Rogério Yukio Tamaoki Rodriguez - 10772709

PME3380 - Modelagem de Sistemas Dinâmicos
Lista 5

Brasil

2020

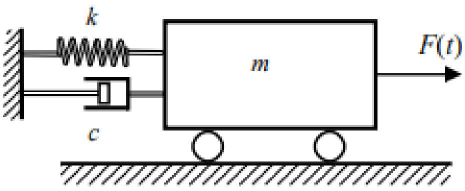
Lista de ilustrações

Sumário

1 Sistema massa mola amortecedor

Neste relatório será feita uma breve análise do sistema massa mola amortecedor exposto na figura 1.

Figura 1 – Sistema massa mola amortecedor



A equação diferencial do movimento do sistema pode ser dada por ??.

$$m\ddot{x} + c\dot{x} + kx = F(t) = F \tag{1.1}$$

Considerando que $[x, \dot{x}]^T = [x_1, x_2]^T$ o espaço de estados do sistema fica dado por 1.2.

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F \tag{1.2}$$

Aplicando a transormada de Laplace, são obtidas as equações 1.3 e 1.4.

$$\mathcal{L}_1 : sX_1 = X_2 \tag{1.3}$$

$$\mathcal{L}_2 : sX_2 = -\frac{k}{m}X_1 - \frac{c}{m}X_2 + \frac{F}{m} \tag{1.4}$$

Como deseja-se obter a saída $Y = X_1$, pode ser empregada a função de transferência 1.5.

$$G(s) = \frac{Y(s)}{F(s)} = \frac{X_1}{msX_2 + kX_1 + cX_2} = \frac{X_1}{ms^2X_1 + kX_1 + csX_1} = \frac{1}{ms^2 + kX_1 + cs} \tag{1.5}$$

1.1 Exercício

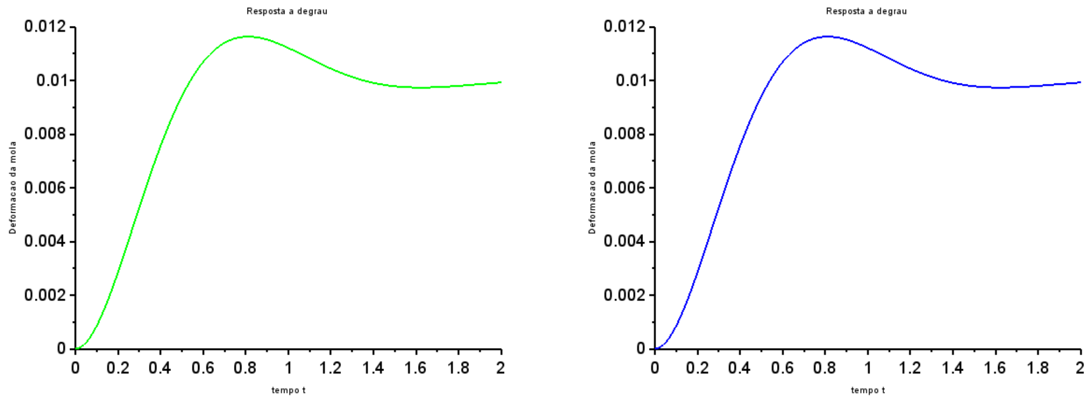
Considerando uma mola com constante elástica k=100 N/m,e uma massa m=5 kg, foram feitas simulações para $\zeta = [0,5; 1,0; 2,0]$, sendo $\zeta = c/2\sqrt{km}$. O código empregado está disposto a seguir.

```

clc()
clear()

// Definindo os parametros do sistema:
m=5
k=100;
zeta=0.5;
c=2*zeta*sqrt(k*m);
// Definindo os polinomios da funcao de transferencia:
// Numerador:
n=1;
// Denominador
d=poly([k c m],'s','coeff'); //observe a ordem contraria dos coeficientes
// Montando a funcao de transferencia, onde o parametro 'c' indica sistema de
// tempo continuo. Se for um sistema de tempo discreto, use o parametro 'd'.
G=syslin('c',n/d)
// Simulando o sistema para uma entrada degrau (u=0 para t<0 e u=1 para t>0):
// Definindo o vetor tempo:
t=0:0.01:2;
// Definindo a entrada:
u=ones(t);
// Definindo o vetor de condicoes iniciais:
// O sistema é de segunda ordem, logo sao duas condicoes iniciais.
// Não definindo as condicoes iniciais o programa assume como sendo nulas.
x0=[0;0]; // x(0) = 0 e a derivada de x(t) no instante inicial tambem eh nula.
// Realizando a simulacao com o comando csim:
[y]=csim(u,t,G,x0);
// Abrindo uma nova janela de graficos:

scf(1)
xset('thickness',2)
xset('font size',4)
plot2d(t,y,2)
xtitle('Resposta a degrau','tempo t','Deformacao da mola')

A=[0 1; -k/m -c/m];
B=[0;1/m];
```

```
C=[1 0];
D=[0];
// Montando o sistema:
suspensao=syslin('c',A,B,C,D);
[y,x]=csim(u,t,suspensao,x0);
// Abrindo uma nova janela de graficos:
scf(2)
// Mostrando o resultado da simulacao:
xset('thickness',2)
xset('font size',4)
plot2d(t,y,3)
xtitle('Resposta a degrau','tempo t','Deformacao da mola')
```

