SIMS, Robert Alan, 1936-
TIME DOMAIN IDENTIFICATION OF THE
ZEROS OF LINEAR SYSTEMS.

The University of Oklahoma, Ph.D., 1968
Engineering, chemical

University Microfilms, Inc., Ann Arbor, Michigan

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

TIME DOMAIN IDENTIFICATION OF THE ZEROS OF LINEAR SYSTEMS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

ROBERT ALAN SIMS

Norman, Oklahoma

1968

TIME DOMAIN IDENTIFICATION OF THE

ZEROS OF LINEAR SYSTEMS

APPROVED BY

C. M. Skyzaruch

Michael L. McTurnan

F. Canfield

E. J. Block

DISSERTATION COMMITTEE

ABSTRACT

In recent research in the Process Control Laboratory of the University of Oklahoma a time domain technique for the identification of linear systems was formulated, at which time the identification of the system's poles was investigated. The major goal of the present work was to complete the development and verification of the technique with particular reference to determination of the system zeros.

The procedure for determining the system's zeros is based on the analysis of the input and response functions. For an n^{th} order system with n-l zeros, n, linearly-independent input-output records are required. A prior knowledge of the number of zeros, perhaps from theoretical considerations, is helpful but not necessary. The correct number of zeros can be determined from interpretation of the identification results.

The investigation of the technique was conducted utilizing analog and digital computer simulations of second and third order systems. Studies were made to determine the sensitivity of the identification program to many factors which might be encountered in chemical processes. In addition a laboratory heat exchange process was also used to confirm the experimental applicability of the technique.

The results of the identification of the laboratory process were compared to models obtained from frequency response testing and pulse testing. An analog simulation of the identified model was also used to compare the response of the model to the response of the actual process.

This technique has two major advantages over the common frequency domain methods. These advantages are: the system parameters are determined explicitly, and an error propagation analysis enables one to determine the uncertainty of the identified model.

ACKNOWLEDGEMENT

I would like to take this opportunity to thank my advisors, Drs. C. M. Sliepcevich and M. L. McGuire for their cooperation and assistance throughout my many years of association with them.

The number of people from whom I have received aid and encouragement are too numerous to name but their help is no less appreciated.

I also express my appreciation to Dr. Michael Heymann, whose work at the Process Control Laboratory at the University of Oklahoma inspired this study. I also thank him for his advice and assistance during the initial portions of this research.

Thanks are also extended to the National Science Foundation and the Union Carbide Corporation for the financial assistance during this work.

Many thanks are due the secretarial staff of the Flame Dynamics Laboratory for their assistance in the preparation of this manuscript.

Finally, I thank my wife, Sue Ellen, for her patience, understanding, encouragement, and assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure | Page

Figure | Page

TIME DOMAIN IDENTIFICATION OF THE

ZEROS OF LINEAR SYSTEMS

CHAPTER I

INTRODUCTION

In recent years great advances have been made in the techniques for the design and analysis of process controls. Basic to these techniques is a complete description of the dynamic behavior of the process. Theoretical considerations are useful in determining the general topology of the equations describing the process; however, it is frequently impossible to evaluate the actual parameters numerically with any degree of confidence. Since the theoretical approach may not yield a satisfactory dynamic model of the process, it is frequently necessary to rely upon experimental methods for determining a model.

Several analysis and testing techniques for system identification have been developed in the past. Although there is diversity in the mathematical techniques employed for data manipulation and in the types of test signals applied, all of the testing methods have been based on the analysis of input-output relationships. The basic methods may be classified as: 1. sinusoidal (frequency response) testing, 2. pulse testing and Fourier transformation, and 3. random or statistical testing. It is possible to consider all of these identification methods as frequency-domain techniques because at some stage of the analytic process, the data is transformed into the frequency domain. Each of these methods depend on frequency domain representations, usually magnitude and phase lag versus frequency diagrams (Bode plots), for the final plant evaluation.

Direct frequency response testing was adapted from electrical engineering. Although it has been widely used in the past, Hougen (H5) points out that it is generally unsatisfactory in the process industries. The principal virtue of this method is the simplicity with which the data can be interpreted. General drawbacks of the method are: (1) the lengthy tests may cause extreme deviations of the system from its normal mode of operation, (2) many tests over a wide range of frequencies are required regardless of the complexity of the system, and (3) even slight nonlinearities will distort the expected sinusoidal response (S3).

The necessity for evaluating the system model from the Bode plots or other frequency domain representations is a limitation which will be discussed later. Although work has been done on the utilization of direct frequency response testing, usually it has been used for model verification, rather than identification.

Pulse testing and Fourier transformation was first introduced by Hougen (H5). The method was developed to overcome some of the major problems encountered with direct sinusoidal testing. The forcing variable is excited by a single pulse and the input and response are recorded. They are then transformed numerically into the Fourier domain. The frequency response function is obtained as the ratio of the transform of the response to the transform of the input.

This method has a notable advantage over direct sinusoidal testing. The power spectral density of an impulse covers the whole frequency range; therefore a single test is theoretically adequate to evaluate the complete frequency response function, and it can replace a whole series of sinusoidal tests.

In practice the above statement must be qualified. It is physically impossible to generate an impulse, necessitating the use of a pulse with a power spectral density less than one. Dreifke (D2) has demonstrated the sensitivity of the identification to pulse shape and duration. Therefore some experimentation is necessary to determine the proper pulse for any given system. Dreifke, Hougen, and Mesmer (D3) have studied the problems of truncation error due to the use of limited (finite) record segments, i.e., use of response records which end before returning to steady-state. The effects of this truncation error and of improper input pulses appear in the resulting frequency response function as a "cut-off frequency". Cycling occurs at frequencies above the cut-off frequency and the scatter of the results becomes excessive, indicating that the response function is unreliable. To eliminate the problem of truncation one might record the response function until steady state is "practically" reached; however, in chemical processes where the time constants are long, the possible drifts in steady state operating conditions may offset the decrease in truncation error.

Noise will effectively increase the truncation error. In the later stages of response curves, where the amplitudes are low, the effect of noise may completely overshadow the signal, and the frequency response function is unreliable.

The limitations due to noise have been incorporated into another identification technique, random testing followed by statistical analysis of the response curves. Since this present work does not involve this technique the reader is referred to Gallier (G1) for a further discussion of random testing.

The methods discussed have an inherent disadvantage in the representation of the results. This representation is usually in the form of a magnitude versus frequency plot and phase lag versus frequency plot, commonly called the Bode plot. From the Bode plot it may be very difficult, if not impossible, to evaluate numerically the parameters of complex systems (system order greater than one).

Generally, the useful range of the magnitude ratio plot does not exceed one to two decades on the logarithmic frequency scale. Thus the identifiable poles and zeros are limited to this range. The identifiable order of the system is also limited to this range. Since the phase lag versus frequency is usually less reliable than the magnitude plot, it offers little additional information.

Heymann (H2,H3) has proposed a technique for linear system identification based exclusively on time domain analysis. This technique circumvents the problem of obtaining the system parameters from frequency response representations. The system parameters are explicitly determined; in addition, error bounds associated with each of the system parameters are ascertained. However, Heymann has formulated and verified only a portion of the technique, the pole identification. It is the major goal of this work to complete the formulation of the technique and to demonstrate its usefulness for system identification.

The completion of the time domain identification technique is accomplished by implementation of the theory describing the determination of the system zeros. An error propagation analysis is also developed to predict the reliability of the identified results.

The test for the applicability of the technique is conducted in two parts. Computer studies are used to determine the sensitivity of the method to the many factors which are encountered in actual chemical processes. The technique is then used to identify a laboratory process, and the results of the identification are compared to the results obtained from direct frequency response testing and pulse testing.

CHAPTER II

THEORY

A synoptic review of the background theory necessary for the zero identification problem is presented here. For a complete analysis of the theory of the time domain identification technique the reader is referred to Heymann (H2,H3).

Any one of the outputs (i^{th}) of a general linear system of the form

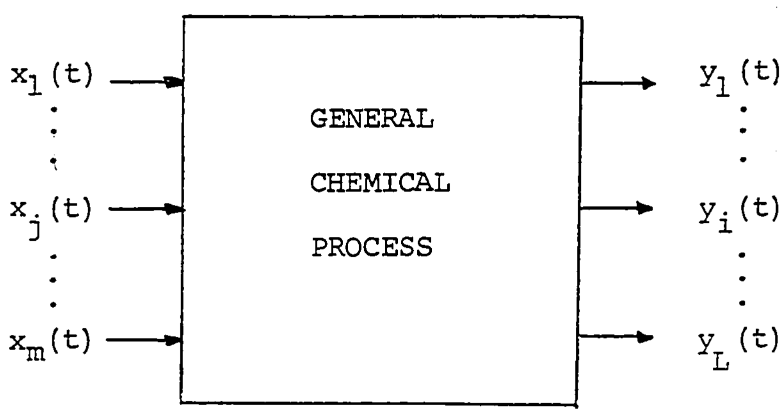


FIGURE 2-1 General Chemical Process

can be described in terms of an n^{th}- order linear ordinary differential equation of the form

$$\frac{d^n y_i(t)}{dt^n} + b_1(t)\,\frac{d^{n-1} y_i(t)}{dt^{n-1}} + \ldots + b_n(t) y_i(t) =$$

$$\sum_{j=1}^{m} g_{j1}(t)\frac{d^{n-1} x_j(t)}{dt^{n-1}} + \ldots + g_{jn}(t) x_j(t)\,. \tag{2-1}$$

The system can also be expressed in operator notation as

$$L^n[y_i(t)] = \sum_{j=1}^{m} M_j^{(k_j)}[x_j(t)] \tag{2-2}$$

where $L^n[\ ]$ and $M_j^{(k_j)}[\ ]$ are linear differential operators, with time dependent parameters, of order n and k_j respectively, where $k_j \leq n-1$. If the right hand side of Equation (2-2) is lumped into a general time function f(t), Equation (2-1) transforms into

$$L^n[y_i(t)] = f(t)\,. \tag{2-3}$$

The response of the system given by Equation (2-3) can also be expressed in terms of the system weighting function, $W(t,\lambda)$, i.e. the impulse response, by the following equation:

$$y_i(t) = \int_{-\infty}^{t} W(t,\lambda)\ f(\lambda)\ d\lambda \tag{2-4}$$

If $y_i(t) = 0$ prior to $t = 0$, i.e. the system is operating at steady state, then

$$y_i(t) = \int_0^t W(t,\lambda)\ f(\lambda)\ d\lambda \tag{2-5}$$

Considering Equations (2-2) and (2-3), Equation (2-5) can be rewritten as

$$\begin{aligned} y_i(t) &= \int_0^t W(t,\lambda) \sum_{j=1}^{m} M_j^{(k_j)}[x_j(\lambda)]d\lambda \\ &= \sum_{j=1}^{m} \int_0^t W(t,\lambda)\ M_j^{(k_j)}[x_j(\lambda)]d\lambda \end{aligned} \tag{2-6}$$

Equation (2-6) can be expressed in an equivalent, but more convenient, form as

$$y_i(t) = \sum_{j=1}^{m} \int_0^t W_j(t,\lambda)\ x_j(\lambda)\ d\lambda \tag{2-7}$$

where the weighting function* $W_j(t,\lambda)$ is related to the weighting function $W(t,\lambda)$ in Equation (2-6) by the expression

$$\begin{aligned} W_j(t,\lambda) = (-1)^{k_j} \frac{d^{k_j}}{dt^{k_j}}[g_{j1}(t)W(t,\lambda)]+ \dots + \\ g_{jk_j}(t)W(t,\lambda). \end{aligned} \tag{2-8}$$

---

*The weighting function $W_j(t,\lambda)$, which relates a particular input variable $x_j(t)$ to the output $y_i(t)$, will be called the Particular Weighting Function.

In operator form, Equation (2-8) can be written as

$$W_j(t,\lambda) = M_j^{(k_j)*}[W(t,\lambda)] \tag{2-9}$$

where $M_j^{(k_j)*}[\ ]$ is the adjoint operator of $M_j^{(k_j)}[\ ]$.

Equation (2-7) relates the i^{th} response to all of the input variables. If all of the input variables except $x_e(t)$ are kept zero, then Equation (2-7) can be written as:

$$y_i(t) = \int_0^t W_e(t,\lambda)x_e(\lambda)\ d\lambda \tag{2-10}$$

where

$$W_e(t,\lambda) = (-1)^{k_e}\frac{d^{k_e}}{dt^{k_e}}[g_{e1}(t)W(t,\lambda)] + \dots + g_{ek_e}(t)W(t,\lambda). \tag{2-11}$$

$W(t,\lambda)$ is determined by the methods described by Heymann [H2, H3]; thus, in order to determine $W_e(t,\lambda)$ the operator $M_e^{(k_e)}[\ ]$ remains to be evaluated.

If the process is stationary (constant parameter), the evaluation of the operator $M_e^{(k_e)}[\ ]$ is simplified. Equation (2-11) can be written as:

$$W_e(t,\lambda) = (-1)^{k_e}\frac{d^{k_e}}{dt^{k_e}}[g_{e1}W(t-\lambda)] + \dots + g_{ek_e}\ W(t-\lambda) \tag{2-12}$$

The coefficients g_{ej} can be determined numerically by the following procedure: the system is forced k_e times with linearly independent forcing functions, which are recorded together with their corresponding responses. By numerical solution of k_e relations of the type,

$$y_i(t) = (-1)^{k_e} g_{e1} \int_0^t \frac{d^{k_e}}{dt^{k_e}} W(t-\lambda) x_e(\lambda) d\lambda + \dots + g_{ek_e} \int_0^t W(t-\lambda) x_e(\lambda) d\lambda \qquad (2\text{-}13)$$

the coefficients g_{ej} are evaluated.

For simplicity in the following development, only one of the input-output relationships will be considered. Equation (2-13) then reduces to

$$y(t) = (-1)^k g_1 \int_0^t \frac{d^k}{dt^k} W(t-\lambda) x(\lambda) d\lambda + \dots + g_k \int_0^t W(t-\lambda) x(\lambda) d\lambda \qquad (2\text{-}14)$$

The computation of these integrals is accomplished by first calculating the coefficients of the weighting function and k-1 of its derivatives.

The homogeneous transfer function as determined by the pole identification is of the form

$$H(s) = \frac{1}{(s+\rho_1) \dots (s+\rho_n)}. \qquad (2\text{-}15)$$

The weighting function is then

$$W(t-\lambda) = a_1 e^{-\rho_i(t-\lambda)} + \ldots + a_n e^{-\rho_n(t-\lambda)} \tag{2-16}$$

$$= \sum_{i=1}^{n} a_i e^{-\rho_i(t-\lambda)}$$

where

$$a_i = \frac{1}{(\rho_1-\rho_i) \;\cdots\; (\rho_{i-1}-\rho_i)(\rho_{i+1}-\rho_i)\cdots\;(\rho_n-\rho_i)} . \tag{2-17}$$

The j^{th} derivation of the weighting function is then given by

$$\frac{d^j W(t-\lambda)}{dt^j} = (-1)^j (a_1\rho_1^j e^{-\rho_1(t-\lambda)} + \ldots + a_n\rho_n^j e^{-\rho_n(t-\lambda)}) \tag{2-18}$$

$$= (-1)^j \sum_{i=1}^{n} a_i\rho_i^j e^{-\rho_1(t-\lambda)} .$$

Equation (2-14) can now be written as a general term

$$y(t) = \sum_{j=0}^{k-1} (-1)^j g_{k-j} \int_0^t \sum_{i=1}^{n} (-1)^j a_i\rho_i^j e^{-\rho_i(t-\lambda)} x(\lambda)\, d\lambda \tag{2-19}$$

$$= \sum_{j=0}^{k-1} g_{k-j} \int_0^t \sum_{i=1}^{n} a_i\rho_i^j e^{-\rho_i(t-\lambda)} x(\lambda)\, d\lambda$$

Consider Equation (2-19) as composed of two parts, the integral and a coefficient. The integral for any i is the same for all of the coefficients. Therefore, only k integrations are required where k^2 integrations would be required by a straightforward solution of Equation (2-14).

To determine the unknowns g_{k-j}, k simultaneous equations of the type of Equation (2-19) are necessary. These equations are generated by varying $x(\lambda)$ (the forcing function). The only constraint upon these $x(\lambda)$'s is that they must all be linearly independent.

Theoretically the solution of the k simultaneous equations at any time T is sufficient to determine the particular weighting function. In practice, however, the uncertainty of the data portends the danger of relying upon the one solution. It is therefore desirable to obtain a solution at many points of time and to have some measure of the reliability of the results at these various points.

The error propagation analysis to be discussed in the next chapter provides this measure of the reliability.

CHAPTER III

ERROR PROPAGATION ANALYSIS

Two types of error exist in a numerical computation. One is the error associated with the uncertainty of the data, which is primarily responsible for the uncertainty of the results. It is the purpose of the error propagation analysis to predict when the interaction of the uncertainties of the data have a minimum effect upon the final identification results. The other source of error is computer round-off or truncation error. This error effects the numerical value of the computation, but it has no appreciable effect upon the uncertainty of the results. This type of error is discussed more fully in the computer studies section on the effect of the number of significant figures in the data.

If a true value were known for the results of a computation, then a true error could be calculated. Whenever the true result is not known it is necessary to predict an error limit for each stage of the computation. This error limit may be either an extreme or an expected error. The present work is concerned with the latter approach.

Before proceeding with the details of the error propagations analysis, it may be helpful to review the sources of error and the computational steps involved.

The sources of error in the data are:

1. The accuracy of the homogeneous weighting function, i.e. the poles of the system.
2. The reliability and accuracy of the input and response functions.

Several factors enter into the reliability of the input and response functions. They are: values of the steady-state, measurement error, and noise. In this work it is assumed that any noise present is stationary and ergodic random noise. The identification values will therefore have an equal chance of being distributed above or below their mean value and essentially no drift will occur because of the integration of the input function. It is further assumed that the steady-state and measurement error can be lumped together into a percentage error term which is relatively small.

The computational steps involved in the identification are:

1. Calculation of the integral coefficients of g_{ej} in the operator, $M_e^{(k_e)}[\ ]$.
2. Solution of the simultaneous algebraic equations thus obtained.

For the purposes of the error analysis it is assumed that the calculation of the coefficients is done in one step. The solution of the simultaneous equations is performed in two steps, inversion of the coefficient matrix and premultiplication of the response vector by the coefficient inverse matrix.

Expected Error in the Coefficients of g_{ei}

If F is a function of x,y, and z, the linear term in the Taylor series can be used to express the effect of a small error in x, y, and z. Thus, if Δx is the error in x and Δy is the error in y and Δz is the error in z, the expected error in F is ΔF. These errors may be related by

$$\Delta F = \frac{\partial F}{\partial x}\Delta x + \frac{\partial F}{\partial y}\Delta y + \frac{\partial F}{\partial z}\Delta z \tag{3-1}$$

Applying this relationship then to the integral part of Equation (2-19)

$$G_k = \sum_{i=1}^{n} \int_0^t a_i e^{-\rho_i(t-\lambda)} x(\lambda)\, d\lambda \tag{3-2}$$

yields

$$\Delta G_k = \frac{\partial G_k}{\partial \rho_1}\Delta\rho_1 + \ldots + \frac{\partial G_k}{\partial \rho_n}\Delta\rho_n + \frac{\partial G_k}{\partial x}\Delta x \tag{3-3}$$

where $\Delta\rho_1, \ldots, \Delta\rho_n$ are the expected errors in the poles of the homogeneous weighting function and Δx is the expected error in the input. The coefficient is given by Equation (2-17) as

$$a_i = \frac{1}{(\rho_1-\rho_i)\ldots(\rho_{i-1}-\rho_i)(\rho_{i+1}+\rho_i)\ldots(\rho_n-\rho_i)} \tag{2-17}$$

The various partial derivatives are required to evaluate the expected error ΔG_k. Differentation of the general term of Equation (3-2)

$$G_k(i) = \int_0^t a_i e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda \tag{3-4}$$

with respect to ρ_j yields two results:

when $j \neq i$

$$\frac{\partial G_k(i)}{\partial \rho_j} = \frac{\partial a_i}{\partial \rho_j} \int_0^t e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda = \frac{\partial a_i}{\partial \rho_j} \frac{G_k(i)}{a_i} \tag{3-5}$$

and when $j = i$

$$\frac{\partial G_k(i)}{\partial \rho_i} = \frac{\partial a_i}{\partial \rho_i} \int_0^t e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda + a_i \int_0^t (t-\lambda)\, e^{-\rho_i (t-\lambda)} x(\lambda) d\lambda$$

(3-6)

$$= \frac{\partial a_i}{\partial \rho_i} \frac{G_k(i)}{a_i} + t\, G_k(i) - a_i \int_0^t \lambda e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda .$$

From the definition of a_i, Equation (2-17), the various partials of a_i can be obtained. When $i \neq j$ only one term of a_i contains ρ_j so the derivative is straightforward and after rearrangement is

$$\frac{\partial a_i}{\partial \rho_j} = \frac{a_i}{\rho_j - \rho_i} . \tag{3-7}$$

When $i = j$, ρ_i is contained in every term. It is necessary to differentiate the function as a group of successive products. After differentiation and rearrangement the result is

$$\frac{\partial a_i}{\partial \rho_i} = -\, a_i \left(\frac{1}{\rho_1 - \rho_i} + \cdots + \frac{1}{\rho_{i-1} - \rho_i} + \frac{1}{\rho_{i+1} - \rho_i} + \cdots + \frac{1}{\rho_n - \rho_i} \right) \tag{3-8}$$

For simplicity let

$$da_i = \frac{1}{\rho_1 - \rho_i} + \cdots + \frac{1}{\rho_n - \rho_i} \tag{3-9}$$

and Equation (3-8) becomes

$$\frac{\partial a_i}{\partial \rho_i} = -\, a_i \; d\, a_i . \tag{3-10}$$

The last term of Equation (3-6)

$$H_k(i) = a_i \int_0^t \lambda e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda \tag{3-11}$$

can be expressed in terms of $G_k(i)$. When the right hand side of Equation (3-11) is integrated by parts, the resultant equation, with the use of Equation (3-4), becomes

$$H_k(i) = t\, G_k(i) - \int_0^t G_k(i)\, d\lambda . \tag{3-12}$$

After substitution of Equations (2-7), (3-10), and (3-12) into Equation (3-5) and (3-6), the resultant partial derivatives are

$$\frac{\partial G_k(i)}{\partial \rho_j} = \frac{G_k(i)}{\rho_j - \rho_i} \qquad \text{(a)}$$

$$\frac{\partial G_k(i)}{\partial \rho_i} = - G_k(i)\, da_i + \int_0^t G_k(i)\, d\lambda . \qquad \text{(b)} \qquad (3\text{-}13)$$

The next term of the operator $M^{(k)}[\ ]$ is

$$G_{k-1} = \int_0^t \frac{d\, w(t-\lambda)}{dt}\, x(\lambda)\, d\lambda . \qquad (3\text{-}14)$$

The form of the general term of this expression can be expressed by

$$\begin{aligned} G_{k-1} &= \int_0^t a_i \frac{d}{dt} e^{-\rho_i (t-\lambda)}\, x(\lambda)\, d\lambda \\ &= -a_i \rho_i \int_0^t e^{-\rho_i (t-\lambda)}\, x(\lambda)\, d\lambda \qquad (3\text{-}15) \\ &= -\rho_i\, G_k(i) \end{aligned}$$

The partial derivatives of Equation (3-15) are

$$\frac{\partial G_{k-1}(i)}{\partial \rho_j} = - \frac{\rho_i}{\rho_j - \rho_i}\, G_k(i), \qquad \text{(a)}$$

$$\frac{\partial G_{k-1}(i)}{\partial \rho_i} = - \rho_i \frac{\partial G_k(i)}{\partial \rho_i} - G_k(i). \qquad \text{(b)} \qquad (3\text{-}16)$$

This procedure can be carried out for all of the terms of the operator $M^{(k)}[\ ]$, and general expressions can be developed to describe the coefficients and their partial derivatives. These general expressions are

$$G_n(i) = (-1)^{k-n} \rho_i^{\,k-n} G_k(i) \qquad (a)$$

$$\frac{\partial G_n(i)}{\partial \rho_j} \Delta\rho_j = (-1)^{k-n} \rho_i^{\,k-n} \frac{\partial G_k(i)}{\partial \rho_j} \Delta\rho_j \qquad (b) \qquad (3\text{-}17)$$

$$\frac{\partial G_n(i)}{\partial \rho_i} \Delta\rho_i = (-1)\, \rho_i^{\,k-n-1} - \left[\rho_i \frac{\partial G_k(i)}{\partial \rho_i} + (k-n) G_k(i)\right] \Delta\rho_i . \qquad (c)$$

All that remains to describe completely the error propagation in the computation of the integral coefficients is to define the last term in Equation (3-3), $(\partial G/\partial x)\Delta x$. From the general term of the describing equation for G_k

$$G_k(i) = \int_0^t a_i\, e^{-\rho_i (t-\lambda)} x(\lambda)\, d\lambda \qquad (3\text{-}4)$$

the error caused by a small change in $x(\lambda)$ can be determined. This error is

$$\frac{\partial G_k(i)}{\partial x} \Delta x = \int_0^t a_i e^{-\rho_i (t-\lambda)} \Delta x(\lambda)\, d\lambda \qquad (3\text{-}18)$$

The errors caused by the input for the other terms of the operator $M^{(k)}[\ ]$ can be found in a similar manner. The general expression is

$$\frac{\partial G_n(i)}{\partial x} \Delta x = a_i (-1)^{k-n} \rho_i^{k-n} \int_0^t e^{-\rho_i (t-\lambda)} \Delta x(\lambda) \, d\lambda . \tag{3-19}$$

$G_n(i)$ is calculated from the general term of the weighting function and is obtained by summing all of the terms of the weighting function:

$$G_n = \sum_{i=1}^{k} G_n(i) \qquad \text{(a)}$$

(3-20)

$$\frac{\partial G_n}{\partial \rho_j} \Delta\rho_j = \sum_{i=1}^{k} \frac{\partial G_n(i)}{\partial \rho_j} \Delta\rho_j . \qquad \text{(b)}$$

Substituting Equations (3-19) and (3-20) into Equation (3-3), the total error in G_n is obtained

$$\Delta G_n = \sum_{j=1}^{k} \sum_{i=1}^{k} \frac{\partial G_n(i)}{\partial \rho_j} \Delta\rho_j + \sum_{i=1}^{k} \frac{\partial G_n(i)}{\partial x} \Delta x. \tag{3-21}$$

As n goes from 1 to k, k-element row vector is generated which corresponds to the row vector for the integral coefficients G_n. When k inputs have been computed in this manner, a k x k matrix of error values is obtained corresponding to the coefficient matrix.

Expected Error in the System Zeros

The next stage of the computation is the inversion of the coefficient matrix and premultiplication of the response vector by the inverse. In matrix notation

$$g = G^{-1}\,Y \tag{3-22}$$

where G is the coefficient matrix, g is the solution vector and Y is the response vector. The desired error is the error in the g vector. Again using the total differential

$$\Delta g = \frac{\partial g}{\partial G^{-1}}\Delta G^{-1} + \frac{\partial g}{\partial Y}\,\Delta Y \tag{3-23}$$

a relationship for the error is obtained. The partial derivative of g with respect to G^{-1} and Y are easy to find as they are independent functions. They are

$$\frac{\partial g}{\partial G^{-1}} = Y \qquad \text{(a)}$$

$$\frac{\partial g}{\partial Y} = G^{-1}. \qquad \text{(b)} \tag{3-24}$$

ΔG^{-1} is harder to determine as it is derived from the error matrix obtained at stage one of the calculation. It is not simply the inverse of this matrix. The easiest way to determine ΔG^{-1} is a brute force method

$$\Delta G^{-1} = (G + \Delta G)^{-1} - G^{-1} \tag{3-25}$$

but it does give a good estimate of the error limits in the inverse. ΔY is estimated from the response curves.

CHAPTER IV

COMPUTER STUDIES

Many tests of the identification technique were conducted using digital and analog computer generated data. This technique was used to provide complete control of all conditions which might affect the identification. In this manner it was possible to study the specific effect caused by a potential identification problem. Some of the conditions of interest were: errors in the poles and steady state values, transportation delays, non-linearities, and noise. Other tests were run to determine the best forcing functions and also to determine how the relative location of the poles and zeros affected the identification.

The digitally-generated data were obtained as the solution of the following general transfer function

$$Y(s) = \frac{G\ (As+1)\ (Bs+1)}{(Cs+1)\ (Ds+1)\ (Es+1)} \qquad (4\text{-}1)$$

with three different forcing functions

$$X(s) = \frac{K}{s},\ \frac{L}{s^2},\ \frac{M}{s^2+\omega^2}\ . \qquad (4\text{-}2)$$

In the time domain these are respectively: a step of amplitude, K; a ramp of slope, L; and a sinusoid of amplitude, M, and angular frequency, ω.

The time domain expressions including transport delay and steady state were programmed as:

$$Y_{step}(t+\tau) + Y_{ss} = X_{ss} + KG(1-a_1e^{-t/C}-a_2e^{-t/D}-a_3e^{-t/E}) \quad \text{(a)}$$

$$\begin{aligned} Y_{ramp}(t+\tau) + Y_{ss} = X_{ss} &+ LG(A+B-C-D-E+t+Ca_1e^{-t/C} \\ &+ Da_2e^{-t/D}+Ea_3e^{-t/E}) \quad \text{(b)} \end{aligned} \qquad (4\text{-}3)$$

$$\begin{aligned} Y_{sine}(t+\tau) + Y_{ss} = X_{ss} &+ MG[-(b_1+b_2+b_3)\cos\omega t \\ &+ (1-\omega b_1-\omega b_2-\omega b_3)\sin\omega t \\ &+ b_1e^{-t/C}+b_2e^{-t/D} \\ &+ b_3e^{-t/E}] \quad \text{(c)} \end{aligned}$$

where

$$a_1 = (A-C)\ (B-C)/(D-C)\ (E-C)$$
$$a_2 = (A-D)\ (B-D)/(C-D)\ (E-D)$$
$$a_3 = (A-E)\ (B-E)/(C-E)\ (D-E)$$
$$b_1 = C\omega a_1/(1+C^2\omega^2)$$
$$b_2 = D\omega a_2/(1+D^2\omega^2)$$
$$b_3 = E\omega a_2/(1+E^2\omega^2).$$

Effect of Forcing Functions

The first series of tests were conducted to determine the best range of frequencies for a forcing sine wave. Figure 4-1 shows some of the results of the gain identification obtained when the system

$$\frac{Y}{X}(s) = \frac{80(25s+1)(5s+1)}{(10s+1)(20s+1)(50s+1)} \qquad (4\text{-}4)$$

was forced with a step, a ramp, and a series of sine waves, ranging from 0.0001 radians/second to 1 radian/second.

As can be seen in Figure 4-1, the results were not very good at the two extremes (0.0001 radians/second and 1 radian/second). The results did, however, tend to converge to the correct answer. The best results were obtained when the forcing frequencies were between 0.01 and 0.2 radians/second. As this condition was also the case when two or three sine functions of different frequencies were used for the input functions, a general rule of thumb is obtained. The angular frequency of the forcing sine functions should be between one half the lowest natural frequency and twice the highest natural frequency.

The identification of the zeros is affected in the same manner as the gain. Part B of Figure 4-1 shows the identification results of the zeros.

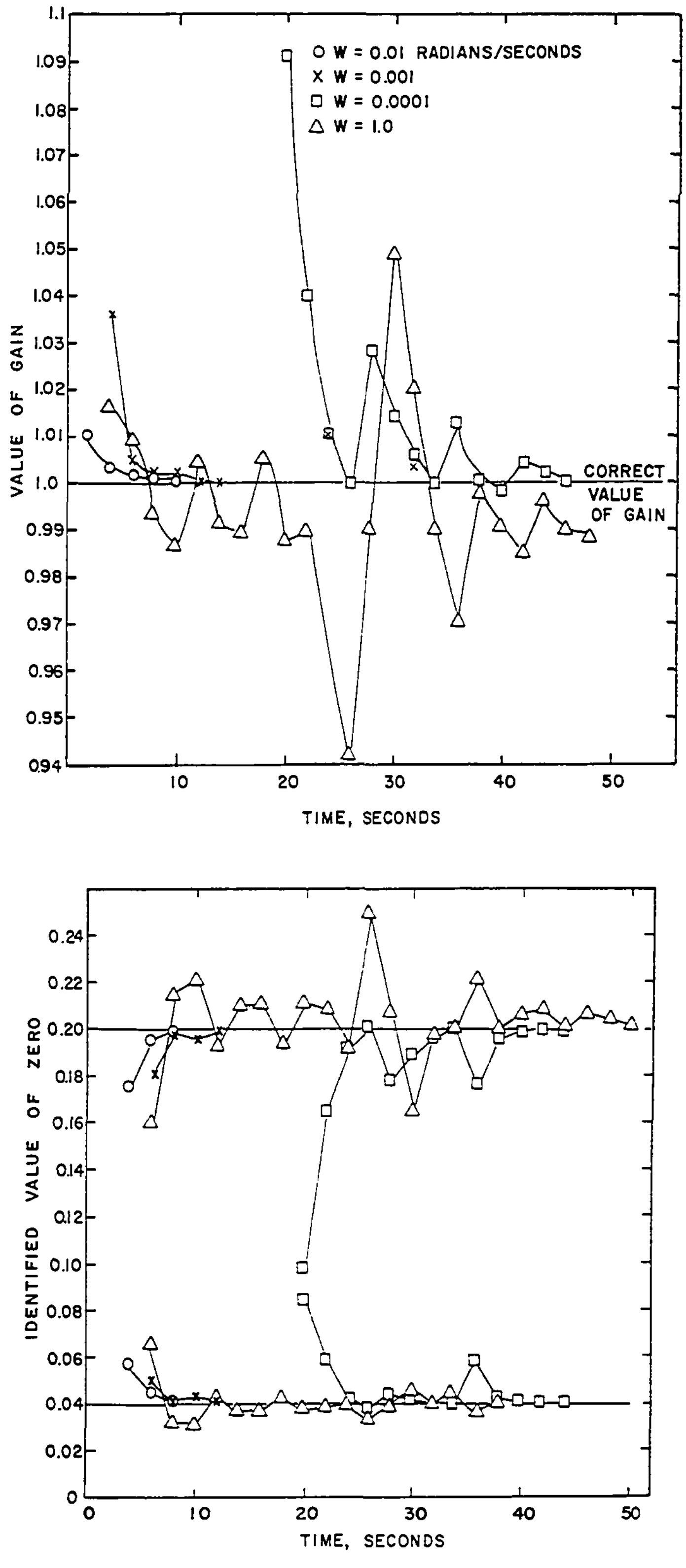


Figure 4-1. Effect of Frequency of Forcing Sine Function upon the Identification.

Effect of the Relative Location of Poles and Zeros

The relative natural frequencies of the poles and zeros had very little effect upon the identification. Tests were made on third order systems where factors of as much as 250 existed between the largest and smallest poles, with the zero ranging from less than to greater than the poles. The rate of convergence to the correct answer was the only effect upon the identification. The convergence was the slowest when the zeros were less than the poles (Table 4-1).

TABLE 4-1

EFFECT OF THE RELATIVE LOCATION OF THE POLES AND ZEROS

Pole 1	Pole 2	Pole 3	Zero 1	Zero 2	Time to Converge to Less Than 1-Percent Error (Sec)
0.1	0.05	0.02	0.04	0.01	18
0.1	0.05	0.005	0.2	0.01	12
0.5	0.05	0.02	0.2	0.04	12
0.1	0.05	0.02	0.1	0.04	8
0.1	0.05	0.02	0.2	0.05	8
0.1	0.05	0.02	0.2	0.02	10
0.1	0.02083	0.02	0.2	0.02	10
0.1	0.02083	0.02	0.02083	0.02	16
0.1	0.05	0.02	0.2	0.04	8
0.5	0.02	0.002	0.2	0.04	24

Effect of the Order of the System

To determine the effect of an incorrect estimate of the number of poles of a system, i.e. the order of the system, upon the zero identification, a series of identification tests was conducted with poles deleted. In addition a series of tests with poles added was carried out. When the assumed order of the system was greater than the actual order, no problems arose since zeros were determined which cancelled the extra poles.

When the assumed system order was less than the actual order, the quality of the identification depended upon the relative values of the poles and zeros. In all cases the gain and zeros approach their values asymptotically; the rate of convergence and initial displacement again depend upon the relative values of the poles and zeros.

