

MGM's

Jawaharlal Nehru Engineering College

Laboratory Manual

POWER SYSTEM DYNAMICS AND STABILITY

FOR MASTER OF ENGINEERING (ELECTRICAL POWER SYSTEM) Students

Manual made by

Prof. P.G.LONE

Ó Author JNEC, Aurangabad.

FORWARD

It is my great pleasure to present this laboratory manual for Master of Engineering (Electrical Power System) students for the subject of Power System Dynamics and Stability. Keeping in view the vast coverage required for visualization of concepts of Power System Dynamics and Stability.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual. Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Prof. Dr. S.D.Deshmukh

Principal

LABORATORY MANNUAL CONTENTS

This manual is intended for the Master of Engineering (Electrical Power System) branch in the subject of Power System Dynamics and Stability.

This manual typically contains practical/Lab Sessions related of Power System Dynamics and Stability covering various aspects related the subject to enhanced understanding.

Although, as per the syllabus, only descriptive treatment is prescribed, we have made the efforts to cover various aspects of Power System Dynamics and Stability subject covering types of different topics to make it conceptually understanding.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to cover the theoretical aspects in the books.

Good Luck for your Laboratory Sessions!

Prof.P.G.LONE

SUBJECT INDEX

A. 1. Do's and Don'ts

B. 2. Lab Exercise:

1. Introduction to Matlab Software.
2. Solution of Swing Equation by any one method.
3. Represent power angle curve of Synchronous machines.
4. Representation of a single machine connected to infinite bus.
5. State Space Representation of Synchronous machines.
6. Representation of Equal Area Criteria for different conditions.
7. Introduction to simulink application in Matlab software.
8. Representation of Excitation Systems.
9. Modeling of Synchronous machines.
10. Analysis of Multimachine Systems.

A. Do's and Don'ts in Laboratory

1. Understand the matlab software
2. Understand the all the matlab library.
3. Study the use of all the equipments provided in the library of matlab.
4. Study the simulation demos provided by matlab.
5. Develop the program and simulation as per instructions in the manual.
6. Follow all the rules and regulation of the computer lab.
7. Carefully handle the devices in computer lab.

Instruction for Laboratory Teachers

1. Submission related to whatever lab work has been completed should be done during the next lab session. The printouts related to submission have to be checked on the day of practical.
2. Students should be taught for taking the observations/readings of different measuring instruments under the guidance of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

B. Lab Exercises

Exercise No1: (2 Hours) – 1 Practical

TITLE:-Introduction to Matlab Software.

AIM:- To study the Matlab software .

THEORY:-

Starting Matlab:-After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

- The Command Window
- The Command History
- The Workspace
- The Current Directory
- The Help Browser
- The Start button

When MATLAB is started for the First time, the screen looks like the one that shown in the Figure 1.1.

Getting help:- To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. On the other hand, information about any command is available by typing

```
>> help Command
```

