

**DYNAMIC ANALYSIS OF SDoF & MDoF SYSTEMS
SUBJECTED TO SINUSOIDAL, STEP & RAMP LOADING
USING MATLAB**

By

SIDDHARTH NARAYAN BHARDWAJ	101450
NEHA GUPTA	091634
PANKAJ CHAUDHARY	101641

Under the guidance of

Mrs. Poonam Dhiman



Submitted in Partial Fulfilment of the Degree of

BACHELOR OF TECHNOLOGY

In

CIVIL ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN (H.P.)

MAY-2014

CERTIFICATE

This is to certify that the work entitled, **“Dynamic analysis (free & forced vibration) of SDoF & MDoF systems subjected to sinusoidal, step and ramp loading using MATLAB”** submitted by **Siddharth Narayan Bhardwaj, Neha Gupta & Pankaj Chaudhary** in partial fulfillment for the award of degree of Bachelor of Technology in Civil Engineering of Jaypee University of Information Technology has been carried out under our supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Mrs. Poonam Dhiman

Assistant Professor

Civil Engineering

Dr. Ashok Kumar Gupta

Professor and Head of Department

Civil Engineering

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to my elite guide & mentors **Dr. Ashok Kumar Gupta**, Professor and Head of Department, Civil Engineering and **Mrs. Poonam Dhiman**, Assistant Professor, Civil Engineering, under whom this project work has been successfully completed. I am fortunate enough to get the invaluable opportunity of doing project under the able guidance of my esteemed and erudite guides. I am thankful for their persistent interest, constant encouragement, vigilant supervision and critical evaluation. Their encouraging attitude has always been a source of inspiration for me. Their helping nature, invaluable suggestions and scholastic guidance are culminated in the form of the present work.

It gives me pleasure to express my thanks to all my teachers for their ingenious and sturdy motivation throughout my thesis work.

All the words do not sum up the pain and hardship that my parents had to face in my ascent to this achievement, whose sacrifice and love has been the guiding principles of life. Despite the agony and inconvenience, they provided me everything throughout my academics. Their unlimited faith and confidence has made me whatever I am today.

Finally by no means least, I thank the **Almighty** for all the things he has given and for the things to be given in future.

Place: Waknaghat

Date: 15th May, 2014

Siddharth Narayan Bhardwaj (101450)

Neha Gupta (091634)

Pankaj Chaudhary (101641)

ABSTRACT

Impulsive force is a force of large magnitude that acts over a short time interval. In practice, vehicles and cranes are subjected to impulsive loads. In this chapter, the response of the single-degree-of-freedom system with or without damping subjected to impulsive loads is considered. The concept of response spectrum, which is a very useful tool in the design, is also illustrated.

Structural engineers are familiar with the analysis of structures for static loads in which a load is applied to the structure and a single solution is obtained for the resulting displacements and member forces. When considering the analysis of structures for dynamic loads, the term dynamic simply means “time-varying.” Hence, the loading and all aspects of the response vary with time. This results in possible solutions at each instant during the time interval under consideration. From an engineering standpoint, the maximum values of the structural response are usually the ones of particular interest, especially when considering the case of structural design.

Two different approaches, which are characterized as either deterministic or nondeterministic, can be used for evaluating the structural response to dynamic loads. If the time variation of the loading is fully known, the analysis of the structural response is referred to as a deterministic analysis. This is the case even if the loading is highly oscillatory or irregular in character. The analysis leads to a time history of the displacements in the structure corresponding to the prescribed time history of the loading. Other response parameters such as member forces and relative member displacements are then determined from the displacement history.

If the time variation of the dynamic load is not completely known but can be defined in a statistical sense, the loading is referred to as a random dynamic loading, and the analysis is referred to as nondeterministic. The nondeterministic analysis provides information about the displacements in a statistical sense, which results from the statistically defined loading. Hence, the time variation of the displacements is not determined, and other response parameters must be evaluated directly from an independent nondeterministic analysis rather than from the displacement results. Methods for nondeterministic analysis are described in books on random vibration.

LIST OF FIGURES

Fig. No	Description	Page No.
1.6	Prescribed Dynamic Loading	3
1.7	Basic Dynamic System	6
1.8	Simple Harmonic Motion	8
2.1	Single Degree of Freedom System	11
2.2	Mass Spring Damper System	11
2.3	Undamped Free Vibration	14
2.3.3	Free Vibration of underdamped, critically damped and overdamped	16
2.5	Response for undamped and overdamped for various cases	18
2.6.1	Simple Harmonic Motion (Response of Undamped case to Harmonic Force)	22
2.6.2	Simple Harmonic Motion (Response of Damped case to Harmonic Force)	23
3.1	Step Loading	26
3.4	Response for Step Loading for different natural frequencies and different levels of damping	30
4.4	Response for Ramp Loading for different natural frequencies and different levels of damping	35
5.1	Multi Degree of Freedom System	37
5.2	Mass Spring Damper System	38
5.3.1	Typical Two Degree of Freedom Free Oscillator	38
5.3.2	Free Body Diagram of Typical Two Degree of Freedom Free Oscillator	38
5.5	Response for MDoF System for free vibrations (Undamped)	41
5.6.1	Typical Two Degree of Freedom Free Oscillator	45
5.6.2	MDoF Damped Oscillator	45
5.6.3	Free Body Diagram of MDoF Damped Oscillator	46
5.8	Response for MDoF System for free vibrations (Damped)	49
6.2	Response for Universal Vibration Solver (Free n-degree)	55
6.4	Response for Universal Vibration Solver (Forced n-degree)	57

CONTENTS

CERTIFICATE	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
LIST OF FIGURES	iv
Chapter 1 Introduction	1
1.1 What is Dynamics	
1.2 Linear and Rotational Dynamics	
1.3 Force	
1.4 Newton's Law	
1.5 Essential Characteristics of Dynamic Problem	
1.6 Dynamic Loads and Analysing Methods	
1.7 Formulation of Equation of Motion	
1.8 Basic Definition (SHM)	
Chapter 2 Single Degree of Freedom System	11
2.1 Introduction	
2.2 Mass Spring Damper System	
2.3 Analysis of Free Vibrations	
2.4 MATLAB Coding of SDoF System for Free Vibrations	
2.5 Response for MATLAB Coding of SDoF System for Free Vibrations	
2.6 Analysis of Forced Vibration	
2.7 MATLAB Coding of SDoF System for Forced Vibrations	

Chapter 3 Step Response of a SDoF System 26

- 3.1 Introduction
- 3.2 Step Loading
- 3.3 MATLAB Coding of SDoF System for Step Loading
- 3.4 Response for MATLAB Coding of SDoF System for Step Loading

Chapter 4 Response of SDoF System to Ramp Input 33

- 4.1 Introduction
- 4.2 Ramp Loading
- 4.3 MATLAB Coding of SDoF System for Ramp Loading
- 4.4 Response for MATLAB Coding of SDoF System for Ramp Loading

Chapter 5 Multi Degree of Freedom System 37

- 5.1 Introduction
- 5.2 Mass Spring Damper System
- 5.3 Analysis of Free Vibrations (Undamped)
- 5.4 MATLAB Coding of MDoF System for Free Vibration (Undamped)
- 5.5 Response for MATLAB Coding of MDoF System for Free Vibration (Undamped)
- 5.6 Analysis of Free Vibrations (Damped)
- 5.7 MATLAB Coding of MDoF System for Free Vibration (Damped)
- 5.8 Response for MATLAB Coding of MDoF System for Free Vibration (Damped)

Chapter 6 A Universal Vibration Solver 53

- 6.1 MATLAB Coding for Universal Vibration Solver (Free n-degree)
- 6.2 Response for MATLAB Coding for Universal Vibration Solver (Free n-degree)
- 6.3 MATLAB Coding for Universal Vibration Solver (Forced n-degree)
- 6.4 Response for MATLAB Coding for Universal Vibration Solver (Forced n-degree)

CONCLUSIONS 58

REFERENCES 59

CHAPTER 1

INTRODUCTION

1.1 WHAT IS DYNAMICS?

The dynamic behavior of structures is an important topic in many fields. Aerospace engineers must understand dynamics to simulate space vehicles and aeroplanes, while mechanical engineers must understand dynamics to isolate or control the vibration of machinery. In civil engineering, an understanding of structural dynamics is important in the design and retrofit of structures to withstand severe dynamic loading from earthquakes, hurricanes and strong winds, or to identify the occurrence and location of damage within an existing structure.

Dynamics is a branch of physics (specifically classical mechanics) concerned with the study of forces and torques and their effect on motion, as opposed to *kinematics*, which studies the motion of objects without reference to its causes. Isaac Newton defined the fundamental physical laws which govern dynamics in physics, especially his second law of motion.

1.2 LINEAR AND ROTATIONAL DYNAMICS

The study of dynamics falls under two categories: **linear and rotational**. Linear dynamics pertains to objects moving in a line and involves such quantities as force, mass/inertia, displacement (in units of distance), velocity (distance per unit time), acceleration (distance per unit of time squared) and momentum (mass times unit of velocity). Rotational dynamics pertains to objects that are rotating or moving in a curved path and involves such quantities as torque, moment of inertia/rotational inertia, angular displacement (in radians or less often, degrees), angular velocity (radians per unit time), angular acceleration (radians per unit of time squared) and angular momentum (moment of inertia times unit of angular velocity). Very often, objects exhibit linear and rotational motion.

For classical electromagnetism, it is Maxwell's equations that describe the dynamics. And the dynamics of classical systems involving both mechanics and electromagnetism are described by the combination of Newton's laws, Maxwell's equations, and the Lorentz force.

1.3 FORCE

From Newton, force can be defined as an exertion or [pressure](#) which can cause an object to move. The concept of force is used to describe an influence which causes a free body (object)

to accelerate. It can be a push or a pull, which causes an object to change direction, have new velocity, or to deform temporarily or permanently. Generally speaking, force causes an object's state of motion to change.

1.4 NEWTON'S LAWS

Newton described force as the ability to cause a mass to accelerate. His three laws can be summarized as follows:

1. **First law**: If there is no net force on an object, then its **velocity** is constant. The object is either at rest (if its velocity is equal to zero), or it moves with constant speed in a single direction.
2. **Second law**: The rate of change of linear momentum **P** of an object is equal to the net force **F_{net}**, i.e., $d\mathbf{P}/dt = \mathbf{F}_{net}$.
3. **Third law**: When a first body exerts a force **F₁** on a second body, the second body simultaneously exerts a force **F₂ = -F₁** on the first body. This means that **F₁** and **F₂** are equal in magnitude and opposite in direction.

Newton's Laws of Motion are valid only in an inertial frame of reference

1.5 ESSENTIAL CHARACTERISTICS OF A DYNAMIC PROBLEM

1.5.1) Typical dynamical problems:

- Structural vibration under seismic excitation
- Vibration of large span bridge and high-rise building under turbulent wind loads
- Vehicle vibration due to road roughness
- Bomb blast pressure on civil engineering structures

1.5.2) Differences between static and dynamic loads:

- **Static loads**: magnitude, direction and position remains constant or varies slowly with time, such as structural self weight, snow loads and etc.
- **Dynamic loads**: magnitude, direction and position varies rapidly with time or the loads are suddenly applied to or removed from the structure.

1.5.3) Essential characteristics :

- **Time varying nature**: a succession of solutions has to be established corresponding to all times of interest in the response history.
- **Inertial forces**: represent a significant component of the total loads, the dynamic characteristics of the problem are reflected in its solution.

1.6 DYNAMIC LOADS AND ANALYZING METHODS

1.6.1) Prescribed dynamic loading:

- The time varying characteristics of the loading is fully prescribed, even though it may be highly oscillatory or irregular in form.
- Deterministic analysis: a direct structural time history responses computation corresponding to the prescribed loading history.



1.6 .2) Random dynamic loading:

- The time varying characteristics is not completely prescribed but can be defined in a statistical sense.
- Nondeterministic analysis: provides only statistical information about the structural response from the statistically defined loading.

1.6.3) Prescribed dynamic loading can be categorized as periodic loadings and nonperiodic loadings:

- **Periodic loading**

(1) Harmonic loading -loading in the form of sinusoidal variation

e.g. loading due to rotating machinery

(2) Other periodic loadings -Complex periodic loading which can not be represented by a harmonic function but by the sum of a series of simple harmonic components

e.g. hydrodynamic pressures generated by a propeller at the stern of a ship

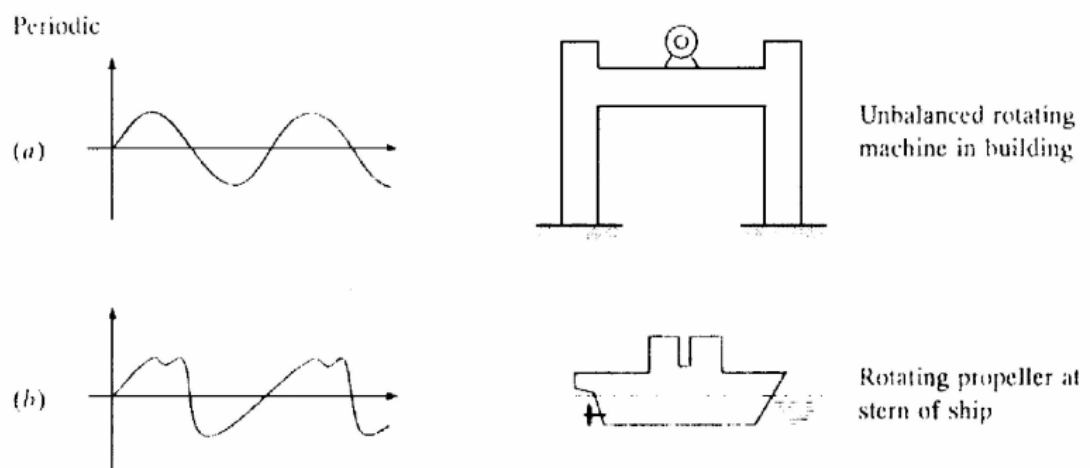
- **Nonperiodic loading**

(1) Impulsive loading- short-duration high intensity loading which is generally described by sine-wave, triangular or rectangular function

e.g. blast or explosion loadings

(2) General forms of loading - Represented by any function or just some data sets.

e.g. earthquake excitations



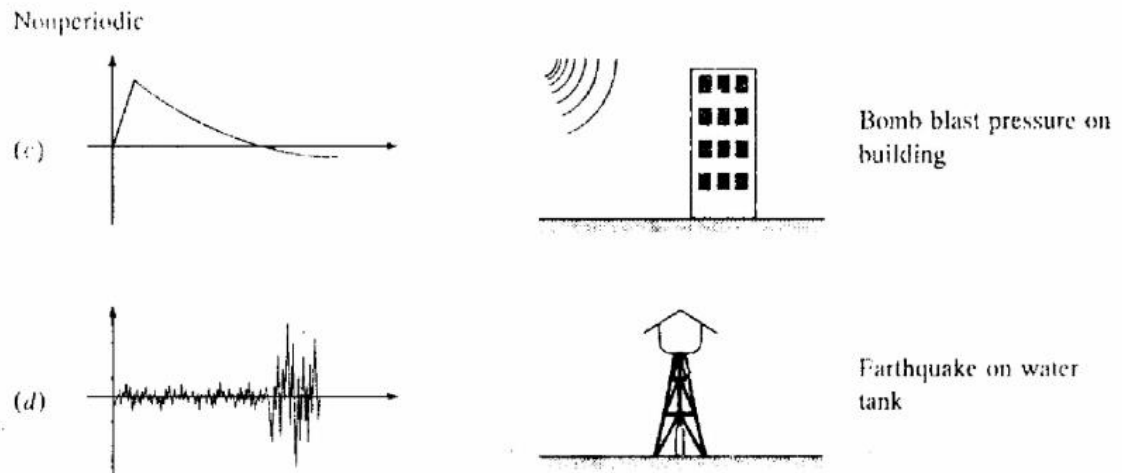


Fig 1.6

1.7 FORMULATION OF EQUATIONS OF MOTION

Equation of motion (EOM): the mathematical expressions defining the dynamic displacements. Its solution provides the required displacement time histories.

