Old Dominion University Research Foundation

[redacted] 120336

IN

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529-0247

SINGLE-MODE PROJECTION FILTERS FOR MODAL PARAMETER IDENTIFICATION FOR FLEXIBLE STRUCTURES

By

Jen-Kuang Huang, Principal Investigator

and

Chung-Wen Chen

Final Report
For the period ended December 31, 1987

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

Under
NASA Research Grant NAG1-655
Jer-Nan Juang, Technical Monitor
SDD-Structural Dynamics Branch

February 1988

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529-0247

SINGLE-MODE PROJECTION FILTERS FOR MODAL PARAMETER IDENTIFICATION FOR FLEXIBLE STRUCTURES

By

Jen-Kuang Huang, Principal Investigator

and

Chung-Wen Chen

Final Report
For the period ended December 31, 1987

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

Under
NASA Research Grant NAG1-655
Jer-Nan Juang, Technical Monitor
SDD-Structural Dynamics Branch

Submitted by the
Old Dominion University Research Foundation
P. O. Box 6369
Norfolk, Virginia 23508

February 1988

Single-Mode Projection Filters for Modal Parameter Identification for Flexible Structures

by

Jen-Kuang Huang[1] and Chung-Wen Chen[2]

Abstract

Single-mode projection filters are developed for eigensystem parameter identification from both analytical results and test data. Explicit formulations of these projection filters are derived using the orthogonal matrices of the controllability and observability matrices in the general sense. A global minimum optimization algorithm is applied to update the filter parameters by using the interval analysis method. The updated modal parameters represent the characteristics of the test data. For illustration of this new approach, a numerical simulation for the MAST beam structure is shown by using a one-dimensional global optimization algorithm to identify modal frequencies and damping. Another numerical simulation of a ten-mode structure is also presented by using a two-dimensional global optimization algorithm to illustrate the feasibility of the new method. The projection filters are practical for parallel processing implementation.

[1] Assistant Professor, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23529-0247.

[2] Research Assistant, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23529-0247.

TABLE OF CONTENTS

Page

Chapter

Chapter 1

INTRODUCTION

The problems of deriving control algorithms and state estimators for maneuvering flexible structures have been investigated by many researchers in recent years. The control design demands an accurate model of the system dynamics which will adequately describe the dynamic behavior of the system. System identification methods use experimental measurements to estimate dynamic properties such as natural frequencies, damping factors, mode shapes and modal masses which are referred to as modal parameters. Several different time-domain and frequency-domain methods are possible for the identification of structures. Various techniques may share the same mathematical foundation via system realization theory[1]. However, most techniques do not account explicitly for the factors which affect the actual performance in practice significantly. These factors include nonlinearities, local modes, and system and measurement noise. In order to achieve the final purpose of identification, i.e., control of flexible structures, an on-line estimation technique needs to be used.

For linear time-invariant systems, optimal model-reduction and state estimation has been developed via optimal projection equations based on modified Riccati and Lyapunov equations[2]. Other filtering approaches in both time and frequency domains are easy to implement and effective in rejecting uncorrelated measurement noise from simulated data[3]. However, previous time-domain filters usually involve an unacceptable computational burden for multimode systems such as large flexible structures.

In this paper, simple projection filters are developed using system realization theory which has been considerably used in deriving system identification methods[1]. The development of projection filters was initially motivated by the need of an on-line technique for state estimation of linear dynamical systems. Because of modeling errors due to system uncertainties, the projection filters are naturally required to be verified and updated from measurement data. The main objective of this paper is to present a novel approach to update the projection filters which in turn yields the modal parameter identification for linear dynamical systems.

Each projection filter is formulated with a single mode only and its explicit expression can be derived using the orthogonal matrices of the controllability and observability matrices in the general sense[4]. The modal parameters (including modal frequency, modal damping and mode shapes) required for formulating the projection filters are initially obtained from an analytical model in modal space. The experimental data are then passed through the projection filters to determine whether there is a discrepancy between the analytical model and the experimental testing. Each projection filter is then updated by varying the corresponding modal parameters to minimize a cost function defined by the norm of a specified error matrix. Both one and two-dimensional global minimum optimization algorithms[5-8] are applied for the filter update by using the interval analysis method which guarantees finding the smallest value of a cost function throughout a specified closed region of modal parameters. The updated projection filters thus produce the modal parameter identification for the system.

Finally, two numerical examples, one for the MAST truss beam structure[9] and another for a ten-mode structure are given to illustrate this new method.

Chapter 2

PROJECTION FILTERS FORMULATIONS

The projection filters are developed using system realization theory. A finite-dimensional, linear, time-invariant dynamic system can be represented by the state-variable equations in discrete-time form:

$$q(k+1) = A\, q(k) + B\, u(k) \qquad (1)$$

$$y(k) = C\, q(k) \qquad (2)$$

where q is an n-dimensional state vector, u is an m-dimensional control or input vector, and y is a p-dimensional measurement or output vector. The integer k is the sample indicator. For flexible structures, the state transition matrix A is a representation of mass, stiffness, and damping properties. The control influence matrix B characterizes the locations and type of input control vector u. The measurement influence matrix C describes the relationship between the state vector q and the output measurement vector y, and characterizes the mode shapes of the system.

For the state-variable Eqs. (1) and (2) with free pulse response, the time domain description is given by the function known as the Markov parameter

$$Y(k) = CA^{k-1} B \qquad (3)$$

or in the case of initial state response (zero input)

$$Y(k) = CA^{k}\, q(0)$$

where q(0) represents the initial conditions of the state vector and k is an integer. The functions Y(k) can be obtained from the measured data and used to form the (r+1) by (s+1) block data matrix (generalized Hankel matrix)

$$H(k-1) = \begin{bmatrix} Y(k) & Y(k+t_1) \ldots\ldots & Y(k+t_s) \\ Y(j_1+k) & Y(j_1+k+t_1) \ldots. & Y(j_1+k+t_s) \\ \vdots & \vdots & \vdots \\ Y(j_r+k) & Y(j_r+k+t_1) \ldots. & Y(j_r+k+t_s) \end{bmatrix} \qquad (4)$$

where $j_i (i=1,\ldots,r)$ and $t_i (i=1,\ldots,s)$ are arbitrary integers. For the system with initial state response measurements, simply replace $H(k-1)$ by $H(k)$.

From Eqs. (3) and (4), it can be shown that

$$H(k) = V_r \, A^k \, W_s; \quad V_r = \begin{bmatrix} C \\ CA^{j_1} \\ \vdots \\ CA^{j_r} \end{bmatrix}$$

and

$$W_s = [B, \; A^{t_1} B, \ldots, \; A^{t_s} B] \qquad (5)$$

where V_r and W_s are generalized observability and controllability matrices respectively. The dimensions of V_r and W_s are $(r+1)p \times n$ and $n \times m(s+1)$ respectively. Now observe that

$$H(0) = V_r \, W_s \qquad (6)$$

If A is nonsingular, $(r+1)\, p > n$ and $m(s+1) > n$, we can derive

$$V_r^{\#} \; H(0) \; W_s^{\#} = I_n \quad \text{with} \quad V_r^{\#} V_r = W_s \, W_s^{\#} = I_n \qquad (7)$$

where $V_r^{\#}$ and $W_s^{\#}$ are the orthogonal matrices of V_r and W_s, respectively, and I_n is an identity matrix of order n. Now, instead of using the matrices $V_r^{\#}$ and $W_s^{\#}$ for an n-dimensional multi-mode system, we develop simpler forms of $V^{\#}$ and $W^{\#}$ which represent the orthogonal matrices of the respective generalized observability and controllability matrices derived from a single-mode model only. Note that $V^{\#}$ and $W^{\#}$ are rectangular matrices with dimensions

2x(r+1)p and m(s+1)x2 respectively. The general explicit expressions of $V^{\#}$ and $W^{\#}$ will be derived later. The matrices $V^{\#}$ and $W^{\#}$, which are formulated only for the specific modes of interest from the analytical results, will be used as the left and right projection filters respectively. The Hankel matrix H(0), which is formed from measured data, will then pass through the projection filters to identify the modal parameters which represent the characteristics of the measured data. If the modes of the measured data are uncoupled and distinct and the projection filters have the same modal characteristics as the measured data have, then from Eq. (7) we have

$$V^{\#} H(0) W^{\#} \approx I_2 \tag{8}$$

where I_2 is a 2x2 identity matrix. The approximate equality in Eq. (8) is caused by the finite sampling time T used for these digital filters. If T is equal to zero for a finite data length, we have (see Appendix I for proof)

$$V^{\#} H(0) W^{\#} = I_2 \tag{9}$$

On the otherhand, if Eq. (8) does not hold, it indicates that the modal parameters of the projection filters are different from those of the measured data. The projection filters should be tuned in order to match the modes of the measured data. The algorithm for the filter update is developed in the next section.

Now, the explicit expressions of the projection filters $V^{\#}$ and $W^{\#}$ can be derived as follows. A single mode, continuous-time, linear, time-invariant dynamic system has the state-variable equations in modal space

$$\dot{x} = \bar{A} x + \bar{B} u \tag{10}$$

$$y = \bar{C}\ x \tag{11}$$

with

$$\bar{A} = \begin{bmatrix} -\sigma & \omega \\ -\omega & -\sigma \end{bmatrix} \tag{12}$$

where ω is the modal damped frequency (the imaginary part of the eigenvalue) and $-\sigma$ is the damping (the real part of the eigenvalue). If we denote the ω_n and ζ as the natural frequency and damping ratio respectively, then we have $\omega = \omega_n (1 - \zeta^2)^{1/2}$ and $\sigma = \zeta\ \omega_n$. The corresponding SISO discrete-time system can be represented by Eqs. (1) and (2) with

$$A = \begin{bmatrix} e^{-\sigma T} \cos\omega T & e^{-\sigma T} \sin\omega T \\ -e^{-\sigma T} \sin\omega T & e^{-\sigma T} \cos\omega T \end{bmatrix} \tag{13}$$

and

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \qquad C = [c_1 \quad c_2] \tag{14}$$

where b_1 , b_2, c_1, c_2 are scalars. From Eq. (5) with $j_o = t_o = 0$

$$V = \begin{matrix} CA^{j_o} \\ CA^{j_1} \\ \vdots \\ CA^{j_r} \end{matrix} = [c_2\ V_1(r) + c_1\ V_2\ (r),\ -\ c_1\ V_1(r) + c_2\ V_2(r)] \tag{15}$$

$$W = [A^{t_o} B,\ A^{t_1} B, \ldots,\ A^{t_s} B] = \begin{bmatrix} -b_2\ V_3^T(s) + b_1\ V_4^T(s) \\ b_1\ V_3^T(s) + b_2\ V_4^T(s) \end{bmatrix} \tag{16}$$

where

$$V_1^T(r) = [\ldots,\ V_{1i},\ \ldots],\ \ V_2^T(r) = [\ldots,\ V_{2i},\ \ldots] \tag{17}$$

$$V_{1i} = -e^{-j_i \sigma T} \sin(j_i \omega T),\ \ V_{2i} = e^{-j_i \sigma T} \cos(j_i \omega T) \tag{18}$$

with $i = 0, 1, 2, \ldots, r$

$$V_3^T(s) = [\ldots, V_{3k}, \ldots], \quad V_4^T(s) = [\ldots, V_{4k}, \ldots] \tag{19}$$

$$V_{3k} = -e^{-t_k\sigma T} \sin(t_k\omega T), \quad V_{4k} = e^{-t_k\sigma T} \cos(t_k\omega T) \tag{20}$$

with $k = 0, 1, 2, \ldots, s$

Assume we choose j_i, t_k as follows

$$j_r - j_{r-i} = j_i \qquad i = 0, 1, 2, \ldots, \text{integer } [\tfrac{r}{2}] \tag{21}$$

$$t_s - t_{s-k} = t_k \qquad k = 0, 1, 2, \ldots, \text{integer } [\tfrac{s}{2}] \tag{22}$$

Then, the projection filters, $V^\#$ and $W^\#$, have the following explicit expressions (see Appendix II for proof):

$$V^\# = \begin{bmatrix} (c_2\, V_a^\#(r) + c_1\, V_b^\#\,(r))/(c_1^2 + c_2^2) \\ (-c_1\, V_a^\#(r) + c_2\, V_b^\#\,(r))/(c_1^2 + c_2^2) \end{bmatrix} \tag{23}$$

$$W^\# = [(-b_2\, V_c^\#(s) + b_1\, V_d^\#(s))^T, (b_1\, V_c^\#(s) + b_2\, V_d^\#(s))^T]/(b_1^2 + b_2^2) \tag{24}$$

where

$$V_a^\#(r) = [\ldots, V_{ai}^\#, \ldots], \quad V_b^\#(r) = [\ldots, V_{bi}^\#, \ldots] \tag{25}$$

$$V_{ai}^\# = e^{j_i\sigma T}(-\sin(j_i\omega T)\ (1/\lambda_1(r) + 1/\lambda_2(r)) - \sin((j_r-j_i)\omega T)$$

$$(1/\lambda_1(r) -1/\lambda_2(r))/2$$

$$V^{\#}_{b_i} = e^{j_i \sigma T} \left(\cos(j_i\omega T)\ (1/\lambda_1(r) + 1/\lambda_2(r)) + \cos((j_r - j_i)\omega T)\right.$$

$$\left.(1/\lambda_1(r) - 1/\lambda_2(r)\right)/2 \tag{26}$$

with $i = 0,1,2,\ldots,r$

and

$$V^{\#}_c(s) = [\ldots,\ V^{\#}_{c_k},\ \ldots],\quad V^{\#}_d(s) = [\ldots,\ V_{d_k}^{\#},\ \ldots] \tag{27}$$

$$V^{\#}_{c_k} = e^{t_k \sigma T} (-\sin(t_k\omega T)\ (1/\lambda_3(s) + 1/\lambda_4(s)) - \sin((t_s - t_k)\omega T)$$

$$(1/\lambda_3(s) - 1/\lambda_4(s)))/2 ,$$

$$V^{\#}_{d_k} = e^{t_k \sigma T} \left(\cos(t_k\omega T)\ (1/\lambda_3(s) + 1/\lambda_4(s)) + \cos((t_s - t_k)\omega T)\right.$$

$$\left.(1/\lambda_3(s) - 1/\lambda_4(s))\right)/2 \tag{28}$$

with $k = 0,\ 1,\ 2,\ldots,\ s$

$\lambda_1(r)$ and $\lambda_2(r)$ are the eigenvalues of V^TV, and $\lambda_3(s)$ and $\lambda_4(s)$ are the eigenvalues of WW^T when $b_i = c_2 = 1$ and $b_2 = c_1 = 0$. $\lambda_1(r)$, $\lambda_2(r)$, $\lambda_3(s)$ and $\lambda_4(s)$ can be derived as follows

$$\lambda_1(r) = \begin{cases} 1+m+Y(m) & \text{if } r \text{ is even} \\ m+Y(m) & \text{if } r \text{ is odd} \end{cases} \tag{29}$$

$$\lambda_2(r) = m - Y(m) \tag{30}$$

with

$$m = \begin{cases} \frac{r}{2} & \text{if } r \text{ is even} \\ \frac{r+1}{2} & \text{if } r \text{ is odd} \end{cases}, \qquad Y(m) = \sum_{i=0}^{m-1} \cos((j_r - 2j_i)\omega T) \tag{31}$$

$$\lambda_3(s) = \begin{cases} 1+n+Z(n) & \text{if } s \text{ is even} \\ n+Z(n) & \text{if } s \text{ is odd} \end{cases} \tag{32}$$

$$\lambda_4(s) = n - Z(n) \tag{33}$$

with

$$n = \begin{cases} \frac{s}{2} & \text{if } s \text{ is even} \\ \frac{s+1}{2} & \text{if } s \text{ is odd} \end{cases}, \qquad Z(n) = \sum_{k=0}^{n-1} \cos((t_s - 2t_k)\omega T) \tag{34}$$

Note that $V^{\#}$ and $W^{\#}$ are rectangular matrices with dimensions $2x(r+1)$ and $(s+1)x2$ respectively.

The corresponding MIMO single-mode subsystem can be represented by Eqs. (1) and (2) with the A matrix shown in Eq. (13) and

$$B = \begin{bmatrix} \cdots & b_{1f} & \cdots \\ \cdots & b_{2f} & \cdots \end{bmatrix} \qquad C = \begin{bmatrix} \vdots & \vdots \\ C_{g1} & C_{g2} \\ \vdots & \vdots \end{bmatrix} \tag{35}$$

with $f = 1, 2, \ldots, m$ and $g = 1, 2, \ldots, p$.

