UNIVERSIDADE DE SÃO PAULO

PME 3380 - Modelagem de Sistemas Dinâmicos

Lista G - Suspensão veicular – Modelo de $\frac{1}{4}$ de carro Não-Linear

Vitor Gregio Lourencini (8956387)

São Paulo
Novembro de 2020

1 Modelagem do Sistema

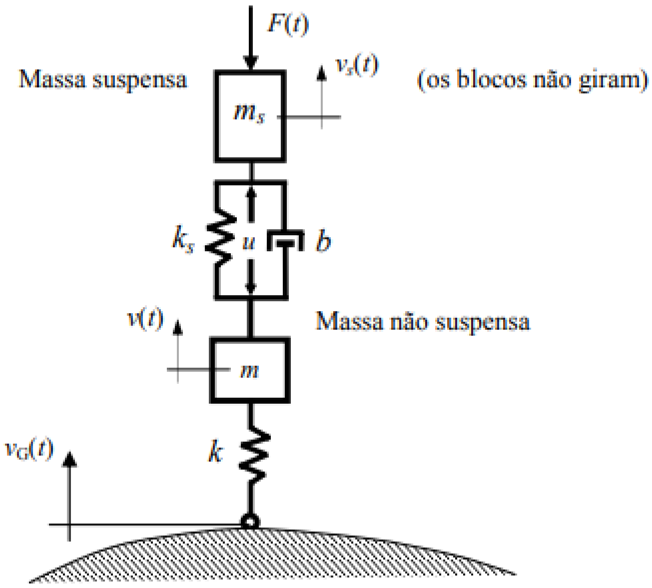

Figura 1: Modelo de Suspensão de $\frac{1}{4}$ de carro

O modelo a ser analisado possui 2 graus de liberdade (x e x_s). para cada bloco, aplica-se a 2ª Lei de Newton para se obter a dinâmica deles, porém existem as condições de não linearidade para cada bloco, 2 para a relação entre as massas (a mola foi comprimida ao máximo, e a mola não foi comprimida ao máximo) e 3 para a massa não suspensa (a mola está relaxada, a mola foi comprimida ao máximo ou a mola está comprimida, porém não ao máximo), e devemos considerar a sobreposição delas, sendo assim, temos 6 casos possíveis. São denotados os seguinte parâmetros:

- x_s: Posição da massa suspensa
- x: Posição da massa não suspensa
- x_G: Posição de contato com o solo
- k_s: Constante de mola entre as massas
- k_{sb}: Constante de mola comprimida entre as massas
- k: Constante de mola da massa não suspensa
- k_b: Constante de mola comprimida da massa não-suspensa
- v_s: Velocidade da massa suspensa
- v: Velocidade da massa não suspensa

- v_G: Velocidade de contato com o solo
- l_{sc}: Comprimento da mola entre as massas comprimida
- l_c: Comprimento da mola da massa não suspensa comprimida
- l: Comprimento da mola da massa não suspensa relaxada

1.1 Mola entre as massas não comprimida ao máximo

Condição: $l_{sc} < x_s - x$

1.1.1 Não há contato com o solo

Condição: $x - x_G > l$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_s(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_s(x_s - x - l_s) + b(v_s - v) \end{cases} \tag{1}$$

1.1.2 Existe contato com o solo, mas ainda não se atingiu o batente:

Condição: $l_C < x - x_G < l$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_s(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_s(x_s - x - l_s) + b(v_s - v) - k(x - x_G - l) \end{cases} \tag{2}$$

1.1.3 Existe contato com o solo e o batente foi atingido:

Condição: $x - x_G < l_C$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_s(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_s(x_s - x - l_s) + b(v_s - v) - k_b(x - x_G - l) \end{cases} \tag{3}$$

1.2 Mola entre as massas comprimida ao máximo

Condição: $x_s - x < l_{sc}$

1.2.1 Não há contato com o solo

Condição: $x - x_G > l$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_{sb}(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_{sb}(x_s - x - l_s) + b(v_s - v) \end{cases} \tag{4}$$