1.7.1) Basic dynamic system

- SDOF (Single-Degree-of-Freedom-System): structural vibration state can be determined by only one generalized coordinates.
- Some system which can be simplified to be a SDOF system:

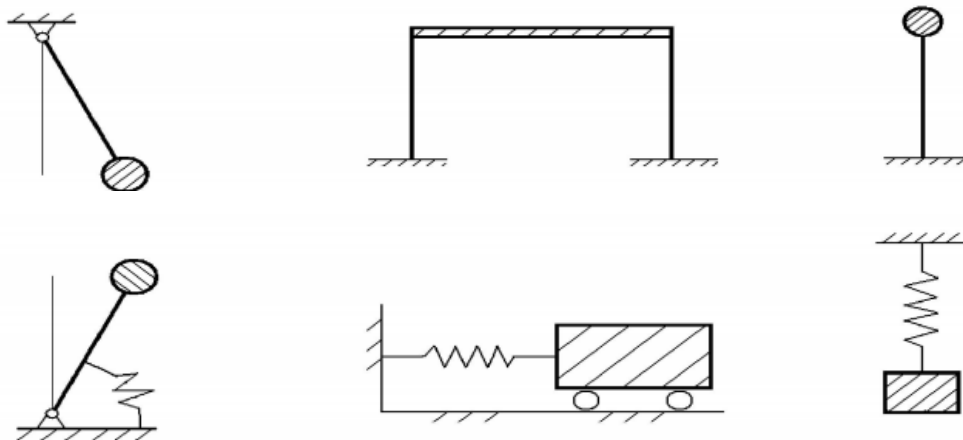


Fig .1.7a

1.7.2) Two Typical SDOF Systems

- Essential physical properties: lumped mass m ; damping c ; stiffness k
- Two systems are the equivalent and thus their vibration responses can be described by the same EOM

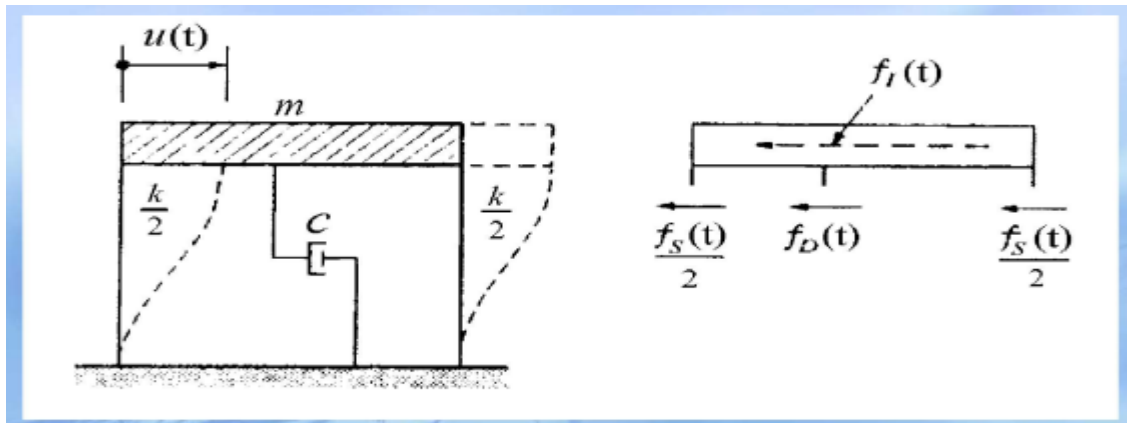


Fig 1.7b.1 A one-floor shear type frame

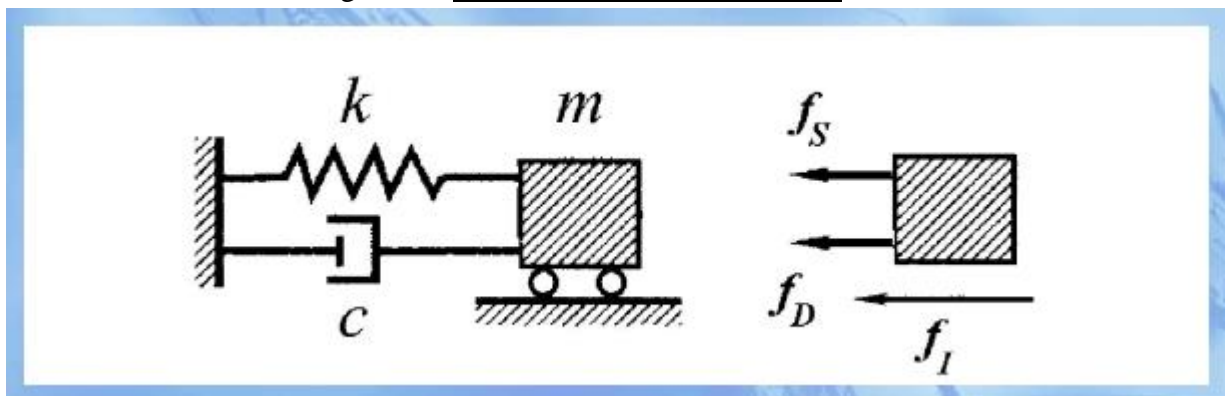


Fig 1.7b.2 Mass-spring-damper system

1.7.3) Inertial Force

The force which keep the original state of the system. It equals to the product of the mass and acceleration and its direction is opposite to the direction of acceleration.

$$f_I = m\ddot{u}$$

I – Inertial;

m – Mass;

\ddot{u} – Acceleration of the mass

Spring Resisting Force

Spring resisting force. It equals to the product of the spring stiffness and displacement and its direction is opposite to the direction of displacement.

$$f_s = ku$$

s — Spring

k — Spring Stiffness

u — Displacement of the mass

Damping Force

The force which induces the energy loss of the system. It equals to the product of the damping coefficient and velocity of mass.

$$f_D = c\dot{u}$$

D — Damping

c — Damping coefficient

\dot{u} — Velocity of mass

1.8. BASIC DEFINITIONS

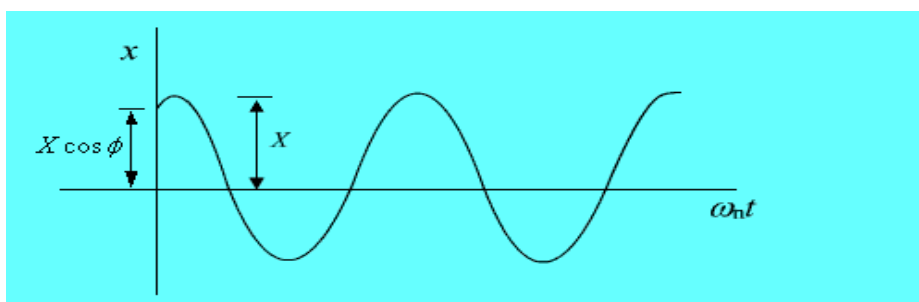


Fig 1.8 Simple harmonic motion

1.8.1) Oscillating Motions:

- The study of vibrations is concerned with the oscillating motion of elastic bodies and the force associated with them.
- All bodies possessing mass and elasticity are capable of vibrations.
- Most engineering machines and structures experience vibrations to some degree and their design generally requires consideration of their oscillatory motions.
- Oscillatory systems can be broadly characterized as linear or nonlinear. All systems tend to become nonlinear with increasing amplitudes of oscillations.
- There are two general classes of vibrations– free and forced.

1.8.2) Free vibrations

- Free vibration takes place when a system oscillates under the action of forces inherent in the system itself due to initial disturbance, and when the externally applied forces are absent.
- The system under free vibration will vibrate at one or more of its natural frequencies, which are properties of the dynamical system, established by its mass and stiffness distribution.
- Natural frequency is the frequency at which a system naturally vibrates once it has been set into motion.

1.8.3) Forced Vibrations

- The vibration that takes place under the excitation of external forces is called forced vibration.
- If excitation is harmonic, the system is forced to vibrate at excitation frequency . If the frequency of excitation coincide with one of the natural frequencies of the system, a condition of resonance is encountered and dangerously large oscillations may result, which results in failure of major structures, i.e., bridges, buildings, or airplane wings etc.

- Thus calculation of natural frequencies is of major importance in the study of vibrations
- Because of friction & other resistances vibrating systems are subjected to damping to some degree due to dissipation of energy
- .
- Damping has very little effect on natural frequency of the system, and hence the calculations for natural frequencies are generally made on the basis of no damping.
- Damping is of great importance in limiting the amplitude of oscillation at resonance.

1.8.4) Damping

- Damping is an effect that reduces the amplitude of oscillations in an oscillatory system particularly in a harmonic oscillator.
- This effect is linearly related to the velocity of the oscillations.
- This restriction leads to a linear differential equation of motion, and a simple analytic solution.

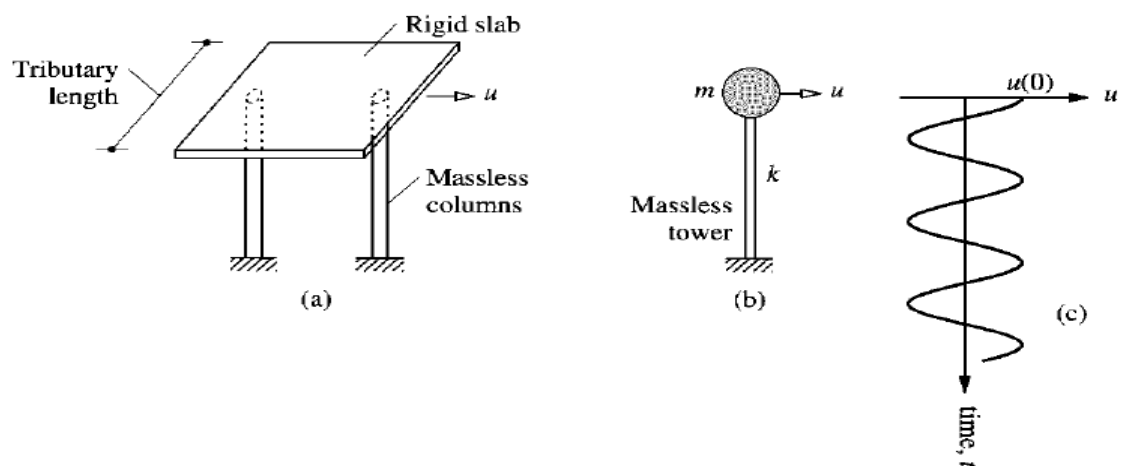
CHAPTER 2

Single degree of freedom systems

2.1 Introduction

We begin our study of structural dynamics with simple structures, such as an elevated water tank. We are interested in understanding the vibrations of these structures when subjected to a lateral (horizontal) force at the top or horizontal ground motion due to an earthquake.

We call these structures simple because they can be idealized as a concentrated lumped mass (M) supported by a massless structure with stiffness (K) in the lateral direction.



a) Idealized pergola b) idealized water tank c) free vibration due to initial displacement

fig 2.1a)

The system considered is shown in the figure. It consists of a mass (m) concentrated at the roof-level, a mass less frame that provides stiffness to the system, and a viscous damper that dissipates vibrational energy of the system. The beam and columns are assumed to be inextensible axially.

This system may be considered as an idealisation of a one storey structure each structure member (beam, column, wall etc.) of the actual structure contributes to the inertial (mass), elastic(stiffness & flexibility), and energy dissipation (damping) properties of the structure,. In the idealized system, however, each of these properties is concentrated in 3 separate, pure components: mass component, stiffness component, and damping component

The number of independent displacements required to define the displaced positions of all the masses relative to original position is called the number of degrees of freedom (DOFs) for dynamic analysis. More Dofs are typically necessary to define the stiffness properties of a structure compared to the dofs necessary for dynamic analysis.

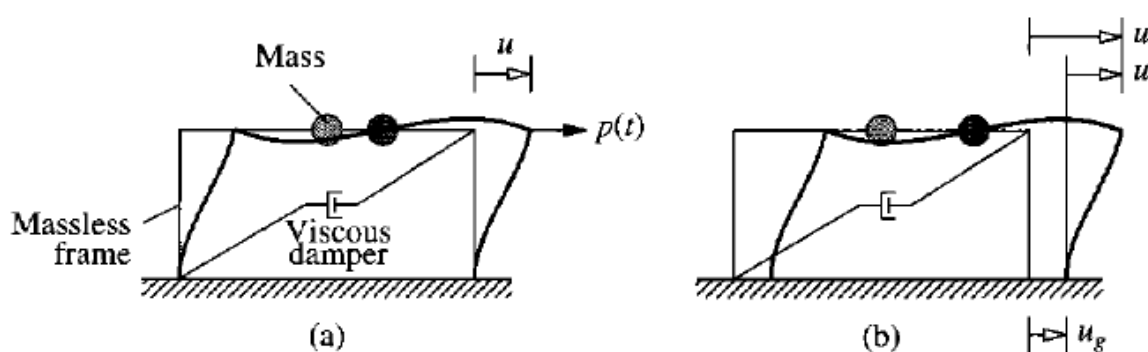


Fig 2.1b) single degree of freedom system: a) applied force $p(t)$ b) earthquake induced motion

2.2 Mass-Spring-Damper System

We have introduced the SDF system by idealizing a one-story structure (fig.2.2a).

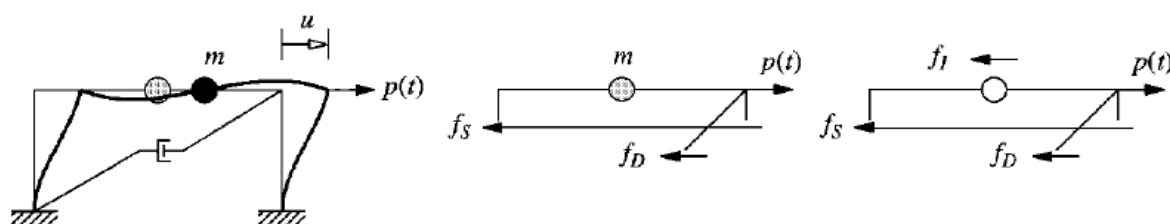
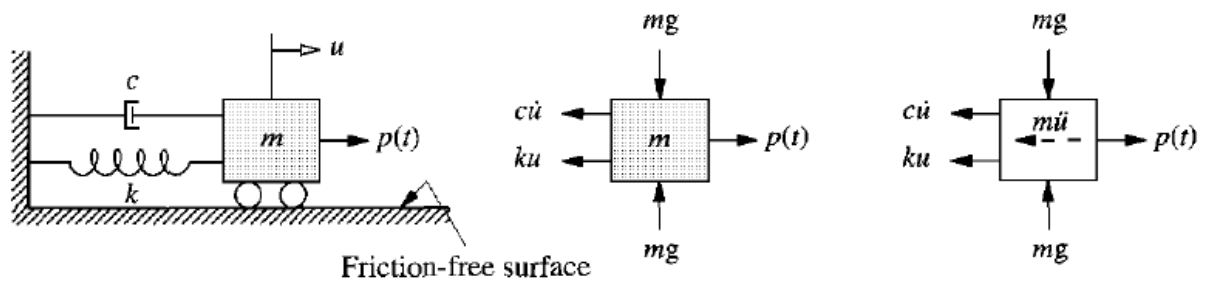
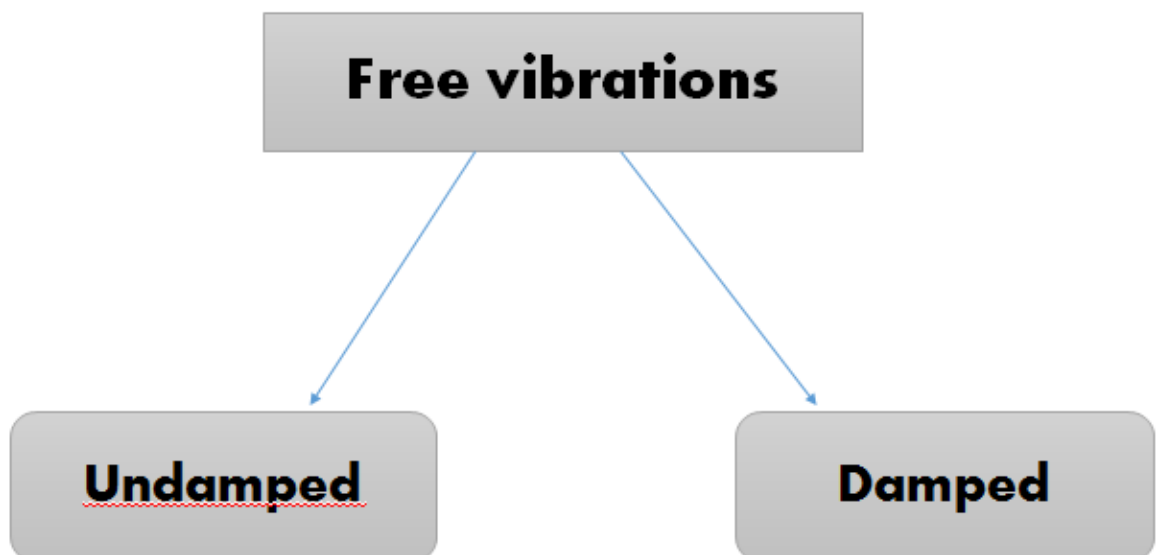


Fig 2.2a

However the classic sdf system is the mass-spring-damper system of (fig 2.2b). If we consider the spring and damper system to be massless, the mass to be rigid, and the motion to be in one direction- x-axis, then we have a sdf system. The fig shows the forces acting on the mass; these include the resisting forces $f_S = ku$, exerted by linear spring of stiffness k, and the damping resisting force $f_D = c\dot{u}$, due to linear viscous damper.

Fig 2.2b) Spring-Mass-Damper System

2.3 Analysis of Free Vibrations



2.3.1) Introduction

A structure is said to undergo free vibration when it is disturbed from its static equilibrium position and then allowed to vibrate without any external dynamic excitation. In this chapter we study vibration leading to the notions of the natural vibrational frequency and damping ratio for a SDF system. The rate at which the motion decays depends on the damping ratio.

2.3.2) Undamped free vibration

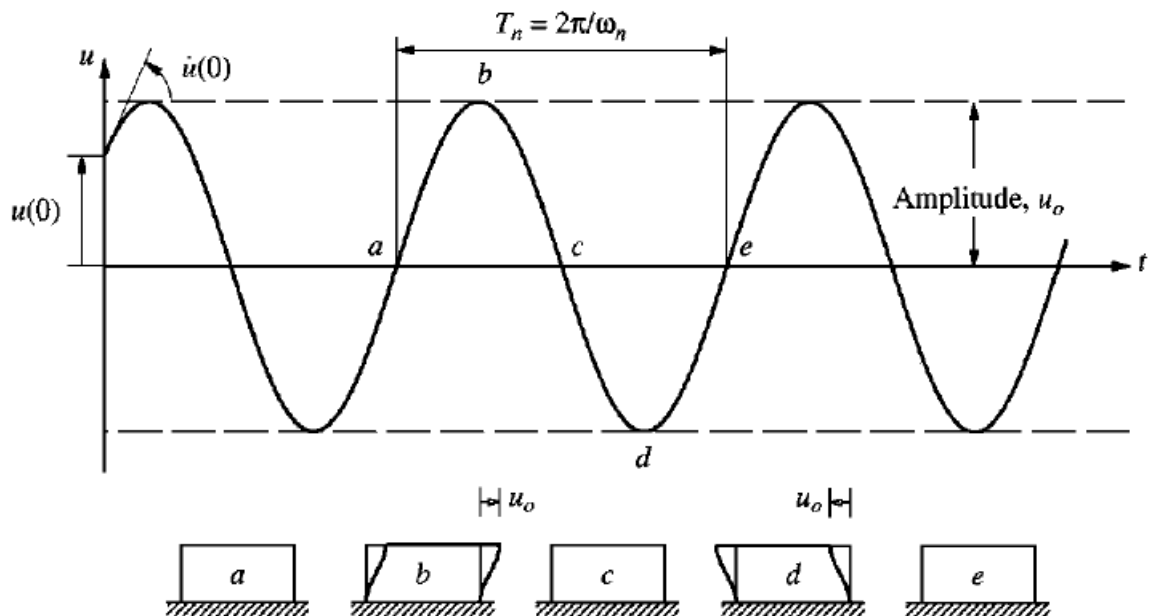


Fig 2.3.2) Free vibration of a system without damping

The motion of linear SDF systems, visualized as an idealized one-story frame or a mass spring damper system, subjected to external force $p(t)$ is given by-

$m\ddot{u} + c\dot{u} + ku = p(t)$. Setting $p(t) = 0$ gives the differential equation governing free vibration of the system (for systems without damping).

$$m\ddot{u} + ku = 0$$

Thus the corresponding equation of motion is-

Free vibration is initiated by disturbing the system from its static equilibrium position by imparting the mass some displacement $u(0)$ and velocity $v(0)$ at time zero, defined

as at the instant the motion is initiated: $u = u(0)$ $\dot{u} = \dot{u}(0)$

Solution to the equation of motion is-

$$u(t) = u(0) \cos \omega_n t + \frac{\dot{u}(0)}{\omega_n} \sin \omega_n t$$

where

$$\omega_n = \sqrt{\frac{k}{m}}$$

2.3.3 VISCOUSLY DAMPED FREE VIBRATION

EQUATION OF MOTION: $m\ddot{u} + c\dot{u} + ku = 0$

$$\zeta = \frac{c}{2m\omega_n} = \frac{c}{c_{cr}}$$

Defining ζ : ζ , where c_{cr} is the critical damping Coefficient and ζ is the damping Ratio or fraction of damping.

