Polos, zeros e funções de transferência

Vilma A. Oliveira

USP São Carlos

Aula 2a

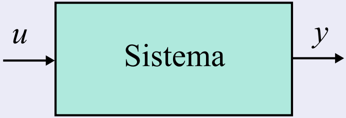
Colaboradores

Elmer Alexis Gamboa Peñaloza
Rodolpho Vilela Alves Neves
Rafael Fernando Quirino Magossi
Rafael Mariano

Introdução

Esta aula é para ser ministrada de forma interativa utilizando um microcomputador por aluno ou por um grupo de 2 alunos com o aplicativo Matlab instalado. Tem por objetivo apresentar aos alunos os conceitos e funções Matlab básicos nos tópicos sobre representação entrada-saída tratado no Capítulo 2. No final da apresentação será proposta uma tarefa extraclasse a ser iniciada na sala de aula. Pode ser atribuída nota à tarefa realizada desde que a sua solução seja encaminhada e a aluna ou aluno tenha participado da aula. A solução deve ser enviada via área do aluno, pode ser via a plataforma Moodle de disciplinas, por exemplo

Representação entrada-saída



Se a relação entre u e y for descrita por uma equação algébrica o sistema é dito ser estático, se por equação diferencial o sistema é dito ser dinâmico. Se u e y são relacionados por uma equação diferencial não linear tem-se um sistema dinâmico não linear. Se a equação diferencial for linear e tiver coeficientes constantes tem-se um sistema linear e invariante no tempo. Equação diferencial linear de ordem n

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + a_2 \frac{d^{n-2} y(t)}{dt^{n-2}} + \cdots + a_{n-1} \frac{dy(t)}{dt} + a_n y(t) = b_m \frac{d^m u(t)}{dt^m} + \cdots + b_1 \frac{du(t)}{dt} + b_0 u(t)$$

$y(t)$ saída, $u(t)$ entrada, $a_i, i = 1, \cdots, n$, $b_i, i = 0, \cdots, m$ constantes com $m \leq n$.

Função de transferência

A função de transferência é definida para sistemas lineares invariantes no tempo e pode ser interpretada como a relação entrada-saída:

$$G(s) = \frac{y(s)}{u(s)}.$$

A transformada de Laplace da derivada é dada por:

$$\mathcal{L}[\frac{dy}{dt}] = sy(s) - y(0^-)$$

e a transformada da derivada segunda pode ser obtida usando o resultado anterior:

$$\begin{aligned} \mathcal{L}[\frac{d^2y}{dt^2}] &= s\mathcal{L}[\frac{dy}{dt}] - \dot{y}(0^-) \\ &= s^2y(s) - sy(0^-) - \dot{y}(0^-). \end{aligned}$$

Usando a propriedade de diferenciação no tempo da transformada de Laplace n vezes e fazendo as condições iniciais iguais a zero:

$$\mathcal{L}[\frac{d^n y}{dt^n}] = s^n y(s) - s^{n-1}y(0^-) - \cdots - sy^{n-2}(0^-) - y^{n-1}(0^-).$$

A solução é dada por:

$$y(s) = \frac{N(s)}{D(s)}u(s)$$

em que

$$\begin{aligned} D(s) &= s^n + a_1 s^{n-1} + \cdots + a_n \\ N(s) &= b_m s^m + b_{m-1} s^{m-1} + \cdots + b_0 \end{aligned}$$

Algumas funções do Matlab comumente usadas em controle

»$D = 4s^2 + 10s - 8$; N=s+2
roots([4 10 -8]) % raízes de polinômios
r=[6.5 4 2.3 -1.2 0.5];
» p=poly(r)
» pzmap(N,D)% plota os polos e zeros no plano-s
Gerando um sistema G1:
G1=tf([1 2],[4 10 -8]);
G1=zpk(G1) % sistema com numerador e denominador fatorado
$[z, p, k]$=zpkdata(G1,'v') % extrai polos, zeros e ganho na forma de vetor
G1.z; G1.p % extrai os zeros e polos de G1
G1=ss(G1) % sistema na forma espaço de estado
$[A, B, C, D]$=ssdata(G1) % extrai matrizes do sistema
G1.a, G1.b, G1.c, G1.d % extrai as matrizes do sistema

