To the Graduate Council:

I am submitting herewith a thesis written by Pen Yuan Chou entitled "Second Order Plus Dead Time Analysis of Experimental Data." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Chemical Engineering.

Charles F. Moore

C. F. Moore, Major Professor

We have read this thesis and recommend its acceptance:

H. W. Haw

Deane D Bruns

Accepted for the Council:

L Evan Pahl

Vice Chancellor
Graduate Studies and Research

Thesis
80
.C368
cop 2

SECOND ORDER PLUS DEAD TIME ANALYSIS OF EXPERIMENTAL DATA

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Pen Yuan Chou

June 1980

3041835

ACKNOWLEDGEMENTS

I am indebted to the following persons for their assistance in producing this thesis:

Dr. C. F. Moore, my advisor, provided much help in conceiving and directing the course of this investigation.

Dr. H. H. Hsu and Dr. D. D. Bruns served as Committee members and gave their valuable time.

I am also deeply indebted to my parents for their support and encouragement.

ABSTRACT

Exerimental modeling has been an approach used most often in constructing a system model for the purpose of control system studies. One of the experimental modeling techniques is to use a nonlinear optimization method to search the parameters of a model for the parameters which give the best fit to the experimental data.

An attractive feature of the technique is that the technique can adapt to either linear or nonlinear models without involving mathematical manipulations. However, the final results and the efficiency of this technique depends on the relations between the objective function and its independent variables. The relations are usually visualized as the contour surface of the objective function. Search of the optimum becomes difficult if two or more independent variables are all related to same characteristic of the objective function. On the contour surface this causes inclined ridges. Unfortunately, most nonlinear optimization methods were not particularly designed to deal with the ridge problem.

In this work it demonstrated that the ridge problem could be eliminated by proper selection of the independent variables and design of experiment. Also a nonlinear optimization method was modified to deal with objective functions with extremely narrow curved ridges. With proper

design of experiment and the modified optimization technique, the ridge problem in experimental modeling was successfully overcome.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	THEORY AND LITERATURE REVIEW	4
	A. Digital Sampled-data System	4
	B. Experimental Modeling	7
	C. Nonlinear Programming	10
III.	STATEMENT OF THE PROBLEM	18
IV.	EXPERIMENTAL MODELING BY USING A NONLINEAR REGRESSION METHOD	29
	A. Coordinate Transformation	29
	B. Design of Experiment	39
	C. Modification of the Pattern Search Method	57
V.	APPLICATIONS OF THE MODIFIED PATTERN SEARCH METHOD	71
	A. Experimental Modeling	71
	B. Controller Tuning	83
VI.	CONCLUSIONS AND RECOMMENDATIONS	92
LIST OF REFERENCES		94
APPENDICES		97
	A. Derivation of Second Order Plus Dead Time System Equation in Z-domain	98
	B. Software Programs	103
VITA		133

LIST OF TABLES

TABLE		PAGE
I.	Step Response Data Modeling of System I by Using the Pattern Search Method	20
II.	Step Response Data Modeling of System II by Using the Pattern Search Method	21
III.	Number of Function Evaluations for the Minimization of Function I	70
IV.	Final Point Reached in the Minimization of Function II	70
V.	Experimental Modeling by Using Noiseless Response Data of a Second Order System	72
VI.	Models Obtained from Noise Corrupted Response Data of System III.	75
VII.	Models Obtained from Noise Corrupted Response Data of System IV	77
VIII.	Second Order Models Obtained by Using Step Response Data of a Third Order System.	82
IX.	Second Order Models Obtained by Using Impulsive Response Data of a Third Order System.	82

LIST OF FIGURES

FIGURE		PAGE
1.	Digital Sampled-data Control System.	5
2.	Second Order Plus Dead Time System with a Zero Order Hold Element.	6
3.	System Identification by Using a Nonlinear Optimization Method.	9
4.	Flexible Polyhedron Search	15
5.	Strategy of the Pattern Search	16
6.	Program Structure of the Pattern Search Method.	19
7.	Responses of System II and Model 1 in TABLE II.	23
8.	Projections of the Step Response Modeling Objective Function of System I	24
9.	Elimination of an Inclined Ridge by Transformation of Coordinate	31
10.	A Second Order System.	31
11.	Unit Step Response of a Second Order System. . . .	33
12.	Elimination of a Curved Ridge in Step Response Modeling by Coordinate Transformation.	35
13.	Projections of the Step Response Modeling Objective Function by Using Difference Equation Coefficients as Search Parameters	37
14.	The Effect of Sampling Period on the Step Response Modeling Objective Function	41
15.	Response of a Second Order System to a Combination of Impulsive Forcing Function.	45
16.	Projections of the Impulsive Response Modeling Objective Function of System I	47

FIGURE PAGE

17. Projections of the Impulsive Response Modeling Objective Function by Using the Coefficients of Difference Equation as the Search Parameters 49

18. Response of a Second Order System to Pseudo-random Binary Sequence. 51

19. Projections of the Contour Surface of PRBS Response Data Modeling Objective Function. 52

20. Synchronous Random Telegraph Waves with Different Switching Probabilities. 56

21. Normalization of the Ridge Orientation by Scaling the Coordinate System. 60

22. Search of Rotation Angle in a Properly Scaled Space 64

23. Elimination of a Ridge by Coordinate Scaling and Rotation 64

24. Flow Chart of the Subroutine SEARCH of the Modified Pattern Search. 66

25. Program Structure of the Modified Pattern Search Method. 67

26. System Frequency Response and Power Density Spectrum of System Response to SRTW. 73

27. Power Density Spectrum of SRTW and Frequency Response of System III 80

28. Contour Surface of PI-Controller Tuning for Set Point Change 85

29. Contour Surface of PI-Controller Tuning for Load Change. 86

30. Contour Surface of PID-Controller Tuning for Set Point Change 88

31. Contour Surface of PID-Controller Tuning for Load Change. 90

CHAPTER I

INTRODUCTION

In a digital computer control system, system models are sometimes used to design the control algorithms. A system model becomes an absolute requirement when highly theoretical designs (optimal control) are used. As a matter of fact, modeling is one of the most important parts in the analysis of system dynamic behaviors. System models become intractable if they have to be precise. Fortunately, dynamic behaviors of most systems in chemical industry can be described by simple models without losing too much accuracy.

One of the techniques which have frequently been used to construct system models is the technique of experimental modeling. The concept of this approach is that a model is acceptable if the model and the system response identically or very similarly to the same input signals. Therefore, a straightforward approach is to use a numerical optimization method to adjust the parameters of a model to fit the response data of a system. In this approach the difference between the system response and the model response to a input signal is defined as the objective function to be minimized. A numerical optimization method called pattern search (1) has been used to fit step response data of a discrete system with a second order plus dead time model.

An investigation of the results showed that the pattern search could have stopped on a narrow ridge on the contour surface of the objective function before reaching the optimum.

To resolve the ridge problem, the characteristics of the contour surface of the objective function for modeling were investigated, and three different approaches were proposed to eliminate the effect of ridges on the contour surface. These approaches are described briefly as follows:

(a) Coordinate transformation

A ridge will be formed when the independent variables of an objective function interact one another strongly. From the knowledge of the relations among the parameters of a model, less interactive parameters were chosen as the independent variables.

(b) Design of experimental

The contour surface of system response data depends on the type of the system input signal applied as well as the system itself. The effect of the ridge problem can be reduced by choosing a suitable type of system input signal.

(c) Modification of the pattern search

An optimization method may fail to follow a ridge if its search directions are confined to certain directions, or it is unable to detect precisely the direction of a ridge. The pattern search was modified such that it could search the direction of a ridge and rotate the coordinates

accordingly. This enables the pattern search to follow a curved ridge easily.

In Chapter V the modified pattern search was applied to both experimental process modeling and to controller parameter tuning. It should be noted that the optimization techniques developed in this thesis are general and can be directly applied to other optimization problms.

CHAPTER II

THEORY AND LITERATURE REWIEW

A. Digital Sampled-data Control System

In a digital computer control system, the signals from the sensors of a process and the necessary information are transferred to a digital computer or a data processor. Using those input information, the computer or the data processor performs the computations of control algorithms and decides the magnitudes of control signals to be transferred to control devices. Since a digital computer can work only on digital numbers, a digital-to-analog converter and an analog-to-digital converter are required to work as an interface between the real world and the computer. In addition, systems in the chemical industry are generally continuous so that a hold element is used to transfer impulsive control signals into continuous signals. A typical digital computer control system is illustrated in Figure 1.

In the analysis of a digital computer control system, it is convenient to use a discrete expression for the computer. In order to have a consistent form of expression for the whole control system, the expression for the process of the real world has to be in discrete form also. A discrete expression of a continuous system can be derived by

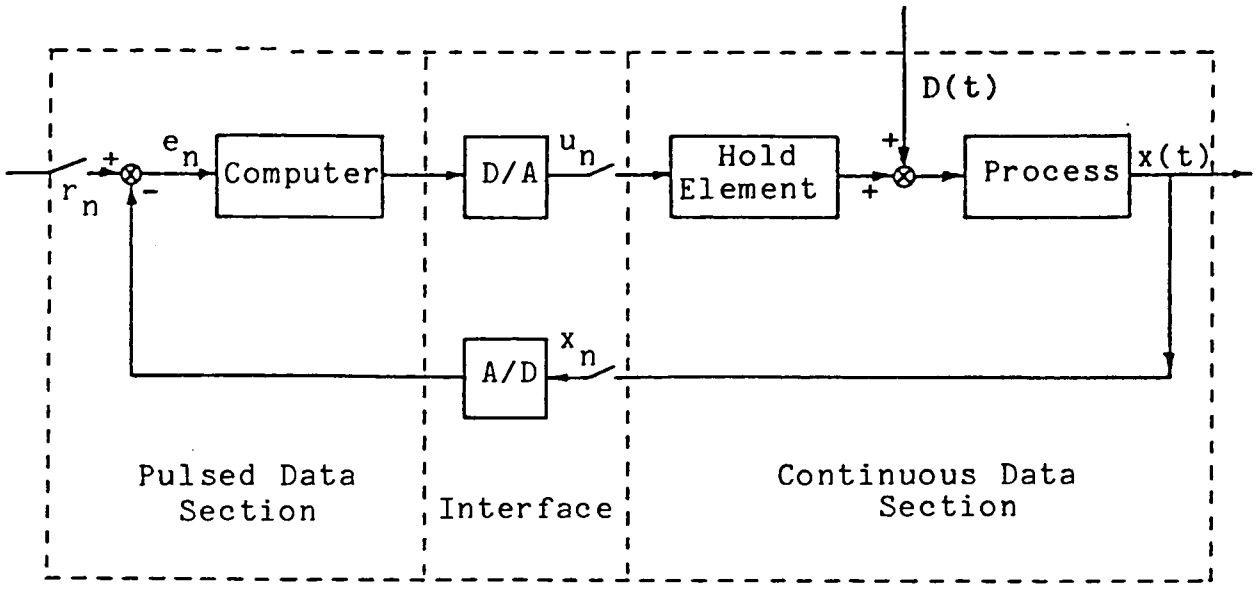


Figure 1. Digital Sampled-data Control System.

using the modified z-transform.

The z-domain expression of the second order plus dead time and the zero order hold element in Figure 2 , was derived in Appendix A, and has the form

$$G(z) = \frac{z^{-N}\,(b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3})}{1 - a_1 z^{-1} - a_2 z^{-2}} = \frac{X(z)}{U(z)} \qquad (1)$$

where $G(z) = X(z)/U(z)$, and N is an integer. N satisfies $N = \theta/T-(1-m)$, where T is the sampling period, θ is the dead time of the system, and $0 < m < 1$.

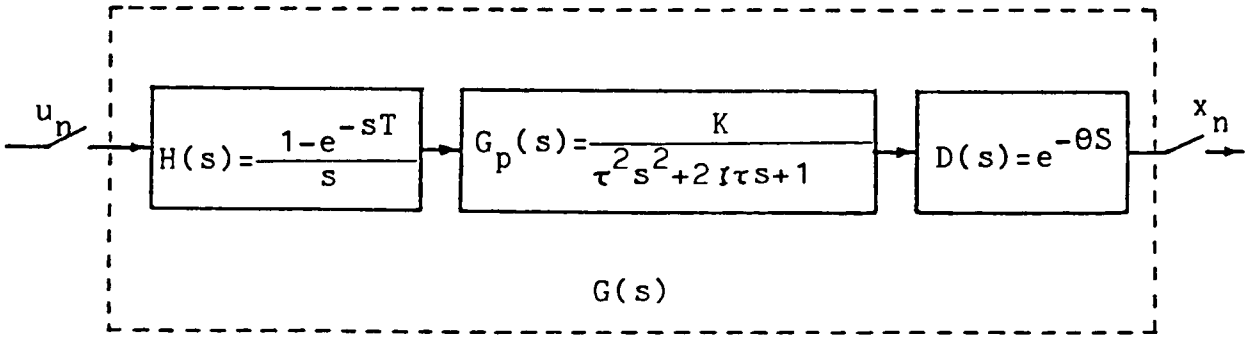


Figure 2. Second Order Plus Dead Time System with a Zero Order Hold Element.

In difference equation form, equation (1) becomes

$$x_i = b_1 u_{i-(N+1)} + b_2 u_{i-(N+2)} + b_3 u_{i-(N+3)} + a_1 x_{i-1} + a_2 x_{i-2} \qquad (2)$$

where u_i and x_i are input and output of the system at the ith sampling time.

As a matter of fact, equation (2) can also be used as a general form for a first order system or a second-order-lag-plus-lead system. If what we need is just an equation to represent the system input-output relation, a difference equation is often used without derivation and has the general form:

$$x_i = \sum_{j=1}^{n} a_j x_{(i-j)} + \sum_{j=0}^{n} b_j u_{(i-N-j)} \qquad (3)$$

where n is the order of the model.

B. Experimental Modeling

For most processes in industry, it is very complex to derive a system model analytically. A more straightforward approach is the experimental modeling (2, 3, 4, 5). In this approach a model can be derived from the input-output data either in the time domain or in the frequency domain.

(a) Frequency, step, and impulse response method

In the frequency response method, the frequency response data are used to construct a Bode diagram. A model is then derived from the Bode diagram. There are several difficulties involved in the application of frequency response method. First, it needs a device to generate a precise sinusoidal input to the system. In addition, a rather long period of time is required to let the response reach the final frequency, and the experiment has to be repeated at several different frequencies.

If a step or an impulse response is used, the Bode diagram can be constructed from the response data by using numerical Fourier transforms. There are some other graphic methods which can determine the parameters of a simplified model from the step or impulse response data directly (6, 7).

(b) Correlation method

In this method the input is restricted to a special type of signal such as white noise or pseudo-random binary

sequence (PRBS). Because the auto-correlation integral of a white noise input is a delta function, a simple relation between the impulse transfer function and the cross-correlation function of the input and the output can be derived. A Bode diagram is then constructed from the input-output data by using numerical Fourier transforms.

(c) Linear regression method

If a linear equation with unknown parameters is used to describe a system, the equation must satisfy the input-output data at each sampling time. Therefore, a set of linear equations can be derived for a set of response data. When the number of data is larger than the number of parameters, the unknown parameters can be determined by solving the set of overdetermined linear equations with the least-squares method.

A sequential regression method can be derived from the linear regression method. In the sequential regression method, the parameters of a model is formulated in a sequential expression. Therefore, the parameters can be renewed at each sampling time.

(d) Quasilinearization method

In this method a problem is linearized by using a first order Taylor series expression. The solution of the linearized problem is then expressed in terms of two functions which can be obtained easily by solving a set of

linear initial condition problems.

(e) Heuristic identification method

In this method, a model with adjustable parameters is constructed from knowledge of the system. The difference between the response of the model and the experimental response data is defined as the objective function to be minimized, and the parameters of the model are the independent variables of the objective function. A nonlinear optimization method will follow a strategy to search the parameters of the model to minimize the objective function value. The procedures are illustrated in Figure 3. Nonlinear optimization techniques will be discussed in the next section.

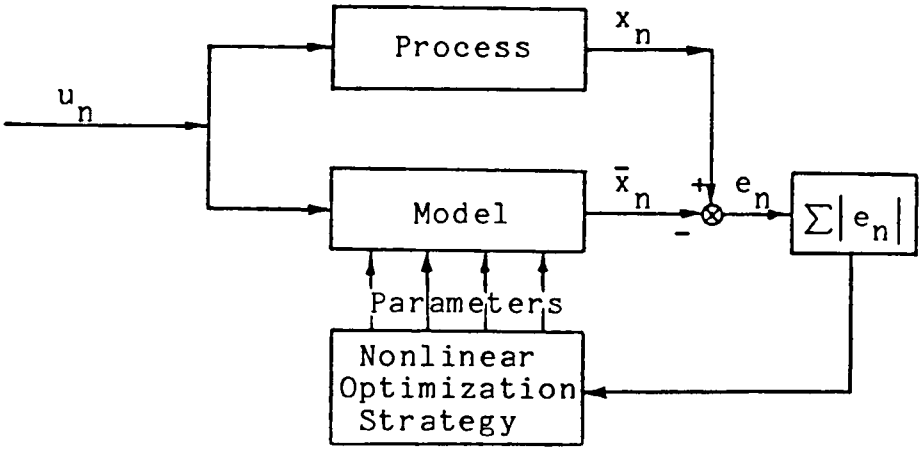


Figure 3. System Identification by Using a Nonlinear Optimization Method.

In the method (a) and (b), the model is derived from a Bode diagram. The advantage of these two methods is their capability of determining the order of a system. The disadvantages are long experimental time, unacceptable form of the input signal in some cases, and high sensitivity to noise. Although PRBS can deal with uncorrelated noise, PRBS may be too fast and/or weak for some slow processes. Therefore, these two methods are good for preliminary study of a system. For more accurate estimates of the model parameters, some other methods are recommended. In the method (c) and (d) the model has to be linear or approximated by a linear model. This involves mathematical manipulations, and sometimes linearization is not desirable. As for the method (e), any form of model can be adapted easily. Therefore, the heuristic method is more attractive for general uses.

C. Nonlinear Programming

By a characteristic of nonlinear optimization techniques, the techniques can be divided into two categories, i.e., methods using derivatives of the objective function and derivative-free methods.

(a) Methods using derivatives

For years most emphasis has been given to methods using derivatives due to the fact that most of these methods have the property of quadratic convergence. It can be proven

theoretically that when those methods are applied to a quadratic function of n variables, the minimum of the function can be reached in n steps of search. Those methods have also been applied to general cases quite well because in practical applications many objective functions are close to a quadratic function. However, two conditions are required to guarantee successful results. First, the analytical derivatives of the objective function are available, or they can be reasonably substituted by difference schemes. The second condition is that there are no strong interactions among the independent variables of the objective function.

The iteration of these methods has the general form

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda^{(k)} \bar{s}^{(k)} \tag{4}$$

where $\bar{x}^{(k)}$ is the point reached at the kth stage of iteration,

$\lambda^{(k)}$ is a scalar, and

$\bar{s}^{(k)}$ is the direction of search and is related to the derivative of the objective function $f(\bar{x})$ at point $\bar{x}^{(k)}$.

(i) Gradient (Steepest descent) method

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda^{(k)} \bar{\nabla} f(\bar{x}^{(k)}) \tag{5}$$

where

$$\bar{\nabla} f(\bar{x}^{(k)}) = \begin{bmatrix} \frac{\partial f(\bar{x}^{(k)})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\bar{x}^{(k)})}{\partial x_n} \end{bmatrix}$$

Either a fixed or adjustable value of the scalar $\lambda^{(k)}$ can be used in this method.

(ii) Newton's method

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda^{(k)} \left[\bar{\bar{\nabla}}^2 f(\bar{x}^{(k)}) \right]^{-1} \bar{\nabla} f(\bar{x}^{(k)}) \qquad (6)$$

where

$$\bar{\bar{\nabla}}^2 f(\bar{x}^{(k)}) = \begin{bmatrix} \frac{\partial^2 f(\bar{x}^{(k)})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\bar{x}^{(k)})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\bar{x}^{(k)})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\bar{x}^{(k)})}{\partial x_n^2} \end{bmatrix}$$

The value of $\lambda^{(k)}$ can be unit or a value which will minimize the objective function in the direction of $\bar{s}^{(k)}$. In the latter case, a unidimensional search is performed in the direction of $\bar{s}^{(k)}$.

(iii) Conjugate direction method

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} + \lambda^{(k)} \left[- \bar{\nabla} f(\bar{x}^{(k)}) + \bar{s}^{(k-1)} \frac{\bar{\nabla}^T f(\bar{x}^{(k)}) \, \bar{\nabla} f(\bar{x}^{(k)})}{\bar{\nabla}^T f(\bar{x}^{(k-1)}) \, \bar{\nabla} f(\bar{x}^{(k-1)})} \right] \qquad (7)$$

The value of $\lambda^{(k)}$ is chosen such that the objective function has the minimun value in the direction of $\bar{s}^{(k)}$, and $\bar{s}^{(0)} = -\bar{\nabla} f(\bar{x}^{(0)})$. One of this type method has been proposed by Fletcher and Reeves (8).

(iv) Variable metric method

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda^{(k)} \bar{\bar{\eta}}(\bar{x}^{(k)}) \bar{\nabla} f(\bar{x}^{(k)}) \qquad (8)$$

where $\bar{\bar{\eta}}(\bar{x}^{(k)})$ is the so called direction matrix, and it is just an approximation of $[\bar{\bar{\nabla}}^2 f(\bar{x}^{(k)})]^{-1}$ in the Newton's method. The direction matrix can be calculated from several different schemes by using the derivatives. Quite a lot of methods (9, 10) fall in this category.

(b) Method without using derivative

There are some derivative-free methods retaining the property of quadratic convergence by searching along a set of conjugate or orthogonal directions (11, 12). The other methods determine search directions by a certain pattern (1, 13).

(i) Rosenbrock's method

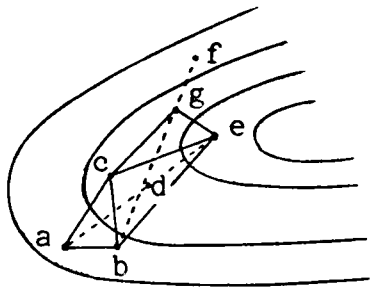
In the initial stage, the search starts with successive unidimensional searches in the coordinate axis directions. Then a set of orthogonal search directions is constructed by the Gram-Schmidt procedure. In each stage a new set of orthogonal search directions is generated from the search results. Generally one of the orthogonal directions is chosen to be the direction of the best improvement.

(ii) Powell's mehtod

In this method a set of conjugate search direction is constructed by using the following theorem. If, starting at $\bar{x}^{(0)}$, the point $\bar{x}^{(a)}$ is located in the direction $\bar{s}$ at the minimum of $f(\bar{x})$, and if, starting from the point $\bar{x}^{(1)} \neq \bar{x}^{(0)}$, the point $\bar{x}^{(b)}$ is located in the same direction $\bar{s}$ at the minimum of $f(\bar{x})$, then if $f(\bar{x}^{(b)}) < f(\bar{x}^{(a)})$, the direction $(\bar{x}^{(b)} - \bar{x}^{(a)})$ is conjugate to $\bar{s}$. In the initial stage, the coordinate axis directions are chosen as the search directions. According to the theorem, a new conjugate direction can be generated after each stage of unidimensional search. Therefore, a quadratic function of n variables can be minimized after n conjugate directions being generated.

(iii) Flexible polyhedron search

In this method the objective function values are compared at the vertices of a polyhedron. A new polyhedron is constructed by discarding the worst point after reflecting the worst point through the centroid of the other vertices. The distance of reflection can be extended (expension of the polyhedron) or shortened (contraction of the polyhedron) depending on the results of the reflection. If the reflection does not reach any point better than the worst point. The size of the polyhedron will be reduced. The procedures are illustrated in Figure 4.



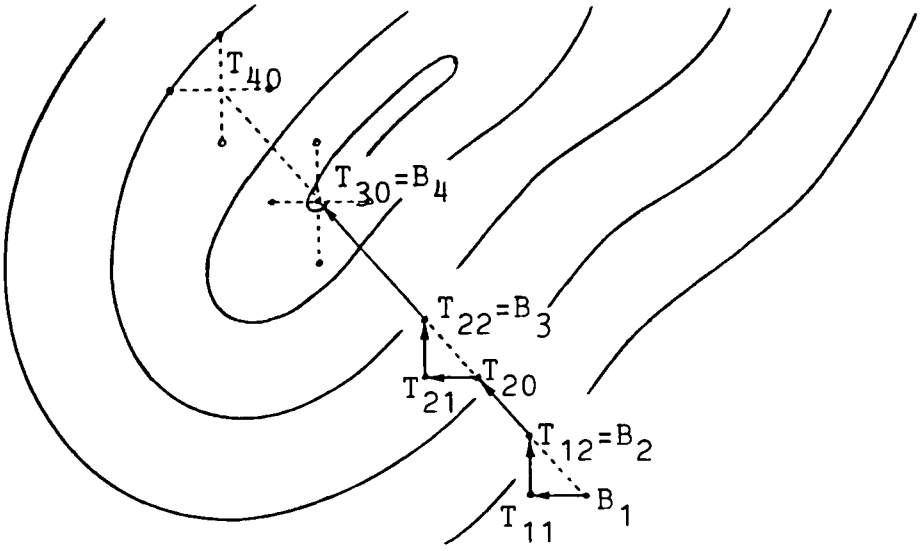
$\triangle abc$: a $\xrightarrow{\text{Reflection}}$ d $\xrightarrow{\text{Expansion}}$ e

$\triangle bce$: b $\xrightarrow{\text{Reflection}}$ f $\xrightarrow{\text{Contraction}}$ g

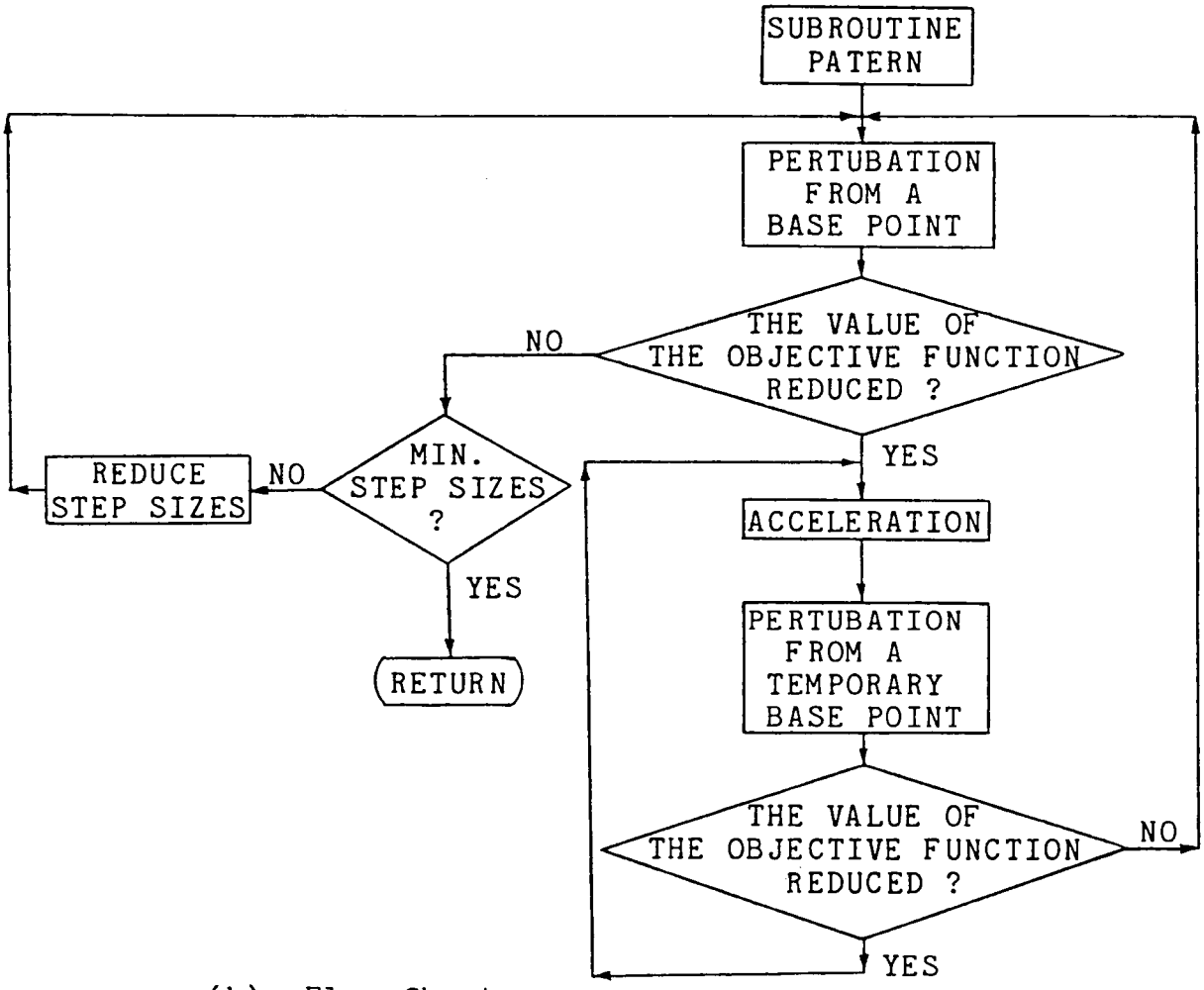
Figure 4. Flexible Polyhedron Search.

(iv) Pattern search

The pattern search method of Hooke and Jeeves (1) has been used in experimental modeling. The pattern search consists of iterations of two types of search, i.e., the perturbation and the acceleration search. A two-dimension pattern search and the flow chart of the pattern search are shown in Figure 5. The search starts from an arbitrary base point, B, and a step size is chosen for each variable. In the perturbation stage, the variables are changed by the prescribed step size one by one. The new searched point which reduces the value of the objective function is accepted as the temporary base point, T. The exploration of the next variable will than start from the temporary base point. After each variable being explored, the acceleration stage starts. In this stage the search advances a distance of $|\vec{BT}|$ in the direction of $\vec{BT}$. Then the temporary base



(a) Two-dimension Pattern Search



(b) Flow Chart

Figure 5. Strategy of the Pattern Search.

point becomes the new base point, and the advanced point becomes the new temporary base point from which a new perturbation stage will start again. A new acceleration search will be performed if the objective function has a lower value at the temporary base point than at the base point after the perturbation stage. Otherwise the temporary base point will retreat to the base point, and the search will start from the perturbation stage again. If still no point better than the base point can be found, all the step sizes are reduced and all the search procedures will be repeated again until the step sizes are reduced to the prescribed minimal step sizes.

CHAPTER III

STATEMENT OF THE PROBLEM

A heuristic identification method has been used to identify the parameters of a second order plus dead time model from a set of step response data. The pattern search of Hooke and Jeeves (1) was used to search the optimal parameters. The pattern search was selected as the optimization technique because its idea is simple and easily understood. In addition to the reason of simplicity, the pattern search can be applied to either linear or nonlinear objective functions with constraints. Therefore, the pattern search is most commonly used for these applications. The structure of the computer programs is shown in Figure 6. The step response data were simulated with a second order plus dead time system on a digital computer. Therefore, the optimal model parameters should differ the system parameters by not more than the minimal step size used in the pattern search. However, for different initial guesses of the model parameters inconsistent model parameters (TABLE I and TABLE II) were given by the pattern search used in the modeling of the following two systems.

$$G_p(s) = \frac{K\, e^{-\theta s}}{\tau^2 s^2 + 2\tau\zeta s + 1} \qquad (9)$$

System I : $K = 1.5$, $\tau = 3.0$, $\zeta = 1.4$, $\theta = 2.3$

System II : $K = 1.5$, $\tau = 8.0$, $\zeta = 1.4$, $\theta = 2.3$

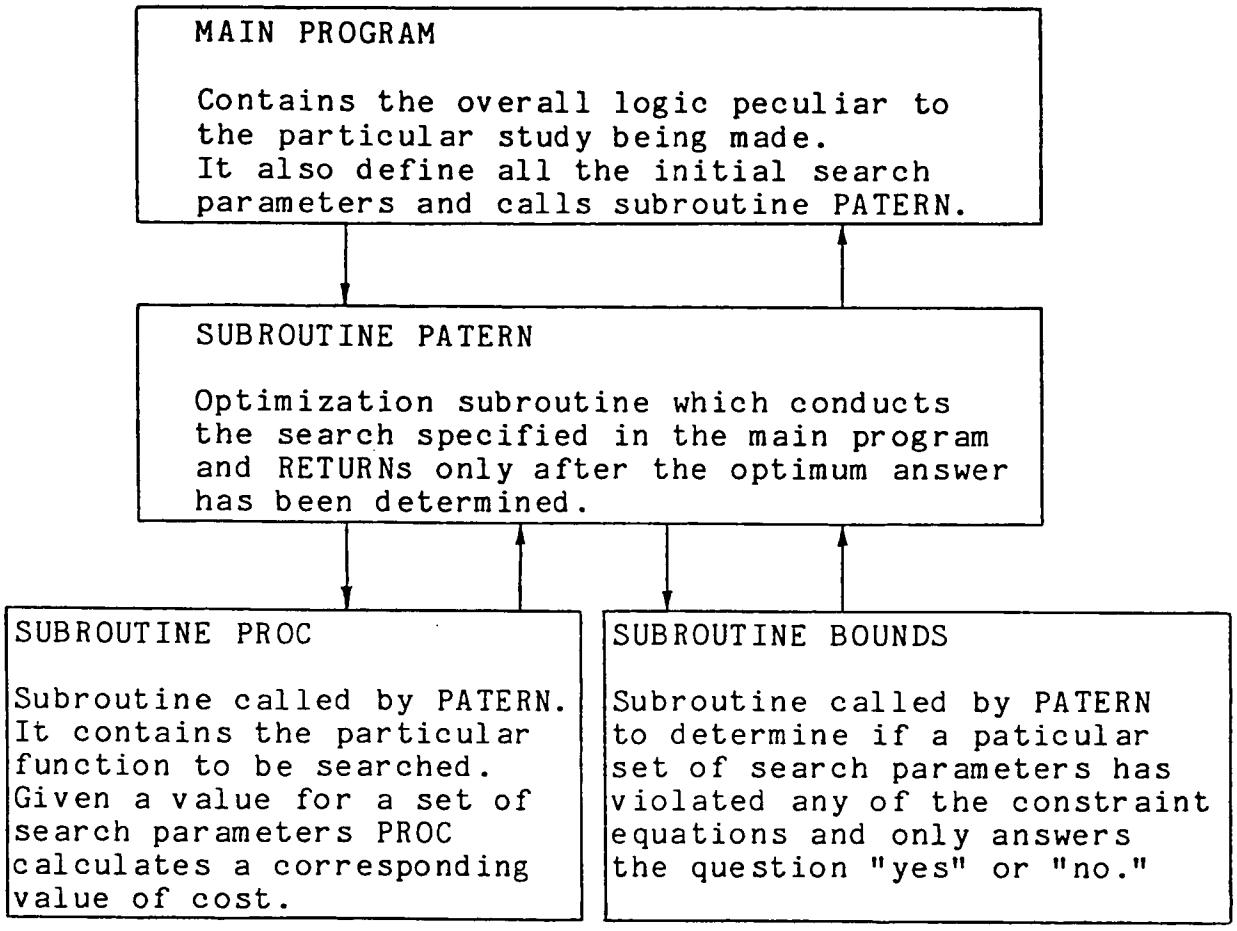


Figure 6. Program Structure of the Pattern Search Method.