Then the observability and controllability matrices V and W can be derived as follows

$$V = \begin{bmatrix} CA^{j_o} \\ CA^{j_1} \\ \vdots \\ CA^{j_r} \end{bmatrix} = \begin{bmatrix} \vdots \\ V_i \\ \vdots \end{bmatrix}, \qquad \begin{aligned} W &= [A^{t_o}, A^{t_1} B, \ldots, A^{t_s} B] \\ &= [\ldots, W_k, \ldots] \end{aligned} \tag{36}$$

where

$$V_i = \begin{bmatrix} C_{12}\, V_{1i} + C_{11}\, V_{2i} & -C_{11}\, V_{1i} + C_{12}\, V_{2i} \\ C_{22}\, V_{1i} + C_{21}\, V_{2i} & -C_{21}\, V_{1i} + C_{22}\, V_{2i} \\ \vdots & \vdots \\ C_{p2}\, V_{1i} + C_{p1}\, V_{2i} & -C_{p1}\, V_{1i} + C_{p2}\, V_{2i} \end{bmatrix}$$

$$W_k = \begin{bmatrix} -b_{21}\, V_{3k} + b_{11}\, V_{4k}, & -b_{22}\, V_{3k} + b_{12}\, V_{4k}, & \dots, & -b_{2m}\, V_{3k} + b_{1m}\, V_{4k} \\ b_{11}\, V_{3k} + b_{21}\, V_{4k}, & b_{12}\, V_{3k} + b_{22}\, V_{4k}, & \dots, & b_{1m}\, V_{3k} + b_{2m}\, V_{4k} \end{bmatrix}$$

with $\quad i = 0, 1, \dots, r, \quad k = 0, 1, \dots, s$

V_{1i}, V_{2i}, V_{3k} and V_{4k} are shown in Eqs. (18) and (20). The corresponding orthogonal matrices, $V^{\#}$ and $W^{\#}$ are

$$V^{\#} = [\dots,\ V_i^{\#},\ \dots], \quad W^{\#} = \begin{bmatrix} \vdots \\ W_k^{\#} \\ \vdots \end{bmatrix} \tag{37}$$

where

$$V_i^{\#} = \begin{bmatrix} (c_{12}\, V_{ai}^{\#} + c_{11}\, V_{bi}^{\#})/(pc_{11}^2 + pc_{12}^2), \dots, (c_{p2}\, V_{ai}^{\#} + c_{p1}\, V_{bi}^{\#})/(pc_{p1}^2 + pc_{p2}^2) \\ (-c_{11} V_{ai}^{\#} + c_{12}\, V_{bi}^{\#})/(pc_{11}^2 + pc_{12}^2), \dots, (-c_{p1} V_{ai}^{\#} + c_{p2}\, V_{bi}^{\#})/(pc_{p1}^2 + pc_{p2}^2) \end{bmatrix}$$

$$W_k^{\#} = \begin{bmatrix} (-b_{21}\, V_{ck}^{\#} + b_{11}\, V_{dk}^{\#})/(mb_{21}^2 + mb_{11}^2) & (b_{11}\, V_{ck}^{\#} + b_{21}\, V_{dk}^{\#})/(mb_{21}^2 + mb_{11}^2) \\ (-b_{22}\, V_{ck}^{\#} + b_{12}\, V_{dk}^{\#})/(mb_{22}^2 + mb_{12}^2) & (b_{12}\, V_{ck}^{\#} + b_{22}\, V_{dk}^{\#})/(mb_{22}^2 + mb_{12}^2) \\ \vdots & \vdots \\ (-b_{2m}\, V_{ck}^{\#} + b_{1m}\, V_{dk}^{\#})/(mb_{2m}^2 + mb_{1m}^2) & (b_{1m}\, V_{ck}^{\#} + b_{2m}\, V_{dk}^{\#})/(mb_{2m}^2 + mb_{1m}^2) \end{bmatrix}$$

$V^{\#}_{ai}$, $V^{\#}_{bi}$, $V^{\#}_{ck}$ and $V^{\#}_{dk}$ are shown in Eqs. (26) and (28). Note that $V^{\#}$ and $W^{\#}$ are matrices with dimensions 2 x(r+1) p and (s+1)mx2 respectively for the MIMO case.

In order to update the projection filters to identify the modal parameters from measured data within a specific range of accuracy, a cost function is formed as follows:

$$J = \frac{1}{2} U^T U \tag{38}$$

where

$$U^T = [E_{11}, E_{12}, E_{21}, E_{22}]$$

and

$$\begin{bmatrix} E_{11} & E_{12} \\ E_{12} & E_{22} \end{bmatrix} = V^{\#} H(0) W^{\#} - I_2$$

From Eq. (8), the cost function J would go to a minimum value (ideally, zero) when the projection filters have the same modal characteristics as the measured data have. On the other hand, for the specified region of system parameters of interest, we may update those system parameters of the projection filters within the specified region so that the cost function is globally minimized. This global minimum of the cost function in the specified region will provide the "best" estimates for the modal parameters which represent the characteristics of the measured data. Although the cost function may be corrupted by system or measurement noise, the system parameters corresponding to the global minimum are expected to be quite insensitive to noise that is not correlated with the parameters being estimated.

Chapter 3

GLOBAL OPTIMIZATION USING INTERVAL ANALYSIS

3.1 One-Dimensional Global Optimization

The method for computing a global minimum within a specified region of system parameters is based on the algorithm developed by Hansen[5-8]. This algorithm can deal with problems in the multi-variable case with inequality constraints[6]. In this section, only a single variable[5] (either modal frequency or modal damping) is considered. Two-Dimensional case will be given in the next section. The global optimization algorithm basically uses a Newton method[8] in conjunction with the interval analysis to solve a system of nonlinear equations. The term "global minimum" used herein refers to the smallest value of the cost function J throughout a closed interval of a system parameter. Because of the interval analysis, the computational procedure of this algorithm requires explicit expressions for the first derivative (J') and the second derivative (J'') of the cost function J, Eq. (38). This can be derived easily by using the explicit expressions for the modal filters shown in Eqs. (23) and (24). The algorithm developed by Hansen[5] has been slightly modified. Instead of using three lists (L_0, L_1 and L_2), only two lists are applied. The algorithm is summarized as follows:

Initial Step: The algorithm starts with an initial interval X_0. This interval is equally subdivided into subintervals which are stored in a list L_0. A list L_1 (initially empty) consists of intervals for which the width is smaller than a specified value w_1 and the corresponding width of J is smaller than a specified value w_2. Let $\bar{x}$ denote a feasible approximation to the global minimum. If the feasible point is not given, the upper limit of the cost function is set to $\bar{J} = \infty$ with $\bar{x}$ indefinite. Let $[j_L, j_R]$, $[j_L', j_R']$ and $[j_L'', j_R'']$ denote the interval resulting from evaluating J, J' and J'' in interval arithmetic using the argument X, respectively. That is,

$$J(X) = [j_L \; j_R], \quad J'(X) = [j_L', j_R'], \quad J''(X) = [j_L'', j_R''] \qquad (39)$$

and
$$X = [x_L, x_R]$$

Then, use the interval analysis to find the corresponding J, J' and J'' for all the subintervals in L_0.

Main Steps:

1. If the list L_0 is empty, go to step 11. Otherwise, find the subinterval X in L_0 for which the left endpoint of J(X), i.e. j_L, is smallest.

2. If $\bar{x} \in X$, set $x = \bar{x}$. Otherwise, set $x = m(X)$ = midpoint of X. If $j_L > \bar{J}$, the cost of any point inside the intervals in L_0 exceeds the upper limit $\bar{J}$. Then L_0 is set empty and go to step 11.

3. Concavity Check

   If $j_R'' < 0$, J is concave in X and cannot have a minimum in the interior of X. Then, X is deleted and go to step 1.

4. Monotonicity Check

   If $j_R' < 0$ or $j_L' > 0$, the gradient of J is strictly positive or strictly negative over X. Then, X is deleted and go to step 1.

5. Gaussian Elimination

   Denote $E = \bar{J} - J(x)$, $\Delta = [J'(x)]^2 + 2Ej_L''$

   If $j_L'' > 0$ and $\Delta > 0$, it implies that $J(y) > \bar{J}$ for any $y \in X$. Then, X is deleted and go to step 1. Note that this is true only for $j_L'' > 0$, which is not indicated in Ref. 5.

6. Interval Newton Method[5]

If $j_L'' > 0$, denote $S' = x - J'(x)/J''(X)$ and S = intersection of S' and X. Otherwise, denote $S = S_1 \cup S_2$

Here S_1 and S_2 are defined as follows:

Denote $c = x - J'(x)/j_L''$ for $j_L'' \neq 0$.

and $d = x - J'(x)/j_R''$ for $j_R'' \neq 0$.

$$\text{If } J'(x) \geq 0 \quad S_1 = \begin{cases} [x_L,\ d] & \text{when } j_R''>0 \text{ and } d \geq x_L \\ \text{empty} & \text{when } j_R''=0 \text{ or } d<x_L \end{cases}$$

$$S_2 = \begin{cases} [c,\ x_R] & \text{when } j_L''<0 \text{ and } c \leq x_R \\ \text{empty} & \text{when } j_L''=0 \text{ or } c>x_R \end{cases}$$

$$\text{If } J'(x)<0 \quad S_1 = \begin{cases} [x_L,\ c] & \text{when } j_L''<0 \text{ and } c \geq x_L \\ \text{empty} & \text{when } j_L''=0 \text{ or } c<x_L \end{cases}$$

$$S_2 = \begin{cases} [d,\ x_R] & \text{when } j_R''>0 \text{ and } d \leq x_R \\ \text{empty} & \text{when } j_R''=0 \text{ or } d>x_R \end{cases}$$

7. If S is empty, go to step 1. If $S=X$, then split X in half.

8. For each new generated subinterval, $\hat{X} = S_1$, S_2, or S, repeat steps 3 and 4.

9. Update $\bar{J}$

For each new subinterval $\hat{X}$, denote

$w[\hat{X}]$ = width of $\hat{X}$, $\hat{x}$ = mid $[\hat{X}]$ = midpoint of $\hat{X}$

$\hat{X} = [\hat{x}_L,\ \hat{x}_R]$ and $J(\hat{X}) = [\hat{j}_L,\ \hat{j}_R]$

If $J(\hat{x})<\bar{J}$, simply replace $\bar{J}$ by $J(\hat{x})$ or conduct a line search to reduce $\bar{J}$ as follows:

a. If $J'(\hat{x})>0$, denote $\hat{x}_1 = \hat{x}_L$. Otherwise, denote $\hat{x}_1 = \hat{x}_R$. Set $\hat{x}_0 = \hat{x}$.

b. Denote $\hat{x}_2 = (\hat{x}_0 + \hat{x}_1)/2$. If $J(\hat{x}_2) \geq \max\ [J(\hat{x}_0), J(\hat{x}_1)]$, go to step e.

c. If $J(\hat{x}_0) \leq J(\hat{x}_1)$, replace $\hat{x}_1$ by $\hat{x}_2$. Otherwise replace $\hat{x}_0$ by $\hat{x}_2$.

d. If $|\hat{x}_1 - \hat{x}_0| > \frac{1}{16} w[X]$, go to step b.

e. Set $\bar{J} = \min[J(\hat{x}), J(\hat{x}_0), J(\hat{x}_1)]$ and set $\bar{x}$ to the corresponding arugument of $\bar{J}$.

10. Store new intervals

For each new interval $\hat{X}$, if $\hat{j}_L > \bar{J}$, delete $\hat{X}$.

If $w[\hat{X}] \leq w_1$ and $\hat{j}_R - \hat{j}_L \leq w_2$, store $\hat{X}$ in L_1. Otherwise, store $\hat{X}$ in L_0. Go to step 1.

11. If the list L_1 is empty, go to step 13. Otherwise, delete subintervals X for which $j_L > \bar{J}$ where $J(X) = [j_L, j_R]$.

12. If the list L_1 is empty, go to step 13. Otherwise, the midpoints of each interval remaining in L_1 are used as the global minima. Note that there may exist multiple global minima.

13. Because L_1 is empty, the global minimum is located on one of the two boundaries of the initial interval X_0, which corresponds to a smaller J.

3.2 Two-Dimensional Global Optimization

3.2.1 The Statement of the Problem

Given a scalar objective or cost function $f(x)$ of two independent variables, find a point x^* within a specified rectangular region $X^{(0)}$, which is called "box" in the sequel, $X^{(0)}$ R^2, such that

$$f(x^*) \leqslant f(x) \qquad (40)$$

for any $x \in X^{(0)}$.

There is no constraint except that x^* must be within a rectangular region with sides parallel to the coordinate axes. Here, we assume the objective function is twice continuously differentiable.

The algorithm presented here is composed of four main separate parts. Let X be a sub-box of $X^{(0)}$ and is the box under treatment. One part uses an interval version of Newton's method to find a sub-box or some sub-boxes of X such that any stationary point within X will still be contained in these sub-boxes. By this method, we can shrink the box under treatment to a smaller box or several smaller boxes without losing any possible extreme point. A second part eliminates points of X where f is greater than the smallest currently known value. This is because only the global minimum is of interest. Any point which is obviously not the minimum could be deleted. A third part tests the monotonicity of the box. If f in the box X increases or decreases montonically, this box can be eliminated, except the box contains the boundary points of the initial box $X^{(0)}$. A fourth part checks the convexity of the sub-box X. If f is not convex anywhere in X, there cannot be a stationary minimum of f in X. Therefore, the whole box can be deleted. Note that if the global minimum point x^* occurs on the boundary of $X^{(0)}$, it is not necessary to be a stationary point. So when we use the fourth part of the algorithm, we must retain all the boundary points.

The following several sections will describe these four parts in more detail.

3.2.2 The Interval Newton's Method

This method seeks the zeros of the gradient function g of the objective function f and hence the stationary points of f. In ordinary functions, by using Newton's method, we can locate the stationary points of f. In interval analysis, by using the interval version of Newton's method, we can find a new box or several boxes N(X), which are smaller than the previous box X, such that any point in X not in N(X) cannot be a zero of g and can be discarded, unless it is a boundary point of $X^{(0)}$. The method is described as follows.

f(x) is a scalar objective function, x is a column vector, and g(x) is the gradient function of f. Note that g(x) is also a column vector. We want to solve the equation

$$g(x) = 0 \tag{41}$$

Let y be an approximate solution to Eq. (41) and J(x) denote the Jacobian matrix. Then using Taylor's theorem to expand g(x) about y, we can get the following equation

$$g(x) = g(y) + J(\tilde{x})\ (x-y) = 0. \tag{42}$$

where $\tilde{x}$ is a point between x and y. If x, y ε X then $\tilde{x}$ ε X. We will obtain a box or several boxes as its solution if substitute real point $\tilde{x}$ by a box X and solve Eq. (42).

To solve Eq. (42) with $\tilde{x}$ replaced by X, first we find a real matrix J_c which is the center of the interval Jacobian matrix J(X). That is, each element of J_c is the center of the correspondent interval element of the interval Jacobian matrix. Let B be an inverse of J_c, then on premultiplication of Eq. (42) by B, we get

$$Bg(y) + BJ(X)\ (x - y) = 0 \tag{43}$$

Any point in X solving Eq. (41) should be contained in the interval solution of Eq. (43), which could be solved by the following method. Write

$$BJ(X) = L + D + U \tag{44}$$

where L, D and U are the lower triangular, diagonal and upper triangular part of BJ(X) respectively. The interval matrix

$$D^{-1} = \text{diag}\ [1/D_{11},\ 1/D_{22}] \tag{45}$$

is the inverse interval matrix of D. Then the solution of Eq. (43) is

$$X' = y - D^{-1}\ [Bg(y) + L(X' - y) + U(X - y)] \tag{46}$$

Though X' also appears in the right hand side of Eq. (46), its elements can be solved recursively, because it is multiplied with a lower triangular matrix L. If $0 \notin D_{11}$ and $0 \notin D_{22}$, X' is a single box, otherwise X' could be several boxes. After intersecting with the previous box X, the answer could be the empty set, one box which is the same as X or a reduced box or several reduced boxes.

3.2.3 Bounding $\bar{f}$

As we proceed with the algorithm, we should evaluate f(x) at a number of points of $X^{(0)}$. Let $\bar{f}$ denote the currently smallest value of f found so far, and let X be any sub-box of $X^{(0)}$. We can delete any point x of X where $f(x) > \bar{f}$. The method we used is described as follows.

Expanding f about a point $x \varepsilon X$, one obtains

$$f(y) = f(x) + (y - x)^T\ g(z) \tag{47}$$

where z is a point between x and y, g is the gradient function of f. If $x \varepsilon X$, $y \varepsilon X$, then $z \varepsilon X$. Now we substitute z by the box X. If

$$f(x) + (y - x)^T g(X) > \bar{f} \tag{48}$$

then $f(y) > \bar{f}$. We want to retain points where Eq. (48) is not satisfied. This is the same as to say we want to retain those points where the following inequality is satisfied.

$$f(x) + (y - x)^T g(X) \leqslant \bar{f} \tag{49}$$

Let

$$\bar{f} - f(x) = e$$

then

$$(y - x)^T g(X) - e \leqslant 0$$

i.e.

$$[(y_1 - x_1), (y_2 - x_2)] \begin{bmatrix} g_1(X) \\ g_2(X) \end{bmatrix} - e \leqslant 0$$

Let

$$y_1 - x_1 = \bar{y}_1$$

$$y_2 - x_2 = \bar{y}_2$$

then

$$\bar{y}_1 \, g_1(X) + \bar{y}_2 \, g_2(X) - e \leqslant 0 \tag{50}$$

If we expand the function about the center of X, and first use X_2 of $X(X_1, X_2)$ as $\bar{y}_2$, then Eq. (50) has the general form of a linear equation with interval coefficient

$$A + Bt \leqslant 0 \tag{51}$$

Here, $\bar{y}_1 = t$, $A = X_2\ g_2(X) - e$, $B = g_1\ (X)$. Let $A = [a_1, a_2]$, $B = [b_1, b_2]$, $T = \{t\text{: } at + b \leqslant 0,\ a \varepsilon A,\ b \varepsilon B\}$, then the general solution T to Eq. (51) is as follows

$$T = \begin{cases} [-a_1/b_2,\ \infty] & \text{if } a_1 \leqslant 0,\ b_2 < 0 \\ [-a_1/b_1,\ \infty] & \text{if } a_1 > 0,\ b_1 < 0,\ b_2 \leqslant 0 \\ [-\ \infty,\ \infty\ ] & \text{if } a_1 \leqslant 0,\ b_1 \leqslant 0 \leqslant b_2 \\ [-\infty,\ -a_1/b_2] \cup [-a_1/b_1,\ \infty] & \text{if } a_1 > 0,\ b_1 < 0 < b_2 \\ [-\infty,\ -a_1/b_1] & \text{if } a_1 \leqslant 0,\ b_1 > 0 \\ [-\infty,\ -a_1/b_2] & \text{if } a_1 > 0,\ b_1 \geqslant 0,\ b_2 > 0 \\ \text{empty set} & \text{if } a_1 > 0,\ b_1 = b_2 = 0 \end{cases} \qquad (52)$$

Note the solution of Eq. (50) may be empty set, one interval or two intervals. Though the interval may be semi-infinite, after intersecting with the previous box X, the result is finite. After obtaining Y_1, we treat Y_2 as a variable and solve it similarly.

