

Gabriel Jenner de Faria Orsi
Nº USP 10772800

Lista E - Modelagem de Sistemas Dinâmicos

Brasil

2020

Gabriel Jenner de Faria Orsi
Nº USP 10772800

Lista E - Modelagem de Sistemas Dinâmicos

Apresentação da Lista E da disciplina
PME3380 - Modelagem de Sistemas Dinâmicos

Universidade de São Paulo
Escola Politécnica
PME3380 - Modelagem de Sistemas Dinâmicos

Orientador: Prof. Dr. Agenor de Toledo Fleury e Prof. Dr. Decio Crisol
Donha

Brasil
2020

Lista de ilustrações

Figura 1 – Sistema massa mola amortecedor	1
Figura 2 – $\zeta = 0,5$	2
Figura 3 – $\zeta = 1$	3
Figura 4 – $\zeta = 1,5$	3
Figura 5 – Polos e espaço de estados para $\zeta = 0,5$	7
Figura 6 – Polos e espaço de estados para $\zeta = 1$	8
Figura 7 – Polos e espaço de estados para $\zeta = 1,5$	8

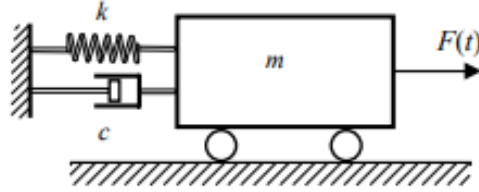
Sumário

1	SISTEMA MASSA MOLA AMORTECEDOR	1
1.1	Modelagem	1
1.2	Resultados	2
2	COMPARAÇÃO DE MÉTODOS - AUTOVALORES E POLOS . . .	5
3	ANÁLISE DE POLOS E ESPAÇO DE ESTADOS	7
3.1	Sistema subamortecido - $\zeta < 1$	7
3.2	Sistema criticamente amortecido - $\zeta = 1$	8
3.3	Sistema superamortecido - $\zeta > 1$	8
3.4	Código	8

1 Sistema massa mola amortecedor

O sistema a ser modelado é o que segue na figura 1.

Figura 1 – Sistema massa mola amortecedor



1.1 Modelagem

A equação diferencial que rege o sistema é a seguinte.

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

Rearranjando a equação 1.1 para o espaço de estados (EE), tem-se.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (1.2)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m}(-bx_2 - kx_1 + F(t)) \end{bmatrix} \quad (1.3)$$

Transformando a equação 1.3 na forma

$$\dot{x} = Ax + Bu$$

sendo $u = F(t)$ e $x = [x_1, x_2]$, chega-se a.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \quad (1.4)$$

A equação de saídas dada por

$$y = Cx + Du$$

fica.

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u \quad (1.5)$$

Aplicando a transformada de Laplace na equação 1.3, e transferindo o problema do domínio do tempo para o domínio da frequência, obtém-se.

$$\mathcal{L}_1 : sX_1 = X_2 \quad (1.6)$$

$$\mathcal{L}_2 : sX_2 = -\frac{k}{m}X_1 - \frac{b}{m}X_2 + \frac{F}{m} \quad (1.7)$$

A saída esperada do sistema é $Y = X_1$, com entrada $F = U$. Assim é possível obter a função de transferência.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{ms^2 + bs + k} \quad (1.8)$$

A partir da modelagem aqui descrita foi feita a simulação no *scilab*.

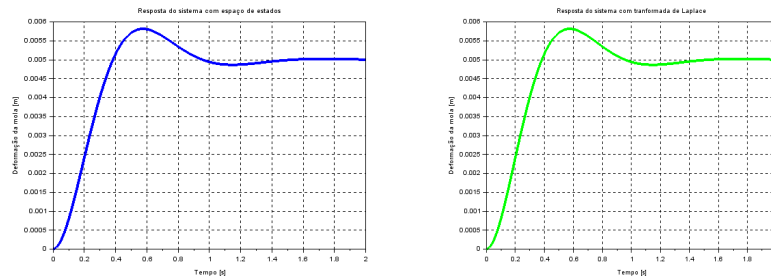
1.2 Resultados

O sistema foi simulado tanto no domínio do tempo (através de EE) quanto no domínio da frequência (através da função de transferência). As simulações foram feitas com