TABLE I. Step Response Data Modeling of System I by Using the Pattern Search Method

Model or System	Initial Guesses				Final Results					
	K	τ	ζ	θ	K	τ	ζ	θ	$\tau\zeta$	$\tau+\theta$
Model 1	1.50	1.50	3.00	5.00	1.500	1.314	2.838	3.305	3.729	4.619
Model 2	3.00	1.00	3.00	1.00	1.500	1.498	2.510	3.247	3.759	4.745
Model 3	1.00	2.00	2.50	5.00	1.500	1.506	2.499	3.240	3.763	4.746
Model 4	1.40	2.40	2.10	5.00	1.500	1.884	2.046	3.042	3.855	4.926
Model 5	1.40	2.40	2.50	5.00	1.500	2.110	1.850	2.956	3.904	5.066
System I					1.500	3.000	1.400	2.300	4.200	5.300

Initial Step Size : 0.1

Final Step Size : 10^{-7}

Number of Data : 60

TABLE II. Step Response Data Modeling of System II by Using the Pattern Search Method

Model or System	Initial Guesses				Final Results					
	K	τ	ζ	θ	K	τ	ζ	θ	$\tau\zeta$	$\tau+\theta$
Model 1	2.50	5.00	3.00	5.00	1.501	4.212	2.396	4.726	10.09	8.938
Model 2	1.53	5.42	1.93	4.30	1.501	5.901	1.781	3.844	10.51	9.745
Model 3	1.00	12.0	1.00	6.00	1.500	7.992	1.401	2.304	11.19	10.296
Model 4	1.60	12.0	0.80	0.00	1.500	8.525	1.335	1.872	11.38	10.397
Model 5	1.65	10.0	1.18	0.40	1.499	9.090	1.276	1.368	11.59	10.458
System II					1.500	8.000	1.400	2.300	11.20	10.300

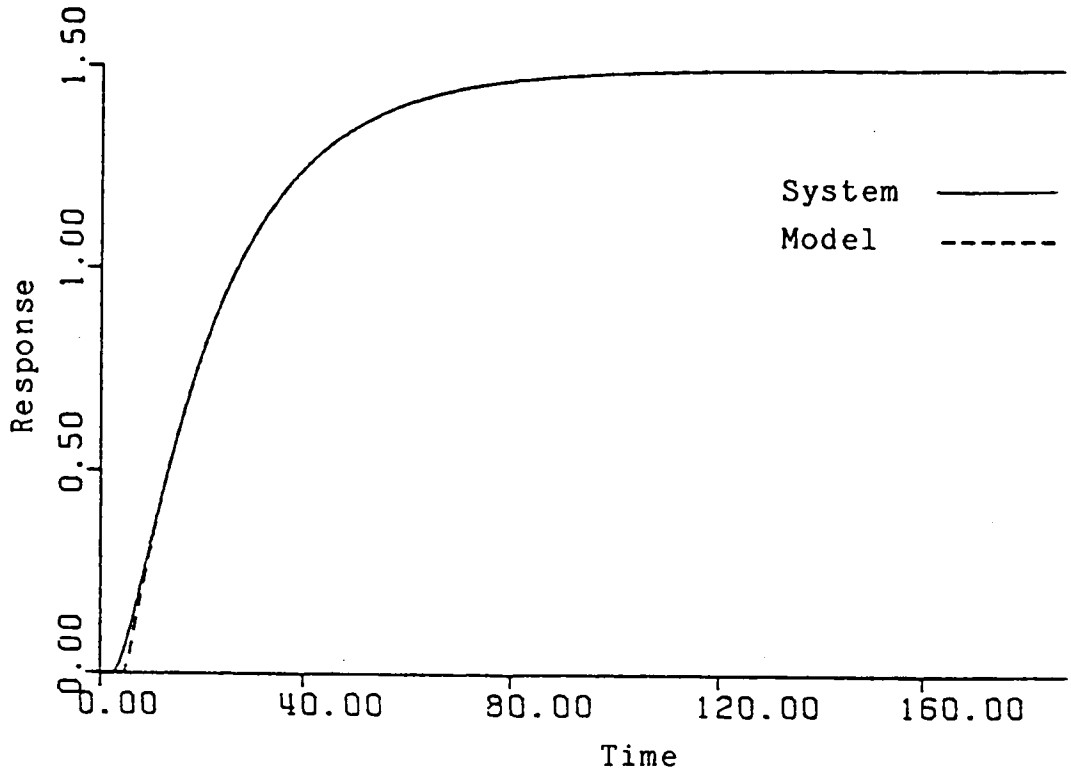
Initial Step Size : 0.1

Final Step Size : 10^{-7}

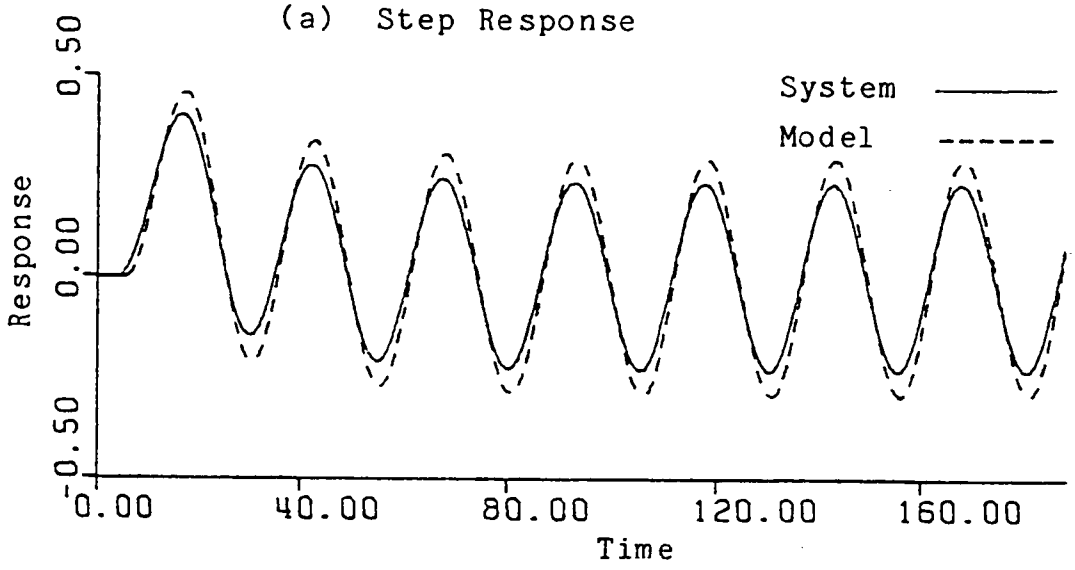
Number of Data : 120

For simplicity the sampling period is taken as one unit of time in this thesis unless otherwise noted. In Figure 7, the response of Model 1 in TABLE II was compared with the response of the system generating the response data. For a step input the response of the model was very close to the response of the system. However, for some other inputs, e.g., a sinusoidal input, the response of the model became more different from the response of the system. Apparently, the pattern search did not reach the global optimum. There are two cases when the search will be terminated before reaching the global optimum. One case is when the search reaches a local optimum. Since different starting points were used, and most models had quite different values of parameters, termination at a local optimum was less likely the case. Another possibility was that the search stalled on a ridge. In Figure 5(a) (page 16), it shows that when the pattern search reaches the point B_4, the only way to make progress is to reduce the step size. If the ridge is very sharp, the step has to be reduced to a very small size. However, the step size is limited by the roundoff in adding or subtracting in a digital computer. Also a step size which can not cause the objective function to change by an amount larger than the error of the objective function is meaningless.

In Figure 8, the contour surface of the step response modeling objective function of System I was projected on

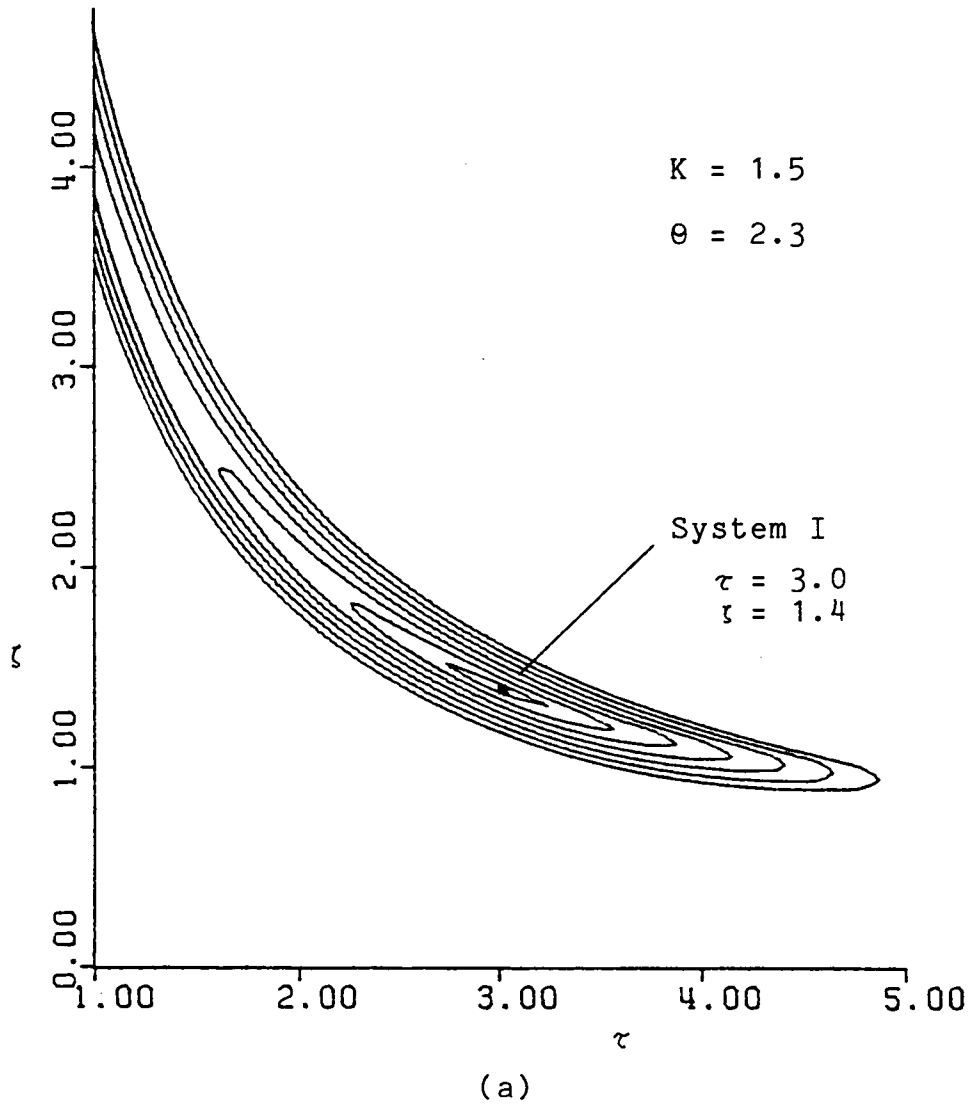


(a) Step Response



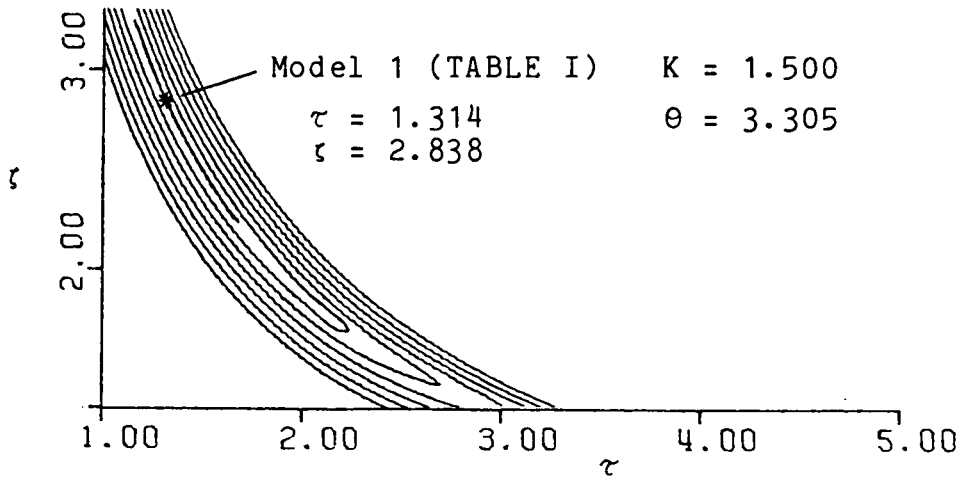
(b) Sinusoidal Response

Figure 7. Responses of System II and Model 1 in TABLE II.

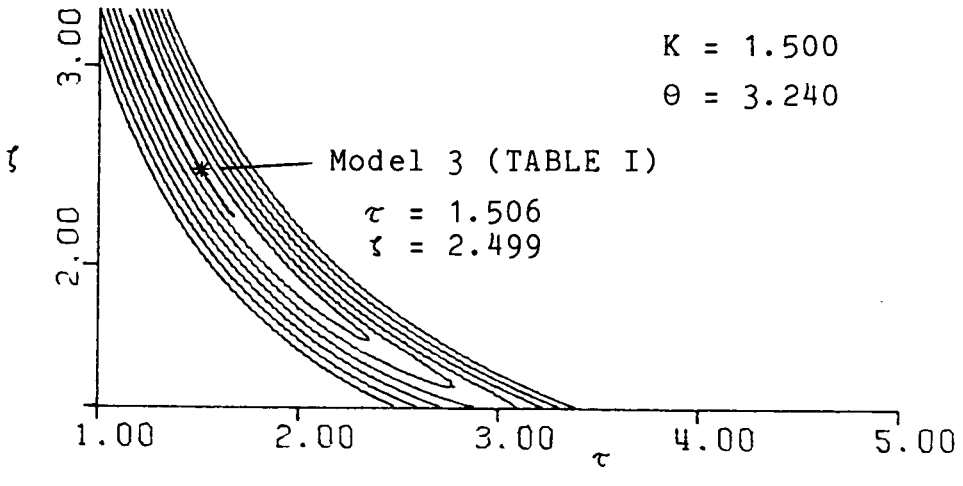


(a)

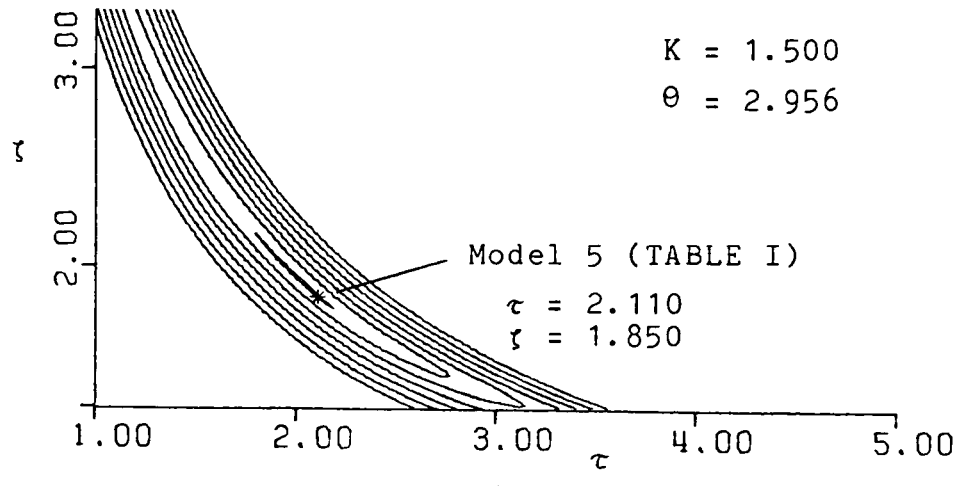
Figure 8. Projections of the Step Response Modeling Objective Function of System I.



(b)

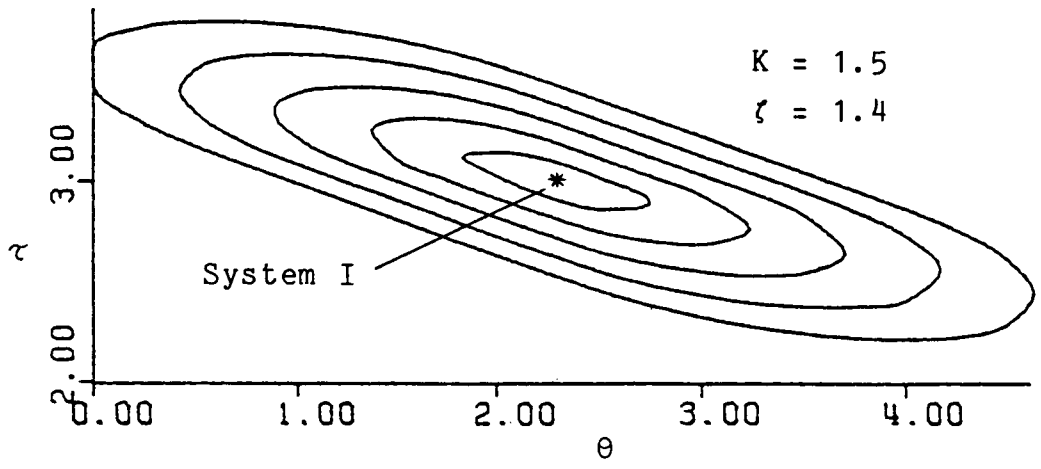


(c)

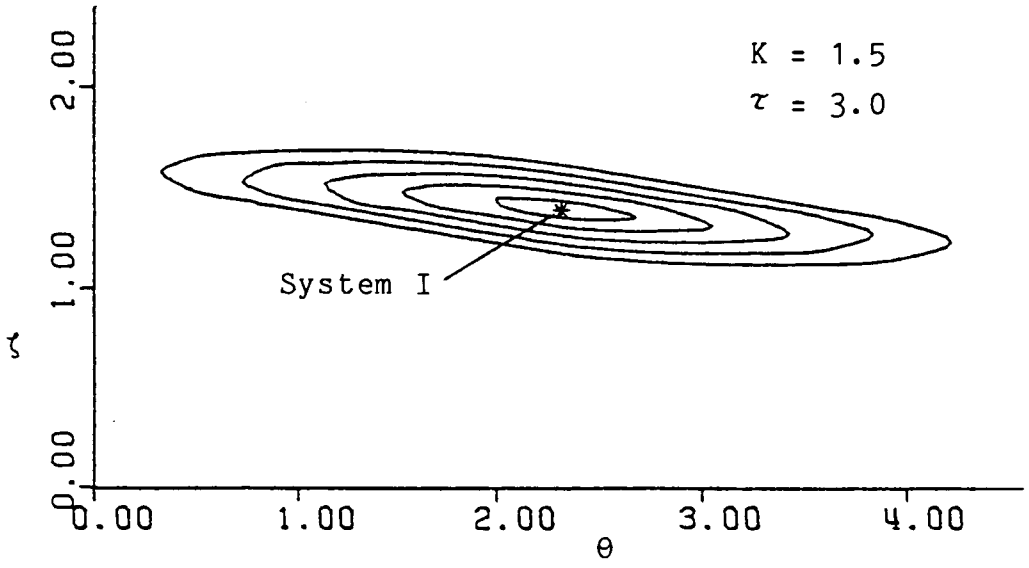


(d)

Figure 8. (Continued)



(e)



(f)

Figure 8. (Continued)

several parameter planes. A curved ridge which lay on the line $\tau\zeta \cong$ constant appeared on the τ-ζ plane. This explained the results that $\tau\zeta \cong$ constant in TABLE I and TABLE II (pages 20 and 21). Ridges also appeared on the projections of the other two parameter planes. Because these ridges were not parallel to the coordinate axes, they all might cause the pattern search to stop.

Although the optimization techniques which use conjugate search directions give successful results when they are applied to a quadratic function, they often fail when a narrow curved ridge is encountered. The main reason is that the conjugate directions do not necessarily lie along the ridge of a nonquadratic function. One of the methods which can follow a curved ridge is the flexible polyhedron method of Nelder and Mead (13). In that method only those vertexes that yield high value of the objective function will be rejected during the generation of a new polyhedron. Therefore, if one of the vertexes is on a very narrow ridge, unless the new vertex also hits right on the ridge, the polyhedron will be stuck on the ridge at that point. To follow a very narrow ridge, an optimization method should have a strategy which can directly detect the direction of the ridge.

Another reason that ridges are not desirable is narrow ridges usually imply poor quality (lack of certain information) of the experimental data. As the parameters

change along the direction of a ridge the cost of the corresponding model does not change very much. In other words the parameters are not clearly defined by the data in a certain way. Therefore, we should avoid the appearance of ridges in the first place.

CHAPTER IV

EXPERIMENTAL MODELING BY USING A NONLINEAR REGRESSION METHOD

In the heuristic modeling technique a nonlinear regression method is used to search the model parameters. Therefore, there is no restrictions on the form of model. In addition to this advantage, this technique does not involve mathematical manipulations. However, the nonlinear regression method may stall on ridges on the contour surface of the objective function to be minimized. To make the results more reliable, the ridge should be eliminated, or the regression method should be designed to deal with the ridge problem.

A. Coordinate Transformation

Since search directions in the pattern search method are parallel to the coordinate axes, an approach to solve the ridge problem is to transform the coordinate system such that the ridge becomes parallel to one of the coordinate axes. This idea can be illustrated with a quadratic objective function of the form

$$G(x,y) = A\,x^2 + B\,xy + C\,y^2 + D\,x + E\,y + F \qquad (10)$$

where x and y are the search parameters.

The contour map of G(x,y) consists of a family of

elipses, and a ridge lies on the major axis of the elipses. If $B \neq 0$, the ridge will not be parallel to either of the coordinate axes. When pattern search reaches such an inclined ridge as shown in Figure 9(a), the search will stall if the ridge is very narrow or the step sizes can not be further reduced. A solution is to remove the interaction between the independent variables by changing the variables. Let

$$x = \bar{x} \cos \phi - \bar{y} \sin \phi \tag{11}$$

$$y = \bar{x} \sin \phi + \bar{y} \cos \phi \tag{12}$$

where $\phi = (1/2) \tan^{-1}[B/(A-C)]$. The quadratic function becomes

$$\bar{G}(\bar{x},\bar{y}) = \bar{A}\, \bar{x}^2 + \bar{C}\, \bar{y}^2 + \bar{D}\, \bar{x} + \bar{E}\, \bar{y} + \bar{F} \tag{13}$$

where $\bar{x}$ and $\bar{y}$ are the new search parameters.

After the transformation, the ridge is parallel to one of the coordinate axes as shown in Figure 9(b), and the pattern search can now proceed in the direction of the ridge. The basic change by using the new search parameters is removal of the cross product term of the search parameters in equation (10).

The ridge problem in modeling can be solved in a similar way. For all second order systems there is an equivalent system as shown in Figure 10. The system can be

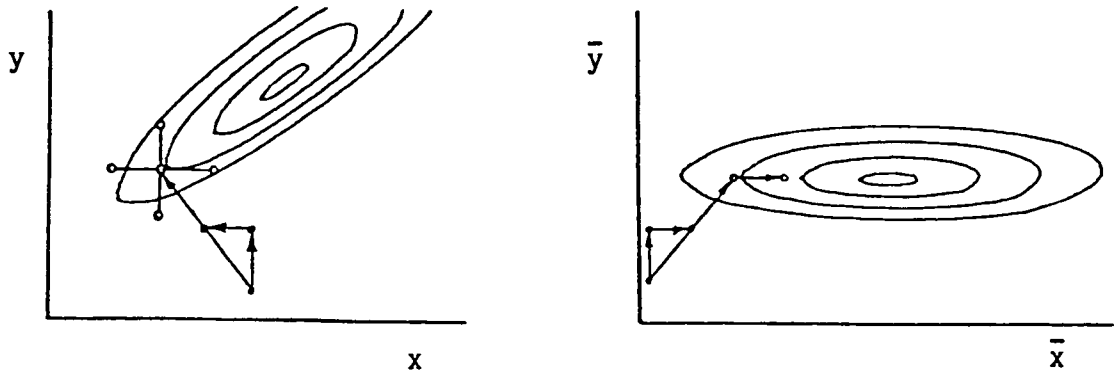


Figure 9. Elimination of an Inclined Ridge by Transformation of Coordinate.

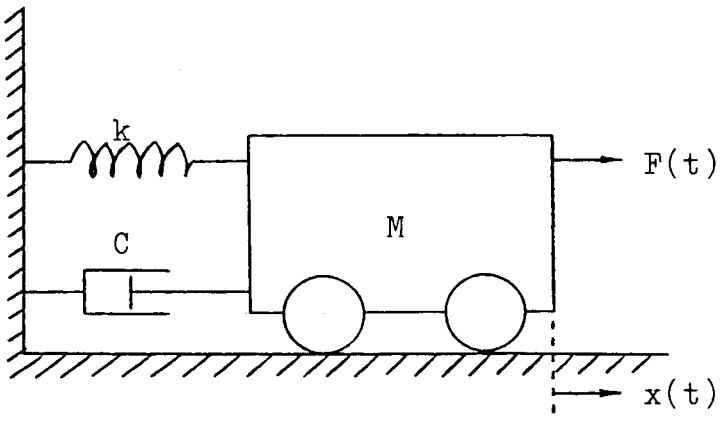


Figure 10. A Second Order System.

described by a second order differential equation:

$$M \frac{d^2x(t)}{dt^2} + C \frac{dx(t)}{dt} + k\, x(t) = F(t) \qquad (14)$$

or $$\frac{M}{k} \frac{d^2x(t)}{dt^2} + \frac{C}{k} \frac{dx(t)}{dt} + x(t) = \frac{1}{k} F(t) \qquad (15)$$

or in terms of the parameters used in the model

$$\tau^2 \frac{d^2x(t)}{dt^2} + 2\tau\zeta \frac{dx(t)}{dt} + x(t) = K\, F(t) \qquad (16)$$

where $K = 1/k$, $\tau = \sqrt{M/k}$, and $\zeta = \sqrt{C^2/4Mk}$.

In Figure 11, a plot of F(t), x(t), dx(t)/dt and $d^2x(t)/dt^2$ of the system step response shows that for most period of time the value of $d^2x(t)/dt^2$ is nearly equal to zero. This makes the first terms of these equations lightly weighted in a step response. Therefore, models with $K \cong 1/k$ and $2\tau\zeta \cong C/k$ will have similar step responses even if these models have different values of τ. When we try to identify a system from a set of step response data, if a large value of ζ is used in the model, the value of τ must be small to satisfy the relation $2\tau\zeta \cong C/k$. On the contour surface this interaction caused the curved ridge in Figure 8(a) (page 24).

If we let $\tau = P_1$, $\zeta = P_2/\tau$, and $K = P_3$,

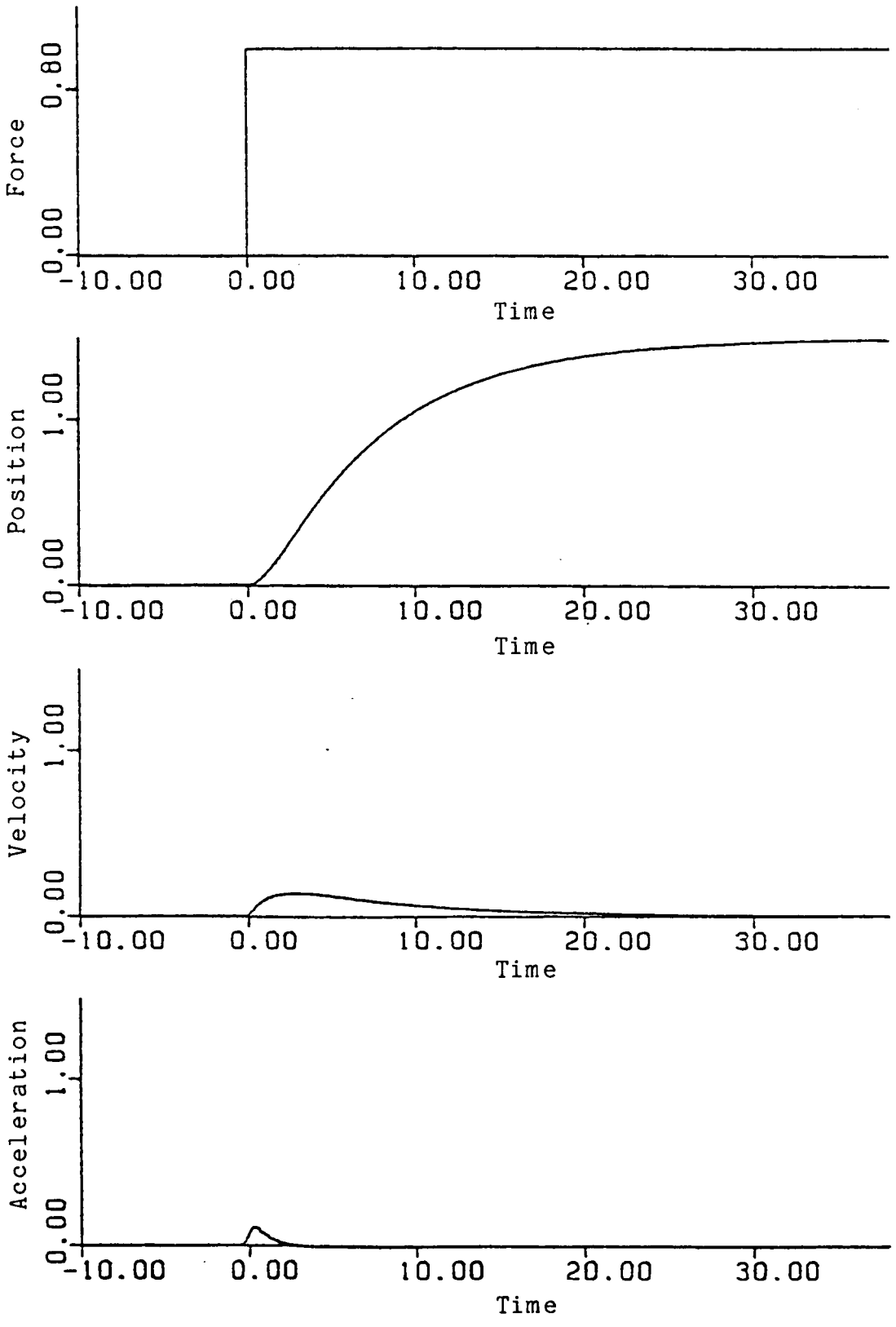


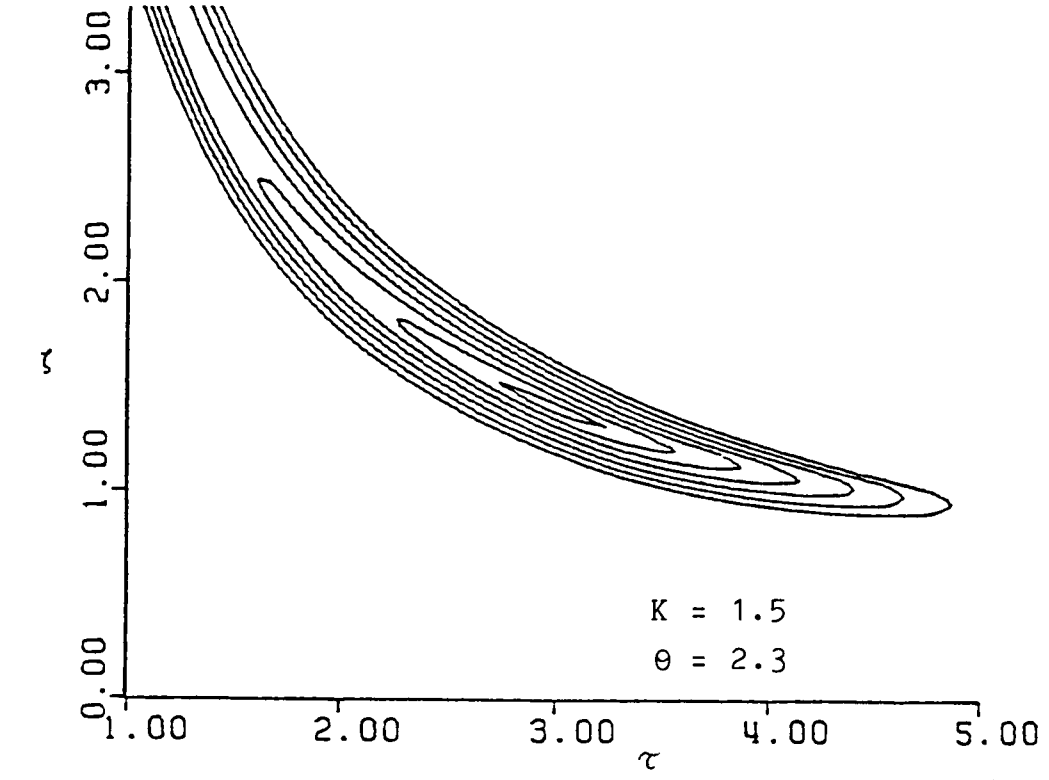
Figure 11. Unit Step Response of a Second Order System.

equation (16) becomes

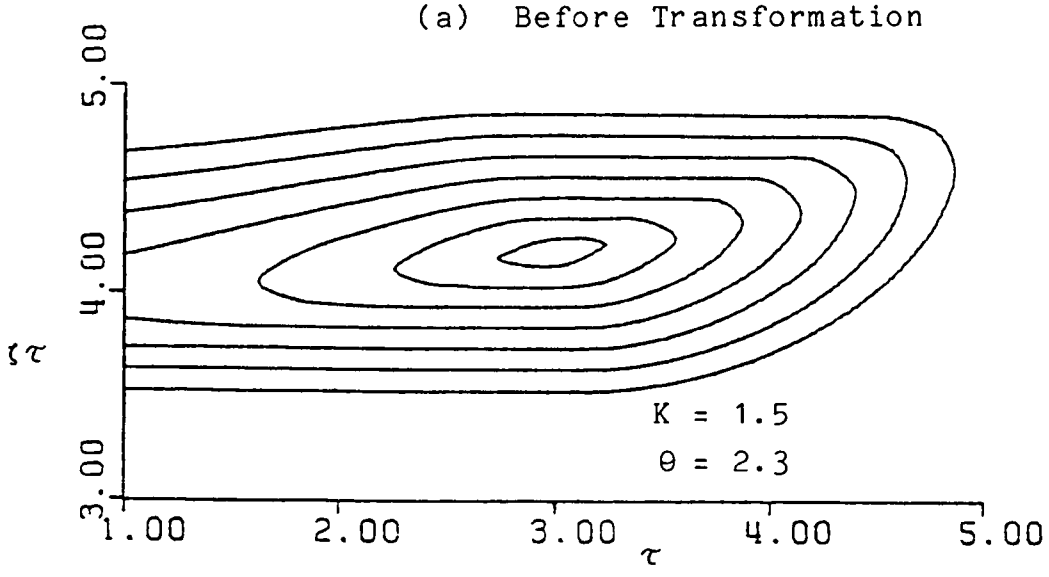
$$P_1 \frac{d^2x(t)}{dt^2} + 2 P_2 \frac{dx(t)}{dt} + x(t) = P_3 F(t) \qquad (17)$$

In equation (17) P_1, P_2, and P_3 specify the characteristics (accleration, velocity, and steady state gain) of the system independently. Therefore, the interaction among the parameters of the equation is eliminated. In Figure 12, the projection of the contour surface on the P_1-P_2 plane shows that the curved ridge caused by the interaction of τ and ζ was transformed into a straight ridge which was parallel to one of the coordinate axes.

