IDENTIFICATION OF NONLINEAR DYNAMICAL SYSTEMS BY A MODIFIED DIFFERENTIAL APPROXIMATION TECHNIQUE

By

RAJAMOULI GUNDA

Bachelor of Engineering
Osmania University
Hyderabad, India
1966

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1969

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1971

IDENTIFICATION OF NONLINEAR DYNAMICAL SYSTEMS BY A MODIFIED DIFFERENTIAL APPROXIMATION TECHNIQUE

Thesis Approved:

Karl N. Reid
Thesis Adviser

James E. Bose

Henry R. Sebesta

Charles M. Bacon

D. D. Durham
Dean of the Graduate College

ACKNOWLEDGMENT

I would like to express my appreciation for the help given by the following people:

Professor Karl N. Reid, my major adviser and chairman of the thesis committee, who provided invaluable encouragement, counsel, criticism and guidance throughout my graduate program.

Professors Henry R. Sebesta, James E. Bose and Charles M. Bacon, other members of the committee, who provided so much of their time in encouraging me throughout the preparation of the thesis. Their suggestions, cooperation and criticism were very helpful.

I would like to acknowledge the financial support received from the School of Mechanical Engineering as a graduate research assistant and from the Center for Systems Science (under NSF Grant GU-3160).

I wish to thank Bob Woods, Scott Babcock and Chris Smith for their encouragement.

Finally I want to thank my brother-in-law who supported me in many respects since my childhood; my father-in-law who encouraged me for graduate studies abroad; my mother for her blessings; my wife, Sujatha, and daughter, Madhavi, whose patience and affection were of great help.

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page

NOMENCLATURE

A	Coefficient matrix of the linearized differential equations (associated with the state vector)
B	Coefficient matrix of the linearized differential equations (associated with the input vector)
C	A set of model coefficients (usually an array)
d_i	Standard deviation of the random variable R_i
E	Computational effort without decomposition
E_d	Computational effort with decomposition
F	Vector of Functions (f_1 and f_2)
I	Number of pairs of initial conditions
J	Performance index
K	Constant of proportionality
K_c	Number of coefficients determined at a time
K_g	Number of grid divisions in the X-U hyperspace which is considered to determine K_c coefficients
K_p	Number of subperformance indices (PSSE)
m_k	Number of levels into which the k-th input is divided
MSE	Modified sum of squared errors
n_i	Number of levels into which the i-th state is divided
N_c	Number of model coefficients
N_g	Number of grid divisions in the total X-U hyperspace

NSR	Noise to signal ratio (0 to 1)
p_t	Tank pressure (Example 6)
p_{xi}	Degree of the polynomial in x_i
p_{ui}	Degree of the polynomial in u_i
PSSE	Partial sum of squared errors
R	Vector of three random variables
s	Laplace transform operator
S	Vector of transformed (averaged) state variables
SSE	Sum of squared errors
t_0	Initial time
t_f	Final time
t_s	Settling time of the system
U	Vector of inputs (u_1 and u_2)
U_{max}	Vector of maximum limits of the inputs (u_1 and u_2)
U_{min}	Vector of minimum limits of the inputs (u_1 and u_2)
U_{op}	Vector of values of the inputs (u_1 and u_2) at the operating point
U_{step}	Vector of values of the steps in the inputs
v_c	Capacitor voltage
X	Vector of state variables (x_1 and x_2)
X_{max}	Vector of maximum limits of the states (x_1 and x_2)
X_{min}	Vector of minimum limits of the states (x_1 and x_2)
X_{op}	Values of the states at the operating point
X_{ss}	Values of the steady-state vector
Z	Vector of transformed (averaged) input variables
$\partial F / \partial U$	Partial derivative of F with respect to U
$\partial F / \partial X$	Partial derivative of F with respect to X

CHAPTER I

INTRODUCTION

If a mathematical model which describes the input-output relation of a physical system is known, off-line tests can be conducted efficiently and economically on the model without disturbing the system. For example, an optimum input, which causes the system to produce a desired output, can be determined by perturbing the input to the mathematical model and observing the output. Such a model can be found by considering the fundamental physical phenomena governing the system. This procedure becomes difficult for complex systems. A system model can also be found by using one of many available identification techniques. However, most identification techniques utilize large computational effort and/or some <u>a priori</u> knowledge about the system. The technique presented in this thesis does not require <u>a priori</u> knowledge. In addition, the computational requirements are reduced.

System Identification

System identification is the process of determining a suitable mathematical model for a system from experiments conducted on the system. An impulse or step response is sufficient to identify a completely controllable linear system. Such a simplified approach does not exist for nonlinear systems. The process of identifying nonlinear systems consists of formulation of a system model with free parameters and determination of these model parameters by minimizing some performance index. Figure 1 illustrates the general procedure of an identification technique. The model responses to test inputs need not be simulated for all the techniques. In some techniques, the model parameters are determined uniquely.

Review of Literature

Identification techniques which are applicable to nonlinear systems can be broadly divided into the following four classes; 1) Functional Power Series, 2) Pattern Recognition, 3) State-Space(with known model form), and 4) State-Space(with unknown model form). Only the relevant techniques are discussed below.

Functional power series and pattern recognition techniques are based on the fact that any system operates on an input over certain interval of time and produces an output. The identification problem is to find the present value of the output $y(t_1)$ as a function of the input $u(t)$ over an interval $t_1-t_s \leq t \leq t_1$, where t_s is the settling time.

Kwatny and Schen (19) represented nonlinear systems by functional power series models as follows:

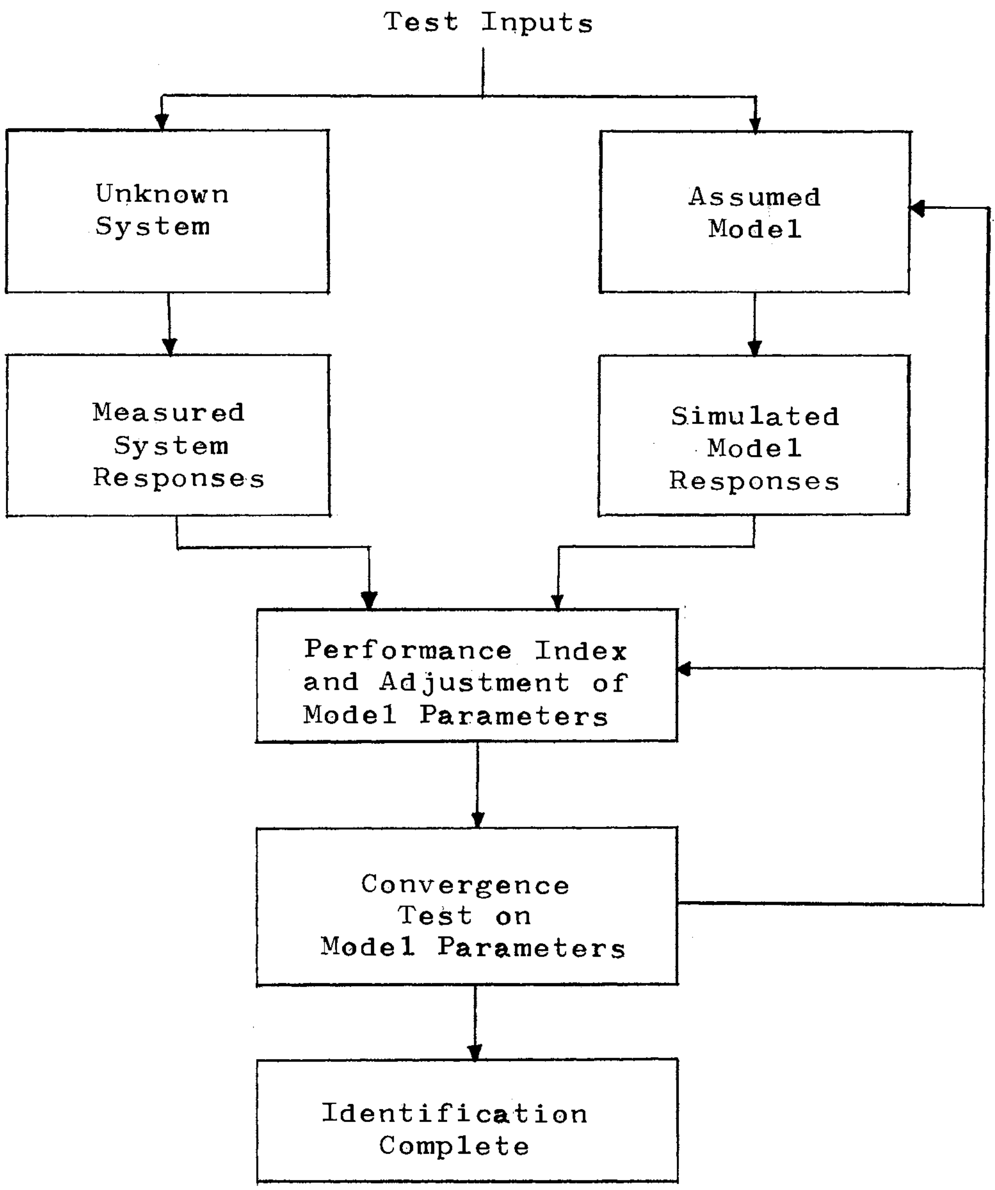


Figure 1. General Procedure of an Identification Technique

$$y(t_1) = a_0 + \sum_i a_i y_i(t_1) + \sum_{i,j} a_{ij} y_i(t_1) y_j(t_1) + \dots \qquad (1-1)$$

$$y_i(t_1) = \int_0^{\infty} h_i(t) u(t-t_1)\, dt \qquad (1-2)$$

where, $h_i(t)$; $i=1,2,\dots$ are some orthogonal functions.

In a technique by Arozullah (2) the unknown system is represented by a single-input, multi-output linear part followed by a multi-input, single-output, zero-memory nonlinear part. The linear part is formed by expanding the past history of the input in a Fourier series in terms of a set of orthonormal functions. The coefficients of this series are inputs to the nonlinear part the output of which is a multidimensional gating function and a piecewise multidimensional linear function of these coefficients.

If the past of the input is sampled at n instants and quantized into m levels, there will be m^n possible input patterns. One method of identification is to tabulate the output for all the input patterns. This method, called the table lookup method, requires prohibitively large computer memory. The memory requirements can be reduced by using pattern recognition techniques as discussed below.

Miller and Roy (21) proposed to measure certain feature of the input instead of the entire pattern. From n samples of the input pattern, only k samples are considered as a feature. The method reduces the memory requirement from m^n to $m^k(n)!/(n-k)!(k)!$. The memory requirement is further reduced at the expence of accuracy by a "mode learning machine" technique proposed by Roy and Schley (26).

As before, the past history of the input is sampled at n instants to form an n-dimensional pattern space. The output is quantized into p levels each of which is called a 'category'. A category which can be obtained from j patterns is assumed to be obtained from only k 'prototype patterns' where, k is less than j. To determine the output $y(t_1)$, the input u(t) is sampled at n instants over the interval $t_1 - t_s \leq t \leq t_1$ and the closest prototype pattern is selected.

Most dynamic systems can be adequately described in state variable notation by a set of first-order ordinary differential equations of the form:

$$\dot{X} = F(X,U,P) \tag{1-3}$$

where, X, U and P are the state, the input and the parameter vectors respectively. Identification techniques based on state-space approaches require a model with known forms of the differential equations. Usually an iterative method is required to find the model parameters. Two state-space techniques are discussed below.

A quasilinearization technique as presented by Bellman, Kalaba and Sridhar (5), Sage and Eisenberg (27) and Allison (1), can be used to determine the parameters in Equation (1-3) by minimizing a general error squared performance index. This is accomplished by solving a sequence of linear differential equations. If this sequence converges, the resulting parameters are optimum. A major weakness of this technique is that the above sequence may diverge.

The differential approximation technique as presented by Sage (27), Bellman, Kalaba and Sridhar (5) and Bose (6) utilizes the fact that the correct parameters must minimize the following performance index:

$$PI = \int_{t_0}^{t_f} (\dot{X}-F(X,U,P))^T (\dot{X}-F(X,U,P))\, dt \qquad (1\text{-}4)$$

where, the superscript T stands for the transpose of the vector. The advantage of this technique is that the model parameters can be found by solving a set of nonlinear algebraic simultaneous equations instead of repeatedly solving a set of differential equations.

The identification techniques discussed above require knowledge of the settling time and normal operating input-output records of the system. The input must be general enough to cause the system to respond over the entire X-U hyperspace of interest. The data required for identification can be reduced by conducting a specific set of tests on the system.

The author (15) has proposed an alternate state-space identification technique for stationary deterministic systems. The functions F(X,U,P) in Equation (1-3) are assumed to be polynomials. The system is subjected to various pulse inputs with various initial conditions on the system. The assumed model is also subjected to the same inputs. The polynomial coefficients are determined by matching the simulated model responses to the measured system responses in some sense. These coefficients are allowed to depend on the pulse amplitude and the system initial condition. The application of this technique to single-input, first-order systems gave very satisfactory results.

In summary, the functional power series and pattern recognition techniques do not utilize *a priori* model form based on the physics of the system, but require considerable experimental data and computational effort. Also, these techniques are limited to single-input systems and do not allow determination of linearized differential equations which are valid in the vicinity of an operating point. In contrast, some of the state-space techniques which utilize an *a priori* system model require small computational effort. The modified differential approximation technique developed in this thesis assumes a generalized polynomial, tabular or mixed form of the model. The amount of test data and the computational effort required for identification are reduced considerably by conducting a specific set of tests on the system.

Scope of Thesis

The identification technique developed in this thesis is applicable to stationary nonlinear systems which can be described by lumped parameter models. The method is formulated and evaluated for first-order systems with one or two inputs and for second-order systems with one input. The effect of additive, zero-mean, Gaussian noise in the test inputs and in the measured system responses, on the results of identification is investigated for single-input, first-order systems. The technique is limited to zero-memory nonlinearities. The technique is applicable to multiple-input, higher-order systems, but no evaluation of doing so is presented. The application of the technique is illustrated through a number of example systems with known mathematical models and two real physical systems. The necessary computer tools are developed for identification of systems, for prediction of system response and for determination of linearized differential equations valid in the small about an operating point. The efforts required for identification and prediction of system response are determined.

Outline of Identification Technique

The identification problem is to specify the test conditions which are feasible in practice and to find a system model using the measured responses for the above test conditions. The modified differential approximation technique is summarized below.

Selection of the Model Form

The system is modeled by the following vector differential equation

$$\dot{X} = F(X,U,C) \tag{1-5}$$

where, C is a set of N_c model coefficients. The vector function F(X,U,C) can be assumed to be: 1) A vector of polynomials in X and U(polynomial form), 2) A vector of tables of numbers in terms of X and U(tabular form), or 3) A vector of polynomials in X and tables of numbers in terms of U (mixed form). The coefficients of the polynomials and/or the numbers in the tables are called the model coefficients.

Specification of Test Conditions

A specific set of tests must be conducted on the system to ensure that the system responds over the entire X-U hyperspace of interest. With proper selection of the range of step inputs and various initial conditions on the system, the data required for identification is minimized.

Measurement and Processing of Data

The identification technique requires the system responses(states and first derivatives of the states) for all the test conditions. When only X(states) is available, $\dot{X}$ can be obtained by numerical differentiation. The time responses X and $\dot{X}$ must be sampled and stored. The sampling interval depends on the characteristics of the responses.

Determination of Model Coefficients

The model coefficients can be found by minimizing the following discrete performance index:

$$J = \sum_{k=1,N_d} (\dot{X}(k)-F(k))^T(\dot{X}(k)-F(k)) \qquad (1\text{-}6)$$

$$F(k) = F(X(k),U(k),C)$$

where, X(k), $\dot{X}(k)$ and U(k) are the k-th stored values of X, $\dot{X}$ and U respectively and N_d is the total number of stored data points. The computational effort required to solve for the optimum model coefficients which minimize J can be reduced considerably by defining a modified performance index.

Full details of the above steps are presented in Chapter II. The necessary steps in identification and the applications of the technique to a number of examples are presented in Chapter III. A qualitative comparison of the modified differential approximation technique with other identification techniques and the conclusions are included in Chapters IV and V respectively. The necessary computer tools are presented in the appendices.

Summary of Results

The application of the modified differential approximation technique to a number of systems with known models and to real physical systems yielded model responses which were within 3% of the system responses.

The technique yields a model which is valid for the complete range of inputs and system initial conditions. The model may be used to compute the response to any arbitrary input(s) within the range of the test data. Also, the model allows determination of linearized differential equations valid in the small about any operating point. Of the three model forms, the mixed form(polynomial in X and tabular in U) requires the least computational effort for identification, the tabular form is most efficient for the prediction of system response and the polynomial form gives the most accurate results requiring the minimum number of model coefficients.

The modified differential approximation technique is inherently a smoothing process and is found to be insensitive to additive, zero-mean, Gaussian noise in the test inputs and in the measured system responses(states and the first derivatives of the states).

CHAPTER II

THE MODIFIED DIFFERENTIAL APPROXIMATION TECHNIQUE

The development of the identification technique presented in this chapter is divided into the following phases:

1. The problem
2. Basic assumptions
3. The model
4. Test conditions
5. Performance index
6. Model coefficients.

The Problem

The problem of system identification considered in this thesis is:

1. To specify the test conditions for nonlinear first-order systems with one or two inputs and for nonlinear second-order systems with one input. The test inputs should be feasible in practice.
2. To identify a mathematical model for the unknown system, best in a least squares sense, using the system responses for the above test conditions. The responses include both the state vector and the first derivatives of the state vector.

The identified model should allow prediction of the system responses to any arbitrary input(s) other than the test inputs, and the determination of the linearized differential equations of the model which are valid in the vicinity of an operating point. The identification technique should be insensitive to additive, zero-mean, Gaussian noise in the test inputs and in the measured system responses.

Basic Assumptions

The identification technique developed in this thesis assumes that the nonlinear systems to be identified can be adequately described by a set of nonlinear ordinary differential equations with constant coefficients.

Although the technique is applicable to multiple-input, higher-order systems, first-order systems with one or two inputs and second-order systems with one input are considered in detail. Figures 2 and 3 show the class of nonlinear systems to which this technique is applicable. The knowledge of the forms of the nonlinear functions N_{u1}, N_{u2}, N_{f1}, N_{f2}, N_{b1} and N_{b2} is not required for identification. It is assumed that these nonlinear functions can be approximated by polynomials.

When the output of each of the above zero-memory, nonlinear elements is either a monotonically increasing or monotonically decreasing function of its input, the system under consideration will not have multiple steady-state responses.

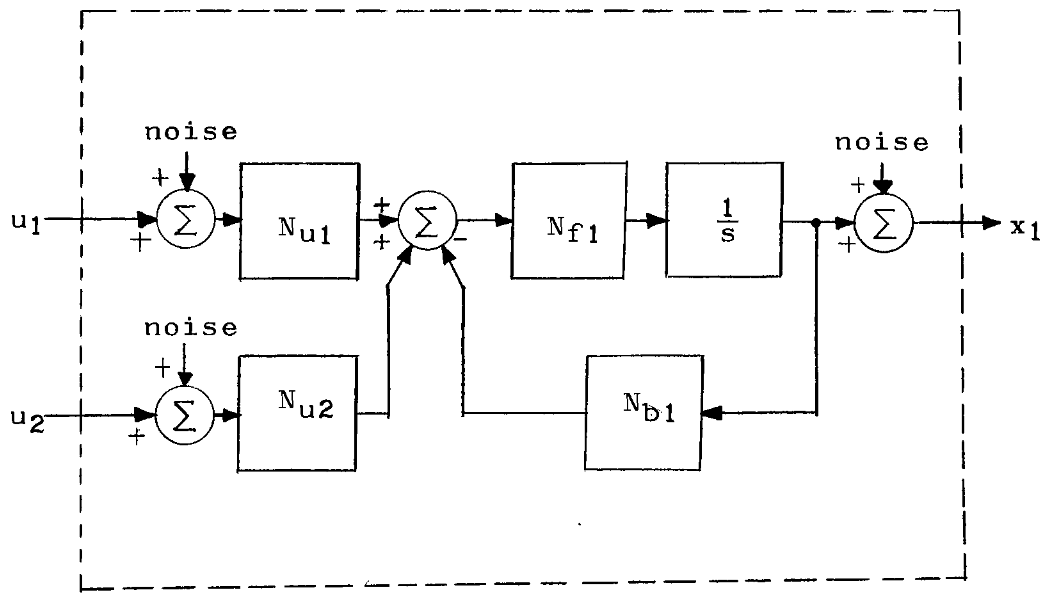


Figure 2. A General Nonlinear First-Order System With One or Two Inputs

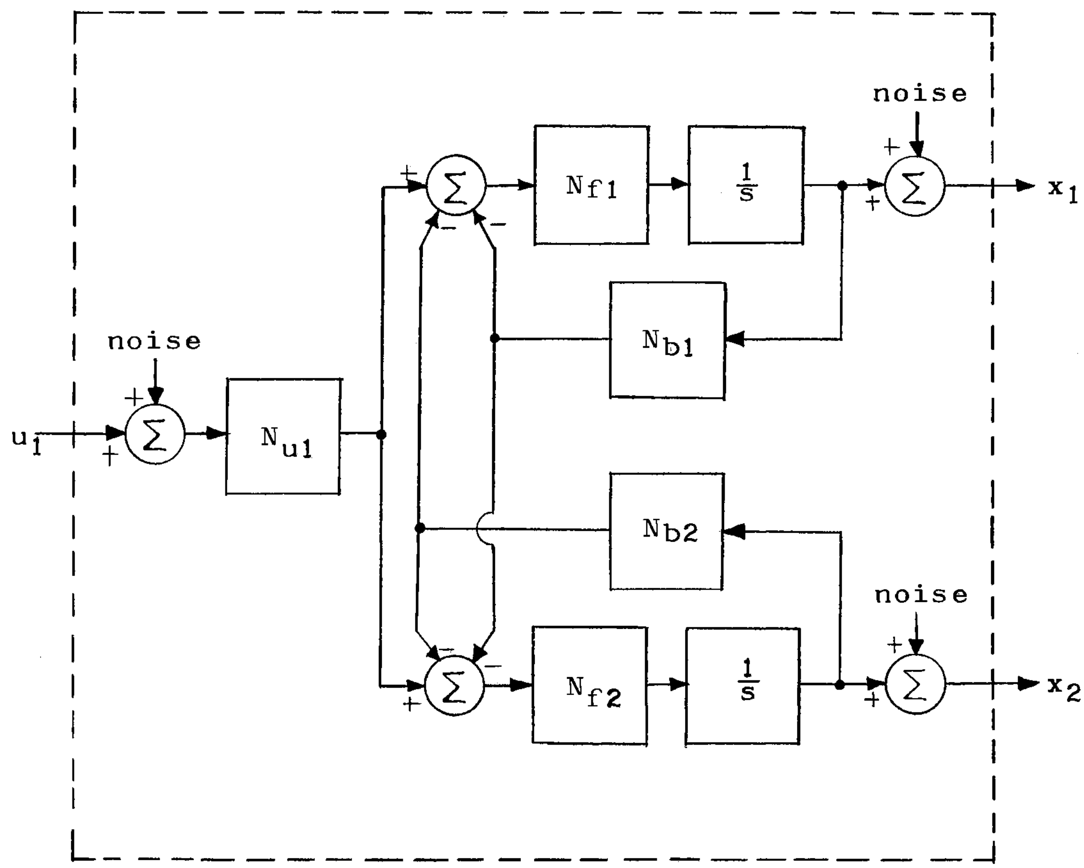


Figure 3. A General Nonlinear Second-Order System With One Input

The Model

The system is modeled by the following vector differential equation:

$$\dot{X} = F(X,U,C) \qquad (2\text{-}1)$$

where, X and U are the state and the input vectors, C is a set of N_c model coefficients. The function vector F(X,U,C) can be: 1) A vector of polynomials in X and U (polynomial form), 2) A vector of tables of numbers in terms of X and U (tabular form), or 3) A vector of polynomials in X and tables of numbers in terms of U (mixed form).