3.2.4 Check for Monotonicity

Let X be any interior sub-box of $X^{(0)}$, which means X contains no boundary points of $X^{(0)}$. Note if any optimal point should occur in the interior it must be a stationary point. Therefore, if we find f increases or decreases monotonically throughout X with respect to each variable x_i, we can delete the whole box X, since X does not contain any stationary point. To make use of this characteristic, we first evaluate the interval gradient function g of f and check each component. Denote $[d_j, w_j]$ for the interval value of the j-th component $g_j(X)$ of g, $(j = 1, 2)$. It is obvious that $d_j < g_j(x) < w_j$ for any $x \varepsilon X$. Denote the i-th component of X by $X_i = [x_i^L, x_i^R]$, if $d_j = 0$ and $w_j > 0$, in the x_j direction $f(x)$ is smallest for $x_j = x_j^L$ (in X). So all of X except the boundary line of which x_j^L is its component can be deleted. If $d_j > 0$ we can delete all of X, except if x_j is a boundary point of $X^{(0)}$. In this case, we save this boundary point. Similar results can be obtained if $w_i = 0$ and $d_i < 0$ or if $w_i < 0$. We check $g_i(X)$ $(i = 1, 2)$ one by one, delete points as many as possible and save those that cannot be deleted. By this method, we may delete the whole box, reduce the box to a degenerated box, a line segment, or reduce the box to a point. In the latter case, we evaluate f at this point. If $f > \bar{f}$, this point can be deleted. If $f < \bar{f}$, we replace $\bar{f}$ by f and store it.

3.2.5 Test for Convexity

If f is not convex in a given box, none of its stationary points can be a minimum. Because we are only interested in the global minimum, if f in a subbox X of $X^{(0)}$ is not convex, we just delete it before doing any other treatment. But any boundary point of the initial box $X^{(0)}$ should be retained, for if the minimum occurs in the boundary it is not necessarily a stationary point.

To check the convexity of a function, we need to check its second derivative. For a scalar function f of two variables, the second derivative is a 2 x 2 Hessian matrix

$$H = \begin{bmatrix} \dfrac{\partial^2 f}{\partial X_1^2} & \dfrac{\partial^2 f}{\partial X_1 \, \partial X_2} \\ \dfrac{\partial^2 f}{\partial X_2 \, \partial X_1} & \dfrac{\partial^2 f}{\partial X_2^2} \end{bmatrix} \qquad (53)$$

For a real function, the stationary point is a minimum if the Hessian matrix evaluated at this point is positive definite. For interval function, it is not easy to check if the Hessian matrix is positive definite. But we can use a quick way to rule out those points which are definitely not qualified. This can be done by checking the diagonal elements of the interval Hessian matrix. For a positive definite square matrix, each diagonal element must be greater than zero. Taking advantage of this fact, we first evaluate the interval value of the diagonal elements. Denote the i-th diagonal element by $d_{ii}(x)$ and its corresponding interval value of the i-th diagonal element by $[D^L_{ii}, D^R_{ii}](i = 1, 2)$, if $D^R_{ii} < 0$ then $d_{ii}(x) < 0$ for any $x \varepsilon X$, the Hessian matrix can not be positive definite and we can eliminate X, except those boundary points of $X^{(0)}$.

3.2.6 Boundary Points

If the global minimum is a stationary point, then we are free to delete all the boundary points. Otherwise, we should be careful not to delete boundary points which might be the global minima within the specified region. The interval Newton method and the procedure which checks the convexity cannot delete any boundary point of $X^{(0)}$. In these two procedures, if we find that the treated sub-box contains boundary points, we just bypass the procedures. However, the procedures of checking montonicity and bounding $\bar{f}$ as described above can eliminate some boundary points. When checking monotonicity, if we find $g_i\ (X_1, X_2) > 0\ (i = 1, 2)$ then we can delete all points of X except if the points at the left boundary line of X with respect to x_i are also the boundary points of $X^{(0)}$. In this case, we save this boundary line and delete all the other points including some boundary points of $X^{(0)}$ which might exist. If $g_i\ (X_1, X_2) < 0$, similar treatment can be applied. When bounding $\bar{f}$, we can delete any point (including boundary point) x if $f(x) > \bar{f}$.

We can also separate the boundary points from the interior points of $X^{(0)}$ in the beginning of the process, and treat the interior as a box wherein the global minimum must be a stationary point. Those separated boundary points are processed later. This procedure is more straightforward but will be impractical when the dimension of $X^{(0)}$ is getting larger because it will generate too many boxes.

3.2.7 The Lists of Boxes

Our algorithm begins with an initial box $X^{(0)}$. By processing $X^{(0)}$ through the above procedures for one run, we will either get an unchanged box, one smaller box or several sub-boxes. For the unchanged box, we equally divide it by two and save them in a list denoted by L_1. For the smaller box, we check how much it was reduced. A criterion was set to make a judgement. If it is only slightly reduced, we divide it by two along its longest dimension and save them in L_1. If it has been reduced a lot, then just save it in L_1. For the case of several sub-boxes, we save them all in L_1 for further treatment. At the beginning, the number of boxes saved in L_1 will increase rapidly. But when the size of the box becomes smaller, we will have more chances to delete the whole box through the procedures described above. Then the number of boxes in L_1 will begin to decrease.

Before saving any box in L_1 at the end of each run, we will check whether the size of the box and the range of the objective function corresponding to the box are smaller than the specified accuracies of the solution (e_1 and e_2 described in the next section). If it does, we save it in another list denoted by L_2. Eventually, the number of boxes in L_1 will become zero and only one or a few boxes will be left in L_2.

3.2.8 Termination Criterions

Two numbers, e_1 and e_2, are used to specify the accuracies of the answer we want. By which we mean the width of each dimension of the solution box must be smaller than e_1 and the interval of its objective function must be smaller than e_2. If we denote the width of the i-th element by $w(X_i)$ and the interval of the objective function by $[F^L, F^R]$, then the requirement can be expressed by

$$\max_{i=1,2} [w(X_i)] < e_1 \tag{54}$$

$$F^R - F^L < e_2 \tag{55}$$

After L_1 becomes empty, if there are a few boxes, say s, remaining in L_2, we have to continue to work on them to yield the best answer. Denote the boxes by $X^{(i)}$ $(i = 1, 2, ..., s)$. For each $1 = 1, 2, ..., s$, evaluate $f(X^{(i)})$ and denote the result by $[F_i^L, F_i^R]$. Then for every $x \varepsilon X^{(i)}$, $F_i^L \leqslant f(x) \leqslant F_i^R$. Now denote

$$\underline{f} = \min_{1 \leqslant i \leqslant s} F_i^L \tag{56}$$

then

$$\underline{f} \leqslant f(x) \tag{57}$$

for x in any of the boxes $X^{(i)}$ $(i = 1, 2, ..., s)$. Since any global minimum x^* must be one point within a certain box among $X^{(i)}$, we can infer that $\underline{f} \leqslant f(x^*)$. From the preceding section, $\bar{f}$ is an upper bound of $f(x^*)$, i.e. $\bar{f} \geqslant f(x^*)$ for anytime, hence

$$\underline{f} \leqslant f^* \leqslant \bar{f} \tag{58}$$

If $e = \bar{f} - \underline{f}$ is smaller than the desired value e_2, we terminate the algorithm and list out all the remaining boxes in L_2 as our answers. If e is bigger than e_2, then we find the value j for which $\underline{f} = F_j^L$, remove this box from L_2 to L_1 and re-run the main algorithm. This procedure can always increase $\underline{f}$ and hence decrease e. We repeat this process until e is smaller than e_3. Some boxes in L_2 could be deleted by this procedure. Those remaining in L_2 are the answers. Multiple answers are possible.

3.2.9 Steps of the Algorithm

We now describe the steps involved in the algorithm. First, the box $X^{(0)}$ is divided into several sub-boxes to reduce its size because if a box is too big, it will usually remain unchanged after passing through all the steps. Then we evaluate the value of the objective function at the center of each box and choose the smallest one as the initial $\bar{f}$. The value of $\bar{f}$ will keep on being reduced during the process. The subsequent steps are to be done in the following order except as indicated by branching:

1. Choose the first one box remaining in L_1, call it X. If list L_1 is empty, go to step 8.

2. Check for montonicity. Evaluate the gradient interval function g(X). If $g_i(X) > 0$ (or < 0) and the boundary of X at $x_i = x_i^L$ (or $= x_i^R$) does not contain any boundary point of $X^{(0)}$, delete X and go back to step 1. If $g_i(X) \geqslant 0$ (or $\leqslant 0$), replace $X_i = [x_i^L, x_i^R]$ by $[x_i^L, x_i^L]$ (or $[x_i^R, x_i^R]$).

3. Test for non-convexity. If X contains boundary points of $X^{(0)}$, go to step 4. Otherwise evaluate h_{ii} (X_1, X_2), (i = 1, 2). If either h_{11} or h_{22} is strictly negative, delete X and go back to step 1.

4. Bounding $\bar{f}$. Evaluate $g(X)$ according to Eqs. (50) and (51) and solve for interval Y. Intersect Y with X. If the result is empty, go to step 1. If the result is a single box, rename it as X. If it ends up with two boxes, save one in L_1, rename the other one as X and go on.

5. Interval Newton method. Evaluate the interval Jacobian matrix and solve for X' according to Eqs. (43) to (46). If the denominator interval contains zero, the element X_i' in X' becomes two boxes. Then save the gap between the two boxes and denote it by g_i. Find the smallest box which contains these two boxes and rename it as X_i'. Finally, intersect X' with X and rename it as X. If the result is empty, go back to step 1.

6. Check the gap saved. Choose the largest one, say g_j, and divide the box X along g_j. If there is no gap saved, check the size of the current box X. If it does not satisfy the accuracy requirements e_1 and e_2, divide it into two boxes along its greatest dimension and go on.

7. Evaluate f at the center of each of the boxes before storing them. Update $\bar{f}$ by choosing the smallest one from $\bar{f}$ and $f(x)$ at the center of each box as the new $\bar{f}$. If the box satisfies the accuracy requirements, store it in L_2. Then store all the remaining boxes in L_1 and go back to step 1.

8. Evaluate $f(X^{(i)})$ for each box $X^{(i)}$ remaining in L_2. Denote the result by $[F_i^L, F_i^R]$ and find

$$\underline{f} = \min_{1 \leqslant i \leqslant s} F_i^L \qquad (59)$$

If $e = \bar{f} - \underline{f} > e_3$, put the box X_j which has $\underline{f}$ as its left end-point back to L_1 and go back to step 1. If $e < e_3$, stop the process and print out all the boxes remaining in L_2 as the answers.

The algorithm described above can be expressed roughly by a chart shown in Fig. 3.1.

3.3 Numerical Examples

In this section, we use two examples to illustrate the application of the two-dimensional global optimization algorithm. In case one, we consider the so-called three hump camel function

$$f(x) = 2\,x_1^2 - 1.05\,x_1^4 + \frac{1}{6}\,x_1^6 - x_1\,x_2 + x_2^2 \qquad (60)$$

which has three relative minima and two saddle points. Among them, there is only one global minimum. The gradient g(x) has two components

$$g_1(x) = 4\,x_1 - 4.2\,x_1^3 + x_1^5 - x_2 \qquad (61)$$

$$g_2(x) = 2\,x_2 - x_1 \qquad (62)$$

The Hessian matrix of f has four components

$$h_{11}(x) = 4 - 12.6\,x_1^6 + 5\,x_1^4 \qquad (63)$$

$$h_{12}(x) = -1 = h_{21}(x) \qquad (64)$$

$$h_{22}(x) = 2 \qquad (65)$$

The interval extension of the gradient function and the Hessian matrix can be derived by substituting the interval variables X_1 and X_2 for the real variable x_1 and x_2. The subroutines for evaluating real and interval value of the functions are written. Running the two-dimensional global optimization program by using the initial intervals X_1 = [-100, 100], X_2 = [-100, 100], the global minimum is shown in Table 3.1

It takes 2.8 cpu sec to obtain the answer in CDC Cyber 830 computer.

In case two, we consider a function called Himmelblau function

$$F(x_1, x_2) = - (x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2 \qquad (66)$$

The isometric representation and the corresponding plot of its contours or level curves are shown in Fig. 3.3 and Fig. 3.4 respectively. In Fig. 3.3, the flat base represents a function value of -150. This function has four peaks and several saddle points. The global optimization problem is to find out its global extremes. Though the program is written for searching global minimum and the function here has maxima to be located, the problem can be easily solved by setting $f(x_1, x_2) = -F(x_1, x_2)$ and find global minimum of $f(x_1, x_2)$. Now

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \qquad (67)$$

The gradient $g(x)$ has two components

$$g_1(x) = 4 (x_1^2 + x_2 - 11) x_1 + 2 (x_1 + x_2^2 - 7) \qquad (68)$$

$$g_2(x) = 2 (x_1^2 + x_2 - 11) + 4 (x_1 + x_2^2 - 7) x_2 \qquad (69)$$

The Hessian matrix has four components

$$h_{11}(x) = 12 x_1^2 + 4 x_2 - 42 \qquad (70)$$

$$h_{12}(x) = 4 x_1 + 4 x_2 = h_{21}(x) \qquad (71)$$

$$h_{22}(x) = 4 x_1 + 12 x_2^2 - 26 \qquad (72)$$

We write the subroutines for calculating the value of real function and the corresponding interval functions and then run the program with an initial box of X_i = [-100, 100], X_2 = [-100, 100]. The results are listed in Table 3.2.

This result is exactly the same as the result found in Ref. [12]. It takes 2.82 CPU sec to get the answers in CDC Cyber 830 computer.

Chapter 4

MODAL PARAMETER IDENTIFICATION USING PROJECTION FILTERS

In this chapter, the global optimization algorithm using interval analysis will be applied to the single-mode projection filters for modal parameter identification. A numerical simulation of a ten-mode structure is first presented to demonstrate the feasibility of the projection filters as well as the two-dimensional global optimization algorithm. Then another numerical simulation of the MAST beam structure is shown by using a one-dimensional global optimization algorithm. The results show that the projection filters are feasible for modal parameter identification.