So far the dead time of the system has not been taken into consideration. The step response curve of a second or higher order system is a S-shaped curved. The response curve rises more slowly for a system with large τ. The slugishly rising curve is very similar to the response curve of a system with large dead time. Therefore, when we try to identify a model from step response data, if a large dead time is used to account for the retardation of the response, the model would have smaller value of τ. The results in TABLE I and TABLE II (pages 20 and 21) show the interaction that $\tau + \theta \equiv$ constant. The situation is very similar to the interaction between τ and ζ. As a matter of fact, Harrison (14) tried to use the dead time in a model to



(a) Before Transformation



(b) After Transformation

Figure 12. Elimination of a Curved Ridge in Step Response Modeling by Coordinate Transformation.

replace some time constants of a high order system. However, it is not clear how dead time interacts with τ, and the degree of interation depends on the type of input signal. Interation also exists between ζ and dead time. Fortunately, the ridges caused by these interactions are not as serious as that caused by the interaction between τ and ζ.

In Chapter II difference equations were derived for discrete systems. Because of simplicity, difference equations rather than differential equations have been used to represent system by most people. Therefore, using the coefficients of the difference equation as the search parameters is very attractive. However, in Figure 13, the projections of the contour surface show that there were strong interactions among the coefficients of the difference equation. The interactions were caused by the fact that the coefficients a_1 and a_2 in equation (2) all related the current response to the previous responses, which were separated by only a sampling period. Therefore, a_1 and a_2 had very similar effect on the characteristics of the system behavior. The same situation existed among the coefficients b_1, b_2, and b_3. These interactions can be described by $P_i + P_j$ = constant. This type of interaction usually forms nearly 45° ridges on the contour surface as shown in Figure 13. These 45° ridges can be transformed into ridges parallel to one of the coordinate axes by using P_i and

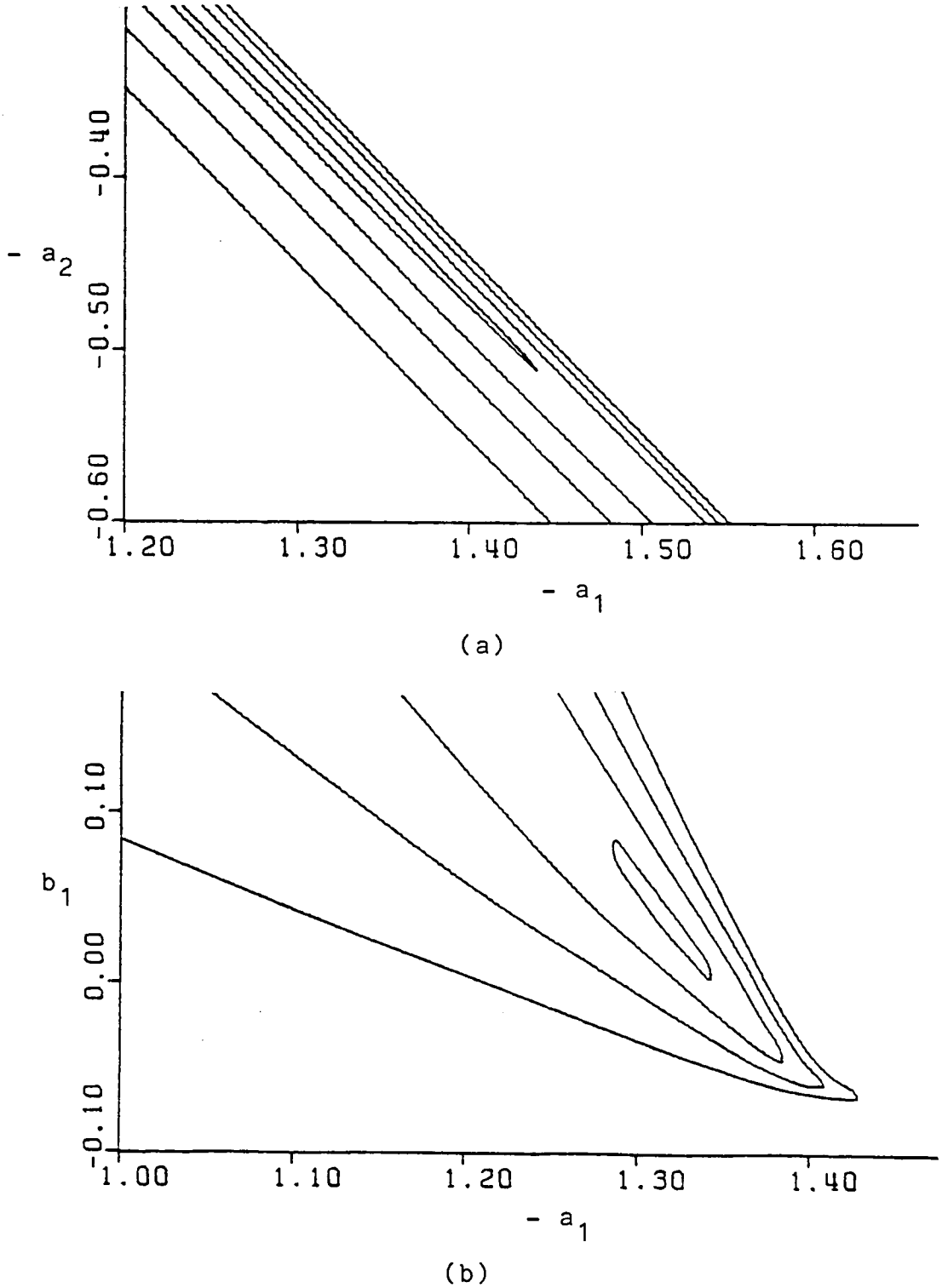
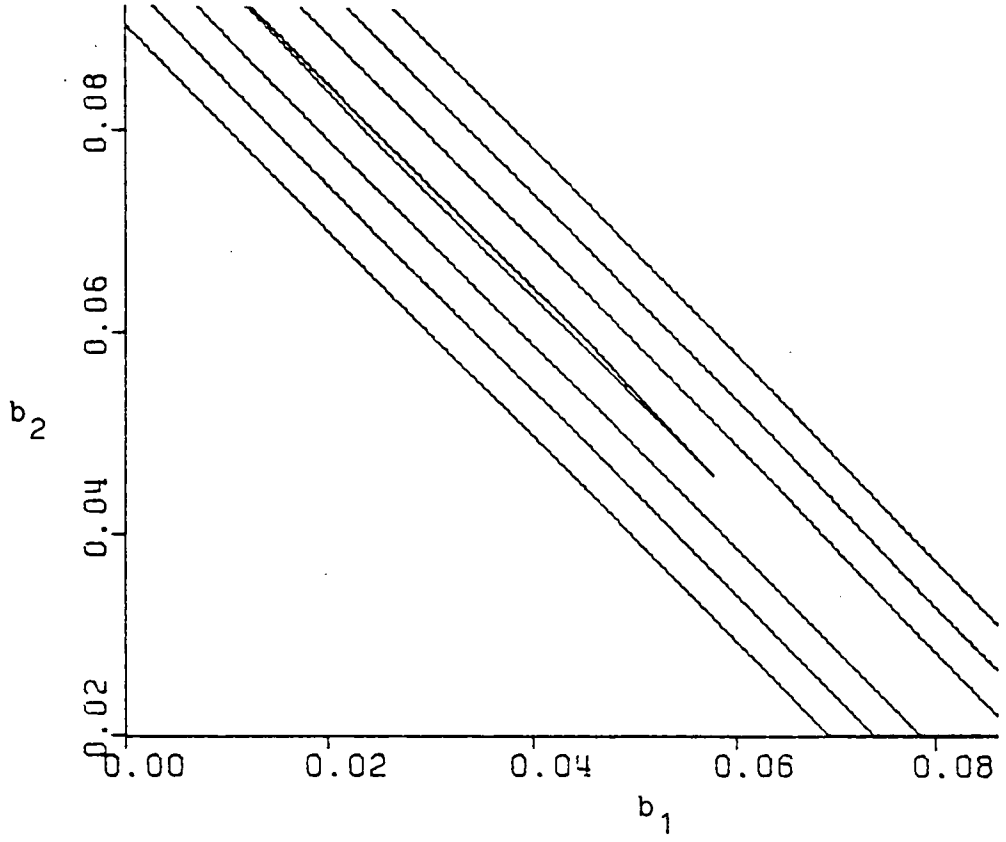
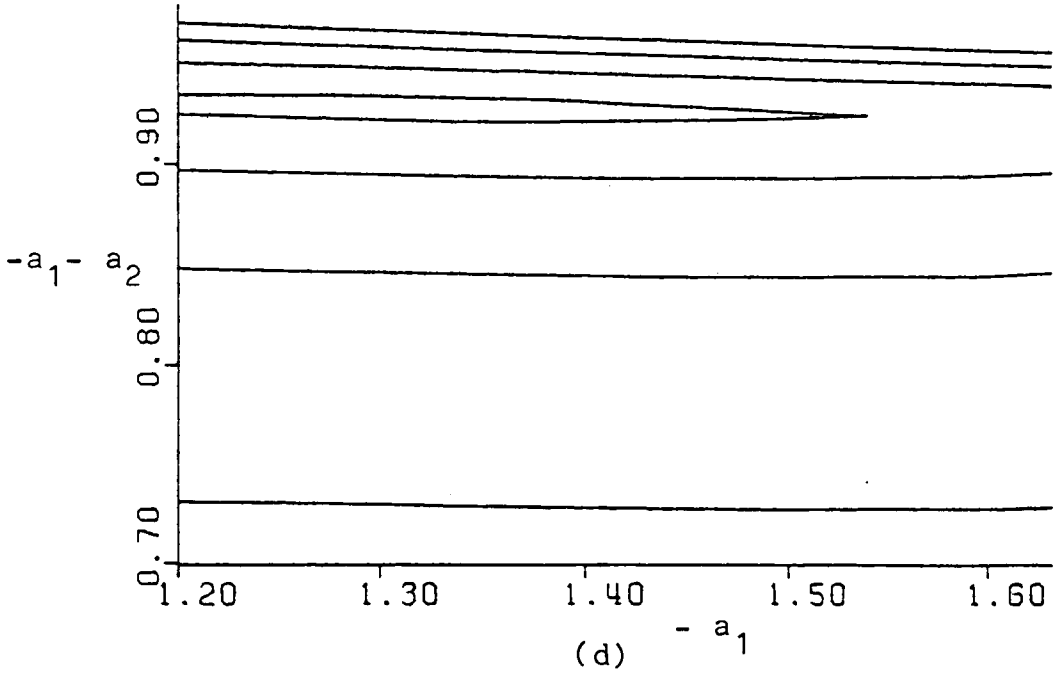


Figure 13. Projections of the Step Response Modeling Objective Function by Using Difference Equation Coefficients as Search Parameters.



(c)



(d)

Figure 13. (Continued)

$P_i + P_j$ as the search parameters. A projection of the transformed contour surface is shown in Figure 13(d).

It has been shown that whenever two parameters of a model have similar effect on the behavior of the model, ridges would appear on the contour surface and identifing these parameters from system response data would be difficult. A general rule is to choose parameters which independently specify the characteristics of the system behavior. Unfortunately this requires a thorough knowledge of the relation between system and parameters. Sometimes, it is impossible or undesired to do so. Then a more powerful optimization technique is required.

B. Design of Experiment

Basically all techniques in experimental modeling are only different means to extract information from experimental data. Therefore, the quality of experimental data is a major factor which decides the accuracy of the model.

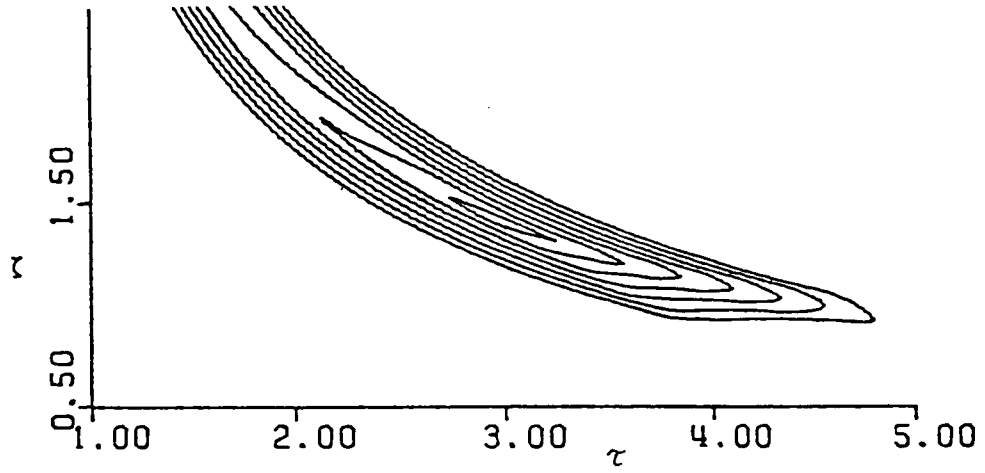
Although sometimes the ridge problem can be solved by transformation of the contour surface, the surface may still not be ideal for search. For example, in Figure 13(d), the cost is very insensitive to the parameter a_1. Usually narrow ridges implies poor quality of the data, and noise in the experimental data could make the situation even worse.

The quality of data can be improved by proper design of

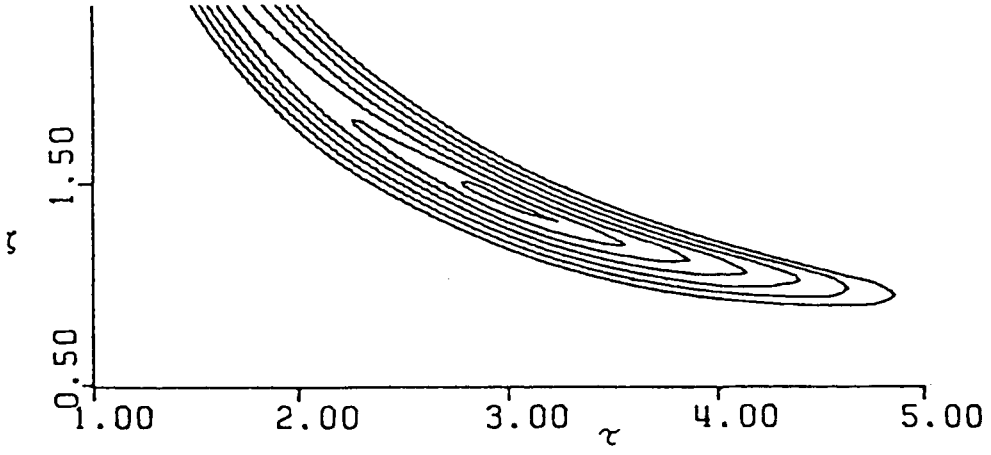
experiment. Two major areas concerned with the quality of experimental data were investigated.

(a) Acquisition of experimental data

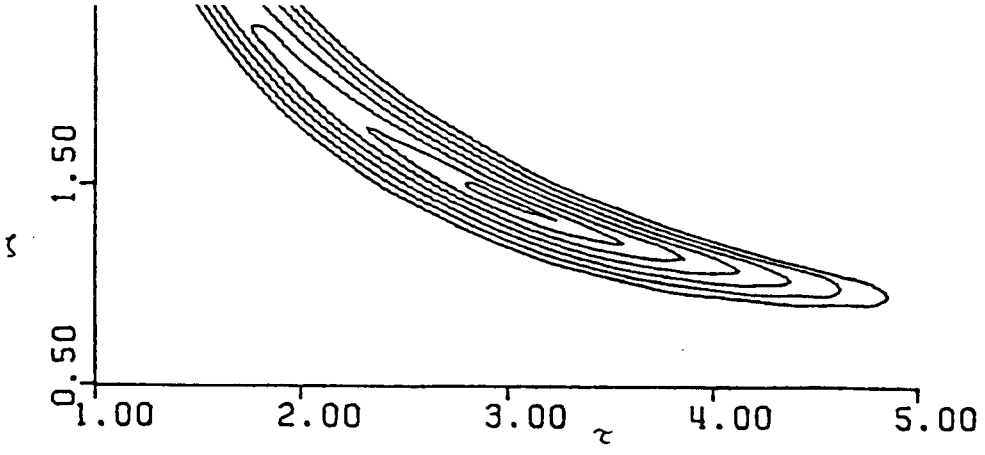
The response of a discrete system is generally recorded through a sampling process. According to the sampling theorem the highest frequency of the information which can be reconstructed from the sampled data is one half the sampling frequency. This is mainly because the high frequency information is overlapped by the complementary components in the sampled data. Since response of most systems are not band-limited, theoretically, it requires infinite sampling frequency to recover all information of a system response. However, a low sampling frequency can often be used because the amplitude ratio of most systems decreases sharply with frequency, and what we would loss by using low sampling frequency is very weak high frequency information in the original response. Usually sampling period equal to the smallest time constant of interest is adequate if the system has other large time constants. In Figure 14 , projections of the contour surface show that the shape of ridge did not change much when the sampling period was shorter than four units of time for step response modeling of a second order system (System I, page 18) with time constants equal to 7.14 and 1.26 units of time. It is obvious that for a step response the sampling period is not a major cause of the existence of ridge.



(a) Sampling Period : 8 units of time



(b) Sampling Period : 4 units of time



(c) Sampling Period : 0.5 unit of time

Figure 14. The Effect of Sampling Period on the Step Response Modeling Objective Function.

Another factor which should be considered is the experiment length. In practice the experiment length is limited by reasons such as operation safty or drift of system variables. Usually the experiment length should be at least ten times the largest time constant. Generally speaking, the model becomes more accurate as we increase the experiment length. This is particularly true if the noises in the data are completely random. However, the computation cost also increases with the experiment length. For most cases noises in data can be reduced if more sophisticated equipments are used to collect the data and all variables which affect the system are under close control.

(b) Generation of experimental data

As mentioned earlier, techniques in experimental modeling are only different tools used to extract information from experimental data. Therefore, the experimental data in the first place should contain enough information about the characteristics of the system to be identified.

A second order system can be described by a differential equation of the form

$$\tau^2 \frac{d^2x(t)}{dt^2} + 2\tau\zeta \frac{dx(t)}{dt} + x(t) = K\,F(t) \qquad (18)$$

At any instance the position, velocity, and acceleration of the system must be under the constraint of equation (18).

On the other hand, if a set of system response data is given and the parameters of the equation are yet to be determined from the data, the parameters must satisfy equation (18) for each data point, which contains the information of $x(t)$, $dx(t)/dt$, and $d^2x(t)/dt^2$. It has been shown in Figure 11 (page 33) that value of $d^2x(t)/dt^2$ of a system step response is much smaller than the value of $F(t)$, $x(t)$, or $dx(t)/dt$, and is nearly equal to zero for most period of step response. The effect is that step response data have little constraint on the parameter of the first term of equation (18). This is the reason why it is easy to identify the value of $\tau\zeta$ and K from step response data while the value of τ and ζ of the model can be arbitrary.

It has been shown that the problem was caused by the lack of information about $d^2x(t)/dt^2$ in the step response data. Rearranging equation (18), we obtain

$$\tau^2 \frac{d^2x(t)}{dt^2} = K\,F(t) - 2\tau\zeta \frac{dx(t)}{dt} - x(t) \qquad (19)$$

Equation (19) shows that the acceleration, $d^2x(t)/dt^2$, is proportional to the right side of the equation, the net force applied to the system. This equation gives clues of how to increase the value of $d^2x(t)/dt^2$. If the system is originally at steady state, $d^2x(t)/dt^2$ has the maximum value when a disturbance of step change starts to apply, and $d^2x(t)/dt^2$ then decreases as $dx(t)/dt$ and $x(t)$ start

growing. Since $d^2x(t)/dt^2$ is larger than $dx(t)/dt$ or $x(t)$ only in the first few moment when the system is exposed to the disturbance, the disturbance should be sustained only a short period of time to prevent $dx(t)/dt$ or $x(t)$ from becoming dominant. Equation (19) also shows that the strength of the disturbance, $F(t)$, should be kept at a maximum value within the range of linear operation and safty. Disturbance such as an impulse or a series of impulses can satisfy these requirements. If a series of impulses is used, two consecutive impulses should be separated by a sufficient period of time to allow $dx(t)/dt$ and $x(t)$ to subside. As a matter of fact, equation (19) also suggests that $d^2x(t)/dt^2$ can have an even larger value if $F(t)$ has an opposite sign with $dx(t)/dt$ or $x(t)$. This can be achieved somewhat by using a combination of two impulses of opposite directions. During the first impulse, $F(t)$, $dx(t)/dt$, and $x(t)$ have the same sign. During the second impulse the sign of $F(t)$ is changed while the signs of $dx(t)/dt$ and $x(t)$ remain unchanged for a period of time because of the inertia of the system. An input signal of a series of such combinaion can also be used. In Figure 15, it shows the response of System I (page 18) to such input signal. Since $d^2x(t)/dt^2$ had substantial value during the response, the coefficient of the first term in equation (18) was then heavily weighted. Once the value of τ could be specified from the response data, ζ could also be specified from the

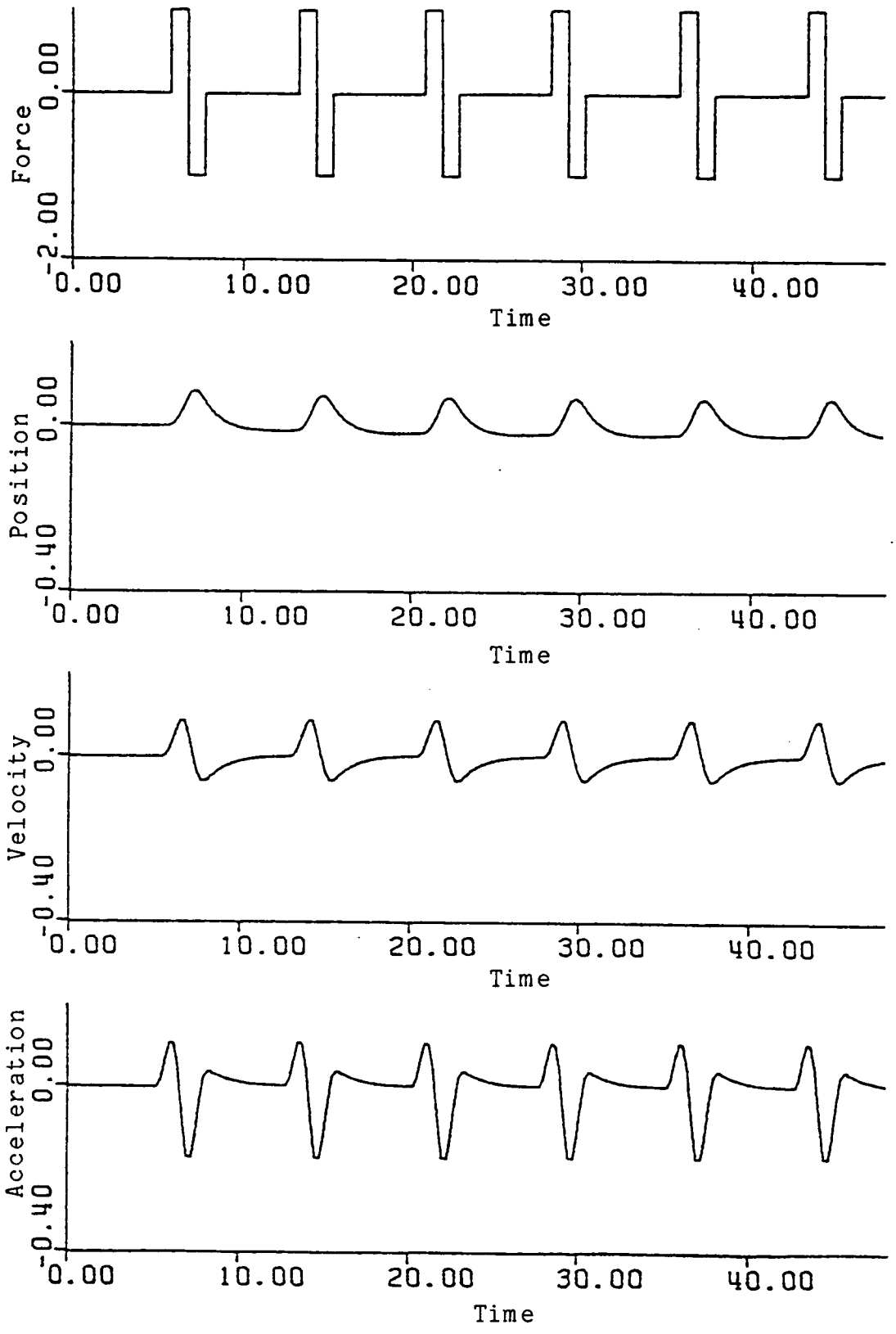
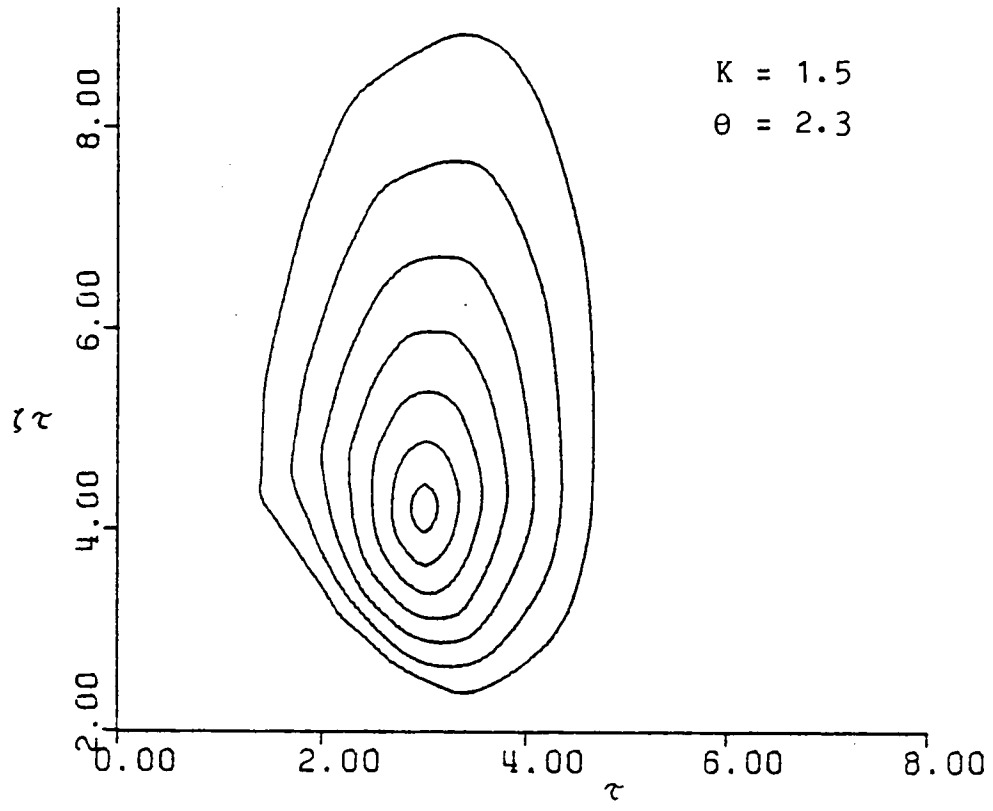


Figure 15. Response of a Second Order System to a Combination of Impulsive Forcing Function.

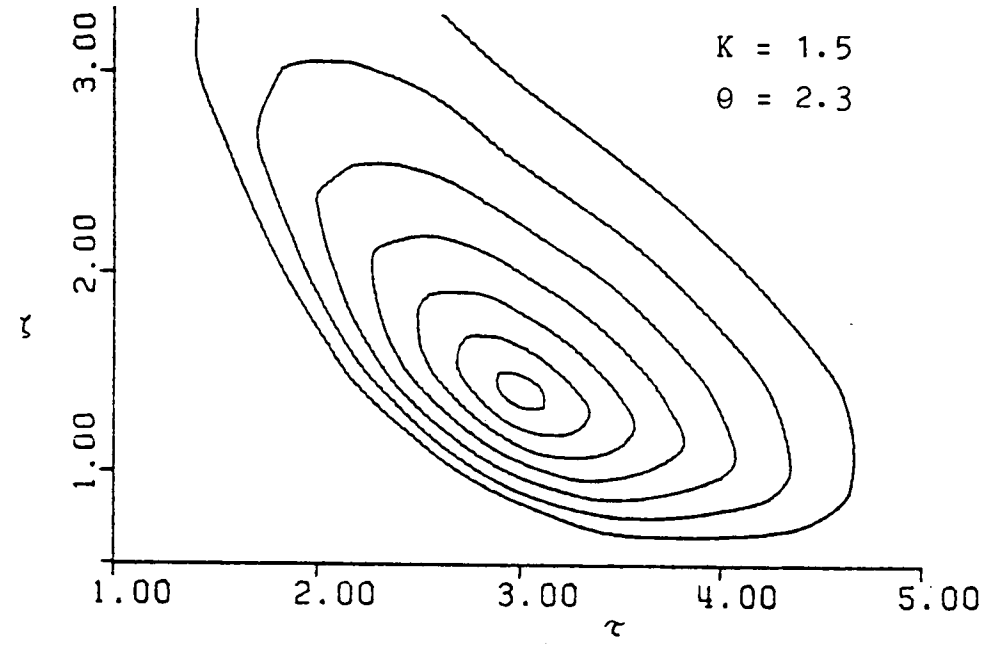
coefficient of the second term in equation (18).

In Figure 16 the projection of the impulsive response data modeling objective function on the τ-ζ plane shows that the ridge in Figure 8 (page 24) almost disappeared. An interesting effect was the ridge caused by the interactions among τ, ζ and dead time also disappeared. This effect also applied to the ridges in Figure 13 (page 37), as shown in Figure 17. At this point, it is quite obvious that the information content of response data has a prominent influence on the shape of contour surface.

Although the content of response data can be improved by using impulsive input signals, it could have a weak point in practical applications. Because generally there is a limit to the amplitude of the impulses, the amplitude of the input signal may not be large enough to cause a reasonable size of response. In the presence of noises, the low signal-to-noise ratio may ruin the information contained in the response data, and some other test signals which can cause higher amplitude of response are more desirable. In Figure 18, it shows that for the same second order system the amplitude of the response to pseudo-random binary sequence (PRBS) was about twice that of the impulsive signal response and that the response also contained adequate information of $dx(t)/dt$ and $d^2x(t)/dt^2$. The projections of the PRBS response data modeling objective function illustrated in Figure 19 show no serious ridge problem. The

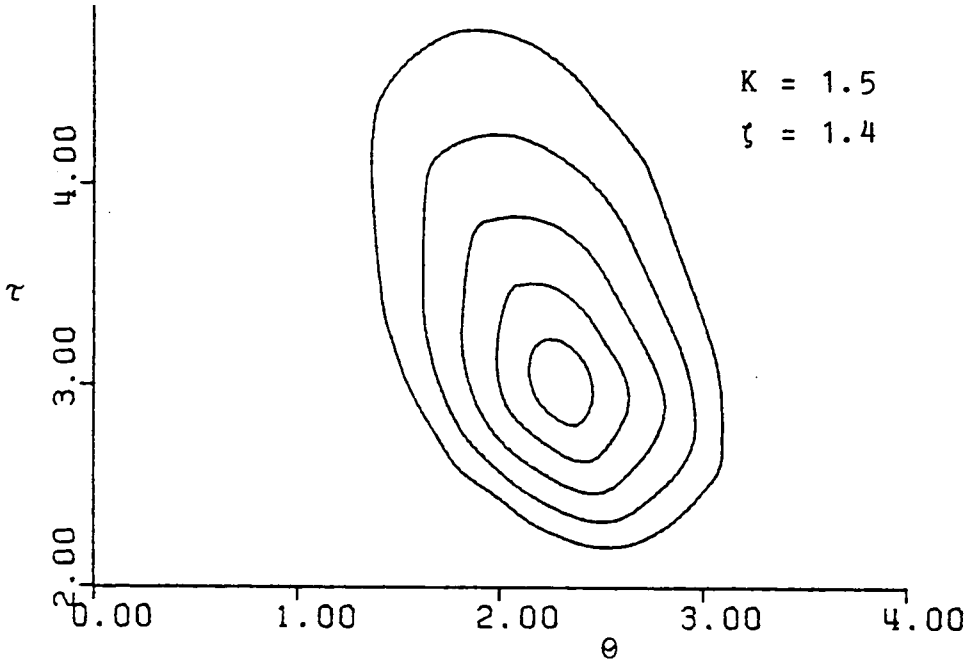


(a)

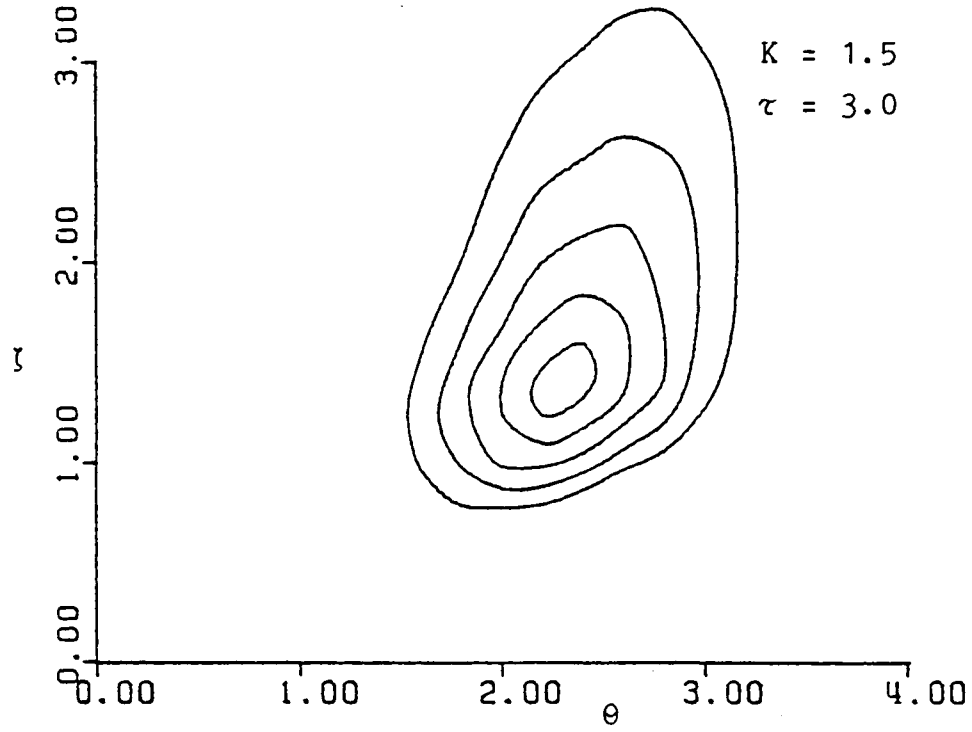


(b)

Figure 16. Projections of the Impulsive Response Modeling Objective Function of System I.

