MATLAB BASIC CHEAT SHEET

Basics

`clc`	Clear command window
`clear`	Clear all variables
`close all`	Close all plots
`help function`	Print help page for function
`% This is a comment`	Comments
`ctrl-c`	Abort the current operation
`format short`	Display 4 decimal places
`format long`	Display 15 decimal places

Defining and Changing Variables

`a = 3`	Define variable a to be 3
`x = [1, 2, 3]`	Set x to be the row vector $[1, 2, 3]$
`x = [1; 2; 3]`	Set x to be the column vector $[1, 2, 3]^T$

`A = [1, 2, 3, 4;`
`5, 6, 7, 8;`
`9, 10, 11, 12]`
	Set A to be a 3×4 matrix

`x(2) = 7`	Change x from $[1, 2, 3]$ to $[1, 7, 3]$
`A(2,1) = 0`	Change $A_{2,1}$ from 5 to 0
`syms x`	Define variable x to be symbolic
`double(x)`	Convert x from symbolic to double
`subs(x)`	Replace all variables in the symbolic expression of x with their values taken from the MATLAB workspace

Constants

`pi`	$\pi = 3.141592653589793$
`NaN`	Not a number (i.e. $0/0$)
`Inf`	Infinity

Basic Arithmetic and Trigonometric Functions

`3*4, 7+4, 2-6, 8/3`	multiply, add, subtract and divide
`3^7`	Compute 3^7
`sqrt(5)`	Compute $\sqrt{5}$
`log(3)`	Compute $\ln(3)$
`log10(100)`	Compute $\log_{10}(100)$
`abs(-5)`	Compute $\|-5\|$
`sin(5*pi/3)`	Compute $\sin(5\pi/3)$, angle expressed in rad
`cos(pi/2)`	Compute $\cos(\pi/2)$, angle expressed in rad
`tan(pi/4)`	Compute $\tan(\pi/2)$, angle expressed in rad
`asin(0.5)`	Compute $\arcsin(0.5)$, result expressed in rad
`atan(3)`	Compute $\arctan(\pi/2)$, result in rad
`atan2(2,3)`	Compute $\arctan(2/3)$, result $\in [-\pi, \pi]$
`sind(300)`	Compute $\sin(300)$, angle expressed in deg
`cosd(90)`	Compute $\cos(90)$, angle expressed in deg

Complex Numbers (either numeric or symbolic)

`a = 4+i*3`	define a with 4 as the real part and 3 as the imaginary part
`abs(a)`	compute $\|a\|$ (i.e. $\sqrt{4^2 + 3^2}$)
`angle(a)`	compute $\arg(a)$ (i.e. `atan2(3,4)`)

Constructing Matrices and Vectors

`zeros(12, 5)`	Make a 12×5 matrix of zeros
`ones(12, 5)`	Make a 12×5 matrix of ones
`eye(5)`	Make a 5×5 identity matrix
`linspace(0, 50, 1000)`	Make a vector with 1000 elements evenly spaced between 0 and 50
`0:10`	Row vector of $0, 1, \ldots, 9, 10$
`0:0.001:50`	Row vector of elements from 0 to 50 with 0.001 step

Operations on Matrices and Vectors

`3 * x`	Multiply every element of x by 3
`x + 2`	Add 2 to every element of x
`x + y`	Element-wise addition of two vectors x and y
`A * y`	Product of a matrix and vector
`A * B`	Product of two matrices
`A ^ 3`	Square matrix A to the third power
`A .^ 3`	Every element of A to the third power
`exp(A)`	Compute the exponential of every element of A
`expm(A)`	Compute the exponential matrix of A (i.e. e^A)
`abs(A)`	Compute the absolute values of every element of A
`A'`	Transpose of A
`inv(A)`	Compute the inverse of A
`det(A)`	Compute the determinant of A
`eig(A)`	Compute the eigenvalues of A
`rank(A)`	Compute the rank of A

Entries of Matrices and Vectors

`x(2:12)`	The 2nd to the 12th elements of x
`x(2:end)`	The 2nd to the last elements of x
`x(1:3:end)`	Every third element of x from the first to last
`A(5,:)`	Get the 5th row of A
`A(:,5)`	Get the 5th column of A
`A(5, 1:3)`	Get the first to third elements in the 5th row

Plotting

`plot(x,y)`	Plot y versus x (must be the same length)
`loglog(x,y)`	Plot y versus x on a log-log scale (both axes have a logarithmic scale)
`semilogx(x, y)`	Plot y versus x with x on a log scale
`semilogy(x, y)`	Plot y versus x with y on a log scale
`axis equal`	Force the x and y axes to be scaled equally
`grid on`	Add a grid to the plot
`hold on`	Multiple plots on single figure
`figure`	Start a new plot

