VOLTAGE REGULATION OF A 3-Φ NON-SALIENT POLE TYPE ALTERNATOR USING MATLAB PROGRAMMING

Presented by :
Sambhav R Jain
107108103
Department of Electricals and Electronics Engineering

Voltage Regulation

1. Ampere-turns method (M.M.F. method)
   - Optimistic method
   - Doesn't require stator resistance
   - Two field current components
   - Phasor addition-graphical method
2. Synchronous-impedance method (E.M.F. method)
   - Pessimistic method
   - Effect of saturation neglected
   - Calculation of synchronous impedance-analytical method

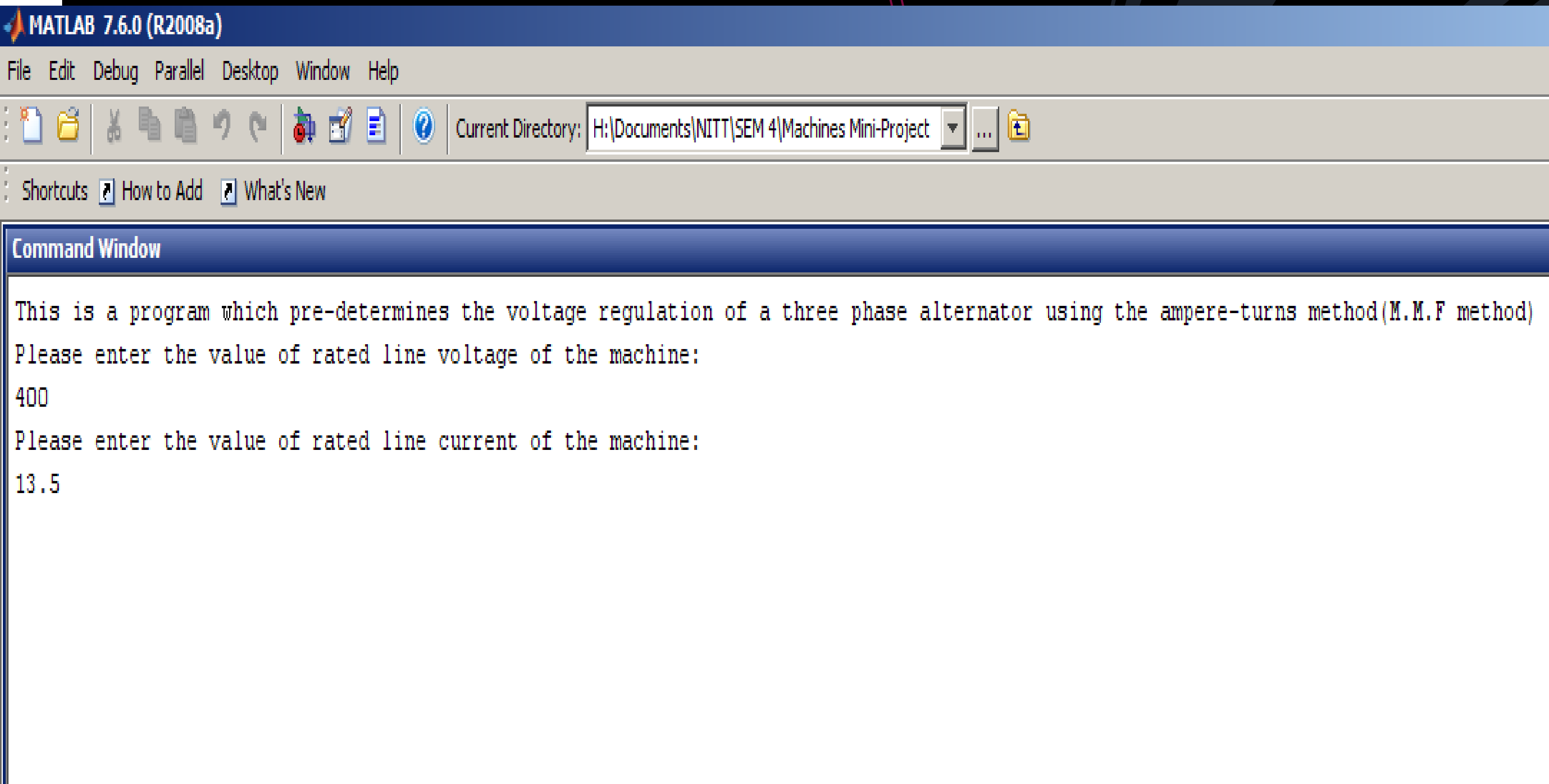
Ampere-turns method

MACHINE PARAMETERS :

Firstly, the machine ratings are accepted from the user as :

```
clear all;
close all;
clc;
disp('This is a program which pre-determines the voltage regulation of a three phase alternator using the ampere-turns method(M.M.F method)');
%Accepting data from the user:
V=input('Please enter the value of rated line voltage of the machine:\n');
I=input('Please enter the value of rated line current of the machine:\n');
```

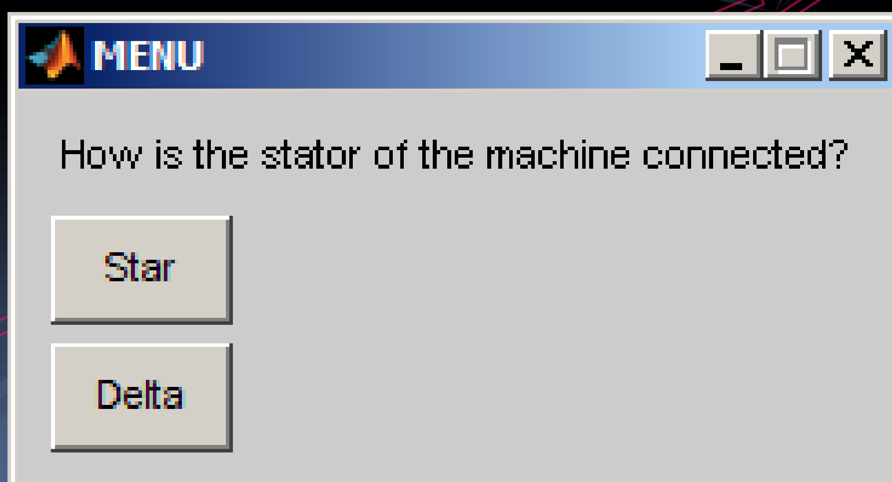
OUTPUT :


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Current Directory: H:\Documents\NITT\SEM 4\Machines Mini-Project
Shortcuts How to Add What's New
Command Window
This is a program which pre-determines the voltage regulation of a three phase alternator using the ampere-turns method(M.M.F method)
Please enter the value of rated line voltage of the machine:
400
Please enter the value of rated line current of the machine:
13.5

STATOR CONNECTION :

A menu is provided for the user to select from :

```
d=menu('How is the stator of the machine connected?','Star','Delta');
if d==1
    Vr=V/sqrt(3);
    Ir=I;
elseif d==2
    Vr=V;
    Ir=I/sqrt(3);
end
```



O.C. and S.C. TEST READINGS :

User is now required to enter the open circuit and short circuit test readings in a row-matrix form :

```
If0=input('Please enter the open-circuit field currents recorded in a row matrix form:\n');
E0=input('Please enter the corresponding values of open circuit voltages in a row matrix form:\n');
Ifs=input('Enter the field current during short circuit test (including the zero value)\nin a row matrix form:');
Isc=input('Enter the corresponding short circuit current that is circulated (including the zero value)\nin a row matrix form:');
```

