

T125
10

To the Graduate Council:

I am submitting herewith a thesis written by Chun-Yao Lien entitled "Linear Techniques for Dynamic Chemical Process Simulation and Direct Digital Control Algorithm Implementation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Chemical Engineering.

Duane D. Bruns

Duane D. Bruns, Major Professor

We have read this thesis
and recommend its acceptance:

Charles F. Moore

Frank W. Allen

Jack Fowler

Accepted for the Council:

Leuminkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this may be granted by my major professor, or in his absence, by the Head of Interlibrary Service when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Chun Yao Lien

Date June 13, 1986

LINEAR TECHNIQUES FOR DYNAMIC CHEMICAL PROCESS
SIMULATION AND DIRECT DIGITAL CONTROL
ALGORITHM IMPLEMENTATION

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Chun-Yao Lien

August 1986

DEDICATION

This thesis is dedicated to the author's parents,
Mr. George Lien, Mrs. Yue-Suan Lien, and his wife
Ya-Chwen Chen.

ACKNOWLEDGMENTS

The author is grateful for Dr. Duane Bruns, his advisor, for his assistance during the author's study at UTK, especially for his help to complete the thesis. He is also indebted to Professors Jack Lawler, Charles Moore, and Fred Webber, who served as his committee.

ABSTRACT

Digital simulation algorithms are studied with a goal of establishing guidelines to select accurate methods for simulating the dynamic response of linear continuous models. Linear models play a large role in many simulation applications, especially in applied industrial process simulation and control. To date no in depth study can be found in the literature where digital simulation techniques are surveyed and compared to provide any guidelines.

One major problem in digital computer simulation is the conversion of a process oriented model to a digital computer oriented model. In other words, algorithms are needed to convert the continuous process equations into difference equations which can be eventually expressed as recursive formulas and implemented in digital computers. Numerical integration schemes are commonly and widely used; however, with respect to linear models these methods may be unnecessary and inefficient. Moreover, the requirement of relatively small simulation step size to keep the numerical integration schemes stable may limit their applications. A major concept in this work is to capitalize on the use of methods that take advantage of the models' linearity.

Six optimal discretization methods were applied to the simulation of three transfer function models under the criteria of best matching the time domain transient response. This study indicated the Taylor method, a relatively new technique, provides perhaps the best discretization algorithm in the survey. Also two input approximation methods, one was the popular zero-order hold with phase compensation, gave good results.

Also presented is a new procedure for calculating the discrete transition state matrix, P , and discrete input matrix, Q , which are needed to solve state space models. The new procedure formulates an augmented matrix whose matrix exponential provides both P and Q even when the state transition matrix is singular.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
II. LITERATURE REVIEW	5
2.1. Overview of Modeling	5
2.2. Review of Numerical Integration	12
2.2.1. Euler's Method	14
2.2.2. Trapezoid Method	14
2.2.3. Runge-Kutta Methods	17
2.3. Development of Discrete Transfer Functions	18
2.3.1. Overview of Sampled-Data Theory	19
2.3.2. Sampling Process	19
2.3.3. Z-Transform	22
2.3.4. Modified Z-Transform	24
2.3.5. Relationship Between Z-Trans- forms with Laplace-Transforms..	25
2.3.6. Sampling Theorem	26
2.4. Discretization of Continuous Transfer Functions	30
2.4.1. Substitution Method	30
2.4.2. Matched Z-Transform	34
2.4.3. Input Approximation	38
2.4.4. Zero-Order Hold	39
2.4.5. Triangular Hold	41
2.5. Summary	44

CHAPTER	PAGE
III. LINEAR SYSTEM SIMULATION	45
3.1. Discrete Approximation	45
3.2. Linear System Simulation Formulas	48
3.3. Pulse Transfer Function	49
3.4. Development of Simulation Formulas for Overall Input-Output Description	50
3.4.1. Step Invariant Discrete Transfer Function	51
3.4.2. Ramp Invariant Discrete Transfer Function	51
3.5. Simulation Formulas Based on State Space Descriptions	54
3.5.1. Step Invariant Discrete State Space Equation	58
3.5.2. Ramp Invariant Discrete State Space Equation	59
3.6. Matrix Exponential Computation	60
3.6.1. Taylor Series Method	60
3.6.2. Pade' Approximation	61
3.6.3. Companion Matrix Method	64
3.6.4. Jordan Canonical Matrix Method	67
3.6.5. Q Matrix Calculation	69
3.7. Stability and Accuracy Analysis	71
3.7.1. Input Approximation Methods ...	72
3.7.2. Classical Numerical Integration Methods	74
3.8. Summary	82

CHAPTER	PAGE
IV. COMPARISON OF SIMULATION TECHNIQUES WITH ARBITRARY INPUTS	84
4.1. Optimal Discrete Models and Integrators	85
4.2. Overall Redesign	87
4.3. Optimal Discretization Methods	88
4.3.1. Development of the Taylor Method	89
4.4. Comparison of Discretization Methods ..	100
4.4.1. Comparison Criteria	101
4.4.2. Selected Discretization Methods and Linear Models	104
4.5. Comparison I	105
4.5.1. Tustin Method	106
4.5.2. Match Pole-Zero Method with Infinite Zeroes Mapped onto $z = 0$	107
4.5.3. Match Pole-Zero Method with Infinite Zeroes Mapped onto $z = -1$	108
4.5.4. Input Approximation Method with Zero-Order Hold and Adjustable Phase Compensator	109
4.5.5. Input Approximation Method with Triangular Hold	112
4.5.6. Taylor Method	112
4.5.7. General Form of Discrete Transfer function for First Order Lag	115
4.5.8. Results of Comparison I	118
4.6. Comparison II	136

CHAPTER	PAGE
IV. (Continued)	
4.7. Comparison III	150
4.8. Summary	172
V. GUIDELINES, CONCLUSIONS AND FUTURE STUDIES ..	178
5.1. Linear Model Simulation	178
5.2. Optimal Discretization	179
5.3 Simulation Study for Arbitrary Inputs .	180
5.4 Simulation for Arbitrary Input: Conclusions and Guildeline	180
5.5 Simulation Method for Invariant Inputs	
5.6 New Matrix Exponential Algorithm	187
5.7 Future Studies	188
5.8 Closely Related Topics	188
5.9 Other Toppoics	189
LIST OF REFERENCES	190
VITA	196

LIST OF TABLES

TABLE	PAGE
2.1 Discrete Integrators for Substitution Method	36
3.1. Step Invariant Discrete Transfer Function ...	52
3.2. Ramp Invariant Discrete Transfer Function ...	55
4.1. Coefficients of the Discrete Model In Comparison I Synthesized by Taylor Method ...	116
4.2. Coefficients of Discrete Models in Comparison I	117
4.3. List of Discretization Methods	125
4.4. Coefficients of Discrete Models in Comparison II	145
4.5. Coefficients of Discrete Models in Comparison III	169

LIST OF FIGURES

FIGURE	PAGE
2.1. Flow Diagram of Simulation Design Process ..	6
2.2. Block Diagram System Description	13
2.3. Rectangular Rule for Numerical Integration .	15
2.4. Trapezoid Rule for Numerical Integration ...	15
2.5. Relation Between Numerical Approximation and Sampled-Data Transformation	20
2.6. Input and Output Signals of the Sampler	23
2.7. Frequency Domain Characteristics of Sampled Signal	28
2.8. Mapping Relationship of S- and Z-plane by Bilinear Transformation	33
2.9. Frequency Distortion Due to the Bilinear Transformation	35
2.10. The Process of Synthesizing a Discrete- Representation of Continuous System	40
2.11. Time and Frequency Domain Characteristics of Zero-Order Hold	42
2.12. Time and Frequency Domain Characteristics of Triangular Hold	43
3.1. (a) A Continuous Transfer Function (b) An Approximating Discrete Transfer Function	47
3.2. Digital Methods for Solving the State Space Equation (a) Input Approximation Method (b) Numerical Integration Method	73
3.3. Discrete Integrator	76
3.4. Frequency Response of Discrete Integrator. A—Continuous Integrator and the Numerical Value indicated in Each Curve Associated the Compensated Phase Magnitude r	79

FIGURE	PAGE
3.5. The Magnitude and Phase Distortion of Discrete Integrators	81
4.1. System Configuration of Input Approximation Method	111
4.2. Frequency Response of 1st-Order System with Input Approximation Method with Zero-Order Hold and Various Phase Compensation ($T = 0.1$)	120
4.3. EMAG and EPHAS Plots of Input Approximation Method with Zero-Order Hold and Various Phase Compensation ($T = 0.1$)	121
4.4. MAG and PHAS Plots of 1st-Order System ($T = 0.1$, $r = 0.5$)	123
4.5. EMAG and EPHAS Plots of 1st-Order System ($T = 0.1$, $r = 0.479$)	124
4.6. EMAG and EPHAS Plots of 1st-Order System ($T = 0.5$, $r = 0.4895$)	126
4.7. EMAG and EPHAS Plots of 1st-Order System ($T = 0.02$, $r = 0.4957$)	127
4.8. ERR and NERR Plots of 1st-Order System ($T = 0.1$, $NSC = 8$, $r = 0.479$)	130
4.9. ERR and NERR Plots of 1st-Order System ($T = 0.1$, $NSC = 32$, $r = 0.479$)	131
4.10. ERR and NERR Plots of 1st-Order System ($T = 0.02$, $NSC = 8$, $r = 0.4957$)	132
4.11. ERR and NERR Plots of 1st-Order System ($T = 0.02$, $NSC = 32$, $r = 0.4957$)	133
4.12. Frequency Response of Comparison II System with Tustin Methods. (P) with Frequency Prewarping at W_p . (U) Without Frequency Prewarping	139
4.13. Frequency Response of the Input Approximation Method with Zero-Order Hold and Various Phase Compensator for Comparison II. ($T = 0.1$)	140

FIGURE	PAGE
4.14. EMAG and EPHAS Plots of the Input Approximation Method with Zero-Order Hold and Various Phase Compensation for Comparison II. (T = 0.1)	141
4.15. MAG and PHAS Plots of Comparison II System (T = 0.1 , r = 0.5)	143
4.16. EMAG and EPHAS Plots of Comparison II System (T = 0.1, r = 0.5)	144
4.17. ERR and NERR Plots of Comparison II System (T = 0.1, NSC = 8, r = 0.5)	147
4.18. ERR and NERR Plots of Comparison II System (T = 0.1, NSC = 32, r = 0.5)	148
4.19. Step Response of Comparison III System (a) T=0.03 and r=0.5 (b) T=0.5 and r=0.5, the Dash Line Indicates the Response of the Continuous System	151
4.20. MAG and PHAS Plots of Comparison III System with Taylor Method (T = 0.167)	154
4.21. EMAG and EPHAS Plots of Comparison III System with Taylor Method (T = 0.167)	155
4.22. EMAG and EPHAS Plots of Comparison III System with Taylor Method (T = 0.167)	156
4.23. ERR and NERR Plots of Comparison III System with Taylor Method (T = 0.167, NSC = 8)	157
4.24. ERR and NERR Plots of Comparison III System with Taylor Method (T = 0.167, NSC = 16) ...	158
4.25. ERR and NERR Plots of Comparison III System with Taylor Method (T = 0.167, NSC = 32) ...	159
4.26. Frequency Response of Comparison III System with Input Approximation with Zero-Order Hold and Various Phase Compensator (T = 0.167)	163

FIGURE	PAGE
4.27. EMAG and EPHAS Plots of Comparison III System Input Approximation with Zero-Order Hold and Various Phase Compensator ($T = 0.167$)	164
4.28. MAG and PHAS Plots of Comparison III System ($T = 0.167$, $r = 0.5$)	165
4.29. EMAG and EPHAS Plots of Comparison III System ($T = 0.167$, $r = 0.5$)	166
4.30. EMAG and EPHAS Plots of Comparison III System ($T = 0.167$, $r = 0.46$)	167
4.31. EMAG and EPHAS Plots of Comparison III System. ($T = 0.03$, $r = 0.49$)	168
4.32. ERR and NERR Plots of Comparison III System ($T = 0.167$, $NSC = 8$, $r = 0.5$)	173
4.33. ERR and NERR Plots of Comparison III System ($T = 0.167$, $NSC = 32$, $r = 0.5$)	174
4.34. ERR and NERR Plots of Comparison III System ($T = 0.03$, $NSC = 8$, $r = 0.49$)	175
4.35. ERR and NERR Plots of Comparison III System ($T = 0.03$, $NSC = 32$, $r = 0.49$)	176

CHAPTER I

INTRODUCTION

The expanding application of digital techniques to dynamic process simulation and direct digital control algorithm implementation has made the development and evaluation of these techniques an important subject area. In the last decade, the great advancement of digital computer technology and the dropping cost to computation capability coupled with the increasing cost of raw materials and labor have made applications of digital process simulation and control to industrial processes not only a possibility, but in competitive situations it is becoming a necessity. Utilization of digital simulation methodologies is occurring in all computer equipment environments ranging from micro-computers, super-micro-computers programmable controllers to main frame computers.

The linear or linearized mathematical model plays a large role in applied industrial process simulation and control. This research specifically addresses the various techniques available for linear model simulation. Digital algorithms are studied with the goal to provide guidelines in selecting accurate methods for the dynamic behavior of linear continuous systems.

A major concept in this work is to utilize digital simulation techniques which are built on the fact that the models are linear. By taking advantage of the models' linearity, the simulation development and application may take less time and effort, need less computer memory, use less computation time and be implemented in smaller computer system environments.

Motivations for linear process simulation are wide ranging but do include the need to conduct low cost studies or designs for a system to verify a system's mathematical model, to build teaching and training simulators, to forecast the performance of a system to complex policies to evaluate policy alternatives, and to implement advanced control strategy algorithms.

The selection of a proper simulation algorithm depends heavily on the system's mathematical model, on the specific application, and on the computer environment in which it is implemented. The "universal" algorithm for linear model simulation is unavailable, in its place there is a multitude of methodologies. So far no in depth study can be found in the literature in which these methods for designing digital simulations are surveyed, and compared, so that one can easily, and comfortably select a proper technique for their application. The lack of such guidelines frequently

leads to employing only the techniques with which one is familiar. Sometimes this might not only lead to an inefficient but also costly simulation, at best, and inaccuracy results, at worst.

A common situation is the use of Runge-Kutta routines which does not explicitly take advantage of the fact that the model being simulated is linear. For example, CSMP or ACSL macros are widely used to implement transfer function block diagram simulation. The macros convert the block diagrams back into linear ODE's and then integrate using a classical numerical scheme as the 4-th order Runge Kutta.

The stated objective of this research is to provide guidelines to select appropriate and accurate methods for linear model simulation. The emphasis throughout is on applications that are immediately and directly useful to anyone involved in digital simulation and its implementation.

The next chapter reviews the fundamental literature to this research which includes sampled data transformations, z-transforms, and numerous discretization methods. Chapter III derives simulation formulations for predetermined inputs; these formulas can be applied to either signal input signal output systems or to multiple input multiple output systems.

Algorithms for matrix exponential computation are discussed and a Fortran program for calculating matrix exponential is developed promising. In Chapter IV, a recent and promising discretization method developed from truncating Taylor series expansion is compared with numerous existing methods from both frequency and time domain analysis. Final guidelines, conclusions and recommendations for future research are summarized in Chapter V.

CHAPTER II

LITERATURE REVIEW

A schematic diagram of a simulation design procedure is shown in Figure 2.1. At least four distinct steps are identified in the diagram. First a mathematical description of the system is prepared; in general, the description at this stage is a processed-oriented continuous model which needs to be converted to a computer-oriented model, e.g. a numerical solution procedure. Second an algorithm is selected to convert the model from the process environment to the computer environment. Third a computer program to implement the computer-oriented model is developed. Finally verification and checkout of the simulation is required.

2.1 Overview of Modeling

In general, physical processes are often too complex to be extensively studied experimentally. In addition experimental studies are usually costly and very time consuming. In order to analyze the dynamic behavior of a process, we often resort to process simulation, thus the need for a mathematical description to describe the physical and/or chemical phenomena

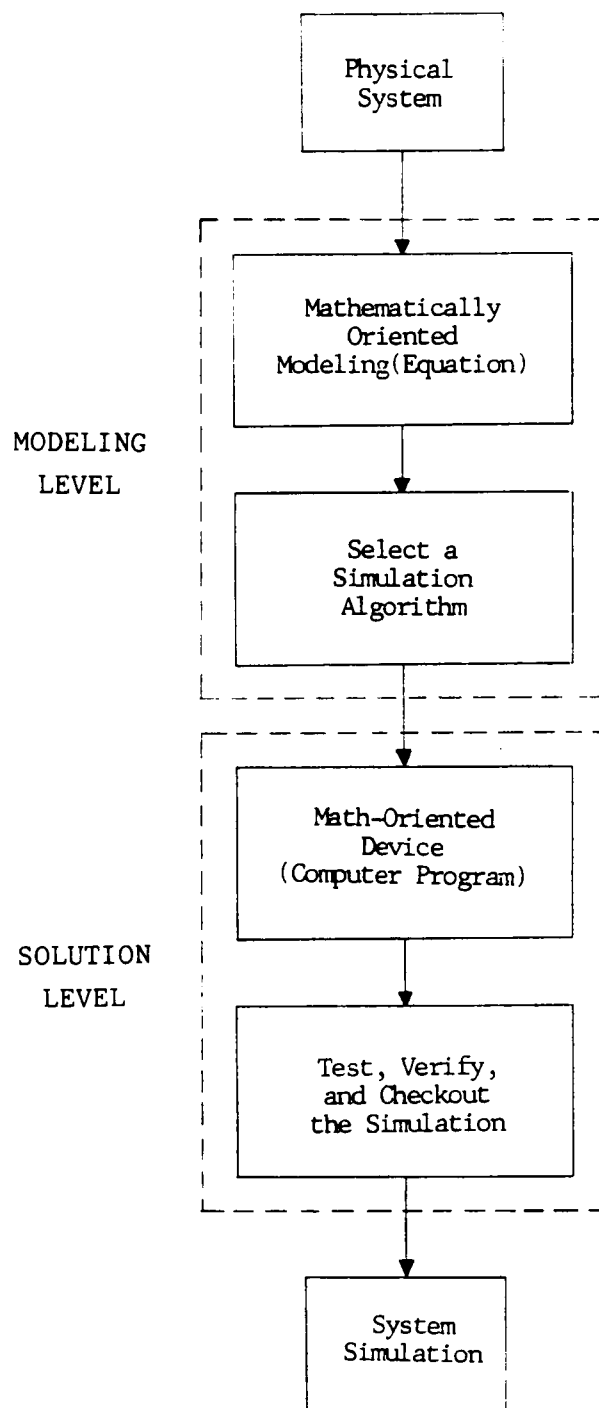


Figure 2.1. Flow Diagram of Simulation Design Process

taking place in the process. Such a mathematical description constitutes a process model.

Process models can be developed by two distinct formalisms, the theoretical and experimental approaches. Theoretical modeling consists of systematically applying basic physical laws to system components and to the interconnection of these components to establish a process model. Models resulting from this approach are usually represented in the time domain as a set of differential and algebraic equations. Experimental modeling is built around selecting a set of mathematical relationships and fitting parameters in these relationships using input-output data. Time and frequency domain techniques are used in selecting the best parameters, thus models arising from this approach may be either in the time domain or frequency domain.

A system model will have various forms due to the different methods used in establishing the model, or because of the different questions and concerns the investigator has for the process. A transfer function that describes only the final output properties may be called an external or input-output description. A set of differential equations that describe the internal as well as output behavior may be called an internal or state-variable description.

From a simulation point of view, Process models can be categorized into three types of simulation-oriented models.

1. Systems that are described by a state space model:

$$\begin{aligned}\dot{X}(t) &= A X(t) + B U(t) \\ Y(t) &= C X(t) + D U(t)\end{aligned}\tag{2.1}$$

where: $X(t)$: is a $n \times 1$ state vector.

$U(t)$: is a $p \times 1$ input vector.

$Y(t)$: is a $q \times 1$ output vector.

A : is a $n \times n$ state transition matrix.

B : is a $n \times p$ state input matrix.

C : is a $q \times n$ state output matrix.

D : is a $q \times p$ feedforward matrix.

In general, a n th-order linear dynamic system is described by a n th-order linear ODE (Ordinary Differential Equation) which can be rewritten as n first-order linear ODEs by defining new variables. The set of first-order linear ODEs can be expressed in the state space form. For illustration, consider a third-order linear ODE:

$$\frac{d^3 y}{dt^3} + a_2 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = b_0 u\tag{2.2}$$

If the following new variables are defined:

$$\begin{aligned}x_1 &= y \\x_2 &= \frac{dy}{dt} \\x_3 &= \frac{d^2y}{dt^2}\end{aligned}\tag{2.3}$$

then, Equation 2.2 can be decomposed into three first-order linear ODEs

$$\begin{aligned}\frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= x_3 \\ \frac{dx_3}{dt} &= b_0 u - a_2 x_3 - a_1 x_2 - a_0 x_1\end{aligned}\tag{2.4}$$

These 1st-order ODEs can be expressed in matrix form

$$\dot{\mathbf{Y}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [u] \quad (2.6)$$

More general relationships between state space equations and high-order differential equations can be found in some material (21,22). Numerical integration methods are commonly used to solve such equations.

2. Systems that are described by overall external models

For the single-input single-output case:

$$\frac{Y(s)}{U(s)} = G(s) \quad (2.7)$$

For the multiple-input multiple-output case:

$$Y = G U$$

where:

$$G = \begin{bmatrix} G_{11}(s), \dots, G_{1p}(s) \\ G_{q1}(s), \dots, G_{qp}(s) \end{bmatrix} \quad (2.8)$$

$$Y = [Y_1(s), \dots, Y_q(s)]^T$$

$$U = [U_1(s), \dots, U_p(s)]^T$$

$Y_i(s)$: i-th output

$U_j(s)$: j-th input

$G_{ij}(s)$: the transfer function relates the
i-th output and j-th input.

Discretization techniques can convert continuous transfer functions to discrete transfer functions, then the continuous system's dynamics are described by "equivalent" difference equations. Using difference equations, the system's dynamics can be simulated by recursive formulas. Alternatively, a continuous transfer function can be decomposed by various decomposition schemes (21) and ultimately expressed as state space equations.

State space models can be converted to overall external models using the Laplace transform. Thus either category of model can be reformulated into the other category. To illustrate taking the Laplace transform of Equation 2.1 gives

$$s X(s) = A X(s) + B U(s)$$

$$Y(s) = C X(s) + D U(s)$$

Solving for $Y(s)$ in terms of $U(s)$ yields

$$\begin{aligned} Y(s) &= (sI - A)^{-1} B U + D U \\ &= [(sI - A)^{-1} B + D] U \end{aligned}$$

Noting that the above equation gives the process output in terms of the input provides

$$\begin{aligned} Y(s) &= G(s) U(s) \\ \text{where } G(s) &= [(sI - A)^{-1} B + D] \end{aligned}$$

3. Systems described by individual transfer function blocks corresponding to each physical component or part of the process. Figure 2.2. illustrates such a systems.

2.2 Review of Numerical Integration

Continuous systems are those whose motions are characterized by continuous equations. To investigate the dynamic behavior of continuous systems involves the solution of a set of differential and algebraic equations. The principle difficulty in solving such equations on digital computers is integration, which must be done numerically. Limitations on the accuracy and stability of digital numerical integration schemes can cause many problems. Numerous algorithms are

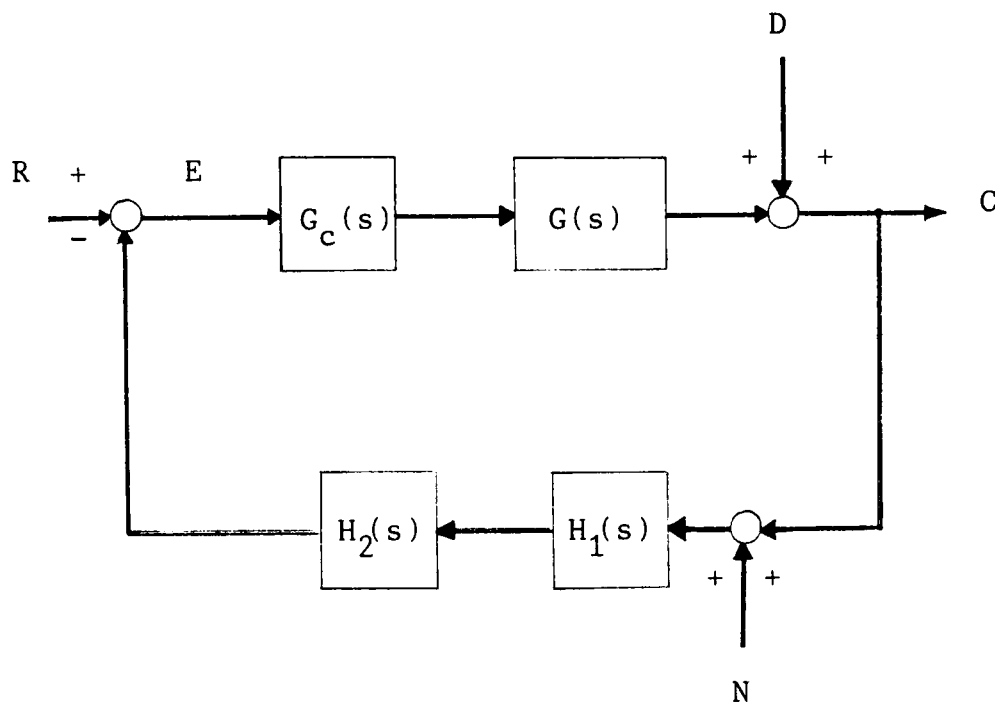


Figure 2.2. Block Diagram System Description

available for digital numerical integration. The Euler, Trapezoid, and Runge-Kutta methods are reviewed and discussed in this research. These methods treat nonlinear as well as linear ODEs. The formulation of each method is illustrated here in terms of a differential equation

$$\frac{dx}{dt} = f(t,x) \quad (2.9)$$

2.2.1 Euler's Method

Euler's method (7,14,31) approximates the integration with the rectangular rule. It provides an explicit numerical integration formula, see Figure 2.3,

$$x(kT+T) = x(kT) + T \dot{x}(kT) \quad (2.10)$$

where T is a uniform step size.

By substituting Equation 2.9 into the above formula for $x(kT)$ and starting from the initial condition on x , the system's response can be computed iteratively.

2.2.2 Trapezoid Method

Trapezoidal method (14) approximates the integration with the trapezoidal rule, Figure 2.4, and

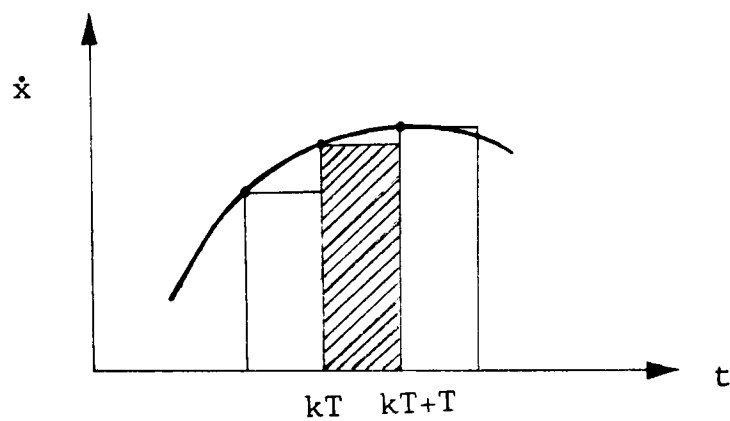


Figure 2.3. Rectangular Rule for Numerical Integration

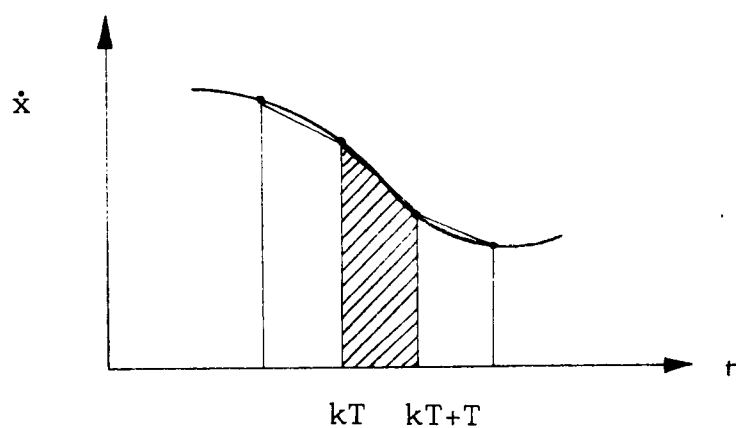


Figure 2.4. Trapezoid Rule for Numerical Integration

gives an implicit integration formula:

$$x(kT+T) = x(kT) + \frac{T}{2} [\dot{x}(kT) + \dot{x}(kT+T)] \quad (2.11)$$

This numerical integration formula includes the $x(kT+T)$ term on the right hand side as well as on the left hand side, thus, to calculate $\dot{x}(kT+T)$ requires knowing $x(kT+T)$. Two approaches for solving this problem will now be presented.

1. If the system is linear, substituting for the term $\dot{x}(kT+T)$ according to the system's equation, Equation 2.9, into Equation 2.11 allows the term $x(kT+T)$ to be solved analytically.

2. Prediction-Corrector Algorithms:

A different rule, commonly the rectangular rule is first used, to predict $x(kT+T)$. The predicted value, for $x(kT+T)$ is then used in Equation 2.9 to obtain $\dot{x}(kT+T)$. Then substitute this value into trapezoidal rule again to correct the value of $x(kT+T)$. The predictor-corrector procedure is summarized by the following steps:

$$\begin{aligned}
(1) \quad \dot{x}(kT) &= f(x(kT), kT) \\
(2) \quad x(kT+T) &= x(kT) + T \dot{x}(kT) \\
(3) \quad \dot{x}(kT+T) &= f(x(kT+T), kT+T) \\
(4) \quad x(kT+T) &= x(kT) + \frac{T}{2} (\dot{x}(kT+T) + \dot{x}(kT))
\end{aligned} \tag{2.12}$$

In Equation 2.12, step (2) is the predictor step and step (4) is corrector step.

2.2.3 Runge-Kutta Methods

Runge-Kutta methods (7,14) are perhaps the most popular and widely used numerical integration techniques. Runge-Kutta methods are based on implicitly developing increasingly higher orders of the Taylor series expansion for the function being integrated. The higher order Taylor series are developed using combination of the function's derivatives numerically evaluated over certain intervals of the dependent (integration) variable.

The forth-order Runge-Kutta is one of the most common forms presently in use. The algorithm is

$$x_{n+1} = x_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \tag{2.13}$$

where:

$$\begin{aligned}
 k_1 &= T f(x_n, t_n) \\
 k_2 &= T f(x_n + \frac{1}{2}k_1, t_n + \frac{1}{2}T) \\
 k_3 &= T f(x_n + \frac{1}{2}k_2, t_n + \frac{1}{2}T) \\
 k_4 &= T f(x_n + k_3, t_{n+1})
 \end{aligned}
 \tag{2.14}$$

2.3 Development of Discrete Transfer Functions

A continuous system model relates inputs and outputs with continuous equations or continuous transfer functions. When engineering simulations are implemented on digital computers, discrete representations of continuous functions become an important subject to study. The challenge is to relate a discrete transfer function, $H(z)$, to its continuous counterpart, $G(s)$, and to have $H(z)$ provide a satisfactory approximation of $G(s)$. The search for the "proper" method of representing continuous systems discretely has not, so far, produced one universally accepted "best approach." It is important to have a survey and compare the available methods. Such a survey forms the central part of this research.

To set the stage for a comparison of discretization techniques, sampled-data theory and Z-transformations are reviewed. Next, methods of

discretization of continuous transfer function are surveyed.

2.3.1 Overview of Sampled-Data Theory

Sampled-data transformations form the basis of techniques used to convert continuous systems to corresponding discrete systems. The interrelationship between representations of systems in the continuous-domain (S-domain) and those in the discrete-domain (Z-domain) are illustrated in Figure 2.5 (28). The irreversible conversion from $G(s)$ to $H(z)$ and from differential equations to difference equations implies that many approximation methods are available to convert a continuous system to many different discrete systems.

2.3.2 Sampling Process

In discrete systems, the signal flow is represented as a sequence of numbers as compared to a continuous function in continuous systems. A continuous signal flow can be translated to a discrete signal flow (a sequence of numbers) by a sampling process (20). This process consists of sampling a continuous signal, $f(t)$, every sampling interval T (the sampling interval is not

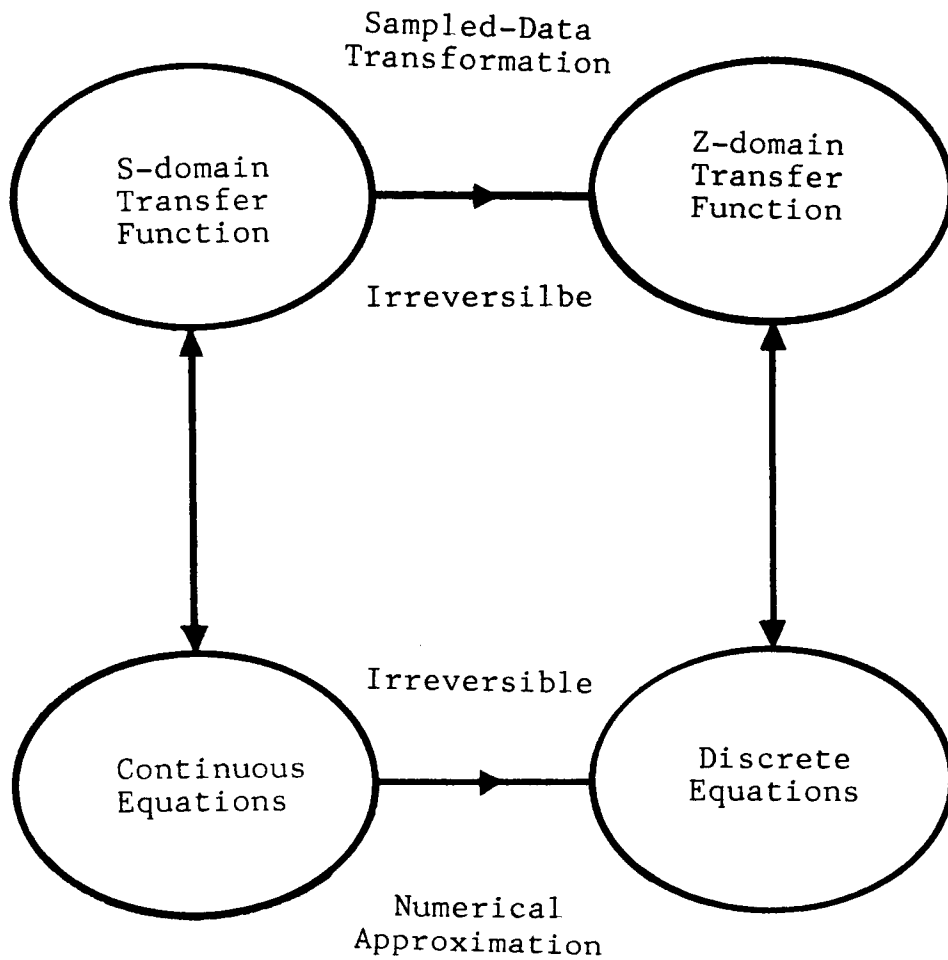


Figure 2.5. Relation Between Numerical Approximation and Sampled-Data Transformation

necessarily a constant). A "fictitious sampler" is introduced in the forward path of the continuous signal flow to sample the continuous signal. If the "sampling duration" is very small compared with the "sampling interval", the sampler can be closely approximated by an "impulse sampler" or "ideal sampler". The mathematical expression for this sampling procedure may be thought of as being the result of multiplying the original time function with a sampling function, which is called the "Dirac delta function or train of unit impulses", $\delta_T(t)$. This modulation process may be described by the following equations

$$\delta_T(t) = \sum_{n=0}^{\infty} \delta(t-nT) \quad (2.15)$$

where T is the sample interval and $\delta(t)$ is the unit impulse function and is defined as

$$\delta(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}$$

The resulting sampled signal is

$$f^*(t) = f(t) \delta_T(t) \quad (2.16)$$

A pictorial representation of sampling process is shown in Figure 2.6.

2.3.3 Z-Transform

The purpose of this subsection is to review Z-transforms and their application to the representation of sampled functions of time. The use of Z-transforms offers a very simple and elegant method for analyzing discrete systems. It allows the development of input-output models for discrete-time systems (discrete transfer functions), which constitute the basis for the dynamic analysis and design of digitally controlled processes. In other words, Z-transforms play the same roles for discrete-time systems as that played by Laplace transforms for the dynamic analysis and design of continuous systems.

The basic Z-transform (one-sided) is defined for time functions sampled at uniform intervals. Recalling the previous section, the continuous function, $f(t)$, is sampled every time interval T and represented as a sequence of numbers, $f(0)$, $f(T)$, $f(2T)$, ..., $f(nT)$. The Z-transform is defined as:

$$Z [f(t)] = \sum_{n=0}^{\infty} f(nT) z^{-n} \quad (2.17)$$

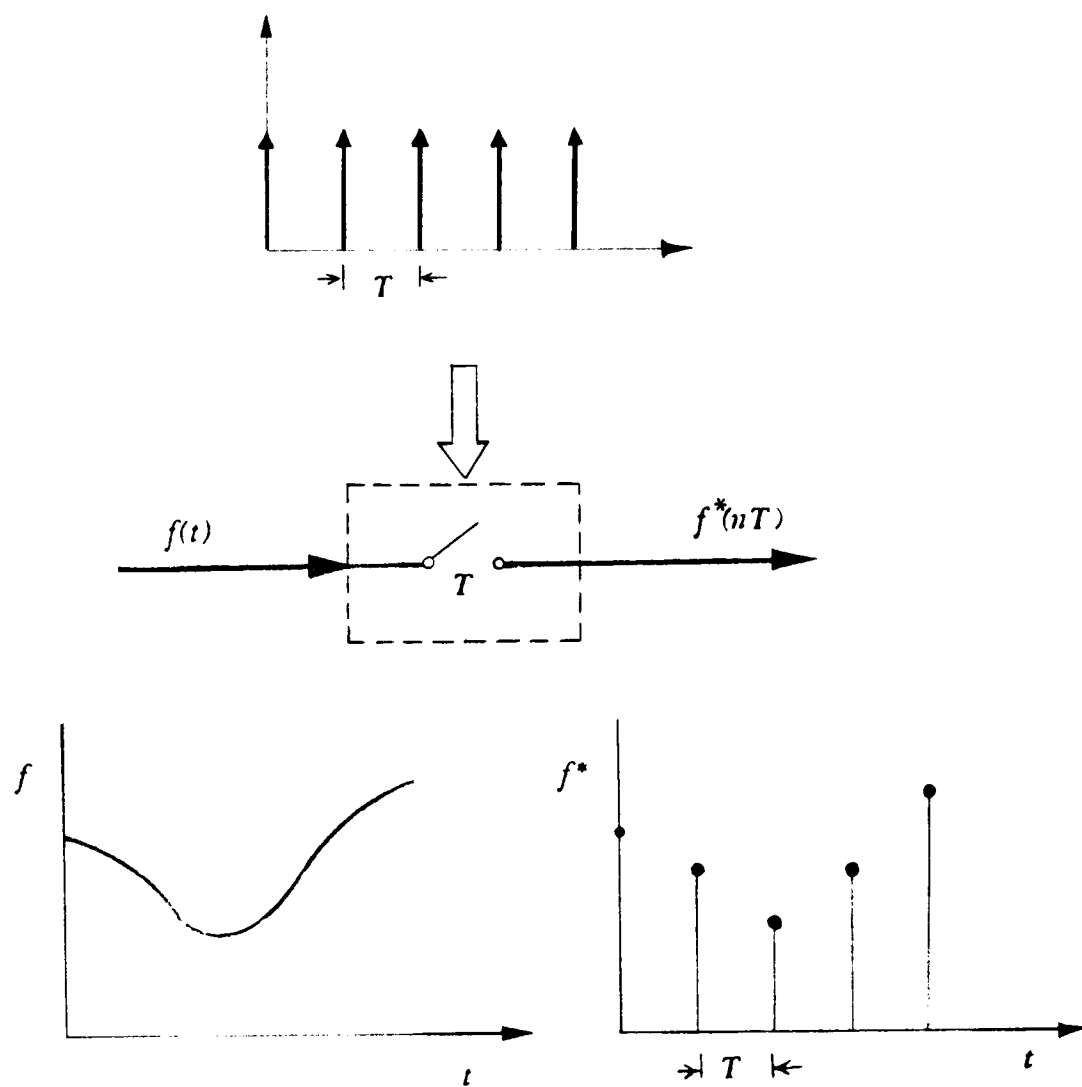


Figure 2.6. Input and Output Signals of the Sampler

Although the definition above refers to the sequence of sampled values of a continuous function, it is customary to use the following notation for the Z-transform of a function:

$$F(z) = \sum_{n=0}^{\infty} f(nT) z^{-n} \quad (2.18)$$

where $F(z)$ is the Z-transform of $f(t)$ sampled at fixed intervals of time, T .