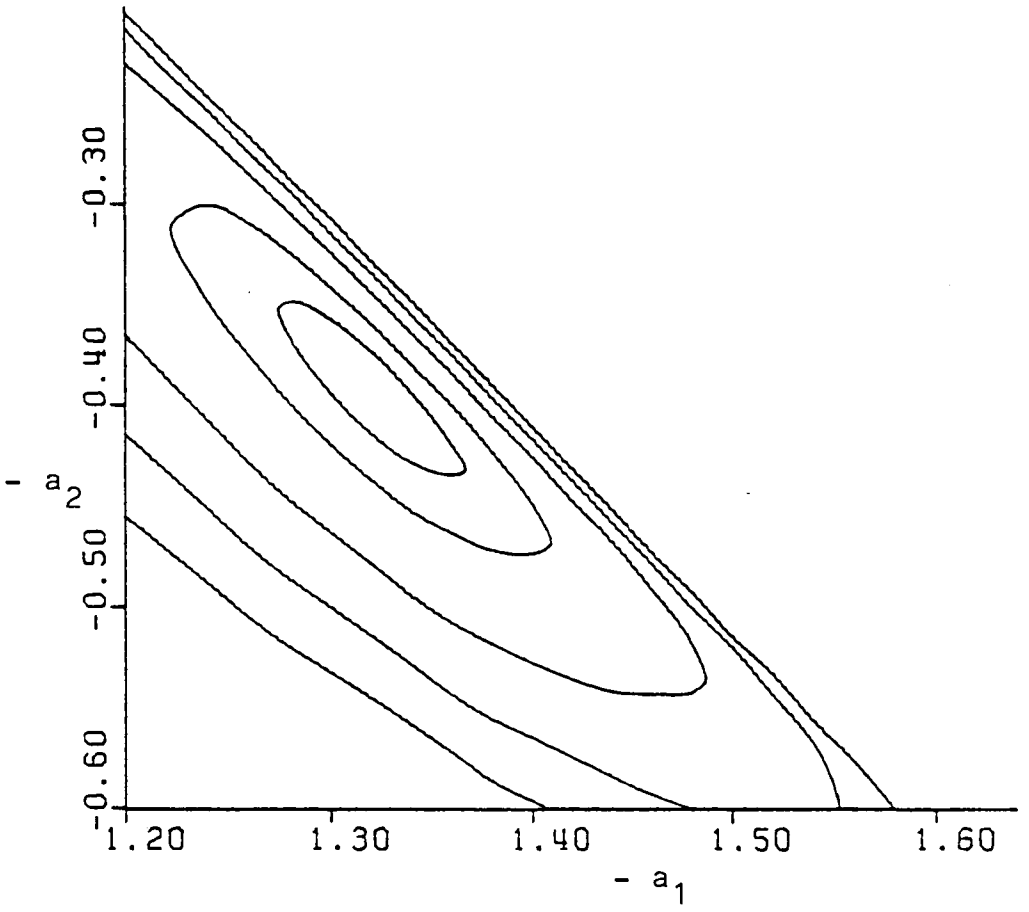


(c)

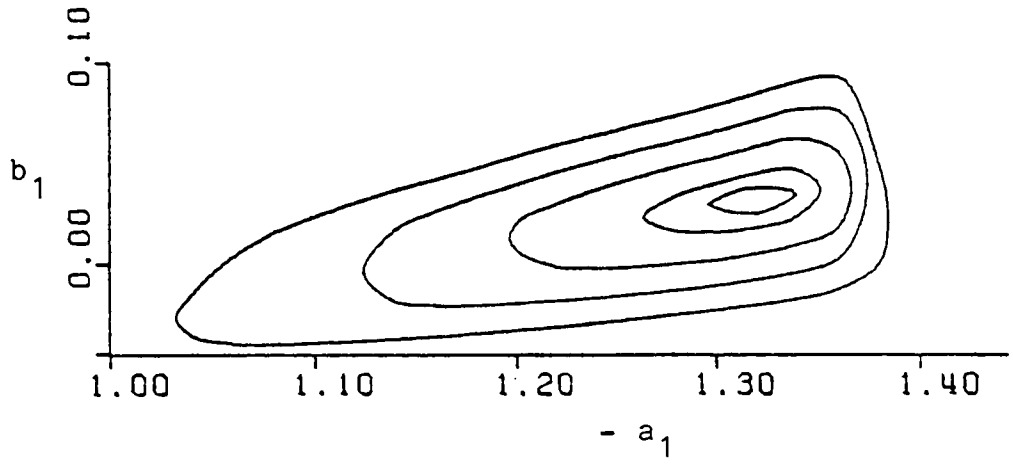


(d)

Figure 16. (Continued)

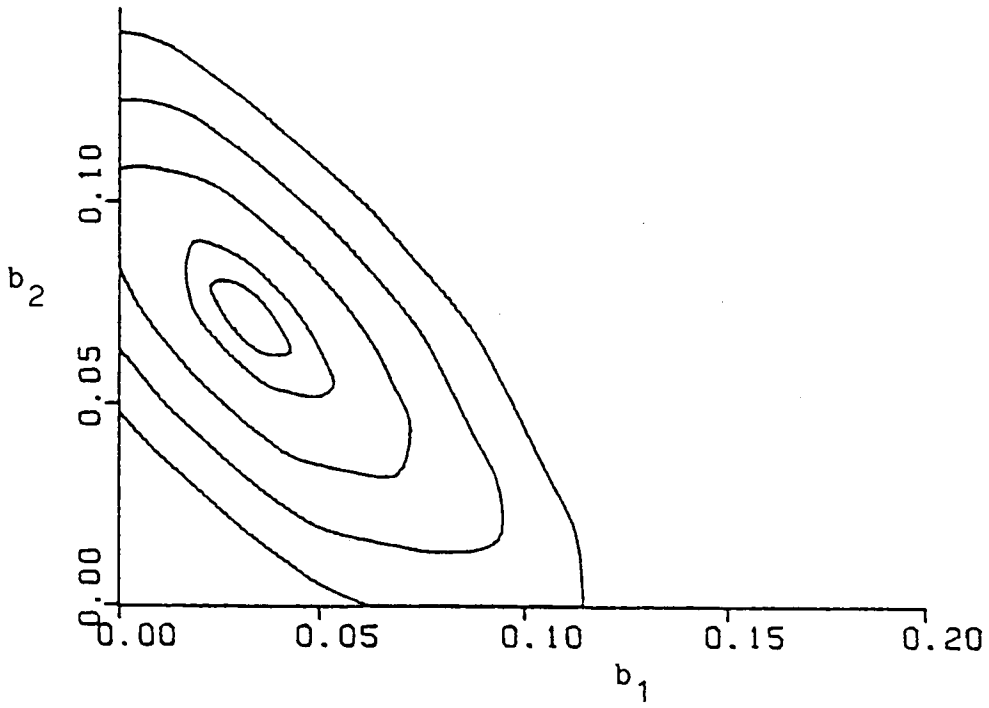


(a)



(b)

Figure 17. Projections of the Impulsive Response Modeling Objective Function by Using the Coefficients of Difference Equation as the Search Parameters.



(c)

Figure 17. (Continued)

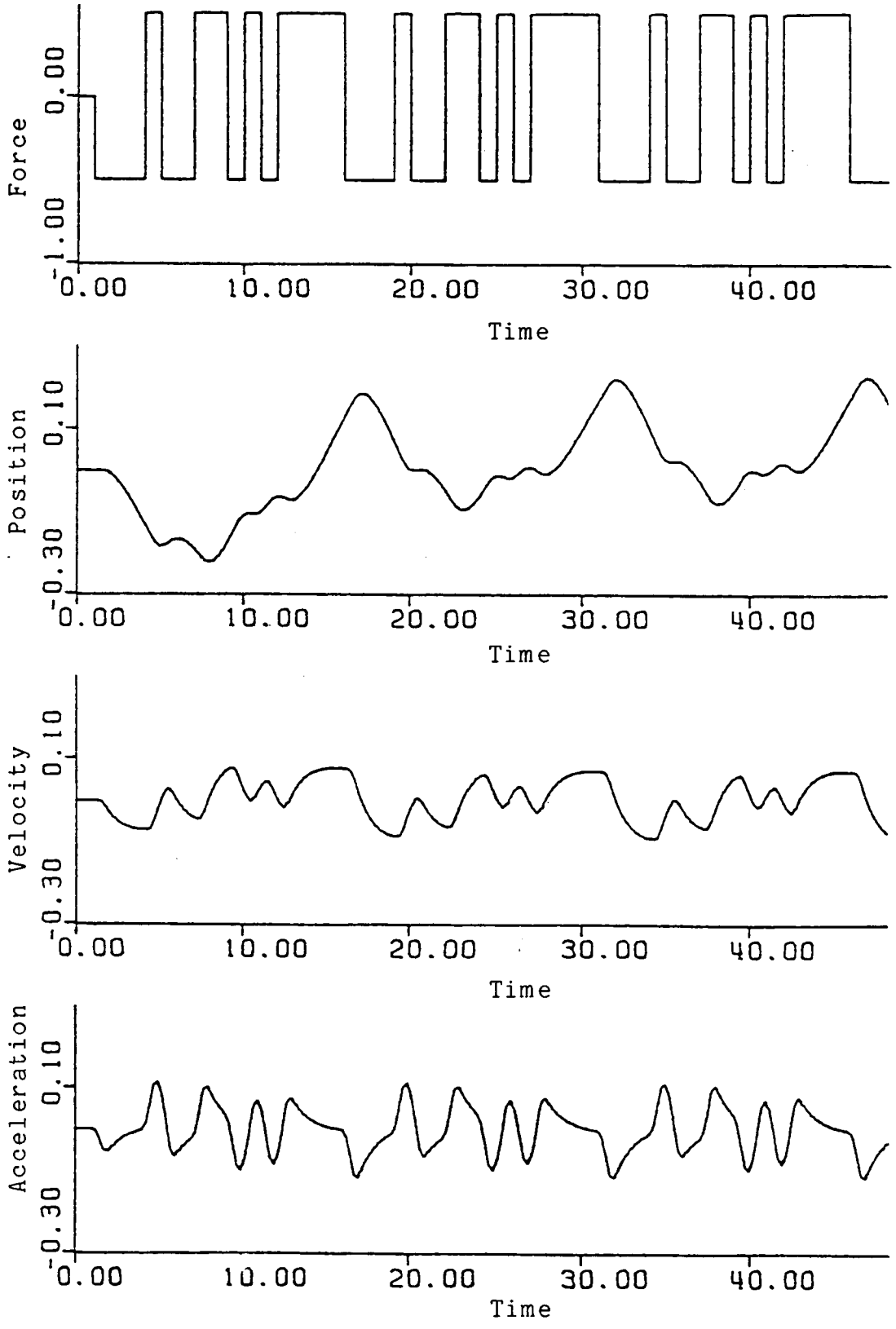
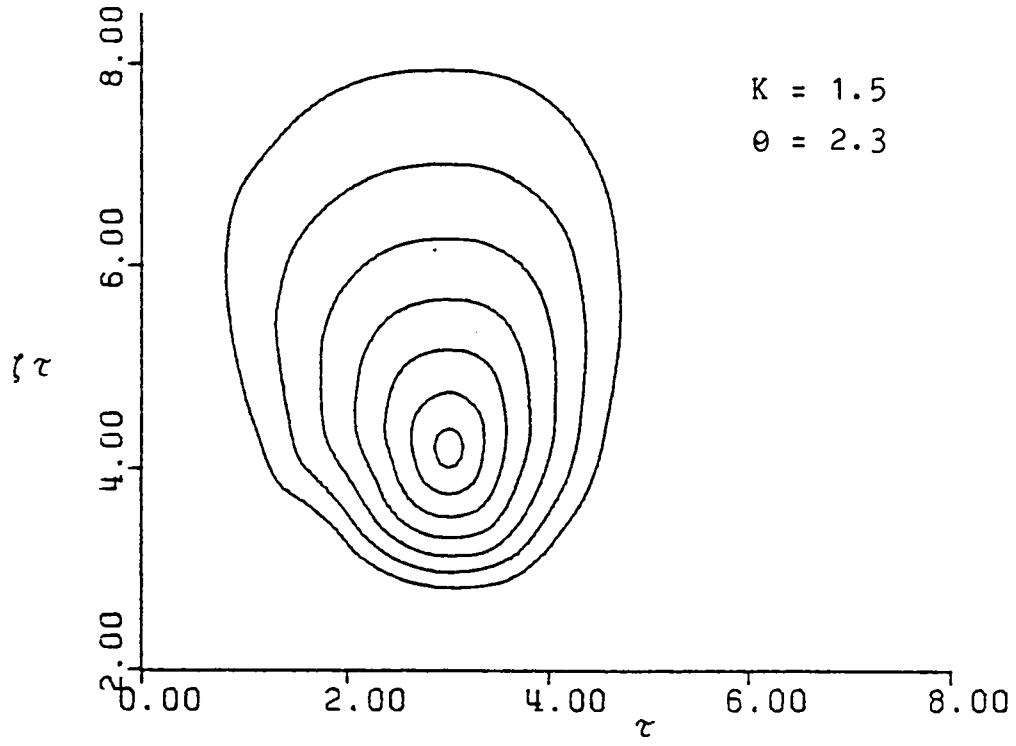
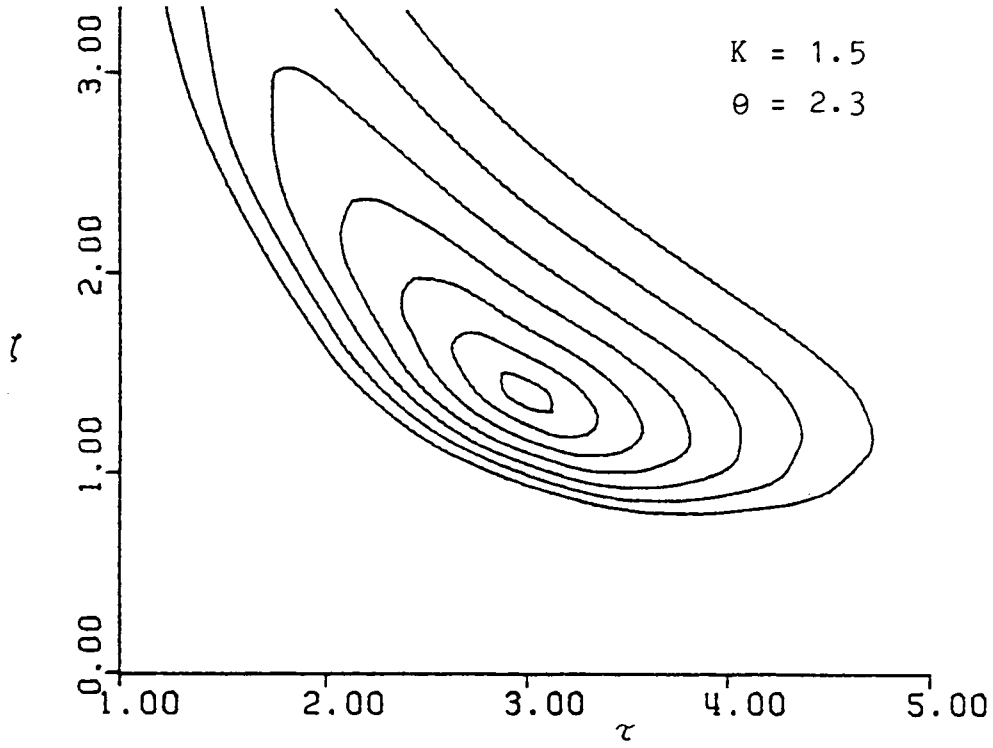


Figure 18. Response of a Second Order System to Pseudo-random Binary Sequence.

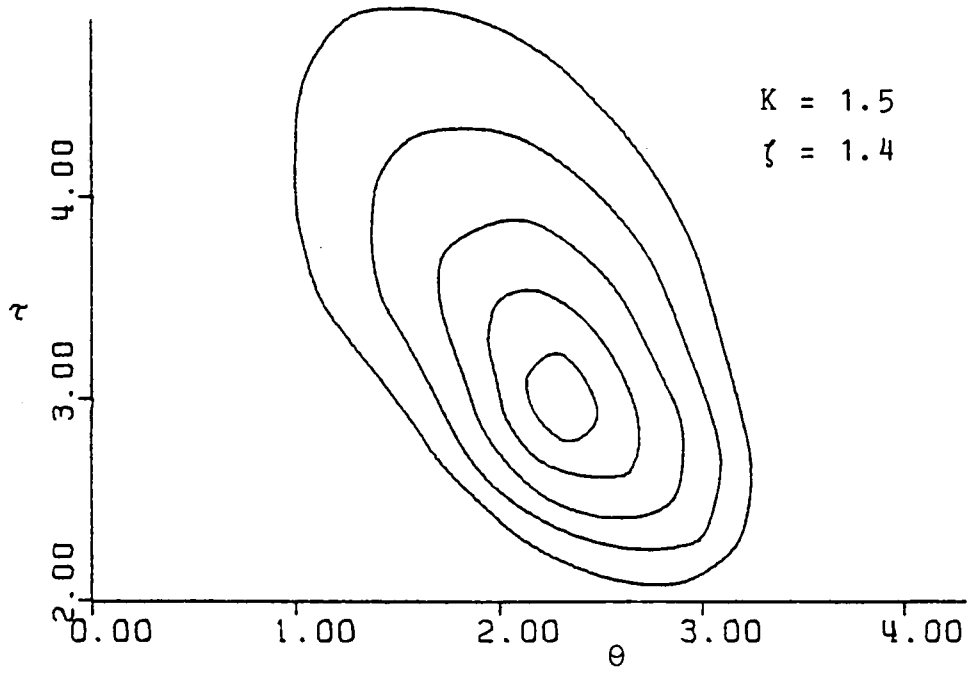


(a)

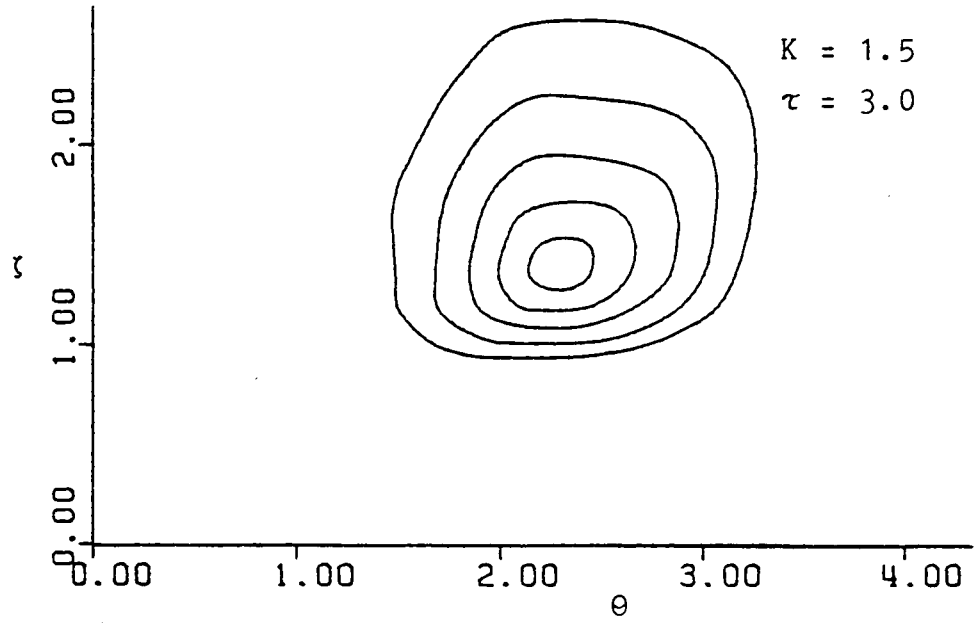


(b)

Figure 19. Projections of the Contour Surface of PRBS Response Data Modeling Objective Function.



(c)



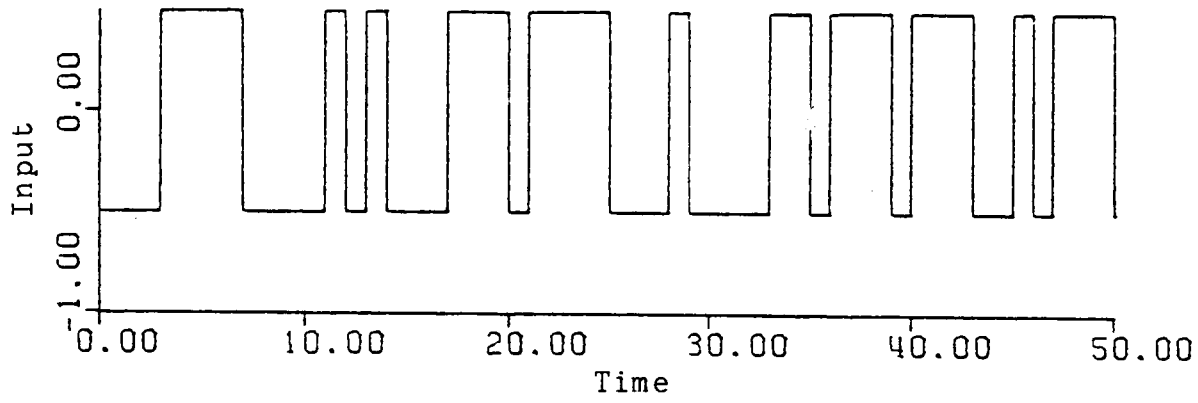
(d)

Figure 19. (Continued)

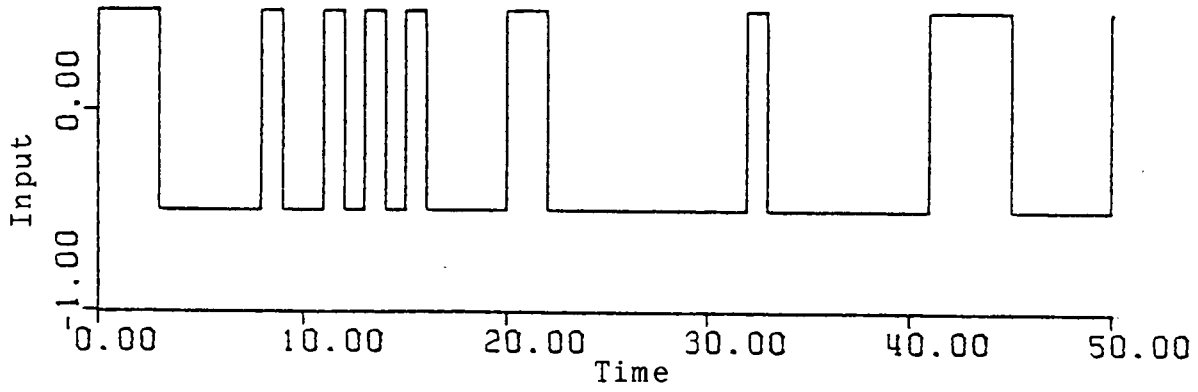
difference between impulsive signals and PRBS is that the power of impulsive signals is more evenly distributed from low frequency to high frequency while PRBS has more power distributed in the low frequency region. Since the amplitude ratio of the frequency response of most system decreases with frequency of the input signals, the information brought out by high frequency signal is often so weak that the information can be overshadowed by noises easily. Therefore, in the presence of noises, the accuracy of a model can be improved very little by increasing the power in the high frequency region, and we can get more payoff by emphasising the low frequency region. A general rule is that the system should be adequately excited, and the signal-to-noise ratio should be kept at a reasonable high value. A system can be adequately excited by using more drastic test signals, and the effect on the contour surface is elimination of the ridge problem. On the other hand, the signal-to-noise ratio can be increased by increasing the power of the test signals in the low frequency region so that the information contained in the data would not be completely overshadowed by noises. Since the power of an input signal is limited, it requires a compromise of the distributions of power in the low frequency and the high frequency region. Cumming (15) demonstrated with an underdamped system that test signals with a bandwidth (half power bandwidth) of about one half

that of the system gave the best results. One of the signals with easily controlled bandwidth is the synchronous random telegraph wave (SRTW), which can be generated on a digital computer. The bandwidth of SRTW can be adjusted by changing the switching probability of the signal as shown in Figure 20. Goodwin (16) showed that an optimal input signal designed from any non-optimal input-output data gave some improvement compared with a PRBS test signal.

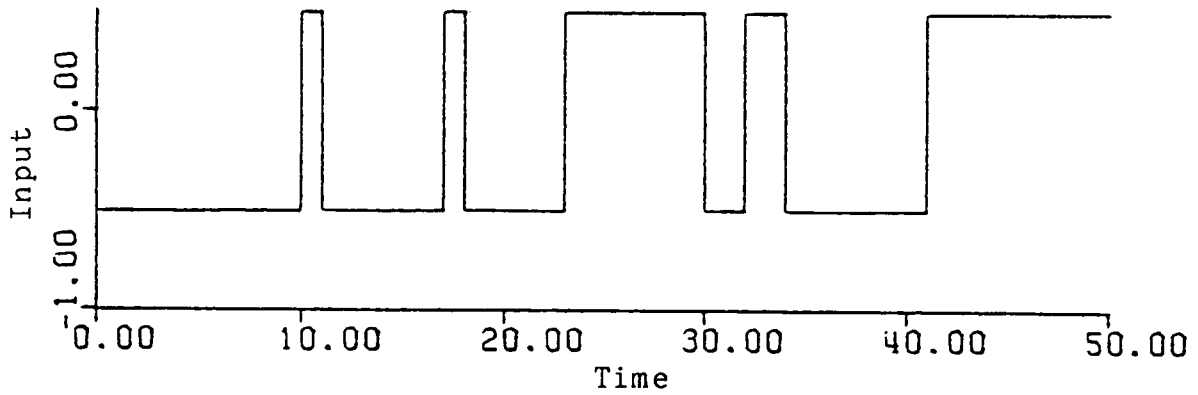
The power density spectrum of a test signal has another effect on the estimates of model parameters when the model is not exact, e.g., a second order model representing a higher order system. The characteristic of system response is usually dominated by large time constants of the system, and high order system can be represented by low order model quite well. However, small time constants may become influential as the high frequency content of the input signal is increased. The effects are more serious if the small time constants are very close to the large ones. If this is the case, the models obtained from input signals of different power spectrums are inconsistent. Under this situation either a higher order model should be used or it would be better to use an input signal with spectrum most close to that of the disturbance to which the system is exposed under normal operation condition.



(a) Switching Probability λ = 0.50



(b) Switching Probability λ = 0.30



(c) Switching Probability λ = 0.20

Figure 20. Synchronous Random Telegraph Waves with Different Switching Probabilities.

C. Modification of the Pattern Search Method

The ridge problem in experimental modeling can sometimes be eliminated by selecting a test signal such that the system is adequately excited. However, the choices of input signal are often limited because of presence of noises or slow response of the system. In these cases, ridges may still exist and a numerical method which can deal with ridges is required to locate the optimum.

A search method always runs the risk of being stuck on a ridge if the directions of search are confined to certain directions such as a set of conjugate directions or the directions of coordinate axes because the direction of a ridge may not be parallel to one of these directions. Since the search directions of the pattern search are parallel to the coordinate axes, the ridge problem can be solved by implementing the pattern search method with the ability to find the orientation of a ridge and to rotate the coordinate axes until one of the coordinate axes is parallel to the ridge whenever the search stalls on a ridge.

To find the proper rotation angles may requires a lot of computations especially when the number of search variables is large or the ridge is a curved one. The computations can be reduced by proper scalings of the coordinates when a curved ridge is encountered and the rotation angles have to be changed very often. Since in most applications the objective function is close to a

quadratic function, or at least it can be approximated by a quadratic function locally, the effect of scaling can be illustrated with a quadratic function.

The contour surface of a quadratic function consists of a family of ellipsoids and has a ridge lying on the major axis of the ellipsoids. A general form of a quadratic function of two independent variables is

$$G(x,y) = A\,x^2 + B\,xy + C\,y^2 + D\,x + E\,y + F \tag{20}$$

The angle of the major axis is

$$\phi = \frac{1}{2}\tan^{-1}\frac{B}{A - C} \tag{21}$$

The major axis can be represented by

$$y = m\,x - \frac{m\,(BE - 2CD)}{4AC - B^2} + \frac{BD - 2AE}{4AC - B^2} \tag{22}$$

where

$$m = \tan\phi = \frac{\sqrt{1 - \dfrac{A - C}{\sqrt{(A - C)^2 + B^2}}}}{\sqrt{1 + \dfrac{A - C}{\sqrt{(A - C)^2 + B^2}}}} \tag{23}$$

If the contour lines are circles, the contour surface is considered to have no ridge. The ridge appears as the contour lines become ellipses, and the ridge is getting narrow as the ellipses approach the shape of a parabola, i.e., the value of $B^2 - 4AC$ approaches 0^-.

It will be shown that the angle of the ridge (major

axis) can be normalized to $\pm \pi/4$ by scaling the coordinates when the contour lines approach the shape of a parabola (Figure 21).