Figure 4-2 shows the results of the gain identification when the system is correctly identified as a third order system,

$$Y(s) = \frac{gX(s)}{(s+0.5)(s+0.02)(s+0.002)}, \qquad (4\text{-}5)$$

as a second order system

$$Y(s) = \frac{gX(s)}{(s+0.02)(s+0.002)}, \qquad (4\text{-}6)$$

and as a first order system

$$Y(s) = \frac{gX(s)}{(s+0.002)}. \qquad (4\text{-}7)$$

When the correct poles are used the result of the identification is within 0.2 percent of the correct value of 1.0 by the time of two seconds. However, when identified as a second order system the value of the gain increases with time, eventually becoming asymptotic to 2.0, which is the correct gain divided by the magnitude of the neglected pole (1/0.5). The DC gain of the system of Equation (4-5) is determined by replacing s by $j\omega$ and setting the frequency ω equal to zero, DC gain = 1/(0.5x0.02x0.002) = 50000. To match this DC gain, the gain, g, of the system of Equation (4-6) must be equal to the neglected term (1/0.5). When identified as a first order system, the initial error is very large, but the gain may be approaching the theoretical value of the gain of the third order system divided by the neglected poles, 1/(0.5)(0.02) = 100.

Figure 4-3 gives the results of the same type of tests when the actual system contains one zero. When identified as a second order system the zero approaches the correct value of 0.01 while the gain approaches the value of the correct gain divided by the omitted pole, 2.0. When identified as a first order system g/s+0.002, the gain levels off at about 1.6 and does not approach the theoretical value of 1.0. No explanation has been found for this behavior.

Figure 4-4 shows the results of these tests on a system which contains two zeros. When one pole is neglected, a zero is also omitted from necessity. The omitted zero is the highest frequency zero (s+0.2). The zero at 0.04 is

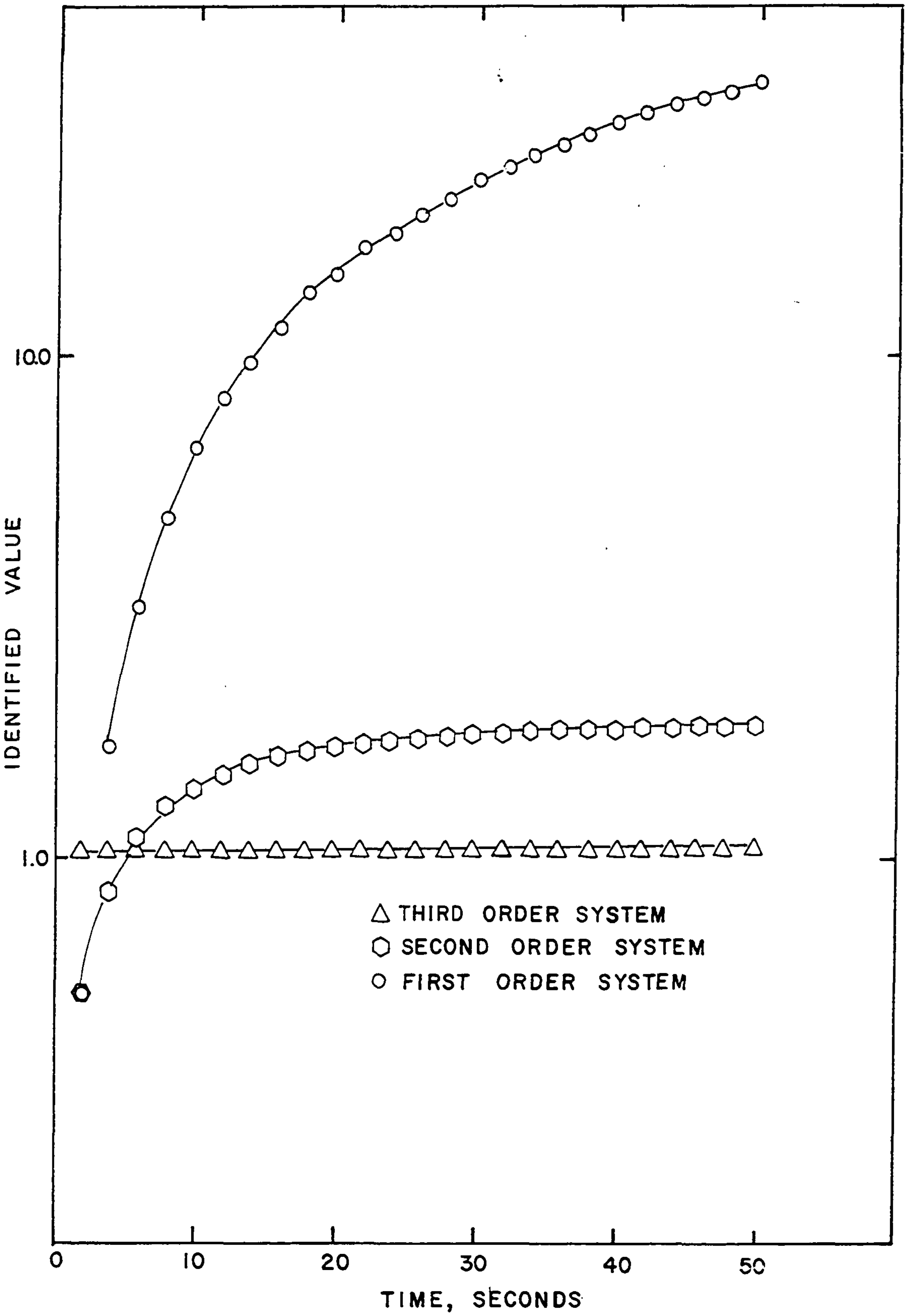


Figure 4-2. Approximate Identification of a Three Pole System Using Lesser Order Models.

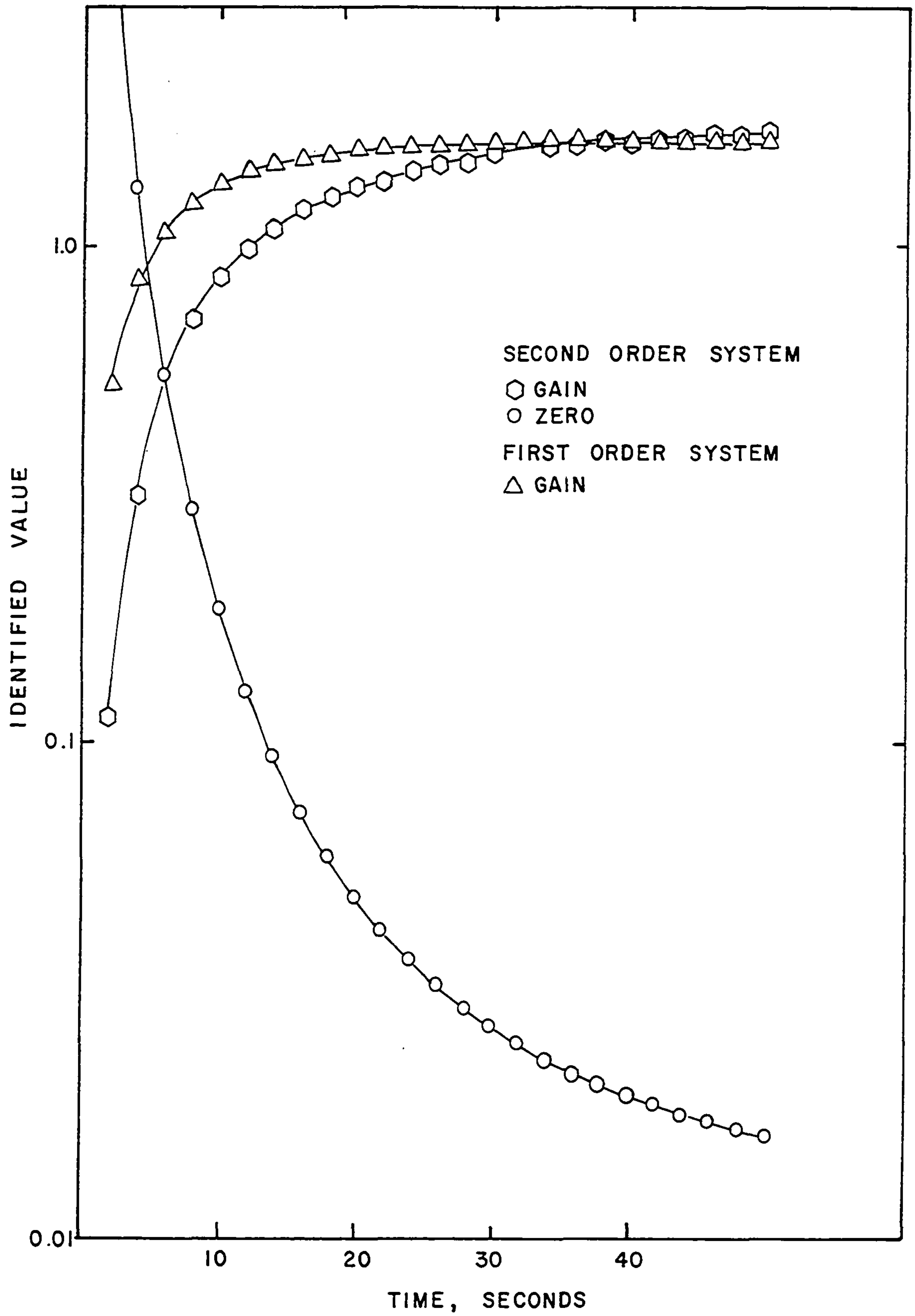


Figure 4-3. Approximate Identification of a One Zero-Three Pole System Using Lesser Order Models.

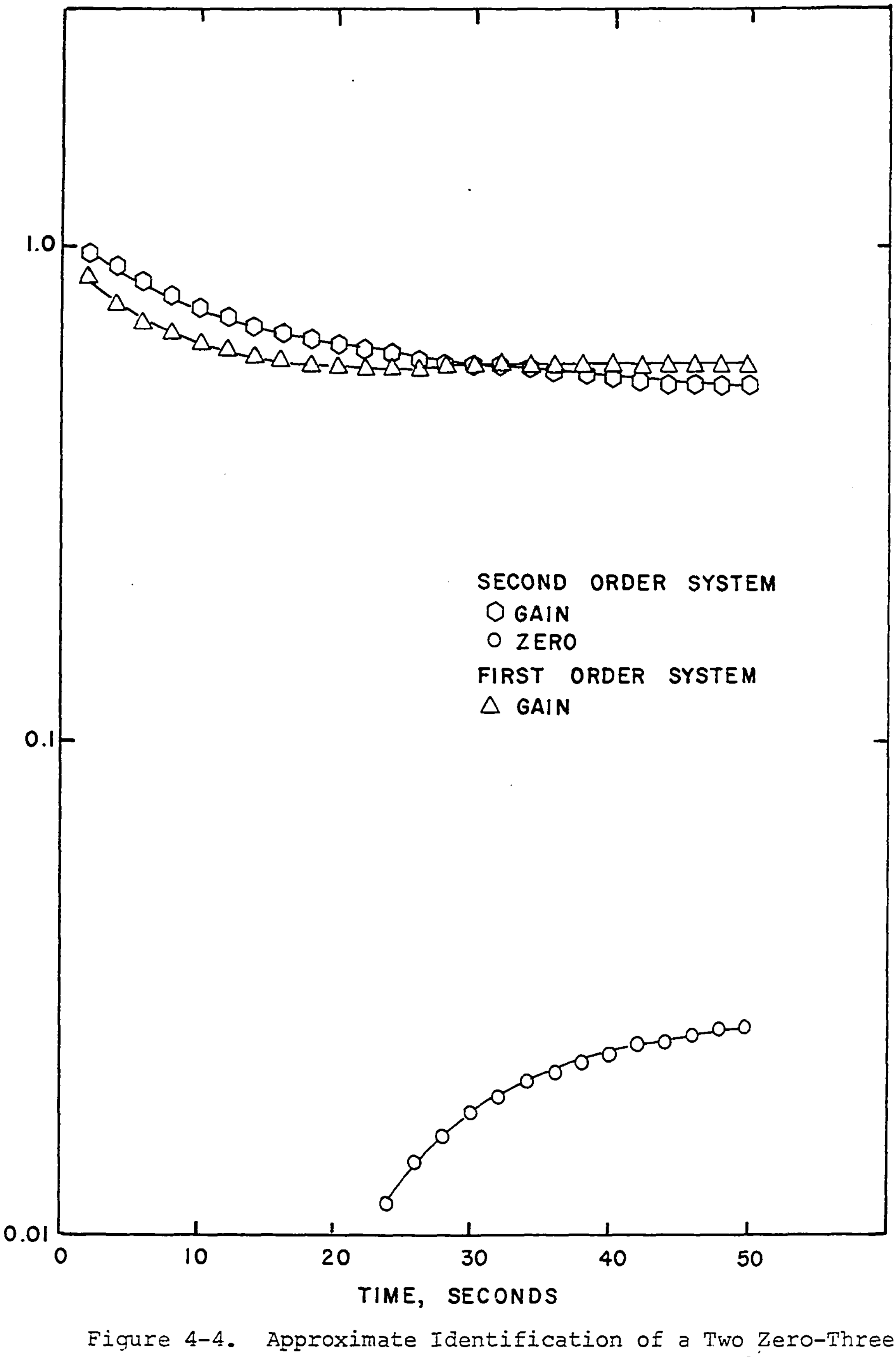


Figure 4-4. Approximate Identification of a Two Zero-Three Pole System Using Lesser Order Models.

approached asymptotically. The value of the gain approaches the correct gain times the omitted zero divided by the neglected pole 1.0 x 0.2/0.5 = 0.4. When the technique is applied assuming a first order system g/s + 0.002, the gain appears to level out at a value of 0.57. The theoretical result is 0.2 x 0.04/0.5 x 0.02 = 0.8.

To determine the effect of the neglected pole upon the identified gain in relation to the magnitude of the neglected pole, a series of three tests of third order systems was run using two poles ($\rho_1 = -0.02$, $\rho_2 = -0.01$) in the identification. The third pole, which was omitted in the identification, was varied from 0.1 to 0.50. The results of these tests are given in Figure 4-5. To show the relative rates of convergence, the scales have been normalized. If the result of a correct third order identification were also shown on Figure 4-5, it would be a straight line at 1.0. When the system was identified correctly, the gain was in error only 0.2 percent at time t = 2 in the worst case.

The preceding tests were conducted upon a system which had only a gain and no zeros. When a zero is added, convergence is also dependent upon the natural frequency of the zero relative to the omitted pole. The results of the gain identification of a system when the neglected pole had a value four times the magnitude of the zero and one where the neglected pole was only twice the magnitude of the zero are also shown in Figure 4-5. As can be seen, the identification

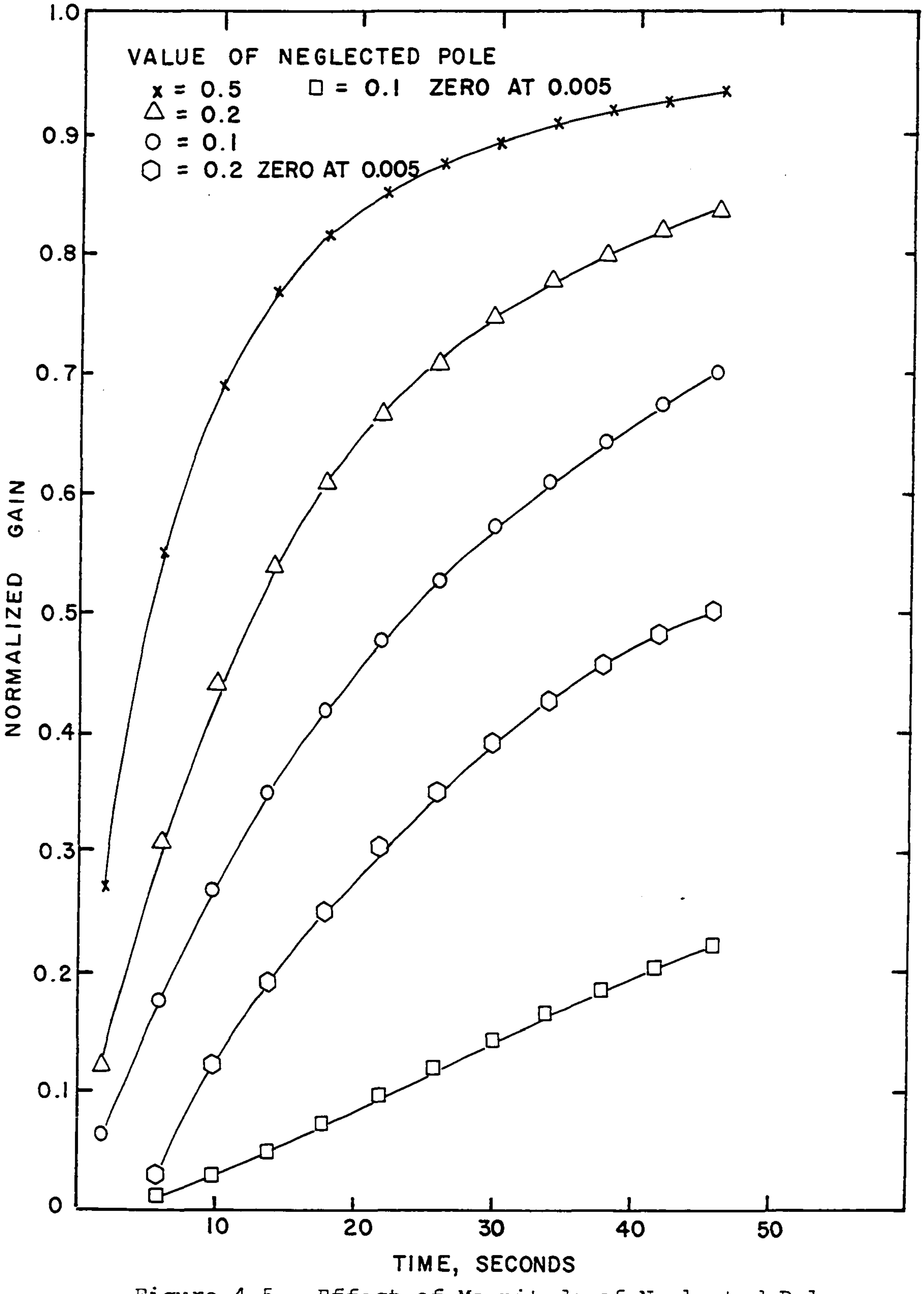


Figure 4-5. Effect of Magnitude of Neglected Pole on an Identification of Incorrect Order.

TABLE 4-2.

EFFECT ON THE IDENTIFICATION OF THE RELATIVE LOCATION OF THE POLES FOR AN ASSUMED SYSTEM ORDER LESS THAN THE ACTUAL ORDER.

Pole 1	Pole 2	Pole 3	Zero 1	Zero 2	Neglected Poles		Omitted Zeros		Percent Error in Identified Gain at t = 50 seconds
					Pole 1	Pole 2	Zero 1	Zero 2	
0.5	0.02	0.002			0.5				6.5
0.5	0.02	0.002			0.5	0.02			65.15
0.5	0.02	0.002	0.01		0.5				15.0
0.5	0.02	0.002	0.01		0.5	0.02	0.01		16.0
0.5	0.02	0.002	0.2	0.04	0.5		0.2		30.5
0.5	0.02	0.002	0.2	0.04	0.5	0.02	0.2	0.04	68.3
0.5	0.02	0.002	0.04	0.01	0.5		0.04		171
0.5	0.02	0.002	0.04	0.01	0.5	0.02	0.04	0.01	122
0.5	0.02	0.01	0.05	0.025	0.5		0.05		154
0.2	0.02	0.01	0.05	0.025	0.2		0.05		116
0.1	0.02	0.01	0.05	0.025	0.1		0.05		66.8
0.1	0.02	0.01	0.05		0.1				77.9
0.2	0.02	0.01	0.05		0.2				49.5
0.1	0.02	0.01			0.1				30.0
0.2	0.02	0.01			0.2				16.4
0.5	0.02	0.01			0.5				6.5

is much worse than when there is a gain only. From other tests which were conducted, where the neglected pole was less than or equal to the zero, it can be concluded that the closer the neglected pole approaches to the magnitude of the largest zero, the worse the approximate identification. When the pole and zero are equal, no approximation is possible.

Effect of Errors in Identified Values of the Poles

The preceding sections have given an idea of the consequences of an incorrect identification of the order of the system, with the resultant conclusion that it is better to include a pole even though its validity may be in question. Assuming the order of the system is correctly determined, there are still many things which could affect the identification. One of these problems, which is very probable, is an error in the identification of the poles. To determine this effect, a series of tests was run in which an actual ± 5-percent error was introduced into the poles of the system

$$Y(s) = \frac{(s+0.04)\ (s+0.2)\ X(s)}{(s+0.1)\ (s+0.05)\ (s+0.02)} \tag{4-8}$$

The worst effect occurs when the errors in all of the poles are in the same direction, either positive or negative. The other combinations of positive and negative errors are confined within these limits.

The results of these two limiting cases are plotted in Figure 4-6. It is interesting to note that even though

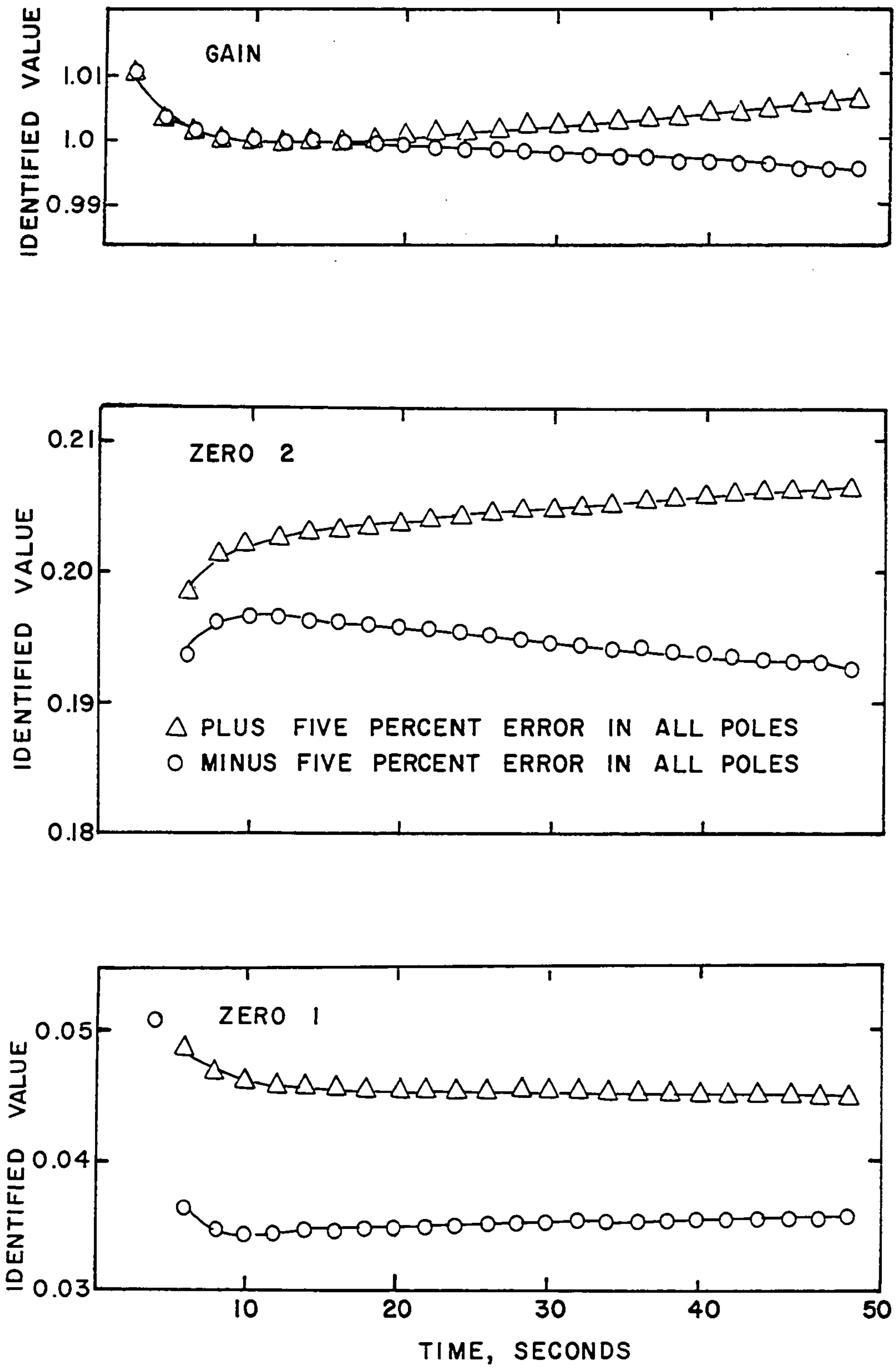


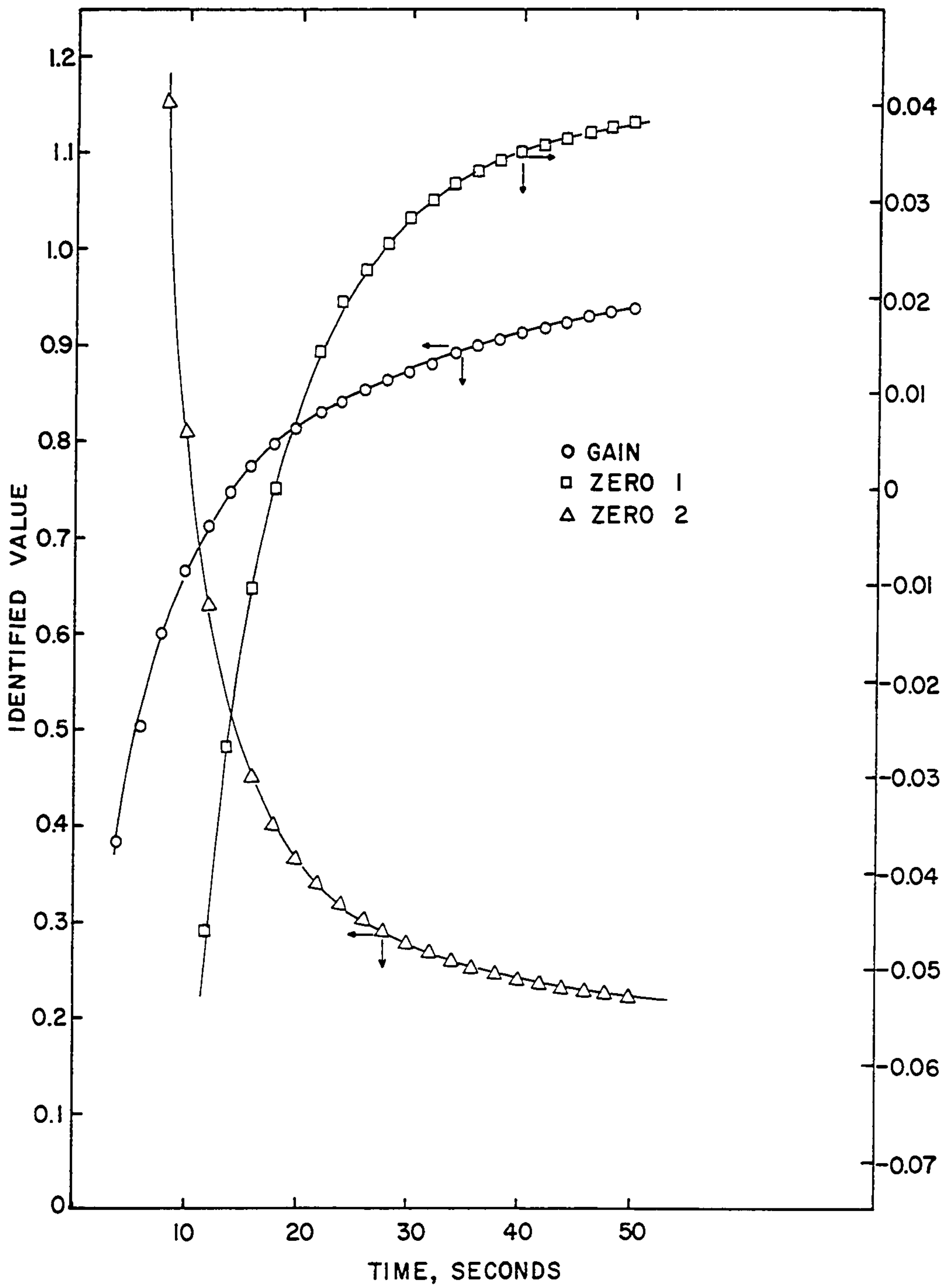
Figure 4-6. Identification of a Third Order System with a Five Percent Error in All of the Poles.

all three poles were in error by 5-percent, the effect on the gain was at most 0.59-percent, and in the largest zero, only 3.5-percent after fifty seconds. The error in the smallest zero was nearly twelve percent at the same time. However, this error is decreasing with time while the other errors are increasing.

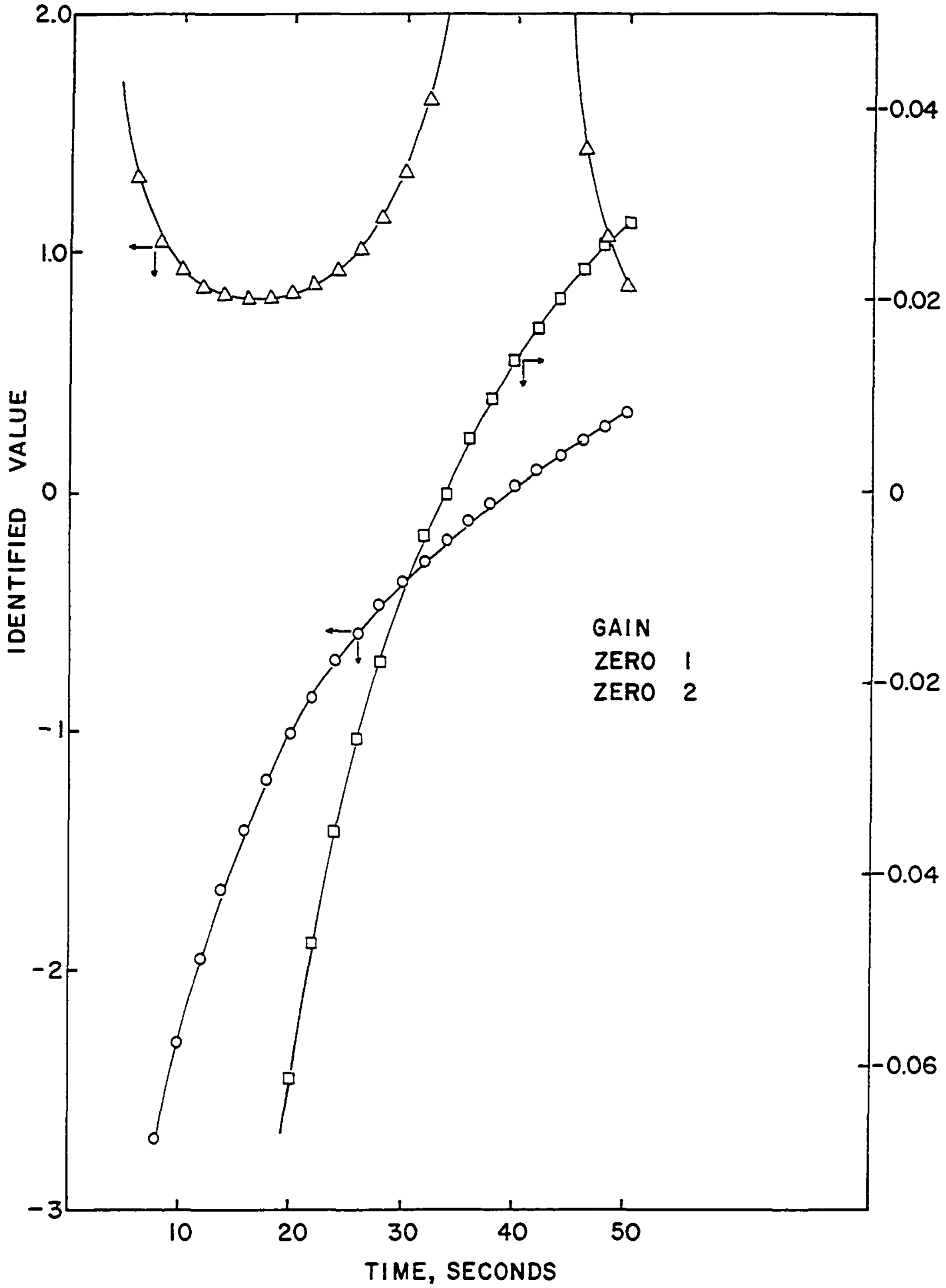
Effect of Steady State Errors

Another cause of a poor identification could be that the steady state value of both the forcing and response function has been incorrectly determined. It was expected that a steady state error in the input would have a serious effect on the quality of the identification because of the integration of the homogeneous weighting function times the input, which was required in the identification procedure. However, from the tests which were conducted, it was shown that correct identification is still possible as long as this steady state error is not too great. Even when the steady state error is as much as ten percent of the amplitude of the inputs, the identification appears to converge eventually to the correct value, as shown in Figures 4-7 and 4-8.

When there is an error in the response, its effect upon the identification is not nearly as great as an error in the forcing function. As can be seen from Figure 4-9 the identification did tend to converge to the correct values, with the errors in the identified zeros and gain being less than the error in the response by the time of fifty seconds.



4-7. Effect of Input Steady State Error of One-percent.



4-8. Effect of Input Steady State Error of Ten-percent.

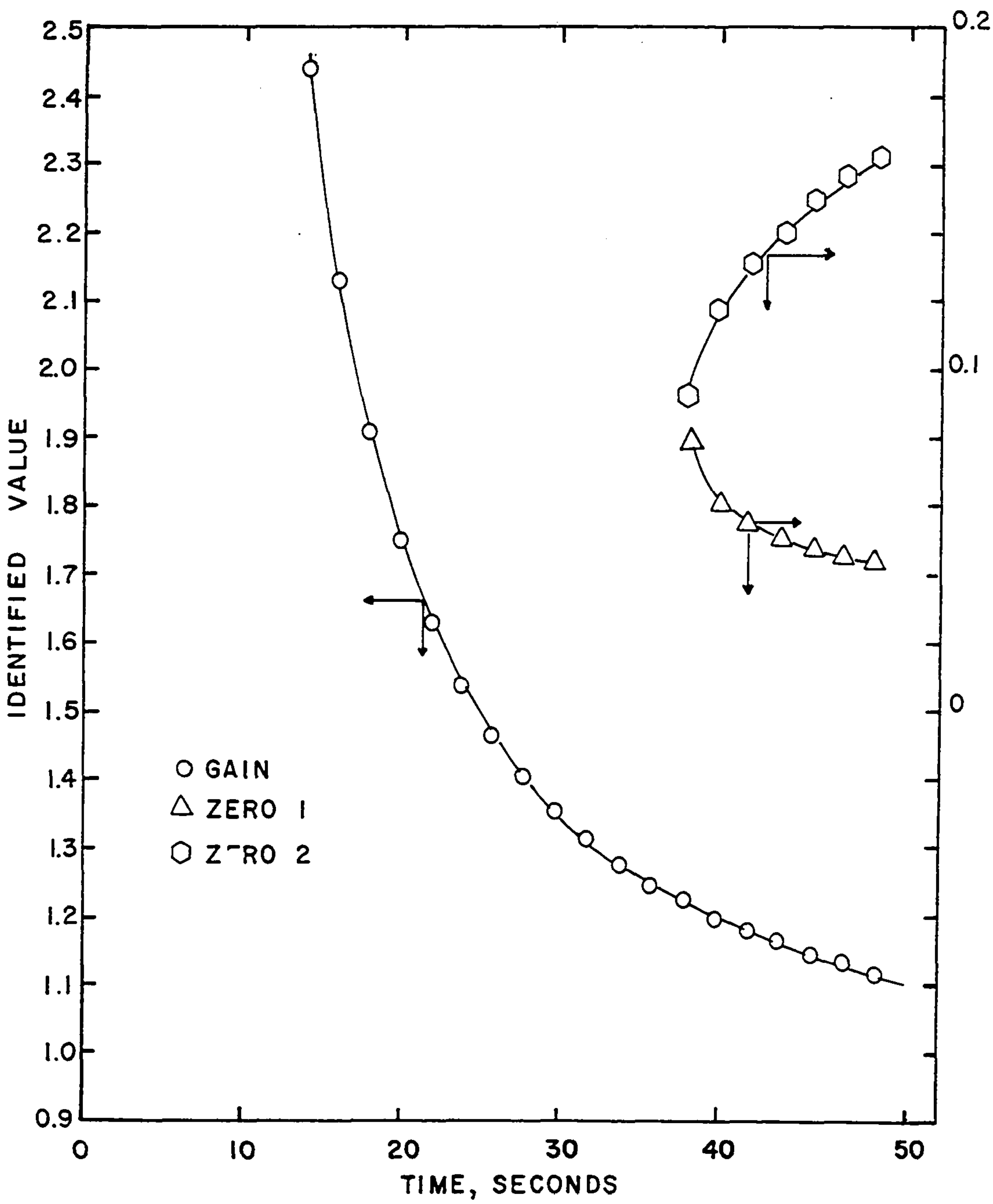


Figure 4-9. Effect of Steady State Error of Twenty Percent in the Response.

