

Multirate signal processing	
decimate	Resample data at a lower sample rate
downsample	Downsample input signal
interp	Resample data at a higher sample rate
interp1	General 1-D interpolation (MATLAB Toolbox)
resample	Resample sequence with new sampling rate
spline	Cubic spline interpolation
upfirdn	Up sample, FIR filter, down sample
upsample	Upsample input signal

Power Spectral Density estimation	
periodogram	Periodogram method
pwelch	Welsh's method
pburg	Burg's method
pcov	Covariance method
peig	Eigenvector method
pmcov	Modified Covariance method
pmtm	Thomson multitaper method
pmusic	MUSIC method
pyulear	Yule-Walker AR Method
psdplot	Plot Power Spectral Density data

Transforms	
fft	Fast Fourier transform
fft2	2-D fast Fourier transform
fftshift	Swap vector halves
dftmtx	Discrete Fourier transform matrix
dct	Discrete cosine transform
hilbert	Discrete-time analytic signal via Hilbert transform
ifft	Inverse fast Fourier transform
ifft2	Inverse 2-D fast Fourier transform
idct	Inverse discrete cosine function

Statistics	
mean	Average or mean value of signal
std	Standard deviation
xcorr	Estimation of cross-correlation function
xcov	Estimation of cross-covariance function
corrmtx	Autocorrelation matrix

Graphical User Interfaces

- sptool** **Signal Processing Tool**
- Viewing signals
 - Applying filters
 - Viewing and creating spectra
- fdatool** **Filter Design and Analysis Tool**
- Filter design
 - Import filters
 - Filter Transformation
 - Filter realization
 - Export to workspace, SPTool or M-file
- fvtool** **Filter Visualization Tool** (linked to FDATool)
- Frequency response (magnitude, phase, group delay, phased delay)
 - Impulse and step response
 - Pole/zero plot
- wvtool** **Window Viewer Tool** (linked to FDATool and to WinTool)
- Time and frequency response
 - Leakage factor, relative sidelobe attenuation, mainlobe width
 - Export to workspace
- wintool** **Window Design and Analysis Tool**
- Design and analyze windows
 - Time and response
 - Export to workspace

Waveform generation	
rectpuls	Sampled aperiodic rectangle generator
sawtooth	Sawtooth function
sinc	Sinc or $\sin(\pi*x)/(\pi*x)$ function
square	Square wave function
chirp	Swept-frequency cosine generator
diric	Dirichlet (periodic sinc) function
pulstran	Pulse train generator
vco	Voltage controlled oscillator

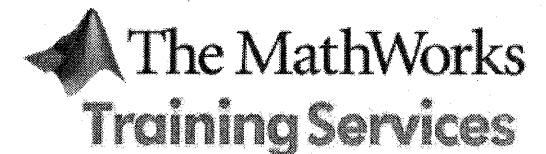
Signal visualization	
plot	Linear 2-D plot
stem	Plot discrete sequence data
stairs	Stairstep plot

MATLAB[®]