The partial derivatives of the objective function are

$$\frac{\partial G}{\partial x} = 2A\ x + B\ y + D \tag{24a}$$

$$\frac{\partial G}{\partial y} = 2C\ y + B\ x + E \tag{24b}$$

Substitution of equation (22) for y, equations (24a) and (24b) gives the partial derivatives on the ridge.

$$\frac{\partial G}{\partial x} = \frac{(8A^2C + 2AB^2 + 4mABC - mB^3)\ x}{4AC - B^2} + \frac{(2mBCD - mB^2E - 2ABE + 4ACD)}{4AC - B^2} \tag{25a}$$

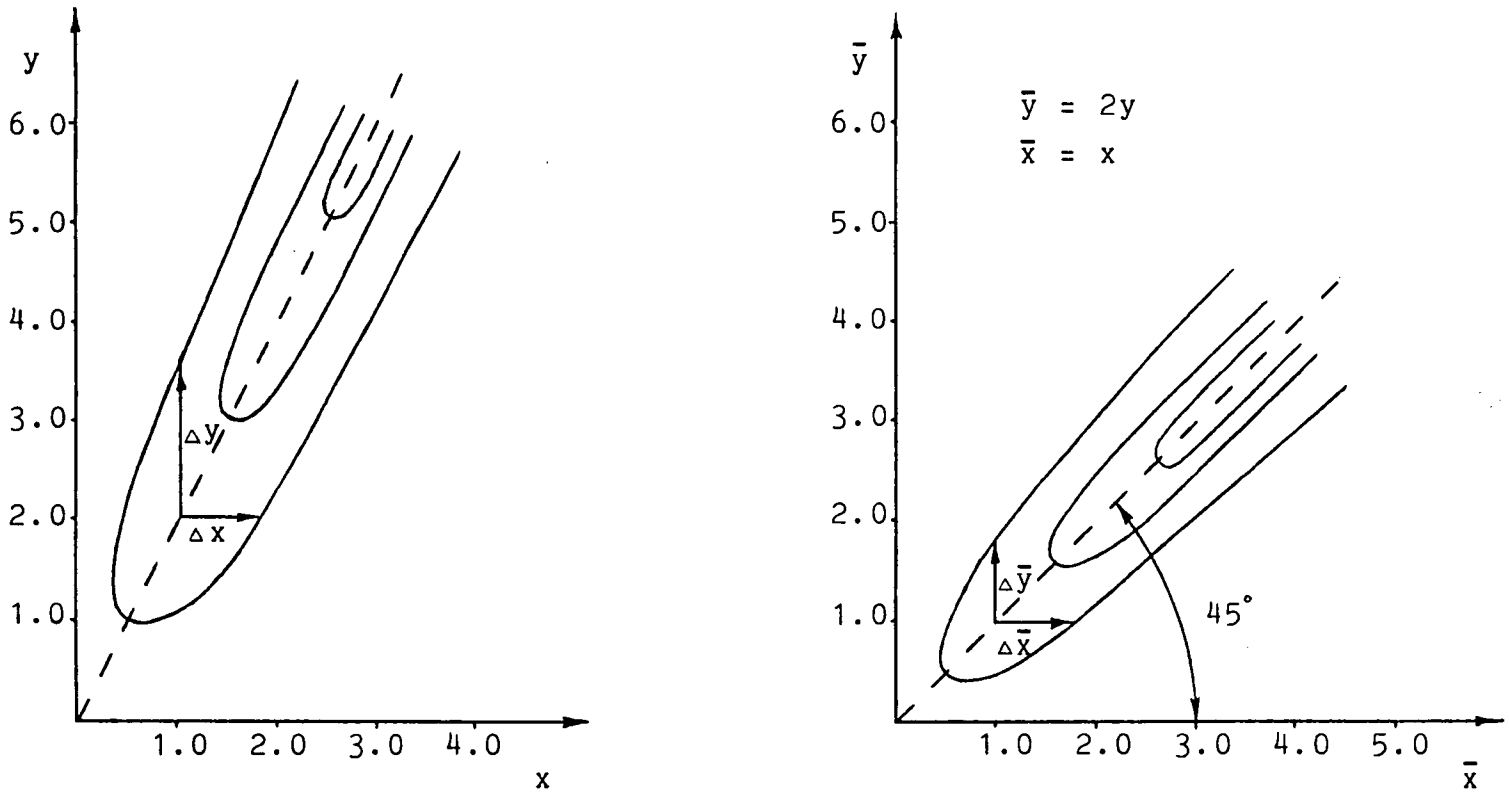
$$\frac{\partial G}{\partial y} = \frac{(8mAC^2 + 2mB^2C + 4ABC - B^3)\ x}{4AC - B^2} + \frac{(2BCD - B^2E - 2mBCE + 4mC^2D)}{4AC - B^2} \tag{25b}$$

Let

$$\frac{\partial G}{\partial x} = P + Q \tag{26a}$$

$$\frac{\partial G}{\partial y} = R + S \tag{26b}$$

where P, Q, R, and S are equal to the corresponding terms in the brackets in equations (25a) and (25b). The derivatives



(a) Ridge in the Original Space (b) Ridge in a Properly Scaled Space

Figure 21. Normalization of the Ridge Orientation by Scaling the Coordinate System.

can be approximated by

$$\frac{\partial G}{\partial x} \cong \frac{G(x+\Delta x,y) - G(x,y)}{\Delta x} \tag{27a}$$

$$\frac{\partial G}{\partial y} \cong \frac{G(x,y+\Delta y) - G(x,y)}{\Delta y} \tag{27b}$$

Then

$$G(x+\Delta x,y) - G(x,y) \cong (P + Q)\,\Delta x \tag{28a}$$

$$G(x,y+\Delta y) - G(x,y) \cong (R + S)\,\Delta y \tag{28b}$$

By selecting Δx, and Δy, we can get

$$G(x+\Delta x,y) - G(x,y) = G(x,y+\Delta y) - G(x,y) \tag{29}$$

or
$$(P + Q)\,\Delta x = (R + S)\,\Delta y \tag{30}$$

Let $\Delta y/\Delta x = J$, equation (30) becomes

$$(P + Q)\,\Delta x = (R + S)\,J\,\Delta x \tag{31}$$

or

$$\left[\frac{(8A^2C + 2AB^2 + 4mABC - mB^3)\,x}{4AC - B^2} + \frac{(2mBCD - mB^2E - 2ABE + 4ACD)}{4AC - B^2}\right]\Delta x$$

$$= \left[\frac{(8mAC^2 + 2mB^2C + 4ABC - B^3)\,x}{4AC - B^2} + \frac{(2BCD - B^2E - 2mBCE + 4mC^2D)}{4AC - B^2}\right]J\Delta x \tag{32}$$

If the value of $B^2 - 4AC$ appoaches zero (a narrow ridge) equation (23) can be squared twice and gives the relation

$$m - \sqrt{A/C} = 0 \tag{33}$$

Substitution of equation (33) for m in equation (32) gives

$$J = \sqrt{A/C} \tag{34}$$

If the scaling factor J is used to construct a new coordinate system, we have

$$x = \bar{x} \qquad \text{and} \qquad y = \sqrt{A/C}\ \bar{y}$$

The objective function, equation (19), can be written in the new coordinate system as

$$\bar{G}(\bar{x},\bar{y}) = A\bar{x}^2 + B\sqrt{A/C}\ \bar{x}\bar{y} + A\bar{y}^2 + D\bar{x} + E\sqrt{A/C}\ \bar{y} + F \tag{35}$$

In the rescaled coordinate system, the angle of the major axis becomes

$$\phi = \frac{1}{2} \tan^{-1} \frac{B\sqrt{A/C}}{A - A} = + \frac{\pi}{4} \tag{36}$$

The scaling of coordinate in the pattern search method is to use Δx, Δy, which satisfy equation (29), as the scales of the new coordinates. When the pattern search stalls on a ridge, by properly scaling the coordinate system, we can normalize the angle of the ridge to $\pm \pi/4$ for the case of two-dimension search. A rotation of $\pi/4$ after scaling should align one of the coordinate axes with the ridge. If we have a curved ridge, constant changes of rotation angles can be replaced by rescaling the search parameters locally whenever the pattern search stalls on the ridge. The idea can also be applied to an objective function of n variables or a function with extremely narrow ridge such that the contour lines are almost parallel to each other.

If the pattern search is to be used for the search of rotation angles, we need to define a cost function of which rotation angles are the independent variables. When the search of parameters stalls on a ridge, the value of the objective function can be reduced the most by searching in a coordinate system which has been rotated such that one of the coordinate axes is parallel to the ridge. The cost of a set rotation angles can be defined as the minimum objective function value which can be achieved after searching at a given distance in each axis direction of the rotated coordinate system. Therefore, the rotation with the minimum cost which the pattern search can find is the rotation which will bring one of the coordinate axes parallel to the ridge. Searching the rotation angles in the rescaled space is equivalent to locating the point with the minimun objective function value on a sphere of unit radius which is centered at the point where search of parameters has stalled. In Figure 22, it shows that the best rotation angle is ϕ beause the minimum objective function on the circle can be achieved in the direction of $\bar{x}'$.

If the objective function is quadratic, one of the coordinate axes will be parallel to the major axis of the ellipses after rotation. A scaling in the rotated space can transform the ellipses into circles. In other words the ridge will disappear if we scale the rotated space properly. The procedures to eliminate a ridge are illustrated in Figure 23.

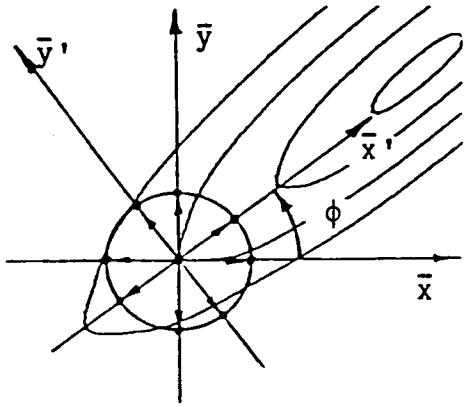


Figure 22. Search of Rotation Angle in a Properly Scaled Space.

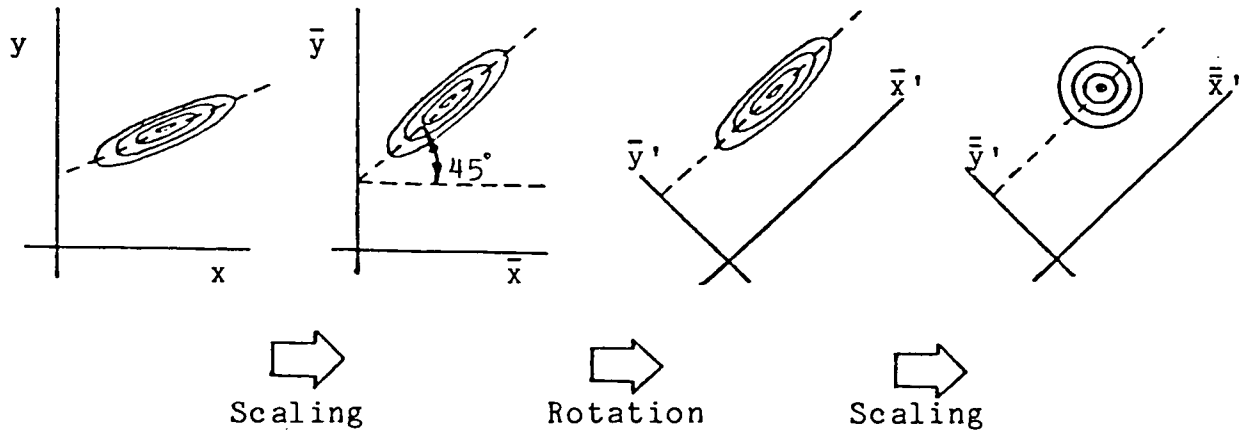


Figure 23. Elimination of a Ridge by Coordinate Scaling and Rotation.

Although the secondary scaling is not a necessary step to make the search follow the ridge when an axis is already parallel to the ridge, the secondary scalling will make the pattern search more efficient. Secondary scaling is helpful especially when the ridge is extremely narrow or the rotation angles are not precisely determined.

The pattern search method was modified by adding a SEARCH subroutine to direct the strategy of scaling, parameter search, and angle search. Also a subroutine ROTATE was used to perform the coordinate rotation. The flow chart of the SEARCH subroutine is shown in Figure 24, and the program structure of the modified pattern search method is shown in Figure 25.

The strategy starts with a search without scaling and rotation. When the search stalls, it is assumed that the search have stalled on a ridge, and search with scaling and rotation is then performed. The orientation of the ridge is first normalized by scaling. Then the search of rotation angles is performed. If the search stalls because the direction of the ridge changes afterward, the local orientation of the ridge can be normalized simply by scaling the coordinate system again, and the rotation angles are still the same. If the search fails to make any progress after normalization of the ridge by scaling, search of new rotation angles is then performed. The roundoff error in

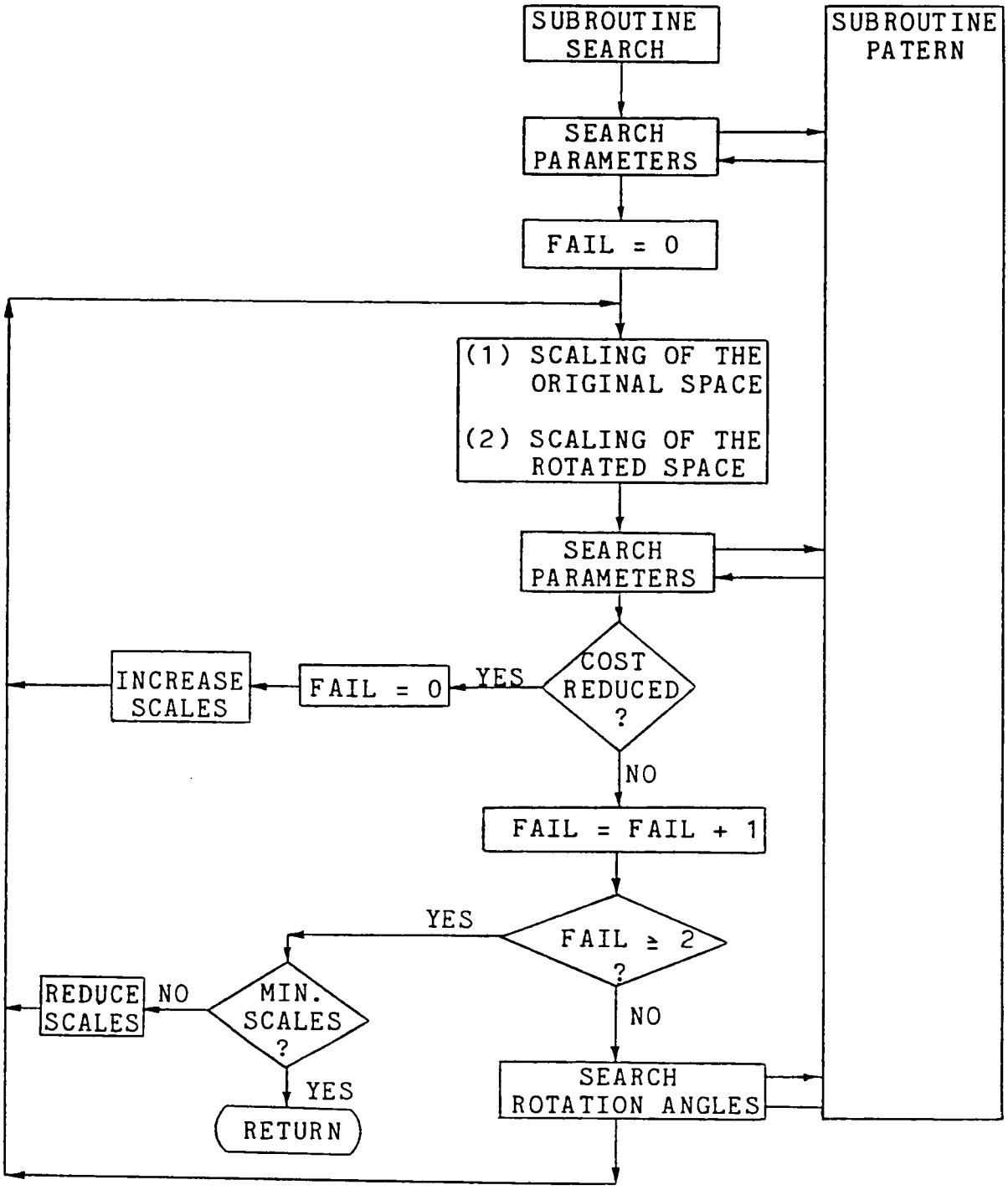


Figure 24. Flow Chart of the Subroutine SEARCH of the Modified Pattern Search.

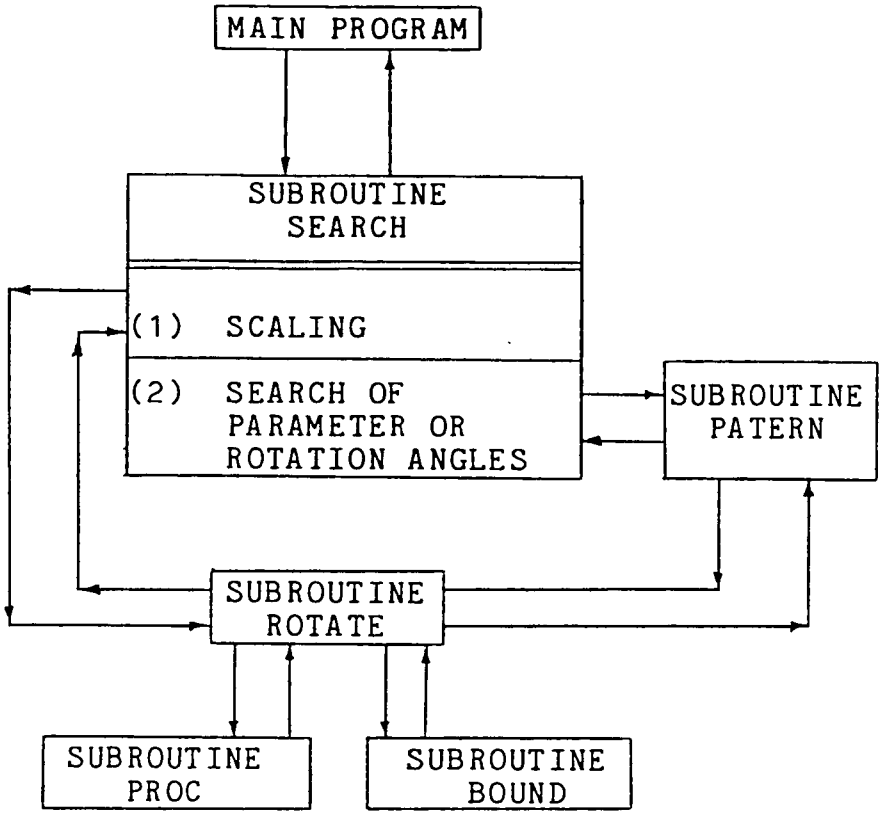


Figure 25. Program Structure of the Modified Pattern Search Method.

computer may also cause search to stop if the contour surface is nearly flat and the scales of the coordinate system is too small. Therefore, the pattern search is allowed to search within a range of the current objective function value when search fails after new rotation angles have been tried. To make the search convergent and more efficient, the scales of the coordinate system will be reduced when all search strategies fail and will be increased if cost is reduced when search returns from the subroutine PATERN. The procedures are repeated until the minimum scales are reached.

To compare the modified pattern search with some other nonlinear optimization techniques, the following two objective functions were used.

Function I :

$$\text{Cost} = \left[(x^2 + y^2 - 1)\, G\right]^2 + \left\{\left[z - \tan^{-1}\left(\frac{y}{x}\right)\right] G\right\}^2 + 0.1z^2 \qquad (37)$$

where $x > 0$, $y > 0$, and $z > 0$. The minimum of the function is (1,0,0).

Function II :

$$\text{Cost} = \left|(x^2 + y^2 - 1)\, G\right| + 0.1\left|x - 1\right| \qquad (38)$$

The minimum of the function is (1,0).

The ridges of both objective functives can be made narrow by increasing the value of G. TABLE III shows that for Function I the Fletcher-Reeve method failed when $G > 100$, and the Powell method failed when $G > 500$. Although the flexible polyhedron search requires less function evaluations than the modified pattern search did, the flexible polyhedron search failed to reach the optimum of Function II when $G > 10$ (TABLE IV).

TABLE III. Number of Function Evaluations for the Minimization of Function I

Method	G = 10	G = 100	G = 500
Fletcher-Reeve	23,095	Failed	Failed
Powell	391	2,029	Failed
Flexible Polyhedron Search	737	2,754	7,971
Modified Pattern Search	1,844	7,065	17,429

TABLE IV. Final Point Reached in the Minimization of Function II

Method	G = 1	G = 10	G = 100
Flexible Polyhedron Search	(0.999,0.044)	(0.976,0.217)	(0.222,0.974)
Modified Pattern Search	(0.999,0.000)	(0.999,0.001)	(0.999,0.010)

CHAPTER V

APPLICATION OF THE MODIFIED PATTERN SEARCH METHOD

A. Experimental Modeling

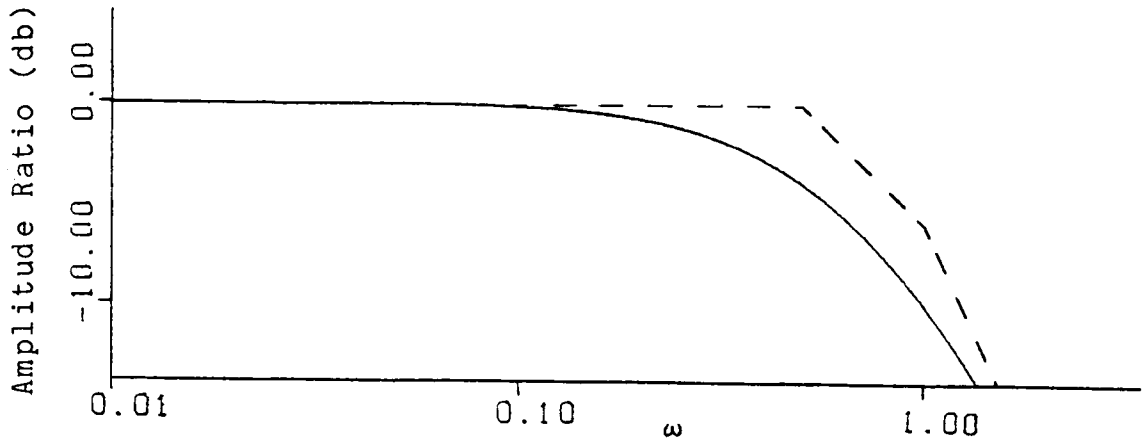
After the modification the pattern search method is able to follow a curved ridge to reach the optimum of an objective function. Therefore, the selection of input signal is not an essential problem for system experimental modeling if the data can be collected without errors. In TABLE V, it shows that experimental modeling by using modified pattern search and noiseless response data of a second order system gave consistent and correct model for several different input signals. In practical application, however, experimental data are usually corrupted with noises. It is obvious that the accuracy of a model is quite dependent on the signal-to-noise ratio of the data. For a given level of noise, the accuracy of the model is then decided by the information contained in the data. The information contained in response data can be evaluated in terms of quality and quantity. The quality is related to the frequency of response signal components. The quantity is the power of the response signal.

For most systems the frequency response curves are nearly flat in the low frequency region and more distinctive in the high frequency region as shown in Figure 26(a). Therefore, from the viewpoint of system identification, high

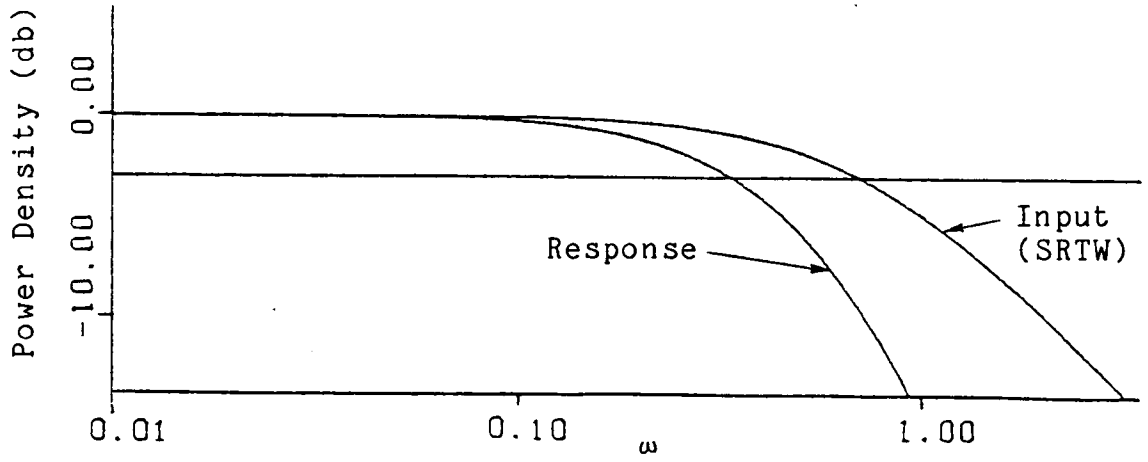
TABLE V. Experimental Modeling By Using Noiseless Response Data of a Second Order System

Input Signal	K	τ	$\tau\zeta$	θ
Unit Step Change	1.5000	1.4142	1.5000	2.3000
SRTW (λ = 0.05)	1.5000	1.4142	1.5000	2.3000
SRTW (λ = 0.10)	1.5000	1.4142	1.5000	2.3000
SRTW (λ = 0.25)	1.5000	1.4142	1.5000	2.3000
PRBS (λ = 0.50)	1.5000	1.4142	1.5000	2.3000
Impulsive Signal	1.5000	1.4142	1.5000	2.3000

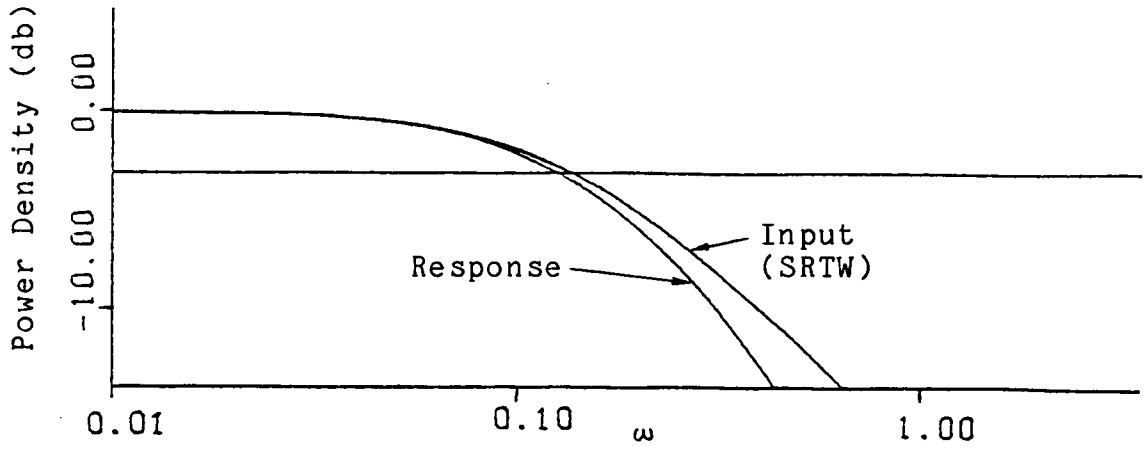
λ : switching probability



(a) System Frequency Response



(b) SRTW Switching Probability $\lambda = 0.25$



(c) SRTW Switching Probability $\lambda = 0.0625$

Figure 26. System Frequency Response and Power Density Spectrum of System Response to SRTW.

frequency components in the response signal stand for information of high quality. To improve the quality of response data, an input signal contains more high frequency components such as an impulsive input signal is favorable.

In the presence of noise we have to increase the amplitude of the response signal in order to get a reasonable high signal-to-noise ratio. Because the amplitude of a system frequency response generally decreases with frequency, it would be more desired to use input signal of low frequency if only the power of system response is of our concern. Figure 26 shows that the power can be more efficiently recovered by reducing the bandwidth of the input signal. Therefore, a slow changing input signal such as a unit step change, would most satisfy the requirement of power recovery. However, as we reduce the bandwidth of the input signal, we also reduce the bandwidth of the response and hence the quality of the response data.

TABLE VI and TABLE VII show the results of modeling by using noise corrupted response data of the following two systems:

$$G_p(s) = \frac{K\, e^{-\theta s}}{\tau^2 s^2 + 2\tau\zeta s + 1} \qquad (39)$$

System III : $K = 1.5$, $\tau = 1.414$, $\tau\zeta = 1.5$, $\theta = 2.3$

System IV : $K = 1.5$, $\tau = 4.583$, $\tau\zeta = 5.0$, $\theta = 2.3$

To simulate the noise, random numbers in normal

TABLE VI. Models Obtained from Noise Corrupted Response Data of System III

Input Signal	K	τ	$\tau\zeta$	θ
Unit Step Change	1.4983	1.1493	1.4592	2.4705
	1.5006	1.1977	1.4381	2.4565
	1.4969	1.3659	1.4731	2.3511
	1.5017	1.3270	1.4983	2.3482
	1.4995	1.5121	1.5011	2.2440
	1.4990	1.1029	1.3956	2.4875
	1.5006	1.4700	1.5326	2.2688
	1.4990	1.3487	1.4610	2.3508
	1.5014	1.3597	1.4815	2.3254
	1.4985	1.6457	1.5204	2.2124
σ	0.0015	0.1725	0.0452	0.0859
SRTW (λ = 0.05)	1.4972	1.3219	1.4350	2.3847
	1.5007	1.3622	1.4767	2.3449
	1.4944	1.3688	1.4772	2.3352
	1.4973	1.4442	1.5035	2.2918
	1.5019	1.3292	1.4407	2.3889
	1.5005	1.5326	1.5207	2.2293
	1.5008	1.4473	1.5152	2.2603
	1.4992	1.4664	1.4970	2.2400
	1.5019	1.3985	1.4857	2.3141
	1.4982	1.3662	1.4865	2.3035
σ	0.0024	0.0644	0.0314	0.0537
SRTW (λ = 0.10)	1.4956	1.4206	1.5260	2.2879
	1.4986	1.3917	1.4893	2.3205
	1.4951	1.3654	1.4890	2.3346
	1.4968	1.3381	1.4880	2.3434
	1.5005	1.4873	1.5351	2.2398
	1.4995	1.3071	1.4573	2.4001
	1.5000	1.4147	1.5212	2.3037
	1.4998	1.3828	1.4726	2.3568
	1.5019	1.4434	1.5083	2.2785
	1.4978	1.4593	1.5058	2.2326
σ	0.0025	0.0542	0.0233	0.0505

TABLE VI. (Continued)

Input Signal	K	τ	τ_I	θ
	1.4963	1.4693	1.5087	2.2384
	1.4986	1.4052	1.5089	2.3162
	1.4947	1.4183	1.4841	2.2933
	1.4965	1.4099	1.5151	2.2957
SRTW	1.5013	1.4371	1.4938	2.3129
(λ = 0.25)	1.5006	1.4262	1.5295	2.2739
	1.5004	1.4783	1.5216	2.2693
	1.4990	1.3524	1.5055	2.3485
	1.5019	1.4515	1.5140	2.2670
	1.4980	1.4245	1.5221	2.2707
σ	0.0026	0.0364	0.0165	0.0320
	1.4983	1.4398	1.4626	2.2691
	1.5002	1.4141	1.5050	2.2807
	1.4948	1.4000	1.5035	2.3166
	1.4969	1.3925	1.4855	2.3195
PRBS	1.5031	1.4339	1.5238	2.2749
(λ = 0.50)	1.5001	1.4438	1.5011	2.2708
	1.5006	1.3996	1.5094	2.3234
	1.5008	1.4218	1.4978	2.3216
	1.5019	1.4022	1.4935	2.3239
	1.4986	1.3655	1.5007	2.3319
σ	0.0023	0.0268	0.0153	0.0246
	1.4943	1.2499	1.4718	2.3885
	1.4995	1.4300	1.4564	2.2992
	1.4971	1.4638	1.3341	2.1916
	1.5013	1.5047	1.6442	2.2887
Impulsive	1.5003	1.4228	1.4095	2.3069
Signal	1.4980	1.4230	1.4270	2.2916
	1.5002	1.4221	1.5721	2.2962
	1.4988	1.4481	1.4582	2.2365
	1.5019	1.3205	1.5558	2.3573
	1.4982	1.5477	1.3995	2.2213
σ	0.0024	0.0812	0.0920	0.0578

λ : switching probability

σ : standard deviation from system parameter

TABLE VII. Models Obtained from Noise Corrupted Response Data of System IV

Input Signal	K	τ	$\tau\zeta$	θ
Unit Step Change	1.4980	4.7810	5.0305	2.3350
	1.5033	4.8404	5.0804	2.0992
	1.4966	4.7651	5.0194	2.1577
	1.5013	4.9019	5.0653	2.1238
	1.4988	4.8217	4.9903	2.2216
	1.4994	3.5128	4.6700	3.0474
	1.4999	4.8221	5.1515	2.0959
	1.5007	4.5012	4.9555	2.4128
	1.4983	4.8335	4.9862	2.1673
	1.5015	4.7777	5.0958	2.1017
σ	0.0016	0.4010	0.1246	0.2773
SRTW (λ = 0.05)	1.4973	4.5762	5.0168	2.2990
	1.5005	4.5583	4.9814	2.3020
	1.4958	4.6415	4.9554	2.2512
	1.5009	4.5113	4.9063	2.2852
	1.5004	4.3535	4.9329	2.4387
	1.4978	4.6982	5.0390	2.1485
	1.4966	4.5337	4.9458	2.3533
	1.4998	4.7664	5.0443	2.1427
	1.4990	4.5826	5.0038	2.2032
	1.5016	4.5524	5.0119	2.2771
σ	0.0021	0.1059	0.0474	0.0907
SRTW (λ = 0.10)	1.4969	4.7975	5.0396	2.2328
	1.4997	4.6584	5.0611	2.2387
	1.4951	4.5417	4.9738	2.3399
	1.5012	4.5553	5.0503	2.3995
	1.4993	4.7180	5.0805	2.1703
	1.4985	4.5597	5.0974	2.2524
	1.4979	4.7149	5.0742	2.2158
	1.4996	4.5265	4.9615	2.3955
	1.4999	4.3645	5.0147	2.4676
	1.5027	4.4914	5.0388	2.3682
σ	0.0022	0.1222	0.0576	0.0937

TABLE VII. (Continued)

Input Signal	K	τ	$\tau\zeta$	θ
	1.4962	4.6773	4.9808	2.1751
	1.4992	4.5784	5.0528	2.3654
	1.4950	4.5951	5.0618	2.2245
	1.5016	4.2439	4.9661	2.5394
SRTW	1.4990	4.6508	4.9173	2.2785
(λ = 0.25)	1.4984	4.8402	5.0770	2.1620
	1.4963	4.6514	5.0578	2.2548
	1.4997	4.3950	4.9696	2.4933
	1.4986	4.6427	4.8677	2.3264
	1.5019	4.6827	5.0364	2.2245
σ	0.0027	0.1576	0.0663	0.1217
	1.4963	4.4275	4.9844	2.1924
	1.5004	4.7754	4.8207	2.1093
	1.4953	4.5198	5.0661	2.4159
	1.5001	4.6727	4.7200	2.2136
PRBS	1.4986	4.7211	4.8076	2.2023
(λ = 0.50)	1.4968	4.7095	4.8562	2.3275
	1.4979	4.5613	4.8996	2.2555
	1.5024	4.6247	5.1681	2.3198
	1.5003	4.3618	4.7743	2.4388
	1.5022	4.5571	4.8949	2.3123
σ	0.0025	0.1266	0.1653	0.1005
	1.4932	3.2238	8.1319	2.6912
	1.5003	3.9327	8.3349	2.8022
	1.4963	3.8771	3.4232	1.7589
	1.5017	4.9584	5.4692	3.0233
Impulsive	1.4996	3.1907	9.0875	3.0115
Signal	1.4986	3.8667	2.1643	2.0173
	1.4973	3.0837	8.7240	3.0037
	1.5004	5.5996	3.1927	2.2187
	1.4990	4.4997	4.1615	1.9957
	1.5022	4.3468	4.4024	2.5025
σ	0.0028	0.9328	2.5803	0.4942

λ : switching probability

σ : standard deviation from system parameter

distribution were added to the noiseless data. For both systems random numbers with zero mean and variance equal to 0.04 were used. All input signals except the unit step change had an amplitude ± 0.5. The results show that the slow system, i.e., the system with large time constant, favored slow changing input signals, and the fast system favored fast input signals. However, for the two extreme input signals, impulsive signal and unit step change, the frequency components in the response data were either too weak to be recovered from the noise or too poor in quality to be useful.

