

Comandos e Funções do MATLAB

<u>Comando</u>	<u>Descrição</u>
clear	limpa a área de trabalho (ou clear all)
close	fecha figura (ou close all)
who	lista as variáveis que estão na memória atual
whos	lista as variáveis atuais e seus tamanhos de memória
abs	calcula o valor absoluto
exp	calcula a exponencial na base e
sqrt	calcula a raiz quadrada
log	calcula o logaritmo natural
log10	calcula o logaritmo na base 10
sin	calcula o seno
asin	calcula o arco seno
cos	calcula o co-seno
acos	calcula o arco co-seno
tan	calcula a tangente
atan	calcula o arco tangente (2 quadrantes)
atan2	calcula o arco tangente (4 quadrantes)
ans	variável criada automaticamente como resposta para expressões
sym	cria números, variáveis e objetos simbólicos, ex.: $x = \text{sym}('x')$
syms x y z	define as variáveis simbólicas x y z (1)
pzmap	esboça o diagrama de polos e zeros de um sistema linear
rlocus	gera o lugar das raízes, ex.: <code>rlocus(num,den)</code>
sgrid	adiciona linhas de grade no <i>root locus</i> ou mapa de polos e zeros
bode	gera os diagramas logarítmicos de Bode, ex.: <code>bode(num,den)</code>
nyquist	gera o diagrama polar de Nyquist
margin	calcula a margem de ganho e a margem de fase
nichols	gera o gráfico em coordenadas de Nichols
roots	acha as raízes de um polinômio, ex.: <code>roots([1 -6 11 -6])</code>
poly	converte raízes em polinômio, ex.: <code>poly([0 -2 -1.3])</code>
polyval	calcula o valor de um polinômio
polyfit(x,y,n)	determina os coeficientes de um polinômio $p(x)$ de grau “n” que representa os dados $p(x(i)) \sim y(i)$, usando o método dos mínimos quadrados do erro
conv	multipla (convolução) polinômios, ex.: <code>conv([1 2],[1 3 5])</code>
plot	gera um gráfico, ex.: <code>plot(t,y)</code> , onde $t=0:0.05:4\pi$ e $y=3\sin(t)$
title	adiciona um título ao gráfico atual, ex.: <code>title('Resposta ao degrau')</code>
text	adiciona texto ao gráfico atual
xlabel	adiciona um rótulo ao eixo x do gráfico
ylabel	adiciona um rótulo ao eixo y do gráfico
grid	adiciona linhas de grade ao gráfico (ou <code>grid on</code> / <code>grid off</code>)
hold	mantém o gráfico atual na janela gráfica (ou <code>hold on</code>)