Command Window

```
Please enter the open-circuit field currents recorded in a row matrix form:
[0 0.64 0.76 0.92 1.08 1.20 1.28 1.60 1.84 1.96 2.20 2.32 2.60 2.80 2.96 3.20 3.40 3.56 3.84];
Please enter the corresponding values of open circuit voltages in a row matrix form:
[0 160 184 212 248 276 292 348 380 400 424 440 464 480 488 504 512 520 536];
Enter the field current during short circuit test (including the zero value)
in a row matrix form:[0 3.56];
Enter the corresponding short circuit current that is circulated (including the zero value)
in a row matrix form:[0 13.5];
```

CURVE-FITTING
polyfit()

POLYNOMIAL CURVE-FITTING :

To plot the O.C.C of the alternator in MATLAB, a predefined function called polyfit() is used, which evaluates the best fitting curve of a specified degree

```
%Getting the best fitting 3rd degree curve by method of least squares:
PO=polyfit(If0,E0,3);
```

We choose a third degree polynomial for least residual error

INTERPOLATION :

- Magnitude of field current required for excitation and to overcome leakage reactance is one that corresponds to the rated voltage on the O.C.C.
- Magnitude of field current required to overcome armature reaction effect is one that corresponds to rated current on S.C.C.

$$y=f(x)$$

$$y=P_0(1)x^3+P_0(2)x^2+P_0(3)x+P_o(4)$$

- At Vr, the corresponding value of field current is If1

$$Vr=P_0(1)x^3+P_0(2)x^2+P_0(3)x+P_o(4)$$

$$0=P_0(1)x^3+P_0(2)x^2+P_0(3)x+P_o(4)-Vr$$

$$P_1=[P_0(1)\ \ P_0(2)\ \ P_0(3)\ \ P_o(4)-Vr]$$

INTERPOLATION :

```
%Finding out the value of field current at rated terminal voltage(If1):
%To find the root of the above polynomial fit at the rated voltage:
P1=[P0(1) P0(2) P0(3) P0(4)-Vr];
c=roots(P1); %This gives out three roots, out of which we need to select the appropriate one:
i=1;
while E0(i)<=Vr
    i=i+1;
end
if Vr==E0(i-1);
    If1=If0(i-1);
else
    if (If0(i-2)<c(1))&&(If0(i)>c(1))
        If1=c(1);
    elseif (If0(i-2)<c(2))&&(If0(i)>c(2))
        If1=c(2);
    elseif (If0(i-2)<c(3))&&(If0(i)>c(3))
        If1=c(3);
    end
end
disp('Thus the magnitude of field current corresponding to rated terminal voltage is:');
disp(If1);
```

INTERPOLATION :

```
%To find the field current when rated short circuit current flows(If2):
m=(Isc(2)/sqrt(3))/Ifs(2); %Slope 'm' is found assuming the S.C.C. to be a straight line variation.
If2=Ir/m;
disp('The magnitude of field current required to overcome armature reaction effect is:');
disp(If2);
```

Command Window

```
Thus the magnitude of field current corresponding to rated terminal voltage is:
    1.9600

The magnitude of field current required to overcome armature reaction effect is:
    3.5600
```

