

LISTA G

PME 3380- Modelagem de Sistemas Dinâmicos



Escola Politécnica

Universidade de São Paulo

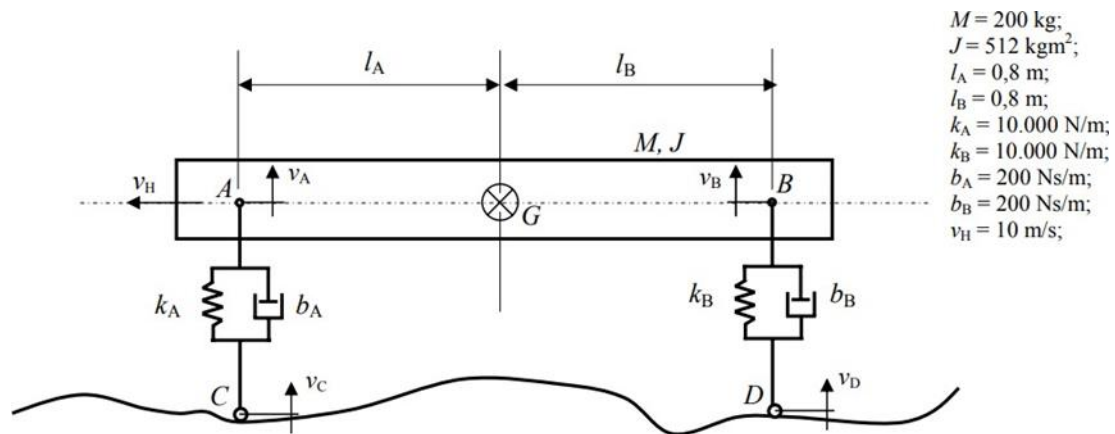
São Paulo

2020

Gabriel Rodrigues Camargo

NUSP: 10772460

Parte 1- Modelagem do sistema



Para esse relatório é realizado a simulação de um sistema de $\frac{1}{2}$ carro, para isso são utilizadas 4 variáveis de estado listadas a seguir:

- velocidade vertical v_{tt} do centro de massa G;
- velocidade angular ω de AB em torno de G;
- elongação x_A da mola de rigidez k_A ;
- elongação x_B da mola de rigidez k_B .

Além disso, as saídas serão velocidade vertical v_{tt} do centro de massa e velocidade angular ω de AB em torno de G, já as entradas serão as velocidades verticais (v_C e v_D) dos pontos C e D. Também se destacam as seguintes hipóteses simplificadoras do sistema:

- Movimento apenas no plano dado
- Molas e amortecedores ideais
- As retas que ligas as molas e amortecedores se mantem na vertical
- $\sin(\alpha) \approx \alpha; \cos(\alpha) \approx 1$

Para iniciar a modelagem do sistema usa-se o Teorema do Movimento do Baricentro (TMB) e Teorema da Quantidade de Momento Angular (TQMA) obtendo-se as seguintes equações:

$$M\dot{v}_G = -Mg - k_A x_A - k_B x_B - b_A \dot{x}_A - b_B \dot{x}_B \quad (1)$$

$$J\dot{\omega} = l_A(k_A x_A + b_A \dot{x}_A) - l_B(k_B x_B + b_B \dot{x}_B) \quad (2)$$

Além disso, são calculas as velocidade e acelerações em A e B:

$$x_A = x_{A,0} + x_G - x_C - l_A \theta \quad (3)$$

$$\dot{x}_A = v_G - v_C - l_A \omega \quad (4)$$

$$x_B = x_{B,0} + x_G - x_D - l_B \theta \quad (5)$$

$$\dot{x}_B = v_G - v_D - l_B \omega \quad (6)$$

Dessa forma, usando as equações obtidas pode-se chegar ao seguinte sistema:

$$\begin{cases} M\dot{v}_G = -Mg - k_A x_A - k_B x_B - b_A \dot{x}_A - b_B \dot{x}_B \\ J\dot{\omega} = l_A(k_A x_A + b_A \dot{x}_A) - l_B(k_B x_B + b_B \dot{x}_B) \\ \dot{x}_A = v_G - v_D - l_B \omega \\ \dot{x}_B = v_G - v_D - l_B \omega \end{cases} \quad (7)$$