<u>Comando</u>	<u>Descrição</u>
A'	matriz transposta conjugada de A
inv(A)	calcula a matriz inversa de A
eig(A)	calcula os autovalores da matriz A
[P,D]=eig(A)	calcula os autovetores (P) e autovalores (matriz diagonal D) da matriz A
det(A)	calcula o determinante da matriz A
CO=ctrb(A,B)	calcula a matriz de controlabilidade [B AB A^2B ...] ou CO=ctrb(sys)
OB=obsv(A,C)	calcula a matriz de observabilidade [C; CA; CA^2 ...] ou OB=obsv(sys)
rank(A)	determina o posto de uma matriz
expm(A)	calcula a exponencial matricial na base e, ex.: syms t; expm(A*t) ou A=[0 -2;1 -3];dt=0.2;phi=expm(A*dt)
eye	gera matriz identidade
cov	matriz de co-variância (se X é um vetor, COV(X) calcula a variância)
K=place(A,B,P)	calcula a matriz K de ganhos de realimentação onde os autovalores de A-B*K são especificados no vetor P. Nenhum autovalor deve ter multiplicidade maior do que o número de entradas.
K=acker(A,B,P)	calcula a matriz K de ganhos de realimentação onde um sistema com uma entrada $\dot{x} = A x + B u$ e com realimentação de $u = -K x$ tem polos de malha fechada especificados no vetor P, i.e., $P = \text{eig}(A-B*K)$.
residue	calcula a expansão em frações parciais, ex.: [r,p,k]=residue(num,den)
laplace	acha a transformada de Laplace, ex.: syms t a x; f = exp(-a*t); laplace(f,x)
ilaplace	acha a transformada inversa de Laplace. Ex.: syms a t; g = 1/(t-a)^2; ilaplace(g)
i ou j	$\sqrt{-1}$
nº complexo	x=4+3i ou x=4+3j ou x=complex(4,3) ou x=5*exp(0.6435i)
abs(x)	calcula o módulo de x
angle(x)	calcula o ângulo de x [rad] ou angle(x)*180/pi \Rightarrow ans=36.8699°
real(x)	calcula a parte real de um número complexo
imag(x)	calcula a parte imaginária de um número complexo
conj(x)	calcula o conjugado do número complexo x
pi	3.1416
inf	representa o infinito
max	determina o valor máximo
min	determina o valor mínimo
mean	determina o valor médio
median	determina o valor mediano
std	determina o desvio padrão
rand	gera números e matrizes randômicas
impulse	gera o gráfico de resposta ao impulso unitário
step	gera o gráfico de resposta ao degrau unitário
for	gera um laço, ex.: for i=1:n
format	define um formato de apresentação
end	finaliza uma estrutura de controle ou laço
exit	término do programa

Operadores

<code>==</code> \Rightarrow igual	<code>~=</code> \Rightarrow diferente
<code><</code> \Rightarrow menor que	<code>></code> \Rightarrow maior que
<code><=</code> \Rightarrow menor ou igual a	<code>>=</code> \Rightarrow maior ou igual a
<code>&</code> \Rightarrow and	<code> </code> \Rightarrow or
<code>+</code> \Rightarrow adição	<code>-</code> \Rightarrow subtração
<code>*</code> \Rightarrow multiplicação	<code>^</code> \Rightarrow potenciação
	<code>~</code> \Rightarrow not

Caracteres especiais

`[]` \Rightarrow usado para formação de vetores e matrizes
`()` \Rightarrow procede aos comandos e circunscreve expressões aritméticas
`,` \Rightarrow separa subscritos e argumentos de funções e comandos
`;` \Rightarrow encerra linhas e suprime impressões
`!` \Rightarrow força a execução de comandos do sistema operacional
`%` \Rightarrow introduz comentários

Conversão de Modelos

`sys = tf(num,den)` cria o sistema “sys” (contínuo no tempo) representado por função de transferência com o numerador “num” e o denominador “den”
`sys = ss(A,B,C,D)` cria o sistema “sys” (contínuo no tempo) representado no espaço de estados através das matrizes A,B,C,D
`[num,den] = ss2tf(A,B,C,D,iu)` converte da forma de variáveis de estado para função de transferência para a *iu*-ésima entrada
`[A,B,C,D] = tf2ss(num,den)` converte da forma de função de transferência para variáveis de estado (SISO)
`[z,p,k] = ss2zp(A,B,C,D,iu)` converte da forma de espaço de estados para zeros (z), polos (p) e ganho (k)
`[A,B,C,D] = zp2ss(z,p,k)` converte zeros (z), polos (p) e ganho (k) para a forma de espaço de estados
`[z,p,k] = tf2zp(num,den)` converte da forma de função de transferência para zeros (z), polos (p) e ganho (k)
`[num,den] = zp2tf(z,p,k)` converte zeros (z), polos (p) e ganho (k) para a forma de função de transferência

`sys = series(sys1,sys2)` \Rightarrow “sys” = conexão em série dos sistemas “sys1” e “sys2”
`sys = parallel(sys1,sys2)` \Rightarrow “sys” = conexão em paralelo dos sistemas “sys1” e “sys2”
`sys = feedback(sysg,sys_h)` \Rightarrow “sys” = conexão em realimentação dos sistemas “sysg” e “sys_h”. Para realimentação positiva: `sys = feedback(sysg,sys_h,+1)`