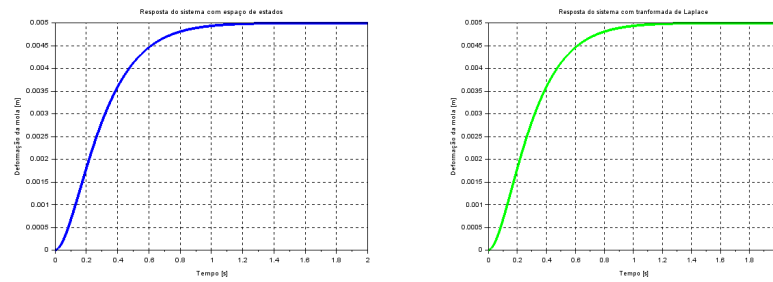
$$\zeta = \frac{b}{2\sqrt{km}}$$

, para valores de $\zeta = 0,5$, $\zeta = 1$ e $\zeta = 1,5$. Segue os resultados e o código usado.

Figura 2 – $\zeta = 0,5$

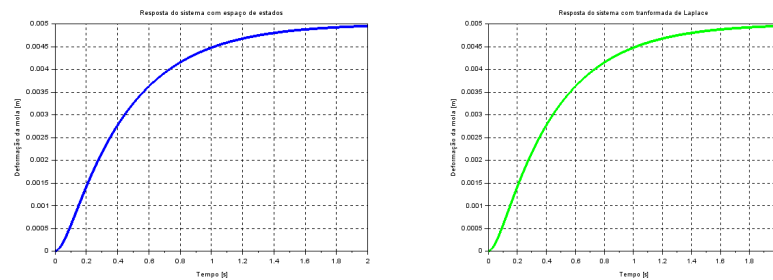


(a) Domínio do tempo - espaço de estados (b) Domínio da frequência - função de transferência

Figura 3 – $\zeta = 1$ 

(a) Domínio do tempo - espaço de estados

(b) Domínio da frequência - função de transferência

Figura 4 – $\zeta = 1,5$ 

(a) Domínio do tempo - espaço de estados

(b) Domínio da frequência - função de transferência

Segue o código utilizado.

```

1 clc()
2 clear()
3
4 //par metros
5 Z=1.5 //fator de amortecimento
6 k=200 //constante elástica [N/m]
7 m=5 //massa do corpo [kg]
8 b=Z*2*sqrt(k*m) //constante de amortecimento [Ns/m]
9
10
11 //vetor tempo
12 t=0:0.01:2
13
14 //entrada
15 u=ones(t)
16
17 x0=[0;0]
18
19 //RESOLU O POR ESPA O DE ESTADOS – dominio do tempo

```

```
20 //matrizes do sistema
21 A=[0 1;-k/m -b/m]
22 B=[0;1/m]
23 C=[1 0]
24 D=[0]
25
26 //sistema
27 massa_mola_amortecedor=syslin('c',A,B,C,D)
28
29 //simulando com a função csim
30 [y,x]=csim(u,t, massa_mola_amortecedor,x0)
31
32 //RESOLU O POR FUNCAO DE TRANSFERENCIA – dominio da frequencia
33 //polinomio s
34 s=poly(0,'s')
35
36 //funcao de tranferencia
37 f='1/(m*s^2+b*s+k)'
38
39 //calculando a expressao literal
40 g=evstr(f)
41
42 //monstando a funcao de transferencia
43 G=syslin('c',g)
44
45 //simulacao do sistema
46 [z]=csim(u,t,g,x0)
47
48 //PLOTAGENS
49 //plotagem do dominio do tempo
50 f1=scf(1)
51 plot(t,y,'LineWidth',4)
52 xtitle('Resposta do sistema com espaço de estados','Tempo [s]','Deformação
    da mola [m]')
53 xgrid()
54
55 //plotagem do dominio de amostras
56 f2=scf(2)
57 plot(t,z,'g','LineWidth',4)
58 xtitle('Resposta do sistema com tranformada de Laplace','Tempo [s]','
    Deformação da mola [m]')
59 xgrid()
```


2 Comparação de métodos - autovalores e polos

Para a obtenção dos autovalores de $[A]$ basta fazer:

$$\det(A - \lambda I) = 0$$

chegando no seguinte polinômio.

$$m\lambda^2 + b\lambda + k = 0 \quad (2.1)$$

Comparando o polinômio da equação 2.1 com o polinômio no denominador da equação 1.8, percebe-se que são iguais. Sendo assim os polos do sistema (raízes do polinômio do denominador da função de transferência) são iguais aos autovalores da matriz $[A]$.

Para $\zeta < 1$ com os parâmetros $m = 5 \text{ kg}$ e $k = 200 \text{ N/m}$, foram obtidos os polos e autovalores mostrados na tabela 1.