Assim, escreve-se o espaço de estados definindo os vetores $X = [x_A \ x_B \ v_G \ \omega]^T$, $U = [v_C \ v_D]^T$ e saída $Y = [v_G \ \omega]^T$, que gera o seguinte sistema

$$\dot{X} = \begin{bmatrix} 0 & 0 & 1 & l_A \\ 0 & 0 & 1 & l_B \\ \frac{-k_A}{M} & \frac{k_B}{M} & \frac{-b_A - b_B}{M} & \frac{l_A b_A + l_B b_B}{M} \\ \frac{l_A k_A}{J} & \frac{-l_B k_B}{J} & \frac{l_A b_A + l_B b_B}{J} & \frac{-l_A^2 b_A - l_B^2 b_B}{J} \end{bmatrix} X + \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ \frac{b_A}{M} & \frac{b_B}{M} \\ \frac{-l_A b_A}{J} & \frac{-l_B b_B}{J} \end{bmatrix} U \quad (8)$$

$$Y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} U \quad (9)$$

Dessa forma pode se escrever o sistema matricial do tipo:

$$\dot{X} = AX + BU \quad (10)$$

$$Y = CX + DU \quad (11)$$

Parte 2- Simulação em Scilab

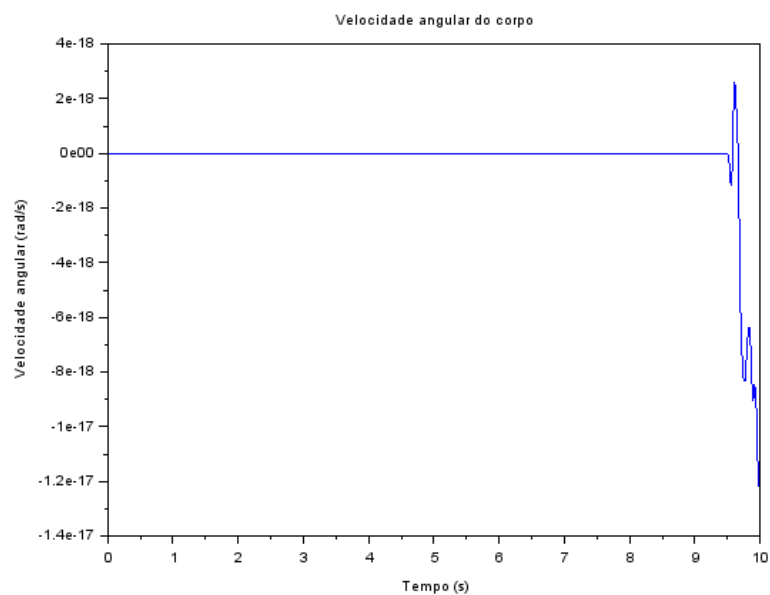
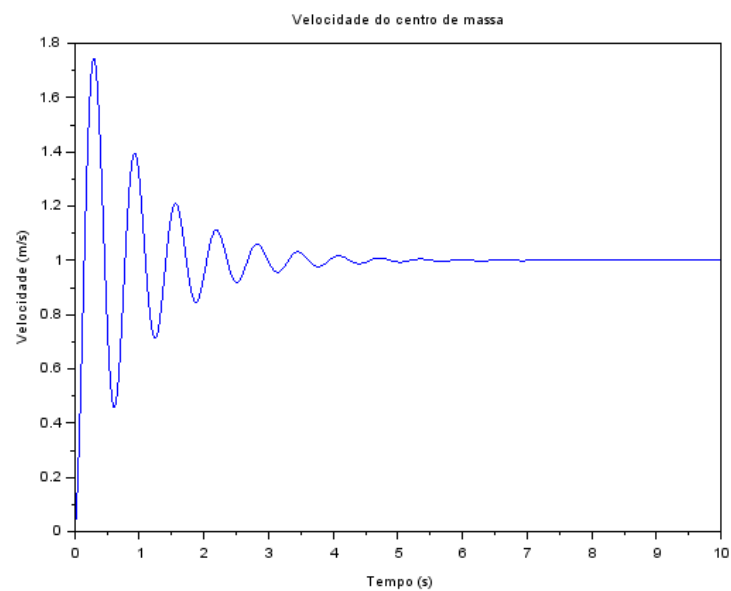
A partir do sistema de equações obtido anteriormente foram feitas 3 simulações do problema com as seguintes condições:

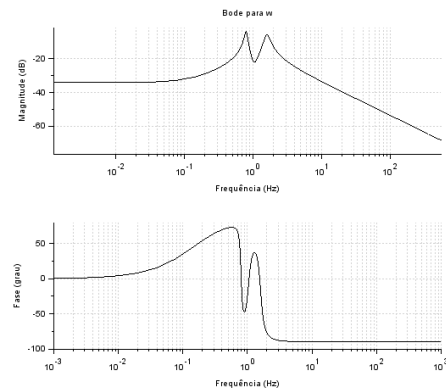
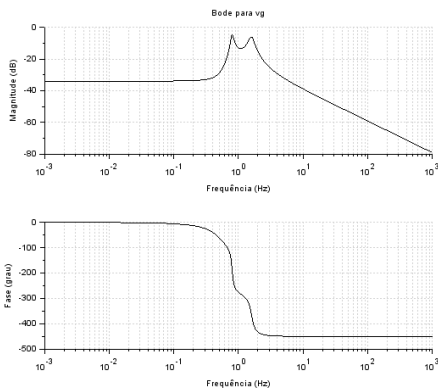
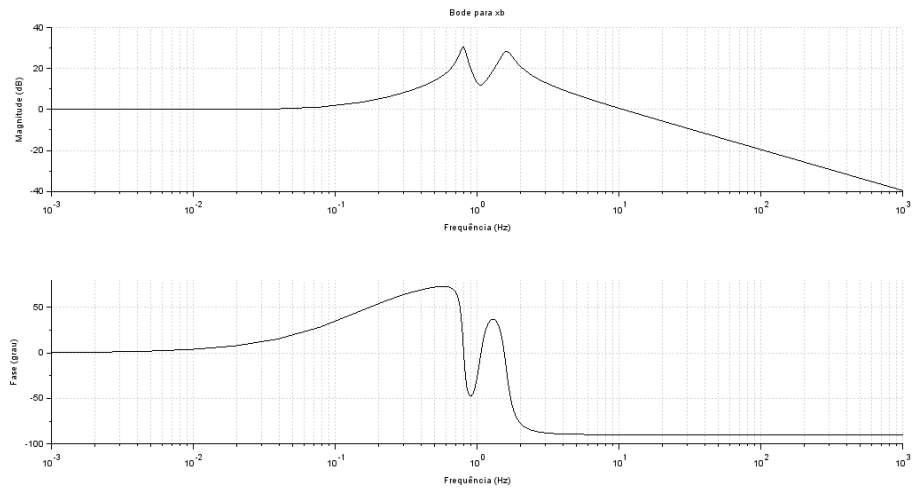
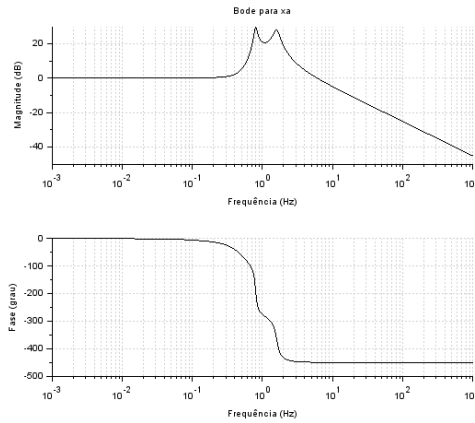
Simulação 1

Condições impostas:

$$v_D = \begin{cases} 1 & \text{se } t \geq t_D \\ 0 & \text{se } t < t_D \end{cases}; v_C = \begin{cases} 1 & \text{se } t \geq 0 \\ 0 & \text{se } t < 0 \end{cases}$$

Resultados obtidos:





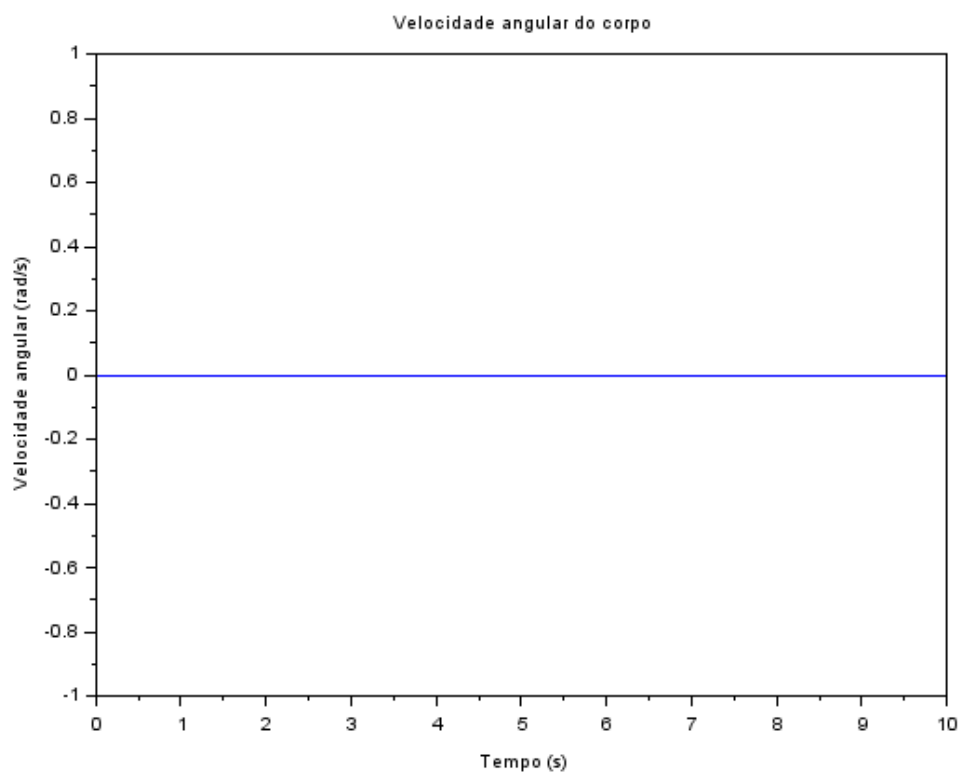
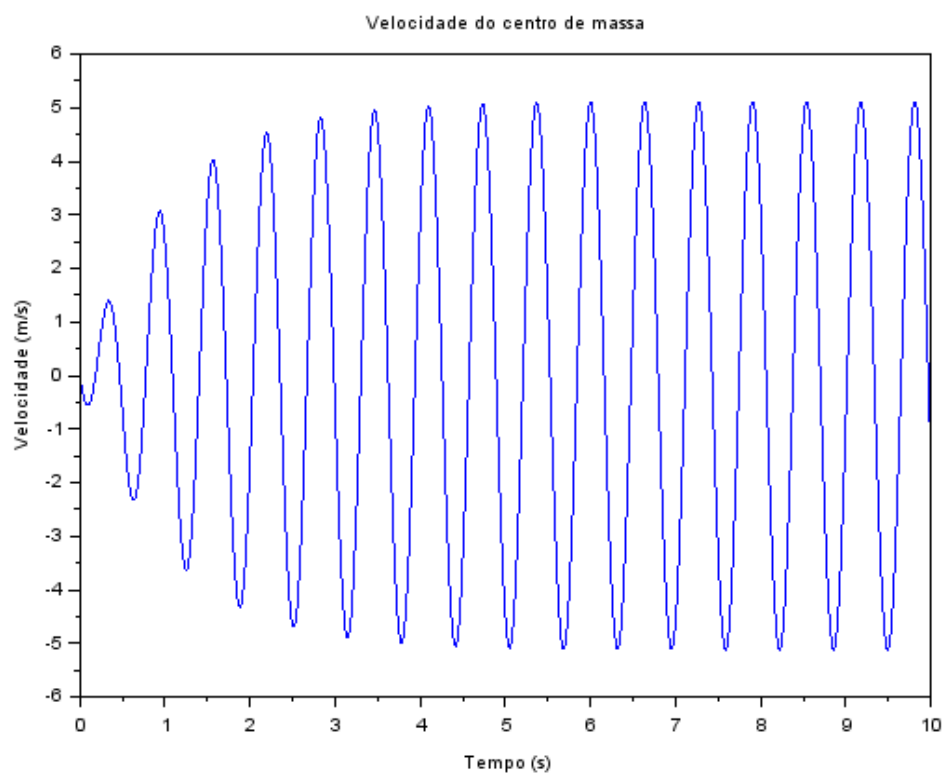
Nesse cenário pode-se perceber a estabilidade do sistema pela análise dos gráficos de seu movimento, tendendo a cessar sua velocidade do centro de massa G e sua velocidade angular. Nos gráficos de bode fica visível principalmente as velocidades de ressonância nos picos de magnitude.

