

17  
72

**Parametric Identification  
of Nonlinear Structural Dynamic Systems**

by

James Brian Normann

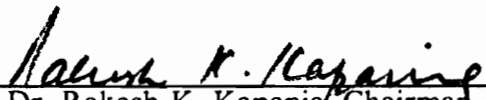
Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Master of Science

in

Aerospace Engineering

APPROVED:

  
\_\_\_\_\_  
Dr. Rakesh K. Kapania, Chairman

  
\_\_\_\_\_  
Dr. Eric R. Johnson

  
\_\_\_\_\_  
Dr. Mahendra P. Singh

May 17, 1989

Blacksburg, Virginia

2

LD  
5655  
V855  
1989  
Nu74  
C.2

**Parametric Identification  
of Nonlinear Structural Dynamic Systems**

by

James Brian Normann

Dr. Rakesh K. Kapania, Chairman

Aerospace Engineering

(ABSTRACT)

The identification of linear structural dynamic systems has been dealt with extensively in past studies. Identification methods for nonlinear structures have also been introduced in previous articles, including procedures based on the method of multiple scales, iterative and noniterative direct methods, and state space mappings. Here, a procedure is introduced for the identification of nonlinear structural dynamic systems which is readily applicable to simple as well as more complex multiple degree of freedom systems. The procedure is based on multiple step integration methods for the solution of differential equations. The multiple step integration procedure and the iterative direct method are applied to a number of nonlinear single degree of freedom examples, and are applied to a simple two degrees of freedom example as well. RMS based noise is added to a simulated measured response in order to monitor the effects of measurement errors on the procedures. The input data is filtered before final processing in the identification algorithms. The multistep algorithm is compared to the iterative direct method on the basis of criteria such as accuracy, ease of use, and numerical efficiency.

## **Acknowledgements**

I am greatly indebted to my parents for their support throughout my accademic career, and would like to take this opportunity to thank them. Special thanks also to Dr. Rakesh K. Kapania for his time and consideration in advising me on both research and nonresearch related topics. Thanks are also due to Dr. E. R. Johnson and Dr. M. P. Singh for serving on my advisory committee.

# Table of Contents

<b>1.0 Introduction</b> .....	<b>1</b>
<b>2.0 Iterative Direct Method : Single Degree of Freedom Systems</b> .....	<b>9</b>
2.1 Objective Function Formulation .....	11
2.2 Objective Function Differentiation .....	12
2.3 Objective Function Minimization .....	13
2.4 Treatment of Noise .....	17
2.5 Applications of The Iterative Direct Method .....	19
2.5.1 Quadratic Nonlinearity Example .....	23
2.5.2 Quadratic and Cubic Nonlinearity Example .....	28
2.5.3 Drag Nonlinearity Example .....	31
2.6 Results .....	34
<b>3.0 Single Step Integration Iterative Method : Single Degree of Freedom Systems</b> .....	<b>52</b>
3.1 Objective Function Formulation .....	54
3.2 Objective Function Differentiation .....	55
3.3 Objective Function Minimization .....	57

3.4 Applications of the One Step Integration Iterative Method .....	57
3.4.1 Quadratic Nonlinearity Example .....	58
3.4.2 Quadratic and Cubic Nonlinearity Example .....	60
3.4.3 Drag Nonlinearity Example .....	62
3.5 Results .....	65
<b>4.0 Multiple Step Integration Iterative Method : Single Degree of Freedom Systems .....</b>	<b>74</b>
4.1 Objective Function Formulation .....	75
4.2 Objective Function Differentiation .....	76
4.3 Objective Function Minimization .....	77
4.4 Applications of the Multistep Method .....	78
4.4.1 Quadratic Nonlinearity Example .....	80
4.4.2 Quadratic and Cubic Nonlinearity Example .....	81
4.4.3 Drag Nonlinearity Example .....	82
4.5 Results .....	83
<b>5.0 Iterative Direct Method: Two Degrees of Freedom System .....</b>	<b>98</b>
5.1 Objective Function Formulation .....	99
5.2 Objective Function Differentiation .....	99
5.3 Objective Function Minimization .....	100
5.4 Cubic Nonlinear Two Degrees of Freedom System .....	101
5.5 Results .....	108
<b>6.0 Single Step Integration Method: Two Degrees of Freedom System .....</b>	<b>109</b>
6.1 Objective Function Formulation .....	109
6.2 Objective Function Differentiation .....	112
6.3 Objective Function Minimization .....	115
6.4 Results .....	116

<b>7.0 Multiple Step Integration Method: Two Degrees of Freedom System</b>	<b>119</b>
7.1 Objective Function Formulation	119
7.2 Objective Function Differentiation	120
7.3 Objective Function Minimization	123
7.4 Results	123
<b>8.0 Conclusions and Future Work</b>	<b>131</b>
8.1 Future Work	132
<b>Bibliography</b>	<b>134</b>
<b>Appendix A. Program Mult</b>	<b>138</b>
<b>Vita</b>	<b>165</b>

# List of Illustrations

Figure 1. Spectral Density of Quadratic and Cubic Nonlinearity Example . . . . .	20
Figure 2. Frequency Response of Low Pass Filter . . . . .	21
Figure 3. Effect of Filtering of Input Displacement Data . . . . .	22
Figure 4. Identified/Measured Response of Quadratic Nonlinearity Example . . . .	46
Figure 5. Identified/Measured Response of Quadratic and Cubic Nonlinearity Ex- ample . . . . .	47
Figure 6. Identified/Measured Response of Drag Nonlinearity Example . . . . .	48
Figure 7. Identified/Measured Response of Quadratic Nonlinearity Example . . . .	49
Figure 8. Identified/Measured Response of Quadratic and Cubic Nonlinearity Ex- ample . . . . .	50
Figure 9. Identified/Measured Response of Drag Nonlinearity Example . . . . .	51
Figure 10. Identified/Measured Response of Quadratic Nonlinearity Example . . . .	92
Figure 11. Identified/Measured Response of Quadratic and Cubic Nonlinearity Ex- ample . . . . .	93
Figure 12. Identified/Measured Response of Drag Nonlinearity Example . . . . .	94
Figure 13. Identified/Measured Response of Quadratic Nonlinearity Example . . . .	95
Figure 14. Identified/Measured Response of Quadratic and Cubic Nonlinearity Ex- ample . . . . .	96
Figure 15. Identified/Measured Response of Drag Nonlinearity Example . . . . .	97
Figure 16. Identified/Measured Response of 2-D Nonlinearity Example . . . . .	127
Figure 17. Identified/Measured Response of 2-D Nonlinearity Example . . . . .	128



Figure 18. Identified/Measured Response of 2-D Nonlinearity Example . . . . . 129  
Figure 19. Identified/Measured Response of 2-D Nonlinearity Example . . . . . 130

# List of Tables

Table 1.	Iterative Direct Method; Quadratic Numerical Results, Unfiltered Input Data . . . . .	40
Table 2.	Iterative Direct Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data . . . . .	41
Table 3.	Iterative Direct Method; Drag Numerical Results, Unfiltered Input Data . . . . .	42
Table 4.	Iterative Direct Method; Quadratic Numerical Results, Filtered Input Data . . . . .	43
Table 5.	Iterative Direct Method; Quadratic, Cubic Numerical Results, Filtered Input Data . . . . .	44
Table 6.	Iterative Direct Method; Drag Numerical Results, Filtered Input Data . . . . .	45
Table 7.	Single Step Integration Method; Quadratic Numerical Results, Unfiltered Input Data . . . . .	68
Table 8.	Single Step Integration Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data . . . . .	69
Table 9.	Single Step Integration Method; Drag Numerical Results, Unfiltered Input Data . . . . .	70
Table 10.	Single Step Integration Method; Quadratic Numerical Results, Filtered Input Data . . . . .	71
Table 11.	Single Step Integration Method; Quadratic, Cubic Numerical Results, Filtered Input Data . . . . .	72
Table 12.	Single Step Integration Method; Drag Numerical Results, Filtered Input Data . . . . .	73
Table 13.	Multistep Integration Method; Quadratic Numerical Results, Unfiltered Input Data . . . . .	86

Table 14. Multistep Integration Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data .....	87
Table 15. Multistep Integration Method; Drag Numerical Results, Unfiltered Input Data .....	88
Table 16. Multistep Integration Method; Quadratic Numerical Results, Filtered Input Data .....	89
Table 17. Multistep Integration Method; Quadratic, Cubic Numerical Results, Filtered Input Data .....	90
Table 18. Multistep Integration Method; Drag Numerical Results, Filtered Input Data .....	91
Table 19. Two Degrees of Freedom Results; Single Step Method, Unfiltered Data	117
Table 20. Two Degrees of Freedom Results; Single Step Method, Filtered Data	118
Table 21. Two Degrees of Freedom Results, Multiple Step Method, Unfiltered Data .....	125
Table 22. Two Degrees of Freedom Results, Multiple Step Method, Filtered Data	126

# 1.0 Introduction

Space technologies have become increasingly interested in the identification of structural systems because of a need to completely describe the behavior of the systems under general loading conditions and environments. The need for describing structural dynamic systems arises naturally because of ever-increasing costs of materials and construction, as well as other constraints such as weight restrictions. For example, in the construction of space structures, the need for complete description of the systems becomes evident when considering the effects of a gravity free environment, as well as those of determinate and random forces applied to the structure. Without a complete knowledge of the possible responses of the structure, catastrophic failures could occur as a result of loading conditions which may not be reproducible under ordinary laboratory conditions. Thus, by determining the parameters which govern a structural dynamic system under loadings which are determinate and simple in nature, the response of the system may be accurately predicted under more complex loadings and environments. In a previous article dealing with such space structures, Juang and Pappa [1] implement a modal parameter identification algorithm on data obtained from measurements of the response of the Galileo spacecraft.

Identification of structural dynamic systems involves searching for the parameters which govern a given system. The parameters generally appear in the differential equations which describe the systems as coefficients such as mass, stiffness, and damping, as well as nonlinear terms involving displacement and velocity. The objective of an identification algorithm applied to a system is to take measurements of the input and output of the system, either knowing a priori some or none of the system parameters, and to fit a curve to the output of the system. The curve is fitted via making corrections to the parameters of some analytically formulated solution to the differential equation. Once the parameters have been found which minimize some objective function describing the difference between measured and analytical responses, the system is said to be identified.

In a simple linear example, measurements of one or all of output displacements, velocities, and accelerations would be compared to a formulated analytical response, and the analytical response adjusted in such a way as to fit a curve or curves to the measured response. The parameters to be identified might be the natural frequency and the damping ratio. Parameters of the forcing function could also be included in the objective function, and depending on the algorithm used, could possibly be found in the minimization process along with the usual system parameters. For an actual structural system, once measurements are taken and the objective function minimized, the response of the system may then be predicted under more general input forcing functions. An overview of the state of the art of the identification process and the methods used is presented by Natke [2].

Extensive literature exists for cases where linear differential equations may be used to describe both single and multiple degrees of freedom systems. For the case of undamped linear MDOF identification, Gladwell [3] presents a review of the available literature.

A discussion of recent linear and nonlinear identification procedures is presented by Ibanez [4].

In recent years, it has become apparent that linear analyses of vibrating structures are not necessarily accurate to desired levels in some cases, and that nonlinear analyses must be implemented to rectify this problem. However, in most cases the appropriate linear analysis will provide accurate results, without becoming quite as involved as a nonlinear analysis of the same problem. Thus, the use of a simpler linear analysis of a structural system should be employed whenever possible. Nevertheless, cases involving sizeable nonlinearities, such as large space structures and laminated structural elements, are becoming more common. In the case of large space structures, the large area to mass ratios involved lend themselves to nonlinear analyses because of the moderately large deflections involved. Denman [5] presents a control scheme for use on large space structures, after a modal identification algorithm has been implemented. A working model of the systems is also desirable because of the costs involved in the production of space structures. The importance of identification of such structures follows directly from a need to minimize costs in both construction and damage prevention. In the case of laminated structures, the usual linear analysis is not sufficient in many cases, and nonlinear effects must be taken into account. Kapania and Raciti [6] present a nonlinear analysis of unsymmetrically laminated beams.

Gawthrop [7] investigates the nonlinear roll motion of ships for identification purposes. The importance of correct identification of the parameters in this example is obvious because of the nature of the operating conditions encountered in shipping. The forces input to the system are intrinsically indeterminate and correct predictions of the response are essential for the safety of the crew and prevention of damage to the vessel. The

nonlinear nature of the differential equation which describes the motion lends itself to nonlinear analyses in cases where the coefficients for damping and displacement nonlinearities are moderately large. In cases where nonlinear effects are sufficiently small, a linear approximation to the system could be implemented. Thus, a need exists for useable algorithms to find the correct values of the parameters of the nonlinear differential equations governing ship roll motions.

Another example where nonlinear identification algorithms have been used is in the case of aircraft landing gear. The parameters governing a highly nonlinear differential equation are identified in an article by Batill and Bacarro [8]. In this case, nonlinearities in the single degree of freedom differential equation governing an oleo-pneumatic shock strut are incorporated in order to predict strut shock response to runway irregularities. The differential equation describing the behavior depends on five parameters, and numerical as well as experimental data are used in an identification algorithm. The identification algorithm involves relating the variation of the equations in the state variables to the corresponding equations dealing with the initial conditions. The variation of the error function is then made to vanish via changes in the parameters, which are treated as state variables, along with the displacement and velocity of the system.

Crawley and O'Donnell [9] have investigated the nonlinear identification of space joints, by using a force-state mapping of the systems, including the measured quantities displacement, velocity, acceleration and input forces to systems. The noniterative direct method presented by Hanagud, Meyyappa, and Craig [10] also requires the measurement of displacement, velocity, acceleration, and the forcing function to the system for identification.

An algorithm based on the method of multiple scales presented by Nayfeh [11] has been introduced in Reference [10]. The method involves finding an approximate analytical solution for a single degree of freedom system, and correcting the parameters of the solution in order to fit the measured response of the system. The method of multiple scales is used most profitably where coefficients of the nonlinear terms of the differential equations are moderately small, and is increasingly (decreasingly) accurate for decreasing (increasing) nonlinear term coefficients. The method is compared to the iterative direct and noniterative direct methods of identification on the basis of the accuracy of the final parameter estimates. It is found that the identification technique using the method of multiple scales is more accurate, and involves less computational time than the iterative direct and noniterative direct methods. Also, the method only requires the measurement of one quantity of displacement, velocity, or acceleration, whereas the noniterative direct method requires the measurement of all three quantities. The methods are applied to a single degree of freedom system with a cubic nonlinearity, subjected to an impulse forcing function at  $t = 0$ .

In this study, the iterative direct method is applied to three other examples, and compared to results obtained using a new method. The iterative direct method is found to be analytically simple, and may be applied to a large class of nonlinear single degree of freedom systems. The analytical solution to be fitted to measured data is found numerically, leading to simplified algebra prior to programming the method. Also, the method requires only one of the quantities of displacement, velocity, and acceleration to be measured.

The method of multiple scales has not been included here for the sake of simplicity, because of difficulties in minimization, once the approximate solution of the system has



been obtained. When initial conditions are incorporated in the solution of a system, difficulties arise in differentiating the solutions to some systems, since the equations describing the initial conditions are often differential equations or complex algebraic equations. It is found that these problems may be circumvented using either the iterative direct method or the method presented here. The method of multiple scales was found to be an excellent choice for the simple case presented in Reference [10], but extension of the method to more complicated systems is not necessarily straightforward. Similarly, the noniterative direct method has not been treated here, since the method requires the measurement of the four quantities displacement, velocity, acceleration, and the input forcing function, which may not all be readily measurable, especially for large degrees of freedom systems.

The principal aim of this study was to examine the effectiveness of two identification procedures for simple single degree of freedom systems, and their application to multiple degrees of freedom systems. Therefore, some assumptions have been made in the analyses of the systems. One of the assumptions throughout the analyses that follow is that no time varying forcing functions are applied to the systems. The initial conditions used are simply initial velocities in the case of single degree of freedom systems, and initial displacements in the case of the two degrees of freedom example. An impulse forcing function centered at  $t=0$  has been used in all the single degree of freedom examples, which may be viewed as an initial velocity with no initial displacement. More details of this matter will be introduced in the second chapter. The assumption of initial displacements in two degrees of freedom systems is made for the sake of simplicity, since no attempt is made to identify the forcing function parameters in the two degrees of freedom example.

The input values of motion in all the examples treated here are generated synthetically by solving the governing equations for a given set of parameters. The inputs to the algorithms in these cases are only corrupted by the truncation and roundoff errors due to numerical solution of the equations. Therefore, in order to attempt to simulate measurements which might actually be taken, white noise has been added to the pure "measured" responses of the systems as a percentage of the RMS values of the total number of points recorded in each example. The final values for the parameter estimates are tabulated using the pure and corrupted input values, and a filter is also implemented in order to attempt to reduce the amount of noise introduced. The filter used is of the low pass variety, and has a frequency response curve with a cutoff frequency slightly above the frequency seen in spectral estimates as a peak, and tapers off gradually to zero. Details of the filter will be discussed in Chapter 2.

A principal motivation for this study is the need for reasonably simple procedures for the identification of single degree of freedom examples which may be extended to more general multiple degrees of freedom systems and forcing functions. A comparative analysis of three existing identification algorithms has been presented in Reference [10]. Much of the work performed in this study is a direct extension of this article, primarily in the use of the iterative direct method of the second chapter. The method is applied to three single degree of freedom examples in Chapter 2. The examples include quadratic and cubic nonlinearities, and a drag nonlinearity encountered in cases involving vibrations in water, such as ship roll motion and offshore structures. In Chapter 3, a new objective function is formulated which simplifies numerical and analytical computations. The objective function is found by utilization of approximate solutions to the governing differential equations. Once the objective function is formulated, it is found that differentiation is simplified greatly over previous methods. In Chapter 4, an extension of the

method presented in Chapter 3 is introduced and applied to the same three examples as the earlier two methods. In Chapters 5, 6 and 7, the methods of Chapters 2, 3, and 4 are extended for use on a simple two degrees of freedom example. In the Appendix, the computer program used in the two degrees of freedom example is presented, with appropriate documentation.

## 2.0 Iterative Direct Method : Single Degree of Freedom Systems

The iterative direct method (described briefly in the following) can be applied directly to a number of single degree of freedom examples. Distefano and Rath [12] apply the method to a single degree of freedom system with cubic nonlinearities in displacement and velocity. The method will be applied here to three types of nonlinear single degree of freedom examples, and the associated equations developed. Later, the iterative direct method will be compared to two other methods developed during this study, using the same three examples.

The usual ordinary differential equation which describes a linear structural dynamic system with one degree of freedom is

$$m\ddot{u} + c\dot{u} + ku = f(t) \quad [2.1]$$

where  $u$  is the displacement at the node under consideration, dots are derivatives with respect to time, and  $m$  is the mass,  $c$  is the damping coefficient, and  $k$  is the stiffness of

the system. On the right of the equation is the input forcing function,  $f(t)$ . If the equation is divided by  $m$ , it takes the form

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u = F(t) \quad [2.2]$$

where  $\gamma$  is defined as  $c/2m$ ,  $\Omega$  is the natural frequency of the system, and  $F(t)$  is  $f(t)/m$ . This equation depends on the coefficients  $\gamma$  and  $\Omega^2$ . The forcing function will also in general depend on parameters such as the amplitude and the phase angle of the forcing function for the case of sinusoidal inputs, and the amplitude in the case of an impulse. The objective of an identification algorithm is to measure the response of an actual system described by this equation and to fit an analytical solution of equation [2.2] to the measured response via adjustments to an initial set of the parameters  $\gamma$  and  $\Omega^2$ , which are contained in the vector  $\{X\}$ .

A nonlinear single degree of freedom structural dynamic system is described by the equation

$$\ddot{u} = f(u, \dot{u}; x_1, x_2, \dots, x_n) \quad [2.3]$$

where  $u$  is the displacement of the system and  $x_i$  is the  $i$ th system parameter describing a system physical property such as geometry, mass, stiffness, or damping. The function  $f(\ )$  may also include the input forcing function, analogous to the linear case mentioned above. The function  $f(\ )$  is nonlinear in nature, meaning at least one term in  $f(\ )$  preceded by an  $x_i$  is itself nonlinear in  $u$  or  $\dot{u}$ . In the identification of an actual system, the correct values of the  $x_i$ 's are sought in order to completely describe the system. In the algorithms presented here, it is assumed that the form of the function  $f(\ )$  is known, while the coefficients of the function are partially or completely unknown. Thus, the name parametric identification. It is also assumed that the coefficients are constant with re-

spect to time. As in the linear single degree of freedom case, given an actual structural system described by the nonlinear form of the differential equation and the form of  $f(t)$ , measurements are taken of the response of the system under a determinate or indeterminate dynamic force, and an analytical solution of equation [2.3] is fitted to the measured response via corrections in the parameters  $x_i$ .

As a first step, an objective function must be formulated describing a residual between the measured response of a given system and the analytical response of the formulated model. Therefore, some model must be developed which can be shown to describe the response of the system, once given the parameters of the model. So, a differential equation must be chosen to describe the system, and the solution of the differential equation must be fitted to the actual measured response of the system.

## ***2.1 Objective Function Formulation***

The objective function used here and presented in Reference [10] is of the form

$$L\{X\} = \int_0^{T^*} (u_a - u_m)^2 dt \quad [2.4]$$

where  $u_a$  is the analytical response, which may be displacement, velocity, or acceleration, and  $u_m$  is the measured quantity corresponding to the values of  $u_a$ .  $T^*$  is the record length of the measurements taken. The values for  $u_m$  are in the form of a time series of equally spaced points.  $L\{X\}$  is the value of the objective function for a given vector of parameter values  $\{X\}$ .  $u_a$  is found as a time series of the solution of the differential

equation which describes the system under consideration.  $\{X\}$  is the vector of parameters being sought for the system, the principal object of the identification process. Given an initial guess for the values of  $\{X\}$ , the differential equation is solved, and the integral is approximated, giving a value for the area of the squared difference between the measured and analytical responses. The values for  $u$ , must be found by solving a differential equation of the form

$$\ddot{u} = f(u, \dot{u}, x_1, x_2, \dots, x_n) \quad [2.5]$$

In order to minimize  $L\{X\}$ , the correct values for  $\{X\}$  must be found. In general, this minimization requires that derivatives of  $L\{X\}$  be calculated with respect to the various  $x_i$ 's. These derivatives are calculated and the objective function is minimized using the iterative direct method, which is discussed in the following sections.

## ***2.2 Objective Function Differentiation***

The derivatives needed to minimize  $L\{X\}$  will be at least first order in nature, and will be needed in the form of time series of points in order to utilize a quadrature formula for the required integral approximations. A quadrature formula will be used to approximate the integrals found in the expressions for the derivatives of the objective function of equation [2.4]. The time series  $u_m$  does not depend on  $\{X\}$ . Therefore, the first derivative of  $L\{X\}$  with respect to  $x_i$  takes the form

$$\frac{\partial L\{X\}}{\partial x_i} = \int_0^{T^*} 2(u_a - u_m) \frac{\partial u_a}{\partial x_i} dt \quad [2.6]$$

In order to find the partial derivative of  $u_a$  with respect to  $x_i$ , the governing differential equation is differentiated with respect to  $x_i$  for all  $i$ , and transformed to a system of first order differential equations. The resulting system is solved to find the displacement or velocity  $u_a$  as well as the first partial derivatives of the quantity. When differentiating the equation, it must be noted that the displacement, velocity, and acceleration are all dependent on  $\{X\}$ . The way in which the differentiation is performed will be covered in a later section. Once the proper system of equations is set up and solved over the interval record length, the time series for  $u_a$  and all its partial derivatives with respect to the  $x_i$ 's are used in an optimization algorithm to minimize the objective function  $L\{X\}$ .

### ***2.3 Objective Function Minimization***

The method used to minimize  $L\{X\}$  with respect to the vector of parameters  $\{X\}$  is presented by Levenberg [13] and Marquardt [14]. The method involves only first partial derivatives of  $u_a$  with respect to the parameters  $x_i$ , and is of the single step update type, with iterations on  $\{X\}$ . At a point when iterations fail to change each component of  $\{X\}$  by a percentage of the previous values of  $\{X\}$ , iterations are terminated. One step methods take the general form

$$\{X\}^{l+1} = \{X\}^l + h^l \{\Delta X\} \quad [2.7]$$



where  $\{\Delta X\}$  is the correction vector, and  $h^i$  is the stepsize.  $\{X\}^{i+1}$  is the most recent value for the vector  $\{X\}$ , and what remains to be found is the correction vector. In the classical Newton-Raphson method, the stepsize  $h^i$  is set equal to one, and the correction vector is found as follows. The direction in which the steepest descent in the value of  $L$  takes place is the negative gradient of the function with respect to the parameters  $\{X\}$ . Therefore, in order to minimize the function  $L$ , steps in  $\{X\}$  are taken in the direction of the vector  $\{g\} = - \left[ \frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \dots, \frac{\partial L}{\partial x_n} \right]^T$  for an  $n$ -component vector of parameters. The distance needed to travel in the negative gradient direction is found to be the inverse of the second derivative of  $L$  with respect to the parameters, or inverse Hessian of the function: a matrix with components of the form

$$\frac{\partial}{\partial x_k} \left( \frac{\partial L}{\partial x_l} \right) \quad [2.8]$$

Therefore, the correction vector  $\{\Delta X\}$  takes the form  $- [H]^{-1}\{g\}$  where the Hessian and gradient are as specified above. Rather than inverting the Hessian matrix in order to find the correction vector as above, it is possible to solve the linear algebraic nonhomogeneous system of  $n$  equations in  $n$  unknowns

$$[H]\{\Delta X\} = - \{g\} \quad [2.9]$$

At each iteration, the gradient and Hessian are calculated, the system of equations [2.9] is solved, and a new vector  $\{X\}^{i+1}$  is found from equation [2.7]. The iterations are terminated when a specified convergence criterion is met.

For the objective function above, the gradient vector is computed using the first derivatives with respect to the parameters sought. Since a set of first order differential equations has been defined for this purpose, no problem is encountered in finding these

quantities. Because the differential equations for each example are problem dependent, the details of these equation formulations will be deferred to later sections.

The gradient for the objective function defined above has components of the form

$$g_k = 2 \int_0^{T^*} \left( \frac{\partial u_a}{\partial x_k} \right) (u_a - u_m) dt \quad [2.10]$$

A quadrature formula is used to approximate the value of the integral, given the measured time series  $u_m$  and a time series for the calculated quantity  $u_a$ . The time series for the first partial derivatives are found in the simultaneous solution of the first order differential equations defined for this purpose.

The Hessian matrix for the objective function above must be approximated not only via a numerical solution of the first order differential equations and integration of the Hessian components, but also in the formulation of the integrand of each component. The exact expression for the  $kl$  component of the Hessian is given as

$$H_{kl} = 2 \int_0^{T^*} \left\{ \left( \frac{\partial u_a}{\partial x_k} \right) \left( \frac{\partial u_a}{\partial x_l} \right) + \frac{\partial}{\partial x_k} \left( \frac{\partial u_a}{\partial x_l} \right) (u_a - u_m) \right\} dt \quad [2.11]$$

Unfortunately, a time series for the second derivative term may not be calculated at the same time as the first derivative terms, since doing so would require large computational costs involved in numerically solving a very large system of differential equations. In order to circumvent this problem, the Levenberg-Marquardt method is used [13, 14]. In

this method, the second derivative term is not used in the expression for the Hessian. Instead of solving the usual set of linear algebraic equations in the Newton-Raphson method, the following set of linear algebraic equations is solved for the correction vector:

$$[\{N\}^\star + \lambda[\Gamma]]\{\Delta X\}^\star = \{g\}^\star \quad [2.12]$$

where

$$g_k^{i\star} = \frac{g_k^i}{\sqrt{N_{kk}^i}}$$

$$N_{kl}^{i\star} = \frac{N_{kl}^i}{\sqrt{N_{kk}^i N_{ll}^i}}$$

$$\Delta x_k^{i\star} = \Delta x_k^i \sqrt{N_{kk}^i} \quad [2.13]$$

$$g_k^i = -2 \int_0^{T^\star} \left( \frac{\partial u_a^i}{\partial x_k} \right) (u_m - u_a^i) dt$$

$$N_{kl}^i = 2 \int_0^{T^\star} \left( \frac{\partial u_a^i}{\partial x_k} \right) \left( \frac{\partial u_a^i}{\partial x_l} \right) dt$$

$k$  is the general component of the gradient and  $\{X\}$  vectors, and  $l$  is the column of the Hessian matrix.  $\lambda$  is a scaling factor, chosen to increase the size of the correction vector components if the objective function value has been found to decrease in the preceding

step. The initial value for  $\lambda$  is 0.01, and the value is multiplied (divided) by ten if the objective function has increased (decreased) in the past step in  $\{X\}$ . The stars in equations [2.12] and [2.13] denote scaled versions of the Hessian, gradient, and correction vector. The scaling is done in order to aid in the solution of the system of algebraic equations in cases where equations [2.12] are ill conditioned. After the system is solved for  $\{\Delta X\}^*$ , each component of the scaled version of the correction vector is scaled back by dividing by  $\sqrt{N_{kk}^i}$ . The step is then taken in  $\{X\}$  and the new values of the parameters are used to solve the set of first order differential equations for the values of  $u_e$  and its first partial derivatives. The gradient and Hessian are again calculated, and the system of algebraic equations [2.12] is solved. The iteration process continues until the value of each parameter fails to change sufficiently as a fraction of its previous value.

## ***2.4 Treatment of Noise***

In order to simulate the noise effects intrinsic to the measurement process for finding the time series  $u_m$ , random amounts have been added to the input quantities to the procedure. The input quantity  $u_m$  has been generated synthetically by solving the governing differential equation [2.5] for an assumed set of system parameters. Filtering is then used on the input "measured" data in order to attempt to improve on the final parameter estimates obtained using corrupted input data. The filtering is performed using a low-pass digital filter subroutine supplied in the BMD software package [15] called BMD03T. The filter is a low pass frequency filter, designed to pass data at frequencies up to a value slightly above the system natural frequency. To simulate white noise effects before filtering, random numbers are generated using the IMSL subroutine DRNUN [16]. The

subroutine generates random values between 0 and 1, which are scaled to percentage values, such as [-20%,20%]. The percentage values are multiplied by the RMS value found for each set of "measured" input points, and the resulting random amounts are added to the pure generated data from the solution of the original differential equation, for a set of chosen parameter values.

The system natural frequency may be estimated via a spectral density analysis subroutine, such as in the IMSL subroutine DSSWD [16]. A typical plot of the spectral density of a set of input data is presented in Figure 1. Note that the peak of the natural frequency of the data shows up at nearly the value substituted into the differential equations initially (5 rad/sec), later to be found using the minimization procedure. The peak frequency also gives an excellent initial estimate of the natural frequency for use in the procedure. The frequency of the noise added to the system extends past the frequencies analyzed in the spectral analysis, as expected.

In order to use the filter mentioned above, the number of data points must be doubled to account for decimation, or halving of the number of data points filtered. The filter operates by estimating each data point via a linear combination of surrounding data values, with weights dependent on a sinusoidal function of the point numbers under consideration, the half length of the filter, and the number of data points analyzed. The half length of the filter is simply half the number of points included in the linear combination for the value of the filtered data point under consideration. The subroutine returns the filtered data, as well as a spectral analysis and cross spectral analysis for cases where more than one data series is analyzed. The frequency response of the filter is plotted in Figure 2. Time series analysis and digital filters are presented by Bloomfield [16] and Stearns [17].

In Figure 3, the effects of filtering on a typical set of input data are illustrated. In general, peaks found in the corrupted data are reduced, with the most effective reductions of noise appearing to take place early along the time record length. Similar results of filtering of input data were obtained for the other examples and cases. In Figure 1, an estimate of the spectral density of a representative case is plotted for the corrupted and filtered data, as well as the response of the system using the estimated parameters. The smooth curve for the identified response is due to the fact that the only noise introduced to its analyzed time series is due to numerical noise introduced in the solution of the differential equation, for the estimated parameter values.

Although the input data of Figure 3 appears to be only marginally cleaner after filtering, the filter used is in most cases effective in that the parameter estimates found using filtered input quantities are improved over those found without filtering the data. More sophisticated digital filtering algorithms, such as presented in References [16] and [17], would probably improve the final parameter estimates to a larger degree.

