OPTIMUM SYSTEM MODELLING

USING

RECENT GRADIENT METHODS

OPTIMUM SYSTEM MODELLING
USING
RECENT GRADIENT METHODS

by

NICHOLAS D. MARKETTOS, B.Sc.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering

McMaster University

April 1972

MASTER OF ENGINEERING — McMASTER UNIVERSITY

(Electrical Engineering) — Hamilton, Ontario.

Title: Optimum System Modelling Using Recent Gradient Methods.

Author: Nicholas Denis Markettos,
B.Sc. (University of Sussex)

Supervisors: N. K. Sinha, B.Sc. (Eng.) (Banaras),
Ph.D. (University of Manchester).

J. W. Bandler, B.Sc. (Eng.), Ph.D. (University of London),
D.I.C. (Imperial College).

Number of Pages: viii ; 87

Scope and Contents: A study of gradient optimization techniques, in particular as applied to system modelling problems, is made. Three efficient techniques are used to derive optimum second-order and third-order models for a seventh-order system. The optimization techniques are the Fletcher-Powell method, a more recent method proposed by Fletcher and a method based on a more general objective function proposed by Jacobson and Oksman.

The approximation is carried out in the time domain. Least squares and least pth criteria are used, and almost minimax results are obtained for large values of p. Values of p up to 10^{12} are successfully used. The results are compared with other minimax type algorithms.

ACKNOWLEDGEMENTS

I wish to express my appreciation to Dr. J. W. Bandler and Dr. N. K. Sinha for their guidance and supervision throughout this work. I am particularly grateful to Dr. D. H. Jacobson and Dr. R. Fletcher for making their minimization algorithms available at the beginning of this work.

Thanks go to C. Charalambous, G. Bereznai and other colleagues for fruitful discussions and to T. V. Srinivasan for making the results of the grazor search technique available for comparison.

The financial support provided by the National Research Council through Research Grants No. A7239 and No. A3374 is gratefully acknowledged.

I would like to thank my wife for typing this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of obtaining linear low-order models of high-order complex systems is to simplify the preliminary design and optimization of such systems. For on-line operation of a complex slowly varying system, it would often be more desirable to obtain a linear low-order model and perform the computation for an optimal control on that model. Although this will give a sub-optimal control for the real system, the result will be considerable economies in computing time and equipment. We are looking, therefore, for models which are computationally and analytically simple, and still provide sufficient information about the system for them to be useful.

In chapter 2 a review is given of existing modelling methods. The most promising of these methods is selected and, as the method is based on minimizing a function, three of the most efficient minimization techniques are applied to solve the problem. These techniques are all gradient methods and are described in chapter 3. Their rapid rate of convergence makes gradient methods suitable for this type of problem.

The modelling method with the optimization techniques is applied to a test problem, which is described in chapter 4, and the results

are shown in chapter 5. Firstly, a least squares criterion is used and, secondly, an effective minimax criterion is used. The minimax result is achieved by minimizing the pth norm of the absolute errors, where p takes very large values - up to 10^{12} has been successfully used. For many nonlinear approximation problems, a minimax result does not necessarily imply an equiripple answer in the Chebyshev sense. That is, not all the extrema are necessarily equal and the number of ripples is not necessarily n+1, where n is the number of variables.

The purpose of this work has been to improve modelling techniques so that on-line control of slowly varying complex systems can be simplified. At the same time a thorough examination is conducted of two recent optimization techniques, by comparing them to one which has been widely used. Parts of this work have been published, and appear in references [1-3].

CHAPTER 2

MODELLING METHODS

The need for low-order approximations of complex high-order systems has resulted in several model derivation techniques. These techniques can be divided into two main groups. In one group a model is obtained by neglecting modes of the original system which contribute little to the overall response of the system. A number of variations based on this approach have been proposed by Davison [4], Chidambara [5], Mitra [6] and Marshall [7]. The other main approach is to search in some way for the coefficients of a set of differential or difference equations of specified order, the response of which is approximated as closely as possible to that of the system, when both are driven by the same inputs. Anderson's method [8] comes under this category, as do the approaches of Sinha and Pille [9] and Sinha and Bereznai [10]. Other methods, such as that of Chen and Shieh [11] and that of Kokotovic and Sannuti [12] do not seem to fit into either of these general groups.

Davison's method is based on the principle that one may neglect those eigenvalues of the original system which are farthest from the $j\omega$-axis in the s-plane. The retention of the dominant eigenvalues makes the response of the reduced model approximate that of the system, since the eigenvalues neglected make a very insignificant contribution

to the response, except at the beginning. Relationships from the time-solution of the vector differential equation of the original model are used to develop a reduced model which maintains both the correct proportion of the eigenvectors and the desired eigenvalues. Although the method is intuitively appealing, it may be relevant to ask if a greater choice of the eigenvalues in the reduced model could produce a better approximation.

Anderson [8] has proposed a method which does not depend on the computation of the high-order system matrix. In this method an attempt is made to determine a low-order model, the response of which approaches that of the system, so that the mean-square error between the two responses, over a given finite interval, is minimized. This is accomplished by using the orthogonal projection theorem in the theory of linear vector spaces.

The method due to Sinha and Pille [9] is similar to Anderson in that it minimizes the sum of the squares of the errors between the response of the system and that of the model. It is based on the iterative application of the matrix pseudoinverse algorithm and it was shown that the method was easy to apply, especially as no matrix inversions are required. The main drawback, however, of this method is that the objective function can only have the form of the sum of the squares of the errors. In many practical situations it may be more desirable to use other criteria for obtaining low-order models; in these cases the matrix pseudoinverse cannot be used.

Sinha and Bereznai [10] developed an approach which can provide a low-order model with respect to any specified criterion. The method is based on the pattern search algorithm of Hooke and Jeeves [13] Although this method provides flexibility in the choice of criteria, it will, in general, require so much computer time as to make it unsuitable for on-line applications.

The present work is based on the above approach with the use of efficient minimization techniques, These techniques, known as gradient methods because they utilize gradient information, show a rapid rate of convergence and an investigation of these methods is considered useful. For this purpose, a comparative study is made between three of the most efficient techniques which are described in the next chapter.

CHAPTER 3

OPTIMIZATION TECHNIQUES

In order to describe the optimization techniques it is necessary to give some definitions. It is desired to minimize a function $F(\underset{\sim}{x})$ called the objective function, where

$$\underset{\sim}{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{1}$$

called the parameter vector. It is assumed there are no constraints present. If constraints are imposed, however, then the problem can be transformed to an unconstrained one by use of appropriate transformations.

Let

$$\underset{\sim}{g}(\underset{\sim}{x}) \triangleq \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \vdots \\ \frac{\partial F}{\partial x_n} \end{bmatrix} \tag{2}$$

called the gradient vector and

$$\underset{\sim}{G} \triangleq \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \cdots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_n^2} \end{bmatrix} \tag{3}$$

is the corresponding Hessian matrix. $\underset{\sim}{H}$ denotes the inverse Hessian, $\underset{\sim}{G}^{-1}$, which will be approximated by different formulae in each of the gradient techniques.

Let

$$\underset{\sim}{\delta} \triangleq \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \tag{4}$$

called the increment. In all minimization methods $\underset{\sim}{\delta}$ is chosen so that

$$F(\underset{\sim}{x} + \underset{\sim}{\delta}) < F(\underset{\sim}{x}) \tag{5}$$

Another n-dimenstional vector $\underset{\sim}{s}$ will denote the direction in which $\underset{\sim}{\delta}$ is taken. For brevity in some expressions in the following sections, F will be used for $F(\underset{\sim}{x})$ and $\underset{\sim}{g}$ for $\underset{\sim}{g}(\underset{\sim}{x})$.

Algorithms terminate after one or more of the following criteria are satisfied:

(a) if the change in the objective function becomes less than ϵ_1, a small positive number.

(b) if the absolute values of the elements of the increment vector become smaller than ϵ_2, a small positive number.

(c) if the norm of the gradient vector becomes less than ϵ_3, another small positive number.

As a safeguard the algorithm should go through n iterations, where n is the number of variables, after the terminating criterion is satisfied, before the program terminates.

3.1 The Fletcher-Powell Method

The main feature of the Fletcher-Powell method [14] is that the increment $\underset{\sim}{\delta}$ is taken along the direction $\underset{\sim}{s}$ where

$$\underset{\sim}{s} = - \underset{\sim}{H} \underset{\sim}{g} \tag{6}$$

That is

$$\underset{\sim}{\delta} = \alpha \underset{\sim}{s} \tag{7}$$

where α is that value of λ which minimizes $F(\underset{\sim}{x} + \lambda \underset{\sim}{s})$ along the direction

of $\underset{\sim}{s}$. The method of obtaining the minimum along the line is not central to the theory, however, cubic interpolation has been found simple and satisfactory.

The inverse Hessian $\underset{\sim}{H}$ is updated at each iteration from information presently available using the formula

$$\underset{\sim}{H}_{i+1} = \underset{\sim}{H}_i + \frac{\underset{\sim}{\delta}\,\underset{\sim}{\delta}^T}{\underset{\sim}{\delta}^T\,\underset{\sim}{\gamma}} - \frac{\underset{\sim}{H}_i\,\underset{\sim}{\gamma}\,\underset{\sim}{\gamma}^T\,\underset{\sim}{H}_i}{\underset{\sim}{\gamma}^T\,\underset{\sim}{H}_i\,\underset{\sim}{\gamma}} \tag{8}$$

where

$$\underset{\sim}{\delta} = -\alpha\,\underset{\sim}{H}_i\underset{\sim}{g}_i \tag{9}$$

and

$$\underset{\sim}{\gamma} = \underset{\sim}{g}_{i+1} - \underset{\sim}{g}_i \tag{10}$$

and the subscript i denotes the value at the ith iteration. It can be shown [14] that the process is stable, that is, formula (8) has the following property of positive definiteness: If $\underset{\sim}{H}_i$ is positive definite then $\underset{\sim}{H}_{i+1}$ is also positive definite. Since $\underset{\sim}{H}_i$ is initially chosen as the identity matrix then all $\underset{\sim}{H}_{i+1}$ will be positive definite. It can also be shown that if the objective function is in the quadratic form the procedure terminates in n iterations. This property of quadratic convergence depends on accurate location of the minimum along each direction of search, and this is the main disadvantage of the method.

3.2 The Fletcher Method

The Fletcher method [15] is basically similar to the Fletcher-Powell method in that both methods consider quadratic objective functions and the increment $\underset{\sim}{\delta}$ depends on the gradient and the updating matrix $\underset{\sim}{H}$. The difference between the methods is that the Fletcher method dispenses with the problem of linear search which is time consuming. But, as mentioned in the previous section, quadratic convergence depends on accurate location of the minimum along each direction using linear search. Therefore the property of quadratic convergence is replaced by a property which requires, for quadratic functions, that the eigenvalues of the matrix $\underset{\sim}{H}$ tend monotonically towards those of $\underset{\sim}{G}^{-1}$, the exact inverse Hessian.

The abandonment of linear searches requires that something is done to force a sufficiently large decrease in F at each iteration to guarantee ultimate convergence. The change ΔF in F on an iteration would be expected by Taylor's series to be approximately $\underset{\sim}{g}^T\underset{\sim}{\delta}$ for small $\underset{\sim}{\delta}$, but much less than $\underset{\sim}{g}^T\underset{\sim}{\delta}$ when the position of the minimum along the line is overestimated. Therefore the change in $F(\underset{\sim}{x})$ relative to $\underset{\sim}{g}^T\underset{\sim}{\delta}$ cannot become arbitrarily small if

$$\frac{\Delta F}{\underset{\sim}{g}^T\underset{\sim}{\delta}} \geq \mu \qquad (11)$$

where $o<\mu<<1$, a pre-assigned small quantity set at 0.0001.

If corrections are determined by

$$\underset{\sim}{\delta} = - \lambda \underset{\sim}{H} \underset{\sim}{g} \tag{12}$$

then trying values of $\lambda = 1, w, w^2, w^3, \ldots$ for $w = 0.1$ will eventaully produce a $\underset{\sim}{\delta}$ that satisfies inequality (11).

Although the above tests can be simply included in the Fletcher-Powell program, $\underset{\sim}{H}$ can become ill-conditioned and a new formula for updating $\underset{\sim}{H}$ is needed. It is necessary for the new formula to possess the properties of positive definiteness and eigenvalue convergence. The new formula derived by Fletcher is

$$\underset{\sim}{H}_{i+1} = \underset{\sim}{H}_i - \frac{\underset{\sim}{\delta}\,\underset{\sim}{\gamma}^T \underset{\sim}{H}_i}{\underset{\sim}{\delta}^T \underset{\sim}{\gamma}} - \frac{\underset{\sim}{H}_i \underset{\sim}{\gamma}\,\underset{\sim}{\delta}^T}{\underset{\sim}{\delta}^T \underset{\sim}{\gamma}} + \left(1 + \frac{\underset{\sim}{\gamma}^T \underset{\sim}{H}_i \underset{\sim}{\gamma}}{\underset{\sim}{\delta}^T \underset{\sim}{\gamma}} \right) \frac{\underset{\sim}{\delta}\,\underset{\sim}{\delta}^T}{\underset{\sim}{\delta}^T \underset{\sim}{\gamma}} \tag{13}$$

where $\underset{\sim}{\gamma}$ and $\underset{\sim}{\delta}$ are defined in (9) and (10). It can be shown that formula (13) possesses the above mentioned properties. The use of formula (13) alone might, however, cause $\underset{\sim}{H}$ to become unbounded. For this reason a choice is made between the two updating formulae by the following test. If

$$\underset{\sim}{\delta}^T \underset{\sim}{\gamma} \geq \underset{\sim}{\gamma}^T \underset{\sim}{H}_i \underset{\sim}{\gamma} \tag{14}$$

then formula (13) is used; otherwise formula (8) is used.

3.3 The Jacobson-Oksman Method

The Jacobson-Oksman method [16] differs from the previous two in that it is not based on quadratic functions, but on homogeneous functions. Consider the homogeneous function

$$F(\underset{\sim}{x}) = \frac{1}{\theta} (\underset{\sim}{x} - \check{\underset{\sim}{x}})^T \underset{\sim}{g}(\underset{\sim}{x}) + F(\check{\underset{\sim}{x}}) \tag{15}$$

where θ is the degree of homogeneity and $\check{\underset{\sim}{x}}$ is the location of the minimum. The quadratic function considered earlier can be expressed as

$$F(\underset{\sim}{x}) = \frac{1}{2} (\underset{\sim}{x} - \check{\underset{\sim}{x}})^T \underset{\sim}{Q} (\underset{\sim}{x} - \check{\underset{\sim}{x}}) + F(\check{\underset{\sim}{x}}) \tag{16}$$

where $\underset{\sim}{Q}$ is a constant positive definite matrix. By comparing equations (15) and (16) it can be seen that (16) is a special case of (15).

The basis of this method is the following. By rearranging equation (15) we have

$$\check{\underset{\sim}{x}}^T \underset{\sim}{g}(\underset{\sim}{x}) + \theta F(\underset{\sim}{x}) - \theta F(\check{\underset{\sim}{x}}) = \underset{\sim}{x}^T \underset{\sim}{g}(\underset{\sim}{x}) \tag{17}$$

Let

$$\begin{aligned} v &\triangleq \underset{\sim}{x}^T \underset{\sim}{g}(\underset{\sim}{x}) \\ \underset{\sim}{y} &\triangleq [\underset{\sim}{g}^T(\underset{\sim}{x}) \quad F(\underset{\sim}{x}) \quad -1]^T \\ \underset{\sim}{\alpha} &\triangleq [\check{\underset{\sim}{x}}^T \quad \theta \quad \phi]^T \end{aligned} \tag{18}$$

where

$$\phi = \theta F(\check{\underset{\sim}{x}})$$

and $\underset{\sim}{\alpha}$ and $\underset{\sim}{y}$ are (n+2)-vectors, with $\underset{\sim}{\alpha}$ containing the unknowns. For some point $\underset{\sim}{x}_i$ equation (17) now becomes

$$\underset{\sim}{y}_i^T \quad \underset{\sim}{\alpha} = v_i \tag{19}$$

If v and $\underset{\sim}{y}$ are evaluated at n+2 distinct points $\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots \underset{\sim}{x}_{n+2}$, so that the $\underset{\sim}{y}_i$ are linearly independent, we have

$$\begin{bmatrix} \underset{\sim}{y}_1^T \\ \underset{\sim}{y}_2^T \\ \vdots \\ \underset{\sim}{y}_{n+2}^T \end{bmatrix} \underset{\sim}{\alpha} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n+2} \end{bmatrix} \tag{20}$$

or, in the matrix form

$$\underset{\sim}{Y}\, \underset{\sim}{\alpha} = \underset{\sim}{v} \tag{21}$$

Since the $\underset{\sim}{y}_i$ are linearly independent, the matrix $\underset{\sim}{Y}$ is non-singular giving

$$\underset{\sim}{\alpha} = \underset{\sim}{Y}^{-1} \underset{\sim}{v} \tag{22}$$

Matrix inversion is avoided by using a recursive formula as new $\underset{\sim}{y}_i$ and v_i are evaluated. Starting with $\underset{\sim}{P}_0 = \underset{\sim}{I}$, an (n+2) x (n+2) identity matrix and $\underset{\sim}{v}_0 = \underset{\sim}{\alpha}_0$, an arbitrary initial guess, successive estimates of the vector

$\underset{\sim}{x}$ are given by

$$\underset{\sim}{x}_{i+1} = \underset{\sim}{x}_i + \frac{\underset{\sim}{P}_i \; \underset{\sim}{e}_{i+1} \; (v_{i+1} - \underset{\sim}{y}_{i+1}^T \; \underset{\sim}{x}_i)}{\underset{\sim}{y}_{i+1}^T \; \underset{\sim}{P}_i \; \underset{\sim}{e}_{i+1}} \qquad (23)$$

where $\underset{\sim}{e}_{i+1}$ is a unit (i+1)-vector having unity as the (i+1)th element and zero elsewhere, and where $\underset{\sim}{P}_i$ are obtained successively from the formula

$$\underset{\sim}{P}_{i+1} = \underset{\sim}{P}_i - \frac{\underset{\sim}{P}_i \; \underset{\sim}{e}_{i+1} \; (\underset{\sim}{y}_{i+1}^T \; \underset{\sim}{P}_i - \underset{\sim}{e}_{i+1}^T)}{\underset{\sim}{y}_{i+1}^T \; \underset{\sim}{P}_i \; \underset{\sim}{e}_{i+1}} \qquad (24)$$

It can be shown that, for homogeneous functions, the algorithm finds the minimum $\overset{v}{\underset{\sim}{x}}$, the degree of homogeneity θ, and the value of the minimum $F(\overset{v}{\underset{\sim}{x}})$ after n+2 iterations.

CHAPTER 4

THE TEST PROBLEM

In order to compare the various optimization techniques on modelling, a realistic and valid system is selected. The system is one of the designs studied for a supersonic transport aircraft [17] Some variables are given values, consistent with design description and maintaining stability, so as to obtain a transfer function with poles distributed over the entire left half s-plane. The reason is that if the poles are close to the origin there is bias in favour of Davison's method.

The transfer function obtained is

$$\frac{C(s)}{R(s)} = \frac{375000(s + 0.0833)}{s^7+83.64s^6+4097s^5+70342s^4+853703s^3+2814271s^2+3310875s+281250} \quad (25)$$

Since in most realistic slowly varying dynamic systems a transfer function would not be available, the above transfer function is used to obtain the response of the system to a specified input. A step is a widely used input, therefore the step response of the system is calculated at discrete intervals of time. This in reality would correspond to direct measurements at the output of the system, at discrete instants. This means that a knowledge of the system transfer function is not required in the derivation of the model, an advantage only a few of the modelling methods described in chapter 2 possess.

