in 1968-69 and by a student assistantship in 1969-71.

I am indebted to Dr. A.C. Soudack for his help and encouragement.

I would like to thank Dr. M.S. Davies for reading the manuscript and indicating improvements. Special thanks are due to Mr. Koit Teng for his help with CSMP.

Finally, I would like to thank Miss L. Morris for typing this thesis.

1. INTRODUCTION

1.1 Identification

In order to control a process, it is necessary to obtain a mathematical model of that process. The procedure by which this model is obtained is called identification. When the input and the response of the system are used in the identification procedure[7], rather than some test signal, the procedure is known as "on-line" identification. Many procedures exist for the identifying, tracking, and estimating of system states and (or) parameters[1-6].

Applications of parameter identification lie in the two areas of process optimization and model building. In process optimization, the parameters are chosen to optimize the response of a system with respect to a particular criteria. This can be seen in Figure 1.1.

For model building, one would like to determine a model of the system so that the model behaviour and that of the system approximate one another. The model parameters are chosen to accomplish the optimum match. This can be seen in Figure 1.2.

The model building approach is considered in this thesis. Let the unknown system parameter vector be A. A model is constructed with parameter vector replacing the unknown system parameter vector. The tracking parameter vector α is determined by minimizing an error function $f(e)$.

Many procedures exist to minimize an error function[12-17,27]. The gradient or steepest descent method[28] is used to identify the unknown parameter vector in this work. If γ is an element of α, the steepest descent equation is

$$\frac{d\gamma}{dt} = -K\frac{\partial f(e)}{\partial e}\frac{\partial e}{\partial \gamma} \qquad \text{for } K > 0 \qquad (1.1)$$

We will define e as the output error i.e. the error between the model

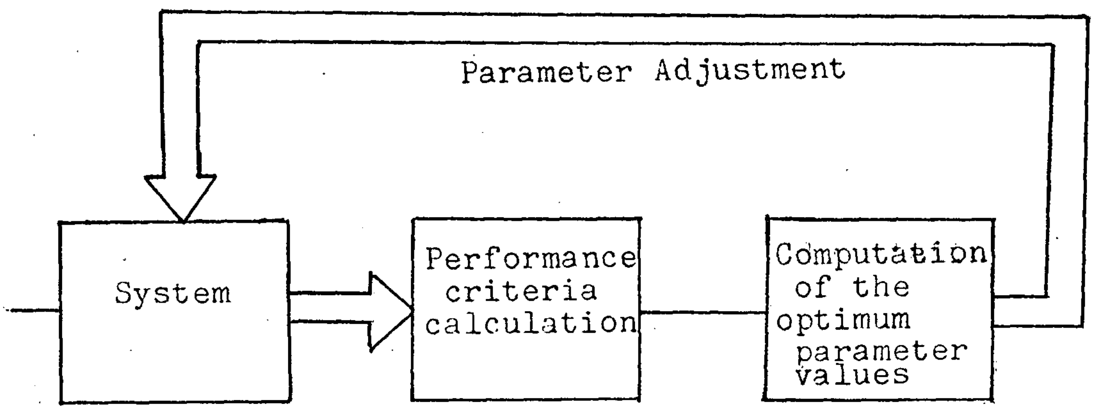

Parameter Adjustment
System
Performance criteria calculation
Computation of the optimum parameter values

Figure 1.1

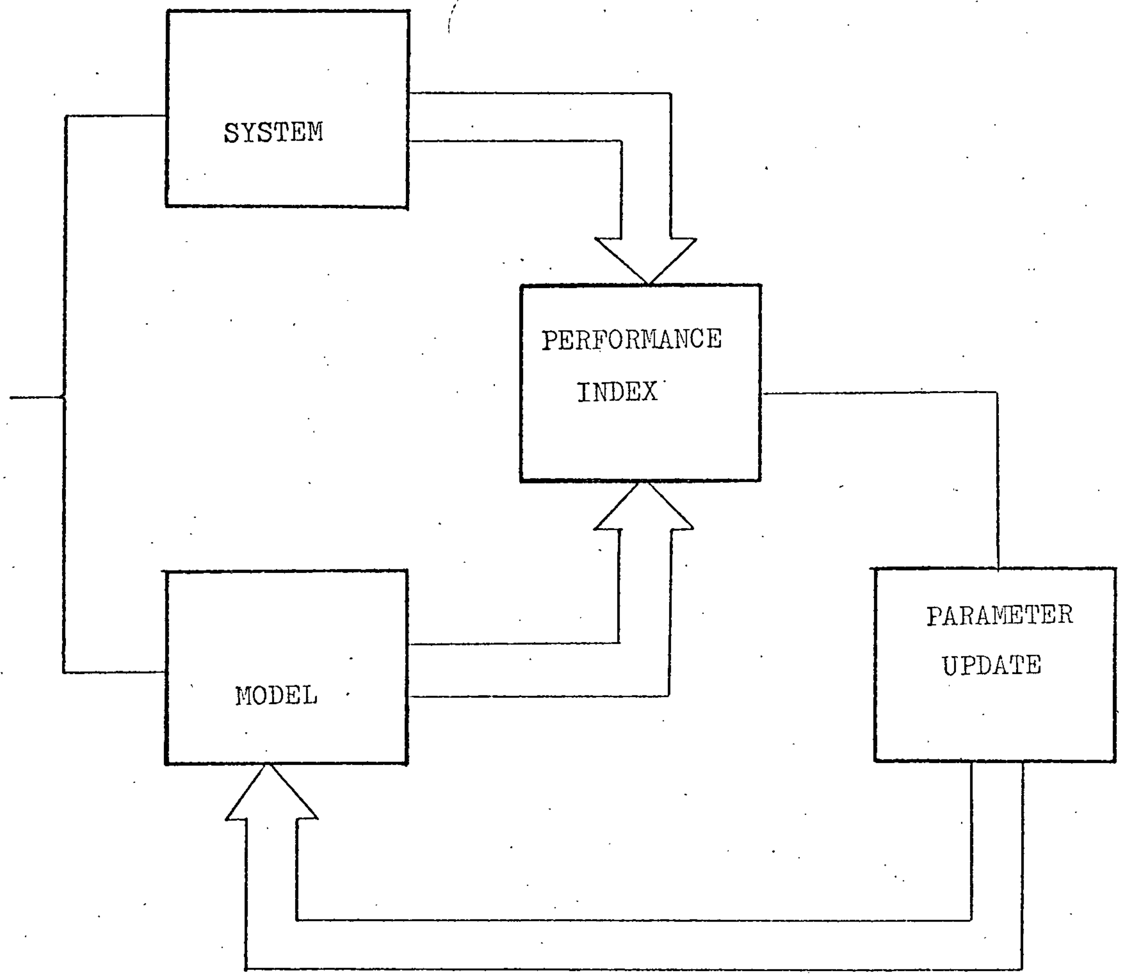


FIGURE 1.2

response Y_m and that of the system Y. Since the system response is not a function of α, $\partial e/\partial\alpha = \partial Y_m/\partial\alpha$.

The next section will introduce us to the methods for obtaining $\partial Y_m/\partial\alpha$, the gradient of the model response with respect to the parameters under study.

1.2 Sensitivity Analysis

Sensitivity analysis arose from the need to consider the deviations from some set value of the parameters in a control system. It is possible to establish an indirect relationship between the parameter variations and the resulting deviation of the object function. This relation is obtained by means of parameter sensitivity coefficients[11], which are defined as the gradient of the model response with respect to the parameters under study. Meissinger's method for obtaining the parameter sensitivity coefficients is the basis for the work which uses the model structure to generate the sensitivity coefficients[8-11].

With the knowledge of the sensitivity coefficients, the tracking parameters γ in equation (1.1) can be obtained, as will be seen in the next section.

1.3 Parameter Optimization

The parameters are obtained by using the iterative scheme developed by Bingulac and Kokotovic[26]. This method consists of successive iterations to determine the parameters which minimize the performance function. This is summarized in Figure 1.3.

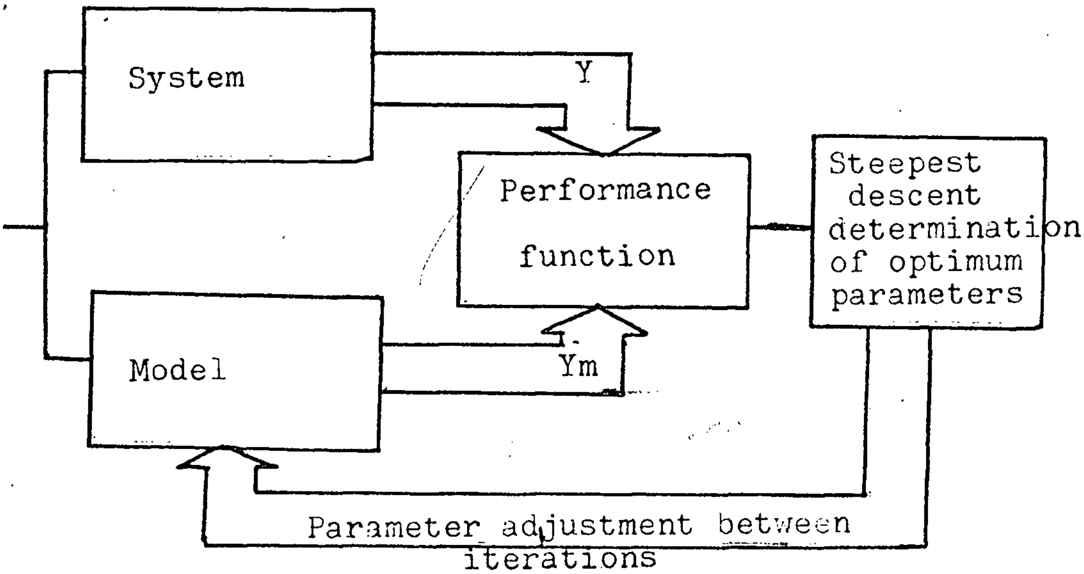


Figure 1.3

1.4 The Problem Considered

This thesis deals with the "on-line" identification of distributed parameter systems. The model building scheme is used, and so the form of the system is assumed known. The distributed parameter systems under study are the one and the two dimensional parabolic differential equations in which the unknown parameters are spatially dependent. The one dimensional parabolic differential equation with spatially dependent parameters is the heat equation for a rod which is constructed of an inhomogeneous medium, i.e. the specific heat of the material is spatially dependent. The two dimensional parabolic differential equation under consideration is the heat equation for a slab of inhomogeneous material. The object of the work is to identify the spatially dependent specific heat parameters.

1.5 Previous Work Done on Identifying Distributed Parameter Systems

Identification of distributed parameter systems has been attempted by various authors[18-23]. Perdreauville and Goodson[20] have extended

Shinbrot's method[21] to make it applicable to the identification of partial differential equations. This scheme consists of obtaining an integral transform in the spatial domain, which when applied to the partial differential equation and integrated by parts yields an algebraic equation. This method can not be applicable to a partial differential equation with spatially dependent parameters, because no integral transform can be found in the spatial domain.

Others[22,23] have identified parameters in distributed parameter systems by statistical means. An extensive literature search has been done and it has been found that no one has identified spatially dependent parameters in the parabolic differential equation.

1.6 Scope of the Thesis

Certain classes of ordinary differential equations have been identified using the method of Bingulac and Kokotovic. This method requires the sensitivity of the model response with respect to the parameters under study. Two methods are employed to obtain the sensitivity coefficients, namely, Meissinger's method, and that of structural sensitivity.

This thesis extends the method of Bingulac and Kokotovic, Meissinger's method; and the method of structural sensitivity to the identification of spatially dependent parameters in parabolic partial differential equations. The distributed parameter systems were discretized spatially and the above method were employed.

Chapter 2 deals with the schemes available to discretize partial differential equations.

Chapter 3 develops Meissinger's method and the method of structural sensitivity for discretized distributed parameter systems. It is found that structural sensitivity is of little use when applied to the systems under study.

Chapter 4 deals with the identification of distributed parameter systems using the method of Bingulac and Kokotovic and the sensitivity methods developed in Chapter 3. The S/360 continuous modeling programme used on the IBM 360/67 computer is applied to specific examples.

Chapter 5 consists of a recapitulation of the high points and a suggestion for future work.

2. DISCRETIZING PARTIAL DIFFERENTIAL EQUATIONS

2.1 Introduction

The purpose of this chapter is to introduce the various schemes available for discretizing partial differential equations, and to discuss the pros and cons of the methods studied.

Let us consider the one dimensional heat equation with the spatially dependent parameter $\alpha(x)$.

$$\frac{\partial q(x,t)}{\partial t} = \alpha(x) \frac{\partial^2 q(x,t)}{\partial x^2} \tag{2.1}$$

where $x\varepsilon(0,x_f)$ and $t\varepsilon(0,t_f)$

with boundary conditions

$$q(x,t)\big|_{x=0} = L(t) \qquad \text{and} \qquad q(x,t)\big|_{x=x_f} = M(t) \tag{2.2}$$

and initial conditions

$$q(x,t)\big|_{t=0} = K(x) \tag{2.3}$$

Equation (2.1) has two independent variable; the spatial variable x, and the temporal variable t. If we were to discretize equations (2.1) - (2.3) we could discretize either or both of these variables.

The basic discretization schemes are:

(i) Spatial discretization. The spatial variable is discretized.

(ii) Temporal discretization. The temporal variable is discretized.

(ii) Space time discretization. Both the spatial and the temporal variables are discretized.

Let us now consider the pros and cons of the various methods.

2.2 Spatial Discretization

To obtain the spatially discretized model we use the central differencing formula[25]

$$\frac{\partial q^2(x,t)}{\partial x^2} \simeq \frac{q_{j+1}(t) - 2q_j(t) + q_{j-1}(t)}{(\Delta x)^2} \tag{2.4}$$

where

$$q_{j+1}(t) = q(x + \Delta x, t)$$
$$q_j(t) = q(x,t)$$
$$q_{j-1}(t) = q(x - \Delta x, t)$$

The discretized version of equations (2.1) - (2.3) becomes

$$\dot{q}_j(t) = \alpha_j(q_{j+1}(t) - 2q_j(t) + q_{j-1}(t))/(\Delta x)^2 \tag{2.5}$$

where j=1, .., n

with boundary conditions

$$q_0(t) = L(t) \quad \text{and} \quad q_{n+1}(t) = M(t) \tag{2.6}$$

and initial conditions

$$q_j(t)\big|_{t=o} = K_j \tag{2.7}$$

We now have n first order equations which represent the partial differential equation. This can be represented by the following vector differential equation

$$\dot{Q} = A\lambda Q + ABQ_B$$

where Q is an (nxl) vector, A is an (nxn) matrix, λ is an (nxn) matrix, B is an (nx2 matrix, and Q_B is a (2xl) vector with the following form:)

$$Q = \begin{vmatrix} q_1 \\ \cdot \\ \cdot \\ q_n \end{vmatrix} ; \quad A = \begin{vmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \cdot & \\ & & & \cdot \\ & & & & \alpha_n \end{vmatrix} ; \quad \lambda = \begin{vmatrix} -2 & 1 & & 0 \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ 0 & & \ddots & 1 \end{vmatrix} ; \quad B = \begin{vmatrix} 1 & 0 \\ 0 & 0 \\ \cdot & \cdot \\ 0 & 1 \end{vmatrix}$$

$$Q_B = \begin{vmatrix} q_0 & q_{n+1} \end{vmatrix}^T$$

To implement this discretization scheme on the analogue computer one requires n integrators. The more accuracy one desires, the greater the number of integrators needed, since Δx gets smaller and therefore n gets larger.

2.3 Temporal Discretization

The temporal discretization scheme is arrived at by the backward differencing discretization

$$\frac{\partial q}{\partial t} \approx \frac{q_j(x) - q_{j-1}(x)}{\Delta t} \tag{2.10}$$

The discretized version of equations (2.1) - (2.3) becomes

$$\alpha(x)\frac{\partial^2 q_j(x)}{\partial x^2} = \frac{q_j(x) - q_{j-1}(x)}{\Delta t} \tag{2.11}$$

where $q_j = q(x_j,t)$ and $j=1, \ldots, n$

with boundary conditions

$$q_j(0) = L_j \quad \text{and } q_j(x_f) = M_j \tag{2.12}$$

and initial conditions

$$q_0(x) = K(x) \tag{2.13}$$

Temporal discretization requires that $q_{j-1}(x)$ be stored from the previous calculation so that $q_j(x)$ is the only unknown in equation (2.11). This is the reason why backward differencing is used in equation (2.10).

The above equations can be represented by the following vector differential equation

$$\Delta t\,\alpha(x)\frac{d^2Q}{dx^2} = AQ + ABQ_B \tag{2.14}$$

where

$$Q = \begin{vmatrix} q_1 \\ \cdot \\ \cdot \\ q_n \end{vmatrix}; \qquad A = \begin{vmatrix} 1 & 0 & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & & \\ & 0 & & -1 \end{vmatrix}$$

$$B = \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} \quad ; \qquad Q_B = \begin{vmatrix} q_0 & q_{n+1} \end{vmatrix}^T$$

Implementing the temporal discretization scheme on the analogue computer presents a few difficulties. Continuous memory is needed, since the

q_{j-1} function is required in the solution of equation (2.11). Since this is an analogue computer scheme, initial conditions on all the integrators must be obtained. Both the initial conditions on $\partial q/\partial x$ and q are not usually given and so they must be determined by using an iterative scheme.

This method employs a feedback loop with four operational amplifiers, as can be seen in Figure 2.1. If the loop gain exceeds unity, the circuit becomes unstable due to the positive feedback introduced into the circuit by the four operational amplifiers. The stability requirement is $1/\alpha(x)\Delta t<1$[25].

The difficulties just outlined make temporal discretization unattractive for use with this problem.

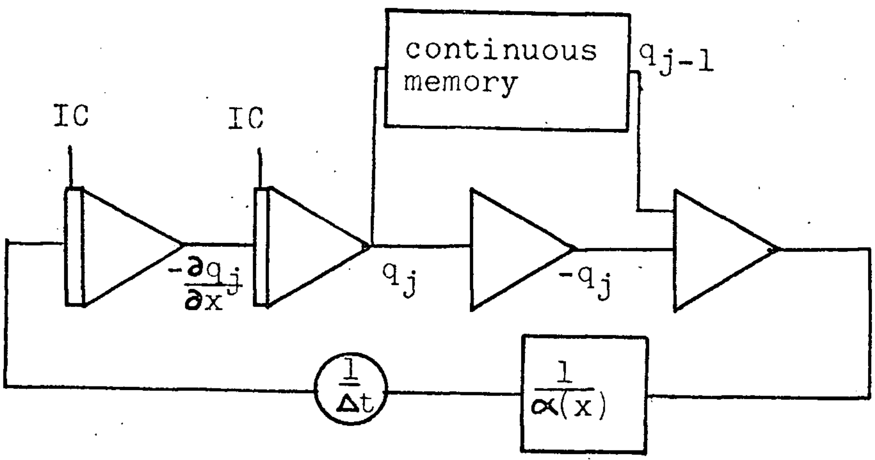


Figure 2.1

2.4 Space Time Discretization

If all the independent variables in a partial differential equation are approximated by finite difference expressions, the partial differential equation is replaced by a system of simultaneous algebraic equations which can be solved on a digital computer. The time derivative can either be forward or backward differenced.