The coefficients of the polynomials and/or the numbers in the tables are called the model coefficients. For convenience, these coefficients will be represented by multidimensional arrays. The nature of the model coefficients for the three forms is explained in detail below for a first-order, single-input system.

Consider the following model coefficient matrix:

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1j} & \cdots \\ c_{21} & c_{22} & \cdots & c_{2j} & \cdots \\ \cdot & & & & \\ \cdot & & & & \\ c_{i1} & c_{i2} & \cdots & c_{ij} & \cdots \\ \cdot & & & & \\ \cdot & & & & \end{bmatrix}$$

Let p_{x1} and p_{u1} be the degrees of the polynomials in x_1 and u_1. Let m_1 and n_1 be the numbers of levels into which the input u_1 and the state x_1 are divided. For a polynomial form of the model the elements c_{ij} of the above matrix will be the coefficients of the following differential equation:

$$\dot{x}_1 = \sum_{i=1}^{p_{u1}+1} \sum_{j=1}^{p_{x1}+1} c_{ij} u_1^{i-1} x_1^{j-1}. \qquad (2\text{-}2)$$

For a tabular form, the element c_{ij} will be the actual value of $\dot{x}_1$ at $x_1 = x_{1j}$ and $u_1 = u_{1i}$, where

$$x_{1j} = x_{1min} + (x_{1max} - x_{1min})(j-1)/(n_1-1) \qquad (2\text{-}3)$$

$$u_{1i} = u_{1min} + (u_{1max} - u_{1min})(i-1)/(m_1-1). \qquad (2\text{-}4)$$

For a mixed form, the elements c_{ij} in the i-th row will be the coefficients of the differential equation

$$\dot{x}_1 = \sum_{j=1}^{p_{x1}+1} c_{ij} x_1^{j-1}. \qquad (2\text{-}5)$$

Note that the $p_{x1}+1$ coefficients, c_{ij}, represent the system for a constant (step) input of $u_1 = u_{1i}$.

After the model coefficients are determined, the values of $\dot{X}$ for known X and U can be obtained by evaluating the polynomials and/or by interpolating using the numbers in the tables.

Test Conditions

Some identification techniques can use normal operating input-output records of the system. However, for those techniques which do not require the form of the model differential equation, the input must be general enough to cause the system to respond over the complete X-U hyperspace of interest. This section describes a specific set of tests which cause the system to respond over the range of interest. In this latter case, the amount of test data and the computational effort required for identification can be reduced. As explained later in this chapter, the use of the specified test conditions permits decomposition of the performance index which results in further reduction in the computational effort.

There exist appropriate test conditions for any higher-order, multiple-input systems. However, the difficulty of performing these tests increases with the order of the system and with the number of the inputs. The test conditions are outlined below for first-order systems with one or two inputs and second-order systems with one input.

First-Order Systems

Consider the two extreme initial conditions on a single-input, first-order system as follows: 1) $x_1(0)=x_{1min}$ and 2) $x_1(0)=x_{1max}$. If the total range in the input u_1 (u_{1min} to u_{1max}) is divided into m_1 levels, $2m_1$ step response tests will cause the system to respond over the complete x_1-u_1 plane of interest. These tests can be classified into m_1 groups of two tests each. In each group the initial conditions on the system for the first and the second test are respectively x_{1min} and x_{1max}. The amplitude of the step input in the i-th group is u_{1i} where, $u_{1i} = u_{1min} + (u_{1max} - u_{1min})(i-1)/(m_1-1)$. Figure 4 shows the responses of the system for one of the m_1 groups of tests. Note that when $u_1 = u_{1min}$ in the first group, the system response for one initial condition $x_1(0) = x_{1max}$ will cover the total range in x_1 and vice versa in the last group.

For dual-input, first-order systems the total ranges in the two inputs u_1 and u_2 are respectively divided into m_1 and m_2 levels. The same two initial conditions on the system are considered. There will be $2m_1m_2$ step response tests which will cause the system to respond over the region of interest in x_1-u_1-u_2 space. These tests can be classified into m_2 groups each of which contains $2m_1$ tests. Each group can be further classified into m_1 subgroups of two tests each. These m_1 subgroups are the same as the m_1 groups for a single-input system as discussed above, except the second input u_2 in the j-th group is a step of amplitude u_{2j}, where $u_{2j} = u_{2min} + (u_{2max} - u_{2min})(j-1)/(m_2-1)$. In order to perform each of the above tests, it is necessary to obtain the two step inputs (u_{1i} and u_{2j}) simultaneously.

The test conditions can be generalized for a multiple-input, first-order system. If there are M inputs, there

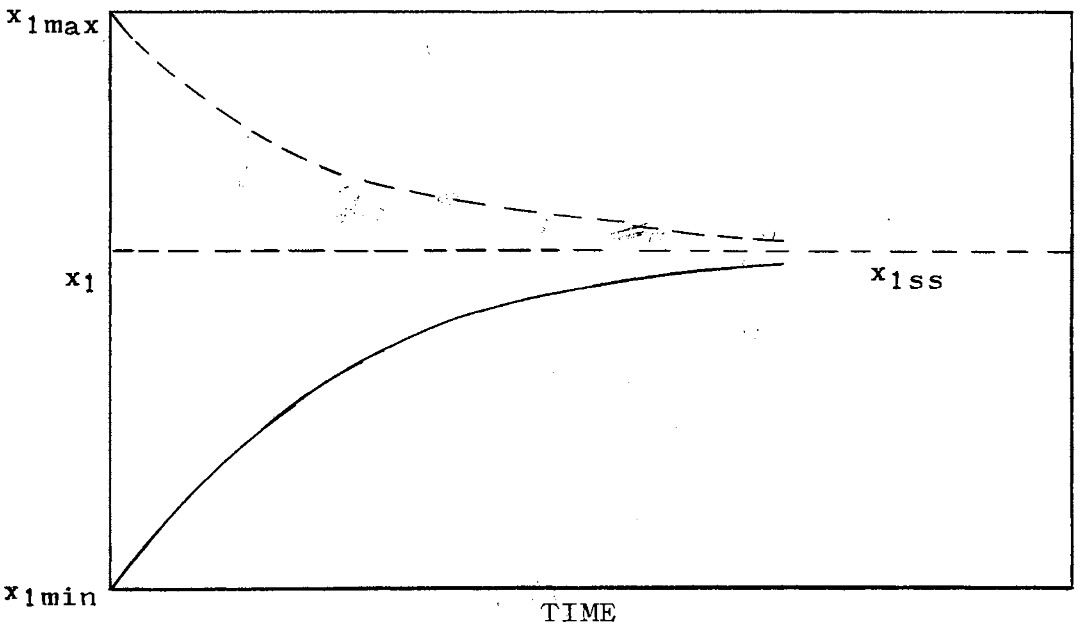


Figure 4. Responses of a First-Order System to a Typical Test Input (Step) With the Two Extreme Initial Conditions

will be $2m_1m_2...m_k...m_M$ tests. Note that the total range in the k-th input is divided into m_k levels.

Second-Order Systems

Consider the following two step response tests and the initial conditions on the system: 1) $u_1=u_{1max}$, $x_1(0)=x_{1min}$ and $x_2(0)=0$; and 2) $u_1=u_{1min}$, $x_1(0)=x_{1max}$ and $x_2(0)=0$. The curves ABC and CDA in Figure 5 are the portions of the x_1-x_2 plane responses of the system for the above two tests. The path ABCDA is defined as the "locus of initial conditions". This locus encompasses the total range in the x_1-x_2 plane which can be covered by the system responses to any allowable input. Note the above two tests are required to establish the test conditions for a second-order system.

The total range in the input u_1 is divided into m_1 levels as before. If I pairs of initial conditions are chosen along the locus of initial conditions, Im_1 step response tests will cause the system to respond over the complete region of interest in the x_1-x_2-u_1 space. These tests can be classified into m_1 groups. In the j-th group of I tests, the input is a step of amplitude u_{1j}. The initial conditions are the corresponding I pairs chosen along the locus. These I pairs need not be the same in number or value for each of the m_1 groups. The curves emanating from the I points along the locus represent the system responses for one of the m_1 groups of I tests. Note that in the first group when $u_1=u_{1min}$, the pairs of initial conditions may be

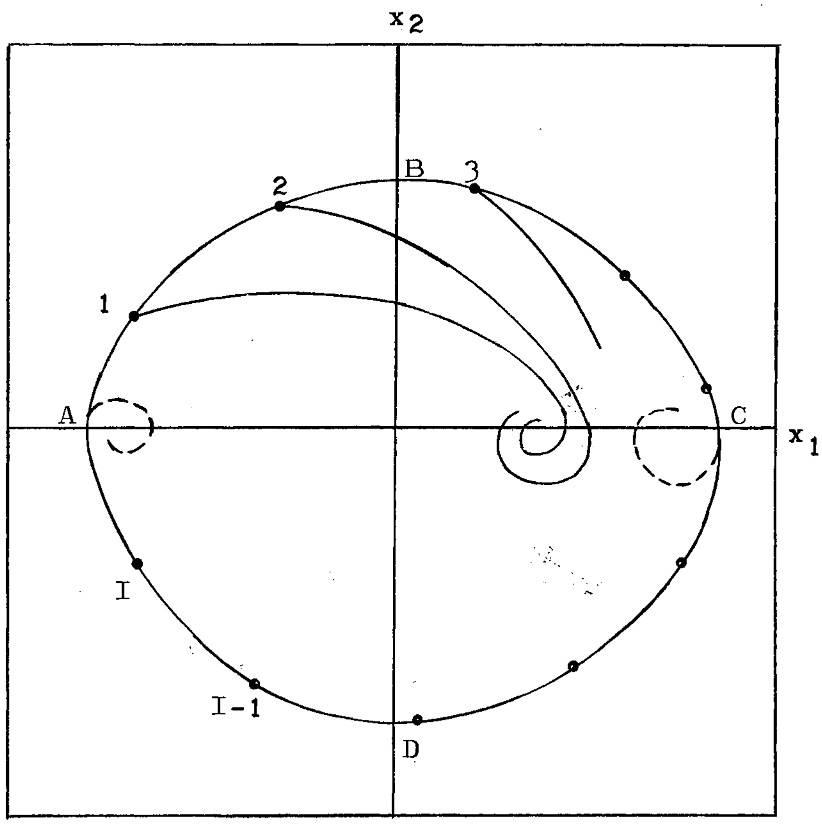


Figure 5. Responses of a Second-Order System to Typical Test Inputs (Steps) With Various Initial Conditions Along the Locus (ABCDA)

chosen only along the upper part of the locus (ABC). When $u_1 = u_{1max}$ in the last group the lower part may be used.

Generalization of the above result to an M-input, second-order system gives $Im_1 m_2 \cdot\cdot m_k \cdot\cdot m_M$ tests, where the k-th input is divided into m_k levels. Note that a system with no inputs can be considered as a single-input system with $m_1 = 1$ and $u_1 = 0$.

Performance Index

The inputs and the time responses of the system for all the test conditions are sampled, stored and numbered from 1 through N_d. A sum of squared errors (SSE) is defined as

$$SSE = \sum_{k=1,N_d} (\dot{X}(k) - F(k))^T (\dot{X}(k) - F(k)) \qquad (2\text{-}6)$$

$$F(k) = F(X(k), U(k), C).$$

The computational effort required to determine the model coefficients which minimize the above performance index is directly proportional to the number of data points. This effort can be reduced considerably by defining a modified sum of squared errors (MSSE) and finding the near optimal model coefficients. The entire X-U hyperspace of interest is divided into a multidimensional grid. All the individual grids are numbered from 1 through N_g, where N_g is the number of grid divisions. New variables $S(i)$, $\dot{S}(i)$ and $Z(i)$ are defined respectively as the average values of all the stored data points $X(j)$, $\dot{X}(j)$ and $U(j)$, $j=1,N$, which fall in the i-th individual grid. The modified sum of squared errors is defined as,

$$MSSE = \sum_{k=1,N_g} (\dot{S}(k)-F(k))^T(\dot{S}(k)-F(k)) \qquad (2\text{-}7)$$

$$F(k) = F(S(k),Z(k),C).$$

Note that the computational effort is reduced by a factor of N_g/N_d. The coefficients which minimize the MSSE satisfy the necessary condition,

$$\frac{\partial(MSSE)}{\partial C} = 0. \qquad (2\text{-}8)$$

When the vector function F is a vector of polynomials in X and U, the above equation contains N_c linear simultaneous algebraic equations in N_c unknown coefficients. The effort required to solve these equations is found to be approximately proportional to the square of the number of unknown coefficients. This effort can be reduced further by decomposing the MSSE into subperformance indices and determining fewer coefficients at a time. As discussed earlier in this chapter, the specified test conditions can be classified into groups and subgroups. A separate subperformance index may be defined for each group or subgroup.

The computational effort required to determine all the model coefficients is directly proportional to the following three factors: 1) The square of K_c, the number of the model coefficients determined at a time; 2) The number of individual grids, K_g, in the X-U hyperspace considered to determine the above coefficients; and 3) The number of the sub-performance indices, K_p, into which the MSSE is decomposed. Thus, the computational effort, E, can be computed as,

$$E = K\, K_c^2 K_g K_p. \qquad (2\text{-}9)$$

where, K is a constant of proportionality.

The decomposition of the MSSE and consequent saving in the computational effort can be illustrated for a single-input, first-order system. Let p_{x1} and p_{u1} be the degrees of the polynomials in x_1 and u_1. Let the total ranges in x_1 and u_1 be divided into n_1 and m_1 levels respectively.

When the MSSE is directly minimized, a polynomial form of the model is obtained. All the model coefficients are determined at one time. The computational effort, E, required to minimize the MSSE is computed from Equation (2-9).

$$K_c = N_c = (p_{x1}+1)(p_{u1}+1)$$

$$K_g = N_g = (n_1-1)(m_1-1)$$

$$K_p = 1$$

$$\therefore \quad E = K((p_{x1}+1)(p_{u1}+1))^2(n_1-1)(m_1-1)$$

When the decomposed MSSE is minimized, a mixed form of the model (polynomial in x_1 and tabular in u_1) is obtained. The MSSE is decomposed into m_1 partial sums of squared errors (PSSE) as follows:

$$MSSE = PSSE_1 + PSSE_2 + \ldots + PSSE_{m1}$$

$$PSSE_i = \sum_{k=1,(n_1-1)} (\dot{s}_1 - f(s_1, c_i))^2$$

where, s_1 is the averaged x_1. In the i-th group of tests, the amplitude of the step input is constant. Therefore, the function f is assumed to be a polynomial in x_1 alone. The coefficients of the polynomial are subscripted to denote that this set of coefficients represent the system for one constant (step) input of u_{1i}. The i-th row of the coefficients in the tabular form of the model is found when $PSSE_i$ is minimized using the system responses for the i-th group of tests.

The computational effort with decomposition, E_d, from Equation (2-9) is,

$$K_c = (p_{x1}+1)$$

$$K_g = (n_1-1)$$

$$K_p = m_1$$

$$\therefore \quad E_d = K(p_{x1}+1)^2(n_1-1)m_1$$

Thus, the ratio of the computational efforts is,

$$E/E_d = (p_{u1}+1)^2 (m_1-1)/m_1 .$$

When $p_{u1} = 4$ and $m_1 = 10$, the ratio is $45/2$.

In summary, the advantages of determining the model coefficients by minimizing the decomposed MSSE are the following: 1) The MSSE involves averaging which is a smoothing technique. Also, the determination of the model coefficients by minimizing the sum of squared errors is a smoothing technique. Because of these two smoothing processes the identification technique is insensitive to additive, zero-mean noise in the measured system responses; 2) Determination of the model coefficients is considerably faster; and 3) Numerical round-off errors are minimized by determining fewer coefficients at a time.

Model Coefficients

The optimum model coefficients which minimize the undecomposed or the decomposed MSSE are uniquely determined by solving system(s) of linear simultaneous algebraic equations. Iterations, as required in other techniques are avoided. When the undecomposed MSSE is used, a polynomial form of the model is obtained. When the decomposed MSSE is used, a mixed form of the model (polynomial in X and tabular in U) is obtained. The model coefficients of one form can be generated from those of the other form. To obtain a tabular form from a polynomial form, the polynomials are evaluated at various points. To obtain a polynomial form from a tabular form, least squares fitting is used.

The computer tools presented in Appendix A can be used to determine the coefficients of a mixed form of the model (polynomial in X and tabular in U). However, the coefficients of a polynomial form or a tabular form can be obtained using the conversion subroutine presented in Appendix B.

CHAPTER III

APPLICATIONS OF THE TECHNIQUE

In this chapter the applications of the identification technique are discussed and illustrated through examples.

Necessary Steps in Identification

The computer tools developed in this thesis are based on the minimization of the decomposed MSSE. The identified model form is polynomial in the state(s) and tabular in the input(s). Figure 6 illustrates the procedure followed with both identification programs, SYSID1 and SYSID2 (see Appendix A). Data from system responses for each group of the tests is read in, smoothed if necessary before differentiation, and processed (averaged over the grid divisions in the X-U hyperspace). Then, the model coefficients which minimize the decomposed MSSE are determined by solving system(s) of linear simultaneous algebraic equations. The necessary steps for the system identification are listed below.

1. Specify the region of interest in the X-U hyperspace by defining the minimum and the maximum limits on the state(s) and on the input(s).
2. Specify the numbers of levels n_1, n_2, m_1, and m_2 into which the total ranges in x_1, x_2, u_1 and u_2

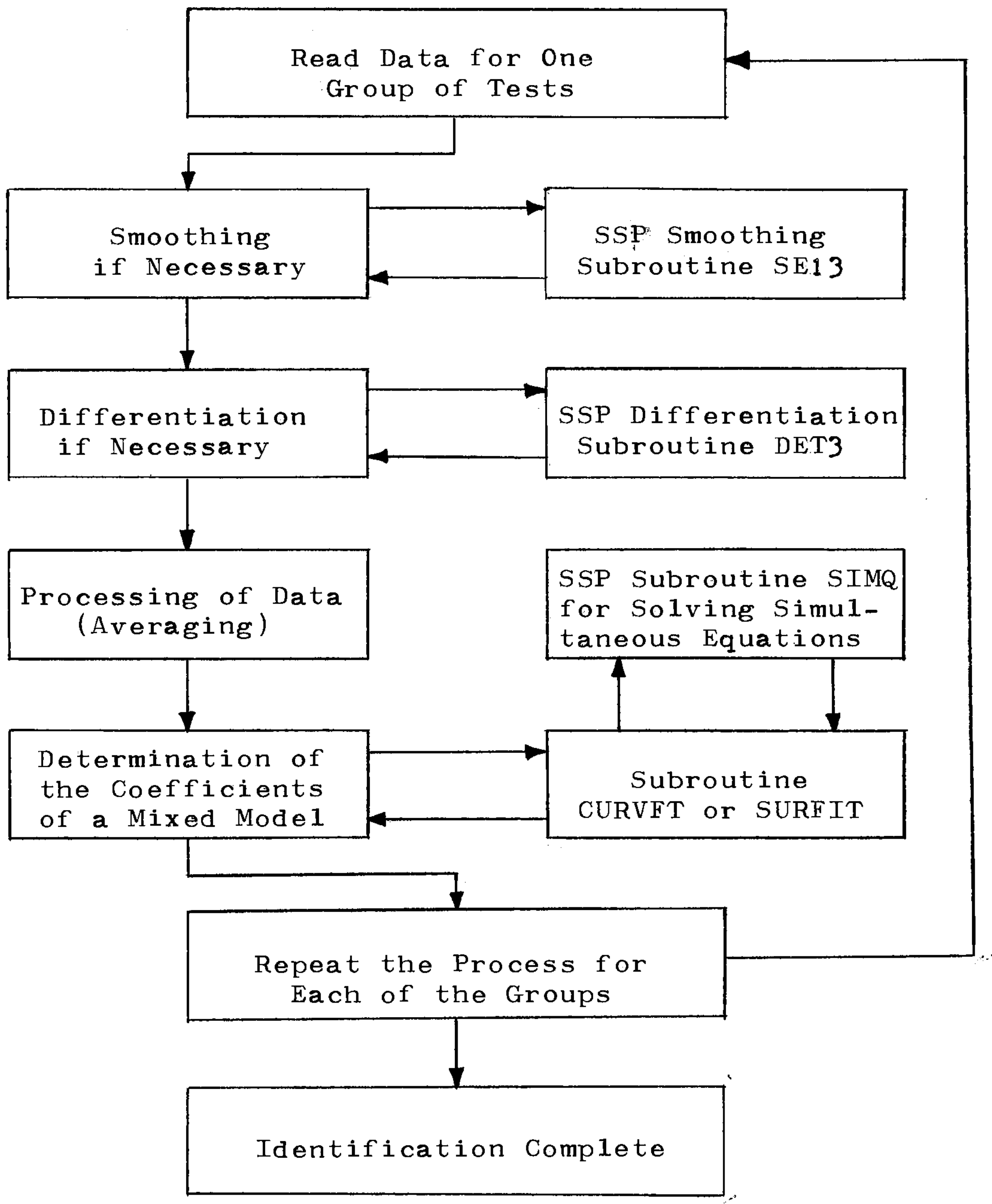


Figure 6. Identification Procedure Adopted in the Computer Subroutines SYSID1 and SYSID2

are divided respectively.

3. Determine the set of initial conditions. For first order systems, the two extreme initial conditions on the system are sufficient. For second-order systems obtain the locus of initial conditions and choose I pairs of initial conditions on the locus.
4. Conduct $2m_1m_2$ tests for a first-order system and Im_1 tests for a second-order system. For a single-input system $m_2 = 1$. A system with no inputs can be considered as a single-input system with $m_1 = 1$ and $u_1 = 0$. Measure the system states and the first derivatives of the states for all tests. If the derivatives are not measurable, they must be obtained by differentiation.
5. Sample all the measured data and store in punched card form (FORMAT 3X, 7E11.4). A variable sampling interval may be used depending upon the frequency content of the measured data. However, when $\dot{X}$ is not measurable, a constant interval is necessary for smoothing and differentiation.
6. Specify the degrees of the polynomials p_{x1} and p_{x2}.
7. Use SYSID1 for first-order systems and SYSID2 for second-order systems (see Appendix A). These subroutines give the mixed form of the model (polynomial in the state(s) and tabular in the input(s)).
8. Use CONVRT (see Appendix B) if a polynomial or a tabular form of the model is desired.

