MINIMAX SYSTEM MODELLING AND DESIGN

MINIMAX SYSTEM MODELLING AND DESIGN

by

Thandangorai V. Srinivasan, B. Tech. (Hons.), M.E.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

July 1973

DOCTOR OF PHILOSOPHY (1973) McMaster University
Hamilton, Ontario

TITLE : Minimax System Modelling and Design

AUTHOR : Thandangorai V. Srinivasan
B. Tech. (Hons.) (Indian Institute of Technology, Kharagpur)
M.E. (Birla Institute of Technology and Science, Pilani)

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

NUMBER OF PAGES: x, 175

SCOPE AND CONTENTS:

Computer-aided system modelling and design for minimax objectives have been considered in detail. A new algorithm for minimax approximation, called the grazor search method, has been proposed and successfully used on a number of network design problems to test the reliability and efficiency of the method. A critical comparison of the method with existing algorithms has shown the grazor search algorithm to be reliable in most of the problems considered. Practical ideas have been presented to deal with constrained minimax optimization problems and to investigate a solution for minimax optimality. Two user-oriented computer programs incorporating these ideas have been included as part of the thesis. Lower-order modelling of a high-order system has been considered for minimax objectives, and the suggested ideas make it feasible to design automated models for a variety of transient and steady-state constraint specifications.

ACKNOWLEDGEMENTS

The author is greatly indebted to Dr. J.W. Bandler for his expert guidance and supervision throughout the course of this work. The constant encouragement and inspiration the author derived from him are thankfully acknowledged.

The author wishes to acknowledge, in particular, Dr. N.K. Sinha for his advice and interest during the research effort. Special thanks are due to Dr. C.M. Crowe for his constructive criticisms and useful suggestions for the thesis.

Thanks are due to Dr. C. Charalambous for helping with the proof of convergence of the grazor search algorithm and many discussions the author had with him on a number of occasions. The author is very thankful to Dr. G. T. Bereznai for timely advice and helpful suggestions on various aspects of the research work.

The author wishes to acknowledge with thanks the useful discussions he had with his colleagues, in particular, B.L. Bardakjian, M. Beshai, J. Chen, V.K. Jha, A.G. Lee-Chan, P.C. Liu, N.D. Markettos, J.R. Popović, J. Roitman and S.K. Tam.

The financial assistance provided by the National Research Council of Canada through grants A7239 and C154, and the Department of Electrical Engineering is gratefully acknowledged.

The competence and reliability of Mrs. Hazel Coxall, who typed this thesis, has been very much appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page
Fig. 4.4(a)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_1, ℓ_2 fixed at ℓ_q and impedances varied. Starting point Z_1=1.0, Z_2=3.0.	53
Fig. 4.4(b)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. Z_1, Z_2 fixed at optimum values and lengths varied. Starting point ℓ_1/ℓ_q=0.8, ℓ_2/ℓ_q=1.2.	54
Fig. 4.4(c)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_2, Z_2 fixed at optimum values and ℓ_1, Z_1 varied. Starting point ℓ_1/ℓ_q=1.2, Z_1=3.5.	55
Fig. 4.4(d)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 4 parameters varied. Starting point ℓ_1/ℓ_q= 1.2, ℓ_2/ℓ_q=0.8, Z_1=3.5, Z_2=3.0.	56
Fig. 4.5(a)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_1,ℓ_2,ℓ_3 fixed at ℓ_q and impedances varied. Starting point Z_1=1.0, Z_2=3.16228, Z_3=10.0.	57
Fig. 4.5(b)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point ℓ_1/ℓ_q=0.8, ℓ_2/ℓ_q=1.2, ℓ_3/ℓ_q=0.8, Z_1=1.5, Z_2=3.0, Z_3=6.0.	58
Fig. 4.5(c)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point ℓ_1/ℓ_q= ℓ_2/ℓ_q=ℓ_3/ℓ_q= 1.0, Z_1=1.0, Z_2=3.16228, Z_3=10.0.	59
Fig. 4.6	Problem 1. Cascaded transmission-line filter operating between $R_g(\omega)$= $R_L(\omega)$=$377/\sqrt{1-(f_c/f)^2}$, where f_c=2.077 GHz and ℓ=1.5 cm.	64

Figure		Page

CHAPTER I

INTRODUCTION

Computer-aided design is now increasingly being accepted as a valuable tool whenever classical design techniques fail to achieve acceptable and realistic design criteria. This is especially true in electrical network analysis and synthesis where classical circuit theory restricts the network configuration and the degrees of freedom that may be demanded by the designer. Computer-aided network design has thus become a state-of-art which tries to accommodate the design specifications and constraints in a meaningful way so that design objectives, which would have been considered difficult by classical designers have now not only become feasible but are regularly being implemented on the digital computer. Many optimization algorithms have now been tested on a number of circuit design problems with the aim of improving circuit performance and convergence towards an optimal solution. The algorithms differ both in the way they generate downhill directions (directions of decreasing objective function value) and the computational effort involved.

It is thus apparent that there are two steps which are relevant to the circuit designer - the first one being that the design specifications, constraints involving the model parameters, and the objective function, have to be explicitly specified in advance, and the other being that a reliable and efficient algorithm has to be chosen for the optimization of the design variables. The emphasis of this work has been to bring both the system modelling and optimization techniques into the foreground so that the advantages and pitfalls encountered in the area of computer-aided design can be well appreciated.

This thesis concentrates mainly on minimax objectives, and Chapter II gives a brief review of existing minimax optimization methods, such as those by Osborne and Watson (1969), Bandler and Macdonald (1969b), and Bandler and Charalambous (1972d).

A new algorithm called the grazor search method has been developed which is guaranteed to converge under certain conditions. See Bandler and Srinivasan (1971) and Bandler, Srinivasan and Charalambous (1972). The problem of function minimization subject to constraints can now be formulated as a minimax problem (Bandler and Charalambous 1972a). This approach can be extended to tackle minimax optimization problems subject to constraints (Bandler and Srinivasan 1973a). Once a minimax solution has been achieved by the systems designer, it may be required to investigate the solution for optimality, and suitable methods are available for this investigation (Bandler and Srinivasan 1973c). Chapter III considers the above mentioned approaches to the minimax problem.

Chapter IV deals with the area of computer-aided electrical circuit design for minimax objectives. The problems considered include the design of lumped LC transformers and cascaded transmission-line networks acting as transformers or filters. A critical comparison has been made between the grazor search method and other optimization schemes for reliability and efficiency in convergence towards the optima.

System modelling is an area which demands attention primarily because of the complexity and computational effort involved when considering the original system, and the introduction of judiciously chosen models can not only reduce the complexity but also improve the computation time. It is now possible to model a high-order system and control this system on-line or off-line by dealing with the lower-order models directly. Chapter V deals with lower-order modelling of high-order systems for a variety of objectives and design considerations. Minimax objectives subject to arbitrary transient and steady-state constraints have been considered, and a method suggested by means of which the whole modelling procedure can be automated. See Bandler, Markettos and Srinivasan (1972, 1973), and Bandler and Srinivasan (1973b, 1973e).

Discussions and conclusions on the proposed methods are included in Chapter VI, while the Appendices A and B provide two computer program descriptions for minimax objectives (Bandler and Srinivasan 1972, 1973d).

The adjoint network method of evaluating the first-order derivatives was used for network design problems (Director and Rohrer 1969, Bandler and Seviora 1970). The CDC 6400 computer was used for the numerical experiments.

The purpose of this work can be described as an attempt to fill some of the gaps existing in the areas of approximation, optimization and system modelling.

CHAPTER II

REVIEW OF MINIMAX METHODS

2.1 Introduction

Minimax optimization methods are assuming significance in the computer-aided system design area and much effort has gone into the development of suitable algorithms for minimax objectives. The methods have been used to optimize electrical networks where the objective is to minimize the maximum deviation of a network response from an ideal response specification. This chapter gives a brief review of minimax optimization techniques.

2.2 Function Minimization

The problem of unconstrained function minimization consists of minimizing with respect to $\underset{\sim}{\phi}$ a real function

$$f \triangleq f(\underset{\sim}{\phi}) \tag{2.1}$$

where

$$\underset{\sim}{\phi} \triangleq [\phi_1 \; \phi_2 \; \cdots \; \phi_k]^T \tag{2.2}$$

is a column vector consisting of k independent parameter elements, T denotes the matrix transpose and f is the objective function.

The constrained version of the above problem, also known as the nonlinear programming problem, consists of minimizing $f(\underset{\sim}{\phi})$ subject to

$$g_i(\underset{\sim}{\phi}) \geq 0 \qquad i = 1,2, \ldots , m \tag{2.3}$$

where the g_i are, in general, nonlinear functions of the parameters.

2.3 Least pth Approximation for Single Specified Function

2.3.1 The Error Function

Define

$$e(\underset{\sim}{\phi},\psi) \triangleq w(\psi)(F(\underset{\sim}{\phi},\psi) - S(\psi)) \tag{2.4}$$

where

- $S(\psi)$ is a specified function (real or complex)
- $F(\underset{\sim}{\phi},\psi)$ is an approximating function (real or complex)
- $w(\psi)$ is a positive weighting function
- $e(\underset{\sim}{\phi},\psi)$ is the weighted error or deviation between $S(\psi)$ and $F(\underset{\sim}{\phi},\psi)$
- ψ is an independent variable (e.g., frequency or time)

2.3.2 Continuous Approximation

Define the norm

$$||e||_p \triangleq \left(\int_{\psi_\ell}^{\psi_u} |e(\underset{\sim}{\phi},\psi)|^p \, d\psi\right)^{1/p}, \quad 1 \leq p < \infty \tag{2.5}$$

where ψ_ℓ and ψ_u are lower and upper bounds, respectively, on the interval of approximation. Minimization of $||e||_p$ is called least pth approximation. For $p = 2$, we have the well-known least squares approximation.

Assume, for example, that $|e(\underset{\sim}{\phi},\psi)|$ is continuous on a finite closed interval $[\psi_\ell,\psi_u]$. The Chebyshev or uniform norm is given by

$$||e||_\infty \triangleq \max_{[\psi_\ell,\psi_u]} |e(\underset{\sim}{\phi},\psi)| \tag{2.6}$$

The process of minimization of $||e||_\infty$ is called minimax or Chebyshev approximation.

It may be noted that

$$||e||_\infty = \lim_{p\to\infty} \left[\frac{1}{\psi_u - \psi_\ell} \int_{\psi_\ell}^{\psi_u} |e(\phi,\psi)|^p \, d\psi \right]^{1/p} \qquad (2.7)$$

The larger the value of p, the more emphasis will be given to the maximum absolute error, and the optimal least pth solution should be closer to the optimal minimax solution.

2.3.3 Discrete Approximation

In practice the various functions contained in (2.4) are usually evaluated at discrete values ψ_i. It is thus appropriate to consider discrete approximation.

Define the norm

$$||e||_p \triangleq \left(\sum_{i\in I} |e_i(\phi)|^p \right)^{1/p} \quad 1 \le p < \infty \qquad (2.8)$$

where

$$e(\phi) \triangleq [e_1(\phi) \;\; e_2(\phi) \;\; \ldots \;\; e_n(\phi)]^T \qquad (2.9)$$

and

$$I \triangleq \{1,2,\ldots, n\} \qquad (2.10)$$

The process of minimization of $||e||_p$ is called discrete least pth approximation. The discrete minimax norm may be defined as

$$||e||_\infty \triangleq \max_{i\in I} |e_i(\phi)| \qquad (2.11)$$

and minimization of $||\underset{\sim}{e}||_\infty$ is called discrete minimax approximation.

As mentioned earlier,

$$||\underset{\sim}{e}||_\infty = \lim_{p\to\infty} ||\underset{\sim}{e}||_p \tag{2.12}$$

and the same comments hold as in the continuous case.

For a sufficiently large number of uniformly sampled values of ψ and with suitable weighting factors, the discrete approximation approaches the continuous approximation.

2.4 The Minimax Problem

Unless otherwise mentioned, the unconstrained discrete nonlinear minimax problem that is considered throughout this work consists of minimizing

$$U(\underset{\sim}{\phi}) \triangleq \max_{i\in I} y_i(\underset{\sim}{\phi}) \tag{2.13}$$

where I, as defined in (2.10), is an index set relating to discrete elements corresponding to the i, and the y_i are, in general, nonlinear differentiable functions. It is desired to find a point $\check{\underset{\sim}{\phi}}$ such that

$$\check{U} \triangleq U(\check{\underset{\sim}{\phi}}) = \min_{\underset{\sim}{\phi}} \max_{i\in I} y_i(\underset{\sim}{\phi}) \tag{2.14}$$

where $\check{\underset{\sim}{\phi}}$ is a local or global minimax optimum.

2.5 Minimax Methods

Many methods use the direct minimax formulation of (2.13) which, in general, gives rise to discontinuous partial derivatives of the objective function with respect to the variable parameters. Otherwise efficient optimization methods may slow down or even fail to reach an optimum in such circumstances, particularly when the response hypersurface has a narrow curved valley along which the path of discontinuous derivatives lies.

In direct search strategies, the minimax problem has been explored using pattern search and razor search (Bandler and Macdonald 1969a, 1969b). Of the gradient strategies, there are methods involving the penalty function approach (Fiacco and McCormick 1964a, 1964b), linear programming (Osborne and Watson 1969, Ishizaki and Watanabe 1968), quadratic programming (Heller 1969), and a method proposed by Bandler and Lee-Chan (1971).

Whenever efficient methods of finding derivatives are not available, direct search methods are useful. For electrical networks, in particular, it is now possible to evaluate the derivatives of network responses with respect to network parameters rather easily using the adjoint network approach (Director and Rohrer 1969, Bandler and Seviora 1970), and the gradient methods are thus more suited for such cases. The quadratic programming methods are usually more time-consuming than solution of linear programming problems, while penalty function methods rely on suitable function minimization algorithms.

2.5.1 The Razor Search Method

The razor search method of Bandler and Macdonald (1969b, 1971) essentially begins with a modified version of the pattern search (Hooke and Jeeves 1961) until this fails. A random point is selected automatically in the neighbourhood and a second pattern search is initiated until this one fails. Using the two points where pattern search failed, a new pattern in the direction of the optimum is established and a pattern search strategy resumed until it too fails. This process is repeated until any of several possible terminating criteria is satisfied. Thus, the strategy tries to negotiate certain kinds of "razor sharp" valleys in multidimensional space. The method has been compared with other direct search methods on some test problems, and has been found to be reliable and computationally efficient in most of the cases.

2.5.2 Sequential Unconstrained Minimization Technique

The nonlinear minimax optimization problem of Section 2.4 may be transformed into a nonlinear programming problem (Waren, Lasdon and Suchman 1967) of Section 2.2 as follows

$$\text{Minimize} \qquad \phi_{k+1} \tag{2.15}$$

subject to

$$\phi_{k+1} - y_i(\underset{\sim}{\phi}) \geq 0 \qquad i\epsilon I \tag{2.16}$$

The nonlinear programming problem may, in turn, be solved by well-established methods such as the Sequential Unconstrained Minimization Technique (SUMT) due to Fiacco and McCormick (1964a, 1964b), which is a development of the Created Response Surface Technique (CRST) suggested by Carroll (1961). The problem of (2.15) and (2.16) may be reformulated as follows. Minimize

$$P(\underset{\sim}{\phi},\phi_{k+1},r) = \phi_{k+1} + r \sum_{i\in I} \frac{w_i}{\phi_{k+1}-y_i(\underset{\sim}{\phi})} \tag{2.17}$$

where

$$\phi_{k+1} \text{ is an independent variable, and}$$
$$r,w_i > 0 \quad i\epsilon I \tag{2.18}$$

$P(\underset{\sim}{\phi},\phi_{k+1},r)$ is an unconstrained objective where points close to the constraint boundaries are penalized.

Define the interior of the region of feasible points as

$$R^0 \triangleq \{\underset{\sim}{\phi},\phi_{k+1} | \phi_{k+1} - y_i(\underset{\sim}{\phi}) > 0 , \quad i\epsilon I\} \tag{2.19}$$

where the region of feasible points is

$$R \triangleq \{\underset{\sim}{\phi},\phi_{k+1} | \phi_{k+1} - y_i(\underset{\sim}{\phi}) \geq 0 , \quad i\epsilon I\} \tag{2.20}$$

Starting with a point $\underset{\sim}{\phi},\phi_{k+1}$ and a value of r, initially r_1, such that $\underset{\sim}{\phi},\phi_{k+1}\epsilon R^0$ and $r_1>0$ the unconstrained function $P(\underset{\sim}{\phi},\phi_{k+1},r_1)$ is minimized with respect to $\underset{\sim}{\phi}$ and ϕ_{k+1}. The form of (2.17) leads one to expect that a minimum will lie in R^0, since as any one of the $\phi_{k+1} - y_i(\underset{\sim}{\phi})$ approaches 0, P approaches ∞. The location of the minimum will depend on the value of r_1 and is denoted by $\check{\underset{\sim}{\phi}}(r_1),\check{\phi}_{k+1}(r_1)$.

This procedure is repeated for a decreasing sequence of r values such that

$$r_1 > r_2 > \ldots > r_j > 0 \tag{2.21}$$

$$\lim_{j\to\infty} r_j = 0 \tag{2.22}$$

each minimization being started at the previous minimum. For example, the minimization of $P(\phi, \phi_{k+1}, r_2)$ would be started at $\check{\phi}(r_1)$ and $\check{\phi}_{k+1}(r_1)$. Every time r is reduced, the effect of the penalty is reduced, so that one would expect in the limit as $j \to \infty$ and $r_j \to 0$ that $\check{\phi}(r_j) \to \check{\phi}$ and, consequently, that $\phi_{k+1}(r_j) \to U(\check{\phi})$, the minimax optimum.

Conditions which guarantee convergence have been proved by Fiacco and McCormick. It is important that the initial value of r chosen is realistic, and r should be reduced systematically after each iterative cycle of minimization of P.

2.5.3 Algorithm due to Osborne and Watson

This minimax algorithm (Osborne and Watson 1969, Watson 1970) deals with minimax formulations by following two steps - a linear programming part that provides a given step in the parameter space, followed by a linear search along the direction of the step. This algorithm is very similar to the one proposed by Ishizaki and Watanabe (1968) and works very well for many minimax problems. In cases where the linear approximation is not very good in the vicinity of the optimum, the method may fail to converge toward the optimum for successive iterations.

Consider the problem of minimizing $||e(\phi)||_\infty$ in (2.11), where e consists of real elements. Linearizing $e_i(\phi)$ at some point ϕ^j the problem may be stated as

Minimize ϕ_{k+1}

subject to

$$\phi_{k+1} - e_i(\underset{\sim}{\phi}^j) - \underset{\sim}{\nabla}^T e_i(\underset{\sim}{\phi}^j)\Delta\underset{\sim}{\phi}^j \geq 0 \qquad i\varepsilon I \tag{2.23}$$

$$\phi_{k+1} + e_i(\underset{\sim}{\phi}^j) + \underset{\sim}{\nabla}^T e_i(\underset{\sim}{\phi}^j)\Delta\underset{\sim}{\phi}^j \geq 0$$

where

$$\underset{\sim}{\nabla} = \left[\frac{\partial}{\partial\phi_1} \ \frac{\partial}{\partial\phi_2} \ \cdots \ \frac{\partial}{\partial\phi_k}\right]^T \tag{2.24}$$

$$n > k \tag{2.25}$$

$\underset{\sim}{\nabla}$ is the first partial derivative operator with respect to the parameter vector $\underset{\sim}{\phi}$,

Δ denotes incremental changes, and

n is the number of elements of I.

Noting that the variables for linear programming should all be nonnegative, and imposing a rather practical constraint that the elements of $\underset{\sim}{\phi}$ should not change sign we have the linear programming problem in

$$\underset{\sim}{x} \triangleq [x_1 \ x_2 \ \cdots \ x_{k+1}]^T \tag{2.26}$$

as follows.

Step 1

$$\text{Minimize } x_{k+1} \tag{2.27}$$

subject to (2.25) and

$$\pm(e_i(\underset{\sim}{\phi}^j) + \underset{\sim}{\nabla}^T e_i(\underset{\sim}{\phi}^j) \begin{bmatrix} \phi_1^j x_1 - \phi_1^j \\ \phi_2^j x_2 - \phi_2^j \\ \vdots \\ \phi_k^j x_k - \phi_k^j \end{bmatrix}) \leq x_{k+1} \quad i\varepsilon I \tag{2.28}$$

$$\underset{\sim}{x} \geq \underset{\sim}{0} \qquad (2.29)$$

where

$$x_\ell \triangleq \frac{\Delta\phi_\ell^{\,j}}{\phi_\ell^{\,j}} + 1 \qquad \ell = 1,2, \ldots, k$$
$$x_{k+1} \triangleq \phi_{k+1} \qquad (2.30)$$

The solution produces a direction given by $\Delta\underset{\sim}{\phi}^j$.

Step 2

Next we find γ^{j*} such that

$$\max_{i\in I} |e_i(\underset{\sim}{\phi}^j + \gamma^j \Delta\underset{\sim}{\phi}^j)| \qquad (2.31)$$

is a minimum with respect to γ^j. Set

$$\underset{\sim}{\phi}^{j+1} = \underset{\sim}{\phi}^j + \gamma^{j*} \Delta\underset{\sim}{\phi}^j \qquad (2.32)$$

and return to Step 1.

The convergence of the method holds under certain conditions (Osborne and Watson 1969). This approach is directly applicable to linear functions such as polynomials, for which k+1 equal extrema results at the optimum.

2.5.4 Method due to Bandler and Lee-Chan

The nonlinear minimax objective given by (2.13) is minimized here by exploiting the gradient information of the local discrete maxima of the functions $y_i(\underset{\sim}{\phi})$ to get a downhill direction by solving a set of simultaneous equations. The method works very well, except that in the case of linear dependence of the equations, some problems may arise in the convergence toward the optimum. See Bandler and Lee-Chan (1971).

2.6 Near-Minimax Methods

As is well-known to network designers, least pth approximation for sufficiently large values of p can result in an optimal solution very close to the optimal minimax solution (Temes and Zai 1969, Temes 1969, Bandler 1969a, Seviora, Sablatash and Bandler 1970).

When appropriate error functions are raised to a power p given by

$$\bar{f}(\underset{\sim}{\phi}) = \sum_{i \in I} |e_i(\underset{\sim}{\phi})|^p \tag{2.33}$$

and $f(\underset{\sim}{\phi})$ is minimized, ill-conditioning may result for nominal values of p (usually greater than or equal to about 10). The objective function of the form (2.33) has been used by a number of authors (Temes and Zai 1969, Temes 1969, Bandler 1969a, Bandler and Seviora 1970).

Bandler and Charalambous (1972c, 1972d) have given a unified approach to the least pth approximation problems, as encountered in network and system design, having upper and lower response specifications e.g., as in filter design. The ill-conditioning is removed by proper scaling, and least pth optimization has been carried out for extremely large values of p, typically 10^3 to 10^6. This approach has been used extensively in a variety of computer-aided network design problems (Bandler and Bardakjian 1973, Bandler and Charalambous 1972d, Bandler, Charalambous and Tam 1972, Bandler and Jha 1972, Popovic 1972, Charalambous 1973).

The least pth approximation problem can effectively be tackled by efficient gradient minimization techniques such as the Fletcher - Powell method (1963), Jacobson - Oksman algorithm (1972), and a more recent method due to Fletcher (1970). These methods have been compared critically for near-minimax approximation problems in the area of lower-order modelling of high-order systems (Bandler, Markettos and Srinivasan 1972, 1973).

The discrete nonlinear minimax approximation problem of Section 2.4 can be formulated as a least pth approximation problem (Bandler 1972). Suppose at least one of the functions $y_i(\underset{\sim}{\phi})$ is positive. Then, since $U(\underset{\sim}{\phi}) > 0$,

$$U(\underset{\sim}{\phi}) = \lim_{p \to \infty} U(\underset{\sim}{\phi}) \left(\sum_{i \in I} \left(\frac{w_i y_i(\underset{\sim}{\phi})}{U(\underset{\sim}{\phi})} \right)^p \right)^{1/p} \tag{2.34}$$

where

$$w_i = \begin{cases} 0 & \text{for } y_i < 0 \\ 1 & \text{for } y_i \geq 0 \end{cases} \tag{2.35}$$

Suppose all the functions y_i are negative. Then, since $U(\underset{\sim}{\phi}) < 0$,

$$U(\underset{\sim}{\phi}) = \lim_{p \to -\infty} U(\underset{\sim}{\phi}) \left(\sum_{i \in I} \left(\frac{w_i y_i(\underset{\sim}{\phi})}{U(\underset{\sim}{\phi})} \right)^p \right)^{1/p} \tag{2.36}$$

where

$$w_i = 1 \quad \text{for all} \quad y_i < 0 \tag{2.37}$$

Therefore, the minimization function is chosen as

$$f(\underset{\sim}{\phi}) = U(\underset{\sim}{\phi}) \left(\sum_{i \in I} \left(\frac{w_i y_i(\underset{\sim}{\phi})}{U(\underset{\sim}{\phi})} \right)^q \right)^{1/q} \tag{2.38}$$

where

$$q \triangleq \frac{U(\phi)}{|U(\phi)|} p \quad \begin{cases} 1<p<\infty \text{ for } U>0 \\ 1\leq p<\infty \text{ for } U<0 \end{cases} \tag{2.39}$$

A number of interesting features of $f(\phi)$ can be stated. For $1<|q|<\infty$, q having the appropriate sign, and for appropriate values of w_i, in accordance with (2.35) for $U(\phi) > 0$ and (2.37) for $U(\phi) < 0$, we have a continuous function $f(\phi)$ with continuous derivatives with respect to ϕ so long as $U(\phi) \neq 0$. When $U(\phi) > 0$, $f(\phi)$ is like penalty term including violated constraints, in this case only positive y_i, which it is desired to make feasible (or acceptable). If $\min_{\phi} f(\phi) > 0$, the constraints remain violated. In least pth approximation this indicates that the specifications have not been satisfied. When $U(\phi) < 0$ the specifications are satisfied and $f(\phi)$ is like a penalty term designed to move a solution as far from the boundary of the feasible region as possible.

CHAPTER III

NEW APPROACHES TO THE MINIMAX PROBLEM

3.1 Introduction

In this chapter a new gradient algorithm for minimax objectives called the grazor search (or gradient razor search) method is introduced (Bandler and Srinivasan 1971, Bandler, Srinivasan and Charalambous 1972). As the name suggests, the method attempts to follow the path of discontinuous derivatives when encountering razor-sharp valleys in multidimensional parameter space. The method is especially suitable for nonlinear minimax optimization of network and system responses. This algorithm uses the gradient information of one or more of the highest ripples in the error function to produce a downhill direction by solving a suitable linear programming problem. A linear search follows to find the minimum in that direction, and the procedure is repeated. This type of descent process is repeated with as many ripples as necessary until a minimax solution is reached to some desired accuracy. Unlike the razor search method due to Bandler and Macdonald (1969b), the present method overcomes the problem of discontinuous derivatives characteristic of minimax objectives without using random moves. It can fully exploit the advantages of the adjoint network method of evaluating partial derivatives of the response function with respect to the variable parameters (Director and Rohrer 1969, Bandler and Seviora 1970).

The problem of constrained minimax optimization is considered next. This problem has been reformulated as an unconstrained minimax problem by two methods, one extending a recently proposed method due to Bandler and Charalambous (1972a, 1973b) and the other using weighting functions. The reformulated problem can then be tackled by efficient unconstrained minimax algorithms. The method has a number of applications, including high-order system modelling and control system designs, where constraints have to be imposed on the pole-zero locations of the models chosen. Appropriate constraints can also be imposed on the upper and lower bounds of the parameter values. See Bandler and Srinivasan (1973a, 1973e).

Investigation of optimality conditions of a proposed or a design solution is of great practical importance to the system designer wishing to approximate a desired response by a system response. Conditions for optimality in the minimax sense in conventional synthesis problems involving polynomials and rational functions are fairly widely appreciated. However, with the ever-increasing need for network designs containing elements not conducive to the rational function approach, e.g., a mixture of lumped and distributed elements, and the application of automatic optimization methods involving least pth and minimax objectives, some means of testing for convergence to an optimum for more arbitrary problems is highly desirable. Depending on the optimization method employed, a satisfactory minimax solution may be obtained for a problem after a number of iterations of the algorithm on the computer. It may then be required to investigate the solution for minimax optimality (Bandler 1971) so as to verify whether the solution is optimal or not. Though the necessary optimality conditions may seem to be straightforward to verify, they are both tedious and difficult to implement in practice. A practical way of implementing them is considered in detail. See Bandler and Srinivasan (1973c, 1973d).

3.2 The Grazor Search Strategy

3.2.1 Theoretical Considerations

The grazor search algorithm is a generalization of the method due to Bandler and Lee-Chan (1971), and is basically of the steepest descent type. The nonlinear minimax optimization problem is the one already stated in Section 2.4.

Define a subset $J \subseteq I$ such that

$$J(\underset{\sim}{\phi}^j, \varepsilon^j) \triangleq \{i \mid U(\underset{\sim}{\phi}^j) - y_i(\underset{\sim}{\phi}^j) \leq \varepsilon^j, \ i \varepsilon I\} \tag{3.1}$$

$$\varepsilon^j \geq 0 \tag{3.2}$$

where

$\underset{\sim}{\phi}^j$ denotes a feasible point at the beginning of the jth iteration, and

ε^j is the tolerance with respect to the current $\max\limits_{i \varepsilon I} y_i(\underset{\sim}{\phi}^j)$ within which the y_i for $i \varepsilon J$ lie.

Linearizing y_i at $\underset{\sim}{\phi}^j$, we can consider the first-order changes

$$\delta y_i(\underset{\sim}{\phi}^j) = \underset{\sim}{\nabla}^T y_i(\underset{\sim}{\phi}^j) \Delta \underset{\sim}{\phi}^j \qquad i \varepsilon J(\underset{\sim}{\phi}^j, \varepsilon^j) \tag{3.3}$$

A sufficient condition for $\Delta \underset{\sim}{\phi}^j$ to define a descent direction for $U(\phi^j)$ is

$$\nabla^T y_i(\phi^j)\Delta\phi^j < 0 \qquad i\varepsilon J(\phi^j,\varepsilon^j) \tag{3.4}$$

Consider

$$\Delta\phi^j = - \sum_{i\varepsilon J} \alpha_i^j \nabla y_i(\phi^j) \tag{3.5}$$

$$\sum_{i\varepsilon J} \alpha_i^j = 1 \tag{3.6}$$

$$\alpha_i^j \geq 0 \tag{3.7}$$

(3.4) may now be written as

$$-\nabla^T y_i(\phi^j) \sum_{i\varepsilon J} \alpha_i^j \nabla y_i(\phi^j) < 0 \qquad i\varepsilon J(\phi^j,\varepsilon^j) \tag{3.8}$$

which suggests the linear program:

Maximize

$$\alpha^j_{k_r+1}(\phi^j,\varepsilon^j) \geq 0 \tag{3.9}$$

subject to

$$-\nabla^T y_i(\phi^j) \sum_{i\varepsilon J} \alpha_i^j \nabla y_i(\phi^j) \leq -\alpha^j_{k_r+1} \qquad i\varepsilon J(\phi^j,\varepsilon^j) \tag{3.10}$$

plus (3.6) and (3.7), where k_r denotes the number of elements of $J(\phi^j,\varepsilon^j)$. Note that if

$$\Delta\phi^j = 0 \qquad \text{for } \varepsilon^j = 0$$

the necessary conditions for a minimax optimum are satisfied at ϕ^j

(Bandler 1971). Observe that J is non-empty and that if J has only one element, we obtain the steepest descent direction for the corresponding maximum of the $y_i(\phi)$.

3.2.2 Proof of Convergence

Before proving the convergence of the algorithm it may be worth restating the following lemma due to Farkas (Lasdon 1970).

Let $\{p_0, p_1, \ldots, p_n\}$ be an arbitrary set of vectors. There exist

$$\beta_i \geq 0 \tag{3.11}$$

such that

$$p_0 = \sum_{i=1}^{n} \beta_i p_i \tag{3.12}$$

if and only if

$$p_0^T q \geq 0 \tag{3.13}$$

for all q satisfying

$$p_i^T q \geq 0 \qquad i = 1, 2, \ldots, n \tag{3.14}$$

It is, therefore, possible to find nonnegative values of α_i^j in the expression for (3.5) if and only if

$$(-\Delta\phi^j)^T(-\Delta\phi^j) \geq 0 \tag{3.15}$$

for all $\Delta\underset{\sim}{\phi}^j$ satisfying

$$\underset{\sim}{\nabla}^T y_i(\underset{\sim}{\phi}^j)(-\Delta\underset{\sim}{\phi}^j) \geq 0 \qquad i\epsilon J(\underset{\sim}{\phi}^j, \epsilon^j) \tag{3.16}$$

where (3.13) and (3.14) correspond to (3.15) and (3.16), respectively, and $-\Delta\underset{\sim}{\phi}^j$, $\underset{\sim}{\nabla} y_i(\underset{\sim}{\phi}^j)$, $-\Delta\underset{\sim}{\phi}^j$ take the place of $\underset{\sim}{p}_0$, $\underset{\sim}{p}_i$, $\underset{\sim}{q}$.

Now (3.15) is always satisfied, though it may not be possible to satisfy (3.16) if ϵ^j is too large. By suitably decreasing ϵ^j, (3.16) may be forced to hold.

3.2.3 Practical Implementation

Fig. 3.1 illustrates how the different subroutines are called and their relative hierarchy. Flow charts of subroutines GRAZOR, SELEC and GOLDEN appear in Figs. 3.2 - 3.4. See Appendix A for further details and definitions. The objective function $U(\underset{\sim}{\phi}^j)$ is calculated by subroutine LOCATE.

As given by linear programming (see, for example, Subroutine SIMPLE), $\Delta\underset{\sim}{\phi}^j$ is normalised to

$$\Delta\underset{\sim}{\phi}_n^j = \frac{\Delta\underset{\sim}{\phi}^j}{||\Delta\underset{\sim}{\phi}^j||} \tag{3.17}$$

by subroutine NORM. Starting at $\underset{\sim}{\phi}^j$, a step $\alpha^j\Delta\underset{\sim}{\phi}_n^j$ is taken for $\alpha^j = \alpha_o^j$; if no improvement in U results, α^j is reduced by factors of β until a better point is obtained or $\alpha^j < \check{\alpha}$. Let α^{j*} produce the first improved point from $\underset{\sim}{\phi}^j$. Then

$$\Delta\underset{\sim}{\phi}^o = \alpha^{j*}\,\Delta\underset{\sim}{\phi}_n^j \tag{3.18}$$

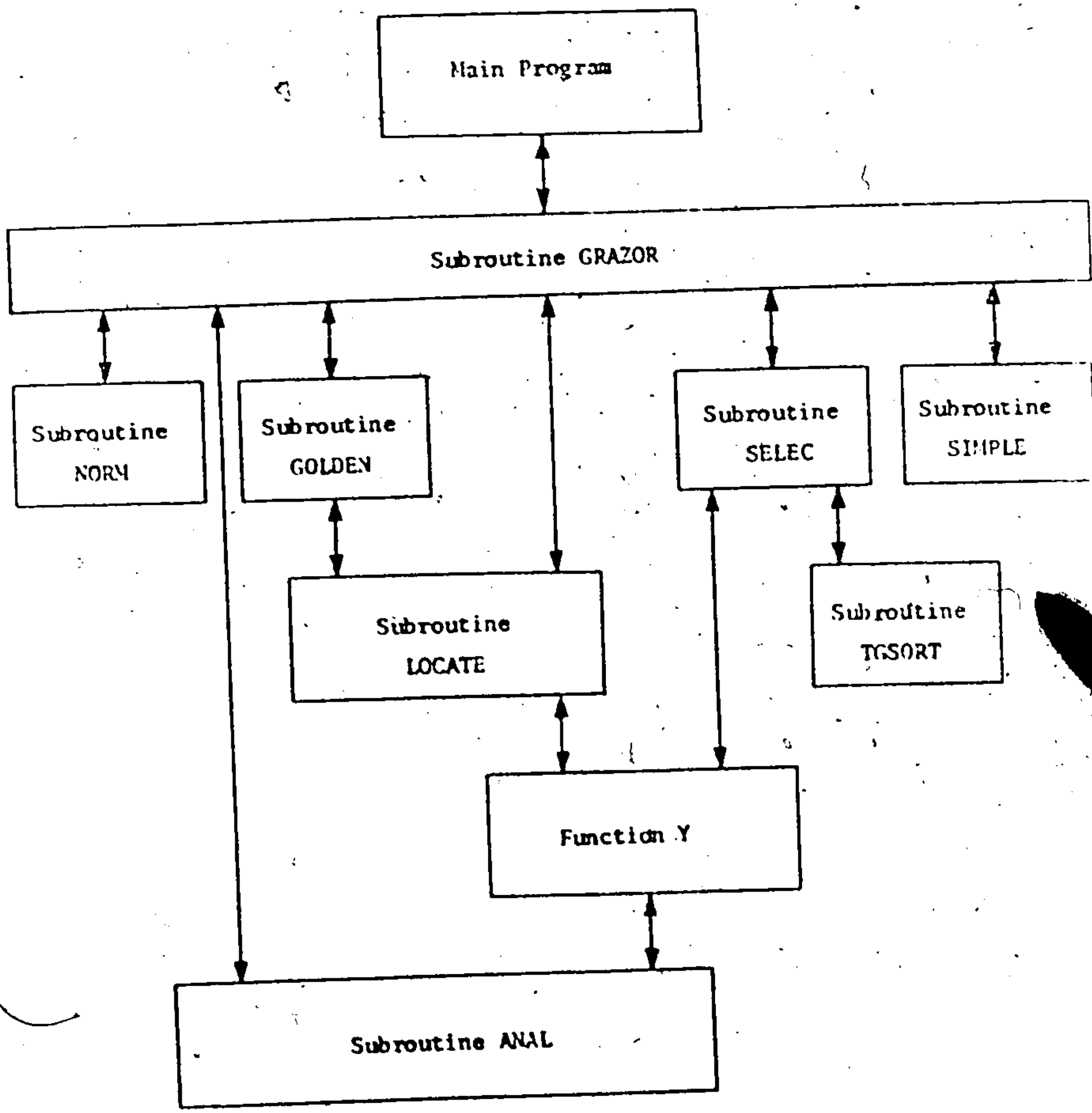


Fig. 3.1 Block diagram summarizing the computer program structure and illustrating the relative hierarchy of the subprograms.

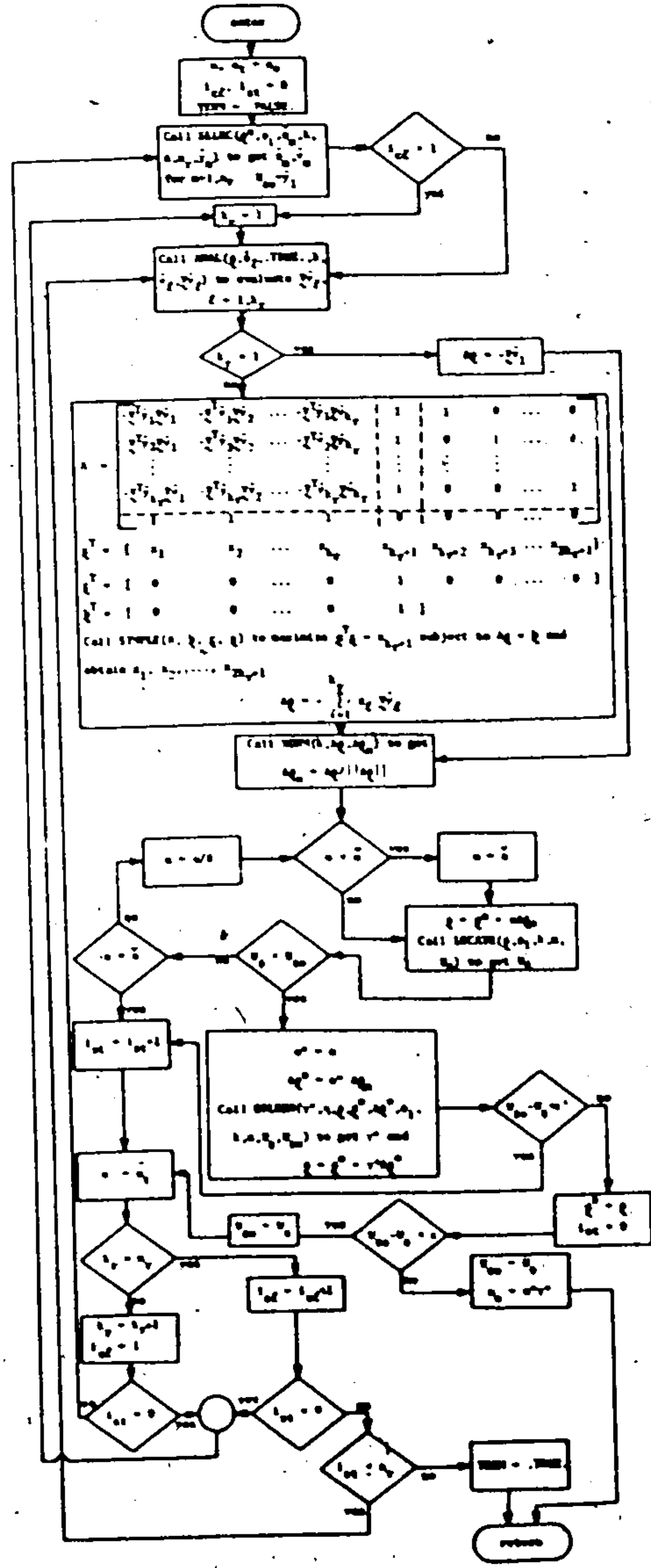


Fig. 3.2 Mathematical flow diagram of subroutine

GRAZOR (α_0, χ, β, ϵ, ϵ', η, ϕ^o, ϕ_1, k, k_r, n, n_r, $U_{\phi o}$, TERM)

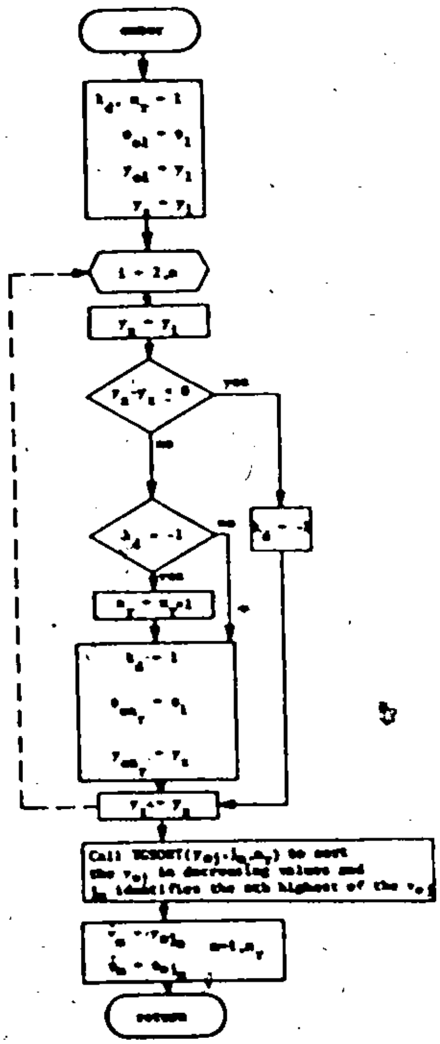


Fig. 3.3 Mathematical flow diagram of subroutine

SELEC ($\underset{\sim}{\phi}^{o}, \psi_i, \hat{\psi}_m, k, n, n_r, \hat{y}_m$)

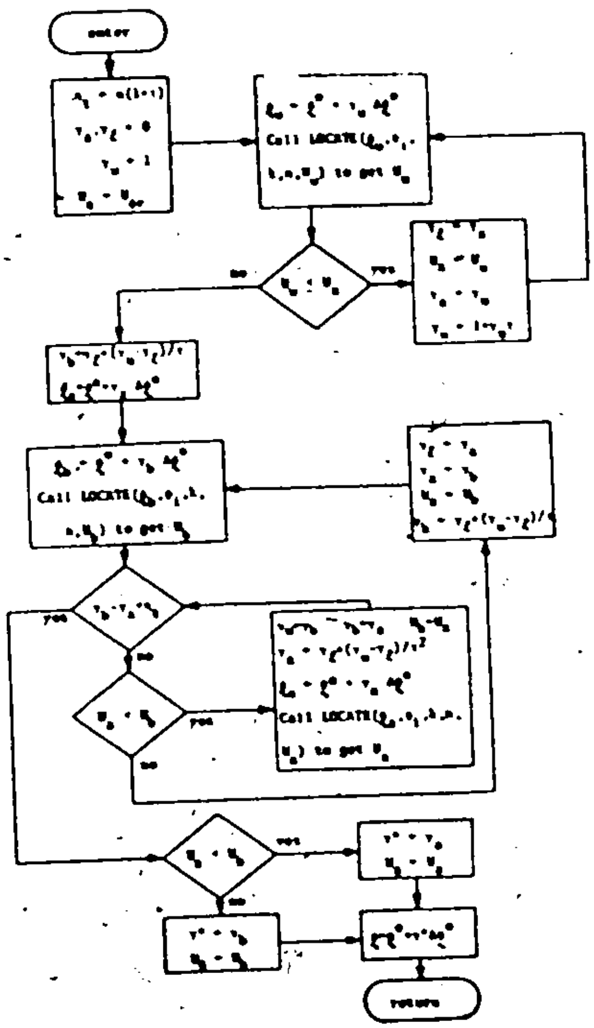


Fig. 3.4 **Mathematical flow diagram of subroutine**

GOLDEN $(\gamma^*, \eta, \underset{\sim}{\phi}, \underset{\sim}{\phi}^o, \Delta\underset{\sim}{\phi}^o, \psi_i, k, n, U_\phi, U_{\phi o})$

is defined.