4.1 Numerical Simulation for a 10-Mode Structure

As described in Chap. 2, a finite-dimensional, linear, time-invariant, continuous time dynamic system can be expressed by the state-variable equations

$$x(t) = Ax(t) + Bu(t) \tag{73}$$

$$y(t) = Cx(t) \tag{74}$$

After appropriate linear transformation, matrix A could be expressed as follows

$$A = \begin{bmatrix} -\sigma_1 & \omega_1 & & & & & \\ -\omega_1 & \sigma_2 & & & & & \\ & & -\sigma_2 & \omega_2 & & 0 & \\ & & -\omega_2 & -\sigma_2 & & & \\ & & & & \ddots & & \\ & 0 & & & & -\sigma_m & \omega_m \\ & & & & & -\omega_m & -\sigma_m \end{bmatrix} \tag{75}$$

which is a block diagonal matrix. Here, ω_i and σ_i are respectively the damped modal frequency and the modal damping of the i-th mode of the system. For an m-mode system, the dimension of A is 2mx2m. For the discrete-time system, matrix A becomes

$$A = \begin{bmatrix} A_1 & & & 0 \\ & A_2 & & \\ & & \ddots & \\ 0 & & & A_m \end{bmatrix}$$

$$A_i = \begin{bmatrix} e^{-\sigma_i T} \cos(\omega_i T) \ , & e^{-\sigma_i T} \sin(\omega_i T) \\ - e^{-\sigma_i T} \sin(\omega_i T) \ , & e^{-\sigma_i T} \cos(\omega_i T) \end{bmatrix} \qquad (76)$$

$$i = 1, 2, \ldots m$$

where T is the sampling time interval. Note that each 2x2 block in the main diagonal corresponds to one of the system modes. Next we want to construct the impulse response function y according to Eq. (3). Note that

$$A^k = \begin{bmatrix} {A_1}^k & & & 0 \\ & A_2^k & & \\ & & \ddots & \\ 0 & & & A_m^k \end{bmatrix} \qquad (77)$$

and note that

$$A_i^k = \begin{bmatrix} e^{-k\sigma_i T} \cos(k\omega_i T) \ , & e^{-k\sigma_i T} \sin(k\omega_i T) \\ -e^{-k\sigma_i T} \sin(k\omega_i T) \ , & e^{-k\sigma_i T} \cos(k\omega_i T) \end{bmatrix} \qquad (78)$$

If the system is a single-input single-output system and we denote matrix B and C as

$$B = [b_1, b_2, \ldots, b_m]^T$$

$$b_i = [b_{i1}, b_{i2}]^T \qquad (79)$$

$$C = [c_1, c_2, \ldots, c_m]$$

$$c_i = [c_{i1}, c_{i2}] \qquad (80)$$

$$i = 1, 2, \ldots, m$$

then

$$\begin{aligned} y(k) &= C\,A^{k-1}B \\ &= [c_1, c_2, \ldots, c_m] \begin{bmatrix} A_1^{k-1} & & & 0 \\ & A_2^{k-1} & & \\ & & \ddots & \\ 0 & & & A_m^{k-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \\ &= \sum_{i=1}^{m} c_i\, A_i^{k-1}\, b_i \qquad (81) \\ &= \sum_{i=1}^{m} [e^{-(k-1)\,\sigma_i T} (c_{i1} \cos((k-1)\omega_i\, T) - c_{i2} \sin((k-1)\,\omega_i T))\, b_{i1} \\ &\quad + e^{-(k-1)\,\sigma_i T} (c_{i1} \sin((k-1)\,\omega_i T) - c_{i2} \cos((k-1)\,\omega_i T))\, b_{i2}] \end{aligned}$$

By assigning modal frequency and modal damping to each mode, the free decay impulse response function y can be constructed from Eq. (81). A series white noise signals are generated from a computer program and added to y to form a sequence of simulated output data. Using these output data, we can form the generalized Hankel matrix from Eq. (4). The explicit expressions of the single-mode filters are described by Eqs. (23) and (24).

Assume we have a ten-mode system with the following separate frequencies:

3, 7, 11, 17, 23, 31, 37, 43, 53, 67. (rad/sec)

The damping ratios are assumed to be the same for each mode. For a large flexible structure system, the damping coefficient is always very small. To simplify the problem, matrices B and C are assumed to be as follows

$$B = [1,\ 0,\ 1,\ 0,\ \ldots\ldots,\ 1,\ 0]^T_{1\times 20} \tag{82}$$

$$C = [0,\ 1,\ 0,\ 1,\ \ldots\ldots,\ 0,\ 1]_{1\times 20} \tag{83}$$

$$r = s,\ j_i = t_i = i,\ i = 1,\ 2,\ \ldots\ldots,\ r$$

Then the projection filters, Eqs. (23) and (24) become

$$V^{\#} = \begin{bmatrix} V_a^{\#}(r) \\ V_b^{\#}(r) \end{bmatrix} \tag{84}$$

$$W^{\#} = [V_c^{\#}(s)^T,\ V_d^{\#}(s)^T] \tag{85}$$

$$\lambda_1(r) = \lambda_3(s),\ \lambda_2(r) = \lambda_4(s)$$

$$V_c^{\#}(s) = V_a^{\#}(r),\ V_d^{\#}(s) = V_b^{\#}(r)$$

Now the cost function can be expressed as follows.

$$J(\sigma,\ \omega) = E_{11}^2 + 0.5 * (E_{12}^2 + E_{21}^2) \tag{86}$$

where

$$E_{11} = V_a^{\#}\ H(0)\ V_b^{\#^T} - 1$$

$$E_{12} = V_a^{\#}\ H(0)\ V_a^{\#^T}$$

$$E_{21} = V_b^{\#} H(0) V_b^{\#^T}$$

$$E_{22} = V_b^{\#} H(0) V_a^{\#^T} - 1$$

Note that E_{11}, E_{12}, E_{21}, E_{22} are all functions of ω and σ.

The first and second partial derivative of J can be derived as follows

$$\frac{\partial J}{\partial \omega} = 2 \frac{\partial E_{11}}{\partial \omega} E_{11} + \frac{\partial E_{12}}{\partial \omega} E_{12} + \frac{\partial E_{21}}{\partial \omega} E_{21} \tag{87}$$

$$\frac{\partial^2 J}{\partial \omega^2} = 2 \frac{\partial^2 E_{11}}{\partial \omega^2} E_{11} + 2 \left(\frac{\partial E_{11}}{\partial \omega}\right)^2 + \frac{\partial^2 E_{12}}{\partial \omega^2} E_{12} + \left(\frac{\partial E_{12}}{\partial \omega}\right)^2$$

$$+ \left(\frac{\partial^2 E_{21}}{\partial \omega^2}\right) E_{21} + \left(\frac{\partial E_{21}}{\partial \omega}\right)^2 \tag{88}$$

$$\frac{\partial J}{\partial \sigma} = 2 \frac{\partial E_{11}}{\partial \sigma} E_{11} + \frac{\partial E_{12}}{\partial \sigma} E_{12} + \frac{\partial E_{21}}{\partial \sigma} E_{21} \tag{89}$$

$$\frac{\partial^2 J}{\partial \sigma^2} = 2 \frac{\partial^2 E_{11}}{\partial \sigma^2} E_{11} + 2 \left(\frac{\partial E_{11}}{\partial \sigma}\right)^2 + \frac{\partial^2 E_{12}}{\partial \sigma^2} E_{12} + \left(\frac{\partial E_{12}}{\partial \sigma}\right)^2$$

$$+ \left(\frac{\partial^2 E_{21}}{\partial \omega^2}\right) E_{21} + \left(\frac{\partial E_{21}}{\partial \omega}\right)^2 \tag{90}$$

$$\frac{\partial^2 J}{\partial \omega \partial \sigma} = 2 \frac{\partial^2 E_{11}}{\partial \omega \partial \sigma} E_{11} + 2 \left(\frac{\partial E_{11}}{\partial \omega}\right) \left(\frac{\partial E_{11}}{\partial \sigma}\right) + \frac{\partial^2 E_{12}}{\partial \omega \partial \sigma} E_{12} + \left(\frac{\partial E_{12}}{\partial \omega}\right) \left(\frac{\partial E_{12}}{\partial \sigma}\right)$$

$$+ \frac{\partial^2 E_{21}}{\partial \omega \partial \sigma} E_{21} + \left(\frac{\partial E_{21}}{\partial \omega}\right) \left(\frac{\partial E_{21}}{\partial \sigma}\right) \tag{91}$$

with

$$\frac{\partial E_{11}}{\partial \omega} = V_a^{\#} H(0) \frac{\partial (V_b^{\#^T})}{\partial \omega} + V_b^{\#} H(0) \frac{\partial (V_a^{\#^T})}{\partial \omega} \tag{92}$$

$$\frac{\partial E_{12}}{\partial \omega} = 2\ V_a^{\#}\ H(0)\ \frac{\partial (V_a^{\#^T})}{\partial \omega} \tag{93}$$

$$\frac{\partial E_{21}}{\partial \omega} = 2\ V_b^{\#}\ H(0)\ \frac{\partial (V_b^{\#^T})}{\partial \omega} \tag{94}$$

$$\frac{\partial^2 E_{11}}{\partial \omega^2} = V_b^{\#}\ H(0)\ \frac{\partial^2 (V_a^{\#^T})}{\partial \omega^2} + \ V_a^{\#}\ H(0)\ \frac{\partial^2\ (V_b^{\#^T})}{\partial \omega^2} + 2\ \frac{\partial (V_a^{\#^T})}{\partial \omega}\ H(0)\ \frac{\partial (V_b^{\#^T})}{\partial \omega} \tag{95}$$

$$\frac{\partial^2 E_{12}}{\partial \omega^2} = 2\ \frac{\partial (V_a^{\#})}{\partial \omega}\ H(0)\ \frac{\partial (V_a^{\#^T})}{\partial \omega} + 2\ V_a^{\#}\ H(0)\ \frac{\partial^2 (V_a^{\#^T})}{\partial \omega^2} \tag{96}$$

$$\frac{\partial^2 E_{21}}{\partial \omega^2} = 2\ \frac{(V_b^{\#})}{\partial \omega}\ H(0)\ \frac{\partial (V_b^{\#})^T}{\partial \omega} + 2\ V_b^{\#}\ H(0)\ \frac{\partial^2 (V_b^{\#^T})}{\partial \omega^{\ 2}} \tag{97}$$

The first and second partial derivatives with respect to σ are all the same except substituting σ for ω. The cross partial derivative terms are

$$\frac{\partial^2 E_{11}}{\partial \omega \partial \sigma} = \frac{\partial V_a^{\#}}{\partial \sigma}\ H(0)\ \frac{\partial (V_b^{\#})^T}{\partial \omega} + V_a^{\#}\ H(0)\ \frac{\partial^2 (V_b^{\#})^T}{\partial \omega \partial \sigma}$$

$$+ \frac{\partial V_b^{\#}}{\partial \sigma}\ H(0)\ \frac{\partial^2 (V_a^{\#})^T}{\partial \omega \partial \sigma} + V_b^{\#}\ H(0)\ \frac{\partial^2 (V_a^{\#})^T}{\partial \omega \partial \sigma} \tag{98}$$

$$\frac{\partial^2 E_{12}}{\partial \omega \partial \sigma} = 2\ \frac{\partial V_a^{\#}}{\partial \sigma}\ H(0)\ \frac{\partial (V_a^{\#})^T}{\partial \omega} + 2\ V_a^{\#}\ H(0)\ \frac{\partial^2 (V_a^{\#})^T}{\partial \omega \partial \sigma} \tag{99}$$

$$\frac{\partial^2 E_{21}}{\partial \omega \partial \sigma} = 2\ \frac{\partial V_b^{\#}}{\partial \sigma}\ H(0)\ \frac{\partial (V_b^{\#})^T}{\partial \omega} + 2\ V_b^{\#}\ H(0)\ \frac{\partial^2 (V_b^{\#})^T}{\partial \omega \partial \sigma} \tag{100}$$

In numerical simulation, the following data are used

$$r = s = 49$$

$$m = 10$$

$$T = 0.04 \text{ sec}$$

which means the dimension of the Hankel matrix is (49+1) x (49+1) and there are ten modes in the system.

A FORTRAN program was written based on the above information. We specify

$$e_1 = 0.001$$

$$e_2 = 0.01$$

where e_1 and e_2 are the desired accuracies of the identified system parameters (either frequency or damping) and of the cost function respectively. The noise level is the ratio of the noise standard deviation with respect to the maximum value of y(k), i.e. the peak of the free impulse response. Specific modal frequencies with different damping coefficient and noise level are shown in Table 4.1. Corresponding to each modal frequency, ten frequency intervals are given for the projection filters:

$$[0.1, 5], [5, 10], [10, 15], [15, 20], [20, 25]$$

$$[25, 35], [34, 40], [40, 50], [50, 60], [60, 70]$$

in rad/sec

For each frequency interval, the range of the damping ratio is specified to be zero to five percent. The frequency interval along with the damping interval form a rectangular box. Within this box, the global minimum of the cost function generated by the projection filters is to be located.

The time history of the response y(k) with 30% noise is shown in Fig. 4.1. The white noise with 30% noise level is shown in Fig. 4.2. Figure 4.3 is a cross section view of the cost function vs. updated frequencies as damping equals to zero. Two values of measurement noise (0% and 30%) are shown to illustrate the effect of the noise. The result shows that the cost function is distorted by the noise, but the minima are not affected significantly. The percentage errors for the estimated modal frequency and damping are listed in Table 4.1 and Table 4.2 respectively. From Table 4.1, as the damping factor varies from 0.3% to 2%, the errors of estimated modal frequencies fall within 1% for the noise free case. As the noise level increases, the error increases proportionally and stay within 3% for 30% noise. For a fixed noise level, the errors increase for most of the modes as the damping factor increases. This is caused by the fact that the signal will damp out faster for higher damping factors. From Table 4.2, similar results are found for the modal damping errors except that the percentage errors are higher. For low frequency and low damping modes, the relative errors are higher because the contribution of the damping is comparably smaller.

4.2 Numerical Simulation for the MAST Beam Structure

For the multi-input multi-output system, a numerical example is given here based on the finite element model of the MAST truss beam structure as shown in Figure 4.4 (see Refs. 9 and 11 for detailed description). There are four actuators and 68 displacement sensors (four on each bay) distributed along the MAST truss beam structure[9,11]. There are five modal frequencies (see Table 4.3) and mode shapes derived from the finite element model. The mode shapes for the first and the second mode are shown in Figures 4.5 and 4.6, respectively. The odd-numbered sensors measure the deflections in one direction of bending, whereas the even-numbered sensors measure the deflections in another direction of bending. Note that for these two repeated modes, the mode shapes are orthogonal to each other. The C matrix used for the simulated data in Eq. (3) and for the projection filters in Eq. (35) is obtained from the five mode shapes derived from the analytical model. The B matrix used for this numerical simulation is arbitrarily determined from

$$Y(1) = CB$$

$$B = C^{+}\ Y(1)$$

where C^{+} is the pseudo-inverse matrix of the C matrix and Y(1) is the first test data.

The numerical example is illustrated by using the following parameters (see Eqs. (13)-(35)):

$$m = 4,\ p = 68,\ r = 3,\ s = 99,\ M = 5,\ T = 1/75 \text{ sec.}$$

Specific modal frequencies with different damping ratios and noise levels are shown in Table 4.3. The noise level is the ratio of the noise standard deviation with respect to the maximum value of Y(k), i.e. the peak of the free impulse response. For each case, the simulation starts by forming a Hankel matrix for this five-mode structure with a damping factor for all modes and a specific noise level. The simulated free impulse response data with 10% noise and zero damping are shown in Fig. 4.7. For each modal frequency, the frequency interval given for the projection filters is 0.1 to 10 Hz. With a fixed zero damping, the projection filters first update their frequencies by using the interval analysis method to find the global minimum of the cost function within the frequency interval. The midpoint of the final frequency interval (width is smaller than 0.001) is used for the first estimate of the modal frequency. With this estimated frequency, the projection filters then update their damping ratio with an initial interval from 0 to 5% by using the same interval analysis method to determine the damping from the Hankel matrix. The midpoint of the final damping interval (width is also smaller than 0.001) is used for the first estimate of the damping. With this new damping, the whole procedure is repeated. Since the second estimates of the modal frequency and damping are quite similar to the first estimates, further estimates are prohibited. The percentage errors for the second estimates of the damped modal frequency and damping are then calculated for each mode and listed in Tables 4.3 and 4.4. The cost function J for the first mode is plotted in Fig. 4.8 as the projection filters updated their frequencies with a fixed zero damping for noise free case. The cost function is almost unchanged when 10% noise is added. From Table 4.3, the errors of estimated modal frequencies fall within 1%. For repeated modal frequencies, the projection filters are shown to be effective to identify those frequencies. For a fixed noise level, the errors increase for most of the modes as the damping factor increases. This is caused by the fact that the signals decay faster for higher damping factors. However, there are some modes for which the estimates of the frequency improve slightly when the measurement noise level is increased. For a fixed data length, if the sample time T is reduced, the errors in frequency decrease for the noise free case and increase for higher noise level. From Table 4.4, the modal damping errors are much higher as compared to the modal frequency errors. Because the damping ratios of the large flexible structure are low, they are harder to be identified. As a result, this numerical simulation shows that the projection filters are feasible for estimating the modal frequencies and damping. The global optimization algorithm using interval analysis is also shown to be effective.

Chapter 5

CONCLUDING REMARKS

Projection filters are derived for possible application to the state estimation of linear dynamical systems. An approach to update the projection filter through the use of measurement data is developed to identify frequency and damping of the system. Numerical results for the MAST truss beam structure demonstrate that repeated modal frequencies can be identified within 1% error for 10% measurement noise.

There are two characteristics of the projection filters. First, each projection filter is developed based on a single-mode subsystem which identifies only one modal frequency and one modal damping within a specified region. Second, for an n modes structure (based on the analytical model), n single-mode projection filters can be implemented for parallel processing to reduce the computational burden. Application of the projection filters to the state estimation for linear dynamical systems is currently under investigation.

APPENDIX I

For a finite data length and a decoupled distinct M modes system as shown in Eqs. (37) and (38) without a measurement noise, it can be verified that

$$\lim_{T\to 0} V_j^{\#} H(0) W_j^{\#} = I_2 \tag{A1}$$

where $V_j^{\#}$ and $W_j^{\#}$ are the projection filters for j^{th} mode. On the other hand,

$$V_j^{\#} V_j W_j W_j^{\#} = I_2 \tag{A2}$$

where V_j and W_j are the respective generalized observability and controllability matrices for j^{th} mode subsystem.