Based on the experience with a number of examples, the following numerical values are normally adequate for the variables used in the above steps: $m_1=m_2=n_1=n_2=11$; $p_{x1}=p_{x2}=p_{u1}=p_{u2}=3$ or 4; I=20; and 100 samples should be used for each test. Subroutines XDOT1 (for first-order systems) and XDOT12 (for second-order systems) which are presented in Appendix A may be used to evaluate the derivatives of the states for predicting the system response. When a polynomial form or a tabular form is used the corresponding subroutines (XDOT1 or XDOT12) which are presented in Appendix B must be used.

Model Simulation

This section describes the use of the identified model in the prediction of system responses for arbitrary inputs and arbitrary initial conditions on the system. If the system states at time t_0, and the inputs $U(t)$; $t_0 \leq t \leq t_f$, where t_f is the final time, are known; the system response $X(t)$ can be obtained by numerically integrating the model differential equations from t_0 to t_f. A Runge-Kutta integration program may be used. The integration program requires the derivatives of the states for known values of the states and the inputs. These derivatives can be evaluated by using the subroutines XDOT1 or XDOT12 (see Appendices A and B).

The model responses simulated as above will not be identical to the actual system responses because of the following two sources of error: 1) Insufficient accuracy of the identified model, and 2) The difference between the model states and the actual system states at the initial time. The numerical integration program can be assumed to be sufficiently accurate by properly selecting the integration scheme and the integration step size. For the class of nonlinear systems considered in this thesis, the error between the model response and the actual system response is found to converge to an allowable amount within the capability of the identified model.

The above result can be used to predict the response of a real process for any arbitrary input without the knowledge of the initial state of the process. The model can be simulated with zero initial conditions which introduce an initial error. The predicted system response will be meaningful only after one or two settling times when the error converges to an allowable amount.

Model Analysis

Another application of the model is to describe the system in the small about an operating point (usually a steady-state operating point). This is done by linearizing the model differential equations about the operating point. The steady-state response can also be found by setting $\dot{X}=0$. That is,

$$F(X_{ss}, U_{step}, C) = 0. \qquad (3\text{-}1)$$

In the above equation U_{step} is a vector of step inputs and

X_{ss} is a vector of steady-state responses of the system. The responses X_{ss} can be found analytically, without actually integrating the model equations, by solving a set of nonlinear algebraic equations when the model form is polynomial and by inverse interpolation when the model form is tabular. The set of linearized differential equations valid in the vicinity of an operating point (X_{op}, U_{op}) can be obtained as follows:

$$\delta \dot{X} = A\ \delta X + B\ \delta U \tag{3-2}$$
$$A = \partial F / \partial X$$
$$B = \partial F / \partial U$$

where, the coefficient matrices A and B are obtained by evaluating the partial derivatives at the operating point. These evaluations are performed analytically when the model form is polynomial and numerically when the model form is tabular.

The linearized equations can be used to investigate the stability of the model in the small about any operating point of interest. Also, these equations can be used to continuously find the optimal control for a closed-loop process. The results of optimal control theory, which are applicable to linear systems, can be used to generate the optimum control for a nonlinear system in a small neighborhood around an operating point in the X-U hyperspace. The coefficient matrices A and B can be evaluated for each new operating point of the system.

Examples

A number of examples were worked to validate the identification technique. Six examples are presented in this section to illustrate the technique. In the first four examples the unknown system was simulated by numerically integrating known mathematical models. In the fourth example zero-mean Gaussian noise was superimposed on the inputs and the measured responses. Both the state and its derivative were assumed available. In Examples 1, 2 and 3 the derivatives of the states were obtained by numerical differentiation. Examples 5 and 6 were actual physical systems. In these examples the measured responses were smoothed before differentiation.

The results of identification were verified by comparing the responses of the system and of the identified model to the same but arbitrary input(s). The arbitrary inputs were sequences of pulses whose amplitude and width were independent random variables. Mean squared error (MSE), as defined below, was considered as a measure of closeness.

$$MSE = (1/t_f) \int_0^{t_f} (x_s - x_m)^2 \, dt$$

Convergence of the model response was verified by starting the model and the system from different initial conditions.

The computational times for identification (includes smoothing and differentiation where applicable) and for simulation of the system and the model for 500 Runge-Kutta integration steps are summarized for each example. An IBM

360 model 65 digital computer was used. The repeatability of the CPU time on this machine was within ± 0.5 seconds.

To illustrate the choice of the three model forms, tabular, mixed and polynomial forms were used in the examples. For convenience, special programs were developed for each of the examples to generate the required test data and to identify the model. These programs are not included in this thesis. However, the necessary computer subroutines for identifying a mixed form of the model (polynomial in X and tabular in terms of U) are presented in Appendix A. If a model form which is polynomial both in X and in U or tabular in terms of both X and U is desired, a conversion subroutine is presented in Appendix B. In addition, the subroutines STEADY and LINRIZ used in the examples for model analysis are presented in the Appendix C.

Example 1

A single-input, first-order system was simulated by Equation (3-3). The system was modeled by Equation (3-4).

$$\dot{x}_{1s} = (\mathrm{ABS}(u_1 - x_{1s}))^{1.8}\ \mathrm{SIGN}(u_1 - x_{1s}) \qquad (3\text{-}3)$$

$$\dot{x}_{1m} = f_1(x_{1m}, u_1, c_1) \qquad (3\text{-}4)$$

In the above equations x_{1s} and x_{1m} represent the system state and the model state. A tabular model form was used. The range in the input (u_{1min}=-1 and u_{1max}=1) was divided into 11 levels (m_1=11). The range of the state (x_{1min}=-1 and x_{1max}=1) was divided into 11 levels (n_1=11). To generate the numbers in the table, $\dot{x}_1$ was assumed to be a fourth degree polynomial in x_1 ($p_{x1}=4$) for each of the input levels. The third row of the table (when the input level $u_{13}=-0.6$) has the following numbers: 0.227E 01 0.185E 01 0.141 E 01 0.101E 01 0.667E 00 0.395E 00 0.199E 00 0.709E-01 -0.836E-02 -0.694E-01 -0.154E 00.

The identification time was 1.12 seconds, the times required for the simulation of the system and the model were respectively 3.89 and 4.35 seconds. Figure 7 shows the arbitrary input and the responses of the system and the model when the initial conditions were 44.5% off. For a step input of 1.0, the steady-state value x_{1ss} was found to be 0.973. The coefficient matrices of the linearized differential equation were found to be A=[-0.247E 01] and B=[0.286E01] for operation in the vicinity of $x_{1op}=1.0$ and $u_{1op}=-1.0$.

<u>Example 2</u>

A dual-input, first-order system was simulated by Equation (3-5). This system was modeled by Equation (3-6). The

$$\dot{x}_{1s} = (ABS(e))^{1.7}\ SIGN(e) \qquad (3\text{-}5)$$

$$e = u_1 - SIN(11u_2/7) - 2x_{1s}$$

$$\dot{x}_{1m} = f_1(x_{1m}, u_1, u_2, c_1) \qquad (3\text{-}6)$$

ranges in the inputs ($u_{1min}=u_{2min}=-1$ and $u_{1max}=u_{2max}=1$) were divided into 11 levels each ($m_1=11$ and $m_2=11$). The range in the state ($x_{1min}=-1$ and $x_{1max}=1$) was divided into 11 levels ($n_1=11$). A tabular form of the model was used. p_{x1} was 4.

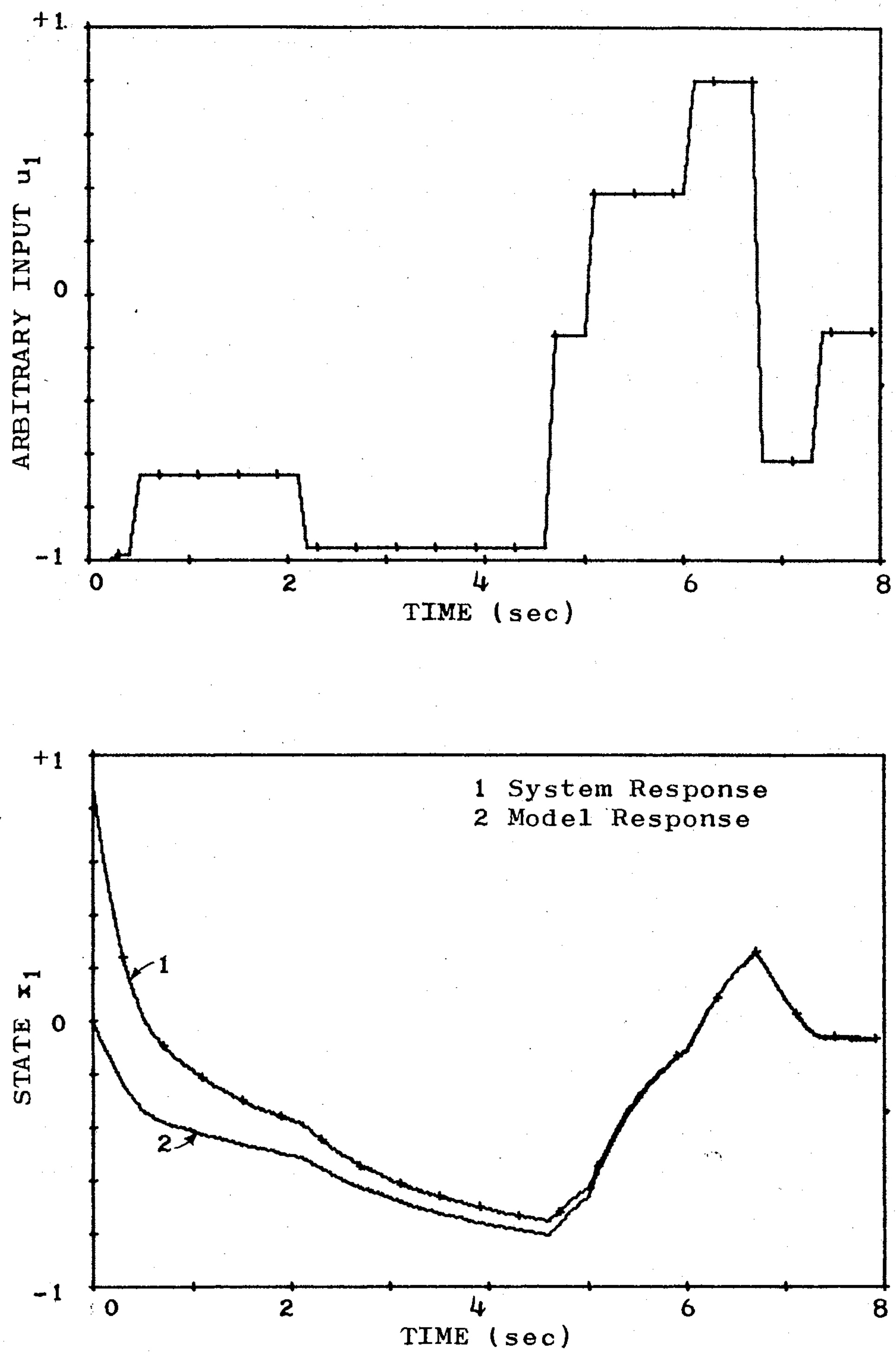


Figure 7. Arbitrary Input (u_1) and the Response (x_1) of the System and the Model (Example 1)

The identified model coefficients are represented by a three dimensional array $c_1=c_1(11,11,11)$. The 11 numbers of this array for a pair of input levels ($u_{14}=-0.4$ and $u_{22}=-0.8$) are the following: 0.481E01 0.371E01 0.262E01 0.167E01 0.928E00 0.408E00 0.722E-01 -0.172E00 -0.472E00 -0.103 E01 -0.209E01.

The identification time was 12.59 seconds and the times required for the simulation of the system and the model were 7.85 and 8.27 seconds respectively. Figure 8 shows the arbitrary input and the responses of the system and the model when the initial conditions were 100% off. For a pair of step inputs, $u_{1step}=1.0$ and $u_{2step}=0.8$, the steady-state value x_{1ss} was analytically found to be 0.240E-01. The coefficient matrices of the linearized differential equation were found to be A=[0.568E00] and B= [0.111E00 0.765E-01] for operation in the vicinity of an operating point ($x_{1op}=1.00$, $u_{1op}=-1.00$ and $u_{2op}=-0.63$).

Example 3

A single-input, second-order system was simulated by Equations (3-7) and (3-8). Bose (6) showed that under certain circumstances these equations represent an hydraulic spool type valve. This system was modeled by Equations (3-9) and (3-10) for identification . The range of the input ($u_{1min}=0$ and $u_{1max}=1$) was divided into 11 levels ($m_1=11$) and the ranges of the states ($x_{1min}=-0.266$, $x_{1max}=1.140$, $x_{2min}=-0.493$ and $x_{2max}=0.906$) were divided into 11 levels

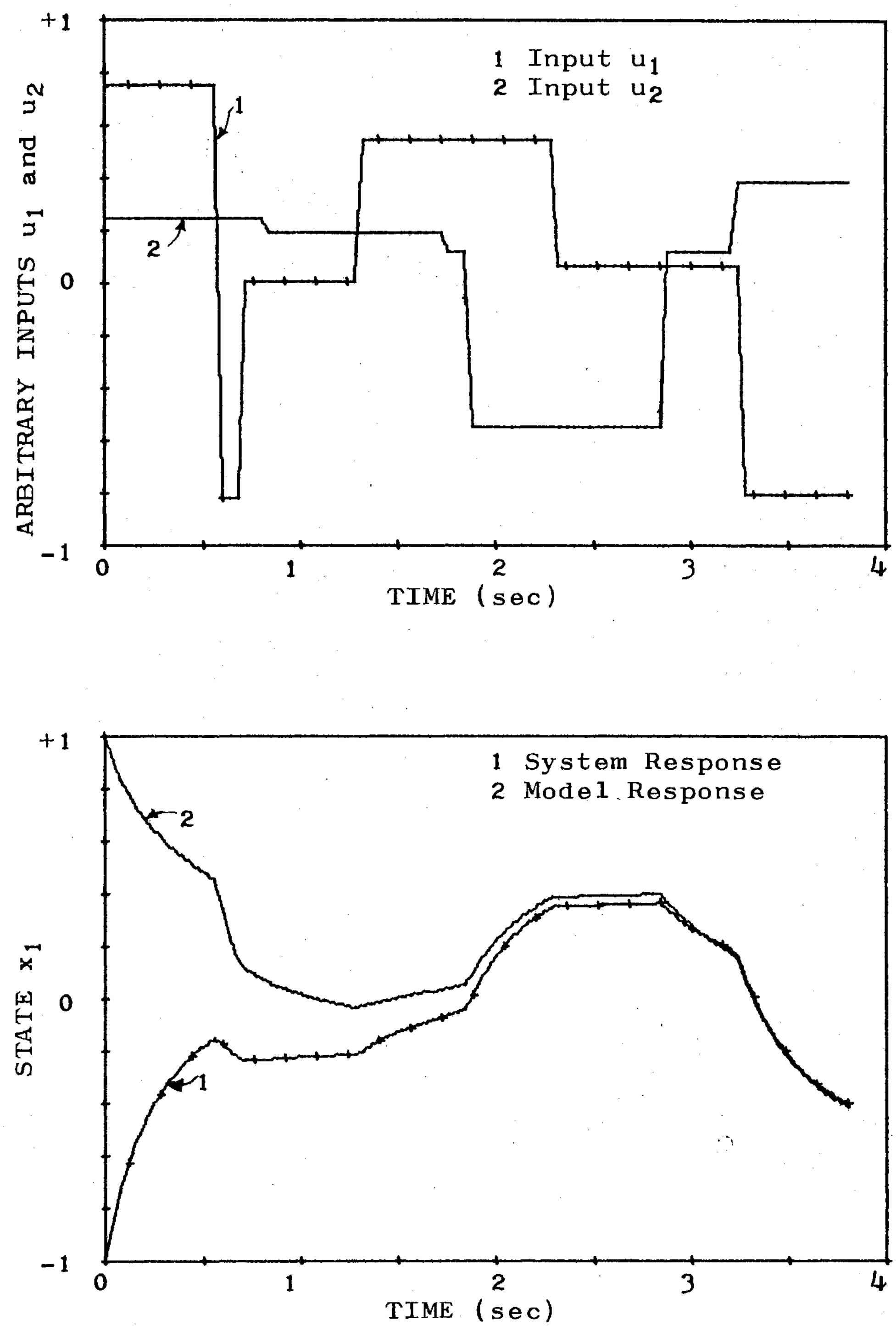


Figure 8. Arbitrary Inputs (u_1 and u_2) and the Response (x_1) of the System and the Model (Example 2)

$$\dot{x}_{1s} = x_{2s} \tag{3-7}$$

$$\dot{x}_{2s} = u_1 - 0.36x_{2s} - 0.24x_{1s} - 0.86122(x_{1s})^3 - 1.3174x_{1s}x_{2s} \tag{3-8}$$

$$\dot{x}_{1m} = f_1(x_{1m}, x_{2m}, u_1, c_1) \tag{3-9}$$

$$\dot{x}_{2m} = f_2(x_{1m}, x_{2m}, u_1, c_2) \tag{3-10}$$

each (n_1=11 and n_2=11). For each group of tests 20 pairs of initial conditions were chosen. Two responses of the system to step inputs of $u_1 = u_{1min}$ and $u_1 = u_{1max}$, with zero initial conditions on the system, gave the minimum and maximum limits on the first state. These two limits for the 2nd state were found from the following two tests: 1) The input was a step of $u_1 = u_{1min}$, and the initial conditions were $x_1(0) = x_{1max}$ and $x_2(0) = 0$; and 2) The input was a step of $u_1 = u_{1max}$, and the initial conditions were $x_1(0) = x_{1min}$ and $x_2(0) = 0$. These latter two tests established the locus of initial conditions. A mixed form of the model, polynomial in x_1 and x_2 and tabular in u_1, was used with $p_{x1} = p_{x2} = 3$. The coefficients (4 x 4 matrices for one input level) of the polynomials in x_1 and x_2 are given below for a specific input level of 0.9 (tenth level).

$$c_1 = \begin{bmatrix} -0.944E-04 & 0.135E-04 & -0.518E-03 & 0.702E-03 \\ 0.100E\ 01 & -.552E-02 & 0.130E-01 & -0.723E-02 \\ -0.513E-05 & 0.138E-01 & -0.341E-01 & 0.196E-01 \\ -0.157E-03 & -.865E-02 & 0.219E-01 & -0.129E-01 \end{bmatrix}$$

$$c_2 = \begin{bmatrix} 0.900E\ 00 & -.246E\ 00 & 0.270E-01 & -0.887E\ 00 \\ -0.363E\ 00 & -.128E\ 01 & -0.172E\ 00 & 0.128E\ 00 \\ 0.112E-01 & -.161E\ 00 & 0.447E\ 00 & -0.251E\ 00 \\ -0.752E-02 & 0.122E\ 00 & -0.298E\ 00 & 0.146E\ 00 \end{bmatrix}$$

The identification time was 14.47 seconds. The times required for the simulation of the system and the model for 500 Runge-Kutta integration steps were 6.27 and 9.82 seconds respectively. Figures 9 and 10 show the arbitrary input and the responses of the system and the model when the initial conditions were the same. Note that the correspondence was so close that it is difficult to distinguish between the two responses. For a step input of u_{1step}=0.43, the steady-state response was analytically found to be x_{1ss}=0.676E 00 and x_{2ss}=-0.949E-05. The coefficient matrices of the linearized differential equations were found to be,

$$A = \begin{bmatrix} -0.184E-03 & 0.100E\ 01 \\ -0.338E\ 00 & -0.864E\ 00 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -0.178E-02 \\ 0.100E\ 01 \end{bmatrix}$$

for operation in the small about the point (x_{1op}=0.380, x_{2op}=-0.210 and u_{1op}=0.447).

Example 4

A single-input, first-order system was simulated by the following equations:

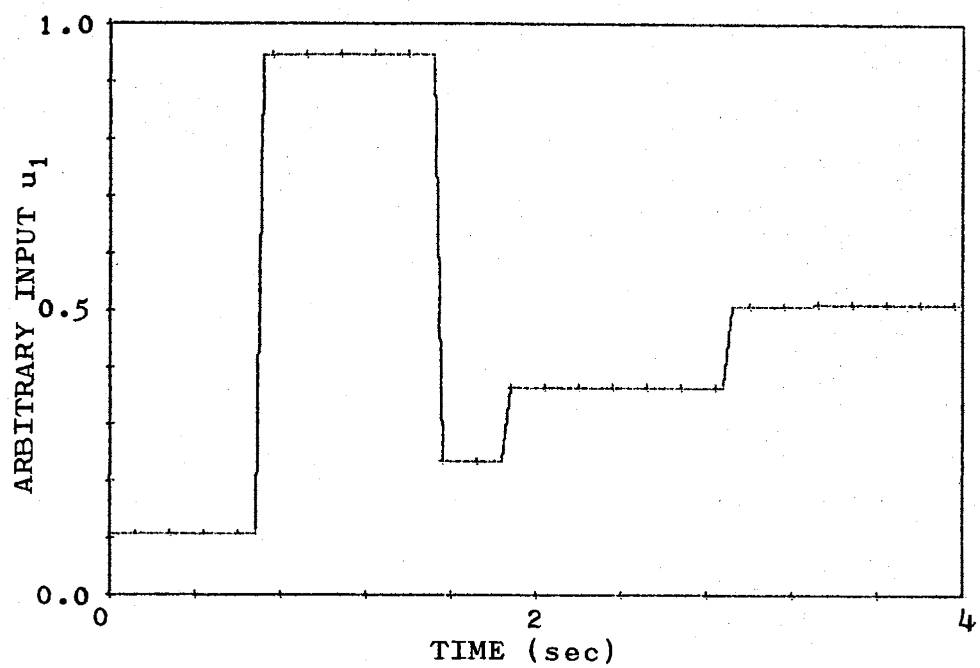
$$\dot{x}_{1s} = (\text{ABS}(u_1' - x_{1s}))^{1.8}\ \text{SIGN}(u_1' - x_{1s}) \qquad (3\text{-}11)$$

$$u_1' = u_1 + R_1$$

$$x_{1s}' = x_{1s} + R_2$$

$$\dot{x}_{1s}' = \dot{x}_{1s} + R_3$$

where, R_1, R_2 and R_3 were three independent Gaussian random variables with mean zero and standard deviations d_1, d_2 and