1.2.2 Existe contato com o solo, mas ainda não se atingiu o batente:

Condição: $l_C < x - x_G < l$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_{sb}(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_{sb}(x_s - x - l_s) + b(v_s - v) - k(x - x_G - l) \end{cases} \tag{5}$$

1.2.3 Existe contato com o solo e o batente foi atingido:

Condição: $x - x_G < l_C$

$$\begin{cases} m\dot{v}_s = -mg + u - F - k_{sb}(x_s - x - l_s) - b(v_s - v) \\ m\dot{v} = -mg - u + k_{sb}(x_s - x - l_s) + b(v_s - v) - k_b(x - x_G - l) \end{cases} \tag{6}$$

Sendo assim, podemos representar nosso sistema como um vetor de estados

$$x = \begin{bmatrix} x \\ x_s \\ v \\ v_s \\ x_G \end{bmatrix} \tag{7}$$

2 Entradas

Como visto na figura 1, existem 3 entradas para o sistema, distúrbio externo F, saturação da entrada u e a velocidade de contato com o solo v_G. Podemos escrever então nosso vetor de entradas como

$$[h]u = \begin{bmatrix} F \\ u \\ v_G \end{bmatrix} \tag{8}$$

2.1 Distúrbio Externo $F(t)$

Para o distúrbio externo foi randomizado 5 pontos no intervalo $t \in [1, 3]$ com valores de 0 a 1000 N, e a entrada foi considerada a interpolação por *splines* destes pontos. A entrada é nula antes de $t = 1s$.

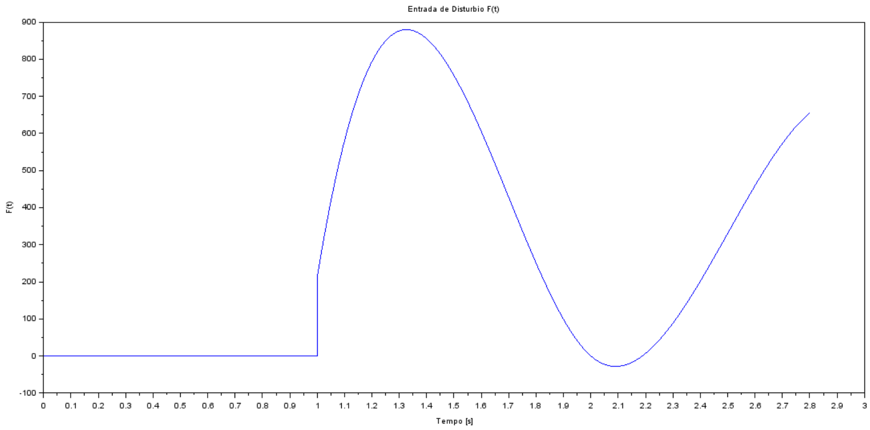

Figura 2: Entrada F(t)

Equação:

$$\begin{cases} F(t) = 0 & \Rightarrow t < 1 \\ F(t) = spline(rand) & \Rightarrow t \geq 1 \end{cases} \tag{9}$$

2.2 Saturação de Entrada $u(t)$

Para saturação de entrada será utilizada uma entrada senoidal com frequência e amplitude unitárias. Sendo nula antes de $t = 1s$ e com defeasagem para começar neste mesmo instante.

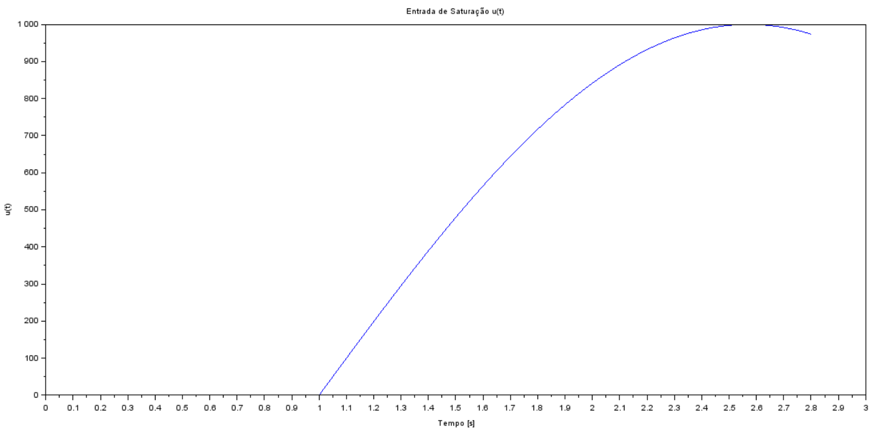

Figura 3: Entrada u(t)

