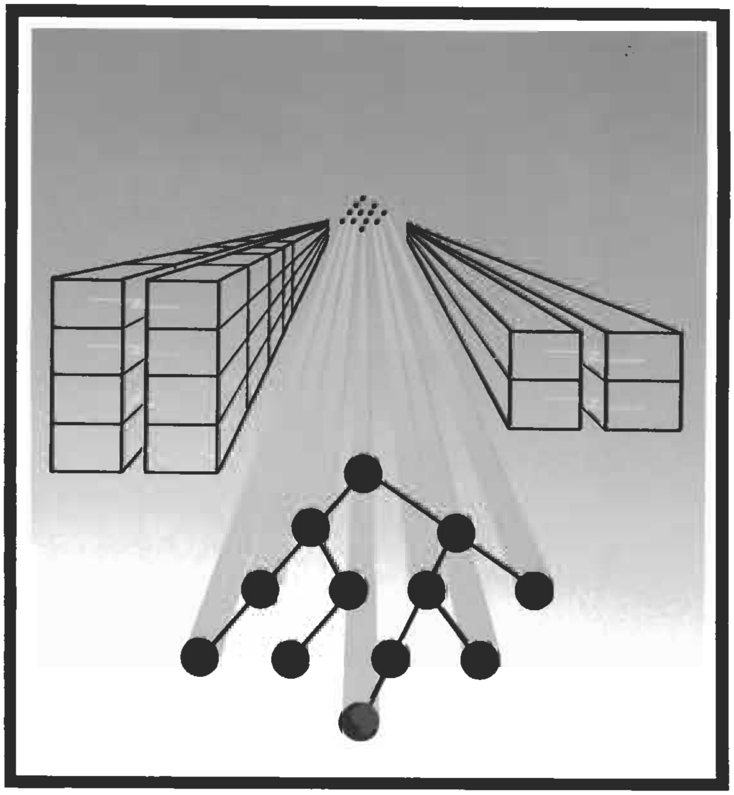
Series 9800 Desktop Computers

Linear Systems Analysis

For the System 35/45



HEWLETT PACKARD

Linear Systems Analysis

Hewlett-Packard Desktop Computer Division
3404 East Harmony Road, Fort Collins, Colorado 80525

Table of Contents

Preface

The Linear Systems Analysis package provides you with several tools which are used for the analysis of linear systems. These programs run on the HP 9835A, 9845B, and 9845C Desktop Computer.

The instructions given in this manual assume that you have a working knowledge of your desktop computer and the Linear Systems Analysis field. Complete operating instructions are given in the appropriate manuals supplied with each desktop computer. The major program difference between the computers is that the graphic CRT can be used for plotted outputs on the 9845B and 9845C, but the 9835A **must** use a plotter. Otherwise, program operation is essentially the same. In this manual, where a difference in key definitions occurs, all definitions are given. The keys CONTINUE (9835A) and CONT (9845B/C) have the same function; the word CONTINUE is used throughout this manual to represent both.

If any program which is described in this manual fails, please contact your local Sales and Service Office.

Chapter 1
Introduction

Description

The Linear Systems Analysis Software contains a set of subprograms which can be used for analysis and design of single-input, single-output, linear, time-invariant systems. The subprograms are accessed through a main driver program by pressing the appropriate special function key. Subprograms are available to tabulate or graphically display system step, impulse, and frequency response characteristics. The frequency response data is available in both Bode and Nyquist diagram formats. There is also a subprogram to tabulate or plot root loci.

For input, systems are assumed to be represented by a transfer function or an interconnection of transfer functions (a block diagram). Individual transfer functions are input by entering the coefficients of their numerator and denominator polynomials. Block diagrams are input by representing all cascade subblocks, feedback subblocks, parallel subblocks, and simple transfer functions as nodes of a binary tree. Then you enter the tree structure into the machine. The theory of binary tree representations, as well as examples of the conversion process, are given later in this manual.

Once a block diagram is input, a subprogram is available for generating the corresponding overall transfer function. The step, impulse, frequency response programs, and the root locus program operate only on the overall system transfer function. The binary tree representation of a block diagram can also be edited and modified or stored on a mass storage medium for later retrieval.

The plotting routines include the following features. Output can be displayed on the CRT, (9845B/C only), thermal printer, or plotter if available. You can either have the computer automatically scale your plots or scale them yourself. You can also re-scale and display your plots without re-calculating data points (except root locus). As the overall transfer function is modified, multiple plots can be superimposed on the same graph. There is also a labeling procedure that allows you to title and label plots. Finally, cursor and digitization procedures are available so that data values may be located by cursor and easily read from plots.

System Configuration

To use the Linear Systems Analysis Software with the HP 9835A, you need the following items:

- HP 9835A Desktop Computer, Option 201 – 128K bytes of Read/Write Memory
- A plotter, such as the HP 9872
- A printer (optional)
- Linear Systems Analysis Software – HP part number 09835-15190. This includes a tape cartridge containing the programs, a user's manual, and a key overlay. Part numbers are listed in Appendix C.

To use the Linear Systems Analysis Software with the HP 9845B or 9845C, you need the following items:

- HP 9845B/C Desktop Computer, Option 204 – 187K bytes of Read/Write Memory
- Graphics Display Subsystem, Option 311 or Option 700
- Thermal Line Printer, Option 560 or Option 561
- A plotter (optional)
- Linear Systems Analysis Software – HP part number 09845-15190. This includes a tape cartridge containing the programs, a user's manual, and a key overlay. Part numbers are listed in Appendix C.

Theory

It is assumed that you are familiar with Linear Systems Analysis. This section is simply meant as a refresher.

System Representation

A single-input, single-output, linear, time-invariant system is completely characterized by the differential equation relating the input signal u(t) to the output signat y(t):

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots a_1 \frac{dy}{dt} + a_o y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \ldots b_o u$$

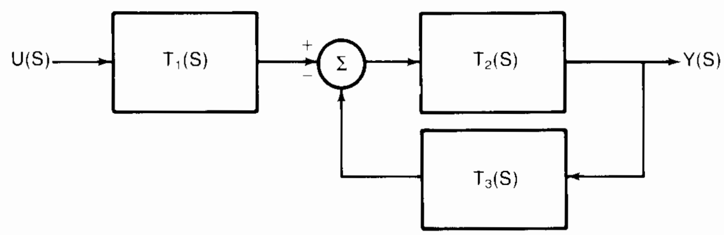
The system transfer function T(s) is defined as the ratio of the Laplace transforms of the input and output signals. Thus T(s) =Y(s)/U(s) can be determined by transforming the differential equation as follows:

$$T(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_o}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_o} \triangleq \frac{N(s)}{D(s)}$$

The transfer function is then in the form of a ratio of two polynomials, a numerator polynomial N(s) and a denominator polynomial D(s).

Block Diagrams

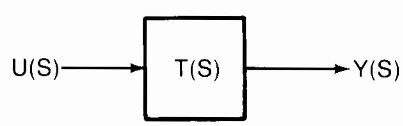
Complex linear systems can be represented as an interconnection of simpler subsystems each of which is described by a transfer function. The interconnections are usually described by a block diagram such as the one shown below:



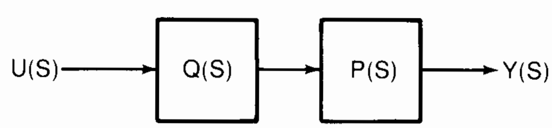
For any block diagram, there is a set of rules which can be used to calculate the overall transfer function relating the input U(s) to the output Y(s).

The Linear Systems Analysis Package allows block diagrams (which are of the following forms) to be entered:

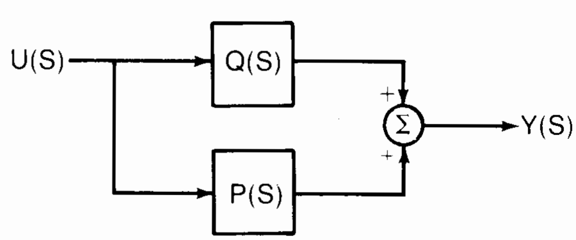
Simple Transfer Function



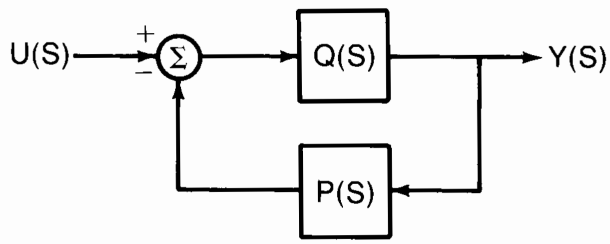
Cascade Interconnection



Parallel Interconnection

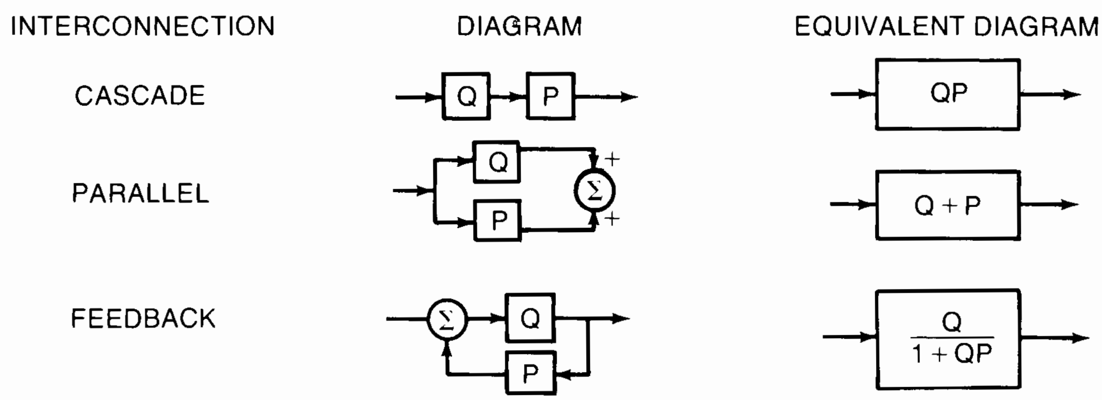


Feedback Interconnection



The blocks Q(s) and P(s) can in turn be either simple transfer functions or other cascade, parallel, or feedback subsystems.

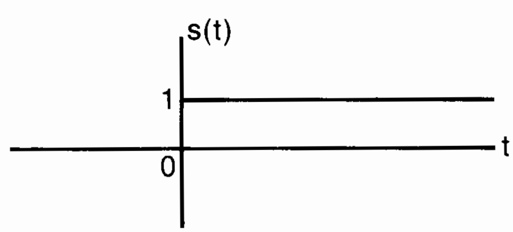
For cascade, parallel, or feedback interconnections, the following equivalence rules can be applied to calculate overall system transfer functions.



The program that generates overall system transfer functions from block diagrams of this type applies these equivalence rules.

Step and Impulse Response

The step response of a system is defined as the output y(t) as a function of time when the system is initially at rest and the input signal is the unit step function s(t) shown below:



Similarly, the impulse response is the output y(t) when the system is initially at rest and the input is a unit impulse function, $\delta(t)$. The function $\delta(t)$ has the properties:

$$\delta(t) = 0 \quad t \neq 0$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

These responses are calculated by first deriving a state equation model for the overall system transfer function T(s) = Y(s)/U(s) of the form:

$$\frac{d\bar{X}(t)}{dt} = A\bar{X}(t) + Bu(t)$$

$$y(t) = C\bar{X}(t) + Eu(t)$$

m, the degree of N(s), must be less than or equal to n, the degree of D(s). In this case, $\bar{X}(t)$ is an n-dimensional column vector, and A, B, C, E are (nxn), (nx1), (1xn), and (1x1) dimensional constant matrices. For such a model, the step response and the impulse response can be obtained by solving the following systems of equations for $t \geqslant t_0$:

STEP RESPONSE

$$\frac{d\bar{X}(t)}{dt} = A\,\bar{X}(t) + B$$

$$y(t) = C\,\bar{X}(t) + E$$

$$\bar{X}(t_0) = 0$$

IMPULSE RESPONSE

$$\frac{d\bar{X}(t)}{dt} = A\,\bar{X}(t)$$

$$y(t) = C\,\bar{X}(t) + E\,\delta(t)$$

$$\bar{X}(t_0) = B$$

These are solved numerically by using a differential equation solving program provided in the HP-9845 Numerical Analysis Package based upon the Adams-Bashford-Moulton method. For further details refer to the Numerical Analysis Package manual (Part No. 09845-10251). The step and impulse responses are always calculated assuming the system is at rest just prior to the beginning of the specified time interval.

Frequency Response (Bode and Nyquist diagrams)

The response of a linear system to an arbitrary input consists of two components: a transient response and a steady-state response. If the input is a sinusoid with frequency ω (radians/sec), then the steady-state response will be a sinusoid of the same frequency but with different amplitude and phase. Let $M(\omega)$ be the ratio of the output sinusoid amplitude to the input sinusoid amplitude, and let $\theta(\omega)$ (radians) be the difference between the output sinusoid phase and the input sinusoid phase. There is a simple relationship between the overall system transfer function $T(s)$, and $M(\omega)$ and $\theta(\omega)$. If $j \triangleq \sqrt{-1}$ then

$$T(s)|_{s=j\omega} = T(j\omega) = R(\omega) + jI(\omega) = M(\omega)e^{j\theta(\omega)}$$

where $R(\omega)$ = real part of $T(j\omega)$
$I(\omega)$ = imaginary part of $T(j\omega)$
$M(\omega) = (R^2(\omega) + I^2(\omega))^{1/2}$
$\theta(\omega) = \arctan(I(\omega)/R(\omega))$

$T(j\omega)$ is commonly referred to as the frequency response of a system. Two common methods of displaying frequency response information are by use of Bode and Nyquist diagrams.

The Bode diagram consists of two separate plots: the magnitude plot which graphs

$$20 \log_{10} M(\omega) \text{ vs. } \omega, \ \omega > 0$$

and the phase plot which graphs

$$\theta(\omega) \text{ vs. } \omega, \ \omega > 0$$

In both cases, the ω-axis is scaled logarithmically. $20 \log_{10} M(\omega)$ is known as the decibel value of $M(\omega)$, and $\theta(\omega)$ is plotted in degrees.

The Nyquist diagram is a plot of $T(j\omega) = R(\omega) + jI(\omega)$ in the complex plane as a function of ω, $-\infty < \omega < \infty$.

Points on these plots are calculated for frequency values between two positive user-specified limits, ω_{min} and ω_{max}:

$$0 < \omega_{min} < \omega_{max} < \infty$$

For the Nyquist diagram $T(-j\omega)$ is obtained by the relation

$$T(-j\omega) = R(\omega) - jI(\omega)$$

The interval $[\omega_{min}, \omega_{max}]$ is quantized into 200 points. Since $T(j\omega)$ is a nonlinear function of ω, to more evenly distribute the values of $T(j\omega)$, the values of ω are incremented geometrically as follows:

$$\omega_0 = \omega_{min}$$

$$\omega_{i+1} = \omega_i \times \text{Inc}, \ 1 \leq i \leq 199$$

$$\text{Inc} = 10^{\left[\log_{10}\left(\frac{\omega_{max}}{\omega_{min}}\right) / N-1\right]}$$

Phase and Gain Margin

Phase and gain margin are important parameters in stability analysis and are sometimes specified as design criteria. Let ω_p be the value of ω such that $M(\omega_p) = 1$ or $20 \log_{10} M(\omega_p) = 0$. Similarly, let ω_g be the value of ω such that $\theta(\omega_g) = -180°$. Then in terms of the Bode plots:

$$\text{Phase Margin} = 180° + \theta(\omega_p)$$

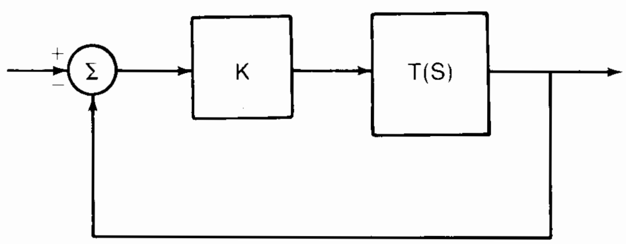
and

$$\text{Gain Margin} = -20\log M(\omega_g)$$

If the gain and phase margins are defined for a given system, then the Bode program automatically calculates them and prints the result.

Root Locus

The stability and transient response characteristics of a linear system with transfer function $T(s) = N(s)/D(s)$ are dependent upon the values of the roots of the denominator polynomial $D(s)$. One simple way of modifying these characteristics is to apply constant gain feedback to $T(s)$. The resulting block diagram becomes:



where K is the gain. The new overall transfer function becomes

$$T_K(s) = \frac{KT(s)}{1 + KT(s)} = \frac{KN(s)}{D(s) + KN(s)}$$

The stability and transient response characteristics of $T_K(s)$ are then dependent upon the values of the roots of $D(s) + KN(s)$. The root locus is a plot of these roots in the complex plane as a function of $K \geq 0$.