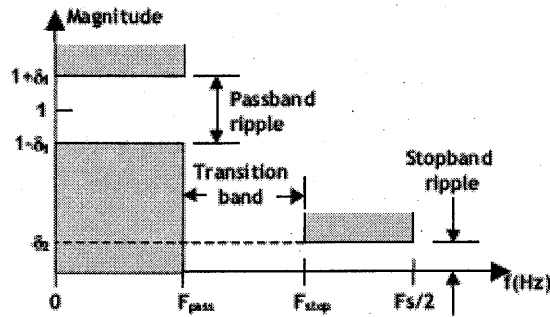
for

Signal Processing

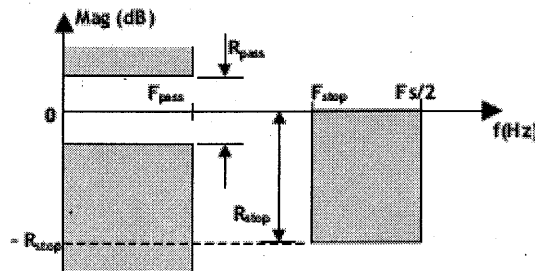
Quick Reference



Absolute filter specifications

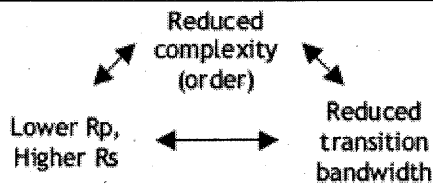


Relative filter specifications



Filter design tradeoffs

Finite Impulse Response (FIR)	Infinite Impulse Response (IIR)
<ul style="list-style-type: none"> Higher order Always stable Passband phase linear Remains stable when quantized 	<ul style="list-style-type: none"> Lower order Can be unstable Nonlinear phase Could become unstable when quantized



In MATLAB, type `doc functionname` for more information on any of these functions, or type `helpwin signal` for a complete list of available functions.

Filter and window objects

<code>dfilt</code>	Construct a discrete-time, filter object
<code>qfilt</code>	Construct a quantized filter
<code>sigwin</code>	Construct a window object
<code>methods</code>	Display method names

Filtering

<code>conv</code>	Convolution and polynomial multiplication
<code>filter</code>	Filter data with an IIR or FIR filter
<code>filtfilt</code>	Zero-phase digital filtering
<code>fftfilt</code>	FFT-based FIR filtering using the overlap-add method

Finite Impulse Response (FIR) filter design

<code>fir1</code>	Design a window-based finite impulse response filter
<code>fir2</code>	Frequency sampling-based finite impulse response filter
<code>firpm</code>	Parks-McClellan optimal FIR filter design (equiripple)
<code>fircls</code> , <code>fircls1</code>	Constrained least square FIR filter design
<code>firrcos</code>	Raised cosine FIR filter design
<code>cfirpm</code>	Complex and nonlinear-phase equiripple FIR filter design
<code>window</code>	Compute a specific window (eg hamming)

Infinite Impulse Response (IIR) filter design

<code>butter</code>	Butterworth analog and digital filter design
<code>cheby1</code>	Chebyshev Type I analog and digital filter design (passband ripple)
<code>cheby2</code>	Chebyshev Type II analog and digital filter design (stopband ripple)
<code>ellip</code>	Elliptic (Cauer) analog and digital filter design
<code>yulewalk</code>	Yule-Walker filter design
<code>besself</code>	Bessel analog filter design
<code>bilinear</code>	Bilinear transformation method for analog-to-digital filter conversion
<code>impinvar</code>	Impulse invariance method for analog-to-digital filter conversion
<code>lp2bp</code> , <code>lp2bs</code> , <code>lp2hp</code> , <code>lp2lp</code>	Filter type transformations

Filter analysis

<code>abs</code>	Magnitude
<code>angle</code>	Phase angle
<code>freqs</code>	Laplace transform frequency response
<code>freqspace</code>	Frequency spacing for frequency response
<code>freqz</code>	Z-transform frequency response
<code>grpdelay</code>	Group delay
<code>impz</code>	Discrete impulse response
<code>phasez</code>	Digital filter phase response
<code>phasedelay</code>	Phase delay of a digital filter
<code>unwrap</code>	Unwrap phase
<code>zerophase</code>	Zero-phase response of a real filter
<code>zplane</code>	Discrete pole-zero plot
<code>fvtool</code>	Filter visualization tool

Window Design

<code>window</code>	Compute a specific window
<code>bartlett</code>	Bartlett window
<code>blackman</code>	Blackman window
<code>chebwin</code>	Chebyshev window
<code>hamming</code>	Hamming window
<code>hann</code>	Hann (Hanning) window
<code>kaiser</code>	Kaiser window
<code>rectwin</code>	Rectangular window
<code>triang</code>	Triangular window

In the MATLAB Help browser browse to *Signal Processing Toolbox: Special Topics: Windows* for more information on windowing and window functions.

