University of New Mexico
UNM Digital Repository

Electrical and Computer Engineering ETDs

Engineering ETDs

7-14-1972

A New Optimization Procedure for Digital Simulation

Donald Howard Schroeder

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

Part of the Electrical and Computer Engineering Commons

Recommended Citation

Schroeder, Donald Howard. "A New Optimization Procedure for Digital Simulation." (1972). https://digitalrepository.unm.edu/ece_etds/469

This Dissertation is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact amywinter@unm.edu.

UNIVERSITY OF NEW MEXICO-UNIVERSITY LIBRARIES
A14429 092205

A NEW
OPTIMIZATION
PROCEDURE
FOR DIGITAL
SIMULATION

SCHROEDER

LD
3781
N564Sch75
cop. 2

THE UNIVERSITY OF NEW MEXICO
ALBUQUERQUE, NEW MEXICO 87106

POLICY ON USE OF THESES AND DISSERTATIONS

Unpublished theses and dissertations accepted for master's and doctor's degrees and deposited in the University of New Mexico Library are open to the public for inspection and reference work. *They are to be used only with due regard to the rights of the authors.* The work of other authors should always be given full credit. Avoid quoting in amounts, over and beyond scholarly needs, such as might impair or destroy the property rights and financial benefits of another author.

To afford reasonable safeguards to authors, and consistent with the above principles, anyone quoting from theses and dissertations must observe the following conditions:

1. Direct quotations during the first two years after completion may be made only with the written permission of the author.
2. After a lapse of two years, theses and dissertations may be quoted without specific prior permission in works of original scholarship provided appropriate credit is given in the case of each quotation.
3. Quotations that are complete units in themselves (e.g., complete chapters or sections) in whatever form they may be reproduced and quotations of whatever length presented as primary material for their own sake (as in anthologies or books of readings) ALWAYS require consent of the authors.
4. The quoting author is responsible for determining "fair use" of material he uses.

This thesis/dissertation by Donald Howard Schroeder has been used by the following persons whose signatures attest their acceptance of the above conditions. (A library which borrows this thesis/dissertation for use by its patrons is expected to secure the signature of each user.)

NAME AND ADDRESS	DATE

July 1971—1,000—GS

This dissertation, directed and approved by the candidate's committee, has been accepted by the Graduate Committee of The University of New Mexico in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Title A NEW OPTIMIZATION PROCEDURE
FOR DIGITAL SIMULATION

Donald Howard Schroeder
Candidate

Electrical Engineering
Department

Charles L. Beckel
Dean

July 14, 1972
Date

Committee

Chairman

Arnold H. Koschmann

Sam D Stearns

S. Karni

A NEW OPTIMIZATION PROCEDURE FOR DIGITAL SIMULATION

BY
DONALD HOWARD SCHROEDER
B.S., University of Nebraska, 1965
M.S., University of New Mexico, 1967

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering
in the Graduate School of
The University of New Mexico
Albuquerque, New Mexico
August 1972

UNIVERSITY OF NEW MEXICO LIBRARY

LD
3781
N564Sch75
Cop. 2

ACKNOWLEDGEMENTS

For his patience and encouragement over the past years of coursework and his guidance through the selection, research, and preparation of this dissertation, the author is much indebted to Professor A. Erteza, the Chairman of the Committee on Studies. To Professor S. Stearns a similar obligation is owed for he not only suggested the problem and served on the Committee, but also made innumerable recommendations and comments regarding the scope and direction of this dissertation. For their interest in the topic, their guidance as members of the Committee, and their advice and counsel during preparation of the dissertation, appreciation is hereby expressed to Professors S. Karni and A. Koschmann.

The author is also indebted to his wife, Miriam, for making a presentable draft of this paper from nearly illegible notes and to Barbra Ford for the excellent job she did of preparing the final manuscript in a very limited time. Any mistakes that remain are solely the responsibility of the author; without the help of Miriam and Barbra, many additional errors would be present.

Finally, a note of deepest gratitude is expressed to Sandia Laboratories for their continued interest in, and support of, their employee's higher education.

626607

A NEW OPTIMIZATION PROCEDURE FOR DIGITAL SIMULATION*

Donald Howard Schroeder, Ph.D.
Department of Electrical Engineering
The University of New Mexico, 1972

ABSTRACT

The theory needed to define and obtain an optimized digital simulation of a given continuous system is presented in this dissertation. The frequency domain approach to simulation is used and a nonlinear function minimization algorithm, implemented on a digital computer, is incorporated to obtain the optimum simulation.

The discrete transfer function obtained from z-transform theory is used to represent the digital system. The frequency domain response of this discrete transfer function can then be expressed simply as a function of frequency, for frequencies up to one-half the sampling frequency. A frequency domain error measure, in the complex plane, is defined to be the sum of the squared magnitudes of the difference in the frequency domain responses of the continuous and discrete systems at a finite number of frequencies. The coefficients of the discrete system are adjusted, by means of the function minimization algorithm, to minimize the defined error.

---

*This research supported by the Atomic Energy Commission.

Similar optimization procedures have been previously described. These methods, in some instances, produce an unstable, or nearly unstable, simulation of a stable continuous system. A technique is introduced here of constraining the discrete system coefficients to insure a stable simulation of a stable system. In addition, these parameters may be further restrained to keep the transient behavior of the discrete system similar to that of the system simulated. Essentially these constraints limit the time domain error while the frequency domain error is minimized.

To evaluate the effectiveness of the above procedure, it is compared with the bilinear transformation and the impulse, step, and ramp invariant methods of obtaining a discrete simulation of both a first and second order continuous system. These comparisons are made in the time domain, for impulse, step, ramp, and sinusoidal inputs, as well as in the frequency domain. In addition, the effects of the constraints on the optimization technique are shown. In some instances, a small percentage increase in the frequency domain error will allow orders of magnitude decreases in the time domain errors.

TABLE OF CONTENTS

Page

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. THE OPERATOR APPROACH AND ITS APPLICATION TO DISCRETE SYSTEMS 7

2.1 Historical Resume 7

2.2 Continuous Systems 8

2.2.1 Laplace Transforms 9

2.3 Sampled Data Systems 12

2.4 Discrete Digital Systems 18

2.4.1 The z-Transform 19

2.4.2 Linear Difference Equations 22

2.4.3 The Digital Transfer Function 24

2.4.4 The Digital Impulse Function 26

2.4.5 Stability of the Digital Transfer Function 28

2.4.6 Finite Register Length of Discrete Systems 28

2.5 Summary of the Simulation System 29

2.6 A General Discussion of Simulation Techniques 33

CHAPTER 3. FREQUENCY DOMAIN SIMULATION BY DIRECT SUBSTITUTION 44

3.1 Tustin's Method 45

3.2 The Bilinear Transformation 53

3.3 The Method of Rich and Shaw 59

3.4 z-Forms 64

Page

CHAPTER 4. INVARIANT SIMULATIONS 68

4.1 The Impulse Invariant Simulation 70

4.2 The Step Function Invariant Simulation 76

4.3 Higher Order Invariant Simulation 82

CHAPTER 5. DIGITAL SIMULATION THROUGH OPTIMIZATION PROCEDURES 85

5.1 Simulation System Errors 85

5.2 Some Investigated Optimization Procedures 88

CHAPTER 6. COMBINED FREQUENCY DOMAIN - TIME DOMAIN ERROR CRITERIA AND OPTIMIZATION PROCEDURE 94

6.1 Limiting the Time Domain Error 97

6.2 Defining the Digital System and the Frequency Domain Error 99

6.3 Optimization Procedure 102

CHAPTER 7. A COMPARISON OF SIMULATION METHODS 121

7.1 Derivation of the Desired Responses and the Discrete Simulation Coefficients for the First Order Continuous System 123

7.2 Derivation of the Desired Responses and the Discrete Simulation Coefficients for the Second Order Continuous System 131

7.3 Error Comparisons of Simulations of the First Order System 138

7.3.1 Frequency Domain Errors 139

7.3.2 Time Domain Errors 146

7.4 Error Comparisons of Simulations of the Second Order System 171

7.4.1 Frequency Domain Errors 171

7.4.2 Time Domain Errors 178

Page

7.5 Summary of Results 197

CHAPTER 8. CONCLUSIONS 202

APPENDIX A 205

APPENDIX B 211

REFERENCES 213

LIST OF FIGURES

Figure		Page
1	Sampled Data Control System	12
2	Simulated Continuous System	29
3	Discrete Simulation System	29
4	Continuous Equivalent of Discrete System	29
5	Area of Tustin's Δ-Elements	45
6	Flow Chart of Optimization Procedure	120
7	Frequency Domain Error of Impulse, Adjusted Impulse and Step Invariant Simulations of First Order System	140
8	Frequency Domain Error of Ramp Invariant, Bilinear Approximation and First Order Optimized Simulations of First Order System	142
9	Frequency Domain Error of Three, Second Order Optimized Simulations of First Order System	143
10	Frequency Domain Error of Three, Third Order Optimized Simulations of First Order System	144
11	Frequency Domain Error of Three, Fourth Order Optimized Simulations of First Order System	145
12	Frequency Domain Error of a First, Second, Third, and Fourth Order Optimized Simulations of First Order System	147
13	Response and Error of Impulse, Adjusted Impulse and Step Invariant Simulations of First Order System with Impulse Input	149

Figure Page

14 Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Impulse Input 150

15 Response and Error of Three, Second Order Optimized Simulations of First Order System with Impulse Input 151

16 Response and Error of Three, Third Order Optimized Simulations of First Order System with Impulse Input 152

17 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with Step Input 158

18 Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Step Input 160

19 Response and Error of Three, Second Order Optimized Simulations of First Order System with Step Input 161

20 Response and Error of Three, Third Order Optimized Simulations of First Order System with Step Input 163

21 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with Ramp Input 164

22 Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Ramp Input 166

23 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with sin 3t Input 167

24 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with sin 7t Input 168

25 Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with sin 3t Input 169

Figure		Page
26	Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with sin 7t Input	170
27	Frequency Domain Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System	172
28	Frequency Domain Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System	173
29	Frequency Domain Error of Three, Third Order Optimized Simulations of Second Order System	175
30	Frequency Domain Error of Three, Fourth Order Optimized Simulations of Second Order System	176
31	Frequency Domain Error of Second, Third, and Fourth Order Optimized Simulations of Second Order System	177
32	Frequency Domain Error of Ramp Invariant and Third Order Optimized Simulations of Third Order System	179
33	Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with Impulse Input	180
34	Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with Impulse Input	182
35	Response and Error of Three, Third Order Optimized Simulations of Second Order System with Impulse Input	183
36	Response and Error of Three, Fourth Order Optimized Simulations of Second Order System with Impulse Input	184
37	Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with Step Input	186

Figure Page

38 Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulation of Second Order System with Step Input 187

39 Response and Error of Three, Third Order Optimized Simulations of Second Order System with Step Input 188

40 Response and Error of Three, Fourth Order Optimized Simulations of Second Order System with Step Input 189

41 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with Ramp Input 191

42 Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with Ramp Input 192

43 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with sin 3t Input 193

44 Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with sin 3t Input 194

45 Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with sin 7t Input 195

46 Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with sin 7t Input 196

LIST OF ABBREVIATIONS AND SYMBOLS

$\delta(t)$	Dirac delta function, impulse at $t = 0$.
ε	Small positive constant.
σ	Real part of the complex frequency variable, s.
$\sum$	Summation of the indicated terms.
ω	Imaginary part of the complex frequency variable, s.
f_s	Sampling frequency, $\frac{1}{T}$.
$h(t)$	Impulse response of $H(s)$.
$H(s)$	Continuous transfer function.
$\tilde{H}(z)$	Discrete transfer function.
$\bar{H}(s)$	Continuous function obtained from $H(z)$.
Imag(p)	Imaginary part of the complex expression, p.
$L(f)$	Laplace transform of the function, f.
$L^{-1}(F)$	Inverse Laplace transform of the function, F.
$Q(s)$	Transfer function of a sampled data system interpolator.
RAD	Maximum allowed radius of z-plane poles.
Real(p)	Real part of the complex expression, p.
s	Complex frequency variable, $s = \sigma + j\omega$.
t	Time.
T	Sampling interval.
$u(t)$	Unit step function at $t = 0$.
$x(t)$	Continuous system input.
x_n	Discrete system input, $x_n = x(nT)$.
$y(t)$	Continuous system output.

y_n	Discrete system output, $y_n \neq y(nT)$ in general.
z	Discrete system variable, unit advance operator.
$Z(f_n)$	z-transform of sequence $\left\{f_n\right\} = F(z)$.
$Z^{-1}[F(z)]$	Inverse z-transform of $F(z) = \left\{f_n\right\}$.

CHAPTER 1. INTRODUCTION

The ever increasing availability of digital signal processing equipment has encouraged system analysts and designers not only to incorporate these discrete systems into new designs but also to simulate present or future continuous systems on a digital computer. In order that this digital processing be done with maximum efficiency, it is necessary to select the digital system parameters in some optimum fashion. A method is presented here of obtaining these optimized parameters. In particular, the objectives of this paper are:

1. To review the available literature for simulation techniques based on the z-transform approach and to describe the most widely used techniques available today.
2. To define simulation error criteria.
3. To describe an optimization procedure which will produce the "best" discrete approximation to a given continuous system for the error criterion specified.
4. To compare the time and frequency domain errors of different simulation methods.

The term "digital simulation" is often associated with the simulation languages available at most computing centers.

Four of the most common of these languages are MIDAS, CSMP, MIMIC, and DSL-90, of which the latter two are described in detail by Chu.[1] Just as ordinary differential equations can often be solved by strictly time domain methods, these simulation languages are based on the time domain approach. In general, the set of differential equations describing the continuous system is solved by the standard techniques for numerically integrating differential equations. The fourth order Runge-Kutta method is often chosen for its stability, the ease of adjusting the step size, and the fact that it is self starting. These languages are relatively easy to use and in general provide very accurate approximations to the response of the real system being simulated. The "cost" of these advantages is that a large computer is required to efficiently execute the simulation and, because of the integration techniques used and the high accuracy provided, the sampling rate must be very high in comparison to the minimum specified by the sampling theorem.[2] Thus these techniques are not in general usable for small digital processors where a reasonably accurate approximation is desired with a minimum number of calculations per sample and a near minimal sampling rate. Neither are they useful for real time simulations where the computational time is severely limited. For these reasons this approach to simulation is not considered here.

This paper is concerned only with the z-transform approach to digital simulation. Just as the Laplace transform is a commonly used operator technique for solving

linear differential equations, the z-transform is the dominant operator approach to the analysis of linear difference equations. A common approach to the analysis of continuous systems is to consider the system in terms of its transfer function. From this transfer function the stability of the system can be determined, its transient response can be characterized, and its steady state response for any given input signal can be specified.[3] The z-transform transfer function uniquely determines similar properties of the discrete system. Thus the z-transform approach to digital simulation can be defined as a method of obtaining a discrete system z-transform transfer function whose 3 characteristics mentioned above approximate, in some sense, the corresponding properties of the Laplace transform function of the simulated system. Some of the more commonly used of such methods are described in this paper together with a method of optimizing a discrete system based on a given error criterion.

The content of the remaining chapters of this paper are briefly described below.

Chapter 2 presents a brief history of digital methods and a general discussion of the mathematics involved in continuous, sampled data and discrete systems. In particular, the z-transform is discussed and most of the notation to be used throughout the remainder of this paper is introduced here. The criteria for discrete systems for stability and minimizing the effects of finite register length are briefly discussed. Both the simulated and simulation systems are

defined and restricted to a manageable level for the purposes of this paper. Finally, a brief review of general z-transform simulation techniques is presented.

Chapter 3 describes the derivation and resulting form of some simulation methods obtained by the direct substitution of a function of z for s in the transfer function H(s). One of the first simulation methods was proposed by Tustin in 1947.[4] It is described here and shown to be equivalent to the bilinear substitution, which is one of the most widely used simulation techniques available. The method of Rich and Shaw[5,6,7] is discussed, even though it is not a widely used technique, because of its method of determining pole and zero locations. Finally the z-forms of Boxer and Thaler[8] are described.

Chapter 4 discusses invariant simulation techniques. The impulse invariant method, which is widely used, is also known as the standard z-transform. The step and ramp invariant methods are not as well known; yet, it is shown that their derivation is not a great deal more difficult than that of the impulse invariant form and that, for some simulations, the results obtained are far better.

Chapter 5 briefly reviews some of the optimization procedures recently described in the literature. Of particular interest are the error criteria used for error minimization, the form of the discrete system transfer function assumed, and the method used to insure stability.

Chapter 6 discusses time domain and frequency domain errors and the difficulty of developing a "universal" error measure.

A method of limiting the time domain error is introduced which can be incorporated into the frequency domain error definition chosen for this paper. The technique for minimizing this frequency domain error is then defined, within the constraints of the limited time domain error.

Chapter 7 consists primarily of figures comparing the time and frequency domain errors of the various simulation techniques considered. The results of these comparisons are discussed and summarized. The conclusions made from this paper are presented in Chapter 8.

The experienced digital system designer will find Chapters 4, 6, and 7 of particular interest. The invariant simulations of Chapter 4 are not new; however, the derivation and description differ from that found in most other presentations. The frequency domain error measure presented in Chapter 6 differs from that used with other optimization procedures as here both phase and magnitude errors are included. Similarly, the form of the discrete system assumed is unique in that the numerator and denominator polynomials may be of even or odd order and need not be equal in order. More importantly, the real and complex poles are separated so that the poles may be kept within a circle of radius less than 1, to limit the time domain errors. The resultant form of the

discrete system remains as a cascade or parallel combination of first and second order systems. The comparisons of Chapter 7 are complete for the first and second order systems considered. They clearly demonstrate the characteristics of the simulation methods considered.

CHAPTER 2. THE OPERATOR APPROACH AND ITS APPLICATION TO DISCRETE SYSTEMS

2.1 Historical Resume

In the past 25 years, the introduction and widespread use of the digital computer has emphasized the use of numerical methods dating back to the early 1600's. In fact, the origin of numerical techniques must be dated with man's first calculations as these first mathematical operations were certainly digital. However, the first real efforts to develop these numerical methods were made in the 17th century when the need for linear computational algorithms arose in the generation of mathematical tables and reduction of astronomical data. Many of the techniques developed at that time are the classical methods of today. For example, the names of Gregory (1638-1675) and Newton (1642-1727) are found in many present texts relating to numerical analysis[9,10] or the calculus of finite differences.[11,12,13]

World War II brought the need for more sophisticated techniques of information processing and pushed the development of sampled data control systems. Since that time, several books[14,15,16,17,18] on these systems, in addition to an abundance of literature on discrete systems in general, have been published.

In the past few years, the advent of LSI (Large Scale Integration) has only increased the rate at which digital

signal processing is supplementing or replacing the corresponding analog processors. A few of the advantages of digital processing include high stability, small size, low weight, high reliability, and ease of modification of parameters. Other advantages and disadvantages too numerous to mention here are discussed in recent special issues of the IEEE Transactions on Audio and Electroacoustics[19,20,21] as well as in many other publications.

2.2 Continuous Systems[3]

In order to simulate any continuous system, the system itself must be well defined and mathematically describable, generally in terms of differential equations. Consider the simple system given below:

$$\frac{dy(t)}{dt} + b_o y(t) = a_1 \frac{dx(t)}{dt} + a_o x(t) \tag{1}$$

where the input x(t), is a known function of time, a_o, a_1, and b_o are constants, and y(t) is the desired response with y(t) at t = 0 known. (1) can be rewritten as

$$\frac{dy(t)}{dt} = y'(t) = a_1 x'(t) + a_o x(t) - b_o y(t) \ . \tag{2}$$

Now the right side of (2) can be written as f(y,t) where

$$f(y,t) = a_1 x'(t) + a_o x(t) - b_o y(t) \tag{3}$$

and (2) becomes

$$y'(t) = f(y,t) \ . \tag{4}$$

2.2.1 Laplace Transforms[3]

The Laplace transform of (1) is

$$sY(s) - y(0) + b_oY(s) = a_1sX(s) + a_oX(s) \tag{5}$$

or

$$Y(s) = \frac{(a_1s + a_o)X(s) + y(0)}{(s + b_o)} \ . \tag{6}$$

This is usually written as

$$Y(s) = H(s)X(s) + \frac{y(0)}{s + b_o} \tag{7}$$

where H(s) is defined as the system transfer function

$$H(s) = \frac{a_1s + a_o}{s + b_o} = a_1 + \frac{a_o - a_1b_o}{s + b_o} \ . \tag{8}$$

A few properties of the Laplace transform are given below (further information may be found in Aseltine,[22] Goldman,[3] or any standard engineering analysis text).

1. $$f(s) = L[f(t)] = \int_0^{\infty} f(t)e^{-st}\,dt \tag{9}$$

given $s = a + jb$ and $\int_0^{\infty} \left|f(t)e^{-at}\right|dt$ exists and is finite, and $f(t) = 0,\ t < 0$.

2. $$f(t) = L^{-1}[F(s)] = \int_{c-j\infty}^{c+j\infty} F(s)e^{st}\, ds \qquad (10)$$

provided c is greater than the real part of any singularities of F(s).

3. $$L[f'(t)] = sF(s) - f(0) \qquad (11)$$

4. $$L\left[\int_0^t f(u)du\right] = \frac{1}{s} F(s) \qquad (12)$$

5. $$L^{-1}[F(s)G(s)] = \int_0^t f(\tau)g(t - \tau)d\tau$$

$$= \int_0^t g(\tau)f(t - \tau)d\tau \ . \qquad (13)$$

The initial value theorem for Laplace transforms is

$$\lim_{t\to 0} h(t) = \lim_{s\to\infty}[sH(s)] \ . \qquad (14)$$

Thus, if H(s) is not a proper fraction in s, that is, if the degree of the numerator polynomial is not less than the degree of the denominator polynomial, then (14) implies h(t) is unbounded as t approaches 0. This type of response will not be considered in detail; however, the problem can be isolated by synthetic division as was done in (8), and the resulting constant term can be handled separately from the remainder of the problem.

The impulse response of a continuous system described by the differential equation of (1) or by the system transfer function of (8) is discussed below. The unit impulse function itself can be described in many ways, one of which is

$$\delta(t - k) = 0 \ , \quad t \neq k \qquad \int_{k-\varepsilon}^{k+\varepsilon} \delta(t - k)dt = 1 \tag{15}$$

where ε is an arbitrarily small positive value and k is any constant.

The Laplace transform of the impulse function is, by definition

$$L[\delta(t - k)] = \int_0^{\infty} \delta(t - k)e^{-st} \, dt$$

$$= \lim_{\varepsilon \to 0} \int_{k-\varepsilon}^{k+\varepsilon} \delta(t - k)e^{-st} \, dt$$

$$L[\delta(t - k)] = e^{-sk} \qquad \text{for } k \geq 0 \ . \tag{16}$$

Hence, if the input $x(t)$ is $\delta(t)$, the impulse function at $t = 0$, and the output $y(0) = 0$ then, from (8),

$$Y(s) = H(s)X(s) = H(s) \cdot 1 = H(s) \tag{17}$$

and

$$y(t) = h(t) = L^{-1}[H(s)] \ . \tag{18}$$

Thus, the transfer function $H(s)$ of any physical system can be obtained by measuring the response to a unit impulse input.

2.3 Sampled Data Systems[15]

As discussed in the introduction, sampled data systems were subjected to extensive study during and after World War II. By following the notation of Jury,[15] a sampled data system may be represented as shown in Figure 1.

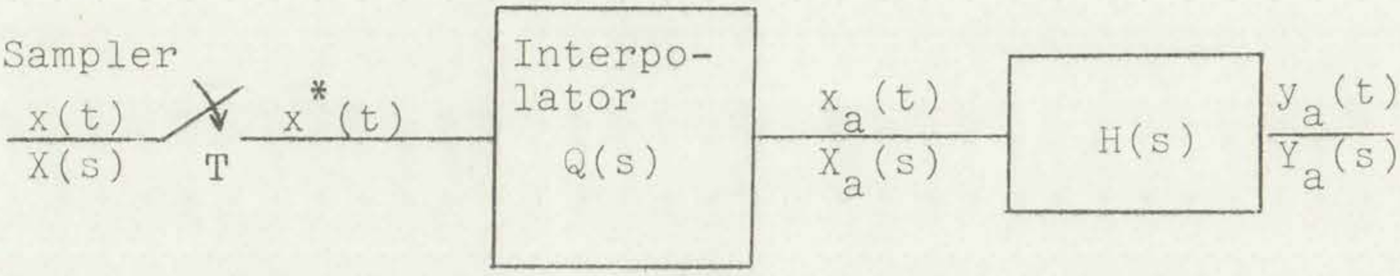


Figure 1

The sampler is assumed to be ideal. The sampling interval is T seconds and x(t) is a sequence of weighted impulses:

$$x^*(t) = x(t) \sum_{n=0}^{\infty} \delta(t - nT) = \sum_{n=0}^{\infty} x(nT)\delta(t - nT) \ . \tag{19}$$

The Laplace transform of $x^*(t)$ follows from (16)

$$X^*(s) = L\left[\sum_{n=0}^{\infty} x(nT)\delta(t - nT)\right]$$

$$= \int_0^{\infty} \sum_{n=0}^{\infty} x(nT)\delta(t - nT)e^{-st} \, dt \ , \tag{20}$$

thus

$$X^*(s) = \sum_{n=0}^{\infty} x(nT)e^{-nsT} \ . \tag{21}$$

The usual substitution of x_n for x(nT) in (21) gives

$$X^*(s) = \sum_{n=0}^{\infty} x_n e^{-nsT} . \tag{22}$$

The interpolator shown in Figure 1 is used to attempt to reconstruct x(t) from $x^*(t)$. This reconstruction is labeled $x_a(t)$ and in general is not equal to x(t). From Figure 1,

$$X_a(s) = X^*(s)Q(s) \tag{23}$$

or, from (22)

$$X_a(s) = \sum_{n=0}^{\infty} x_n e^{-nsT} Q(s) \tag{24}$$

and

$$x_a(t) = L^{-1}\left[X_a(s)\right] . \tag{25}$$

One of the simplest and most useful interpolators is the zero-order hold which effectively "holds" the value of $x_a(t)$ at x_n until x_{n+1} is available. Then $x_a(t)$ is "held" at the value of x_{n+1} until x_{n+2} is available, etc. Now $x_a(t)$ can be written as

$$x_a(t) = \sum_{n=0}^{\infty} x_n[u(t - nT) - u\{t - (n + 1)T\}] \tag{26}$$

and

$$X_a(s) = \sum_{n=0}^{\infty} x_n \left(\frac{e^{-nTs} - e^{-(n+1)Ts}}{s} \right) \tag{27}$$

$$= \sum_{n=0}^{\infty} x_n \; e^{-nTs} \; \frac{1 - e^{-sT}}{s} \; . \tag{28}$$

Hence Q(s) can be written, from (24) and (18), as

$$Q(s) = \frac{1 - e^{-sT}}{s} \; . \tag{29}$$

Higher order interpolators have been designed; however, it must be noted that $x_a(t)$, $nT \leq t < (n + 1)T$, cannot depend on the value of $x[(n + 1)T]$ in any physical system that can be represented by Figure 1. This means that true linear interpolation between points cannot be obtained nor can any higher order approximation.

The goal of these systems was, of course, to force $y_a(t)$ to equal $y(t)$ where $y(t) = L^{-1}[H(s)X(s)]$. This required $x_a(t)$ to equal $x(t)$. If the comparison of $y_a(t)$ and $y(t)$ was made only at the sampling instants, $t = nT$, the zero order hold achieved this goal if the input $x(t)$ was composed of step functions with the discontinuities of $x(t)$ occurring only at $t = nT^-$. Other inputs are inaccurately reconstructed in the interpolator. These sampled data systems will be referenced in later chapters.

The Laplace transform of $x^*(t)$ requires further examination; (22) is repeated here

$$X^*(s) = \sum_{n=0}^{\infty} x_n \; e^{-nsT} \; . \qquad (22)$$

Letting $s = \sigma + j\omega$ it is apparent that $X^*(s)$ is periodic in ω as

$$e^{\pm j2\pi n} = 1 \; ; \quad n = 0, 1, 2, \ldots . \qquad (30)$$

hence

$$e^{-nsT} = e^{-(\sigma+j\omega)nT} = e^{-\sigma nT} \; e^{-jn\omega T} = e^{-\sigma nT} e^{-jnT\left(\omega \pm \frac{2\pi}{T}\right)} \qquad (31)$$

and

$$X^*(s) = X^*\left(s \pm jn \, \frac{2\pi}{T}\right) . \qquad (32)$$

Another more interesting form of $X^*(s)$ is developed below:

If $\int_{-0}^{\infty} |x(t)|dt$ exists then, since $x(t) = 0$ for $t < 0$,

$$X^*(j\omega) = X^*(s)\Big|_{s=j\omega} \qquad (33)$$

and

$$X^*(j\omega) = \sum_{n=0}^{\infty} x_n \; e^{-nj\omega T} \; . \qquad (34)$$

The second form of $X^*(j\omega)$ may be obtained in the following manner. Rewrite (19)

$$x^*(t) = x(t) \sum_{n=0}^{\infty} \delta(t - nT) \tag{19}$$

or

$$x^*(t) = x(t)d(t) \tag{35}$$

where

$$d(t) = \sum_{n=0}^{\infty} \delta(t - nT) \ . \tag{36}$$

Now since d(t) is a periodic function, it may be expressed as a Fourier series

$$d(t) = \sum_{n=-\infty}^{\infty} C_n \, e^{j2\pi nt/T} \tag{37}$$

with

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} \sum_{n=0}^{\infty} \delta(t - nT) e^{-\frac{j2\pi t}{T}} \, dt \tag{38}$$

or

$$C_n = \frac{1}{T} \qquad \text{for all } n \ ; \tag{39}$$

thus

$$d(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j2\pi nt/T} \qquad (40)$$

and

$$x^{*}(t) = \frac{1}{T}\, x(t) \sum_{n=-\infty}^{\infty} e^{j2\pi nt/T} , \qquad (41)$$

now

$$X^{*}(j\omega) = \frac{1}{T} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(t) e^{j2\pi nt/T} e^{-j\omega t}\, dt . \qquad (42)$$

Reversing the order of the integral and summation gives

$$X^{*}(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) e^{j\left(\omega - \frac{2\pi n}{T}\right)t}\, dt \qquad (43)$$

or

$$X^{*}(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X\left[j\left(\omega - \frac{2\pi n}{T}\right)\right] . \qquad (44)$$

Now by arguments similar to those leading to (33)

$$X^{*}(s) = X^{*}(j\omega)\Big|_{j\omega=s} = \frac{1}{T} \sum_{n=-\infty}^{\infty} X\left(s - j\frac{2\pi n}{T}\right) . \qquad (45)$$

The form of (44) and (45) illustrate the problem of folding, or aliasing, that occurs in $X^{*}(j\omega)$ if $X(j\omega)$ is not zero

for $|\omega| > \pi/T$. For example, let $\omega_o = (2\pi/T)$ then, for $-\pi/T < \omega < \pi/T$

$$TX^*(j\omega) = \ldots . + X(j\omega - j\omega_o) + X(j\omega) + X(j\omega + j\omega_o) + \ldots . \tag{46}$$

Observe that a frequency component of $X(j\omega)$ at $\omega = \omega_1 \pm n\omega_o$ has exactly the same effect on $X^*(j\omega)$ as does a component at ω_1. These frequencies at $\omega = \omega_1 \pm n\omega_o$ are folded into, or aliased with, the frequency ω_1. Once these additional frequency components are a part of $X^*(j\omega)$, there is no way to identify them. Thus, unless $X(j\omega)$ is limited to the range $-\pi/T \leq \omega \leq \pi/T$, $X^*(j\omega)$ does not uniquely represent $x(t)$. This is simply another statement of the sampling theorem previously mentioned; the sampling frequency, $f_s = \frac{1}{T}$, must be equal to or greater than twice the highest frequency component of the sampled signal in order to uniquely identify the original signal from the samples.

2.4 Discrete Linear Systems[23]

By definition, the signals flowing in a discrete linear system are not continuous functions of time but only a sequence of samples of the continuous function. A digital transfer function must manipulate this incoming sequence in such a fashion as to produce the desired output sequence. The z-transform, briefly described below, will be used here for analysis of this discrete signal processing.

2.4.1 The z-Transform[24]

A few of the properties of the z-transform are given below, other properties and proofs may be found in most texts concerned with discrete systems. A most comprehensive discussion of z-transforms is found in Jury,[24] while others,[3,22,25,26] provide the information essential to this paper.

1. Definition: the z-transform of a sequence $\left\{f_n\right\}$ is defined to be

$$\tilde{F}(z) = Z\left\{f_n\right\} = \sum_{n=0}^{\infty} f_n z^{-n} \tag{47}$$

where $\left\{f_n\right\}$ is the sequence of samples of f(t) obtained at times t = nT

$$f_n = f(nT) \ , \tag{48}$$

thus (47) is often considered to be, and written as, the z-transform of f(t):

$$\tilde{F}(z) = Z[f(t)] = \sum_{n=0}^{\infty} f(nT) z^{-n} = \sum_{n=0}^{\infty} f_n z^{-n} \ . \tag{49}$$

If a jump discontinuity occurs in f(t) at t = nT, the value of f(nT) will be taken as that of $f(nT^+)$.

2. Shifting theorem: if $Z[f(t)] = \tilde{F}(z)$ then

$$Z[f(t + T)] = z[\tilde{F}(z) - f(0^+)] \tag{50}$$

$$Z[f(t + mT)] = z^{m}[\tilde{F}(z) - \sum_{k=0}^{m-1} f(kT)z^{-k}] \quad \text{for } m > 0 \tag{51}$$

and

$$Z[f(t - mT)u(t - mT)] = z^{-m}\tilde{F}(z) \tag{52}$$

where u(t - kT) is defined to be the unit step at time t = kT.

3. Initial and final values:

$$f(0) = \lim_{z\to\infty} \tilde{F}(z) \tag{53}$$

and

$$\lim_{n\to\infty} f(nT) = \lim_{z\to 1} (z - 1)\tilde{F}(z) \tag{54}$$

if the limit exists.

4. Convolution: If $f_1(t)$ and $f_2(t)$ have the z transforms $\tilde{F}_1(z)$ and $\tilde{F}_2(z)$, respectively, then

$$\tilde{F}_1(z)\tilde{F}_2(z) = Z\left\{\sum_{k=0}^{n} f_1(kT)f_2[(n - k)T]\right\} . \tag{55}$$

5. Inverse z-transform: If $Z\left\{f_n\right\} = \tilde{F}(z)$ then

$$f_n = Z^{-1}[\tilde{F}(z)] . \tag{56}$$

Several methods are available for obtaining the inverse z-transform, some of which are given here:

a. power series methods; by definition

$$\tilde{F}(z) = f_o + f_1 z^{-1} + f_2 z^{-2} + \ldots. \tag{57}$$

thus if $\tilde{F}(z)$ can be expressed as (57) then the coefficients, f_n, can be simply read off the series.

b. partial fraction expansion: if $\tilde{F}(z)$ is a rational function of z, analytic at ∞, it can be expressed by a partial fraction expansion

$$\tilde{F}(z) = \tilde{F}_1(z) + \tilde{F}_2(z) + \ldots. + \tilde{F}_m(z) \ . \tag{58}$$

The inverse of $\tilde{F}(z)$ can now be obtained as the sum of the individual inverses obtained from the expansion, that is

$$\begin{aligned} f(nT) &= Z^{-1}[\tilde{F}(z)] \\ &= Z^{-1}[\tilde{F}_1(z)] + Z^{-1}[\tilde{F}_2(z)] \\ &\quad + \ldots. + Z^{-1}[\tilde{F}_m(z)] \ . \end{aligned} \tag{59}$$

The inverse of the individual terms, $\tilde{F}_m(z)$, can usually be found from a table or from the power series.

c. complex integral formula:

$$f(nT) = \frac{1}{2\pi j} \oint_{\Gamma} \tilde{F}(z) z^{n-1} dz \tag{60}$$

where Γ is a path which encloses all the singularities of $\tilde{F}(z)$. If $\tilde{F}(z)$ has only

isolated singularities, Cauchy's integral formula yields

$$f(nT) = \text{sum of the residues of } [\tilde{F}(z)z^{n-1}] \ . \tag{61}$$

d. derivative form:

$$f(nT) = \frac{1}{n!}\left[\left(\frac{\partial}{\partial z^{-1}}\right)^n \tilde{F}(z)\right]_{z=\infty} \ . \tag{62}$$

The above, d., is described in a paper by Rader;[27] the other inversion techniques are fully described by Jury.[24] In practice the table look-up procedure of b. above, is generally used.

2.4.2 Linear Difference Equations

One of the most important uses of the z-transform is in the solution of m-th order linear difference equations as described by Gold and Rader.[28] The general m-th order difference equation is given by

$$y(nT) = \sum_{i=0}^{r} L_i x(nT - iT) - \sum_{i=1}^{m} K_i y(nT - iT) \ . \tag{63}$$

The form of (63) illustrates the iterative nature of the difference equation. The present output, $y(nT)$, depends on the m previous output values $y[(n-1)T]$, $y[(n-2)T]$ .... $y[(n-m)T]$ as well as the present and previous input values, $x(nT)$, $x[(n-1)T]$, $x[(n-2)T]$, ... $x[(n-r)T]$.

The z-transform may now be used to obtain a general solution for y(nT) in terms of the system and its input. For convenience here assume the system of (63) to be initially at rest so that all x's and y's are zero when the iteration begins. Rewrite (63) as

$$\sum_{i=0}^{m} K_i y(nT - iT) = \sum_{i=0}^{r} L_i x(nT - iT) \ , \tag{64}$$

$$K_o = 1 \ , \ r \leq n \ , \ m \leq n$$

and take the z-transform of both sides, giving

$$\sum_{i=0}^{m} K_i \sum_{n=0}^{\infty} y(nT - iT) z^{-n} = \sum_{i=0}^{r} L_i \sum_{n=0}^{\infty} x(nT - iT) z^{-n} \ . \tag{65}$$

Using the shifting theorem, (52), with all initial values set equal to 0, gives

$$\tilde{Y}(z) \sum_{i=0}^{m} K_i z^{-i} = \tilde{X}(z) \sum_{i=0}^{r} L_i z^{-i} \tag{66}$$

or

$$\tilde{Y}(z) = \frac{\sum_{i=0}^{r} L_i z^{-i}}{\sum_{i=0}^{m} K_i z^{-i}} \tilde{X}(z) = \tilde{H}(z) \tilde{X}(z) \ . \tag{67}$$

$\tilde{H}(z)$ is defined to be the discrete system transfer function just as H(s) is the continuous system transfer function. Note that $\tilde{H}(z)$ must be a rational function in z^{-1} in order to compute y(nT) as in (63).

2.4.3 The Digital Transfer Function

In this paper the primary importance of $\tilde{H}(z)$ is its interpretation as a frequency selective function. This allows the digital system to be compared in the frequency domain with H(jω), the continuous system transfer function. In order to illustrate this property assume that the discrete input, x(nT), is obtained by sampling a complex exponential wave

$$x(t) = e^{j\omega t} \quad , \; x(nT) = e^{j\omega nT} \; . \tag{68}$$

The solution for y(nT) of (63) can be written as

$$y(nT) = F(e^{j\omega T})e^{j\omega nT} \tag{69}$$

which can be verified by substituting (69) into (63)

$$F(e^{j\omega T})e^{j\omega nT} = \sum_{i=0}^{r} L_i \; e^{j\omega nT}e^{-j\omega iT}$$

$$- \sum_{i=1}^{m} K_i F(e^{j\omega T})e^{j\omega nT}e^{-j\omega iT} \tag{70}$$

or

$$F(e^{j\;T})e^{j\omega nT} = e^{j\omega nT}\left[\sum_{i=0}^{r} L_i\ e^{-j\omega iT} - \sum_{i=1}^{m} K_i F(e^{j\omega T})e^{-j\omega iT}\right] \qquad (71)$$

dividing through by $e^{j\omega nT}$ and rearranging terms gives

$$\sum_{i=0}^{m} K_i\ e^{-j\omega iT}\ F(e^{j\omega T}) = \sum_{i=0}^{r} L_i\ e^{-j\omega iT} \qquad (72)$$

or

$$F(e^{j\omega T}) = \frac{\displaystyle\sum_{i=0}^{r} L_i\ e^{-j\omega iT}}{\displaystyle\sum_{i=0}^{m} K_i\ e^{-j\omega iT}}\ . \qquad (73)$$

Now, on comparing (67) and (73), it is apparent that $\tilde{H}(e^{j\omega T}) = F(e^{j\omega T})$. Thus we can obtain the frequency response characteristics of any digital system transfer function $\tilde{H}(z)$ simply by replacing z by $e^{j\omega T}$ in $\tilde{H}(z)$. This frequency response function is a continuous function of frequency and will be represented by $\overline{H}(j\omega)$.

$$\overline{H}(j\omega) = \tilde{H}(z)\Big|_{z=e^{j\omega T}} = \tilde{H}(e^{j\omega T})\ . \qquad (74)$$

Similarly, by comparing (22) with (47) the continuous transfer function of the digital system, $\overline{H}(s)$ may be obtained from $\tilde{H}(z)$ by substituting e^{sT} for z in $\tilde{H}(z)$.

$$\overline{H}(s) = \tilde{H}(z)\Big|_{z=e^{sT}} = \tilde{H}(e^{sT}) \ . \tag{75}$$

Both $\overline{H}(j\omega)$ and $\overline{H}(s)$, $s = \sigma + j\omega$, are periodic functions of ω with a period of $\frac{2\pi}{T}$. This periodicity was previously noted in $X^*(s)$, the Laplace transform of $x^*(t)$ which in turn is the sequence of samples taken from $x(t)$ as shown in (22) and (32).

2.4.4 The Digital Impulse Function

Just as with the continuous system transfer function the digital transfer function can be characterized by its response to a "digital impulse function." This is always defined as the sequence $\{k, 0, 0, 0, \ldots.\}$ where k is the value of $x_o = x(0)$. In some references[29] k is chosen to be 1; however, Stearns[30] has shown that choosing $k = \frac{1}{T}$ is more meaningful in most cases. The continuous system impulse as defined in (15) is seen to have energy content constant with frequency, see (16), over all real frequencies. This is an acceptable and useful definition for continuous systems where in general the higher frequencies are attenuated by H(s). However, for a digital system where $\overline{H}(s)$ is periodic with frequency such a definition of the impulse is unacceptable as the frequency content of the output is limited neither by the input nor the transfer function. Since the period of the digital transfer function is $\frac{2\pi}{T}$ it would be more meaningful to define the digital impulse function as having a constant energy content over the frequency interval $-\pi/T \leq \omega \leq \pi/T$ and zero elsewhere. Just as the continuous system is uniquely defined by its

response to a signal whose frequency content is constant for all frequencies; the digital system can be uniquely defined by its response to a signal whose frequency content is constant over one period of the system transfer function and zero elsewhere. Hence the Laplace transform of the digital impulse function is given as

$$L[\text{digital impulse function}] = \begin{cases} c & -\pi/T \leq \omega \leq \pi/T \\ 0 & |\omega| > \pi/T \end{cases} \tag{76}$$

The digital impulse function, δ_D, can be written as

$$\delta_D = \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} c\ e^{j\omega T}\ d\omega = \frac{c}{2\pi}\ \frac{e^{j\pi t/T} - e^{-j\pi t/T}}{jt} \tag{77}$$

or

$$\delta_D = \frac{c}{T} \sin \pi t/T \big/ (\pi t/T)\ . \tag{78}$$

Now if c is chosen to be 1 as with the continuous impulse function, then δ_D is the sequence obtained from (78) evaluated at t = nT. Hence

$$x_o = \frac{1}{T}\ ; \qquad x_n = 0 \qquad n \neq 0 \tag{79}$$

and the digital impulse function is

$$\delta_D = \left\{\frac{1}{T}\ ,\ 0,\ 0,\ \ldots.,\ 0,\ \ldots.\right\}\ . \tag{80}$$

2.4.5 Stability of the Digital Transfer Function

The stability or instability of a continuous system whose transfer function, $H(s)$, can be expressed as a rational function of s is simply determined by finding the poles of $H(s)$. (Here a stable system is defined as being one for which the output is bounded for any bounded input.) If any poles lie in the right half s-plane the system is unstable, if all poles lie in the left half plane the system is stable. Multiple poles on the $j\omega$ axis cause the system to be unstable. Systems with single poles on the $j\omega$ axis are generally considered to be stable although a bounded input signal with a pole falling directly on a pole of the system transfer function will result in an unbounded output.

Similarly, the poles of the digital transfer function, $\tilde{H}(z)$, must lie inside the unit circle for $\tilde{H}(z)$ to be stable.[31] Again a single pole on the boundary, the unit circle, is usually acceptable in a stable system. As before, multiple poles on the unit circle or any pole outside the unit circle result in an unstable system.

2.4.6 Finite Register Length of Discrete Systems

A great deal of literature exists relating to the errors induced in digital signal processing due to the finite register length of any realizable machine. The results of most studies of these errors indicate that a given digital network should be realized as either a cascade or parallel combination of first and second order systems because these realizations

are the least sensitive to the errors induced by finite register length. Gold and Rader[32] describe these errors in detail. Related discussions are found in [33,34,35,36,37,38, 39,40].

2.5 Summary of the Simulation Problem

Figure 2 represents the continuous system to be simulated, Figure 3 represents the digital simulation system, and Figure 4 represents the continuous system whose characteristics are identical to those of the digital system.

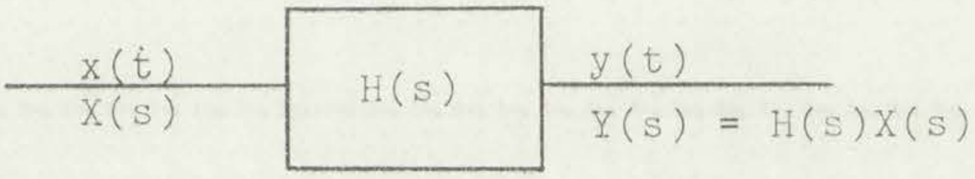


Figure 2

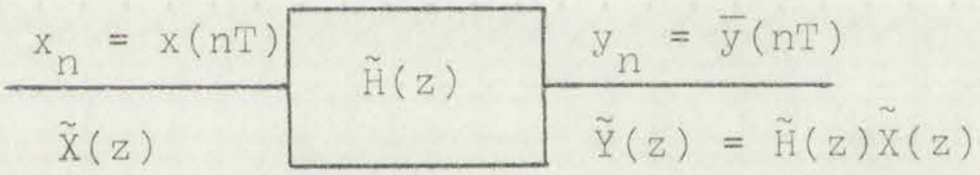


Figure 3

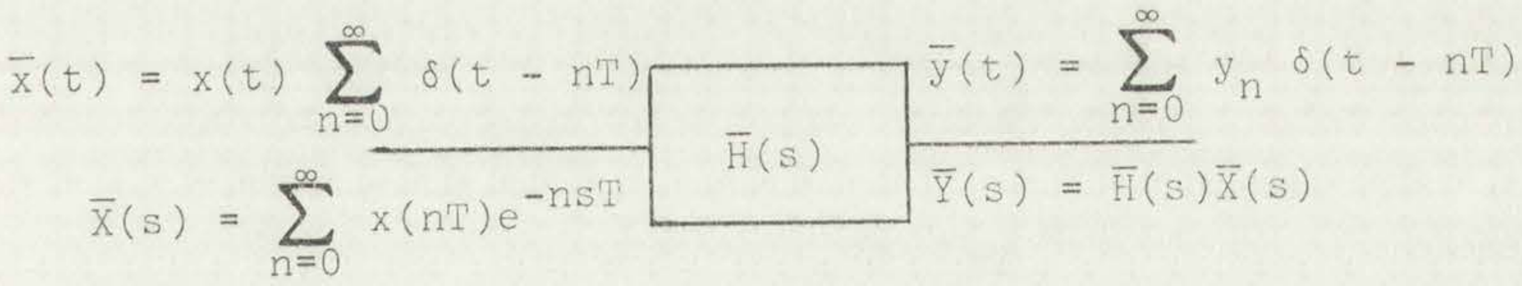

$\bar{x}(t) = x(t) \sum_{n=0}^{\infty} \delta(t - nT)$
$\bar{X}(s) = \sum_{n=0}^{\infty} x(nT)e^{-nsT}$
$\bar{H}(s)$
$\bar{y}(t) = \sum_{n=0}^{\infty} y_n \delta(t - nT)$
$\bar{Y}(s) = \bar{H}(s)\bar{X}(s)$

Figure 4

The goal of digital simulation is to find $\tilde{H}(z)$ such that the error in the approximation of samples of $y(t)$, given as $y(nT)$, by y_n is minimized for a given digital system within a specified amount of computational time.

Before proceeding to a discussion of simulation techniques it is necessary first to define the class of continuous transfer functions of interest here and then to describe the types of discrete transfer functions to be considered. Most physical processes are nonlinear functions of several variables and in fact the coefficients of these nonlinear terms are often functions of time as well. For mathematical analysis it is usually necessary to make several simplifying assumptions so that the physical process can be described in terms of ordinary, linear, time-invariant differential equations.

