Version 9.0

VisSim/Analyze User's Guide

By Altair Engineering, Inc.

Altair Engineering, Inc.
Altair Engineering User's Guides for VisSim Products - Version 9.0

Copyright

Altair Engineering, Inc.
487 Groton Road
Westford, MA 01886

Trademarks

Copy and use restrictions

Acknowledgments

Contents

Introduction

This section contains…

What is VisSim/Analyze

VisSim/Analyze v9 is an add-on module that extends VisSim's functionality by providing the capability to generate a linear approximation of a nonlinear system at a specified operating point. The region in which the linear approximation retains its validity depends on the degree of nonlinearity of the system. The greater the degree of nonlinearity, the smaller the region of validity of the linear approximation.

VisSim/Analyze provides the capability to linearize the system in both time and frequency domain representations. A state-space representation is used for the time-domain and transfer function is available for the frequency domain.

You can use VisSim/Analyze to analyze data in the following ways:

- Perform pole-zero and gain analysis
- Perform root locus analysis
- Perform Bode magnitude and phase analysis
- Perform Nyquist stability analysis

The state-space form of the linearized system can also be saved as a .M file and then read into MatLab, allowing you to perform matrix methods of analysis. The system can then be read back into VisSim for verification

The VisSim product family

The VisSim product family includes several base products and product suites, as well as a comprehensive set of targeted add-on modules that address specific problems in areas such as data communications, data acquisition, linearization and analysis, and digital signal processing.

Base products and product suites

Product	Function
Professional VisSim	Model-based design, simulation, testing, and validation of dynamic systems. A personal version, VisSim PE, is also available. VisSim PE limits diagram size to 100 blocks.
VisSim/Comm Suite	Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks. Includes Professional VisSim and VisSim/Comm blockset. A personal version, VisSim/Comm Suite PE, is also available. VisSim/Comm PE limits diagram size to 100 blocks and limits the Communication blockset. See the VisSim/Comm datasheet for details. VisSim/Comm Suite add-on modules are available for real-time data acquisition (Red Rapids digital tuner card); modeling PCCC turbo codes, including UMTS specification; and for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.
VisSim/Embedded Controls Developer Suite	Rapidly prototypes and creates embedded controls for DSPs, DSCs, and MSP430 microcontrollers. You can simulate and generate scaled, fixed-point ANSI C code, as well as code for on-chip peripherals. Includes Professional VisSim, VisSim/C-Code, VisSim/Fixed-Point, and one user-specified target support. A personal version, VisSim/Embedded Controls Developer PE, is also available. VisSim/Embedded Controls Developer PE limits diagram size to 100.
VisSim Viewer (free)	Lets you share VisSim models with colleagues and clients not licensed to use VisSim.

Add-on modules

Add-On Module	Function
VisSim/Analyze	Performs frequency domain analysis of a linearized nonlinear subsystem.
VisSim/CAN	Interfaces with a USB CAN device to read and write CAN messages on the CAN bus.
VisSim/C-Code	Generates highly-optimized, ANSI C code that can be compiled and run on any platform that supports an ANSI C compiler.
VisSim/C-Code Support Library Source	Provides source code for the Support Library.
VisSim/Comm blockset	Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks. A personal version, VisSim/Comm PE, is also available. VisSim/Comm PE is a subset of the Communication blockset. See the VisSim/Comm datasheet for details You can purchase VisSim/Comm add-on modules for real-time data acquisition (Red Rapids digital tuner cards); for modeling PCCC turbo codes, including UMTS specification; for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.
VisSim/Digital Power Designer	Provides high-level blocks for simulation and code generation of power supply and digital power components and controls.
VisSim/Fixed-Point	Simulates the behavior of fixed-point algorithms prior to code generation and implementation of the algorithm on the fixed-point target.
VisSim/Knobs and Gauges	Provides dynamic gauges, meters, and knobs for process control, and measurement and validation systems.
VisSim/Model-Wizard	Generates transfer function model from historic or real-time data.
VisSim/Motion	Simulates motor control systems with customizable amplifiers, controllers, filters, motors, sensors, sources, tools, and transforms.
VisSim/Neural-Networks	Performs nonlinear system identification, problem diagnosis, decision-making prediction, and other problems where pattern recognition is important.
VisSim/OPC	Connects to any OPC server and log data or run a virtual plant in VisSim for offline tuning.
VisSim/OptimizePRO	Performs generalized reduced gradient method of parameter optimization.
VisSim/Real-TimePRO	Performs real-time data acquisition and signal generation using I/O cards, PLCs, and DCSs.
VisSim/Serial	Performs serial I/O with other computers.
VisSim/State Charts	Creates, edits, and executes event-based systems.
VisSim/UDP	Performs data exchange over the internet using UDP.
VisSim Viewer (free)	Lets you share VisSim models with colleagues and clients not licensed to use VisSim.

Resources for learning VisSim/Analyze

For those of you that are new to VisSim, we have provided several free services to make your transition to VisSim fast, smooth, and easy:

- Interactive webinars
- VisSim movies
- Sample diagrams

Interactive webinars

Interactive webinars offer you the opportunity to meet with Altair product specialists who will introduce and demonstrate our software products live on your computer and answer any questions you have. Each webinar is approximately 45 minutes long. To learn more about our interactive webinars, go to http://www.vissim.com/webinars/webinars.html.

VisSim movies

Designed by Altair application engineers, the VisSim movies guide you through the creation, simulation, debugging, and optimizing of block diagrams that cover a broad range of engineering disciplines.

You can access the movies from your VisSim CD under the \MOVIES directory. Or, you can go to http://www.vissim.com/support/vissim_instructional_movies.html and download the movies to your computer.

Sample diagrams

VisSim 9.0 includes a directory of fully documented sample diagrams. These diagrams illustrate both simple and complex models spanning a broad range of engineering disciplines, including aerospace, biophysics, chemical engineering, control design, dynamic systems, electromechanical systems, environmental systems, HVAC, motion control, process control, and signal processing.

To access sample diagrams

1. Click on the **Diagrams** menu in VisSim.
2. Click on **Examples** > **Applications**.

To access online Help examples

- Click here to analyze Van Der Pol's nonlinear dynamical system

Training

Altair offers training sessions for learning and gaining expertise in VisSim and the VisSim family of add-on products. Training sessions are conducted at Altair training facility in Westford, MA, as well as at customer sites and as online webinars.

For information on setting up a training session, contacts sales@vissol.com.

Preparing a System for Analysis

Linear approximations of nonlinear systems provide the basis for most analysis and design techniques used in signal processing, communication, and control. This chapter explains how to prepare a system for analysis and how to extract the ABCD state-space matrix information. Later sections describe the other analysis methods provided by VisSim/Analyze.

Preparing a system for linearization

Before you can linearize a system, make sure that a Signal Consumer block (for example, a plot block or export block) is directly or indirectly connected to the system to be linearized.

If you have not simulated the block diagram containing the system to be linearized, VisSim automatically simulates for one time step before linearization. To simulate to a specific operating point, use the System menu’s Go, Stop, and Continue commands, or the corresponding ▶, ⏸, and ⁘ buttons in the toolbar.

Selecting the system for linearization

The blocks that you select constitute the system that VisSim linearizes. The following rules apply to the block set:

- There must be an input and output block that are left unselected and connected directly to the subsystem so that VisSim has an input and output reference point for the analysis. If there are multiple inputs and outputs, VisSim will select one for you and highlight the selected connectors. If you want to select from among multiple connectors, use the Select Input/Output Points command and click the mouse on the reference input and output you desire.
- To ensure that the system contains a state order, an integrator or transferFunction block must be part of the block set. If neither of these blocks is selected, VisSim/Analyze cannot generate a transfer function, root locus plot, Bode plot, or Nyquist plot. The ABC matrices will be null and the D matrix will contain the static gain of the system at the specified operating point.
- If, at the point of linearization, a `unitDelay`, `timeDelay`, `sampleHold`, Boolean, or any nonlinear block is present and is in a zero sensitivity state (for example, in a deadband or on a limit), then one or more rows of the resulting matrices will be zero.

To select a block set to linearize

- Use the single block selection method or area select method to select the block set. Selected blocks appear in reverse video. (See the *VisSim User's Guide* if you need instructions for using these selection methods.)

Selecting reference points

The Select Input/Output Points command lets you choose the input and output reference points for the linearization. If you don't choose reference points, VisSim/Analyze chooses arbitrary points when it linearizes the system. Unless you're linearizing a SISO (single input/single output) system, you should use Select Input/Output Points to ensure that the proper input and output points are selected.

Alternate way to select reference points: You can also select reference points with Analyze > Linearize, as described on page 8.

To select reference points

1. Choose **Analyze > Select Input/Output Points**.

   The pointer becomes a crossbar.

2. To select an input reference point, point to the output connector tab on the block that produces the signal with which you want to linearize the system and click the mouse.

   The output connector tab changes color. If the connector tab you try to select is invalid, VisSim beeps.

3. To select an output reference point, point to the input connector tab on the block that consumes the signal against which you wish to linearize and click the mouse.

   The input connector tab changes color. If the connector tab you try to select is invalid, VisSim beeps.

4. Click the mouse over empty screen, and at least **Error! Objects cannot be created from editing field codes.** from any block, to quit the Select Input/Output Points command.

5. VisSim linearizes the system with respect to the selected reference points.

To cancel a reference point selection

1. Choose **Analyze > Select Input/Output Points**.
2. The pointer becomes a crossbar.
3. Point to the selected connector tab and click the right mouse button.
4. Click the mouse over empty screen, and at least $\frac{1}{2}$" from any block, to quit the Select Input/Output Points command.

Generating ABCD state-space matrices

The Linearize command generates the ABCD state-space matrices from a nonlinear system by numerically evaluating the matrix perturbation equations at the time the simulation was halted and VisSim/Analyze was invoked. For a general multiple-input-multiple-output (MIMO) system, the ABCD state-space matrices contain the following coefficients to the vector equations:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$
$$\bar{y} = C\bar{x} + D\bar{u}$$

where:

$\dot{\bar{x}}$ = state vector derivative

$\bar{x}$ = state vector

$\bar{y}$ = output vector

$\bar{u}$ = input vector

and:

A = system matrix

B = input gain matrix

C = output gain matrix

D = direct link gain matrix

The Linearize command uses a central differencing scheme to approximate the partial derivative terms at the specified operating point. The four resulting matrices can be displayed directly on the screen or saved in a MATLAB-compatible .M file.

To generate and display ABCD state-space matrices on the screen

1. Prepare the system for linearization.
2. Choose **Analyze > Linearize**.

   The System Linearization dialog box appears.
3. Under Output Selection, activate **Linearize To Screen Display**.
4. Click **OK**, or press **ENTER**.

   Four dialog boxes appear in succession showing each matrix.
5. Click **OK** at the bottom of the dialog box, or press **ENTER**, to close the current matrix dialog box and open the next one.

To generate and write ABCD state-space matrices to an .M file

1. Prepare the system for linearization.
2. Choose **Analyze > Linearize**.

   The System Linearization dialog box appears.
3. Under Output Selection, activate **Linearize To MatLab File**.
4. In the Result File box, specify the name of the MatLab compatible .M file to which the matrices are to be written. When you select this parameter, enter the complete pathname for the MatLab file in the Result File box. VisSim/Analyze uses the following default specifications for the parts of the pathname that you omit:

If you omit	VisSim uses
The disk drive	C:
The directory name	The directory from which you opened VisSim
The file name	VABCD
The file extension	.M

5. Click on the **OK** button, or press **ENTER**.

The Linearize command writes the matrices to the specified file. To read the file into MatLab, see the MatLab documentation.

To select reference points with the Linearize command

1. Prepare the system for linearization.
2. Choose **Analyze > Linearize**.

   The System Linearization dialog box appears.
3. Click **Select Input/Output Points**.