Cumming (15) derived a conclusion from an underdamped system that test signal with a bandwidth (half power bandwidth) of about one half that of the system gave the best results. In Figure 27 it shows that by Cumming's suggestion the best input signal for System III would be SRTW with switching probability equal to 0.1. However, the results in TABLE VI shows that the overdamped System III favored input signals with bandwidth wider than Cumming's suggestion. This may be explained by the characteristics of system frequency response curve. For an underdamped system the frequency response curve is more distinctive at the relative low frequency region where the system shows the phenomena of resonance. Therefore, a signal of low bandwidth can be used for an underdamped system. Also the resonance of the system, which allows the use of input

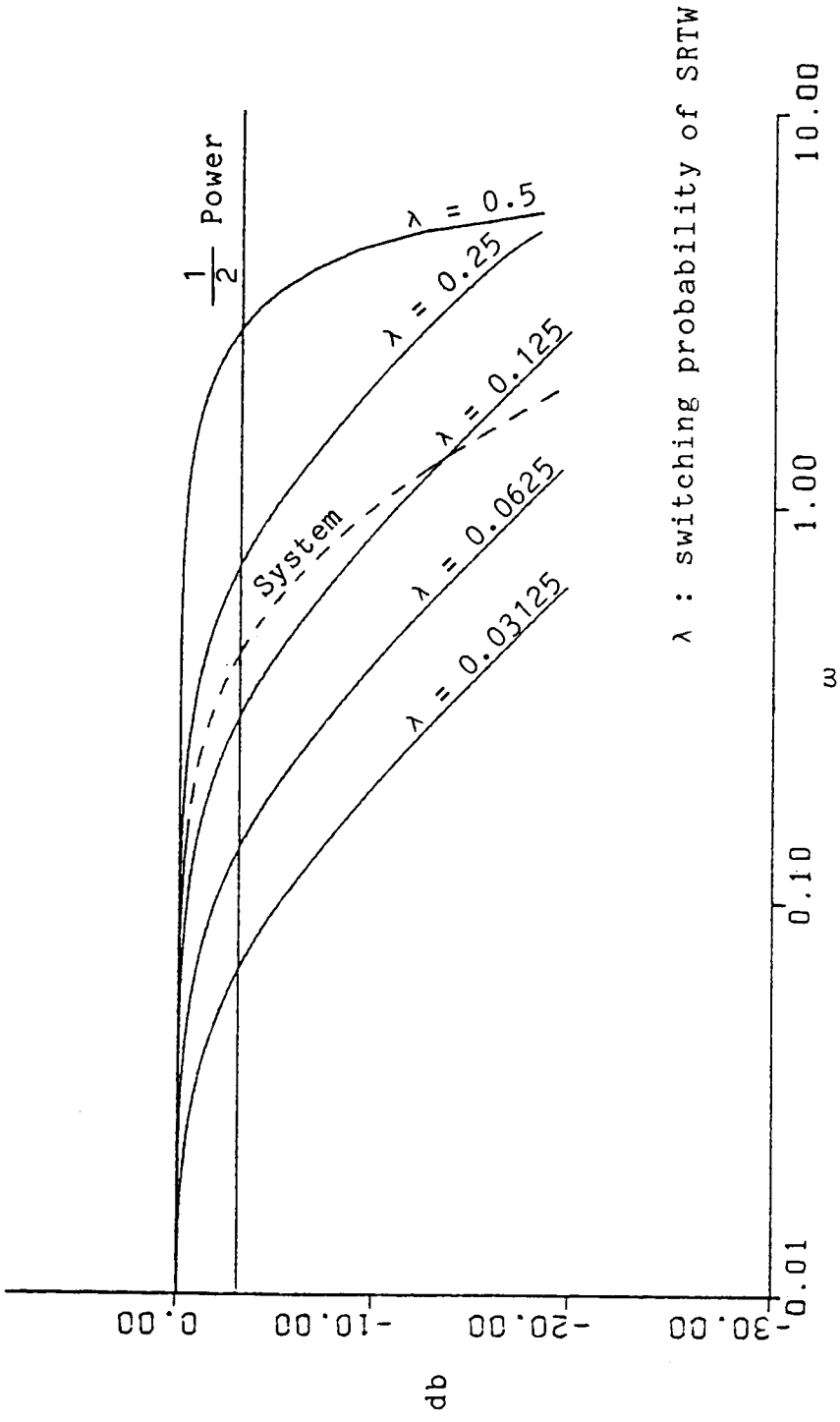


Figure 27. Power Density Spectrum of SRTW and Frequency Response of System III.

signals containing less power in the resonant frequency region, further reduces the bandwidth of the optimal input signal. For an overdamped system the frequency response curve is more distinctive in the corner frequency region. This accounts for the best result obtained from PRBS response data in TABLE VI, because PRBS contained more adequate power in the region of corner frequency than any other input signal (Figure 27).

As mentioned earlier, the parameters obtained from different input signal may be inconsistent if a model can not represent the system exactly. TABLE VIII and TABLE IX show some second order plus dead time models obtained from third order system response data. The results show that the parameters were biased because of an unaccounted characteristic of the system, and that influence of the third time constant increased as the time constant became close to the second largest time constant. The situation was even worse when an input signal contained more high frequency components (TABLE IX). In system modeling the examination of the parameter consistence provids a check of the feasibility of the model. If there is serious inconsistence, some important characteristics of the system must be missing in the model.

TABLE VIII. Second Order Models Obtained by Using Step Response Data of a Third Order System

System or Model	K	T1	T2	T3	θ
System V	1.500	7.139	1.261	0.100	2.300
Model 1	1.500	7.139	1.263	-	2.397
System VI	1.500	7.139	1.261	0.316	2.300
Model 2	1.500	7.138	1.286	-	2.594
System VII	1.500	7.139	1.261	1.000	2.300
Model 3	1.500	7.117	1.648	-	2.938

TABLE IX. Second Order Models Obtained by Using Impulsive Response Data of a Third Order System

System or Model	K	T1	T2	T3	θ
System IIX	1.500	7.139	1.261	0.100	2.300
Model 1a	1.492	7.080	1.277	-	2.393
Model 1b*	1.500	7.127	1.270	-	2.395
System IX	1.500	7.139	1.261	0.316	2.300
Model 2a	1.481	6.919	1.368	-	2.561
Model 2b*	1.500	7.051	1.352	-	2.563
System X	1.500	7.139	1.261	1.000	2.300
Model 3a	1.335	4.825	2.687	-	2.836
Model 3b*	1.500	5.816	2.529	-	2.789

* Model with fixed value of K

B. Controller Tuning

Once we obtain a model to represent a system, the parameters of a control algorithm can be tuned on a digital computer. The control algorithm used most often is the discrete velocity form of PID-controller.

$$m_n - m_{n-1} = K_c \left[(e_n - e_{n-1}) + \frac{T}{T_I} e_n + \frac{T_D}{T} (e_n - 2e_{n-1} + e_{n-2}) \right] \tag{40}$$

where K_c = controller gain,

T_I = integral time,

T_D = derivative time,

m_n = controller output, and

e_n = deviation of controlled variable from set point.

The procedures of controller tuning are

(i) Simulate the close loop control system on a digital computer

(ii) Apply a change to the load of the system or the set point of the controller

(iii) Optimize system response by adjusting the controller parameters

The most commonly used optimization criteria are

(i) Integral of the square error (ISE)

$$ISE = \int_0^\infty \left[e(t) \right]^2 dt \tag{41a}$$

or $$ISE = \sum_{n=0}^{\infty} e_n^2 \tag{41b}$$

(ii) Integral of the absolute value of the error (IAE)

$$IAE = \int_0^\infty |e(t)|\, dt \tag{42a}$$

$$\text{or} \quad IAE = \sum_{n=0}^{\infty} |e_n| \tag{42b}$$

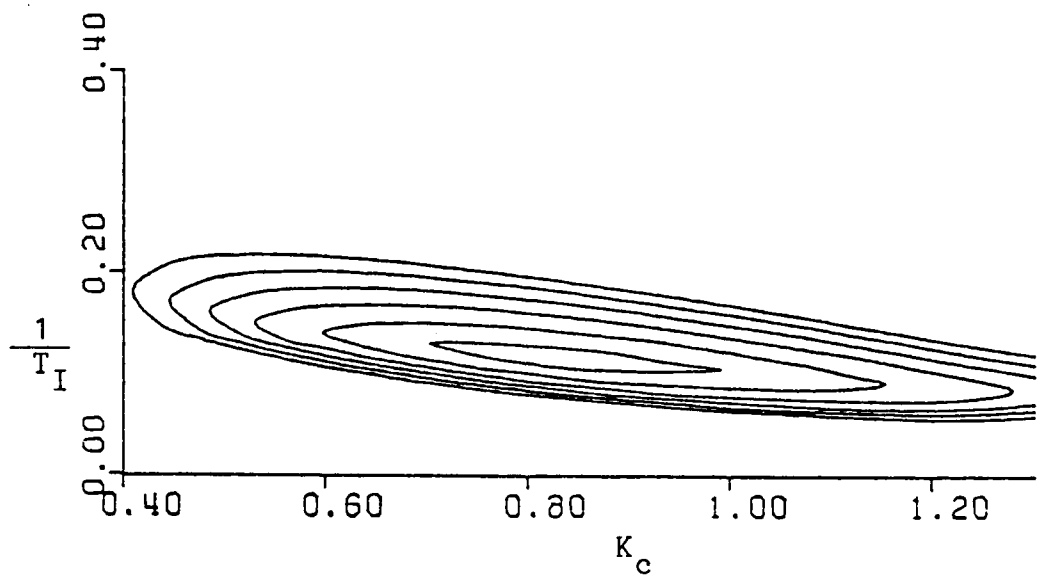
(iii) Integral of time multiplied by the absolute value of the error (ITAE)

$$ITAE = \int_0^\infty t\,|e(t)|\, dt \tag{43a}$$

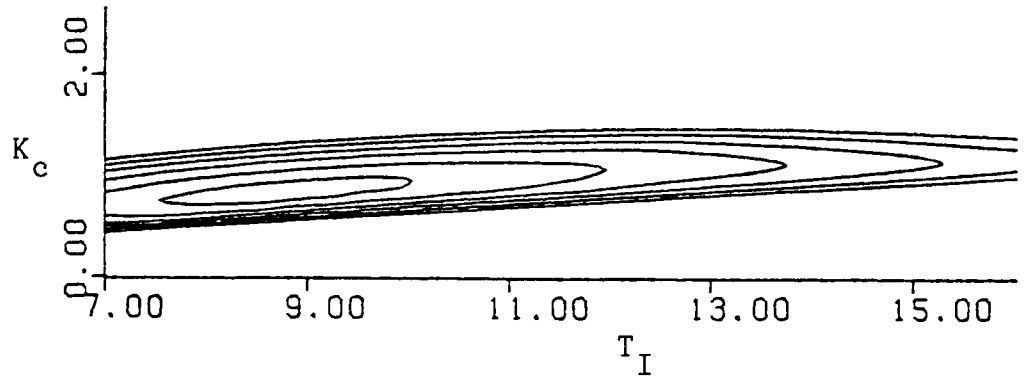
$$\text{or} \quad ITAE = \sum_{n=0}^{\infty} nT\,|e_n| \tag{43b}$$

For tuning of a PI-controller equation (40) can be used by setting $T_D = 0$. The optimization of the controller parameters can be made more efficient by selecting the proper search parameters. In Figure 28 and Figure 29, the contour surface of PI-controller tuning for set point change and load change shows that the best search parameters were K_c and K_c/T_I because the ridge was closely paralell to one of the coordinate axes. This again confirmed a previous conclusion that the best search parameters are those independently related to the characteristics of a system. In a PI-controller the proportional control action is determined by the value of K_c, and the integral control action is determined by the value of K_c/T_I rather than by the value of $1/T_I$ only.

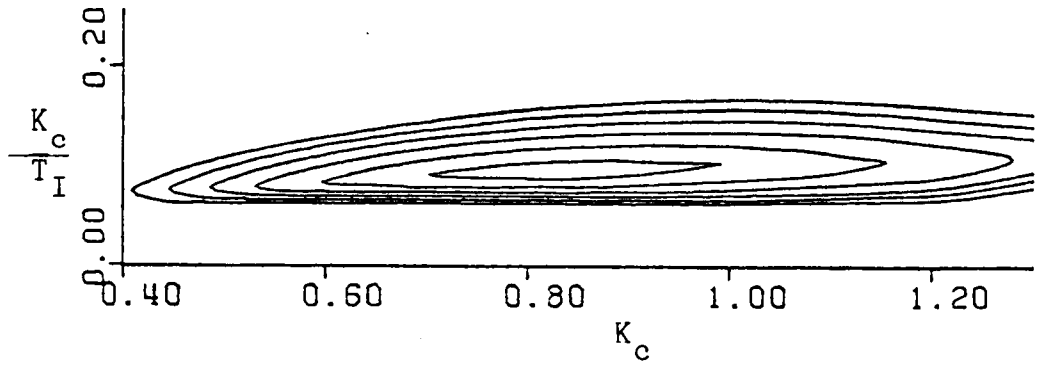
For PID-controller tuning, the reasonable choice of



(a)

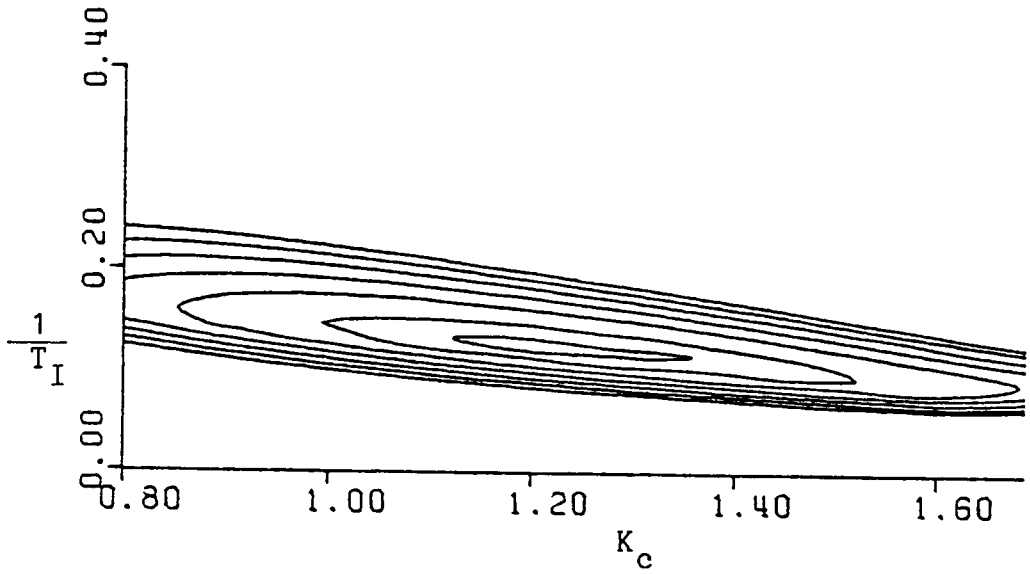


(b)

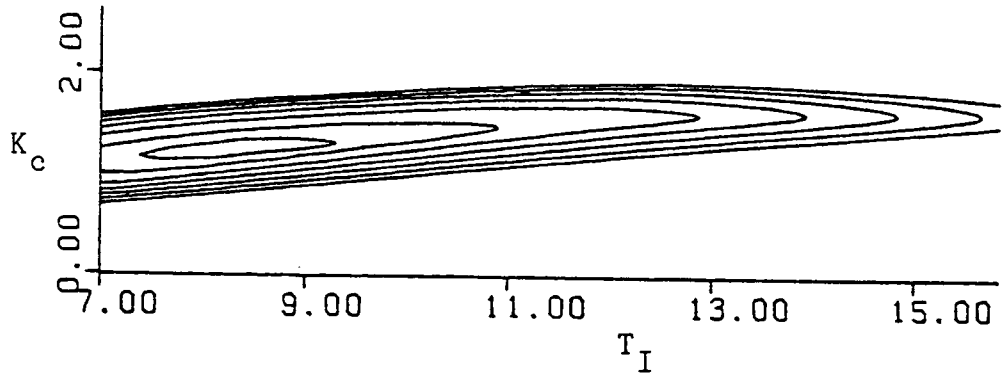


(c)

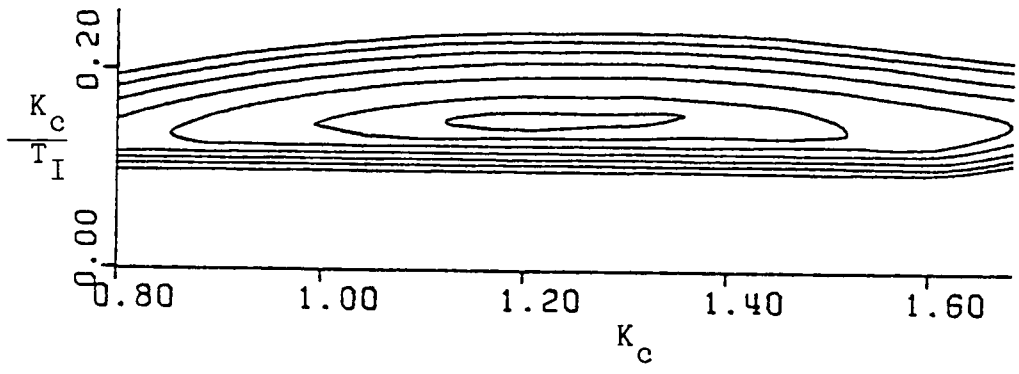
Figure 28. Contour Surface of PI-Controller Tuning for Set Point Change.



(a)



(b)

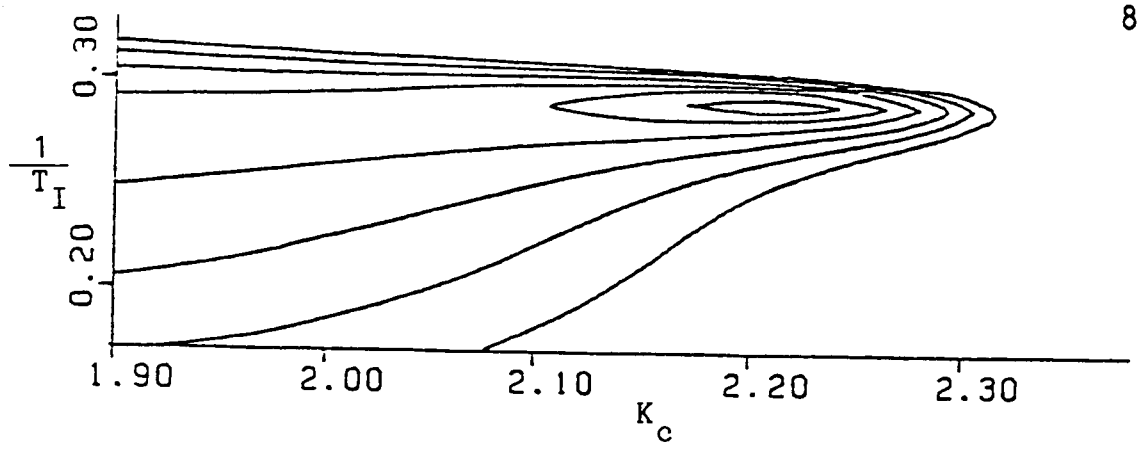


(c)

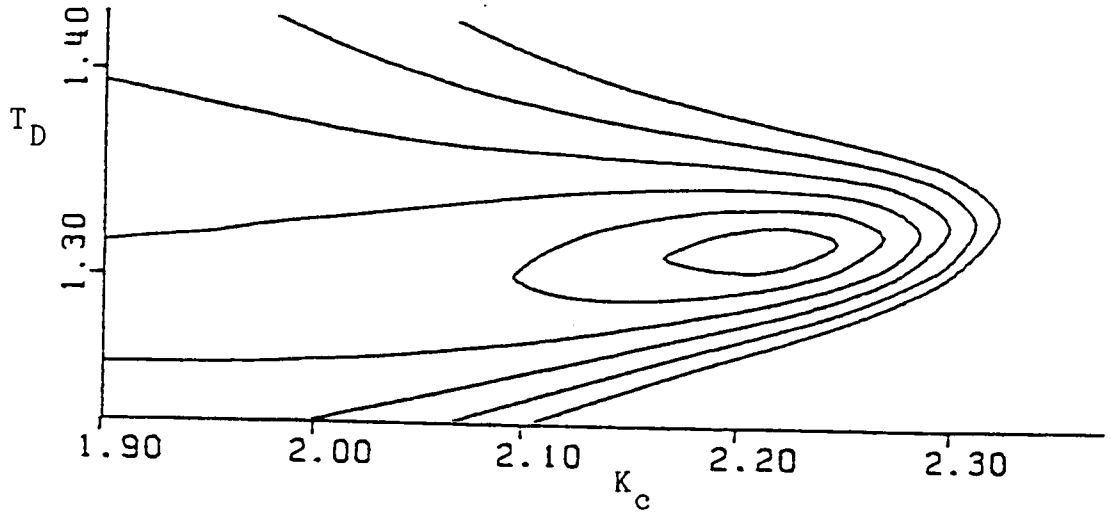
Figure 29. Contour Surface of PI-Controller Tuning for Load Change.

search parameters are K_c, K_c/T_I, and K_cT_D because they are related to differnet control actions separately. The projections of the contour surfaces of PID-controller tunings for different combinations of search parameters are shown in Figure 30 to Figure 31. For the tuning of set point change, interaction between $1/T_I$ and T_D caused an inclined ridge. Projection on the K_c/T_I and K_cT_D plane did not change the shape of the ridge except the scales of the axes. The interaction implies that $1/T_I$ and T_D are both related to a certain characteristic of the controller in some degree. As for tuning of load change, Figure 31(f) shows that the cost was almost independent of the value of T_D. These two facts all agreed with a theory that a PI-controller can almost take over the job of a PID-controller.

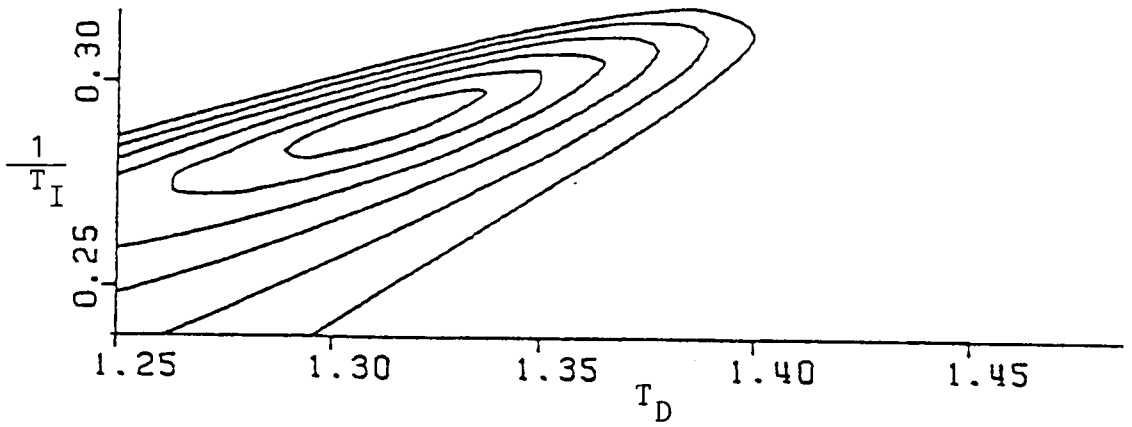
Since controller tuning is done on a computer, there is no problem of noise. The optimum is defined when the control algorithm and the optimization criteria have been selected. Selection of search parameters can only affect the efficiency of search but not the accuracy of the final results.



(a)

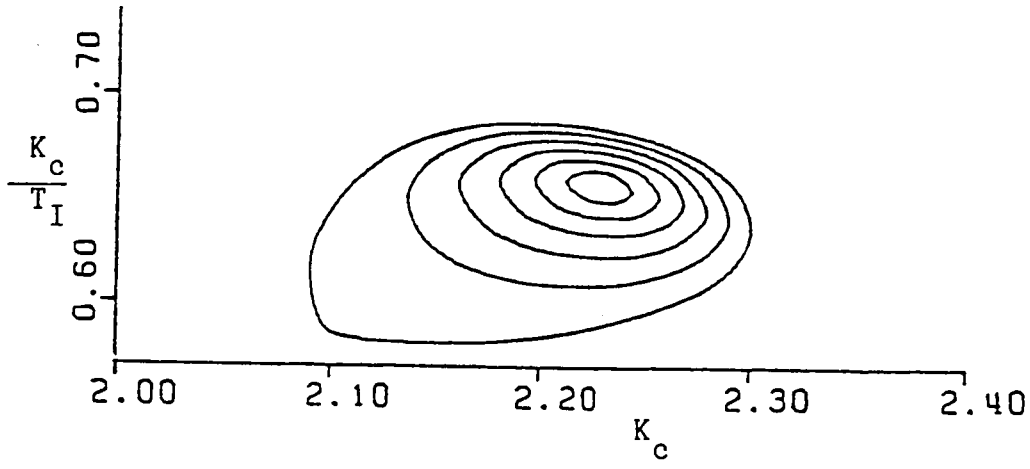


(b)

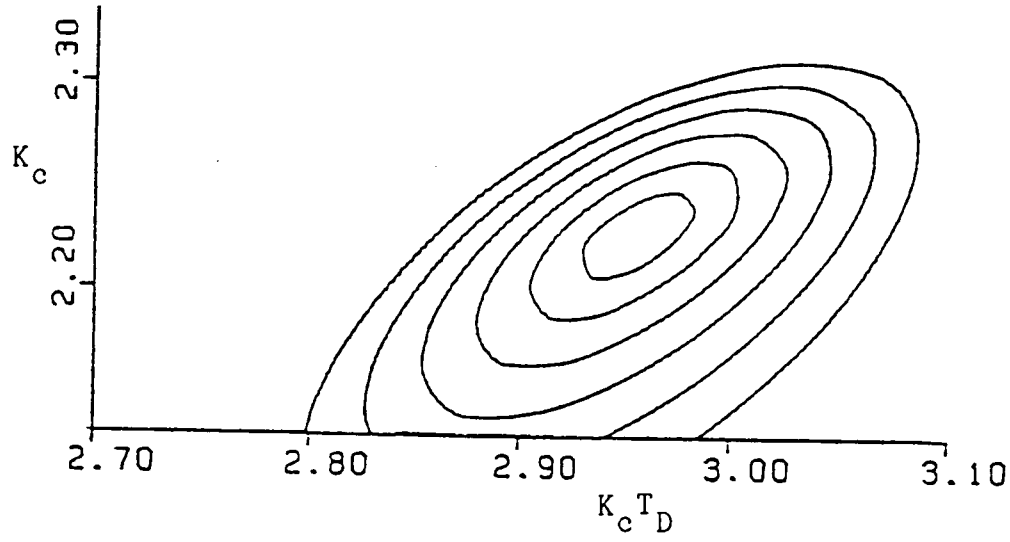


(c)

Figure 30. Contour Surface of PID-Controller Tuning for Set Point Change.



(d)



(e)

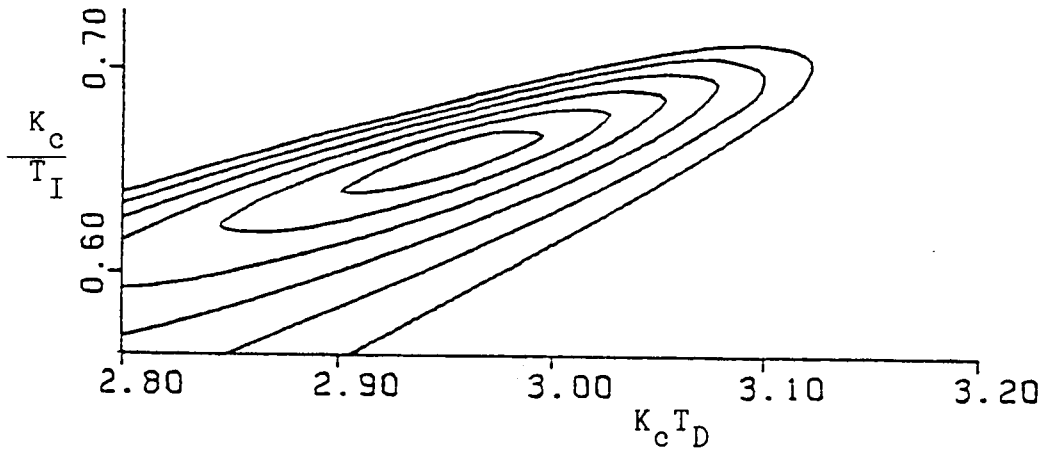
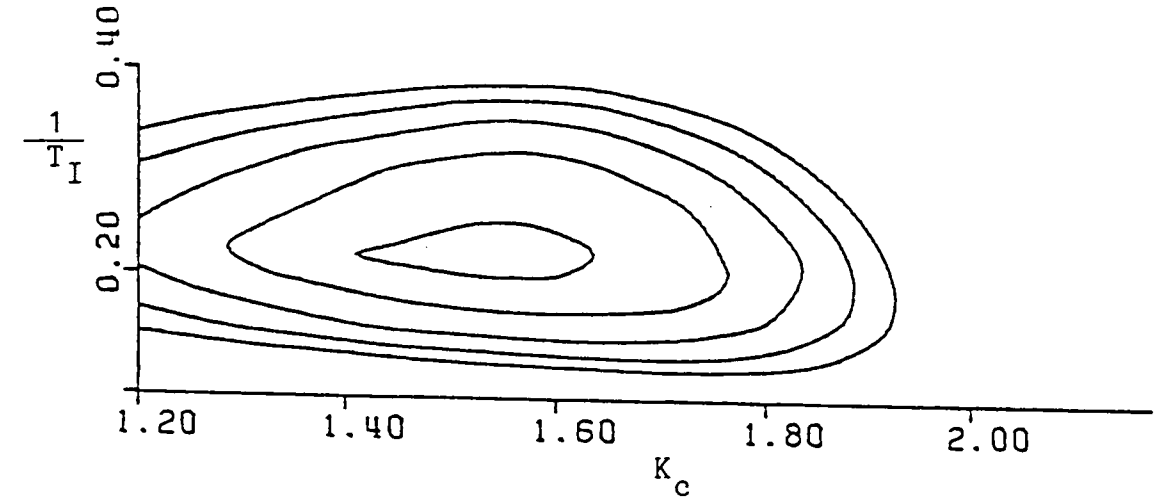
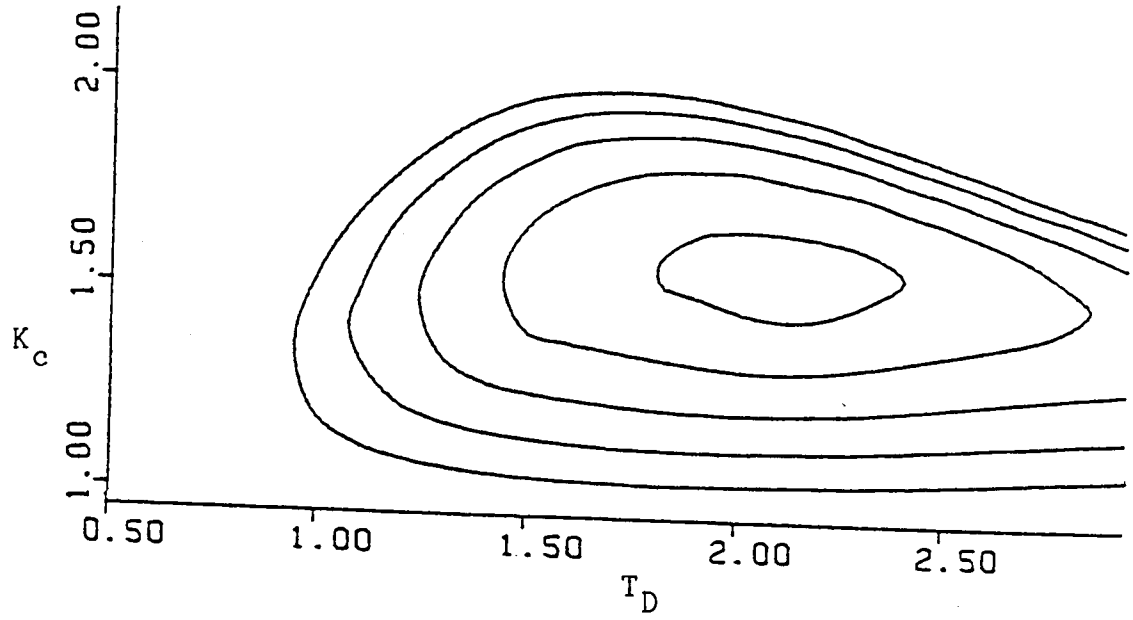


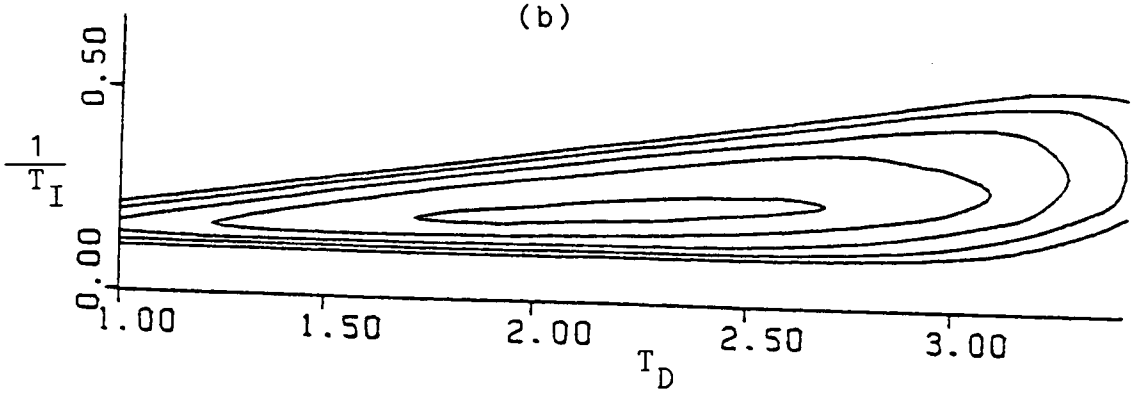
Figure 30. (Continued)



(a)

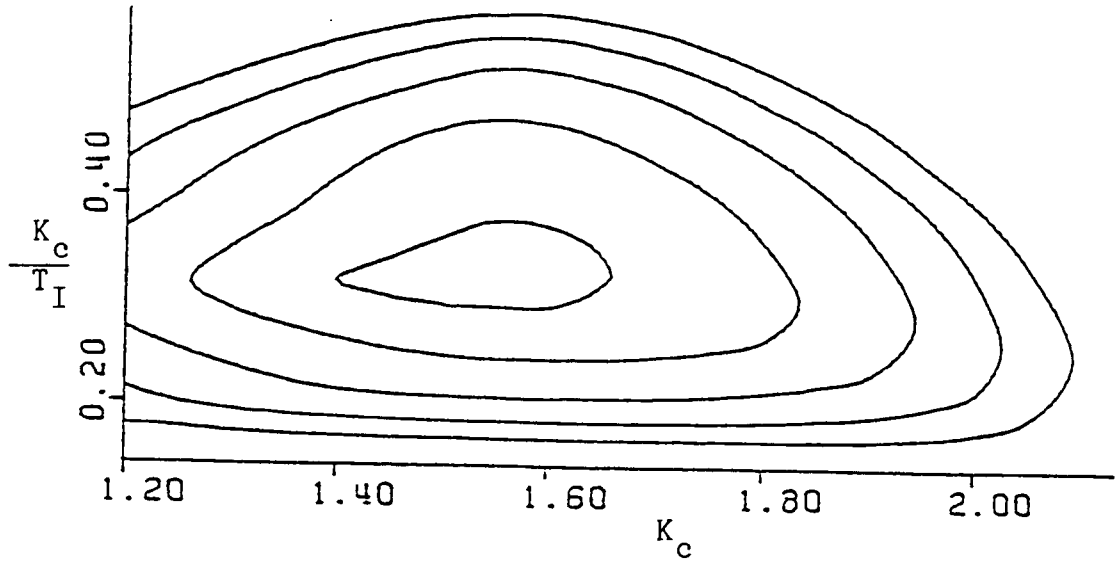


(b)

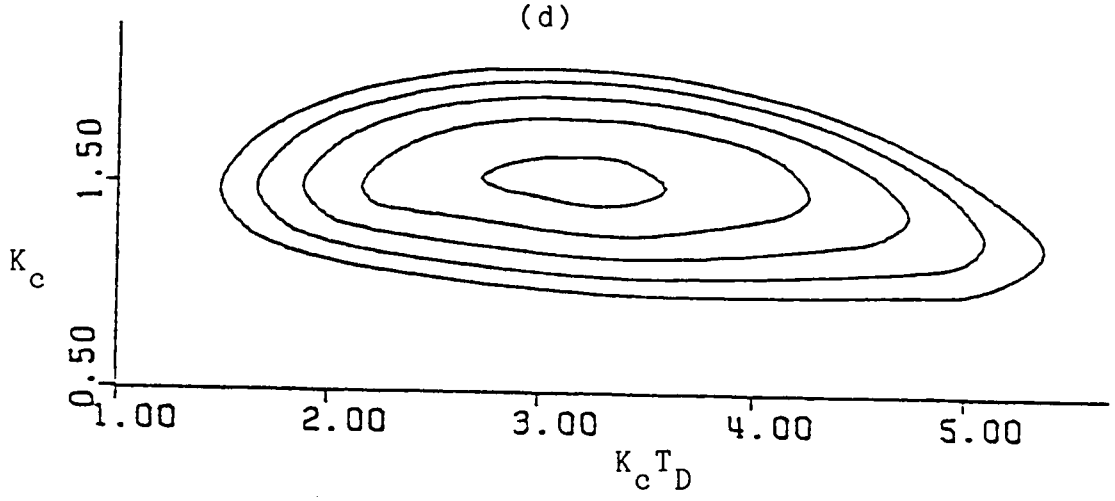


(c)

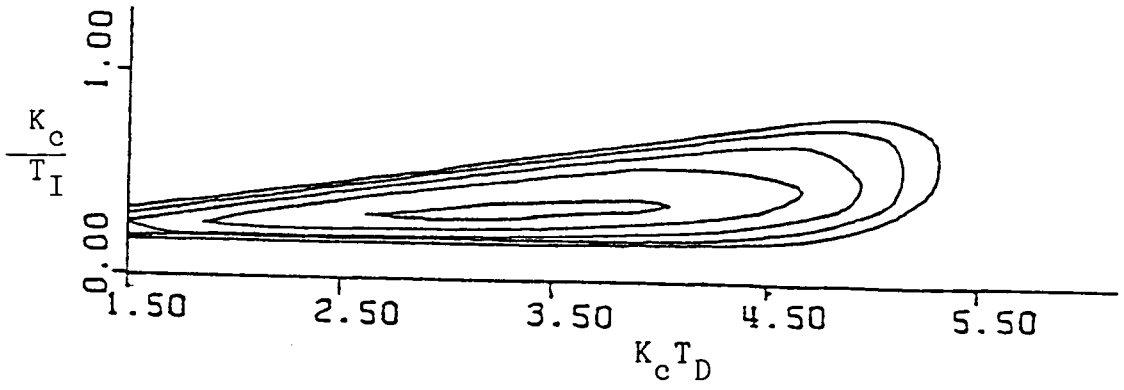
Figure 31. Contour Surface of PID-Controller Tuning for Load Change.