Equation (1) is an example of a simple form of such a mathematical description, a first order system. Equation (8) is the Laplace transform of (1) and is a rational function; that is, it is the ratio of two finite polynomials in s. The transfer function of a system describable by an ordinary, linear, time invariant differential equation is a rational function. Thus in this paper we are concerned only with transfer functions, H(s), which are rational functions. Let H(s) be represented as

$$H(s) = P(s)/Q(s) \ . \tag{81}$$

If the order of P(s) equals that of Q(s) then H(s) can be written as

$$H(s) = P(s)/Q(s) = a + R(s)/Q(s) \tag{82}$$

where

$$P(s) = aQ(s) + R(s) \ , \tag{83}$$

R(s) is of lower order than Q(s), and "a" is simply a constant. If P(s) is of higher order than Q(s) then "a" in (82) and (83) must be replaced by a polynomial P'(s) corresponding to successive levels of differentiation in the transfer function H(s). In this paper only transfer functions whose numerator is of order less than or equal to that of the denominator will be considered. With this restriction on H(s) either H(s) or R(s)/Q(s) as given in (82) is a proper fraction. Now, as shown by Karni,[41] a proper rational fraction can be expanded in partial fractions. Thus for the H(s) of interest here we may write H(s) as

$$H(s) = \frac{P(s)}{Q(s)} = a + \frac{k_i}{s - P_i} + \frac{k_{j,q}}{(s - P_j)^q}$$

$$+ \frac{k_{j,q-1}}{(s - P_j)^{q-1}} + \cdots + \frac{k_{j,1}}{(s - P_j)} \qquad (84)$$

where a is zero if H(s) itself is a proper fraction. Here we have shown only the terms due to a simple pole at P_i and a multiple pole at P_j. We require Q(s) to have real coefficients, thus if P_i is complex then P_i-conjugate = P_i' must also be a pole and the two terms are combined as

$$H(s) = \frac{\alpha_1}{(s - P_i)} + \frac{\alpha_2}{(s - P_i')}$$

$$= \frac{(\alpha_1 + \alpha_2)\left[s - \text{Real}(P_i)\right] + (\alpha_1 - \alpha_2)\text{Imag}(P_i)}{(s - P_i)(s - P_i')} . \qquad (85)$$

Another method of representing H(s) is in the form of a cascade of first and second order systems for real and complex poles, respectively.

$$H(s) = \frac{R(s)}{Q(s)} = \cdots \frac{b_i s + a_i}{(s - P_i)} \cdots \frac{c_j s^2 + b_j s + a_j}{(s - P_j)(s - P_j')} \cdots \qquad (86)$$

Thus any order H(s) may be expressed in partial fraction and/or cascade form with no terms of order greater than two. (If q in (84) is greater than two, the effected terms may be expressed as a product of two or more terms).

In order to limit the complexity of the systems discussed in the remainder of this paper only first and second order systems will be examined in detail. However, as shown above, the analysis can be simply extended to any order system. One further restriction placed on all transfer functions considered here is that they be stable.

The form of $\tilde{H}(z)$, the discrete transfer function, to be considered here is similarly restricted. $\tilde{H}(z)$ will be a rational function of z (or z^{-1}), and all coefficients of powers of z in $\tilde{H}(z)$ must be real since these are the coefficients of the corresponding linear difference equation of (63). A discussion of recursive and nonrecursive digital systems is delayed until the next section, we simply will note here that $\tilde{H}(z)$ is not required to be a proper fraction.

The advantage of forming $\tilde{H}(z)$ as a parallel or cascade combination of first and second order system was discussed in section 2.4.6. Here we will assume all discrete transfer

functions are a cascade of first and second order systems. In addition, we will consider only stable discrete systems as discussed in section 2.4.5.

All signals, continuous and discrete, are assumed to be zero for t less than zero.

The sampling rate $f_s = \frac{1}{T}$, is assumed to be constant.

2.6 A General Discussion of Simulation Techniques

As noted in the introduction, this paper is concerned neither with simulation languages nor the numerical integration techniques incorporated in these languages. The methods of interest here are those usually associated with a frequency domain approach to digital simulation, consequently the previous work to be discussed is primarily found in publications relating to digital filtering. A brief review of these frequency domain simulation (filtering) concepts follows. The linear difference equation, (63), is repeated here

$$y_n = \sum_{i=0}^{r} L_i x(nT - iT) - \sum_{i=1}^{m} K_i y(nT - iT) . \qquad (63)$$

Essentially all linear simulations must be of the form of (63) or reducible to this form. Thus the present discrete system output, y_n, is a linear combination of the present value of the input and past values of both the input and output. Some filtering techniques use future values of the input as well. However, in this paper we are concerned only with simu-

lating physically realizable systems and are therefore limited to using past and present sample values.

Digital filters are usually designated recursive or nonrecursive dependent upon the presence or absence, respectively, of any non-zero coefficient, K_i, in (63). Thus a recursive digital system depends on past values of both the input and output while with a nonrecursive design the output depends only on the present and past values of the input. Another separate method of distinguishing between two classes of filters is in terms of the duration of their impulse responses. These two classes are: Finite Impulse Response (FIR), and Infinite Impulse Response (IIR), whose impulse responses are of finite and infinite duration, respectively. Traditionally, a recursive filter is one whose transfer function contains both poles and zeroes and whose impulse response is of infinite duration. Similarly, the nonrecursive filter has been assumed to have zeroes only and an impulse response of finite duration. However, it has been shown[42,43] that FIR filters can be realized both recursively and nonrecursively as can IIR filters. Thus the distinction must be made between the filter synthesis procedure and its characteristics. In most cases the traditional definitions are correct. That is, most IIR filters are synthesized recursively and most FIR filters are synthesized nonrecursively.

Rader,[27] Kaiser,[44,45] and Rader and Gold[46] have written excellent reviews of the methods available to the digital filter designer. In addition, Gold and Rader[26] not only review

these methods but develop them in detail together with related subjects. In this paper the digital systems not of direct interest will only be briefly mentioned.

The FIR filter design techniques have been thoroughly investigated and are often implemented. In general the design goals are to obtain a digital transfer function characteristic which closely matches either some predetermined frequency response curve or the response of some simulated continuous transfer function. The advantages of the FIR method as discussed by Rabiner[47] are given below:

1. An FIR filter synthesized nonrecursively, as most are, is stable, unconditionally.
2. The effects of finite word length are usually insiginficant in nonrecursive FIR designs as there is no "cumulative effect" of roundoffs and quantizations.
3. FIR filters can approximate a desired magnitude frequency response curve with arbitrary accuracy while maintaining exactly linear phase.
4. FIR filters may be synthesized nonrecursively either by direct convolution or through high speed convolution using the fast Fourier Transform[48,49] and recursively using a comb filter and a bank of resonators.[50,51,52]

Three basic techniques of obtaining FIR filters are briefly reviewed here following the discussion in [47]. Kaiser[44] describes a windowing technique based on the non-recursive synthesis of a digital filter using the Fourier series coefficients of the desired frequency response as the coefficients of the digital filters impulse response. In general there are an infinite number of non-zero coefficients. Thus, to obtain an FIR filter the series must be truncated. This truncation leads to Gibbs oscillations at any discontinuities which, of course, are undesirable in the frequency response of the digital filter. To avoid this problem a weighting function, called a window, is used to modify these Fourier coefficients. Various forms of a window have been proposed in addition to that in [44]. Three of these are the Hamming and the Blackman discussed in [47] and a Dolph-Chebyshev window proposed by Helms.[50] The resulting digital system is of the form

$$y_n = \sum_{m=0}^{N-1} h_m x_{n-m} \tag{87}$$

where h_m, m = 0, 1, ... N - 1 are the coefficients obtained by multiplying the terms of the Fourier series found for the desired frequency response by the weighting function of the window. These coefficients are also the terms of the filters impulse response (if $x_o = 1$ for the impulse input). The realization of (87) is obtained either by direct convolution

or through high speed convolution. The resulting filter is an FIR of nonrecursive form.

Another commonly used method of finding the coefficients of an FIR is frequency sampling. The desired amplitude response is sampled at N equally spaced frequencies where N is the number of samples in the filter impulse response. These samples are then set equal in magnitude to the DFT (Discrete Fourier Transform) coefficients of the filter impulse response. In many cases some of the coefficients are left unspecified in the transition band between the passband and stop band of the filter and an optimization procedure is used in order to obtain the "best" filter in some sense. These FIR filters may be realized either recursively or nonrecursively. In general the above choice is based on speed; that is, which realization will accomplish the desired processing in the least time. Many papers have been published on frequency sampling techniques; two of the more recent ones are by Rabiner, Gold, and McConegal,[51] and Rabiner and Schafer.[52]

A third method of designing FIR filters is that of solving a system of nonlinear equations in order to obtain a minmax or Chebychev type approximation to the desired frequency response. Initially, a guess at the N coefficients of the nonrecursive filter is made. Some algorithm is then required to adjust these coefficients to produce a discrete transfer function whose frequency characteristics deviate from those of the desired function in an equal ripple fashion. Some of the

work in this area has been done by Hermann[53] while Parks and McCellan[54] offer a recent review of this method.

Four other papers that discuss related techniques of obtaining the coefficients of nonrecursive, FIR filters are [55,56,57,58]. A recent paper [59] suggests a technique for removing many of the coefficients (settimg them to zero) in a nonrecursive filter with the result that the direct convolution is faster than FFT convolution for some examples even though a large number of past input values (up to 300) are required. This is in sharp contrast to standard comparisons of the speed of direct convolution versus high speed convolution where a crossover point of about 30 coefficients is assumed. Filters with less than 30 coefficients should be synthesized directly while high speed convolution should be used for more complex filters. Stockham[49] shows a crossover number of 28 for one specific example.

The major disadvantage to nonrecursive filter designs is that in general many more terms are required to match a desired frequency response than are needed with recursive designs. In fact, Kaiser[44] has shown in one example that a nonrecursive design with 7200 terms was needed to obtain the equivalent cutoff characteristics of a 22nd order recursive filter.

As stated in the introduction, this paper is concerned with obtaining the best digital simulation of a continuous system for a given digital system complexity. Thus the advantages listed above for FIR filters of nonrecursive design must be given up in the interest of obtaining a relatively simple,

but accurate, digital system. In addition, the recursive designs to be considered in the remainder of this paper will be those of infinite duration impulse response, IIR.

The bilinear transformation, to be discussed in Chapter 3, and the impulse invariant approximation, to be described in Chapter 4, are the two most widely used recursive simulation techniques. The discussion of these methods will be delayed until the later chapters.

The unique properties of mirror image polynomials (MIP), described below, are used in the design of sin and/or tan filters. Only the amplitude response is simulated in this method; no phase information is considered. This method is applicable to the exact realization of digital filters with certain squared-magnitude functions. The restriction on the squared magnitude functions, $G(j\omega) = |H(j\omega)|^2$ is that ωT may appear in $G(j\omega)$ only as the argument of a squared trigonometric function such as $\sin^2(\omega T/2)$ or $\cos^2(\omega T/2)$. These functions may be combined only by addition, subtraction, multiplication, or division. Obviously any of the squared trigonometric functions are allowable. If $G(j\omega)$ is of this form then $G(z) = |\tilde{H}(z)|^2$ may be obtained directly from $G(j\omega)$ by replacing ωT by $-j \ln z$ or equivalently

$$z = e^{j\omega T} \tag{88}$$

thus, from (88)

$$\sin^2 \frac{\omega T}{2} \longrightarrow \frac{(z - 1)^2}{-4z} \tag{89}$$

$$\cos^2 \frac{\omega T}{2} \longrightarrow \frac{(z + 1)^2}{4z} \tag{90}$$

and the other substitutions are directly obtainable.

With the above restrictions on $G(j\omega)$, the substitution of (89) and (90) in $G(j\omega)$ to obtain $G(z)$ and the simplification of $G(z)$ to reduce it to the ratio of two polynomials in z, gives

$$G(z) = \frac{P(z)}{Q(z)} . \tag{91}$$

We can now examine each of these polynomials, P and Q. In general, $Q(z)$ can be written as

$$Q(z) = q_o z^n + q_1 z^{N-1} + \ldots. + q_1 z + q_o \tag{92}$$

which upon multipling by z^{-N} becomes

$$z^{-N}Q(z) = q_o + q_1 z^{-1} + \ldots. + q_1 z^{-(N-1)} + q_o z^{-N} . \tag{93}$$

On comparing (92) and (93) it is apparent that the roots of the polynomial in z^{-1} are the same as those of the polynomial in z. Consequently, the roots either occur in recpirocal pairs or are self-reciprocals (=±1). Since the coefficients of Q are assumed real, all roots must occur not only in

reciprocal pairs but also a complex root must occur as part of a complex conjugate pair.

It can be shown, [60], that the distance from any point on the unit circle to one of the above roots is in constant ratio to the distance to the conjugate reciprocal of this root (simply to the reciprocal root for real roots). Now for z on the unit circle, $z = e^{j\omega T}$, the magnitude function G(z) is equal to $|H(z)|^2$, where

$$H(z) = k \frac{P'(z)}{Q'(z)} \tag{94}$$

where P'(z) and Q'(z) are formed from the roots of P(z) and Q(z) which lie within or on the unit circle, respectively. Thus $\tilde{H}(z)$ is a stable digital filter whose magnitude frequency response $|\bar{H}(j\omega)|$ exactly matches that of $|H(j\omega)|$, $|\omega| < \pi/T$. This technique can be extended by approximating any desired squared magnitude frequency response arbitrarily closely by a function $G(j\omega)$ meeting the above restrictions. In addition to [26], several other related papers are [61,62,63]. One disadvantage of this method is that phase is not considered. Consequently, for many simulation applications where phase is important this technique is of little value. Another potential disadvantage is that poles may be located on or arbitrarily near the unit circle. As a result the transient response of such a system may decay very slowly which again is unacceptable in many simulations.

Golden[64] discusses the matched z-transform as well as the impulse invariant and bilinear techniques. The poles of the former method are obtained as for the impulse invariant approximation, $z = e^{sT}$. The zeroes of the matched z-transform are obtained from the continuous system via the same substitution. The advantage here is that the digital system is very easy to obtain, but more accurate methods are available.

Shanks[65] describes a method of obtaining a zero-phase digital processing system. The input sequence is processed through the filter, then this first step output sequence is reversed in time and processed through the filter again. The result of this second step is the desired output (in a time-reversed sequence). The phase characteristics of this process are zero for any filter while the magnitude vs. frequency response is the square of the filters magnitude response. Gibbs[66] also discusses this time reversal filtering with additional references.

Burrus and Parks[67] discuss a recursive filter design based on obtaining a desired impulse response, i.e., based on a time domain response criterion. The stability of such a filter is not discussed.

A recently described method of obtaining optimal digital simulation coefficients from generalized spline functions is that of DeFigueredo and Netravali.[68] In general, this method assumes the system input belongs to a certain class of inputs. A spline is fitted to the input samples and then, based on

this spline and the knowledge of the class of input samples, an optimal digital simulator is obtained.

Still another technique for obtaining the digital coefficients is that of considering the discrete system to be a device for obtaining numerical approximations to the convolution integral. This method is discussed in both [69] and [70] and varies in complexity from the rectangular approximation through the trapezoidal and quadratic to m-th order polynomial methods.

Other methods for obtaining these coefficients exist; however, only those of direct interest to this paper will be discussed in detail in the following chapters.

CHAPTER 3. FREQUENCY DOMAIN SIMULATION BY DIRECT SUBSTITUTION

The simulation techniques described here will determine $\tilde{H}(z)$ from $H(s)$ by substituting a rational function of z for s in $H(s)$. Since a rational function of a rational function is itself a rational function [30], $\tilde{H}(z)$ is thus a rational function of z.

The goal of these substitutions is, of course, to obtain $\tilde{H}(z)$ such that

$$\overline{H}(s) = \tilde{H}(e^{sT}) \tag{95}$$

very nearly approximates $H(s)$ in the principal range of s,

$$s = \alpha + j\omega \qquad -\pi/T \leq \omega \leq \pi/T \ . \tag{96}$$

To obtain $\overline{H}(s) = H(s)$ for $|\omega| < \pi/T$, it is necessary for $\tilde{H}(z)$ to be found from $H(s)$ by substituting $\frac{1}{T} \ell n\ z$ for s. Then

$$\tilde{H}(z) = H\left(\frac{1}{T} \ell n\ z\right) \tag{97}$$

and

$$\overline{H}(s) = \tilde{H}(e^{sT}) = H\left(\frac{1}{T} \ell n\ e^{sT}\right) = H(s) \ . \tag{98}$$

However, $\tilde{H}(z)$ must be a rational function of z (or z^{-1}) and (97) is rational in ($\ell n\ z$) rather than z. Obviously some

method of approximating ℓn z by a rational function of z is required; these techniques will be investigated in this chapter.

3.1 Tustin's Method

One of the earliest methods developed for the digital simulation of continuous systems was reported by Tustin.[44]

He proposed that a linear system, with inputs and responses being "serial numbers" or "multiplace numbers" representing the continuous waveforms, could be analyzed by "serial operators." The multiplace numbers are simply the input or output sequence of a discrete system, assumed to have a constant sampling interval. The basic algebraic operations involving two such numbers follow directly from the rules governing addition, subtraction, multiplication, and division of polynomials.

Time functions are approximated by joining the tips of the ordinates spaced T apart, as shown in Figure 5.

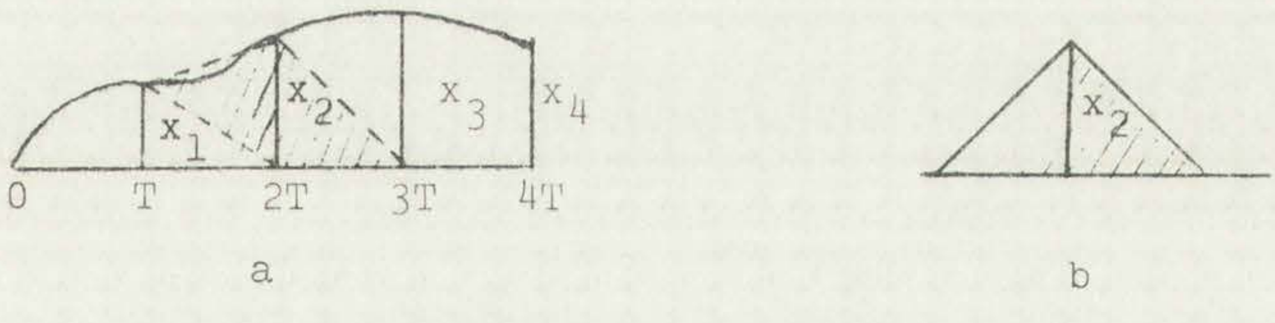


Figure 5. Area of Tustins' Δ-Elements

The area beneath the curve is approximated by a sum of "Δ-elements," one of which is shown by the shaded area in Figure 5a but can be considered to be the isosceles triangle of Figure 5b as the shaded areas are equal. A "Δ-element" of unit height is defined as a "Δ-unit."

Since the interest here was only in linear systems, the system response can be defined in terms of its response to a Δ-unit. Now inputs in "parallel" simply combine additively in the output while the response of two systems in series can be found by a discrete convolution. That is, if the output of the first system due to a Δ-unit input is the sequence d_0, d_1, d_2, ... while the corresponding output of the second system is the sequence e_0, e_1, e_2, ... then the output sequence f_0, f_1, f_2, ... from the two systems in series due to a Δ-unit input is

$$\begin{aligned} f_0 &= e_0 d_0 \\ f_1 &= e_0 d_1 + e_1 d_0 \\ f_2 &= e_0 d_2 + e_1 d_1 + e_2 d_0 \\ f_n &= \sum_{i=0}^{n} e_i d_{n-i} \ . \end{aligned} \tag{99}$$

This is obviously identical to the process of multiplying two polynomials

$$F(x) = D(x)E(x) \ . \tag{100}$$

From Figure 5a it is apparent that any function of time, x(t), composed of linear segments from t = nT to t = (n + 1)T can be represented as a sum of weighted Δ-units, shifted in time. This can be simply expressed as the sequence of weights; that is, let $F = \left\{f_0, f_1, \ldots\ldots f_n\right\}$ represent a finite length, piecewise linear function of time. In particular, let

$$F = \{0, 1, 2, 1, 0\} \tag{101}$$

represent f(t) where

$$f(t) = \begin{cases} 0 \quad , & t < 0 \\ t/T & 0 \leq t \leq 2T \\ 4 - t/T & 2T < t \leq 4T \\ 0 & t > 4T \end{cases} \tag{102}$$

Thus, f(t) can be visualized as a single Δ-unit at $t = T$ and at $t = 3T$ with a Δ-element of height 2 at $t = 2T$.

Any similar function of time can be constructed from Δ-units and represented as a serial number, just as a polynomial in x is constructed from powers of x. This similarity of serial numbers to polynomials implies that any serial number can be represented in terms of a product or quotient of two other serial numbers. In most cases the representation is not exact, just as with division where there is normally a remainder. However, the approximation may be carried to any desired level of accuracy.

Assume that a given signal D(t) is applied to a system, whose response is then R(t) and that $D = \left\{d_0, d_1, d_2, \ldots.\right\}$ and $R = \left\{r_0, r_1, r_2, \ldots.\right\}$ are known. Now the response of the system per Δ-unit is $R/D = A = \left\{a_0, a_1, a_2, \ldots.\right\}$. The response to any other time function D_1 is given by $(R/D)D_1$. (R/D) is a constant of the system and independent of the units used (Δ-units above). Assume the units used were U-units, then in place

of R, the response would be $R/U = R_u$ and D would be replaced by $D/U = D_u$.

The system could then be represented by

$$R_u/D_u = (R/u)\big/(D/u) = R/D \tag{103}$$

as before. Thus, the series that represents the relation between the input and output series is independent of the units used to describe the time function.

For example, consider a closed loop control system whose transfer function is given by

$$H_T(s) = \frac{H(s)}{1 + G(s)H(s)} \tag{104}$$

where

$$Y(s) = \frac{H(s)}{1 + G(s)H(s)} \quad X(s) = H_T(s)X(s) \ . \tag{105}$$

To approximate this system by Tustin's serial numbers proceed as follows:

1. Find the serial number $H = \left\{h_0, h_1, h_2, \ldots.\right\}$, specifying the response of H(s) per Δ-unit input.
2. As in 1, above, find $G = \left\{g_0, g_1, g_2, \ldots.\right\}$
3. By definition $1 = \{1, 0, 0, 0, \ldots.\}$
4. Find the discrete transfer function approximating (104)

$$\tilde{H}_T = \frac{\left\{h_0, h_1, h_2 \ldots.\right\}}{\{1, 0, 0, 0 \ldots.\} + \left\{g_0, g_1, g_2 \ldots.\right\}\left\{h_0, h_1, h_2 \ldots.\right\}} \tag{106}$$

5. From a given sequence representing x

$$X = \left\{x_0, x_1, x_2, \ldots.\right\} \tag{107}$$

find the sequence for Y

$$Y = \left\{y_0, y_1, y_2, \ldots.\right\} \tag{108}$$

where

$$\left\{y_0, y_1, \ldots.\right\} = \frac{\left\{h_0 \ldots.\right\}\left\{x_0 \ldots.\right\}}{\{1, 0, \ldots.\} + \left\{g_0 \ldots.\right\}\left\{h_0 \ldots.\right\}} . \tag{109}$$

Tustin shows that the "transfer function" given in (106) is stable if the transfer function of (104) is stable.

One of the most interesting aspects of Tustin's work is the development of differentiating, (ρ), and integrating, ($\frac{1}{\rho}$), operators. Consider a function of time f(t) defined as

$$f(t) = \begin{cases} 0 & t < 0 \\ t/T & 0 \leq t \leq T \\ 2 - t/T & T < t \leq 2T \\ 0 & t > 2T \end{cases} \tag{110}$$

that is f(t) is a Δ-unit represented by the sequence $F = \{1, 0, 0, 0, 0, 0, \ldots.\}$.

The integral of f(t) is represented by

$$\int_0^t f(r)dr = \frac{1}{\rho} f(t) \; . \tag{111}$$

For f(t) as defined in (110).

$$\int_0^T f(r)dr = \frac{1}{2} T \tag{112}$$

$$\int_0^{2T} f(r)dr = T \tag{113}$$

$$\int_0^t f(r)dr = T \qquad t > 2T \; . \tag{114}$$

Therefore,

$$\frac{1}{\rho} F = T\left\{\frac{1}{2}, 1, 1, 1, \ldots.\right\} \; . \tag{115}$$

Thus, the integrating operator written as a serial number, hereafter called the integrating serial operator $(1/\rho)$, must multiply the sequence $F = \{1, 0, 0, 0, \ldots.\}$ to obtain the sequence $\left(\frac{1}{\rho}\right)F = \frac{T}{2}\{1, 2, 2, 2, \ldots.\}$. Thus,

$$\left(\frac{1}{\rho}\right) = \frac{T}{2}\{1, 2, 2, \ldots.\} \; . \tag{116}$$

The differentiating operator (ρ) is obtained simply by inverting (116)

$$\rho = \frac{2}{T} \frac{1}{\{1, 2, 2, \ldots.\}} = \frac{2}{T} \frac{\{1, -1\}}{\{1, 1\}} = \frac{2}{T}\{1, -2, 2, -2, \ldots.\} \tag{117}$$

The suggestion is now made that instead of finding the discrete transfer function in the manner used to obtain (106) from an equation of the form of (104) a direct substitution should be made in (104). Just as s is the continuous differentiating operator in (104); $\frac{2}{T}\frac{\{1,\ -1\}}{\{1,\ 1\}}$ is the discrete differentiating operator in (117). Thus, to obtain the discrete transfer function given in (106) the simple substitution

$$\frac{2}{T}\frac{\{1,\ -1\}}{\{1,\ 1\}} \longrightarrow s \tag{118}$$

is made in (104). Before making this substitution (104) must be simplified to a rational function in s.

In order to understand (118), consider a simple transfer function

$$H(s) = \frac{1}{s + 1} \tag{119}$$

where

$$Y(s) = H(s)X(s) \ . \tag{120}$$

Given an input sequence, represented by

$$X = \left\{x_0,\ x_1,\ x_2,\ \ldots.\right\} \tag{121}$$

instead of a continuous function x(t), requires a discrete transfer function $\tilde{H}$ instead of H(s). From (118) $\tilde{H}$ is obtained from H(s) below

$$\tilde{H} = \frac{1}{s+1}\bigg|_{s \to \frac{2}{T}\frac{\{1,\ -1\}}{\{1,\ 1\}}} = \frac{\{1,\ 1\}}{\{1,\ 1\} + \frac{2}{T}\{1,\ -1\}} = \frac{\{1,\ 1\}}{\left\{1 + \frac{2}{T},\ 1 - \frac{2}{T}\right\}} \tag{122}$$

Now $Y = \left\{y_0,\ y_1\ \ldots .\right\}$ is found from (122) as

$$\left\{y_0,\ y_1,\ y_2\ \ldots .\right\} = \frac{\{1,\ 1\}}{\left\{1 + \frac{2}{T},\ 1 - \frac{2}{T}\right\}}\left\{x_0,\ x_1,\ x_2\ \ldots .\right\} \tag{123}$$

or

$$\left\{1 + \frac{2}{T},\ 1 - \frac{2}{T}\right\}\left\{y_0,\ y_1,\ \ldots .\right\} = \{1,\ 1\}\left\{x_0,\ x_1,\ x_2\ \ldots .\right\} \tag{124}$$

By expanding (124) for some $n > 1$ the computational algorithm for y_n is obtained

$$\left(1 + \frac{2}{T}\right)y_n + \left(1 - \frac{2}{T}\right)y_{n-1} = x_n + x_{n-1} \tag{125}$$

or

$$y_n = \frac{T}{T+2}\left[x_n + x_{n-1} - \frac{T-2}{T}\ y_{n-1}\right] . \tag{126}$$

Using the z-transform of a difference equation as given in (64) through (67), (126) can be written as

$$\left[\left(1 + \frac{2}{T}\right) + \left(1 - \frac{2}{T}\right)z^{-1}\right]\tilde{Y}(z) = \left[1 + z^{-1}\right]\tilde{X}(z) \tag{127}$$

or

$$\tilde{Y}(z) = \frac{1 + z^{-1}}{\left(1 + \frac{2}{T}\right) + \left(1 - \frac{2}{T}\right)z^{-1}}\ \tilde{X}(z) = \tilde{H}(z)\tilde{X}(z)\ . \tag{128}$$

$\tilde{H}(z)$ can be rewritten as

$$\tilde{H}(z) = \frac{1 + z^{-1}}{(1 + z^{-1}) + \frac{2}{T}(1 - z^{-1})} = \frac{1}{1 + \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)} \tag{129}$$

or

$$\tilde{H}(z) = H(s)\Big|_{s \to \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}} = H\left[\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right]. \tag{130}$$

Thus, Tustin's method, in z-transform notation, provides $\tilde{H}(z)$ from $H(s)$ by the simple substitution of

$$\frac{2}{T}\left(\frac{z - 1}{z + 1}\right) \quad \text{for } s \tag{131}$$

in $H(s)$. This is commonly known as the bilinear transformation and will be discussed in detail in the next section.

3.2 The Bilinear Transformation

The bilinear transformation is widely used to obtain a discrete rational transfer function from a desired continuous rational transfer function. In many cases the discrete function is a very good approximation of the continuous function and in many others it is acceptable to the user. The bilinear transformation possesses several characteristics, listed below, which enhance its usefulness:

1. It is the simplest transformation from the s-plane to the z-plane which maps the left half s-plane into the unit circle of the

z-plane, the $j\omega$-axis of the s-plane onto the z-plane unit circle, and the right-hand s-plane onto the exterior of the unit circle on the z-plane. Thus, stable continuous systems are transformed into stable discrete systems.

2. It is very simple to use:

$$\tilde{H}(z) = H(s)\Big|_{s \to \frac{2}{T}\left(\frac{z-1}{z+1}\right)} = \left[H\ \frac{2}{T}\left(\frac{z-1}{z+1}\right)\right] . \qquad (130)$$

3. The frequency response of the discrete transfer function $\overline{H}(j\omega)$ is obtained directly from H(s), as shown by Stearns,[30] and below

$$\overline{H}(j\omega) = \tilde{H}(e^{j\omega T}) = H\left[\frac{2}{T}\left(\frac{e^{j\omega T}-1}{e^{j\omega T}+1}\right)\right] \qquad (132)$$

from (75) and (130). Now

$$j\ \tan\frac{\omega T}{2} = \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{e^{j\omega T/2} + e^{-j\omega T/2}} = \frac{e^{j\omega T}-1}{e^{j\omega T}+1} . \qquad (133)$$

Thus,

$$\overline{H}(j\omega) = H\left(j\ \frac{2}{T}\tan\frac{\omega T}{2}\right) . \qquad (134)$$

The form of (134) shows that $\overline{H}(j\omega)$ is not only simple to obtain from $H(j\omega)$ but is in

in fact equal to $H(j\omega_1)$, where ω_1 is a "warped" frequency scale.

$$\bar{H}(j\omega) = H(j\omega_1) \ , \quad \omega_1 = \frac{2}{T} \tan \frac{\omega T}{2} \ . \tag{135}$$

The frequency scale is not only warped, it is also compressed, so that for ω in the principal range

$$-\pi/T \leq \omega \leq \pi/T \ . \tag{136}$$

ω_1 continuously covers the entire spectrum of frequencies

$$-\infty < \omega_1 < \infty \ . \tag{137}$$

$\bar{H}(j\omega)$, $-\pi/T < \omega < \pi/T$ must therefore cover the entire domain of $H(j\omega)$, $-\infty < \omega < \infty$. Now if $H(s)$ is a proper rational function of s, then for $s = j\omega$

$$\lim_{\omega \to \pm\infty} |H(j\omega)| = 0 \tag{138}$$

which implies

$$\lim_{\omega \to \pm\pi/T} |\bar{H}(j\omega)| = 0 \ . \tag{139}$$

Hence the problem of folding, or aliasing, is completely eliminated in the sense it was described in (45) and (46).[44]

4. The poles in the z-plane, z_p, can be found from the s-plane poles, s_p, directly

$$s_p = \frac{2\left(z_p - 1\right)}{T\left(z_p + 1\right)}, \qquad z_p = \frac{2 + Ts_p}{2 - Ts_p} \tag{140}$$

for $s_p = \sigma_p + j\omega_p$, $\left|z_p\right|$ is given by

$$\left|z_p\right| = \left|\frac{2 + T\sigma_p + Tj\omega_p}{2 - T\sigma_p - Tj\omega_p}\right| = \left[\frac{(2 + T\sigma_p)^2 + (T\omega_p)^2}{(2 - T\sigma_p)^2 + (T\omega_p)^2}\right]^{\frac{1}{2}} . \tag{141}$$

It follows that any continuous system pole with $\sigma_p < 0$ transforms into a discrete system pole with $\left|z_p\right| < 1$. In addition, note that points on the $j\omega$ axis in the s-plane, $s = j\omega$, map directly onto the unit circle $|z| = 1$, in the z-plane. Thus, the bilinear transformation always transforms a stable continuous system into a stable discrete system.

5. The dc gain of the digital transfer function is identical to that of the simulated continuous system. This follows immediately from (134) as $\tan(0) = 0$, giving $\overline{H}(0) = H(0)$.

The coefficient $\frac{2}{T}$ was used by Tustin as shown in (131) and can also be found from an expansion of $\tan \frac{\omega T}{2}$ in Taylor's series about $\omega = 0$.

$$\tan \frac{\omega T}{2} = \tan 0 + \omega_1 \frac{d}{d\omega} \tan \frac{\omega T}{2} + \frac{\omega^2}{2!} \frac{d^2}{d\omega^2} \tan \frac{\omega T}{2} + \ldots. \qquad (142)$$

$$= 0 + \frac{\omega T}{2} + \ldots.$$

Thus, to a first approximation for ω near 0

$$\omega \cong \frac{2}{T} \tan \frac{\omega T}{2} . \qquad (143)$$

However, if at some critical frequency ω_o, $\omega_o < \pi/T$ the digital response must be identical to that of the continuous system,this coefficient may be adjusted so that

$$\omega_o = a \tan \frac{\omega_o T}{2} \qquad (144)$$

and

$$\overline{H}(j\omega_o) = H\left(ja \tan \frac{\omega_o T}{2}\right) = H(j\omega_o) . \qquad (145)$$

$\overline{H}(j\omega)$ is now an exact simulation of $H(j\omega)$ for $\omega = 0$ and for $\omega = \omega_o$. An interesting and often useful result of this coefficient adjustment occurs in digital filters. For example, consider a low pass filter whose transfer function is

$$H(j\omega) = \frac{1}{1 + j\omega} \qquad (146)$$

and the critical frequency is chosen to be at the 3 dB point, $\omega_o = 1$. From (144) "a" is found to be

$$a = \frac{\omega_o}{\tan \frac{\omega_o T}{2}} = \frac{1}{\tan T/2} = \text{ctn } T/2 , \qquad (147)$$

resulting in

$$\overline{H}(j\omega) = \frac{1}{1 + j \operatorname{ctn} T/2 \tan \frac{\omega T}{2}}. \tag{148}$$

The transfer functions $\overline{H}(j\omega)$ and $H(j\omega)$ are equal for $\omega = 0$ and $\omega = 1$ as desired; however, for $|\omega| < 1$ we have

$$(\operatorname{ctn} T/2)\left(\tan \frac{\omega T}{2}\right) < \omega \quad , \qquad |\overline{H}(j\omega)| > |H(j\omega)| \tag{149}$$

and, for $1 < \omega < \pi/T$

$$(\operatorname{ctn} T/2)\left(\tan \frac{\omega T}{2}\right) > \omega \ , \qquad |\overline{H}(j\omega)| < |H(j\omega)| \ . \tag{150}$$

Thus, the digital filter has a sharper cutoff than the continuous filter it was derived from.

Kaiser[44] describes a technique called "prewarping" used in the design of digital filters with the bilinear transformation. Essentially this consists of choosing a type of filter, such as Butterworth or Chebychev, to meet certain requirements at specified critical frequencies. These critical frequencies, ω_i, are then "prewarped," $\omega_i' = \tan \omega_i T/2$ and the continuous filter is designed for the new critical frequencies, ω_i'. The digital filter is then obtained by the substitution

$$\frac{z - 1}{z + 1} \longrightarrow s \tag{151}$$

in the continuous filter $H(s)$. Now

$$\overline{H}(j\omega) = H\left(j \tan \frac{\omega T}{2}\right) \tag{152}$$

and for the critical frequencies ω_i

$$\overline{H}(j\omega_i) = H\left(j \tan \frac{\omega_i T}{2}\right) = H(j\omega_i') \ . \tag{153}$$

Hence, the "prewarping" of the continuous filter frequency scale results in the desired characteristics for the digital filter.

Although these are digital filter design techniques they are useful in many simulation problems.

3.3 The Method of Rich and Shaw[5,6,7]

Consider the differential equation

$$P(D)y(t) = \alpha Q(D)x(t) \tag{154}$$

where D is the differential operator Df(t) = df/dt and P and Q are polynomials of degrees k and k', respectively, each with a leading coefficient of unity. Here x(t) is a known function and is the system input; y(t) is the output.

Let the roots of P(z) be ρ_j, j = 1, 2, ... k, and the roots of Q(z) be ρ_j', j = 1, 2, ... k'. Then

$$P(D) = \prod_{j=1}^{k} (D - \rho_j) \tag{155}$$

and

$$Q(D) = \prod_{j=1}^{k'} (D - \rho_j'). \tag{156}$$

Now construct the polynomials

$$P^*(E) = \prod_{j=1}^{k} \left(E - e^{\rho_j T}\right) \tag{157}$$

and

$$Q^*(E) = \prod_{j=1}^{k'} \left(E - e^{\rho_j' T}\right) \tag{158}$$

where E is the shifting operator,

$$E[f(t)] = f(t + T) \tag{159}$$

and T is the sampling interval.

The difference equation corresponding to the differential equation of (154) is obtained from the "substitution rule"[5] as

$$P^*(E)y_n = \alpha\left[TE^{\frac{1}{2}}\right]^{k-k'} Q^*(E)x_n \ . \tag{160}$$

Since k is the order of the polynomial $P^*(E)$ and k' is the order of $Q^*(E)$, k must not be less than k' to be consistent with the previous requirements that $\tilde{H}(z)$ shall not have excess zeroes. Now if (k' - k) is not an even number, then values $x_{n-\frac{1}{2}}$, $x_{n+\frac{1}{2}}$, etc., are required by the form of (160). This implies that the sampling rate must be doubled, which in general is impractical. If (k - k') is odd then the term $E^{n+\frac{1}{2}}$, n = (k - k' - 1)/2 can be modified to E^n or E^{n+1}, resulting in a phase shift in the transfer function. A simple example of this method follows.

Assume the differential equation defining a continuous system is

$$D^2y(t) + 3Dy(t) + 2y(t) = ax(t) . \tag{161}$$

The Laplace transform of this system gives

$$(s^2 + 3s + 2)Y(s) = aX(s) \qquad y(0) = y'(0) = 0 \tag{162}$$

or

$$Y(s) = \frac{aX(s)}{s^2 + 3s + 2} \tag{163}$$

which can be written as

$$Y(s) = H(s)X(s) , \qquad H(s) = \frac{a}{s^2 + 3s + 2} . \tag{164}$$

H(s) can be rewritten as

$$H(s) = a\left[\frac{1}{s + 1} - \frac{1}{s + 2}\right] . \tag{165}$$

Applying the substitution rule to (161) gives

$$(E - e^{-2T})(E - e^{-T})y_n = aT^2Ex_n \tag{166}$$

or

$$\left(E^2 - (e^{-2T} + e^{-T})E + e^{-3T}\right)y_n = aT^2Ex_n . \tag{167}$$

Multiplying both sides of (167) by E^{-2} gives

$$y_n - (e^{-2T} + e^{-T})y_{n-1} + e^{-3T}y_{n-2} = aT^2x_{n-1} \tag{168}$$

or

$$y_n = aT^2x_{n-1} + (e^{-2T} + e^{-T})y_{n-1} - e^{-3T}y_{n-2} \ . \tag{169}$$

Refer back to (164) and assume that the input is a unit impulse, $X(s) = 1$. Then $Y(s) = H(s)$ and, from (165)

$$y(t) = a(e^{-t} - e^{-2t}) \tag{170}$$

and it follows that

$$\tilde{Y}(z) = Z[y(t)] = a\left(\frac{z}{z - e^{-T}} - \frac{z}{z - e^{-2T}}\right)$$

$$= a\,\frac{z(e^{-T} - e^{-2T})}{(z - e^{-T})(z - e^{-2T})} \ . \tag{171}$$

Now if $\tilde{X}(z) = \frac{1}{T}$, the discrete impulse, then

$$\tilde{H}(z) = T\tilde{Y}(z) = \frac{aTz(e^{-T} - e^{-2T})}{(z - e^{-T})(z - e^{-2T})} \ . \tag{172}$$

Since

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) \tag{173}$$

we have

$$\tilde{Y}(z)\left(z^2 - (e^{-T} + e^{-2T})z + e^{-3T}\right) = aTz(e^{-T} - e^{-2T})\tilde{X}(z) \ . \qquad (174)$$

Multiplying both sides of (174) by z^{-2} gives

$$\tilde{Y}(z)\left(1 - (e^{-T} + e^{-2T})z^{-1} + e^{-3T}z^{-2}\right) = aTz^{-1}(e^{-T} - e^{-2T})\tilde{X}(z) \ . \qquad (175)$$

Equation (175) is the "impulse invariant" approximation which will be described in the next chapter. Using the first two terms of the Taylor series expansion of e^x

$$e^x = 1 + x + \frac{x^2}{2!} + \ldots . \qquad (176)$$

gives, for $e^{-T} - e^{-2T}$

$$e^{-T} - e^{-2T} \cong \left(1 - T + \frac{T^2}{2!}\right) - \left(1 - 2T + \frac{4T^2}{2!}\right) \qquad (177)$$

or

$$(e^{-T} - e^{-2T}) \cong T \ . \qquad (178)$$

Substituting (178) into the right side of (175) gives the z-transform equivalent of (169).

Thus, the method of Rich and Shaw results in a difference equation similar to that obtained by the impulse invariant technique for the second order equation of (161). Golden[64] discusses a somewhat similar procedure.

3.4 z-Forms

The use of z-forms, introduced by Boxer and Thaler,[8] is not, by the previous definitions, a digital simulation technique. However, the analysis is of interest here and the method provides a means of numerically finding the response of a continuous system to a continuous input.

Assume a rational input function, X(s), is given, then

$$Y(s) = H(s)X(s) \tag{179}$$

which can be expressed as

$$Y(s) = \frac{\sum_{i=0}^{n} a_i s^i}{\sum_{i=0}^{m} b_i s^i} \qquad m > n\ . \tag{180}$$

The actual response y(t) may be approximated at times $t = nT$, $n = 0, 1, \ldots.$ by y_n obtained by the numerical inversion of (180). The method used to obtain y_n is given below:

1. Divide the numerator and denominator of (180) by s^m in order to express Y(s) as a rational function of s^{-1}.
2. Substitute for s^{-k}, $k = 1, 2, \ldots.\ m$, a rational fraction in powers of z^{-1} obtained from a table of z-forms and rearrange Y(s) as a rational fraction in powers of z^{-1}.
3. Divide the resulting expression by T.

4. Expand the fraction by synthetic division into a series of the form

$$y_0 + y_1 z^{-1} + y_2 z^{-2} + \dots + y_n z^{-n} + \dots \tag{181}$$

where y_n is the approximate value of $y(nT)$.

A brief summary of this method will be presented here. The inverse Laplace transform of $Y(s)$, $L^{-1}[Y(s)]$ is

$$y(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} Y(s)e^{st}\, ds \tag{182}$$

where all poles of $Y(s)$ lie to the left of c. For stable systems c can be taken to be 0. Equation (182) can be rewritten as

$$y(t) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} Y(s)e^{st}\, ds \tag{183}$$

which can be approximated by

$$y(t) \cong \frac{1}{2\pi j}\int_{-j\pi/T}^{j\pi/T} Y(s)e^{st}\, ds\ . \tag{184}$$

Replace s by $\frac{1}{T}\,\ell n\ z$ and obtain

$$y_n = \frac{1}{2\pi j}\oint \frac{1}{T}\, Y\left(\frac{1}{T}\,\ell n\ z\right) z^{n-1}\, dz \tag{185}$$

which, except for a factor of $\frac{1}{T}$, is the same equation as the inverse z-transforms given in (60) for principal values of the logarithm. Now with $z = e^{sT}$

$$\frac{1}{T} \ln z = \frac{1}{T} \ln e^{sT} = s \; . \tag{186}$$

Thus, the inverse transform of $Y\left(\frac{1}{T} \ln z\right)$ is the approximation desired. $Y(s)$ is a rational function in s^{-1} and therefore $Y\left(\frac{1}{T} \ln z\right)$ is a rational function in $T/\ln z$. A suitable expansion for $\ln z$ is

$$\ln z = 2\left[u + \frac{1}{3} u^3 + \frac{1}{5} u^5 + \ldots.\right] \tag{187}$$

where

$$u = \frac{z - 1}{z + 1} \; . \tag{188}$$

Now $T/\ln z$ can be expressed as a Laurent Series

$$\frac{1}{s} = T/\ln z \tag{189}$$

$$= T/2\left[\frac{1}{u} - \frac{1}{3} u - \frac{4}{45} u^3 \; \ldots.\right] \tag{190}$$

z-form tables are constructed by retaining the principal part and the constant term of the series of (190).

Steps 2, 3, and 4 may now be followed directly to obtain y_n.

This technique is directly applicable only to systems with zero initial conditions. A later paper by Boxer[71] includes initial conditions by expanding $y(t)$ about $y(0)$ in Taylor's series.

Further details of the method, extended tables and examples may be found in [24] and [72]. Kaiser[44] discusses the problems of applying this method directly to the synthesis of digital filters.

CHAPTER 4. INVARIANT SIMULATIONS

The standard z-transform or, equivalently, the impulse invariant method of obtaining a digital simulation system is well documented.[26,44,73,74,75] The extension of these methods to higher order invariant digital systems has been discussed by Wait[75,76] as well as by others.[74,77]

In this chapter the impulse, step, and ramp invariant simulations will be defined and developed. An invariant simulation is defined here to be one in which the discrete system output values, y_n, agree exactly with corresponding samples of the simulated continuous system output, $y(nT)$, for an input of specified form. For example: the impulse invariant simulation of a continuous system, $H(s)$, generates an output sequence $\left\{y_0, y_1, y_2, \ldots.\right\}$ identical to that sequence obtained by sampling the impulse response of $H(s)$, $\left\{y(0), y(T), y(2T) \ldots.\right\}$ when excited by the discrete impulse function. However, for other inputs the simulation is not exact. Similarly, the step and ramp invariant simulations are exact only for step and ramp inputs, respectively. Thus, for invariance, the impulse inputs must be of the form $a_k\delta(t - kT)$, the step inputs of the form $a_k u(t - kT)$ and ramp inputs of the form $a_k(t - kT)u(t - kT)$, $k = 0, 1, 2, 3, \ldots.$

These simulations are exact in spite of the sampling frequency and the high frequency energy content of the input signal. With the form of the input specified and its value at

the sample times known, the input signal is actually completely defined as a continuous function of time. Thus, effectively the invariant simulation has an infinite sample rate, for the invariant input, as it is continuously known. For other unknown input forms the effective sample rate is 1/T.

These invariant simulations have evolved from the interpolators of the sampled data systems shown in Figure 1. For example, an interpolator with $Q(s) = 1$ would be impulse invariant while the zero order hold discussed in Chapter 2 and later in this chapter with $Q(s) = (1 - e^{-sT})/s$ is step invariant. These and other higher order "holds" are discussed in most of the previously referenced texts on sampled data systems and in particular by Jury.[24] The disadvantage of these physically realizable interpolators is that true linear interpolation (first order hold) cannot be obtained, nor can any higher order hold. This is easily shown for the first order case where

$$x(t) = x(nT) + \big(x[(n + 1)T] - x(nT)\big)(t - nT)/T \qquad (191)$$

for $nT < t < (n + 1)T$. Thus, $x(t)$ for $t < (n + 1)T$ depends on the value of $x[(n + 1)T]$ which is not yet available.

The linear or ramp invariant method for a purely digital system can achieve this true linear input simulation since these systems require no interpolation of the input between input samples. The input, $x(t)$, is required only at the sampling instants, $t = nT$, to obtain $x_n = x(nT)$.

Second and higher order invariant simulations can be obtained. However, the result is often an unstable digital system simulating a stable continuous system.

Previous references to step and ramp invariant simulations have been made, as mentioned above, as an outgrowth of the zero and first order hold circuits of sampled data systems. In this paper we define these invariant simulations for the purely digital system without regard to the hold circuits or physical realizability.

4.1 The Impulse Invariant Simulation

The impulse response of a continuous system is found, as usual, by taking the inverse Laplace transform of $Y(s)$ where $Y(s) = H(s)X(s)$. For an impulse input $X(s) = 1$ and $Y(s) = H(s)$. Thus

$$y(t) = L^{-1}[Y(s)] = L^{-1}[H(s)] = h(t) \ . \tag{192}$$

The discrete impulse is represented by the sequence

$$\left\{\frac{1}{T}\ , 0,\ 0,\ \ldots.\right\} \tag{193}$$

thus

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) = \frac{1}{T}\,\tilde{H}(z) \tag{194}$$

as

$$\tilde{X}(z) = \sum_{n=0}^{\infty} x_n z^{-n} = \frac{1}{T} \ . \tag{195}$$

The discrete system output, y_n, must be identical to samples of the continuous system output, $y(nT) = h(nT)$ for an impulse invariant system given an impulse input. Therefore

$$\tilde{Y}(z) = \sum_{n=0}^{\infty} y_n z^{-n} = \sum_{n=0}^{\infty} h_n z^{-n} \tag{196}$$

and, since $\tilde{Y}(z) = \frac{1}{T} \tilde{H}(z)$

$$\tilde{H}(z) = T\tilde{Y}(z) = TZ\left[h_n\right] \tag{197}$$

$$= TZ[h(t)] \ . \tag{198}$$

With $\tilde{H}(z)$ defined as in (198) the output of the discrete system is, by definition, identical to that of the continuous system, given that the system is initially at rest and that the input is the impulse function.

Since the systems are linear and time invariant the simulation is invariant not only to a single impulse at $t = 0$ but also to any combination of weighted impulses at $t = nT$, $n = 0, 1, 2, \ldots$. For example, consider $x(t)$ as defined as

$$x(t) = \sum_{n=0}^{N} x_n \delta(t - nT) \tag{199}$$

then

$$X(s) = \sum_{n=0}^{N} x_n \ e^{-nsT} \tag{200}$$

as shown in (22). Now

$$Y(s) = H(s)X(s) = \sum_{n=0}^{N} x_n H(s) e^{-nsT} \tag{201}$$

$$= x_o H(s) + x_1\ e^{-sT}\ H(s) + \ldots + x_N\ e^{-NsT}\ H(s) \tag{202}$$

and

$$y(t) = x_o h(t) + x_1 h(t - t) + \ldots + x_N h(t - NT) \tag{203}$$

where $h(t - kT) = 0$ for $t < kT$.