## ***2.5 Applications of The Iterative Direct Method***

The iterative direct method has been applied to three nonlinear examples, chosen to illustrate the way in which the problems are set up and solved. In the first example, a simple quadratic nonlinearity is introduced. In the second example, a cubic nonlinearity is added to the previous quadratic nonlinear equation, and the resulting system is identified. In the third example, a drag nonlinearity is introduced to the original linear equation [2.2], and the resulting system is set up and identified using the iterative direct

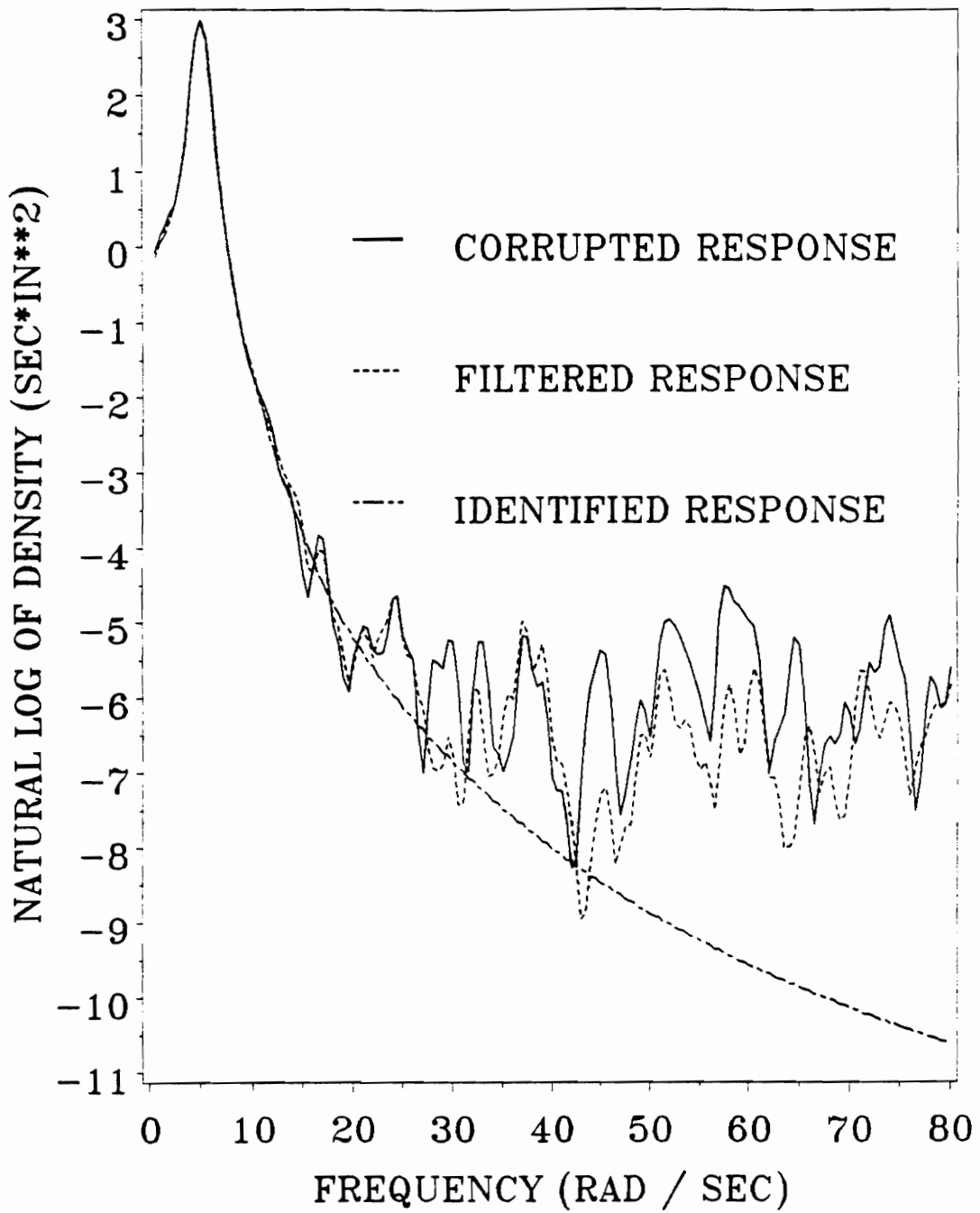


Figure 1. Spectral Density of Quadratic and Cubic Nonlinearity Example: 20% corrupted input data,  $\alpha_1 = 2.5$ ,  $\alpha_2 = 5.0$

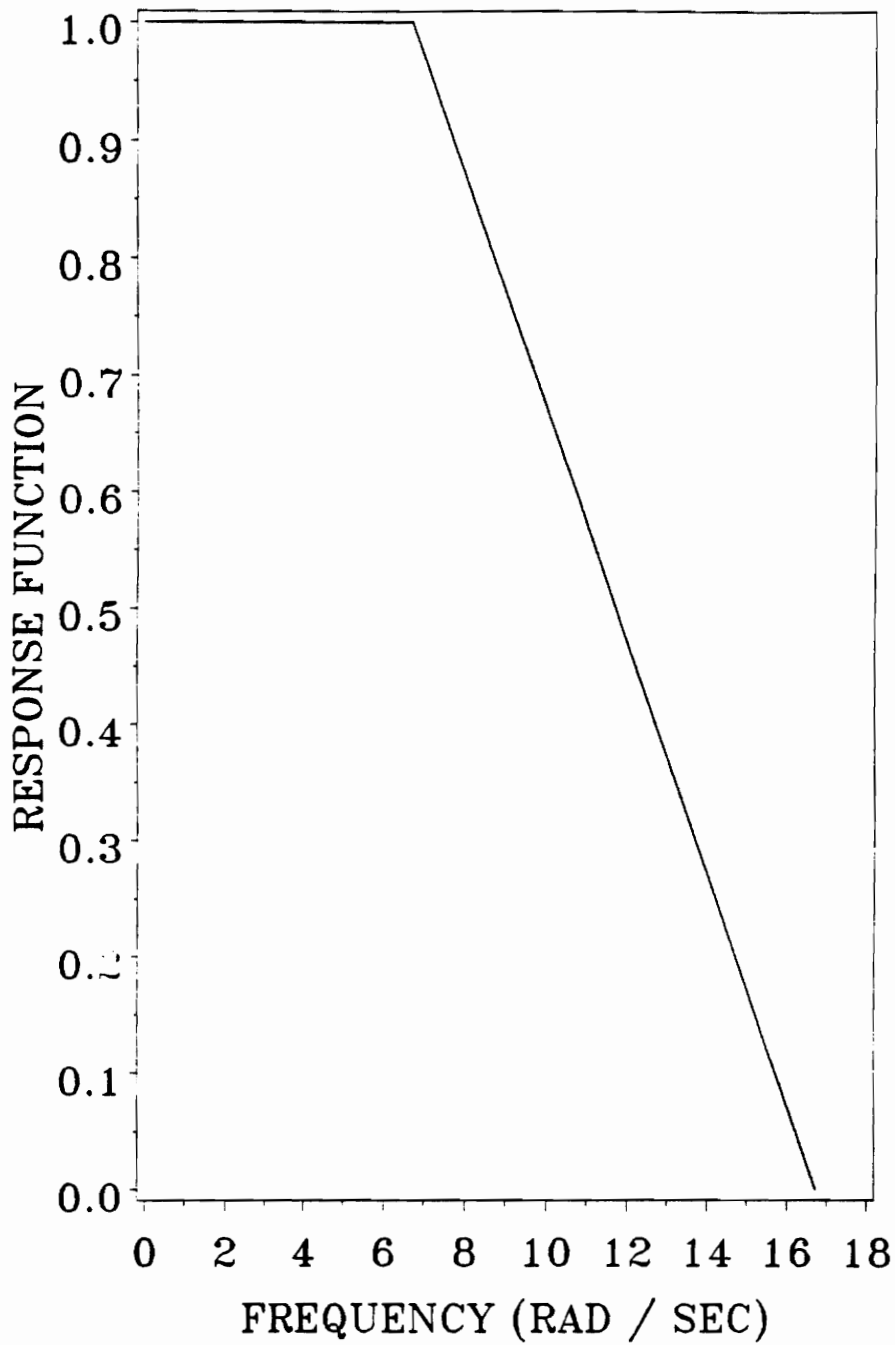


Figure 2. Frequency Response of Low Pass Filter



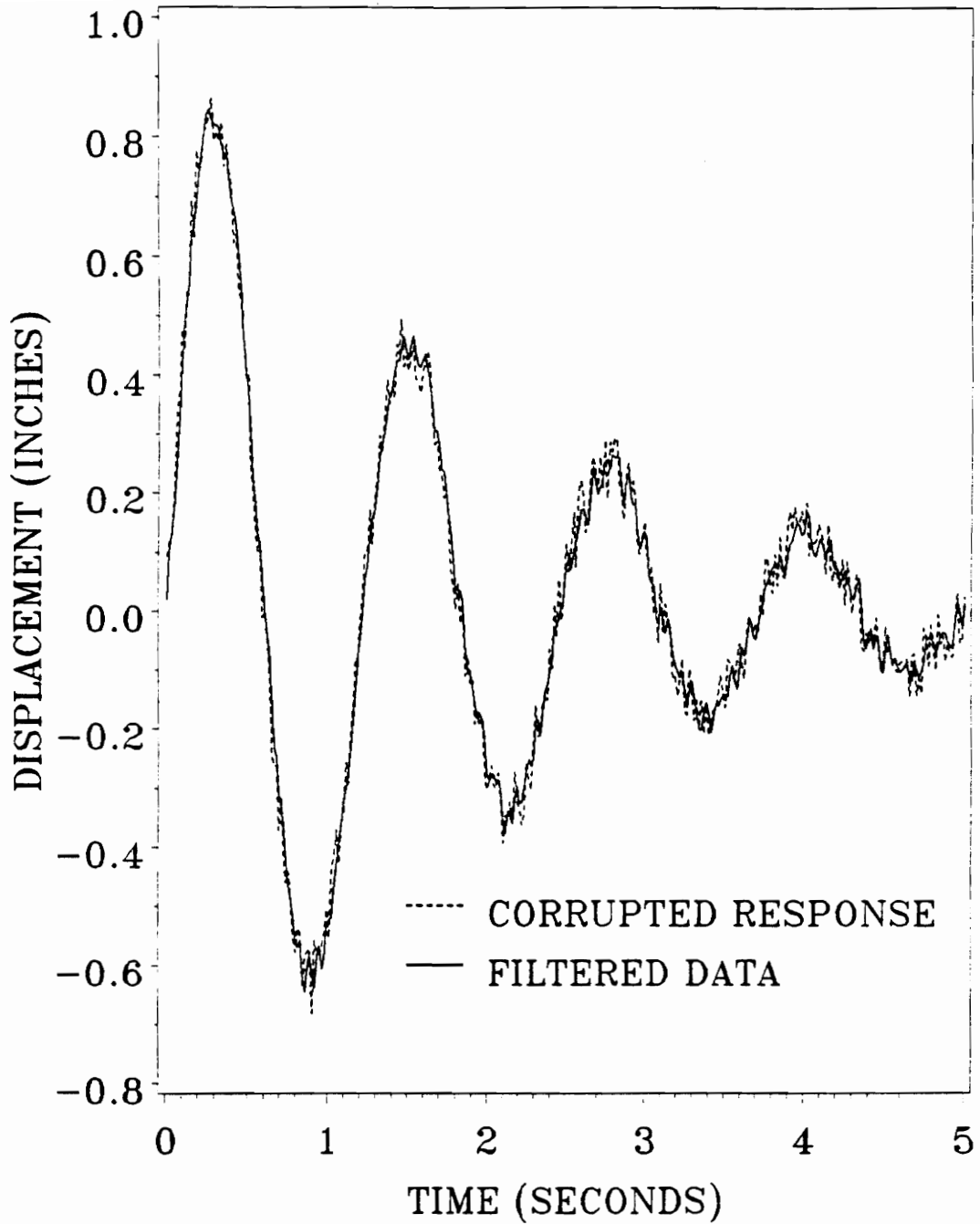


Figure 3. Effect of Filtering of Input Displacement Data: Quadratic and Cubic Nonlinearity  
 Example, 20% corrupted input data,  $\alpha_1 = 2.5$ ,  $\alpha_2 = 5.0$

method. In all the three cases, the measured input to the algorithm is taken as the numerical approximation to the solution of the nonlinear equation, for chosen parameter values. In effect, the solutions to the examples are known prior to calculations in order to evaluate the method's effectiveness.

### 2.5.1 Quadratic Nonlinearity Example

The equation governing the quadratic nonlinearity case is

$$m\ddot{u} + c\dot{u} + ku + \beta u^2 = f(t) \quad [2.14]$$

The forcing function used in the examples that follow is an impulse function of the form  $Q\delta(t)$  where  $Q$  is the size of the impulse, which is centered at  $t=0$ . If the equation is divided by  $m$ , equation [2.14] takes the form

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0 \quad [2.15]$$

with initial conditions  $u(t=0^+) = 0$ , and  $\dot{u}(t=0^+) = Q$ . The impulse may be treated as an initial condition by energy arguments, which are presented by Craig [18]. The parameters to be found are  $\gamma, \Omega^2, \alpha$ , and  $Q$ . For the iterative direct method, it is possible to find the initial conditions of the system as well as the usual parameters. Essentially, this problem is a free vibration example, treating the initial conditions as unknown parameters as well.

The values used in the differential equation for the parameters are arbitrary, but are representative values of an actual system. Once the parameter values are chosen, the

differential equation [2.15] is solved as a first order system of two coupled nonlinear differential equations using the IMSL subroutine DIVPRK [19]. The subroutine uses a Runge-Kutta-Verner fifth and sixth order predictor-corrector pair. Predictor-corrector methods are treated extensively by Stoer and Bulirsh [20]. The output from the subroutine is a time series of displacements or velocities. The quantity used here is the displacement, which is used in the minimization algorithm for  $u_e$ .

Once the differential equation has been solved in order to find input quantities (displacement) for the algorithm, an analytical model ( $u_e$ ) must be formulated which describes the "measured" time series. For the parametric algorithm used here, it is assumed that the form of the equation describing the system is known a priori, so that the differential equation above is used as a model of the behavior of the "measured" time series.

First, the following quantities must be defined:

$$x_1 = \gamma$$

$$x_2 = \Omega^2$$

$$x_3 = \alpha$$

$$x_4 = Q$$

$$y_1 = u$$

$$y_2 = u_{,t}$$

$$y_3 = u_{,x_1}$$

$$y_4 = (u, x_1)_{,t} \quad [2.16]$$

$$y_5 = u, x_2$$

$$y_6 = (u, x_2)_{,t}$$

$$y_7 = u, x_3$$

$$y_8 = (u, x_3)_{,t}$$

$$y_9 = u, x_4$$

$$y_{10} = (u, x_4)_{,t}$$

The acceleration term of the governing differential equation [2.15] is now isolated, and derivatives are taken with respect to each of the four  $x_i$ 's. Note that the velocity  $\dot{u}$  and displacement  $u$  depend on all of the parameters. Using the quantities above, the following set of differential equations is formed:

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -2x_1y_2 - x_2y_1 - x_3y_1^2$$

$$\dot{y}_3 = y_4$$

$$\dot{y}_4 = -2y_2 - 2x_1y_4 - x_2y_3 - 2x_3y_1y_3$$

$$\dot{y}_5 = y_6 \quad [2.17]$$

$$\dot{y}_6 = -2x_1y_6 - y_1 - x_2y_5 - 2x_3y_1y_5$$

$$\dot{y}_7 = y_8$$

$$\dot{y}_8 = -2x_1y_8 - x_2y_7 - y_1^2 - 2x_3y_1y_7$$

$$\dot{y}_9 = y_{10}$$

$$\dot{y}_{10} = -2x_1y_{10} - x_2y_9 - 2x_3y_1y_9$$

with the following initial conditions:

$$y_1 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = 0 \quad [2.18]$$

$$y_2 = x_4$$

$$y_{10} = 1.0$$

In the last initial condition,  $y_{10}$  is the derivative of the initial velocity with respect to  $Q$ , so since the initial velocity is  $Q$ , the derivative value is 1. For the solution of the above equations, the differential equation solution subroutine used earlier for generating input values for  $u_m$  is again used. The solution of the system of equations [2.17] is in the form of time series of the displacement and derivatives of the displacement with respect to the parameters. The time series are to be used in the objective function, gradient, and Hessian calculations of equations [2.12] and [2.13]. Simpson's rule for integration is used to approximate the integrals in the objective function, gradient, and Hessian expressions. In these calculations, the interval 0 to  $T^*$  is divided into evenly spaced intervals, and the integrals approximated by summing up approximate integral values for each group of three points. No attempt is made to use a step size control, as is used in many other advanced integral approximations, since the time series in  $u$  and its derivatives are found

as time series of equally spaced points. A quadrature method of higher order could be implemented as well, but for the sake of simplicity, the third order method is used. The differential operators used on the integrals in the gradient and Hessian expressions may be placed inside the integrals, since the integrals are approximated by linear combinations of the integrands and their derivative values.

Once the gradient, Hessian, and objective function have been approximated with the quadrature formula, the system of four linear algebraic equations [2.12] must be solved in order to find the correction vector  $\{\Delta X\}$ . The system of equations is solved using the subroutine GAUSS, presented by Johnson and Riess [21]. The subroutine uses Gaussian elimination to solve linear systems of the form  $AX = B$ , where  $A$  is the coefficient matrix of the equations and  $B$  is the constant vector. For this example, where four parameters are sought,  $A$  is the four by four Hessian matrix and  $B$  is the four component gradient vector. Once the correction vector is solved for, the new vector  $\{X\}^{i+1}$  is found from equation [2.7]. The objective function is then calculated using the new values of the parameters, and if the objective function has been found to decrease, the new value for  $\{X\}$  is retained, and the process repeated.

The correction vector is also checked for sufficient size by checking the absolute value of the ratio of the correction components to the previous values of the vector components. If the ratio is found to be less than  $5.00E-4$  for all the components, the iteration process is terminated. If the objective function value is found to increase, the process of finding a different  $h'$  begins. The new value for  $h'$  is found by making calculations presented by Bard [22]. The objective function is approximated using a quadratic polynomial, and the minimum of the quadratic approximation is taken to be the position of the most desirable step size in the parameter vector  $\{X\}$ . The polynomial approxi-

mation of the objective function  $L\{X\}$  takes the form  $a + bh + ch^2$ , where  $a = L(\{X\}^i)$ ,  $b = -\{g\}^T[R]^i\{g\}^i$  with  $[R]^i = [[N]_{ii}^i + \lambda^i[I]]^{-1}$  and  $c = (L(\{X\}^{i+1}) - L(\{X\}^i) - bh^i)/h^2$ . Therefore, the step size which minimizes the quadratic approximation to the objective function is  $h^{\star i} = -b/2c$ .

Rarely is the objective function found to increase for these examples, so this branch is used only occasionally. In fact, the branch may be ignored, and the same results obtained, since the change in the objective function and parameters when the alternate branch is taken is found to be relatively small.

## 2.5.2 Quadratic and Cubic Nonlinearity Example

This example is governed by an equation of the form found in the description of unsymmetrically laminated composite materials (see Reference [6]) and the nonlinear vibrations of shells and imperfect plates. The differential equation takes the form

$$m\ddot{u} + c\dot{u} + ku + \beta_1 u^2 + \beta_2 u^3 = f(t) \quad [2.19]$$

The forcing function  $f(t)$  is the same for this example as in the preceding example. After dividing the equation by  $m$ , the equation takes the form

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0 \quad [2.20]$$

with initial conditions  $u(t = 0^+) = 0$  and  $\dot{u}(t = 0^+) = Q$ , where  $Q$  is the size of an impulse centered at  $t = 0$ , as in the quadratic nonlinearity example. The parameters to be found are  $\gamma, \Omega^2, \alpha_1, \alpha_2$ , and  $Q$ . The formulation of the problem is exactly as in the quadratic

nonlinearity example, with the exception that two new differential equations are formulated in order to find the derivative of  $u$ , with respect to the fourth parameter  $\alpha_2$ .

The following quantities must be defined:

$$x_1 = \gamma$$

$$x_2 = \Omega^2$$

$$x_3 = \alpha_1$$

$$x_4 = \alpha_2$$

$$x_5 = Q$$

$$y_1 = u$$

$$y_2 = u_{,t}$$

$$y_3 = u_{,x_1}$$

$$y_4 = (u_{,x_1})_{,t} \quad [2.21]$$

$$y_5 = u_{,x_2}$$

$$y_6 = (u_{,x_2})_{,t}$$

$$y_7 = u_{,x_3}$$

$$y_8 = (u_{,x_3})_{,t}$$



$$y_9 = u, x_4$$

$$y_{10} = (u, x_4), t$$

$$y_{11} = u, x_5$$

$$y_{12} = (u, x_5), t$$

As in the quadratic nonlinearity example, the following system of equations must be solved to find the displacement and its derivatives with respect to the parameters along the interval of time:

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -2x_1y_2 - x_2y_1 - x_3y_1^2 - x_4y_1^3$$

$$\dot{y}_3 = y_4$$

$$\dot{y}_4 = -2y_2 - 2x_1y_4 - x_2y_3 - 2x_3y_1y_3 - 3x_4y_1^2y_3$$

$$\dot{y}_5 = y_6$$

[2.22]

$$\dot{y}_6 = -2x_1y_6 - y_1 - x_2y_5 - 2x_3y_1y_5 - 3x_4y_1^2y_5$$

$$\dot{y}_7 = y_8$$

$$\dot{y}_8 = -2x_1y_8 - x_2y_7 - y_1^2 - 2x_3y_1y_7 - 3x_4y_1^2y_7$$

$$\dot{y}_9 = y_{10}$$

$$\dot{y}_{10} = -2x_1y_{10} - x_2y_9 - 2x_3y_1y_9 - y_1^3 - 3x_4y_1^2y_9$$

$$\dot{y}_{11} = y_{12}$$

$$\dot{y}_{12} = -2x_1y_{12} - x_2y_{11} - 2x_3y_1y_{11} - 3x_4y_1^2y_{11}$$

with the following initial conditions:

$$y_1 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = y_{11} = 0$$

$$y_2 = x_5 \tag{2.23}$$

$$y_{12} = 1.0$$

where the derivatives of the displacement and velocity with respect to the fifth parameter have initial conditions as specified in the quadratic nonlinearity example. The iterative process is exactly as in the quadratic case, but with a five by five Hessian, and five component gradient and parameter vectors. Again, the process is terminated when the change in each vector component is less than 5.0E-4 of the previous value.

### 2.5.3 Drag Nonlinearity Example

This example contains a drag term in its differential equation found in many equations describing submersed structures, such as ship roll motion (see Reference [7]) and offshore structures, studied by Wu, Wang, and Price [23]. The equation is of the form

$$m\ddot{u} + c\dot{u} + ku + \beta\dot{u}|\dot{u}| = f(t) \tag{2.24}$$

Again, the forcing function  $f(t)$  is the same for this example as in the preceding examples.

After dividing by  $m$ , the equation takes the form

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0 \quad [2.25]$$

with initial conditions  $u(t = 0^+) = 0$  and  $\dot{u}(t = 0^+) = Q$ . The formulation of the differential equations is the same as the above examples, with the exception of the derivatives of the absolute value of the velocity. As in the above examples, the following quantities are defined:

$$x_1 = \gamma$$

$$x_2 = \Omega^2$$

$$x_3 = \alpha$$

$$x_4 = Q$$

$$y_1 = u$$

$$y_2 = u_{,t}$$

$$y_3 = u_{,x_1}$$

$$y_4 = (u_{,x_1})_{,t} \quad [2.26]$$

$$y_5 = u_{,x_2}$$

$$y_6 = (u_{,x_2})_{,t}$$

$$y_7 = u_{,x_3}$$

$$y_8 = (u_{,x_3})_{,t}$$

$$y_9 = u_{,x_4}$$

$$y_{10} = (u_{,x_4})_{,t}$$

The system of equations to be solved for the displacement and the associated derivatives takes the form

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -2x_1y_2 - x_2y_1 - x_3y_2|y_2|$$

$$\dot{y}_3 = y_4$$

$$\dot{y}_4 = -2y_2 - 2x_1y_4 - x_2y_3 - x_3y_4|y_2| - (\text{sign})x_3y_2|y_4|$$

$$\dot{y}_5 = y_6$$

[2.27]

$$\dot{y}_6 = -2x_1y_6 - y_1 - x_2y_5 - x_3y_6|y_2| - (\text{sign})x_3y_2|y_6|$$

$$\dot{y}_7 = y_8$$

$$\dot{y}_8 = -2x_1y_8 - x_2y_7 - y_2|y_2| - x_3y_8|y_2| - (\text{sign})x_3y_2|y_8|$$

$$\dot{y}_9 = y_{10}$$

$$\dot{y}_{10} = -2x_1y_{10} - x_2y_9 - x_3y_{10}|y_2| - (\text{sign})x_3y_2|y_{10}|$$

with the following initial conditions:

$$y_1 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = 0$$

$$y_2 = x_4 \tag{2.28}$$

$$y_{10} = 1.0$$

where “sign” is negative when the quantity  $y_2$  is negative. The derivative of the absolute value of the velocity is undefined at zero unless the value of the derivative is zero. In practice, it is assumed that the value of the velocity is never exactly zero for the points measured, and this assumption never seems to cause problems in the computations. Also, no value has been defined in the equations for this problem analogous to equations [2.16] for the derivative of the absolute value of the velocity. Therefore, it is necessary to check the sign of the velocity before differentiating, and to multiply by the appropriate sign. The Hessian for this example is four by four, and the gradient and parameter vectors each have four components. The same convergence criterion is used for this case as in the preceding cases, with an overall tolerance of 5.0E-4

## ***2.6 Results***

The numerical results of the application of the iterative direct method to the three non-linear identification examples above are presented in Tables 1 through 6. The computations performed were all in double precision, and the numbers presented have been rounded to two decimal places. As mentioned above, the record length for all the examples was five seconds, divided into 335 points along which to perform integrations

both for the generation of input data, and for the integrations required in the objective function, gradient, and Hessian calculations. The initial values of the parameters and the chosen parameter values used to generate input data are arbitrary, but chosen to attempt to simulate initial guesses and actual measurements taken for real structural systems. In an actual structural system identification study, initial guesses for the parameters could be generated by examination of static test results and spectral analyses of the measured data. For all of the examples presented, records were kept of final parameter estimates, the number of iterations performed before the convergence criterion was met, and the amount of CPU time required to perform the computations. For all the cases studied, the exact parameter values were obtained when pure data was used in the minimization algorithm. The number of iterations for all the examples was six or seven, requiring six to seven seconds of CPU time. All computations were performed on an IBM 3084 mainframe computer, using the VM/SP HPO Release 4.2 operating system.

The final results for the quadratic nonlinearity example are the most accurate of the three single degree of freedom examples studied. For corrupted unfiltered "measured" input data, the method produced final parameter estimates within 28% in all cases. The RMS value of the pure generated input displacement was approximately 0.31 for all the cases, which was the base amount by which the percentage noise amounts were multiplied. The RMS value was simply computed from the total 335 input displacement points generated. The results for this example were increasingly accurate for larger nonlinearities, with steadily decreasing accuracy for each case with increasing noise values, as expected. The final estimate of the system natural frequency was nearly exact for all cases, while the least accurate results were obtained for the final estimates of the

quadratic nonlinearity coefficient, although the estimates increased in accuracy for larger nonlinearities on a percentage basis.

The combined quadratic and cubic nonlinearity example results are slightly less accurate than the quadratic results, but are nonetheless reasonable on a percentage basis, especially for nonlinearities larger than the midpoint parameter values studied. In the worst case, the inaccuracy of the cubic nonlinear parameter estimate reaches 76%, for 20% added noise. For the case where the parameter has increased from 0.5 in the worst case to 2.5, the inaccuracy of the final estimate has decreased to only 12%. The quadratic nonlinear term parameter estimates follow the same trend, increasing in accuracy from 60% error to 2.4% error in the case where the largest nonlinearities were studied. The final estimates for the natural frequency, damping, and impulse coefficients were all within 2%, 0.4%, and 1.4% of the exact values sought in all the cases studied, respectively.

In the drag nonlinearity cases studied, increases in nonlinear parameter values resulted in reduced RMS values used in the percentage noise calculations. The RMS value of generated displacement values decreased from 0.2 in the case of a nonlinearity coefficient of 0.5 to just 0.08 for the largest value case of 5.0. Therefore, the corresponding noise levels also decreased. On a percentage basis, the nonlinear parameter estimates improved from the smallest values sought to the largest, while the drag coefficient estimates decreased in accuracy for increasing nonlinearity coefficients. The former increased in accuracy from 8% error to only 1.2% error, while the latter decreased in accuracy from 3.6% error to 10.6% error for the largest nonlinearity case studied. The natural frequency and impulse parameter estimates remained essentially unchanged for increasing nonlinearities, but decreased slightly in accuracy for increasing noise levels, as expected.

After obtaining the final results above using corrupted input displacement data, digital filtering was used to attempt to improve on the final parameter estimates. Details of the filter used were presented earlier. The final estimates for the first two examples above improved dramatically in most cases, while the final estimates found in the drag nonlinearity case were found to decrease in accuracy slightly in many cases. In the quadratic nonlinearity example, the accuracy of the worst case parameter estimate increased from the 28% error mentioned above to just 6% error after filtering of the input data. In general, estimates in the other parameters had similar decreases in final percentage errors. For the example with two types of nonlinearities, the percentage error of 76% mentioned above was reduced to a more modest value of 20% after input data filtering. In general, similar reductions in final estimate errors were found for the other parameters. For the drag nonlinearity example, estimates for the linear drag coefficient remained essentially the same with filtering, but the natural frequency, impulse parameter, and nonlinearity coefficient were all found to decrease in accuracy by values ranging from 0.02% for some of the natural frequency estimates to approximately 3% increases in error for nonlinearity coefficients in the 20% noise cases.

The iterative direct method was effective for the identification of the three nonlinear examples studied. The results from the quadratic nonlinearity example were among the best of the three studies. The results for the example involving two types of nonlinearities were not as accurate as those for the quadratic example, perhaps because of the larger number of parameters sought. The drag nonlinearity example provided the least accurate results, possibly because of the decreasing nonlinear effects with increasing nonlinearity coefficient values. Increased drag to the system obviously leads to a faster rate of decay, and therefore a lower RMS measured value for displacement. Since increases in the drag nonlinearity coefficient cause the "measured" displacement ampli-



tudes to decrease, less influence from the nonlinearity term is exerted on the differential equation by this term. Perhaps this explains the decreases in accuracy for estimates of this coefficient for increasing values of the assumed nonlinear coefficient, in contrast to the other two examples. For the exact parameter estimates generated without the introduction of noise, pure and identified responses would look the same, since the same subroutine is used to solve the differential equation in both cases. Representative case plots are presented in Figures 4 and 5 for the quadratic and quadratic and cubic nonlinearity examples. The largest nonlinearity drag case is plotted in Figure 6 to illustrate the dramatic change which takes place in amplitude for increased values of  $\alpha$  in the governing differential equation.

For the first two examples, filtering increased the accuracy of final estimates considerably. For the drag nonlinearity example, filtering increased accuracy only for linear drag coefficient estimates. For the other parameters involved, parameter estimates became less accurate than those obtained prior to input data filtering for increasing nonlinear coefficient values, perhaps because the amplitudes in displacement values became smaller, causing filtering to become more difficult. The same examples plotted for corrupted unfiltered input data are plotted for the parameter estimates obtained after filtering of the input data in Figures 7, 8, and 9.

In order to apply the method to an actual structural system, generation of synthetic measured data would be necessary in order to evaluate the method and the filter effectiveness for the particular nonlinearity type involved, before attempting to make estimates of actual system parameters. The use of the actual Newton-Raphson method would probably have increased the rate of convergence, since the incorrect step sizes used tended to cause the intermediate values of the parameters to jump across the actual

values sometimes before convergence. The second derivatives needed for the Hessian expression presented above would require the solution of a much larger number of differential equations, with quantities defined as in equations [2.16] for the second derivatives of the displacement and velocity with respect to the parameters.