Next a method based on golden section search (Temes 1969) is used to find γ^{j^*} corresponding to the constrained minimum value of $\max_{i \in I} y_i(\underset{\sim}{\phi}^j + \gamma^j \Delta\underset{\sim}{\phi}^j)$. The jth iteration ends by setting

$$\underset{\sim}{\phi}^{j+1} = \underset{\sim}{\phi}^j + \gamma^{j^*} \Delta\underset{\sim}{\phi}^o \qquad (3.19)$$

and

$$\alpha_o^{j+1} = \alpha^{j^*} \gamma^{j^*} \qquad (3.20)$$

In Fig. 3.4,

$$\tau = \frac{1}{2} (1+\sqrt{5}) \qquad (3.21)$$

is the factor associated with the golden section. Subscripts ℓ and u denote lower and upper limits, respectively, and a and b denote interior points of the interval of search. An attempt to bound the minimum is made. Then golden section search is used to locate the minimum to a desired accuracy. The search is terminated when the resolution between two interior points falls below a factor η of the initial interval.

In Fig. 3.3 the maxima implied by the functions y_i, sampled in a certain order, are located and sorted out in decreasing magnitude (by, say, Subroutine TGSORT).

Fig. 3.2 shows the grazor search strategy. Note that in setting up

$$\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{b} \qquad (3.22)$$

slack variables $(x_{k_r+2}, x_{k_r+3}, \ldots, x_{2k_r+1})$ are introduced. We try to generate a descent direction based on the gradient of the maximum function ($k_r = 1$), proceed to the minimum of U in that direction, and repeat the process. If, at any stage, this process or the linear program does not yield a direction of decreasing U, or does not provide an improvement greater than ϵ, the procedure is repeated after including the function corresponding to the next largest of the current n_r discrete local maxima (i.e., ripples) if one exists. When all local maxima have been included and U can still not be reduced or improved satisfactorily by a value greater than ϵ, we repeat the procedure with k_r functions corresponding to the first k_r largest of the candidates, beginning with $k_r = 1$, in another series of attempts to reduce U. The algorithm terminates only when there are no more suitable functions left and when there are either no improvements or improvements less than ϵ! over one complete cycle of k_r, starting from 1 and ending with n_r.

3.2.4 Example

The design of a two-section 10Ω to 1Ω quarter-wave transmission-line transformer network over a 100 percent bandwidth centred at 1GHz is considered (Matthaei, Young and Jones 1964) as an example for testing the grazor search strategy. This problem has already received attention from the optimization point of view (Bandler and Macdonald 1969a, 1969b). The lengths ℓ_1, ℓ_2 are fixed at ℓ_q, the quarter-wavelength at centre frequency, and the impedances Z_1, Z_2 are varied.

Table 3.1, in association with Fig. 3.5, illustrates how the grazor search strategy effectively follows the path of discontinuous derivatives to locate the optimum in the course of minimax optimization

TABLE 3.1

SUMMARY OF IMPORTANT STEPS IN THE EXAMPLE
ILLUSTRATING THE GRAZOR SEARCH STRATEGY

$\check{\underset{\sim}{\phi}} = (2.23605,\ 4.47210),\ U'(\check{\underset{\sim}{\phi}}) = 0.42857$

Iteration Number	Points of Iteration	Starting Point of Iteration	Values of Scale Factors		k_j
			Point	Scale Factor	
1	1-5	$\underset{\sim}{\phi}^1=(1.0,\ 3.0)$	$\underset{\sim}{\phi}^2$	$\alpha^*=1.00$	1
		$U'(\underset{\sim}{\phi}^1)=0.70954$	$\underset{\sim}{\phi}^4$	$\gamma=1+\tau$	
			$\underset{\sim}{\phi}^5=\underset{\sim}{\phi}^2$	$\gamma^*=1.000$	
2	5-12	$\underset{\sim}{\phi}^5=(1.99996,\ 3.00893)$	$\underset{\sim}{\phi}^6$	$\alpha=1.00$	1
		$U'(\underset{\sim}{\phi}^5)=0.63086$	$\underset{\sim}{\phi}^7$	$\alpha^*=0.10$	
			$\underset{\sim}{\phi}^{12}$	$\gamma^*=2+\tau$	
3	12-20	$\underset{\sim}{\phi}^{12}=(1.69865,\ 3.20921)$	$\underset{\sim}{\phi}^{13}$	$\alpha=0.1(\tau+2)$	1
		$U'(\underset{\sim}{\phi}^{12})=0.48073$	$\underset{\sim}{\phi}^{14}$	$\alpha=0.01(\tau+2)$	
			$\underset{\sim}{\phi}^{15}$	$\alpha^*=0.001(\tau+2)$	
			$\underset{\sim}{\phi}^{20}$	$\gamma^*=\tau+1$	
4	20-26	$\underset{\sim}{\phi}^{20}=(1.70806,\ 3.20821)$	$\underset{\sim}{\phi}^{21}$	$\alpha=9.472\times10^{-3}$	[illegible]
		$U'(\underset{\sim}{\phi}^{20})=0.47843$	$\underset{\sim}{\phi}^{22}$	$\alpha^*=9.472\times10^{-4}$	
			$\underset{\sim}{\phi}^{26}$	$\gamma^*=1.000$	
5	26-35	$\underset{\sim}{\phi}^{26}=(1.70723,\ 3.20865)$	$\underset{\sim}{\phi}^{30}$	$\alpha^*=1.0\times10^{-6}$	
		$U'(\underset{\sim}{\phi}^{26})=0.47794$	$\underset{\sim}{\phi}^{35}$	$\gamma^*=\tau+1$	
6	35-64	$\underset{\sim}{\phi}^{35}=(1.70723,\ 3.20866)$	$\underset{\sim}{\phi}^{36}$	$\alpha^*=9.472\times10^{-4}$	
		$U'(\underset{\sim}{\phi}^{35})=0.47794$	$\underset{\sim}{\phi}^{64}$	$\gamma^*=1.096\times10^3$	

TABLE 3.1 (continued)

SUMMARY OF IMPORTANT STEPS IN THE EXAMPLE
ILLUSTRATING THE GRAZOR SEARCH STRATEGY

$\check{\phi} = (2.23605, 4.47210)$, $U'(\check{\phi}) = 0.42857$

Iteration Number	Points of Iteration	Starting Point of Iteration	Values of Scale Factors		k_r
			Point	Scale Factor	
7	64-72	$\phi^{64}=(2.05489, 4.18669)$ $U'(\phi^{64})=0.44084$	ϕ^{64}	$\gamma^*=\tau+2$	2
8	72-78	$\phi^{72}=(2.09028, 4.17411)$ $U'(\phi^{72})=0.43199$	ϕ^{78}	$\gamma^*=1.000$	2
9	78-96	$\phi^{78}=(2.09380, 4.17280)$ $U'(\phi^{78})=0.43146$	ϕ^{96}	$\gamma^*=60.69$	2
10	96-103	$\phi^{96}=(2.18832, 4.38018)$ $U'(\phi^{96})=0.42929$	ϕ^{98} ϕ^{103}	$\alpha^*=2.279\times10^{-3}$ $\gamma^*=1.000$	2
11	103-117	$\phi^{103}=(2.19040, 4.37924)$ $U'(\phi^{103})=0.42886$	ϕ^{117}	$\gamma^*=30.03$	2
12	117-126	$\phi^{117}=(2.22029, 4.44082)$ $U'(\phi^{117})=0.42864$	ϕ^{126}	$\gamma^*=10.47$	3
13	126-132	$\phi^{126}=(2.23088, 4.46221)$ $U'(\phi^{126})=0.42862$			1
13	133-136	$\phi^{133} = \phi^{126}$ $U'(\phi^{133}) = U'(\phi^{126})$	ϕ^{134} ϕ^{136}	$\alpha^*=2.279\times10^{-3}$ $\gamma^*=\tau+2$	2
18	169-176	$\phi^{169}=(2.23595, 4.47237)$ $U'(\phi^{169})=0.42861$	ϕ^{176} $*=\phi^{169}$	$\gamma^*=1.000$	3

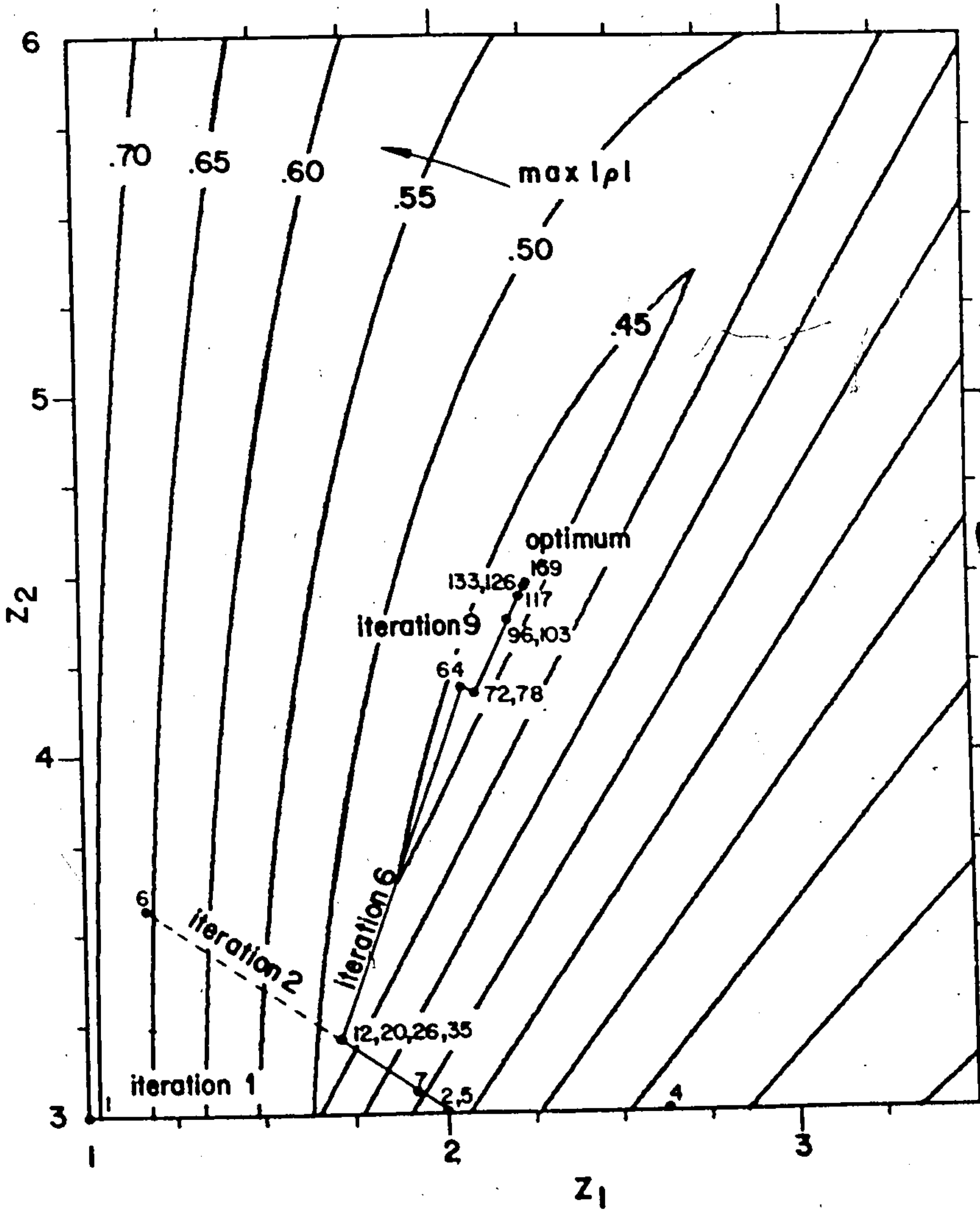


Fig. 3.5 Example illustrating how the grazor search strategy follows the narrow path of discontinuous derivatives.

of the network (see Fig. 3.6). Let

$$y_i(\underset{\sim}{\phi}) = \frac{1}{2}|\rho(\underset{\sim}{\phi},\psi_i)|^2 \tag{3.23}$$

and define

$$U'(\underset{\sim}{\phi}) = \max_i |\rho(\underset{\sim}{\phi},\psi_i)| \tag{3.24}$$

where $\underset{\sim}{\phi} = [Z_1 \;\; Z_2]^T$, and ρ is the reflection coefficient on 11 uniformly spaced frequencies ψ_i in the band 0.5-1.5 GHz.

The grazor search strategy starts at

$$\underset{\sim}{\phi}^1 = [1.0 \quad 3.0]^T$$

$$U'(\underset{\sim}{\phi}^1) = 0.70954$$

and the values of the parameters used are $\alpha_0 = 1$ (at start), $\check{\alpha} = 10^{-6}$, $\beta = 10$, $\eta = 0.5$, $\epsilon = 10^{-4}$ and $\epsilon' = 10^{-6}$.

The first iteration extends from $\underset{\sim}{\phi}^1$ to $\underset{\sim}{\phi}^5$; $\underset{\sim}{\phi}^2$ is the new point obtained when taking a unit step along the direction suggested by the negative gradient. Since $\underset{\sim}{\phi}^2$ is a satisfactory improvement, a golden section search is initiated, yielding $\underset{\sim}{\phi}^3(\gamma=1+\tau)$ which is not an improvement over $\underset{\sim}{\phi}^2$. The interval of search is thus found. $\underset{\sim}{\phi}^4(\gamma=\tau)$ is found to be no improvement over $\underset{\sim}{\phi}^2$. The golden section search is now terminated, since the current resolution between two interior points of search falls below the minimum allowed value. $\underset{\sim}{\phi}^5 = \underset{\sim}{\phi}^2$ is thus the best point attained at the end of iteration 1. At the end of iteration 5, $U(\underset{\sim}{\phi}^{26})-U(\underset{\sim}{\phi}^{35})<\epsilon$, so k_r is increased from 1 to 2 in the next

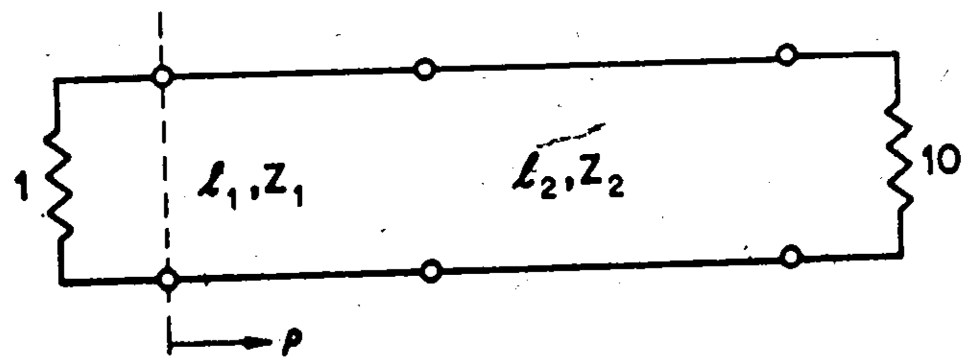


Fig. 3.6 2-section 10Ω to 1Ω quarter-wave transmission-line transformer.

iteration. For a similar reason, k_r is increased from 2 to 3 for iteration 12, and reset to 1 from 3 for iteration 13. During iteration 18, the parameter values remain the same to 5 significant digits, and the improvement in U at the end is less than ϵ'; all successive attempts to achieve a better point with an improvement greater than ϵ' (by considering 1, 2 and 3 ripples) fail, and the procedure is terminated.

3.3 Constrained Minimax Optimization

3.3.1 Statement of the Problem

The constrained minimax problem considered may be stated as follows.

Minimize

$$U(\underset{\sim}{\phi}) = \max_{i \in I} y_i(\underset{\sim}{\phi}) \tag{3.25}$$

subject to

$$g_j(\underset{\sim}{\phi}) \geq 0 \qquad j \in M \tag{3.26}$$

where

$$I \triangleq \{1,2,\ldots,n\} \tag{3.27}$$

$$M \triangleq \{1,2,\ldots,m\} \tag{3.28}$$

(see Sections 2.2 and 2.4)

It will be assumed that the functions y_i and g_j are continuous with continuous partial derivatives, and that the inequality constraints (3.26) are such that a Kuhn-Tucker solution exists (Lasdon 1970, Zangwill 1969).

Let $\hat{y}_\ell(\underset{\sim}{\phi})$ for $\ell \in L$ be the largest local discrete maxima (ripples) of $y_i(\underset{\sim}{\phi})$ for $i \in I$, in decreasing magnitude, where

$$L \triangleq \{1, 2, \ldots, n_r\} \tag{3.29}$$

3.3.2 Formulation 1

The constrained minimax problem of (3.25) and (3.26) can be formulated as a non-linear programming problem as follows.

$$\text{Minimize} \qquad \phi_{k+1} \tag{3.30}$$

subject to (3.26) and

$$\phi_{k+1} - y_i(\underset{\sim}{\phi}) \geq 0 \qquad i \in I \tag{3.31}$$

The above problem can then be reformulated as an unconstrained minimax problem as follows.

Minimize with respect to $\underset{\sim}{\phi}$ and ϕ_{k+1}

$$V(\underset{\sim}{\phi}, \phi_{k+1}, \underset{\sim}{\alpha}) = \max_{\substack{i \in I \\ j \in M}} \begin{bmatrix} \phi_{k+1}, \phi_{k+1} - \alpha_1(\phi_{k+1} - y_i(\underset{\sim}{\phi})), \\ \phi_{k+1} - \alpha_{j+1} g_j(\underset{\sim}{\phi}) \end{bmatrix} \tag{3.32}$$

where

$$\underset{\sim}{\alpha} \triangleq [\alpha_1 \; \alpha_2 \; \ldots \; \alpha_{m+1}]^T \tag{3.33}$$

$$\alpha_j > 0 \qquad j = 1, 2, \ldots, m+1 \tag{3.34}$$

For a large enough value of $\underset{\sim}{\alpha}$ one can obtain, in principle, the exact optimal solution for the original problem by minimizing this reformulated objective function.

When implementing this scheme one can, for the problem defined earlier, slightly modify the formulation in order to save on computational effort, so that the minimization function chosen is

$$V'(\underset{\sim}{\phi},\phi_{k+1},\underset{\sim}{\alpha}) = \max_{\substack{\ell\epsilon L\\ j\epsilon M}} \begin{bmatrix} \phi_{k+1}, \phi_{k+1} - \alpha_1(\phi_{k+1} - \hat{y}_\ell(\underset{\sim}{\phi})), \\ \\ \phi_{k+1} - \alpha_{j+1}\, g_j(\underset{\sim}{\phi}) \end{bmatrix} \tag{3.35}$$

3.3.3 Formulation 2

In this formulation, weighting functions are used to convert the original problem into an unconstrained minimax problem as follows.

Minimize with respect to $\underset{\sim}{\phi}$

$$W(\underset{\sim}{\phi},\underset{\sim}{w}) = \max_{\substack{i\epsilon I\\ j\epsilon M}} [y_i(\underset{\sim}{\phi}), -w_j g_j(\underset{\sim}{\phi})] \tag{3.36}$$

where

$$\underset{\sim}{w} \triangleq [w_1\ w_2\ \dots\ w_m]^T \tag{3.37}$$

$$w_j > 0 \qquad j\epsilon M \tag{3.38}$$

For purposes of practical implementation, as long as $U(\underset{\sim}{\phi}) > 0$ and one wishes to apply nonzero weights only to violated constraints of (3.26), the minimization function may be chosen as

$$W'(\underset{\sim}{\phi},\underset{\sim}{w}') = \max_{\substack{\ell\epsilon L\\ j\epsilon M}} [\hat{y}_\ell(\underset{\sim}{\phi}), -w_j'\, g_j(\underset{\sim}{\phi})] \tag{3.39}$$

where

$$\underset{\sim}{w}' \triangleq [w_1' \; w_2' \; \dots \; w_m']^T \tag{3.40}$$

$$\begin{aligned} w_j' &> 0 \text{ for } g_j(\underset{\sim}{\phi}) < 0 \\ w_j' &= 0 \text{ for } g_j(\underset{\sim}{\phi}) \geq 0 \end{aligned} \qquad j \in M \tag{3.41}$$

The advantage of this formulation is apparent when $U > 0$ implies that certain specifications are violated and $U < 0$ implies that they are satisfied. In this case, comparison with violated and satisfied constraints seems appropriate.

3.3.4 Comments

By proper choice of the elements of $\underset{\sim}{\alpha}$, $\underset{\sim}{w}$, or $\underset{\sim}{w}'$, the reformulated functions V, V', W or W' can be minimized by a suitable minimax or near-minimax algorithm. In case of parameter constraints, upper and lower specifications can be considered as follows.

$$\begin{aligned} g_{2i-1}(\underset{\sim}{\phi}) &= \phi_i - \phi_{i\ell} \geq 0 \\ g_{2i}(\underset{\sim}{\phi}) &= -(\phi_i - \phi_{iu}) \geq 0 \end{aligned} \qquad i = 1, 2, \dots, k \tag{3.42}$$

$$g_j(\underset{\sim}{\phi}) \geq 0 \qquad j = 2k+1,\ 2k+2,\ \ldots\ ,\ m \tag{3.43}$$

3.4 Practical Investigation of Minimax Optimality Conditions

3.4.1 Introduction

In recent paper (Bandler 1971), the conditions for a minimax optimum were derived for a general nonlinear minimax approximation problem from the Kuhn-Tucker (1950) conditions for a constrained optimum in nonlinear programming. See also Dem'yanov (1970), Medanic (1970). The minimax optimality conditions have also been derived from conditions for optimality in generalized least pth approximation problems for $p \to \infty$ by Bandler and Charalambous (1971, 1972b, 1973a).

3.4.2 Conditions for a Minimax Optimum

The minimax problem considered is the unconstrained version of the problem stated in Section 3.3.1 (i.e., when (3.26) is ignored). The necessary (Theorem 1) and sufficient (Theorem 2) conditions for a minimax optimum are stated as follows.

<u>Theorem 1</u>

At an optimum point $\underset{\sim}{\phi}^{o}$ for the minimax approximation problem there exist

$$u_\ell \geq 0 \qquad \ell = 1, 2, \ldots, k_r \tag{3.44}$$

such that

$$\sum_{\ell=1}^{k_r} u_\ell \nabla \hat{y}_\ell(\phi^o) = 0 \tag{3.45}$$

$$\sum_{\ell=1}^{k_r} u_\ell = 1 \tag{3.46}$$

where $\hat{y}_\ell(\phi^o)$ for $\ell = 1, 2, \ldots, k_r$ are the equal maxima.

Theorem 2

If the relations in Theorem 1 are satisfied at a point ϕ^o and all the functions $y_i(\phi)$ for $i \in I$ are convex, then ϕ^o is optimal.

Theorems 1 and 2 have been proved by Bandler (1971), and the optimality conditions as derived by Curtis and Powell (1966) follow immediately from these theorems.

3.4.3 Practical Implementation

Once a proposed or a design solution is obtained for a minimax problem, it may be necessary to investigate the necessary optimality conditions. If the point ϕ, corresponding to a solution, is to be tested for optimality, an attempt is made to solve

$$\sum_{\ell=1}^{k_r} u_\ell \nabla \hat{y}_\ell(\phi) = 0 \tag{3.47}$$

plus (3.44) and (3.46) for $k_r = 1, 2, \ldots$ until for a value of k_r^* ($\le n_r$), (3.44), (3.46) and (3.47) are satisfied. If this is not possible, the necessary conditions are not satisfied.

A computer program has been developed which can test a solution for the necessary conditions for a minimax optimum by two formulations. One uses a linear programming approach, and the other the solution of a set of linear independent equations. See Appendix B, Bandler and Srinivasan (1973c, 1973d).

3.4.4 Method 1

(3.44), (3.46) and (3.47) are solved here by minimizing

$$u_{k_r+1} \ge 0 \tag{3.48}$$

such that (3.44), (3.46) are satisfied and

$$\left| \sum_{\ell=1}^{k_r} u_\ell \frac{\partial \hat{y}_\ell}{\partial \phi_i} \right| \le u_{k_r+1} \qquad i = 1, 2, \ldots, k \tag{3.49}$$

Linear programming ensures that

$$u_\ell \ge 0 \qquad \ell = 1, 2, \ldots, k_r+1 \tag{3.50}$$

3.4.5 Method 2

Here, we solve a set of linearly independent equations

$$\sum_{\ell=1}^{k_r} u_\ell \frac{\partial \hat{y}_\ell}{\partial \phi_i} = 0 \qquad i \in K' \qquad (3.51)$$

and (3.46), where K' is a suitable subset of {1, 2, ..., k}.

There is no guarantee, however, that (3.44) will hold. When k_r-1 is greater than the number of elements of K', the system of equations (3.46) and (3.51) have more unknowns than equations, and we use Method 1 to get the u_ℓ.

3.4.6 Comments

Appendix B contains a program description incorporating the ideas of the previous two sections. The program package can be called from the user's main program and either of the two, or both the methods can be used to test the optimality conditions. The user can either specify the value of k_r or a tolerance ξ relative to $\hat{y}_1$ within which some of the $\hat{y}_2, \ldots, \hat{y}_{n_r}$ lie. The necessary conditions for optimality are satisfied when the norm $||\underset{\sim}{r}||$ of the residual vector

$$\underset{\sim}{r} \triangleq \sum_{\ell=1}^{m_r} u_\ell \nabla \hat{y}_\ell \qquad (3.52)$$

falls within a user-specified value ε, and (3.44), (3.46) hold, for a value of m_r starting with 1. If the conditions are not satisfied for m_r=1, m_r is incremented by 1 and the procedure is repeated. The investigation ends as soon as the conditions are satisfied for a value of $m_r \leq k_r$, or the conditions are not satisfied for $m_r = 1, 2, \ldots, k_r$. The user-specified definitions of $||\cdot||$ and the value of ϵ should be realistic so that the program may give meaningful results.

The importance of this investigation cannot be underestimated especially when there may be a number of solutions obtained by the same, or different optimization methods for a given problem and one wishes to test these solutions for optimality so as to be able to detect local optima, and to compare the methods for convergence towards the optima. This program may be used in such a way that it is possible to investigate the solutions after a certain number of iterations of the algorithm, or when a certain convergence criterion is reached, so that one may decide whether to carry on with further optimization, or to terminate altogether.

The program also makes it possible to find the maxima which are active in the vicinity of the optimum, so that the user may gain insight into the various scaling factors associated with the problem.

3.4.7 Example

The problem chosen was the lower-order modelling of a ninth-order nuclear reactor system when the operating reactor power level is in the 90-100 percent range of the full power (Bereznai 1971). A second-order model was chosen and the step-response of the system was approximated by that of the model for a minimax objective over a time-interval of 0-10 seconds. A solution was obtained for this problem and the program described in Appendix B was used to test the solution for optimality.

The relevant input parameters are: $k = 2$, $n_r = 4$, $\epsilon = 10^{-6}$, $\xi = 0.01$, and the norm chosen is given by :

$$\|\underset{\sim}{r}\| = \max_{1 \le i \le k} |r_i|$$

$\underset{\sim}{\nabla} \underset{\sim}{\hat{y}}$ is given by

$$\underset{\sim}{\nabla}\hat{y}_1 = \begin{bmatrix} .38711013 \times 10^{-3} \\ -.14208087 \times 10^{-3} \end{bmatrix}, \underset{\sim}{\nabla}\hat{y}_2 = \begin{bmatrix} -.29632883 \times 10^{-1} \\ .10876118 \times 10^{-1} \end{bmatrix}$$

$$\underset{\sim}{\nabla}\hat{y}_3 = \begin{bmatrix} .79840875 \times 10^{-3} \\ .68487328 \times 10^{-2} \end{bmatrix}, \underset{\sim}{\nabla}\hat{y}_4 = \begin{bmatrix} .17968278 \times 10^{-2} \\ -.14014776 \times 10^{-3} \end{bmatrix}$$

and $\underset{\sim}{\hat{y}}$ is given by

$$\hat{y}_1 = .29234162 \times 10^{-2}, \quad \hat{y}_2 = .29234034 \times 10^{-2}$$

$$\hat{y}_3 = .23141899 \times 10^{-2}, \quad \hat{y}_4 = .62431057 \times 10^{-3}$$

Corresponding to $\xi = 0.01$, the value of k_r is equal to 2. Both the methods were used to test the solution for optimality, and the results obtained are shown below.

(i) $m_r = 1$

Both the methods give the same result as there is only one function under consideration.

$u_1 = 1$

$$\underset{\sim}{r} = [0.38711013 \times 10^{-3} \quad -.14208087 \times 10^{-3}]^T$$

$$||\underset{\sim}{r}|| = 0.38711013 \times 10^{-3}$$

(3.44) and (3.46) are satisfied, while $||\underset{\sim}{r}||$ is not less than ε. Thus the conditions are not satisfied for $m_r = 1$.

(ii) $m_r = 2$

<u>Method 1</u>

$$\underset{\sim}{u} = [0.98710491 \quad 0.12895086 \times 10^{-1}]^T$$

$$\underset{\sim}{r} = [-0.25789922 \times 10^{-9} \quad 0.25789922 \times 10^{-9}]^T$$

$$||\underset{\sim}{r}|| = 0.25789922 \times 10^{-9}$$

<u>Method 2</u>

$$\underset{\sim}{u} = [0.98710492 \quad 0.12895077 \times 10^{-1}]^T$$

$$\underset{\sim}{r} = [0. \qquad -0.35255563 \times 10^{-9}]^T$$

$$||\underset{\sim}{r}|| = 0.35255563 \times 10^{-9}$$

(3.44)and (3.46) are satisfied and $||\underset{\sim}{r}||<\varepsilon$ for both the methods. The necessary optimality conditions are thus satisfied for $m_r = 2$. It is also observed that due to the type of formulation of the problem in Method 1, the elements of $\underset{\sim}{r}$ have equal magnitude.

3.5 Conclusions

A new minimax algorithm called grazor search has been proposed. Conditions which guarantee the convergence of the algorithm have also been stated. The spectrum of problems that can be accomodated has been extended to include constrained minimax objectives, and any efficient unconstrained minimax method can suitably be used for this purpose. The practical investigation of a solution for necessary optimality conditions has been implemented on the computer, so that it is now possible to check solutions at any stage of the optimization process. The subject matter of this chapter makes it possible to tackle unconstrained and constrained minimax problems by a new gradient algorithm, and to test intermediate or final solutions for optimality, on line.

CHAPTER IV

COMPUTER-AIDED CIRCUIT DESIGN

4.1 Introduction

This chapter primarily concentrates on applying the ideas presented in Chapter III to computer-aided design of electrical networks. Minimax designs are of special interest to the designer mainly because they attempt to achieve an equiripple behaviour of the response error function, which is useful in many cases. The problems considered include the design of LC transformers and cascaded transmission-line transformers and filters. Appropriate constraints have been incorporated whenever necessary, and the grazor search algorithm has been compared with the Osborne and Watson method and razor search strategy for reliability and efficiency (See Bandler, Srinivasan and Charalambous 1972, Bandler and Srinivasan 1973a). Unless otherwise mentioned, the objective function to be minimized is chosen as (2.13).

4.2 Lumped LC Transformer

The problem considered (Hatley 1967) is the design of a 3-section lumped-element LC transformer to match a 1Ω load to a 3Ω generator over the angular frequency range of 0.5 - 1.179 radians/sec. Fig. 4.1 shows the structure of the network, and the objective is to minimize

$$U(\underset{\sim}{\phi}) = \max_i |\rho_i(\underset{\sim}{\phi})| \tag{4.1}$$

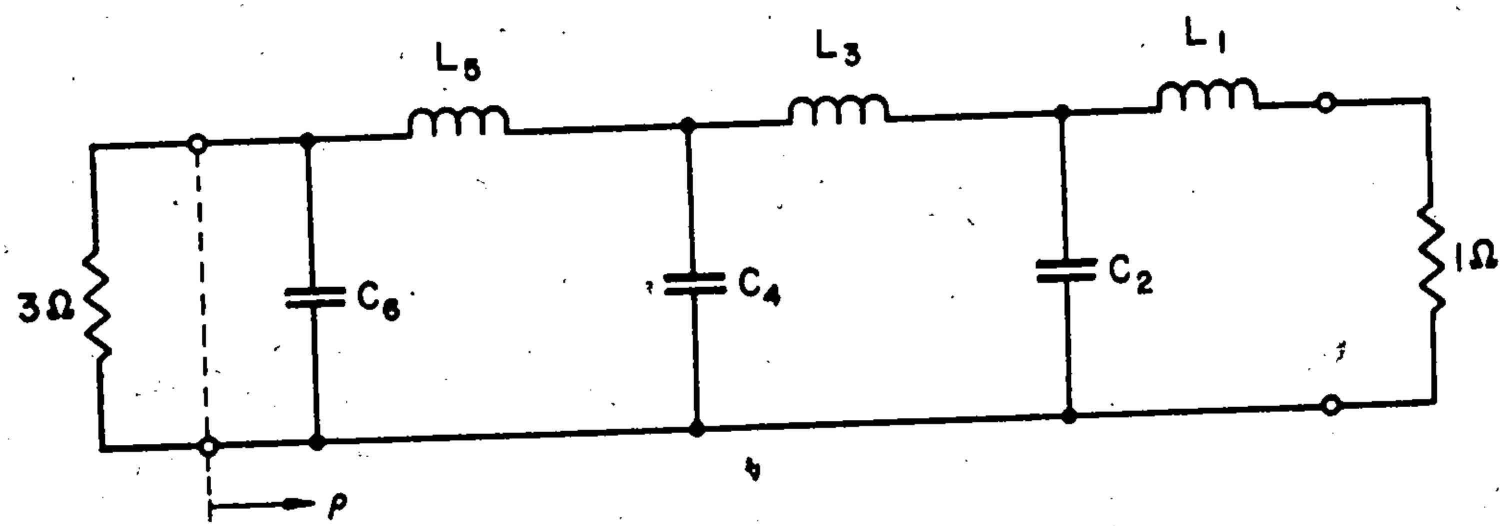


Fig. 4.1 3-section LC transformer problem. Optimum matching over a frequency range of 0.5-1.179 radians/sec occurs at the following parameter values: L_1=1.04088, C_2=0.979035, L_3=2.34044, C_4=0.780157, L_5=2.93714, C_6=0.346960 and $\check{U}=\max_i |\rho(\check{\phi},\psi_i)|$ =0.075820.

where $\rho_i(\underset{\sim}{\phi}) = \rho(\underset{\sim}{\phi}, \psi_i)$ is the reflection coefficient over 21 uniformly spaced frequencies ψ_i in the passband, and

$$\underset{\sim}{\phi} = [L_1\ C_2\ L_3\ C_4\ L_5\ C_6]^T \tag{4.2}$$

The six parameters were optimized by the grazor search strategy and the Osborne and Watson method, and Fig. 4.2 shows a typical graph of objective function against function evaluations for the two methods for identical starting points. As can be seen from the graph, the Osborne and Watson method fails to reach the vicinity of the optimum, while the grazor search algorithm achieves an optimal solution. Table 4.1 shows the number of function evaluations needed to get within 0.01 percent of the optimum for different values of η, the factor of resolution between two interior points of the golden section for the grazor search, and it is clear that the value of η chosen need not be very small.

4.3 Quarter-Wave Transmission-Line Transformer

The problem considered is the design of 2-section and 3-section 10Ω to 1Ω transmission-line transformers over a 100 percent relative bandwidth centred at 1 GHz (Matthaei, Young and Jones 1964, Bandler and Macdonald 1969a, 1969b). The objective is to minimize $\max_i |\rho(\underset{\sim}{\phi}, \psi_i)|$ on 11 frequencies ψ_i in the band 0.5-1.5 GHz for the network shown in Fig. 4.3, where ρ_i is the reflection coefficient of the network at ψ_i.

The grazor search method and the Osborne and Watson algorithm were used for minimax optimization. For both the methods, the objective

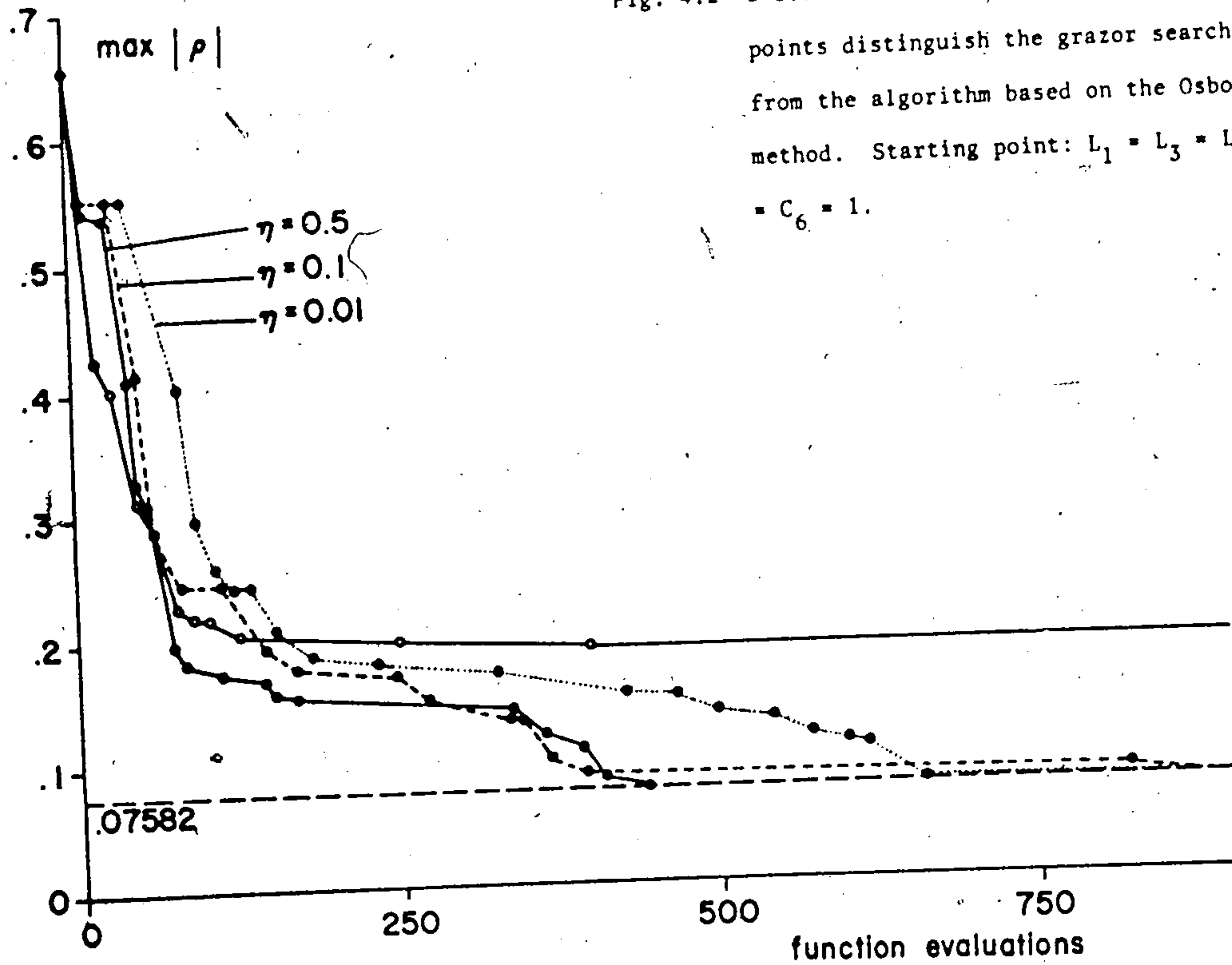


Fig. 4.2 3-section LC transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. Starting point: $L_1 = L_3 = L_5 = C_2 = C_4 = C_6 = 1$.

