

Universidade Federal de Uberlândia



FEELT - Faculdade de Engenharia Elétrica

Sistemas e Controle

Roteiro 04B

Gabriel Alves Caixeta Custódio - 12011ECP008

Uberlândia 2023

Questão 03

residue: a função residue quando usada da seguinte maneira, calcula os coeficientes das frações parciais de uma função, os polos correspondentes e os resíduos: [r,p,k] = residue(b,a) Quando é utilizada na forma a seguir, ela converte a expansão em frações parciais de volta para a forma polinomial: [b,a] = residue(r,p,k)

poly: converte as raízes de um polinômio nos coeficientes correspondentes. Quando o parâmetro passado é um vetor, a função retorna os coeficientes de um polinômio cujas raízes são os elementos do vetor passado. Quando o parâmetro é uma matriz n×n, retorna os coeficientes do polinômio característico da matriz.

roots: retorna as raízes de um polinômio passado através de seus coeficientes em um vetor.

symbolic: através da função syms, pode ser usada para definir variáveis simbólicas, expressões simbólicas, entre outros.

partfrac: encontra a decomposição em frações parciais de uma expressão em relação a uma variável (se não for passada, a decomposição é feita em relação pela variável simbólica).

conv: retorna a convolução de dois vetores.

polyval: avalia um polinômio em determinados pontos desejados, sendo que polinômio é passado através de seus coeficientes.

tf: usada para se obter uma função de transferência a partir dos coeficientes do numerador e denominador passados como parâmetros.

tf2zp: a função encontra a matriz de zeros, o vetor de polos e o vetor de ganhos associados a partir dos parâmetros b e a (numerador e denominador respectivamente) de uma função de transferência.

tfdata: retorna os coeficientes do numerador e do denominador de uma função de transferência.

impulse: traça o gráfico de resposta de um sistema dinâmico à um impulso de entrada.

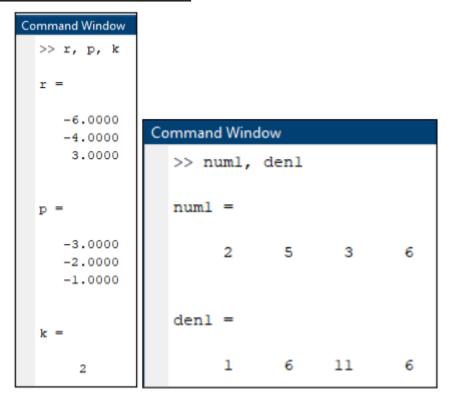
step: traça o gráfico de resposta de um sistema dinâmico à uma entrada do tipo step.

pretty: imprime o resultado de uma forma mais legível.

Questão 04

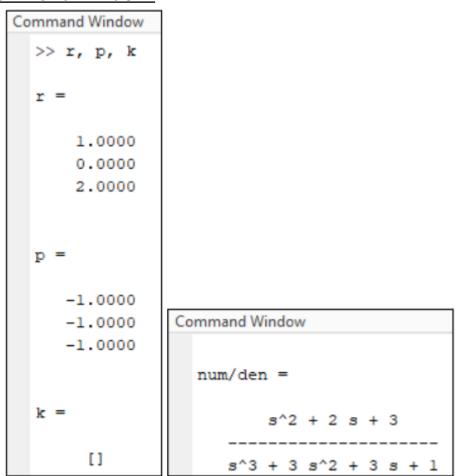
Exemplo 2-1 Código no Matlab:

```
exemplo2_1.m × +
       % Exemplo 2-1 do livro passado
2
       clc, clear, close all
3 -
       %% Definindo coeficientes da função de transferência
       num = [2 5 3 6];
       den = [1 6 11 6];
8
9
       %% Calculando resíduos
10 -
      [r, p, k] = residue(num, den);
11
12
       %% Fazendo processo inverso
13 -
      r1 = [-6 -4 3];
       p1 = [-3 -2 -1];
14 -
       k1 = 2;
15 -
16
17 -
      [numl, denl] = residue(rl, pl, kl);
```