Table 1. Iterative Direct Method; Quadratic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = Q$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$	Q
Initial		1.00	0.35	22.50	3.50
	0%	0.50	0.50	25.00	5.00
	5%	0.47	0.50	25.00	4.98
	10%	0.43	0.49	25.00	4.97
	20%	0.36	0.49	25.01	4.94
Exact		0.50	0.50	25.00	5.00
Initial		0.50	0.35	22.50	3.50
	0%	1.50	0.50	25.00	5.00
	5%	1.47	0.50	25.00	4.98
	10%	1.44	0.49	25.00	4.97
	20%	1.38	0.49	25.01	4.94
Exact		1.50	0.50	25.00	5.00
Initial		1.50	0.35	22.50	3.50
	0%	2.50	0.50	25.00	5.00
	5%	2.47	0.50	25.00	4.98
	10%	2.45	0.49	25.00	4.97
	20%	2.40	0.49	25.00	4.94
Exact		2.50	0.50	25.00	5.00
Initial		2.50	0.35	22.50	3.50
	0%	5.00	0.50	25.00	5.00
	5%	4.98	0.50	25.00	4.99
	10%	4.97	0.49	25.00	4.97
	20%	4.94	0.49	25.00	4.94
Exact		5.00	0.50	25.00	5.00

Table 2. Iterative Direct Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t=0) = 0$$

$$\dot{u}(t=0) = Q$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$	Q
Initial		0.50	1.00	0.35	22.50	3.50
	0%	0.10	0.50	0.50	25.00	5.00
	10%	0.12	0.31	0.49	25.05	4.97
	20%	0.13	0.12	0.49	25.10	4.94
Exact		0.10	0.50	0.50	25.00	5.00
Initial		1.00	1.00	0.35	22.50	3.50
	0%	0.50	2.50	0.50	25.00	5.00
	10%	0.54	2.30	0.49	25.00	4.97
	20%	0.57	2.10	0.49	25.10	4.93
Exact		0.50	2.50	0.50	25.00	5.00
Initial		0.50	1.00	0.35	22.50	3.50
	0%	1.00	2.50	0.50	25.00	5.00
	10%	1.04	2.29	0.49	25.00	4.96
	20%	1.07	2.10	0.49	25.10	4.93
Exact		1.00	2.50	0.50	25.00	5.00
Initial		0.50	2.50	0.35	22.50	3.50
	0%	1.00	5.00	0.50	25.00	5.00
	10%	1.06	4.79	0.49	25.04	4.96
	20%	1.12	4.57	0.49	25.10	4.93
Exact		1.00	5.00	0.50	25.00	5.00
Initial		1.00	2.50	0.35	22.50	3.50
	0%	2.50	5.00	0.50	25.00	5.00
	10%	2.56	4.79	0.49	25.04	4.96
	20%	2.62	4.58	0.49	25.10	4.93
Exact		2.50	5.00	0.50	25.00	5.00

Table 3. Iterative Direct Method; Drag Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = Q$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$	Q
Initial		1.00	0.35	22.50	3.50
	0%	0.50	0.50	25.00	5.00
	5%	0.49	0.50	24.99	4.96
	10%	0.48	0.50	24.99	4.92
	20%	0.46	0.50	24.97	4.84
Exact		0.50	0.50	25.00	5.00
Initial		0.50	0.35	22.50	3.50
	0%	1.50	0.50	25.00	5.00
	5%	1.49	0.50	24.98	4.96
	10%	1.48	0.49	24.97	4.92
	20%	1.47	0.48	24.94	4.83
Exact		1.50	0.50	25.00	5.00
Initial		1.50	0.35	22.50	3.50
	0%	2.50	0.50	25.00	5.00
	5%	2.50	0.49	24.98	4.96
	10%	2.49	0.48	24.96	4.92
	20%	2.49	0.47	24.92	4.84
Exact		2.50	0.50	25.00	5.00
Initial		2.50	0.35	22.50	3.50
	0%	5.00	0.50	25.00	5.00
	5%	5.02	0.49	24.97	4.97
	10%	5.03	0.47	24.95	4.95
	20%	5.06	0.45	24.91	4.89
Exact		5.00	0.50	25.00	5.00

Table 4. Iterative Direct Method; Quadratic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = Q$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$	Q
Initial		1.00	0.35	22.50	3.50
	0%	0.50	0.50	25.00	5.00
	5%	0.49	0.50	24.93	4.99
	10%	0.48	0.50	24.93	4.98
	20%	0.47	0.50	24.94	4.97
Exact		0.50	0.50	25.00	5.00
Initial		0.50	0.35	22.50	3.50
	0%	1.50	0.50	25.00	5.00
	5%	1.49	0.50	24.93	4.99
	10%	1.49	0.50	24.93	4.98
	20%	1.48	0.50	24.94	4.97
Exact		1.50	0.50	25.00	5.00
Initial		1.50	0.35	22.50	3.50
	0%	2.50	0.50	25.00	5.00
	5%	2.49	0.50	24.93	4.99
	10%	2.49	0.50	24.93	4.98
	20%	2.50	0.50	24.94	4.97
Exact		2.50	0.50	25.00	5.00
Initial		2.50	0.35	22.50	3.50
	0%	5.00	0.50	25.00	5.00
	5%	5.00	0.50	24.93	4.99
	10%	5.01	0.50	24.93	4.98
	20%	5.03	0.50	24.93	4.97
Exact		5.00	0.50	25.00	5.00

Table 5. Iterative Direct Method; Quadratic, Cubic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = Q$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$	Q
Initial		0.50	1.00	0.35	22.50	3.50
	0%	0.10	0.50	0.50	25.00	5.00
	10%	0.10	0.45	0.50	24.94	4.98
	20%	0.10	0.40	0.50	24.96	4.97
Exact		0.10	0.50	0.50	25.00	5.00
Initial		1.00	1.00	0.35	22.50	3.50
	0%	0.50	2.50	0.50	25.00	5.00
	10%	0.51	2.44	0.50	24.95	4.98
	20%	0.52	2.38	0.50	24.98	4.97
Exact		0.50	2.50	0.50	25.00	5.00
Initial		0.50	1.00	0.35	22.50	3.50
	0%	1.00	2.50	0.50	25.00	2.50
	10%	1.01	2.44	0.50	24.95	4.98
	20%	1.03	2.38	0.50	24.98	4.97
Exact		1.00	2.50	0.50	25.00	5.00
Initial		0.50	2.50	0.35	22.50	3.50
	0%	1.00	5.00	0.50	25.00	5.00
	10%	1.02	4.92	0.50	24.96	4.98
	20%	1.04	4.86	0.50	25.00	4.97
Exact		1.00	5.00	0.50	25.00	5.00
Initial		1.00	2.50	0.35	22.50	3.50
	0%	2.50	5.00	0.50	25.00	5.00
	10%	2.53	4.92	0.50	24.97	4.98
	20%	2.56	4.86	0.50	25.00	4.97
Exact		2.50	5.00	0.50	25.00	5.00

Table 6. Iterative Direct Method; Drag Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = Q$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$	Q
	-----	-----	-----	-----	-----
Initial		1.00	0.35	22.50	3.50
	0%	0.50	0.50	25.00	5.00
	5%	0.50	0.51	24.93	4.98
	10%	0.49	0.51	24.93	4.97
	20%	0.48	0.52	24.94	4.94
Exact		0.50	0.50	25.00	5.00
Initial		0.50	0.35	22.50	3.50
	0%	1.50	0.50	25.00	5.00
	5%	1.49	0.51	24.93	4.97
	10%	1.47	0.52	24.93	4.95
	20%	1.45	0.54	24.94	4.91
Exact		1.50	0.50	25.00	5.00
Initial		1.50	0.35	22.50	3.50
	0%	2.50	0.50	25.00	5.00
	5%	2.48	0.51	24.93	4.96
	10%	2.46	0.52	24.93	4.93
	20%	2.42	0.55	24.94	4.88
Exact		2.50	0.50	25.00	5.00
Initial		2.50	0.35	22.50	3.50
	0%	5.00	0.50	25.00	5.00
	5%	4.95	0.52	24.92	4.94
	10%	4.91	0.53	24.93	4.90
	20%	4.83	0.56	24.93	4.82
Exact		5.00	0.50	25.00	5.00



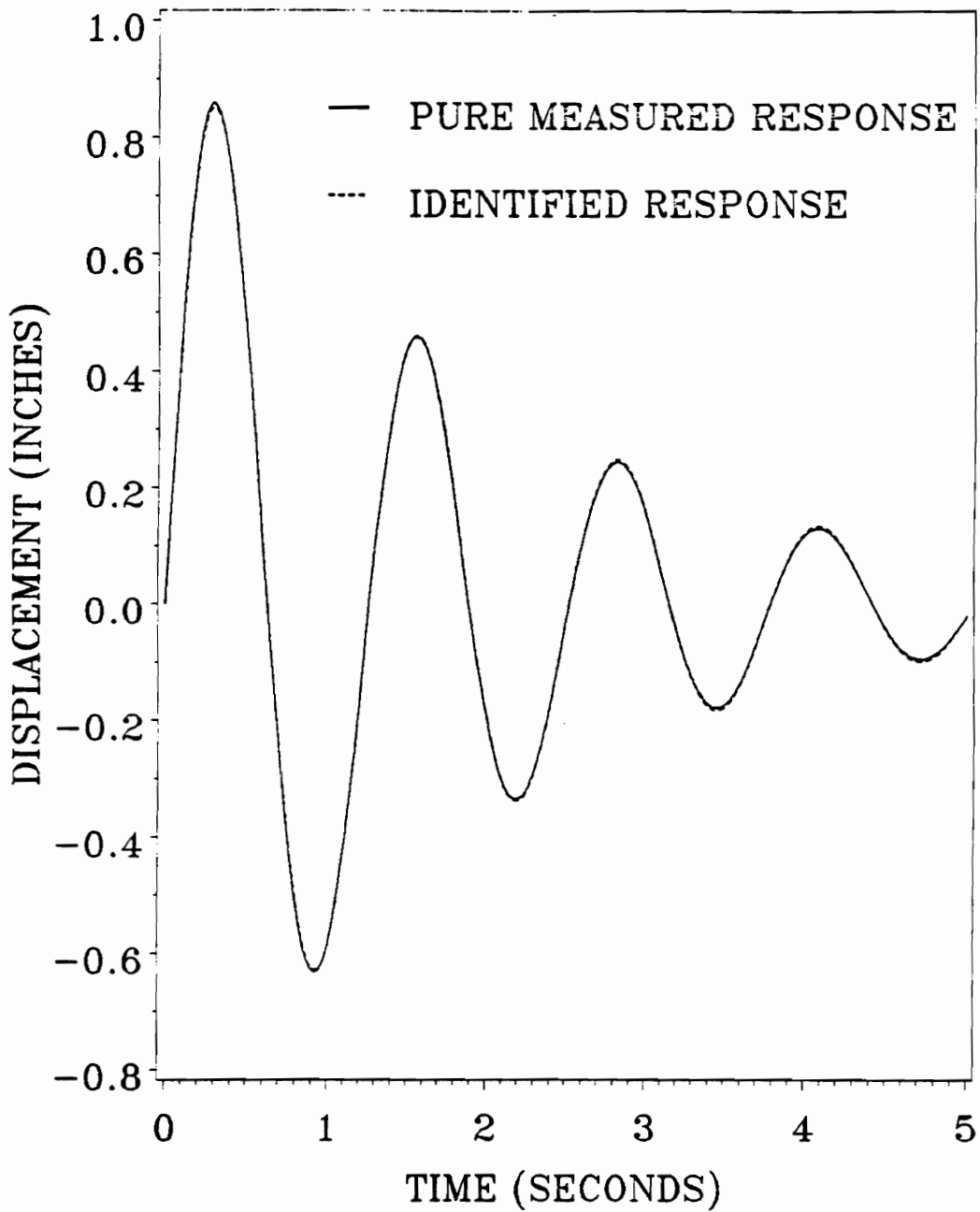


Figure 4. Identified/Measured Response of Quadratic Nonlinearity Example: Iterative Direct Method; 20% corrupted input data, unfiltered,  $\alpha = 0.5$

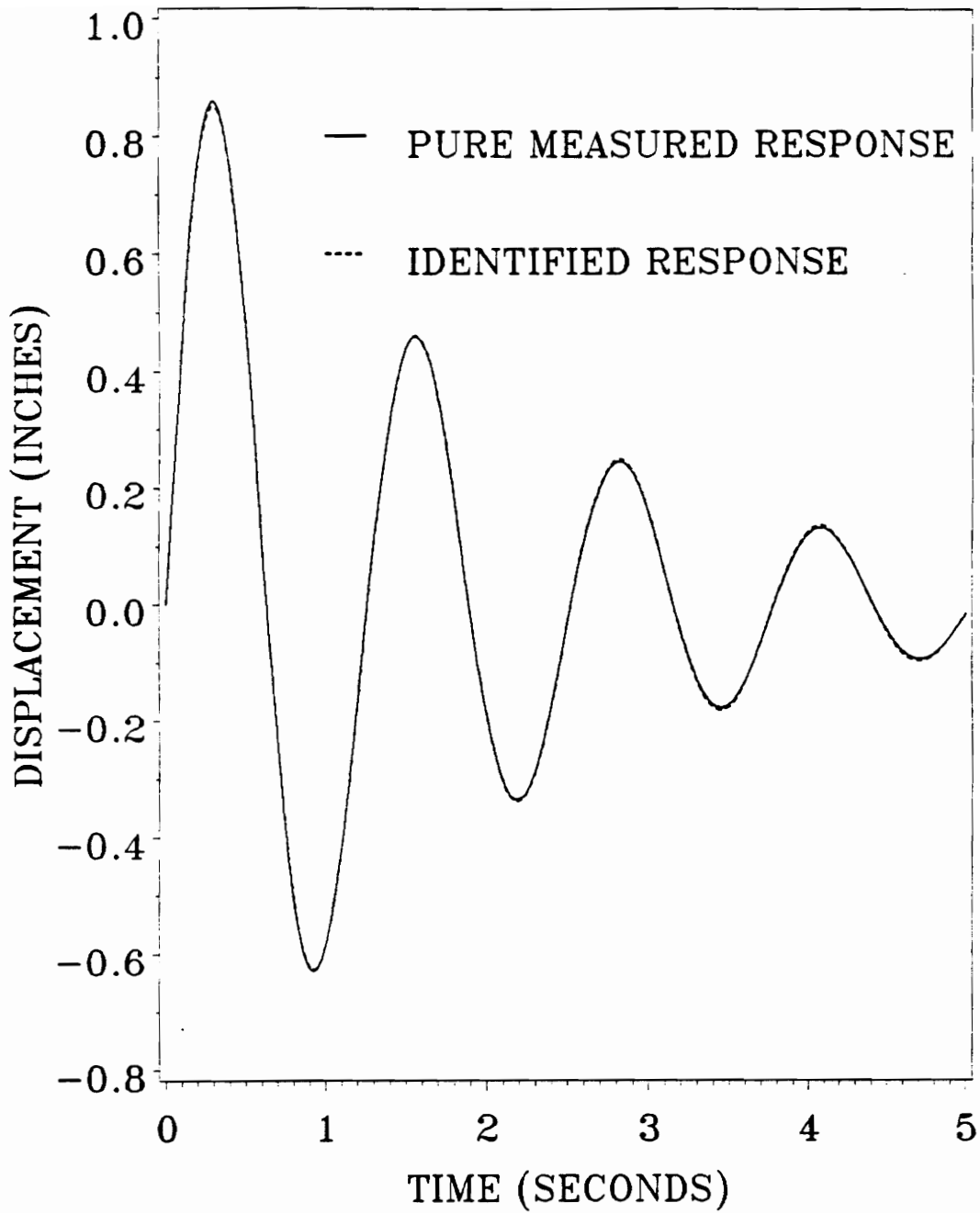


Figure 5. Identified/Measured Response of Quadratic and Cubic Nonlinearity  
 Example: Iterative Direct Method; 20% corrupted input data, unfiltered ,  
 $\alpha_1 = 0.1$ ,  $\alpha_2 = 0.5$

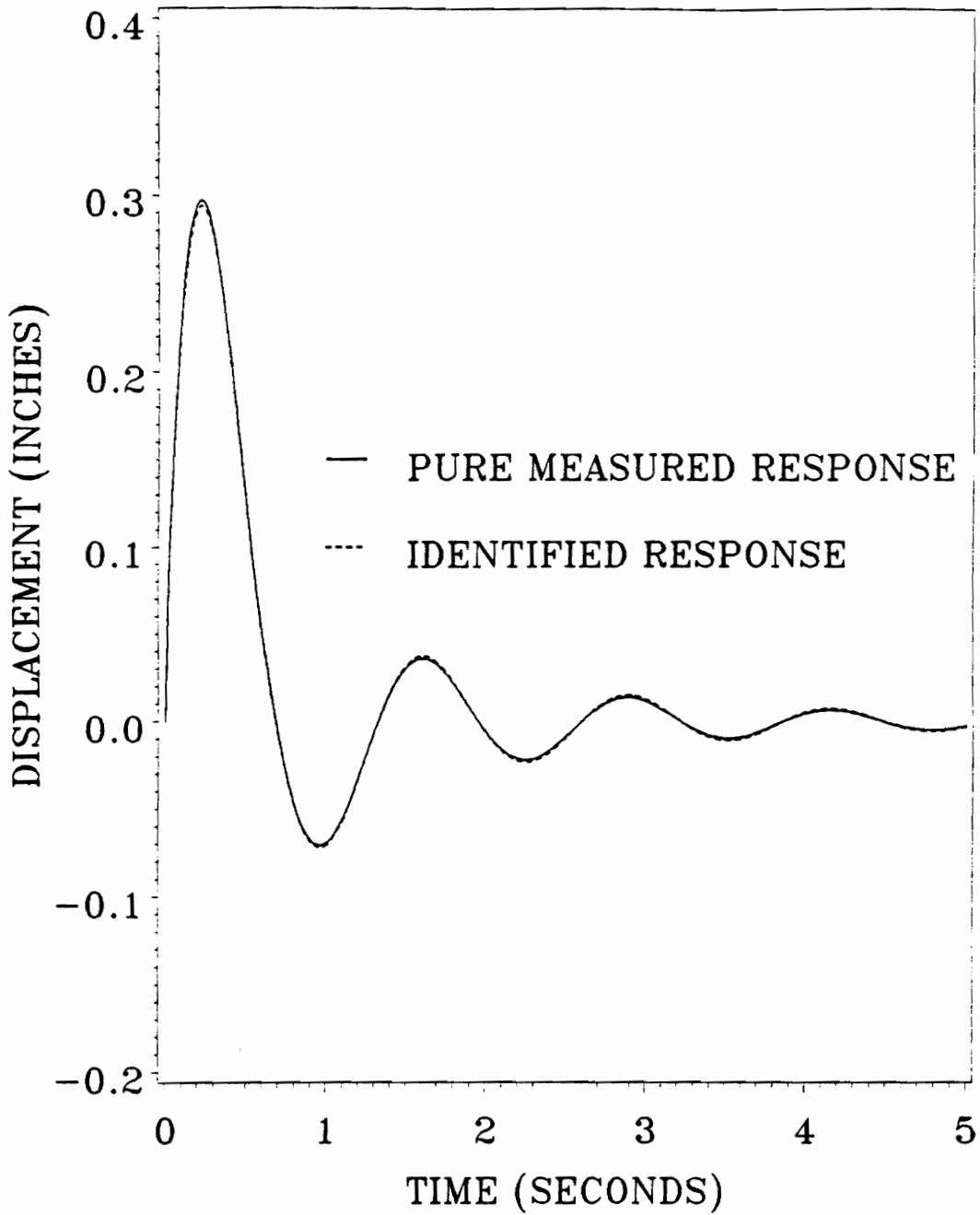


Figure 6. Identified/Measured Response of Drag Nonlinearity Example: Iterative Direct Method; 20% corrupted input data, unfiltered,  $\alpha = 5.0$

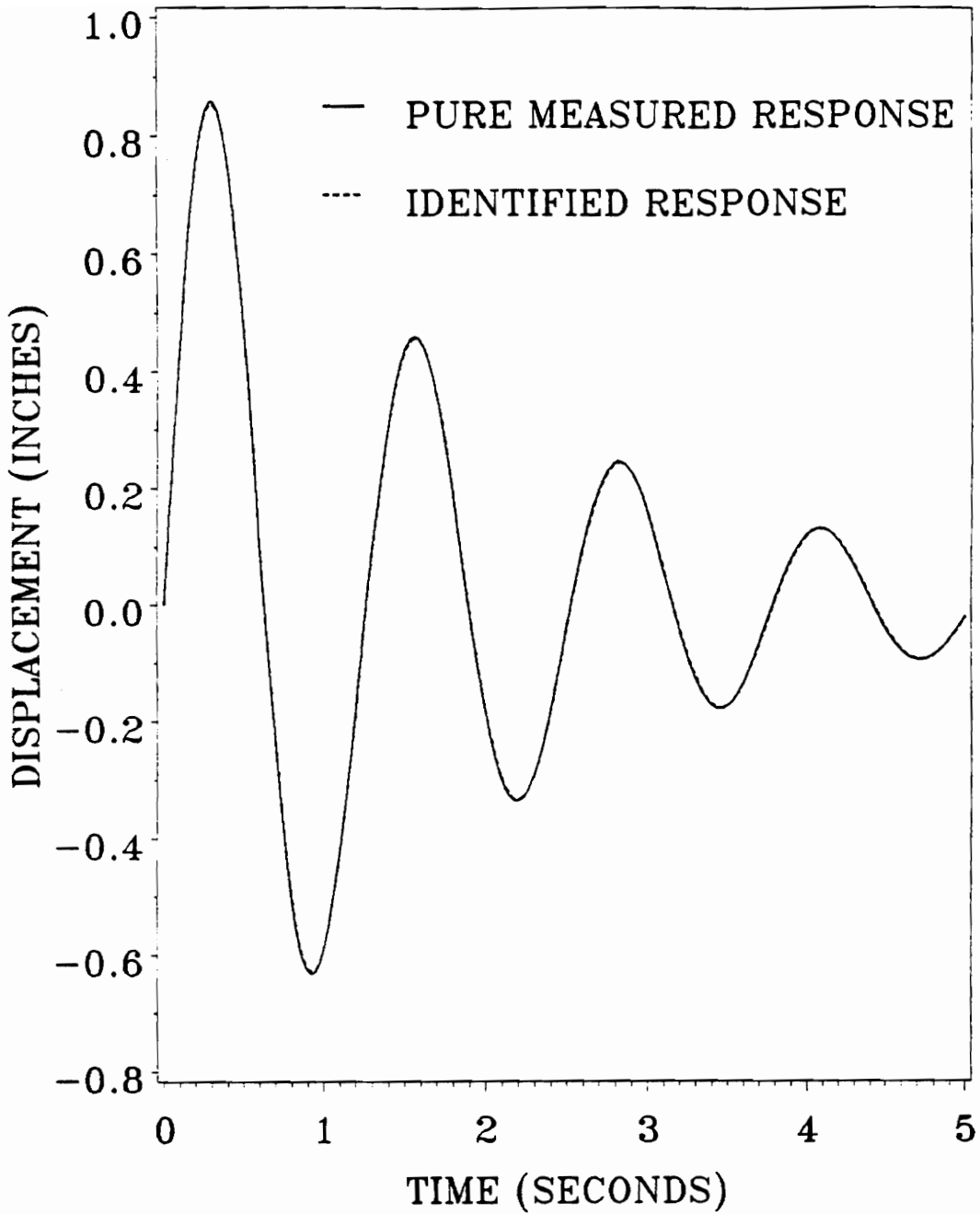


Figure 7. Identified/Measured Response of Quadratic Nonlinearity Example: Iterative Direct Method; 20% corrupted input data, filtered,  $\alpha = 0.5$

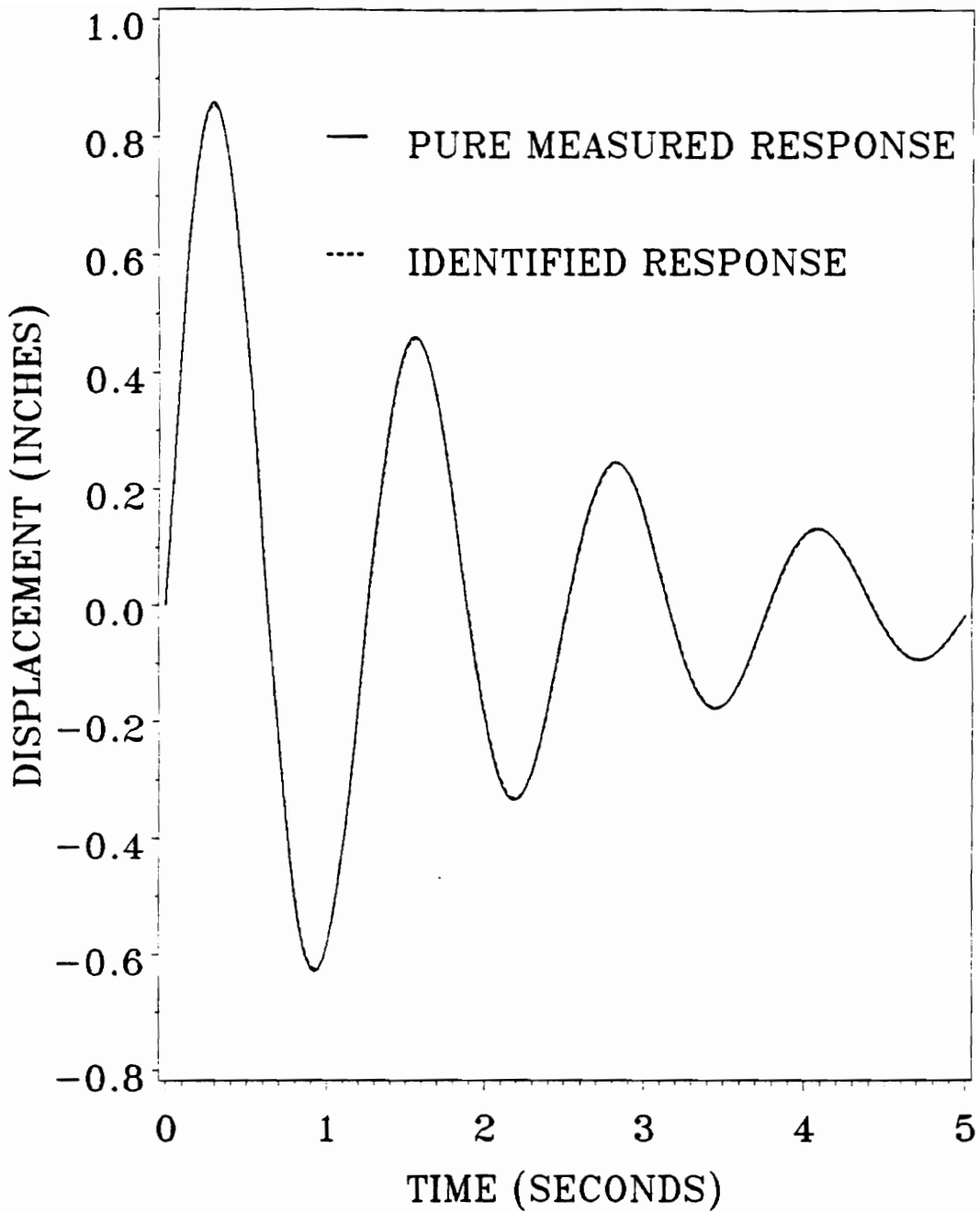


Figure 8. Identified/Measured Response of Quadratic and Cubic Nonlinearity  
 Example: Iterative Direct Method; 20% corrupted input data, filtered ,  
 $\alpha_1 = 0.1$ ,  $\alpha_2 = 0.5$

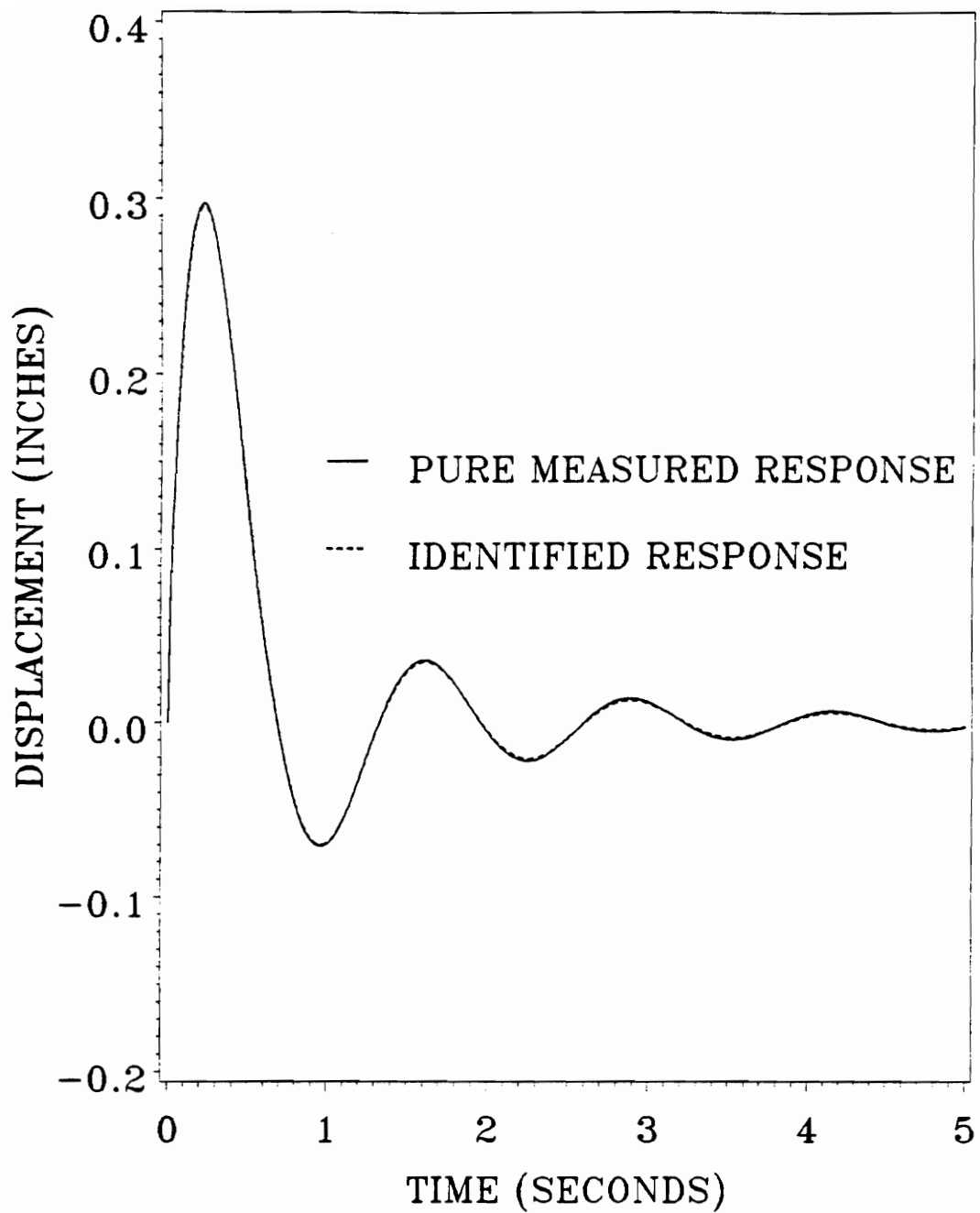


Figure 9. Identified/Measured Response of Drag Nonlinearity Example: Iterative Direct Method; 20% corrupted input data, filtered,  $\alpha = 5.0$

## 3.0 Single Step Integration Iterative Method : Single Degree of Freedom Systems

In the formulation of the objective function using the iterative direct method, it is assumed that the complete solution of differential equations such as equations [2.17] is needed for each new set of parameters  $\{X\}^{i+1}$ . The objective function formulation presented in this chapter circumvents this problem. Also, in the iterative direct method, the second order partial derivative terms of equation [2.11] are neglected, leading to slower convergence of the method than could be expected using the classical Newton-Raphson method. This drawback of the iterative direct method is also avoided by the reformulation of the objective function. In order to introduce the method, some basic concepts of numerical integration of differential equations are required.

Any  $n$ th order ordinary differential equation may be transformed to a system of  $n$  first order differential equations by introducing auxiliary variables. In the simple linear differential equation [2.2], omitting a forcing function  $F(t)$ , an extra variable for velocity

may be introduced, so that rather than solving the equation in the form [2.2], it is transformed to the first order system of two equations

$$\dot{y}_1 = y_2 = f_1 \quad [3.1]$$

$$\dot{y}_2 = -2\gamma y_2 - \Omega^2 y_1 = f_2$$

where  $y_1$  is the displacement  $u$  and  $y_2$  is the velocity  $\dot{u}$ . In order to solve the system, a number of numerical procedures could be implemented. For the procedure introduced below, a single step Runge-Kutta method is implemented of the form

$$\{y\}^{l+1} = \{y\}^l + h\{\phi\} \quad [3.2]$$

where  $\{y\}$  is the numerical approximation to the solution of the differential equations in the form of a two component vector,  $\{\phi\}$  is in general a two component vector of nested function evaluations dependent on the value of  $\{y\}^l$ , and  $h$  is the time step used for the fixed step procedure used. In order to use the method for the solution of the system [3.1], an initial set of conditions for  $\{y\}$  must be specified, a step size  $h$  must be chosen, and the function evaluations made at each step for  $\{\phi\}$ . The solution of the equations is then given as time series in the displacement  $y_1$  and the velocity  $y_2$ . For a simple solution procedure, the function  $\{\phi\}$  may take the form

$$\{\phi\} = \frac{1}{2} \{F(t^l, y^l) + F(y^l + hF(y^l))\} \quad [3.3]$$

The method above is accurate to order two in the step size  $h$ . For this example, the function  $F$  is composed of components  $f_1$  and  $f_2$  as given in equation [3.1]. An algorithm is usually implemented where a subroutine is called in order to evaluate  $F$  for the values needed in  $\{\phi\}$ .



### 3.1 Objective Function Formulation

In the equations above, if it is assumed that the solution of the differential equation is known a priori, a difference function may be formulated at each point of the solution which depends on the parameters  $\{X\}$ . The values for  $\{y\}^{i+1}$  are now given as a function of the previous values for  $\{y\}$ . For each required value for  $\{y\}$ , the measured displacement and velocity may be substituted in each side of equation [3.2], and a corresponding error function dependent on the parameters  $\{X\}$  may be found by moving all the terms of the equation to one side. This gives an error function dependent on the parameters governing the system. At each point  $i+1$ , the error function takes the form

$$\{E\}^{i+1} = -\{y\}^{i+1} + \{y\}^i + h\{\phi(\{y\}^i, t^i, \{X\})\} \quad [3.4]$$

which for the  $\{\phi\}$  given in equation [3.3] is

$$\{E\}^{i+1} = -\{\bar{y}_m\}^{i+1} + \{\bar{y}_m\}^i + \frac{h}{2} \{F(\{\bar{y}_m\}^i) + F(\{\bar{y}_m\}^i + hF(\{\bar{y}_m\}^i))\} \quad [3.5]$$

where  $\{\bar{y}\}$  is the measured vector of displacement and velocity. The error function is thus defined for discrete values, as is the case for the objective function of Chapter 2, equation [2.4]. Therefore, an objective function is used for this procedure of the form

$$L\{X\} = \int_0^{T^{\star}} \left\{ \frac{1}{T^{\star 2}} E_1^2 + E_2^2 \right\} dt \quad [3.6]$$

where  $E_1$  is the displacement component of the error function described above, and  $E_2$  is the velocity component.

$E_1^2$  is divided by  $T^{\star 2}$  in order to have the dimensions of both terms of the integrand in units of velocity squared. Note that the notation used for  $L$  is not exactly correct, since the integrand is not a continuous function, but is comprised of discrete values, as in equation [2.4]. As in the objective function of Chapter 2, a quadrature formula may be applied to the integral, since the integrand is given as a time series of equally spaced values. Simpson's rule is again used to approximate the integral expression of the objective function. Essentially, a quadratic polynomial is passed through each group of three points, and the areas between the polynomials are summed to give an approximation of an actual integral of the difference between the two functions. Unfortunately, because of the nature of one step solution methods, the initial conditions may not be included as parameters which are to be identified with the usual system parameters. Therefore, the number of parameters identified in each of the following examples is one less than the corresponding problem of the second chapter.

### ***3.2 Objective Function Differentiation***

The objective function of the iterative direct method is found using a time series of solution points of the differential equations of the form [2.17]. The first derivatives are found as time series from the solution of a set of differential equations (such as equations [2.17]). In order to compute the exact expression for the Hessian, it was observed in Chapter 2 that a set of differential equations would need to be solved so as to

find the second derivatives of the displacement with respect to the parameters. Therefore, the second derivatives were ignored, and the step size of each iteration was found not to be an optimum amount, as specified in the classical Newton-Raphson method. For the simple form of the objective function [3.6] above, the first and second derivatives may be calculated analytically. Since in many cases the functions  $f_1$  and  $f_2$  of equation [3.5] are linear in the parameters, the derivatives with respect to the parameters are simple to compute analytically. This also implies that the classical formulation for the Newton-Raphson method may be implemented.