TABLE 4.1

COMPARISON OF THE NUMBER OF FUNCTION EVALUATIONS REQUIRED BY THE GRAZOR SEARCH METHOD TO REACH WITHIN 0.01 PERCENT OF THE OPTIMUM FOR DIFFERENT VALUES OF η FOR IDENTICAL STARTING POINTS

$$L_1 = L_3 = L_5 = C_2 = C_4 = C_6 = 1$$

Function Evaluations	η
1316	0.01
880	0.10
561	0.50

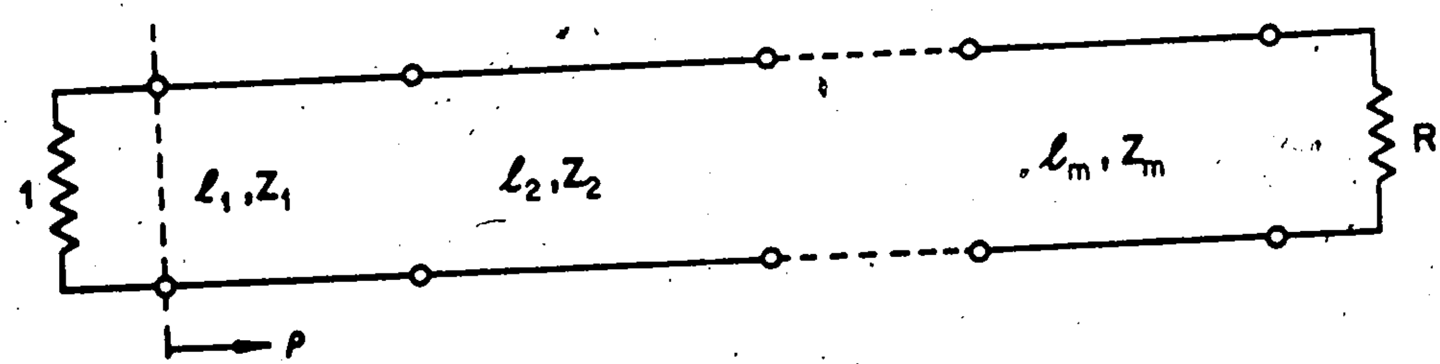


Fig. 4.3 The m-section resistively terminated cascade of transmission lines. Optimum matching over 100 percent band centred at 1 GHz for R=10 occurs for the following parameter values.

2-section: $\ell_1 = \ell_2 = \ell_q$, $Z_1 = 2.23605$, $Z_2 = 4.4721$

3-section: $\ell_1 = \ell_2 = \ell_3 = \ell_q$, $Z_1 = 1.63471$, $Z_2 = 3.16228$,

$Z_3 = 6.11729$

$\ell_q = 7.49481$ cm is the quarter-wavelength at centre frequency.

function is given by (2.13) where

$$y_i(\underset{\sim}{\phi}) = \frac{1}{2}|\rho_i(\underset{\sim}{\phi})|^2 \tag{4.3}$$

In the 2-section examples, the 11 frequencies were uniformly spaced. In the 3-section examples, the frequencies were 0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.30, 1.40, and 1.50 GHz. The progress of the algorithms from identical starting points with respect to the number of function evaluations (one corresponding to 11 evaluations of ρ) is recorded in Figs. 4.4 and 4.5. The points shown mark the successful end of a linear search or the beginning of linear programming.

A comparison was made between the grazor search, Osborne and Watson, and razor search methods, as shown in Tables 4.2 and 4.3. From Table 4.2, it is clear that the grazor search algorithm is, in general, faster than the razor search technique for the 2-section case when the lengths are kept fixed and the impedances are varied. From Table 4.3, it is clear that the grazor search algorithm is the best. The Osborne and Watson algorithm, though fairly fast initially, may in some cases fail or slow down near the optimum.

The grazor search method and the Osborne and Watson algorithm were further compared on the 3-section transformer problem when the lengths were fixed at quarter-wavelength values and the impedances were varied. For a starting point of $Z_1 = 3.16228$, $Z_2 = 1.0$ and $Z_3 = 10.0$, the former took 184 and 218 function evaluations, while the latter consumed 151 and 219 function evaluations to reach within 0.01 and 0.001 percent of the optimum value of the maximum reflection coefficient,

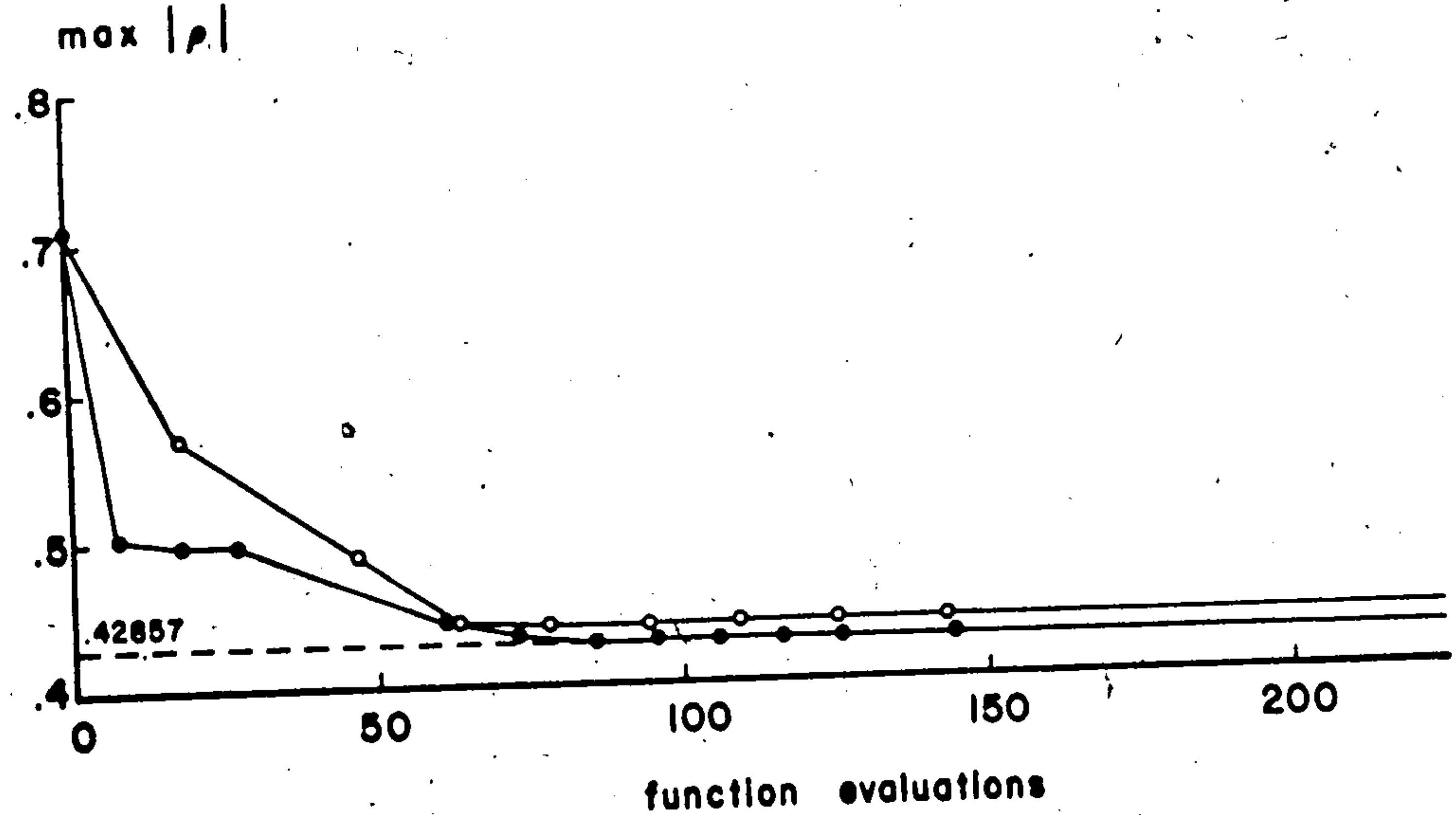


Fig. 4.4(a) The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_1, ℓ_2 fixed at ℓ_q and impedances varied. Starting point Z_1=1.0, Z_2=3.0.

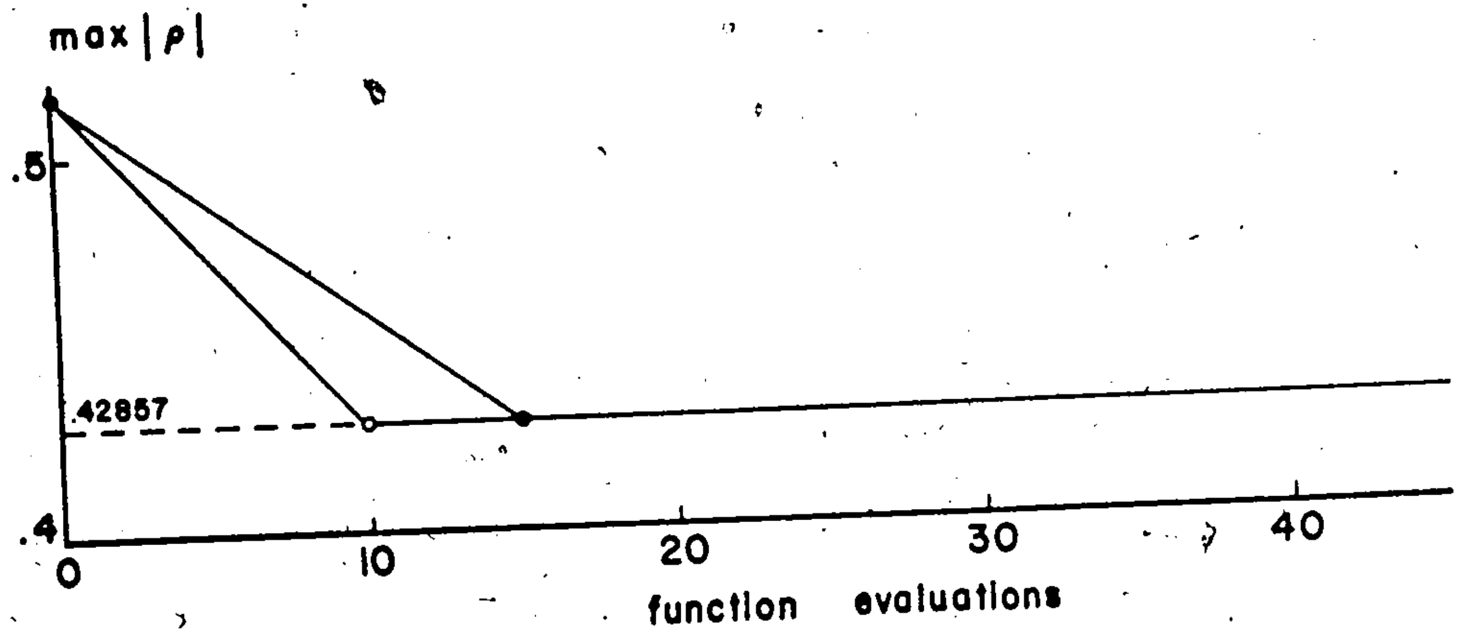


Fig. 4.4(b) The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. Z_1, Z_2 fixed at optimum values and lengths varied. Starting point $\ell_1/\ell_q=0.8$, $\ell_2/\ell_q=1.2$.

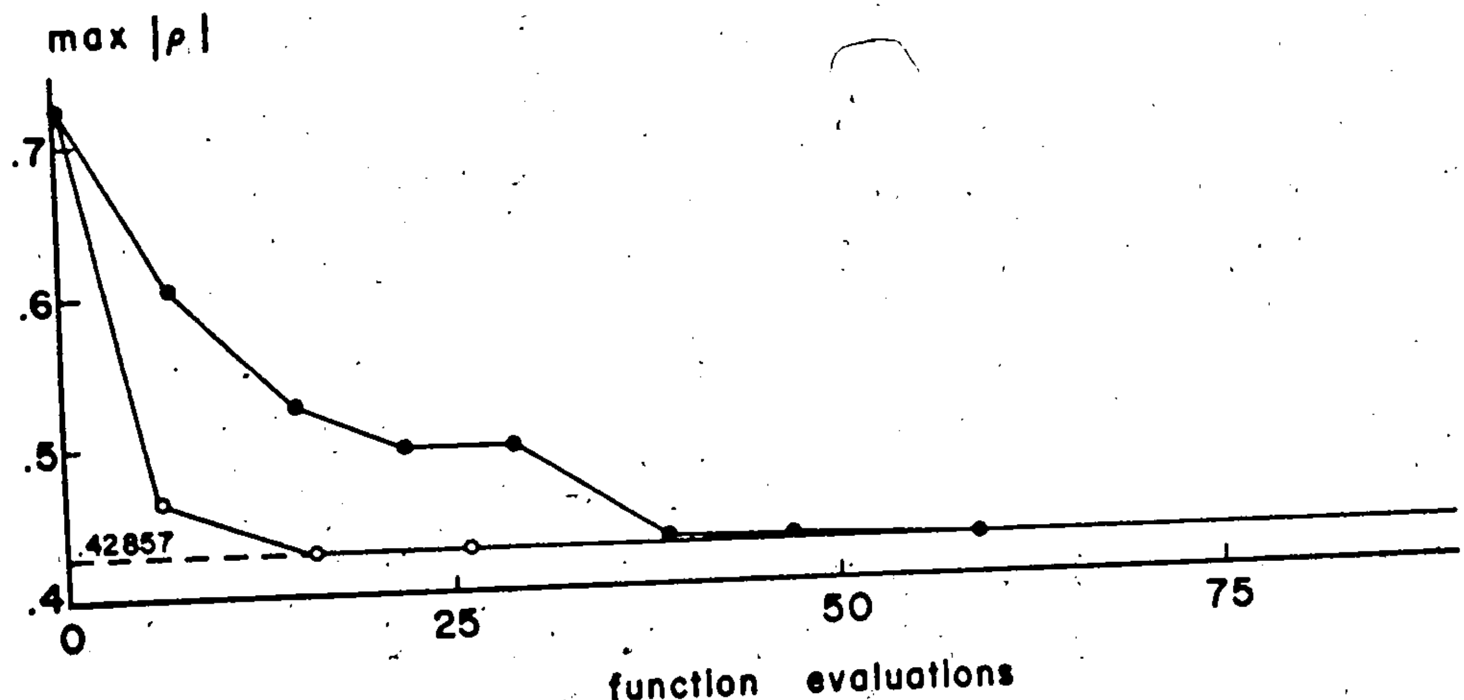


Fig. 4.4(c) The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_2, Z_2 fixed at optimum values and ℓ_1, Z_1 varied. Starting point $\ell_1/\ell_q=1.2$, $Z_1=3.5$.

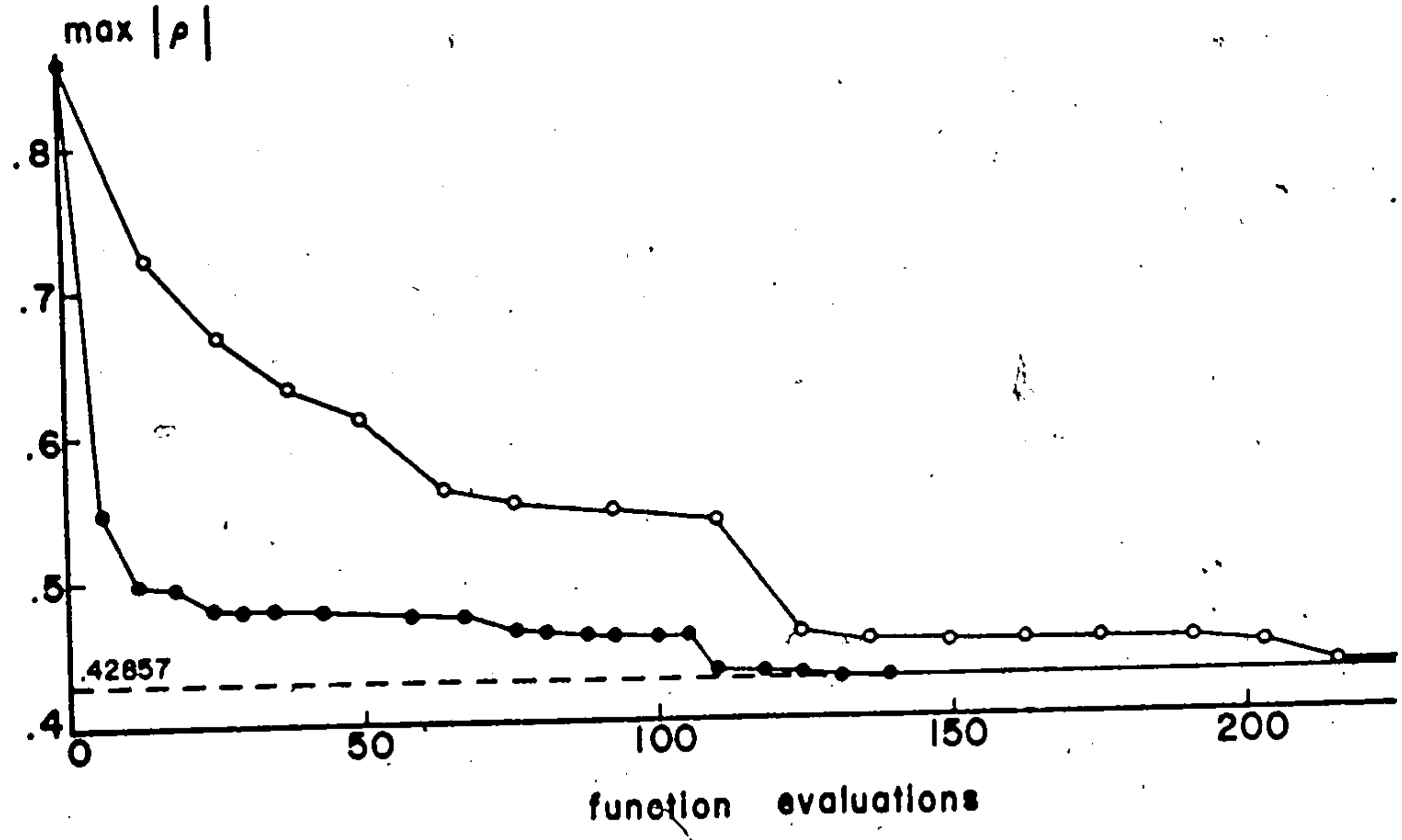


Fig. 4.4(d) The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 4 parameters varied. Starting point $\ell_1/\ell_q=1.2$, $\ell_2/\ell_q=0.8$, $Z_1=3.5$, $Z_2=3.0$.

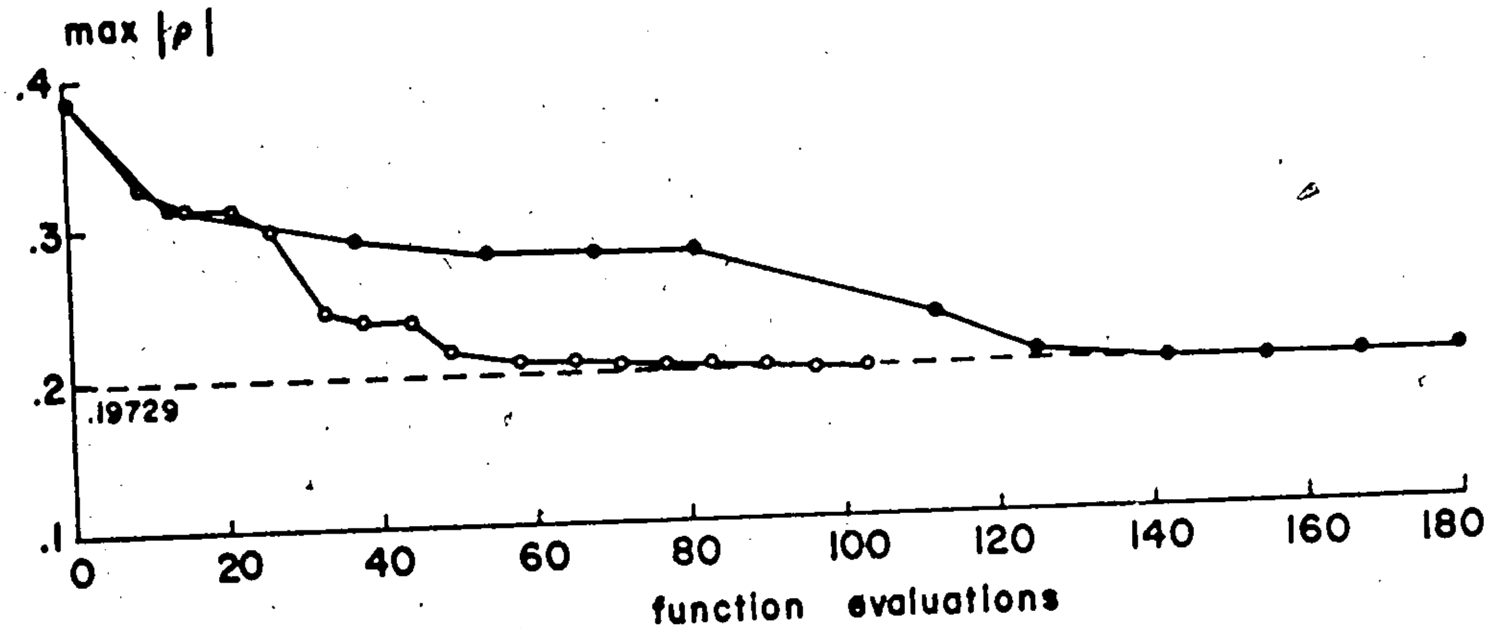


Fig. 4.5(a) The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. ℓ_1, ℓ_2, ℓ_3 fixed at ℓ_q and impedances varied. Starting point $Z_1=1.0$, $Z_2=3.16228$, $Z_3=10.0$.

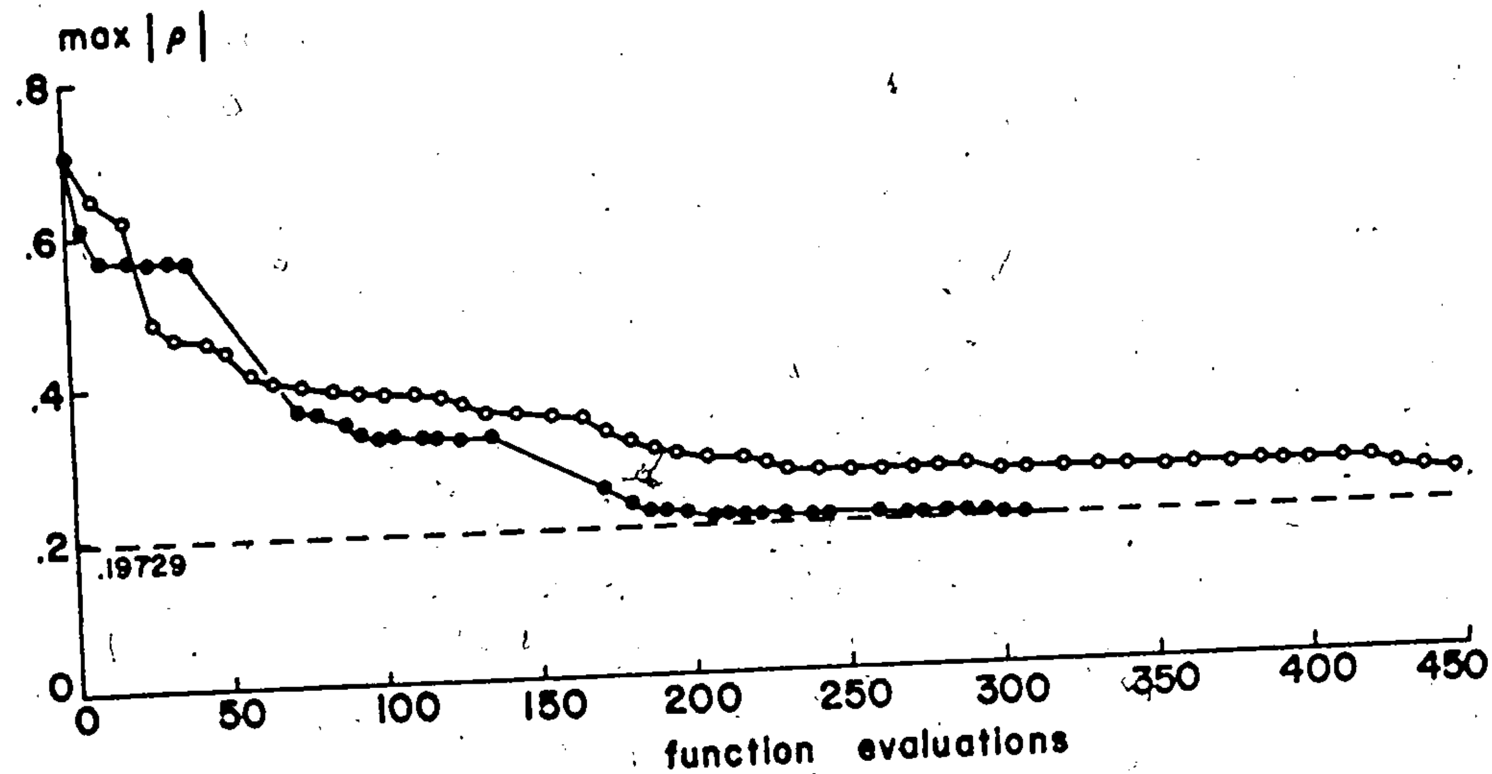


Fig. 4.5(b) The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point $\ell_1/\ell_q=0.8$, $\ell_2/\ell_q=1.2$, $\ell_3/\ell_q=0.8$, $Z_1=1.5$, $Z_2=3.0$, $Z_3=6.0$.

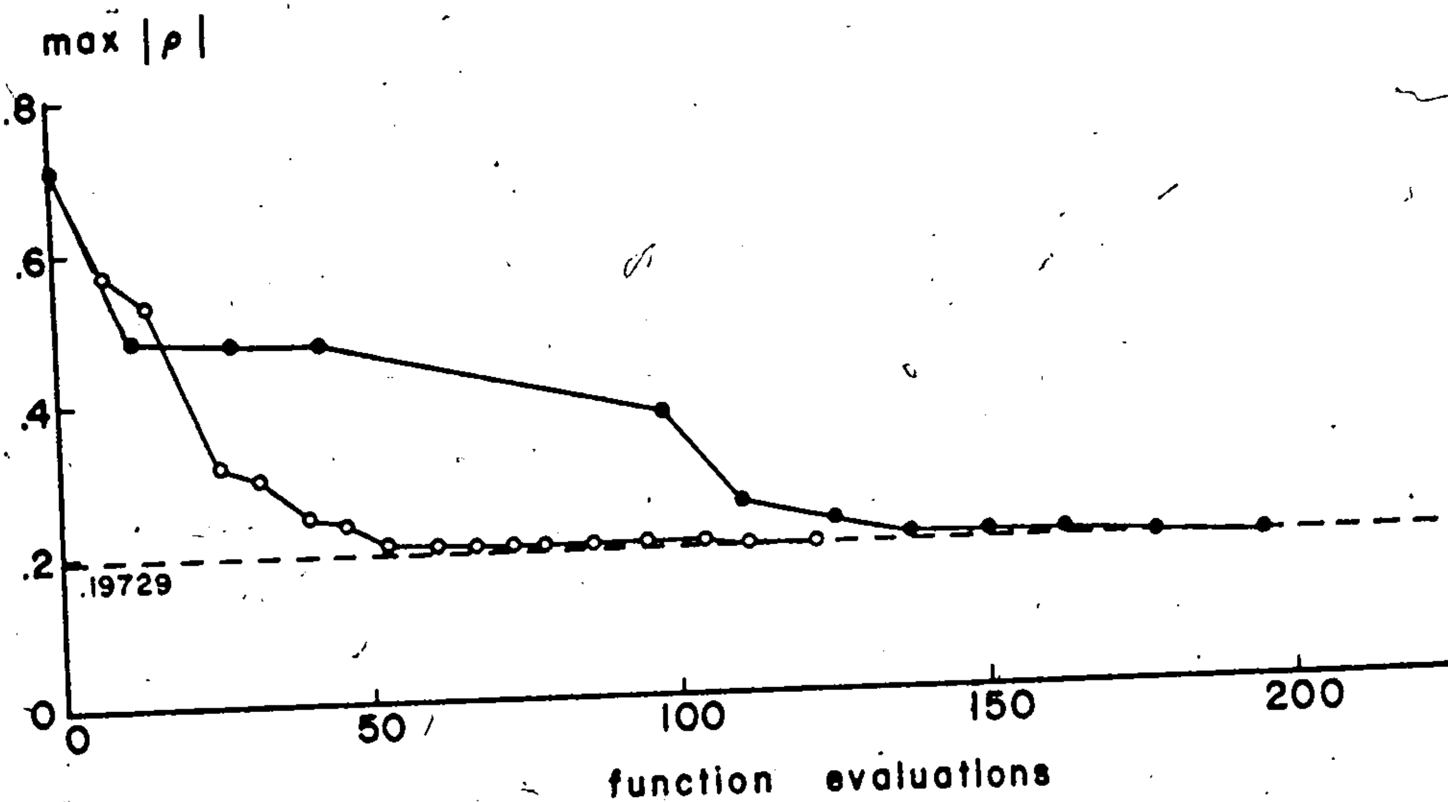


Fig. 4.5(c) The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point $\ell_1/\ell_q = \ell_2/\ell_q = \ell_3/\ell_q = 1.0$, $Z_1=1.0$, $Z_2=3.16228$, $Z_3=10.0$.

TABLE 4.2

OPTIMIZATION OF A 2-SECTION 10Ω TO 1Ω

TRANSMISSION-LINE TRANSFORMER OVER 100 PERCENT RELATIVE BANDWIDTH

Starting Point		Function Evaluations†	
Z_1	Z_2	Razor Search	Grazor Search
1.0	3.0	157	126
		207	
1.0	6.0	34	83
		152	
3.5	6.0	223	52
		100	
3.5	3.0	210	29
		163	

† Number of function evaluations required to bring the reflection coefficient within 0.01 percent of its optimum value.

TABLE 4.3

OPTIMIZATION OF A 3-SECTION 10Ω TO 1Ω TRANSMISSION-LINE TRANSFORMER OVER 100 PERCENT RELATIVE BANDWIDTH

	Fixed Lengths		Variable Lengths and Impedances			
Parameters ϕ_i	Starting Point	Maximum Reflection Coefficient at Start	Starting Point	Maximum Reflection Coefficient at Start	Starting Point	Maximum Reflection Coefficient at Start
ℓ_1/ℓ_q	1.0		1.0		0.8	
Z_1	1.0		1.0		1.5	
ℓ_2/ℓ_q	1.0		1.0		1.2	
Z_2	3.16228	0.70930	3.16228	0.70930	3.0	0.38865
ℓ_3/ℓ_q	1.0		1.0		0.8	
Z_3	10.0		10.0		6.0	
Razor Search Algorithm	Final Maximum Reflection Coefficient	0.19729		0.19733		0.19731
	Number of Function Evaluations	406		1300		1250
Grazor Search Algorithm	Final Maximum Reflection Coefficient	0.19729		0.19729		0.19729
	Number of Function Evaluations	219		696		498

TABLE 4.3 (continued)

OPTIMIZATION OF A 3-SECTION 10Ω TO 1Ω TRANSMISSION-LINE

TRANSFORMER OVER 100 PERCENT RELATIVE BANDWIDTH

		Fixed Lengths	Variable Lengths and Impedances	
Algorithm due to Osborne and Watson (1969)	Final Maximum Reflection Coefficient	0.19729	0.20831	0.19788
	Number of Function Evaluations	199	860	237

respectively. This case illustrates how the two algorithms compare when both methods work efficiently.

4.4 Cascaded Transmission-Line Filters

In this section, the grazor search algorithm is used to achieve the minimax design of cascaded transmission-line filters with desired attenuation characteristics. Three examples are chosen, and the ideas presented in Chapter III are applied to the problems.

4.4.1 Problem 1

The design of a 7-section cascaded transmission-line filter with frequency-dependent terminations is considered here (see Fig. 4.6). This problem has been considered by Carlin and Gupta (1969). The frequency variation of the terminations is like that of rectangular waveguides operating in the H_{10} mode with cutoff frequency 2.077 GHz. All section lengths were kept fixed at 1.5cm so that the maximum stopband insertion loss would occur at about 5 GHz. The passband 2.16 to 3 GHz was selected, for which a maximum passband insertion loss of 0.4dB was specified.

Fig. 4.7 shows the response of Carlin and Gupta which was used as an initial design. The other responses in Fig. 4.7 are a least 10th optimum obtained by Bandler and Seviora (1970) and a minimax optimum obtained by the grazor search strategy. In both cases only the passband was optimized. The minimax response has a maximum passband insertion

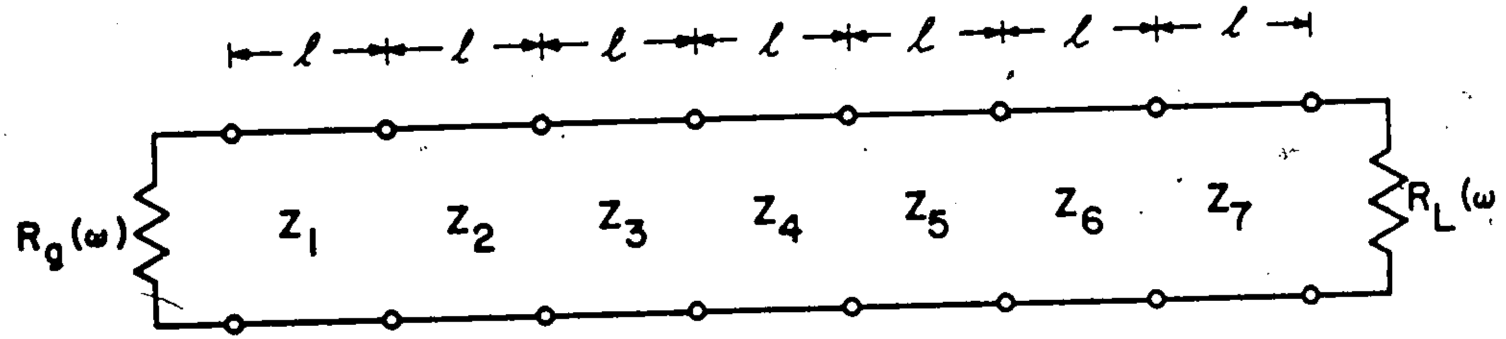
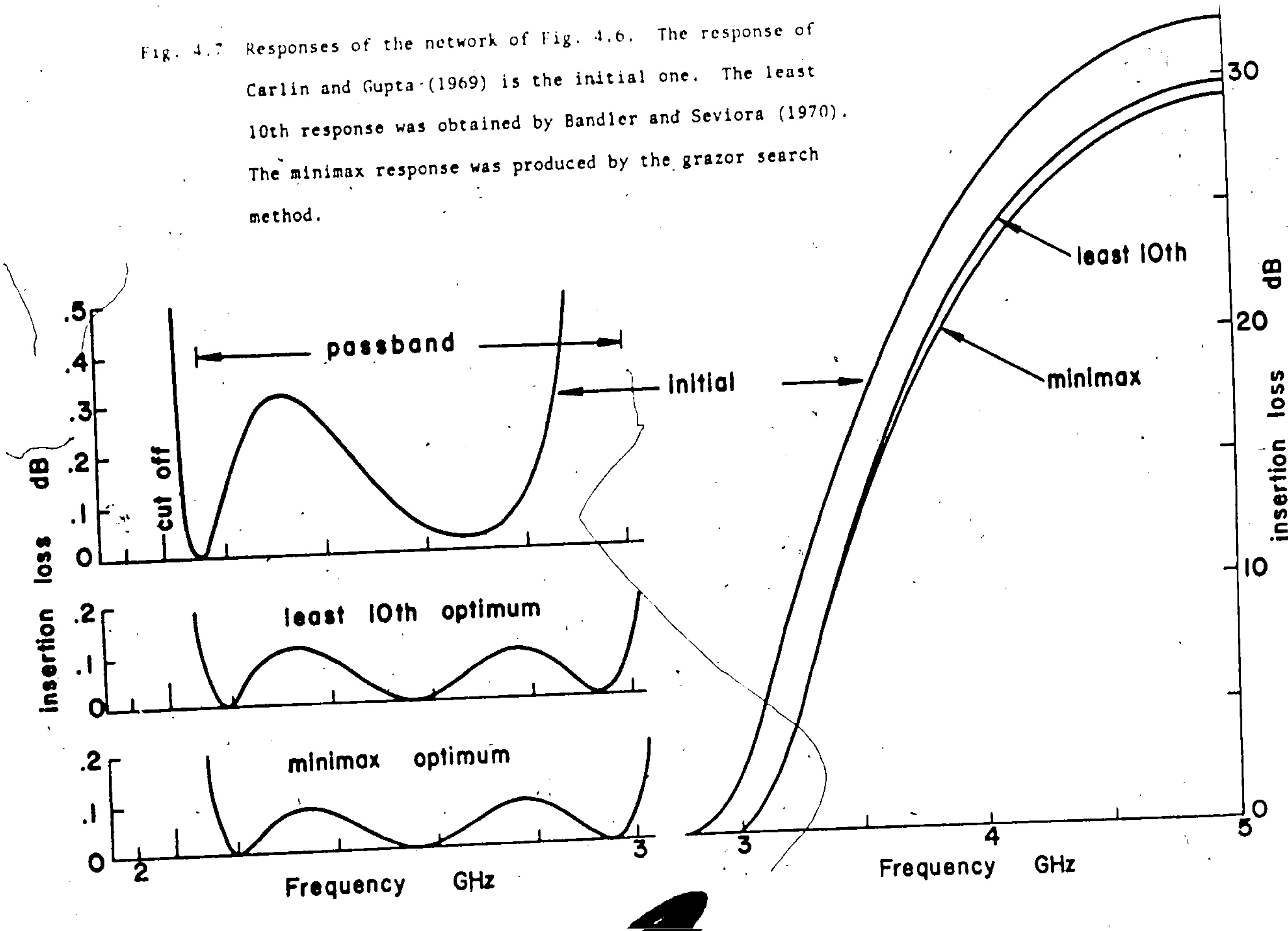


Fig. 4.6 Problem 1. Cascaded transmission-line filter operating between $R_g(\omega) = R_L(\omega) = 377/\sqrt{1-(f_c/f)^2}$, where $f_c = 2.077$ GHz and $\ell = 1.5$ cm.

Fig. 4.7 Responses of the network of Fig. 4.6. The response of Carlin and Gupta (1969) is the initial one. The least 10th response was obtained by Bandler and Seviora (1970). The minimax response was produced by the grazor search method.



loss of 0.086 dB. Table 4.4 gives the appropriate parameter values.

Fig. 4.8 shows the results of applying the grazor search method to optimize the sections in a filtering sense. Thus, it was desired to meet the 0.4 dB passband insertion loss while maximizing the stopband insertion loss at a single frequency (5GHz). Let

$$y_i(\underset{\sim}{\phi}) = \begin{cases} \frac{1}{2}(|\rho_i(\underset{\sim}{\phi})|^2 - r^2) & \text{in the passband} \\ \frac{1}{2}(\Psi - |\rho_i(\underset{\sim}{\phi})|^2) & \text{in the stopband} \end{cases} \tag{4.4}$$

where

$$\underset{\sim}{\phi} = [Z_1 \; Z_2 \; \dots \; Z_7]^T \tag{4.5}$$

and r is the reflection coefficient magnitude corresponding to an insertion loss of 0.4 dB. Here 22 uniformly-spaced points were selected from the passband. Table 4.4 gives the resulting parameter values. A similar response was attained by the grazor search technique when the section impedances were assumed symmetrical i.e., $Z_5=Z_3$, $Z_6=Z_2$, $Z_7=Z_1$.

4.4.2 Problem 2

The problem chosen consists of a 5-section cascaded transmission-line low-pass filter design and has been previously considered by Brancher, Maffioli and Premoli (1970). The filter structure is the same as in Fig. 4.3 for R=1. The terminating impedances are real and normalised to be 1Ω. It is required to have a passband insertion loss of less than 0.01 dB from 0 to 1 GHz and as high a stopband insertion loss as possible at 5 GHz. Twenty-one uniformly spaced points were chosen in the passband and one point in the stopband (5 GHz). The length of each section is

TABLE 4.4

COMPARISON OF PARAMETER VALUES FOR THE 7-SECTION FILTER (PROBLEM 1)

Characteristic Impedances (Normalized)	Carlin and Gupta (1969)	Minimax Design (Fig. 4.7)	Minimax Design (Fig. 4.8)
Z_1	1476.5	1305.2	3069.4
Z_2	733.6	607.8	2856.4
Z_3	1963.6	1323.3	25871.2
Z_4	461.8	362.7	10573.3
Z_5	1963.6	1323.2	25874.0
Z_6	733.6	607.9	2856.7
Z_7	1476.5	1305.2	3069.8

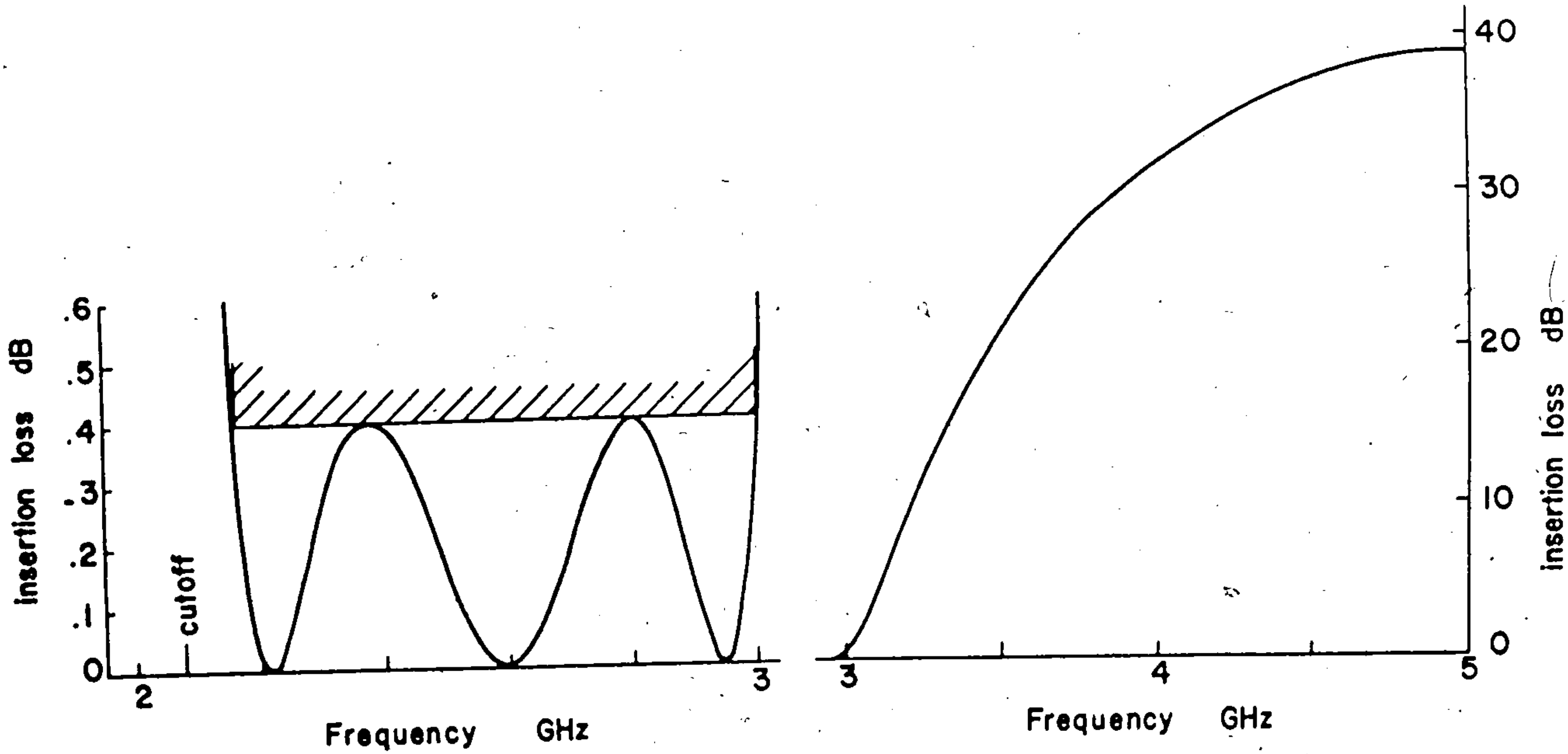


Fig. 4.8 Response of the minimax design of the network of Fig. 4.6 with 0.4 dB passband insertion loss produced by the grazor search method.

normalized with respect to ℓ_q = 1.49896 cm, the quarter-wavelength at 5 GHz.