From (195)

$$\tilde{X}(z) = \frac{1}{T} \sum_{n=0}^{N} x_n z^{-n} \tag{204}$$

and

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) = \frac{1}{T} \sum_{n=0}^{N} x_n \tilde{H}(z) z^{-n} \tag{205}$$

$$= \frac{1}{T}\left[x_o \tilde{H}(z) + x_1 \tilde{H}(z) z^{-1} + \ldots.\right] \tag{206}$$

The shifting theorem of Chapter 2 is given as

$$z^{-1}\left[z^{-k} F(z)\right] = f[(n - k)T] = f_{n-k} \tag{207}$$

where $f_{n-k} = 0$ for $n < k$.

Thus,

$$y_n = \frac{1}{T}\left[x_o Th_n + x_1 Th_{n-1} + \ldots + x_N Th_{n-N}\right] \tag{208}$$

$$= x_o h_n + x_1 h_{n-1} + \ldots + x_N h_{n-N} \tag{209}$$

and from (203)

$$y(nT) = x_o h_n + x_1 h_{n-1} + \ldots + x_N h_{n-N} \tag{210}$$

showing that $y_n = y(nT)$ for $x(t) = \sum_{n=0}^{N} x_n \delta(t - nT)$.

This impulse invariant $\tilde{H}(z)$ of (197) is also obtainable from the "zero order" approximation to the convolution integral as shown by Stearns.[30]

Consider the convolution equivalent of (190)

$$y(t) = \int_o^t x(u)h(t - u)du = \int_o^t h(u)x(t - u)du \tag{211}$$

the rectangular, or "zero order" approximation to this integral is given as

$$y_n = T\sum_{m=0}^{n} x_m h_{n-m} = T\sum_{m=0}^{n} h_m x_{n-m} \ . \tag{212}$$

Now if $x(t) = \delta(t)$

$$x_o = \frac{1}{T}\ , \quad x_n = 0 \qquad n \neq 0 \tag{213}$$

and, from (212)

$$y_n = T\left[\frac{1}{T} h_n\right] = h_n \tag{214}$$

also

$$\tilde{Y}(z) = \sum_{n=0}^{\infty} y_n z^{-n} = \sum_{n=0}^{\infty} h_n z^{-n} . \tag{215}$$

Since $\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) = \frac{1}{T}\tilde{H}(z)$ we have

$$\tilde{H}(z) = T\tilde{Y}(z) = T \sum_{n=0}^{\infty} h_n z^{-n} . \tag{216}$$

This impulse invariant or zero-order simulation is also referred to as the "standard z-transform" by Golden and Kaiser.[73]

The advantage of Stearns definition of the discrete impulse, $x_0 = \frac{1}{T}$, over that used by others, $x_0 = 1$, can be shown here. Assume $x_0 = 1$, $x_n = 0$ for $n \neq 0$, then (194) becomes

$$\tilde{Y}(z) = \tilde{H}(z) = \sum_{n=0}^{\infty} h_n z^{-n} \tag{217}$$

and (198) is changed to

$$\tilde{H}(z) = Z[h(t)] = \sum_{n=0}^{\infty} h_n z^{-n} . \tag{218}$$

However, (211) and (212) are unchanged and with $x_0 = 1$, (214) becomes

$$y_n = Th_n \tag{219}$$

requiring

$$\tilde{Y}(z) = T \sum_{n=0}^{\infty} h_n z^{-n} \ . \tag{220}$$

Comparing (215) with (220) illustrates the disadvantage of choosing $x_0 = 1$ for the discrete impulse. The transfer function obtained for the impulse invariant simulation has an overall gain factor of $\frac{1}{T}$ greater than the gain of the zero order approximation to the convolution integral. Thus, in general, for $T < 1$ the gain of the transfer function obtained by assuming the discrete impulse to be $x_0 = 1$, $x_n = 0$, $n = 1, 2, \ldots$ is too great.

The major disadvantage of the impulse invariant method is that, unless T is very small, considerable aliasing is present. From (216) and section 2.3, $\overline{H}(j\omega)$ can be written

$$\overline{H}(j\omega) = \sum_{n=-\infty}^{\infty} H\left[j\left(\omega - \frac{2n\pi}{T}\right)\right] \ . \tag{221}$$

For $|\omega| < \pi/T$ (221) gives, as shown in [73],

$$\overline{H}(j\omega) = \ldots + H[j(\omega + 2\pi/T)] + H(j\omega) + H[j(\omega - 2\pi/T)] + \ldots \tag{222}$$

In the limit, as T approaches 0, $\overline{H}(j\omega)$ approaches $H(j\omega)$ as $H(\pm j\infty) = 0$. However, for finite T aliasing is present.

This disadvantage is often offset by the simplicity of obtaining $\tilde{H}(z)$ from $H(s)$. In addition, $y_n = y(nT)$ for the impulse response. This implies the stability of $\tilde{H}(z)$ is as great as that of $H(s)$. To obtain $\tilde{H}(z)$ from $H(s)$ simply use (197) or (198). For most systems this is a matter of table lookup. These tables are available in most of the referenced texts as well as in mathematical handbooks [78].

A slight modification to the impulse invariant method can be made by adjusting the gain of $\tilde{H}(z)$ so that $\overline{H}(0) = H(0)$. The result is a much improved frequency domain simulation although the exact impulse response is lost. This will be referred to as the dc gain adjusted impulse invariant simulation.

4.2 The Step Function Invariant Simulation

Let x(t) be a step function

$$x(t) = au(t) \tag{223}$$

then

$$X(s) = a/s \tag{224}$$

and

$$Y(s) = H(s)X(s) = \frac{a}{s} H(s) \tag{225}$$

with

$$y(t) = L^{-1}\left[\frac{a}{s} H(s)\right] . \tag{226}$$

Correspondingly,

$$x_n = a,\ n = 0,\ 1,\ 2,\ \ldots \tag{227}$$

and

$$\tilde{X}(z) = a \sum_{n=0}^{\infty} z^{-n} = \frac{az}{z - 1} \tag{228}$$

with

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) = \frac{az}{z - 1} \tilde{H}(z) . \tag{229}$$

In order for the discrete system to produce output samples, y_n, identical to samples of $y(t) = y(nT)$, $\tilde{Y}(z)$ must be

$$\tilde{Y}(z) = Z[y(nT)] = Z\left\{L^{-1}\left[\frac{a}{s} H(s)\right]\right\} . \tag{230}$$

Now from (229),

$$\tilde{H}(z) = \frac{z - 1}{az} \tilde{Y}(z)$$

thus

$$\tilde{H}(z) = \frac{z - 1}{az} Z\left\{L^{-1}\left[\frac{a}{s} H(s)\right]\right\} . \tag{231}$$

With $\tilde{H}(z)$ as shown in (231) the response of the discrete system, $y_n = Z^{-1}[\tilde{Y}(z)]$, is identical to that of the continuous system at the sampling instants given that the input, $x(t)$, is a step function, $a_o u(t)$. Hence, the method is "step invariant." Just as with the impulse invariant system this system produces the correct output for any combination of step functions occurring at $t = nT$, $n = 0, 1, 2, \ldots.$ For example, if

$$x(t) = \sum_{n=0}^{N} a_n u(t - nT) \tag{232}$$

then

$$X(s) = \sum_{n=0}^{N} a_n \frac{e^{-nsT}}{s} \tag{233}$$

and

$$\tilde{X}(z) = \sum_{m=0}^{\infty} x_m z^{-m} = \frac{z}{z - 1} \sum_{n=0}^{N} a_n z^{-n} \tag{234}$$

with

$$Y(s) = \sum_{n=0}^{N} a_n \, e^{-nsT} \frac{H(s)}{s} \tag{235}$$

and

$$\tilde{Y}(z) = \sum_{n=0}^{N} a_n z^{-n} \frac{z}{z - 1} \tilde{H}(z) \tag{236}$$

or, from (231)

$$\tilde{Y}(z) = \sum_{n=0}^{N} a_n z^{-n} Z\left\{L^{-1}\left[\frac{H(s)}{s}\right]\right\} . \tag{237}$$

Now y(t), from (209), can be written as

$$y(t) = \sum_{n=0}^{N} a_n L^{-1}\left[e^{-nsT} \frac{H(s)}{s}\right] \tag{238}$$

or, from the shifting theorem

$$y(t) = \sum_{n=0}^{N} a_n g(t - nT) \tag{239}$$

where

$$G(s) = \frac{H(s)}{s} \text{ and } g(t) = L^{-1}[G(s)] . \tag{240}$$

Similarly, from (237) y_n can be written as

$$y_n = \sum_{n=0}^{N} a_n Z^{-1}\left[z^{-n} G(z)\right] \tag{241}$$

where

$$G(z) = Z[g(t)] . \tag{242}$$

Now using the z-transform shifting theorem

$$y_m = \sum_{n=0}^{N} a_n g[(m - n)T] . \tag{243}$$

Comparing (243) and (239), it is apparent that $y_n = y(nT)$, as required.

Referring back to the sampled data system of Chapter 2, Figure 1, we see that the step function invariant digital simulation is mathematically equivalent to an interpolator which "holds" the sample value x_n from time $t = nT$ until $t = nT + T$ at which time x_{n+1} is received and held, etc. Such an interpolator is referred to as a "zero-order hold" and has a transfer function, $Q(s)$, given by

$$Q(s) = \frac{1 - e^{-sT}}{s} \tag{244}$$

which is characterized in the time domain as

$$q(t) = u(t) - u(t - T) . \tag{245}$$

The overall transfer function of the sampled data system is $Q(s)H(s)$ and

$$Y^*(s) = Q(s)H(s)X^*(s) \tag{246}$$

where $X^*(s)$ is the Laplace Transform of the sequence of samples input to the interpolator. From (22) we have,

$$X^*(s) = \sum_{n=0}^{\infty} x_n e^{-nsT} . \tag{247}$$

Thus,

$$X^*(s) = \frac{e^{sT}}{e^{sT} - 1} \qquad \text{for } x_n = 1,\ n = 0, 1, 2, \ldots . \tag{248}$$

and for $x(t) = \sum_{n=0}^{N} a_n u(t - nt)$

$$X^*(s) = \sum_{n=0}^{N} a_n \frac{e^{-nsT}\ e^{sT}}{e^{sT} - 1} . \tag{249}$$

Combining (246) and (249) gives

$$Y^*(s) = \frac{1 - e^{sT}}{s} H(s) \sum_{n=0}^{N} a_n \frac{e^{-nsT}\ e^{sT}}{e^{sT} - 1} \tag{250}$$

or

$$Y^*(s) = \sum_{n=0}^{N} a_n\ e^{-nsT} \frac{H(s)}{s} . \tag{251}$$

Now (237) is the equivalent z-transform of the Laplace transforms of (251).

The step invariant transfer function of (231) is nearly as easy to obtain as is the impulse invariant transfer function, (197), and is probably a closer approximation to the actual transfer function for most input signals encountered in physical systems.

As was shown in Chapter 2, this method of obtaining $\tilde{H}(z)$ is the highest order "hold" that can be realized by a physical interpolator.

4.3 Higher Order Invariant Forms

The previous two sections have described the impulse and step function invariant simulation techniques. In this section the previous methods will be extrapolated to higher order polynomials.

The procedure for obtaining discrete transfer functions that will provide invariant simulation of H(s) for a given input form is described below. If x(t) is given as

$$x(t) = \sum_{n=0}^{N} a_n (t - nT)^k u(t - nT) \tag{252}$$

the general form of the substitution can be obtained by simply taking

$$x_o(t) = t^k u(t) \tag{253}$$

as the systems considered are linear. As before

$$Y(s) = H(s)X(s) = H(s) \frac{k!}{s^{k+1}} \tag{254}$$

since

$$L(t^k) = \frac{k!}{s^{k+1}} . \tag{255}$$

Similarly,

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z) = \tilde{H}(z)Z\left[t^k\right] \qquad (256)$$

where

$$Z\left[t^k\right] = (-T)^k z^k \frac{d^k}{dz^k}\left(\frac{z}{z-1}\right) \qquad (257)$$

$$= Z\left[L^{-1}\left(\frac{k!}{s^{k+1}}\right)\right] . \qquad (258)$$

For the digital system to be invariant to the input t^k we must have

$$\tilde{Y}(z) = Z(y_n) = Z\left\{L^{-1}[Y(s)]\right\} . \qquad (259)$$

Now from (254) and (256), (259) requires

$$\tilde{H}(z)Z\left[t^k\right] = Z\left\{L^{-1}\left[\frac{H(s)k!}{s^{k+1}}\right]\right\} \qquad (260)$$

or

$$\tilde{H}(z) = \frac{1}{Z\left[t^k\right]} Z\left\{L^{-1}\left[\frac{H(s)k!}{s^{k+1}}\right]\right\} . \qquad (261)$$

To be completely general (261) can be rewritten as

$$\tilde{H}(z) = \frac{1}{Z\left[x_o(t)\right]} Z\left\{L^{-1}\left(H(s)L\left[x_o(t)\right]\right)\right\} \qquad (262)$$

where $x_0(t)$ is simply the "form" of the input. Equation (262) is correct as long as $Z[x(t)]$ and $L[x(t)]$ exist and are rational functions of z and s, respectively.

If $\tilde{H}(z)$ is found from (262) and the conditions following (262) are met, then, for an input x(t) as assumed in (262), the output samples calculated from $\tilde{Y}(z)$ are

$$y_n = Z^{-1}[\tilde{Y}(z)] = Z^{-1}[\tilde{H}(z)\tilde{X}(z)] \ . \qquad (263)$$

These values of y_n are identical within the limits of roundoff error to the samples obtained from y(t) at t = nT as given below.

$$y_n = y(nT) = L^{-1}[Y(s)] = L^{-1}[H(s)X(s)] \ . \qquad (264)$$

Similar equations are developed by Jury[24] for interpolators to be used in sampled data systems for inputs of the form $x(t) = t^k$, k = 1, 2, 3, ... These are called normal interpolators of order k and, as previously mentioned, are not physically realizable for $k > 0$.

In Chapter 7 the impulse, dc gain adjusted impulse, step, and ramp invariant simulations of a first and second order continuous system are derived. The characteristics of these simulations are then compared with those obtained by other methods.

CHAPTER 5. DIGITAL SIMULATION THROUGH OPTIMIZATION PROCEDURES

The previously defined methods for obtaining the recursive digital system parameters from the continuous system share a common characteristic. Once the method of simulation is chosen for the given continuous system, all the discrete system parameters are specified. If the form of the input is known or if the input frequency is known and constant, then the invariant or bilinear methods respectively may be used to provide nearly perfect simulation. However, in many physical situations the input cannot be categorized in this fashion and the best simulation method may be impossible to obtain or even determine from the commonly used methods.

In this chapter some methods of determining "optimum simulations" based on various definitions of "optimum" are discussed.

5.1 Simulation System Errors

In order to determine an optimum simulation one must define what is to be considered a system error and then the search for the optimum method may be initiated. As previously stated, the ultimate goal of digital simulation is to produce a sequence of numbers, $\left\{y_n\right\}$, from an input sequence, $\left\{x_n\right\}$, that exactly match the sequence, $\{y(nT)\}$, obtained by sampling the output of the desired continuous system supplied with the continuous input $x(t)$. This match should occur for all $x(t)$

contained in the ensemble of possible input functions. Usually such an exact match is impossible and we are led to the problem of determining a valid error measure.

Error in the time domain, for a given input function, can simply be determined from the difference of corresponding values of the sequences $\left\{y_n\right\}$ and $\{y(nT)\}$ for some finite length of the sequences. This could be represented as

$$\text{Error} = \sum_{n=0}^{N} \left| y_n - y(nT) \right| \tag{265a}$$

with the input specified. Many modifications of (265a) are possible and potentially valid, two of which are: a weighting term, w_n, could be included in the summation:

$$\text{Error} = \sum_{n=0}^{N} w_n \left| y_n - y(nT) \right| \tag{265b}$$

or a squared error term $\left[y_n - y(nT)\right]^2$ could be used in place of the absolute value

$$\text{Error} = \sum_{n=0}^{N} \left[y_n - y(nT)\right]^2 . \tag{265c}$$

This type of error term is manageable, but it is only defined for one specific input function. To be generally useful (265a), (265b), or (265c) must be modified to include an

ensemble of input functions, $x_m(t)$, $m = 1, 2, \ldots M$. Now (265a) can be rewritten

$$\text{Error} = \sum_{m=1}^{M} \sum_{n=0}^{N} \left| y_{mn} - y_n(nT) \right| \tag{266}$$

with further modifications of a weighting function, W_{mn}, or squared error term of course still possible.

Now as M and/or N increase, (266) rapidly becomes analytically untractable. Thus, time domain errors, as defined here, do not appear to singularly fulfill the requirement of a valid error criterion.

Frequency domain errors are much easier to define and determine as we are limited to comparing only real and imaginary values of the desired and simulation transfer function for frequencies from 0 to $\frac{1}{2T}$. Further, the transfer functions for most systems are relatively smooth functions of frequency and only a few comparisons are needed over the frequency range to insure a reasonably valid measure of error.

This frequency domain error criterion is quite simple; however, some problems do arise. The first is that, by itself, this error measure presents no requirements for system stability. In fact, the most accurate simulations place poles in the z-plane outside the unit circle. Even if the poles are restricted to lie within the unit circle and the poles and zeroes adjusted to give the minimum frequency domain error the transient response of the discrete system may still be a

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]. [illegible]

[illegible]

[illegible]

Frequency domain [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

The frequency domain [illegible]

[illegible]

[illegible]

[illegible]. In fact, [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

grossly inaccurate representation of the similar response of the continuous system. This will be shown in Chapter 7. Another definition of frequency domain error is often chosen to be the difference in the amplitude responses of the transfer functions of the continuous and discrete systems as a function of frequency. This method not only possesses all the disadvantages listed above but also disregards phase information, which is often equally as important as the amplitude characteristic.

The preceding discussion is not intended as a direct criticism of any particular error definition used in conjunction with a specific problem. However, it does illustrate the problems involved in finding a suitable definition of simulation error.

The following section describes some previous methods used for obtaining "optimum" simulation coefficients.

5.2 Some Investigated Optimization Procedures

Fleischer[79] describes a method of obtaining coefficients for a nonrecursive digital simulation of a specified transfer function. The coefficients of the digital simulator are found not by a search procedure but rather from a set of simultaneous linear equations. His error measure is of interest here and is given by

$$\text{Error} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} W(j\omega)\,|F(j\omega) - \overline{F}(j\omega)|^2 d\omega \qquad (267)$$

where $W(j\omega)$ is a weighting function equal to the power spectral density of the expected input signal; $F(j\omega)$ and $\overline{F}(j\omega)$ are the desired continuous and discrete system transfer functions, respectively. No stability constraints were necessary here as no poles are present in the nonrecursive design. No consideration was given to response in the time domain.

Tufts, Rorabacher, and Mosier[58] also discuss an optimization procedure for nonrecursive digital filters. In general, if the desired frequency response is specified at all frequencies below the Nyquist frequency, then the best n-th order nonrecursive approximation to this desired response, in the least mean-squared error sense, is found by using the first n Fourier series coefficients of the desired frequency response. In this article a method is described of obtaining an optimum solution when only a finite number of points of the desired response are specified. This solution is obtained by solving a set of linear simultaneous equations. In addition, a brief summary is given of optimizing the approximation in a "minimax" sense. Of interest here are the two error criteria used; the first being the least mean-squared error,

$$E = \int_R [H(j\omega) - \overline{H}(j\omega)]^2 d\omega \qquad (268)$$

and the latter being the minimax error

$$E = \max\left[|H(j\omega) - \overline{H}(j\omega)|\right] , \quad \omega \varepsilon R \qquad (269)$$

where R is the set of values of ω where $H(j\omega)$ is specified, obviously $|\omega| < \pi/T$ for all $\omega \varepsilon R$.

Many of the other nonrecursive filter synthesis techniques discussed in Chapter 2 use optimization procedures to choose some or all of the filter coefficients. In all cases the error criterion used is either the least mean square of (268) or the minimax of (269).

The optimization techniques used are not of interest here as they differ from those to be discussed that are used with recursive designs. However, the error definitions are of interest as intuitively there is no reason to require different definitions of the approximation error based solely on the form of the realization of the digital filter.

The remainder of this chapter is a discussion of the optimization procedures used for the synthesis of recursive digital systems.

Steiglitz[80] applies an unconstrained optimization algorithm to the design of recursive digital filters. He defines the error measure to be the squared magnitude error of the simulation at a finite number of discrete frequencies, that is

$$E = \sum_{i=1}^{M} \left[|H(j\omega_i)| - |\overline{H}(j\omega_i)| \right]^2 , \quad 0 \leq \omega_1 < \omega_2 \ldots < \omega_m < \pi/T \tag{270}$$

where $H(j\omega_i)$ is the desired system characteristic at $\omega = \omega_i$ and $\overline{H}(j\omega_i)$ is the corresponding simulation characteristic. $\tilde{H}(z)$ is assumed to be a cascade of second order terms

$$\tilde{H}(z) = A \prod_{i=1}^{K} \frac{1 + a_k z^{-1} + b_k z^{-2}}{1 + c_k z^{-1} + d_k z^{-2}} . \qquad (271)$$

Thus, there are 4K + 1 unknowns, (a_1, b_1, c_1, d_1, ... a_K, b_K, c_K, d_K, A) to adjust for a minimum error as defined in (270). The optimization procedure used, to be discussed later, is based on a method originated by Fletcher and Powell[81] and is available as a FORTRAN IV program, FMFP (and DFMFP, double precision) in [82].

First the variable A is removed as an unknown parameter by analytically minimizing (270) with respect to A, leaving 4K unknowns. From starting values supplied for the unknowns the optimization algorithm searches for a minimum of (270). This minimum is not guaranteed to be a global minimum. However, by supplying various starting values the "lowest" local minimum may be found and used.

The next step is to find the poles and zeroes for the coefficients found from this minimum. For reasons of stability all poles must be inside or on the unit circle. The zeroes are likewise required to lie within or on the unit circle to produce minimum phase. However, this unconstrained minimization algorithm produces poles and zeroes outside the unit circle as well as inside. Fortunately, for z on the unit circle and α real

$$\left| \frac{z - \alpha}{z - 1/\alpha} \right| = \alpha \qquad (272)$$

with similar properties holding for complex pairs of roots as discussed in Chapter 2. Thus, all poles and zeroes lying outside the unit circle may be inverted; which, with an adjustment to the gain constant A has no effect on the error defined in (270). Following this inversion further optimization is attempted with the Fletcher-Powell algorithm. Following convergence of this second optimization, the final values of $(a_1, b_1, c_1, d_1 \ldots. a_K, b_K, c_K, d_K, A)$ are used for the digital filter coefficients.

No consideration is given in this method to phase error or the transient response of the digital filter. In practice a pole may fall on, or very nearly on, the unit circle, with the resulting transient response decaying very slowly.

Athanassopoulos and Waren,[83] describe an optimization method somewhat similar to the above, in that a cascade of second order systems is used to obtain the final digital system. In this method the poles are constrained to lie within the unit circle through the use of a penalty function. This penalty function, together with the error function, is then minimized. The error function may be derived from the amplitude, phase, or time domain response of the discrete system.

Helms[84] discusses optimization procedures in general and in [56] he defines an optimization procedure somewhat similar to that of Steiglitz in addition to providing an excellent list of references to related work.

The characteristic common to all the previously mentioned optimization procedures is that the error measure was

arbitarily selected and the coefficients of the digital system were chosen to minimize this defined error. The disadvantage of these same methods for general simulation is that, while minimizing the frequency domain error, there was no concern with the time domain errors. The next chapter describes an error criterion based on the frequency domain system characteristics but with constraints imposed to limit the time domain error.

CHAPTER 6. COMBINED FREQUENCY DOMAIN-TIME DOMAIN ERROR CRITERIA AND OPTIMIZATION PROCEDURE

In this chapter a simulation technique is described which will combine the advantages of the previously described digital simulation techniques while avoiding many of the disadvantages. These advantages and disadvantages are briefly reviewed below.

In the previous chapter we described various optimization procedures used to reduce the frequency domain error of digital filters. In some cases only magnitude response was considered while in other examples the complex frequency characteristics were used. However, none of these methods required more of the time domain response than that it be bounded (stable). Thus, the transient response of the digital systems obtained may be a poor approximation to the transient response of the continuous system simulated.

The method of Burrus and Parks,[67] briefly mentioned in Chapter 2, was based on the requirement of simulating a desired time-domain impulse response rather than a particular frequency response. However, no requirement of system stability was made; thus, this method is of little value in the general simulation problem.

Chapter 3 discussed the bilinear transformation, which, as will be shown later, is not particularly good at approximating either the frequency domain or time domain response. However, its extreme simplicity of realization and guaranteed

stability (in simulating a stable system) often outweigh the above disadvantages.

The invariant simulations discussed in Chapter 4 achieve perfect time domain response for a specific input signal. While the step and ramp invariant simulations have the correct dc gain, the impulse invariant simulation must be multiplied by a constant to achieve the correct dc gain. With this adjustment these three invariant simulation techniques provide good time domain response for a general class of inputs as well as a relatively good frequency domain response, particularly for the ramp invariant method. The disadvantage here is the difficulty encountered in obtaining the desired invariant simulation. Although the technique is straightforward, as shown in Chapter 4, it is tedious and error prone. A further difficulty is that the continuous system cannot be broken down into a cascade of first and/or second order terms, each of these simulated individually, and the total digital system obtained as a cascade of these first or second order simulations. If this is done, the resultant digital system will not be an invariant simulation of the total continuous system, as is shown below. Let

$$H_T(s) = H_1(s)H_2(s) \tag{273}$$

where

$$H_1(s) = H_2(s) = \frac{1}{s + 1} \tag{274}$$

thus,

$$H_T(s) = \left(\frac{1}{s + 1}\right)^2 . \tag{275}$$

Now let $\tilde{H}_T(z)$ be the impulse invariant approximation to $H_T(s)$ and $\tilde{H}_1(z) = \tilde{H}_2(z)$ be the impulse invariant approximations to $H_1(s)$ and $H_2(s)$, respectively. From (216)

$$\tilde{H}_1(z) = \tilde{H}_2(z) = TZ\left[L^{-1}\left(\frac{1}{s + 1}\right)\right] = \frac{Tz}{z - e^{-T}} \tag{276}$$

while

$$\tilde{H}_T(z) = TZ\left\{L^{-1}\left[\left(\frac{1}{s + 1}\right)^2\right]\right\} = \frac{T^2 z\, e^{-T}}{\left(z - e^{-T}\right)^2} \tag{277}$$

Thus, we see that

$$\tilde{H}_1(z)\tilde{H}_2(z) = \frac{T^2 z^2}{\left(z - e^{-T}\right)^2} \neq \tilde{H}_T(z) \tag{278}$$

and in general this is true for any $\tilde{H}_1(z)\tilde{H}_2(z)$. The reason is that, if $H_T(s)$ is broken into a cascade of simpler systems and we wish to find $\tilde{H}_T(z)$ to be an impulse invariant approximation to $H_T(s)$, only the first subsystem in the cascade has an impulse input. The second subsystem would have to be invariant to its input, which is the output of the first subsystem, etc., in order that the cascade of subsystems be invariant to an impulse input. Thus, we must synthesize the entire system, $\tilde{H}_T(z)$, from $H_T(s)$ directly without simplification. Another restriction

and potential disadvantage of the invariant simulations is that once a particular method is chosen there is no flexibility in choosing coefficients of the digital system; they are all fixed.

6.1 Limiting the Time Domain Error

In order to obtain good time domain response, as the invariant simulations do, we must examine these techniques in more detail. Consider the system

$$H(s) = \frac{1}{s + 1} \tag{279}$$

whose impulse response is

$$y(t) = e^{-t} . \tag{280}$$

From (216) the impulse invariant digital simulation of (279) is

$$\tilde{H}(z) = \frac{Tz}{z - e^{-T}} = \frac{T}{1 - e^{-T}z^{-1}} . \tag{281}$$

Requiring $\overline{H}(0) = H(0)$ for the dc gain adjusted impulse invariant simulation gives

$$\tilde{H}(z) = \frac{1 - e^{-T}}{1 - e^{-T}z^{-1}} . \tag{282}$$

Similarly, the response of (279) to a unit step input is

$$y(t) = 1 - e^{-t} \tag{283}$$

and the step invariant simulation is

$$\tilde{H}(z) = \frac{(1 - e^{-T})z^{-1}}{1 - e^{-T}z^{-1}} \quad . \tag{284}$$

The ramp input response of (279) is

$$y(t) = t + e^{-t} - 1 \tag{285}$$

while the ramp invariant simulation is

$$\tilde{H}(z) = \frac{1}{T} \frac{(T + e^{-T} - 1) + (1 - T e^{-T} - e^{-T})z^{-1}}{1 - e^{-T}z^{-1}} \quad . \tag{286}$$

These simulations and others are derived in greater detail in Chapter 7. Here we only wish to show the form of these invariant simulations. First note that regardless of the form of the input; impulse, step, or ramp we always obtain an output term of the form e^{-t}. This is simply the transient response of the system of (279) and will be present for any input, as is well known in electrical network theory.[85] (If a pole of H(s) lies on the $j\omega$ axis then this characteristic response is often labeled a free response rather than a transient response.) However, in this paper we will use the term transient response and assume all poles of H(s) to lie in the open left-half plane. Now just as $\frac{1}{s + 1}$ is the Laplace transform of e^{-t}, $\frac{Tz}{z - e^{-T}}$ is the z-transform of e^{-nT}. Thus, it is necessary for the three invariant simulations above to have poles at $z = e^{-T}$. Note that even though the denominators of all three invariant simulations are identical, this does not imply that, for example, the step invariant simulation will be exact for a ramp input.

However, it does imply that the transient response of the step invariant simulation will decay at the same rate as the continuous system for a ramp, or any other input. Thus, we can limit the time domain error by restraining the poles of the digital simulation to be found from the following relationship

$$\text{z-plane poles} = \exp[T(\text{s-plane poles})] \ . \tag{287}$$

Obviously, we have not minimized this error; we have only limited it so that the transient response of the digital system decays as fast as that of the continuous system. Thus, a less restrictive form of (287) is simply to limit the magnitude of the z-plane poles.

$$|z| \leq \left|e^{sT}\right| = e^{T[\mathrm{Real}(s)]} \tag{288}$$

where z is a z-plane pole and s is an s-plane pole. This will force the transient of the digital system to decay as fast or faster than that of the continuous system with no other restrictions.

Following a discussion of the form of the digital system and the optimization algorithm, we will return to this restriction on the z-plane poles.

6.2 Defining the Digital System and the Frequency Domain Error

In order to define an effective optimization technique for a digital simulation system, we must first specify the error measure to be used, the constraints imposed on the variables, the form of the digital system, and the optimization algorithm.

The definition of error chosen here is

$$E = \sum_{m=1}^{M} \left| H(j\omega_m) - \overline{H}(j\omega_m) \right|^2 \tag{289}$$

where $0 < \omega_1 < \omega_2 < \ldots . < \omega_M < \pi/T$. Both the amplitude and phase errors are included in a least mean-squared sense. The frequencies of comparison, $f_m = \omega_m/2\pi$, may be chosen in any desired manner. In addition, this form of error measure and its derivatives with respect to the digital coefficients are directly obtainable on a digital computer. The other error criteria discussed previously either do not include both magnitude and phase or are more difficult to implement on a digital computer.

The form of the digital system is given below

$$\tilde{H}(z) = A \frac{\prod_{k=1}^{K} \left(1 + a_{2k-1} z^{-1} + a_{2k} z^{-2}\right)\left(1 + a'_{NZ} z^{-1}\right)}{\prod_{k=1}^{NCP/2} \left(1 + 2b_{2k} \cos b_{2k-1} z^{-1} + b_{2k}^2 z^{-2}\right) \prod_{k=NCP+1}^{NP} \left(1 + b_k z^{-1}\right)} \tag{290}$$

where

NZ ≡ number of zeroes

NP ≡ number of poles

NCP ≡ number of complex poles

K ≡ largest integer $\leq$ NZ/2

$a'_{NZ} = 0$ if NZ = 2K (even number of zeroes)

$a'_{NZ} \neq 0$ if NZ = 2K + 1 (odd number of zeroes).

This model is somewhat similar to that proposed by Steiglitz in (271) with the following exceptions:

1. We do not require the order of the numerator to be the same as that of the denominator.
2. The number of zeroes may be even or odd. ($a'_{NZ} = 0$ for an even number of zeroes).
3. The zeroes are not constrained to lie within the unit circle, although this would be a simple modification if desired.
4. The complex poles are separated from the real poles to allow the pole magnitudes to be constrained. The complex pole pairs are represented by

   $$-b_{2k}\, e^{\pm j b_{2k-1}} \quad \text{in (290)} .$$

The basic concept of a cascade of second (and/or first) order systems is retained and any desired rational function in z^{-1} may be constructed in the form of (290). Also the simulator may be realized directly in the form least sensitive to quantization and roundoff errors, as discussed in Chapter 2, from the coefficients obtained.

The need to restrict the pole magnitude has forced the separation of the real and complex poles as there is no general method known (to the author) of restricting the magnitude of the roots of a second order polynomial simply by restricting the coefficients. Once this separation is incorporated into the optimization algorithm, it produces few problems as it is

usually very simple to determine the optimum number of complex poles for any given simulation.

The constraints imposed on the system coefficients are simple; the poles, both real and complex, are required to lie within a circle of predetermined radius in the z-plane as in (288). As discussed previously, a natural choice for this maximum radius is exp(T[Real(s-plane pole)]); however, it is not necessary to make this choice.

6.3 Optimization Procedure

The optimization technique to be used is a more difficult choice. The function to be minimized is, on substituting (290) into (289) and replacing z by $e^{j\omega T}$

$$E = \sum_{m=1}^{M} \left\{ \left| H(j\omega_m) - A \frac{\prod_{k=1}^{K} \left(1 + a_{2k-1} e^{-j\omega_m T} + a_{2k} e^{-j2\omega_m T}\right)\left(1 + a'_{NZ} e^{-j\omega_m T}\right)}{\prod_{k=1}^{NCP/2} \left(1 + 2b_{2k} \cos b_{2k-1} e^{-j\omega_m T} + b_{2k}^2 \, e^{-2j\omega_m T}\right) \prod_{k=NCP+1}^{NP} \left(1 + b_k e^{-j\omega_m T}\right)} \right|^2 \right\} \tag{291}$$

Both Steiglitz[80] and Helms[84] have noted that this error function is not convex with respect to the variables A; a_i, i = 1, 2, ... NZ; b_i, i - 1, 2, ... NP. Thus, optimization methods which depend upon the gradient of the function may find a

local minimum and not a global minimum. Direct search procedures may also be used; a good, brief review of these techniques is presented by Lawrence and Steiglitz[86] together with a proposal for a randomized pattern search. In addition, the paper contains references to most of the recent literature concerned with optimization. However, due to the availability of the Fletcher-Powell algorithm, it was used for this paper. Here we will only briefly review the technique as it is implemented in the FORTRAN subroutines FMFP (single precision) or DFMFP (double precision) in the IBM manual [82]. The method is described in detail in the original paper [81]. The problem of finding a global minimum instead of a local minimum is at least circumvented by supplying various starting values to the optimization algorithm. The local minimums can then be compared to choose the digital system with the least error. All results in this paper were obtained with the single precision version as no convergence problems arose and the amount of computational time is significantly less than that required by the double precision version.

The generalized Taylor expansion for a function of several variables is

$$f(x + u) = f(x) + g(x)u + \frac{1}{2} u^T G(x)u + \dots . \qquad (292)$$

where x is an n-dimensional argument vector, u is an n-dimensional incremental vector, g(x) is the gradient vector, u^T is the transpose of u and G(x) is the n x n matrix of second

order partial derivatives. Now in the neighborhood of the minimum of f we assume that f is approximated closely by the three terms on the right side of (292). In addition g(x) must be 0 at the minimum giving

$$f(x) = f(x_{min}) + \frac{1}{2}(x - x_{min})^T G(x_{min})(x - x_{min}) . \qquad (293)$$

The gradient at x is approximately given by

$$g(x) = G(x_{min})(x - x_{min}) . \qquad (294)$$

Now assume that the symmetric matrix G is positive definite.[87] From (294) we have

$$(x - x_{min}) = G^{-1}(x_{min})g(x) . \qquad (295)$$

If $G^{-1}(x_{min})$ were known, we could calculate x_{min} in one step. However, in general it is not known and here a method of successive linear searches in G-conjugate directions is used. Initially $G^{(o)}$ is chosen to be the identity matrix I; from this beginning a sequence of symmetric matrices $G^{(i)}$ is generated which tends toward G^{-1}. At the (i + 1)st step a linear search is made in the direction

$$h_i = -G^{(i)}g^{(i)} \qquad (296)$$

where $g^{(i)}$ is the gradient vector at $x^{(i)} = g(x^i)$.

This search is terminated when:

1. The function value, $f(x^i)$ has not decreased in the last step and the gradient vector is

sufficiently small or the argument and direction vectors have changed very little, provided at least n (the number of variables) iterations have been performed. Either of these conditions indicate the search has been successful and at least a local minimum has been found.

2. A predetermined number of iterations have been performed, or
3. A linear search indicated no minimum exists.

By itself this algorithm imposes no constraints on the argument so artificial limits must be placed on the pole locations. Also, from (296), we see that we must have the gradient vector $g(x^i)$ at each interation.

Before beginning a detailed discussion of the method used, an assumption made for the purposes of this paper should be stated: the d.c. gain of the digital simulation is required to be exact, i.e., $\overline{H}(0) = H(0)$. From (290) we can now define A in terms of a_i and b_i

$$\overline{H}(0) = \tilde{H}(1) = H(0) \tag{297}$$

thus

$$A = H(0) \frac{\prod_{k=1}^{NCP/2} \left(1 + 2b_{2k} \cos b_{2k-1} + b_{2k}^2\right) \prod_{k=NCP+1}^{NP} (1 + b_k)}{\prod_{k=1}^{K} (1 + a_{2k-1} + a_{2k})(1 + a'_{NZ})} \tag{298}$$

With A eliminated in (290) we now have NZ numerator coefficients and NP denominator coefficients with which to minimize the error, E, of (291). In order to constrain the poles of (290) to lie within a radius RAD, of the origin, several restrictions were tried. The most successful, and one of the simplest, was simply to set

$$b_\ell = \text{RAD} \sin (\arg_{\ell+\text{NZ}}) \tag{299}$$

in (290) for the magnitude of the complex poles, $\ell = 2r$, $r = 1, 2, \ldots \text{NCP}/2$; and, for the real poles, $\ell = \text{NCP}+1, \ldots \text{NP}$. Here b_ℓ is the pole magnitude and $\arg_{\ell+\text{NZ}}$ is the unconstrained variable whose value is optimized to minimize the error of (291). In the remainder of this chapter the equations needed for this optimization procedure are developed.

The preceding discussion of the Fletcher-Powell optimization technique indicates that a method is needed to obtain not only the value of the function to be optimized, but also its gradient vector at each step of the iterative procedure. The equations needed are developed here; the FORTRAN IV listing of the main program and two required subroutines is included as Appendix A. The equations require only real variables as the programming was done to allow use on smaller computers whose FORTRAN capabilities do not include complex arithmetic. The mathematics is simple, but the algebra is tedious. We repeat here equation (289)

$$E = \sum_{m=1}^{M} \left|H(j\omega_m) - \overline{H}(j\omega_m)\right|^2, \quad 0 < \omega_1 < \omega_2 < \ldots < \omega_M < \pi/T \qquad (289)$$

which can be written as

$$E = \sum_{m=1}^{M} \left[(HR_m - HBR_m)^2 + (HI_m - HBI_m)^2\right] \qquad (300)$$

where

$$HR_m = \text{Real}\left(H(j\omega_m)\right)$$
$$HI_m = \text{Imag}\left(H(j\omega_m)\right)$$
$$HBR_m = \text{Real}\left(\overline{H}(j\omega_m)\right)$$
$$HBI_m = \text{Imag}\left(\overline{H}(j\omega_m)\right) \ .$$

Now the value of E, the function to be minimized is directly obtainable from (300). As was shown in (298), the constant A in (290) is directly obtainable from the d.c. gain of the simulated function H(0) and the poles and zeroes of the digital system. Thus, E is a function only of the poles and zeroes and consequently the gradient vector required is

$$g = \left(\frac{\partial E}{\partial a_1}, \frac{\partial E}{\partial a_2}, \ldots . \frac{\partial E}{\partial a_{NZ}}, \frac{\partial E}{\partial b_1}, \frac{\partial E}{\partial b_2}, \ldots . \frac{\partial E}{\partial b_{NP}}\right) . \qquad (301)$$

From (300) we can write $\frac{\partial E}{\partial a_i}$ as

$$\frac{\partial E}{\partial a_i} = -2 \sum_{m=1}^{M} \left[(HR_m - HBR_m)\frac{\partial HBR_m}{\partial a_i} + (HI_m - HBI_m)\frac{\partial HBI_m}{\partial a_i}\right] \qquad (302)$$

with $\frac{\partial E}{\partial b_i}$ similar.

Now we must obtain $\frac{\partial HBR_m}{\partial a_i}$ and $\frac{\partial HBI_m}{\partial a_i}$ and the corresponding terms for the denominator coefficients. Substituting $e^{j\omega T}$ for z in (290) gives

$$\tilde{H}(e^{j\omega T}) = \overline{H}(j\omega) =$$

$$A \frac{\prod_{k=1}^{K} \left(1 + a_{2k-1}e^{-j\omega T} + a_{2k}e^{-j2\omega T}\right)\left(1 + a'_{NZ}e^{-j\omega T}\right)}{\prod_{k=1}^{NCP/2}\left(1 + 2b_{2k}\cos b_{2k-1}e^{-j\omega T} + b_{2k}^2 e^{-j2\omega T}\right)\prod_{k=NCP+1}^{NP}\left(1 + b_k e^{-j\omega T}\right)} \tag{303}$$

Recalling the definition of A in (298), and writing $\overline{H}(j\omega_\ell)$ as $HBR_\ell + jHBI_\ell$, $\ell = 1, 2, \ldots M$ gives

$$HBR_\ell + jHBI_\ell = H(0)$$

$$\prod_{k=1}^{K} \frac{\left(1 + a_{2k-1}e^{-j\omega_\ell T} + a_{2k}e^{-j2\omega_\ell T}\right)}{(1 + a_{2k-1} + a_{2k})} \frac{\left(1 + a'_{NZ}e^{-j\omega_\ell T}\right)}{(1 + a'_{NZ})}$$

$$\left[\prod_{k=1}^{NCP/2} \frac{\left(1 + b_{2k}\cos b_{2k-1} + b_{2k}^2\right)}{\left(1 + b_{2k}\cos b_{2k-1}e^{-j\omega_\ell T} + b_{2k}^2 e^{-j2\omega_\ell T}\right)}\right.$$

$$\left.\prod_{k=NCP+1}^{NP} \frac{(1 + b_k)}{\left(1 + b_k e^{-j\omega_\ell T}\right)}\right] \tag{304}$$

The following algebraic manipulation isolates a given variable and makes all the derivatives obtainable in the same manner. Assume we have a function $F(x,y,z) = e(x)g(y)h(z)$, which can be written as

$$F(x,y,z) = \frac{e(x)g(y)h(z)}{h(z)}\, h(z) \tag{305}$$

or

$$F(x,y,z) = F_z(x,y)h(z) \tag{306}$$

where

$$F_x(x,y) = \frac{e(x)g(y)h(z)}{h(z)} = e(x)g(y) \ . \tag{307}$$

Now

$$\frac{\partial F(x,y,z)}{\partial z} = F_z(x,y)\ \frac{\partial h(z)}{\partial z} \tag{308}$$

or

$$\frac{\partial F(x,y,z)}{\partial z} = \frac{F(x,y,z)}{h(z)}\ \frac{\partial h(z)}{\partial z} \ . \tag{309}$$

Using this manipulation on (304) we have, for i even

$$\frac{\partial}{\partial a_i}\left[HBR_\ell + jHBI_\ell\right] = \Bigg[(HBR_\ell + jHBI_\ell)$$

$$\left.\left(\frac{1 + a_{i-1} + a_i}{1 + a_{i-1}e^{-j\omega_\ell T} + a_i e^{-2j\omega_\ell T}}\right)\right]\ \frac{\partial}{\partial a_i}\left[\frac{1 + a_{i-1}e^{-j\omega_\ell T} + a_i e^{-j2\omega_\ell T}}{1 + a_{i-1} + a_i}\right] \tag{310}$$

for i odd, $i \neq NZ$

$$\frac{\partial}{\partial a_i}\left[HBR_\ell + jHBI_\ell\right] = \Bigg[(HBR_\ell + jHBI_\ell)$$

$$\left.\left(\frac{1 + a_i + a_{i+1}}{1 + a_i e^{-j\omega_\ell T} + a_{i+1} e^{-j2\omega_\ell T}}\right)\right] \frac{\partial}{\partial a_i}\left[\frac{1 + a_i e^{-j\omega_\ell T} + a_{i+1} e^{-j2\omega_\ell T}}{1 + a_i + a_{i+1}}\right] \tag{311}$$

and finally, for i odd, $i = NZ$

$$\frac{\partial}{\partial a_{NZ}}\left[HBR_\ell + jHBI_\ell\right] = \Bigg[\left(HBR_\ell + jHBI_\ell\right)$$

$$\left.\left(\frac{1 + a_{NZ}}{1 + a'_{NZ} e^{-j\omega_\ell T}}\right)\right] \frac{\partial}{\partial a_{NZ}}\left[\frac{1 + a_{NZ} e^{-j\omega_\ell T}}{1 + a_{NZ}}\right] \tag{312}$$

The expressions for the partial derivatives with respect to the denominator terms are similar.

For i even, $i \leq NCP$

$$\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right]$$

$$= \left[\left(HBR_\ell + jHBI_\ell\right)\left(\frac{1 + 2b_i \cos b_{i-1} e^{-j\omega_\ell T} + b_i^2 e^{-j2\omega_\ell T}}{1 + 2b_i \cos b_{i-1} + b_i^2}\right)\right]$$

$$\frac{\partial}{\partial b_i}\left[\frac{1 + 2b_i \cos b_{i-1} + b_i^2}{1 + 2b_i \cos b_{i-1} e^{-j\omega_\ell T} + b_i^2 e^{-j2\omega_\ell T}}\right] \tag{313}$$

For i odd, i < NCP

$$\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right]$$

$$= \left[\left(HBR_\ell + jHBI_\ell\right)\left(\frac{1 + 2b_{i+1}\cos b_i e^{-j\omega_\ell T} + b_{i+1}^2 e^{-j2\omega_\ell T}}{1 + 2b_{i+1}\cos b_i + b_{i+1}^2}\right)\right]$$

$$\frac{\partial}{\partial b_i}\left[\frac{1 + 2b_{i+1}\cos b_i + b_{i+1}^2}{1 + 2b_{i+1}\cos b_i\, e^{-j\omega_\ell T} + b_{i+1}^2 e^{-j2\omega_\ell T}}\right] \qquad (314)$$

For i > NCP

$$\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right]$$

$$= \left[\left(HBR_\ell + jHBI_\ell\right)\left(\frac{1 + b_i e^{-j\omega_\ell T}}{1 + b_i}\right)\right]\frac{\partial}{\partial b_i}\left[\frac{1 + b_i}{1 + b_i e^{-j\omega_\ell T}}\right]. \qquad (315)$$

In equations (310) through (315) the terms before the derivative on the right side of the equal sign can simply be treated as a complex constant. The expression following the derivative on the right is broken down into its real and imaginary components before its partial derivative is found. For example, consider

$$\frac{\partial}{\partial a_i}\left[\frac{1 + a_{i-1}e^{-j\omega_\ell T} + a_i\, e^{-j2\omega_\ell T}}{1 + a_{i-1} + a_i}\right] \qquad (316)$$

from (310). This can be expanded as

$$\frac{\partial}{\partial a_i}\left[\frac{\left(1 + a_{i-1}\cos\,\omega_\ell T + a_i\,\cos\,2\omega_\ell T\right) - j\left(a_{i-1}\sin\,\omega_\ell T + a_i\,\sin\,2\omega_\ell T\right)}{1 + a_{i-1} + a_i}\right] \tag{317}$$

Taking the indicated partial derivative gives

$$\left[\frac{1}{1 + a_{i-1} + a_i}\right]^2$$

$$\left\{\left[\cos\,2\omega_\ell T(1 + a_{i-1} + a_i) - (1 + a_{i-1}\,\cos\,\omega_\ell T + a_i\,\cos\,2\omega_\ell T)\right]\right.$$
$$\left.-j\left[\sin\,2\omega_\ell T(1 + a_{i-1} + a_i) - (a_{i-1}\,\sin\,\omega_\ell T + a_i\,\sin\,2\omega_\ell T)\right]\right\} \tag{318}$$

This can now be simplified to the expression

$$\left[\frac{1}{1 + a_{i-1} + a_i}\right]^2\left\{\left[(1 + a_{i-1})\cos\,2\omega_\ell T - 1 - a_{i-1}\,\cos\,\omega_\ell T\right]\right.$$
$$\left.+ j\left[a_{i-1}\sin\,\omega_\ell T - (1 + a_{i-1})\sin\,2\omega_\ell T\right]\right\}\,. \tag{319}$$

Thus, (310) can be rewritten as

$$\frac{\partial}{\partial a_i}\left[HBR_\ell + jHBI_\ell\right] = \left[\frac{HBR_\ell + jHBI_\ell}{1 + a_{i-1} + a_i}\right]$$

$$\left\{\frac{\left[(1 + a_{i-1})C2_\ell - 1 - a_{i-1}C_\ell\right] + j\left[a_{i-1}S_\ell - (1 + a_{i-1})S2_\ell\right]}{(1 + a_{i-1}C_\ell + a_iC_{2\ell}) - j(a_{i-1}S_\ell + a_iS2_\ell)}\right\} \tag{320}$$

where $C_\ell = \cos \omega_\ell T$, $C2_\ell = \cos 2\omega_\ell T$, $S_\ell = \sin \omega_\ell T$, and $S2_\ell = \sin 2\omega_\ell T$.