The gradient of the objective function takes a form much like equation [2.6]:

$$\frac{\partial L\{X\}}{\partial x_i} = 2 \int_0^{T^*} \left\{ \frac{1}{T^{*2}} E_1 \frac{\partial E_1}{\partial x_i} + E_2 \frac{\partial E_2}{\partial x_i} \right\} dt \quad [3.7]$$

The Hessian takes the form

$$\begin{aligned} \frac{\partial}{\partial x_k} \left( \frac{\partial L\{x\}}{\partial x_i} \right) = & 2 \int_0^{T^*} \left\{ \frac{1}{T^{*2}} \frac{\partial E_1}{\partial x_k} \frac{\partial E_1}{\partial x_i} + \frac{E_1}{T^{*2}} \frac{\partial}{\partial x_k} \left( \frac{\partial E_1}{\partial x_i} \right) \right\} dt + \\ & 2 \int_0^{T^*} \left\{ \frac{\partial E_2}{\partial x_k} \frac{\partial E_2}{\partial x_i} + E_2 \frac{\partial}{\partial x_k} \left( \frac{\partial E_2}{\partial x_i} \right) \right\} dt \end{aligned} \quad [3.8]$$

where  $E_1$  and  $E_2$  are as given in equation [3.5]. In order to take derivatives of  $\{E\}$  for the expressions above, simple analytical derivatives are formulated. The derivatives of the

measured displacements and velocities with respect to the parameters are zero, so that the only derivatives of  $\{\phi\}$  required are of  $\{F\}$ , such as in equation [3.1].

### ***3.3 Objective Function Minimization***

Since the derivatives required in the gradient and Hessian expressions are found to be simple to compute analytically, the classical Newton-Raphson method is now implemented, using the formulation from Chapter 2, equations [2.7] through [2.9]. Iterations are again performed until a specified convergence criterion is met. For the examples presented, the same criterion is implemented as in Chapter 2, with an overall tolerance of 5.0E-4 between old and new components of  $\{X\}$ .

### ***3.4 Applications of the One Step Integration Iterative***

#### ***Method***

The examples presented in Chapter 2 will also be presented here in order to perform a comparison of the iterative direct and one step integration methods. Again, the record length of time is five seconds, with 335 divisions. As noted above, the forcing function parameter  $Q$  of the examples is not identified as in Chapter 2.

### 3.4.1 Quadratic Nonlinearity Example

Equation [2.15] is again the governing differential equation for this example, with the same forcing function  $Q\delta(t)$  as well. The first step in the procedure, as noted above, is to transform the differential equation to a system of two first order differential equations. The system of equations takes the form

$$\begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{Bmatrix} = \begin{Bmatrix} y_2 \\ -x_1 y_1 - 2x_2 y_2 - x_3 y_1^2 \end{Bmatrix} \quad [3.9]$$

where  $\{X\}$  is the vector  $\{\Omega^2, \gamma, \alpha\}$ ,  $y_1$  is the displacement, and  $y_2$  is the velocity. The objective function takes the general form specified in equation [3.6], where for this example

$$E_1 = -\bar{y}_1^{i+1} + \bar{y}_1^i + h\bar{y}_2^i - \frac{h^2}{2} x_1 \bar{y}_1^i - h^2 x_2 \bar{y}_2^i - \frac{h^2}{2} x_3 \bar{y}_1^{i2} \quad [3.10]$$

and

$$\begin{aligned} E_2 = & -\bar{y}_2^{i+1} + \bar{y}_2^i + \frac{h}{2} (-x_1 \bar{y}_1^i - 2x_2 \bar{y}_2^i - x_3 \bar{y}_1^{i2}) - \frac{h}{2} x_1 (\bar{y}_1^i + h\bar{y}_2^i) \\ & + \frac{h}{2} (-2x_2 \bar{y}_2^i + 2hx_1 x_2 \bar{y}_1^i + 4hx_2^2 \bar{y}_2^i + 2hx_2 x_3 \bar{y}_1^{i2} - x_3 (\bar{y}_1^i + h\bar{y}_2^i)^2) \end{aligned} \quad [3.11]$$

Therefore, the derivatives of  $\{E\}$  with respect to  $\{X\}$  take the form

$$\frac{\partial E_1}{\partial x_1} = -\frac{h^2}{2} \bar{y}_1^i$$

$$\frac{\partial E_1}{\partial x_2} = -h^2 \bar{y}_2^i$$

$$\frac{\partial E_1}{\partial x_3} = -\frac{h^2}{2} \bar{y}_1^{i2} \quad [3.12]$$

$$\frac{\partial E_2}{\partial x_1} = -\frac{h}{2} \bar{y}_1^i - \frac{h}{2} (\bar{y}_1^i + h\bar{y}_2^i) + \frac{h}{2} (2hx_2\bar{y}_1^i)$$

$$\frac{\partial E_2}{\partial x_2} = -2h\bar{y}_2^i + \frac{h}{2} (2hx_1\bar{y}_1^i + 8hx_2\bar{y}_2^i + 2hx_3\bar{y}_1^{i2})$$

$$\frac{\partial E_2}{\partial x_3} = -\frac{h}{2} \bar{y}_1^{i2} + \frac{h}{2} (2hx_2\bar{y}_1^{i2} - (\bar{y}_1^i + h\bar{y}_2^i)^2)$$

The second derivatives of {E} are calculated in the same way, with many of the values equal to zero since the error functions are linear in the most of the parameters. The second derivatives which are not identically zero are

$$\frac{\partial}{\partial x_1} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_1} \right) = h^2(\bar{y}_1^i)$$

$$\frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{h}{2} (8h\bar{y}_2^i) \quad [3.13]$$

$$\frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_3} \right) = \frac{\partial}{\partial x_3} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{h}{2} (2h\bar{y}_1^{i2})$$

The derivatives required in equations [3.7] and [3.8] are thus computed for all times  $t$  in the interval 0 to  $T^*$ . The integrals of the derivatives are again approximated using Simpson's rule for each group of three points in the integrand formulations. The Newton-Raphson algorithm is now implemented, with iterations on {X} until changes in the components of {X} are reduced to 5.0E-4 of the previous values.

### 3.4.2 Quadratic and Cubic Nonlinearity Example

For this example, equation [2.20] is again used to describe the system. The procedure follows exactly as in the quadratic example above, with the exception that there are now four parameters to be identified rather than three. Again, the number of parameters to be identified has been reduced by one from the procedure of Chapter 2 applied to this example (five), since the initial conditions are required to begin the process of minimization of the error functions  $\{E\}$ . The transformed equation of motion takes the form

$$\begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{Bmatrix} = \begin{Bmatrix} y_2 \\ -x_1 y_1 - 2x_2 y_2 - x_3 y_1^2 - x_4 y_1^3 \end{Bmatrix} \quad [3.14]$$

The error functions  $E_1$  and  $E_2$  therefore take the form

$$E_1 = -\bar{y}_1^{i+1} + \bar{y}_1^i + h\bar{y}_2^i - \frac{h^2}{2} x_1 \bar{y}_1^i - h^2 x_2 \bar{y}_2^i - \frac{h^2}{2} x_3 \bar{y}_1^{i2} - \frac{h^2}{2} x_4 \bar{y}_1^{i3} \quad [3.15]$$

and

$$\begin{aligned} E_2 = & -\bar{y}_2^{i+1} + \bar{y}_2^i + \frac{h}{2} (-x_1 \bar{y}_1^i - 2x_2 \bar{y}_2^i - x_3 \bar{y}_1^{i2} - x_4 \bar{y}_1^{i3}) \\ & + \frac{h}{2} x_1 (\bar{y}_1^i + h\bar{y}_2^i) + \frac{h}{2} (-2x_2 \bar{y}_2^i + 2hx_1 x_2 \bar{y}_1^i \\ & + 4hx_2^2 \bar{y}_2^i + 2hx_2 x_3 \bar{y}_1^{i2} + 2hx_2 x_4 \bar{y}_1^{i3} - x_3 (\bar{y}_1^i + h\bar{y}_2^i)^2 - x_4 (\bar{y}_1^i + h\bar{y}_2^i)^3) \end{aligned} \quad [3.16]$$

The derivatives are calculated analytically, as in the quadratic nonlinearity example, and take the form

$$\frac{\partial E_1}{\partial x_1} = -\frac{h^2}{2} \bar{y}_1^i$$

$$\frac{\partial E_1}{\partial x_2} = -h^2 \bar{y}_2^i$$

$$\frac{\partial E_1}{\partial x_3} = -\frac{h^2}{2} \bar{y}_1^{i2} \quad [3.17]$$

$$\frac{\partial E_1}{\partial x_4} = -\frac{h^2}{2} \bar{y}_1^{i3}$$

$$\frac{\partial E_2}{\partial x_1} = -\frac{h}{2} \bar{y}_1^i + \frac{h}{2} (2hx_2 \bar{y}_1^i - (\bar{y}_1^i + h\bar{y}_2^i))$$

$$\frac{\partial E_2}{\partial x_2} = -2h\bar{y}_2^i + \frac{h}{2} (2x_1 h\bar{y}_1^i + 8hx_2 \bar{y}_2^i + 2hx_3 \bar{y}_1^{i2} + 2hx_4 \bar{y}_1^{i3})$$

$$\frac{\partial E_2}{\partial x_3} = -\frac{h}{2} \bar{y}_1^{i2} + \frac{h}{2} (2hx_2 \bar{y}_1^{i2} - (\bar{y}_1^i + h\bar{y}_2^i)^2)$$

$$\frac{\partial E_2}{\partial x_4} = -\frac{h}{2} \bar{y}_1^{i3} + \frac{h}{2} (2hx_2 \bar{y}_1^{i3} - (\bar{y}_1^i + h\bar{y}_2^i)^3)$$

Many of the second derivatives are identically zero, with those which are not zero given as

$$\frac{\partial}{\partial x_1} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_1} \right) = h^2 (\bar{y}_1^i)$$

$$\frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{h}{2} (8h\bar{y}_2^i) \quad [3.18]$$

$$\frac{\partial}{\partial x_3} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_3} \right) = \frac{h}{2} (2h\bar{y}_1^{i2})$$



$$\frac{\partial}{\partial x_4} \left( \frac{\partial E_2}{\partial x_2} \right) = \frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_4} \right) = \frac{h}{2} (2h\bar{y}_1^3)$$

The derivatives required in equations [3.7] and [3.8] are thus computed, and the iteration process implemented. The convergence criterion of Chapter 2 is again used in this example.

### 3.4.3 Drag Nonlinearity Example

Equation [2.25] is the governing equation for this example, as in the problem of Chapter 2. After transformation to a system of first order equations, the equation takes the form

$$\begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{Bmatrix} = \begin{Bmatrix} y_2 \\ -x_1 y_1 - 2x_2 y_2 - x_3 y_2 |y_2| \end{Bmatrix} \quad [3.19]$$

The corresponding error function vector components take the form

$$E_1 = -\bar{y}_1^{t+1} + \bar{y}_1^t + h\bar{y}_2^t - \frac{h^2}{2} x_1 \bar{y}_1^t - h^2 x_2 \bar{y}_2^t - \frac{h^2}{2} x_3 \bar{y}_2^t |\bar{y}_2^t| \quad [3.20]$$

and

$$\begin{aligned} E_2 = & -\bar{y}_2^{t+1} + \bar{y}_2^t + \frac{h}{2} (-x_1 \bar{y}_1^t - 2x_2 \bar{y}_2^t - x_3 \bar{y}_2^t |\bar{y}_2^t|) - \frac{h}{2} x_1 (\bar{y}_1^t + h\bar{y}_2^t) \\ & + \frac{h}{2} (-2x_2 \bar{y}_2^t + 2hx_1 x_2 \bar{y}_1^t + 4hx_2^2 \bar{y}_2^t + 2hx_2 x_3 \bar{y}_2^t |\bar{y}_2^t|) \\ & - x_3 \frac{h}{2} (\bar{y}_2 - hx_1 \bar{y}_1^t - 2hx_2 \bar{y}_2^t - x_3 h\bar{y}_2^t |\bar{y}_2^t|) |\bar{y}_2^t - hx_1 \bar{y}_1^t - 2hx_2 \bar{y}_2^t - x_3 h\bar{y}_2^t |\bar{y}_2^t| \end{aligned} \quad [3.21]$$

The corresponding derivatives of the error functions with respect to the parameters are

$$\frac{\partial E_1}{\partial x_1} = -\frac{h^2}{2} \bar{y}_1^i$$

$$\frac{\partial E_1}{\partial x_2} = -h^2 \bar{y}_2^i$$

$$\frac{\partial E_1}{\partial x_3} = -\frac{h^2}{2} \bar{y}_2^i |\bar{y}_2^i| \quad [3.22]$$

$$\begin{aligned} \frac{\partial E_2}{\partial x_1} = & -\frac{h}{2} \bar{y}_1^i - \frac{h}{2} (\bar{y}_1^i + h\bar{y}_2^i) + \frac{h}{2} (2hx_2\bar{y}_1^i) + \frac{h}{2} x_3(h\bar{y}_1^i) |p| \\ & + \frac{h}{2} x_3(p)h\bar{y}_1^i(\text{sign}p) \end{aligned}$$

$$\begin{aligned} \frac{\partial E_2}{\partial x_2} = & -2h\bar{y}_2^i + \frac{h}{2} (2hx_1\bar{y}_1^i + 8hx_2\bar{y}_2^i + 2hx_3\bar{y}_2^i |\bar{y}_2^i|) \\ & + x_3 h^2 \bar{y}_2^i |p| + x_3 \frac{h}{2} (p) 2h\bar{y}_2^i(\text{sign}p) \end{aligned}$$

$$\begin{aligned} \frac{\partial E_2}{\partial x_3} = & -\frac{h}{2} \bar{y}_2^i |\bar{y}_2^i| + \frac{h}{2} (2hx_2\bar{y}_2^i |\bar{y}_2^i|) - \frac{h}{2} (p) |p| \\ & + x_3 \frac{h^2}{2} \bar{y}_2^i |\bar{y}_2^i| |p| + x_3 \frac{h}{2} p h \bar{y}_2^i |\bar{y}_2^i| (\text{sign}p) \end{aligned}$$

where  $p = \bar{y}_2^i - hx_1\bar{y}_1^i - 2hx_2\bar{y}_2^i - x_3h\bar{y}_2^i |\bar{y}_2^i|$ , and 'signp' is the sign of p. Again, the assumption that a quantity ( p here) never takes the value of exactly zero is made. The assumption does not cause a computational problem in this example just as no problem was encountered in the iterative direct method applied to this example. For this particular example, none of the second derivatives of the error function  $f_2$  are zero, unlike the second derivatives of  $f_1$ . The second derivatives take the form

$$\frac{\partial}{\partial x_1} \left( \frac{\partial E_2}{\partial x_1} \right) = -\frac{h^2}{2} x_3 \bar{y}_1^i h \bar{y}_1^i (\text{sign}p) - \frac{h}{2} x_3 (h \bar{y}_1^i) h \bar{y}_1^i (\text{sign}p)$$

$$\frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_1} \right) = \frac{\partial}{\partial x_1} \left( \frac{\partial E_2}{\partial x_2} \right) = h^2 \bar{y}_1^i - \frac{h^2}{2} x_3 \bar{y}_1^i 2h\bar{y}_2^i (\text{sign} p) - \frac{h}{2} x_3 (2h\bar{y}_2^i) h\bar{y}_1^i (\text{sign} p)$$

$$\frac{\partial}{\partial x_3} \left( \frac{\partial E_2}{\partial x_1} \right) = \frac{\partial}{\partial x_1} \left( \frac{\partial E_2}{\partial x_3} \right) = \frac{h^2}{2} x_3 \bar{y}_1^i h\bar{y}_2^i |\bar{y}_2^i| (\text{sign} p) - \frac{h}{2} x_3 (h\bar{y}_2^i |\bar{y}_2^i|) h\bar{y}_1^i (\text{sign} p)$$

$$\frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_2} \right) = h^2 4\bar{y}_2^i - h^2 x_3 \bar{y}_2^i (\text{sign} p) 2h\bar{y}_2^i - \frac{h}{2} x_3 (2h\bar{y}_2^i) 2h\bar{y}_2^i (\text{sign} p) \quad [3.23]$$

$$\begin{aligned} \frac{\partial}{\partial x_3} \left( \frac{\partial E_2}{\partial x_2} \right) &= \frac{\partial}{\partial x_2} \left( \frac{\partial E_2}{\partial x_3} \right) = h^2 \bar{y}_2^i |\bar{y}_2^i| + h^2 \bar{y}_2^i |p| + \frac{h}{2} (p) 2h\bar{y}_2^i (\text{sign} p) \\ &\quad - \frac{h^2}{2} x_3 \bar{y}_2^i |\bar{y}_2^i| 2h\bar{y}_2^i (\text{sign} p) - \frac{h}{2} x_3 (2h\bar{y}_2^i) h\bar{y}_2^i |\bar{y}_2^i| (\text{sign} p) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial x_3} \left( \frac{\partial E_2}{\partial x_3} \right) &= \frac{h}{2} (h\bar{y}_2^i |\bar{y}_2^i|) |p| + \frac{h}{2} (p) (\text{sign} p) h\bar{y}_2^i |\bar{y}_2^i| + \frac{h^2}{2} \bar{y}_2^i |\bar{y}_2^i| |p| \\ &\quad - \frac{h^2}{2} x_3 \bar{y}_2^i |\bar{y}_2^i| (\text{sign} p) h\bar{y}_2^i |\bar{y}_2^i| + \frac{h}{2} (p) (\text{sign} p) h\bar{y}_2^i |\bar{y}_2^i| \\ &\quad - \frac{h}{2} x_3 (h\bar{y}_2^i |\bar{y}_2^i|) (\text{sign} p) h\bar{y}_2^i |\bar{y}_2^i| \end{aligned}$$

where  $p = \bar{y}_2 - hx_1\bar{y}_1 - 2hx_2\bar{y}_2 - x_3h\bar{y}_2|\bar{y}_2|$ . The Hessian and gradient expressions are computed using Simpson's rule on the points computed using the above formulations, and the iterative process proceeds as in the above examples, until the usual convergence criterion is met.

### ***3.5 Results***

The final estimates of the single step integration method used above to identify the three single degree of freedom examples were not as accurate as those for the iterative direct method examples. As Table 7 illustrates, the final results using unfiltered input data for the algorithm yielded unacceptable results for the quadratic nonlinearity example, with final percentage errors of up to 1300% in the estimate for the nonlinearity parameter in the smallest parameter value case, with 20% noise. For pure input data, the method yielded very reasonable results, with final values for the nonlinearity parameter estimates within 2% for all cases. The final estimates for the linear damping coefficient and natural frequency were all within 0.18% and 0.09% respectively, for pure input data. The RMS values of the displacement and velocity used in the noise calculations were 0.3 and 1.6 for all cases, respectively.

The estimates for the parameters of the other two examples studied were not in good agreement with the exact parameter values, as illustrated in Tables 8 and 9. For pure input displacement values, the method yielded reasonable results, but for corrupted input data, the final estimates for the parameters were inaccurate to an extent which would not warrant the use of the results to predict the structural response of actual systems. The RMS values used in the noise calculations of the drag nonlinearity system ranged from 0.2 and 1.1 to 0.08 and 0.5 for displacement and velocity, respectively. The RMS values took on maximum values for the smallest nonlinearity coefficients, since more damping is introduced to the systems with larger nonlinear damping coefficients. The RMS values for the displacement and velocity of the quadratic and cubic nonlinear system were 0.3 and 1.6, respectively.

For filtered input data, reductions of the high frequency data components was not sufficient to yield reasonable estimates for most cases, as illustrated in Tables 10, 11, and 12. Since the method is so sensitive to any additional noise in input data, perhaps the low frequency components of the noise passed by the low pass filter are sufficient to introduce large inaccuracies in the final parameter estimates.

For all of the cases studied, the method above converged to the final parameter estimates with three to six iterations, the most iterations required for the drag nonlinearity example. The most remarkable aspect of the computations was that the process required only one to two seconds of CPU time, in contrast to six or seven seconds required for the iterative direct method. The primary source of savings in computational time is the fact that the solution of the governing differential equation is not required at each iteration in  $\{X\}$ . The differential equation is never actually solved for the output of the system for intermediate values of the parameters, so the effort required to find the gradient and Hessian values is much reduced from the amount required to solve a system of equations such as equations [2.17].

Since the method gave good results for the cases where pure input data was used, but unreasonable results for corrupted data values, it would probably be considered too sensitive to impure input measurements for use on actual structural systems. Perhaps the reason for the inaccuracies incurred in the use of corrupted input data is that at each point for the error function  $\{E\}$  of equation [3.5], the method depends only on the previous value of the solution of the differential equation and its function evaluations. The same type of inaccuracy is found in the actual solution of differential equations (see Ref. [20]). If the error function at each point were to depend on a number of previous measured values and their function evaluations, the effects of noise could be reduced.

In order to use a higher order function for  $\{\phi\}$  in equation [2.3], the derivatives required for the expressions of equations [3.7] and [3.8] would be of nested function evaluations, and would require a much larger amount of algebraic manipulation. The problems above may be circumvented by use of a multiple step algorithm for the solution of differential equations, as presented in Reference [19]. The advantages of reduced numerical and algebraic computations are also preserved with the use of a multiple step algorithm, since the differential equations need not be solved for each iteration on  $\{X\}$  for the gradient and Hessian calculations.

Table 7. Single Step Integration Method; Quadratic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	0%	0.50	0.50	24.98
	10%	0.37	0.62	24.99
	20%	7.11	0.93	25.18
Exact		0.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.49	0.50	24.98
	10%	5.73	0.62	25.07
	20%	9.25	0.93	25.35
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	4.98	0.50	24.98
	10%	8.30	0.62	25.18
	20%	11.87	0.92	25.59
Exact		5.00	0.50	25.00

Table 8. Single Step Integration Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$
Initial		0.50	1.00	0.35	22.50
	0%	0.10	0.50	0.50	25.00
	10%	3.07	1.34	0.62	24.72
	20%	5.11	7.45	0.93	23.02
Exact		0.10	0.50	0.50	25.00
Initial		1.00	1.00	0.35	22.50
	0%	0.50	2.49	0.50	25.00
	10%	3.53	3.79	0.62	24.57
	20%	5.65	10.61	0.94	22.70
Exact		0.50	2.50	0.50	25.00
Initial		1.00	2.50	0.35	22.50
	0%	2.49	4.98	0.50	24.97
	10%	5.64	6.98	0.62	24.41
	20%	8.03	14.80	0.93	22.34
Exact		2.50	5.00	0.50	25.00



Table 9. Single Step Integration Method; Drag Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	0%	0.50	0.50	24.98
	10%	0.61	0.52	25.94
	20%	0.48	1.03	27.32
Exact		0.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.54	0.47	24.97
	10%	2.54	0.68	26.62
	20%	2.11	1.58	29.10
Exact		2.50	0.50	25.00
Initial		3.50	0.35	22.50
	0%	5.31	0.36	24.86
	10%	5.25	0.69	27.05
	20%	4.54	1.87	30.44
Exact		5.00	0.50	25.00

Table 10. Single Step Integration Method; Quadratic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
	-----	-----	-----	-----
Initial		1.00	0.35	22.50
	0%	0.50	0.50	24.98
	10%	0.34	0.52	25.10
	20%	0.27	0.58	25.12
Exact		0.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.49	0.50	24.98
	10%	2.37	0.52	25.09
	20%	2.33	0.58	25.10
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	4.98	0.50	24.98
	10%	4.91	0.52	25.08
	20%	4.90	0.58	25.08
Exact		5.00	0.50	25.00

Table 11. Single Step Integration Method; Quadratic, Cubic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$
Initial		0.50	1.00	0.35	22.50
	0%	0.10	0.50	0.50	24.98
	10%	-0.10	0.65	0.52	25.10
	20%	-0.26	1.03	0.58	24.96
Exact		0.10	0.50	0.50	25.00
Initial		1.00	1.00	0.35	22.50
	0%	0.50	2.49	0.50	24.97
	10%	0.28	2.87	0.52	24.99
	20%	0.08	3.59	0.58	24.78
Exact		0.50	2.50	0.50	25.00
Initial		1.00	2.50	0.35	22.50
	0%	2.49	4.98	0.50	24.97
	10%	2.30	5.72	0.52	24.87
	20%	2.09	7.03	0.58	24.48
Exact		2.50	5.00	0.50	25.00

Table 12. Single Step Integration Method; Drag Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
	-----	-----	-----	-----
Initial		1.00	0.35	22.50
	0%	0.50	0.50	24.98
	10%	0.89	0.03	24.72
	20%	0.80	0.22	24.74
Exact		0.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.54	0.47	24.97
	10%	3.18	-0.20	24.32
	20%	3.10	-0.01	24.32
Exact		2.50	0.50	25.00
Initial		3.50	0.35	22.50
	0%	5.31	0.36	24.86
	10%	5.96	-0.26	24.30
	20%	5.86	-0.05	24.33
Exact		5.00	0.50	25.00

## **4.0 Multiple Step Integration Iterative Method :**

### **Single Degree of Freedom Systems**

For pure input values of the quantity  $u_m$ , it was found in Chapter 3 that the single step integration iterative method gave excellent results, but that the results obtained for corrupted input values were unreasonable for many cases. In order to use a solution method of higher order than that of equation [3.3] to find the objective function of equation [3.6] and its derivatives, a great deal more algebraic work must be supplied because of the nested function evaluations and their derivatives. As pointed out in Chapter 3, the corrupted input values to the algorithm lead to unacceptable results primarily because only one point and its function evaluations are utilized at each step of the objective function. Since the objective function depends on just one point and the function evaluations there, the fluctuations in the input quantity and function evaluations causes the Taylor expansion basis of the Runge-Kutta method to be inaccurate (see Ref. [20]). In order to circumvent these problems, a linear multistep method of the type presented by Stoer and Bulirsch [19] may be implemented rather than the one step method of Chapter 3. Multistep methods depend on the measured values of the input

quantity in a way such as to average the effects of corrupted data. Since the methods give a difference formula for each point  $\{y\}^{i+1}$  which depends on a number of past points, the effects of rapidly fluctuating data may be reduced. The methods also provide a linear combination of function evaluations which may be differentiated analytically in a simple manner. Therefore, by redefining the objective function of equations [3.5] and [3.6], the effects of noise and the effort involved in the use of higher order approximations to the solution of the differential equations may both be reduced.

### 4.1 Objective Function Formulation

The use of linear multistep methods for the solution of differential equations involves finding an approximation to the solution of the equations at a point  $\{y\}^{i+1}$  via an equation involving a linear combination of function evaluations at the points  $\{y^{i+1}, y^i, y^{i-1}, \dots, y^{i-n}\}$  for an  $n + 2$  point method. The point  $\{y\}^{i+1}$  is included in the linear combination for the case of an implicit  $n + 2$  point method. The general form for a linear multistep method is

$$\begin{aligned} \{y\}^{i+1} = & a_{n-1}\{y\}^i + a_{n-2}\{y\}^{i-1} + \dots + a_0\{y\}^{i+1-m} \\ & + h\{b_m F(t^{i+1}, \{y\}^{i+1}) + b_{m-1} F(t^i, \{y\}^i), + \dots + b_0 F(t^{i+1}, \{y\}^{i+1-m})\} \end{aligned} \tag{4.1}$$

where the a's and b's are constants chosen for the particular method. In order to implement such a method for the solution of a differential equation, initial values must be used as well as all previous values of  $\{y\}$  needed to complete the function evaluations. A single step method is normally used to start the solution process until the method can generate the  $\{y\}$  values needed to continue in the solution process. As in the formu-

lation of the objective function in the use of single step methods, for each point  $\{y\}^{i+1}$ , equation [4.1] may be viewed as a difference between the present value of the measured quantity and a combination of previous values of  $\{y\}$ , the functions  $F$  evaluated for these values, and the parameters governing the system. Therefore, by moving the present value of the solution of the equations (measured value for  $\{y\}^{i+1}$ ) to the right side of equation [4.1] and representing the measured values for  $\{y\}$  as  $\{\bar{y}\}$ , an error vector may be formulated at each time point  $t^{i+1}$  of the form

$$\{E\} = -\{\bar{y}\}^{i+1} + a_{n-1}\{\bar{y}\}^i + a_{n-2}\{\bar{y}\}^{i-1} + \dots + a_0\{\bar{y}\}^{i+1-m} + h\{b_m F(t^{i+1}, \{X\}, \{\bar{y}\}^{i+1}) + b_{m-1} F(t^i, \{X\}, \{\bar{y}\}^i) + \dots + b_0 F(t^{i+1}, \{x\}, \{\bar{y}\}^{i+1-m})\} \quad [4.2]$$

For the single degree of freedom examples of Chapters 2 and 3, the error vector is comprised of the components for measured time series of velocity and displacement. An error function is thus formulated analogous to equation [3.5]. The general form of the objective function is that of equation [3.6], but the error functions  $E_1$  and  $E_2$  take on simpler forms and only involve linear combinations of the function evaluations, as in equation [4.2] above.

## 4.2 Objective Function Differentiation

For the differentiation of the objective function of this method, since the error functions  $E_1$  and  $E_2$  are linear combinations of the function evaluations, the derivatives of the functions with respect to the parameters must only be calculated once analytically, and evaluated for various values of  $\{\bar{y}\}$ . The gradient and Hessian expressions for this method are the same as equations [3.7] and [3.8], with differences only in the formu-

lations of the derivatives of  $\{E\}$  with respect to the parameters. For the parameter  $x_i$ , the derivative of  $E_1$  takes the form

$$\begin{aligned} \frac{\partial E_1}{\partial x_i} = h \left\{ b_m \frac{\partial f_1}{\partial x_i} (t^{i+1}, \{X\}, \{\bar{y}\}^{i+1}) + b_{m-1} \frac{\partial f_1}{\partial x_i} (t^i, \{X\}, \{\bar{y}\}^i) \right. \\ \left. + \dots + b_0 \frac{\partial f_1}{\partial x_i} (t^{i+1}, \{X\}, \{\bar{y}\}^{i+1-m}) \right\} \end{aligned} \quad [4.3]$$

Note that the analytical derivative of  $f_1$  with respect to  $x_i$  is used repeatedly, but evaluated at different times and  $\{\bar{y}\}$  values. This suggests the use of a subfunction in a computer program that simply computes the derivatives needed, given the value for time and the measured values for  $\{\bar{y}\}$ . The derivatives of  $E_2$  are computed in the same way, with the equation for the derivative of  $E_2$  with respect to  $x_i$  given as

$$\begin{aligned} \frac{\partial E_2}{\partial x_i} = h \left\{ b_m \frac{\partial f_2}{\partial x_i} (t^{i+1}, \{X\}, \{\bar{y}\}^{i+1}) + b_{m-1} \frac{\partial f_2}{\partial x_i} (t^i, \{X\}, \{\bar{y}\}^i) \right. \\ \left. + \dots + b_0 \frac{\partial f_2}{\partial x_i} (t^{i+1}, \{X\}, \{\bar{y}\}^{i+1-m}) \right\} \end{aligned} \quad [4.4]$$

### 4.3 Objective Function Minimization

As in the one step iterative method, the classical Newton-Raphson method may be implemented here, since the second derivatives of the error functions may be calculated analytically in a simple manner. The second derivatives for this procedure are simpler to compute than those of the one step method, since the error functions involve only linear combinations of the function evaluations as opposed to the nested function evaluations found in the  $\{\phi\}$  functions of single step methods. The first and second deriva-



tives are computed at each point  $\{y\}^{i+1}$  as in the single step method, and the integrals for the Hessian and gradient are approximated as in equations [3.7] and [3.8]. Again, the integral expressions of these equations are not exactly correct, since the integrands are not continuous functions, but are sets of discrete values.

The error functions formulated above are linear combinations of the function  $\{F\}$ , and are evaluated for various values of  $\{y\}$ . The notable difference between this and the single step method is that in the single step method, the derivatives with respect to the parameters are of nested function values, and therefore require use of the chain rule for implicit dependence on the parameters. For this procedure, only simple derivatives are required of the function  $\{F\}$ , and since the function is usually linear in the parameters, the second derivatives are identically zero in many cases. Therefore the Hessian is comprised for the most part of first derivatives of  $\{F\}$  in product pairs. Since the objective function is quadratic in the parameters and the Newton-Raphson method is a quadratic method, convergence to the exact estimates is expected in just one iteration on  $\{X\}$ . Once the integrals for the Hessian and gradient have been approximated using a quadrature formula such as Simpson's rule, the system of equations [2.9] must again be solved for the correction vector. The iterative process is terminated when the same convergence criterion of the iterative direct and single step methods has been met.

## ***4.4 Applications of the Multistep Method***

The examples presented in Chapters 2 and 3 are again presented using the multistep method, in order to make a comparison of the three methods. The time record length

is again five seconds, divided into 335 points at which to find the function and derivative values. A six step method, presented in Reference [19], was chosen to approximate the solution of the governing differential equation, as required in equation [4.1]. In vector form, the equation for the six step method is the following:

$$\begin{aligned} \{y\}' = \{y\}^{i-6} + \frac{6h}{840} \{ & 41F(\{y\}^{i-6}) + 216F(\{y\}^{i-5}) + 27F(\{y\}^{i-4}) \\ & + 272F(\{y\}^{i-3}) + 27F(\{y\}^{i-2}) + 216F(\{y\}^{i-1}) + 41F(\{y\}^i) \} \end{aligned} \quad [4.5]$$

Note that the quantity  $\{y\}'$  appears on both sides of the solution approximation equation. Normally, to solve a system of differential equations, this equation would be used in a predictor- corrector algorithm as the corrector step, since an initial value for  $\{y\}'$  is needed to calculate a new value for  $\{y\}'$ . The values of  $\{y\}$  required for the first difference between the present point and function evaluations of past points of the solution are given as measured values. Therefore, rather than "start" the multistep method by generating values needed for the first difference point by use of a single step method of solution, the first 6 points have been dropped from use in the difference formulas used in the objective function.