Solution of equation of motion is:

$$u(t) = e^{-\zeta\omega_n t} \left[u(0) \cos \omega_D t + \left(\frac{\dot{u}(0) + \zeta\omega_n u(0)}{\omega_D} \right) \sin \omega_D t \right]$$

where

$$\omega_D = \omega_n \sqrt{1 - \zeta^2}$$

Fig. 2.3.3 shows a plot of the motion $u(t)$ due to initial displacement $u(0)$ for three values ζ . If $c = c_{cr}$ or $\zeta = 1$ the system returns to its equilibrium without oscillating. If

$c > c_{cr}$ or $\zeta > 1$, again the system does not oscillate and returns to its equilibrium position, as in the case $\zeta = 1$ case, but at a slower rate. If $c < c_{cr}$ or $\zeta < 1$, the system oscillates about its equilibrium position with a progressively decreasing amplitude.

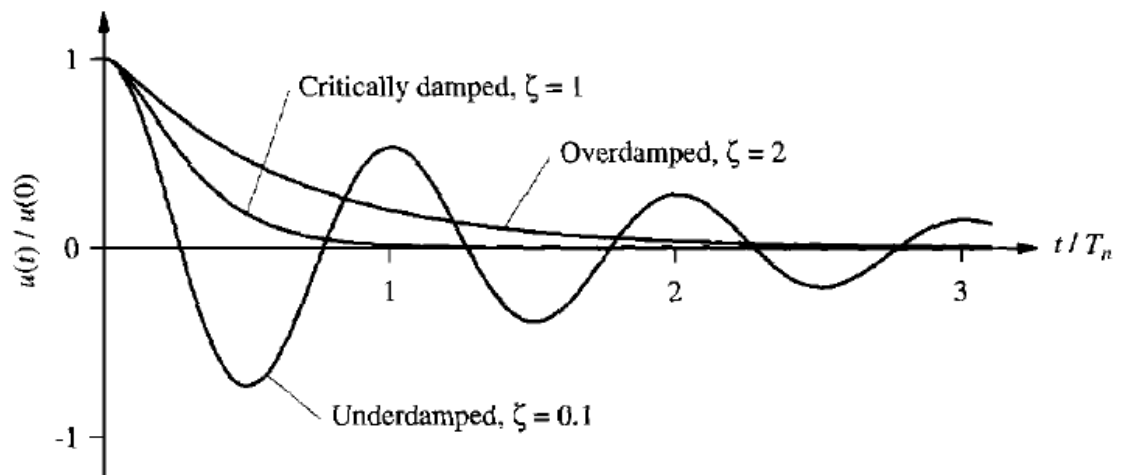


Fig 2.3.3) Free vibration of underdamped, critically damped and overdamped systems

2.4 MATLAB coding for a SDoF system for free vibrations (damped and undamped).

```
clear all
m = 100;
k = 1000;
% m = input('Enter mass of the structure in kg : ');
% k = input('Enter stiffness in N/m : ');
z = [0 0.1 1 1.5];
% zeta = input('Enter damping ratio (zero for undamped case) : ');
% check = input('Enter 0 for free vibration, 1 for forced vibration : ');
wn = sqrt(k/m);
Tn = 2*pi/wn;
dt = 0.01; % time step no.
t = 0:dt:N;
```

Free vibration analysis

```
U0 = input('Enter initial displacement of the structure (m) : ');
V0 = input('Enter initial velocity of the structure (m/s) : ');
j=1;
if zeta == 1      %% Check if critical damping
for i = 0:dt:N
    zeta=z(3);
    U_crit(j) = exp(-wn*i)*(U0+(V0+wn*U0)*i);
    j=j+1;
end
```

If overdamped

```

zeta = z(4);
A=(V0+(zeta+sqrt(zeta^2-1))*wn*U0)/(2*wn*sqrt(zeta^2-1));
B=(-V0-(zeta-sqrt(zeta^2-1))*wn*U0)/(2*wn*sqrt(zeta^2-1));
j=1;
for i = 0:dt:N
    U_over(j) =A*exp((-zeta+sqrt(zeta^2-1))*wn*i)+B*exp((-zeta-sqrt(zeta^2-1))*wn*i);
    j=j+1;
end

```

Underdamped case

```

zeta=z(2);
wd = wn*sqrt(1-zeta^2);
j=1;
for i = 0:dt:N
    U_under(j)=exp(-zeta*wn*i)*(U0*cos(wd*i)+(V0+zeta*wn*U0)*sin(wd*i)/wd);
    %%% Underdamped case
    j=j+1;
end
zeta = z(1);
j=1;
for i = 0:dt:N
    U_un(j)=exp(-zeta*wn*i)*(U0*cos(wn*i)+(V0+zeta*wn*U0)*sin(wd*i)/wn);
    %%% Underdamped case
    j=j+1;
end
subplot(2,1,1);
plot(t,U_un);
xlabel('Time (s)');
ylabel('Response (m)');
title('Displacement vs time curve for UNDAMPED system');
grid on
gtext('Xi = 0');
subplot(2,1,2)
plot(t,U_over,t,U_crit,t,U_under);
grid on
xlabel('Time (s)');
ylabel('Response (m)');
title('Displacement vs time curve for DAMPED system');
gtext('Xi < 1');

```



```

gtext('Xi = 1');
gtext('Xi > 1');
data = [t,U_un',U_over',U_crit',U_under'];
xlswrite('freevibration.xlsx',data,'Sheet1');

```

2.5 Response of the above source code

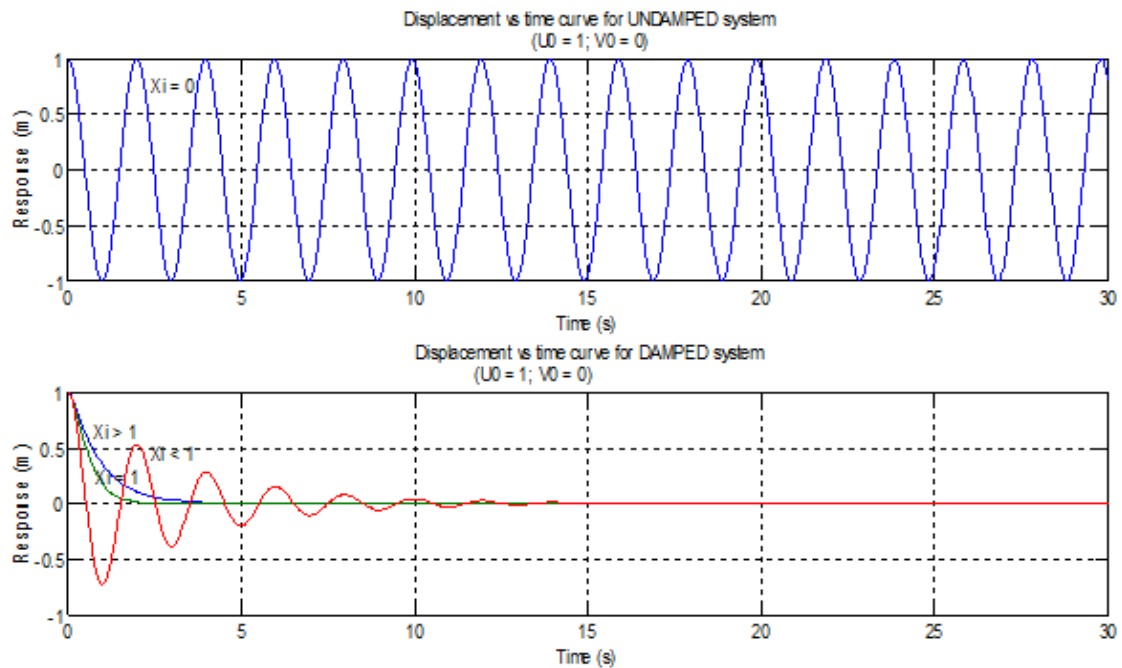


Fig. 2.5.1) Displacement vs time curve for undamped and damped vibration for $U(0)=1$ and $V(0)=0$

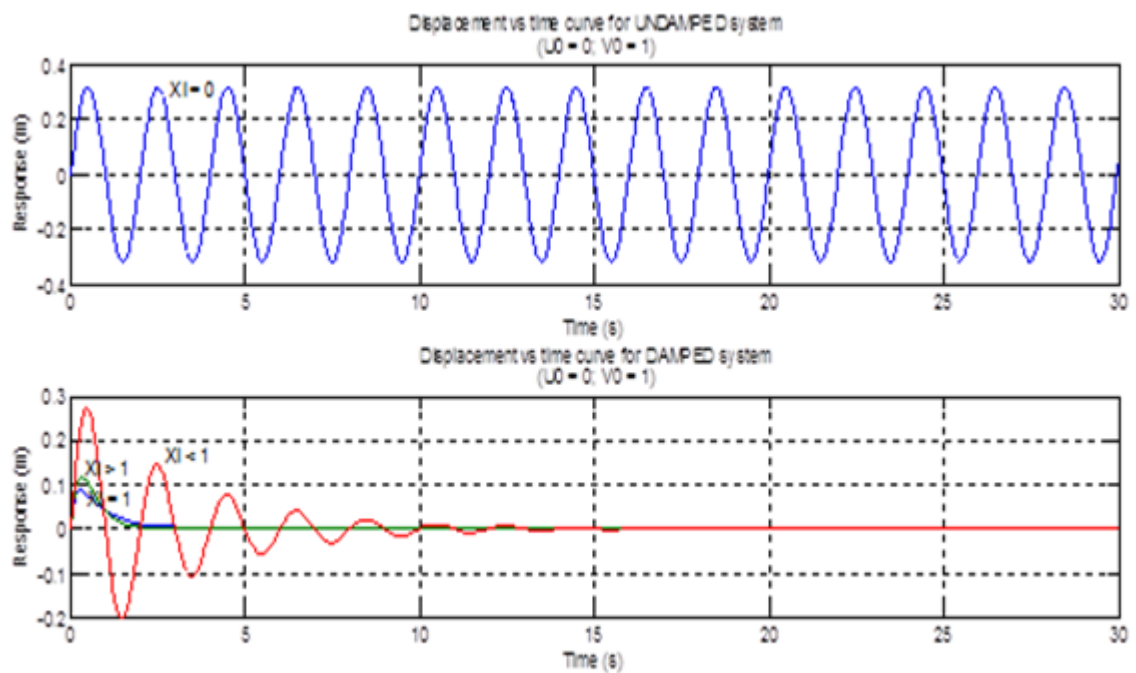


Fig 2.5.2) Displacement vs time curve for underdamped and damped vibration for $U(0)=0$

and $V(0)=1$

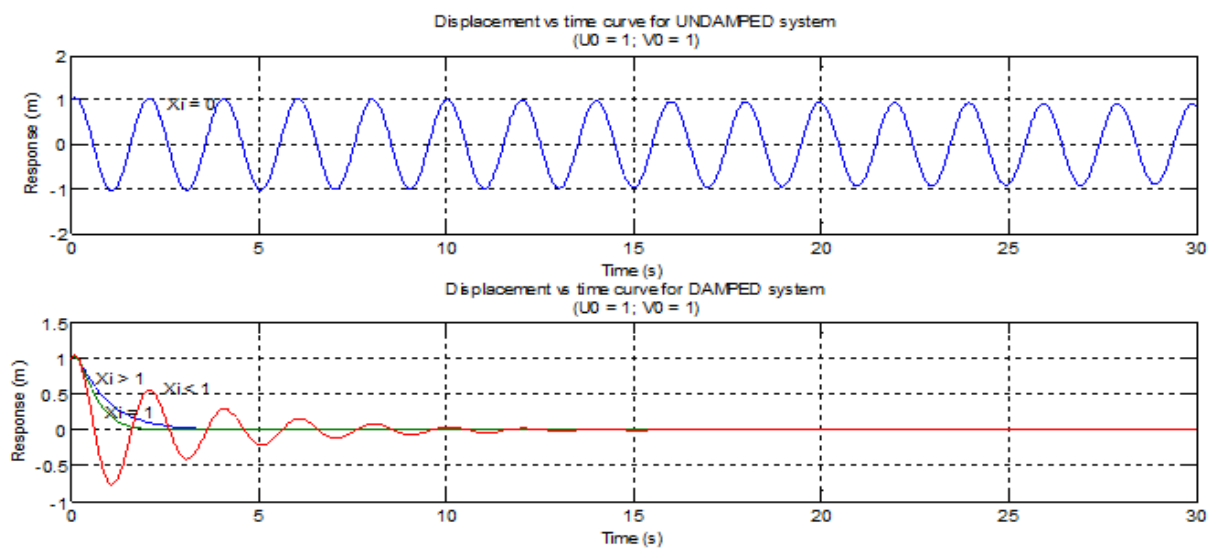
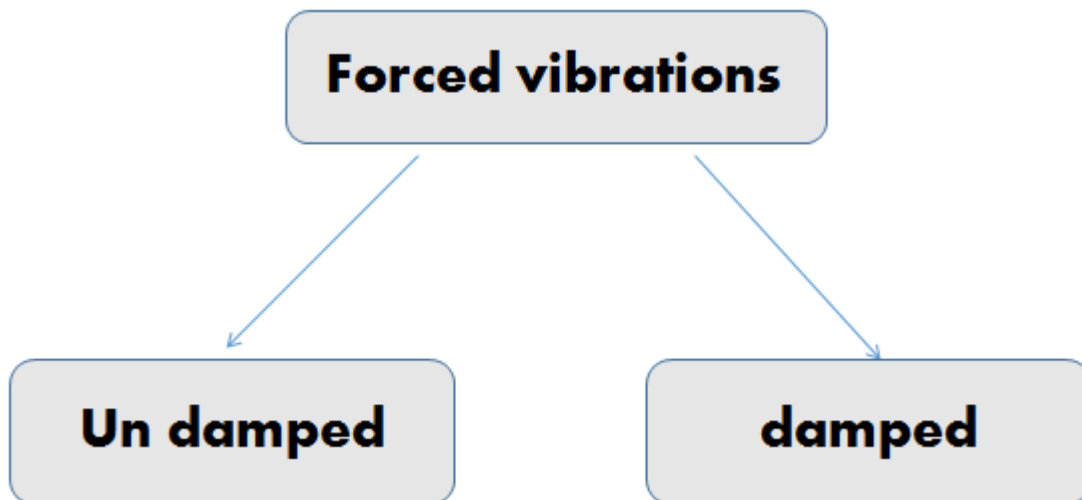


Fig 2.5.3) Displacement vs time curve for system for $U(0)=1$ and $V(0)=1$

2.6 Analysis of forced vibration



2.6.1) Introduction

A harmonic force is $p(t) = p_o \sin \omega t$ or $p_o \cos \omega t$, where p_o is the amplitude or maximum value of the force and its frequency ω is called the exciting frequency or forcing frequency.

Setting $p(t) = p_o \sin \omega t$ gives the differential eqn governing the forced harmonic vibration of the system, which for system without damping specializes to

$$m\ddot{u} + ku = p_o \sin \omega t$$

This eqn is to be solved for the displacement and deformation $u(t)$ subject to initial conditions

$$u = u(0) \quad \dot{u} = \dot{u}(0)$$

Where $u(0)$ and $\dot{u}(0)$ are the displacement and velocity at the time instant force is applied.

2.6.2 Undamped forced vibrations

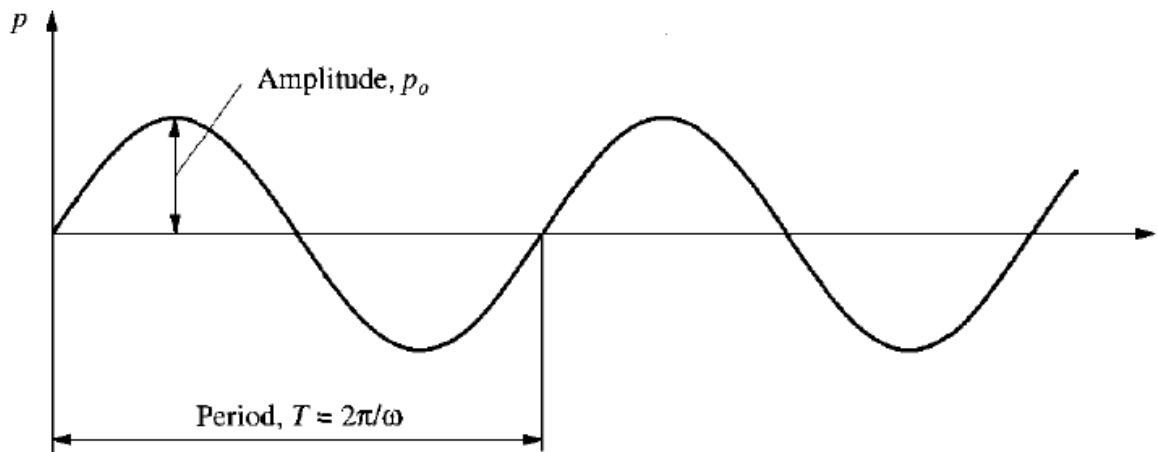
Equation of Motion-

$$m\ddot{u} + ku = p_o \sin \omega t$$

Solution to equation of motion is-

$$u(t) = \underbrace{u(0) \cos \omega_n t + \left[\frac{\dot{u}(0)}{\omega_n} - \frac{p_o}{k} \frac{\omega/\omega_n}{1 - (\omega/\omega_n)^2} \right] \sin \omega_n t}_{\text{transient}}$$

$$+ \underbrace{\frac{p_o}{k} \frac{1}{1 - (\omega/\omega_n)^2} \sin \omega t}_{\text{steady state}}$$



for

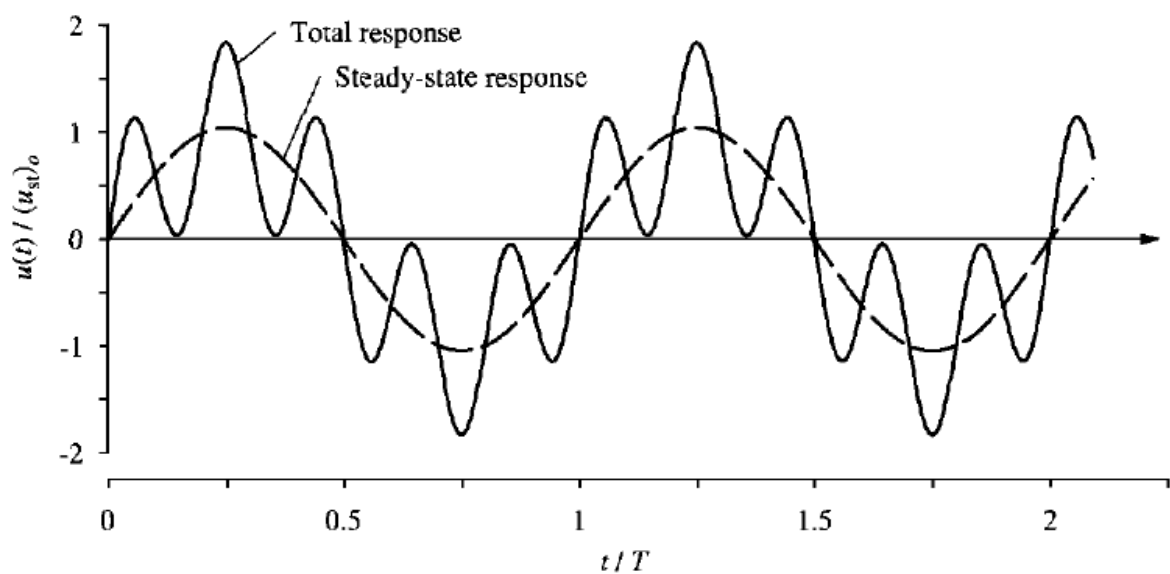


Fig 2.6.1) a) Simple harmonic motion b) response of undamped system to harmonic force ; $\omega/\omega_n = 0.2$, $u(0) = 0$, and $\dot{u}(0) = \omega_n p_0/k$.