The $y_i(\underset{\sim}{\phi})$ are given by (4.4) where r is the reflection coefficient magnitude corresponding to an insertion loss of 0.01dB, and

$$\underset{\sim}{\phi} = [\ell_{n1}\ Z_1\ \ell_{n2}\ Z_2\ \ell_{n3}\ Z_3\ \ell_{n4}\ Z_4\ \ell_{n5}\ Z_5]^T \tag{4.6}$$

$$\ell_{ni} = \ell_i/\ell_q \qquad i = 1,2,\ldots,5 \tag{4.7}$$

The lengths were initially fixed at ℓ_q, and the impedances varied. Levy (1965) has derived an optimal solution to this problem analytically. The grazor search method was used on this problem for minimax optimization, and the result obtained was identical to the one derived by Levy and Fig. 4.9 shows the optimal response obtained.

Brancher, Maffioli and Premoli (1970) have achieved some results for the problem, and an observation of their responses leads one to suspect that the results are not optimal. The grazor search method was used to test whether an improvement on the results of Brancher, Maffioli and Premoli was possible, and improved results were obtained.

Fig. 4.10 and Table 4.5 show the results for the problem where the impedances are fixed at some practical values and only the lengths are allowed to vary. As the final values obtained by the grazor search method indicate, the response at finish represents a good improvement over the response at start, both from passband and stopband considerations.

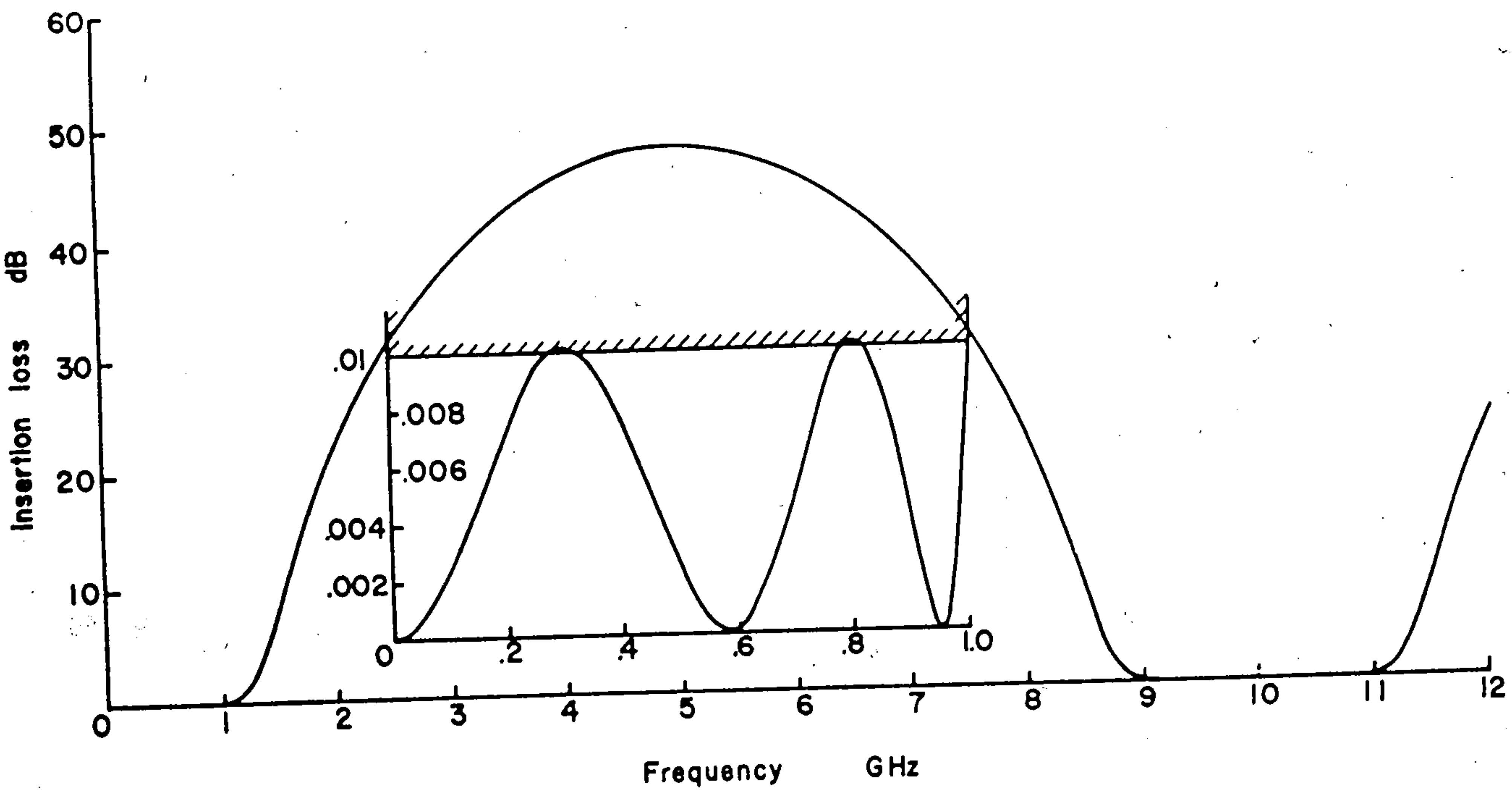
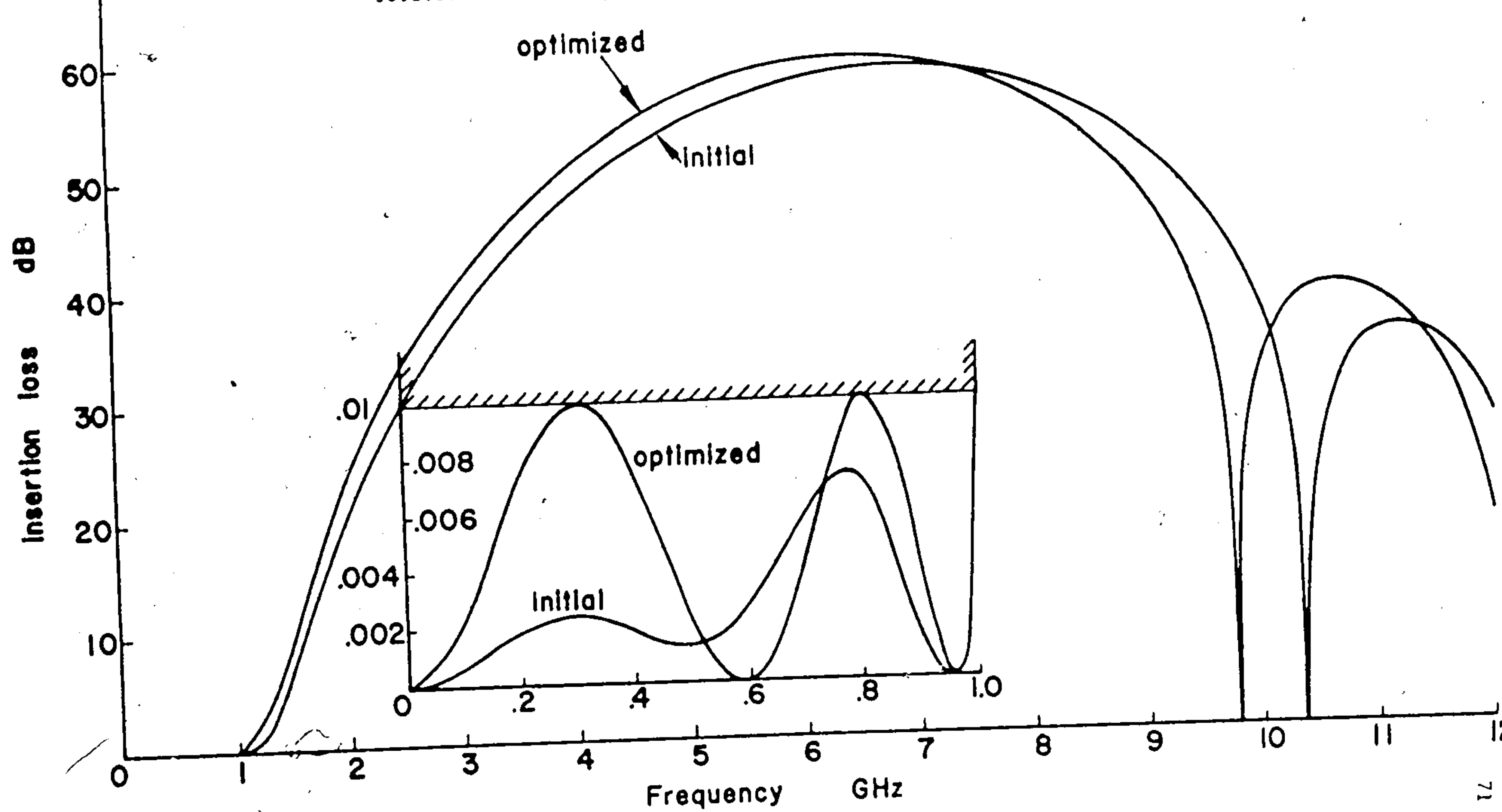


Fig. 4.9 Optimal response for Problem 2 with lengths fixed at ℓ_q and impedances varied. Optimal parameters are: $Z_1=2.528=Z_5$, $Z_2=0.254=Z_4$, $Z_3=4.842$.



Responses for Problem 2 when impedances are fixed and lengths are allowed to vary. The parameter values at start and finish are shown in Table 4.5. The initial response corresponds to best results obtained by Brancher, Maffioli and Premoli (1970), and the optimized response corresponds to the optimal solution obtained by the grazor search method.

TABLE 4.5

5-SECTION FILTER DESIGN (PROBLEM 2)

IMPEDANCES FIXED AT $Z_1 = Z_3 = Z_5 = 0.2$, $Z_2 = Z_4 = 5$, AND LENGTHS VARIED

Parameter	Start	Finish
ℓ_{n1}	0.389	0.480
ℓ_{n2}	0.788	0.814
ℓ_{n3}	0.924	0.990
ℓ_{n4}	0.806	0.814
ℓ_{n5}	0.448	0.480

4.4.3 Problem 3

The design of a 5-section cascaded transmission-line filter subject to parameter constraints is considered here, and the ideas presented in Section 3.3 are used to tackle this problem. The filter structure is the same as the one considered in Section 4.4.2. The problem has been previously considered by Carlin (1971) for fixed lengths at a quarter-wavelength of $\ell_q = 2.5$ cm corresponding to 3 GHz, and for a required attenuation of 0.4 dB in the passband (0-1 GHz). Optimal values have been derived for characteristic impedance values when a stopband frequency of 3 GHz was chosen (Levy 1965). The objective function to be minimized was chosen as (2.13) where

$$y_i(\underset{\sim}{\phi}) = \begin{cases} |\rho_i(\underset{\sim}{\phi})| - r & \psi_i \in 0\text{-}1 \text{ GHz} \\ 1 - |\rho_i(\underset{\sim}{\phi})| & \psi_i = 3 \text{ GHz} \end{cases} \tag{4.8}$$

$\underset{\sim}{\phi}$ corresponds to (4.6) and r corresponds to an attenuation of 0.4 dB. Twenty-one uniformly-spaced points were chosen in the passband.

Initially the lengths were fixed at ℓ_q and the impedances Z_i were varied. The impedance constraints imposed were

$$0.5 \leq Z_i \leq 2.0 \qquad i = 1,2,\ldots, 5 \tag{4.9}$$

and the minimization function was chosen as $W'(\underset{\sim}{\phi},\underset{\sim}{w}')$ of (3.39) where $\underset{\sim}{w}'$ is given by (3.40), $n = 22$, $m = 10$, and

$$\begin{aligned} g_{2i-1}(\underset{\sim}{\phi}) &= Z_i - 0.5 \geq 0 \\ & \qquad\qquad i = 1,2,\ldots, 5 \\ g_{2i}(\underset{\sim}{\phi}) &= -(Z_i - 2.0) \geq 0 \end{aligned} \tag{4.10}$$

$$w_j' = \begin{cases} 1000 \text{ for } g_j(\underset{\sim}{\phi}) < 0 \\ 0 \text{ for } g_j(\underset{\sim}{\phi}) \geq 0 \end{cases} \qquad j = 1,2,\ldots,m \qquad (4.11)$$

The result of optimizing the impedances using the grazor search method is shown in Table 4.6 where U corresponds to $\max_i y_i(\underset{\sim}{\phi})$, and y_i is given by (4.8). It is observed that some of the impedances of the constrained solution lie on constraint boundaries. Moreover, there are two distinct solutions, for which the impedances are reciprocals of each other.

As a further step, it was desired to investigate the possibility of improving the unconstrained optimal solution (for length fixed at ℓ_q) of Table 4.6, by allowing both the lengths and impedances to vary, and imposing the following constraints:

$$0 \leq \ell_{ni} \leq 2 \qquad i = 1,2,\ldots,5 \qquad (4.12)$$

$$0.4416 \leq Z_i \leq 4.419 \qquad i = 1,2,\ldots,5 \qquad (4.13)$$

$$0 \leq \sum_{j=1}^{5} \ell_{nj} \leq 5 \qquad (4.14)$$

where the ℓ_{ni} correspond to (4.7) and the upper and lower bounds of Z_i in (4.13) correspond to upper and lower values of the unconstrained optimal values of Table 4.6.

TABLE 4.6

5-SECTION TRANSMISSION-LINE LOWPASS FILTER DESIGN (PROBLEM 3)

FOR LENGTHS FIXED AT ℓ_q

Parameters	Unconstrained Optimal Solution	Constrained Solution (i)	Constrained Solution (ii)
Z_1	3.151	0.5683	1.760
Z_2	0.4416	2.000	0.5000
Z_3	4.419	0.5000	2.000
Z_4	0.4416	2.000	0.5000
Z_5	3.151	0.5683	1.760
U	3.951×10^{-5}	3.255×10^{-3}	3.255×10^{-3}
W'	2.419×10^{3}	3.255×10^{-3}	3.255×10^{-3}

The function to be minimized was chosen as $V(\underset{\sim}{\phi}, \phi_{k+1}, \underset{\sim}{\alpha})$ in (3.32) where $\underset{\sim}{\alpha}$ is given by (3.33) and (3.34), n = 22, m = 22,

$$\alpha_j = 10 \qquad j = 1,2,\ldots,m+1 \tag{4.15}$$

and $g_j(\underset{\sim}{\phi})$, j = 1,2,...,m correspond to the constraints (4.12)-(4.14). It was observed that no improvement could be achieved from the starting value (corresponding to the unconstrained optimal solution of Table 4.6) and that the starting point satisfies the necessary conditions for a minimax optimum, as verified by the method described in Section 3.4 and Appendix B.

4.5 Conclusions

The results indicate that the grazor search algorithm is generally more reliable in reaching an optimal minimax solution than the Osborne and Watson algorithm, and is faster than the razor search technique. Typically 1 min is sufficient time to optimize a six-parameter design, and 2 to 3 min are sufficient to optimize a ten-parameter problem, depending on how far from the optimum one starts and how close one wishes to get, on a CDC 6400 computer. The grazor search algorithm is capable of handling, without any difficulty, filter design problems with upper and lower specifications over many frequency bands. The method should be very useful in design problems for which exact methods are not available.

CHAPTER V

SYSTEM MODELLING

5.1 Introduction

Lower-order modelling of complex high-order systems is now widely being used in the area of systems design and control both on-line and off-line. The modelling can be performed for a variety of performance criteria and objectives, using different model derivation techniques. Some of the techniques obtain a model by neglecting modes of the original system which contribute little to the overall response of the system (Davison 1966, Chidambara 1969, Mitra 1969, Marshall 1966). Other methods search for optimal coefficients of a set of differential or difference equations of a given 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 1967, Sinha and Pille 1971, Sinha and Bereznai 1971, Markettos 1972). The search of these coefficients has been, in the past, carried out using both direct search and gradient methods of optimization for a least-squares or quadratic cost function, but for this work, the investigation is mainly on near-minimax and minimax objectives, and the input-output data of the system is assumed to be known. See also Chen and Shieh (1968) and Kokotović and Sannuti (1968).

5.2 Statement of the Problem

It is required to find a transfer function of a model of a given order, the response of which is the best approximation to the response of the actual system to a particular input for a specified error criterion.

In general the transfer function of a given order n may be written as

$$H_{m,n}(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}$$

$$= \frac{\sum_{i=0}^{m} b_{m-i} s^{m-i}}{s^n + \sum_{i=1}^{n} a_{n-i} s^{n-i}} \qquad (5.1)$$

where $m \leq n$ for physical systems. For this work the input is a unit step and the criterion chosen is to directly or indirectly minimize an error function over a specified time interval $[0,T]$. The problem, therefore, is the determination of the parameters $\tilde{\phi}$, given by

$$\tilde{\phi} = [a_0\ a_1\ \ldots\ a_{n-1}\ b_0\ b_1\ \ldots b_m]^T \qquad (5.2)$$

such that an error function is minimized. Optimization of model parameters for a least-squares error criterion has already received attention (Bandler, Markettos and Sinha 1973, Markettos 1972).

5.3 Minimax System Modelling

The error criterion chosen is to minimize the maximum error between the system and model responses over [0,T], where $\underset{\sim}{\phi}$ is given by (5.2). The following notation is introduced.

t_i	is an i th time instant in [0,T]
I	is an index set of i such that $t_i \epsilon [0,T]$
c_i^s	is the response of the system at t_i
$c_i^m(\underset{\sim}{\phi})$	is the response of the approximating model at t_i
$e_i(\underset{\sim}{\phi})=c_i^m(\underset{\sim}{\phi})-c_i^s$	is the error between the system and the model responses at t_i
c_∞^s	is the steady-state value of the system
c_∞^m	is the steady-state value of the model

In Section 5.4, the approximation problem considered assumes that c_∞^m is fixed at a convenient value (usually c_∞^s or c_i^s at t_i=T), so that the objective is to minimize

$$U(\underset{\sim}{\phi}) = \max_{t_i \epsilon [0,T]} y_i(\underset{\sim}{\phi}) \tag{5.3}$$

where

$$y_i(\underset{\sim}{\phi}) = |e_i(\underset{\sim}{\phi})| \tag{5.4}$$

This problem can now be solved by an efficient minimax or near-minimax optimization method as suggested in Sections 2.5 and 2.6.

5.4 Example

The problem considered is the modelling of a seventh-order system representing the control system for the pitch rate of a supersonic transport aircraft (Dorf 1967, Bandler, Markettos and Sinha 1973). The transfer function of the system is given by

$$G(s) = \frac{375000(s+0.08333)}{\left(\begin{array}{r} s^7+83.64s^6+4097s^5+70342s^4+853703s^3+ \\ 2814271s^2+3310875s+281250 \end{array}\right)} \tag{5.5}$$

with a steady-state value of 0.11111 for a unit step.

Minimax optimization of the model parameters as performed by the grazor search method consists of minimizing (5.3), while near-minimax optimization minimizes

$$f(\underset{\sim}{\phi}) = U(\underset{\sim}{\phi})\left(\sum_{t_i \epsilon [0,T]} \left|\frac{e_i(\underset{\sim}{\phi})}{U(\underset{\sim}{\phi})}\right|^p\right)^{1/p} \tag{5.6}$$

for large values of p (Bandler and Charalambous 1972d). Let $J \subset I$ be an index set relating only to the extrema of the error functions $y_i(\underset{\sim}{\phi})$ given by (5.4). If I is replaced by J in (5.6), considerable economy in computing time results at a slight risk of creating false optima. The larger the value of p, the closer the solution gets to the minimax result, but the central processor time increases considerably. For this work, a value of p=1000 was considered suitable for optimization purposes. For least p th optimization, three gradient methods due to Fletcher and Powell (1963), Jacobson-Oksman(1972) and Fletcher (1970) have been used for the modelling problem.

5.4.1 Second- and Third-Order Models

The time-interval over which the approximation was made was 0-8 seconds (T=8 sec). 101 uniformly-spaced sample points were chosen over the interval. The steady state value of the model for a unit step ($E=c_\infty^m$), was set at 0.11706, corresponding to the response of the system at the final sample point (c_i^s for $t_i=T$). See Bandler, Markettos and Srinivasan (1972, 1973).

Two second-order and one third-order models were considered for minimax approximation of the system. The transfer functions of the chosen models were

$$H_{02}(s) = \frac{Ea_0}{s^2+a_1s+a_0} \tag{5.7}$$

$$H_{12}(s) = \frac{b_1s+Ea_0}{s^2+a_1s+a_0} \tag{5.8}$$

$$H_{23}(s) = \frac{b_2s^2+b_1s+Ea_0}{s^3+a_2s^2+a_1s+a_0} \tag{5.9}$$

$$= \frac{x_5s^2+x_4s+Ex_1x_3}{(s+x_3)(s^2+x_2s+x_1)} \tag{5.10}$$

where

$$E \triangleq c_\infty^m \tag{5.11}$$

For this work, the response of the models in the time domain were obtained by using standard Laplace Transform Tables to invert from the s to the t domain.

(a) 2-Parameter Problem

The model transfer function chosen is (5.7) and the parameter vector is given by

$$\underset{\sim}{\phi} = [a_0 \; a_1]^T \qquad (5.12)$$

The optimum parameters using the grazor search method were

$a_0 = 3.06472$, $a_1 = 2.38338$

resulting in a four-ripple error curve with a maximum error value

$$U = 3.76347 \times 10^{-3}$$

The response and error curves are shown in Figs. 5.1(a) and 5.1(b) respectively.

The optimum parameters using least pth approximation for p=1000 were

$a_0 = 3.06549$, $a_1 = 2.38414$

resulting in a similar four-ripple curve with a maximum error value

$$U = 3.76510 \times 10^{-3}$$

Table 5.1 shows the number of function evaluations required for each of the methods to reach a maximum error value of 3.76619×10^{-3}. For this problem the Fletcher method and Jacobson-Oksman method appeared to be the most efficient.

(b) 3-Parameter Problem

By allowing the model to have a zero, as indicated by (5.8) a 3-variable problem results, where

$$\underset{\sim}{\phi} = [a_0 \; a_1 \; b_1]^T \qquad (5.13)$$

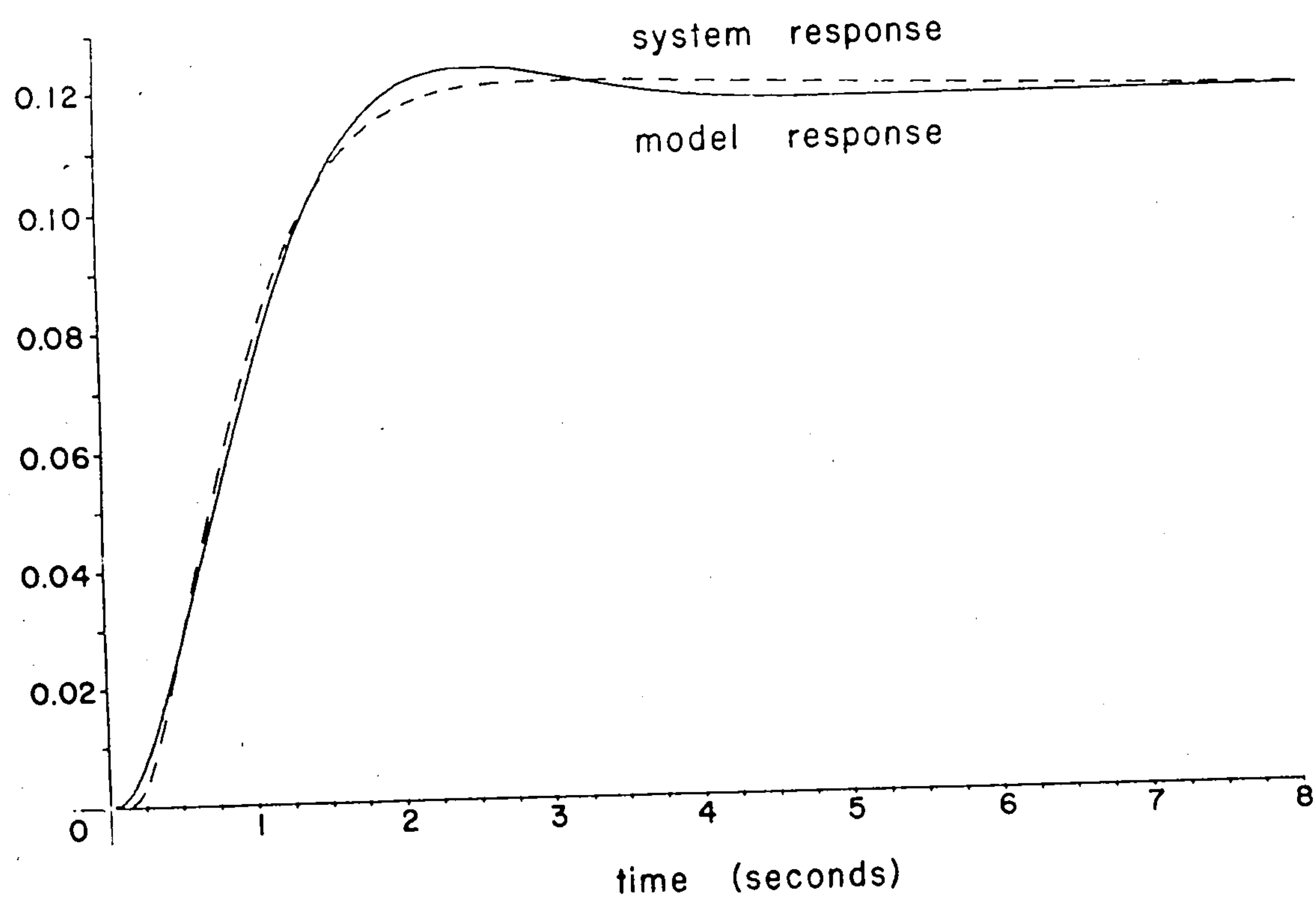


Fig. 5.1(a) Seventh-order system modelling example. 2-parameter optimum response.

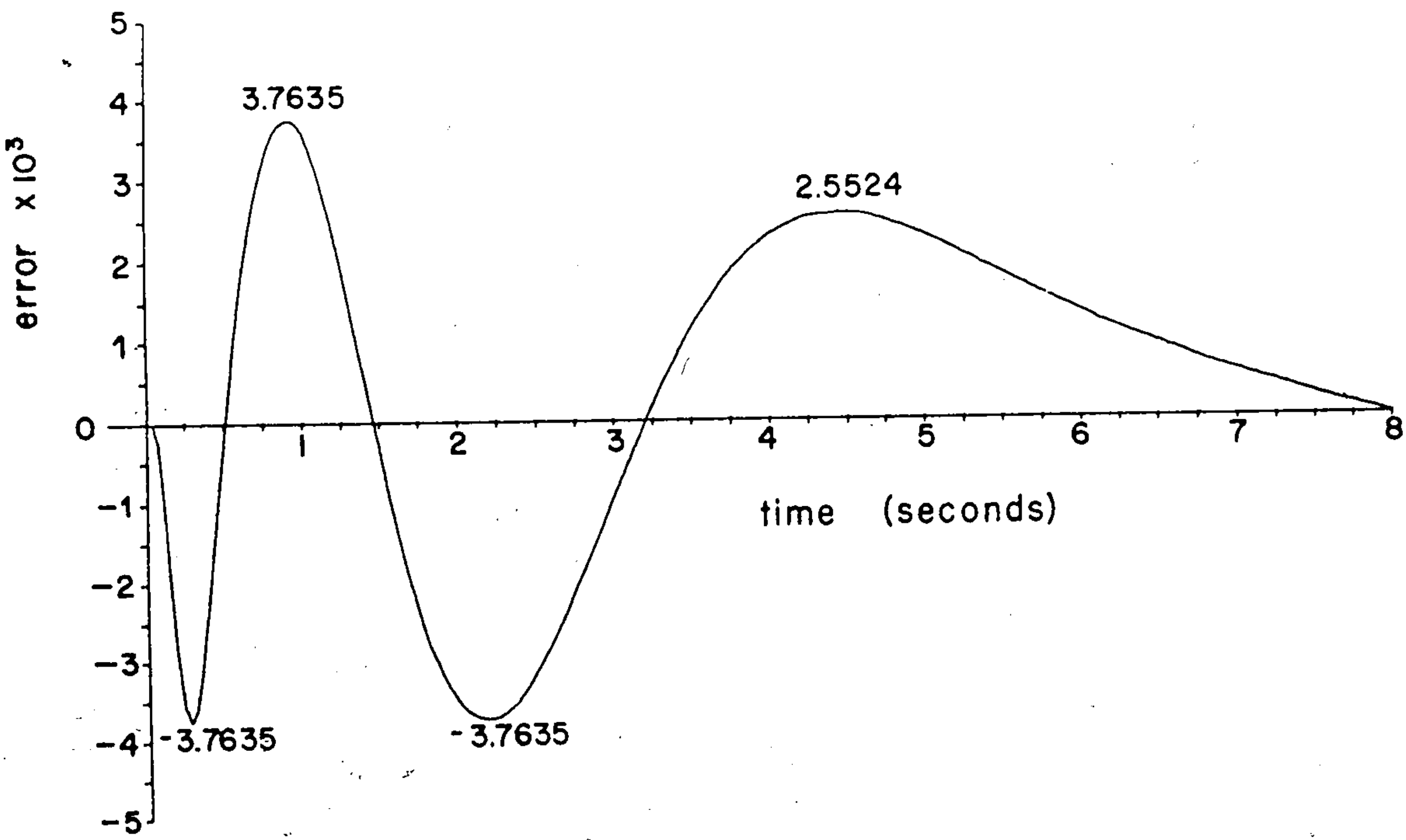


Fig. 5.1(b) Seventh-order system modelling example. 2-parameter optimum error curve.

TABLE 5.1

SEVENTH-ORDER SYSTEM MODELLING EXAMPLE

NUMBER OF FUNCTION EVALUATIONS REQUIRED TO REACH U = 3.76619×10^{-3} FOR THE 2-PARAMETER MODEL

Starting point $\underset{\sim}{\phi}$	Minimization of $U(\underset{\sim}{\phi})$ Grazor	Minimization of $f(\underset{\sim}{\phi})$		Jacobson - Oksman	
		Fletcher	Fletcher-Powell	Quadratic Step Prediction	Homogeneous Step Prediction
3.0 2.0	107	42	59	36	36
1.0 1.0	130	78	334	91	127
1.0 4.0	165	96	718	834	*
4.0 1.0	129	64	false optimum	41	45

*Indicates an ARGUMENT TOO LARGE message was given by the computer.

The optimum parameters using the grazor search method were

$a_0 = 3.83255$, $a_1 = 3.00365$, $b_1 = -.0176390$

giving a maximum error value

$U = 2.48724x10^{-3}$

The response and error curves are shown in Figs. 5.2(a) and 5.2(b) respectively.

For $p=1000$ the optimum parameters obtained were

$a_0 = 3.83592$, $a_1 = 3.00605$, $b_1 = -.0177277$

giving similar response and error curves as in Figs. 5.2(a) and 5.2(b) and

$U = 2.48794x10^{-3}$

The number of function evaluations needed for the three parameter problem to reach the value $U = 2.48794x10^{-3}$ are shown in Table 5.2. The grazor search technique and the Fletcher method required a smaller number of function evaluations.

(c) 5-Parameter Problem

The third-order model of (5.9) is considered next. For computional efficiency, the transfer function of the form (5.10) is chosen. The model has five parameters given by

$$\underset{\sim}{\phi}=[x_1\ x_2\ x_3\ x_4\ x_5]^T \qquad (5.14)$$

The optimum parameters obtained using the grazor search method

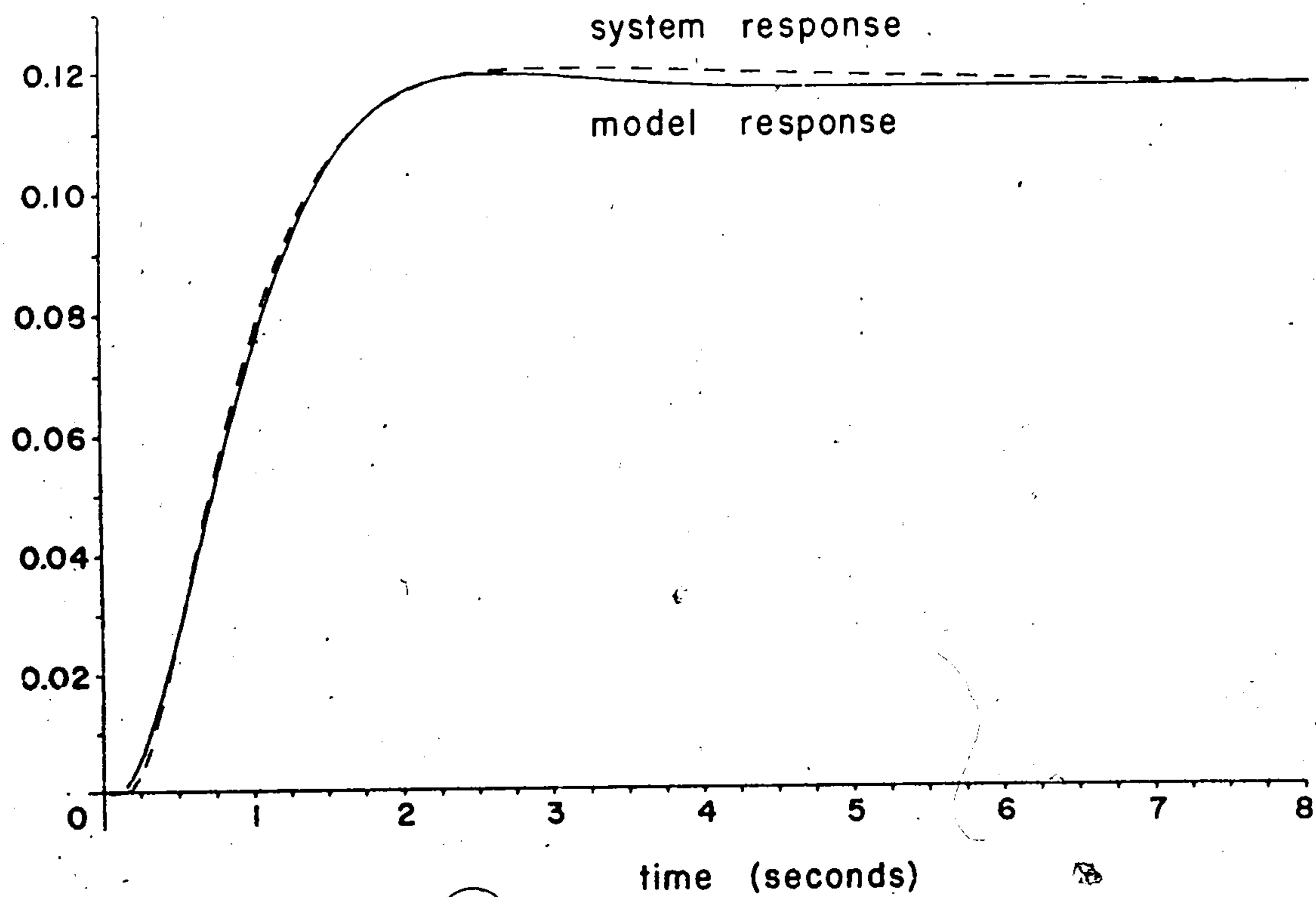


Fig. 5.2(a) Seventh-order system modelling example. 3-parameter optimum response.

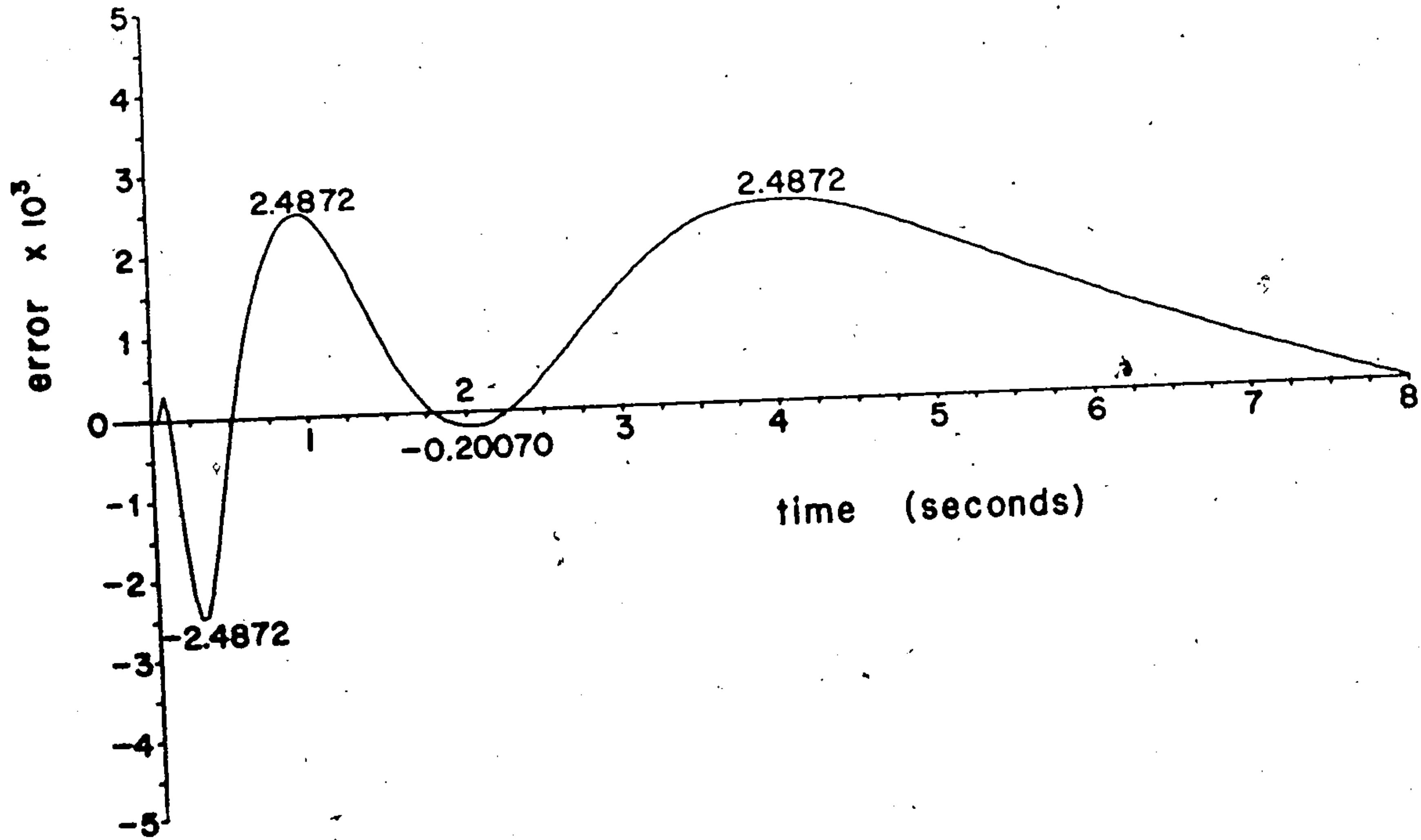


Fig. 5.2(b) Seventh-order system modelling example. 3-parameter optimum error curve.

TABLE 5.2

SEVENTH-ORDER SYSTEM MODELLING EXAMPLE

NUMBER OF FUNCTION EVALUATIONS REQUIRED TO REACH U = 2.48794×10^{-3} FOR THE 3-PARAMETER MODEL

Starting point $\underset{\sim}{\phi}$	Minimization of $U(\underset{\sim}{\phi})$ Grazor	Minimization of $f(\underset{\sim}{\phi})$: Fletcher	Minimization of $f(\underset{\sim}{\phi})$: Fletcher-Powell	Jacobson - Oksman: Quadratic Step Prediction $\rho = 1$	Jacobson - Oksman: Quadratic Step Prediction $\rho = 0.5$	Jacobson - Oksman: Homogeneous Step Prediction
2.5 2.0 -2.0	149	339	500	279	•	339
1.0 1.0 -1.0	368	362	+	104	276	137
4.0 3.0 0.01	165	242	184	142	97	260
3.5 1.5 -1.0	358	280	342	217	151	•
5.0 1.0 -1.0	325	193	+	•	205	•
5.0 1.0 3.0	406	245	+	159	119	•

+Indicates time limit of 64 seconds was reached.

•Indicates an ARGUMENT TOO LARGE message was given by the computer.

were

$$x_1 = 4.34547,\ x_2 = 3.36809,\ x_3 = .108248$$

$$x_4 = .514475,\ x_5 = -.0356180$$

resulting in a six-ripple error curve with a maximum error value

$$U = 1.02062\text{x}10^{-3}$$

The response and error curves are shown in Figs. 5.3(a) and 5.3(b) respectively.

The optimum parameters using p=1000 were

$$x_1 = 4.34682,\ x_2 = 3.36738,\ x_3 = .0996086$$

$$x_4 = .514728,\ x_5 = -.0356154$$

giving response and error curves similar to those of Figs. 5.3(a) and 5.3(b) and a maximum error

$$U = 1.02063\text{x}10^{-3} \quad .$$

Some runs with the Fletcher-Powell method, on the five-parameter problem, indicated that the method was the slowest and since this was already established in the previous models, as indicated in Tables 5.1 and 5.2, further runs with Fletcher-Powell method were considered unnecessary. The results of optimization by the other three methods are shown in Table 5.3.