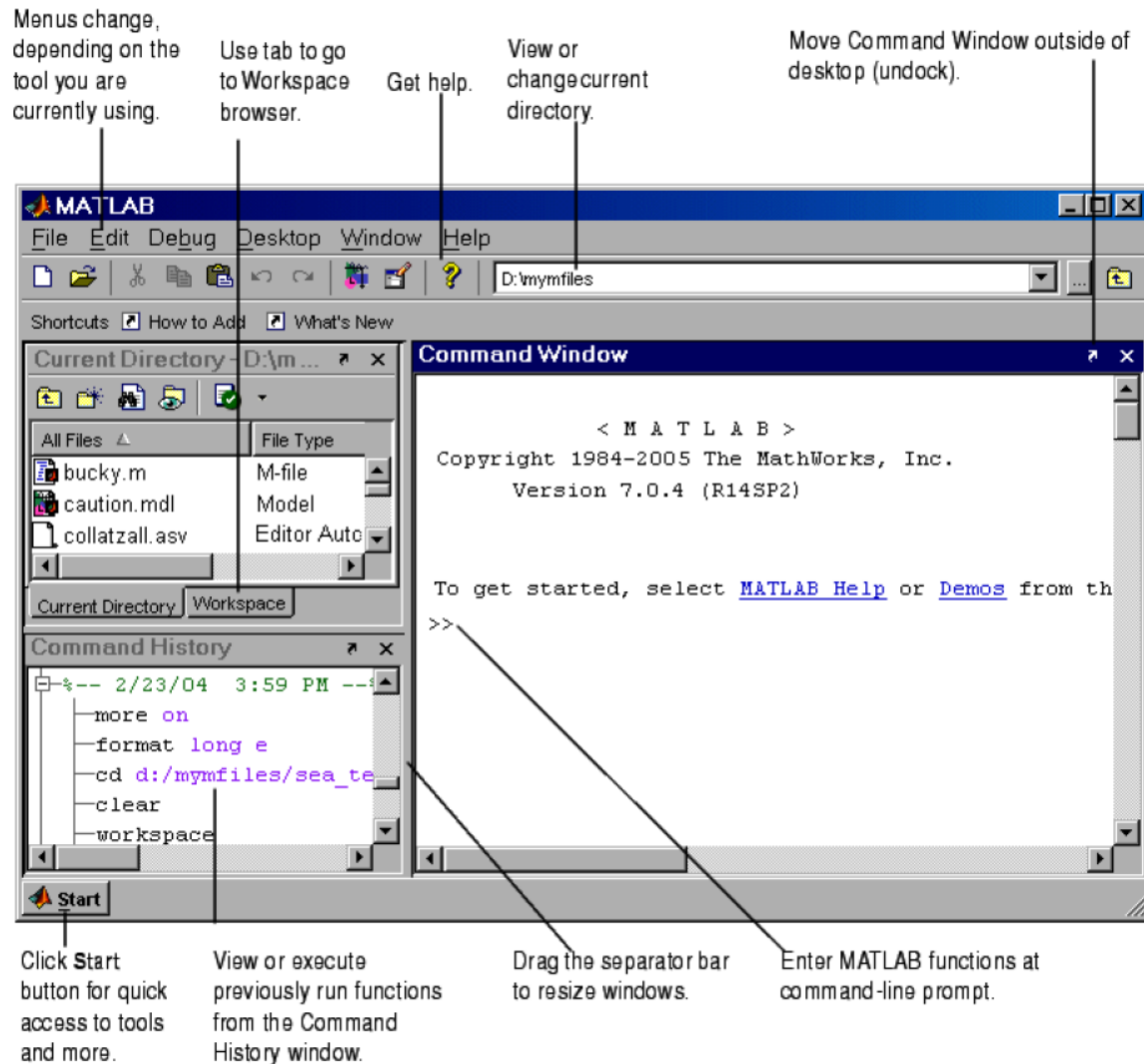


Fig.1.1 Matlab window

Introduction to programming in Matlab

Introduction:-

So far in these lab sessions, all the commands were executed in the Command Window. The problem is that the commands entered in the Command Window cannot be saved and executed again for several times. Therefore, a different way of executing repeatedly commands with MATLAB is :-

1. To create a file with a list of commands
2. Save the file, and
3. Run the file.

If needed, corrections or changes can be made to the commands in the file. The files that are used for this purpose are called script files or scripts for short. This section covers the following topics:

1. M-File Scripts
2. M-File Functions

1. M-File Scripts:-

A script file is an external file that contains a sequence of MATLAB statements. Script files have a filename extension .m and are often called M-files. M-files can be scripts that simply execute a series of MATLAB statements, or they can be functions that can accept arguments and can produce one or more outputs.

Examples

Here is simple script.

Example 1

Consider the system of equations:-

$$x + 2y + 3z = 1$$

$$3x + 3y + 4z = 1$$

$$2x + 3y + 3z = 2$$

Find the solution x to the system of equations.

Solution:

Use the MATLAB editor to create afile: File ! New ! M-file.

Enter the following statements in the file:

$$A = [1 \ 2 \ 3; 3 \ 3 \ 4; 2 \ 3 \ 3];$$

$$b = [1; 1; 2];$$

$$x = A \setminus b$$

Save the file, for example, example1.m.

Run the file, in the command line, by typing:

```
>> example1
```

```
x =  
-0.5000  
1.5000  
-0.5000
```

When execution completes, the variables (A, b, and x) remain in the workspace. To see a listing of them, enter `whos` at the command prompt.

Note: The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The MATLAB editor has numerous menus for tasks such as saving, viewing, and debugging. Because it performs some simple checks and also uses color to differentiate between various elements of codes, this text editor is recommended as the tool of choice for writing and editing M-files.

There is another way to open the editor:

```
>> edit
```

or

```
>> edit filename.m
```

To open filename.m.

CONCLUSION:-

Exercise No2 : (2 Hours) – 1 Practical

TITLE:-Solution of Swing Equation by any one method.

AIM:-To solve swing equation by Point by Point method

- (a) plot swing curve for a sustained fault up to a time of 5 secs.
- (b) plot swing curve if fault is cleared by isolating line in 0.1 seconds.
- (c) Find the critical clearing angle

THEORY:-Swing equation being a non linear equation, numerical methods are use to solve it. Point by Point method is one of the classical solutions to solve swing equation. Below is a solution of swing equation for a machine connected to infinite bus through two parallel lines. Swing equation is drawn for a persisting fault in one of the parallel line and also after fault is cleared. Stability of system is concluded after analyzing the swing curve. clearing angle is calculated for system stability.

Solve the following problem.

Problem: - $f = 50\text{Hz}$ generator 50 MVA supplying 50 MW with inertia constant 'H' = 2.7 MJ/MVA at rated speed. $E = 1.05 \text{ pu}$, $V = 1 \text{ pu}$, $X1 = X2 = 0.4 \text{ pu}$. three phase fault at line 2.

- (a) plot swing curve for a sustained fault up to a time of 5 secs.
- (b) plot swing curve if fault is cleared by isolating line in 0.1 seconds.
- (c) Find the critical clearing angle.