Similarly, we can rewrite (311), for i odd, as

$$\frac{\partial}{\partial a_i}\Big[HBR_\ell + jHBI_\ell\Big]$$

$$= \Big[HBR_\ell + jHBI_\ell\Big]\Big[\Big(C_\ell(1 + a_{i+1}) - 1 - a_{i+1}C2_\ell\Big)$$

$$+ j\Big(a_{i+1}S2_\ell - (1 + a_{i+1})S_\ell\Big)\Big] \Big/ \Big[1 + a_i + a_{i+1}\Big]\Big[(1 + a_iC_\ell$$

$$+ a_{i+1}C2_\ell) - j(a_{i-1}S_\ell + a_iS2_\ell)\Big] \; . \qquad (321)$$

and (312), for NZ odd, becomes

$$\frac{\partial}{\partial a'_{NZ}}\Big[HBR_\ell + jHBI_\ell\Big] = \frac{\Big[HBR_\ell + jHBI_\ell\Big]\Big[(C_\ell - 1) - j(S_\ell)\Big]}{(1 + a'_{NZ})\Big[(1 + a'_{NZ}C_\ell) - ja'_{NZ}S_\ell\Big]} \; . \qquad (322)$$

The derivatives with respect to the denominator coefficients are expanded in the same manner, first (313), for i even

$$
\begin{aligned}
&\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right] = \frac{2\left[HBR_\ell + jHBI_\ell\right]}{\left[1 + 2b_i \cos b_{i-1} + b_i^2\right]} \\
&\Big\{\Big[(\cos b_{i-1} + b_i)\left(1 + 2b_i C_\ell \cos b_{i-1} + b_i^2 C2_\ell\right) \\
&\quad - \left(1 + 2b_i \cos b_{i-1} + b_i^2\right)(\cos b_{i-1} C_\ell + b_i C2_\ell)\Big] \\
&\quad + j\Big[\left(1 + 2b_i \cos b_{i-1} + b_i^2\right)(\cos b_{i-1} S_\ell + b_i S2_\ell) \\
&\quad - (\cos b_{i-1} + b_i)\left(2b_i \cos b_{i-1} S_\ell + b_i^2 S2_\ell\right)\Big]\Big\}\Big/ \\
&\Big\{\left[1 + 2b_i \cos b_{i-1} C_\ell + b_i^2 C2_\ell\right] - j\left[2b_i \cos b_{i-1} S_\ell + b_i^2 S2_\ell\right]\Big\} \; .
\end{aligned}
\tag{323}
$$

Similarly, for i odd, (314) becomes

$$
\begin{aligned}
&\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right] = \frac{2\left[HBR_\ell + jHBI_\ell\right]}{\left(1 + 2b_{i+1} \cos b_i + b_{i+1}^2\right)} \\
&\Big\{\Big[\left(1 + 2b_{i+1} \cos b_i + b_{i+1}^2\right) b_{i+1} C_\ell \sin b_i \\
&\quad - b_{i+1} \sin b_i \left(1 + 2b_{i+1} \cos b_i C_\ell + b_{i+1}^2 C2_\ell\right)\Big] \\
&\quad + j\Big[b_{i+1} \sin b_i \left(2b_{i+1} \cos b_{i+1} S_\ell + b_{i+1}^2 S2_\ell\right) \\
&\quad - \left(1 + 2b_{i+1} \cos b_i + b_{i+1}^2\right) b_{i+1} \sin b_i S_\ell\Big]\Big\}\Big/ \\
&\Big\{\left[1 + 2b_{i+1} \cos b_i C_\ell + b_{i+1}^2 C2_\ell\right] - j\left[2b_{i+1} \cos b_i S_\ell + b_i^2 S2_\ell\right]\Big\} \; .
\end{aligned}
\tag{324}
$$

Finally, for i = NCP + 1, NCP + 2, ... NP; (315) can be written as

$$\frac{\partial}{\partial b_i}\left[HBR_\ell + jHBI_\ell\right] = \frac{\left[HBR_\ell + jHBI_\ell\right]\left[(1 - C_\ell) + jS_\ell\right]}{(1 + b_i)\left[(1 + b_i C_\ell) - jb_i S_\ell\right]} \ . \qquad (325)$$

Now we have the partial derivatives of a complex function given as a complex expression in (320) through (325). However, since HBR_ℓ and HBI_ℓ are both real as are all the coefficients, the separation of the partial derivative of $\overline{H}(j\omega)$ with respect to the coefficients a_i or b_i is simple. That is

$$\frac{\partial}{\partial a_i} HBR_\ell = \text{Real}\left[\frac{\partial}{\partial a_i}(HBR_\ell + jHBI_\ell)\right] \qquad (326)$$

$$\frac{\partial}{\partial a_i} HBI_\ell = \text{Imag}\left[\frac{\partial}{\partial a_i} HBR_\ell + jHBI_\ell\right] \ . \qquad (327)$$

Similarly, the partial derivatives with respect to the denominator coefficients may be found.

Now with (326) and (327) we have an explicit expression for the gradient vector of (291). One further step needs to be taken. In the optimization method used we must minimize E by finding the optimum location of NT unconstrained arguments, NT = NP + NZ, where NP and NZ are the number of poles and zeroes, respectively, in the digital simulation. We represent these arguments by the variables arg_n; n = 1, 2, ... NT. Now we must define the relationship between these arguments and the numerator and denominator coefficients a_i and b_i previously

discussed. Since the numerator coefficients are unconstrained, we let

$$a_i = arg_i \ , \ i = 1, 2, \ldots. \ NZ \tag{328}$$

and

$$\frac{\partial \overline{H}(j\omega)}{\partial a_i} = \frac{\partial \overline{H}(j\omega)}{\partial arg_i} \ , \ i = 1, 2, \ldots. \ NZ \ . \tag{329}$$

Thus, we can simply equate a_i and arg_i, $i = 1, 2, \ldots. \ NZ$ in all cases.

Some of the denominator coefficients are constrained and for these we must show the relationship between the arguments arg_i; $i = NZ + 1, NZ + 2, \ldots. \ NT$, and the denominator coefficients b_i; $i = 1, 2, \ldots. \ NP$. As previously discussed we will let RAD represent the maximum distance of any pole from the origin. Thus, for complex poles where b_i, i odd, represents the radius of the pole we set

$$b_i = RAD \ \sin(arg_{i+NZ}) \ , \ i = 1, 3, \ldots. \ NCP\text{-}1 \ . \tag{330}$$

Similarly for all real poles

$$b_i = RAD \ \sin(arg_{i+NZ}) \ , \ i = NCP+1, NCP+2, \ldots. \ NP \ . \tag{331}$$

Correspondingly, the gradient vector with respect to the arguments is obtained from the gradient vector with respect to the denominator coefficients as follows

$$\frac{\partial \overline{H}(j\omega)}{\partial \arg_{i+NZ}} = \frac{\partial \overline{H}(j\omega)}{\partial b_i} \text{RAD} \cos(\arg_{i+NZ}) \qquad (332)$$

for i as in (330) and (331).

The remaining b_i, i = 2, 4, ... NCP are unconstrained and can be equated directly to the corresponding arguments

$$b_i = \arg_{i+NZ} \text{ , } i = 2, 4, 6, \text{ .... NCP} \qquad (333)$$

and

$$\frac{\partial \overline{H}(j\omega)}{\partial \arg_{i+NZ}} = \frac{\partial \overline{H}(j\omega)}{\partial b_i} \text{ , } i = 2, 4, 6, \text{ .... NCP .} \qquad (334)$$

We have now defined explicitly the equations needed to obtain the value of the error and the gradient of this error function with respect to its unconstrained arguments at each step of the iteration.

The following is a summary of the method used to obtain the optimized coefficients of the digital simulation systems compared in the next chapter.

1. The form of the desired continuous system to be simulated is defined as $\overline{H}(j\omega) = HR(j\omega) + jHI(j\omega)$ for $0 \leq \omega \leq \pi/T$.
2. The sampling interval, T, is specified.
3. The frequencies at which the error measure is evaluated are specified. That is, for

$$E = \sum_{m=1}^{M} \left| H(j\omega_m) - \overline{H}(j\omega_m) \right|^2 \qquad (335)$$

the ω_m must be specified.

4. The maximum radius of the poles, RAD, and the expected minimum error value must be specified.
5. The number of zeroes, poles, and complex poles of the digital system must be specified.
6. Initial guesses for the numerator and denominator coefficients must be supplied.

Three FORTRAN programs were written to accomplish this optimization. The first OPT, the main program, accepts the digital system parameters and the initial guesses for the coefficients. It then converts these coefficients to the corresponding arguments for the optimization routine and calls FMFP. This call to FMFP supplies the name of a subroutine to calculate the error value and gradient vector, the number of arguments, the argument vector, a variable to contain the minimum error value, an array for the gradient vector to be stored and transferred in, an expected value for the minimum error, an absolute error term which is machine dependent, a maximum number of allowed iterations, a variable to return a flag indicating the success or failure of the optimization, and a blank "working space" array.

Following the return from FMFP to OPT; the minimum error value, the flag and the number of calls to the evaluation subroutine are printed. In addition, the numerator and denominator coefficients are printed as are the coefficients of the

gradient vector. Then, if desired, the real and imaginary values of the desired function and the simulation are printed out for $\omega = 0, \pi/12T, 2\pi/12T, 3\pi/12T, \ldots . 11\pi/12T, \pi/T$.

The program OPT then either terminates or reads in new digital system specifications and initial guesses for another simulation of the same continuous system.

The second program written is FUN2, a subroutine called for in each iteration of FMFP to provide the value of the error function and its gradients for the present arguments. The actual calculations are not done in FUN2, only the bookkeeping is performed. That is, the arguments used in FMFP are converted to the numerator and denominator coefficients used elsewhere, the values of $\omega = \omega_m$ for which the error is calculated are determined, the actual value of the error is calculated from the values of $H(j\omega_m)$ and $\overline{H}(j\omega_m)$ and the gradient vector is constructed from the values of $\frac{\partial HBR_m}{\partial a_i}$, etc. This error function value and the gradient vector are then returned to FMFP.

The third program, GRAD3, is a subroutine called from FUN2 for each value of ω_m. It not only calculates the value of HBR_m and HBI_m at ω_m, but also the individual partial derivatives, $\frac{\partial HBR_m}{\partial a_i}$, etc., required to form the gradient vector in FUN2. GRAD3 is also called from OPT to provide the values of HBR and HBI required in printing the real and imaginary parts of the optimized $\overline{H}(j\omega)$ at $\pi/12T$ increments. These programs were used with an IBM 360/67 at the University of New Mexico. They can easily be modified for any other FORTRAN IV compiler. A flow chart of this technique is included as Figure 6.

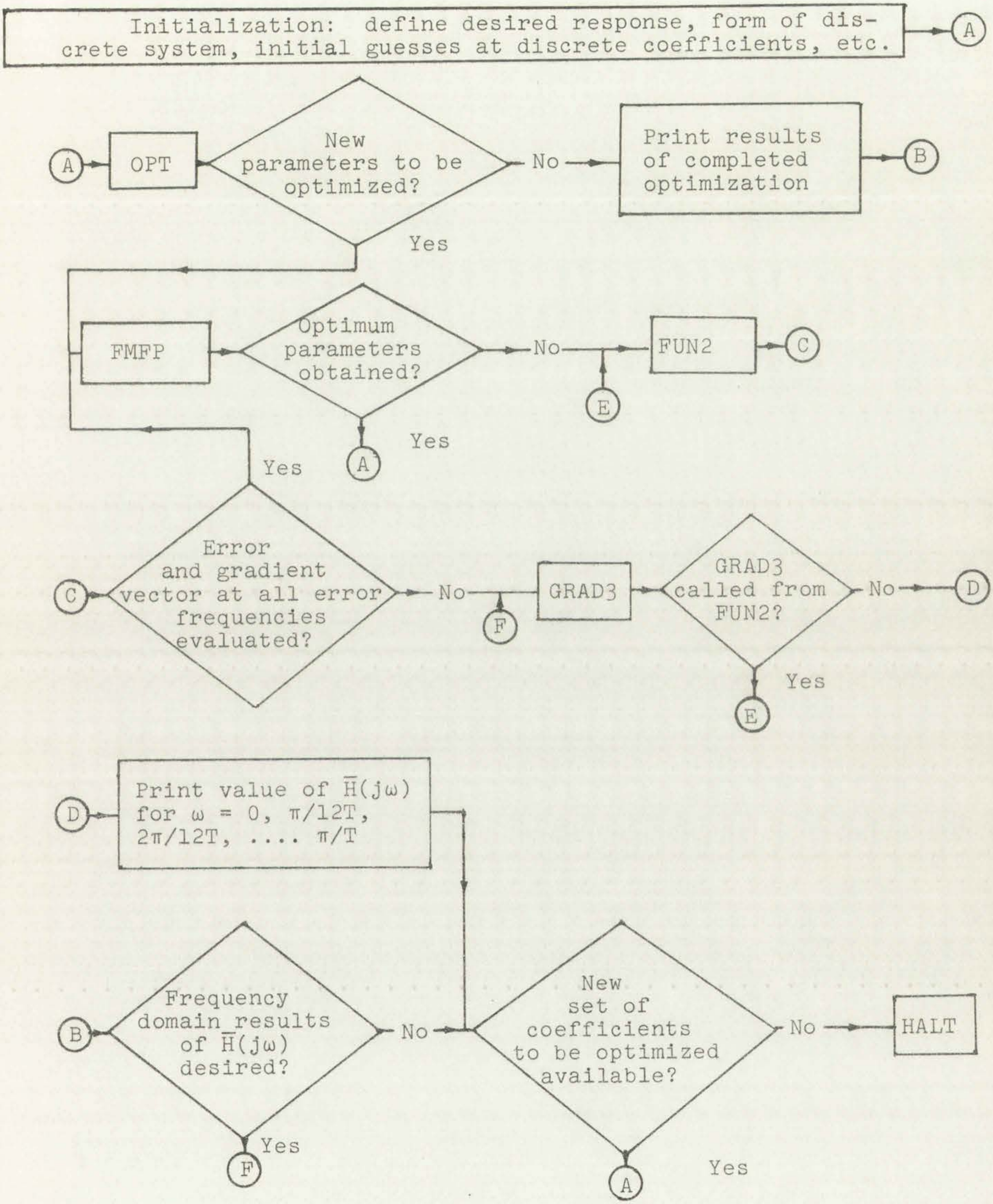

Initialization: define desired response, form of discrete system, initial guesses at discrete coefficients, etc.
A
OPT
New parameters to be optimized?
No
Yes
Print results of completed optimization
B
FMFP
Optimum parameters obtained?
No
Yes
FUN2
C
E
Error and gradient vector at all error frequencies evaluated?
No
Yes
GRAD3
F
GRAD3 called from FUN2?
No
Yes
D
Print value of $\overline{H}(j\omega)$ for ω = 0, π/12T, 2π/12T, ... π/T
Frequency domain results of $\overline{H}(j\omega)$ desired?
New set of coefficients to be optimized available?
HALT

Figure 6. Flow Chart of Optimization Procedure

CHAPTER 7. A COMPARISON OF SIMULATION METHODS

One of the stated objectives of this paper is to compare the digital simulation obtained by the previously described optimization procedure with the bilinear and invariant simulations. In this chapter, the first and second order systems to be used for comparison will be defined, the necessary equations will be derived, and the comparisons will be presented. A brief divergence to the simulation of a third order system is made simply to show the increasing advantage of optimization with the higher order system.

In Chapter 6 an argument was presented for restricting the magnitude of the discrete z-plane poles to be less than 1, and in particular to be less than exp[T Real(s-plane pole)]. Here we will present not only the results of the optimization with the above restriction, but also the results from the optimized simulation whose poles are allowed to reach a radius of .9 and .999. Even though these latter simulations are not considered to be of practical value, they do demonstrate the value of this simulation method. In particular, they show the small decrease in frequency domain error together with the large increase in the time domain error associated with the increase in pole magnitude.

Due to the number of simulations being compared and the potential resulting confusion, a three-character code is introduced here to standardize and simplify the notation to

to be used in this chapter. The first character will represent the order of the continuous system being simulated and is limited to the digits 1, 2, or 3. The second character will represent the type of discrete simulation system being compared. For example, the letter B will represent the bilinear transformation while the number 3 represents a third order optimized discrete system. The third character represents the maximum magnitude of the poles of the optimized discrete system. The complete code is given below:

(First character, second character, third character)

First character;

1 ≡ first order simulated continuous system
2 ≡ second order simulated continuous system
3 ≡ third order simulated continuous system

Second character;

A ≡ the discrete simulation is dc gain adjusted, impulse invariant
B ≡ the discrete simulation is obtained from the bilinear transformation
I ≡ the discrete simulation is impulse invariant
R ≡ the discrete simulation is ramp invariant
S ≡ the discrete simulation is step invariant
1 ≡ the discrete simulation is first order, optimized
2 ≡ the discrete simulation is second order, optimized

3 ≡ the discrete simulation is third order, optimized

4 ≡ the discrete simulation is fourth order, optimized

Third character;

0 ≡ no maximum specified for the z-plane pole radius

1 ≡ maximum z-plane pole magnitude is exp[T Real(s-plane pole)]

2 ≡ maximum z-plane pole magnitude is .9

3 ≡ maximum z-plane pole magnitude is .999

For example, the third order optimized digital simulation of the first order continuous system considered here with the pole magnitude restricted to .9 is coded (1,3,2). This code will be used throughout the remainder of this chapter.

7.1 Derivation of the Desired Responses and the Discrete Simulation Coefficients for the First Order Continuous System

The first order system to be simulated is

$$H(s) = \frac{1}{s + 1} \tag{336}$$

with a sampling interval T of π/10 seconds. The frequency response of this transfer function is found simply by substituting jω for s in (336). This gives

$$H(j\omega) = \frac{1}{1 + j\omega} = \frac{1 - j\omega}{1 + \omega^2} . \qquad (337)$$

The actual response of this system will be compared with the responses of the simulation systems for the following inputs: impulse, step, ramp, sin 3t and sin 7t. These inputs were chosen for their distinct spectral characteristics in relation to the sampling rate. The sinusoidal inputs, sin 3t and sin 7t, are at 3/10 and 7/10 of the Nyquist frequency, respectively. The impulse, step, and ramp inputs contain energy at all frequencies; only the shape of the energy density vs. frequency curves vary. The energy density vs. frequency curve of the impulse function is flat while that of the step function decreases as $1/|f|$ and that of the ramp falls off as $1/f^2$. The continuous system output for these inputs is given below:

1. Impulse response; $x(t) = \delta(t)$, $X(s) = 1$

$$Y(s) = H(s) = \frac{1}{s + 1} \qquad (338)$$

$$y(t) = e^{-t} \qquad (339)$$

2. Step response; $x(t) = u(t)$, $X(s) = \frac{1}{s}$

$$Y(s) = \frac{1}{s} H(s) = \frac{1}{s(s + 1)} \qquad (340)$$

$$y(t) = 1 - e^{-t} \qquad (341)$$

3. Ramp response; $x(t) = .2t$, $X(s) = \frac{.2}{s^2}$

$$Y(s) = \frac{.2}{s^2} H(s) = \frac{.2}{s^2(s + 1)} \tag{342}$$

$$y(t) = .2(t + e^{-t} - 1) \tag{343}$$

4. Sin 3t response; $x(t) = \sin 3t$, $X(s) = \frac{3}{s^2 + 3^2}$

$$Y(s) = \frac{3}{s^2 + 9} H(s) = \frac{3}{(s^2 + 9)(s + 1)} \tag{344}$$

$$y(t) = .3\left(e^{-t} - \cos 3t + \frac{1}{3} \sin 3t\right) \tag{345}$$

5. Sin 7t response, $x(t) = \sin 7t$, $X(s) = \frac{7}{s^2 + 7^2}$

$$Y(s) = \frac{7}{s^2 + 49} H(s) = \frac{7}{(s^2 + 49)(s + 1)} \tag{346}$$

$$y(t) = .14\left(e^{-t} - \cos 7t + \frac{1}{7} \sin 7t\right) . \tag{347}$$

The simulation system obtained from the bilinear substitution is derived here. We have chosen the constant "a" in (147) to be 5 so that $\overline{H}(j\omega) = H(j\omega)$ for $\omega = \pi/2T$, i.e., half the Nyquist frequency. This gives

$$\overline{H}(j\omega) = H(j5 \tan \omega T/2) \tag{348}$$

and for $\omega = \pi/2T = 5$ with $T = .1\pi$

$$\overline{H}(j5) = H(j5 \tan \pi/4) = H(j5) . \tag{349}$$

Now $\tilde{H}(z)$ is obtained from H(s) simply by substituting $5\,\frac{z-1}{z+1}$ for s in H(s)

$$\tilde{H}(z) = H(s)\Big|_{s\to 5\frac{z-1}{z+1}} = \frac{1}{5\,\frac{z-1}{z+1}+1} = \frac{z+1}{6z-4} \qquad (350)$$

or

$$\tilde{H}(z) = \frac{1}{6}\,\frac{1+z^{-1}}{1-\frac{2}{3}z^{-1}} \qquad (351)$$

with the three character code given as (1,B,0). The corresponding difference equation is

$$y_n = \frac{1}{6}(x_n + x_{n-1}) + \frac{2}{3}y_{n-1} \,. \qquad (352)$$

The impulse invariant simulation is found from (339). For $y(t) = e^{-t}$ and $x(t) = \delta(t)$ we have

$$\tilde{Y}(z) = \frac{z}{z-e^{-T}} = \frac{1}{1-e^{-T}z^{-1}} \qquad (353)$$

and, for the discrete impulse function,

$$\tilde{X}(z) = \frac{1}{T} \,. \qquad (354)$$

Thus,

$$\tilde{H}(z) = \frac{\tilde{Y}(z)}{\tilde{X}(z)} = \frac{T}{1-e^{-T}z^{-1}} \,; \qquad (1,I,0) \qquad (355)$$

The dc gain adjusted impulse invariant simulation is found directly from (355). We simply require $\tilde{H}(1) = H(0) = 1$ or

$$\tilde{H}(z) = \frac{(1 - e^{-T})}{1 - e^{-T}z^{-1}} \; ; \qquad (1,A,0) \tag{356}$$

The step invariant simulation can be found from (341). For $y(t) = 1 - e^{-t}$ and $x(t) = u(t)$ we have

$$\tilde{Y}(z) = \frac{z}{z - 1} - \frac{z}{z - e^{-T}} = \frac{z}{z - 1} \, \frac{1 - e^{-T}}{z - e^{-T}} \tag{357}$$

and

$$\tilde{X}(z) = \frac{z}{z - 1} \; . \tag{358}$$

Thus,

$$\tilde{H}(z) = \frac{\tilde{Y}(z)}{\tilde{X}(z)} = \frac{(1 - e^{-T})z^{-1}}{1 - e^{-T}z^{-1}} \; ; \qquad (1,S,0) \tag{359}$$

Similarly, the ramp invariant simulation is found from (343). For $y(t) = .2(t + e^{-t} - 1)$ and $x(t) = .2t$, we have

$$\tilde{Y}(z) = \frac{Tz}{(z - 1)^2} \, \frac{z(T - 1 + e^{-T}) + (1 - Te^{-T} - e^{-T})}{T(z - e^{-T})} \tag{360}$$

and

$$\tilde{X}(z) = \frac{Tz}{(z - 1)^2} \; . \tag{361}$$

Thus,

$$\tilde{H}(z) = \frac{\tilde{Y}(z)}{\tilde{X}(z)} = \frac{(T - 1 + e^{-T}) + (1 - Te^{-T} - e^{-T})z^{-1}}{T(1 - e^{-T}z^{-1})} ; \qquad (362)$$

(1,R,0)

Several different optimized systems were obtained and their performance compared to the above systems. In order to keep the comparisons based strictly on the number of coefficients and allowed magnitude of the poles, the frequencies at which the error was determined were always the same. These "error frequencies" were at the following values of ωT: $\pi/140$, $2\pi/140$, $3\pi/140$, ... $\frac{100\pi}{140}$. This is a relatively large number of frequencies; however, it was chosen to closely approximate a least mean squared error measure for frequencies from d.c. to approximately .7 the Nyquist frequency. In order to reduce the computational time needed to obtain these coefficients, a two step procedure was used. First the "error frequencies" were chosen to be at $\omega T = \pi/7$, $2\pi/7$, ... $5\pi/7$, and the optimum coefficients found. Then, using these coefficients as "guesses" for the optimum system based on the error at $\pi/140$, $2\pi/140$, ..., etc., the final values of the coefficients were obtained. Note that the above frequencies are used only for the optimization process; they are not used for the comparisons of this chapter. Although the coefficient values from the first optimization differed little from the final ones, the transfer function error vs. frequency curve obtained from the final coefficients was smoother than that found from the preliminary coefficients.

All poles were assumed to be real in the optimized system. Intuitively, this would be expected as the continuous system pole is real. Empirically, it was found that if a pair of complex poles were used the resulting simulation was poor. All optimum simulations in this paper have an equal number of poles and zeroes. This is the form generally assumed for a rational function and again it was empirically found to produce the best results for a given number of coefficients.

The effects of both the number of poles and zeroes allowed and the maximum radius of the poles were compared. Optimized simulations with one, two, three, and four pole-zeroes were found with the maximum pole radius restricted to $e^{-T} = .730403$, .9, and .999. Thus, twelve simulations were found with all of the first order simulations being the same, since the optimum simple pole location was within the minimum of the allowed radii, e^{-T}.

These simulations are listed below, first those with the poles restricted to e^{-T}, then .9, and finally .999. All coefficients are rounded to six significant digits.

$|\text{Poles}| \leq e^{-T} = .730403$

1. $$\tilde{H}(z) = K\,\frac{1 + .986629z^{-1}}{1 - .713722z^{-1}}\,;\quad (1,1,1) \tag{363}$$

2. $$\tilde{H}(z) = K\,\frac{1 + 2.42315z^{-1} + .518598z^{-2}}{(1 - .729328z^{-1})(1 + .730403z^{-1})}\,;\quad (1,2,1) \tag{364}$$

3. $$\tilde{H}(z) = K \frac{(1 + 2.91600z^{-1} + 1.07906z^{-2})(1 + .434916z^{-1})}{(1 - .727669z^{-1})(1 + .730403z^{-1})^{2}} \qquad (1,3,1) \qquad (365)$$

4. $$\tilde{H}(z) = K(1 + 3.21734z^{-1} + .452531z^{-2}) (1 + 1.26318z^{-1} + .571899z^{-2}) \Big/ (1 - .730277z^{-1})\left(1 + .730403z^{-1}\right)^{3} \ ; \quad (1,4,1) \qquad (366)$$

$|\mathrm{Poles}| \leq .9$

5. Same as (363); (1,1,2) (367)

6. $$\tilde{H}(z) = K \frac{(1 + 2.78393z^{-1} + .743926z^{-2})}{(1 - .728308z^{-1})(1 + .900000z^{-1})} \ ; \quad (1,2,2) \qquad (368)$$

7. $$\tilde{H}(z) = K \frac{(1 + 3.05733z^{-1} + .946039z^{-2})(1 + .777604z^{-1})}{(1 - .727542z^{-1})(1 + .900000z^{-1})^{2}} \qquad (1,3,2) \qquad (369)$$

8. $$\tilde{H}(z) = K\left(1 + 3.72221z^{-1} + .727741z^{-2}\right) \left(1 + 1.55583z^{-1} + .808569z^{-2}\right) \Big/ (1 - .729904z^{-1})\left(1 + .900000z^{-1}\right)^{3} \ ; \quad (1,4,2) \qquad (370)$$

$|\mathrm{Poles}| \leq .999$

9. Same as (363); (1,1,3) (371)

10. $$\tilde{H}(z) = K \frac{(1 + 3.01207z^{-1} + .885306z^{-2})}{(1 - .727803z^{-1})(1 + .999z^{-1})} ; \quad (1,2,3) \tag{372}$$

11. $$\tilde{H}(z) = K \frac{(1 + 3.14574z^{-1} + .973324z^{-2})(1 + .943297z^{-1})}{(1 - .727515z^{-1})(1 + .999z^{-1})^2} \quad (1,3,3) \tag{373}$$

12. $$\tilde{H}(z) = K\left(1 + 4.05869z^{-1} + .892474z^{-2}\right)\left(1 + 1.71773z^{-1} + .972806z^{-2}\right)\Big/ (1 - .729697z^{-1})(1 + .999z^{-1})^3 ; \quad (1,4,3) \tag{374}$$

In every equation, (363) through (374), K is chosen so that $\tilde{H}(z)\Big|_{z=1} = 1$. Now the d.c. gain of the simulation system is identical to that of the desired system.

7.2 Derivation of the Desired Responses and the Discrete Coefficients for the Second Order Continuous System

Similar equations must be derived for the second order system, defined to be

$$H(s) = \frac{1}{(s + 1)^2 + 2^2} = \frac{1}{s^2 + 2s + 5} \tag{375}$$

with the frequency response given by

$$H(j\omega) = \frac{1}{(5 - \omega^2) + j2\omega} = \frac{5 - \omega^2 - j2\omega}{(5 - \omega^2)^2 + 4\omega^2} . \tag{376}$$

The response of this system to the various inputs is given below:

1. Impulse response; $x(t) = \delta(t)$, $X(s) = 1$

$$Y(s) = H(s) = \frac{1}{s^2 + 2s + 5} \tag{377}$$

$$y(t) = .5e^{-t} \sin 2t \tag{378}$$

2. Step response; $x(t) = u(t)$, $X(s) = \frac{1}{s}$

$$Y(s) = \frac{1}{s} H(s) = \frac{1}{s(s^2 + 2s + 5)} \tag{379}$$

$$y(t) = .2 - .1e^{-t}(\sin 2t + 2 \cos 2t) \tag{380}$$

3. Ramp response; $x(t) = .2t$, $X(s) = \frac{.2}{s^2}$

$$Y(s) = \frac{.2}{s^2} H(s) = \frac{.2}{s^2(s^2 + 2s + 5)} \tag{381}$$

$$y(t) = .04\left(t - .4(1 - e^{-t} \cos 2t) - .3e^{-t} \sin 2t\right) \tag{382}$$

4. Sin 3t response; $x(t) = \sin 3t$, $X(s) = \frac{3}{s^2 + 3^2}$

$$Y(s) = \frac{3H(s)}{s^2 + 9} = \frac{3}{(s^2 + 9)(s^2 + 2s + 5)} \tag{383}$$

$$y(t) = \left(\frac{3}{26}\right)\left(e^{-t}(\cos 2t + 1.5 \sin 2t) - \cos 3t - \frac{2}{3} \sin 3t\right) \tag{384}$$

5. Sin 7t response; $x(t) = \sin 7t$, $X(s) = \frac{7}{s^2 + 7^2}$

$$Y(s) = \frac{7H(s)}{s^2 + 49} = \frac{7}{(s^2 + 49)(s^2 + 2s + 5)} \tag{385}$$

$$y(t) = A \sin 7t + B \cos 7t + e^{-t}\left(\frac{C - D}{2} \sin 2t + D \cos 2t\right) \tag{386}$$

where

$$A = -11/537,\ B = \frac{-7}{1074},\ C = \frac{84}{537},\ D = -B\ . \tag{387}$$

The simulation system obtained from the bilinear substitution is given, as before, by

$$\tilde{H}(z) = H(s)\Big|_{s \to 5\, \frac{z - 1}{z + 1}} \tag{388}$$

$$\tilde{H}(z) = \frac{1}{\left(5\, \frac{z - 1}{z + 1}\right)^2 + 2\left(5\, \frac{z - 1}{z + 1}\right) + 5} \tag{389}$$

or

$$\tilde{H}(z) = \frac{\left(1 + z^{-1}\right)^2}{40(1 - z^{-1} + .5z^{-2})}\ ;\quad (2,B,0) \tag{390}$$

The impulse invariant simulation is found from (378). With $x_0 = \frac{1}{T}$ and $y(t) = .5e^{-t} \sin 2t$ we have

$$\tilde{X}(z) = \frac{1}{T} \tag{391}$$

and

$$\tilde{Y}(z) = \frac{1}{2} \frac{e^{-T} \sin 2Tz}{z^2 - 2e^{-T} \cos 2Tz + e^{-2T}} \tag{392}$$

giving

$$\tilde{H}(z) = \frac{T}{2} \frac{e^{-T} \sin 2Tz^{-1}}{1 - 2e^{-T} \cos 2Tz^{-1} + e^{-2T}z^{-2}} \; ; \quad (2,I,0) \tag{393}$$

The d.c. gain adjusted impulse invariant simulation is obtained directly from (393) and noting that H(0) = .2, the result is

$$\tilde{H}(z) = \frac{.2(1 - 2e^{-T} \cos 2T + e^{-2T})z^{-1}}{1 - 2e^{-T} \cos 2Tz^{-1} + e^{-2T}z^{-2}} \; ; \quad (2,A,0) \tag{394}$$

The step invariant simulation is found from (380). From $x(t) = u(t)$ and $y(t) = .2 - .1e^{-t}(\sin 2t + 2 \cos 2t)$ we have

$$\tilde{X}(z) = \frac{z}{z - 1} \tag{395}$$

and

$$\tilde{Y}(z) = .2 \frac{z}{z - 1} \left[\left(1 - e^{-T}\cos 2T - \frac{1}{2} e^{-T} \sin 2T\right) + \left(\frac{1}{2} e^{-T}\sin 2T - e^{-T}\cos 2T + e^{-2T}\right)z^{-1} \right] \Big/ \left[1 - 2e^{-T}\cos 2Tz^{-1} + e^{-2T}z^{-2}\right] \tag{396}$$

giving

$$\tilde{H}(z) = .2\left[\left(1 - e^{-T}\cos 2T - \frac{1}{2}e^{-T}\sin 2T\right) + \left(\frac{1}{2}e^{-T}\sin 2T - e^{-T}\cos 2T + e^{-2T}\right)z^{-1}\right] \Big/ \left[1 - 2e^{-T}\cos 2Tz^{-1} + e^{-2T}z^{-2}\right] \; ; \quad (2,S,0) \qquad (397)$$

Similarly, the ramp invariant simulation is found from (382). We have $x(t) = .2T$ and $y(t) = .04\left(t - .4(1 - e^{-t}\cos 2t) - .3e^{-t}\sin 2t\right)$ with

$$\tilde{X}(z) = \frac{Tz}{(z - 1)^2} \qquad (398)$$

and

$$\tilde{Y}(z) = \frac{.2z}{(z - 1)^2}\left\{\left[T + .4(e^{-T}\cos 2T - 1) - .3e^{-T}\sin 2T\right]z^2 + \left[-2e^{-T}T\cos 2T + .4(1 - e^{-2T}) + .6e^{-T}\sin 2T\right]z + \left[Te^{-2T} + .4e^{-2T} - .4e^{-T}\cos 2T - .3e^{-T}\sin 2T\right]\right\} \Big/ \left[z^2 - 2e^{-T}\cos 2Tz + e^{-2T}\right] . \qquad (399)$$

Thus,

$$\tilde{H}(z) = \frac{.2}{T}\Big\{\Big[T + .4(e^{-T}\cos 2T - 1) - .3e^{-T}\sin 2T\Big]$$

$$+ \Big[-2Te^{-T}\cos 2T + .4(1 - e^{-2T}) + .6e^{-T}\sin 2T\Big]z^{-1}$$

$$+ \Big[Te^{-2T} + .4e^{-2T} - .4e^{-T}\cos 2T - .3e^{-T}\sin 2T\Big]z^{-2}\Big\}\Big/$$

$$\Big[1 - 2e^{-T}\cos 2Tz^{-1} + e^{-2T}z^{-2}\Big] \; ; \qquad (2,R,0) \qquad (400)$$

The procedure used to obtain the optimized system here is very similar to that for the first order system. The two differences are: no first order simulation of the second order system is shown as it is very poor, and two of the poles of the simulation systems are assumed to be complex. Intuitively one would expect complex z-plane poles in the simulation of a system with complex s-plane poles; empirically this was verified as it was found that assuming all real poles produced very poor results. In the fourth order system one could assume either one or two pairs of complex poles as the third and fourth pole occur as a pair on the real axis; thus, either assumption results in the same pole placement. These optimized systems are listed below:

$\underline{|\text{Poles}| \leq e^{-T} = .730403}$

1.

$$\tilde{H}(z) = K \frac{(1 + 6.39557z^{-1} + .539820z^{-2})}{(1 - .730403e^{j.628746}z^{-1})(1 - .730403e^{-j.628746}z^{-1})};$$

$$(2,2,1) \qquad (401)$$

2.

$$\tilde{H}(z) = K(1 + 7.09568z^{-1} + .891171z^{-2})(1 + .635019z^{-1}) \Big/$$

$$\Big(1 - 2(\cos .630774).730403z^{-1}$$

$$+ .730403^2 z^{-2}\Big)(1 + .730403z^{-1}) \quad ; \quad (2,3,1) \qquad (402)$$

3.

$$\tilde{H}(z) = K(1 + 7.96772z^{-1} + .409879z^{-2})(1 + 1.35049z^{-1}$$

$$+ .559528z^{-2}) \Big/ \Big(1 - 2(\cos .627864).730403z^{-1}$$

$$+ .730403^2 z^{-2}\Big)(1 + .730403z^{-1})^2 \quad ; \quad (2,4,1) \qquad (403)$$

<u>$|\text{Poles}| \leq .9$</u>

4.

$$\tilde{H}(z) = K \frac{(1 + 6.34432z^{-1} + .512971z^{-2})}{\Big(1 - 2(\cos .627962).731516z^{-1} + .731516z^{-2}\Big)} ; \qquad (404)$$

$$(2,2,2)$$

5.

$$\tilde{H}(z) = K(1 + 7.23990z^{-1} + .845040z^{-2})(1 + .793856z^{-1}) \Big/$$

$$\Big(1 - 2(\cos .630216).730900z^{-1}$$

$$+ .730900^2 z^{-2}\Big)(1 + .9z^{-1}) \quad ; \quad (2,3,2) \qquad (405)$$

6.

$$\begin{aligned}\tilde{H}(z) = K(1 &+ 8.46211z^{-1} + .582782z^{-2})(1 + 1.62063z^{-1} \\ &+ .771421z^{-2}) \Big/ \big(1 - 2(\cos .628114).730675z^{-1} \\ &+ .730675^{2}z^{-2}\big)\big(1 + .9z^{-1}\big)^{2} \quad ; \quad (2,4,2) \qquad (406)\end{aligned}$$

<u>|Poles| ≤ .999</u>

7. Same as (404 ; (2,2,3) (407)

8.

$$\begin{aligned}\tilde{H}(z) = K(1 &+ 7.38356z^{-1} + .848842z^{-2})(1 + .880023z^{-1}) \Big/ \\ &\big(1 - 2(\cos .630185).730851z^{-1} \\ &+ .730851^{2}z^{-2}\big)(1 + .999z^{-1}) \quad ; \quad (2,3,3) \qquad (408)\end{aligned}$$

9.

$$\begin{aligned}\tilde{H}(z) = K(1 &+ 8.81417z^{-1} + .669612z^{-2})(1 + 1.77512z^{-1} \\ &+ .912479z^{-2}) \Big/ \big(1 - 2(\cos .628326).730658z^{-1} \\ &+ .730658z^{-2}\big)\big(1 + .999z^{-1}\big)^{2} \quad ; \quad (2,4,3) \qquad (409)\end{aligned}$$

7.3 Error Comparisons of Simulations of the First Order System

With the equations derived above the various simulations can be compared, both in the frequency domain and the time domain, for selected inputs.

7.3.1 Frequency Domain Errors

The frequency domain error comparisons will be made first. Since the definition of the error used here involves both the magnitude and phase of the transfer functions, the comparisons will be made as a plot of the approximation error rather than as a plot of the desired and simulation system transfer functions. The error function minimized was given as

$$E = \sum_{k=1}^{K} \left|H(j\omega_k) - \overline{H}(j\omega_k)\right|^2 , \quad 0 < \omega_1 < \omega_2 < \ldots < \omega_K < \pi/T \tag{289}$$

The error plotted in Figures 7-12 and 27-32 is given by

$$E(\omega) = \log_{10}\left(\left|H(j\omega) - \overline{H}(j\omega)\right| + 10^{-6}\right). \tag{410}$$

The range of $E(\omega)$ is usually from -1 to -4, thus the term 10^{-6} has no noticeable effect on the curve. However, at values of ω where $\overline{H}(j\omega) = H(j\omega)$ this term merely keeps the curve on the plot and avoids the problem of finding $\log_{10}(0)$. Neglecting this term, 10^{-6}, the error function minimized, given in (289), can be obtained on the plot by simply doubling the ordinate values. Figure 7 is a plot of the error of the impulse invariant (1,I,0), adjusted impulse invariant (1,A,0), and step invariant (1,S,0) simulations of the first order system. Note that while the impulse invariant simulation has nearly constant error over the entire frequency range the adjusted impulse invariant and step invariant simulations are exact for $\omega = 0$

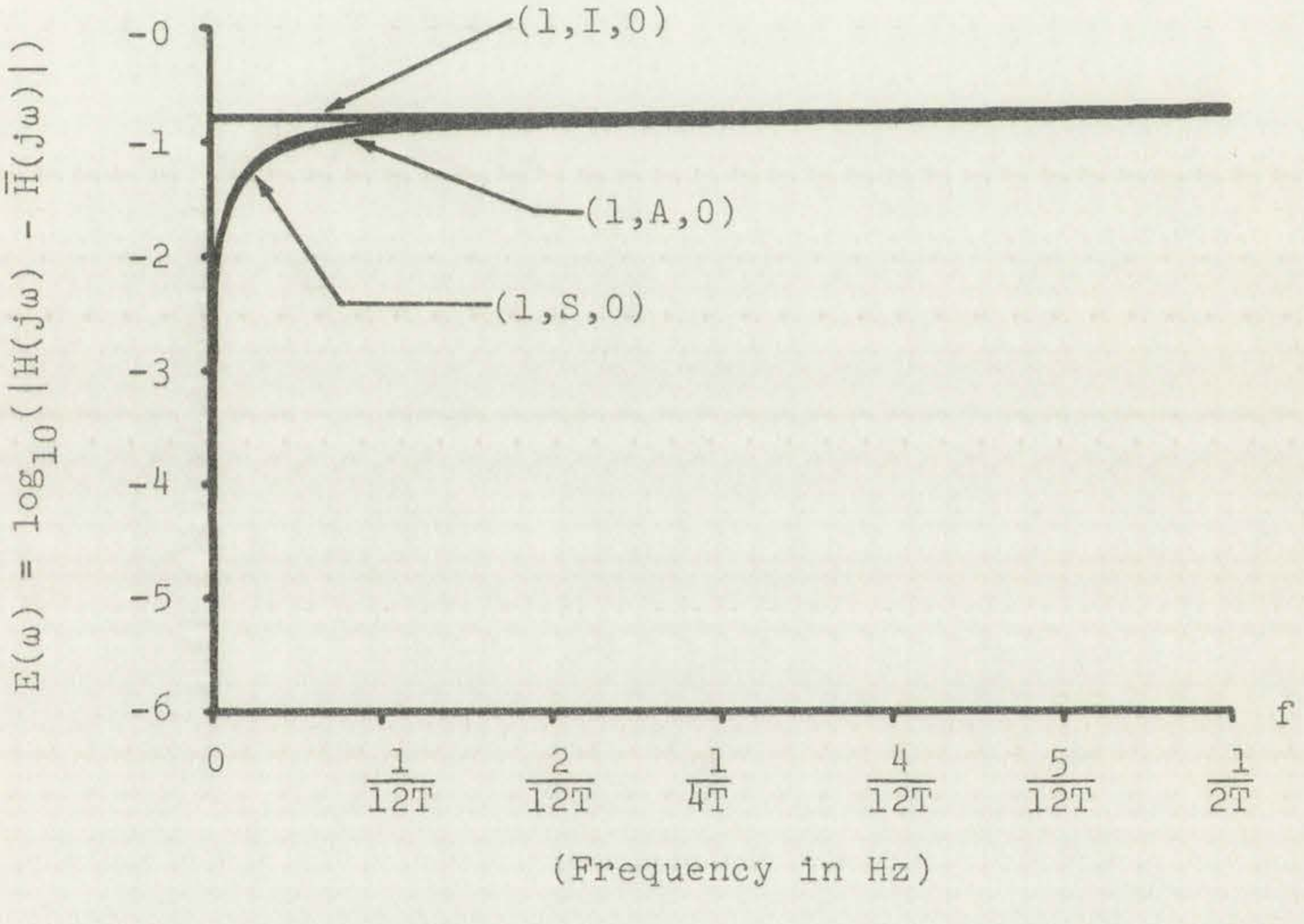


Figure 7. Frequency Domain Error of Impulse, Adjusted Impulse and Step Invariant Simulations of First Order System

and asymptotically approach the error of the impulse invariant systems at the higher frequencies.

Figure 8 is a comparison of the frequency domain errors of the bilinear approximation (1,B,0), the ramp invariant (1,R,0), and the optimized first order (1,1,1) simulations. These approximations are all better than those of Figure 7 throughout the frequency range. It is difficult here, visually, to determine that the optimized systems error is less than that of the ramp invariant simulation (although it can easily be shown by performing the computation of (289)). In fact, the difference is small and for general usage the smooth error curve of the ramp invariant simulation may be preferable.

Figure 9 is a comparison of the second order optimized digital systems (1,2,1), (1,2,2), and (1,2,3) simulation error. By allowing the pole on the negative real axis to approach -1 the overall error is reduced slightly; but this small reduction in frequency domain error will later be shown to be very costly in terms of the time domain response. However, on comparing the curves of Figure 9 to those of Figure 8, we see that the error has been reduced significantly from that of the first order systems.

Figure 10 is similar to Figure 9 except that here we are comparing third order simulations (1,3,1), (1,3,2), and (1,3,3). Again some improvement is noted by allowing the poles to approach -1. Similarly, Figure 11 represents the fourth order systems (1,4,1), (1,4,2), and (1,4,3).

