Escola Politécnica da Universidade de São Paulo

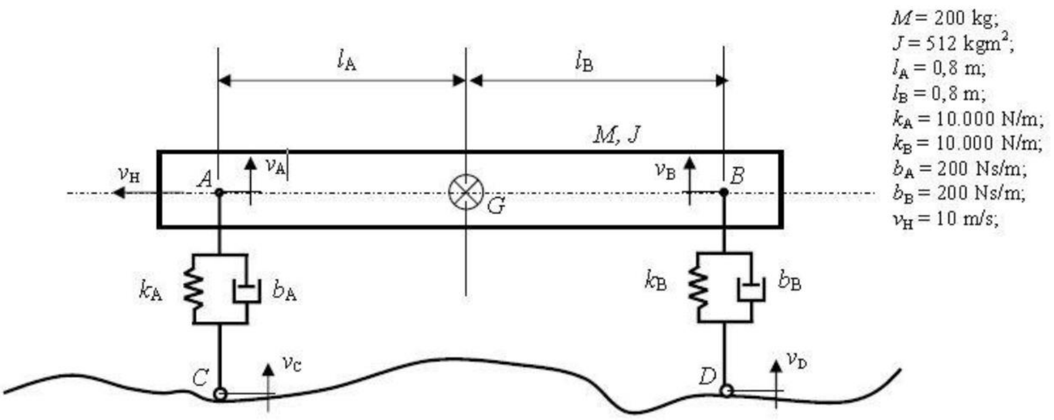
PME3380 – Modelagem de Sistemas Dinâmicos

Lista G

Henrique Aquino - 10772543

São Paulo, 2020

Obtenção do Modelo



Hipóteses adotadas:

i. AC e BD sempre verticais
ii. Movimento apenas no sentido xy
iii. Molas e amortecedores lineares (simplificação)
iv. Pequenos deslocamentos

Modelo Matemático:

Primeiro aplicamos o teorema do momento da quantidade de movimento e o teorema do movimento do baricentro:

$$M\ddot{x_G} = (k_A \cdot x_A - k_B \cdot x_B) - (b_A \cdot [\dot{x_G} - \dot{x_C} - \theta \cdot l_A] + b_B \cdot [\dot{x_G} - \dot{x_D} - \theta \cdot l_B])$$

$$J\ddot{\theta} = (k_A \cdot x_A \cdot l_A - k_B \cdot x_B \cdot l_B) + (b_A \cdot l_A [\dot{x_G} - \dot{x_C} - \theta \cdot l_A] - b_B \cdot l_B [\dot{x_G} - \dot{x_D} - \theta \cdot l_B])$$

Representando na forma de espaço de estados:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

$$x = \begin{bmatrix} x_A \\ x_B \\ v_G \\ \omega \end{bmatrix} \qquad u = \begin{bmatrix} v_C \\ v_D \end{bmatrix} \qquad y = \begin{bmatrix} v_G \\ \omega \end{bmatrix}$$

Obtemos as seguintes matrizes:

$$A = \begin{bmatrix} 0 & 0 & 1 & -l_A \\ 0 & 0 & 1 & l_B \\ {k_A}/{M} & {-k_B}/{M} & {-(b_A + b_B)}/{M} & {(b_A l_A + b_B l_B)}/{M} \\ {k_A l_A}/{J} & {-k_B l_B}/{J} & {-(b_A l_A + b_B l_B)}/{J} & {-\left(b_A {l_A}^2 + b_B {l_B}^2\right)}/{J} \end{bmatrix}$$

$$B = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ {b_A}/{M} & {b_B}/{M} \\ {-b_A l_A}/{J} & {b_B l_B}/{J} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

