

Escola Politécnica da Universidade de São Paulo

Modelagem de sistemas dinâmicos

Lista G

Vítor Facchini

10772605

Professor: Décio Crisol e Agenor Fleury

São Paulo

2020

Sumário

1	Equacionamento	3
2	Simulações	4
2.1	Condição 1	4
2.2	Condição 2	5
2.3	Condição 3	7
3	Código utilizado	9

Lista de Figuras

1.1	Sistema de meio carro analisado	3
2.1	Resultados obtidos	4
2.2	Resultados obtidos	5
2.3	Resultados obtidos	6
2.4	Resultados obtidos	6
2.5	Resultados obtidos	7
2.6	Resultados obtidos	7
2.7	Resultados obtidos	8
2.8	Resultados obtidos	8

1 Equacionamento

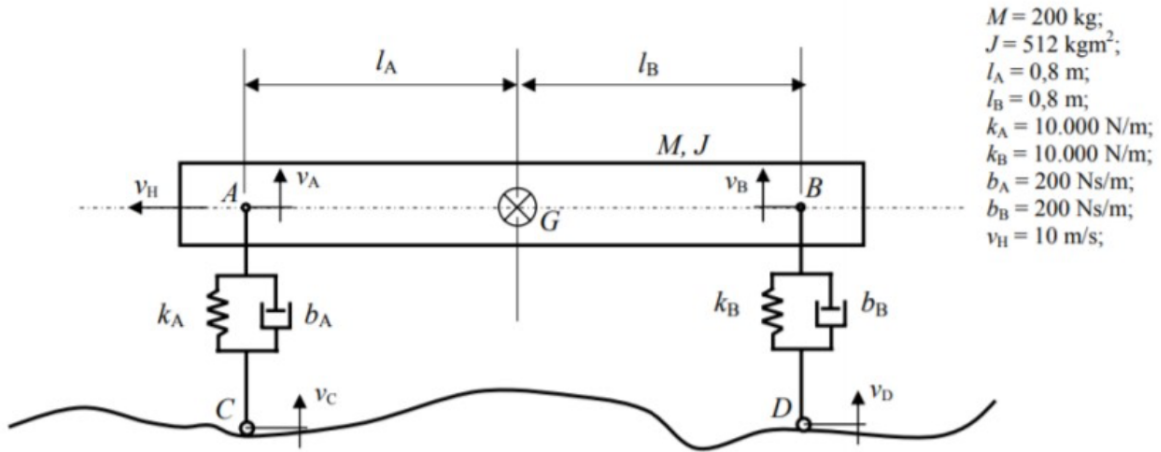


Figura 1.1: Sistema de meio carro analisado

$$M_G^{ext} = J\dot{\omega} = l_A k_A (x_A - x_C) - l_B k_B (x_B - x_D) + l_A b_A (\dot{x}_A - \dot{x}_C - l_B b_B (\dot{x}_B - \dot{x}_D)$$

$$m\ddot{a}_G = -k_A (x_A - x_C) - b_A (\dot{x}_A - \dot{x}_C - k_B (x_B - x_D) - b_B (\dot{x}_B - \dot{x}_D)$$

Logo: $x_A = \dot{V}_G - l_A \omega$ e $x_B = V_G - l_B \omega$.

Assim:

$$\dot{V}_G = \frac{k_A x_A}{m} - \frac{b_B x_B}{m} - \frac{b_A (V_G - l_A \omega)}{m} - \frac{b_B (V_G + l_B \omega)}{m} + \frac{k_A x_C}{m} + \frac{k_B x_D}{m} + \frac{b_A \dot{x}_C}{m} + \frac{b_B \dot{x}_D}{m}$$

$$\dot{\omega} = \frac{l_A k_A x_A}{J} - \frac{l_B k_B x_B}{J} + \frac{l_A b_A (V_G - l_A \omega)}{J} - \frac{l_B b_B (V_G + l_B \omega)}{J} - \frac{l_A k_A x_C}{J} + \frac{l_B k_B x_D}{J} - \frac{l_A b_A \dot{x}_C}{J} + \frac{l_B b_B \dot{x}_D}{J} - \frac{l_A b_B \dot{x}_D}{J}$$