Funções para manipulação de expressões algébricas:

<code>diff(f)</code>	calcula a derivada de f
<code>compose(f,g)</code>	determina a composta f(g(x))
<code>expand(expr)</code>	expande uma expressão <i>expr</i>
<code>finverse(expr)</code>	determina a inversa funcional da expressão <i>expr</i>
<code>pretty(expr)</code>	exibe a expressão <i>expr</i> numa forma mais bonita
<code>simple(expr)</code>	procura encontrar uma forma mais simples de escrever a expressão <i>expr</i>
<code>simplify(expr)</code>	simplifica a expressão <i>expr</i> (1)

`solve(expr)` acha a(s) solução(es) da equação $expr = 0$
`subs(expr,x,a)` substitui na expressão $expr$ a variável x por a

`lsim(sys,u,t)` simula a resposta no tempo de modelos LIT para entradas arbitrárias.
Exemplo: `t = 0:0.01:5; u = sin(t); lsim(sys,u,t)`

`help` ajuda na descrição dos comandos e funções, ex.: `help bode`
`lookfor XYZ` procura o *string* XYZ na primeira linha dos *help's*

Exemplos:

(1) `>> syms x`
 `>> simplify((sin(x))^2+(cos(x))^2)`

Matlab Commands List

The following list of commands can be very useful for future reference. Use "help" in Matlab for more information on how to use the commands.

In these tutorials, we use commands both from Matlab and from the Control Systems Toolbox, as well as some commands/functions which we wrote ourselves. For those commands/functions which are not standard in Matlab, we give links to their descriptions. For more information on writing Matlab functions, see the [function](#) page.

Note: Matlab commands from the control system toolbox are highlighted in **red**.

Non-standard Matlab commands are highlighted in **green**.

Command	Description
abs	Absolute value
acker	Compute the K matrix to place the poles of A-BK, see also place
axis	Set the scale of the current plot, see also plot, figure
bode	Draw the Bode plot, see also logspace, margin , nyquist1
c2dm	Continuous system to discrete system
clf	Clear figure (use clg in Matlab 3.5)
conv	Convolution (useful for multiplying polynomials), see also deconv
ctrb	The controllability matrix, see also obsv
deconv	Deconvolution and polynomial division, see also conv
det	Find the determinant of a matrix
dimpulse	Impulse response of discrete-time linear systems, see also dstep
dlqr	Linear-quadratic regulator design for discrete-time systems, see also lqr
dlsim	Simulation of discrete-time linear systems, see also lsim
dstep	Step response of discrete-time linear systems, see also stairs
eig	Compute the eigenvalues of a matrix
eps	Matlab's numerical tolerance
feedback	Feedback connection of two systems.
figure	Create a new figure or redefine the current figure, see also subplot, axis
for	For, next loop
format	Number format (significant digits, exponents)
function	Creates function m-files
grid	Draw the grid lines on the current plot
gtext	Add a piece of text to the current plot, see also text
help	HELP!
hold	Hold the current graph, see also figure
if	Conditionally execute statements
imag	Returns the imaginary part of a complex number, see also real
impulse	Impulse response of continuous-time linear systems, see also step , lsim , dlsim