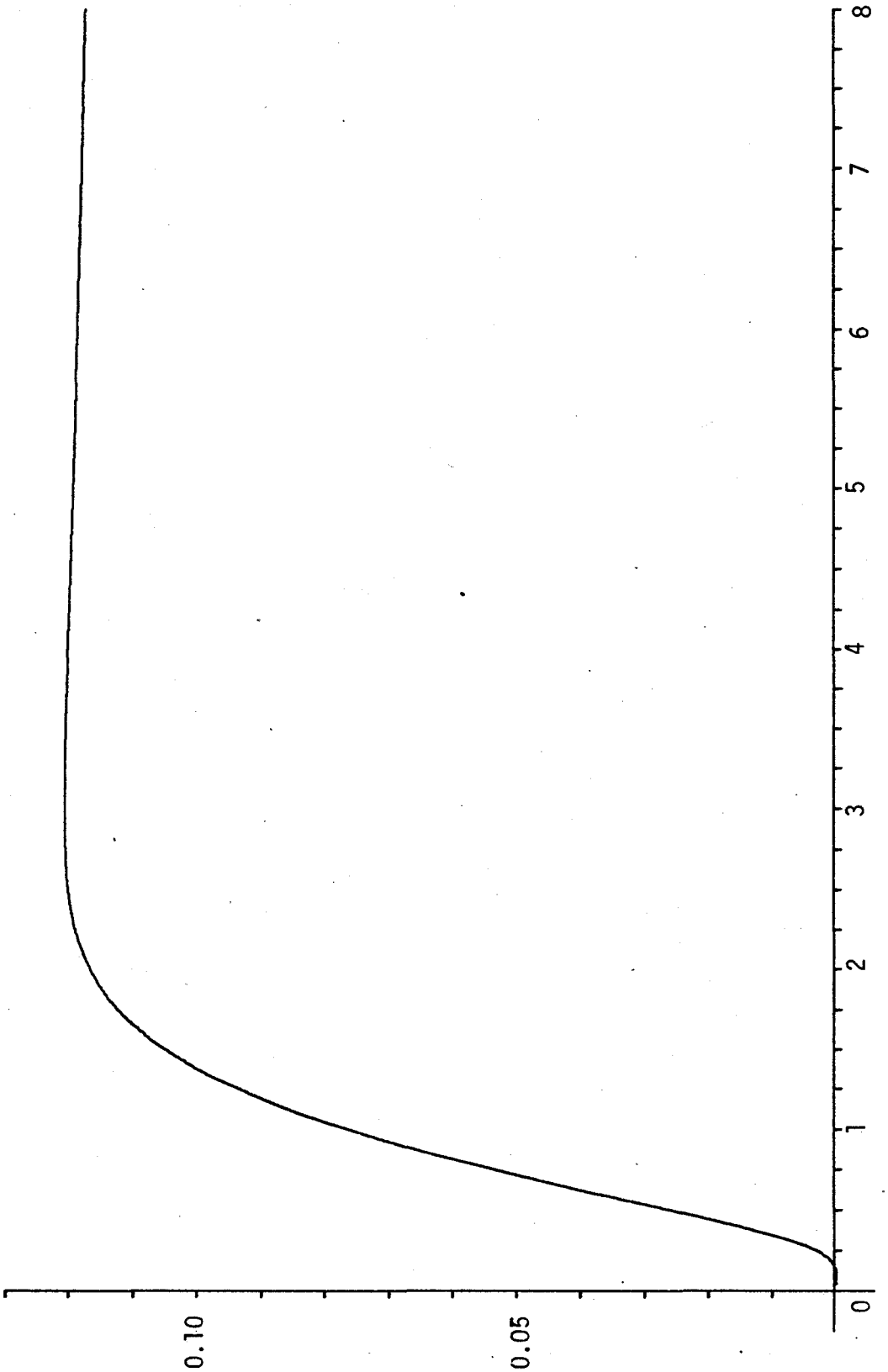

Fig. 1. Response of the system to a unit step.

The main features of the response are:

10% to 90% rise time	=	1 second
Steady-state value	=	0.11111
Initial slope	=	0
Time to reach first peak	=	2.9 seconds
Response at maximum overshoot	=	0.12069
Maximum overshoot	=	8.62%

Figure 1 shows the response of the system to a unit step. Second-order and third-order models of the above mentioned system can have the following forms

$$M_1(s) = \frac{b_0}{s^2 + a_1 s + a_0} \tag{26}$$

$$M_2(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \tag{27}$$

$$M_3(s) = \frac{b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{28}$$

$$M_4(s) = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{29}$$

$$M_5(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{30}$$

From the above models $M_1(s)$, $M_2(s)$ and $M_5(s)$ were selected to approximate the system, the most useful being $M_1(s)$ since this is the simplest and by the use of the matrix Ricatti equation an optimal control can be derived

easily [22]. The other two models M_2 and M_5 were also included to test the methods as the number of variables increased.

A steady state constraint can be imposed by use of the final value theorem, however, this is further discussed in the next chapter.

CHAPTER 5

RESULTS

Computational Information

The computer used for all the problems was a CDC6400. The approximation was made over 0 to 8 seconds, with 21 uniformly spaced points for the least squares case and 101 uniformly spaced points for the least pth approximation.

The terminating criterion for the Fletcher-Powell method, was set at 1.0×10^{-6} and the algorithm terminated if the change in the objective function or parameters was less than that number. The terminating criterion for the Fletcher method was also set at 1.0×10^{-6} and the algorithm terminated if the change in parameters was less than this number. In the Jacobson-Oksman algorithm there is the facility that the algorithm terminates when the change in the objective function is less than a number, set at 1.0×10^{-6}, and also when the norm of the gradient becomes smaller than a number set at 1.0×10^{-9}. Listings of the programs for the least pth approximation case is given in Appendix B.

5.1 Least Squares Models

As mentioned in chapter 4 three model transfer functions were selected to approximate the system in a certain sense. For least squares models, the objective function to be minimized is

$$F(\underset{\sim}{x}) = \sum_{i \in I} e_i(\underset{\sim}{x})^2 \qquad (31)$$

where $e_i(\underset{\sim}{x})$ is the error between the model and the system response to a unit step and I is an index set relating to the sampled points at equal intervals of time. The sampling does not have to be done at equal intervals, however, as no prior information about the error function is assumed, there is no justification to vary the sampling rate.

Having decided on a uniform sampling rate, the next question is how many sample points should be taken. If the number of sample points is too large, then computational time will be wasted. If the number is too small, then there will be insufficient information about the error function included in the objective function. The number of sample points used for least squares approximation was 21 over an interval of 0 to 8 seconds;with one exception this number was found satisfactory.

The gradients defined in chapter 3, can be shown to have the form

$$\underset{\sim}{\nabla} F(\underset{\sim}{x}) = 2 \sum_{i \in I} e_i(\underset{\sim}{x}) \; \underset{\sim}{\nabla} e_i(\underset{\sim}{x}) \qquad (32)$$

where $e_i(\underset{\sim}{x})$ is real. Since $e_i(\underset{\sim}{x})$ is the difference between the response of the system and the response of the model and since the system response is independent of the model parameters, $\underset{\sim}{\nabla} e_i(\underset{\sim}{x})$ is the gradient of the model response. Gradients for the responses are given in the appendix. Although the gradients are given in the Laplace form, the inverses can be found in any book of Laplace tables [23].

5.1.1. Unconstrained Least Squares Approximation

First the transfer function given by expression (26) was used to model the system using a least squares error criterion. The response of the model to a unit step input was derived analytically, by considering separately the real and complex conjugate roots of

$$q(s) = s^2 + a_1 s + a_0 \qquad (33)$$

The objective function shown in expression (31) had been formulated and gradients corresponding to that objective function, had been analytically derived. A simple perturbation test was done to ensure that the expressions for the gradients were correct. Since this was the first of the modelling problems to be tackled and little experience had at this stage been gained with the new minimization methods, the technique used was Fletcher-Powell.

The program was run from several arbitrary starting points. Although the algorithm appeared to converge, the terminating point point was not unique. For the purpose of this work, a point was considered unique if there was agreement to within four significant figures amongst the corresponding elements. Some of the terminating points are shown in table 1, with the corresponding objective function values. The difference in the objective function values appeared to be small, but even so the algorithm should have proceeded further. A close examination of the terminating points indicated that a linear dependence existed amongst the terminating points. This is shown in the lower half of table 1.

	$F \times 10^4$	$b_0 \times 10$	a_0	a_1	b_0/a_0
1	3.9149	3.8127	3.2945	2.5584	0.1157
2	3.8708	3.9621	3.4213	2.6581	0.1158
3	3.8719	3.9817	3.4381	2.6709	0.1158
4	3.8790	4.0238	3.4743	2.6990	0.1158
Ratio of Row 2 to Row:					
1		1.0392	1.0385	1.0390	
3		0.9951	0.9951	0.9952	
4		0.9847	0.9847	0.9848	

TABLE 1. Terminating points for the least squares unconstrained problem and indication of linear dependence between points.

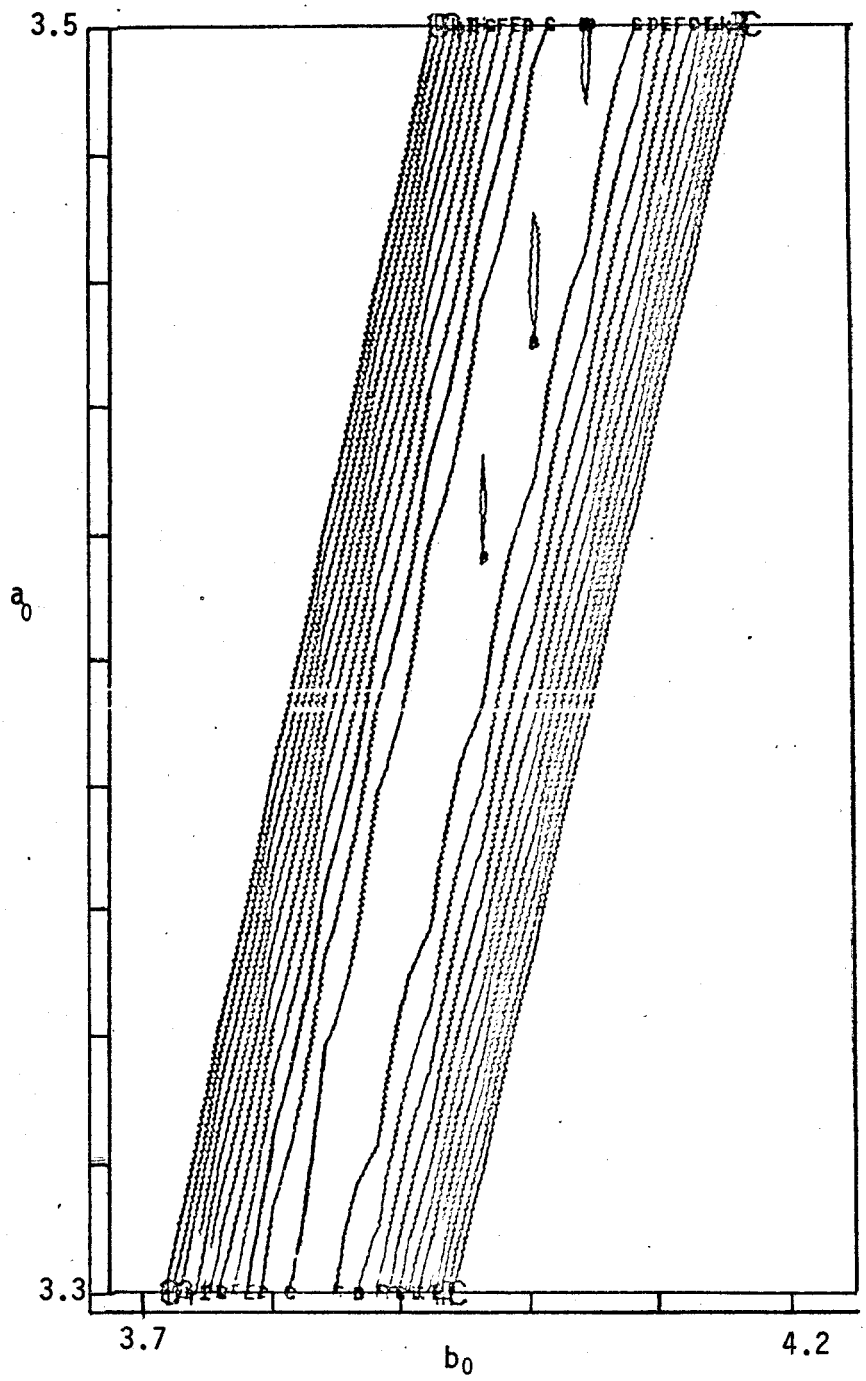

Fig. 2a. Contours of the unconstrained objective function (a_1= 2.70382)

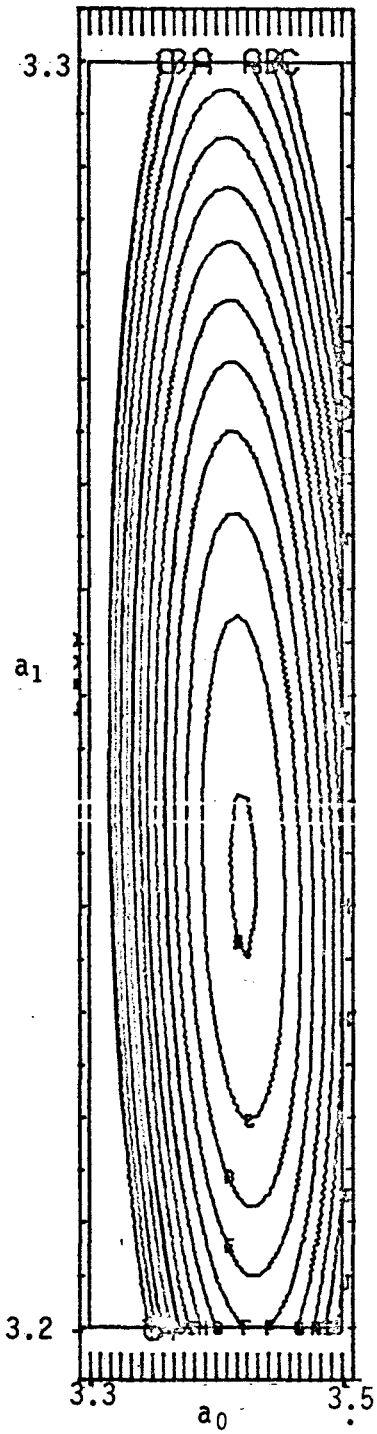

Fig. 2b. Contours of the unconstrained objective function (b_0= 0.39261)

When using gradient techniques with Newton-type termination, where information about the inverse Hessian is utilized, singularities in the Hessian are very critical. Some contours of the objective function were drawn in the area where the algorithm terminated. The contours are shown in Figures 2 a-b. Figure 2a shows the contours of the objective function when holding a_1 constant and vary a_0 and b_0 . The terminating points seem to lie on a straight line through the origin, with a slope of 0.11580. By applying the final value theorem to the model transfer function, for a unit step input, we can see that b_0/a_0 is the steady-state value of the model.

The linear relationship between the terminating points is due to the linear relationship between a_0 and b_0. There are many interpretations to this, however, the simplest is that $b_0 = E\ a_0$ define a plane through the origin in the three dimensional space and the points in table 1 lie in that plane.

5.1.2. Constrained Least Squares Approximation

If a comparison between minimization techniques on the modelling problem was to be made, a unique solution was desirable. A steady-state constraint, therefore, was considered necessary for our purpose. Since the equality constraint was in a very simple linear form, a substitution was made of b_0 by $E\ a_0$.

The steady state value, E, for the model to a step input can be found either from the transfer function of the system, which in our

problem will be 0.11111 or, if the transfer function of the system is not available, then the final sample point of the system response can be used, which in our problem will be 0.11706.

With the steady state constraint imposed the second-order without zeros model shown by equation (26) now becomes

$$M_1(s) = \frac{Ea_0}{s^2 + a_1 s + a_0} \tag{34}$$

with E = 0.11111. Here, as in all the following problems, real and complex conjugate roots were considered separately. At least three different starting points were tried for each of the three minimization techniques and the algorithms ultimately converged to the same unique parameters

$$a_0 = 3.19591$$
$$a_1 = 2.28106$$

with the value of the objective function 7.50758×10^{-4} and the components of the gradient less than 1.0×10^{-9}. Figure 3 shows the corresponding response. Table 2 compares the number of function evaluations required to reach the objective function value of 7.50759×10^{-4} this value being 1.0×10^{-9} higher than the solution ultimately obtained. In the table it is shown that both the Fletcher method and the Jacobson-Oksman method

STARTING POINT	FLETCHER	JACOBSON-OKSMAN	FLETCHER-POWELL
0.5 0.5	22	19	49
3.0 2.0	21	21	29
1.0 1.0	19	14	32

TABLE 2. Number of function evaluations required to reach the objective function value 7.50759×10^{-4} for the two-parameter least squares approximation problem.

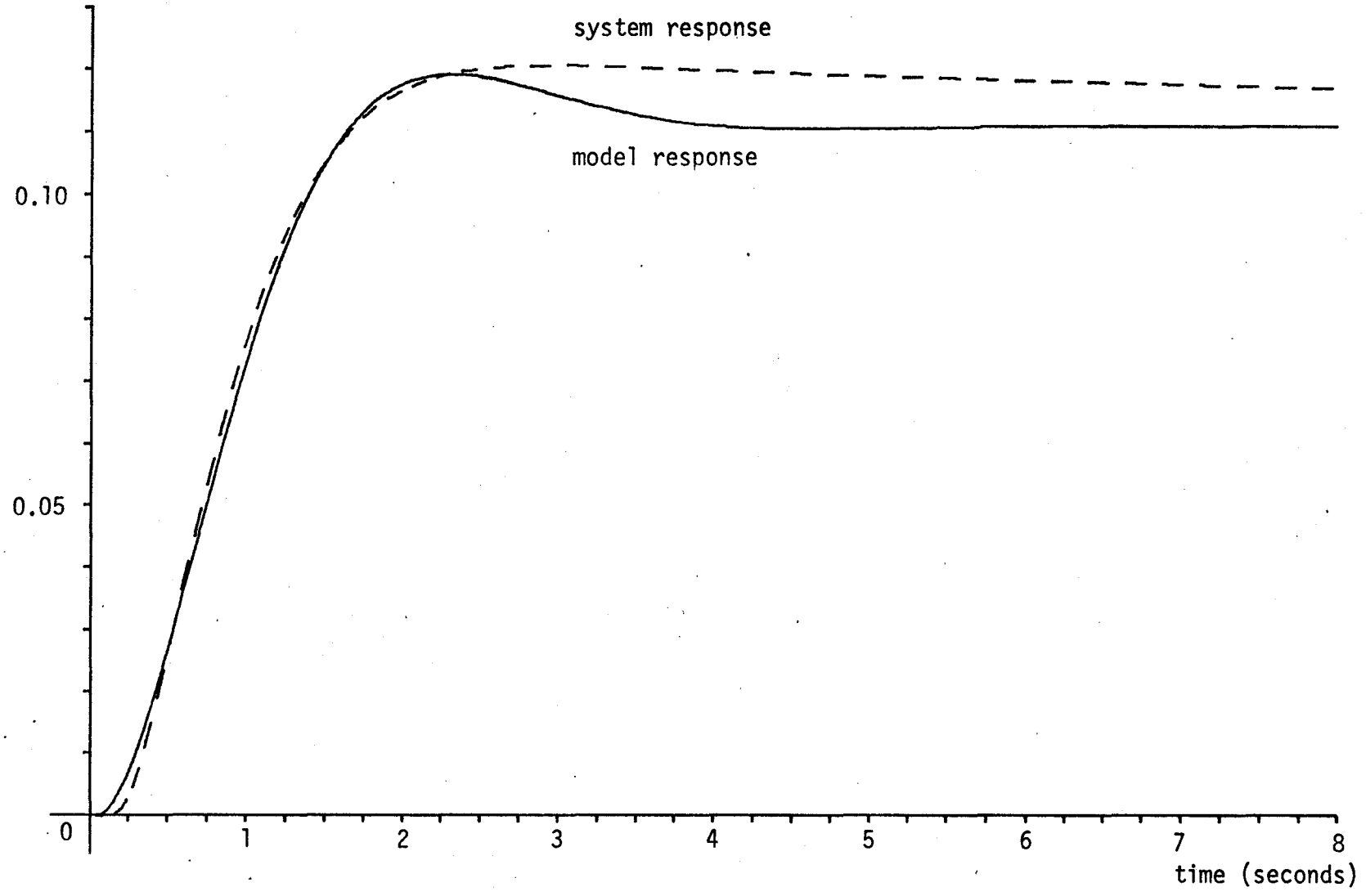

Fig. 3. Response for two-parameter least squares problem (E = 0.11111).