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

$$\begin{bmatrix} \dot{x}_A \\ \dot{x}_B \\ \dot{V}_G \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & -l_A \\ 0 & 0 & 1 & l_B \\ -\frac{k_A}{m} & -\frac{k_B}{m} & -\frac{b_A + b_B}{m} & \frac{b_A l_A - b_B l_B}{m} \\ \frac{l_A k_A}{J} & -\frac{l_B k_B}{J} & \frac{l_A b_A - l_B b_B}{J} & -\frac{b_A l_A^2 + b_B l_B^2}{J} \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ V_G \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{k_A}{m} & \frac{k_B}{m} & \frac{b_A}{m} & \frac{b_B}{m} \\ -\frac{l_A k_A}{J} & \frac{l_B k_B}{J} & -\frac{l_A b_B}{J} & \frac{l_B b_B}{J} \end{bmatrix} \begin{bmatrix} x_B \\ x_D \\ \dot{x}_C \\ \dot{x}_D \end{bmatrix}$$

2 Simulações

A modelagem descrita no capítulo anterior foi então simulada para as condições descritas nas seções seguintes.

2.1 Condição 1

$$u = [v_C \quad v_D]$$

Restrições:

- $t < 0 \Rightarrow v_C = 0$;
- $t > 1 \Rightarrow v_C = 1$;
- $t < t_d \Rightarrow v_D = 0$;
- $t > t_d \Rightarrow v_D = 1$