<code>input</code>	Prompt for user input
<code>inv</code>	Find the inverse of a matrix
<code>jgrid</code>	Generate grid lines of constant damping ratio (zeta) and settling time (sigma), see also <code>sgrid</code> , <code>sigrid</code> , <code>zgrid</code>
<code>legend</code>	Graph legend
<code>length</code>	Length of a vector, see also <code>size</code>
<code>linspace</code>	Returns a linearly spaced vector
<code>lnyquist1</code>	Produce a Nyquist plot on a logarithmic scale, see also <code>nyquist1</code>
<code>log</code>	natural logarithm, also <code>log10</code> : common logarithm
<code>loglog</code>	Plot using log-log scale, also <code>semilogx</code> / <code>semilogy</code>
<code>logspace</code>	Returns a logarithmically spaced vector
<code>lqr</code>	Linear quadratic regulator design for continuous systems, see also <code>dlqr</code>
<code>lsim</code>	Simulate a linear system, see also <code>step</code> , <code>impulse</code> , <code>dlsim</code> .
<code>margin</code>	Returns the gain margin, phase margin, and crossover frequencies, see also <code>bode</code>
<code>norm</code>	Norm of a vector
<code>nyquist1</code>	Draw the Nyquist plot, see also <code>lnyquist1</code> . Note this command was written to replace the Matlab standard command <code>nyquist</code> to get more accurate Nyquist plots.
<code>obsv</code>	The observability matrix, see also <code>ctrb</code>
<code>ones</code>	Returns a vector or matrix of ones, see also <code>zeros</code>
<code>place</code>	Compute the K matrix to place the poles of A-BK, see also <code>acker</code>
<code>plot</code>	Draw a plot, see also <code>figure</code> , <code>axis</code> , <code>subplot</code> .
<code>poly</code>	Returns the characteristic polynomial
<code>polyadd</code>	Add two different polynomials
<code>polyval</code>	Polynomial evaluation
<code>print</code>	Print the current plot (to a printer or postscript file)
<code>pzmap</code>	Pole-zero map of linear systems
<code>rank</code>	Find the number of linearly independent rows or columns of a matrix
<code>real</code>	Returns the real part of a complex number, see also <code>imag</code>
<code>rlocfind</code>	Find the value of k and the poles at the selected point
<code>rlocus</code>	Draw the root locus
<code>roots</code>	Find the roots of a polynomial
<code>rscale</code>	Find the scale factor for a full-state feedback system
<code>set</code>	Set(<code>gca</code> , <code>'Xtick'</code> , <code>xticks</code> , <code>'Ytick'</code> , <code>yticks</code>) to control the number and spacing of tick marks on the axes
<code>series</code>	Series interconnection of Linear time-independent systems
<code>sgrid</code>	Generate grid lines of constant damping ratio (zeta) and natural frequency (ω_n), see also <code>jgrid</code> , <code>sigrid</code> , <code>zgrid</code>
<code>sigrid</code>	Generate grid lines of constant settling time (sigma), see also <code>jgrid</code> , <code>sgrid</code> , <code>zgrid</code>
<code>size</code>	Gives the dimension of a vector or matrix, see also <code>length</code>
<code>sqrt</code>	Square root

ss	Create state-space models or convert LTI model to state space, see also tf
ss2tf	State-space to transfer function representation, see also tf2ss
ss2zp	State-space to pole-zero representation, see also zp2ss
stairs	Stairstep plot for discrete response, see also dstep
step	Plot the step response, see also impulse , lsim , dlsim .
subplot	Divide the plot window up into pieces, see also plot , figure
text	Add a piece of text to the current plot, see also title , xlabel , ylabel , gtext
tf	Creation of transfer functions or conversion to transfer function, see also ss
tf2ss	Transfer function to state-space representation, see also ss2tf
tf2zp	Transfer function to Pole-zero representation, see also zp2tf
title	Add a title to the current plot
wbw	Returns the bandwidth frequency given the damping ratio and the rise or settling time.
xlabel/ylabel	Add a label to the horizontal/vertical axis of the current plot, see also title , text , gtext
zeros	Returns a vector or matrix of zeros
zgrid	Generates grid lines of constant damping ratio (ζ) and natural frequency (ω_n), see also sgrid , jgrid , sigrid
zp2ss	Pole-zero to state-space representation, see also ss2zp
zp2tf	Pole-zero to transfer function representation, see also tf2zp