Effect of Transport Delay

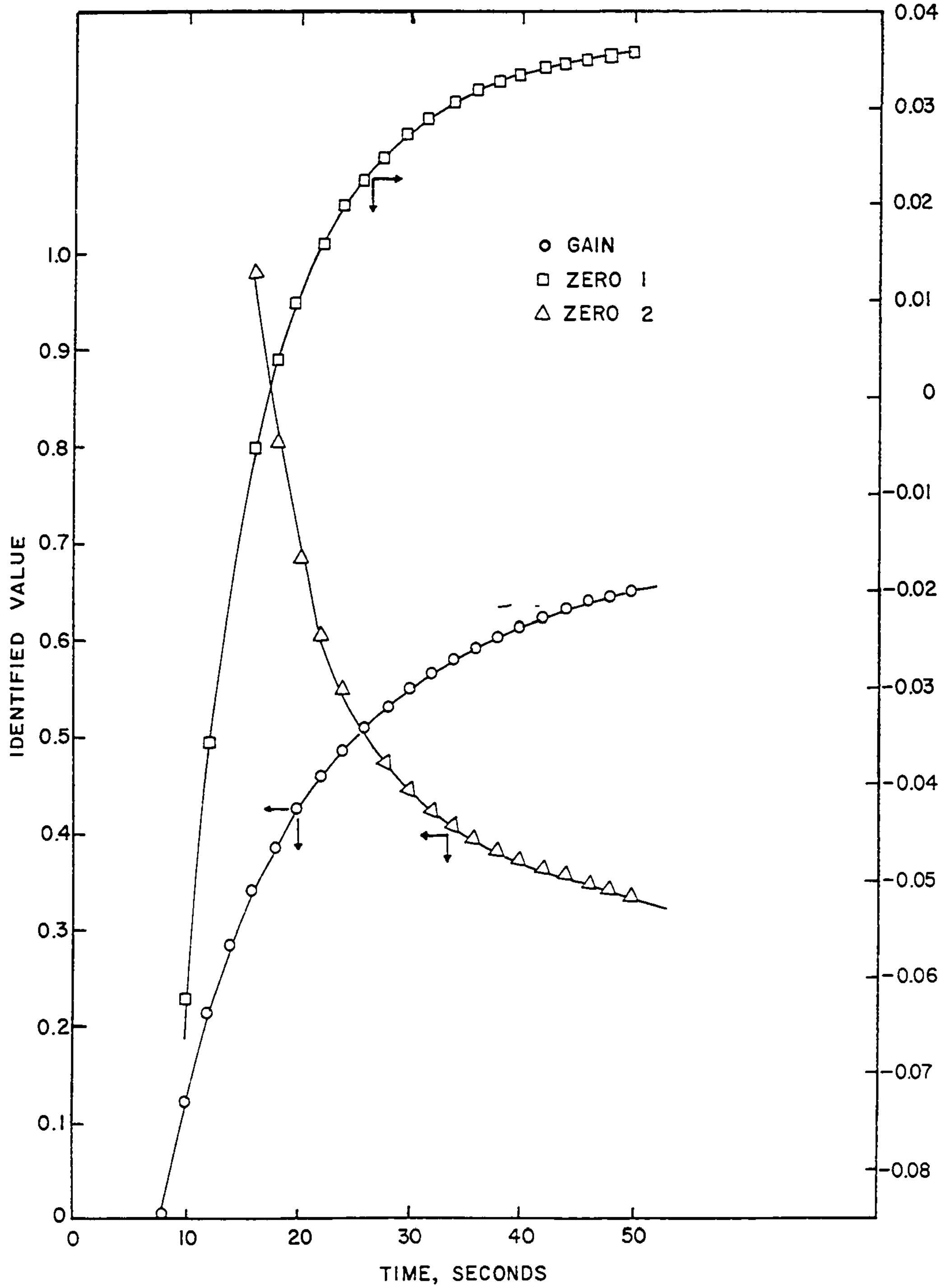
Any transport delay contained in the process or measurement and transducing equipment will normally cause a poor identification. To study the exact effect that transport delay has upon the identification, the response from a three-pole, two-zero system was actually shifted to simulate transport delay of one, five, and ten seconds. The identification was then carried out with these shifted data. The results when the shift was one second are plotted in Figure 4-10. The gain and zeros tend to converge to the correct values. The lowest frequency zero is least affected by this time delay.

Figure 4-11 shows the results when the response is shifted by ten seconds. Here again the lowest frequency zero is less affected and appears to converge to the correct value. The other zero and gain, however, converge to a value which is the negative of the correct value.

Effect of Number of Significant Figures in the Data

To explain some effects which were noted in the results of both the analog computer studies and the identification of the experimental process, a series of studies were conducted to determine the effect of data precision. This series of studies consisted of digitally computing the response of the system

$$T_w = \frac{0.000156(s+0.033)W}{(s+0.0117)(s+0.045)(s+0.0583)} \qquad (4\text{-}10)$$



4-10. Effect of One Second Transport Delay.

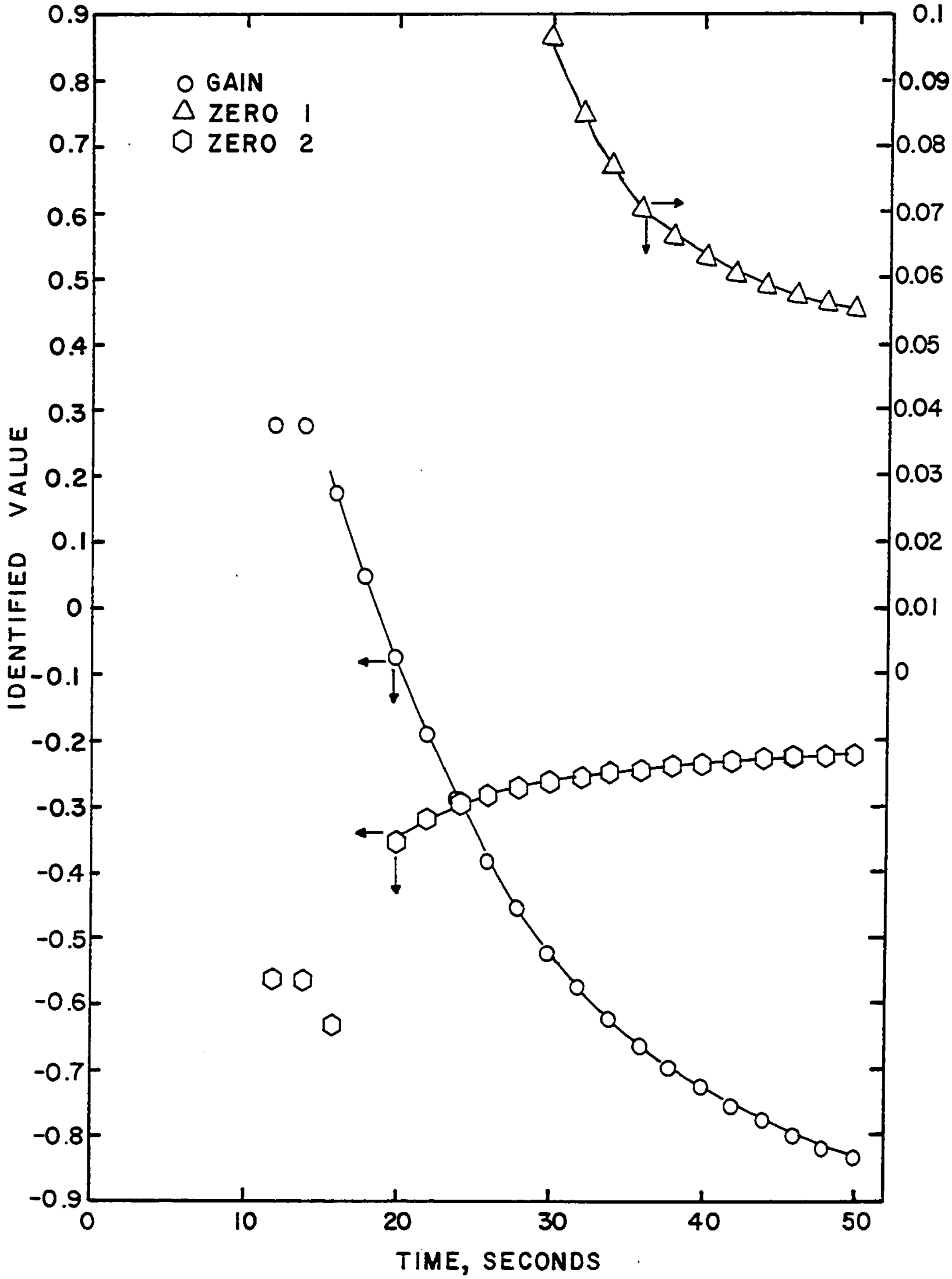


Figure 4-11. Effect of Ten Second Transport Delay.

for inputs of a step of amplitude, 13.0169; a ramp of slope, 0.09918; and a sine wave of amplitude, 30.6253, and angular frequency, 0.0628.

The data from this system were recorded to seven significant figures and were used as input to the identification program. The data were then truncated to six significant figures and the identification computed. This process of deleting one significant figure and computing the identification was repeated through two significant figures. Partial results of this series of studies for step and sinusoidal forcing are shown in Figure 4-12. The identification has less than 0.1-percent error when seven or six significant figures are used. With five figures, a deviation of 3.6-percent occurs at 5 seconds and 55 seconds. With four significant figures these deviations are greater (34-percent) with a new deviation occurring at t=175 seconds. With two or three significant figures these deviations are very pronounced and a new deviation occurs at t=130 seconds. The initial convergence is also seen to be slow.

These deviations, with the exception of the one at t=130, are caused by the approximate singularity of the coefficient matrix generated by k solutions of Equation (2-19). In this specific case, the determinant of the coefficient matrix changes sign at t=55, 130, and 175 seconds. At these points two ten digit numbers are being subtracted to obtain a six digit number, i.e., a loss of four significant figures.

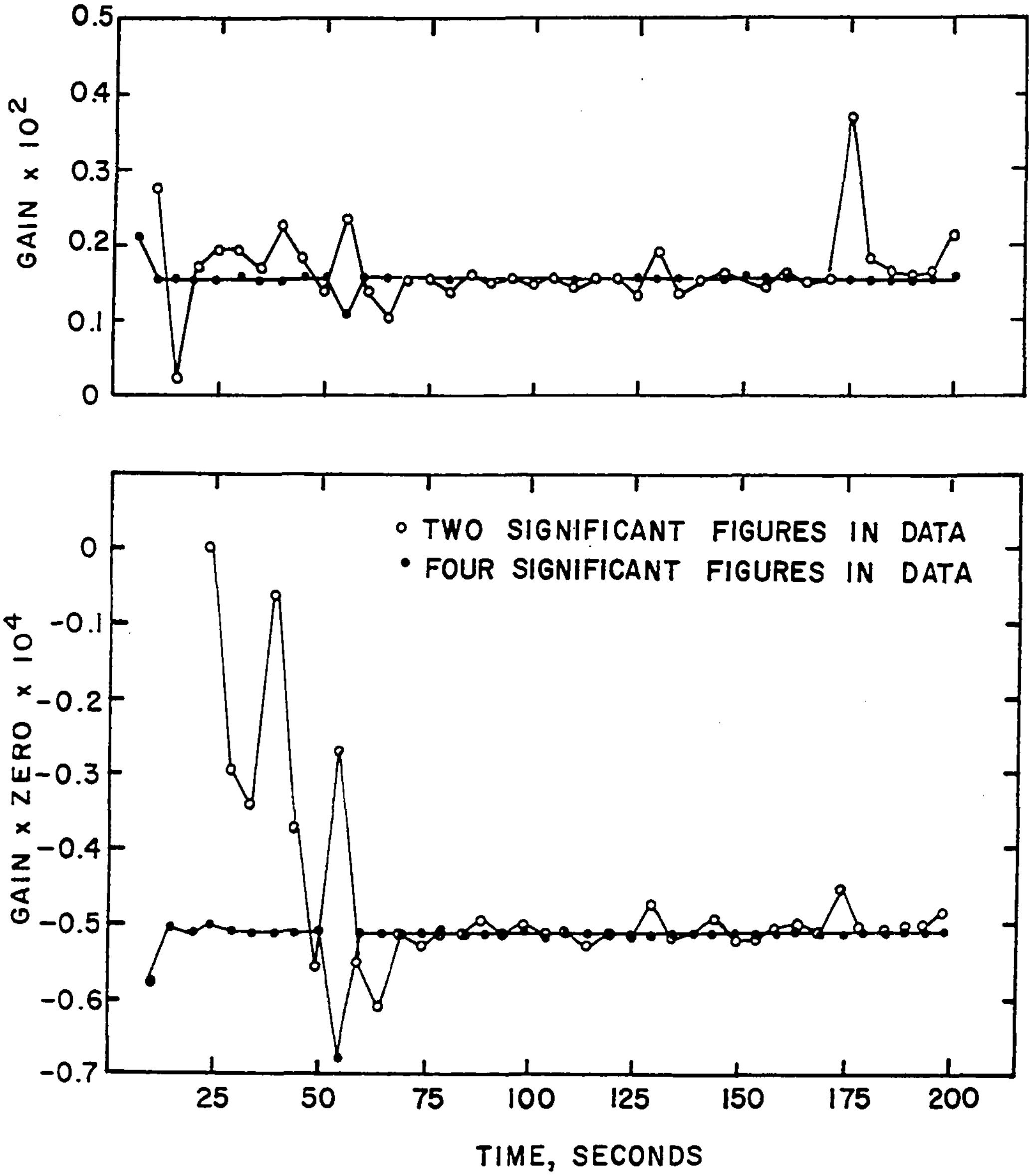


Figure 4-12. Effect upon the Identification of the Number of Significant Figures in the Data.

At t=130, the response is also crossing zero which has a greater effect than the truncation of the determinant. The determinant is still an eight digit number.

Effect of Noise

To investigate the effect of noise on the identification, and in preparation for a study of the effect of a non-linearity in the system, the analog computer was used to simulate the theoretical model of the experimental equipment described in the next chapter. For this discussion and the discussion on non-linearities the experimental equipment can be represented by the schematic diagram of Figure 4-13.

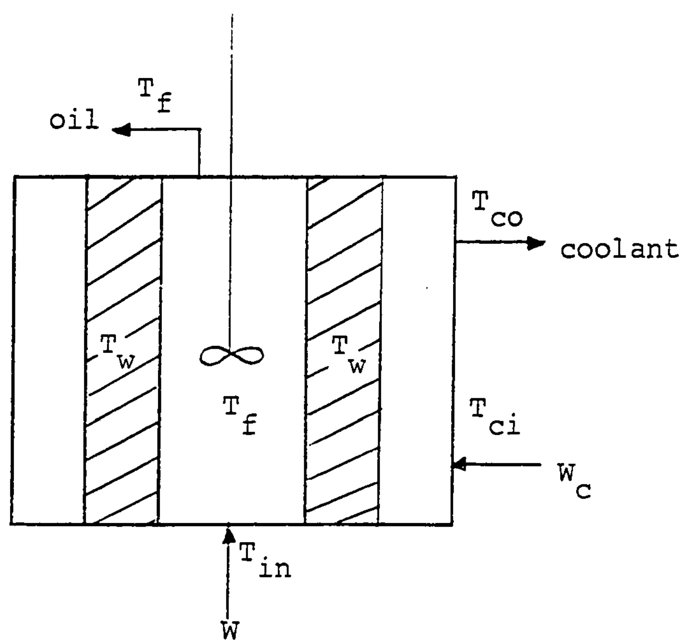


Figure 4-13. Schematic Diagram of Experimental Equipment.

The linear model was used and is described by the differential equations

$$\dot{T}_f = -0.0569T_f + 0.006T_w + 0.00293W \qquad \text{(a)}$$

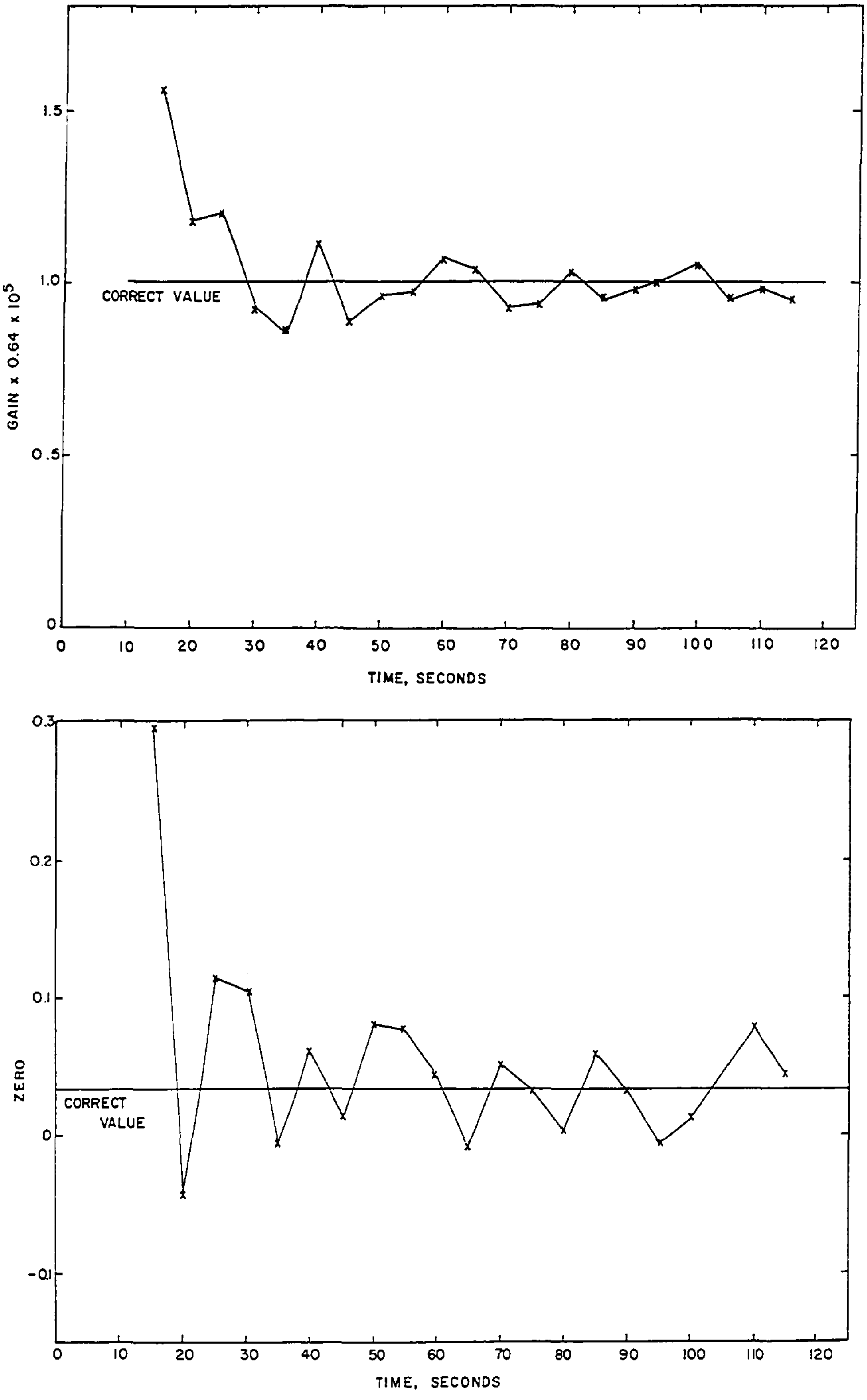
$$\dot{T}_w = 0.00532T_f - 0.0252T_w + 0.00994T_{co} \qquad \text{(b)} \qquad (4\text{-}9)$$

$$T_{co} = 0.0273T_w - 0.329T_{co} - 0.208W_c \qquad \text{(c)}$$

The noise was assumed to be caused by either of two mechanisms. The first is the introduction of noise into the system from noise in the forcing input function, with no other noise sources present. The second mechanism of noise generation assumed no noise in the forcing input function. It was further assumed that all of the noise was either generated within the system, or introduced from an input other than the one being forced, or was generated in the transducing and measurement equipment. To study these two mechanisms, a random, square-wave, noise generator designed especially for the Process Control Laboratory was used as a noise source.

For the first case the noise was superimposed on the forcing function to provide the system input. As the noise was a part of the input, and therefore measured as a part of the total input variable for the zero identification, there was no appreciable effect upon the quality of the identification.

For the second mechanism, which is by far the most likely to occur, the result was not so simple. To study this case, the signal from the noise generator was superimposed upon the response signal before recording. The noise



4-14. Effect of Five-percent Noise in the Response Function.

signal was of constant amplitude, independent of the amplitude of the response. The amplitude of the noise signal was approximately ± five percent of the maximum amplitude of the response signal. Figure 4-14A shows the results of the gain identification for the system of Equations (4-9) when the forcing functions for the hot oil flow (W) input were a step of 100 volts and a ramp of slope 1.0. The average frequency of the noise used was 0.77 zero crossing per second. Figure 4-14B shows the zero identification for this same system. As can be seen from these graphs, the identifications tend to oscillate about the correct values with an error of approximately ± five percent in the gain and ± one hundred percent in the zero. However, the average value of the gain past the time of twenty seconds is 0.998, or only 0.2-percent in error, and the average value for the zero is 0.039, about eighteen percent in error. From this result it appears that to best determine the gain and zero, it is better to calculate these quantities at many points of time and then average the results, than to depend upon an identification at any one point in time.

Effect of System Non-linearities

The system used in this study was very similar to the non-linear model of the experimental system. The system is given by

$$\dot{T}_f = 0.0569T_f + 0.006T_w + 0.0293W - 0.00307\,WT_f \qquad (a)$$

$$\dot{T}_w = 0.00532T_f - 0.0252T_w + 0.00994T_{co} \qquad (b) \qquad (4\text{-}11)$$

$$\dot{T}_{co} = 0.0273T_w - 0.032T_{co} - 0.208W_c - 0.00844W_c\,T_{co}. \qquad (c)$$

The linearized version of this system is:

$$\dot{T}_f = -0.0569T_f + 0.006T_w + 0.0293W \qquad (a)$$

$$\dot{T}_w = 0.00532T_f - 0.0252T_w + 0.00994T_{co} \qquad (b) \qquad (4\text{-}12)$$

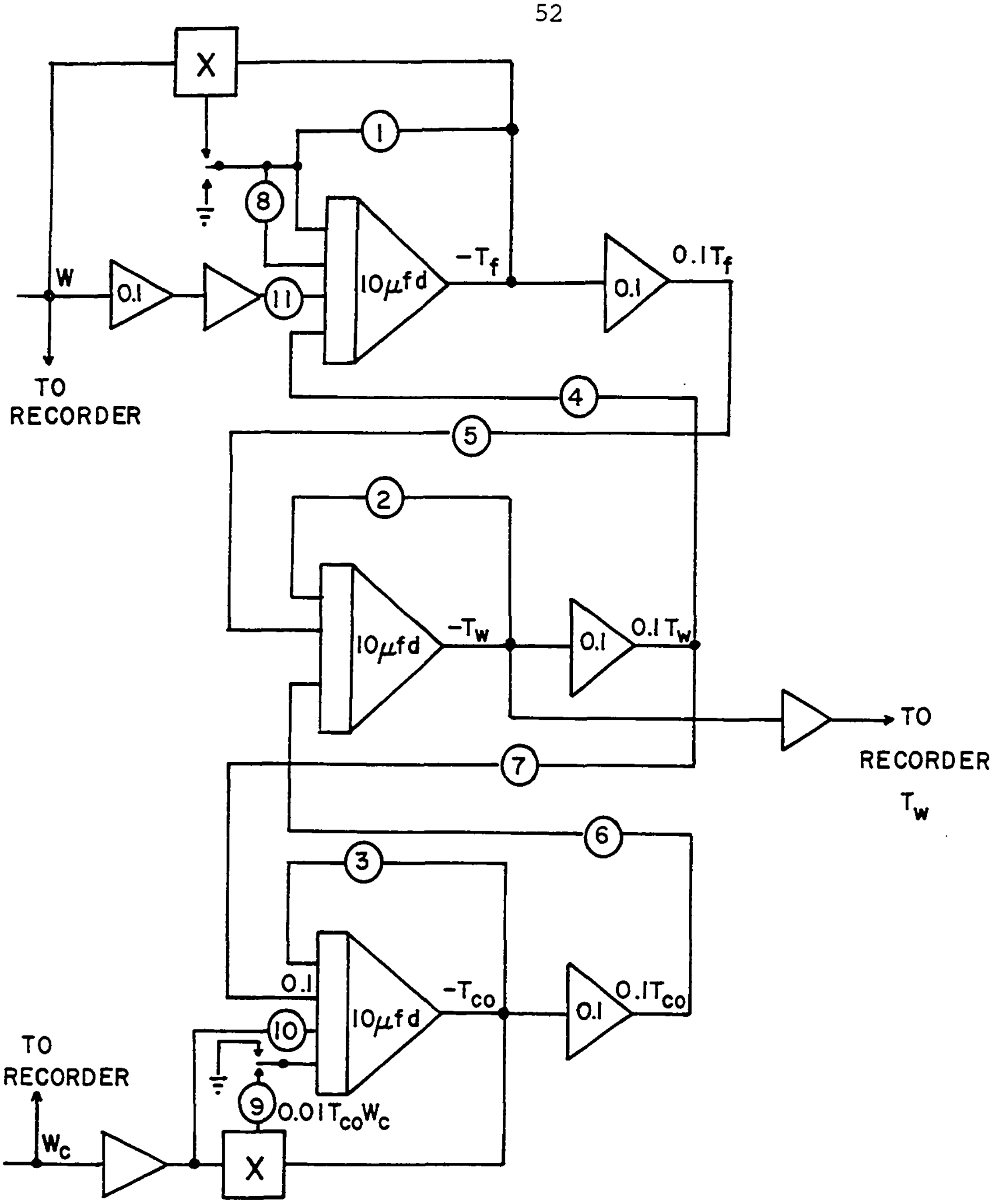
$$\dot{T}_w = 0.0273T_w - 0.032T_{co} - 0.208W_c \qquad (c)$$

When the wall temperature is the response, and the two flow rates are the forcing variables, the transfer function for this system is

$$T_w = \frac{1.56\times10^{-3}(s+0.0329)W - 2.07\times10^{-3}(s+0.0569)W_c}{(s+0.0117)\ (s+0.045)\ (s+0.05833)} \qquad (4\text{-}13)$$

This system was programmed on the analog computer in such a manner as to allow both the linear and non-linear versions to be investigated. The analog program is shown in Figure 4-15.

The same forcing was used for both the linear and non-linear models so that these identifications could be compared. The greatest difference between the forcing of the models and the actual experimental equipment was the magnitude of the forcing function and consequently, the magnitude of the deviation from the steady state and the relative magnitudes of the non-linear term. Table 4-3 gives a quick comparison of these systems for step forcings.



COEFFICIENT POTENTIOMETERS

1. 0.569	5. 0.532	9. 0.844
2. 0.252	6. 0.994	10. 0.208
3. 0.329	7. 0.273	11. 0.293
4. 0.6	8. 0.307	

Figure 4-15. Analog Computer Circuit for Simulation of the Experimental Equipment.

TABLE 4-3.

COMPARISON OF THE MAGNITUDE OF THE NON-LINEARITIES IN THE ANALOG SYSTEM AND THE EXPERIMENTAL PROCESS

	Analog System			Experimental Process		
Forcing	Magnitude	T_w	Ratio	Magnitude	T_w	Ratio
oil	+13 v	+18.82 v	1.115	+0.0125 lb/sec	5.15°F	1.082
	-13 v	-21.81 v		-0.0125 lb/sec	5.58°F	
coolant	+10 v	-26.31 v	1.89	+0.002 lb/sec	6.54°F	1.37
	-10 v	+49.64 v		-0.002 lb/sec	8.95°F	

As can be seen from this table, the analog system, especially for coolant forcing, is quite a bit more non-linear than the actual experimental process.

If in Equation (4-11a) the non-linear term is lumped in with the fluid temperature, and in Equation (4-11c), it is lumped in with the coolant term, a good idea can be obtained as to the extent that the non-linear term affects the system.

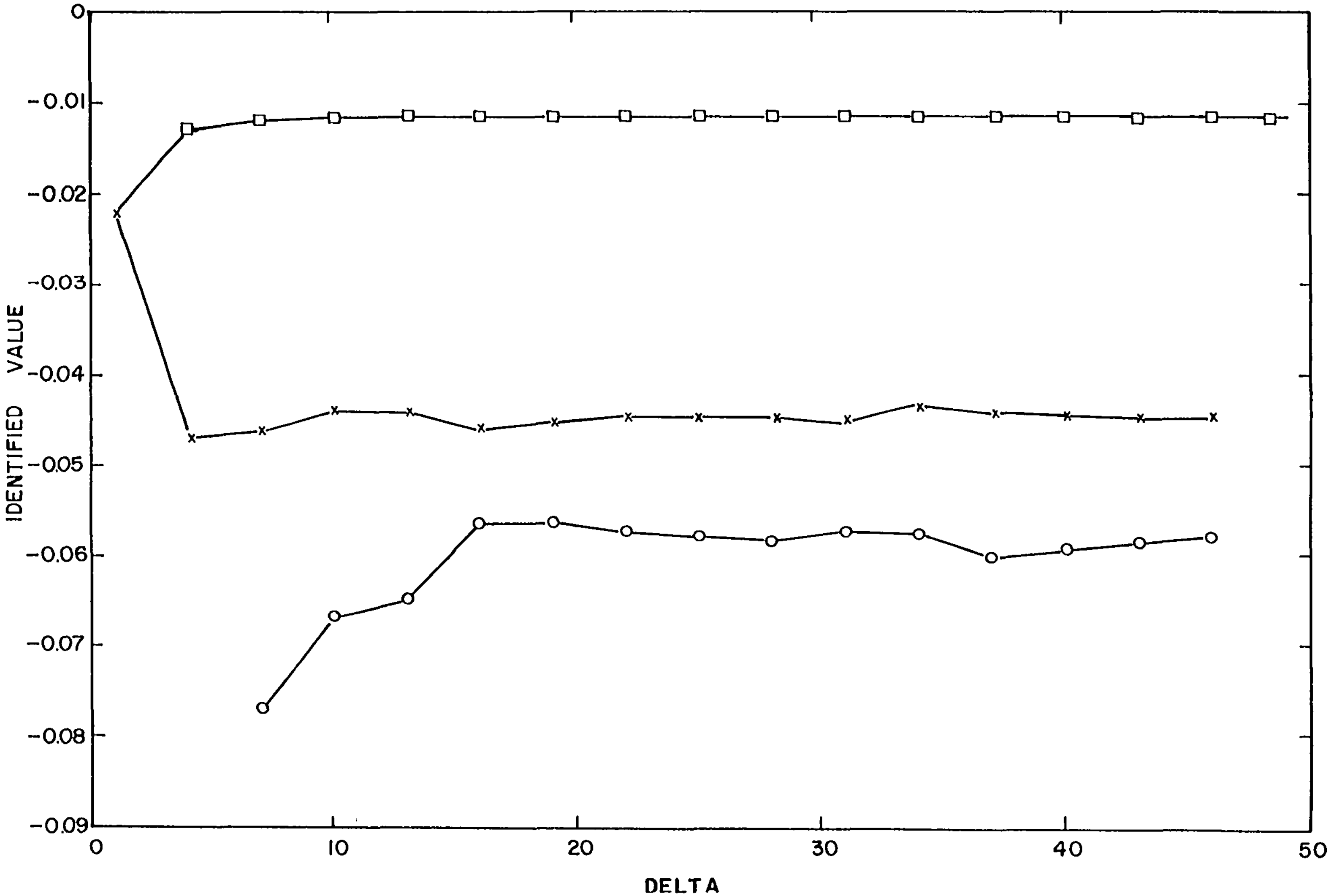
$$\begin{aligned} T_f &= -(0.0569 + 0.00307W)T_f + 0.006T_{co} + 0.0293W \\ T_{co} &= 0.0273T_w - (0.0329 + 0.00844W_c)T_{co} - 0.208W_c \end{aligned} \tag{4-14}$$

If the magnitude of the step is then substituted into these lumped terms,

$$\begin{aligned} (0.0569 \pm 0.00307 \times 13)T_f &= 0.0969T_f \text{ or } 0.0169T_f \\ (0.0329 \pm 0.00844 \times 10)T_{co} &= 0.1173T_{co} \text{ or } -0.0515T_{co} \end{aligned} \tag{4-15}$$

It can easily be seen that the non-linear term has a drastic effect upon these coefficients. In the case of the oil, the coefficient is changed by $\pm$ 70-percent and for the coolant by $\pm$ 250-percent. The coolant coefficient even changes signs.

The non-linearity, however, has no effect upon the identification of the poles of the model as can be seen in Figure 4-16. This result is to be expected since the tests used in the pole identification are relaxed responses, where both of the forcing variables are equal to zero, and the non-linear terms drop out of the describing equations. The identified poles are therefore the same for the linear and



4-16. Identification of Poles of Analog Simulation of Theoretical Non-linear Experimental Equipment Model.

non-linear system, even though in Equation (4-15) it is shown that the poles are drastically changed by the non-linearity. This result would lead one to the conclusion that it would be impossible to determine a gain and zero for the non-linear system.

In the case of the oil forcing, which has been shown to be the weaker of the non-linearities, a good linear approximation is obtained. This model is, under certain conditions, very similar to the model for the linear system. This approximation, however, cannot be considered a general model because it applies only under the conditions of the tests which produced it.

Figure 4-17 shows the results of an identification of the non-linear system in which the forcings were such that a very good linear model is obtained. This linear approximation varies little from the results of the identification of the linear system, which also are plotted in Figure 4-17 for comparison purposes.

As can be seen, the system gains are the same until t = 175 seconds, at which time the gain of the non-linear system begins to decrease. This is probably due to the increasing magnitude of the non-linearity caused by the ramp forcing. The results of another identification of the non-linear system are plotted in Figure 4-18. Again for comparison purposes, the results for the linear system are also shown. Unlike Figure 4-17 there is quite a difference between the models. This case, however, was the worst one encountered in all of the test combinations.

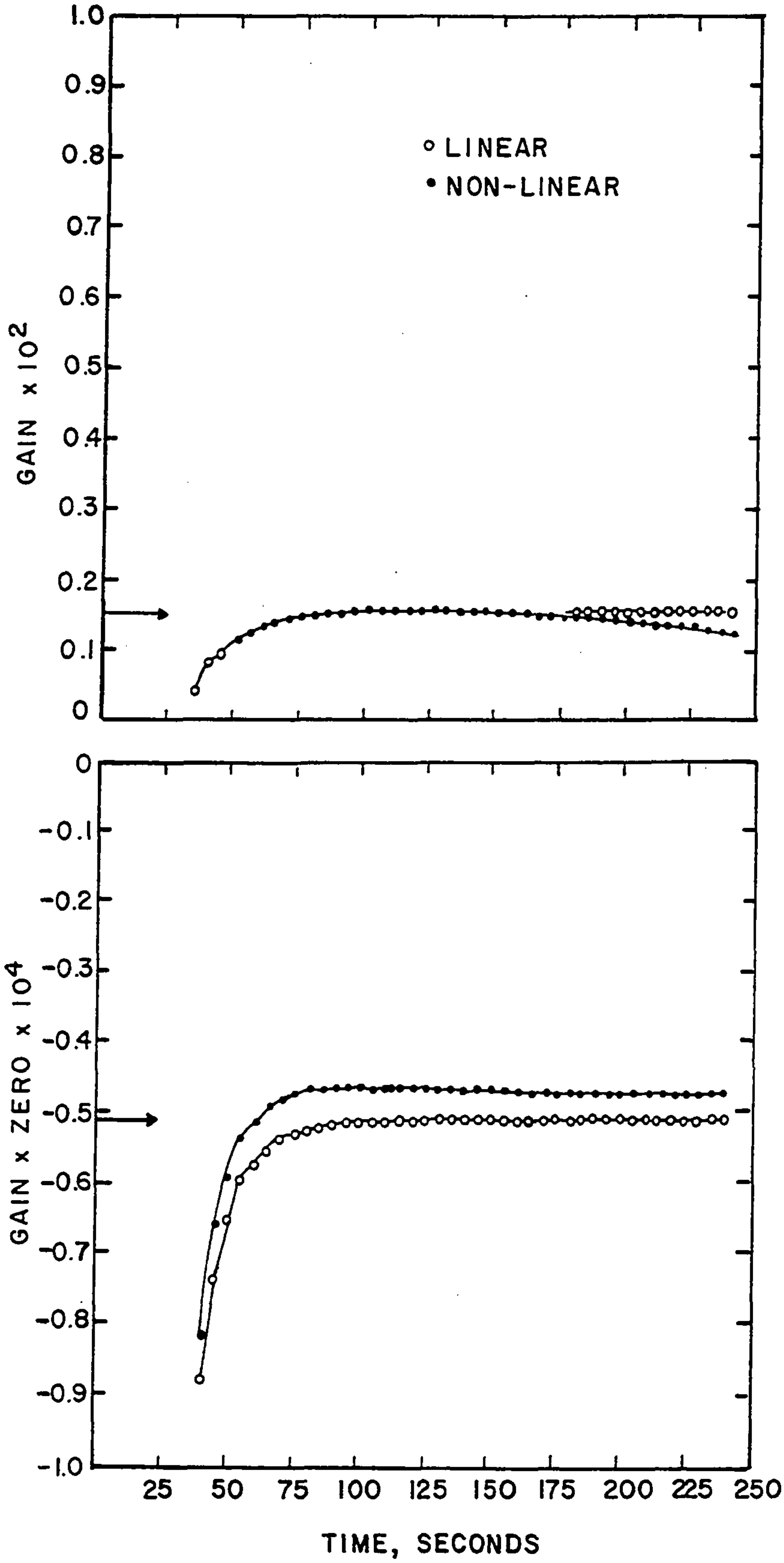


Figure 4-17. Identification of the Oil-forced Non-linear Model of the Experimental Equipment, Positive Ramp and Step Forcing.