Equação:

$$\begin{cases} u(t) = 0 & \Rightarrow t < 1 \\ u(t) = sin(t-1) & \Rightarrow t \geq 1 \end{cases} \tag{10}$$

2.3 Velocidade de Contato v_G

Para a velocidade de contato será utilizada a mesma entrada de exemplo da lista G, composta por uma lombada com formato de senóide

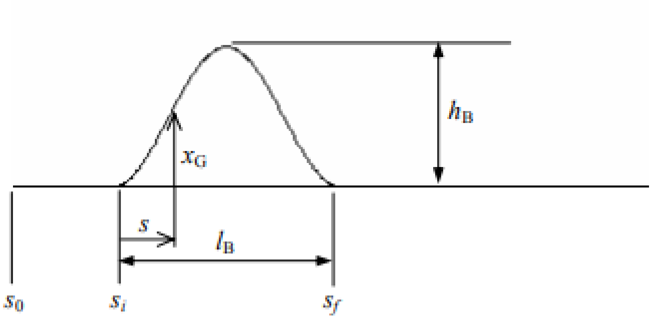

Figura 4: Entrada v_G

Equação:

$$\begin{cases} v_G(t) = 0 & \Rightarrow t < t_i \\ v_G(t) = \frac{h_B}{2}\frac{2\pi}{l_B} v_C sin(v_C \frac{2\pi}{l_B}(t - t_i)) & \Rightarrow t_i \leq t < t_f \\ v_G(t) = 0 & \Rightarrow t_f \leq t \end{cases} \tag{11}$$

3 Simulação

A implementação do modelo foi feita no *software Scilab* com a utilização do comando "ode"para a solução de equações diferenciais não-lineares. O código é apresentado no Apêndice I e abaixo temos as respostas da deflexão de ambas as molas, assim como a projeção do solo de contato, e a trajetória de ambas as massas.

No modelo foram consideradas os seguinte parâmetros:

- $m = 250kg$
- $b = 2000Ns/m$
- $g = 9.8m/s^2$
- $k_s = 15000N/m$
- $k_{sB} = 200000N/m$
- $k = 50000N/m$
- $k_B = 300000N/m$
- $l_s = 0.6m$
- $l_{sc} = 0.2m$
- $l = 0.4m$
- $l_c = 0.1m$
- $h_B = 0.2m$
- $l_B = 2m$
- $t_i = 0.1s$
- $v_{ch} = 35km/h$
- $v_c = vch/3.6m/s$
- $x_0 = [l - \frac{2mg}{k}; l - \frac{2mg}{k} + l_s - \frac{mg}{k_s}; 0; 0; 0]$ - Condições Iniciais

