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

Table of Contents

Parametros	1
Função phi	1
Aloca os valores das matrizes de massa e rigidez	2
Autovetores e autovalores	2
Funções deslocamento dos 4 primeiros modos	2
Gráficos	3

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

Parametros

```
E = 97e9; %(Pa) modulo de elasticidade
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 = [0];% 0.24]; %(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 inconsistências do
    tipo m ter mais elementos que x_m e vice-versa
end
```

Função phi

```
phi = @(x) [sin(x) cos(x) sinh(x) cosh(x) sin(2*x) cos(2*x) sinh(2*x)
    cosh(2*x)]; %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 variável 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 j
        phi_i_deriv = eval(['@(x)' char(diff(phi_xx(i)))]); % faz a
        derivada simbólica e transforma para function handle
        phi_j_deriv = eval(['@(x)' char(diff(phi_xx(j)))]); % faz a
        derivada simbólica e transforma para function handle

        k_ij_deriv = @(x) E*A*phi_i_deriv(x)*phi_j_deriv(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

        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

        M(i,j) = real(eval(int(rho*xx*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

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);

Y1 = @(x,t) U1(x).*cos(Wn(1)*t); % obtem a função do deslocamento no
    tempo do sistema
Y2 = @(x,t) U2(x).*cos(Wn(2)*t);
Y3 = @(x,t) U3(x).*cos(Wn(3)*t);
Y4 = @(x,t) 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));

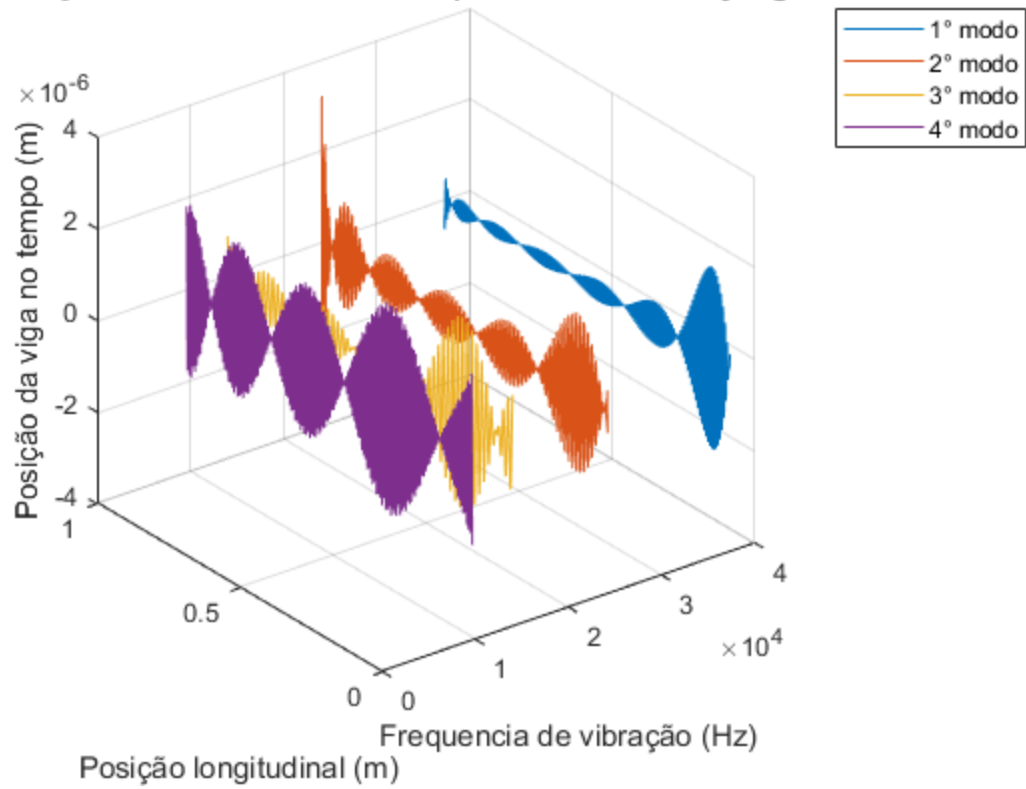
hold on

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('1º modo', '2º modo', '3º modo', '4º modo');
```

Vibração com massa adicionada pelo método de Rayleigh-Ritz



Published with MATLAB® R2017a