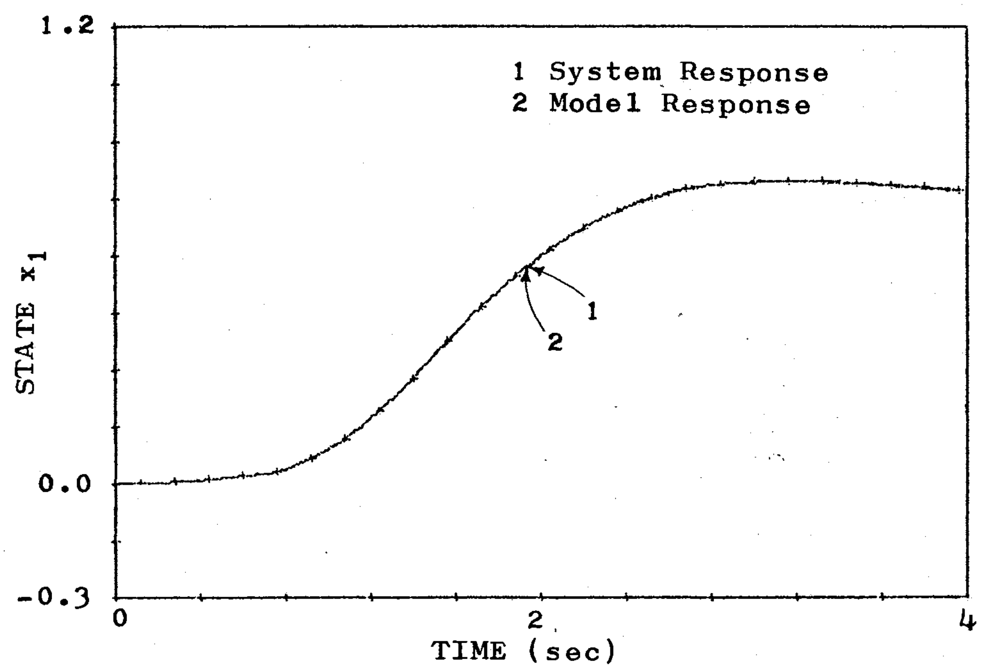


Figure 9. Arbitrary Input (u_1) and the Response (x_1) of the System and the Model (Example 3)

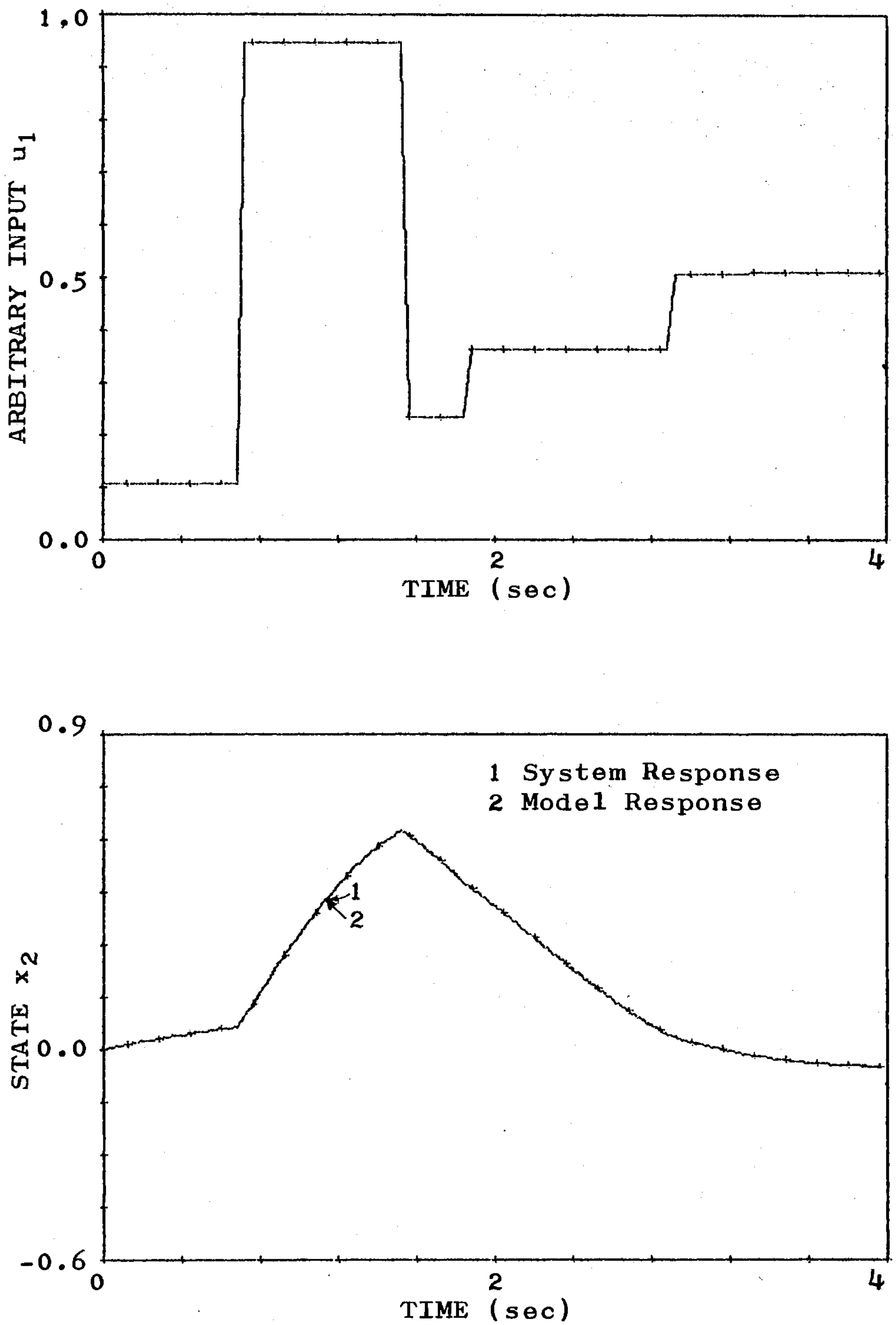


Figure 10. Arbitrary Input (u_1) and the Response (x_2) of the System and the Model (Example 3)

d_3 as follows:

$$d_1 = 1/3 \text{ NSR } (u_{1max} - u_{1min})$$

$$d_2 = 1/3 \text{ NSR } (x_{1max} - x_{1min})$$

$$d_3 = 1/3 \text{ NSR } (\dot{x}_{1max} - \dot{x}_{1min})$$

In the above relations NSR is defined as the noise to signal ratio ($0 \leq \text{NSR} \leq 1$). Note that the input seen by the system was u_1' and the system responses were x_{1s} and $\dot{x}_{1s}$. But only u_1, x_{1s}' and $\dot{x}_{1s}'$ were used for the identification purposes. Figure 11 shows the system responses to a typical test input with and without the measurement noise.

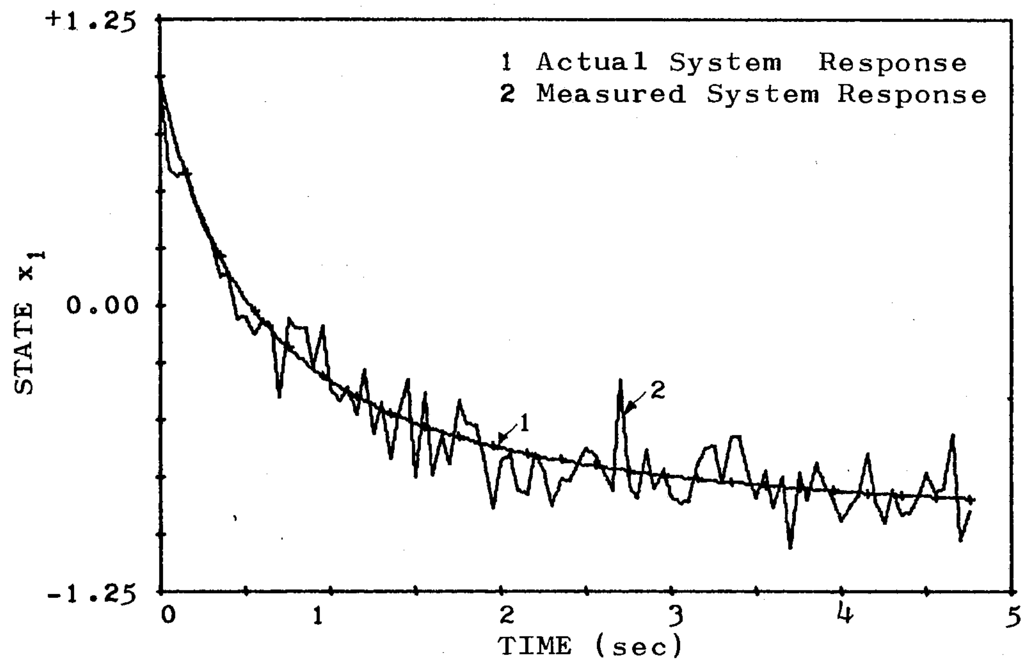
The system was modeled as,

$$\dot{x}_{1m} = f_1(x_{1m}, u_1, c_1). \qquad (3\text{-}12)$$

The following data was used to simulate the system responses and to identify a polynomial form of the model: $u_{1min} = -1$, $u_{1max} = 1$, $x_{1min} = -1$ and $x_{1max} = 1$; $m_1 = 9$ and $n_1 = 9$; and $p_{x1} = 3$ and $p_{u1} = 3$. The 4 x 4 matrix of model coefficients was,

$$c_1 = \begin{bmatrix} 0.242\text{E-}01 & -0.489\text{E } 00 & -0.255\text{E-}01 & -0.456\text{E } 00 \\ 0.458\text{E } 00 & -0.175\text{E } 00 & 0.158\text{E } 01 & 0.162\text{E } 00 \\ -0.660\text{E-}01 & -0.149\text{E } 01 & 0.592\text{E-}01 & 0.801\text{E } 00 \\ 0.550\text{E } 00 & 0.327\text{E } 00 & -0.973\text{E } 00 & -0.272\text{E } 00 \end{bmatrix}$$

Figure 12 shows the arbitrary input and the responses of the system and the model to this input when NSR=0.4. The mean squared error (MSE) was 2.05E-04. The above identification problem was repeated for nine different values of NSR from 0 to 0.4. A plot of the mean squared error versus NSR is shown in Figure 13.



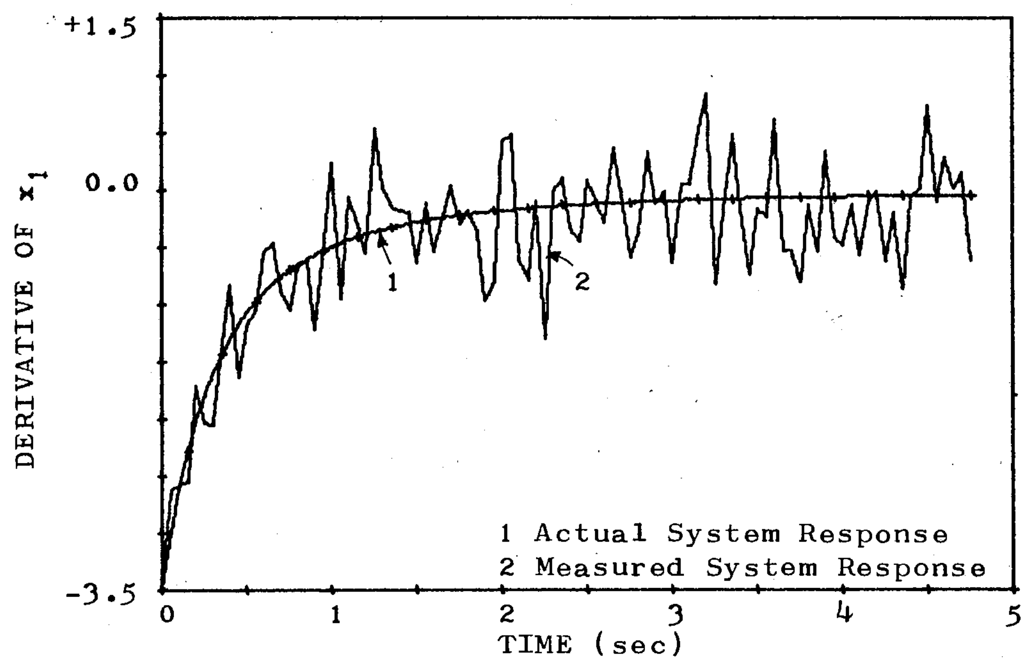
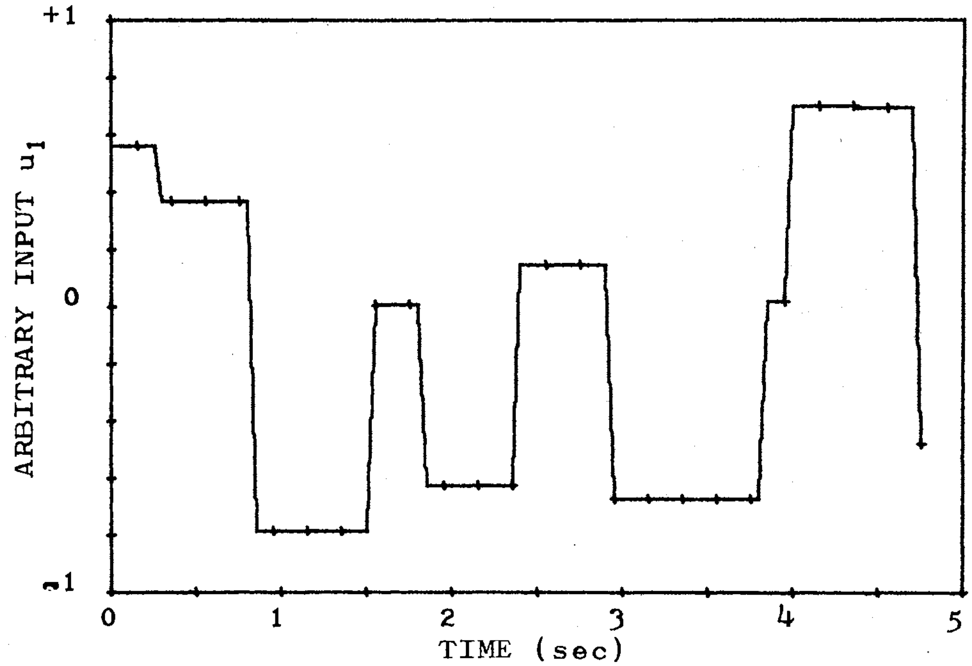


Figure 11. Actual and Measured Responses (x_1 and $\dot{x}_1$) of the System (Example 4) When NSR = 0.4



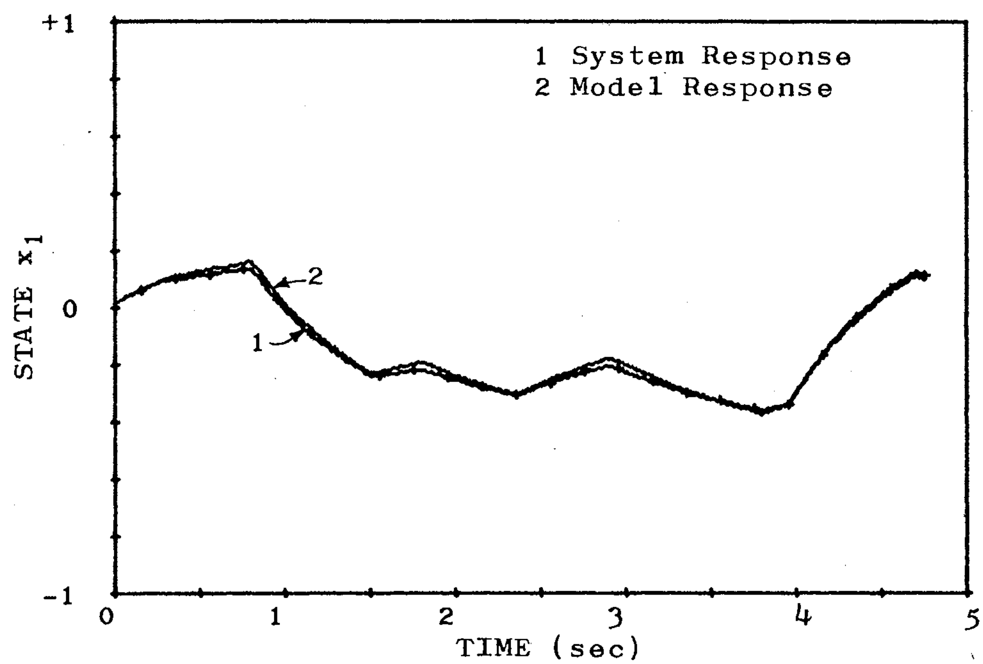


Figure 12. Arbitrary Input (u_1) and the Response (x_1) of the System and the Model (Example 4) When NSR = 0.4

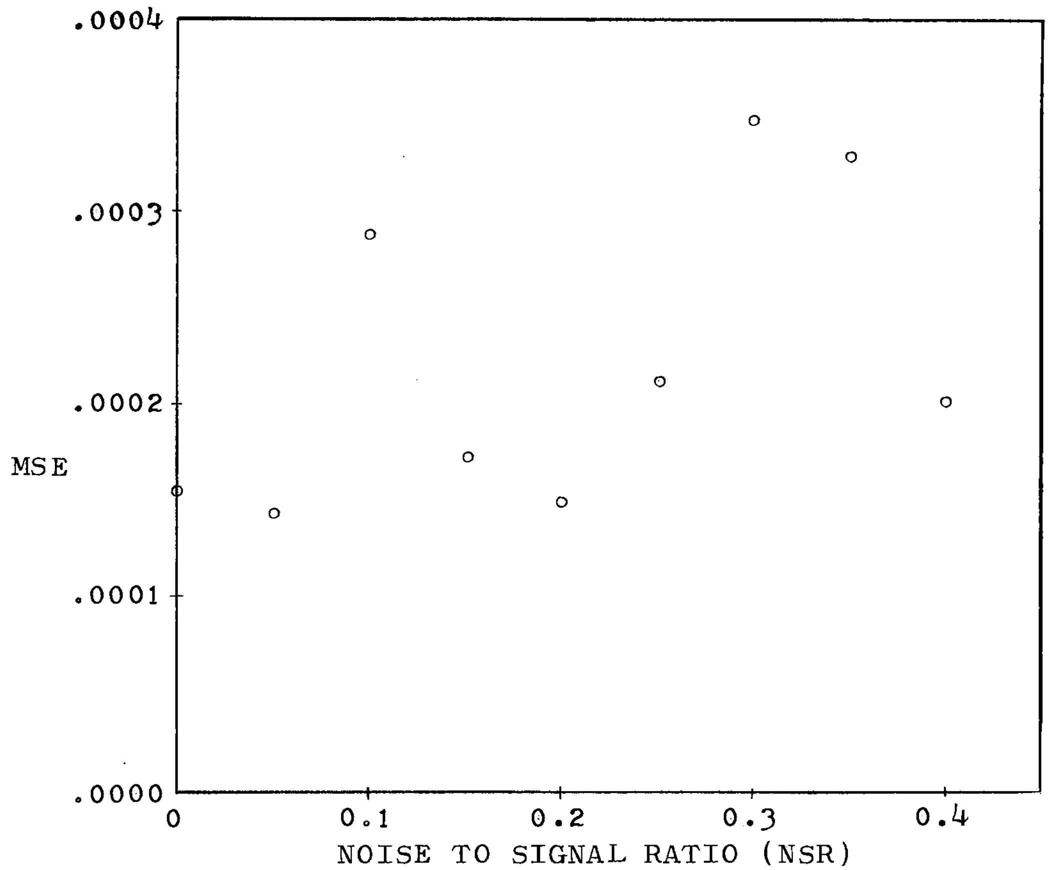


Figure 13. A Plot of Mean Squared Error (MSE) Versus Noise to Signal Ratio (NSR)

Example 5

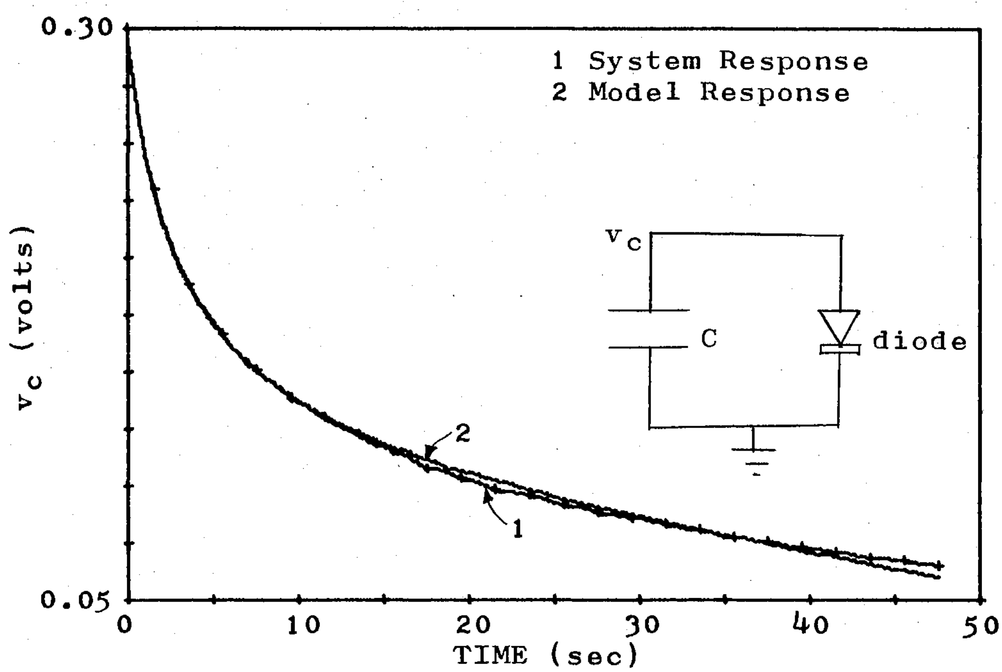
A real physical system, which consisted of an electrical capacitor discharging through a diode (see inserts in Figure 14), was considered. The capacitor voltage, v_c, was recorded as a function of time for two initial conditions (v_1=0.3 volts and v_2=0.25 volts). The data with the first initial condition was used for identification. The data with the second initial condition was used to verify the accuracy of identification. The system was modeled by the following equation,

$$\dot{v}_c = c_1(0)+c_1(1)v_c+c_1(2)v_c^2+c_1(3)v_c^3+c_1(4)v_c^4 \qquad (3\text{-}13)$$

The total range in v_c was divided into 21 levels. The five coefficients were found to be: $c_1(0)$=0.951E-04, $c_1(1)$= -0.759E-01, $c_1(2)$=0.134E 01, $c_1(3)$=0.839E01 and $c_1(4)$= 0.738E 01. Figure 14 shows the actual experimental responses and the identified responses for the two initial conditions. The identification time was 1.7 seconds.