2.3.4 Modified Z-Transform

The Z-transform enables the determination of a system's response only at the fixed sampling instants. To obtain response values between sampling instants, or to generally treat continuous time functions possessing time delays, modified Z-transforms are useful. Consider a delayed function $f(t-\theta)$, where $\theta = (k+\Delta)T$ with $0 < \Delta < 1$, then the modified Z-transform is defined as

$$F(z,m) = Z_m[f(t)] = z^{-k} \sum_{n=0}^{\infty} f(nT+mT) z^{-n} \quad (2.19)$$

where $m = 1 - \Delta$

Thus to find the response at any time $kT + \Delta T$ $F(z, m)$ is evaluated at $m = 1 - \Delta$ where $0 < \Delta < 1$. Note that if $\Delta = 1$ or 0 , then $m = 1$ or 0 correspondingly and the modified Z-transform reduces to the Z-transform.

2.3.5 Relationship Between Z-Transforms and Laplace Transforms

By taking the Laplace transform of a sampled function, the relationship between the Z-transform and the Laplace transform can be developed. From the previous section an ideally sampled continuous function can be described mathematically as:

$$f^*(t) = \sum_{n=0}^{\infty} f(nT) \delta(t - nT) \quad (2.20)$$

The Laplace transform of the sampled function is:

$$F^*(s) = L[f^*(t)] = \int_0^{\infty} f^*(t) e^{-st} dt \quad (2.21)$$

Substitution of Equation 2.20 into Equation 2.21 gives

$$F^*(s) = \int_0^{\infty} \left[\sum_{n=0}^{\infty} f(nT) \delta(t - nT) \right] e^{-st} dt \quad (2.22)$$

The value of the integral is zero except at sampling instants, thus the equation becomes

$$F(s) = \sum_{n=0}^{\infty} f(nT) e^{-nsT} \quad (2.23)$$

if we define:

$$z = e^{sT} \quad (2.24)$$

the Z-transform is equal to the Laplace transform of the sampled function when the substitution of Equation 2.24 is made

$$F(z) = F^*(s) \Big|_{e^{sT}=z} \quad (2.25)$$

This is a very important result, since it shows that the Z-transform is a special case of the Laplace transform. It allows us to derive Z-transfer functions directly from Laplace transfer functions.

2.3.6 Sampling Theorem

Once a sampling operation is cascaded with a continuous signal, both the frequency and time domain behavior of the signal are modified. This emphasizes that the signal is at best an

approximation of the original continuous signal. Also it highlights that different approaches to sampling the signal (or discretizing a set of equations) will give different approximations.

An alternative discretized form of $F(s)$ is obtained using Fourier series. Since $\delta_T(t)$ is a periodic function, it can be expressed by a Fourier series expansion

$$f^*(t) = \frac{1}{T} \sum_{-\infty}^{\infty} f(t) e^{-jn\omega_s t} \quad (2.26)$$

where $\omega_s = \frac{2\pi}{T}$, is sampling frequency.

The Laplace transform of the previous equation is

$$F^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} F(s + jn\omega_s) \quad (2.27)$$

The frequency spectrum of a band limited continuous signal and the spectrum of the corresponding sampled signal are shown in Figure 2.7.

As indicated in Equation 2.27, $F(s)$ is a period function with period $j\omega_s$. In other words, the s -plane is divided into an infinite number of periodic strips and the strip between $w = -\omega_s/2$ and $w = \omega_s/2$ is called the "primary strip." Each strip

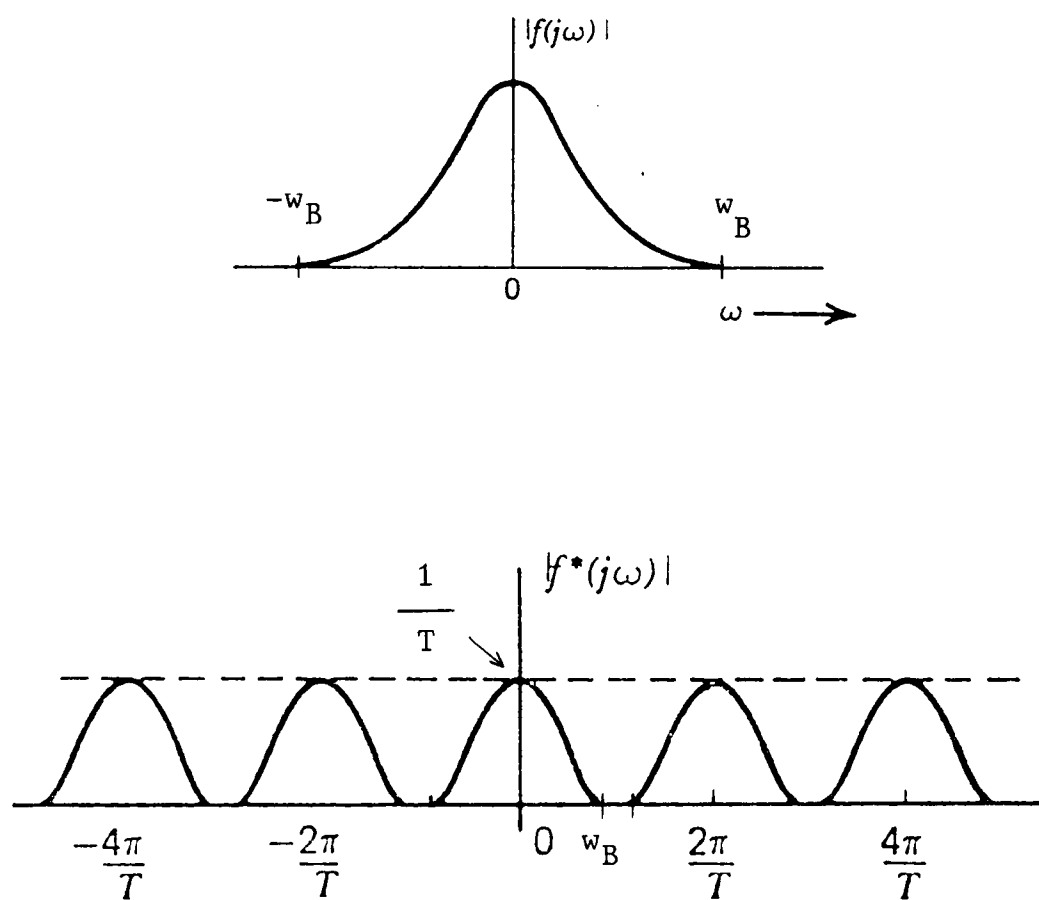


Figure 2.7. Frequency Domain Characteristics of Sampled Signal

in the left half of the s-plane is mapped into the unit disc on the z-plane. The selection of sampling frequency, w_s , determines the s-plane width of the primary strip. Thus, the sample frequency plays a crucial role in converting a continuous function to a discrete function. The well known sampling theorem developed by Shannon and Nyquist (2) states that a sampled continuous signal may be reconstructed from its samples if, and only if, the frequency content of the signal is lower than $w_s/2$. Unfortunately, in the simulation of overall continuous systems or the digital control of continuous processes, time delays occur and these must be considered together with the Shannon theorem.

In sampled-data processes a continuous signal is sampled to give a sequence of numbers with a sample interval, T . Intuitively, it seems that by increasing the sampling rate ($T \rightarrow 0$), the discrete approximation should asymptotically approach the continuous output. This assumption is not true for digital systems with finite word length. The very small numbers encountered can not be stored digitally with sufficient accuracy. Addition, subtraction and multiplication of such numbers

leads to truncation and roundoff errors. An important consideration in designing a simulation work is the trade off between a more accurate simulation and computer run-off error.

2.4 Discretization of Continuous Transfer Functions

Over the past 30 years several methods of discretizing continuous transfer functions have been developed and employed to obtain computational algorithms for digital computer simulation (13,20,22). These methods are reviewed in the following sections.

2.4.1 Substitution Method

One of the simplest methods to convert a continuous transfer function to discrete transfer function is to substitute for the continuous operator s in terms of z . There are many ways to implement such substitutions.

The Bilinear Transformation (27,32), also called the Tustin transformation, and the trapezoid integration method are a common substitution methods. The Tustin method (38) was originally presented as a general approach to linear system analysis through the representation of time functions in terms of sequences of numbers. The

practical application of the Tustin method may be reduced to the use of Tustin's approximation of the integration operator:

$$z = e^{-sT} \quad (2.28a)$$

$$s = \frac{1}{T} \ln(z) \quad (2.28b)$$

by truncating the Laurent series expansion of $\ln(z)$ to give

$$s \triangleq \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (2.29a)$$

or

$$s^{-1} \triangleq \frac{T}{2} \left(\frac{1 + z^{-1}}{1 - z^{-1}} \right) \quad (2.29b)$$

A linear continuous transfer function, $G(s)$, expressed as a ratio of polynomials in s is readily digitalized by substituting for s in terms of the Tustin integration operator.

$$H(z) = G(s) \Big|_{s = \frac{T}{2} \left(\frac{1 + z^{-1}}{1 - z^{-1}} \right)} \quad (2.30)$$

where T is the sample interval.

The advantages of the Bilinear transformation are: (i) it is very easy to apply, (ii) the whole left-hand side of the s-plane is transformed into the unit circle in the z-plane, hence there is no aliasing effect, Figure 2.8, (iii) it is an intrinsically stable transformation method. One disadvantage with Bilinear transformation is "frequency warping" which occurs due to the compression of frequency scale. Again consider the mapping from the s-plane to z-plane. If s is replaced by $j\omega$ and z by $e^{j\omega_z T}$ in the Bilinear transformation, Equation 2.29a, the result is

$$j\omega = \frac{2}{T} \frac{1 - e^{-j\omega_z T}}{1 + e^{j\omega_z T}} \quad (2.31)$$

Upon using De Moirre's theorems (20), Equation 2.31 can be rewritten as

$$\omega = \frac{2}{T} \tan \frac{\omega_z T}{2} \quad (2.32)$$

The last equation reflects the distortion of frequency scale in Bilinear transformation as the relation between ω and ω_z is nonlinear. The

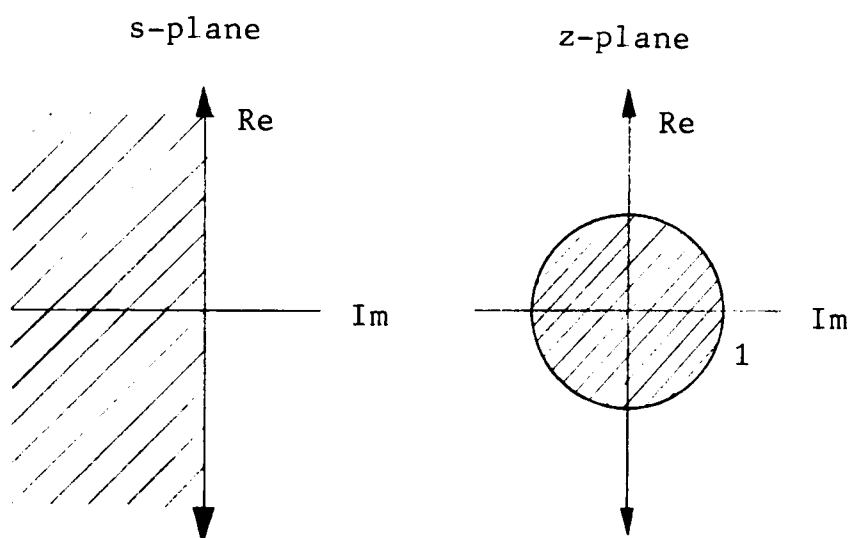


Figure 2.8. Mapping Relationship of S- and Z-plane by Bilinear Transformation

frequency distortion is also illustrated in Figure 2.9. To eliminate this at a specified frequency, w_1 , transformation can be modified as

$$s = \frac{w_1}{\tan \frac{w_1 T}{2}} \frac{z - 1}{z + 1} \quad (2.33)$$

then, the continuous transfer function and its discrete counterpart will have the same frequency response at frequency w_1 . This modified transformation to circumvent frequency compression around w_1 is called the "Bilinear transformation with prewarping."

There is a fairly extensive literature on substitution methods; Some substitution methods are listed in Table 2.1 (34,32).

2.4.2 Matched Z-Transform

The matched Z-transform method extensively uses mapping relations to map poles and zeroes of continuous transfer functions on the s-plane to poles and zeroes of discrete transfer functions on the z-plane. Since the dynamic behavior of continuous transfer functions is characterized by its poles, zeroes, and final value; heuristically,

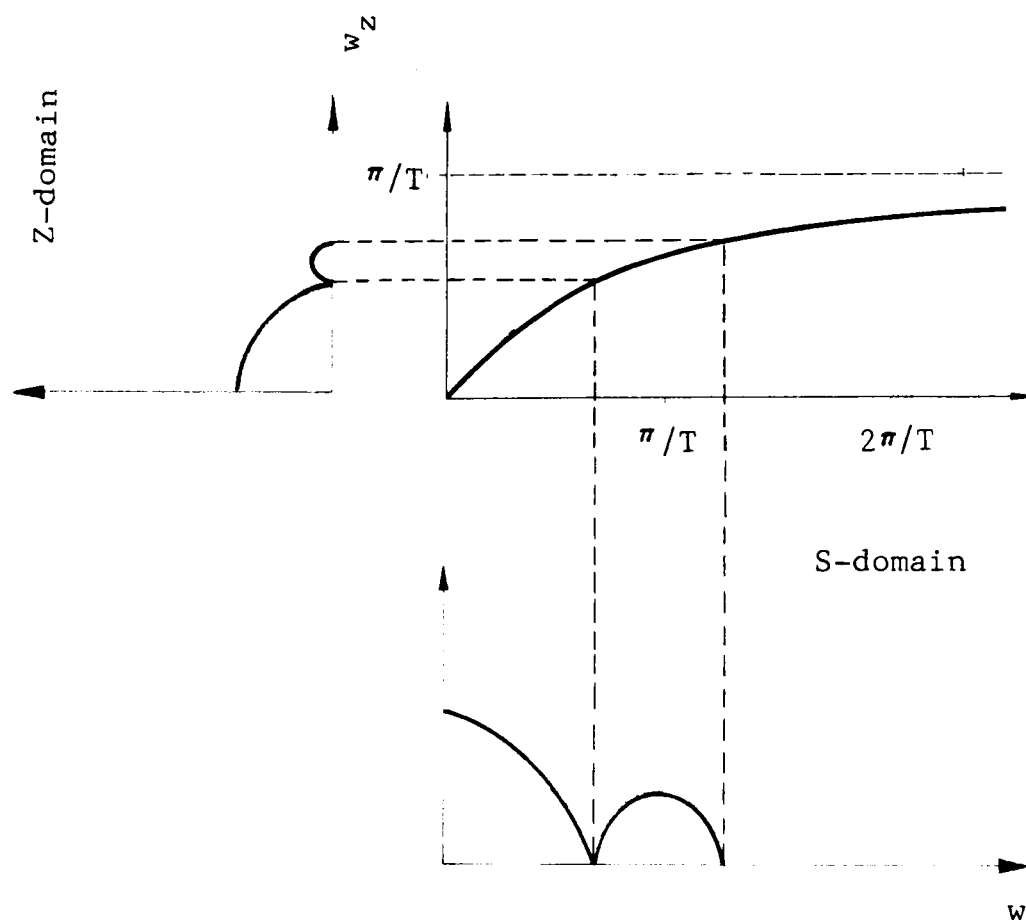


Figure 2.9. Frequency Distortion Due to the Bilinear Transformation

Table 2.1. Discrete Integrators for Substitution Method

$1/S^k$	Method	Operator	Method	Operator
$1/S$	Tustin	$\frac{T}{2} \frac{z+1}{z-1}$	Exact z transform	$\frac{Tz}{z-1}$
	Madwed-Truxal	$\frac{T}{2} \frac{z+1}{z-1}$	z form	$\frac{T}{2} \frac{z+1}{z-1}$
$1/S^2$	Tustin	$\frac{T^2}{4} \frac{z^2+2z+1}{(z-1)^2}$	Exact z transform	$\frac{T^2 z}{(z-1)^2}$
	Madwed-Truxal	$\frac{T^2}{6} \frac{z^2+4z+1}{(z-1)^2}$	z form	$\frac{T^2}{12} \frac{z^2+10z+1}{(z-1)^2}$
$1/S^3$	Tustin	$\frac{T^3}{8} \frac{z^3+3z^2+3z+1}{(z-1)^3}$	Exact z transform	$\frac{T^3}{2} \frac{z(z+1)}{(z-1)^3}$
	Madwed-Truxal	$\frac{T^3}{24} \frac{z^3+11z^2+11z+1}{(z-1)^3}$	z form	$\frac{T^3}{2} \frac{z(z+1)}{(z-1)^3}$
$1/S^4$	Tustin	$\frac{T^4}{16} \frac{z^4+4z^3+6z^2+4z+1}{(z-1)^4}$	Exact z transform	$\frac{T^4}{6} \frac{z(z^2+4z+1)}{(z-1)^4}$
	Madwed-Truxal	$\frac{T^4}{120} \frac{z^4+26z^3+66z^2+26z+1}{(z-1)^4}$	z form	$\frac{T^4}{6} \frac{z(z^2+4z+1)}{(z-1)^4}$

Source: Smith, J. M., Mathematical Modeling and Digital Simulation for Engineering and Scientists, Jone Wiley and Sons, Inc., New York, 1977.

it seems reasonable to expect that if a discrete transfer function is synthesized such that its poles, zeroes and final value match those of a continuous transfer function, a useful simulation model would be obtained.

$$K H(z) = G(s) \Big|_{z \rightarrow e^{sT}}$$

$$\text{where: } z_{\text{pole}} = e^{s_{\text{pole}} T} \quad (2.34)$$

$$z_{\text{zero}} = e^{s_{\text{zero}} T}$$

K = is the gain needed to match the continuous system steady-state for some specific input.

Some difficulties may be encountered with this method: (i) since the Z- transform is not a one-on-one mapping, if the continuous transfer function has zeroes with frequency greater than $\omega_s/2$ their z-plane positions will be aliased, (ii) a proper way to map infinite zeroes is difficult to find. Some literature recommends that the infinite zeroes all be placed at $z=0$ (28,34); other papers recommends they be placed at $z=-1$ (20). Astrom (1) proposed a scheme to derive the numerator of discrete transfer functions while the sampling rate is high, but no final practical conclusion is available.

2.4.3 Input Approximation

The input approximation method is conceptually unlike the substitution method or the matched Z-transform method. They are analytical methods for synthesizing a discrete transfer function through direct substitution for s in terms of z . In contrast, the input approximation method first synthesizes a discrete system to approximate a continuous system called a pseudo-continuous system. Then a discrete transfer function that describes the pseudo-continuous system is derived. The pseudo-continuous system is synthesized in a way that its signal flow remains analogous to that in the continuous system and the continuous elements also remain the same. First the continuous signal flow is broken into a discrete signal flow by inserting a "fictitious sampler" to sample the continuous signal. These sampled-signals are then reconstructed to approximate the original continuous signal by some data-hold device. The reconstructed signal replaces the original continuous signal as the process input. Sampling the process output signal

and making a sampled-data transformation, the discrete transfer function is obtained. The scheme presented in Figure 2.10 illustrates this procedure.

The discrete transfer functions derived from the input approximation method vary with data-holds used to reconstruct the continuous signals from the sampled signals.

2.4.4 Zero-Order Hold

The zero-order hold is the simplest and most popular data reconstruction device, so named because it uses the zero order collating polynomial to approximate the continuous input signals as a sequence of "stair-steps". Given a continuous function, $f(t)$, the mathematical approximation is

$$f(t) = f(nT) \quad nT \leq t < nT+T \quad (2.35)$$

where T is the interval between samples.

The s -domain transfer function is

$$H_{zo}(s) = \frac{1 - e^{-sT}}{s} \quad (2.36)$$

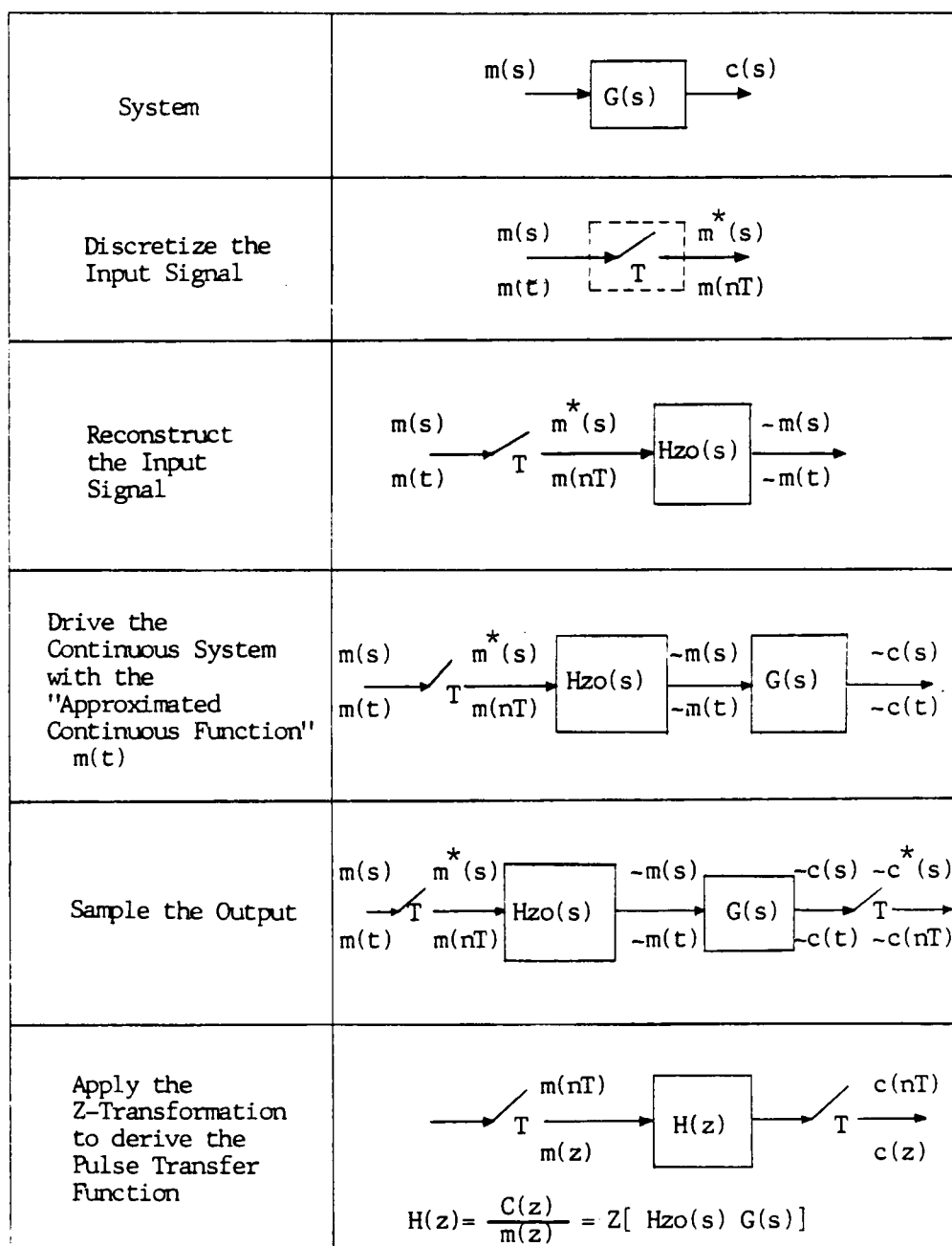


Figure 2.10. The Process of Synthesizing a Discrete-Representation of Continuous System

The time and frequency domain characteristics are shown in Figure 2.11.

2.4.5 Triangular Hold

Mathematically many sampled-data devices for reconstructing a continuous signal behave as truncated Taylor series. Such sampled-data devices generate more phase lag as higher order approximations are adopted. The triangular hold is not subject to these delays, it approximates the input signal with series of straight-line segments, Figure 2.12. The mathematical formulation and s-domain transfer function are

$$f(t) = f(nT) + \frac{(t - nT)}{T} [f(nT+T) - f(nT)]$$

$$H_{TR}(s) = -\frac{e^{sT}(1 - e^{-sT})^2}{Ts^2}$$

In a true physical sense this hold is unrealizable because its transfer function has a sample period lead which is impossible to implement in a hardware system. However, if our concern is only to use holds in mathematical models from which discrete transfer functions are derived, then it

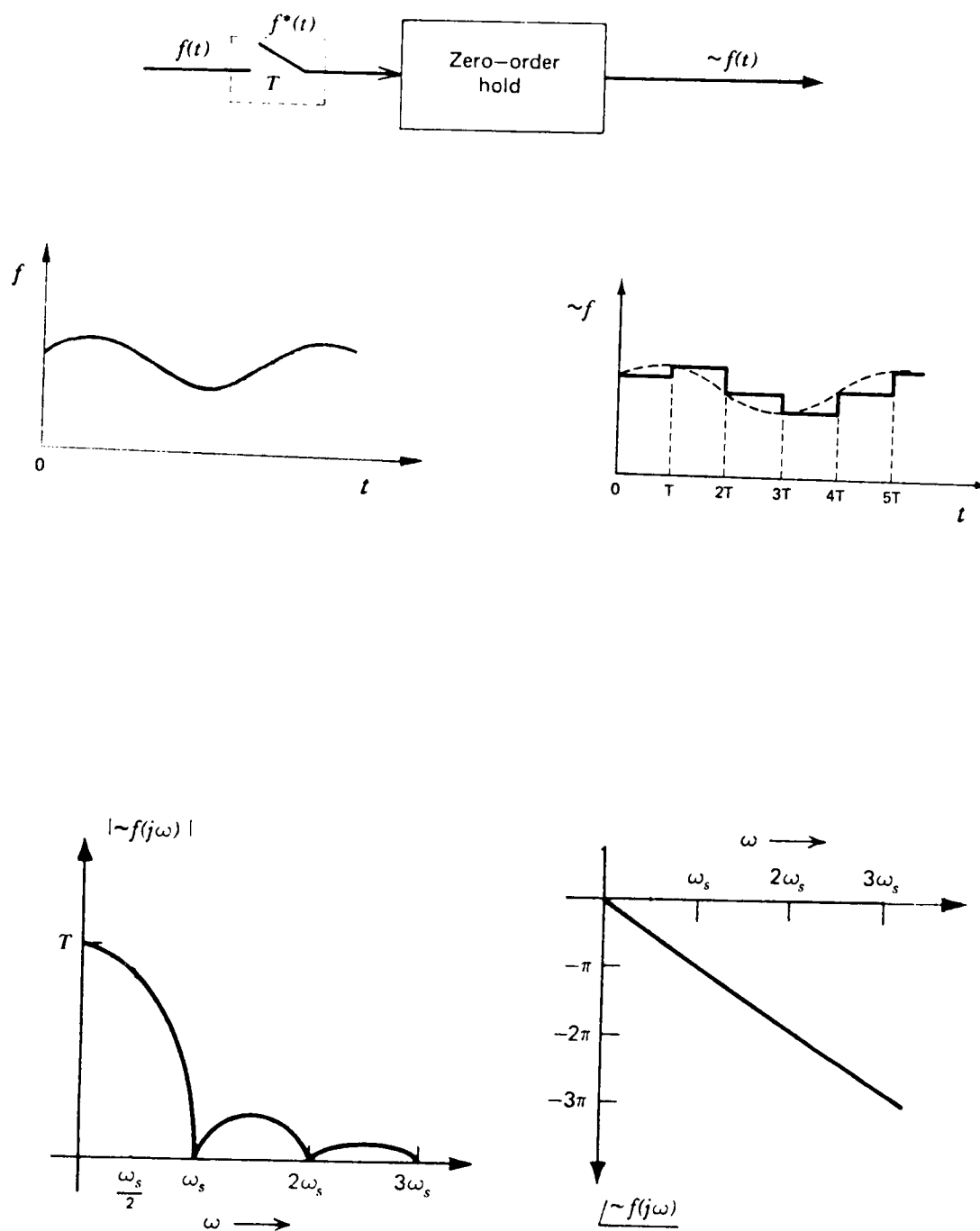


Figure 2.11. Time and Frequency Domain Characteristics of Zero-Order Hold

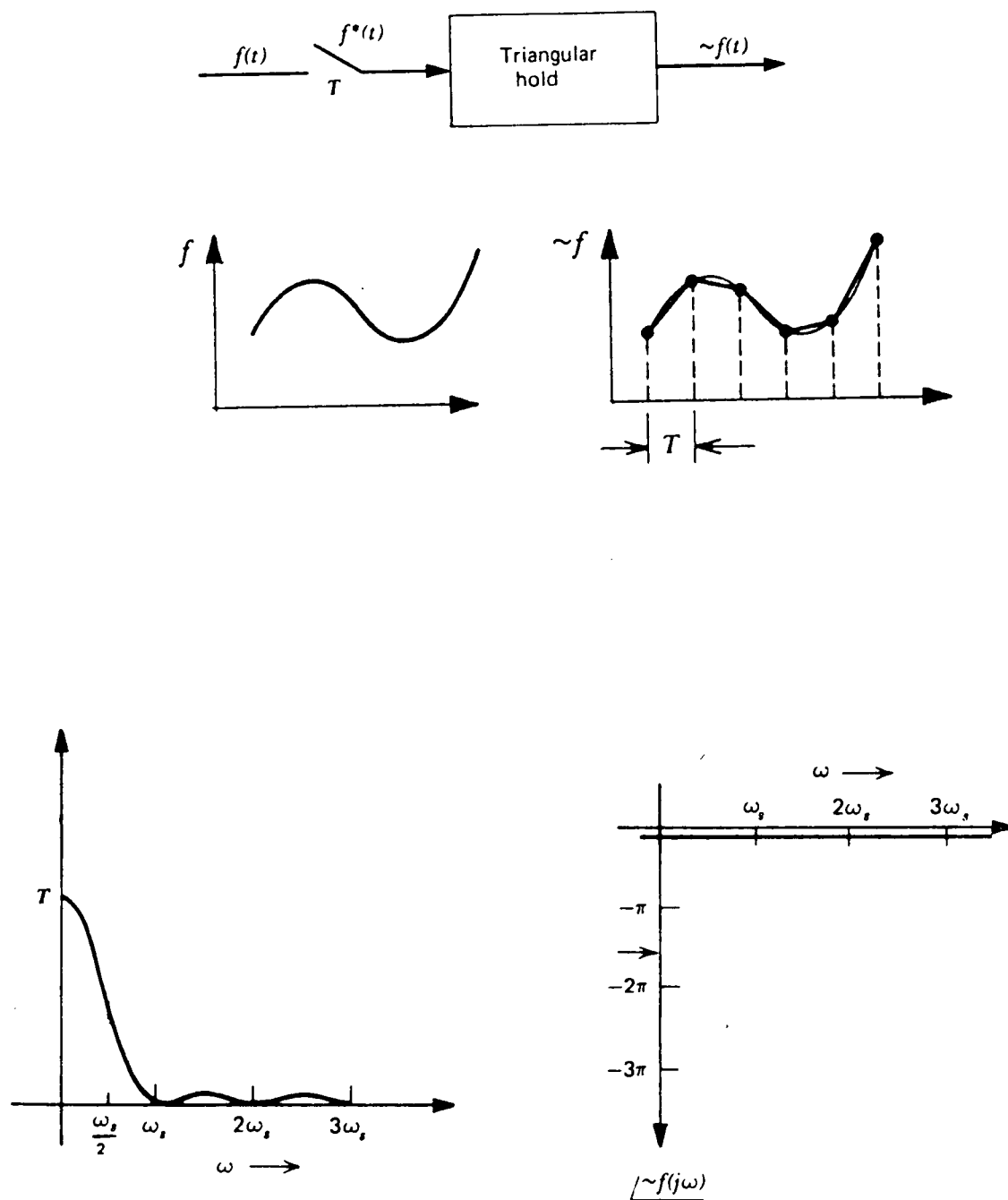


Figure 2.12. Time and Frequency Domain Characteristics of Triangular Hold

can be used to formulate simulations of continuous processes. Mathematically the triangular hold leads to an implicit formulation of the discretized model that can be solved for linear time invariant systems.

2.5 Summary

This chapter reviewed several simulation algorithms for linear systems. They are divided into numerical integration and discretization of continuous transfer function methods.

Unfortunately, each of these methods has its drawbacks. At the same time, the algorithms reviewed are by no means to include all available simulation algorithms, but have been selected to set the stage for the rest of this research.

CHAPTER III

LINEAR SYSTEM SIMULATION

Mathematical models were categorized into three types of simulation-oriented models at the beginning of the previous chapter. The different categories employed different simulation algorithms. Frequently, the purpose of a simulation is to investigate the dynamic behavior of processes under "some specific" input type rather than an arbitrary input type. This is true especially in the field of digital computer process control. From this standpoint, simulations can be separated into two divisions depending on whether the inputs to the simulated process are predetermined or not. In general, it is not possible for digital simulations to provide exact equivalent responses for their continuous counterparts in the case of indeterminate inputs.

3.1 Discrete Approximation

Suppose $G(s)$ is a continuous transfer function, as shown in Figure 3.1(a), where $E(s)$ is the input to the system then the corresponding output $Y(s)$ is

$$Y(s) = G(s) E(s) \quad (3.1)$$

If the continuous output is sampled, the sampled

response is expressed as

$$Y^*(s) = [G(s) E(s)]^* \quad (3.2)$$

If a sampled-data system, Figure 3.1(b), is used to approximate the original continuous system, the output of discrete system is

$$Y_d(s) = H^*(s) E^*(s) \quad (3.3)$$

The ideal discrete system for application here would be one that yields

$$y_d(t) = y(t) \quad \text{for all } t > 0 \quad (3.4)$$

However, due to the discontinuous nature of discrete systems, the best possible case would be

$$y_d(nT) = y(nT) \quad n=0,1,2,\dots \quad (3.5) \quad (3.5)$$

or

$$Y_d(s) = Y(s) \quad (3.6)$$

In order for Equation 3.5 or 3.6 to be satisfied, Equations 3.2 and 3.3 show the following equality is needed.

$$H^*(s) E^*(s) = [G(s) E(s)]^* \quad (3.7) \quad (3.7)$$

Taking the Z-transform of both sides of Equation 3.7 yields

$$H(z) E(z) = Z [G(s) E(s)] \quad (3.8) \quad (3.8)$$

Thus, the discrete transfer function which would allow the best possible results as stated in Equation 3.5 or 3.6 to be obtained is given by

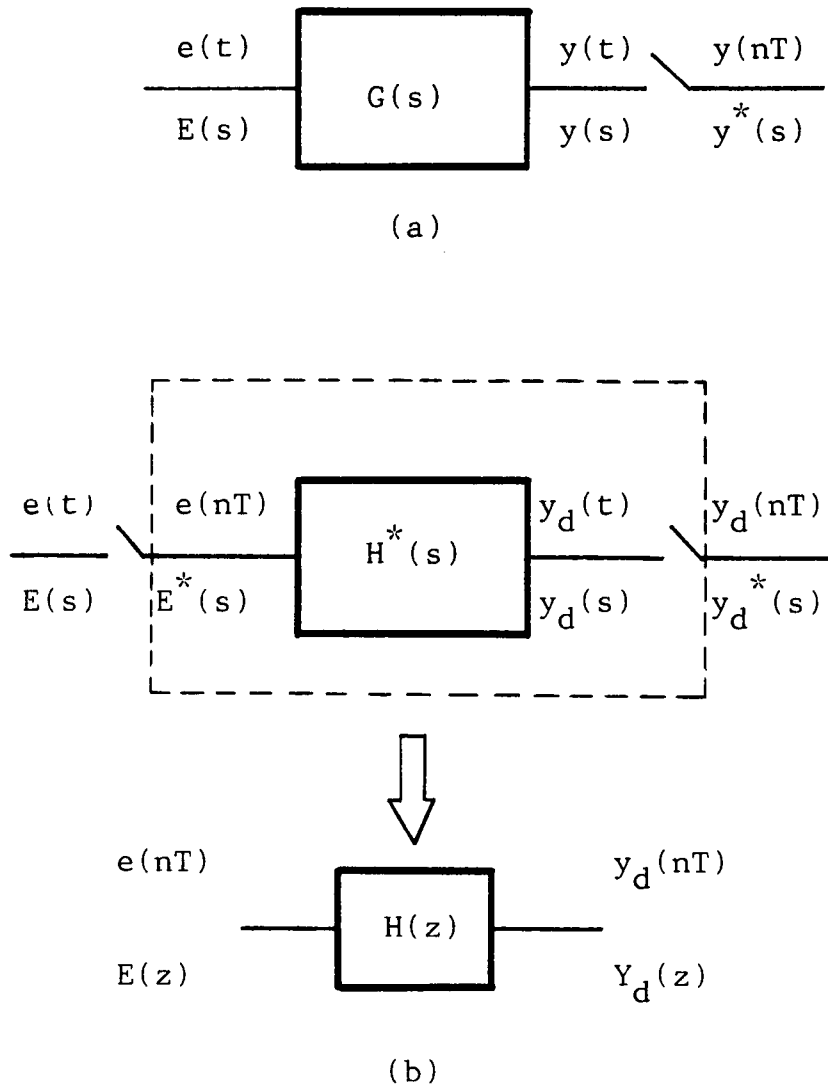


Figure 3.1. (a) A Continuous Transfer Function
 (b) An Approximating Discrete Transfer Function

$$H(z) = \frac{Z[G(s) E(s)]}{E(z)} \quad (3.9)$$

However, Equation 3.8 or 3.9 shows that $H(z)$ cannot be obtained independent of $E(z)$, or $e(t)$. This justifies the previous assertion that an exact discrete equivalent of the continuous transfer function for an arbitrary input cannot be obtained.

3.2 Linear System Simulation Formulas

The ideal simulation model is one that will reflect the exact response of a continuous process for any kind of input. Actually, it is impossible to obtain the ideal simulation model as illustrated in the previous section. Obviously, there exists a great gap between finding a flexible simulation model and retaining the accuracy of the simulation. To reduce this gap or to optimize the overall performance of a simulation is a tough problem, since so many factors should be taken into account and usually the importance of the various factors is different from one case to another. Before starting to address the question of a general simulation, it is imperative to consider when it is necessary to synthesize a simulation model for arbitrary inputs. Likewise, the question of when will system performance only be studied for specified inputs should

be asked. Often simulations are used to analyze system behavior, formulate design changes, test advanced control strategies, etc. under certain types of inputs. In such cases it would seem useful to derive simulation models for the specified inputs and know that the simulation gives the same response as its continuous linear system counterpart. Properly employing these "exact simulation models" will greatly help to eliminate many stability and accuracy problems in the simulation study and facilitate the simulation work.

Based on the above discussion, simulation models in the following sections are derived for impulse, step, and ramp inputs. Chapter IV then returns to the broader problem of simulation models for systems under arbitrary inputs.

3.3 Pulse Transfer Function

As mentioned earlier, the original purpose for developing discretization techniques was to observe and analyze the behavior of continuous systems subjected to discrete inputs, e.g. the application of digital controllers. Sampled-data theory uses the Laplace transform approach to calculate the response of a continuous system to equally spaced impulses. The Z-transformation gives directly the impulse response of

the continuous system. Therefore, the Z-transformation of an input-output system is called the "pulse-transfer function" or "Z-transfer function of a sampled-data system." The Z-transform gives the exact response of a linear continuous process with sampled inputs.

Given a continuous transfer function, $G(s)$, the pulse transfer function, $H(z)$, is the Z-transformation of $G(s)$. The Z-transformation is also called the impulse invariant transformation, since the discrete transfer functions exactly match the impulse response of continuous system at every sample instant. Z-transforms of continuous transfer functions and hence impulse invariant discrete transfer functions abound in literature (18,37).

3.4 Development of Simulation Formulas for Overall Input-Output Description

A discrete transfer function which performs exactly as its continuous counterpart under specific inputs can be derived from Equation 3.9. These discrete transfer functions can also be realized by cascading an ideal sampler with an appropriate data-hold device to the original system, then applying the sampled-data transformation to derive the discrete-transfer function.

3.4.1 Step Invariant Discrete Transfer Function

Suppose discrete transfer functions are synthesized to match the step response of continuous transfer functions at each sampling instant. Then,

$$e(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (3.10)$$

the Laplace transform of $e(t)$ is

$$E(s) = L[e(t)] = \frac{1}{s} \quad (3.11)$$

and the Z-transformation of $e(t)$ is

$$E(z) = \frac{z}{z - 1} \quad (3.12)$$

Substituting Equations 3.11 and 3.12 into 3.9, the step invariant discrete-transfer function is obtained as

$$G(z) = \frac{z - 1}{z} Z \left[\frac{G(s)}{s} \right] \quad (3.13)$$

Equation 3.13 is equivalent to cascading an ideal sampler followed by a zero-order hold to the continuous transfer function. Examples of step invariant discrete transfer functions which have been derived as part of this work are listed in Table 3.1.