Filter order estimation

<code>buttord</code>	Butterworth filter order estimation
<code>cheb1ord</code>	Chebyshev Type I filter order estimation
<code>cheb2ord</code>	Chebyshev Type II filter order estimation
<code>ellipord</code>	Elliptic filter order estimation
<code>kaiserord</code>	Kaiser window design based filter order estimation
<code>remezord</code>	Remez design based filter order estimation

Working with the Scope Block

Zoom Controls

Zoom in x, zoom in y, autoscale (binoculars icon) for full view

General Properties

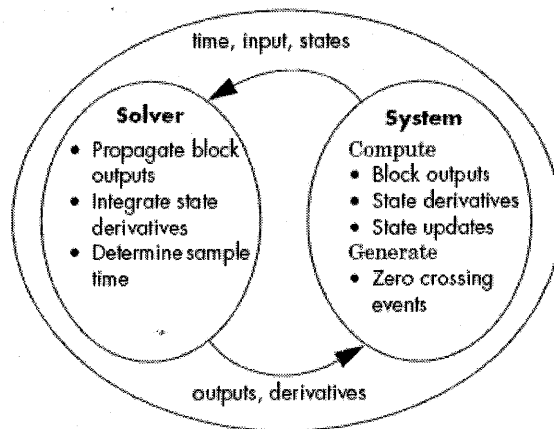
axes = # input ports

Right-click individual axes to set axes properties

Displaying Vector Signals

When displaying a vector signal, the Scope block uses colors in this order: yellow, magenta, cyan, red, green, and dark blue. The Scope Viewer has a slightly different color order: dark blue, red, magenta, cyan, yellow, and green. When more than six signals are displayed, both the Scope block and Scope Viewer cycle through the colors in the order listed above.

Solver-System Interaction



Overview of State Equations in S-Functions

$$x_d(k+1) = f_d(x(k), u(k), t_k) \rightarrow \text{mdlUpdate}$$

$$\dot{x}(t) = f_c(x, u, t) \rightarrow \text{mdlDerivatives}$$

$$y(t) = g(x, u, t) \rightarrow \text{mdlOutput}$$

Sample Time Colors

Color	Use
Black	Continuous blocks
Magenta	Constant blocks
Yellow	Hybrid (subsystems, grouping blocks, or Mux or Demux blocks grouping signals with varying sample times)
Red	Fastest discrete sample time
Green	Second fastest discrete sample time
Blue	Third fastest discrete sample time
Light Blue	Fourth fastest discrete sample time
Dark Green	Fifth fastest discrete sample time
Orange	Sixth fastest discrete sample time
Cyan	Blocks in triggered subsystems
Gray	Fixed in minor step

For More Information

MathWorks Web Site

<http://www.mathworks.com>

MathWorks FTP Server

<ftp://ftp.mathworks.com>

MathWorks User Community

<http://www.mathworks.com/matlabcentral/>
Includes newsgroup and file exchange

MathWorks Technical Support

Phone: (508) 647-7000

Contact support:

<http://www.mathworks.com/support/>, select

Contact Support (upper right)

Solution search: At

<http://www.mathworks.com/support/>, select

Bug Reports, Solutions, Tech Notes under Search.


MathWorks Training Services

Phone: (508) 647-7468

E-mail address: training@mathworks.com

Simulink[®]

Quick Reference



The MathWorks
Training Services

Abbreviations Used in Tables

MB is mouse button, L = left, R = right, C = center

Mouse and Keyboard Actions for Blocks

Task	Mouse and Keyboard Actions
Select one block	LMB
Select multiple blocks	Shift + LMB; on UNIX, CMB alone
Select next block	Tab
Select previous block	Shift + Tab
Copy block from another window	Drag block
Move block	Drag block
Duplicate block	Ctrl + LMB + drag block; or RMB + drag block
Connect blocks	LMB
Disconnect block	Shift + drag block; or CMB + drag block
Open selected subsystem	Enter
Go to parent of selected subsystem	Esc