The Fletcher method reached a unique six-ripple solution in all the cases tried, although there was a large variation in the number of function evaluations required. The grazor search technique reached the

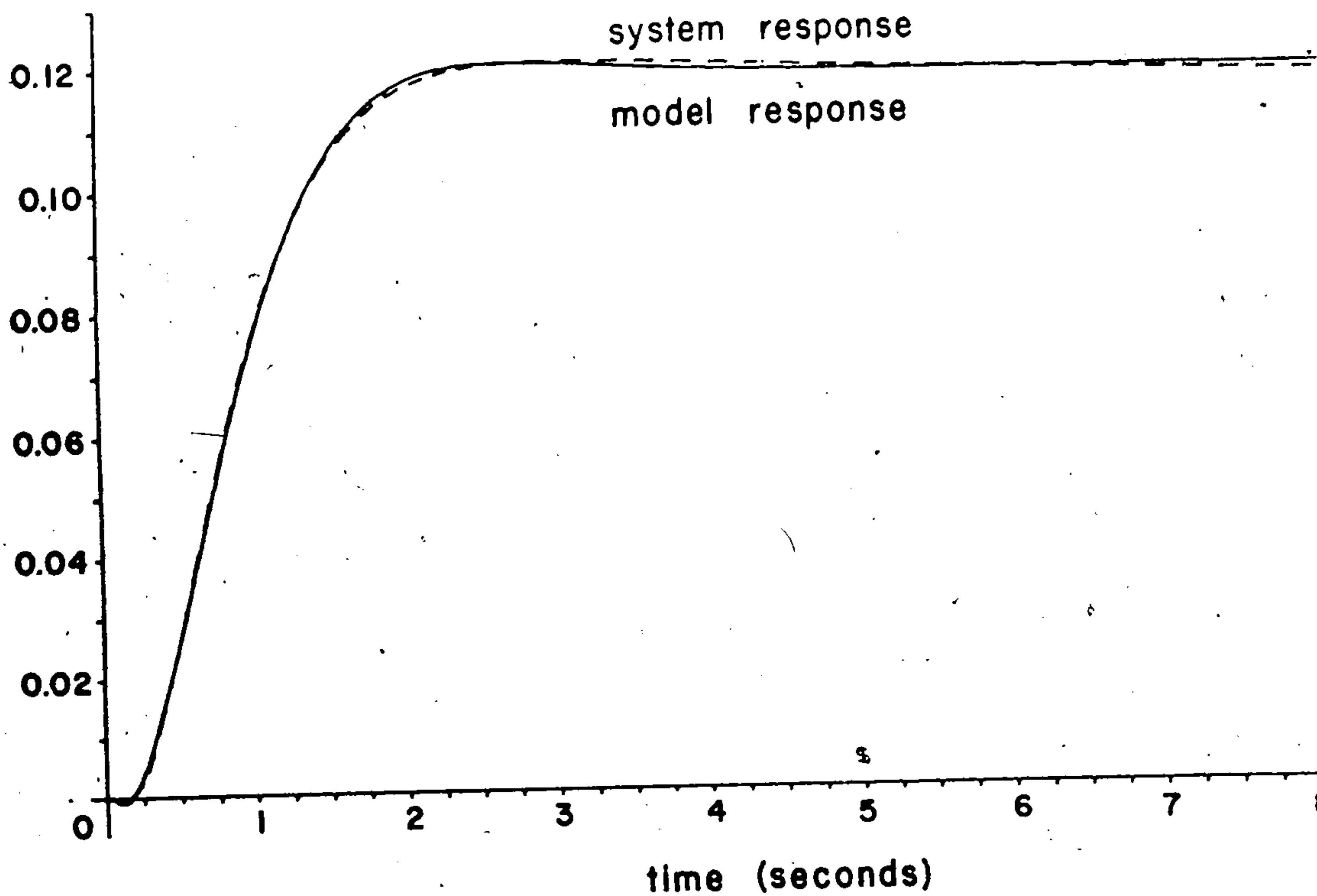


Fig. 5.3(a) Seventh-order system modelling example. 5-parameter six-ripple optimum response.

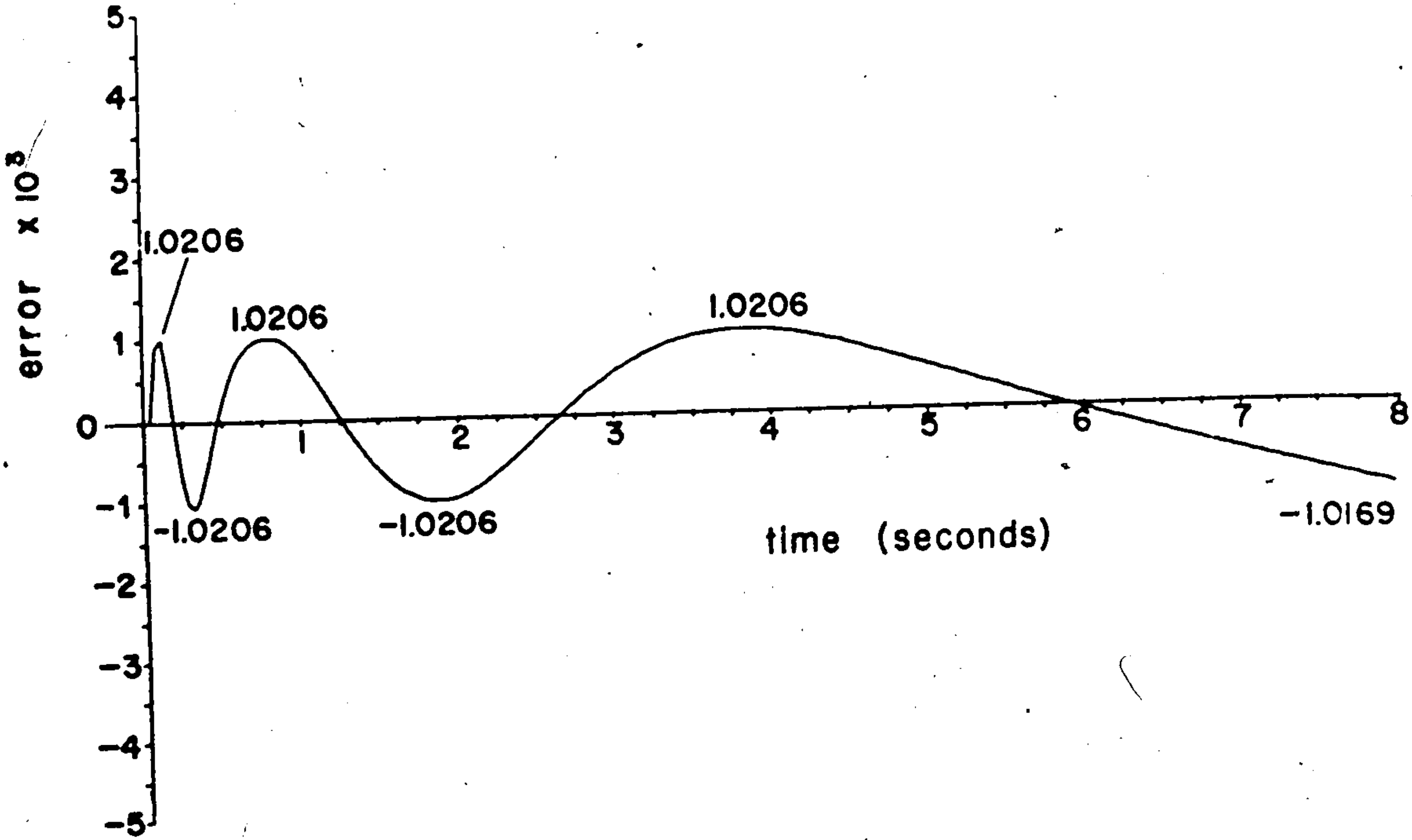


Fig. 5.3(b) Seventh-order system modelling example. 5-parameter six-ripple optimum error curve.

TABLE 5.3

SEVENTH-ORDER SYSTEM MODELLING EXAMPLE

NUMBER OF FUNCTION EVALUATIONS REQUIRED TO REACH THE INDICATED VALUE OF 1000 U FOR THE 5-PARAMETER MODEL

Starting point $\underset{\sim}{\phi}$	Minimization of $U(\underset{\sim}{\phi})$ Grazor	Minimization of $f(\underset{\sim}{\phi})$ Fletcher	Jacobson-Oksman Quadratic Step Prediction $\rho = 1$	Jacobson-Oksman Quadratic Step Prediction $\rho = 0.5$
3.0 3.0 1.5 0.5 -0.1	437 1.2139	530 1.0207	886 1.0206	778 1.0206
1.5 3.0 2.5 1.0 0.1	782 1.2473	768 1.0207	931 1.0206	325* 45.086
4.0 3.0 0.1 0.5 -0.03	489 1.0206	177 1.0207	114* 1.5061	108 1.0206
3.0 5.0 0.2 0.3 -0.1	634 1.1720	862 1.0207	248 1.0206	350 1.0207
5.0 4.0 0.5 1.0 -0.5	817 1.0337	484 1.0207	17* 19.660	582 1.0207
Least Squares Optimum	537 1.2472	799 1.0206	263* 1.8954	1208++ 1.0283

++Indicates time limit of 128 seconds was reached.

*Indicates an ARGUMENT TOO LARGE message was given by the computer.

six-ripple solution in one of the cases shown, while in some of the other cases it terminated in a five-ripple solution.

In some instances, the real pole of the model had the tendency to move to the right-hand side of the s-plane and since this would produce an unstable model, the last parameters giving stable results were taken as the final values. In all cases, however, the real pole seems to lie very close to the axis and any constraint, although easily implemented in the form of a square transformation, would have made the pole go to zero.

It was further noted that when the Fletcher method, used with p=1000, was started from one of the five-ripple solutions where the grazor search technique terminated, a direction was found which decreased $f(\underset{\sim}{\phi})$ while temporarily increasing $U(\underset{\sim}{\phi})$ and the method converged towards the six-ripple minimax solution, though slowly. When the same procedure was repeated with $p=10^6$, the algorithm failed to move from that point. Figs. 5.4(a) and 5.4(b) show the response and error curves for a five-ripple solution obtained by the grazor search method.

5.4.2 Optimality of Model Parameters

The conditions for minimax optimality, as mentioned in Section 3.4.2, were applied to the final parameter values arrived at through optimization of the grazor search method (the corresponding responses are shown in Figs. 5.1-5.4), and the results are indicated in Tables 5.4-5.7. The necessary conditions are satisfied in all the cases, as observed from the tables. The $g_\ell(\underset{\sim}{\phi})$ for $\ell = 1,2...,n_r$ are the local

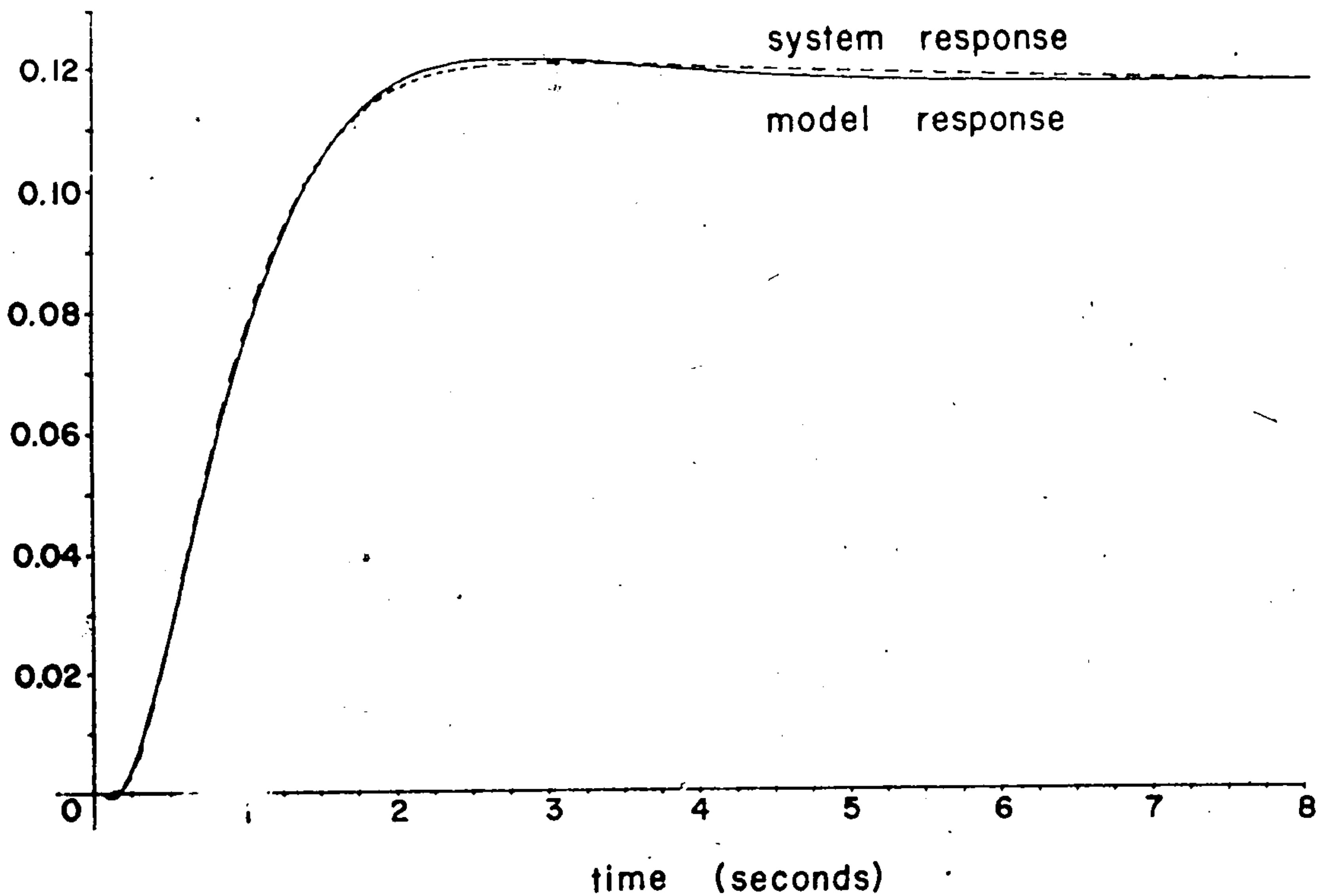


Fig. 5.4(a) Seventh-order system modelling example. 5-parameter five-ripple solution response.

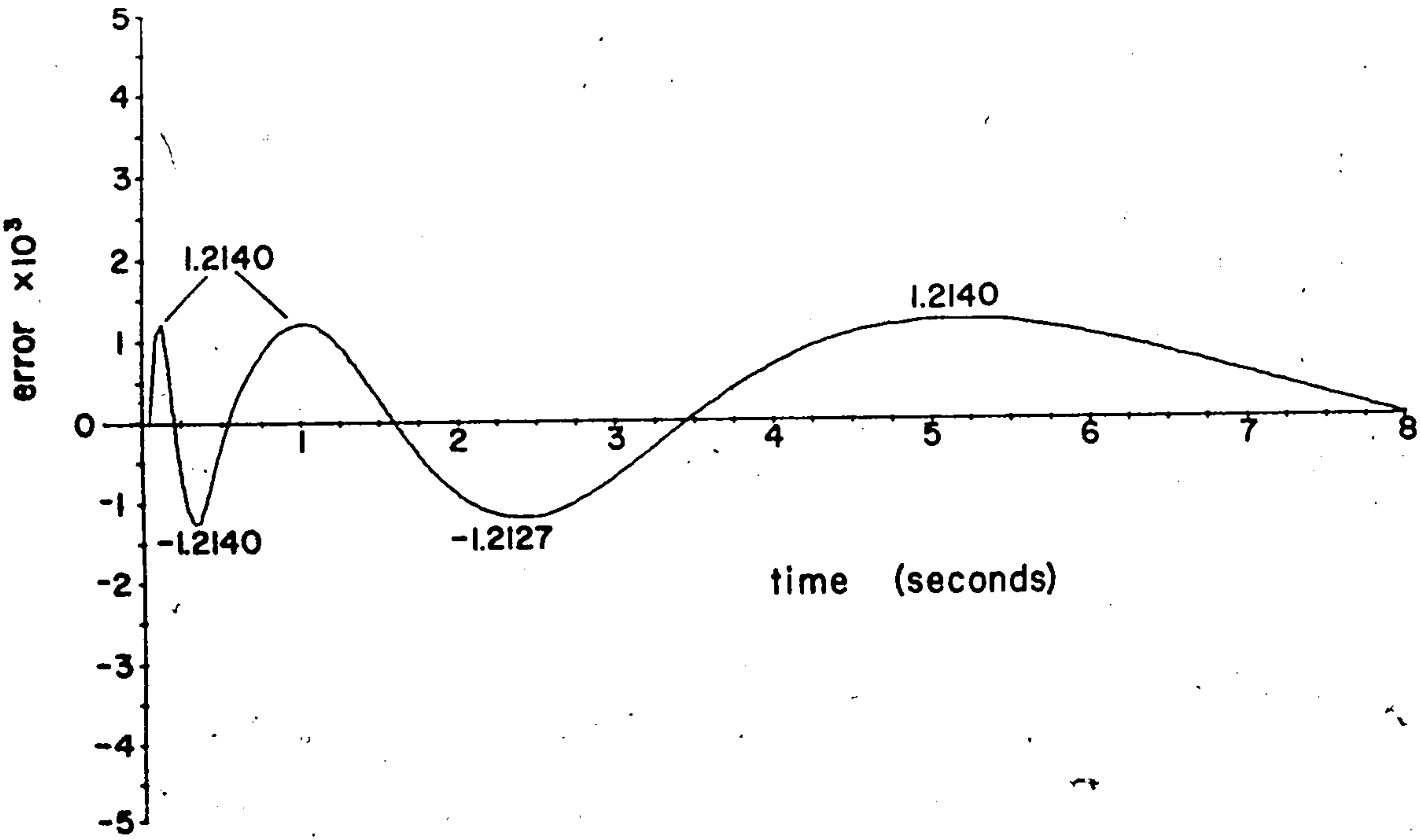


Fig. 5.4(b) Seventh-order system modelling example. 5-parameter five-ripple solution error curve.

TABLE 5.4 .

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

2-PARAMETER SOLUTION CORRESPONDING TO FIG. 5.1

$n_r = 4, \quad k_r^* = 3$

ℓ	Time Instant	Error Maximum $(1000\hat{y}_\ell)$	Multiplier (u_ℓ)
1	0.24	3.76347	0.75047
2	0.88	3.76347	0.16519
3	2.16	3.76347	8.4342×10^{-2}
4	4.40	2.55235	-

$$\underset{\sim}{r} = \sum_{\ell=1}^{k_r^*} u_\ell \underset{\sim}{\nabla} \hat{y}_\ell = [0.0 \quad . \; 0.0]^T$$

$$\sum_{\ell=1}^{k_r^*} u_\ell = 1.0$$

TABLE 5.5

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

3-PARAMETER SOLUTION CORRESPONDING TO FIG. 5.2

$n_r = 4, \quad k_r^* = 3$

ℓ	Time Instant	Error Maximum $(1000\hat{y}_\ell)$	Multiplier (u_ℓ)
1	4.0	2.48724	0.90758
2	0.24	2.48724	4.2744×10^{-2}
3	0.96	2.48724	4.9680×10^{-2}
4	2.00	2.00700×10^{-1}	-

$$\underset{\sim}{r} = \sum_{\ell=1}^{k_r^*} u_\ell \nabla \underset{\sim}{\bar{y}}_\ell = [0.0 \quad 0.0 \quad 1.1\text{x}10^{-5}]^T$$

$$\sum_{\ell=1}^{k_r^*} u_\ell = 1.0$$

TABLE 5.6

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

5-PARAMETER, 6-RIPPLE SOLUTION CORRESPONDING TO FIG. 5.3

$n_r = 6, \; k_r^* = 6$

ℓ	Time Instant	Error Maximum ($1000\hat{y}_\ell$)	Multiplier (u_ℓ)
1	1.84	1.020616	3.6510×10^{-2}
2	0.72	1.020616	8.4333×10^{-2}
3	0.08	1.020616	0.51806
4	3.76	1.020616	2.7915×10^{-2}
5	0.24	1.020616	0.32227
6	8.00	1.016870	1.0910×10^{-2}

$$\underset{\sim}{r} = \sum_{\ell=1}^{k_r^*} u_\ell \nabla \underset{\sim}{y}_\ell = [0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$$

$$\sum_{\ell=1}^{k_r^*} u_\ell = 1.0$$

TABLE 5.7

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

5-PARAMETER, 5-RIPPLE SOLUTION CORRESPONDING TO FIG. 5.4

$n_r = 5, \quad k_r^* = 5$

ℓ	Time Instant	Error Maximum ($1000\hat{y}_\ell$)	Multiplier (u_ℓ)
1	0.32	1.213988	0.23428
2	5.12	1.213988	0.19815
3	0.08	1.213988	0.39281
4	0.96	1.213986	0.10217
5	0.32	1.212651	7.2598×10^{-2}

$$\underset{\sim}{r} = \sum_{\ell=1}^{k_r^*} u_\ell \nabla \underset{\sim}{\bar{y}}_\ell = [-1.5 \times 10^{-5} \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0]^T$$

$$\sum_{\ell=1}^{k_r^*} u_\ell = 1.0$$

discrete maxima of $y_i(\underset{\sim}{\phi})$, $i \in I$ as mentioned in Section 3.3.1, and Method 2 described in Section 3.4.5 is used for verifying the optimality conditions.

For the cases corresponding to Tables 5.5 and 5.7, k_r^* is equal to k and there are k_r^*+1 equations and k_r^* unknowns for the solution of (3.46) and (3.51). The non-zero values of the components of $\underset{\sim}{r}$ for these cases correspond to the residuals of the dependent equntions (refer to Sections 3.4.5, 3.4.6 and Appendix B).

In interpreting these results one may associate the results corresponding to Tables 5.4 and 5.6 in saying that the main criterion is how close to equal the ripples are and the results of Tables 5.5 and 5.7 in how small the size of the linear combination is in comparison with the sizes of the individual gradient vectors. In the first case we are satisfied with the criterion from a practical point of view, in the second the linear combination is about 2 to 4 orders of magnitude smaller than the gradient vectors.

5.4.3 Discussion

The grazor search algorithm is found to be more efficient than the Fletcher-Powell method on the problems chosen. The method proposed by Fletcher appears to be the most efficient of the methods used for near-minimax results in efficiency and consistency in reaching the vicinity of the optimum. The Jacobson-Oksman method, although giving good results, appeared to be sensitive to scaling.

It has to be mentioned that the Fletcher-Powell package, as available in the IBM Scientific Subroutine Package, has a programming error. Appropriate corrections have been made and the Fletcher-Powell method has been applied to a number of test problems. The results have indicated that very little improvement is obtained for the corrected version. The Fletcher-Powell results, as shown in Tables 5.1-5.2, correspond to the uncorrected version, and it is expected that the corrected version might improve the function evaluations slightly.

5.5 New Approaches to Minimax System Modelling

In this section, some new ideas are presented so as to satisfy stringent design requirements (Bandler and Srinivasan 1973b, 1973e). In Section 5.4, c_∞^m was assumed fixed. It may, however, be unacceptable to fix c_∞^m at a certain value, in which case a realistic trade-off between transient and steady-state errors can be achieved. The design requirement may be such that arbitrary transient and steady-state response specifications need be imposed on the model for a desired performance criterion. It would also be realistic to expect the modelling procedure to be automated in such a way that it is possible to move from lower-order models to high-order ones whenever, say, the solutions satisfy the necessary optimality conditions.

5.5.1 A Generalized Objective Function

It is possible to extend the ideas of constrained minimax optimization (discussed in Section 3.3) to system modelling so that a generalized objective function can be defined to take into account both the transient and steady-state response errors. The following additional notation is introduced.

$S_{u\infty}$ is the upper bound of the system specifications at steady-state

$S_{\ell\infty}$ is the lower bound of the system specifications at steady-state

$e_{u\infty}=c_\infty^m-S_{u\infty}$ is the error between upper steady-state specifications and model steady-state value

$e_{\ell\infty}=c_\infty^m-S_{\ell\infty}$ is the error between lower steady-state specifications and model steady-state value

The problem may now be formulated into two forms as follows. The first one minimizes with respect to $\underset{\sim}{\phi}$ and ϕ_{k+1}

$$V(\underset{\sim}{\phi},\phi_{k+1},\alpha,\alpha_{\ell\infty},\alpha_{u\infty}) = \max_{t_i\in[0,T]} [\phi_{k+1},\phi_{k+1}-\alpha(\phi_{k+1}-|e_i(\underset{\sim}{\phi})|) ,$$

$$\phi_{k+1}-\alpha_{\ell\infty}e_{\ell\infty},\phi_{k+1}+\alpha_{u\infty}e_{u\infty}] \tag{5.15}$$

where α, $\alpha_{\ell\infty}$, $\alpha_{u\infty}$ are positive. If c_∞^m is fixed such that $e_{\ell\infty}$ and $-e_{u\infty}$ are positive, the objective function (5.15) reduces essentially to $U(\underset{\sim}{\phi})$ in (5.3). The second one minimizes with respect to $\underset{\sim}{\phi}$

$$W(\underset{\sim}{\phi},w_{\ell\infty},w_{u\infty}) = \max_{t_i\in[0,T]} [|e_i(\underset{\sim}{\phi})|,-w_{\ell\infty}e_{\ell\infty},w_{u\infty}e_{u\infty}] \tag{5.16}$$

where

$$w_{\ell\infty} \begin{cases} = 0 & \text{for } -e_{\ell\infty} < 0 \\ > 0 & \text{for } -e_{\ell\infty} \geq 0 \end{cases} \tag{5.17}$$

$$w'_{u\infty} \begin{cases} = 0 & \text{for } e_{u\infty} < 0 \\ > 0 & \text{for } e_{u\infty} \geq 0 \end{cases} \tag{5.18}$$

If c^m_∞ is fixed within satisfied specifications the above objective function reduces to $U(\phi)$ in (5.3).

In cases where suitable constraints - including parameter constraints - are imposed, the above procedure may be used to incorporate them in the objective function. In many cases, it is convenient to choose $S_{\ell\infty} = S_{u\infty} = c^s_\infty$.

5.5.2 Automated Lower-order Models

One of the major problems that is encountered in modelling is to decide whether a certain lower-order model is acceptable or not. If the model is too simple so that computing time for optimizing model parameters is small, the approximation to the original system may be very bad, while if the model is complex, then the very need for system modelling is lost. If one were to strike a reasonable compromise between the speed with which the model is optimized, and the accuracy of the approximation, it would not be unreasonable to devise a scheme whereby one could increase the complexity of the model in an automated fashion after a certain number of iterations or computer time. It is, however, important to keep in mind the desirability of making this increase in complexity as smooth as possible, so that the objective function value is not degraded. Thus, either the number of parameters could be increased for a model with a certain order, or the order of the model itself can be increased.

Let $H^*_{m,n}$ denote an optimized model of the form (5.1). Three possibilities occur as follows.

(i) Increase in parameters only

$$H^*_{m,n}(s) \rightarrow H_{m+p,n}(s)$$

Here $b_{m+p}, b_{m+p-1}, \ldots, b_{m+1}$ are initially assumed to be zero so that $H_{m+p,n} = H^*_{m,n}$ in the first iteration.

(ii) Increase in order

$$H^*_{m,n}(s) \rightarrow H_{m+q,n+q}(s)$$

Here q poles of $H_{m+q,n+q}(s)$ are assumed to cancel with q zeros initially, so that $H_{m+q,n+q} = H^*_{m,n}$ in the first iteration. In this case, initial guesses for q poles (or zeros) are necessary.

(iii) Increase in order and parameters

$$H^*_{m,n}(s) \rightarrow H_{m+p+q,n+q}(s)$$

Here $b_{m+q+p}, \ldots, b_{m+q+1}$ are assumed to be zero initially and that there is a cancellation of q zeros and q poles at start, so that

$H_{m+p+q,n+q} = H^*_{m,n}$ in the first iteration.

A careful choice of initial parameters can make the increase in model complexity smooth so that the whole modelling procedure can be automated on a small digital computer on-line.

5.5.3 Optimality Conditions

When a certain low-order model is being optimized, it may be useful to investigate intermediate or final solutions after a certain number of iterations of the modelling algorithm, or after a certain convergence criterion is reached, so that one may decide whether to carry on with further optimization, to increase the order of the model, or to terminate altogether. For minimax objectives, it is possible to test the optimality by the procedure outlined in Section 3.4.

5.5.4 Results

Two examples were considered, and two second-order models and a third-order model were chosen as follows.

$$H_{02}(s) = \frac{b_0}{s^2+a_1 s+a_0} \tag{5.19}$$

$$H_{12}(s) = \frac{B_1 s+B_0}{s^2+A_1 s+A_0} \tag{5.20}$$

$$H_{23}(s) = \frac{x_6 s^2+x_5 s+x_4}{(s+x_3)(s^2+x_2 s+x_1)} \tag{5.21}$$

The transition between the models can be made smooth by making the following substitutions at the start of the new model.

$$H_{02}^{*} \rightarrow H_{12} : A_0 = a_0^{*},\ A_1 = a_1^{*},\ B_0 = b_0^{*},\ B_1 = 0$$

$$H_{02}^{*} \rightarrow H_{23} : x_1 = a_0^{*},\ x_2 = a_1^{*},\ x_3 = \text{positive value},\ x_4 = x_3 b_0^{*},$$

$$x_5 = b_0^{*},\ x_6 = 0$$

$$H_{12}^{*} \rightarrow H_{23} : x_1 = A_0^{*},\ x_2 = A_1^{*},\ x_3 = \text{positive value},\ x_4 = B_0^{*} x_3,$$

$$x_5 = B_1^{*} x_3 + B_0^{*},\ x_6 = B_1^{*}$$

Two cases were considered for both examples.

In the first case, $\underset{\sim}{c}^{m}$ is fixed, and

$$w_{\ell m} = w_{um} = 0$$

$$U(\underset{\sim}{\phi}) = \max_{t_i \in [0,T]} |e_i(\underset{\sim}{\phi})|$$

In the second case, $\underset{\sim}{c}^{m}$ is varied, and

$$w_{\ell m} = w_{um} = w_m$$

$$U(\underset{\sim}{\phi}) = \max_{t_i \in [0,T]} [(|e_i(\underset{\sim}{\phi})|, -w_m e_{\ell m}, w_m e_{um})]$$

A 9th-order nuclear reactor system was chosen for one example, where a step input is considered so that the power level of the reactor system changes from 90 to 100 percent of the full power (See Bereznai 1971 and Section 3.4.7). T was equal to 10 seconds.

The results, shown in Table 5.8, indicate that the increase in order of the model did not produce any large improvement in $\check{U}$, the minimum value of U, and in this case a model increase is quite wasteful from the computing viewpoint. On the other hand, an improvement in the transient error at a slight expense on the steady-state error is obtained.

Another system considered was the 7th-order control system problem mentioned in Section 5.4. T was equal to 8 seconds though the responses shown in Figs. 5.5-5.7 were taken up to 20 seconds. c_∞^5 was equal to 0.11111. The results are summarized in Table 5.9.

5.5.5 Discussion

The results indicate that when c_∞^m is fixed increasing the order of the model does improve the transient errors, and it has been shown in Section 5.4.2 that for the third-order model both the 5-ripple and 6-ripple solutions satisfy the necessary minimax optimality conditions. It is interesting to note that in all the cases considered, the third-order model gives the best result corresponding to the same transient error and three different steady-state errors. Some of the optimal parameters when c_∞^m is fixed tend to have nearly zero real parts which may make the model oscillatory. Using appropriate parameter constraints (as indicated in an earlier section) satisfactory results can be obtained which would guarantee a minimum damping of the model for a step input.

TABLE 5.8

RESULTS FOR NUCLEAR REACTOR SYSTEM MODELLING

Case	Model	1000 $\check{U}$	1000 max$[-e_{\ell\infty}, e_{u\infty}]$
c_∞^m fixed at c_∞^s	H_{02}	2.9234	0
	H_{12}	2.7018	0
	H_{23}	2.4040	0
c_∞^m varied, $w_\infty = 1$, $S_{\ell\infty}=S_{u\infty}=c_\infty^s$	H_{23}	1.2167	1.2166

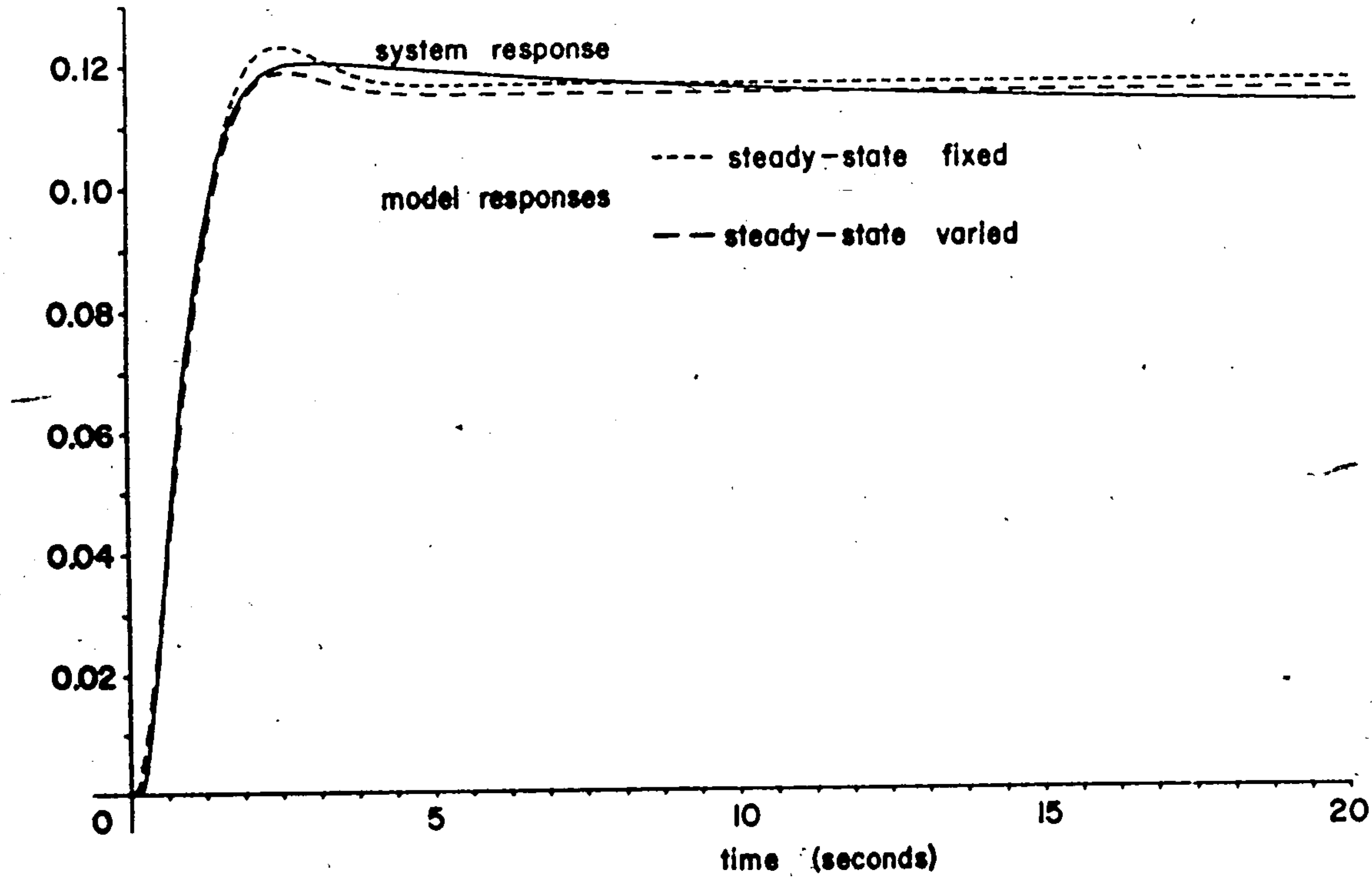


Fig. 5.5(a) Seventh-order system modelling example. Optimal responses for a second-order model with no zeros.

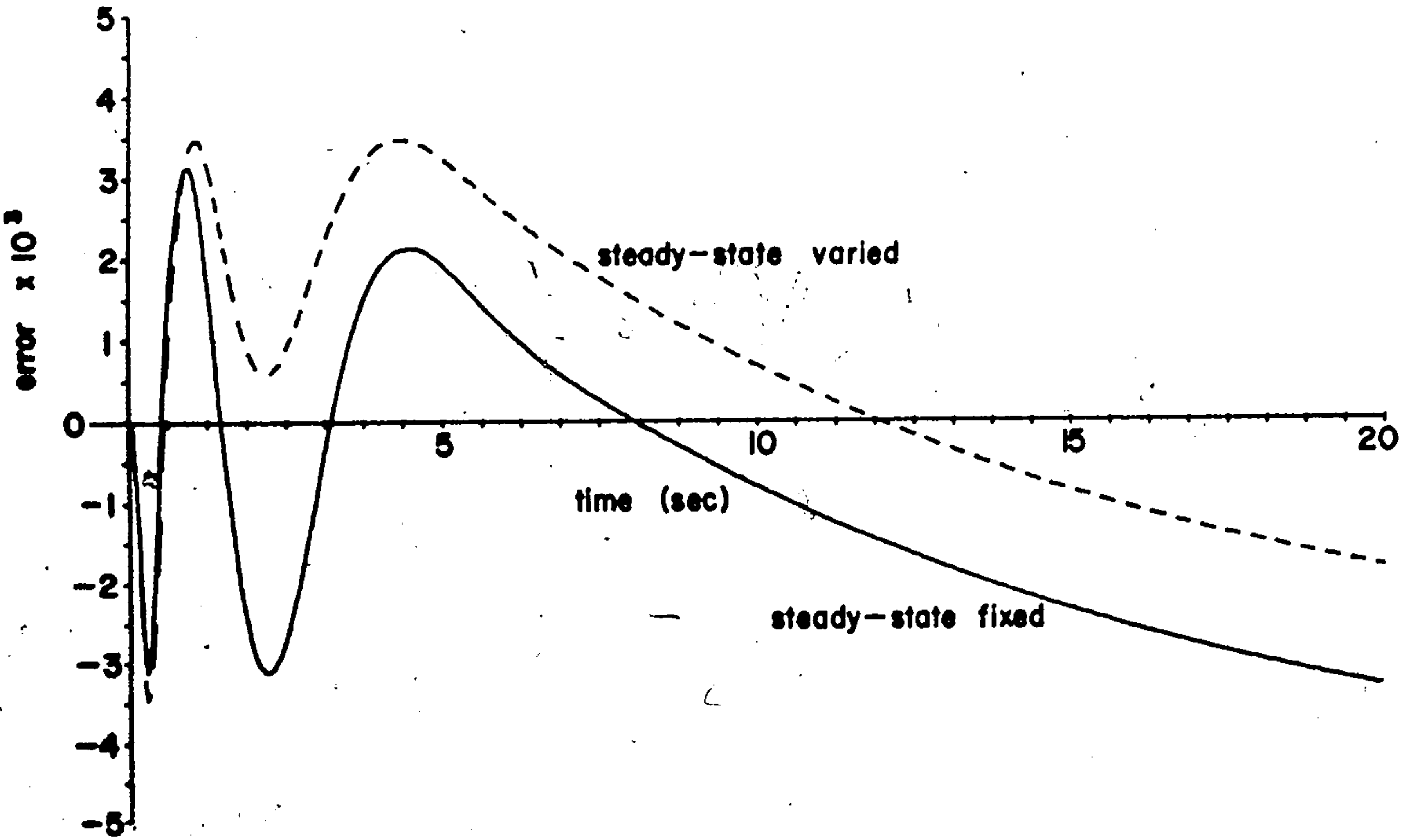


Fig. 5.5(b) Seventh-order system modelling example. Optimal error curves for a second-order model with no zeros.

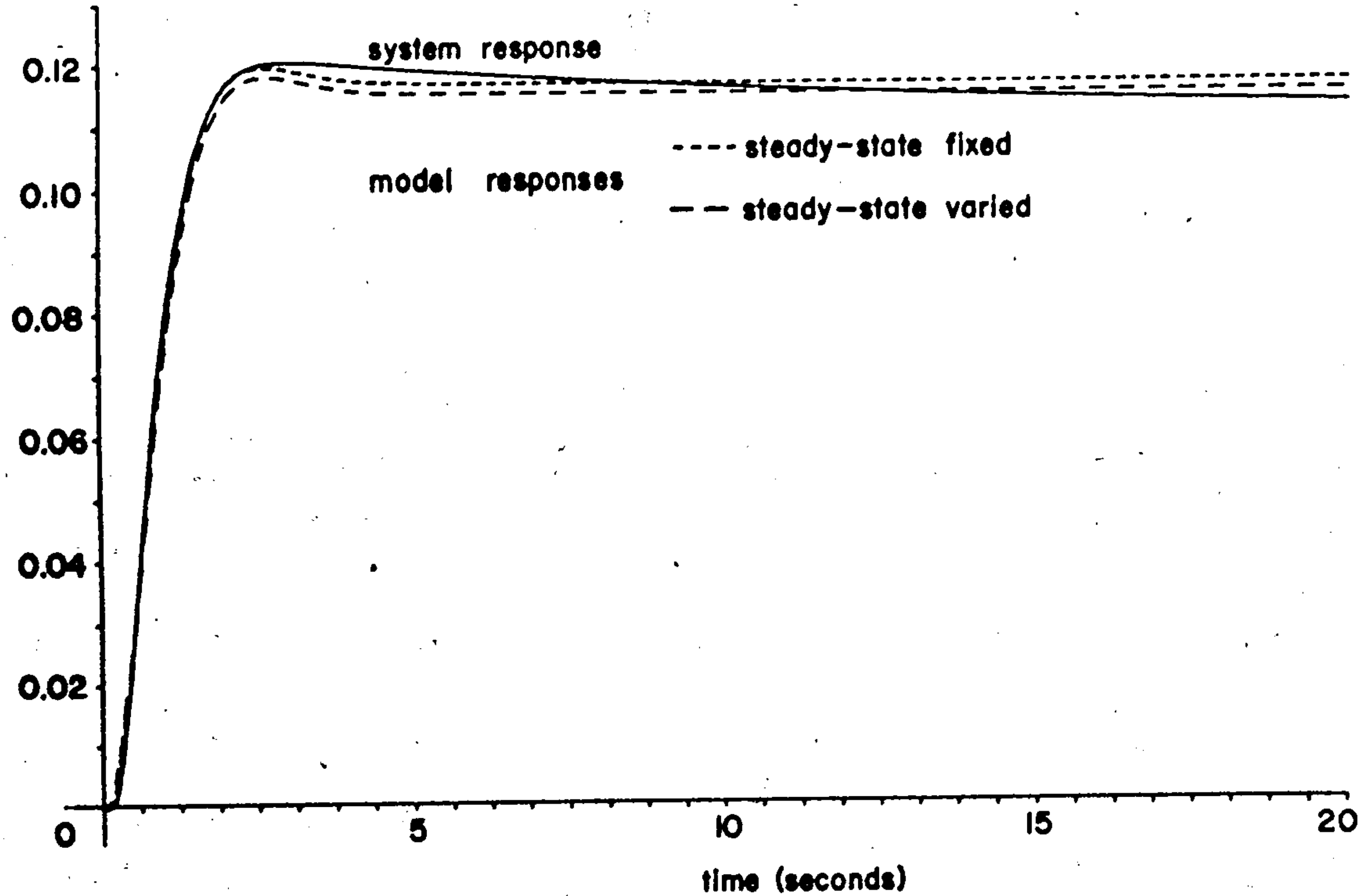


Fig. 5.6(a) Seventh-order system modelling example. Optimal responses for a second-order model with one zero.