Forward differencing yields

$$q_{i,j+1} - q_{i,j} = \alpha_i(q_{i+1,j} - 2q_{i,j} + q_{i-1,j})\Delta t/\Delta x^2 \quad (2.15)$$

with boundary conditions

$$q_{0j} = L_j \quad \text{and} \quad q_{n+1j} = M_j \quad (2.16)$$

and initial conditions

$$q_{i0} = K_i \quad (2.17)$$

Backward differencing yields[24].

$$q_{i,j+1} - q_{i,j} = \alpha_i(q_{i+1,j+1} - 2q_{i,j}+1 + q_{i-1,j+1})\Delta t/\Delta x^2 \quad (2.18)$$

The forward differencing method, i.e. equations (2.15) - (2.17), is usually solved explicitly on the digital computer. Forsythe and Wasow[29] have proven that the forward differencing method is stable and the discretization error is of the order of $(\Delta x)^2$ provided that $\alpha_i > 0$ and

$$\Delta t \leq \Delta x^2/2\alpha_i \quad (2.19)$$

If these relations are not satisfied, instability results due to the propagation of roundoff and truncation errors.

The implicit differencing method using equations (2.16) - (2.18) has no stability requirement, but requires the solution of nxm simultaneous equations (2.18). This can be accomplished by using Gauss Elimination or Pivotal techniques. For many nodes this is very time consuming.

2.5 Conclusion

It was originally hoped to use the Pace 231-R analogue computer for the work done in this thesis. The analogue computer could not be used due to an insufficiency in the number of multipliers and integrators available in the machine. This difficulty will be explained fully in Appendix B.

To circumvent the difficulties with the analogue computer, the s/360 continuous modeling programme (CSMP) was used on the IBM 360/67 computer. CSMP is a problem oriented programme designed to facilitate the digital simulation of continuous processes. The continuous process is discretized by the programme. A complete explanation of CSMP appears in Appendix A.

Since the digital computer was used all the variables had to be discretized, and so, space time discretization had to be used. The problem was looked at as though the spatial variable was discretized and the temporal variable was maintained in the continuous form. The CSM program was then used to discretize the temporal variable. Since both the spatial and the temporal variables were ultimately discretized, equation (2.19) had to be satisfied. Though we will call the discretization scheme used spatial discretization and will use the equations developed in Section (2.2), the space time discretization scheme is really what is obtained using CSMP, as can be seen from the preceeding argument.

In the next chapter the parameter sensitivity coefficients are obtained for the spatially discretized partial differential equations obtained in this chapter.

3. SENSITIVITY ANALYSIS

3.1 Introduction

This chapter considers two methods to obtain the parameter sensitivity coefficients, namely; Meissinger's method, and the method of structural sensitivity. These methods will be extended to obtain the parameter sensitivity coefficients of the discretized parabolic differential equations obtained in Section (2.2).

Though the method of structural sensitivity is useful for ordinary differential equations, it will become evident that there is little advantage to it over the other method when applied to distributed parameter systems.

3.2 Meissinger's Method

Meissinger's method for the generation of the parameter sensitivity coefficients can be described by the following example.

Consider the second order differential equation

$$\ddot{Y}(t) + a_1\dot{Y}(t) + a_0Y(t) = X(t) \tag{3.1}$$

where X(t) is the forcing function.

Meissinger's parameter influence coefficients are defined as

$$U_1 = \partial Y/\partial a_1 \quad \text{and} \quad U_0 = \partial Y/\partial a_0 \tag{3.2}$$

Differentiating equation (3.1) partially with respect to a_1 and a_0, we obtain

$$\ddot{U}_1 + a_1\dot{U}_1 + a_0U_1 = -\dot{Y} \tag{3.3}$$

$$\ddot{U}_0 + a_1\dot{U}_0 + a_0U_0 = -Y \tag{3.4}$$

To obtain the sensitivity coefficients, equations (3.3) and (3.4) must be solved. These equations are structurally the same as equation (3.1) but have different inputs.

Let us now apply Meissinger's method to the discretized differential equation obtained in the previous chapter.

3.3 Application of Meissinger's Method to the One Dimensional Parabolic Differential Equation

In Section (2.2) it was found that a one dimensional parabolic partial differential equation can be put in the following vector form

$$\dot{Q} = A\lambda Q + ABQ_B$$

where A is a matrix of the unknown parameters. Let α be an element of A. Differentiating equation (2.8) with respect to α we obtain

$$\begin{aligned} \frac{\partial \dot{Q}}{\partial \alpha} &= \frac{\partial}{\partial \alpha}[A\lambda Q + ABQ_B] \\ &= \frac{\partial A}{\partial \alpha}\lambda Q + A\frac{\partial \lambda}{\partial \alpha}Q + A\lambda\frac{\partial Q}{\partial \alpha} + \frac{\partial A}{\partial \alpha}BQ + A\frac{\partial B}{\partial \alpha}Q_B + AB\frac{\partial Q_B}{\partial \alpha} \end{aligned} \tag{3.5}$$

Let $U \triangleq \partial Q/\partial \alpha$ be called the sensitivity coefficient vector. Since λ, B, and Q_B are not functions of α. We obtain

$$\dot{U} = \frac{\partial A}{\partial \alpha}\lambda Q + A\lambda U \tag{3.6}$$

Applying Meissinger's method to the discretized version of equation (2.8), namely equations (2.5) - (2.7) we obtain

$$\dot{U}_{i,j}(t) = \frac{\alpha_i}{\Delta x^2}[U_{i+1,j} - 2U_{i,j} + U_{i-1,j}] + \frac{\delta_{i,j}}{\Delta x^2}[q_{i+1} - 2q_i + q_{i-1}] \tag{3.7}$$

where

$$U_{i,j} \triangleq \partial q_i/\partial \alpha_j \tag{3.8}$$

with boundary conditions

$$U_{i,j}(t)\Big|_{i=0} = 0. \qquad \text{and} \qquad U_{i,j}(t)\Big|_{i=n+1} = 0. \tag{3.9}$$

and initial conditions

$$U_{i,j}(0) = 0. \tag{3.10}$$

where $i=1, \ldots, n$ and $j=1,\ldots, n$

and

$$\delta ij=1 \quad \text{for} \quad i=j$$

$$\delta ij = 0 \quad \text{for} \quad i \neq j$$

If equation (2.5) consists of n first order equations, Meissinger's method requires the generation of n×n first order sensitivity equations in order to arrive at all the parametric sensitivity coefficients.

Let us now apply Meissinger's method to the two dimensional parabolic differential equation.

3.3.1 Application of Meissinger's Method to the Two Dimensional Parabolic Differential Equation

Let us consider the two dimensional parabolic differential equation

$$\frac{\partial q}{\partial t}(x,y,t) = \alpha(x)\frac{\partial^2 q}{\partial x^2}(x,y,t) + \beta(y)\frac{\partial^2 q}{\partial y^2}(x,y,t) \tag{3.11}$$

where $x\varepsilon(0,x_f)$, $y\varepsilon(0,y_f)$, $t\varepsilon(0,t_f)$

with boundary conditions

$$q(x,y,t)\big|_{x=0} = L_1(y,t) \quad \text{and} \quad q(x,y,t)\big|_{y=0} = L_2(x,t) \tag{3.12}$$

$$q(x,y,t)\big|_{x=x_f} = M_1(y,t) \quad \text{and} \quad q(x,y,t)\big|_{y=y_f} = M_2(x,t)$$

and initial conditions

$$q(x,y,t)\big|_{t=0} = K(x,y) \tag{3.13}$$

We can discretize equation (3.9) with respect to the two spatial variables x and y. Using the discretization developed in Section (2.2) we obtain

$$\dot{q}_{i,j} = \frac{\alpha_i}{\Delta x^2}[q_{i+1} - 2q_i + q_{i-1}]_j + \frac{\beta j}{\Delta v^2}[q_{j+1} - 2q_j + q_{j-1}]_i \tag{3.14}$$

where i=1, ..., n and j=1, ..., m. Let us pick m=n just for simplicity.

Applying Meissinger's method to equation (3.10) we obtain

$$\dot{U}_{i,j,k} = \frac{\alpha_i}{\Delta x^2}[U_{i+1} - 2U_i + U_{i-1}]_{j,k} + \frac{\beta j}{\Delta y^2}[U_{j+1} - 2U_j + U_{j-1}]_{i,k} + \frac{\delta i,j,k}{\Delta x^2}[q_{i+1} - 2q_i + q_{i-1}]_j \tag{3.15}$$

and

$$\dot{U}_{i,j,m} = \frac{\alpha_i}{\Delta x^2}[U_{i+1} - 2U_i + U_{i-1}]_{j,m} + \frac{\beta j}{\Delta y^2}[U_{j+1} - 2U_j + U_{j-1}]_{i,m} + \frac{\delta i,j,m}{\Delta y^2}[q_{j+1} - 2q_j + q_{j-1}]_i \tag{3.16}$$

where $U_{i,j,k} = \partial q_{ij}/\partial\alpha_k$ and $U_{i,k,m} = \partial q_{ij}/\partial\beta_m$ are defined as the parameter sensitivity coefficients, and where

$$\delta_{ijk} = 1. \quad \text{when } i=j=k$$
$$= 0. \quad \text{else}$$

and

$$\delta_{ijm} = 1. \quad \text{when } i=j=m$$
$$= 0. \quad \text{else}$$

If equation (3.14) consists of $(n \times n)$ first order equations, we can see that Meissinger's method requires us to generate $(2 \times n \times n^2)$ first order sensitivity equations to arrive at all the parametric sensitivity coefficients.

The next section deals with the application of structural sensitivity to the preceding problem.

3.4 Structural Sensitivity

The purpose of this section is to extend structural sensitivity so that it will be useful in determining the sensitivity functions of a discretized partial differential equation.

Kokotovic has shown that for a class of systems with one direct path all the sensitivity functions may be obtained simultaneously with the use of one additional system model[8]. This is superior to Meissinger's method because Meissinger's method requires the generation of a system model for each parameter, as seen in Section (3.2).

The methods developed by Kokotovic[8-10], Vuscovic and Ciric[31] deal only with systems where the m^{th} branch transmittance in a signal flow graph is $W_m(s, \alpha_i)$ and where α_i, the parameter of interest only appears in that branch. Structural sensitivity will be extended to those systems where the parameters of interest appear in more than one branch transmittance i.e. α_i might appear in $W_k(s, \alpha_i)$ and $W_l(s, \alpha_i, \alpha_j)$.

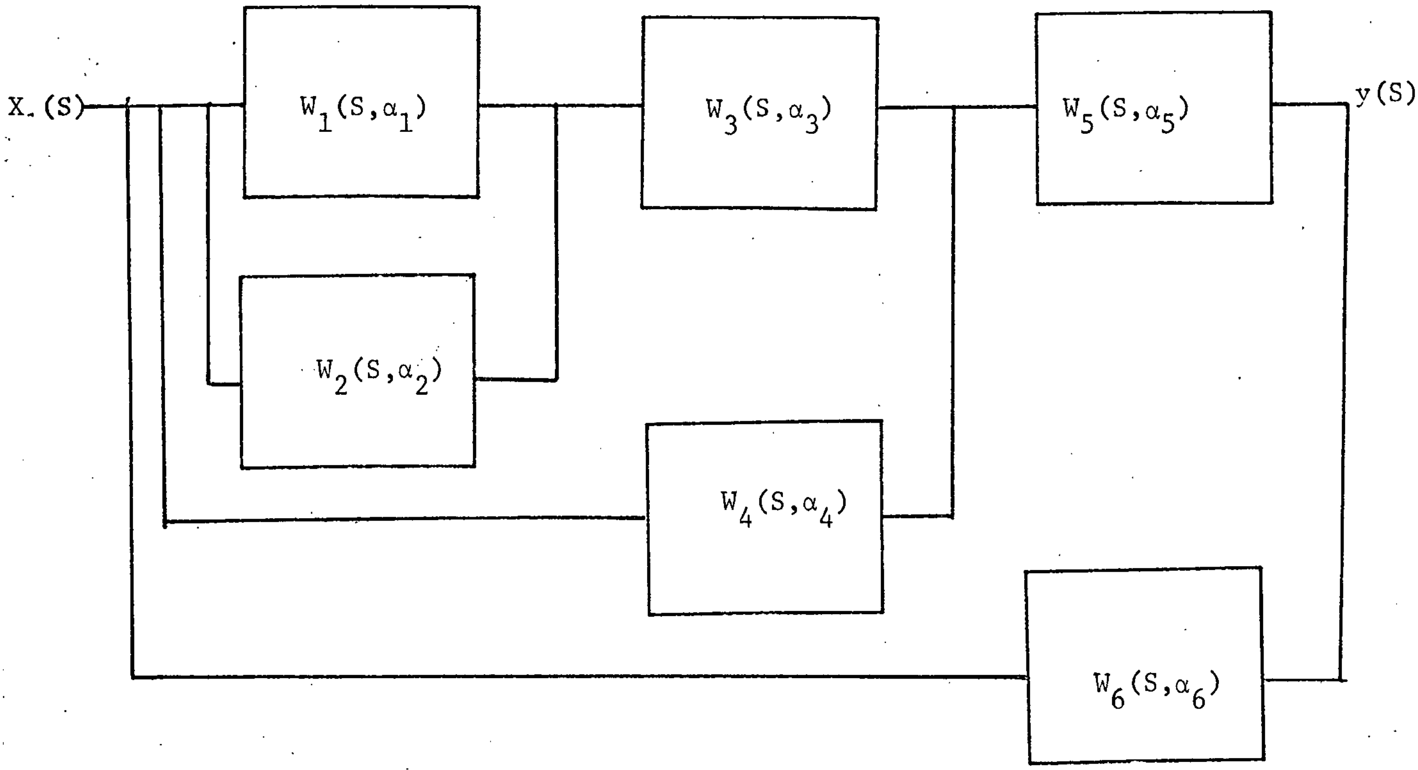
The class of systems considered are those in which the outputs of the feedback paths are brought together and subtracted from the input and there is one feedforward path. This class of systems is called the Kokotovic class. A three loop Kokotovic system is shown in Figure 3.1.

A spatially discretized parabolic differential equation is not of the Kokotovic class, as can be seen in Figure 3.2. By applying signal flow graph theory to the discretized system, it can become of the Kokotovic class provided that one of the boundary conditions is set to zero (one feedforward path). The parameters now appear in a more complicated form in the transmittance in the feedback paths, as can be seen in Figure 3.3.

The sensitivity functions considered will be that of equation (3.8) i.e.

$$U_{i,j} \triangleq \partial q_j / \partial \alpha_i$$

With the partial differential equation in the Kokotovic class it is evident that the system equation becomes



X. (S)
$W_1(S,\alpha_1)$
$W_2(S,\alpha_2)$
$W_3(S,\alpha_3)$
$W_4(S,\alpha_4)$
$W_5(S,\alpha_5)$
$W_6(S,\alpha_6)$
y(S)

Figure 3.1

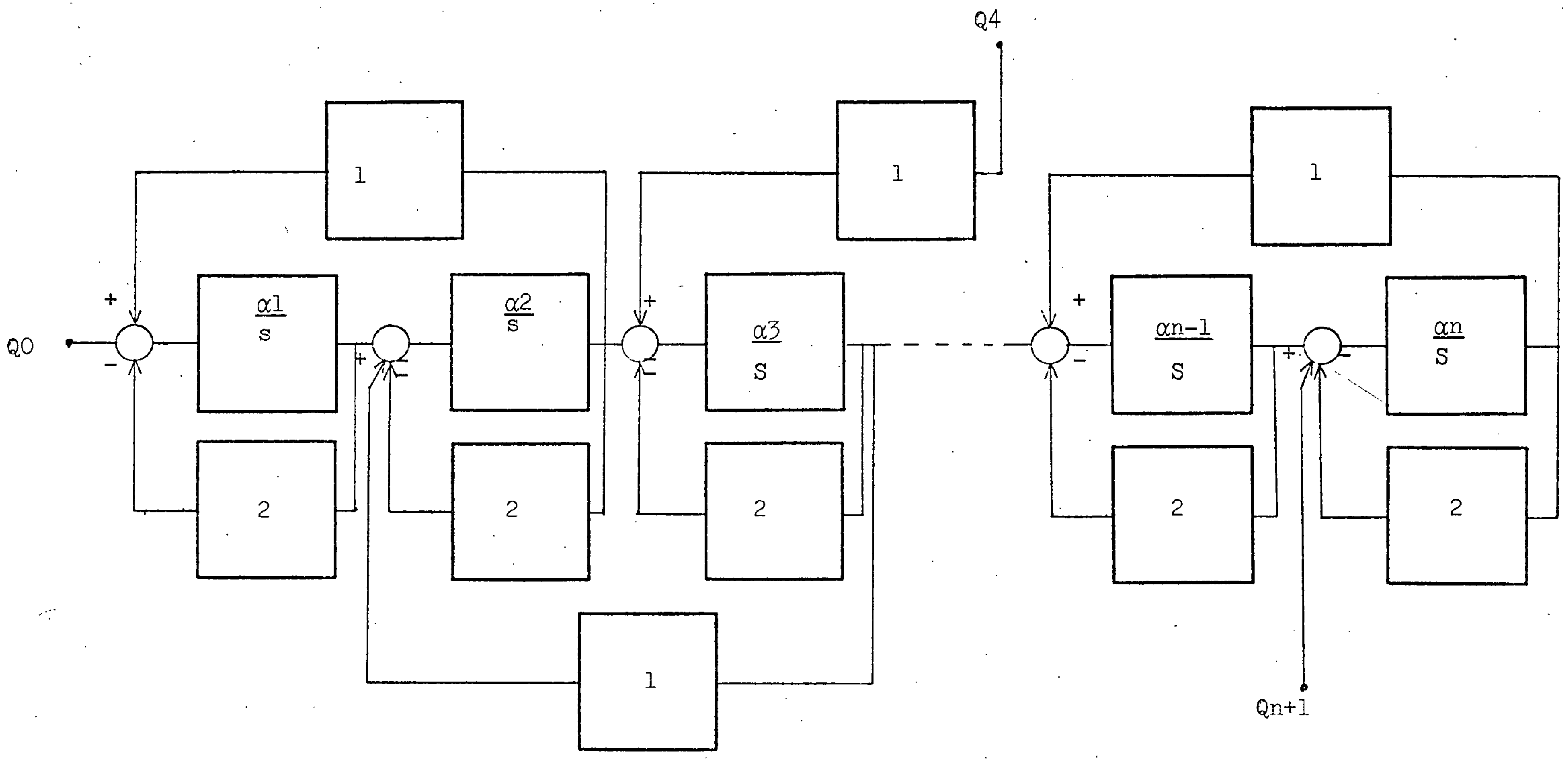
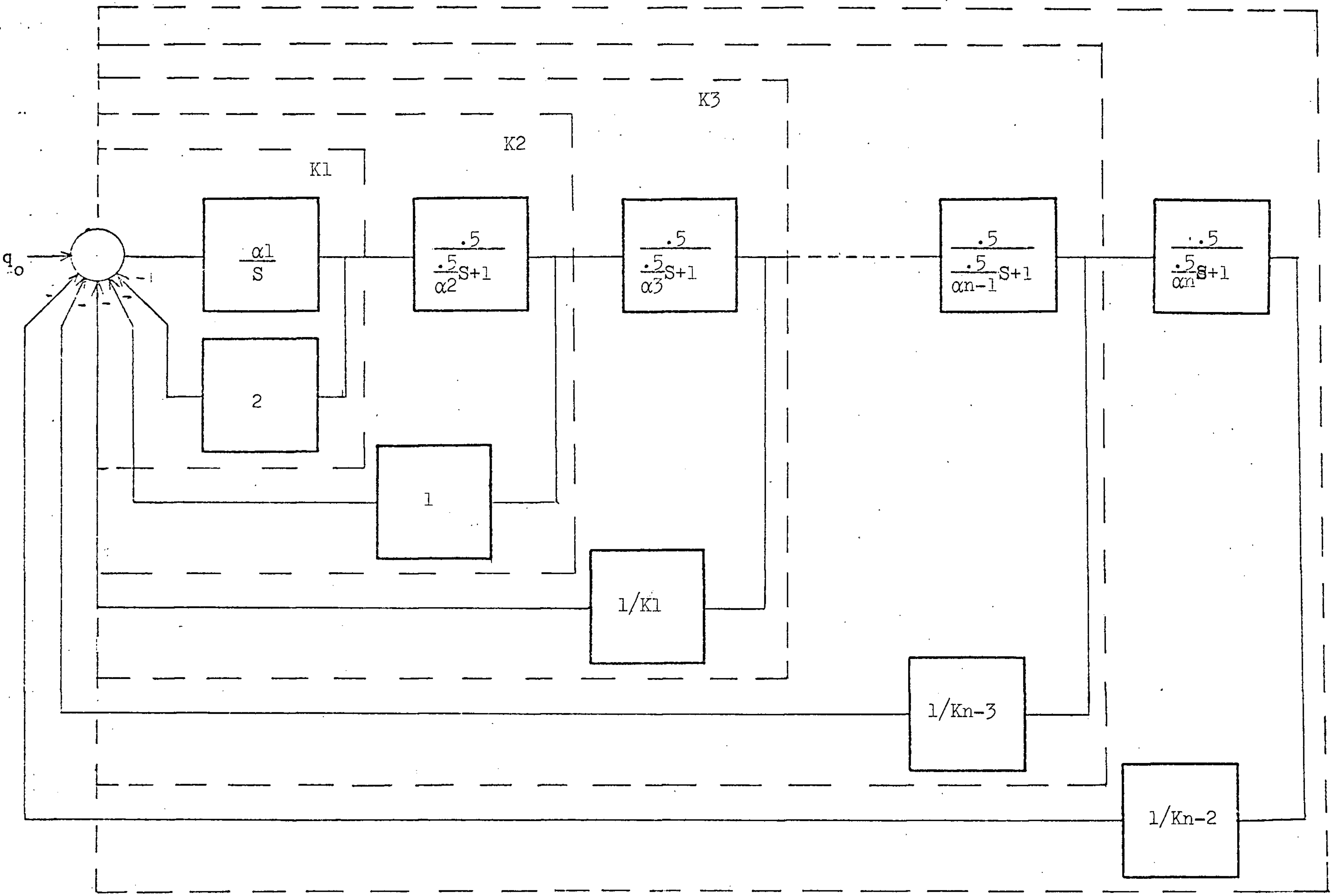