Mouse and Keyboard Actions for Lines

Task	Mouse and Keyboard Actions
Select one line	LMB
Select multiple lines	Shift + LMB; on UNIX, CMB alone
Draw branch line	Ctrl + drag line; or RMB + drag line
Route lines around blocks	Shift + draw line segments; on UNIX, CMB + draw segments
Move line segment	Drag segment
Move vertex	Drag vertex
Create line segments	Shift + drag line; on UNIX, CMB + drag line

Mouse and Keyboard Actions for Signal Labels

Task	Mouse and Keyboard Actions
Create signal label	Double-click line, then type label
Copy signal label	Ctrl + drag label
Move signal label	Drag label
Edit signal label	Click in label, then edit
Delete signal label	Shift + click label, then Delete
Propagate label from previous signal(s)	Click in label, type '<', then click elsewhere in model

Mouse and Keyboard Actions for Annotations

Task	Mouse and Keyboard Actions
Create annotation	Double-click in diagram, then type text
Copy annotation	Ctrl + drag label
Move annotation	Drag label
Edit annotation	Click in text, then edit
Delete annotation	Shift + select annotation, then press Delete

Keyboard Shortcuts: Blocks

Task	Shortcut
Cut block	Ctrl + X
Copy block	Ctrl + C; or RMB + drag to copy and paste
Flip selected block	Ctrl + I
Paste block	Ctrl + V
Rotate selected block clockwise	Ctrl + R

Keyboard Shortcuts: Diagrams

Task	Shortcut
Create subsystem	Ctrl + G
Mask subsystem	Ctrl + M
Look under mask	Ctrl + U
Start/stop simulation	Ctrl + T
Update diagram	Ctrl + D
Zoom out	V
Zoom in	R
Zoom-fit selection to view	F
Zoom-fit model to view	Space
Pan model view	P or Q + LMB

Simulink Functions

Function	Description
dlinmod	Extract linear model from discrete-time system
get_param	Get system and block parameter values
linmod	Extract linear model from continuous-time system
linmod2	Extract linear model, advanced method
set_param	Set Simulink system and block parameters (including callbacks)
sim	Simulate a Simulink model
simget	Get options structure
simplot	Plot simulation data in a figure window
simset	Define options structure for sim
[sizes, x0, xord] = modelname	Get size and state information
sldebug	Debug a Simulink model
trim	Find steady-state operating point

Creating Data for Simulink

General Form

First column: time

Following columns: signal value

Simulink uses linear interpolation for undefined data points.

Piecewise Linear

```
t = 0:10;
u = [0 1 0 1 0 1 0 1 0 1 0];
sim_data = [t',u'];
```

Function of Time

```
t = 0:time_step:final_time;
u = func(t);
sim_data = [t',u'];
```

Sharp Edge

Define the same point twice.

Axis Control	
axes	Create axes in arbitrary positions
axis	Control axis scaling and appearance
box	Display axis border
grid	Control grid lines
hold	Hold current graph
rotate3d	Interactively rotate view of 3-D plot
subplot	Create axes in tiled positions
view	Set three-dimensional graph viewpoint specifications
zoom	Zoom in and out on 2-D plot

Lighting	
diffuse	Reflectance for a surface
lighting	Lighting mode
material	Material reflectance mode
specular	Specular reflectance
surfl	Three-dimensional shaded surface with lighting
surfnorm	Surface normals

Type helpbrowser to find more information on available Handle Graphics® properties.