From Eqs. (5), (6) and (A2), it is shown that

$$V_j^{\#} H(0) W_j^{\#} = V_j^{\#} V_r W_s W_j^{\#} = \sum_{k=1}^{M} V_j^{\#} V_k W_k W_j^{\#}$$

$$= I_2 + \sum_{\substack{k=1 \\ k\neq j}}^{M} V_j^{\#} V_k W_k W_j^{\#} \tag{A3}$$

Now, with the aids of Eqs. (15) and (23) and using the subscripts j and k to indicate the corresponding values for the j^{th} and k^{th} mode respectively, one obtains

$$V_j^{\#} V_k = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{A4}$$

where

$$a_{11} = (c_{2j}\, c_{2k}\, V_{aj}^{\#}\, V_{1k} + c_{2j}\, c_{1k}\, V_{aj}^{\#}\, V_{2k} + c_{1j}\, c_{2k}\, V_{bj}^{\#}\, V_{1k}$$

$$+ c_{1j}\, c_{1k}\, V_{bj}^{\#}\, V_{2k})/(c_{ij}^2 + c_{2j}^2)$$

with the aids of Eqs (17), (25) and (29)-(31), one derives

$$V^{\#}_{aj} V_{1k} = \sum_{i=0}^{r} e^{j_i(\sigma_j-\sigma_k)T} (\sin(j_i\ \omega_j T)\ \sin(j_i\ \omega_k T)$$

$$+ \sin(j_i\ \omega_k T)\ \sin((j_r-j_i)\omega_j T))/2\lambda_{1j} + \sum_{i=0}^{r} e^{j_i(\sigma_j-\sigma_k)T} (\sin(j_i\ \omega_j T)\sin(j_i\ \omega_k T)$$

$$- \sin(j_i\ \omega_k T)\ \sin((j_r-j_i)\omega_j T))/2\lambda_{2j} = F_1/(2\lambda_{1j}) + F_2/(2\lambda_{2j}) \qquad \text{(A5)}$$

If r is odd

$$\lambda_{1j} = (r+1)/2 + \sum_{i=0}^{(r-1)/2} \cos((j_r-2j_i)\omega_j T) = (r+1)/2 + Y(m) \qquad \text{(A6)}$$

Assume we choose j_i, t_k as follows

$$j_i = h_1 i, \qquad i = 0, 1, 2, \ldots, r \qquad \text{(A7)}$$

$$t_k = h_2 k, \qquad k = 0, 1, 2, \ldots, S \qquad \text{(A8)}$$

where h_1 and h_2 are positive integers. If h_1, h_2 and the sampling time T are chosen so that h_1T and h_2T are small, then

$$Y(m) = \sum_{i=0}^{(r-1)/2} \cos((r-2i)h_i\ \omega_j T)$$

$$= \sum_{i=0}^{(r-1)/2} \cos((2i+1)\omega_j\ h_1 T)$$

$$= \cos(\omega_j\ h_1 T)\ [\sum_{i=0}^{(r-1)/2} \cos(2\ \omega_j\ i\ h_1 T)]$$

$$- \sin(\omega_j\ h_1 T)\ [\sum_{i=0}^{(r-1)/2} \sin(2\omega_j\ i\ h_1 T)]$$

$$\approx \cos(\omega_j\ h_1T) \int_o^{(r-1)h_1T/2} \cos(2\omega_j t)/(h_1T)\ dt$$

$$- \sin(\omega_j\ h_1T) \int_o^{(r-1)h_1T/2} \sin(2\omega_j t)/(h_1T)\ dt$$

$$= \cos(\omega_j\ h_1T)\ \sin((r-1)\ \omega_j\ h_1T)/(2\ w_j\ h_1T)$$

$$+ \sin(\omega_j\ h_1T)\ [\cos((r-1)\omega_j\ h_1T) - 1]/(2\ \omega_j\ h_1T) \qquad \text{(A9)}$$

$$= (\sin(r\ \omega_j\ h_1T) - \sin(\omega_j\ h_1T))/(2\ \omega_j\ h_1T)$$

$$|Y(m)| \leqslant 1/\ (\omega_j\ h_1T) < \infty \qquad \text{(A10)}$$

Similary, from Eq. (A5), it is shown that

$$F_1 = \sum_{i=0}^{r} (\sin(j_i\ \omega_jT)\ \sin(j_i\ \omega_kT) + \sin(j_i\ \omega_kT)\ \sin((j_r-j_i)\omega_jT))\ e^{j_i(\sigma_j-\sigma_k)T}$$

$$= \sin\ (r\ \omega_j\ h_1T) \sum_{i=0}^{r} \sin(\omega_k\ i\ h_1T)\ \cos(\omega_j\ i\ h_1T)\ e^{(\sigma_j-\sigma_k)\ i\ h_1T}$$

$$+ (1 - \cos(r\ \omega_j\ h_1T)) \sum_{i=0}^{r} \sin\ (\omega_k\ i\ h_1T)\ \sin\ (\omega_j\ i\ h_1T)\ e^{(\sigma_j-\sigma_k)i\ h_1T}$$

$$\leqslant e^{|\sigma_j-\sigma_k|\ r\ h_1T} \{\sin(r\ \omega_j\ h_1T) \sum_{i=0}^{r} [\sin((\omega_k-\omega_j)i\ h_1t) + \sin((\omega_k+\omega_j)\ i\ h_1T)]$$

$$+ (1-\cos(r\ \omega_j\ h_1T)) \sum_{i=0}^{r} [\cos((\omega_k-\omega_j)i\ h_1T) - \cos((\omega_k+\omega_j)i\ h_1T)]\}$$

$$\approx e^{|\sigma_j-\sigma_k|r\ h_1T} \{\sin(r\ \omega_j\ h_1T)\ [(1-\cos((\omega_k-\omega_j)r\ h_1T))/(2(\omega_k-\omega_j)h_1T)$$

$$+ (1-\cos((\omega_k+\omega_j)\ r\ h_1T)/(2(\omega_k+\omega_j)\ h_1T)\ ]$$

$$+ (1-\cos(r\ \omega_j\ h_1T))\ [\sin((\omega_k-\omega_j)r\ h_1T)/((2(\omega_k-\omega_j)\ h_1T)$$

$$- \sin((\omega_k+\omega_j)\ r\ h_1T)/(2(\omega_k+\omega_j)\ h_1T)]\} \qquad \text{(A11)}$$

$$|F_1| \leqslant (2/(\omega_k-\omega_j) + 2/\ (\omega_k+\omega_j))\ e^{|\sigma_j-\sigma_k|\ r\ h_1T}/h_1T \qquad \text{(A12)}$$

For a finite data length, $r\ h_1T$ = constant, if $T\to 0$, then $r\to\infty$. From Eqs. (A6), (A10) and (A12), one obtains

$$\lim_{\substack{T\to 0\\ k\neq j}} F_1/(2\lambda_{1j}) = 0 \qquad \text{(A13)}$$

This is also true if r is even. Similarly, it can be proved that

$$\lim_{\substack{T\to 0\\ k\neq j}} F_2/(2\lambda_{2j}) = 0 \qquad \text{(A14)}$$

Substitution of Eqs. (A13) and (A14).into (A5) yields

$$\lim_{\substack{T\to 0\\ k\neq j}} V^{\#}_{aj}\ V_{1k} = 0 \qquad \text{(A15)}$$

For Eq. (A4), similar procedures can be used to verify

$$\lim_{\substack{T\to 0\\ k\neq j}} V^{\#}_{aj}\ V_{2k} = \lim_{\substack{T\to 0\\ k\neq j}} V^{\#}_{bj}\ V_{1k} = \lim_{\substack{T\to 0\\ k\neq j}} V^{\#}_{bj}\ V_{2k} = 0 \qquad \text{(A16)}$$

and

$$\lim_{\substack{T\to 0\\ k\neq j}} V^{\#}_{j}\ V_k = \lim_{\substack{T\to 0\\ k\neq j}} W_k\ W^{\#}_j = 0_2 \qquad \text{(A17)}$$

where 0_2 is a 2x2 zero matrix. Substution of Eq. (A17) into (A3) derives Eq. (A1). Note that a better approximation of Eq. (8) can be achieved if smaller T are used for a finite data length.

APPENDIX II

For the controllability matrix W and the observability matrix V shown in Eqs. (15) and (16), the corresponding orthogonal matrices, $W^{\#}$ and $V^{\#}$, can be derived as follows. First, observe that

$$V^{\#}\, V = W\, W^{\#} = I_2. \qquad \text{(B1)}$$

From Eqs. (17) and (25), it is shown that

$$V_a^{\#}(r)\; V_1(r) = \sum_{i=0}^{r} [\sin^2(j_i\omega T)\;(1/\lambda_1(r) + 1/\lambda_2(r)) + \sin(j_i\omega T)\sin((j_r - j_i)\omega T)$$

$$(1/\lambda_1(r) - 1/\lambda_2(r))]/2 = (1/2\lambda_1(r)) \sum_{i=0}^{r} [\sin^2 (j_i\omega T)$$

$$+ \sin (j_i\omega T)\sin((j_r - j_i)\omega T)] + 1/(2\lambda_2(r)) \sum_{i=0}^{r} [\sin^2 (j_i\omega T)$$

$$- \sin (j_i\omega T)\; \sin((j_r - j_i)\omega T)] \qquad \text{(B2)}$$

If r is even, with the aid of Eqs. (21) and (29)-(31), one obtains

$$\sum_{i=0}^{r} [\sin^2(j_i\omega T) + \sin(j_i\omega T)\; \sin((j_r - j_i)\omega T)]$$

$$= \sum_{i=0}^{r/2-1} [\sin(j_i\omega T) + \sin((j_r - j_i)\omega T)]^2 + 2\sin^2 (j_{r/2}\, \omega T)$$

$$= \sum_{i=0}^{r/2-1} [4\cos^2((j_i\; j_r/2)\; \omega T)\; \sin^2 (j_r\omega T/2) + 2\sin^2 (j_r\omega T/2)$$

$$= 2\sin^2 (j_r\omega T/2)\; [1 + \sum_{i=0}^{r/2-1} 2\cos^2((j_i - j_r/2)\omega T)]$$

$$= 2 \sin^2 (j_r\omega T/2) \left[1 + \sum_{i=0}^{r/2-1} (1 + \cos ((j_r - 2 j_i)\omega T))\right]$$

$$= 2 \sin^2 (j_r\omega T/2) \lambda_1(r) \tag{B3}$$

$$\sum_{i=0}^{r} [\sin^2 (j_i\omega T) - \sin(j_i\omega T) \sin((j_r - j_i)\omega T)]$$

$$= \sum_{i=0}^{r/2-1} [\sin(j_i\omega T) - \sin((j_r - j_i)\omega T)]^2$$

$$= \sum_{i=0}^{r/2-1} [4 \cos^2(j_r\omega T/2) \sin^2((j_i - j_r/2)\omega T)]$$

$$= 2 \cos^2 (j_r\omega T/2) \sum_{i=0}^{r/2-1} [1 - \cos((j_r - 2j_i)\omega T)]$$

$$= 2 \cos^2 (j_r\omega T/2) \lambda_2(r) \tag{B4}$$

If r is odd, with the aid of Eqs. (21) and (29)-(31), one arrives

$$\sum_{i=0}^{r} [\sin^2 (j_i\omega T) + \sin(j_i\omega T) \sin((j_r - j_i)\omega T)]$$

$$= \sum_{i=0}^{(r-1)/2} [\sin(j_i\omega T) + \sin((j_r - j_i)\omega T)]^2$$

$$= 2 \sin^2 (j_r\omega T/2) \sum_{i=0}^{(r-1)/2} [1 + \cos((j_r - 2j_i)\omega T)]$$

$$= 2 \sin^2 (j_r\omega T/2) \lambda_1(r) \tag{B5}$$

$$\sum_{i=0}^{r} [\sin^2 (j_i\omega T) - \sin(j_i\omega T) \sin ((j_r - j_i) \omega T)]$$

$$= \sum_{i=0}^{(r-1)/2} [\sin(j_i\omega T) - \sin((j_r - j_i)\omega T)]^2$$

$$= 2\cos^2(j_r\omega T/2) \sum_{i=0}^{(r-1)/2} [1-\cos((j_r-2j_i)\omega T)]$$

$$= 2\cos^2(j_r\omega T/2)\,\lambda_2(r) \tag{B6}$$

Substitution of Eqs. (B3)-(B6) into (B2) yields

$$V_a^{\#}(r)\,V_1(r) = [2\sin^2(j_r\omega T/2)\,\lambda_1(r)]/2\lambda_1(r) + [2\cos^2(j_r\omega T/2)\lambda_2(r)]/2\lambda_2(r)$$

$$= 1 \tag{B7}$$

Similarly

$$V_b^{\#}(r)\,V_2(r) = 1 \tag{B8}$$

Next, from Eqs. (18) and (25), it is shown that

$$V_a^{\#}(r)\,V_2(r) = -\sum_{i=0}^{r} [\sin(j_i\omega T)\cos(j_i\omega T) + \sin((j_r - j_i)\omega T)\cos(j_i\omega T)]/2\lambda_1(r)$$

$$+ \sum_{i=0}^{r} [-\sin(j_i\omega T)\cos(j_i\omega T) + \sin((j_r - j_i)\omega T)\cos(j_i\omega T)]/2\lambda_2(r) \tag{B9}$$

If r is even, with the aid of Eqs. (21) and (29)-(31), one obtains

$$\sum_{i=0}^{r} [\sin(j_i\omega T)\cos(j_i\omega T) + \sin((j_r - j_i)\omega T)\cos(j_i\omega T)]$$

$$= \sum_{i=0}^{r/2-1} [\sin(j_i\omega T) + \sin((j_r - j_i)\omega T)]\,[\cos(j_i\omega T) + \cos((j_r - j_i)\omega T)]$$

$$+ 2 \sin (j_r \omega T/2) \cos(j_r \omega T/2)$$

$$= \sum_{i=0}^{r/2-1} 4 \sin(j_r \omega T/2) \cos((j_i - j_r/2)\omega T) \cos(j_r \omega T/2) \cos((j_i - j_r/2)\omega T)$$

$$+ \sin(j_r \omega T)$$

$$= \sin(j_r \omega T) \left[1 + \sum_{i=0}^{r/2-1} 2 \cos^2((j_i - j_r/2)\omega T)\right]$$

$$= \sin(j_r \omega T) \left[1 + \sum_{i=0}^{r/2-1} (1 + \cos((j_r - 2 j_i)\omega T)\right]$$

$$= \sin(j_r \omega T) \lambda_1(r) \qquad \text{(B10)}$$

$$\sum_{i=0}^{r} [- \sin(j_i \omega T) \cos(j_i \omega T) + \sin ((j_r - j_i)\omega T) \cos(j_i \omega T)]$$

$$= \sum_{i=0}^{r/2-1} [\sin((j_r - j_i)\omega T) - \sin(j_i \omega T)] [\cos(j_i \omega T) - \cos((j_r - j_i)\omega T)]$$

$$= \sum_{i=0}^{r/2-1} 4 [\sin((j_r/2 - j_i)\omega T) \cos(j_r \omega T/2)] [- \sin(j_r \omega T/2) \sin((j_i - j_r/2)\omega T)]$$

$$= \sin(j_r \omega T) \sum_{i=0}^{r/2-1} 2 \sin^2((j_r/2 - j_i)\omega T)$$

$$= \sin(j_r \omega T) \sum_{i=0}^{r/2-1} [1 - \cos((j_r - 2 j_i)\omega T)]$$

$$= \sin(j_r \omega T) \lambda_2(r) \qquad \text{(B11)}$$

Substitution of Eqs. (B10) and (B11) into (B13) yields

$$V_a^{\#}(r)\ V_2(r) = -\ [\sin(j_r\omega T)\lambda_1(r)]/2\lambda_1(r) + [\sin(j_r\omega T)\lambda_2(r)]/2\lambda_2(r) = 0 \tag{B12}$$

This is also true if r is odd. Similarly, it can be proved that

$$V_b^{\#}(r)V_1(r) = 0 \tag{B13}$$

Observation of Eqs. (15), (23), (B7), (B8), (B12) and (B13) leads to

$$V^{\#}\ V = I_2 \tag{B14}$$

Similar procedures can be used to verify

$$WW^{\#} = I_2 \tag{B15}$$

Note that, if $\sigma = 0$, from Eqs. (15), (16), (23), (24), it can be proved that

$$(VV^{\#})^T = VV^{\#} \quad , \quad (W^{\#}W)^T = W^{\#}W \tag{B16}$$

From Eqs. (B14) - (B16), it shows that $V^{\#}$ and $W^{\#}$ are the pseudo inverse matrices of V and W respectively. However, for $\sigma \neq 0$, $V^{\#}$ and $W^{\#}$ are not the pseudo inverse matrices. This raises doubts about the uniqueness of the modal parameters identified. The effect of this non-unique parameters needs further study in the future.

REFERENCES

[1] Juang, J.-N., "Mathematical Correlation of Modal Parameter Indentification Methods Via System Realization Theory," The International Journal of Analytical and Experimental Modal Analysis, Vol. 2, No. 1, Jan. 1987, pp. 1-18.

[2] Hyland, D. C. and Bernstein, D. S., "The Optimal Projection Equations for Model Reduction and the Relationships Among the Methods of Wilson, Skelton, and Moore," IEEE Transactions on Automatic Control, Vol. AC-30, No. 12, Dec. 1985, pp. 1201-1211.

[3] Mottershead, J. E. and Stanway, R., "Identification of Structural Vibration Parameters by Using a Frequency Domain Filter," Journal of Sound and Vibration, Vol. 109(3), 1986, pp. 495-506.

[4] Juang, J.-N. and Pappa, R. S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," Journal of Guidance, Control, and Dynamics, Vol. 8, Sept.-Oct. 1985, pp. 620-627.