FIGURE 3.2



K3
K2
K1
q_o
$\frac{\alpha 1}{S}$
$\frac{.5}{\frac{.5}{\alpha 2}S+1}$
$\frac{.5}{\frac{.5}{\alpha 3}S+1}$
$\frac{.5}{\frac{.5}{\alpha n-1}S+1}$
$\frac{.5}{\frac{.5}{\alpha n}S+1}$
2
1
1/K1
1/Kn-3
1/Kn-2

FIGURE 3.3

$$q_j(s) = W_j(s,\alpha_1,\ldots,\alpha_n)q_o(s) \tag{3.17}$$

where $W_j(s,\alpha_1, \ldots,\alpha_n)$ is the overall transfer function needed to obtain the output $q_j(s)$, and where $q_o(s)$ is the input (one of the boundary conditions). Subsituting equation (3.17) into equation (3.8) we obtain

$$U_{ij}(s) = \frac{}{W_j(s\ \alpha_1\ldots\alpha_n)} \cdot \frac{\partial W_j(s,\alpha_1 \cdot \alpha_n)}{\partial \alpha_i}\, q_j(s) \tag{3.18}$$

If the parameter α_i appears in more than one loop transfer function, we must apply the chain rule to equation (3.18) to obtain

$$U_{i,j} = \frac{q_j}{W_j(s,\alpha_1\ldots\alpha_n)} \sum_{k=1}^{n} \frac{\partial W_j(s,\alpha_1\ldots\alpha_n)}{\partial W_k(s,\alpha_i)} \cdot \frac{\partial W_k(s,\alpha_i)}{\partial \alpha_i} \tag{3.19}$$

$$= q_j \sum_{k=1}^{n} B_{j,k} \cdot C_{k,i} = \sum_{k=1}^{n} V_{i,j,k}$$

where

$$B_{j,k} \triangleq \frac{W_k}{W_j} \frac{\partial W_j}{\partial W_k} \tag{3.20}$$

and

$$C_{k,i} \triangleq \frac{1}{W_k} \frac{\partial W_k}{\partial \alpha_i} \tag{3.21}$$

The functions $B_{j,k}$ depend on the system's structure, and the functions $C_{k,i}$ depend on the structure of the links W_k.

If

$$W_k(s) = \frac{\alpha i}{s+2\alpha_i} \quad \text{then} \quad C_{k,i} = \frac{s}{\alpha_i(s+2\alpha_i)}$$

Equation (3.20) defines the transfer functions from the system's input q_j to the point $S_{k,j}$. These points are called the sensitivity points.

To determine the sensitivity functions $U_{i,j}$, it suffices to send the signals from the sensitivity points $S_{k,j}$ to the blocks $C_{k,i}$ and sum the results, as in equation (3.19). All the sensitivity functions are determined

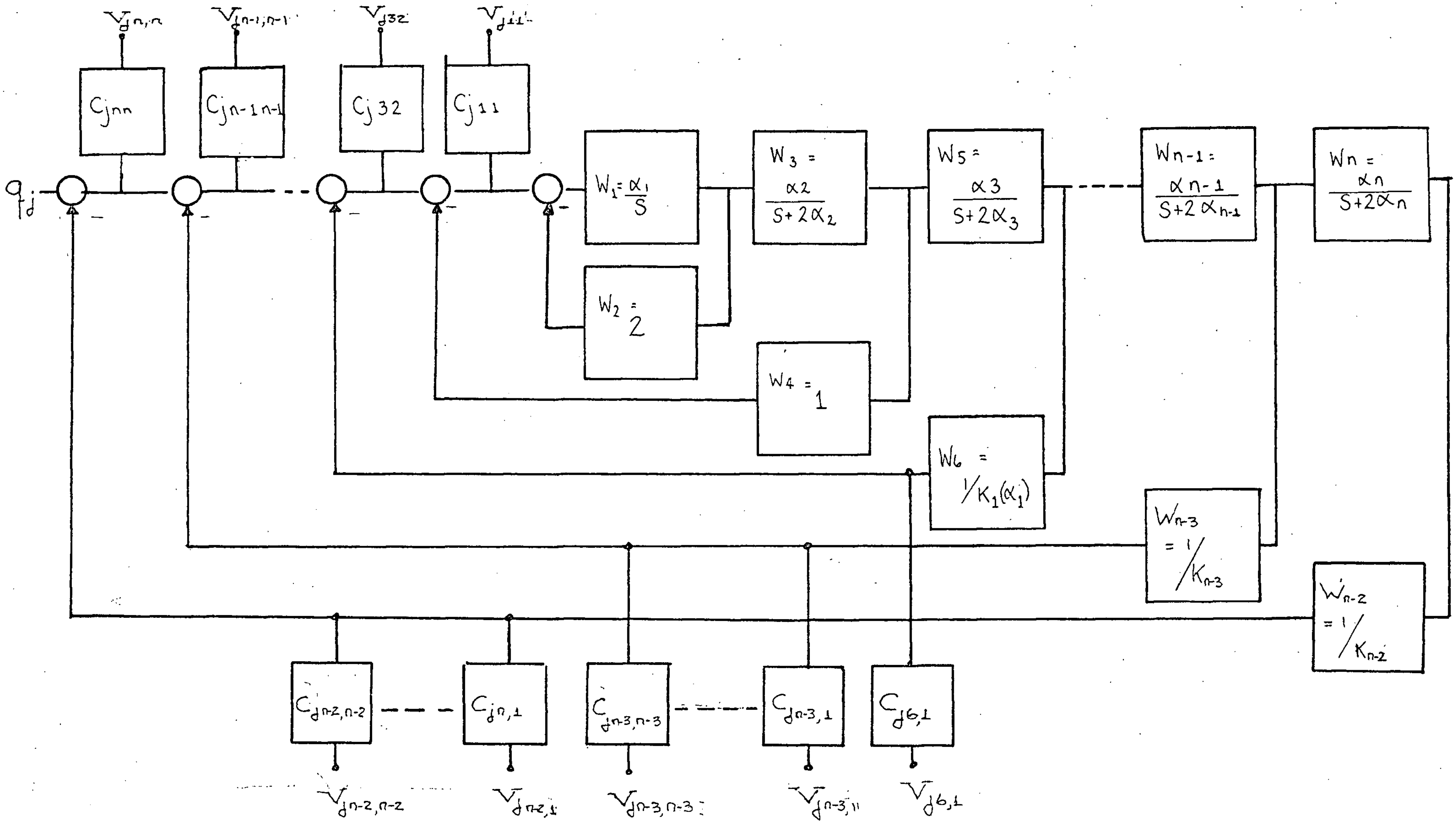

$V_{jn,n}$
$V_{jn-1,n-1}$
V_{j32}
V_{j11}
C_{jnn}
$C_{jn-1n-1}$
C_{j32}
C_{j11}
q_j
$W_1 = \frac{\alpha_1}{S}$
$W_2 = 2$
$W_3 = \frac{\alpha 2}{S+2\alpha_2}$
$W_4 = 1$
$W_5 = \frac{\alpha 3}{S+2\alpha_3}$
$W_6 = 1/K_1(\alpha_1)$
$W_{n-1} = \frac{\alpha n-1}{S+2\alpha_{n-1}}$
$W_n = \frac{\alpha n}{S+2\alpha n}$
$W_{n-3} = 1/K_{n-3}$
$W_{n-2} = 1/K_{n-2}$
$C_{jn-2,n-2}$
$C_{jn,1}$
$C_{jn-3,n-3}$
$C_{jn-3,1}$
$C_{j6,1}$
$V_{jn-2,n-2}$
$V_{jn-2,1}$
$V_{jn-3,n-3}$
$V_{jn-3,1}$
$V_{j6,1}$

FIGURE 3.4

by varying the input q_j of the sensitivity model. This can be seen in Figure 3.4. Since the system under study does not have a single output (as is the case with ordinary differential equations), but has n outputs, the number of equations required to generate the sensitivity coefficients is not reduced by using structural sensitivity.

3.5 Conclusion

Structural sensitivity can not be applied to partial differential equations in general. The systems considered using this method must be of the Kokotovic class. This implies putting the systems into the Kokotovic form, and hence limiting the systems studied to those in which only one boundary condition is considered.

Since the system equation has n outputs, structural sensitivity requires us to solve $(n \times n)$ equations i.e. the same number of equations as in Meissinger's method and hence no saving results.

The only advantage to the application of structural sensitivity is that once the sensitivity model is obtained we need only vary the input to the model to obtain all the sensitivity coefficients.

Meissinger's method has none of the above constraints.

The next chapter will deal with the identification of parameters using the method of Bingulac and Kokotovic and the sensitivity methods developed in this chapter.

4. PARAMETER IDENTIFICATION IN DISTRIBUTED PARAMETER SYSTEMS

4.1 Introduction

In this chapter the sensitivity methods developed in the previous chapter are used in conjunction with the method of Bingulac and Kokotovic in order to identify parameters in the first and second order parabolic differential equations. The method of Bingulac and Kokotovic is presented in the next section.

4.2 The Method of Bingulac and Kokotovic

The method of Bingulac and Kokotovic is applicable to parameter identification problems using a mode controlled analogue computer (or in our case CSMP).

The parameter vector $\alpha = [\alpha_1 \; . \; . \; \alpha_n]^T$ is adjusted by means of steepest descent in order to minimize a performance index $J(\alpha)$ of the form

$$J(\alpha) = \iint_{so}^{T} F(e)\,dt\; d\Omega \tag{4.1}$$

The sensitivity methods developed in the previous chapter are used in obtaining

$$\frac{\partial J}{\partial \alpha} = \iint_{so}^{T} \frac{\partial F}{\partial e}\, U\, dt \quad d\Omega \tag{4.2}$$

It is assumed that the parameter vector α is held constant in deriving equation (4.2). For this reason the computer is operated in two modes. In the compute mode, the parameter vector is held constant while the constitutents of equation (4.2) are calculated. In the reset mode the parameter vector is adjusted according to the steepest descent law

$$\dot{\alpha}_j = -K \frac{\partial J}{\partial \alpha_j} \tag{4.3}$$

This method is summarized in Figures (4.1) and (4.2). The increments in α_j are required to be kept small because the direction of steepest descent is steepest only in the vicinity of α_j. The increments in α_j can be kept small by an appropriate choice of K in the above equation. A compromise must be struck between the stability of the method and the speed of convergence. The larger the value of K is, the faster is the rate of convergence, and the greater is the risk of instability.

Let us consider now, the identification of the one dimensional parabolic differential equation.

4.3 Identification of the One Dimensional Parabolic Differential Equation

We would like to identify the unknown parameters of the following distributed parameter system,

$$\frac{\partial s}{\partial t}(x,t) = \beta(x) \frac{\partial^2 s}{\partial x^2}(x,t) \tag{4.4}$$

with boundary and initial conditions previously used in equations (2.5) - (2.6) and where $\beta(x)$ is the unknown spatially dependent parameter.

Let the model be

$$\frac{\partial q}{\partial t}(x,t) = \alpha(x) \frac{\partial^2 q}{\partial x^2}(x,t) \tag{4.5}$$

with the same initial and boundary conditions as above and where $\alpha(x)$ is a chosen parameter.

Let the performance function be

$$J = \int_0^{x_f} \int_0^{t_f} (q(x,t) - s(x,t))^2 dxdt \tag{4.6}$$

which is quadratic and covers the entire space time domain.

Discretizing equations (4.4) - (4.6) spatially as in Section (2.2) we obtain

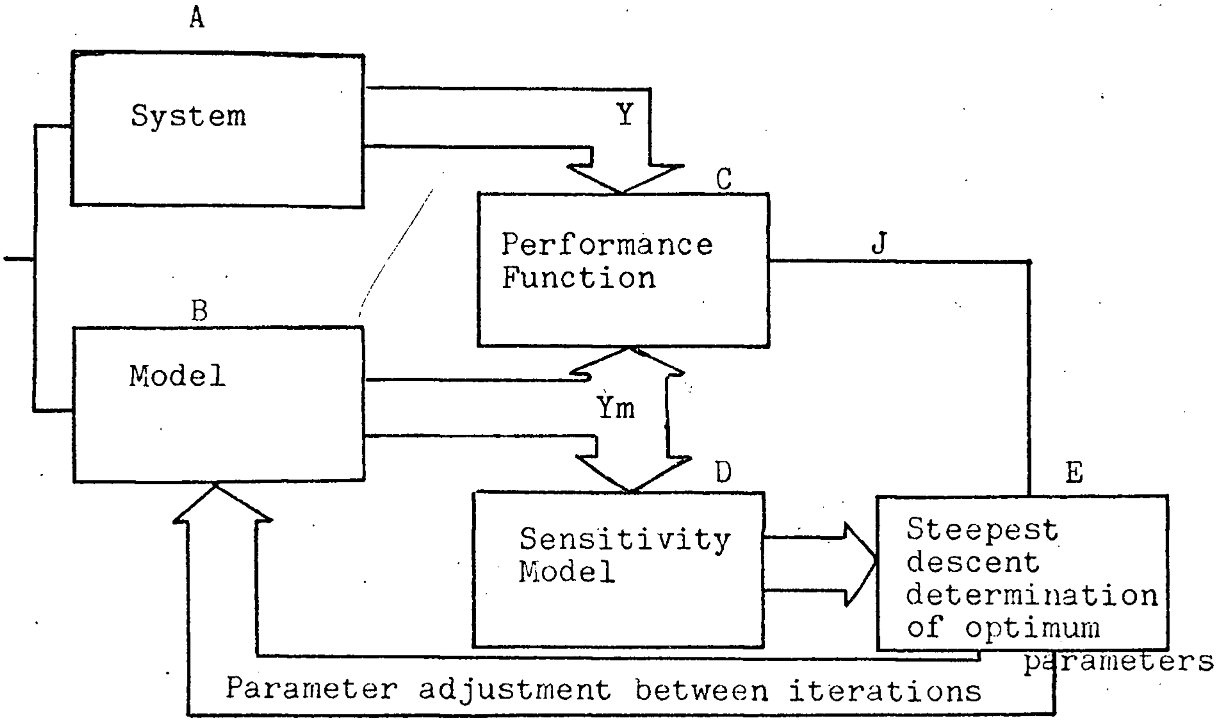


Figure 4.1

MODE

BLOCK	COMPUTE	RESET
A,B,C,D	Compute $\frac{\partial J}{\partial \alpha_i}$ according to (4.2)	Return to initial conditions for the next iteration
E	Hold α_i constant	Adjust α_i for the next iteration according to (4.3)

Figure 4.2

$$\frac{ds_i}{dt} = \beta_i(s_{i+1}(t) - 2s_i(t) + S_{i-1}(t))/\Delta x^2 \tag{4.7}$$

$$\frac{dq_i}{dt} = \alpha_i(q_{i+1}(t) - 2q_i(t) + q_{i-1}(t))/\Delta x^2 \tag{4.8}$$

$$J = \Delta x \int_o^{t_f} \sum_{i=1}^{n} (q_i - s_i)^2 dt \tag{4.9}$$

Differentiating equation (4.9) with respect to α and substituting into equation (4.3) yields

$$\dot{\alpha}_j = -K \int_o^{t_f} \sum_{i=1}^{n} (q_i - s_i)\, U_{ij} dt \tag{4.10}$$

where $j=1, \ldots, n$

The sensitivity coefficients U_{ij} are obtainable by using either of the schemes developed in Chapter 3.

Before dealing with a few examples, a word is in order about the choice of boundary and initial conditions.