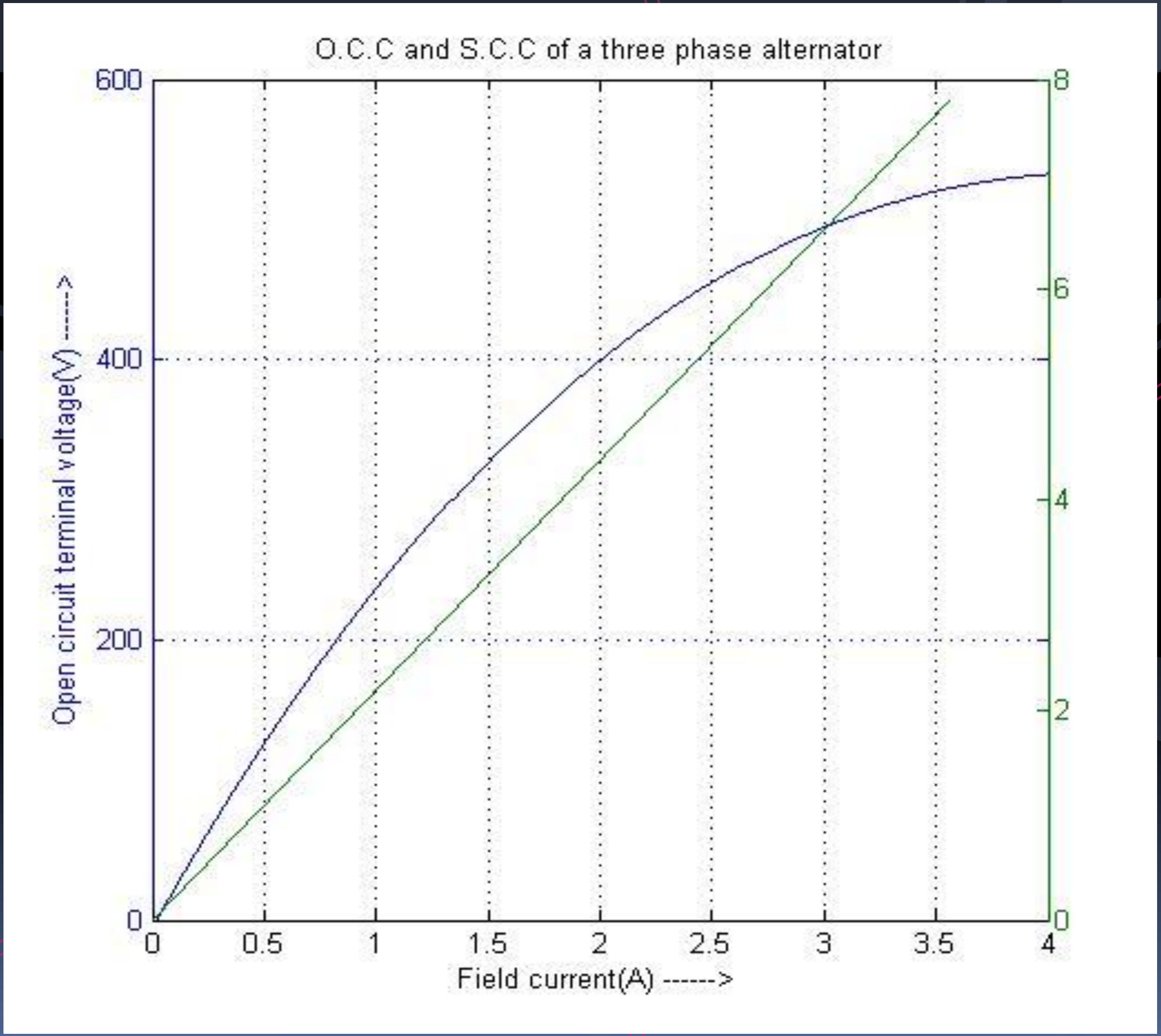
PLOT TOOLS
plotyy()

PLOT THE O.C.C. AND S.C.C. :

- plotyy() is a function of MATLAB which plots a graph with two independent Y-axes against one X-axis
- polyval(P,a) fetches the value of polynomial P at ‘a’

```
%To plot the O.C.C. and S.C.C.:
a=0:.01:max(If0);
b=polyval(P0,a);
plotyy(a,b,Ifs,Isc./sqrt(3));
ylim([0 1.5*Vr]);
xlabel('Field current(A) ------>');
ylabel('Open circuit terminal voltage(V) ------>');
title('O.C.C and S.C.C of a three phase alternator')
grid on;
```

Characteristics :