3.4.2 Ramp Invariant Discrete Transfer Function

Suppose discrete transfer functions are synthesized to preserve the ramp input response of continuous transfer functions. Then

Table 3.1. Step Invariant Discrete Transfer Function

$G(s)$	$H(z)$
$\frac{1}{s}$	$\frac{T}{z-1}$
$\frac{1}{s^2}$	$\frac{T^2 (z+1)}{2 (z-1)^2}$
$\frac{a}{s+a}$	$\frac{1 - \exp(-aT)}{z - \exp(-aT)}$
$\frac{a}{s(s+a)}$	$a_0 = (aT - 1 + \exp(-aT)) / a$ $a_1 = (1 - \exp(-aT) - aT(\exp(-aT))) / a$ $b_1 = -(1 + \exp(-aT))$ $b_2 = \exp(-aT)$
$\frac{a}{(s+a)^2}$	$a_0 = 1 - (1+aT)\exp(-aT)$ $a_1 = (aT - 1 + \exp(-aT))\exp(-aT)$ $b_1 = -2\exp(-aT)$ $b_2 = \exp(-2aT)$
$\frac{1}{(s+a)(s+b)}$	$a_0 = (b(1-\exp(-aT)) - a(1-\exp(-bT))) / (ab(b-a))$ $a_1 = (a(1-\exp(-bT))\exp(-aT) - b(1-\exp(-aT))\exp(-bT)) / (ab(b-a))$ $b_1 = -(\exp(-aT) + \exp(-bT))$ $b_2 = \exp(-(a+b)T)$

Table 3.1. (Continued)

$G(s)$	$H(z)$
$\frac{a^2 + b^2}{(s+a)^2 + b^2}$	$a_0 = 1 - \exp(-aT)(\cos(bT) + a \sin(bT)/b)$ $a_1 = \exp(-2aT) + \exp(-aT)(a \sin(bT)/b - \cos(bT))$ $b_1 = -2 \exp(-aT) \cos(bT)$ $b_2 = \exp(-2aT)$
$\frac{s}{(s^2 + a^2)}$	$a_0 = T \exp(-aT)$ $a_1 = 0$ $b_1 = -2 \exp(-aT)$ $b_2 = \exp(-2aT)$
$\frac{s}{(s+a)(s+b)}$	$a_0 = (\exp(-aT) - \exp(-bT)) / (b-a)$ $a_1 = 0$ $b_1 = -(\exp(-aT) + \exp(-bT))$ $b_2 = \exp(-(a+b)T)$
$\frac{s}{(s+a)^2 + b^2}$	$a_0 = \exp(-aT) \sin(bT) / b$ $a_1 = 0$ $b_1 = -2 \exp(-aT) \cos(bT)$ $b_2 = \exp(-2aT)$

For second order systems the discrete transfer function is

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

$$e(t) = t \quad (3.14)$$

the Laplace transformation of $e(t)$ is,

$$E(s) = \frac{1}{s^2} \quad (3.15)$$

and the Z-transformation of $e(t)$ is

$$E(z) = \frac{Tz}{(z-1)^2} \quad (3.16)$$

where T is the sampling interval.

Substituting Equations 3.15 and 3.16 into 3.9 yields

$$G(z) = \frac{(z-1)^2}{Tz} Z\left[\frac{G(s)}{s^2}\right] \quad (3.17)$$

This result equivalent to cascading an ideal sampler followed by a tranagular hold onto the continuous transfer function. Again several examples of ramp invariant discrete transfer functions have been derived in this work and are collected in Table 3.2.

3.5 Simulation Formulas Based on State Space

Descriptions

The basic approach of solving state space formulated models is to use sampled values of the input signals along with interpolating or extrapolating sampled values via some type of data-hold device (ie, an input approximation algorithm) and then employing the analytical solution of the state space equation for the approximated inputs to obtain the output responses. An

Table 3.2. Ramp Invariant Discrete Transfer Function

$G(s)$	$H(z)$
$\frac{1}{s}$	$\frac{T}{2} \frac{z+1}{z-1}$
$\frac{1}{s^2}$	$\frac{T}{6} \frac{z+z+1}{(z-1)^2}$
$\frac{a}{(s+a)}$	$\frac{(aT-1+\exp(-aT))z + (1-\exp(-aT)-aT \exp(-aT))}{aT(z - \exp(-aT))}$
$\frac{a}{(s+a)^2}$	$a_0 = [aT-2+(2+aT)\exp(-aT)] / T$ $a_1 = [2-4aT \exp(-aT)-2 \exp(-2aT)] / T$ $a_2 = [(aT-2)\exp(-aT)+(aT+2)\exp(-2aT)] / T$ $b_1 = -2 \exp(-aT)$ $b_2 = \exp(-2aT)$
$\frac{a^2 b^2}{(s+a)(s+b)}$	$a_0 = [abT(a-b) - (a^2 - b^2) + a \exp(-bT) - b \exp(-aT)] / (T(a-b))$ $a_1 = [(a^2 b^2) + (a^2 + b^2)(\exp(-aT) - \exp(-bT)) - (a^2 - b^2)\exp(-(a+b)T) - abT(a-b)(\exp(-aT) + \exp(-bT))] / (T(a-b))$ $a_2 = [b \exp(-bT) - a \exp(-aT) + (a-b)(a+b+abT)\exp(-(a+b)T)] / (T(a-b))$ $b_1 = -2 \exp(-aT)$ $b_2 = \exp(-2aT)$
$\frac{(a^2 + b^2)}{(s+a)^2 + b^2}$	$a_0 = [(a^2 + b^2)T - 2a + 4a \exp(-aT)\cos(bT) - \exp(-aT)\sec\theta \cos(bT+\theta)] / T$ $a_1 = [2a - 2a \exp(-2aT) - (4a + 2(a^2 + b^2)T)\exp(-aT)\cos(bT) + 2\exp(-aT)\sec\theta \cos(bT+\theta)] / T$ $a_2 = [T(2a + (a^2 + b^2)\exp(-2aT) - \exp(-aT)\sec\theta \cos(bT+\theta))] / T$ $b_1 = -2 \exp(-aT)\cos(bT)$ $b_2 = \exp(-2aT)$ $\theta = [\tan^{-1}(b-a)/2ab]$

Table 3.2. (Continued)

$G(s)$	$H(s)$
$\frac{as}{(s+a)}$	$a_0 = [1 - \exp(-aT) - aT \exp(-aT)]/T$ $a_1 = [2aT \exp(-aT) - 1 + \exp(-2aT)]/T$ $a_2 = [(1-aT)\exp(-aT) - \exp(-2aT)]/T$ $b_1 = -2 \exp(-aT)$ $b_2 = \exp(-2aT)$
$\frac{s}{(s+a)(s+b)}$	$a_0 = [b(1 - \exp(-aT)) - a(1 - \exp(-bT))]/(Tab(b-a))$ $a_1 = [b(1 - \exp(-aT))(1 + \exp(-bT)) - a(1 - \exp(-bT))(1 + \exp(-aT))]/(Tab(b-a))$ $a_2 = [b \exp(-bT)(1 - \exp(-aT)) - a \exp(-aT)(1 - \exp(-bT))]/(Tab(b-a))$ $b_1 = -(\exp(-aT) + \exp(-bT))$ $b_2 = \exp(-(a+b)T)$
$\frac{(a^2 + b^2)s}{(s+a)^2 + b^2}$	$a_0 = [1 - \exp(-aT)(\cos(bT) + a \sin(bT)/b)]/T$ $a_1 = [1 - 2a \sin(bT)/b - \exp(-2aT)]/T$ $a_2 = -[\exp(-aT)(a \sin(bT)/b - \cos(bT)) + \exp(-2aT)]/T$ $b_1 = -2 \exp(-2aT)\cos(bT)$ $b_2 = \exp(-2aT)$

For second order systems the discrete transfer function is

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

important feature of this approach which is different from classical numerical approaches is that the approximation occurs in sampling and reconstructing the input functions. The responses of the model to the reconstructed input functions replaces the numerical approximation of the integration operation. It can be shown (21) that the analytical solution of the state space Equation

$$\dot{X}(t) = A X(t) + B U(t) \quad (3.18)$$

is

$$X(t) = e^{A(t-t_0)} X(t_0) + \int_{t_0}^t e^{A(t-r)} B U(r) dr \quad (3.19)$$

where t_0 is the initial time.

Setting $t_0 = kT$, the beginning of a sample interval, and $t = (k+1)T$, the end of a sample interval, Equation 3.20 can be used to give the solution for the sample interval kT to $(k+1)T$

$$X(k+1) = e^{AT} X(k) + \int_{kT}^{kT+T} e^{A(kT+T-r)} B U(r) dr \quad (3.20)$$

The solution of Equation 3.20 requires, however, that the input function, $U(t)$, be known, or approximated by some polynomial. Once the data-hold is selected this polynomial or the sampled-data process is specified and the integration in Equation 3.20 can be carried out to obtain the discrete system's state space solution.

3.5.1 Step Invariant Discrete State Space Equation (21)

If the input is assumed constant between sampling instants, $U(t) = U(kT)$, $kT \leq t < kT+T$, then the discrete approximation is given by

$$X(k+1) = P X(k) + Q U(k) \quad (3.21)$$

where: $P = \exp(AT) \quad (3.22)$

$$Q = \left[\int_0^T e^{A(T-\sigma)} d\sigma \right] B \quad (3.23)$$

Calculation of $\exp(AT)$, the matrix exponential of AT , is treated in section 3.6.2.

If A is invertable, then

$$Q = (P - I) A^{-1} B \quad (3.24)$$

Calculation of Q in the case of A being singular (not invertable) is addressed in section 3.6.5.

3.5.2 Ramp Invariant Discrete State Space Equation

If the input is assumed to be linear between sampling instants, the input function can be approximated as:

$$U(t) = U_k + \frac{U_{k+1} - U_k}{T} t \quad (3.25)$$

$$\text{for } kT \leq t < (k+1)T$$

Substituting the approximating ramp input function, Equation 3.25, into Equation 3.20, the discrete State Space solution is

$$X(k+1) = P X(k) + Q U(k+1) + W U(k) \quad (3.26)$$

$$\text{where: } P = \exp(AT) \quad (3.27)$$

If A is invertable, then

$$Q = \left[-A^{-1} + \frac{1}{T} (e^{AT} - I) A^{-2} \right] B \quad (3.28)$$

$$W = \left[e^{AT} A^{-1} - \frac{1}{T} (e^{AT} - I) A^{-2} \right] B \quad (3.29)$$

3.6 Matrix Exponential Computation

A central problem in employing the discrete state space equation solutions lies in the matrix exponential computation, $\exp(AT)$, required in Equations 3.21 and 3.22. Numerous algorithms for computing the matrix exponential, $\exp(AT)$, have been proposed based on various techniques (40) such as eigenvalue/eigenvector representations, numerical integration, truncated power series approximations, and rational approximations. Some of the methods have proposed specific algorithms, while others are based on less constructive characterizations. These algorithms have been less than completely satisfactory. In addressing the utility that the state space solutions could provide if the matrix exponential could be calculated, the "mighty algorithm" which would be suitable to compute the matrix exponential for any matrix is not sought, but focuses on those algorithms most suitable for process simulation.

3.6.1 Taylor Series Method

To evaluate the matrix exponential, a common method often used in industry is the truncated power series (26) of $\exp(AT)$. The fundamental computation technique comes from a Taylor series expansion of $\exp(AT)$,

$$e^{AT} = I + TA + \frac{T^2}{2!} A^2 + \frac{T^3}{3!} A^3 + \dots \quad (3.30)$$

There are two proposed approaches to truncate the infinite series:

1. The first approach is to simply sum the series until the difference between the current and previous sum is within a predefined vaule.
2. The second approach is to approximat the infinite series by r terms according to Takahashi's criteria (36) where r is determined from

$$(nr)^r e^{nm} = 0.001 r! \quad (3.31)$$

n is the order of the system, m is the absolute value of the largest element in the matrix, and the number 0.001 was found by experience to give the best results in the simulation. Presently, however, the series expansion calculation of the matrix exponential is not recommended (19), since usually the difficulty is not truncation of the series, but truncation of arising from addition.

3.6.2 Pade' Approximation

Pade' approximation (31,26) is one of the rational function approximations for a continuous function, $f(x)$, with a rational function, $r(x)$. A rational function $r(x)$ of degree N has the form

$$r(x) = \frac{p(x)}{q(x)} \quad (3.32)$$

where $p(x)$ is a polynomial of x of a degree p and $q(x)$ is a polynomial of x of degree q , and p and q sum to N .

Pade' approximation techniques choose the $N+1$ coefficients in $r(x)$ so that $f^{(k)}(0) = r^{(k)}(0)$ for each $k = 0$ to N . This is the most significant feature of the approximation. Pade' approximation is the extension of Taylor polynomials to rational functions. When $p = N$ and $q = 0$, the Pade' approximation is a Taylor polynomial of degree N expanded about zero.

The matrix Pade' approximation extensively applies the scalar Pade' approximation technique to compute the matrix exponential. The form of the approximations are

$$R_{pq}(V) = [D_{pq}(V)]^{-1} N_{pq}(V) \quad (3.33)$$

where

$$N_{pq}(V) = \sum_{j=0}^p \left[\frac{(p+q-j)! p!}{(p+q)! j! (p-j)!} \right] V^j \quad (3.34)$$

$$D_{pq}(V) = \sum_{j=0}^q \left[\frac{(p+q-j)! q!}{(p+q)! j! (q-j)!} \right] (-V)^j \quad (3.35)$$

where V is a general square matrix. In the case of the state space solution given by Equation 3.20 or 3.21 V would be equal to AT . It has been shown that the

diagonal Pade' approximations ($p = q$) are preferred over the off diagonal Pade' approximations ($p \neq q$).

The major difficulties for the matrix Pade' approximation algorithm appear as the norm of computed matrix becomes large, i.e. as the spread between the largest and smallest eigenvalue of the computed matrix grows. An algorithm involving scalar matrix scaling before the matrix exponential is calculated followed by a procedure on the result called squaring has been suggested (26). The scaling and squaring operations are directed at reducing inaccuracy problems encountered with matrices having large matrix norms. The method of Van Loan first reduces the infinite norm of a matrix by dividing the matrix by two repeatedly until the infinite norm of the scaled matrix is less than unity. Diagonal Pade' approximation formulas are used to compute the matrix exponential. Once the exponential of the scaled matrix is found, the exponential of the original matrix is obtained by reversing the scaling procedure, which is called squaring.

A potential difficulty can arise during the scaling and squaring procedure. The norm of the scaled matrix exponential when squared may be larger than the norm of the original matrix exponential depending on the value of m , that is,

$$|| e^V || \ll || e^{V/m} ||^m$$

Thus the original goal of the scaling and squaring operations are not achieved, as the round off errors in the m -th power of a matrix, say B^m , are usually small relative to $|| B ||^m$ rather than $|| B^m ||$ (26,39).

A Fortran program, documented in Applied Control Laboratories Report 8601¹, for the matrix exponential computations has been created according to Van Loan's algorithm and some new procedures presented in section 3.6.5.

3.6.3 Companion Matrix Method

A class of State Space equations generally arise upon the formulation of a state space model for a high order ODE model (SISO systems). The state transition matrix, A , for such models are generally formed in either the companion canonical form (21) or in the Jordan canonical form (21) when A has distinct or repeated eigenvalues respectively. Chapter II equations 2.2 to 2.6 show an illustration for a 3rd order ODE with distinct eigenvalues.

¹ACl Report 8601, Dr. D. D. Bruns, Department of Chemical Engineering, University of Tennessee, Knoxville, Tn 37996-2200.

A discussion of algorithms which involve the computation of $\exp(V)$, where V is a companion matrix, is now undertaken. If V is of the companion form, which is expressed as,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot \\ 0 & 0 & \dots & & & 1 \\ c_0 & c_1 & \dots & & & c_{n-1} \end{bmatrix} \quad (3.36)$$

and if all the eigenvalues of V are real and distinct, then a diagonal matrix related to V can be obtained by a similarity transformation,

$$\Lambda = P^{-1} V P \quad (3.37)$$

or

$$V = P \Lambda P^{-1} \quad (3.38)$$

where: Λ is a diagonal matrix.

$$\Lambda = \text{diag} (\lambda_1 , \lambda_2 , \dots , \lambda_n) \quad (3.39)$$

It can be shown that the P which transforms V to Λ is the Vandamonde Matrix (21,26),

$$P = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & & \lambda_n^2 \\ \vdots & \vdots & \dots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & & \lambda_n^{n-1} \end{bmatrix} \quad (3.40)$$

where $\lambda_1, \dots, \lambda_n$ are the real distinct eigenvalues of V .

When $V = P \Lambda P^{-1}$, the series for the matrix exponential is easily verified to be

$$e^V = P e^\Lambda P^{-1} \quad (3.41)$$

Note that e^Λ can be computed easily, since Λ is a diagonal matrix

$$e^\Lambda = \text{diag} (e^{\lambda_1}, e^{\lambda_2}, \dots, e^{\lambda_n}) \quad (3.42)$$

Equations 3.40, 3.41 and 3.42 provide a very easy scheme to calculate the matrix exponential of V .

when a matrix has real and distinct eigenvalues, the above procedure looks like a very attractive technique to calculate the exponential of the matrix. The results for all steps in the calculation procedure are based on analytic results, especially for e^Λ . This approach has not been treated numerically in the

literature that was reviewed in the course of this research. It is suggest as a topic worthy of addition investigation.

3.6.4 Jordan Canonical Matrix Method

In general, when V has multiple-order or repeated eigenvalues, a diagonal matrix can not be obtained under a similarity transformation. However, there exists a similarity transformation to generate a Jordan canonical form of V (21)

$$V = P J P^{-1} \quad (3.43)$$

where J is called the Jordan canonical form

$$J = \begin{bmatrix} \lambda_1 & 1 & 0 & | & 0 & 0 \\ 0 & \lambda_1 & 1 & | & 0 & 0 \\ 0 & 0 & \lambda_1 & | & 0 & 0 \\ \hline 0 & 0 & 0 & | & \lambda_2 & 0 \\ 0 & 0 & 0 & | & 0 & \lambda_3 \end{bmatrix} \quad (3.44)$$

The repeated eigenvalues form typical blocks associated with these eigenvalues called Jordan Blocks. In Equation 3.44 the Jordan Blocks are enclosed by dashed lines, the upper block is for λ_1 which is repeated three times and the two lower blocks are associated with the distinct eigenvalues λ_2 and λ_3 respectively. The J matrix can be expressed as

$$V = P[J_1, \dots, J_k]P^{-1} \quad (3.45)$$

where $J_1, \dots, J_k$ are Jordan blocks. Then the matrix exponential for VT can be expressed as

$$e^{VT} = P[e^{J_1 T} \dots e^{J_k T}] P^{-1} \quad (3.46)$$

The exponentials of Jordan blocks J_i can be given in closed form (20), for example, if

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{bmatrix} \quad (3.47)$$

then

$$e^{J_i T} = e^{\lambda_i T} \begin{bmatrix} 1 & T & \frac{T^2}{2!} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

Again, as in the case of distinct real eigenvalues, this section provides the basis for a potentially attractive method of calculating the matrix exponential of a matrix that has real repeated eigenvalues. Some

problems to be addressed in using the outline provided here are to decide how close two eigenvalues need to be before they are considered equal and what effect this criteria has on the matrix exponential accuracy.

3.6.5 Q matrix Calculation

If the inverse of the state transition matrix, A , in Equation 3.18 exists, then the Q matrix in Equations 3.21 and 3.22 is easy to calculate according to Equation 3.24.

However, when A is singular, its inverse does not exist, calculating Q then becomes numerically much more difficult. One common approach is to numerically integrate Equation 3.23. Another approach is based on the Taylor series expansion of Equation 3.23 which is

$$Q = [I + \frac{T}{2!}A + \dots] B \quad (3.49)$$

Using a truncated form of this expansion allows Q to be approximated.

Recently Birdwell (4) has suggested a potentially much more attractive approach to evaluate Q for step invariant discrete state space solutions. This revised algorithm has been developed during the cause of this study and is presented in the following. The algorithm

requires an augmented matrix, R , to be formed as

$$R = \begin{bmatrix} AT & B \\ 0 & 0 \end{bmatrix} \quad (3.50)$$

where A : state transient matrix.

B : input matrix.

T : simulation step size.

The matrix exponential of R is

$$e^R = I + R + \frac{R^2}{2!} + \frac{R^3}{3!} + \dots \quad (3.51)$$

$$= \begin{bmatrix} P & K \\ 0 & I \end{bmatrix}$$

where P and K are obtained as

$$P = I + TA + \frac{T^2}{2!} A^2 + \frac{T^3}{3!} A^3 + \dots \quad (3.52)$$

$$K = \left[I + \frac{T}{2!}A + \frac{T^2}{3!}A^2 + \dots \right] B \quad (3.53)$$

Comparing the definition of Q given in Equation 3.49 to Equations 3.53 shows that

$$Q = K * T \quad (3.54)$$

Therefore once the matrix exponential of R is obtained using state-of-the-art matrix exponential software both the P and Q matrices needed in the discrete step invariant state space solution, Equation 3.22, are available. This approach provides a new technique for calculating Q which appears to be superior to previous approaches. This is the algorithm used for obtaining both P and Q in the matrix exponential program documented in ACL Report 8601 (see footnote on page 64). Detailed evaluation and comparison of this algorithm with those presently used to calculate Q has not been carried out but is left as an interesting topic for future study.

3.7 Stability and Accuracy Analysis

In all digital simulations the key goal is to match the dynamics of continuous equations with discrete

equations. It is apparent there is no unique methodology for formulating difference equations. However, no matter what approach is used the accuracy and stability of the simulation algorithm should be considered carefully. The major difference between classical numerical integration methods and input approximation methods can be interpreted from Figure 3.2. Since these two approaches are based on different concepts, it might be expected that the stability and accuracy of these two categories of methods possess different characteristics.

3.7.1 Input Approximation Methods

Input approximation methods, previously called discrete analog methods, first synthesize a discrete system (or sampled-data system) to emulate the continuous system then formulate difference equations of the synthesized discrete system to simulate the continuous system. Recall the discrete analog system is developed by introducing an ideal sampler followed by a data-reconstruction procedure. Due to these operations, the questions of stability and accuracy can be analyzed separately.

Since the system is an open loop system between two

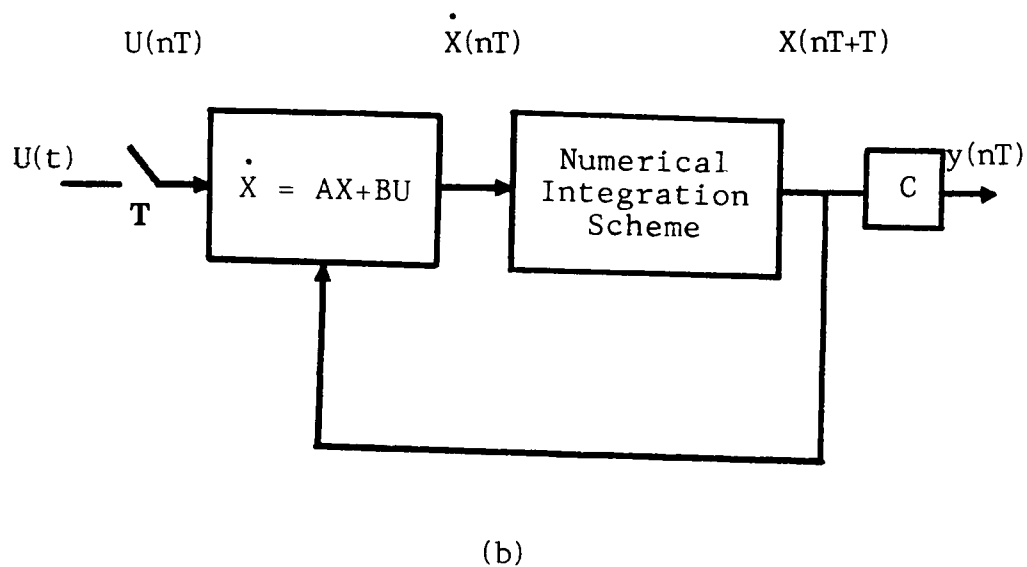
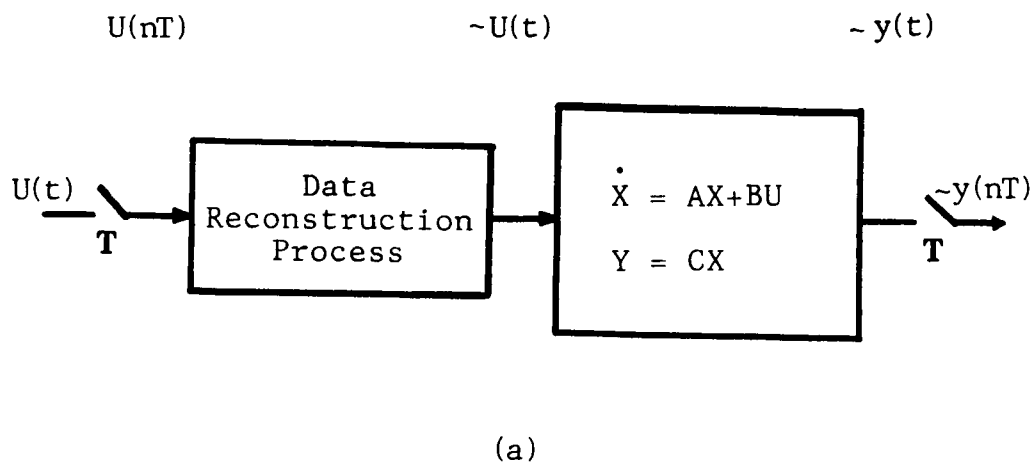


Figure 3.2. Digital Methods for Solving the State Space Equation (a) Input Approximation Method
(b) Numerical Integration Method

sampling instants, the introduction of error due to the sampler will not effect the stability. Therefore, the algorithm's stability is fixed by the stability characteristics between the reconstructed input signal and the last sampler. In short, the difference equation that describes that solution from one sampled period to another will have the same global stability characteristics as the continuous process.

An important feature of the difference equations generated by input approximation methods is that they are intrinsically stable regardless of step size, provided that the simulated process is intrinsically stable.

On the other hand, the accuracy of input approximation methods depends on how closely (accurately) the sampled signal can be reconstructed. Therefore the concern between the input sampler and its data hold is the reconstruction of the input with sufficient fidelity to closely approximate the original input.

3.7.2 Classical Numerical Integration Methods

Stability and accuracy characteristics must be considered together when solving state space equations with numerical integration algorithms, since both the

phase and magnitude distortions due to the discretization of the continuous integrator are feedback. To investigate these distortions between various discrete integrators a variable phase compensator, $G_c(s)$, and a zero-order hold, $H_{zo}(s)$, are cascaded with the continuous integrator, as illustrated in Figure 3.3. By sampling the input and output signals, discrete integrators are derived as:

$$H(z) = \frac{X(z)}{Y(z)} = Z \left[G_c(s) H_{zo}(s) \frac{1}{s} \right] \quad (3.55)$$

where $G_c(s) = e^{rsT}$

$$H_{zo}(s) = \frac{1 - e^{-sT}}{s} \quad (3.56)$$

where r is the variable phase parameter. Therefore, substituting Equations 3.56 into Equation 3.55 gives

$$H(z) = Z \left[e^{rsT} \frac{1 - e^{-sT}}{s} \frac{1}{s} \right] \quad (3.57)$$

and using modified Z-transform to incorporate the phase compensator provides

$$H(z) = \frac{rT + (1-r)T z^{-1}}{1 - z^{-1}} \quad (3.58)$$

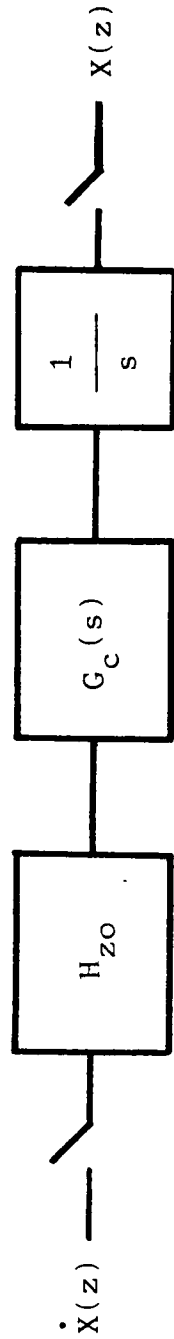


Figure 3.3. Discrete Integrator

By appropriately selecting the r value the form of three common integrators can be obtained from the last equation, the values of r and numerical integrators are:

1. when $r=0$, which means without phase compensation, the Euler method is obtained

$$H(z) = \frac{T z^{-1}}{1 - z^{-1}} \quad (3.59)$$

which gives the difference formula

$$x_{k+1} = x_k + T \dot{x}_k \quad (3.60)$$

2. when $r=0.5$, the Trapezoid method is the result

$$H(z) = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \quad (3.61)$$

which gives the difference formula

$$x_{k+1} = x_k + \frac{T}{2} [\dot{x}_{k+1} + \dot{x}_k] \quad (3.62)$$

3. when $r=1$, the backwards difference method is obtained

$$H(z) = \frac{T}{1 - z^{-1}} \quad (3.63)$$

which gives the difference formula

$$x_{k+1} = x_k + T \dot{x}_{k+1} \quad (3.64)$$

Figure 3.4 illustrates the frequency domain characteristics of these discrete integrators. The bode plot of the discrete transfer function is defined as the magnitude ratio in dB and phase angle in degrees vs. frequency:

$$\text{MAG(dB)} = 20 \log_{10} | H(e^{j\omega n}) | \quad (3.65)$$

$$\text{PHAS(deg)} = [H(e^{j\omega n})] \quad (3.66)$$

Here the normalized frequency, ωn , is used

$$\omega n = \frac{\omega}{\omega_s} \quad (3.67)$$

where ω is the input frequency and ω_s is the sample frequency. The frequency response of the discrete integrators is compared with the continuous integrator in Figure 3.4.

Frequency response analysis provides an analytical indication of the discrete integrators phase and magnitude distortion from the continuous integrator. As

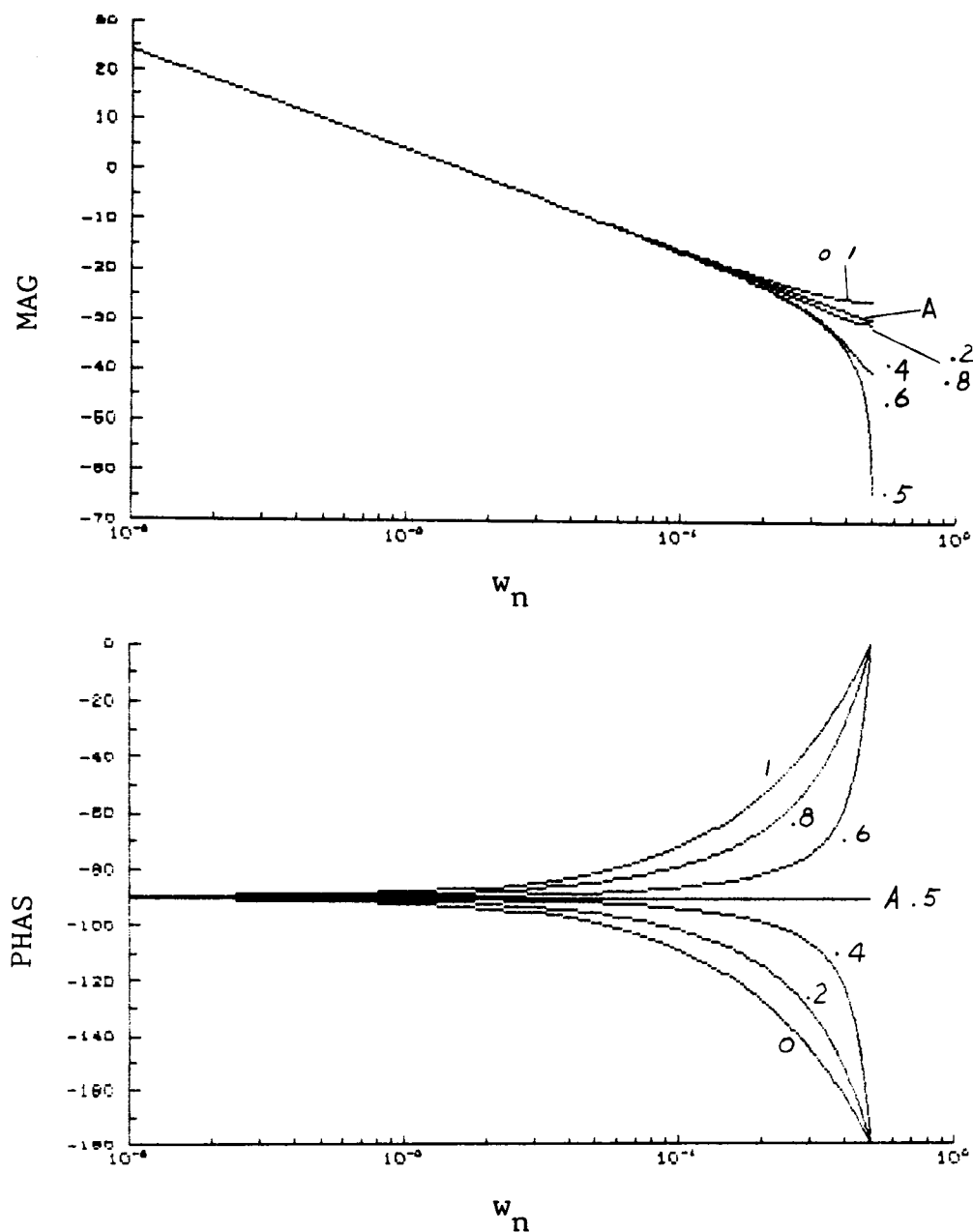


Figure 3.4. Frequency Response of Discrete Integrator.
 A—Continuous Integrator and the Numerical
 Value indicated in Each Curve Associated the
 Compensated Phase Magnitude r

such, it might provide a guideline for selecting the proper numerical integration algorithm. Figure 3.5 shows the magnitude and phase deviations of the discrete integrators from the continuous integrator. The deviation of the magnitude ratio, EMAG, and the deviation of phase angle, EPHAS, are defined as

$$\text{EMAG} = 20 \log_{10} \frac{|H(e^{j\omega n})|}{|G(j\omega n)|} \quad (3.68)$$

$$\text{EPHAS} = \angle H(e^{j\omega n}) - \angle G(j\omega n) \quad (3.69)$$

The EMAG and EPHAS vs. ωn plots provide a clear comparison of the discrete integrators performance with different phase compensation. The best choice of r is around 0.2 and 0.8 from a magnitude standpoint, and the obvious choice of r , from a phase standpoint, is $r=0.5$. In practical applications, one may need to evaluate and trade off the effects of phase and magnitude distortion for the simulation.

This discussion built around data sampling followed by a zero order hold set up a theoretical analysis for the source of error and instability of three numerical integrators. Similar analysis techniques are hard to

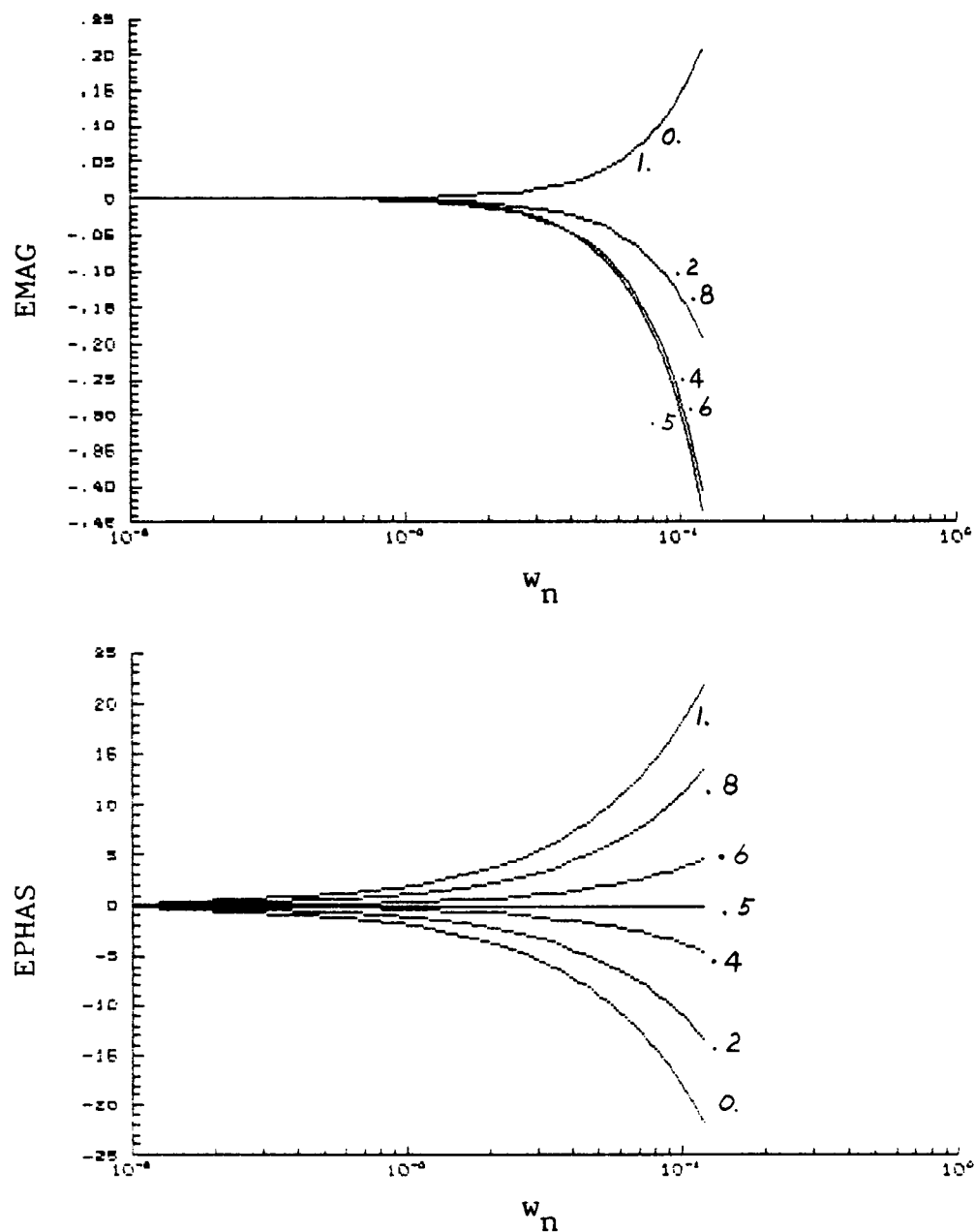


Figure 3.5. The Magnitude and Phase Distortion of Discrete Integrators

extend to higher order numerical integration methods, however, in general, n th-order numerical integrators introduce as much as $n/2$ of the integration step size as lag which is undesirable from the stability viewpoint (34).

From the stability viewpoint, input approximation methods are superior to classical numerical integration methods for solving a state space equation, since the former possesses the feature of being intrinsically stable. However, proceeding accuracy analysis need to include more complex considerations, e.g. the error criterion, computation facility etc., and heavily depends on practical applications.

3.8 Summary

In this chapter, simulation formulas are synthesized for models having predetermined input types, called input invariant discrete models. Input invariant discretizations are developed for piecewise step and ramp inputs. Discrete models are presented for both the external or input-output transfer function descriptions and the interval or state space formulation. These formulas are suitable for simulating both SISO and MIMO cases. Finally the algorithms for matrix exponential computation are

discussed and a new procedure for calculating both the discrete state transition matrix, Q , and the discrete input matrix, P , by taking the matrix exponential of an augmented matrix is introduced and developed.

CHAPTER IV

COMPARISON OF SIMULATION TECHNIQUES WITH ARBITRARY INPUTS

Simulation algorithms which synthesize discrete models to approximate linear system behavior subjected to arbitrary inputs are the topic of this chapter and the main focus of the thesis. In the previous chapter many formulas for digital simulation of linear processes with predetermined input types were derived and discussed. These formulas can work satisfactorily if the input can be categorized into one of the predetermined types addressed. However for many simulations the type of input can not be predetermined or, perhaps, can not be categorized into a type for which a discrete invariant model can be formulated. In such cases one of the discrete invariant algorithms could be used, anyhow, knowing that the signal reconstruction is approximate and thus the simulated response is also approximate. This is very commonly done especially in the use of the step invariant discrete algorithm and the application of the matrix exponential (13). In such applications these formulas need to be employed with even more caution.

The impossibility of synthesizing exact discrete

equivalents for continuous systems was demonstrated in the previous chapter. However, an alternate approach would still be to attempt to address the problem of arbitrary inputs as part of the simulation formulation. Techniques which do this are separated into three classes and discussed below; they are: (1) optimal discrete models and integrators, (2) overall digital redesign, and (3) optimal discretization methods.

4.1 Optimal Discrete Models and Integrators

Some algorithms synthesize discrete models containing parameters that can be adjusted for certain types of inputs (33,35,8). In other words, a discrete model for the given continuous system may not be unique but may be a function of the input. Optimal values for the parameters are generally obtained through an interactive scheme. The procedure is summarized as follows:

1. Define the performance criterion or objective function.
2. Select the form of the discrete model containing the optimizing parameters.
3. Apply an optimization procedure to find the optimal parameters for the discrete model.

