ISLAMIC UNIVERSITY OF TECHNOLOGY

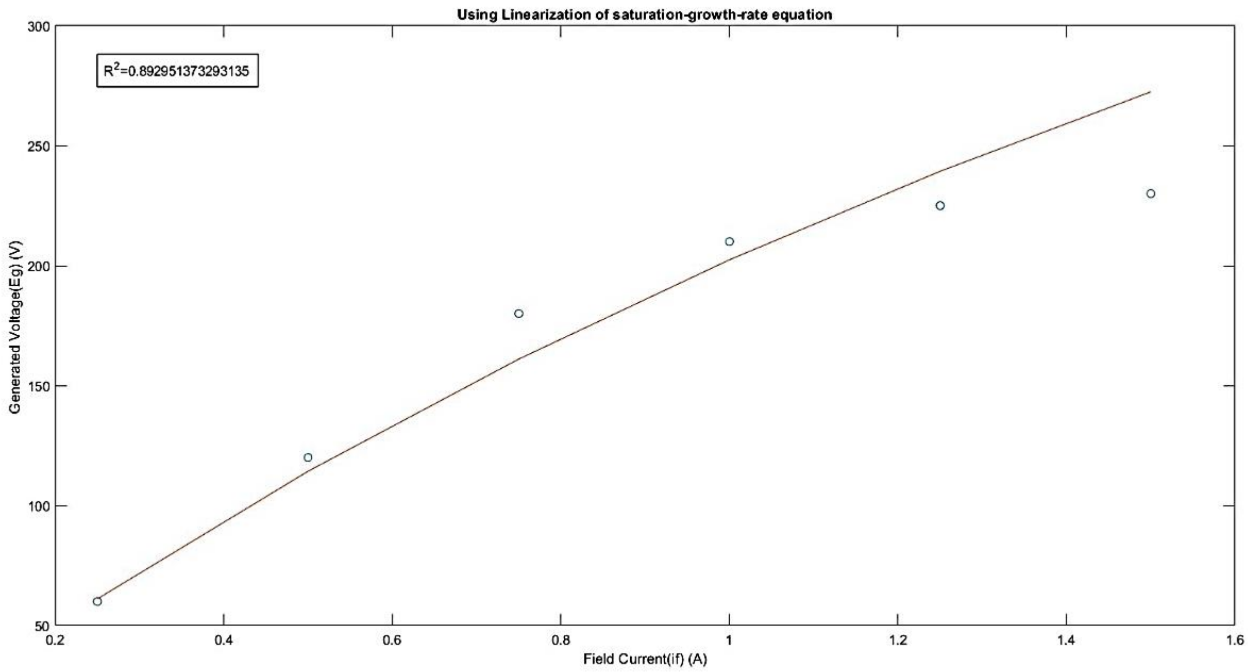
(IUT)