2.6.3 Forced vibrations with viscous damping

Equation of motion:
$$m\ddot{u} + c\dot{u} + ku = p_0 \sin \omega t$$

This equation can be solved subject to initial conditions $u = u(0)$ $\dot{u} = \dot{u}(0)$

Solution to equation of motion:

$$u(t) = \underbrace{e^{-\zeta\omega_n t} (A \cos \omega_D t + B \sin \omega_D t)}_{\text{transient}} + \underbrace{C \sin \omega t + D \cos \omega t}_{\text{steady state}}$$

Where, $\omega_D = \omega_n \sqrt{1 - \zeta^2}$.

$$C = \frac{p_0}{k} \frac{1 - (\omega/\omega_n)^2}{[1 - (\omega/\omega_n)^2]^2 + [2\zeta(\omega/\omega_n)]^2}$$

$$D = \frac{p_0}{k} \frac{-2\zeta\omega/\omega_n}{[1 - (\omega/\omega_n)^2]^2 + [2\zeta(\omega/\omega_n)]^2}$$

And,

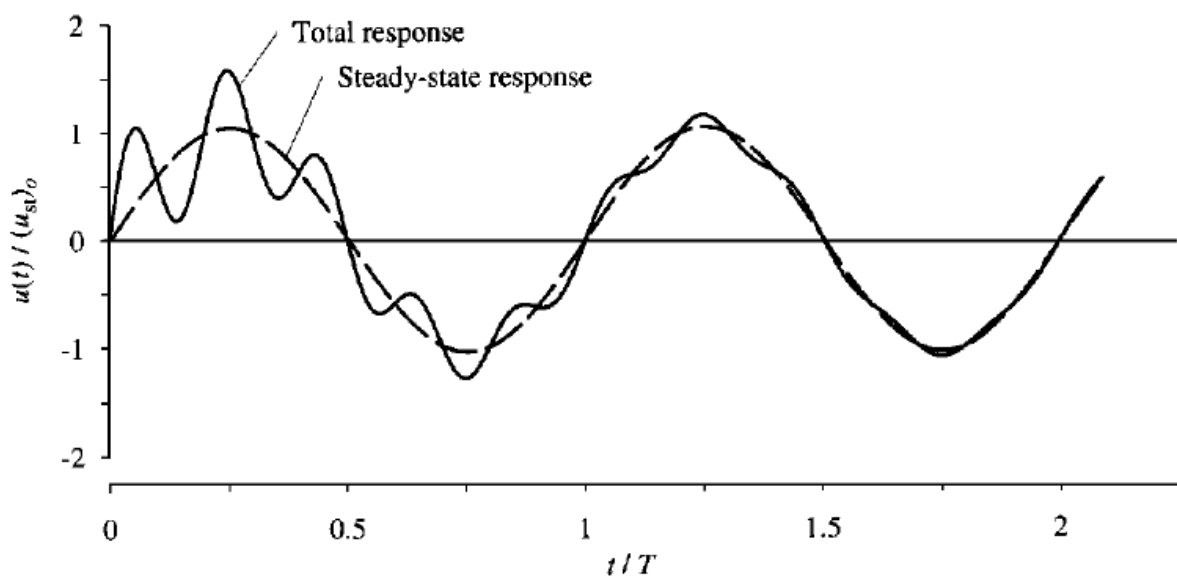


Fig 2.6.2) Response of damped system to harmonic force $\omega/\omega_n = 0.2$, $\zeta = 0.05$, $u(0) = 0$, and $\dot{u}(0) = \omega_n p_0/k$.

2.7 MATLAB coding for a SDoF system for forced vibrations (damped and undamped).

```

clear all
% m = 0.2; % k/in
% k = 120;
% c = 0.27;
m = input('Enter mass of the system in kg : ');
k = input('Enter stiffness of the system in N/m : ');
zeta = input('Enter damping ratio : ');
c = zeta*2*sqrt(k*m);
dt = 0.01; % Time step for evaluating response
Force_type = input('Enter 1 if force is in Excel file, else 0 : ');
if Force_type==1
    Load=xlsread('Load_data.xlsx','Sheet1');
    tt=Load(:,1); ff=Load(:,2);
    neq=size(tt,1);
    t = 0:dt:tt(neq);
    f = interp1(tt,ff,t);
    N = size(t,2); % Number of sampling points
else
    w_bar = input('Enter force frequency in rad/s : ');
    Td = input('Enter duration of force : ');
    i=1;
    for t=0:dt:Td
        Load(i)=sin(w_bar*t);
        f(i)=Load(i);
        i=i+1;
    end
    N = i-1;
    t=0:dt:Td;
end
% zeta = c/(2*sqrt(k*m));
wn = sqrt(k/m); wd=wn*sqrt(1-zeta^2);
Tn = 2*pi/wn;

n=[1:2*N];

```

```
g=dt*exp(-(n-1)*zeta*wn*dt).*sin((n-1)*wd*dt)/(m*wd);
c0=conv(f,g);
% c1=c0*wn^2;
for i=1:2*N
    tt(i)=(i-1)*dt;
    c1(i)=c0(i)*wn^2;
end
subplot(2,1,1);
plot(t,f);
subplot(2,1,2);
plot(tt,c1);
```

Chapter 3

Step Response of a SDOF System

3.1 Introduction

In this example, the force is assumed to be applied instantaneously, but it will be sustained out to infinity. This is essentially an on-function. If a force of this sort is plotted versus time, the force looks like a step up. So then, the behavior of the system under this type of load is considered the step response of the single degree of freedom system.

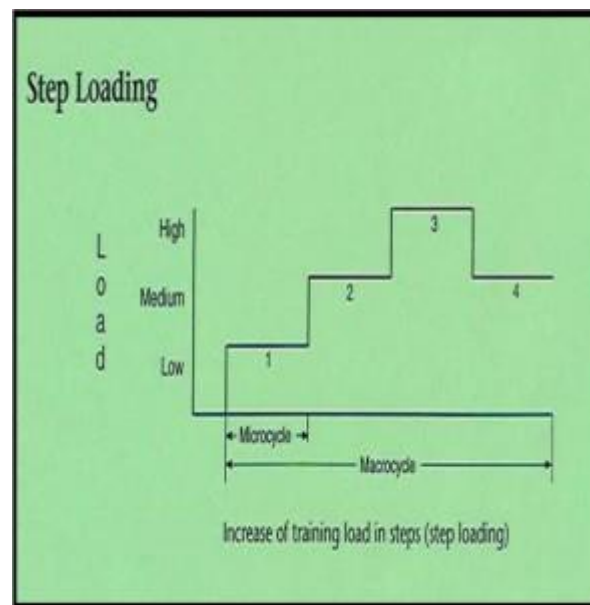


Fig 3.1 step loading

3.2 Step loading

We assume the system is underdamped and will have zero initial conditions. We already know the equation of motion,

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = F(t)/m,$$

Equation of motion: -

$$F(t) = \begin{cases} 0 & \text{if } 0 < t < t_o \\ F_o & \text{if } t \geq t_o \end{cases}$$

Where,

Solution to the equation of motion :

$$x(t) = \frac{F_o}{k} \left\{ 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_o)} \cos [\omega_d(t-t_o) - \phi] \right\}, t \geq t_o,$$

$$\phi = \tan^{-1} \frac{\zeta}{\sqrt{1-\zeta^2}}.$$

Where,

NOTE: It is important to remember that this equation is only valid for the time after the force is applied; the response is zero before application of the force.

3.3 MATLAB coding for SDoF system for step loading

```

3.3.1 %This program calculates the step response
      %of a single degree of freedom system. This
      %version tests three different force magnitudes.
      %The program assumes a system mass of 1.
      %
      zeta=0.05;
      tf=10;
      npts=1000;
      t=linspace(0,tf,npts);
      to=2;
      wn=12;
      k=(wn)^2;
      for i=1:3
      Fm(i,1)=input('Enter a force magnitude. '); % Forcing Fm to be a
      column vector.
      end
      Fint=Fm*ones(1,npts); % Fint is thus 3-by-npts.
      %
      % Now, we perform the logical operation t>to,
      % and make a 3-by-npts matrix of the result.
      %
      qtest=t>to;
      fmult=[qtest;qtest;qtest]; % Three rows of qtest.

```

```

Fo=Fint.*fmult; % The force matrix.
wd=wn*sqrt(1-zeta^2);
phi=atan2(zeta,sqrt(1-zeta^2));
for n=1:3
A=Fo(n,+)/k;
B=Fo(n,)/(k*sqrt(1-zeta^2));
x(n,:)=A-B.*exp(-zeta*wn*t).*cos(wd*t-phi);
end
for l=1:3
subplot(3,1,l)
plot(t,x(l,:))
title(['Response for wn=.',num2str(wn),', Fmax=.', ...
num2str(Fm(l)),', and time=.', num2str(to)]);
ylabel('Response x.')
grid
end
xlabel('Time, seconds.')

```

3.3.2 %This program calculates the step response
 %of a single degree of freedom system. This
 %version tests three different natural frequencies.
 %The program assumes a system mass of 1.
 %
 %
 Fm=5;
 npts=1000;
 zeta=0.05;
 tf=10;
 t=linspace(0,tf,npts);
 to=2;
 qtest=t>to; % Only one force vector here, so we can do this here.
 Fo=Fm*qtest; % Scalar Fm * 1-by-npts vector.
 for i=1:3
 wn(i)=input('Enter a natural frequency. ');
 k(i)=wn(i)^2;
 end
 %
 % This could also be vectorized.
 %
 for n=1:3
 wd=wn(n)*sqrt(1-zeta^2);
 A=Fo/k(n);
 B=Fo/(k(n)*sqrt(1-zeta^2));
 phi=atan2(zeta,sqrt(1-zeta^2));
 x(n,:)=A-B.*exp(-zeta*wn(n)*t).*cos(wd*t-phi);
 end
 for l=1:3
 subplot(3,1,l)
 plot(t,x(l,:))

```

title(['.Response for wn=.,num2str(wn(1)),. Fmax=., ...
num2str(Fm),. and time=., num2str(to)']);
ylabel('.Response x.')
grid
end
xlabel('.Time, seconds.')

```

```

3.3.3 % This program calculates the step response
      % of a single degree of freedom system. This
      % version tests three different natural frequencies.
      % The program assumes a system mass of 1.
      %
      npts=1000;
      Fm=5;
      tf=10;
      t=linspace(0,tf,npts);
      to=2;
      qtest=t>to;
      Fo=Fm*qtest;
      wn=12;
      k=(wn)^2;
      for i=1:3
          zeta(i)=input('.Enter a damping ratio (zeta). ');
      end
      %
      % This could also be vectorized.
      %
      for n=1:3
          wd=wn*sqrt(1-zeta(n)^2);
          A=Fo/k;
          B=Fo/(k*sqrt(1-zeta(n)^2));
          phi=atan2(zeta(n),sqrt(1-zeta(n)^2));
          x(n,:)=A-B.*exp(-zeta(n)*wn*t).*cos(wd*t-phi);
      end
      for l=1:3
          subplot(3,1,l)
          plot(t,x(l,:))
          title(['.Response for wn=.,num2str(wn),. zeta=., ...
num2str(zeta(l)),. and time=., num2str(to)']);
          ylabel('.Response x.')
          grid
          end
          xlabel('.Time, seconds.')

```

3.4 Response of the above source codes

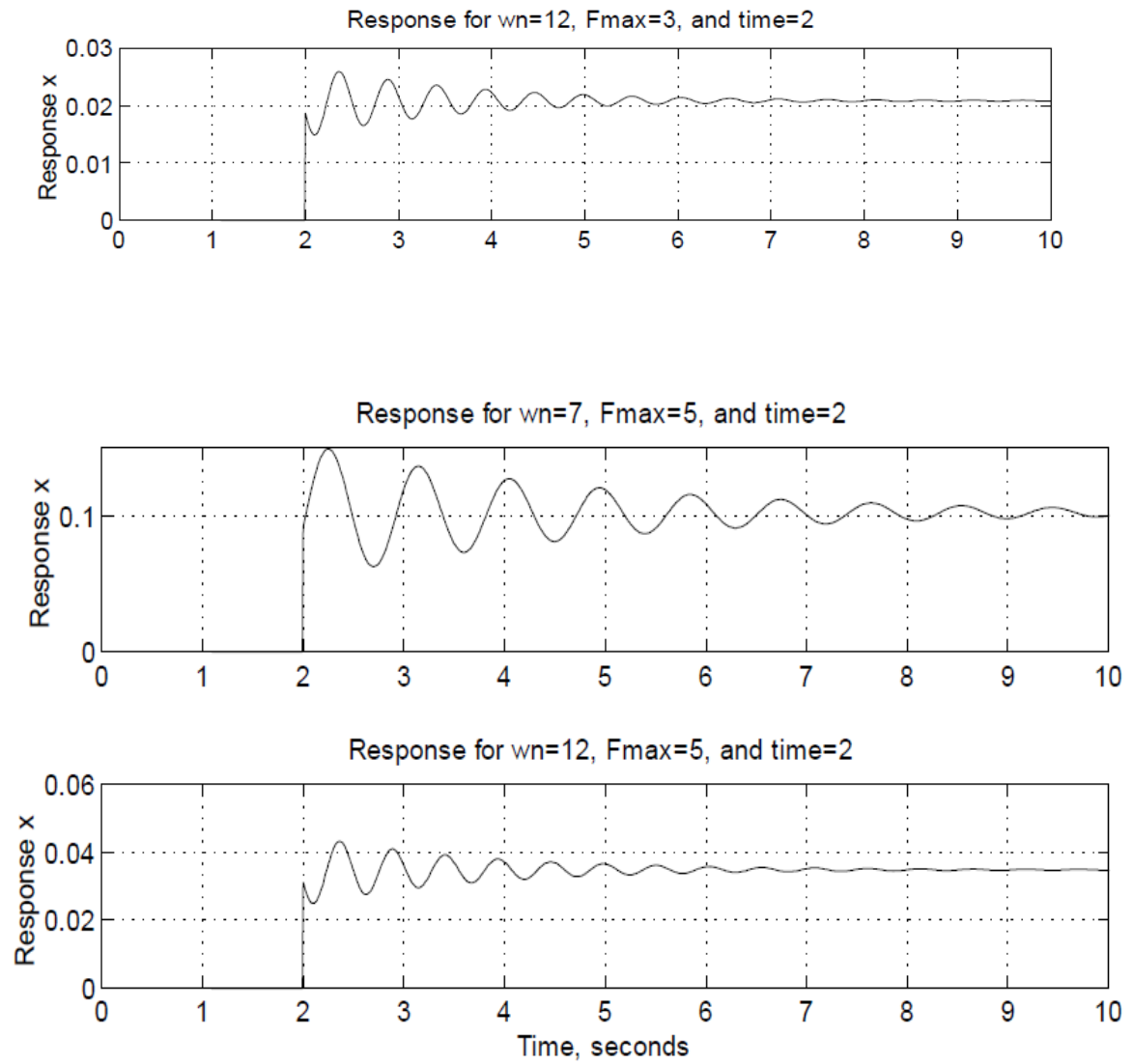


Fig 3.4.1 Step response of single degree of freedom systems having different natural frequencies.

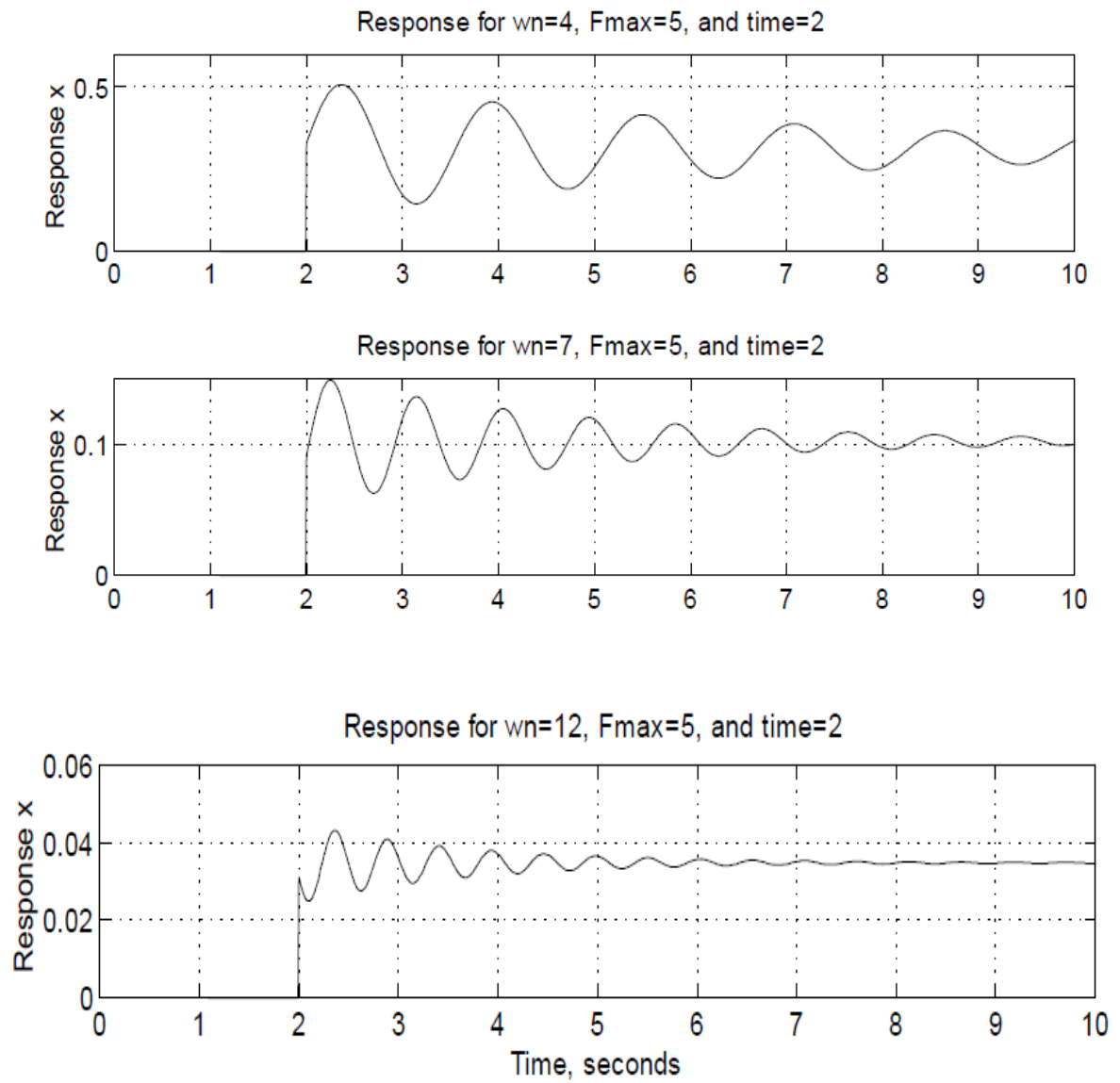


Fig 3.4.2 Step response of single degree of freedom systems having different natural frequencies.