The root locus program plots the open loop roots of $N(s)$ and $D(s)$ and then the roots of $D(s) + KN(s)$ for values of K between two limits K_{min} and K_{max} specified by you where

$$0 < K_{min} < K_{max} < \infty$$

The gain K is quantized into N values over the interval $[K_{min}, K_{max}]$ where N is also specified by you. Because of the nonlinear relation between K and the roots of $D(s) + KN(s)$, to more evenly distribute the root values, K is incremented geometrically as follows:

$$K_0 = K_{min}$$

$$K_{i+1} = K_i \times \text{Inc},\ 1 \leq i \leq N - 1$$

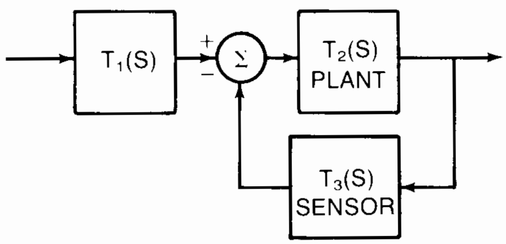
$$\text{Inc} = 10^{[\log_{10} (\frac{K_{max}}{K_{min}})/N - 1]}$$

All roots are calculated by using the polynomial root finding subroutine in the HP Numerical Analysis Software manual (Part No. 09845-10251). For details on this program refer to the manual for that package.

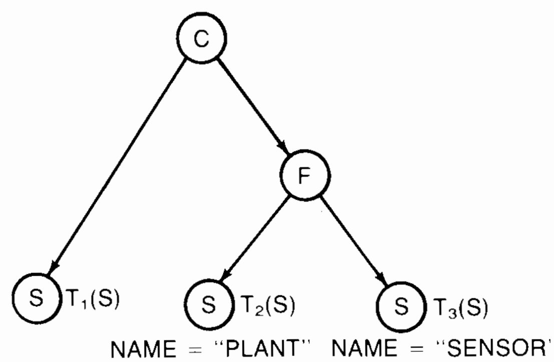
Binary Tree Representation of a Block Diagram

A system can be entered as either a simple transfer function or as a block diagram containing cascade, parallel, and feedback interconnections of transfer functions. This section gives instructions for deriving a binary tree representation of a block diagram. There is certain terminology associated with binary trees such as **node**, **root**, **leaf**, and **branch**. If you are unfamiliar with these concepts, refer to Appendix A. Further, Appendix B defines terms such as **summation point** and **branch point** which are used in describing block diagrams.

An example of a block diagram and the corresponding binary tree representation are shown below.



Block Diagram



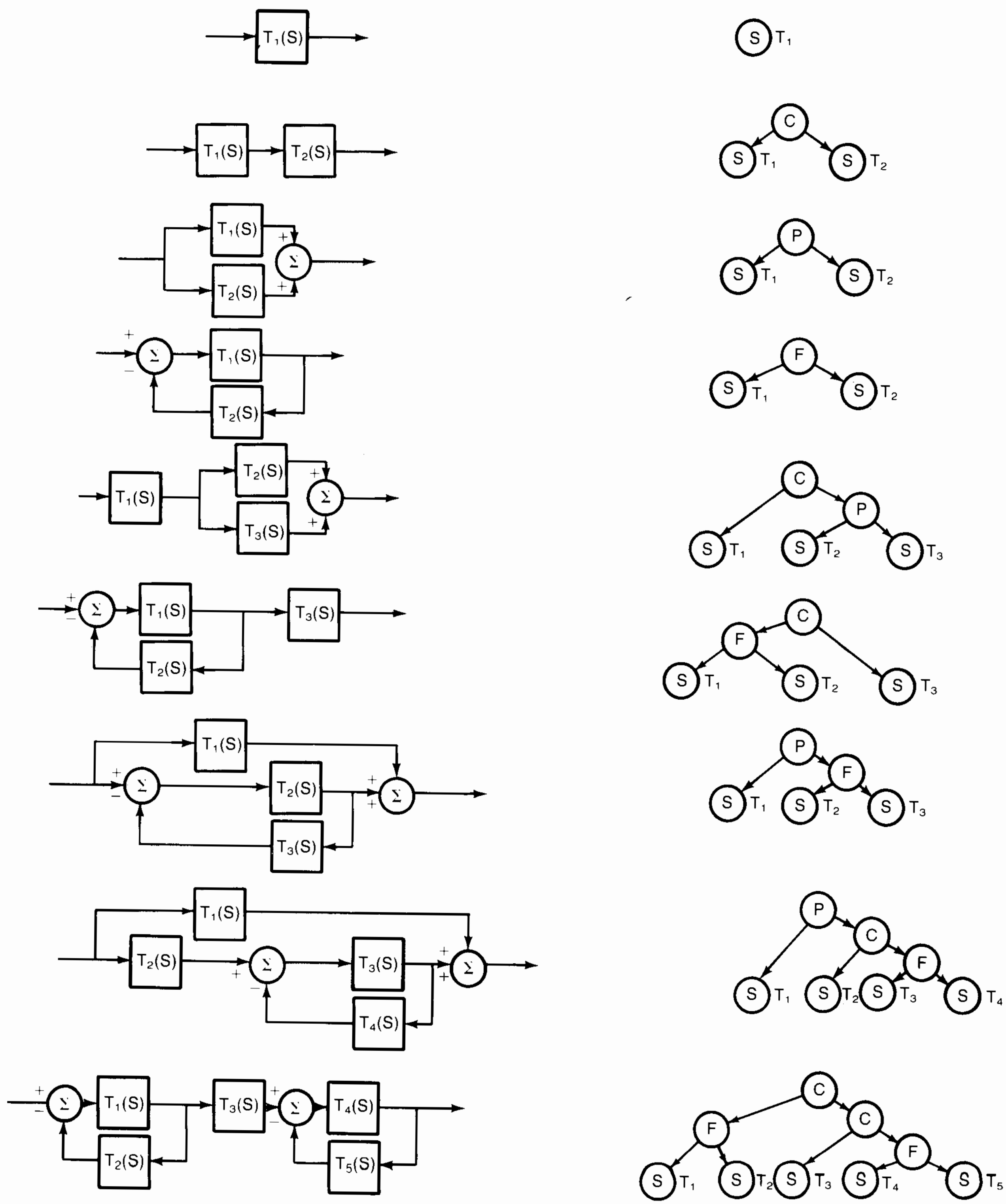
Binary Tree

In the binary tree representation of a block diagram each node corresponds to either a simple transfer function or a cascade, parallel, or feedback interconnection of subblocks and is indicated by the type designators S, C, P, or F respectively. A node can also be optionally assigned a name. If a node is of type S, then it also has a transfer function associated with it.

Some representative block diagrams and their corresponding binary tree representations are shown in the table which follows. In order to generate a binary tree representation, you must first write the block diagram in a specific form. A set of rules for constructing an appropriate block diagram are given followed by a procedure for generating the binary tree.

Sample Block Diagrams and Their Binary Tree Representation

BLOCK DIAGRAM | BINARY TREE

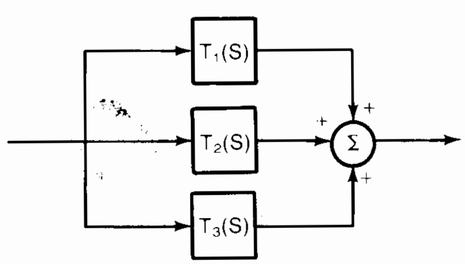


Rules For Block Diagram Construction

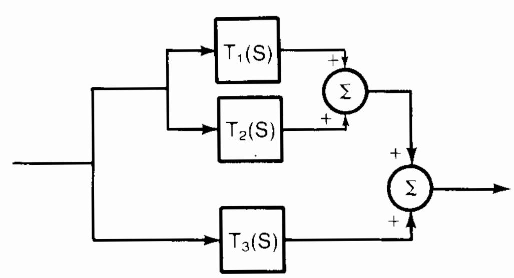
1. The input signal must enter the diagram at a single point.
2. The output signal must leave the diagram at a single point.
3. All branch points must have only two departing signals.
4. All summation points must have two entering signals and one departing signal.
5. A feedback signal must enter a summation point with a negative sign signifying subtraction. All other signals must enter summation points with positive signs signifying addition.
6. All paths in a feedback or parallel connection must contain at least one transfer function block. If not, unity transfer functions should be inserted.

The following three examples show incorrectly specified block diagrams and the corresponding correct diagram.

Example 1: Branch point with three departing signals and summation point with three entering signals.

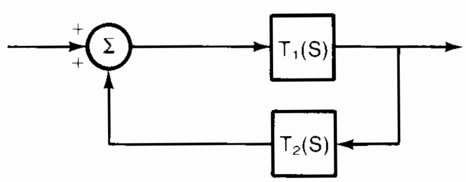


Incorrect Diagram

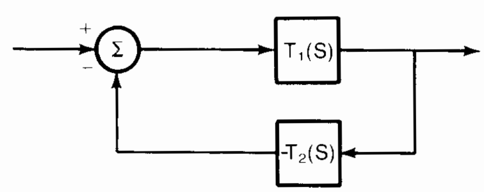


Correct Diagram

Example 2: A feedback signal entering summation point with positive sign.

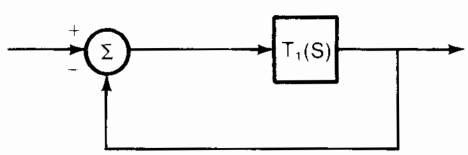


Incorrect Diagram

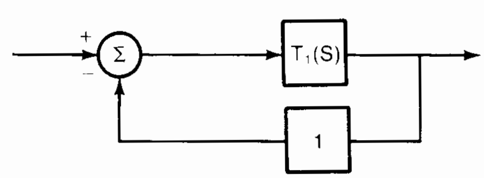


Correct Diagram

Example 3: A feedback path without a transfer function block.



Incorrect Diagram



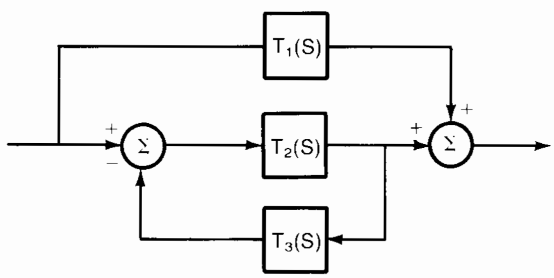
Correct Diagram

Procedure for Constructing a Binary Tree Representation of a Block Diagram

1. Draw a row of circles, one for each block diagram transfer function. These will correspond to leaf nodes in the tree and should be labeled type S. Assign to each node one of the transfer functions and a name.
2. Look for pairs of transfer functions which form cascade, parallel, or feedback connections. In case of ambiguity, use each transfer function only once. Draw a row of circles above the previous row, one for each connection. Associate each circle with a connection and label it by its type C, P, or F. Draw two branches out from each of these nodes and into the two nodes in the previous row whose transfer functions are involved in the connection. By convention, the ~~left~~ right branch of a feedback node should go to the feedback transfer function node.
3. Redraw a simplified block diagram by replacing each connection found in step 2 by a transfer function block. Temporarily associate each of these transfer functions with the corresponding connection node found in step 2.
4. If the simplified diagram contains a single block, then the tree has been constructed. Otherwise repeat steps 2 and 3.

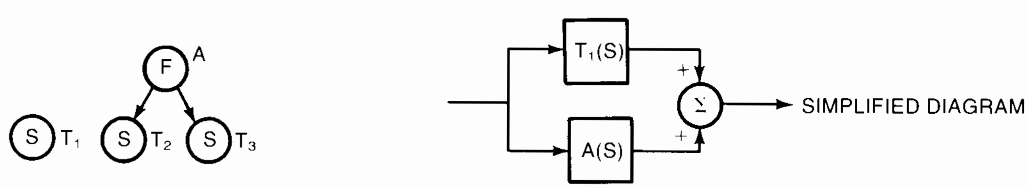
The following two examples illustrate the procedure:

Example 4:

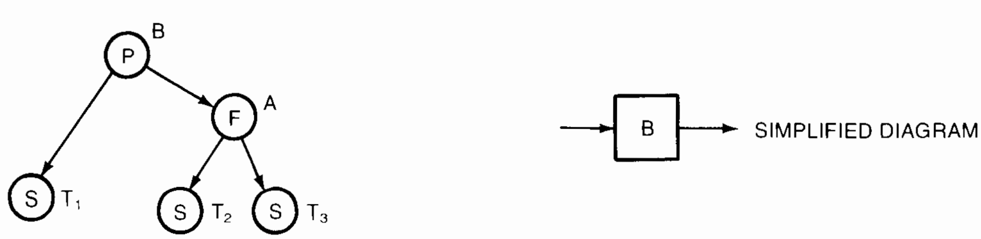


Performing step 1:

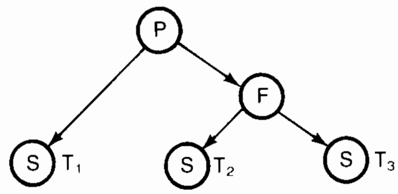
Performing steps 2 and 3, you can see that $T_2(s)$ and $T_3(s)$ form a feedback connection:



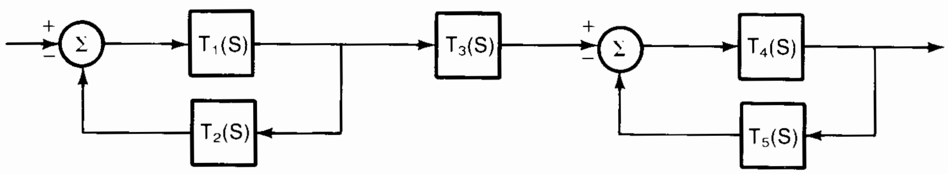
Repeating steps 2 and 3, you can see that $T_1(s)$ and $A(s)$ form a parallel connection:



Since the simplified diagram has only one block, the procedure is complete and the resulting tree is:



Example 5:

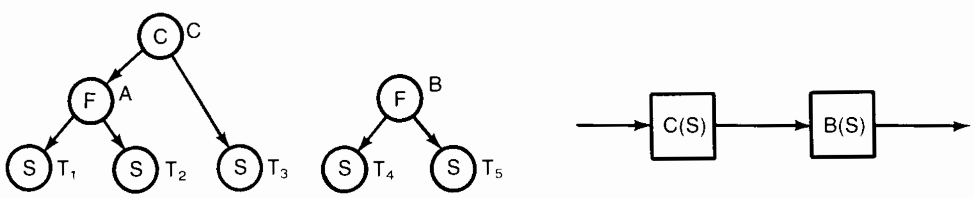


Performing step 1: 

Performing steps 2 and 3, you can see that $T_1(s)$ and $T_2(s)$, and $T_3(s)$ and $T_4(s)$ form feedback connections:

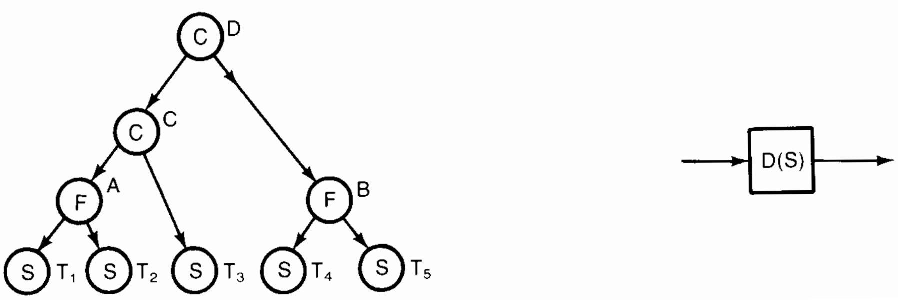
SIMPLIFIED DIAGRAM

Repeating steps 2 and 3, you can see that there is an ambiguity since $T_3(s)$ forms a cascade connection with A(s) and B(s). Since $T_3(s)$ can only be used once, we arbitrarily connect A(s) and $T_3(s)$:



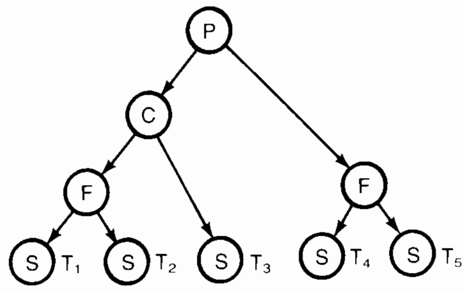
SIMPLIFIED DIAGRAM

Repeating steps 2 and 3, you can see that C(s) and B(s) form a cascade connection:



SIMPLIFIED DIAGRAM

Since the simplified diagram has only one block the procedure is complete and the resulting tree is:



Although the above procedure is somewhat cumbersome to apply, after a little practice you should be able to construct the binary tree for a diagram almost by inspection.

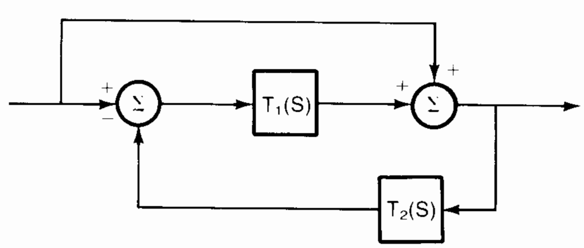
Limitations
Although binary tree representations using simple transfer function, and cascade, parallel, and feedback nodes exist for a large useful class of linear block diagrams, there are some linear block diagrams which cannot be represented in this manner. For example, two classes of such diagrams can be readily characterized:

Class 1: those which contain summation points inside feedback loops which add signals to the loop which are external to the loop

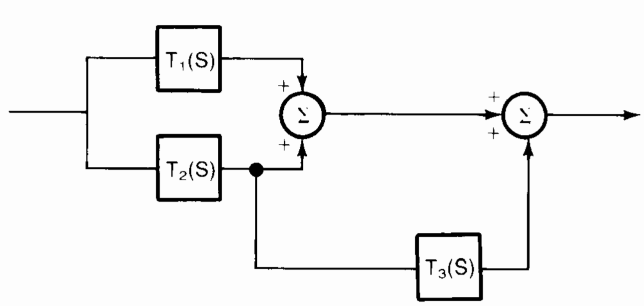
Class 2: those which contain parallel connections with additional branch points whose signals add to other signals outside the parallel connection

Example 6 is an example of a class 1 diagram and Example 7 is an example of a class 2 diagram.