In order to use the Newton-Raphson method in conjunction with the sixth order method above, the first derivative of the error function with respect to  $x_i$  must be computed, and is given as

$$\begin{aligned} \frac{\partial}{\partial x_i} \{E\} = \frac{6h}{840} \left\{ & 41 \frac{\partial}{\partial x_i} F_{i-6} + 216 \frac{\partial}{\partial x_i} F_{i-5} + 27 \frac{\partial}{\partial x_i} F_{i-4} + \right. \\ & \left. 272 \frac{\partial}{\partial x_i} F_{i-3} + 27 \frac{\partial}{\partial x_i} F_{i-2} + 216 \frac{\partial}{\partial x_i} F_{i-1} + 41 \frac{\partial}{\partial x_i} F_i \right\} \end{aligned} \quad [4.6]$$

where the subscripts denote F evaluated at at various values of  $\{y\}$ .

### 4.4.1 Quadratic Nonlinearity Example

The governing differential equation for this example is presented as equation [2.15], and is again transformed to the function {F} as in equation [3.9] for the single step method. Given the overall objective function of the examples as equation [3.6] and the error function {E} from equation [4.5], the derivatives of the function vector used in the derivatives of {E} for this case are given as

$$\frac{\partial f_1}{\partial x_i} = 0 \quad i = 1, 2, 3$$

$$\frac{\partial f_2}{\partial x_1} = -\bar{y}_1^k \quad [4.7]$$

$$\frac{\partial f_2}{\partial x_2} = -2\bar{y}_2^k$$

$$\frac{\partial f_2}{\partial x_3} = -\bar{y}_1^{k^2}$$

where k denotes the measured value point number, to be used in the evaluations of the derivatives. Note that for this case, all second derivatives of {F} with respect to the parameters are identically zero. This greatly simplifies the expression for the Hessian of the objective function.

The quantities needed for the gradient and Hessian expressions are thus calculated, and the integral expressions are approximated, using Simpson's rule as in Chapters 2 and 3. The system of equations [2.9] is then solved for the correction vector {ΔX}. The new

value for  $\{X\}$  is then found using equation [2.7]. The new values for  $\{X\}$  are then used in the derivative equations and the iterative process repeated. Note that the first derivative equations for this example do not depend on the parameters, which leads to extremely fast convergence of the Newton-Raphson method.

### 4.4.2 Quadratic and Cubic Nonlinearity Example

The governing differential equation for this method is presented as equation [2.20], and is transformed to the function  $\{F\}$  as in equation [3.14] for the single step method. The derivatives of  $\{F\}$  are computed as in the quadratic nonlinearity example, and take the similar form

$$\frac{\partial f_1}{\partial x_i} = 0 \quad i = 1, 2, 3, 4$$

$$\frac{\partial f_2}{\partial x_1} = -\bar{y}_1^k$$

$$\frac{\partial f_2}{\partial x_2} = -2\bar{y}_2^k \quad [4.8]$$

$$\frac{\partial f_2}{\partial x_3} = -\bar{y}_1^{k^2}$$

$$\frac{\partial f_2}{\partial x_4} = -\bar{y}_1^{k^3}$$

Note that the only difference between this example and the preceding one is the addition of the fourth parameter. The derivatives formulated for this example and the quadratic nonlinearity example are more similar than the derivatives of the two corresponding single step method examples. This is because of the linear dependence of the error functions on {F} and its parameters. The second derivatives of {F} are identically zero as in the quadratic nonlinearity, because {F} is linear in all the parameters. Therefore, the Hessian expression is composed of pairs of multiplied first derivatives; the first and third terms in equation [3.8]. The iterative process continues as in equations [2.7] to [2.9] until the convergence criterion of Chapters 2 and 3 is met.

### 4.4.3 Drag Nonlinearity Example

Again, the equations describing the vibrations are of the same form as equations [2.25] and [3.19]. The derivatives needed for implementation of the method are

$$\frac{\partial f_1}{\partial x_i} = 0 \quad i = 1, 2, 3$$

$$\frac{\partial f_2}{\partial x_1} = -\bar{y}_1^k \quad [4.9]$$

$$\frac{\partial f_2}{\partial x_2} = -2\bar{y}_2^k$$

$$\frac{\partial f_2}{\partial x_3} = -\bar{y}_2^k |y_2^k|$$

Note that for this example, no assumptions need be made about any quantity taking on the exact value of zero, as in the derivative formulations of Chapters 2 and 3. Again, the second derivatives of  $\{F\}$  vanish as in the above examples. The iterative process follows exactly as in the examples presented above. Rapid convergence to the parameters is again expected (1 iteration), since the second derivatives of the functions  $\{F\}$  vanish.

## ***4.5 Results***

The use of the multiple step integration procedure on the three examples yielded improved results from the single step method, while preserving the desirable qualities of the single step procedure. Since the multiple step method depends on a number of previous points and function evaluations at those points, the effects of rapidly fluctuating data have been reduced, leading to more accurate final estimates of the system parameters. The RMS values used in the noise calculations were the same as in the examples of Chapter 3. The results of the three examples are compiled in Tables 13, 14, and 15.

For the quadratic nonlinearity case, the final parameter estimates were in general better than those from the single step method. The worst parameter estimate is that for the nonlinearity coefficient in the 0.5 value case, with 20% added noise. A 240% error for this parameter was found, though the other parameter estimates for this case were only in error by approximately 5%. For the other cases, reasonable results were obtained,

with final estimates slightly less accurate than those obtained using the iterative direct method.

The drag and quadratic and cubic nonlinear examples yielded results similar to those obtained for the quadratic example, with the results of the drag nonlinearity example comparable to the corresponding numbers generated using the iterative direct method. The dramatic improvement in estimates using this method over those from the single step method is due to a dependence of the error function  $\{E\}$  at each point on the previous six points and their function evaluations, rather than on only one previous value, which in most cases does not represent an accurate solution value of the differential equation. For pure input values, the method yielded exact values for the parameter estimates in all cases, as opposed to the nearly correct estimates of the single step method.

After filtering, the 240% error discussed above was reduced to 62%. Though this still represents a sizeable error, the other values found after filtering are in very good agreement with the actual parameter values, as in Table 16. For the drag and quadratic and cubic nonlinearity examples, the results after filtering are mixed, with some estimates improving while others actually decrease in accuracy, as is the case for the single step and iterative direct methods (see Tables 17 and 18). The three cases graphed in Chapter 2 are also plotted in Figures 10 through 15 for results found using the multiple step integration method. Note the improvements mentioned above for the first two examples with filtered input data, and the worsening of fit for the filtered data used in the drag nonlinearity example. The parameter estimates have been substituted back into the governing differential equations and the responses of the systems for the given initial conditions plotted. Note that no model for the response exists, as in the iterative direct

method for  $u_r$ . For this method, the model of the system to be identified is simply the governing differential equation.

The number of iterations required for convergence in all cases was only two, meaning that the method had actually found the final values in one to two iterations, since the last iteration gives values of the parameters which are only different from the previous estimates by  $5.0E-2\%$ . This is probably because of the linear dependence of the  $\{E\}$  components on the parameters for these examples. The amount of CPU time required for all three examples for all cases was less than one second.



Table 13. Multistep Integration Method; Quadratic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	0%	0.50	0.50	25.00
	5%	0.67	0.51	24.92
	10%	0.85	0.52	24.84
	20%	1.21	0.53	24.63
Exact		0.50	0.50	25.00
Initial		0.50	0.35	22.50
	0%	1.50	0.50	25.00
	5%	1.68	0.51	24.93
	10%	1.85	0.52	24.85
	20%	2.21	0.54	24.66
Exact		1.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.50	0.50	25.00
	5%	2.68	0.51	24.94
	10%	2.86	0.52	24.87
	20%	3.21	0.54	24.69
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	5.00	0.50	25.00
	5%	5.18	0.51	24.96
	10%	5.36	0.52	24.91
	20%	5.71	0.54	24.77
Exact		5.00	0.50	25.00

Table 14. Multistep Integration Method; Quadratic, Cubic Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$
Initial		0.50	1.00	0.35	22.50
	0%	0.10	0.50	0.50	25.00
	10%	0.38	0.83	0.52	24.73
	20%	0.76	0.81	0.54	24.52
Exact		0.10	0.50	0.50	25.00
Initial		0.50	1.00	0.35	22.50
	0%	1.00	2.50	0.50	25.00
	10%	1.23	3.20	0.52	24.64
	20%	1.56	3.54	0.54	24.33
Exact		1.00	2.50	0.50	25.00
Initial		1.00	1.00	0.35	22.50
	0%	0.50	2.50	0.50	25.00
	10%	0.74	3.16	0.52	24.64
	20%	1.07	3.44	0.54	24.34
Exact		0.50	2.50	0.50	25.00
Initial		0.50	2.50	0.35	22.50
	0%	1.00	5.00	0.50	25.00
	10%	1.19	6.02	0.52	24.56
	20%	1.47	6.64	0.54	24.17
Exact		1.00	5.00	0.50	25.00
Initial		1.00	2.50	0.35	22.50
	0%	2.50	5.00	0.50	25.00
	10%	2.70	6.08	0.52	24.56
	20%	3.00	6.78	0.54	24.17
Exact		2.50	5.00	0.50	25.00

Table 15. Multistep Integration Method; Drag Numerical Results, Unfiltered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	0%	0.50	0.50	25.00
	5%	0.53	0.48	24.98
	10%	0.56	0.46	24.94
	20%	0.62	0.41	24.83
Exact		0.50	0.50	25.00
Initial		0.50	0.35	22.50
	0%	1.50	0.50	25.00
	5%	1.53	0.49	25.00
	10%	1.57	0.47	24.99
	20%	1.66	0.42	24.91
Exact		1.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.50	0.50	25.00
	5%	2.54	0.49	25.02
	10%	2.58	0.48	25.03
	20%	2.67	0.43	24.99
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	5.00	0.50	25.00
	5%	5.03	0.50	25.07
	10%	5.08	0.48	25.12
	20%	5.19	0.44	25.15
Exact		5.00	0.50	25.00

Table 16. Multistep Integration Method; Quadratic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha u^2 = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	5%	0.50	0.50	25.00
	5%	0.42	0.49	25.06
	10%	0.34	0.49	25.12
	20%	0.19	0.49	25.22
Exact		0.50	0.50	25.00
Initial		0.50	0.35	22.50
	0%	1.50	0.50	25.00
	5%	1.43	0.49	25.05
	10%	1.36	0.49	25.11
	20%	1.23	0.49	25.21
Exact		1.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.50	0.50	25.00
	5%	2.44	0.49	25.05
	10%	2.38	0.49	25.10
	20%	2.26	0.49	25.20
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	5.00	0.50	25.00
	5%	4.96	0.49	25.04
	10%	4.92	0.49	25.09
	20%	4.84	0.49	25.17
Exact		5.00	0.50	25.00

Table 17. Multistep Integration Method; Quadratic, Cubic Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha_1 u^2 + \alpha_2 u^3 = 0$$

$$u(t=0) = 0$$

$$\dot{u}(t=0) = 5$$

	Noise	$\alpha_1$	$\alpha_2$	$\gamma$	$\Omega^2$
Initial		0.50	1.00	0.35	22.50
	0%	0.10	0.50	0.50	25.00
	10%	-0.19	1.00	0.49	24.97
	20%	-0.46	1.46	0.49	24.94
Exact		0.10	0.50	0.50	25.00
Initial		1.00	1.00	0.35	22.50
	0%	0.50	2.50	0.50	25.00
	10%	0.22	3.16	0.49	24.91
	20%	-0.05	3.78	0.49	24.83
Exact		0.50	2.50	0.50	25.00
Initial		0.50	1.00	0.35	22.50
	0%	1.00	2.50	0.50	25.00
	10%	0.73	3.18	0.49	24.90
	20%	0.47	3.82	0.49	24.81
Exact		1.00	2.50	0.50	25.00
Initial		0.50	2.50	0.35	22.50
	0%	1.00	5.00	0.50	25.00
	10%	0.74	5.84	0.49	24.86
	20%	0.49	6.63	0.49	24.71
Exact		1.00	5.00	0.50	25.00
Initial		1.00	2.50	0.35	22.50
	0%	2.50	5.00	0.50	25.00
	10%	2.28	5.89	0.49	24.83
	20%	2.07	6.72	0.49	24.67
Exact		2.50	5.00	0.50	25.00

Table 18. Multistep Integration Method; Drag Numerical Results, Filtered Input Data

$$\ddot{u} + 2\gamma\dot{u} + \Omega^2 u + \alpha\dot{u}|\dot{u}| = 0$$

$$u(t = 0) = 0$$

$$\dot{u}(t = 0) = 5$$

	Noise	$\alpha$	$\gamma$	$\Omega^2$
Initial		1.00	0.35	22.50
	0%	0.50	0.50	25.00
	5%	0.42	0.57	25.06
	10%	0.41	0.59	25.09
	20%	0.39	0.61	25.13
Exact		0.50	0.50	25.00
Initial		0.50	0.35	22.50
	0%	1.50	0.50	25.00
	5%	1.30	0.65	25.15
	10%	1.29	0.66	25.16
	20%	1.28	0.67	25.16
Exact		1.50	0.50	25.00
Initial		1.50	0.35	22.50
	0%	2.50	0.50	25.00
	5%	2.13	0.74	25.30
	10%	2.12	0.74	25.29
	20%	2.11	0.75	25.26
Exact		2.50	0.50	25.00
Initial		2.50	0.35	22.50
	0%	5.00	0.50	25.00
	5%	4.00	1.00	25.80
	10%	4.00	1.00	25.77
	20%	4.01	1.00	25.71
Exact		5.00	0.50	25.00

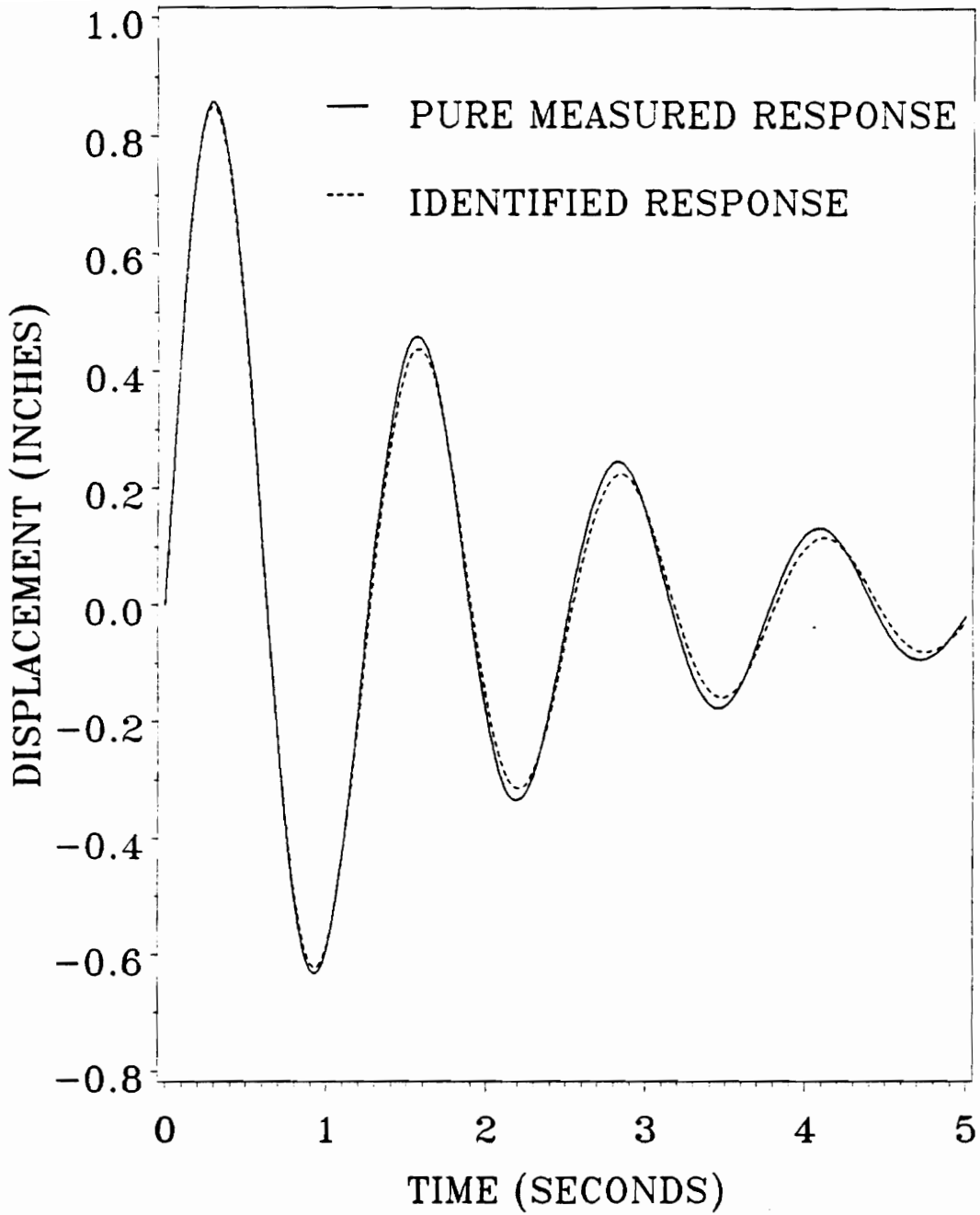


Figure 10. Identified/Measured Response of Quadratic Nonlinearity Example: Multistep integration method; 20% corrupted input data, unfiltered,  $\alpha = 0.5$

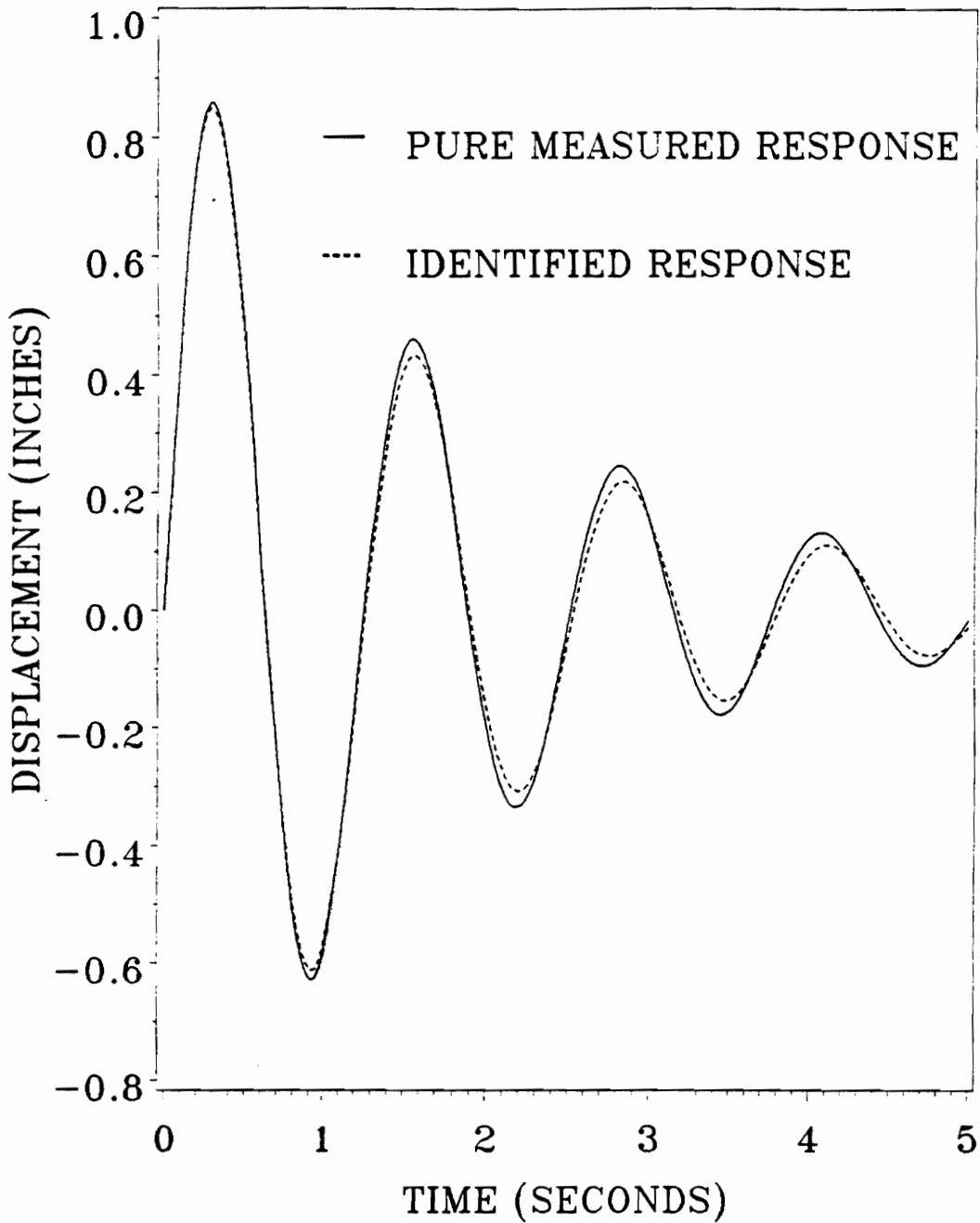


Figure 11. Identified/Measured Response of Quadratic and Cubic Nonlinearity  
 Example: Multistep integration method; 20% corrupted input data, unfiltered ,  
 $\alpha_1 = 0.1$ ,  $\alpha_2 = 0.5$



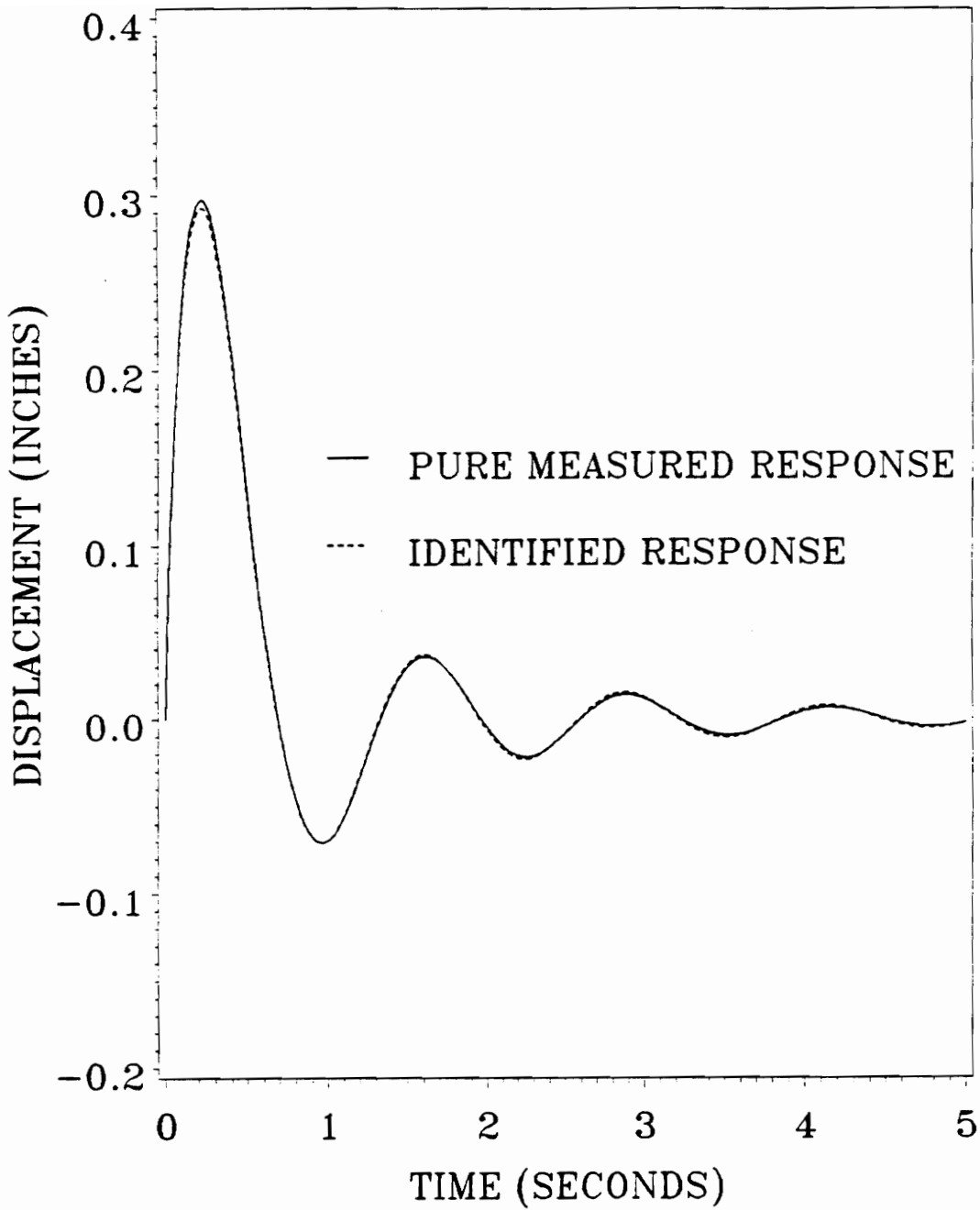


Figure 12. Identified/Measured Response of Drag Nonlinearity Example: Multistep integration method; 20% corrupted input data, unfiltered,  $\alpha = 5.0$

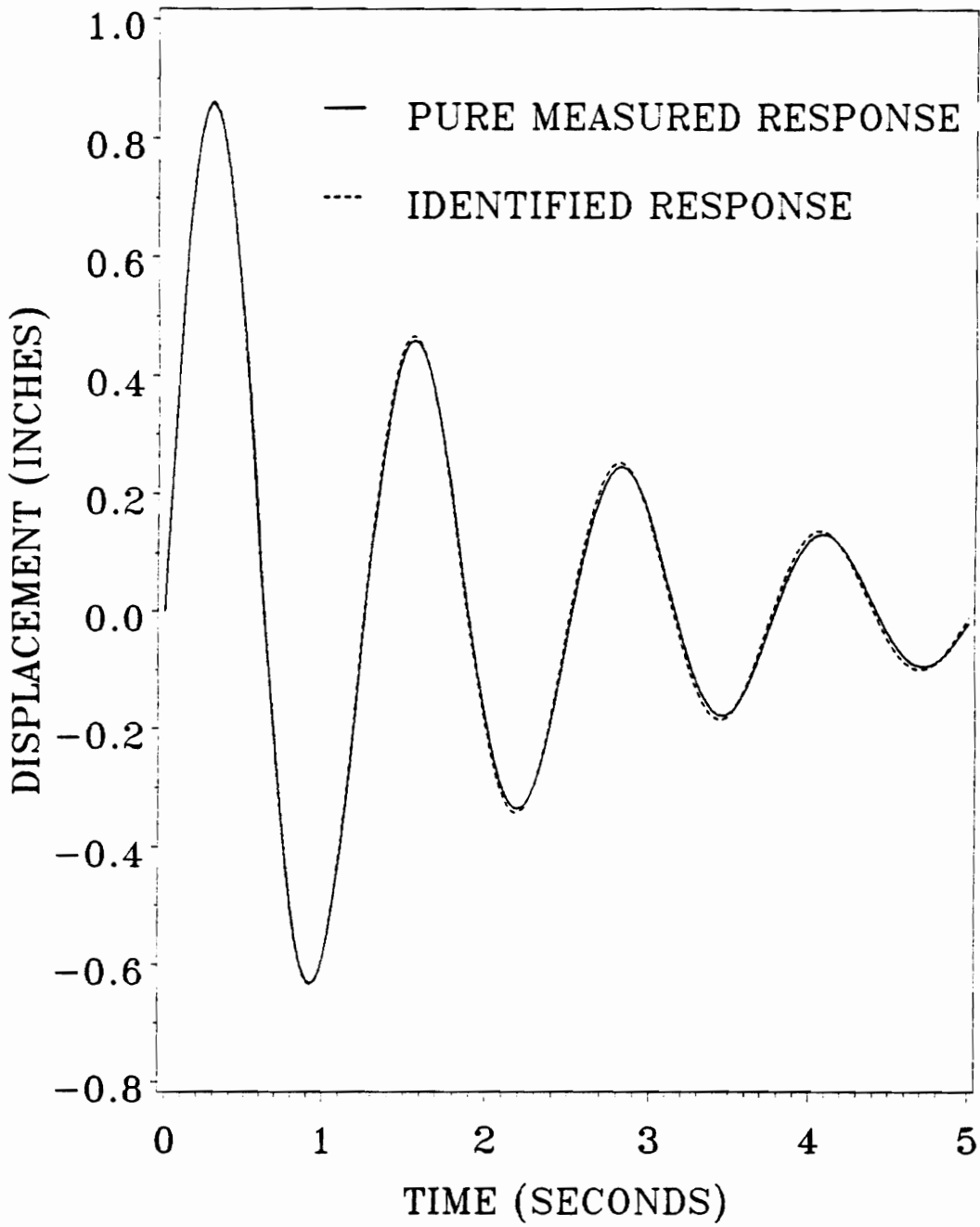


Figure 13. Identified/Measured Response of Quadratic Nonlinearity Example: 20% Multi-step integration method; corrupted input data, filtered,  $\alpha = 0.5$

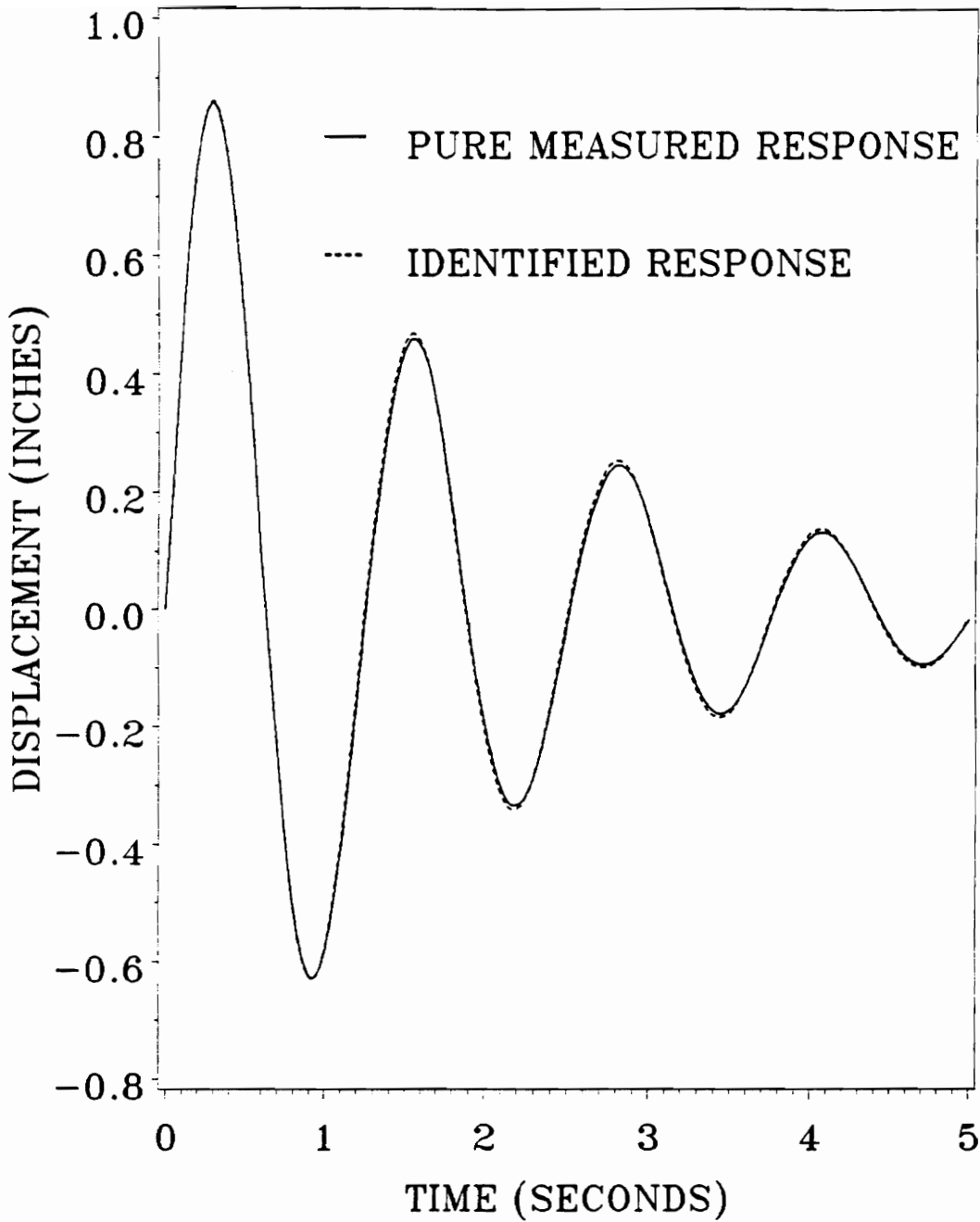


Figure 14. Identified/Measured Response of Quadratic and Cubic Nonlinearity  
 Example: Multistep integration method; 20% corrupted input data, filtered ,  
 $\alpha_1 = 0.1$ ,  $\alpha_2 = 0.5$

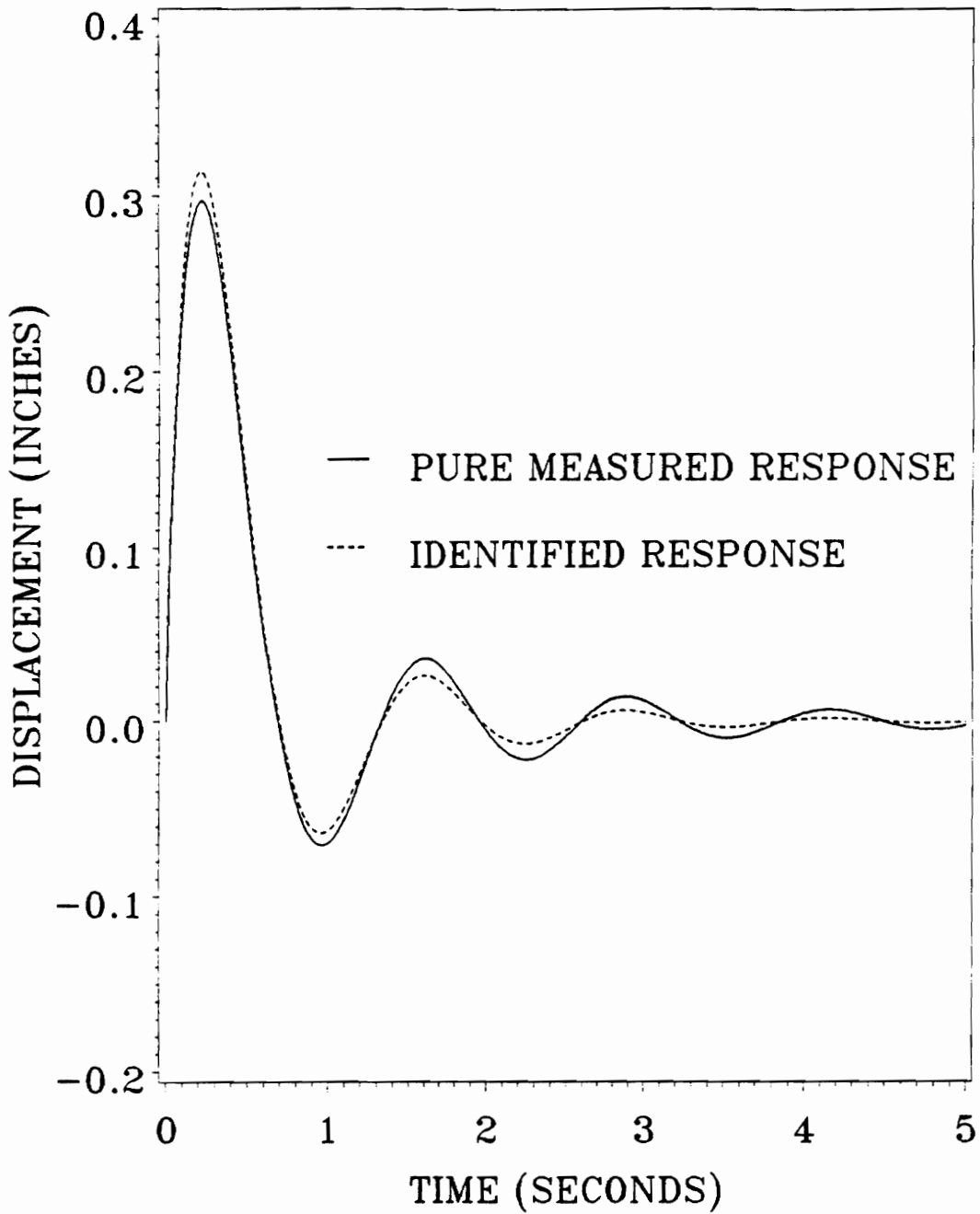


Figure 15. Identified/Measured Response of Drag Nonlinearity Example: Multistep integration method; 20% corrupted input data, filtered,  $\alpha = 5.0$

## **5.0 Iterative Direct Method: Two Degrees of Freedom System**

The extension of the iterative direct method to two degrees of freedom systems is straightforward, and in this Chapter the method will be used to identify a simple two degrees of freedom system with a cubic nonlinearity. The only differences between the use of the method on one and two degrees of freedom systems are in the formulation of the objective function, and the increased number of differential equations which must be solved for each iteration performed to obtain the parameter values. As in the single degree of freedom examples, the first step in the analysis is to formulate an objective function which depends on the system parameters.