4.3.1 Boundary and Initial Conditions

The boundary and the initial conditions are of the form

$$q(x,t)\big|_{x=0} = L(t) \quad ; \quad q(x,t)\big|_{x=x_f} = M(t) \tag{4.11}$$

$$q(x,t)\big|_{t=0} = K(x) \tag{4.12}$$

The conditions are assumed not to be functions of the parameters under study. It is also assumed that they are given for both the system and the model.

Discretizing equations (4.11) - (4.12) spatially we obtain

$$q_0(t) = L(t) \quad ; \quad q_{n+1}(t) = M(t) \tag{4.13}$$

$$q_i(0) = K_i \quad \text{where } i = i, \ldots, n \tag{4.14}$$

The form of the boundary conditions presents no problem provided that each value of $q_0(t)$ and $q_{n+1}(t)$ is introduced into equations (4.7) and (4.8) at the correct point in time i.e. as the time sweep progresses. This is easily obtainable using CSMP.

Since it is assumed that the parameters under study are not functions of the boundary and initial conditions, let us choose the easiest conditions to deal with, namely

$$q(x,t)\big|_{x=0.} = 0. \quad ; \qquad q(x,t)\big|_{x=x_f} = 0 \tag{4.15}$$

$$q(x,t)\big|_{t=0} = K(x) \tag{4.16}$$

We are now in a position to deal with some examples.

4.3.2 Example #1

In this example Meissinger's method is used to obtain the sensitivity coefficients. Here n=3 is chosen.

The system equation is

$$\frac{dS_i}{dt} = \frac{\beta i}{\Delta x^2}\left[S_{i+1} - 2S_i + S_{i-1}\right] \qquad i=1, \ldots, n \tag{4.17}$$

with boundary conditions $S_o = S_4 = 0$ and initial conditions $S_i(0) = 1$.

The model equation is

$$\frac{dq_i}{dt} = \frac{\alpha i}{\Delta x^2}\left[q_{i+1} - 2q_i + q_{i-1}\right] \tag{4.18}$$

with boundary conditions $q_o = q_4 = 0$ and initial conditions $q_i(0) = 1$.

Ther performance function is

$$J = \int_o^{t_f} \sum_{i=1}^{n} (q_i - S_i)^2 dt \tag{4.19}$$

The Meissinger sensitivity model is

$$U_{i,j} = \frac{\alpha i}{\Delta x^2}\left[U_{i+1} - 2U_i + U_{i-1}\right]_j + \frac{\delta_{ij}}{\Delta x^2}\left[q_{i+1} - q_i + q_{i-1}\right] \tag{4.20}$$

where

$$i = 1, \ldots, n \quad ; \quad j=1, \ldots, n$$

Where the boundary and initial conditions are

$$U_{ij}(0) = 0; \quad U_{oj} = U_{4j} = 0 \tag{4.21}$$

The update algorithm is

$$\Delta\alpha_j = -K \int_o^T \sum_{i=1}^{n} (q_i - S_i)U_{ij}\, dt \tag{4.22}$$

The gain constant in equation (4.23) is set at K=.02. Since n=3. is chosen and the increment in x is set at .2, the length of rod considered is .8 units. Figure 4.3 shows the steepest descent of the controller parameters in the (α_1 α_2) plane, plotted against contours of equal J. To obtain the contours of equal J, α_3 is set at a constant.

Figure 4.4 shows the steepest descent of the controller parameter parameters in the (α_2, α_3) plane, plotted against contours of equal J, where the contours of equal J are obtained by setting α_1 to a constant.

The iterative procedure was initiated from three locations to check for local minima. As can be seen in Figure (4.3) and (4.4), the initial guesses of (.2,.2,.2), (.05,.05,.05) and (.2,.05,.2) converged to the optimum values of (.1,.15,.18).

An unfortunate characteristic of the steepest descent method is its slow convergence in the region of the optimum[28]. If J has a narrow ridge, i.e. if the J contours can be approximated by ellipses with high eccentricity, then the convergence rate is even slower[30]. This can be overcome somewhat by increasing the gain constant K as the optimum is approached.

One dimensional parabolic case
n=3.

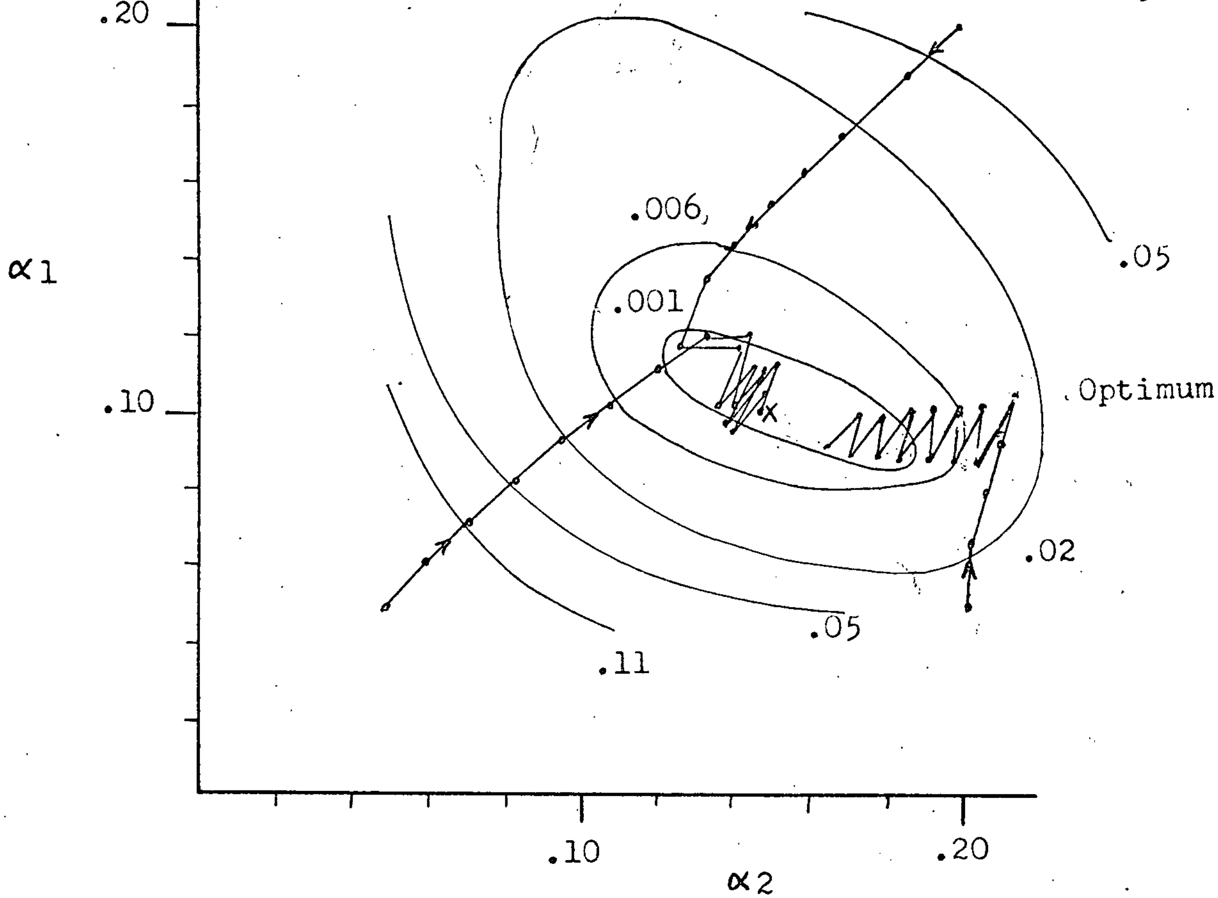


Figure 4.3

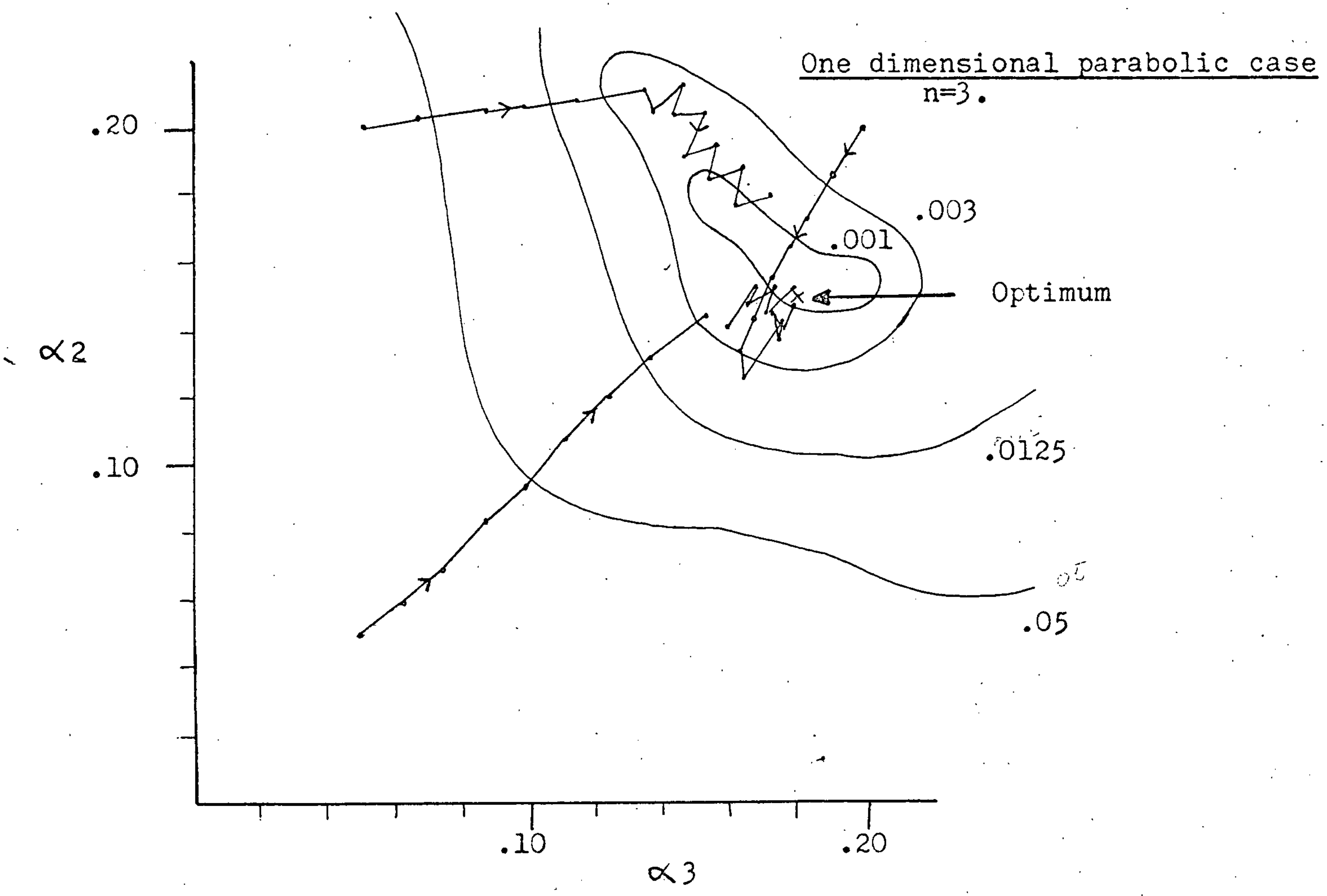

One dimensional parabolic case
n=3.
.20
.10
α_2
.003
.001
Optimum
.0125
.05
.10
.20
α_3

Figure 4.4

4.3.3 Example #2 - Structural Sensitivity

In this example the method of structural sensitivity is applied to the preceeding problem with the same choice of K, and initial guesses of α.

For n=3, we require six W_k links, as seen in Figure 3.4, where

$$W_1(S,\alpha_1) = \alpha_1/s$$
$$W_2(S) = 2$$
$$W_3(S,\alpha_2) = \alpha_2/(S+2\alpha_2)$$
$$W_4(S) = 1$$
$$W_5(S,\alpha_3) = \alpha_3/S+2\alpha_3$$
$$W_6(S,\alpha_1) = \frac{S+2\alpha_1}{\alpha_1}$$

Using equation (3.21) we obtain

$$C_{11} = 1/\alpha_1 \qquad C_{12} = C_{13} = \ldots C_{15} = 0$$
$$C_{21} = \ldots = C_{26} = 0$$
$$C_{32} = S/\alpha_2(S+2\alpha_2) \qquad C_{31} = C_{33} \ldots = C_{36} = 0$$
$$C_{41} = \ldots C_{46} = 0$$
$$C_{53} = S/\alpha_3(S+2\alpha_3) \qquad C_{51} = C_{52} = C_{54} = \ldots C_{56} = 0$$
$$C_{61} = -S/\alpha_1(S+2\alpha_1) \qquad C_{62} = \ldots = C_{66} = 0$$

From equation (3.19) we obtain

$$U_{ij} = q_j \sum_{k=1}^{6} B_{j,k} \; C_{k,i} = \sum_{k=1}^{6} V_{j,k,i} \qquad \text{for } i=1,..,3 \quad j=1,..,3$$

The nine sensitivity equations become

$$U_{1,j} = q_j[B_{j,1} \, C_{11} + B_{j,6} \; C_{j,6}] = S_{j,1} \cdot C_{11} + S_{j,6} \cdot C_{j,6}$$

$$U_{2,j} = q_j[B_{j,3} \cdot C_{3,2}] = S_{j,3} \cdot C_{3,2}$$

$$U_{3,j} = q_j[B_{j,5} \cdot C_{5,3}] = S_{j,5} \cdot C_{5,3}$$

The identification scheme using structural sensitivity produces precisely the same results and requires the same computing time as in Example #1 i.e. Figure (4.3) and (4.4).

4.3.4 Example #3

In this example, the same equations were used as in example #1. Here $\beta(x) = x^2 + ax + b$ where $a = -.25$, and $b = .11$. The discretized version of β becomes (.1, .17, .32).

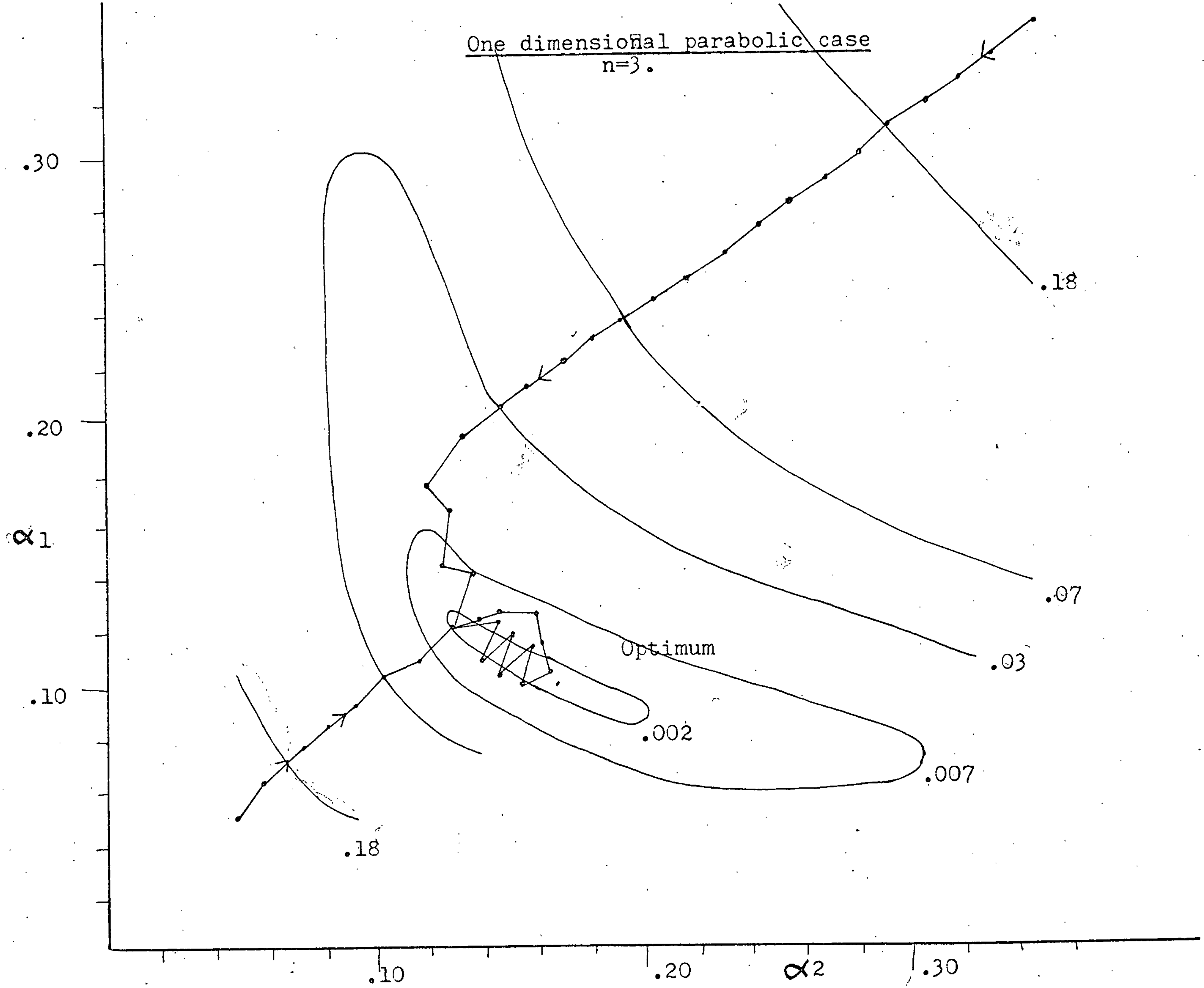
Figure 4.5 and Figure 4.6 show the steepest descent of the controller parameters in the (α_1, α_2) and (α_2, α_3) plane respectively.

The iterative procedure was initiated from two locations to check for local minima. As can be seen in the Figures, the initial guesses of (.05, .05, .05), (.35, .35, .35) converged to the optimum value of (.1, .17, .32). In Figure 4.6 a great deal of oscillation occurs due to the ridge at the minimum.

4.3.5 Example #4

In this example the same equations were used as in Example #1. Here n = 5. was chosen. With n = 5. and the increment in x set at .2, the length of the rod considered is 1.2 units. $\beta(x)$ was set as a linear function of x i.e. $\beta(x) = ax + b$ where $a = -.1$ and $b = .22$. The discretized parameters β become (.2, .18, .16, .14, .12). The α values converged to these values from the initial guess of (.1, .1, .1, .1, .1) within twenty iterations. This can be seen in Figures (4.7) - (4.9).