A similar effort has been employed to find the optimal discrete representation of the continuous integrator (15,3); a "random search optimization algorithm" was used to determine the optimal parameters. The class of systems being investigated are those which can be represented by the set of state equations

$$\dot{X} = f(X, U)$$

thus, in general, they can be nonlinear ODEs.

In order to use the optimal discrete model approach requires the selection of an optimization technique and algorithm. Therefore the questions of algorithm suitability and verification of optimality need to be at least partially addressed. In addition, optimization routines are too complex to apply in some practical situations. Perhaps the major drawback of optimal discrete models and integrators is the selection of a performance criterion and the dependence of the optimized model on the criterion. Since any time domain criteria is input dependent, it is hard to find a universal performance index in the time domain. The frequency domain criteria does provide an input independent and thus, perhaps, a more general performance index; however, to minimize the frequency response error will not guarantee the time domain error is minimal.

4.2 Overall Redesign

Overall redesign schemes attempt to achieve more accurate results by developing the discrete equivalent model to make it match some specific characteristics of the continuous system. Fowler's method (5,12,11), synthesizes the discrete simulation model by matching its root loci with that of the simulated continuous system. Another scheme proposed by Sage (32) is to approximate the transfer functions of each element in the continuous system (loops) so that the error squared is minimized. Compared to Fowler's method, this technique eliminates the need for inserting a delay in feedback path, since realizability problems have been taken into account during the design procedure (27).

A simulation algorithm built on a point-by-point state comparison method is proposed by Kuo (22,23,24). The system considered is a continuous feedback system. It is synthesized by inserting sample- and hold-devices and then modifying the input gain (or input gain matrix) and feedback gain (or feedback gain matrix) of the sampled-data model such that its response is as close as possible in the time-domain to that of original system at every sampling instant. However, the point-by-point matching scheme is based on the step input. Thus, the performance of the sampled-data system is only good when the input frequency is sufficiently low in comparison

to the sampling frequency. Instead of matching the time-domain performance, Rattan (29,30) proposed a similar scheme based on matching the frequency response. Rattan developed his method by inserting a forward path compensator, rather than changing system's gains, which makes the overall frequency response of sampled-data system as close as possible to that of the continuous system. Rattan's technique matches the continuous output more closely than that of Kuo's method for high frequency inputs.

Overall redesign techniques use either time or frequency domain optimal schemes to eliminate the discretization distortion while converting a continuous system to a discrete system. These methods do provide accurate results with respect to their stated objective function; however their performance in a broader context (e.g. at both high and low frequencies) is not as satisfactory. In addition, the complexity of the design procedures makes their general application to simulation building somewhat restrictive.

4.3 Optimal Discretization Methods

The impossibility of synthesizing exact discrete equivalents for continuous systems has been discussed. However, the "best" discretization technique for

arbitrary inputs is of considerable interest. Any information or guidelines that can be provided during the building stage of a simulation would be of great utility. Perhaps such information can be gained by comparing both the time- and/or frequency-domain performance of discrete models at their design stage. The overall redesign methods treated above only addressed performance criteria in one of the two domains.

The next section reviews a recent discretization algorithm, the Taylor method (9,10) which incorporates both time and frequency domain fitting into its development. This new algorithm is one selected for the numerical study of linear simulation techniques.

4.3.1 Development of the Taylor Method

A matching pole-zero method is recommended by Edwards (19,6). An obstacle in employing such a mapping technique is that it is not possible to give an easy formula for mapping infinite zeroes in the s -plane to corresponding zeros in the z -plane. The zeroes in the z -plane depend on the discrete approximation function used to represent the continuous input. Zeroes which are mapped directly from the s -plane onto the z -plane will not guarantee accuracy in the time-domain

performance, unless the sampling period is very small (1). Again, this points out that from an overall simulation point of view, discrete models which are based on both time-domain and frequency-domain considerations might be expected to be better than those based on one criteria alone.

Irby (16) presented a technique for numerical solution of a class of high order differential equations based on a linear combination of Taylor series expansions. Recently, Forsythe (10,9) extensively applied this technique to develop a new algorithm, called the Taylor method that does possess considerations both in the time- and frequency-domain. The concept is that the poles of the continuous transfer function are mapped directly onto the z -plane, but the positions of z -plane zeroes are derived by a more complex procedure involving the use of the Taylor series expansion. Mapping the poles directly from the s -plane onto the z -plane will preserve the stability relation between the discrete and continuous transfer function, which is a frequency-domain consideration. Using the Taylor series expansion to locate the zeroes' positions in the z -plane is directed at making the dynamic response of discrete transfer function as close to that of continuous transfer function as possible, which is

the time-domain consideration. To investigate this relatively new method consider a continuous transfer function, $G(s)$,

$$G(s) = \frac{Y(s)}{X(s)} = \frac{n_0 + n_1s + n_2s^2 + \dots + n_ps^p}{m_0 + m_1s + m_2s^2 + \dots + m_qs^q} \quad (4.1)$$

the corresponding differential equation is

$$\begin{aligned} m_0y + m_1y + m_2y + \dots + m_qy \\ = n_0x + n_1x + n_2x + \dots + n_px \end{aligned} \quad (4.2)$$

and polynomials of dependent and independent variables are defined as:

$$w_x = m_0y + m_1y + m_2y + \dots + m_qy$$

$$w_y = n_0x + n_1x + n_2x + \dots + n_px$$

If the equivalent discrete transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_kz^{-k}}{1 + b_1z^{-1} + b_2z^{-2} + \dots + b_rz^{-r}} \quad (4.3)$$

the corresponding difference equation is

$$\begin{aligned}
 y(t) + b_1 y(t-T) + b_2 y(t-2T) + \dots + b_r y(t-rT) \\
 = a_0 x(t) + a_1 x(t-T) + a_2 x(t-2T) + \dots + a_k x(t-kT)
 \end{aligned}
 \tag{4.4}$$

Expanding each functional evaluation for y in the left hand side of Equation 4.4 with a Taylor series expansion provides

$$\begin{aligned}
 y(t) &= y(t) \\
 y(t-T) &= y(t) - T y'(t) + \frac{T^2}{2!} y''(t) - \dots \\
 y(t-2T) &= y(t) - 2T y'(t) + \frac{4T^2}{2!} y''(t) - \dots \\
 &\cdot \\
 &\cdot \\
 &\cdot \\
 y(t-rT) &= y(t) - rT y'(t) + \frac{r^2 T^2}{2!} y''(t) - \dots
 \end{aligned}$$

then substituting these expansions into the left hand side of Equation 4.4 gives

$$y(t) + b_1 y(t-T) + b_2 y(t-2T) + \dots + b_r y(t-rT)$$

$$= \begin{bmatrix} 1 \\ +b_1 \\ +b_2 \\ \vdots \\ +b_r \end{bmatrix} y(t) + T \begin{bmatrix} -b_1 \\ -2b_2 \\ -3b_3 \\ \vdots \\ -rb_r \end{bmatrix} y'(t) + \frac{T^2}{2!} \begin{bmatrix} b_1 \\ +4b_2 \\ +9b_3 \\ \vdots \\ +r^2 b_r \end{bmatrix} y''(t) + \dots$$

(4.5a)

Likewise making a Taylor series expansion for all functional evaluations of x in the right hand side of Equation 4.4 leads to

$$a_0 x(t) + a_1 x(t-T) + a_2 x(t-2T) + \dots + a_k x(t-kT)$$

$$= \begin{bmatrix} a_0 \\ +a_1 \\ +a_2 \\ \vdots \\ +a_k \end{bmatrix} x(t) + T \begin{bmatrix} -a_1 \\ -2a_2 \\ -3a_3 \\ \vdots \\ -ka_k \end{bmatrix} x'(t) + \frac{T^2}{2!} \begin{bmatrix} a_1 \\ +4a_2 \\ +9a_3 \\ \vdots \\ +k^2 a_k \end{bmatrix} x''(t) + \dots$$

(4.5b)

A set of auxiliary coefficients, f_i , is introduced into Equation 4.2,

$$\begin{aligned} & f_0(m_0 y + m_1 y^{(1)} + \dots + m_q y^{(q)}) + f_1 w_y^{(1)} + f_2 w_y^{(2)} + \dots \\ &= f_0(n_0 x + n_1 x^{(1)} + \dots + n_p x^{(p)}) + f_1 w_x^{(1)} + f_2 w_x^{(2)} + \dots \end{aligned} \quad (4.6)$$

Expanding Equation 4.6 yields

$$(f_0 m_0) y + \begin{bmatrix} f_0 m_1 \\ +f_1 m_0 \end{bmatrix} y' + \begin{bmatrix} f_0 m_2 \\ +f_1 m_1 \\ +f_2 m_0 \end{bmatrix} y'' + \dots \quad (4.7a)$$

$$= (f_0 n_0) x + \begin{bmatrix} f_0 n_1 \\ f_1 n_0 \end{bmatrix} x' + \begin{bmatrix} f_0 n_2 \\ f_1 n_1 \\ f_2 n_0 \end{bmatrix} x'' + \dots$$

or

$$\begin{aligned} & f_0 m_0 y + \sum_{l=1}^{\infty} \left(\sum_{i=0}^l f_i m_{l-i} \right) y^{(l)} \\ &= f_0 n_0 x + \sum_{v=1}^{\infty} \left(\sum_{i=0}^v f_i n_{v-i} \right) x^{(v)} \end{aligned} \quad (4.7b)$$

Equating Equation 4.5a to the left hand side of 4.7 and Equation 4.5b to the right hand side of Equation 4.7 gives Equations 4.8 and 4.9 respectively,

$$\begin{bmatrix} 1 & 1 & 1 & . & . & . & 1 \\ 0 & -T & -2T & . & . & . & -rT \\ 0 & \frac{T^2}{2!} & \frac{4T^2}{2!} & . & . & . & \frac{r^2 T^2}{2!} \\ 0 & \frac{-T^3}{3!} & \frac{-8T^3}{3!} & . & . & . & \frac{-r^3 T^3}{3!} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ b_2 \\ \vdots \\ b_r \end{bmatrix}$$

(4.8)

$$= \begin{bmatrix} m_0 & 0 & 0 & \dots \\ m_1 & m_0 & 0 & 0 & \dots \\ m_0 & m_1 & m_0 & 0 & 0 & \dots \\ \vdots & & & & & \\ \vdots & & & & & \\ m_q & m_{q-1} & \dots & m_0 & 0 & \dots \\ 0 & m_q & \dots & & & m_0 \\ . & & & & & \\ . & & & & & \\ . & & & & & \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ . \\ . \\ . \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & -T & -2T & \dots & -kT \\ 0 & \frac{T^2}{2!} & \frac{4T^2}{2!} & \dots & \frac{k^2 T^2}{2!} \\ 0 & \frac{-T^3}{3!} & \frac{-8T^3}{3!} & \dots & \frac{-k^3 T^3}{3!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_k \\ \vdots \end{bmatrix}$$

(4.9)

$$= \begin{bmatrix} n_0 & 0 & 0 & \dots & 0 \\ n_1 & n_0 & 0 & 0 & \dots \\ n_2 & n_1 & n_0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ n_p & \dots & \dots & n_0 & 0 \\ 0 & n_p & \dots & \dots & n_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

Since the s-plane poles are mapped directly onto the z-plane, the order of the denominator of the discrete transfer function is the same as that of the continuous transfer function, therefore, $r = q$ in Equation 4.8 (see also Equation 4.2 and 4.4). If k numerator or a-coefficients of the discrete transfer function (see Equation 4.3 or 4.5b) are wanted, there must be k equations to produce them; therefore, k f-coefficients are needed in Equations 4.6 through 4.9. At this point, the b-coefficients in $H(z)$ are determined from the poles of the continuous transfer function using the s- and z-plane mapping relationship $z = \exp(sT)$, i.e. Equation 2.24. Then, it remains to obtain the corresponding a-coefficients provided via Equations 4.8 and 4.9. Updating Equation 4.8 by accounting for $r = q$ and Equation 4.9 for choosing k a-coefficients in the discrete transfer function gives Equations 4.10 and 4.11, respectively,

$$\begin{bmatrix}
 1 & 1 & 1 & \dots & 1 \\
 0 & -T & -2T & \dots & -qT \\
 0 & \frac{T^2}{2!} & \frac{4T^2}{2!} & \dots & \frac{(-qT^2)}{2!} \\
 0 & \frac{-T^3}{3!} & \frac{-8T^3}{3!} & \dots & \frac{(-qT)^3}{3!} \\
 \vdots & & & & \\
 0 & \frac{(-T)^k}{k!} & \frac{(-2T)^k}{k!} & \dots & \frac{(-qT)^k}{k!}
 \end{bmatrix}
 \begin{bmatrix}
 1 \\
 b_1 \\
 b_2 \\
 \vdots \\
 b_q
 \end{bmatrix}$$

(4.10)

$$= \begin{bmatrix}
 m_0 & 0 & 0 & \dots \\
 m_1 & m_0 & 0 & 0 & \dots \\
 m_2 & m_1 & m_0 & 0 & 0 & \dots \\
 \vdots & & & & & \\
 \vdots & & & & & \\
 \vdots & & & & & \\
 m_q & \dots & & m_0 & 0 & \dots \\
 \vdots & & & & & \\
 \vdots & & & & & \\
 0 & & m_q & \dots & & m_0
 \end{bmatrix}
 \begin{bmatrix}
 f_0 \\
 f_1 \\
 f_2 \\
 \vdots \\
 f_k
 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & -T & -2T & \dots & -kT \\ 0 & \frac{T^2}{2!} & \frac{4T^2}{2!} & \dots & \frac{(-kT)^2}{2!} \\ 0 & \frac{-T^3}{3!} & \frac{-8T^3}{3!} & \dots & \frac{(-kT)^3}{3!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{(-T)^k}{k!} & \frac{(-2T)^k}{k!} & \dots & \frac{(-kT)^k}{k!} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_k \end{bmatrix}$$

(4.11)

$$= \begin{bmatrix} n_0 & 0 & 0 & \dots \\ n_1 & n_0 & 0 & 0 & \dots \\ n_2 & n_1 & n_0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_p & \dots & & n_0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & & n_p & \dots & & n_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_k \end{bmatrix}$$

A summary of the methodology to establish the Taylor Series algorithm is:

1. Select the form of the discrete transfer function, that is, select the number of terms in

numerator or a-coefficients of $H(z)$, Equation 4.3.

2. Find the poles of the continuous transfer function, Equation 4.1.
3. Determine the denominator coefficients of the discrete transfer function by mapping the poles on the s-plane directly onto the z-plane. This is done by employing the pole matching technique given by Equation 2.34.
4. As the b-coefficients are now known, find the auxiliary coefficients, f_i , utilizing Equation 4.10.
5. Finally with the f-coefficients known, compute the numerator coefficients, a_i , of the discrete transfer function from Equation 4.11.

4.4 Comparison of Discretization Methods

The major task undertaken by this research is a comparison of linear simulation techniques to answer some of the questions that arise in building and using a simulation. This kind of study has not been presented in the current literature. It is felt it will be of value especially in view of the wide range of techniques proposed and the continuing activity in the area. The ultimate purpose of such a study is important in establishing guidelines for selecting the "best"

discretization techniques for a given simulation task.

The derivation of a discrete equivalent model is basically a fairly simple task; once a discretization algorithm is selected. However, the major question of whether this discrete model is of any practical use to the engineer for simulation and application purposes remains. The accuracy capabilities of the discrete model varies with the discretization technique used and upon the specific application, but it still seems reasonable that some relative precision level can be established in comparing some of the most promising techniques.

The remainder of this section deals with those discretization techniques selected for the comparison study used to provide guidelines to successful simulation.

4.4.1 Comparison Criteria

One of the biggest obstacles in making a comparison study is the definition of a general performance criteria, this arises because the time domain error is input dependent making it difficult to obtain a general criteria suitable for all inputs. In contrast, frequency domain performance is the preferred comparison criteria for a lot of investigators. There are unique

relationships between the time and frequency response characteristics; however, the frequency domain response can not give us a direct feeling for the transient time domain performance. Still our ultimate purpose is to find an optimal time domain discretization technique.

If an ideal discrete equivalent transfer function is defined as one whose response matches that of the continuous transfer function at every sample instant (see Equation 3.5), then a time domain performance criteria seems to be the natural choice.

Two time domain error indexes for the open loop response are selected as objective functions for the purposes of this research. They provide the major criteria for quantifying the comparisons and in making the judgement as to the best digital simulation algorithms. One error index is directed at the local error while the other index is concerned with the global error. The definition of the two error indexes are the error, ERR, and normalized error, NERR:

$$(ERR)_k = y_k^* - y_k \quad (4.12)$$

$$(NERR)_k = \frac{1}{k} \sum_{i=1}^k | (ERR)_i | \quad (4.13)$$

where y^* is the discrete model output while y is the continuous process output at each sample instant, and k

is the running sample interval number. The error, ERR , provides an indication of the local error, whereas, the normalized error, $NERR$, gives information on the global error.

A secondary comparison in the frequency domain will also be given in terms of the Bode plot and the errors in the Bode plot as defined in Equations 3.65 through 3.69. This comparison is included largely due to the pervasiveness of this criteria in the associated literature.

Another important decision is the selection of the input type used as the forcing function. The sine of unit magnitude, $\sin(\omega t)$, is the input chosen for this study. The frequency of the input is varied, as well as, the ratio of the sampling frequency to the input frequency, defined as the normalized frequency by Equation 3.67. In this way, insight into establishing discrete equivalent models which are most suitable for arbitrary inputs might be gained, and thus, the discretization methods are more properly evaluated.

Although these error functions for the chosen input do provide a quantification for how close the response of the discrete equivalent model is to that of the continuous system, unfortunately, they by no means offer

an absolute verdict in determining the adequacy of the discretization techniques.

4.4.2 Selected Discretization Methods and Linear Models

Six discretization methodologies were selected to include in the numerical study. They are:

1. Tustin method.
2. Matched pole-zero method with infinite zeroes mapped to $z=0$.
3. Matched pole-zero method with infinite zeroes mapped to $z=-1$.
4. Input approximation method with zero order hold and phase compensator.
5. Input approximaition method with triangular hold.
6. Taylor method.

The numbering used in this list will be used as a legend in presenting simulation results throughout the chapter. "N" will be used instesd of "6" for the Taylor method and the analytical solution will be designated with "A". The six discretization methodologies were chosen to encompass those most commonly used in engineering simulation (algorithm 4 with $r = 0$), those that are very easy to implement (algorithm 1), those that have received considerable attention in the

literature (algorithm 1, 2, 3), and those that look most promising for successful simulation (algorithm 4, 5, 6).

The basis of all these methods has been introduced in chapter II and first portion of this chapter. The 4th method is the extensive utilization of the input approximation technique, and its derivation will be presented in detail in section 4.5.4.

The three linear models chosen for the study were: I. First order lag, II. Underdamped second order lag, and III. Nonminimal phase second order process. The first two processes contain the two most basic classes of process characteristics that are often used to describe higher order processes. The third process represents a mixture of two first order processes whose gains and time constant ratios are such that the process is nonminimum phase.

Thus the study is formed around applying six different discretization methods to the three different continuous process models.

4.5 Comparison I

The first comparison, called Comparison I, made for six discretization techniques will be based on a first order lag system,

$$G(s) = \frac{5}{s + 5} \quad (4.14)$$

The corresponding discrete models are derived, in turn, in the following six sections.

4.5.1 Tustin Method

To apply the Tustin method simply substitute for the continuous operator, s , in both the numerator and denominator of the continuous transfer function the relation $s = T(z+1)/2(z-1)$ to give (see Equation 2.29)

$$\begin{aligned}
 H(z) &= \frac{5}{s + 5} \bigg| \frac{T}{2} \frac{z + 1}{z - 1} \\
 &= \frac{Tz + T}{(aT + 2)z + (aT - 2)}
 \end{aligned}
 \tag{4.15}$$

or upon rearranging to give

$$H(z) = \frac{\left(\frac{T}{aT + 2}\right) + \left(\frac{T}{aT + 2}\right) z^{-1}}{1 + \left(\frac{aT + 2}{aT + 2}\right) z^{-1}}
 \tag{4.16}$$

4.5.2 Match Pole-Zero Method with Infinite Zeroes Mapped onto $z=0$

The first step of match pole-zero method is mapping the s-plane poles and zeroes directly onto the z-plane. For the first order system, $G(s)$, consisting of a simple pole at -5 , the pole is mapped onto z-plane according to Equation 2.34

$$z_{\text{pole}} = e^{sT} \Big|_{s = -5} \quad (4.17)$$

and the one infinite zero in the s-plane is mapped onto zero in the z-plane. Thus, the discrete equivalent transfer function is

$$H(z) = \frac{Kz}{(z - e^{-5T})} \quad (4.18)$$

where K is used to adjust the steady state value of the discrete system to agree with that of the continuous system. In this study, the K is chosen to match the discrete system's unit step response with that of continuous system's. According to the final value theorem

$$\left. \frac{a}{s + a} \right|_{s \rightarrow 0} = \left. \frac{K}{z - e^{-5T}} \right|_{z \rightarrow 1} \quad (4.19)$$

K is therefore

$$K = 1 - e^{-5T}$$

and the final form of the discrete transfer function is

$$H(z) = \frac{(1 - e^{-5T}) z}{z - e^{-5T}} \quad (4.20)$$

or upon rearrangement is

$$H(z) = \frac{1 - e^{-5T}}{1 - e^{-5T} z^{-1}} \quad (4.21)$$

4.5.3 Match Pole-Zero Method with Infinite Zeroes Mapped Onto $z=-1$

Remembering that the mapping relationship between the s- and z-plane, $z=\exp(sT)$, maps "the primary strip"

on the s-plane into the unit circle on z-plane, thus, mapping the infinite s-plane zeroes to the point $z=-1$ will make the frequency response at higher frequencies of the discrete transfer function similar to that of the continuous transfer function. Intuitively, the mapping will be more suitable as sampling period approaches zero.

The discrete transfer function has the same denominator as Equation 4.21 but has its zero at -1, instead of 0,

$$H(z) = \frac{K(z + 1)}{z - e^{-5T}} \quad (4.22)$$

Again, applying the final value theorem to find K, the final form of discrete transfer function is

$$H(z) = \frac{a_0 + a_1 z^{-1}}{1 - e^{-5T} z^{-1}} \quad (4.23)$$

where:

$$a_0 = a_1 = \frac{1 - e^{-5T}}{2}$$

4.5.4 Input Approximation Method with Zero-Order Hold and Adjustable Phase Compensator

Sampled-data systems with the zero-order hold reconstruction of the input signal will introduce as

much as a half period of time delay in the reconstructed signal (2). This signal reconstruction distortion is a principle source of numerical error and instability in the simulation algorithm. To address this drawback, a compensator is inserted to eliminate the phase shift and a modified discrete transfer function is derived. Since the best phase compensation is under investigation, a variable phase compensator is introduced. The synthesizing procedure is presented below and the configuration of the modified sampled-data process is illustrated in Figure 4.1. This figure shows

$$H(z) = Z[H_{zo}(s) G_c(s) G(s)] \quad (4.24)$$

where: $H_{zo}(s)$ is the zero-order hold transfer function $(1-\exp(sT))/s$.

$G_c(s)$ is the phase compensator transfer function, $\exp(rsT)$, where r is a number between 0 and 1.

$G(s)$ is the continuous system transfer function

For the 1st-order lag transfer function, Equation 4.24 becomes

$$H(z) = (z - 1) z^{\frac{e^{rsT} G(s)}{s}} \quad (4.25)$$

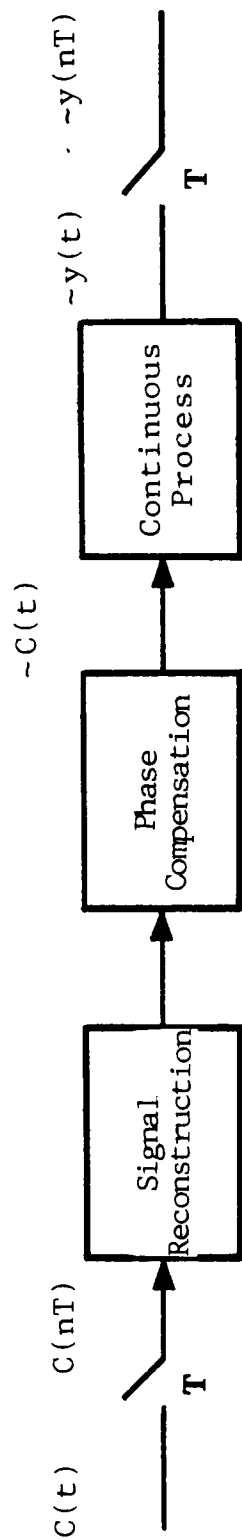


Figure 4.1. System Configuration of Input Approximation Method

The discrete transfer function can be found from the modified Z-transform table (21), to give for $H(z)$,

$$H(z) = \frac{(1 - e^{-arT}) + (e^{-arT} - e^{-aT}) z^{-1}}{1 - e^{-aT} z^{-1}} \quad (4.26)$$

4.5.5 Input Approximation Method with Triangular Hold

Discrete transfer functions synthesized by the input approximation method with triangular hold are available directly in Table 3.2.

4.5.6 Taylor Method

The following step-by-step derivation illustrates Taylor method for the 1st- order lag system:

Step 1. Choose the form of the discrete transfer function to have three numerator coefficients,

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1}} \quad (4.27)$$

The numerator was structured to have three terms by carrying out some preliminary comparisons of Taylor models having 2, 3, and 4 terms. The case with two numerator coefficients gave relatively poor results while the use of 4 a-coefficients over 3 a-coefficients

showed only a small gain. Thus the Taylor method with three numerator coefficients was selected.

This procedure is illustrated in comparison III.

Step 2. The pole of $G(s)$, $s = -5$, is mapped directly onto the z -plane, thus the pole of $H(z)$ is

$$z = \exp(-5T)$$

and

$$b_1 = -\exp(-5T)$$

Step 3. $H(z)$ has three a -coefficients, therefore three f -coefficients are required. Expressing Equations 4.10 and 4.11 in matrix form gives

$$R'B = MF \quad (4.28)$$

and

$$RA = NF \quad (4.29)$$

where

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -e^{-5T} \end{bmatrix}$$

$$M = \begin{bmatrix} m_0 & 0 & 0 \\ m_1 & m_0 & 0 \\ m_2 & m_1 & m_0 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 5 & 0 \\ 0 & 1 & 5 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 1 \\ 0 & -T \\ 0 & \frac{T}{2!} \end{bmatrix}$$

$$R' = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -T & -2T \\ 0 & \frac{T^2}{2!} & \frac{4T^2}{2!} \end{bmatrix}$$

and T is the sample interval.

If for example $T=0.05$, A can be computed from Equations 4.28 and 4.29:

$$A = R^{-1} N M^{-1} R' B = \begin{bmatrix} .0967 \\ .1428 \\ -.0184 \end{bmatrix}$$

The corresponding difference equation is:

$$Y(k) = .7788 y(k-1) + .0968 x(k) \\ + .1428 x(k-1) - .0184 x(k-2)$$

The coefficients of $H(z)$ for different value of T are listed in Table 4.1.

4.5.7 General Form of Discrete Transfer Functions for First-Order Lag

The discrete transfer functions which are derived from methods (1) to (5), see Equations 4.16, 4.21, 4.23, and 4.26 respectively, all possess the following form:

$$H(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}} \quad (4.30)$$

The coefficients of $H(z)$ in Equation 4.30 for different discretization techniques are summarized in Table 4.2.

The discrete transfer function obtained from the Taylor method was designed to fit Equation 4.27 repeated here

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1}}$$

comparing this equation to Equation 4.30 shows the simulation algorithm derived from the Taylor method has another a -coefficient or another zero. The fact that this scheme has another parameter should give the

Table 4.1. Coefficients of the Discrete Model In
Comparison I Synthesized by Taylor Method

T	a0	a1	a2
0.1	0.1804	0.2457	-0.0326
0.05	0.0968	0.1428	-0.0184
0.02	0.0404	0.0626	-0.0079

Table 4.2. Coefficients of Discrete Models in Comparison I

Method	Coefficient
Tustin	$a0 = aT/(aT+2)$ $a1 = aT/(aT+2)$ $b1 = (aT-2)/(aT+2)$
Matched pole-zero (z=0)	$a1 = 1 - \exp(-aT)$ $a0 = 0$ $b1 = -\exp(-aT)$
Match pole-zero	$a0 = (1 - \exp(-aT))/2$ $a1 = (1 - \exp(-aT))/2$ $b1 = -\exp(-aT)$
Input- Approximation with Hzo	$a0 = 1 - \exp(-aT)$ $a1 = \exp(-aT) - \exp(-aT)$ $b1 = -\exp(-aT)$
Input- Approximation with Htr	$a0 = (aT - 1 + \exp(-aT))/(aT)$ $a1 = (1 - \exp(-aT) - aT \exp(-aT))/(aT)$ $b1 = -\exp(-aT)$

Defined Constants:

T = simulation step size
 a = 5
 r = magnitude of the phase compensator

For 1st-order System:

$$G(s) = \frac{a}{s + a}$$

resulting simulation algorithm some advantages over the other methods. Recall that according to this technique we can pick the number of a -coefficients desired. It should be noted that method 4 incorporates a variable phase lead compensator, and thus, also has another adjustable parameter even though it has only two a -coefficients. The effect of the number of a -coefficients used with the Taylor formulation is partially addressed in section 4.7.

4.5.8 Comparison I

Since the input forcing function is arbitrary, the selection of the sampling rate will first be based on the system's dynamic characteristics. For a 1st-order lag system, the system's dynamics are characterized by its rise time, T_r , or time constant. To simulate its dynamic behavior, it is reasonable to select the sample rate to be more than 2 samples per rise time. For the present model the rise time is 0.2. Thus, the sampled periods selected to use in simulation were 0.1, 0.05, and 0.02 which correspond to 2, 4 and 10 samples per rise time respectively.

First the frequency response of each discrete transfer function will be compared with that of the continuous transfer function. The input approximate

method with zero-order hold and various phase compensators are investigated first to select the best amount of phase compensation, i.e. select the best value for r . Figures 4.2 and 4.3 show these results where A labels the continuous transfer function frequency response. The numerical value indicated on the other curves is the r value used in the phase compensator. When $r=0$ no phase compensation is used, as the value of r increases more phase compensation is included up to $r=1$ where one sample period of compensation is incorporated into the algorithm.

MAG(dB), PHAS(deg), EMAG(dB), and EPHAS(deg) are of the same definition as those in Chapter III, Equations 3.65 to 3.69. The best choice of r is around 0.8 from the magnitude standpoint and around 0.5 from phase angle standpoint. In the following Comparisons, I, II, III, the value used in method 4 was selected by trial and error to eliminate the phase lag due to the zero-order hold.

The coordinate in these figures and the remainder of the frequency response figures in this work is that of the normalized frequency W_n , defined above in Equation 3.67 as the ratio of the input frequency to the sample frequency. It is important to keep in mind that normalized frequency is used in presenting the results

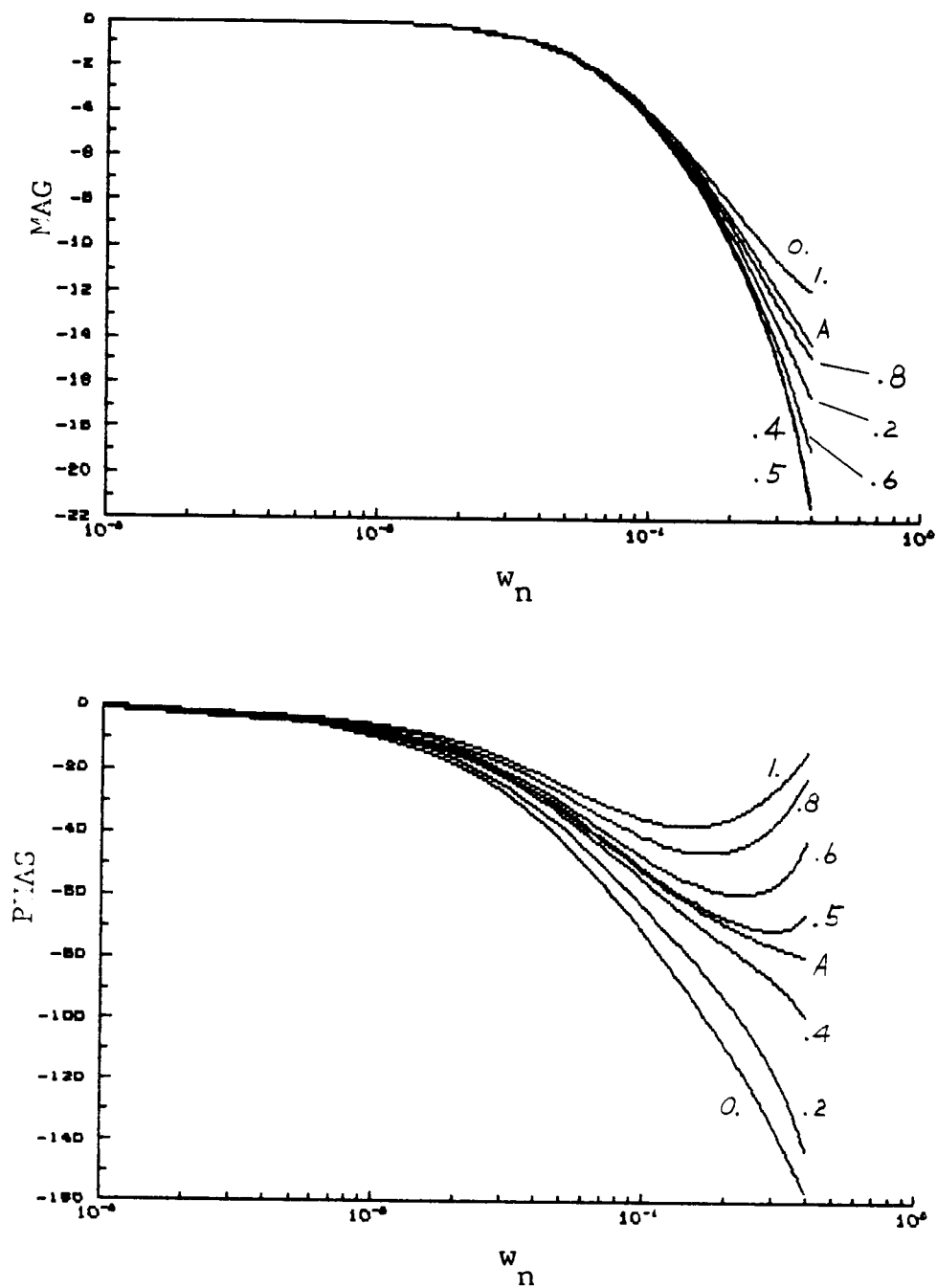


Figure 4.2. Frequency Response of 1st-Order System with Input Approximation Method with Zero-Order Hold and Various Phase Compensation ($T=0.1$)

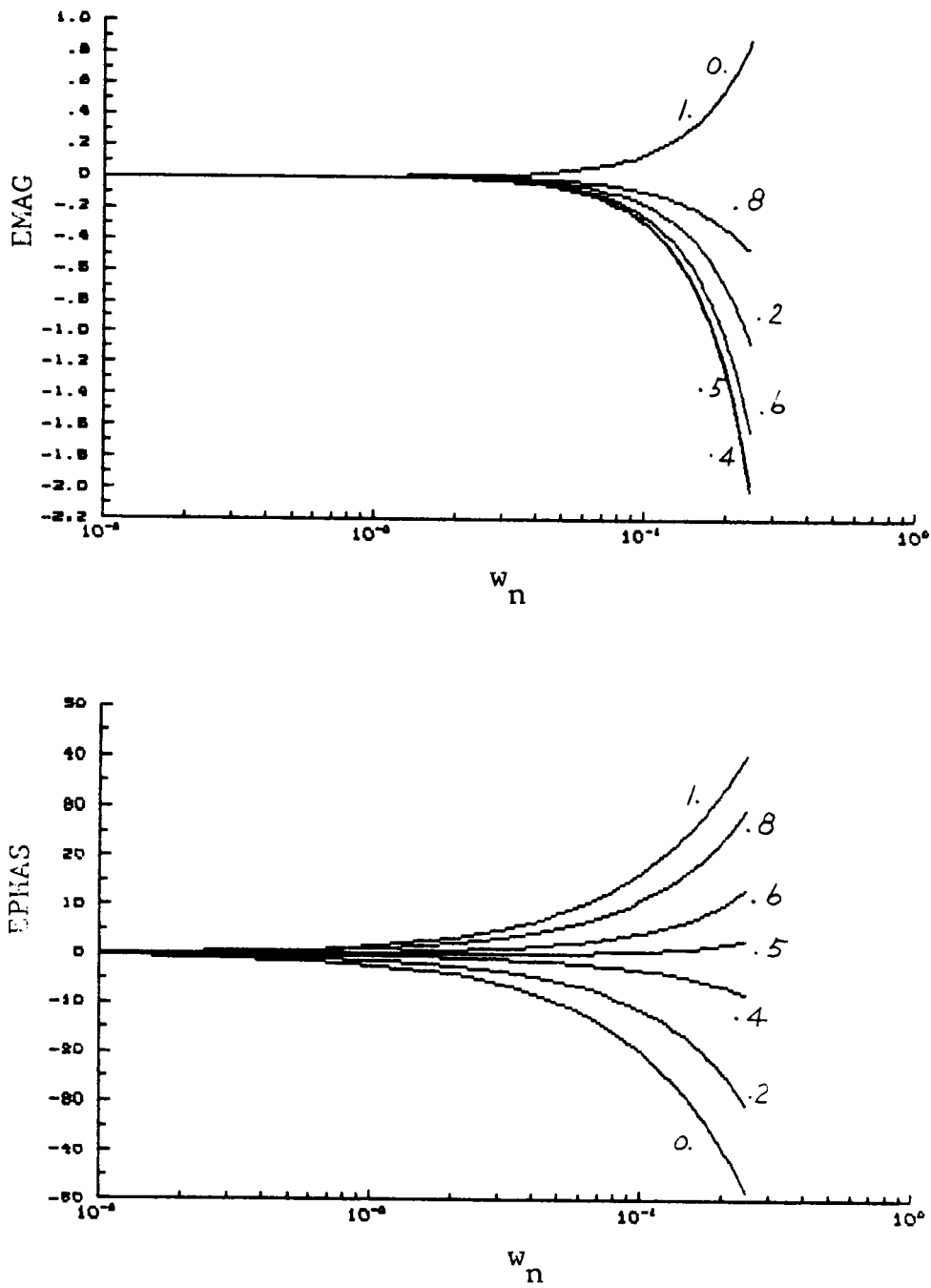


Figure 4.3. EMAG and EPHAS Plots of Input Approximation Method with Zero-Order Hold and Various Phase Compensation ($T = 0.1$)

in figure form as the same numerical value of ω_n on different figures does not correspond to the same input frequency unless the two figures are for the same sampling frequency, ω_s , or sample period, T .

Figure 4.4 illustrates the Bode plot of all five discrete equivalent models along with the continuous transfer function for a sample period of 0.1. Figure 4.5 presents the same data as the EMAG (see Equation 3.68) and EPHAS (see Equation 3.69) up to $\omega_n=0.125$. Table 4.3 gives the legend employed in these figures and the remainder of frequency response figures in this chapter.