Perceba que os resultados foram obtidos através do espaço de estados, e através da função de transferência, podendo ser conferidos nas figuras 2, 3, e 4. Neles estão representados a deformação da mola obtida através de uma excitação degrau, note que os resultados correpondem.

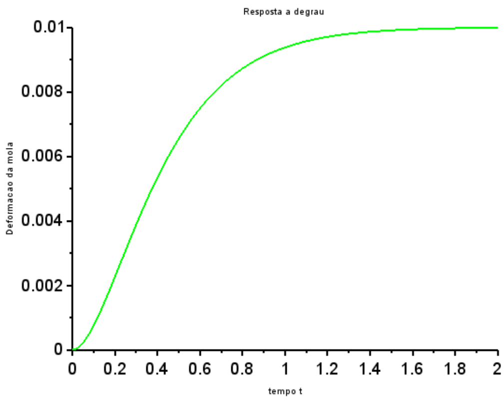
Figura 2 – $\zeta = 0,5$



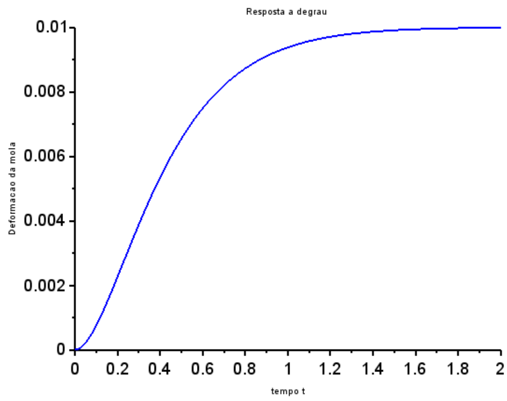
(a) Espaço de estados

(b) Função de transferência

Figura 3 – $\zeta = 1,0$

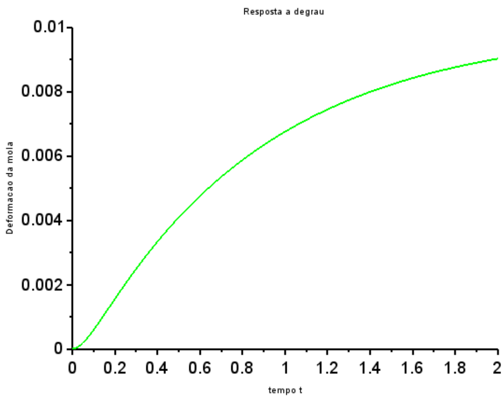


(a) Espaço de estados

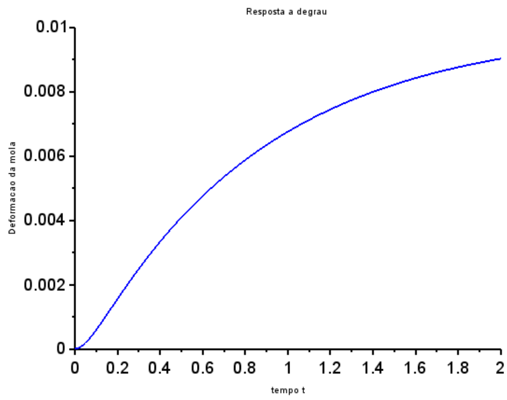


(b) Função de transferência

Figura 4 – $\zeta = 2,0$



(a) Espaço de estados



(b) Função de transferência

1.2 Análise dos pólos

Os pólos do sistema podem ser obtidos de duas formas distintas, através dos autovalores da matriz A ou das raízes do polinômio de entradas.