Os resultados estão ilustrados nas Figuras 2.1 e 2.2

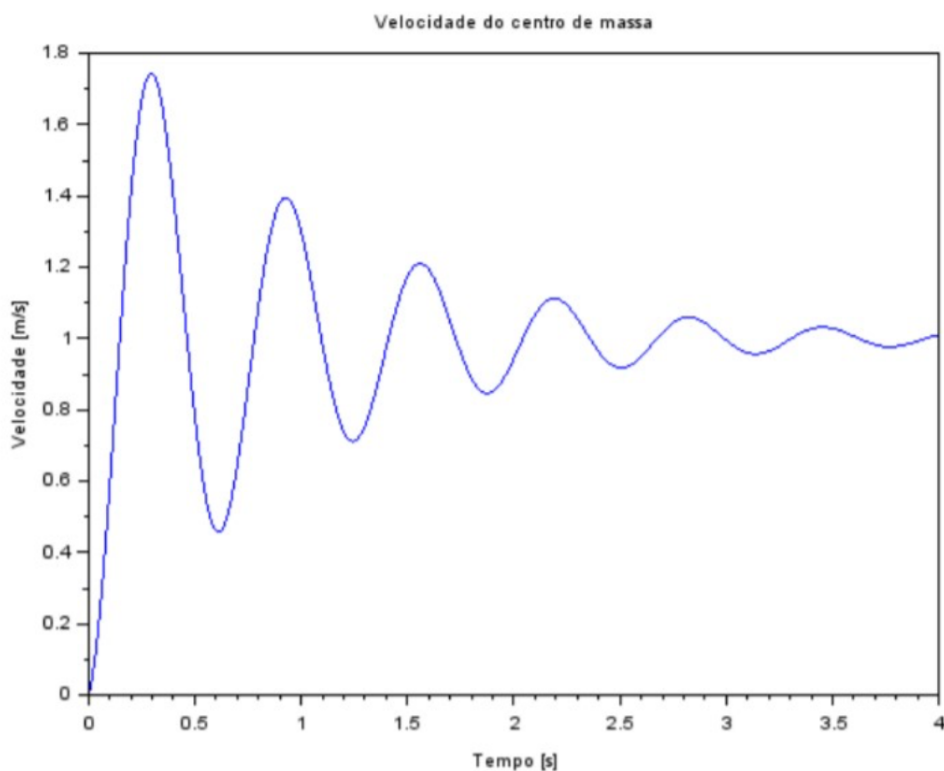


Figura 2.1: Resultados obtidos

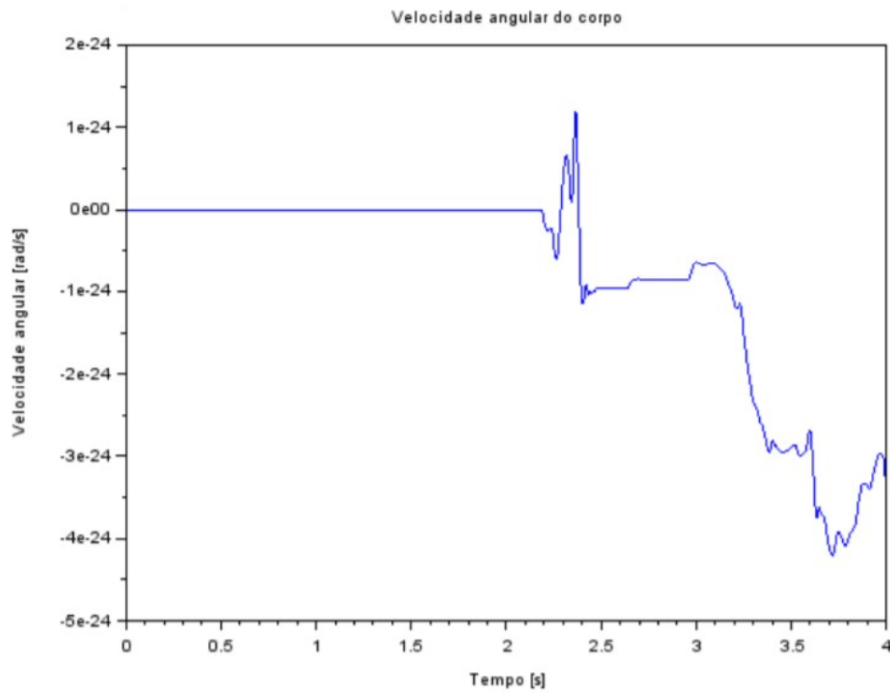


Figura 2.2: Resultados obtidos

2.2 Condição 2

$$u = [\sin 9,8995t \quad \sin 9,8995t]$$

Obteve-se as Figuras 2.3 e 2.4 como resultados.