## 5.1 Objective Function Formulation

The formulation of the objective function follows directly from the single degree of freedom examples, and is given as

$$L\{X\} = \int_0^{T^*} \{(u_{1m} - u_{1a})^2 + (u_{2m} - u_{2a})^2\} dt \quad [5.1]$$

where the first subscript denotes the node where one of the quantities of displacement, velocity, or acceleration is measured. The objective function is dependent on the vector  $\{X\}$  as in the single degree of freedom examples, where  $\{X\}$  is the vector of parameters describing the system, including damping, natural frequency, and nonlinearity coefficients. The object of the procedure is to minimize the objective function by fitting the analytical quantities  $u_{1a}$  and  $u_{2a}$  to the measured quantities  $u_{1m}$  and  $u_{2m}$  by making corrections to the parameter vector  $\{X\}$ . Once the objective function has been minimized with respect to  $\{X\}$ , the system is said to be identified, and accurate predictions of the response of the system may be made. Minimization of the objective function requires the computation of derivatives of  $L\{X\}$  with respect to the parameters.

## 5.2 Objective Function Differentiation

The quantities  $u_{1a}$  and  $u_{2a}$  of equation [5.1] depend on the parameters  $\{X\}$ , and must be differentiated in order to minimize the value of the objective function. The measured

quantities of the objective function are independent of the parameters  $\{X\}$ , giving for the first derivative of  $L\{X\}$  with respect to  $x_i$

$$\frac{\partial L\{X\}}{\partial x_i} = -2 \int_0^{T^*} \left\{ (u_{1m} - u_{1a}) \left( \frac{\partial u_{1a}}{\partial x_i} \right) + (u_{2m} - u_{2a}) \left( \frac{\partial u_{2a}}{\partial x_i} \right) \right\} dt \quad [5.2]$$

The expression analogous to the approximation to the Hessian in equations [2.13] takes the form

$$N_{kl} = 2 \int_0^{T^*} \left\{ \left( \frac{\partial u_{1a}}{\partial x_k} \right) \left( \frac{\partial u_{1a}}{\partial x_l} \right) + \left( \frac{\partial u_{2a}}{\partial x_k} \right) \left( \frac{\partial u_{2a}}{\partial x_l} \right) \right\} dt \quad [5.3]$$

where the superscript  $i$  of equations [2.13] is understood, since all evaluations are for the current values of  $\{X\}$ . The first derivatives required in the expressions above are found as time series solutions of a set of differential equations defined for this purpose, along with the analytical quantities  $u_{1a}$  and  $u_{2a}$ .

### 5.3 Objective Function Minimization

The iterative direct method is used in conjunction with the Levenberg-Marquardt method to minimize the objective function of equation [5.1]. The equations used in the iterative process are the same as equations [2.12] and [2.13], with the exception of the

gradient and Hessian expressions as given above. In general, the number of parameters sought will be larger for a two degrees of freedom system than for a one degree of freedom system. Therefore, the order of the Hessian and the dimension of the gradient vector will also be larger. The process of updating the vector  $\{X\}$  is again implemented, with iterations terminated when a convergence criterion is met.

## 5.4 Cubic Nonlinear Two Degrees of Freedom System

The iterative direct method of identification presented in Chapter 2 is applied here to a simple two degrees of freedom system, presented by Nayfeh [23], of the form

$$\ddot{u}_1 = -\Omega_1^2 u_1 - 2\gamma_1 \dot{u}_1 - \alpha_1 u_1^3 - \alpha_2 u_1^2 u_2 - \alpha_3 u_1 u_2^2 - \alpha_4 u_2^3 \quad [5.4]$$

$$\ddot{u}_2 = -\Omega_2^2 u_2 - 2\gamma_2 \dot{u}_2 - \alpha_5 u_1^3 - \alpha_6 u_1^2 u_2 - \alpha_7 u_1 u_2^2 - \alpha_8 u_2^3$$

The procedure outlined in Chapter 2 is again followed for this example, with the exception that the number of parameters has increased to 12, with the parameter vector  $\{X\}$  given as  $\{X\} = \{\Omega_1^2, \gamma_1, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \Omega_2^2, \gamma_2, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}^T$ . Note that for the sake of simplicity, for this example, the initial conditions have not been treated as unknown parameters as in the single degree of freedom examples using the iterative direct method.

In order to use the method, differential equations analogous to equations [2.17] must be set up for the two degrees of freedom cubic nonlinear system above. First, define the quantities



$$\begin{array}{ll}
 y_1 = u_1 & y_{27} = u_2 \\
 y_2 = u_{1,t} & y_{28} = u_{2,t} \\
 y_3 = u_{1,x_1} & y_{29} = u_{2,x_1} \\
 y_4 = u_{1,x_1,t} & y_{30} = u_{2,x_1,t} \\
 y_5 = u_{1,x_2} & y_{31} = u_{2,x_2} \\
 y_6 = u_{1,x_2,t} & y_{32} = u_{2,x_2,t} \\
 y_7 = u_{1,x_3} & y_{33} = u_{2,x_3} \\
 y_8 = u_{1,x_3,t} & y_{34} = u_{2,x_3,t} \\
 y_9 = u_{1,x_4} & y_{35} = u_{2,x_4} \\
 y_{10} = u_{1,x_4,t} & y_{36} = u_{2,x_4,t} \\
 y_{11} = u_{1,x_5} & y_{37} = u_{2,x_5} \\
 y_{12} = u_{1,x_5,t} & y_{38} = u_{2,x_5,t} \\
 y_{13} = u_{1,x_6} & y_{39} = u_{2,x_6} \\
 y_{14} = u_{1,x_6,t} & y_{40} = u_{2,x_6,t} \\
 y_{15} = u_{1,x_7} & y_{41} = u_{2,x_7}
 \end{array}$$

[5.5]

$$y_{16} = u_{1,x_7,t} \quad y_{42} = u_{2,x_7,t}$$

$$y_{17} = u_{1,x_8} \quad y_{43} = u_{2,x_8}$$

$$y_{18} = u_{1,x_8,t} \quad y_{44} = u_{2,x_8,t}$$

$$y_{19} = u_{1,x_9} \quad y_{45} = u_{2,x_9}$$

$$y_{20} = u_{1,x_9,t} \quad y_{46} = u_{2,x_9,t}$$

$$y_{21} = u_{1,x_{10}} \quad y_{47} = u_{2,x_{10}}$$

$$y_{22} = u_{1,x_{10},t} \quad y_{48} = u_{2,x_{10},t}$$

$$y_{23} = u_{1,x_{11}} \quad y_{49} = u_{2,x_{11}}$$

$$y_{24} = u_{1,x_{11},t} \quad y_{50} = u_{2,x_{11},t}$$

$$y_{25} = u_{1,x_{12}} \quad y_{51} = u_{2,x_{12}}$$

$$y_{26} = u_{1,x_{12},t} \quad y_{52} = u_{2,x_{12},t}$$

At each step of the iterative process, the following set of 52 nonlinear differential equations must be solved, in order to obtain the time series for  $u_{1a}$ ,  $u_{2a}$  and all their first derivatives with respect to the parameters:

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -x_1 y_1 - 2x_2 y_2 - x_3 y_1^3 - x_4 y_1^2 y_2 - x_5 y_1 y_2^2 - x_6 y_2^3$$

$$\dot{y}_3 = y_4$$

$$\begin{aligned}\dot{y}_4 = & -y_1 - x_1 y_3 - 2x_2 y_4 - 3x_3 y_1^2 y_3 - 2x_4 y_1 y_2 y_3 - x_4 y_1^2 y_2 y_3 \\ & - x_5 y_3 y_2^2 - 2x_5 y_1 y_2 y_2 y_3 - 3x_6 y_2^2 y_2 y_3\end{aligned}$$

$$\dot{y}_5 = y_6$$

$$\begin{aligned}\dot{y}_6 = & -x_1 y_5 - 2y_2 - 2x_2 y_6 - 3x_3 y_1^2 y_5 - 2x_4 y_1 y_2 y_5 - x_4 y_1^2 y_3 y_5 \\ & - x_5 y_5 y_2^2 - 2x_5 y_1 y_2 y_3 y_5 - 3x_6 y_2^2 y_3 y_5\end{aligned}$$

$$\dot{y}_7 = y_8$$

$$\begin{aligned}\dot{y}_8 = & -x_1 y_7 - 2x_2 y_8 - y_1^3 - 3x_3 y_1^2 y_7 - 2x_4 y_1 y_2 y_7 - x_4 y_1^2 y_3 y_7 \\ & - x_5 y_7 y_2^2 - 2x_5 y_1 y_2 y_3 y_7 - 3x_6 y_2^2 y_3 y_7\end{aligned}$$

$$\dot{y}_9 = y_{10}$$

$$\begin{aligned}\dot{y}_{10} = & -x_1 y_9 - 2x_2 y_{10} - 3x_3 y_1^2 y_9 - y_1^2 y_{27} - 2x_4 y_1 y_2 y_9 - x_4 y_1^2 y_3 y_9 \\ & - x_5 y_9 y_2^2 - 2x_5 y_1 y_2 y_3 y_9 - 3x_6 y_2^2 y_3 y_9\end{aligned}$$

$$\dot{y}_{11} = y_{12}$$

$$\begin{aligned}\dot{y}_{12} = & -x_1 y_{11} - 2x_2 y_{12} - 3x_3 y_1^2 y_{11} - 2x_4 y_1 y_2 y_{11} - x_4 y_1^2 y_3 y_{11} \\ & - y_1 y_{27}^2 - x_5 y_{11} y_2^2 - 2x_5 y_1 y_2 y_3 y_{11} - 3x_6 y_2^2 y_3 y_{11}\end{aligned}$$

$$\dot{y}_{13} = y_{14}$$

$$\begin{aligned}\dot{y}_{14} = & -x_1 y_{13} - 2x_2 y_{14} - 3x_3 y_1^2 y_{13} - 2x_4 y_1 y_2 y_{13} - x_4 y_1^2 y_3 y_{13} \\ & - x_5 y_{13} y_2^2 - 2x_5 y_1 y_2 y_3 y_{13} - y_{27}^3 - 3x_6 y_2^2 y_3 y_{13}\end{aligned}$$

$$\dot{y}_{15} = y_{16}$$

$$\dot{y}_{16} = -x_1 y_{15} - 2x_2 y_{16} - 3x_3 y_1^2 y_{15} - 2x_4 y_1 y_{27} y_{15} - x_4 y_1^2 y_{41} \\ - x_5 y_{15} y_{27}^2 - 2x_5 y_1 y_{27} y_{41} - 3x_6 y_{27}^2 y_{41}$$

$$\dot{y}_{17} = y_{18}$$

$$\dot{y}_{18} = -x_1 y_{17} - 2x_2 y_{18} - 3x_3 y_1^2 y_{17} - 2x_4 y_1 y_{27} y_{17} - x_4 y_1^2 y_{43} \\ - x_5 y_{17} y_{27}^2 - 2x_5 y_1 y_{27} y_{43} - 3x_6 y_{27}^2 y_{43}$$

$$\dot{y}_{19} = y_{20}$$

$$\dot{y}_{20} = -x_1 y_{19} - 2x_2 y_{20} - 3x_3 y_1^2 y_{19} - 2x_4 y_1 y_{27} y_{19} - x_4 y_1^2 y_{45} \\ - x_5 y_{19} y_{27}^2 - 2x_5 y_1 y_{27} y_{45} - 3x_6 y_{27}^2 y_{45}$$

$$\dot{y}_{21} = y_{22}$$

$$\dot{y}_{22} = -x_1 y_{21} - 2x_2 y_{22} - 3x_3 y_1^2 y_{21} - 2x_4 y_1 y_{27} y_{21} - x_4 y_1^2 y_{47} \\ - x_5 y_{21} y_{27}^2 - 2x_5 y_1 y_{27} y_{47} - 3x_6 y_{27}^2 y_{47}$$

$$\dot{y}_{23} = y_{24}$$

$$\dot{y}_{24} = -x_1 y_{23} - 2x_2 y_{24} - 3x_3 y_1^2 y_{23} - 2x_4 y_1 y_{27} y_{23} - x_4 y_1^2 y_{49} \\ - x_5 y_{23} y_{27}^2 - 2x_5 y_1 y_{27} y_{49} - 3x_6 y_{27}^2 y_{49}$$

$$\dot{y}_{25} = y_{26}$$

[5.6]

$$\dot{y}_{26} = -x_1 y_{25} - 2x_2 y_{26} - 3x_3 y_1^2 y_{25} - 2x_4 y_1 y_{27} y_{25} - x_4 y_1^2 y_{51} \\ - x_5 y_{25} y_{27}^2 - 2x_5 y_1 y_{27} y_{51} - 3x_6 y_{27}^2 y_{51}$$

$$\dot{y}_{27} = y_{28}$$

$$\dot{y}_{28} = -x_7 y_{27} - 2x_8 y_{28} - x_9 y_1^3 - x_{10} y_1^2 y_{27} - x_{11} y_1 y_{27}^2 - x_{12} y_{27}^3$$

$$\dot{y}_{29} = y_{30}$$

$$\begin{aligned} \dot{y}_{30} = & -x_7 y_{29} - 2x_8 y_{30} - 3x_9 y_1^2 y_3 - 2x_{10} y_1 y_{27} y_3 - x_{10} y_1^2 y_{29} \\ & - x_{11} y_3 y_{27}^2 - 2x_{11} y_1 y_{27} y_{29} - 3x_{12} y_{27}^2 y_{29} \end{aligned}$$

$$\dot{y}_{31} = y_{32}$$

$$\begin{aligned} \dot{y}_{32} = & -x_7 y_{31} - 2x_8 y_{32} - 3x_9 y_1^2 y_5 - 2x_{10} y_1 y_{27} y_5 - x_{10} y_1^2 y_{31} \\ & - x_{11} y_5 y_{27}^2 - 2x_{11} y_1 y_{27} y_{31} - 3x_{12} y_{27}^2 y_{31} \end{aligned}$$

$$\dot{y}_{33} = y_{34}$$

$$\begin{aligned} \dot{y}_{34} = & -x_7 y_{33} - 2x_8 y_{34} - 3x_9 y_1^2 y_7 - 2x_{10} y_1 y_{27} y_7 - x_{10} y_1^2 y_{33} \\ & - x_{11} y_7 y_{27}^2 - 2x_{11} y_1 y_{27} y_{33} - 3x_{12} y_{27}^2 y_{33} \end{aligned}$$

$$\dot{y}_{35} = y_{36}$$

$$\begin{aligned} \dot{y}_{36} = & -x_7 y_{35} - 2x_8 y_{36} - 3x_9 y_1^2 y_9 - 2x_{10} y_1 y_{27} y_9 - x_{10} y_1^2 y_{35} \\ & - x_{11} y_9 y_{27}^2 - 2x_{11} y_1 y_{27} y_{35} - 3x_{12} y_{27}^2 y_{35} \end{aligned}$$

$$\dot{y}_{37} = y_{38}$$

$$\begin{aligned} \dot{y}_{38} = & -x_7 y_{37} - 2x_8 y_{38} - 3x_9 y_1^2 y_{11} - 2x_{10} y_1 y_{27} y_{11} - x_{10} y_1^2 y_{37} \\ & - x_{11} y_{11} y_{27}^2 - 2x_{11} y_1 y_{27} y_{37} - 3x_{12} y_{27}^2 y_{37} \end{aligned}$$

$$\dot{y}_{39} = y_{40}$$

$$\begin{aligned} \dot{y}_{40} = & -x_7 y_{39} - 2x_8 y_{40} - 3x_9 y_1^2 y_{13} - 2x_{10} y_1 y_{27} y_{13} - x_{10} y_1^2 y_{39} \\ & - x_{11} y_{13} y_{27}^2 - 2x_{11} y_1 y_{27} y_{39} - 3x_{12} y_{27}^2 y_{39} \end{aligned}$$

$$\dot{y}_{41} = y_{42}$$

$$\dot{y}_{42} = -y_{27} - x_7 y_{41} - 2x_8 y_{42} - 3x_9 y_1^2 y_{15} - 2x_{10} y_1 y_{27} y_{15} - x_{10} y_1^2 y_{41} \\ - x_{11} y_{15} y_{27}^2 - 2x_{11} y_1 y_{27} y_{41} - 3x_{12} y_{27}^2 y_{41}$$

$$\dot{y}_{43} = y_{44}$$

$$\dot{y}_{44} = -x_7 y_{43} - 2y_{28} - 2x_8 y_{44} - 3x_9 y_1^2 y_{17} - 2x_{10} y_1 y_{27} y_{17} - x_{10} y_1^2 y_{43} \\ - x_{11} y_{17} y_{27}^2 - 2x_{11} y_1 y_{27} y_{43} - 3x_{12} y_{27}^2 y_{43}$$

$$\dot{y}_{45} = y_{46}$$

$$\dot{y}_{46} = -x_7 y_{45} - 2x_8 y_{46} - y_1^3 - 3x_9 y_1^2 y_{19} - 2x_{10} y_1 y_{27} y_{19} - x_{10} y_1^2 y_{45} \\ - x_{11} y_{19} y_{27}^2 - 2x_{11} y_1 y_{27} y_{45} - 3x_{12} y_{27}^2 y_{45}$$

$$\dot{y}_{47} = y_{48}$$

$$\dot{y}_{48} = -x_7 y_{47} - 2x_8 y_{48} - 3x_9 y_1^2 y_{21} - y_1^2 y_{27} - 2x_{10} y_1 y_{27} y_{21} - x_{10} y_1^2 y_{47} \\ - x_{11} y_{21} y_{27}^2 - 2x_{11} y_1 y_{27} y_{47} - 3x_{12} y_{27}^2 y_{47}$$

$$\dot{y}_{49} = y_{50}$$

$$\dot{y}_{50} = -x_7 y_{49} - 2x_8 y_{50} - 3x_9 y_1^2 y_{23} - 2x_{10} y_1 y_{27} y_{23} - x_{10} y_1^2 y_{49} - y_1 y_{27}^2 \\ - x_{11} y_{23} y_{27}^2 - 2x_{11} y_1 y_{27} y_{49} - 3x_{12} y_{27}^2 y_{49}$$

$$\dot{y}_{51} = y_{52}$$

$$\dot{y}_{52} = -x_7 y_{51} - 2x_8 y_{52} - 3x_9 y_1^2 y_{25} - 2x_{10} y_1 y_{27} y_{25} - x_{10} y_1^2 y_{51} \\ - x_{11} y_{25} y_{27}^2 - 2x_{11} y_1 y_{27} y_{51} - y_{27}^3 - 3x_{12} y_{27}^2 y_{51}$$

For a given vector of parameters  $\{X\}$ , the above system of coupled differential equations must be solved in order to find the expressions for the gradient and Hessian approxi-

mation. The integrals of the expressions are again approximated using Simpson's rule over the 335 points, in point groups of three. Once the Hessian and the gradient expressions have been evaluated, the 12 by 12 system of equations [2.9] must be solved for the correction vector  $\{\Delta X\}$ . The parameter vector is then updated using equation [2.7]. The new parameters are then used to solve the system of equations [5.6], and the process repeated. Iterations are again terminated when each component of the parameter vector fails to change from the previous value of the component by a ratio of 5.0E-4.

## ***5.5 Results***

For the identification of the two degrees of freedom example presented above, the iterative direct method was found to be ineffective using initial parameter values similar to those used for the single degree of freedom examples. The method did not converge to any values, but bounced between unreasonable parameter estimates, causing solution of the differential equations [5.6] to become increasingly expensive. The initial values used are presented in the tables of Chapters 6 and 7. The Hessian approximation used for the Levenberg-Marquadt method leads to step sizes in  $\{X\}$  which are not optimum, as in the Newton-Raphson method. Perhaps this is why the method jumped between unreasonable values, without converging to specific parameter estimates. For initial values which were nearly exact before iterations were begun, the method did in fact converge, but more work is required to find upper bounds on the amounts by which initial guesses may be inaccurate, since the accuracy of these guesses was greater than could be expected in an actual structure identification study.

## 6.0 Single Step Integration Method: Two Degrees of Freedom System

The single step method presented in Chapter 3 may be extended to the simple two degrees of freedom system of Chapter 5. Extension of the method for use on this system requires a new formulation for the objective function, and therefore similar changes in the gradient and Hessian formulations must be made for use in the Newton-Raphson method.

### *6.1 Objective Function Formulation*

The error functions  $E_1$  and  $E_2$  presented in Chapter 3 represent residuals which may be reduced by changes made in the parameter vector  $\{X\}$ . The component  $E_1$  represents a collection of residual points along the record time length in displacement, and  $E_2$  likewise represents a collection of points in velocity residuals. The extension of the single step



differential equation solvers to systems of equations is simple, and is presented in Reference [21]. The two component error vector may be extended to systems of differential equations in the same way, where pairs of components representing error residuals in displacements and velocities may be formulated for each node under consideration. The error vector of equation [3.4] may thus be viewed as a four component expression of the difference between the present value of the measured displacements and velocities, and functions of the previous measured values. The specific  $\{\phi\}$  function of equation [3.5] may be used for a two degrees of freedom system by redefining the function vector  $\{F\}$  as a four component system analogous to the function vector used in solving second order single degree of freedom differential equations. For the system of equations [5.4], the transformed system of four first order equations takes the form

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} y_2 \\ -\Omega_1^2 y_1 - 2\gamma_1 y_2 - \alpha_1 y_1^3 - \alpha_2 y_1^2 y_3 - \alpha_3 y_1 y_3^2 - \alpha_4 y_3^3 \\ y_4 \\ -\Omega_2^2 y_3 - 2\gamma_2 y_4 - \alpha_5 y_1^3 - \alpha_6 y_1^2 y_3 - \alpha_7 y_1 y_3^2 - \alpha_8 y_3^3 \end{bmatrix} = \{F\} \quad [6.1]$$

The functions  $f_1$ ,  $f_2$ ,  $f_3$ , and  $f_4$  are thus evaluated for combinations of  $\{y\}$  and other function evaluations as in equation [3.3] for  $\{\phi\}$ . The four component error function may be incorporated into a simple objective function analogous to the form found in Chapter 3. This is given as

$$L\{X\} = \int_0^{T^*} \left\{ \frac{E_1^2}{T^{*2}} + E_2^2 + \frac{E_3^2}{T^{*2}} + E_4^2 \right\} dt \quad [6.2]$$

where  $E_1$  and  $E_3$  have been divided by the square of the record length in order to have all the terms in units of velocity squared. In order to minimize the objective function, derivatives of the error components must be supplied.

After substituting the function vector evaluations of equation [6.1] into the four component form of the error vector function of equation [3.4], the four components of  $\{E\}$  take the form

$$\begin{aligned}
 E_1 &= y_1^i - y_1^{i+1} + \frac{h}{2} \{ 2y_2^i + h(-x_1y_1^i - 2x_2y_2^i - x_3y_1^3 - x_4y_1^2y_3 - x_5y_1y_3^2 - x_6y_3^3) \} \\
 E_2 &= y_2^i - y_2^{i+1} + \frac{h}{2} \{ -x_1y_1^i - 2x_2y_2^i - x_3y_1^3 - x_4y_1^2y_3^i - x_5y_1y_3^2 - x_6y_3^3 \\
 &\quad - x_1(y_1^i + hy_2^i) - 2x_2y_2^i \\
 &\quad - 2hx_2(-x_1y_1^i - 2x_2y_2^i - x_3y_1^3 - x_4y_1^2y_3^i - x_5y_1y_3^2 - x_6y_3^3) \\
 &\quad - x_3(y_1^i + hy_2^i)^3 - x_4(y_1^i + hy_2^i)^2(y_3^i + hy_4^i) \\
 &\quad - x_5(y_1^i + hy_2^i)(y_3^i + hy_4^i)^2 - x_6(y_3^i + hy_4^i)^3 \} \tag{6.3}
 \end{aligned}$$

$$E_3 = y_3^i - y_3^{i+1} + \frac{h}{2} \{ 2y_4^i + h(-x_7y_3^i - 2x_8y_4^i - x_9y_1^3 - x_{10}y_1^2y_3^i - x_{11}y_1y_3^2 - x_{12}y_3^3) \}$$

$$\begin{aligned}
 E_4 &= y_4^i - y_4^{i+1} + \frac{h}{2} \{ -x_7y_3^i - 2x_8y_4^i - x_9y_1^3 - x_{10}y_1^2y_3^i - x_{11}y_1y_3^2 - x_{12}y_3^3 - x_7(y_3^i + hy_4^i) \\
 &\quad - 2x_8y_4^i - 2hx_8(-x_7y_3^i - 2x_8y_4^i - x_9y_1^3 - x_{10}y_1^2y_3^i - x_{11}y_1y_3^2 - x_{12}y_3^3) \\
 &\quad - x_9(y_1^i + hy_2^i)^3 - x_{10}(y_1^i + hy_2^i)^2(y_3^i + hy_4^i) - x_{11}(y_1^i + hy_2^i)(y_3^i + hy_4^i)^2 - x_{12}(y_3^i + hy_4^i)^3 \}
 \end{aligned}$$

where  $\{X\} = \{\Omega_1^2, \gamma_1, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \Omega_2^2, \gamma_2, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}^T$  and bars are understood, with all values of  $\{y\}^i$  measured. The error vector function is thus formulated, and supplied in the objective function formulation above.

## 6.2 Objective Function Differentiation

The gradient and Hessian expressions for the objective function of equation [6.2] are analogous to the expressions for the one degree of freedom examples of Chapter 3, with the exception that the two degrees of freedom expressions contain extra terms for derivatives of the error functions of the second node. The gradient expression takes the form

$$\begin{aligned} \frac{\partial L\{X\}}{\partial x_l} = 2 \int_0^{T^{\star}} \left\{ \frac{E_1}{T^{\star 2}} \left( \frac{\partial E_1}{\partial x_l} \right) + \left( \frac{\partial E_2}{\partial x_l} \right) E_2 \right. \\ \left. + \frac{E_3}{T^{\star 2}} \left( \frac{\partial E_3}{\partial x_l} \right) + \left( \frac{\partial E_4}{\partial x_l} \right) E_4 \right\} dt \end{aligned} \quad [6.4]$$

and the Hessian expression takes the form

$$\begin{aligned} \frac{\partial}{\partial x_k} \left( \frac{\partial L\{X\}}{\partial x_l} \right) = 2 \int_0^{T^{\star}} \left\{ \frac{1}{T^{\star 2}} \left( \frac{\partial E_1}{\partial x_k} \right) \left( \frac{\partial E_1}{\partial x_l} \right) + \frac{E_1}{T^{\star 2}} \frac{\partial}{\partial x_k} \left( \frac{\partial E_1}{\partial x_l} \right) \right. \\ \left. + \frac{\partial E_2}{\partial x_k} \left( \frac{\partial E_2}{\partial x_l} \right) + E_2 \frac{\partial}{\partial x_k} \left( \frac{\partial E_2}{\partial x_l} \right) \right. \\ \left. + \frac{1}{T^{\star 2}} \left( \frac{\partial E_3}{\partial x_k} \right) \left( \frac{\partial E_3}{\partial x_l} \right) + \frac{E_3}{T^{\star 2}} \frac{\partial}{\partial x_k} \left( \frac{\partial E_3}{\partial x_l} \right) \right. \\ \left. + \frac{\partial E_4}{\partial x_k} \left( \frac{\partial E_4}{\partial x_l} \right) + E_4 \frac{\partial}{\partial x_k} \left( \frac{\partial E_4}{\partial x_l} \right) \right\} dt \end{aligned} \quad [6.5]$$

The derivatives required in the expressions above are of the error function components with respect to the parameters. The first derivatives of the error functions take the form

$$DD11 = -\frac{h^2}{2} y_1^i$$

$$DD12 = -h^2 y_2^i$$

$$DD13 = -\frac{h^2}{2} y_1^{i3}$$

$$DD14 = -\frac{h^2}{2} y_1^{i2} y_3^i$$

$$DD15 = -\frac{h^2}{2} y_1^i y_3^{i2}$$

$$DD16 = -\frac{h^2}{2} y_3^{i3}$$

$$DD17 = DD18 = \dots = DD112 = 0$$

$$DD21 = DD22 = \dots = DD26 = 0$$

$$DD27 = -\frac{h^2}{2} y_3^i$$

$$DD28 = -h^2 y_4^i$$

$$DD29 = -\frac{h^2}{2} y_1^{i3}$$

$$DD210 = -\frac{h^2}{2} y_1^{i2} y_3^i$$

$$DD211 = -\frac{h^2}{2} y_1^i y_3^{i2}$$

$$DD212 = -\frac{h^2}{2} y_3^{i3}$$

$$DV11 = \frac{h}{2} \{-y_1^i - (y_1^i + hy_2^i) + 2hx_2 y_1^i\}$$

$$DV12 = h\{-2y_2^i - h(-x_1 y_1^i - 2x_2 y_2^i - x_3 y_1^{i3} - x_4 y_1^{i2} y_3^i - x_5 y_1^i y_3^{i2} - x_6 y_3^{i3}) + 2hx_2 y_2^i\}$$

$$DV13 = \frac{h}{2} \{-y_1^{i3} + 2hx_2 y_1^{i3} - (y_1^i + hy_2^i)^3\} \quad [6.6]$$

$$DV14 = \frac{h}{2} \{-y_1^{i2} y_3^i + 2hx_2 y_1^{i2} y_3^i - (y_1^i + hy_2^i)^2 (y_3^i + hy_4^i)\}$$

$$DV15 = \frac{h}{2} \{-y_1^i y_3^{i2} + 2hx_2 y_1^i y_3^{i2} - (y_1^i + hy_2^i)(y_3^i + hy_4^i)^2\}$$

$$DV16 = \frac{h}{2} \{-y_3^{i3} + 2hx_2 y_3^{i3} - (y_3^i + hy_4^i)^3\}$$

$$DV17 = DV18 = \dots = DV112 = 0$$

$$DV21 = DV22 = \dots = DV26 = 0$$

$$DV27 = \frac{h}{2} \{-y_3^i - (y_3^i + hy_4^i) + 2hx_8 y_3^i\}$$

$$DV28 = h\{-2y_4^i - h(-x_7 y_3^i - 2x_8 y_4^i - x_9 y_1^{i3} - x_{10} y_1^{i2} y_3^i - x_{11} y_1^i y_3^{i2} - x_{12} y_3^{i3}) + 2hx_8 y_4^i\}$$

$$DV29 = \frac{h}{2} \{-y_1^{i3} + 2hx_8 y_1^{i3} - (y_1^i + hy_2^i)^3\}$$

$$DV210 = \frac{h}{2} \{ -y_1^i y_3^i + 2hx_8 y_1^i y_3^i - (y_1^i + hy_2^i)^2 (y_3^i + hy_4^i) \}$$

$$DV211 = \frac{h}{2} \{ -y_1^i y_3^{i2} + 2hx_8 y_1^i y_3^{i2} - (y_1^i + hy_2^i)(y_3^i + hy_4^i)^2 \}$$

$$DV212 = \frac{h}{2} \{ -y_3^{i3} + 2hx_8 y_3^{i3} - (y_3^i + hy_4^i)^3 \}$$

where DDij is the derivative of the ith displacement node error function component with respect to the jth component of the parameter vector {X}, and DVij is the analogous derivative of the ith node velocity error function. The second derivatives required in the Hessian expression are simple to compute, and will not be presented here because of the large number of expressions involved.