The greater the number of discretizations; the more equations need be solved i.e. for n = 5; 25 sensitivity equations are required to be solved. The beauty of using CSMP is that there is no constraint on the number



One dimensional parabolic case
n=3.
.18
.07
.03
.007
.002
Optimum
.18
.30
.20
.10
α_1
.10
.20
.30
α_2

Figure 4.5

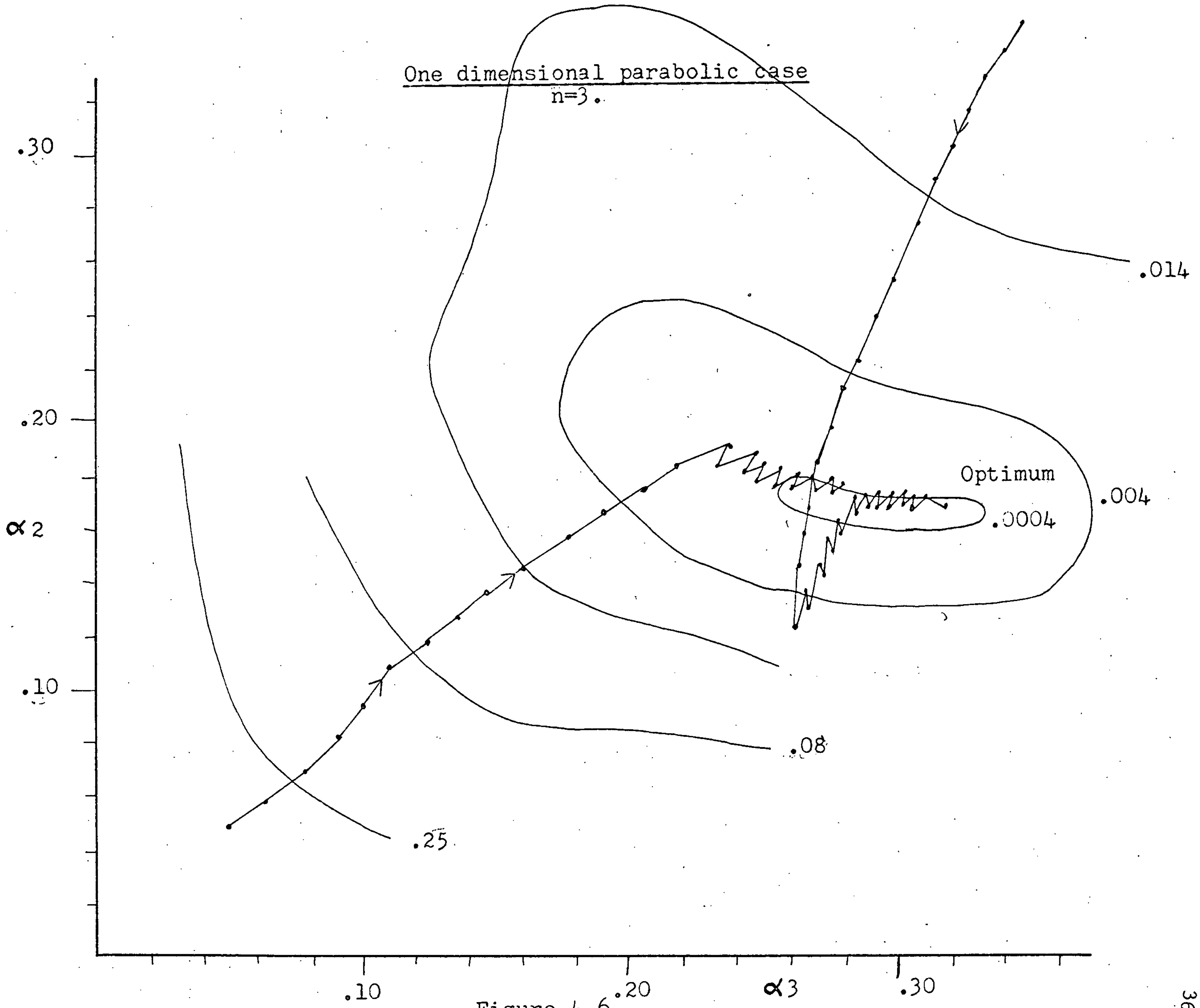

One dimensional parabolic case
n=3.
.30
.20
.10
α_2
.10
.20
α_3
.30
.014
.004
.0004
Optimum
.08
.25

Figure 4.6

ONE DIMENSIONAL PARABOLIC CASE
$n=5; \ \alpha(x)=ax+b$

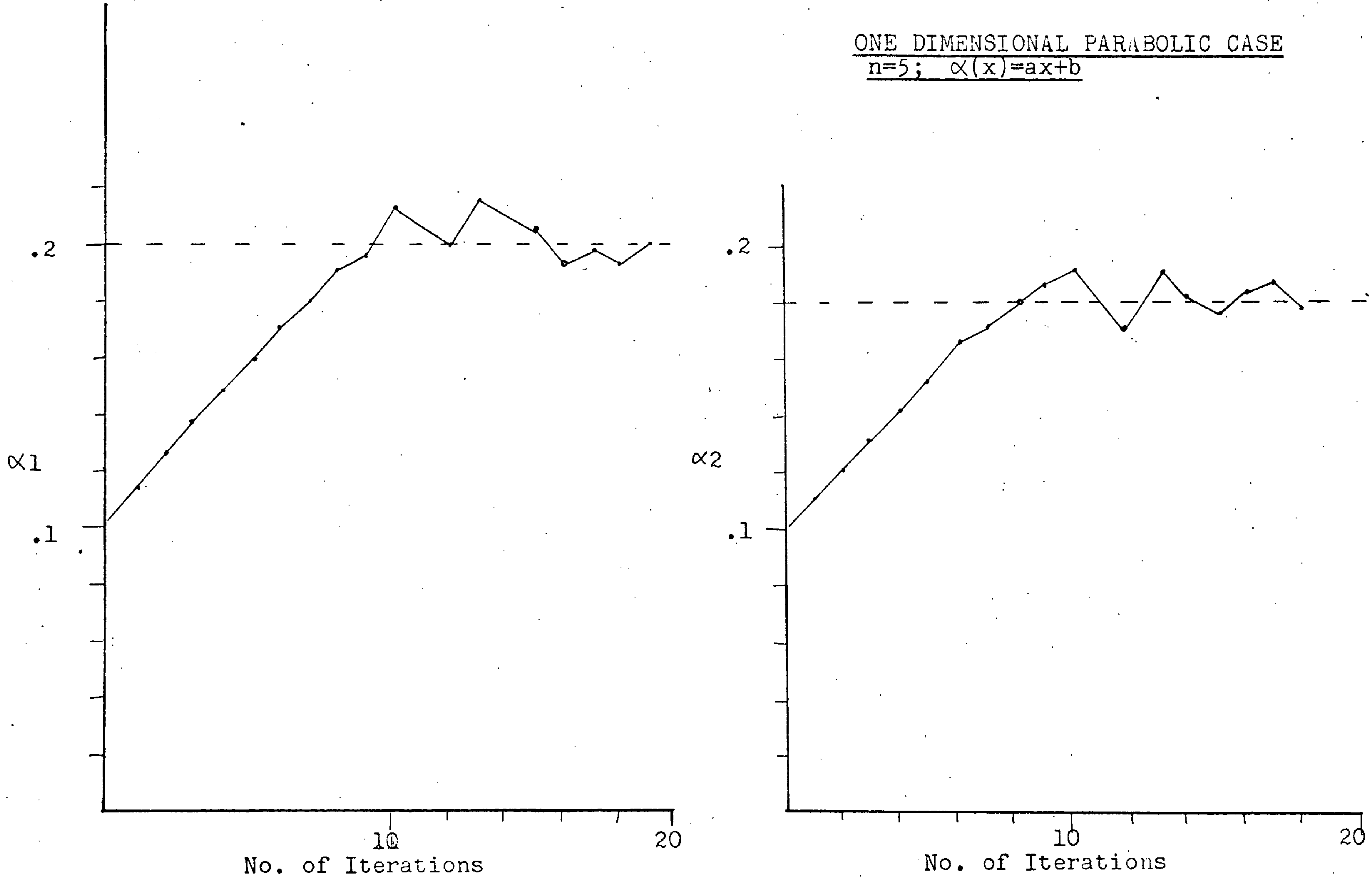


Figure 4.7

ONE DIMENSIONAL PARABOLIC CASE
$n=5; \quad \alpha(x)=ax+b$

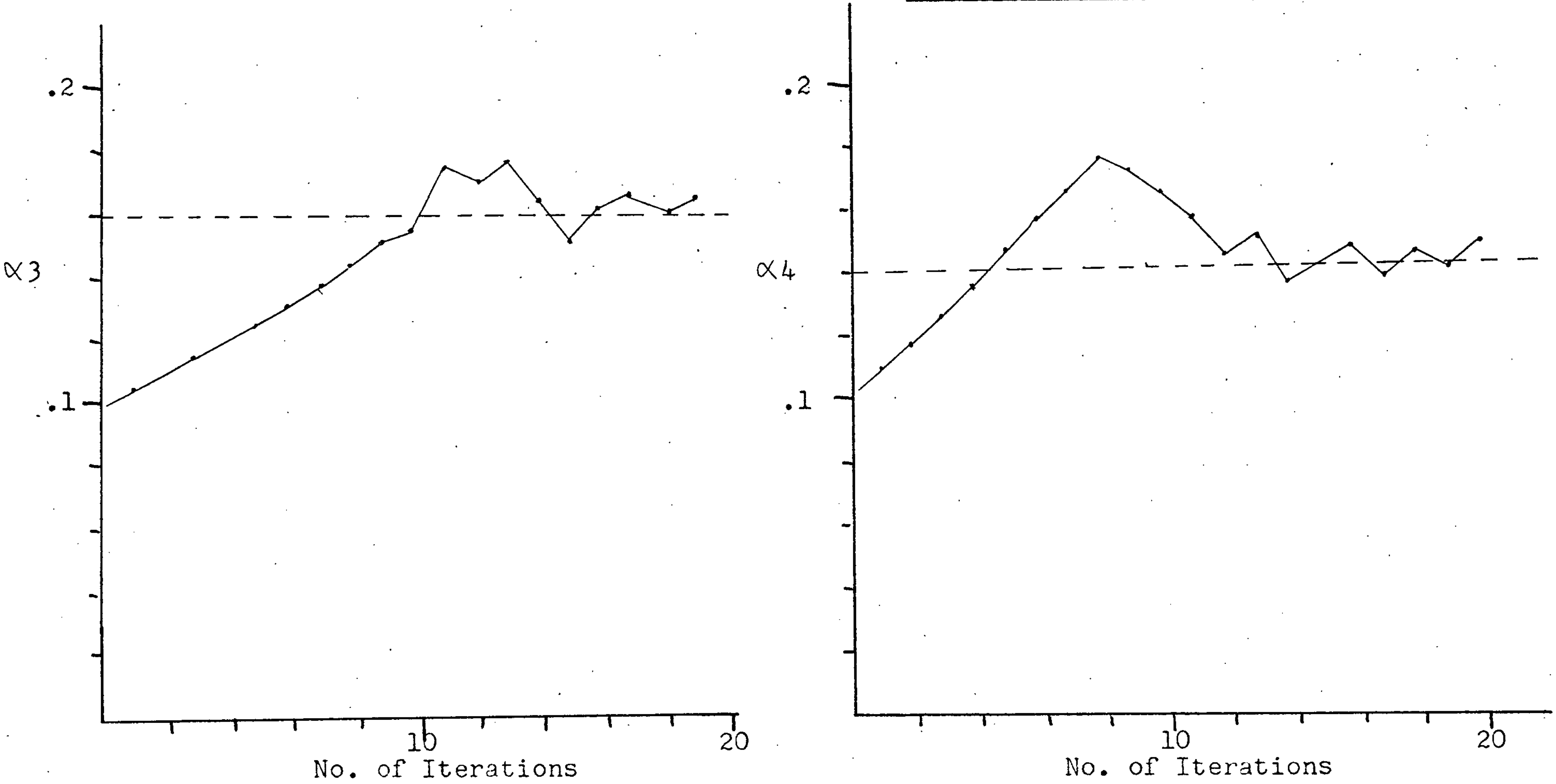


Figure 4.8

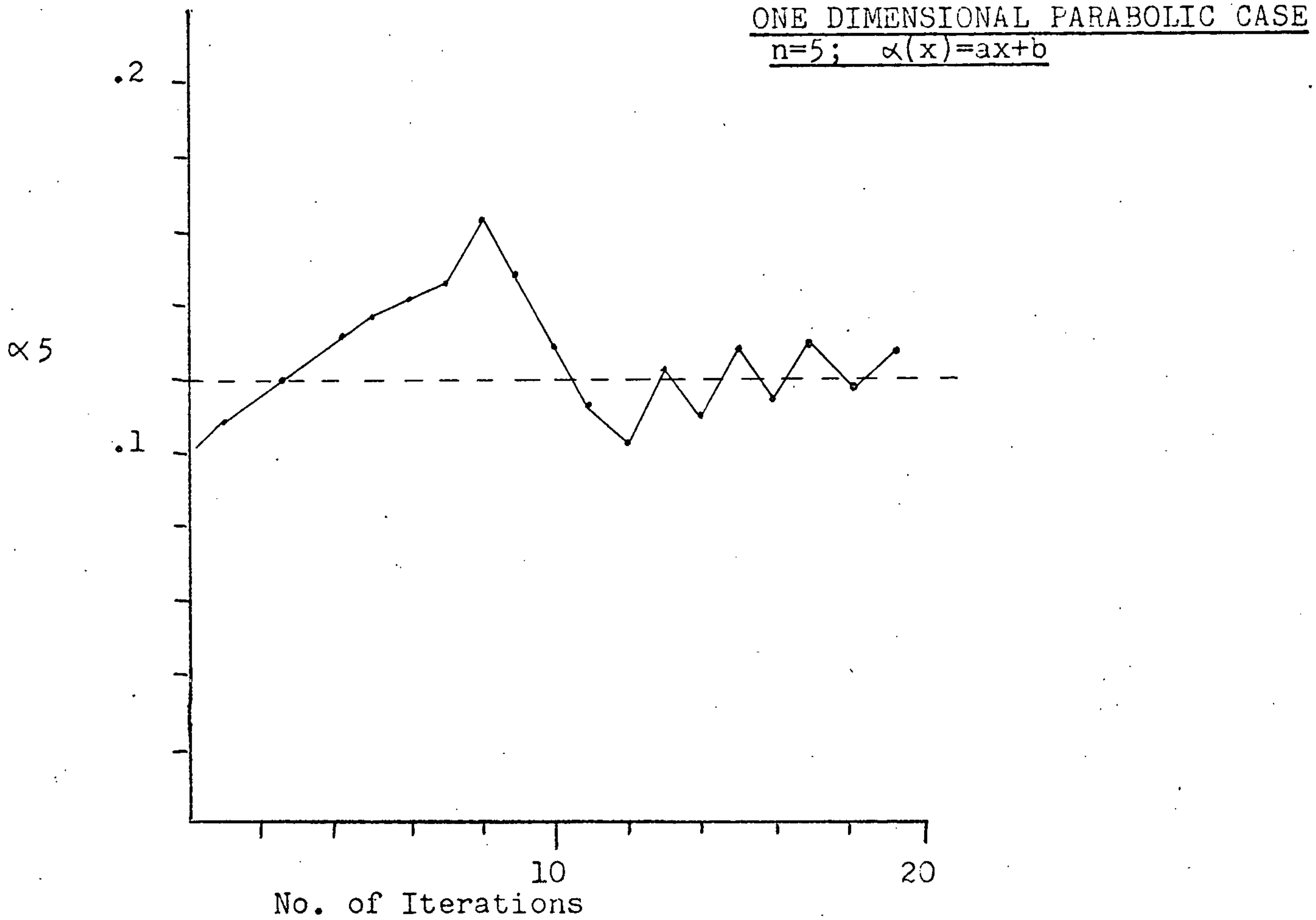

ONE DIMENSIONAL PARABOLIC CASE
n=5; α(x)=ax+b
.2
α5
.1
10
20
No. of Iterations

Figure 4.9

of equations to be solved, other than the computer time required.

4.4 Identification of the Two-Dimensional Parabolic Differential Equation

We would like to identify the unknown parameters of the following distributed parameter system

$$\frac{\partial s}{\partial t}(x,y,t) = r(x)\frac{\partial^2 s}{\partial x^2}(x,y,t) + v(y)\frac{\partial^2 s}{\partial x^2}(x,y,t) \tag{4.23}$$

where $x\varepsilon(0,x_f)$, $y\varepsilon(0,y_f)$, $t\varepsilon(0,t_f)$ with boundary conditions

$$\begin{aligned} s(x,y,t)\big|_{x=o.} &= M_1(y,t) \; ; \qquad & s(x,y,t)\big|_{x=x_f} &= M_2(y,t) \\ s(x,y,t)\big|_{y=o.} &= N_1(x,t) \; ; \qquad & s(x,y,t)\big|_{y=y_f} &= N_2(x,t) \end{aligned} \tag{4.24}$$

and initial conditions

$$s(x,y,t)\big|_{t=o.} = P(x,y)$$

Let the model be

$$\frac{\partial q}{\partial t}(x,y,t) = \alpha(x)\frac{\partial^2 q}{\partial x^2}(x,y,t) + \beta(y)\frac{\partial^2 q}{\partial x^2}(x,y,t) \tag{4.25}$$

with the above initial and boundary conditions.

Let the performance function be

$$J = \int_o^{t_f}\int_o^{x_f}\int_o^{y_f}(q(x,y,t) - s(x,y,t))^2 dx\, dy\, dt \tag{4.26}$$

Discretizing equations (4.23) - (4.26) with respect to the spatial variables x and y, we obtain

$$\dot{s}_{i,j} = \frac{r_i}{\Delta x^2}[s_{i+1} - 2s_i + s_{i-1}]_j + \frac{v_j}{\Delta y^2}[s_{j+1} - 2s_j + s_{j-1}]_i \tag{4.27}$$

$$\dot{q}_{i,j} = \frac{\alpha i}{\Delta x^2}[q_{i+1} - 2]_i + q_{i-1}]_j + \frac{\beta j}{\Delta y^2}[q_{j+1} - 2q_j + q_{j-1}]_i \tag{4.38}$$

where $i = 1, ..n$ and $j = 1, ..., p$

with boundary conditions

$$s_{0,j} = M_{1,j} \quad ; \qquad s_{n+1,j} = M_{2,j}$$
$$s_{i,0} = M_{1,i} \quad ; \qquad s_{i,n+1} = N_{2,i} \tag{4.29}$$

and initial conditions

$$s_{i,j}(0) = P_{i,j}$$

and performance function

$$J = \Delta x \Delta y \int_o^{t_f} \sum_{i=1}^{n} \sum_{j=1}^{m} (q_{i,j} - s_{i,j})^2 dt \tag{4.30}$$

Differentiating J with respect to $_k$ and $_m$ we obtain

$$\frac{\partial J}{\partial \alpha k} = 2\Delta x \Delta y \int_o^{t_f} \sum_{i=1}^{n} \sum_{j=1}^{p} (q_{i,j} - s_{i,j}) U_{i,j,k} \, dt \tag{4.31}$$

$$\frac{\partial j}{\partial \beta_m} = 2\Delta x \Delta y \int_o^{t_f} \sum_{i=1}^{n} \sum_{j=1}^{p} (q_{i,j} - s_{i,j}) U_{i,j,m} \, dt \tag{4.32}$$

Substituting equations (4.28) - (4.29) into the two dimensional update algorithms

$$\Delta\alpha_k = -K \frac{\partial J}{\partial \alpha_k} \quad \text{and} \quad \Delta\beta_m = -L \frac{\partial J}{\partial \beta_m}$$

yields

$$\Delta\alpha_k = -K \int_o^{t_f} \sum_{i=1}^{n} \sum_{j=1}^{p} (q_{i,j} - s_{i,j}) U_{i,j,k} \, dt \tag{4.33}$$

$$\Delta\beta_m = -L \int_o^{t_f} \sum_{i=1}^{n} \sum_{j=1}^{p} (q_{i,j} - s_{i,j}) U_{i,j,m} \, dt \tag{4.34}$$

The sensitivity coefficients $U_{i,j,k}$ and $U_{i,j,m}$ are obtained by using equations (3.15) and (3.16).