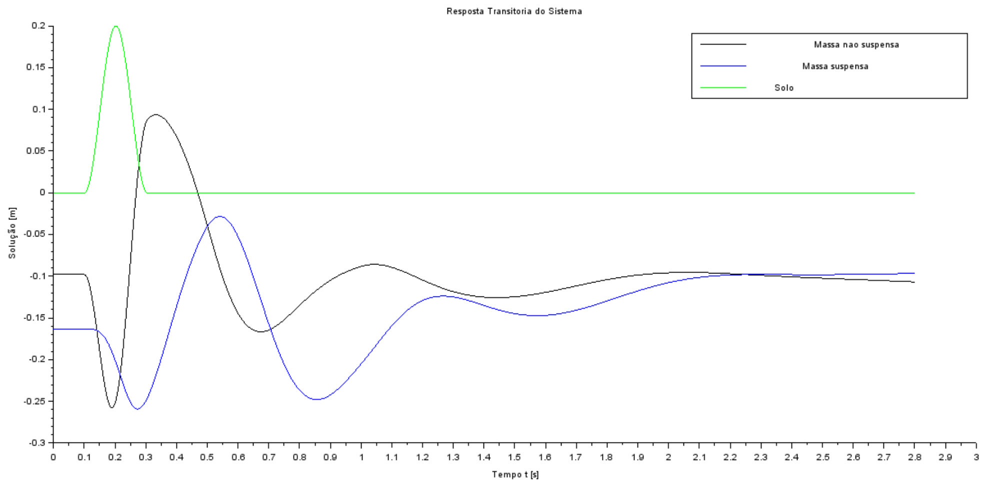

Figura 5: Deflexão das Molas

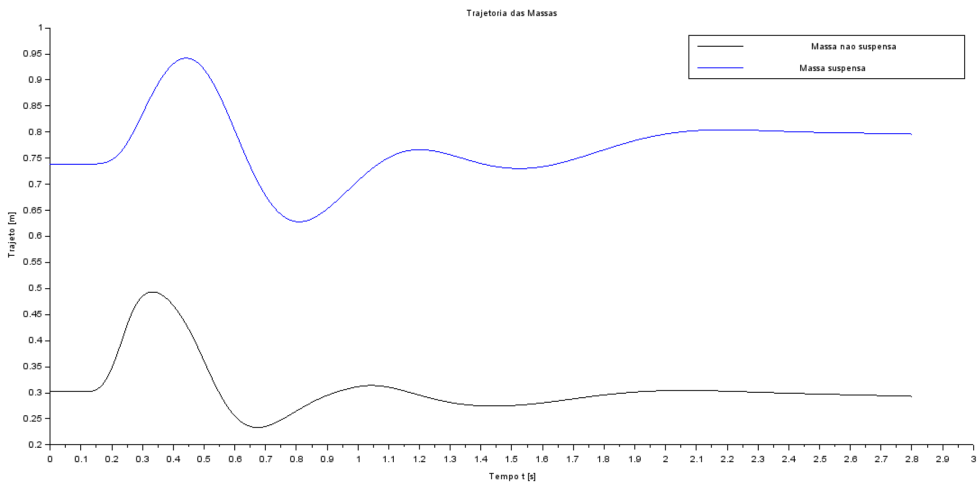

Figura 6: Trajetória das Massas

4 Apêndice I

```
// Carregar a funcao que implementa o modelo matematico do sistema
function [xdot]=sistema(t,x,entrada)
    [F,u,vG] = entrada(t);
    if lsc < x(2) - x(1) then
        if l < x(1) - x(5) then
            xdot = [x(3);x(4);-g + (- u + ks*(x(2) - x(1) - ls) ...
                + b*(x(4) - x(3))) / m;-g + (u - F - ks*(x(2) - ...
                x(1) - ls) - b*(x(4) - x(3))) / m;vG];
        elseif x(1) - x(5) < lc then
            xdot = [x(3);x(4);-g + (- u + ks*(x(2) - x(1) - ls) ...
                + b*(x(4) - x(3)) - kB*(x(1) - x(5) - l)) / m;-g ...
                + (u - F - ks*(x(2) - x(1) - ls) - b*(x(4) - ...
                x(3))) / m;vG];
        else
            xdot = [x(3);x(4);-g + (- u - F + ks*(x(2) - x(1) - ...
                ls) + b*(x(4) - x(3)) - k*(x(1) - x(5) - l)) / ...
                m;-g + (u - ks*(x(2) - x(1) - ls) - b*(x(4) - ...
                x(3))) / m;vG];
        end
    else
        if l < x(1) - x(5) then
            xdot = [x(3);x(4);-g + (- u + ksB*(x(2) - x(1) - ls) ...
                + b*(x(4) - x(3))) / m;-g + (u - F - ksB*(x(2) - ...
                x(1) - ls) - b*(x(4) - x(3))) / m;vG];
        elseif x(1) - x(5) < lc then
            xdot = [x(3);x(4);-g + (- u + ksB*(x(2) - x(1) - ls) ...
                + b*(x(4) - x(3)) - kB*(x(1) - x(5) - l)) / m;-g ...
                + (u - F - ksB*(x(2) - x(1) - ls) - b*(x(4) - ...
                x(3))) / m;vG];
        else
            xdot = [x(3);x(4);-g + (- u + ksB*(x(2) - x(1) - ls) ...
                + b*(x(4) - x(3)) - k*(x(1) - x(5) - l)) / m;-g + ...
                (u - F - ksB*(x(2) - x(1) - ls) - b*(x(4) - ...
                x(3))) / m;vG];
        end
    end
return
endfunction

// Carregar a funcao que implementa a entrada
function[F,u,vG]=entrada(t)
    if t < 1 then
        F = 0
        u = 0;
    else
        F = interp(t,1:0.5:3,a,d);
        u = 1000*sin(t - 1);
    end
```

```
    if t < ti then
        vG = 0;
        elseif t < (ti + lB / vc) then
