

Universidade Federal de Minas Gerais - Departamento de Engenharia Eletrônica

**ELT062 - OFICINA DE SIMULAÇÃO ANALÓGICA E DIGITAL  
EM CONTROLE**

**TUTORIAL 6**

Instrutores: Alexandre R. Mesquita e Bruno O. S. Teixeira

PARTE 1

Na aula anterior aprendemos como podemos aproximar o comportamento de um sistema em torno de um ponto de operação pelo comportamento de um sistema linear. Uma das vantagens de se trabalhar com sistemas lineares é que conhecemos soluções analíticas para EDOs lineares independentemente de sua dimensão. Outra vantagem é que podemos usar métodos de transformadas para estudá-los.

Na aula anterior consideramos o sistema que consiste num pêndulo rígido com ângulo  $\theta(t)$  sujeito a um torque  $\tau(t)$  em sua base:

$$ml^2\ddot{\theta} + b\dot{\theta} + gl \sin \theta = \tau(t) .$$

A linearização desse sistema em torno do ponto de operação ( $\bar{\theta}, \bar{\tau} = gl \sin \bar{\theta}$ ) resultou na seguinte função de transferência

$$G(s) = \frac{\Theta(s)}{T(s)} = \frac{1}{ml^2s^2 + bs + gl \cos \bar{\theta}} .$$

Assumimos que o torque é aplicado ao sistema através de um motor DC com função de transferência entre a tensão de campo  $V(s)$  e o torque  $T(s)$  dada por:

$$F(s) = \frac{T(s)}{V(s)} = \frac{L}{s + L} ,$$

onde  $L > 0$  é uma constante. Aqui podemos observar uma das utilidades do conceito de função de transferência. Se quisermos avaliar o efeito da tensão  $V$  sobre a variação do ângulo  $\Theta$ , basta multiplicarmos as duas funções de transferência:

$$H(s) = \frac{\Theta(s)}{V(s)} = G(s)F(s) = \frac{L}{(s + L)(ml^2s^2 + bs + gl \cos \bar{\theta})} .$$

Queremos simular o sistema acima, obtendo a resposta  $\theta(t)$  em função da entrada de tensão  $v(t)$ . Crie um script com nome `tut6a.m`. Iniciamos definindo os parâmetros do sistema:

```
clear
close all
clc
```

```
m=0.1;
l=0.5;
g=9.8;
b=0.01;
thetabar=45/180*pi;
```

Em seguida, usamos o comando `tf` para criar objetos do tipo função de transferência para  $F$  e  $G$ . Damos uma pausa após a execução do comando para visualizarmos seu resultado.

```
G=tf([1],[m*1^2 b g*1*cos(thetabar)])
F=tf([100],[1 100])
pause
```

Note como uma função de transferência racional é definida a partir dos coeficientes de seu numerador e de seu denominador.

Para gerarmos  $H$ , usamos o comando `conv`. Se `pol1` e `pol2` são vetores com os coeficientes de dois polinômios, então o produto desses polinômios terá coeficientes dados por `conv(pol1,pol2)`.

```
numH=conv(G.num{1},F.num{1});
denH=conv(G.den{1},F.den{1});
H=tf(numH,denH)
pause
```

Note que `G.num{1}` denota o numerador do primeiro elemento de  $G$  (neste caso  $G$  possui apenas um elemento).

Para visualizar os pólos e zeros da função de transferência  $H$ , usamos o comando `zpk`.

```
zpk(H)
pause
```

Usamos o comando `residue`

```
[Residuos,Polos,GanhoDireto]=residue(H.num{1},H.den{1})
```

para decompor  $H$  em frações parciais de forma que

$$H(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \frac{r_2^*}{s - p_2^*}$$

Essa decomposição nos permite obter a transformada inversa de forma explícita:

$$h(t) = [r_1 e^{p_1 t} + e^{\operatorname{Re}[p_2]t} (|r_2| \cos(\operatorname{Im}[p_2]t + \angle r_2))] \operatorname{degrau}(t)$$

Podemos com isso escrever uma expressão simbólica no Matlab para representar  $h(t)$ :

```
syms t
h=Residuos(1)*exp(Polos(1)*t)*heaviside(t)+...
    2*(real(Residuos(2))*cos(imag(Polos(2))*t)...
    -imag(Residuos(2))*sin(imag(Polos(2))*t))...
*exp(real(Polos(2))*t)*heaviside(t);
pretty(h)
pause
```