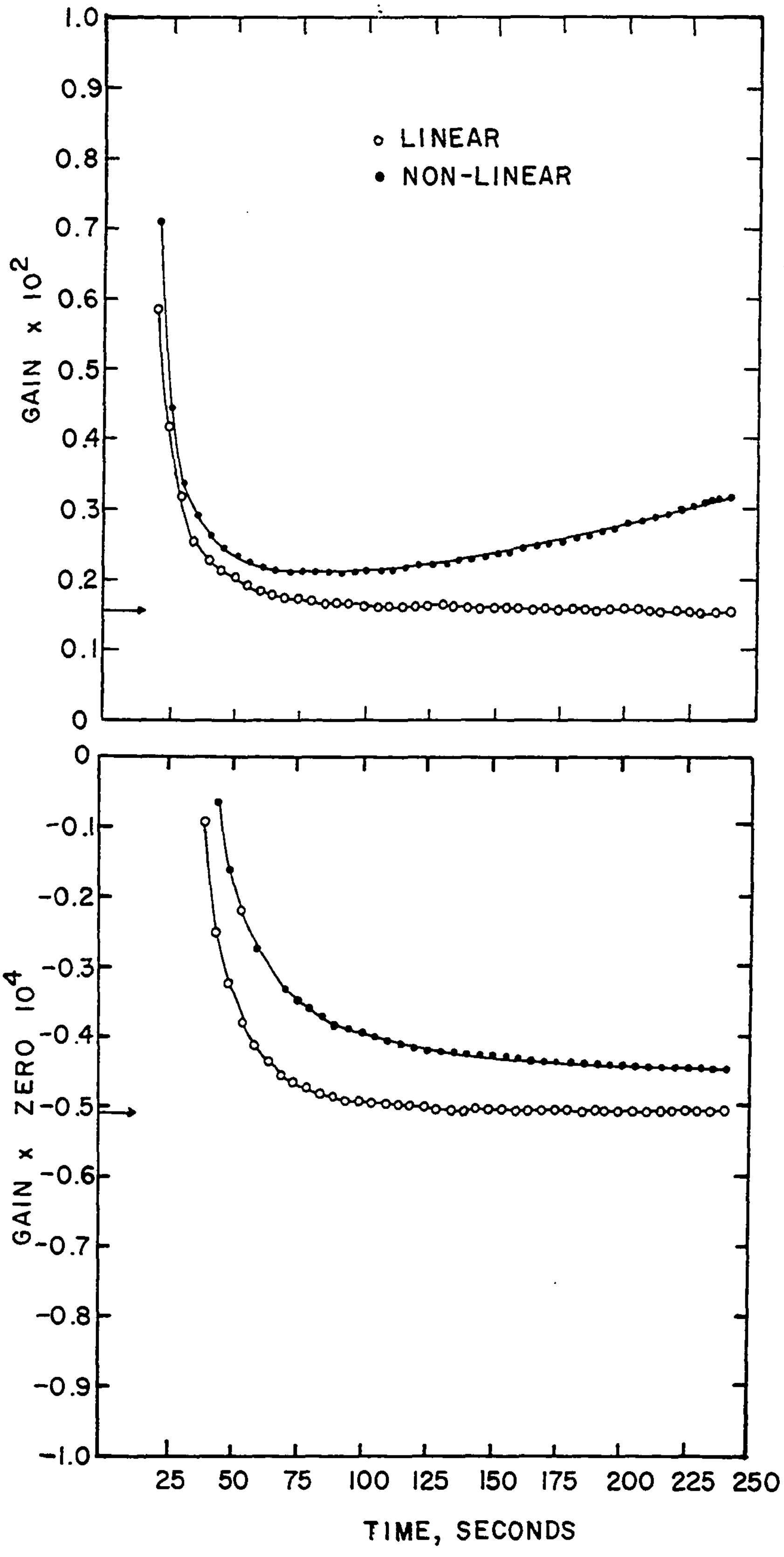


Figure 4-18. Identification of the Oil-forced Non-linear Model of the Experimental Equipment, Negative Ramp and Positive Step Forcing.

The forcing inputs used in the construction of Figure 4-18 were a positive step and a negative ramp. If a negative step and a positive ramp are used, the deviation in the identification is similar but opposite in direction. The combination of inputs used in Figure 4-17 was a positive step and a positive ramp. If a negative step and a negative ramp are used, the identification is again similar but with opposite deviations. This trait is very well pointed out in Figure 4-19 where the forcings are a positive step and a sine wave.

The identified value alternates above and below the identification results for the linear system at a frequency equal to the forcing frequency. The discontinuities which occur at t=55, 130, and 180 seconds are explained in the section on the effect of the number of significant figures in the data. Figure 4-20 shows how the predicted error varies with time. For both the gain and the zero, the error minimizes within the periods when the gain and the zero are closest to the values for the linear model. The error in the gain minimizes rapidly following a discontinuity, while the error for the zero minimizes just prior to the next discontinuity.

When an identification of the coolant forced system was tried, the results were not nearly as good as for the oil forced system. This behavior was to be expected because of the larger magnitude of the non-linearity. The best identification using step and ramp forcings occurred when these two forcings were in opposite directions. The results of these

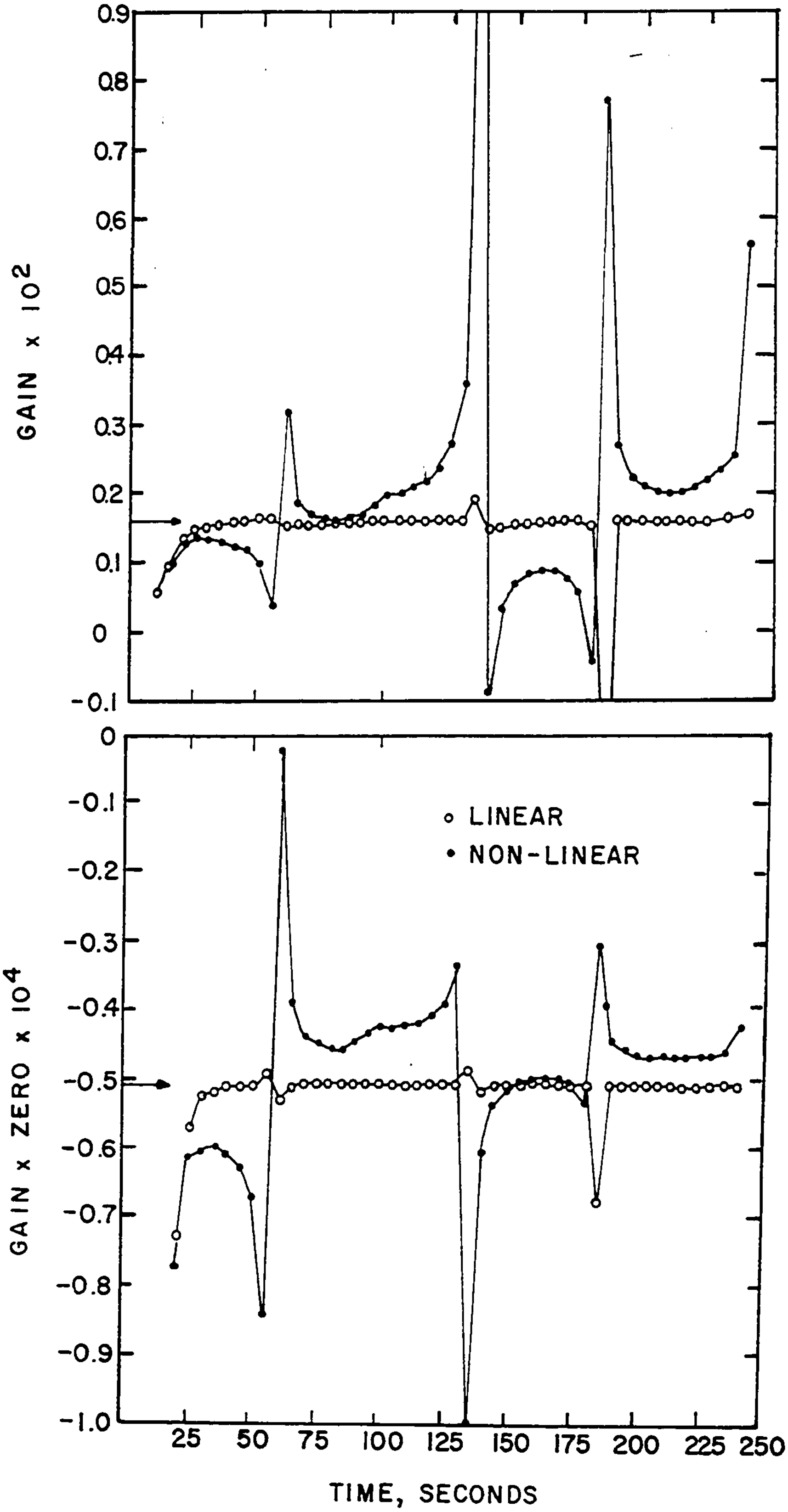


Figure 4-19. Identification of Oil-forced Non-linear Model of the Experimental Equipment, Sinusoidal Forcing.

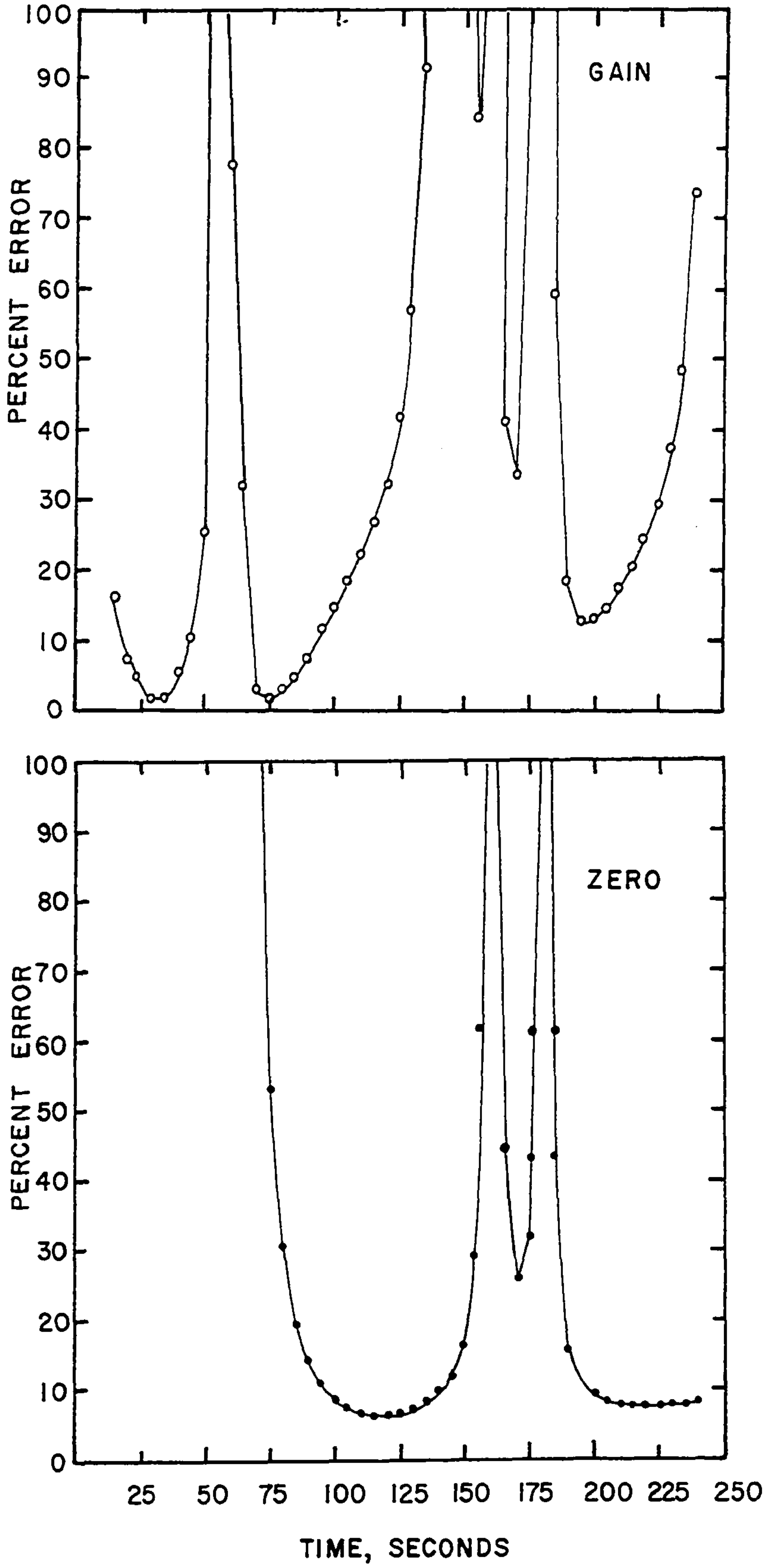


Figure 4-20. Predicted Error for the Identification Shown in Figure 4-19.

identifications are shown in Figure 4-21, along with the identified results for the corresponding linear system. Here again the results of the identified linear model are bounded by the results of the two oppositely forced non-linear model identifications.

It is interesting to note the times at which the error analysis predicted the minimum error would occur. This time was t=235 seconds for the identified gain of (A). It is at this point that the identified gain of (A) is equal to the gain of the linear system. For the gain of (B) no minimum predicted error occurred. However, at t = 240 the predicted error is still decreasing and the identified gain is still approaching the gain of the linear model. The times at which the minimum predicted errors for the identified zeros occurred are (A) t=185 seconds and (B) t=160 seconds.

When the forcings were in the same direction, an identification could not be made because of the drift in the results. The error propagation analysis did, however, predict that the error was a minimum at the time the identified gain of the non-linear system was the same as the gain of the linear system.

Figure 4-22 shows a representative identification of the coolant forced non-linear system when one of the forcing functions was a sine wave.

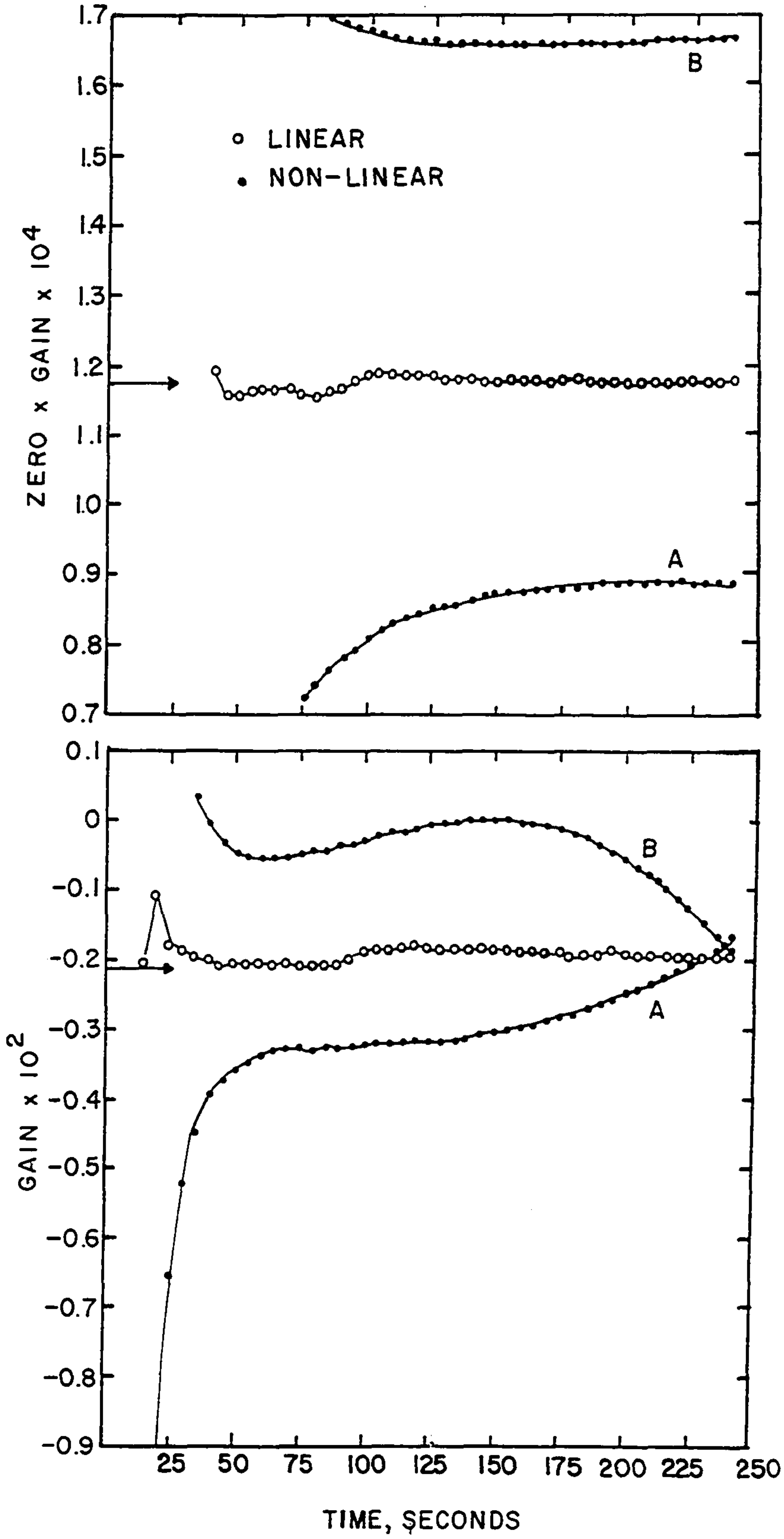


Figure 4-21. Identification of the Coolant-forced, Non-linear Model with Forcing Functions of a Step and a Ramp.

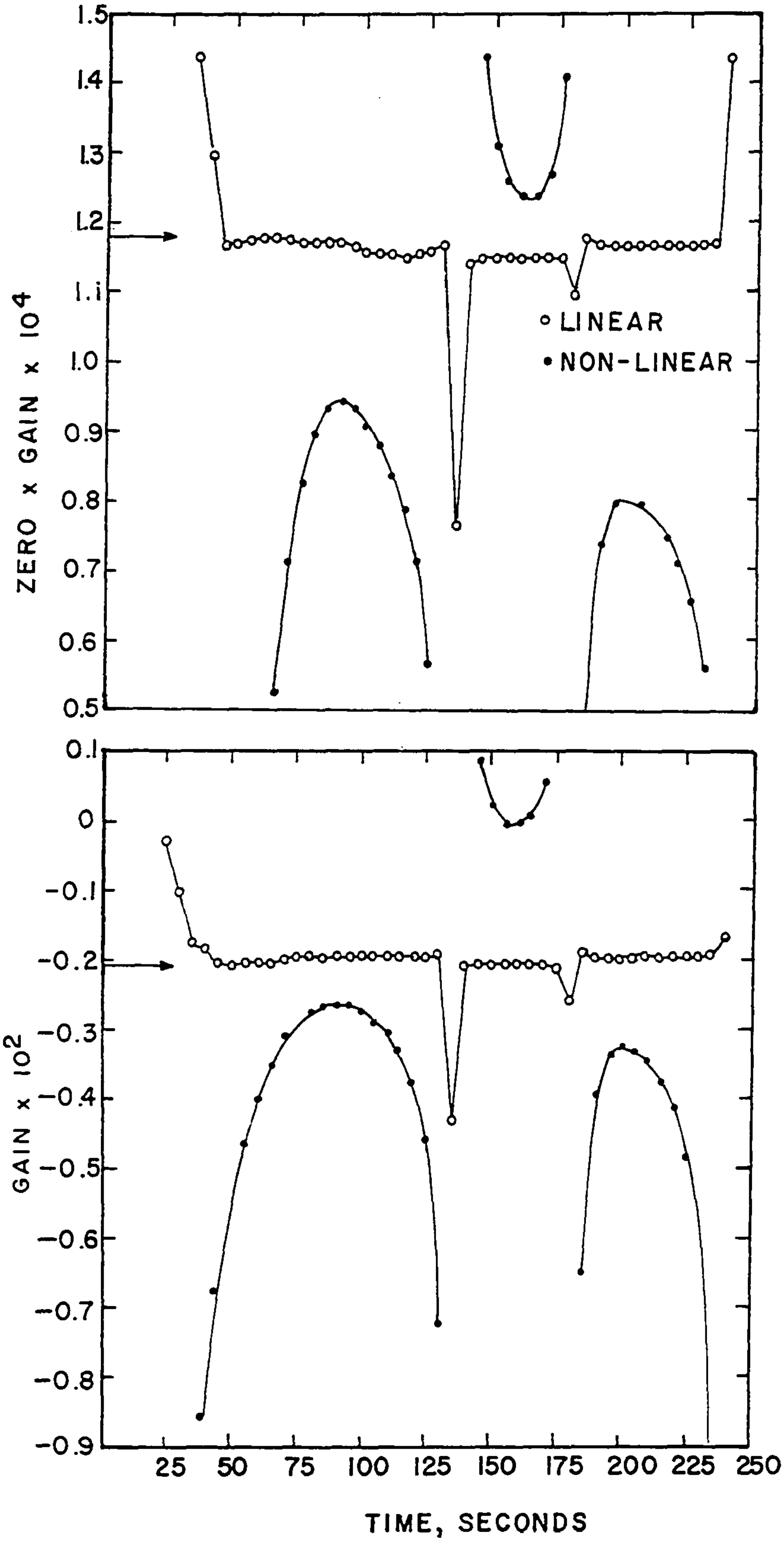


Figure 4-22. Identification of the Coolant-forced, Non-linear Model with One of the Forcing Functions a Sine Wave.

Conclusions About the Identification of Non-linear Systems

Because of the results which have been discussed in this section, and the fact that the system is so strongly non-linear, the author feels that it is possible to determine a linear model of a non-linear system using this technique. This model could be either a general model or a specific model. For example, a specific model could be used when there is to be a programmed change in plant operating conditions, such as a new steady state flow. It would only be necessary to test the plant between these steady state conditions. A general model would be useful for control purposes where the flow changes are small. The best model in this case would be the linear system. This model could be obtained by averaging the results of the various combinations of opposing identifications. It could also be easily obtained when one of the forcing functions was a sinusoid because of the fact that the identification for sinusoidal forcing gives the two distinct levels which bound the linear model.

CHAPTER V

EXPERIMENTAL STUDIES

In the Heymann (H2) dissertation and other chapters of this work the identification technique has been thoroughly computer tested on analog and digitally simulated processes. This technique was used in the identification of a laboratory process to evaluate its performance under conditions similar to those which would be encountered in a chemical processing plant.

Although the laboratory process was a simple stirred tank heat exchanger, it served as a good test of the identification procedure because of the unknown quantities present. Some of these unknowns include system non-linearities, steady state drifts, heat losses, measurement errors, and noise from several sources: the process, instrumentation, and transmission lines. Although every possible means was employed to eliminate or minimize these conditions some of these effects were still noticed.

Experimental Equipment

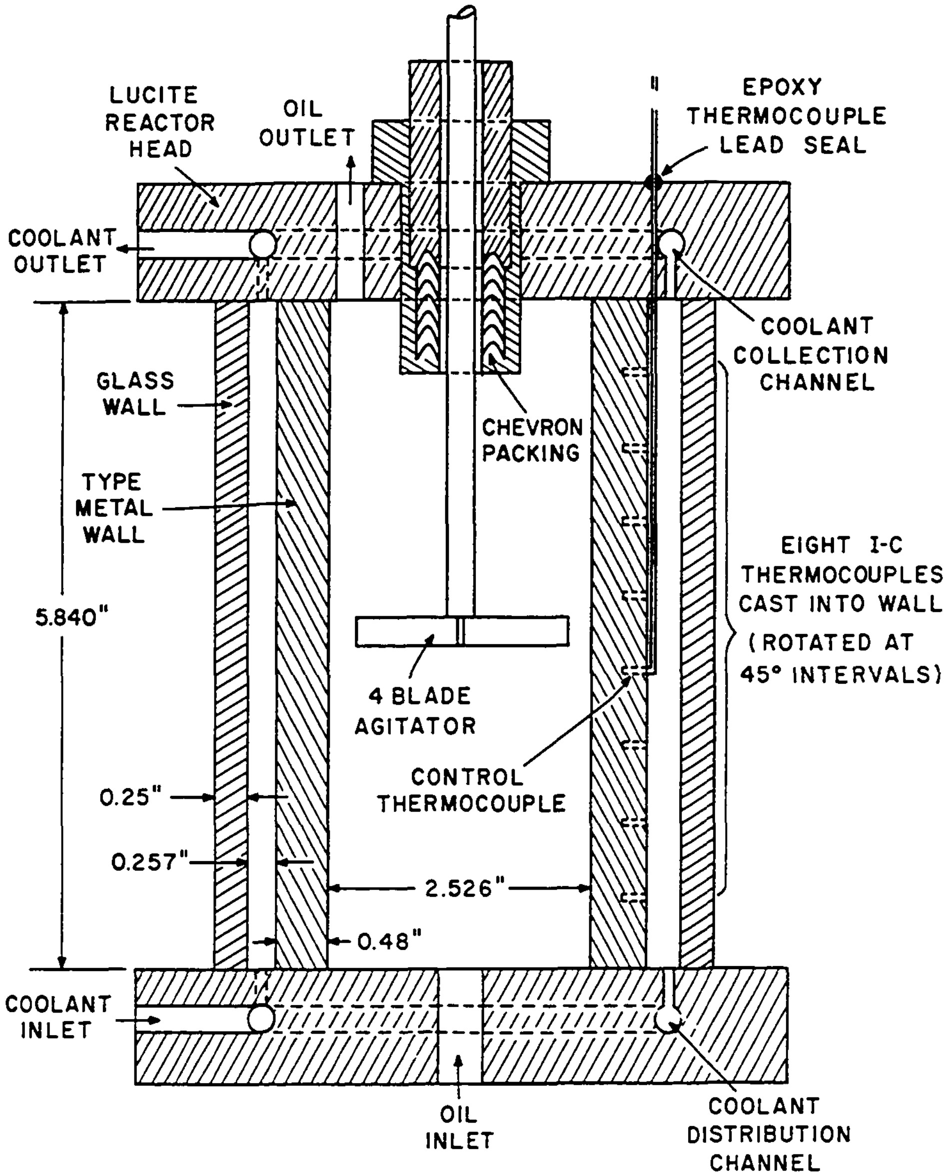
The experimental work described in the following sections was performed on equipment located in the Process Control Laboratory at the University of Oklahoma. This equipment is the evolutionary result of several previous identification studies (B1, F2, G1, S3) and has been used by two researchers (H1, L2) in the investigation of control systems. Because many changes have been made in the equipment since the most recent identification study, especially in the heat transfer-reactor simulator, a complete description of the equipment will be given.

The heart of the experimental system is a simulated continuous stirred tank reactor. Here, by simulated, is meant that heat transfer is the only rate operation occurring. This simulation is done as a matter of convenience in that it is easily accomplished and that no Arrenhius type non-linearity is needed in the model.

The Reactor

The reactor (Figure 5-1) is similar to that used in the identification study by Bishop (B1) and the invariance investigation of Haskins (H1). It is of a thick-walled design so that the heat capacitance of the wall must be considered creating a third order system. Many improvements have been made in the design of the reactor since these studies.

The new reactor wall has been precisely machined following the rough casting. The wall construction material is Type metal and contains Lead 75%, Antimony 15%, Tin 8%, Copper 2%. Because of this careful construction the volumes and heat transfer areas are well defined. Eight Iron-Constantan thermocouples were inserted in approximately the middle of



5-1. Detail of Reactor.

the wall thickness at 45° intervals around the wall and equally spaced along the height of the wall. In the experiments which were conducted, all of these thermocouples were connected in parallel to obtain an average wall temperature. The outer wall of pyrex provides an annulus for the flow of the coolant solution. Pyrex was chosen for its low thermal capacitance and because of its relatively low thermal conductivity (1:40) compared to the wall, as well as for the visibility provided.

The two walls are terminated on the ends by lucite plates, chosen for the same thermal properties as the pyrex outer wall. Also, the machinability of the lucite was desirable for several reasons: both the hot oil and the coolant enter through these ends, as do the thermocouples and the impeller shaft. These end plates include "distributors" for the coolant solution. These distributors are annular rings within the lucite with holes to distribute the coolant solution flowing concurrently with the oil.

The hot oil enters the bottom of the reactor and leaves through the top of the reactor. The oil in the reactor is stirred by a four-bladed, paddle-type impeller, 2.25 inches in diameter, located at the vertical midpoint of the reactor. The stirrer was driven by a 1/10 Hp. 1800 rpm motor.

Constant Temperature Baths

The light turbine oil is maintained at the desired temperature in two thirty gallon insulated stainless steel tanks, each containing heating and cooling coils. The temperature in the tanks are independently controlled. The cooling water flow is changed manually to adjust the control system characteristics. With steam used as a heating medium, the flow rate to each tank is controlled by Research Control valves type 75 G with D trim. The pneumatic control signals for the valves are generated by Minneapolis-Honeywell model 152P14P recording controlling pyrometers. The measurement signals are obtained from copper constantan thermocouples located near the outlets of the tanks.

The two tanks are used in series because of the nature of the control loops. If used separately a drift of about ± 1°F in steady state temperature occurs. However, when in series, the temperature of the first tank is allowed to oscillate over this range at a fairly high frequency (approximately 1 cycle per minute). The second tank is then used to smooth this variation and to correct for any long term drifts. In this manner the oil temperature to the reactor never drifted more than ± .25°F. Both tanks were stirred with 1/8 Hp Lightning model NC2 mixers. The oil was circulated by a Gould 1/2 inch helical gear pump.

The coolant, a 33-percent mixture of ethylene glycol in water, was maintained at its desired temperature in a 25 gallon bath. The coolant was stirred with a Lightning type RR 1/4 HP 100-1800 rpm mixer. The glycol mixture was cooled by a Copeland 1.5 ton, Freon-12 compressor with evaporator coils located in the glycol bath. A certain amount of manual temperature control was provided by means of the expansion valve, a Hoke type 4RB281, 20 turn, 1/16 inch orifice valve with a micrometer adjust handle. This valve could be used to change the evaporator pressure, and thus, the evaporator temperature.

Automatic control of the temperature was provided by another set of heat exchange coils in the bath. Another Minneapolis Honeywell pyrometer, Model 152P14, was used as the recording controller on this bath, with the temperature being sensed by a copper constantan type thermocouple. The final control element was a Research Controls type, 175 D, G trim, 1/4 inch air-to-close control valve which controlled hot water flow through the heat exchanger. In operation, the temperature of this bath never drifted over $\pm$ .25°F. The glycol solution was circulated by a 1/2 inch, gear pump.

Flow Measurement and Control

To take advantage of the operating characteristics of the Research Controls control valves used to regulate both the oil and coolant flow rates, it was necessary to utilize a flow splitting arrangement on both of these flow systems.

The trim used in the control valves was linear, i.e., for a constant differential pressure across the valve, the flow is proportional to the stem position which in turn is proportional to the pressure on the top. To maintain the constant differential pressure across the valves the flow through the pumps had to remain constant; therefore, a by-pass arrangement was used. Flow to the reactor was varied by an air-to-open valve. The remaining portion of the flow was returned to the feed tanks through an air-to-close-valve. The air-to-open and air-to-close valves were positioned by the same pneumatic signal.

All of the valves were Research Controls, type 75B, 1/4 inch valves, with 3 to 15 psi range springs, I trim for coolant, G trim for oil. The pneumatic signals for these valves were generated by Taylor Transet electro-pneumatic transducers, type 701TF111SIS124, which had a pneumatic output range of 3 to 15 psi with a 9 psi center. The electrical signal required was ± 2.5 milliamps. These transducers are specially designed for good frequency response.

The desired electrical signals were generated on the analog computer. Use of this computer made it possible to develop controllers to overcome the sluggishness and hysteresis of the control valves.

The flow rate of the oil was measured by a Waugh, type 6FLS, turbine flow meter located in the constant hot temperature stream and a Waugh, type F111, pulse rate converter with an output of 0-250 millivolts. This output was then amplified on the analog computer and was used as a measure of the flow rate and as an input to the valve position controller (Figure 5-2).

It was necessary to have the same sort of controller for the coolant flow system (Figure 5-3). However, because of equipment availability, the flow rate measurement was quite different. The measurement here was the pressure drop

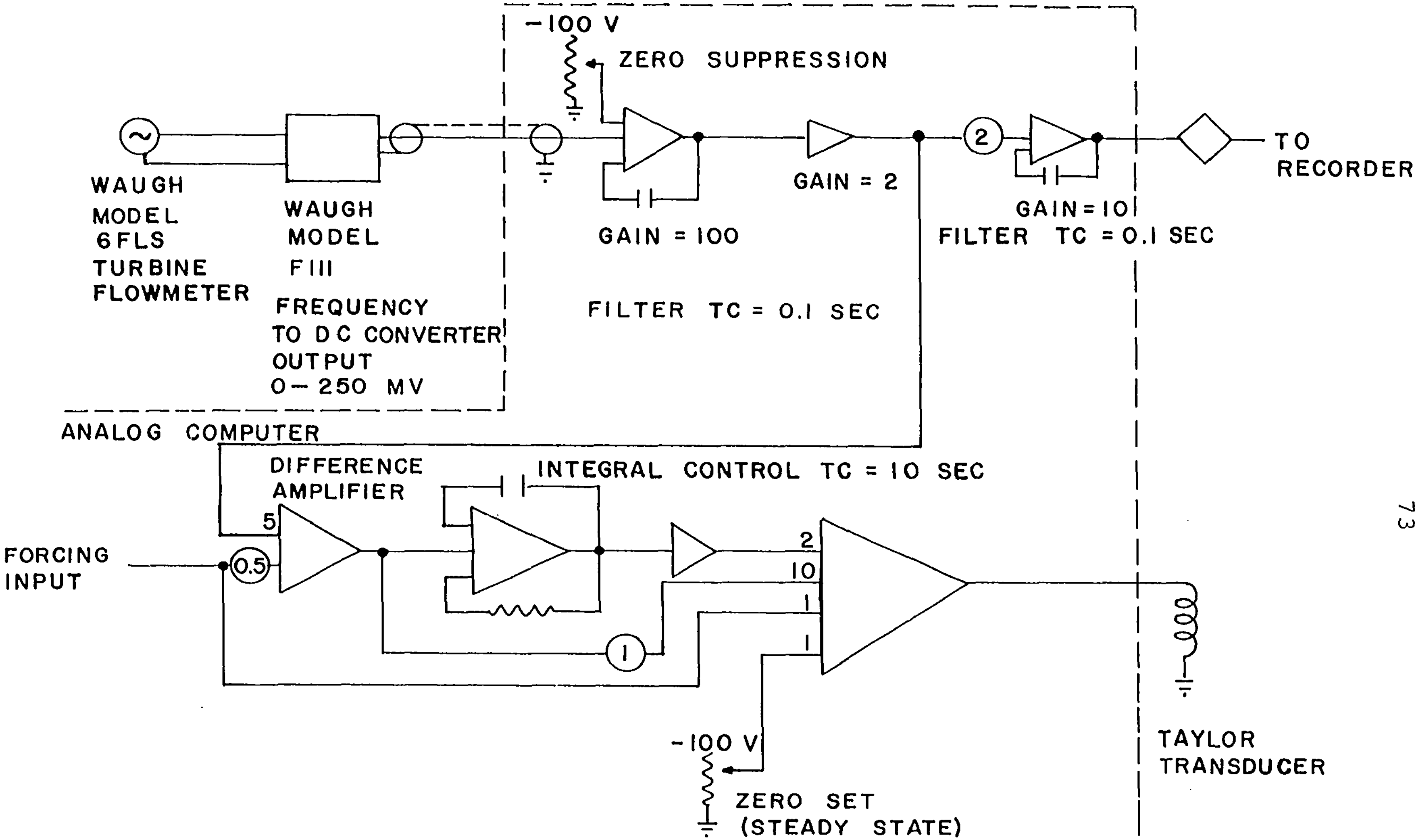


Figure 5-2. Oil Flow Control System.