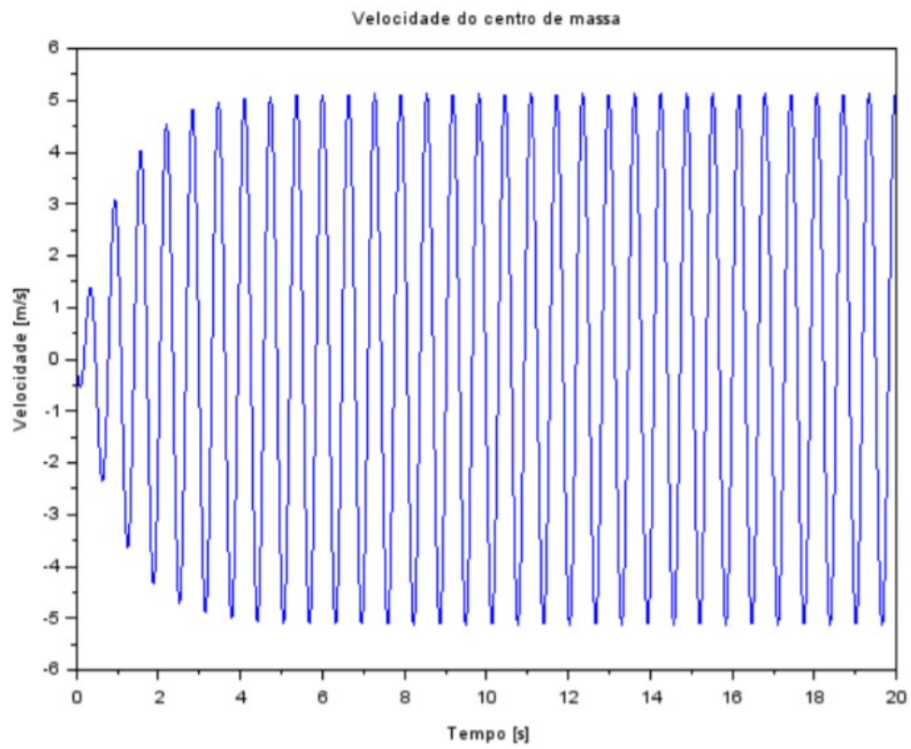


Figura 2.3: Resultados obtidos

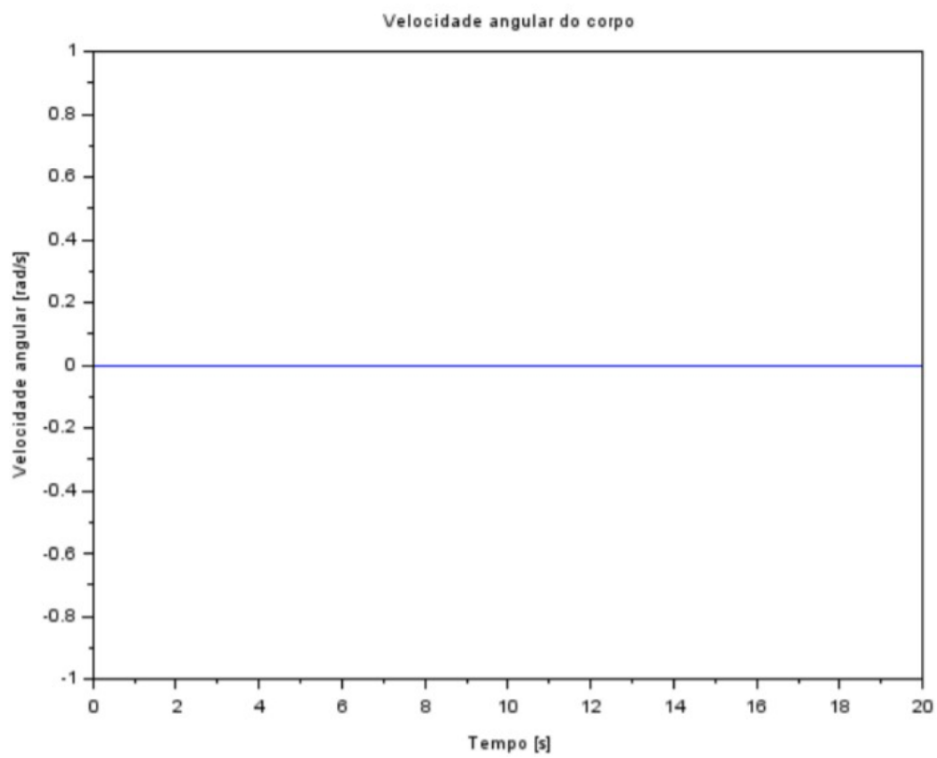


Figura 2.4: Resultados obtidos

2.3 Condição 3

$$u = [\sin 4,9875t \quad -\sin 4,9875t]$$

Obteve-se as Figuras 2.5 e 2.7 como resultados.

