Método de cálculo da vibração utilizando rayleigh-ritz para viga com massa adicionada

Table of Contents

Código para solução de vibração de viga com massa adicionada utilizando o método de Rayleigh-Ritz.

Parametros

```
E = 97e9; %(Pa) modulo de elasticidade

h = 4.9e-3; %(m) altura da seção da viga
b = 38.8e-3;%(m) largura da seção da viga

I = b*(h^3)/12;%(m^4) segundo momento de area da viga

A = (4.9e-3)*(38.8e-3); %(m²) area da seção
rho = 7860.687092; %(kg/m³) densidade do material
L = 1; %(m) comprimento da viga

k = 1000; % discretização numérica

m = [0.0439 0.0457]; %(kg) massas pontuais colocadas na viga
x_m = [L L-0.25]; %(m) posição respectiva de cada massa na viga

if length(m) ~= length(x_m)
    error('A quantidade de pontos de massa é diferente da quantidade
 de posições, checar os vetores m e x_m'); % evitar inconsistencias do
 tipo m ter mais elementos que x_m e virce-versa
end
```

Função phi

```
phi =@(x) [Y_1(x) Y_2(x) Y_3(x) Y_4(x) Y_1(2*x)];
%phi = @(x) [cos(pi*x/(2*L)) cos(3*pi*x/(2*L)) cos(5*pi*x/(2*L))
 cos(7*pi*x/(2*L)) cos(9*pi*x/(2*L)) cos(11*pi*x/(2*L))]; %função
 proposta para resolver o problema
```

```
n = length(phi(1)); % quantidade de elementos na função, por tanto
 quantidades de modos e autovetores a serem obtidos

% Inicializa matrizes de massa e rigidez
K = zeros(n,n);
M = zeros(n,n);

syms xx; % variável simbólica para resolver equações
phi_xx = phi(xx); % vetor de funções com a variavel simbólica aplicada
```

Aloca os valores das matrizes de massa e rigidez

```
for i = 1:1:n
    for j =1:1:n
        massa = 0; % zera o valor para o somatório

        phi_i = phi_xx(i); % toma o valor de phi para o elemento i
        phi_j = phi_xx(j); % toma o valor de phi para o elemento i
        phi_i_deriv = eval(['@(x)' char(diff(phi_i))]); % faz a
 derivada simbólica e transforma para function handle
        phi_j_deriv = eval(['@(x)' char(diff(phi_j))]); % faz a
 derivada simbólica e transforma para function handle
        phi_i_deriv2 = eval(['@(x)' char(diff(diff(phi_i)))]); % faz a
 derivada simbólica e transforma para function handle
        phi_j_deriv2 = eval(['@(x)' char(diff(diff(phi_j)))]); % faz a
 derivada simbólica e transforma para function handle

        for p = 1:1:length(m)
            massa = massa +
 m(p)*eval(subs(phi_i,x_m(p)))*eval(subs(phi_j,x_m(p))); % calcula o
 elemento de massa pontual a ser aplicado na matriz M
        end

        k_ij_deriv =@(x) E*I*phi_i_deriv2(x)*phi_j_deriv2(x); % monta
 os elementos da matriz K para serem integrados
        K(i,j) = eval(int(k_ij_deriv(xx),0,L)); % faz a integral
 simbólica de 0 a L dos elementos e substitui os valores obtendo a
 matriz K


        M(i,j) = real(eval(int(rho*A*phi_i*phi_j,0,L)) +
 subs(massa)); % monta a matriz M

    end
end
```

Autovetores e autovalores

```
[autovet,lambda] = eig(M\K); % calcula os autovalores e autovetores de
 M^-1*K
```

```
Wn = sqrt(diag(lambda)); % obtêm os modos de vibração pelo método de
 Rayleigh Ritz

Wn_sort = sort(Wn);
autovet_org = zeros(size(autovet));

for q = 1:1:length(autovet)
    autovet_org(:,q) = autovet*(Wn_sort(q) == Wn);
end

Wn = Wn_sort;
autovet = autovet_org;


x = linspace(0,L,k)'; % cria o vetor "distancia da origem" para
 substituir nas equações e plotar gráficos
t = linspace(0,20,k)'; % cria o vetor "tempo" para substituir nas
 equações e plotar gráficos
```