   The pointer becomes a crossbar.
4. To select an input reference point, point to the output connector tab on the block that produces the signal with which you want to linearize the system and click the mouse.

   The output connector tab changes color. If the connector tab you try to select is invalid, VisSim beeps.
5. To select an output reference point, point to the input connector tab on the block that consumes the signal against which you want to linearize and click the mouse.

   The input connector tab changes color. If the connector tab you try to select is invalid, VisSim beeps.
6. Click the mouse over empty screen, at least $\frac{1}{2}$" from any block to exit this mode.

Reusing plots

VisSim/Analyze can display root locus, Bode phase and magnitude, and Nyquist plots. When a plot is displayed on the screen, for example, a root locus plot, and you want to change the ABCD state space system, and regenerate the plot, you have the choice of overwriting the existing plot or creating a new plot window.

To reuse plots

- Choose **Analyze > Reuse Plots**.

A check mark in front of the command indicates that the command is active.

Pole-Zero and Gain Analysis

For certain applications, a transfer function representation of the linearized system may be more desirable than a state-space representation. The Transfer Function Info command presents a linearized system in transfer function form and includes the following information:

- The numerator and denominator polynomials are presented in powers of the complex variable *s* defined as:

$s = \sigma + j\omega$

where σ and ω are real variables and $j = \sqrt{-1}$

- The gain of the transfer function
- The factors of the numerator (called zeros) and the denominator (called poles). The zeros and poles may appear in the following format:
- (real-part, imaginary-part)

VisSim/Analyze computes the transfer function from the state-space matrices using a similarity transformation. The poles and zeros are computed by converting the numerator and denominator polynomials to square matrices and computing the corresponding eigenvalues. If numerical round-off errors occur during the factoring process, they can be corrected by using Analyze > Preferences.

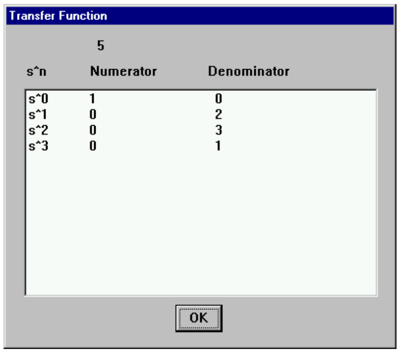
Generating a linear transfer function

Before you can generate a linear transfer function of the system, prepare the system for linearization.

To generate a linear transfer function

- Choose **Analyze > Transfer Function Info**.

VisSim displays the Transfer Function and Zeros and Poles dialog boxes, similar to the ones shown below, in succession. Click on the OK buttons, or press ENTER, to close the dialog boxes.

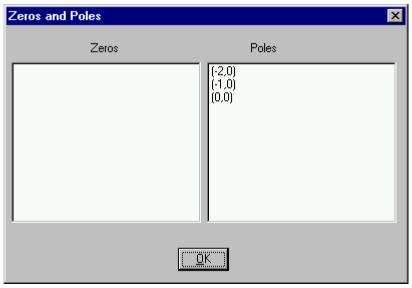


Transfer Function

5

s^n	Numerator	Denominator
s^0	1	0
s^1	0	2
s^2	0	3
s^3	0	1

Numerator and Denominator Polynomials, and Gain



Zeros and Poles

Zeros	Poles
	(-2,0) (-1,0) (0,0)

Zeros and Poles

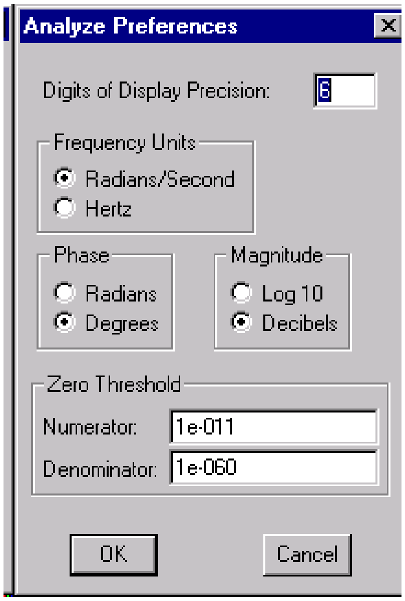
Correcting numerical round-off errors

Zeros and poles are computed by converting the linearized numerator and denominator polynomials to square matrices and then computing the corresponding eigenvalues. In general, this method is reliable and robust; however, to compensate for numerical round-off error (a floating point number only holds 15 digits of precision), you may want to clean the numerator and denominator polynomials prior to computing the poles and zeros. This is because very small terms in the transfer function (for example, 1e-15) will become very large terms after a matrix inverse (part of the operation of finding eigenvalues), resulting in erroneously large roots or poles. To fix the problem, set a threshold value under which VisSim/Analyze will treat a coefficient as zero. You can set thresholds for the numerator or denominator polynomial, or the eigenvalues themselves.

To correct the polynomials

1. Choose **Analyze > Preferences**.

   The Analyze Preferences dialog box appears.



Analyze Preferences

Digits of Display Precision: 6

Frequency Units
● Radians/Second
○ Hertz

Phase
○ Radians
● Degrees

Magnitude
○ Log 10
● Decibels

Zero Threshold
Numerator: 1e-011
Denominator: 1e-060

2. In the **Digits Of Display Precision** box, enter a number that indicates the round-off for poles and zeros.

3. Under **Zero Threshold** enter the appropriate values in the **Numerator** and **Denominator** boxes. Based on your knowledge of the system, you may raise or lower the threshold for zero treatment. The defaults are 1e-011 (for the numerator) and 1e-060 (for the denominator). Polynomial coefficients whose absolute values are less than these values multiplied by the absolute value of the largest coefficient in the polynomial will be set to zero.
4. Click on the **OK** button, or press **ENTER**.

For example, a polynomial with numerical error can be given as:

s^0	20000
s^1	1000
s^2	1e-10

The s^2 coefficient contains a numerical error. The actual value should be exactly zero. The Numerator parameter in the Preferences command should be set to a value less than (1e-10)*(20000) or 2e-6 to remove the error and set the s^2 coefficient to zero.

Root Locus Analysis

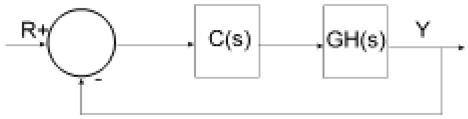
A root locus plot computes the locus of closed-loop poles for a transfer function while it's under gain feedback. The gain is swept over a wide range of values, typically from a small value near zero to a larger one.

For brevity, the transfer function can be denoted as *GH*(*s*), which represents a ratio of polynomials in *s*, and can be written as:

$$GH(s) = \frac{N(s)}{D(s)}$$

where *N*(*s*) is the numerator polynomial and *D*(*s*) is the denominator polynomial. The root locus gain is represented as *K*.

The following block diagram presents the structure VisSim/Analyze uses to perform the Root Locus command calculations:



The closed-loop poles are the roots of the characteristic equation (or denominator of the closed-loop transfer function) and can be computed as the roots of:

$$\rho(s) = 1 + KGH(s) = 0 = 1 + K\frac{N(s)}{D(s)} = 0 = D(s) + KN(s) = 0$$

For small values of *K*, the closed-loop poles are approximately the roots of *D*(*s*) = 0 (the open-loop poles); and for large values of *K* (positive or negative), the closed-loop poles are approximately the roots of *N*(*s*) = 0 (the open-loop zeros). The root locus plot presents the paths of the closed-loop poles as the value of *K* varies.

Generating a root locus plot

The Root Locus command automatically linearizes the system and produces a plot of the locus of closed-loop poles for the transfer function while it's under gain feedback. The auto-range capability built into VisSim/Analyze determines the appropriate dynamic range of interest for performing the gain sweep. The number of steps in the sweep determines the number of points in the root locus plot.

Before you can generate a root locus plot of the system, prepare the system for linearization.

1. Choose **Analyze** > **Root Locus Options** to define the point count resolution of the root locus gain (*K*) values to be investigated.
2. Choose **Analyze** > **Root Locus** to generate a root locus plot.

A root locus plot appears in the upper left-hand corner of the screen graphing the root locus of the specified system.

Manipulating a root locus plot

You can easily move, resize, magnify, and delete root locus plots.

To resize a plot

- Click on the **Maximize** button in the upper right-hand corner to expand the plot to full screen.
- Drag the plot's borders or corners to adjust its size.

To magnify a plot

1. Point to one corner of the area you want to zoom in on.
2. To anchor the corner, hold down the mouse button and **CTRL** key simultaneously.
3. Drag the pointer until the box encloses the area you want to magnify. A status box in the lower left-hand corner of the plot displays the point position.
4. Release the mouse button and **CTRL** key.

To zoom out

- Hold down the **CTRL** key and click the right mouse button over the plot.

To delete a plot

- Select the plot; then press the **DEL** key.

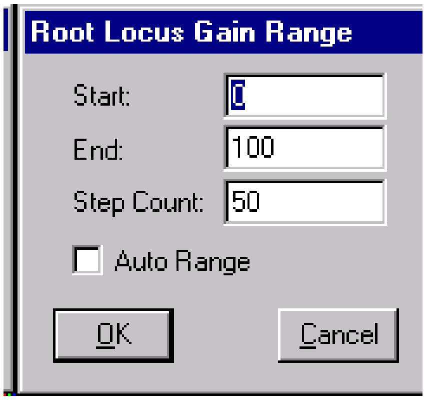
Setting root locus resolution

You can define the number of steps in the gain sweep, which in turn determines the point count resolution of the root locus gain.

To define the resolution for a root locus plot

1. Choose **Analyze > Root Locus Options**

   The Root Locus Gain Range dialog box appears.



Start: 0

End: 100

Step Count: 50

□ Auto Range

2. In the **Step Count** box, enter a value that indicates the step size for moving the gain. For a finer resolution, increase the step count. The default value is 50.

   Note: It is important to realize that the larger the step count value, the longer it takes for VisSim/Analyze to compute the new root locus plot.

3. Click on the **OK** button, or press **ENTER**.
4. Choose **Analyze > Root Locus** to generate a root locus plot with the specified step count.

Displaying gain, damping ratio, natural frequency, and real and imaginary coordinates in the *s*-plane

By clicking on the Read Coordinates button in the dialog box for a root locus plot, you can display a status bar that tracks the position of the crosshairs along the real (X) and imaginary (Y) axes, and the root locus gain (K). When you select a point in the left-half plane (to the left of the imaginary axis), the status bar also displays the natural frequency (w) and damping ratio (z). If you select a point in the right-half plane, the damping ratio and natural frequency parameters are not displayed in the status bar. In this situation, the parameters are meaningless performance measures for unstable systems. If you select a point off the loci, VisSim/Analyze uses the point on the loci nearest the selected point.

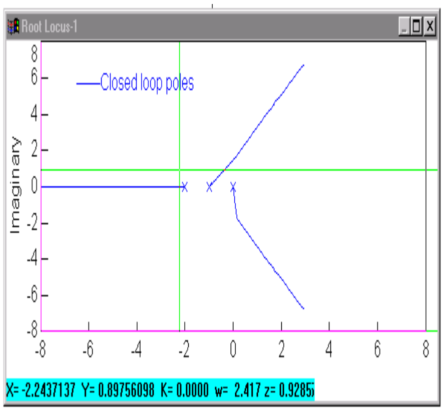
To activate the status bar

1. Choose **Edit > Block Properties**.
2. Point to the root locus plot and click the mouse.

   The Plot Properties dialog box appears.

3. Click on the **Options** tab.
4. Click on the **Read Coordinates** button.

VisSim redraws the plot with a status bar along the bottom of the plot. Crosshairs also appear over the plot and are controlled by the mouse.



X= -2.2437137 Y= 0.89756098 K= 0.0000 w= 2.417 z= 0.9285

You may have to extend the length of the plot block to fit the damping ratio and natural frequency values on the display.