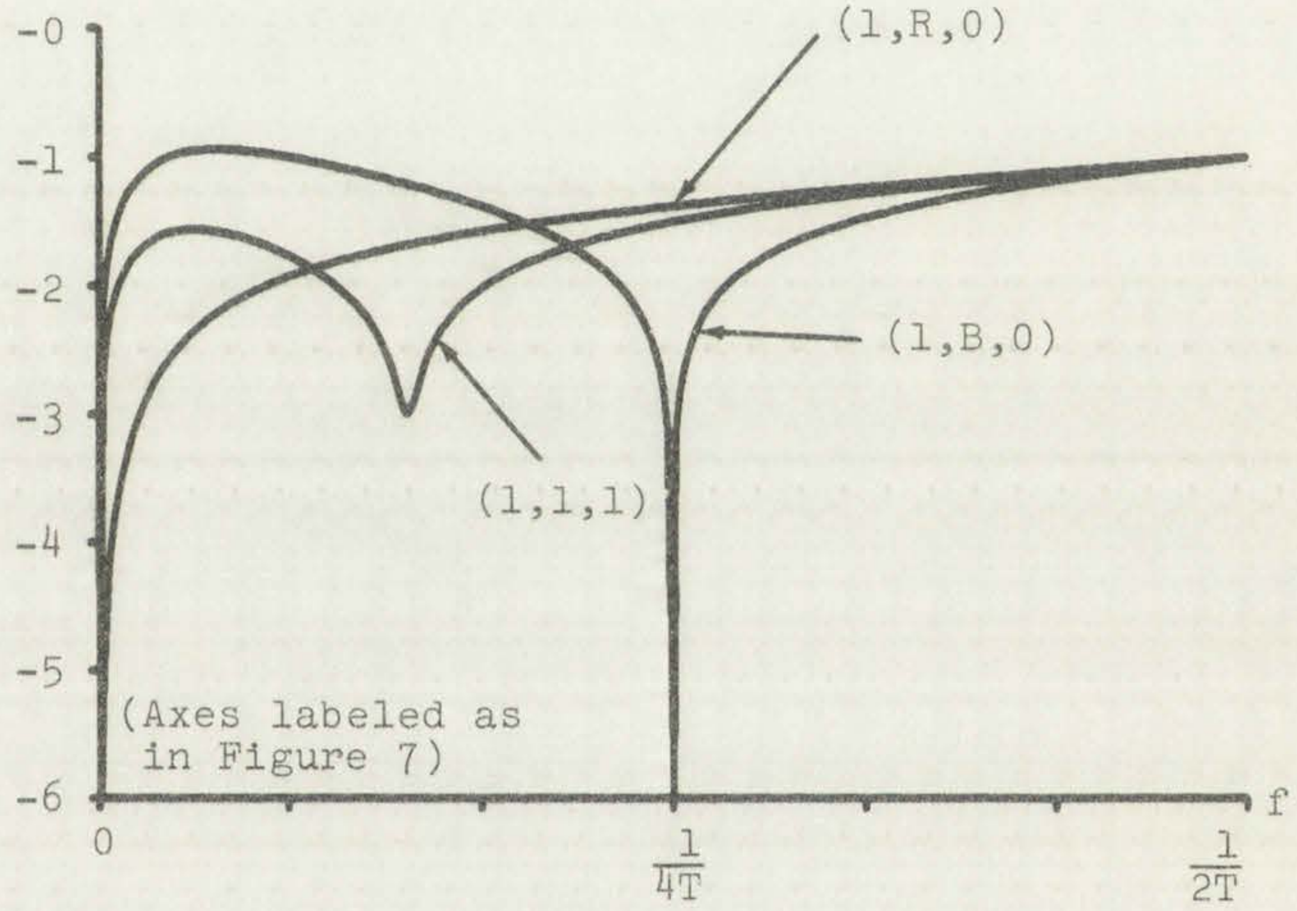


Figure 8. Frequency Domain Error of Ramp Invariant, Bilinear Approximation and First Order Optimized Simulations of First Order System

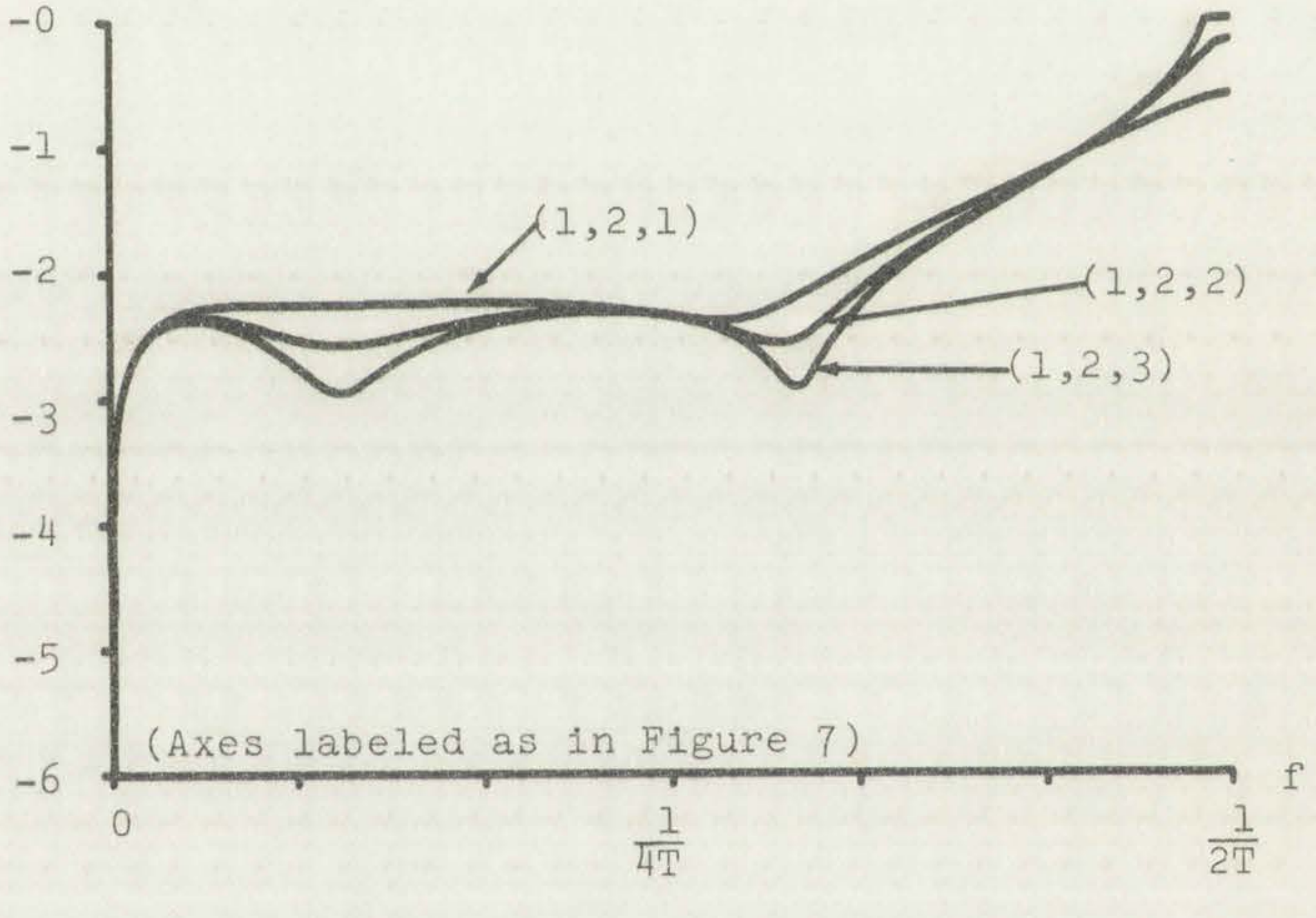


Figure 9. Frequency Domain Error of Three, Second Order Optimized Simulations of First Order System

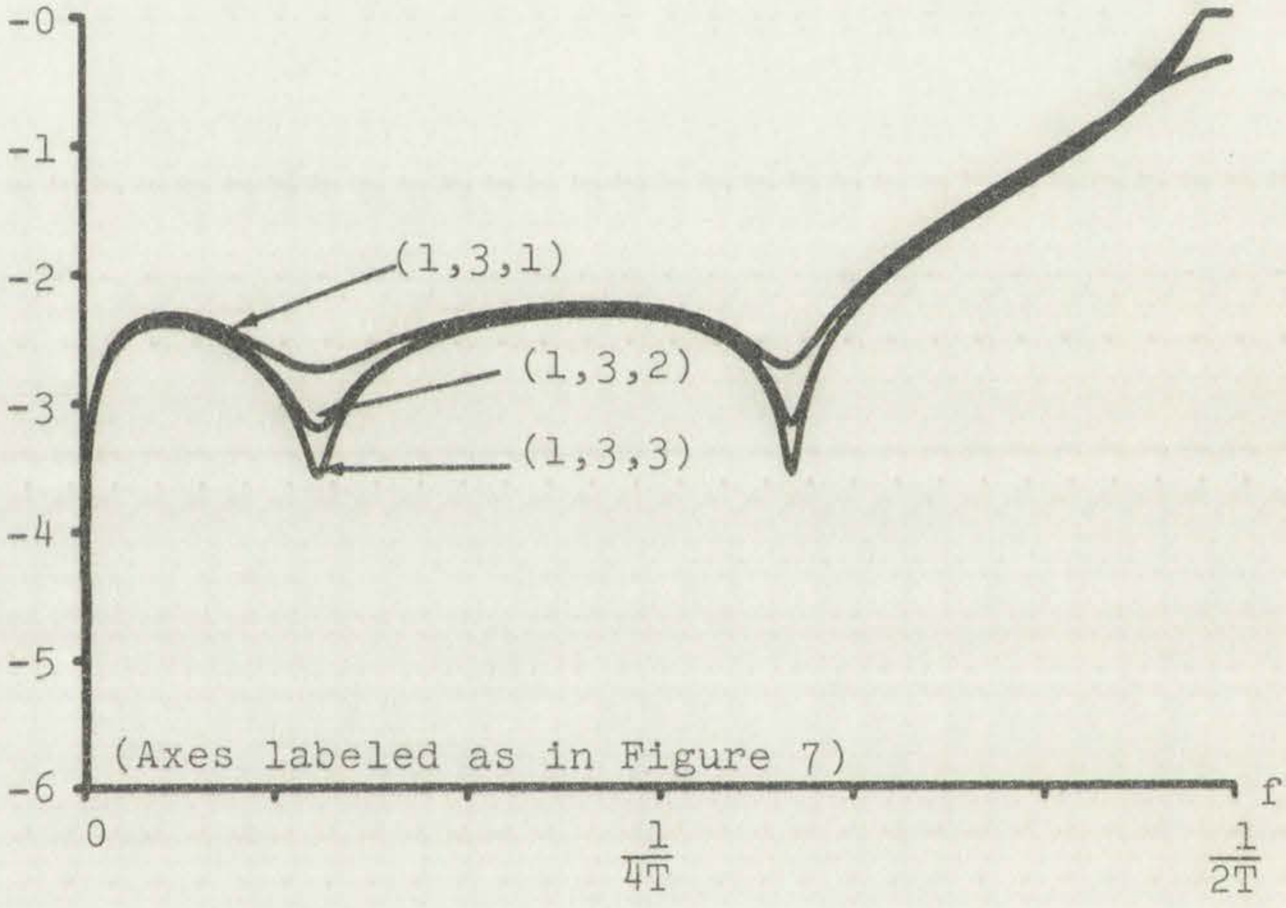


Figure 10. Frequency Domain Error of Three Third Order Optimized Simulations of First Order System

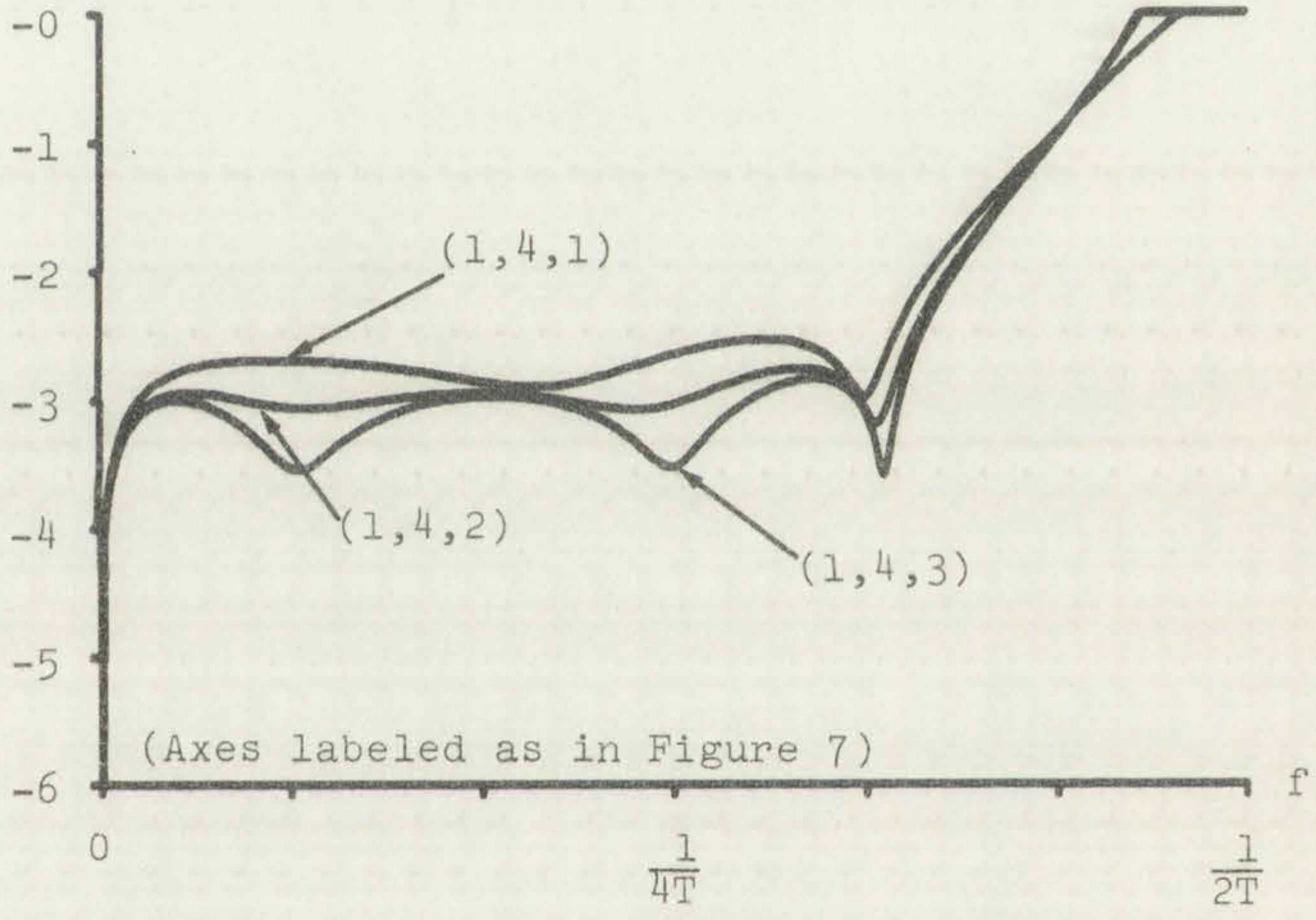


Figure 11. Frequency Domain Error of Three Fourth Order Optimized Simulations of First Order System

Figure 12 represents the errors involved in the first (1,1,1), second (1,2,1), third (1,3,1), and fourth (1,4,1) order systems. Well over an order of magnitude decrease in the simulation error is observed in going from a first to fourth order system. The second order system offers significant improvement over the first order while little over all further improvement is obtained by going from the second to the third order system.

The six figures previously discussed clearly show the reduction in frequency domain error obtained by either allowing the poles to approach the borderline of stability, the unit circle, or increasing the order of the system. However, in order to fully evaluate these simulation techniques, we must also consider the effects on the time domain response of these methods of reducing frequency domain error.

7.3.2 Time Domain Errors

In this section we will compare the outputs of these systems for the five inputs for which the continuous systems response was given in equations (339), (341), (343), (345), and (347). These inputs are: the impulse, $\delta(t)$, the unit step, $u(t)$, the ramp, $.2t$, $\sin 3t$, and $\sin 7t$. Each comparison will show the desired continuous response and the simulation systems output at each sampling time. In addition, the error, $\left|y_n - y(nT)\right|$, is plotted on a logarithmic scale directly above the actual responses y_n and $y(nT)$. With this error plot the comparisons are much easier to interpret as it is often diffi-

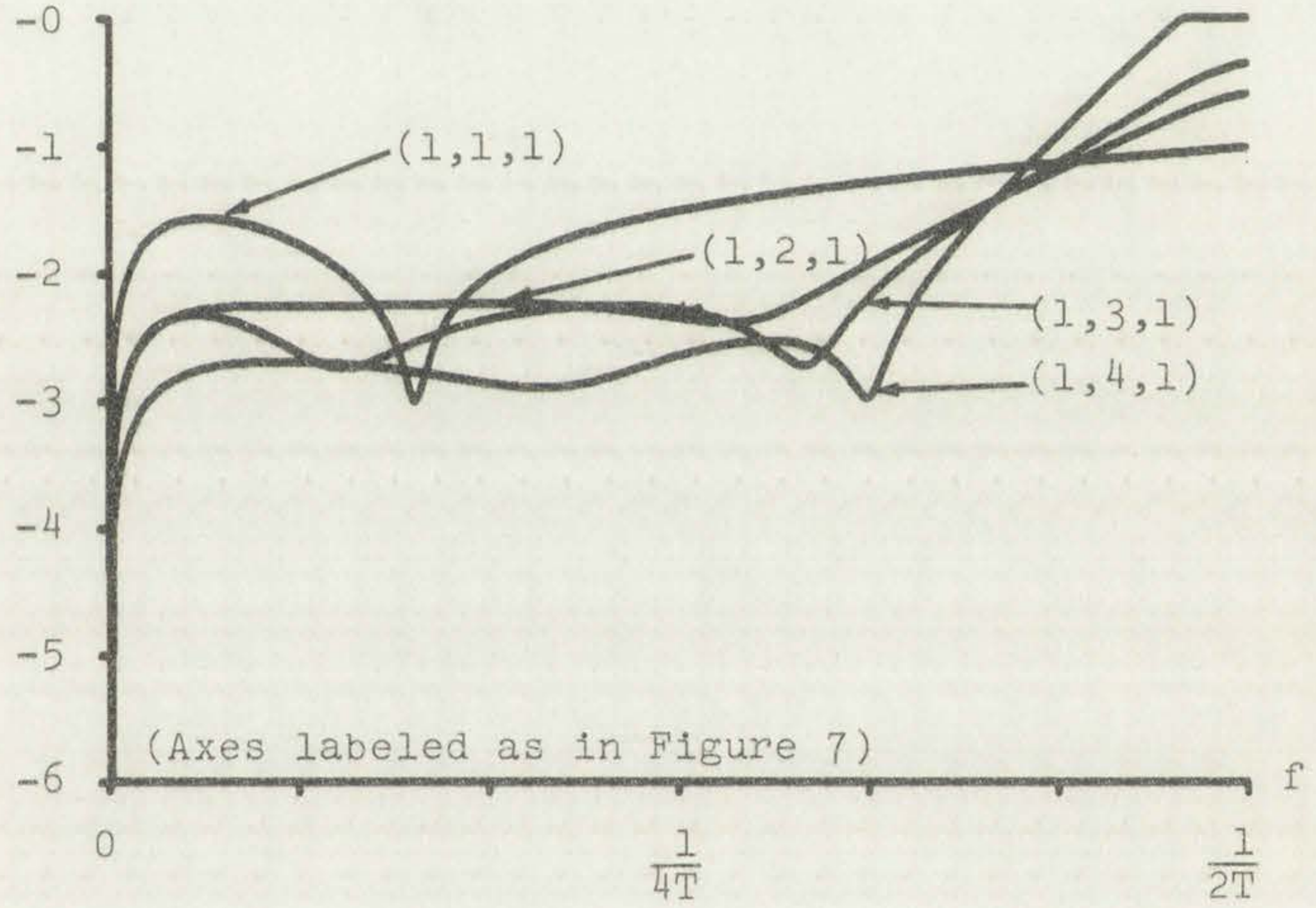


Figure 12. Frequency Domain Error of a First, Second, Third, and Fourth Order Optimized Simulations of First Order System

cult to determine which simulation, y_n, is better on an actual plot of the response. Again a "bias constant" of 10^{-6} was added to the actual error, thus the error term plotted is

$$E = \log_{10}\left(\left|y_n - y(nT)\right| + 10^{-6}\right) . \tag{411}$$

First the impulse responses will be compared. Figure 13 is a plot of the impulse response of the impulse invariant (1,I,0), adjusted impulse invariant (1,A,0), and step invariant (1,S,0) simulations. Obviously the impulse invariant simulation is exact and the error shown, 10^{-6}, is simply due to the bias constant in (411). The other two simulations have nearly equal errors, one positive and one negative and have the same rate of decay, as would be expected since their pole locations are identical.

Figure 14 represents the impulse response of the ramp invariant (1,R,0), bilinear (1,B,0), and first order optimized (1,1,1) systems. The ramp invariant is obviously the best; however, the others are converging at a rate of decay predictable from their pole placements.

Figures 15 and 16 illustrate the impulse responses of second and third order systems (1,2,1), (1,2,2), (1,2,3), and (1,3,1), (1,3,2), (1,3,3), respectively.

Each plot shows the response of a system whose poles are restricted to a radius of e^{-T}, .9, and .999. With the poles restricted to the radius e^{-T} the error is limited in duration as can be seen from the rate of decay shown in Figures 15 and 16. Using partial fraction expansions of (364) and (365) we

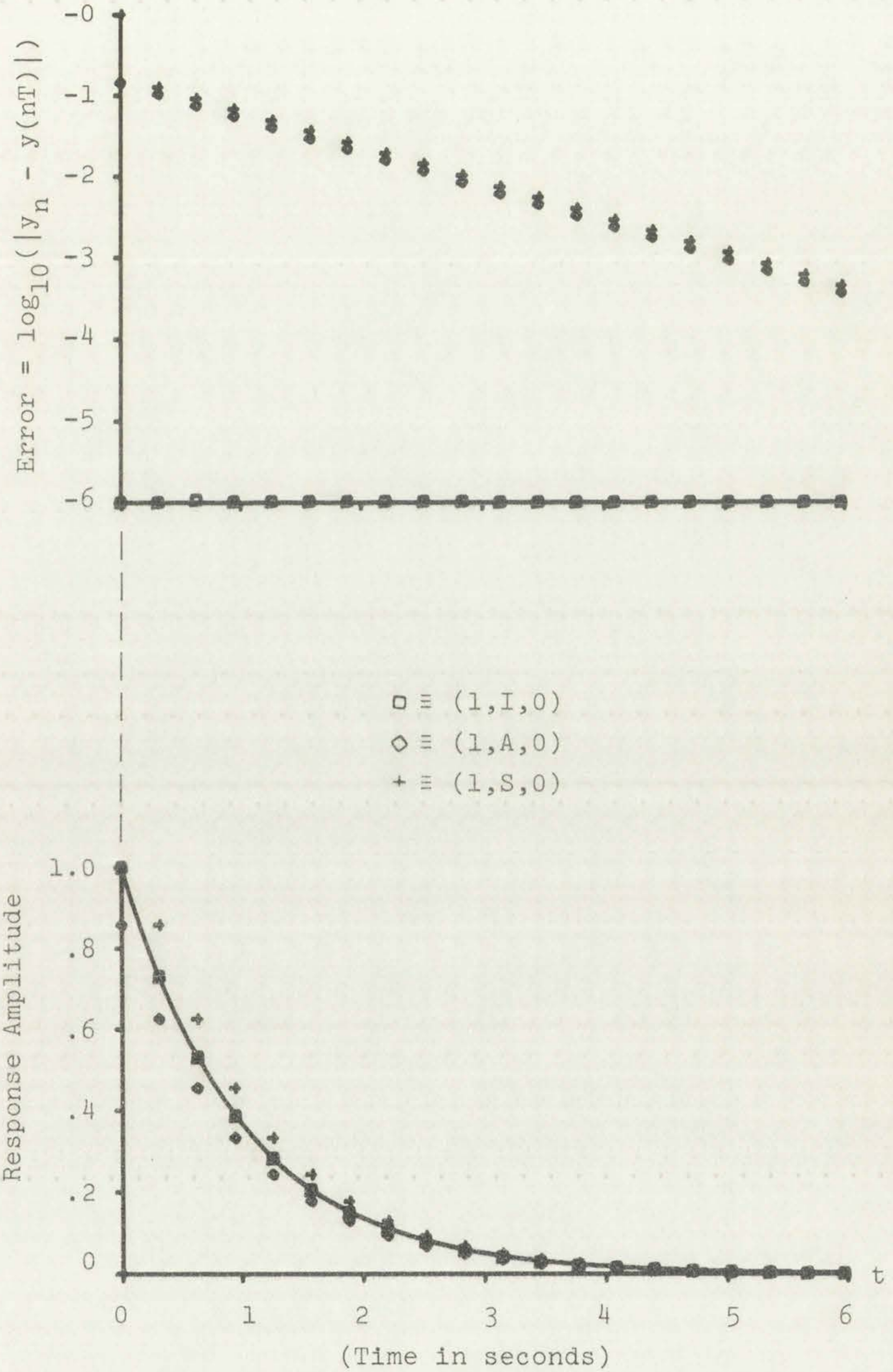


Figure 13. Response and Error of Impulse, Adjusted Impulse and Step Invariant Simulations of First Order System with Impulse Input

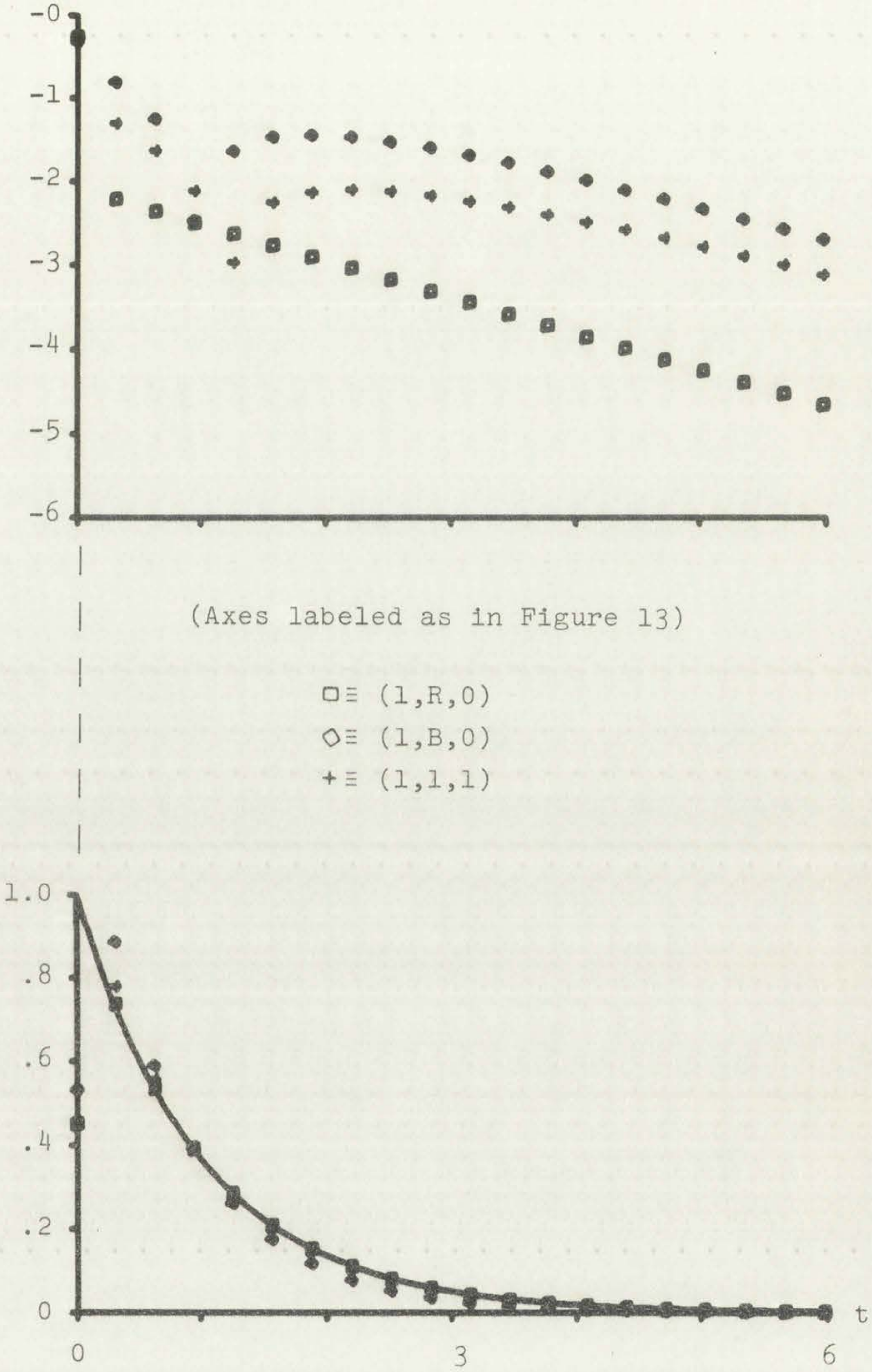
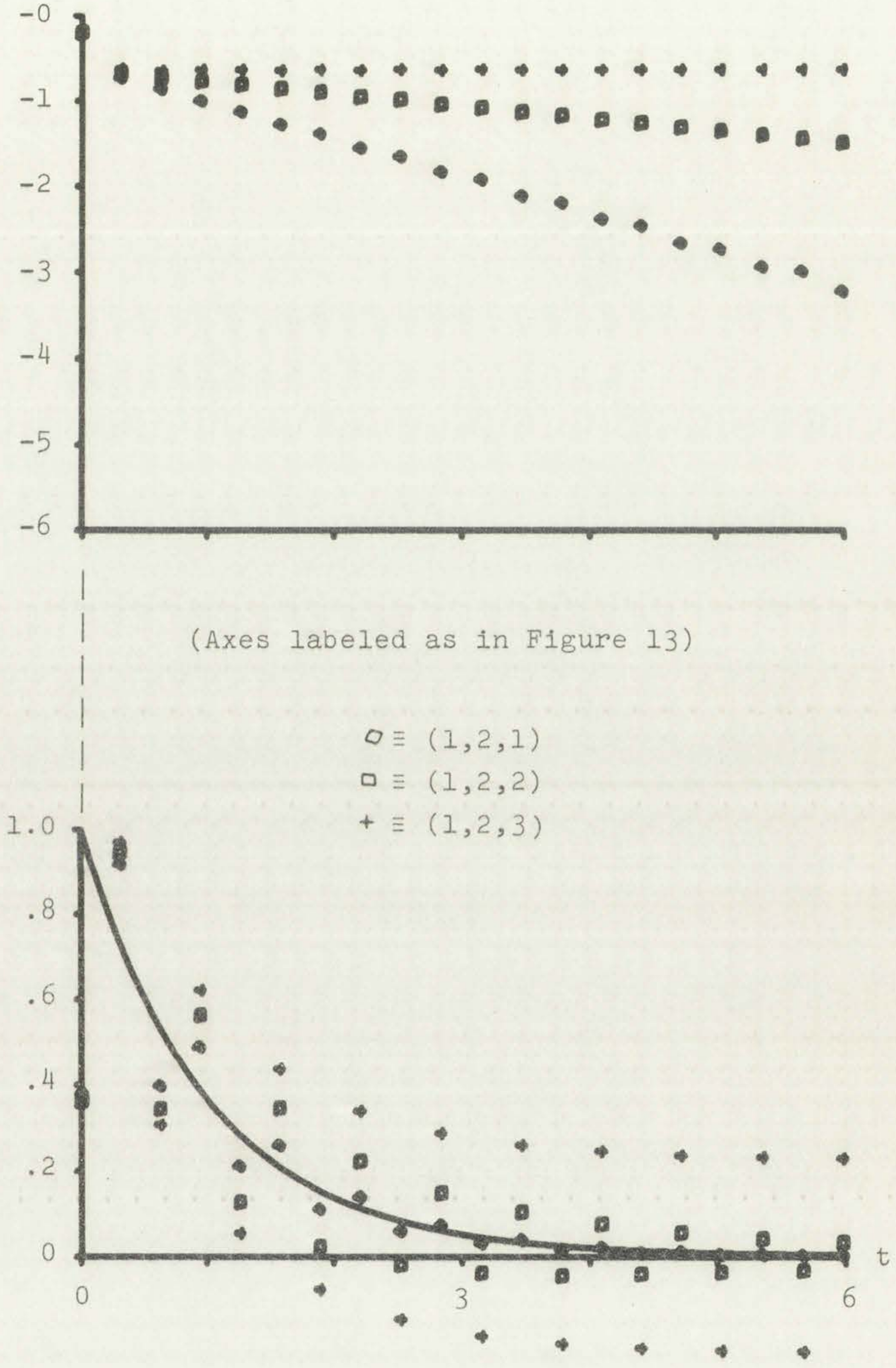


Figure 14. Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Impulse Input



(Axes labeled as in Figure 13)

Figure 15. Response and Error of Three, Second Order Optimized Simulations of First Order System with Impulse Input

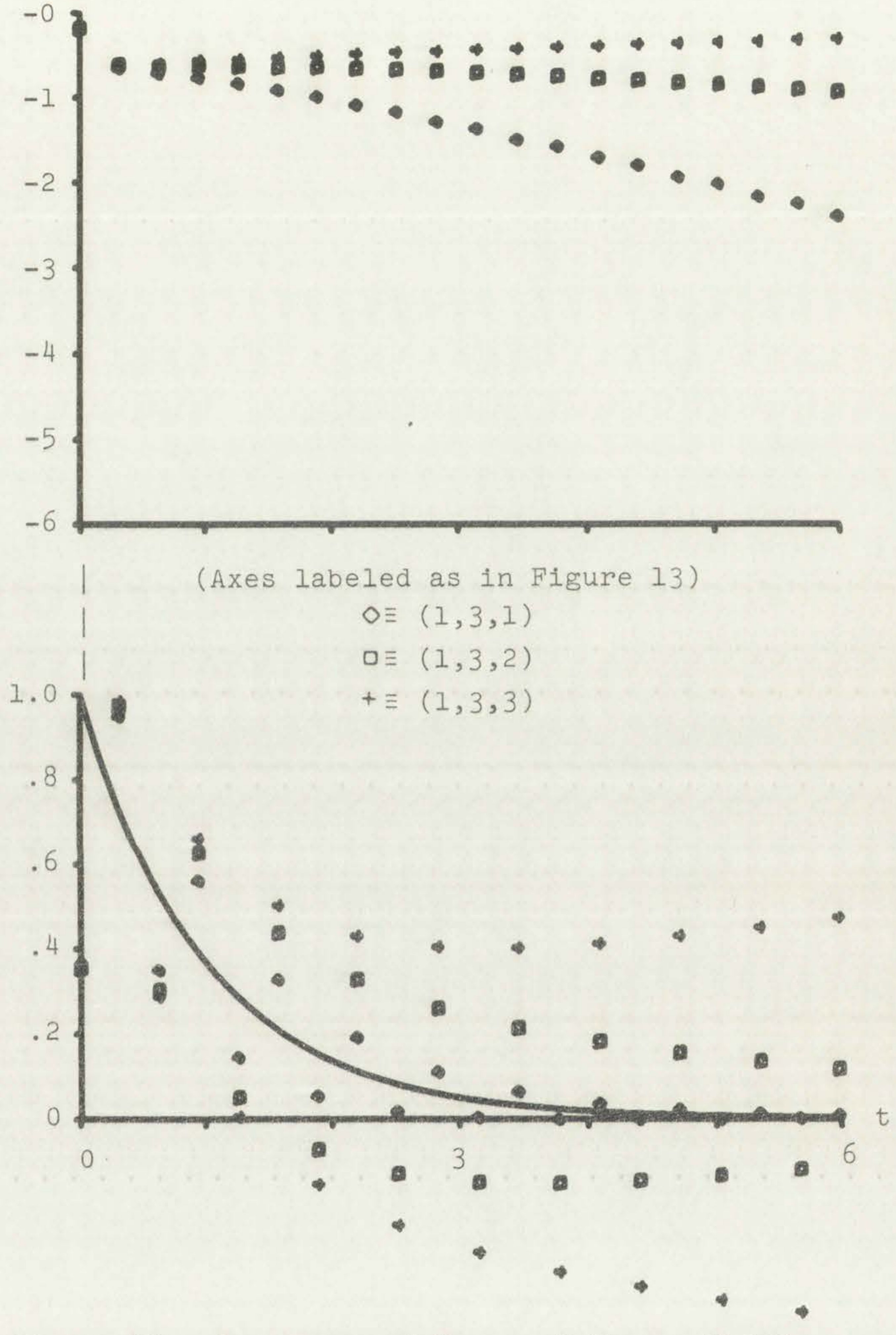


Figure 16. Response and Error of Three, Third Order Optimized Simulations of First Order System with Impulse Input

see that the second order system will have a term of the form

$$\left(\frac{1}{1 + \mathrm{RAD}\ z^{-1}}\right) \tag{412}$$

while the third order system will have a term of the form

$$\frac{1}{(1 + \mathrm{RAD}\ z^{-1})^2} \tag{413}$$

where RAD is the maximum radius allowed for the z-plane poles. Now by representing RAD by the expression

$$\mathrm{RAD} = e^{-\alpha T} , \tag{414}$$

we can solve for α, the reciprocal of the time constant of the impulse response of the simulation system. Thus, for $\mathrm{RAD} = .730403$, $\alpha^{-1} = 1$, the time constant of the continuous system. However, for $\mathrm{RAD} = .9$ we have $\alpha = .336$ and for $\mathrm{RAD} = .999$, $\alpha = .00318$. The time domain equivalent of (412) and (413) is

$$y_n = e^{-\alpha nT} \tag{415}$$

and

$$y_n = nT\ e^{-\alpha nT} , \tag{416}$$

respectively. In order to find the maximum of these expressions we may represent them by the continuous functions they are considered samples of. That is,

$$y_n = e^{-\alpha nT} \Longrightarrow y(t) = e^{-\alpha t} . \tag{417}$$

and

$$y_n = nTe^{-\alpha nT} \Longrightarrow y(t) = te^{-\alpha t} . \tag{418}$$

The maximum of (417) occurs at t = 0 while the maximum of (418) is found below

$$\frac{d}{dt} y(t) = \frac{d}{dt} (te^{-\alpha t}) = e^{-\alpha t} - t\alpha e^{-\alpha t} . \tag{419}$$

The maximum of y(t) occurs where the derivative found above equals zero, that is, where $\alpha t = 1$. Thus, as α becomes small (RAD approaches 1), this term of the response continues to increase for large values of t.

In the particular example shown in Figure 16 for $\alpha = .00318$ the impulse response does not begin its monotonic decrease until $t = \frac{1}{.00318} = 314$ seconds. With T = .314 seconds this implies that the value of the impulse response does not begin to monotonically decrease to zero until the 1000-th sampling interval. In addition, with a time constant of 314 seconds this final system decay requires many thousands of sample times. In fact to reach about 1% of its maximum value the expression $y(t) = te^{-.00318t}$ takes over 2400 seconds or about 8000 samples as is shown below.

$$y(314) = 314e^{-1} = 116 \tag{420}$$

$$y(2400) = 2400e^{-.00318 \times 2400} \cong 1.2 \ . \qquad (421)$$

Thus, although this third order system of (373) is stable by definition the transient response cannot be neglected for the first 8000 samples. Using such a digital system to simulate a continuous system $\left[H(s) = \frac{1}{s+1}\right]$ whose transient response is reduced to less than 1% of its maximum value in less than 15 sample periods is of little value in most cases. The second order system contains an impulse response term of the form $y(t) = e^{-.00318t}$ with RAD = .999, and decays to 1% of its maximum value in approximately 1450 seconds or about 4600 sample periods. The fourth order simulation (not plotted) with RAD = .999 contains an impulse response term of the form

$$y(t) = t^2e^{-.00318t} \qquad (422)$$

whose derivative with respect to time is

$$\frac{dy(t)}{dt} = (2t - .00318t^2)e^{-.00318t} \qquad (423)$$

and correspondingly whose maximum value occurs for

$$2t = .00318t^2 \qquad (424)$$

or

$$t = 2/.00318 \cong 628 \ . \qquad (425)$$

Comparing (425) with (419) we see that the maximum value here takes twice as long to reach and worse, the maximum value is

$$y(628) = (628)^2 e^{-2} \cong 5400 . \qquad (426)$$

Thus, this fourth order system is nearly unusable with RAD = .999 as the polynomial P(z) of the term $\dfrac{P(z)}{(1 - RADz^{-1})^3}$ in the partial fraction expansion of the fourth order system given in (374) cannot in general be assumed to be small enough to make this term negligible. However, by limiting RAD to be less than or equal to $e^{-\alpha T}$ where α^{-1} is the time constant of the continuous system we have forced the transient response of the first and second order digital simulation (363) and (364) to decay at least as rapidly as the transient response of the continuous system. The third and fourth order systems (365) and (366) contain a double and triple pole, respectively, at $z = -e^{-T}$. These second and third order poles lead to impulse response terms of the form

$$y(t) = te^{-t} \qquad (427)$$

and

$$y(t) = t^2 e^{-t} , \qquad (428)$$

respectively. The maximum of (427) occurs at t = 1 and of (428) at t = 2 as follows from the development leading to (419) and (425). These maximums are, for (427)

$$y(1) = .368 \qquad (429)$$

and for (428)

$$y(2) = .545 \ . \tag{430}$$

The time required for (427) to reach 1% of the value of (429), .368 is about 7.7 seconds or 24 sample intervals as

$$y(7.7) = (7.7)e^{-7.7} = .0033 \ . \tag{431}$$

Similarly, the time required for the expression of (428) to reach 1% of the value of (430), .545, is about 10 seconds or 32 sample intervals as

$$y(10) = (10)^2 e^{-10} = .00454 \ . \tag{432}$$

Thus, by limiting the maximum radius of the poles of the digital simulation we have effectively limited the duration of the discrete system impulse response to approximately that of the continuous system.

Figure 17 shows the unit step input response of the impulse invariant (1,I,0) adjusted impulse invariant (1,A,0), and step invariant (1,S,0) simulations of $H(s) = \frac{1}{s+1}$. Here the disadvantage of the impulse invariant simulation is clearly shown, that is, its d.c. gain is too large. Note that this figure is plotted using Stearns' definition of the unit impulse; $x_o = \frac{1}{T}$, $x_n = 0$, $n \neq 0$. From (355) the impulse invariant simulation (1,I,0) is $\tilde{H}(z) = \frac{T}{1 - e^{-T}z^{-1}}$. This simulation transfer function approaches the correct dc gain as T becomes small, as shown below.

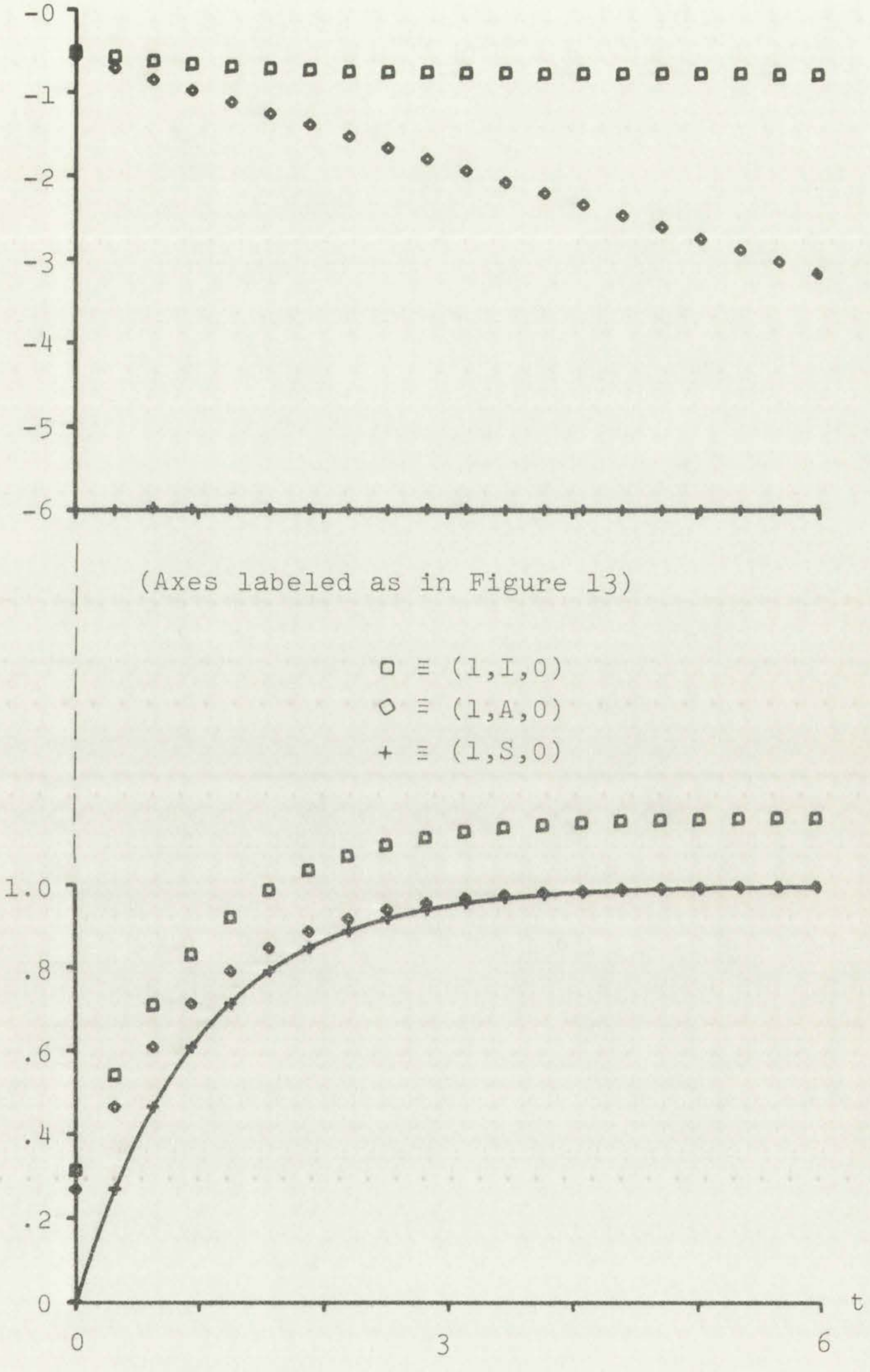

(Axes labeled as in Figure 13)

Figure 17. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with Step Input

$$(1 - e^{-T}) = 1 - \left(1 - T + \frac{T^2}{2!} - \frac{T^3}{3!} \ldots\right) = T - \frac{T^2}{2!} + \frac{T^3}{3!} + \ldots \tag{433}$$

or

$$1 - e^{-T} \cong T \tag{434}$$

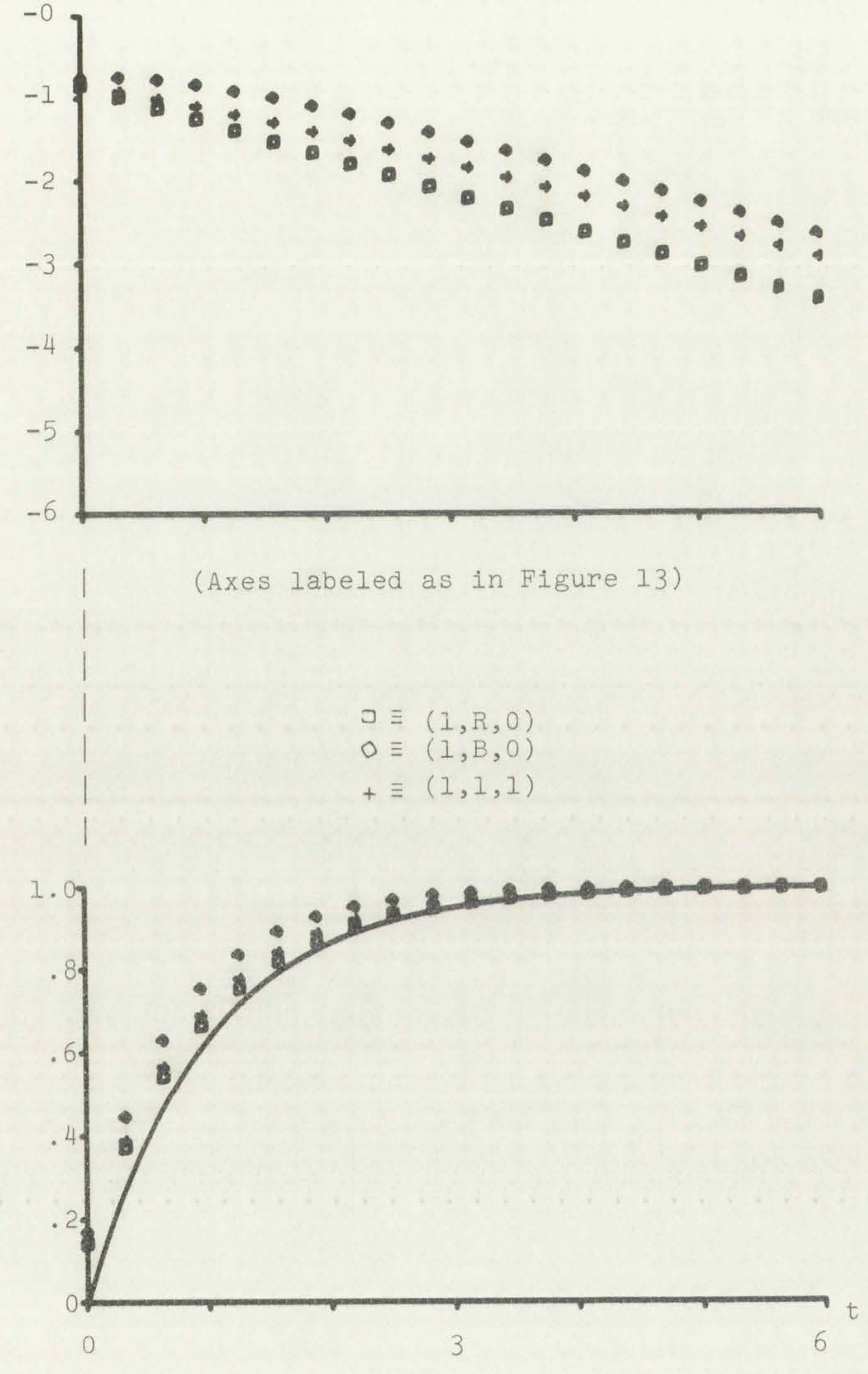
for small T.

If the usual definition of the unit impulse was used here, $x_o = 1$, $x_n = 0$ for $n \neq 0$, then the impulse invariant simulation becomes $\tilde{H}(z) = \frac{z}{z - e^{-T}}$ for $H(s) = \frac{1}{s + 1}$ and the d.c. gain of the digital system grows without limit as T becomes small.

Figure 17 illustrates the advantage of choosing the d.c. gain adjusted impulse invariant simulation. That is, the correct d.c. gain is obtained irrespective of T at some small cost in the impulse response as shown in Figure 13.

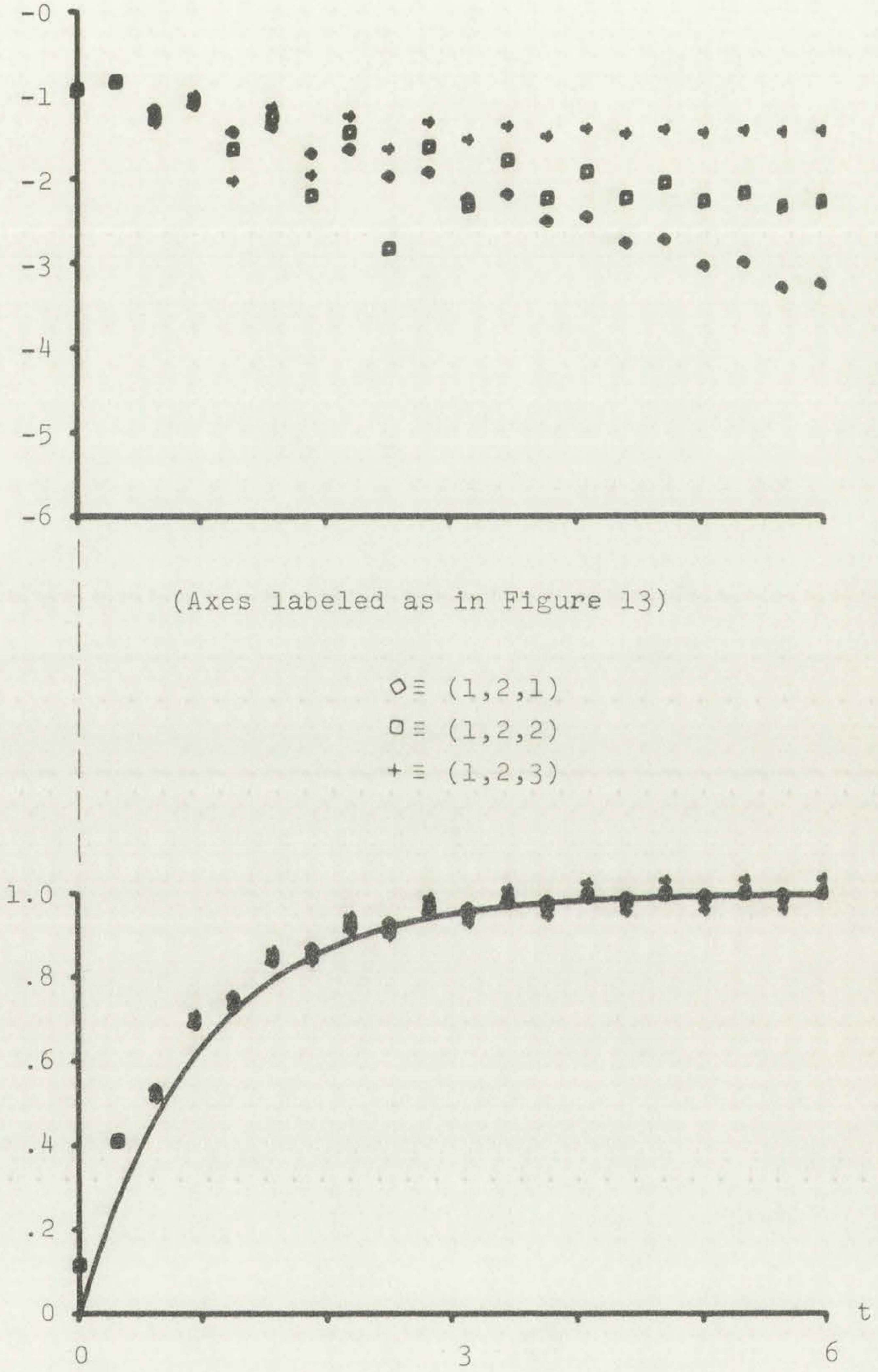
Figure 18 shows the step response of the bilinear approximation (1,B,0), ramp invariant (1,R,0), and first order optimized (1,1,1) simulations. All of these systems have the correct d.c. gain and all the transients are decaying at an acceptable rate.