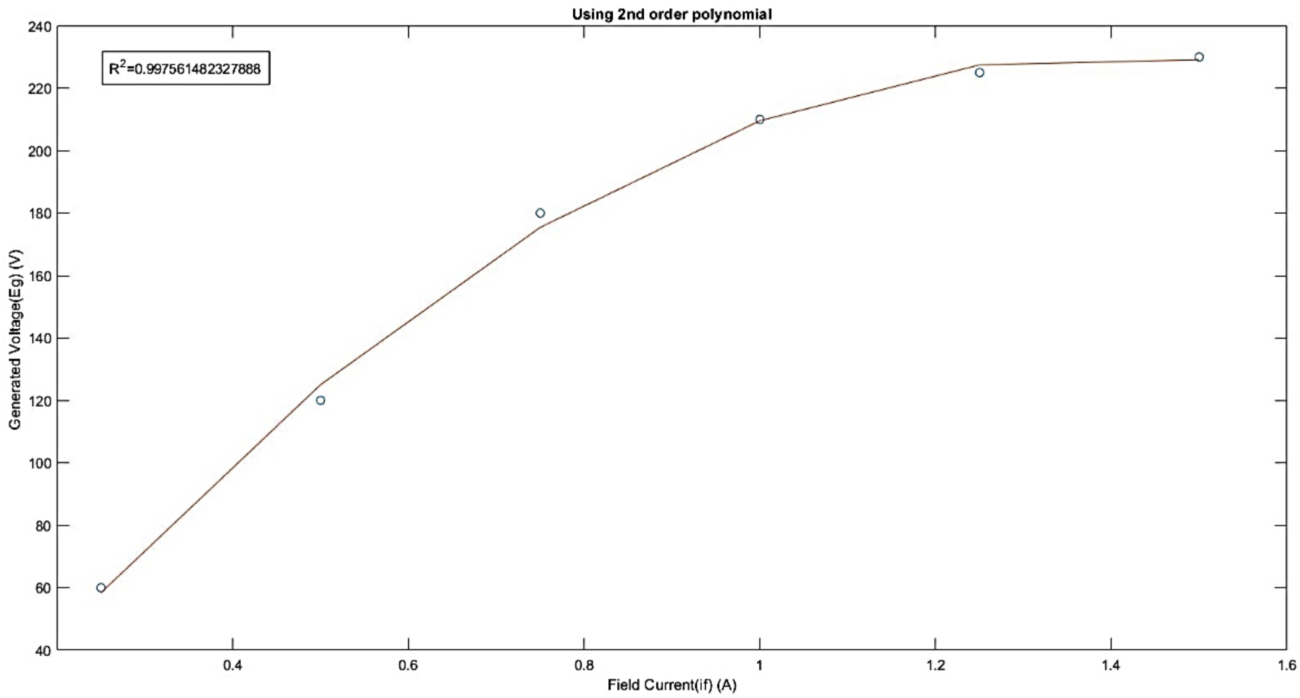
CLASS TEST 4 (COMPULSORY)

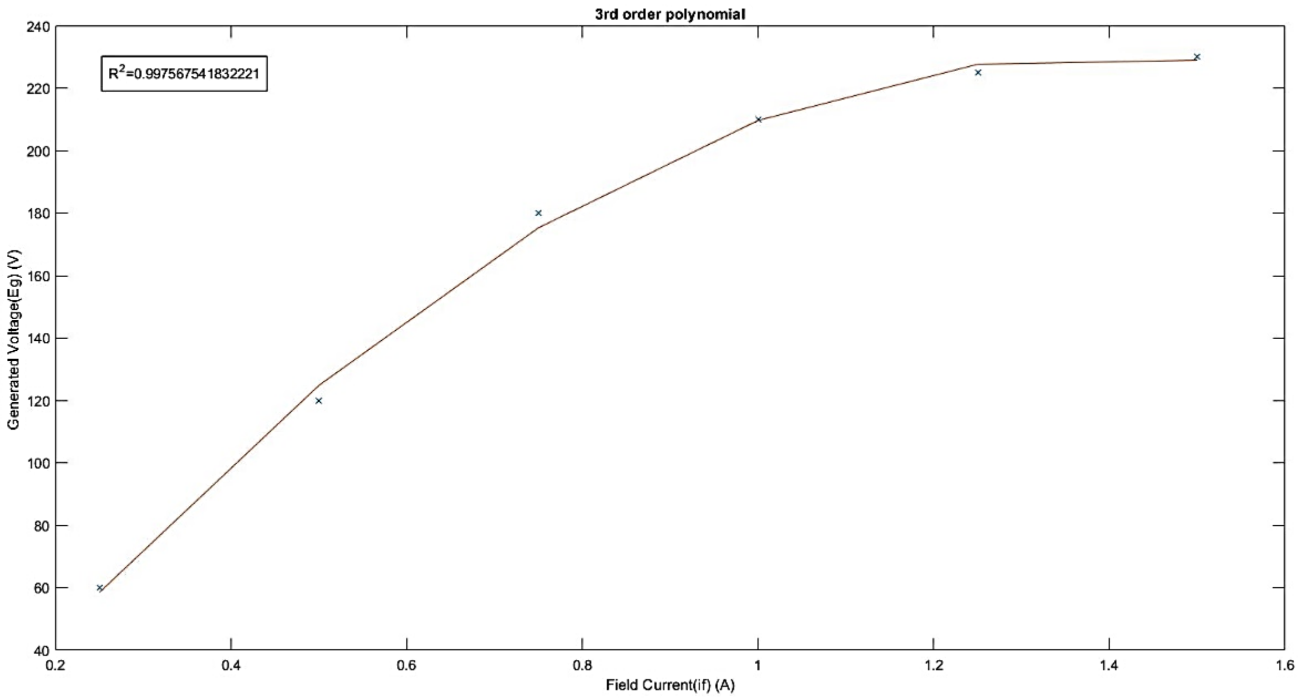
(COMPLEX ENGINEERING PROBLEM)