Bode Magnitude and Phase Analysis

Bode analysis comprises graphical representations of magnitude and phase information, both plotted as a function of frequency (*w*). Bode plots clearly illustrate the relative stability of a system. Gain and phase margin requirements of a closed-loop system are often defined in terms of Bode plots.

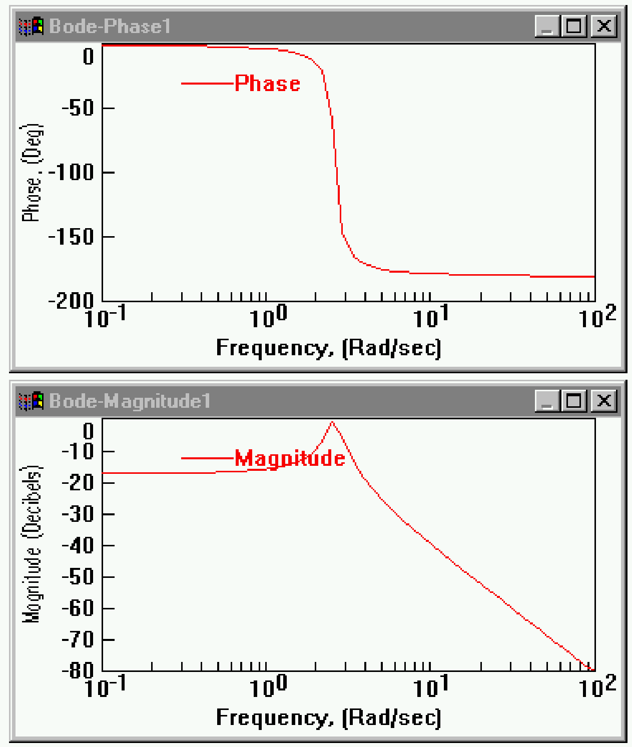
Bode plots have a variety of applications, including:

- Analysis of system closed-loop performance characteristics
- Determination of resonant frequencies and maximum power amplitude

Bode plots can also be used for the specification and design of compensators, as described in Compensator Design.

Generating Bode plots

When you choose the Frequency Response command, VisSim/Analyze produces two Bode plots similar to the ones shown below:



Both plots are relative to log frequency (in rad/s units). The magnitude units are log and the phase units are degrees.

Before you can generate Bode magnitude and phase plots of a system, prepare the system for linearization.

To generate Bode magnitude and phase plots

1. Choose **Analyze > Frequency Range** to define the range and resolution of the natural frequency.
2. Choose **Analyze > Frequency Response** to create the Bode phase and magnitude plots.

Manipulating a Bode plot

You can easily move, resize, magnify, and delete Bode plots.

To resize a plot

- Click on the **Maximize** button in the upper right-hand corner to expand the plot to full screen.
- Drag the plot's borders or corners to adjust its size.

To magnify a plot

1. Point to one corner of the area you want to zoom in on.

2. To anchor the corner, hold down the mouse button and **CTRL** key simultaneously.

3. Drag the pointer until the box encloses the area you want to magnify. A status box in the lower left-hand corner of the plot displays the point position.

4. Release the mouse button and **CTRL** key.

To zoom out

- Hold down the **CTRL** key and click the right mouse button over the plot.

To delete a plot

- Select the plot; then press the **DEL** key.

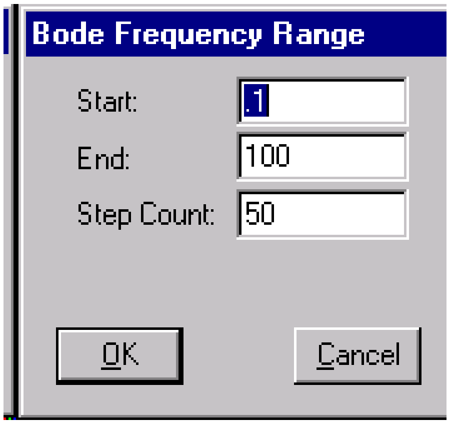
Specifying the frequency range, frequency units, and step count

You can change the frequency range, frequency units, and step count using the Frequency Range and Preferences commands. After you make the changes, choose the Frequency Response command to regenerate the Bode plots with the new values.

To change the frequency range

1. Choose **Analyze > Frequency Range**.

   The Bode Frequency Range dialog box appears.



Start: .1

End: 100

Step Count: 50

2. Do the following:

   - In the **Start** box, enter a starting point for the natural frequency range. The default value is 0.

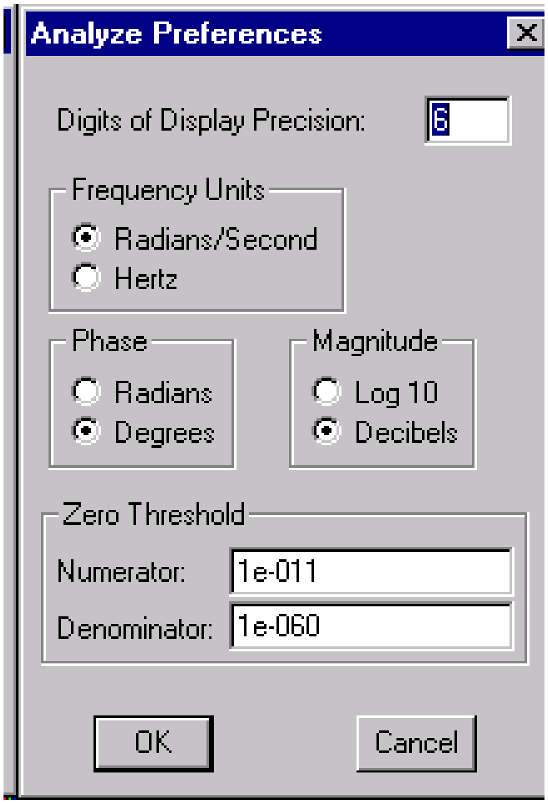
   - In the **End** box, enter an ending point for the natural frequency range. The default value is 100.

3. Click on the **OK** button, or press **ENTER**.

To specify frequency units

1. Choose **Analyze > Preferences**.

   The Analyze Preferences dialog box appears.



Digits of Display Precision: 6

Frequency Units
● Radians/Second
○ Hertz

Phase
○ Radians
● Degrees

Magnitude
○ Log 10
● Decibels

Zero Threshold
Numerator: 1e-011
Denominator: 1e-060

2. Specify the units.

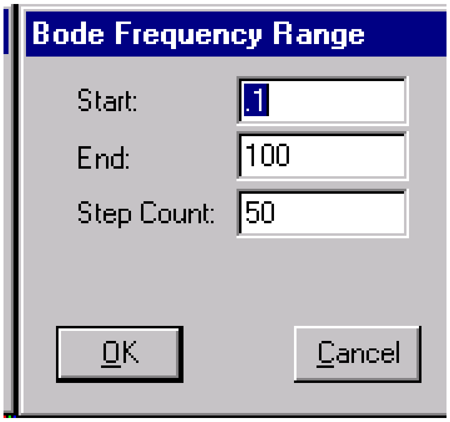
    - For the frequency units for phase and magnitude plots, choose **Radians/Second** or **Hertz** under **Frequency Units**.
    - For the phase unit in frequency response phase plots, choose **Radian** or **Degrees** under **Phase**.
    - For the magnitude unit in frequency response magnitude plots, choose **Log 10** or **Decibels** under **Magnitude**.

3. Click on the OK button, or press ENTER.

To change the step count

1. Choose **Analyze > Frequency Range**.

    The Bode Frequency Range dialog box appears.



Start: .1
End: 100
Step Count: 50

2. In the **Step Count** box, enter a step size for moving the frequency. For a finer resolution, increase the step count. The default value is 50.

3. Click on the **OK** button, or press **ENTER**.

Using plot crosshairs to pinpoint data coordinates

By clicking on the Read Coordinates button in the dialog box for a Bode plot, you can pinpoint any data point coordinates in the plot.

To activate the crosshairs

1. Choose **Edit** > **Block Properties**.

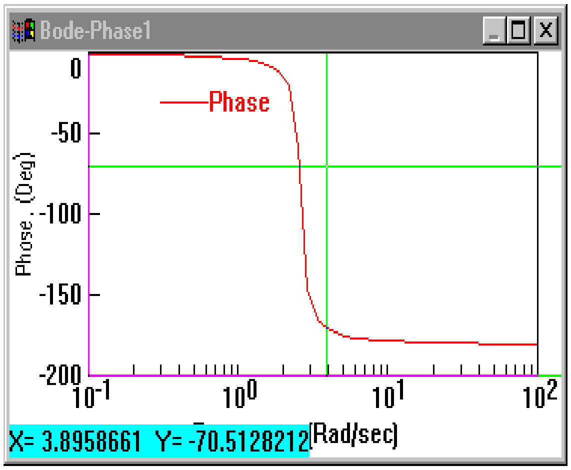
2. Point to the Bode plot and click the mouse.

   The Plot Properties dialog box appears.

3. Click on the **Options** tab.

4. Click on the **Read Coordinates** button.

VisSim redraws the plot a status bar along the bottom of the plot. Crosshairs also appear over the plot and are controlled by the mouse.



X= 3.8958661 Y= -70.5128212

The status bar tracks the position of the crosshairs with respect to the X and Y axes.

Nyquist Stability Analysis

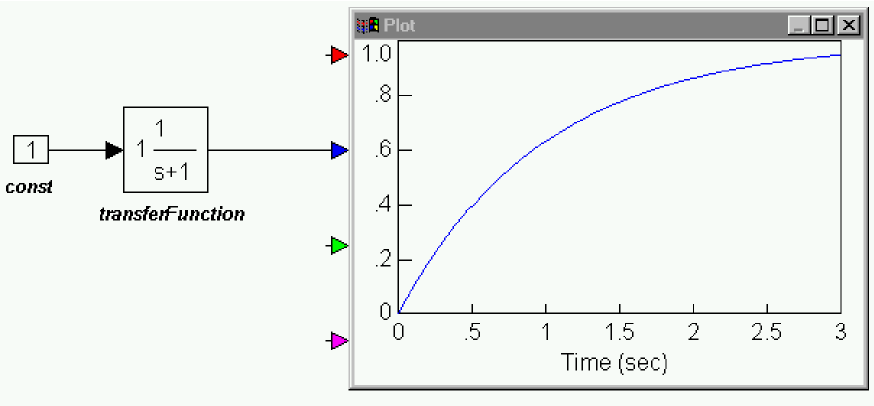
Nyquist analysis is a frequency response method for determining absolute and relative stability of closed-loop control systems. Most other methods, such as the Routh-Hurwitz criterion, can only determine absolute stability, and they require that the characteristic equation is a finite polynomial in *s*. Additionally, in the presence of delays in the system, these other methods only yield approximate information regarding system stability. In contrast, Nyquist analysis yields exact information about both absolute and relative stability of the system, even in the presence of delays.

Nyquist stability analysis of a type 0 system

To perform a Nyquist stability analysis, consider a simple type 0 system with the open-loop transfer function:

$$GH(s) = \frac{1}{(s+1)}$$

as shown in the diagram below.



To generate the Nyquist plot

1. Create the above VisSim diagram using a **const**, **transferFunction**, and **plot** block.

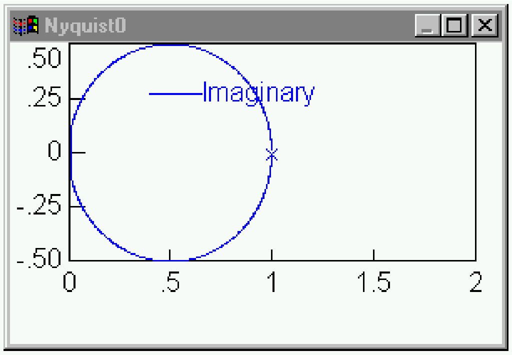
2. Enter the following polynomial coefficients to the transferFunction block:

   Numerator: 1

   Denominator: 1 1

Note: Always leave spaces between coefficient values.

