Sistemi vibranti ad 1 gdl

Testo:

Si consideri un sistema vibrante con un solo grado di libertà forzato l'equazione che descrive tale sistema è la seguente:

$$m\ddot{x} + c\dot{x} + kx = F_o sin(\omega t) \quad (1)$$

dove, con riferimento alla figura 1, m, k e c sono i parametri caratteristici del sistema e $F_0 \sin(\omega t)$ è la forzante armonica.

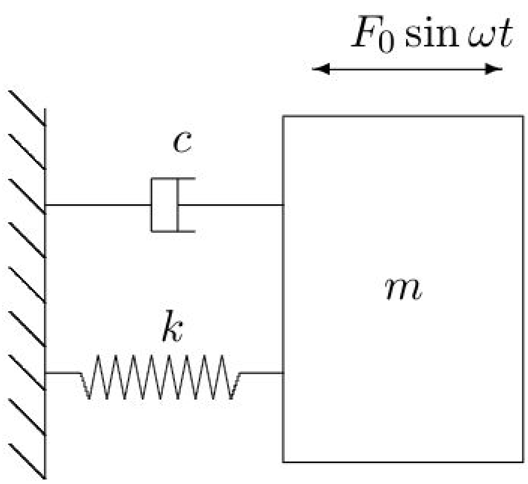

Figura 1: Sistema vibrante ad 1 gdl con forzante armonica generica

utilizzando la notazione vettoriale:

$$F = F_0 e^{i\omega t}; \, x = \mathbf{X} e^{i\omega t}; \, \mathbf{X} = e^{i\delta t} \quad (2)$$

derivando si ottiene:

$$\dot{x} = i\omega \mathbf{X} e^{i\omega t} \qquad \ddot{x} = -\omega^2 \mathbf{X} e^{i\omega t} \quad (3)$$

e sostituendo nell'equazione completa:

$$-m\omega^2 \mathbf{X} e^{i\omega t} + ci\omega \mathbf{X} e^{i\omega t} + k\mathbf{X} e^{i\omega t} = F_0 e^{i\omega t} \quad (4)$$

si ottiene infine:

$$\mathbf{X} = \frac{\frac{F_0}{k}}{1 - (\frac{\omega}{\omega_n})^2 + i2\varsigma \frac{\omega}{\omega_n}} \quad (5)$$

La figura sottostante mostra l'andamento dell'ampiezza e della fase in funzione della pulsazione della forzante e dello smorzamento adimensionale.

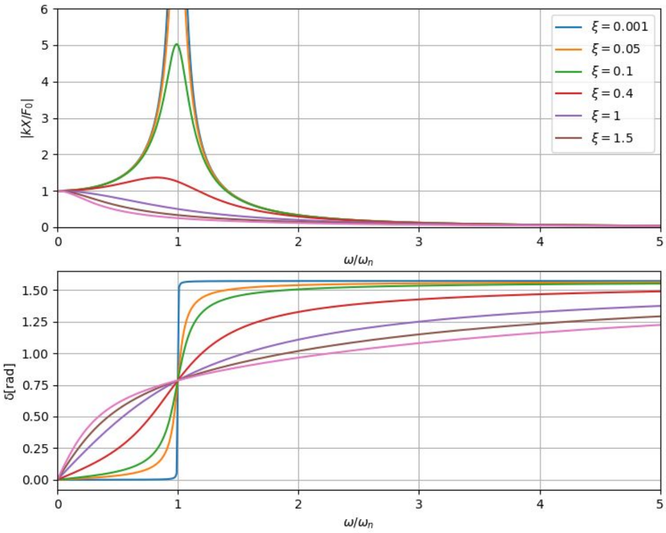

Figura 2: Ampiezza e fase con forzante generica

Spostamento di vincolo:

Con riferimento al sistema di figura 3 in cui $y = Y_0 sin(\omega t)$ rappresenta lo spostamento imposto al vincolo, si può scrivere l'equazione di equilibrio delle forze in direzione verticale:

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) = 0 \quad (6)$$

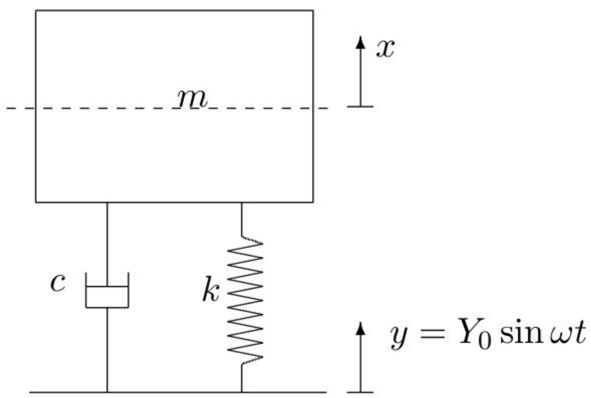

Figura 3: Sistema vibrante ad 1 gdl forzato da spostamento di vincolo

sostituendo z = x — y e $y = Y_0 sin(\omega t)$ si ha:

$$m\ddot{z} + c\dot{z} + kz = -m\ddot{y} = m\omega^2 Y_0 sin(\omega t) \quad (7)$$

usando la notazione vettoriale si ricava la funzione complessa:

$$\mathbf{Z} = \frac{Y_0(\frac{\omega}{\omega_n})^2}{1 - (\frac{\omega}{\omega_n})^2 + i2\varsigma(\frac{\omega}{\omega_n})} \quad (8)$$