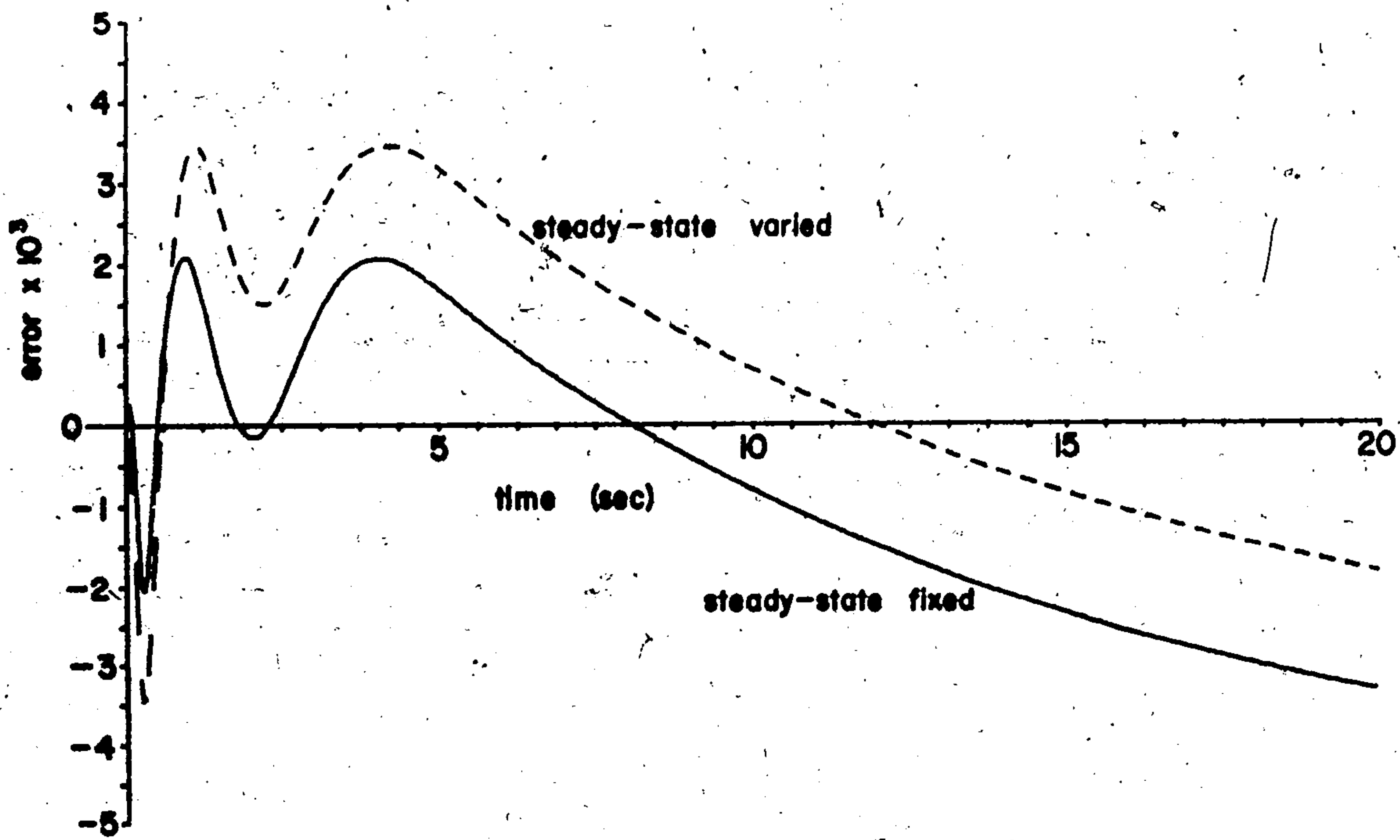


Fig. 5.6(b) Seventh-order system modelling example. Optimal error curves for a second-order model with one zero.

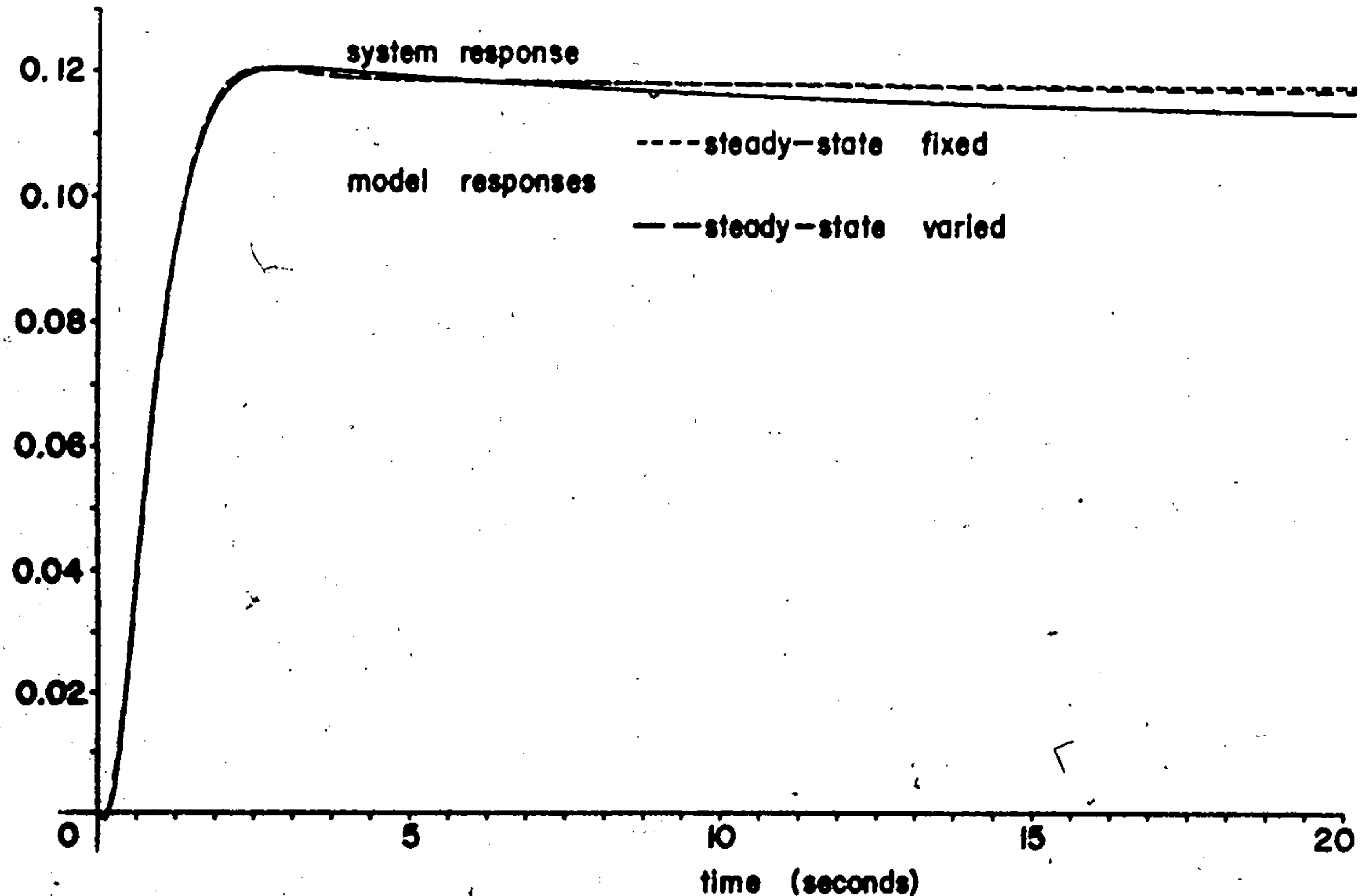


Fig. 5.7(a) Seventh-order system modelling example. Optimal responses for a third-order model with two zeros.

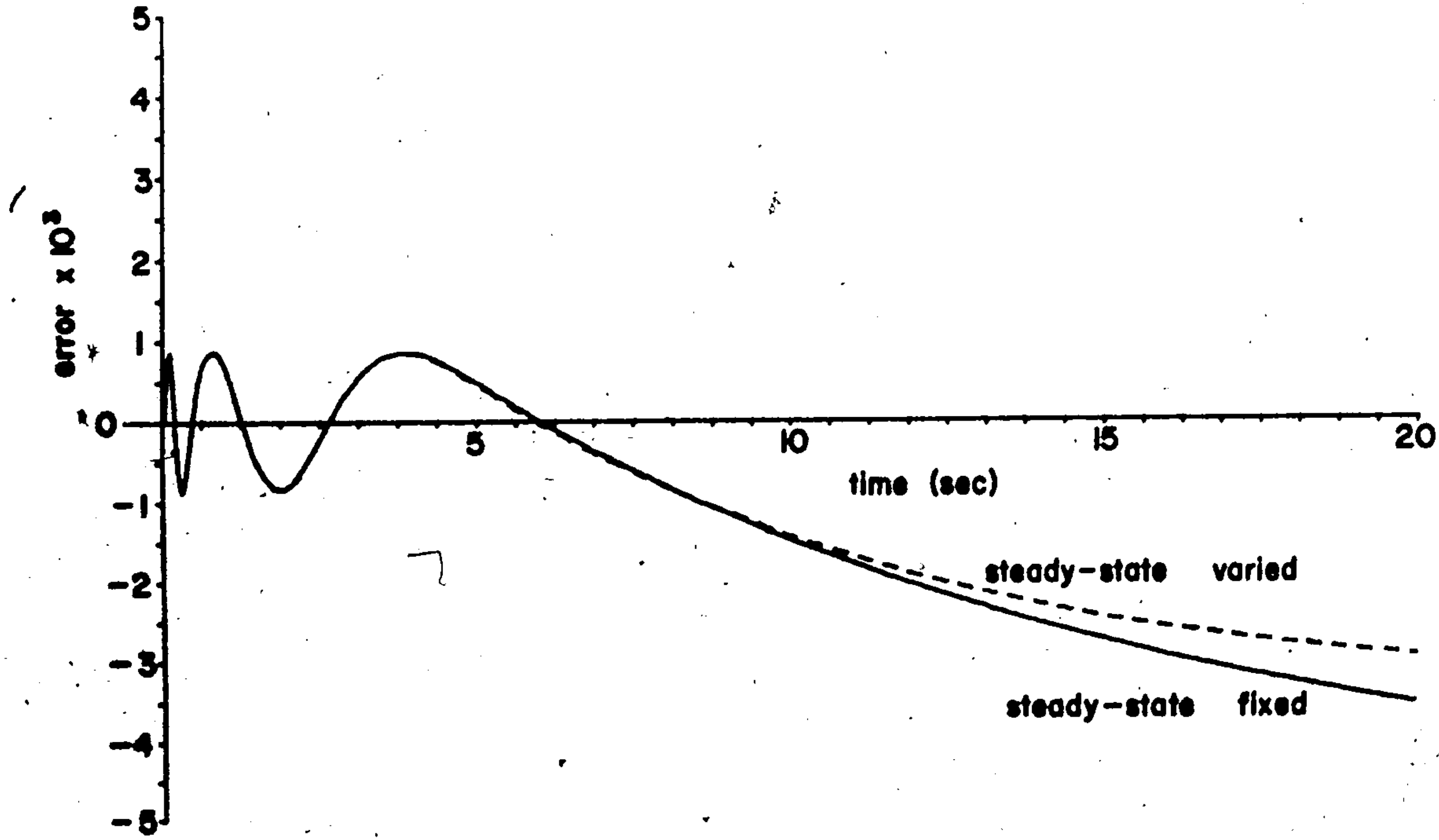


Fig. 5.7(b) Seventh-order system modelling example. Optimal error curves for a third-order model with two zeros.

TABLE 5.9

RESULTS FOR SEVENTH-ORDER SYSTEM MODELLING

Case	Model	1000 $\check{U}$	1000 max $[-e_{\ell\infty}, e_{u\infty}]$	Fig.
	H_{02}	3.7635		5.5
c_{∞}^{m} fixed at	H_{12}	2.4872		5.6
c_{i}^{s} for t_i=T	H_{23}			
	(6 ripple)	1.0207		5.7
	(5 ripple)	1.2140		-
c_{∞}^{m} varied	H_{02}	4.1656	4.1656	5.5
w_{∞} = 1	H_{12}	4.1582	4.1582	5.6
$S_{\ell\infty}$=$S_{u\infty}$=c_{∞}^{s}	H_{23}	1.0201	0.91785	5.7
c_{∞}^{m} varied	H_{02}	7.7657	7.6945 x 10^{-6}	-
w_{∞} = 10^{6}	H_{12}	7.8624	0.	-
$S_{\ell\infty}$=0.11061				
	H_{23}	1.0201	9.8483 x 10^{-7}	-
$S_{u\infty}$=0.11161				

5.6 Conclusions

The lower-order modelling of high-order systems for minimax objectives has been considered in detail, and the grazor search method has been critically compared with efficient minimization methods for least pth objectives. The grazor search method is very reliable, and the Fletcher method has been observed to be both reliable and efficient. The ideas proposed in this chapter make it possible to automate the modelling procedure, and with the availability of efficient optimization techniques, on-line system modelling and control is entirely feasible. The suggested procedures can be effectively used to get desired optimal models in the minimax sense within user-specified computing times and error allowances.

CHAPTER VI

DISCUSSION AND CONCLUSIONS

The thesis covers the areas of minimax approximation methods as applied to electrical network design and system modelling in great detail. A reliable algorithm has been proposed and applied to a variety of practical minimax design problems. The method has been critically compared with existing methods for efficiency and reliability, and works very well on most of the problems considered. The philosophy of system modelling is discussed at length, including various techniques involved in implementing the models. Automated modelling and design of high-order systems is shown to be feasible, and the present state of minimax circuit design is considered in detail.

The new ideas presented in the thesis have been verified and used in computer-aided design of a variety of electrical networks subject to different objectives and various constraint specifications. Filters can now easily be designed to meet upper and lower response specifications at predetermined frequencies, within reasonable computing time and desired accuracy. The choice of a circuit model and objective function are as important as the choice of a reliable and efficient optimization technique to give optimal model parameters. If suitable optimization techniques or modelling procedures do not exist for a particular system, the designer is confronted with the task of improving the modelling technique and developing an efficient algorithm to evolve a realistic design. This involves a great deal of system experience and expertise in the state of the art methods of computer-aided design.

The contributions of this work may be listed as follows.

(1) A new method called the grazor search algorithm has been proposed for minimax objectives. This method has been tested extensively on a number of problems including electrical network design and system modelling.

(2) A practical way of accommodating constraints in the minimax optimization problem has been proposed and applied to some problems.

(3) Methods for investigating a solution for minimax optimality have been proposed and used to test the optimality conditions on a variety of design problems.

(4) The grazor search method has been critically compared on lower-order minimax modelling of a high-order system with three efficient methods.

(5) Some ideas have been presented for automated system modelling, by means of which the order of the models can be increased in an automated fashion whenever certain criteria are satisfied, and optimality conditions can be directly implemented on the computer. Suitable transient and steady-state constraints can also be taken into account. The proposed approach makes it feasible to automate on-line modelling.

(6) The grazor search method and the method for investigating minimax optimality conditions have been programmed on a digital computer and user-oriented computer program packages have been developed.

It is felt that replacing the present linear search by a more efficient search technique will improve the efficiency of the grazor search algorithm. Further, the concept of automated modelling could be extended to include automated control so that it may be applicable to on-line system modelling and control.

APPENDIX A

GRAZOR SEARCH PROGRAM FOR MINIMAX OPTIMIZATION

A.1 Introduction

The grazor search program is a package of subroutines that optimizes the designable parameters of networks or systems to meet minimax objectives. Full details of the method, including mathematical flow charts and a discussion of computational experience, have already been covered in Chapters III, IV and V. A computer program written in Fortran (Version 2.3 and Scope Version 3.3 for the CDC 6400 computer) is listed at the end of this Appendix.

A.2 Nomenclature

The following is a list of some of the arguments and important variables of the grazor search package as indicated in the flow charts of Figs. 3.2-3.4.

α scale factor for determining the magnitude of the parameter step to be taken at the end of linear program

α_o initial specified value of α, previous value of α which gave a satisfactory improvement

$\check{\alpha}$ minimum allowable α

β reduction factor for α

γ^* factor of the step $\Delta\phi^o$ which gives the best new point, when starting from ϕ^o

ε	number of discrete maxima under consideration, k_r, is increased by one (if $k_r \leq n_r-1$) or set equal to one (if $k_r = n_r$) if the improvement of the objective function at the new best point as compared to the value at the previous point is less than this quantity
ε'	main program is eventually terminated if the improvement of objective function at the new best point as compared to the value at the previous point is repeatedly less than this quantity
η	specified factor of the initial interval of linear search which determines the final resolution between two internal points of the search
$\underset{\sim}{\phi}$	current point
$\underset{\sim}{\phi}^o$	starting point, current best point
$\Delta\underset{\sim}{\phi}^o$	increment from $\underset{\sim}{\phi}^o$ which gives the first improved point obtained in each iteration on entering the linear search
ψ_i	ith sample point
$\hat{\psi}_i$	sample points corresponding to the $\hat{y}_i$
ψ_{oi}	sample points corresponding to the y_{oi}
DERIV	logical variable; if .TRUE. the $\underset{\sim}{\nabla} y_i$ are calculated, otherwise they are not calculated
j_i	identifies the ith highest of the y_{oj}
k	dimensionality of parameter space
k_r	number of local discrete maxima $\hat{y}_i$ under consideration
n	number of sample points ψ_i
n_r	available number of discrete local maxima $\hat{y}_i$
U_ϕ	value of the objective function at $\underset{\sim}{\phi}$

$U_{\phi o}$	value of the objective function at $\underset{\sim}{\phi}^o$
y_i	function value at ϕ_i for a given $\underset{\sim}{\phi}$
$\hat{y}_i$	ith highest discrete local maximum
∇y_i	gradient of y_i with respect to $\underset{\sim}{\phi}$
$\underset{\sim}{y}_{oj}$	discrete local maxima implied by the y_i
TERM	logical variable, initially set to .FALSE., is reset to .TRUE. only if there are failures or improvements in objective function value less than ε' after considering values of k_r from 1 to n_r in one complete cycle.

A.3 Program Description

The user may call the package from his own program as follows.

```
CALL GRAZOR (ALPHAO,ALPMIN,BETA,EPS,EPS1,ETA,PHO,PSI,K,KR,N,
NR,UPHO,TERM)
```

The variables in the argument list are:

FORTRAN Name	Variable
ALPHAO	α_o
ALPMIN	$\check{\alpha}$
BETA	β
EPS	ε
EPS1	ε'
ETA	η
PHO	$\underset{\sim}{\phi}^o$
PSI	ψ_i

K	k
KR	k_r
N	n
NR	n_r
UPHO	$U_{\phi o}$
TERM	TERM

The input variables are α_o, $\check{\alpha}$, β, ε, ε', η, $\underset{\sim}{\phi}^o$, ϕ_i, k, k_r and n while the output variables are α_o, $\underset{\sim}{\phi}^o$, k_r, n_r, $U_{\phi o}$ and TERM.

It was convenient to place the following user-specified variables in

COMMON/GRZR/NCOUNT, IPRINT, UNIT, IOPT, IDATA

NCOUNT number of function evaluations at any stage of the iterative cycle of grazor, is initially set to zero by the user.

IPRINT logical variable which, if .TRUE., enables all intermediate and final results to be printed out, and no print-outs otherwise.

UNIT integer variable specifying the data set reference number of the output unit.

IOPT integer variable denoting the number of times grazor search package was called by the user, is set to zero initially by the user.

IDATA logical variable which, if .TRUE., enables the input data to be printed out; otherwise not.

Fig. A.1 shows a typical main program for calling the package and the form of a typical analysis program while Fig. A.2 shows typical print-outs of the package.

```
C     ..............................................................
C     .A TYPICAL MAIN PROGRAM FOR THE GRAZOR SEARCH ALGORITHM
C      FOLLOWS-------
      DIMENSION PHO(15),PSI(11)
      LOGICAL TERM,IPRINT,IDATA
      INTEGER UNIT
      COMMON/GRZR/NCOUNT,IPRINT,UNIT,IOPT,IDATA
C     TYPICAL INPUT VALUES FOLLOW
      ALPHAO=1.
      ALPMIN=1.0E-06
      BETA=10.
      ETA=0.01
      KR=1
      NCOUNT=0
      IOPT=0
C
C     INPUT VALUES FOR THE SPECIFIC PROBLEM FOLLOW
      IPRINT=.TRUE.
      IDATA=.TRUE.
      UNIT=6
      EPS=1.0E-03
      EPS1=1.0E-06
      K=2
      N=11
      PHO(1)=1.
      PHO(2)=3.
      PSI(1)=0.5
      DO 1 I=2,N
1     PSI(I)=PSI(I-1)+0.1
C
C     MINIMAX OPTIMIZATION STARTS
      DO 2 I=1,100
      CALL GRAZOR(ALPHAO,ALPMIN,BETA,EPS,EPS1,ETA,PHO,PSI,K,KR,
     1N,NR,UPHO,TERM)
C     ..............................................................
      IF(TERM) GO TO 3
2     CONTINUE
3     STOP
      END
C
C
C     A TYPICAL ANALYSIS PROGRAM FOR GRAZOR SEARCH ALGORITHM
C     FOLLOWS-------
      SUBROUTINE ANAL (PHO,F,DERIV,K,Y,GRADY)
      DIMENSION PHO(1),GRADY(1)
      LOGICAL DERIV
C     THE VALUE OF Y AT A SINGLE SAMPLE POINT F IS CALCULATED
C     HERE
      IF(.NOT.DERIV) RETURN
C     THE DERIVATIVES GRADY(1),GRADY(2),....,GRADY(K) OF THE
C     FUNCTION Y WITH RESPECT TO PARAMETERS PHO(1),PHO(2),....
C     PHO(K) ARE CALCULATED HERE
      RETURN
      END
```

Fig. A.1 Typical main program and analysis program for the grazor search package

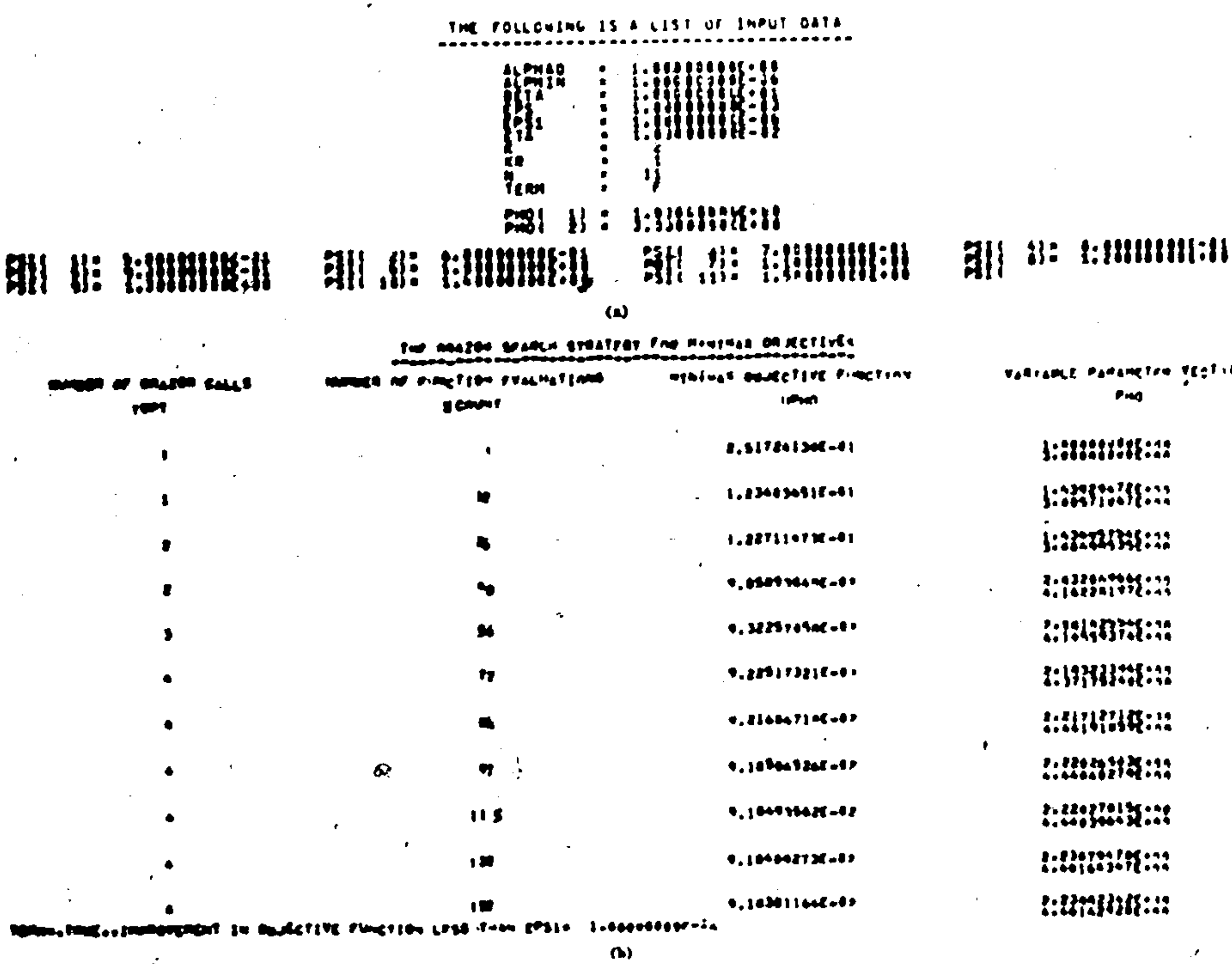


Fig. A.2 (a) Typical printout if IDATA is .TRUE. (b) Typical printout if IPRINT is .TRUE.

A.4 Subprograms

The subroutine ANAL($\underset{\sim}{\phi}$, ψ_i, DERIV, k, y_i, $\underset{\sim}{\nabla}$ y_i) is a user-supplied analysis program to evaluate y_i and/or $\underset{\sim}{\nabla}$ y_i at a given point $\underset{\sim}{\phi}$. If DERIV is .TRUE., the $\underset{\sim}{\nabla} y_i$ are calculated, otherwise they are not calculated.

The following subroutine need not be written by the user, but is part of the grazor search package. The function subprogram Y($\underset{\sim}{\phi}$,ψ_i,k) calculates the y_i corresponding to the point $\underset{\sim}{\phi}$ by calling ANAL. The subroutine LOCATE ($\underset{\sim}{\phi}$,ψ_i,k,n,U_ϕ) evaluates the objective function U_ϕ by calling Y($\underset{\sim}{\phi}$,ψ_i,k) for i = 1,2,...,n. The grazor search package also uses a linear program solving routine called SIMPLE (see Subroutine SIMPLE), which is a modified version of a program documented with the SHARE Distribution Agency, and written by R.J. Clasen (Reference No. SDA 3384). Section A.7 includes a listing of this subroutine.

A.5 Comments

As it stands the package has been programmed to handle up to 15 variable parameters and 15 ripples. The choice of input parameters including scale factors may be critical to efficiency of the algorithm, and the grazor search strategy should be well-understood before the user attempts to use this program.

This program was run and tested on a CDC 6400 computer. The Fortran deck consists of 901 cards which includes detailed comments at appropriate places. The package requires roughly 20,000 octal units of computer memory.

A.6 Discussion

The grazor search algorithm has been programmed in such a way that it allows a certain amount of flexibility to the user. Thus, when GRAZOR is called once, one complete iterative step of the algorithm results, and by introducing GRAZOR in a DO loop, the user has the complete freedom to make his own decision about termination subject to his own convergence criteria, or printing out intermediate results according to a preferred format, or branching out to another optimization package if desired. Appropriate diagnostic messages are provided in the program wherever necessary.

As this is a gradient strategy, it is important that the gradients as evaluated by the analysis program are correct.