Example 6

A real physical system, which consisted of a pressurized pneumatic tank discharging into atmosphere through an orifice with nonlinear resistance, was considered. The set up is shown in the inserts of Figure 15. The tank pressure, p_t, was recorded as a function of time for two initial conditions on the system (p_1=25 psig and p_2=15 psig). The data with one initial pressure was used for identification. The



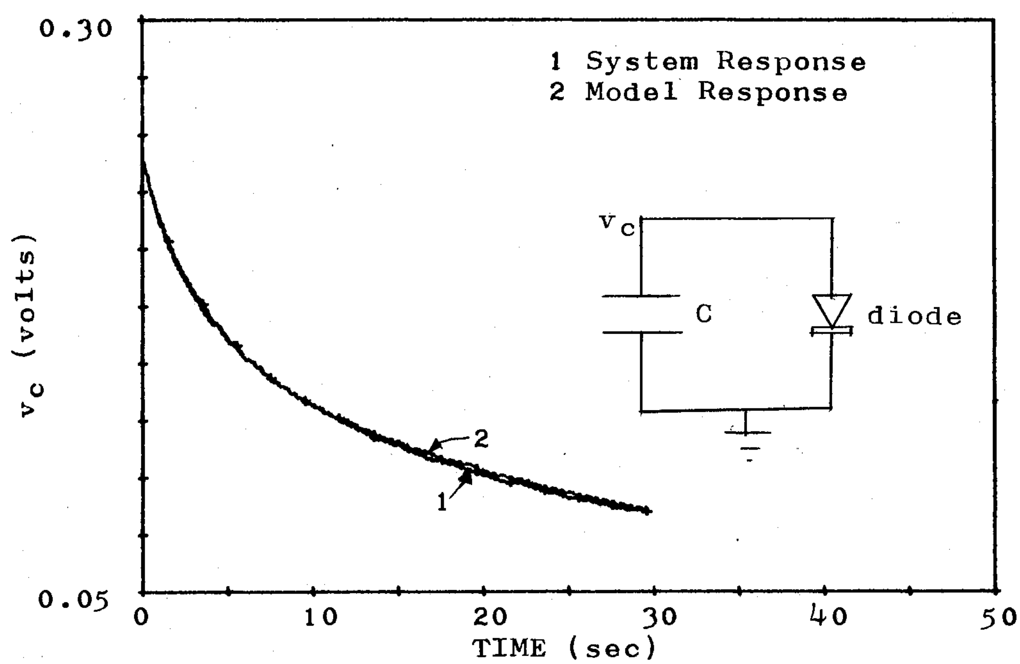
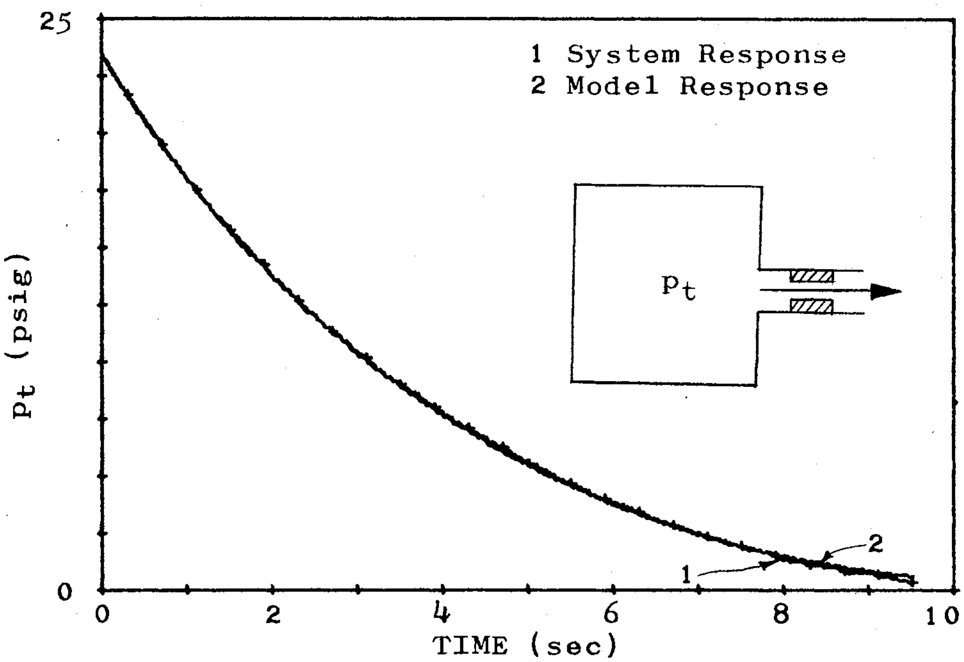


Figure 14. Responses of the System and the Model for the Two Initial Capacitor Voltages (Example 5)



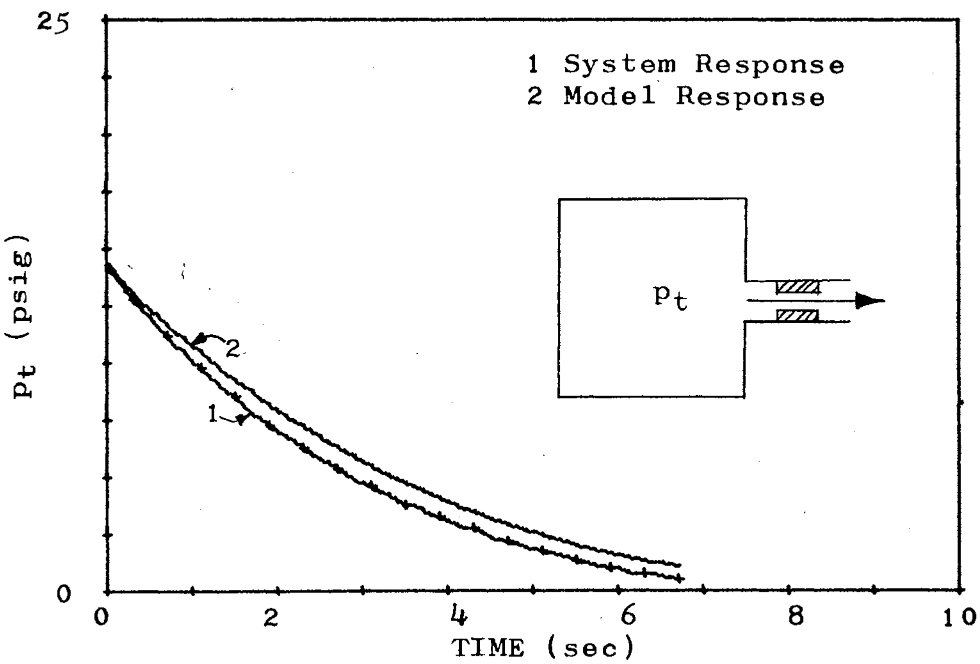


Figure 15. Responses of the System and the Model for the Two Initial Tank Pressures (Example 6).

data with the other initial pressure was used to verify the accuracy of identification. The system was modeled as,

$$\dot{p}_t = c_1(0)+c_1(1)p_t+c_1(2)p_t^2+c_1(3)p_t^3+c_1(4)p_t^4 \qquad (3\text{-}14)$$

The total range in p_t was divided into 21 levels. The coefficients of the above relation were found to be: $c_1(0)=$ -0.202E 00, $c_1(1)=$-0.418E 00, $c_1(2)=$0.235E-01, $c_1(3)=$-0.102 E-02 and $c_1(4)=$0.127E-04. The actual system responses and the identified model responses for the two initial pressures are shown in Figure 15. The mean squared error was 0.0127. The accuracy was within 4%.

Greater accuracy is not possible with a first-order system model. Intuition leads one to the conclusion that the system could be modeled more accurately by a second-order system which accounts implicitly for the heat transfer effects in the process.

CHAPTER IV

COMPARISON WITH OTHER TECHNIQUES

In comparing the modified differential approximation technique with other known techniques the following factors are considered:

1. *A priori* knowledge about the system model (form)
2. Computational requirements (storage and time)
3. Data required for identification
4. Method of determining the model coefficients
5. Applications of the identified model
6. Limitations of the technique.

It is difficult to make a meaningful quantitative comparison of various available identification techniques. There are more than several hundred techniques, each of which has its own advantages and disadvantages. Some are general purpose and some are special purpose techniques. However, the computational requirements depend not only on the identification technique, but also on the programming skill. A qualitative comparison of the identification techniques which are applicable to nonlinear systems is given in Figure 16. A cross mark (X) is placed if the technique has an undesirable factor.

	Functional Power Series Techniques	Pattern Recognition Techniques	Other State-Space Techniques	Modified Differential Approximation Technique
A priori knowledge	Model form not required	Model form not required	Model form required	Model form not required
Data required	Normal operating system responses	Normal operating system responses	Normal operating system responses	System responses for specific tests
Computer storage	Large	Large	Reasonable	Small
Computer time	Large	Large	Reasonable	Small
Accuracy	Poor	Poor	Reasonable	Reasonable
Applications	Simulation only	Simulation only	Simulation and analysis	Simulation and analysis
Determining the coefficients of the model	Noniterative technique	Noniterative technique	Iterative and requires integration of differential equations	Noniterative and requires no integrations
Utility of the identified model	Used for any input within the test data	Used for any input within the test data	Not used for any arbitrary input and initial conditions in the X-U hyperspace	Used for any arbitrary input and initial conditions in the X-U hyperspace of interest
Limitations	Limited to single-input systems	Limited to single input systems	Applicable to multiple-input systems	Applicable to multiple-input systems

Figure 16. A Qualitative Comparison of the Identification Techniques Which are Applicable to Nonlinear Systems
(A cross mark, X, has been placed over each of the undesirable factors.)

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The modified differential approximation technique is applicable to stationary nonlinear systems which can be described by lumped parameter models. The following are the principal features of the technique:

1. The technique does not require a priori knowledge about the form of the system mathematical model
2. It is found that a wide class of nonlinear systems can be adequately described by polynomial, tabular or mixed form of the model
3. Specified inputs allow decomposition of the MSSE which results in reduced computational effort
4. The model coefficients are determined uniquely without iterations
5. The identified model can be used to compute the re-responses to any arbitrary input(s)
6. The technique is insensitive to zero-mean noise in the test inputs and in the measured responses
7. The model allows determination of linearized differential equations valid in the vicinity of an operating point

8. Three different forms of the model can be found. The mixed form is the most efficient for identification. The tabular form is the most efficient for predicting the system response. The polynomial form gives the most accurate results with the minimum number of model coefficients

The primary drawback of the technique is that a specific set of tests must be conducted on the system. It can be concluded that for the class of systems considered the modified differential approximation technique is superior to other known techniques in identification time, accuracy and storage requirement.

Recommendations

Future work could be performed in the following areas to improve and extend the identification technique:

1. Simplification of the identified model equations
2. The use of normal operating input-output records of the system for identification purposes
3. The use of *a priori* knowledge, where available, to reduce the identification time
4. The use of orthogonal functions instead of polynomials to reduce numerical round-off errors
5. Investigation of on-line applications
6. Consideration of time delays and hysteresis in the systems to be identified.

SELECTED BIBLIOGRAPHY

(1) Allison, J. S. "On the Comparison of Two Methods of Off-Line Parameter Identification." Journal of Mathematical Analysis and Applications, (1967), pp. 229-237.

(2) Arozullah, M. "Characterization and Optimization of Nonlinear Systems." JACC 1969, pp. 81-90.

(3) Bacon, C. M. "Linear State-Space Model From Solution Specifications." Proceedings, Eighth Midwest Symposium on Circuit Theory, Colorado State University(June 1965).

(4) Bacon, C. M. "Parameter Optimization in Linear Systems - A State-Space Approach." Proceedings, Third Annual Allerton Conference on Circuit and System Theory, Monticello, Illinois(Oct. 1965).

(5) Bellman, R. E., Kalaba, R., and Sridhar, R. "Adaptive Control via Quasilinearization and Differential Approximation." IEEE Transactions on Automatic Control, Vol. AC-10, No. 2 (1965), p. 199.

(6) Bose, J. E. "Design of Nonlinear Second-Order Dynamic Systems to Meet Arbitrary Performance Criteria." (Ph. D. Thesis, Oklahoma State University, 1966).

(7) Cox, H. "Estimation of State Variables via Dynamic Programing." JACC 1964, pp. 376-381.

(8) Cox, H. "On the Estimation of State Variables and Parameters for Noisy Dynamic Systems." IEEE Transactions, January 1964.

(9) Detcmendy, D. M., and Sridhar, R. "Sequential Estimation of States and Parameters in Noisy Nonlinear Dynamical Systems." Transactions of the ASME, June 1966, pp. 362-368.

(10) Diskind, T., and Esterson, G. L. "Parameter Evaluation by Complex Domain Analysis With Application to Chemical Kinetics." JACC 1969, pp. 770-781.

(11) Flake, R. H. "Volterra Series Representation of Nonlinear Systems." Transactions of the ASME, Vol. 82(Applications and Industry Part II), January 1963, pp. 178-189.

(12) Frank, P., and McGhee, R. "Determining Input-Output Relations of Nonlinear Systems by Inversions." Transactions, IEEE, Vol. CT-10, June 1963.

(13) Giese, C., and McGhee, R. "Estimation of Nonlinear System States and Parameters by Regression Methods." JACC 1965, pp. 46-55.

(14) Gorman, D., and Zaborszky, J. "Functional Representation of Nonlinear Systems, Interpolation and Lagrange Expansion for Functionals." JACC 1965.

(15) Gunda, R. "Identification of Nonlinear Systems." (A Research Proposal, Oklahoma State University).

(16) Kirvatis, K., and Fu, K. S. "Identification of Nonlinear Systems by Stochastic Approximation." JACC 1966, p. 255.

(17) Kohr, R. H. "On the Identification of Linear and Nonlinear Systems." Simulation, March 1967.

(18) Kumar, K. S. P., and Sridhar, R. "On the Identification of Linear Systems." JACC 1964, pp. 361-65.

(19) Kwatny, H. G., and Schen, D. W. C. "Identification of Nonlinear Systems Using a Method of Stochastic Approximation." JACC 1967, pp. 814-820.

(20) Meissenger, H. F. "The Use of Parameter Influence Coefficients in Computer Analysis of Dynamical Systems." Proceedings, Western Joint Computer Conference, 1960, p. 181.

(21) Miller, R. W., and Roy, R. "Nonlinear Process Identification Using Statistical Pattern Matrices." JACC 1964, p. 349.

(22) Pazdera, J. S., and Pottinger, H. J. "Linear System Identification via Liapunov Design Technique." JACC 1969.

(23) Robert, D. R., and Kerr, B. "Statistical, Structural and Regression Methods in Linear Process Identification."

(24) Rose, R. E., and Lance, G. M. "An Approximate Design Method for Parameter Identification." JACC 1969.

(25) Rosenberg, A. E., and Schen, D. W. C. "Regression Analysis and Its Applications to the System Identification Problem." JACC 1963, pp. 446-51.

(26) Roy, R., and Schley, C. "Process Identification Using a Mode Learning Machine." JACC 1966, pp. 639-648.

(27) Sage, A. P., and Eisenberg, B. R. "Experiments in Nonlinear and Nonstationary System Identification via Quasilinearization and Differential Approximation." JACC 1966, pp. 522-530.

(28) Sunahara, Y. "An Approximate Method of State Estimation for Nonlinear Dynamical Systems." JACC 1969, p. 162.

(29) Sastry, V. A., Vetter, W. J., and Caston, R. "The Application of Identification Methods to an Industrial Process." JACC 1969, pp. 787-794.

(30) Smith, C. K. "Quasilinear and Steepest Descent Identification Methods With Hybrid Computers." (M. S. Thesis, Tennessee Technological University, August 1969).

(31) Unruh, D. R. "Determination of Optimal Parameters for Dynamical Systems." (Ph. D. Thesis, Oklahoma State University, May 1970).

(32) Wells, C. H. "Application of Modern Estimation and Identification Techniques to Chemical Process." JACC 1969.

(33) IBM Application Program. "System/360 Scientific Subroutine Package (360A-CM-03X)." Version III Programmer's Manual.

APPENDIX A

IDENTIFICATION SUBROUTINES

In this appendix the computer subroutines (SYSID1 and SYSID2) which can be used for the identification of first-order and second-order systems are presented. Two of the required external subroutines (included in this appendix) are: CURVFT and SURFIT for fitting curves and surfaces through arbitrary data points in a least squares sense. In addition to the above the following subroutines are required from the IBM Scientific Subroutine Package (SSP): SE13 for smoothing, DET3 for differentiation and SIMQ for solving a set of simultaneous linear algebraic equations.

The subroutines SYSID1 and SYSID2 yield a standard form of the model which is tabular in terms of the inputs and polynomial in the states. This form is explained below for a single-input, first-order system. Although $\dot{x}_1$ is a function of both x_1 and u_1, the following relation is obtained:

$$\dot{x}_1 = c_0+c_1x_1+c_2x_1^2+\ldots$$

where, the coefficients of the polynomial depend on the input level. If there are m_1 levels in the input, there will be m_1 sets of the coefficients for the above polynomial. These m_1 sets of coefficients are conveniently represented and stored in a multidimensional table (in this case it is a matrix). After the coefficients are identified, the value of $\dot{x}_1$ for any arbitrary values of x_1 and u_1 can be found by evaluating the polynomial at x_1 twice using two proper rows of the coefficient matrix and interpolating in u_1.