MATLAB PROGRAM:-

MVA base = 50

GIVEN:-

$E = 50$; $V = 1$; $X_d = 0.2$; $X_1 = 0.4$; $X_2 = 0.4$; $H = 2.7$;

prefault condition

$\delta = 0$: $\pi/10$: π ;

$\delta_1 = \delta$;

$\delta_2 = \delta$;

$M = 2.7/(180*50)$; % angular momentum = $H/180*f$

$P_{e0} = (1.05/0.4)*\sin(\delta)$; % Initial power curve

$P_o = 1$; % power output in pu = 50 MW/50 MVA

$\delta_o = \text{asind}(0.4/1.05)$; % initial load angle in degrees // $P_e = (E*V/X) \sin(\delta_o)$

During fault

$P_{e2} = 1.05*\sin(\delta_1)$; % Power curve during fault

Post fault condition

$P_{e3} = (1.05/0.6)*\sin(\delta_2)$; % Power curve after clearing fault

plot(δ , P_{e0});

set(gca,'XTick',0: $\pi/10$: π);

set(gca,'XTickLabel',{'0',' ',' ',' ',' $\pi/2$ ',' ',' ',' π });

title('Power Curve');

xlabel('Load angle');

ylabel('Genpower');

text($(2/3)*\pi$, $(1.05/0.4)*\sin((2/3)*\pi)$,'\leftarrow intial curve','HorizontalAlignment','left');

text($\pi/2$,2.75,'2.625*\sin\delta','HorizontalAlignment','center');

hold all

```

plot(del1,Pe2);
text((2/3)*pi,1.05*sin((2/3)*pi),'\leftarrow during fault','HorizontalAlignment','left');
text(pi/2,1.80,'1.05*sin\delta','HorizontalAlignment','center');
plot(del2,Pe3);
text((2/3)*pi,(1.05/0.6)*sin((2/3)*pi),'\leftarrow fault cleared','HorizontalAlignment','left');
text(pi/2,1.1,'1.75*sin\delta','HorizontalAlignment','center');
hold off
t = 0.05;    % time step preferably 0.05 seconds
t1 = 0:t:0.5;

```

Similarly develop for

(a) sustained fault at $t = 0$

for discontinuity at $t = 0$, we take the average of accelerating power

before and after the fault

at $t = 0^-$, $Pa1 = 0$

at $t = 0^+$. $Pa2 = Pi - Pe2$

at $t = 0$, $Pa = Pa1 + Pa2/2$

(b) Fault cleared in 0.10 seconds ,2nd step ---- 3rd element [1]0 [2] 0.05,[3]0.10

(c) critical clearing angle

```
delo = degtorad(delo);    % initial load angle in rad
```

```
delm = pi - asin(1/1.75); % angle of max swing
```

```
c1 = ((delm-delo)-(1.05*cos(delo))+(1.75*cos(delm)))/(1.75-1.05);
```

```
cclang = acos(c1);          % critical clearing angle in rad
```

```
cclang = radtodeg(cclang); % critical clearing angle in degree
```

```
cclang = int16(cclang);    % converting to integer
```

```
fprintf('\n\n\t\t Critical Clearing angle is %d degree ',cclang);
```

PROGRAM RESULTS:-

CONCLUSION:-

Exercise No3: (2 Hours) – 1 Practical

TITLE:- Represent power angle curve of Synchronous machines.

AIM:-To study and plot power angle curve of synchronous machine.

THEORY:-

- a) Explain in detail power angle curve.
- b) Consider a problem and solve the problem and use the values in the following program.

MATLAB PROGRAM:-

```
V=_; Xd = _; Xdd = _; Xq=_; th = acos_  
S=_;  
Ia = conj(S)/V  
disp('Neglecting Saliency')  
Ed = V + j*Xdd*Ia  
Ed = abs(Ed)  
Pmax = Ed*V/Xdd  
delta = 0:.01:pi;  
P = Pmax*sin(delta);  
disp('Considering the saliency effect')  
d = atan(Xq*abs(Ia)*./(V + Xq* abs(Ia)*_));  
dd=d*180/pi  
E = V*cos(d)+Xd*abs(Ia)*sin(d+th)  
Edq = (Xdd*E-(Xdd - Xd)*V*cos(d))/Xd  
Pmax1 = Edq*V/Xdd, Pmax2=V^2*(Xdd - Xq)/(2*Xdd*Xq)  
Ps =Pmax1 *sin(delta)+ Pmax2*sin(2*delta);  
plot the graph  
plot(delta*180/pi, P, delta*180/pi, Ps), grid
```

PROGRAM RESULTS:-

CONCLUSION:-

Exercise No4: (2 Hours) – 1 Practical

TITLE:-Representation of a single machine connected to infinite bus

AIM:-To study and plot power angle curve of synchronous machine.

THEORY:-

- a) Explain in detail power angle curve.
- b) Consider a problem and solve the problem and use the values in the following program.

MATLAB PROGRAM:-

% program for single m/c connected to infinite bus

clear

t=0

tf=0

tfinal=0.5

tc=0.125

tstep=0.05

M=2.52/(180*50)

i=2

delta=21.64*pi/180

ddelta=0

time(1)=0

ang(1)=21.64

Pm=0.9

Pmaxbf=2.44

Pmaxdf=0.88

Pmaxaf=2.00

while t<tfinal,

 if (t==tf)

 Paminus=0.9-Pmaxbf*sin(delta)

 Paplus=0.9-Pmaxdf*sin(delta)

 Pavg=(Paminus+Paplus)/2

 Pa=Pavg

 end

if (t==tc),

similar develop as above

end

if (t>tc)

similar develop as above

end

t,Pa

ddelta=ddelta+(tstep*tstep*Pa/M)

delta=(delta*180/pi+ddelta)*pi/180

deltadeg=delta*180/pi

t=t+tstep

time(i)=t

ang(i)=deltadeg

i=i+1

end

plot the graph

PROGRAM RESULTS:-

CONCLUSION:-

Exercise No5: (2 Hours) – 1 Practical

TITLE:-State Space Representation of Synchronous machines.