Example 6:



Example 7:

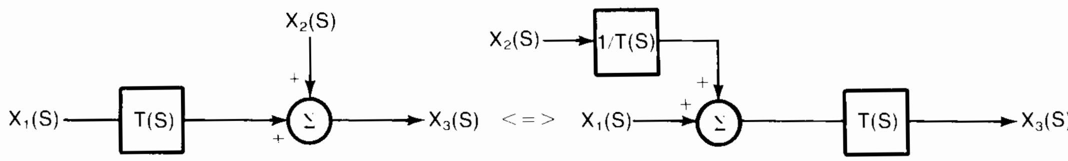


The following two rules can be applied to such diagrams to produce diagrams which are "analytically" equivalent but which enable binary tree representations.

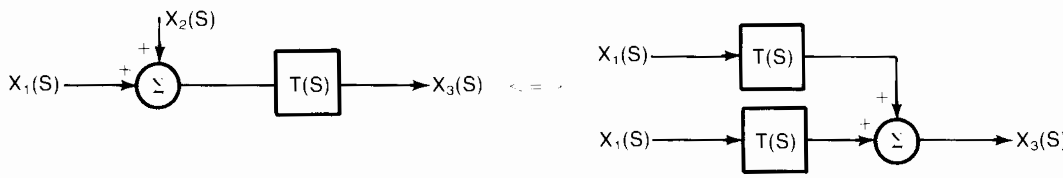
Rule 1: Interchange summation points and blocks

The following diagrams are equivalent:

(a)

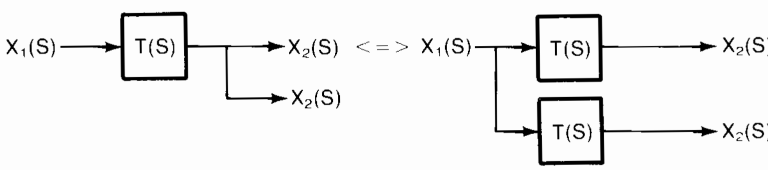


(b)

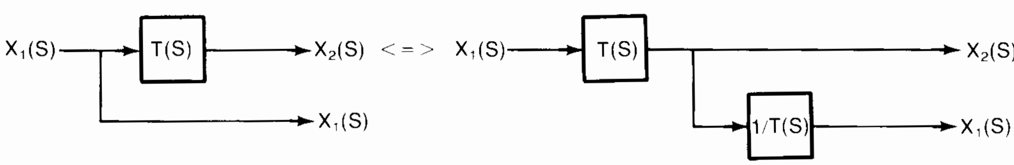


Rule 2: Interchanging branch points and blocks

(a)

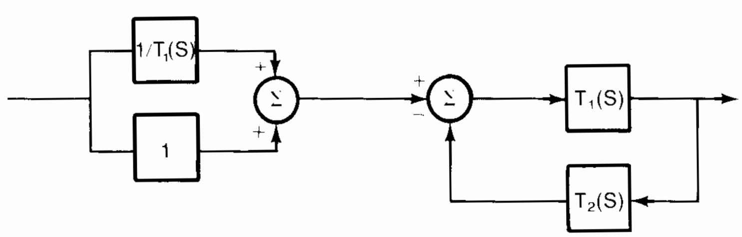


(b)



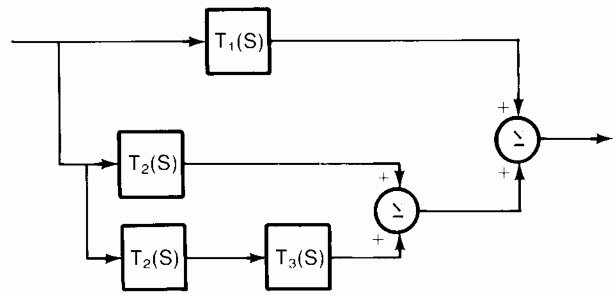
For example, by applying rule (1a) to the diagram of Example 6 we obtain:

Example (6a):



Similarly by applying rule (2a) to the diagram in Example 7 we obtain:

Example (7a):

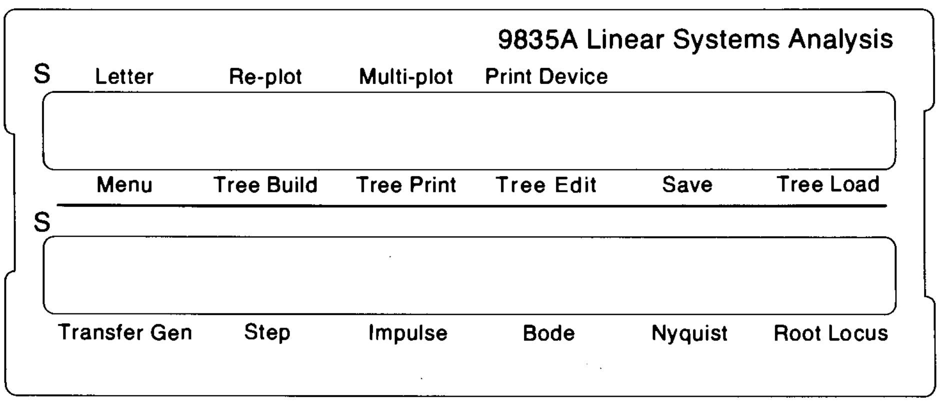


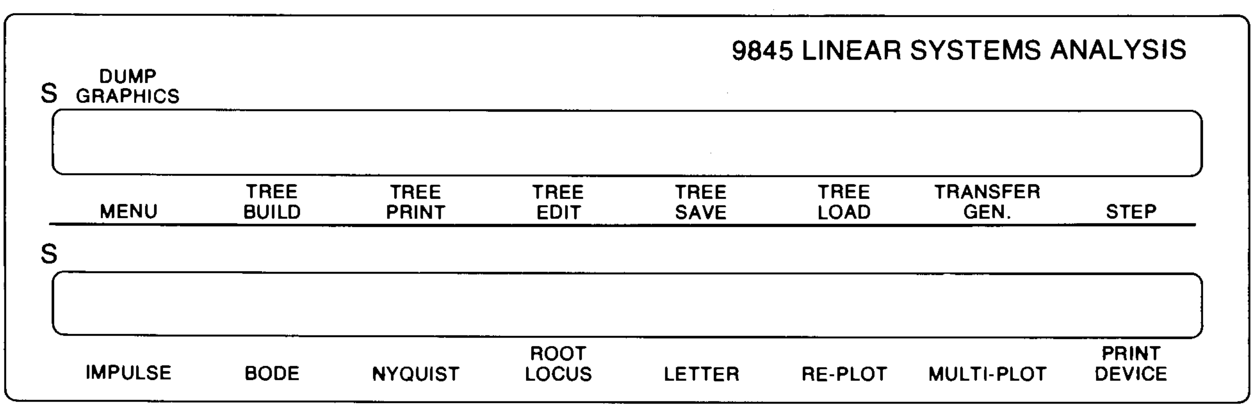
Care should be taken in using these equivalent diagrams for system analysis. Although they are equivalent analytically, the rules involve adding blocks to the diagram. Numerically, the resulting overall transfer functions can be of higher order. Although this does not affect frequency response plots, it can affect the step and impulse response plots. In particular, if some of the blocks involved correspond to unstable subsystems, the higher order equivalent block diagram may generate a state model which is internally unstable, and this may affect the solution produced by the numerical differential equation solving program.

Chapter 2
Instructions

Program Overview

This software package can be separated into two major categories of subprograms or operations. The operations in the System Specification category enable binary tree representations of block diagrams to be entered displayed and modified, and also allow calculation of the overall system transfer function associated with the entered diagram. Operations in the System Analysis category allow generation of Step, Impulse, Bode, Nyquist, and Root Locus data. Any operation can be executed by selecting one of the special function keys. The special function key templates are shown below.





The keys TREE BUILD, TREE PRINT, TREE EDIT, TREE SAVE, TREE LOAD, and TRANSFER GEN correspond to System Specification operations while STEP, IMPULSE, BODE, NYQUIST, ROOT LOCUS, LETTER, RE-PLOT, and MULTI-PLOT correspond to System Analysis operations. MENU displays a brief description of each key, PRINT DEVICE changes printing options, and DUMP GRAPHICS prints plots on the internal thermal printer (9845B/C only).

Answering Prompts

When the program needs information from you, a message is displayed at the bottom of the CRT. This is a prompt. All prompts in this software package have certain features in common:

- Prompts appear at the bottom of the CRT and the program waits until you respond and press CONTINUE.
- If the response is not proper for the prompt, the answer is ignored and the prompt is repeated.
- Certain prompts may contain parenthesized default values. If this value is acceptable, you simply press CONTINUE and the program uses the default value as the answer to the prompt.
- The prompt may contain a parenthesized list of possible responses. You must respond by typing in one of the responses from the list.
- The plotting programs display a table showing default values for the plotting parameters. If the approiate default value is acceptable, respond to the corresponding prompt by simply pressing CONTINUE.

Start Up Procedure

To begin operation of the program, the program and special function key definitions must be loaded from the program tape cartridge. Then the program is run. To do this:

Type MASS STORAGE IS [msus] (specify the mass storage device you are using) and press EXECUTE.

Load and run the program with LOAD "AUTOST",1 and press EXECUTE. The Main Menu appears.

System Specification Operations

Entering the Transfer Function or Block Diagram

Both simple transfer functions and block diagrams are input by entering an appropriate binary tree as described in Chapter 1. A simple transfer function is entered as a binary tree with only one node; the root node. A prompting sequence has been specifically designed to facilitate entering the binary tree. A general procedure is:

1. Use the rules given in Chapter 1 to draw a block diagram.
2. Write each transfer function as a ratio of two polynomials: a numerator polynomial and a denominator polynomial.
3. Use the procedure given in Chapter 1 to generate a binary tree representation of the block diagram.
4. Press TREE BUILD and answer all the prompts displayed.

Calculating the Overall Transfer Function

In order to use any of the plotting programs, an overall system transfer function must first be generated. However, if the entered program is just a simple transfer function so that corresponding binary tree has only one node, this step can be skipped. To calculate and display the overall transfer function, press TRANSFER GEN.

Displaying Tree Information

All information pertaining to an entered binary tree can be displayed in tabular form at any time. To do this press TREE PRINT and answer the prompts.

Modifying a Tree

Binary tree node attributes such as Name and Type can be changed. Nodes can be added or deleted, and transfer functions associated with simple nodes can be modified by using the tree editing subprogram. To do this press TREE EDIT and answer the prompt.

When modifying a transfer function, the programs ask for the degree and the coefficients of the corresponding polynomials. If any of these values remains unchanged, when asked for that value, press CONTINUE.

Saving and Retrieving Tree Information

All information about a binary tree can be saved on a mass storage medium and retrieved for later use. To save a tree, press TREE SAVE and answer the prompts.

To retrieve a tree press TREE LOAD and answer the prompts.

NOTE

Each file uses fourteen 1 000 byte records or 14 000 bytes.

Displaying Information on CRT or Thermal Printer

Output from all System Specification operations and the tables of default parameter values for the System Analysis programs are printed on the CRT by default. To get this information printed on the thermal printer or to revert back to printing on the CRT press PRINT DEVICE and answer the prompt.

System Analysis Operations

Generating and Displaying Plot Data

To generate Step, Impulse, Bode, Nyquist, or Root Locus data, press the appropriate key and answer the prompts.

Graphical or Tabular Output

The programs enable plots to be generated, or calculated data to be tabulated on the thermal printer. The choice is made by answering a prompt.

CRT or Pen Plotter

Plots can be displayed on the CRT or on an optional HP 9872 Graphics Plotter. The choice is made by answering a prompt.

Automatic or Self-Scaling[1]

Plot axes can be automatically scaled by the program or you can enter limits. The choice is made by answering a prompt. If the self-scaling option is involved, then the program prompts for the minimum and maximum value of each axis.

Thermal Printer Copies of CRT Graphics (9845B/C only)

A copy of a plot displayed on the CRT can be printed on the thermal printer. To do this press DUMP GRAPHICS.

Re-plotting[2]

Once one of the plotting subprograms has calculated and displayed data (either graphically or in a table), it is possible to re-display the data without re-calculation of the data points. To do this press RE-PLOT and answer the prompts.

NOTE

Use of other key functions between the finish of a plotting prog-ram and the use of RE-PLOT results in an error.

---

1 Automatic scaling is not available for Root Locus plots.

2 For Root Locus data, the re-plot key causes re-calculation of the data.

Using the Cursor to Digitize Plotted Data

Each plotting subprogram contains an option for using the cursor to locate and read data values from a plot. (Plots on the 9845B/C can be read from the CRT.) After a plot has been displayed, the option can be invoked by answering a prompt. The cursor routines are used in the following manner:

1. The cursor is moved by pressing the "ARROW" keys on either the CRT or the 9872 Plotter. On the CRT, pressing the "SHIFT" key enables finer movements to be made. On the 9872 Plotter the speed of the cursor movement is controlled by the FAST key.
2. A point is read (digitized) by pressing the CONTINUE key on the CRT or the ENTER key of the 9872 Plotter.
3. The cursor routine is terminated by pressing the CONTINUE key twice on the CRT or the ENTER key twice on the 9872 Plotter.

A table of the digitized data is displayed on the thermal printer.

Multi-plots

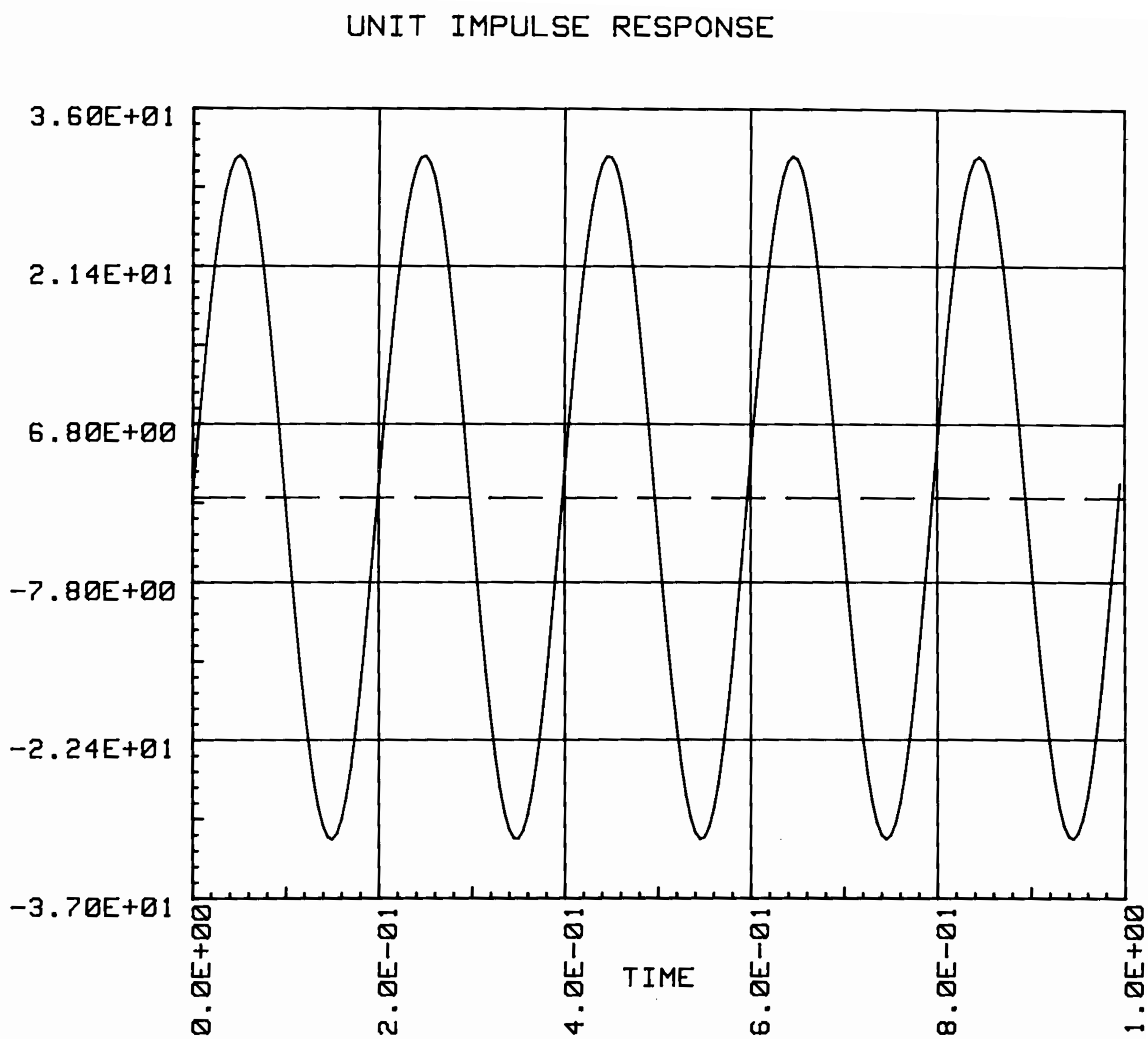
After a plot has been displayed, it is possible to change the overall transfer function and superimpose a plot of the new transfer function's data on the previous graph. To do this, when the first plot is finished:

1. Use the tree editing subprogram (TREE EDIT) to modify the tree.
2. Calculate the new overall transfer function (TRANSFER GEN).
3. Press ~~RE-PLOT~~ MULTI-PLOT and answer the prompts.