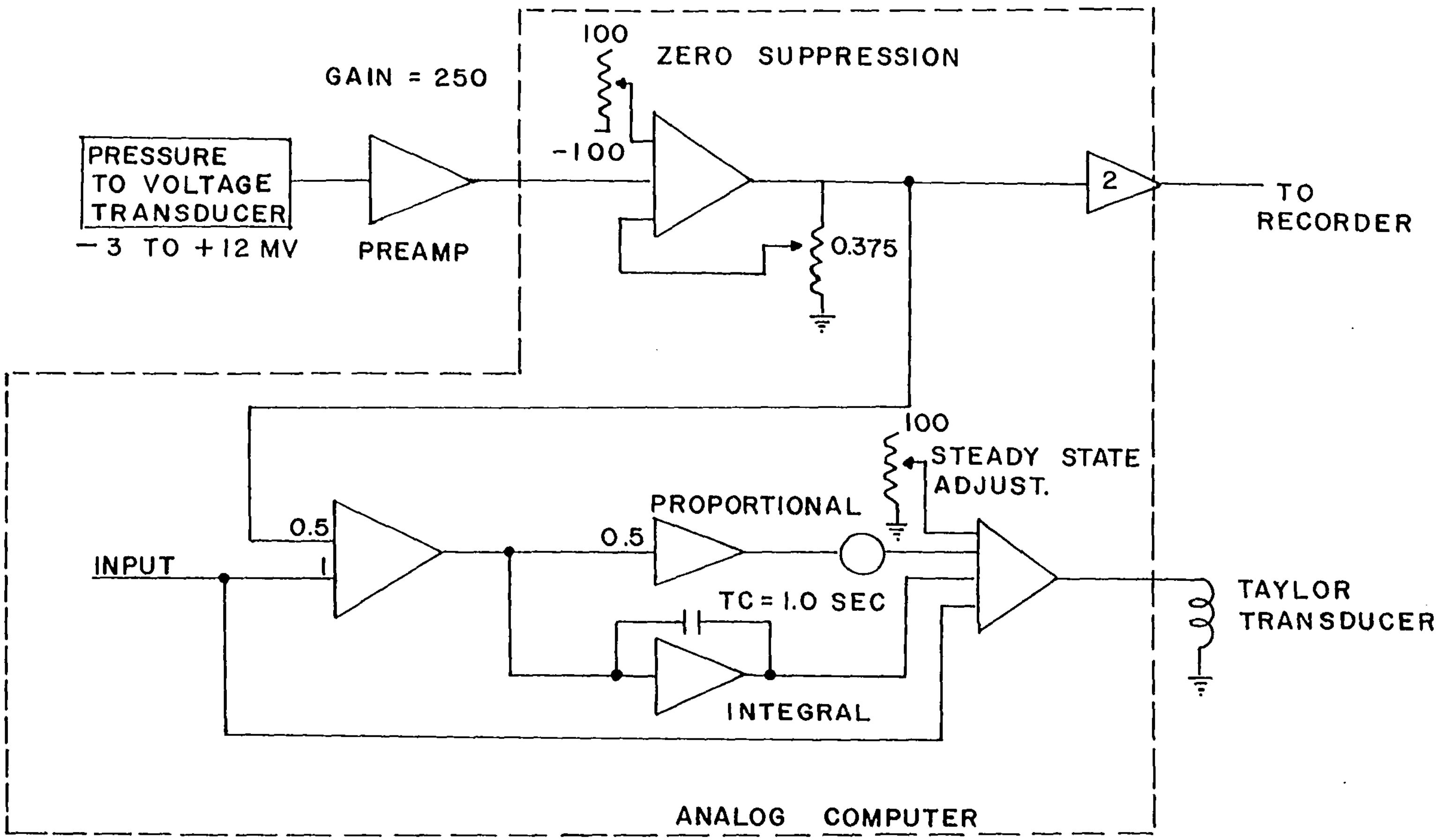


Figure 5-3. Coolant Flow Control System.

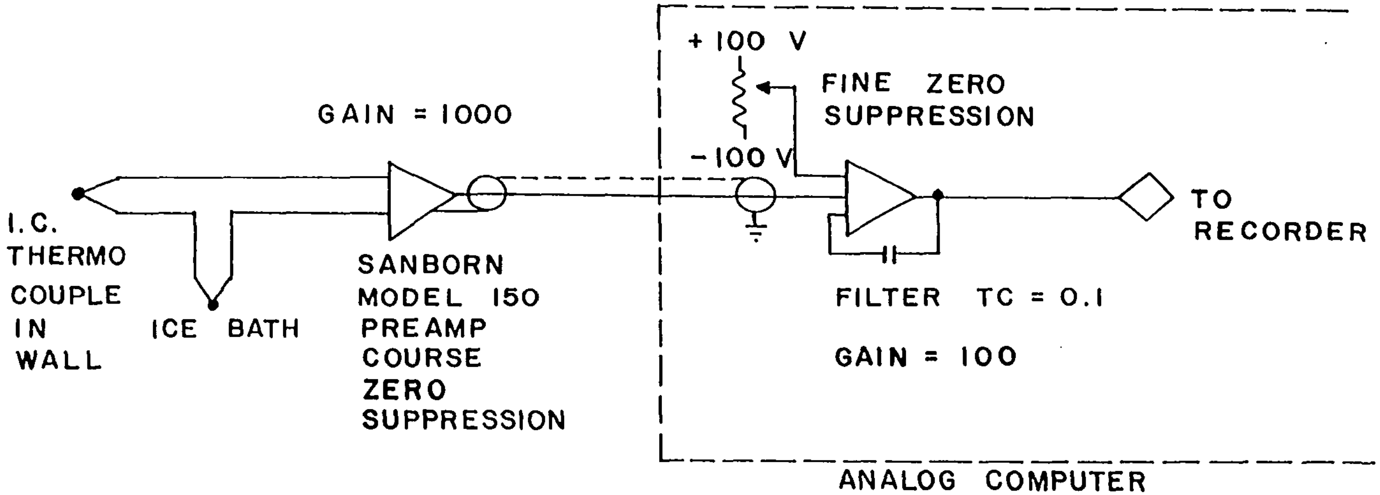


Figure 5-4. Wall Temperature Measurement System.

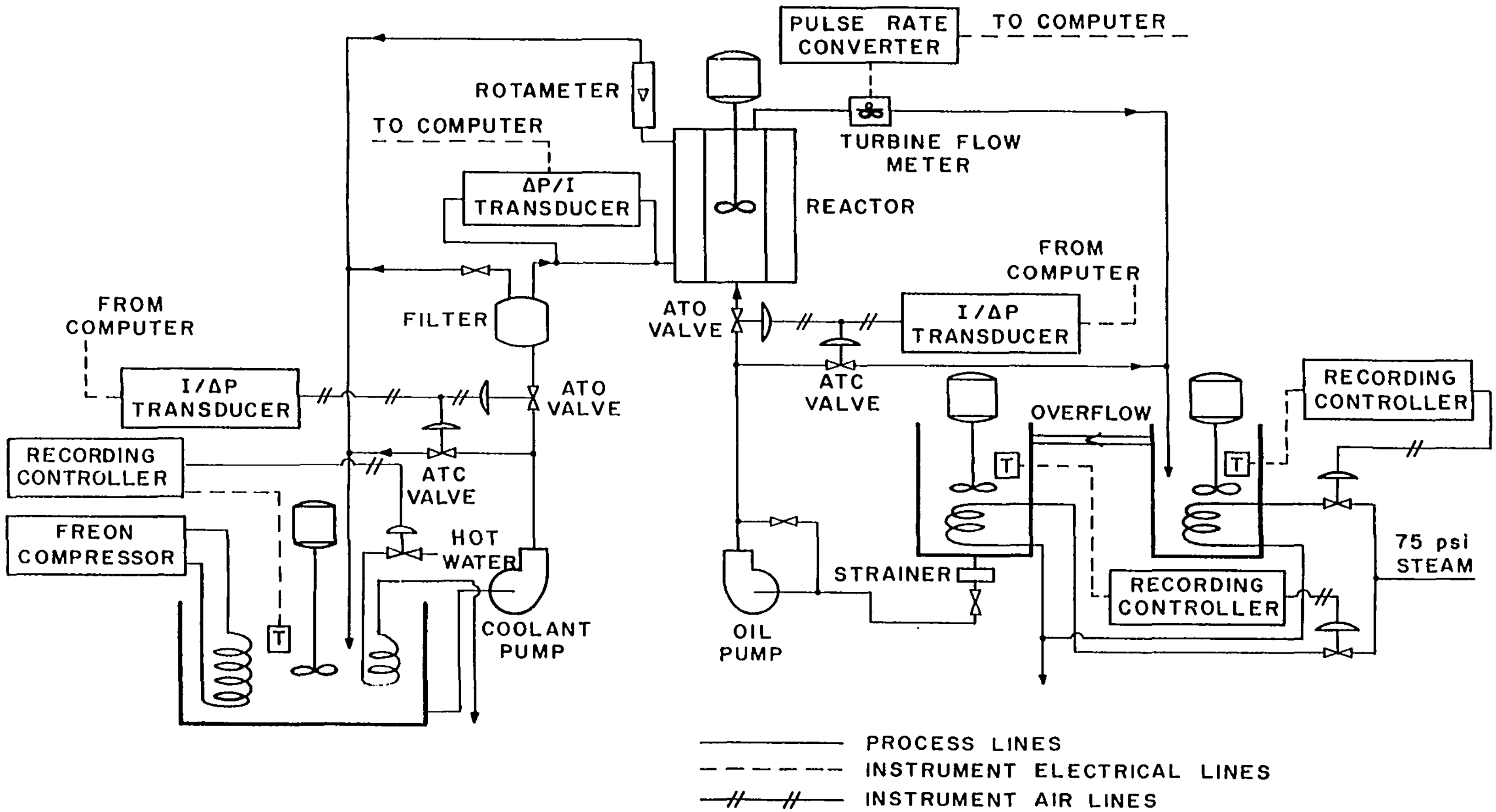


Figure 5-5. Schematic Flow Sheet of Experimental System.

across a length of capillary tubing, which was used instead of an orifice because of the linear relation between flow rate and differential pressure. The capillary consisted of the annulus formed by inserting a 10 gauge, vinyl-insulated wire (O.D. = 0.165) through a 10 foot length of Imperial polyflo tubing (I.D. = 0.190). This capillary was located in the coolant feed stream where the temperature was constant to eliminate errors caused by viscosity or density changes. This location required the measurement of both the upstream and downstream pressures. These measurements were accomplished by use of a strain-gage type, differential pressure transducer obtained from a Beckmann, model 112, Data Logger. The supply voltage of 7.000 volts was also obtained from this data-logger. The output of the transducers was -3 to 12 millivolts for 0 to 15 psig with this supply voltage. These millivolt signals were then amplified differentially using a Sanborn model 350-1500 preamplifier. This amplified signal representing the flow rate was transmitted to the analog computer for use in a feedback controller similar to the one used for the oil flow rate (Figure 5-3).

The controllers for both the oil and coolant flow rates consisted of a proportional controller to overcome the friction and a small amount of integral control to correct for variations from the set point (hysteresis). The time constants of these integral controllers were about one twentieth the size of the smallest time constant of the system.

Because of the high frequency noise developed in the transducers, the thermocouples and their transmission lines, it was necessary to provide filtering for these signals. These filters were programmed on the analog computer and were simply exponential dampers. The time constants of these exponential filters were several orders of magnitude less than the time constants of the system so that they would not interfere, or even be noticed in the identification.

Support Equipment

The basic support equipment consisted of the analog computer, its associated panel board and transmission lines, and the analog to digital conversion and recording equipment located in one room. The analog computer, panel board, transmission lines, etc. have been described in detail by Bishop and Sims (B3). The analog computer is a Donner, Model 3400, thirty-amplifier, ±100V computer with six electronic multipliers. The computer has been modified (B2) to accommodate thirty additional amplifiers, four, variable-base diode function generators, two transport delay generators, and a quarter square multiplier.

The panel board, located in an adjacent room containing the process equipment, includes the various temperature recorder controllers, transducers, and process equipment power control switches. Also on the panel board are the terminations of coaxial cables which are used to transmit low level signals between the process equipment room and the analog computer room. The other ends of these lines terminate within the analog computer programming area, making it easy to patch into the computer, signals from the remote equipment.

The analog to digital converter made it possible to process the large volumes of data that were taken. The converter consisted of a Dymec, Model 42900B, Input Scanner; a Dymec, Model 3440A, Digital Voltmeter with a Model 3443A, Range Unit; a Hewlet Packard, Model V562A, line printer; a Dymec, Model 2540, coupler; and a Friden, Model SP2, paper tape punch. With this equipment it was possible to scan the input and response signals and to record them at the rate of one point per second on the paper tape. It was possible to analyze these data directly with the Osage High Speed computer using a special read-paper-tape program.

Theoretical Description of the Experiment Process

A mathematical model of the heat transfer process is obtained from energy balances on the oil, the coolant, and the wall. The following assumptions are used to simplify the balances.

1. The fluid within the reactor vessel is all at the same temperature, i.e., perfect mixing.
2. The oil inlet temperature is constant.
3. The coolant inlet temperature is constant.
4. Densities, heat capacities, volumes, heat transfer coefficients and areas are constant.
5. The heat transfer to the coolant occurs at the mean temperature of the coolant; $T_{cm} = (T_{co} + T_{ci})/2$.

With these assumptions the following equations are obtained from the energy balances:

$$(\rho VC_p)_f \dot{T}_f^* = h_i A_i T_w^* - h_i A_i T_f^* + C_{pf} T_{in} W^* - C_{pf} W^* T_f^* \quad \text{(a)}$$

$$(\rho VC_p)_w \dot{T}_w^* = h_i A_i T_f^* + \frac{h_o A_o}{2} T_{co}^* - (h_i A_i + h_o A_o) T_w^* \quad \text{(b)} \qquad (5\text{-}1)$$

$$(\rho VC_p)_c \dot{T}_{co}^* = 2h_o A_o T_w^* - h_o A_o T_w^* + 2C_{pc} T_{ci} W_c^* - 2C_{pc} W_c T_{co}^* - h_o A_o T_{ci} \quad \text{(c)}$$

where ρ = density
V = volume
C_p = specific heat
T^* = temperature
h = heat transfer coefficient
A = heat transfer area
W = flow rate
Q = heat loss

subscripts f = oil
w = wall
c = coolant
i = inside
o = outside
in = oil in
ci = coolant in
ss = steady state

superscript * = total variable

Equations (5-1) are the non-linear, total variable model of the system. For convenience these equations can be changed to the perturbation model by assuming that each of the variables is composed of a transient and steady-state portion. Thus

$$T_f^* = T_f + T_{fss}^* \quad (a) \qquad W^* = W + W_{ss}^* \quad (d)$$

$$T_w^* = T_w + T_{wss}^* \quad (b) \qquad W_c^* = W_c + W_{css}^* \quad (e) \qquad (5\text{-}2)$$

$$T_{co}^* = T_{co} + T_{coss}^* \quad (c)$$

When Equations (5-2) are substituted into Equations (5-1) and the steady state equations corresponding to Equations (5-1) (the time derivatives equal to zero) are subtracted from these substituted equations, the pertubation model results:

$$(\rho VC_p)_f = -(h_iA_i + C_{pf}W_{ss}^*)\ T_f + h_iA_iT_w + (C_{pf}\ T_{in} - C_{pf}\ T_{fss}^*)$$
$$W - C_{pf}\ WT_f \qquad (a)$$

$$(\rho VC_p)_w\dot{T}_w = h_iA_iT_f - (h_iA_i + h_oA_o)T_w + \frac{h_oA_o}{2}\ T_{co} \qquad (b) \qquad (5\text{-}3)$$

$$(\rho VC_p)_c\dot{T}_w = 2h_oA_oT_w - (h_oA_o + 2C_{pc}W_{css}^*)T_{co} - 2C_{pc}(T_{coss}^* - T_{ci})$$
$$- 2C_{pc}W_cT_w\ . \qquad (c)$$

These pertubation equations contain product-type non-linearities involving the flow rates and their respective temperatures. A linear model is obtained by neglecting the product of these terms, which was shown by Stewart (S3) as being equivalent to using a Taylor Series expansion and retaining the constant and first order terms.

When the values from Table 5-1 are substituted into Equation (5-3) the resultant non-linear model for these specific operating conditions are

TABLE 5-1

LIST OF SYSTEM CONSTANTS

Symbol	Nomenclature	Value	Units	Source
C_{pc}	coolant heat capacity	0.823	BTU/lb°F	1
C_{pw}	wall heat capacity	0.037	BTU/lb°F	1
C_{pf}	oil heat capacity	0.405	BTU/lb°F	2
h_i	oil side heat transfer coefficient	0.0078	BTU/sec°F ft^2	3
h_o	coolant side heat transfer coefficient	0.0166	BTU/sec°F ft^2	3
A_i	inside heat transfer area	0.322	ft^2	2
A_o	outside heat transfer area	0.444	ft^2	2
T_{ci}	coolant inlet temperature	46.8	°F	3
T_{coss}	coolant steady state outlet temperature	71.5	°F	3
T_{fss}	oil steady state temperature	146.8	°F	3
T_{in}	oil inlet temperature	156.2	°F	3
T_{wss}	steady state wall temperature	73.5	°F	3
V_c	coolant volume	0.0102	ft^3	2
V_f	oil volume	0.0170	ft^3	2
V_w	wall volume	0.0153	ft^3	2
W_{ss}	steady state oil flow	0.0467	lb/sec	3
W_{css}	steady state coolant flow	0.00667	lb/sec	3
ρ_c	coolant density	65.4	lb/ft^3	2
ρ_f	oil density	53.0	lb/ft^3	2
ρ_w	wall density	640.5	lb/ft^3	2
Q_L	heat loss	0.035	BTU/sec	3

Sources: 1. Handbook; 2. Laboratory measurements; 3. Steady state data.

$$\dot{T}_c = -\,0.0583\ T_f + 0.00688\ T_w + 10.54\ W - 1.11\ WT_f \qquad \text{(a)}$$

$$\dot{T}_w = 0.00693\ T_c - 0.0273\ T_w + 0.0102\ T_{co} \qquad \text{(b)} \qquad (5\text{-}4)$$

$$\dot{T}_{co} = 0.0268\ T_w - 0.0334\ T_{co} - 74.05\ W_c - 3.00\ W_c\ T_{co}. \qquad \text{(c)}$$

The linearized process model is then

$$\dot{T}_f = -0.0583\ T_f + 0.00688T_w + 10.34\ W \qquad \text{(a)}$$

$$\dot{T}_w = 0.00693T_f - 0.0273T_w + 0.0102T_{co} \qquad \text{(b)} \qquad (5\text{-}5)$$

$$\dot{T}_{co} = 0.0268T_w - 0.0334T_{co} - 74.05W_c\,.. \qquad \text{(c)}$$

When the wall temperature is the response variable, the transfer function corresponding to the linear model is:

$$T_w = \frac{0.073\ (s+0.0334)W - 0.752(s+0.0587)W_c}{(s+0.0129)(s+0.0458)(s+0.0607)}. \qquad (5\text{-}6)$$

This equation represents the transfer function for which the time domain identification technique will be testing.

Experimental Determination of the System Characteristics by Standard Methods

To evaluate the performance of the identification technique, which is the basis of this work, two other identification procedures were used also: frequency response testing and pulse testing. Transient response tests were also conducted for an independent determination of the system gains. The individual techniques and their ramifications and results will be discussed first, and a general comparison will be made along with pertinent conclusions.

Transient Response Tests

This technique is called transient response testing because the experimental system is forced away from the steady state and is allowed to come to a new steady state. However, no data were taken on the actual transient portion of the tests, which served only as a convenient means for determining DC gains of the system for comparison purposes. The results of these tests are listed in Table 5-2.

TABLE 5-2

DC GAIN OF EXPERIMENTAL EQUIPMENT AS DETERMINED BY TRANSIENT TESTS

Forcing	Direction	DC Gain °F/lb/min.
coolant	+	19.7
coolant	-	42.9
oil	+	1.35
oil	-	2.97

The data of these tests were recorded in two ways: the voltage output of the transducers and amplifier systems versus potentiometric readings of the temperatures and versus scale and stop watch measurements on the flow systems. These recordings provided calibration of the measurement systems.

Frequency Response Tests

Frequency response tests were conducted following the standard procedure of forcing the system with sinusoidal variations of many different frequencies. The frequency of the forcings ranged from 0.0118 cycles/minute to 17.65 cycles/ minute. The electrical signals for these forcing functions were generated on the analog computer by solving the equations

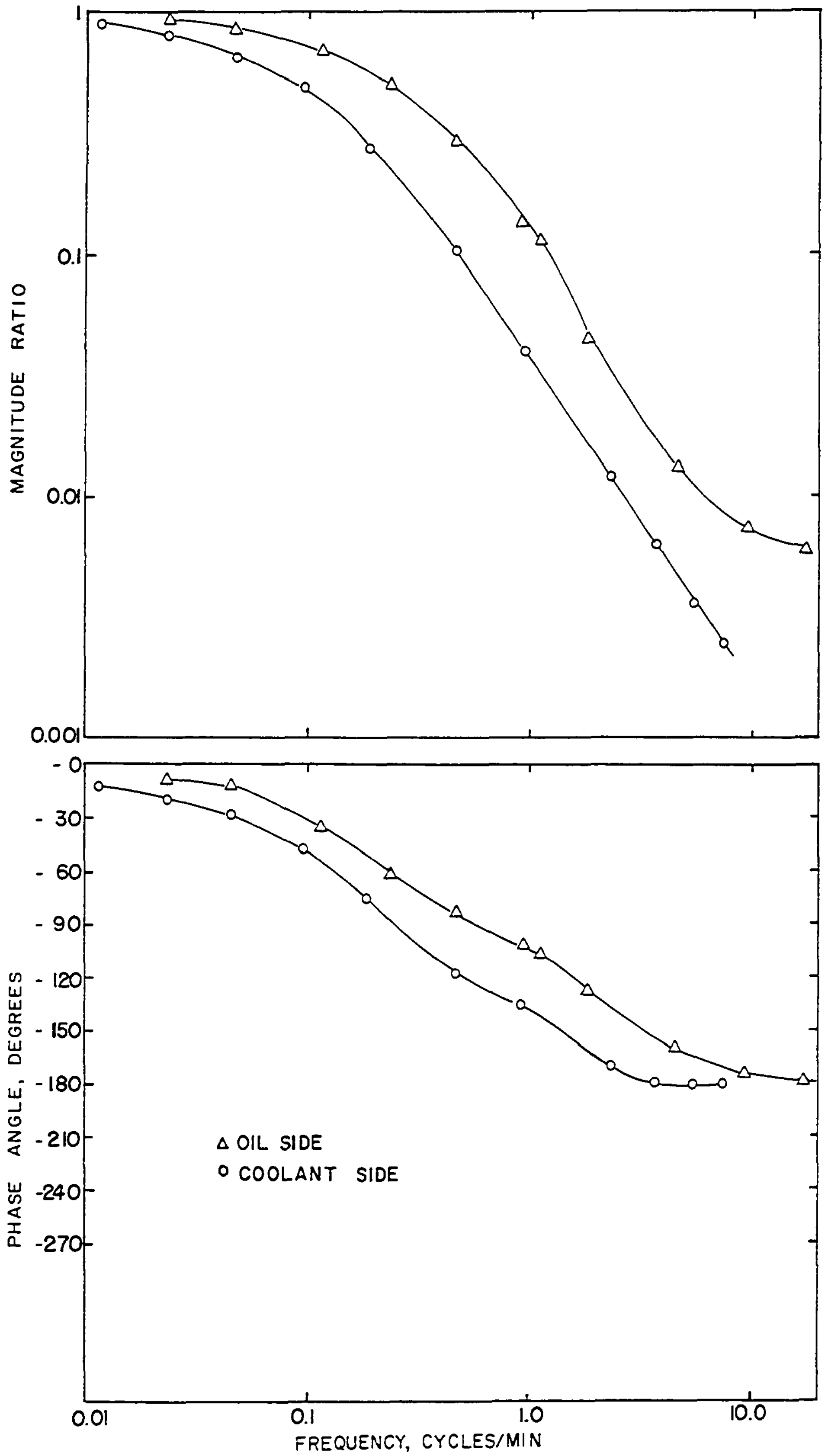
$$
\begin{aligned}
x &= -\omega \sin \omega t & \text{(a)} \\
y &= \omega \cos \omega t & \text{(b)} \qquad (5\text{-}7) \\
y(0) &= A \text{ (volts)} & \text{(c)}
\end{aligned}
$$

where x was the desired sine function and A was the desired amplitude.

These electrical signals from the analog computer were then converted to pneumatic signals by the Taylor transducers, and the flow rate was varied by the control valves in response to the pressure signal. The signals from the flow rate transducers and amplifiers were then recorded on the Sanborn 6-channel recorder. The amplitude and phase lag of the resonse function were then determined from these response records as a function of the forcing frequency. The results of the tests are given in the form of a standard Bode Plot in Figure 5-6 and Table 5-3.

The difficulties with these tests are many and varied, three of which will be listed here.

1. The length of time required to conduct the test is very great. The lowest frequency used in the tests required



5-6. Frequency Response Tests of Experimental System.

about ninety minutes for just one cycle, and extrapolation still was necessary to determine the DC gain.

2. The disturbances to the system were great. The variations do tend to average out, however, due to the plus-minus nature of the forcing. However, there is still a transient response associated with this forcing.

3. Results obtained in the form of the Bode Plot are limited in usefulness for further theoretical considerations.

TABLE 5-3

DC GAIN OF EXPERIMENTAL EQUIPMENT AS DETERMINED BY FREQUENCY RESPONSE TESTS

Forcing	DC gain °F/lb/min.
coolant	33.2
oil	2.36

Pulse Tests

Due to the product non-linearities associated with flow forcing of the reactor, pulse testing provides a means of estimating the extent to which these non-linearities affect the system description. This estimate is given as a result of forcing the flow in both directions from the steady state. The pulses used in the tests were sawtooth-shaped with an approximate duration of seventy five seconds. This magnitude represents a long pulse, far from an impulse. Because of the physical limitations imposed by the flow systems, the magnitude of the forcing pulse could not be made great enough to provide sufficient energy to the system in a shorter period of time. However, this long pulse did not disturb the system greatly. The wall temperatures changed about six degrees, or about eight percent of the steady state temperature.

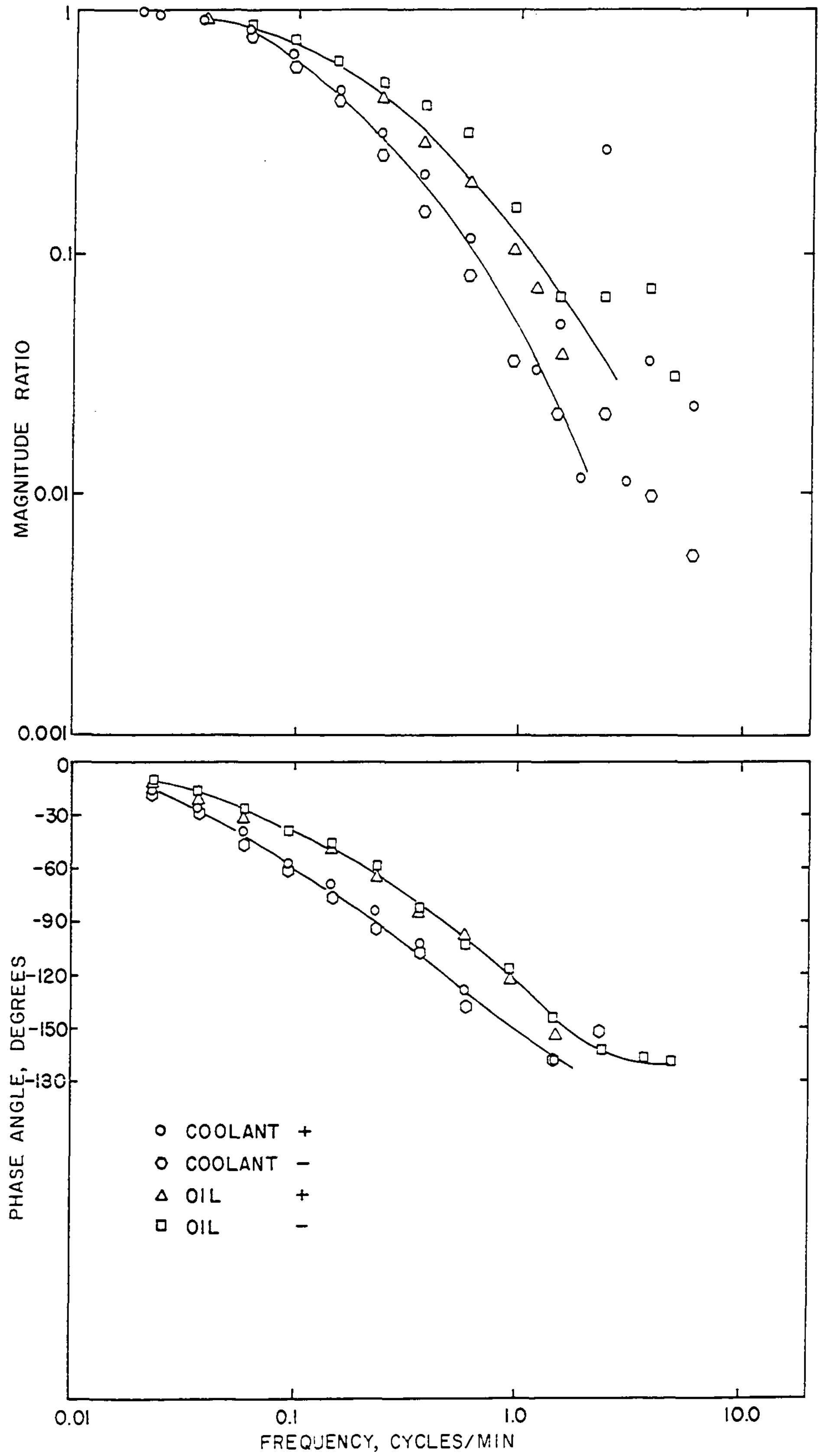
As can be seen from Figure 5-7, which is a standard Bode Plot, i.e., magnitude ratio and phase lag versus frequency, the non-linearity did not have a great effect on the system dynamics. However, there was a significant effect upon the DC gains of the system (Table 5-4) when both coolant and oil were forced. In both cases there was a factor of approximately 2 between the respective DC gains.

TABLE 5-4

DC GAIN OF EXPERIMENTAL EQUIPMENT AS DETERMINED BY PULSE TESTING

Forcing	Direction	DC Gain °F/lb/min.
coolant	+	22.8
coolant	-	39.8
oil	+	1.66
oil	-	2.98

The pulse testing technique seems to be very sensitive to pulse duration, pulse shape, noise, and amount of energy developed in the system. In the pulse tests conducted



5-7. Results of Pulse Testing of Experimental System.

on this system, it was impossible to obtain good information over more than one decade of magnitude in the Bode plot. This limitation is not intended to imply that pulse testing technique is not a valuable tool in general. Because the application of this technique is not the prime purpose of this work, essential refinements in the technique, such as smoothing the data and using a wide variety of forcing pulses, were not employed.

The computations involved in pulse testing are much more difficult than those of frequency response testing, but once a computer program is developed, the results are quickly obtained. Theoretically the identification can be made with only one short test. The computational procedures used in this work were presented by Dreifke and Hougen (D2) and were programmed by the author for the Osage computer.

Time-Domain Identification

The determination of a model for the experimental process was conducted in two stages. First the pole identification was carried out and then the major concern of this present work, the zero identification, was executed.

Pole Identification

The theoretical model of the process describes it as having three poles. In order to identify the system as third order, it is necessary to have three linearly independent responses of the unforced process (H2, p. 43). In an effort to obtain at least three independent responses, six tests were conducted using various combinations of the two flow streams to force the process from steady state. These combinations are listed in Table 5-5.

TABLE 5-5

FORCINGS USED FOR POLE IDENTIFICATION

Test Number	Forcing (Flow Rate) Oil	Coolant
1	increased	
2	decreased	
3		increased
4		decreased
5	increased	increased
6	decreased	decreased

With each of these forcings, the system was driven a small amount from steady state (a wall temperature change of about six degrees F). The forcing was then removed and the wall temperature recorded as it again approached steady state.

All twenty possible combinations of these six tests (taken three at a time) were used in the digital computations for the poles. Many of these combinations yielded no significant results. The precise reasons are now known, but there are several possibilities.

1. The tests were not linearly independent: A probable cause, but not conclusive, since the tests appear

to have different initial values, initial slopes, and initial rates of change of slope which would indicate they were linearly independent.

2. The noise level was too great: An improbable cause as an inspection of the recorder traces shows them to have a very low noise level (especially in the early portions of the tests where the energy levels were high) of no more than one or two percent. Heymann (H2) was able to obtain identifications with as much as eight percent noise.

3. The steady state conditions changed while the tests were being conducted: A most probable cause of trouble as there were basically four factors controlling the steady state. They include the oil flow rate, the coolant flow rate, the oil temperature and the coolant temperature. Every possible effort was made to control these factors as closely as possible, but there were limitations on the equipment available. In controlling the oil and coolant temperatures a deadband existed in the controllers, making it impossible to hold these temperatures any closer to the desired values than $\pm$ 0.25 degrees. Because of the measurement technique required by available equipment for the coolant flow, the pressure drop and the recorded voltage could be maintained constant, but the flow rate changed because of air bubbles adhering to the walls of the capillary.

All of these factors add up to what the author believes to be a rather unreliable steady state, and therefore the most probable cause of trouble. Heymann (H2, p.199) states, "Low frequency disturbances such as changes or drifts of steady state operation level cannot be tolerated for this identification." (He makes no other reference to this problem except for one other short qualitative statement as to the effect of steady state changes.) There exists no quantitative measure of this effect, so that it is impossible to tell the extent of the corruption of the identifications by the oil and coolant temperature changes, etc. It was possible, however, to obtain from many of the tests good identification results. The results of a mean and standard deviation analysis of the identifications are listed in Table 5-6.

TABLE 5-6

RESULTS OF POLE IDENTIFICATION

Pole	Mean Value	Standard Deviation
1	-0.0498	0.0098
2	-0.0182	0.0055
3	-0.00987	0.0024

This averaging was necessitated by the general instability of the identifications, and it is believed that they provide a reliable identification of the process dynamics in this range of operation.

The instability of the identifications can be seen in the plots of Figures (5-8) and (5-9). In both cases the high frequency pole is the most unstable, but this result does not seem unreasonable when compared with Figure 34 of Heymann (H2, p. 184) which shows that even under ideal computer conditions this pole is not identified as well as the others.

During the early parts of the identifications, the two lowest frequency poles are shown as being equal because they were identified as being complex. The complex part was very unstable, and consequently it was assumed that these poles were real and close to being equal. Heymann (H2, p. 167) found that when two poles were very nearly equal the identifications tended to oscillate between real and complex poles with the complex part being unstable.

In Figure 5-9 there is an abrupt change in the identification at delta = 61, which corresponds to time = 183 seconds. This identification is with tests 2, 4, and 6. A close look at the trace of response 4 showed a sudden shift in the response at about 180 seconds, which probably causes this instability in the identification.

To summarize the results of the pole identification, this author believes that a reliable identification has been made through the use of the mean of all twenty combinations of tests even though some instability exists within each test.