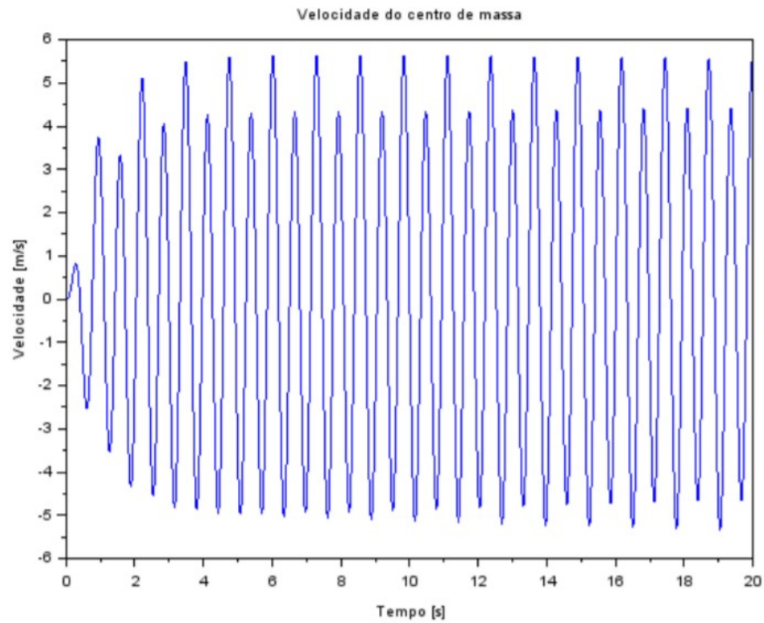


Figura 2.5: Resultados obtidos

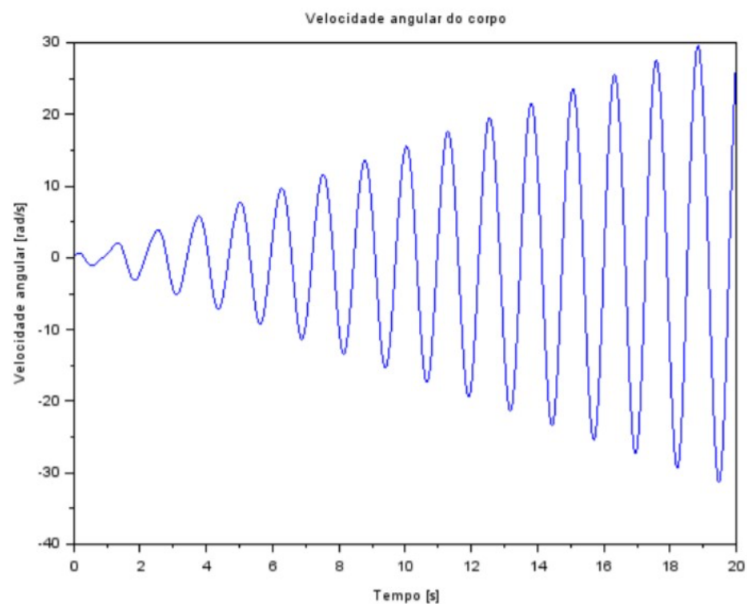


Figura 2.6: Resultados obtidos

No domínio da frequência:

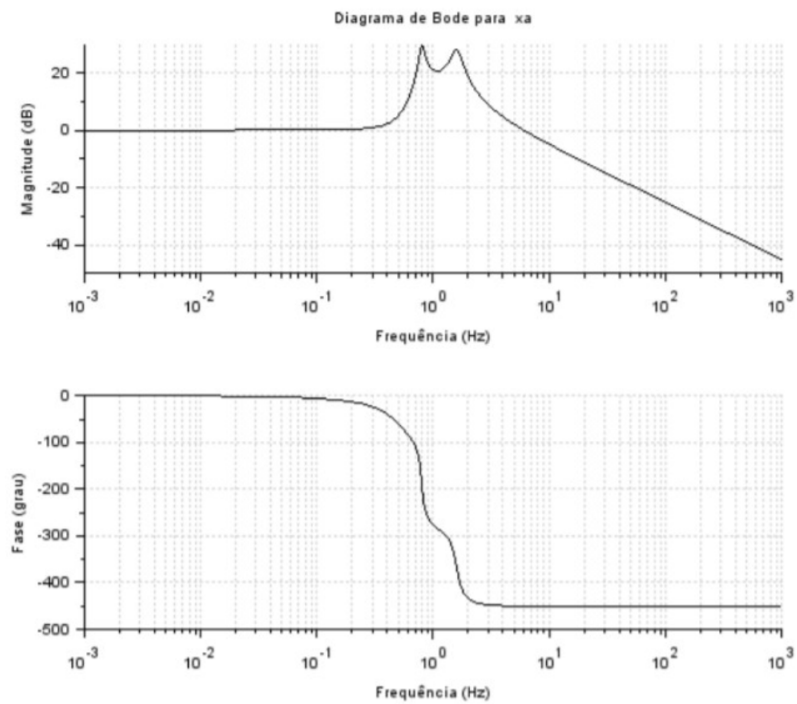


Figura 2.7: Resultados obtidos

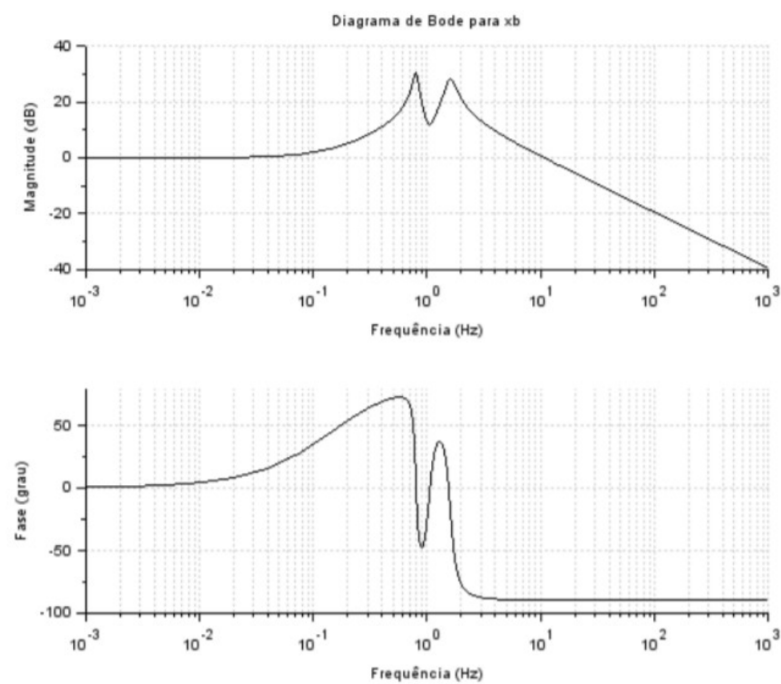


Figura 2.8: Resultados obtidos

3 Código utilizado

```

1  kA = 10000;
2  kB = 10000;
3  bA = 200;
4  M = 200;
5  J = 512;
6  lA = 0.8;
7  lB = 0.8;
8
9  td = (lA + lB)/vH; //Tempo de resposta de D [s]
10
11 t_inicial = 0;
12 t_final = 1;
13 t = linspace(t_inicial,t_final,1000); simulacao = 3;
14 xA0 = 0; xB0 = 0; vG0 = 0; w0 = 0;
15
16 if simulacao == 1 then
17 function fun=u1(t), fun = t, endfunction if t < td then
18 function fun=u2(t), fun = 0, endfunction else
19 function fun=u2(t), fun = t, endfunction end
20 function fun=u3(t), fun = 1, endfunction if t < td then
21 function fun=u4(t), fun = 0, endfunction else
22 function fun=u4(t), fun = 1, endfunction end
23 elseif simulacao == 2 then function fun=u1(t), fun function fun=u2(t
    ), fun function fun=u3(t), fun function fun=u4(t), fun
24 = -cos(9.8995*t)/9.8995, endfunction = -cos(9.8995*t)/9.8995,
    endfunction = sin(9.8995*t), endfunction
25 = sin(9.8995*t), endfunction
26 elseif simulacao == 3 then
27 function fun=u1(t), fun = -cos(9.8995*t)/4.9875, endfunction
    function fun=u2(t), fun = cos(4.9875*t)/4.9875, endfunction
    function fun=u3(t), fun = sin(4.9875*t), endfunction
28 function fun=u4(t), fun = -sin(4.9875*t), endfunction end
29
30 funcprot(0)
31 function dy=estados(t, y)
32 dy(1) = y(3) - lA*y(4);
33 dy(2) = y(3) + lB*y(4);
34 dy(3) = -(kA/M)*y(1) - (kB/M)*y(2) - ((bA + bB)/M)*y(3) + ((bA*lA -
35 bB*lB)/M)*y(4) + (kA/M)*u1(t) + (kB/M)*u2(t) + (bA/M)*u3(t) + (bB/M)
    *u4(t);
36 dy(4) = (lA*kA/J)*y(1) - (lB*kB/J)*y(2) + ((lA*bA - lB*bB)/J)*y(3) -
    ((bA*lA^2 - bB*lB^2)/M)*y(4) - (lA*kA/J)*u1(t) + (lB*kB/J)*u2(t)

```

```

    - (lA*bA/J)*u3(t) + (lB*bB/J)*u4(t);
37 endfunction
38 result = ode([xA0;xB0;vG0;w0],0,t,estados);
39 xA = result(1,:); xB = result(2,:); vG = result(3,:); w = result
    (4,:);
40 scf(1)
41 xtitle("Velocidade do centro de massa"); xlabel("Tempo [s]");
42 ylabel("Velocidade [m/s]");
43 plot(t,vG);
44 scf(2)
45 xtitle("Velocidade angular do corpo"); xlabel("Tempo [s]"); ylabel("
    Velocidade angular [rad/s]"); plot(t,w);
46
47 A = [0,0,1,-lA;0,0,1,lB;-kA/M,-kB/M,-(bA+bB)/M,(bA*lA - bB*lB)/M;lA*
    kA/J,-lB*kB/J,(lA*bA-lB*bB)/J,-(bA*lA^2 + bB*lB^2)/J];
48 B = [0,0,0,0;0,0,0,0;kA/M,kB/M,bA/M,bB/M;-lA*kA/J,lB*kB/J,-lA*bA/J,
    lB*bB/J]; sl = syslin('c',A,B,[1,1,1,1]);
49 h = ss2tf(sl);
50 scf(3);
51 bode(h(1,1));
52 xtitle("Diagrama de Bode para xa");
53 scf(4);
54 bode(h(1,2));
55 xtitle("Diagrama de Bode para xb"); scf(5);
56 bode(h(1,3));
57 xtitle("Diagrama de Bode para vg");
58 scf(6);
59 bode(h(1,4));
60 xtitle("Diagrama de Bode para w");

```