Exemplos

Os comandos tf(num,den) and tf('s') gera um objeto TF que pode ser usado nas rotinas Matlab:

» G1 = tf([1 1],[1 2 1]);

» s = tf('s') % define s como a variável de Laplace

» G1 = (s+1)/(s^ 2+2*s+1);

$$G_1 = \frac{(s+1)}{(s^2 + 2 * s + 1)}$$

»G2 = tf([-10 20 0],[1 7 20 28 19 5])

»zpk(G2)

Zero/pole/gain:

$$\frac{-10s(s-2)}{(s+1)^3(s^2+4s+5)}$$

A solução via transformada de Laplace pode ser obtida via a expansão em frações parciais o que é sempre trabalhoso. No Matlab esse trabalho é simplificado com o uso do comando *residue* ou *ilaplace*. Seja, por exemplo, uma função de transferência definida pelos coeficientes de N(s) e D(s):

$$G(s) = \frac{N(s)}{D(s)} = \frac{5s^3 + 3s^2 - 2s + 7}{-4s^3 + 8s^2 + 3}$$

N = [5 3 -2 7]; D = [-4 0 8 3]. Usando o comando *residue*:

$$[r, p, k] = residue(N, D)$$

obtém-se os coeficientes da expansão:
r = -1.4167 -0.6653 1.3320; p = 1.5737 -1.1644 -0.4093; k = -1.2500
e a decomposição pode ser obtida:

$$G(s) = \frac{-1.4167}{s - 1.5737} + \frac{-0.6653}{s + 1.1644} + \frac{1.3320}{s + 0.4093} - 1.2500.$$

Os modos do sistema são então:

$$r(1)e^{p(1)}, r(2)e^{p(2)}, r(3)e^{p(3)}$$

Modos com polos complexos

Para G1=(s+2)/(s^ 2+2*s+5)
verificar que os modos são do tipo:
$2\,\mathrm{Re}(r)(cos(Im(p)*t), 2\,\mathrm{Im}(r)(sen(Im(p)*t)$
com r(1)= 0.5000 + 0.2500i, r(2)= 0.5000 - 0.2500i
Usar o comando 'ilaplace' para conferir.

Resposta x localização dos polos no plano-s

```
clear all; close all; clc;
s = tf('s');
```

Caso 1: Polos simples	Case 2: Polos reais positivos
p1 = 1;	p5 = 5;
G1 = 1/(s+p1);	G2 = 1/(s-p5);
figure(1)	figure(2)
impulse(G1)	impulse(G2)

Resposta x localização dos polos no plano-s

Caso 3: Polos complexos

```
s = tf('s')
wn = 100; % Natural frequency
zeta1 = [0 0.5 1 1.5]; % Damping values
% Obter funções de transferência
for n = 1:4
zeta = zeta1(n);
G2n=wn^2 /(s^2+ 2*zeta*wn*s+wn^2);
end
```

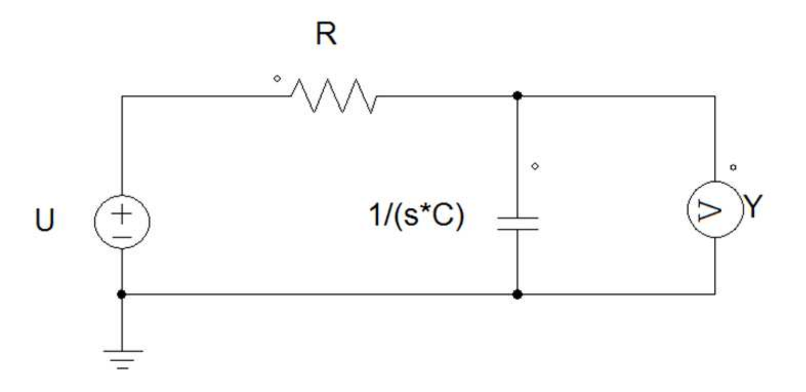
```
%Plotando respostas típicas
for k = 1:length(G2)
figure(2)

hold on
step(G2{k},0:.0001:.2)
hold off
end
```

Tarefa!

Case 4: Selecionar polos com parte real nula e obter a função de transferência e resposta impulsional.
Case 5: Repetir o item anterior para polos complexos com parte real positiva.

Tarefa!