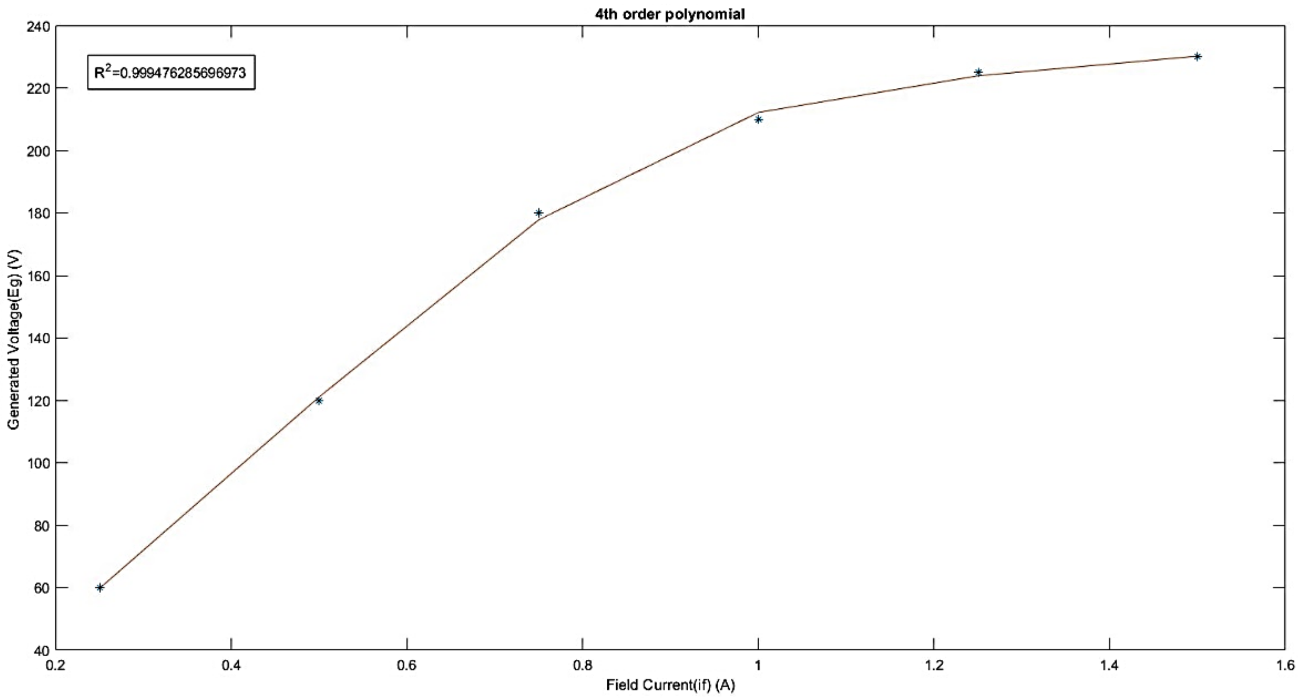
Course	:	Math 4522 (Numeric Methods Lab)
Student ID	:	**190021119**
Name	:	Redwan-Ul-Bari
Department	:	Electrical & Electronic Engineering (EEE)
Section	:	A
Group	:	A1

Modeling the OCC by different curves using the least square error method using Matlab:









Values of R^2 in percentage (%) indicating to the evaluation of the goodness of fit:

Linearization of saturation-growth-rate equation	2nd order polynomial	3rd order polynomial	4th order polynomial
89.2951373293135	99.7561482327888	99.7567541832221	99.9476285696973

