Sixteenth Midwest Symposium on Circuit Theory

University of Waterloo
Waterloo, Ontario, Canada
April 12 - 13, 1973

Proceedings, vol. 2

siam

PRACTICAL INVESTIGATION OF CONDITIONS FOR MINIMAX OPTIMALITY

John W. Bandler and T.V. Srinivasan
Department of Electrical Engineering
McMaster University
Hamilton, Ontario, Canada.

Abstract

This paper investigates the practical implementation of testing the optimality conditions of a solution to an approximation problem with minimax objectives. Two methods are presented to test the conditions, one using a linear programming approach, and the other the solution of a set of linear independent equations. Full details of a user-oriented computer program written in Fortran IV are given so that the user may choose a number of options to define the optimality of a design or a proposed solution in a meaningful way. A suitable example has been chosen to indicate how the program can be handled by the user.

INTRODUCTION

Optimization techniques for minimax objectives have been of great interest to the system designer, and a lot of momentum has been gained in the last few years in this area. Computer-aided design for minimax and near-minimax objectives have been carried out using both the direct search [1] and gradient algorithms [2-5] on a variety of problems, including optimization of electrical networks and systems [6,7].

Depending on the optimization method employed, a satisfactory solution may be obtained for a problem after a number of iterations of the algorithm on the computer. Once a solution for minimax objectives is obtained, it may be required to investigate the solution for conditions for minimax optimality [8] so as to verify whether the solution is optimal or not. Though the necessary optimality conditions may seem to be straightforward to verify,

---

This work was supported by the National Research Council of Canada under grants A7239 and C154. A listing of the program is available from J.W. Bandler at a nominal charge.

it is both tedious and difficult to implement in practice.

CONDITIONS FOR A MINIMAX OPTIMUM [8]

Let $\hat{y}_\ell(\underset{\sim}{\phi})$, $\ell=1,\ldots, n_r$ be n_r discrete maxima in descending magnitude at a given point $\underset{\sim}{\phi}$ in a k-parameter space. If $\hat{y}_\ell(\underset{\sim}{\phi})$, $\ell=1,\ldots,k_r$ are taken as equal, then for $\underset{\sim}{\phi}$ to satisfy the necessary conditions for a minimax optimum, there exist $u_\ell \geq 0$ for $\ell=1,\ldots,k_r$ such that

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

$$\sum_{\ell=1}^{k_r} u_\ell = 1 \tag{2}$$

When testing the conditions for optimality at a point $\underset{\sim}{\phi}$, we attempt to solve (1) and (2) for $k_r=1,2,\ldots$ until for a value of k_r^* ($\leq n_r$) (1) and (2) are satisfied. If this is not possible then the necessary conditions are not satisfied.

PRACTICAL IMPLEMENTATION

A computer program has been developed which can test a solution for the necessarv conditions for a minimax optimum by two different formulations. One uses a linear programming approach, and the other the solution of a set of linear independent equations.

<u>Method 1</u>

Equations (1) and (2) are solved here by minimizing $u_{k_r+1} \geq 0$ such that (2) is satisfied and

$$\left| \sum_{\ell=1}^{k_r} u_\ell \frac{\partial \hat{y}_\ell}{\partial \phi_i} \right| \le u_{k_r+1}, \quad i=1,2,\ldots,k \qquad (3)$$

Linear programming ensures that $u_\ell \ge 0$ for $\ell=1,2,\ldots,k_r+1$.

Method 2

Here, we solve a set of linearly independent equations

$$\sum_{\ell=1}^{k_r} u_\ell \frac{\partial \hat{y}_\ell}{\partial \phi_i} (\underset{\sim}{\phi}) = 0, \quad i \epsilon J \qquad (4)$$

and (2), where J is a suitable subset of $\{1,2,\ldots,k\}$.

There is no guarantee, however, that $u_\ell \ge 0$ for $\ell=1,\ldots,k_r$. When k_r-1 is greater than the number of elements of J, the system of equations (2) and (4) have more unknowns that equations, and we use Method 1 to get the u_ℓ.