Figure 19 is a plot of the step response of the second order optimized simulations: (1,2,1), (1,2,2), and (1,2,3). The response of all three systems is shown to oscillate as is expected with the poles on the negative real axis; however, with the pole radius restricted to e^{-T} this error is no worse than that obtained from the first order systems of Figure 18 and is decaying at the same rate. The rate of error decay



(Axes labeled as in Figure 13)

Figure 18. Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Step Input



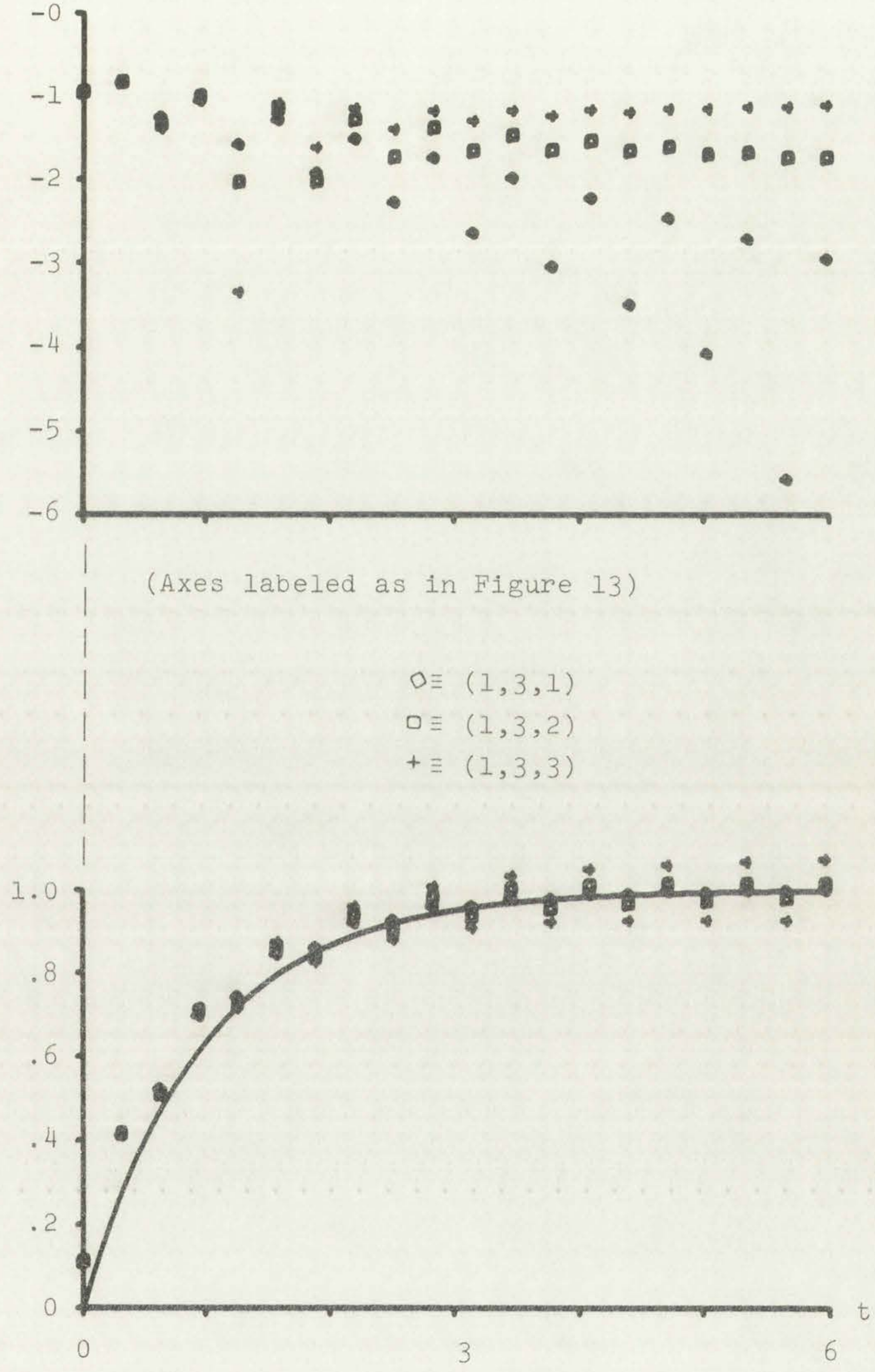
(Axes labeled as in Figure 13)

Figure 19. Response and Error of Three, Second Order Optimized Simulations of First Order System with Step Input

becomes very slow as the poles are allowed to approach the unit circle.

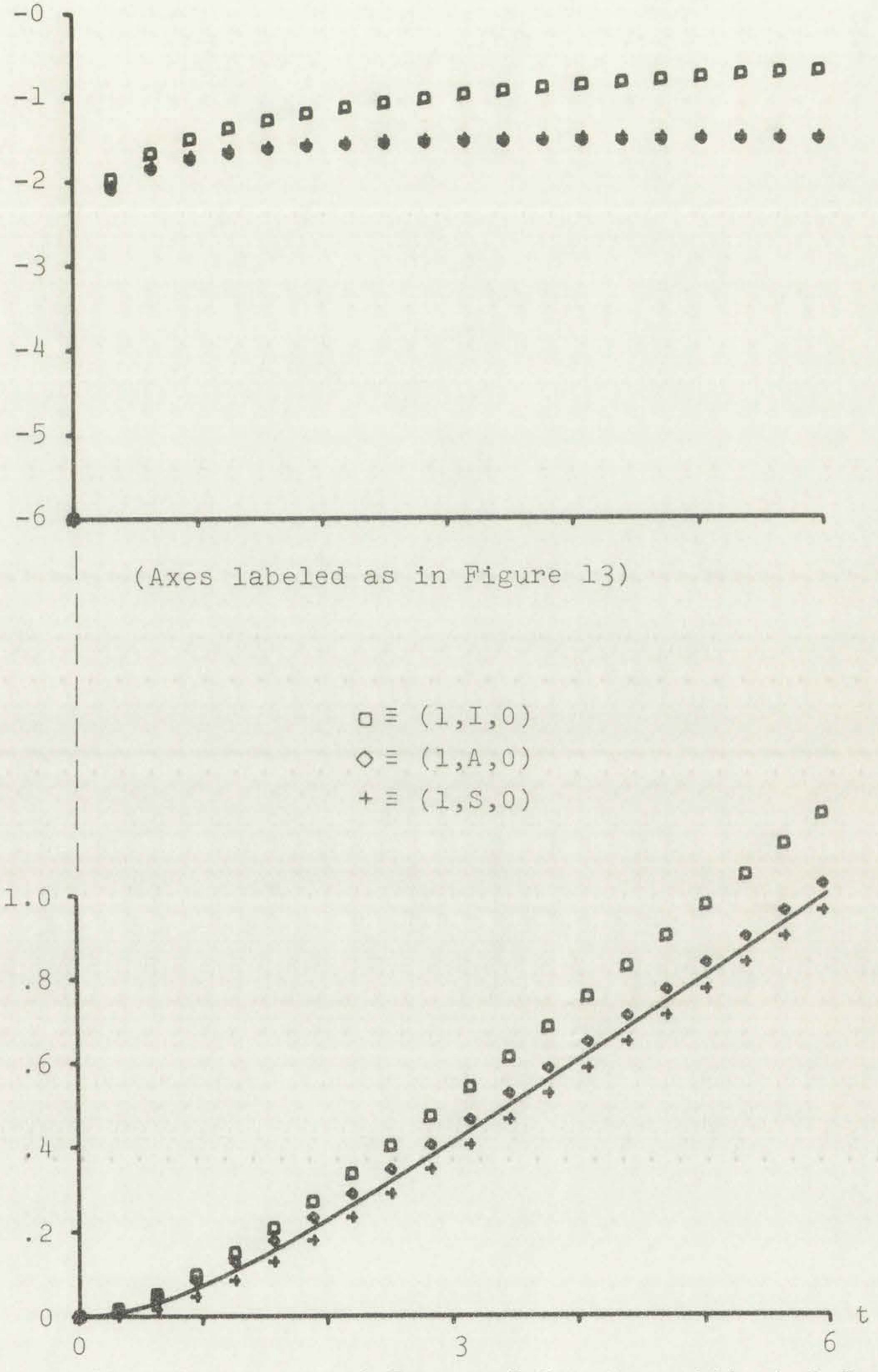
Figure 20 represents the step response of the third order systems: (1,3,1), (1,3,2), and (1,3,3). Again with the poles restricted to a radius of e^{-T} the error is comparable to the first order systems; however, with the poles allowed to reach -.999 the error is rapidly increasing. At the last sample shown here, the twentieth, the error is about 7%; after 100 sample periods it has increased to 22% and is still increasing as expected from the discussion of Figure 16. The responses of the fourth order systems: (1,4,1), (1,4,2), and (1,4,3) are not shown as the errors of the systems with poles at -.9 and -.999 increase too rapidly for these plots. However, with the poles restricted to e^{-T} the error of the fourth order system has decreased to 10^{-5} in about 49 sample intervals compared with 40, 34, and 37 intervals for the comparable first, second, and third order systems, respectively.

The response of the impulse (1,I,0), adjusted impulse (1,A,0), and step invariant (1,S,0) simulations to a ramp input are shown in Figure 21. Equations (356) and (359) show the adjusted impulse and step invariant forms to be identical except for a one-step delay in (359) from (356). This delay is apparent here. The errors for the ramp input do not approach zero; however, since the response continues to increase and the errors are essentially constant, the relative error for those simulations with the correct d.c. gain do approach zero. The



(Axes labeled as in Figure 13)

Figure 20. Response and Error of Three, Third Order Optimized Simulations of First Order System with Step Input



(Axes labeled as in Figure 13)

Figure 21. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with Ramp Input

error of the impulse invariant simulation continues to increase indefinitely.

Figure 22 shows the response of the ramp invariant (1,R,0), bilinear approximation (1,B,0), and optimized first order (1,1,1) simulation. Neglecting the ramp invariant system, which is an exact simulation by definition, the error of the optimized first order system is the least of those simulations considered here.

The response of the impulse (1,I,0), adjusted impulse (1,A,0), and step invariant (1,S,0) simulations to sine wave inputs of sin 3t and sin 7t are shown in Figures 23 and 24, respectively. Here again we see the one step delay of the step invariant simulation compared to the adjusted impulse invariant. It is apparent that the time domain error, $\left|y_n - y(nT)\right|$, plotted in these figures, is inappropriate for a periodic input as phase is not considered. By visually fitting a sinusoid to the discrete systems output shown on the lower plot of these two figures, it is obvious that the magnitude and frequency of this curve is approximately correct. However, due to the phase difference the error plotted is meaningless after the transient response dies out. Thus, for periodic inputs the frequency domain error analysis is superior.

Figures 25 and 26 illustrate the ramp invariant (1,R,0), bilinear approximation (1,B,0), and optimized first order (1,1,1) simulation responses to sin 3t and sin 7t inputs, respectively. In general the errors here are considerably less than those shown in the two preceding figures, 23 and 24 as

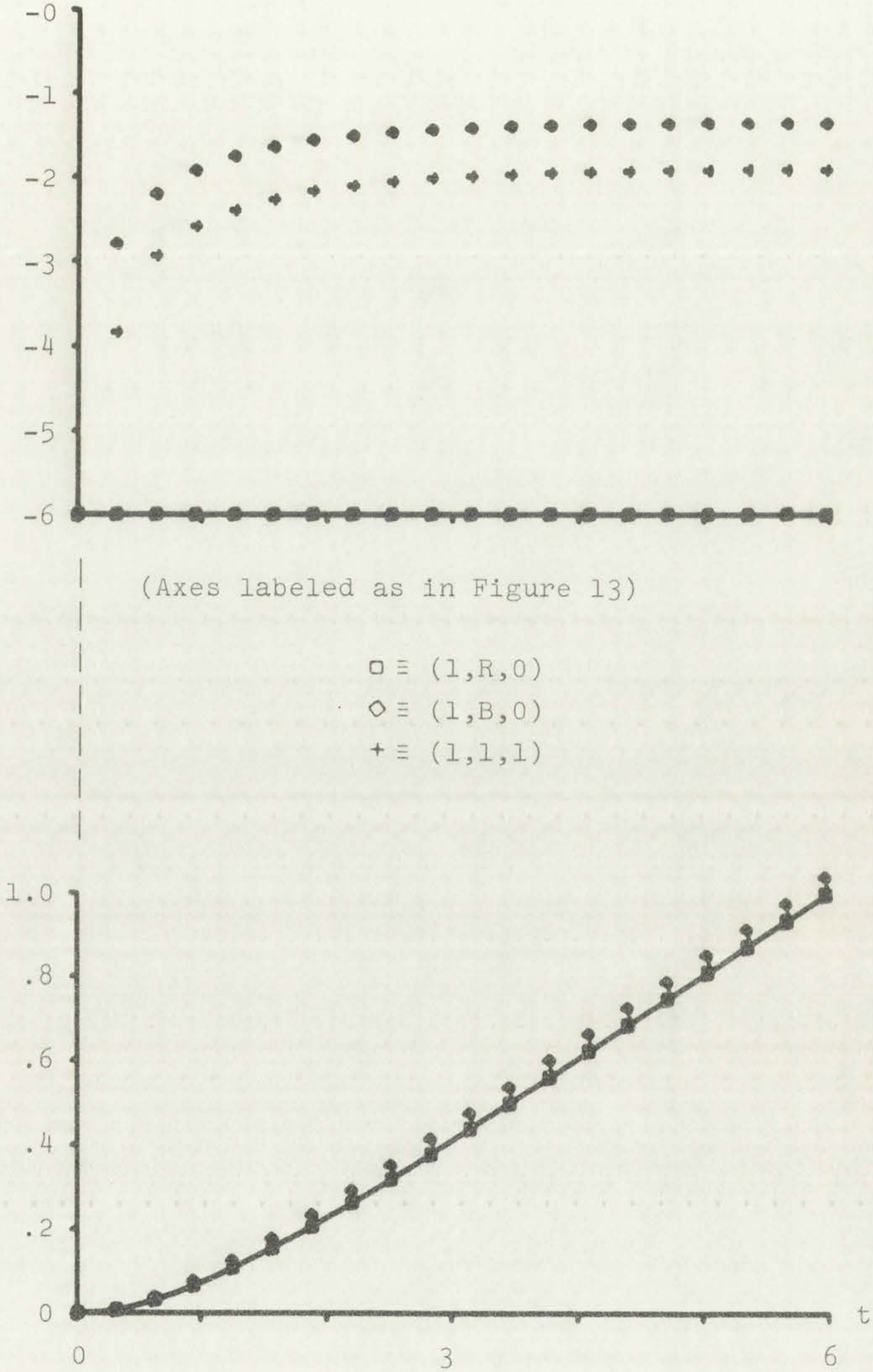
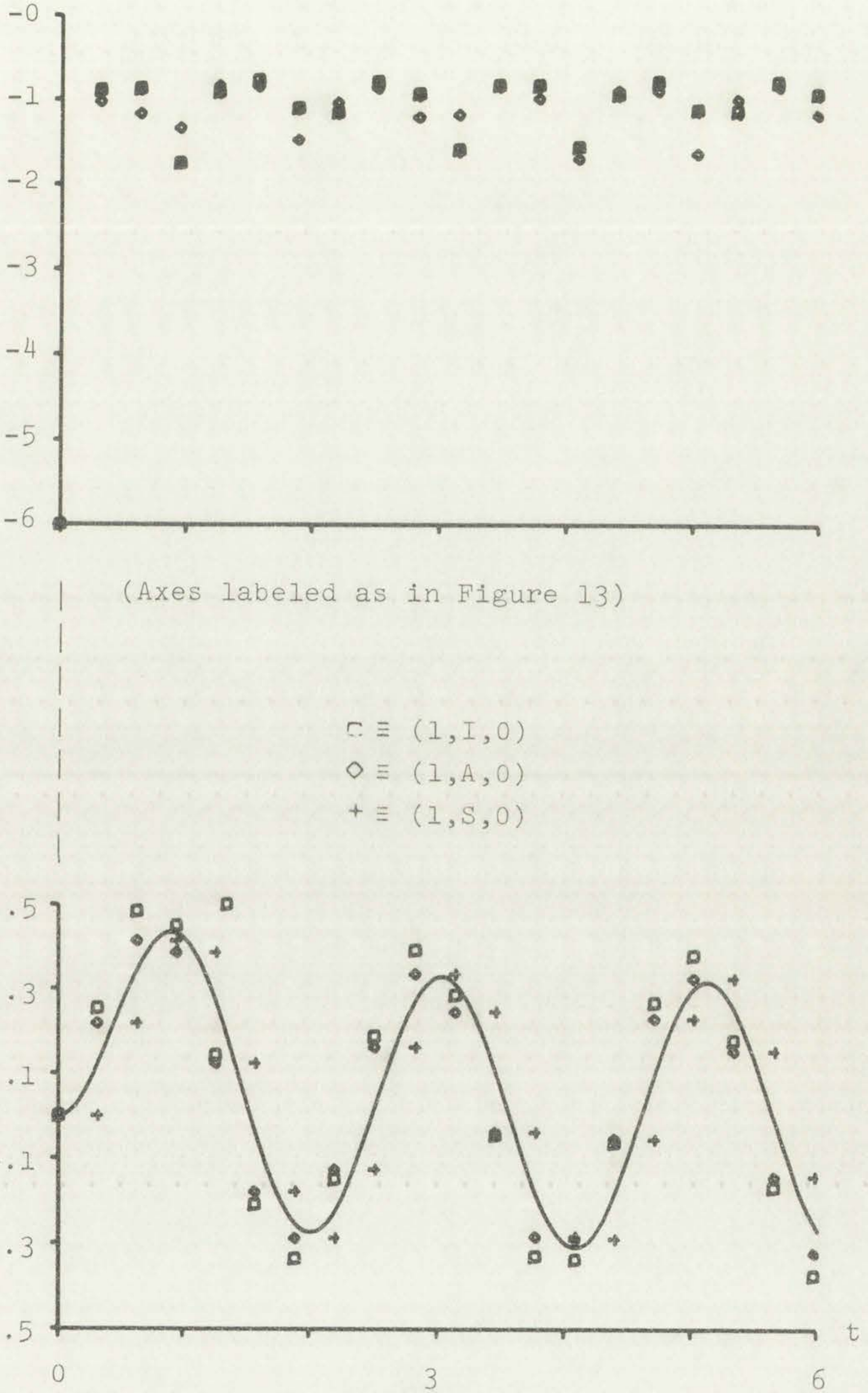
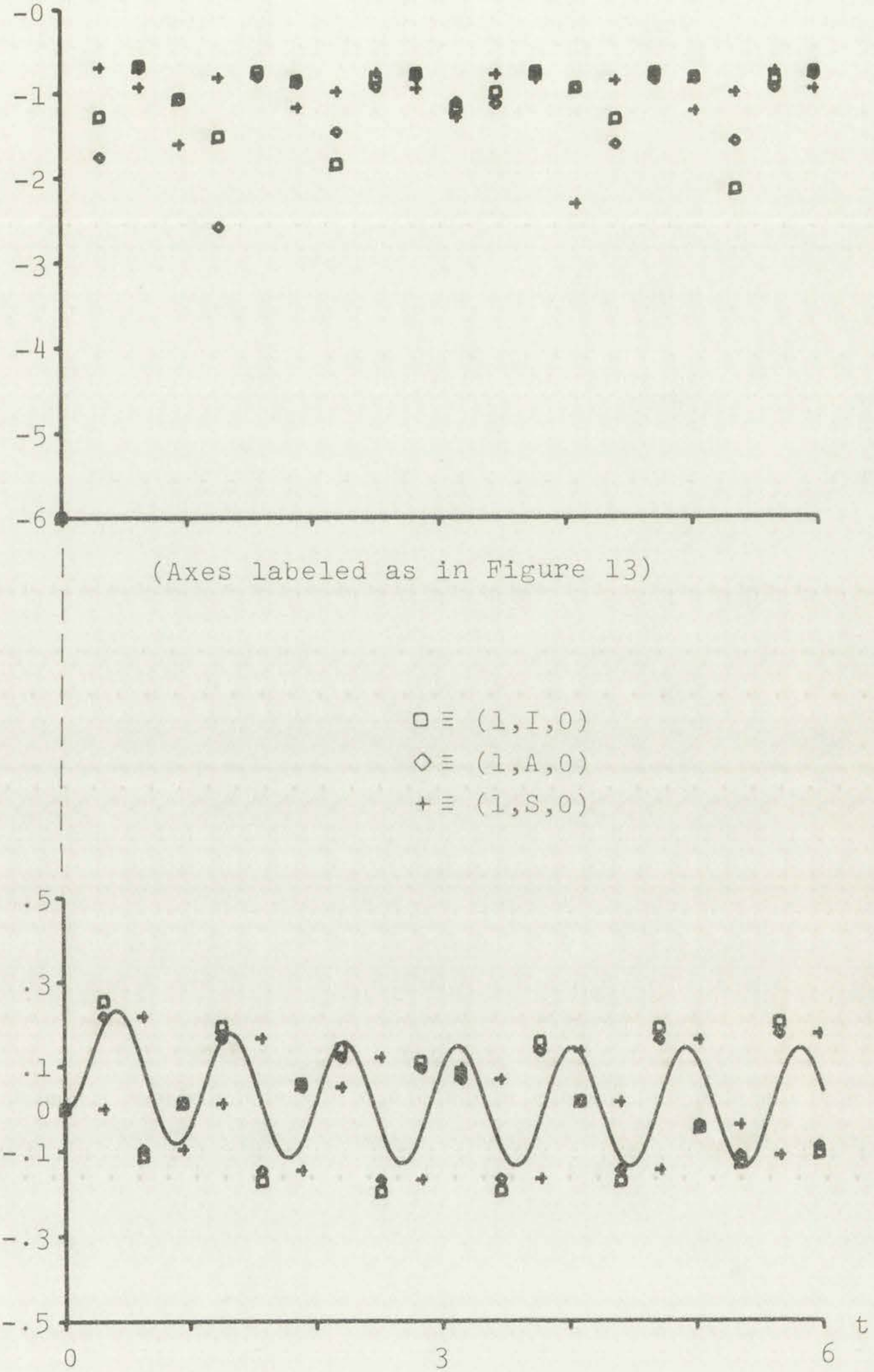


Figure 22. Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with Ramp Input



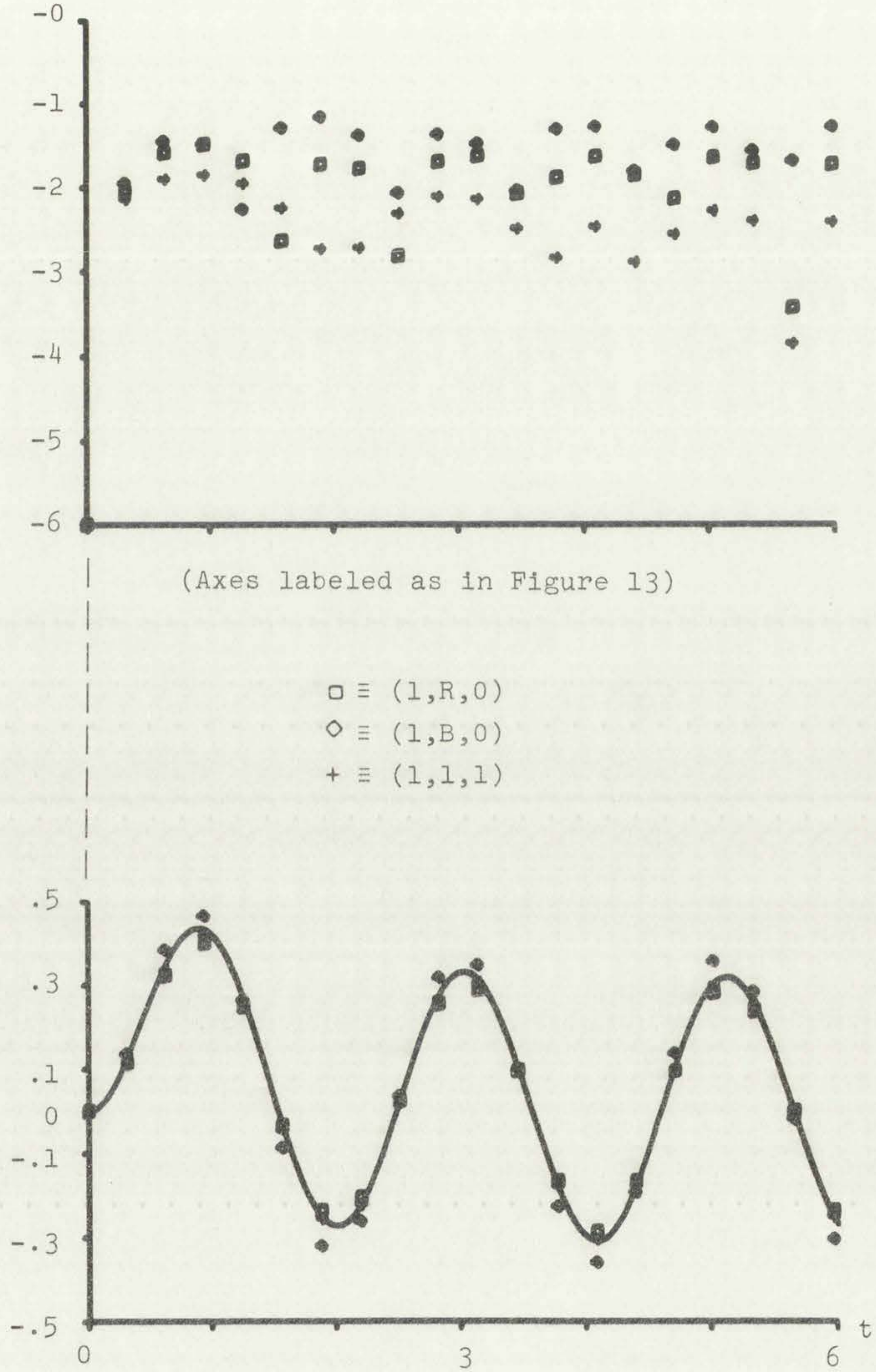
(Axes labeled as in Figure 13)

Figure 23. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with sin 3t Input



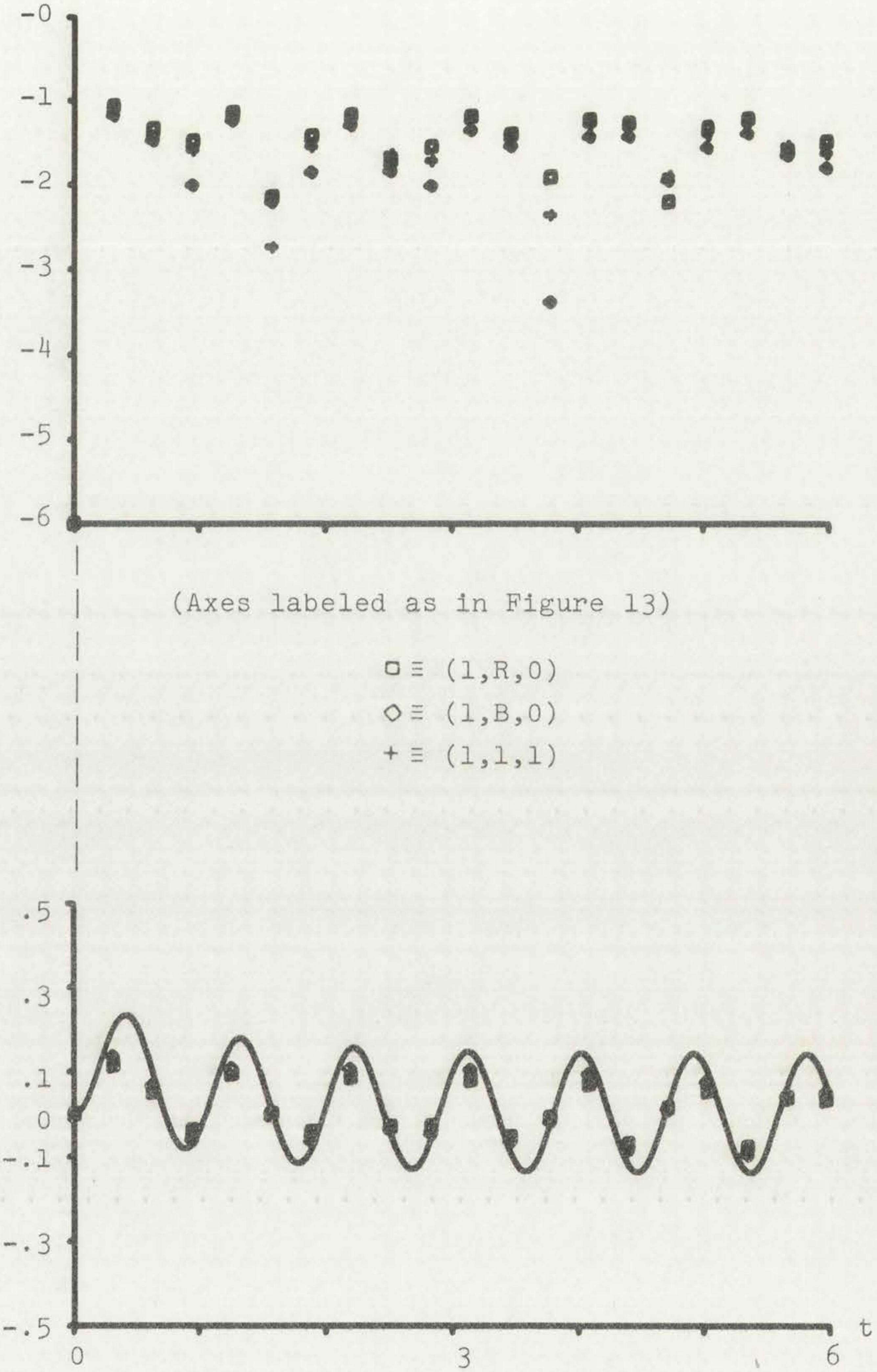
(Axes labeled as in Figure 13)

Figure 24. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of First Order System with sin 7t Input



(Axes labeled as in Figure 13)

Figure 25. Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with sin 3t Input



(Axes labeled as in Figure 13)

Figure 26. Response and Error of Ramp Invariant, Bilinear Approximation, and First Order Optimized Simulations of First Order System with sin 7t Input

would be expected from the frequency domain curves of Figures 7 and 8.

7.4 Error Comparisons of Simulations of the Second Order System

Similar comparisons were made for the simulations of the second order system and will be presented below.

7.4.1 Frequency Domain Errors

Figure 27 compares the frequency domain errors of the impulse (2,I,0), adjusted impulse (2,A,0), and step invariant (2,S,0) simulations. The most noticeable difference of these curves from those of Figure 7 for the first order system is that the frequency domain error shown here for the impulse and adjusted impulse invariant simulations is nearly an order of magnitude less. Another noticeable but less desirable difference is that the step invariant simulation error is much worse here than the others over most of the frequency range while for the first order systems the errors of the three discrete systems were nearly the same.

The comparison of the ramp invariant (2,R,0), bilinear approximation (2,B,0), and second order optimized (2,2,1) simulations of Figure 28 is more interesting. Here the optimized system is clearly superior to either of the others over the entire frequency range of comparison, $0 \leq \omega \leq \frac{5\pi}{7T}$ (except at $\omega = \pi/2T$). The margin of superiority is significant. The error is approximately an order of magnitude less than the ramp invariant simulation over 50% of the interval and obviously

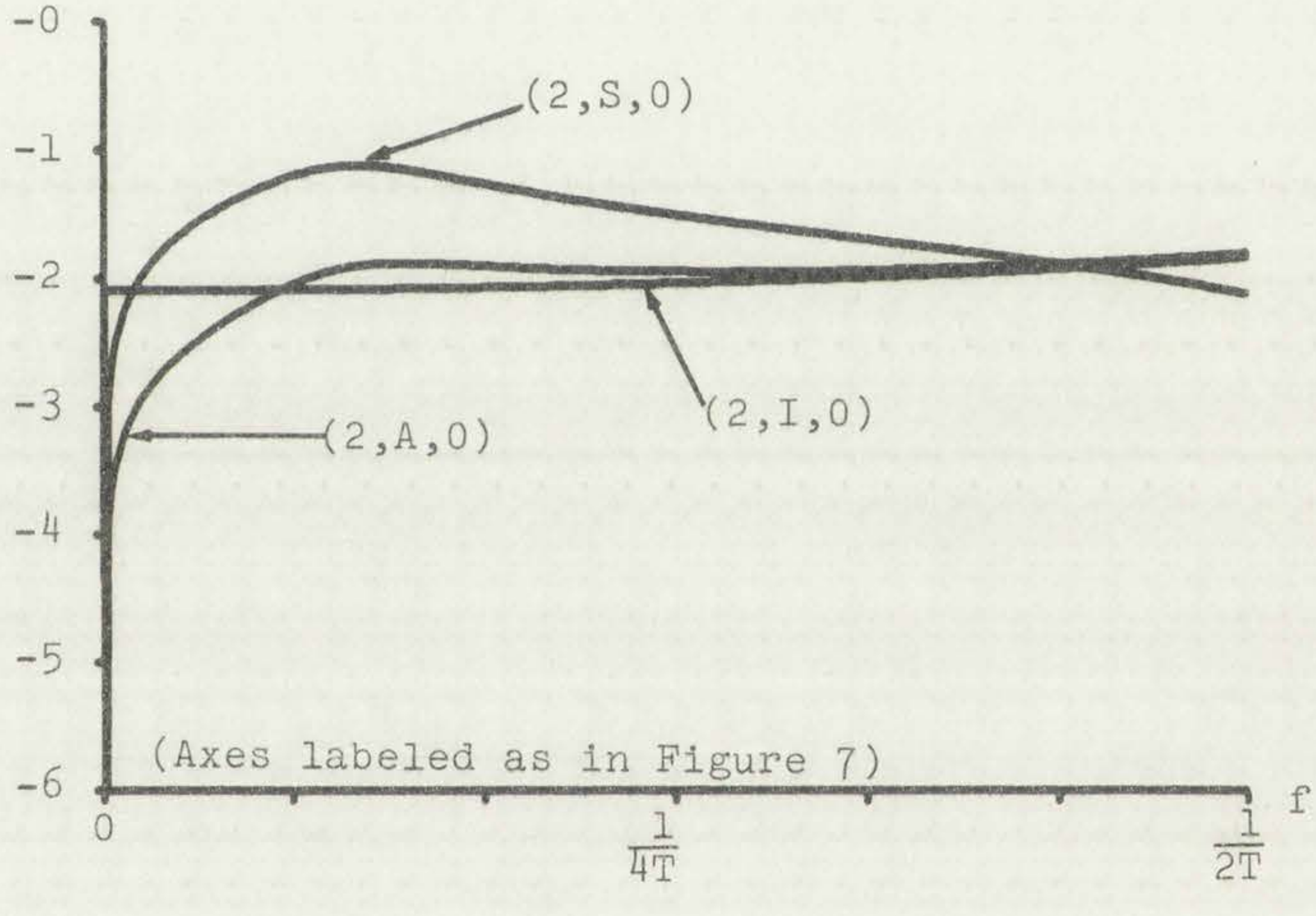


Figure 27. Frequency Domain Error of Impulse, Adjusted Impulse and Step Invariant Simulations of Second Order System

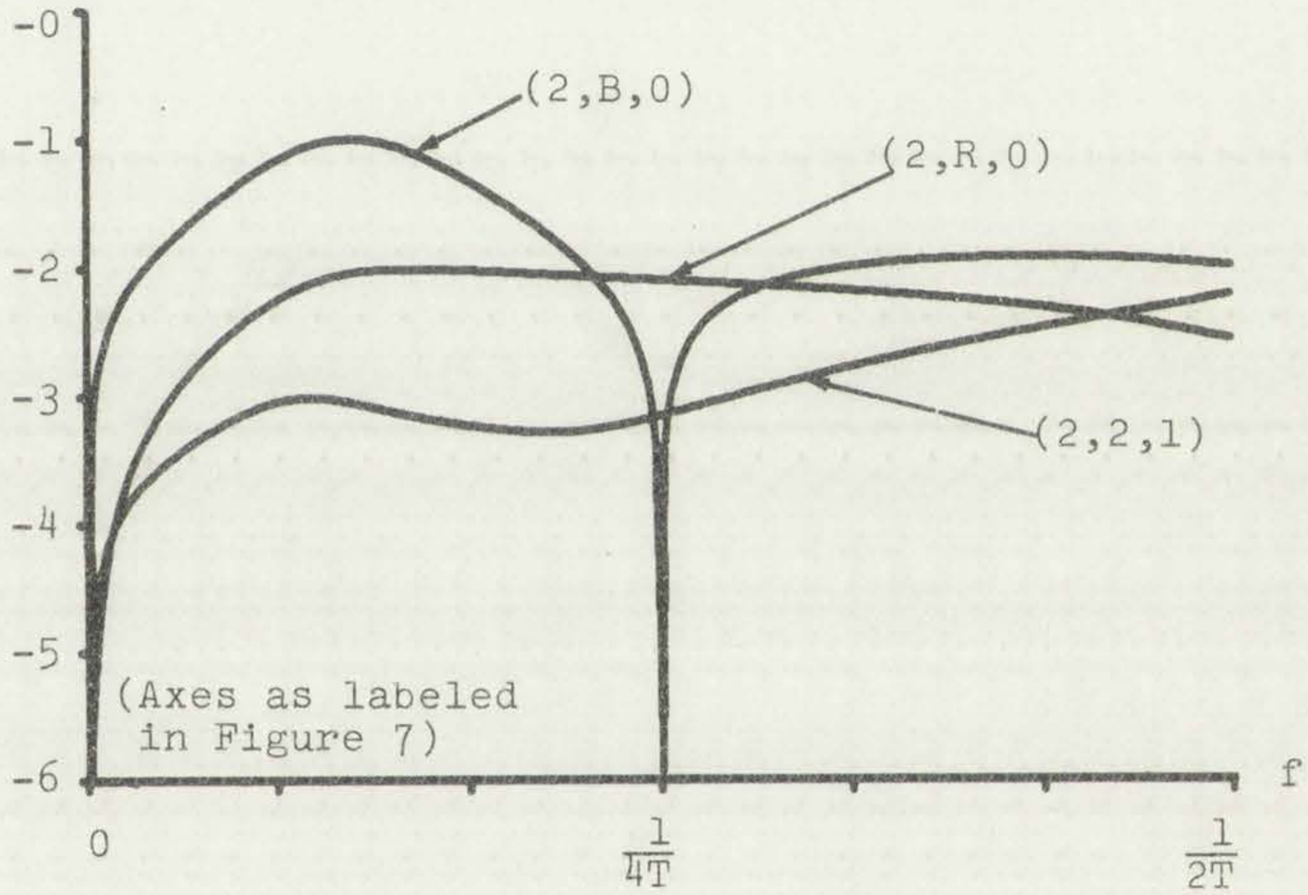


Figure 28. Frequency Domain Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System

a much larger margin over the bilinear approximation. It is also apparent that here the ramp invariant simulation is not significantly better, in the frequency domain, than are the impulse or adjusted impulse invariant simulations. This is in sharp contrast to the first order system simulation errors of Figures 7 and 8. There the ramp invariant error was much less than that of the other invariant solutions.

The three curves of Figure 29 represent the frequency domain errors of the third order optimized digital systems (2,3,1), (2,3,2), and (2,3,3). Similarly, Figure 30 represents the fourth order simulation errors of (2,4,1), (2,4,2), and (2,4,3). Particularly in Figure 29 but also in 30, it is apparent that little is lost in terms of the frequency domain response by restricting the poles to a maximum radius of e^{-T}.

Finally, the errors of the second (2,2,1), third (2,3,1), and fourth (2,4,1) order systems, all with the poles restricted to a maximum radius of e^{-T}, are compared in Figure 31. Here we see little advantage in going from a second to a third order system for the error measure used here; however, the fourth order system does show significant improvement over that of second order. It is interesting to compare these curves to those of Figure 12, the corresponding comparisons for the simulations of the first order system. In general, the errors shown in Figure 31 for the simulation of $H(s) = \frac{1}{s^2 + 2s + 5}$ are an order of magnitude less than those of Figure 12, just as was noted in comparing Figures 7 and 27. Thus, it appears that simulations of second order continuous systems are

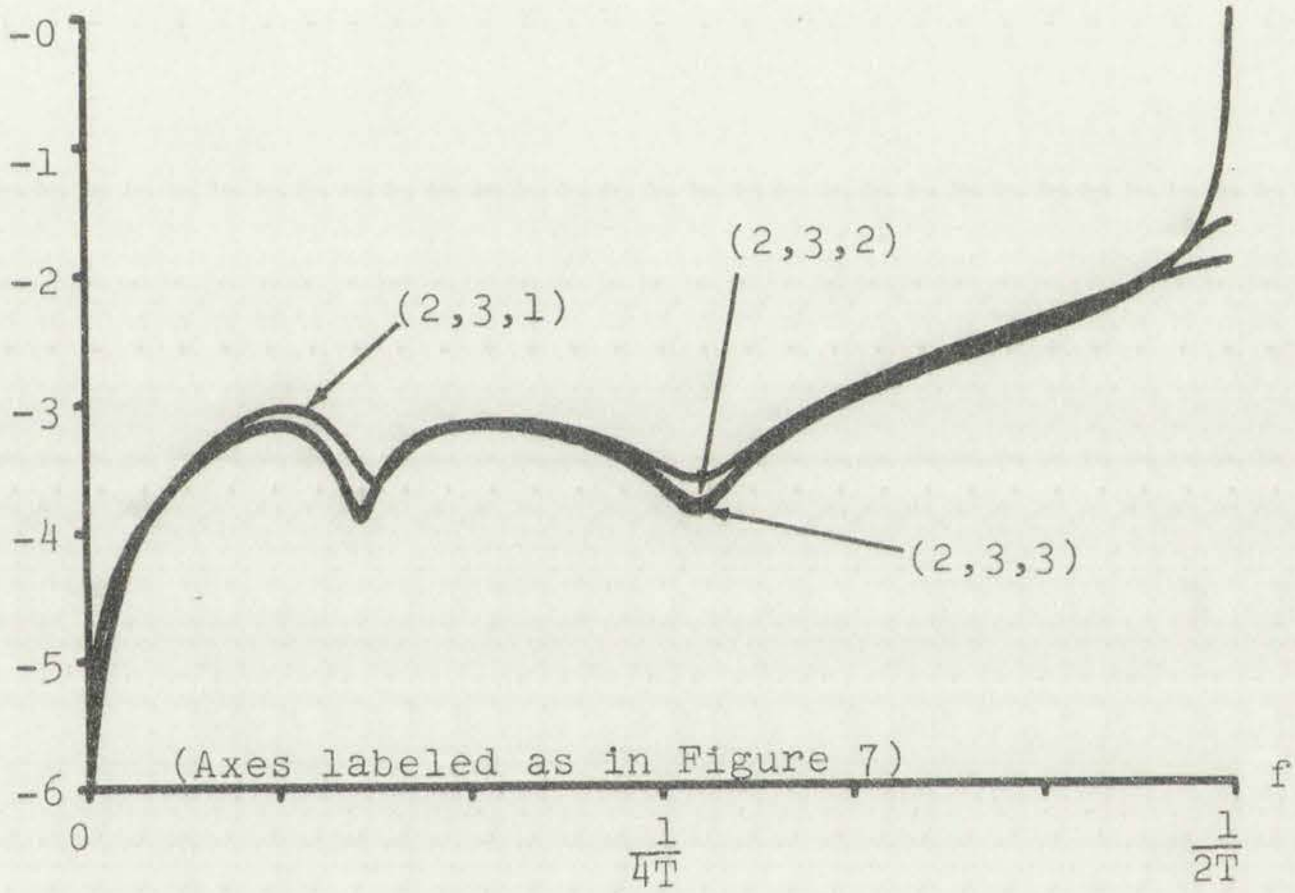


Figure 29. Frequency Domain Error of Three, Third Order Optimized Simulations of Second Order System

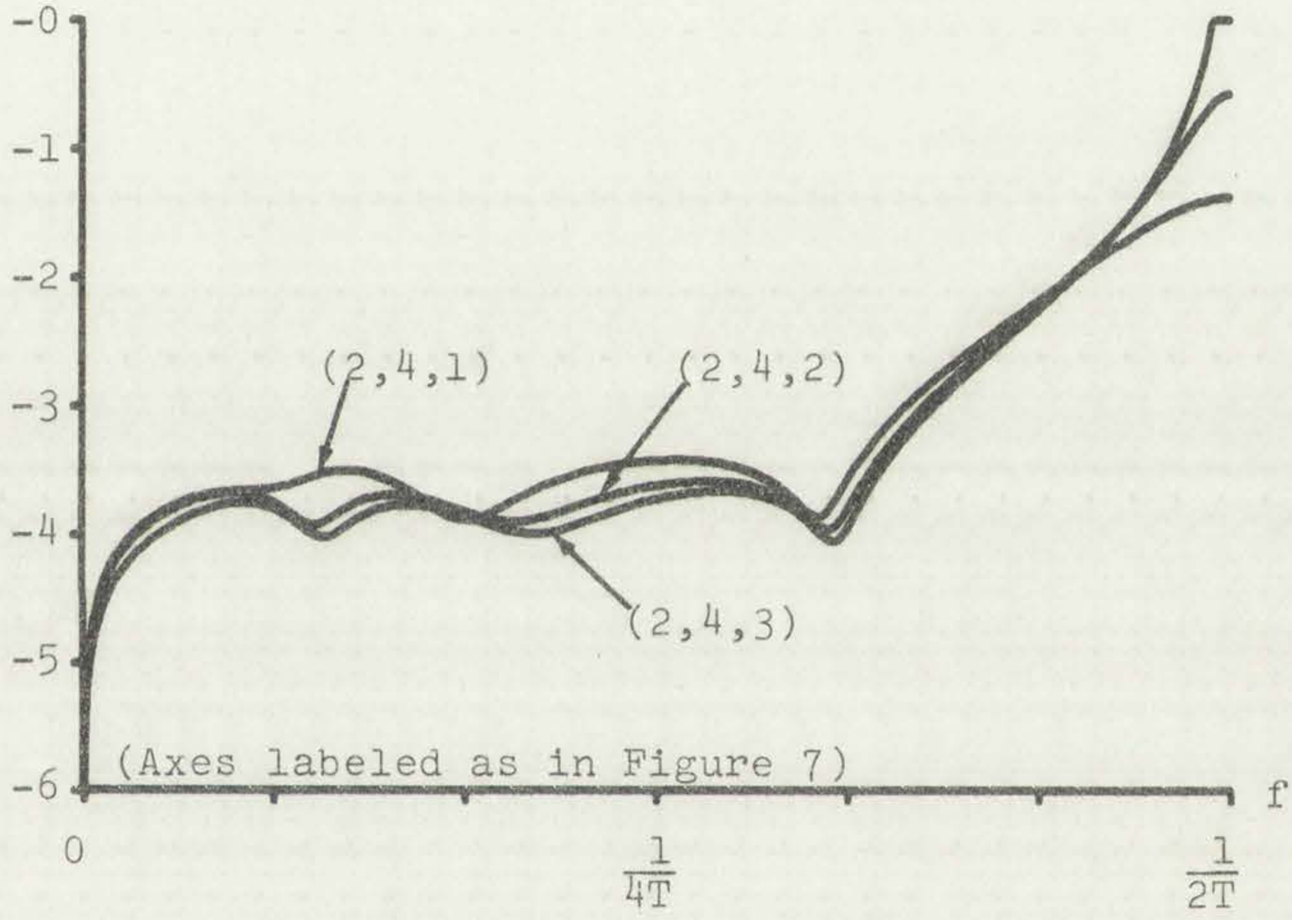


Figure 30. Frequency Domain Error of Three, Fourth Order Optimized Simulations of Second Order System

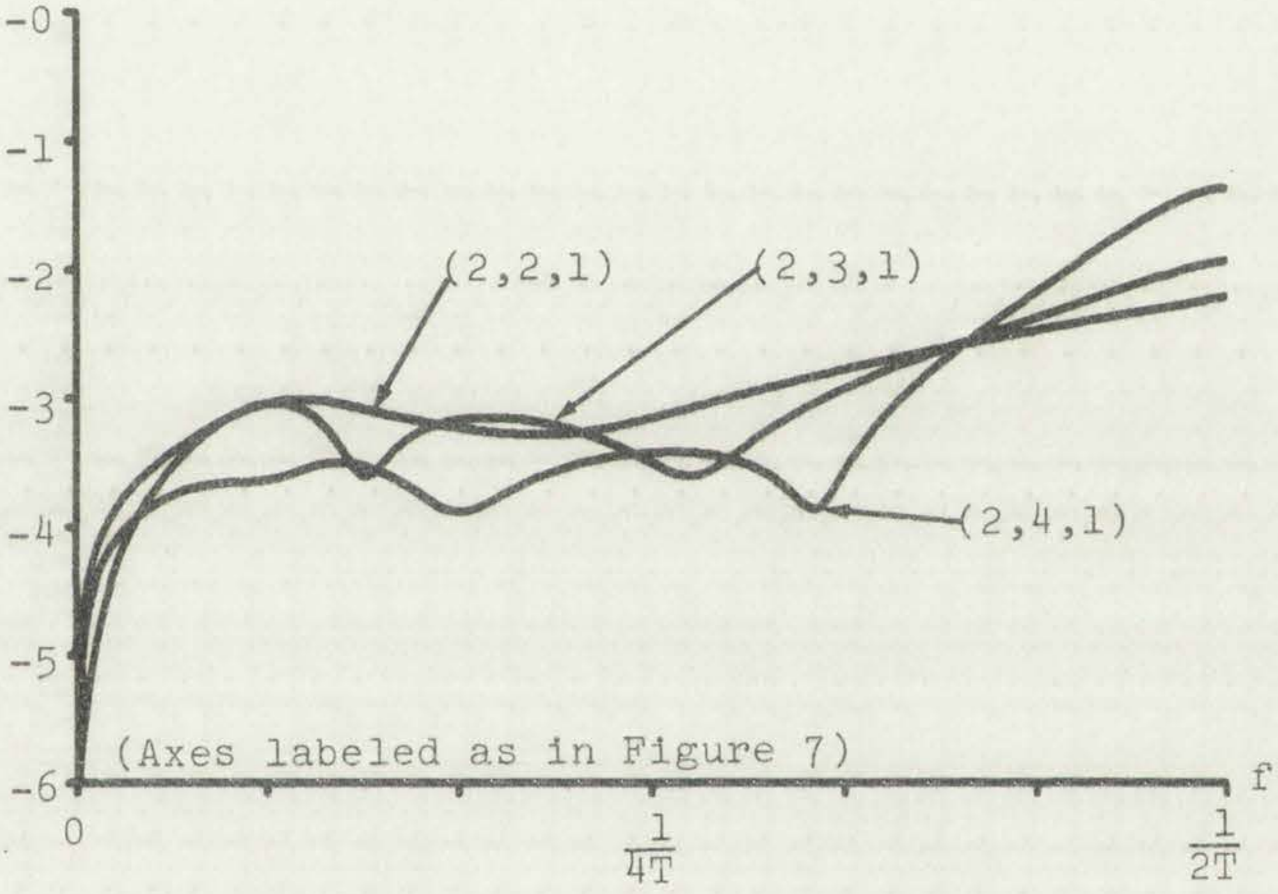


Figure 31. Frequency Domain Error of Second, Third, and Fourth Order Optimized Simulations of Second Order System

inherently more accurate than those of first order continuous systems. It is also apparent that the advantages of optimization are much greater in the second, than in the first, order system. In order to test this "improvement with order" one step further, the third order system,

$$H(s) = \frac{1}{(s + 1)(s^2 + 2s + 5)} , \qquad (435)$$

was simulated by both a ramp invariant and an optimized third order (3,3,1), digital system. The resulting frequency domain error is shown in Figure 32. Comparing these curves to those of Figure 28 for the second order system, we see that the ramp invariant error is less, the optimized system error is less, and that the improvement of the optimized system over that of the ramp invariant is greater in Figure 32 than in Figure 28. Thus, for the specific continuous transfer functions simulated here, the frequency domain error of a given simulation technique decreases as the order of the simulated system increases.