3. Choose **System** > **Go**, or click on ▶ in the toolbar to simulate the diagram.
4. Select the **transferFunction** block.
5. Choose **Analyze** > **Nyquist Response**.
6. The Nyquist plot appears in the upper left corner of the screen.
7. Drag on its borders to adjust its size.



The Nyquist plot for this system is a circle, with the real part of $GH(s)$ on the horizontal axis and the imaginary part of $GH(s)$ on the vertical axis. On this plot, the origin represents $GH(j\infty)$ and the point of intersection with the horizontal axis ($Re(GH) = 1$) represents $GH(j0)$.

Manipulating a Nyquist plot

You can easily move, resize, magnify, and delete Nyquist plots.

To resize a plot

- Click on the Maximize button in the upper right-hand corner to expand the plot to full screen.
- Drag the plot's borders or corners to adjust its size.

To magnify a plot

1. Point to one corner of the area you want to zoom in on.
2. To anchor the corner, hold down the mouse button and CTRL key simultaneously.
3. Drag the pointer until the box encloses the area you want to magnify. A status box in the lower left-hand corner of the plot displays the point position.
4. Release the mouse button and CTRL key.

To zoom out

- Hold down the CTRL key and click the right mouse button over the plot.

To delete a plot

- Select the plot; then press the DEL key.

Absolute stability

A closed-loop system is absolutely stable if the roots of the characteristic equation have negative real parts.

Equivalently, the poles of the closed-loop transfer function, or the roots of the transfer function denominator polynomial 1 + *GH*(*s*), must lie in the left-half plane.

The Nyquist stability criterion establishes the number of poles of the closed-loop transfer function denominator polynomial 1 + *GH*(*s*) that are in the right-half plane, directly from the Nyquist stability plot of the open-loop transfer function *GH*(*s*).

Nyquist Stability Criterion

The closed-loop system, whose open-loop transfer function is *GH*(*s*), is stable if and only if $N = -P \le 0$, where:

- $P \equiv$ number of poles of *GH*(*s*) in the right-half plane ($P \ge 0$)
- $N \equiv$ total number of clockwise encirclements of the (-1,0) point ($GH(s) = -1$) in the *GH*(*s*)-plane

If $N > 0$, the number of zeros Z of 1 + *GH*(*s*) in the right-half plane is determined by $Z = N + P$.

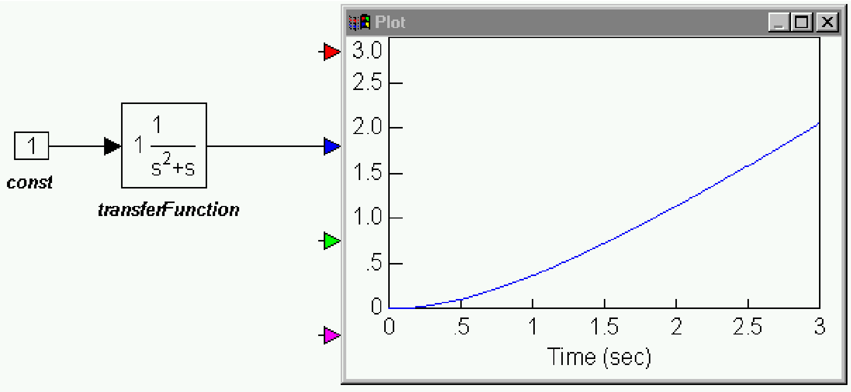
If $N \le 0$, the (-1,0) point is not enclosed by the Nyquist stability plot. Conversely, $N \le 0$ if the region to the right of the contour in the clockwise direction does not include the (-1,0) point.

Nyquist stability analysis of a stable type 1 system

Consider a type 1 system with the open-loop transfer function:

$$GH(s) = \frac{1}{s(s+1)}$$

as shown below.



To generate the Nyquist plot

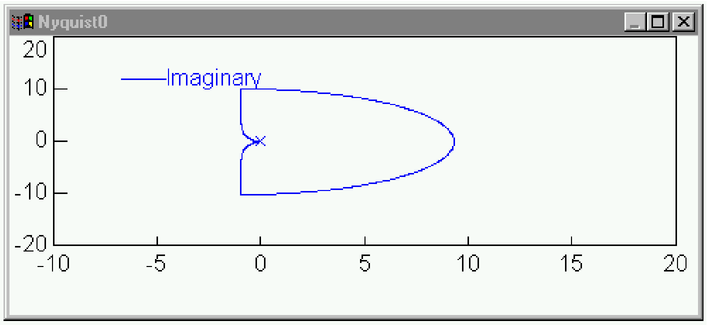
1. Create the above VisSim diagram using a const, transferFunction, and plot block.
2. Enter the following polynomial coefficients to the transferFunction block:

   Numerator: 1

   Denominator: 1 1 0

Note: Always leave spaces between coefficient values.

3. Choose **System > Go**, or click on ▶ in the toolbar to simulate the diagram.
4. Select the **transferFunction** block.
5. Choose **Analyze > Nyquist Response**.
6. VisSim/Analyze reminds you that the system has poles on the imaginary axis, which will result in Nyquist circles at infinity. Click on the OK button, or press ENTER.
7. In the Nyquist dialog box, you have the option to change the maximum frequency range. The default value is 10. Leave it unchanged and click on the OK button, or press ENTER.
8. The Nyquist plot appears on the screen.
9. Drag on its borders to adjust its size.



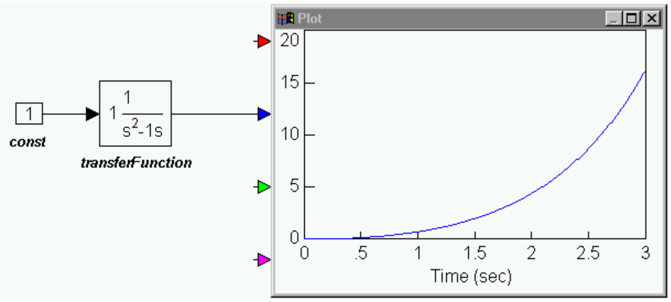
You can observe that the point (-1,0) is not enclosed by the Nyquist contour. Consequently $N \le 0$. The poles of $GH(s)$ are at $s = 0$ and $s = -1$, neither of which are in the right-half plane, which means that $P = 0$. Therefore, $N = -P = 0$ and the system is absolutely stable.

Nyquist stability analysis of an unstable type 1 system

Another example of a type 1 system is the open-loop transfer function:

$$GH(s) = \frac{1}{s(s-1)}$$

as shown below:



To generate the Nyquist plot

1. Create the above VisSim diagram using a const, transferFunction, and plot block.

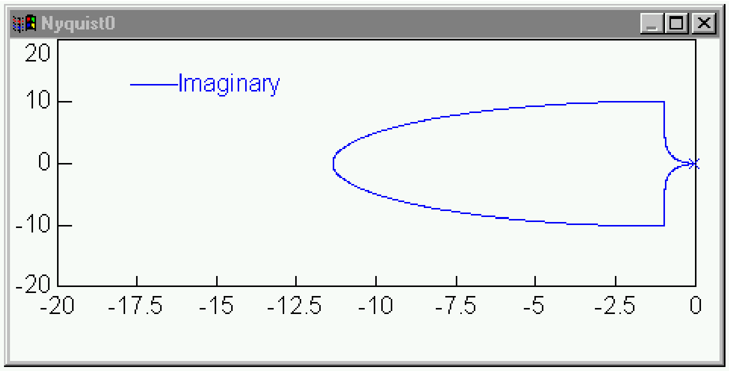
2. Enter the following polynomial coefficients to the transferFunction block:

   Numerator: 1

   Denominator: 1 -1 0

   Note: Always leave spaces between coefficient values.

3. Choose **System** > **Go**, or click on ▶ in the toolbar to simulate the diagram.
4. Select the **transferFunction** block.
5. Choose **Analyze > Nyquist Response**.
6. VisSim/Analyze reminds you that the system has poles on the imaginary axis, which will result in Nyquist circles at infinity. Click on the **OK** button, or press **ENTER**.
7. In the Nyquist dialog box, you have the option to change the maximum frequency range. The default value is 10. Leave it unchanged and click on the OK button, or press ENTER.
8. The Nyquist plot appears on the screen.
9. Drag on its borders to adjust its size.



In this case, you can observe that the point (-1,0) is enclosed by the Nyquist contour. Consequently $N > 0$. Moreover, since the number of clockwise encirclements of the point (-1, 0) is one, $N = 1$. The poles of $GH(s)$ are at $s = 0$ and $s = +1$, with the second pole appearing in the right-half plane. This implies that P, the number of poles in the right-half plane, equals one.

In this case, $N \neq -P$, which indicates that the system is unstable.

The number of zeros of $1 + GH(s)$ in the right-half plane is given by:

$Z = N + P = 1 + 1 = 2$

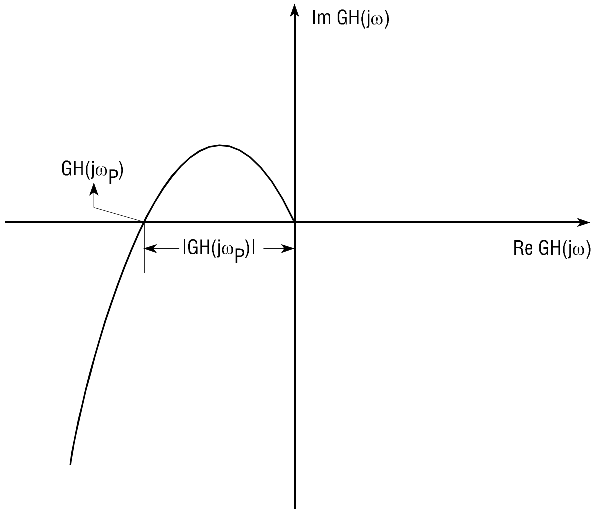
Relative stability

The most common relative stability measures are the gain margin, gain crossover frequency, phase margin, and phase crossover frequency. These quantities can be deduced from the Nyquist stability plot.

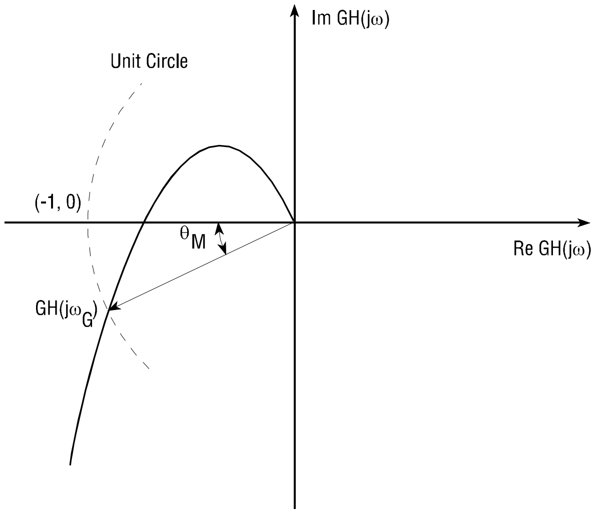
The phase crossover frequency is the frequency at which the phase angle of GH(*jw*) is -180o. This is the frequency at which the Nyquist plot crosses the negative real axis. The gain margin is equal to:

$$\frac{1}{|GH(j\omega)|}$$

These quantities are shown in the following figure:

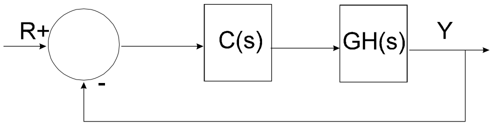


The gain crossover frequency is the frequency at which |GH(j□)| = 1. The phase margin is the angle by which the Nyquist plot must be rotated in order to intersect (pass through) the (-1,0) point. These quantities are depicted in the figure below.



