

PEC00025: Introduction to Vibration Theory

TRABALHO 2.2

Henrique Cardoso Koch, Matricula: 00312128

In [92]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg as sc

# Matriz de Flexibilidade H

H = np.array ([[1.886 , 3.019, 3.005, 2.157, 0.7886],
               [3.185, 7.888, 8.181, 5.972, 2.199],
               [3.109, 8.285, 10.85, 8.285, 3.109],
               [2.199, 5.972, 8.181, 7.888, 3.185],
               [0.7886, 2.157, 3.005, 3.019, 1.886]])*1e-8

K = np.linalg.inv(H)

# Corrigida a matriz de massa do trabalho para que a menor frequência seja abaixo de 2
Hz

M = np.array ([[30, 0, 0, 0, 0],
               [0, 30, 0, 0, 0],
               [0, 0, 30, 0, 0],
               [0, 0, 0, 30, 0],
               [0, 0, 0, 0, 30]])*1e3

# Equação de autovalores e autovetores e ordenamento das respostas

w21, Phi1 = sc.eig(K, M)

iw = w21.argsort()
w21 = w21[iw]
Phi1 = Phi1[:,iw]

# Cálculo das frequências naturais

wk1 = np.sqrt(np.real(w21))
fk1 = wk1/2/np.pi
```

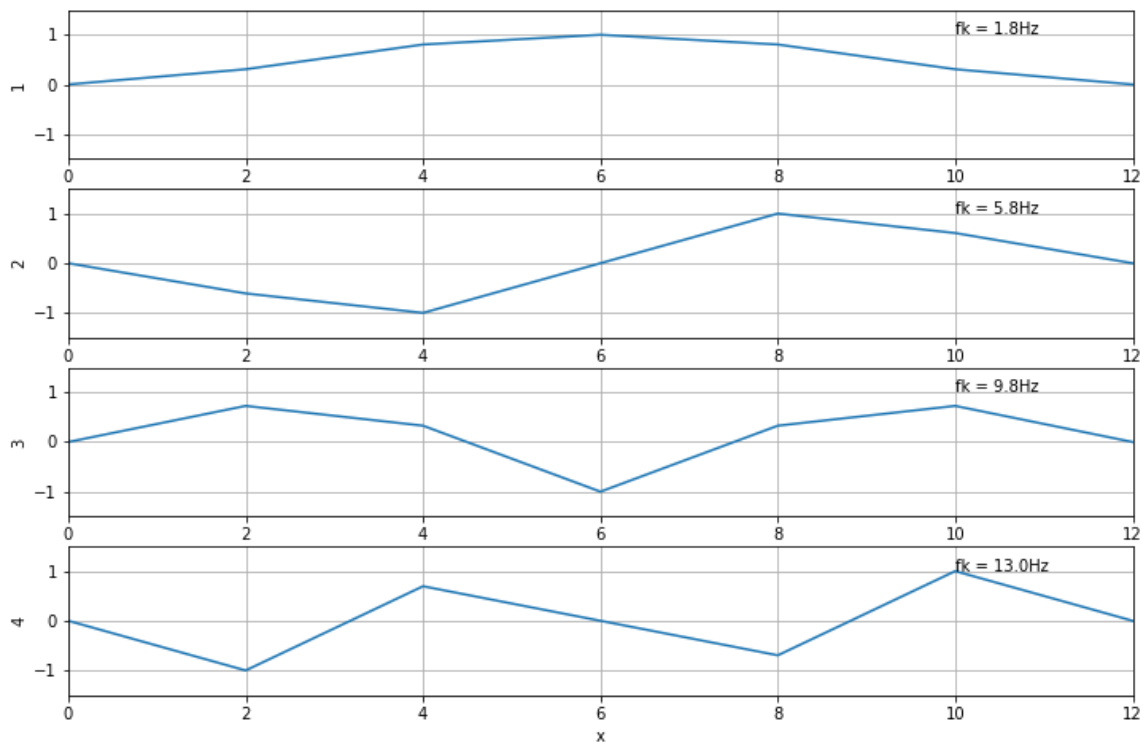
In [91]:

```
# Plotagem dos primeiros 4 modos de vibração da estrutura
```

```
plt.figure(1, figsize=(12,10))  
x = np.arange(0, 14, 2)
```

```
for k in range(4):  
    pk = np.zeros(7)  
    pk[1:-1] = Phi1[:,k]  
    pk /= np.max(np.abs(pk))  
  
    plt.subplot(5,1,k+1)  
    plt.plot(x, pk)  
  
    plt.xlim( 0.0, 12.0);  
    plt.ylim(-1.5, 1.5); plt.ylabel(str(k+1));  
    plt.text(10, 1, 'fk = {0:3.1f}Hz'.format(fk1[k]));  
    plt.grid(True)
```

```
plt.xlabel('x');
```



In [93]:

```
#Menor frequência natural = primeiro modo
freq = np.max(fk1)
print("Frequência utilizada: %f" %freq)

#Força escolhida

F = 30000

Tn = 1/freq
td = Tn/4

#massa ao longo do comprimento linear de cada nó

m = 15000

#Impulso
Vo = F*td/m

print("Velocidade inicial: %f" %Vo)

#VELOCIDADE INICIAL NO NÓ QUE APRESENTA MAIOR DESLOCAMENTO
Vo_v = [0,0,Vo,0,0]
uo_v = np.zeros(5)

Phit=Phi1
for k in range(12):
    Phit=Phi1.transpose()

angulo = np.pi/2
DESLMAX = np.zeros(5)

for k in range (5):
    M = np.dot(np.dot(Phit[k], M), Phi1[k])
    deslmax = (np.dot(np.dot(Phit[k], M), Vo_v))/(wk1[k]*np.sin(angulo)*M)
    DESLMAX[k]= deslmax

t = np.linspace (0,12,1000)

uk = np.zeros ((5,1000))

for k in range(5):
    uk[k,:]=DESLMAX[k]*np.cos(-wk1[k]*t - np.pi/2)

u = np.dot(Phi1,uk)

#Somatório de deslocamentos no nó 3 = Maior deslocamento
plt.figure(2, figsize=(10,4))
plt.plot(t,u[3,:])
plt.xlim(0.0, 6.0); plt.xlabel("t");
plt.ylim(-0.0013,0.0013); plt.ylabel ("u(t)")
plt.grid (True)

zeta=np.zeros(5)
zeta[0]=0.01

alpha0=zeta[0]*2*wk1[0]
alpha1=(zeta[0]*2)/wk1[0]
```

```
print("Alpha 0 =",alpha0)
print("Alpha 1 =",alpha1)
```

Frequência utilizada: 14.157579

Velocidade inicial: 0.035317

Alpha 0 = 0.22610135497141431

Alpha 1 = 0.0017691181021474705