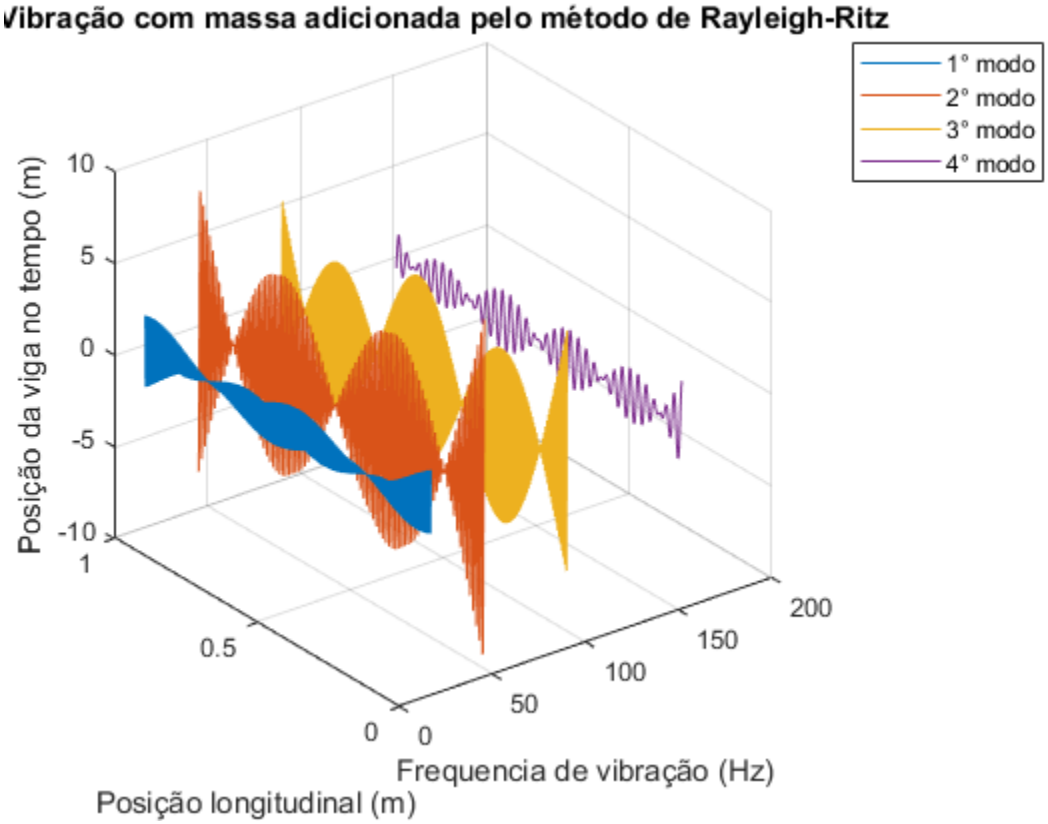
Funções deslocamento dos 4 primeiros modos

```
U1 = @(x) phi(x)*autovet(:,1); % cria cada uma das funções
U2 = @(x) phi(x)*autovet(:,2);
U3 = @(x) phi(x)*autovet(:,3);
U4 = @(x) phi(x)*autovet(:,4);

%caso de ajuste de 1 massa adicionada:
C1 = 2.3/4.5467; C2 = 9.5/14.5584; C3 = 7.3/21.8532; C4 = 2.4/15.5363;

%ccaso de ajuste de 2 massas adicionadas:
%C1 = 4.5/4.5467; C2 = 8/14.5584; C3 = 6/21.8532; C4 = 2.5/15.5363;

Y1 = @(x,t) C1*U1(x).*cos(Wn(1)*t); % obtem a fução do deslocamento no
 tempo do sistema
Y2 = @(x,t) C2*U2(x).*cos(Wn(2)*t);
Y3 = @(x,t) C3*U3(x).*cos(Wn(3)*t);
Y4 = @(x,t) C4*U4(x).*cos(Wn(4)*t);
```

Gráficos

```
plot3 (Wn(1)*ones(length(x),1)/(2*pi),x,Y1(x,t));
grid on
hold on

title('Vibração com massa adicionada pelo método de Rayleigh-Ritz');
xlabel('Frequencia de vibração (Hz)');
ylabel('Posição longitudinal (m)');
zlabel('Posição da viga no tempo (m)');

plot3 (Wn(2)*ones(length(x),1)/(2*pi),x,Y2(x,t));

hold on
```

```
plot3 (Wn(3)*ones(length(x),1)/(2*pi),x,Y3(x,t));

hold on

plot3 (Wn(4)*ones(length(x),1)/(2*pi),x,Y4(x,t));
hold off
legend('1° modo','2° modo', '3° modo', '4° modo');
```



Published with MATLAB® R2017a