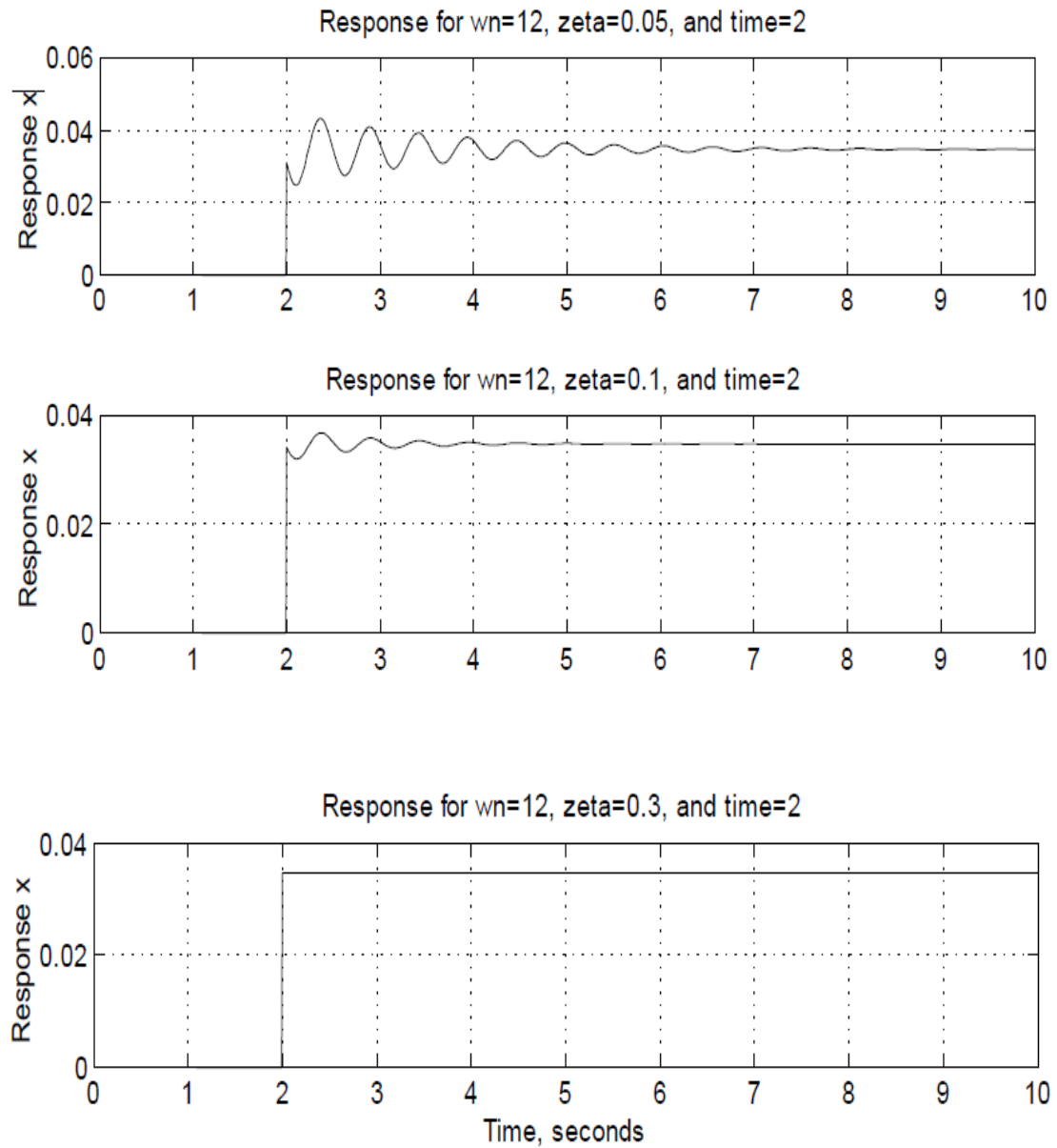


Fig 3.4.3 Step response of a single degree of freedom system to different levels of damping.

Chapter 4

Response of SDOF System to Ramp Input

4.1 Introduction

Another common form of input encountered in real applications is the ramp input. To examine the response of a single degree of freedom system to this sort of input, we must again apply the convolution integral.

4.2 Ramp loading

Assuming that the load is increased uniformly at a rate of f_0 per second and reaches its maximum at time t_d ; then the expression for the external force is:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = F(t)/m,$$

Equation of motion:

$$F(t) = \begin{cases} f_0t & \text{for } 0 \leq t < t_d, \\ f_0t_d & \text{for } t \geq t_d. \end{cases}$$

Where,

Solution of equation of motion:

$$x(t) = \frac{f_0}{m\omega_d} e^{-\zeta\omega_n t} \left(\frac{1}{(\zeta^2\omega_n^2 + \omega_d^2)^2} \right) \{ \omega_d e^{\zeta\omega_n t} [t\zeta^2\omega_n^2 + t\omega_d^2 - 2\zeta\omega_n] + 2\zeta\omega_n\omega_d \cos(\omega_d t) \\ + (\zeta^2\omega_n^2 - \omega_d^2) \sin(\omega_d t) \}, \quad 0 \leq t < t_d,$$

$$x(t) = \frac{f_0}{m\omega_d} e^{-\zeta\omega_n t} \left(\frac{1}{(\zeta^2\omega_n^2 + \omega_d^2)^2} \right) \{ \omega_d e^{\zeta\omega_n t} [t\zeta^2\omega_n^2 + t\omega_d^2 - 2\zeta\omega_n] + 2\zeta\omega_n\omega_d \cos(\omega_d t) \\ + (\zeta^2\omega_n^2 - \omega_d^2) \sin(\omega_d t) \} + \frac{f_0 t_d}{k} - \frac{f_0 t_d}{k\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_d)} \cos(\omega_d(t-t_d) - \phi), \quad t \geq t_d.$$

4.3 MATLAB coding for SDof for ramp loading

```

% This program solves the ramp response of a
% single degree of freedom system. The external
% force is assumed of the form  $F(t)=f_0*t$ , and
% is assumed to last for  $t_e$  seconds, after which
% it levels off at  $F=f_0*t_e$  until infinity.
% Again, a unit system mass is assumed.
npts=1000;
wn=5;
zeta=.05;
wd=wn*sqrt(1-zeta^2);
k=wn^2;
te=4;
for kr=1:3
fo(kr)=input('Enter a force magnitude (fo). ');
end
fo=fo(:); % Force fo to be a column vector.
tf=10;
t=linspace(0,tf,npts);
qtest1=t<te;
qtest2=t>te;
% This time, we need two conditional matrices.
amult1=[qtest1;qtest1;qtest1];
amult2=[qtest2;qtest2;qtest2];
% This constant is used for both parts.
a1=fo/(k*(zeta^2*wn^2+wd^2)^2);
% The solution for  $t < t_e$ .
num1=t*(zeta*wn)^2+t*wd^2-2*zeta*wn;
num2=exp(-zeta*wn*t).*(2*zeta*wd*wn*cos(wd*t)+ ...
((zeta*wn)^2-wd^2)*sin(wd*t));
x1=a1*(num1+num2); % (3-by-1)*(1-by-npts)=(3-by-npts).
% Now, the solution for  $t > t_e$ .
num3=te*(zeta*wn)^2+te*wd^2-2*zeta*wn;
num4=exp(-zeta*wn*te).*(2*zeta*wd*wn*cos(wd*te)+ ...
((zeta*wn)^2-wd^2)*sin(wd*te));
a=fo*te/k;
q=sqrt(1-zeta^2);
phi=atan2(zeta,q);
r=a/q*(1-exp(-zeta*wn*(t-te)).*cos(wd*(t-te)-phi));
s=a1*(num3+num4)*ones(1,npts); % num3 and num4 are scalars, so we
must vectorize this.
x2=r+s;
x=x1.*amult1+x2.*amult2; % This gives a matrix of three 1-by-npts solution
vectors.
for i=1:3
subplot(3,1,i)
plot(t,x(i,:))

```



```

title(['.Response for wn=.',num2str(wn),', fo=.', ...
num2str(fo(i)),',zeta=.',num2str(zeta)])
ylabel('Response x.')
grid
end
xlabel('Time, seconds.')

```

4.4 Response of the above source code

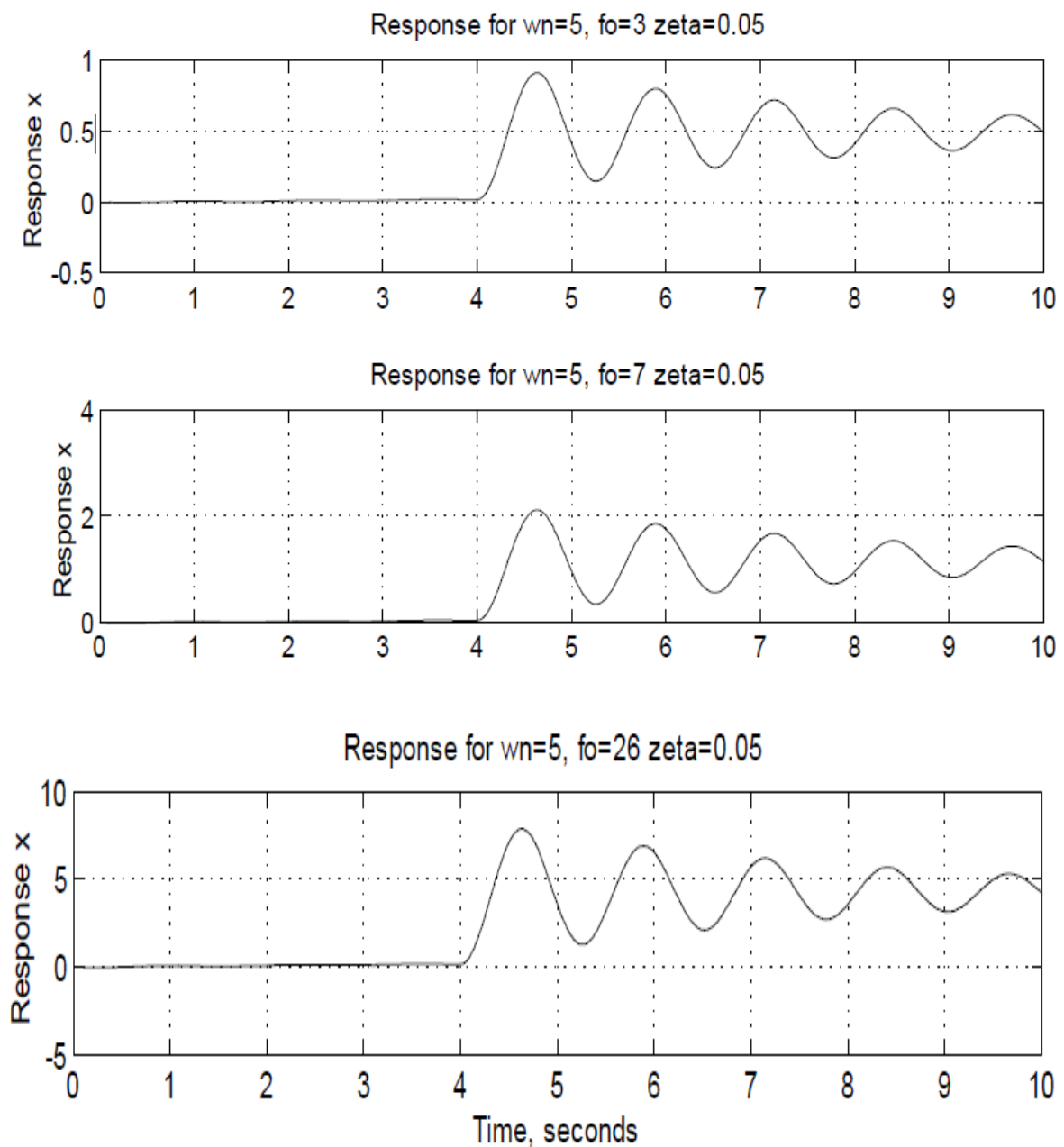


Fig 4.4.1 Response of a single degree of freedom system to different rates of loading.

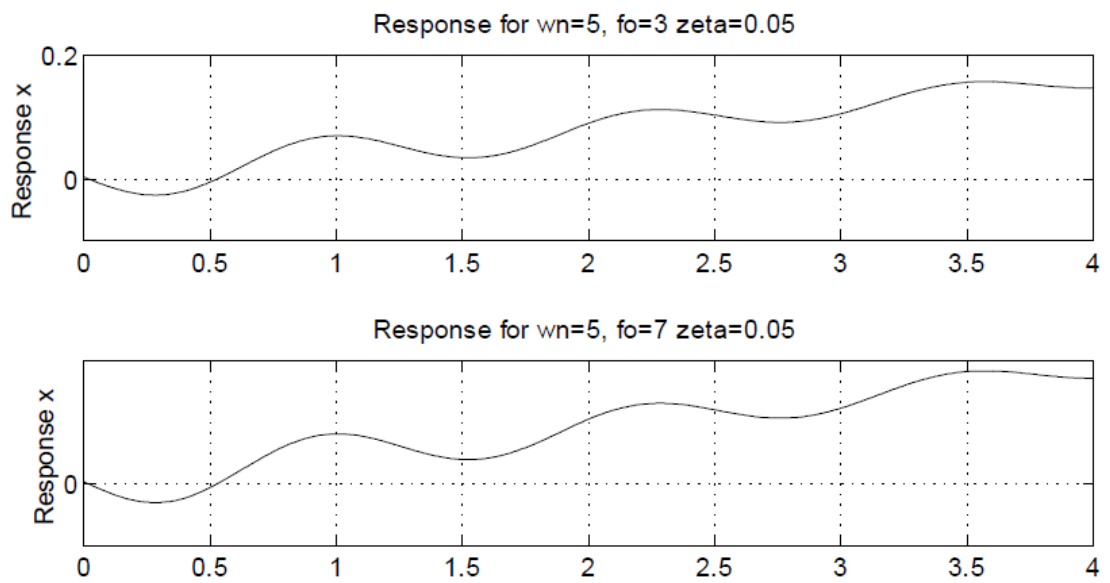


Fig4.4.2 Focusing on the first few seconds of oscillation, showing that the system does oscillate during the transient period.

Chapter 5

Multi degree of freedom systems

5.1 Introduction

We first formulate the equations of motion for the simplest MDoF system, a highly idealized two storey frame subjected to external forces $p_1(t)$ and $p_2(t)$ (fig 5.1)

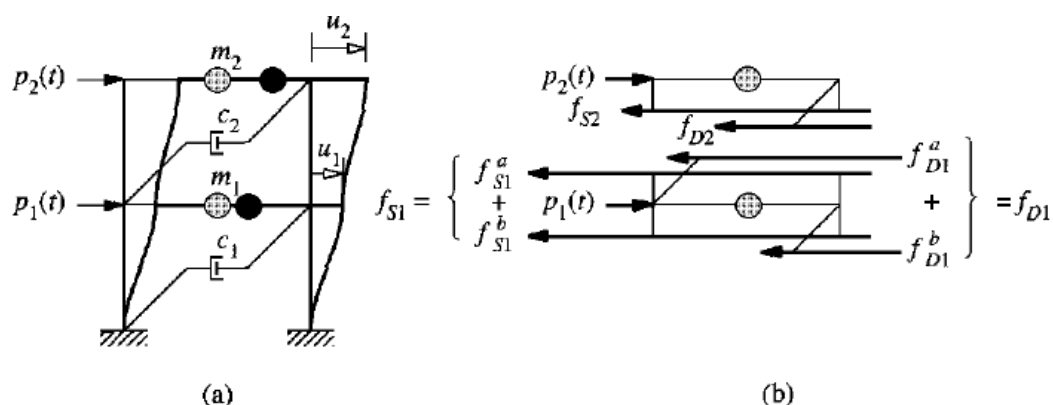


Fig 5.1 a) two-storey frame b) forces acting on the frame

In this idealization the beams and the floor systems are rigid(infinitely stiff) in flexural, and several factors are neglected : axial deformation of the beams and columns; and the effect of axial force on the stiffness of columns. The mass is distributed throughout the building, but we will idealize it as concentrated at the floor levels. This assumption is generally appropriate for multistory buildings because most the building mass is indeed at the floor levels.

5.2 Mass-Spring Damper System

We have introduced the linear two DoF system by idealizing a two- storey frame , however the classic two DoF system is shown in the fig(5.2) , consists of two masses connected by linear springs and linear viscous dampers subjected to external forces $p_1(t)$ and $p_2(t)$. At any instant of time the forces of the two masses are shown in the free body diagram .