La forza trasmessa al vincolo è data da due contributi, uno dello smorzatore, l'altro della molla:

$$F_T = ci\omega \mathbf{Z} e^{i\omega t} + k\mathbf{Z} e^{i\omega t} \quad (9)$$

che può essere scritta come:

$$\frac{F_T}{kY_0} = (\frac{\omega}{\omega_n})^2 \frac{1 + i2\varsigma\frac{\omega}{\omega_n}}{1 - (\frac{\omega}{\omega_n})^2 + i2\varsigma\frac{\omega}{\omega_n}} \quad (10)$$

In figura 4 sono rappresentati i relativi andamenti di modulo e fase.

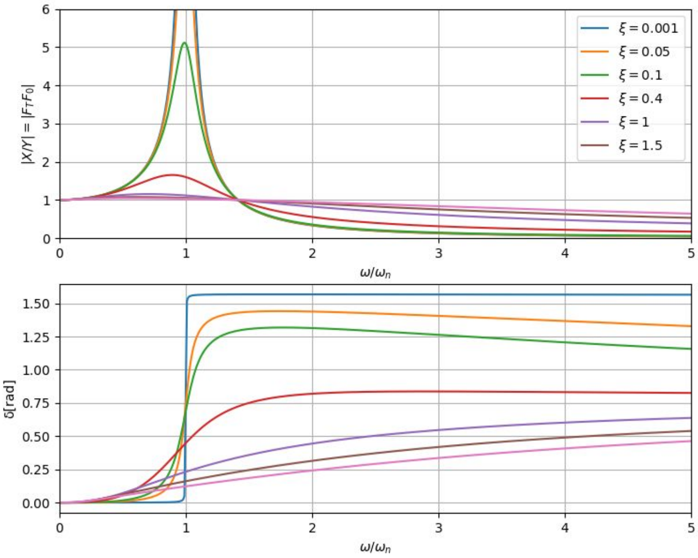

Figura 4: Ampiezza e fase con spostamento di vincolo

Simulazione nel caso in cui il sistema non venga smorzato con condizioni iniziali nulle e con pulsazione della forzante coincidente con quella naturale del sistema si ottiene il seguente andamento:

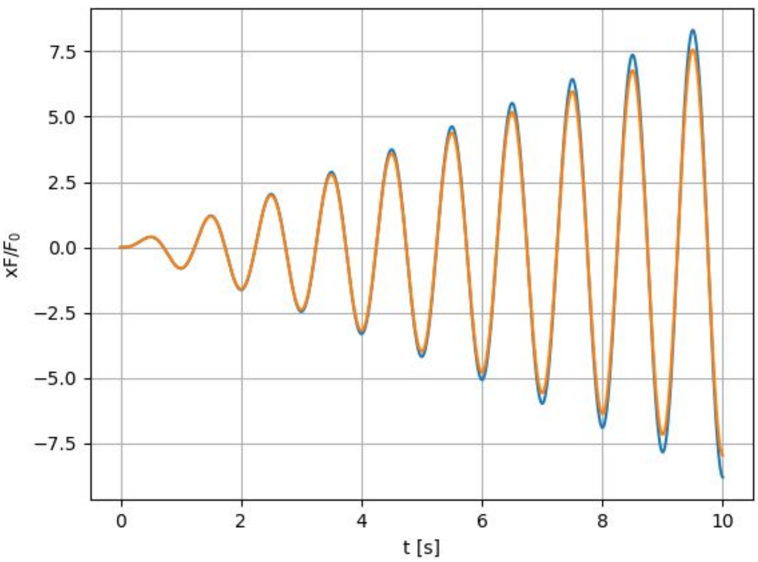

Figura 5: Risposta del sistema con $\omega = \omega_n$, $\zeta = 0$, $x(0) = 0$ e $\dot{x}(0) = 0$

in questo caso l'integrale particolare assume la forma:

$$x_p = -\frac{1}{2}\frac{F_0}{m\omega_n} t cos\omega_n t \qquad (11)$$

di conseguenza l'integrale generale sarà:

$$x_g = A cos\omega_n t + B sin\omega_n t - \frac{1}{2}\frac{F_0}{m\omega_n} t cos\omega_n t \qquad (12)$$

per la determinazione dei coefficienti A e B si determinano in base alle condizioni iniziali:

$$\begin{cases} A = x_0 \\ B = \frac{\dot{x_0}}{\omega_n} + \frac{1}{2}\frac{F_0}{m\omega_n^2} \end{cases} \qquad (13)$$