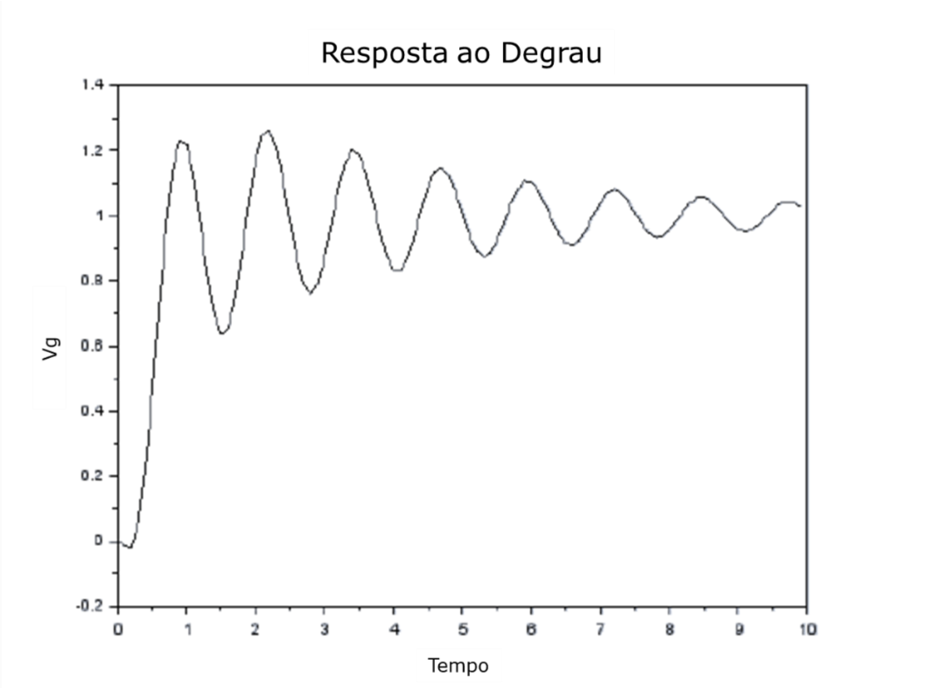
Simulações:

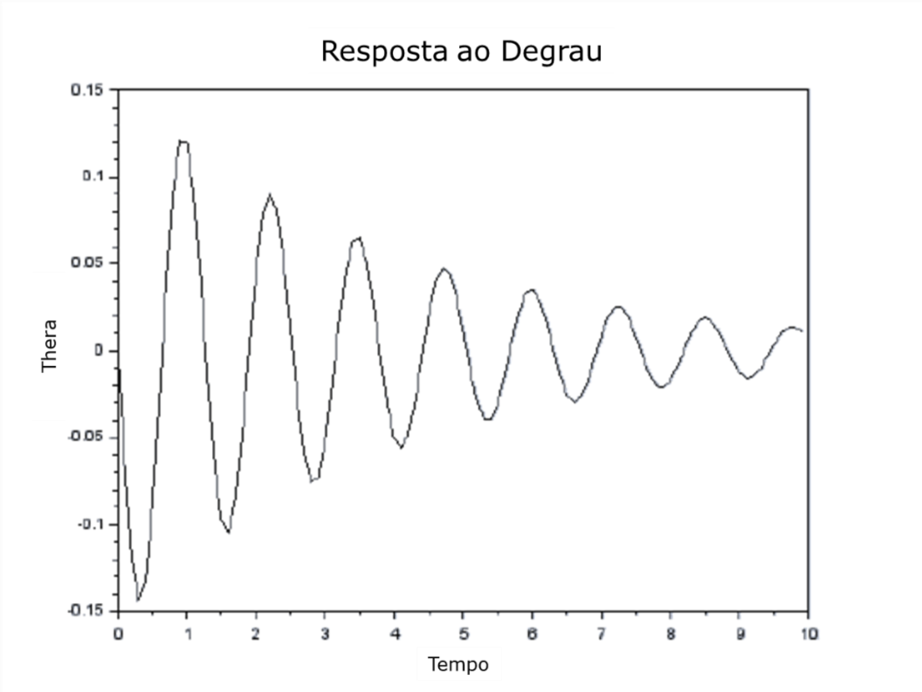
Agora simularemos primeiramente uma entrada em degrau, com as condições iniciais definidas abaixo:

$$u = \begin{bmatrix} v_C \\ v_D \end{bmatrix}$$

$$t_D = \frac{(l_A - l_B)}{v_H} = 0{,}16s$$

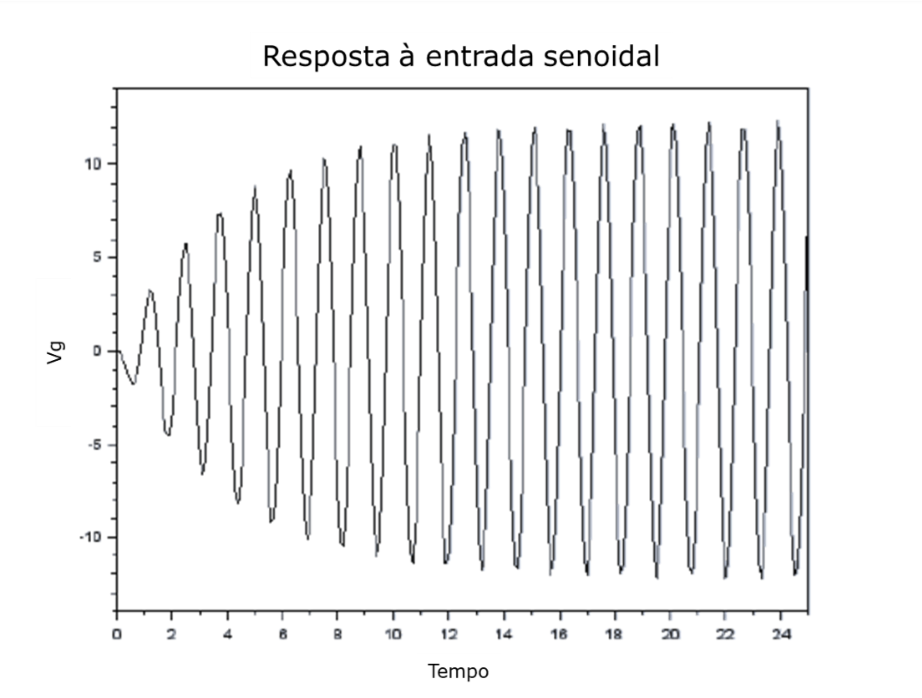
Simulação em degrau:





Simulação de entrada senoidal:

$$v_C = v_D = sen\,(9{,}88\ \cdot t)$$



Realizando a simulação para o caso em que $v_C = v_D = -sen\ (9{,}88\ \cdot t)$, seria obtido um perfil inverso. No caso de Theta, vemos que os valores de fato são nulos, pois oscilam muito próximos da origem (10^{-15}) e podemos atribuir a erros de simulação no Scilab.

Nessa etapa, para encontrarmos as frequências de ressonância, devemos encontrar a matriz de massa, de amortecimento e rigidez:

$$M = \begin{bmatrix} 200 & 0 \\ 0 & 512 \end{bmatrix} \qquad K = \begin{bmatrix} 20000 & 0 \\ 0 & 12800 \end{bmatrix} \qquad B = \begin{bmatrix} 400 & 0 \\ 0 & 256 \end{bmatrix}$$

Calculamos:

$$\zeta_1 = \frac{c_1}{2\ \cdot\ w_1 \cdot m_1} = 0{,}1$$

$$\zeta_2 = \frac{c_2}{2\ \cdot\ w_2 \cdot m_2} = 0{,}05$$

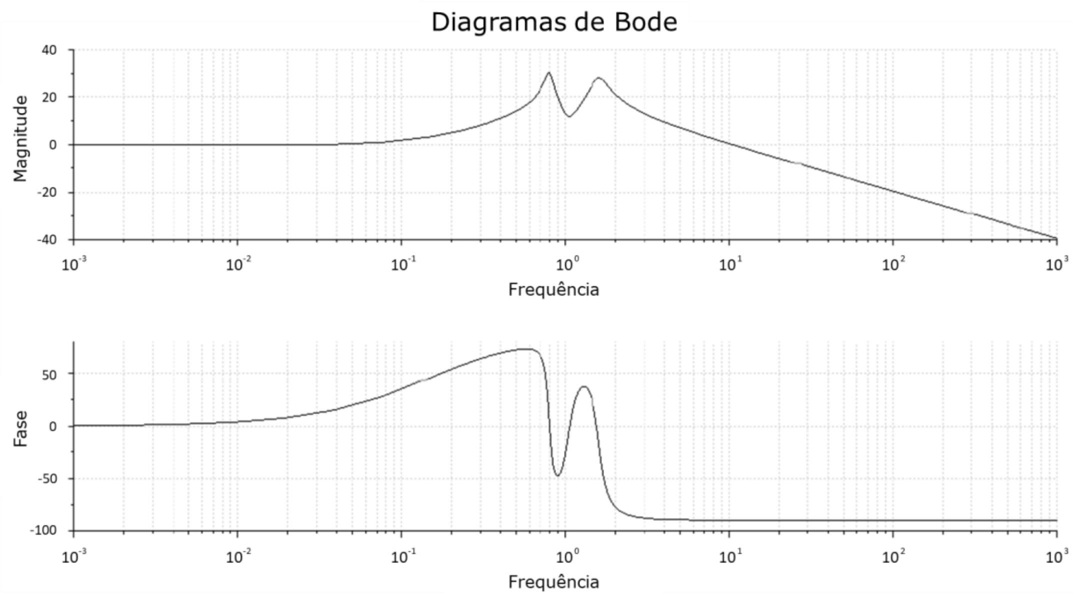
$$w_1 = \sqrt{\frac{k_1}{m_1}} = 10\ rad/s$$

$$w_2 = \sqrt{\frac{k_2}{m_2}} = 5\ rad/s$$

Com esses valores, tem-se:

$$w_{d1} = w_1 \sqrt{1 - {\zeta_1}^2} = 9{,}95\ rad/s$$

$$w_{d2} = w_2 \sqrt{1 - {\zeta_2}^2} = 4{,}99\ rad/s$$



```
// 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
```