Compensator Design

A compensator is an additional component that is incorporated into a system to equalize or compensate for performance insufficiency. There are several basic types of compensators used in many classical SISO control applications. The majority of these compensators are normally placed in series with the system to be compensated with feedback applied around the combination. This configuration is often called cascade compensation and is illustrated below:



The system to be compensated is defined as *GH*(*s*) and the compensator as *C*(*s*). Negative feedback is applied around the series combination. The signal *R* is a reference signal, and the signal *Y* is the output signal.

The basic types of compensators designed in this chapter are:

- Gain or Proportional (P)
- Proportional - Derivative (PD)
- Lead
- Proportional - Integral (PI)
- Lag

System characteristics

In designing each compensator, the following system characteristics hold true:

- LM = Log magnitude, base 10
- ω_c = Cross-over frequency of open-loop Bode LM plot
- ϕ_{pm} = Phase margin of open-loop Bode phase plot
- τ = Time constant, s
- ς = Damping ratio

The open-loop Bode frequency response can be used to approximate closed-loop system behavior, provided the open-loop frequency response satisfies three conditions:

- The gain at low frequencies is relatively high for low steady-state errors
- The gain at high frequencies is relatively low for disturbance attenuation
- The slope of the Bode plot is approximately -1 $LM/decade$ in the range $(0.4\omega_c, 4\omega_c)$

When the above conditions are approximately satisfied, the following correlations are valid:

$$\tau \approx \frac{1}{\omega_c} \equiv \text{time constant, s}$$

$$\varsigma \approx \begin{cases} .01 * \phi_{pm} & \text{for } \phi_{pm} < 60^0 \\ .027 * \phi_{pm} - 1 & \text{for } 75^0 > \phi_{pm} > 60^0 \end{cases}$$

Gain compensator

Pure gain compensators are generally known as proportional or P controllers. Such controllers can be used to satisfy phase margin requirements of a compensator-plant system. However, the frequency at which the phase margin requirement is satisfied, cannot be controlled by gain compensation alone.

To design a gain compensator

1. Generate the Bode magnitude and phase plots of the open-loop transfer function *GH*(*s*).
2. Determine the frequency ω_{pm} at which the desired phase margin occurs.
3. Determine the magnitude of GH*(s)* corresponding to ω_{pm}. Let this magnitude be $|G(j\omega_{pm})H(j\omega_{pm})|$.
4. Use a gain of:

$$K_c \approx \frac{1}{|G(j\omega_{pm})H(j\omega_{pm})|}.$$

5. Generate the Bode magnitude and phase plots and check the phase margin. To fine-tune the gain, make small changes about:

$$K_c \approx \frac{1}{|G(j\omega_{pm})H(j\omega_{pm})|}$$

Interactive gain compensator design

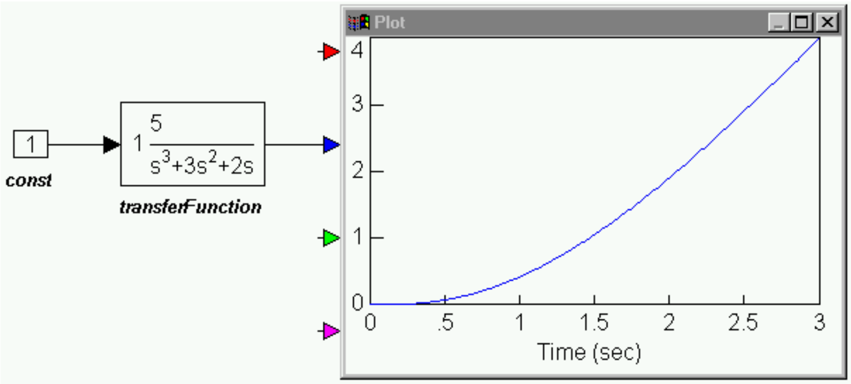
Gain compensator design can be performed interactively in VisSim/Analyze. For example, consider a system with an open-loop transfer function:

$$GH(s) = \frac{5}{(s+0)(s+1)(s+2)} = \frac{5}{s^3 + 3s^2 + 2s}$$

You are required to determine the appropriate gain compensation such that the phase margin is 45°.

To design the gain compensator

1. Create and simulate the following system.



2. Enter the following polynomial coefficients to the transferFunction block:

   Numerator: 5

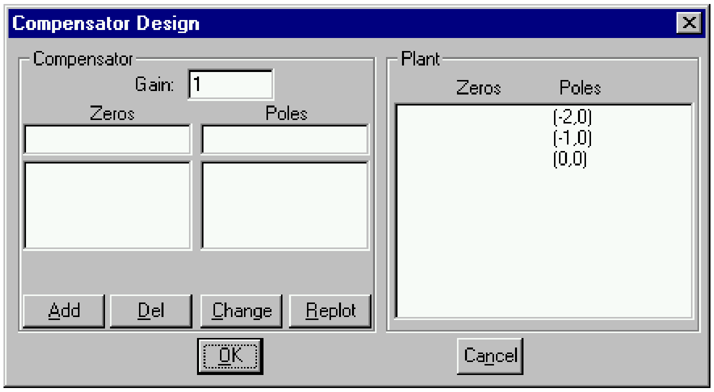
   Denominator: 1 3 2 0

   Note: Always leave spaces between coefficient values.

3. Select the **transferFunction** block.

4. Choose **Analyze > Compensator Design**.

   The Compensator Design box is displayed.



Compensator Design

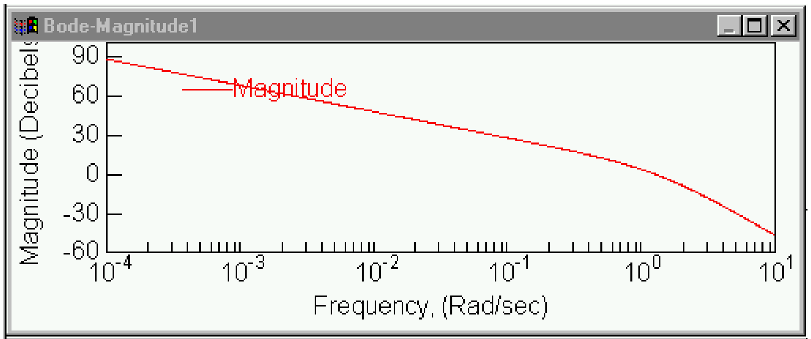
Compensator

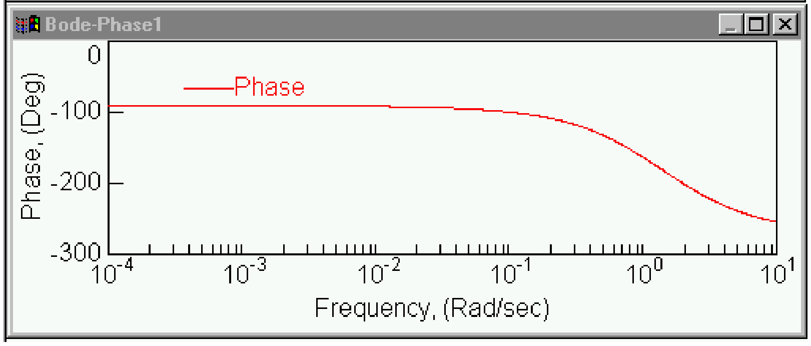
Gain: 1

Plant

Zeros	Poles
	(-2,0) (-1,0) (0,0)

5. Click on the **Replot** button. The Bode magnitude and phase plots, and the root locus plot are displayed. For this exercise, you can ignore the root locus plot.





The phase margin of the uncompensated system is approximately 13°. For a phase margin of 45°, the corresponding ω_{pm} from the Bode phase plot is 0.54 rad. For this frequency, the magnitude of the open-loop transfer function $|G(j\omega_{pm})H(j\omega_{pm})|$ is 3.541. The compensator gain is:

$$K_c \approx \frac{1}{\left|G(j\omega_{pm})H(j\omega_{pm})\right|} = \frac{1}{3.541} = 0.282\,.$$

6. In the Compensator Design dialog box, do the following:

   - In the **Gain** box, enter 0.**282**.
   - Click on the **Replot** button.

   The updated Bode magnitude and phase plots are displayed.

7. If necessary, make minor adjustments to the gain value until the Bode plots indicate that the phase margin requirement is satisfied.

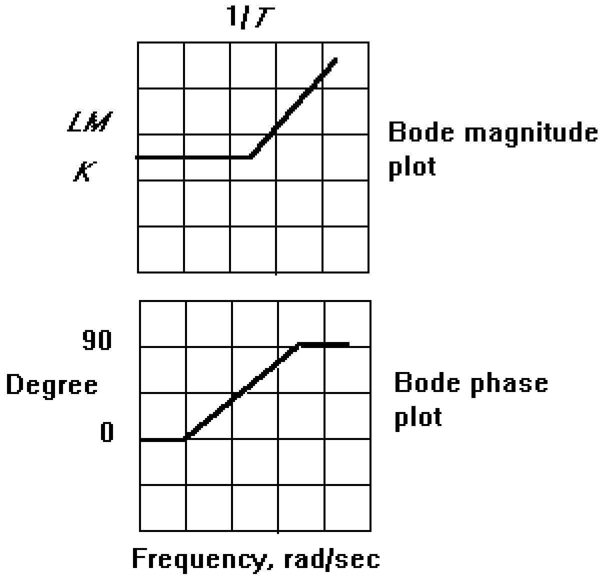
Proportional - derivative (PD) compensator

The primary function of the proportional-derivative (PD) compensator is to add phase lead to a system such that the system reaches the steady state quickly and smoothly.

Transfer function

$$C(s) = K(Ts + 1)$$

Bode plots



The characteristics of the Bode plots are summarized in the following PD table:

Frequency	*LM*	Phase
$<<<\frac{1}{T}$	K	0
$\frac{1}{T}$	K	45
$>>>\frac{1}{T}$	∞	90

To design a PD compensator

1. Using the PD table, determine the frequency (ω*) where phase is to be added, and determine the amount of phase to be added.
2. Select the PD zero at the frequency that produces the desired phase (and interpolate where necessary).
3. Implement the compensator and adjust the K gain for a satisfactory transient response, beginning with $K = 1$.

Although the PD compensator provides phase lead at frequencies above $1/T$, it also adds more and more amplification as frequency increases. When noise is present, as it usually is, this is an undesirable property because the PD compensator amplifies the noise to a large degree.

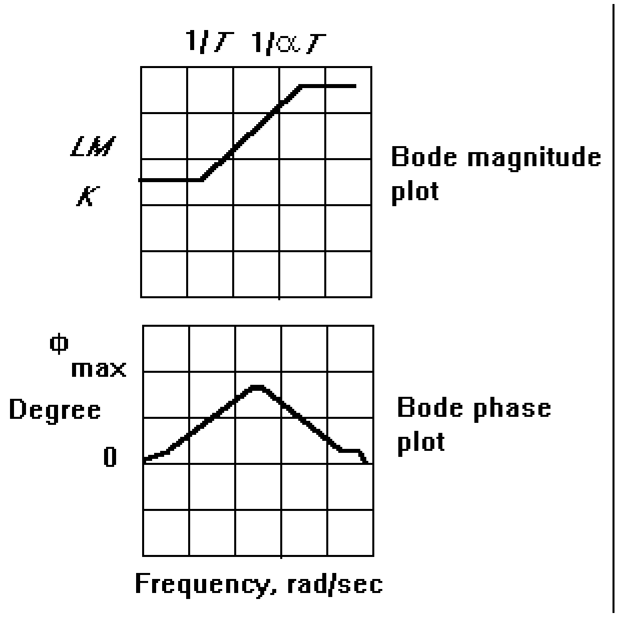
Lead compensator

Similar to the PD compensator, the lead compensator adds phase to a system such that the system reaches the steady state quickly and smoothly. However, unlike the PD compensator, the lead compensator controls the amount of gain added at high frequencies, such that noise is not amplified.