File Input/Output	
dlmread dlmwrite	Read/write ASCII delimited file
fopen	Open generic text or binary file
fprintf	Write to generic text file
fread	Read generic binary file
fscanf	Read generic text file
fwrite	Write to generic binary file
importdata	Load data from file
imread imwrite	Read/write image
load	Load MAT-file
save	Save variables to MAT-file
textread	Read formatted data from text file
uiimport	Open Import Wizard to load data
wavread wavwrite	Read/write Microsoft WAV-file
wk1read wk1write	Read/write spreadsheet file
xlsread	Read data from Microsoft Excel file
fgetl	Read line from file, discard return

Programming	
eval	Evaluate string as MATLAB expression
function	Keyword to create function M-file, e.g., function[out1,out2]=myfile(in1,in2)
nargin nargout	Number of input/output arguments to M-file
pcode	Create pseudocode of M-file
profile viewer	Create profile report of your M-file's performance
varargin varargout	Variable number of input/output arguments
F10	Step through debugging

Sample Function M-File	
mystats.m	<pre>function [mean,size] = mystats(x) % Comments: This is the M-file help. % Type this test in an editor; save % as mystats.m. % Typing help mystats displays these % comments. % From the MATLAB command line, call % this function as % [mn,data_size] = mystats([1 2 3]) % Actual computations done below. if nargin == 1 mean = sum(x)/length(x) size = length(x) else error('Please enter input vector') end</pre>

Commonly Used Commands	
get	Obtain properties of a graphical object
set	Specify properties of a graphical object
propedit	Open the Property Editor for viewing and editing properties of graphical objects
demo	View listing of available demonstrations
guide	Open graphical user interface building tool
image	Display image
polyfit	Fit polynomial to data
sound	Play sound

MATLAB®

Quick Reference



The MathWorks
Training Services

In MATLAB, type `doc functionname` for more information on any of these functions, or type `help` for a complete list of available functions.

Operators and Special Characters	
<code>+, -, *, /</code>	Matrix math operators
<code>.*, ./</code>	Array multiplication and division
<code>^, .^</code>	Matrix and array power
<code>\</code>	Left division or linear optimization
<code>:</code>	Create linearly spaced vectors (e.g., <code>A=1:.01:10</code> , <code>A(90:end)</code>)
<code>()</code>	Index into matrix (e.g., <code>A(3,2)</code> pulls the third row, second column of A), and enclosed function input arguments
<code>[]</code>	Create matrix (e.g., <code>A = [1 2; 3 4]</code> , <code>x=[1 2 3]</code>)
<code>{ }</code>	Create and index cell array
<code>.</code>	Decimal point, or in a variable, denotes a structure (e.g., <code>user.Name = 'Paul'</code>)
<code>...</code>	Line continuation
<code>,</code>	Separate commands or elements in matrix
<code>;</code>	Denote new row in matrix definition, or at the end of command, suppress output (e.g., <code>A=[1;3]</code> or <code>x=5;</code>)
<code>%</code>	Comment
<code>!</code>	Call the operating system
<code>\</code>	Transpose, or quote for defining strings
<code>.'</code>	Nonconjugated transpose (for complex numbers)
<code>=</code>	Variable assignment
<code>==</code>	Equality
<code><>, <=, >=</code>	Relational operators
<code>&, , ~, xor</code>	Logical AND, OR, NOT and XOR

Starting and Quitting MATLAB	
<code>finish</code>	MATLAB finish M-file
<code>matlabrc</code>	MATLAB startup M-file for system administrator or single-user system
<code>quit</code>	Terminate MATLAB
<code>startup</code>	MATLAB startup M-file for each user

Managing Commands and Functions	
<code>addpath</code>	Add/remove directories from MATLAB search path
<code>rmpath</code>	Clear the command line
<code>clc</code>	Clear the command line
<code>doc</code>	Show documentation for functions
<code>edit</code>	Open MATLAB Editor/Debugger
<code>help</code> or <code>helpwin</code>	Display help for MATLAB functions and M-files
<code>helpbrowser</code>	Find and display documentation
<code>lookfor</code>	Keyword search through M-file help
<code>path</code>	Control MATLAB directory search path
<code>type</code>	List file
<code>version</code>	MATLAB version number
<code>what</code>	Directory of M-files, MAT-files, and MEX-files
<code>which</code>	Locate functions and files