$$det \begin{bmatrix} -\lambda & 1 \\ -\frac{k}{m} & -\frac{c}{m} - \lambda \end{bmatrix} = 0 \tag{1.6}$$

Desenvolvendo a expressão 1.6, confirma-se que o resultado é igual ao denominador empregado na função de tranferência 1.5. Com o auxílio do código descrito a seguir, foram obtidos os valores da tabela 1.

```
clc()
clear()
```

```

// Definindo os parametros do sistema:
m=5
k=100;
zeta=0.5;
c=2*zeta*sqrt(k*m);

d=poly([k c m],'s','coeff');

p=roots(d)

w=sqrt(k/m)

mod=abs(p)

Z = abs(real(p))/mod

disp(tlist('pólos:',p(1),p(2)))
disp(tlist('frequência natural:',w))
disp('módulo dos pólos',mod)
disp('Re(p)/mod(p)',Z)
```

Tabela 1 – Análise dos pólos

ζ	Pólos (P)	ω_n	|P|	$\|Re(P)\|/\|P\|$
0,5	-2.236 + 3.873i	4.472	4.472	0.500
	-2.236 - 3.873i			

Note que para $\zeta = 0,5$, os valores dos pólos são complexos, sendo o módulo deles igual a frequência natural de oscilação do sistema. Além disso, a razão entre o módulo da parte real dos pólos e o valor absoluto deles corresponde ao fator de amortecimento.

1.3 Condições iniciais variadas

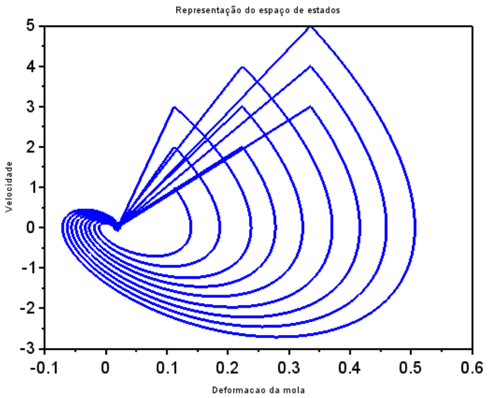
Com o intuito de observar o comportamento do sistema para diferentes condições inicias, foram realizadas simulações para $x(0)$ entre 1 e 3, e para $\dot{x}(0)$, variaram entre 1 e 5. Dessa forma, a partir do código a seguir, foram obtidas as figuras 5, 6, ??2.

```
clc()
clear()

// Definindo os parametros do sistema:
m=5
k=100;
zeta=2;
```