[5] Hansen, E. R., "Global Optimization Using Interval Analysis: The One-Dimensional Case," Journal of Optimization Theory and Applications, Vol. 29, No. 3, Nov. 1979, pp. 331-344.

[6] Hansen, E. R. and Sengupta, S., "Global Constrained Optimization Using Interval Analysis," Interval Mathematics, Academic Press, 1980, pp. 25-47.

[7] Walster, G. W., Hansen, E. R. and Sengupta, S., "Test Results for a Global Optimization Algorithm," Numerical Optimization 1984, SIAM Publication, 1985, pp. 272-287.

[8] Hansen, E. R. and Greenberg, R. J., "An Interval Newton Method," Applied Math. and Computation, Vol. 12, 1983, pp. 89-98.

[9] Horta, L. G., Walsh, J. L. and Horner, G. C., "Analysis and Simulation of the MAST (COFS-I Flight Hardware)," First NASA/DOD CSI Technology Conference, Norfolk, VA, November 18-21, 1986, pp. 515-532.

[10] Huang, J.-K., Juang, J.-N. and Chen, C.-W., "Single-Mode Projection Filters for Identification and State Estimation of Flexible Structures," AIAA Guidance, Navigation and Control Conference, Monterey, CA, August 17-19, 1987, pp. 595-604.

[11] Talcott, R. C. and Shipley, J. W., "Description of the MAST Flight System," First NASA/DOD CSI Technology Conference, Norfolk, VA, November 18-21, 1986, pp. 253-263.

[12] Guthbert, T. R., Jr., "Optimization Using Personal Computers with Applications to Electrical Networks," A Wiley-Interscience Publication, John Wiley & Sons, N. Y., 1986.

Table 3.1 The Global Minimum of the Three Hump Camel Function

No.	x_1	x_2	$f(x_1, x_2)$
1	0.000	0.000	0.00

Table 3.2 The Global Maxima of the Himmelblau Function

No.	x_1	x_2	$F(x_1,x_2)$
1.	3.0000	2.0000	0.00
2.	3.5844	-1.8481	0.00
3.	-3.7793	-3.2832	0.00
4.	-2.0851	3.1313	0.00

Table 4.1. Percentage Error of Estimated Modal Frequency

Damping ratios	0.3%				2%			
Noise Frequency (rad/sec)	0%	5%	10%	30%	0%	5%	10%	30%
3	0.86	1.10	1.33	2.28	0.75	1.00	1.26	2.23
7	0.03	0.33	0.66	1.95	0.04	0.42	0.65	2.03
11	0.18	0.26	0.59	1.63	0.17	0.39	0.82	1.92
17	0.01	0.17	0.33	0.99	0.01	0.25	0.53	1.72
23	0.02	0.08	0.13	0.31	0.01	0.12	0.21	0.48
31	0.01	0.17	0.32	0.84	0.02	0.57	0.96	1.60
37	0.01	0.02	0.06	0.35	0.02	0.42	0.93	1.70
43	0.01	0.18	0.36	0.93	0.00*	0.79	1.17	1.54
53	0.00*	0.03	0.06	0.14	0.01	0.13	0.21	0.36
67	0.00*	0.13	0.27	0.63	0.00*	0.90	0.95	1.01

*after numerical truncation

Table 4.2. Percentage Error of Estimated Modal Damping

Damping ratios Noise Frequency (rad/sec)	0.3%				2%			
	0%	5%	10%	30%	0%	5%	10%	30%
3	128.20	*	*	*	17.21	28.29	73.22	*
7	3.51	*	*	*	0.33	53.06	*	*
11	20.79	*	*	*	3.34	25.72	52.03	*
17	0.21	4.88	11.03	60.52	0.13	0.18	0.46	15.92
23	13.19	1.93	14.73	65.56	0.81	7.42	14.42	35.66
31	10.53	1.42	17.92	*	2.68	1.66	13.07	68.62
37	1.99	24.73	46.44	126.50	0.11	13.03	9.71	27.50
43	0.13	0.77	7.07	*	0.89	12.90	34.60	85.88
53	0.49	15.97	32.01	91.41	0.19	14.54	25.78	55.24
67	1.65	9.09	5.23	76.90	0.31	40.22	58.45	88.26

* The estimated modal damping is 0 %

Table 4.3 Percentage Error of Estimated Modal Frequency of the MAST Beam Structure

Damping ratio	0%			0.3%			2%		
Noise Level Freq (Hz)	0%	5%	10%	0%	5%	10%	0%	5%	10%
1.4222	0.00#	0.01	0.02	0.01	0.00#	0.01	0.00#	0.00#	0.01
1.4222	0.02	0.06	0.16	0.03	0.06	0.15	0.03	0.09	0.18
8.5545	0.21	0.21	0.20	0.23	0.23	0.23	0.38	0.40	0.42
9.4954	0.82	0.80	0.78	0.84	0.81	0.79	0.97	0.95	0.90
9.4954	0.29	0.30	0.30	0.30	0.31	0.31	0.43	0.43	0.42

# after numerical truncation

Table 4.4 Percentage Error of Estimated Modal Damping of the MAST Beam Structure

Damping ratio	0%			0.3%			2%		
Noise Level Freq (Hz)	0%	5%	10%	0%	5%	10%	0%	5%	10%
1.4222	*	*	*	0.33	3.33	6.67	0.00#	0.50	1.00
1.4222	*	*	*	6.0	10.00	13.33	0.00#	0.50	1.00
8.5545	*	*	*	*	*	*	41.0	41.50	42.00
9.4954	*	*	*	*	*	*	75.0	77.00	78.50
9.4954	*	*	*	83.3	96.67	*	38.5	42.00	44.00