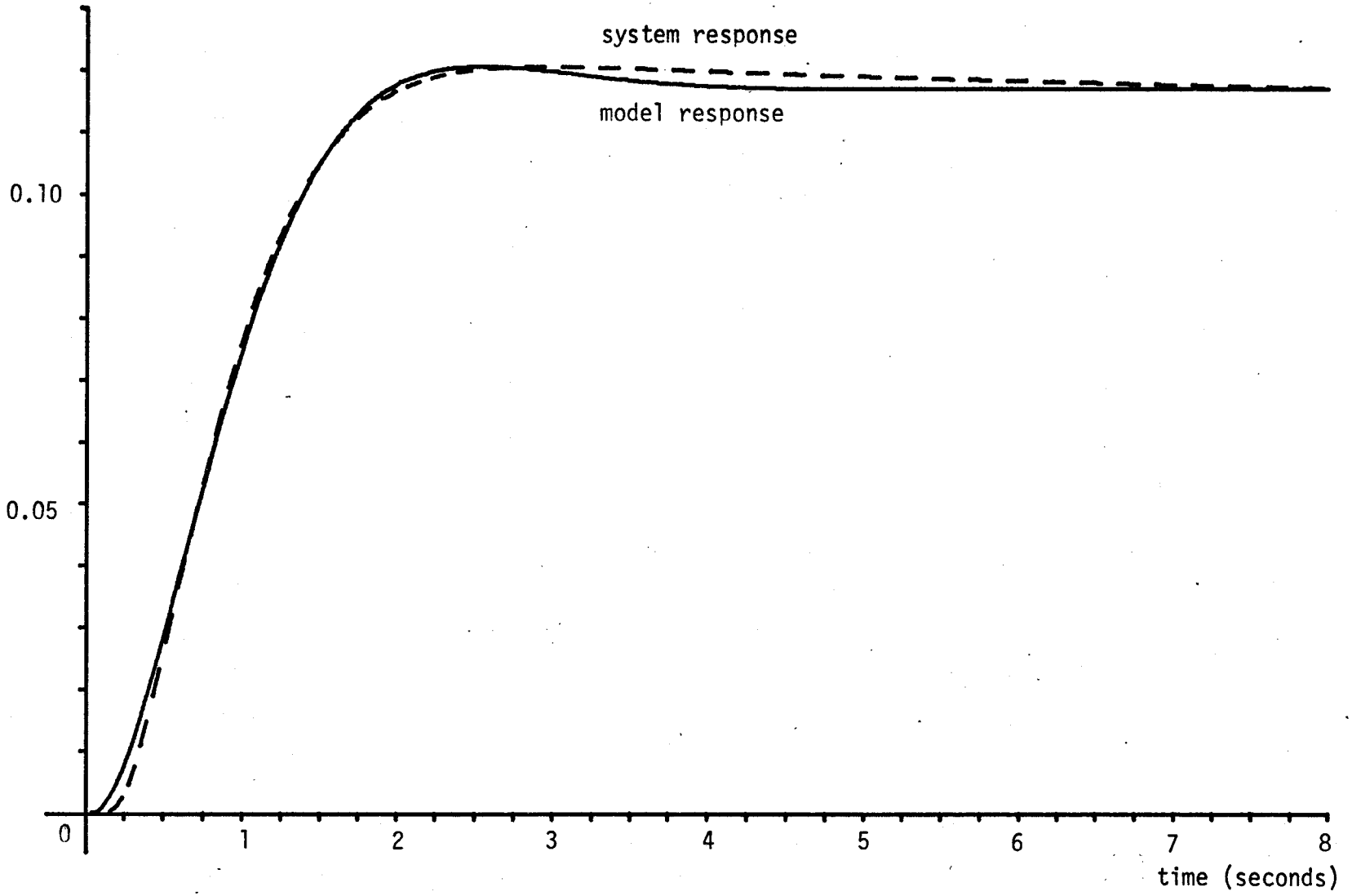

Fig. 4a. Response for two-parameter least squares problem (E = 0.11706)

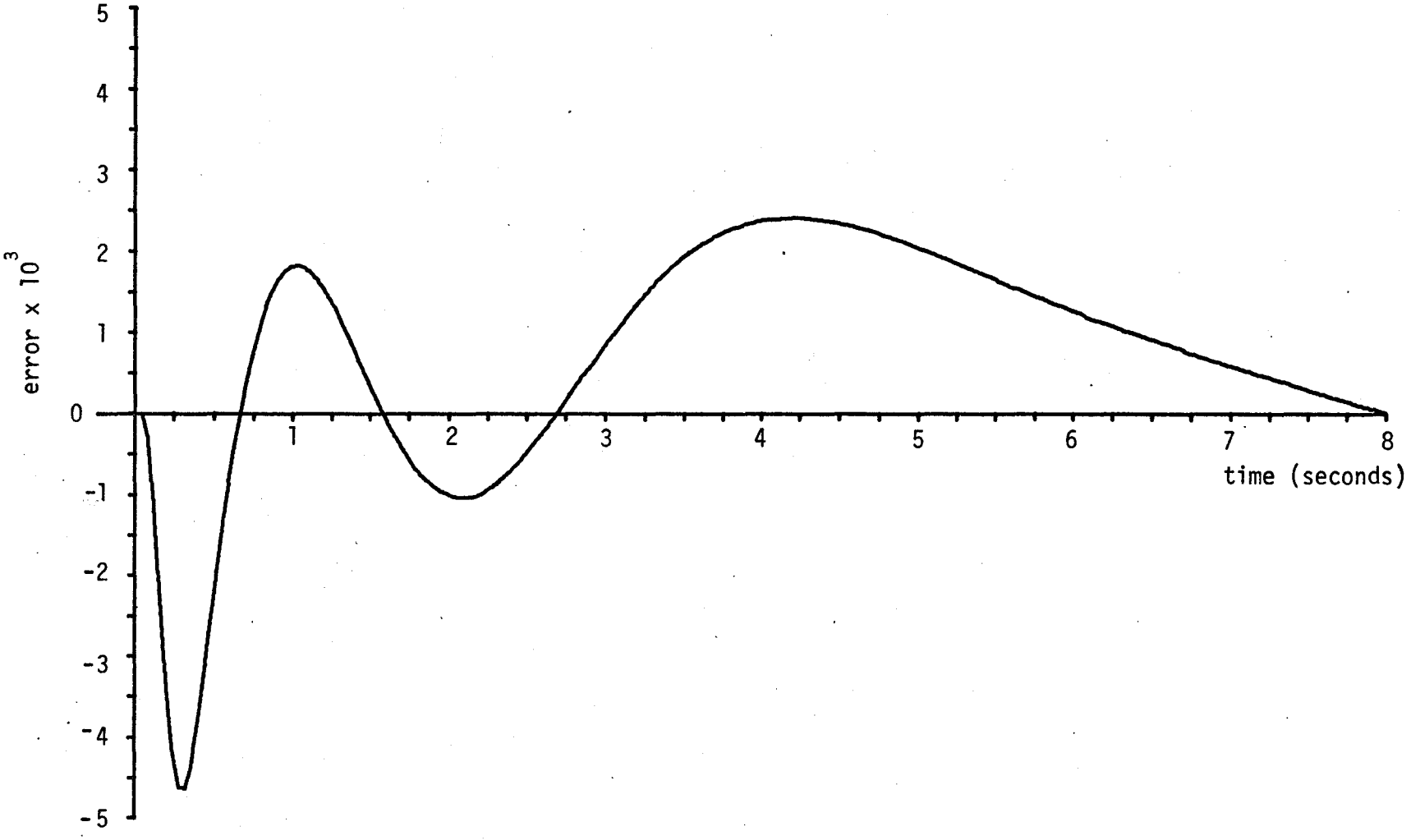

Fig. 4b. Error curve for two-parameter least squares problem (E = 0.11706)

performed better than the Fletcher-Powell method.

With E = 0.11706, the value of the system response at 8 seconds, the optimum parameters obtained were

$$a_0 = 3.47571$$
$$a_1 = 2.76681$$

giving an optimum objective function value of 4.75016×10^{-5}. Figures 4a and 4b show the corresponding response and the error curve, respectively.

If the steady-state constraint is imposed on the model based on expression (27) then the transfer function is

$$M_2(s) = \frac{b_1 s + Ea_0}{s^2 + a_1 s + a_0} \qquad (35)$$

resulting in a three-parameter problem. With E = 0.11111 three different starting points were tried. In each case the algorithms converged to the same optimum parameters

$$a_0 = 1.99740$$
$$a_1 = 1.66066$$
$$b_1 = 4.37072 \times 10^{-2}$$

giving an optimum objective function value of 1.58222×10^{-4} and gradient components less than 1.0×10^{-9}. Figure 5 shows the corresponding response. Table 3 compares the number of function evaluations required by each method to reach the objective function value of 1.58225×10^{-4}. Here again it is seen that the Fletcher method and the Jacobson-Oksman method are superior to the Fletcher-Powell method. There was one case not shown in the table, however, when the Jacobson-Oksman method failed.

With E = 0.11706 the 3-parameter problem converged to the optimum values of

$$a_0 = 3.97531$$
$$a_1 = 3.03900$$
$$b_1 = -2.08787 \times 10^{-2}$$

giving an objective function value of 2.26148×10^{-5}. Figure 6a shows the corresponding response and Figure 6b shows the error curve for that response.

For a third-order model the transfer function in expression (30) was used having two zeros. For ease of computation the model transfer function was put in the form

STARTING POINT	FLETCHER	JACOBSON-OKSMAN	FLETCHER-POWELL
1.0 1.0 1.0	27	39	60
0.5 2.0 4.0	76	39	274
1.0 0.5 0.1	35	29	58

TABLE 3. Number of function evaluations required to reach the objective function value 1.58223×10^{-4} for the three-parameter least squares approximation problem.

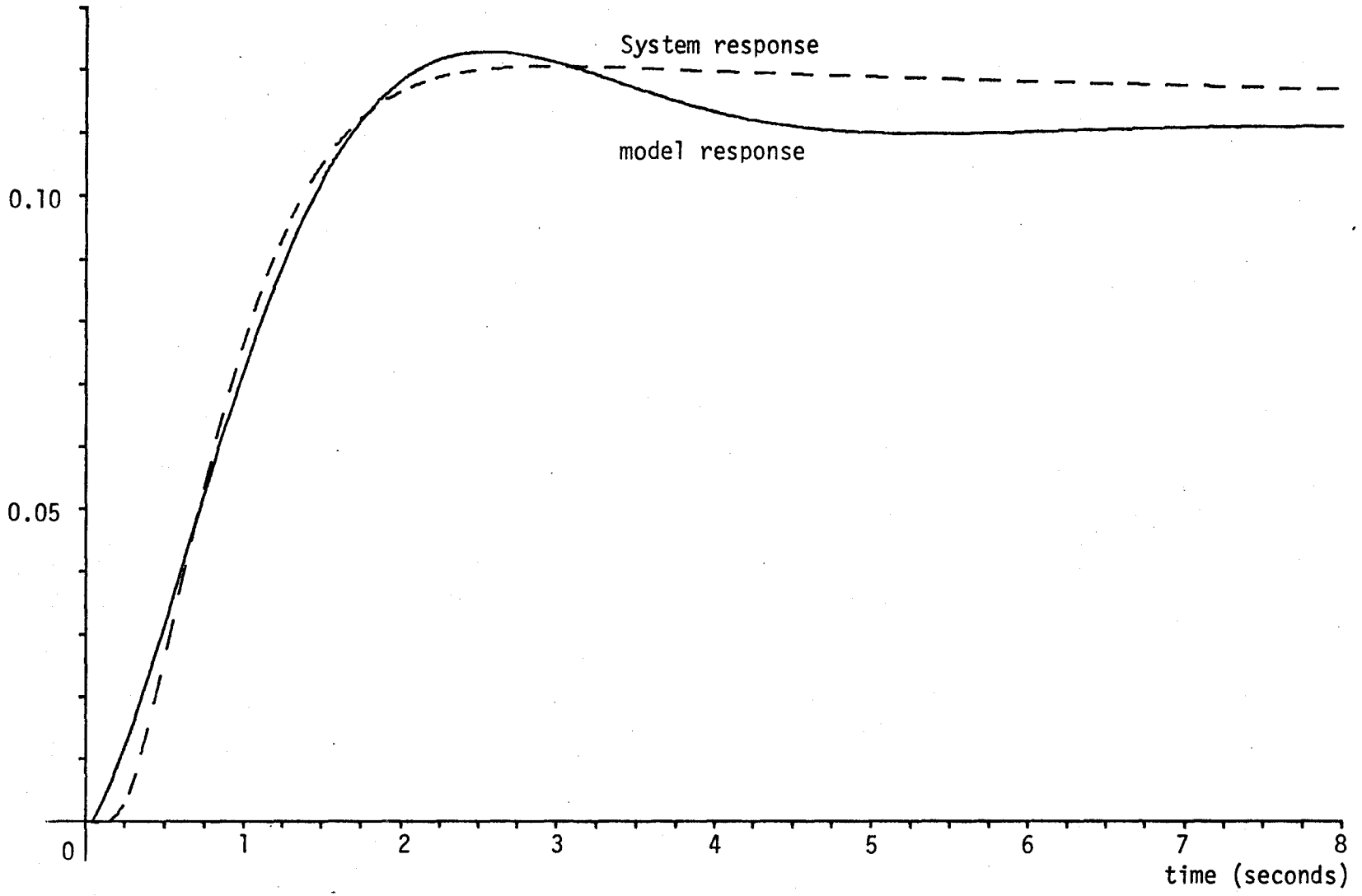

Fig. 5. Response for three-parameter least squares problem (E = 0.11111)

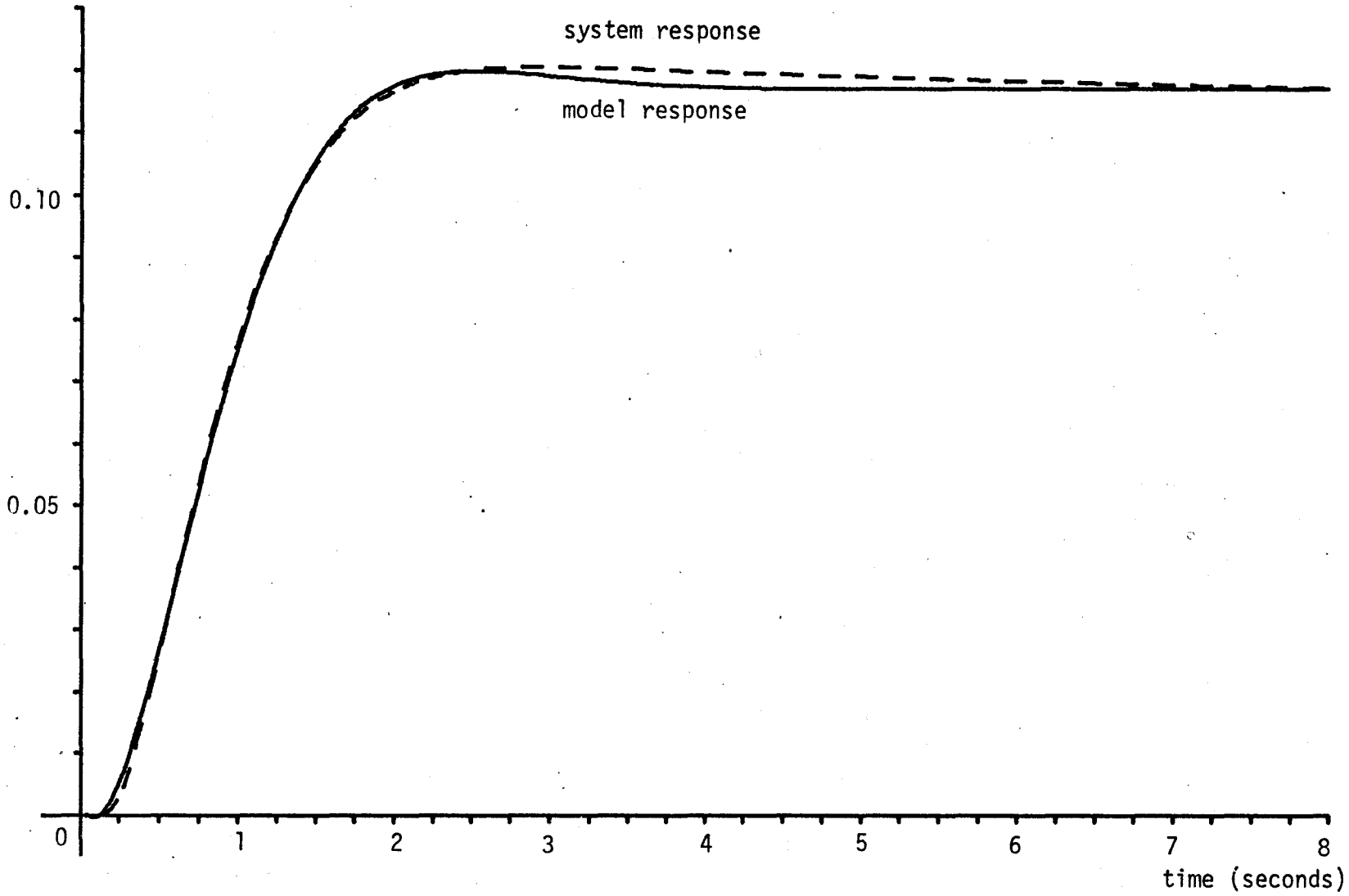

Fig. 6a. Response for three-parameter least squares problem (E = 0.11706).

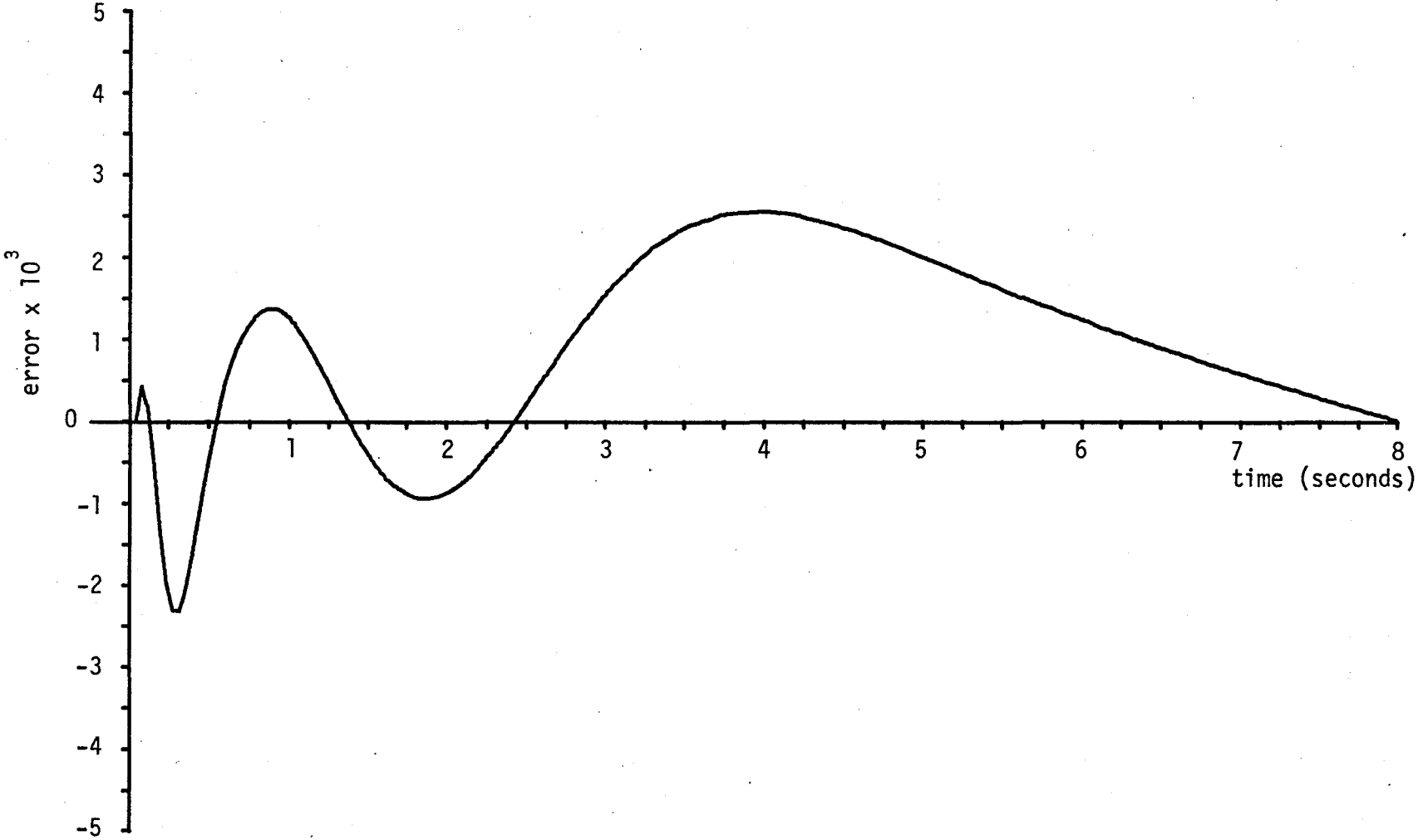

Fig. 6b. Error curve for three-parameter least squares problem (E = 0.11706)

$$M_5(s) = \frac{b_2 s^2 + b_1 s + E x_1 x_3}{(s + x_3)(s^2 + x_2 s + x_1)} \qquad (36)$$

where E, the steady state constraint was set at 0.11706. A number of starting points were considered in an effort to get convergence with the Jacobson-Oksman method, but with the exception of one case, it always failed. The Fletcher method in every case converged to the same optimum parameters, which were

$$x_1 = 1.02741$$
$$x_2 = 2.85536$$
$$x_3 = 2.30125$$
$$x_4 = 6.62057 \times 10^{-1}$$
$$x_5 = -7.6045 \times 10^{-2}$$

giving an objective function value of 1.02741×10^{-6}. Figures 7a and 7b show the corresponding response and error curve, respectively. The Fletcher-Powell method appeared to be much slower and converged to the optimum in only one of the cases tried, while in the other cases the time limit of 64 seconds was reached. The results of this problem are shown in Table 4.