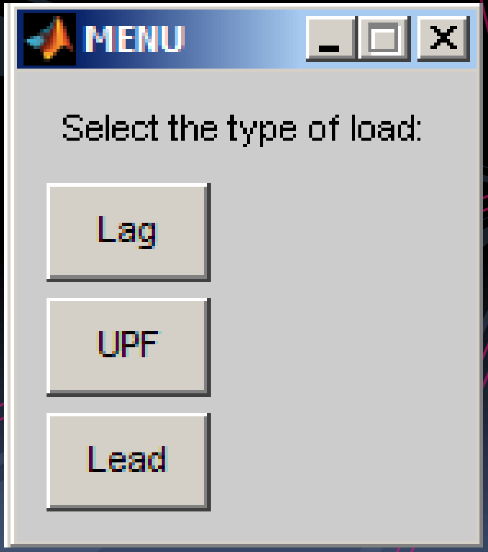
Type of load and power factor :

A menu allows the user to select the type of load and its power factor

```
k=menu('Select the type of load:','Lag','UPF', 'Lead');
switch(k)
    case 1
        pf=input('Enter the power factor of load:');
        phi=(rad2deg(acos(pf)));
        If2=complex(If2*cos(deg2rad(90-phi)),If2*sin(deg2rad(90-phi)));
    case 2
        If2=complex(0,If2);
    case 3
         pf=input('Enter the power factor of load:');
         phi=(rad2deg(acos(pf)));
         If2=complex(-If2*cos(deg2rad(90-phi)),If2*sin(deg2rad(90-phi)));
end
```

Type of load and power factor :

After selecting the type of load, its power factor is to be specified



PHASOR ADDITION :

```
If=If1+If2;
mag=sqrt((real(If))^2+(imag(If))^2);
disp('The magnitude of the resultant vector sum of two field currents is:');
disp(mag);
E=polyval(PO,mag);
disp('The corresponding value of generated voltage(E) to overcome all loading effects is:');
```

The two field current components are added vectorially and the resultant magnitude is extended on the O.C.C. to find the required generated voltage

OUTPUT :

Command Window

```
Enter the power factor of load:0.8
The magnitude of the resultant vector sum of two field currents is:
    2.8534

The corresponding value of generated voltage(E) to overcome all loading effects is:
  484.6279

Hence the pre-determined voltage regulation is:
   21.1570
```

Thus the percentage voltage regulation of the alternator is pre-determined at the given loading conditions using the ampere-turns method in **MATLAB**

Synchronous impedance method

Synchronous impedance, Z_s

$$Zs = \frac{Open - circuit\ voltage\ corresponding\ to\ a\ field\ current}{Short - circuit\ current\ corresponding\ to\ the\ same\ field\ current}$$

Finding Zs and Xs

```
R=input('Enter the value of measured stator resistance(a.c.):');
%Getting the best fitting 3rd degree curve by method of least squares:
P0=polyfit(If0,E0,3);
disp('The rated phase current of the machine is given to be:');
disp(Ir);
%To find the field current when rated short circuit current flows:
m=(Isc(2)/sqrt(3))/Ifs(2);
If=Ir/m;
disp('The field current(If) required to circulate rated short-circuit current is:');
disp(If);
%To find the value of open circuit voltage corresponding to the above field current If
E=polyval(P0,If);
disp('The corresponding open-circuit voltage(E) is:');
disp(E);
Zs=E/Ir;
disp('Thus the value of synchronous impedance(Zs) is');
disp(Zs);
Xs=sqrt(Zs^2-R^2);
disp('The value of synchronous reactance(Xs) calculated is:');
disp(Xs);
```

Finding E1 for different power factors

```
while c~=2
    k=menu('Select the type of load:','Lag','UPF', 'Lead');
switch(k)
    case 1
        pf=input('Enter the power factor of load:');
        phi=(acosd(pf)); %Returns the cos inverse in degrees
        E1=sqrt((Vr*cosd(phi)+Ir*R)^2+(Vr*sind(phi)+Ir*Xs)^2);
    case 2
        E1=sqrt((Vr+Ir*R)^2+(Ir*Xs)^2);
    case 3
        pf=input('Enter the power factor of load:');
        phi=(acosd(pf)); %Returns the cos inverse in degrees
        E1=sqrt((Vr*cosd(phi)+Ir*R)^2+(Vr*sind(phi)-Ir*Xs)^2);
end
```

THANK YOU