7.4.2 Time Domain Errors

The remaining figures illustrate the time domain responses of the second order continuous system and its various simulations to the same inputs that were used for the first order system.

Figure 33 represents the impulse response of the impulse (2,I,0), adjusted impulse (2,A,0), and step invariant (2,S,0) simulations. The latter two simulations oscillate about the true value; however, their convergence to the final value is at

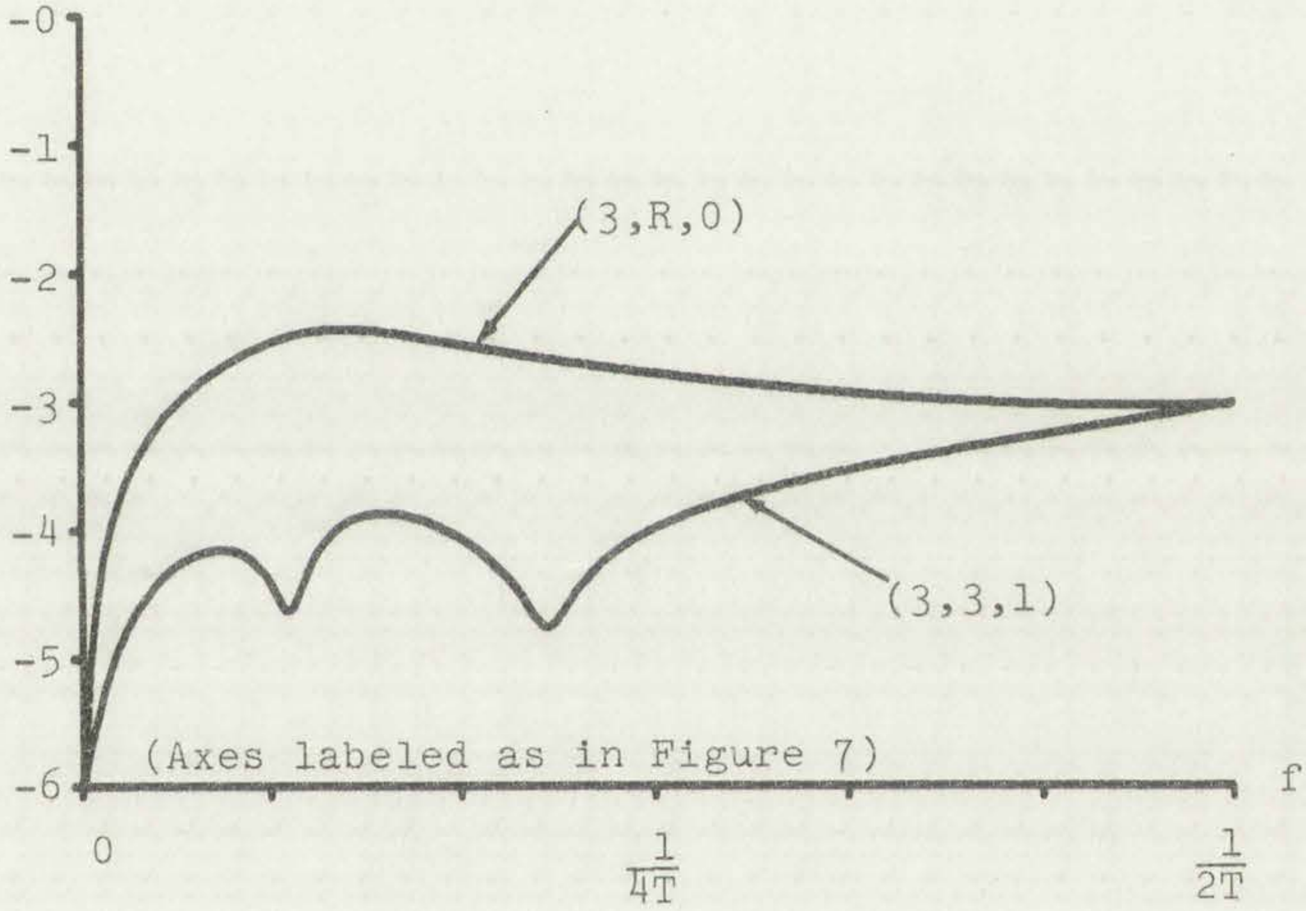
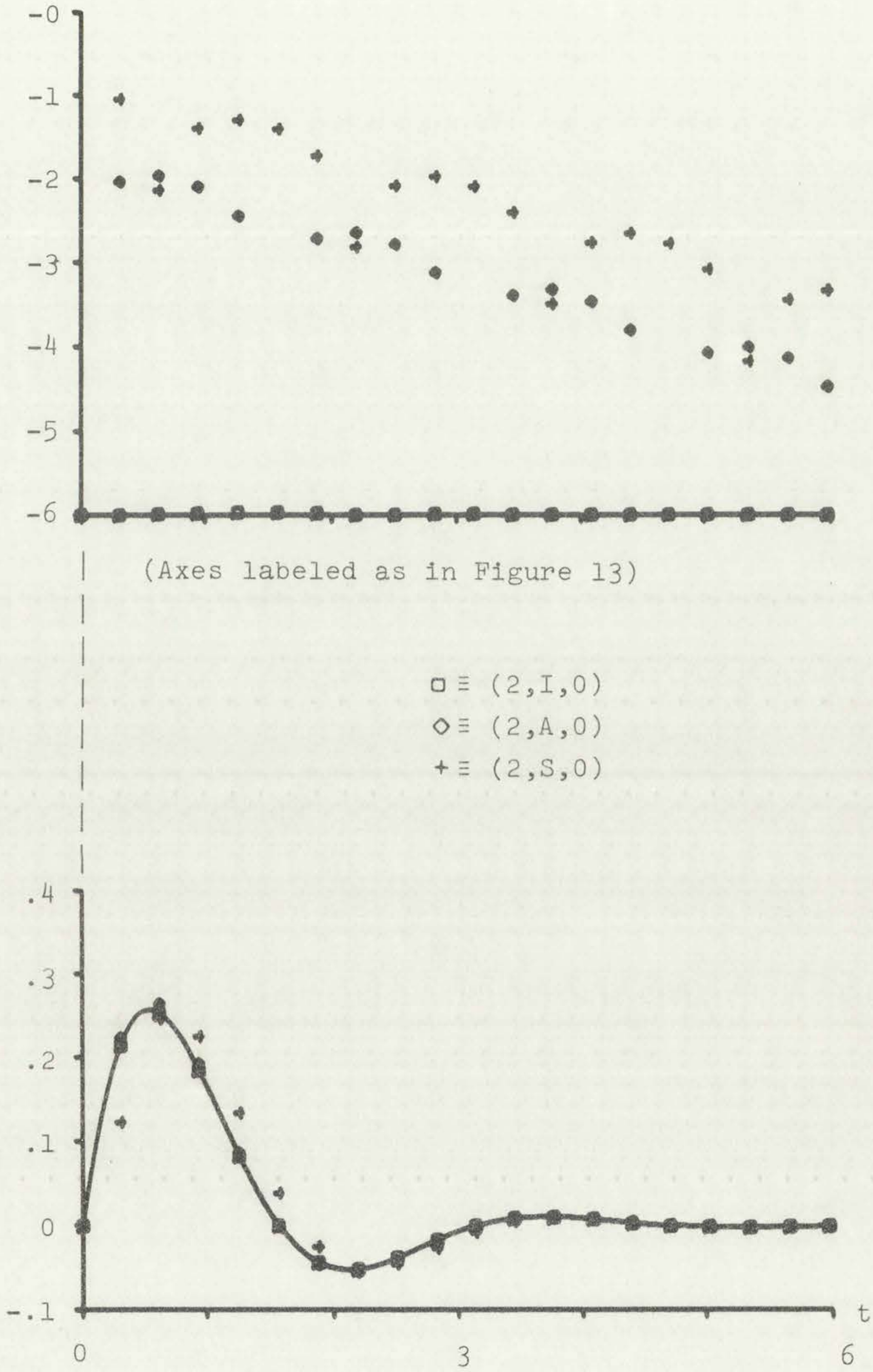


Figure 32. Frequency Domain Error of Ramp Invariant and Third Order Optimized Simulations of Third Order System



(Axes labeled as in Figure 13)

Figure 33. Response and Error of Impulse, Adjusted Impulse and Step Invariant Simulations of Second Order System with Impulse Input

the same rate as the decay of the continuous transient response to zero.

The impulse responses of the ramp invariant (2,R,0), bilinear approximation (2,B,0), and optimized second order (2,2,1) simulations are shown in Figure 34. All three discrete system responses oscillate here; however, they again decay at nearly the same rate as the continuous transient response. The optimized system is clearly the most accurate and, on comparing Figures 33 and 34, we see that it is better than any other simulation shown except, of course, the impulse invariant.

The impulse response of the third order discrete systems (2,3,1), (2,3,2), and (2,3,3) are shown in Figure 35. Since two of the three poles are a complex pair at approximately .73 $e^{\pm j.2\pi}$, only one pole will move outward with the maximum allowed radius of the poles. Thus, the additional term of the impulse response behaves as $e^{-\alpha T}$ where α becomes small as the pole magnitude approaches ∞ and the error becomes nearly constant, as shown. By restricting this "free" pole to a radius of e^{-T}, the transient response error of the third order system (2,3,1) is roughly equivalent to that of the previously discussed second order simulations (2,2,1) and (2,2,2). Similarly, the fourth order digital systems impulse responses are shown in Figure 36. These systems (2,4,1), (2,4,2), and (2,4,3) have two poles free to increase with the maximum allowed radius, giving an unwanted transient response of the form of $te^{-\alpha t}$. Thus, the simulation error increases, as shown, until $t = 1/\alpha$. However, with this pair of poles restricted to a magnitude of

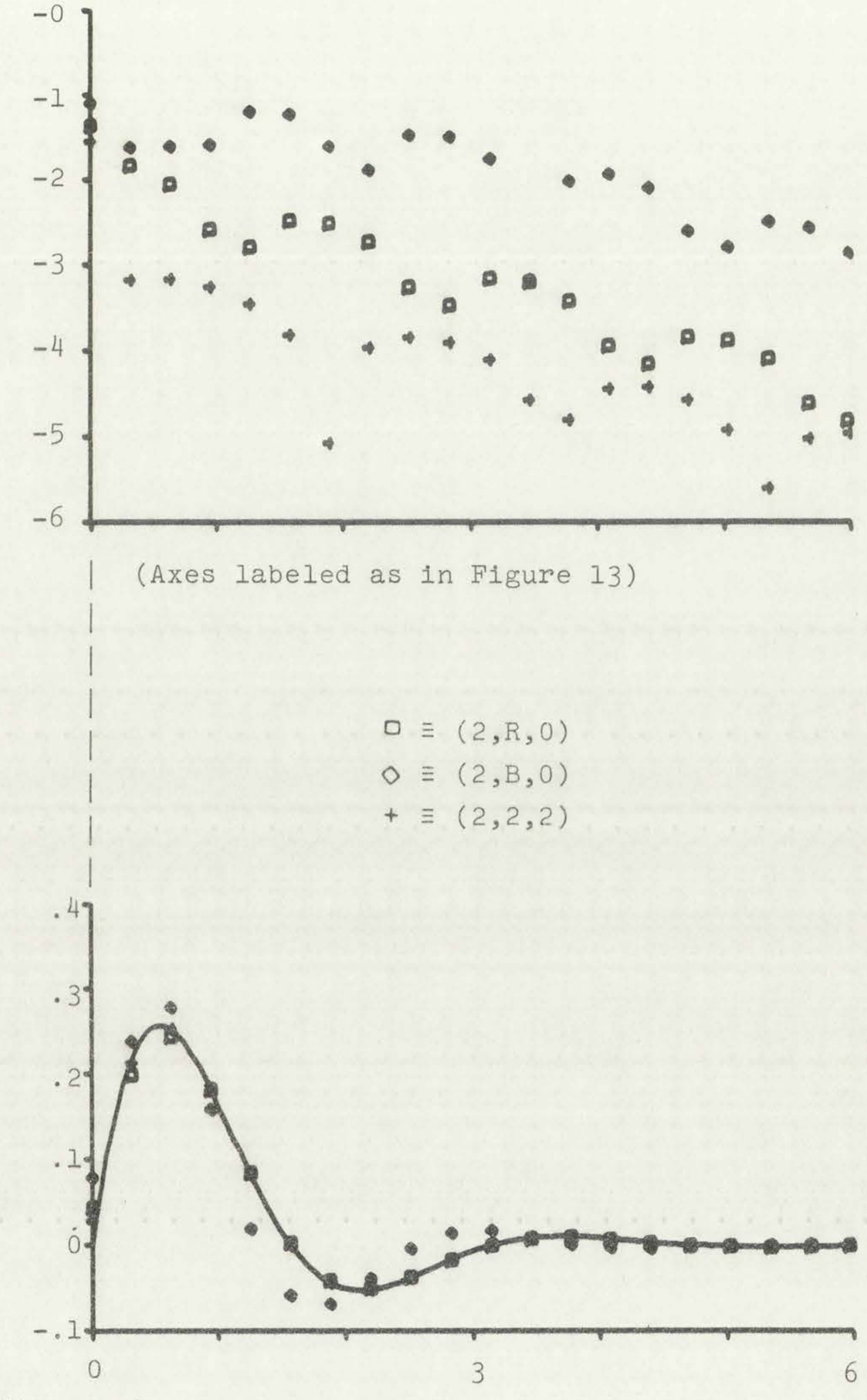
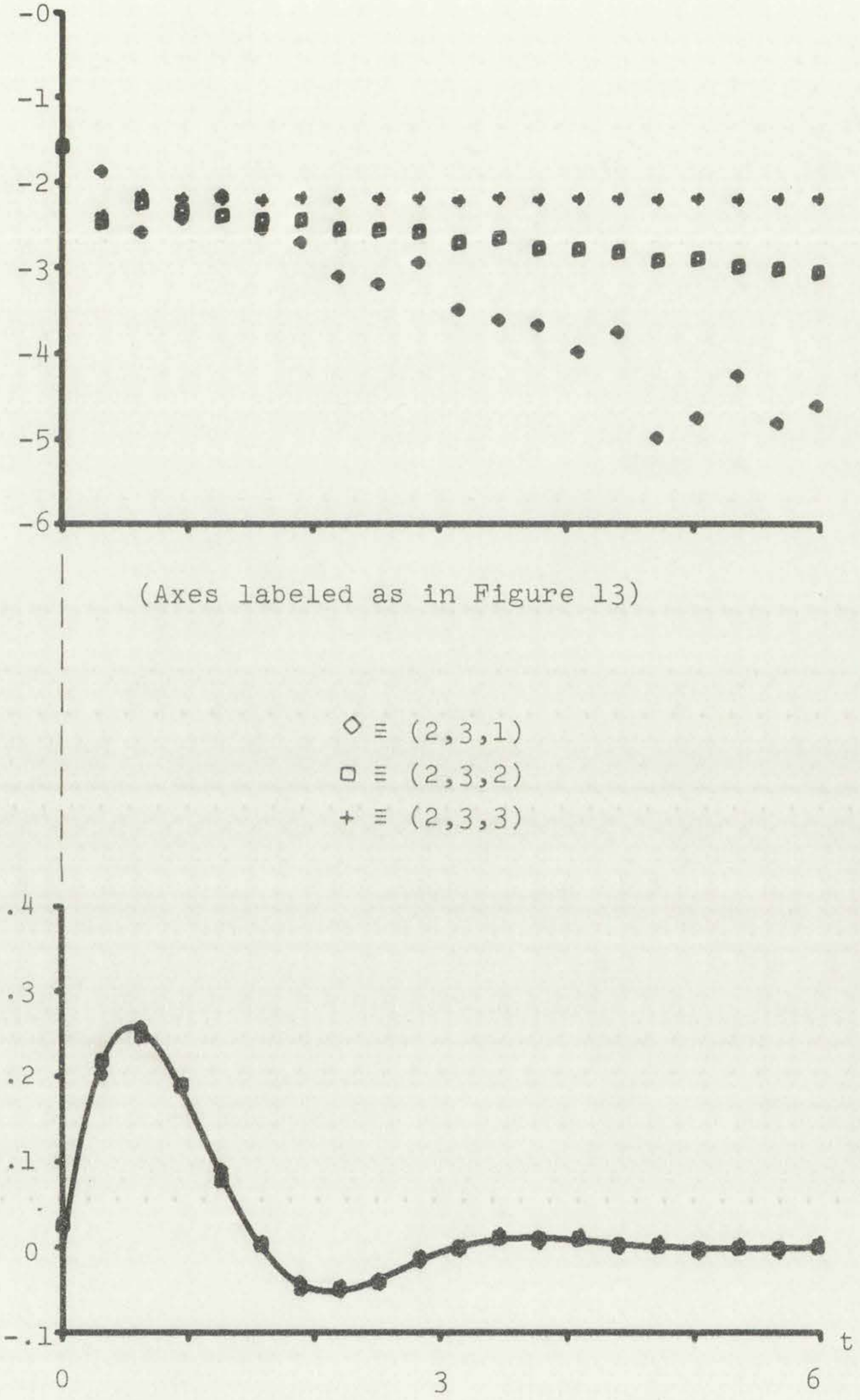
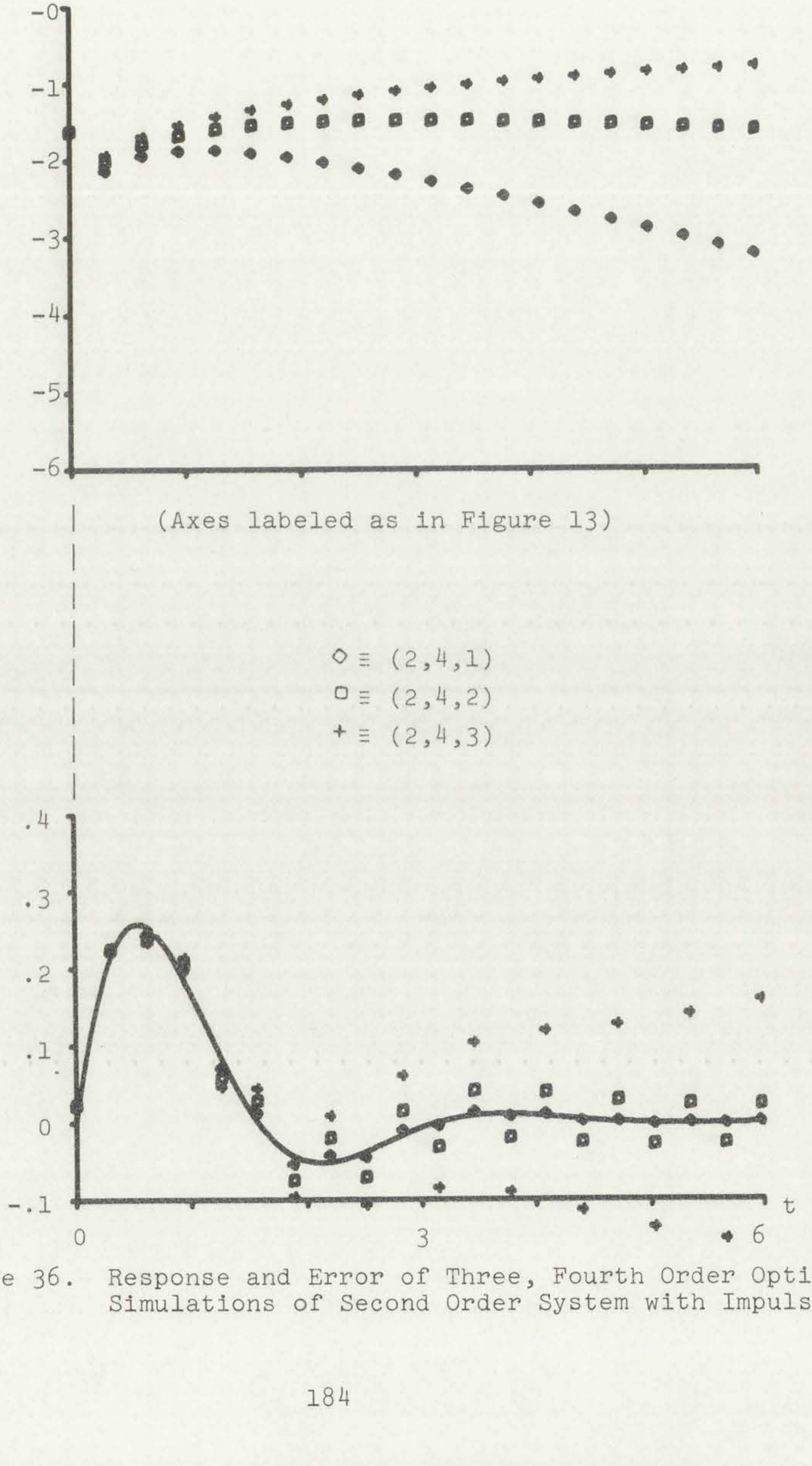

(Axes labeled as in Figure 13)

Figure 34. Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with Impulse Input



(Axes labeled as in Figure 13)

Figure 35. Response and Error of Three, Third Order Optimized Simulations of Second Order System with Impulse Input



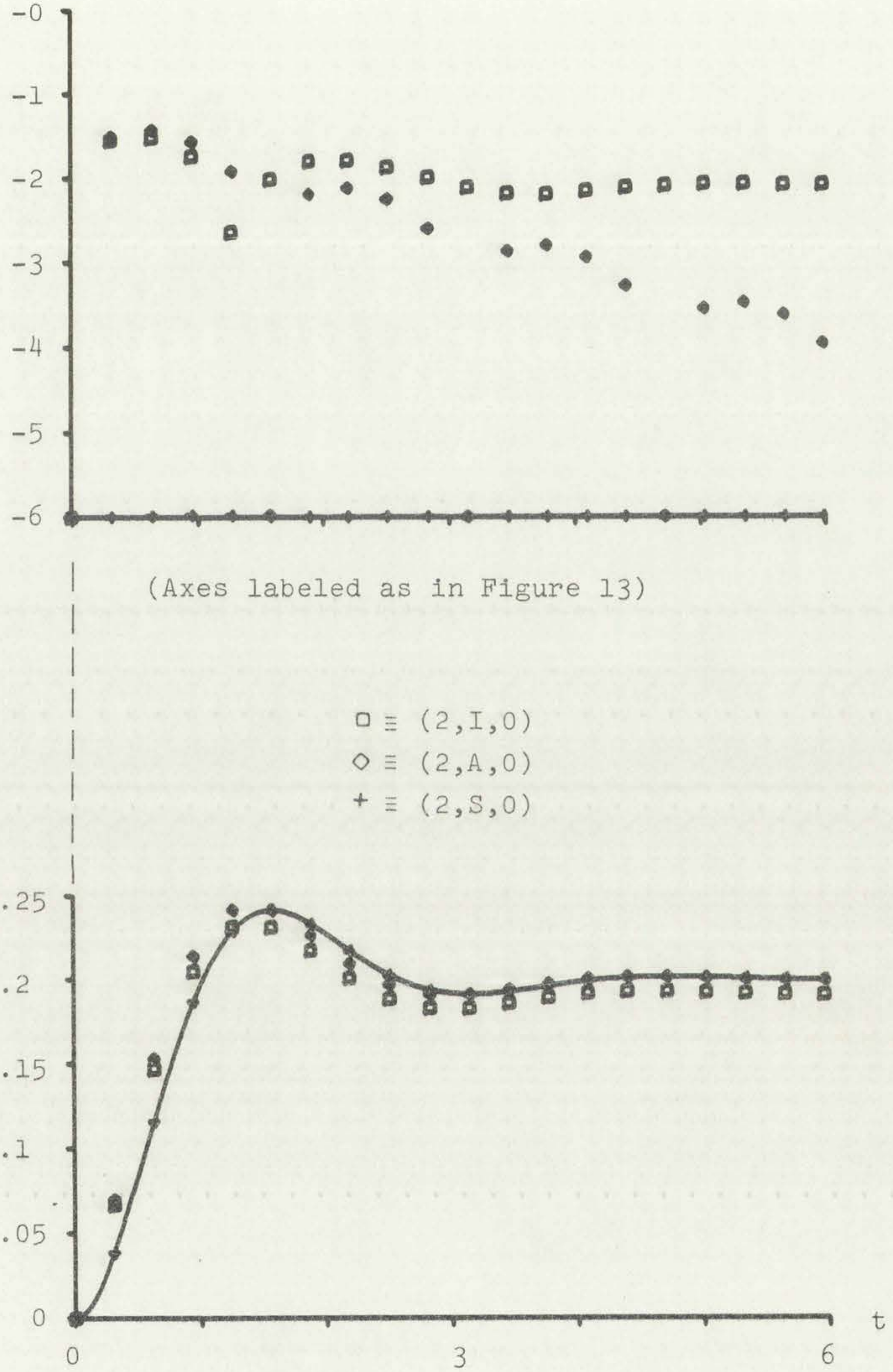
(Axes labeled as in Figure 13)

Figure 36. Response and Error of Three, Fourth Order Optimized Simulations of Second Order System with Impulse Input

e^{-T}, the maximum error occurs at about the third sample time and beyond that time it decays to zero at nearly the same rate as the second and third order systems. With the pair of poles at a magnitude of .9 the maximum error occurs at about the 15th sample and begins to decay at a much slower rate, as shown, while for a pole magnitude of .999 the maximum error occurs at about the 1000th sample, where a value in excess of 3.0 is reached, and then this transient begins to die out at a very slow rate.

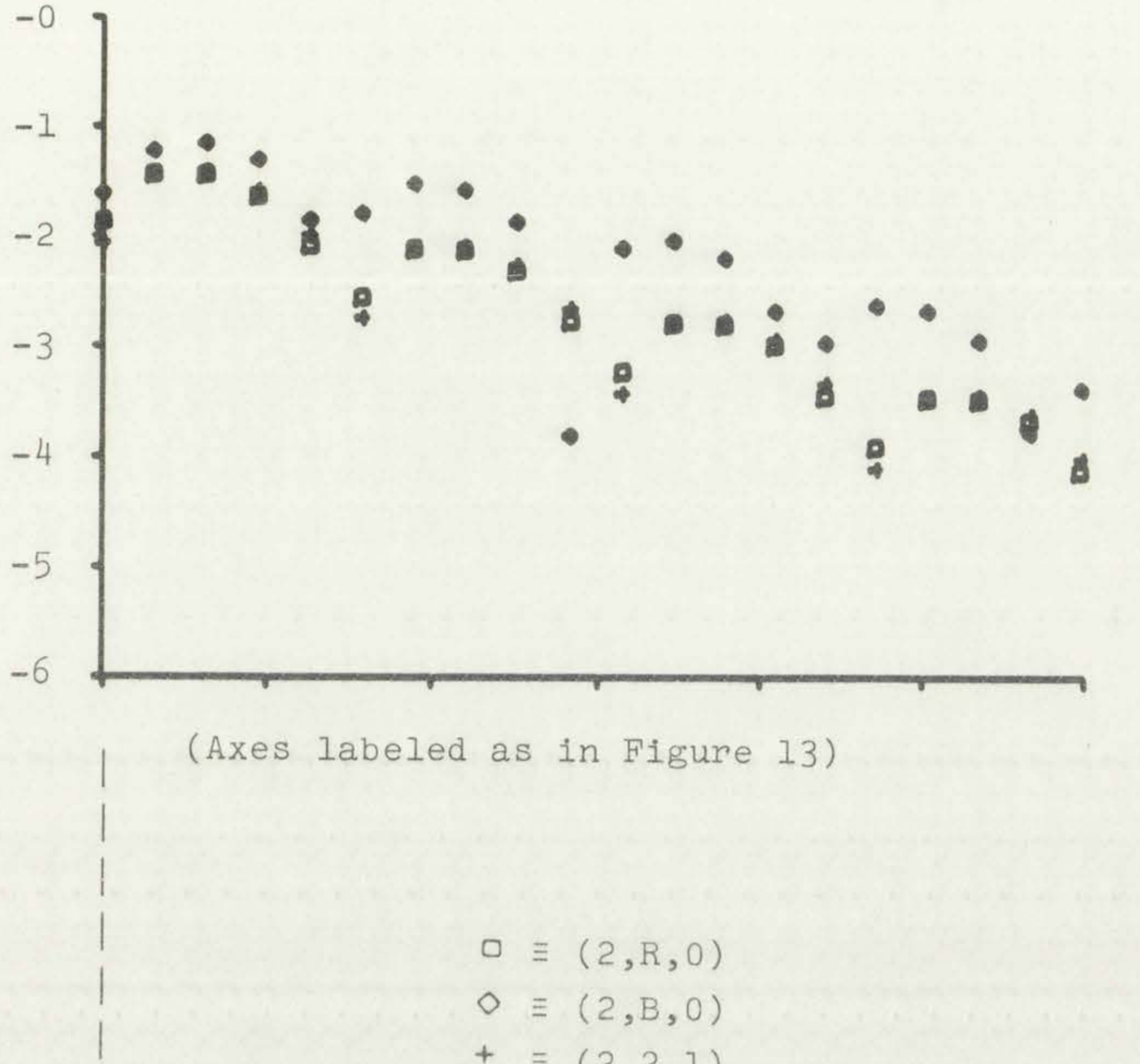
The step response of the impulse (2,I,0), adjusted impulse (2,A,0), and step invariant (2,S,0) simulations is given in Figure 37 while Figure 38 presents the similar responses of the ramp invariant (2,R,0), bilinear approximation (2,B,0), and second order optimized (2,2,1) simulations. The step invariant method is, of course, exact while the impulse invariant method exhibits its dc offset. The optimized system here is at least as good as the other "inexact" simulations.

Figures 39 and 40 present the step responses of the third (2,3,1), (2,3,2), and (2,3,3) and fourth (2,4,1), (2,4,2), and (2,4,3) order optimized systems, respectively. In both figures the response of the systems with the poles restricted to e^{-T} has errors which approach zero at approximately the same rate as the continuous system approaches its final value. However, as the magnitude of the poles is allowed to increase to .999 the third order system error dies out very slowly while the fourth order system error continues to increase for the first 1000 samples, at which time the response is oscillating between



(Axes labeled as in Figure 13)

Figure 37. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with Step Input



(Axes labeled as in Figure 13)

Figure 38. Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with Step Input

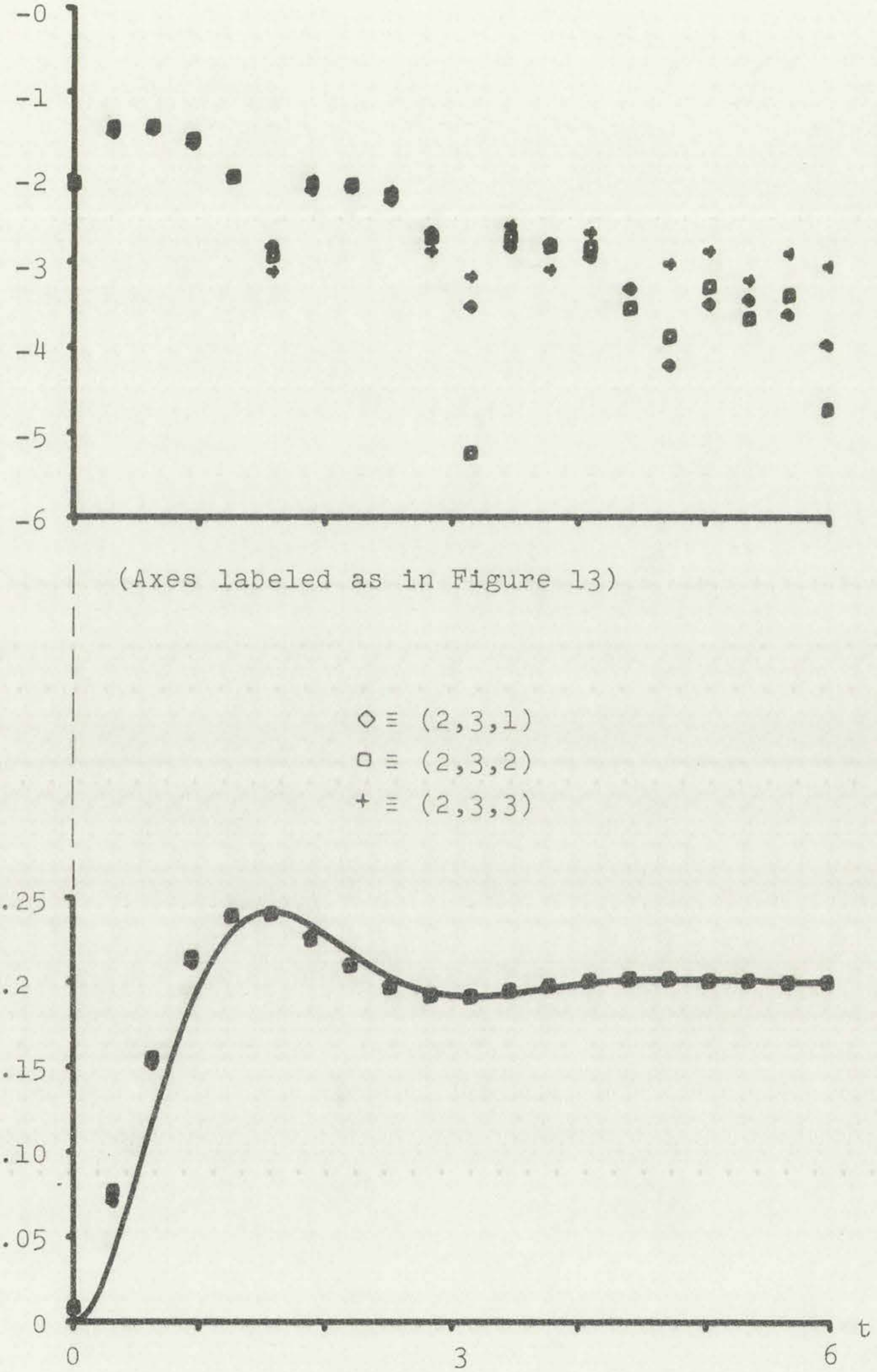
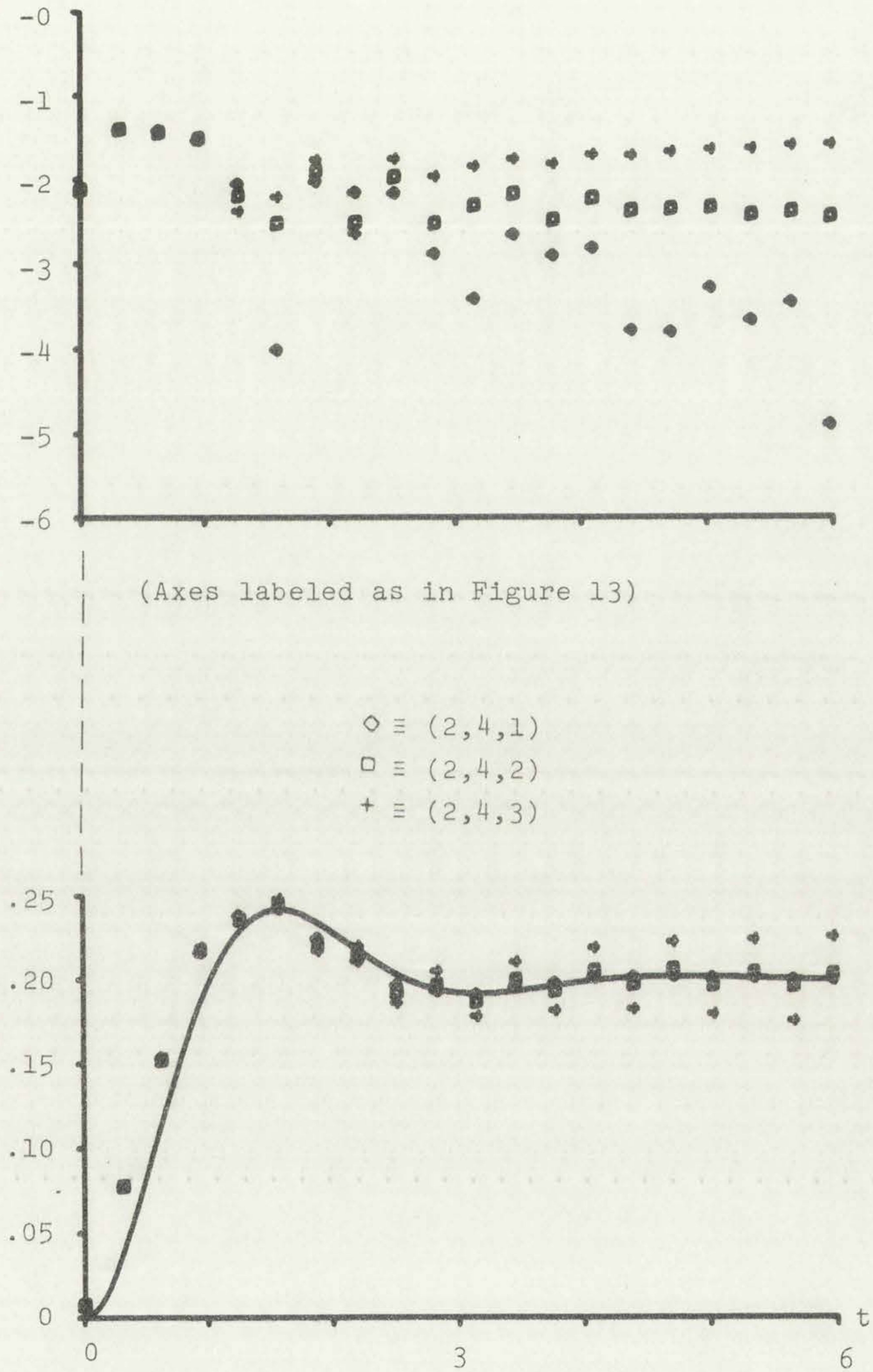

(Axes labeled as in Figure 13)

Figure 39. Response and Error of Three, Third Order Optimized Simulations of Second Order System with Step Input



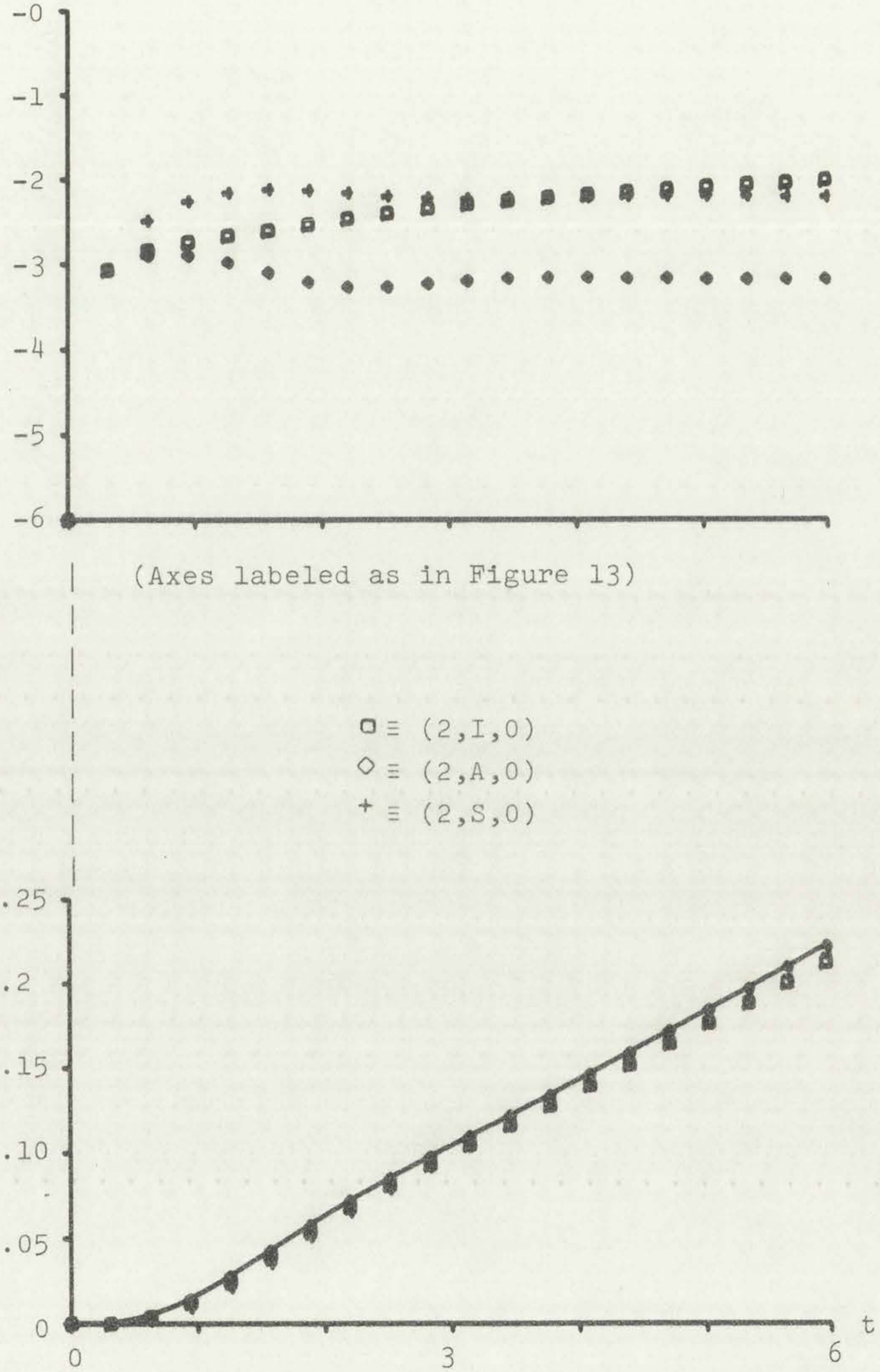
(Axes labeled as in Figure 13)

Figure 40. Response and Error of Three, Fourth Order Optimized Simulations of Second Order System with Step Input

.683 and -.283. In fact, after 10,000 sample intervals the response error is still .0005, about the same as the error of the fourth order system with the poles restricted to e^{-T} after 20 sample intervals.

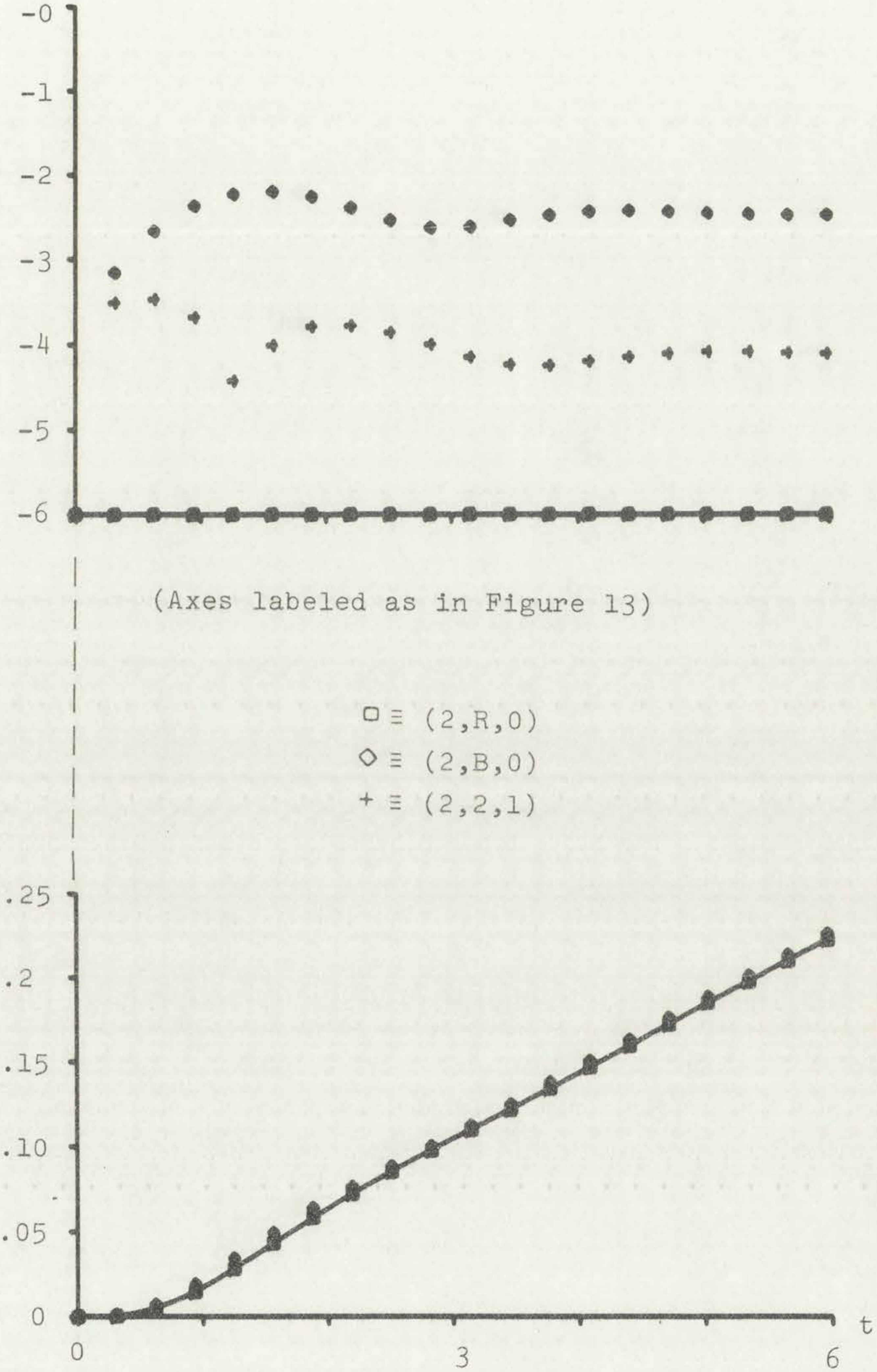
Figure 41 shows the ramp input responses of the impulse (2,I,0), adjusted impulse (2,A,0), and step invariant (2,S,0) simulations while Figure 42 gives the responses of the ramp invariant (2,R,0), bilinear approximation (2,B,0), and second order optimized (2,2,1) simulations to the ramp input. Just as for the simulations of the first order system of Figures 21 and 22 the errors here remain relatively constant after about the 10th sample interval for all except the impulse invariant simulation, whose error continues to increase. It is also apparent that the second order optimized simulation (2,2,1) is better than any of the others, with the exception, of course, of the exact ramp invariant system.