Obter a função de transferência do circuito RC e plotar a curva de carga e descarga do capacitor com $R = 1k\ \Omega$ e $C = 1000uF$. Verificar que a curva de carga pode ser obtida com o comando ‘step’ e a de descarga com o comando ‘impulse’.



Tarefa para nota

A solução da tarefa deve conter o que foi estudado, o que foi feito e analisado.

Tarefa

Considerar os casos estudados de sistemas de segunda ordem + um polo real e do sistema de segunda ordem + 1 zero. Normalizar (ver anexo).

1. Analisar a dominância do polo complexo ou do polo real (plotar apenas esses casos.
2. Analisar o efeito da posição do zero à esquerda no sobressinal da resposta ao degrau (qual a relação entre a localização do zero e a localização dos polos complexos para sobressinal diminuir?
3. Analisar o efeito na resposta ao degrau do zero adicionado à direita (verificar o efeito de *undershooting*).

Anexo: Normalizando o sistema de segunda ordem

O objetivo da normalização é que, ao inserir os polos, o valor final seja igual a 1 para entrada degrau unitária. Para normalizar, divida a frequência natural ao quadrado no denominador e no numerador:

w_n frequência natural

ξ amortecimento

$T_n(s)$ FT normalizada

$$T(s) = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2} \Rightarrow = \frac{\frac{w_n^2}{w_n^2}}{\frac{(s^2+2\xi w_n s + w_n^2)}{w_n^2}} = \frac{1}{\frac{s^2}{w_n^2} + \frac{2\xi}{w_n} s + 1} = T_n(s)$$

$$\Rightarrow \quad T_n(s) = \frac{1}{\frac{s^2}{w_n^2} + \frac{2\xi}{w_n} s + 1}$$

Anexo: Normalizando o sistema de segunda ordem

Para situações em que o numerador é diferente de w_n^2, tem-se:

$$T_k(s) = \frac{K}{s^2 + 2\xi w_n s + w_n^2}$$

Fazendo $K = w_{n^2} K_n \rightarrow K_n = \frac{K}{w_n^2}$:

$$T_k(s) = K_n \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2} = K_n T(s) = K_n T_n(s) = \frac{K}{w_n^2} \frac{1}{\frac{s^2}{w_n^2} + \frac{2\xi}{w_n} s + 1}$$

Caso queira manter o valor final em um para uma entrada degrau:

$$T_{kn}(s) = \frac{w_n^2}{K} T_k(s)$$

w_n frequência natural

ξ amortecimento

$T_n(s)$ FT normalizada

Adicionando um polo real a T(s) na forma normalizada

w_n freqüência natural

ξ amortecimento

α parâmetro

Para polos, a forma normalizada, o polo adicionado deve ter o formato:

$polo = -\alpha\xi w_n$, e $\alpha = -\frac{polo}{\xi w_n}, \alpha > 0$

$$T_p(s) = T_n(s) * \frac{1}{(s - polo)} = T_n(s)\frac{1}{-polo * \left(-\frac{s}{polo} + 1\right)} = \frac{T_n(s)}{\alpha\xi w_n\left(\frac{s}{\alpha\xi w_n} + 1\right)} = \frac{1}{\alpha\xi w_n} * \frac{T_n(s)}{\left(\frac{s}{\alpha\xi w_n} + 1\right)}$$

Como o objetivo é normalizar, ou seja, não alterar a resposta em regime permanente, então, $T_{pn}(s) = \alpha\xi w_n T_p(s)$. Portanto:

$$T_{pn}(s) = \frac{1}{\left(\frac{s^2}{w_n^2} + \frac{2\xi}{w_n}s + 1\right)\left(\frac{s}{\alpha\xi w_n} + 1\right)}$$

Adicionando um pólo real a T(s) na forma normalizada

$$T_{pn}(s) = \frac{1}{\left(\frac{s^2}{w_n^2} + \frac{2\xi}{w_n}s + 1\right)\left(\frac{s}{\alpha\xi w_n} + 1\right)}$$

w_n freqüência natural

ξ amortecimento

α parâmetro

MATLAB

```
%com polos complexos
s=tf('s')
wn=100;
zeta=0.6;
T2 = wn^2/(s^2+2*zeta*wn*s+wn^2);
figure(1)
 step(T2,'g--');
  hold on
```

```
for i=1:2:40
alfa=i*0.2;
Tpn =
1/((s^2/(wn^2)+2*zeta/wn*s+1)*(s/(alfa
*zeta*wn)+1));
step(Tpn);
hold on
end
lineobj = findobj('type', 'line');
set(lineobj, 'linewidth', 1.5);
```