Managing Variables and the Workspace	
<code>clear</code>	Remove items from memory
<code>disp</code>	Display text or array
<code>length</code>	Length of vector
<code>load</code>	Retrieve variables from disk
<code>pack</code>	Consolidate workspace memory
<code>save</code>	Save workspace variables on disk
<code>size</code>	Array dimensions
<code>who, whos</code>	List directory of variables in memory

Elementary X-Y Graphs	
<code>loglog</code>	Log-log scale plot
<code>plot</code>	Linear plot
<code>plotyy</code>	Graphs with y tick labels on the left and right
<code>polar</code>	Polar coordinate plot
<code>semilogx</code>	Semi-log scale plot for x- and y-axes
<code>semilogy</code>	

LinSpec Arguments for plot	
Line Style	
<code>-</code>	Solid line (default)
<code>- -</code>	Dashed line
<code>:</code>	Dotted line
<code>- .</code>	Dash-dot line
Marker	
<code>+</code>	Plus sign
<code>o</code>	Circle
<code>*</code>	Asterisk
<code>.</code>	Point
<code>x</code>	Cross
<code>s</code>	Square
<code>d</code>	Diamond
<code>^</code>	Upward pointing triangle
<code>v</code>	Downward pointing triangle
<code>></code>	Right pointing triangle
<code><</code>	Left pointing triangle
<code>p</code>	Five-pointed star (pentagon)
<code>h</code>	Six-pointed star (hexagon)
Color	
<code>r</code>	Red
<code>g</code>	Green
<code>b</code>	Blue
<code>c</code>	Cyan
<code>m</code>	Magenta
<code>y</code>	Yellow
<code>k</code>	Black
<code>w</code>	White

Working with the Scope Block

Zoom Controls

Zoom in x, zoom in y, autoscale (binoculars icon) for full view

General Properties

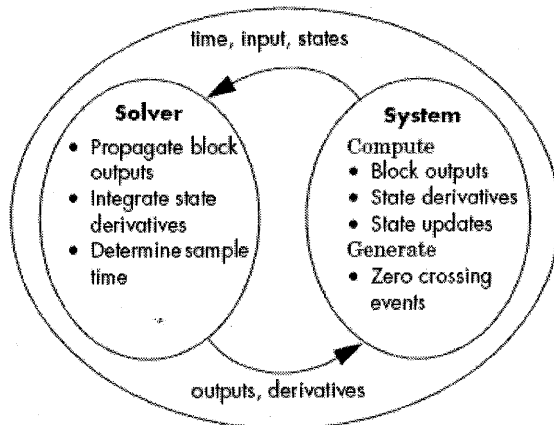
axes = # input ports

Right-click individual axes to set axes properties

Displaying Vector Signals

When displaying a vector signal, the Scope block uses colors in this order: yellow, magenta, cyan, red, green, and dark blue. The Scope Viewer has a slightly different color order: dark blue, red, magenta, cyan, yellow, and green. When more than six signals are displayed, both the Scope block and Scope Viewer cycle through the colors in the order listed above.

Solver-System Interaction



Overview of State Equations in S-Functions

$$x_d(k+1) = f_d(x(k), u(k), t_k) \rightarrow \text{mdlUpdate}$$

$$\dot{x}(t) = f_c(x, u, t) \rightarrow \text{mdlDerivatives}$$

$$y(t) = g(x, u, t) \rightarrow \text{mdlOutput}$$

Sample Time Colors

Color	Use
Black	Continuous blocks
Magenta	Constant blocks
Yellow	Hybrid (subsystems, grouping blocks, or Mux or Demux blocks grouping signals with varying sample times)
Red	Fastest discrete sample time
Green	Second fastest discrete sample time
Blue	Third fastest discrete sample time
Light Blue	Fourth fastest discrete sample time
Dark Green	Fifth fastest discrete sample time
Orange	Sixth fastest discrete sample time
Cyan	Blocks in triggered subsystems
Gray	Fixed in minor step

For More Information

MathWorks Web Site

<http://www.mathworks.com>

MathWorks FTP Server

<ftp://ftp.mathworks.com>

MathWorks User Community

<http://www.mathworks.com/matlabcentral/>

Includes newsgroup and file exchange

MathWorks Technical Support

Phone: (508) 647-7000

Contact support:

<http://www.mathworks.com/support/>, select

Contact Support (upper right)

Solution search: At

<http://www.mathworks.com/support/>, select

Bug Reports, Solutions, Tech Notes under Search.