The error curve would indicate that for this problem 21 sample points was insufficient and as a result there was a large initial error.

STARTING POINT	FLETCHER	JACOBSON-OKSMAN	FLETCHER-POWELL
1.25 2.8 2.3 0.7 -0.1	535 48 seconds	88 1.027952×10^{-6} 12 seconds	225 22 seconds
5.0 4.5 3.5 4.0 2.0	140 16 seconds	failed	780 1.528774×10^{-6}
3.2 0.8 5.3 -2.6 2.1	465 45 seconds	failed	745 8.449308×10^{-6}
1.0 1.5 3.0 4.0 5.0	298 30 seconds	failed	failed
2.5 1.5 3.5 0.1 31.0	132 15 seconds	failed	800 1.02883×10^{-6}

TABLE 4. Number of function evaluations required to reach the objective function value of 1.027406×10^{-6} and central processor time if it is less than the time limit of 64 seconds, or the objective function value reached in that time.

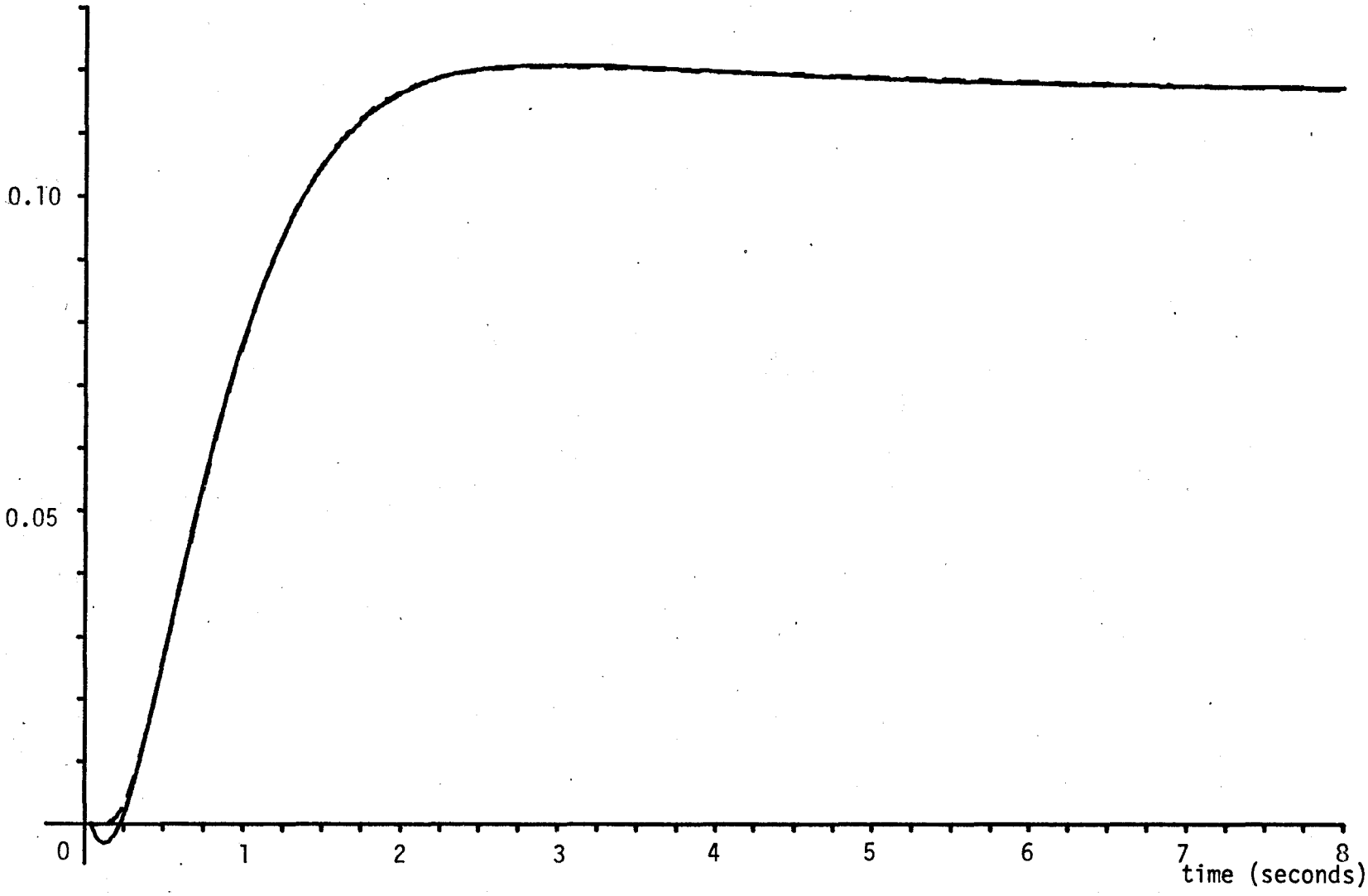

Fig. 7a. Response for five-parameter least squares problem (E = 0.11706)

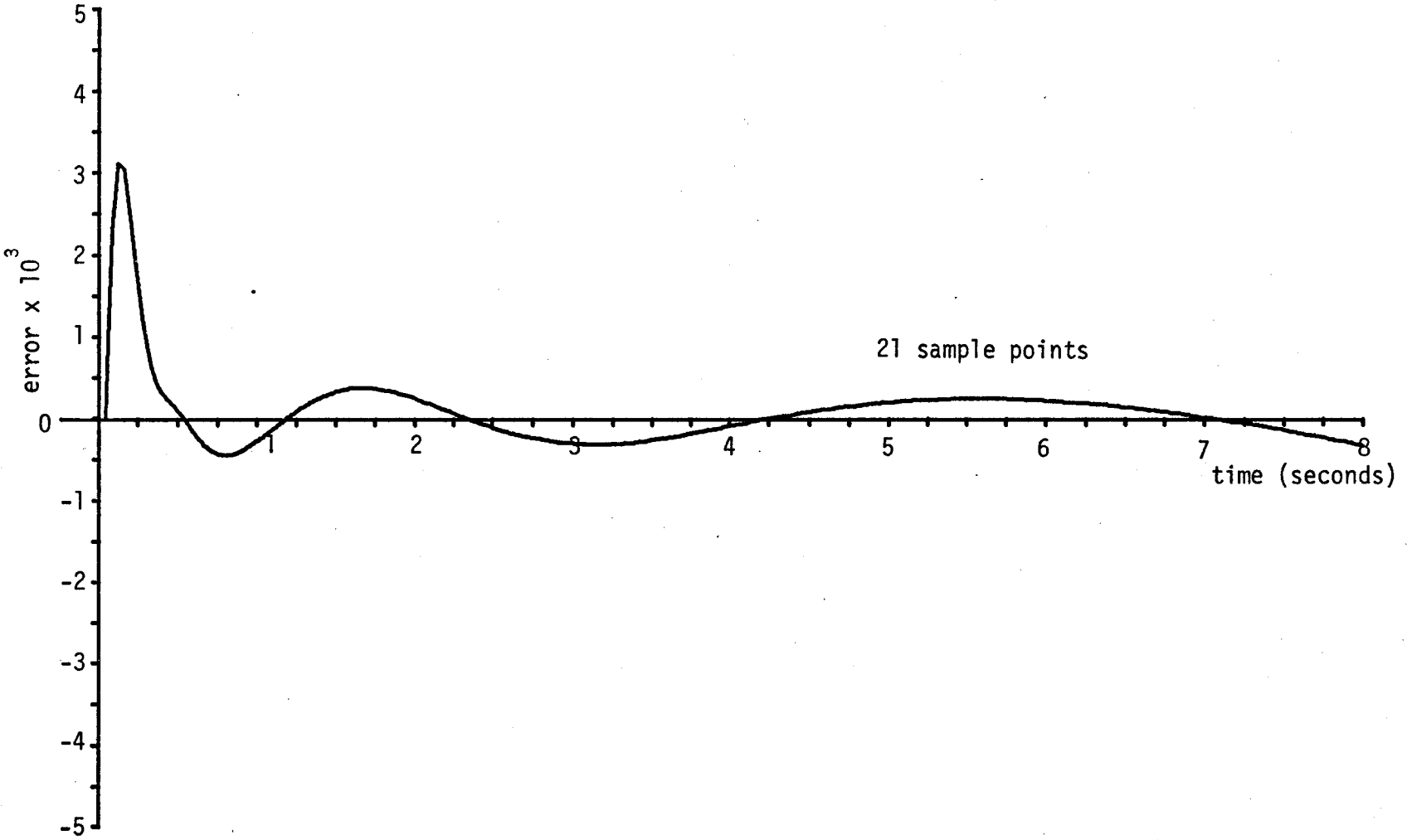

Fig. 7b. Error curve for five-parameter least squares problem (E = 0.11706)

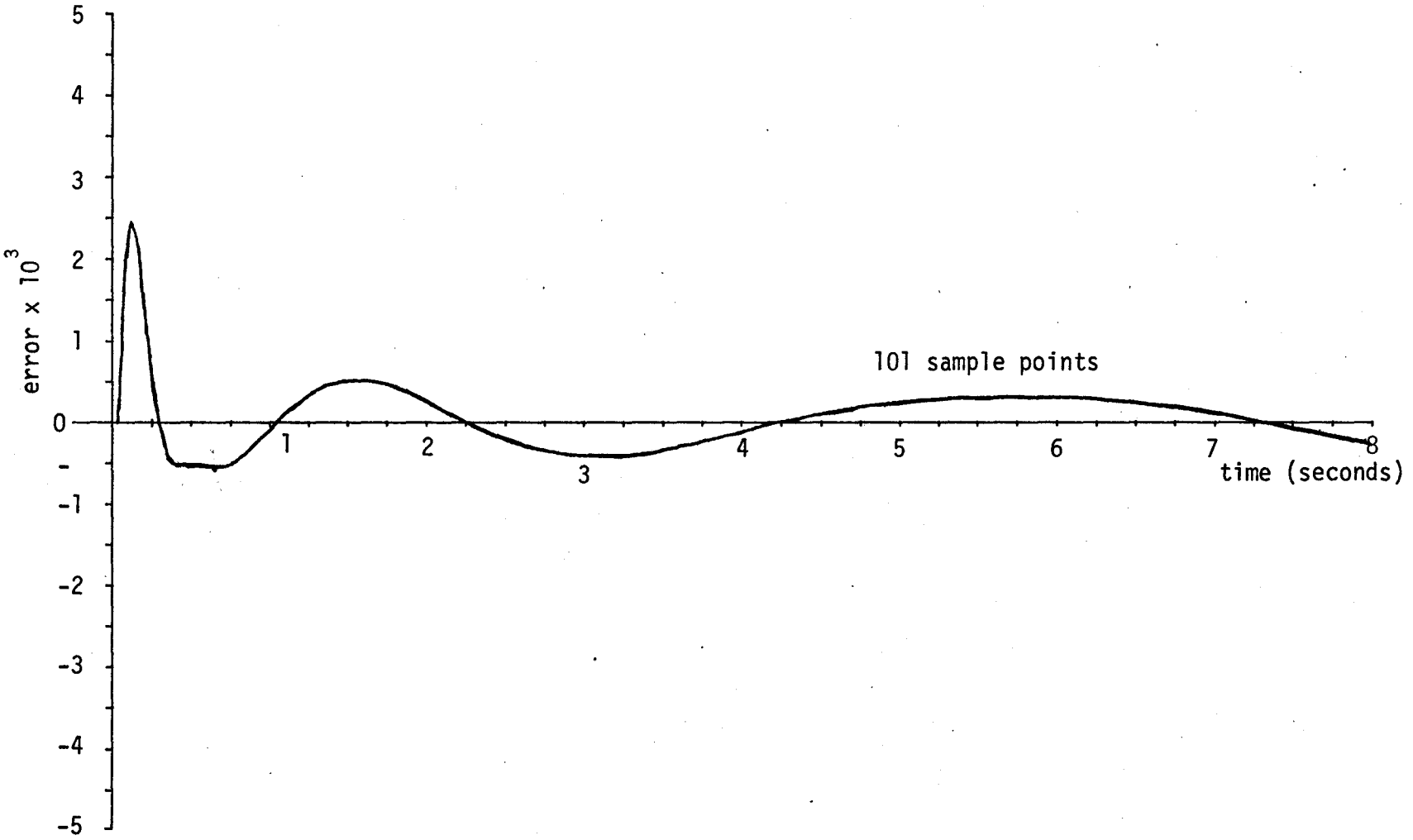

Fig. 7c. Error curve for five-parameter least squares problem (E= 0.11706)

For this reason a larger number (101) of sample points were taken and the program repeated. The otpimum parameters were

$$x_1 = 1.34731$$
$$x_2 = 2.84002$$
$$x_3 = 2.23786$$
$$x_4 = 6.38527 \times 10^{-1}$$
$$x_5 = -6.48648 \times 10^{-2}$$

The error curve is shown in Figure 7c. It can be seen that by using 101 points the maximum error is reduced by approximately 20% of the maximum error with 21 sample points.

Figures 8a and 8b show some convergence curves for the two-parameter problem, while Figure 8c shows a convergence curve for the three-parameter problem. The curves show the difference of the objective function from the assumed optimum, on a logarithmic scale versus the number of function evaluations.

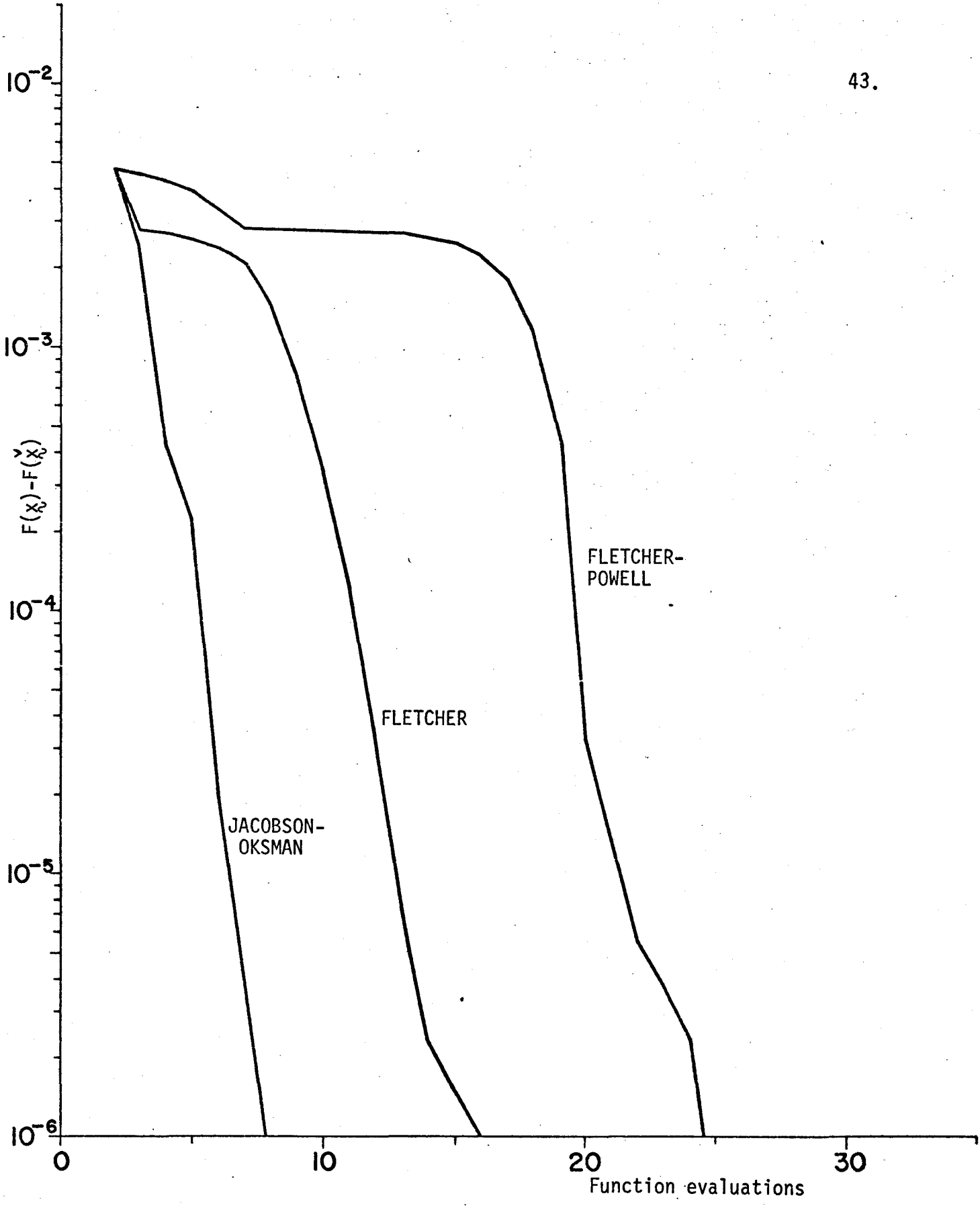

Fig. 8a. Convergence curves for two-parameter least squares problem (E = 0.11111).

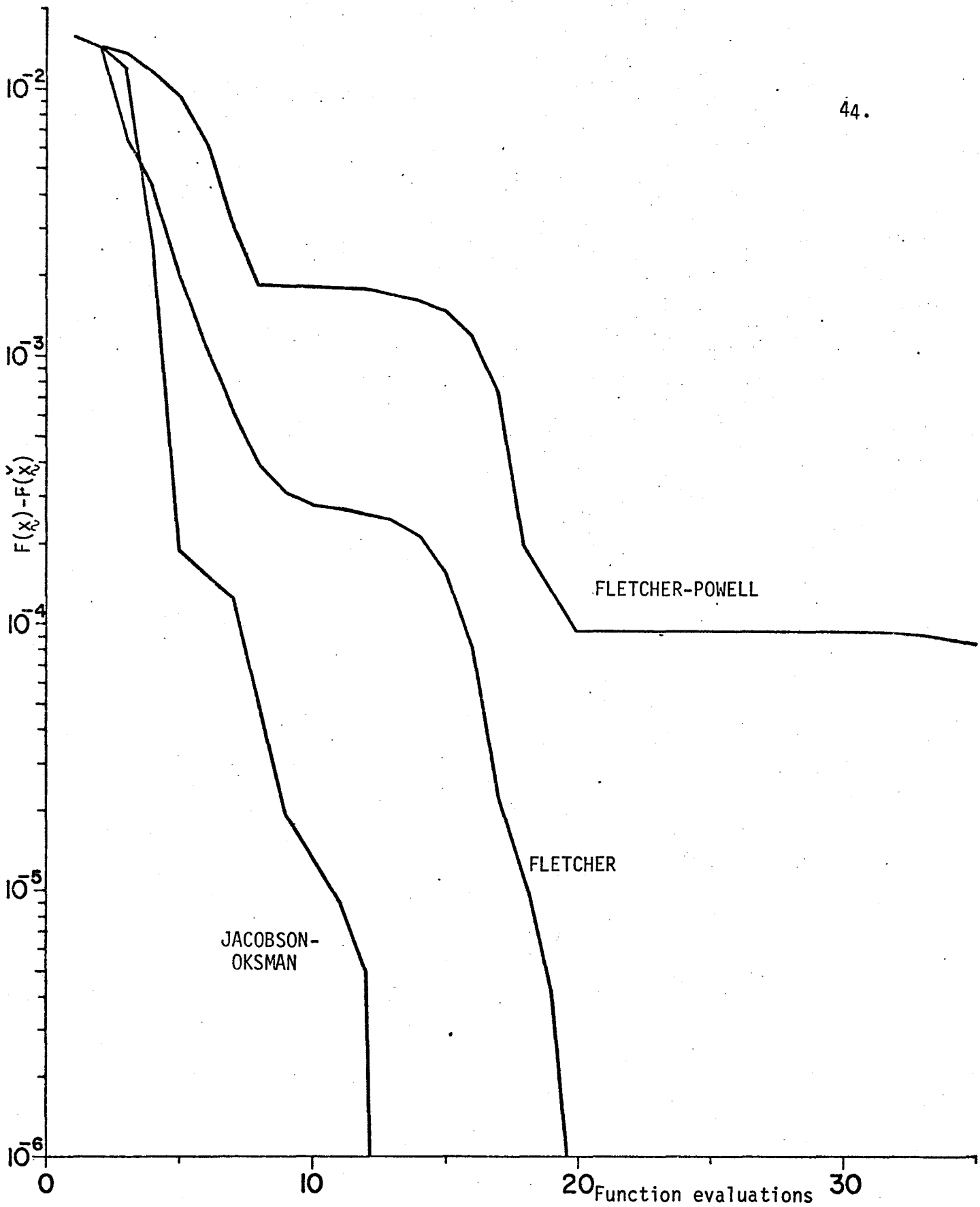

Fig. 8b. Convergence curves for two-parameter least squares problem (E = 0.11111).