Transfer function

$$C(s) = K\frac{(Ts+1)}{(\alpha Ts+1)}$$

Bode plots



The quantity 1/α is called the lead ratio. It is responsible for the amount of phase added by the lead compensator. The larger the lead ratio, the more spread apart are the pole and zero of the lead compensator, and consequently, the more phase added. The peak phase added lies at a frequency halfway between the pole and zero at:

$$\frac{1}{\sqrt{\alpha * T}}$$

The characteristics of the Bode plots are summarized in the following lead table:

Lead ratio 1/α	Phase
1	0
2	17.5
5	37.5
10	55
20	62.5
50	72.5
100	80

To design a lead compensator

1. Based on the Bode plots of the uncompensated system, determine the following:
    - The frequency (ω*) where phase is to be added
    - The amount of phase to be added ((ϕ^*)
2. Using the lead table, compute the lead ratio, given the ϕ^*.
3. Compute the lead pole and zero:

$$\frac{1}{T} = \sqrt{\alpha} * \omega^* \text{ for the zero}$$

$$\frac{1}{\alpha} * \frac{1}{T} = \text{pole}$$

4. Implement the lead compensator and set *K* slightly less than 1 to counteract the additional gain of the compensator.
5. Adjust the *K* for the desired transient response.

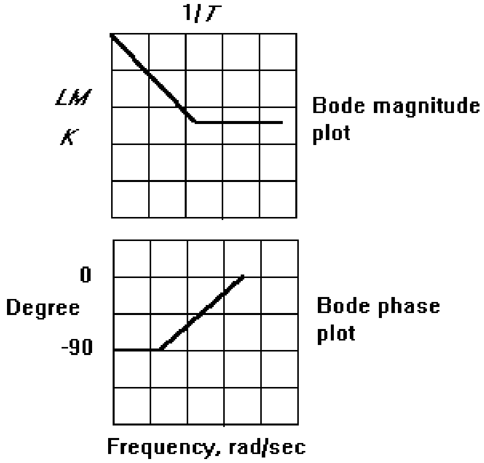
Proportional - integral (PI) compensator

The primary function of the proportional-integral (PI) compensator is to increase the low frequency gains to reduce steady-state errors and decrease sensitivity to noise and parameter variations.

Transfer function

$$C(s) = K\frac{(Ts+1)}{s}$$

Bode plots



To design a PI compensator

Because the function of the PI compensator is to increase low frequency gain, the design steps differ from those for the previous two compensators.

1. Determine the frequency (ω*) that defines the region ω < ω*in which the gain is to be increased.
2. Determine the amount of gain to be added (*K**) using the error coefficient specifications for the system.
3. Identify the cross-over frequency and phase margin from the uncompensated system Bode plots, ω_c and ϕ_{pm}.
4. Select the zero of the PI compensator to satisfy the steady-state error specifications. If, however, a phase margin problem exists (for example, the phase margin is less than 30 degrees), do not allow it to move past $0.4\omega_c$.

5. Initially select $K = 1$ and tune the compensator slightly by modifying the value of K for the desired performance.

Lag compensator

The primary function of the lag compensator is to increase the low frequency gain to reduce steady-state errors and decrease sensitivity to noise and parameter variations.

To design a lag compensator

1. Adjust the DC gain of the open-loop transfer function $GH(s)$ to satisfy low-frequency specifications. Let the adjusted gain be K_c.
2. Find the frequency ω_1 for which the angle of $G(j\omega)\, H(j\omega)$, θ_{GH}, is equal to $(-180^o + \theta_M + 5^o)$, where θ_M is the specified phase margin.
3. The magnitude of the lag zero is then equal to $\omega_0 = 0.1\ \omega_1$.
4. The ratio of the lag pole to lag zero is given by:

$$\frac{\omega_p}{\omega_0} = \frac{1}{\left|G(j\omega_1)\, H(j\omega_1)\right|}$$

5. Solve the equations in steps 3 and 4 for ω_0 and ω_p.
6. The compensator transfer function is then given by:

$$C(s) = \frac{K_c(1 + s/\omega_0)}{1 + s/\omega_p}$$

This procedure is used to interactively design a lag compensator in the following section.

Interactive lag compensator design

VisSim/Analyze can also be used to perform interactive compensator design functions. Consider a system with the open-loop transfer function:

$$GH(s) = \frac{1}{(s+0)(s+1)(s+2)} = \frac{1}{s^3 + 3s^2 + 2s}$$

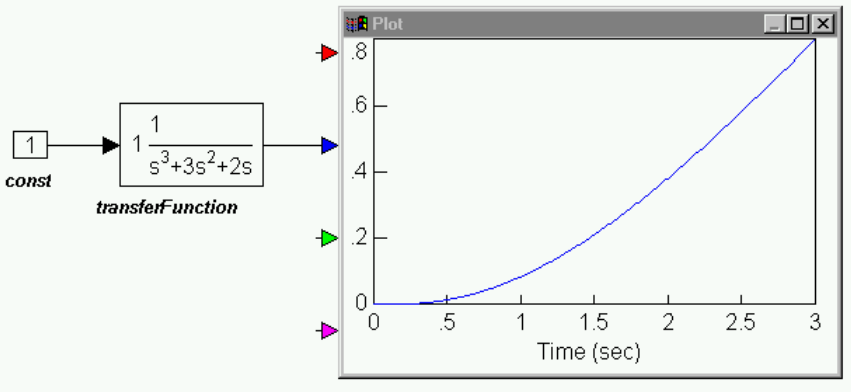
You are required to design a lag compensator for this system such that a phase margin of 45^o is maintained. Assume that $K_c = 1$.

The angle θ_{GH} is computed from the equation:

$\theta_{GH} = -180^o + 45^o + 5^o = -130^o$

To design a lag compensator

1. Create and simulate the following system.



2. Set the polynomial coefficients for the `transferFunction` block to the following values:

   Numerator: 1

   Denominator: 1 3 2 0

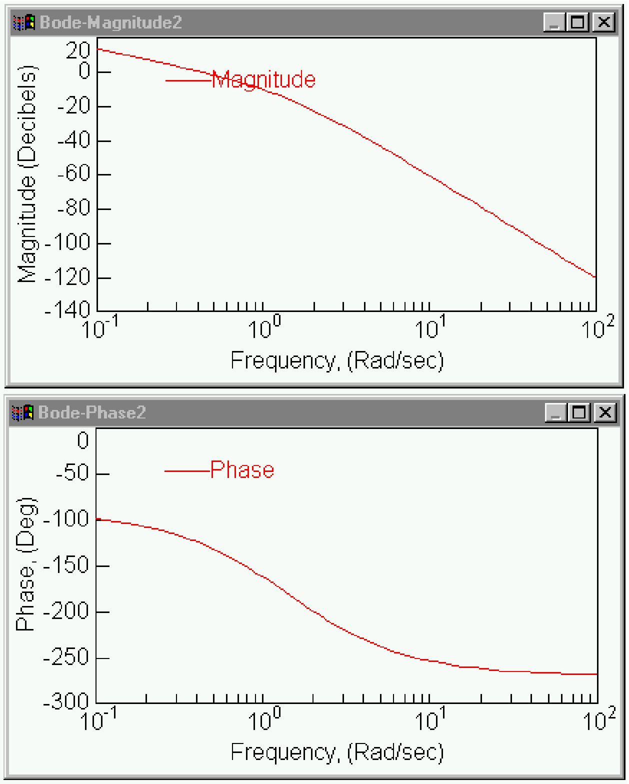
   Note: Always leave spaces between coefficient values.

3. Select the **transferFunction** block.

4. Choose **Analyze > Frequency Response**

   The Bode magnitude and phase plots are displayed.

5. Move and resize them for better viewing.



As can be seen from the Bode phase plot, the frequency corresponding to a phase of -130° is approximately 0.48 rad. Therefore, ω_1 = 0.48. The corresponding magnitude of the open-loop transfer function, from the Bode magnitude plot, is 0.853.

The lag zero frequency is $\omega_0 = 0.1\ \omega_1 = 0.048$. The lag pole frequency is given by:

$$\frac{\omega_p}{\omega_0} = \frac{1}{\left|G(j\omega_1)\,H(j\omega_1)\right|} = 1/0.853 = 1.172$$

$\omega_p = (1.172)\,(0.048) = 0.056$ rad.

The compensator transfer function is:

$$C(s) = \frac{K_c(1 + s/\omega_0)}{1 + s/\omega_p} = \frac{1(1 + s/0.048)}{(1 + s/0.056)}$$

or:

$$C(s) = \frac{1.167(s + 0.048)}{s + 0.056}$$

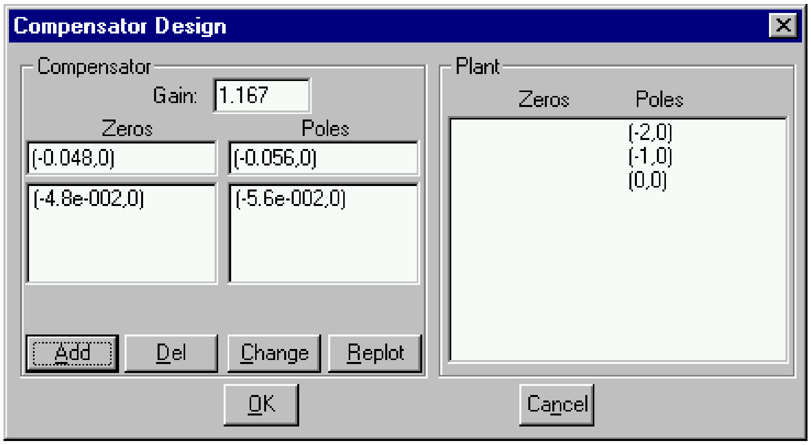
6. Select the **transferFunction** block from the diagram.
7. Choose **Analyze > Compensator Design**

   The Compensator Design dialog box appears with the plant poles and zeros displayed.
8. Do the following:

   - In the **Compensator Zeros** box, enter (-0.048,0) and click on the **Add** button.

     This zero is displayed in the Compensator Zeros list. The entry corresponds to a zero in the complex plane, with a real part of -0.048 and an imaginary part of 0.
   - In the **Compensator Poles** box, enter (-0.056,0) and click on the **Add** button.

     This pole is displayed in the **Compensator Poles** list.
   - In the **Gain** box, enter **1.167**.



Compensator Design

Compensator

Gain: 1.167

Zeros: (-0.048,0)

Poles: (-0.056,0)

Zeros list: (-4.8e-002,0)

Poles list: (-5.6e-002,0)

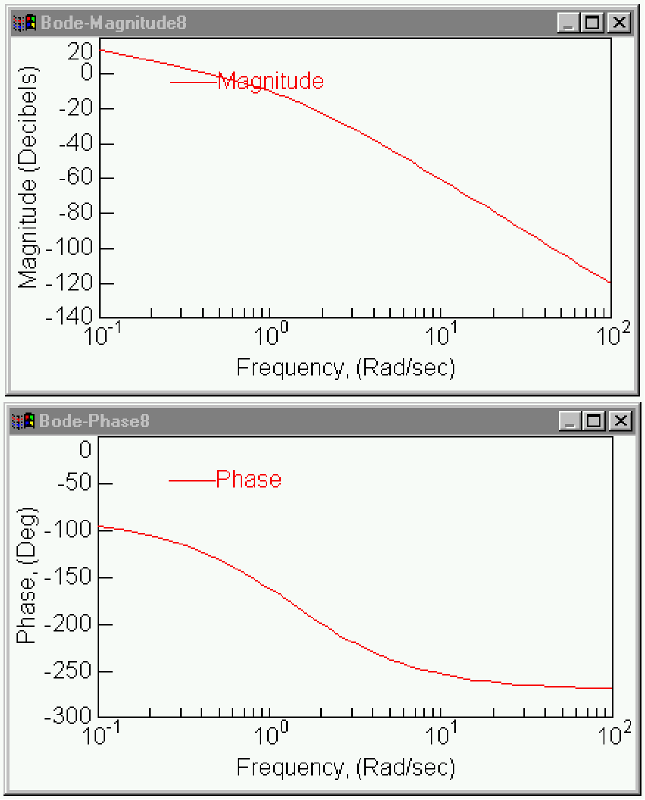
Plant

Zeros

Poles: (-2,0) (-1,0) (0,0)

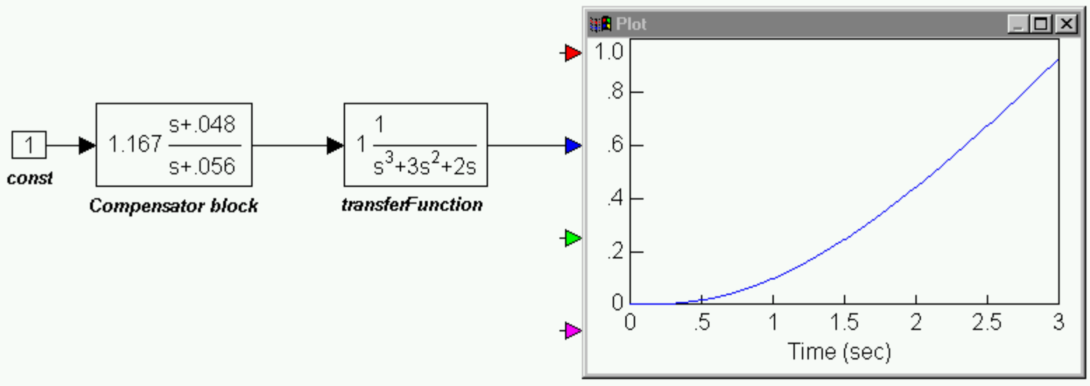
9. Click on the **Replot** button.