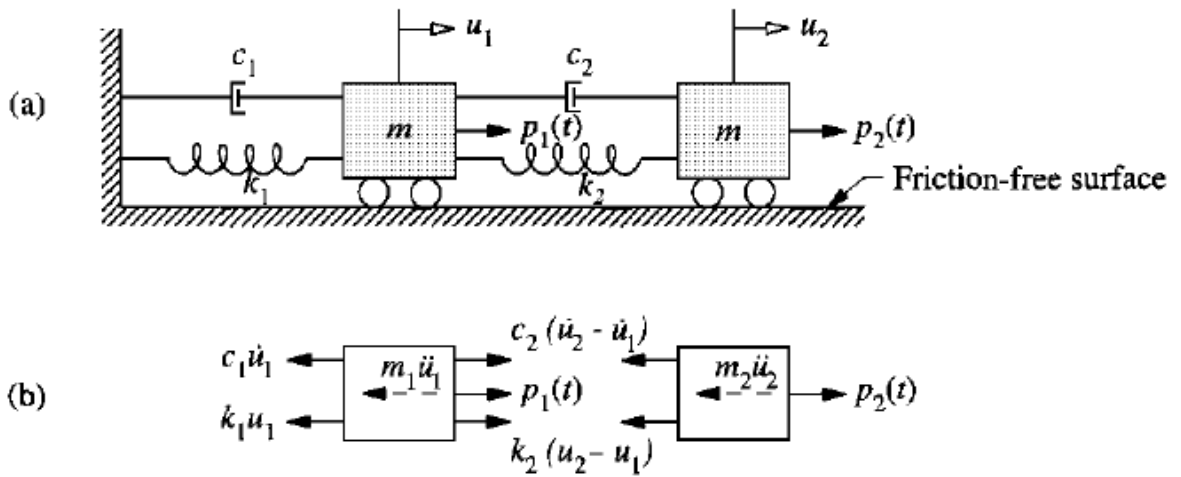


Fig 5.2 a) two degree of freedom system b) free body diagrams.

5.3 Analysis of free vibrations (without damping)

The simplest multi degree of freedom system is a two degree freedom system .

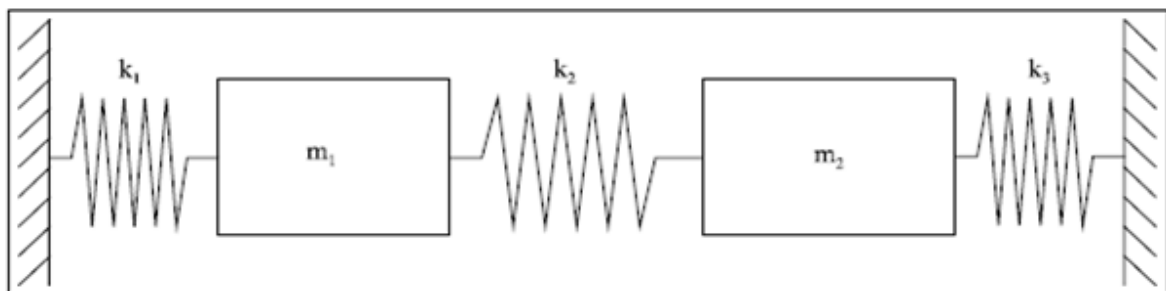


Fig 5.3.1 Typical two degree of freedom free oscillator

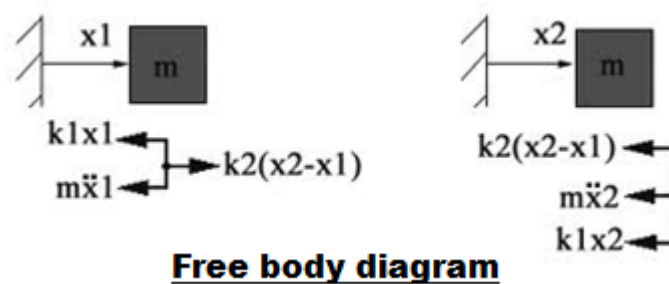


Fig 5.3.2 free body diagram

Applying newton's second law to each mass –

Equation of motion:

$$m_1 \ddot{x}_1 + (k_1 + k_2)x_1 - k_2 x_2 = 0$$

$$m_2 \ddot{x}_2 - k_2 x_1 + (k_2 + k_3)x_2 = 0.$$

Solution of equation of motion:

$$\mathbf{x}(t) = \sum_{i=1}^2 \{u\}_i \left(\{u\}_i^T M \mathbf{x}(0) \cos \omega_i t + \frac{1}{\omega_i} \{u\}_i^T M \dot{\mathbf{x}}(0) \sin \omega_i t \right)$$

5.4 MATLAB coding for MDoF system for free vibration (undamped)

```

%
clear;
M=input('Enter the mass matrix: ');
[n,o]=size(M);
if n~=o then
error('M matrix must be square!');
end
K=input('Enter the stiffness matrix: ');
[n,o]=size(K);
if n~=o then
error('K matrix must be square!');
end
qu=0;
[u,l]=eig(K,M);
% Using .eig.in this way allows us to subtract M*w^2
% from K, instead of I*w^2 (where I is the n by n identity
% matrix).
% The output from .eig.gives unit-length eigenvectors.
% We need to scale them with respect to M.
%
for s=1:n
alfa=sqrt(u(:,s).*M*u(:,s));
u(:,s)=u(:,s)/alfa;
end
x0=input('Enter the initial displacement column vector: ');
xd0=input('Enter the initial velocity column vector: ');
tf=input('Enter the final time: ');
t=0:0.1:tf;
q=tf/0.1;
x=zeros(size(n,q));
% Applying Equation 7.183.

```

```
%
for j=1:n
w(j)=sqrt(l(j,j));
xt=u(:,j)*(u(:,j).*M*x0*cos(w(j).*t)+u(:,j).*M*xd0/...
w(j)*sin(w(j).*t));
x=x+xt;
end
% Plotting the modes in a subplot format.
% Note that, for more than 3 or 4 degrees
% of freedom, the plots will become nearly
% unreadable.
%
for r=1:n
subplot(n,1,r)
plot(t,x(r,:))
xlabel('Time, seconds. ');
ylabel(['.Response x.',num2str(r)]);
end
```

5.5 Response of the above source code

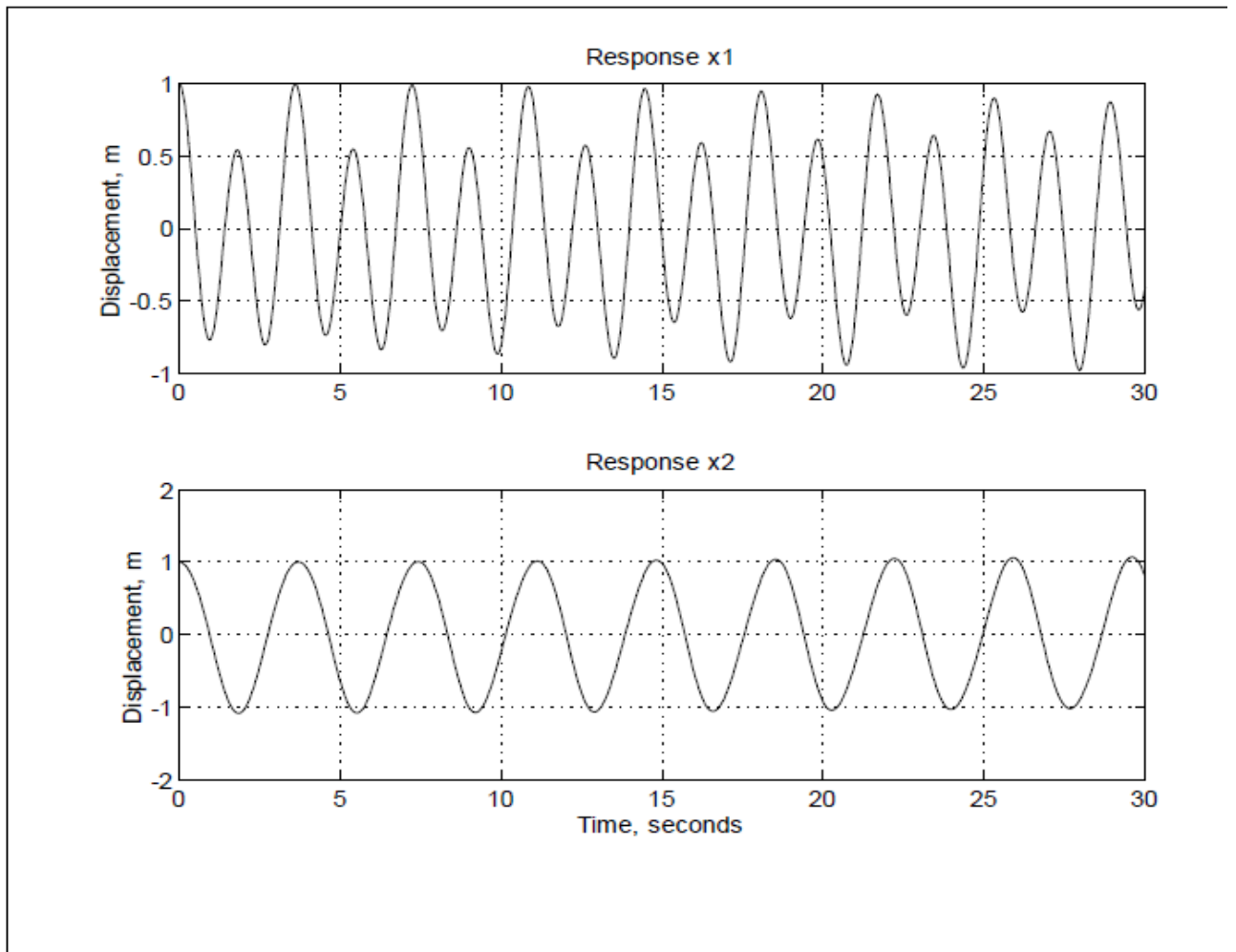


Fig 5.5.1 response to initial vector displacement [1 1]