AIM:-To study state space representation of synchronous machine.

THEORY:-

- a) Study and develop state space model of synchronous machine.
- b) Consider a problem and solve the problem and use the values in the following program.

MATLAB PROGRAM:-

```
E = _; V = _; H = _; X = _; Pm = _; D = _; f0 = _;
Pmax = E*V/X, d0 = asin(Pm/Pmax)           % Max. power
Ps = Pmax*cos(d0)           % Synchronizing power coefficient
wn = sqrt(pi*60/H*Ps)      % Undamped frequency of oscillation
z = D/2*sqrt(pi*60/(H*Ps))           % Damping ratio
wd = wn*sqrt(1-z^2), fd = wd/(2*pi) % Damped frequency oscill.
tau = 1/(z*wn)              % Time constant
th = acos(z)                % Phase angle theta
Dd0 = 10*pi/180;           % Initial angle in radian
t = 0:.01:3;
Dd = Dd0/sqrt(1-z^2)*exp(-z*wn*t).*sin(wd*t + th);
d = (d0+Dd)*180/pi;        % Load angle in degree
Dw = -wn*Dd0/sqrt(1-z^2)*exp(-z*wn*t).*sin(wd*t);
f = f0 + Dw/(2*pi);       % Frequency in Hz

plot the graph
A = [_ _; -wn^2 -2*z*wn]; % wn, z and t are defined earlier
B = [_; _];              % Column B zero-input
C = [_ _; _ _]; % Unity matrix defining output y as x1 and x2
D = [_; _];
Dx0 = [Dd0; 0]; % Zero initial cond., Dd0 is defined earlier
[y,x]= initial(A, B, C, D, Dx0, t);
Dd = x(:, 1); Dw = x(:, 2); % State variables x1 and x2
d = (d0 + Dd)*180/pi;      % Load angle in degree
f = f0 + Dw/(2*pi);       % Frequency in Hz
```

plot the graph

PROGRAM RESULTS:-

CONCLUSION:-

Exercise No6: (2 Hours) – 1 Practical

TITLE:-Representation of Equal Area Criteria for different conditions.

AIM:-To study state space representation of synchronous machine.

THEORY:-

This program obtains the power angle curve for a one-machine system during normal operation. Using equal area criterion the maximum input power that can be suddenly applied for the machine to remain critically stable is obtained.

- a) Explain in detail equal area criterion
- b) Consider a problem and solve the problem and use the values in the following program.

MATLAB PROGRAM:-

```
function eacpower(P0, E, V, X)
if exist('P0')~=1
P0 = input('Generator initial power in p.u. P0 = '); else, end
if exist('E')~=1
E = input('Generator e.m.f. in p.u. E = '); else, end
if exist('V')~=1
V = input('Infinite bus-bar voltage in p.u. V = '); else, end
if exist('X')~=1
X = input('Reactance between internal emf and infinite bus in p.u. X = '); else, end
Pemax= E*V/X;
if P0 >= Pemax
    fprintf('\nP0 must be less than the peak electrical power Pemax = %5.3f p.u. \n', Pemax)
    fprintf('Try again. \n\n')
return, end
d0=asin(P0/Pemax);
delta = 0:.01:pi;
Pe = Pemax*sin(delta);
dmax=pi;
Ddmax=1;
while abs(Ddmax) > 0.00001
Df = cos(d0) - (sin(dmax)*(dmax-d0)+cos(dmax));
J=cos(dmax)*(dmax-d0);
Ddmax=Df/J;
dmax=dmax+Ddmax;
```



```

end
dc=pi-dmax;
Pm2=Pemax*sin(dc);
Pmx =[0 pi-d0]*180/pi; Pmy=[P0 P0];
Pm2x=[0 dmax]*180/pi; Pm2y=[Pm2 Pm2];
x0=[d0 d0]*180/pi; y0=[0 Pm2]; xc=[dc dc]*180/pi; yc=[0 Pemax*sin(dc)];
xm=[dmax dmax]*180/pi; ym=[0 Pemax*sin(dmax)];
d0=d0*180/pi; dmax=dmax*180/pi; dc=dc*180/pi;
x=(d0:.1:dc);
y=Pemax*sin(x*pi/180);
% y1=Pe2max*sin(d0*pi/180);
% y2=Pe2max*sin(dc*pi/180);
x=[d0 x dc];
y=[Pm2 y Pm2];
xx=dc:.1:dmax;
h=Pemax*sin(xx*pi/180);
xx=[dc xx dmax];
hh=[Pm2 h Pm2];
delta=delta*180/pi;
%clc
use fprintf option to enter initial values
fill(x,y,'m')
hold;
fill(xx,hh,'c')
plot the graph

```

PROGRAM RESULTS:-

CONCLUSION:-

Exercise No7 : (2 Hours) – 1 Practical

TITLE:-Introduction to simulink application in matlab software .

AIM:-To study simulink application in matlab software

THEORY:-

- 1) To start Open MATLAB and select the Simulink icon in the Toolbar.
- 2) Then the following window appears (Simulink Library Browser)

The Simulink Library Browser is the library as shown in figure7.1 where you find all the blocks you may use in Simulink. Simulink software includes an extensive library of functions commonly used in modeling a system.