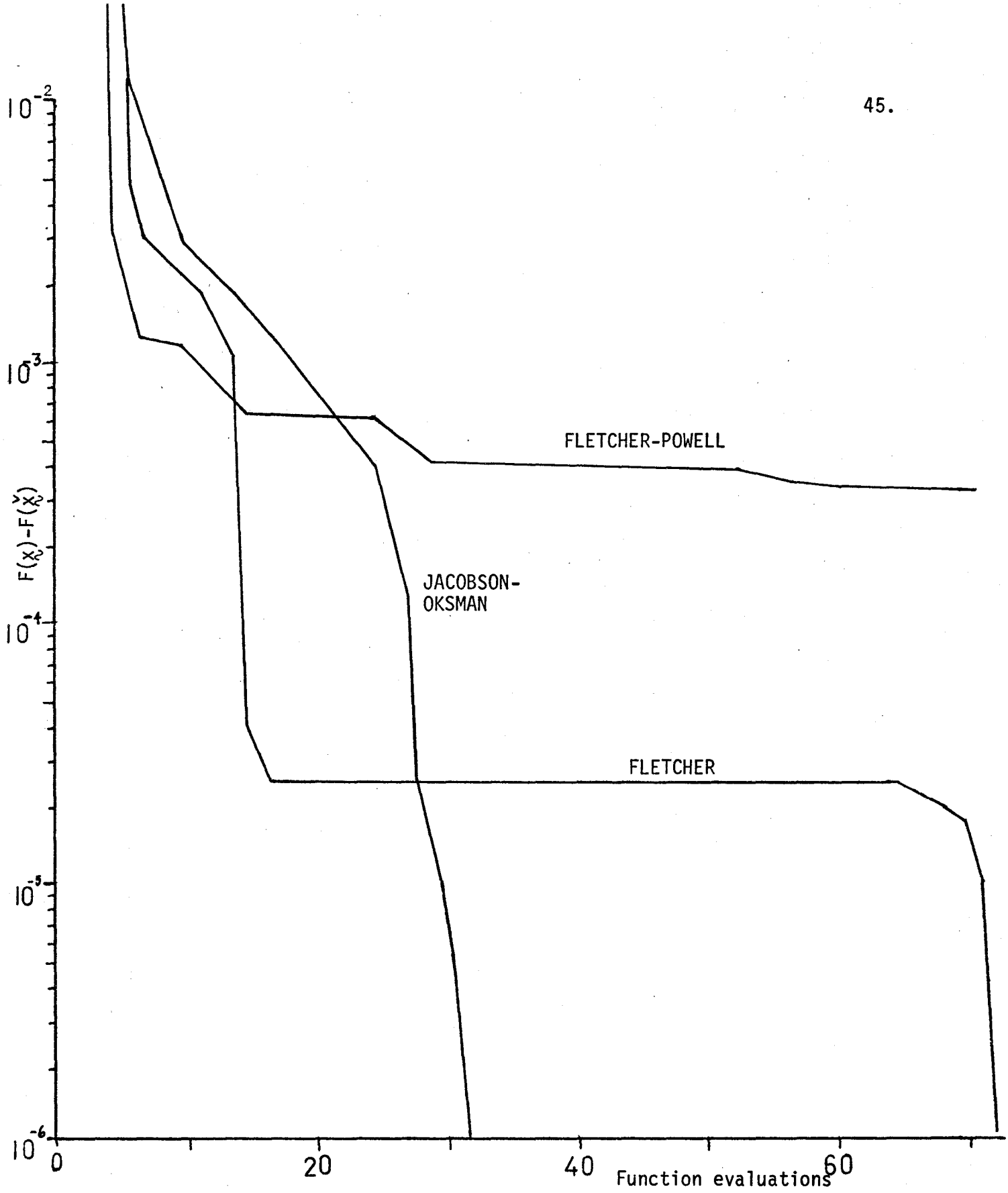

Fig. 8c. Convergence curves for three-parameter least squares problem (E = 0.11111)..

5.2 Least pth Approximation

Although a least squares criterion can give us an acceptable model, in many cases it may be desirable to derive a model, where the maximum error between the system and the model response is minimized. This model will give a near minimax error and the response ultimately obtained, should be a closer approximation to the response of the system.

In attempting this problem one would be tempted in simply defining as an objective function the maximum absolute error and minimizing it. This will rarely work. The reason is that as two of the extrema approach each other, by considering only the larger extremum, in absolute value, a parameter change that gives a decrease in that extremum, might cause the other one to increase. This could result in oscillations and often false local optima.

An alternative approach would be in defining a least pth objective function

$$F(\underset{\sim}{x}) = \left(\sum_{i \in I} \left| e_i(\underset{\sim}{x}) \right|^p \right)^{\frac{1}{p}} \qquad p>1 \qquad (37)$$

where $e_i(\underset{\sim}{x})$ represents the error between the system and the model responses at some sample point i of a finite set I, relating to all the sampled points. It is assumed that $e_i(\underset{\sim}{x})$ is continuous with continuous partial derivatives for all i.

It is desirable to increase the value of p as much as possible, since the larger the value of p the nearer to minimax should the solution be. There are two computational limitations,however. One is that if $|e_i(\underset{\sim}{x})| > 1$, when using large values of p, the numbers tend to become too large for the computer to handle. The other one is that if $|e_i(\underset{\sim}{x})| < 1$ when raised to a large power, the numbers tend to zero and most of the information is lost.

A normalization proposed by Bandler and Charalambous [18], permitted the use of extremely large values of p. The objective function used is

$$F(\underset{\sim}{x}) = M(\underset{\sim}{x}) \left(\sum_{i \in I} \left| \frac{e_i(\underset{\sim}{x})}{M(\underset{\sim}{x})} \right|^p \right)^{\frac{1}{p}} \tag{38}$$

where

$$M(\underset{\sim}{x}) \triangleq \max_{i \in I} \left| e_i(\underset{\sim}{x}) \right| \tag{39}$$

where $e_i(\underset{\sim}{x})$ and I are defined above. Values of p up to 10^{12} have been used.

In the objective function in (38), if I is replaced by J, an index set relating to the extrema of the error function, considerable economies in computing time will result at a slightly greater risk of creating false optima. Moreover, as can be seen from the expression of the gradient

$$\nabla F(\underset{\sim}{x}) = \left(\sum_{i \in I} \left| \frac{e_i(\underset{\sim}{x})}{M(\underset{\sim}{x})} \right|^p \right)^{\frac{1}{p} - 1} \left(\sum_{i \in I} \left| \frac{e_i(\underset{\sim}{x})}{M(\underset{\sim}{x})} \right|^{p-2} \frac{e_i(\underset{\sim}{x})}{M(\underset{\sim}{x})} \nabla e_i(\underset{\sim}{x}) \right) \qquad (40)$$

the coefficients of $\nabla e_i(\underset{\sim}{x})$ will, for most points and large enough p, be very small, thus contributing very little to the gradient. Analytical expressions for $\nabla e_i(\underset{\sim}{x})$ appear in the appendix A.

Having formulated the objective function, the question now is what value of p should be used to obtain a minimax or near minimax approximation? With p = 2 we have a least squares type of formulation. Obviously, the higher value of p the more emphasis will be given to those deviations which are largest. So, since the requirement is to concentrate more on minimizing the maximum error, a sufficiently large value of p must be chosen. The basis of such a formulation [19] is the fact that

$$M(\underset{\sim}{x}) = \lim_{p \to \infty} F(\underset{\sim}{x}) \qquad (41)$$

So the value of p should be preferably as large as possible. The following values of p have been used 10, 10^2, 5 x 10^2, 10^3, 10^4, 10^6, 10^9, 10^{12}, and it was found that although agreement in significant figures increased as p increased, the central processing time increased considerably for values of p above 10^3. Thus for comparison of the minimization techniques p = 10^3 was considered suitable. In addition to the comparison of the three gradient techniques, a comparison is made in the tables,

with a more direct minimax technique called the grazor search technique [20]. This method solves a linear programming problem by using the gradient information of one or more largest extrema in the error function to produce a downhill direction. A linear search is carried out in this direction to find a minimum of $M(\underset{\sim}{x})$.

For the two-parameter problem, corresponding to the model given by equation (34), the steady-state value E was set at 0.11706, corresponding to the response of the system at the final sample point. The optimum parameters obtained, for p = 1000, were

$$a_0 = 3.06549$$
$$a_1 = 2.38414$$

giving an objective function value of 3.76618×10^{-3}. Figure 9a shows the corresponding response and Figure 9b shows the error curve corresponding to the response. There are 4 extrema and the largest in magnitude is 3.76510×10^{-3}.

Table 5 shows the number of function evaluations required to reach the objective function value of 3.76619×10^{-3}.x It is shown that the Fletcher method gives consistently good results. The Jacobson-Oksman method only once converged faster than the Fletcher method. Both the Fletcher and Jacobson-Oksman methods were found more efficient than the Fletcher-Powell method, however, from one starting point the Jacobson-Oksman algorithm diverged. The least pth approach with the Fletcher method performed better than the grazor technique. The values of the extrema with $p = 10^3$ were.

STARTING POINT	MINIMIZATION OF $F(\underset{\sim}{x})$			MINIMIZATION OF $M(\underset{\sim}{x})$
	FLETCHER	JACOBSON-OKSMAN	FLETCHER-POWELL	GRAZOR
3.0 2.0	47	46	73	107
1.0 1.0	82	127	346	130
1.0 4.0	98	failed	725	165
4.0 1.0	72	45	false optimum	129

TABLE 5. Number of function evaluations required to reach the objective function value 3.76619×10^{-3} for the two-parameter problem.

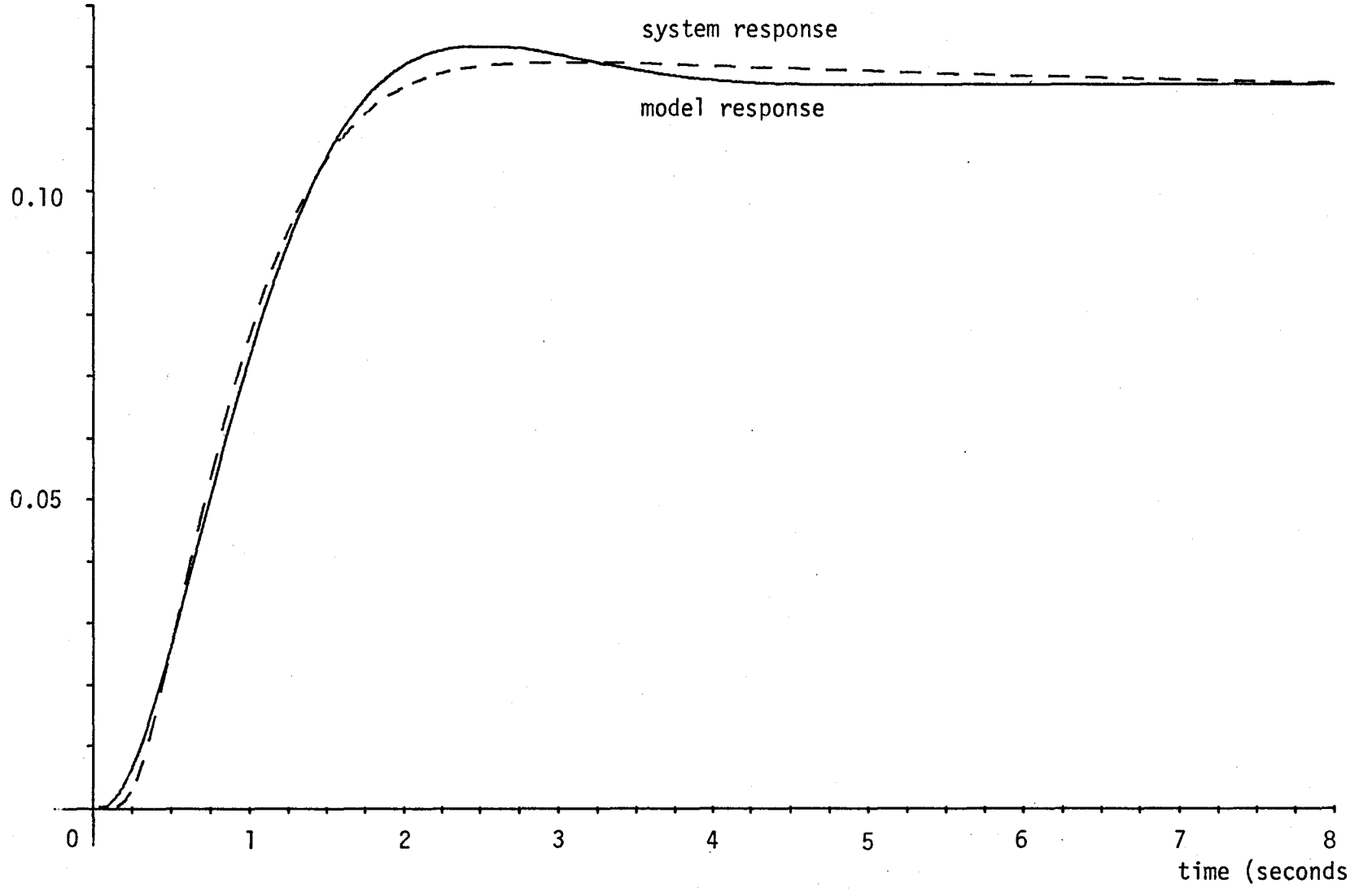

Fig. 9a. Response for two-parameter least pth problem (p = 1000)

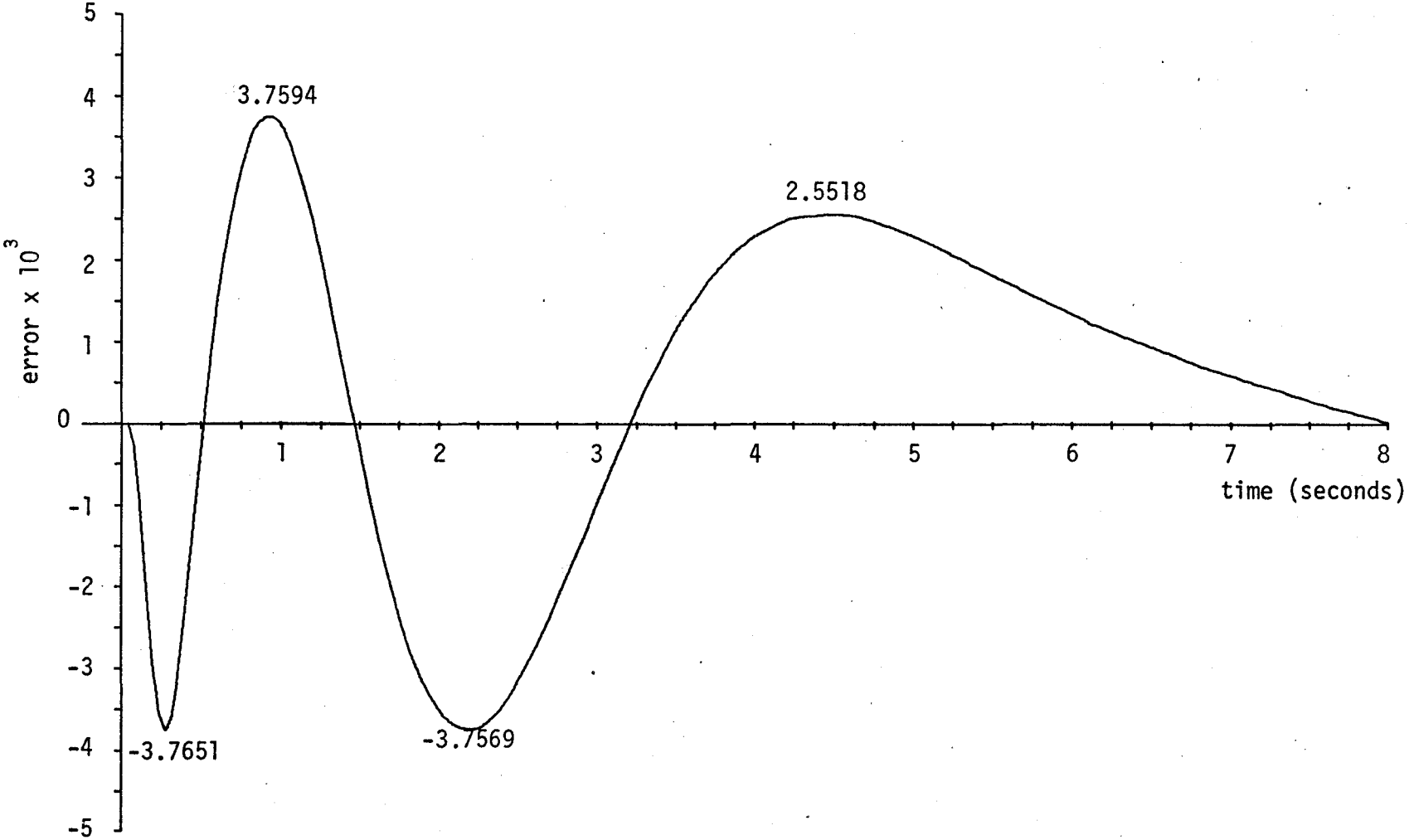

Fig. 9b. Error curve for two-parameter least pth problem (p = 1000)

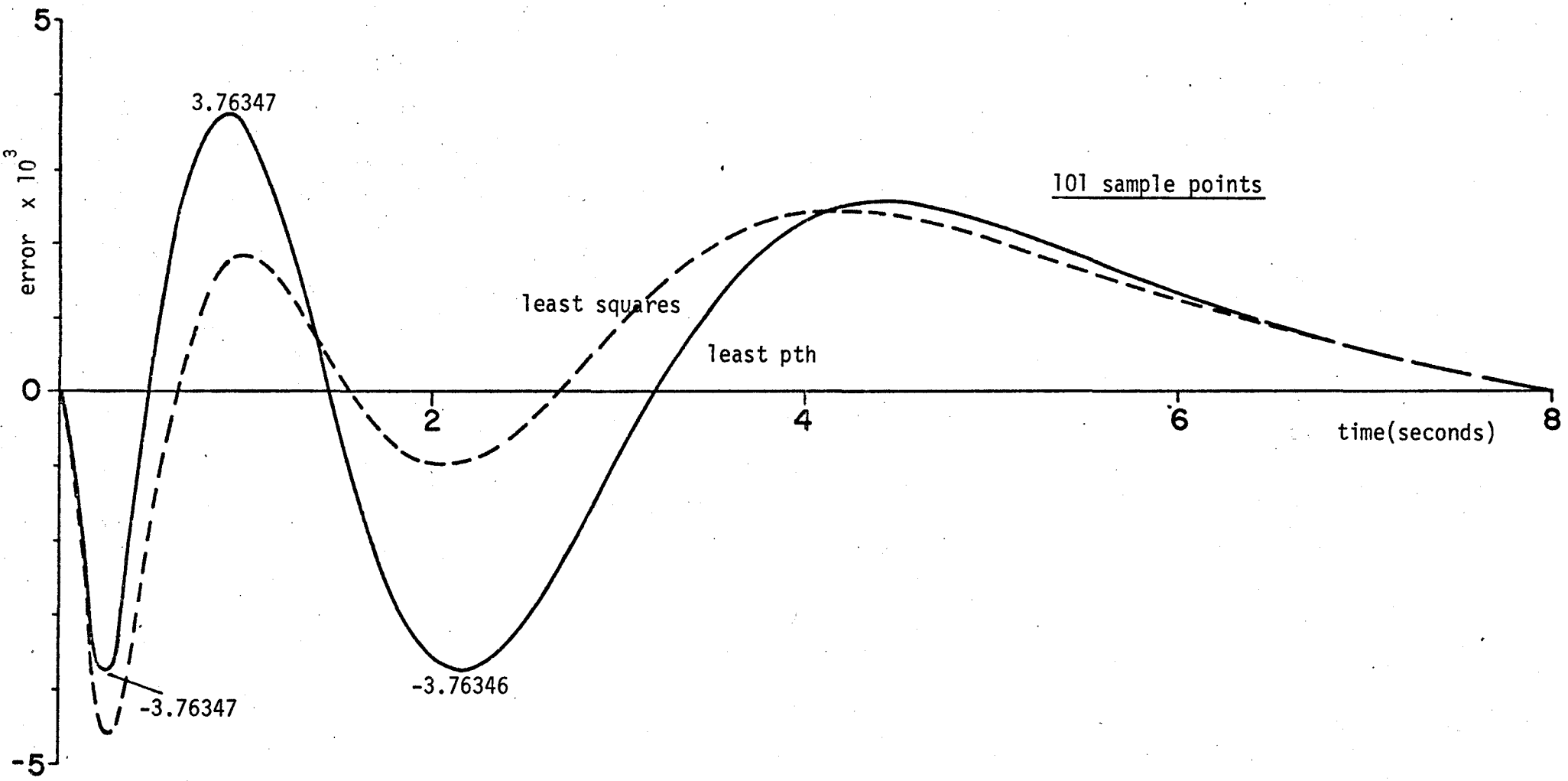

Fig. 9c. Error curve for two-parameter least pth problem (p=1000000) compared to curve for least squares problem.

$$-3.7651 \times 10^{-3}$$
$$3.7594 \times 10^{-3}$$
$$-3.7569 \times 10^{-3}$$
$$2.5518 \times 10^{-3}$$

which show agreement to almost three significant figures amongst three of the extrema. For the same problem and with $p = 10^6$, the values of the extrema were

$$-3.7635 \times 10^{-3}$$
$$3.7635 \times 10^{-3}$$
$$-3.7635 \times 10^{-3}$$
$$2.5524 \times 10^{-3}$$

which show agreement to five significant figures amongst three of the extrema. Figure 9c shows the corresponding error curve.