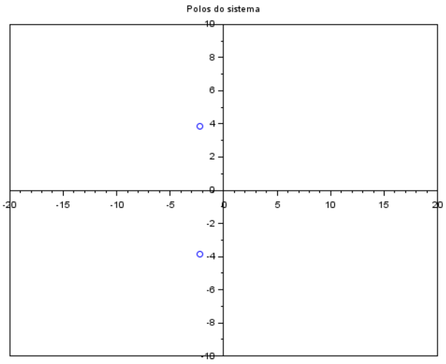
```
c=2*zeta*sqrt(k*m);
// Definindo os polinomios da funcao de transferencia:
// Numerador:
n=1;
// Denominador
d=poly([k c m],'s','coeff'); //observe a ordem contraria dos coeficientes
// Montando a funcao de transferencia, onde o parametro 'c' indica sistema
    de
// tempo continuo. Se for um sistema de tempo discreto, use o parametro 'd
    '.
G=syslin('c',n/d)
// Simulando o sistema para uma entrada degrau (u=0 para t<0 e u=1 para t
    >0):
// Definindo o vetor tempo:
t=0:0.01:2;
// Definindo a entrada:
u=ones(t);

x0=zeros(2,9); // x(0) = 0 e a derivada de x(t) no instante inicial tambem
    eh nula.

for i=1:3
    x0(1,i)=i
    x0(1,i+3)=i
    x0(1,i+6)=i
    x0(2,i)=i
    x0(2,i+3)=i+1
    x0(2,i+6)=i+2
end

[y1,x1]=csim(u,t,G,x0(:,1));

[y2,x2]=csim(u,t,G,x0(:,2));

[y3,x3]=csim(u,t,G,x0(:,3));

[y4,x4]=csim(u,t,G,x0(:,4));

[y5,x5]=csim(u,t,G,x0(:,5));

[y6,x6]=csim(u,t,G,x0(:,6));

[y7,x7]=csim(u,t,G,x0(:,7));

[y8,x8]=csim(u,t,G,x0(:,8));

[y9,x9]=csim(u,t,G,x0(:,9));
```

```
p=roots(d)

r1 = real(p(1))

r2 = real(p(2))

i1 = imag(p(1))

i2 = imag(p(2))

R(1,1) = r1

R(1,2) = r2

I(1,1) = i1

I(1,2) = i2

scf(1)
xset('thickness',2)
xset('font size',4)
plot([y1 y2 y3 y4 y5 y6 y7 y8 y9],[x1(2,:) x2(2,:) x3(2,:) x4(2,:) x5(2,:)
    x6(2,:) x7(2,:) x8(2,:) x9(2,:)],'linewidth',3)
xtitle('Representação do espaço de estados','Deformacao da mola','
    Velocidade')

scf(2)
a=get('current_axes')
a.x_location = 'middle'
a.y_location = 'middle'
a.data_bounds=[-20,-10;20,10]
plot(R,I,'o','linewidth',1)
title('Polos do sistema')
```

Figura 5 – $\zeta = 0,5$

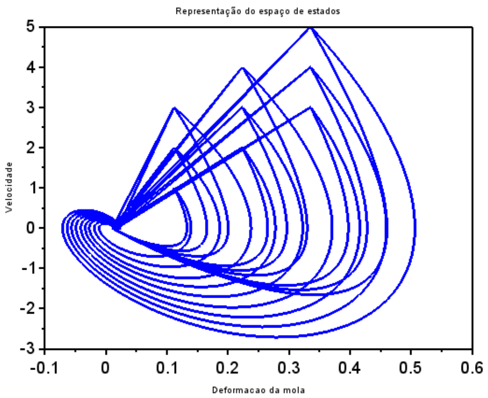


(a) Deformação em função da velocidade

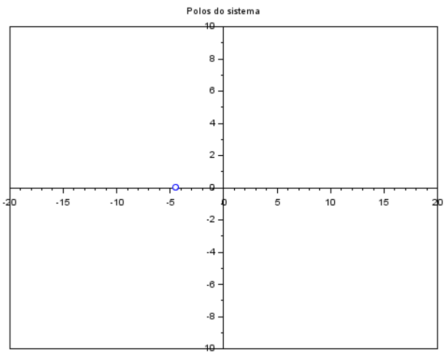


(b) Pólos complexos

Figura 6 – $\zeta = 1,0$

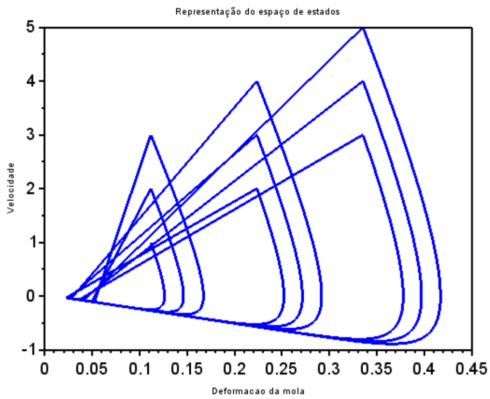


(a) Deformação em função da velocidade

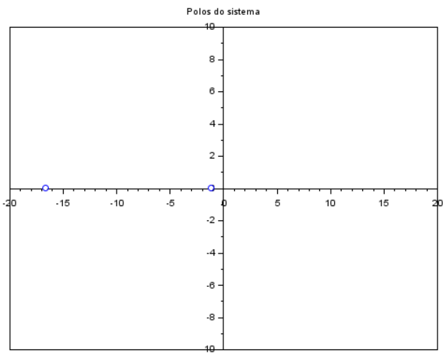


(b) Pólos reais e iguais

Figura 7 – $\zeta = 2,0$



(a) Deformação em função da velocidade



(b) Pólos reais e distintos