These plots provide an initial evaluation of the discretization technique in which method 2 and method 4 have the worst performance from the standpoint of phase shift. The frequency response of method 2 was left off Figures 4.5 to 4.7 so that scale could be expanded in looking at the remaining results. Figures 4.6 and 4.7 contain additional Bode plots for different sample periods. Conclusions from this study on the given 1st-order lag model are:

1. From magnitude attenuation standpoint, the best choice is the Tayler method, while all other methods relatively had much more error.
2. From the phase angle standpoint, the input

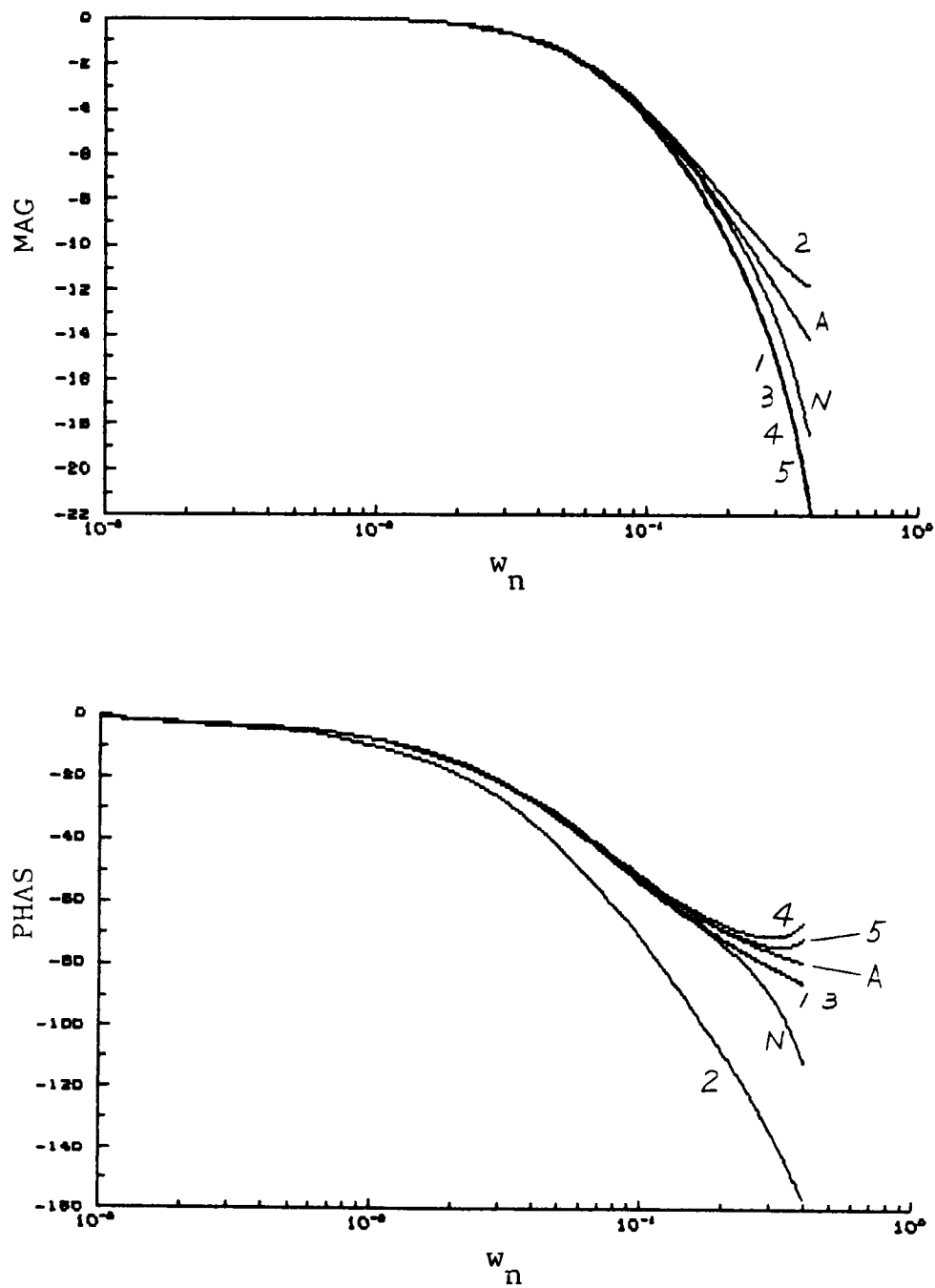


Figure 4.4. MAG and PHAS Plots of 1st-Order System
($T = 0.1$, $r = 0.5$)

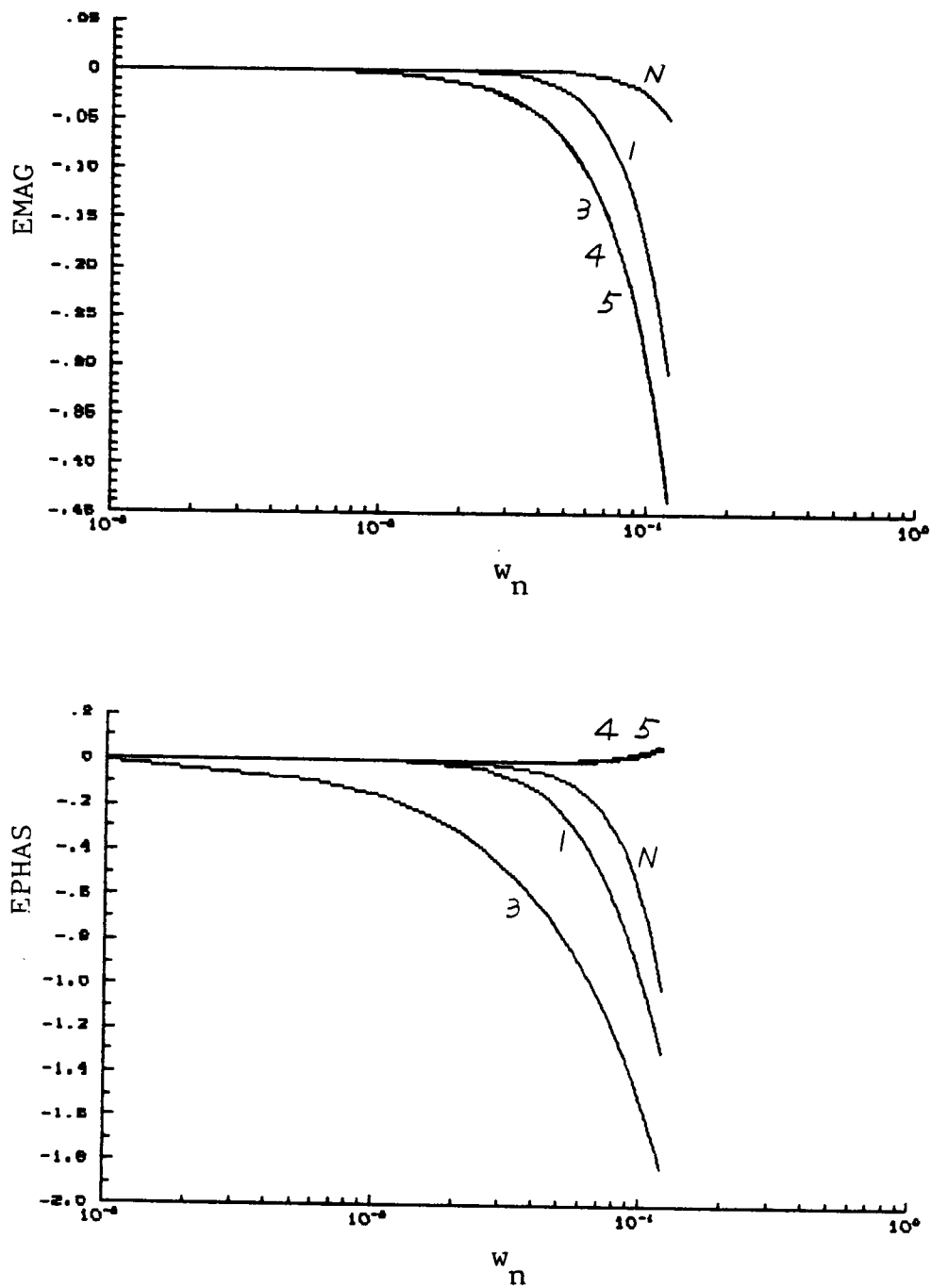


Figure 4.5. EMAG and EPHAS Plots of 1st-Order System
($T = 0.1$, $r = 0.479$)

Table 4.3. List of Discretization Methods

-
- 1: Tustin's Method.
 - 2: Match Pole-Zero Method with Infinite Zeroes Mapped to $z = 0$.
 - 3: Match Pole-Zero Method with Infinite Zeroes Mapped to $z = -1$.
 - 4: Input Approximation Method with Zero-Order Hold and Phase Compensator.
 - 5: Input Approximation Method with Triangular Hold.
 - A: Continuous Transfer Function.
 - N: Taylor Method.
-

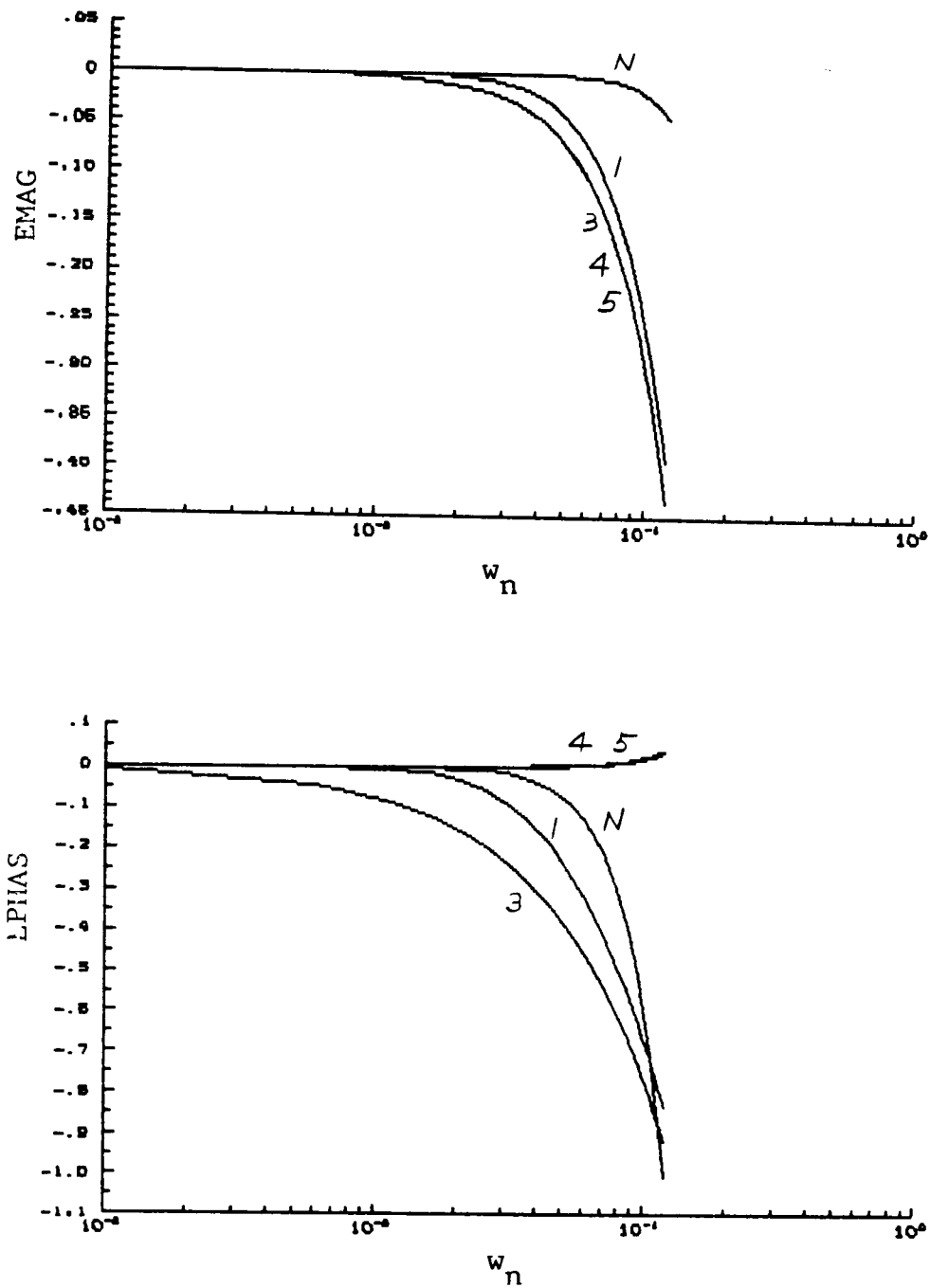


Figure 4.6. EMAG and EPHAS Plots of 1st-Order System
($T = 0.5$, $r = 0.4895$)

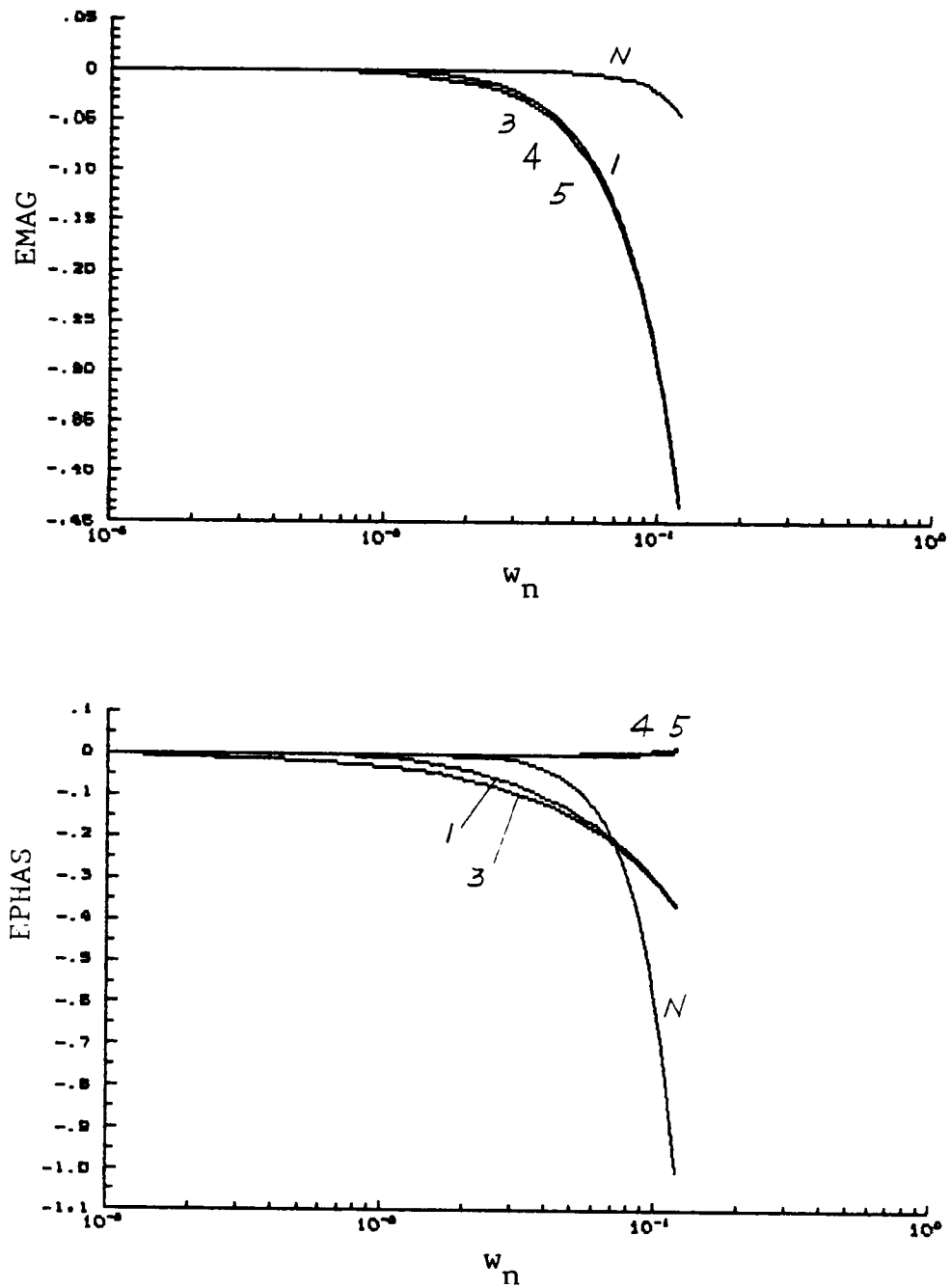


Figure 4.7. EMAG and EPHAS Plots of 1st-Order System
($T = 0.02$, $r = 0.4957$)

approximation methods gave the best results followed by the Taylor method. The implementation of the zero order hold schemes, however, involves the optimization of the r value in the phase compensator.

3. The utilization of the phase compensator with the zero order hold for input approximation merits some discussion. The rule of thumb (2) that the sampling operation is equivalent to adding a dead time equal to one half the sample time is illustrated in these results, especially at smaller sampling times. Recall that the value of r was optimized based on the phase angle response. The best values of r were 0.4790, 0.4895, and 0.4957 for sampling times of 0.1, 0.05, and 0.02 respectively. A value of $r=0.5$ would correspond to compensating for a dead time equal to one half of the sampling time. Another implication is that in simulations allowing larger step sizes to be used, the value of r should be adjusted from the usual 0.5 to obtain more accuracy.
4. A surprising result is that the Taylor method shows little improvement in either of the frequency response plots as T is changed from 0.1

to 0.05 and to 0.02. Also surprising is the fact that the Tustin method magnitude ratio becomes slightly worse as T decreased. Methods 3, 4, and 5 improved in the magnitude ratio plots approaching the Tustin method results while methods 1 and 3 improved considerable in their phase angle response as T decreased. Recall that the plots' abscissa is the normalized frequency so that $\omega_n=0.1$ in Figures 4.5 and 4.7 corresponds to $\omega=10$ Hz and 50 Hz, respectively, thus the error plots extend half a decade further in Figure 4.7 than Figure 4.5 in terms of ω .

In identifying the best method, the evaluation should consider both the magnitude attenuation and phase shift; however, none of the discretization methods in our survey provides the best performance in both magnitude and phase response for the 1st-order lag process.

Plots of the time domain response are now presented to address our major concern for good representation of the time-domain process characteristics. Figure 4.8 through 4.11 present the time domain performance of the discrete models subjected to sinusoid inputs of amplitude one. The input sinusoid wave frequency is controlled by the value of NSC (Number of Samples per

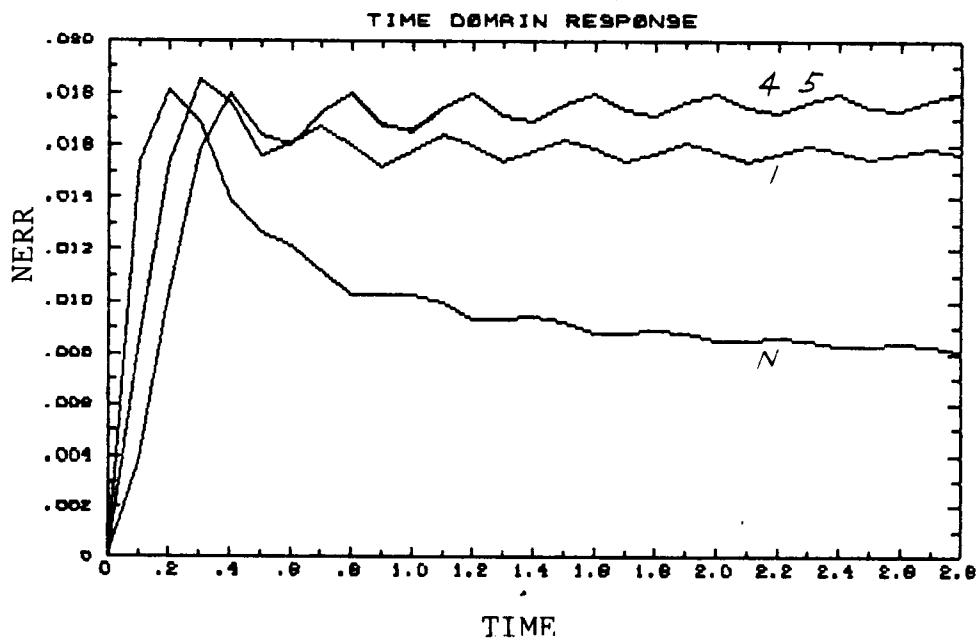
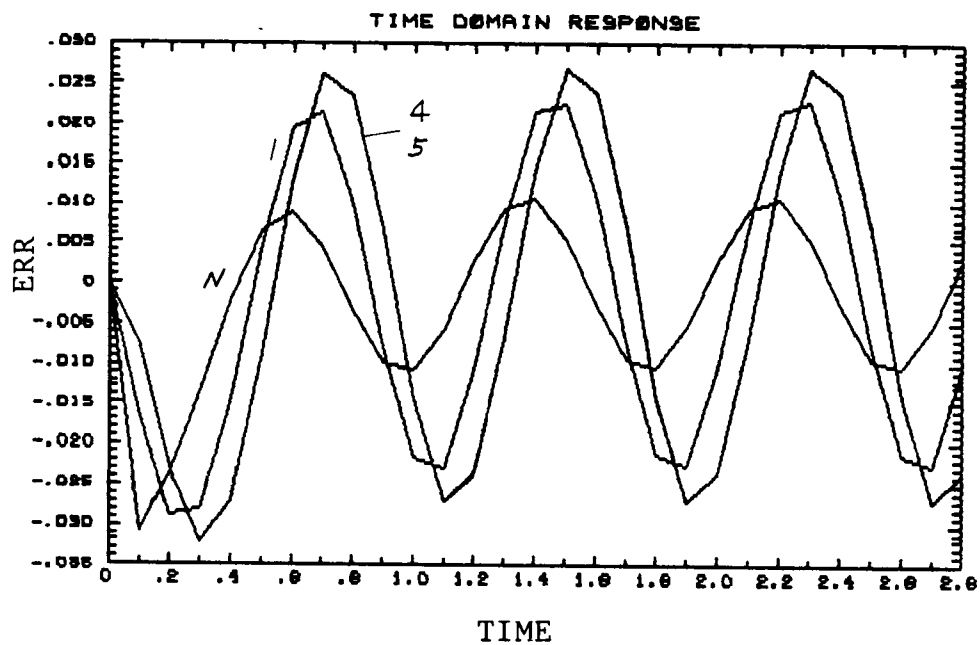


Figure 4.8. ERR and NERR Plots of 1st-Order System
($T = 0.1$, $NSC = 8$, $r = 0.479$)

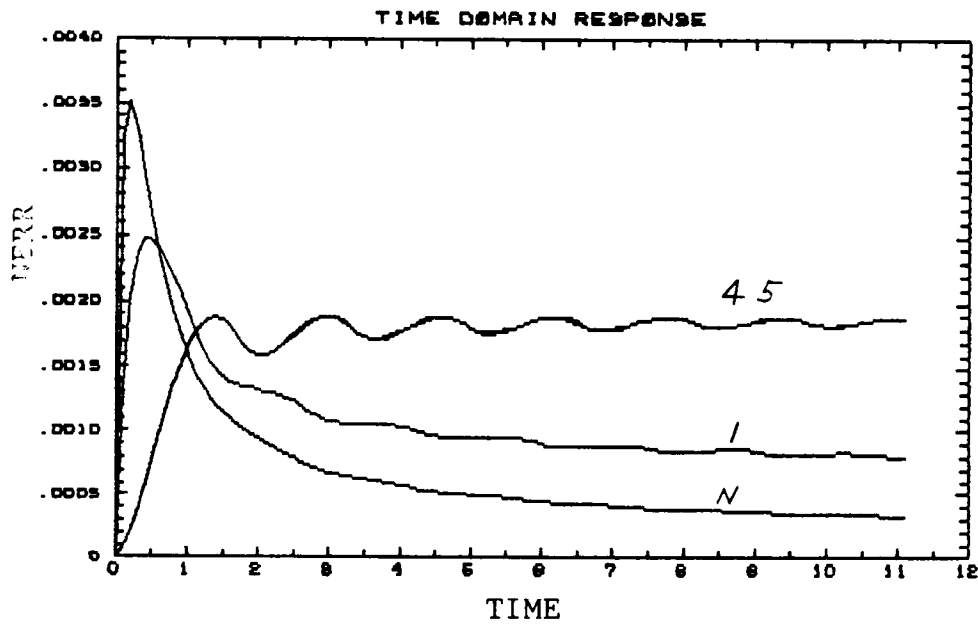
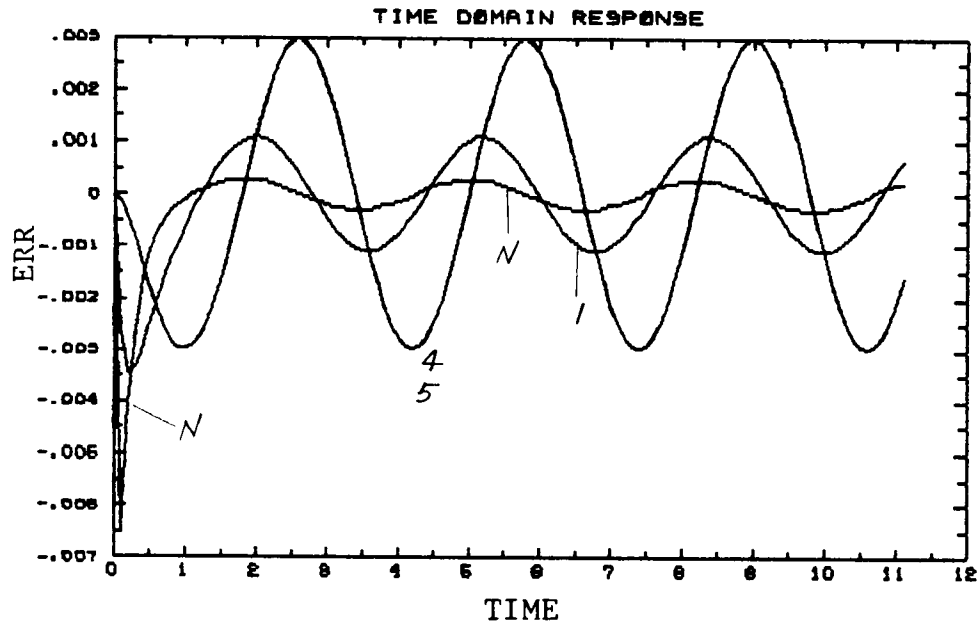


Figure 4.9. ERR and NERR Plots of 1st-Order System
($T = 0.1$, $NSC = 32$, $r = 0.479$)

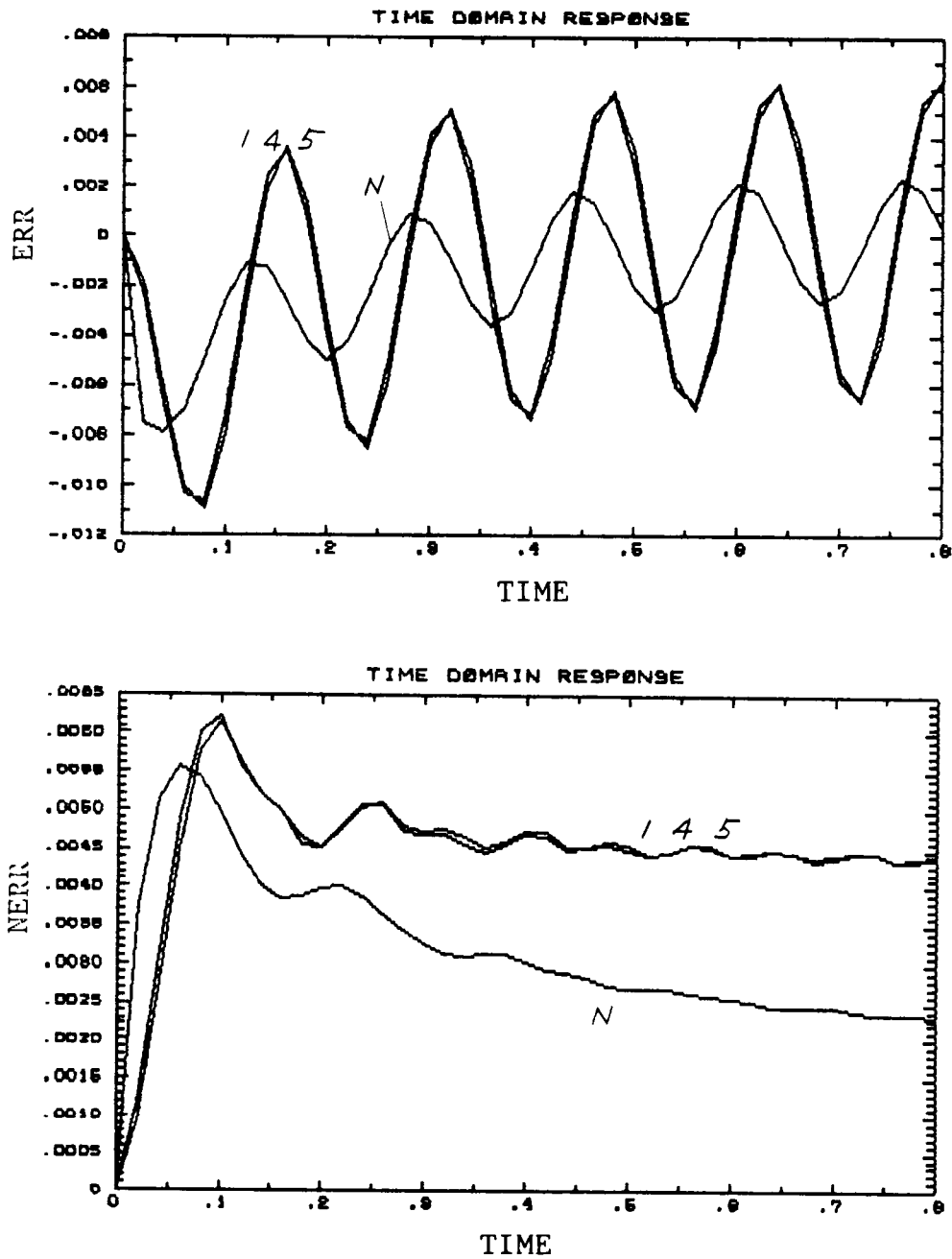


Figure 4.10. ERR and NERR Plots of 1st-Order System
($T = 0.02$, $NSC = 8$, $r = 0.4957$)

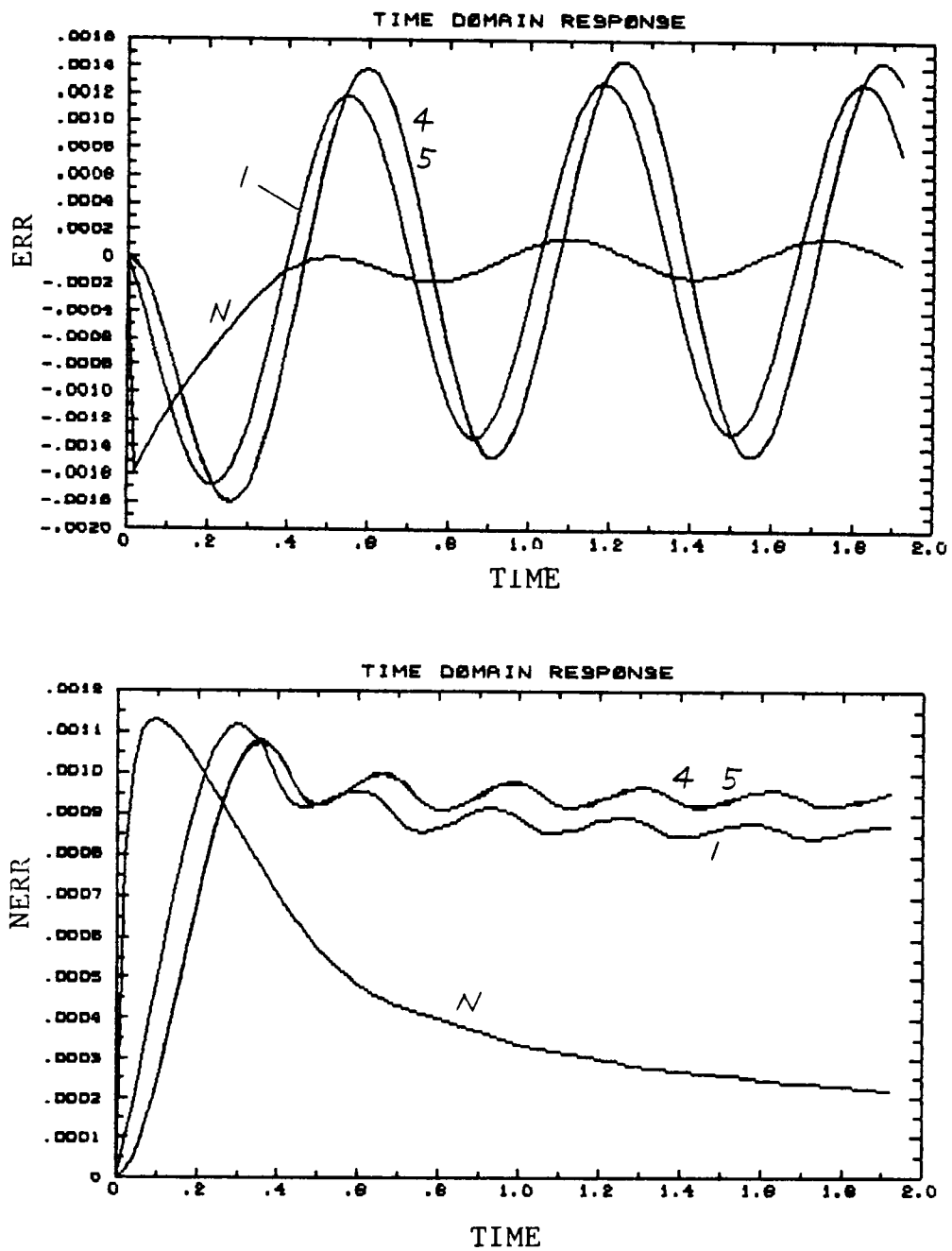


Figure 4.11. ERR and NERR Plots of 1st-Order System
($T = 0.02$, $NSC = 32$, $r = 0.4957$)

Input Cycle), which is defined of the as the reciprocal normalized frequency.

$$NSC = \frac{f_s}{f_i} = \frac{1}{w_n} \quad (4.31)$$

For example, if the sample period, T , is 0.2 and the chosen NSC is 8, then,

$$\text{sample frequency, } f_s = \quad f_s = \frac{1}{T} = 5$$

$$\text{input frequency, } f_i = \quad f_i = \frac{f_s}{NSC} = \frac{5}{8}$$

The time domain responses for the sinusoid wave tests reveal a clear picture of each discrete transfer function's performance. Methods 2 and 3 was left off the the figures as they have significantly larger errors.

Conclusions drawn from the time domain response and the associated indexes for the 1st-order lag process are:

1. The local errors, ERR , are reduced as the sample period is decreased (as the NSC is increased); this behavior is intuitively expected.
2. The major conclusion drawn is that the Taylor method clearly provides the best digital

algorithm for the first order lag process

3. The Taylor method does have errors larger than the other methods at the beginning of the simulation. This initial error is usually comparable in magnitude to the errors the other methods experience at larger times. The time frame during which the ERR index of the Taylor method remains larger ranges from one fourth to one process time constant ($T_r = 0.2$).

The phenomena of the Taylor method experiencing larger initial error than the other methods might arise from the fact that its numerator has three terms versus two for the others, thus three sample periods are needed in a simulation before the full algorithm is in place. This behavior might be enhanced since a first order process has an immediate response to an input, e.g. in the step response of such a process the derivative of the response has its maximum value at time=0. Some of the other methods also show this start up characteristic.

The magnitude of the ERR levels compared to the magnitude of the input signal (sinunoid wave with amplitude of one) range from $\pm 3\%$ in Figure 4.8 to $\pm 0.5\%$ in Figure 4.9.

When including the frequency domain criteria, as

well as the time domain criteria the best discretization method within this survey is still concluded to be the Taylor method.

4.6 Comparison II

Comparison II follows the same procedure as used in comparison I, to evaluate the discretization methods when used to develop a simulation for an underdamped second-order lag system. The underdamped second order lag transfer function chosen for comparison II is

$$G(s) = \frac{10}{s^2 + s + 10} \quad (4.32)$$

The natural frequency, ω_0 , and damping coefficient, ξ , are

$$\omega_0 = \sqrt{10} \quad , \text{ is the natural frequency.}$$

$$\xi = -.158 \quad , \text{ is the damping ratio}$$

First reasonable sample periods are chosen based on the system dynamic behavior. The sample frequencies are taken to be at least ten times greater than the conditional frequency and also of high enough frequency that more than 2 samples are taken per rise time. The conditional frequency, ω_c , and the rise time, T_r , of a

2nd-order system are defined as follows (21)

$$T_r = \frac{1 + 1.1\xi + 1.4\xi^2}{w_0} = 0.4 \quad (4.33)$$

$$w_c = \frac{w_0 \sqrt{1 - \xi^2}}{2\pi} = 0.5 \text{ (Hz)} \quad (4.34)$$

Therefore, the selected sample periods should be less than 0.2.

One problem with the Tustin method as discussed in Chapter III is the frequency scale distortion. The effect is called "frequency warping" due to the nonlinear frequency transformation. The underdamped second order lag process provides a good illustration of the warping phenomena. The resonant frequency and the frequency response at this frequency are important characteristics of this process. To avoid frequency distortion or warping of the discrete model at the resonant frequency pre-warping can be incorporated according to Equation 2.33. This procedure allows the frequency response of the continuous system and the approximating discretized system to be the same at one frequency; here at the resonant frequency. The resonant frequency, w_p , of a 2nd-order system is

$$w_p = w_n \sqrt{1 - 2\xi^2} = 3.08 \quad (4.35)$$

and following Equation 2.33 the discrete model with prewarping at w_p , $H(z)$, is

$$H(z) = G(s) \left| s = \frac{\frac{w_p}{w_p T} \frac{z-1}{z+1}}{\tan \frac{P}{2}} \right. \quad (4.36)$$

Figure 4.12 illustrates the frequency response of the Tustin method with and without prewarping. The discrete model with prewarping preserves the frequency response at the desired frequency; however, it does not preserve the magnitude and phase response at other frequencies. It should be noted however that the Bode plot with prewarping is improved over all frequencies in this example. In later comparisons, the Tustin method without prewarping is again used.

The frequency response of the input approximation discrete models with zero-order hold and various phase compensators are shown in Figures 4.13 and 4.14. The best choice of the r value for the phase compensator is 0.5 from both the magnitude and phase response standpoint. This is an unusual situation.