The responses of these same six digital systems to a sin 3t input are compared in Figures 43 and 44. Little can be said here except that the optimized system is clearly the best, that is, its error is the least at almost every sample interval. Similarly, the systems are compared for an input of sin 7t in Figures 45 and 46. Here again the optimized systems response error is the minimum at over half the sampling instants shown and on this basis it could be assumed to be the best of the simulations shown. However, as stated previously in the discussion of Figures 23 through 26, this type of time domain comparison for sinusoidal inputs is not as meaningful as the



(Axes labeled as in Figure 13)

Figure 41. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with Ramp Input



(Axes labeled as in Figure 13)

Figure 42. Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with Ramp Input

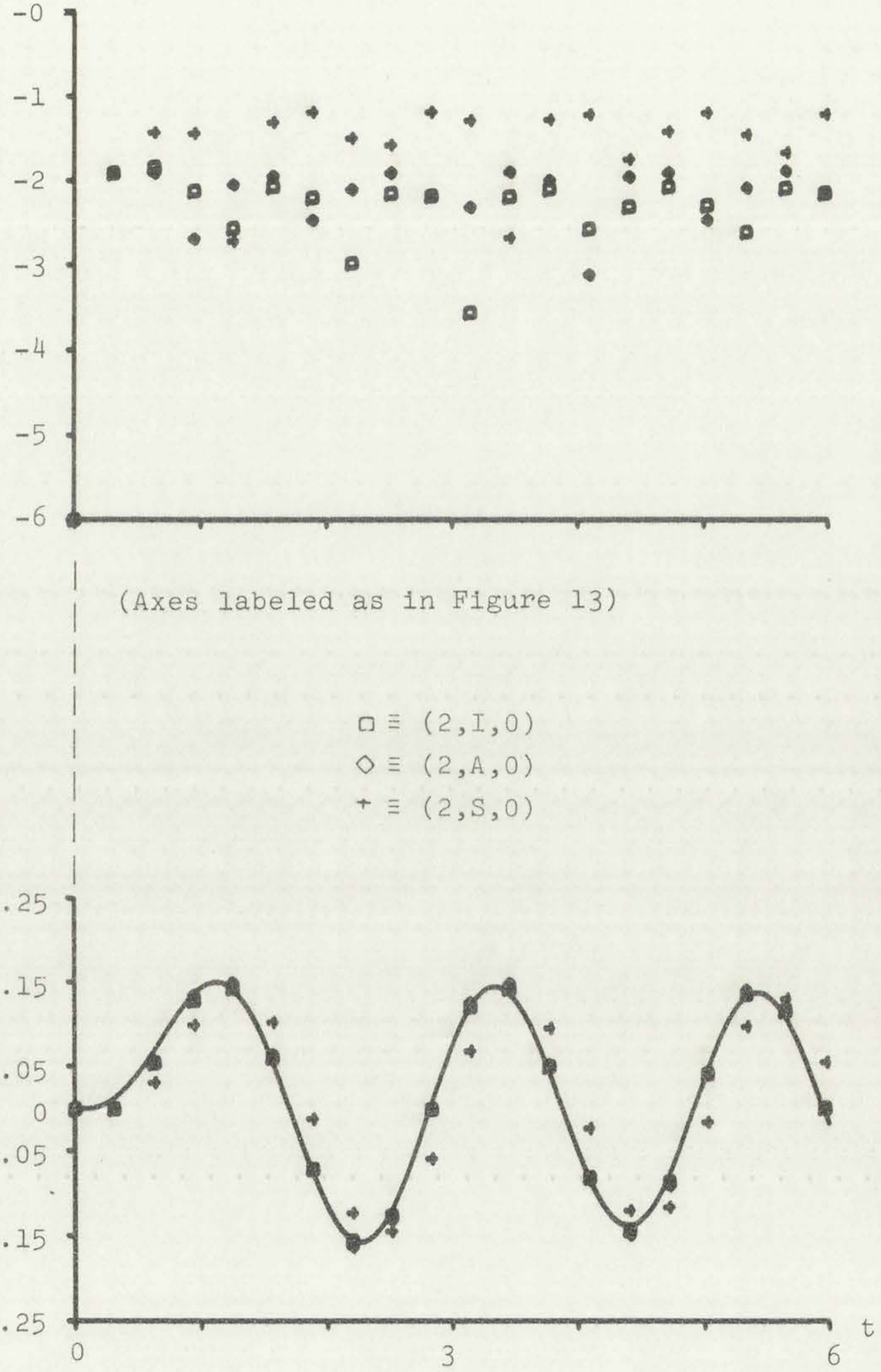


Figure 43. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with sin 3t Input

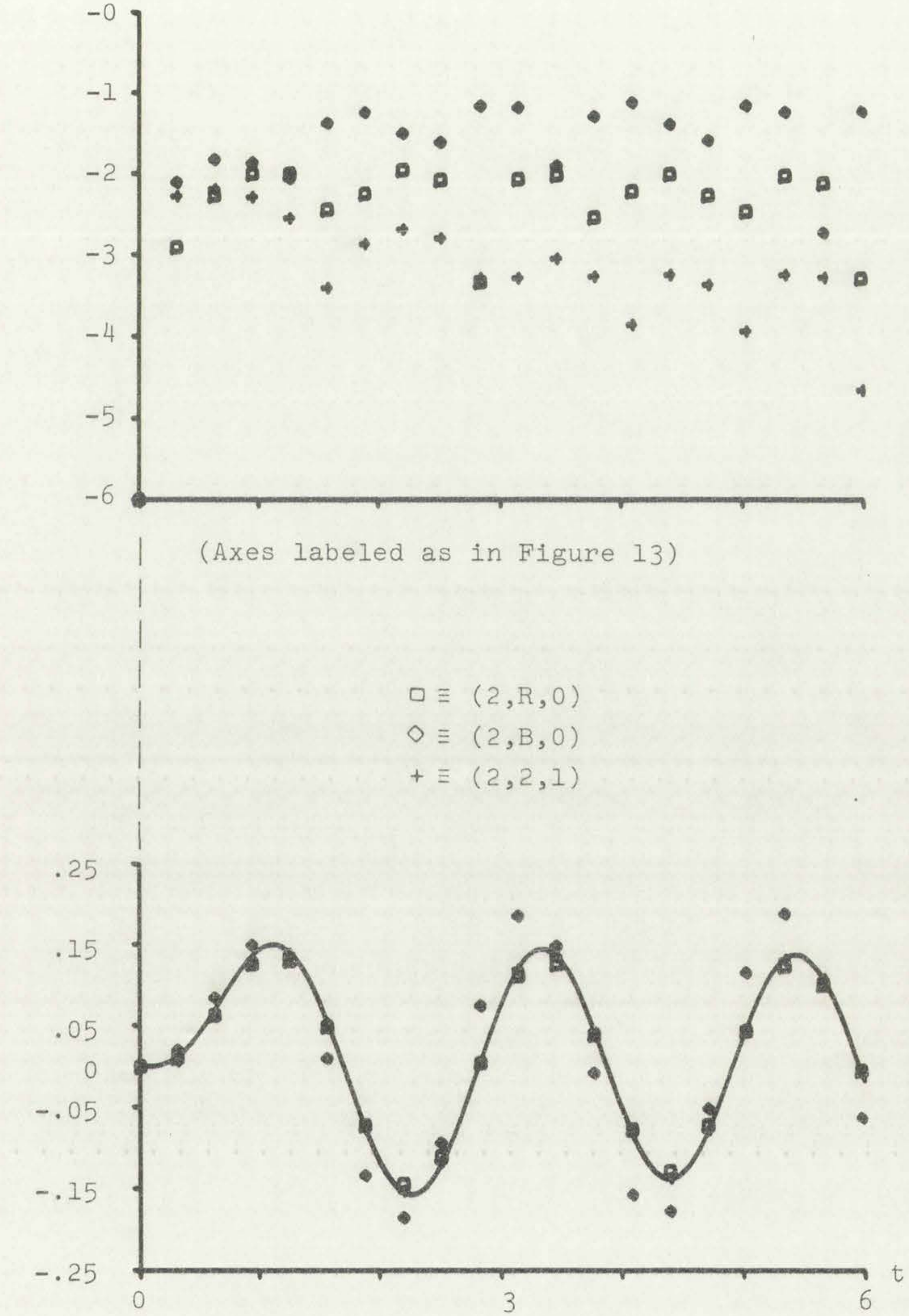


Figure 44. Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with sin 3t Input

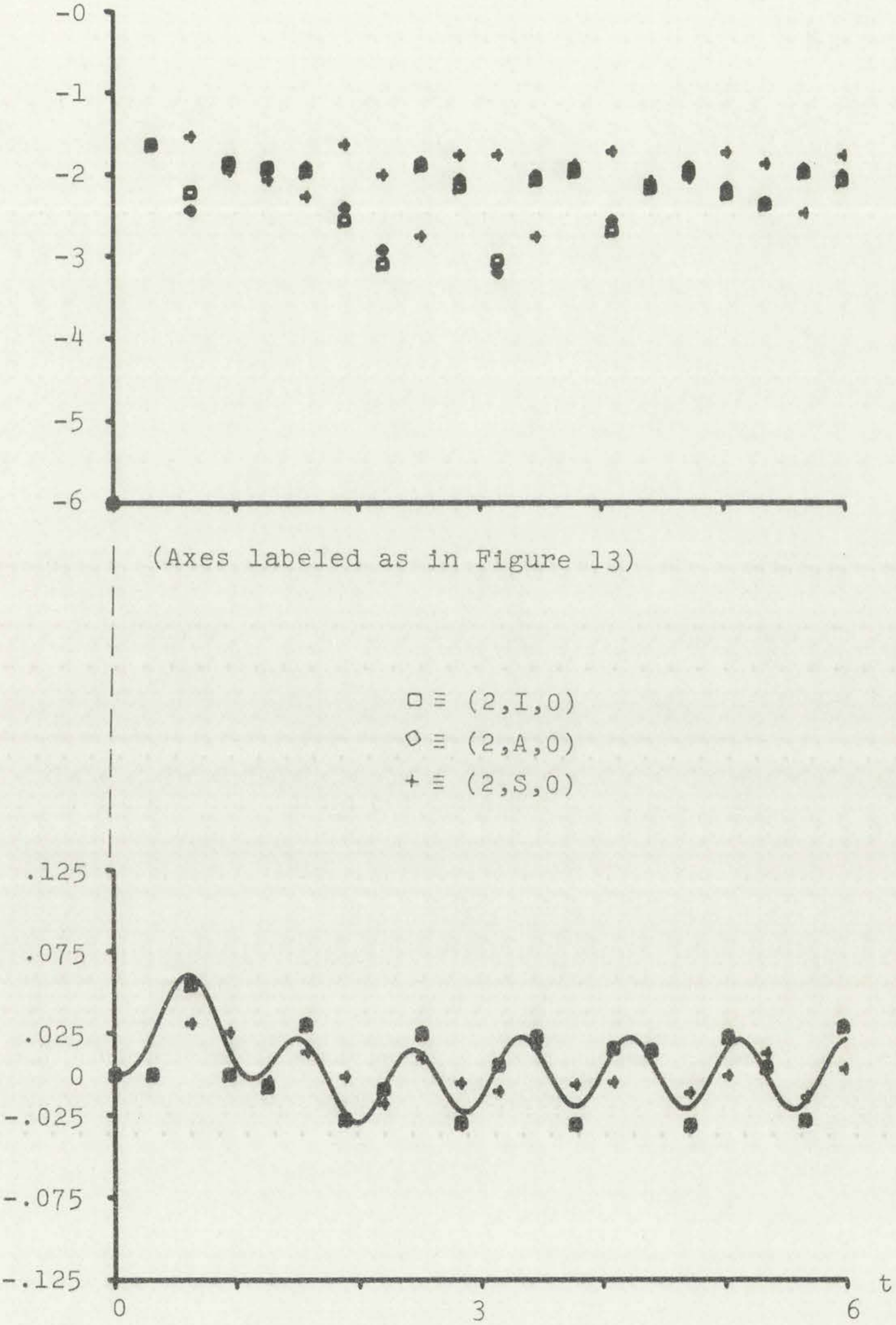


Figure 45. Response and Error of Impulse, Adjusted Impulse, and Step Invariant Simulations of Second Order System with sin 7t Input

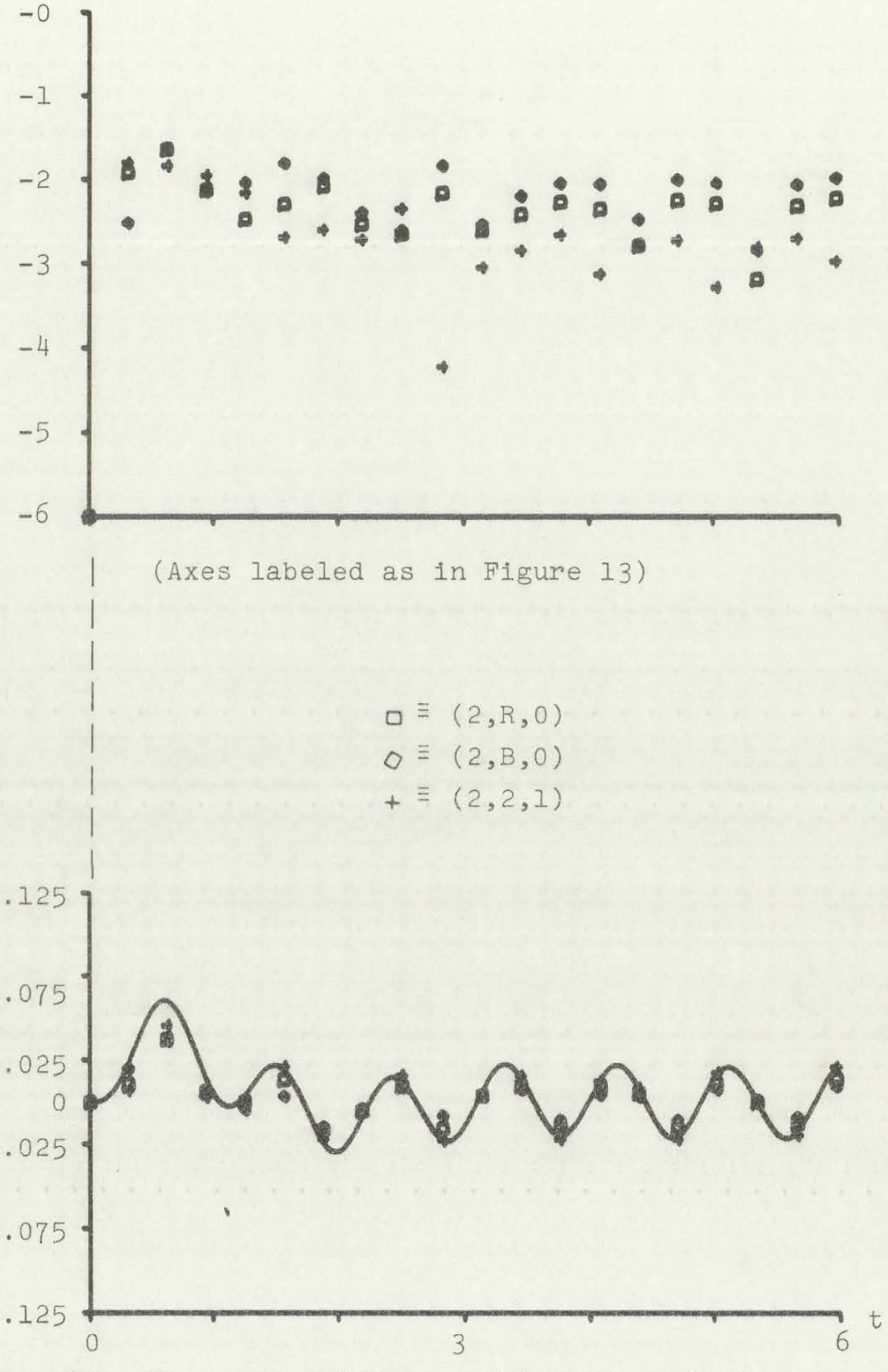


Figure 46. Response and Error of Ramp Invariant, Bilinear Approximation, and Second Order Optimized Simulations of Second Order System with sin 7t Input

frequency domain comparison of Figures 27 and 28. There the optimized system was clearly shown to be the best of the six systems compared here.

7.5 Summary of Results

In this chapter the optimized simulation proposed in this paper has been compared with other simulations of a first and second order system. These comparisons have been made both in the frequency domain, $0 \leq \omega \leq \pi/T$, and in the time domain for a variety of input signals.

From Figures 7 through 26 the comparisons of the simulations of the first order system are summarized below.

The frequency domain error comparisons of the first order simulations, Figures 7 and 8, show the ramp invariant (1,R,0) and first order optimized (1,1,1) simulations to be in general the best. Although the bilinear approximation (1,B,0) does have less error over nearly 50% of the frequency range, it does suffer from a large error at very low frequencies. The remaining three simulations of Figure 7 all contain much more error than those of Figure 8.

In order to reduce the frequency domain error further, the order of the simulation must be increased to second, third, fourth, or higher. The frequency domain error of the first through fourth order optimized systems (1,1,1), (1,2,1), (1,3,1), and (1,4,1) is compared in Figure 12, showing the decrease in error as the order of the simulation system is increased. In Figures 9, 10, and 11 it is shown that the frequency domain comparison of digital and [illegible]

[illegible] system was [illegible] shown to be [illegible]

[illegible]

[illegible]

In this chapter [illegible]

[illegible] second order systems. [illegible] the frequency domain, [illegible] and [illegible] of input signals.

From Figures 1 through [illegible] the [illegible] of the [illegible]

The frequency domain error comparison [illegible] simulations, Figures 7 and 8, [illegible] and first order [illegible] (1,1,1) [illegible]. Although the [illegible] error over [illegible] [illegible] error is very [illegible] the three simulations of Figure 8 [illegible] than that of Figure 9 [illegible]

In order to reduce the frequency [illegible] order of the simulation must be increased [illegible] fourth, or higher. The frequency domain [illegible] through fourth order optimized [illegible] and [illegible] is illustrated in Figure [illegible] in [illegible] of the order of the [illegible] in Figures 9, 10, and [illegible] error of these systems continues to decrease as the poles are allowed to increase in magnitude. Thus, to use the optimization procedure the maximum pole radius must be restricted, as is done here to limit the time domain error.

The time domain errors of the first order simulations (1,I,0), (1,A,0), (1,S,0), (1,R,0), (1,B,0), and (1,1,1) are shown in Figures 13 and 14 for the impulse input, in Figures 17 and 18 for the step input, in Figures 21 and 22 for the ramp input, and in Figures 23, 24, 25, and 26 for the sinusoidal inputs. None of these systems appear to be better, or worse, than any of the others in an overall, time domain error sense. The impulse invariant (1,I,0) simulation does have large, non-diminishing errors with the step and ramp inputs due to its incorrect dc gain and is perhaps the poorest of the group for this reason.

The remaining figures show the effects on the time domain response of allowing the z-plane poles to approach the unit circle. The response and approximation error of the second order simulations (1,2,1), (1,2,2), and (1,2,3) for the impulse and step inputs are shown in Figures 15 and 19, respectively. Similarly, the third order simulations (1,3,1), (1,3,2), and (1,3,3) response and approximation error are shown in Figures 16 and 20. The errors of these higher order simulations decay at a rate similar to the first order systems <u>if</u> the poles are restrained to a magnitude of less than e^{-T}. However, as these poles are allowed to approach the unit circle the second order simulation error decays very slowly while for

the third order simulation the error increases with time until some maximum is reached, as discussed in Section 7.3.2.

In general, for the simulation of the first order system, $H(s) = \frac{1}{s + 1}$, this chapter has shown that:

1. if the input is of a known form, use the simulation which is invariant to that form.
2. if the input is unknown, then either the ramp invariant (1,R,0) or first order optimized (1,1,1) is the preferable first order simulation.
3. the higher order approximations (1,2,1), (1,3,1), or (1,4,1) will provide significantly reduced frequency domain error with little increase in the time domain errors.
4. poles of these higher order simulations must be restricted to a maximum radius of about e^{-T} in order that the transient response of the discrete system decays at nearly the same rate as that of the continuous system.

Figures 27 through 46 illustrate similar properties of the simulations of the second order system. Figure 32 is actually a comparison of simulation errors for a third order system. The frequency domain error comparisons of Figures 27 and 28 show clearly the advantages of the second order optimized (2,2,1) simulation over all others of second order (2,I,0), (2,A,0), (2,S,0), (2,R,0), and (2,B,0). In fact, it (2,2,1)

is nearly an order of magnitude better than any of the others over 2/3 of the frequency range.

The decrease in frequency domain error with the increasing discrete system order is shown in Figure 31. Figures 29 and 30 show the almost insignificant decrease in error as the higher order system poles are allowed to approach the unit circle. However, this decrease in error will force the optimization algorithm to place the "extra" poles at the maximum allowed radius.

The frequency domain error of the first order optimized simulation (1,1,1) of the first order system was not significantly less than that of the ramp invariant simulation (1,R,0). However, for the second order system the optimized simulation (2,2,1) was much better than that of the ramp invariant (2,R,0). Figure 32 is a similar comparison of the frequency domain error of the optimized (3,3,1) and ramp invariant (3,R,0) simulations of a third order system. Here the margin of superiority of (3,3,1) to (3,R,0) is even greater than that for the second order system.

The time domain error comparisons of the six second order simulations (2,I,0), (2,A,0), (2,S,0), (2,R,0), (2,B,0), and (2,2,1) differ significantly from those of the first order systems. On comparing Figures 33 and 34, 37 and 38, 41 and 42, 43 and 44, and, finally, 45 and 46, we see that in every example the optimized simulation (2,2,1) introduces the least time domain error of any of these second order simulations which are not invariant to the particular input.

The remaining time domain error comparisons simply illustrate again the adverse effects of allowing the z-plane pole magnitude to exceed e^{-T} and the nearly negligible time domain error effect of increasing the discrete system order while restraining the pole magnitude.

Thus, for simulations of the second order system, $H(s) = \frac{1}{s^2 + 2s + 5}$, the chapter has shown that:

1. the invariant simulation of a known input form should be used.

2. the second order optimized simulation (2,2,1) is clearly superior to all other second order simulations compared here in both the time and frequency domains for unknown inputs.

3. in order to further decrease frequency domain simulation error higher order discrete systems must be used with the poles restricted to a magnitude of about e^{-T} or less.

In addition, we have shown for these particular simulations, the increasing advantage of optimization techniques as the order of the simulated system increases.

Although the comparisons of this chapter have been of very specific systems, the theory involved may be applied to the general simulation problem.

CHAPTER 8. CONCLUSIONS

In Chapter 7 it was found that the impulse, step, and ramp invariant simulations of both the first and second order systems were each distinct and unique. It is shown in Appendix B that a discrete invariant simulation of a given continuous system is invariant only for an input of a specific form. The general simulation problem involves inputs of unknown form; or inputs made up from an ensemble of known forms. In either case an invariant simulation is nonexistent.

With no general invariant simulation method possible, error in the time domain will usually be present, i.e., $y_n \neq y(nT)$. Similarly, simulation errors in the frequency domain will also be present, as discussed in Chapter 3. Thus, the optimum simulation of a specific continuous transfer function is optimum only in the sense that these errors are minimized for a given complexity simulation; they cannot be eliminated.

In this paper a frequency domain optimization procedure was defined and its characteristics presented. These characteristics were developed only for a first and second order system. However, since any continuous or discrete rational transfer function can be represented as a sum or cascade of first and second order subsystems, the properties found may be extended to any order system.

7. Optimized simulations of higher order than the simulated system may place the "excess z-plane poles," if unconstrained, on or outside the unit circle to obtain minimum frequency domain error.

8. The magnitude of the higher order discrete system poles may be restricted to be less than 1 at little "cost" in frequency domain error, while significantly decreasing the time domain error.

9. Greater error reduction is obtainable with optimization as the order of the simulated system increases. That is, the margin of superiority of optimization over other available methods increases with increasing system order.

These properties show the described optimization procedure to be applicable to a wide range of simulation problems. The results obtained, for the error frequencies chosen here, indicate the optimized simulation is generally superior to the other methods used for comparison.

The error criterion used here does not require the continuous transfer function to be rational. The simulation of irrational functions should be investigated. In addition, a weighted error measure might be useful; the effects of various "weights" deserve future study.

APPENDIX A

********** OPT **********

```
        EXTERNAL FUNZ
        COMMON A(10),B(10),NZ,NP,NCP,NT,V1,KOUNT,PI,T,ET,DUM(20)
        DIMENSION H(300),GRAD(20),ARG(20)
        ASIN(X)=ARSIN(X)
        PI=3.14159
        T=.1*PI
        NIN=5
        NOUT=6
1       READ(NIN,20) NZ,NP,NCP,NO,NEXP,ET
        IF(NZ.EQ.9) GO TO 9
        IF(NZ.EQ.0) GO TO 2
        IF(NEXP.EQ.0) NEXP=5
        IF(ET.EQ.0.) ET=EXP(-T)
        EXMIN=-NEXP
        EXMIN=10.**EXMIN
        ET1=ET
        NP1=NP
        NZ1=NZ
        NO1=NO
        NT=NP+NZ
        NCP1=NCP+1
        READ(NIN,21) (ARG(I),I=1,10)
2       KOUNT=0
        V1=100000
        IF(NZ.NE.0) GO TO 94
        NZ=NZ1
        ET=ET1
        NP=NP1
        NCP=NCP1-1
        NO=NO1
        GO TO 106
94      CONTINUE
        IF(NCP.EQ.0) GO TO 101
        DO 100 L=2,NCP,2
        IF(ARG(L+NZ).GT. ET) ARG(L+NZ)=ET*.9
        IF(ARG(L+NZ).LT.-ET)  ARG(L+NZ)=-ET*.9
100     ARG(L+NZ)=ASIN(ARG(L+NZ)/ET)
101     IF(NP.EQ.NCP) GO TO 103
        DO 102 L=NCP1,NP
        IF(ARG(L+NZ).GT.ET) ARG(L+NZ)=ET*.9
        IF(ARG(L+NZ).LT.-ET) ARG(L+NZ)=-ET*.9
102     ARG(L+NZ)=ASIN(ARG(L+NZ)/ET)
103     CONTINUE
```

```
 106    WRITE(NOUT,22) NZ,NP,NCP,NT,EXMIN,ET
 104    WRITE(NOUT,23) (ARG(L),L=1,NT)
 105    IF(IER.EQ.1) KOUNT=0
        CALL FMFP(FUN2,NT,ARG,F,GRAD,EXMIN,.000011   ,300,IER,H)
        WRITE(NOUT,24) F,IER,KOUNT
        WRITE(NOUT,25) (ARG(I),I=1,NZ)
        IF(NCP.EQ.0) GO TO 215
        WRITE(NOUT,26) (B(J),J=1,NCP)
 215    IF(NCP.EQ.NP) GO TO 250
        WRITE(NOUT,27) (B(J),J=NCP1,NP)
 250    WRITE(NOUT,28) (GRAD(J),J=1,NT)
        IF(NO.EQ.0) GO TO 1
        DO 3000 J=1,13
        WT=(J-1)*PI/12.
        W=WT/T
        D=(5.-W**2)**2+4.*W**2
        H1=(5.-W**2)/D
        H2=-2.*W/D
        H21=H1
        H22=H2
        D1=1.+W**2
        H11=1./D1
        H12=-W/D1
        H1=H11*H21-H12*H22
        H2=H11*H22+H12*H21
        H1=1./D1
        H2=-W/D1
        CALL GRAD3(0,WT,HBR,HBI,GRAD,H)
        ER=SQRT((H1-HBR)**2+(H2-HBI)**2)
        ERR=ALOG10(ER+.000001)
        WRITE(NOUT,29) WT,ERR,H1,HBR,H2,HBI
3000    CONTINUE
        DO 3100 J=1,NT
3100    ARG(J)=FLOAT(IFIX(ARG(J)*1.E06+.5))/1.E06
        CALL FUN2(NT,ARG,F,GRAD)
        WRITE(NOUT,31) F,(ARG(J),J=1,NT)
        GO TO 1
  20    FORMAT(5I1,F10.5)
  21    FORMAT(10F8.4)
  22    FORMAT(1H0,//' NZ,NP,NCP,NT,EXMIN,ET ,4I5,E12.4,F8.4)
  23    FORMAT('0ARG SENT ,(9E11.3))
  24    FORMAT('0 VAL  ,E20.6,'   ERROR CODE IS ,I5,'  AFTER
    /   ,I5,' CALLS TO FUN2 )
  25    FORMAT('0NUM COEF ARE ,4E20.9/(2E20.9,10X,2E20.9))
  26    FORMAT('0COMPLEX POLES ARE ,5E20.9/(2E20.9,10X,2E20.9))
  27    FORMAT('0NEG REAL POLES ,(4E20.9))
  28    FORMAT('0GRADIENTS ARE ,(5E20.9))
  29    FORMAT( F7.3,6F12.6)
  31    FORMAT('06 PLACE ACC FOR POLES AND ZEROES,VAL  ,E20.6,
    /   '  ARGUMENTS ARE /(5E20.8))
   9    CONTINUE
        STOP
        END
```

********** GRAD3 **********

```
      SUBROUTINE GRAD3(L,OMEG,HBR,HBI,G1,G2)
      COMMON A(10),B(10),NZ,NP,NCP,NT,V1,KOUNT,PI,T,ET,ARG(20)
      DIMENSION G1(1),G2(1)
      ASIN(X)=ARSIN(X)
      NZE=NZ-MOD(NZ,2)
      NCP1=NCP+1
      C=COS(OMEG)
      S=SIN(OMEG)
      C2=COS(2.*OMEG)
      S2=SIN(2.*OMEG)
      HNR=.2
      HNR=1.
      HNI=0.
      HDR=1.
      HDI=0.
      IF(NZ.EQ.NZE) GO TO 95
      HNRO=HNR
      HR=(1.+C*A(NZ))/(1.+A(NZ))
      HI=-S*A(NZ)/(1.+A(NZ))
      HNR=HNRO*HR-HNI*HI
      HNI=HNRO*HI+HNI*HR
95    IF(NZE.EQ.0) GO TO 110
      DO 100 J=1,NZE,2
      HNRO=HNR
      HR=(1.+A(J)*C+A(J+1)*C2)/(1.+A(J)+A(J+1))
      HI=-(A(J)*S+A(J+1)*S2)/(1.+A(J)+A(J+1))
      HNR=HNRO*HR-HNI*HI
100   HNI=HNRO*HI+HR*HNI
110   IF(NCP.EQ.0) GO TO 130
      DO 120 J=1,NCP,2
      B1=B(J+1)
      CB=COS(B(J))
      D=1.+2.*B1*CB+B1**2
      HDRO=HDR
      B12=B1**2
      HR=(1.+2.*P1*CB*C+(B1**2)*C2)/D
      HI=-(2.*B1*CB*S+B1**2*S2)/D
      HDR=HDRO*HR-HDI*HI
120   HDI=HDRO*HI+HDI*HR
130   IF(NCP.EQ.NP) GO TO 150
      DO 140 J=NCP1,NP
      HDRO=HDR
      HR=(1.+B(J)*C)/(1.+B(J))
      HI=-B(J)*S/(1.+B(J))
      HDR=HDRO*HR-HDI*HI
140   HDI=HDRO*HI+HDI*HR
150   HD=HDR**2+HDI**2
```

```
      HBR=(HNR*HDR+HNI*HDI)/HD
      HBI=(HDR*HNI-HNR*HDI)/HD
      IF(L.EQ.0)  RETURN
      IF(NZ.EQ.NZE) GO TO 200
      HR=HBR/(1.+A(NZ))
      HI=HBI/(1.+A(NZ))
      DR=1.+A(NZ)*C
      DI=-A(NZ)*S
      HRO=HR
      HR=HRO*(C-1.)+HI*S
      HI=-HRO*S+HI*(C-1.)
      DEN=DR**2+DI**2
      G1(NZ)=(HR*DR+HI*DI)/DEN
      G2(NZ)=(DR*HI-HR*DI)/DEN
200   IF(NZE.EQ.0)  GO TO 300
      DO 250 J=1,NZE,2
      DR=(1.+A(J)+A(J+1))*(1.+A(J)*C+A(J+1)*C2)
      DI=(1.+A(J)+A(J+1))*(-A(J)*S-A(J+1)*S2)
      HR=(1.+A(J+1))*C-1.-A(J+1)*C2
      HI=-(1.+A(J+1))*S+A(J+1)*S2
      GNR=HR*HBR-HI*HBI
      GNI=HR*HBI+HI*HBR
      DEN=DR**2+DI**2
      G1(J)=(GNR*DR+GNI*DI)/DEN
      G2(J)=(DR*GNI-DI*GNR)/DEN
      HR=(1.+A(J))*C2-1.-A(J)*C
      HI=-(1.+A(J))*S2+A(J)*S
      GNR=HR*HBR-HI*HBI
      GNI=HR*HBI+HI*HBR
      G1(J+1)=(GNR*DR+GNI*DI)/DEN
      G2(J+1)=(DR*GNI-DI*GNR)/DEN
250   CONTINUE
300   IF(NCP.EQ.0) GO TO 350
      DO 310 J=1,NCP,2
      B1=B(J+1)
      CB=COS(B(J))
      SB=SIN(B(J))
      F1=(1.+2.*B1*CB+B1**2)
      DR=F1*(1.+2.*B1*CB*C+B1**2*C2)
      DI=F1*(-2.*B1*CB*S-B1**2*S2)
      HI=-F1*2.*B1*SB*S-2.*B1*SB*(-2.*B1*CB*S-B1**2*S2)
      HR=F1*2.*B1*SB*C-2.*B1*SB*(1.+2.*B1*CB*C+B1**2*C2)
      GNR=HR*HBR-HI*HBI
      GNI=HR*HBI+HI*HBR
      DEN=DR**2+DI**2
      G1(J+NZ)=(GNR*DR+GNI*DI)/DEN
      G2(J+NZ)=(GNI*DR-GNR*DI)/DEN
      HI=F1*2.*(CB*S+B1*S2)-2.*(CB+B1)*(2.*B1*CB*S+B1**2*S2)
      HR=-F1*2.*(CB*C+B1*C2)+2.*(CB+B1)*(1.+2.*B1*CB*C+B1**2*C2)
      GNR=HR*HBR-HI*HBI
      GNI=HR*HBI+HI*HBR
      J1=J+1
      G1(J1+NZ)=ET*COS(ARG(J1+NZ))*(GNR*DR+GNI*DI)/DEN
      G2(J1+NZ)=ET*COS(ARG(J1+NZ))*(GNI*DR-GNR*DI)/DEN
```

```
310   CONTINUE
350   IF(NCP.EQ.NP) RETURN
      DO 370 J=NCP1,NP
      HR=1.-C
      HI=S
      DR=(1.+B(J))*(1.+B(J)*C)
      DI=(1.+B(J))*(-B(J)*S)
      GNR=HBR*HR-HBI*HI
      GNI=HR*HBI+HI*HBR
      DEN=DR**2+DI**2
      G1(J+NZ)=ET*COS(ARG(J+NZ))*(GNR*DR+GNI*DI)/DEN
      G2(J+NZ)=ET*COS(ARG(J+NZ))*(GNI*DR-GNR*DI)/DEN
370   CONTINUE
      RETURN
      END
```

APPENDIX B

It will be shown here that if $\tilde{H}(z)$ is an invariant simulation of $H(s)$ for the input form $x_a(t)$ and the input form $x_b(t)$, then $x_a(t) = x_b(t)$.

The continuous system output is uniquely determined by the convolution integral

$$y(t) = \int_0^t x(\tau)h(t - \tau)d\tau \tag{B1}$$

while the discrete system output is found from the sum

$$y_n = \sum_{k=0}^{n} x_k h_{n-k} \ . \tag{B2}$$

For invariant simulation

$$y_n = y(nT) \ , \quad n = 0, 1, 2, \ldots . \tag{B3}$$

with T simply specified to be a positive constant. Since $x_a(t)$ and $x_b(t)$ represent only a particular form of input, they may be chosen so that

$$x_a(nT) = x_b(nT) \ , \quad n = 0, 1, 2, \ldots . \tag{B4}$$

without loss of generality. Now from (B2) and (B4) we obtain

$$y_{an} = y_{bn} \ , \quad n = 0, 1, 2, \ldots . \tag{B5}$$

which, with (B3), requires

$$y_a(nT) = y_b(nT) \qquad n = 0, 1, 2, \ldots. \tag{B6}$$

thus,

$$\int_0^{nT} x_a(\tau)h(nT - \tau)d\tau = \int_0^{nT} x_b(\tau)h(nT - \tau)d\tau \tag{B7}$$

or

$$\int_0^{nT} \left[x_a(\tau) - x_b(\tau)\right]h(nT - \tau)d\tau = 0 \tag{B8}$$

for all n and T. Let $t = nT$ and (B8) becomes

$$\int_0^{t} \left[x_a(\tau) - x_b(\tau)\right]h(t - \tau)d\tau = 0 \tag{B9}$$

for all t. Except for the trivial case, $h(t) = 0$, (B9) is equivalent to

$$x_a(t) = x_b(t) \ , \qquad t \geq 0 \ . \tag{B10}$$

Thus, a simulation may be invariant to one, and only one, input form.

REFERENCES

1. Chu, Y., Digital Simulation of Continuous Systems, McGraw-Hill Book Co., New York, 1969.

2. Black, H. S., Modulation Theory, Van Nostrand Co. Inc., New York, 1953.

3. Goldman, S., Transformation Calculus and Electrical Transients, Prentice-Hall, Inc., New York, 1949.

4. Tustin, A., "A Method of Analyzing the Behaviour of Linear Systems in Terms of Time Series," J. of the IEEE, vol. 94, part IIA, pp. 130-142, 1947.

5. Kelly, L. G., Handbook of Numerical Methods and Applications, Addison-Wesley Publishing Co., Reading, Massachusetts, 1967, p. 247.

6. Rich, R. P. and H. Shaw, Jr., "An Application of Digital Filtering," Appl. Phys. Lab. Tech. Dig., vol. 4, no. 3, pp. 13-16, January-February 1965.

7. Shaw, H., "Discrete Analogs for Continuous Filters," J. of the ACM, vol. 13, no. 4, pp. 600-604, October 1966.

8. Boxer, R. and S. Thaler, "A Simplified Method of Solving Linear and Nonlinear Systems," Proc. of the IRE, vol. 44, no. 1, pp. 89-101, January 1956.

9. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Co., New York, 1962.

10. Lanczos, C., Applied Analysis, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1956.

11. Jordan, C., Calculus of Finite Differences, Chelsea, New York, 1960 (reprint of 1939 edition).

12. Milne-Thomson, L. M., The Calculus of Finite Differences, MacMillan and Co., Ltd., London, 1933.

13. Whittaker, E. T. and G. Robinson, The Calculus of Observations, Blackie and Son, Ltd., Glasgow, 1929.

14. James, H. M., N. B. Nichols and R. S. Phillips, Theory of Servomechanisms, Radiation Laboratory Series, vol. 25, McGraw-Hill Book Co., New York, 1947.

15. Jury, E. I., Sampled Data Control Systems, John Wiley and Sons, Inc., New York, 1958.

16. Kuo, B. C., Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.

17. Ragazzini, J. R. and G. F. Franklin, Sampled Data Control Systems, McGraw-Hill Book Co., New York, 1958.

18. Tou, J. T., Digital and Sampled Data Control Systems, McGraw-Hill Book Co., New York, 1959.

19. "IEEE Transactions on Audio and Electroacoustics," Special Issue on Digital Filters, vol. AU-16, no. 3, September 1968.

20. "IEEE Transactions on Audio and Electroacoustics," Special Issue on Digital Filtering, vol. AU-18, no. 2, June 1970.

21. "IEEE Transactions on Audio and Electroacoustics," Special Issue on Digital Signal Processing, vol. AU-18, no. 4, December 1970.

22. Aseltine, J. A., Transform Method in Linear System Analysis, McGraw-Hill Book Company, New York, 1958.

23. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, Chapt. 2, 1969.

24. Jury, E. I., Theory and Application of the z-Transform Method, John Wiley and Sons, Inc., New York, 1964.

25. Doetsch, G., Guide to the Applications of the Laplace and z-Transforms, Van Nostrand Reinhold Co., London, 1971.

26. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, 1969.

27. Rader, C. M.,"On Digital Filtering," IEEE Trans. Audio Electroacoust., vol. AU-16, no. 3, pp. 303-314, September 1968.

28. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, p. 39, 1969.

29. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, p. 41, 1969.

30. Stearns, S. D., E.E. 695, Spring Semester 1971, University of New Mexico, Class Notes.

31. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York p. 28, 1969.

32. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, Chapt. 4, 1969.

33. Corrington, M. S.,"Sensitivity Factors for Digital Filters," IEEE Conference-Digital Filtering, The State of the Art, Newark, January 1970, pp. 28-36.

34. Gold, B. and C. M. Rader, "Effects of Quantization Noise in Digital Filters," Spring Joint Computer Conf., vol. 28, pp. 213-219, Spartan, Washington D.C., 1966.

35. Halyo, N. and G. A. McAlpine, "A Discrete Model for Product Quantization Errors in Digital Filters, IEEE Trans. Audio Electroacoust.," vol. 19, no. 3, pp. 255-256, September 1971.

36. Knowles, J. B. and R. Edwards, "Effect of Finite Word-Length Computer in a Sampled-Data Feedback System," Proc. IEE, vol. 112, no. 6, pp. 1197-1207, June 1965.

37. Knowles, J. B. and E. M. Olcayto, "Coefficient Accuracy and Digital Filter Response," IEEE Trans. Circuit Theory, vol. CT-15, no. 1, pp. 31-41, March 1968.

38. Oppenheim, A. V., "Realization of Digital Filters Using Block Floating Point Arithmetic," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 130-136, June 1970.

39. Otnes, R. K. and L. P. McNamee, "Instability Thresholds in Digital Filters due to Coefficient Rounding," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 4, pp. 456-463, December 1970.

40. White, S. A., "Minimizing Word Lengths for Recursive Digital Filters, Computers and Communications Conference," Rome, New York, pp. 175-187, 1969.

41. Karni, S., Network Theory: Analysis and Synthesis, Allyn and Bacon, Boston, 1966, p. 114.

42. Gold, B. and K. L. Jordan, Jr., "A Note on Digital Filter Synthesis," IEEE Proc., pp. 1717-1718, October 1968.

43. Voelker, H. B. and E. E. Hartquist, "Digital Filtering via Block Recursion," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 169-176, June 1970.

44. Kaiser, J. F., "Digital Filters" in System Analysis by Digital Computer, J. F. Kaiser and F. Kuo, Editors, John Wiley and Sons, Inc., New York, 1966.

45. Kaiser, J. F., "The Digital Filter-Its Implementation and Application," IEEE Conference-Digital Filtering, The State of the Art, Newark, January 1970, pp. 28-36.

46. Rader, C. M. and B. Gold, "Digital Filter Design Techniques in the Frequency Domain," Proc. IEEE, vol. 55, no. 2, pp. 149-171, February 1967.

47. Rabiner, L. R., "Techniques for Designing Finite-Duration Impulse-Response Digital Filters," IEEE Trans. Commun. Technol., vol. COM-19, no. 2, pp. 188-195, April 1971.

48. Helms, H. D., "Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters," IEEE Trans. Audio Electroacoust., vol. AU-15, no. 2, pp. 85-90, June 1967.

49. Stockham, T. G., Jr., "High Speed Convolution and Correlation," Proc. Spring Joint Computer Conference, 1966, vol. 28, pp. 229-233.

50. Helms, H. D., "Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response," IEEE Trans. Audio Electroacoust., vol. AU-16, no. 3, pp. 336-342, September 1968.

51. Rabiner, L. R., B. Gold and C. A. McConegal, "An Approach to the Approximation Problem for Digital Filters," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 83-106, June 1970.

52. Rabiner, L. R. and R. W. Schafer, "Recursive and Nonrecursive Realizations of Digital Filters Designed by Frequency Sampling Techniques," IEEE Trans. Electroacoust., vol. AU-19, no. 3, pp. 200-207, September 1971.

53. Herrmann, O., "Design of Nonrecursive Digital Filters with Linear Phase," Electron. Lett., vol. 6, no. 11, pp. 328-329, May 1970.

54. Parks, T. W. and J. H. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," IEEE Trans. Circuit Theory, vol. CT-19, no. 2, pp. 189-194, March 1972.

55. Gold, B. and K. L. Jordan, Jr., "A Direct Search Procedure for Designing Finite Duration Impulse Response Filters," IEEE Trans. Audio Electroacoust., vol. AU-17, no. 1, pp. 33-36, March 1969.

56. Helms, H. D., "Digital Filters with Equiripple or Minimax Responses," IEEE Trans.Audio Electroacoust., vol. AU-19, no. 1, pp. 87-93, March 1971.

57. Tufts, D. W. and J. T. Francis, "Designing Digital Low-Pass Filters-Comparison of Some Methods and Criteria," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 4, pp. 487-494, December 1970.

58. Tufts, D. W., D. W. Rorabacher and W. E. Mosier, "Designing Simple Effective Digital Filters," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 142-158, June 1970.

59. Nelson, G. A., L. L. Pfeifer and R. C. Wood, "High-Speed Octave Band Digital Filtering," IEEE Trans. Audio Electroacoust., vol. AU-20, no. 1, pp. 58-65, March 1972.

60. Gold, B. and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Co., New York, p. 95, 1969.

61. Constantinides, A. G., "Family of Equiripple Lowpass Digital Filters," Electron. Lett., vol. 6, no. 11, May 1970.

62. Holtz, H. and C. T. Leondes, "The Synthesis of Recursive Digital Filters," J. ACM, vol. 13, no. 2, pp. 262-280, April 1966.

63. Otnes, R. K., "An Elementary Design Procedure for Digital Filters," IEEE Trans. Audio Electroacoust., vol. AU-16, no. 3, pp. 330-335, September 1968.

64. Shanks, J. L., "Recursion Filters for Digital Processing," Geophysics, vol. XXXII, no. 1, pp. 33-51, February 1967.

65. Shanks, J. L., "Recursion Filters for Digital Processing," Geophysics, vol. XXXII, no. 1, pp. 33-51, February 1967.

66. Gibbs, A. J., "On the Frequency Domain Response of Causal Digital Filters," Ph.D. dissertation, University of Wisconsin, Madison, January 1967.

67. Burrus, C. S. and T. W. Parks, "Time Domain Design of Recursive Digital Filters," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 137-141, June 1970.

68. DeFigueiredo, R. J. P. and A. N. Netravali, "Optimal Spline Digital Simulators of Analog Filters," IEEE Trans. Circuit Theory, vol. CT-18, no. 6, pp. 711-717, November 1971.

69. Corrington, M. S., "Improving the Accuracy and Speed of Digital Filters," Proc. 2nd. Biennial Cornell Conf. Computerized Electronics, pp. 293-304, 1969.

70. Harrison, S. R. and B. J. Leon, "Digital Filters for Approximating Continuous Convolutions," IEEE Trans. Circuit Theory, vol. CT-18, no. 6, pp. 743-745, November 1971.

71. Boxer, R., "A Note on Numerical Transform Calculus," Proc. of the IRE, vol. 45, no. 10, pp. 1401-1406, October 1957.

72. Gibson, J. E., Nonlinear Automatic Control, McGraw-Hill Book Co., New York, 1963, p. 147.

73. Golden, R. M. and J. F. Kaiser, "Design of Wideband Sampled Data Filters," Bell System Technical Journal, vol. 43, no. 4, part 2, pp. 1533-1546, 1964.

74. IBM Manual E20-0029-1; Numerical Techniques for Real Time Digital Flight Simulation.

75. Wait, J. V., "Digital Filters" in Active Filters: Lumped, Distributed, Integrated, Digital, and Parametric; L. P. Huelsman, Editor, McGraw-Hill Book Co., New York, 1970.

76. Wait, J. V., "State Space Methods for Designing Digital Simulations of Continuous Fixed Linear Systems," IEEE Trans. on Electronic Computers, vol. EC-16, no. 3, pp. 351-354, June 1967.

77. Fowler, M. E., "A New Numerical Method for Simulation," Simulation, vol. 5, no. 5, pp. 324-330, May 1965.

78. Weast, R. C., Handbook of Tables for Mathematics, The Chemical Rubber Co., Cleveland, 1969.

79. Fleischer, P. E., "Digital Realization of Complex Transfer Functions," Simulation, vol. 6, no. 3, pp. 171-180, March 1966.

80. Steiglitz, K., "Computer Aided Design of Recursive Digital Filters," IEEE Trans. Audio Electroacoust., vol. AU-18, no. 2, pp. 123-129, June 1970.

81. Fletcher, R. and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer Journal, vol. 6, no. 2, pp. 163-168, 1963.

82. IBM Manual GH20-0205-4, System/360 Scientific Subroutine Package, Version III, 1970.

83. Athanassopoulos, J. A. and A. D. Waren, "Design of Discrete Time Systems by Mathematical Programming," Proc. 1968 Hawaii Int. Conf. Syst. Sci., University of Hawaii Press, Honolulu, 1968, pp. 224-227.

84. Helms, H. D., "Designing Digital Filters with Constraints," IEEE Conference-Digital Filtering, The State of the Art, Newark, January 1970, pp. 37-55.

85. Seshu, S. and N. Balabanian, Linear Network Analysis, John Wiley and Sons Inc., New York, 1959, p. 41.

86. Lawrence, J. P. III and K. Steiglitz, "Randomized Pattern Search," IEEE Trans. Computers, vol. C-21, no. 2, pp. 382-385, April 1972.

87. Thrall, R. M. and L. Tornheim, Vector Spaces and Matrices, John Wiley and Sons, New York, 1957, p. 168.

CURRICULUM VITAE

The author was born in Elwood, Nebraska on September 6, 1943. He attended public schools there and graduated from Elwood High School in 1961. During the next four years he attended the University of Nebraska. In 1965 he graduated with high distinction, receiving a Bachelor of Science degree in Electrical Engineering. He received a Master of Science degree in Electrical Engineering from the University of New Mexico in 1967, at the conclusion of his participation in the Technical Development Program at Sandia Laboratories. During the school year 1971-72, he completed his residency requirement at the University of New Mexico under Sandia's Doctoral Support Program.

Since June 1965, he has been employed at Sandia Laboratories, Albuquerque, New Mexico.

In February 1969, he was married to the former Miriam Griswold of Albuquerque.

The author is a member of Sigma Tau, Eta Kappa Nu, Pi Mu Epsilon, Sigma Xi, and IEEE.