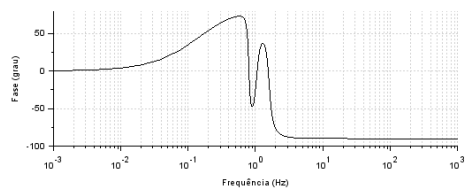
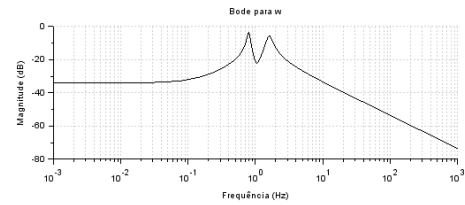
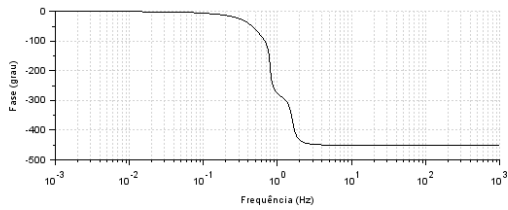
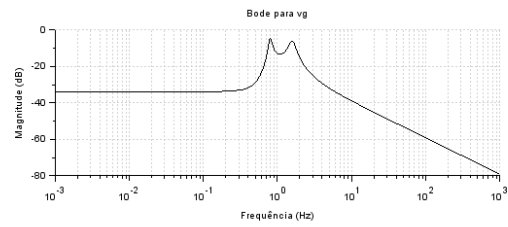
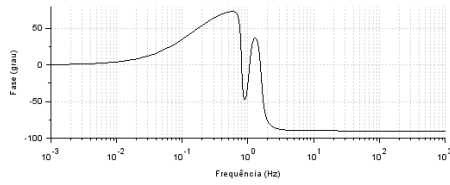
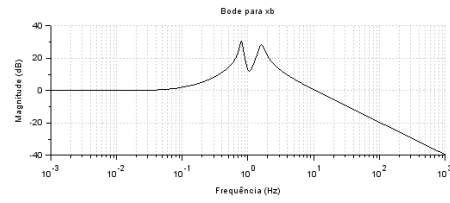
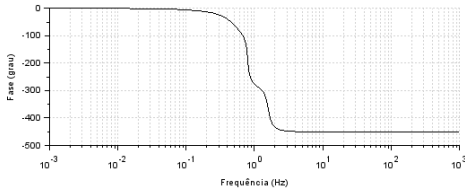
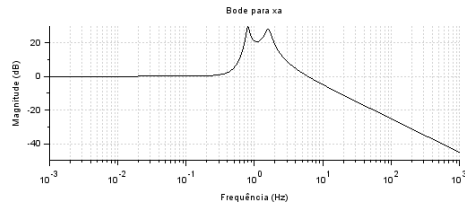
Simulação 2

Condições impostas:

$$v_D = \sin(9,8995t); v_C = \sin(9,8995t)$$

Resultados obtidos:





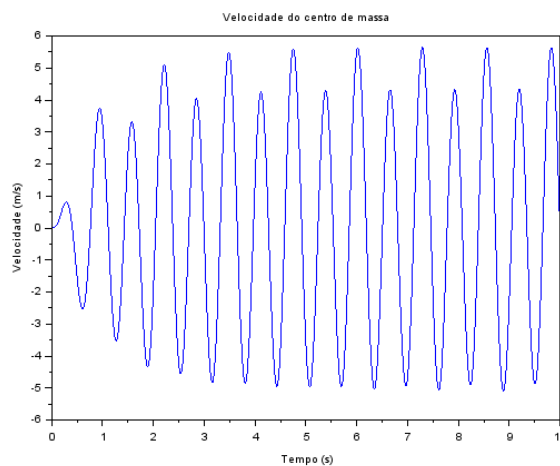
Nesse cenário pode-se perceber que sua velocidade angular cessa e a horizontal tender a manter o padrão da entrada colocada conforme o tempo. Nos gráficos de bode fica visível principalmente as velocidades de ressonância nos picos de magnitude.

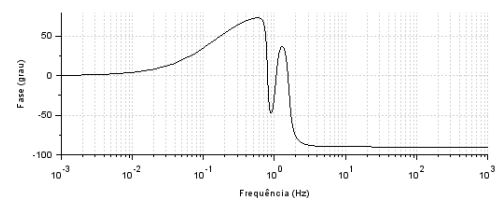
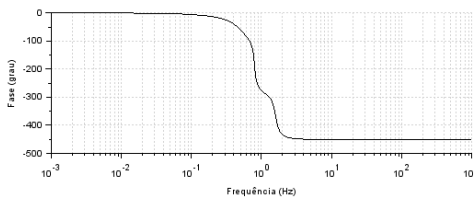
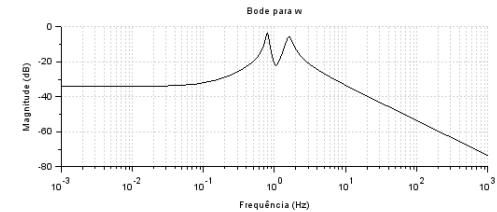
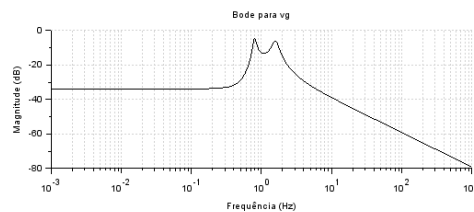
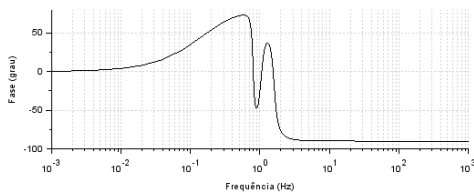
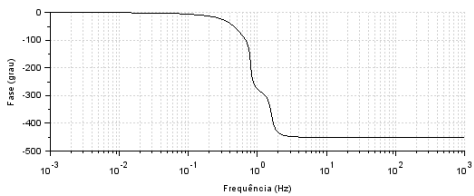
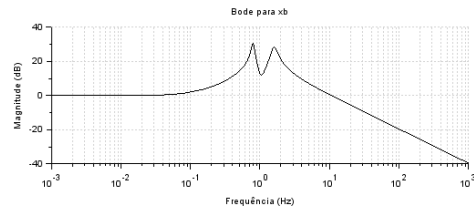
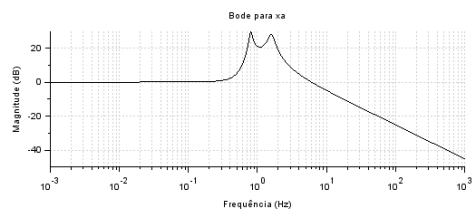
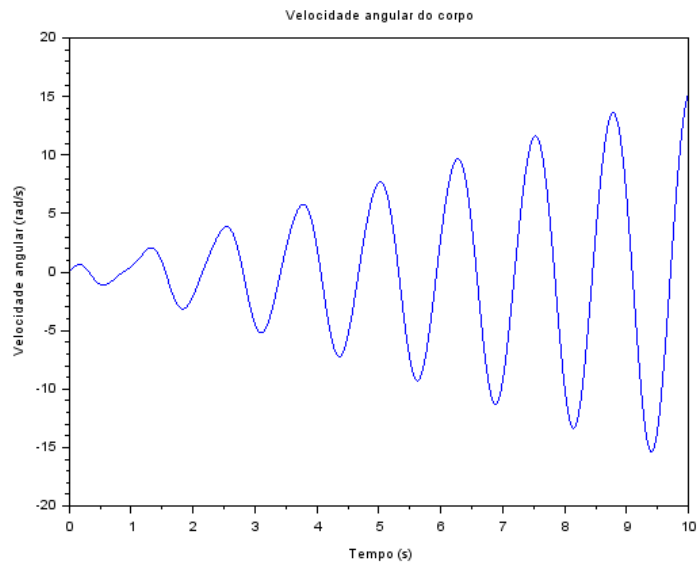
Simulação 3