Exemplo 2-2 Código no Matlab:

```
exemplo2_2.m × +
       % Exemplo 2-2 do livro passado
3 -
      clc, clear, close all
       %% Definindo coeficientes da função de transferência
       num = [0 1 2 3];
       den = [1 \ 3 \ 3 \ 1];
8
9
       %% Calculando resíduos
10 -
       [r, p, k] = residue(num, den);
11
12
       %% Fazendo processo inverso
13 -
      printsys(num, den, 's');
```



Exemplo 2-3 Código no Matlab:

```
exemplo2_3.m × +
       % Exemplo 2-3 do livro passado
2
3 -
       clc, clear, close all
5
      %% Definindo coeficientes da função de transferência
6 -
      num = [5 30 55 30];
7 -
       den = [1 9 33 65];
8
9
       %% Calculando zeros polos e ganho do sistema
10 -
       [z, p, K] = tf2zp(num, den);
11
      %% Fazendo processo inverso
12
13 -
      z1 = [-1; -2; -3];
       p1 = [-2 - 3*j; -2 + 3*j; -5];
14 -
15 -
       K1 = 5;
16
17 -
       [numl, denl] = zp2tf(zl, pl, Kl);
       printsys(numl, denl, 's');
```

```
z =

-3.0000
-2.0000
-1.0000

p =

-5.0000 + 0.0000i
-2.0000 + 3.0000i
-2.0000 - 3.0000i

K =
```

Exemplo 2-4 Código no Matlab:

```
exemplo2_8.m × exemplo2_5.m × exemplo2_4.m × +

1 % Exemplo 2-4 do livro passado

2 3 - clc, clear, close all

5 % Definindo coeficientes da função de transferência
6 - num = [100 1000];
7 - den = [1 10 100 600 0];

8 % Calculando os resíduos

10 - [r, p, k] = residue(num, den);
```

Saída na Command Window:

```
r =

-0.6845 + 0.2233i
-0.6845 - 0.2233i
-0.2977 + 0.0000i
1.6667 + 0.0000i

p =

-1.2898 + 8.8991i
-1.2898 - 8.8991i
-7.4204 + 0.0000i
0.0000 + 0.0000i
```

Exemplo 2-5 Código no Matlab:

```
exemplo2_8.m × exemplo2_5.m × exemplo2_4.m × +
       % Exemplo 2-5 do livro passado
       clc, clear, close all
3 -
 4
       %% Definindo coeficientes da função de transferência
       num = [1 8 23 35 28 3];
7 -
       den = [1 6 8 0];
9
       %% Calculando os resíduos
       [r, p, k] = residue(num, den);
10 -
11
12
       %% Calculando inversa de Laplace
13 -
       syms s
14 -
       F = s^2 + 2*s + 3 + 0.375/(s + 4) + 0.25/(s + 2) + 0.375/s;
15
16 -
       f = ilaplace(F);
17 -
       simplify(f);
18 -
       pretty(f);
```

```
r =

0.3750
0.2500
0.3750

p =

-4
-2
0

k =

1 2 3
```

```
Command Window

exp(-2 t) exp(-4 t) 3 3
-----+ + ------+ 3 dirac(t) + 2 dirac'(t) + dirac''(t) + -
4 8 8
```

Exemplo 2-6 Código no Matlab:

```
exemplo2_6.m × +
       % Exemplo 2-6 do livro passado
1
2
      clc, clear, close all
3 -
5
       %% Caso 1
      z1 = [];
      pl = [-1 + 2*j; -1 - 2*j];
       K1 = 10;
      [numl, den1] = zp2tf(z1, p1, K1);
10 -
11 -
      printsys(numl, denl);
12
13
       %% Caso 2
14 -
      z2 = [0];
      p2 = [-1 + 2*j; -1 - 2*j];
15 -
       K2 = 10;
16 -
17
18 -
      [num2, den2] = zp2tf(z2, p2, K2);
19 -
      printsys(num2, den2);
20
21
       %% Caso 3
22 -
      z3 = [-1];
23 -
      p3 = [-2; -4; -8];
24 -
       K3 = 12;
25
       [num3, den3] = zp2tf(z3, p3, K3);
26 -
27 -
       printsys(num3, den3);
```