* The estimated modal damping is 0%
# after numerical truncation

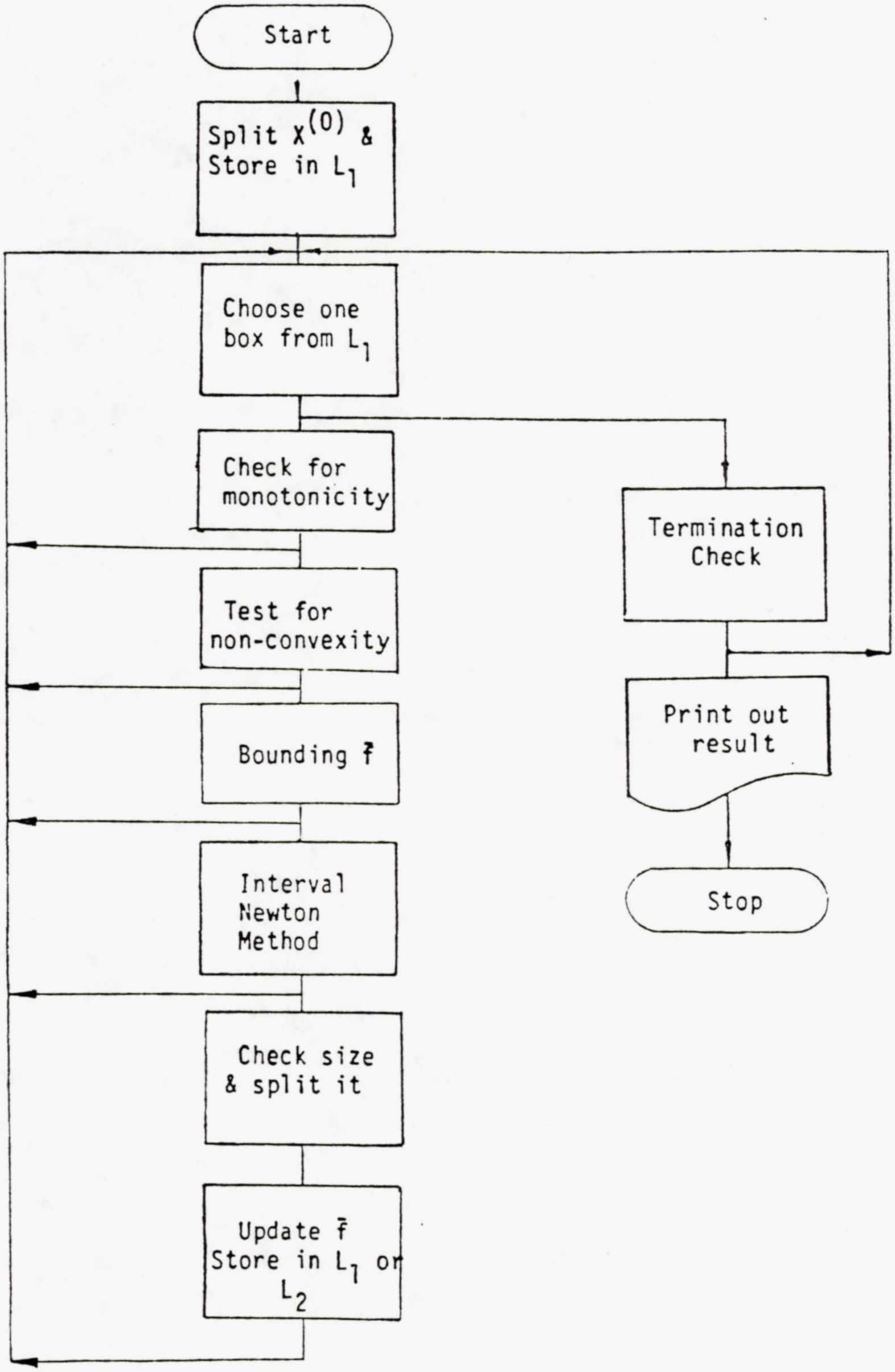

Fig. 3.1. The Steps of the Optimization Algorithm

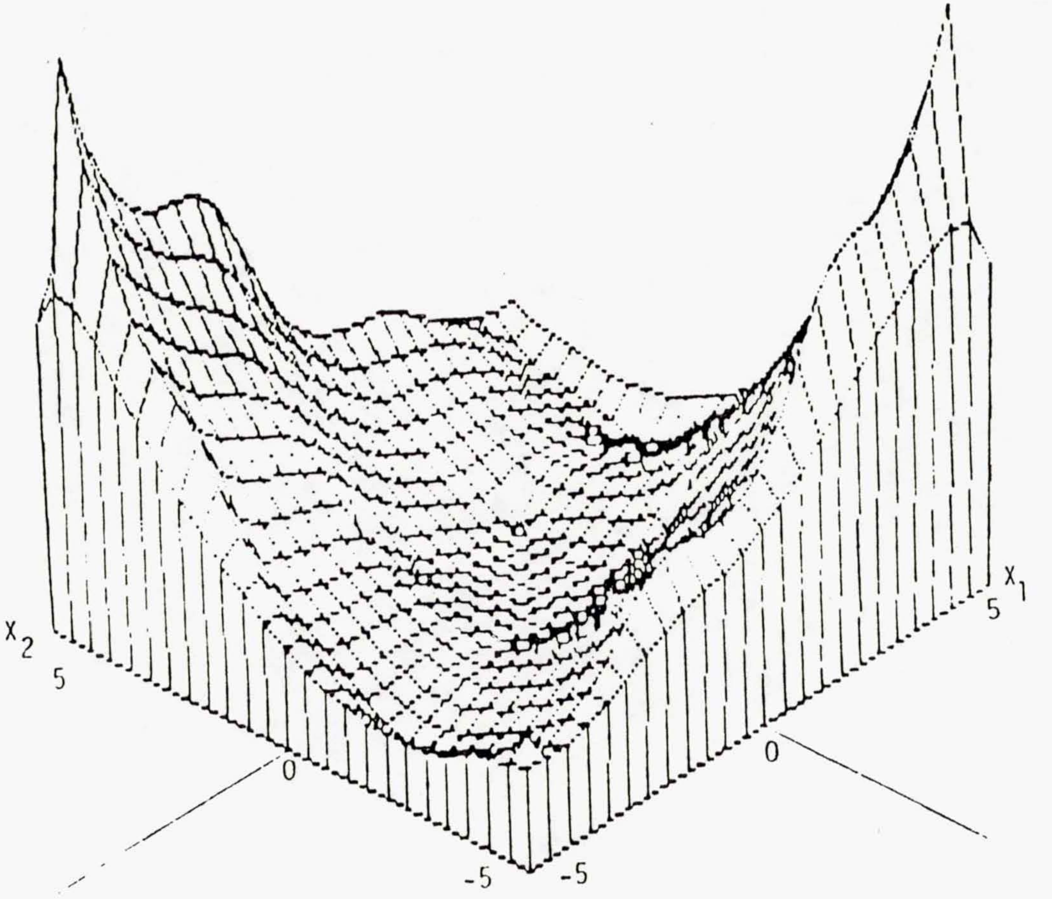

Fig. 3.2 A Surface Plot of the Three Hump Camel Function

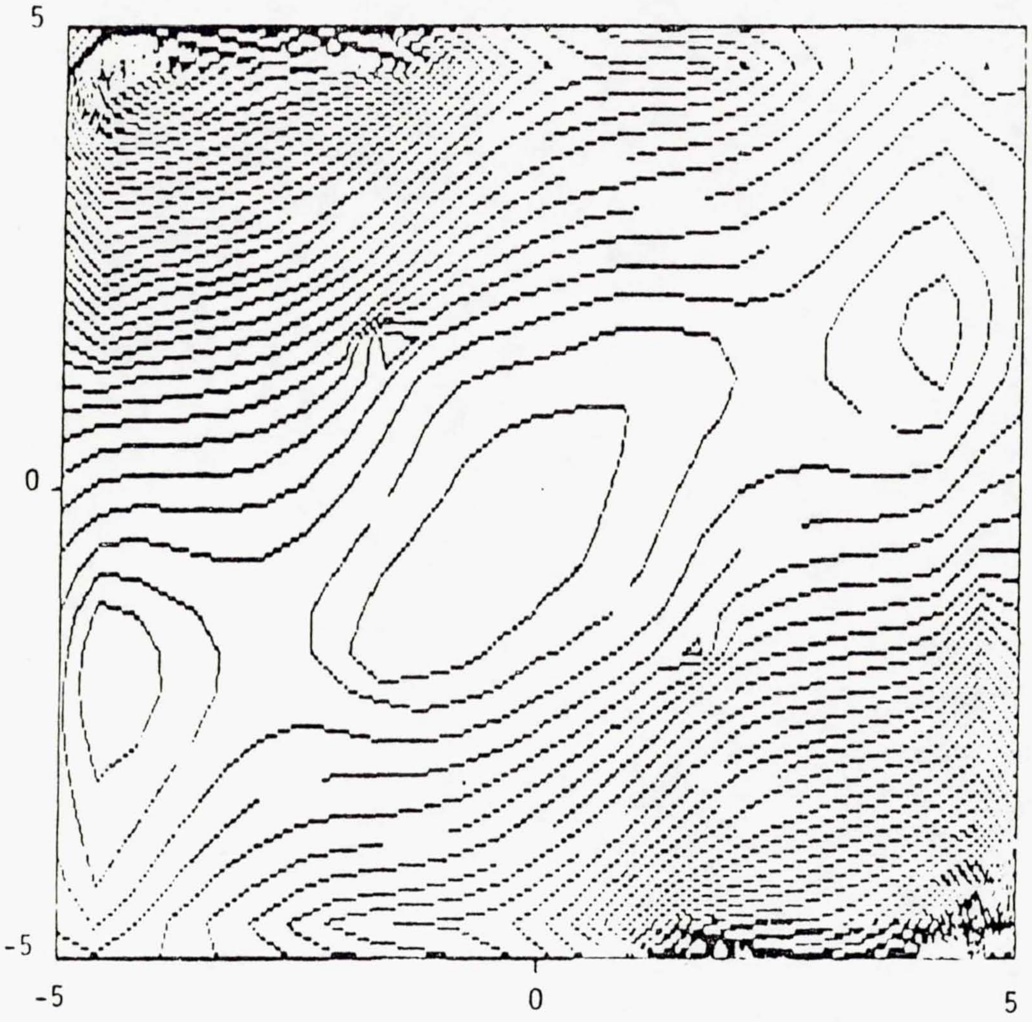

Fig. 3.3 Contours of the Surface of the Three Hump Camel Function

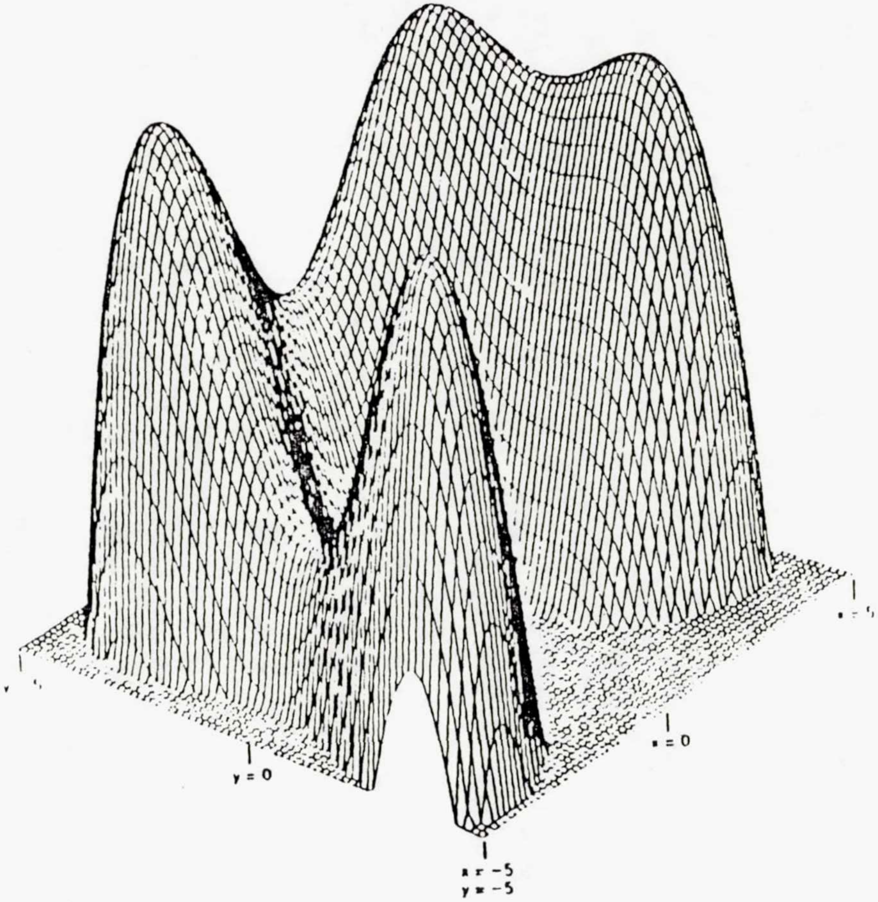

Fig. 3.4 A Surface Plot of the Himmelblau Function

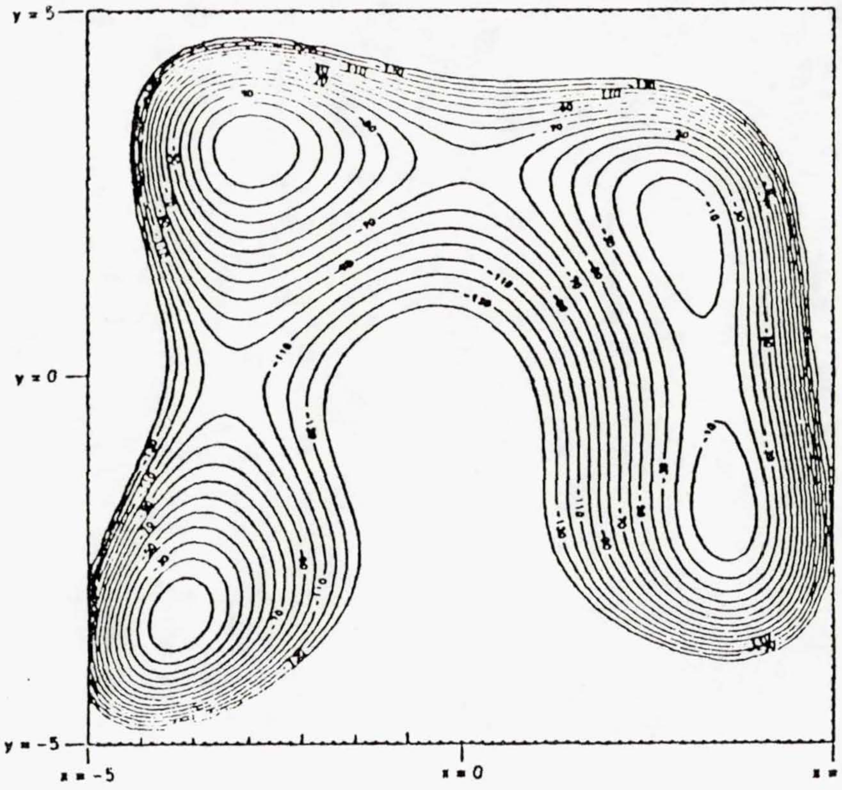

Fig. 3.5 Contours of the Surface of the Himmelblau Function

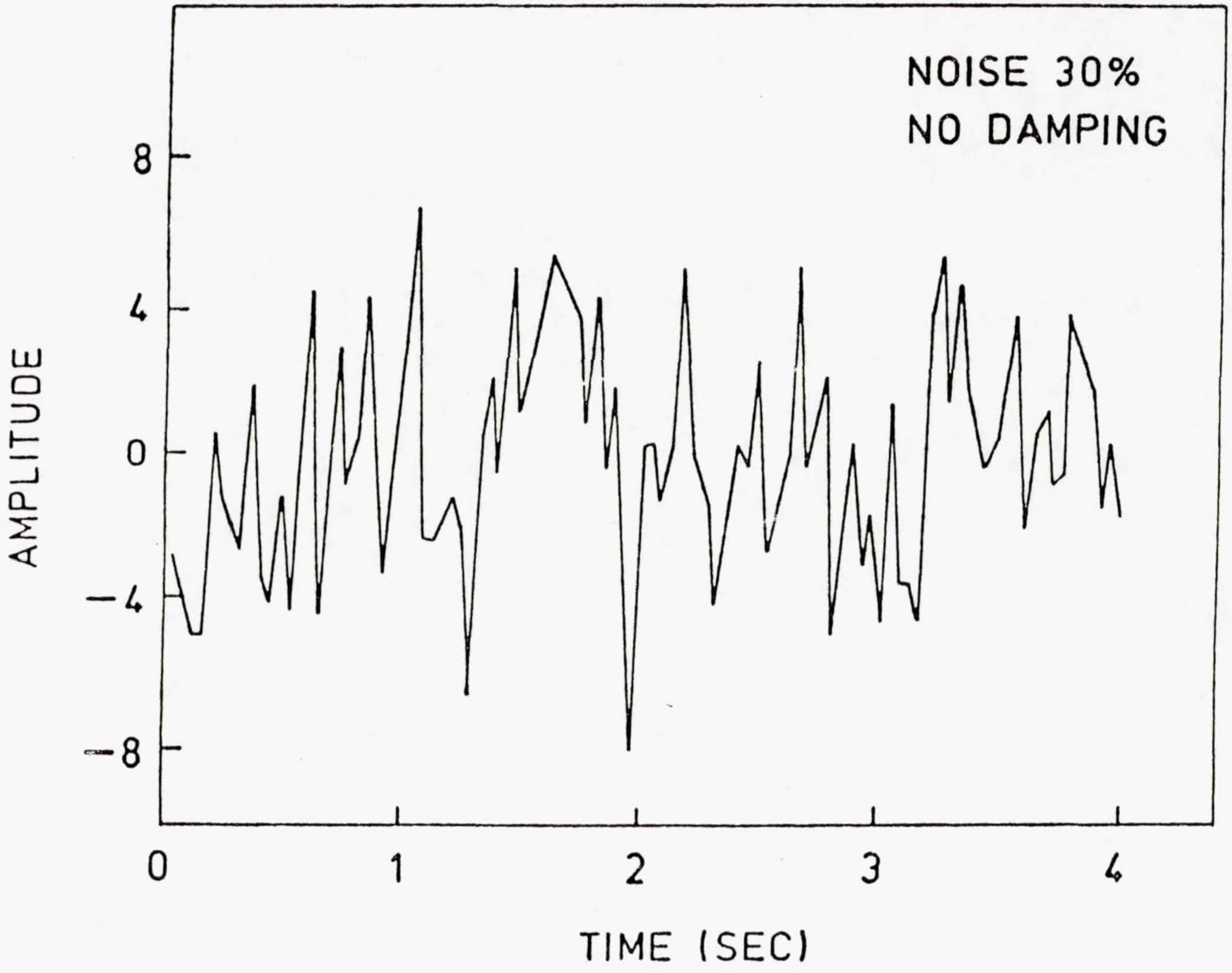

Fig. 4.1 Simulated Free Impulse Response Data

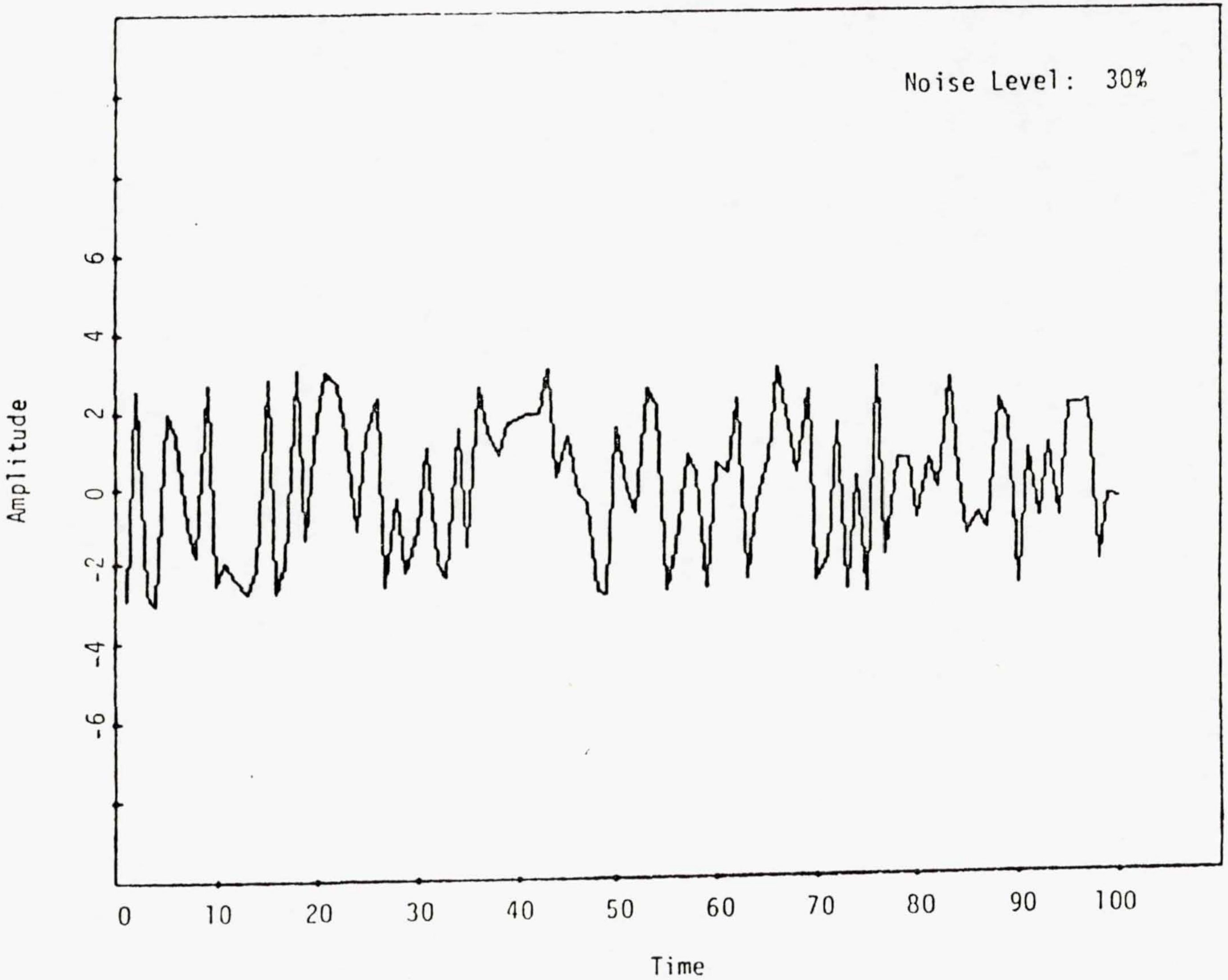

Fig. 4.2 Simulated Measurement White Noise

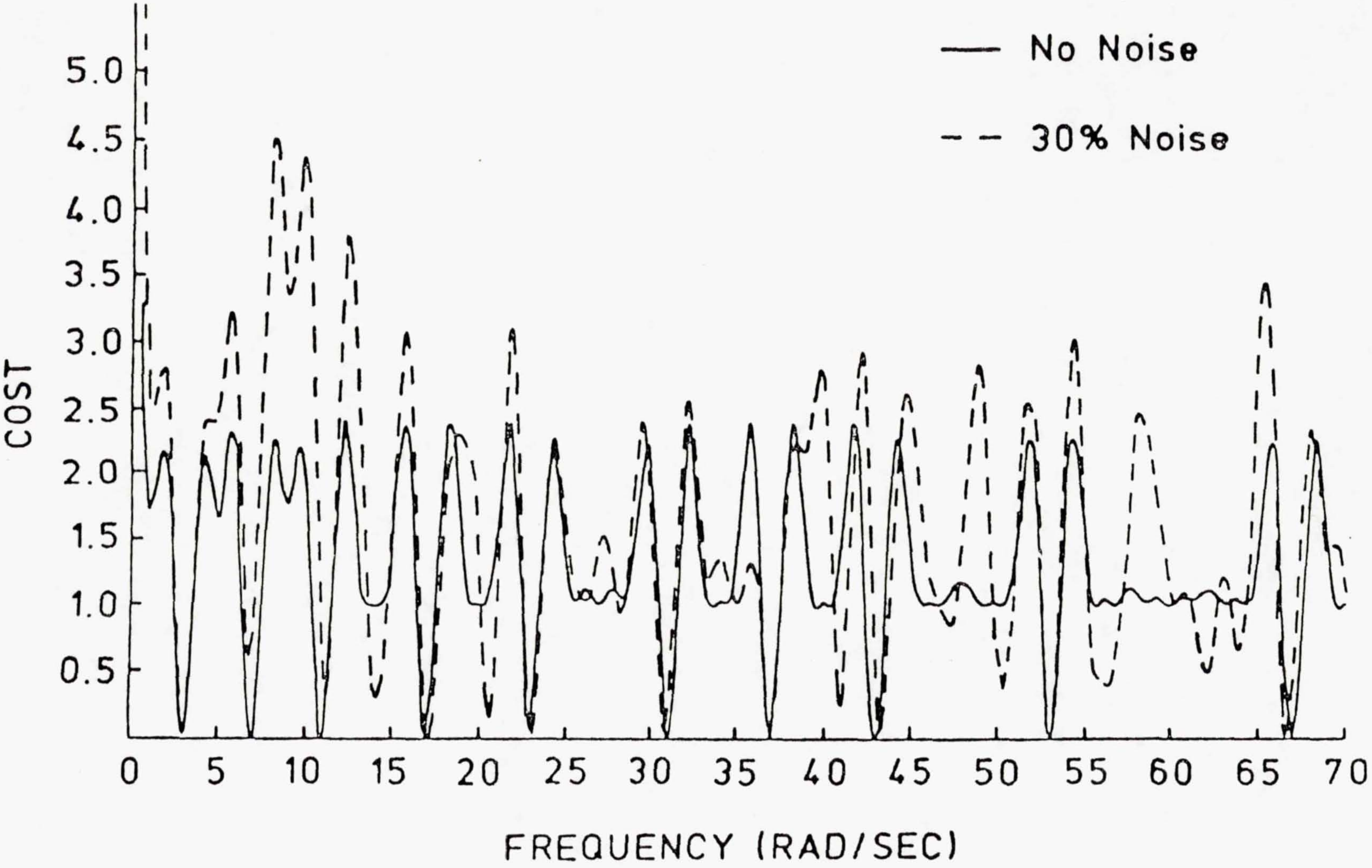

Fig. 4.3 Cost as a Function of Updated Frequencies

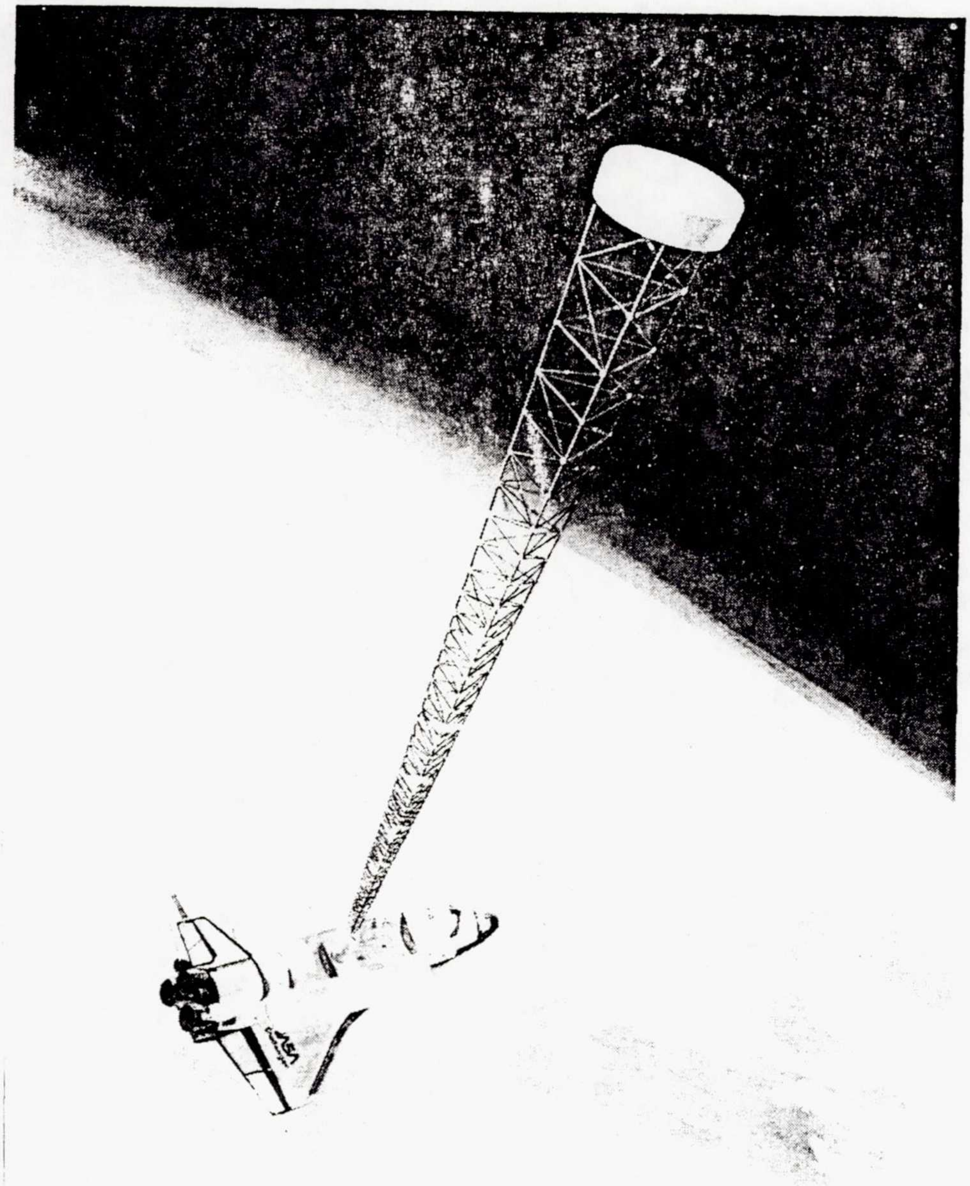

Fig. 4.4. The MAST truss beam structure