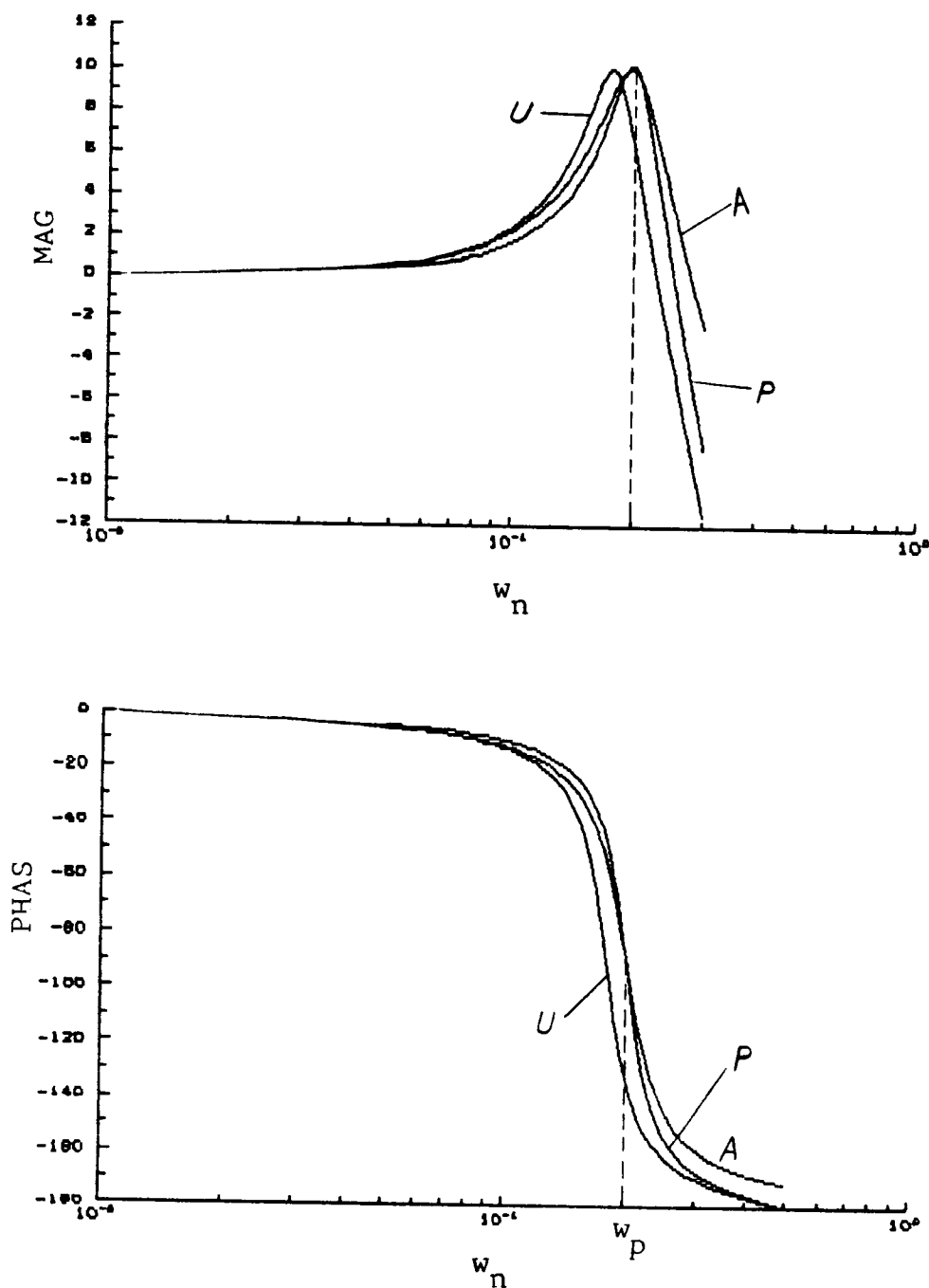


Figure 4.12. Frequency Response of Comparison II System with Tustin Methods.
 (P) with Frequency Prewarping at w_p
 (U) Without Frequency Prewarping

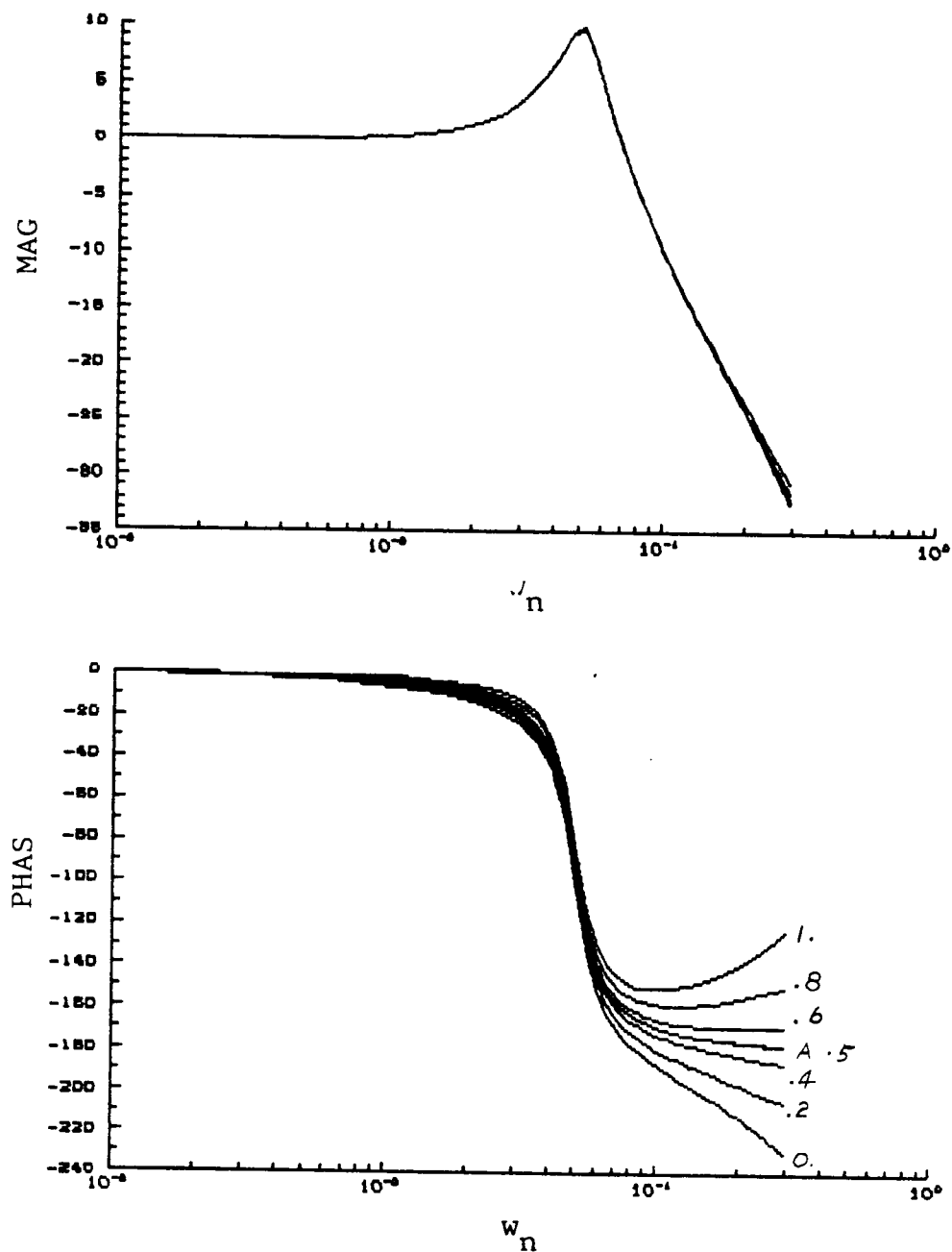


Figure 4.13. Frequency Response of the Input Approximation Method with Zero-Order Hold and Various Phase Compensator for Comparison II ($T=0.1$)

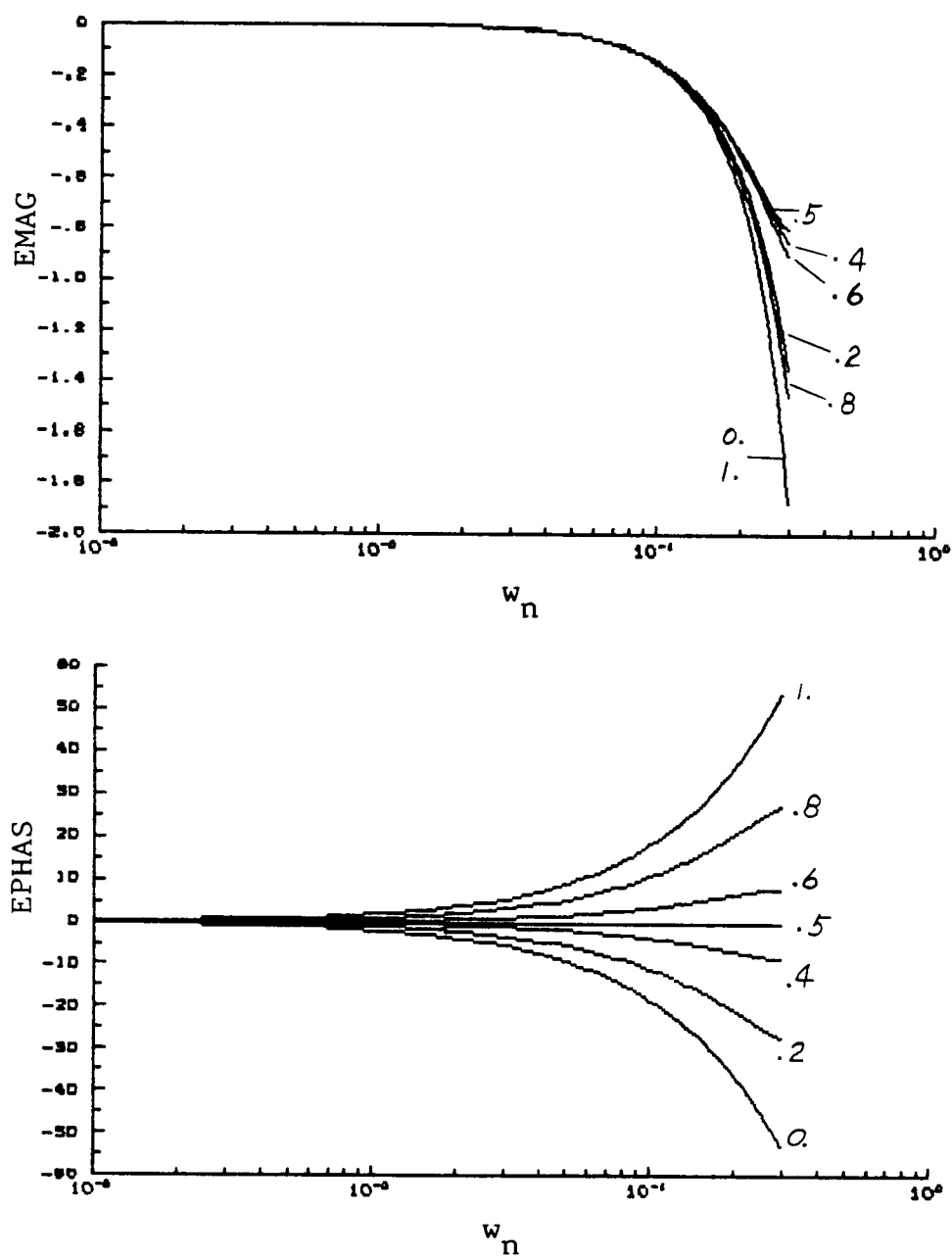


Figure 4.14. EMAG and EPHAS Plots of the Input Approximation Method with Zero-Order Hold and Various Phase Compensation for Comparison II ($T=0.1$)

Figures 4.15 and 4.16 contain the frequency domain comparison of the discretization techniques, discrete models' coefficients shown in Table 4.4. Note method 2 is again left off of Figures 4.16 to 4.18 as it has significantly larger deviations compared to all other methods. Based on these figures the following conclusions are made:

1. The Taylor method and the two input approximation methods are found to be better than the others in maintaining both the magnitude ratio and phase angle.
2. Figure 4.14 shows the optimum value of r for the phase compensator is 0.5 for the magnitude ratio, as well as, the phase angle.
3. The Tustin method has considerable accuracy problems around the resonant frequency. Thus prewarping would make EMAG and EPHAS zero at $\omega=3.08$ radians or 0.49 Hz which corresponds to $\omega_n=0.049$ in Figure 4.16, as well as, make the Bode plot errors smaller in general. The question of what effect this enhancement would have on the time domain characteristics has not been explored.

The time domain performance for each discrete model (less method 2 as it had significantly more error) is

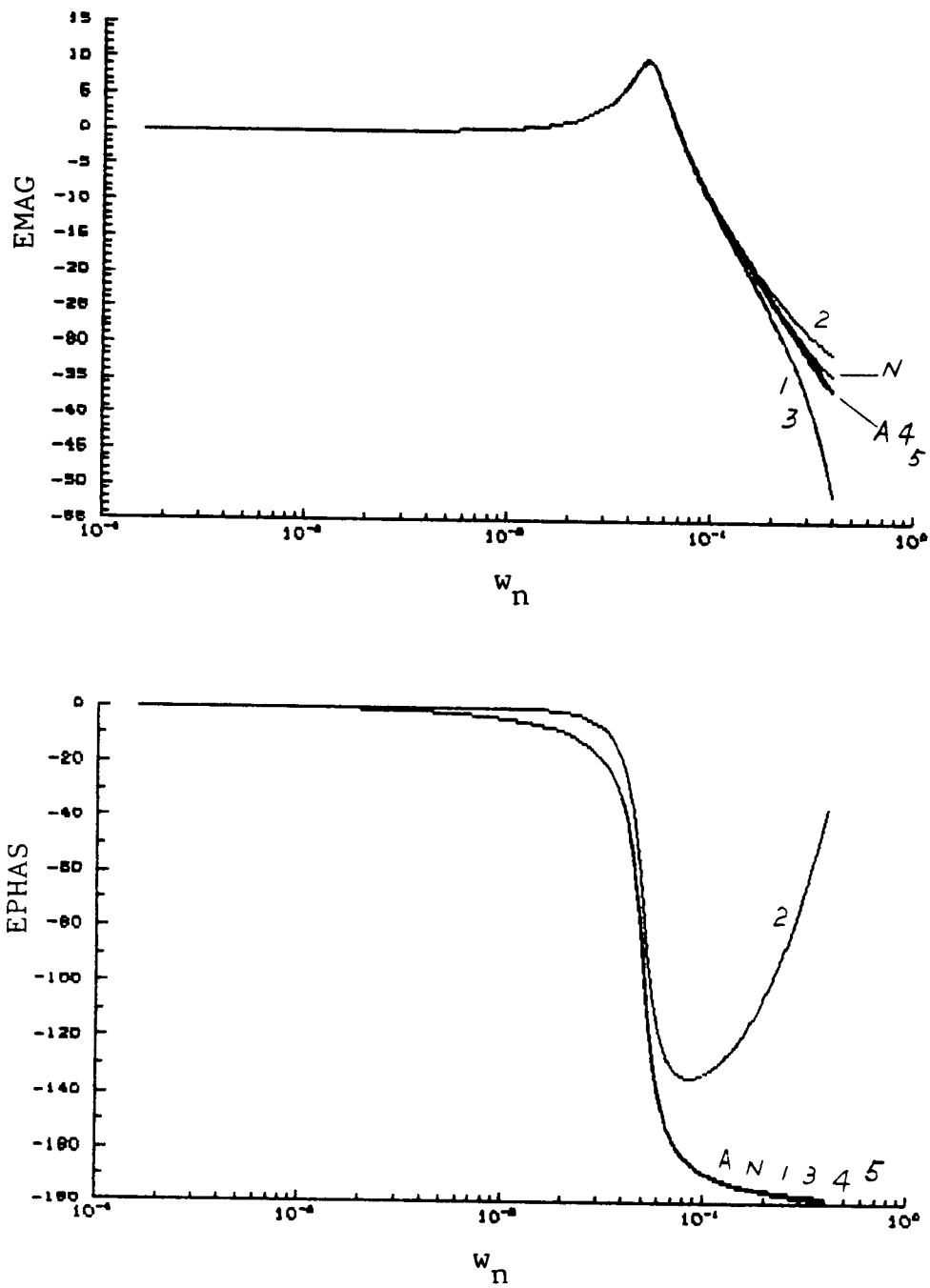


Figure 4.15. MAG and PHAS Plots of Comparison II
System ($T = 0.1$, $r = 0.5$)

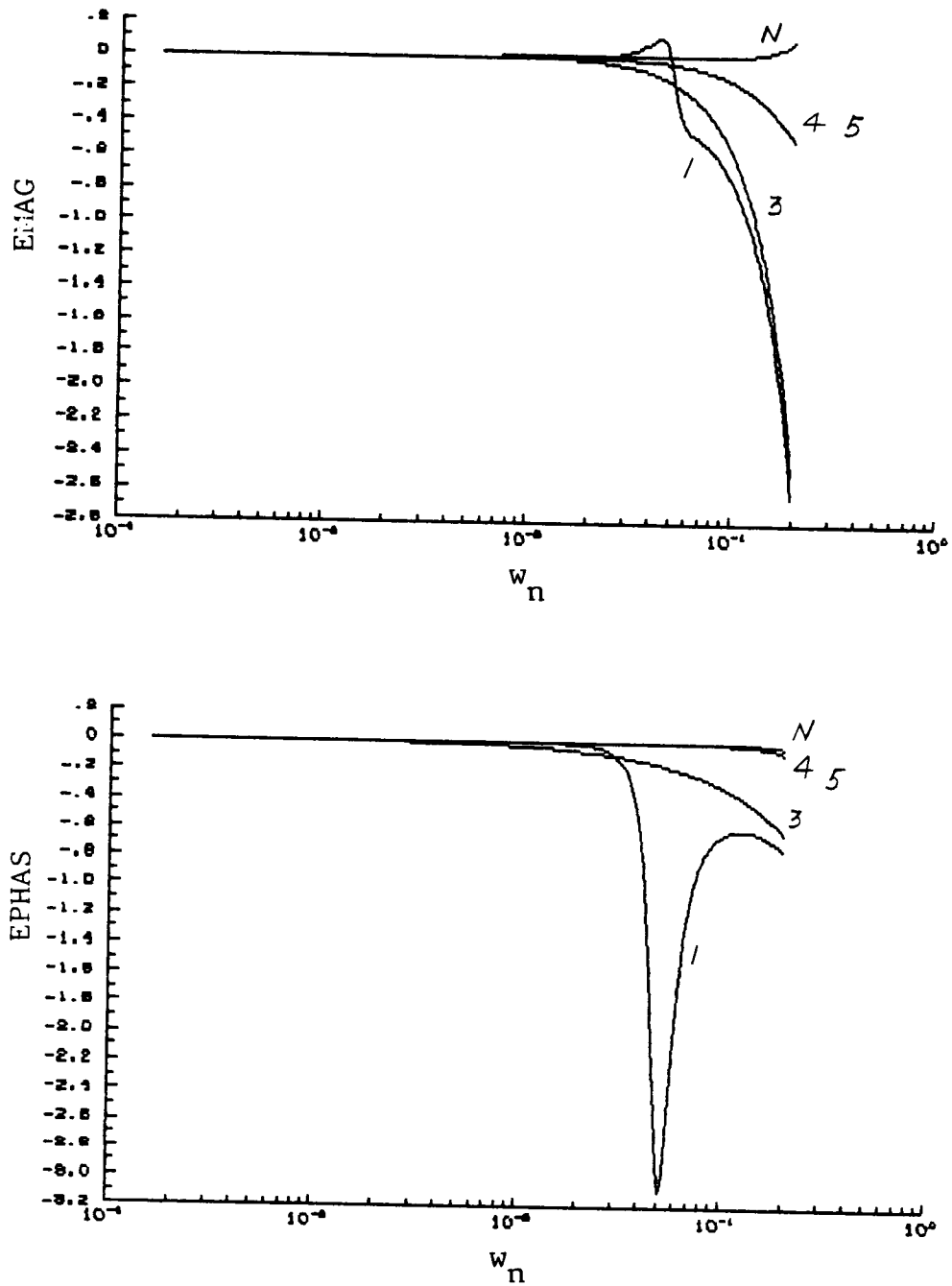


Figure 4.16. EMAG and EPHAS Plots of Comparison II System ($T = 0.1$, $r = 0.5$)

Table 4.4. Coefficients of Discrete Models in Comparison II

Method	Coefficient
Tustin without prewarping	$a_0 = sab \cdot t_2 / dc_1$ $a_1 = 2 \cdot sab \cdot t_2 / dc_1$ $a_2 = sab \cdot t_2 / dc_1$ $b_1 = (2 \cdot sab \cdot t_2 - 8) / dc_1$ $b_2 = (sab \cdot t_2 - 4 \cdot a \cdot T + 4) / dc_1$
Tustin with prewarping	$a_0 = sab / dc_2$ $a_1 = 2 \cdot sab / dc_2$ $a_2 = sab / dc_2$ $b_1 = (2 \cdot sab - 2 \cdot pw^{**2}) / dc_2$ $b_2 = (sab + pw^{**2} - 2 \cdot pw^{**}a) / dc_2$
Matched Pole-Zero ($z=0$)	$a_0 = \text{denom}$ $a_1 = 0$ $a_2 = 0$ $b_1 = \text{etcos}$ $b_2 = e^{2t}$
Matched Pole-Zero ($z=-1$)	$a_0 = \text{denom}/4$ $a_1 = \text{denom}/2$ $a_2 = \text{denom}/4$ $b_1 = \text{etcos}$ $b_2 = e^{2t}$
Input Approximation with Hzo and phase compensator	$a_0 = 1-p$ $a_1 = \text{etcos} \cdot p + q$ $a_2 = e^{2t} - q$ $b_1 = \text{etcos}$ $b_2 = e^{2t}$
Input Approximation with Triangular Hold	$a_0 = (c_2 - 4 \cdot a + c_1 - 2 \cdot a \cdot \text{etcos}) / (sab \cdot T)$ $a_1 = (2 \cdot c_2 - 2 \cdot a + c_1 \cdot \text{etcos} - 2 \cdot a \cdot e^{2t}) / (sab \cdot T)$ $a_2 = (c_2 + c_1 \cdot e^{2t}) / (sab \cdot T)$ $b_1 = \text{etcos}$ $b_2 = e^{2t}$

Table 4.4. (Continued)

Defined constant:

1. common constants:

```

w0    = sqrt(a**2+b**2)
eta   = a/(2*w0)
wp    = w0*sqrt(1-2*eta**2)
sab   = a**2+b**2
et     = exp(-a*T)
e2t   = exp(-2*a*T)
etcos = -2*et*cos(b*T)
t2    = t**2
w2    = w**2
w4    = w**4

```

2. Constants for each methods:

```

denom = 1+etcos+e2t
pw     = wp/tan(wp*T/2)
dc1    = sab*t2+4*a*T+4
dc2    = pw**2+2*pw*a+sab
phai   = atan(-a/b)
p      = exp(-a*r*T)*cos(r*b*T+phai)/cos(phai)
q      = exp(-a*(1+r)*T)*cos((1-r)*b*T-phai)/cos(phai)
ang    = atan((b**2-a**2)/(2*a*b))
c1     = sab*T+2*a
c2     = -2*a*et*cos(b*T+ang)/cos(ang)

```

For 2nd-order System:

$$G(s) = \frac{a^2 + b^2}{(s+a)^2 + b^2}$$

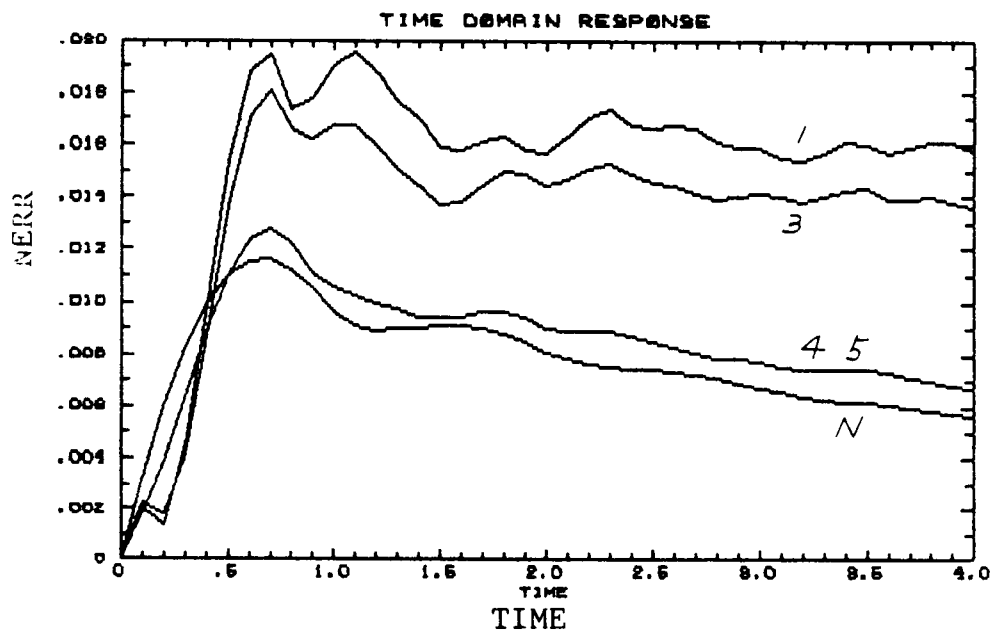
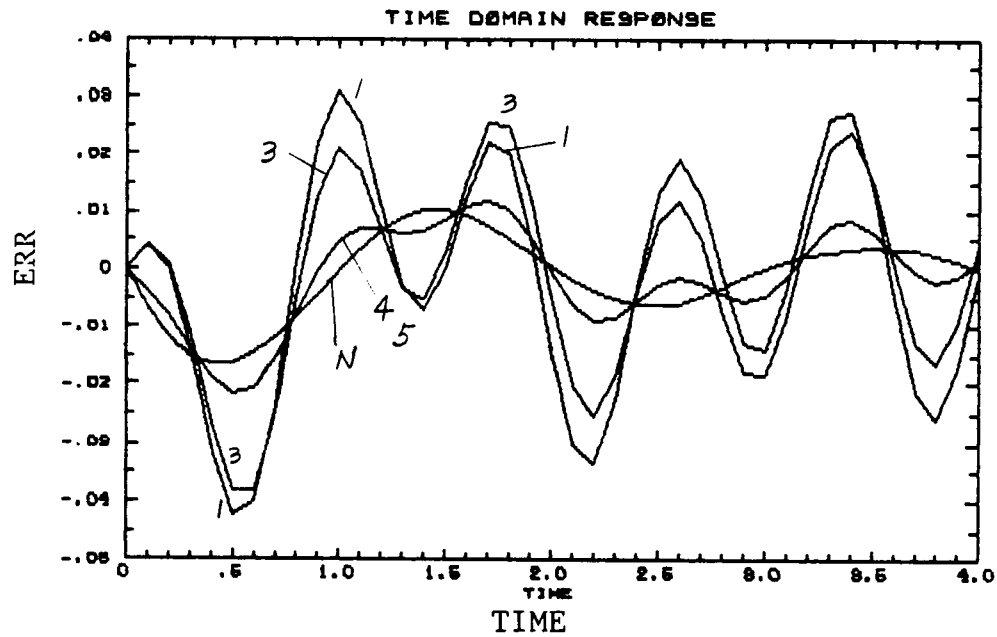


Figure 4.17. ERR and NERR Plots of Comparison II
System ($T = 0.1$, $NSC = 8$, $r = 0.5$)

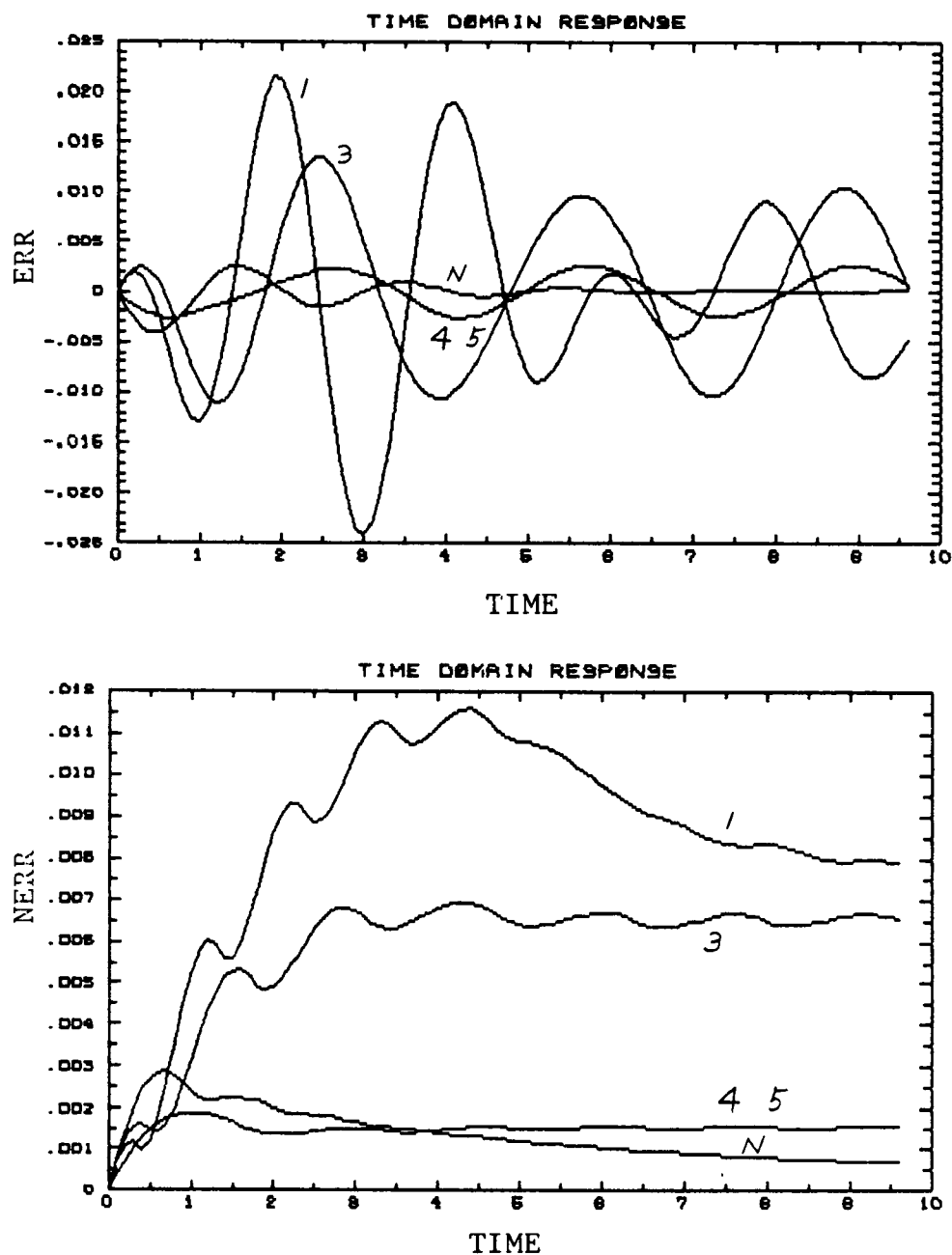


Figure 4.18. ERR and NERR Plots of Comparison II
System ($T = 0.1$, $NSC = 32$, $r = 0.5$)

presented in Figures 4.17 and 4.18 from which it is observed:

1. As expected, the local errors, ERR, again show a decrease as NSC is increased from 8 to 32.
2. The Taylor method and the two input approximation methods provide nearly the same performance capabilities and are clearly superior to the other techniques. These methods also provided the best results in terms of the frequency criteria.
3. In contrast to Comparison I, here the Taylor method does not exhibit a larger initial error than the error level it has at later times. Presently, the only speculation for this difference would be a possible connection with the different derivatives of the response at time zero that 1st and 2nd order process exhibit.

For the 2nd-order underdamped lag, in both the frequency and time domains the Taylor and the two input approximation methods provide the best simulation capabilities. Many more plots were made for both domains as part of Comparison II but have not been included as they confirm the conclusion made here.

4.7 Comparison III

The final study, called comparison III, uses a 2nd-order nonminimum phase system. A dynamic characteristic of such a process is the initial response is in the opposite direction compared to the later and final response. This characteristic is shown in Figure 4.19a. Such behavior is denoted by various names including inverse response, nonminimum phase response, wrongway behavior and sagging process.

The 2nd-order nonminimum phase system, $G(s)$, selected is

$$G(s) = \frac{1 - s}{(s + 2)(s + 3)}$$

or

$$G(s) = \frac{3}{s + 2} - \frac{4}{s + 3}$$

Note that the transfer function can be written as the sum of two transfer functions. This is equivalent to the input having an effect on the output through parallel paths. Often the nonminimum phase characteristic in physical processes can be explained due to this type of phenomena.

The eigenvalues of the selected transfer function are -2 and -3, thus the effective time constants are 0.5 and 0.33, respectively. Based on the smallest time constant sample periods chosen for the study are 0.167

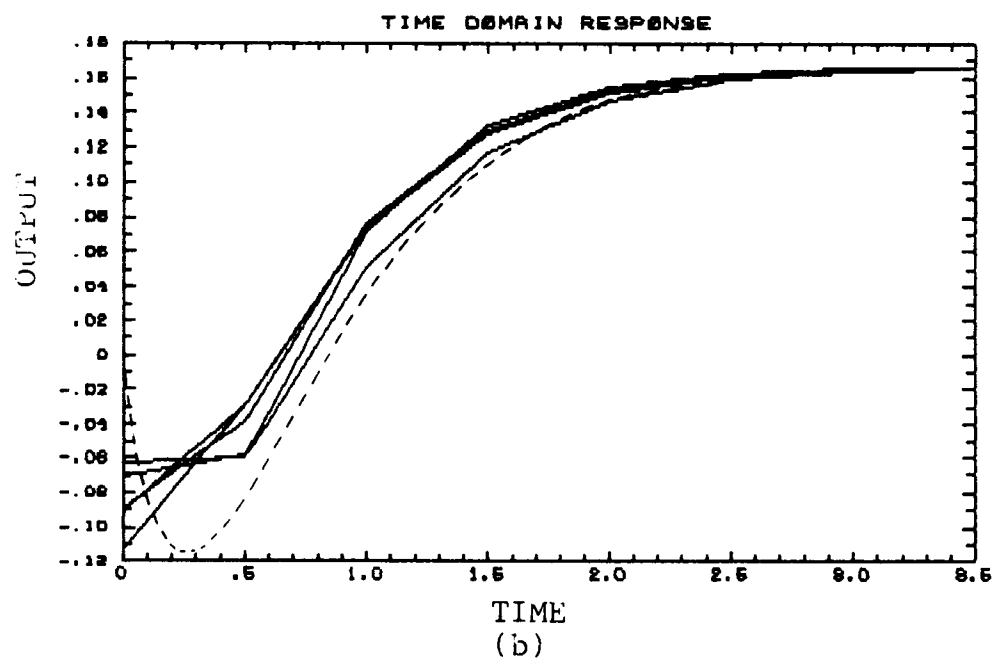
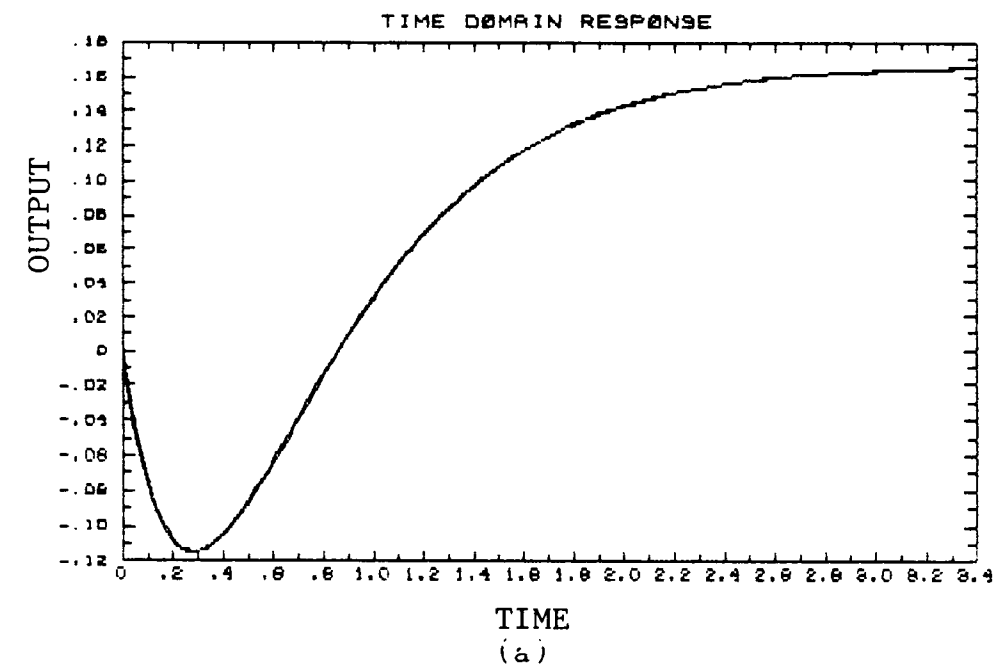


Figure 4.19. Step Response of Comparison III System
 (a) $T=0.03$ and $r=0.5$ (b) $T=0.5$ and $r=0.5$, the Dash Line Indicates the Response of the Continuous System.

and 0.03 associated with 2 and 10 samples per time constant. Unit step input responses of the continuous system and the discrete models with $T = 0.03$ and 0.5 are shown in Figure 4.19. A sample time of 0.03 is a fairly small step size while that of 0.5 is a large step size, in fact, it is equal to the larger effective time constant. This figure is included to show that for a small enough step size all of the discrete algorithms being evaluated work well, whereas, if the step size is obviously too large none of the algorithms can provide a description of the dynamics. All these simulation techniques do yield correct steady state result as discussed earlier.

Before proceeding with the established comparison study format, this example is used to introduce the question and show some calculations for applications of the Taylor method using a different number of numerator a -coefficient. Recall that this method maps poles of the continuous transfer function directly to the z -domain so that number of poles in the equivalent discrete transfer functions is fixed by the continuous process. However the number of zeroes can be selected to best fit the Taylor series expansion in the time-domain. For the 2nd-order nonminimum phase system the Taylor method was developed using 2, 3, 4, 5, and 6

terms in the numerator. Figures 4.20, 4.21, and 4.22 show the corresponding frequency response with the numerical value on each curve indicating the number of terms in the numerator of the Taylor models. The discrete transfer function results for the case of 2 terms in the numerator are not shown in Figures 4.21 to 4.25 because it contained significantly greater error than the other Taylor models. Observations from the Bode plot errors for the Taylor method with different degrees of the numerator polynomial are:

1. Initial intuitive feeling might suggest that the more terms used in the numerator the better the approximation model. This trend might be expected to break down if the numerator contained many more terms than the denominator. Figure 4.20 to 4.22 are mostly consistent with this intuition, with both EPHAS and EMAG getting smaller as additional terms are added in the numerator. Note that the error in magnitude ratio with 4 terms is less than with 5 terms.
2. As more terms are added it might be expected that the error would approach or "oscillate" about the highest term approximation error. That is, as a new terms is added it would change the error in the direction of smaller error and

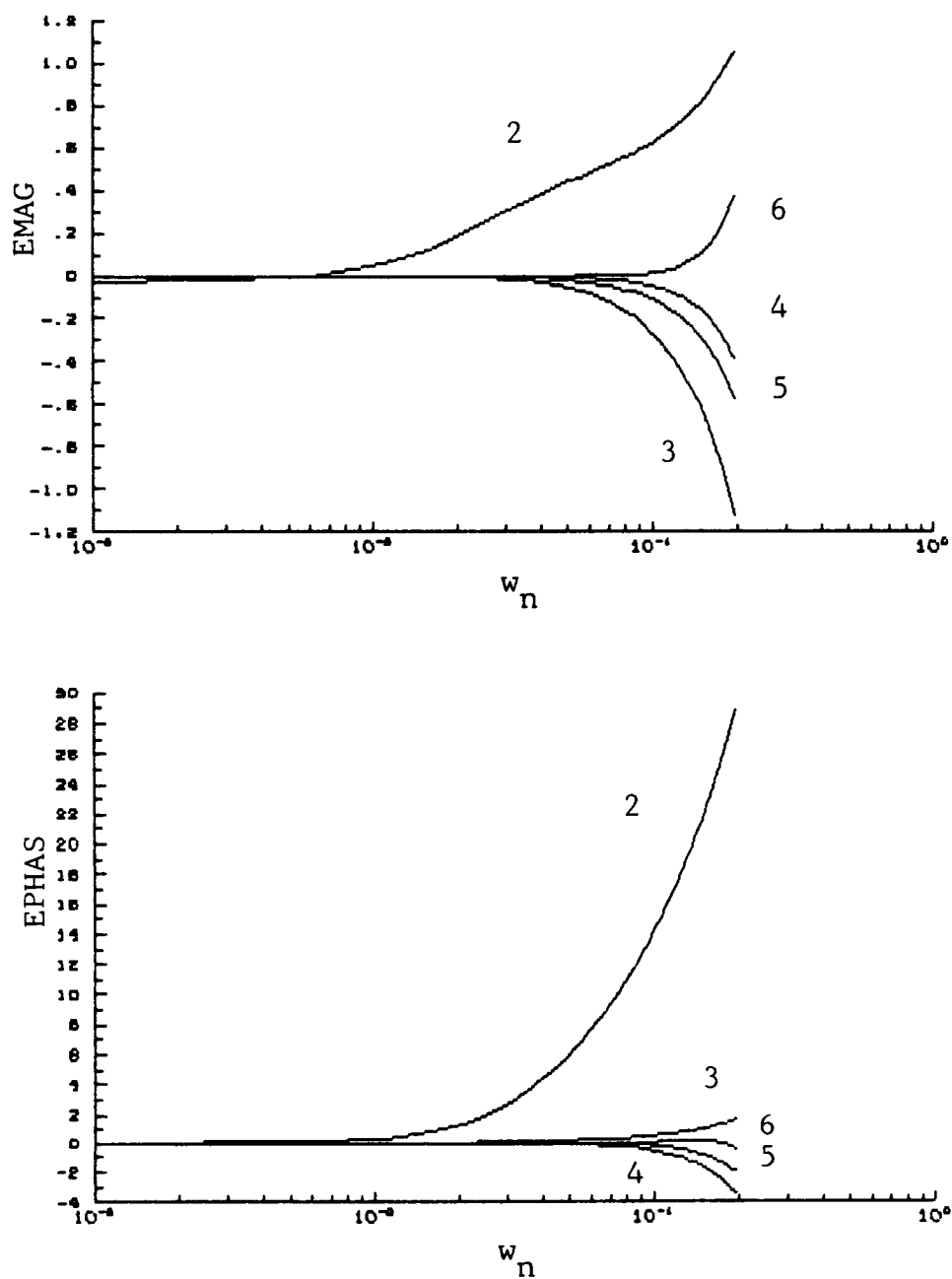


Figure 4.20. MAG and PHAS Plots of Comparison III System with Taylor Method ($T = 0.167$)

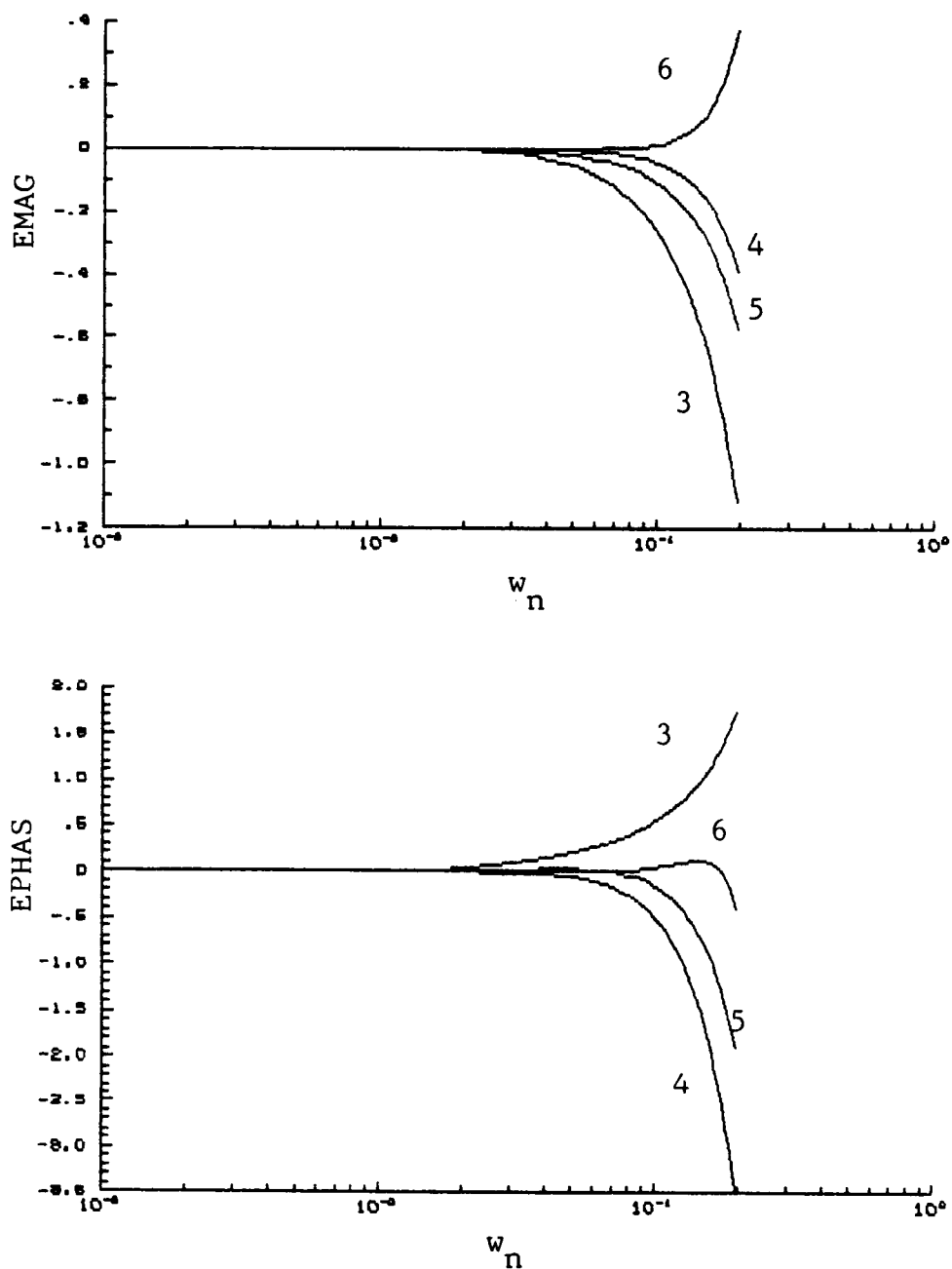


Figure 4.21. EMAG and EPHAS Plots of Comparison III System with Taylor Method ($T = 0.167$)