Lettering

Once a plot has been displayed, it is possible to add additional titles and labels to a plot manually by using a lettering routine. Lettering can be inserted in either the horizontal or vertical direction. The general procedure follows: After a plot has been displayed:

1. Press LETTER and answer the prompts.
2. Move the cursor to the desired location by pressing the "ARROW" keys as described in the section Using the Cursor to Digitize Plotted Data.
3. Type in a label using the desktop keyboard.
4. Press CONTINUE to terminate.



Correct Impulse Response of Harmonic Oscillator

Chapter 3
Examples

This chapter presents three examples that use the Linear Systems Analysis Package. In the first example, a transfer function is directly read into the program and various plotting routines are generated. The second example presents a block diagram involving feedback, parallel, and cascade blocks and illustrates how this diagram can be entered in binary tree form. The resulting overall transfer function is then computed and response plots are generated. A servo-mechanism with tachometer control is presented as a final example for which this program is used to specify design gain parameters.

Example 1: Second-Order Transfer Function

Performance of a system is normally analyzed by applying test inputs and observing the corresponding outputs. Design objectives may include small steady state error between test input and generated response. To attain such system objectives, many systems are designed so that their performance closely approximates that of a second-order transfer function of the form.

$$T(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

where ω_n is the natural frequency of oscillation and ξ is the damping coefficient. The response of T(s) to various test inputs can be determined analytically, however, here we analyze T(s) using this package (in particular, for $\xi_n = 0.25$, 0.5, and 1.0).

Preliminaries

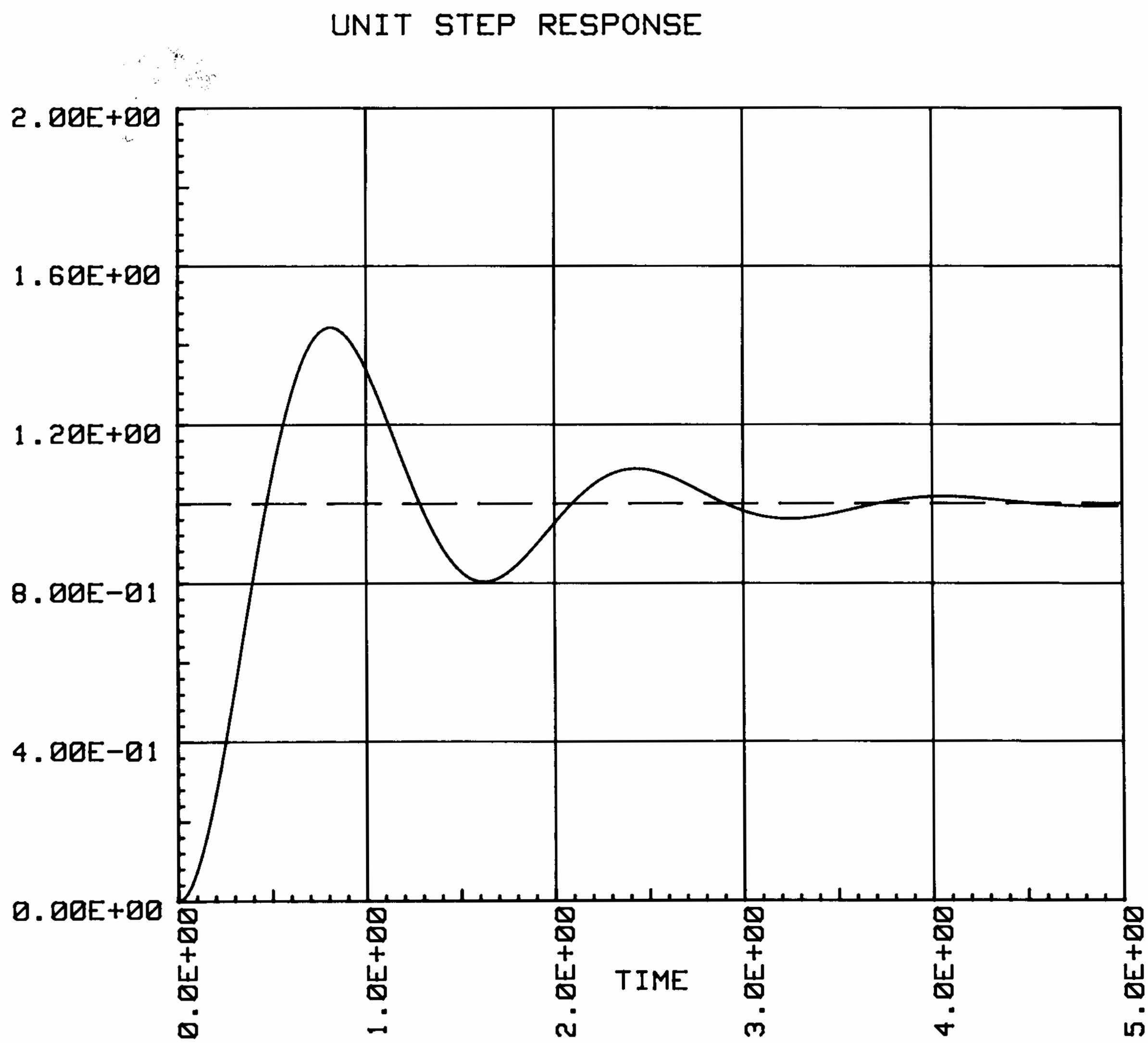
In order to generate the desired plots of T(s), the coefficients of the transfer function are read in. Pressing TREE BUILD generates the prompts for building a binary tree. Since only one transfer function block needs to be input, the binary tree root node contains the overall transfer function. Initially, let $\xi = 0.25$, i.e.,

$$T(s) = \frac{16}{s^2 + 2s + 16}$$

By pressing the key functions, plots can then be constructed for analysis purposes.

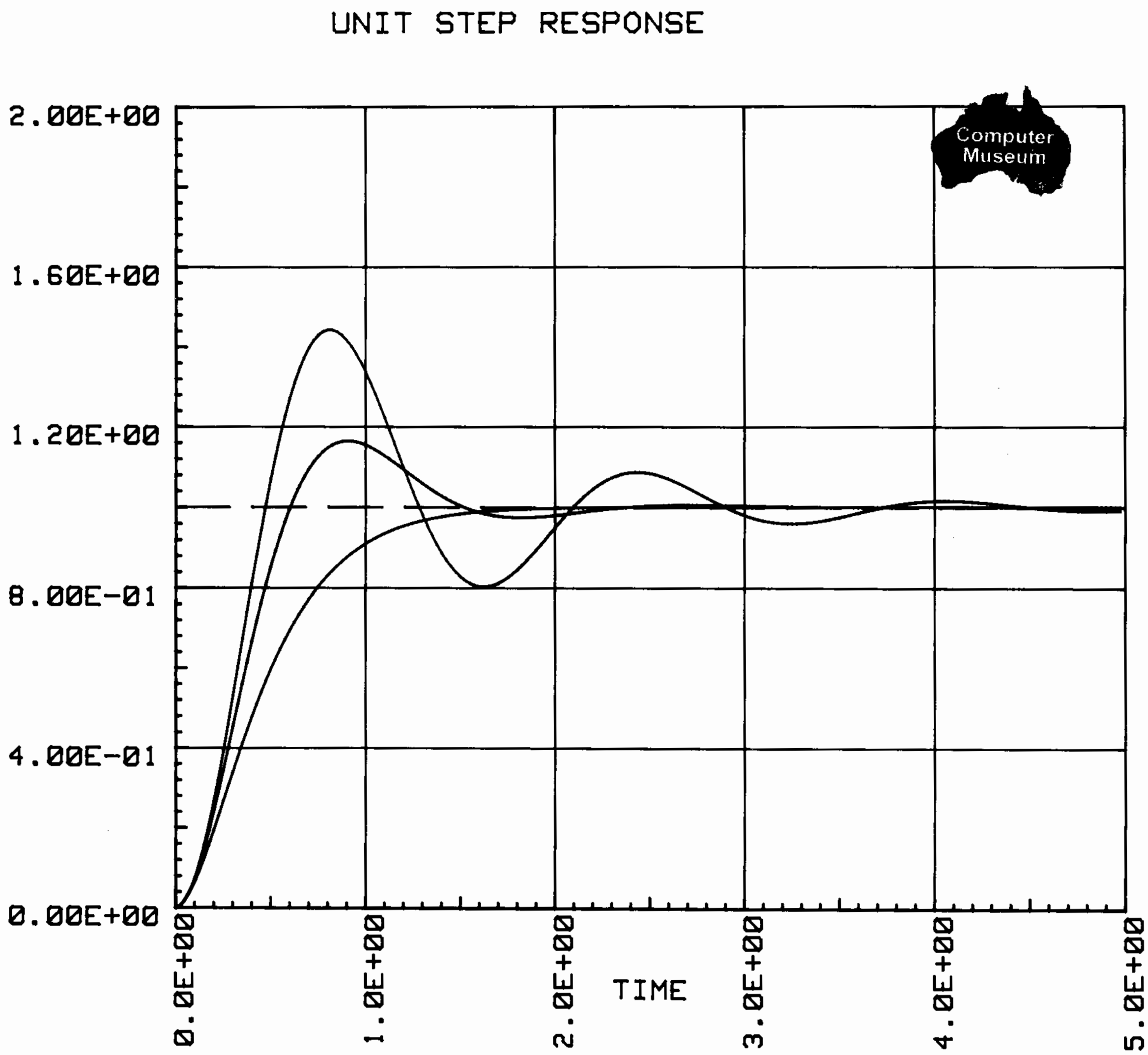
Unit Step Response

Pressing STEP generates prompts for the unit step response. For this example, the default values are used to generate the response plot, with the self-scaling plot option for the amplitude response interval. That is, the routine uses a time interval of zero to five units and amplitude response interval of zero to two units to plot the unit step response. This results in the response plot given below:



Unit Step Response of a Second-Order System

In order to investigate the effects of the damping coefficient on system response, the multi-plot feature of this program is invoked. To do this, the transfer function T(s) must be modified for $\xi = 0.5$. Pressing TREE EDIT enables this modificAtion of T(s). By pressing MULTI-PLOT, the unit step response for $\xi = 0.5$ is generated. A similar procedure for $\xi = 1.0$ results in a multiple unit step response plot as shown below:

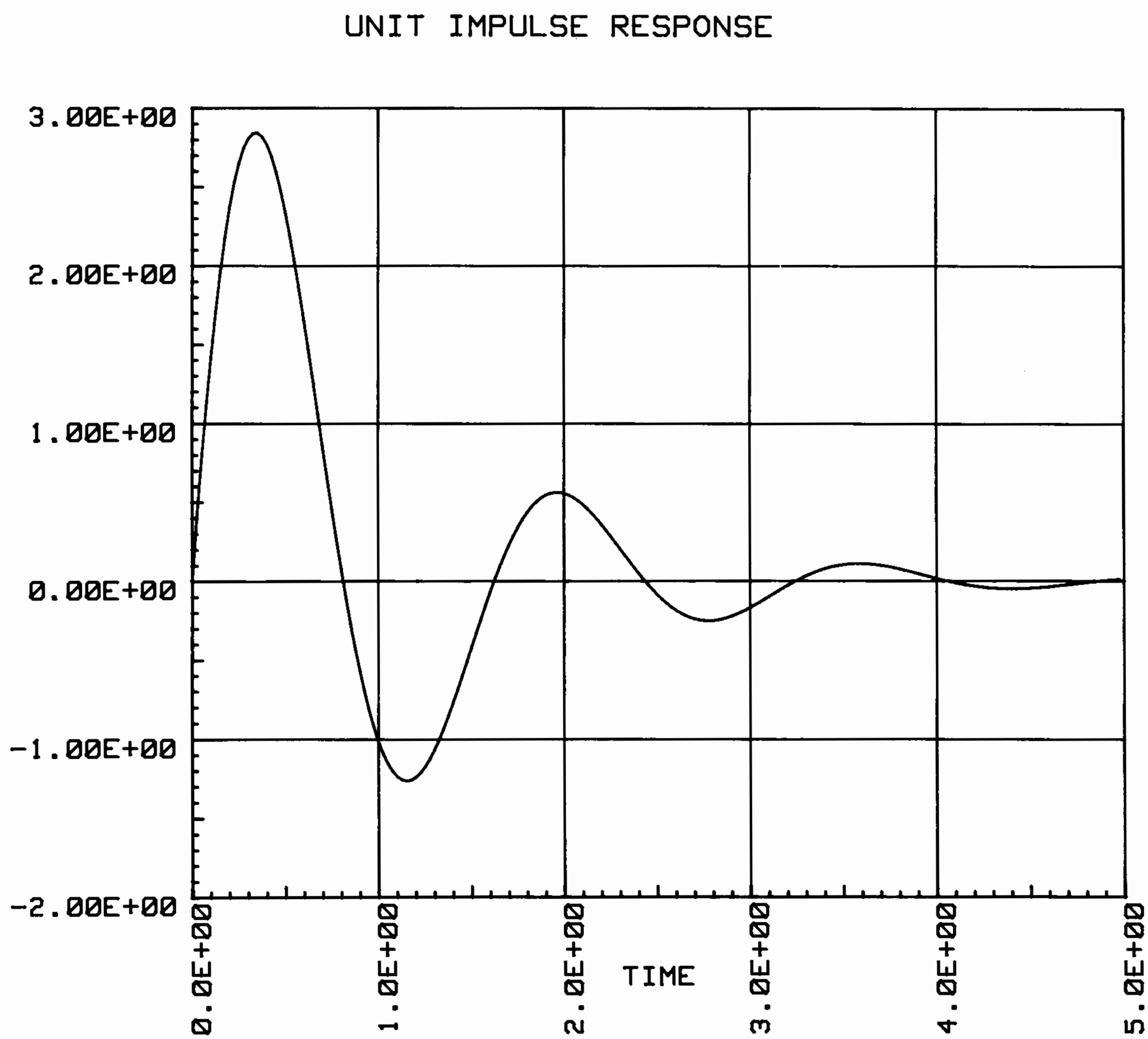


Unit Step Response for Various Damping Coefficients of a Second-Order System

From this plot, note that increasing the damping coefficient decreases the steady state oscillations at the expense of slower rise time.

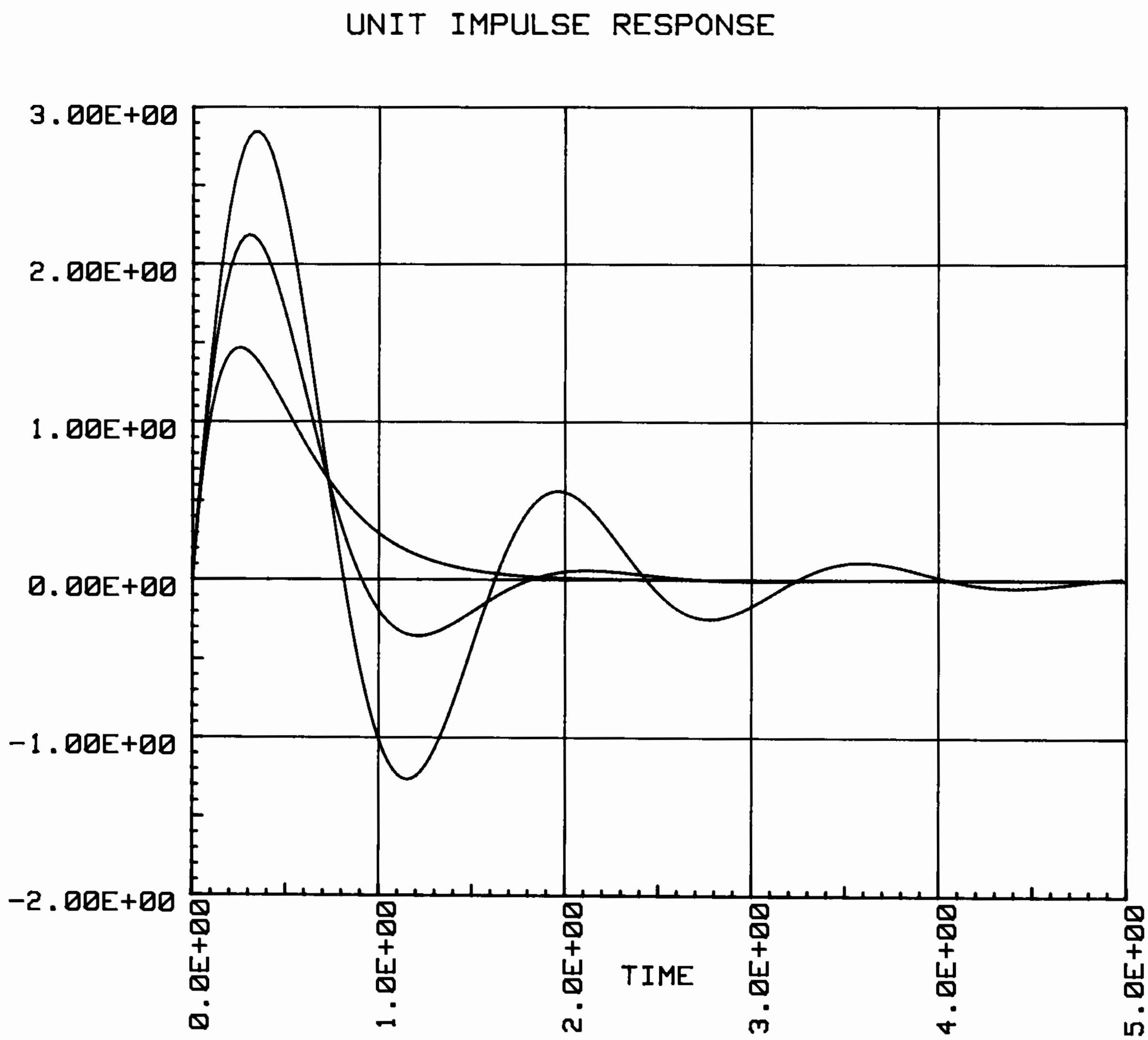
Unit Impulse Response

Pressing IMPULSE enables the prompts for the unit impulse response. Below is the response using the default values, with the self-scaling plot option for the amplitude response interval, for $\xi = 0.25$.