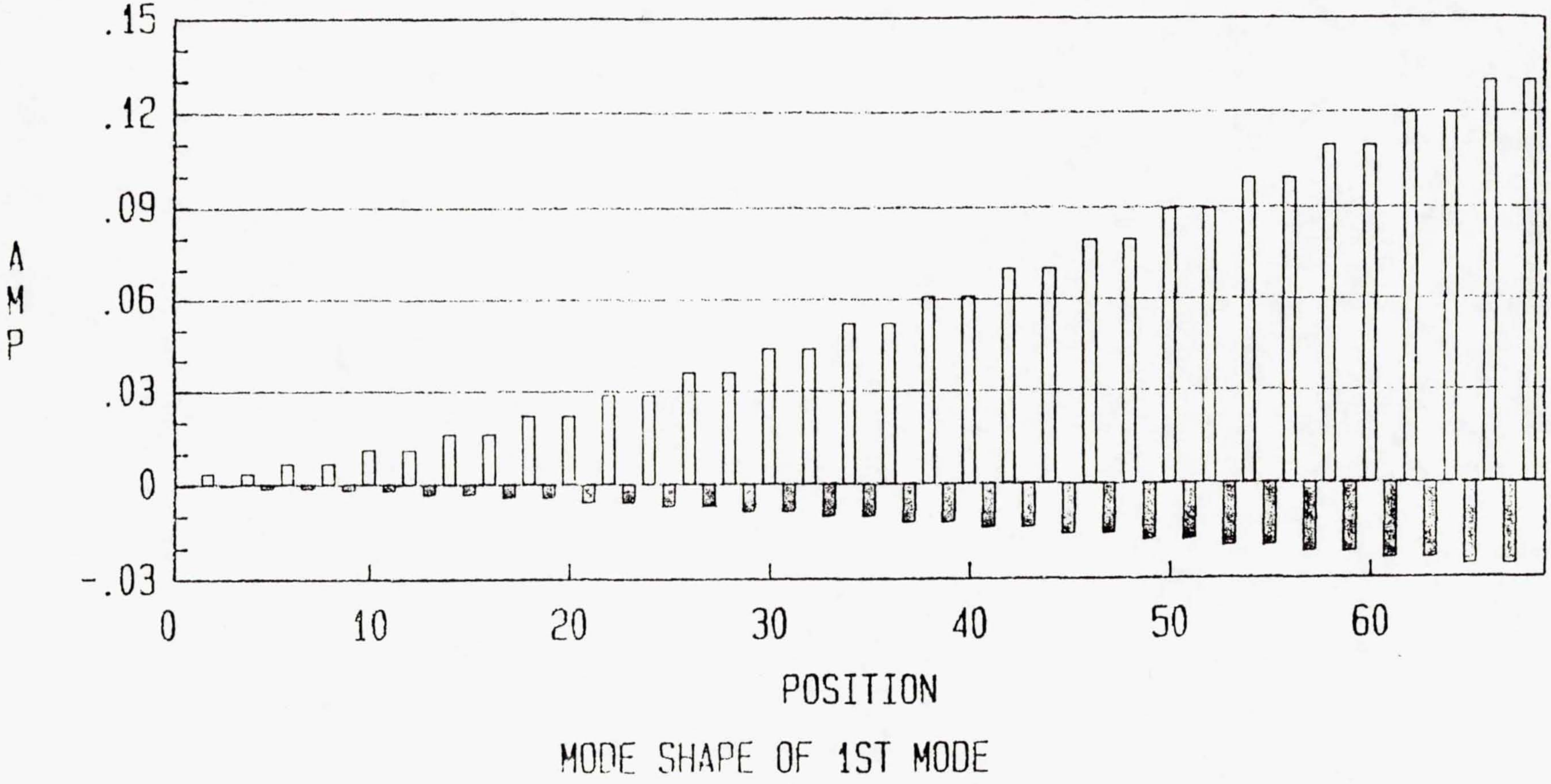

Fig. 4.5 Mode shape of the first mode of the MAST truss beam structure. The solid bars indicate the displacement in the x-direction. The empty bars indicate the displacements in the y-direction. There are two measurements on each bay for each direction.

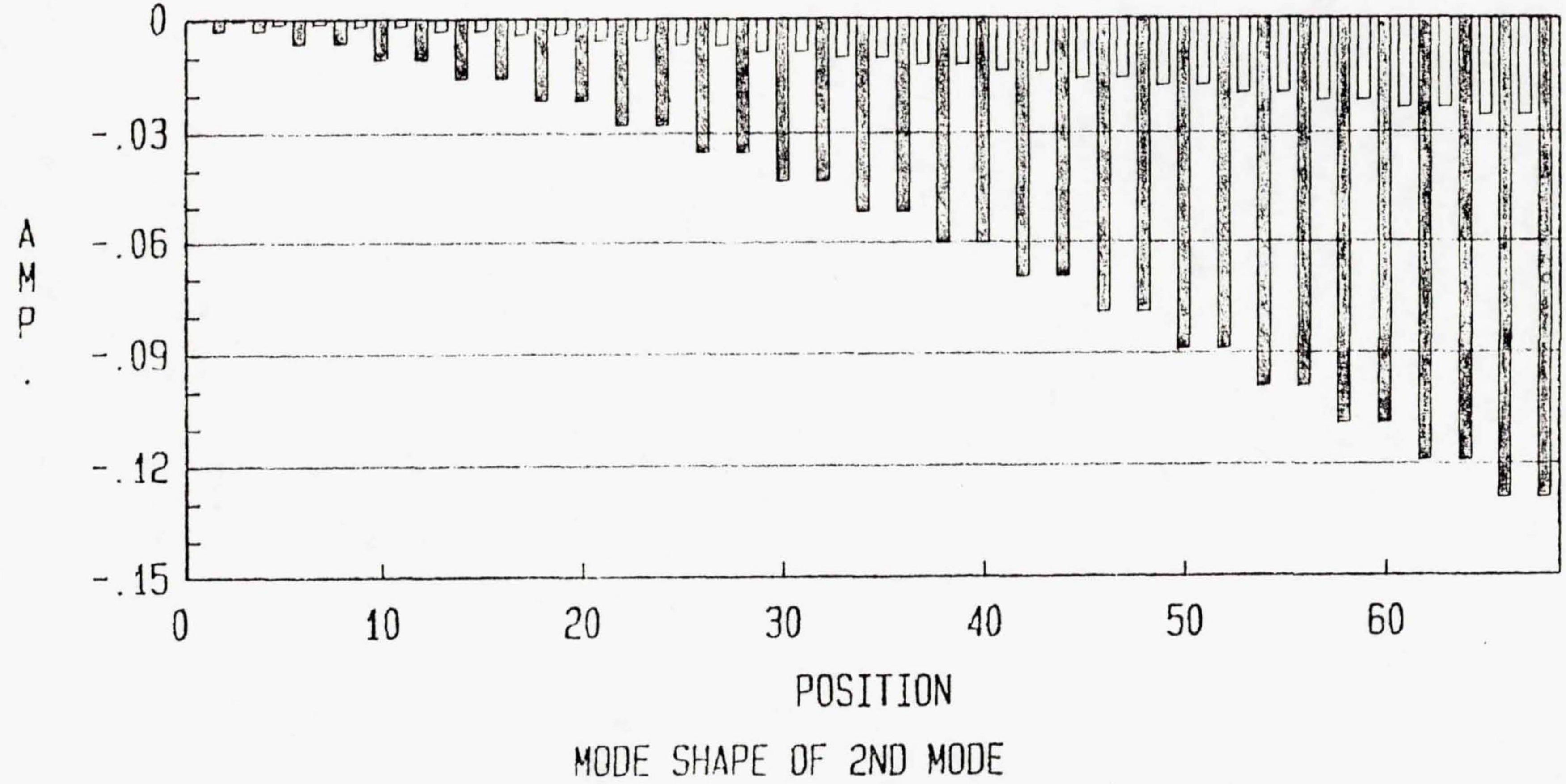

Fig. 4.6 Mode shape of the second mode of the MAST truss beam structure. The solid bars indicate the displacements in the x-direction. The empty bars indicate the displacements in the y-direction. There are two measurements on each bay for each direction.

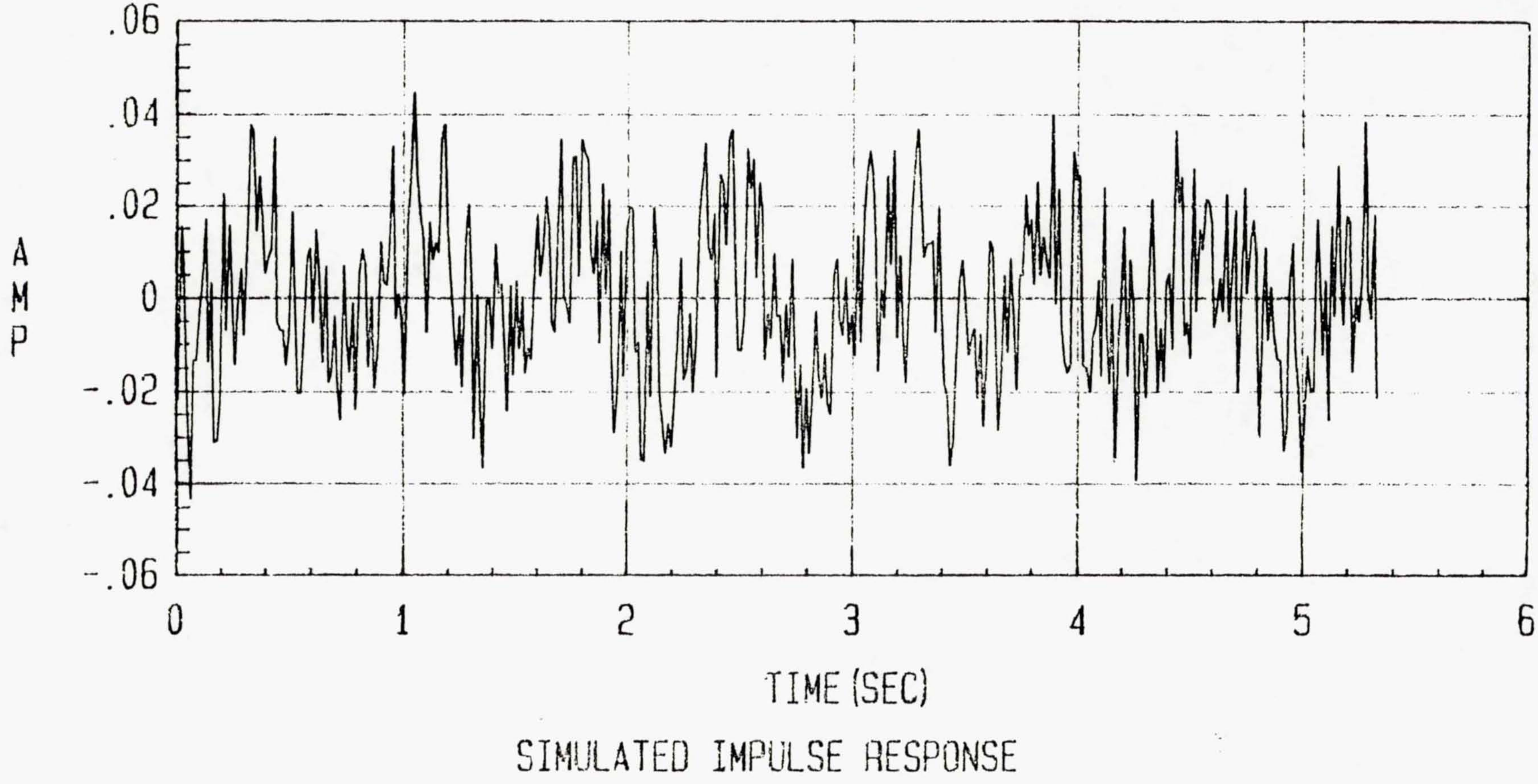

Fig. 4.7. The simulated free impulse response data with 10% noise and zero damping.

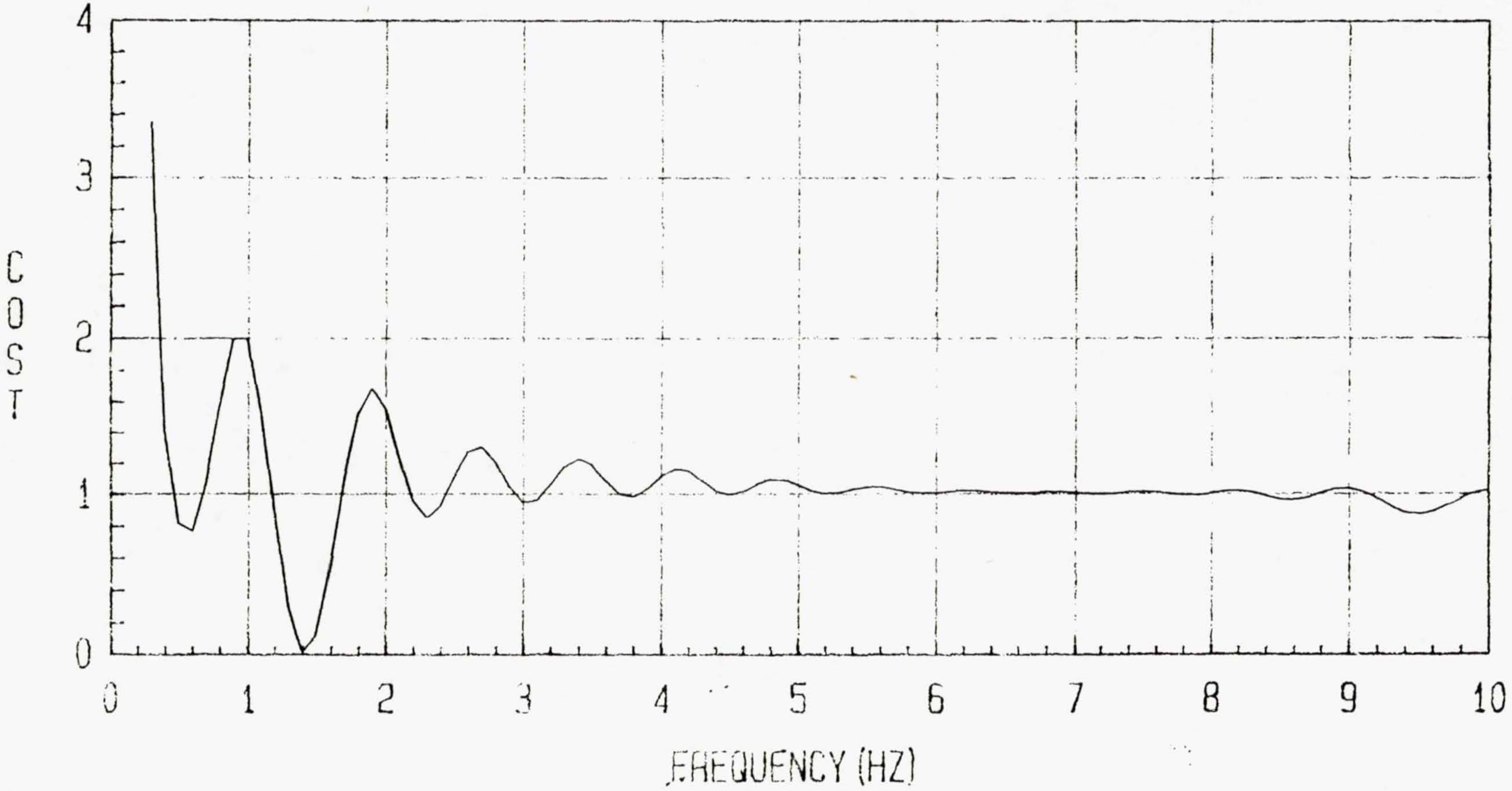

Fig. 4.8 The cost function J for the first mode as the projection filters update their frequencies with a fixed zero damping for noise free case.