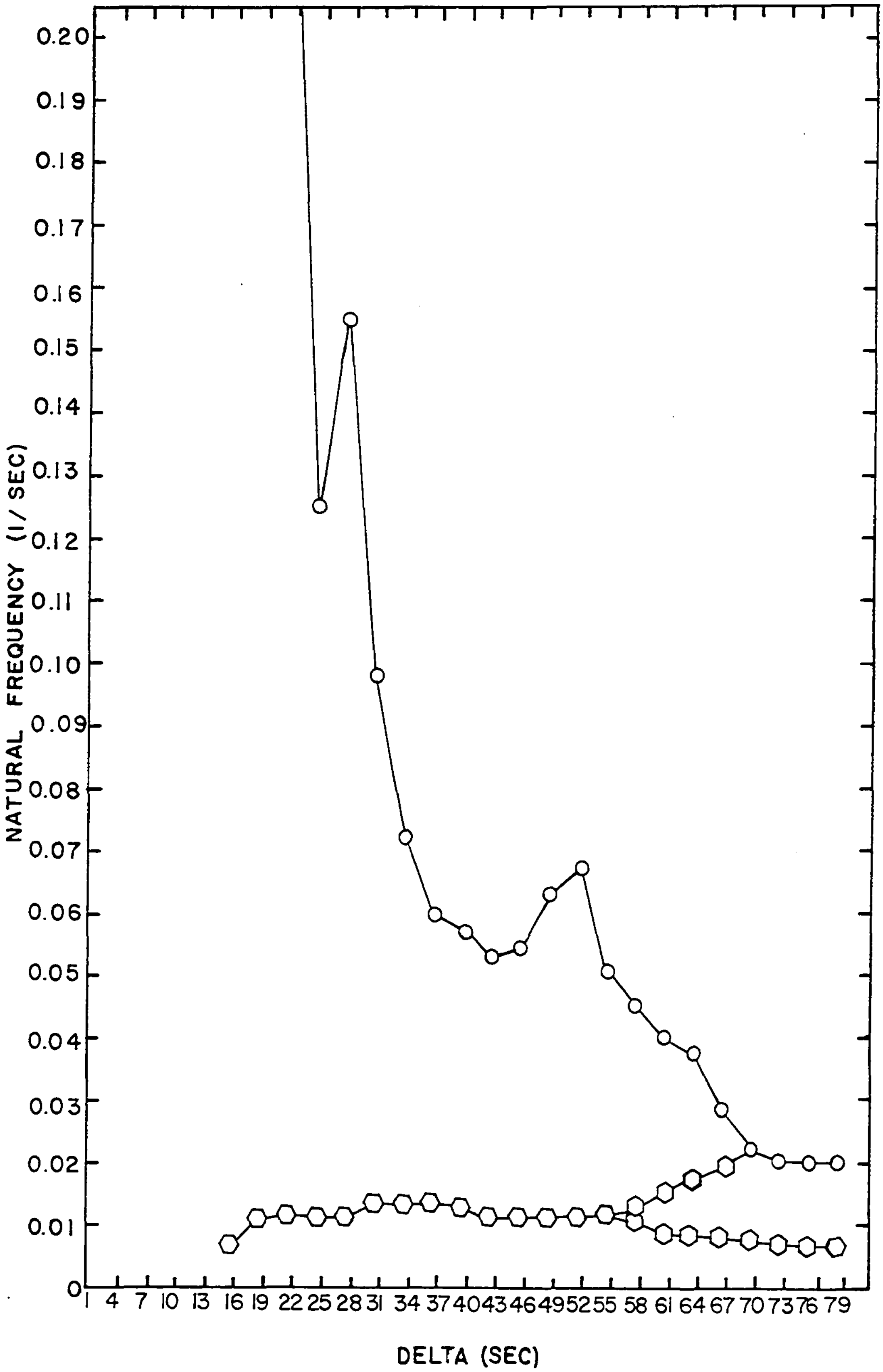


Figure 5-8. Identification of the Poles of the Rxperimental System, Tests 1, 3, 5.

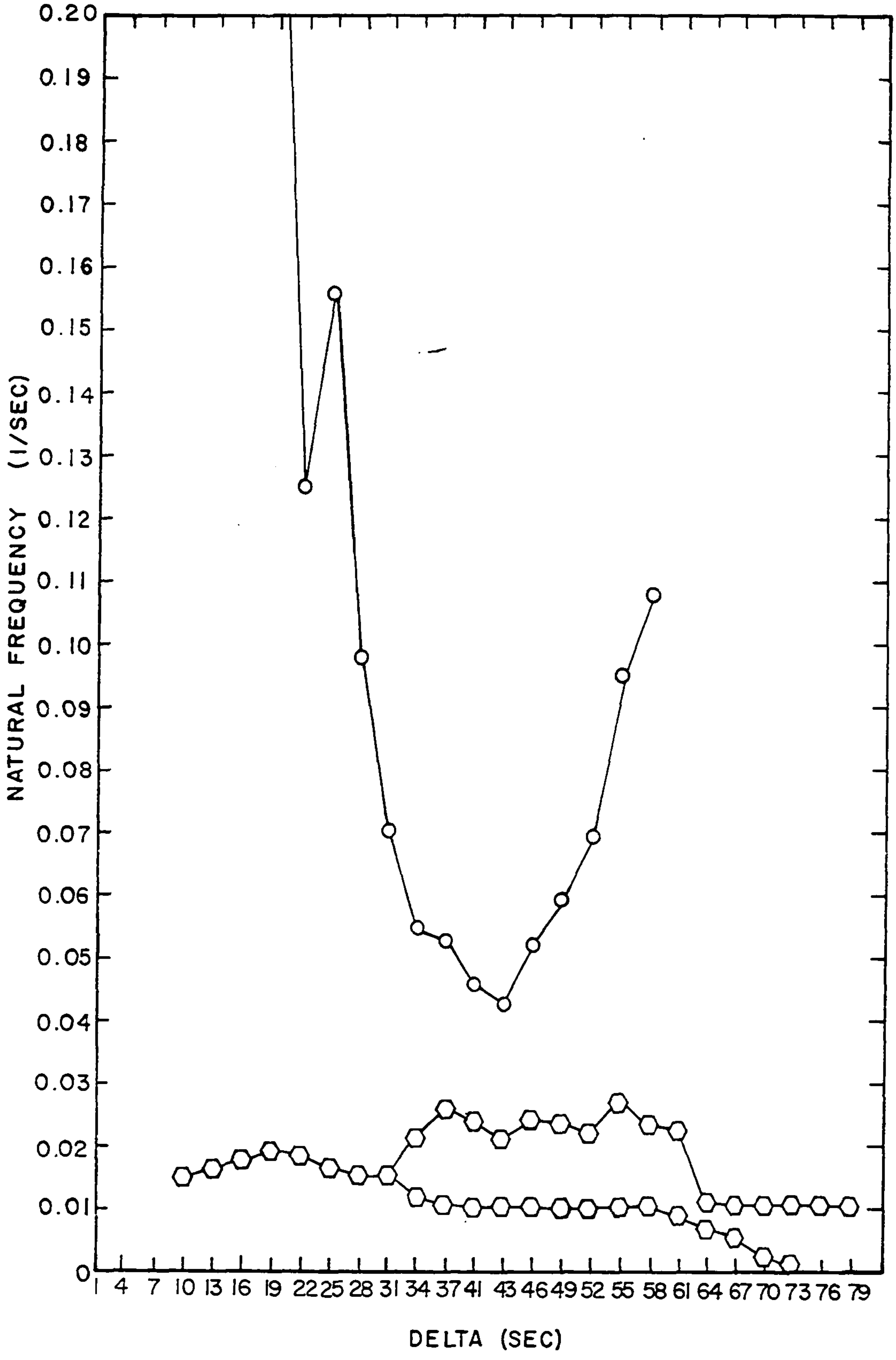


Figure 5-9. Identification of the Poles of the Experimental System, Tests 2, 4, 6.

Zero Identification

In all of the computer studies the zero identification technique worked very well and an identification was obtained, even when there was a quite large error in some of the factors. However, in all of these tests there was only one error, which had a known magnitude. In order to test the technique under more realistic circumstances, the experimental process was used in place of the computer. In the process there were a multiplicity of errors, the magnitude of which were unknown. These include all of the errors assumed in the computer studies: misidentification of poles, incorrect steady state, noise, transportation delay, non-linearities, etc., and probably some items which were not considered in the computer studies.

Here again, as with the pole tests, in order to assure a good selection of independent tests, seven tests were conducted with both the oil flow rate forcing and coolant flow rate forcing. Table 5-7 is a summary of these forcings.

Unlike the pole tests where the required information is the relaxed response for the zero test, the forced response is the desired information along with the forcing. The system was therefore forced from steady state by each of the forcings of Table 5-7, while the response and input were alternately recorded versus time as output of the analog to digital conversion equipment.

In the analog computer tests of the non-linear system the coolant forced process was much more non-linear than the

TABLE 5-7

FORCINGS USED FOR ZERO IDENTIFICATION

Test	Oil flow lb/sec	Coolant flow lb/sec
1	- 0.0125	+ 0.001
2	+ 0.0125	- 0.001
3	- 0.00025t	- 0.00001t
4	+ 0.00025t	+ 0.00001t
5	- 0.025 sin (0.091t)	- 0.002 sin (0.091t)
6	- 0.025 sin (0.050t)	- 0.002 sin (0.050t)
7	- 0.025 sin (0.010t)	- 0.002 sin (0.015t)

oil forced system. In the actual experimental process the opposite is true. If one assumes that the theoretical model of the experimental system given by Equations (5-4) is correct, then an idea of the degree of non-linearity can be garnered by lumping the non-linear terms with the respective temperatures as

$$\dot{T}_f = -(0.0583+1.11W)T_f + 0.00688T_w + 10.54\,W \quad \text{(a)}$$

$$\dot{T}_{co} = 0.0268T_w - (0.0334 + 3.00\,W_c)T_{co} - 74.05\,W_c \quad \text{(b)} \qquad (5\text{-}8)$$

Substituting the values of the step forcing into these lumped terms,

$$0.0583 + 1.11W = 0.0583 \pm 0.0139 \quad \text{(a)}$$

$$0.0334 + 3.00W_c = 0.0334 \pm 0.003 \quad \text{(b)} \qquad (5\text{-}9)$$

It can be seen that this term for oil forcing will vary ± 24-percent, and the coolant forced term will vary ± 9-percent, thus indicating that the oil is almost 3 times more non-linear than the coolant side. On the basis of this information, the coolant side identification should be better than the oil side. The truth of this prediction will be seen as the results are discussed.

Nineteen combinations of these tests were used in the identification of each transfer function. Table 5-8 summarizes these combinations.

Table 5-9 gives a summary of the mean values of the identified gains and zeros for the nineteen combinations of tests with both the oil and coolant forced systems. These values are all within about ± 50-percent of the overall mean value and approximately half of them are within ± 10-percent of the overall mean value.

A close inspection of some of the extreme cases will show why their individual mean values should be eliminated from the overall, or at least modified. One good example is in the case of sets eleven and fourteen, where the forcings were ramps and a very low frequency sine wave. As was determined in the computer studies of linear systems, this condition did not produce good results (page 25). The early part of the sine forcing appears very similar to a ramp, and therefore the linear independence of these tests is tenuous. In practice these identifications were poor with a large drift and slow convergence. In the case of set fourteen of the oil system, no identification at all was made for the gain of the system. These tests should therefore be eliminated from the overall identification.

Sets eighteen and nineteen of the coolant tests, which have the extreme values of the gain are valid in that taken as a pair, they represent the worse cases of step and ramp forcing, but their identified results when averaged give the linearlized model.

Sets eleven and fourteen are the only ones for which it seems reasonable to eliminate the mean values. The corrected overall mean values are:

$$\text{oil gain} = 5.47 \times 10^{-3}$$

$$\text{oil zero} = 3.75 \times 10^{-5}$$

$$\text{coolant gain} = 4.08 \times 10^{-3}$$

$$\text{coolant zero} = 5.88 \times 10^{-5}.$$

Figure 5-10 shows the results of a pair of identifications of the coolant system. For a negative step and negative ramp the gain and zero curves have positive slopes, while for positive forcings the slopes are negative. It is possible to see that these curves bound the overall mean value with the gain curves crossing at almost the exact value of the mean. This result would indicate that the mean value is a very good linear identification for this pair of tests.

In the computer analysis of a non-linear system, when step and sine forcings were used, the identification divided into distinct groups bounding the linear system. The same is

TABLE 5-8

TEST COMBINATIONS USED FOR ZERO IDENTIFICATION

	Set Numbers																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Oil																			
step -	x	x	x	x														x	
step +					x	x	x	x											x
ramp -	x								x	x	x								x
ramp +					x							x	x	x				x	
sin (0.091t)		x				x			x			x			x	x			
sin (0.05t)			x				x			x			x		x		x		
sin (0.01t)				x				x			x			x		x	x		
Coolant																			
step +	x	x	x	x														x	
step -					x	x	x	x											x
ramp -					x				x	x	x							x	
ramp +	x											x	x	x					x
sin (0.091t)		x				x			x			x			x	x			
sin (0.05t)			x				x			x			x		x		x		
sin (0.015t)				x				x			x			x		x	x		

TABLE 5-9

RESULTS OF ZERO IDENTIFICATION OF EXPERIMENT EQUIPMENT

	Oil Forced System		Coolant Forced System	
	Gain x 10^3	Zero x 10^5	Gain x 10^3	Zero x 10^5
1	5.55	3.66	4.64	3.86
2	5.62	3.80	3.41	6.69
3	5.53	3.20	3.60	4.25
4	4.54	2.61	5.79	7.10
5	7.05	3.77	4.14	7.19
6	5.84	4.05	3.97	5.07
7	5.24	3.32	4.02	6.45
8	3.41	3.98	5.34	6.17
9	5.93	3.88	3.00	6.98
10	5.71	3.75	3.92	6.08
11	4.92	2.44	3.67	9.67
12	6.14	4.67	2.52	5.21
13	6.16	4.46	3.81	4.96
14	*	5.25	4.68	9.60
15	5.29	5.17	3.98	6.97
16	5.39	4.02	4.21	7.01
17	5.29	3.27	4.30	7.21
18	6.73	4.34	6.33	1.91
19	4.54	3.98	2.48	7.95
Mean	5.45	3.78	4.09	6.28

*no identification obtained.

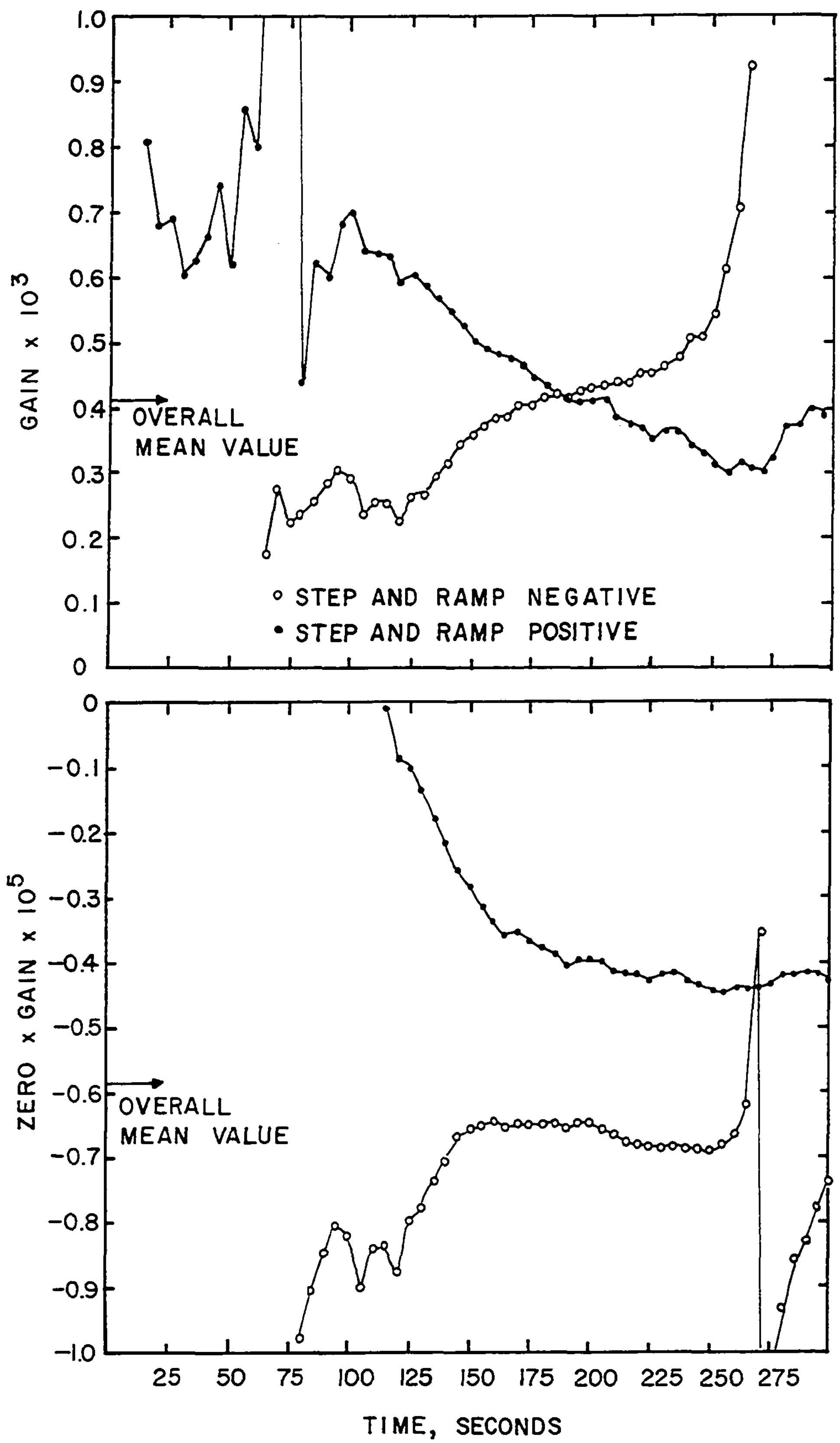


Figure 5-10. Identification of Coolant-forced, Experimental System with Step and Ramp Forcings.

true with the results from the experimental process, which can be seen in Figure 5-11. The overall mean values fit pretty well within the identification results, indicating that these mean values are also valid for these tests.

In the identification for the oil system it appeared that the ramp was the predominant test when the forcings were a ramp and a step. Therefore, Figure 5-12 shows the results of the pair of ramp forcings with a positive step. Although these tests do not show the clear cut opposition in the gain, as was noticed with the coolant system, the overall mean value does appear to be a good linear identification of the tests.

In Figure 5-13 the identification for positive step and sine forcings is plotted. Here again, as in Figure 5-12, there is not the distinct separation in the identification of the gain as was noticed with the coolant system. However, again the overall mean value does appear to be a good linear identification, in that the mean value of the 45 points which are plotted for the gain is 0.000524, plus the nine points which are off the graph, is 0.00101.

With this verification that these overall mean values comprise a good identification, they will be used to determine the actual model in terms of the wall temperature and flow rates. The actual zero is merely the identified zero divided by the gain. The values are given in Table 5-10. To determine the gain in terms of °F/lb/sec, it is necessary to multiply the identified gain, which is in terms of volts

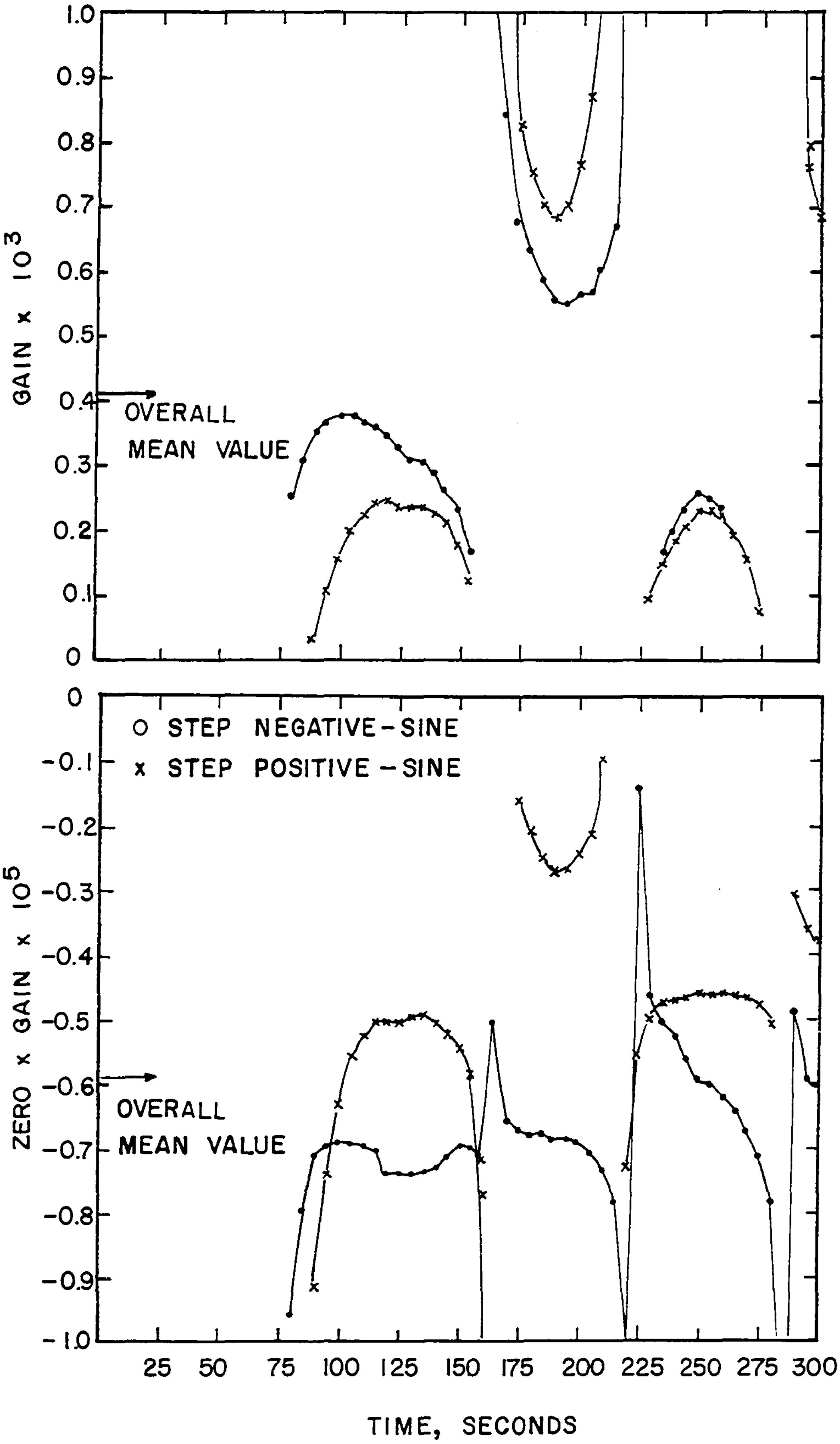


Figure 5-11. Identification of Coolant-forced, Experimental System with Step and Sinusoidal Forcings.

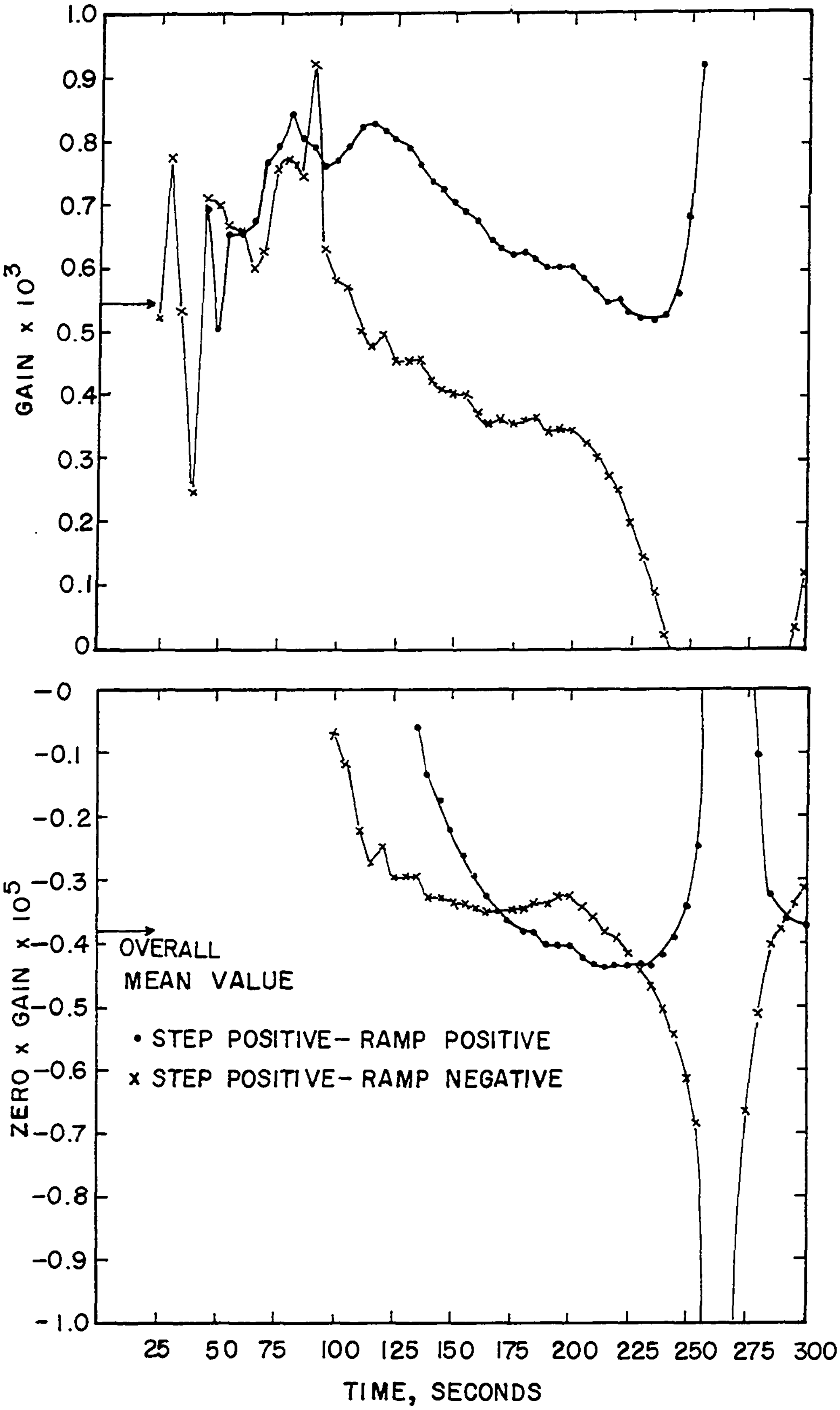


Figure 5-12. Identification of Oil-forced, Experimental System with Step and Ramp Forcing.

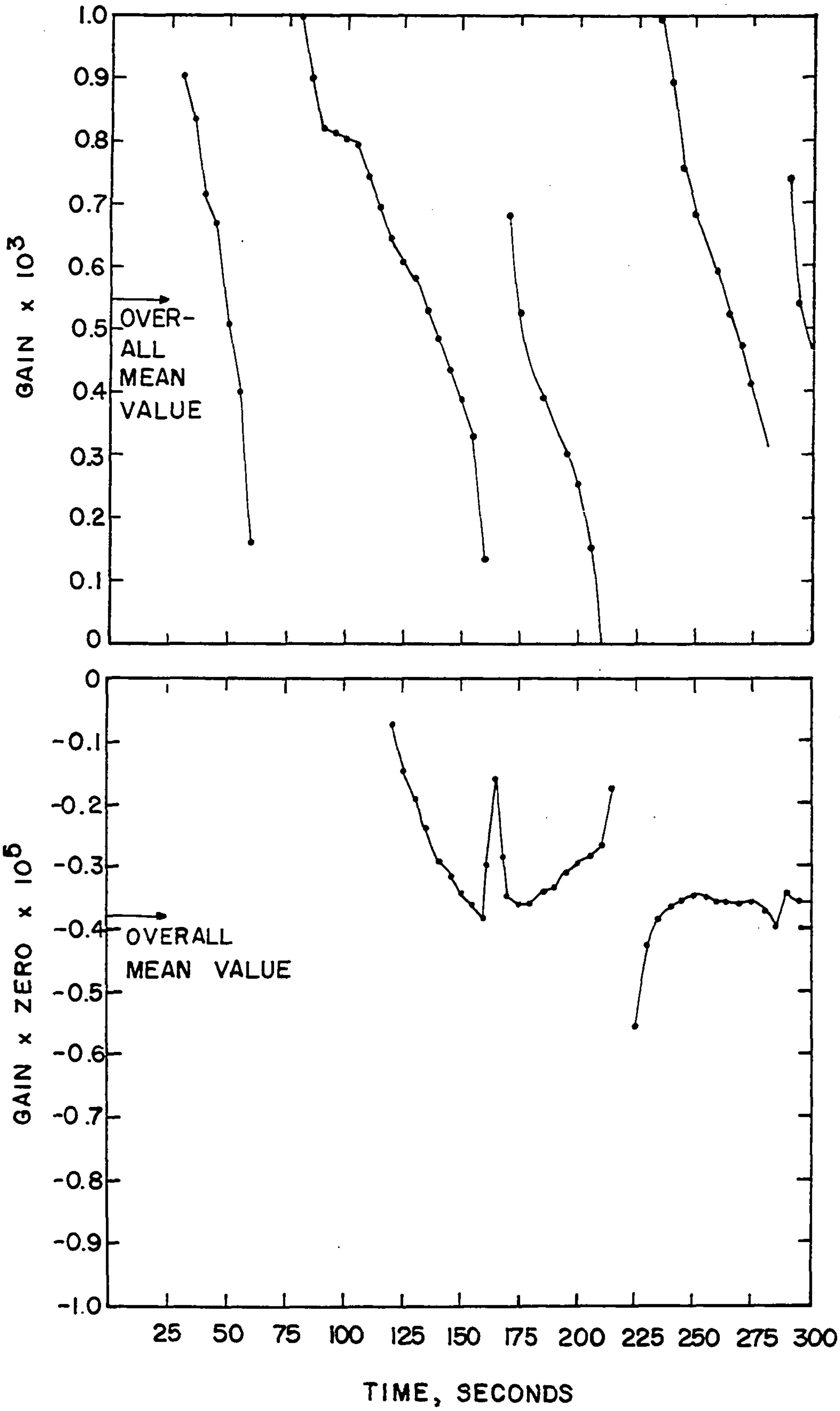


Figure 5-13. Identification of Oil-forced, Experimental System with Step and Sinusoidal Forcings.

temperature sec^2/volts flow by a factor of °F/$volt_{temp}$ x $volt_{flow}$/lb/sec.

For the oil system this factor is 333, and for the coolant system it is 2083, as determined by the measurement and amplification equipment. The results are given in Table 5-10.

TABLE 5-10

FINAL MEAN RESULTS OF ZERO IDENTIFICATION
OF EXPERIMENTAL EQUIPMENT

System	Gain	Zero
oil	0.181	- 0.00675
coolant	0.852	- 0.0154

The linearized transfer function as obtained by the pole and zero identification is then

$$T_W = \frac{0.181\ (s+0.00675)W - 0.852\ (s+0.0154)W_c}{(s+0.498)\ (s+0.0183)\ (s+0.00987)} \qquad (5\text{-}10)$$

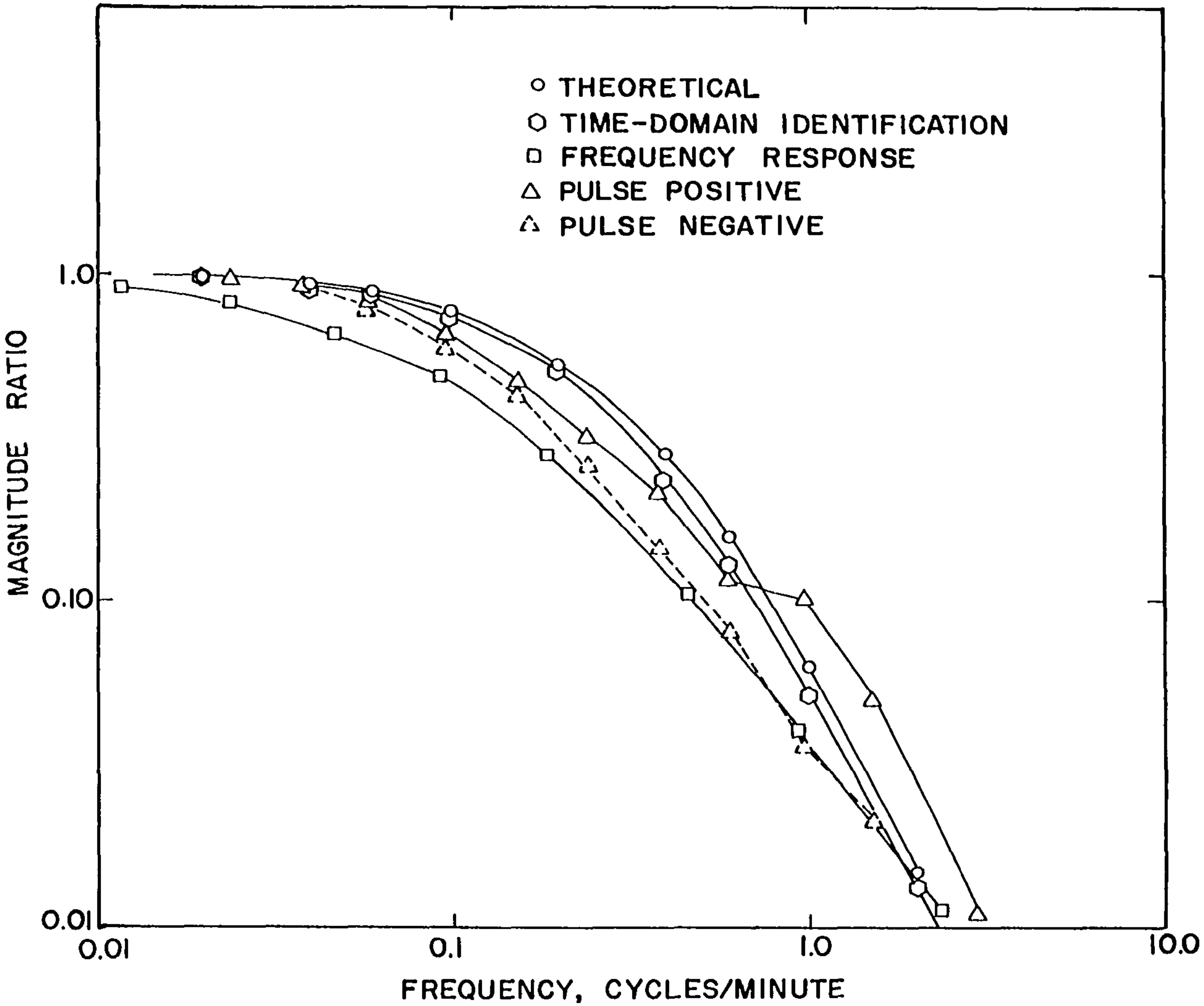
Comparison of Results

The Bode diagram is used to present the results of the three identification methods for comparison. This presentation was chosen because it is the final form of two of the identification techniques, and the time-domain results are easily converted to this form. The reverse is not nearly as easy or as accurate.

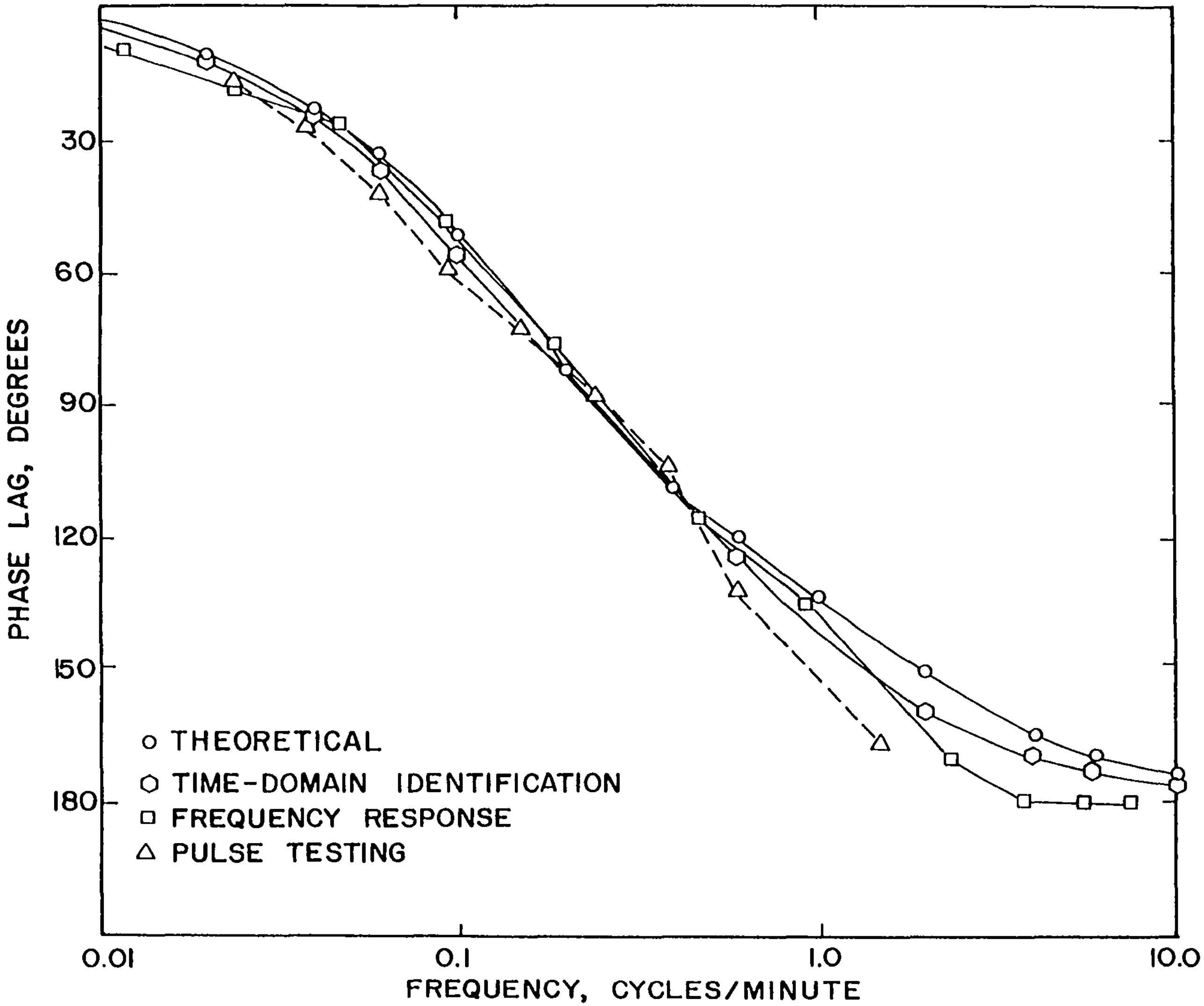
Figures 5-14 and 5-15 give the Bode diagram of the coolant forced system. These graphs show that the results of the time domain identification compare very favorably with the results of the other testing methods and the theoretical model. The Bode diagrams of the oil forced system (Figures 5-16 and 5-17) do not show this good comparison. Examination of the curve for the time domain identification reveals a peak in the magnitude ratio at approximately 0.1 cycles per minute. This peak is a result of the low natural frequency of the identified zero.