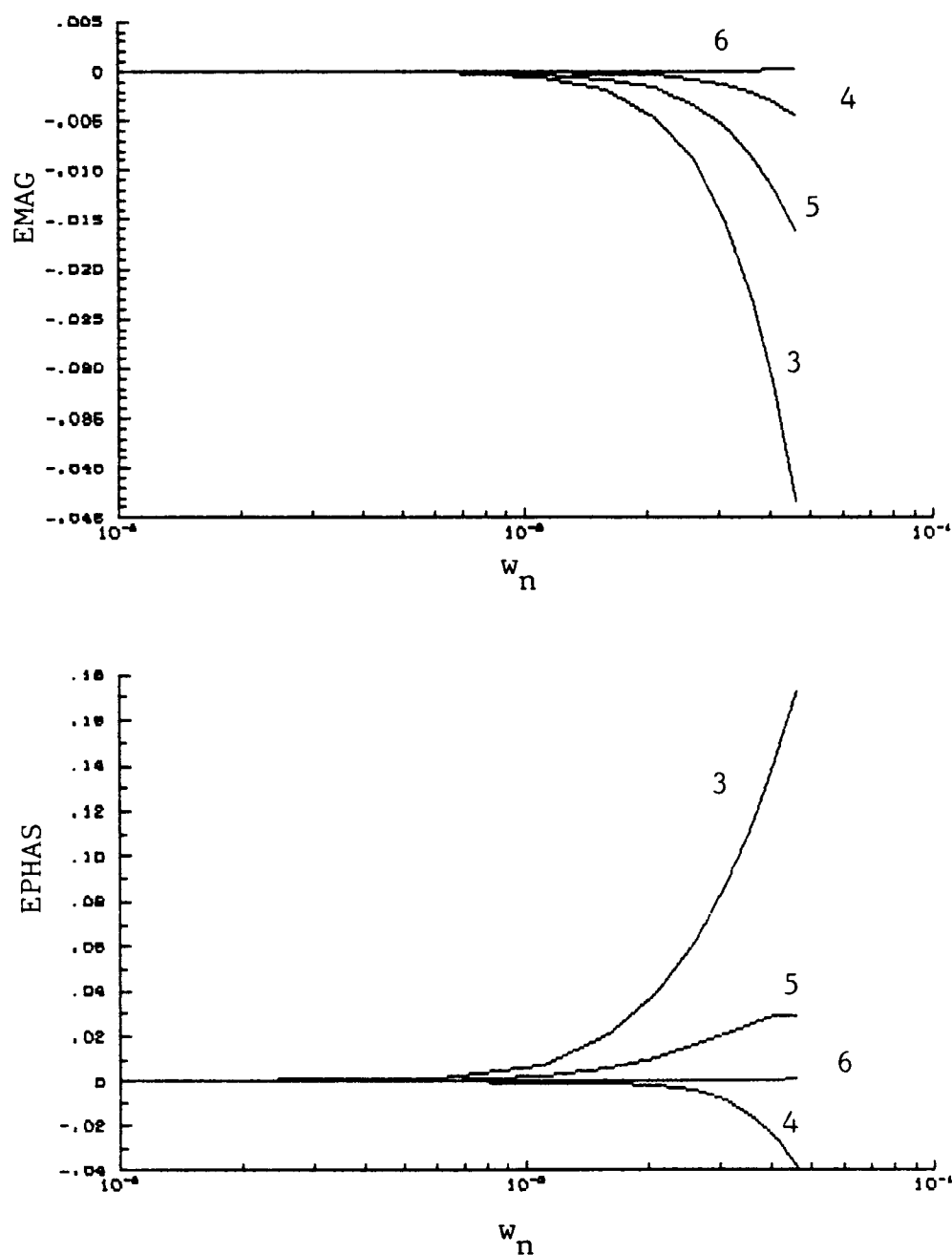


Figure 4.22. EMAG and EPHAS Plots of Comparison III System with Taylor Method ($T = 0.167$)

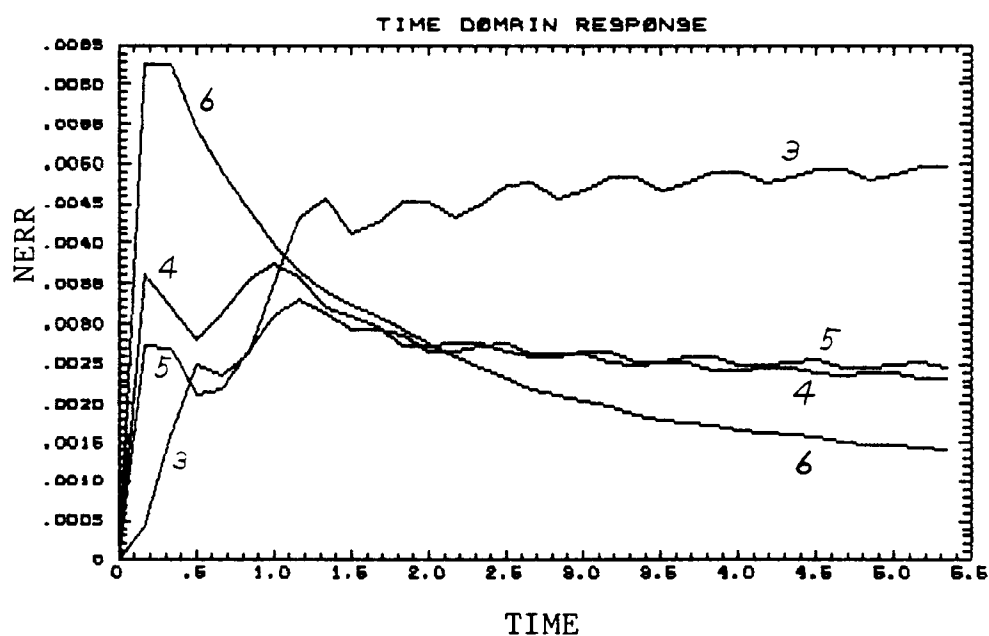
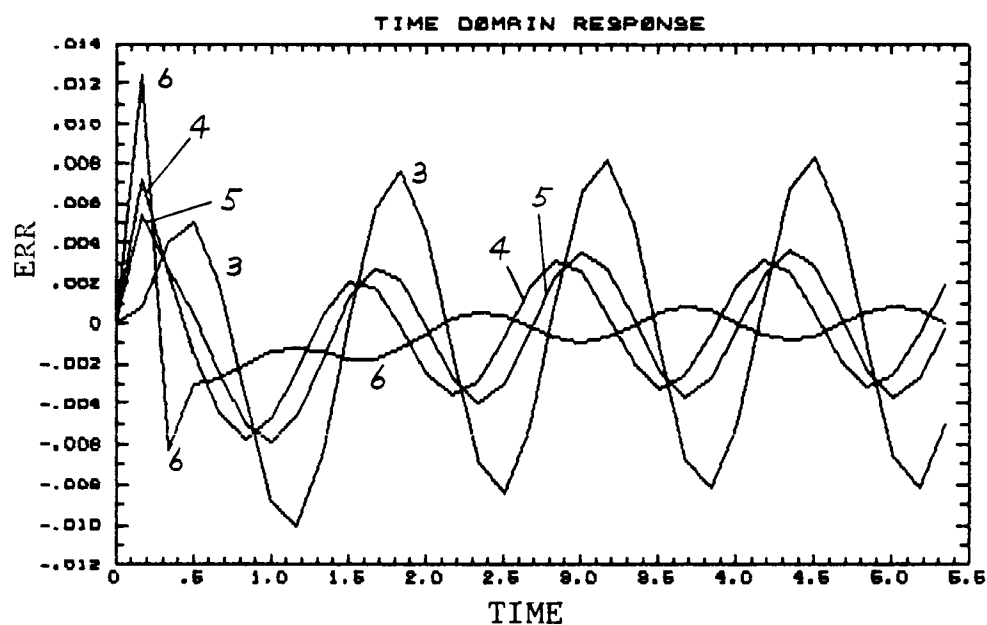


Figure 4.23. ERR and NERR Plots of Comparison III System with Taylor Method ($T = 0.167$, $NSC = 8$)

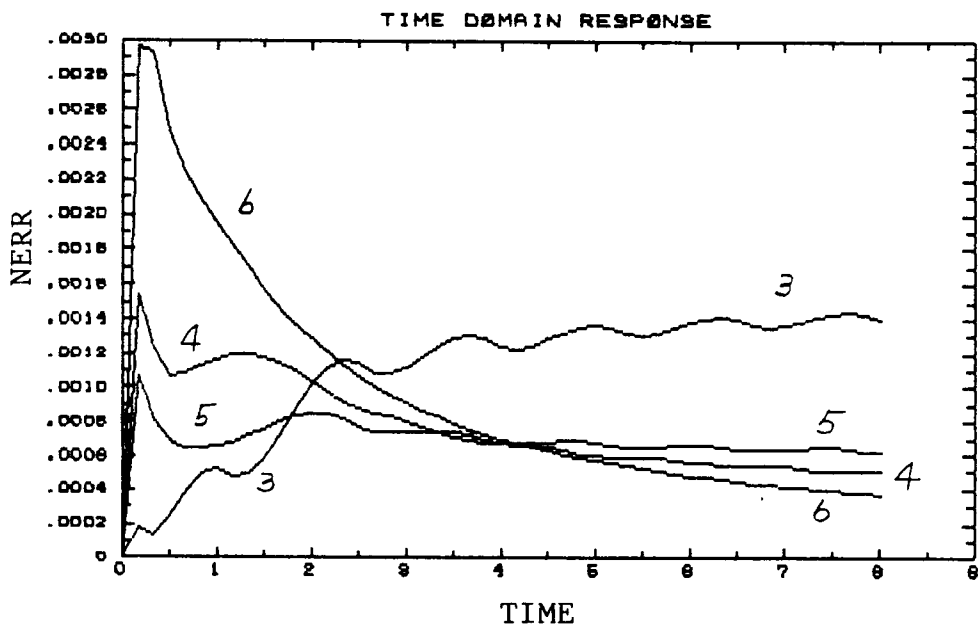
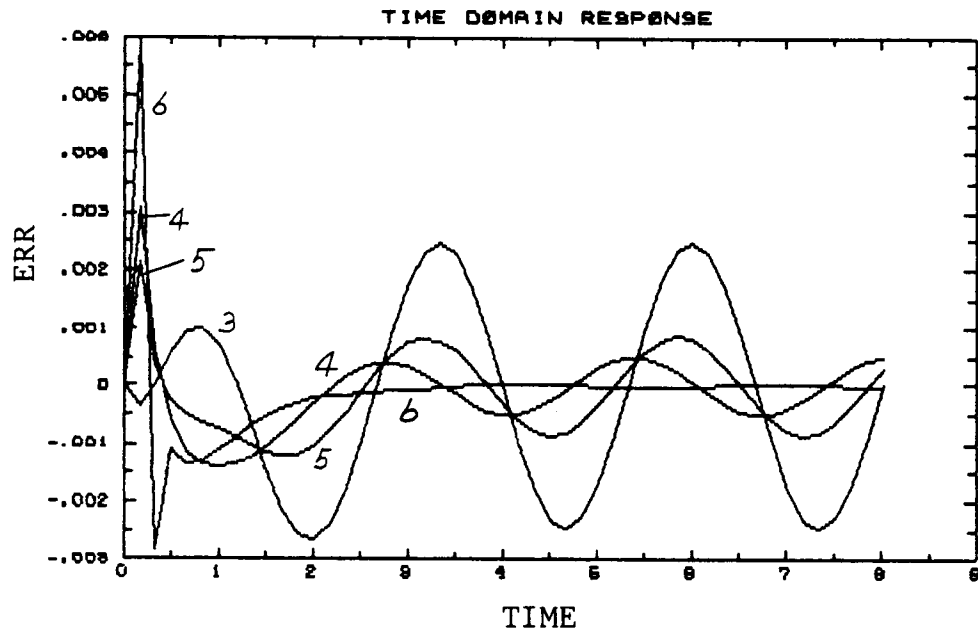


Figure 4.24. ERR and NERR Plots of Comparison III System with Taylor Method ($T = 0.167$, $NSC = 16$)

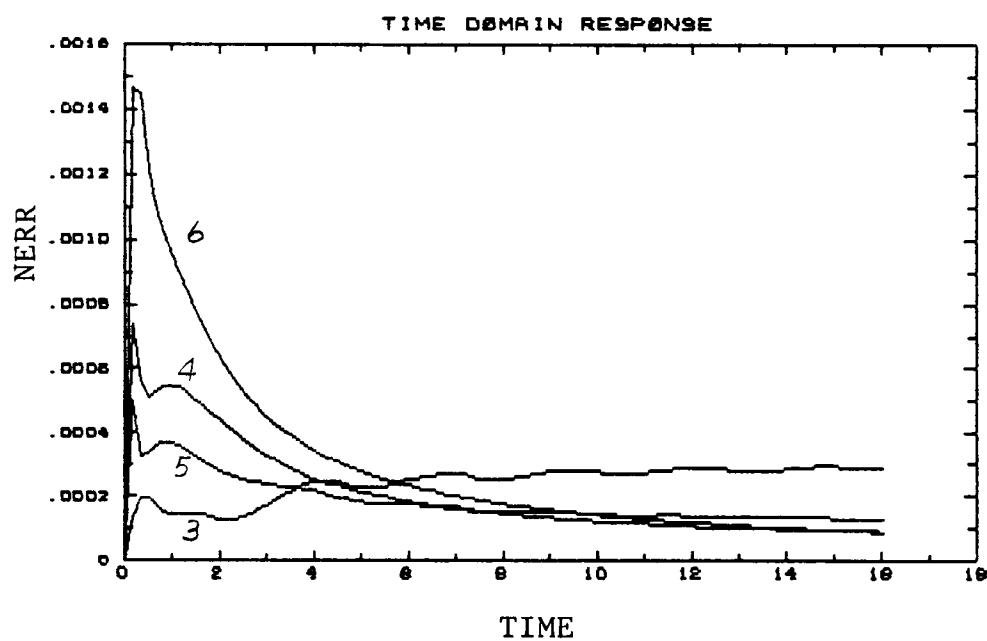
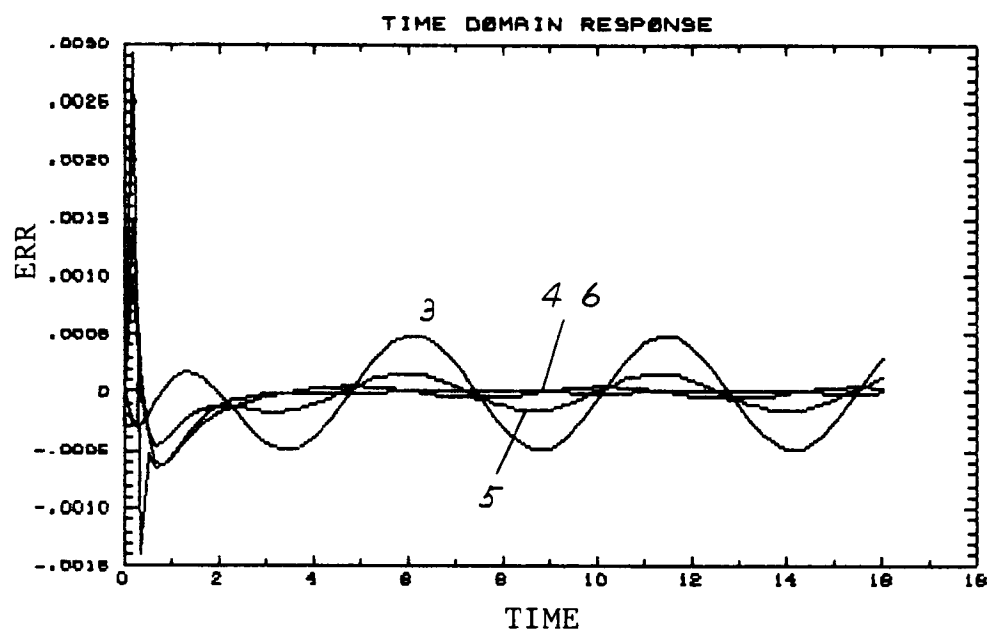


Figure 4.25. ERR and NERR Plots of Comparison III System with Taylor Method ($T = 0.167$, $NSC = 32$)

perhaps have the error change sign. This happens explicitly in the EPHAS plot in Figure 4.22 and is observed in most of the other plots.

The study of the various numerator order models in the time-domain is presented in Figures 4.23 through 4.25 for a sample time of 0.167 and samples per input cycle of 8, 16, and 32. The local error in the simulation, ERR, in relation to 1, the amplitude of the input, ranges from 0.3 to 1.2 .

Observations from the error index plots for this example are:

1. The level of the error indexes decrease as NSC is increased. For the set of three figures the error decrease is inversely proportional to the NSC increase, i.e. when NSC is doubled, the error indexes are reduced by about half.
2. The phenomena of the approximation experiencing its largest error indexes at the start of the simulation again appears. Here the simulation is equivalent to that of two 1st-order processes in parallel, thus some of the behavior shown by the Taylor method in Comparison I might be expected.
3. The initial error phenomena is accentuated as the order of the numerator is increased. This is

easily seen in Figure 4.24. The model with six coefficients in the numerator has the largest ERR and NERR after one simulation step while the one with 3 coefficients has the smallest ERR and NERR. The 4 coefficient model has slightly larger errors than the 5 coefficient model.

The magnitude of the initial error appears to be connected with the fact that the n -th numerator term does not enter into the recursive formula until the n -th sample time. In Figure 4.24 the plot for 6 numerator coefficients shows a discontinuity through the first 4 sample times. Thus in some sense it takes 4 sample times for the algorithm to get initialized.

4. As suggested by the magnitude ratio error plots (Figures 4.21 and 4.22), the higher coefficient models have the best error characteristics as the simulation proceeds with the level of NERR at the ending time on each of the time-domain plots having the same order from smallest to largest error as the error exhibited at the highest frequencies shown in Figures 4.21 and 4.22.
5. The flexibility that the Taylor method provides through the numerator coefficient appears

to provide a trade-off between the amount of initial error versus the amount of error that occurs later in the simulation. For example, in Figure 4.24 the results from the 6 term model have the largest initial ERR and NERR and smallest final ERR and NERR while the results from the 3 term model are vice versa.

The performance of the 3 term model will be found to be similar to that of discrete models synthesized by the input approximation methods given in later figures.

Due to the tradeoffs discussed between the time and frequency domains and between initial simulation and later simulation errors, The 4 term Taylor model is selected for the Comparison III study. This model also has only one more numerator term when compared to the other techniques.

Figures 4.26 and 4.27 illustrate the frequency response comparisons for the input approximation method with zero-order hold and various phase compensation at $T = 0.167$. These plots show that the best magnitude match occurs at r around 0.8 while the best phase response occurs at around $r=0.4$.

Figures 4.28 through 4.31 illustrate the discrete models frequency response. Discrete models' coefficients are as given in Table 4.5. The legend of each plots

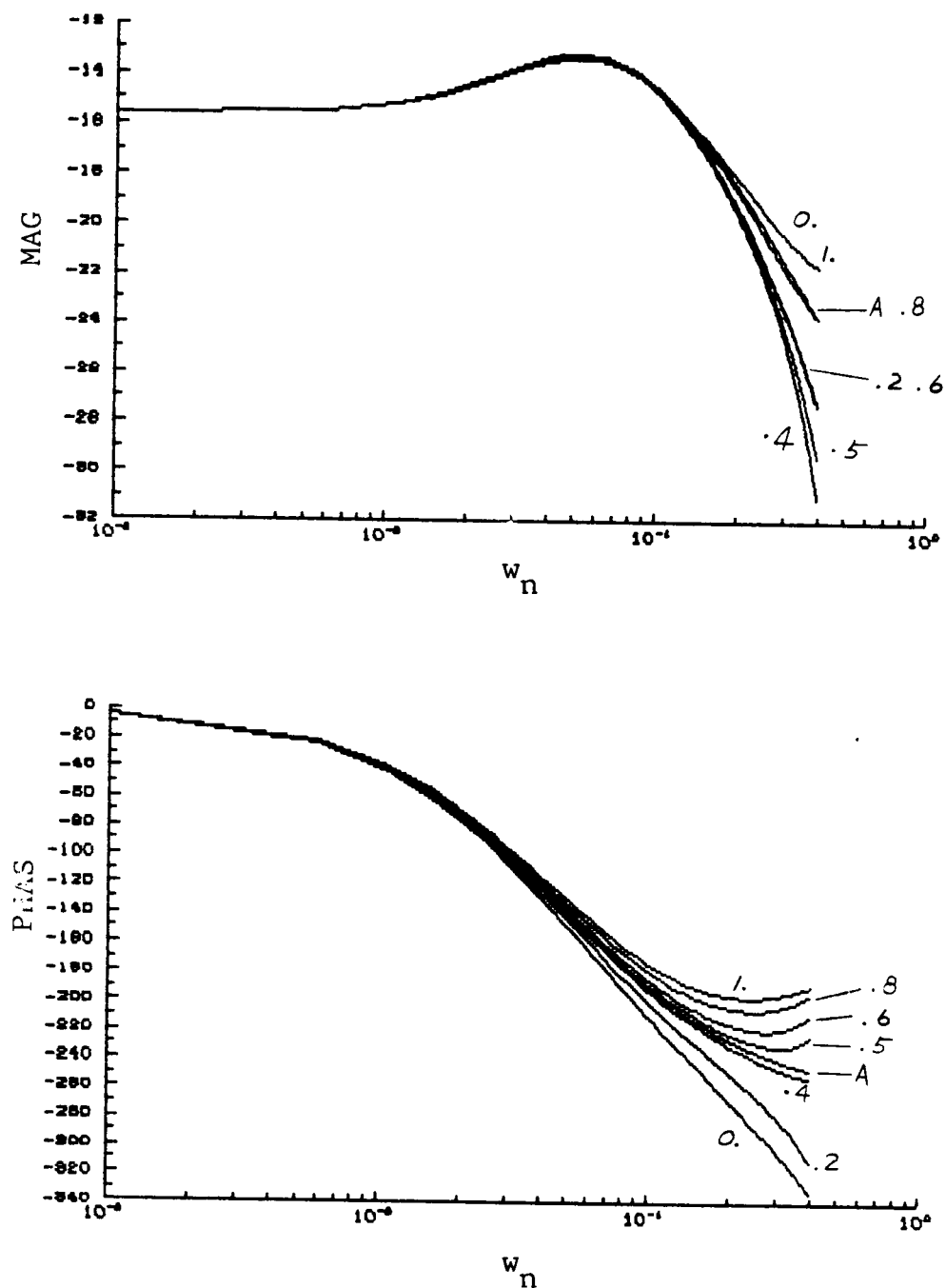


Figure 4.26. Frequency Response of Comparison III System with Input Approximation with Zero-Order Hold and Various Phase Compensator ($T = 0.167$)

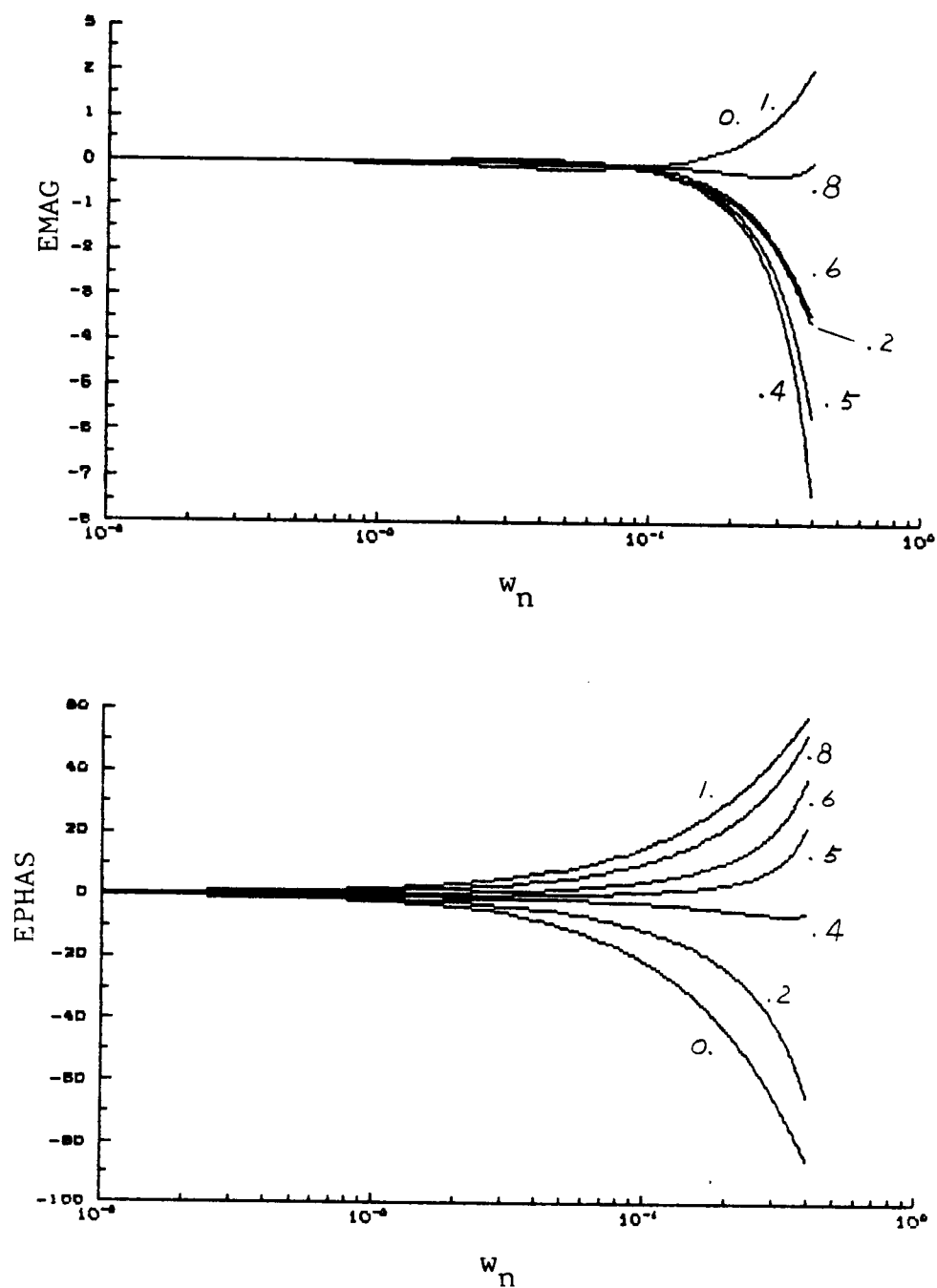


Figure 4.27. EMAG and EPHAS Plots of Comparison III
 System Input Approximation with Zero-Order
 Hold and Various Phase Compensator
 ($T = 0.167$)

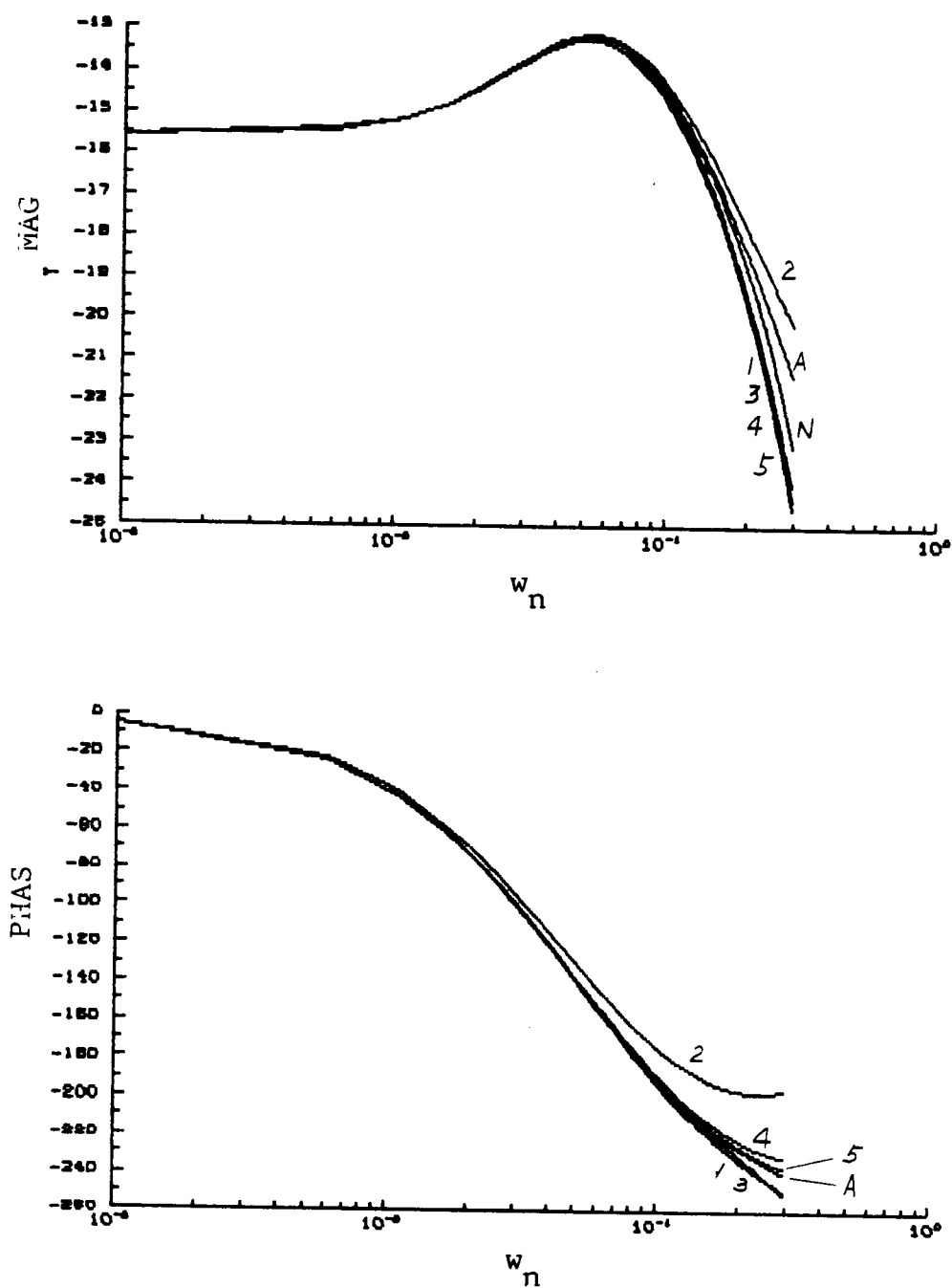


Figure 4.28. MAG and PHAS Plots of Comparison III System ($T = 0.167$, $r = 0.5$)

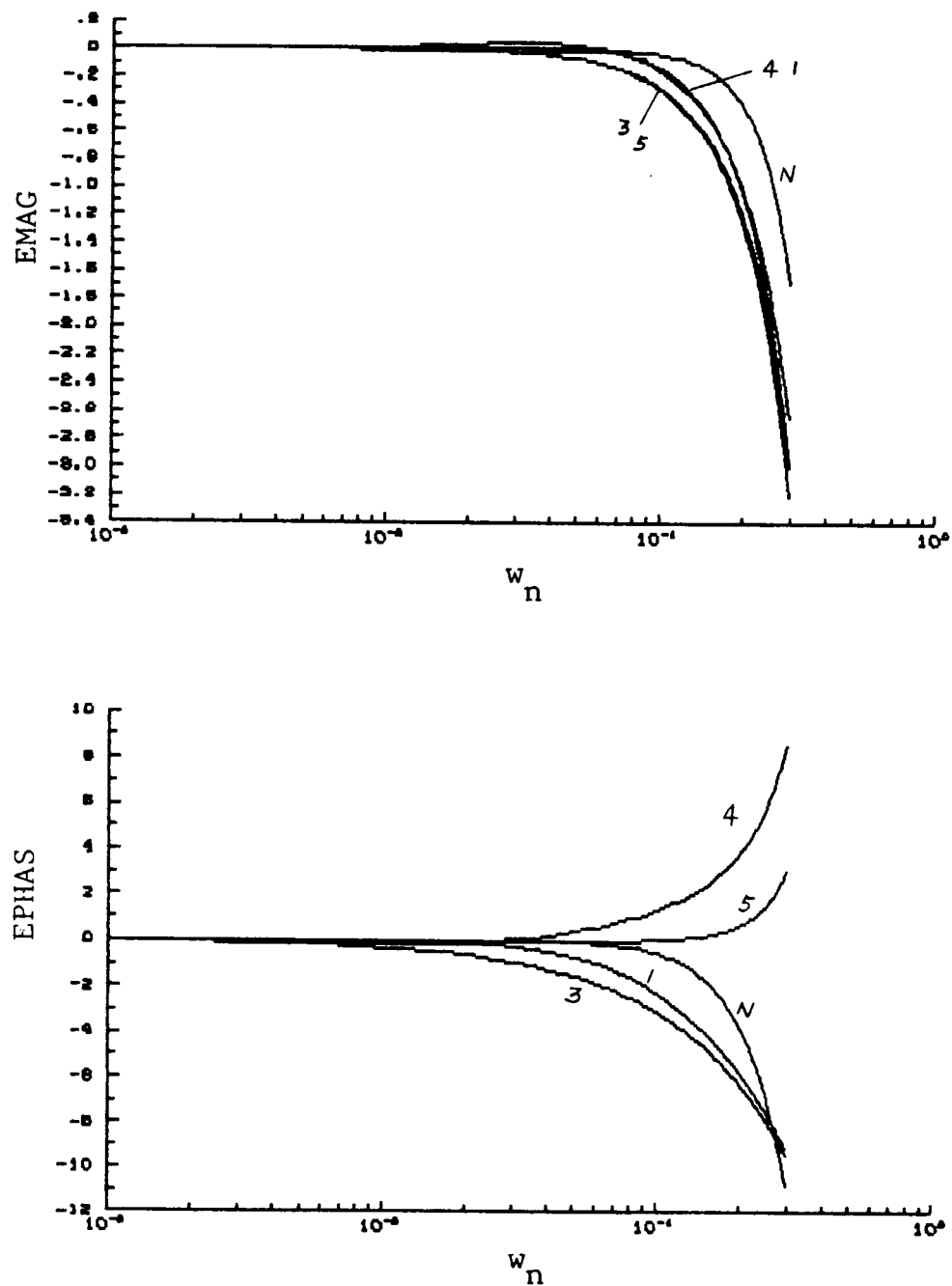


Figure 4.29. EMAG and EPHAS Plots of Comparison III System ($T = 0.167$, $r = 0.5$)

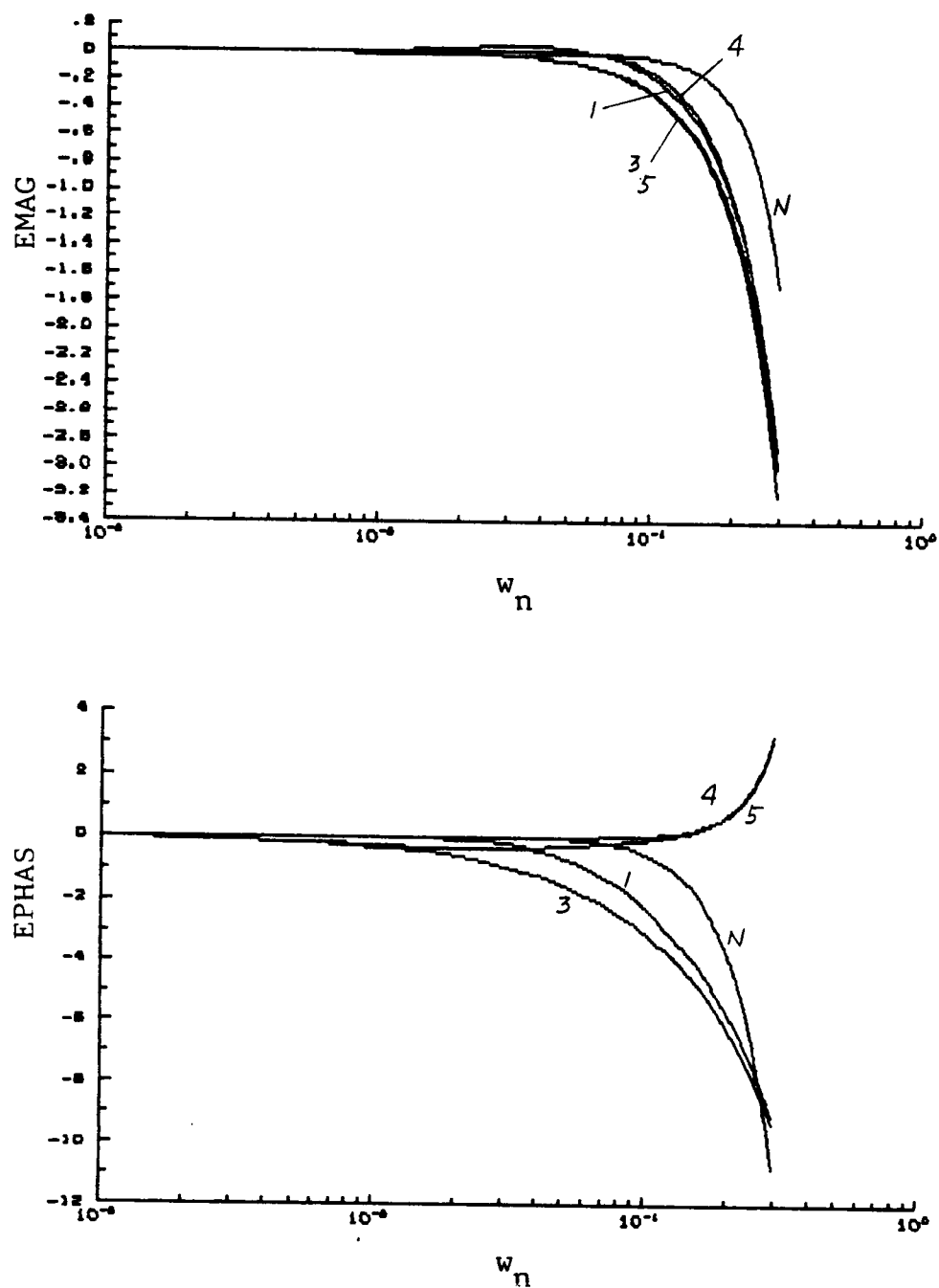


Figure 4.30. EMAG and EPHAS Plots of Comparison III
System ($T = 0.167$, $r = 0.46$)