The Bode magnitude and phase plots of the Compensator-Plant system, $C(s)$ $GH(s)$ will be displayed. The Bode plots indicate that the phase margin is approximately 48°.



10. Click on the **OK** button in the **Compensator Design** dialog box.

11. Place the new **Compensator** block ahead of the plant transfer function *GH*(*s*), and connect it as shown below. The Compensator-Plant system is now ready for use or further analysis.



Modifying compensator blocks

After preliminary design of a feedback compensator, it may be necessary to fine-tune the characteristics of the compensator. Moreover, in a system that contains multiple compensators, you may wish to modify a particular compensator in isolation.

To modify a compensator block

1. Select the **plant transfer function** block.
2. Choose **Analyze > Select Compensator Block**.
3. Point to the **compensator** block to be modified and click the mouse.
4. Click the mouse on empty screen space.
5. Choose **Analyze > Compensator Design**.
6. Make the desired changes in the **Compensator Design** dialog box.

Analyzing Van Der Pol's Nonlinear Dynamical System

Van Der Pol's nonlinear dynamic system is represented in the following differential equation:

$$\frac{d^2x}{dt^2} - (1-\alpha x^2)\frac{dx}{dt} + x = 0$$

where at least one of the following initial conditions is met:

$$\frac{dx}{dt}(t_0) \neq 0$$

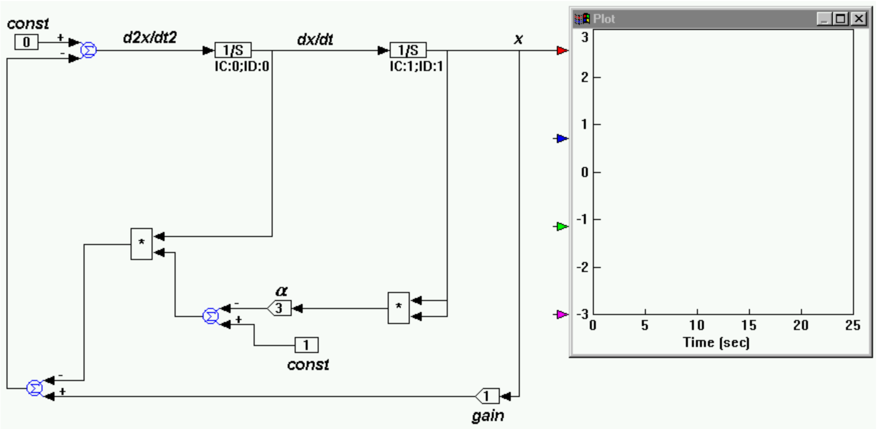
$$x(t_0) \neq 0$$

This section steps you through the process of building Van Der Pol's system in block diagram form, and generating ABCD state-space matrices, transfer function information, and Bode and root locus plots.

Building Van Der Pol's nonlinear system

To build Van Der Pol's nonlinear dynamical system

1. Convert Van Der Pol's system to block diagram form, as shown below.



2. To satisfy the conditions of the equation, make the following block parameter assignments:

   - Set the gain (α) to 3.
   - Set the initial condition of the first `integrator` (the block from which dx/dt is generated) to 0.
   - Set the initial condition for the second `integrator` to 1

3. Choose **System** > **System Properties**.

   The System Properties dialog box appears.

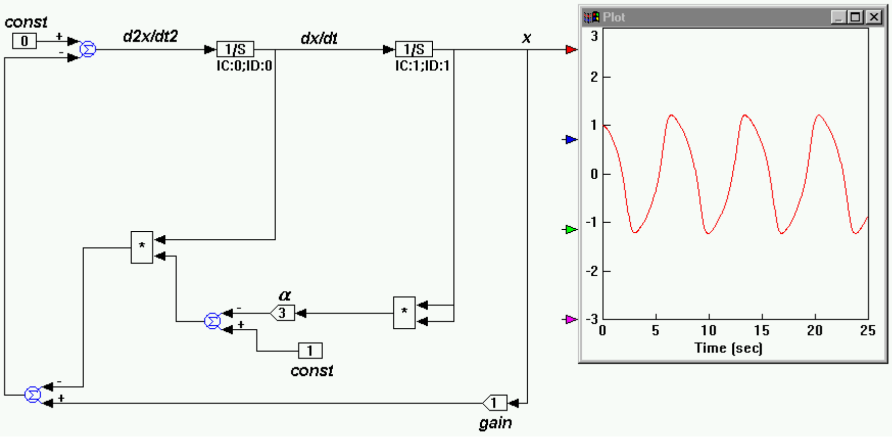
4. Click on the **Range** tab and make the following changes:

   - In the Range Start box, enter 0.
   - In the Range End box, enter 25.
   - In the Step Size box, enter 0.05.

5. Click on the **Integration Method** tab and make the following changes:

   - Activate **Euler**.
   - Click on the **OK** button, or press **ENTER**.

6. Choose **System** > **Go**, or click the ▶ toolbar button, to start the simulation.



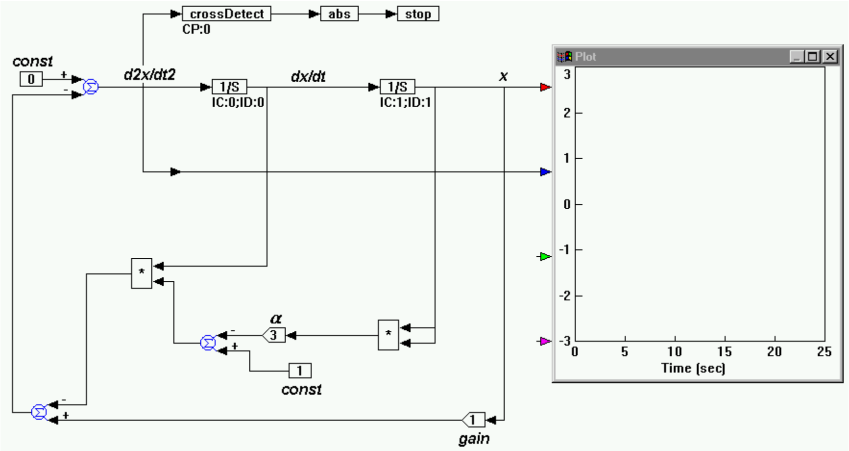
Linearizing Van Der Pol's nonlinear system

To linearize the system about a specific operating point

An interesting operating point about which to linearize the system is when the d2x/dt2 signal is equal to 0. At this point, the linearization results in stable poles.

1. Wire a **crossDetect** block to the **d2x/dt2** signal.
2. Feed the output into an **abs** block, which is wired to a **stop** block.
3. Wire the **d2x/dt2** signal into the **plot** block.

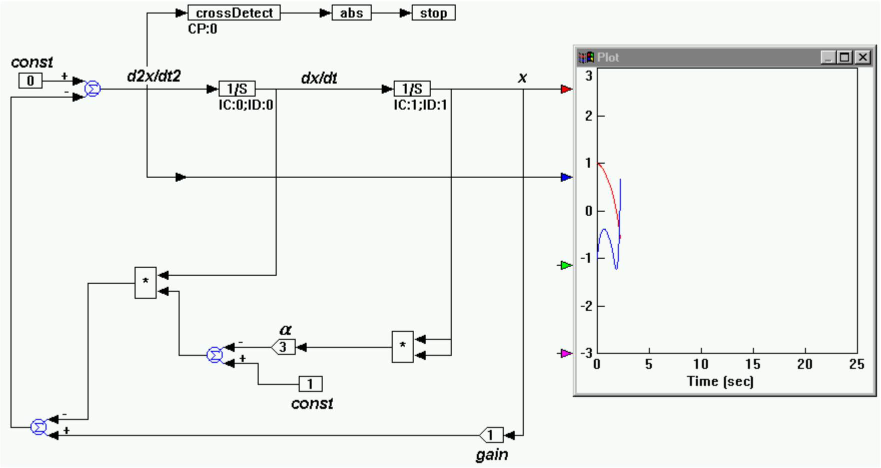
   Upon completion, the block diagram will look like this:



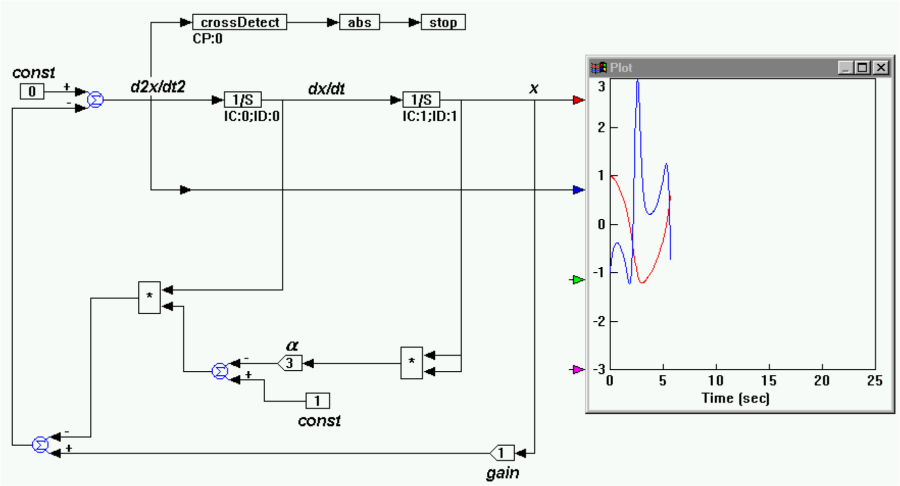
In this configuration, VisSim/Analyze automatically stops the simulation when the d2x/dt2 signal is exactly 0. The crossDetect block outputs 1 or -1 depending on whether the crossing occurs with a positive or negative slope. Since the `stop` block stops the simulation only when the input is 1 or greater, an `abs` block is introduced between them to ensure that the `stop` block receives only positive inputs.

4. Choose **System** > **Go**, or click the ▶ toolbar button, to start the simulation.

   The simulation runs to the first occurrence of signal d2x/dt2 = 0.