### 6.3 Objective Function Minimization

Once the expressions above are evaluated for all the points along the time interval, Simpson's rule is used to approximate the integrals involved in the gradient and Hessian expressions. The system of 12 equations [2.9] is now solved for the correction vector, and the new values for {X} are found as in equation [2.7]. The new values for {X} are now used to find the first and second derivatives of the error vector function {E}, and the integral expressions for the gradient and Hessian are approximated. The system of 12 equations is solved for the correction vector {ΔX}, and new values for {X} are found. The iteration process continues until each component of the vector {X} fails to change as a fraction of its previous value by 5.0E-4.

## ***6.4 Results***

The results of the application of the single step integration method to the simple two degrees of freedom system above are presented in Tables 19 and 20. For pure input displacement and velocity data, the method converged to the exact parameter values. For the cases where noise was introduced to the data, the estimates rapidly decrease in accuracy, until in the 20% noise case, the estimates for most of the parameters become unreasonable. The RMS values of the "measured" quantities used in the noise calculations were 0.5, 2.7, 0.5, and 2.3 for the displacement and velocity components at nodes one and two, respectively. The method took four iterations and three CPU seconds to converge in all cases.

Filtering of the input data to the algorithm changed the parameter estimates slightly for the better, especially in cases where the estimates were not at all close to the exact values for unfiltered data. As noted in Chapter 3, the reason for the rapidly decreasing accuracy of final results with increasingly corrupted input data seems to be the fact that the single step integration method error function  $\{E\}$  depends only upon one set of previous measured values for displacement and velocity and the function evaluations there. This problem may be circumvented by use of the multiple step integration algorithm of the fourth chapter.

Table 19. Two Degrees of Freedom Results; Single Step Method, Unfiltered Data

$$\ddot{u}_1 = -\Omega_1^2 u_1 - 2\gamma_1 \dot{u}_1 - \alpha_1 u_1^3 - \alpha_2 u_1^2 u_2 - \alpha_3 u_1 u_2^2 - \alpha_4 u_2^3$$

$$\ddot{u}_2 = -\Omega_2^2 u_2 - 2\gamma_2 \dot{u}_2 - \alpha_5 u_1^3 - \alpha_6 u_1^2 u_2 - \alpha_7 u_1 u_2^2 - \alpha_8 u_2^3$$

$$u_1(t=0) = 1.5 \quad u_2(t=0) = -1.0$$

$$\dot{u}_1(t=0) = 0 \quad \dot{u}_2(t=0) = 0$$

Noise	Initial	0%	5%	10%	20%	Exact
$\Omega_1^2$	20.00	25.00	23.78	23.45	24.92	25.00
$\gamma_1$	0.50	0.35	0.37	0.55	1.24	0.35
$\alpha_1$	3.00	5.00	6.32	5.57	-0.21	5.00
$\alpha_2$	0.20	5.00	1.23	0.70	-2.59	0.5
$\alpha_3$	0.15	0.25	1.34	3.01	7.41	0.25
$\alpha_4$	4.00	3.00	2.74	0.11	-10.21	3.00
$\Omega_2^2$	14.00	17.00	17.17	16.74	14.57	17.00
$\gamma_2$	0.55	0.25	0.22	0.32	0.84	0.25
$\alpha_5$	4.50	2.50	3.07	5.42	14.74	2.50
$\alpha_6$	0.35	0.75	2.64	7.84	26.58	0.75
$\alpha_7$	0.35	0.20	1.35	3.48	10.03	0.20
$\alpha_8$	2.00	5.00	3.66	3.09	3.42	5.00



Table 20. Two Degrees of Freedom Results; Single Step Method, Filtered Data

$$\ddot{u}_1 = -\Omega_1^2 u_1 - 2\gamma_1 \dot{u}_1 - \alpha_1 u_1^3 - \alpha_2 u_1^2 u_2 - \alpha_3 u_1 u_2^2 - \alpha_4 u_2^3$$

$$\ddot{u}_2 = -\Omega_2^2 u_2 - 2\gamma_2 \dot{u}_2 - \alpha_5 u_1^3 - \alpha_6 u_1^2 u_2 - \alpha_7 u_1 u_2^2 - \alpha_8 u_2^3$$

$$u_1(t=0) = 1.5 \quad u_2(t=0) = -1.0$$

$$\dot{u}_1(t=0) = 0 \quad \dot{u}_2(t=0) = 0$$

Noise	Initial	0%	5%	10%	20%	Exact
$\Omega_1^2$	20.00	25.00	25.37	25.47	26.13	25.00
$\gamma_1$	0.50	0.35	0.37	0.40	0.55	0.35
$\alpha_1$	3.00	5.00	4.42	4.00	2.23	5.00
$\alpha_2$	0.20	5.00	0.03	-0.28	-2.36	0.5
$\alpha_3$	0.15	0.25	-0.03	0.03	-0.67	0.25
$\alpha_4$	4.00	3.00	2.63	1.94	-0.15	3.00
$\Omega_2^2$	14.00	17.00	16.81	16.51	15.48	17.00
$\gamma_2$	0.55	0.25	0.25	0.27	0.35	0.25
$\alpha_5$	4.50	2.50	2.24	2.13	2.37	2.50
$\alpha_6$	0.35	0.75	0.67	1.10	3.50	0.75
$\alpha_7$	0.35	0.20	0.77	1.78	5.10	0.20
$\alpha_8$	2.00	5.00	5.42	6.02	7.81	5.00

## **7.0 Multiple Step Integration Method: Two Degrees of Freedom System**

The multiple step integration method presented in Chapter 4 may be extended and used to identify the two degrees of freedom system of Chapter 5. In order to do so, the objective function must be redefined as in Chapter 6 and the derivatives needed for the gradient and Hessian expressions reformulated.

### ***7.1 Objective Function Formulation***

The general form of the objective function presented in Chapter 6 will be used again here, except that the error vector  $\{E\}$  will be redefined as was done for the error function of Chapter 4. In order to find the error function components, the system of governing differential equations must first be transformed to a system of first order differential equations, equation [6.1]. Next, the six step differential equation solution method of

equation [4.5] may be viewed as a four component (rather than two) method for the solution of the four first order differential equations. The equations for the updated vector  $\{y\}$  in equation [4.1] may then be used for the error function components if the new value of  $\{y\}$ ,  $\{y\}^i$ , is shifted to the right side of the equation, and measured values for  $\{y\}$ ,  $\{\bar{y}\}$ , are substituted for all the needed values of  $\{y\}$ . The resulting error function vector is

$$\{E\} = \frac{6h}{840} \{41F(\{y^{i-6}\}) + 216F(\{y^{i-5}\}) + 27F(\{y^{i-4}\}) + 272F(\{y^{i-3}\}) + 27F(\{y^{i-2}\}) + 216F(\{y^{i-1}\}) + 41F(\{y^i\})\} \quad [7.1]$$

The objective function is composed of the error function components from the above equation, and is approximated using a quadrature formula such as Simpson's rule for integration over the record length of time. The objective function integrand is in the form of discrete values of the sum of the squared error functions, so the notation used for the objective function and its derivatives is not exactly correct, as in the previous chapters. In order to minimize the objective function, derivatives of the error functions must be supplied.

## ***7.2 Objective Function Differentiation***

The general form for the gradient and Hessian of the objective function above are presented as equations [6.4] and [6.5]. The derivatives of the error components with respect to the parameters take on a simpler form than those of Chapter 6, involving linear combinations of the derivatives of the function  $\{F\}$ . The derivative of the error vector with respect to the  $i$ th parameter takes the same form as equation [4.6], with the function

{F} and the error function {E} comprised of four components for the two degrees of freedom example. As in equation [4.6], the derivatives of the components of {E} involve linear combinations of the derivatives of the components of {F}, so that only simple derivatives of {F} need be formulated analytically. The derivatives are used for seven separate {F} evaluations for each point in time of the error residuals, since the differential equation solution method chosen requires six previous values and the present value of {y}.

In order to compute the derivatives needed for the error function components, the following derivatives of {F} with respect to the parameters must be formulated.

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial f_1}{\partial x_2} = \dots = \frac{\partial f_1}{\partial x_{12}} = 0$$

$$\frac{\partial f_2}{\partial x_1} = -\bar{y}_1$$

$$\frac{\partial f_2}{\partial x_2} = -2\bar{y}_2$$

$$\frac{\partial f_2}{\partial x_3} = -\bar{y}_1^3$$

$$\frac{\partial f_2}{\partial x_4} = -\bar{y}_1^2 \bar{y}_3$$

$$\frac{\partial f_2}{\partial x_5} = -\bar{y}_1 \bar{y}_3^2$$

$$\frac{\partial f_2}{\partial x_6} = -\bar{y}_3^3 \tag{7.2}$$

$$\frac{\partial f_2}{\partial x_7} = \frac{\partial f_2}{\partial x_8} = \dots = \frac{\partial f_2}{\partial x_{12}} = 0$$

$$\frac{\partial f_3}{\partial x_1} = \frac{\partial f_3}{\partial x_2} = \dots = \frac{\partial f_3}{\partial x_{12}} = 0$$

$$\frac{\partial f_4}{\partial x_1} = \frac{\partial f_4}{\partial x_2} = \dots = \frac{\partial f_4}{\partial x_6} = 0$$

$$\frac{\partial f_4}{\partial x_7} = -\bar{y}_3$$

$$\frac{\partial f_4}{\partial x_8} = -2\bar{y}_4$$

$$\frac{\partial f_4}{\partial x_9} = -\bar{y}_1^3$$

$$\frac{\partial f_4}{\partial x_{10}} = -\bar{y}_1^2 \bar{y}_3$$

$$\frac{\partial f_4}{\partial x_{11}} = -\bar{y}_1 \bar{y}_3^2$$

$$\frac{\partial f_4}{\partial x_{12}} = -\bar{y}_3^3$$

The expressions above are used in the gradient and Hessian expressions for each set of points  $\{\bar{y}\}'$  along the record length of time  $T^*$ . Note that for this specific example, the second derivatives of the function  $\{F\}$  are zero. This leaves a simple expression for the Hessian, since half the terms in its integrand would otherwise be second derivatives of

the functions  $\{F\}$ . The integrals contained in the Hessian and gradient expressions are again approximated using Simpson's rule over each group of three points computed using the above derivatives in the error function derivative expressions.

### ***7.3 Objective Function Minimization***

The objective function is again minimized with the Newton-Raphson method, using the gradient and Hessian computations from above. The procedure is the same as that followed in the preceding chapters, with 12 parameters on which to iterate. The iterations are again terminated when the convergence criterion of the preceding chapters is met.

### ***7.4 Results***

The results of application of the multiple step integration method to the simple two degrees of freedom system above are presented in Tables 21 and 22. The RMS values used in the noise calculations are presented in Chapter 6. The computer program used for this example is presented in the Appendix. The natural frequencies of the example were chosen in a way so as to avoid internal resonance of the system (see the book by Nayfeh [24]). The method took just two iterations to converge for all cases, which implies actual convergence in just one step, since the convergence criterion requires an extra iteration. The amount of CPU time used for all cases was just three seconds.

For pure input values of displacement and velocity to the algorithm, the parameter values are estimated exactly. The estimates for unfiltered corrupted input data range from nearly exact values in natural frequency estimates to unacceptable values for the cross term coefficients of the differential equations in the 20% noise case. Parameter estimates for the dominating coefficients of the equations were in general more accurate than for those with small sought values. The minimization algorithm may be considered more responsive to the dominating coefficients of the differential equations.

After filtering of the input data, many of the parameter estimates have improved, though not sufficiently in some cases. The cross term coefficients, in particular, have not improved to a degree where the numbers would represent useful estimates for an actual structural system. Nonetheless, the method could provide valuable first estimates of parameters dominating an actual structural system.

Plots of the fitted displacements for the two nodes are presented in Figures 16 and 17 for unfiltered input data. Note the nonzero initial values in displacement for these cases. All plots are made using parameter estimates from the worst cases, where the noise was 20%. The fits for both nodes are good, even though some of the estimates for the less dominant parameters were poor. The fits are still good since the less dominant terms of the differential equations obviously have less influence on the look of the response of the system than the more dominant terms. The curve fits for filtered input data are presented in Figures 18 and 19 for the 20% noise case. The fit for displacement at the second node has improved slightly over that for unfiltered input data. The fit for the displacement at the first node has not improved quite as much as the second node fit with filtering.

Table 21. Two Degrees of Freedom Results, Multiple Step Method, Unfiltered Data

$$\ddot{u}_1 = -\Omega_1^2 u_1 - 2\gamma_1 \dot{u}_1 - \alpha_1 u_1^3 - \alpha_2 u_1^2 u_2 - \alpha_3 u_1 u_2^2 - \alpha_4 u_2^3$$

$$\ddot{u}_2 = -\Omega_2^2 u_2 - 2\gamma_2 \dot{u}_2 - \alpha_5 u_1^3 - \alpha_6 u_1^2 u_2 - \alpha_7 u_1 u_2^2 - \alpha_8 u_2^3$$

$$u_1(t=0) = 1.5 \quad u_2(t=0) = -1.0$$

$$\dot{u}_1(t=0) = 0 \quad \dot{u}_2(t=0) = 0$$

Noise	Initial	0%	5%	10%	20%	Exact
$\Omega_1^2$	20.00	25.00	24.58	24.23	23.72	25.00
$\gamma_1$	0.50	0.35	0.34	0.34	0.35	0.35
$\alpha_1$	3.00	5.00	5.97	6.76	7.77	5.00
$\alpha_2$	0.20	0.50	1.72	2.72	4.07	0.5
$\alpha_3$	0.15	0.25	0.77	1.23	1.99	0.25
$\alpha_4$	4.00	3.00	2.92	2.79	2.44	3.00
$\Omega_2^2$	14.00	17.00	17.11	17.27	17.75	17.00
$\gamma_1$	0.55	0.25	0.26	0.26	0.27	0.25
$\alpha_5$	4.50	2.50	2.73	2.93	3.21	2.50
$\alpha_6$	0.35	0.75	0.61	0.34	-0.56	0.75
$\alpha_7$	0.35	0.20	-0.30	-0.91	-2.40	0.20
$\alpha_8$	2.00	5.00	4.75	4.40	3.46	5.00



Table 22. Two Degrees of Freedom Results, Multiple Step Method, Filtered Data

$$\ddot{u}_1 = -\Omega_1^2 u_1 - 2\gamma_1 \dot{u}_1 - \alpha_1 u_1^3 - \alpha_2 u_1^2 u_2 - \alpha_3 u_1 u_2^2 - \alpha_4 u_2^3$$

$$\ddot{u}_2 = -\Omega_2^2 u_2 - 2\gamma_2 \dot{u}_2 - \alpha_5 u_1^3 - \alpha_6 u_1^2 u_2 - \alpha_7 u_1 u_2^2 - \alpha_8 u_2^3$$

$$u_1(t=0) = 1.5 \quad u_2(t=0) = -1.0$$

$$\dot{u}_1(t=0) = 0 \quad \dot{u}_2(t=0) = 0$$

Noise	Initial	0%	5%	10%	20%	Exact
$\Omega_1^2$	20.00	25.00	24.92	24.81	24.64	25.00
$\gamma_1$	0.50	0.35	0.35	0.34	0.34	0.35
$\alpha_1$	3.00	5.00	5.17	5.34	5.57	5.00
$\alpha_2$	0.20	0.50	1.00	1.50	2.35	0.5
$\alpha_3$	0.15	0.25	0.42	0.62	0.97	0.25
$\alpha_4$	4.00	3.00	2.82	2.61	2.14	3.00
$\Omega_2^2$	14.00	17.00	17.09	17.21	17.51	17.00
$\gamma_2$	0.55	0.25	0.24	0.24	0.22	0.25
$\alpha_5$	4.50	2.50	2.20	1.88	1.15	2.50
$\alpha_6$	0.35	0.75	0.28	-0.25	-1.46	0.75
$\alpha_7$	0.35	0.20	0.41	0.59	0.83	0.20
$\alpha_8$	2.00	5.00	4.92	4.81	4.43	5.00

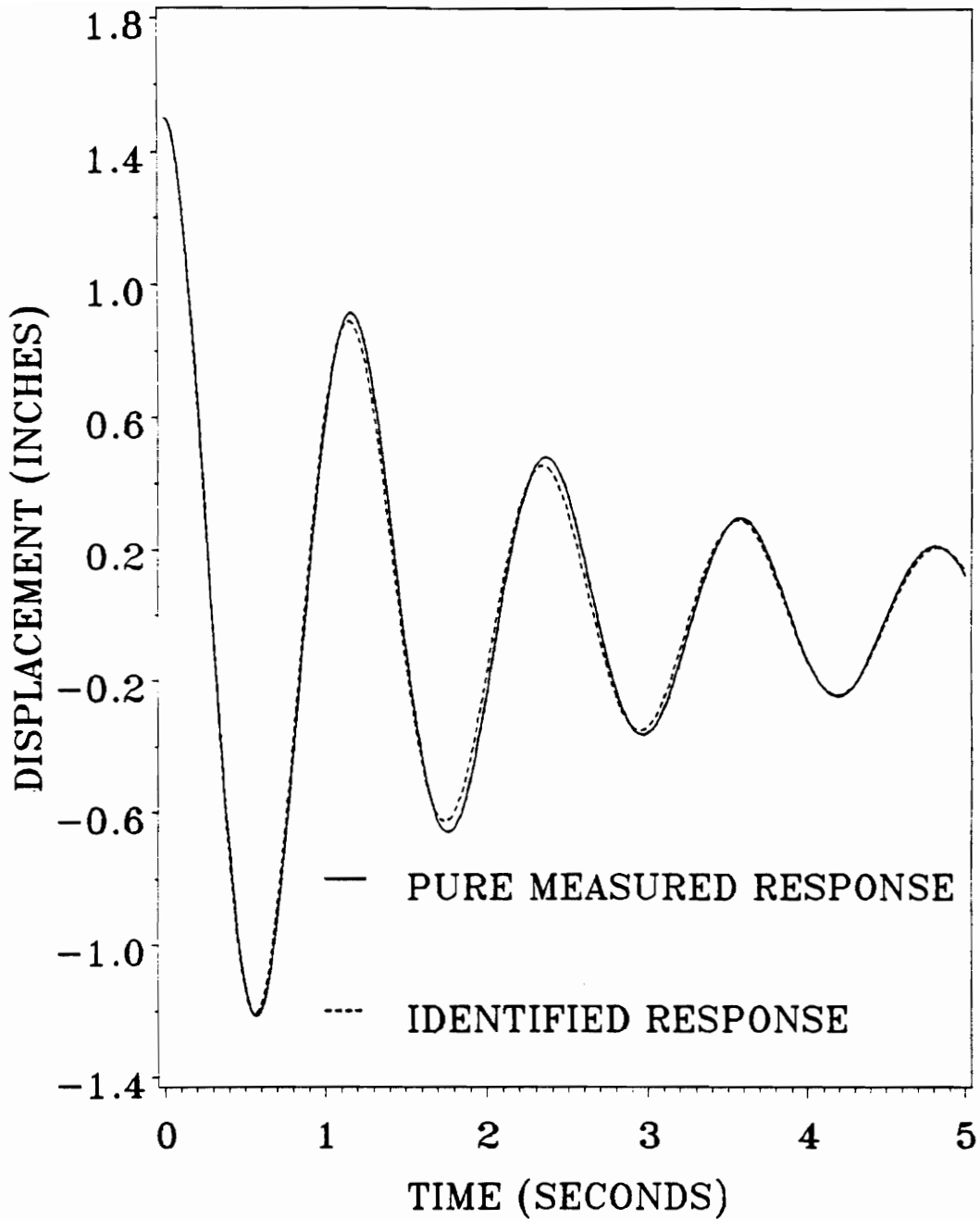


Figure 16. Identified/Measured Response of 2-D Nonlinearity Example: Multistep integration method; 20% corrupted input data, unfiltered, Node 1.

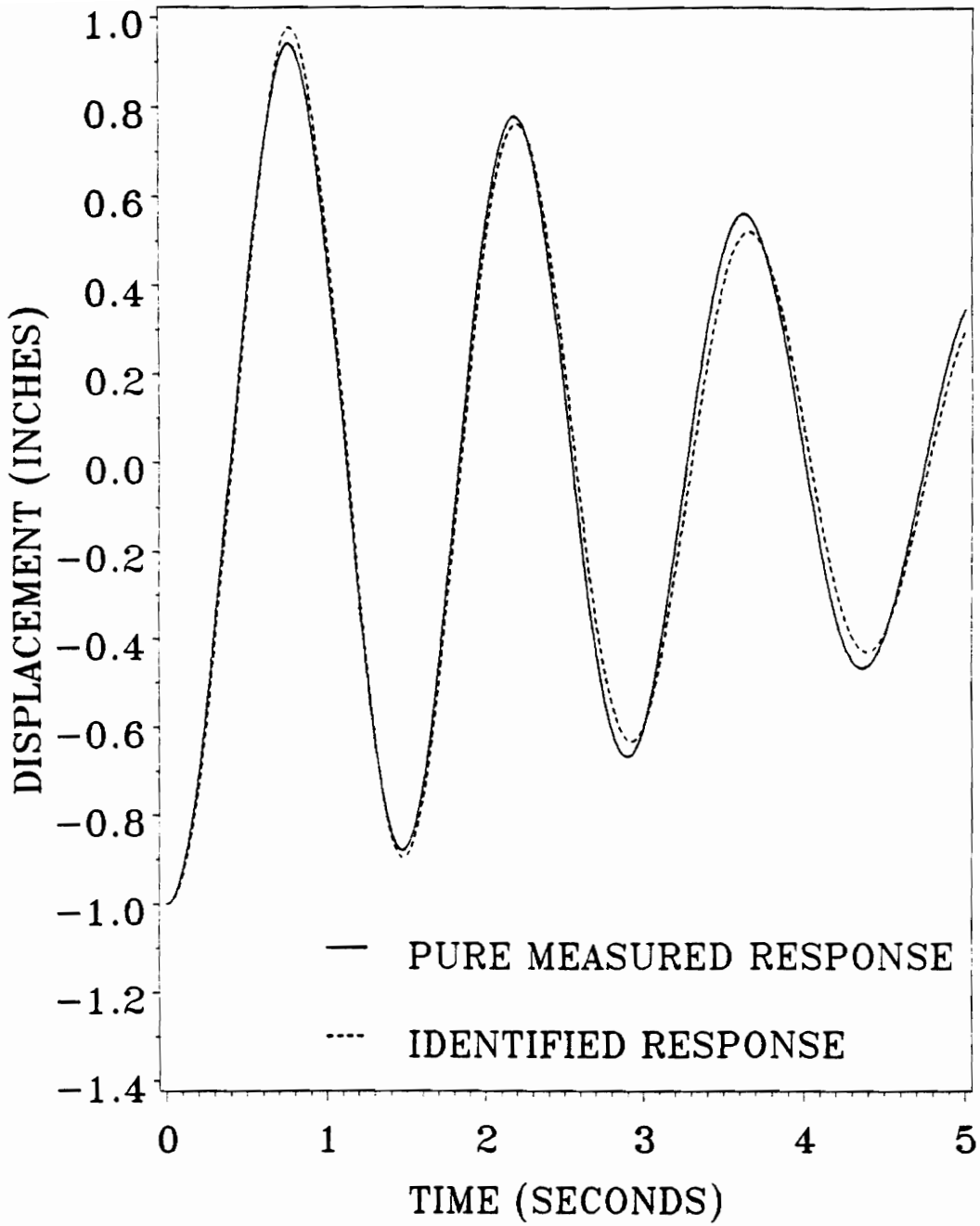


Figure 17. Identified/Measured Response of 2-D Nonlinearity Example: Multistep integration method; 20% corrupted input data, unfiltered, Node 2.

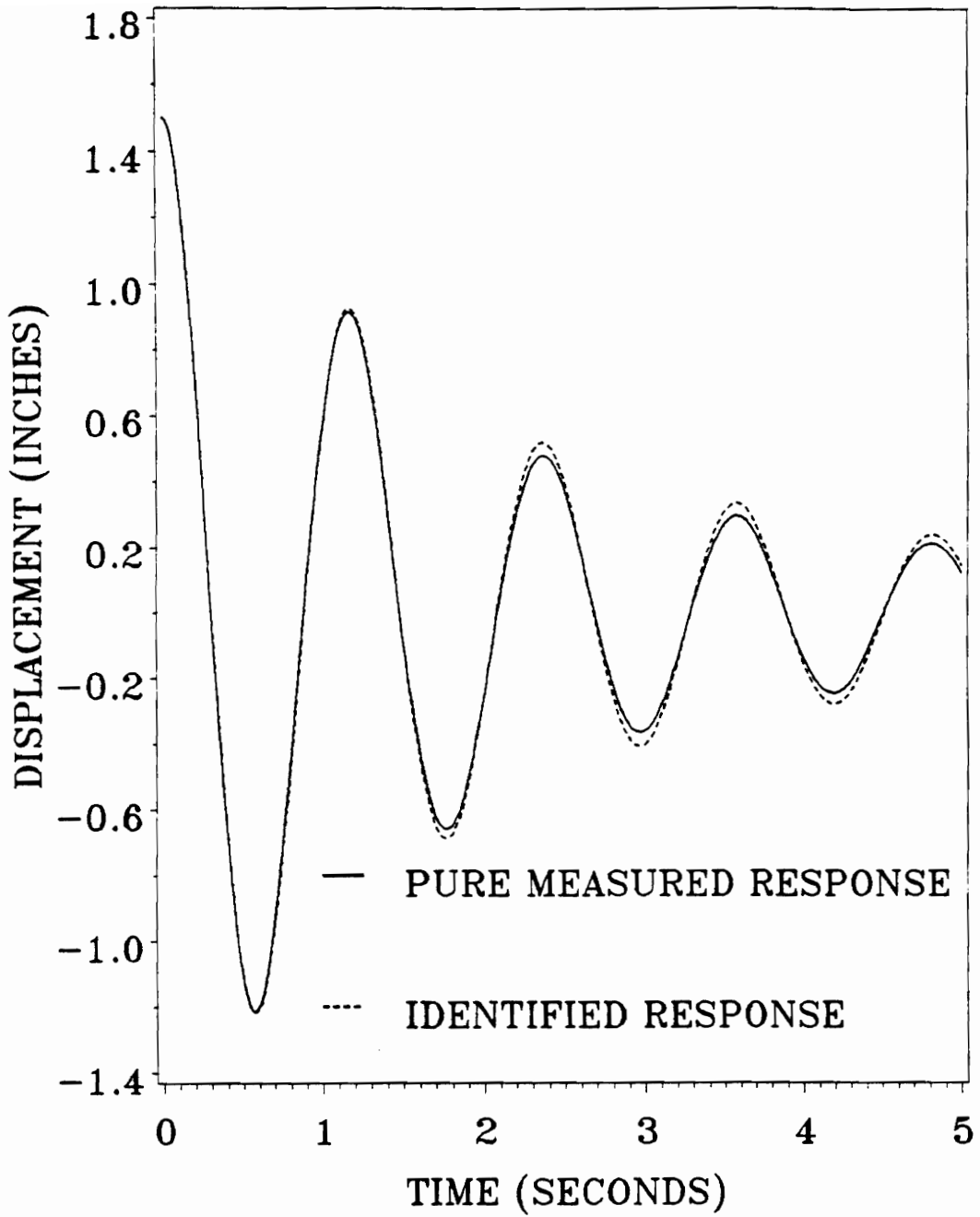


Figure 18. Identified/Measured Response of 2-D Nonlinearity Example: Multistep integration method; 20% corrupted input data, filtered, Node 1.

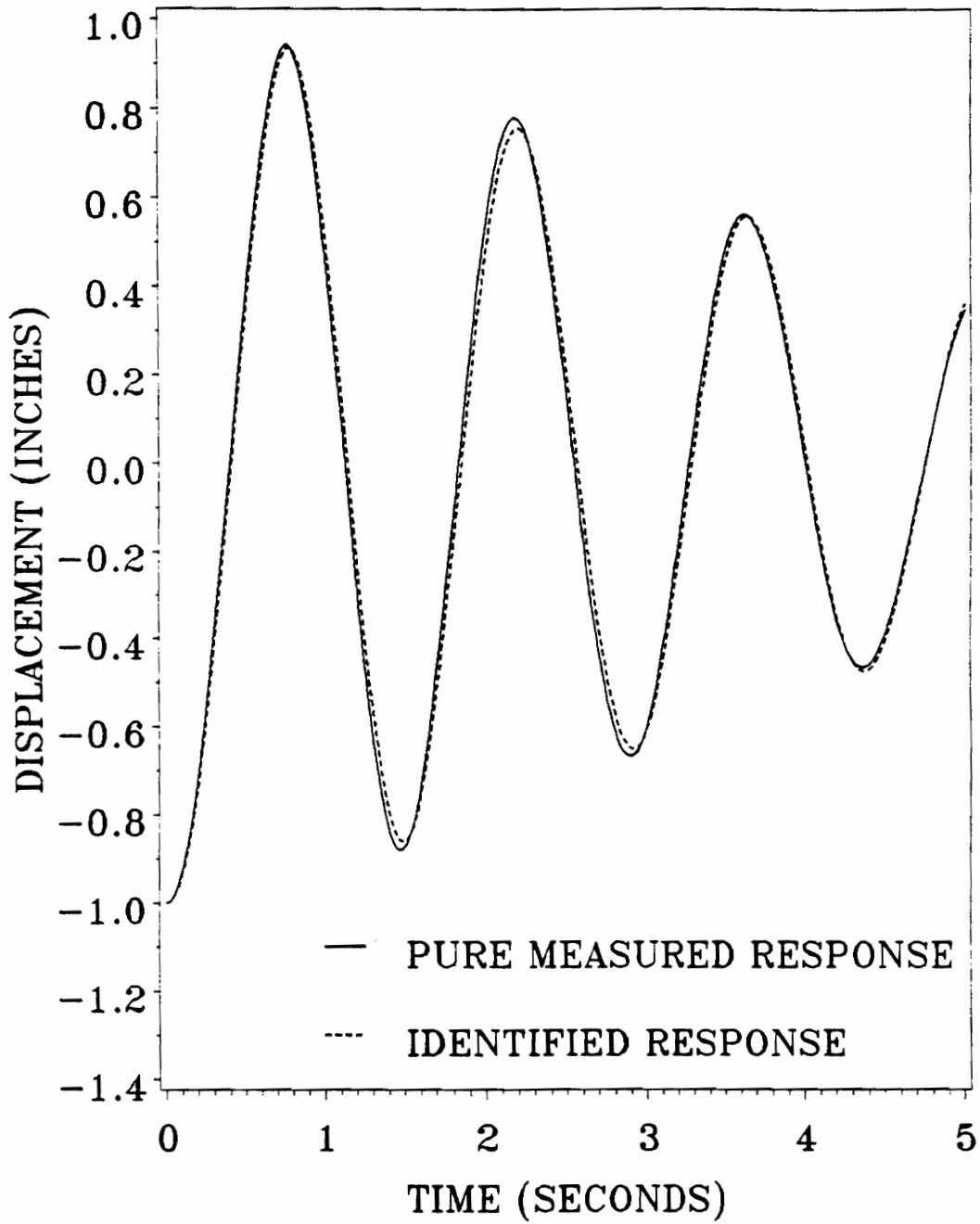


Figure 19. Identified/Measured Response of 2-D Nonlinearity Example: Multistep integration method; 20% corrupted input data, filtered, Node 2.

## 8.0 Conclusions and Future Work

The multiple step integration method has been found to be effective for the identification of four nonlinear structural dynamic systems. The most desirable qualities of the method are the simplicity of use of the method, the low computational costs involved in the use of the algorithm, the good accuracy of the estimates the method generates, and the extendability of the method to even larger degree of freedom systems.

In the use of the multiple step method, analytical differentiation of the governing differential equation with respect to the parameters is usually extremely simple, as in the examples studied. The ease of finding the required derivatives allows use of the Newton-Raphson method, which in turn guarantees faster convergence to the estimates than the Levenberg-Marquardt method used in conjunction with the iterative direct method.

Since the Newton-Raphson method is implemented on a system where derivatives are simple to compute with simple resulting expressions, the method converges to final estimates of the parameters extremely quickly. The high speed of convergence of the method implies lower computational costs than the slower iterative direct method.

The estimates generated using the multiple step integration algorithm were in general slightly less accurate than those from the iterative direct method, but were much better than those from the single step integration method. Since the improvement of the estimates in using a number of previous measured values over the use of just one point and its function evaluations was so great, further refinement of the method is suggested, since the method is so desirable in the other aspects mentioned. Perhaps the inclusion of more than six points in the error functions for the objective function would grant more accurate results in the final parameter estimates. Finally, the ease of extension of the method to two degrees of freedom system studied suggests that the method could be extended to higher degree of freedom systems.

## ***8.1 Future Work***

As mentioned above, refinement of the method presented is warranted, since, in the author's opinion, the desirable qualities mentioned outweigh the slightly inaccurate results obtained for noisy input data. Although the results for noisy data were not found to be as good as those for the more sophisticated iterative direct method, the extension of the iterative direct method to larger degree of freedom systems is not simple, and requires the solution of extremely large systems of differential equations at each iteration on the parameter values, for systems with three and more degrees of freedom.