4.4.1 Example #5

In this example Meissinger's method was employed to obtain the sensitivity coefficients. Here $n = m = 3$. was chosen. The equations considered were (4.27) and (4.28) with boundary conditions $q_{o,j} = q_{i,o} = q_{x_f,j} = q_{i,y_f} = 0$ for $i = 1,.n$ and $j = 1,.m$ and initial conditions $q_{i,j}(0) = 1$. These were chosen simply for convenience.

$r(x)$ and $v(y)$ were set as a linear function of x and y i.e. $r(x) = ax + b$ and $v(y) = ay + b$. With $a = .25$ and $b = .05$, the discretized parameters r_i and v_j become (.1, .15, .2). Figures (4.10) - (4.13) show that the parameter guesses of (.2, .2, .2) and (.05, .05, .05) converged to the optimum.

For the two dimensional case Meissinger's method requires the solution of $(2xnxn^2)$ equations, in our case this leads to 54 sensitivity equations.

The problem associated with identification using the method of Bingulac and Kokotovic is the number of equations that are required to be solved. For the two dimensional case as outlined above; nine system equations are required, nine model equations are required, as well as the 54 sensitivity equations. For this example CSMP requires 62 seconds to compile the program versus the 12 seconds of compilation time needed for example #1. This, of course, is due to the number of equations that must be solved.

4.5 Conclusion

In this chapter five examples of identifying parameters in distributed parameter systems using the method of Bingulac and Kokotovic were attempted. Systems for n = 3 were easily identified using Meissinger's method to obtain the sensitivity coefficients. It was found that structural sensitivity yielded no economy in the number of equations that had to be solved, and yielded exactly the

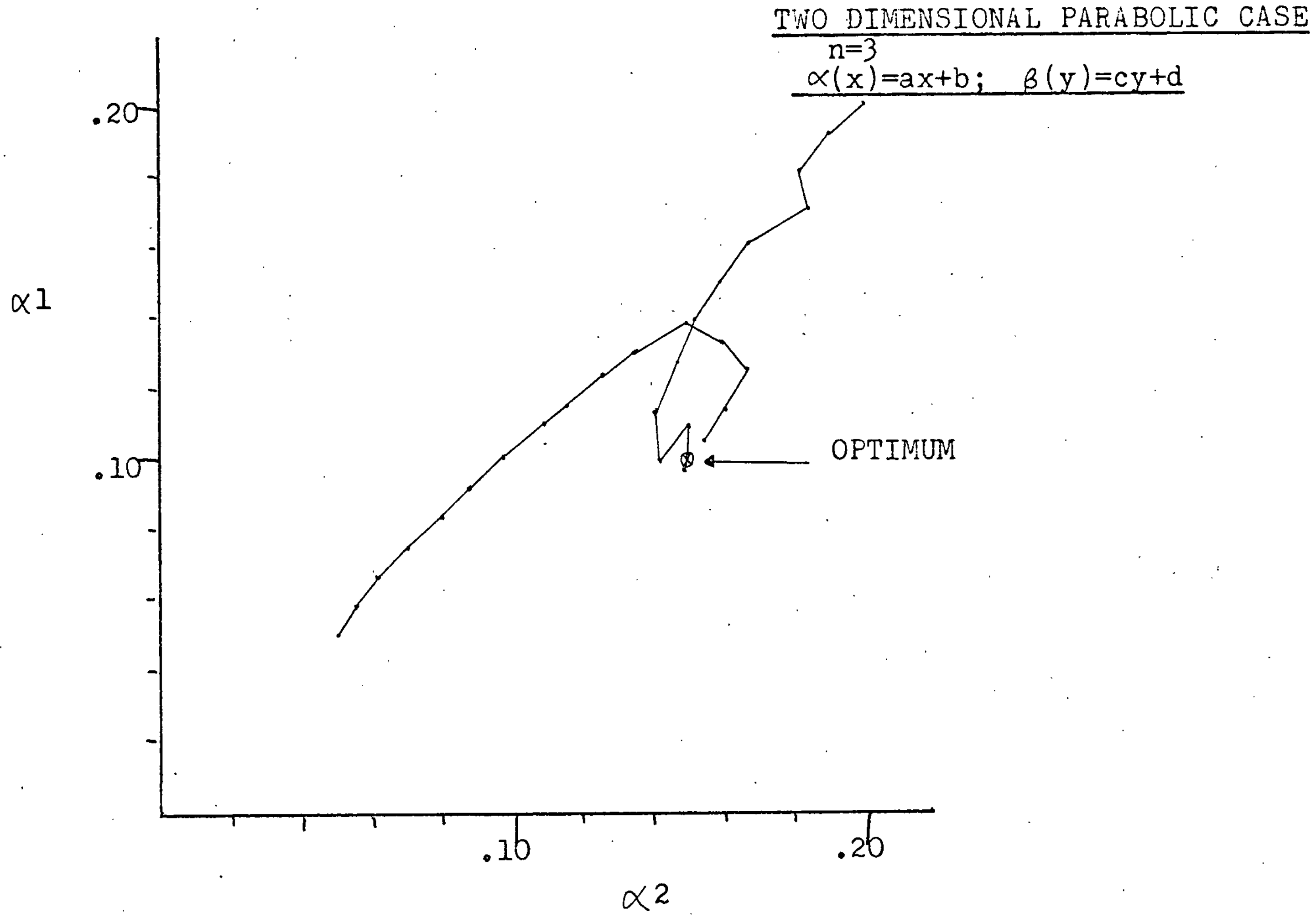

TWO DIMENSIONAL PARABOLIC CASE
n=3
α(x)=ax+b; β(y)=cy+d
.20
.10
α1
OPTIMUM
.10
.20
α2

Figure 4.10

TWO DIMENSIONAL PARABOLIC CASE

$n=3$

$\alpha(x)=ax+b;\quad \beta(y)=cy+d$

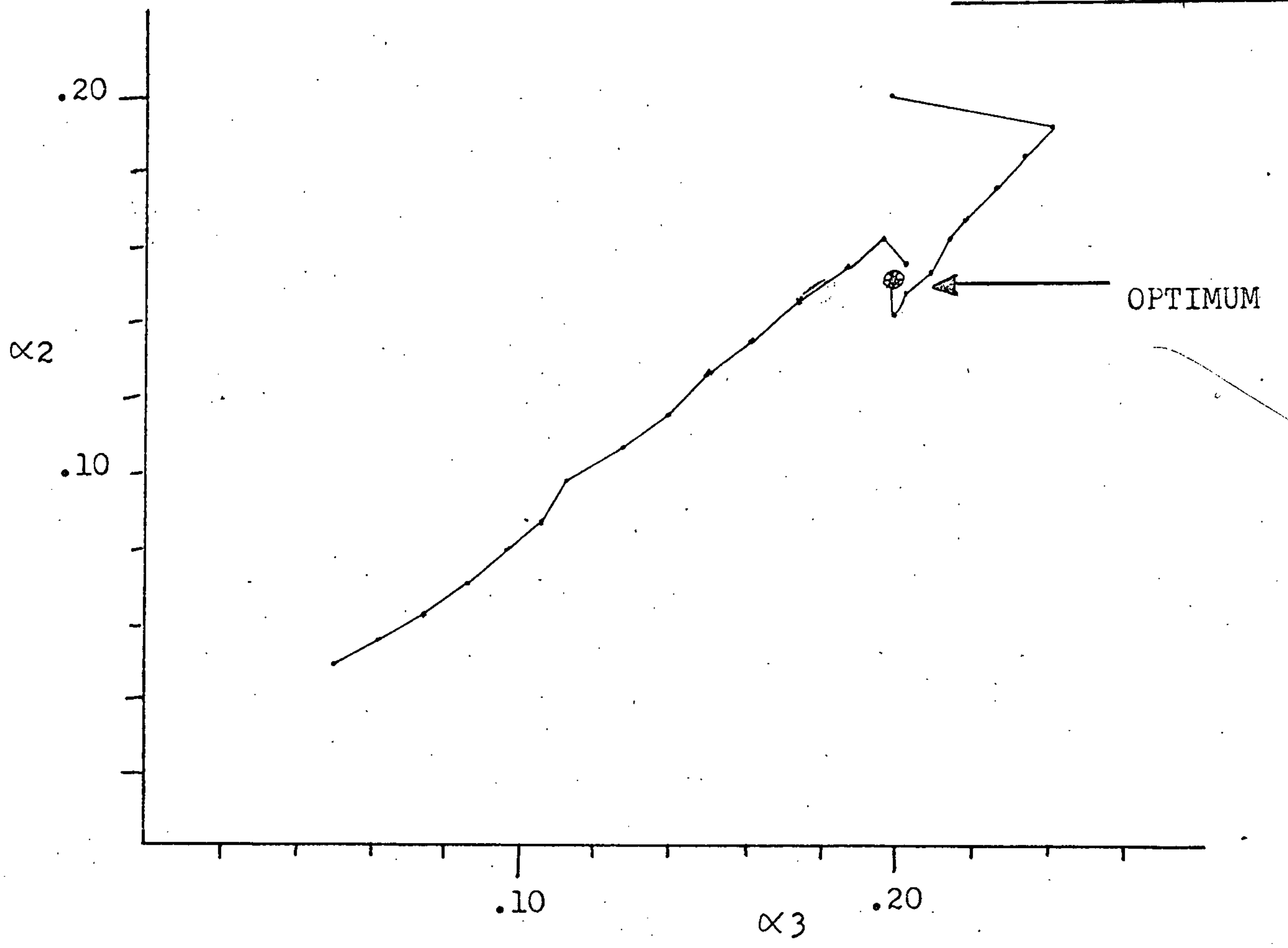


Figure 4. 11

TWO DIMENSIONAL PARABOLIC CASE

$n=3$

$\alpha(x)=ax+b; \quad \beta(y)=cy+d$

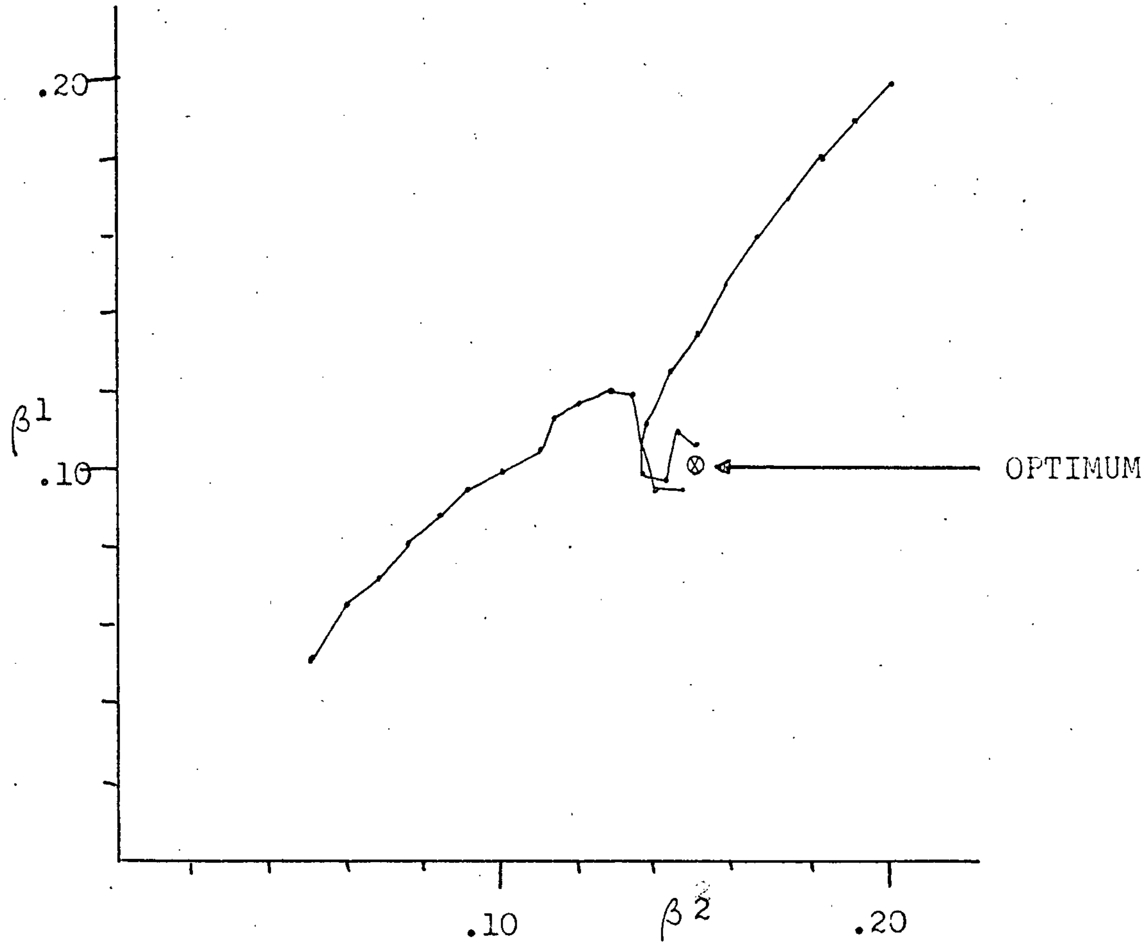


Figure 4 · 12

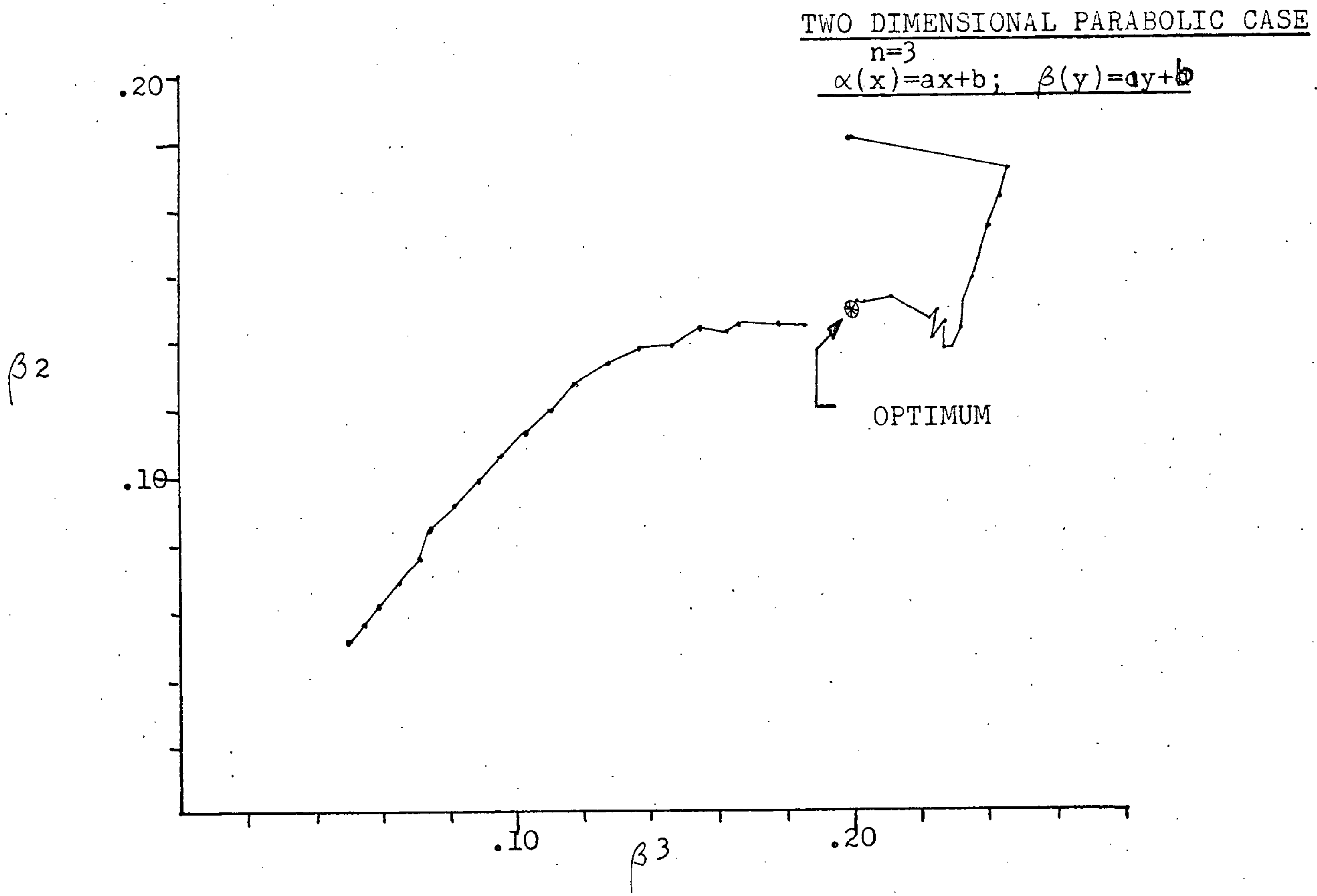

TWO DIMENSIONAL PARABOLIC CASE
n=3
α(x)=ax+b; β(y)=ay+b
.20
.10
β2
OPTIMUM
.10
.20
β3

Figure 4 . 13

same results as Meissinger's method as can be seen in Examples #1 and #2.

For the nonlinear parameter choice in Example #3, it was found that a great deal of oscillation occurred while approaching the optimum. This was due to the ridge in the performance function.

For Example #4 in which n = 5, the five parameters were identified using Meissinger's method, however, the compilation time of CSMP was double that of Example #1. This resulted because 25 sensitivity equations had to be solved versus 9 sensitivity equations in Example #1.

For the two dimensional case, Example #5, the parameters were easily identified using Meissinger's method, however compilation time of CSMP was six times that of Example #1. This resulted because 54 sensitivity equations had to be solved in this example.

Though the method of Bingulac and Kokotovic can successfully be employed to identify spatially dependent parameters in distributed parameter systems, its ultimate failing is in the number of sensitivity equations that must be solved.

5. CONCLUSION

5.1 Summary

This work has been concerned with the identification of distributed parameter systems of the parabolic type with spatially dependent parameters. The method of Bingulac and Kokotovic, Meissinger's method, and structural sensitivity were extended to identify the above mentioned systems.

Chapter 2 deals with the discretization schemes available to discretize partial differential equations. It was seen that spatial discretization could be considered with CSMP provided that equation (2.19) is satisfied.