Unit Impulse Response of a Second-Order System

Following in a similar fashion to that of the unit step response, multi-plots of the unit impulse response for the various damping coefficients can be generated. This is shown below:

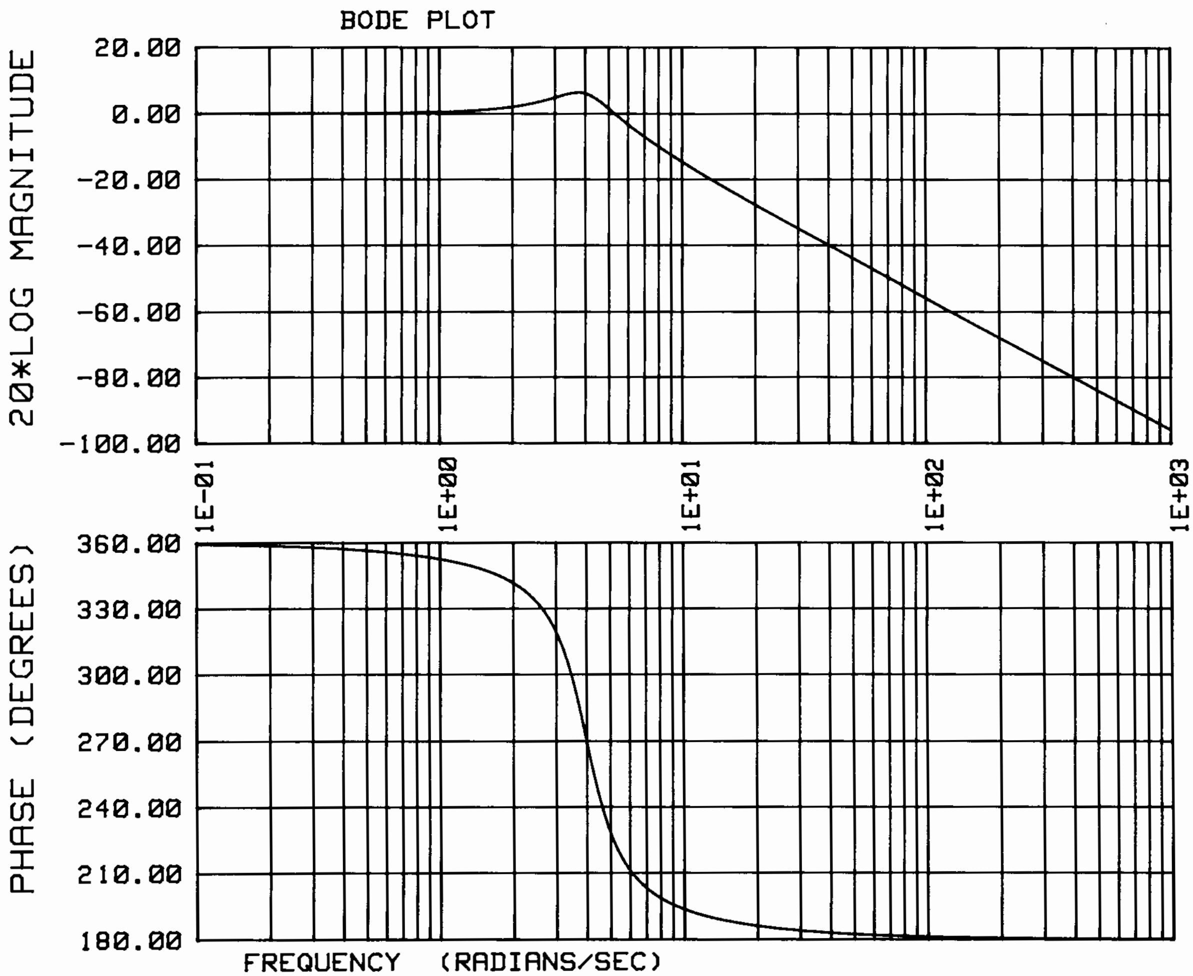


Unit Impulse Response for Various Damping Coefficients of a Second-Order System

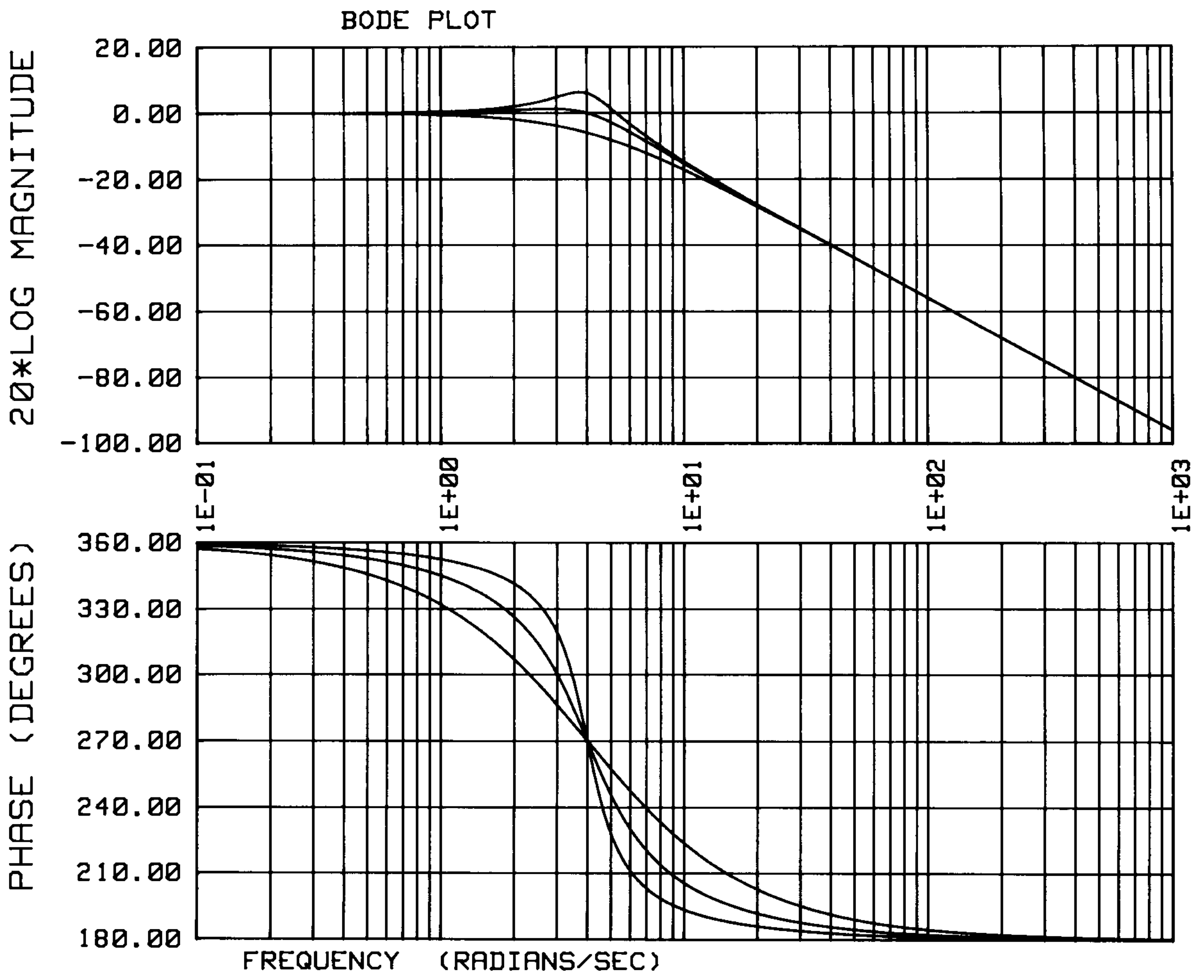
As with the unit step response, note that increasing the damping coefficient decreases the steady state oscillation.

Frequency Plots

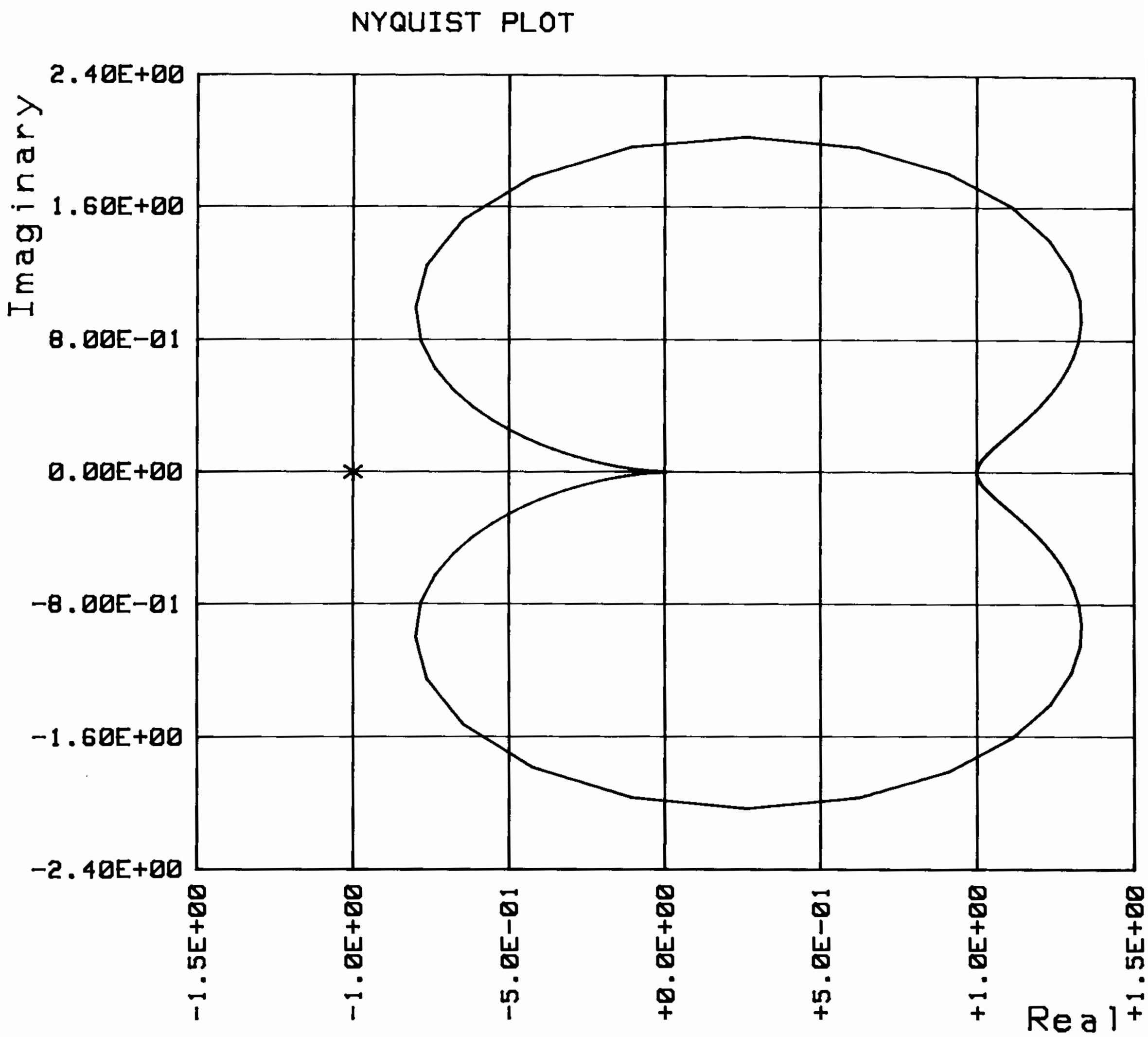
By pressing BODE, NYQUIST, and ROOT LOCUS, along with the MULTI-PLOT, the various frequency plots as a function of damping coefficient can be generated. These are presented below for $\xi = 0.25$ as well as the multiplots for $\xi = 0.25$, 0.5, and 1.0:



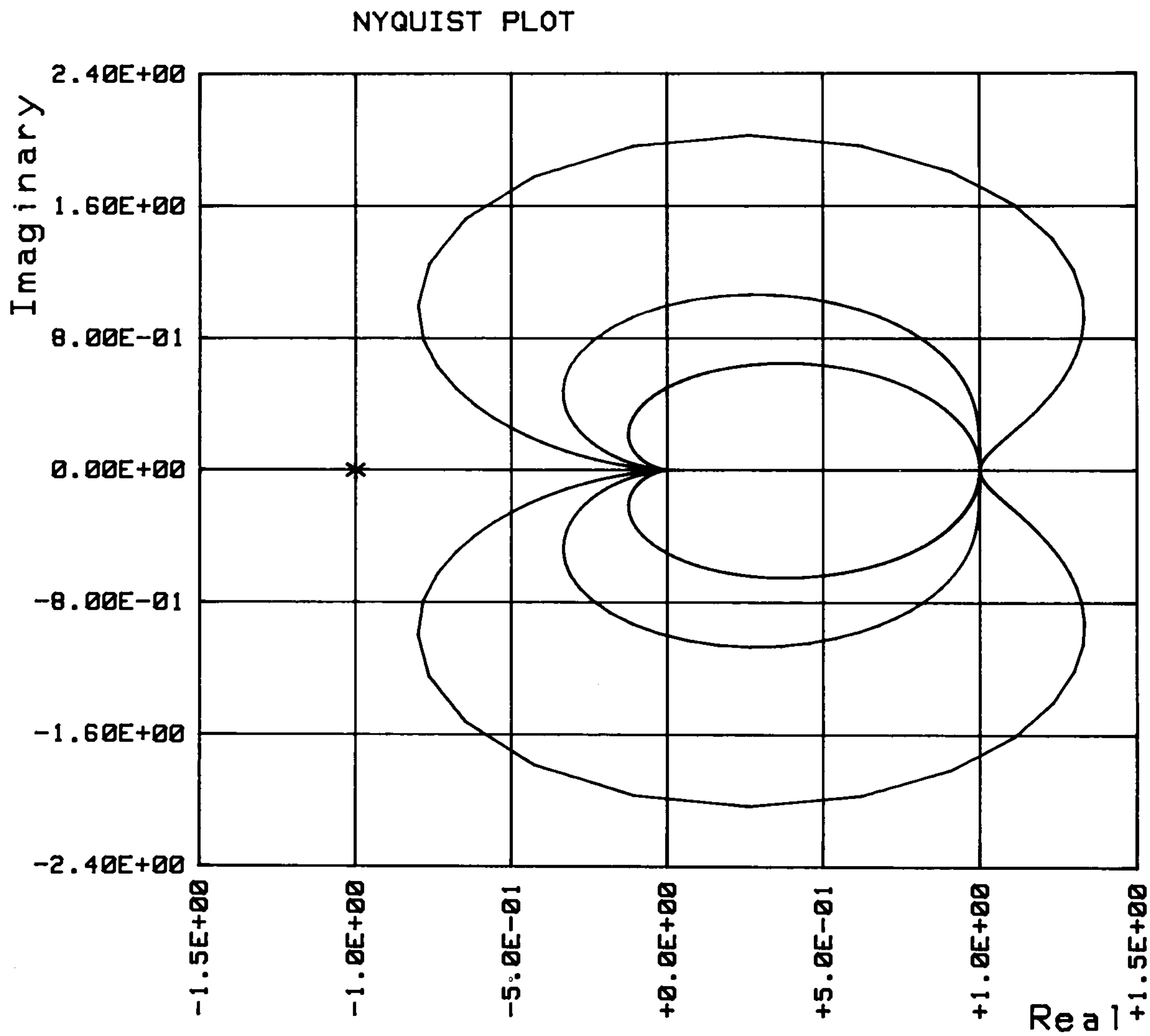
Bode Plot of a Second-Order System



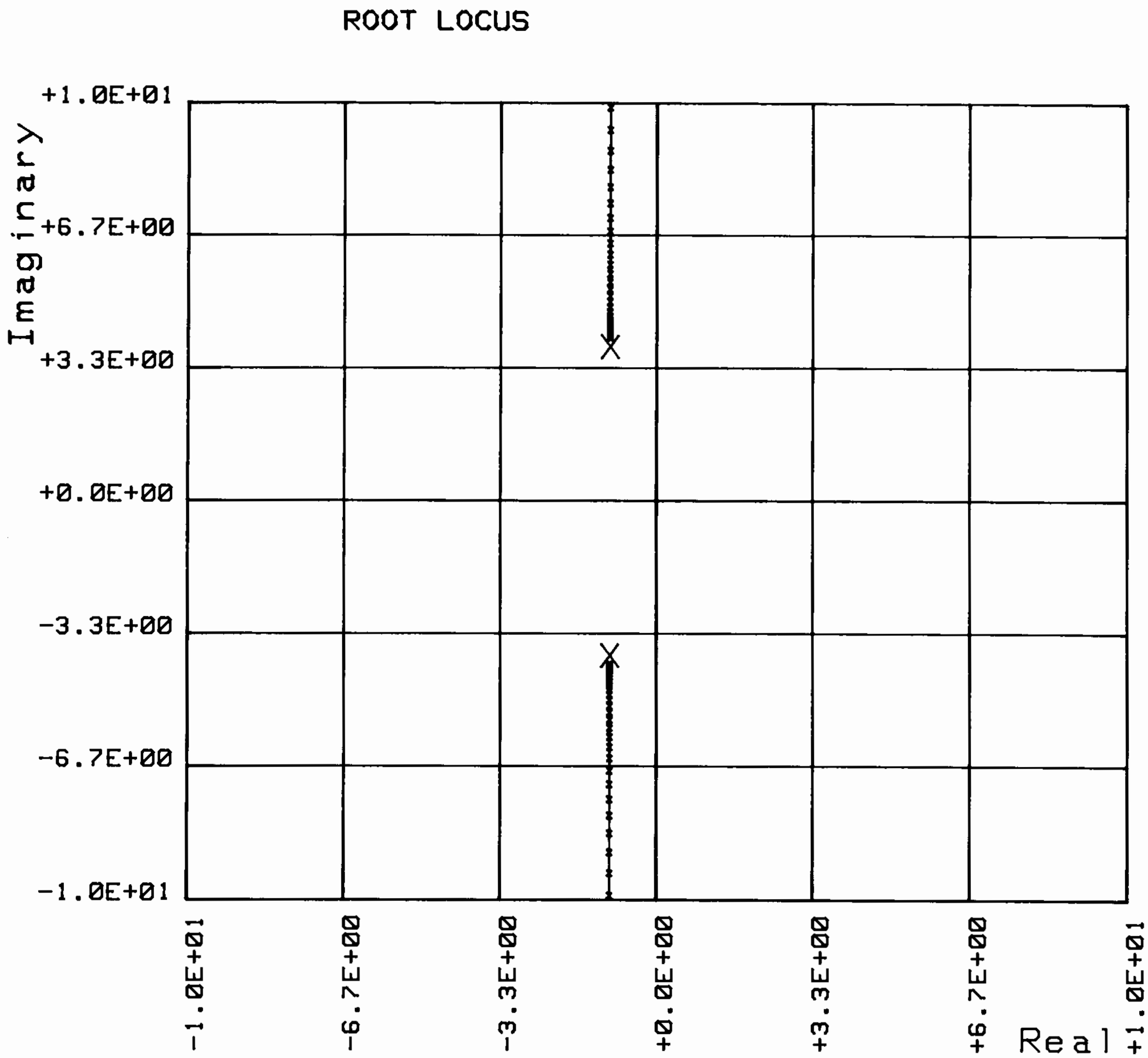
Bode Plot for Various Damping Coefficients of a Second-Order System



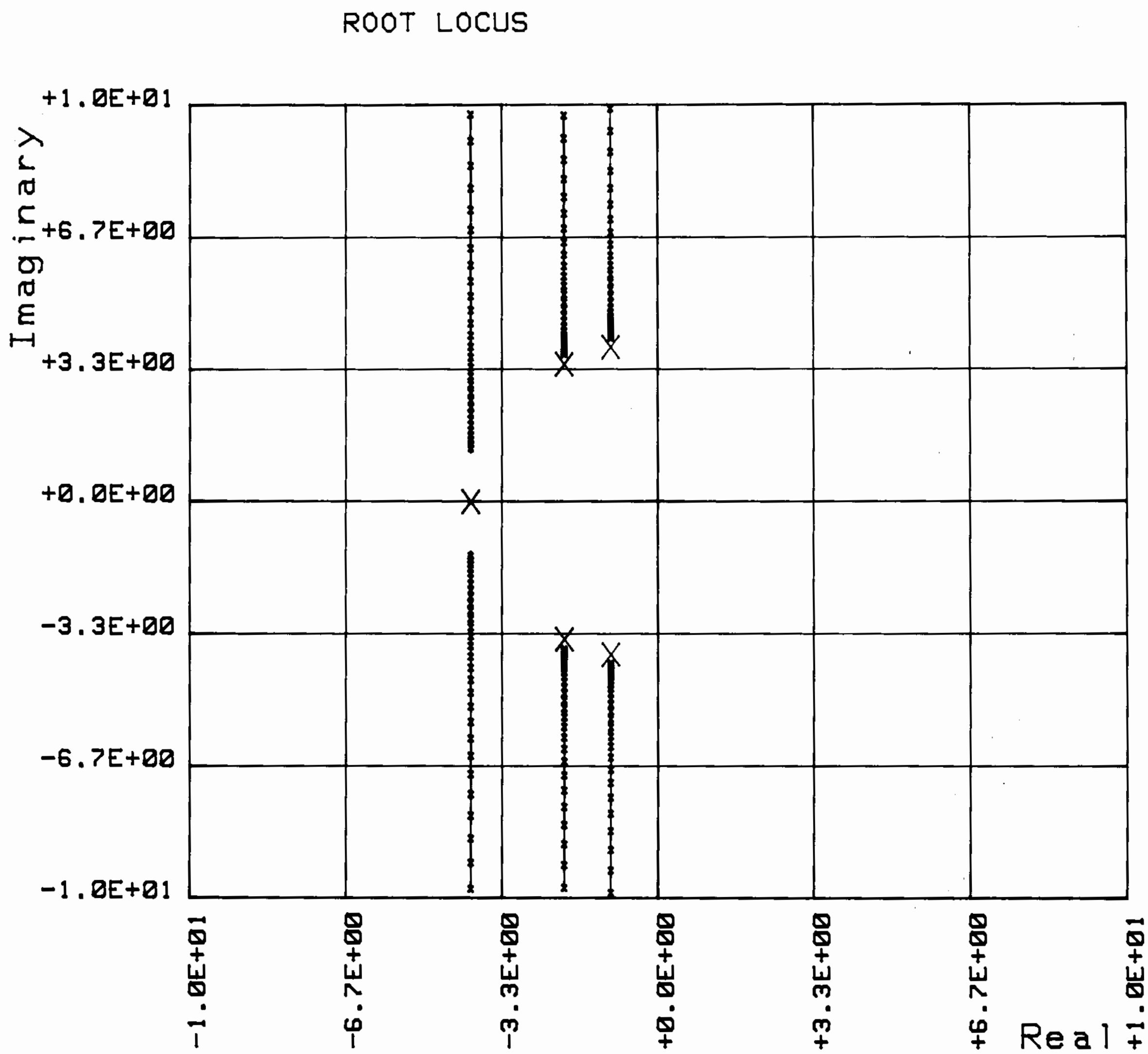
Nyquist Plot of a Second-Order System



Nyquist Plot for Various Damping Coefficients of a Second-Order System



Root Locus of a Second-Order System

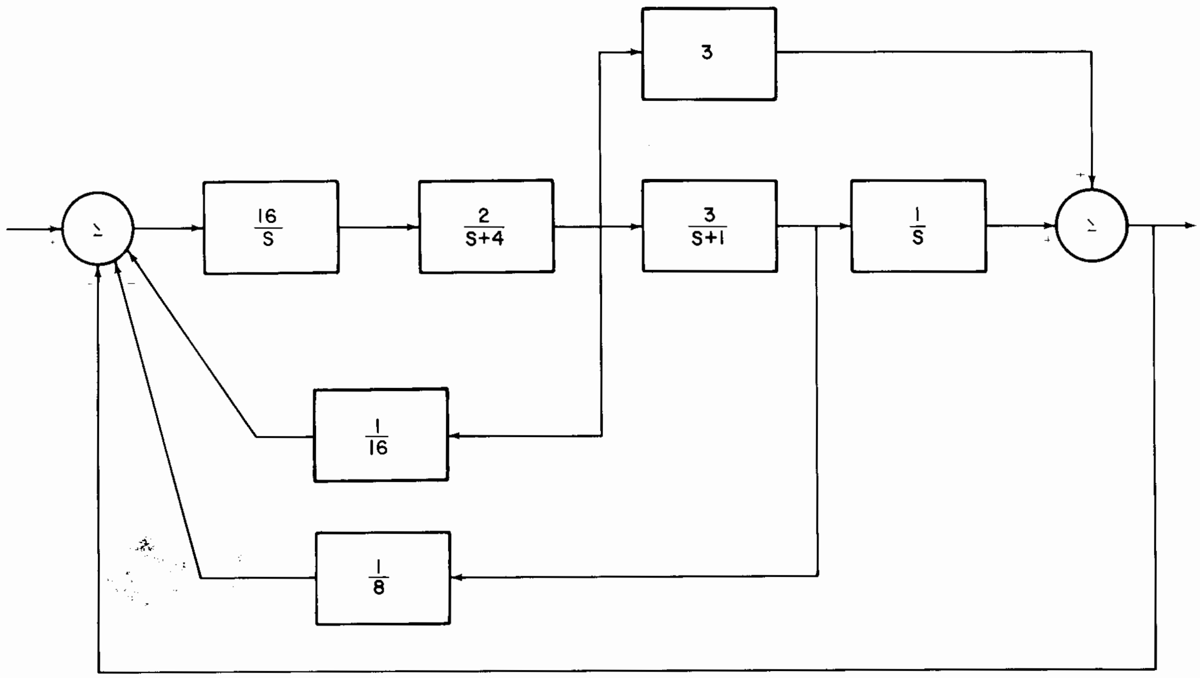


Root Locus for Various Damping Coefficients of a Second-Order System

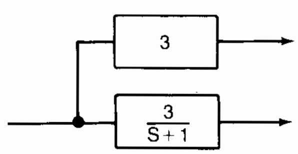
This example illustrates the use of the multi-plot option in conjunction with the time and frequency plot key functions when a transfer function is directly read in. Cursor options are available if you desire particular values of a plot.

Example 2: Block Diagram-to-Tree Construction

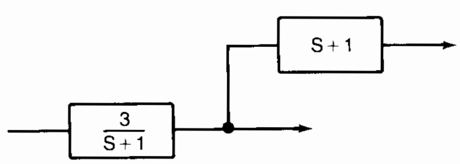
This example illustrates the procedure necessary to transform a complex block diagram into binary tree representation in order to generate the overall transfer function and corresponding plots. Consider the following block diagram:



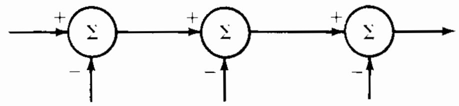
The first step is to redefine the block diagram into a diagram satisfying the rules given in Chapter 1. In particular, the blocks of



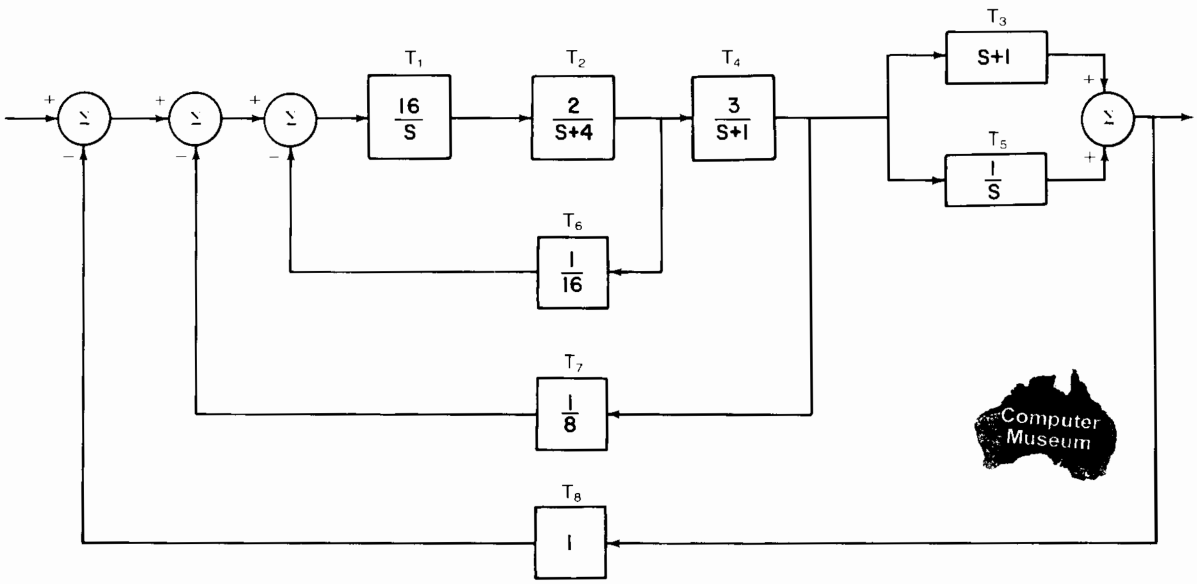
must be redrawn, using Rule 2 discussed in Chapter 1 under Rules for Block Diagram Construction as:



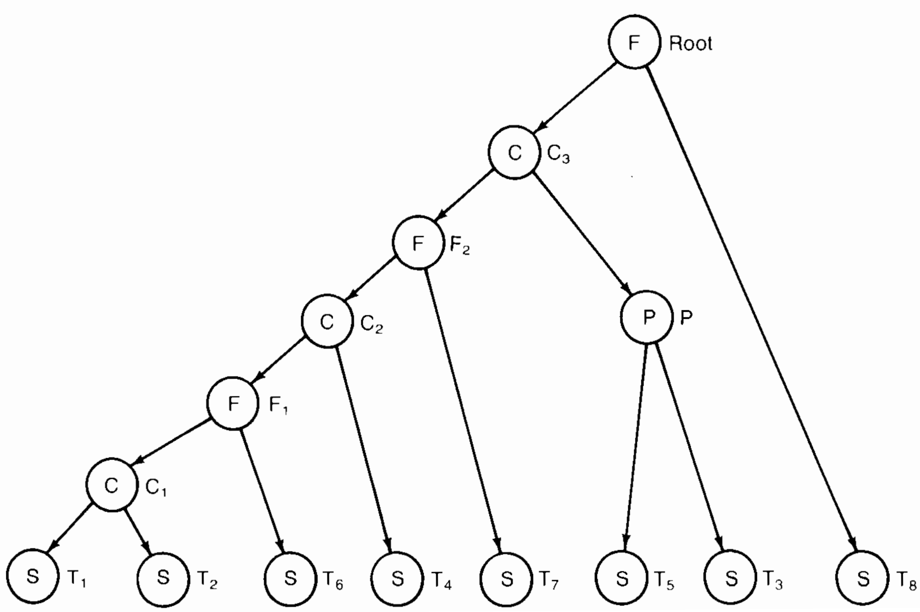
Furthermore, the input summer must be redrawn so that only two signals enter the junction and one signal departs. This results in:



The block diagram, therefore, can be represented by an equivalent structure:



This equivalent structure consists of the basic simple, cascade, parallel, and feedback blocks necessary for the binary tree representation. The binary tree can thus be readily constructed as:



In tabular form, the node information of this binary tree is given in the following figure:

NODE INFORMATION TABLE

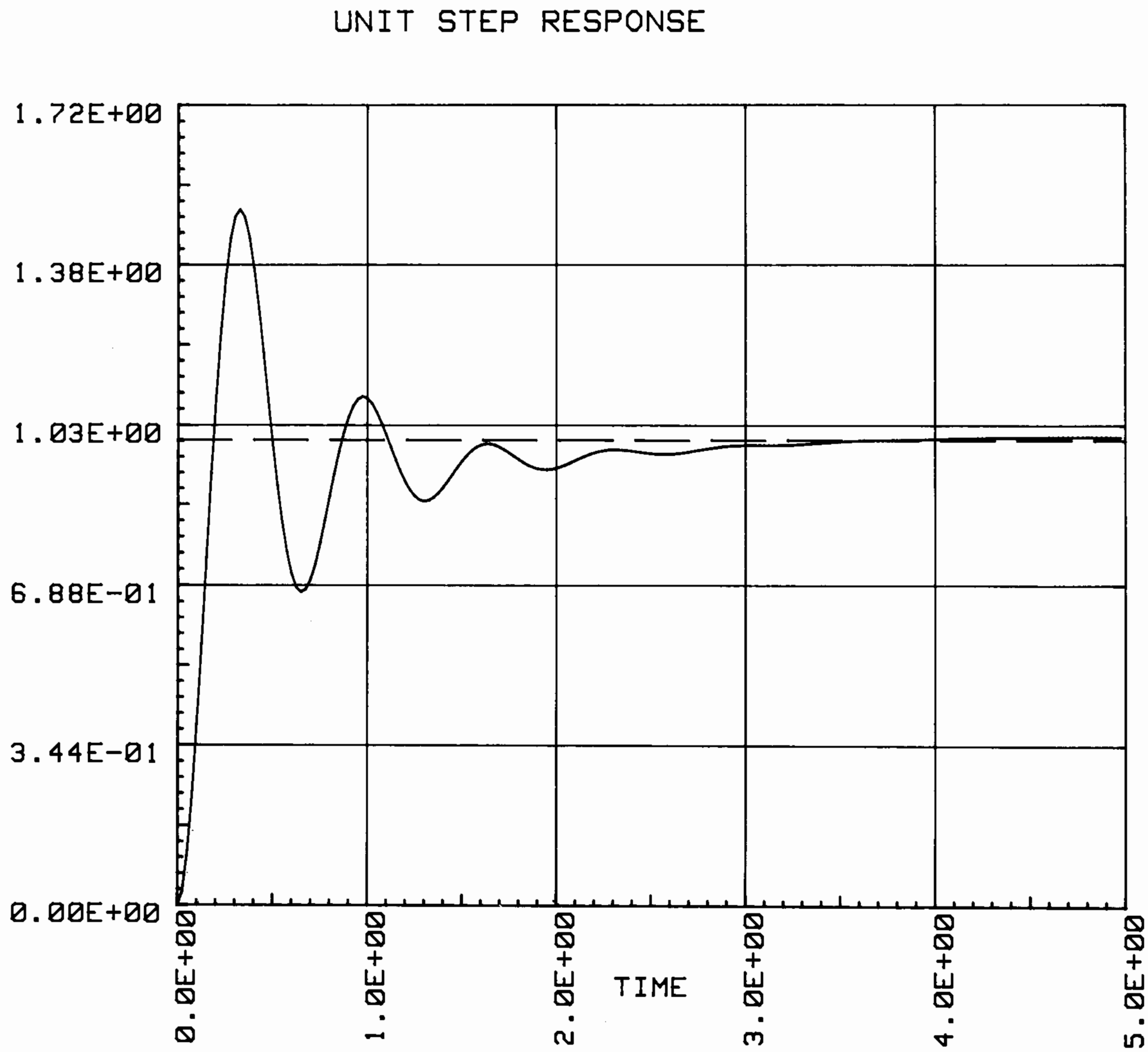
NUMBER	NAME	LEVEL	TYPE
0	ROOT	0	F
1	C3	1	C
2	F2	2	F
3	C2	3	C
4	F1	4	F
5	C1	5	C
6	T1	6	S
7	T2	6	S
8	T6	5	S
9	T4	4	S
10	T7	3	S
11	P	2	P
12	T5	3	S
13	T3	3	S
14	T8	1	S

By pressing TREE BUILD you can enter the appropriate nodes corresponding to the block diagram. To find the overall transfer function, press TRANSFER GEN to obtain:

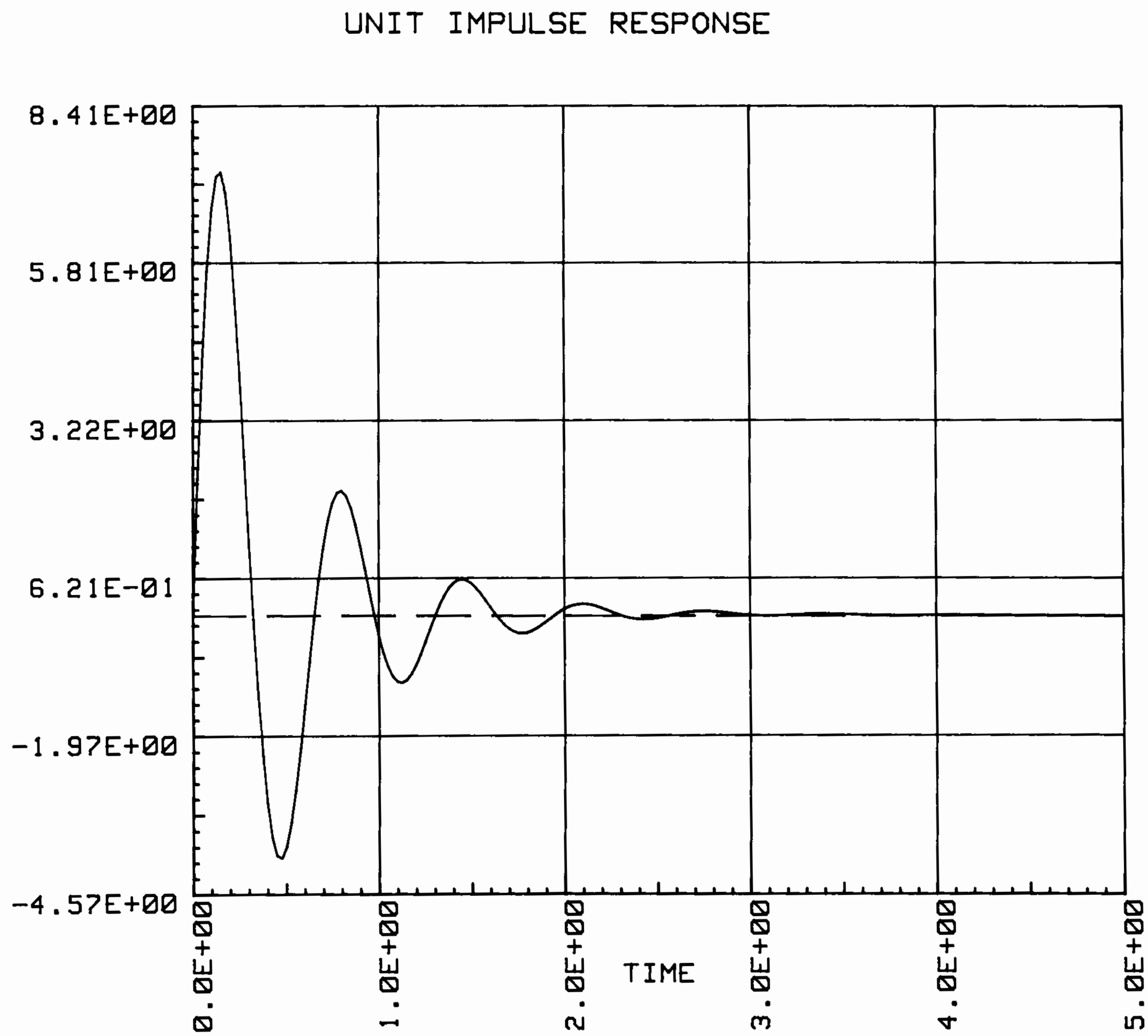
THE OVERALL BLOCK DIAGRAM TRANSFER FUNCTION

DEGREE	NUMERATOR	DENOMINATOR
0	1.2288000E+04	1.2288000E+04
1	1.2288000E+04	1.4080000E+04
2	1.2288000E+04	1.3056000E+04
3		6.4000000E+02
4		1.2800000E+02

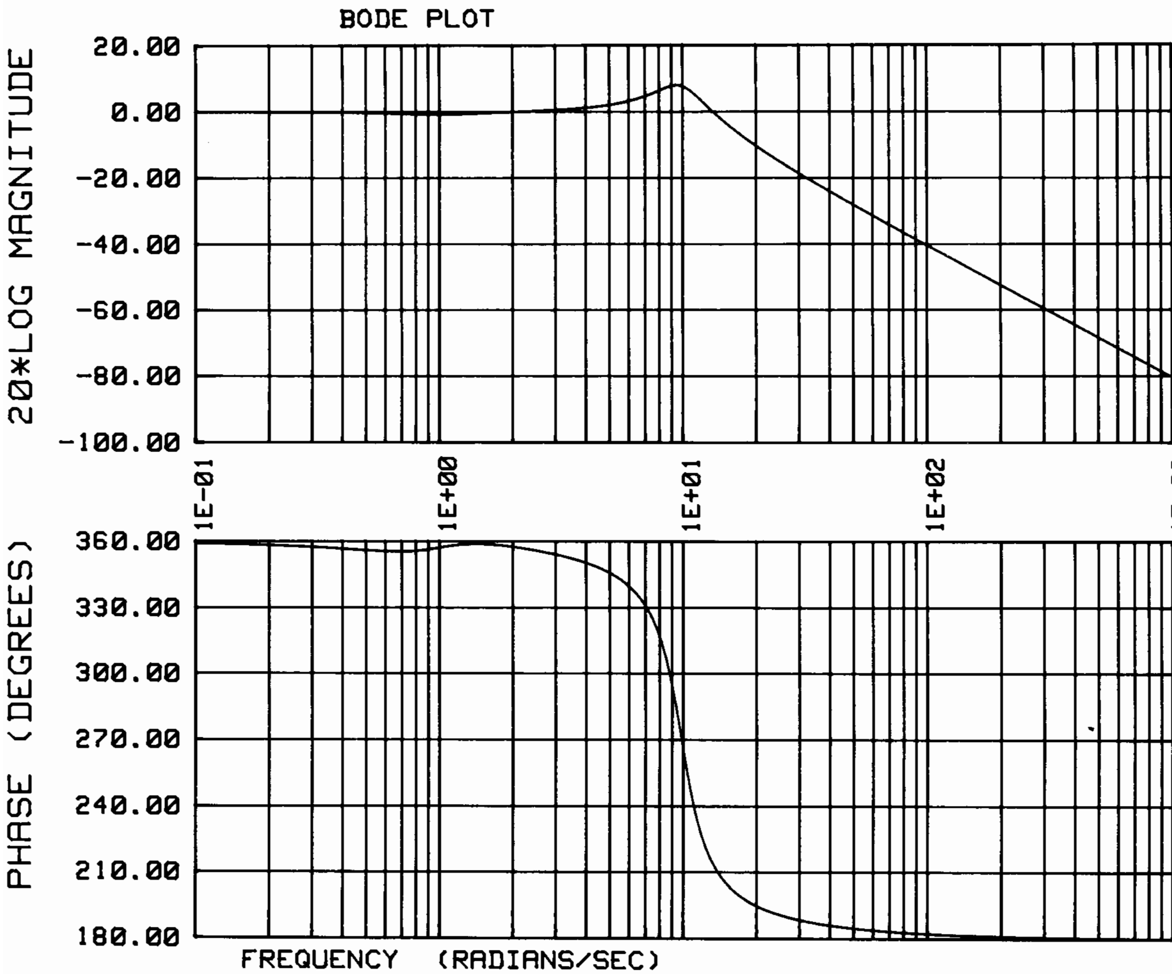
With the overall transfer function generated, you may find the unit step response, unit impulse response, Bode Plot, Nyquist Plot, and Root Locus by pressing the appropriate special function key. The following are the plotting results:



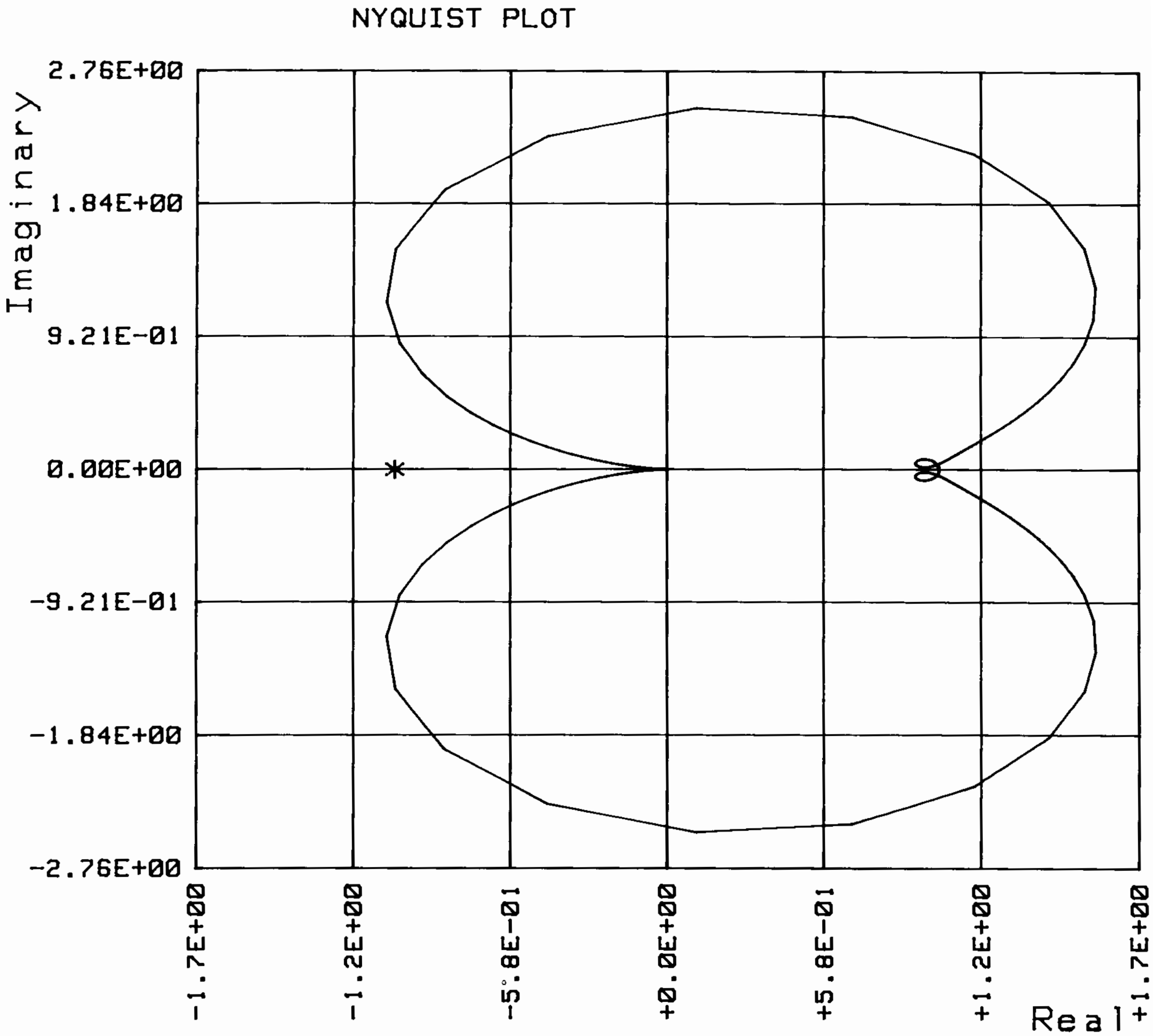
Unit Step Response of Example 2



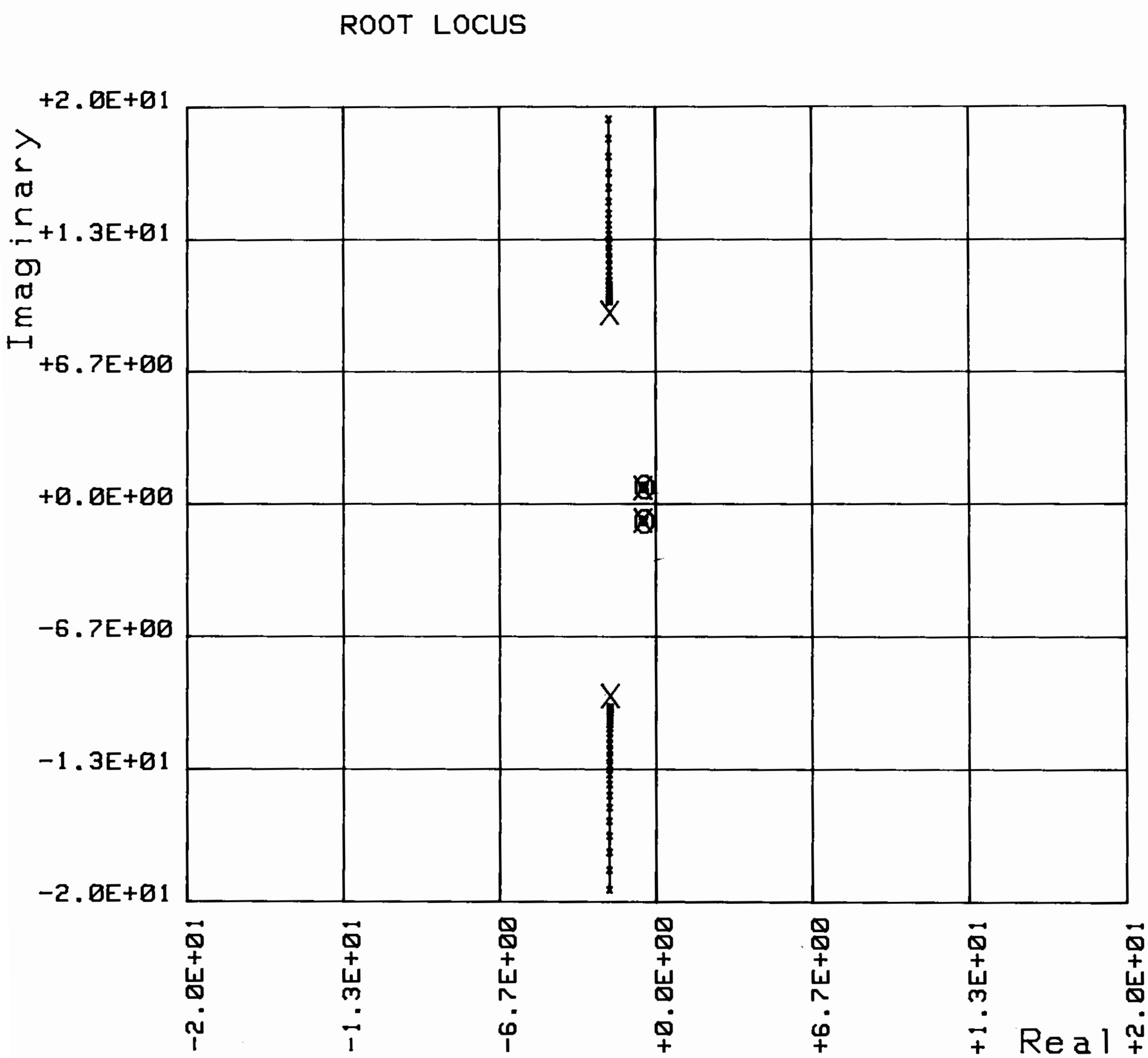
Unit Impulse Response of Example 2



Bode Plot of Example 2



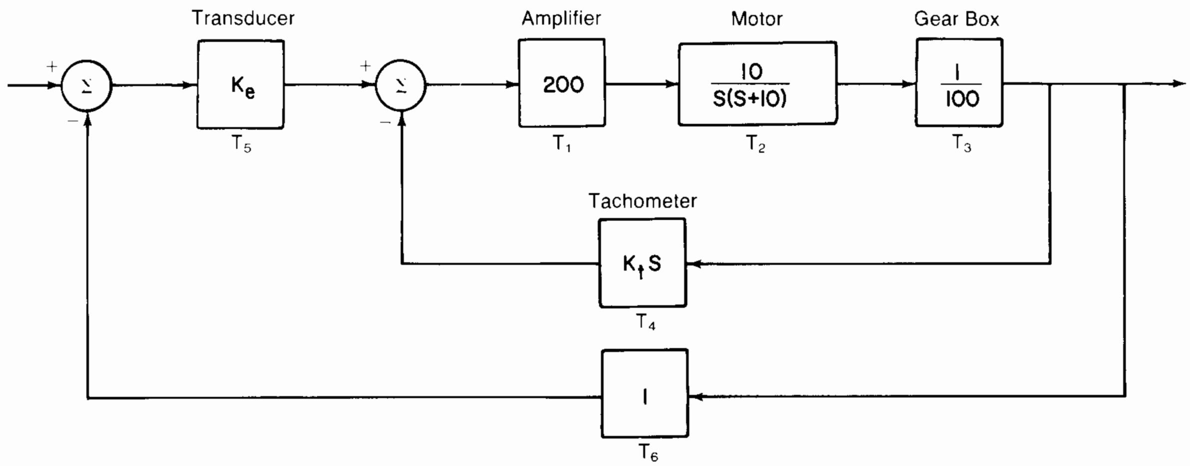
Nyquist Plot of Example 2



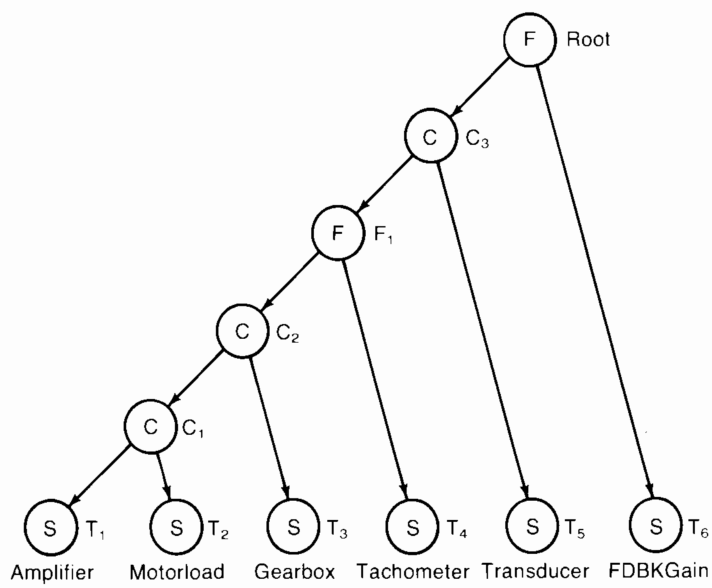
Root Locus of Example 2

Example 3: Tachometer Control

As a final example, consider the following block diagram representation of a servomechanism system. The goal is to choose the gain parameters K_e (error gain) and K_t (tachometer gain) to attain quick rise time and no overshoot in the unit step response.



Using the procedure discussed above, you may construct a binary tree representation of the block diagram as follows:

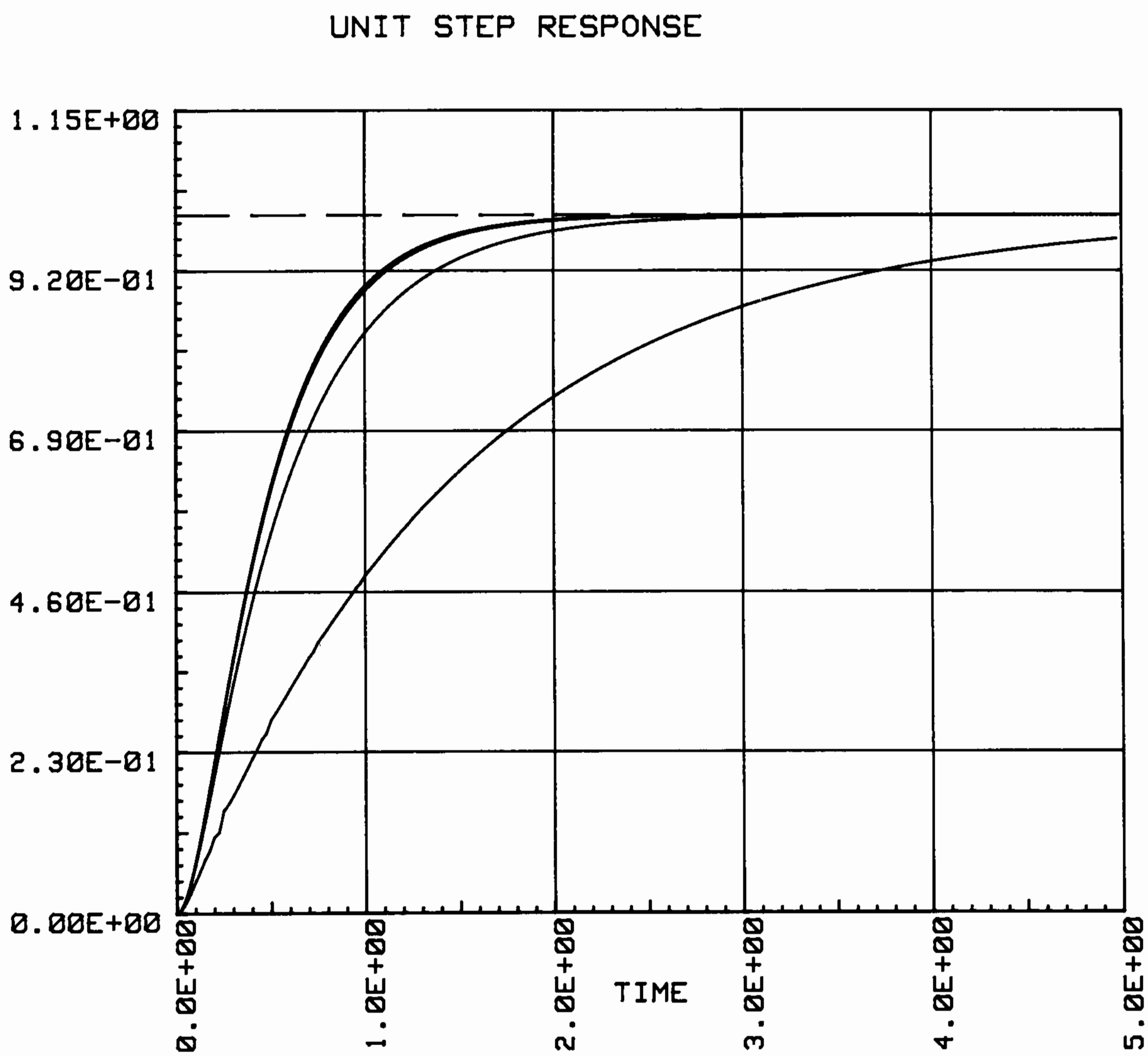


The corresponding node information table is shown:

NODE INFORMATION TABLE

NUMBER	NAME	LEVEL	TYPE
0	ROOT	0	F
1	C3	1	C
2	F1	2	F
3	C2	3	C
4	C1	4	C
5	AMPLIFIER	5	S
6	MOTORLOAD	5	S
7	GEARBOX	4	S
8	TACHOMETER	3	S
9	TRANSDUCER	2	S
10	FDBCKGAIN	1	S

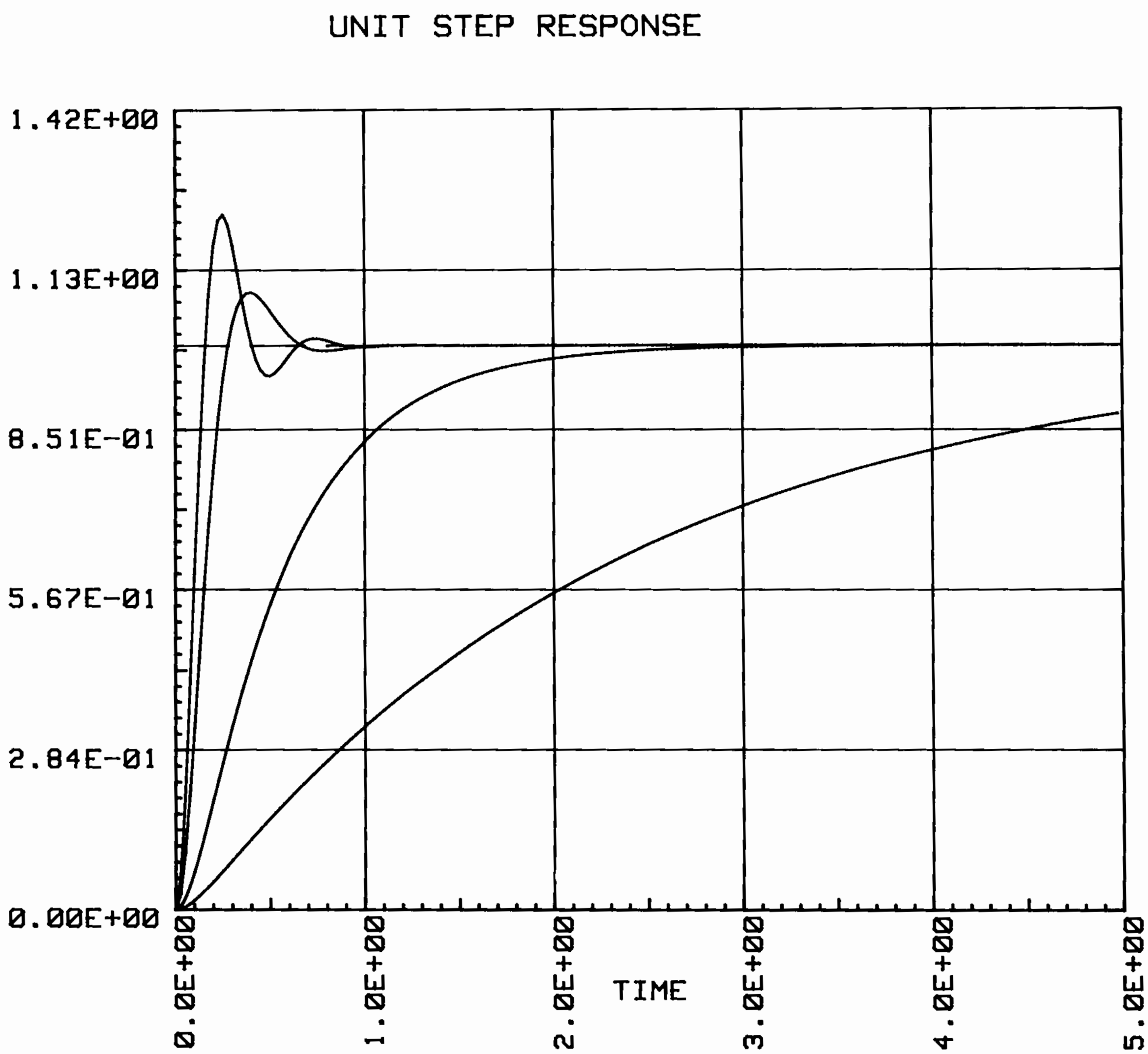
Since there are two parameters to choose from, you may hold one of the parameters constant and observe the influence of variation of the other parameter on the unit step response. Suppose you temporarily hold the gain $K_e = 1$ and let K_t vary. In particular, suppose you let $K_t = 0.001, 0.01, 0.1$, and 1.0. The unit step response for these values of K_t are given below:



Multiplot of Unit Step Response for Various Tachometer Gains

As K_t decreases, the rise time decreases and the system reaches steady state faster. So you can choose $K_t = 0.1$.

Suppose you then vary K_e = 0.25, 1., 5., and 10. The unit step response for these values of K_e are given below:

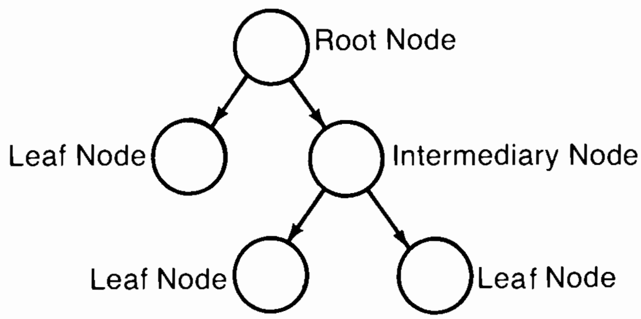


Multiplot of Unit Step Response for Various Transducer Gains

It is seen that as K_e increases, the unit step response overshoots more. Hence an adequate design would use K_t = 0.1 and K_e = 1.

Appendix A
Binary Tree Notation

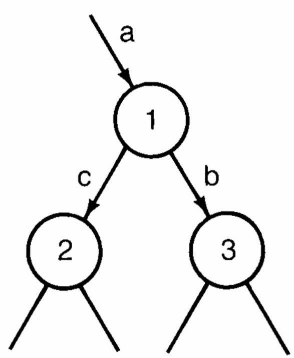
An example of a binary tree is shown below.



It consists of a number of **nodes** depicted by circles above, and a number of directed **branches** entering by arrows above. This type of binary tree has the following properties:

- There is a starting or **root node** where the tree begins. It has no branches entering in.
- There are one or more terminating or **leaf** nodes. These have only one branch entering in.
- There are possibly a set of intermediary nodes with one branch entering in and two branches leaving.

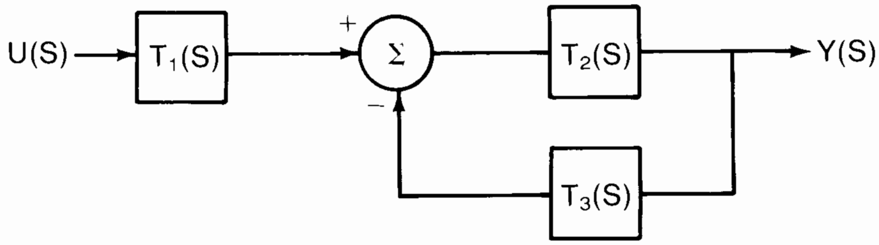
Consider the following section of a tree:



Node 1 is called the **predecessor** of nodes 2 and 3. Node 2 is called the **left successor** or **descendent** to node 1 while node 3 is called the **right successor** or **descendent** of node 1. Branch a is a branch directed **into** node 1 while branches b and c are the **left** and **right** branches out of node 1

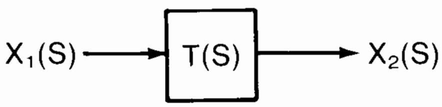
Appendix B
Block Diagrams

An example of a block diagram is shown below:



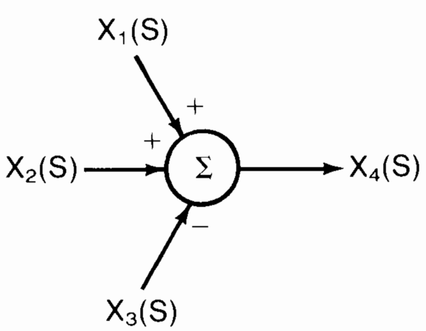
It describes the interconnection of simple subsystems. For a linear system, a block diagram can be composed of the following:

Simple Transfer Function Blocks



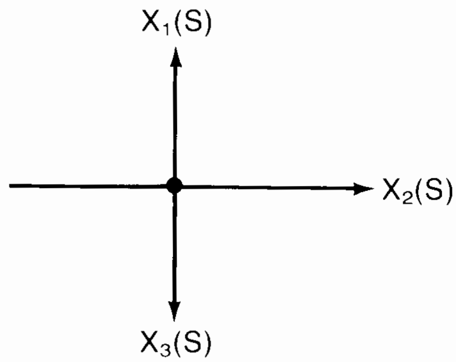
$$X_2(S) = T(S)X_1(S)$$

Summation Points



$$X_4 = X_1(S) + X_2(S) - X_3(S)$$

Branch Points



$$X_1(S) = X_2(S) = X_3(S)$$

Appendix C
HP Part Numbers

Linear Systems Analysis Package for the HP 9845B/C (09845-15190)

- Tape Cartridge 09845-15194
- User's Manual 09845-15191
- Key Overlay 7121-0707

Linear Systems Analysis Package for the HP 9835 (09835-15190)

- Tape Cartridge 09835-15194
- User's Manual 09835-15191
- Key Overlay 7120-0706

HEWLETT PACKARD

Part No. 09845-15191
Rev. B

Printed in U.S.A.
November 2, 1980