A.7 Grazor Search Fortran Program Listing

```
C               GRAZOR SEARCH PROGRAM FOR MINIMAX OBJECTIVES
C               ------------------------------------------
C
C     AUTHORS   J.W.BANDLER AND T.V.SRINIVASAN,COMMUNICATIONS RESEARCH
C               LABORATORY AND DEPARTMENT OF ELECTRICAL ENGINEERING,
C               MCMASTER UNIVERSITY,HAMILTON,ONTARIO,CANADA
C
C
C     THE GRAZOR SEARCH ALGORITHM HAS TO BE PLACED IN A DO LOOP BY THE
C     USER SO THAT THE ALGORITHM MAY BE CALLED AS MANY TIMES AS IS
C     NECESSARY TO GET SATISFACTORY IMPROVEMENTS IN THE OBJECTIVE
C     FUNCTION
C
C     ..................................................................
C     A TYPICAL MAIN PROGRAM FOR GRAZOR SEARCH ALGORITHM FOLLOWS--------
C     DIMENSION PHO(15),PSI(11)
C     LOGICAL TERM,IPRINT,IDATA
C     INTEGER UNIT
C     COMMON/GRZR/NCOUNT,IPRINT,UNIT,IOPT,IDATA
C     TYPICAL INPUT VALUES FOLLOW
C     ALPHAO=1.
C     ALPMIN=1.0E-06
C     BETA=10.
C     ETA=0.01
C     KR=1
C     NCOUNT=0
C     IOPT=0
C
C     INPUT VALUES FOR THE SPECIFIC PROBLEM FOLLOW
C     IPRINT=.TRUE.
C     IDATA=.TRUE.
C     UNIT=6
C     EPS=1.0E-03
C     EPS1=1.0E-06
C     K=2
C     N=11
C     PHO(1)=1.
C     PHO(2)=3.
C     PSI(1)=0.5
C     DO 1 I=2,N
C   1 PSI(I)=PSI(I-1)+0.1
C
C     MINIMAX OPTIMIZATION STARTS
C     DO 2 I=1,100
C     CALL GRAZOR(ALPHAO,ALPMIN,BETA,EPS,EPS1,ETA,PHO,PSI,K,KR,N,NR,UPHO
C    1,TERM)
C     IF(TERM) GO TO 3
C   2 CONTINUE
C   3 STOP
C     END
C
C
C
C     ..................................................................
C     A TYPICAL ANALYSIS PROGRAM FOR GRAZOR SEARCH ALGORITHM FOLLOWS----
C     SUBROUTINE ANAL (PHO,F,DERIV,K,Y,GRADY)
C     DIMENSION PHO(1),GRADY(1)
C     LOGICAL DERIV
```

```
C     THE VALUE OF Y AT A GIVEN SAMPLE POINT F IS CALCULATED HERE
C     IF(.NOT.DERIV) RETURN
C     THE DERIVATIVES GRADY(1),GRADY(2),.....,GRADY(K) OF THE FUNCTION Y
C     WITH RESPECT TO PARAMETERS PHO(1),PHO(2),.....,PHO(K) ARE
C     CALCULATED HERE
C     RETURN
C     END
C
C
C
C     .................................................................
      SUBROUTINE GRAZOR (ALPHAO,ALPMIN,BETA,EPS,EPS1,ETA,PHO,PSI,K,KR,N,
     1NR,UPHO,TERM)
C
C
C     THE USER HAS TO SPECIFY VALUES FOR ALPHAO,ALPMIN,BETA,EPS,EPS1,ETA
C     ,PHO,PSI,K,KR,N
C
C     STARTING VALUES ---------
C     ALPHAO=1.
C     BETA=10.
C     KR=1
C     SUGGESTED STARTING VALUES--------
C     ALPMIN=1.0E-06
C     ETA=0.01
C     EPS1,THE MINIMUM IMPROVEMENT IN THE OBJECTIVE FUNCTION BETWEEN
C     SUCCESSIVE ITERATIONS,MUST BE SPECIFIED BY THE USER
C     EPS=EPS1*1000.
C
C
C     THE FOLLOWING COMMON STATEMENT IS TO BE SPECIFIED BY THE USER
C     COMMON/GRZR/NCOUNT,IPRINT,UNIT,IOPT,IDATA
C     NCOUNT=NUMBER OF FUNCTION EVALUATIONS AT ANY STAGE OF THE
C     ITERATIVE CYCLE OF GRAZOR
C     NCOUNT IS INITIALLY SET TO ZERO BY THE USER
C     IOPT CORRESPONDS TO AN ITERATIVE CYCLE OF THE GRAZOR SEARCH ALGORI
C     THM,AND IS THE NUMBER OF TIMES OPTIMIZATION PACKAGE GRAZOR HAS
C     BEEN CALLED.IOPT IS INITIALLY SET TO ZERO BY THE USER
C     IF IPRINT IS .TRUE. ALL INTERMEDIATE AND FINAL RESULTS ARE TO BE
C     PRINTED OUT,OTHERWISE THERE ARE NO PRINT-OUTS
C     UNIT IS AN INTEGER VARIABLE SPECIFYING THE DATA SET REFERENCE
C     NUMBER OF THE OUTPUT UNIT
C     IF IDATA IS .TRUE. THE INPUT DATA IS PRINTED OUT,OTHERWISE NOT
C     THE USER HAS TO SPECIFY VALUES FOR IPRINT,UNIT,IDATA
C
C
C     THE VARIABLES PSI AND PHO HAVE TO BE DIMENSIONED IN THE CALLING PR
C     OGRAM CORRESPONDING TO MAXIMUM VALUES OF N,AND K(=15),RESPECTIVELY
C     THE USER HAS TO INDICATE IN HIS MAIN PROGRAM THAT TERM,IPRINT,
C     IDATA ARE LOGICAL VARIABLES AND THAT UNIT IS AN INTEGER VARIABLE
C     IF TERM IS .TRUE. AT THE END OF AN ITERATIVE CYCLE OF GRAZOR.
C     THE USER HAS TO DECREASE THE VALUES OF ALPMIN AND ETA BEFORE
C     GRAZOR CAN BE CALLED AGAIN IN THE MAIN PROGRAM
C     THE USER HAS TO FURNISH SUBROUTINE ANAL FOR IMPLEMENTING THE
C     GRAZOR SEARCH STRATEGY
C
C
```

```
C     THE FOLLOWING IS A BRIEF SUMMARY OF THE VARIABLES IN GRAZOR------
C     PHO= THE PARAMETER VECTOR,IT IS EITHER THE STARTING POINT OR THE
C     CURRENT BEST POINT
C     PHI= CURRENT PARAMETER VECTOR
C     K=NUMBER OF PARAMETERS PHO
C     PSI= VECTOR OF SAMPLE POINTS
C     N= NUMBER OF SAMPLE POINTS PSI
C     UPHO= OBJECTIVE FUNCTION AT PHO
C     UPHI= OBJECTIVE FUNCTION AT PHI
C     YMAX = VECTOR CONSISTING OF THE LOCAL DISCRETE MAXIMA IMPLIED BY
C     THE FUNCTIONS Y ,ARRANGED IN DECREASING MAGNITUDE,OVER N SAMPLE
C     POINTS PSI
C     PSIMAX= VECTOR OF SAMPLE POINTS CORRESPONDING TO THE VECTOR YMAX
C     NR= NUMBER OF DISCRETE LOCAL MAXIMA YMAX
C     KR= NUMBER OF DISCRETE LOCAL MAXIMA YMAX UNDER CONSIDERATION,KR IS
C     LESS THAN OR EQUAL TO NR
C     GRAD = MATRIX OF FIRST DERIVATIVES OF  VECTOR YMAX WITH RESPECT
C     TO THE PARAMETERS PHO
C     TERM= LOGICAL VARIABLE WHICH,IF TRUE,INDICATES THE CONVERGENCE OF
C     THE GRAZOR SEARCH ALGORITHM
C
C
C     THE DIMENSION OF SUBSCRIPTED VARIABLES IN GRAZOR CORRESPOND TO
C     MAXIMUM VALUES OF K=15 AND NR=15
C     THE SUBSCRIPTED VARIABLES DUMMY,PHI,DELPHI,DELPHN,DELP ARE
C     DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE OF K=15
C     THE SUBSCRIPTED VARIABLES YMAX,PSIMAX ARE DIMENSIONED
C     CORRESPONDING TO A MAXIMUM VALUE OF NR=15
C     MATRIX GRAD IS DIMENSIONED CORRESPONDING TO MAXIMUM VALUES OF
C     NR=15 AND K=15
C
C
C     THE USER HAS TO SUPPLY AN ANALYSIS PROGRAM AND THE FOLLOWING IS A
C     BRIEF DESCRIPTION OF ITS ARGUMENTS
C     SUBROUTINE ANAL (PHO,F,DERIV,K,Y,GRADY) CALCULATES THE VALUE OF
C     FUNCTION Y AND ITS FIRST PARTIAL DERIVATIVES GRADY(1),GRADY(2),...
C     ...,GRADY(K) WITH RESPECT TO THE PARAMETERS PHO(1),PHO(2),........
C     ...,PHO(K) FOR A GIVEN SAMPLE POINT F
C     PHO AND GRADY ARE TO BE VARIABLE-DIMENSIONED IN ANAL,OR
C     DIMENSIONED CORRESPONDING TO THE MAXIMUM VALUE FOR K(=15)
C     DERIV=LOGICAL VARIABLE WHICH,IF TRUE,ALLOWS THE GRADY TO BE
C     EVALUATED, OTHERWISE GRADY ARE NOT EVALUATED
C
C
      DIMENSION PHO(1), PSI(1), DUMMY(15), PHI(15), YMAX(15), PSIMAX(15)
     1, GRAD(15,15), DELPHI(15), DELPHN(15), DELP(15), X(31), A(16,31),
     2B(16), C(31), KO(6), PS(16), JH(16), XX(16), YY(16), PE(16), E(16,
     316)
      LOGICAL TERM,IPRINT,IDATA
      INTEGER UNIT
      COMMON /GRZR/ NCOUNT,IPRINT,UNIT,IOPT,IDATA
      IOPT=IOPT+1
      IF (NCOUNT.EQ.0) TERM=.FALSE.
      IF (TERM) GO TO 32
      ALPHA=ALPHAO
      ALPHAT=ALPHAO
      ICLOCK=0
```

```
      ISTOP=0
1     CALL SELEC (PHO,PSI,PSIMAX,K,N,NR,YMAX)
      UPHO=YMAX(1)
      NCOUNT=NCOUNT+1
      IF (NCOUNT.GE.2) GO TO 2
      IF (IDATA) WRITE (UNIT,38) ALPHAO,ALPMIN,BETA,EPS,EPSI,ETA,K,KR,N,
     1TERM,(I,PHO(I),I=1,K)
      IF (IDATA) WRITE (UNIT,39) (I,PSI(I),I=1,N)
      IF (IPRINT) WRITE (UNIT,34) IOPT,NCOUNT,UPHO,(PHO(I),I=1,K)
2     IF (ICLOCK.GT.1) GO TO 3
      IF (KR.NE.1) GO TO 4
3     KR=1
4     DO 6 L=1,KR
      CALL ANAL (PHO,PSIMAX(L),.TRUE.,K,YMAX(L),DUMMY)
      DO 5 I=1,K
      GRAD(L,I)=DUMMY(I)
5     CONTINUE
6     CONTINUE
      IF (KR.EQ.1) GO TO 22
      KR1=KR+1
      KR2=KR+2
      KR3=KR1+KR
      DO 9 I=1,KR
      DO 8 J=1,KR
      IF (I.GT.J) GO TO 8
      A(I,J)=0.
      DO 7 MM=1,K
      A(I,J)=A(I,J)-GRAD(I,MM)*GRAD(J,MM)
7     CONTINUE
      A(J,I)=A(I,J)
8     CONTINUE
9     CONTINUE
      DO 10 I=1,KR
      A(I,KR1)=1.0
10    CONTINUE
      DO 12 I=1,KR
      DO 11 J=KR2,KR3
      A(I,J)=0.0
      IF (J.EQ.(I+KR1)) A(I,J)=1.0
11    CONTINUE
12    CONTINUE
      DO 13 J=1,KR
      A(KR1,J)=1.0
13    CONTINUE
      DO 14 J=KR1,KR3
      A(KR1,J)=0.0
14    CONTINUE
      DO 15 I=1,KR
      B(I)=0.0
15    CONTINUE
      B(KR1)=1.0
      DO 16 I=1,KR3
      C(I)=0.0
      IF (I.EQ.KR1) C(I)=-1.0
16    CONTINUE
C
C
```

```
C       SUBROUTINE SIMPLE IS NOW GOING TO BE CALLED,ANY ALTERNATIVE
C       CHOICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT
C       PERFORMS THE FOLLOWING OPERATION------
C       SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF
C       MINIMIZING C*X SUBJECT TO A*X=B,WHERE X,C,B ARE VECTORS OF LENGTH
C       KR3,KR3,KR1 RESPECTIVELY,AND A IS A MATRIX OF SIZE KR1*KR3
C
C       SUBROUTINE SIMPLE ATTACHED TO THIS PACKAGE IS A MODIFIED VERSION
C       OF A PROGRAM AVAILABLE WITH SHARE DISTRIBUTION AGENCY,REFERENCE
C       NUMBER SDA 3384 AND WRITTEN BY R.J.CLASEN
C       THE MODIFIED VERSION IS IN THE MCMASTER UNIVERSITY DATA PROCESSING
C       AND COMPUTING CENTRE LIBRARY,INFORMATION SHEET MILIS 5.7.130
C
C       NA AND IFLAG ARE TO BE SPECIFIED BEFORE CALLING SIMPLE
C       IFLAG IS SET EQUAL TO ZERO
C       NA IS THE FIRST DIMENSION OF THE ARRAY A AND IS SET EQUAL TO THE
C       MAXIMUM VALUE OF NR+1(=16)
C       X IS THE VECTOR OF DIMENSION 2*NA-1
C       THE FOLLOWING SUBSCRIPTED VARIABLES ARE PART OF THE ARGUMENT LIST
C       OF SIMPLE AND ARE TEMPORARY STORAGE SPACES TO BE DIMENSIONED IN
C       THE CALLING PROGRAM (GRAZOR)
C       PS,JH,XX,YY AND PE ARE TEMPORARY STORAGE VECTORS OF DIMENSION NA
C       E IS A TEMPORARY STORAGE MATRIX OF DIMENSION (NA,NA*2-1)
C       KO IS A VECTOR OF LENGTH 6,UPON COMPLETION OF THE EXECUTION OF
C       SIMPLE, KO(1)=0 IF THE LINEAR PROGRAMMING PROBLEM WAS FEASIBLE.
C       THE SOLUTION LIES IN X(J),J=1,KR3
C
        IFLAG=0
        NA=16
        CALL SIMPLE (IFLAG,KR1,KR3,A,B,C,KO,X,PS,JH,XX,YY,PE,E,NA)
C
        DO 18 J=1,K
        DELPHI(J)=0.0
        DO 17 I=1,KR
        DELPHI(J)=DELPHI(J)-X(I)*GRAD(I,J)
17      CONTINUE
18      CONTINUE
C
C       THE INCREMENTAL PARAMETER STEP DELPHI IS NORMALIZED TO UNIT
C       LENGTH BY SUBROUTINE NORM
C
19      CALL NORM (K,DELPHI,DELPHN)
C
C       THE LINEAR SEARCH BEGINS
C       ALPHA IS A SCALE FACTOR FOR DETERMINING THE MAGNITUDE OF THE
C       NORMALIZED STEP DELPHN TO BE TAKEN FOR THE LINEAR SEARCH
C       ALPHAO IS THE INITIALLY SPECIFIED VALUE OF ALPHA OR THE PREVIOUS
C       VALUE OF ALPHA WHICH GAVE A SATISFACTORY IMPROVEMENT
C       ALPMIN IS THE MINIMUM ALLOWABLE ALPHA
        IF (ALPHA.LT.ALPMIN) ALPHA=ALPMIN
20      DO 21 I=1,K
        PHI(I)=PHO(I)+ALPHA*DELPHN(I)
21      CONTINUE
        GO TO 24
C       A STEP TAKEN IN THE NEGATIVE GRADIENT DIRECTION OF HIGHEST RIPPLE
22      DO 23 I=1,K
        DELPHI(I)=-GRAD(1,I)
```

```
23    CONTINUE
      GO TO 19
24    CALL LOCATE (PHI,PSI,K,N,UPHI)
      NCOUNT=NCOUNT+1
      IF (UPHI.LT.UPHO) GO TO 25
      IF (ALPHA.EQ.ALPHMIN) GO TO 30
C
C     ALPHA REDUCED BY FACTORS OF BETA
C
      ALPHA=ALPHA/BETA
      IF (ALPHA.LE.ALPHMIN) ALPHA=ALPHMIN
      GO TO 20
25    DO 26 I=1,K
      DELP(I)=ALPHA*DELPHN(I)
26    CONTINUE
C
C     DELP= INCREMENT FROM PHO WHICH GIVES THE FIRST IMPROVED POINT
C     OBTAINED ON ENTERING THE LINEAR SEARCH
C     ETA IS THE SPECIFIED FACTOR OF THE INITIAL INTERVAL OF LINEAR
C     SEARCH WHICH DETERMINES THE FINAL RESOLUTION BETWEEN TWO INTERNAL
C     POINTS OF THE SEARCH
C
      FINT=ETA
      CALL GOLDEN (GAMA,FINT,PHI,PHO,DELP,PSI,K,N,UPHI,UPHO)
C
C     TERM IS SET TO .TRUE. AND GRAZOR RETURNS TO THE CALLING PROGRAM IF
C     UPHO-UPHI IS REPEATEDLY LESS THAN EPSI
C
      IF ((UPHO-UPHI).LT.EPSI) GO TO 30
      ISTOP=0
      DO 27 I=1,K
      PHO(I)=PHI(I)
27    CONTINUE
      IF (IPRINT) WRITE (UNIT,37) IOPT,NCOUNT,UPHI,(PHO(I),I=1,K)
C
C     IF THE OBJECTIVE FUNCTION UPHI AT A NEW POINT PHI IS LESS THAN THE
C     VALUE UPHO AT THE PREVIOUS POINT PHO BY A VALUE GREATER THAN OR
C     EQUAL TO EPS,THE NEW POINT IS CONSIDERED A SATISFACTORY
C     IMPROVEMENT.IF NOT,KR IS INCREMENTED BY 1 (FOR KR LESS THAN OR
C     EQUAL TO NR-1) OR SET EQUAL TO 1 (FOR KR=NR)
C
      IF ((UPHO-UPHI).LT.EPS) GO TO 31
      UPHO=UPHI
      ALPHAO=ALPHA*GAMA
      GO TO 33
28    ICLOCK=ICLOCK+1
      IF (ISTOP.EQ.0) GO TO 1
      IF (ISTOP.LE.NR) GO TO 3
      TERM=.TRUE.
      WRITE (UNIT,39) EPSI
      GO TO 33
29    IF (KR.EQ.NR) GO TO 28
      KR=KR+1
      ICLOCK=1
      IF (ISTOP.EQ.0) GO TO 1
      GO TO 4
30    ALPHA=ALPHAT
```

```
      ISTOP=ISTOP+1
      GO TO 29
31    UPHO=UPHI
      ALPHA=ALPHAT
      GO TO 29
32    WRITE (UNIT,36)
33    RETURN
C
C
C
34    FORMAT (*1 */43X,* THE GRAZOR SEARCH STRATEGY FOR MINIMAX OBJECTIV
     1ES */43X,*-----------------------------------------------------*//6X,
     2*NUMBER OF GRAZOR CALLS*,8X,*NUMBER OF FUNCTION EVALUATIONS*,7X,*M
     3INIMAX OBJECTIVE FUNCTION*,10X,*VARIABLE PARAMETER VECTOR*//15X,*I
     4OPT*,30X,*NCOUNT*,30X,*UPHO*,32X,*PHO*////14X,I5,30X,I5,23X,E16.8,
     518X,F16.8/(111X,E16.8))
35    FORMAT (* TERM=.TRUE.,IMPROVEMENT IN OBJECTIVE FUNCTION LESS THAN
     1EPS1=*,E16.8)
36    FORMAT (* THE GRAZOR SEARCH ROUTINE CANNOT RESTART AS TERM IS EQUA
     1L TO .TRUE. FROM THE PREVIOUS ITERATION, AND CONVERGENCE CRITERION
     2 HAS BEEN*/* REACHED FOR A SPECIFIED EPS1.THE ONLY WAY TO RESTART
     3IS TO DECREASE THE VALUES OF ALPHIN AND ETA.  *)
37    FORMAT (//14X,I5,30X,I5,23X,E16.8,18X,E16.8/(111X,E16.8))
38    FORMAT (49X,* THE FOLLOWING IS A LIST OF INPUT DATA*/49X,*--------
     1-------------------------------*//55X,*ALPHAO      =*,E16.8/55X,*ALPH
     2IN     =*,E16.8/55X,*BETA       =*,E16.8/55X,*EPS        =*,E16.8/55X,*E
     3PS1      =*,E16.8/55X,*ETA        =*,E16.8/55X,*K          =*,I5/55X,*K
     4R        =*,I5/55X,*N          =*,I5/55X,*TERM       =*,L5//(55X,*PHO(*
     5,I3,*)  =*,E16.8))
39    FORMAT (/(8X,*PSI(*,I3,*)=*,E16.8,5X,*PSI(*,I3,*)=*,E16.8,5X,*PSI(
     1*,I3,*)=*,E16.8,5X,*PSI(*,I3,*)=*,E16.8))
      END
C
C
C
C
C     ******************************************************************
      SUBROUTINE SELEC (PHI,PSI,PSIMAX,K,N,NR,YMAX)
C
C
C     IN THIS SUBROUTINE THE RIPPLES OF THE FUNCTIONS Y AT A POINT PHI
C     OVER N SAMPLE POINTS PSI ARE LOCATED AND SORTED OUT IN DECREASING
C     MAGNITUDE
C     ITAG,PMAX,MAX ARE DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE
C     OF NR=15
C     MAX = DISCRETE LOCAL MAXIMA IMPLIED BY THE FUNCTIONS Y AT A POINT
C     PHI AS SAMPLING PROCEEDS FROM PSI(1) TO PSI(N)
C     PMAX= SAMPLE POINTS CORRESPONDING TO MAX
C
C
      DIMENSION PSI(1), PHI(1), YMAX(1), PSIMAX(1), ITAG(15), PMAX(15),
     1MAX(15)
      REAL MAX
      NR=1
      PMAX(1)=PSI(1)
      MAX(1)=Y(PHI,PSI(1),K)
      YZ=MAX(1)
```

```
      KD=1
      DO 4 I=2,N
      YX=Y(PHI,PSI(I),K)
      IF (YX-YZ) 1,1,2
1     KD=-1
      GO TO 3
2     YMAXT=YX
      LT=I
      IF (KD.EQ.-1) NR=NR+1
      KD=1
      PMAX(NR)=PSI(LT)
      MAX(NR)=YMAXT
3     YZ=YX
4     CONTINUE
C
C
      CALL TGSORT (MAX,ITAG,NR,1)
      DO 5 J=1,NR
      LD=ITAG(J)
      YMAX(J)=MAX(LD)
      PSIMAX(J)=PMAX(LD)
5     CONTINUE
      RETURN
      END
C
C
C
C
C     ................................................................
      SUBROUTINE GOLDEN (GAMA,ETA,PHI,PHO,DELP,PSI,K,N,UPHI,UPHO)
C
C
C     THIS SUBROUTINE USES THE GOLDEN SECTION SEARCH TO FIND THE GAMA
C     CORRESPONDING TO THE MINIMUM OF THE OBJECTIVE FUNCTION AT THE
C     POINT PHO+GAMA*DELP
C     PHIA,PHIB,PHIU ARE DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE
C     OF K=15
C
C
      COMMON /GRZR/ NCOUNT,IPRINT,UNIT,IOPT,IDATA
      LOGICAL IPRINT,IDATA
      INTEGER UNIT
      DIMENSION PHO(1), PHI(1), DELP(1), PHIA(15), PHIB(15), PHIU(15), P
     1SI(1)
      TAU=0.5*(1.0+(5.0)**0.5)
      TAUSQ=TAU*TAU
      ETA=ETA*(TAU+1.)
      GAMAL=0.
      GAMAU=1.0
      GAMAA=0.0
      UPHIA=UPHO
1     DO 2 I=1,K
      PHIU(I)=PHO(I)+GAMAU*DELP(I)
2     CONTINUE
      CALL LOCATE (PHIU,PSI,K,N,UPHIU)
      NCOUNT=NCOUNT+1
      IF (UPHIU.LE.UPHIA) GO TO 4
```

```
      GAMAB=GAMAL+(GAMAU-GAMAL)/TAU
      DO 3 I=1,K
      PHIA(I)=PHO(I)+DELP(I)*GAMAA
    3 CONTINUE
      GO TO 7
    4 GAMAL=GAMAA
      UPHIA=UPHIU
      GAMAA=GAMAU
      GAMAU=1.+GAMAU*TAU
      GO TO 1
    5 DO 6 I=1,K
      PHIA(I)=PHO(I)+DELP(I)*GAMAA
    6 CONTINUE
      CALL LOCATE (PHIA,PSI,K,N,UPHIA)
      NCOUNT=NCOUNT+1
      GO TO 9
    7 DO 8 I=1,K
      PHIB(I)=PHO(I)+DELP(I)*GAMAB
    8 CONTINUE
      CALL LOCATE (PHIB,PSI,K,N,UPHIB)
      NCOUNT=NCOUNT+1
    9 IF ((GAMAB-GAMAA).LT.ETA) GO TO 11
      IF (UPHIA.GE.UPHIB) GO TO 10
      GAMAU=GAMAB
      GAMAB=GAMAA
      UPHIB=UPHIA
      GAMAA=GAMAL+(GAMAU-GAMAL)/TAUSQ
      GO TO 5
   10 GAMAL=GAMAA
      GAMAA=GAMAB
      UPHIA=UPHIB
      GAMAB=GAMAL+(GAMAU-GAMAL)/TAU
      GO TO 7
   11 IF (UPHIA.LT.UPHIB) GO TO 12
      GAMA=GAMAB
      UPHI=UPHIB
      GO TO 13
   12 GAMA=GAMAA
      UPHI=UPHIA
   13 DO 14 I=1,K
      PHI(I)=PHO(I)+GAMA*DELP(I)
   14 CONTINUE
C
C     THIS VALUE OF GAMA IS THE FACTOR OF THE STEP DELP WHICH GIVES THE
C     BEST NEW POINT,WHEN STARTING FROM PHO
C
      RETURN
      END
C
C
C
C
C     ***********************************************************************
      SUBROUTINE LOCATE (PHI,PSI,K,N,UPHI)
C
C
C     LOCATE CALCULATES THE MINIMAX OBJECTIVE FUNCTION OF THE Y AT A
```

```
C     POINT PHI OVER A GIVEN SET OF SAMPLE POINTS PSI
C
C
      DIMENSION PHI(1), PSI(1)
      DO 1 I=1,N
      YT=Y(PHI,PSI(I),K)
      IF (I.EQ.1) UPHI=YT
      IF (YT.GT.UPHI) UPHI=YT
1     CONTINUE
      RETURN
      END
C
C
C
C
C     ..................................................................
      FUNCTION Y (PHI,F,K)
C
C     HERE THE FUNCTION VALUE Y AT A POINT PHI CORRESPONDING TO A SAMPLE
C     POINT F IS CALCULATED
C     DUMMY HAS BEEN DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE OF
C     K=15
C
      DIMENSION PHI(1), DUMMY(15)
      CALL ANAL (PHI,F,.FALSE.,K,Y1,DUMMY)
      Y=Y1
      RETURN
      END
C
C
C
C
C     ..................................................................
      SUBROUTINE NORM (K,W,WN)
      DIMENSION W(1), WN(1)
      SUM=0.
      DO 1 I=1,K
      SUM=SUM+W(I)*W(I)
1     CONTINUE
      SUMRT=SQRT(SUM)
      DO 2 I=1,K
      WN(I)=W(I)/SUMRT
2     CONTINUE
      RETURN
      END
C
C
C
C
C     ..................................................................
      SUBROUTINE TGSORT (A,I,N,M)
C
C
C     SUBROUTINE TGSORT (MAX,ITAG,NR,NM) FORMS A VECTOR OF TAGS ITAG SO
C     THAT ITAG(1),ITAG(2),......,ITAG(NR) ARE ORDERED SUBSCRIPTS OF
C     VECTOR MAX SUCH THAT MAX(ITAG(1)),MAX(ITAG(2)),.....,MAX(ITAG(NR))
C     ARE IN ALGEBRAIC ORDER
```

```
C       MM IS POSITIVE FOR A HIGH TO LOW ORDERING AND NEGATIVE FOR LOW TO
C       HIGH ORDERING
C       THIS SUBROUTINE LISTING WAS OBTAINED FROM THE DATA PROCESSING AND
C       COMPUTING CENTRE,LIBRARY INFORMATION SHEET MILIS 5.3.34,MCMASTER
C       UNIVERSITY
C
C
        DIMENSION A(1), I(1)
        LOGICAL HILO,TIME1
        HILO=M.LT.0
        N1=N+1
        N2=N1/2
        DO 1 J=1,N
        I(J)=-1
1       CONTINUE
        DO 6 K=1,N2
        TIME1=.TRUE.
        DO 4 J=1,N
        IF (I(J).GT.0) GO TO 4
        IF (.NOT.TIME1) GO TO 2
        TIME1=.FALSE.
        SMALL=BIG=A(J)
        JS=JB=J
        GO TO 4
2       IF (A(J).GT.SMALL) GO TO 3
        SMALL=A(J)
        JS=J
3       IF (A(J).LT.BIG) GO TO 4
        BIG=A(J)
        JB=J
4       CONTINUE
        L=N1-K
        I(JB)=IABS(I(JB))
        I(JS)=IABS(I(JS))
        IF (HILO) GO TO 5
        I(L)=ISIGN(JS,I(L))
        I(K)=ISIGN(JB,I(K))
        GO TO 6
5       I(L)=ISIGN(JB,I(L))
        I(K)=ISIGN(JS,I(K))
6       CONTINUE
        RETURN
        END
C
C
C
C
C       ***************************************************************
        SUBROUTINE SIMPLE (INFLAG,MX,NN,A,B,C,KO,KB,P,JH,X,Y,PE,E,NA)
C             CDC 6400 1172 OCTAL WORDS ARE REQUIRED
$       IBFTC SIMPLE  REF
C       AUTOMATIC SIMPLEX        REDUNDANT EQUATIONS CAUSE INFEASIBILITY
        REAL A(NA,136)
        REAL B(1),C(1),P(1),X(1),Y(1),PE(1),E(1)
        INTEGER INFLAG,MX,NN,KO(6),KB(1),JH(1)
        EQUIVALENCE (XX,LL)
C       THE FOLLOWING DIMENSION SHOULD BE THE SAME HERE AS IT IS IN
```

```
C       CALLER.
        REAL AA,AIJT,BB,COST,DT,NCOST,TEXP,TPIV,TY,XOLD,XX,XY,YI,YMAX
        INTEGER I,IA,INVC,IR,ITER,J,JT,K,KBJ,L,LL,M,M2,MM,N
        INTEGER NCUT,NPIV,NUMVR,NVER
        LOGICAL FEAS,VER,NEG,TRIG,KQ,ABSC
C
C            SET INITIAL VALUES, SET CONSTANT VALUES
        ITER=0
        NUMVR=0
        NUMPV=0
        M=MX
        N=NN
        TEXP=.5**16
        NCUT=4*M+10
        NVER=M/2+5
        M2=M**2
        FEAS=.FALSE.
        IF (INFLAG.NE.0) GO TO 3
C*      'NEW'     START  PHASE ONE WITH SINGLETON BASIS
        DO 2 J=1,N
        KB(J)=0
        KQ=.FALSE.
        DO 1 I=1,M
        IF (A(I,J).EQ.0.0) GO TO 1
        IF (KQ.OR.A(I,J).LT.0.0) GO TO 2
        KQ=.TRUE.
1       CONTINUE
        KB(J)=1
2       CONTINUE
3       DO 4 I=1,M
        JH(I)=-1
4       CONTINUE
C*      'VER'     CREATE INVERSE FROM 'KB' AND 'JH'     (STEP 7)
5       VER=.TRUE.
        INVC=0
        NUMVR=NUMVR+1
        TRIG=.FALSE.
        DO 6 I=1,M2
        E(I)=0.0
6       CONTINUE
        MM=1
        DO 7 I=1,M
        E(MM)=1.0
        PE(I)=0.0
        X(I)=B(I)
        IF (JH(I).NE.0) JH(I)=-1
        MM=MM+M+1
7       CONTINUE
C                     FORM INVERSE
        DO 14 JT=1,N
        IF (KB(JT).EQ.0) GO TO 14
        GO TO 30
C       30    CALL JMY
C                           CHOOSE PIVOT
8       TY=0.0
        KQ=.FALSE.
        DO 13 I=1,M
```

```
      IF (JH(I).NE.-1.OR.ABS(Y(I)).LE.TPIV) GO TO 13
      IF (KO) GO TO 10
      IF (X(I).EQ.0.) GO TO 9
      IF (ABS(Y(I)/X(I)).LE.TY) GO TO 13
      TY=ABS(Y(I)/X(I))
      GO TO 12
9     KO=.TRUE.
      GO TO 11
10    IF (X(I).NE.0..OR.ABS(Y(I)).LE.TY) GO TO 13
11    TY=ABS(Y(I))
12    IR=I
13    CONTINUE
      KB(JT)=0
C                         TEST PIVOT
      IF (TY.LE.0.) GO TO 14
C                         PIVOT
      GO TO 43
C     43    CALL PIV
14    CONTINUE
C                   RESET ARTIFICIALS
      DO 15 I=1,M
      IF (JH(I).EQ.-1) JH(I)=0
      IF (JH(I).EQ.0) FEAS=.FALSE.
15    CONTINUE
16    VER=.FALSE.
C                ***         PERFORM ONE ITERATION          ***
C*    'XCK'    DETERMINE FEASIBILITY                    (STEP 1)
      NEG=.FALSF.
      IF (FEAS) GO TO 18
      FEAS=.TRUE.
      DO 17 I=1,M
      IF (X(I).LT.0.0) GO TO 20
      IF (JH(I).EQ.0) FEAS=.FALSE.
17    CONTINUE
C*    'GET'    GET APPLICABLE PRICES                    (STEP 2)
      IF (.NOT.FEAS) GO TO 21
18    DO 19 I=1,M
      P(I)=PE(I)
      IF (X(I).LT.0.) X(I)=0.
19    CONTINUE
      ABSC=.FALSE.
      GO TO 27
20    FEAS=.FALSE.
      NEG=.TRUE.
21    DO 22 J=1,M
      P(J)=0.
22    CONTINUE
      ABSC=.TRUE.
      DO 26 I=1,M
      MM=I
      IF (X(I).GE.0.0) GO TO 24
      ABSC=.FALSE.
      DO 23 J=1,M
      P(J)=P(J)+E(MM)
      MM=MM+M
23    CONTINUE
      GO TO 26
```

```
24    IF (JH(I).NE.0) GO TO 26
      IF (X(I).NE.0.) ABSC=.FALSE.
      DO 25 J=1,M
      P(J)=P(J)-E(MM)
      MM=MM+M
25    CONTINUE
26    CONTINUE
C*    'MIN'     FIND MINIMUM REDUCED COST                 (STEP 3)
27    JT=0
      BB=0.0
      DO 29 J=1,N
      IF (KB(J).NE.0) GO TO 29
      DT=0.0
      DO 28 I=1,M
      DT=DT+P(I)*A(I,J)
28    CONTINUE
      IF (FEAS) DT=DT+C(J)
      IF (ABSC) DT=-ABS(DT)
      IF (DT.GE.BB) GO TO 29
      BB=DT
      JT=J
29    CONTINUE
C     TEST FOR NO PIVOT COLUMN
      IF (JT.LE.0) GO TO 50
C     TEST FOR ITERATION LIMIT EXCEEDED
      IF (ITER.GE.NCUT) GO TO 49
      ITER=ITER+1
C*    'JMY'     MULTIPLY INVERSE TIMES A(.,JT)            (STEP 4)
30    DO 31 I=1,M
      Y(I)=0.0
31    CONTINUE
      LL=0
      COST=C(JT)
      DO 34 I=1,M
      AIJT=A(I,JT)
      IF (AIJT.EQ.0.) GO TO 33
      COST=COST+AIJT*PE(I)
      DO 32 J=1,M
      LL=LL+1
      Y(J)=Y(J)+AIJT*E(LL)
32    CONTINUE
      GO TO 34
33    LL=LL+M
34    CONTINUE
C        COMPUTE PIVOT TOLERANCE
      YMAX=0.0
      DO 35 I=1,M
      YMAX=AMAX1(ABS(Y(I)),YMAX)
35    CONTINUE
      TPIV=YMAX*TEXP
C             RETURN TO INVERSION ROUTINE, IF INVERTING
      IF (VER) GO TO 8
C       COST TOLERANCE CONTROL
      RCOST=YMAX/BB
      IF (TRIG.AND.BB.GE.-TPIV) GO TO 50
      TRIG=.FALSE.
      IF (BB.GE.-TPIV) TRIG=.TRUE.
```

```
C*    'ROW'     SELECT PIVOT ROW                              (STEP 5)
C     AMONG EQS. WITH X=0, FIND MAXIMUM Y  AMONG ARTIFICIALS, OR, IF
C     NONE,
C     GET MAX POSITIVE Y(I) AMONG REALS.
      IR=0
      AA=0.0
      KQ=.FALSE.
      DO 39 I=1,M
      IF (X(I).NE.0.0.OR.Y(I).LE.TPIV) GO TO 39
      IF (JH(I).EQ.0) GO TO 37
      IF (KQ) GO TO 39
36    IF (Y(I).LE.AA) GO TO 39
      GO TO 38
37    IF (KQ) GO TO 36
      KQ=.TRUE.
38    AA=Y(I)
      IR=I
39    CONTINUE
      IF (IR.NE.0) GO TO 42
      AA=1.0E+20
C               FIND MIN. PIVOT AMONG POSITIVE EQUATIONS
      DO 40 I=1,M
      IF (Y(I).LE.TPIV.OR.X(I).LE.0.0.OR.Y(I)*AA.LE.X(I)) GO TO 40
      AA=X(I)/Y(I)
      IR=I
40    CONTINUE
      IF (.NOT.NEG) GO TO 42
C     FIND PIVOT AMONG NEGATIVE EQUATIONS, IN WHICH X/Y IS LESS THAN THE
C     MINIMUM X/Y IN THE POSITIVE EQUATIONS, THAT HAS THE LARGEST
C     ABSF(Y)
      BB=-TPIV
      DO 41 I=1,M
      IF (X(I).GE.0..OR.Y(I).GE.BB.OR.Y(I)*AA.GT.X(I)) GO TO 41
      BB=Y(I)
      IR=I
41    CONTINUE
C     TEST FOR NO PIVOT ROW
42    IF (IR.LE.0) GO TO 48
C*    'PIV'     PIVOT ON (IR,JT)                              (STEP 6)
      IA=JH(IR)
      IF (IA.GT.0) KB(IA)=0
43    NUMPV=NUMPV+1
      JH(IR)=JT
      KB(JT)=IR
      YI=-Y(IR)
      Y(IR)=-1.0
      LL=0
C                                 TRANSFORM INVERSE
      DO 46 J=1,M
      L=LL+IR
      IF (E(L).NE.0.0) GO TO 44
      LL=LL+M
      GO TO 46
44    XY=E(L)/YI
      PE(J)=PE(J)+COST*XY
      E(L)=0.0
      DO 45 I=1,M
```

```
      LL=LL+1
      E(LL)=E(LL)+XY*Y(I)
   45 CONTINUE
   46 CONTINUE
C                              TRANSFORM X
      XY=X(IR)/YI
      DO 47 I=1,M
      XOLD=X(I)
      X(I)=XOLD+XY*Y(I)
      IF (.NOT.VER.AND.X(I).LT.0..AND.XOLD.GE.0.) X(I)=0.
   47 CONTINUE
      Y(IR)=-YI
      X(IR)=-XY
      IF (VER) GO TO 14
      IF (NUMPV.LE.M) GO TO 16
C     TEST FOR INVERSION ON THIS ITERATION
      INVC=INVC+1
      IF (INVC.EQ.NVER) GO TO 5
      GO TO 16
C*    END OF ALGORITHM, SET EXIT VALUES                   ***
   48 IF (.NOT.FEAS.OR.RCOST.LE.-1000.) GO TO 50
C             INFINITE SOLUTION
      K=2
      GO TO 51
C         PROBLEM IS CYCLING
   49 K=4
      GO TO 51
C         FEASIBLE OR INFEASIBLE SOLUTION
   50 K=0
   51 IF (.NOT.FEAS) K=K+1
      DO 52 J=1,N
      XX=0.0
      KBJ=KB(J)
      IF (KBJ.NE.0) XX=X(KBJ)
      KB(J)=LL
   52 CONTINUE
      KO(1)=K
      KO(2)=ITER
      KO(3)=INVC
      KO(4)=NUMVR
      KO(5)=NUMPV
      KO(6)=JT
      RETURN
      END
```

CD TOT 0901

APPENDIX B

PROGRAM FOR INVESTIGATING MINIMAX OPTIMALITY CONDITIONS

B.1 Introduction

This program is a package of subprograms which investigates the optimality of a design or a proposed solution to an approximation problem in the minimax sense. The program is designed to test a solution for the necessary conditions for a minimax optimum by two different formulations. As indicated in Section 3.4, one uses linear programming, and the other the solution of a set of linear independent equations. A computer program written in Fortran (Version 2.3 and Scope Version 3.4 for the CDC 6400 computer) is listed at the end of the Appendix.

B.2 Program Description

The user may call the package from his main program as follows: CALL MINIMAX (K, KR, NR, YMAX, GRAD, NRMAX, DELTA, EPS, ICRIT, IDATA, IPRINT, MET, NORM, RELTOL, UNIT, K1, K3, MR3, MR1, MR2, X1, X2, X1SUM, X2SUM, R1, R2, R1NORM, R2NORM, OPTIM1, OPTIM2, A, B, C, X, PS, JH, XX, YY, PE, E, D, H, Q, IROW, ICOL, LL, MM).

The variables in the argument list of the above subroutine are ordered as input, output and storage variables respectively, and are listed below in that order.

The input variables are k, k_r, n_r, $\hat{y}([\hat{y}_1 \ldots \hat{y}_{n_r}]^T)$, $(\nabla \hat{y}^T)^T([\nabla\hat{y}_1 \ldots \nabla\hat{y}_{n_r}]^T)$, followed by

$\hat{n}_r$ maximum possible number of the $\hat{y}_\ell$.

δ numerical approximation to zero.

ε a user-specified factor; if $||r_1||$ or $||r_2|| < \varepsilon$ and the multiplier vector u_1 or $u_2 \geq 0$ the conditions are satisfied for Method 1 or 2; otherwise not.

ICRIT for ICRIT = 1, the user specifies the value of RELTOL and considers $\hat{y}_\ell$ for which $(1-\hat{y}_\ell/\hat{y}_1) \leq$ RELTOL for $\ell=2,\ldots,n_r$, to be active while when ICRIT = 2, the user specifies the value of $k_r(\leq n_r)$.

IDATA logical variable which, if .TRUE., enables the input data to be printed out; otherwise not.

IPRINT logical variable which if .TRUE., enables all intermediate and final results to be printed out, and no print-outs otherwise.

MET when MET=1,2, or 3, the package uses Method 1, Method 2 or both the methods, respectively.

NORM NORM=1 corresponds to the Euclidean vector norm and NORM=2 corresponds to the maximum absolute value of the elements of the vector.

RELTOL tolerance relative to $\hat{y}_1$ within which some of the $\hat{y}_2,\ldots,\hat{y}_{n_r}$ lie.

UNIT integer variable specifying the data set reference number of the output unit.

This is followed by $k_1(=k+1)$, $k_3(=2k+1)$ and $m_{r3}(=2k+1+n_r)$

For the output variables that follow, subscripts 1 and 2 correspond to methods 1 and 2, respectively, as shown below.

m_{r1}, m_{r2} number of $\hat{y}_\ell$ (for $\ell=1,\ldots,n_r$) considered when optimal conditions are reached.

$\underset{\sim}{u}_1, \underset{\sim}{u}_2$ vector of multipliers $[u_{11} \ldots u_{1m_{r1}}]^T$, $[u_{21} \ldots u_{2m_{r2}}]^T$

$\underset{\sim}{r}_1, \underset{\sim}{r}_2$ residual vectors $\sum_{\ell=1}^{m_{r1}} u_{1\ell} \nabla \hat{\underset{\sim}{y}}_\ell$, $\sum_{\ell=1}^{m_{r2}} u_{2\ell} \nabla \hat{\underset{\sim}{y}}_\ell$

$||\underset{\sim}{r}_1||, ||\underset{\sim}{r}_2||$ norm of vectors $\underset{\sim}{r}_1$, $\underset{\sim}{r}_2$

OPTIM1,OPTIM2 logical variables; indicate that the necessary conditions for minimax optimum are satisfied if .TRUE., and not satisfied otherwise.

The above output variable list is followed by storage variables, which form the rest of the argument list. The size of the storage arrays and vectors is determined by $\hat{n}_r$, k_1, k_3 and m_{r3}. The values of ε and δ as specified by the user are crucial for the verification of the optimality conditions, and should be carefully chosen. For further details, see Sections 5.4.4-5.4.6.

B.3 Required Subprograms

The user has to have a subprogram by which the discrete values of the n_r functions $\hat{y}_\ell$ (arranged in descending magnitude) and their derivatives $(\nabla \hat{y}^T)^T$ with respect to the parameters $\phi_1, \phi_2, \ldots \phi_k$ are explicitly available. The package uses the following subroutines, the listings of which are available as indicated in the References (see Subroutine ARRAY, Subroutine MINV, Subroutine MFGR, Subroutine SIMPLE, Subroutine SOLVE).

ARRAY converts data arrays from single to double dimension or vice versa while MINV inverts a matrix and calculates its determinant. MFGR determines the rank and linearly independent rows and columns of a given matrix. SIMPLE is a linear-program solving subroutine (listing available in Section A.7) and SOLVE solves a set of linear simultaneous equations.

B.4 Comments

The program was used to test a solution on the problem of lower-order modelling of a ninth-order nuclear reactor system as treated in Section 3.4.7. Fig. B.1 shows a typical printout of the package for this problem.

This program was run and tested on a CDC 6400 computer. The package requires roughly 40,000 octal units of memory for k=15 and n_r=15. A Fortran listing consisting of 721 cards (including comments) is included in Section B.5.

METHOD 1

NUMBER OF HIGHEST MAXIMA CONSIDERED	VECTOR OF MULTIPLIERS	SUM OF MULTIPLIERS	VECTOR OF RESIDUALS	NORM OF RESIDUAL VECTOR	ARE NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM SATISFIED FOR A USER-SPECIFIED VALUE OF KR OR RELTOL
(MR)	(X1)	(X1SUM)	(R1)	(R1NORM)	(YES/NO)
1	.10000000E+01	.10000000E+01	.38711013E-03 -.14208087E-03	.38711013E-03	NO
2	.98710491E+00 .12895086E-01	.10000000E+01	-.25789922E-09 .25789922E-09	.25789922E-09	YES

METHOD 2

NUMBER OF HIGHEST MAXIMA CONSIDERED	VECTOR OF MULTIPLIERS	SUM OF MULTIPLIERS	VECTOR OF RESIDUALS	NORM OF RESIDUAL VECTOR	ARE NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM SATISFIED FOR A USER-SPECIFIED VALUE OF KR OR RELTOL
(MR)	(X2)	(X2SUM)	(R2)	(R2NORM)	(YES/NO)
1	.10000000E+01	.10000000E+01	.38711013E-03 -.14208087E-03	.38711013E-03	NO
2	.98710492E+00 .12895077E-01	.10000000E+01	0. -.35255563E-09	.35255563E-09	YES

Fig. B.1 Typical printout of results for the problem given in the text.