Tabela 1 – Polos e autovalores

		Raízes para $\zeta < 1$	Módulo
Autovalores	λ_1	-3.162 - 5.477i	6.325
	λ_2	-3.162 + 5.477i	6.325
Polos	s_1	-3.162 - 5.477i	6.325
	s_2	-3.162 + 5.477i	6.325

A frequência natural do sistema é dada por.

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{200}{5}} = 6.325 \quad (2.2)$$

Assim, a frequência natural do sistema corresponde ao módulo do polo ou do autovalor da matriz $[A]$. Além disso, ao dividir o módulo da parte real pelo módulo do número complexo (polo) se obtém o fator de amortecimento.

$$\frac{3,1625}{6,325} = \zeta = 0,5$$

O código utilizado para a obtenção das raízes é o seguinte.

```

1 // calculo dos polos
2 if Z<1 then // polos complexos
3     r1=-b/(2*m);
```

```
4      r2=r1 ;
5      i1=sqrt ( abs ( b^2-4*m*k ) ) / ( 2*m ) ;
6      i2=-sqrt ( abs ( b^2-4*m*k ) ) / ( 2*m ) ;
7  elseif Z==1 then //polos reais e iguais
8      r1=-b/(2*m) ;
9      r2=r1 ;
10     i1=0;
11     i2=i1 ;
12 elseif Z>1 then //polos reais e distintos
13     r1=(-b+sqrt ( b^2-4*m*k ) ) / ( 2*m ) ;
14     r2=(-b+sqrt ( b^2-4*m*k ) ) / ( 2*m ) ;
15     i1=0;
16     i2=i1 ;
17 end
18 s1=complex ( r1 , i1 )
19 s2=complex ( r2 , i2 )
20 disp ( s1 , s2 )
21 disp ( abs ( s1 ) , abs ( s2 ) )
22 disp ( sqrt ( k/m ) ) //frequencia natural de oscilacao
```

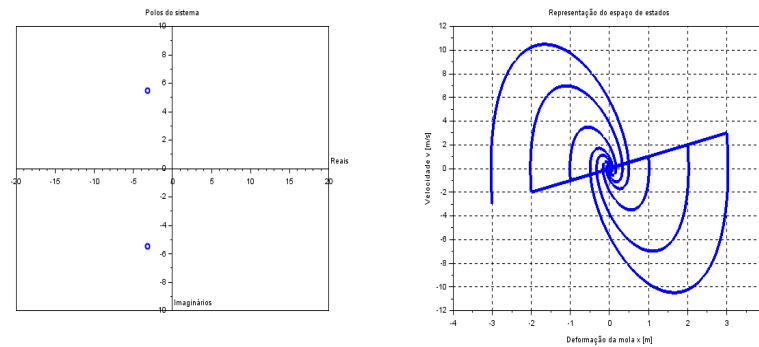
3 Análise de polos e espaço de estados

Nas seções subsequentes é ilustrado o espaço de estados para cada valor de ζ sob diferentes condições iniciais diferentes de zero e diferentes entre si, bem como os polos no plano dos complexos. Os valores de $x(t=0)$ e $\dot{x}(t=0)$ variam de -3 a 3 sem incluir o caso [00] para todos os casos de amortecimento.

3.1 Sistema subamortecido - $\zeta < 1$

Segue o resultado do espaço de estados e os polos no plano complexo.

