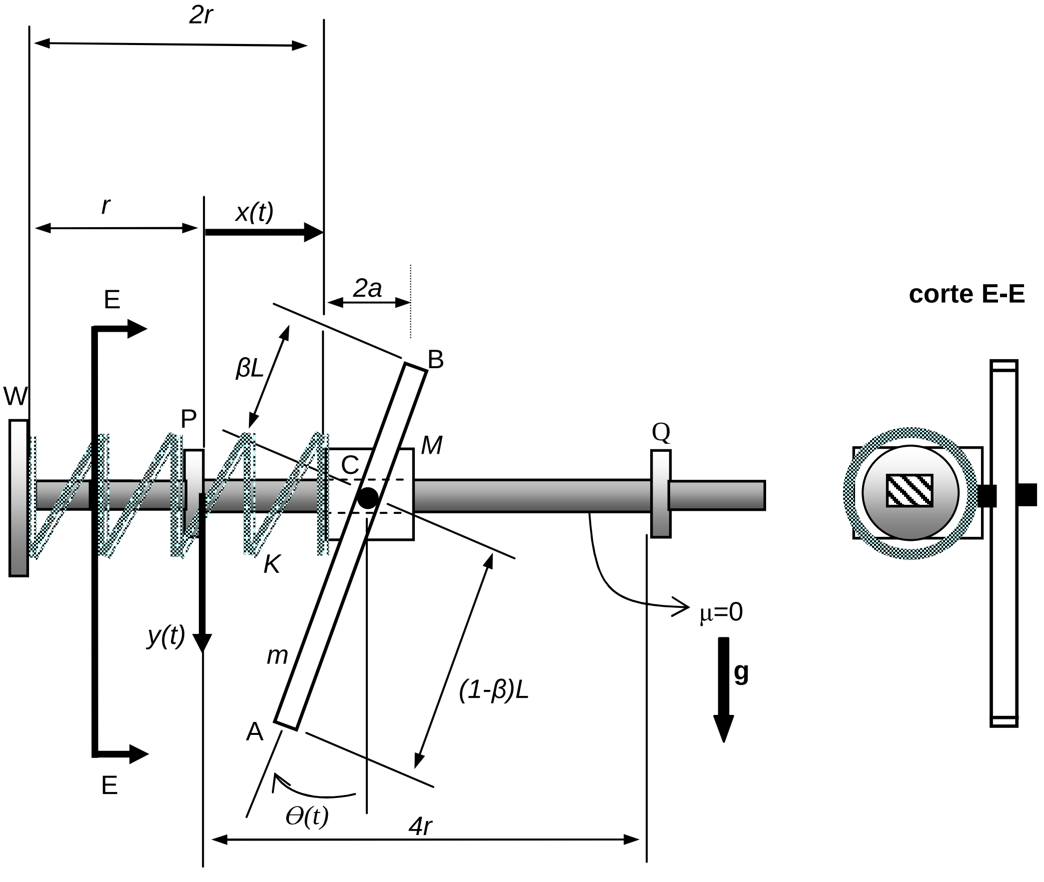
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Departamento de Engenharia Mecânica

PME 2200 – Mecânica B – EMSC#2 – Maio de 2011

Considere o sistema mostrado na figura abaixo, em que uma barra homogênea e delgada *AB* de massa *m* e comprimento *L* é vinculada a um bloco de massa *M* e comprimento *2a*, através de uma articulação ideal em *C*. Este conjunto pode deslizar sem atrito sobre a guia horizontal entre os batentes P e Q, fixos nesta guia e separados por uma distância *4r*. Uma mola ideal que possui comprimento livre 2*r* e constante elástica *K* é acoplada ao bloco e à base fixa W. O coeficiente de restituição entre o bloco de massa *M* e os batentes *P* e *Q* vale *e*. A posição do bloco de massa *M* é dada por *x(t)*. O parâmetro *β* pode ser ajustado para modificar a proporção entre os segmentos BC e AC da barra.