Adicionando um zero real negativo a T(s) na forma normalizada

Para zeros na forma normalizada, o zero adicionado deve ter o formato:

w_n freqüência natural

ξ amortecimento

α parâmetro

$zero = -\alpha\xi w_n$, e $\alpha = -\frac{zero}{\xi w_n}, \alpha > 0$

$$T_{z^-}(s) = T_n(s)(s - zero) = T_n(s)(-zero)\left(-\frac{s}{zero} + 1\right) = T_n(s)(\alpha\xi w_n)\left(\frac{s}{\alpha\xi w_n} + 1\right)$$

Como o objetivo é normalizar, ou seja, não alterar a resposta em regime permanente, então, $T_{zn^-}(s) = \frac{1}{\alpha\xi w_n} T_{z^-}(s)$. Portanto:

$$T_{zn^-}(s) = \frac{\left(\frac{s}{\alpha\xi w_n} + 1\right)}{\left(\frac{s^2}{w_n^2} + \frac{2\xi}{w_n}s + 1\right)}$$

Adição de um Zero

$$T_{zn^-}(s) = \frac{\left(\dfrac{s}{\alpha\xi w_n} + 1\right)}{\left(\dfrac{s^2}{w_n^2} + \dfrac{2\xi}{w_n}s + 1\right)}$$

w_n freqüência natural

ξ amortecimento

α parâmetro

MATLAB

```
%Efeito de um zero localizado a
esquerda na resposta típica de um
sistema de 2a. ordem
%com polos complexos
figure(2)
step(T1,'g--')
hold on
```

```
for i=1:10
alfa=0.5*i;
TznNeg=(1+s/(alfa*zeta*wn))/(s^2/(wn^
2)+2*zeta/wn*s+1);
step(TznNeg);
end
lineobj = findobj('type', 'line');
set(lineobj, 'linewidth', 2);
hold off
```

Adicionando um zero real positivo a T(s) na forma normalizada

Para zeros positivos na forma normalizada, o zero adicionado deve ter o formato:

$zero = \alpha\xi w_n$, e $\alpha = \frac{zero}{\xi w_n}, \alpha > 0$

w_n freqüência natural

ξ amortecimento

α parâmetro

$$T_{z^+}(s) = T_n(s)(s - zero) = T_n(s)(-zero)\left(-\frac{s}{zero} + 1\right) = T_n(s)(-\alpha\xi w_n)\left(-\frac{s}{\alpha\xi w_n} + 1\right) =$$

Como o objetivo é normalizar, ou seja, não alterar a resposta em regime permanente, então, $T_{zn^+}(s) = -\frac{1}{\alpha\xi w_n} T_{z^+}(s)$. Portanto:

$$T_{zn^+}(s) = \frac{-\left(\frac{s}{\alpha\xi w_n} - 1\right)}{\left(\frac{s^2}{w_n^2} + \frac{2\xi}{w_n}s + 1\right)}$$

Adicionando um zero real positivo a T(s) na forma normalizada

$$T_{zn^+}(s) = \frac{-\left(\frac{s}{\alpha \xi w_n} - 1\right)}{\left(\frac{s^2}{w_n^2} + \frac{2\xi}{w_n}s + 1\right)}$$

w_n freqüência natural

ξ amortecimento

α parâmetro

MATLAB

```
% %Efeito de um zero localizado a direita na resposta típica de um sistema de 2a. ordem
% %com polos complexos
T4=30/(s^2+4*s+13);
% step(T4);
% forma normalizada
hold on
for i=2:10
TznPos=-30/13*(s/i-1)/(s^2/13+4*s/13+1);
 step(TznPos)
 end
T4n=30/13/(s^2/13+4*s/13+1);
impulse(T4n,'g--')
grid
 lineobj = findobj('type', 'line');
 set(lineobj, 'linewidth', 1.5);
%
hold off
```

References

[1] Matlab Product Help.

[2]Matlab Demystified. A Self-Teaching Guide, David McMahon, McGraw Hill.

[3] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, Feedback Control of Dynamic Systems, Prentice Hall