Figura 5 – Polos e espaço de estados para $\zeta = 0,5$

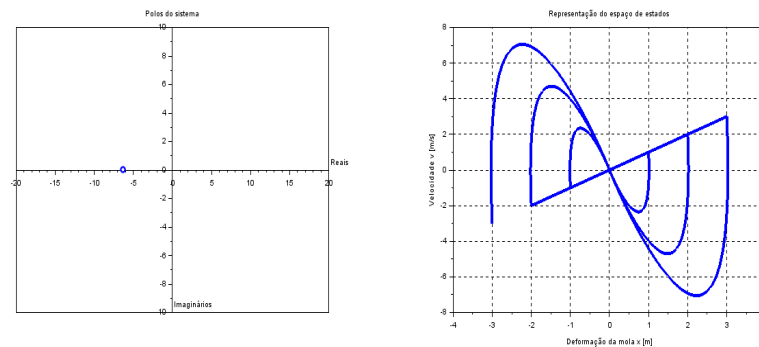


(a) Polos no plano complexo

(b) Representação do espaço de estados - velocidade em função da deformação

3.2 Sistema criticamente amortecido - $\zeta = 1$

Figura 6 – Polos e espaço de estados para $\zeta = 1$

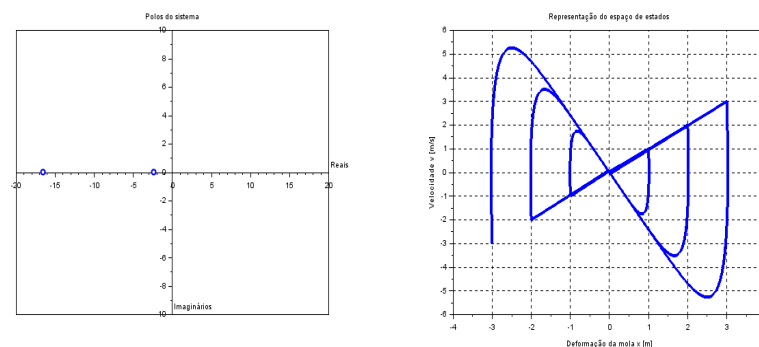


(a) Polos no plano complexo

(b) Representação do espaço de estados - velocidade em função da deformação

3.3 Sistema superamortecido - $\zeta > 1$

Figura 7 – Polos e espaço de estados para $\zeta = 1,5$



(a) Polos no plano complexo

(b) Representação do espaço de estados - velocidade em função da deformação

3.4 Código

Segue o código utilizado para a obtenção dos resultados

```
1 clc()
2 clear()
3
4 //par metros
5 Z=1 //fator de amortecimento
```

```

6 k=200 //constante elástica [N/m]
7 m=5 //massa do corpo [kg]
8 b=Z*2*sqrt(k*m) //constante de amortecimento [Ns/m]
9
10 //raizes do polinomio do denominador — forma complexa
11 if Z<1 then //polos complexos
12     r1=-b/(2*m);
13     r2=r1;
14     i1=sqrt(abs(b^2-4*m*k))/(2*m);
15     i2=-sqrt(abs(b^2-4*m*k))/(2*m);
16 elseif Z==1 then //polos reais e iguais
17     r1=-b/(2*m);
18     r2=r1;
19     i1=0;
20     i2=i1;
21 elseif Z>1 then //polos reais e distintos
22     r1=(-b+sqrt(b^2-4*m*k))/(2*m);
23     r2=(-b-sqrt(b^2-4*m*k))/(2*m);
24     i1=0;
25     i2=i1;
26 end
27 s1=complex(r1, i1)
28 s2=complex(r2, i2)
29
30
31 //vetor tempo
32 t=0:0.01:2
33
34 //entrada
35 u=zeros(t)
36
37 //condicoes iniciais diferentes de 0
38 x0=zeros(2,7)
39 for i=1:7
40     x0(1,i)=i-4
41     x0(2,i)=i-4
42 end
43
44 //RESOLU O POR ESPA O DE ESTADOS
45 //matrizes do sistema
46 A=[0 1;-k/m -b/m]
47 B=[0;1/m]
48 C=[1 0]
49 D=[0]
50
51 //sistema
52 massa_mola_amortecedor=syslin('c',A,B,C,D)

```

```

53
54 //simulando com a função csim
55 [y1,x1]=csim(u,t, massa_mola_amortecedor,x0(:,1))
56 [y2,x2]=csim(u,t, massa_mola_amortecedor,x0(:,2))
57 [y3,x3]=csim(u,t, massa_mola_amortecedor,x0(:,3))
58 [y4,x4]=csim(u,t, massa_mola_amortecedor,x0(:,5))
59 [y5,x5]=csim(u,t, massa_mola_amortecedor,x0(:,6))
60 [y6,x6]=csim(u,t, massa_mola_amortecedor,x0(:,7))
61
62
63 f1=scf(1)
64 plot([y1 y2 y3 y4 y5 y6],[x1(2,:) x2(2,:) x3(2,:) x4(2,:) x5(2,:) x6(2,:)],
        'linewidth',4)
65 xgrid()
66 xtitle('Representação do espaço de estados','Deformação da mola x [m]',
        'Velocidade v [m/s]')
67
68 f2=scf(2)
69 plot([r1 r2],[i1 i2], 'o', 'linewidth',2)
70 a=get('current_axes')
71 a.x_location='middle'
72 a.y_location='middle'
73 a.data_bounds=[-18,-10;18,10]
74 title('Polos do sistema')
75 xlabel('Reais','position',20)
76 ylabel('Imaginários','position',[0,-10],'rotation',0)

```