Etapa I: Modelagem

1) Considerando conhecidos *M*, *m*, *L*, *r* e *a*, demonstre que as equações do movimento para o sistema formado pelo bloco de massa *M* e pela barra *AB*, enquanto não ocorre choque em qualquer dos batentes, são dadas por:

$$\begin{aligned} &(m+M)\ddot{x} - m(1-2\beta)\frac{L}{2}\cos(\theta)\ddot{\theta} + K(x-r) + m(1-2\beta)\frac{L}{2}\operatorname{sen}(\theta)\dot{\theta}^2 = 0 \\ &-m(1-2\beta)\frac{L}{2}\cos(\theta)\ddot{x} + \left(J_{Gz} + m\left((1-2\beta)\frac{L}{2}\right)^2\right)\ddot{\theta} + mg(1-2\beta)\frac{L}{2}\operatorname{sen}(\theta) = 0 \end{aligned} \quad . \qquad (1)$$

Além disso, é possível escrevê-las na forma matricial geral $\Lambda\ddot{\vec{q}} + \Gamma\vec{q} + \vec{f}(\vec{q},\dot{\vec{q}}) = \vec{0}$ ou, de maneira explícita para o caso em questão,

$$\begin{bmatrix} (m+M) & -m(1-2\beta)\frac{L}{2}\cos(\theta) \\ -m(1-2\beta)\frac{L}{2}\cos(\theta) & \left(J_{Gz} + m\left((1-2\beta)\frac{L}{2}\right)^2\right) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x-r \\ \theta \end{bmatrix} + \begin{bmatrix} f_1(\theta,\dot{\theta}) \\ f_2(\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad (2)$$

Note que as equações são funções do par de parâmetros (*K*; *β*) e que, dependendo das condições iniciais, $x(0) = x_0,\ \dot{x}(0) = v_0,\ \theta(0) = \theta_0,\ \dot{\theta}(0) = \omega_0$, e dos valores destes parâmetros, existe possibilidade de ocorrência de choque contra um ou mesmo contra os dois batentes.

2) Utilizando o TRI e o TMI, obtenha as equações para a condição de choque; tais equações serão também utilizadas na simulação do comportamento do sistema.

Etapa II: Simulação e análise

Implemente as equações de movimento em linha de comando no Scilab. Para isso, reproduza os códigos apresentados ao final deste documento, que descrevem o procedimento de implementação computacional com base nas equações de movimento reescritas na forma abaixo:

$$\ddot{\vec{q}} = \Lambda^{-1}\left(-\Gamma\vec{q} - \vec{f}(\vec{q},\dot{\vec{q}})\right) \quad . \qquad (3)$$

Efetue simulações durante 12 segundos utilizando os parâmetros do enunciado (M=1,0kg; m=9,0kg; L=1,5m; r=50mm; a=25mm; g=10m/s^2) e outras condições segundo a Tabela 1.

Tabela 1: conjuntos de parâmetros para simulação

Conjunto	*K* [N/m]	*e*	*β*	$x(0)$ [m]	$\dot{x}(0)$ [m/s]	$\theta(0)$ [rad]	$\dot{\theta}(0)$ [rad/s]
1	100	1,0	0,25	2*r	0	$\pi/3$	0
2	100	1,0	0,25	2*r+a	0	$\pi/3$	0
3	100	1,0	0,25	2*r	0	$\pi/36$	0
4	100	1,0	0,25	2*r	0	$\pi/3$	0
5	100	1,0	0	2*r	0	$\pi/12$	0
6	100	0,5	0,50	2*r+a	0	$\pi/3$	0
7	100	0,5	0,25	2*r+a	0	$\pi/3$	0
8(1)	100	0,5	0	2*r+a	0	$\pi/12$	0
9(2)	0	1,0	0	2*r+a	0	$\pi/64$	0

(1): nesta simulação, observe o comportamento da evolução temporal da posição angular da barra e explique o porquê.

(2): nesta simulação, verifique também o movimento do baricentro do sistema e explique o seu comportamento.

Para todas as simulações, represente graficamente o comportamento de $x(t)$, $\dot{x}(t)$, $\theta(t)$, e $\dot{\theta}(t)$ em função do tempo e procure verificar se os resultados estão de acordo com o que se esperaria da realidade física do problema.

Os algoritmos que se seguem promovem a simulação do sistema acima modelado. Os arquivos “simula_choque.sce” e “eq_mov.sci” referem-se respectivamente ao “script” (programa principal invocado na linha de “prompt” do programa Scilab) e ao arquivo do tipo função que contém as equações de movimento.