For the three-parameter problem represented by equation (33) the optimum parameters were

$$a_0 = 3.83592$$
$$a_1 = 3.00605$$
$$b_1 = -1.77277 \times 10^{-2}$$

giving an objective function value of 2.488186×10^{-3} Figure 10a shows the corresponding response and Figure 10b shows the error curve corresponding that response. The error curve is shown to have five extrema, three of

STARTING POINT	MINIMIZATION OF $F(\underset{\sim}{x})$			MINIMIZATION OF $M(\underset{\sim}{x})$
	FLETCHER	JACOBSON-OKSMAN	FLETCHER-POWELL	GRAZOR
2.5 2.0 -2.0	367	339	630†	149
1.0 1.0 -1.0	378	137	653†	368
4.0 3.0 0.01	247	260	264	165
3.5 1.5 1.0	290	failed	432	358
5.0 1.0 -1.0	197	failed	624†	325
5.0 1.0 3.0	247	failed	638†	406

TABLE 6. Number of function evaluations required to reach the objective function value 2.488187×10^{-3} for the 3-parameter problem.

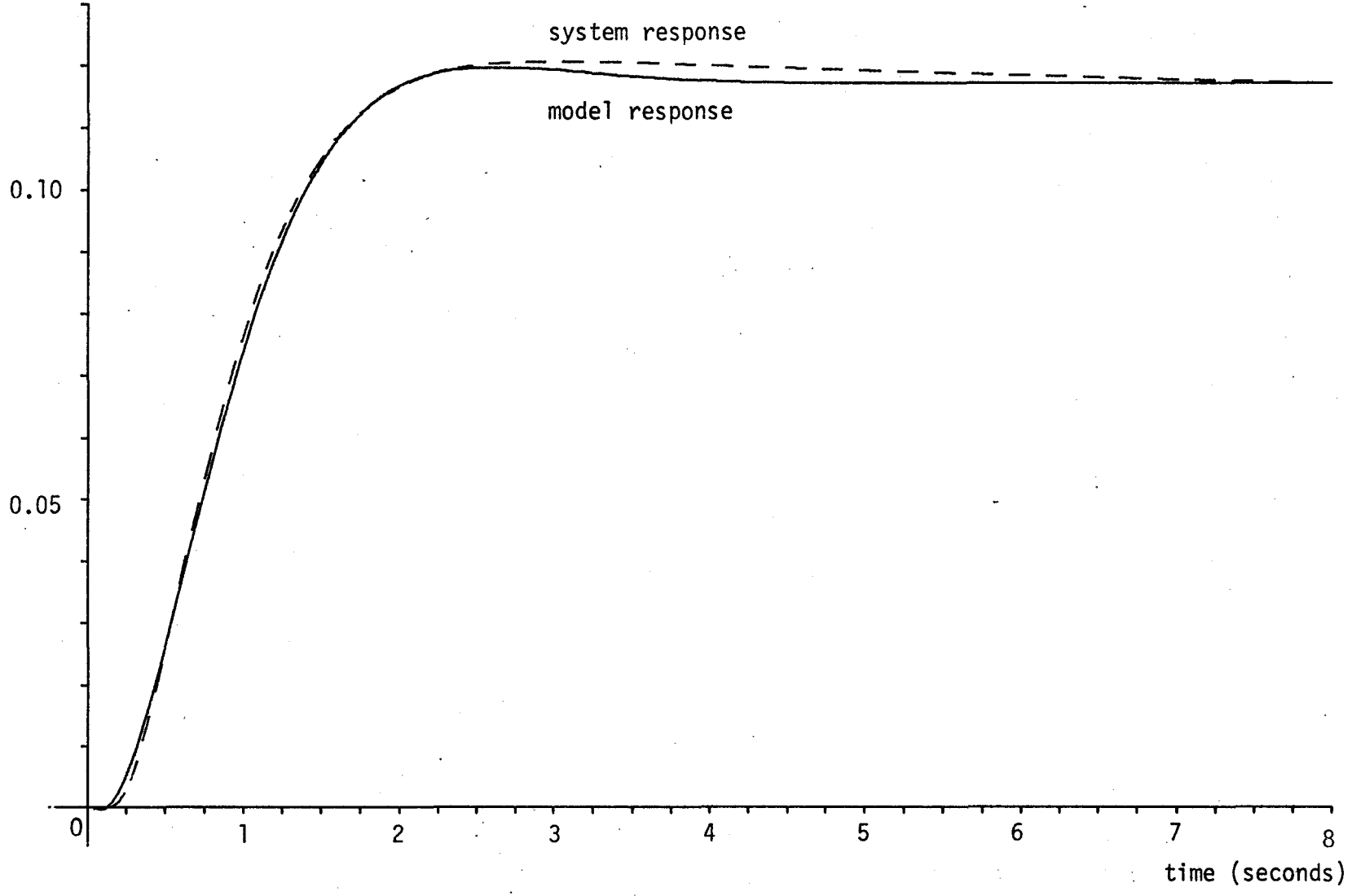

Fig. 10a. Response for three-parameter least pth problem (p = 1000)

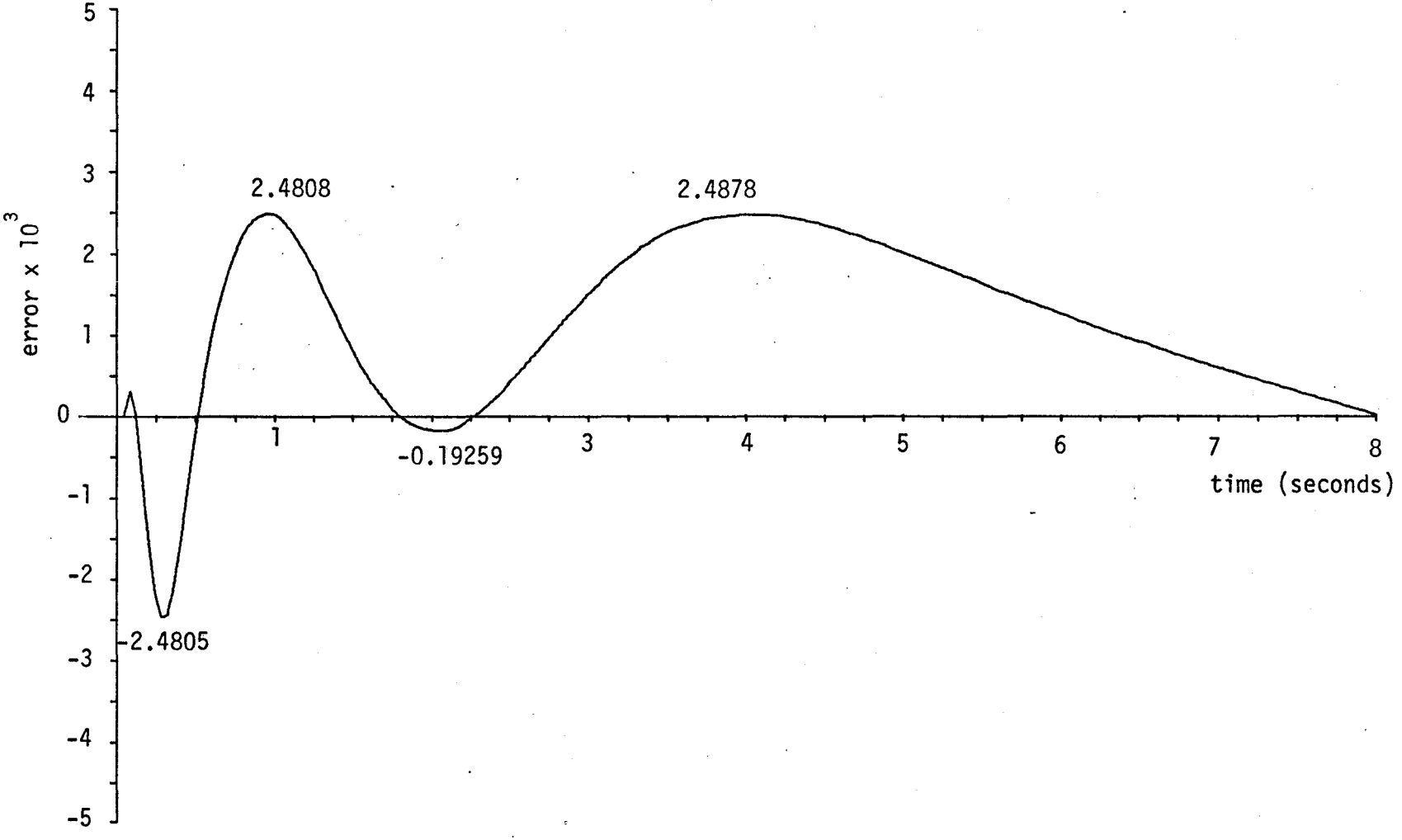

Fig. 10b. Error curve for three-parameter least pth problem (p = 1000)

which tend to be equal with values

$$-2.4805 \times 10^{-3}$$
$$2.4808 \times 10^{-3}$$
$$2.4879 \times 10^{-3}$$

Table 6 shows the number of function evaluations required to reach the objective function value 2.488187×10^{-3}. The Jacobson-Oksman method failed in 50% of the cases tried, while the Fletcher-Powell method only twice reached that value of the objective function in the time available. It should be noted however that the Fletcher-Powell method proceeded towards the optimum and did not diverge. The Fletcher method reached the optimum in all cases and on the basis of reliability and function evaluations appeared to be the most efficient. Table 5 also shows the number of function evaluations required by the grazor search technique to reach the corresponding value of maximum error.

For the five-parameter problem, represented by equation (34), the optimum parameters obtained using the Fletcher method were

$$x_1 = 4.34682$$
$$x_2 = 3.36738$$
$$x_3 = 9.96086 \times 10^{-2}$$
$$x_4 = 5.14728 \times 10^{-1}$$
$$x_5 = 3.56154 \times 10^{-2}$$

giving an objective function value of 1.02134×10^{-3}. Figures 11a and 11b show the corresponding response and error curves, respectively.

STARTING POINT	MINIMIZATION OF $F(\underset{\sim}{x})$		MINIMIZATION OF $M(\underset{\sim}{x})$	
	FLETCHER		GRAZOR	
	N	M x 10^3	N	M x 10^3
3.0 3.0 1.5 0.5 -0.1	530	1.0207	437	1.2139
1.5 3.0 2.5 1.0 0.1	768	1.0207	782	1.2473
4.0 3.0 0.1 0.5 -0.03	177	1.0207	489	1.0206
3.0 5.0 0.2 0.3 -0.1	862	1.0207	634	1.1720
5.0 4.0 0.5 1.0 -0.5	484	1.0207	817	1.0337
LEAST SQUARES OPTIMUM	799	1.0206	537	1.2472

TABLE 7. Number of function evaluations required for the Fletcher method to reach the objective function value corresponding to the shown maximum error.

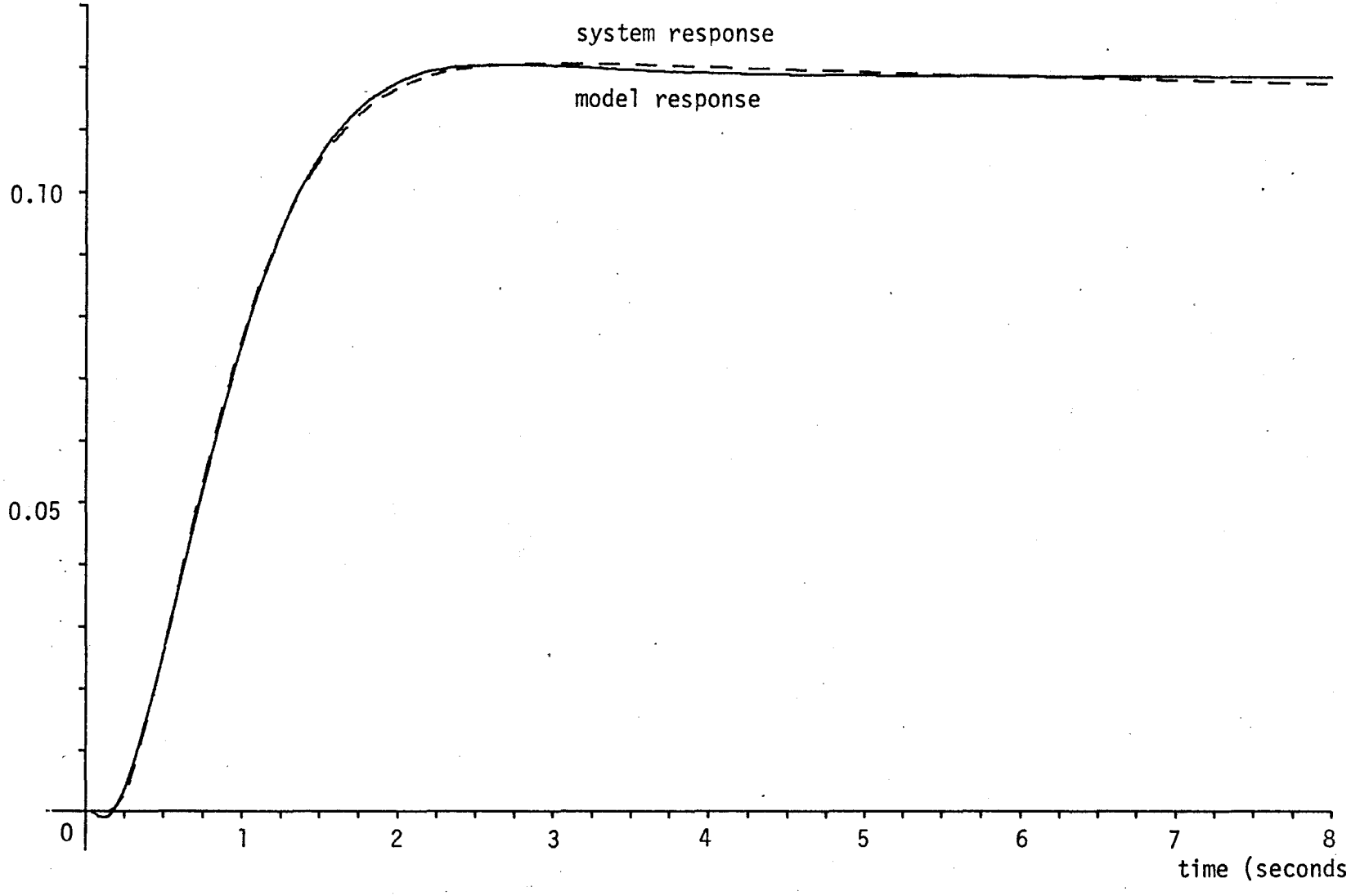

Fig. 11a. Response for five-parameter least pth problem (p = 1000)

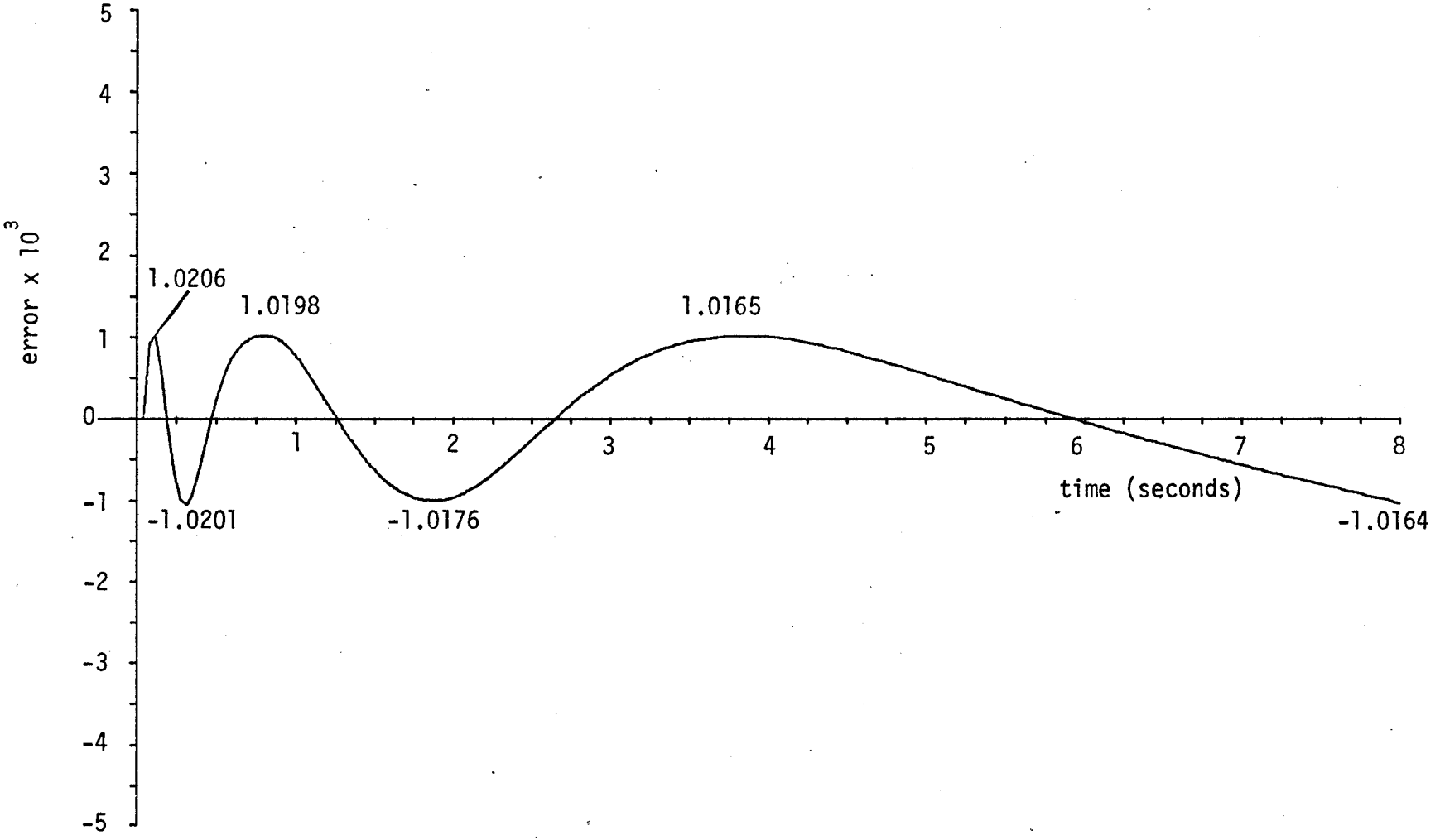

Fig. 11b. Error curve for five-parameter least pth problem (p = 1000)

For this five-parameter problem the error curve has six extrema the values of which are

$$1.0206 \times 10^{-3}$$
$$-1.0201 \times 10^{-3}$$
$$1.0198 \times 10^{-3}$$
$$-1.0176 \times 10^{-3}$$
$$1.0165 \times 10^{-3}$$
$$-1.0164 \times 10^{-3}$$

Some runs with the Fletcher-Powell method indicated that the method was slow and since this was already established in the previous problems, further runs of the Fletcher-Powell method were considered unnecessary. The Jacobson-Oksman method failed from each starting point. The Fletcher method converged to a unique solution all the times, with the above given optimum parameters. Table 7 compares the least pth approximation using the Fletcher method with the grazor search technique. While the Fletcher method reached the same optimum at all times the grazor search technique reached this solution only once. The grazor search technique gave a five equiripple answer. To further investigate the reason for which the grazor method gave a five-extrema optimum, the Fletcher method was started from this optimum. It was noticed that although the least pth objective function decreased, there was a temporary increase in the maximum error value. This brought us to the conjecture that the least pth objective function formulation seems to have the attractive property of overcoming local optima that would appear in the maximum error type of objective function formulation.