The following list summarizes issues encountered in this study which require further study:

- The refinement of the multiple step method for use with noisy input data.
- The possibility of predicting an upper bound on the error of the initial parameter estimates used to begin the iterative process.
- The effects of step size changes in measurements and in the formulation of the objective function on the final parameter estimates.
- The addition of amplitude based noise in input data.
- The use of other types of filters, such as FFT algorithms, in data processing prior to the minimization process.
- The effects of changes in initial conditions and amplitude of measured data on final parameter estimates.
- The incorporation of other determinate and indeterminate forcing functions applied to the systems.
- The possible incorporation of filtering into the error formulations, since the filter used here involves a linear combination of the data points, as do the error functions used in the objective functions.
- The extension of the method to higher degree of freedom systems.



## Bibliography

1. Pappa, R. S., and Juang, J. N., "Galileo Spacecraft Modal Identification Using an Eigensystem Realization Algorithm," AIAA paper 84-1070-CP, May, 1984.
2. Natke, H. G., *Identification of Vibrating Structures*, Springer-Verlag, New York, 1982.
3. Gladwell, G. M. L., "Inverse Problems in Vibration," *Applied Mechanics Review*, Vol. 39, (1986), pp.1013-1016.
4. Ibanez, P., "Identification of Dynamic Parameters of Linear and Non-Linear Structural Models From Experimental Data," *Nuclear Engineering and Design*, Vol. 25, (1973), pp. 30-41.
5. Denman, E. D., and Jeon, G. J., "Control of Large Flexible Systems via Eigenvalue Relocation," *International Journal of Applied Finite Elements and Computer Aided Engineering*, Vol. 1, (1985), pp.241-253.

6. Kapania, R. K., and Raciti, S., "A 20 DOF Element for Nonlinear Analysis of Unsymmetrically Laminated Beams," *AIAA Journal*, Vol. 27, (1989), pp.201-210.
7. Gawthrop, P. J., Kountzeris, A., and Roberts, J. B., "Parametric Identification of Nonlinear Ship Roll Motion from Forced Roll Data," *Journal of Ship Research*, Vol. 32, (1988), pp. 101-111.
8. Batill, S. M., and Bacarro, J. M., "Modeling and Identification of Nonlinear Dynamic Systems With Application to Aircraft Landing Gear," AIAA paper 88-CP-2315.
9. Crawley, E. F., and O'Donnell, K. J., "Force-State Mapping Identification of Nonlinear Joints," *AIAA Journal*, Vol. 25, (1987), pp. 1003-1010.
10. Hanagud, S. V., Meyyappa, M., and Craig, J. I., "Method of Multiple Scales and Identification of Nonlinear Structural Dynamic Systems," *AIAA Journal*, Vol.23 (1985), pp. 802-807.
11. Nayfeh, A. H., *Perturbation Methods*, Wiley-Interscience, New York, 1979.
12. Distefano, N., and Rath, A., "System Identification in Nonlinear Structural Seismic Dynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 5 (1975), pp.353-372.
13. Levenberg, K., "A Method for the Solution of Certain Nonlinear Problems in Least Squares," *Quarterly of Applied Mathematics*, Vol. 2, (1944), pp. 164-168.

14. Marquardt, D. W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters," *SIAM Journal*, Vol. 11, (1963), pp.431-441.
15. Dixon, W. J., *Biomedical Computer Programs*, University of California Press, Berkeley, CA, 1973.
16. Bloomfield, P., *Fourier Analysis of Time Series: An Introduction*, Wiley and Sons, New York, 1976.
17. Stearns, S. D., *Digital Signal Analysis*, Hayden Book Company, New Jersey, 1985.
18. Craig, R. R., Jr., *Structural Dynamics-An Introduction to Computer Methods*, Wiley and Sons, New York, 1981.
19. IMSL Math/Library, Ver. 1.0, April, 1987, MAL-USM-UNBND-1.0.
20. Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.
21. Johnson, L. W., and Riess, R. D., *Numerical Analysis*, Addison-Wesley, Massachusetts, 1982.
22. Bard, Y., *Nonlinear Parameter Estimation*, Academic Press, New York, 1974.
23. Wu, X. J., Wang, Y., and Price, W. G., "Multiple Resonances, Responses, and Parametric Instabilities in Offshore Structures," *Journal of Ship Research*, Vol. 32, (1988), pp.285-296.

24. Nayfeh, A. H., and Mook, D., *Nonlinear Oscillations*, Wiley Interscience, New York, 1979.

## Appendix A. Program Mult

The fortran program Mult, listed below, identifies a two degrees of freedom system of the form found in Chapters 5, 6, and 7. The program reads in 4 measured time series of points UMES, with 335 points each. The format of the read in series is 4(1X,D14.7). An initial set of parameter values must be supplied inside the program, in th variable array XO. The final tolerance between old and new estimates for the parameters is specified in the main program as TOL.

PROGRAM MULT

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      N IS THE NUMBER OF PARAMETERS SOUGHT, CNTR IS A COUNTER TO      C
C      MONITOR THE NUMBER OF ITERATIONS NEEDED FOR CONVERGENCE, MXDIM C
C      IS EQUAL TO N FOR THESE PROBLEMS AND IS USED IN A GAUSSIAN      C
C      ELIMINATION SUBROUTINE, IERR IS AN OUTPUT VALUE FROM THE        C
C      GAUSSIAN ELIMINATION ROUTINE INDICATING CORRECT SOLUTION        C
C      PROCESSING IF IERR IS 1, RNORM1 IS ALSO AN OUTPUT VALUE FOR      C
C      MONITORING THE SOLUTION PROCESS IN THE GAUSS SUBPROGRAM.        C
```

```
C      NVAR IS AGAIN THE NUMBER OF PARAMETERS SOUGHT.      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
INTEGER N, CNTR, IERR, MXDIM, NPOINT, NVAR
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      NPOINT=CUT+2.  CUT SHOULD BE CHOSEN SUCH THAT CUT-1 IS ODD  C
C      AND DIVISIBLE BY 3.  NVAR IS THE NUMBER OF PARAMETERS SOUGHT. C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
PARAMETER (NVAR=12, NPOINT=335)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      XN IS THE UPDATED VALUE OF THE PARAMETER VECTOR, A1 IS THE  C
C      NVAR*NVAR HESSIAN OF THE OBJECTIVE FUNCTION, STEP IS THE  C
C      NVAR COMPONENT VECTOR OF THE STEP TAKEN IN XO, THE OLD VALUE C
C      OF THE PARAMETER VECTOR.  GRAD IS THE NVAR COMPONENT GRADIENT C
C      OF THE OBJECTIVE FUNCTION.  UMES IS THE DISPLACEMENT AND  C
C      VELOCITY MATRIX OF MEASURED VALUES.  THE DIMENSIONS OF UMES  C
C      ARE 2*NUMBER OF NODES BY NPOINT, THE NUMBER OF POINTS MEASUREDC
C      IN TIME.  FOR 2 DIMENSIONS, UMES(1,1-NPOINT) IS THE  C
C      DISPLACEMENT AND UMES(2,1-NPOINT) IS THE VELOCITY AT NODE ONE.C
C      UMES(3,1-NPOINT) IS THE DISPLACEMENT AT NODE 2,UMES(4,1-NVAR) C
C      IS THE VELOCITY AT NODE 2.      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```

DOUBLE PRECISION XN(NVAR), A1(NVAR,NVAR), STEP(NVAR)
DOUBLE PRECISION TOL, GRAD(NVAR), XO(NVAR), INT(2), CUT
DOUBLE PRECISION STEPSZ, UMES(4,335), RNORM1
COMMON /ALL/ STEPSZ, UMES
OPEN (UNIT=2,FILE='MULTIN',STATUS='OLD')
OPEN (UNIT=3,FILE='MULTOUT',STATUS='NEW')
N=NVAR
MXDIM=NVAR

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      N IS THE NO. OF DESIGN VARS, CUT IS THE NO. OF DIVISIONS FOR C
C      INTEGRATION INSIDE INT., THE INTERVAL OF TIME TO BE          C
C      INTEGRATED OVER.                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

INT(1)=0.0D0
INT(2)=5.0D0
CUT=334.0D0
TOL=5.0D-4

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      TOL IS THE FINAL CONVERGENCE TOLERANCE, XO IS THE VECTOR OF C
C      INITIAL VALUES FOR THE PARAMETERS. STEPSZ IS THE SIZE OF THE C

```

C THE STEP TAKEN IN THE RUNGE KUTTA AND QUADRATURE FORMULAS. C

CC

XO(1)=20.0D0

XO(2)=0.5D0

XO(3)=3.00D0

XO(4)=0.2D0

XO(5)=0.15D0

XO(6)=4.00D0

XO(7)=14.0D0

XO(8)=0.55D0

XO(9)=4.50D0

XO(10)=0.35D0

XO(11)=0.35D0

XO(12)=2.00D0

STEPSZ=(INT(2)-INT(1))/CUT

CC

C READ IN THE MEASURED TIME SERIES. C

CC

DO 10 J=1,NPOINT

READ(2,20) UMES(1,J),UMES(2,J),UMES(3,J),UMES(4,J)

10 CONTINUE

20 FORMAT (4(1X,D14.7))



```

CNTR=1
WRITE (3,30)XO(1),XO(2),XO(3)
WRITE (3,40)XO(4),XO(5),XO(6)
WRITE (3,50)XO(7),XO(8),XO(9)
WRITE (3,60)XO(10),XO(11),XO(12)
30  FORMAT (1X,' INITIAL X1, X2, X3',2X,3D16.9)
40  FORMAT (1X,' INITIAL X4, X5, X6',2X,3D16.9)
50  FORMAT (1X,' INITIAL X7, X8, X9',2X,3D16.9)
60  FORMAT (1X,' INITIAL X10, X11, X12',2X,3D16.9)

70  IF (CNTR.EQ.20) GOTO 160
    CNTR=CNTR+1

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    FIND INPUT GRADIENT VECTOR GRAD.                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    CALL COMGRA(XO,GRAD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    FIND INPUT HESSIAN MATRIX A1.                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    CALL HESH(XO,A1)

```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SOLVE N*N SYSTEM OF LINEAR EQUATIONS TO FIND STEP VECTOR.      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      CALL GAUSS(A1,GRAD,STEP,N,MXDIM,IERR,RNORM1)
      WRITE (6,*) 'IERR IS',IERR,'RNORM1 IS',RNORM1
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      UNPDATE PARAMETER VECTOR WITH STEP FOUND WITH GAUSS.      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      DO 80 L=1,N
          XN(L)=XO(L)-STEP(L)
80      CONTINUE
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      CHECK NEW PARAMETER VECTOR FOR CONVERGENCE CRITERIA.      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      DO 100 I=1,N
          AMOUNT=DABS((XN(I)-XO(I))/XO(I))
          IF (AMOUNT.GT.TOL) THEN
              DO 90 J=1,N
                  XO(J)=XN(J)
90      CONTINUE
```

```

        GOTO 70

        ENDIF

100  CONTINUE

        WRITE (3,*)

        WRITE (3,110)XN(1),XN(2),XN(3)

        WRITE (3,120)XN(4),XN(5),XN(6)

        WRITE (3,130)XN(7),XN(8),XN(9)

        WRITE (3,140)XN(10),XN(11),XN(12)

110  FORMAT (1X,'FINAL X1, X2, X3',4X,3D16.9)
120  FORMAT (1X,'FINAL X4, X5, X6',4X,3D16.9)
130  FORMAT (1X,'FINAL X7, X8, X9',4X,3D16.9)
140  FORMAT (1X,'FINAL X10, X11, X12',4X,3D16.9)
150  WRITE (3,*) 'F HAS MINIMIZED'

        STOP

160  WRITE (3,*) 'COUNT REACHED MAXIMUM'

        END

C
C
C   GIVEN THE PRESENT VALUES OF THE PARAMETERS, COMGRA COMPUTES
C   THE GRADIENT OF THE OBJECTIVE FUNCTION, USING SIMPSON'S
C   RULE FOR INTEGRATION.  COMGRA CALLS THE SUBROUTINE GR IN
C   ORDER TO FIND THE VALUES OF THE GRADIENT INTEGRAND BEFORE
C   APPROXIMATING INTEGRATION.
C
        SUBROUTINE COMGRA (X,GRAD)

```

```

INTEGER COUNT, J, K, MAX, NPOINT, NVAR, NDIM
PARAMETER (NVAR=12, NPOINT=335)
DOUBLE PRECISION UMES(4,335)
DOUBLE PRECISION STEPSZ, GRAD(NVAR), X(NVAR)
DOUBLE PRECISION OLD(NVAR), NEW(NVAR), PRES(NVAR)
COMMON /ALL/ STEPSZ, UMES
DO 1 J=1,NVAR
    GRAD(J)=0.000
1 CONTINUE
CALL GR(X,7,OLD)
MAX=NPOINT-1
DO 5 COUNT=8,MAX,2

    CALL GR(X,COUNT+1,NEW)
    CALL GR(X,COUNT,PRES)

DO 3 NDIM=1,NVAR
    GRAD(NDIM)=GRAD(NDIM)+(STEPSZ/3.000)*
*           (PRES(NDIM)*4.000+NEW(NDIM)
*           +OLD(NDIM))
3 CONTINUE

DO 4 K=1,NVAR
    OLD(K)=NEW(K)

```

4 CONTINUE

5 CONTINUE

RETURN

END

C

C SUBROUTINE GR FORMULATES THE VALUES OF

C EACH COMPONENT OF THE GRADIENT OF THE OBJECTIVE

C FUNCTION BEFORE INTEGRATING. GR MAKES

C CALLS TO DERIV TO FIND THE DERIVATIVE OF

C EACH COMPONENT OF THE MULTISTEP UPDATE

C FUNCTION WITH RESPECT TO THE PARAMETERS.

C GR ALSO CALLS DIFF TO FIND THE DIFFERENCE

C BETWEEN MEASURED AND PREDICTED VALUES OF

C DISPLACEMENTS AND VELOCITIES.

C

SUBROUTINE GR (X,NP,A)

INTEGER NP, NVAR,L

PARAMETER (NVAR=12)

DOUBLE PRECISION X(NVAR),A(NVAR),B(4,NVAR)

DOUBLE PRECISION Q(4)

CALL DERIV (X,NP,B)

CALL DIFF(X,NP,Q)

```

DO 5 L=1,NVAR
    A(L)=2.0D0*(Q(1)*B(1,L)/25.0D0+Q(2)*B(2,L)
*           +Q(3)*B(3,L)/25.0D0+Q(4)*B(4,L))
5    CONTINUE

RETURN

END

C
C
C    GIVEN THE PRESENT VALUES OF THE PARAMETERS,
C    HESH COMPUTES THE HESSIAN OF THE OBJECTIVE FUNCTION,
C    USING SIMPSON'S RULE FOR INTEGRATION.  HESH CALLS
C    THE SUBROUTINE H IN ORDER TO FIND THE VALUES OF THE
C    HESSIAN INTEGRAND BEFORE APPROXIMATING INTEGRATION.
C
C
SUBROUTINE HESH (X,HE)
    INTEGER COUNT, J, K, MAX, NPOINT ,NVAR
    PARAMETER (NVAR=12, NPOINT=335)
    DOUBLE PRECISION UMES(4,335)
    DOUBLE PRECISION HE(NVAR,NVAR), STEPSZ
    DOUBLE PRECISION OLD(NVAR,NVAR), NEW(NVAR,NVAR)
    DOUBLE PRECISION PRES(NVAR,NVAR), X(NVAR)
    COMMON /ALL/ STEPSZ, UMES

```

```

DO 2 K=1,NVAR
    DO 1 L=1,NVAR
        HE(K,L)=0.0DO
1    CONTINUE
2    CONTINUE

CALL H(X,7,OLD)

DO 4 M=2,NVAR
    DO 3 N=1,M-1
        OLD(M,N)=0.0DO
3    CONTINUE
4    CONTINUE

MAX= NPOINT-1
DO 10 COUNT=8,MAX,2

    CALL H(X,COUNT+1,NEW)
    CALL H(X,COUNT,PRES)

DO 7 L=1,NVAR
    DO 6 M=L,NVAR
        HE(L,M)=HE(L,M)+(STEPSZ/3.0DO)*
*           (PRES(L,M)*4.0DO+NEW(L,M)
*           +OLD(L,M))

```

```

6             CONTINUE
7             CONTINUE

             DO 9 J=1,NVAR
                 DO 8 K=J,NVAR
                     OLD(J,K)=NEW(J,K)
8             CONTINUE
9             CONTINUE

10            CONTINUE

             DO 15 J=1,NVAR
                 DO 14 K=J,NVAR
                     HE(K,J)=HE(J,K)
14            CONTINUE
15            CONTINUE

             RETURN

             END

```

```

C
C     SUBROUTINE H FORMULATES THE VALUES OF EACH
C     COMPONENT OF THE HESSIAN OF THE OBJECTIVE
C     FUNCTION BEFORE INTEGRATING.  H MAKES CALLS
C     TO DERIV TO FIND THE DERIVATIVE OF EACH COMPONENT
C     OF THE MULTISTEP UPDATE FUNCTION WITH RESPECT

```



C TO THE PARAMETERS. H CALLS DIFF TO FIND THE  
C DIFFERENCE BETWEEN MEASURED AND PREDICTED VALUES  
C OF DISPLACEMENTS AND VELOCITIES. H ALSO CALLS  
C CRS TO FIND THE SECOND PARTIAL DERIVATIVES FOUND  
C IN THE SECOND AND FOURTH TERMS FOR THE HESSIAN OF  
C THE OBJECTIVE FUNCTION.

C

SUBROUTINE H (X,NP,A)

INTEGER NP, NPOINT, NVAR

PARAMETER (NVAR=12, NPOINT=335)

DOUBLE PRECISION UMES(4,335), A(NVAR,NVAR)

DOUBLE PRECISION STEPSZ, X(NVAR)

DOUBLE PRECISION Q(4), DER(4,NVAR)

DOUBLE PRECISION CROSS(4,NVAR,NVAR)

COMMON /ALL/ STEPSZ,UMES

CALL DERIV (X,NP,DER)

CALL DIFF(X,NP,Q)

CALL CRS (X,NP,CROSS)

DO 6 L=1,NVAR

DO 5 M=L,NVAR

```

A(L,M)=2.0D0*(DER(1,L)*DER(1,M)/25.0D0
*           +Q(1)*CROSS(1,L,M)/25.0D0
*           +DER(2,L)*DER(2,M)+Q(2)*CROSS(2,L,M)
*           +DER(3,L)*DER(3,M)/25.0D0
*           +Q(3)*CROSS(3,L,M)/25.0D0
*           +DER(4,L)*DER(4,M)+Q(4)*CROSS(4,L,M))

```

```
5           CONTINUE
```

```
6           CONTINUE
```

```
RETURN
```

```
END
```

```
C
```

```

C   GIVEN THE PARAMETERS AND THE POINT IN QUESTION,
C   DERIV SUPPLIES THE VALUES OF THE FIRST DERIVATIVE
C   OF THE UPDATE FUNCTION FOR THE POINT UNDER
C   INVESTIGATION (NP) AND THE PREVIOUS N POINTS FOR AN
C   IMPLICIT N+1 POINT METHOD.  DERJ(K,L) IS THE
C   DERIVATIVE OF THE FUNCTION COMPONENT K WITH
C   RESPECT TO X SUB L EVALUATED AT POINT J, RELATIVE
C   TO THE POINT UNDER INVESTIGATION, NP.

```

```
C
```

```
SUBROUTINE DERIV (X,NP,Q)
```

```
    INTEGER NVAR, NP, NPOINT
```

```
    PARAMETER (NVAR=12, NPOINT=335)
```

```

DOUBLE PRECISION UMES(4,335)

DOUBLE PRECISION X(NVAR), STEPSZ, Q(4,NVAR)

DOUBLE PRECISION DERO(4,NVAR), DER1(4,NVAR)

DOUBLE PRECISION DER2(4,NVAR), DER3(4,NVAR)

DOUBLE PRECISION DER4(4,NVAR), DER5(4,NVAR)

DOUBLE PRECISION DER6(4,NVAR)

COMMON /ALL/ STEPSZ, UMES

CALL D(X,NP,DERO)

CALL D(X,NP-1,DER1)

CALL D(X,NP-2,DER2)

CALL D(X,NP-3,DER3)

CALL D(X,NP-4,DER4)

CALL D(X,NP-5,DER5)

CALL D(X,NP-6,DER6)

DO 5 L=1,4

    DO 4 M=1,NVAR

        Q(L,M) = (6.0D0*STEPSZ/840.0D0)

*           *(41.0D0*DER6(L,M)

*           +216.0D0*DER5(L,M)+27.0D0*DER4(L,M)

*           +272.0D0*DER3(L,M)+27.0D0*DER2(L,M)

```

```

*                +216.0D0*DER1(L,M)+41.0D0*DER0(L,M))
4                CONTINUE
5                CONTINUE

RETURN

END

C
C   GIVEN THE PRESENT ESTIMATE OF THE PARAMETER
C   VECTOR AND A POINT IN TIME P, SUBROUTINE D FINDS
C   THE DERIVATIVE OF THE FUNCTION VECTOR
C   (2* NO. OF DIMENSIONS OF SYSTEM BY NVAR )
C   WITH RESPECT TO ALL THE PARAMETERS, AND
C   PASSES BACK THE DERIVATIVES IN THE MATRIX DER.
C   UMES IS THE MATRIX OF MEASURED DISPLACEMENTS
C   AND VELOCITIES, COMMON TO D AND THE MAIN PROGRAM.
C   DER(I,J) IS THE DERIVATIVE OF THE FUNCTION F SUB I WITH
C   RESPECT TO THE PARAMETER J.
C
SUBROUTINE D (X, P, DER)
    INTEGER NVAR, P, NPOINT
    PARAMETER (NVAR=12, NPOINT=335)
    COMMON /ALL/ STEPSZ, UMES
    DOUBLE PRECISION STEPSZ, UMES(4,335)
    DOUBLE PRECISION X(NVAR), DER(4,NVAR)
    DER(1,1)= 0.0D0

```

```

DER(1,2)= 0.0D0
DER(1,3)= 0.0D0
DER(1,4)= 0.0D0
DER(1,5)= 0.0D0
DER(1,6)= 0.0D0
DER(1,7)= 0.0D0
DER(1,8)= 0.0D0
DER(1,9)= 0.0D0
DER(1,10)= 0.0D0
DER(1,11)= 0.0D0
DER(1,12)= 0.0D0
DER(2,1)= -UMES(1,P)
DER(2,2)= -2.0D0*UMES(2,P)
DER(2,3)= -UMES(1,P)**3
DER(2,4)= -(UMES(1,P)**2)*UMES(3,P)
DER(2,5)= -(UMES(3,P)**2)*UMES(1,P)
DER(2,6)= -UMES(3,P)**3
DER(2,7)= 0.0D0
DER(2,8)= 0.0D0
DER(2,9)= 0.0D0
DER(2,10)= 0.0D0
DER(2,11)= 0.0D0
DER(2,12)= 0.0D0
DER(3,1)= 0.0D0
DER(3,2)= 0.0D0

```

```

DER(3,3)= 0.0D0
DER(3,4)= 0.0D0
DER(3,5)= 0.0D0
DER(3,6)= 0.0D0
DER(3,7)= 0.0D0
DER(3,8)= 0.0D0
DER(3,9)= 0.0D0
DER(3,10)= 0.0D0
DER(3,11)= 0.0D0
DER(3,12)= 0.0D0
DER(4,1)= 0.0D0
DER(4,2)= 0.0D0
DER(4,3)= 0.0D0
DER(4,4)= 0.0D0
DER(4,5)= 0.0D0
DER(4,6)= 0.0D0
DER(4,7)= -UMES(3,P)
DER(4,8)= -2.0D0*UMES(4,P)
DER(4,9)= -UMES(1,P)**3
DER(4,10)= -(UMES(1,P)**2)*UMES(3,P)
DER(4,11)= -(UMES(3,P)**2)*UMES(1,P)
DER(4,12)= -UMES(3,P)**3

```

```

RETURN

```

```

END

```

C

C GIVEN THE VALUES OF THE PARAMETERS AND  
 C THE POINT IN QUESTION, CRS GIVES THE VALUES  
 C OF THE SECOND PARTIAL DERIVATIVES FOUND IN  
 C THE EXPRESSION FOR THE HESSIAN OF THE OBJECTIVE  
 C FUNCTION. THE FIRST INDEX OF A IS THE NODE  
 C NUMBER (1 FOR DISPL. AT NODE 1, 2 FOR VEL.  
 C AT NODE 1, 3 FOR DISPL. AT NODE 2, AND SO ON),  
 C THE SECOND INDEX IS THE FIRST PARAMETER OF  
 C DIFFERENTIATION, AND THE THIRD INDEX IS THE  
 C SECOND PARAMETER OF DIFFERENTIATION. FOR  
 C EXAMPLE, A(K,L,M) IS THE DERIVATIVE OF F(K)  
 C (THE KTH NUMBER FOR NODE) WITH RESPECT TO THE  
 C LTH PARAMETER, AND AGAIN WITH RESPECT  
 C TO THE MTH PARAMETER.

C

SUBROUTINE CRS (X,NP,A)

INTEGER NVAR, NP, NPOINT

PARAMETER (NVAR=12, NPOINT=335)

DOUBLE PRECISION UMES(4,335), A(4,NVAR,NVAR)

DOUBLE PRECISION X(NVAR), STEPSZ

COMMON /ALL/ STEPSZ, UMES

DO 15 L=1,4

DO 10 M=1,NVAR

DO 5 N=1,NVAR

```

          A(L,M,N)=0.0D0
5          CONTINUE
10         CONTINUE
15        CONTINUE

        RETURN

        END

C
C  GIVEN THE PRESENT VALUES FOR THE PARAMETERS AND
C  THE NUMBER OF THE DATA POINT IN QUESTION, DIFF
C  SUPPLIES THE QUANTITIES OR ERRORS IN THE EXPRESSION
C  ERR=Y(I)-Y(I+1)+G(F(Y(I))) FROM THE MULTISTEP
C  METHOD.  THE VALUE FOR G ( 2 * NDIM. COMPONENTS )
C  IS FOUND BY CALLING SUBROUTINE FCN. TO EVALUATE
C  THE FUNCTION F FROM THE TRANSFORMED
C  DIFFERENTIAL EQUATIONS.
C
        SUBROUTINE DIFF(X,NP,A)
            INTEGER NVAR, NP, NPOINT
            PARAMETER (NVAR=12, NPOINT=335)
            DOUBLE PRECISION UMES(4,335), X(NVAR), STEPSZ
            DOUBLE PRECISION A(4), F1, F2, F3, F4
            COMMON /ALL/ STEPSZ, UMES

            A(1)=UMES(1,NP-6)-UMES(1,NP)

```



```

*      +(6.0D0*STEPSZ/840.0D0)
*      *(41.0D0*F1(X,NP-6)+216.0D0*F1(X,NP-5)
*      +27.0D0*F1(X,NP-4)
*      +272.0D0*F1(X,NP-3)+27.0D0*F1(X,NP-2)
*      +216.0D0*F1(X,NP-1)
*      +41.0D0*F1(X,NP))

```

A(2)=UMES(2,NP-6)-UMES(2,NP)

```

*      +(6.0D0*STEPSZ/840.0D0)
*      *(41.0D0*F2(X,NP-6)+216.0D0*F2(X,NP-5)
*      +27.0D0*F2(X,NP-4)
*      +272.0D0*F2(X,NP-3)+27.0D0*F2(X,NP-2)
*      +216.0D0*F2(X,NP-1)
*      +41.0D0*F2(X,NP))

```

A(3)=UMES(3,NP-6)-UMES(3,NP)

```

*      +(6.0D0*STEPSZ/840.0D0)
*      *(41.0D0*F3(X,NP-6)+216.0D0*F3(X,NP-5)
*      +27.0D0*F3(X,NP-4)
*      +272.0D0*F3(X,NP-3)+27.0D0*F3(X,NP-2)
*      +216.0D0*F3(X,NP-1)
*      +41.0D0*F3(X,NP))

```

A(4)=UMES(4,NP-6)-UMES(4,NP)

```

*      +(6.0D0*STEPSZ/840.0D0)

```

```

*          *(41.0D0*F4(X,NP-6)+216.0D0*F4(X,NP-5)
*          +27.0D0*F4(X,NP-4)
*          +272.0D0*F4(X,NP-3)+27.0D0*F4(X,NP-2)
*          +216.0D0*F4(X,NP-1)
*          +41.0D0*F4(X,NP))

```

```

20          CONTINUE

```

```

          RETURN

```

```

          END

```

```

C

```

```

C  GIVEN THE PRESENT VALUES FOR THE PARAMETERS
C  AND A TIME POINT P, F1, F2, F3 AND F4 EVALUATE
C  THE QUANTITIES ON THE RIGHT IN THE
C  FUNCTION F= [Y PRIME].  UMES IS COMMON TO THE
C  MAIN PROGRAM AND THE F FUNCTIONS.

```

```

C

```

```

          DOUBLE PRECISION FUNCTION F1 (X,P)

```

```

              INTEGER NVAR, P, NPOINT

```

```

              PARAMETER (NVAR=12, NPOINT=335)

```

```

              COMMON /ALL/ STEPSZ, UMES

```

```

              DOUBLE PRECISION STEPSZ, UMES(4,335), X(NVAR)

```

```

              F1=UMES(2,P)

```

RETURN

END

DOUBLE PRECISION FUNCTION F2 (X,P)

INTEGER NVAR, P, NPOINT

PARAMETER (NVAR=12, NPOINT=335)

COMMON /ALL/ STEPSZ, UMES

DOUBLE PRECISION STEPSZ, UMES(4,335), X(NVAR)

F2=-X(1)\*UMES(1,P)-2.0D0\*X(2)\*UMES(2,P)

\* -X(3)\*UMES(1,P)\*\*3

\* -X(4)\*(UMES(1,P)\*\*2)\*UMES(3,P)

\* -X(5)\*(UMES(3,P)\*\*2)\*

\* UMES(1,P)-X(6)\*UMES(3,P)\*\*3

RETURN

END

DOUBLE PRECISION FUNCTION F3 (X,P)

INTEGER NVAR, P, NPOINT

PARAMETER (NVAR=12, NPOINT=335)

COMMON /ALL/ STEPSZ, UMES

DOUBLE PRECISION STEPSZ, UMES(4,335), X(NVAR)

F3=UMES(4,P)

RETURN

END

DOUBLE PRECISION FUNCTION F4 (X,P)

INTEGER NVAR, P, NPOINT

PARAMETER (NVAR=12, NPOINT=335)

COMMON /ALL/ STEPSZ, UMES

DOUBLE PRECISION STEPSZ, UMES(4,335), X(NVAR)

F4=-X(7)\*UMES(3,P)-2.0D0\*X(8)\*UMES(4,P)

\* -X(9)\*UMES(1,P)\*\*3

\* -X(10)\*(UMES(1,P)\*\*2)\*UMES(3,P)

\* -X(11)\*(UMES(3,P)\*\*2)\*

\* UMES(1,P)-X(12)\*UMES(3,P)\*\*3

RETURN

END

C GIVEN A SYSTEM OF LINEAR ALGEBRAIC EQUATIONS  
C AX=B, SUBROUTINE GAUSS FINDS A SOLUTION VECTOR X.  
C IN THESE PROBLEMS, A IS THE NVAR\*NVAR HESSIAN  
C AND B IS THE NVAR COMPONENT GRADIENT VECTOR.  
C THE SOLUTION X IS USED AS THE STEP TO THE NEW

C VALUES FOR THE PARAMETERS SOUGHT.

C

```
SUBROUTINE GAUSS(A,B,XG,N,MAXDIM,IERROR,RNORM)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION A(MAXDIM,MAXDIM),B(MAXDIM),XG(MAXDIM)
DIMENSION AUG(50,51)
NM1=N-1
NP1=N+1
DO 2 I=1,N
    DO 1 J=1,N
        AUG(I,J)=A(I,J)
1    CONTINUE
    AUG(I,NP1)=B(I)
2    CONTINUE
DO 8 I=1,NM1
    PIVOT=0.0D0
    DO 3 J=I,N
        TEMP=DABS(AUG(J,I))
        IF (PIVOT.GE.TEMP) GOTO 3
        PIVOT=TEMP
        IPIVOT=J
3    CONTINUE
    IF (PIVOT.EQ.0.0D0) GOTO 13
    IF (IPIVOT.EQ.I) GOTO 5
    DO 4 K=I,NP1
```

```

        TEMP=AUG(I,K)
        AUG(I,K)=AUG(IPIVOT,K)
        AUG(IPIVOT,K)=TEMP
4      CONTINUE
5      IP1=I+1
        DO 7 K=IP1,N
            Q=-AUG(K,I)/AUG(I,I)
            AUG(K,I)=0.0D0
            DO 6 J=IP1,NP1
                AUG(K,J)=Q*AUG(I,J)+AUG(K,J)
6          CONTINUE
7      CONTINUE
8      CONTINUE
        IF (AUG(N,N).EQ.0.0D0) GOTO 13
        XG(N)=AUG(N,NP1)/AUG(N,N)
        DO 10 K=1,NM1
            Q=0.0D0
            DO 9 J=1,K
                Q=Q+AUG(N-K,NP1-J)*XG(NP1-J)
9          CONTINUE
            XG(N-K)=(AUG(N-K,NP1)-Q)/AUG(N-K,N-K)
10     CONTINUE
        RSQ=0.0D0
        DO 12 I=1,N
            Q=0.0D0

```

```
      DO 11 J=1,N
      Q=Q+A(I,J)*XG(J)
11     CONTINUE
      RESI=B(I)-Q
      RMAG=DABS(RESI)
      RSQ=RSQ+RMAG**2
12    CONTINUE
      RNORM=DSQRT(RSQ)
      IERROR=1
      RETURN
13    IERROR=2
      RETURN
      END
```

## Vita

The author was born in Sacramento, California, on March 31, 1964. At present, he resides in Norfolk, Virginia. After receiving a dual degree in Physics and Mathematics from Virginia Polytechnic Institute and State University in 1987, he commenced graduate school there in Aerospace Engineering to pursue a Master of Science degree, of which this thesis is a result.

*J. Benji Pomeroy*