Usamos o comando `ezplot` para plotar essa solução. Em seguida, comparamos o resultado com o daquele dado pelo comando `impulse`, que calcula e plota diretamente a resposta a impulso do sistema dado.

```
ezplot(h)
hold on
impulse(H,'r')
xlabel('tempo')
```

```
ylabel('\theta(t)')
title('Resposta a impulso')
pause
```

Em seguida, usamos o comando `step` para gerar e plotar a resposta a entrada degrau do sistema  $H$ .

```
figure
step(H)
xlabel('tempo')
ylabel('\theta(t)')
title('Resposta a degrau')
pause
```

Por fim, podemos usar o comando `lsim` para gerar a resposta de  $H$  para uma entrada arbitrária.

```
tempo=0:0.2:30;
u=0.1*cos(tempo);
```

```
figure
lsim(H,u,tempo)
legend('\theta')
```

Note que também poderíamos escrever `y=step(H)` para salvar a resposta a impulso obtida no vetor  $y$ . O mesmo vale para `impulse` e `lsim`. Quando uma variável de saída não é determinada, essas funções plotam a resposta obtida.

Execute o arquivo e observe os resultados obtidos em cada pausa.

## PARTE 2

Em seguida, obteremos o mesmo tipo de resposta acima para um sistema a tempo discreto. Nosso sistema é descrito pelas equações:

$$x[k+1] = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x[k] + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1[k] \\ u_2[k] \end{bmatrix}$$

Esse sistema pode ser usado para representar o movimento de uma aeronave em uma dimensão, estando ela sujeita a aceleração  $u_2[k]$  e a ventos de velocidade  $u_1[k]$ . Aqui  $x_1$  e  $x_2$  representam a posição e a aceleração da aeronave respectivamente.

Crie um script `tut6b.m` para simular este sistema. Iniciamos definido o sistema usando o comando `ss`.

```
clear
close all
clc
```

```
A=[1 1; 0 1];
B=[1 0; 0 1];
C=[1 0; 0 1];
D=0;
```

```
H=ss(A,B,C,D,1)
pause
```

Lembre que `ss` pode ser usado para definir sistemas com estado  $x$ , entrada  $u$  e saída  $y$  dados na forma:

$$\begin{aligned}x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k] .\end{aligned}$$

Como estamos definindo um sistema a tempo discreto, o último argumento de `ss` é o período de amostragem (aqui igual a 1).

Podemos converter o sistema da forma acima para a forma de função de transferência usando o comando `tf`. Como o sistema possui duas saídas e duas entradas, obteremos uma matriz 2x2 que associa uma função de transferência a cada par de entrada e saída.

```
H=tf(H)
pause
```

Queremos focar nossa atenção apenas na influência da aceleração  $u_2$  sobre a posição  $x_1$  da aeronave. Para isso, isolaremos o elemento da matriz que corresponde ao mencionado par de entrada e saída:

```
H=H(1,2)
pause
```

Conforme feito anteriormente, podemos usar o comando `zpk` para visualizar os pólos e zeros dessa função de transferência:

```
zpk(H)
pause
```

Usando `residue`, obtemos a decomposição em frações parciais e a resposta a impulso de forma explícita:

```
[Residuos,Polos,GanhoDireto]=residue(H.num{1},H.den{1})
syms k
h=Residuos(1)*Polos(1)^(k-1)*heaviside(k-1)+...
    Residuos(2)*Polos(2)^(k-1)*(k-1)*heaviside(k-1);
pretty(h)
pause
```

Note que dessa vez obtivemos pólos com multiplicidade 2, o que resulta numa decomposição na forma:

$$H(z) = \frac{r_1}{z - p_1} + \frac{r_2}{(z - p_1)^2}$$

Em seguida plotamos as respostas do sistema para entradas do tipo impulso, degrau e senóide.

```
ezplot(h,[0 20])
hold on
impulse(H,'r',20)
xlabel('tempo')
ylabel('\theta(t)')
title('Resposta a impulso')
```

```
figure
step(H)
```

```
xlabel('tempo')  
ylabel('\theta(t)')  
title('Resposta a degrau')
```

```
tempo=0:1:30;  
u=0.1*cos(tempo);  
figure  
lsim(H,u,tempo)  
legend('\theta')
```

Execute o arquivo e observe os resultados obtidos.