For each of the minimax results obtained the conditions for a minimax optimum [21] were satisfied, although as can be seen from the error curves all the extrema were not equal.

CHAPTER 6

CONCLUSIONS

Optimum second-order and third-order models of a high-order system have been obtained using least squares and least pth objective functions in conjunction with efficient gradient minimization techniques. Analytical expressions for the gradients, were not difficult to derive. Since the expressions of the objective function and gradients are dependent only on the parameters of the model used, and they are independent of the high-order system, the expressions can be stored and used whenever any high-order system is to be modelled.

In general the use of gradient techniques have been found efficient in deriving models. In cases of dependence between some of the parameters a unique solution can not be found, however, for most practical purposes any of the solutions obtained might be acceptable.

The contours of the problem with linear dependence between two parameters, shown in Figure 2a, if plotted over a large range are found to be roughly ellipsoidal. The question, therefore, is: can an easier solution be obtained by making the contours more spherical? This can be achieved for ellipsoids by rotation of the appropriate coordinates and proper scaling. An interpretation of this dependence can be given in

terms of the model response. If an infinite number of sample points are taken, then the contours shown in Figure 2a should be parallel, and in three dimensions the contours should have the form of concentric cylinders. This is because the weight associated with the transient portion of the response is negligible. Now as the interval of approximation is reduced the contours should become more spherical. A point will be reached, however, when the model becomes unstable or the steady-state error becomes unacceptably large.

An attempt has not been made to answer the question of whether the optima obtained are the global optima. If the algorithm converged to a unique point, starting from n arbitrary starting points, where n is the dimension of the parameter vector, then this point was considered an optimum. In addition, in the case of the least pth approximation, the conditions for a minimax optimum had to be satisfied, for the point to be considered an otpimum.

Constraints relating to the steady-state value can be applied by substitution, as shown in chapter 5. In general any such substitution which reduces the dimensionality of the problem is desirable since it increases the rate of convergence. In addition, constraints relating to the energy of the model can also be imposed in the form of equality constraints, but this would require a slightly different formulation of the objective function and of course the gradients. Constraints relating to the stability of the model, such as forcing all the poles to lie

in the left-half of the s-plane, can easily be imposed if it is found necessary.

The normalised least pth approximation is shown to be a simple and effective approach to the minimax problem. The value of p in general would depend on the problem, the number of significant figures the extrema should be equal to and the computer size and time available. For the problems described and for central processing times of less than 64 seconds, p = 1000 was found suitable. This value of p appeared to have the tendency of overcoming any local minima that might appear in a more direct minimax approach or when using $p \geq 10^6$.

The number of sample points was not considered critical in the least squares cases, although a sufficient number should be taken so that the objective function is a reasonable representation of the error. In the least pth approximation accurate location of the extrema was considered important, therefore a large enough number of sample points should be taken to include points close to the actual extrema. A linear search could be made between the sample points, in which case a smaller number of points would be required initially; however, it is doubtful if this will give an increase in the efficiency of the method.

From the minimization techniques used, Fletcher-Powell was found to be reliable in the sense that the algorithm never diverged. This was to be expected since the method has been widely used since it

was originally proposed. The method, however, was found to be slow by comparison to the method proposed by Fletcher and the one by Jacobson and Oksman.

The method proposed by Fletcher was found the most efficient of the methods used. That is, in most of the cases, it required the least number of function evaluations to reach the optimum. Since computing effort is, for these type of problems, measured in terms of function evaluations, the method required the least amount of computing. The choice of values of some of the constants in the algorithm, such as μ and w, do not seem to have any real justification. It might, therefore, be pertinent to ask if a different choice of these constants would result in a better performance for the method. This would depend on the objective function under consideration and any changes made, would have been to suit our particular objective function. That means adapting algorithm to our particular problem, rather than use the problem as a test case to test the efficiency of the algorithm.

The Jacobson-Oksman method by considering homogeneous functions covers a larger class of problems, and on general functions seems to have a fast rate of convergence. This has been shown in the cases where the algorithm converged, as shown in the two-parameter problems. The method was found to be unstable as the number of parameters increased. That instability was due to step sizes becoming very large and consequently the objective function value exceeded the range of the computer. In an effort to control the step sizes, a limit was set on it. However this

reduced the rate of convergence to a rate slower than the Fletcher-Powell method. It was decided, therefore, to leave the algorithm in the form in which it was originally proposed. It was also found that the Jacobson-Oksman method had the tendency to restart several times during a run, by setting the inverse matrix to unity. This means losing all the information that has been obtained in the previous iterations. Therefore, an alternative way of updating the inverse matrix might improve efficiency and perhaps contribute to the stability of the algorithm.

To conclude, the method proposed by Fletcher was found efficient in deriving optimum low-order models and can be considered as a significant improvement over other gradient methods such as Fletcher-Powell.

APPENDICES

APPENDIX A

1. The Two-Parameter Problem

$$M_1(s) = \frac{C(s)}{R(s)} = \frac{Ea_0}{s^2 + a_1 s + a_0} \tag{A-1}$$

For step input, $R(s) = \frac{1}{s}$ (A-2)

Hence,

$$C(s) = \frac{Ea_0}{s(s^2 + a_1 s + a_0)} = \frac{E}{s} - \frac{E(s + a_1)}{s^2 + a_1 s + a_0} \tag{A-3}$$

$$\frac{\partial C(s)}{\partial a_0} = \frac{E(s + a_1)}{(s^2 + a_1 s + a_0)^2} \tag{A-4}$$

$$\frac{\partial C(s)}{\partial a_1} = \frac{Ea_0}{(s^2 + a_1 s + a_0)^2} \tag{A-5}$$

The sensitivity functions $\frac{\partial c(t)}{\partial a_0}$ can now be evaluated by taking the inverse Laplace transforms of equations (A-4) and (A-5). These can be easily obtained from standard tables. For example, if the poles of the transform function are complex i.e.,

$$s^2 + a_1 s + a_0 = (s + a)^2 + \beta^2 \tag{A-6}$$

then from equations (A-4) and (A-5) we have

$$\frac{\partial c(t)}{\partial a_0} = \frac{Ee^{-\alpha t}}{2\beta^3}\left[(a + \beta^2 t)\sin\beta t - \alpha\beta t\cos\beta t\right] \tag{A-7}$$

$$\frac{\partial c(t)}{\partial a_1} = \frac{Ea_0\, e^{-\alpha t}}{2\beta^3}(\sin\beta t - \beta t\cos\beta t) \tag{A-8}$$

2. The Three-Parameter Problem

$$M_1(s) = \frac{b_1 s + Ea_0}{s^2 + a_1 s + a_0} \tag{A-9}$$

$$C(s) = \frac{b_1 s + Ea_0}{s(s^2 + a_1 s + a_0)} = \frac{E}{s} - \frac{E(s + a_1) - b_1}{s^2 + a_1 s + a_0} \tag{A-10}$$

$$\frac{\partial C(s)}{\partial a_0} = \frac{E(s + a_1) - b_1}{(s^2 + a_1 s + a_0)^2} \tag{A-11}$$

$$\frac{\partial C(s)}{\partial a_1} = -\frac{b_1 s + Ea_0}{(s^2 + a_1 s + a_0)^2} \tag{A-12}$$

$$\frac{\partial C(s)}{\partial b_1} = \frac{1}{s^2 + a_1 s + a_0} \tag{A-13}$$

The sensitivities in the time domain are again obtained by taking inverse Laplace transforms.

3. The Five-Parameter Problem

$$M_5(s) = \frac{x_5 s^2 + x_4 s + E_1 x \quad x_3}{(s + x\ )\ (s^2 + x_2 s + x_1)} \tag{A-14}$$

$$C(s) = \frac{x_5 s^2 + x_4 s + E\ x_1\ x_3}{s(s + x_3)\ (s^2 + x_2 s + x_1)}$$

$$= \frac{E}{s} - \frac{E[s^2 + (x_2 + x_3)s + x_1 + x_2\ x_3)] - x_5 s - x_4}{(s + x_3)\ (s^2 + x_2 s + x_1)} \tag{A-15}$$

$$\frac{\partial C(s)}{\partial x_1} = - \frac{x_5 s + x_4 - E x_3 (s + x_2)}{(s + x_3)\ (s^2 + x_2 s + x_1)^2} \tag{A-16}$$

$$\frac{\partial C(s)}{\partial x_2} = - \frac{x_5 s^2 + x_4 s + E\ x_1\ x_3}{(s + x_3)\ (s^2 + x_2 s + x_1)^2} \tag{A-17}$$

$$\frac{\partial C(s)}{\partial x_3} = - \frac{x_5 s + x_4 - E x_1}{(s + x_3)^2\ (s^2 + x_2 s + x_1)} \tag{A-18}$$

$$\frac{\partial C(s)}{\partial x_4} = \frac{1}{(s + x_3)\ (s^2 + x_2 s + x_1)} \tag{A-19}$$

$$\frac{\partial C(s)}{\partial x_5} = \frac{s}{(s + x_3)\ (s^2 + x_2 s + x_1)} \tag{A-20}$$

APPENDIX B

TWO-PARAMETER PROBLEM

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      LOGICAL CONV,UNITH
      DIMENSION C7(501)
      DIMENSION X(3), G(3), H(15), EPS(3)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
      COMMON /COM3/ KNT
      READ (5,2) N,MAXFN
      READ (5,3) (X(I),I=1,N)
      READ (5,4) (C7(I),I=1,501)
      DO 1 I=1,N
      EPS(I)=1.0E-9
1     CONTINUE
      FEST=0.0
      KNT=0
      FO=100.
      UNITH=.TRUE.
      IPRINT=1
      CALL VMM01 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT)
      WRITE (6,5) IEXIT
      STOP
C
2     FORMAT (2I5)
3     FORMAT (8F10.5)
4     FORMAT (10F8.5)
5     FORMAT (2X,* INFORMATION OF CONVERGENCE *,I4)
      END
      SUBROUTINE FUNCT (N,X,F,G)
      INTEGER P
      DIMENSION ER(201), AER(201), INST(201)
      DIMENSION G(5), GR(5)
      DIMENSION X(5), C2(501), C7(501)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
      COMMON /COM3/ KNT
C     SECOND ORDER MODEL  WITH SS CONSTRAINT AND NO ZERO
      P=1000
      KNT=KNT+1
      SS=0.11706
      WRITE (6,9) (X(I),I=1,2)
      F=0.
      G(1)=0.
      G(2)=0.
      Y=X(2)*X(2)/4.-X(1)
      IF (Y.GT.0.) GO TO 3
      YN=X(1)-X(2)*X(2)/4.
      W=SQRT(YN)
      Z=X(2)/2.
      DO 1 I=1,201,2
      T=0.04*FLOAT(I-1)
      C2(I)=SS*(1.-(COS(W*T)+Z*SIN(W*T)/W)/EXP(Z*T))
C
      ER(I)=C7(I)-C2(I)
      AER(I)=ABS(ER(I))
1     CONTINUE
```

```
      M=201
      CALL PEAKS (M,AER,INST,NPKS,K)
      F=AER(K)
      DO 2 I=1,NPKS
      L=INST(I)
      T=0.04*FLOAT(L-1)
      SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
      GR(1)=SS*(T*SIN(W*T)-(Z/(W*W))*(T*W*COS(W*T)-SIN(W*T)))/(EXP(Z*T)*
     12.*W)
      GR(2)=-SS*X(1)*(SIN(W*T)-W*T*COS(W*T))/(EXP(Z*T)*2.*W*W*W)
      G(1)=G(1)-SIGMA*GR(1)
      G(2)=G(2)-SIGMA*GR(2)
2     CONTINUE
C
      GO TO 6
3     A1=-X(2)/2.+SQRT(Y)
      A2=-X(2)/2.-SQRT(Y)
      B=2.*SQRT(Y)
      Q1=SS
      Q2=SS*X(1)/(A1*B)
      Q3=-X(1)/(A2*B)*SS
      V1=-X(1)/(X(1)*X(2))*((A1*A1+A2*A2)/(B*B)-1.)*SS
      V2=(SS*(1.-A1*A1/(B*B))-V1*A1)*X(1)
      V3=-V1
      V4=(X(1)*SS-V2*A2*A2)/(A1*A1)
      DO 4 I=1,201,2
      T=0.04*FLOAT(I-1)
      C2(I)=Q1+Q2*EXP(A1*T)+Q3*EXP(A2*T)
      ER(I)=C7(I)-C2(I)
      AER(I)=ABS(ER(I))
4     CONTINUE
      M=201
      CALL PEAKS (M,AER,INST,NPKS,K)
      F=AER(K)
      DO 5 I=1,NPKS
      L=INST(I)
      T=0.04*FLOAT(L-1)
      SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
      GR(1)=(((X(2)+A1)*T-(A1+A2+2.*X(2))/B)*EXP(A1*T)+((X(2)+A2)*T+(A1+
     1A2+2.*X(2))/B)*EXP(A2*T))/(B*B)
      GR(2)=V1*((V2/V1+A1)*T+1.)*EXP(A1*T)+V3*((V4/V3+A2)*T+1.)*EXP(A2*T
     1)
      GR(2)=-GR(2)
      G(2)=G(2)-SIGMA*GR(2)
      G(1)=G(1)-SIGMA*GR(1)
5     CONTINUE
6     CONTINUE
      AUX1=0.0
      DO 7 I=1,NPKS
      L=INST(I)
      AUX1=AUX1+(AER(L)/F)**P
7     CONTINUE
      RP=1.0/FLOAT(P)
      F=F*(AUX1**RP)
      IF (F.GE.FO) RETURN
      BO=X(1)*SS
      WRITE (6,10) X(1),G(1),F,X(2),G(2),BO
```

```
      FO=F
      IF (F.LT.0.0037662) GO TO 8
      RETURN
8     WRITE (6,11)
      CALL SECOND (T1)
      WRITE (6,12) T1
      RETURN
C
9     FORMAT (86X,3F10.5)
10    FORMAT (2X,6E15.6)
11    FORMAT (3X,*VALUE REACHED*)
12    FORMAT (///20X,*TIME TAKEN*,F10.5)
      END
      SUBROUTINE PEAKS (M,AER,INST,NPKS,K)
      DIMENSION AER(1), INST(1)
      ML=M-2
      J=0
      AER(1)=0.0
      AER(M)=0.0
      DO 1 I=3,ML,2
      IF (AER(I-2).GT.AER(I).OR.AER(I+2).GT.AER(I)) GO TO 1
      J=J+1
      INST(J)=I
      NPKS=J
1     CONTINUE
C
C     TO FIND THE MAX ERROR
      BIG=AER(1)
      DO 2 I=1,NPKS
      L=INST(I)
      IF (AER(L).LE.BIG) GO TO 2
      BIG=AER(L)
      K=L
2     CONTINUE
      RETURN
      END
```

CD TOT 0152

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      LOGICAL CONV,UNITH
      DIMENSION C7(501)
      DIMENSION X(3), G(3), H(15), EPS(3)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
      COMMON /COM3/ KNT
      READ (5,4) N,MAXFN
      READ (5,5) (X(I),I=1,N)
      READ (5,6) (C7(I),I=1,501)
      DO 1 I=1,N
      EPS(I)=1.0E-9
1     CONTINUE
      FEST=0.0
      FO=100.
      KNT=0
      UNITH=.TRUE.
      IPRINT=1
2     CONTINUE
      CALL VMM01 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT)
      WRITE (6,7) IEXIT
      IF (IEXIT.NE.4) STOP
      DO 3 I=1,N
      X(I)=X(I)+0.1
3     CONTINUE
      WRITE (6,8)
      GO TO 2
4     FORMAT (2I5)
5     FORMAT (8F10.5)
6     FORMAT (10F8.5)
7     FORMAT (2X,* INFORMATION OF CONVERGENCE *,I4)
8     FORMAT (3X,*A RESTART HAS OCCURED*)
      END
      SUBROUTINE FUNCT (N,X,F,G)
      INTEGER P
      DIMENSION ER(201), AER(201), INST(201)
      DIMENSION X(1), G(1)
      DIMENSION GR(3), C2(501), C7(501)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
      COMMON /COM3/ KNT
      P=1000
      KNT=KNT+1
      F=0.
      G(1)=0.
      G(2)=0.
      G(3)=0.
      SS=0.11706
      E=0.11706
      B=X(2)-X(3)/E
      C=E*X(1)/X(3)
      Y=X(1)-X(2)*X(2)/4.
      IF (Y.LT.0.) GO TO 3
      W=SQRT(Y)
```

```
      Z=X(2)*0.5
      DO 1 I=1,201,2
      T=0.04*FLOAT(I-1)
      C2(I)=X(3)*SIN(W*T)/(W*EXP(Z*T))+(1.-(Z*SIN(W*T)/W+COS(W*T))/EXP(
     1*T))*E
      ER(I)=C7(I)-C2(I)
      AER(I)=ABS(ER(I))
1     CONTINUE
      M=201
      CALL PEAKS (M,AER,INST,NPKS,K)
      F=AER(K)
      DO 2 I=1,NPKS
      L=INST(I)
      T=0.04*FLOAT(L-1)
      SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
      GR(1)=E*((B-Z+W*W*T)*SIN(W*T)+(Z-B)*W*T*COS(W*T))/(EXP(Z*T)*2.*W*
     1*W)
      GR(2)=-X(3)*((C-Z+W*W*T)*SIN(W*T)+(Z-C)*W*T*COS(W*T))/(2.*W*W*W*E
     1P(Z*T))
      GR(3)=SIN(W*T)/(EXP(Z*T)*W)
      G(1)=G(1)-SIGMA*GR(1)
      G(2)=G(2)-SIGMA*GR(2)
      G(3)=G(3)-SIGMA*GR(3)
2     CONTINUE
      GO TO 6
3     CONTINUE
      YN=-Y
      W=SQRT(YN)
      ZN=-X(2)*0.5
      A1=ZN+W
      A2=ZN-W
      DO 4 I=1,201,2
      T=0.04*FLOAT(I-1)
      C2(I)=X(3)*(C/(A1*A2)+(A1+C)*EXP(A1*T)/((A1-A2)*A1)+(A2+C)*EXP(A2
     1T)/((A2-A1)*A2))
      ER(I)=C7(I)-C2(I)
      AER(I)=ABS(ER(I))
4     CONTINUE
      M=201
      CALL PEAKS (M,AER,INST,NPKS,K)
      F=AER(K)
      DO 5 I=1,NPKS
      L=INST(I)
      T=0.04*FLOAT(L-1)
      SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
      GR(1)=E*(((B+A2)*T-(A1+A2+2.*B)/(A2-A1))*EXP(A2*T)+((B+A1)*T-(A1+
     12+2.*B)/(A1-A2))*EXP(A1*T))/((A1-A2)*(A1-A2))
      GR(2)=-X(3)*(((C+A2)*T-(A1+A2+2.*C)/(A2-A1))*EXP(A2*T)/((A2-A1)*(
     12-A1))+((C+A1)*T-(A1+A2+2.*C)/(A1-A2))*EXP(A1*T)/((A1-A2)*(A1-A2)
     2)
      GR(3)=(EXP(A1*T)-EXP(A2*T))/(A1-A2)
      G(1)=G(1)-SIGMA*GR(1)
      G(2)=G(2)-SIGMA*GR(2)
      G(3)=G(3)-SIGMA*GR(3)
5     CONTINUE
6     CONTINUE
      AUX1=0.0
```

```
      DO 7 I=1,NPKS
      L=INST(I)
      AUX1=AUX1+(AER(L)/F)**P
7     CONTINUE
      RP=1.0/FLOAT(P)
      F=F*(AUX1**RP)
      SCALAR=(1.0/AUX1)**(1.0-RP)
      G(1)=G(1)*SCALAR
      G(2)=G(2)*SCALAR
      G(3)=G(3)*SCALAR
      IF (F.GE.FO) RETURN
      FO=F
      IF (F.LT.2.488187E-3) STOP
      IF (F.LT.2.488187E-3) GO TO 8
      RETURN
8     CONTINUE
      WRITE (6,9)
      WRITE (6,10) X(1),G(1),F,X(2),G(2),X(3),G(3)
      RETURN
9     FORMAT (//* REQUIRED VALUE REACHED *)
10    FORMAT (/1X,*A0=*,E14.6,15X,*GRADIENT=*,E14.6,10X,*OBJ. FUNCTION=
     1,E14.6,/1X,*A1=*,E14.6,24X,E14.6,/1X,*B1=*,E14.6,24X,E14.6)
      END
      SUBROUTINE PEAKS (M,AER,INST,NPKS,K)
      DIMENSION AER(1), INST(1)
      ML=M-2
      J=0
      AER(1)=0.0
      AER(M)=0.0
      DO 1 I=3,ML,2
      IF (AER(I-2).GT.AER(I).OR.AER(I+2).GT.AER(I)) GO TO 1
      J=J+1
      INST(J)=I
      NPKS=J
1     CONTINUE
C
C     TO FIND THE MAX ERROR
      BIG=AER(1)
      DO 2 I=1,NPKS
      L=INST(I)
      IF (AER(L).LE.BIG) GO TO 2
      BIG=AER(L)
      K=L
2     CONTINUE
      RETURN
      END
```