Chapter 3 develops Meissinger's method and the method of structural sensitivity for discretized distributed parameter systems. It was found that structural sensitivity can only be applied to those systems of the Kokotovic class; and this implies that one boundary condition must be set to zero. Structural sensitivity does not yield an economy in the number of sensitivity equations that need be solved as is the case in ordinary differential equations. In fact, structural sensitivity requires the solution of the same number of sensitivity equations as that of Meissinger's method, and so structural sensitivity is of little use.

Chapter 4 deals with the extension of the method of Bingulac and Kokotovic to the identification of distributed parameter systems using the sensitivity methods developed in the previous chapter. Both sensitivity methods were employed and it was found that they yielded the same results. Though the method of Bingulac and Kokotovic can be successfully employed; its ultimate failing lies with the number of sensitivity equations that need be solved.

5.2 Suggestions for Further Research

The problem under study is suited to the hybrid computer and it would be very interesting to approach it from that point of view. This would avoid the need for analogue multipliers, track and hold devices, and external mode control devices, so that the number of equations to be considered in the analogue portion of the computer could be increased.

APPENDIX A

A.1 The Continuous Modeling Programme

The S/360 CSMP is a problem oriented programme designed to facilitate the digital simulation of continuous processes on large scale digital computers.

The general CSMP formulation of a model is divided into three segments: Initial; Dynamic; and Terminal; that describe the computations to be performed before, during, and after each simulation run.

The initial segment is intended exclusively for computation of initial conditions. The dynamic segment consists of the system dynamics. The terminal segment is used for those computations desired after completion of each run.

A.2 The Method of Bingulac and Kokotovic Using CSMP

The method of Bingulac and Kokotovic can easily be programmed using continuous system modeling. The two modes of operation discussed in Section (4.2), i.e. the compute and the operate modes are obtained in the following manner.

The compute mode i.e. the mode in which the parameters are held constant is simply obtained by setting the parameters to a set value in the initial segment of the programme.

The reset mode i.e. the adjustment of the parameter values by means of the steepest descent algorithm (equation 4.3), is obtained by programming the equation in the terminal segment of the programme. The updated parameter values are then used as the initial conditions for the next sweep.

APPENDIX B: ANALOGUE COMPUTER WORK

B.1 Introduction

This appendix deals with the possible analogue computer implementations for the identification of distributed parameter systems, and their failings. This results from a need for a large number of components in the analogue computer, namely; multipliers and integrators. The large number of integrators is needed in order to build the track and hold circuits required.

B.2 The Track and Hold Circuit

The track and hold circuit is obtained by using a pair of mode controlled integrators. To obtain a mode controlled integrator the Memory Logic Group (MLG) unit must be engaged. With the proper configuration on the MLG patch panel, an S = 1 mode pulse will put a designated integrator into the set mode. An $\bar{S}$ = 1 pulse will put the integrator into the reset mode.

A track and hold circuit can be obtained by using a pair of mode controlled integrators connected as in Figure B.4. In the S = 1 mode, integrator 1 follows the input and the relay in the figure sets the output to ground. In the $\bar{S}$ = 1(S=0) mode, integrator 2 tracks integrator 1. Since integrator 1 is held at its final value, integrator 2 holds that value with the relay open.

B.3.1 Meissinger's Method; Implementation #1

The analogue computer implementation of this method can be accomplished by using the repetitive operation mode on the analogue computer.

For a system with n=3., three system, three model, nine sensitivity equations and three update routines are required.

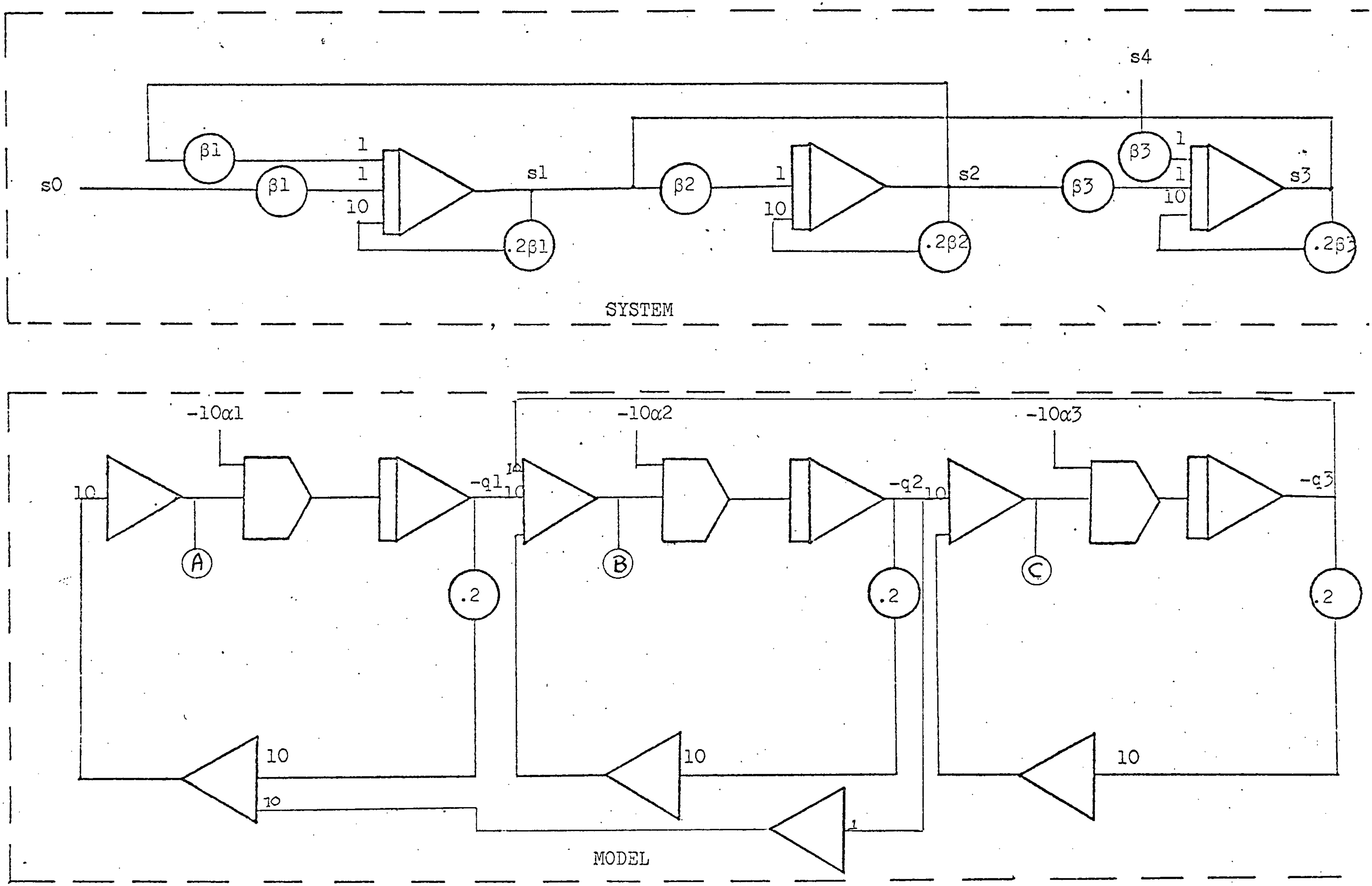

s4
β1
1
1
β1
s0
10
s1
β2
1
10
s2
β3
β3
1
1
10
s3
.2β1
.2β2
.2β3
SYSTEM
-10α1
-10α2
-10α3
10
-q1
10
10
-q2
10
-q3
A
B
C
.2
.2
.2
10
10
10
10
1
MODEL

FIGURE B.1

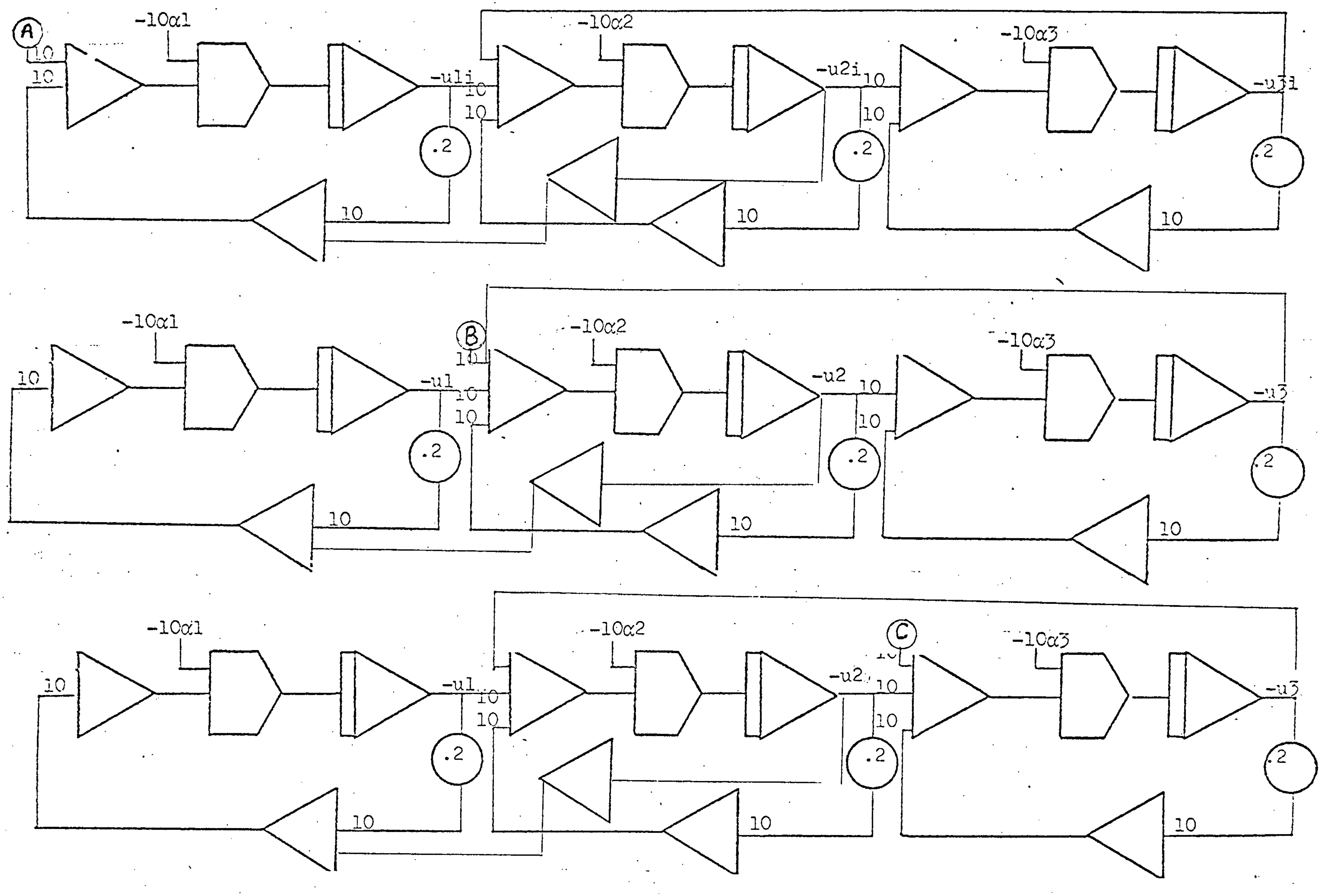


FIGURE B.2

UPDATE SEGMENT

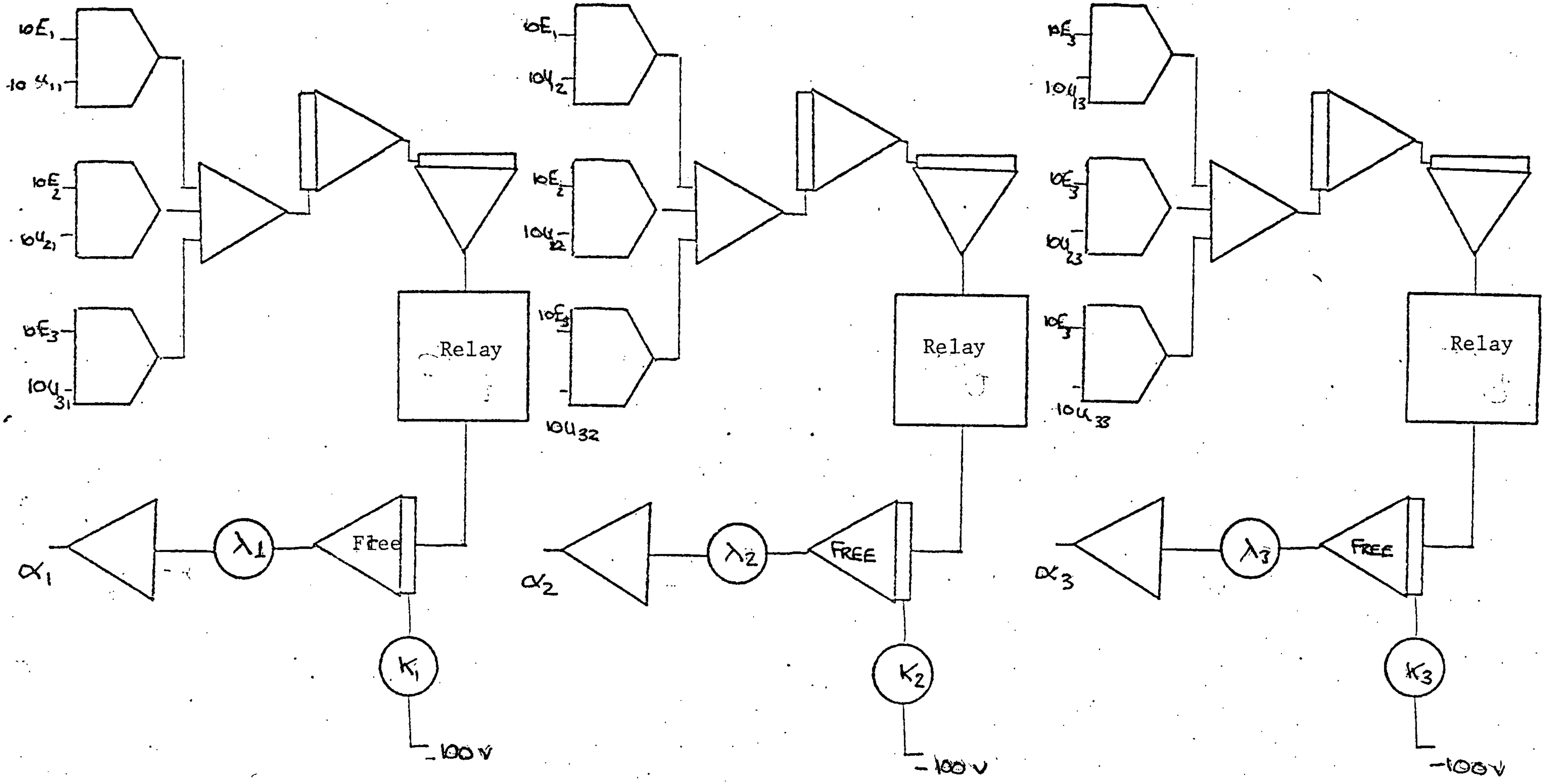


Figure (B.3)

Putting the update routines into the continuous mode and all the other integrators into the repetitive operation mode, we can implement the scheme in Figures (B.1) - (B.3) provided we have a sufficient supply of components in the computer. Since three multipliers are required in the model equation, nine multipliers are required in the update routine, and nine multipliers are required in the sensitivity model, we can not implement the method on the analogue computer as the analogue computer has only twelve multipliers.

B.3.2 Meissinger's Method; Implementation #2

Since the structure of the sensitivity model equations is very similar, as can be seen in equation (3.7), we can cycle through three sensitivity equations with three different inputs to obtain the nine sensitivity equations.

This scheme involves five modes of operation, namely; one mode (M and $\bar{M}$) for the system and the model, three modes (S1, S2, S3) for the sensitivity model, and one mode (S4) for the update routine.

Since the analogue computer has at most two modes of operation, it is necessary to design an external mode controller. This device consists of three flip flops, four AND gates and four inverters, as seen in Figure B.5. The mode control signals previously described appear in Figure B.6.

The analogue computer implementation can be seen in Figures B.7 and B.8. All integrators with no mode control indicated in both the system and the model are in the M mode. All the integrators with no mode control indicated in the sensitivity model are in the $\bar{M}$ mode. Relay #1 and the track and hold devices ensure that a,b,c, enter the circuit once during the cycle and at the correct time.