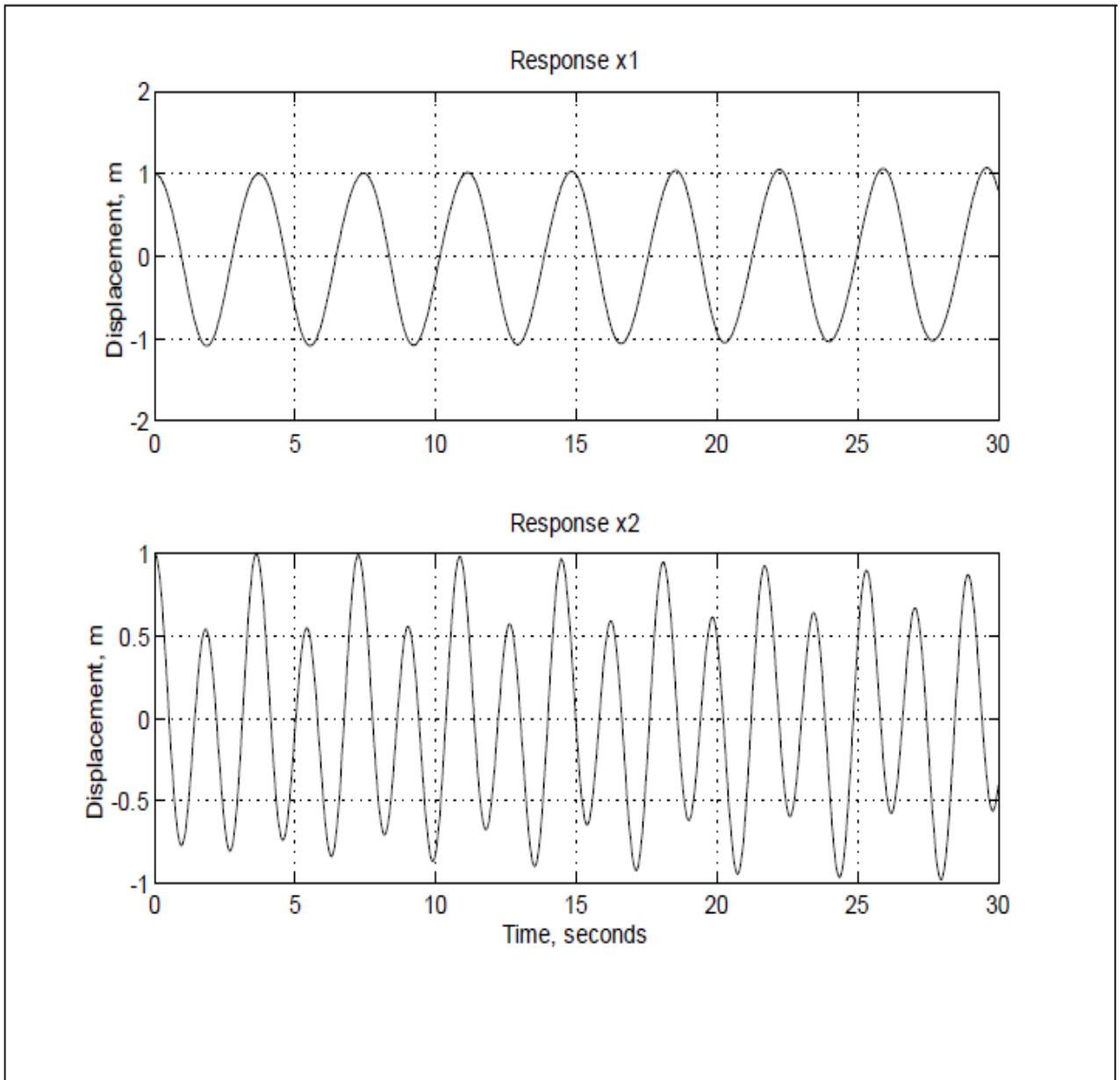


Fig 5.5.2 response after interchanging system masses

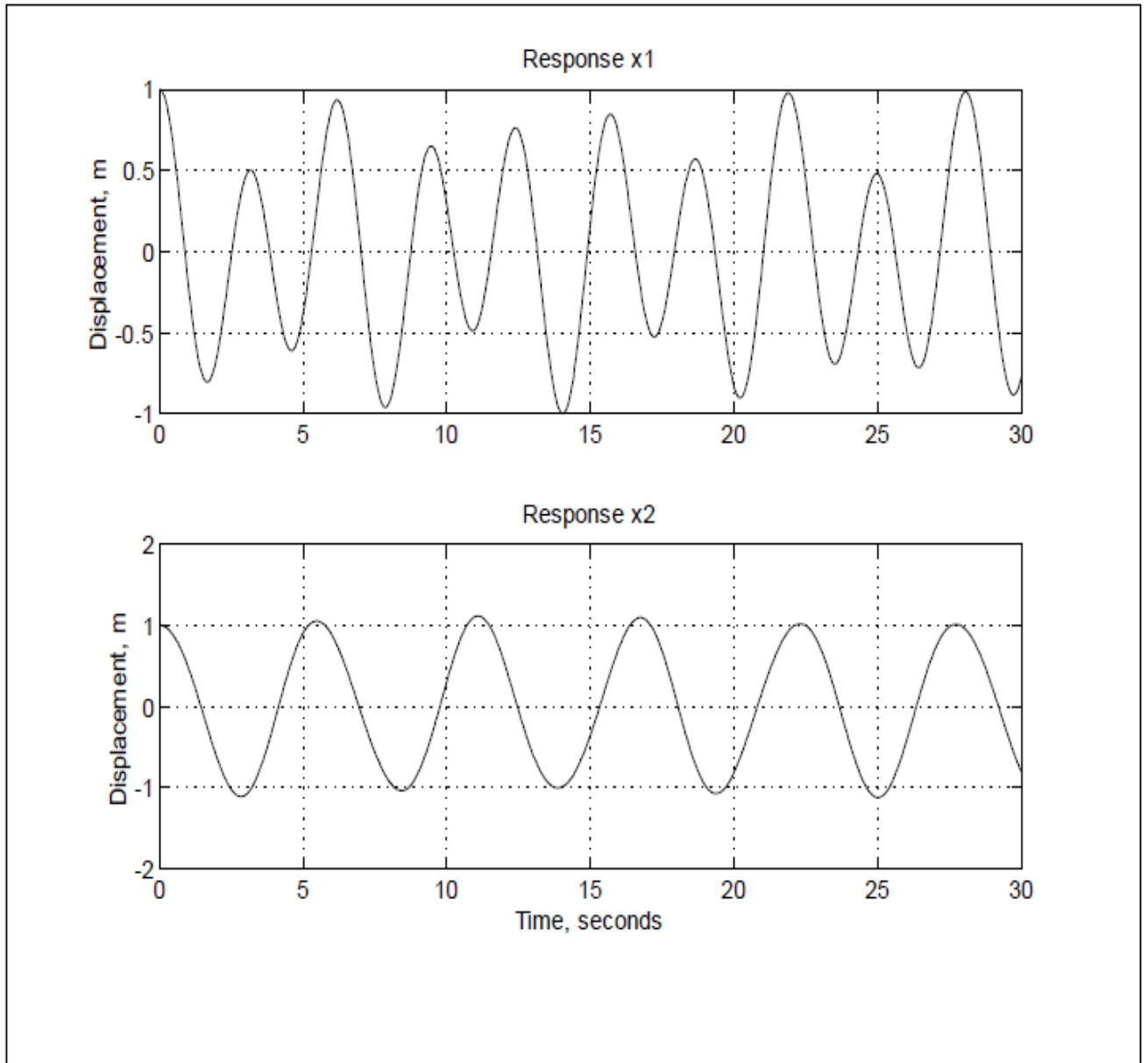


Fig 5.5.3 response with increasing system masses

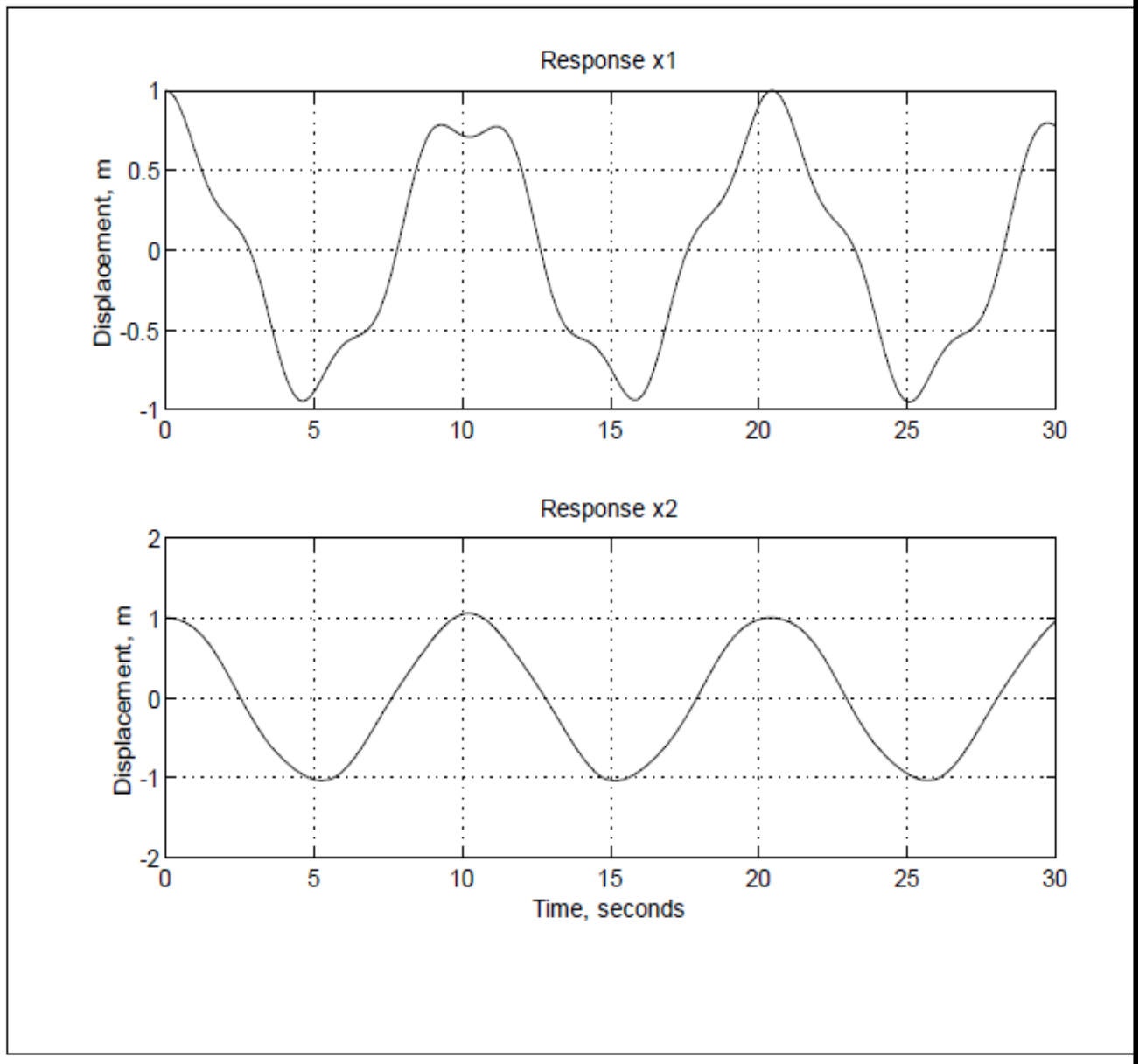


Fig 5.5.4 response with reduced system stiffness

5.6 Analysis of free vibrations(with damping)

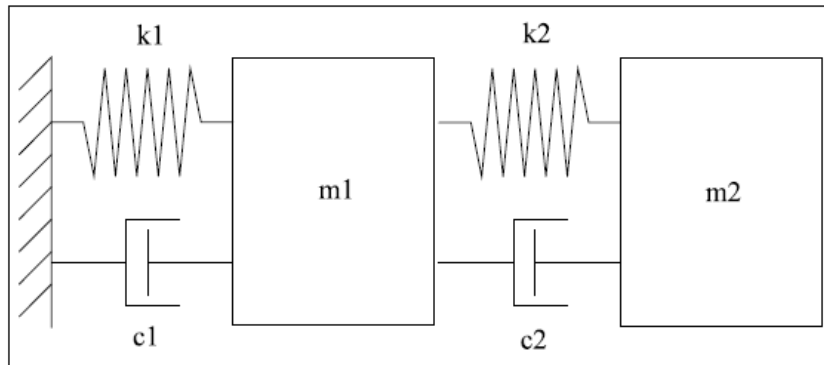


Fig 5.6.1 2-DoF damped oscillator

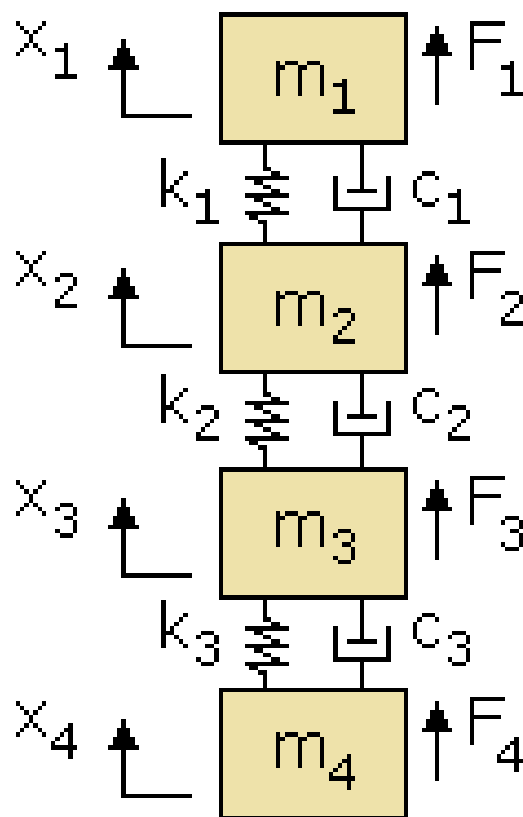


Fig 5.6.2 multi DoF damped oscillator

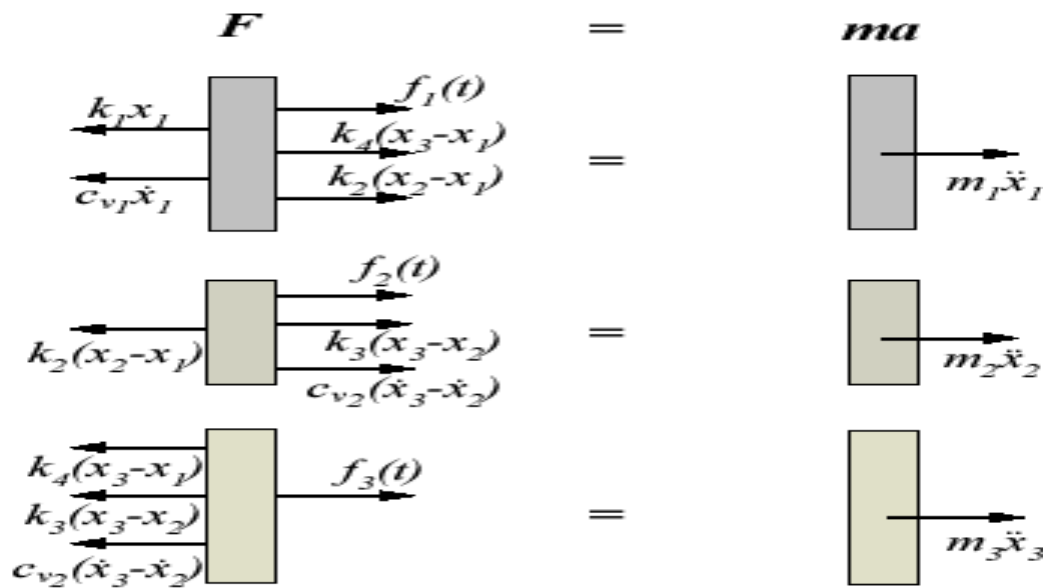


Fig 5.6.3 free body diagram

Equation of motion :

$$m_1 \ddot{x}_1 + (c_1 + c_2) \dot{x}_1 - c_2 \dot{x}_2 + (k_2 + k_1) x_1 - k_2 x_2 = 0$$

$$m_2 \ddot{x}_1 + c_2 (\dot{x}_2 - \dot{x}_1) + k_2 (x_2 - x_1) = 0.$$

Solution to the equation of motion:

$$\ddot{q}_i + 2\zeta_i \omega_i \dot{q}_i + \omega_i^2 q_i = 0, \quad i = 1, 2, \dots,$$

$$q_i(t) = A_i e^{-\zeta_i \omega_i t} \sin(\omega_{di} t + \phi_i),$$

$$\omega_{di} = \omega_i \sqrt{1 - \zeta_i^2}$$

$$A_i = \left[\frac{(\dot{q}_{i0} + \zeta_i \omega_{di} q_{i0})^2 + (q_{i0} \omega_{di})^2}{\omega_{di}^2} \right]^{1/2}$$

$$\phi_i = \tan^{-1} \frac{q_{i0} \omega_{di}}{\dot{q}_{i0} + \zeta_i \omega_{di} q_{i0}},$$

Where:

5.7 MATLAB coding for MDoF of free vibrations(damped)

```

%
clear;
M=input('Enter the mass matrix: ');
[n,o]=size(M);
if n~=o
error('M matrix must be square!');
end
K=input('Enter the stiffness matrix: ');
[n,o]=size(K);
if n~=o
error('K matrix must be square!');
end
% Giving the option to input damping via
% modal damping ratios or through the multiplying
% factors Cm and Ck.
%
fprintf('Press 1 to enter modal damping ratios, ')
qz=input('or anything else to enter Cm and Ck. ');
if qz==1
for iz=1:n
zeta(iz)=input(['Enter the damping for mode ',num2str(iz),'.: ']);
end
else
fprintf('Given that [c]=Cm[M]+Ck[K], ');
Cm=input('Enter the factor Cm: ');
Ck=input('Enter the factor Ck: ');
end
qu=0;
[u,l]=eig(K,M);
% Using .eig.in this way allows us to subtract M*w^2
% from K, instead of I*w^2 (where I is the n by n identity
% matrix).
% The output from .eig.gives unit-length eigenvectors.
% We need to scale them with respect to M.
%
for s=1:n
alfa=sqrt(u(:,s).*M*u(:,s));
u(:,s)=u(:,s)/alfa;
end
x0=input('Enter the initial displacement column vector: ');
xd0=input('Enter the initial velocity column vector: ');
tf=input('Enter the .nal time: ');
t=0:0.1:tf; q=tf/0.1;
x=zeros(size(n,q));
for j=1:n
w(j)=sqrt(l(j,j));
% If modal damping ratios were entered, we already have

```

```
% a zeta vector. If not, we need to calculate from Cm
% and Ck.
%
if qz~=1
zeta(j)=0.5*(Cm/w(j)+Ck*w(j));
end
wd(j)=w(j)*sqrt(1-zeta(j)^2);
xt=u(:,j)*(u(:,j).*M*x0*cos(w(j).*t)/sqrt(1-zeta(j)^2)+u(:,j).*M*xd0/...
w(j)*sin(w(j).*t)/wd(j));
x=x+xt;
end
for i=1:n
x(i,:)=x(i,:).*exp(-zeta(i)*w(i).*t);
end
for r=1:n
subplot(n,1,r)
plot(t,x(r,:))
xlabel('Time, seconds. ');
ylabel(['.Response x.',num2str(r)]);
end
```