Subroutines XDOT1 and XDOT12 presented in this appendix may be used to evaluate the derivatives of the states for numerical integration purposes. All of the subroutines presented in this appendix contain the necessary explanation.

```
      SUBROUTINE SYSID1
C-----
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +        THIS PROGRAM IDENTIFIES A FIRST-ORDER SYSTEM WITH ONE OR +
C     +          TWO INPUTS.  THE IDENTIFIED MODEL IS A POLYNOMIAL IN X1+
C     +          AND A TABLE IN U1 AND U2.  THE COEFFICIENTS OF THE     +
C     +          POLYNOMIAL ARE PRINTED/PUNCHED FOR EACH STEP INPUT(OR  +
C     +          PAIR OF STEP INPUTS).                                  +
C     +        SUBROUTINE REQUIREMENT-SE13, DET3, SIMQ(SSP) AND CURVFT  +
C     +        THE FOLLOWING DATA IS REQUIRED FOR IDENTIFICATION        +
C     +                                                                 +
C     +        FIRST DATA CARD HAS FORMAT 6I10 AND MUST CONTAIN         +
C     +          NINPUT - NUMBER OF INPUTS(1 OR 2)                      +
C     +          NDEGX1 - DEGREE OF THE POLYNOMIAL IN X1                +
C     +          NGRDX1, NGRDU1 AND NGRDU2 - NUMBERS OF LEVELS INTO     +
C     +            WHICH X1, U1 AND U2 ARE DIVIDED RESPECTIVELY         +
C     +          IPUNCH - 1 IF PUNCHED OUTPUT IS DESIRED(0 OTHERWISE)   +
C     +          NOTE THAT NDEGX1 MUST BE LESS THAN NGRDX1.             +
C     +          USUALLY NDEGX1 = 3 OR 4, NGRDX1 = NGRDU1 = NGRDU2 = 11 +
C     +            ARE ADEQUATE.  NGRDU2 = 1 FOR SINGLE-INPUT SYSTEMS.  +
C     +        SECOND DATA CARD HAS FORMAT 6F10.3 AND MUST CONTAIN      +
C     +          X1MIN, X1MAX, U1MIN, U1MAX, U2MIN AND U2MAX - MINIMUM  +
C     +            AND MAXIMUM VALUES OF X1, U1 AND U2 RESPECTIVELY     +
C     +        THERE MUST BE NGRDU2 SETS OF TEST DATA AFTER THE FIRST   +
C     +          TWO DATA CARDS.  IN THE I-TH DATA SET THE VALUE OF THE +
C     +          STEP IN U2 = U2MIN + (U2MAX-U2MIN)*(I-1)/(NGRDU2-1).   +
C     +          EACH DATA SET CONTAINS NGRDU1 SUBSETS.  IN THE J-TH    +
C     +          SUBSET THE AMPLITUDE OF THE STEP IN U1 = U1MIN +       +
C     +          (U1MAX - U1MIN)*(J-1)/(NGRDU1-1).                      +
C     +        EACH SUBSET OF DATA MUST FOLLOW A DATA CARD WHICH HAS    +
C     +          FORMAT 4I10, F10.4 AND CONTAINS                        +
C     +          NIC - NUMBER OF INITIAL CONDITIONS.  USUALLY NIC=2, BUT+
C     +            FOR SOME STEP INPUTS ONE INITIAL CONDITION(NIC=1) MAY+
C     +            CAUSE THE SYSTEM TO RESPOND OVER THE TOTAL X1-RANGE. +
C     +          NDATA - NUMBER OF SAMPLES IN EACH RESPONSE(ABOUT 100)  +
C     +          IDIFF - 1 IF THE DERIVATIVE OF X1 HAS TO BE OBTAINED BY+
C     +            DIFFERENTIATING X1 AND 0 OTHERWISE                   +
C     +          ISMOTH - 1 IF SMOOTHING OF X1 IS REQUIRED(OTHERWISE 0).+
C     +          TDELTA - THE SAMPLING INTERVAL(CONSTANT)               +
C     +          NOTE THAT IF IDIFF=0, ISMOTH AND TDELTA ARE NOT NEEDED.+
C     +        THE SAMPLED RESPONSES IN EACH SUBSET MUST BE SUPPLIED    +
C     +          IN PUNCHED CARD FORM IN FORMAT 3X, 7E11.4 AS FOLLOWS:  +
C     +          THE VALUES OF THE STATE X1     | FOR THE FIRST         +
C     +          THE VALUES OF THE DERIVATIVE   | INITIAL CONDITION     +
C     +          OF X1(ONLY IF IDIFF = 0)       | ON THE SYSTEM         +
C     +          SAME AS ABOVE FOR THE SECOND INITIAL CONDITION         +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      DIMENSION X1S(100),X1DS(100),X1W(21),X1DW(21),N1W(21),C1M(10)
    1 FORMAT(1H1)
    2 FORMAT(1H )
    3 FORMAT(8F10.3)
    4 FORMAT(8I10)
    5 FORMAT(3X,7E11.4)
    6 FORMAT(4I10,F10.4)
   91 FORMAT(10X,'THE IDENTIFIED MODEL IS A POLYNOMIAL IN X1 AND A TABLE
     1 IN U1(AND U2).',/,10X,'THE COEFFICIENTS OF THE POLYNOMIAL FOR',
     2' EACH STEP INPUT(PAIR OF STEP INPUTS) ARE:',//)
      WRITE(6,1)
      READ(5,4) NINPUT,NDEGX1,NGRDX1,NGRDU1,NGRDU2,IPUNCH
      READ(5,3) X1MIN,X1MAX,U1MIN,U1MAX,U2MIN,U2MAX
      NGX1M1 = NGRDX1 - 1
      NDX1P1 = NDEGX1 + 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      IF(NINPUT.EQ.1) NGRDU2 = 1
      WRITE(6,91)
      DO 300 IU2=1,NGRDU2
      DO 200 IU1=1,NGRDU1
      READ(5,6) NIC,NDATA,IDIFF,ISMOTH,TDELTA
      DO 24 I=1,NGX1M1
      X1W(I) = 0.0
      X1DW(I) = 0.0
   24 N1W(I) = 0
      DO 100 IIC=1,NIC
      READ(5,5)(X1S(I),I=1,NDATA)
      IF(IDIFF.EQ.0) GO TO 21
      IF(ISMOTH.EQ.0) GO TO 22
      CALL SE13(X1S,X1S,NDATA,IER)
   22 CALL DET3(TDELTA,X1S,X1DS,NDATA,IER)
      GO TO 23
   21 READ(5,5)(X1DS(I),I=1,NDATA)
   23 DO 30 I=1,NDATA
      IX1 = (X1S(I) - X1MIN)/DX1 + 1.0
      IF(IX1.LT.1) IX1 = 1
      IF(IX1.GT.NGX1M1) IX1 = NGX1M1
      X1W(IX1) = X1W(IX1) + X1S(I)
      X1DW(IX1) = X1DW(IX1) + X1DS(I)
   30 N1W(IX1) = N1W(IX1) + 1
  100 CONTINUE
      K = 0
      DO 40 I=1,NGX1M1
      IF(N1W(I).EQ.0) GO TO 40
      K = K + 1
      DENOM = N1W(I)
      X1W(K) = X1W(I)/DENOM
      X1DW(K) = X1DW(I)/DENOM
   40 CONTINUE
      CALL CURVFT(X1W,X1DW,K,NDEGX1,C1M)
      IF(IPUNCH.EQ.0) GO TO 200
      WRITE(7,5)(C1M(I),I=1,NDX1P1)
  200 WRITE(6,5)(C1M(I),I=1,NDX1P1)
  300 WRITE(6,2)
      WRITE(6,1)
      RETURN
      END
```

```
      SUBROUTINE CURVFT(X,Y,N,NDEGX,C)
C-----
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +        SUBROUTINE CURVFT - LEAST SQUARES CURVE FITTING            +
C     +                                                                   +
C     +        CALLING REQUIREMENTS                                       +
C     +                                                                   +
C     +        X          ARRAY OF VALUES OF INDEPENDENT VARIABLE         +
C     +        Y          ARRAY OF VALUES OF DEPENDENT VARIABLE           +
C     +        N          DIMENSION OF X OR Y                             +
C     +        NDEGX      MAXIMUM DEGREE OF POLYNOMIAL IN X               +
C     +        C          RESULTING COEFFICIENT VECTOR FOR A FUNCTIONAL   +
C     +                   RELATION OF THE FORM Y = SUM( C(I)*(X**(I-1))   +
C     +                                                 I                 +
C     +        SUBPROGRAM REQUIREMENT                                     +
C     +        SIMQ       TO SOLVE LINEAR SIMULTANEOUS EQUATIONS (SSP)    +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      DIMENSION X(1),Y(1),C(1),XKPNM1(21),A(121)
C-----
      NXP1 = NDEGX + 1
      NXT2 = NDEGX*2
      LENTHA = NXP1*NXP1
C-----
      DO 10 IC=1,NXP1
   10 C(IC) = 0.0
      DO 11 IA=1,LENTHA
   11 A(IA) = 0.0
C-----
      DO 100 K=1,N
      XKPNM1(1) = 1.0
      IF(NXT2.EQ.0) GO TO 21
      DO 20 IX=1,NXT2
   20 XKPNM1(IX+1) = XKPNM1(IX)*X(K)
C-----
   21 DO 100 IP=1,NXP1
      IARG = IP - NXP1
      DO 30 I=1,NXP1
      IARG = IARG + NXP1
   30 A(IARG) = A(IARG) + XKPNM1(IP+I-1)
  100 C(IP) = C(IP) + Y(K)*XKPNM1(IP)
C-----
      IF(NXP1.GT.1) GO TO 41
      C(1) = C(1)/A(1)
      RETURN
   41 CALL SIMQ(A,C,NXP1,0)
C-----
      RETURN
      END
```

```
      SUBROUTINE XDOT1(X1,U1,U2,X1DOT)
C-----
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +      THIS SUBROUTINE EVALUATES THE DERIVATIVE FUNCTION X1DOT     +
C     +        FOR ANY VALUES OF X1, U1 AND U2, SPECIFIED THROUGH THE    +
C     +        ARGUMENTS OF THIS PROGRAM.  NOTE THAT THE COEFFICIENTS    +
C     +        C1M(5,11,11) AND NINPUT, X1MIN, X1MAX, NDEGX1, U1MIN,     +
C     +        U1MAX, NGRDU1, U2MIN, U2MAX AND NGRDU2 MUST BE READ IN    +
C     +        A MAIN PROGRAM(WRITTEN BY THE USER) AND TRANSFERRED TO    +
C     +        THIS PROGRAM THROUGH THE COMMON STATEMENT.  THE MODEL     +
C     +        COEFFICIENTS WHICH ARE IDENTIFIED BY THE PROGRAM          +
C     +        'SYSID1' MUST BE READ AS FOLLOWS:                         +
C     +        DO 10 IU2=1,NGRDU2                                        +
C     +        DO 10 IU1=1,NGRDU1                                        +
C     +     10 READ(5,1)(C1M(I,IU1,IU2),I=1,NDX1P1)                      +
C     +      1 FORMAT(3X,7E11.4)                                         +
C     +        WHERE, U1 & U2 ARE DIVIDED INTO NGRDU1 & NGRDU2 LEVELS    +
C     +        AND NDX1P1-1=NDEGX1 IS THE DEGREE OF POLYNOMIAL IN X1.    +
C     +        WHEN NINPUT, THE NUMBER OF INPUTS, IS 1 THEN U2 = 0.0     +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/MNDT1/C1M(5,11,11),NINPUT,X1MIN,X1MAX,NDEGX1,U1MIN,U1MAX,
     1             NGRDU1,U2MIN,U2MAX,NGRDU2
      DIMENSION D1U1(2),D1U2(2)
      NDX1P1 = NDEGX1 + 1
      DU1 = (U1MAX - U1MIN)/(NGRDU1-1)
      IU1 = (U1 - U1MIN)/DU1 + 1.0
      IF(IU1.LT.1) IU1 = 1
      IF(IU1.GE.NGRDU1) IU1 = NGRDU1 - 1
      PERU1 = (U1 - U1MIN - (IU1-1)*DU1)/DU1
      IF(NINPUT.EQ.2) GO TO 10
      IU2 = 1
      GO TO 11
   10 DU2 = (U2MAX - U2MIN)/(NGRDU2 - 1)
      IU2 = (U2 - U2MIN)/DU2 + 1.0
      IF(IU2.LT.1) IU2 = 1
      IF(IU2.GE.NGRDU2) IU2 = NGRDU2 - 1
      PERU2 = (U2 - U2MIN - (IU2-1)*DU2)/DU2
   11 J = IU2 - 1
      DO 20 NU2=1,2
      J = J + 1
      I = IU1 - 1
      DO 30 NU1=1,2
      I = I + 1
      D1U1(NU1) = C1M(J,I,NDX1P1)
      DO 30 IX1=1,NDEGX1
   30 D1U1(NU1) = D1U1(NU1)*X1 + C1M(J,I,NDX1P1-IX1)
      X1DOT = D1U1(1) + (D1U1(2) - D1U1(1))*PERU1
      IF(NINPUT.EQ.2) GO TO 20
      RETURN
   20 D1U2(NU2) = X1DOT
      X1DOT = D1U2(1) + (D1U2(2) - D1U2(1))*PERU2
      RETURN
      END
```

```
      SUBROUTINE SYSID2
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +       THIS PROGRAM IDENTIFIES A SINGLE-INPUT, SECOND-ORDER     +
C     +         SYSTEM.  THE IDENTIFIED MODEL IS A POLYNOMIAL IN X1 AND+
C     +         X2 AND A TABLE IN U1.  THE COEFFICIENTS OF THE POLY-   +
C     +         NOMIAL ARE PRINTED/PUNCHED FOR EACH STEP INPUT.        +
C     +       SUBROUTINE REQUIREMENT-SE13, DET3, SIMQ(SSP) AND SURFIT  +
C     +       THE FOLLOWING DATA IS REQUIRED FOR IDENITIFICATION       +
C     +                                                                +
C     +       FIRST DATA CARD HAS FORMAT 6I10 AND MUST CONTAIN         +
C     +         NDEGX1 AND NDEGX2 - DEGREES OF THE POLYNOMIALS IN X1   +
C     +           AND X2 RESPECTIVELY                                  +
C     +         NGRDX1, NGRDX2 AND NGRDU1 - NUMBERS OF LEVELS INTO     +
C     +           WHICH X1, X2 AND U1 ARE DIVIDED RESPECTIVELY         +
C     +         (USUALLY NDEGX1=NDEGX2=3 AND NGRDX1=NGRDX2=NGRDU1=11)  +
C     +         IPUNCH - 1 IF PUNCHED OUTPUT IS DESIRED(0 OTHERWISE)   +
C     +       SECOND DATA CARD HAS FORMAT 6F10.3 AND MUST CONTAIN      +
C     +         X1MIN, X1MAX, X2MIN, X2MAX, U1MIN AND U1MAX - MINIMUM  +
C     +           AND MAXIMUM VALUES OF X1, X2 AND U1 RESPECTIVELY     +
C     +       THERE MUST BE NGRDU1 SETS OF TEST DATA AFTER THE FIRST   +
C     +         TWO DATA CARDS.  IN THE I-TH DATA SET THE VALUE OF THE +
C     +         STEP IN U1 = U1MIN + (U1MAX-U1MIN)*(I-1)/(NGRDU1-1).   +
C     +       EACH DATA SET MUST FOLLOW A DATA CARD WHICH HAS          +
C     +         FORMAT 4I10, F10.4 AND CONTAINS                        +
C     +         NIC - NUMBER OF INITIAL CONDITIONS(ABOUT 20) WHICH NEED+
C     +           NOT BE SAME(IN VALUE & NUMBER) FOR ALL STEP INPUTS.  +
C     +           WHEN THE INPUT IS U1MAX, THE INITIAL CONDITIONS MAY  +
C     +           BE CHOSEN ONLY ALONG THE LOWER HALF OF THE LOCUS OF  +
C     +           INITIAL CONDITIONS(VICE VERSA FOR U1MIN).            +
C     +         NDATA - NUMBER OF SAMPLES IN EACH RESPONSE(ABOUT 100)  +
C     +         IDIFF - 1 IF THE DERIVATIVES OF X1 AND X2 HAVE TO BE   +
C     +           OBTAINED BY DIFFERENTIATING X1 AND X2(0 OTHERWISE)   +
C     +         ISMOTH - 1 IF SMOOTHING IS REQUIRED(0 OTHERWISE)       +
C     +         TDELTA - THE SAMPLING INTERVAL(CONSTANT)               +
C     +         NOTE THAT IF IDIFF=0, ISMOTH AND TDELTA ARE NOT NEEDED.+
C     +       THE SAMPLED RESPONSES IN EACH DATA SET MUST BE SUPPLIED  +
C     +         IN PUNCHED CARD FORM IN FORMAT 3X, 7E11.4 AS FOLLOWS:  +
C     +         VALUES OF THE STATE X1           | FOR THE FIRST OF THE +
C     +         VALUES OF THE STATE X2           | NIC INITIAL CONDITIONS+
C     +         VALUES OF THE DERIVATIVE OF X1 | (DERIVATIVE VALUES    +
C     +         VALUES OF THE DERIVATIVE OF X2 |  ONLY IF IDIFF=0)     +
C     +         SAME AS ABOVE FOR OTHER INITIAL CONDITIONS             +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      DIMENSION X1S(100),X2S(100),X1DS(100),X2DS(100)
      DIMENSION X1W(11,11),X2W(11,11),X1DW(11,11),X2DW(11,11),NW(11,11)
      DIMENSION C1M(5,5),C2M(5,5),ZX1(121),ZX2(121),ZX1D(121),ZX2D(121)
    2 FORMAT(1H )
    3 FORMAT(8F10.3)
    4 FORMAT(8I10)
    5 FORMAT(3X,7E11.4)
    6 FORMAT(4I10,F10.4)
   91 FORMAT(1H1, 9X,'THE IDENTIFIED MODEL IS A POLYNOMIAL IN X1 AND X2
     1AND A TABLE IN U1',/ ,10X,'THE COEFFICIENTS OF THE POLYNOMIAL FOR
     2EACH OF THE STEP INPUTS ARE:',//)
      WRITE(6,91)
      READ(5,4) NDEGX1,NDEGX2,NGRDX1,NGRDX2,NGRDU1,IPUNCH
      READ(5,3) X1MIN,X1MAX,X2MIN,X2MAX,U1MIN,U1MAX
      NDX1P1 = NDEGX1 + 1
      NDX2P1 = NDEGX2 + 1
      NGX1M1 = NGRDX1 - 1
      NGX2M1 = NGRDX2 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      DX2 = (X2MAX - X2MIN)/NGX2M1
      DO 200 IU1=1,NGRDU1
      DO 20 I=1,NGX1M1
      DO 20 J=1,NGX2M1
      X1W(I,J) = 0.0
      X2W(I,J) = 0.0
      X1DW(I,J) = 0.0
      X2DW(I,J) = 0.0
   20 NW(I,J) = 0
      READ(5,6) NIC,NDATA,IDIFF,ISMOTH,TDELTA
      DO 30 IIC=1,NIC
      READ(5,5)(X1S(I),I=1,NDATA)
      READ(5,5)(X2S(I),I=1,NDATA)
      IF(IDIFF.EQ.0) GO TO 21
      IF(ISMOTH.EQ.0) GO TO 22
      CALL SE13(X1S,X1S,NDATA,IER)
      CALL SE13(X2S,X2S,NDATA,IER)
   22 CALL DET3(TDELTA,X1S,X1DS,NDATA,IER)
      CALL DET3(TDELTA,X2S,X2DS,NDATA,IER)
      GO TO 23
   21 READ(5,5)(X1DS(I),I=1,NDATA)
      READ(5,5)(X2DS(I),I=1,NDATA)
   23 DO 30 I=1,NDATA
      I1 = (X1S(I) - X1MIN)/DX1 + 1.0
      I2 = (X2S(I) - X2MIN)/DX2 + 1.0
      IF(I1.LT.1) I1 = 1
      IF(I1.GT.NGX1M1) I1 = NGX1M1
      IF(I2.LT.1) I2 = 1
      IF(I2.GT.NGX2M1) I2 = NGX2M1
      X1W(I1,I2) = X1W(I1,I2) + X1S(I)
      X2W(I1,I2) = X2W(I1,I2) + X2S(I)
      X1DW(I1,I2) = X1DW(I1,I2) + X2DS(I)
      X2DW(I1,I2) = X2DW(I1,I2) + X2DS(I)
   30 NW(I1,I2) = NW(I1,I2) + 1
      K = 0
      DO 40 I1=1,NGX1M1
      DO 40 I2=1,NGX2M1
      IF(NW(I1,I2).EQ.0) GO TO 40
      K = K + 1
      DENOM = NW(I1,I2)
      ZX1(K) = X1W(I1,I2)/DENOM
      ZX2(K) = X2W(I1,I2)/DENOM
      ZX1D(K) = X1DW(I1,I2)/DENOM
      ZX2D(K) = X2DW(I1,I2)/DENOM
   40 CONTINUE
      CALL SURFIT(ZX1,ZX2,ZX1D,K,NDEGX1,NDEGX2,C1M)
      IF(IPUNCH.EQ.0) GO TO 50
      WRITE(7,5)((C1M(I,J),I=1,NDX1P1),J=1,NDX2P1)
   50 WRITE(6,5)((C1M(I,J),I=1,NDX1P1),J=1,NDX2P1)
      CALL SURFIT(ZX1,ZX2,ZX2D,K,NDEGX1,NDEGX2,C2M)
      IF(IPUNCH.EQ.0) GO TO 51
      WRITE(7,5)((C2M(I,J),I=1,NDX1P1),J=1,NDX2P1)
   51 WRITE(6,5)((C2M(I,J),I=1,NDX1P1),J=1,NDX2P1)
  200 WRITE(6,2)
      RETURN
      END
```

```
      SUBROUTINE SURFIT(X,Y,Z,N,NDEGX,NDEGY,C)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +          SUBROUTINE SURFIT - LEAST SQUARES SURFACE FITTING      +
C     +          CALLING REQUIREMENTS                                  +
C     +          X         ARRAY OF VALUES OF FIRST INDEPENDENT VARIABLE +
C     +          Y         ARRAY OF VALUES OF SECOND INDEPENDENT VARIABLE +
C     +          Z         ARRAY OF VALUES OF DEPENDENT VARIABLE       +
C     +          N         DIMENSION OF X OR Y OR Z                    +
C     +          NDEGX     MAXIMUM DEGREE OF THE POLYNOMIAL IN X       +
C     +          NDEGY     MAXIMUM DEGREE OF THE POLYNOMIAL IN Y       +
C     +          C         RESULTING COEFFICIENT MATRIX FOR THE RELATION +
C     +                    Z = SUM( SUM( C(I,J)*(X**(I-1))*(Y**(J-1))) +
C     +                              J    I                            +
C     +          SUBPROGRAM REQUIREMENT                                +
C     +          SIMQ      TO SOLVE LINEAR SIMULTANEOUS EQUATIONS(SSP) +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      DIMENSION X(1),Y(1),Z(1),C(5,5),XKPNM1(11),YKPNM1(11),A(625),B(25)
      NXP1 = NDEGX + 1
      NYP1 = NDEGY + 1
      NXT2 = NDEGX*2
      NYT2 = NDEGY*2
      LENTHB = NXP1*NYP1
      LENTHA = LENTHB*LENTHB
      DO 10 IB=1,LENTHB
   10 B(IB) = 0.0
      DO 11 IA=1,LENTHA
   11 A(IA) = 0.0
      DO 100 K=1,N
      XKPNM1(1) =1.0
      IF(NXT2.EQ.0) GO TO 21
      DO 20 IX=1,NXT2
   20 XKPNM1(IX+1) = XKPNM1(IX)*X(K)
   21 YKPNM1(1) = 1.0
      IF(NYT2.EQ.0) GO TO 31
      DO 30 IY=1,NYT2
   30 YKPNM1(IY+1) = YKPNM1(IY)*Y(K)
   31 IA = 0
      DO 100 IQ=1,NYP1
      DO 100 IP=1,NXP1
      IA = IA + 1
      IARG = IA - LENTHB
      DO 40 J=1,NYP1
      DO 40 I=1,NXP1
      IARG = IARG + LENTHB
   40 A(IARG) = A(IARG) + XKPNM1(IP+I-1)*YKPNM1(IQ+J-1)
  100 B(IA) = B(IA) + Z(K)*XKPNM1(IP)*YKPNM1(IQ)
      IF(LENTHB.GT.1) GO TO 51
      B(1) = B(1)/A(1)
      GO TO 52
   51 CALL SIMQ(A,B,LENTHB,0)
   52 IJ = 0
      DO 50 J=1,NYP1
      DO 50 I=1,NXP1
      IJ = IJ + 1
   50 C(I,J) = B(IJ)
      RETURN
      END
```

```
      SUBROUTINE XDOT12(X1,X2,U1,X1DOT,X2DOT)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +          THIS SUBROUTINE EVALUATES THE DERIVATIVE FUNCTIONS X1DOT +
C     +            AND X2DOT FOR ANY VALUES OF X1, X2 AND U1, SPECIFIED  +
C     +            THROUGH THE ARGUMENTS OF THE SUBROUTINE.  NOTE THAT THE+
C     +            MODEL COEFFICIENTS C1M(5,5,11),C2M(5,5,11) AND U1MIN, +
C     +            U1MAX, NDEGX1, NDEGX2, AND NGRDU1 MUST BE READ IN A   +
C     +            MAIN PROGRAM(WRITTEN BY THE USER) AND TRANSFERRED TO  +
C     +            THIS SUBROUTINE THROUGH THE COMMON STATEMENT.  THE    +
C     +            MODEL COEFFICIENTS WHICH ARE IDENTIFIED BY THE PROGRAM +
C     +            'SYSID2' MUST BE READ AS FOLLOWS:                     +
C     +            DO 10 K=1,NGRDU1                                      +
C     +            READ(5,1)((C1M(I,J,K),I=1,NDX1P1),J=1,NDX2P1)         +
C     +         10 READ(5,1)((C2M(I,J,K),I=1,NDX1P1),J=1,NDX2P1)         +
C     +          1 FORMAT(3X,7E11.4)                                     +
C     +            WHERE, U1 IS DIVIDED INTO NGRDU1 LEVELS AND           +
C     +            NDX1P1-1 = NDEGX1 AND NDX2P1-1 = NDEGX2 ARE THE DEGREES+
C     +            OF THE POLYNOMIALS IN X1 AND X2 RESPECTIVELY.         +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/MNDT12/C1M(5,5,11),C2M(5,5,11),U1MIN,U1MAX,
     1              NDEGX1,NDEGX2,NGRDU1
      DIMENSION X1DU(2),X2DU(2),X1PNM1(5),X2PNM1(5)
      NDX1P1 = NDEGX1 + 1
      NDX2P1 = NDEGX2 + 1
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1 - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGM1U1 - 1
      PERU1 = (U1 - U1MIN - IU1*DU1)/DU1
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 11
      DO 10 I=1,NDEGX1
   10 X1PNM1(I+1) = X1PNM1(I)*X1
   11 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 21
      DO 20 I=1,NDEGX2
   20 X2PNM1(I+1) = X2PNM1(I)*X2
   21 DO 30 NU=1,2
      IU1 = IU1 + 1
      X1DU(NU) = 0.0
      X2DU(NU) = 0.0
      DO 30 J=1,NDX2P1
      SUM1 = 0.0
      SUM2 = 0.0
      DO 40 I=1,NDX1P1
      SUM1 = SUM1 + C1M(I,J,IU1)*X1PNM1(I)
   40 SUM2 = SUM2 + C2M(I,J,IU1)*X1PNM1(I)
      X1DU(NU) = X1DU(NU) + SUM1*X2PNM1(J)
   30 X2DU(NU) = X2DU(NU) + SUM2*X2PNM1(J)
      X1DOT = X1DU(1) + (X1DU(2) - XIDU(1))*PERU1
      X2DOT = X2DU(1) + (X2DU(2) - X2DU(1))*PERU1
      RETURN
      END
```