These include:

- Continuous and discrete dynamics blocks, such as Integration, Transfer functions, Transport
- Delay, etc.
- Math blocks, such as Sum, Product, Add, etc
- Sources, such as Ramp, Random Generator, Step, etc

- 3) Click the New icon on the Toolbar in order to create a new Simulink model

The window as shown in figure 7.2 appears

You may now drag the blocks you want to use from the Simulink Library Browser to the model surface.

Solve a simple model in this practical.

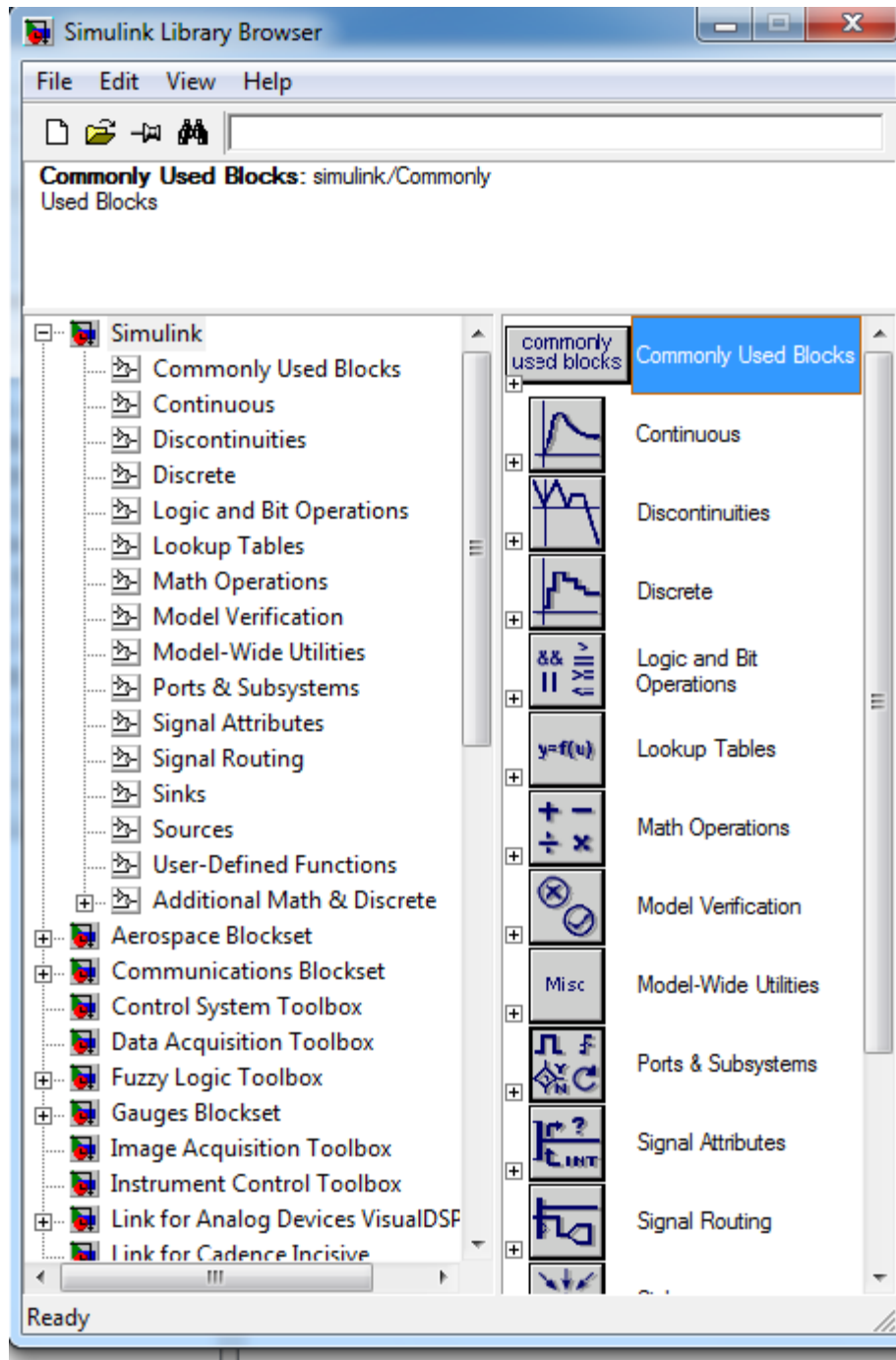


Figure 7.1:- Simulink library window

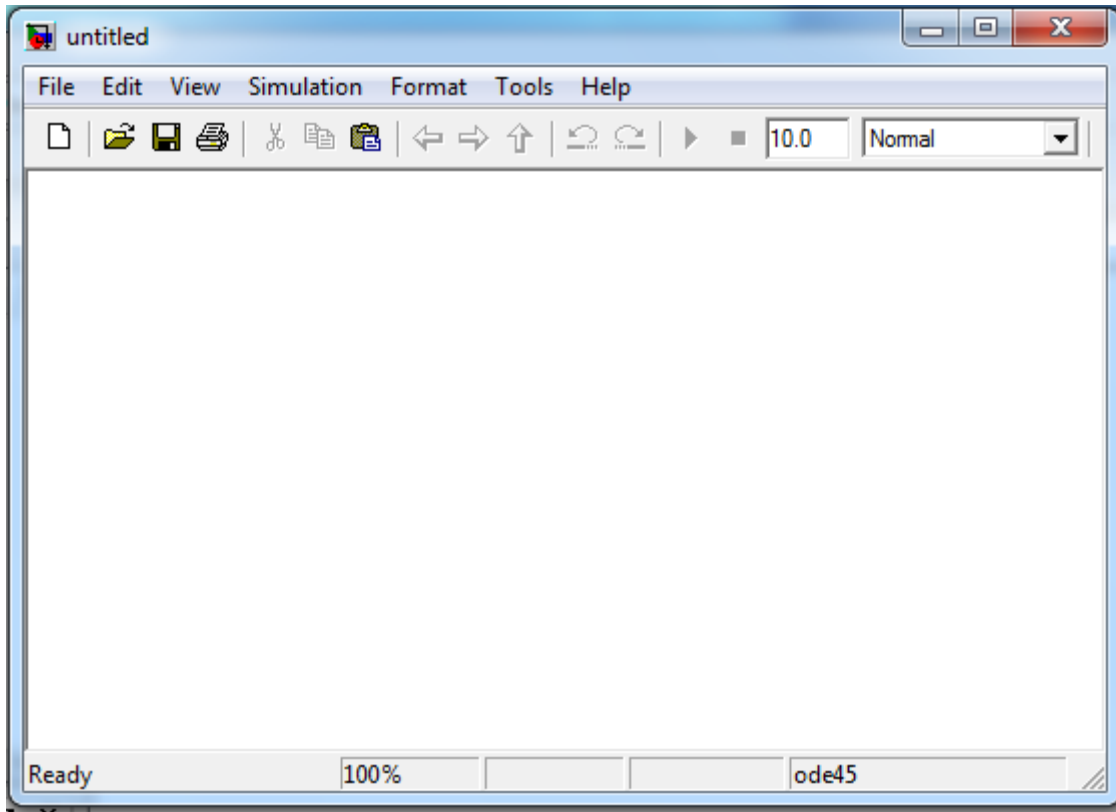


Figure7.2:- New Simulink model

CONCLUSION:-

Exercise No8: (2 Hours) – 1 Practical

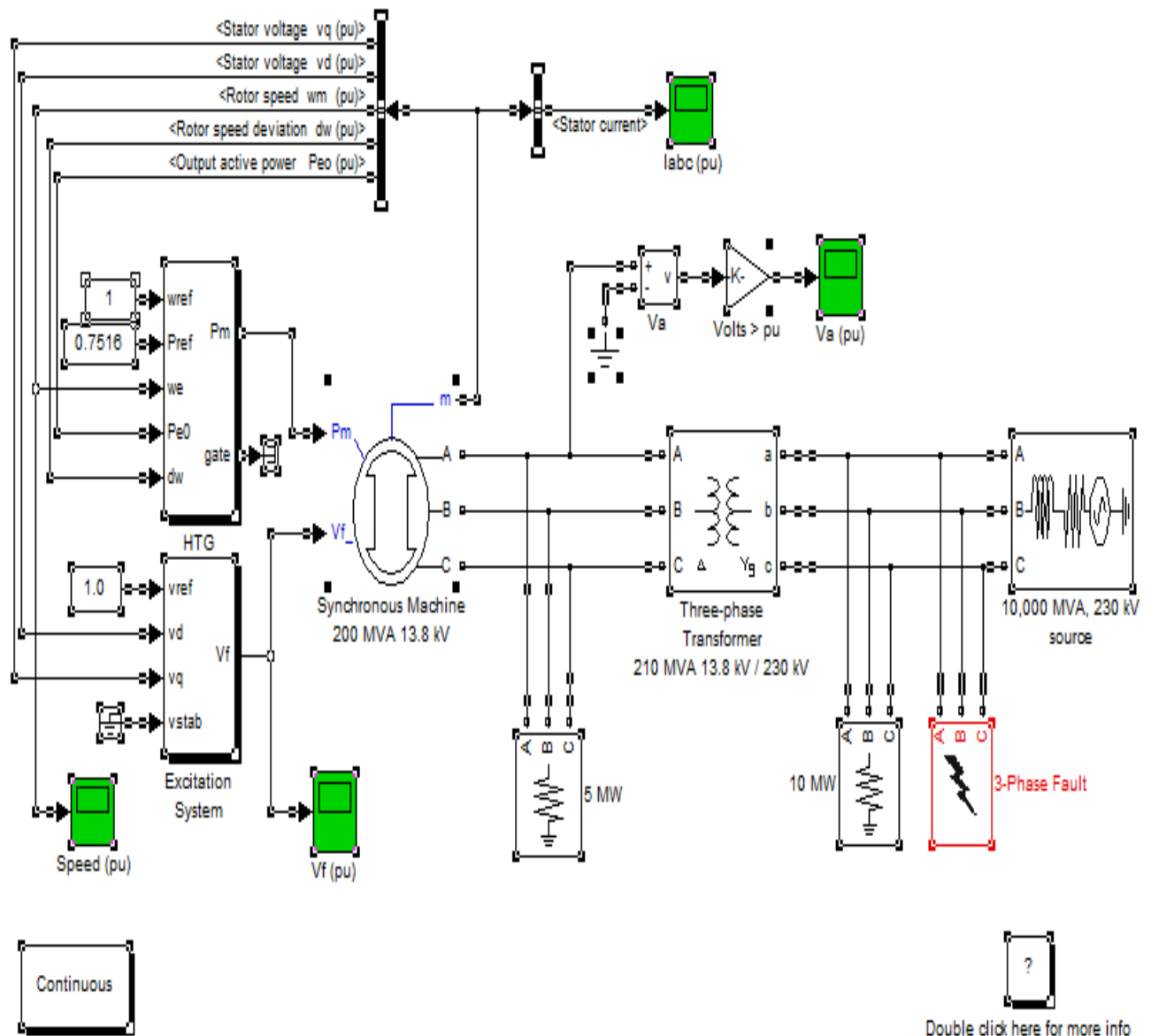
TITLE:-Representation of Excitation Systems.