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, $\underset{\sim}{\hat{y}}([\hat{y}_1 \;.\;.\;.\; \hat{y}_{n_r}]^T)$,

$(\nabla\hat{y}^T)^T([\nabla\hat{y}_1 \; . \; . \; . \; \nabla\hat{y}_{n_r}]^T)$, 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 $\sum_{\ell=1}^{m_{r1}} u_{1\ell} \nabla\underset{\sim}{\hat{y}}_\ell$, $\sum_{\ell=1}^{m_{r2}} u_{2\ell} \nabla\underset{\sim}{\hat{y}}_\ell$
$\|\underset{\sim}{r}_1\|, \|\underset{\sim}{r}_2\|$	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 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 RELTOL. The necessary conditions for optimality are satisfied when the norm $\|\underset{\sim}{r}\|$ of the residual vector $\underset{\sim}{r} = \sum_{\ell=1}^{m_r} u_\ell \nabla\underset{\sim}{\hat{y}}_\ell$, for $m_r = 1,\ldots,k_r$ falls within a user-specified value ε for $u_\ell \geq 0$ and (2) is satisfied, in which case OPTIM is set

.TRUE.. The values of ε and δ as specified by the user should be realistic so that the program may give meaningful results.

REQUIRED SUBPROGRAMS

The user has to have a subprogram by which the discrete values of 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 References [9]-[13].

ARRAY converts data array from single to double dimension or vice versa [9] while MINV inverts a matrix and calculates its determinant [10]. MFGR determines the rank and linearly independent rows and columns of a given matrix [11]. SIMPLE is a linear-program solving subroutine [12],[13], and SOLVE solves a set of linear simultaneous equations [14].

EXAMPLE

The problem chosen was the lower-order modelling of a ninth-order nuclear reactor system [15] when the operating reactor power level is in the 90-100% range of the full power. 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 by the grazor search method [5], and the program was used to test the solution for optimality.

Fig. 1 shows a typical printout of the package for this example. The input parameters are: $k=2$, $\hat{n}_r=15, n_r=4, \delta=10^{-4}, \varepsilon=10^{-6}$, ICRIT=1, IDATA=IPRINT=.T., MET=3, NORM=2, RELTOL=0.01, UNIT=6, $k_1=3$, $k_3=5$, $m_{r3}=20$, while $\nabla \hat{y}$ is given by

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

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

and $\hat{y}$ is given by

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

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

DISCUSSION

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. The program has been successfully applied to problems of minimax optimization involving filter designs [5] and system modelling [7].

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 $\hat{n}_r$=15.

ACKNOWLEDGEMENT

The authors wish to thank Dr. N.K. Sinha and Dr. G.T. Bereznai for providing the nuclear reactor example.

REFERENCES

[1] J.W. Bandler and P.A. Macdonald, "Optimization of microwave networks by razor search", IEEE Trans. Microwave Theory Tech., vol. MTT-17, pp. 552-562, Aug. 1969.

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

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

[4] J.E. Heller, "A gradient algorithm for minimax design", Coordinated Science Lab., University of Illinois, Urbana, Rep. R-406, Jan. 1969.

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

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

[7] J.W. Bandler, N.D. Markettos and T.V. Srinivasan, "Gradient minimax techniques for system modelling", Int. J. of Systems Science, to be published.

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

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

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

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

[12] Subroutine SIMPLE, Data Processing and Computing Centre, Library Information Sheet MILIS 15.02.01, page 130, McMaster University, Hamilton, Canada.

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

[14] Subroutine SOLVE, Data Processing and Computing Centre, Library Information Sheet MILIS 45.04.16, page 27, McMaster University, Hamilton, Canada.

[15] G.T. Bereznai, "Adaptive nuclear reactor control based on optimal low-order linear models," McMaster University, Hamilton, Canada, Ph.D.Thesis, Dec. 1971.

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. 1. Typical printout of results for the example given in the text.