(d)



(e)



(f)

Figure 31. (Continued)

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

To guarantee the success of the experimental modeling discussed in this work, it requires:

(a) A feasible form of model

The form of model should be determined from a prelimilary study or knowledge of the system to be identified if possible. An inadequate model would usually lead to inconsistence of the model parameters when different input signals were used.

(b) An effective optimization technique

Primarily an optimization technique should be able to deal with ridges, which might exist on a contour surface. Most optimization techniques are most efficient if the contour surface is spherical. Ridge may sometimes be eliminated by selecting search parameters which are less interactive, and the contour surface may become more spherical by proper scaling of the coordinate system. Transformation of the contour surface, however, will not improve the accuracy of the final results.

(c) Experimental data of high quality

Basically system response of any type input signal contains complete information to specify the system

parameters even contents of some informations may be very weak. But in the presence of noises the information which can be extracted from a finite number of experimental data is decided by the quality of the data we collected. Then selection of input signal, which is closely related to contents of information and signal-to-noise ratio, becomes more important. However, the most rewarding solution is to eliminate the noises to the minimum level.

B. RECOMMENDATIONS

Recommendations for further study include the following:

(a) The modified pattern search has outstanding ability to search along extremely narrow and curved ridge, but it requires more function evaluations than other optimization techniques. The modified pattern search may be further modified or some other new techniques could be designed.

(b) The selection of input signal depends on the characteristics of a system and the power density spectrum of input signal. Their relation should be further investigated so that general criteria of optimal signal design and selection can be established.

LIST OF REFERENCES

LIST OF REFERENCES

1. Hooke, R., and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problem," J. Assoc. Computer Mach., Vol. 8, 1961, pp. 212-229.

2. Sage, A. P., and J. L. Melsa, System Identification, Academic Press, New York, 1972.

3. Graupe, D., Identification of Systems, Van Nostrand Reinhold Company, New York, 1972.

4. Hsia, T. C., System Identification, Lexington Books, Massachusetts, 1977.

5. Eykhoff, P., System Identification, Parameter and State Estimation, Wiley, London, 1974.

6. Strejc, V., "Approximate Determination of the Control Characteristics of an Aperiodic Response Process," Automatisme, Vol. 5, March 1960, pp. 109-111.

7. Thaler, G. J., and M. P. Pastel, Analysis and Design of Nonlinear Feedback Control System, McGraw-Hill, New York, 1967.

8. Fletcher, R., and C. M. Reeves, "Function Minimization by Conjugate Gradient," Computer J., Vol. 7, 1964, pp. 149-154.

9. Fletcher, R., and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer J., Vol. 6, 1963, pp. 163-168.

10. Fletcher, R., "A New Approach to Variable Metric Algorithms," Computer J., Vol. 13, 1970, pp. 317-322.

11. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or the Least Value of a Function," Computer J., Vol. 3, 1960.

12. Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives," Computer J., Vol. 7, 1964, pp. 155-162.

13. Nelder, J. A., and R. Mead, "A Simplex Method for Function Minimization," Computer J., Vol. 7, 1964, pp. 308-313.

14. Harrison, H. L., and R. Rangel, "Deadtime Term Simplifies Transfer Function," Control Engng., Vol. 14, No 6, 1967, pp. 107-108.

15. Cumming, I. G., "Frequency of Input Signal in Identification," Second IFAC Symp. Identification and Proccess Parameter Estimation, Prague, Paper 7.8, 1970.

16. Goodwin, G. C., J. C. Murdock, and R. L. Rayne, "Optimal Test Signal Design For Linear Single-input Single-output System Identificaion," Int. J. Control, Vol. 17, No. 1, 1973, pp. 45-55.

17. Moore, C. F., C. L. Smith, and P. W. Murril, "Formulating the Nonlinear Least-squares Model Regression for Fast On-line Analysis," 64th National Meeting A.I.Ch.E., New Orleans, March 1969.

18. Tou, J. T., Digital Sampled-data Control System, McGraw Hill, New York, 1959.

19. Himmelbau, D. M., Applied Nonlinear Programming, McGraw-Hill, New York, 1972.

APPENDICES

APPENDIX A

DERIVATION OF SECOND ORDER PLUS DEAD TIME SYSTEM EQUATION IN Z-DOMAIN

A second order plus dead time system as shown in Figure 2 (page 6) was used as the model in this work. The equation in z-domain and the difference equation for this model can be derived as follows.

(a) $\zeta > 1$

$$G(z) = Z\left\{\frac{(1-e^{-sT})Ke^{-\theta s}}{s(As+1)(Bs+1)}\right\} \qquad (A-1)$$

where $A = \dfrac{\tau}{\zeta - \sqrt{\zeta^2-1}}$ and $B = \dfrac{\tau}{\zeta + \sqrt{\zeta^2-1}}$

Let $\theta = NT+(1-m)T$. Then

$$G(z) = Z\left\{\frac{Ke^{-NTs}(1-e^{-Ts})e^{-(1-m)T}}{s(As+1)(Bs+1)}\right\}$$

$$= Kz^{-N}(1-z^{-1})\left\{\frac{z^{-1}}{1-z^{-1}} + \frac{(\frac{A}{B-A})e^{-mT/A}z^{-1}}{1-e^{-T/A}z^{-1}} - \frac{(\frac{B}{B-A})e^{-mT/B}z^{-1}}{1-e^{-T/B}z^{-1}}\right\}$$

$$= \frac{z^{-N}(b_1z^{-1}+b_2z^{-2}+b_3z^{-3})}{1+a_1z^{-1}+a_2z^{-2}} \qquad (A-2)$$

where $a_1 = -e^{-T/A}-e^{-T/B}$

$$a_2 = e^{-T/A}e^{-T/B}$$

$$b_1 = K\left[1+\left(\frac{A}{B-A}\right)e^{-mT/A}-\left(\frac{B}{B-A}\right)e^{-mT/B}\right]$$

$$b_2 = -K\left[e^{-T/A}+e^{-T/B}+\left(\frac{A}{B-A}\right)\left(e^{-T/B}+1\right)e^{-mT/A} -\left(\frac{B}{B-A}\right)\left(e^{-T/B}+1\right)e^{-mT/B}\right]$$

$$b_3 = K\left[e^{-T/A}e^{-T/B}+\left(\frac{A}{B-A}\right)e^{-T/B}e^{-mT/A} -\left(\frac{B}{B-A}\right)e^{-T/A}e^{-mT/B}\right]$$

(b) $\zeta = 1$

$$G(z) = Z\left\{\frac{(1-e^{-sT})Ke^{-\theta s}}{s(\tau s+1)^2}\right\}$$

$$= Z\left\{\frac{(1-e^{-sT})Ke^{-(N+1-m)Ts}}{s(\tau s+1)^2}\right\}$$

$$= Kz^{-N}(1-z^{-1})\left[\frac{z^{-1}}{1-z^{-1}} - \frac{(1-mT/\tau)e^{-mT/\tau}z^{-1}}{1-e^{-T/\tau}z^{-1}} + \frac{(T/\tau)e^{-mT/\tau}e^{-T/\tau}z^{-2}}{(1-e^{-T/\tau}z^{-1})^2}\right]$$

$$= \frac{z^{-N}(b_1z^{-1}+b_2z^{-2}+b_3z^{-3})}{1+a_1z^{-1}+a_2z^{-2}} \qquad \text{(A-3)}$$

where $a_1 = -2e^{-T/\tau}$

$$a_2 = e^{-2T/\tau}$$

$$b_1 = K\left[1-(1+\frac{mT}{\tau})e^{-mT/\tau}\right]$$

$$b_2 = K\left[(1+\frac{mT}{\tau})(1+e^{-T/\tau})e^{-mT/\tau}-2e^{-T/\tau} - \frac{T}{\tau}\, e^{-T/\tau}e^{-mT/\tau}\right]$$

$$b_3 = K\left[e^{-2T/\tau}-(1+\frac{mT}{\tau}-\frac{T}{\tau})\, e^{-T/\tau}e^{-mT/\tau}\right]$$

(c) $\zeta < 1$

$$G(z) = Z\left\{\frac{(1-e^{-sT})Ke^{-\theta s}}{s(\tau^2 s^2+2\tau\zeta s+1)}\right\}$$

$$= Z\left\{\frac{(1-e^{-sT})Ke^{-(N+1-m)Ts}}{(\tau\sqrt{1-\zeta^2})\ (s+\zeta/\tau)^2+W^2}\right\}$$

$$= Kz^{-N}(1-z^{-1})\left\{\frac{z^{-1}}{1-z^{-1}} - \frac{e^{-\zeta mT/\tau}\sec\phi\ \cos(WmT+\phi)z^{-1}-}{1-2z^{-1}e^{-\zeta T/\tau}\cos WT+e^{-2\zeta T/\tau}z^{-2}} \right.$$

$$\left. \frac{e^{-\zeta T/\tau}\cos((1-m)WT-\phi)z^{-2}}{}\right\}$$

$$= \frac{z^{-N}(b_1z^{-1}+b_2z^{-2}+b_3z^{-3})}{1+a_1z^{-1}+a_2z^{-2}} \qquad (A-4)$$

where $W = \sqrt{1-\zeta^2}/\tau$

$$\phi = \tan^{-1}(-\zeta/\tau W)$$

$$a_1 = -2e^{-\zeta T/\tau}\cos(WT)$$

$$a_2 = e^{-2\zeta mT/\tau}$$

$$b_1 = K\left[1-e^{-\zeta mT/\tau}\cos(WmT+\phi)/\cos\phi\right]$$

$$b_2 = K\left[e^{-\zeta T/\tau}e^{-\zeta mT/\tau}\frac{\cos((1-m)WT-\phi)}{\cos\phi}\right.$$

$$\left.+e^{-\zeta mT/\tau}\frac{\cos(WmT+\phi)}{\cos\phi} - 2e^{-\zeta T/\tau}\cos WT\right]$$

$$b_3 = K\left[e^{-2\zeta T/\tau}-e^{-\zeta T/\tau}e^{-\zeta mT/\tau}\frac{\cos((1-m)WT-\phi)}{\cos\phi}\right]$$

Since $$G(z) = \frac{X(z)}{U(z)}$$

$$X(z)(1+a_1z^{-1}+a_2z^{-2}) = U(z)z^{-N}(b_1z^{-1}+b_2z^{-2}+b_3z^{-3}) \qquad (A-5)$$

In form of difference equation,

$$x_i = b_1u_{i-(N+1)}+b_2u_{i-(N+2)}+b_3u_{i-(N+3)}$$

$$-a_1x_{i-1}-a_2x_{i-2} \qquad (A-6)$$

APPENDIX B

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      SIMULATION OF SECOND ORDER PLUS DEAD TIME PROCESS              C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        DIMENSION T(500),U(500),X(500)
        DIMENSION SU(500),SH(500),TEL(500)
C-------------------------------------------------------------
C       GAIN : PROCESS GAIN
C       TAU  : TIME CONSTANT
C       ZET  : DAMPING RATIO
C       DEL  : DEAD TIME
C       N    : NUMBER OF DATA
C       ST   : SAMPLING TIME
C       T(I) : TIME
C       U(I) : INPUT SIGNAL
C       X(I) : SYSTEM RESPONSE
C       ISIG : SELECTION OF INPUT SIGNAL
C              ISIG=0 FOR UNIT STEP CHANGE
C              ISIG=1 FOR SERIES OF IMPULSIVE SIGNAL
C              ISIG=2 FOR PRBS (PERIOD OF PRBS = 2**NN-1)
C              ISIG=3 FOR SRTW (CALL SUBROUTINE SRTW)
C               * DEN : SWITCHING PROBABILITY OF SRTW
C-------------------------------------------------------------
        N=100
        ST=1.0
        ISIG=0
        NN=6
        DEN=0.1
C------SYSTEM PARAMETERS
        GAIN=1.5
        TAU=3.0
        ZET=1.5
        DEL=2.3
C---------------------
        DD=DEL/ST
        NDEL=IFIX(DD)
        NDEL1=NDEL+1
        NDEL2=NDEL+2
        DELI=NDEL+1
        DM=((DELI*ST)-DEL)/ST
C------INITIAL CONDITIONS
        USS=1.0
        U(1)=USS
        T(1)=0.0
        DO 10 I=2,15
        U(I)=USS
        T(I)=T(I-1)+ST
   10   CONTINUE
```

```
C------SELECTION OF INPUT SIGNAL
       ISIG=3
       IF (ISIG.EQ.3) GO TO 27
       IF (ISIG.EQ.2) GO TO 20
       IF (ISIG.EQ.1) GO TO 15
C----- INPUT SIGNAL : UNIT STEP CHANGE
       DO 12 I=16,N
       U(I)=USS+1.0
       T(I)=T(I-1)+ST
  12   CONTINUE
       GO TO 30
C----- INPUT SIGNAL : SERIES OF IMPULSE
  15   INP=1
       DO 19 I=16,N
       T(I)=T(I-1)+ST
       IF (INP.GT.15) GO TO 16
       U(I)=USS
       GO TO 18
  16   IF (INP.GT.16) GO TO 17
       U(I)=0.5+USS
       GO TO 18
  17   IF (INP.GT.16) INP=0
       U(I)=-0.5+USS
  18   INP=INP+1
  19   CONTINUE
       GO TO 30
C----- INPUT SIGNAL : PRBS
  20   NDP=N
       DO 21 I=1,NN
       SH(I)=1.0
  21   CONTINUE
       DO 24 I=1,NDP
       S=SH(NN)+SH(NN-1)
       IF (S.GT.1.5) S=0.0
       DO 23 J=2,NN
       JM=NN+1-J
       SH(JM+1)=SH(JM)
  23   CONTINUE
       SH(1)=S
       SU(I)=S-0.5
  24   CONTINUE
       DO 25 I=16,N
       U(I)=USS+SU(I-1)
       T(I)=T(I-1)+ST
  25   CONTINUE
       GO TO 30
C----- INPUT SIGNAL : SRTW
  27   CALL SRTW (N,TEL,DEN)
       DO 29 I=16,N
       U(I)=TEL(I)+USS
       T(I)=T(I-1)+ST
  29   CONTINUE
```

```
C------COEFFICIENTS OF DIFFERENCE EQUATION
   30   IF (ZET-1.0) 35,40,45
C------UNDERDAMPED SYSTEM
   35   A=ZET/TAU
        W=SQRT(1.0-ZET**2)/TAU
        SIT=ATAN(-A/W)
        EM=EXP(-A*DM*ST)
        ET=EXP(-A*ST)
        ALF1=-2.0*ET*COS(W*ST)
        ALF2=ET**2
        BLT1=EM*COS(W*DM*ST+SIT)/COS(SIT)
        BLT2=-EM*ET*COS((1.0-DM)*W*ST-SIT)/COS(SIT)
        A1=-ALF1
        A2=-ALF2
        B1=GAIN*(1-BLT1)
        B2=GAIN*(ALF1-BLT2+BLT1)
        B3=GAIN*(ALF2+BLT2)
        GO TO 50
C------CRITICALLY DAMPED SYSTEM
   40   ET=EXP(-ST/TAU)
        EM=EXP(-DM*ST/TAU)
        A1=2.0*ET
        A2=-ET**2
        B1=GAIN*(1.-(1.+DM*ST/TAU)*EM)
        B2=GAIN*((1.+DM*ST/TAU)*EM*(1.+ET)-
     C    2.*ET-ST*ET*EM/TAU)
        B3=GAIN*(ET**2-(1.+DM*ST/TAU)*EM*ET+ST*ET*EM/TAU)
        GO TO 50
C------OVERDAMPED SYSTEM
   45   RA=1.0
        RB=2.0*ZET/TAU
        RC=1.0/(TAU**2)
        R1=(-RB+SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
        T1=-1.0/R1
        R2=(-RB-SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
        T2=-1.0/R2
        E1=EXP(-ST/T1)
        E2=EXP(-ST/T2)
        A1=(E1+E2)
        A2=-E1*E2
        EM1=EXP(-DM*ST/T1)
        EM2=EXP(-DM*ST/T2)
        ALF=T1/(T2-T1)
        BLT=T2/(T2-T1)
        B1=GAIN*(1.0+ALF*EM1-BLT*EM2)
        B2=-GAIN*(E1+E2+(E2+1.)*ALF*EM1-(E1+1.0)*BLT*EM2)
        B3=GAIN*(E1*E2+E2*ALF*EM1-E1*BLT*EM2)
C------PROCESS SIMULATION
   50   XSS=USS*GAIN
        DO 52 I=1,NDEL1
   52   X(I)=XSS
```

```
      DO 70 I=NDEL2,N
      K=I-NDEL1
      IF (I.GE.2) XS=A1*(X(I-1)-XSS)
      IF (I.GE.3) XS=XS+A2*(X(I-2)-XSS)
      IF (K.GE.1) US=B1*(U(K)-USS)
      IF (K.GE.2) US=US+B2*(U(K-1)-USS)
      IF (K.GE.3) US=US+B3*(U(K-2)-USS)
      X(I)=XSS+US+XS
   70 CONTINUE
      DO 95 I=1,N
      WRITE (5,90) T(I),U(I),X(I)
   90 FORMAT (1X,2F8.4,F12.7)
   95 CONTINUE
      STOP
      END
```

```
C-----------------------------------------------
C       THIS SUBROUTINE GENERATES SRTW
C-----------------------------------------------
      SUBROUTINE SRTW (N,TEL,DEN)
      DIMENSION TEL(500), TF(500), R1(500)
C------START OF RANDOM NUMBER
C       RAN(.) = RANDOM NUMBER IN UNIFORM DISTRIBUTION
      IRAN=N
      DO 771 LL=1,800
      AA=RAN(8)
  771 CONTINUE
      DO 545 I=1,IRAN
      R1(I)=RAN(8)
  545 CONTINUE
C----- GENERATION OF SRTW
      NF=1
      DO 548 I=1,IRAN
      TF(I)=-ALOG(R1(I))/DEN
  548 CONTINUE
      TEL(1)=0.5
      TT=0.0
      ST=1.0
      I=2
  550 TT=TT+ST
      IF (TF(NF)-TT) 552,556,558
  552 TEL(I)=-TEL(I-1)
      I=I+1
  553 TT=TT-TF(NF)
      NF=NF+1
      IF (TT.LT.TF(NF)) GO TO 560
```

```
      GO TO 553
556   TEL(I)=-TEL(I-1)
      I=I+1
      NF=NF+1
      TT=0.0
      GO TO 560
558   TEL(I)=TEL(I-1)
      I=I+1
560   IF (I.LE.N) GO TO 550
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      MAIN PROGRAM TO INITIALIZE SEARCH OF MODEL PARAMETER           C
C                                                                     C
C-----------------------------------------------------------------    C
C     (1) SUBROUTINE PROC WILL BE USED TO EVALUATE THE OBJECTIVE      C
C         FUNCTION VALUE FOR A GIVEN SET OF PARAMETER.                C
C     (2) SUBROUTINE BOUNDS WILL BE USED TO CHECK THE CONSTRAINT      C
C         EQUATIONS.                                                  C
C     (3) SUBROUTINE SEARCH, PATERN, AND ROTATE WILL BE USED TO       C
C         PERFORM THE MODIFIED PATTERN SEARCH.                        C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION STEP(50),P(50)
      COMMON IFLAG,N,ST,T(100),U(100),X(100),USS
C-----------------------------------------------------------------------
C       N    : NUMBER OF DATA
C       ST   : SAMPLING PERIOD
C       NP   : NUMBER OF SEARCH PARAMETERS
C       SKLM : PRECISION OF FINAL PARAMETERS
C       SKLR : SCALE REDUCING RATE (SKLR > 1.0. USE SMALL SKLR FOR
C              OBJECTION FUNCTION WITH NARROW RIDGE. FOR GENERAL
C              CASES, USE SKLR = 100.0)
C       IANG : PRECISION OF ROTATION ANGLES ( 150/(5**IANG) DEGREES  )
C-----------------------------------------------------------------------
C----- DATA INPUT
      N=75
      ST=1.0
      DO 20 I=1,N
      READ (20,10) T(I),U(I),X(I)
   10 FORMAT (1X,2F8.4,F12.7)
   20 CONTINUE
      USS=U(1)
C----- INITIAL GUESS OF PARAMETERS
      P(1)=2.0
      P(2)=2.0
      P(3)=2.0
      P(4)=2.0
C----- START MODEL SEARCH
      NP=4
      IANG=4
      SKLM=0.0001
      SKLR=30.0
      IFLAG=0
      IO=3
      CALL SEARCH (NP,P,SKLM,SKLR,IANG,IO)
C----- LIST MODEL RESPONSE
      IFALG=1
      CALL PROC(P,COST)
      STOP
      END
```

```
C-------------------------------------------------------------
C       THIS SUBROUTINE EVALUATES THE OBJECTIVE FUNCTION
C       VALUE FOR THE MODELING OF SECOND ORDER PLUS DEAD
C       TIME PROCESS.
C-------------------------------------------------------------
        SUBROUTINE PROC(P,COST)
        DIMENSION P(50),XX(100)
        COMMON IFLAG,N,ST,TM(100),U(100),X(100),USS
C----- PROCESS PARAMETERS
        GAIN=P(1)
        TAU=P(2)
        ZET=P(3)/TAU
        DEL=P(4)
C-----------------------------------
        DD=DEL/ST
        NDEL=IFIX(DD)
        NDEL1=NDEL+1
        NDEL2=NDEL+2
        DELI=NDEL+1
        DM=((DELI*ST)-DEL)/ST
        IF (ZET-1.0) 35,40,45
C------UNDERDAMPED SYSTEM
   35   A=ZET/TAU
        W=SQRT(1.0-ZET**2)/TAU
        SIT=ATAN(-A/W)
        EM=EXP(-A*DM*ST)
        ET=EXP(-A*ST)
        ALF1=-2.0*ET*COS(W*ST)
        ALF2=ET**2
        BLT1=EM*COS(W*DM*ST+SIT)/COS(SIT)
        BLT2=-EM*ET*COS((1.0-DM)*W*ST-SIT)/COS(SIT)
        A1=-ALF1
        A2=-ALF2
        B1=GAIN*(1-BLT1)
        B2=GAIN*(ALF1-BLT2+BLT1)
        B3=GAIN*(ALF2+BLT2)
        GO TO 50
C------CRITICALLY DAMPED SYSTEM
   40   ET=EXP(-ST/TAU)
        EM=EXP(-DM*ST/TAU)
        A1=2.0*ET
        A2=-ET**2
        B1=GAIN*(1.-(1.+DM*ST/TAU)*EM)
        B2=GAIN*((1.+DM*ST/TAU)*EM*(1.+ET)-
     C    2.*ET-ST*ET*EM/TAU)
        B3=GAIN*(ET**2-(1.+DM*ST/TAU)*EM*ET+ST*ET*EM/TAU)
        GO TO 50
C------OVERDAMPED SYSTEM
   45   RA=1.0
        RB=2.0*ZET/TAU
```

```
      RC=1.0/(TAU**2)
      R1=(-RB+SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
      T1=-1.0/R1
      R2=(-RB-SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
      T2=-1.0/R2
      E1=EXP(-ST/T1)
      E2=EXP(-ST/T2)
      A1=(E1+E2)
      A2=-E1*E2
      EM1=EXP(-DM*ST/T1)
      EM2=EXP(-DM*ST/T2)
      ALF=T1/(T2-T1)
      BLT=T2/(T2-T1)
      B1=GAIN*(1.0+ALF*EM1-BLT*EM2)
      B2=-GAIN*(E1+E2+(E2+1.)*ALF*EM1-(E1+1.0)*BLT*EM2)
      B3=GAIN*(E1*E2+E2*ALF*EM1-E1*BLT*EM2)
C----- PROCESS SIMULATION
   50 XSS=USS*GAIN
      DO 52 I=1,NDEL1
   52 XX(I)=XSS
      DO 70 I=NDEL2,N
      K=I-NDEL1
      IF (I.GE.2) XS=A1*(XX(I-1)-XSS)
      IF (I.GE.3) XS=XS+A2*(XX(I-2)-XSS)
      IF (K.GE.1) US=B1*(U(K)-USS)
      IF (K.GE.2) US=US+B2*(U(K-1)-USS)
      IF (K.GE.3) US=US+B3*(U(K-2)-USS)
      XX(I)=XSS+US+XS
   70 CONTINUE
C------MODEL ERROR DETERMINATION
      COST=0.0
      DO 300 I=1,N
      COST=COST+ABS(XX(I)-X(I))
  300 CONTINUE
  350 IF (IFLAG.EQ.0) RETURN
C------PROCESS VS MODEL RESPONSE
      WRITE (5,105) GAIN,TAU,ZET,NDEL,P(1),P(2),P(3),P(4)
  105 FORMAT (1X,8H OPTIMAL,3F7.3,I6/,1X,5HP(I)=,4F7.3)
      DO 400 I=1,N
      WRITE (5,106) TM(I),X(I),U(I),XX(I)
  106 FORMAT (4F10.3)
  400 CONTINUE
      RETURN
      END
```

```
C-----------------------------------------------------
C       THIS SUBROUTINE CHECKS THE CONSTRAINT
C       EQUATIONS FOR THE MODEL PARAMETERS.
C-----------------------------------------------------
        SUBROUTINE BOUNDS(P,IOUT)
        DIMENSION P(50)
   10   IOUT=0
        IF (P(1).LE.0.0) IOUT=1
        IF (P(2).LE.0.0) IOUT=1
        IF (P(3).LE.0.0) IOUT=1
        IF (P(4).LT.0.0) IOUT=1
   20   RETURN
        END



C-----------------------------------------------------
C       THIS SUBROUTINE DIRECTS THE STRATEGY OF
C       THE MODIFIED PATTERN SEARCH METHOD
C-----------------------------------------------------
        SUBROUTINE SEARCH (NP,P,SKLM,SKLR,IANG,IO)
        DIMENSION P(50),T(50),STEP(50),STP(50),ANGL(50)
        DIMENSION SCALE(50),DP(50),CC(50)
        IGM=1
        IR=0
        IFAIL=0
        IEND=0
        NPASS1=3
        NPASS2=IANG
        NA=NP*(NP-1)/2
        IAQ=0
        SKL=0.02
        DO 10 I=1,NA
   10   ANGL(I)=0.0
        DO 15 I=1,NP
        SCALE(I)=0.1
        STEP(I)=0.1
        DP(I)=0.0
   15   STP(I)=1.0
        CALL BOUNDS (P,IOUT)
        IF (IOUT.EQ.1) GO TO 300
        CALL PROC (P,C1)
        CBS=C1
        GO TO 80
C----- COORDINATE SCALING AND STEP SIZE DETERMINATION
   20   IF (IR.EQ.0.OR.IGM.EQ.1) GO TO 22
        CALL ROTATE (NP,DP,ANGL,1)
   22   IF (IGM.EQ.1) CCMAX=0.0
```

```
      DO 75 L=1,2
      IF (IGM.EQ.2.AND.L.EQ.1) GO TO 75
      IF (IEND.EQ.1.AND.L.EQ.2) GO TO 75
      J2=L
      DO 70 J=1,J2
      RDP=1.0+1.0/(5.**J)
      DO 69 I=1,NP
      NN=0
      K1=I
      K2=I
      IF (IGM.EQ.1) GO TO 24
      K1=1
      K2=NP
24    DO 26 K=K1,K2
      DP(K)=0.
26    T(K)=P(K)
      DP(I)=SKL
      IF (IGM.EQ.1.AND.J.EQ.2) DP(I)=SCALE(I)
      IF (IGM.EQ.2) DP(I)=STP(I)
      IF (IEND.EQ.1) SCALE(I)=SKL
      ICK=0
      IF (IGM.EQ.1) GO TO 30
      CALL ROTATE (NP,DP,ANGL,3)
      DO 28 K=1,NP
28    DP(K)=DP(K)*SCALE(K)
30    ID=0
35    DO 36 K=K1,K2
36    P(K)=T(K)+DP(K)
      ID=ID+1
      CALL BOUNDS (P,IOUT)
      IF (IOUT.EQ.0) GO TO 42
      IF (ID.EQ.2) GO TO 52
      DO 40 K=K1,K2
40    DP(K)=-DP(K)
      GO TO 35
42    CALL PROC (P,C2)
45    CC(I)=ABS((C1-C2)/CBS)
      IF (L.EQ.1) GO TO 67
      IF (CC(I).EQ.CCMAX) GO TO 65
      IF (CC(I).LT.CCMAX) GO TO 60
50    IF (ICK.EQ.2) GO TO 65
52    DO 53 K=K1,K2
53    DP(K)=DP(K)/RDP
      IF (L.EQ.1) GO TO 30
      NN=NN-1
      IF (IABS(NN).GT.100) GO TO 65
58    ICK=1
      GO TO 30
60    IF (ICK.EQ.1) GO TO 65
      DO 62 K=K1,K2
62    DP(K)=DP(K)*RDP
```

```
      NN=NN+1
      IF (IABS(NN).GT.100) GO TO 65
      ICK=2
      GO TO 30
   65 IF (IGM.EQ.1) SCALE(I)=DP(I)
      IF (IR.EQ.0) STEP(I)=DP(I)
      IF (IGM.EQ.2) STEP(I)=STP(I)*(RDP**NN)
      IF (IGM.EQ.2) STP(I)=STEP(I)
      GO TO 68
   67 IF (CC(I).GT.CCMAX) CCMAX=CC(I)
   68 DO 69 K=K1,K2
      DP(K)=0.
      P(K)=T(K)
   69 CONTINUE
      IF (L.EQ.1) GO TO 75
   70 CONTINUE
   75 CONTINUE
      IF (IR.EQ.0.OR.IGM.EQ.2) GO TO 77
      IGM=2
      GO TO 20
   77 IF (IR.EQ.0) GO TO 80
      STPM=0.0
      DO 78 K=1,NP
   78 IF (ABS(STEP(K)).GT.ABS(STPM)) STPM=STEP(K)
      DO 79 K=1,NP
   79 STEP(K)=STEP(K)/STPM
   80 IF (IO.LE.0) GO TO 82
           WRITE (5,201) (P(I),I=1,NP)
           WRITE (5,202) (STEP(I),I=1,NP)
           WRITE (5,203) (SCALE(I),I=1,NP)
C----- SEARCH OF PARAMETERS
   82 IGM=1
      AQ=0.0
      IAQ=0
      COST=C1
   83 CALL PATERN (NP,P,STEP,SCALE,NA,ANGL,C1,AQ,NPASS1,IGM,IR)
      IF (ABS(C1).GT.0.001) CBS=ABS(C1)
   84      IF (IO.GE.0) WRITE (5,205) COST,C1,(P(I),I=1,NP)
      IF (COST.EQ.C1) GO TO 85
      IFAIL=0
      SKL=SKL*1.5
      GO TO 20
   85 IFAIL=IFAIL+1
      IF (IFAIL.EQ.1) GO TO 97
      IF (IAQ.EQ.1) GO TO 94
      AQ=CBS*0.0002
      IAQ=1
      GO TO 83
   94 IF (IEND.EQ.1.AND.SKL.LE.SKLM) RETURN
      SKL=SKL/SKLR
```