B.5 Fortran Listing for MINIMAX Program

```
C           PROGRAM FOR INVESTIGATING MINIMAX OPTIMALITY CONDITIONS
C           ---------------------------------------------------------
C
C     AUTHORS    J.W.BANDLER AND T.V.SRINIVASAN,DEPARTMENT OF ELECTRICAL
C                ENGINEERING,MCMASTER UNIVERSITY,HAMILTON,ONTARIO,CANADA
C
C     THIS PROGRAM IS A PACKAGE OF SUBPROGRAMS WHICH INVESTIGATES THE
C     OPTIMALITY OF A DESIGN OR A PROPOSED SOLUTION TO AN APPROXIMATION
C     PROBLEM IN THE MINIMAX SENSE
C     ................................................................
C     A TYPICAL MAIN PROGRAM FOR MINIMAX SOLUTION CHECK FOLLOWS---------
C     DIMENSION YMAX(15),GRAD(15,10),X1(15),X2(15),R1(10),R2(10)
C     DIMENSION A(21,36),B(21),C(36),X(36),PS(21),XX(21),YY(21),PE(21),
C    1 E(21,21)
C     DIMENSION D(15,10),H(11,11),Q(11),IROW(15),ICOL(10),LL(15),MM(15)
C     LOGICAL IDATA,IPRINT,OPTIM1,OPTIM2
C     INTEGER UNIT
C     K=2
C     NRMAX=15
C     NR=4
C     DELTA=.01
C     EPS=1.0E-04
C     ICRIT=2
C     KR=NR
C     IDATA=.T.
C     IPRINT=.T.
C     MET=3
C     NORM=1
C     UNIT=6
C     K1=K+1
C     K3=2*K+1
C     MR3=2*K+1+NRMAX
C     READ(5,1) (YMAX(I),I=1,NR)
C     READ(5,2) ((GRAD(I,J),J=1,K),I=1,NR)
C    1 FORMAT(5E16.8)
C    2 FORMAT(2E16.8)
C     CALL MINIMAX(K,KR,NR,YMAX,GRAD,NRMAX,DELTA,EPS,ICRIT,IDATA,IPRINT,
C    1MET,NORM,RELTOL,UNIT,K1,K3,MR3,MR1,MR2,X1,X2,X1SUM,X2SUM,R1,R2,R1N
C    2NORM,R2NORM,OPTIM1,OPTIM2,A,B,C,X,PS,JH,XX,YY,PE,E,D,H,Q,IROW,ICOL
C    3,LL,MM)
C     STOP
C     END
C
C
C     ................................................................
      SUBROUTINE MINIMAX (K,KR,NR,YMAX,GRAD,NRMAX,DELTA,EPS,ICRIT,IDATA,
     1IPRINT,MET,NORM,RELTOL,UNIT,K1,K3,MR3,MR1,MR2,X1,X2,X1SUM,X2SUM,R1
     2,R2,R1NORM,R2NORM,OPTIM1,OPTIM2,A,B,C,X,PS,JH,XX,YY,PE,E,D,H,Q,IRO
     3W,ICOL,LL,MM)
C
C
C     THE MINIMAX SOLUTION TESTING IS DONE BY TWO METHODS-MET1 AND MET2
C     MET1 CONSIDERS A LINEAR PROGRAMMING FORMULATION
C     MET2 CONSIDERS A FORMULATION CONSISTING OF A SET OF LINEAR
C     EQUATIONS
C
C
C     INPUT-OUTPUT INFORMATION-------
C     THE USER HAS TO SPECIFY VALUES FOR K,KR (OR RELTOL) ,NRMAX,NR,
```

```
C       YMAX.GRAD.DELTA.EPS.ICRIT.IDATA.IPRINT.MET.NORM.UNIT.K1.K3 AND MR3
C       THE OUTPUT VARIABLES ARE MR1.R1.R1NORM.X1.X1SUM.OPTIM1.MR2.R2.
C       R2NORM.X2.X2SUM AND OPTIM2
C       THE VARIABLES A.B.C.X.PS.JH.XX.YY.PE.L.D.H.G.IROW.ICOL.LL.MM ARE
C       TEMPORARY STORAGE SPACES CORRESPONDING TO K AND NRMAX (TO BE
C       DIMENSIONED BY THE USER)
C
C
C       APPENDIX OF VARIABLES-------
C       K      =NUMBER OF VARIABLE PARAMETERS
C       NRMAX  =MAXIMUM NUMBER OF FUNCTIONS YMAX THAT MAY BE ENCOUNTERED BY
C               THE USER.FOR THE SAKE OF SAVING MEMORY SPACE.NRMAX CAN BE
C               PUT EQUAL TO NR IF NR IS KNOWN BEFOREHAND
C       NR     =NUMBER OF HIGHEST FUNCTIONS YMAX(1).YMAX(2)..... WHICH ARE
C               AVAILABLE FOR CHECKING A SOLUTION FOR THE NECESSARY
C               CONDITIONS FOR A MINIMAX OPTIMUM. NR SHOULD NEVER BE
C               GREATER THAN NRMAX
C       KR     =NUMBER OF HIGHEST YMAX(1).....YMAX(NR) THAT MAY BE
C               CONSIDERED ACTIVE BY THE USER FOR CHECKING OPTIMALITY
C               CONDITIONS. KR IS LESS THAN OR EQUAL TO NR. THE VALUE OF KR
C               HAS TO BE SUPPLIED BY THE USER IF ICRIT=2
C       YMAX   =VECTOR OF FUNCTIONS YMAX(1).....YMAX(NR) ARRANGED IN
C               DECREASING MAGNITUDE. THESE FUNCTIONS ARE TO BE TESTED FOR
C               THE NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM. YMAX(I) IS
C               GREATER THAN OR EQUAL TO YMAX(I+1) FOR I=1.....NR-1
C       GRAD   =MATRIX OF FIRST DERIVATIVES OF VECTOR YMAX WITH RESPECT TO
C               THE K PARAMETERS. THE ROWS OF GRAD CORRESPOND TO THE
C               GRADIENTS OF YMAX(1).YMAX(2).....YMAX(NR) RESPECTIVELY.
C               GRAD IS OF SIZE (NRMAX.K)
C       DELTA  =TEST FACTOR FOR ZERO.AFFECTED BY ROUNDOFF NOISE. THE VALUE
C               OF DELTA DEPENDS UPON THE MAGNITUDE OF ELEMENTS OF GRAD
C       EPS    =SCALE FACTOR. WHEN CONSIDERING METHOD 1 IF THE MULTIPLIERS
C               X1(1).....X1(MR1) ARE NON-NEGATIVE AFTER CONSIDERING MR1
C               HIGHEST FUNCTIONS YMAX(1).....YMAX(MR1).THE NORM OF THE
C               RESIDUAL VECTOR R1 IS COMPARED WITH EPS. IF R1NORM IS LESS
C               THAN OR EQUAL TO EPS THE NECESSARY CONDITIONS FOR A MINIMAX
C               OPTIMUM ARE SATISFIED BY YMAX(1).....YMAX(MR1). A SIMILAR
C               SITUATION HOLDS FOR METHOD 2 WHEN MR2 HIGHEST FUNCTIONS ARE
C               CONSIDERED.
C       ICRIT  =THERE ARE TWO CRITERIA AVAILABLE FOR CHECKING THE NECESSARY
C               CONDITIONS FOR A MINIMAX OPTIMUM. FOR ICRIT=1.THE USER HAS
C               TO SPECIFY THE VALUE OF RELTOL AND FOR ICRIT=2.THE USER HAS
C               TO SPECIFY THE VALUE OF KR. IF THE USER HAS NO IDEA OF HOW
C               MANY OF THE HIGHEST FUNCTIONS TO CHOOSE OUT OF YMAX(1).....
C               YMAX(NR).HE COULD SPECIFY A VALUE OF KR EQUAL TO NR. IF.ON
C               THE OTHER HAND.THE USER WISHES TO SPECIFY A TOLERANCE BAND
C               BELOW YMAX(1) WITHIN WHICH HE CONSIDERS THE FUNCTIONS TO BE
C               ACTIVE.HE COULD SPECIFY THE VALUE OF RELTOL
C       IDATA  =LOGICAL VARIABLE.WHICH IF .TRUE. ENABLES INPUT DATA TO BE
C               PRINTED OUT..OTHERWISE NOT.
C       IPRINT =LOGICAL VARIABLE.WHICH IF .TRUE. ENABLES ALL INTERMEDIATE
C               AND FINAL RESULTS TO BE PRINTED OUT.AND NO PRINTOUTS
C               OTHERWISE.
C       MET    =INTEGER VARIABLE WHICH ENABLES THE USER TO CHECK THE
C               NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM BY METHODS 1 OR
C               2 OR BOTH FOR MET=1 OR 2 OR 3
C       NORM   =VARIABLE WHICH ALLOWS TWO NORMS TO BE AVAILABLE FOR VECTORS
```

```
C           R1 AND R2. IF NORM=1,THE EUCLIDEAN NORM OF A VECTOR IS
C           CALCULATED.IF NORM=2,THE VECTOR NORM IS EQUAL TO THE
C           MAXIMUM ABSOLUTE VALUE OF THE VECTOR ELEMENTS
C     RELTOL=TOLERANCE RELATIVE TO YMAX(1) WITHIN WHICH YMAX(2),....,
C           YMAX(NR) LIE.THIS FACTOR HAS TO BE SPECIFIED BY THE USER IF
C           ICRIT=2,WHEN THE USER CONSIDERS THOSE FUNCTIONS FOR WHICH
C           (1.-YMAX(L)/YMAX(1)),L=1,2,....,NR, IS LESS THAN RELTOL,TO
C           BE ACTIVE FOR OPTIMALITY CONDITIONS
C     UNIT  =INTEGER VARIABLE SPECIFYING THE DATA SET REFERENCE NUMBER
C           OF THE OUTPUT UNIT.
C
C     IN THE FOLLOWING SECTION SUBSCRIPTS 1 AND 2 DENOTE METHODS 1 AND 2
C     RESPECTIVELY.
C     MR1,MR2          =NUMBER OF HIGHEST FUNCTIONS YMAX(1),....,YMAX(NR)
C                       WHICH SATISFY THE THE NECESSARY CONDITIONS FOR A
C                       MINIMAX OPTIMUM AS VERIFIED BY METHODS 1 AND 2
C                       RESPECTIVELY
C     X1,X2            =VECTOR OF MULTIPLIERS OF LENGTH MR1 AND MR2
C                       RESPECTIVELY,WHEN THE NECESSARY CONDITIONS FOR A
C                       MINIMAX OPTIMUM ARE SATISFIED AS VERIFIED BY METHODS
C                       1 AND 2. AT AN OPTIMUM,THE ELEMENTS OF THE
C                       MULTIPLIERS ARE ALL NON-NEGATIVE
C     X1SUM,X2SUM      =SUM OF ELEMENTS OF VECTORS X1 AND X2 RESPECTIVELY.
C                       AT THE OPTIMUM,THE ELEMENTS OF THE VECTORS ARE ALL
C                       NON-NEGATIVE AND ADD UP TO UNITY
C     R1               =RESIDUAL VECTOR OF LENGTH K GENERATED BY LINEAR
C                       COMBINATION OF THE GRADIENTS OF YMAX(1),YMAX(2),....,
C                       YMAX(MR1) BY THE MULTIPLIERS X1(1),X1(2),.....,X1(MR1)
C                       GOT FROM METHOD 1. THUS R1 IS A PRODUCT OF THE ROW-
C                       VECTOR X1 POST-MULTIPLIED BY THE MR1 ROWS OF GRAD
C     R2               =RESIDUAL VECTOR OF LENGTH K GENERATED BY LINEAR
C                       COMBINATION OF THE GRADIENTS OF YMAX(1),YMAX(2),....,
C                       YMAX(MR2) BY THE MULTIPLIERS X2(1),X2(2),.....,X2(MR2)
C                       GOT FROM METHOD 2. THUS R2 IS A PRODUCT OF THE ROW-
C                       VECTOR X2 POST-MULTIPLIED BY THE MR2 ROWS OF GRAD
C     R1NORM,R2NORM=NORMS OF VECTORS R1 AND R2
C     OPTIM1,OPTIM2=LOGICAL VARIABLES.IF .TRUE. INDICATE THAT NECESSARY
C                       CONDITIONS ARE MET FOR A USER-SPECIFIED VALUE OF EPS
C                       AS VERIFIED BY METHODS 1 AND 2 RESPECTIVELY. IF THEY
C                       ARE .FALSE. THE NECESSARY CONDITIONS ARE NOT
C                       SATISFIED
C
C     K1               =K+1
C     K3               =2*K+1
C     MR3              =2*K+1+NRMAX
C     K1,K3,MR3 ARE INTEGERS WHICH ARE NECESSARY FOR EFFICIENT USE OF
C     COMPUTER CORE MEMORY FOR SOME TEMPORARY STORAGE VECTORS AND ARRAYS
C
C
C     DIMENSIONING INFORMATION -------
C     THE USER HAS TO DIMENSION IN HIS MAIN PROGRAM THE FOLLOWING
C     ARRAYS AND VECTORS.
C     YMAX =VECTOR OF DIMENSION NRMAX
C     GRAD =ARRAY OF DIMENSION (NRMAX,K)
C     X1,X2=VECTORS OF LENGTH NRMAX
C     R1,R2=VECTORS OF LENGTH K
C     A     =ARRAY OF SIZE (K3,MR3)
```

```
C       B,PS,JH,XX,YY,PE=VECTORS OF LENGTH K3
C       C,X   =VECTORS OF DIMENSION MR3
C       F     =ARRAY OF SIZE (K3,K3)
C       D     =ARRAY OF DIMENSION (NRMAX,K)
C       D     =ARRAY OF DIMENSION (NRMAX,K)
C       H     =SQUARE MATRIX OF SIZE (K1,K1)
C       Q     =VECTOR OF LENGTH K1
C       IROW  =VECTOR OF LENGTH NRMAX
C       ICOL  =VECTOR OF LENGTH K
C       LL,MM=VECTORS OF LENGTH NRMAX
C
C
C       TYPE DECLARATION -------
C       THE USER HAS TO DECLARE THE TYPE OF SOME OF THE VARIABLES AS
C       FOLLOWS.
C       INTEGER UNIT
C       LOGICAL IDATA,IPRINT,OPTIM1,OPTIM2
C
C
C       SUBROUTINE INFORMATION -------
C       THE USER HAS TO SUPPLY THE FOLLOWING SUBROUTINES WITH THIS PACKAGE
C       OR ENSURE THAT THESE SUBROUTINES ARE IN THE PERMANENT LIBRARY OF
C       THE COMPUTER HE IS USING
C       SUBROUTINE ARRAY  -REFERENCE  (2)
C       SUBROUTINE MINV   -REFERENCE  (3)
C       SUBROUTINE HFGR   -REFERENCE  (4)
C       SUBROUTINE SIMPLE -REFERENCES (5),(6)
C
C       IT IS IMPORTANT TO POINT OUT THAT THE VALUES OF EPS AND DELTA AS
C       SPECIFIED BY THE USER ARE CRITICAL FOR TESTING A SOLUTION FOR THE
C       NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM,AND A GREAT DEAL OF
C       CARE HAS TO BE EXCERCISED WHEN SPECIFYING VALUES FOR THEM.
C       IN ADDITION,IT HAS TO BE POINTED OUT THAT *...* IN A FORMAT
C       STATEMENT IS LIKE A HOLLERITH PARAMETER INCLUDING WHATEVER IS
C       WITHIN THE TWO * SYMBOLS IN THE HOLLERITH FIELD.
        LOGICAL IPRINT,IDATA,OPTIM1,OPTIM2,ISP
        INTEGER UNIT
        DIMENSION YMAX(1), X1(1), X2(1), R1(1), R2(1), GRAD(NRMAX,1)
        DIMENSION A(K3,1), B(1), C(1), X(1), PS(1), JH(1), XX(1), YY(1), P
       1E(1), E(K3,1), H(K1,1), Q(1), IROW(1), ICOL(1), LL(1), MM(1), D(NR
       2MAX,1)
        IF (NR.LE.NRMAX) GO TO 1
        WRITE (UNIT,20)
        RETURN
1       CONTINUE
        ISP=.F.
        OPTIM1=.F.
        OPTIM2=.F.
        GO TO (2,6), ICRIT
2       KR=1
        IF (NR.EQ.1) GO TO 4
        DO 3 I=2,NR
        IF ((YMAX(1)-YMAX(I)).LE.(RELTOL*YMAX(1))) KR=KR+1
3       CONTINUE
4       IF (.NOT.IDATA) GO TO 8
        WRITE (UNIT,21) K,KR,NRMAX,NR,DELTA,EPS,ICRIT,IDATA,IPRINT,MET,NOR
       1M,RELTOL,UNIT,K1,K3,MK3,(I,YMAX(I),I=1,NR)
```

```
      DO 5 I=1,NR
      WRITE (UNIT,22) ((I,J,GRAD(I,J)),J=1,K)
    5 CONTINUE
      GO TO 8
    6 CONTINUE
      RELTOL=1.-YMAX(KR)/YMAX(1)
      IF (.NOT.IDATA) GO TO 8
      WRITE (UNIT,23) K,KR,NRMAX,NR,DELTA,EPS,ICRIT,IDATA,IPRINT,MET,NOR
     1M,RELTOL,UNIT,K1,K3,MR3,(I,YMAX(I),I=1,NR)
      DO 7 I=1,NR
      WRITE (UNIT,22) ((I,J,GRAD(I,J)),J=1,K)
    7 CONTINUE
    8 GO TO (9,14,9), MET
    9 CONTINUE
      IF (.NOT.IPRINT) GO TO 10
      WRITE (UNIT,24)
      WRITE (UNIT,25)
      WRITE (UNIT,26)
   10 CONTINUE
      MR=1
      X1(1)=1.
      X1SUM=1.
      DO 11 J=1,K
      R1(J)=GRAD(1,J)
   11 CONTINUE
      CALL SOLCHK (K,MR,MR1,ICRIT,IPRINT,NORM,UNIT,X1,X1SUM,R1,R1NORM,EP
     1S,OPTIM1)
      IF (OPTIM1) GO TO 13
      IF (KR.EQ.1) GO TO 13
      DO 12 II=2,KR
      MR=II
      CALL MET1 (K,K3,MR,NR,MR3,YMAX,GRAD,X1,X1SUM,R1,A,B,C,KO,X,PS,JH,X
     1X,YY,PE,E,NRMAX)
      CALL SOLCHK (K,MR,MR1,ICRIT,IPRINT,NORM,UNIT,X1,X1SUM,R1,R1NORM,EP
     1S,OPTIM1)
      IF (OPTIM1) GO TO 13
   12 CONTINUE
   13 CONTINUE
      IF (MET.NE.3) RETURN
   14 CONTINUE
      IF (.NOT.IPRINT) GO TO 15
      WRITE (UNIT,27)
      WRITE (UNIT,25)
      WRITE (UNIT,28)
   15 CONTINUE
      MR=1
      X2(1)=1.
      X2SUM=1.
      DO 16 J=1,K
      R2(J)=GRAD(1,J)
   16 CONTINUE
      CALL SOLCHK (K,MR,MR2,ICRIT,IPRINT,NORM,UNIT,X2,X2SUM,R2,R2NORM,EP
     1S,OPTIM2)
      IF (OPTIM2) GO TO 19
      IF (KR.EQ.1) GO TO 19
      DO 18 II=2,KR
      MR=II
```

```
      CALL MET2 (K,K1,MR,NR,YMAX,GRAD,DELTA,IPRINT,ISP,UNIT,X2,X2SUM,R2,
     1D,H,Q,IROW,ICOL,LL,MM,NRMAX)
      IF (.NOT.ISP) GO TO 17
C
C     WHEN ISP IS .TRUE.,EITHER THE NUMBER OF UNKNOWN MULTIPLIERS IS
C     GREATER THAN THE NUMBER OF INDEPENDENT EQUATIONS OR THE VALUE OF
C     DELTA IS TOO SMALL,SO THAT WE SWITCH FROM METHOD 2 TO METHOD 1 FOR
C     THE CURRENT VALUE OF MR
C
      CALL MET1 (K,K3,MR,NR,MR3,YMAX,GRAD,X2,X2SUM,R2,A,B,C,KO,X,PS,JH,X
     1X,YY,PE,E,NRMAX)
      ISP=.F.
17    CONTINUE
      CALL SOLCHK (K,MR,MR2,ICRIT,IPRINT,NORM,UNIT,X2,X2SUM,R2,R2NORM,EP
     1S,OPTIM2)
      IF (OPTIM2) GO TO 19
18    CONTINUE
19    CONTINUE
      RETURN
C
C
C
C
C
20    FORMAT (1H1/* NR IS GREATER THAN NRMAX HERE, AND THIS CALLS FOR AN
     1 INCREASE IN NRMAX TO A VALUE GREATER THAN OR EQUAL TO NR*)
21    FORMAT (1H1/60X,*INPUT DATA LIST*/61X,*---------------*///66X,*K=*
     1,I5/65X,*KR=*,I5,*(CORRESPONDING TO RELTOL)*/62X,*NRMAX=*,I5/65X,*
     2NR=*,I5/62X,*DELTA=*,E16.8/64X,*EPS=*E16.8/62X,*ICRIT=*,I5/62X,*ID
     3ATA=*,L5/61X,*IPRINT=*,L5/64X,*MET=*,I5/63X,*NORM=*,I5/61X,*RELTOL
     4=*,E16.8/63X,*UNIT=*,I5/65X,*K1=*,I5/65X,*K3=*,I5/64X,*MR3=*,I5/(4
     5X,*YMAX(*,I2,*)=*,E16.8,9X,*YMAX(*,I2,*)=*,E16.8,9X,*YMAX(*,I2,*)=
     6*,E16.8,9X,*YMAX(*,I2,*)=*,E16.8))
22    FORMAT (* GRAD(*,I2,*,*,I2,*)=*,E16.8,6X,*GRAD(*,I2,*,*,I2,*)=*,E1
     16.8,6X,*GRAD(*,I2,*,*,I2,*)=*,E16.8,6X,*GRAD(*,I2,*,*,I2,*)=*,E16.
     28)
23    FORMAT (1H1/60X,*INPUT DATA LIST*/61X,*---------------*///66X,*K=*
     1,I5/65X,*KR=*,I5/62X,*NRMAX=*,I5/65X,*NR=*,I5/62X,*DELTA=*,E16.8/6
     24X,*EPS=*E16.8/62X,*ICRIT=*,I5/62X,*IDATA=*,L5/61X,*IPRINT=*,L5/64
     3X,*MET=*,I5/63X,*NORM=*,I5/61X,*RELTOL=*,E16.8,*(CORRESPONDING TO
     4KR)*/63X,*UNIT=*,I5/65X,*K1=*,I5/65X,*K3=*,I5/64X,*MR3=*,I5/(4X,*Y
     5MAX(*,I2,*)=*,E16.8,9X,*YMAX(*,I2,*)=*,E16.8,9X,*YMAX(*,I2,*)=*,E1
     66.8,9X,*YMAX(*,I2,*)=*,E16.8))
24    FORMAT (1H1/64X,*METHOD 1*/64X,*----------*///)
25    FORMAT (6X,*NUMBER OF*,14X,*VECTOR*,18X,*SUM*,18X,*VECTOR*,17X,*NO
     1RM*,9X,*ARE NECESSARY CONDITIONS*//7X,*HIGHEST*,17X,*OF*,20X,*OF*,
     221X,*OF*,20X,*OF*,10X,*FOR A MINIMAX OPTIMUM*//7X,*MAXIMA*,14X,*MU
     3LTIPLIERS*,11X,*MULTIPLIERS*,12X,*RESIDUALS*,14X,*RESIDUAL*,7X,*SA
     4TISFIED FOR A USER-*//6X,*CONSIDERED*,80X,*VECTOR*,8X,*SPECIFIED V
     5ALUE OF KR*//110X,*OR RELTOL*/)
26    FORMAT (9X,*(MR)*,17X,*(X1)*,17X,*(X1SUM)*,16X,*(R1)*,18X,*(R1NORM
     1)*,12X,*(YES/NO)*//)
27    FORMAT (1H1/64X,*METHOD 2*/64X,*----------*///)
28    FORMAT (9X,*(MR)*,17X,*(X2)*,17X,*(X2SUM)*,16X,*(R2)*,18X,*(R2NORM
     1)*,12X,*(YES/NO)*//)
      END
C
```

```
C
C
C     ...............................................................
      SUBROUTINE SOLCHK (K,MR,MMR,ICRIT,IPRINT,NORM,UNIT,X,XSUM,R,RNORM,
     1EPS,OPTIM)
C
C     THIS SUBROUTINE CHECKS THE SOLUTION FOR NECESSARY CONDITIONS FOR
C     A MINIMAX OPTIMUM BY FIRST TESTING WHETHER THE MULTIPLIERS X(1),
C     X(2),....,X(MR) ARE NON-NEGATIVE,AND THEN FINDING OUT IF THE NORM
C     RNORM OF THE RESIDUAL VECTOR R IS LESS THAN OR EQUAL TO EPS
C
      DIMENSION X(1), R(1)
      INTEGER UNIT
      LOGICAL IPRINT,OPTIM
      MR1=MR+1
      K1=K+1
      GO TO (1,2), NORM
1     RNORM=ANORM1(K,R)
      GO TO 3
2     RNORM=ANORM2(K,R)
3     CONTINUE
      IF (RNORM.GT.EPS) GO TO 4
      OPTIM=.T.
      MMR=MR
4     CONTINUE
      DO 5 J=1,MR
      IF (X(J).GE.0.) GO TO 5
      OPTIM=.F.
      GO TO 6
5     CONTINUE
6     CONTINUE
      IF (.NOT.IPRINT) RETURN
      IF (OPTIM) GO TO 7
      WRITE (UNIT,15) MR,X(1),XSUM,R(1),RNORM
      GO TO 8
7     WRITE (UNIT,16) MR,X(1),XSUM,R(1),RNORM
8     IF (MR.EQ.1) GO TO 13
      IF (K-MR) 9,9,11
9     DO 10 I=2,K
      WRITE (UNIT,17) X(I),R(I)
10    CONTINUE
      IF (K.EQ.MR) RETURN
      WRITE (UNIT,18) (X(I),I=K1,MR)
      GO TO 14
11    DO 12 I=2,MR
      WRITE (UNIT,17) X(I),R(I)
12    CONTINUE
13    IF (K.EQ.MR) RETURN
      WRITE (UNIT,19) (R(I),I=MR1,K)
14    RETURN
C
15    FORMAT (//10X,I2,11X,E16.8,6X,E16.8,6X,E16.8,6X,E16.8,11X,*NO*)
16    FORMAT (//10X,I2,11X,E16.8,6X,E16.8,6X,E16.8,6X,E16.8,11X,*YES*)
17    FORMAT (/29X,E16.8,28X,E16.8)
18    FORMAT (/23X,E16.8)
19    FORMAT (/67X,E16.8)
      END
```

```
C
C
C
C     ..................................................................
      SUBROUTINE MET1 (K,K3,MR,NR,KR3,YMAX,GRAD,X1,X1SUM,R1,A,B,C,KO,X,P
     1S,JH,XX,YY,PE,E,NRMAX)
      DIMENSION YMAX(1), GRAD(NRMAX,1), X1(1), R1(1)
      DIMENSION A(K3,1), B(1), C(1), X(1), PS(1), JH(1), XX(1), YY(1), P
     1E(1), E(K3,1)
      MR1=MR+1
      MR2=MR+2
      MR3=MR1+2*K
      K2=2*K
      K3=K2+1
      DO 2 I=1,MR
      DO 1 J=1,K
      A(J,I)=GRAD(I,J)
      A(J+K,I)=-A(J,I)
1     CONTINUE
2     CONTINUE
      DO 3 J=1,K2
      A(J,MR1)=-1.
3     CONTINUE
      DO 5 J=1,K2
      DO 4 I=MR2,MR3
      A(J,I)=0.
      IF (J.EQ.(I-MR1)) A(J,I)=1.
4     CONTINUE
5     CONTINUE
      DO 6 I=1,MR
      A(K3,I)=1.
6     CONTINUE
      DO 7 I=MR1,MR3
      A(K3,I)=0.
7     CONTINUE
      DO 8 I=1,K2
      B(I)=0.
8     CONTINUE
      B(K3)=1.
      DO 9 I=1,MR3
      C(I)=0.
      IF (I.EQ.MR1) C(I)=1.
9     CONTINUE
C
C
C     SUBROUTINE SIMPLE IS NOW GOING TO BE CALLED.ANY ALTERNATIVE
C     CHOICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT
C     PERFORMS THE FOLLOWING OPERATION.
C     SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF
C     MINIMIZING C*X SUBJECT TO A*X=B,WHERE X,C,B ARE VECTORS OF LENGTH
C     MR3,MR3,K3 RESPECTIVELY,AND A IS A MATRIX OF SIZE (K3,MR3)
C     SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE
C     WITH SHARE DISTRIBUTING AGENCY,REFERENCE NUMBER SDA 3384 AND
C     WRITTEN BY R.J.CLASEN (REFERENCE NUMBER (5))
C     A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE
C     NUMBER (6).
C
```

```
C     NA AND IFLAG ARE TO BE SPECIFIED BEFORE CALLING SIMPLE
C     IFLAG IS SET EQUAL TO ZERO
C     NA IS THE FIRST DIMENSION OF ARRAY A AND IS SET EQUAL TO K3
C     X IS A VECTOR OF DIMENSION MR3
C     THE FOLLOWING SUBSCRIPTED VARIABLES ARE PART OF THE ARGUMENT LIST
C     OF SIMPLE AND ARE TEMPORARY STORAGE SPACES TO BE DIMENSIONED IN
C     THE CALLING PROGRAM (MINIMAX)
C     PS,JH,XX,YY AND PE ARE TEMPORARY STORAGE VECTORS OF DIMENSION NA
C     E IS A TEMPORARY STORAGE MATRIX OF DIMENSION (NA,NA)
C     KO IS A VECTOR OF LENGTH 6.UPON COMPLETION OF THE EXECUTION OF
C     SIMPLE, KO(1)=0 IF THE LINEAR PROGRAMMING PROBLEM WAS FEASIBLE.
C     THE SOLUTION LIES IN X(J),J=1,MR3
C
C
      IFLAG=0
      NA=K3
      CALL SIMPLE (IFLAG,K3,MR3,A,B,C,KO,X,PS,JH,XX,YY,PE,E,NA)
      DO 10 J=1,MR
      X1(J)=X(J)
 10   CONTINUE
      DO 12 I=1,K
      R1(I)=0.
      DO 11 J=1,MR
      R1(I)=R1(I)+A(I,J)*X1(J)
 11   CONTINUE
 12   CONTINUE
      X1SUM=0.
      DO 13 J=1,MR
      X1SUM=X1SUM+X1(J)
 13   CONTINUE
      RETURN
      END
C
C
C
C
      ...............................................................
      SUBROUTINE MET2 (K,K1,MR,NR,YMAX,GRAD,DELTA,IPRINT,ISP,UNIT,X2,X2S
     1UM,R2,D,H,Q,IROW,ICOL,LL,MM,NRMAX)
      DIMENSION YMAX(1), GRAD(NRMAX,1), X2(1), R2(1)
      LOGICAL IPRINT,ISP
      INTEGER UNIT
      DIMENSION D(NRMAX,1), H(K1,1), Q(1), IROW(1), ICOL(1), LL(1), MM(1
     1)
      MR1=MR+1
      DO 2 I=1,MR
      DO 1 J=1,K
      D(I,J)=GRAD(I,J)
 1    CONTINUE
 2    CONTINUE
C
C
C     SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR
C     VICE VERSA.IT ENABLES VARIABLE DIMENSIONING OF DATA MATRICES IN
C     THE CALLING PROGRAM.REFERENCE NUMBER (2).
C
C
      CALL ARRAY (2,MR,K,NRMAX,K,D,D)
```

```
C     HERE D IS CONVERTED FROM AN ARRAY OF SIZE (NRMAX,K) TO AN ARRAY OF
C     SIZE (MR,K)
C
C
C     SUBROUTINE MFGR DETERMINES RANK AND LINEARLY INDEPENDENT ROWS AND
C     COLUMNS OF A GIVEN MATRIX D OF SIZE (MR,K).REFERENCE (4).
C     DELTA IS A TEST VALUE FOR ZERO AFFECTED BY ROUNDOFF NOISE
C     IRANK IS THE RESULTANT RANK OF D
C     IROW IS AN INTEGER VECTOR OF LENGTH MR CONTAINING THE SUBSCRIPTS
C     OF BASIC ROWS IN IROW(1),....,IROW(IRANK)
C     ICOL IS AN INTEGER VECTOR OF LENGTH K CONTAINING THE SUBSCRIPTS OF
C     BASIC COLUMNS IN ICOL(I) UPTO ICOL(IRANK)
C
C
      CALL MFGR (D,MR,K,DELTA,IRANK,IROW,ICOL)
C
      CALL ARRAY (1,MR,K,NRMAX,K,D,D)
C     HERE D IS RECONVERTED FROM AN ARRAY OF SIZE (NR,K) TO AN ARRAY OF
C     SIZE (NRMAX,K)
C
      IF (IRANK.NE.MR) GO TO 6
      DO 4 I=1,MR
      DO 3 J=1,MR
      D(J,I)=GRAD(IROW(I),ICOL(J))
    3 CONTINUE
    4 CONTINUE
C
      CALL ARRAY (2,MR,MR,NRMAX,K,D,D)
C     HERE D IS CONVERTED FROM AN ARRAY OF SIZE (NRMAX,K) TO AN ARRAY OF
C     SIZE (MR,K)
C
C
C     SUBROUTINE MINV INVERTS SQUARE MATRIX D OF SIZE (MR,MR) AND STORES
C     THE RESULT IN D. DET IS THE DETERMINANT OF THE ORIGINAL MATRIX D,
C     WHILE LL AND MM ARE WORK VECTORS OF SIZE MR. REFERENCE NUMBER (3)
C
C
      CALL MINV (D,MR,DET,LL,MM)
C
      CALL ARRAY (1,MR,MR,NRMAX,K,D,D)
C     HERE D IS RECONVERTED FROM AN ARRAY OF SIZE (NR,K) TO AN ARRAY OF
C     SIZE (NRMAX,K)
C
      ISP=.T.
      IF (ABS(DET).LE.1.0E-10) GO TO 5
      RETURN
    5 CONTINUE
      IF (IPRINT) WRITE (UNIT,16) MR
      RETURN
    6 CONTINUE
      IRANK1=IRANK+1
      IF (IRANK1.GE.MR) GO TO 7
      ISP=.T.
      RETURN
    7 CONTINUE
      DO 9 I=1,IRANK1
      DO 8 J=1,IRANK
```

```
      JJ=ICOL(J)
      H(J,I)=GRAD(I,JJ)
8     CONTINUE
9     CONTINUE
      DO 10 J=1,IRANK1
      H(IRANK1,J)=1.
10    CONTINUE
      DO 11 J=1,IRANK
      Q(J)=0.
11    CONTINUE
      Q(IRANK1)=1.
C
C
C     SUBROUTINE SOLVE SOLVES A SET OF LINEAR SIMULTANEOUS EQUATIONS.
C     H IS A MATRIX OF ROW SIZE IRANK1 IN AN ARRAY ROW DIMENSION K1.
C     A SQUARE SUBMATRIX OF H OF SIZE (IRANK1,IRANK1) IS PART OF THE
C     THE MATRIX EQUATION ON THE LEFTHAND SIDE ,WHILE Q IS INITIALLY THE
C     VECTOR ON THE RIGHTHAND SIDE.Q IS FINALLY THE SOLUTION VECTOR.
C     IDET IS DEFINED BY 2**ID .LT. ABS(DET) .LT. 2**(ID+1),WHERE DET IS
C     THE DETERMINANT OF THE SUBMATRIX.
C     REFFRENCE NUMBER (7)
C     A LISTING OF THIS SUBROUTINE IS ATTACHED TO THE PACKAGE
C
C
      CALL SOLVE (H,Q,IDET,IRANK1,K1)
      DO 12 J=1,IRANK1
      X2(J)=Q(J)
12    CONTINUE
      DO 14 J=1,K
      JJ=ICOL(J)
      Q(J)=0.
      DO 13 I=1,MR
      Q(J)=Q(J)+X2(I)*GRAD(I,JJ)
      R2(JJ)=Q(J)
13    CONTINUE
14    CONTINUE
      X2SUM=0.
      DO 15 I=1,MR
      X2SUM=X2SUM+X2(I)
15    CONTINUE
      RETURN
C
16    FORMAT (10X,I2,98X,*USER IS ADVISED TO*/110X,*INCREASE VALUE OF DE
     1LTA*//)
      END
C
C
C
C     ******************************************************************
      FUNCTION ANORM1 (K,B1)
C     THE EUCLIDEAN NORM OF VECTOR B1 IS CALCULATED HERE
      DIMENSION B1(1)
      ANORM1=0.
      DO 1 I=1,K
      ANORM1=ANORM1+B1(I)*B1(I)
1     CONTINUE
      ANORM1=SQRT(ANORM1)
```

```
      RETURN
      END
C
C
C
C     ............................................................
      FUNCTION ANORM2 (K,B1)
C     MAX(ABS(B1(1)),ABS(B1(2)),....,ABS(B1(K))) IS CALCULATED HERE
      DIMENSION B1(1)
      ANORM2=ABS(B1(1))
      IF (K.LT.2) GO TO 2
      DO 1 I=2,K
      ABSB1=ABS(B1(I))
      IF (ABSB1.GT.ANORM2) ANORM2=ABSB1
1     CONTINUE
2     RETURN
      END
C
C
C
C     ............................................................
      SUBROUTINE SOLVE (A,X,ID,N,NA)
      DIMENSION A(NA,1), X(1)
      D=0.
      DATA DIV/.6931471181/
      DO 6 I=1,N
      AA=0.
      DO 1 J=I,N
      AB=ABS(A(J,I))
      IF (AB.LE.AA) GO TO 1
      K=J
      AA=AB
1     CONTINUE
      D=D+ALOG(AA)
      IF (I.EQ.N) GO TO 7
      IF (K.EQ.I) GO TO 3
      DO 2 J=I,N
      AB=A(I,J)
      A(I,J)=A(K,J)
      A(K,J)=AB
2     CONTINUE
      AB=X(I)
      X(I)=X(K)
      X(K)=AB
3     I1=I+1
      DO 5 J=I1,N
      AA=-A(J,I)/A(I,I)
      A(J,I)=0.
      DO 4 K=I1,N
      A(J,K)=A(J,K)+AA*A(I,K)
4     CONTINUE
      X(J)=X(J)+AA*X(I)
5     CONTINUE
6     CONTINUE
7     ID=D/DIV
      X(N)=X(N)/A(N,N)
      DO 9 II=2,N
```

```
      I=N+1-II
      I1=I+1
      AA=0.
      DO 8 J=I1,N
      AA=AA+A(I,J)*X(J)
8     CONTINUE
      X(I)=(X(I)-AA)/A(I,I)
9     CONTINUE
      RETURN
      END
C
C
C
C     REFERENCES     (1)J.W.BANDLER,'CONDITIONS FOR A MINIMAX OPTIMUM',
C                       IEEE TRANS CIRCUIT THEORY (CORRESP.),VOL. CT-18,
C                       PP 476-479,JULY 1971
C                    (2)SUBROUTINE ARRAY,P98,SYSTEM/360 SCIENTIFIC
C                       SUBROUTINE PACKAGE,VERSION 3,IBM,PROGRAM NUMBER
C                       360A-CM-03X
C                    (3)SUBROUTINE MINV,P118,SYSTEM/360 SCIENTIFIC
C                       SUBROUTINE PACKAGE,VERSION 3,IBM,PROGRAM NUMBER
C                       360A-CM-03X
C                    (4)SUBROUTINE MFGR,P127,SYSTEM/360 SCIENTIFIC
C                       SUBROUTINE PACKAGE,VERSION 3,IBM,PROGRAM NUMBER
C                       360A-CM-03X
C                    (5)SUBROUTINE SIMPLE,DATA PROCESSING AND COMPUTING
C                       CENTRE,LIBRARY INFORMATION SHEET MILIS 15.02.01,
C                       P130,MCMASTER UNIVERSITY,HAMILTON,ONTARIO,CANADA
C                    (6)J.W.BANDLER AND T.V.SRINIVASAN,'THE GRAZOR SEARCH
C                       PROGRAM FOR MINIMAX OBJECTIVES',IEEE TRANS
C                       MICROWAVE THEORY TECH.,(PROGRAM DESCRIPTION),
C                       VOL. MTT-20, PP 784-785, NOVEMBER 1972
C                    (7)SUBROUTINE SOLVE, DATA PROCESSING AND COMPUTING
C                       CENTRE,LIBRARY INFORMATION SHEET MILIS 45.04.16,
C                       P 27,MCMASTER UNIVERSITY,HAMILTON,ONTARIO,CANADA
```

REFERENCES

J. H. Anderson (1967), "Geometrical approach to reduction of dynamic systems", Proc. IEE, vol. 114, pp.1014-1018.

J. W. Bandler (1969a), "Optimization methods for computer-aided design", IEEE Trans. Microwave Theory Tech., vol. MTT-17, pp.533-552.

J. W. Bandler (1969b), "Computer optimization of inhomogeneous waveguide transformers", IEEE Trans. Microwave Theory Tech., vol. MTT-17, pp.563-571.

J. W. Bandler (1971), "Conditions for a minimax optimum", IEEE Trans. Circuit Theory, vol. CT-18, pp.476-479.

J. W. Bandler (1972), "Optimization methods and microwave circuit design", Proc. 6th Princeton Conf. Information Sciences and Systems (Princeton, N.J.), pp.10-14.

J. W. Bandler and B. L. Bardakjian (1973), "Least pth optimization of recursive digital filters", IEEE Int. Symp. Circuit Theory (Toronto, Canada), pp.377-380.

J. W. Bandler and C. Charalambous (1971), "On conditions for optimality in least pth approximation with $p \to \infty$", Proc. 9th Allerton Conf. Circuit and System Theory (Urbana, Ill.), pp.404-413.

J. W. Bandler and C. Charalambous (1972a), "A new approach to nonlinear programming", Proc. 5th Hawaii Int. Conf. System Sciences (Honolulu, Hawaii), pp.127-129.

J. W. Bandler and C. Charalambous (1972b), "Conditions for minimax approximation obtained from the ℓ_p norm", IEEE Trans. Automatic Control, vol. AC-17, pp.257-258.

J. W. Bandler and C. Charalambous (1972c), "Theory of generalized least pth approximation", IEEE Trans. Circuit Theory, vol. CT-19, pp.287-289.

J. W. Bandler and C. Charalambous (1972d),"Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, pp.834-840.

J. W. Bandler and C. Charalambous (1973a), "On conditions for optimality in least pth approximation with $p\to\infty$", J. Optimization Theory and Applications, vol. 11.

J. W. Bandler and C. Charalambous (1973b), "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, accepted for publication.

J. W. Bandler, C. Charalambous and S. K. Tam (1972), "Computer-aided equal ripple design of lumped-distributed-active filters", Int. Filter Symp. (Santa Monica, Calif.), pp.79-80.

J. W. Bandler and V. K. Jha (1972), "Network optimization computer program package", CRL Internal Report Series No. CRL-5, Communications Research Laboratory, Department of Electrical Engineering, McMaster Univ., Hamilton, Canada.

J. W. Bandler and A. G. Lee-Chan (1971), "Gradient razor search method for optimization ", IEEE Int. Microwave Symp. Dig. (Washington, D.C.), pp.118-119.

J. W. Bandler and P. A. Macdonald (1969a), "Cascaded noncommensurate transmission-line networks as optimization problems", IEEE Trans. Circuit Theory, vol. CT-16, pp.391-394.

J. W. Bandler and P. A. Macdonald (1969b), "Optimization of microwave networks by razor search", IEEE Trans. Microwave Theory Tech., vol. MTT-17, pp.522-562.

J. W. Bandler and P. A. Macdonald (1971), "The razor search program", IEEE Trans. Microwave Theory Tech. (Comput. Program Descr.), vol. MTT-19, p.667.

J. W. Bandler, N. D. Markettos and N. K. Sinha (1973), "Optimum system modelling using recent gradient methods", Int. J. Systems Science, vol. 4, pp.33-44.

J. W. Bandler, N. D. Markettos and T. V. Srinivasan (1972), "A comparison of recent minimax techniques on optimum system modelling", Proc. 6th Princeton Conf. Information Sciences and Systems (Princeton, N.J.), pp.540-544.

J. W. Bandler, N. D. Markettos and T. V. Srinivasan (1973), "Gradient minimax techniques for system modelling", Int. J. Systems Science, vol. 4, pp.317-331.

J. W. Bandler and R. E. Seviora (1970), "Current trends in network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-18, pp.1159-1170.

J. W. Bandler and T. V. Srinivasan (1971), "A new gradient algorithm for minimax optimization of networks and systems", Proc. 14th Midwest Symp. Circuit Theory (Denver, Colo.), pp.16.5.1-16.5.11.

J. W. Bandler and T. V. Srinivasan (1972), "The grazor search program for minimax objectives", IEEE Trans. Microwave Theory Tech., vol. MTT-20, pp.784-785.

J. W. Bandler and T. V. Srinivasan (1973a), "Constrained minimax optimization by grazor search", Proc. 6th Hawaii Int. Conf. System Sciences (Honolulu, Hawaii), pp.125-128.

J. W. Bandler and T. V. Srinivasan (1973b), "On realistic minimax modelling of high-order systems", Proc. 7th Princeton Conf. Information Sciences and Systems (Princeton, N.J.).

J. W. Bandler and T. V. Srinivasan (1973c), "Practical investigation of conditions for minimax optimality", Proc. 16th Midwest Symp. Circuit Theory (Waterloo, Canada), pp.XX.4.1-XX.4.10.

J. W. Bandler and T. V. Srinivasan (1973d), "Program for investigating minimax optimality conditions", IEEE Trans. Microwave Theory Tech., vol. MTT-21, p.433.

J. W. Bandler and T. V. Srinivasan (1973e), "Computer-aided system modelling and design", Proc. Int. Conf. Systems and Control (Coimbatore, India).

J. W. Bandler, T. V. Srinivasan and C. Charalambous (1972), "Minimax optimization of networks by grazor search", IEEE Trans. Microwave Theory Tech., vol. MTT-20, pp.596-604.

G. T. Bereznai (1971), "Adaptive nuclear control based on optimal low-order linear models", Ph.D. Dissertation, McMaster Univ., Hamilton, Canada.

C. Brancher, F. Maffioli and A. Premoli (1970), "Computer optimisation of cascaded noncommensurable-line lowpass filters", Electronics Letters, vol. 6, pp.513-515.

H. J. Carlin (1971), "Distributed circuit design with transmission line elements", Proc. IEEE, vol. 59, pp.1059-1081.

H. J. Carlin and O. P. Gupta (1969), "Computer design of filters with lumped-distributed elements or frequency variable terminations", IEEE Trans. Microwave Theory Tech., vol. MTT-17, pp.598-604.

C. W. Carroll (1961), "The created response surface technique for optimizing nonlinear, restrained systems", Operations Research, vol. 9, pp.169-185.

C. Charalambous (1973), "Nonlinear least pth approximation with applications", Ph.D. Dissertation, McMaster Univ., Hamilton, Canada.

C. F. Chen and L. S. Shieh (1968), "A novel approach to linear model simplifications", Proc. JACC (Ann Arbor, Mich.), pp.454-460.

M. R. Chidambara (1969), "Two simple techniques for the simplification of large dynamic systems", Proc. JACC, pp.669-674.

A. R. Curtis and M. J. D. Powell (1966), "Necessary conditions for a minimax approximation", Computer J., vol. 8, pp.358-361.

E. J. Davison (1966), "A method for simplifying linear dynamic systems", IEEE Trans. Automatic Control, vol. AC-11, pp.93-101.

V. F. Dem'yanov (1970), "Sufficient conditions for minimax problems", Zh. Uychisl. Mat. Fiz., vol. 10, pp.1107-1115.

S. W. Director and R. A. Rohrer (1969), "The generalized adjoint network and network sensitivities", IEEE Trans. Circuit Theory, vol. CT-16, pp.318-323.

R. C. Dorf (1967), Modern Control Systems. Reading, Mass. : Addison-Wesley, pp.167-168.

A. V. Fiacco and G. P. McCormick (1964a), "The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method", Management Science, vol. 10, pp.360-366.

A. V. Fiacco and G. P. McCormick (1964b), "Computational algorithm for the sequential unconstrained minimization technique for non-linear programming", Management Science, vol. 10, pp.601-617.

R. Fletcher (1970), "A new approach to variable metric algorithms", Computer J., vol. 13, pp.317-322.

R. Fletcher and M. J. D. Powell (1963), "A rapidly convergent descent method for minimization", Computer J., vol. 6, pp.163-168.

W. T. Hatley, Jr. (1967), "Computer synthesis of wide-band impedance-matching networks", Tech. Rept. 6657-2, Stanford Electronics Laboratories, Stanford Univ., Stanford, Calif.

J. E. Heller (1969), "A gradient algorithm for minimax design", Report R-406, Coordinated Science Lab., Univ. of Illinois, Urbana.

R. Hooke and T. A. Jeeves (1961), " 'Direct search' solution of numerical and statistical problems", J. Ass. Comput. Mach., vol. 8, pp.212-229.

Y. Ishizaki and H. Watanabe (1968), "An iterative Chebyshev approximation method for network design", IEEE Trans. Circuit Theory, vol. CT-15, pp.326-336

D. H. Jacobson and W. Oksman (1972), "An algorithm that minimizes homogeneous functions of N variables in N+2 iterations and rapidly minimizes general functions", J. Math. Anal. Applics., vol. 38, pp.535-552.

P. Kokotović and P. Sannuti (1968), "Single perturbation method for reducing the model order in optimal design", Proc. JACC (Ann Arbor, Mich.), pp.468-476.

H. W. Kuhn and A. W. Tucker (1951), "Nonlinear programming", Proc. 2nd Berkeley Symp. Mathematical Statistics and Probability (Berkeley, Calif.), pp.481-492.

L. S. Lasdon (1970), Optimization Theory for Large Systems. New York : Macmillan.

R. Levy (1965), "Tables of element values for the distributed low-pass prototype filter", IEEE Trans. Microwave Theory Tech., vol. MTT-13, pp.514-536.

N. D. Markettos (1972), "Optimum system modelling using recent gradient methods", M. Eng. Dissertation, McMaster Univ., Hamilton, Canada.

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

G. L. Matthaei, L. Young and E. M. T. Jones (1964), Microwave Filters, Impedance Matching Networks and Coupling Structures. New York : McGraw-Hill.

J. Medanic (1970), "Solution of the convex minimax problem by the Newton-Raphson method", Proc. 8th Allerton Conf. Circuit and System Theory (Urbana, Ill.), pp.13-22.

D. Mitra (1969), "The reduction of complexity of linear time-invariant dynamic systems", Proc. 4th IFAC Congress (Warsaw, Poland), pp.19-33.

M. R. Osborne and G. A. Watson (1969), "An algorithm for minimax approximation in the non-linear case", Computer J., vol.12, pp.63-68.

J. R. Popović (1972), "General programs for least pth and near minimax approximation", M. Eng. Dissertation, McMaster Univ., Hamilton, Canada.

R.E.Seviora, M.Sablatash and J. W. Bandler (1970), "Least pth and minimax objectives for automated network design", Electronics Letters, vol. 6, pp.14-15.

N. K. Sinha and G. T. Bereznai (1971), "Optimum approximation of high-order systems by low-order models", Int. J. Control, vol. 14, pp.951-959.

N. K. Sinha and W. Pille (1971), "A new method for reduction of dynamic systems", Int. J. Control, vol. 14, pp.111-118.

Subroutine ARRAY, System/360 Scientific Subroutine Package, Version III, IBM Programmer's Manual Number 360A-CM-03x, p.98.

Subroutine MFGR, System/360 Scientific Subroutine Package, Version III, IBM Programmer's Manual Number 360A-CM-03x, p.127.

Subroutine MINV, System/360 Scientific Subroutine Package, Version III, IBM Programmer's Manual Number 360A-CM-03x, p.118.

Subroutine SIMPLE, Data Processing and Computing Centre, Library Information Sheet MILIS 15.02.01, McMaster Univ., Hamilton, Canada, p.130.

Subroutine SOLVE, Data Processing and Computing Centre, Library Information Sheet MILIS 45.04.16, McMaster Univ., Hamilton, Canada, p.27.

Subroutine TGSORT, Data Processing and Computing Centre, Library Information Sheet MILIS 5.3.34, McMaster Univ., Hamilton, Canada.

G. C. Temes (1969), "Optimization methods in circuit design", in Computer Oriented Circuit Design, F. F. Kuo and W. G. Magnuson Jr., Eds. Englewood Cliffs, N.J. : Prentice Hall.

G. C. Temes and D. Y. F. Zai (1969), "Least pth approximation", IEEE Trans. Circuit Theory, vol. CT-16, pp.235-237.

A. D. Waren, L. S. Lasdon and D. F. Suchman (1967), "Optimization in engineering design", Proc. IEEE, vol. 55, pp.1885-1897.

G. A. Watson (1970), "On an algorithm for nonlinear minimax approximation", Commun. Ass. Comput. Mach., vol. 13, pp.160-162.

W. I. Zangwill (1969), Nonlinear Programming : A Unified Approach. Englewood Cliffs, N.J., : Prentice Hall.

AUTHOR INDEX