CD TOT 0162

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      LOGICAL CONV,UNITH
      DIMENSION C7(501)
      DIMENSION X(6), G(6), H(39), EPS(6)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
      COMMON /COM3/ KNT
      COMMON /COM12/ P
      P=1000
C     $$$$$$$$$$$$  CONSTRAINT LEAST PTH NORM  $$$$$$$$
      READ (5,7) N,MAXFN
      READ (5,8) (X(I),I=1,N)
      READ (5,9) (C7(I),I=1,501)
      DO 1 I=1,N
      EPS(I)=1.0E-6
1     CONTINUE
2     IK=0
      FEST=0.0
      FO=100.
      KNT=0
      UNITH=.TRUE.
      IPRINT=10
      IPRINT=1
C     CALL  CHECGR(N,X)
3     CONTINUE
      CALL VMMO1 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT)
      WRITE (6,10) IEXIT
      IF (IEXIT.NE.4) GO TO 4
      WRITE (6,11)
      GO TO 3
4     IK=IK+1
      IF (F.LT.1.0E-03) GO TO 5
      IF (IK.LE.N) GO TO 3
5     KNT=-5
C     CALL FUNCT(N,X,F,G)
      IF (P.GT.1000000000000) GO TO 6
      P=P*P
      GO TO 2
6     CONTINUE
      STOP
C
7     FORMAT (2I5)
8     FORMAT (5F10.5)
9     FORMAT (10F8.5)
10    FORMAT (2X,* INFORMATION OF CONVERGENCE *,I4)
11    FORMAT (//3X,*A RESTART HAS OCCURED*)
      END
      SUBROUTINE FUNCT (N,XY,F,G)
      INTEGER P
      DIMENSION ER(201), AER(201), INST(201)
      DIMENSION C7(1)
      DIMENSION GR(6), C3(201)
      DIMENSION X(6), G(6), XY(6)
      COMMON /COM1/ C7
      COMMON /COM2/ FO
```

```
      COMMON /COM3/ KNT
      COMMON /COM8/ INST
      COMMON /COM7/ NPKS
      COMMON /COM6/ ER
      COMMON /COM12/ P
      KNT=KNT+1
      F=0.
      E=0.11706
      X(1)=XY(1)
      X(2)=XY(2)
      X(3)=XY(3)
      X(4)=E*XY(1)*XY(3)
      X(5)=XY(4)
      X(6)=XY(5)
      DO 1 I=1,N
      G(I)=0.0
1     CONTINUE
      Y=X(1)-X(2)*X(2)*0.25
      A1=X(5)/X(6)
      A0=X(4)/X(6)
      CA=-X(4)/(X(1)*X(1)*X(3))
      AUX1=X(3)*X(3)-X(2)*X(3)+X(1)
      CB=(X(4)-X(3)*X(5)+X(3)*X(3)*X(6))/(X(3)*AUX1*AUX1)
      CC=-CA-CB
      CD=CC*2.*X(2)-CA*X(3)-CC*X(3)
      CE=CC*(2.*X(1)+X(2)*X(2))-2.*CA*X(2)*X(3)-CD*X(3)
      CF=CC*2.*X(2)*X(1)-X(6)-(2.*X(1)+X(2)*X(2))*CA*X(3)-CE*X(3)
      AX0=CF/CC
      AX1=CE/CC
      AX2=CD/CC
      DB=(X(5)*X(3)-X(4)-X(6)*X(3)*X(3))/(AUX1*AUX1)
      DC=-DB
      DD=DC*(2.*X(2)-X(3))
      DE=-DD*X(3)-DB*(2.*X(1)+X(2)*X(2))-X(6)
      DF=-2.*X(2)*X(1)*DB-DE*X(3)-X(5)
      AY0=DF/DC
      AY1=DE/DC
      AY2=DD/DC
      EA=-X(4)/(X(1)*X(3)*X(3))
      EB1=X(6)*X(3)*X(3)-2.*X(5)*X(3)-X(1)*X(6)+X(2)*X(5)+3.*X(4)
      EB2=(X(1)*X(4)-2.*X(2)*X(4)*X(3))/(X(3)*X(3))
      EB=(EB1+EB2)/(AUX1*AUX1)
      EC=(X(3)*(X(3)*X(6)-X(5))+X(4))/(X(3)*AUX1)
      ED=-EB-EA
      EE=-(X(5)+EC*X(1)+(EB+2.*EA)*X(1)*X(3))/(X(3)*X(3))-EA*X(2)
      IF (Y.LT.0.) GO TO 5
      Z=X(2)*0.5
      W=SQRT(Y)
      GM1=Z-X(3)
      GM2=W*(2.*Z-A1)
      GM3=(Z*Z-W*W+A0-A1*Z)
      PH1=ATAN2(W,Z)
      PH2=ATAN2(W,GM1)
      PH3=ATAN2(GM2,GM3)
      PHX=PH1+PH2-PH3
C
      Y1=A0/(X(1)*X(3))
```

```
      Y2=(A1*X(3)-A0-X(3)*X(3))/(X(3)*((Z-X(3))*(Z-X(3))+W*W))
      RT1=((Y*(X(2)-A1)*(X(2)-A1)+(Z*Z-Y-A1*Z+A0)*(Z*Z-Y-A1*Z+A0))/(X(1
     1*((Z-X(3))*(Z-X(3))+Y)))
      RT1=SQRT(RT1)
      DO 2 I=1,201,2
      T=0.04*FLOAT(I-1)
      C3(I)=X(6)*(Y1+Y2/EXP(X(3)*T)+RT1*SIN(W*T+PHX)/(W*EXP(Z*T)))
      ER(I)=C7(I)-C3(I)
      AER(I)=ABS(ER(I))
2     CONTINUE
      M=201
      CALL PEAKS (M,AER,INST,NPKS,K)
      F=AER(K)
      Y10=((AX2-Z)*X(1)-Z*(AX1+2.*W*W)+AX0)/(2.*W*W*W)
      Y11=(3.*Z*Z-W*W-2.*AX2*Z+AX1)/(2.*W)
      Y12=(AX0-AX1*Z+AX2*(Z*Z-W*W)+Z*(3.*W*W-Z*Z))/(2.*W*W)
      Y13=((AY2-Z)*X(1)-Z*(AY1+2.*W*W)+AY0)/(2.*W*W*W)
      Y14=(3.*Z*Z-W*W-2.*AY2*Z+AY1)/(2.*W)
      Y15=(AY0-AY1*Z+AY2*(Z*Z-W*W)+Z*(3.*W*W-Z*Z))/(2.*W*W)
      RTZ=SQRT(1.+(EE/ED-Z)*(EE/ED-Z)/(W*W))
      GM6=EE/ED-Z
      PH6=ATAN2(W,GM6)
      Y3=1./(X(3)*X(1))
      Y4=1./(X(3)*((Z-X(3))*(Z-X(3))+W*W))
      Y5=W*SQRT(X(1)*((Z-X(3))*(Z-X(3))+W*W))
      PHY=PH1+PH2
      PH4=ATAN2(W,-GM1)
      Y6=1./(W*W+(X(3)-Z)*(X(3)-Z))
      Y7=W*SQRT((X(3)-Z)*(X(3)-Z)+W*W)
      PH5=ATAN2(W,-Z)
      PHZ=PH5-PH4
      Y8=-X(3)/((X(3)-Z)*(X(3)-Z)+W*W)
      Y9=SQRT(X(1)/(W*W+(X(3)-Z)*(X(3)-Z)))/W
      DO 4 I=1,NPKS
      L=INST(I)
      T=0.04*FLOAT(L-1)
      SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
      GR(1)=CA+CB/EXP(X(3)*T)+CC*(SIN(W*T)*(Y10+Y11*T)+(1.-Y12*T)*COS(W
     1T))/EXP(Z*T)
      GR(2)=DB/EXP(X(3)*T)+DC*(SIN(W*T)*(Y13+Y14*T)+COS(W*T)*(1.-Y15*T)
     1/EXP(Z*T)
      GR(3)=EA+(EB+EC*T)/EXP(X(3)*T)+ED*RTZ*SIN(W*T+PH6)/EXP(Z*T)
      GR(4)=Y3-Y4/EXP(X(3)*T)+SIN(W*T+PHY)/(Y5*EXP(Z*T))
      GR(5)=Y6/EXP(X(3)*T)+SIN(W*T-PH4)/(Y7*EXP(Z*T))
      GR(6)=Y8/EXP(X(3)*T)+Y9*SIN(W*T+PHZ)/EXP(Z*T)
C     STEADY STATE COSTR. IMPOSED
      GR(1)=GR(1)+E*XY(3)*GR(4)
      GR(3)=GR(3)+E*XY(1)*GR(4)
      GR(4)=GR(5)
      GR(5)=GR(6)
      DO 3 J=1,N
      G(J)=G(J)-SIGMA*GR(J)
3     CONTINUE
4     CONTINUE
C
      GO TO 9
C
```

```
5       CONTINUE
        YN=-Y
        B1=X(2)*0.5+SQRT(YN)
        B2=X(2)*0.5-SQRT(YN)
        B3=B1-B2
        BB3=B3*B3
        BBB3=B3*B3*B3
        R1=A0/(X(1)*X(3))
        R2=(B1*B1-A1*B1+A0)/(B1*B3*(X(3)-B1))
        R3=(B2*B2-A1*B2+A0)/(B2*B3*(B2-X(3)))
        R4=(X(3)*X(3)-A1*X(3)+A0)/(X(3)*(B1-X(3))*(X(3)-B2))
        DO 6 I=1,201,2
        T=0.04*FLOAT(I-1)
        C3(I)=X(6)*(R1+R2/EXP(B1*T)+R3/EXP(B2*T)+R4/EXP(X(3)*T))
        ER(I)=C7(I)-C3(I)
        AER(I)=ABS(ER(I))
6       CONTINUE
        M=201
        CALL PEAKS (M,AER,INST,NPKS,K)
        F=AER(K)
        R5=(AX0-AX1*B1+AX2*B1*B1-B1*B1*B1)/(BB3)
        R6=(B1*B1*(3.*B2-B1)-2.*AX2*X(1)+AX1*X(2)-2.*AX0)/(BBB3)
        R7=(AX0-AX1*B2+AX2*B2*B2-B2*B2*B2)/(BB3)
        R8=(B2*B2*(3.*B1-B2)-2.*AX2*X(1)+AX1*X(2)-2.*AX0)/(BBB3)
        R9=(AY0-AY1*B1+AY2*B1*B1-B1*B1*B1)/(BB3)
        R10=(B1*B1*(3.*B2-B1)-2.*AY2*X(1)+AY1*X(2)-2.*AY0)/(BBB3)
        R11=(AY0-AY1*B2+AY2*B2*B2-B2*B2*B2)/(BB3)
        R12=(B2*B2*(3.*B1-B2)-2.*AY2*X(1)+AY1*X(2)-2.*AY0)/(BBB3)
        R13=1.0/(X(1)*X(3))
        R14=1.0/(X(3)*(B1-X(3))*(B2-X(3)))
        R15=1.0/(B2*B3*(X(3)-B2))
        R16=1./(B1*(B2-B1)*(X(3)-B1))
        R17=R16*B1
        R18=R15*B2
        R19=R14*X(3)
        R20=B1/(B3*(X(3)-B1))
        R21=B2/(B3*(B2-X(3)))
        R22=X(3)/((B1-X(3))*(X(3)-B2))
        DO 8 I=1,NPKS
        L=INST(I)
        T=0.04*FLOAT(L-1)
        SIGMA=(AER(L)/F)**(P-2)*(ER(L)/F)
        GR(1)=CA+CB/EXP(X(3)*T)+CC*((R5*T-R6)/EXP(B1*T)+(R7*T+R8)/EXP(B2*
       1))
        GR(2)=DB/EXP(X(3)*T)+DC*((R9*T-R10)/EXP(B1*T)+(R11*T+R12)/EXP(B2*
       1))
        GR(3)=EA+(EB+EC*T)/EXP(X(3)*T)-ED*((EE/ED-B1)/EXP(B1*T)+(B2-EE/ED
       1/EXP(B2*T))/B3
        GR(4)=R13-R14/EXP(X(3)*T)-R15/EXP(B2*T)-R16/EXP(B1*T)
        GR(5)=R17/EXP(B1*T)+R18/EXP(B2*T)+R19/EXP(X(3)*T)
        GR(6)=R20/EXP(B1*T)+R21/EXP(B2*T)+R22/EXP(X(3)*T)
C       STEADY STATE COSTR. IMPOSED
        GR(1)=GR(1)+E*XY(3)*GR(4)
        GR(3)=GR(3)+E*XY(1)*GR(4)
        GR(4)=GR(5)
        GR(5)=GR(6)
        DO 7 J=1,N
```

```
      G(J)=G(J)-SIGMA*GR(J)
7     CONTINUE
8     CONTINUE
9     CONTINUE
      AUX1=0.0
      DO 10 I=1,NPKS
      L=INST(I)
      AUX1=AUX1+(AER(L)/F)**P
10    CONTINUE
      RP=1.0/FLOAT(P)
      F=F*(AUX1**RP)
      SCALAR=(1.0/AUX1)**(1.0-RP)
      DO 11 J=1,N
      G(J)=G(J)*SCALAR
11    CONTINUE
      IF (KNT.GT.0) RETURN
      WRITE (6,12) (ER(I),I=1,201,2)
      RETURN
C
12    FORMAT (/3X,5E16.6)
      END
      SUBROUTINE PEAKS (M,AER,INST,NPKS,K)
      DIMENSION AER(1), INST(1)
      ML=M-2
      J=0
      AER(1)=0.0
      AER(M)=0.0
      DO 1 I=3,ML,2
      IF (AER(I-2).GT.AER(I).OR.AER(I+2).GT.AER(I)) GO TO 1
      J=J+1
      INST(J)=I
      NPKS=J
1     CONTINUE
C
C     TO FIND THE MAX ERROR
      BIG=AER(1)
      DO 2 I=1,NPKS
      L=INST(I)
      IF (AER(L).LE.BIG) GO TO 2
      BIG=AER(L)
      K=L
2     CONTINUE
      RETURN
      END
```

CD TOT 0274

REFERENCES

[1] J. W. Bandler, N. D. Markettos and N. K. Sinha, "Optimum approximation of high-order systems by low-order models using recent gradient methods", Proceedings 9th Annual Allerton Conference on Circuit and System Theory, Monticello, Ill., pp. 170-179, October 1971.

[2] J. W. Bandler, N. D. Markettos and N. K. Sinha, "Optimum system modelling using recent gradient methods", International Journal of Systems Science, accepted for publication.

[3] J. W. Bandler, N. D. Markettos and T. V. Srinivasan, "A comparison of recent minimax techniques on optimum system modelling", Proceedings 6th Annual Princeton Conference on Information Sciences and Systems, Princeton, New Jersey, March 1972.

[4] E. J. Davison, "A method for simplifying linear dynamic systems", IEEE Transactions on Automatic Control, vol. AC-11, pp. 93-101, January 1966.

[5] M. R. Chidambara, "Two simple techniques for the simplification

of large dynamic systems", Proceedings Joint Automatic Control Conference, pp. 669-674, 1969.

[6] D. Mitra, "The reduction of complexity of linear time-invariant dynamic systems", Proceedings 4th IFAC Congress, Warsaw, Poland, pp. 19-33, June 1969.

[7] S. A. Marshall, "An approximate method for reducing the order of linear systems", Control, vol. 10, pp. 642-643, December 1966.

[8] J. H. Anderson, "Geometrical approach to reduction of dynamical systems", Proceedings IEE, vol.114,pp. 1014-1018, July 1967.

[9] N. K. Sinha and W. Pille, "A new method for reduction of dynamic systems", International Journal of Control, vol. 14, pp. 111-118, July 1971.

[10] N. K. Sinha and G. T. Bereznai, "Optimum approximation of high-order systems by low-order models", International Journal of Control, vol. 14, pp. 951-959, November 1971.

[11] C. F. Chen and L. S. Shieh, "A novel approach to linear model simplification", Preprints, Joint Automatic Control Conference, Ann Arbor, Mich., pp. 454-460, June 1968.

[12] P. Kokotovic and P. Sannuti, "Singular perturbation method for reducing the model order in optimal design", Preprints, Joint Automatic Control Conference, Ann Arbor, Mich., pp. 468-476, June 1968.

[13] R. Hooke and T. A. Jeeves,"'Direct search' solution of numerical and statistical problems", Journal of the Association for Computing Machinery, vol. 8, pp. 212-219, April 1961

[14] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization", Computer Journal, vol. 6, pp. 163-168, June 1963.

[15] R. Fletcher, "A new approach to variable metric algorithms", Computer Journal, vol. 13, pp. 317-322, August 1970.

[16] D. H. Jacobson and W. Oksman, "An algorithm that minimizes homogeneous functions of N variables in N + 2 iterations and rapidly minimizes general functions", Technical Report No. 618, Harvard University, October 1970.

[17] R. C. Dorf, Modern Control Systems. Reading, Mass: Addison-Wesley, 1967, pp. 167-168.

[18] J. W. Bandler and C. Charalambous, "Practical least pth approximation with extremely large values of p", Proceedings

5th Asilomar Conference on Circuits and Systems, Pacific Grove, Calif., November 1971.

[19] J. W. Bandler, "Optimization methods for computer-aided design", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-17, pp. 533-552, August 1969.

[20] J. W. Bandler, T. V. Srinivasan and C. Charalambous, "Minimax optimization of networks by grazor search", IEEE Transactions on Microwave Theory Techniques, accepted for publication.

[21] J. W. Bandler and C. Charalambous", Conditions for minimax approximation obtained from the l_p norm", IEEE Transactions on Automatic Control, vol. AC-17, pp. 257-258, April 1972.

[22] N. K. Sinha and H. deBruin, "Near-optimal control of high-order systems using low-order models", International Journal of Control, accepted for publication.

[23] P. A. McCollum and B. F. Brown, Laplace Transforms Tables and Theorems, Holt, Rinehard and Winston.