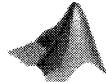
MathWorks Training Services

Phone: (508) 647-7468

E-mail address: training@mathworks.com

Simulink®

Quick Reference



The MathWorks
Training Services

Abbreviations Used in Tables

MB is mouse button, L = left, R = right, C = center

Mouse and Keyboard Actions for Blocks

Task	Mouse and Keyboard Actions
Select one block	LMB
Select multiple blocks	Shift + LMB; on UNIX, CMB alone
Select next block	Tab
Select previous block	Shift + Tab
Copy block from another window	Drag block
Move block	Drag block
Duplicate block	Ctrl + LMB + drag block; or RMB + drag block
Connect blocks	LMB
Disconnect block	Shift + drag block; or CMB + drag block
Open selected subsystem	Enter
Go to parent of selected subsystem	Esc

Mouse and Keyboard Actions for Lines

Task	Mouse and Keyboard Actions
Select one line	LMB
Select multiple lines	Shift + LMB; on UNIX, CMB alone
Draw branch line	Ctrl + drag line; or RMB + drag line
Route lines around blocks	Shift + draw line segments; on UNIX, CMB + draw segments
Move line segment	Drag segment
Move vertex	Drag vertex
Create line segments	Shift + drag line; on UNIX, CMB + drag line

Mouse and Keyboard Actions for Signal Labels

Task	Mouse and Keyboard Actions
Create signal label	Double-click line, then type label
Copy signal label	Ctrl + drag label
Move signal label	Drag label
Edit signal label	Click in label, then edit
Delete signal label	Shift + click label, then Delete
Propagate label from previous signal(s)	Click in label, type '<', then click elsewhere in model

Mouse and Keyboard Actions for Annotations

Task	Mouse and Keyboard Actions
Create annotation	Double-click in diagram, then type text
Copy annotation	Ctrl + drag label
Move annotation	Drag label
Edit annotation	Click in text, then edit
Delete annotation	Shift + select annotation, then press Delete

Keyboard Shortcuts: Blocks

Task	Shortcut
Cut block	Ctrl + X
Copy block	Ctrl + C; or RMB + drag to copy and paste
Flip selected block	Ctrl + I
Paste block	Ctrl + V
Rotate selected block clockwise	Ctrl + R

Keyboard Shortcuts: Diagrams

Task	Shortcut
Create subsystem	Ctrl + G
Mask subsystem	Ctrl + M
Look under mask	Ctrl + U
Start/stop simulation	Ctrl + T
Update diagram	Ctrl + D
Zoom out	V
Zoom in	R
Zoom-fit selection to view	F
Zoom-fit model to view	Space
Pan model view	P or Q + LMB

Simulink Functions

Function	Description
dlinmod	Extract linear model from discrete-time system
get_param	Get system and block parameter values
linmod	Extract linear model from continuous-time system
linmod2	Extract linear model, advanced method
set_param	Set Simulink system and block parameters (including callbacks)
sim	Simulate a Simulink model
simget	Get options structure
simplot	Plot simulation data in a figure window
simset	Define options structure for sim
[sizes, x0, xord] = modelname	Get size and state information
sldebug	Debug a Simulink model
trim	Find steady-state operating point

Creating Data for Simulink

General Form

First column: time

Following columns: signal value

Simulink uses linear interpolation for undefined data points.

Piecewise Linear

```
t = 0:10;
u = [0 1 0 1 0 1 0 1 0 1 0];
sim_data = [t', u'];
```

Function of Time

```
t = 0:time_step:final_time;
u = func(t);
sim_data = [t', u'];
```

Sharp Edge

Define the same point twice.