APPENDIX B

CONVERSION SUBROUTINE

This appendix includes a conversion subroutine, CONVRT, which can be used to generate the coefficients of a model form which is tabular both in X and U or polynomial both in X and U. This subroutine requires the coefficients of the standard form of the model which are obtained by SYSID1 or SYSID2 in Appendix A. In addition, proper versions of the subroutines XDOT1 and XDOT12 which can use the model coefficients generated by CONVRT are also presented. All of these subroutines contain the necessary explanation.

```
      SUBROUTINE CONVRT(NORDER,NINPUT,MFORM)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                +
C     +       THIS SUBROUTINE IS PREPARED FOR FIRST-ORDER SYSTEMS AND  +
C     +         SECOND-ORDER SYSTEMS TO GENERATE THE COEFFICIENTS OF A +
C     +         MODEL FORM WHICH IS COMPLETELY TABULAR OR COMPLETELY   +
C     +         POLYNOMIAL IN THE STATES AND IN THE INPUTS.  THE COEF- +
C     +         FICIENTS OF THE MIXED FORM OF THE MODEL WHICH ARE OBTA-+
C     +         INED BY THE SUBROUTINE SYSID1 OR SYSID2 ARE READ IN BY +
C     +         THIS PROGRAM AS INPUT DATA.  THE GENERATED COEFFICIENTS+
C     +         ARE PRINTED IN THIS SUBROUTINE.  HOWEVER, THESE WILL BE+
C     +         AVAILABLE IN THE USER WRITTEN MAIN PROGRAM THROUGH THE +
C     +         COMMON (BLOCK) STATEMENT IF PUNCHED OUTPUT IS DESIRED. +
C     +         COEFFICIENTS CC2 ARE INDENTED FOR CONVENIENCE.         +
C     +                                                                +
C     +       THE EXPLANATION FOR THE ARGUMENTS IS AS FOLLOWS:         +
C     +         NORDER - 1 FOR FIRST-ORDER SYSTEMS                     +
C     +                  2 FOR SECOND-ORDER SYSTEMS                    +
C     +         NINPUT - 1 FOR SINGLE-INPUT SYSTEMS                    +
C     +                  2 FOR DUAL-INPUT SYSTEM                       +
C     +         NOTE: THE CASE WHERE NORDER=NINPUT=2 IS NOT CONSIDERED +
C     +                                                                +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM (WRITTEN BY USER) THROUGH THE COMMON (BLOCK)   +
C     +         STATEMENT.  THE FOLLOWING IS THE EXPLANATION:          +
C     +           C1 AND C2 - THE COEFFICIENTS OF THE MIXED FORM OF THE+
C     +                       MODEL - THREE DIMENSIONAL ARRAYS         +
C     +           CC1 AND CC2 - THE COEFFICIENTS OF THE DESIRED FORM   +
C     +                         OF THE MODEL (TABULAR OR POLYNOMIAL)   +
C     +           X1MIN,X1MAX,X2MIN,X2MAX,U1MIN,U1MAX,U2MIN AND U2MAX  +
C     +             ARE THE MINIMUM AND THE MAXIMUM LIMITS ON X1, X2,  +
C     +             U1 AND U2 RESPECTIVELY                             +
C     +           NGRDX1, NGRDX2, NGRDU1 AND NGRDU2 ARE THE NUMBERS OF +
C     +             LEVELS INTO WHICH X1, X2, U1 AND U2 ARE DIVIDED    +
C     +           NDEGX1, NDEGX1, NDEGU1 AND NDEGU2 ARE THE DEGREES OF +
C     +             THE POLYNOMIALS IN X1, X2, U1 AND U2 RESPECTIVELY  +
C     +           NOTE: WHEN A TABULAR FORM OF THE MODEL IS DESIRED,   +
C     +                 NDEGU1 AND NDEGU2 NEED NOT BE SPECIFIED.       +
C     +                                                                +
C     +       FOR A SINGLE INPUT SYSTEM NGRDU2 = 1 AND NDEGU2 = 0      +
C     +                                                                +
C     +       SUBROUTINE REQUIREMENT                                   +
C     +                                                                +
C     +         CURVFT - FOR FITTING CURVES THROUGH DATA POINTS IN A   +
C     +                  LEAST SQUARES SENSE ( SEE APPENDIX A).        +
C     +         SIMQ -   FOR SOLVING LINEAR ALGEBRAIC EQUATIONS ( FROM +
C     +                  IBM SCIENTIFIC SUBROUTINE PACKAGE).           +
C     +                                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(11,11,11),C2(11,11,11),CC1(11,11,11),CC2(11,11,11)
     1            ,X1MIN,X1MAX,NGRDX1,NDEGX1,X2MIN,X2MAX,NGRDX2,NDEGX2,
     2            U1MIN,U1MAX,NGRDU1,NDEGU1,U2MIN,U2MAX,NGRDU2,NDEGU2
      DIMENSION X1(21),X2(21),U1(21),U2(21),X1PNM1(10),X2PNM1(10),
     1          WC1(5,5,11),Y1(21),Y2(21),COEF1(10),COEF2(10)
C-----
    1 FORMAT(1H1)
    2 FORMAT(1H )
    3 FORMAT(3X,7E11.4)
    4 FORMAT(3X,11E11.4)
    5 FORMAT(8X,11E11.4)
    6 FORMAT(1H1,2X,'THE FOLLOWING COEFFICIENTS OF A MODEL FORM, WHICH I
     1S TABULAR IN THE INPUT(S) AND POLYNOMIAL IN THE STATE(S),',/,3X,
     2 'WERE READ IN :',//)
    7 FORMAT(1H1,2X,'THE FOLLOWING ARE THE COEFFICIENTS OF THE DESIRED F
     1ORM:',//)
      WRITE(6,6)
C-----
      NDX1P1 = NDEGX1 + 1
      IF(NORDER.EQ.2) GO TO 22
      IF(NINPUT.EQ.1) NGRDU2 = 1
      DO 21 K=1,NGRDU2
      DO 21 J=1,NGRDU1
      READ(5,3) (C1(I,J,K),I=1,NDX1P1)
   21 WRITE(6,4)(C1(I,J,K),I=1,NDX1P1)
      GO TO 25
   22 NDX2P1 = NDEGX2 + 1
      DO 23 K=1,NGRDU1
      WRITE(6,2)
      READ(5,3) ((C1(I,J,K),I=1,NDX1P1),J=1,NDX2P1)
      READ(5,3) ((C2(I,J,K),I=1,NDX1P1),J=1,NDX2P1)
      DO 23 L=1,NDX2P1
      WRITE(6,4)(C1(I,L,K),I=1,NDX1P1)
   23 WRITE(6,5)(C2(I,L,K),I=1,NDX1P1)
C-----
   25 WRITE(6,7)
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      IF(MFORM.EQ.1) GO TO 100
C-----
      X1(1) = X1MIN
      DO 30 I=1,NGX1M1
   30 X1(I+1) = X1(I) + DX1
      IF(NORDER.EQ.2) GO TO 61
C-----
      DO 51 K=1,NGRDU2
      DO 51 J=1,NGRDU1
      DO 50 I=1,NGRDX1
      SUM1 = 0.0
      DO 40 L=1,NDX1P1
   40 SUM1 = SUM1*X1(I) + C1(NDX1P1+1-L,J,K)
   50 CC1(I,J,K) = SUM1
   51 WRITE(6,4)(CC1(I,J,K),I=1,NGRDX1)
      RETURN
C-----
   61 NDX2P1 = NDEGX2 + 1
      NGX2M1 = NGRDX2 - 1
      DX2 = (X2MAX - X2MIN)/NGX2M1
      X2(1) = X2MIN
      DO 62 I=1,NGX2M1
   62 X2(I+1) = X2(I) + DX2
C-----
      DO 91 IU1=1,NGRDU1
      WRITE(6,2)
      DO 91 IX2=1,NGRDX2
      DO 90 IX1=1,NGRDX1
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 82
      DO 81 I=1,NDEGX1
```

```
   81 X1PNM1(I+1) = X1PNM1(I)*X1(IX1)
   82 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 84
      DO 83 I=1,NDEGX2
   83 X2PNM1(I+1) = X2PNM1(I)*X2(IX2)
C-----
   84 SUM1X2 = 0.0
      SUM2X2 = 0.0
      DO 86 J=1,NDX2P1
      SUM1X1 = 0.0
      SUM2X1 = 0.0
      DO 85 I=1,NDX1P1
      SUM1X1 = SUM1X1 + C1(I,J,IU1)*X1PNM1(I)
   85 SUM2X1 = SUM2X1 + C2(I,J,IU1)*X1PNM1(I)
      SUM1X2 = SUM1X2 + SUM1X1*X2PNM1(J)
   86 SUM2X2 = SUM2X2 + SUM2X1*X2PNM1(J)
C-----
      CC1(IX1,IX2,IU1) = SUM1X2
   90 CC2(IX1,IX2,IU1) = SUM2X2
      WRITE(6,4)(CC1(I,IX2,IU1),I=1,NGRDX1)
   91 WRITE(6,5)(CC2(I,IX2,IU1),I=1,NGRDX1)
      RETURN
C-----
  100 NGU1M1 = NGRDU1 - 1
      NDU1P1 = NDEGU1 + 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      U1(1) = U1MIN
      DO 110 I=1,NGU1M1
  110 U1(I+1) = U1(I) + DU1
      IF(NORDER.EQ.2) GO TO 181
C-----
      DO 140 IU2=1,NGRDU2
      DO 140 IX1=1,NDX1P1
      DO 120 NU1=1,NGRDU1
  120 Y1(NU1) = C1(IX1,NU1,NGRDU2)
      CALL CURVFT(U1,Y1,NGRDU1,NDEGU1,COEF1)
      DO 130 IU1=1,NDU1P1
  130 WC1(IX1,IU1,NGRDU2) = COEF1(IU1)
  140 CONTINUE
      IF(NINPUT.EQ.2) GO TO 151
C-----
      DO 150 IU1=1,NDU1P1
      WRITE(6,4)(WC1(I,IU1,1),I=1,NDX1P1)
      DO 150 IX1=1,NDX1P1
  150 CC1(IX1,IU1,1) = WC1(IX1,IU1,1)
      RETURN
C-----
  151 NDU2P1 = NDEGU2 + 1
      NGU2M1 = NGRDU2 - 1
      DU2 = (U2MAX - U2MIN)/NGU2M1
      U2(1) = U2MIN
      DO 152 I=1,NGU2M1
  152 U2(I+1) = U2(I) + DU2
      DO 180 IU1=1,NDU1P1
      DO 180 IX1=1,NDX1P1
      DO 160 NU2=1,NGRDU2
  160 Y1(NU2) = WC1(IX1,IU1,IU2)
      CALL CURVFT(U2,Y1,NGRDU2,NDEGU2,COEF1)
      DO 170 IU2=1,NDU2P1
  170 CC1(IX1,IU1,IU2) = COEF1(IU2)
  180 CONTINUE
      DO 171 K=1,NDU2P1
      WRITE(6,2)
      DO 171 J=1,NDU1P1
  171 WRITE(6,4)(CC1(I,J,K),I=1,NDX1P1)
C-----
  181 DO 200 IX2=1,NDX2P1
      DO 200 IX1=1,NDX1P1
      DO 190 NU1=1,NGRDU1
      Y1(NU1) = C1(IX1,IX2,NU1)
  190 Y2(NU1) = C2(IX1,IX2,NU1)
      CALL CURVFT(U1,Y1,NGRDU1,NDEGU1,COEF1)
      CALL CURVFT(U1,Y2,NGRDU1,NDEGU1,COEF2)
      DO 191 IU1=1,NDU1P1
      CC1(IX1,IX2,IU1) = COEF1(IU1)
  191 CC2(IX1,IX2,IU1) = COEF2(IU1)
  200 CONTINUE
      DO 201 K=1,NDU1P1
      WRITE(6,2)
      DO 201 J=1,NDX2P1
      WRITE(6,4)(CC1(I,J,K),I=1,NDX1P1)
  201 WRITE(6,5)(CC2(I,J,K),I=1,NDX1P1)
      RETURN
      END
```

```
      SUBROUTINE XDOT1(X1,U1,U2,X1DOT)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +       THIS SUBROUTINE EVALUATES THE DERIVATIVE FUNCTION X1DOT  +
C     +         FOR ANY VALUES OF X1, U1 AND U2, SPECIFIED THROUGH THE +
C     +         ARGUMENTS OF THIS PROGRAM.  NOTE THE QUANTITIES WHICH  +
C     +         ARE TRANSFERRED FROM A MAIN PROGRAM THROUGH THE COMMON +
C     +         (MNDT1) STATEMENT.  THIS PROGRAM IS PREPARED FOR FIRST +
C     +         ORDER SYSTEMS.  THE COEFFICIENTS C1 OF THE MODEL WHICH +
C     +         IS COMPLETELY TABULAR OR COMPLETELY POLYNOMIAL IN X1,  +
C     +         U1 AND U2 ARE USED.  THE FOLLOWING IS THE EXPLANATION  +
C     +         FOR THE VARIABLES USED IN THE COMMON STATEMENT:        +
C     +           NINPUT - 1 FOR SINGLE-INPUT SYSTEM                   +
C     +                    2 FOR DUAL-INPUT SYSTEM                     +
C     +           MFORM - 0 FOR TABULAR FORM                           +
C     +                   1 FOR POLYNOMIAL FORM                        +
C     +           X1MIN, X1MAX, U1MIN, U1MAX, U2MIN AND U2MAX ARE THE  +
C     +             MINIMUM AND THE MAXIMUM LIMITS ON X1, U1 AND U2.   +
C     +           NGRDX1, NGRDU1 AND NGRDU2 - THE NUMBERS OF LEVELS    +
C     +             INTO WHICH X1, U1 AND U2 ARE DIVIDED.              +
C     +           NDEGX1, NDEGU1 AND NDEGU2 - THE DEGREES OF THE POLY- +
C     +             NOMIALS IN X1, U1 AND U2 RESPECTIVELY.             +
C     +             NOTE: FOR A SINGLE-INPUT SYSTEM, NGRDU2 = 1 AND    +
C     +                   NDEGU2 = 0                                   +
C     +       IF THE COEFFICIENTS C1 ARE GENERATED BY THE SUBROUTINE   +
C     +         CONVRT, THESE MAY BE READ IN AS FOLLOWS:               +
C     +                                                                +
C     +         3 FORMAT(3X,7E11.4)                                    +
C     +           DO 10 K=1,NU2                                        +
C     +           DO 10 J=1,NU1                                        +
C     +        10 READ(5,3)(C1(1,J,K),I=1,NX1)                         +
C     +                                                                +
C     +       WHERE, NX1 = NGRDX1, NU1 = NGRDU1 AND NU2 = NGRDU2 FOR A +
C     +         TABULAR FORM AND NX1 = NDEGX1 + 1, NU1 = NDEGU1 + 1    +
C     +         AND NU2 = NDEGU2 + 1 FOR A POLYNOMIAL FORM.            +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/MNDT1/C1(11,11,11),NINPUT,MFORM,X1MIN,X1MAX,NGRDX1,NDEGX1,
     1             U1MIN,U1MAX,NGRDU1,NDEGU1,U2MIN,U2MAX,NGRDU2,NDEGU2
      DIMENSION X1PNM1(10),U1PNM1(10),U2PNM1(10)
      IF(MFORM.EQ.1) GO TO 300
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      NX1 = (X1 - X1MIN)/DX1
      IF(NX1.LT.0) NX1 = 0
      IF(NX1.GE.NGX1M1) NX1 = NGX1M1 - 1
      PERX1 = (X1 - X1MIN - NX1*DX1)/DX1
      I = NX1 + 1
      IF(NGRDU1.NE.1) GO TO 210
      DTX1 = C1(I,1,1)
      DTX2 = C1(I+1,1,1)
      GO TO 240
  210 NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      NU1 = (U1 - U1MIN)/DU1
      IF(NU1.LT.0) NU1 = 0
      IF(NU1.GE.NGU1M1) NU1 = NGU1M1 - 1
      PERU1 = (U1 - U1MIN - DU1*NU1)/DU1
      J = NU1 + 1
      IF(NINPUT.EQ.2) GO TO 220
      DTXU11 = C1(I,J,1)
      DTXU12 = C1(I,J+1,1)
      DTXU21 = C1(I+1,J,1)
      DTXU22 = C1(I+1,J+1,1)
      GO TO 230
  220 NGU2M1 = NGRDU2 - 1
      DU2 = (U2MAX - U2MIN)/NGU2M1
      NU2 = (U2 - U2MIN)/DU2
      IF(NU2.LT.0) NU2 = 0
      IF(NU2.GE.NGU2M1) NU2 = NGU2M1 - 1
      PERU2 = (U2 - U2MIN - DU2*NU2)/DU2
      K = NU2 + 1
      DTXU11 = C1(I,J,K) + (C1(I,J,K+1) - C1(I,J,K))*PERU2
      DTXU12 = C1(I,J+1,K) + (C1(I,J+1,K+1) - C1(I,J+1,K))*PERU2
      DTXU21 = C1(I+1,J,K) + (C1(I+1,J,K+1) - C1(I+1,J,K))*PERU2
      DTXU22 = C1(I+1,J+1,K) + (C1(I+1,J+1,K+1) - C1(I+1,J+1,K))*PERU2
  230 DTX1 = DTXU11 + (DTXU12 - DTXU11)*PERU1
      DTX2 = DTXU21 + (DTXU22 - DTXU21)*PERU1
  240 X1DOT = DTX1 + (DTX2 - DTX1)*PERX1
      RETURN
  300 NDX1P1 = NDEGX1 + 1
      NDU1P1 = NDEGU1 + 1
      IF(NINPUT.EQ.1) NDEGU2 = 0
      NDU2P1 = NDEGU2 + 1
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 311
      DO 310 I=1,NDEGX1
  310 X1PNM1(I+1) = X1PNM1(I)*X1
  311 U1PNM1(1) = 1.0
      IF(NDEGU1.EQ.0) GO TO 321
      DO 320 I=1,NDEGU1
  320 U1PNM1(I+1) = U1PNM1(I)*U1
  321 U2PNM1(1) = 1.0
      IF(NDEGU2.EQ.0) GO TO 331
      DO 330 I=1,NDEGU2
  330 U2PNM1(I+1) = U2PNM1(I)*U2
  331 X1DOT = 0.0
      DO 360 IU2=1,NDU2P1
      SUM1U1 = 0.0
      DO 350 IU1=1,NDU1P1
      SUM1X1 = 0.0
      DO 340 IX1=1,NDX1P1
  340 SUM1X1 = SUM1X1 + C1(IX1,IU1,IU2)*X1PNM1(IX1)
  350 SUM1U1 = SUM1U1 + SUM1X1*U1PNM1(IU1)
  360 X1DOT = X1DOT + SUM1U1*U2PNM1(IU2)
      RETURN
      END
```

```
      SUBROUTINE XDOT12(X1,X2,U1,X1DOT,X2DOT)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                            +
C     +      THIS SUBROUTINE EVALUATES THE DERIVATIVE FUNCTIONS X1DOT +
C     +        AND X2DOT FOR ANY VALUES OF X1, X2 AND U1, SPECIFIED  +
C     +        THROUGH THE ARGUMENTS OF THIS PROGRAM.  NOTE THE QUAN- +
C     +        TITIES WHICH ARE TRANSFERRED THROUGH THE COMMON STATE- +
C     +        MENT (MNDT12) FROM A MAIN PROGRAM.  THIS SUBROUTINE IS +
C     +        PREPARED FOR SINGLE-INPUT, SECOND-ORDER SYSTEMS.  THE  +
C     +        COEFFICIENTS OF THE MODEL FORM WHICH IS COMPLETELY     +
C     +        TABULAR OR COMPLETELY POLYNOMIAL IN X1, XI AND U1 ARE  +
C     +        USED.  THE FOLLOWING IS THE EXPLANATION FOR THE VARI-  +
C     +        ABLES USED IN THE COMMON STATEMENT:                     +
C     +                                                            +
C     +          MFORM - 0 FOR TABULAR FORM                        +
C     +                  1 FOR POLYNOMIAL FORM                     +
C     +          NGRDX1, NGRDX2 AND NGRDU1 - THE NUMBERS OF LEVELS  +
C     +            INTO WHICH X1, X2 AND U1 ARE DIVIDED.            +
C     +          NDEGX1, NDEGX2 AND NDEGU1 - THE DEGREES OF THE POLY- +
C     +            NOMIALS IN X1, X2 AND U1 RESPECTIVELY.           +
C     +            NOTE: FOR A SYSTEM WITH NO INPUT, NGRDU1 = 1 AND +
C     +                  NDEGU1 = 0                                  +
C     +                                                            +
C     +      THE DIMENSIONS OF THE COEFFICIENTS C1(NX1,NX2,NU1) AND  +
C     +        C2(NX1,NX2,NU1) ARE AS FOLLOWS:                       +
C     +        NX1 = NGRDX1, NX2 = NGRDX2 AND NU1 = NGRDU1 WHEN THE  +
C     +        MODEL FORM IS TABULAR, AND FOR A POLYNOMIAL MODEL FORM +
C     +        NX1 = NDEGX1+1, NX2 = NDEGX2+1 AND NU1 = NDEGU1+1.    +
C     +                                                            +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/MNDT12/C1(11,11,11),C2(11,11,11),X1MIN,X1MAX,NGRDX1,NDEGX1,
     1        MFORM,X2MIN,X2MAX,NGRDX2,NDEGX2,U1MIN,U1MAX,NGRDU1,NDEGU1
      DIMENSION X1PNM1(10),X2PNM1(10),U1PNM1(10)
      IF(MFORM.EQ.1) GO TO 300
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      NX1 = (X1 - X1MIN)/DX1
      IF(NX1.LT.0) NX1 = 0
      IF(NX1.GE.NGX1M1) NX1 = NGX1M1
      PERX1 = (X1 - X1MIN - NX1*DX1)/DX1
      I = NX1 + 1
      NGX2M1 = NGRDX2 - 1
      DX2 = (X2MAX - X2MIN)/NGX2M1
      NX2 = (X2 - X2MIN)/DX2
      IF(NX2.LT.0) NX2 = 0
      IF(NX2.GE.NGX2M1) NX2 = NGX2M1 - 1
      PERX2 = (X2 - X2MIN - DX2*NX2)/DX2
      J = NX2 + 1
      IF(NGRDU1.NE.1) GO TO 210
      X1DT11 = C1(I,J,1)
      X1DT12 = C1(I,J+1,1)
      X1DT21 = C1(I+1,J,1)
      X1DT22 = C1(I+1,J+1,1)
      X2DT11 = C2(I,J,1)
      X2DT12 = C2(I,4+1,1)
      X2DT21 = C2(I+1,J,1)
      X2DT22 = C2(I+1,J+1,1)
      GO TO 220
  210 NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      NU1 = (U1 - U1MIN)/DU1
      IF(NU1.LT.0) NU1 = 0
      IF(NU1.GE.NGU1M1) NU1 = NGU1 - 1
      PERU1 = (U1 - U1MIN - DU1*NU1)/DU1
      K = NU1 + 1
      X2DT11 = C2(I,J,K) + (C2(I,J,K+1) - C2(I,J,K))*PERU1
      X2DT12 = C2(I,J+1,K) + (C2(I,J+1,K+1) - C2(I,J+1,K))*PERU1
      X2DT21 = C2(I+1,J,K) + (C2(I+1,J,K+1) - C2(I+1,J,K))*PERU1
      X2DT22 = C2(I+1,J+1,K) + (C2(I+1,J+1,K+1) - C2(I+1,J+1,K))*PERU1
      X1DT11 = C1(I,J,K) + (C1(I,J,K+1) - C1(I,J,K))*PERU1
      X1DT12 = C1(I,J+1,K) + (C1(I,J+1,K+1) - C1(I,J+1,K))*PERU1
      X1DT21 = C1(I+1,J,K) + (C1(I+1,J,K+1) - C1(I+1,J,K))*PERU1
      X1DT22 = C1(I+1,J+1,K) + (C1(I+1,J+1,K+1) - C1(I+1,J+1,K))*PERU1
  220 DT1X11 = X1DT11 + (X1DT12 - X1DT11)*PERX2
      DT1X12 = X1DT21 + (X1DT22 - X1DT21)*PERX2
      DT2X11 = X2DT11 + (X2DT12 - X2DT11)*PERX2
      DT2X12 = X2DT21 + (X2DT22 - X2DT21)*PERX2
      X1DOT = DT1X11 + (DT1X12 - DT1X11)*PERX1
      X2DOT = DT2X11 + (DT2X12 - DT2X11)*PERX1
      RETURN
  300 NDX1P1 = NDEGX1 + 1
      NDX2P1 = NDEGX2 + 1
      NDU1P1 = NDEGU1 + 1
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 311
      DO 310 I=1,NDEGX1
  310 X1PNM1(I+1) = X1PNM1(I)*X1
  311 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 321
      DO 320 I=1,NDEGX2
  320 X2PNM1(I+1) = X2PNM1(I)*X2
  321 U1PNM1(1) = 1.0
      IF(NDEGU1.EQ.0) GO TO 331
      DO 330 I=1,NDEGU1
  330 U1PNM1(I+1) = U1PNM1(I)*U1
  331 X1DOT = 0.0
      X2DOT = 0.0
      DO 360 IU1=1,NDU1P1
      SUM1X2 = 0.0
      SUM2X2 = 0.0
      DO 350 IX2=1,NDX2P1
      SUM1X1 = 0.0
      SUM2X1 = 0.0
      DO 340 IX1=1,NDX1P1
      SUM1X1 = SUM1X1 + C1(IX1,IX2,IU1)*X1PNM1(IX1)
  340 SUM2X1 = SUM2X1 + C2(IX1,IX2,IU1)*X1PNM1(IX1)
      SUM1X2 = SUM1X2 + SUM1X1*X2PNM1(IX2)
  350 SUM2X2 = SUM2X2 + SUM2X1*X2PNM1(IX2)
      X1DOT = X1DOT + SUM1X2*U1PNM1(IU1)
  360 X2DOT = X2DOT + SUM2X2*U1PNM1(IU1)
      RETURN
      END
```