Condições impostas:

$$v_D = \sin(9,8995t); v_C = \sin(9,8995t)$$

Resultados obtidos:





Nesse cenário pode-se perceber que o sistema é instável e sua velocidade angular tende a crescer e a horizontal tender a manter o padrão da entrada colocada conforme o tempo passa. Nos gráficos de bode fica visível principalmente as velocidades de ressonância nos picos de magnitude.

Código usado

```
// Lista G
clear();

//Parâmetros
M = 200; //Massa
lA = 0.8; //Comprimento A E B
lB = 0.8;
J = 512; //Momento de inércia
bA = 200; //Coeficiente de amortecimento A e B
bB = 200;
kA = 10000; //Constante elastica A e B
kB = 10000;
vH = 10; //Velocidade horizontal
td = (lA + lB)/vH; //Tempo de resposta de D [s]
//Condições iniciais:
t_i = 0;
t_f = 10;
t = linspace(t_i,t_f,1000);

xA0 = 0;
xB0 = 0;
vG0 = 0;
w0 = 0;
cond = 2;
//Entrada sistema
if cond == 1 then
function fun=u1(t), fun = t, endfunction
if t < td then
function fun=u2(t), fun = 0, endfunction
else
function fun=u2(t), fun = t, endfunction
end

function fun=u3(t), fun = 1, endfunction
if t < td then
function fun=u4(t), fun = 0, endfunction
else
function fun=u4(t), fun = 1, endfunction
end
elseif cond == 2 then
function fun=u1(t), fun = -cos(9.8995*t)/9.8995, endfunction
function fun=u2(t), fun = -cos(9.8995*t)/9.8995, endfunction
function fun=u3(t), fun = sin(9.8995*t), endfunction
function fun=u4(t), fun = sin(9.8995*t), endfunction
elseif cond == 3 then
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
function fun=u4(t), fun = -sin(4.9875*t), endfunction
end
//EE
```

```

funcprot(0)
function dy=estados(t, y)
dy(1) = y(3) - lA*y(4);
dy(2) = y(3) + lB*y(4);
dy(3) = -(kA/M)*y(1) - (kB/M)*y(2) - ((bA + bB)/M)*y(3) + ((bA*lA - bB*lB)/M)*y(4) +
(kA/M)*u1(t) + (kB/M)*u2(t) + (bA/M)*u3(t) + (bB/M)*u4(t);
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);
endfunction
resp = ode([xA0;xB0;vG0;w0],0,t,estados);
xA = resp(1,:);
xB = resp(2,:);
vG = resp(3,:);
w = resp(4,:);
scf(1)
xtitle("Velocidade do centro de massa");
xlabel("Tempo (s)");
ylabel("Velocidade (m/s)");
plot(t,vG);
scf(2)
xtitle("Velocidade angular do corpo");
xlabel("Tempo (s)");
ylabel("Velocidade angular (rad/s)");
plot(t,w);
//Análise das funções de transferência:
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];
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]);
h = ss2tf(sl);
scf(3);
bode(h(1,1));
xtitle("Bode para xa");
scf(4);
bode(h(1,2));
xtitle("Bode para xb");
scf(5);
bode(h(1,3));
xtitle("Bode para vg");
scf(6);
bode(h(1,4));
xtitle("Bode para w");

```