AIM: - Simulate the excitation systems in matlab

THEORY:-

- a) Explain in detail types of excitation systems
- b) Consider a problem and solve the problem and use the values the following blocks.

SIMULINK MODEL:-



MODEL RESULTS:-

CONCLUSION:-

Exercise No9: (2 Hours) – 1 Practical

TITLE: - Modeling of Synchronous machines.

AIM: - Simulate the synchronous machine model in matlab

THEORY:-

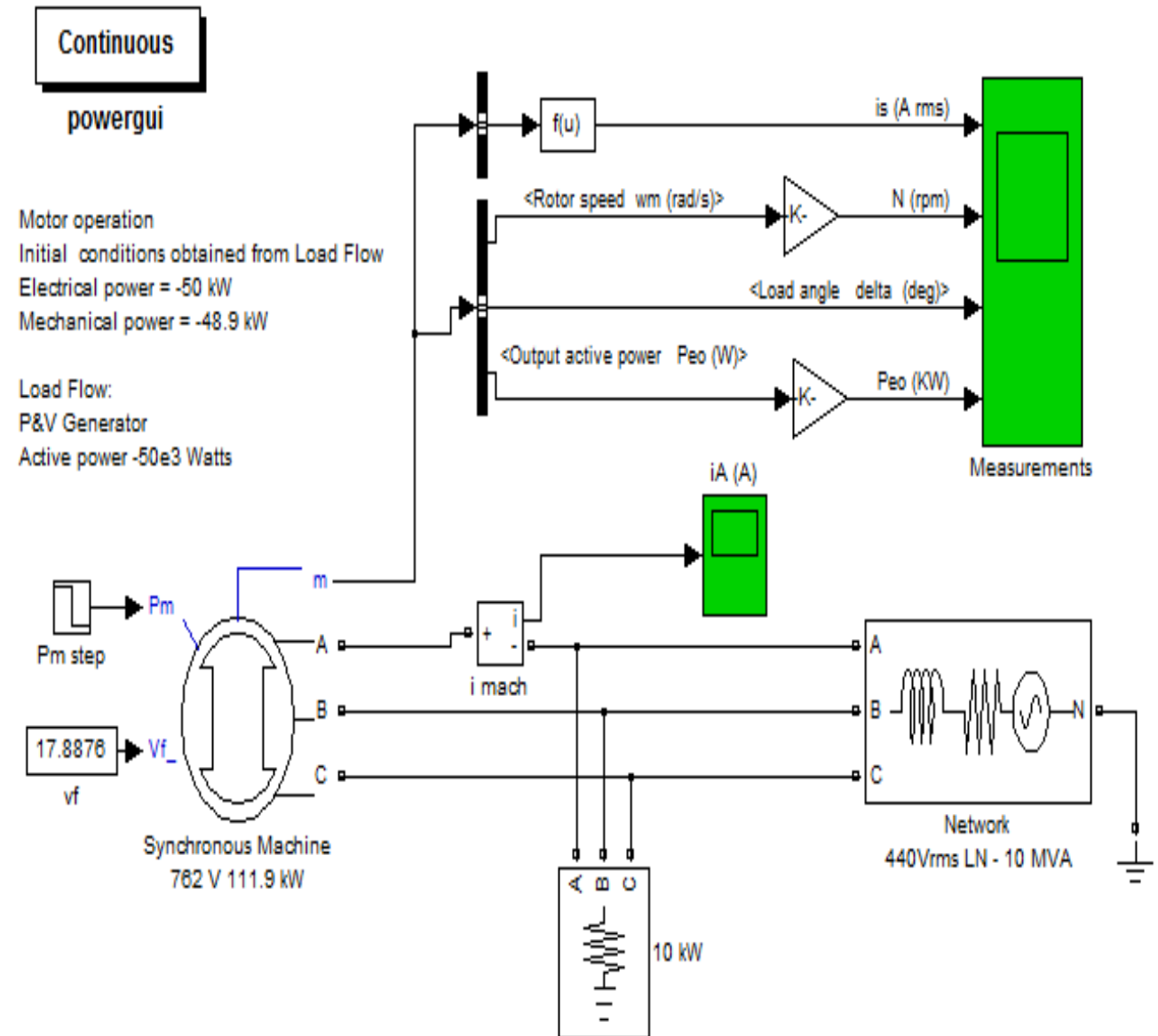
- a) Explain in detail each block of synchronous machine model
- b) Consider a problem and solve the problem and use the values the following blocks.