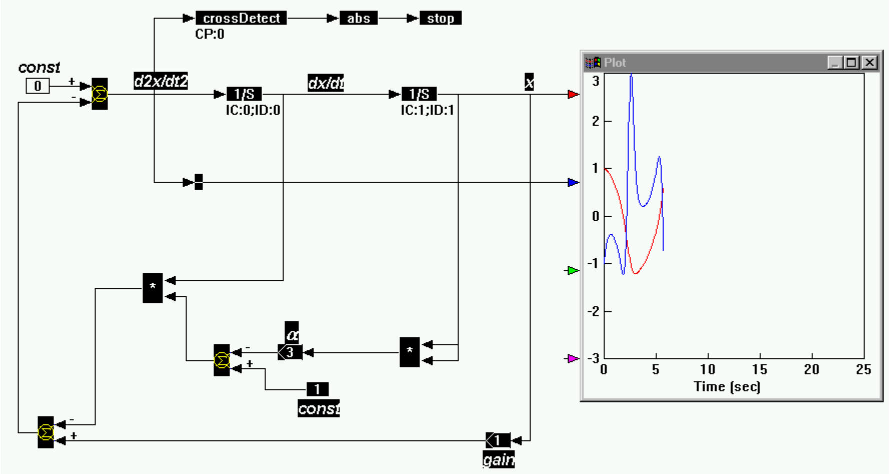


5. Choose **System** > **Continue**, or click on the toolbar button, to continue the simulation to the next occurrence of d2x/dt2 = 0.



To linearize the system

1. Select the block set to be analyzed. The block set includes all but the 0 input const block and the plot block.



2. Choose **Analyze** > **Select Input/Output Points** to specify the reference points for the linearization.

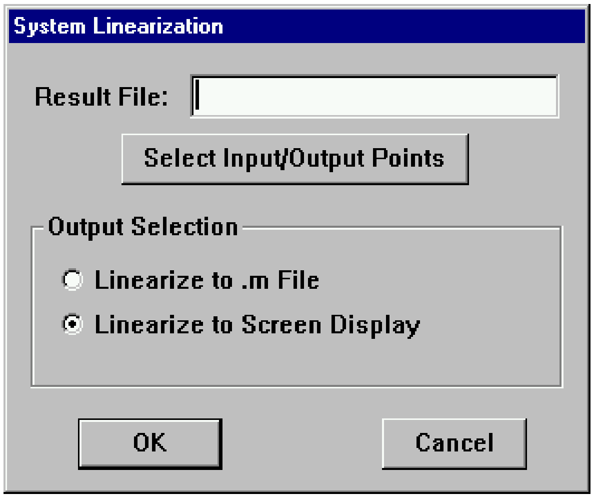
3. Point to the output connector tab on the 0 input **const** block and click the mouse.

4. Point to the input connector tab on the **plot** block to which the **d2x/dt2** signal is wired and click the mouse.

5. Point to empty screen and click the mouse to exit this mode.

6. Choose **Analyze** > **Linearize**.

The System Linearization dialog box appears.



System Linearization

Result File:

Output Selection

○ Linearize to .m File

● Linearize to Screen Display

7. Do one of the following:

    - To display the ABCD matrices in four separate, successive dialog boxes, select **Linearize To Screen Display** parameter.

    - To write the matrix information to a .M file, select the **Linearize To .M File** parameter. You must enter a name for the file in the **Result File** box. For this example, the contents of the file will be:

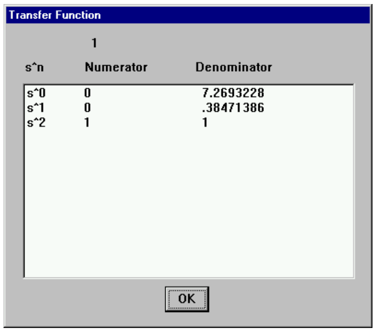
```
function [a,b,c,d] = vabcd
a = [-.384714 -7.26932;
    1 0 ];
b = [1 ;
    0 ];
c = [-.384714 -7.26932];
d = [1 ];
```

Generating transfer function information, root locuys plots, and Bode plots

To generate the transfer function information

- Choose **Analyze > Transfer Function Info**.

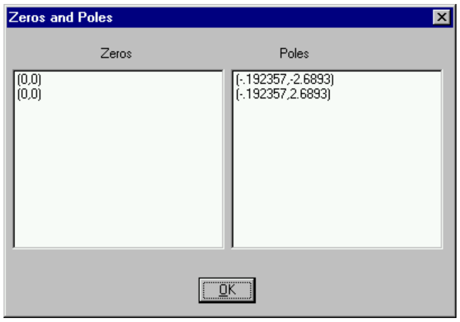
VisSim displays the numerator and denominator coefficients, and the zeros and poles in successive dialog boxes.



Transfer Function

1

s^n	Numerator	Denominator
s^0	0	7.2693228
s^1	0	.38471386
s^2	1	1



Zeros and Poles

Zeros	Poles
(0,0)	(-.192357,-2.6893)
(0,0)	(-.192357,2.6893)

The input to the system is the 0 constant (which is denoted as *R*), and the output is the dx/dt signal (as previously defined). The first dialog box presents the transfer function as numerator and denominator polynomials in power of *s*. Denoting the output as:

$$y \equiv \frac{dx}{dt}$$

the transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{s^2}{s^2 + 0.384714s + 7.26932}$$

The gain (*s* = 0 gain) is 0 by inspection.

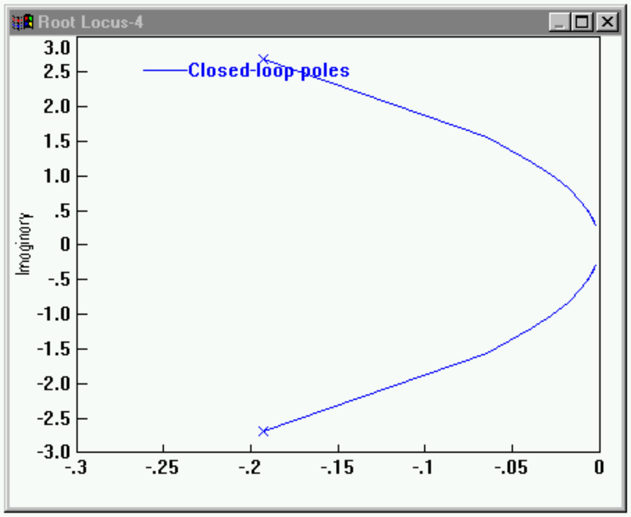
The second dialog box presents the factors of both polynomials. The zeros are the roots of the numerator, and the poles are the roots of the denominator. At this operating point, the system has two real zeros and a complex conjugate pair of poles. The factored transfer function can be expressed as:

$$\frac{Y(s)}{R(s)} = \frac{(s+0)(s+0)}{\left(s-(-0.192-j2.68)\right)\left(s-(-0.192+j2.68)\right)}$$

To generate the root locus plot

1. Choose **Analyze > Root Locus**.

2. Resize and move the plot for easier viewing.

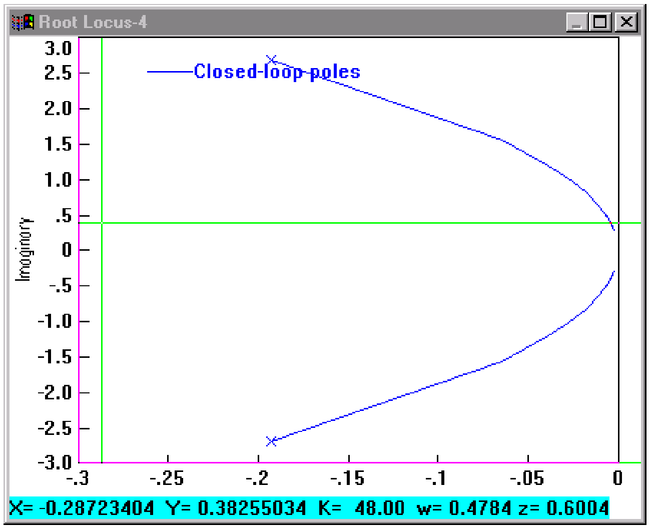


3. Choose **Edit > Block Properties**.

4. Click the mouse over the root locus plot.

5. Click on the **Read Coordinates** button.

The root locus plot reappears with crosshairs and status bar.



```
X= -0.28723404  Y= 0.38255034  K=  48.00  w= 0.4784 z= 0.6004
```

6. At the selected point, a gain of 48 in feedback around this transfer function results in a well-damped $(z \cong 0.6004)$ rapid responding system with:

   - A time constraint of

     $$\tau \approx \frac{1}{zw} = \frac{1}{(0.6004)(0.4784)} = \frac{1}{0.2872} = 3.48\ s$$

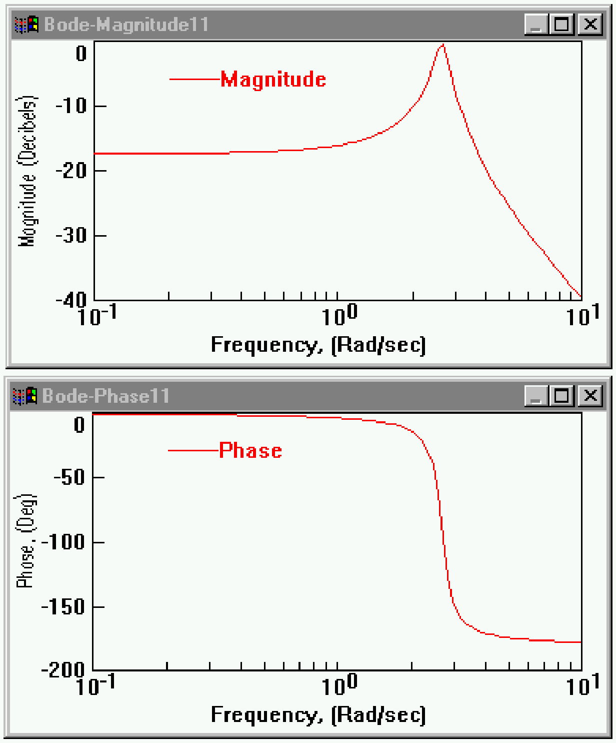
   - A zero steady state step error due to the integration at the origin.

To generate the Bode plots

Bode plots are frequently used to determine the performance characteristics of a closed-loop system in the frequency domain.

1. Choose **Analyze > Frequency Range**.
2. Do the following:
   - In the Start box, enter 0.1.
   - In the End box, enter 10.
   - In the Step Count box, enter 100.
3. Choose **Analyze > Frequency Response**.

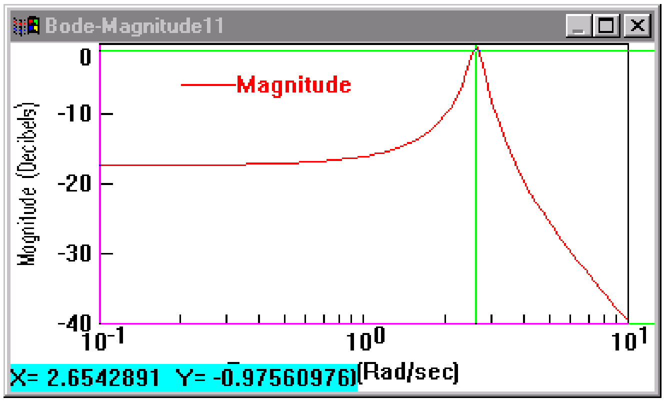
4. Resize and move the plots for easier viewing.



5. To determine the resonant frequency of the magnitude plot, invoke the plot crosshairs by doing the following:

- Choose **Edit > Block Properties**.
- Click the mouse over the Bode magnitude plot.
- Click on the **Options** tab.
- Click on the **Read Coordinates** button.

The Bode magnitude plot reappears with crosshairs and a status bar.



X= 2.6542891 Y= -0.97560976

Inspection of the digital display on the magnitude plot reveals the X coordinate is 2.654 rad/s, the resonant frequency of the system. Note that this value agrees, within the granularity of the digital read-out, with the factored transfer function value of 2.68 rad/s (as solved earlier).

Index