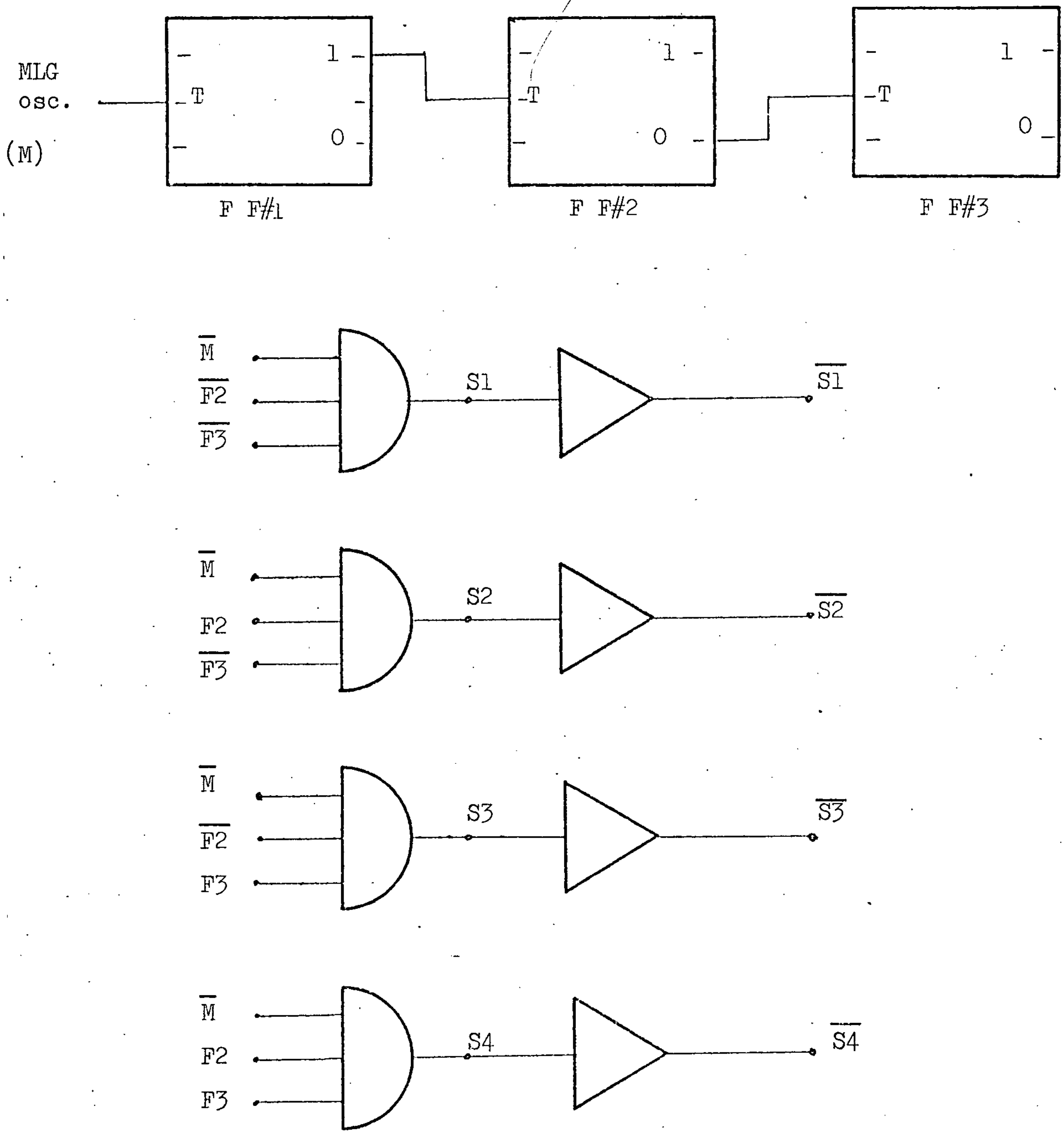

MLG osc.
(M)
T
1
0
F F#1
F F#2
F F#3
$\overline{M}$
$\overline{F2}$
$\overline{F3}$
S1
$\overline{S1}$
$\overline{M}$
F2
$\overline{F3}$
S2
$\overline{S2}$
$\overline{M}$
$\overline{F2}$
F3
S3
$\overline{S3}$
$\overline{M}$
F2
F3
S4
$\overline{S4}$

FIGURE B.5

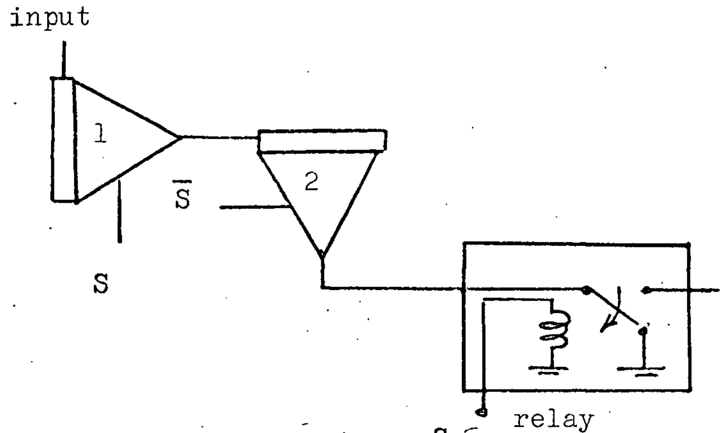


FIGURE B.4

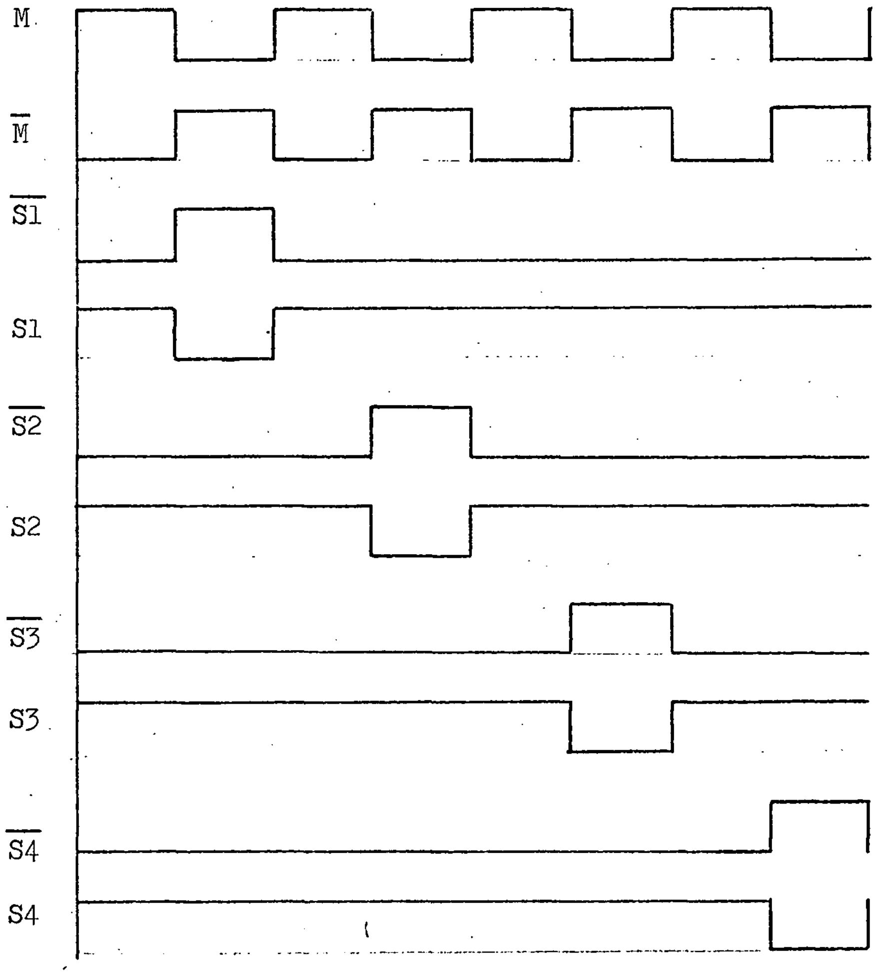


FIGURE B.6

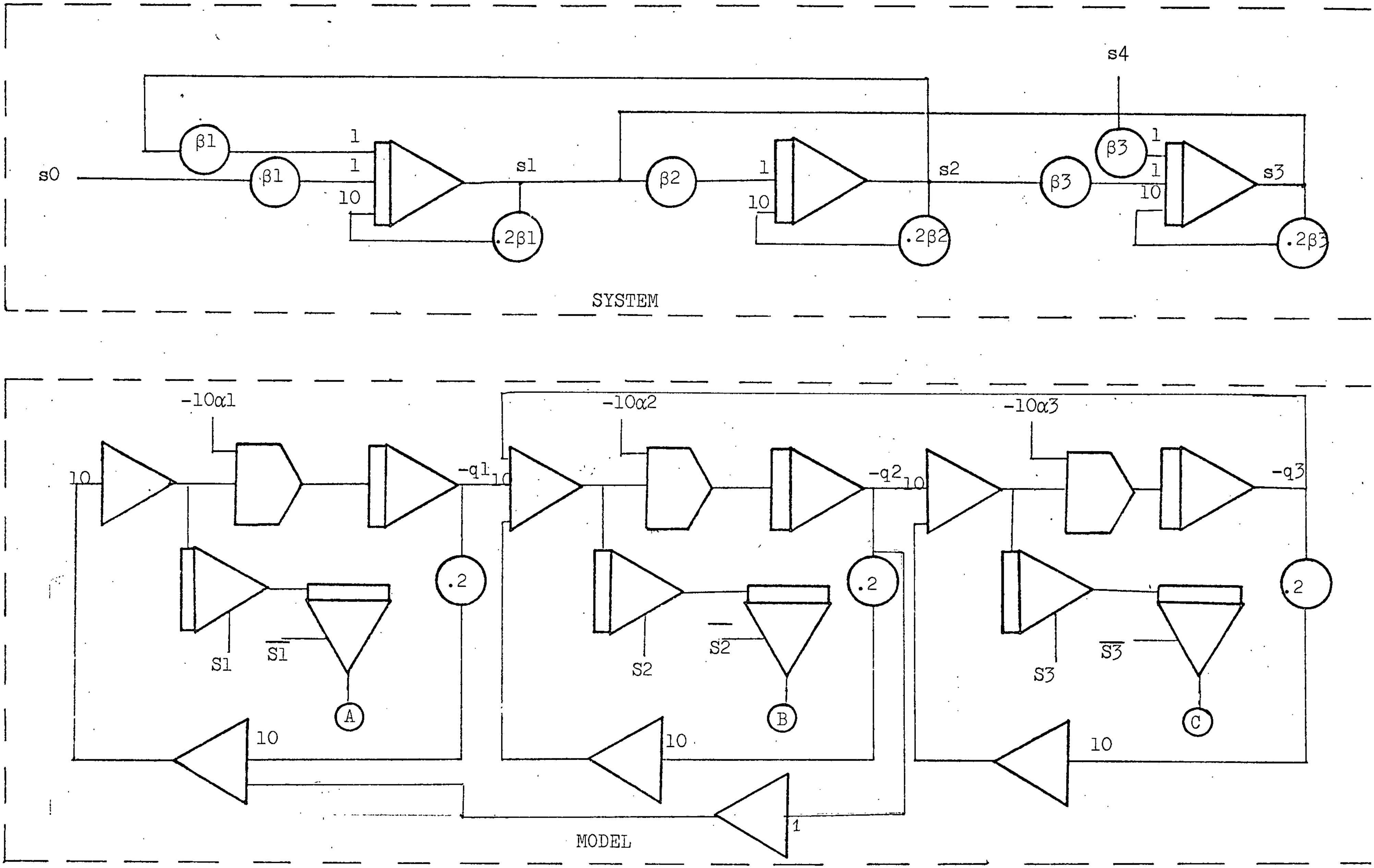

s4
β1
1
1
s0
β1
10
s1
β2
1
10
s2
β3
β3
1
1
10
s3
.2β1
.2β2
.2β3
SYSTEM
-10α1
-10α2
-10α3
10
-q1
10
-q2
10
-q3
.2
.2
.2
S1
$\overline{S1}$
S2
$\overline{S2}$
S3
$\overline{S3}$
A
B
C
10
10
10
1
MODEL

FIGURE B.7

SENSITIVITY MODEL

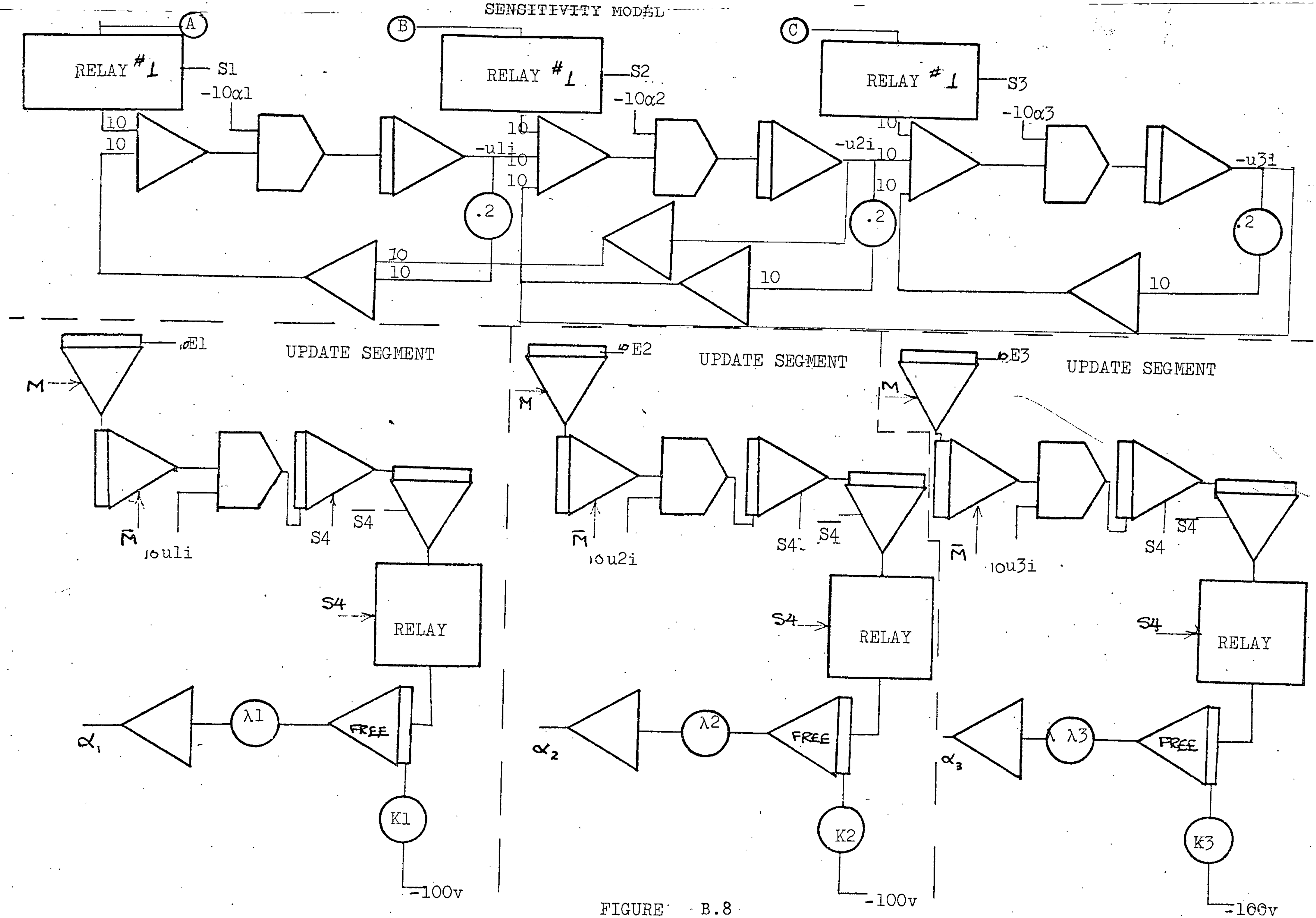


FIGURE B.8

Since the implementation of this scheme requires 30 integrators, the analogue computer could not be used, as only 26 integrators are available on the computer.

B.4 Structural Sensitivity Implementation

Since the sensitivity transfer functions are identical for each input, input switching was considered. The same mode control as the previous example can be used. Using this scheme, the number of integrators needed exceed the number available, and so the computer could not be used.

B.5 Use of the Hybrid Computer

The problem with all the methods cited is either their lack of multipliers or the lack of storage elements (track and hold devices i.e. integrators). Using the hybrid computer for multiplication and storage as well as integrator mode control would eliminate these difficulties.

REFERENCES

1. Hsia, T.S., and V. Vimolvanich, "An On-line Technique for System Identification", Institute of Electrical and Electronic Engineers Transactions on Automatic Control, February, 1969. pp. 92-96.

2. Gavrilov, M., and R. Petrovic, "Parameter Identification via System Linearization and Optimal Improvement of Parameter Values", 5th International Analogue Computation Meetings. pp. 545-549, 1967.

3. Kohr, R., "On the Identification of Linear and Nonlinear Systems", Simulation, March, 1967. pp. 165-174.

4. Nagumo, J., and Noda, A., "A Learning Method for System Identification", Institute of Electrical and Electronic Engineers Transactions on Automatic Controls, vol. AC-12, No. 3, June, 1967, 282-287.

5. Vuskovic, M.I., J.V. Medanic, and W. Schaufelberger, "Investigation of a Class of Fast Adaptive Control Systems", 5th International Analogue Computation Meetings. pp. 567-574, 1967.

6. Butler, R.E., and E.V. Bohn, "An Automatic Identification Technique for a Class of Nonlinear Systems," IEEE Transactions on Automatic Controls, vol. AC-11 no. 2, April, 1966. pp. 292-296.

7. Mishkin, E. and Braun, L. "Adaptive Control Systems", Toronto, McGraw-Hill, 1961.

8. Kokotovic, P., "Method of Sensitivity Points in the Investigation and Optimization of Linear Control Systems", Automatika i Telemekhaneka, vol. 25 no. 12, pp. 1670-1676, December, 1964.

9. Kokotovic, P. and R.S. Rutman, "Sensitivity of Automatic Control Systems (Survey)", Automatika i Telemekhanika, vol. 26, no. 4, pp. 730-750, April, 1965.

10. Tomovic, R., "Sensitivity Analysis of Dynamic Systems", McGraw-Hill, N.Y., 1963.

11. Meissinger, H.F., "The Use of Parameter Influence Coefficients in Computer Analysis of Dynamic Systems", Proceedings of the Western Joint Computer Conference, pp. 181-192, May, 1960.

12. Rosenbrock, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer Journal, vol. #3, pp. 175-184, 1960.

13. Beveridge, G. and R.S. Schechter, "Optimization Theory and Practice", McGraw-Hill, N.Y., 1970 pp. 363-367.

14. Powell, M.J.D., "An Iterative Method for Finding Stationary Values of a Function of Several Variables", Computer Journal, vol. #5, pp. 147-151, 1955.

15. Curry, H.B., "The Method of Steepest Descent for Nonlinear Minimization Problems", Quarterly of Applied Mathematics vol. #2, pp. 258-261, October, 1944.

16. Fletcher, R., and C.M. Reeves, "Function Minimization by Conjugate Gradients", British Computer Journal, pp. 163-171, July, 1964.

17. Bingulac, S.P., "Parameter Optimization on an Analog Computer", 5th International Analogue Computation Meetings, pp. 562-566, 1967.

18. Collins, P.L., and H.C. Khatre, "Identification of Distributed Parameter Systems Using Finite Differences", Transactions of the ASME Journal of Basic Engineering, June, 1969, pp. 239-245.

19. Tzafestas, S.G., "Identification of Stochastic Distributed Parameter Systems, International Journal of Control, vol. 11, no. 4, pp. 619-624, 1970.

20. Perdreauville, F.J. and R.E. Goodson, A.S.M.E. Journal of Basic Engineering, pp. 463- , 1966.

21. Shinbrot, M., NACA Tech. Note IN3288, 1954.

22. Fairman, F.W., and D.W.C. Shen, "Parameter Identification for a Class of Distributed Systems", Int. J. Control, vol. #11, no. 6, pp. 929-940, 1970.

23. Bensoussan, A., "Identification de Systemes Gouvernés per des Equation aux Derivées Particles", Computing Methods in Optimization Problems. no. 2. ed. Zadeh, L.A., Neustadt, L.W., and Balakmshian, A.V. pp. 25-34, A.P. 1969.

24. Crank, J., and P. Nicholson, "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conducting Type", Proc. Cambridge Philos Soc., vol. #43, pp. 50-67, 1947.

25. Bekey, G.A., and W.J. Karplus, Hybrid Computation, Wiley, N.Y., 1968. pp. 214-221.

26. Bingulac, S. and P. Kokotovic, "Automatic Optimization of Linear Feedback Control Systems on an Analogue Computer", Proc Internat. Assoc. for Analogue Computation, vol. #7, pp. 12-17, January, 1965.

27. Korn, G.A., "Random Process Simulation and Measurements", McGraw-Hill, N.Y., 1966.

28. Tomkins, C.B., "Modern Matematics for the Engineer", E.F. Beckenbach, ed. McGraw-Hill, Toronto, 1956, pp. 448-479.

29. Forsythe, G.E. and Wasow, W.Z. "Finite Difference Methods for Partial Differential Equations", Wiley, N.Y., 1960, pp. 88-139.

30. Gilbert, E.G., "A Selected Bibiography, on Parameter Optimization Methods Suitable for Hybrid Computation", Simulation, vol. #8, pp. 350-52, June 1967.

31. Vuscovic, M. and Ciric, V., "Structural Rules for the Determination of Sensitivity Fuctions of Nonlinear Nonstationary Systems", Sensitivity Methods in Control Theory ed. Radanovic L., <u>Pergamon Press</u>, 1964, pp. 153-165.