To obtain some idea of the quality of this identified system, analog computer tests were run to compare the identified system to the theoretical system and to actual data from the plant. The procedure used in these tests was to match first, as closely as possible, the actual plant data with the theoretical non-linear model. The response of the theoretical linear model was then obtained by removing the product term. The last step was to determine the response of the identified model. To compare the dynamics of the models closely, the amplitude of the response of the identified model was adjusted to equal the amplitude of the theoretical linear model over the first quarter cycle.

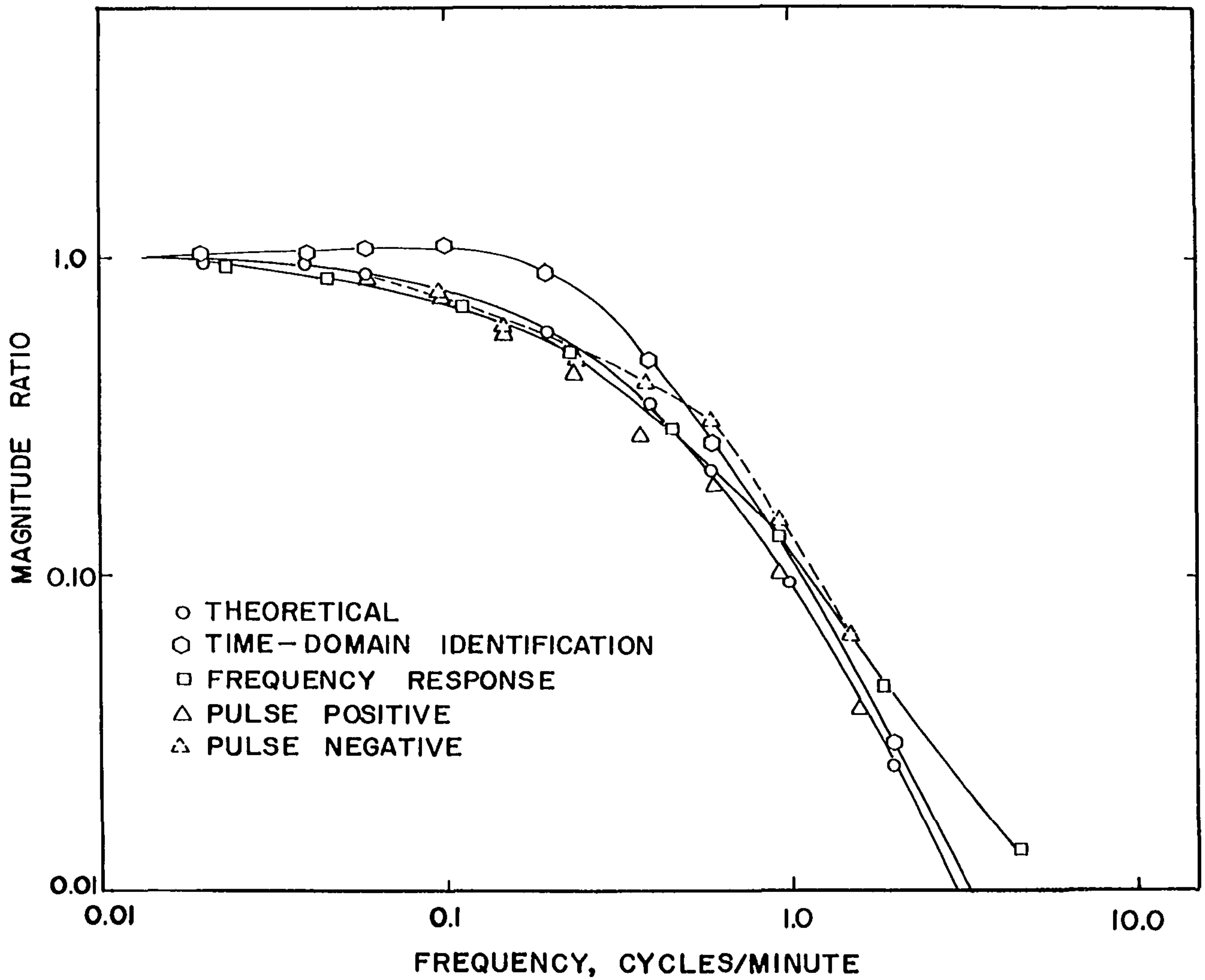
As can be seen in Figure 5-18, the identified model matches the actual data much better than does the theoretical linear model--almost as well as the non-linear model. The dissimilarity between the model in question and the models obtained by other means can therefore be attributed to an



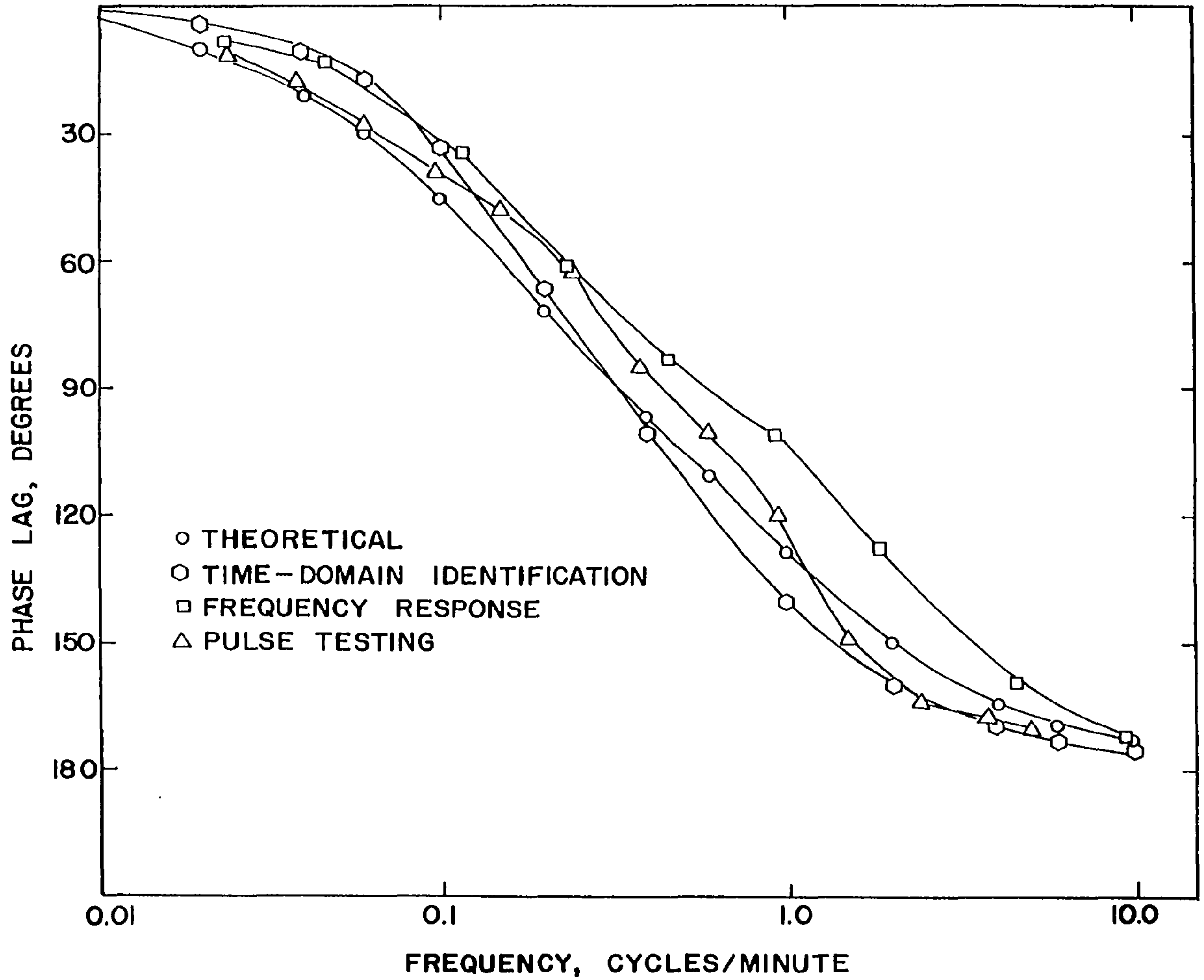
5-14. Bode Magnitude Plot for Comparison of Results of Identification of the Coolant-forced System.



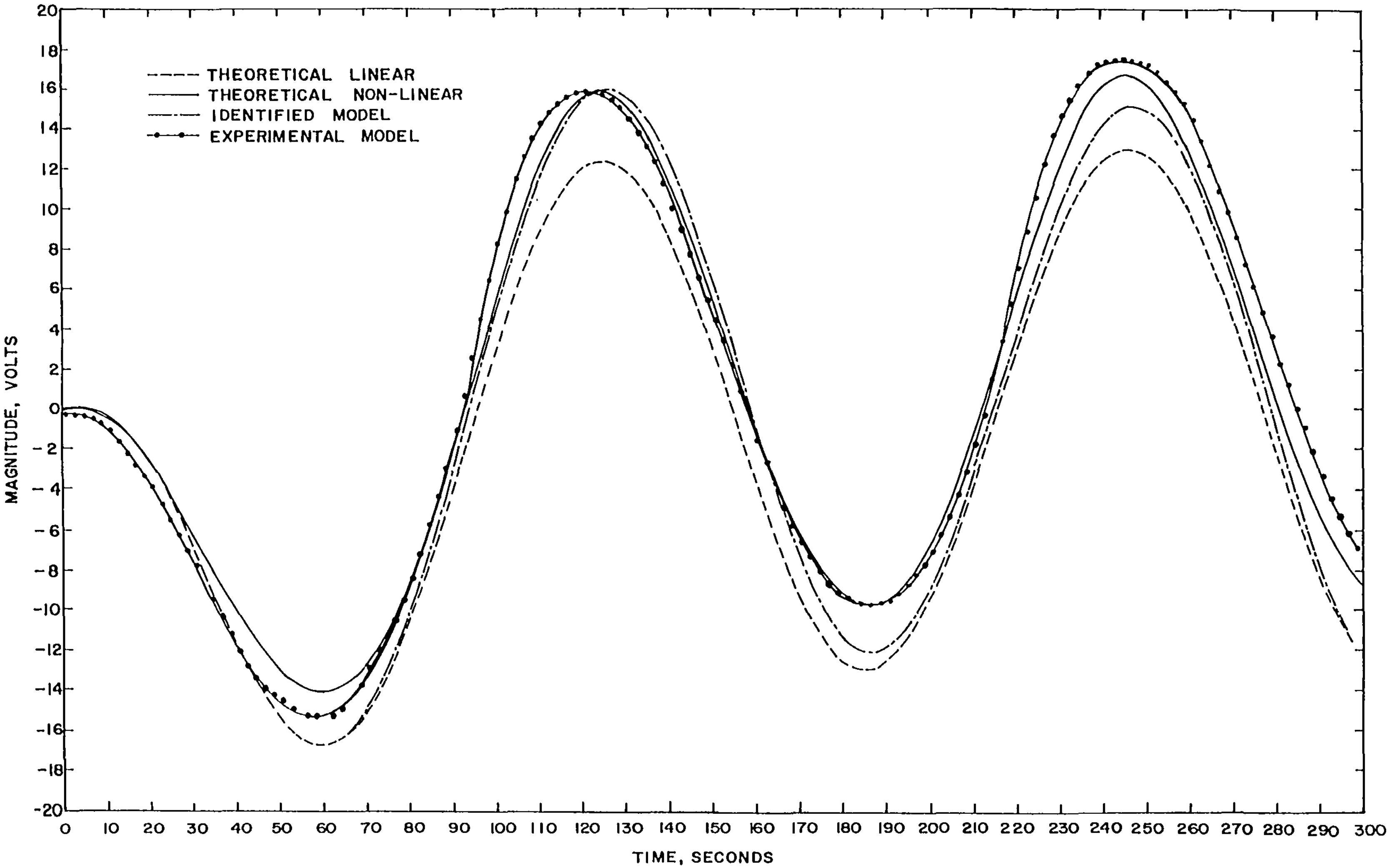
5-15. Phase Diagram of Results of Identification of the Coolant-forced System.



5-16. Bode Magnitude Plot for Comparison of Results of Identification of the Oil-forced System.



5-17. Phase Diagram of Results of Identification of the Oil-forced System.



5-18. Comparison of the Identified Model to Experimental Data for the Oil-forced System.

attempt by the time domain identification technique to approximate the dynamics of the non-linear process. For this reason it is assumed that the identified model is as good a linear approximation as can be determined.

The DC gains of the oil forced system as determined by all of the experimental methods are very similar if an average is used for the pulse and transient tests. The average of the DC gains of the frequency response, pulse, and transient tests is 2.28. The DC gain, as determined by the time domain identification, is 2.27, which appears to be a very reasonable value.

Although the DC gains of the coolant forced systems compare less favorably with the average of the results from the other experimental methods than the DC gain of the oil forced system does, the identified gain lies within the range of values experimentally determined for positive and negative forcings. The identified gain of 24.33 is therefore believed to be a valid result.

TABLE 5-11

SUMMARY OF DC GAINS (°F/lb/min.)

Testing Method	Oil	Coolant
Theoretical	1.18	20.51
Time domain	2.27	24.33
Frequency response	2.36	33.21
Pulse positive	1.66	22.82
Pulse negative	2.98	39.81
Transient positive	1.35	19.70
Transient negative	2.97	42.90

CHAPTER VI

CONCLUSIONS AND RECOMENDATIONS

Conclusions

This technique, when used in conjunction with Heymann's pole identification technique, appears to be a useful method for system identification. The model is obtained in its most useful form, the poles and the zeros, which one can readily transform into a differential equation. The error propagation analysis is a good measure of the reliability of the identified zeros, but it should not be used as the ultimate criterion for determining the best value.

For most real processes, the best values for the zeros will be the arithmetic means of the identified values at the points where the predicted error is relatively small. The use of the mean minimizes the effect of noise.

In most cases errors in the data are not magnified by the computation. Most errors do, however, delay the convergence of the results to the correct value. The error propagation analysis does not indicate this slow convergence; therefore caution should be exercised in the interpretation of the results when the time is less than one-half the longest time constant.

In the computer tests of the technique that were conducted on the Osage High Speed Computer the final form of the computation was the factored-zeros polynomial. For real systems it was found that this polynomial should not be factored until the results are interpreted and the mean values determined. The factoring of this polynomial greatly magnifies the errors in the zeros.

No great limitations of the time domain technique were detected in the investigation. In applying the technique, however, great care should be taken in the preparation of the experimental tests, especially in regards to the steady state and transport delay.

Recommendations for Future Work

The present investigation has focused primarily upon the development and verification of the identification technique. Further work should be done in identification of actual processes. There are several points that should be investigated in this connection. These apply to both the pole and the zero identification.

(1) The linearization of nonlinear systems should be analyzed in greater detail.

(2) A study of time-varying systems should be conconducted.

(3) The use of data-smoothing techniques to obtain more stable identifications should be investigated.

To facilitate the interpretation of the results, the zero identification program should be modified to determine the mean values of the identified results at the times when the predicted error is small. These mean values would then constitute a set of smoothed coefficients of the polynomial, the solution of which determines the system zeros.

REFERENCES

B1. Bishop, K. A. "A Generalized Pulse Testing Technique for Linear System Identification," Ph.D. Dissertation, The University of Oklahoma, 1965.

B2. Bishop, K. A. and Sims, R.A. "Analog Computation Using the Modified Donner Model 3100-D," Internal Report at University of Oklahoma Process Dynamics and Control Laboratory, 1963.

B3. Bishop, K. A. and Sims. R.A. "Electronic Instrumentation for Research in Process Control," Paper presented at 1962 Mid-America Electronics Conference, Kansas City, Mo., 1962.

B4. Brown, G. S. and Campbell, D. P. Principles of Servomechanisms, John Wiley & Sons, Inc., New York, 1948.

D1. Deming, W. E. Statistical Adjustment of Data, Dover Publications, Inc., New York, 1964.

D2. Dreifke, G. E. and Hougen, J. O. "Experimental Determination of System Dynamics by Pulse Methods," Preprint XXII-3, 1963, Joint Automatic Control Conference, University of Minnesota, June 19, 1963.

D3. Dreifke, G. E. and Hougen, J. O. and Mesmer, G., "Effects of Truncation on Time to Frequency Domain Conversions," I. S. A. Trans. 1, No. 4, p. 353, 1962.

E1. Eckman, D. P. Automatic Process Control. John Wiley & Sons, Inc., New York, 1958.

F1. Fadeev, D. K. and Fadeeva, V. N., Computational Methods of Linear Algebra. W. H. Freeman and Co., San Francisco, 1963.

F2. Fanning, R. J. and Sliepcevich, C. M., "The Dynamics of Heat Removal From a Continuous Agitated Tank Reactor," A. I. Ch. E. Journal, 5, No. 2, p. 240, June 1959.

G1. Gallier, P. W., Sliepcevich, C. M. and Puckett, T. H. Some Practical Limitations of Correlation Techniques in Determining Frequency Response," C. E. P. Symposium Series, Vol. 57, No. 36, p. 59, 1961.

H1. Haskins, D. E. and Sliepcevich, C. M. "The Invariance Principle of Control for Chemical Processes," I & E C. Fundamentals, Vol. 4, No. 3. p. 241, 1965.

H2. Heymann, M. "A Time Domain Technique for Linear System Identification," Ph.D. Dissertation, University of Oklahoma, 1965.

H3. Heymann, M., McGuire, M. L. and Sliepcevich, C. M. "New Time Domain Technique for Identification of Process Dynamics," I & E C Fundamentals, Vol. 6, p. 555, Nov. 1965.

H4. Hougen, J. O. and Lees, S. "Pulse Testing A Model Heat Exchange Process," Ind. Eng. Chem., 48, p. 1064, June 1956.

H5. Hougen, J. O. and Walsh, R. A. "Pulse Testing Methods," Chem. Engr. Prog., 57, No. 3, p. 69, March 1961.

L1. Laning, J. H. and Battin, R. H. Random Processes in Automatic Control, McGraw-Hill Book Co., New York, 1956.

L2. Luecke, R. H. "Analysis of Optimal Composite Feedback-Feedforward Control," Ph.D. Dissertation, University of Oklahoma, 1966.

M1. Mellor, J. W. Higher Mathematics, Dover Publications, Inc., New York, 1955.

S1. Stainthorp, F. P. and Axom, A. G. "The Dynamic Behavior of Thick-Walled Jacketed Pans," Chem. Engr. Sci., 20, p. 1, 1965.

S2. Stermole, F. J. and Larson, M. A. "Dynamic Response of Heat Exchangers to Flow Rate Changes," I. & E. C. Fundamentals Quarterly, 2, No. 1, p. 62, Feb. 1963.

S3. Stewart, W. S. "Dynamics of Heat Removal from a Jacketed Agitated Vessel," Ph.D. Dissertation, The University of Oklahoma, 1960.

W1 Worrel, R. B. "The Osage Algorithmic Language - Osage Algol," M.S. Thesis, The University of Oklahoma, 1964.

APPENDICES

APPENDIX A

NOMENCLATURE*

a_i = i^{th} coefficient of homogenous weighting function (2-16)

$b_i(t)$ = coefficient of the $(n-i)^{th}$ derivative of the dependent variable in an n^{th} order scalar differential system (2-1)

da_i = sum of the (n-1) terms of the form $1/(\rho_1-\rho_i)$ (3-9)

$f(t)$ = general time function (general forcing function of a process) (2-3)

g = k element zero vector (3-22)

G = k x k matrix of $G_n(i)$ (3-22)

$G_n(i)$ = convolution integral of the i^{th} term of the $(k-n)^{th}$ derivative of the weighting function (3-2)

$g_{jk}(t)$ = coefficient of the $(n-k)^{th}$ derivative of the j^{th} independent variable of a scalar differential system (2-1)

$H(s)$ = homogeneous transfer function (2-15)

$H_n(i)$ = integral of the i^{th} term of the $(k-n)^{th}$ derivative of the homogeneous weighting function times the input times the variable of integration (3-11)

---

*The numbers at the end of the definitions are the equation numbers in which the symbols are either defined or first used.

k_j	=	order of differential operator (2-2)
L	=	number of system outputs (2-1)
$L^n[\ ]$	=	linear differential operator of order n (2-2)
m	=	number of system inputs (2-1)
$M_j^{(k_j)}[\ ]$	=	linear differential operator of order k_j of the j^{th} forcing function (2-2)
$M_j^{(k_j)}*[\ ]$	=	adjoint linear operator of $M_j^{(k_j)}[\ ]$
n	=	order of the system (2-1)
s	=	Laplace operator
t	=	time
$W(t,\lambda)$	=	unit impulse response (general weighting function) of linear system (2-4)
$W_j(t,\lambda)$	=	particular weighting function of a linear system associated with input $x_j(t)$ (2-7)
$W(t-\lambda)$	=	weighting function for constant parameter linear system (2-12)
$x_i(t)$	=	i^{th} input function to a chemical process (2-1)
Y	=	k element response vector (3-22)
$y(t)$	=	dependent (output) variable of a scalar linear process (2-4)
$y_i(t)$	=	i^{th} dependent variable of a linear process (2-1)

Greek Letters

λ	=	parameter of $W(t,\lambda)$ (2-4)
ρ_i	=	i^{th} pole of the system (2-15)
τ	=	transport delay (4-3)
ω	=	angular frequency of forcing sine functions (4-2)

APPENDIX B

NUMERICAL EXAMPLE OF ZERO CALCULATION

The calculation is performed in five basic steps. These steps are:

1. Calculation of the coefficients of the weighting function and n-1 of its derivatives for the equations shown below.
2. Calculation of the integrals.
3. Multiplication of the integrals of step 2 by their coefficients and summation on i, and calculation of the expected error in these results.
4. Inversion of the coefficient matrix and calculation of the expected error in the inverse.
5. Multiplication of the inverse matrix times the response vector to determine the systems zeros and calculation of the expected error in the zeros.

The system chosen for this example is the same as was used in determining the effect of the number of significant figures in the data. This system is

$$Y(s) = \frac{0.00156\ s + 0.000051324}{(s + 0.0117)(s + 0.045)(s + 0.05833)}\ X(s) \quad \text{(A1-1)}$$

The coefficients of the homogeneous weighting function are:

$$a_1 = 644.006$$
$$a_2 = -2252.815 \qquad \text{(A1-2)}$$
$$a_3 = 1608.807.$$

The system contains one zero, therefore, k=1 and the coefficients of the first derivative are needed. These coefficients are:

$$a_1\rho_1 = -7.535$$
$$a_2\rho_2 = 101.327 \qquad \text{(A1-3)}$$
$$a_3\rho_3 = -93.842.$$

The inputs to be considered are:

$$x_1(t) = 13.017$$
$$x_2(t) = 0.09914t. \qquad \text{(A1-4)}$$

If the time is assumed to be fifteen seconds, the responses of the system (A1-1) to the inputs of (A1-4) are:

$$Y_1(15) = 1.54386$$
$$Y_2(15) = 0.064790. \qquad \text{(A1-5)}$$

The second step of the identification is the calculation of the integrals of Equation (2-19). This calculation is done in two parts. First, calculation of the function $e^{\rho_i\lambda}x(\lambda)$ at one second intervals of time (equal to the sampling time of the input and response data) up to the maximum desired time. The integral of this function is then calculated at each time point of interest. The program allows for selection of the initial time point and for the spacing of all subsequent ones. The integration technique used in the identification is a fifth order quadrature which uses six data points in the calculation of the integral at each point. This use of many points provides a certain degree of smoothing.

The results of the integrations at t = 15 are:

Input	Pole	$e^{15\rho_i} x(15)$	$\int_0^{15} e^{\rho_i \lambda} x(\lambda)\, d\lambda$
1	1	15.5142	213.436
	2	25.5658	278.862
	3	31.1146	312.147
2	1	1.77239	12.5482
	2	2.92071	17.7076
	3	3.56719	20.3979

(A1-6)

Multiplication of the integrals by the proper coefficients and summation of the three products to yield the coefficients of the unknowns is the third step of the computational scheme. From Equation (2-19) an example of one of these calculation is obtained by setting j = 0.

$$y_0(15) = g_1 \sum_{i=1}^{3} a_i e^{-\rho_i t} \int_0^{15} e^{\rho_i \lambda} x(\lambda)\, d\lambda \qquad \text{(A1-7)}$$

The expansion of this summation is

$$Y_0(15) = g_1(a_1e^{-\rho_1 t} \int_0^{15} e^{\rho_1 \lambda} x(\lambda)\,d\lambda + a_2e^{-\rho_2 t} \int_0^{15} e^{\rho_2 \lambda} x(\lambda)\,d\lambda$$

$$+ a_3e^{-\rho_3 t} \int_0^{15} e^{\rho_3 \lambda} x(\lambda)\,d\lambda). \tag{A1-8}$$

Substitution of the values from (A1-1), (A1-2), and (A1-7) into Equation (A1-8) yields

$$Y_0(15) = 4817.00\ g_1. \tag{A1-9}$$

The other coefficients can be determined in a similar manner. In matrix form the results are

$$G = \begin{vmatrix} 4817.00 & 833.027 \\ 149.684 & 36.6890 \end{vmatrix} \tag{A1-10}$$

At the same time the error matrix for an assumed one-percent error in the poles and input as calculated from Equation (3-20) is

$$\Delta G = \begin{vmatrix} 166.977 & 50.5095 \\ 3.02734 & 2.22371 \end{vmatrix} \tag{A1-11}$$

The inverse of the coefficient matrix is

$$G^{-1} = \begin{vmatrix} 0.000705012 & -0.0160074 \\ -0.00287631 & 0.0925630 \end{vmatrix} \tag{A1-12}$$

and the expected error in the inverse is

$$\Delta G^{-1} = \begin{vmatrix} 0.00000772113 & -0.000175677 \\ -0.0000123617 & -0.00127542 \end{vmatrix} \tag{A1-13}$$

The final step of the computation is multiplication of the inverse times the response vector to give the zeros of the system.

$$g = G^{-1}X$$

$$g = \begin{vmatrix} 0.0000513219 \\ 0.00155654 \end{vmatrix} \tag{A1-14}$$

The expected errors in these zeros are calculated from Equation

$$\text{(3-23)} \qquad \Delta g = \Delta G^{-1}X + G^{-1}\,\Delta X$$

When the expected error in the response is assumed to be one-percent, this computation yields

$$\Delta g = \begin{vmatrix} 0.00000105144 \\ -0.0000861536 \end{vmatrix} \tag{A1-15}$$

A more readily understandable form of these errors is an absolute percentage error. These errors are:

$$\Delta g = \begin{vmatrix} 2.04871 \\ 5.53496 \end{vmatrix} \text{ percent} \tag{A1-16}$$

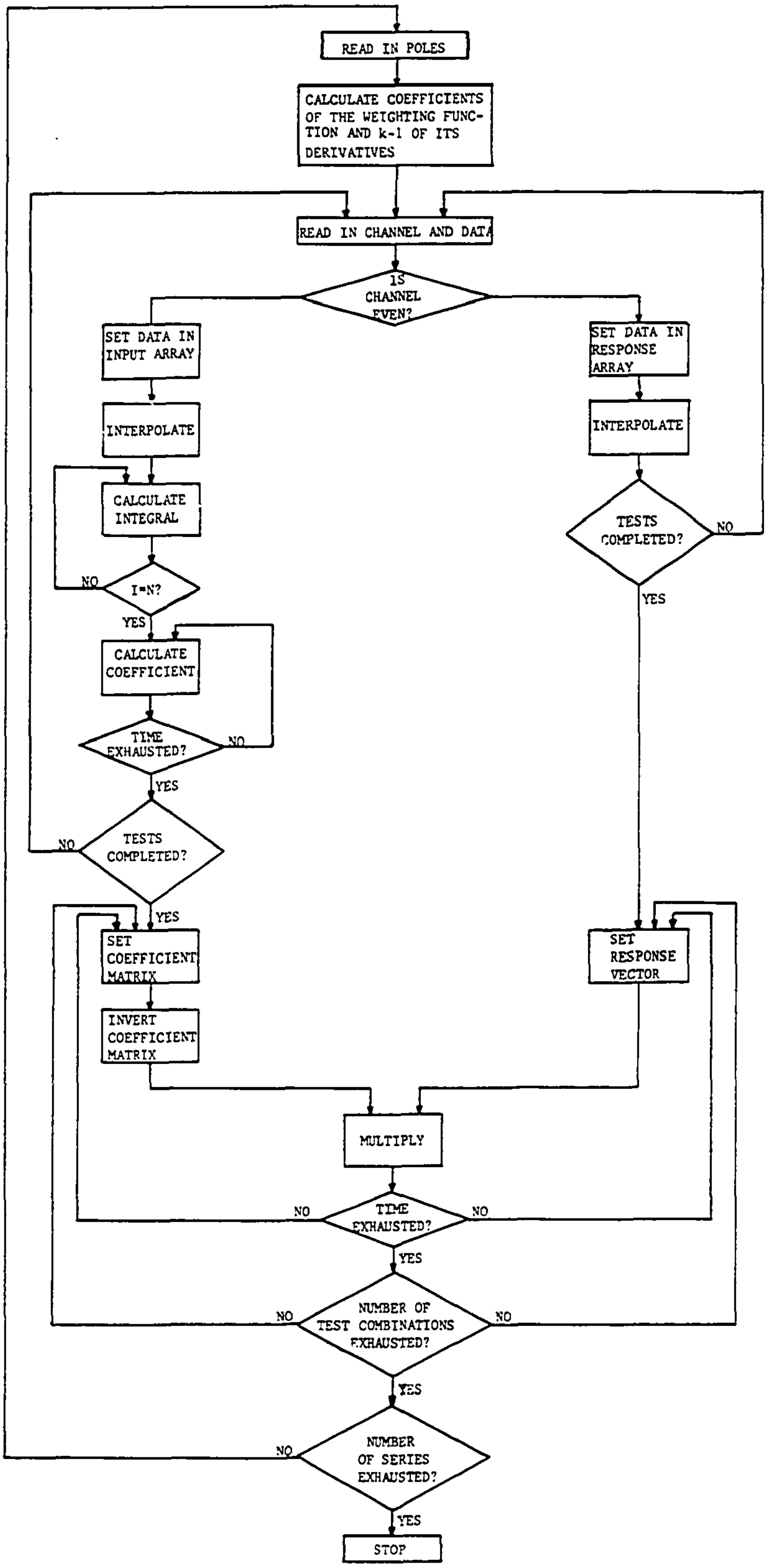
This step completes the identification calculation. (A1-14) are the final results and (A1-16) are the expected percentage errors in the respective terms of (A1-14).

APPENDIX C

LISTINGS OF IDENTIFICATION PROGRAMS

In this appendix the listings of the programs developed for the identification of the system zeros are given. These programs are written in g-level Fortran IV. Comments are inserted throughout the programs to identify the inputs and to explain the computational steps. Figure B-1 is a block diagram of the computational scheme.

The programs used in the pole identification studies were developed by Heymann and are listed in his dissertation. These programs are written in a modified Algol language called Osage Algol (W1).