            vG = (hB * 2*pi * vc / (2*lB)) * sin((vc * 2*pi / ...
                lB) * (t - ti));
        else
            vG = 0;
    end
return
endfunction

// Definir os valores dos parametros
m = 250; // massa [kg]
b = 2000; // constante de amortecimento [Ns/m]
g = 9.8; // aceleracao da gravidade [m/s2]
ks = 15000; // rigidez da mola entre as massas [N/m]
ksB = 200000; // rigidez do batente da mola entre as massas [N/m]
k = 50000; // rigidez da mola da massa nao suspensa [N/m]
kB = 300000; // rigidez do batente da massa nao suspensa [N/m]
ls = 0.6; // comprimento natural da mola entre as massas [m]
lsc = 0.2; // comprimento da mola entre as massas totalmente ...
    comprimida [m]
l = 0.4; // comprimento natural da mola da massa nao suspensa [m]
lc = 0.1; // comprimento da mola da massa nao suspensa ...
    totalmente comprimida [m]
hB = 0.2; // altura da lombada [m]
lB = 2; // comprimento da lombada [m]
ti = 0.1; // tempo percorrido ate atingir a lombada [s]
vch = 35; // velocidade do carro [km/h]
vc = vch/3.6; // velocidade do carro [m/s]
x0 = [l - 2*m*g/k;l - 2*m*g/k + ls - m*g/ks;0;0;0]; // condicoes ...
    iniciais
a = 1000*rand(1,5); // valores randomicos para a entrada F
d = splin(1:0.5:3,a) // spline de interpolacao para a entrada F

// O valor l-m*g/kM reflete a posicao de equilibrio da suspensao
// quando apenas o peso esta atuando.
t0 = 0; // instante inicial
t = t0:0.0001:2.8; // vetor de tempo
x = ode(x0,t0,t,list(sistema,entrada));

// Plotando a diferenca entre a coordenada da massa suspensa e a ...
    massa nao
// suspensa menos o comprimento natural da mola (deflexao):
figure
plot2d(t,x(1,:) - x(5,:) - l,1)
plot2d(t,x(2,:) - x(1,:) - ls,2)
plot2d(t,x(5,:),3)
T = list("Resposta Transitoria do Sistema","Tempo t ...
    [s]","Solucao [m]","Massa nao suspensa","Massa suspensa","Solo");
legends([T(4),T(5),T(6)],[1,2,3],1);
```

```
xtitle(T(1),T(2),T(3));

// Plotando a trajetoria das massas suspensa e nao suspensa
figure
plot2d(t,x(1,:),1)
plot2d(t,x(2,:),2)
T = list("Trajetoria das Massas","Tempo t [s]","Trajeto ...
    [m]","Massa nao suspensa","Massa suspensa");
legends([T(4),T(5)],[1,2],1);
xtitle(T(1),T(2),T(3));
```