Codice:

```
# Author: MohsenGhalbi
# Date: 07/01/2020
# Description: Sistema vibrante

import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
from scipy import integrate

# parametri
fn = 1;
wn = 2*np.pi*fn;
xi = 0; # fattore di smorzamento
m = 1;
F0 = 10; # forzante
W = wn; # pulsazione della forzante
tf = 10; # tempo di integrazione
dt = 0.001; # passo di integrazione
t = np.arange(0,tf,dt)
y1 = [0]; # posizione
y2 = [0]; # velocità
y1p = [];
y2p = [];

def ode45_step(f, x, t, dt, *args):
      """
      One step of 4th Order Runge-Kutta method
      """
      k = dt
      k1 = k * f(t, x, *args)
      k2 = k * f(t + 0.5*k, x + 0.5*k1, *args)
      k3 = k * f(t + 0.5*k, x + 0.5*k2, *args)
      k4 = k * f(t + dt, x + k3, *args)
      return x + 1/6. * (k1 + 2*k2 + 2*k3 + k4)

def ode45(f, t, x0, *args):
      """
      4th Order Runge-Kutta method
      """
      n = len(t)
      x = np.zeros((n, len(x0)))
      x[0] = x0
```

```
      for i in range(n-1):
      dt = t[i+1] - t[i]
      x[i+1] = ode45_step(f, x[i], t[i], dt, *args)
      return x

def sist_1gdl(t, y, xi, wn, F0, W, m):
    dy0 = y[1];
    dy1=-2*xi*wn*y[1]-pow(wn,2)*y[0]+F0/m*np.sin(W*t);
    dy = np.array([dy0, dy1]).transpose()
    return dy;

y = ode45(sist_1gdl, t, [y1[0], y2[0]], xi, wn, F0, W, m)

for i in  np.arange(len(t)):
      y1p.append(y2[i]);
      y2p.append(-2*xi*wn*y2[i]-pow(wn,2)*y1[i]+F0/m*np.sin(W*t[i]));
      y1.append(y1[i]+y1p[i]*dt);
      y2.append(y2[i]+y2p[i]*dt);


plt.plot(t,y1[:-1],t,y[:,0]);
plt.xlabel('t [s]')
plt.ylabel('xF/$F_0$')
plt.grid()
plt.show();

########################################################################
#------------------------vibrazione forzata-------------------------#
########################################################################

# Author: Mohsen Ghalbi
# Date: 07/01/2020
# Description: Sistema vibrante Forzato

import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import pylab as pl
import cmath


a = np.arange(0,10,0.01);
xi = [0.001, 0.05, 0.1, 0.4, 1, 1.5, 2];
```

```
for i in np.arange(len(xi)):
    XXst = list(map(lambda x: 1/(1-pow(x,2)+2*xi[i]*complex(0,1)*x),a))
    XYO = list(map(lambda x:
(1+2*xi[i]*complex(0,1)*x)/(1-pow(x,2)+2*xi[i]*complex(0,1)*x),a))
    plt.figure(1)
    ax1 = plt.subplot(211)
    ax1.set_xlim([0, 5])
    ax1.set_ylim(0,6)
    ax1.set_ylabel('$|kX/F_0|$')
    ax1.set_xlabel('$\omega/\omega_n$')
    plt.legend(loc="upper right")
    ax1.grid()
    plt.plot(a, np.abs(XXst), label=r'$\xi=$'+str(xi[i]))
    ax2 = plt.subplot(212)
    ax2.set_xlim([0, 5])
    ax2.set_ylabel('\u03B4[rad]')
    ax2.set_xlabel('$\omega/\omega_n$')
    ax2.grid()
    fig = plt.gcf()
    fig.canvas.set_window_title('Ampiezza e fase con forzante generica')
    plt.plot(a, -np.pi*(np.angle(XXst, deg = True)/360))

    plt.figure(2)
    ax1 = plt.subplot(211)
    ax1.set_xlim([0, 5])
    ax1.set_ylim(0,6)
    ax1.set_ylabel('$|X/Y|=|F_TF_0|$')
    ax1.set_xlabel('$\omega/\omega_n$')
    ax1.grid()
    plt.legend(loc="upper right")
    plt.plot(a, np.abs(XYO), label=r'$\xi=$'+str(xi[i]))
    ax2 = plt.subplot(212)
    ax2.set_xlim([0, 5])
    ax2.set_ylabel('\u03B4[rad]')
    ax2.set_xlabel('$\omega/\omega_n$')
    fig = plt.gcf()
    fig.canvas.set_window_title('Ampiezza e fase con spostamento di
vincolo')
    ax2.grid()
    plt.plot(a, -np.pi*(np.angle(XYO, deg = True)/360))

plt.show()
```