C-1. Block Diagram of Identification Technique

```
      INTEGER N,CN,SERIES,OT,CTO,CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,       ZER00100
     1U,CV,I,J,L,M,P,Q,CA(80),CK,K,SLON(6),COMB,SET,AA(310),            ZER00110
     2CT401,NP1,NM1,DATA,CAM,CAQ,QM1,SINGUL,                            ZER00120
     3 DELAY(80),QQ,DEL,PAGE,COUNT                                      ZER00130
      REAL T,EPS,EPSIL,CA1(5),CB1(5),CEA1(5),CEB1(5),CZ(5)              ZER00140
      COMB = 0                                                          ZER00150
C***********************************************************************ZER00160
C       THIS PROGRAM IS TO CALCULATE THE ZEROS OF A SYSTEM FROM THE     ZER00170
C       POLES OF THAT SYSTEM AND THE FORCING AND RESPONSE FUNCTIONS     ZER00180
C       SAMPLED AT EQUAL BUT ANTERNATE INTERVALS OF TIME                ZER00190
C         CN         NUMBER OF SYSTEM POLES                             ZER00200
C         N          NUMBER OF DATA POINTS TO BE USED                   ZER00210
C         SERIES     IDENTIFICATION NUMBER FOR THE TESTS                ZER00220
C         CT         NUMBER OF TESTS IN THE SERIES                      ZER00230
C         CTO        NUMBER OF COMBINATIONS OF TESTS TO BE USED         ZER00240
C                      FOR THE IDENTIFICATION OF CN-1 ZEROS             ZER00250
C         CT6        NUMBER OF COMBS FOR CN-2 ZEROS                     ZER00260
C         CT7        NUMBER OF COMBS FOR CN-3 ZEROS                     ZER00270
C         CT8        NUMBER OF SERIES TO BE COMPUTED                    ZER00280
C         U          INITIAL TIME POINT OF IDENTIFICATION               ZER00290
C         CV         STEPPING INTERVAL OF TIME                          ZER00300
C         SLON 1-2 INTERMEDIATE PRINTOUTS 1 PRINT, 0 NO PRINT           ZER00310
C       SLON3        =1 FIND ROOTS OF THIRD ORDER SYSTEM                ZER00315
C         SLON4      =1 READIN THEORETICAL ZEROS AND CALC THEO ERROR    ZER00320
C         SLON5      =1 IDENTIFY FOR CN-2 Z5ROS                         ZER00330
C         SLON6      =1, IDENTIFY FOR CN-3 ZEROS                        ZER00340
C         T          SAMPLING TIME INTERVAL                             ZER00350
C***********************************************************************ZER00360
   52 CONTINUE                                                          ZER00370
      READ(1,124)N,CN,SERIES,CT,CTO,CT6,CT7,CT8,U,CV,(SLON(I),I=1,6)    ZER00380
     1,DATA                                                             ZER00390
      READ(1,126) T                                                     ZER00400
      U = U+1                                                           ZER00410
      COMB = COMB + 1                                                   ZER00420
      NP1=N+1                                                           ZER00430
      CT4 = NP1/CV                                                      ZER00440
```

```
      CT5 = CN*CTO                                                      ZER00450
      EPS = 1.*(10**(-8))                                               ZER00460
      EPSIL = 1.*(10**(-8))                                             ZER00470
      REAL CREX(5,5),CIMX(5,5),CEREX(5,5),CEIMX(5,5),CR(10,310),        ZER00480
     1CG(5,10,100),CE(5,10,100),BB(310),                                ZER00490
     2CSSR(10),CSSI(10),CER(10),A,B,C,D,E,DA,DB,DC,DD,DE,               ZER00500
     3DAA,DBB,DCC,CC1,CC2,CC3,CC4,CC5,CC6,CI(310),CII(310),             ZER00510
     4SS(310),SY(310),KS(310),KY(310),SSIN(5,100),SCOS(5,100),          ZER00520
     5KSIN(5,100),KCOS(5,100),CP,CDP,CS(5),CY(5)                        ZER00530
      WRITE(3,1116)                                                     ZER00540
      WRITE(3,11113)                                                    ZER00550
      WRITE(3,1111)                                                     ZER00560
      DO 1 I=1,CN                                                       ZER00570
      READ(1,120) CA1(I),CEA1(I),CB1(I),CEB1(I)                         ZER00580
      WRITE(3,1115) I,CA1(I),CEA1(I),CB1(I),CEB1(I)                     ZER00590
C***********************************************************************ZER00600
C        THE CALCULATIONS UP TO POINT 7 ARE TO DETERMINE THE            ZER00610
C          COEFFICIENTS OF THE WEIGHTING FUNCTION AND ITS               ZER00620
C          NECESSARY DERIVATIVES                                        ZER00630
C***********************************************************************ZER00640
    1 CONTINUE                                                          ZER00650
      PERERR=0                                                          Z5R00655
      DO 2 I=1,CN                                                       ZER00660
      PERERR=PERERR+CEA1(I)  /(100*CN)                                  ZER00665
      CEA1(I)=-CEA1(I)*CA1(I)/100+CA1(I)                                ZER00670
      CEB1(I)=-CEB1(I)*CB1(I)/100+CB1(I)                                ZER00680
    2 CONTINUE                                                          ZER00690
      CALL COEF(CN,CREX,CIMX,CA1,CB1)                                   ZER00700
      CALL COEF(CN,CEREX,CEIMX,CEA1,CEB1)                               ZER00710
      DO 7 J=1,CN                                                       ZER00720
      JM1 = J-1                                                         ZER00730
      WRITE(3,11115) JM1                                                ZER00740
      WRITE(3,1114)                                                     ZER00750
      DO 7 I=1,CN                                                       ZER00760
      ERRR=(CEREX(J,I)-CREX(J,I))*100/CREX(J,I)                         ZER00770
      IF(CIMX(J,I).EQ.0) GO TO 95                                       ZER00780
```

```
      ERRI=(CEIMX(J,I)-CIMX(J,I))*100/CIMX(J,I)                         ZER00790
   95 CONTINUE                                                          ZER00800
      ERRI=0                                                            ZER00810
      WRITE(3,11117) I,CREX(J,I),ERRR,CIMX(J,I),ERRI                    ZER00820
    7 CONTINUE                                                          ZER00830
       READ(1,128) (CER(I),I=1,CT)                                      ZER00840
      IF(SLON(4).EQ.0) GOTO 9                                           ZER00850
      READ(1,128) (CZ(I), I=1,CN)                                       ZER00860
    9 CONTINUE                                                          ZER00870
      READ(1,127)(CSSR(I),CSSI(I), I=1,CT)                              ZER00880
      L=1                                                               ZER00890
C***********************************************************************ZER00900
C        THE CALCULATIONS FROM 20 TO 35 ARE TO DETERMINE THE INTEGRALS  ZER00910
C          OF THE WEIGHTING  FUNCTION TIMES THE INPUT, ALSO THIS        ZER00920
C          FUNCTION FOR THE NECESSARY DERIVATIVES OF THE WEIGHTING FCN  ZER00930
C***********************************************************************ZER00940
   20 CONTINUE                                                          ZER00950
      READ(1,125) (AA(I),BB(I),I=1,N)                                   ZER00960
      NP1=N+1                                                           ZER00970
      DO 61 I=1,N                                                       ZER00980
      IF(AA(I)-2*(AA(I)/2).NE.1) GO TO 60                               ZER00990
      CII(I+2)=BB(I)-CSSR(L)                                            ZER01000
   60 CONTINUE                                                          ZER01010
      CI(I+1)=BB(I)-CSSI(L)                                             ZER01020
   61 CONTINUE                                                          ZER01030
      CALL INTERP(NP1,CII)                                              ZER01040
      CALL INTERP(NP1,CI)                                               ZER01050
      NP2=N+2                                                           ZER01060
      DO 99 I=1,NP2                                                     ZER01070
      CR(L,I)=CII(I+1)                                                  ZER01080
   99 CONTINUE                                                          ZER01090
      DO 10 J=1,CN                                                      ZER01140
      DO 11 K=1,NP1                                                     ZER01150
      SY(K)=EXP(-(CA1(J))*(K-1))*COS(CB1(J)*(K-1))*CI(K)                ZER01160
      SS(K)=EXP(-(CA1(J))*(K-1))*SIN(CB1(J)*(K-1))*CI(K)                ZER01170
   11 CONTINUE                                                          ZER01180
```

```
      CALL  INTEG(NP1,CN,SY,SCOS,I,J,U,CV)                              ZERO1190
      CALL  INTEG(NP1,CN,SS,SSIN,I,J,U,CV)                              ZERO1200
   10 CONTINUE                                                          ZERO1210
      CT4P1 = CT4+1 -U/CV                                               ZERO1220
      IF(SLON(1).EQ.0) GOTO 12                                          ZERO1230
      DO 77 J=1,CN                                                      ZERO1240
      WRITE(3,11118)(I,SCOS(J,I),SSIN(J,I),I=1,CT4P1)                   ZERO1250
   77 CONTINUE                                                          ZERO1260
   12 CONTINUE                                                          ZERO1270
      M=0                                                               ZERO1280
      DO 13 CK=U,NP1,CV                                                 ZERO1290
      M=M+1                                                             ZERO1300
      ERRIN=CI(CK)*CER(L)/100                                           ZERO1310
      DO 18 I=1,CN                                                      ZERO1320
      CALL CALC(CN,CK,NP1,N,I,CA1,CB1,CREX,CIMX,SSIN,SCOS,CP,           ZERO1330
     1 PERERR,CDP,ERRIN,U,CV)                                           ZERO1340
      CG(I,L,M)=CP                                                      ZERO1350
      CE(I,L,M)=CDP                                                     ZERO1360
   18 CONTINUE                                                          ZERO1370
   13 CONTINUE                                                          ZERO1380
      IF(L.EQ.CT) GOTO 19                                               ZERO1390
      L = L+1                                                           ZERO1400
      GOTO 20                                                           ZERO1410
   19 P=0                                                               ZERO1420
      PAGE=1                                                            ZERO1430
      IF(SLON(1).EQ.0) GOTO 22                                          ZERO1440
      DO 23 J=1,CN                                                      ZERO1450
      DO 23 L=1,CT                                                      ZERO1460
      WRITE(3,11118)(I,CG(N,L,I),CE(J,L,I), I=1,M)                      ZERO1470
   23 CONTINUE                                                          ZERO1480
   22 CONTINUE                                                          ZERO1490
C***********************************************************************ZERO1500
C          THE CALCULATIONS TO THIS POINT 8AVE DETERMINED A SET OF      ZERO1510
C          SIMULTANEOUS EQUATIONS.  THE FOLLOWING SOLVE THESE EQUATIONS ZERO1520
C          BY INVERTING THE COEFFICIENT MAYRIX AND MULTIPLYING          ZERO1530
C          THE RESPONSE VECTOR BY IT.                                   ZERO1540
```

```
C***********************************************************************
      IF(SLON(5).EQ.1) GOTO 51
      IF(SLON(6).EQ.1) GOTO 53
      READ(1,121)(CA(I),DELAY(I),I=1,CT5)
      GOTO 35
   51 CONTINUE
      CN=CN-1
      CT0=CT6
      CT6 = CT6*CN
      READ(1,121)(CA(I),DELAY(I),I=1,CT6)
      GOTO 35
   53 CONTINUE
      CN=CN-2
      CT0=CT7
      CT7 = CT7*CN
      READ(1,121)(CA(I),DELAY(I),I=1,CT7)
   35 CONTINUE
      P = P+1
      Q = CN*(P-1) + 1
      CAQ = CN*P
      WRITE(3,1116)
      WRITE(3,11114) CN, SERIES, P,(CA(I), I=Q,CAQ)
      WRITE(3,11124) SERIES,PAGE
      COUNT=5
      PAGE=PAGE+1
      WRITE(3,11122) (DELAY(I), I=Q,CAQ)
      WRITE(3,1118)
      CK=0
      DEL=0
      DO 25 I=Q,CAQ
      IF(DEL.GE.DELAY(I)) GO TO 25
      DEL=DELAY(I)
   25 CONTINUE
      IF(U.GT.DEL) GO TO 26
      CK=(DEL-U)/CV
      IF(DEL.LE.(U+CK*CV)) GO TO 26
```

```
      CK=CK+1                                                           ZERO1910
   26 CONTINUE                                                          ZERO1920
      CK=CK+1                                                           ZERO1930
      DO 27 I=1,CN                                                      ZERO1940
      DO 27 J=1,CN                                                      ZERO1950
      Q =CN*(P-1)+I                                                     ZERO1960
      CREX(I,J) = CG(J,CA(Q),CK)                                        ZERO1970
      CIMX(I,J) = CE(J,CA(Q),CK)                                        ZERO1980
   27 CONTINUE                                                          ZERO1990
      IF(QERERR.EQ.0) GO TO 30                                          ZERO1992
      CALL MADD(CN,CIMX,CREX,1)                                         ZERO1995
   30 CALL INVERT(CREX,CN,EPS,SINGUL)                                   ZERO2000
      IF(SINGUL.EQ.1) GOTO 32                                           ZERO2010
   31 CONTINUE                                                          ZERO2020
      IF(QERERR.EQ.0) GO TO 33                                          ZERO2025
      CALL INVERT(CIMX,CN,EPS,SINGUL)                                   ZERO2030
      IF(SINGUL.EQ.0) GOTO 33                                           ZERO2040
      Q = U + (CK-1)*CV - 1                                             ZERO2050
      WRITE(3,11110) Q                                                  ZERO2060
      COUNT=COUNT+1                                                     ZERO2070
      IF(COUNT.LE.60) GO TO 33                                          ZERO2080
      WRITE(3,11123) SERIES,PAGE                                        ZERO2090
      PAGE=PAGE+1                                                       ZERO2100
      COUNT=0                                                           ZERO2110
      GO TO 33                                                          ZERO2120
   32 Q = U + (CK-1)*CV - 1                                             ZERO2130
      WRITE(3,1119) Q                                                   ZERO2140
      COUNT=COUNT+1                                                     ZERO2150
      IF(COUNT.LE.60) GO TO 98                                          ZERO2160
      WRITE(3,11123) SERIES,PAGE                                        ZERO2170
      PAGE=PAGE+1                                                       ZERO2180
      COUNT=0                                                           ZERO2190
   98 CONTINUE                                                          ZERO2200
      IF(CK.EQ.CT4P1.AND.P.EQ.CTO) GOTO 50                              ZERO2210
      IF(CK.EQ.CT4P1.AND.P.LT.CTO) GOTO 35                              ZERO2220
      GOTO 26                                                           ZERO2230
```

```
   33 CONTINUE                                                   ZERO2240
      CALL MADD(CN,CIMX,CREX,-1)                                 ZERO2245
      Q=U+(CK-1)*CV                                              ZERO2250
      IF(SLON(2).EQ.0)GOTO 37                                    ZERO2260
      IF(COUNT.LE.55) GO TO 96                                   ZERO2270
      WRITE(3,11123) SERIES,PAGE                                 ZERO2280
      PAGE=PAGE+1                                                ZERO2290
      COUNT=0                                                    ZERO2300
   96 CONTINUE                                                   ZERO2310
      COUNT=COUNT+CN+3                                           ZERO2320
      WRITE(3,1117) Q                                            ZERO2330
      WRITE(3,11111)                                             ZERO2340
   37 CONTINUE                                                   ZERO2350
      DO 38 I=1,CN                                               ZERO2360
      M = CN*(P-1)+I                                             ZERO2370
      QQ=Q-DELAY(M)                                              ZERO2380
      CS(I)=GR(CA(M),QQ)                                         ZERO2390
      CA1(I)=CS(I)*CER(CA(M))/100                                ZERO2400
      IF (SLON(2).EQ.0) GOTO 39                                  ZERO2410
      WRITE(3,11118) CA(M),CS(I),CA1(I)                          ZERO2420
   39 CONTINUE                                                   ZERO2430
   38 CONTINUE                                                   ZERO2440
      IF(COUNT.LT.57) GO TO 97                                   ZERO2450
      WRITE(3,11123) SERIES,PAGE                                 ZERO2460
      PAGE=PAGE+1                                                ZERO2470
      COUNT=0                                                    ZERO2480
   97 CONTINUE                                                   ZERO2490
      COUNT=COUNT+CN+1                                           ZERO2500
      CALL MULTIP(CN,CY,CREX,CS)                                 ZERO2510
      CALL MULTIP(CN,CEB1,CIMX,CS)                               ZERO2520
      CALL MULTIP(CN,CEA1,CREX,CA1)                              ZERO2530
      DO 433 I=1,CN                                              ZERO2540
      CS(I)=CEA1(I)+CEB1(I)                                      ZERO2550
  433 CONTINUE                                                   ZERO2560
      QM1 = Q-1                                                  ZERO2570
      WRITE(3,1117) QM1                                          ZERO2580
```

```
      IF(SLON(4).EQ.0)GOTO 43                                   ZERO2590
      CB1(CN) =(CY(CN) - CZ(CN))*100/CZ(CN)                     ZERO2600
   43 CONTINUE                                                  ZERO2610
      NM1 = CN-1                                                ZERO2620
      IF( SLON(4).EQ.0) GOTO 45                                 ZERO2660
      DO 44 I=1,NM1                                             ZERO2630
      CB1(I) = (CY(I)-CZ(I)) *100/CZ(I)                         ZERO2670
   44 CONTINUE                                                  ZERO2690
   45 CONTINUE                                                  ZERO2680
      IF(CN.NE.3) GOTO 46                                       ZERO2700
      IF(SLON(3).EQ.0) GO TO 46                                 ZERO2705
      DAA = (CY(2)**2-4*CY(1))/4                                ZERO2710
      IF (DAA.GE.0) GO TO 48                                    ZERO2720
      DAA = ABS(DAA)                                            ZERO2730
      DAA = SQRT(DAA)                                           ZERO2740
      DBB=CY(2)*CS(2)/DAA+CS(1)/DAA                             ZERO2750
      DBB=DBB*100/DAA                                           ZERO2760
      CS(CN)=CS(CN)*100/CY(CN)                                  ZERO2770
      CS(2)=CS(2)*100/CY(2)                                     ZERO2780
      WRITE(3,11116) CY(CN),CS(CN)                              ZERO2790
      WRITE(3,11120) CY(2),CS(2)                                ZERO2800
      WRITE(3,11121) DAA,DBB                                    ZERO2810
      GO TO 49                                                  ZERO2820
   48 CONTINUE                                                  ZERO2830
      DAA = SQRT(CY(2)*CY(2)-4*CY(1))/2                         ZERO2840
      CY(1) = (CY(2)/2)-DAA                                     ZERO2850
      CY(2) = (CY(2)/2) + DAA                                   ZERO2860
      CS(1)=(.5-CY(2)/DAA)*CS(2)+2*CS(1)/DAA                    ZERO2870
      CS(2)=(.5+CY(2)/DAA)*CS(2)-2*CS(1)/DAA                    ZERO2880
   46 CONTINUE                                                  ZERO2890
      DO 47 I=1,CN                                              ZERO2900
      CS(I)=CS(I)*100/CY(I)                                     ZERO2910
   47 CONTINUE                                                  ZERO2920
      CS(CN) = ABS(CS(CN))                                      ZERO2930
      WRITE(3,11116) CY(CN),CS(CN),CB1(CN)                      ZERO2940
      NM1 = CN-1                                                ZERO2950
```

```
      DO 49 I=1,NM1                                                     ZERO2960
      CY(I)=-CY(I)                                                      ZERO2970
      CS(I) = ABS(CS(I))                                                ZERO2980
      WRITE(3,1112) I,CY(I),CS(I),CB1(I)                                ZERO2990
   49 CONTINUE                                                          ZERO3000
      IF(CK.LT.GT4P1) GO TO 26                                          ZERO3010
      IF(Q.EQ.CTO) GOTO 50                                              ZERO3020
      GOTO 35                                                           ZERO3030
   50 CONTINUE                                                          ZERO3040
      IF(COMB.EQ.CT8) GOTO 54                                           ZERO3050
      GOTO 52                                                           ZERO3060
   54 CONTINUE                                                          ZERO3070
C***********************************************************************ZERO3080
  120 FORMAT(4F12.0)                                                    ZERO3090
  121 FORMAT(20I2)                                                      ZERO3100
  122 FORMAT(10F4.0)                                                    ZERO3110
  123 FORMAT(6F10.0)                                                    ZERO3120
  124 FORMAT(T10,I4)                                                    ZERO3130
  125 FORMAT(10(I2,F5.3))                                               ZERO3140
  126 FORMAT(T10,F4.0)                                                  ZERO3150
  127 FORMAT(2F12.0)                                                    ZERO3160
  128 FORMAT(F12.0)                                                     ZERO3170
  129 FORMAT(2I2)                                                       ZERO3180
  130 FORMAT(I2)                                                        ZERO3190
 1110 FORMAT('+',40X,3(I2,1X))                                          ZERO3200
 1111 FORMAT(' ','THE POLES OF THE SYSTEM ARE',/,4X,'POLE',5X,          ZERO3210
     1'REAL PART',6X,'REL ERROR',6X,'IMAG PART',6X,'REL ERROR')         ZERO3220
 1112 FORMAT(' ',6X,'ZERO',I2,4X,3(E13.6,4X))                           ZERO3230
 1114 FORMAT(' ',4X,'COEF',6X,'REAL PART', 6X,'REL ERROR',              ZERO3240
     1 6X,'IMAG PART',6X,'REL ERROR')                                   ZERO3250
 1115 FORMAT(' ',4X,I3,6X,F9.6,6X,F9.6,6X,F9.6,6X,F9.6)                 ZERO3260
 1116 FORMAT('1')                                                       ZERO3270
 1117 FORMAT(' ',I3)                                                    ZERO3280
 1118 FORMAT('0','LIMIT',13X,'VALUE',10X,'RELATIVE ERROR')              ZERO3290
 1119 FORMAT(' ',I3 , 11X,'MATRIX OF COEFFICIENTS IS SINGULAR')         ZERO3300
11110 FORMAT(' ',I3 , 11X,'ERROR MATRIX IS SINGULAR')                   ZERO3310
```

```
11111 FORMAT(' ','TEST',12X,'RESPONSE',/)
11113 FORMAT(' ','IDENTIFICATION OF ZEROS     R.A. SIMS')
11114 FORMAT(' ','SYSTEM ORDER',I2,' SERIES',I5,' SET',
     1I3,' TESTS ',I2,',',I2,',',I2,6X,'PAGE ',I4)
11115 FORMAT(' ','COEFFICIENTS OF',1X,I2,1X,
     1 'DERIVATIVE OF WEIGHTING FUNCTION')
11116 FORMAT(' ',6X,'GAIN',6X,3(E13.6,4X))
11117 FORMAT(' ',4X,I3,6X,F9.3,6X,F9.3,6X,F9.3,6X,F9.3)
11118 FORMAT(' ',4X,I3,6X,E13.6,6X,E13.6)
11120 FORMAT(' ',6X,'ZERO',' REAL ',2(E13.6,4X))
11121 FORMAT(' ',10X,' IMAG ',2(E13.6,4X))
11122 FORMAT(' ',28X,'TIME DELAY',I2,',',I2,',',I2,'  SECONDS')
11123 FORMAT('1',15X,'SERIES ',I4,28X,'PAGE ',I4)
11124 FORMAT('+',15X,'SERIES ',I4,28X,'PAGE ',I4)
      STOP
      END
```

```
      SUBROUTINE  INTEG(N,NN,A,C,Y,J,U,CV)                              INTE0100
      DIMENSION  A(5),B(305),C(5,305)                                   INTE0110
      INTEGER U,CV                                                      INTE0120
      B(1)=0.                                                           INTE0130
      B(2)=(95*A(1)+285.4*A(2)-159.6*A(3)+96.4*A(4)-34.6*A(5)           INTE0140
     1 +5.4*A(6))/288                                                   INTE0150
      B(3)=(2.8*A(1)+12.9*A(2)+1.4*A(3)+1.4*A(4)                        INTE0160
     1 -.6*A(5)+.1*A(6))/9                                              INTE0170
      B(4)=(1.275*A(1)+5.475*A(2)+2.85*A(3)+2.85*A(4)                   INTE0180
     1 -.525*A(5)+.075*A(6))/4                                          INTE0190
      B(5)=(2.8*A(1)+12.8*A(2)+4.8*A(3)+12.8*A(4)+2.8* A(5))/9          INTE0200
      DO 1 I=6,N                                                        INTE0210
      B(I)=B(I-5)+(95*A(I-5)+375*A(I-4)+250*A(I-3)                      INTE0220
     1 +250*A(I-2)+375*A(I-1)+95*A(I))/288                              INTE0230
    1 CONTINUE                                                          INTE0240
      M=0                                                               INTE0250
      DO 2  I=U,N,CV                                                    INTE0260
      M=M+1                                                             INTE0270
      C(J,M) = B(I)                                                     INTE0280
    2 CONTINUE                                                          INTE0290
      RETURN                                                            INTE0300
      END                                                               INTE0310
```

```
      SUBROUTINE COEF(CN,CREX,CIMX,CA1,CB1)                              COEF0100
      INTEGER CN                                                         COEF0110
      DIMENSION CREX(5,5),CIMX(5,5),CA1(5),CB1(5)                        COEF0120
      DO 3 J=1,CN                                                        COEF0130
      CREX(1,J)=1                                                        COEF0140
      CIMX(1,J)=0                                                        COEF0150
      DO 3 I=1,CN                                                        COEF0160
      IF(I.EQ.J) GOTO 3                                                  COEF0170
      A = CA1(J) - CA1(I)                                                COEF0180
      B = CB1(J) - CB1(I)                                                COEF0190
      E = A*A + B*B                                                      COEF0200
      C = 1/E                                                            COEF0210
      D = A*C*CREX(1,J) - B*C*CIMX(1,J)                                  COEF0220
      CIMX(1,J) = (-B)*C*CREX(1,J) + C*A*CIMX(1,J)                       COEF0230
      CREX(1,J) = D                                                      COEF0240
      EPSIL=.000000001                                                   COEF0250
      IF(ABS(CIMX(1,J)).GE.EPSIL) GOTO 5                                 COEF0260
      CIMX(1,J) = 0                                                      COEF0270
    5 CONTINUE                                                           COEF0280
    3 CONTINUE                                                           COEF0290
      DO 6 J=2,CN                                                        COEF0300
      DO 6 I=1,CN                                                        COEF0310
      A=CREX(J-1,I)                                                      COEF0320
      B=CIMX(J-1,I)                                                      COEF0330
      C=CA1(I)                                                           COEF0340
      D=CB1(I)                                                           COEF0350
      CREX(J,I)=C*A - D*B                                                COEF0360
      CIMX(J,I) = (-D)*A + C*B                                           COEF0370
    6 CONTINUE                                                           COEF0380
      RETURN                                                             COEF0390
      END                                                                COEF0400
```

```
      SUBROUTINE CALC(CN,CK,NP1,M,I,CA1,CB1,CREX,CIMX,                  CALC0100
     1 SSIN,SCOS,CP,ERRP,CDP,ERRIN,U,CV)                                CALC0110
      INTEGER CK,CN,U,CV                                                CALC0120
      DIMENSION CREX(5,5),CIMX(5,5),CA1(5),CB1(5),                      CALC0130
     1 SCOS(5,100),SSIN(5,100),CEA1(5),Z(5,5),ZZ(5,5)                   CALC0140
      CP=0                                                              CALC0150
      CDP=0                                                             CALC0160
      DO 14 J=1,CN                                                      CALC0170
      DA=COS(CB1(J)*(CK-1))                                             CALC0180
      DB=SIN(CB1(J)*(CK-1))                                             CALC0190
      A=CREX(I,J)*DA - CIMX(I,J)*DB                                     CALC0200
      B=CREX(I,J)*DB + CIMX(I,J)*DA                                     CALC0210
      C=EXP(CA1(J)*(CK-1))                                              CALC0220
      D=C*(A*SCOS(J,M) + B*SSIN(J,M))                                   CALC0230
      IF(CK.EQ.U) GO TO 1                                               CALC0235
      ZZ(I,J)=(Z(I,J)+D)*CV/2+ZZ(I,J)                                   CALC0240
      Z(I,J)=D                                                          CALC0245
      GO TO 3                                                           CALC0250
    1 Z(I,J)=D                                                          CALC0255
      ZZ(I,J)=U*D/2                                                     CALC0260
    3 E=ZZ(I,J)*ERRP                                                    CALC0265
      CINT=-CREX(I,J)*(1-C1/CA1(J)*ERRIN                                CALC0270
      DD=CINT-D*ERRP*(CN-1)+E                                           CALC0280
      CDP=CDP+DD                                                        CALC0290
      CP=CP+D                                                           CALC0300
   14 CONTINUE                                                          CALC0310
      RETURN                                                            CALC0320
      END                                                               CALC0330
```

```
      SUBROUTINE INVERT (A1,N,EPS,SINGUL)                               INV00010
C                                                                       INV00020
C     SUBROUTINE INVERT INVERTS A MATRIC IN IT@S OWN SPACE USING THE    INV00030
C     GAUSS-JORDAN METHOD WITH COMPLETE MATRIX PIVOTING. I.E. AT EACH   INV00040
C     STAGE THE PIVOT HAS THE LARGEST ABSOLUTE VALUE OF ANY ELEMENT IN  INV00050
C     THE REMAINING MATRIX. THE COORIINATES OF THE SUCCRSSIVE MATRIX    INV00060
C     PIVOTS USED AT EACH STAGE OF THE RECUCTION ARE RECORDED IN THE    INV00070
C     SUCCESSIVE ELEMENTS POSITIONS OF THE ROW  CIKYMN UBDEX VECTIRS    INV00080
C     R ABD C . THESE ARE LATER CALLED UPON BY THE PROCEDURE PERMUTE WHIINV00090
C     REARRANGES THE ROWS AND COLUMSS OF THE MATRIX. IF THE MATRIX IS   INV00100
C     SINGULAR THE PROCEDURE EXITS TO AN APPROPRIATE LABEL IN THE MAIN  INV00110
C     PROGRAM. SINGLE = 1.                                              INV00120
      INTEGER SINGUL,I,J,K,L,PIVI,PIVJ,P,R(5),C(5)                      INV00130
      SINGUL = 0                                                        INV00140
      REAL*8 PIVOT,A(5,5),H                                             INV00143
      DIMENSION A1(5,5)                                                 INV00146
      DO 10 I=1,N                                                       INV00149
      DO 10 J=1,N                                                       INV00152
      A(I,J)=DBLE(A1(I,J))                                              INV00155
   10 CONTINUE                                                          INV00158
C     SET ROW AND COLUMN VECTORS                                        INV00160
      DO 1 I=1,N                                                        INV00170
      R(I) = I                                                          INV00180
      C(I) =I                                                           INV00190
1     CONTINUE                                                          INV00200
C     FIND INITIAL PIVOT                                                INV00210
      PIVI=1                                                            INV00220
      PIVJ=1                                                            INV00230
      DO 3 I=1,N                                                        INV00240
      DO 3 J=1,N                                                        INV00250
      IF(DABS(A(PIVI,PIVJ)).GE.DABS(A(I,J))) GO TO 2                    INV00260
      PIVI=I                                                            INV00270
      PIVJ=J                                                            INV00280
2     CONTINUE                                                          INV00290
3     CONTINUE                                                          INV00300
C                                                                       INV00310
```

```
C     START REDUCTION                                                   INV00320
      DO 8 I=1,N                                                        INV00340
      L=R(I)                                                            INV00350
      R(I)=R(PIVI)                                                      INV00360
      R(PIVI)=L                                                         INV00370
      L=C(I)                                                            INV00380
      C(I)=C(PIVJ)                                                      INV00390
      C(PIVJ)=L                                                         INV00400
      INTEGER ICNT,ICNT1,ICNTJ                                          INV00410
      ICNT=R(I)                                                         INV00420
      ICNT1=C(I)                                                        INV00430
      IF (DABS(A(ICNT,ICNT1)).GE.EPS) GO TO 4                           INV00440
      SINGUL=1                                                          INV00460
      RETURN                                                            INV00480
4     CONTINUE                                                          INV00490
      DO 5 J=1,N                                                        INV00500
      JK=N-J+1                                                          INV00510
      IF(JK.EQ.I)  GO TO 5                                              INV00520
      ICNTJ=C(JK)                                                       INV00530
      A(ICNT,ICNTJ)=A(ICNT,ICNTJ)/A(ICNT,ICNT1)                         INV00540
5     CONTINUE                                                          INV00550
      A(ICNT,ICNT1)=1./A(ICNT,ICNT1)                                    INV00560
      PIVOT=0                                                           INV00570
      DO 7 K=1,N                                                        INV00580
      IF (K.EQ.I) GO TO 7                                               INV00590
      DO 6 J=1,N                                                        INV00600
      JK=N-J+1                                                          INV00610
      IF (JK.EQ.I) GO TO 6                                              INV00620
      ICNTJ=C(JK)                                                       INV00630
      ICNTK=R(K)                                                        INV00640
      H=A(ICNT,ICNTJ)*A(ICNTK,ICNT1)                                    INV00650
      A(ICNTK,ICNTJ)=A(ICNTK,ICNTJ)-H                                   INV00660
      IF (I.GE.K.OR.I.GE.JK.OR.DABS(PIVOT).GE.DABS(A(ICNTK,ICNTJ)))     INV00670
     1GO TO 6                                                           INV00680
      PIVI=K                                                            INV00690
      PIVJ=JK                                                           INV00700
```

```
      PIVOT=A(ICNTK,ICNTJ)                                              INV00710
    6 CONTINUE                                                          INV00730
      A(ICNTK,ICNTL)=-A(ICNT,ICNTL)*A(ICNTK,ICNTL)                      INV00740
    7 CONTINUE                                                          INV00750
    8 CONTINUE                                                          INV00760
      DO 9 I=1,N                                                        INV00762
      DO 9 J=1,N                                                        INV00764
      A1(I,J)=SNGL(A(I,J))                                              INV00766
    9 CONTINUE                                                          INV00768
C                                                                       INV00770
C     REARRANGE ROWS                                                    INV00780
C                                                                       INV00790
      CALL PERMUT(A1,R,C,N,0)                                           INV00800
C                                                                       INV00810
C     REARRANGE COLUMNS                                                 INV00820
C                                                                       INV00830
      CALL PERMUT(A1,C,R,N,1)                                           INV00840
      RETURN                                                            INV00850
C                                                                       INV00860
      END                                                               INV00880
```

```
      SUBROUTINE PERMUT(A,S,D,N,JJ)                                     PER00010
C     PERMUTE IS A PROCEDURE USING JENSENaS DEVICE WHICH EXCHANGES ROWS PER00020
C     OR COLUMN SOF A MATRIX TO ACHIEVE A REARRANGEMENT SPECIFIED BY THEPER00030
C     PERMUTATION VECTORS S,D. ELEMENTS OF S SPECIFY THE ORIGINAL SOURCEPER00040
C     LOCATIONS WHILE ELEMENTS OF D SPECIFY THE DESIRED SESTINATION     PER00050
C     LOCATIONS. NORMALLY A AND B WILL BE CALLED AS SUBSCRIPTED VARIABLEPER00060
C     OF THE SAME ARRAY. THE PARAMETERS J,K NOMINATE THE SUBSCRIPTS     PER00070
C     OF THE DIMENSION AFFECTED BY THE PERMUTATION, P IS THE JENSEN     PER00080
C     PARAMETER. AS AN EXAMPLE OF THE USE OF THIS PROCEDURE SUPPOSE R,C PER00090
C     TO CONTAIN THE ROW AND COLUMN SUBSCFIPTS  FO THE SUCCESSIVE MATRIXPER00100
C     PIVOTS USED IN A MATRIX INVERSION OF AN ARRAY A. I.E. R(1) ,C(Z)< PER00110
C     ARE THE RRLATIVE SUBSCRIPTS OF THE FISST PIVOT, R%2<,C%2< OF THE  PER00120
C     SECOND PIVOT AND SO ON. THE TWO CALLS , CALL PERMUTE%A%J,P<, A%K,PPER00130
C     <,J,K,R,C,N,P< AND CALL PERMUTE%%A%P,J<,A%P,K<,J,K,C,R,N,P< WILL  PER00140
C     PERFORM THE REQUIRED REARRANGEMENT OF TOWS AND COLUNNS RESPECTIVELPER00150
      REAL A(5,5),W                                                     PER00160
      INTEGER J,K,N,P,S(5),D(5),TAG(5),LOC(5),I,T,TAGJ,TAGK             PER00170
C                                                                       PER00180
C     SETUP INITIAL VECTOR TAG NUMBER AND ADDRESS ARRAYS                PER00190
C                                                                       PER00200
      DO 1 I=1,N                                                        PER00210
      TAG(I)=I                                                          PER00220
      LOC(I)=I                                                          PER00230
1     CONTINUE                                                          PER00240
C                                                                       PER00250
C     START PERMUTATION                                                 PER00260
C                                                                       PER00270
      DO 4 I=1,N                                                        PER00280
      T=S(I)                                                            PER00290
      J=LOC(T)                                                          PER00300
      K=D(I)                                                            PER00310
      IF (J.EQ.K) GO TO 3                                               PER00320
      IF(WJ.EQ.1) GO TO 5                                               PER00325
      DO 2 P=1,N                                                        PER00330
```

```
      W=A(J,P)                                                          PER00335
      A(J,P)=A(K,P)                                                     PER00340
      A(K,P)=W                                                          PER00345
    2 CONTINUE                                                          PER00350
      GO TO 6                                                           PER00355
    5 CONTINUE                                                          PER00360
      DO 7 P=1,N                                                        PER00363
      W=A(P,J)                                                          PER00366
      A(P,J)=A(P,K)                                                     PER00370
      A(P,K)=W                                                          PER00372
    7 CONTINUE                                                          PER00375
    6 CONTINUE                                                          PER00377
      TAG(J)=TAG(K)                                                     PER00380
      TAG(K)=T                                                          PER00390
      TAGJ=TAG(J)                                                       PER00400
      TAGK=TAG(K)                                                       PER00410
      LOC(T)=LOC(TAGJ)                                                  PER00420
      LOC(TAGJ)=J                                                       PER00430
3     CONTINUE                                                          PER00440
4     CONTINUE                                                          PER00450
      RETURN                                                            PER00460
      END                                                               PER00470
```

```
      SUBROUTINE MULTIP(CN,CA,CB,CC)                                    MULT0100
      INTEGER CN,I,J                                                    MULT0110
      REAL CA(5),CB(5,5),CC(5)                                          MULT0120
      DO 1 I=1,CN                                                       MULT0130
      CA(I) = 0                                                         MULT0140
      DO 2 J=1,CN                                                       MULT0150
      CA(I) = CA(I) + CB(I,J)*CC(J)                                     MULT0160
    2 CONTINUE                                                          MULT0170
    1 CONTINUE                                                          MULT0180
      RETURN                                                            MULT0190
      END                                                               MULT0200
```

```
      SUBROUTINE MADD(N,A,B,M)                                          MADD0100
      DIMENSION A(5,5),B(5,5)                                           MADD0110
      DO 1 I=1,N                                                        MADD0120
      DO 1 J=1,N                                                        MADD0130
      A(I,J)=A(I,J)+M*B(I,J)                                            MADD0140
    1 CONTINUE                                                          MADD0150
      RETURN                                                            MADD0160
      END                                                               MADD0170
```

```
      SUBROUTINE  INTERP(N,A)                                          TERP0100
      DIMENSION A(305)                                                 TERP0110
      A(1) =    6*A(3)   -15*A(5)   +20*A(7)  -15*A(9)                 TERP0120
     1         +6*A(11)   -1*A(13)                                     TERP0130
      A(2) =(693*A(3)-1155*A(5)+1386*A(7)-990*A(9)                     TERP0140
     1       +385*A(11)  -63*A(13))/256                                TERP0150
      A(4) =( 63*A(3)  +315*A(5)  -210*A(7)+126*A(9)                   TERP0160
     1        -45*A(11)   +7*A(13))/256                                TERP0170
      A(6) =( -7*A(3)  +105*A(5)  +210*A(7)  -70*A(9)                  TERP0180
     1        +21*A(11)   -3*A(13))/256                                TERP0190
      NM4=N-4                                                          TERP0200
      DO 1 I=9,NM4,2                                                   TERP0210
      A(I-1)=(3*A(I-6)-25*A(I-4)+150*A(I-2)+150*A(I)                   TERP0220
     1 -26*A(I+2)+3*A(I+4))/256                                        TERP0230
    1 CONTINUE                                                         TERP0270
      IF(N-2*(N/2).EQ.1) GO TO 2                                       TERP0280
      A(N-4)=(-3*A(N-11)+21*A(N-9)-70*A(N-7)+210*A(N-5)                TERP0282
     1 +105*A(N-3)-7*A(N-1))/256                                       TERP0284
      A(N-2)=(7*A(N-11)-45*A(N-9)+126*A(N-7)-210*A(N-5)                TERP0286
     1 +315*A(N-3)+3*A(N-1))/256                                       TERP0288
      A(N)=(-63*A(N-11)+385*A(N-9)-990*A(N-7)                          TERP0290
     1     +1386*A(N-5)-1155*A(N-3)+693*A(N-1))/256                    TERP0300
      A(N+1)=-A(N-11)+6*A(N-9)-15*A(N-7)+20*A(N-5)                     TERP0310
     1        -15*A(N-3)+6*A(N-1)                                      TERP0320
      GO TO 3                                                          TERP0330
    2 CONTINUE                                                         TERP0340
      A(N-3)=(-3*A(N-10)+21*A(N-8)-70*A(N-6)+210*A(N-4)                TERP0342
     1 +105*A(N-2)-7*A(N))/256                                         TERP0344
      A(N-1)=(7*A(N-10)-45*A(N-8)+126*A(N-6)-210*A(N-4)                TERP0346
     1 +315*A(N-2)+63*A(N))/256                                        TERP0348
      A(N+1)=(-63*A(N-10)+385*A(N-8)-990*A(N-6)                        TERP0350
     1       +1386*A(N-4)-1155*A(N-2)+693*A(N))/256                    TERP0360
    3 CONTINUE                                                         TERP0370
      RETURN                                                           TERP0380
      END                                                              TERP0390
```