The use of the Synchronous Machine block in motor mode. The simulated system consists of an industrial grade synchronous motor (150 HP (112 kVA), 762 V) connected to a network with a 10 MVA short-circuit level. In order to start simulation in steady state, the machine is initialized using the **Load Flow and Machine Initialization** option of the Powergui. The machine is initialized for an output electrical power of -50 kW (negative value for motor mode), corresponding to a mechanical power of -48.9 kW. The corresponding values of mechanical power and field voltage have been automatically entered by the **Load Flow** analysis into the Pm Step block and in the Vf Constant block. The Pm Step block has been programmed in order to apply a sudden increase of mechanical power from -48.9 kW to -60 kW at time $t = 0.1$ s.

Run the simulation and observe the RMS current, RMS voltage, speed, load angle δ , and output electrical power of the motor.

Since this is a four-pole machine, the nominal speed is 1800 rpm. The initial speed is 1800 rpm as prescribed. After the load has increased from 48.9 kW to 100 kW at $t = 0.1$ s, the machine speed oscillates before stabilizing to 1800 rpm. The load angle (angle between terminal voltage and internal voltage) increases from -21 degrees to -53 degrees.

SIMULINK MODEL:-



MODEL RESULTS:-

CONCLUSION:-

Exercise No10: (2 Hours) – 1 Practical

TITLE: - Analysis of Multimachine Systems.

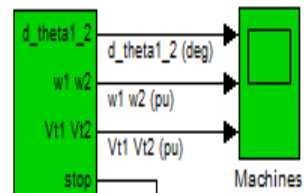
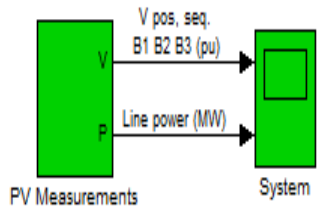
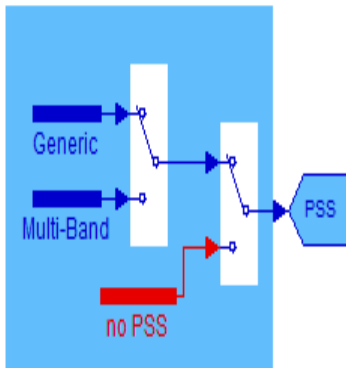
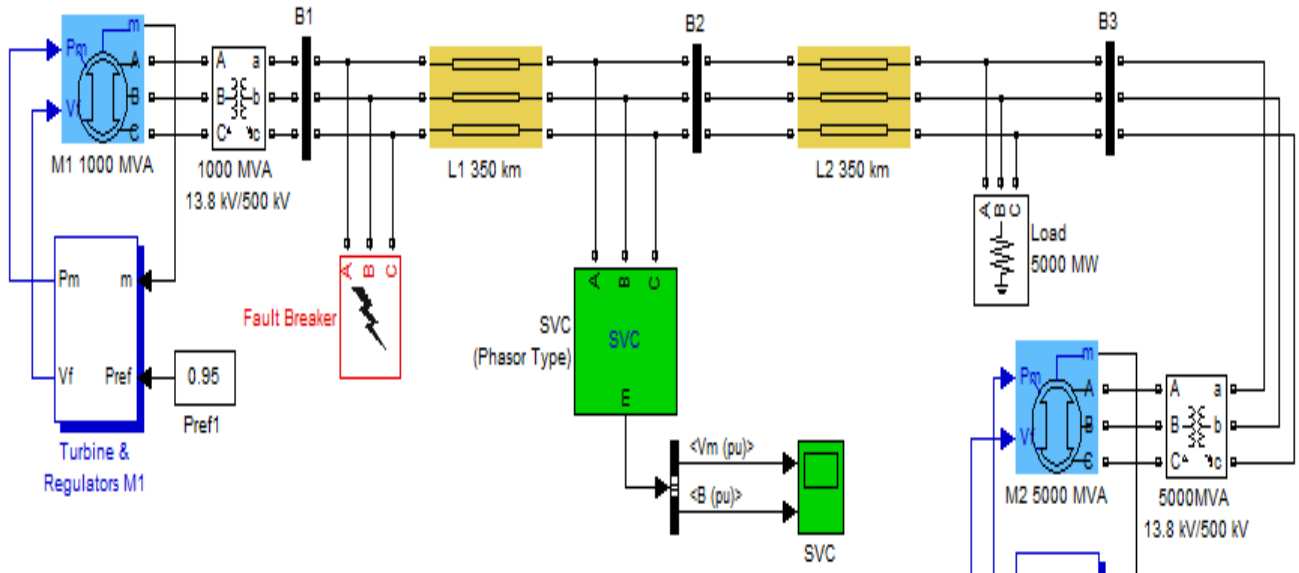
AIM: - Simulate the Multimachine Systems model in matlab

THEORY:-

- a) Explain in detail each block of Multimachine Systems model
- b) Consider a problem and solve the problem and use the values the following blocks.

This model illustrates use of the phasor solution for transient stability analysis of multimachine systems. It analyzes transient stability of a two-machine transmission system with Power System Stabilizers (PSS) and Static Var Compensator (SVC).

SIMULINK MODEL:-



Show comparison Detailed vs Phasor simulation

Show impact of SVC for 3-phase fault

Show impact of PSS for 1-phase fault

Phasors
powergui

?
info

MODEL RESULTS:-

CONCLUSION:-