5.8 Response of the above source code

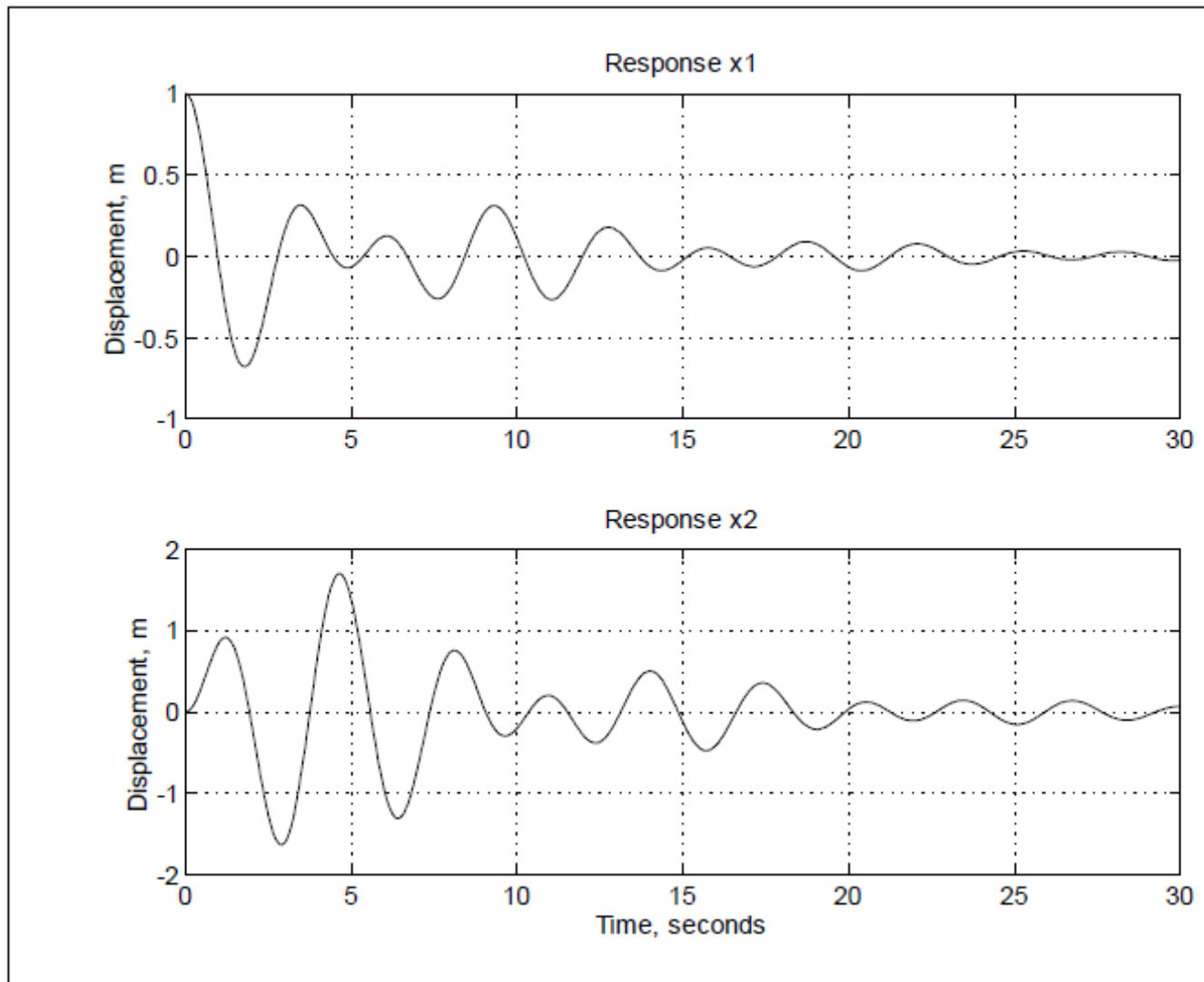


Fig 5.8.1 Damped response of a 2- degree of freedom system

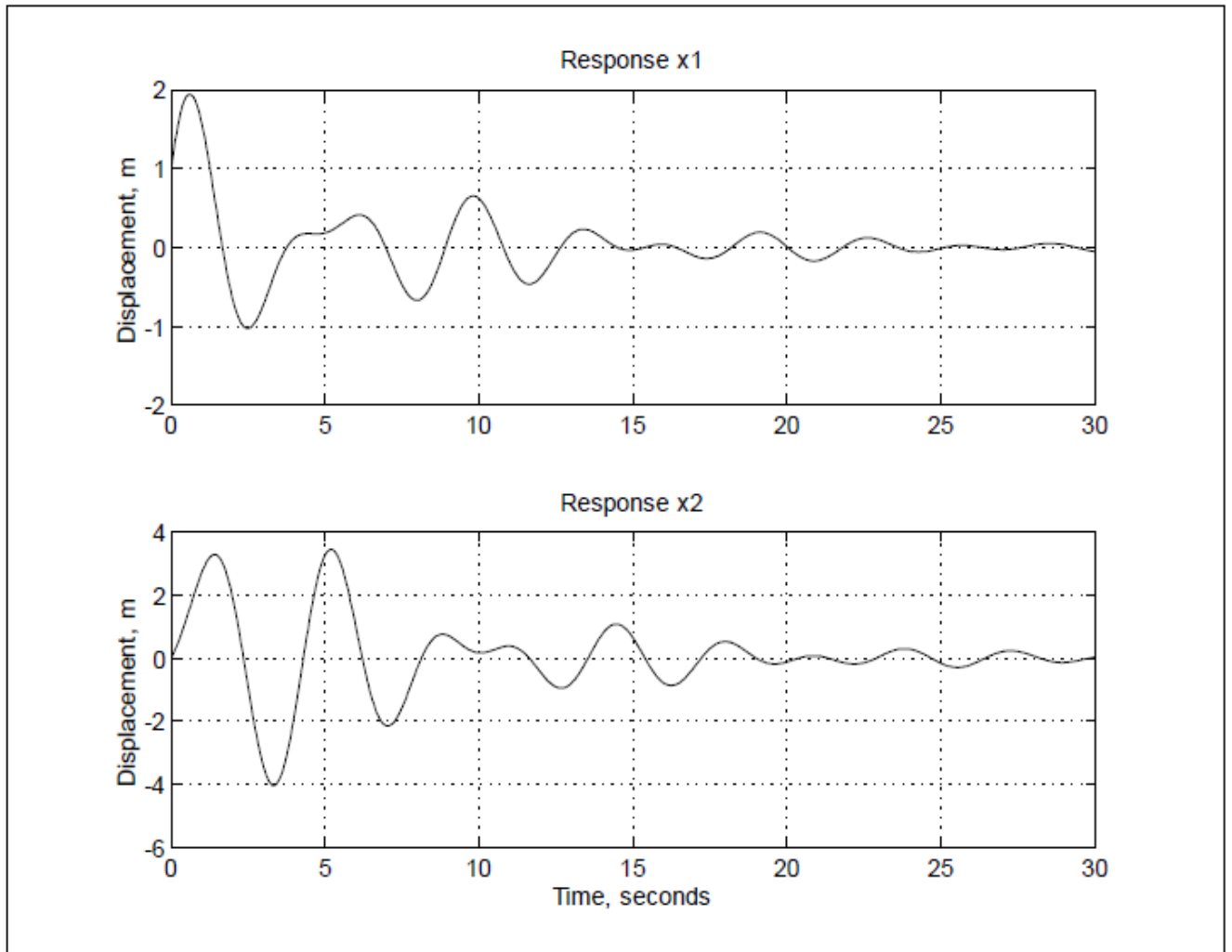


Fig 5.8.2 Response of a damped 2-dof oscillator after adding initial velocity

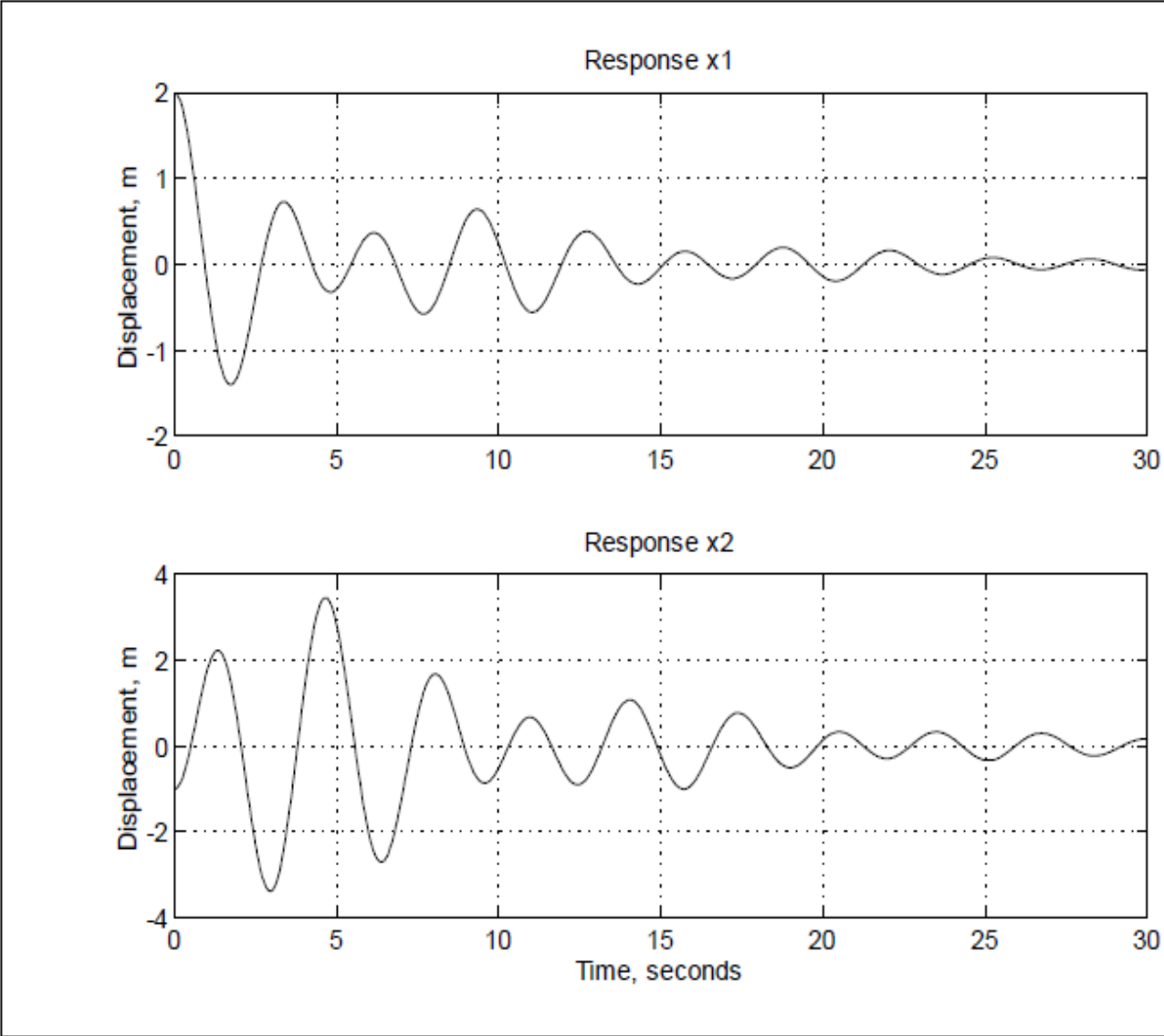


Fig 5.8.3 Response of the same system to a different initial displacement

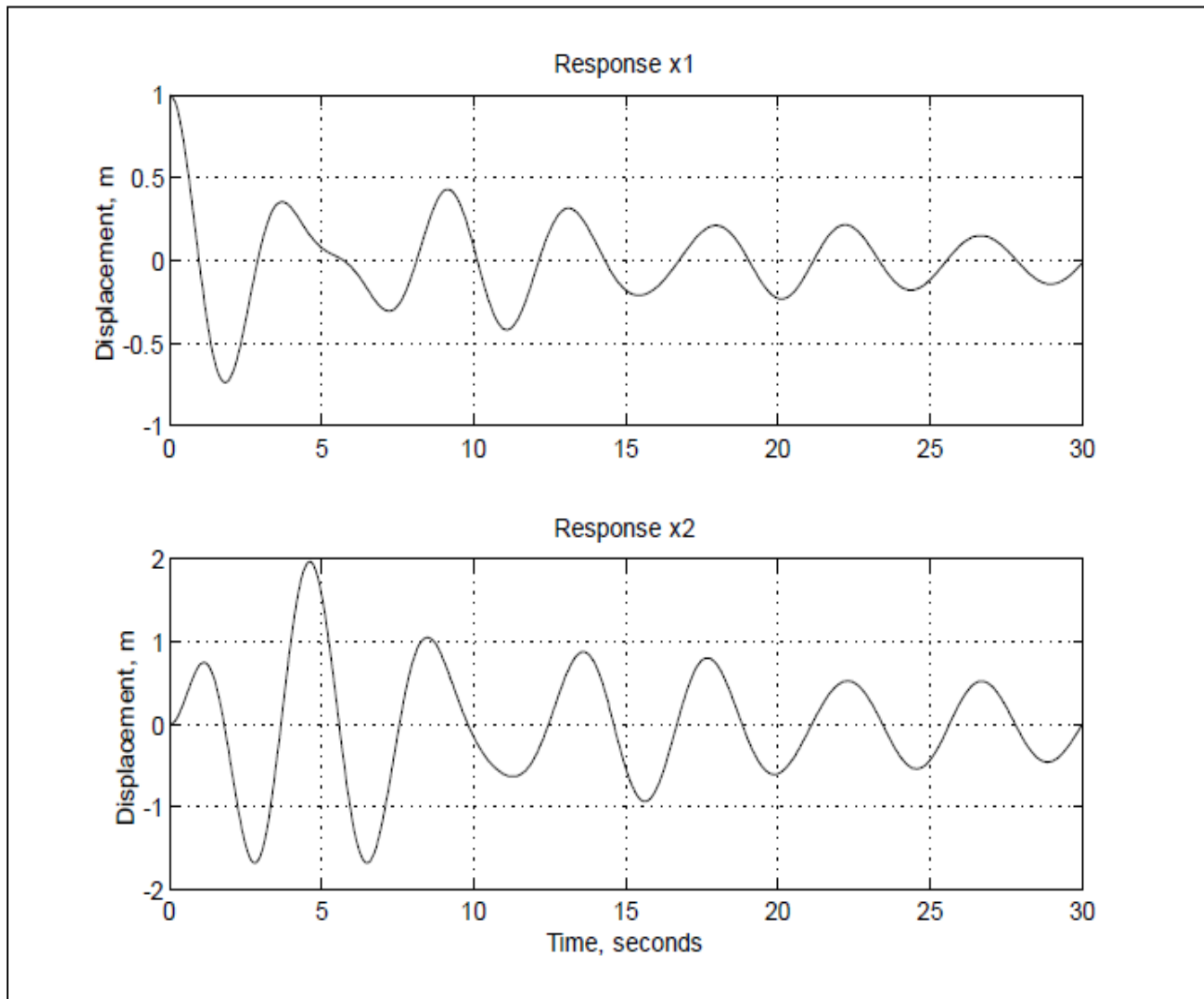


Fig 5.8.4 Response of a 2-dof system with different modal damping ratios

Chapter 6

A universal vibration solver

6.1 MATLAB coding for a universal vibration (free- n degrees)

```

n=input('How many degrees of freedom are present? ');
m=zeros(n,n);
k=zeros(n,n);
s=zeros(n,n);
p=zeros(n,n);
l=zeros(n,n);
ac=zeros(1,n);
phi=zeros(1,n);
m=input('Enter the mass matrix. ');
ko=input('Enter the stiffness matrix. ');
zeta=input('Enter the damping ratio (zeta) matrix. ');
x0=input('Enter the initial displacements. ');
xp0=input('Enter the initial velocities. ');
ti=input('Enter the initial time. ');
tf=input('Enter the final time. ');
t=ti:(tf-ti)/1000:tf;
[b1,b2]=size(t);
x=zeros(n,b2);
r=zeros(n,b2);
a=m^(-1/2);
kt=a*ko*a;
[p,l]=eig(kt);
s=a*p;
si=inv(s);
r0=zeros(n,1);
rp0=zeros(n,1);
r0=si*x0.;
rp0=si*xp0.;
w=sqrt(l);
wd=zeros(size(w));
for k=1:n
wd(k,k)=w(k,k)*sqrt(1-zeta(k)^2);
end
%Note that the same loop control variable can be used for
%every loop, so long as the loops are not nested within
%each other.
for k=1:n
ac(k)=sqrt(wd(k,k)^2*r0(k)^2+(rp0(k)+zeta(k)*w(k,k)*r0(k))^2)/wd(k,k);
phi(k)=atan2(wd(k,k)*r0(k),(rp0(k)+zeta(k)*w(k,k)*r0(k)));
r(k,:)=ac(k)*exp(-zeta(k)*w(k,k)*t).*sin(wd(k,k)*t+phi(k));

```

```
end
x=s*r;
clc
for k=1:n
plot(t,x(k,:))
o=num2str(k);
title(['.Response x.,o])
xlabel(.Time, seconds.)
ylabel(.Displacement.)
grid
figure(gcf)
pause
end
for k=1:n
fprintf(.Natural frequency w% g=% g nn.,k,w(k,k))
end
```

6.2 Response of the above source code

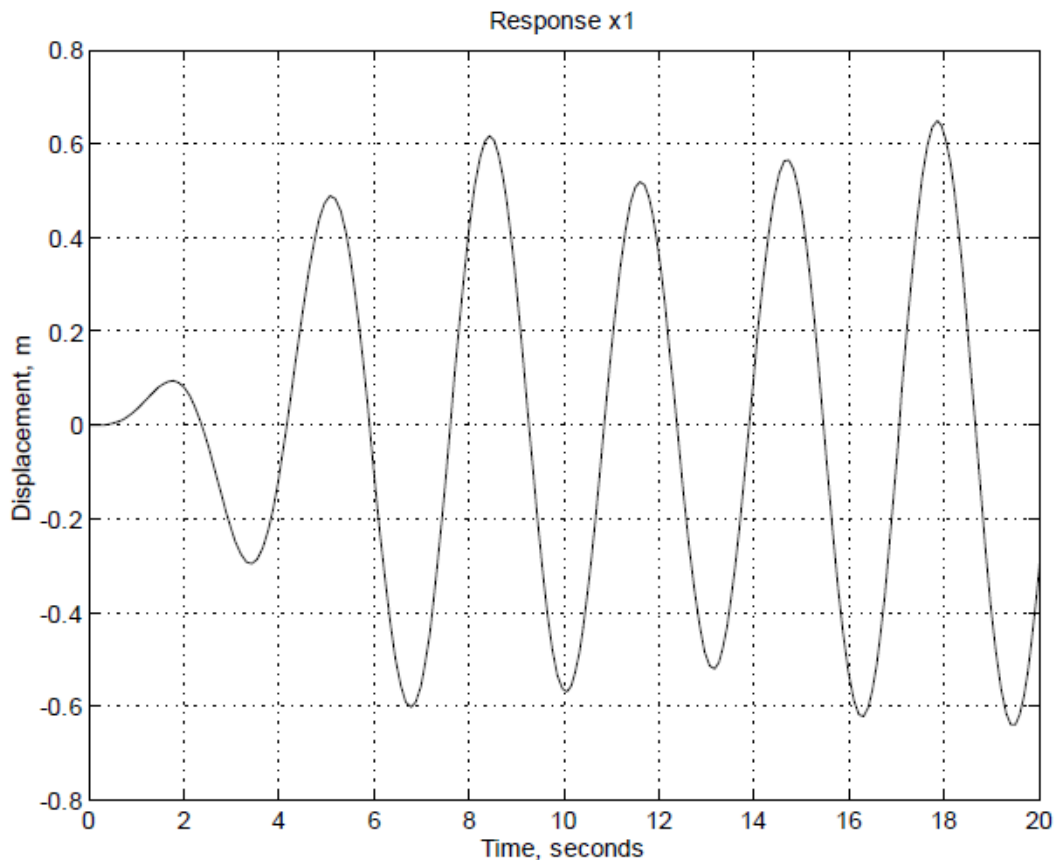


Fig 6.2 displacement vs time curve for n degrees of freedom of free vibration

6.3 MATLAB coding for a universal vibration(forced, n degrees)

```

n=input('How many degrees of freedom are present? ');
x=zeros(1,2*n);
ct=zeros(n,n);
m=input('Enter the mass matrix. ');
aw=2;
while aw==2
fprintf('Press 0 to enter a damping coefficient matrix, or.)
aw=input('press 1 to enter a damping ratio (zeta) matrix. ');
if aw==0
c=input('Enter the damping coefficient matrix. ');
elseif aw==1
zeta=input('Enter the damping ratio matrix. ');
else
fprintf('Please enter 0 or 1..)
aw=2;
end
end
k=input('Enter the stiffness matrix. ');

```

```

a=m^(-0.5);
kt=a*k*a;
[p,q]=eig(kt);
omega=sqrt(q);
if aw==1
for zt=1:n
ct(zt,zt)=zeta(zt)*2*omega(zt,zt);

end
af=a^(-1);
c=af*ct*af;
end
.d=fopen('pdx.m.', 'w+');
fprintf(.d, 'function pdx=pdx(t,x)nn. ');
for i=1:n
fprintf(.d, ['.pdx(.,num2str(i),.)=(-(.)); ');
for j=1:n
if k(i,j)==0
else
fprintf(.d, ['.pdx(.,num2str(k(i,j)),.)*x(.,num2str(j+n),.) ');
end
if c(i,j)==0
else
fprintf(.d, ['.pdx(.,num2str(c(i,j)),.)*x(.,num2str(j),.) ');
end
end
f=input(['Enter F.,num2str(i), (t) in quotes, like .3*cos(2*t).. ']);
fprintf(.d, ['.pdx(.,num2str(f),.)/(.,num2str(m(i,i)),.)nn. ');
end
for i2=1:n
fprintf(.d, ['.pdx(.,num2str(i2+n),.)=x(.,num2str(i2),.)nn. ');
end
status=fclose(.d);
v0=input('Enter the initial velocities as a row vector. ');
x0=input('Enter the initial displacements as a row vector. ');
ti=input('What is your initial time? ');
tf=input('What is your final time? ');
init=[v0,x0];
[t,ny]=ode45('pdx.',ti,tf,init);
for kl=1:n
plot(t,ny(:,kl+n),'.b. ');
title(['Response x.,num2str(kl)']);
xlabel('Time, seconds. ');
ylabel('Displacement, m. ');
grid
figure(gcf)
pause
end
for l=1:n
fprintf('Natural frequency w%g=%g. nn.,l,omega(1,l))

```

end

6.4 Response of the above source code

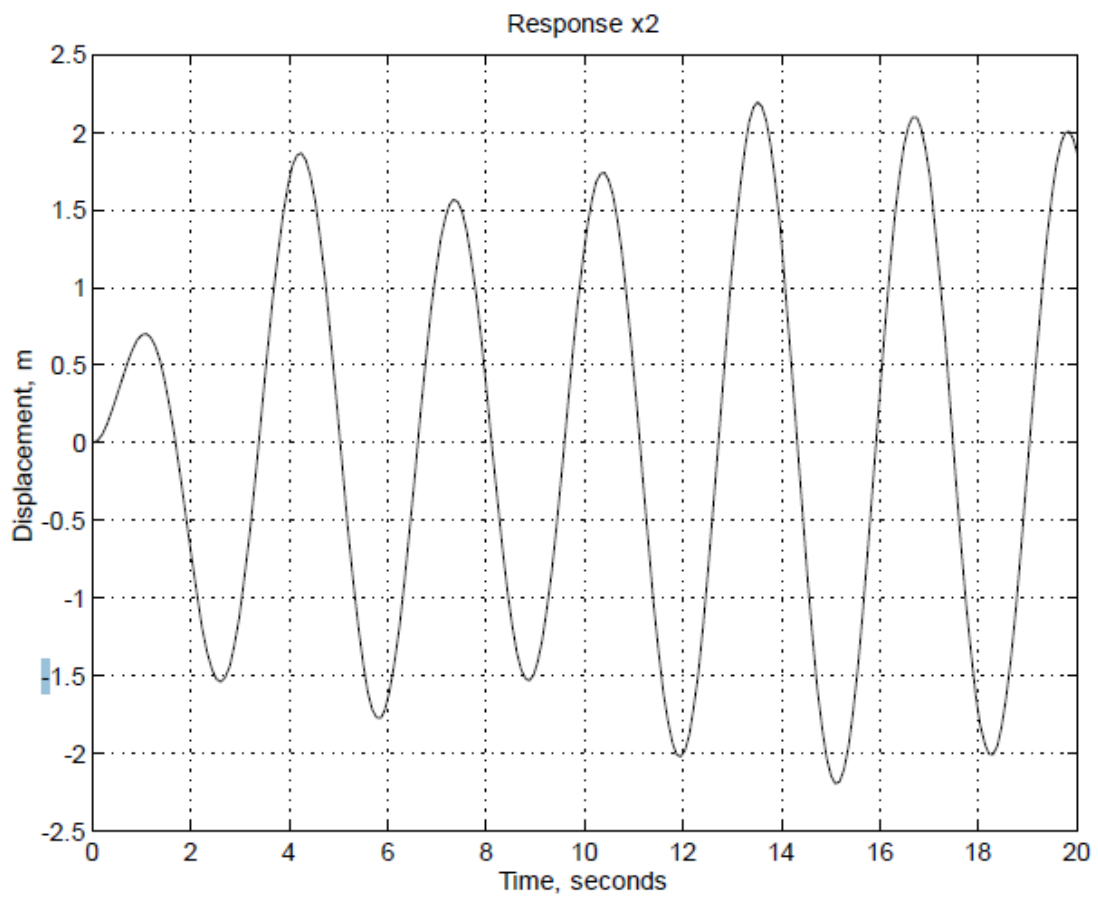


Fig 6.4 displacement vs time curve for n degrees of freedom for forced vibrations

CONCLUSIONS

1. Learned to do dynamic analysis of single degree of freedom systems (free and forced vibrations).
2. Basic concepts of MATLAB.
3. Obtained responses for various types of loadings using Convolution Integral.

REFERENCES

1. Dynamics of Structures by A.K. Chopra.
2. Engineering Mechanics By A.K. Tayal.
3. Introduction to MATLAB for engineering students by David Houcque.
4. MATLAB Programming By Rudrapratap.
5. BASIC STRUCTURAL DYNAMICS by James C. Anderson