Exemplo 2-7 Código no Matlab:

```
exemplo2_7.m  

the semplo2_7.m  

the semplo2_7.m
```

Exemplo 2-8 Código no Matlab:

```
exemplo2_8.m × exemplo2_5.m × exemplo2_4.m ×
       % Exemplo 2-8 do livro passado
1
2
3 -
       clc, clear, close all
       %% Definindo coeficientes da função de transferência
6 -
       num = [25.04 5.008];
       den = [1 5.03247 25.1026 5.008];
7 -
8
       %% Calculando os coeficientes da equação de espaço-estado
9
10 -
       [A, B, C, D] = tf2ss(num, den);
```

```
Command Window

A =

-5.0325 -25.1026 -5.0080
1.0000 0 0
0 1.0000 0

B =

1
0
0

C =

0 25.0400 5.0080

D =
```

Exemplo 2-9 Código no Matlab:

```
exemplo2_9.m × +
1
       % Exemplo 2-9 do livro passado
2
3 -
       clc, clear, close all
4
5
       %% Definindo coeficientes da equação de espaço-estado
       A = [0 \ 1 \ 0; \ 0 \ 0 \ 1; \ -5.008 \ -25.1026 \ -5.03247];
7 -
       B = [0; 25.04; -121.005];
       C = [1 \ 0 \ 0];
9 -
       D = [0];
10
       %% Convertendo em função transferência
11
      [num, den] = ss2tf(A, B, C, D);
```

```
    Command Window

    num =
    0
    0
    25.0400
    5.0080

    den =
    1.0000
    5.0325
    25.1026
    5.0080
```

Exemplo 2-10 Código no Matlab:

```
exemplo2_10.m × +
1
        % Exemplo 2-10 do livro passado
 2
3 -
      clc, clear, close all
 4
5
       %% Definindo coeficientes da equação de espaço-estado
       A = [0 \ 1; -2 \ -3];
7 -
       B = [1 \ 0; \ 0 \ 1];
       C = [1 \ 0];
       D = [0 \ 0];
9 -
10
11
       %% Convertendo em função transferência
       [numl, denl] = ss2tf(A, B, C, D, 1);
13 -
        [num2, den2] = ss2tf(A, B, C, D, 2);
```

```
    Command Window

    num1 =

    0 1 3

    den1 =

    1 3 2

    num2 =

    0 0 1

    den2 =

    1 3 2
```

Exemplo 2-11 Código no Matlab:

```
exemplo2_11.m × +
1
       % Exemplo 2-11 do livro passado
2
3 -
      clc, clear, close all
       %% Definindo coeficientes da equação de espaço-estado
       A = [0 1; -25 -4];
       B = [1 1; 0 1];
       C = [1 \ 0; \ 0 \ 1];
9 -
       D = [0 \ 0; \ 0 \ 0];
10
      %% Convertendo em função transferência
11
12 -
      [NUM1, den1] = ss2tf(A, B, C, D, 1);
      [NUM2, den2] = ss2tf(A, B, C, D, 2);
13 -
```

```
NUM1 =

0 1 4
0 0 -25

den1 =

1.0000 4.0000 25.0000

NUM2 =

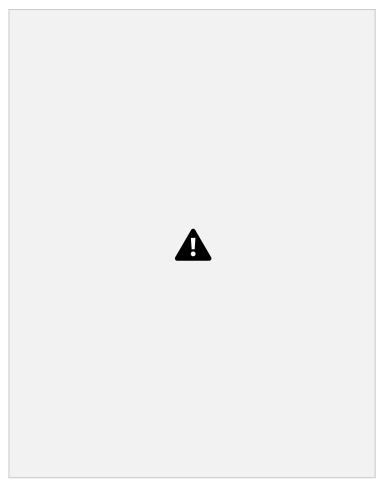
0 1.0000 5.0000
0 1.0000 -25.0000

den2 =

1.0000 4.0000 25.0000
```