```
      IF (SKL.GT.SKLM) GO TO 96
      IEND=1
      SKL=SKLM
   96 IFAIL=0
      GO TO 20
C----- SEARCH OF ROTATION ANGLES
   97 IGM=2
   98 CALL PATERN (NP,P,STEP,SCALE,NA,ANGL,COST,0.0,NPASS2,IGM,IR)
           IF (IO.GE.1) WRITE (5,206) (ANGL(I),I=1,NA)
      DO 99 I=1,NA
   99 IF (ANGL(I).GT.ANGM) IR=1
      IF (IR.EQ.0) IGM=1
      GO TO 20
  300 WRITE (5,208)
  400 RETURN
  201 FORMAT (/1X,28H*** SEARCH OF PARAMETER ***,/1X,6H P(I)=,8F11.7)
  202 FORMAT (1X,9H STEP(I)=,8F11.7)
  203 FORMAT (1X,10H SCALE(I)=,8F11.7)
  205 FORMAT (/1X,22H  COST BEFORE SEARCH =,F12.7,18H ; AFTER SEARCH =
     C          ,F12.7,/1X,6H P(I)=,8F11.7)
  206 FORMAT (/1X,26H *** ROTATION ANGLES ***
     C      ,/1X,9H ANGL(I)=,8F8.4)
  208 FORMAT (1X,28H INITIAL GUESS OUT OF BOUNDS   )
      END
```

```
C-------------------------------------------------------------
C     THIS SUBROUTINE PERFORMS THE PARAMETER SEARCH AND
C     THE ROTATION ANGLE SEARCH.
C-------------------------------------------------------------
      SUBROUTINE PATERN(NP,P,STEP,SCALE,NA,ANGL,COST,AC,NRD,IGM,IR)
      DIMENSION STEP(50),S(50),PS(50),T(50),B1(50),B2(50),P(50)
      DIMENSION PM(50),PP(50),PB1(50),PB2(50),PT(50),ANGL(50)
      DIMENSION DPM(50),DP1(50),DP2(50),SCALE(50)
      L=0
      LF=0
      C1=COST
      ICK=2
      ITTER=0
      IF (IGM.EQ.2) GO TO 5
C----- SEARCH OF PARAMETERS
      RSTP=5.0
      DO 3 I=1,NP
      PS(I)=P(I)
      B1(I)=P(I)
      B2(I)=P(I)
      PM(I)=P(I)
      T(I)=P(I)
```

```
        S(I)=STEP(I)*RSTP
        DP2(I)=0.0
    3   CONTINUE
        NPS=NP
        IF (IR.EQ.0) GO TO 19
        CALL ROTATE (NP,DP2,ANGL,1)
        GO TO 19
C----- SEARCH OF ROTATION ANGLES
    5   DO 6 I=1,NA
        PS(I)=ANGL(I)
        B1(I)=ANGL(I)
        B2(I)=ANGL(I)
        PM(I)=ANGL(I)
        T(I)=ANGL(I)
    6   S(I)=150.0
        NPS=NA
        RSTP=5.0
        DO 8 I=1,NP
        PP(I)=P(I)
        PT(I)=P(I)
    8   DP2(I)=0.0
        SIGN=1.0
        JJ=0
C-------------------------------------------------------------
C----- EVALUATION OF THE COST OF A SET OF ROTATION ANGLES
   10   CALL ROTATE (NP,DP2,PS,1)
        DO 18 I2=1,NP
        DP2(I2)=1.
        CALL ROTATE (NP,DP2,PS,3)
        DO 11 K=1,NP
        DP1(K)=SCALE(K)*DP2(K)
   11   DP2(K)=0.0
        IC2=0
   12   DO 14 K=1,NP
   14   PP(K)=PT(K)+DP1(K)*SIGN
        IC2=IC2+1
        CALL BOUNDS (PP,IOUT2)
        IF (IOUT2.EQ.1) GO TO 16
   15   L=L+1
        CALL PROC (PP,C2)
        IF (I2.EQ.1.AND.IC2.EQ.1) COST2=C2
        IF (C2.LT.COST2) COST2=C2
   16   IF (IC2.EQ.2) GO TO 18
        SIGN=-SIGN
        GO TO 12
   18   CONTINUE
        C2=COST2
        IF (JJ.EQ.1) GO TO 32
        IF (JJ.EQ.2) GO TO 58
C-------------------------------------------------------------
        C1=COST2
```

```
C----- BEGINNING OF PATERN SEARCH STRATEGY
   19   INRD=1
   20   LF=0
        DO 21 I=1,NPS
   21   S(I)=S(I)/RSTP
   22   IFAIL=0
C----- PERTURBATION ABOUT T
        I=1
   24   IC=0
        IF (L.GT.(NPS)*500) GO TO 100
        IF (LF.GT.(NPS*1)) GO TO 99
   25   IF (IR.EQ.1.AND.IGM.EQ.1) GO TO 26
        PS(I)=T(I)+S(I)
        GO TO 28
   26   DP2(I)=S(I)
        CALL ROTATE (NP,DP2,ANGL,3)
        DO 27 K=1,NP
        DP1(K)=SCALE(K)*DP2(K)
        PS(K)=T(K)+DP1(K)
   27   DP2(K)=0.0
   28   IC=IC+1
        JJ=1
        IF (IGM.EQ.1) GO TO 30
        IOUT=0
        DO 29 IBD=1,NPS
   29   IF (PS(IBD).LT.0.0.OR.PS(IBD).GE.90.0) IOUT=1
        IF (IOUT.EQ.1) GO TO 35
        GO TO 10
   30   CALL BOUNDS (PS,IOUT)
        IF (IOUT.EQ.1) GO TO 35
        CALL PROC (PS,C2)
   32   L=L+1
        LF=LF+1
   34   IF (C1-C2+AC) 35,35,43
   35   IF (IC.EQ.2) GO TO 40
        S(I)=-S(I)
        GO TO 25
   40   IFAIL=IFAIL+1
        IF (IR.EQ.0.OR.IGM.EQ.2) GO TO 42
        DO 41 K=1,NPS
   41   PS(K)=T(K)
   42   PS(I)=T(I)
        GO TO 55
   43   T(I)=PS(I)
        IF (IR.EQ.0.OR.IGM.EQ.2) GO TO 53
        DO 45 K=1,NPS
   45   T(K)=PS(K)
   53   IF (C2.GE.C1) GO TO 55
        DO 54 IPM=1,NPS
   54   PM(IPM)=PS(IPM)
        C1=C2
```

```
      LF=0
   55 I=I+1
      IF (I.LE.NPS) GO TO 24
      IF (IFAIL.LT.NPS) GO TO 70
      IF (ICK.EQ.2) GO TO 90
      IF (ICK.EQ.1) GO TO 70
      JJ=2
      IF (IGM.EQ.2) GO TO 10
      CALL PROC (PS,C2)
   58 L=L+1
      LF=LF+1
   60 IF (C1-C2+AC) 62,62,66
   62 ICK=1
      DO 64 I=1,NPS
      B1(I)=B2(I)
      PS(I)=B2(I)
   64 T(I)=B2(I)
      GO TO 22
   66 IF (C2.GE.C1) GO TO 70
      DO 68 IPM=1,NPS
   68 PM(IPM)=PS(IPM)
      C1=C2
      LF=0
   70 IB1=0
      DO 72 I=1,NPS
      B2(I)=T(I)
      IF (ABS(B1(I)-B2(I)).LT.1.0E-20) IB1=IB1+1
   72 CONTINUE
      IF (IB1.EQ.NPS) GO TO 90
      ICK=0
      ITTER=ITTER+1
C----- ACCELERATION STEP
   75 SJ=1.0
      DO 85 II=1,11
      IF (II.EQ.11) SJ=0.0
      DO 76 I=1,NPS
      T(I)=B2(I)+SJ*(B2(I)-B1(I))
   76 PS(I)=T(I)
      SJ=SJ-0.1
      IF (IGM.EQ.1) GO TO 80
      IOUT=0
      DO 78 IBD=1,NPS
   78 IF (PS(IBD).LT.0.0.OR.PS(IBD).GE.90.0) IOUT=1
      GO TO 84
   80 CALL BOUNDS (PS,IOUT)
   84 IF (IOUT.LT.1) GO TO 86
      IF (II.EQ.11) ICK=1
   85 CONTINUE
   86 DO 88 I=1,NPS
   88 B1(I)=B2(I)
      GO TO 22
```

```
 90    DO 91 I=1,NPS
 91    T(I)=PM(I)
 99    INRD=INRD+1
       IF (INRD.LE.NRD) GO TO 20
100    DO 101 I=1,NPS
101    PS(I)=PM(I)
       DO 103 I=1,NPS
       IF (IGM.EQ.2) GO TO 102
       P(I)=PM(I)
       GO TO 103
102    ANGL(I)=PM(I)
103    CONTINUE
       IF (IGM.EQ.2) RETURN
       COST=C1
       RETURN
       END
```

```
C-------------------------------------------------------------
C      THIS SUBROUTINE PERFORMS THE COORDINATE ROTATION
C      FOR A GIVEN SET OF ROTATION ANGLES (AS).
C-------------------------------------------------------------
       SUBROUTINE ROTATE (NP,P,AS,ICR)
       DIMENSION P(10),PP(10),AS(40),RM(10,10),RP(10,10),RRM(10,10)
       IF (ICR.GT.1) GO TO 55
       DO 10 I=1,NP
       DO 10 J=1,NP
       RM(I,J)=0.0
  10   RM(I,I)=1.0
       M=1
       KN=NP-1
C----ROTATION MATRIX
       DO 51 K=1,KN
       LK=K+1
       DO 50 L=LK,NP
       IF (AS(M).EQ.0.0) GO TO 47
       PHI=3.14159*AS(M)/180.0
       DO 20 I=1,NP
       DO 20 J=1,NP
       RP(I,J)=0.0
  20   RP(I,I)=1.0
       RP(K,K)=COS(PHI)
       RP(L,L)=COS(PHI)
       RP(K,L)=SIN(PHI)
       RP(L,K)=-SIN(PHI)
       IF (K.NE.1.OR.L.NE.NP.OR.NP.EQ.2) GO TO 27
       RP(1,NP)=-SIN(PHI)
       RP(NP,1)=SIN(PHI)
```

```
   27    DO 40 I=1,NP
         DO 40 J=1,NP
         SUM=0.0
         DO 30 KP=1,NP
   30    SUM=SUM+RP(I,KP)*RM(KP,J)
         RRM(I,J)=SUM
   40    CONTINUE
         DO 45 I=1,NP
         DO 45 J=1,NP
   45    RM(I,J)=RRM(I,J)
   47    M=M+1
   50    CONTINUE
   51    CONTINUE
C----EVALUATION OF COORDINATES IN ROTATED SPACE
   55    IF (ICR.EQ.3) GO TO 70
         DO 60 I=1,NP
         SUM=0.0
         DO 58 J=1,NP
   58    SUM=RM(I,J)*P(J)+SUM
   60    PP(I)=SUM
         DO 65 I=1,NP
   65    P(I)=PP(I)
         RETURN
C----EVALUATION OF COORDINATES IN ORIGINAL SPACE
   70    DO 80 J=1,NP
         SUM=0.0
         DO 75 I=1,NP
   75    SUM=RM(I,J)*P(I)+SUM
   80    PP(J)=SUM
         DO 85 I=1,NP
   85    P(I)=PP(I)
         RETURN
         END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C       MAIN PROGRAM TO INITIALIZE SEARCH OF                       C
C             CONTROLLER PARAMETERS                                C
C                                                                  C
C                                                                  C
C------------------------------------------------------------------C
C      (1) SUBROUTINE PROC WILL BE USED TO EVALUATE                C
C          THE OBJECTIVE FUNCTION VALUE FOR CONTROLLER             C
C          TUNING.                                                 C
C      (2) SUBROUTINE BOUNDS WILL BE USED TO CHECK                 C
C          THE CONSTRAINT EQUATIONS FOR CONTROLLER                 C
C          PARAMETERS.                                             C
C      (3) SUBROUTINE SEARCH, PATERN, AND ROTATE WILL              C
C          BE USED TO PERFORM THE MODIFIED PATTERN                 C
C          SEARCH.                                                 C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        DIMENSION P(10)
        COMMON IFLAG,N,ST,T(200),U(200),D(200),Y(200),R(200)
        COMMON GAIN,TAU,ZET,DEL,USS,DSS,YSS,RSS,XSS
C----PROCESS PARAMETERS-------
C      GAIN : PROCESS GAIN
C      TAU  : TIME CONSTANT
C      ZET  : DAMPING RATIO
C      DEL  : DEAD TIME
C-----------------------------
C      N    : NUMBER OF DATA
C      ST   : SAMPLING PERIOD
C-----------------------------
        GAIN=1.5
        TAU=3.0
        ZET=1.4
        DEL=2.3
        N=100
        ST=1.0
        NP=3
C-----INITIAL CONDITIONS-------
C      USS : CONTROLLER OUTPUT
C      DSS : PROCESS LOAD
C      YSS : PROCESS INPUT
C      XSS : PROCESS OUTPUT
C      RSS : SET POINT
C-----------------------------
        USS=1.0
        DSS=0.5
        YSS=USS+DSS
        XSS=YSS*GAIN
        RSS=XSS
        D(1)=DSS
        U(1)=USS
        Y(1)=YSS
```

```
        R(1)=RSS
        T(1)=0.0
C----SELECTION OF TUNING METHOD----------
C       ITUN=1 FOR SET POINT CHANGE TUNING
C       ITUN=2 FOR LOAD CHANGE TUNING
C---------------------------------------
        ITUN=1
        DO 20 I=2,N
        T(I)=T(I-1)+ST
        D(I)=DSS
        R(I)=RSS
        IF (ITUN.EQ.1) R(I)=R(I)+1.0
        IF (ITUN.EQ.2) D(I)=D(I)+1.0
   20   CONTINUE
C----PARAMETER DEFINITION----------
C       P(1) : CONTROLLER GAIN
C       P(2) : 1.0/(INTEGRAL TIME)
C       P(3) : DERIVATIVE TIME
C---------------------------------
C----INITIAL CONTROLLER PARAMETERS
        WRITE (5,30)
   30   FORMAT (1X,26H ENTER KC,TI,TD (*,*,*)          )
        READ (5,*) (P(I),I=1,NP)
C----START SEARCH
        NP=3
        IANG=4
        SKLM=0.001
        SKLR=50.0
        IO=3
        IFLAG=0
        CALL SEARCH (NP,P,SKLM,SKLR,IANG,IO)
C----LIST PROCESS RESPONSE
   50   IFLAG=1
        CALL PROC(P,COST)
        STOP
        END
```

```
C-------------------------------------------------------
C     THIS SUBROUTINE EVALUATES THE OBJECTIVE FUNCTION
C     VALUE FOR THE CONTROLLER TUNING.
C-------------------------------------------------------
        SUBROUTINE PROC(P,COST)
        DIMENSION P(10),X(200),E(200),DU(200)
        COMMON IFLAG,N,ST,T(200),U(200),D(200),Y(200),R(200)
        COMMON GAIN,TAU,ZET,DEL,USS,DSS,YSS,RSS,XSS
C------PARAMETERS DETERMINATION
        CKC=P(1)
        TI=1./P(2)
        TD=P(3)
        NDEL=IFIX(DEL)
        NDEL1=NDEL+1
        NDEL2=NDEL+2
        DELI=NDEL+1
        DM=DELI-DEL
        IF (ZET-1.0) 35,40,45
C------UNDERDAMPED SYSTEM
   35   A=ZET/TAU
        W=SQRT(1.0-ZET**2)/TAU
        SIT=ATAN(-A/W)
        EM=EXP(-A*DM*ST)
        ET=EXP(-A*ST)
        ALF1=-2.0*ET*COS(W*ST)
        ALF2=ET**2
        BLT1=EM*COS(W*DM*ST+SIT)/COS(SIT)
        BLT2=-EM*ET*COS((1.0-DM)*W*ST-SIT)/COS(SIT)
        A1=-ALF1
        A2=-ALF2
        B1=GAIN*(1-BLT1)
        B2=GAIN*(ALF1-BLT2+BLT1)
        B3=GAIN*(ALF2+BLT2)
        GO TO 50
C------CRITICALLY DAMPED SYSTEM
   40   ET=EXP(-ST/TAU)
        EM=EXP(-DM*ST/TAU)
        A1=2.0*ET
        A2=-ET**2
        B1=GAIN*(1.-(1.+DM*ST/TAU)*EM)
        B2=GAIN*((1.+DM*ST/TAU)*EM*(1.+ET)-
     C    2.*ET-ST*ET*EM/TAU)
        B3=GAIN*(ET**2-(1.+DM*ST/TAU)*EM*ET+ST*ET*EM/TAU)
        GO TO 50
C------OVERDAMPED SYSTEM
   45   RA=1.0
        RB=2.0*ZET/TAU
        RC=1.0/(TAU**2)
        R1=(-RB+SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
        T1=-1.0/R1
        R2=(-RB-SQRT(RB**2-4.0*RA*RC))/(2.0*RA)
```

```
      T2=-1.0/R2
      E1=EXP(-ST/T1)
      E2=EXP(-ST/T2)
      A1=(E1+E2)
      A2=-E1*E2
      EM1=EXP(-DM*ST/T1)
      EM2=EXP(-DM*ST/T2)
      ALF=T1/(T2-T1)
      BLT=T2/(T2-T1)
      B1=GAIN*(1.0+ALF*EM1-BLT*EM2)
      B2=-GAIN*(E1+E2+(E2+1.)*ALF*EM1-(E1+1.0)*BLT*EM2)
      B3=GAIN*(E1*E2+E2*ALF*EM1-E1*BLT*EM2)
C------INITIAL CONDITION
   50 DO 60 I=1,NDEL1
      X(I)=XSS
      E(I)=R(I)-X(I)
      IF (I.EQ.1) U(I)=USS
      IF (I.EQ.2) DU(I)=CKC*(E(I)-E(I-1)+(ST/TI)*E(I))
      IF (I.GE.3) DU(I)=CKC*(E(I)-E(I-1)+(ST/TI)*E(I)+
     C                    (TD/ST)*(E(I)-2.*E(I-1)+E(I-2)))
      IF (I.GT.1) U(I)=U(I-1)+DU(I)
      Y(I)=U(I)+D(I)
   60 CONTINUE
      DO 70 I=NDEL2,N
      K=I-NDEL1
      IF (I.GE.2) XS=A1*(X(I-1)-XSS)
      IF (I.GE.3) XS=XS+A2*(X(I-2)-XSS)
      IF (K.GE.1) YS=B1*(Y(K)-YSS)
      IF (K.GE.2) YS=YS+B2*(Y(K-1)-YSS)
      IF (K.GE.3) YS=YS+B3*(Y(K-2)-YSS)
      X(I)=XSS+YS+XS
      E(I)=R(I)-X(I)
      IF (I.EQ.2) DU(I)=CKC*(E(I)-E(I-1)+(ST/TI)*E(I))
      IF (I.GE.3) DU(I)=CKC*(E(I)-E(I-1)+(ST/TI)*E(I)+
     C                    (TD/ST)*(E(I)-2.*E(I-1)+E(I-2)))
      U(I)=U(I-1)+DU(I)
      Y(I)=U(I)+D(I)
   70 CONTINUE
C------COST OF THE OBJECTIVE FUNCTION
      COST=0.0
      DO 80 I=1,N
      COST=COST+ABS(E(I))
   80 CONTINUE
      IF (IFLAG.EQ.0) RETURN
C------PROCESS  RESPONSE
      WRITE (5,100) CKC,TI,TD,COST
  100 FORMAT (3X,4H KC=,F7.4,5X,4H TI=,F7.4,5X,4H TD=,F7.4,E16.5)
      WRITE (5,101)
  101 FORMAT (/5X,5H TIME,4X,8H SET PT.,3X,5H LOAD,4X,8H CONTROL,
     C        3X,6H INPUT,4X,7H OUTPUT,/1X)
```

```
      DO 120 I=1,N
      WRITE (5,102) T(I),R(I),D(I),U(I),Y(I),X(I)
  102 FORMAT (1X,6F10.3)
  120 CONTINUE
      RETURN
      END



C-------------------------------------------------
C     THIS SUBROUTINE CHECKS THE CONSTRAINT
C     EQUATIONS FOR THE CONTROLLER PARAMETERS.
C-------------------------------------------------
      SUBROUTINE BOUNDS(P,IOUT)
      DIMENSION P(10)
   10 IOUT=0
      IF (P(1).LE.0.0) IOUT=1
      IF (P(2).LE.0.0) IOUT=1
      IF (P(3).LT.0.0) IOUT=1
   20 RETURN
      END
```

```
C-------------------------------------------------------
C      THIS SUBROUTINE PERFORMS THE PARAMETER SEARCH BY
C      USING THE PATTERN SEARCH (HOOKE AND JEEVES)
C-------------------------------------------------------
       SUBROUTINE PATERN (NP,P,STEP,NRD,IO,COST)
C-----THE SIZE OF B1,B2,T,AND S NEED ONLY BE EQUAL TO THE NUMBER OF PAR
      DIMENSION P(10),STEP(10),B1(10),B2(10),T(10),S(10)
C-----STARTING POINT
      L=1
      ICK=2
      ITTER=0
      DO 5 I=1,NP
      B1(I)=P(I)
      B2(I)=P(I)
      T(I)=P(I)
  5   S(I)=STEP(I)*10.
C-----INITIAL BOUNDARY CHECK AND COST EVALUATION
      CALL BOUNDS(P,IOUT)
      IF(IOUT.LE.0) GO TO 10
                    IF(IO.LE.0) GO TO 6
                    WRITE(5,1005)
                    WRITE(5,1000)(J,P(J),J=1,NP)
  6   RETURN
 10   CALL PROC(P,C1)
                    IF(IO.LE.0) GO TO 11
                    WRITE(5,1001)ITTER,C1
                    WRITE(5,1000)(J,P(J),J=1,NP)
C-------BEGINNING OF PATTERN SEARCH STRATEGY
 11   DO 99 INRD=1,NRD
      DO 12 I=1,NP
 12   S(I)=S(I)/10.0
                    IF(IO.LE.0) GO TO 20
                    WRITE(5,1003)
                    WRITE(5,1000)(J,S(J),J=1,NP)
        WRITE (5,1000) (J,P(J),J=1,NP)
 20   IFAIL=0.0
C-----PRETURBATION ABOUT T
      DO 30 I=1,NP
        IF (L.GT.3000) GO TO 888
      IC=0
 21   P(I)=T(I)+S(I)
      IC=IC+1
      CALL BOUNDS(P,IOUT)
      IF(IOUT.GT.0) GO TO 23
      CALL PROC(P,C2)
                    L=L+1
                    IF(IO.LT.3) GO TO 22
                    WRITE(5,1002)L,C2
                    WRITE(5,1000)(J,P(J),J=1,NP)
 22   IF(C1-C2)23,23,25
 23   IF(IC.GE.2) GO TO 24
```

```
      S(I)=-S(I)
      GO TO 21
 24   IFAIL=IFAIL+1
      P(I)=T(I)
      GO TO 30
 25   T(I)=P(I)
      C1=C2
 30   CONTINUE
      IF(IFAIL.LT.NP) GO TO 35
      IF(ICK.EQ.2) GO TO 90
      IF(ICK.EQ.1) GO TO 35
      CALL PROC(T,C2)
                  L=L+1
                  IF(IO.LT.2) GO TO 31
                  WRITE(5,1002)L,C2
                  WRITE(5,1000)(J,T(J),J=1,NP)
 31   IF(C1-C2)32,34,34
 32   ICK=1
      DO 33 I=1,NP
      B1(I)=B2(I)
      P(I)=B2(I)
 33   T(I)=B2(I)
      GO TO 20
 34     CONTINUE
        C1=C2
 35   IB1=0
      DO 39 I=1,NP
      B2(I)=T(I)
      IF(ABS(B1(I)-B2(I)).LT.1.0E-20)IB1=IB1+1
 39   CONTINUE
      IF(IB1.EQ.NP) GO TO 90
      ICK=0
                  ITTER=ITTER+1
                  IF(IO.LT.2) GO TO 40
                  WRITE(5,1001)ITTER,C1
                  WRITE(5,1000)(J,T(J),J=1,NP)
C-----ACCELERATION STEP
 40   SJ=1.0
      DO 45 II=1,11
        IF (II.EQ.11) SJ=0.0
      DO 42 I=1,NP
      T(I)=B2(I)+SJ*(B2(I)-B1(I))
 42   P(I)=T(I)
      SJ=SJ-.1
      CALL BOUNDS(T,IOUT)
      IF(IOUT.LT.1) GO TO 46
      IF(II.EQ.11)ICK=1
 45   CONTINUE
 46   DO 47 I=1,NP
 47   B1(I)=B2(I)
      GO TO 20
```

```
 90    DO 91 I=1,NP
 91    T(I)=B2(I)
 99    CONTINUE
888      DO 100 I=1,NP
100    P(I)=T(I)
       COST=C1
                    IF(IO.LE.0)RETURN
                    WRITE(5,1004)L,C1
                    WRITE(5,1000)(J,P(J),J=1,NP)
       RETURN
1000     FORMAT (6X,4(4H  P(,I2,2H):,F8.4)/)
1001   FORMAT(//1X14HITTERATION NO.  ,I5/5X,5HCOST= ,E15.6,20X,
      1    10HPARAMETERS)
1002   FORMAT(10X3HNO.,I4, 8X5HCOST=,E15.6)
1003   FORMAT(/1X28HSTEP SIZE FOR EACH PARAMETER    )
1004   FORMAT(1H113HANSWERS AFTER ,I6,2X,23HFUNCTIONAL EVALUATIONS   //
      1    5X5HCOST=,E15.6,20X,18HOPTIMAL PARAMETERS  )
1005   FORMAT(1H135HINITIAL PARAMETERS OUT OF BOUNDS         )
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      OPTIMIZATION METHOD OF NELDER AND MEAD (SIMPLEX)               C
C                                                                     C
C-----------------------------------------------------------------    C
C    (1) SUBROUTINE START WILL BE CALLED TO CONSTRUCT                 C
C        THE INITIAL POLYHEDRON.                                      C
C    (2) SUBROUTINE SUMR WILL BE CALLED TO EVALUATE                   C
C        THE OBJECTIVE FUNCTION VALUE.                                C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C----- NX    : NUMBER OF INDEPENDENT VARIABLE
C----- STEP : INITIAL STEP SIZE
C----- X(I) : INITIAL GUESSES
       COMMON /ONE/ X(30),X1(30,30),NX,STEP,K1,SUM(30),IN
       NX=2
       X(1)=0.1
       X(2)=0.00001
       STEP=0.1
C----- ALFA : REFLECTION COEFFICIENT
C----- BETA : CONTRACTION COEFFICIENT
C----- GAMA : EXPENSION COEFFICIENT
       ALFA=1.0
       BETA=0.5
       GAMA=2.0
       DIFER=0.0
       XNX=NX
       IN=1
       CALL SUMR
C----- K1 : NUMBER OF VERTICES    K2 : INDEX OF CENTROID
C----- K3 : INDEX OF REFLECTION   K3 : INDEX OF EXPENSION OR CONTRACTION
       K1=NX+1
       K2=NX+2
       K3=NX+3
       K4=NX+4
       CALL START
       DO 15 I=1,K1
       DO 10 J=1,NX
   10  X(J)=X1(I,J)
       WRITE (5,12) (X(JK),JK=1,NX)
   12  FORMAT (1X,3H P=,4F7.3)
       IN=I
       CALL SUMR
   15  CONTINUE
C----- SELECT LARGEST VALUE OF SUM(I) IN SIMPLEX
   20  SUMH=SUM(1)
       INDEX=1
       DO 22 I=2,K1
       IF (SUM(I).LE.SUMH) GO TO 22
       SUMH=SUM(I)
       INDEX=I
   22  CONTINUE
```

```
C----- SELECT MINIMUM VALUE OF SUM(I) IN SIMPLEX
       SUML=SUM(1)
       KOUNT=1
       DO 25 I=2,K1
       IF (SUML.LE.SUM(I)) GO TO 25
       SUML=SUM(I)
       WRITE (5,101) SUML,(X1(KOUNT,J),J=1,NX),DIFER
       KOUNT=I
  25   CONTINUE
C----- FIND CENTROID OF POINTS WITH I DIFFERENT THAN INDEX
       DO 28 J=1,NX
       SUM2=0.0
       DO 26 I=1,K1
  26   SUM2=SUM2+X1(I,J)
       X1(K2,J)=1.0/XNX*(SUM2-X1(INDEX,J))
C----- FIND REFLECTION OF HIGH POINT THROUGH CENTROID
       X1(K3,J)=(1.0+ALFA)*X1(K2,J)-ALFA*X1(INDEX,J)
  28   X(J)=X1(K3,J)
       IN=K3
       CALL SUMR
       IF (SUM(K3).LT.SUML) GO TO 40
C----- SELECT SECOND LARGEST VALUE IN SIMPLEX
       IF (INDEX.EQ.1) GO TO 30
       SUMS=SUM(1)
       GO TO 32
  30   SUMS=SUM(2)
  32   DO 35 I=1,K1
       IF ((INDEX-I).EQ.0) GO TO 35
       IF (SUM(I).LE.SUMS) GO TO 35
       SUMS=SUM(I)
  35   CONTINUE
       IF (SUM(K3).GT.SUMS) GO TO 45
       GO TO 64
C----- FORM EXPENSION OF NEW MINIMUM IF REFLECTION HAS PRODUCED ONE MIN.
  40   DO 42 J=1,NX
       X1(K4,J)=(1.0-GAMA)*X1(K2,J)+GAMA*X1(K3,J)
  42   X(J)=X1(K4,J)
       IN=K4
       CALL SUMR
       IF (SUM(K4).LT.SUML) GO TO 60
       GO TO 64
C----- FORM CONTRACTION
  45   IF (SUM(K3).GT.SUMH) GO TO 48
       DO 47 J=1,NX
  47   X1(INDEX,J)=X1(K3,J)
  48   DO 49 J=1,NX
       X1(K4,J)=BETA*X1(INDEX,J)+(1.0-BETA)*X1(K2,J)
  49   X(J)=X1(K4,J)
       IN=K4
       CALL SUMR
```

```
      IF (SUMH.GT.SUM(K4)) GO TO 60
C---- REDUCE SIMPLEX BY HALF IF REFLECTION HAPPENS TO PRODUCE
C     A LARGER VALUE THAN THE MAX.
      DO 50 J=1,NX
      DO 50 I=1,K1
 50   X1(I,J)=0.5*(X1(I,J)+X1(KOUNT,J))
      DO 56 I=1,K1
      DO 54 J=1,NX
 54   X(J)=X1(I,J)
      IN=I
      CALL SUMR
 56   CONTINUE
      GO TO 66
 60   DO 62 J=1,NX
      X1(INDEX,J)=X1(K4,J)
 62   X(J)=X1(INDEX,J)
      IN=INDEX
      CALL SUMR
      GO TO 66
 64   DO 65 J=1,NX
      X1(INDEX,J)=X1(K3,J)
 65   X(J)=X1(INDEX,J)
      IN=INDEX
      CALL SUMR
 66   DO 68 J=1,NX
 68   X(J)=X1(K2,J)
      IN=K2
      CALL SUMR
      DIFER=0.0
      DO 70 I=1,K1
 70   DIFER=DIFER+(SUM(I)-SUM(K2))**2
      DIFER=1.0/XNX*SQRT(DIFER)
      IF (DIFER.GE.0.00001) GO TO 20
      WRITE (5,101) SUML,(X1(KOUNT,J),J=1,NX),DIFER
101   FORMAT (1X,6H SUML=,E16.6,/1X,6H X(I)=,2E16.6,/1X,E16.6)
999   CONTINUE
      STOP
      END
```

```
C-------------------------------------------------------
C     THIS SUBROUTINE CONSTRUCTS THE INITIAL POLYHEDRON.
C-------------------------------------------------------
      SUBROUTINE START
      DIMENSION A(30,30)
      COMMON /ONE/ X(30),X1(30,30),NX,STEP,K1,SUM(30),IN
      VN=NX
      STEP1=STEP/(VN*SQRT(2.))*(SQRT(VN+1.)+VN-1.0)
```

```
      STEP2=STEP/(VN*SQRT(2.))*(SQRT(VN+1.)-1.)
      I=1
      DO 1 J=1,NX
    1 A(I,J)=0.0
      DO 3 I=2,K1
      DO 2 J=1,NX
      A(I,J)=STEP2
      L=I-1
      A(I,L)=STEP1
    2 CONTINUE
    3 CONTINUE
      DO 5 I=1,K1
      DO 4 J=1,NX
    4 X1(I,J)=X(J)+A(I,J)
    5 CONTINUE
      RETURN
      END
```

```
C-------------------------------------------------------
C       THIS SUBROUTINE EVALUATES THE OBJECTIVE FUNCTION
C       VALUE AT A GIVEN VERTEX OF THE POLYHEDRON.
C-------------------------------------------------------
      SUBROUTINE SUMR
      COMMON /ONE/ X(30),X1(30,30),NX,STEP,K1,SUM(30),IN
      A=X(1)
      B=X(2)
      SUM(IN)=ABS(A**2+B**2-1.0)+ 0.1*ABS(A-1.0)
      RETURN
      END
```

VITA

Pen Yuan Chou was born in Taiwan, Republic of China, on November 17, 1949. After graduating from high school in Taipei, Taiwan, in 1968, he attended National Cheng Kung University in Taiwan. He received his B.S.Ch.E. degree from National Cheng Kung University in 1972.

After completing two years' military service, Pen Yuan joined Taiwan Fertilizer Company and worked as an engineer. In 1977 he entered graduate school at the University of Tennessee to pursue an M.S. degree in Chemical Engineering.