APPENDIX C

SUBROUTINES USED IN THE EXAMPLES

This appendix includes the computer subroutines which were used in Examples 1, 2 and 3 for modal analysis. Each example used different versions of the subroutines STEADY and LINRIZ. These programs contain the necessary explanation. Note that the two subroutines used in Example 3 can be used with the coefficients for the standard form of the model obtained by SYSID2 for second-order systems.

```
      SUBROUTINE STEADY(U1STEP,X1SS)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                +
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 1 FOR DETERMINING     +
C     +         THE STEADY-STATE VALUE X1SS FOR A STEP INPUT U1STEP.   +
C     +                                                                +
C     +       THIS SUBROUTINE IS PREPARED FOR SINGLE-INPUT, FIRST-ORDER+
C     +         SYSTEMS.  THE COEFFICIENTS C1(NGRDX1,NGRDU1) OF THE    +
C     +         MODEL FORM WHICH IS TABULAR BOTH IN X1 AND U1 ARE USED.+
C     +                                                                +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.          +
C     +                                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(21,21),X1MIN,X1MAX,NGRDX1,U1MIN,U1MAX,NGRDU1
      DIMENSION X1DEF(21)
C-----
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
C-----
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1STEP - U1MIN)/DU1
      IF(IU1.LT.0) IU1=0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1STEP - U1MIN - DU1*IU1)/DU1
      IU1 = IU1 + 1
C-----
      DO 10 IX1=1,NGRDX1
   10 X1DEF(IX1) = C1(IX1,IU1) + (C1(IX1,IU1+1)-C1(IX1,IU1))*PERU1
C-----
      IEF = 2
      DO 20 IX1=1,NGRDX1
      IF(X1DEF(IX1).LE.0.0) GO TO 21
   20 IEF = IX1
C-----
   21 IEF = IEF + 1
      IF(IEF.LT.2) IEF = 2
      PERX1D = (0.0 - X1DEF(IEF-1))/(X1DEF(IEF) - X1DEF(IEF-1))
      X1SS = X1MIN + (IEF - 2)*DX1 + DX1*PERX1D
C-----
      RETURN
      END
```

```
      SUBROUTINE LINRIZ(X1OP,U1OP,A,B)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                +
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 1 FOR DETERMINING     +
C     +         THE COEFFICIENT MATRICES A AND B (IN THIS CASE SCALARS)+
C     +         OF THE LINEARIZED DIFFERENTIAL EQUATION FOR OPERATION  +
C     +         IN THE SMALL ABOUT THE POINT (X1OP,U1OP).              +
C     +                                                                +
C     +       THIS SUBROUTINE IS PREPARED FOR SINGLE-INPUT, FIRST-ORDER+
C     +         SYSTEMS.  THE COEFFICIENTS C1(NGRDX1,NGRDU1) OF THE    +
C     +         MODEL FORM WHICH IS TABULAR BOTH IN X1 AND U1 ARE USED.+
C     +                                                                +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.          +
C     +                                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(21,21),X1MIN,X1MAX,NGRDX1,U1MIN,U1MAX,NGRDU1
C-----
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      IX1 = (X1OP - X1MIN)/DX1
      IF(IX1.LT.0) IX1 = 0
      IF(IX1.GE.NGX1M1) IX1 = NGX1M1 - 1
      PERX1 = (X1OP - X1MIN - DX1*IX1)/DX1
      IX1 = IX1 + 1
C-----
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1OP - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1OP - U1MIN - IU1*DU1)/DU1
      IU1 = IU1 + 1
C-----
      X1DX1 = C1(IX1,IU1) + (C1(IX1,IU1+1)-C1(IX1,IU1))*PERU1
      X1DX2 = C1(IX1+1,IU1) + (C1(IX1+1,IU1+1)-C1(IX1+1,IU1))*PERU1
      A = (X1DX2 - X1DX1)/DX1
C-----
      X1DU1 = C1(IX1,IU1) + (C1(IX1+1,IU1)-C1(IX1,IU1))*PERX1
      X1DU2 = C1(IX1,IU1+1) + (C1(IX1+1,IU1+1)-C1(IX1,IU1+1))*PERX1
      B = (X1DU2 - X1DU1)/DU1
C-----
      RETURN
      END
```

```
      SUBROUTINE STEADY(U1STEP,U2STEP,X1SS)
C-----
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 2 FOR DETERMINING    +
C     +         THE STEADY-STATE X1SS FOR A PAIR OF STEP INPUTS U1STEP +
C     +         AND U2STEP.                                            +
C     +                                                               +
C     +       THIS SUBROUTINE IS PREPARED FOR DUAL-INPUT, FIRST-ORDER  +
C     +         SYSTEMS.  THE COEFFICIENTS C1(NGRDX1,NGRDU1,NGRDU2) OF +
C     +         THE MODEL FORM , WHICH IS TABULAR IN X1, U1 AND U2     +
C     +         ARE USED.                                              +
C     +                                                               +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.          +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(11,11,11),X1MIN,X1MAX,NGRDX1,U1MIN,U1MAX,NGRDU1,
     1             U2MIN,U2MAX,NGRDU2
      DIMENSION X1DEF(21)
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1STEP - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1STEP - U1MIN - DU1*IU1)/DU1
      J = IU1 + 1
      IF(NGRDU2.NE.1) GO TO 41
      DO 30 IX1=1,NGRDX1
   30 X1DEF(IX1) = C1(IX1,J,1) + (C1(IX1,J+1,1) - C1(IX1,J,1))*PERU1
      GO TO 10
   41 NGU2M1 = NGRDU2 - 1
      DU2 = (U2MAX - U2MIN)/NGU2M1
      DU2 = (U2MAX - U2MIN)/NGU2M1
      IU2 = (U2STEP - U2MIN)/DU2
      IF(IU2.LT.0) IU2 = 0
      IF(IU2.GE.NGU2M1) IU2 = NGU2M1 - 1
      PERU2 = (U2STEP - U2MIN - DU2*IU2)/DU2
      K = IU2 + 1
      DO 40 IX1=1,NGRDX1
      X1DU11 = C1(IX1,J,K) + (C1(IX1,J,K+1) - C1(IX1,J,K))*PERU2
      X1DU12 = C1(IX1,J+1,K) + (C1(IX1,J+1,K+1) - C1(IX1,J+1,K))*PERU2
   40 X1DEF(IX1) = X1DU11 + (X1DU12 - X1DU11)*PERU1
   10 IEF = 2
      DO 20 IX1=1,NGRDX1
      IF(X1DEF(IX1).LE.0.0) GO TO 21
   20 IEF = IX1
   21 IEF = IEF + 1
      IF(IEF.LT.2) IEF = 2
      PERX1D = (0.0 - X1DEF(IEF-1))/(X1DEF(IEF) - X1DEF(IEF-1))
      X1SS = X1MIN + (IEF - 2)*DX1 + DX1*PERX1D
      RETURN
      END
```

```
      SUBROUTINE LINRIZ(X1OP,U1OP,U2OP,A,B)
C-----
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 2 FOR DETERMINING    +
C     +         THE COEFFICIENT MATRICES A AND B OF THE DIFFERENTIAL   +
C     +         EQUATION LINEARIZED ABOUT THE POINT (X1OP,U1OP,U2OP).  +
C     +         NOTE A IS A SCALAR AND B IS A TWO COMPONENT VECTOR.    +
C     +                                                               +
C     +       THIS SUBROUTINE IS PREPARED FOR DUAL-INPUT, FIRST-ORDER  +
C     +         SYSTEMS.  C1(NGRDX1,NGRDU1,NGRDU2) ARE THE COEFFICIENTS+
C     +         OF THE MODEL WHICH IS TABULAR IN X1, U1 AND U2.        +
C     +                                                               +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.          +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(11,11,11),X1MIN,X1MAX,NGRDX1,U1MIN,U1MAX,NGRDU1,
     1             U2MIN,U2MAX,NGRDU2
      DIMENSION B(2)
      NGX1M1 = NGRDX1 - 1
      DX1 = (X1MAX - X1MIN)/NGX1M1
      IX1 = (X1OP - X1MIN)/DX1
      IF(IX1.LT.0) IX1 = 0
      IF(IX1.GE.NGX1M1) IX1 = NGX1M1 - 1
      PERX1 = (X1OP - X1MIN - DX1*IX1)/DX1
      I = IX1 + 1
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1OP - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1OP - U1MIN - IU1*DU1)/DU1
      J = IU1 + 1
      NGU2M1 = NGRDU2 - 1
      DU2 = (U2MAX - U2MIN)/NGU2M1
      IU2 = (U2OP - U2MIN)/DU2
      IF(IU2.LT.0) IU2 = 0
      IF(IU2.GE.NGU2M1) IU2 = NGU2M1 - 1
      PERU2 = (U2OP - U2MIN - IU2*DU2)/DU2
      K = IU2 + 1
      X1U111 = C1(I,J,K) + (C1(I,J,K+1) - C1(I,J,K))*PERU2
      X1U112 = C1(I,J+1,K) + (C1(I,J+1,K+1) - C1(I,J+1,K))*PERU2
      X1U121 = C1(I+1,J,K) + (C1(I+1,J,K+1) - C1(I+1,J,K))*PERU2
      X1U122 = C1(I+1,J+1,K) + (C1(I+1,J+1,K+1) - C1(I+1,J+1,K))*PERU2
      X1DX11 = X1U111 + (X1U112 - X1U111)*PERU1
      X1DX12 = X1U121 + (X1U122 - X1U121)*PERU1
      A = (X1DX12 - X1DX11)/DX1
      U1U211 = C1(I,J,K) + (C1(I+1,J,K) - C1(I,J,K))*PERX1
      U1U212 = C1(I,J,K+1) + (C1(I+1,J,K+1) - C1(I,J,K+1))*PERX1
      U1U221 = C1(I,J+1,K) + (C1(I+1,J+1,K) - C1(I,J+1,K))*PERX1
      U1U222 = C1(I,J+1,K+1) + (C1(I+1,J+1,K+1) - C1(I,J+1,K+1))*PERX1
      X1DU11 = U1U211 + (U1U212 - U1U211)*PERU2
      X1DU12 = U1U221 + (U1U222 - U1U221)*PERU2
      B(1) = (X1DU12 - X1DU11)/DU1
      X1DU21 = U1U211 + (U1U221 - U1U211)*PERU1
      X1DU22 = U1U212 + (U1U222 - U1U212)*PERU1
      B(2) = (X1DU22 - X1DU21)/DU2
      RETURN
      END
```

```
      SUBROUTINE STEADY(U1STEP,X1SS,X2SS,NIT)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                +
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 3 FOR DETERMINING THE+
C     +         STEADY-STATE VALUES X1SS AND X2SS FOR A STEP INPUT OF  +
C     +         U1STEP.  X1SS AND X2SS ARE FOUND BY SOLVING A SET OF   +
C     +         TWO NONLINEAR ALGEBRAIC EQUATIONS BY NEWTON RAPSON     +
C     +         TECHNIQUE.  NIT IS THE NUMBER OF ITERATIONS (ABOUT 20).+
C     +                                                                +
C     +       THIS SUBROUTINE IS PREPARED FOR SINGLE-INPUT SECOND-ORDER+
C     +         SYSTEMS.  THE COEFFICIENTS C1(NDX1P1,NDX2P1,NGRDU1) AND+
C     +         C2(NDX1P1,NDX2P1,NGRDU1) OF THE MIXED MODEL FORM, WHICH+
C     +         IS TABULAR IN U1 AND POLYNOMIAL IN X1 AND X2, ARE USED.+
C     +                                                                +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.  IF THE  +
C     +         COEFFICIENTS C1 AND C2 WERE FOUND BY USING SYSID2 (SEE +
C     +         APPENDIX A), THE FOLLOWING READ STATEMENTS MAY BE USED +
C     +         IN THE MAIN PROGRAM (WRITTEN BY THE USER):             +
C     +                                                                +
C     +           3 FORMAT(3X,7E11.4)                                  +
C     +             DO 10 IU1=1,NGRDU1                                 +
C     +             READ(5,3)((C1(I,J,IU1),I=1,NDX1P1),J=1,NDX2P1)     +
C     +          10 READ(5,3)((C2(I,J,IU1),I=1,NDX1P1),J=1,NDX2P1)     +
C     +                                                                +
C     +         WHERE, NDX1P1 = ONE PLUS THE DEGREE OF THE POLYNOMIAL  +
C     +         IN X1, NDX2P1 = ONE PLUS THE DEGREE OF THE POLYNOMIAL  +
C     +         IN X2 AND NGRDU1 IS THE NUMBER OF LEVELS INTO WHICH    +
C     +         THE INPUT IS DIVIDED.                                  +
C     +                                                                +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(5,5,11),C2(5,5,11),NDEGX1,NDEGX2,
     1             U1MIN,U1MAX,NGRDU1
      DIMENSION SC1(5,5),SC2(5,5),X1PNM1(5),X2PNM1(5)
C-----
      NDX1P1 = NDEGX1+ 1
      NDX2P1 = NDEGX2 + 1
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1STEP - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1STEP - U1MIN- DU1*IU1)/DU1
      IU1 = IU1 + 1
C-----
      DO 100 J=1,NDX2P1
      DO 100 I=1,NDX1P1
      SC1(I,J) = C1(I,J,IU1) + (C1(I,J,IU1+1) - C1(I,J,IU1))*PERU1
  100 SC2(I,J)= C2(I,J,IU1) + (C2(I,J,IU1+1) - C2(I,J,IU1))*PERU1
C-----
      X1SS = 0.0
      X2SS = 0.0
      DO 200 IT=1,NIT
      X1GESS = X1SS
      X2GESS = X2SS
C-----
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 11
      DO 10 I=1,NDEGX1
   10 X1PNM1(I+1) = X1PNM1(I)*X1GESS
   11 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 21
      DO 20 I=1,NDEGX2
   20 X2PNM1(I+1) = X2PNM1(I)*X2GESS
C----- EVALUATE F1(X1,X2) AND F2(X1,X2)
   21 F1 = 0.0
      F2 = 0.0
      DO 30 J=1,NDX2P1
      DO 30 I=1,NDX1P1
      F1 = F1 + SC1(I,J)*X1PNM1(I)*X2PNM1(J)
   30 F2 = F2 + SC2(I,J)*X1PNM1(I)*X2PNM1(J)
C----- EVALUATE A = DF1/DX1, B = DF1/DX2, C = DF2/DX1, D = DF2/DX2
      A = 0.0
      B = 0.0
      C = 0.0
      D = 0.0
      IF(NDEGX1.EQ.0) GO TO 41
      DO 40 J=1,NDX2P1
      DO 40 I=2,NDX1P1
      EIM1 = I - 1
      A = A + EIM1*SC1(I,J)*X1PNM1(I-1)*X2PNM1(J)
   40 C = C + EIM1*SC2(I,J)*X1PNM1(I-1)*X2PNM1(J)
   41 IF(NDEGX2.EQ.0) GO TO 51
      DO 50 I=1,NDX1P1
      DO 50 J=2,NDX2P1
      EJM1 = J - 1
      B = B + EJM1*SC1(I,J)*X1PNM1(I)*X2PNM1(J-1)
   50 D = D + EJM1*SC2(I,J)*X1PNM1(I)*X2PNM1(J-1)
   51 DELTA = A*D - B*C
      SUB1 = (D*F1 - B*F2)/DELTA
      SUB2 = (A*F2 - C*F1)/DELTA
      X1SS = X1GESS - SUB1
      X2SS = X2GESS - SUB2
  200 CONTINUE
C-----
      RETURN
      END
```

```
      SUBROUTINE LINRIZ(X1OP,X2OP,U1OP,A,B)
C-----
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                              +
C     +       THIS SUBROUTINE WAS USED IN EXAMPLE 3 FOR DETERMINING THE+
C     +         COEFFICIENT MATRICES A AND B OF THE DIFFERENTIAL EQUA- +
C     +         TIONS LINEARIZED ABOUT THE POINT (X1OP,X2OP,U1OP).  IN +
C     +         THIS CASE A IS 2 X 2 AND B IS 2 X 1.                   +
C     +                                                              +
C     +       THIS SUBROUTINE IS PREPARED FOR SINGLE-INPUT SECOND-ORDER+
C     +         SYSTEMS.  THE COEFFICIENTS C1(NDX1P1,NDX2P1,NGRDU1) AND+
C     +         C2(NDX1P1,NDX2P1,NGRDU1) OF THE MIXED MODEL FORM, WHICH+
C     +         IS TABULAR IN U1 AND POLYNOMIAL IN X1 AND X2, ARE USED.+
C     +                                                              +
C     +       NOTE THE QUANTITIES WHICH ARE TRANSFERRED FROM A MAIN    +
C     +         PROGRAM THROUGH THE COMMON (BLOCK) STATEMENT.  IF THE  +
C     +         COEFFICIENTS C1 AND C2 WERE FOUND BY USING SYSID2 (SEE +
C     +         APPENDIX A), THE FOLLOWING READ STATEMENTS MAY BE USED +
C     +         IN THE MAIN PROGRAM (WRITTEN BY THE USER):             +
C     +                                                              +
C     +           3 FORMAT(3X,7E11.4)                                  +
C     +             DO 10 IU1=1,NGRDU1                                 +
C     +             READ(5,3)((C1(I,J,IU1),I=1,NDX1P1),J=1,NDX2P1)     +
C     +          10 READ(5,3)((C2(I,J,IU1),I=1,NDX1P1),J=1,NDX2P1)     +
C     +                                                              +
C     +         WHERE, NDX1P1 = ONE PLUS THE DEGREE OF THE POLYNOMIAL  +
C     +         IN X1, NDX2P1 = ONE PLUS THE DEGREE OF THE POLYNOMIAL  +
C     +         IN X2 AND NGRDU1 IS THE NUMBER OF LEVELS INTO WHICH    +
C     +         THE INPUT IS DIVIDED.                                  +
C     +                                                              +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C-----
      COMMON/BLOCK/C1(5,5,11),C2(5,5,11),NDEGX1,NDEGX2,
     1            U1MIN,U1MAX,NGRDU1
      DIMENSION X1PNM1(5),X2PNM1(5),SC1(5,5),SC2(5,5),X1DTU(2),X2DTU(2)
      DIMENSION A(2,2),B(2)
C-----
      NDX1P1 = NDEGX1 + 1
      NDX2P1 = NDEGX2 + 1
      NGU1M1 = NGRDU1 - 1
      DU1 = (U1MAX - U1MIN)/NGU1M1
      IU1 = (U1OP - U1MIN)/DU1
      IF(IU1.LT.0) IU1 = 0
      IF(IU1.GE.NGU1M1) IU1 = NGU1M1 - 1
      PERU1 = (U1OP - U1MIN - IU1*DU1)/DU1
      IU1 = IU1 + 1
C-----
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 11
      DO 10 I=1,NDEGX1
   10 X1PNM1(I+1) = X1PNM1(I)*X1OP
   11 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 21
      DO 20 I=1,NDEGX2
   20 X2PNM1(I+1) = X2PNM1(I)*X2OP
C-----
   21 U1 = U1MIN + (IU1-1)*DU1 - DU1
      DO 30 NU1=1,2
      U1 = U1 + DU1
      X2DTU1(NU1) = 0.0
      X1DTU(NU1) = 0.0
      X2DTU(NU1) = 0.0
      DO 36 J=1,NDX2P1
      SUM1 = 0.0
      SUM2 = 0.0
      DO 35 I=1,NDX1P1
      SUM1 = SUM1 + C1(I,J,NU1)*X1PNM1(I)
   35 SUM2 = SUM2 + C2(I,J,NU1)*X1PNM1(I
      X1DTU(NU1) = X1DTU(NU1) + SUM1*X2PNM1(J)
   36 X2DTU(NU1) = X2DTU(NU1) + SUM2*X2PNM1(J)
   30 CONTINUE
      B(1) = (X1DTU(2) - X1DTU(1))/DU1
      B(2) = (X2DTU(2) - X2DTU(1))/DU1
C-----
      DO 40 J=1,NDX2P1
      DO 40 I=1,NDX1P1
      SC1(I,J) = C1(I,J,IU1) + (C1(I,J,IU1+1) - C1(I,J,IU1))*PERU1
   40 SC2(I,J) = C2(I,J,IU1) + (C2(I,J,IU1+1) - C2(I,J,IU1))*PERU1
C-----
      X1PNM1(1) = 1.0
      IF(NDEGX1.EQ.0) GO TO 51
      DO 50 I=1,NDEGX1
   50 X1PNM1(I+1) = X1PNM1(I)*X1OP
   51 X2PNM1(1) = 1.0
      IF(NDEGX2.EQ.0) GO TO 61
      DO 60 I=1,NDEGX2
   60 X2PNM1(I+1) = X2PNM1(I)*X2OP
C-----
   61 A(1,1) = 0.0
      A(2,1) = 0.0
      IF(NDEGX1.EQ.0) GO TO 71
      DO 70 J=1,NDX2P1
      DO 70 I=2,NDX1P1
      EIM1 = I - 1
      A(1,1) = A(1,1) + EIN1*SC1(I,J)*X1PNM1(I-1)*X2PNM1(J)
   70 A(2,1) = A(2,1) + EIM1*SC2(I,J)*X1PNM1(I-1)*X2PNM1(J)
   71 A(1,2) = 0.0
      A(2,2) = 0.0
      IF(NDEGX2.EQ.0) RETURN
      DO 80 I=1,NDX1P1
      DO 80 J=2,NDX2P1
      EJM1 = J - 1
      A(1,2) = A(1,2) + EJM1*SC1(I,J)*X1PNM1(I)*X2PNM1(J-1)
   80 A(2,2) = A(2,2) + EJM1*SC2(I,J)*X1PNM1(I)*X2PNM1(J-1)
C-----
      RETURN
      END
```

VITA

Rajamouli Gunda

Candidate for the Degree of

Doctor of Philosophy

Thesis: IDENTIFICATION OF NONLINEAR DYNAMICAL SYSTEMS BY A MODIFIED DIFFERENTIAL APPROXIMATION TECHNIQUE

Major Field: Engineering

Biographical:

Personal Data: Born in Warangal, India, October 10, 1941, the son of Mr. and Mrs. Venkataiah Gunda.

Education: Graduated from A. V. V. High School, Warangal, India, March 1961; received Bachelor of Engineering degree from Osmania University, India in March 1966; received the Master of Science degree from Oklahoma State University, Stillwater, Oklahoma in May, 1969; completed the requirements for the Doctor of Philosophy degree in May, 1971.

Professional Experience: Employed by Regional Engineering College, Warangal, India, as an associate lecturer from July 1966 to August 1967; employed by Oklahoma State University as a graduate research assistant from February 1968 to present.

Professional Organizations: American Society of Mechanical Engineers, Phi Kappa Phi.