Exemplo 2-12 Código no Matlab:

```
exemplo2_12.m × +
1
       % Exemplo 2-12 do livro passado
 2
 3 -
       clc, clear, close all
 5
       %% Definindo coeficientes da função transferência
       numl = [10];
7 -
       den1 = [1 2 10];
9 -
      num2 = [5];
10 -
       den2 = [1 5];
11
12
       %% Convertendo em função transferência
13 -
       [num, den] = series(numl, den1, num2, den2);
14 -
       printsys(num, den);
15 -
       [num, den] = parallel(numl, denl, num2, den2);
16 -
      printsys(num, den);
17 -
       [num, den] = feedback(numl, den1, num2, den2);
18 -
       printsys(num, den);
```



Exemplo 2-13 Código no Matlab:

```
exemplo2_12.m × exemplo2_13.m × +
       % Exemplo 2-13 do livro passado
3 -
      clc, clear, close all
4
5 -
      sysgl = [4];
6 -
      numg2 = [2]; deng2 = [1 9 8]; sysg2 = tf(numg2, deng2);
7 -
      numg3 = [1]; deng3 = [1 0]; sysg3 = tf(numg3, deng3);
      sysh = [0.2];
      sys1 = feedback(sysg2, sysh);
9 -
10 -
      sys2 = series(sys1, sysg3);
11 -
      sys3 = series(sysgl, sys2);
12 -
      sys = feedback(sys3, [1]);
```

Exemplo 2-14 Código no Matlab:

```
exemplo2_14.m* × +
      % Exemplo 2-14 do livro passado
2
    clc, clear, close all
3 -
5 -
     numgl = [2]; dengl = [1 0]; sysgl = tf(numgl, dengl);
6 -
     numg2 = [1]; deng2 = [1 0]; sysg2 = tf(numg2, deng2);
7 -
      sysg3 = [1];
8 -
     sys1 = parallel(sysg1, sysg3);
     sys2 = series(sys1, sysg2);
9 -
     sys = feedback(sys2, [1])
10 -
    sys ss = ss(sys)
11 -
```

```
Command Window
 sys =
     s + 2
   s^2 + s + 2
 Continuous-time transfer function.
 sys_ss =
  A =
      x1 x2
   x1 -1 -2
   x2 1 0
   B =
    ul
   x1 2
   x2 0
   C =
       x1 x2
   yl 0.5 1
   D =
     ul
    y1 0
 Continuous-time state-space model.
```

Exemplo 2-15 Código no Matlab:

```
exemplo2_15.m × +
1
       % Exemplo 2-15 do livro passado
3 -
      clc, clear, close all
4
5 -
      numl = [1 2 1]; denl = [1 0 0]; sysl = tf(numl, denl);
6 -
      num2 = [2]; den2 = [1 5]; sys2 = tf(num2, den2);
7 -
      sys ssl = ss(sysl);
      sys ss2 = ss(sys2);
9 -
      sys ss3 = series(sys ss1, sys ss2);
10 -
      sys_ss = feedback(sys_ss3, [1])
11
     sys tf = tf(sys ss)
```

```
Command Window
  sys_ss =
       x1 x2 x3
    x1 -7 2 1
    x2 -2 0 0
x3 0 1 0
   B =
       ul
    x2 2
    x3 0
   C =
       x1 x2 x3
    yl 1 0 0
   D =
       ul
    vl 0
  Continuous-time state-space model.
 sys tf =
     2 s^2 + 4 s + 2
   s^3 + 7 s^2 + 4 s + 2
  Continuous-time transfer function.
```