Daniel Galbes Bassanezi 22/04/2015

Laboratório de Controle 2

1)

'(1)' % Display label.

'How are you?' % Display string.

-3.96 % Display scalar number -3.96.

-4+7i % Display complex number -4+7i.

-5-6j % Display complex number -5-6j.

(-4+7i)+(-5-6i) % Add two complex numbers and display sum.

(-4+7j)\*(-5-6j) % Multiply two complex numbers and display product.

M=5 % Assign 5 to M and display.

N=6 % Assign 6 to N and display.

M+N

M\*N

Resultados:

ans = (1)

ans = How are you?

ans = -3.9600

ans = -4.0000 + 7.0000i

ans = -5.0000 - 6.0000i

ans = -9.0000 + 1.0000i

ans = 62.0000 -11.0000i

M = 5

N = 6

ans = 11

ans = 30

2)

% (10s^2+40s+60)/(s^3+4s^2+5s+7)

numftf=[10 40 60];

denftf=[1 4 5 7];

[z,p,k] = tf2zp(numftf,denftf);

[b,a] = zp2tf(z,p,k);

Resultados:

z =

-2.0000 + 1.4142i

-2.0000 - 1.4142i

p =

-3.1163 + 0.0000i

-0.4418 + 1.4321i

-0.4418 - 1.4321i

k = 10

b = 0 10 40 60

a = 1.0000 4.0000 5.0000 7.0000

3)

Fzpk1 = zpk([-2 -4],[0 -3 -5],10)

Ftf1 = tf(Fzpk1)

Fzpk2 = zpk(Ftf1)

Resultados:

Fzpk1 =

10 (s+2) (s+4)

--------------

s (s+3) (s+5)

Continuous-time zero/pole/gain model.

Ftf1 =

10 s^2 + 60 s + 80

------------------

s^3 + 8 s^2 + 15 s

Continuous-time transfer function.

Fzpk2 =

10 (s+4) (s+2)

--------------

s (s+5) (s+3)

Continuous-time zero/pole/gain model.

4)

numf = 150\*[1 2 7]; % Store 150(s^2+2s+7) in numf and % display.

denf = [1 5 4 0]; % Store s(s+1)(s+4) in denf and % display.

F = tf(numf,denf)

G = zpk(F)

s = tf('s')

F2 = tf(numf,denf)

s = zpk('s')

F3 = tf(numf,denf)

Resultados:

F =

150 s^2 + 300 s + 1050

----------------------

s^3 + 5 s^2 + 4 s

Continuous-time transfer function.

G =

150 (s^2 + 2s + 7)

------------------

s (s+4) (s+1)

Continuous-time zero/pole/gain model.

F2 =

150 s^2 + 300 s + 1050

----------------------

s^3 + 5 s^2 + 4 s

Continuous-time transfer function.

F3 =

150 s^2 + 300 s + 1050

----------------------

s^3 + 5 s^2 + 4 s

Continuous-time transfer function.

5)

syms s % Construct symbolic object for Laplace variable 's'.

F=2/[(s+1)\*(s+2)^2];

iF = ilaplace(F);

pretty(iF)

Resultados:

F =

2/((s + 1)\*(s + 2)^2)

iF =

2\*exp(-t) - 2\*exp(-2\*t) - 2\*t\*exp(-2\*t)

pretty(iF) =

2 exp(-t) - exp(-2 t) 2 - t exp(-2 t) 2

6)

%declarar variaveis

io=1

G=1

H=1

t= -2:0.01:10; %vetor de tempo, eixo x

q=size(t); %mede dimensões do vetor t

r=size(t); %mede dimensões do vetor t

f=zeros(q(1),q(2)); %seta f um vetor de zeros do tamanho de q ou t

ff=zeros(r(1),r(2)); %seta ff um vetor de zeros de mesmo tamanho

q=size(t(201:1201)); %modifica tamanho q para 1000

r=size(t(701:1201)); %modifica tamanho r para 500

f(201:1201)=ones(q(1),q(2)); %seta os ultimos 1000 pontos de f para 1

ff(701:1201)=ones(r(1),r(2));%seta os ultimos 500 pontos de ff para 1

rr=io\*(exp(-(G/H)\*t).\*f-(exp(-(G/H)\*(t-5))).\*ff);

%calcula uma expressao envolvendo f e ff que são pulsos começando em 200 e 700 e acabando em 1200

plot(t,rr) %plota a expressao calculada

title('Fig.1.7a Unit step function'); %título da figura

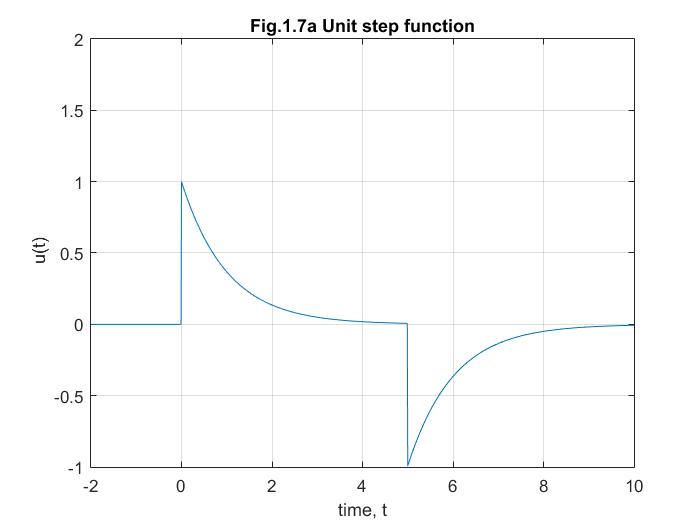
axis([-2,10,-1,2]); % limita eixos x e y

xlabel('time, t'); %legenda eixo x

ylabel(' u(t)'); %legenda eixo y

grid; %grade

Resultados:



7)

% making use of MATLAB's Symbolic Math Toolbox for simplicity and readability.

syms s

G=54\*(s+27)\*(s^3+52\*s^2+37\*s+73)/(s\*(s^4+872\*s^3+437\*s^2+89\*s+65)\*(s^2+79\*s+36));

[numg,deng]=numden(G)

numpoly = sym2poly(numg)

denpoly = sym2poly(deng)

pretty(G)

Resultados:

numg =

54\*(s + 27)\*(s^3 + 52\*s^2 + 37\*s + 73)

deng =

s\*(s^2 + 79\*s + 36)\*(s^4 + 872\*s^3 + 437\*s^2 + 89\*s + 65)

numpoly =

54 4266 77814 57888 106434

denpoly =

Columns 1 through 5

1 951 69361 66004 22828

Columns 6 through 8

8339 2340 0

pretty(G) =

3 2

(54 s + 1458) (s + 52 s + 37 s + 73)

-----------------------------------------------------

2 4 3 2

s (s + 79 s + 36) (s + 872 s + 437 s + 89 s + 65)