TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

LALITPUR, NEPAL

A
LAB REPORT
ON

Simulation of Mass Spring Damper System (Continuous System)

SUBMITTED BY:

REBATI RAMAN GAIRE

073BCT533

SUBMITTED TO:

SIMULATION AND MODELLING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

INSTITUTE OF ENGINEERING, PULCHOWK CAMPUS

LALITPUR, NEPAL

OBJECTIVE:

1. To develop the mathematical modeling of the (continuous system) mass spring damper system.
2. To determine the state of the system i.e. x, distance moved at different points of time.

THEORY

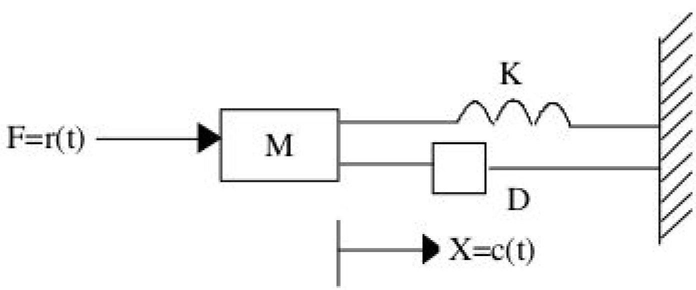
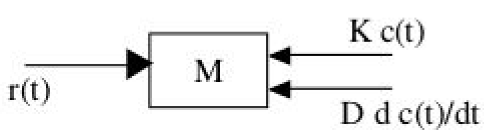


Figure 1: A mass-spring damper system

The free-body diagram of the above system is:



According to Newton's law of motion, we have

Force= Mass × Acceleration

The motion of the system is described by the following differential equation:

$$M.\frac{d2C(t)}{dt2} = K.r(t) - (D.\frac{dC(t)}{dt} + K.C(t))$$

$$M.\frac{d2C(t)}{dt2} + D.\frac{dC(t)}{dt} + K.C(t) = K.r(t)$$

Therefore, the mathematical modeling of a mass-spring damper system with mass M, force r(t), damper coefficient D, spring coefficient K, and displacement C(t) is given by

$$M.\frac{d2C(t)}{dt2} + D.\frac{dC(t)}{dt} + K.C(t) = K.r(t) \text{---------[I]}$$

This equation can be re-written as:

$$\frac{d2C(t)}{dt2} + \frac{D}{M}.\frac{dC(t)}{dt} + \frac{K}{M}.C(t) = \frac{1}{M}.r(t) \text{---------[II]}$$

Equation [I] is the mathematical model of the given Mathematical system.
Comparing equation [II] with the general second-order partial equation.

$$\frac{d2C(t)}{dt2} + 2\xi.W\frac{dC(t)}{dt} + W.W\ C(t) = W.W.F$$

Where,

Damping ration ξ = D/2.M.W
Angular frequency of oscillation W = K / M

The response of this system with the unit-step function depends on the value of ξ. The condition for the motion to occur without oscillation requires that ξ>=1. Show how x varies in response to a steady force applied at time t=0 for the various values of ξ.

SOURCE CODE FOR SIMULATION

Now I've used python programming language with framework numpy, scipy, and matplotlib libraries to simulate the above mathematical model to determine the state of the system i.e. distance moved at different points of time:

```
# Import Libraries
import numpy as np
from scipy.stats import chi2
import matplotlib.pyplot as plt
%matplotlib inline
```

```
F = 4
W = 0.7
t0 = 0
T = 50
dt = 0.01
N = (T-t0)/dt
t = np.arange(0, N-1)*dt
u = np.zeros(int(N))
```

```
def du(u, z):
 return np.array([u[1], (W**2) * (F - u[0]) - 2 * z * W * u[1]])
```

```
zeta = [-0.2, 0.01, 0.7, 1.8]
```

```
for k in range(0, len(zeta)):
 for i in range(0, int(N-2)):
   m1 = du(u[i:i+2], zeta[k])
   m2 = du(u[i:i+2] + (m1/2), zeta[k])
   m3 = du(u[i:i+2] + (m2/2), zeta[k])
   m4 = du(u[i:i+2] + m3, zeta[k])
   u[(i+1):(i+3)] = u[i:i+2] + dt * (m1 + 2 * m2 + 2 * m3 + m4) / 6
 plt.plot(u[:-1], label= f"{zeta[k]}")
plt.legend()
plt.xlabel("Time (sec)")
plt.ylabel('Distance x(t)')
plt.title('Variation of x(t) with different value of Zeta')
```

RESULT AND VISUALIZATION

After running the script, I obtained the following graph that shows the various values of the distance moved at different times for different values of ξ.

DISCUSSION AND CONCLUSION

From this lab session, we became familiar with the concept of a continuous system. We explored the topic with an example of a Mass Spring Damper System. We also learned to develop the mathematical model of the system using a second-order differential equation. We wrote a script in python programming language to simulate the model of the system. We used python’s numpy library as a data structure to store an array of values, the Runge-Kutta 4 method to solve the differential equation, and matplotlib to visualize the graph showing the state of the system i.e. the distence moved at different point in time.