Para invocar o programa principal deve-se, no “prompt” do Scilab, digitar

```
>> exec simula_choque.sce; ,
```

tendo inicialmente o cuidado de executar o Scilab (“prompt”) no mesmo diretório em que os arquivos .sci e .sce estiverem. Os parâmetros a serem variados para as diferentes simulações devem ser alterados nas respectivas linhas do programa principal. O vetor de valores iniciais para cada simulação é o y_0. Para a visualização dos resultados, devem ser elaborados gráficos contendo nas abscissas a variável “temp” (tempo, em segundos) e, nas ordenadas, uma das linhas da matriz "*estado*", que contém $x(t)$, $\dot{x}(t)$, $\theta(t)$, e $\dot{\theta}(t)$ respectivamente em suas linhas de 1 a 4.

Códigos em linguagem Scilab

Arquivo simula_choque.sce

```
// Codigo para simulacao do EMSC2 - 2011

clc
global temp estado

//Incremento de tempo na resposta [s]
Dt = 0.0005;

// massas do problema [kg]
M=1.0;
m=9.0;

// comprimento da barra [m]
L = 1.5;

// metade do comprimento natural da mola [m]
r = 0.050;

// metade do comprimento do bloco [m]
a=0.025;

// fator beta
betah=0.25;

//rigidez da mola [N/m]
k=100.0;

// coeficiente de restituicao
e = 0.5;

// aceleracao da gravidade [m/s^2]
grav = 10.0;

// parametros derivados Jc e d
// Jc eh momento de inercia da barra com polo em C
// d eh a distancia de C a G na barra

Jc= m*L*L*(1/3+betah*betah-betah);

d= L*(0.5 -betah);

tol=1.0e-10;

// carrega definicao da funcao f

// usar este comando nas versoes mais recentes (5.xxx)
//getd("eq_mov.sci");

// usar este comando nas versoes mais antigas (4.xxx)
getf("eq_mov.sci");
```

```
// estado Inicial
y0=[2.0*r+a;0.0;%pi/3;0.0];

// instante final de integracao [s]
t_final = 12.0;

// aloca memoria para vetor tempo
n_tempo=floor(t_final/Dt);
for i=1:n_tempo
  temp(i)=(i-1)*Dt;
end

estado(:,1)=y0;
for i=1:n_tempo-1
   time(1)=temp(i);
   time(2)=temp(i+1);
   [est,w,q]=ode("adams",y0,time(1),time,f);
   y0=est(:,2);
   if ((est(1,2)<tol)&(est(2,2)<-tol)) then
      y0(2)=-e*est(2,2);
      y0(4)=est(4,2)-m*d*est(2,2)*cos(est(3,2))*(1+e)/Jc;
   end
   if ((4*r-2*a-est(1,2)<tol)&(est(2,2)>tol)) then
      y0(2)=-e*est(2,2);
      y0(4)=est(4,2)-m*d*est(2,2)*cos(est(3,2))*(1+e)/Jc;
   end
   if ((est(1,2)<tol)&(est(2,2)==0.0)) then
      y0(2)=est(2,2);
      y0(4)=est(4,2);
   end
   if ((4*r-2*a-est(1,2)<tol)&(est(2,2)==0.0)) then
      y0(2)=est(2,2);
      y0(4)=est(4,2);
   end

   estado(:,i+1)=est(:,2);
end

// elabora um grafico do deslocamento x(t)
// poderia ser dx/dt(t) ou teta(t) ou d(teta)/dt(t)
scf(0);
xtitle('Posicao X(t) (m) vs. Tempo (s)');
plot2d(temp,estado(1,:));
```

Arquivo eq_mov.sci

```
function dx=f(t,x)
  detM= Jc*(M+m)-m*m*d*d*cos(x(3))*cos(x(3));
  // equacoes do movimento dx/dt= f(t,x)
  // onde x eh o vetor estado definido por
  // x = [ x; dx/dt; teta; d(teta)/dt];
  dz1= x(2);
  dz2= (Jc*(-m*x(4)*x(4)*d*sin(x(3))-k*(x(1)-r))+m*d*cos(x(3))*(-m*grav*d*sin(x(3))))/detM;
  dz3= x(4);
  dz4= (m*d*cos(x(3))*(-m*x(4)*x(4)*sin(x(3))*d-k*(x(1)-r))+(M+m)*(-m*grav*d*sin(x(3))))/detM;
  dx = [dz1; dz2; dz3; dz4];

endfunction
```