Equations and Polynomials

`coeffs(x^2*x-1)`	Return the coefficients of a polynomial symbolic expression
`solve(x^2+x==0)`	Compute the solution of a symbolic equation
`solve([x1+x2==0;x2+6==0])`	Compute the solution of a system of symbolic equations

MATLAB SYMBOLIC AND CONTROL SYSTEM TOOLBOXES

Laplace Transform (Symbolic)

`dirac(t)`	Dirac impulse
`heaviside(t)`	Heaviside step function
`laplace(f(t))`	Compute the Laplace transform of a symbolic expression
`ilaplace(G(s))`	Compute the inverse Laplace transform of a symbolic expression

LTI Systems and Transfer Functions

`ctrb(A,B)`	Compute the controllability matrix
`obsv(A,C)`	Compute the observability matrix
`ss(A,B,C,D)`	Get the state-space representation of a LTI system
`tf(num,den)`	Get the transfer function given the coefficients of numerator and denominator
`tf(sys)`	Get the transfer function given the ss representation of a system
`[Num,Den]=ss2tf(A,B,C,D)`	Compute the coefficients of the transfer function of the A,B,C,D system
`s=tf('s')`	Define the laplace s variable as a transfer function
`pole(G)`	Compute the poles of the transfer function G
`zero(G)`	Compute the zeroes of the transfer function G
`dcgain(G)`	Compute the DC gain of the transfer function G (i.e. gain at zero frequency)
`damp(G)`	Print poles, natural frequencies and damping factors of G

Connected Systems and Responses

`parallel(G1,G2)`	Return the G1-G2 parallel connection
`series(G1,G2)`	Return the G1-G2 series connection
`feedback(G1,G2)`	Return the G1-G2 negative feedback connection
`impulse(G)`	Plot the impulse response of G
`impulse(G,t)`	Plot the impulse response of G in time t (t is a sampled vector)
`y=impulse(G)`	Return the impulse response of G as a column vector
`step(G)`	Plot the step response of G
`step(G,t)`	Plot the step response of G in time t (t is a sampled vector)
`y=step(G)`	Return the step response of G as a column vector
`step(G,popt)`	Plot the step response of G with specified time options (popt = timeoptions)
`stepinfo(G)`	Print the characteristics of the step response of G

PID Controllers (standard form)

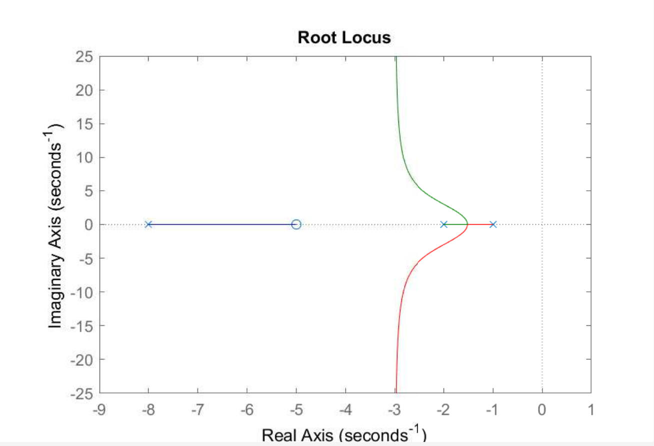
`pidstd(Kp)`	Return a P controller
`pidstd(Kp,Ti)`	Return a PI controller
`pidstd(Kp,Inf,Td)`	Return a PD controller
`pidstd(Kp,Ti,Td)`	Return a PID controller

Root locus and Bode Plots

`rlocus(G)`	Plot the root locus of the transfer function G
`rlocus(G,k)`	Plot the root locus of G with given k values (k is a vector)
`bode(G)`	Plot the Bode diagrams of G (amplitude and phase)
`bode(G,w)`	Plot the Bode diagrams of G at given frequencies w (w is a vector)
`margin(G)`	Plot the Bode diagrams of G specifying the stability margin
`[Gm,Pm,Wpi,Wc]=margin(G)`	Return the gain (Gm) and phase (Pm) margins and the cross frequencies Wpi and Wc

Example: Root Locus

```
s = tf('s'); % Laplace variable
G = (s+5)/((s+1)*(s+2)*(s+8)); % Transfer function
rlocus(G)   % Plot the root locus of G
```



Example: Bode

```
s = tf('s'); % Laplace variable
G = (s+5)/((s+1)*(s+2)*(s+8)); % Transfer function
bode(G)   % Amplitude and phase Bode diagrams
```