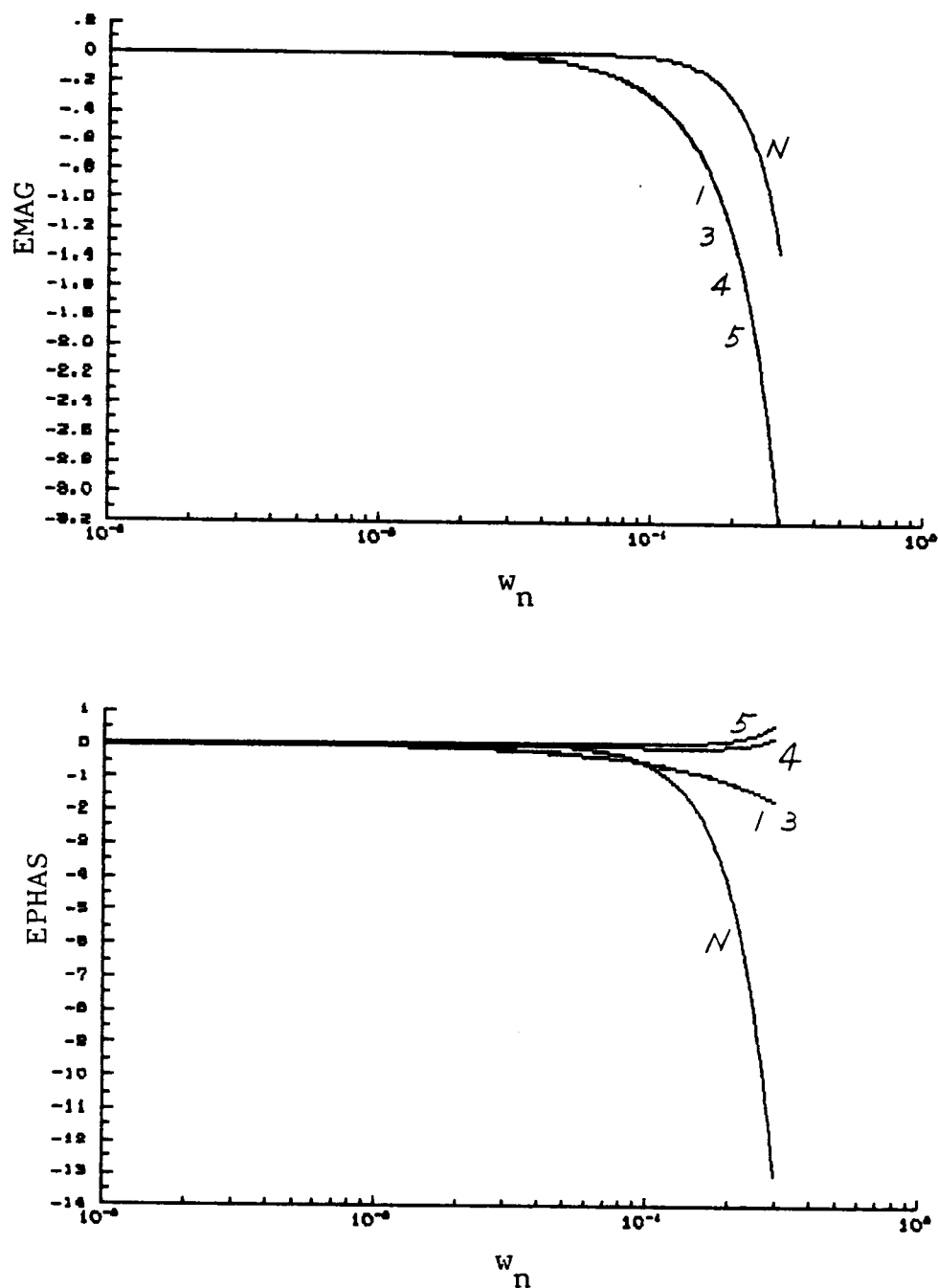


Figure 4.31. EMAG and EPHAS Plots of Comparison III System. ($T = 0.03$, $r = 0.49$)

Table 4.5. Coefficients of Discrete Models in Comparison III

Method	Coefficient
Tustin	$a_0 = c - 2/T$ $a_1 = 2*c$ $a_2 = c + 2/T$ $b_0 = (2/T+a)*(2/T+b)$ $b_1 = 2*a*b - 8/T**2$ $b_2 = (2/T-a)*(2/T-b)$
Match Pole-Zero (z=0)	$a_0 = r_1$ $a_1 = -r_1*exp(c*T)$ $a_2 = 0$ $b_0 = 1$ $b_1 = -(exp(-a*T)+exp(-b*T))$ $b_2 = exp(-(a+b)*T)$
Match Pole-Zero (z=-1)	$a_0 = r_2$ $a_1 = r_2*(1-exp(c*T))$ $a_2 = -r_2*exp(c*T)$ $b_0 = 1$ $b_1 = -(exp(-a*T)+exp(-b*T))$ $b_2 = exp(-(a+b)*T)$
Input Approximation with Hzo and phas compensator	$a_0 = w_1*x_1$ $a_1 = w_2*x_2-w_1*exp(-b*T)-x_1*exp(-a*T)$ $a_2 = -w_2*exp(-b*T)-x_2*exp(-a*T)$ $b_0 = 1$ $b_1 = -(exp(-a*T)+exp(-b*T))$ $b_2 = exp(-(a+b)*T)$
Input Approximation with Htr	$a_0 = tw_1 + tx_1$ $a_1 = tw_2 + tx_2 - tw_1*exp(-b*T) - tx_1*exp(-a*T)$ $a_2 = -tw_2*exp(-b*T) - tx_2*exp(-a*T)$ $b_0 = 1$ $b_1 = -(exp(-a*T)+exp(-b*T))$ $b_2 = exp(-(a+b)*T)$

Table 4.5. (Continued)

Defined constants:

```

p = -(a+c)/(a-b)
q = -1-p
r1 = c*(1-exp(-a*T))*(1-exp(-b*T))/(a*b*(1-exp(c*T)))
r2 = r1/2
w1 = p*(1-exp(-r*a*T))/a
w2 = p*(exp(-r*a*T)-exp(-a*T))/a
x1 = q*(1-exp(-r*b*T))/b
x2 = q*(exp(-r*b*T)-exp(-b*T))/b
tw1 = p*(a*T-1+exp(-a*T))/(T*a**2)
tw2 = p*(1-exp(-a*T)-a*T*exp(-a*T))/(T*a**2)
tx1 = q*(b*T-1+exp(-b*T))/(T*b**2)
tx2 = q*(1-exp(-b*T)-b*T*exp(-b*T))/(T*b**2)

```

For 2nd-order system:

$$G(s) = \frac{c - s}{(s+a)(s+b)}$$

provided lead to the observations:

1. The effect of the amount of phase compensation used with the zero-order hold input approximation method is significant. In Figure 4.29 where $r=0.5$, method 4 starts to show significant phase error for ω_n larger than about 0.005 and has considerable more error than method 5. In Figure 4.30 where r has been adjusted to 0.46, the response of method 4 is very close to that of method 5 up to the highest frequencies plotted.
2. Again, the Taylor method (4 terms) possesses the best discrete model with respect to the magnitude response.
3. The two input approximation methods provide the best phase angle response (when method 4 is optimized).
4. Changing the sample period from 0.167 to 0.03 has essentially no effect on the Bode plot errors exhibited by the Taylor method, while making this reduction in T causes considerable improvement in EPHAS for all the methods. Also as T is reduced the results for EMAG converge for all other method.

Figures 4.32 through 4.35 illustrate the error

indexes used for the time domain response. Method 2 is again excluded due to its much poorer performance. The local error levels, ERR, relative to the input amplitude have a range from $\pm 0.06\%$ to $\pm 1.5\%$.

Conclusions derived from this study are:

1. The Taylor method, again, gives the best performance expect for the first two sample times. The next best methods are the two input approximation methods.
2. As expected the ERR decreases as T is decreased or as NSC is increased. This local error decrease is proportional to the decrease in T and inversely proportional to the increase in NSC. These effects on ERR also appears to be multiplicative, that is, $ERR = \alpha(T/NSC)$. This relation holds quite well for Figures 4.32 to 4.35.

Incorporating the time and frequency domain performance, the Taylor method is the recommended discretization algorithm followed by the input approximation methods.

4.8 Summary

The major concern of this research, providing guidelines to select the best available digital algorithm

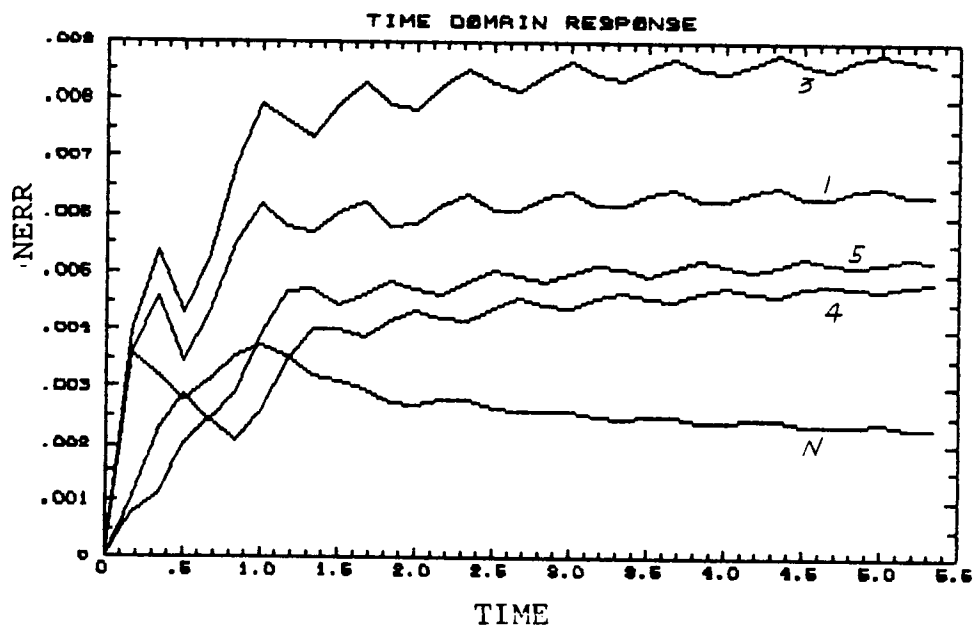
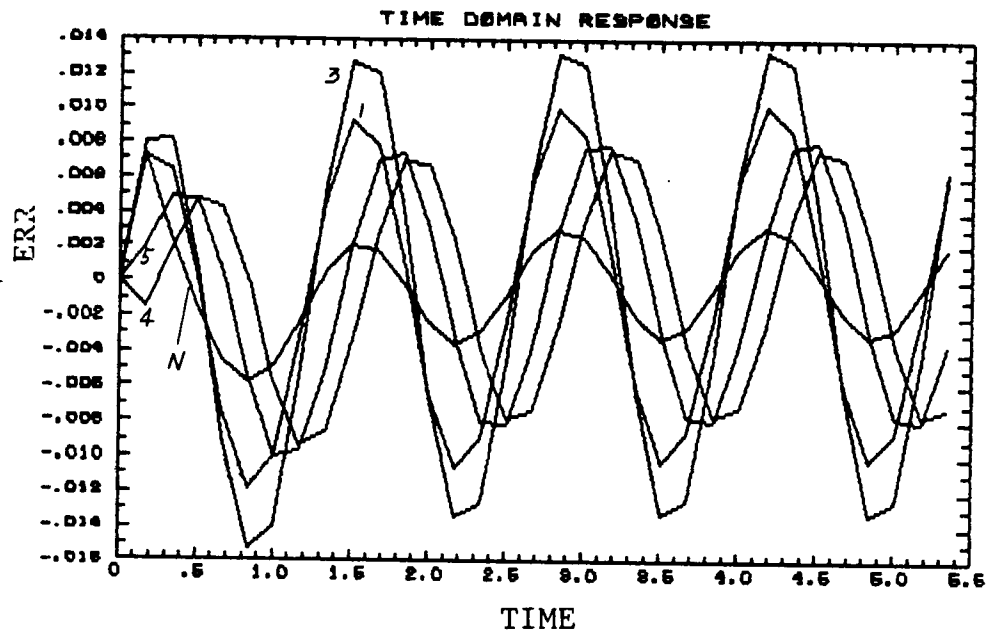


Figure 4.32. ERR and NERR Plots of Comparison III
System ($T = 0.167$, $NSC = 8$, $r = 0.5$)

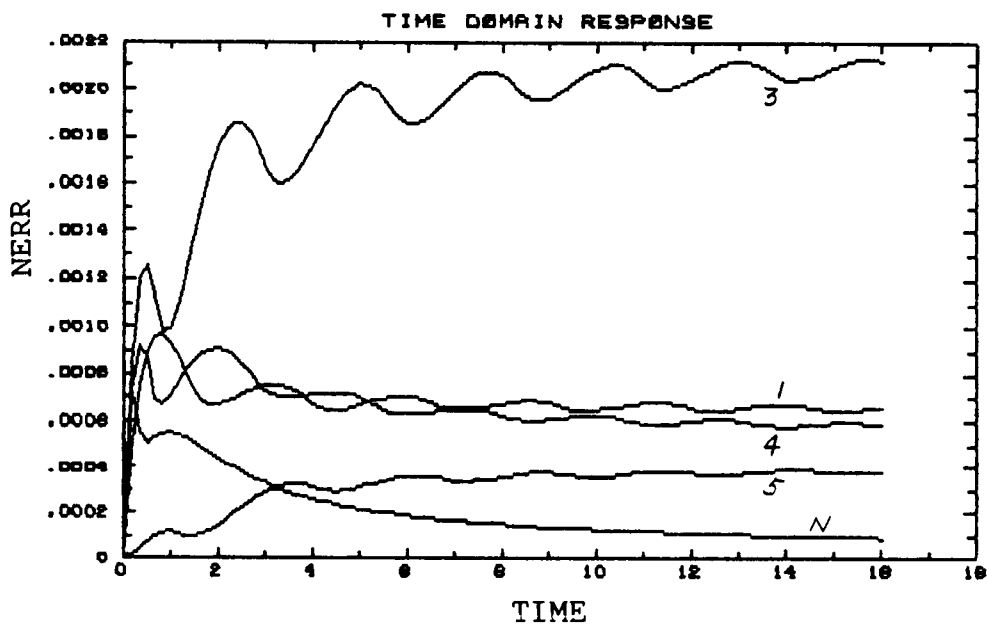
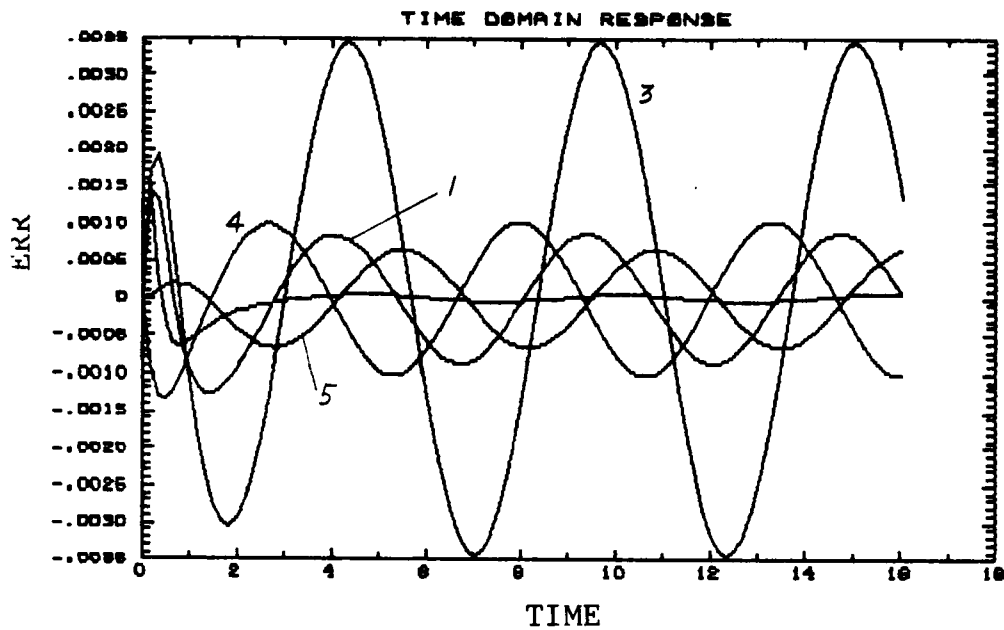


Figure 4.33. ERR and NERR Plots of Comparison III
System ($T = 0.167$, $NSC = 32$, $r = 0.5$)

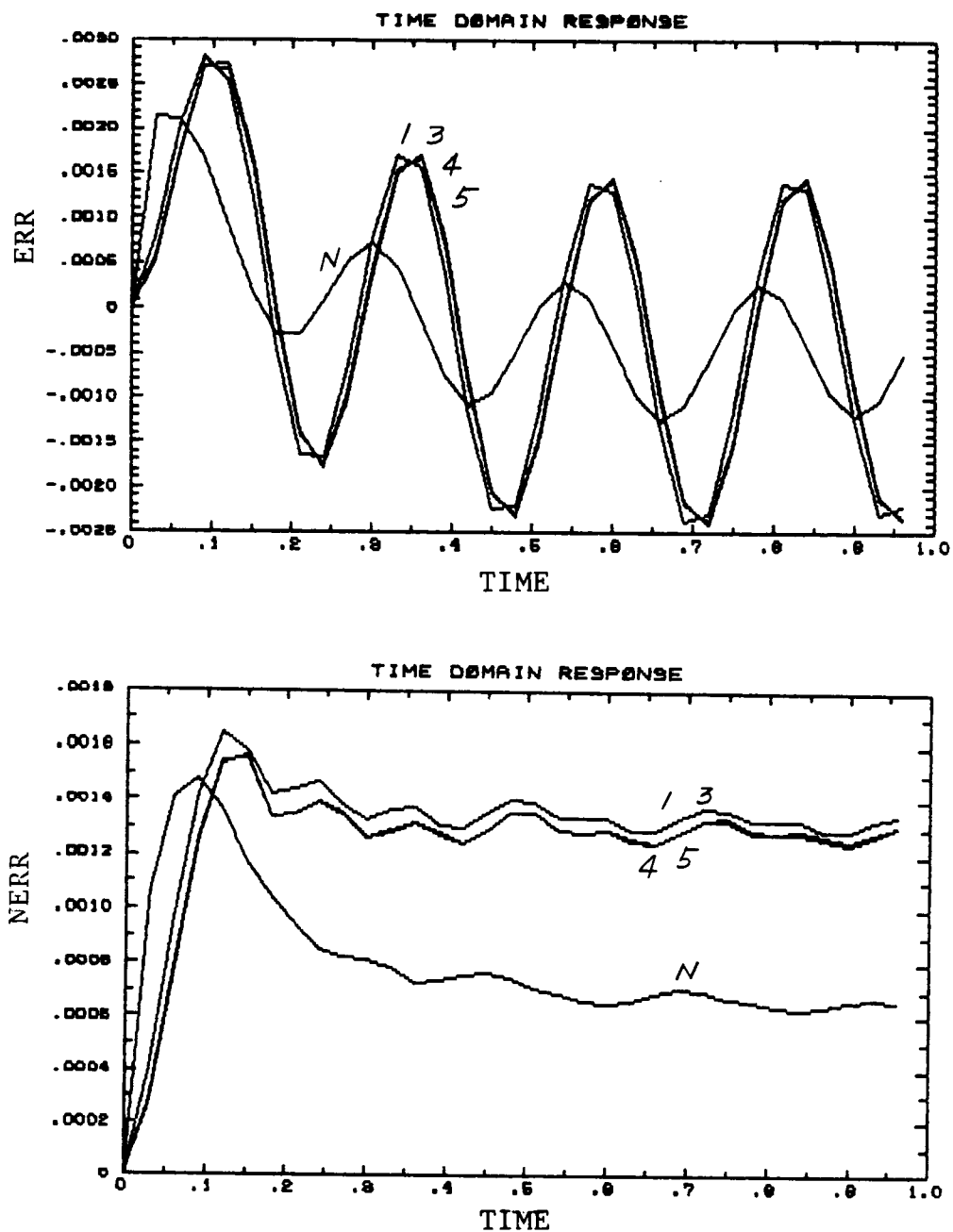


Figure 4.34. ERR and NERR Plots of Comparison III
System ($T = 0.03$, $NSC = 8$, $r = 0.49$)

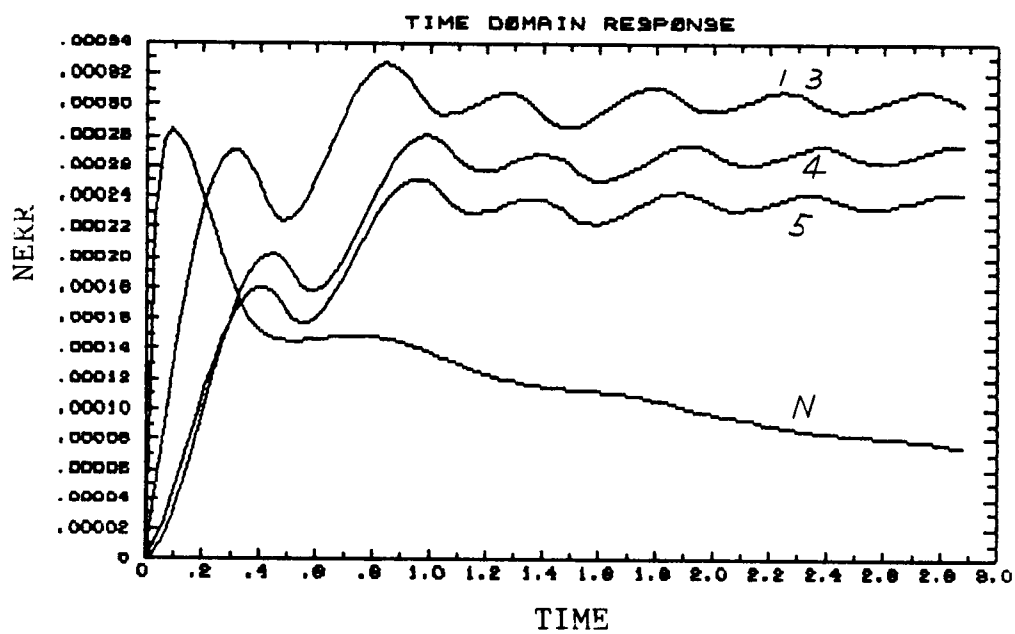
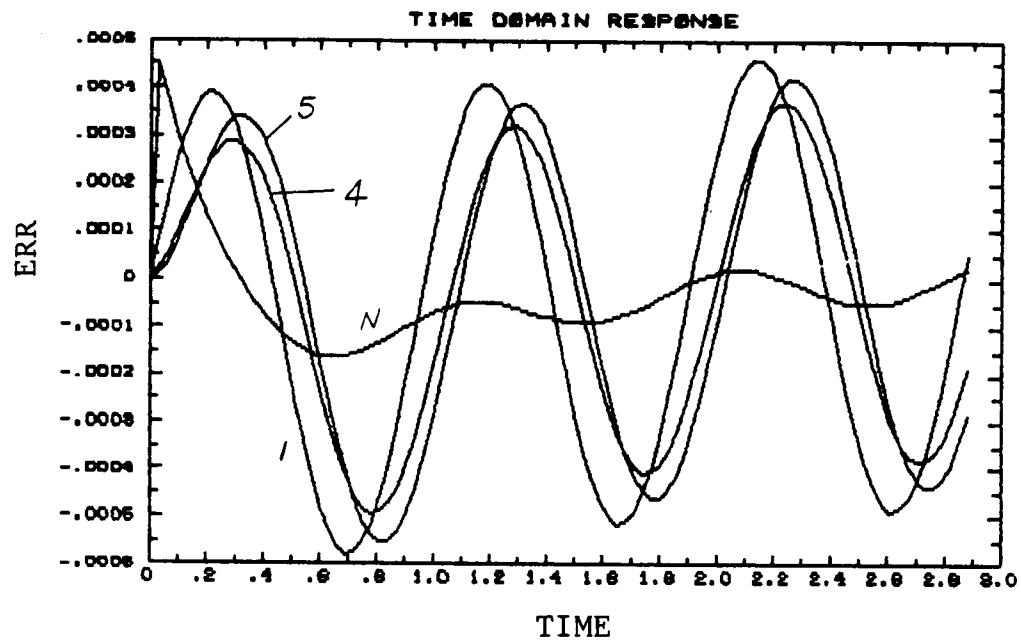


Figure 4.35. ERR and NERR Plots of Comparison III
System ($T = 0.03$, $NSC = 32$, $r = 0.49$)

to simulate an open loop continuous process subjected to arbitrary inputs, was the theme of this chapter. Several detailed studies were carried out to develop such guidelines.

The methods reviewed in the literature were classified into three different approaches: (1) optimal discrete models, (2) overall redesign, and (3) optimal discretization methods. Of these, the approach of optimal discretization methods was selected for this study.

Specifically six discretization methods, listed in Table 4.2, were applied to three transfer function models. The major criteria for the discrete algorithms was to provide the best time domain transient response, therefore the local error, ERR, and global error, NERR, time domain indexes were formulated. The frequency domain response in Bode plot form was also utilized in judging the discretized models.

Detail observations and conclusions were given after completing a study on each transfer function. Presentation of the final or overall conclusions and guidelines based on integrating the results from the three comparison studies is delayed to Chapter V.

CHAPTER V

GUIDELINES, CONCLUSIONS AND FUTURE STUDIES

This research studied digital simulation algorithms with the goal of being able to accurately simulate the dynamic behavior of linear continuous systems. A major concept was to employ digital simulation techniques that incorporate the fact that the models being simulated are linear. By taking advantage of the fact that the models are linear, the simulation development may take less effort, need less computer memory, use less computation time, and be implemented in smaller computer system environments.

5.1 Linear Model Simulation

The simulation problem given the most attention in this work was that of simulating an open loop linear continuous process. Three types of simulation oriented mathematical descriptions were discussed in Chapter II: (1) state space model, (2) transfer function or input-output models, and (3) models made of interacting processes, e.g. block diagram system descriptions. This work directly addressed the first two types of descriptions. It is also well known that state space representations can be transformed into transfer

function models and vice versa. The matrix exponential review and the development of new matrix exponential software were directed at the state space model formulation while the discretization methods address the transfer function model format.

5.2 Optimal Discretization Techniques

Optimal discretization techniques were selected to provide the procedures to synthesize the needed digital algorithms. The major reason for choosing the discretization methods is that they encompass the zero-order hold discretization algorithm (more correctly named the input approximation method with zero-order hold), the step invariant discrete state space equation, and the relatively new and promising Taylor method.

The zero-order hold discretization is of particular interest as it is widely used in applied process control and is very often the technique used to implement control schemes on programmable controllers. The step invariant discrete state space equation is an approximation to linear state space models often used in formulating a solution.

Six discretization methods were selected from the wide ranging and numerous techniques available in the discretization literature. The methods are listed in

Table 4.2. Method 1 is a substitution procedure, methods 2 and 3 are matched z-transform techniques, methods 4 and 5 are input approximation schemes, and method 6 is a new formulation that includes in its development concerns for both the frequency and time domain characteristics.

5.3 Simulation Study for Arbitrary Inputs

The six discretization methods were applied to three transfer functions (first order lag, underdamped second order lag and second order nonminimal phase system) to provide a basis for evaluating the resulting algorithms. The main objective function for the comparison was to match the continuous model's time domain transient response. The frequency response in terms of errors in the Bode plot also provided considerable insight.

5.4 Simulation for Arbitrary Input:

Conclusions and Guidelines

The conclusions and observations that are made from integrating the results from Comparisons I, II and III are now listed:

1. The discretization methods surveyed all provide stable discrete approximation models in open loop simulations.

2. All the digital algorithms give good results if a sufficiently small step size is used; also, all give poor estimates of the transient response if too large a step size is employed but do give the correct steady state value if the forcing function is held at a constant value. For example, see Figure 4.19.
3. Frequency response evaluations only provide information on the system's steady state characteristic (magnitude ratio and phase angle). Thus, if the discrete models' poles and final values match those of the continuous systems, they will have satisfactory frequency responses at low input frequency even when the sample period is large compared to system's transient dynamics. Figure 4.19(b) shows simulations using an obviously too large step size where the discrete models match the steady state response exactly but have very poor transient responses. In the continuous domain, the unique relationship between the frequency and time responses makes the frequency-domain analysis a powerful tool. However, the discontinuous nature of systems in the z -domain needs to be taken into account carefully, when shifting the analysis technique

from the s- to the z-domain. Perhaps, the frequency response results for discrete models are only consistent with the steady state time domain performance in the worst cases. From the simulation standpoint, the selected sample frequency should be high enough to reflect systems' transient behavior and to sufficiently approximate the input forcing function.

4. The accuracy of the approximation models heavily depends on the sample frequency. The selection of sample frequency should be based on both the input forcing function frequency and the system dynamic behavior.

All models have smaller ERR and NERR as the sample period, T , decreases and as the sample rate, NSC, increases. The local error is roughly proportional to T and inversely proportional to NSC. These two effects are multiplicative so ERR is roughly given by $ERR = \alpha (T/NSC)$. This relation holds fairly well for each comparison over the range of T and NSC studied. The value of α is different for Comparison I, II and III.

5. The rule of thumb that for practical implementations the sample frequency should be at least 10 times greater than the input frequency

that is, $wn < 0.1$, is shown in all frequency comparisons. When $wn > 0.1$, significant errors are appearing in the frequency response plots.

6. The major conclusion is that the Taylor method exhibits the best time domain behavior over the three comparisons as it has the smallest local error, ERR, and normalize error, NERR. In Comparisons I and III, this method sometimes has larger errors for the first one or two sample periods.
7. To implement the Taylor method requires the selection of the number of numerator or a-coefficients to be used in the model. This selection involves considering tradeoffs among the initial error, the local error, the global (normalized) error and the computation effort. Empirically the results from this research indicate that one or two more terms be used in the numerator than the contains denominator for optimal results.
8. The Taylor method in Comparisons I and III shows a phenomena of having larger error indexes at the first or second sample times. The study of the Taylor method with different number of numerator terms in Figures 4.23 to 4.25 seems to

confirm that this is arising from the need of more input data (sample periods) to make a good approximation of the input forcing function. When this phenomena is present (Comparison II did not show this character), it appears to be more prominent as the difference between the numerator and denominator terms increases. A possible reason for Comparison II not showing the larger errors at the initial sample times is noted in item 9.

9. The input approximation methods, here including the Taylor methods, seem to be effected by the rate at which the output responds to changes in the input. For illustration, the derivative of the step response at $t = 0$ of the 2nd order underdamped lag in Comparison II is 0 and these methods show no large initial errors. The derivatives of the step response at $t = 0$ for the 1st order lag and nonminimum phase processes in Comparisons I and III are not 0 and the initial errors are present.
10. The two input approximation methods represent the other best simulation methods.
11. The input approximation with zero-order hold and adjustable phase compensator has a high

potential as a good discretization method. For a selected sampled period, the discrete model can achieve high accuracy by adjusting the compensating phase, r , properly. A caution is the optimal value for r , in terms of the t -domain response, could be quite dependent on the type of input to the model.

12. Another rule of thumb often found in proactical applications textbooks is illustrated by this study: The introduction of a sampler in a signal path is equivalent to adding one half of the sample time as deadtime to the process. This would translate into using $r = 0.5$ in the phase compensator. Values of around 0.5 for r are generally found to give the best phase angle plots. This is more true when the sample rates are high. As larger sample times are used the best value for r changes from 0.5; see Figure 4.8 where $r = 0.478$. At least, in some cases the results are quite sensitive to changes in r ; see Figure 4.29 and Figure 4.30 where r is changed from 0.5 to 0.46.
13. If the phase angle response is the only criterion, the input approximation method with triangular hold performed better than the other

discretization techniques.

14. The matched z-transform methods should not be used for general process simulation. They consistently gave poor results in both the time and frequency domains (Method 2 was usually left off the comparison figures). Application of matched pole-zero methods is limited to high sample frequencies and low input frequency. Mapping infinite zeroes to $z = -1$ yields consistently better performance than mapping infinite zeroes to $z = 0$; the later usually generates unacceptable phase deviation.
15. Tustin method is the simplest scheme to apply but its use is not recommended for general simulation. If it is important to match the Bode plot characteristics at some frequency the method might be quite useful due to the prewarping procedure. An example is the discretized notch filter.

These conclusions are drawn from application of the techniques to only three relatively simple transfer functions and only with sin wave inputs. The results merit testing with higher order linear models and other inputs.

5.5 Simulation Methods for Invariant Inputs

Chapter III derived the step and ramp algorithms for several transfer functions which are tabulated in Tables 3.2 and 3.3. The step invariant method treats exactly cases where step changes occur in the input at the sample times. The ramp invariant method treats exactly cases where ramp inputs occur between sample times with slope changes at the sample instant.

Therefore whenever a linear model is subjected to either of these two input types an exact digital algorithm can be developed. Examples of situations where this might occur are in the design of digital controllers and simulation of a process under digital control where the controller updates its output at each sample time.

5.6 New Matrix Exponential Algorithm

An augmented matrix is introduced whose matrix exponential provides both the discretized matrices needed in the step invariant discretized state space equation which is commonly used to solve models stated in the state space form. A Fortran program has been developed and run on some simple examples. This new procedure is being investigated further and compared with other matrix exponential procedures used for state space model solution.

5.7 Future Studies

Finally this work is closed with a list of topics, small questions and thoughts which provide a starting point or spark some addition research related to that reported in this thesis.

5.8 Closely Related Topics

1. Perform additional comparison following the format used in this work on higher order transfer functions.
2. A specific comparison of interest would be for a PI and a PID controller. This might provide a superior implementation for these control laws.
3. Investigate the relationship ERR is roughly proportional to (T/NSC) . This might provide a way for selecting the needed sampling time based on only one simulation. From an initial simulation the proportionality constant could be calculated, then the ratio of (T/NSC) would be fixed once the desired error level, ERR, is specified.
4. Addition error indexes which might be added to the time domain criteria are a local relative error, $(y^*-y)/y$, and a global relative error,

$(y^*-y)/\Delta y$, where y is the continuous system output, y^* is the discrete model output and Δy is the magnitude of the largest output from the continuous model.

5. Formulate the Taylor method with a phase compensator. The Taylor method always performed very well in terms of the magnitude ratio but showed some error with respect to the phase angle.
6. Formulate the input approximation method using a zero-order hold and phase compensator to also include a gain compensator. This method generally performed well but had a larger normalized error, NERR, than the Taylor method. This error might be reduced via a gain compensator.
7. Compare the Tustin method with prewarping to the Taylor method. The example where prewarping was applied led to lower errors in both the magnitude ratio and phase angle over all frequencies (see Figure 4.12)

5.9 Other Topics

1. Application and comparison of the new matrix exponential procedure to existing matrix exponential procedures. This topic is

currently being addressed.

2. Section 3.6.3 outlined a procedure to calculate the matrix exponential of matrices that have real and distinct eigenvalues via a companion matrix while section 3.6.4 present that for matrices having real and repeated eigenvalues. The use of the companion matrix gives a very simple procedure to calculate the matrix exponential exactly, where as, other procedures need to invlive approximations. If a companion matrix exists for matrices with complex conjugates and would allow the matrix exponential to be calculated exactly, the basis might be in place for a new approach to matrix exponential computation.
3. Extend the study of digital algorithms to the simulation of systems involving interacting subsystems. This question is of considerable interest if there are any loops that can be followed through the subsystems as the errors occurring in the simulation of one subsystem can be propagated round the loop. The problem of simulation stability is now involved besides that of accuracy.
4. Apply the discretization methods to models containing deadtime or to stiff models.

LIST OF REFERENCES

LIST OF REFERENCES

1. Astrom, K. J., P. Hagander and J. Sternby, "Zeros of Sampled System," Proc. 19th IEEE Conf. on Decision and Control, Albuquerque, 1077-81, 1980.
2. Astrom, K. J. and B. Witlenmark, Computer Controlled Systems Theory and Design, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1984.
3. Beale, G. O. and G. Cook, "Digital Simulation and Optimization Via Random Search Techniques," Numerical Methods for Differential Equations and Simulation, North-Holland Publishing Company, 1978.
4. Birdwell, D., EE 5271 Class Note, Dept. of Electrical Engineering, University of Tennessee, Knoxville, 1986.
5. Doetsch, G., Guide to the Application of Laplace and Z-Transforms, Van Nostrand Reinhold, New York, 1971.
6. Edwards, J. W., "Flight Test Experience in Digital Control of Remotely Piloted Vehicle," AIAA Guidance and Control Conf., Paper No. 72-883, Aug., 1972.
7. Finlayson, B. A., Nonlinear Analysis in Chemical Engineering, McGraw-Hill, New York, 1984.
8. Fletcher, R. and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer J., V. 6, No. 2, 1963, pp. 163-68.
9. Forsythe, W., "Digital Algorithms for Prediction, Differentiation, and Integration," Trans. of Institute of Measurement and Control, 1:1, 45-56, Jan., 1978.
10. Forsythe, W., "Algorithm for Digital Control," Trans of Institute of Measurement and Control, 5;3, 123-135, July, 1983.
11. Fowler, M. E., "Numerical Methods for the Synthesis of Linear Control Systems," Automation, vol. I, 1963, pp. 207-225.
12. Fowler, M. E., "A New Numerical Method for Simulation," Simulation, vol. 8, pp.308-310, June, 1967.

13. Franklin, G. F. and J. D. Dowch., Digital Control of Dynamic Systems, Addison-Wesley, Philippines, 1980.
14. Hamming, R. W., Numerical Methods for Scientists and Engineerings, McGraw- Hill, Inc., N. Y., 1973.
15. Hartley, T. T. and G. O. Beale, "Integration Operator Design for Real-time Digital Simulation," IEEE Trans on Industrial Electronic, vol.IE-32, No. 4, Nov., 1985.
16. Irby, T. C., "A Numerical Technique for Solving a Class of N-th Order Ordinary Differential Equations," Numerical Methods for Differential Equations and Simulation, IMACS; North-Holland Publishing Company, 1978.
17. Jerri, A. J., "The Shannon Sampling Theorem-Its Various Extensions and Application: A Tutorial Review," Proc. IEEE, 65, 1565-95, 1977.
18. Jury, E. I., Theory and Application of the Z-Transform Method, John Wiley and Sons, Inc., New York, 1973.
19. Katz, P., "Sample Rate Selection for Digital Control of Aircraft," Ph.D. Dissertaion, Stanford, Calif., Sept., 1974.
20. Katz, P., Digital Control Using Microprocessors, Prentice-Hall, Inc., 1981.
21. Kuo, B. C., Automatic Control Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1982.
22. Kuo, B. C., Digital Control System, Holt, Rinehart and Winston, Inc., 1980.
23. Kuo, B. C. and D. W. Peterson, "Optimal Discretization of Continuous-Data Control Systems," IFAC Automatica, vol. 9, No. 1, 1973, pp.125-129.
24. Kuo, B. C., G. Singh and R. A. Yackel, "Digital Approximation of Continuous -Data Control System by Point-by-Point State Comparison," Computers and Elec. Eng., vol. 1, Pergamon Press, 1973, pp. 155-170.
25. Lindonnff, D. P., Theory of Sampled-Data Control Systems, John Wiley and Sons, Inc., New York, 1965.

26. Moler, C. B. and C. F. VanLoan, "Nineteen Dubirous Ways to Compute the Exponential of a Matrix," SIAM Review, vol. 20, No. 4, Oct., 1978.
27. Parrish, E. A., E. S. McVey and G. Cook, "The Investigation of Optimal Discrete Approximations of Real-Time Flight Simulation," NASA CR.N76-19158, 1974.
28. Phillips, C. L. and H. T. Nagle, Digital Control System Analysis and Design, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1984.
29. Rattan, K. S., "Digitalization of Existing Continuous Control Systems," IEEE Trans on Automatic Control, vol. AC-29, no. 3, March, 1984.
30. Rattan, K. S. and H. H. Yeh, "Discretizing Continuous-Data Control Systems," Computer-Aided Design J. vol. 10, No. 5, Sept., 1978.
31. Rurden, R. L. and J. D. Faires, Numerical Aanlysis, PWS Publishers, 1985.
32. Sage, A. P. and S. L. Smith, "Real-time Digital Simulation for System Control," Proceedings of the IEEE, vol.54, Dec., 1966, pp. 1802-1812.
33. Schroeder, D., "A New Optimization Procedure for Digital Simulation," University of New Mexico, 1972.
34. Smith, J. M., Mathematical Modeling and Digital Simulation for Engineering and Scientists, Jone Wiley and Sons, Inc., New York, 1977.
35. Steiglitz, K., "Computer Aided Design of Recursive Digital Filters," IEEE Trans, Audio Electroacoustic, vol. Au-18, no.2, June, 1970.
36. Takahashi, Y., M. J. Rabins, and D. M. Auslander, Control and Dynamic Systems, Addison-Wesley, Reading, MA, 1970.
37. Tou, J. T., Digital and Sampled Data Control Systems, McGraw-Hill, Inc., 1959.
38. Tustin, A., "A Method of Analysing the Behaviour of Linear Systems Intermis of Time Series," Journal of IEEE, vol. 94, II-A, May, 1947.

39. Van Loan, C. F. and G. H. Golub, Matrix Computations, The Johns Hopkins University Press, Baltimore, Maryland, 1983.
40. Ward, R. C., "Numerical Computation of Matrix Exponential with Accuracy Estimate," SIAM J., Numer. Anal., 14, 600-610, 1977.

VITA

Chun-Yao Lien was born in Taipei, Taiwan on November 12, 1959. He completed his elementary and secondary schooling in Taipei. In September 1978, he attended Tunghai University in Taichung, Taiwan and received the Bachelor of Science degree in Chemical Engineering from that University in June 1982.

After graduating from the college, he joined the Chinese Army where he served as an instructor in Chinese Infantry School and completed his 2-year compulsory military services in August 1984. One month later, in fall of 1984, he came to the United States and entered the Graduate School at the University of Tennessee at Knoxville to pursue the Master's degree in Chemical Engineering. He accepted a graduate teaching assistantship in Chemical Engineering Department of that school to support this study. This degree will be awarded in August 1986.