As 4th order polynomial has the highest value of R^2 it has been chosen to model the OCC curve of the DC generator.

$$E_g(I_f) = a_o + a_1 I_f + a_2 {I_f}^2 + a_3 {I_f}^3 + a_4 {I_f}^4$$

From matlab code we get:

$$a_0 = 29.1666666658712$$

$$a_1 = -3.46560846362263$$

$$a_2 = 656.111111101694$$

$$a_3 = -656.296296290122$$

$$a_4 = 186.666666665347$$

$$E_g(I_f) = 29.1666666658712 - 3.46560846362263 I_f + 656.111111101694 {I_f}^2 - 656.296296290122 {I_f}^3 + 186.666666665347 {I_f}^4$$

From matlab code,

A = 1727

Y = 17

Generator Parameters:

$$\text{Shunt field Voltage} = 220\ \text{V}$$

$$\text{Shunt Field Resistance, } R_f = 150\ \Omega$$

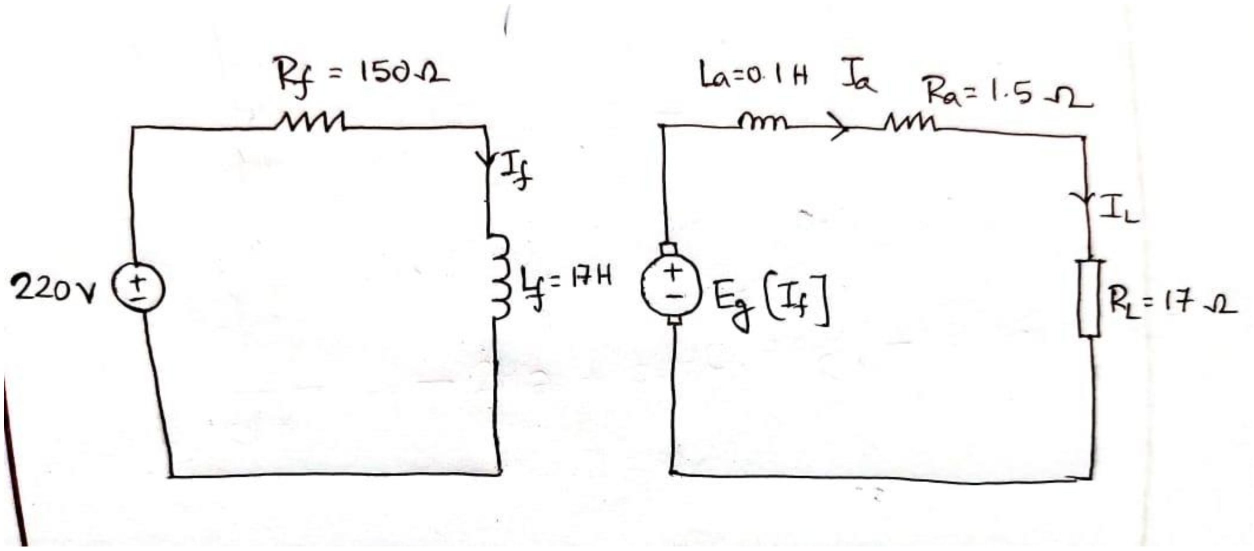
$$\text{Shunt Field Inductance, } L_f = 17\ \text{H}$$

$$\text{Armature Resistance, } R_a = 1.5\ \Omega$$

$$\text{Armature Inductance, } L_a = 0.1\ \text{H}$$

$$\text{Load } R_L = 17\ \Omega$$

Equivalent circuit for separately excited DC generator:



DEs for the system:

$$R_f I_f + L_f \frac{dI_f}{dt} = 220$$

$$L_a \frac{dI_a}{dt} + R_a I_a + R_L I_L = E_g(I_f)$$

Rearranging the ODEs:

$$R_f I_f + L_f \frac{dI_f}{dt} = 220$$

$$\Rightarrow 150I_f + 17\frac{dI_f}{dt} = 220$$

$$\Rightarrow \frac{dI_f}{dt} = 12.941176470588236 - 8.823529411764707I_f$$

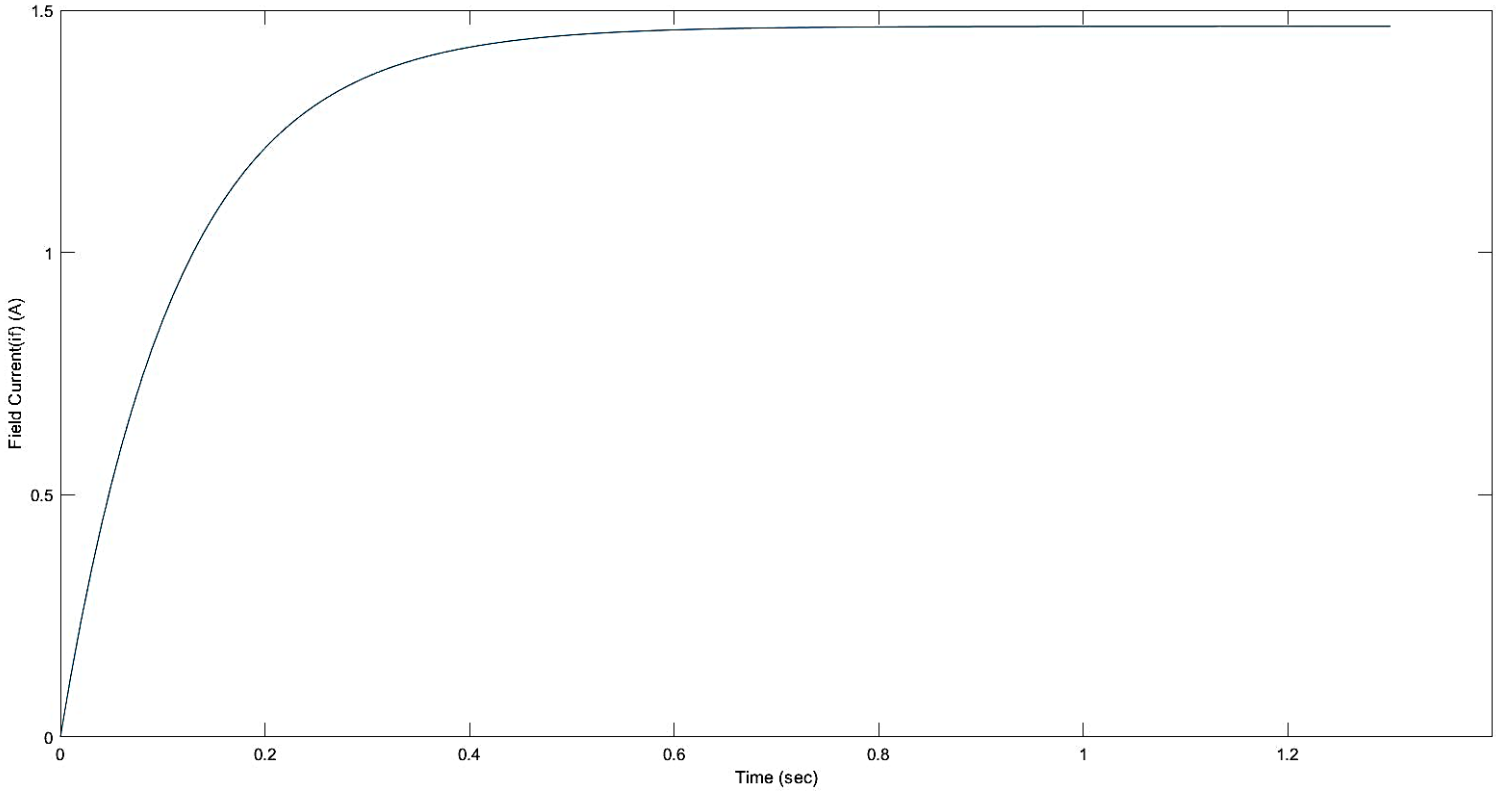
$$\Rightarrow \frac{dI_f}{dt} = f_1(t, I_f, I_a)$$

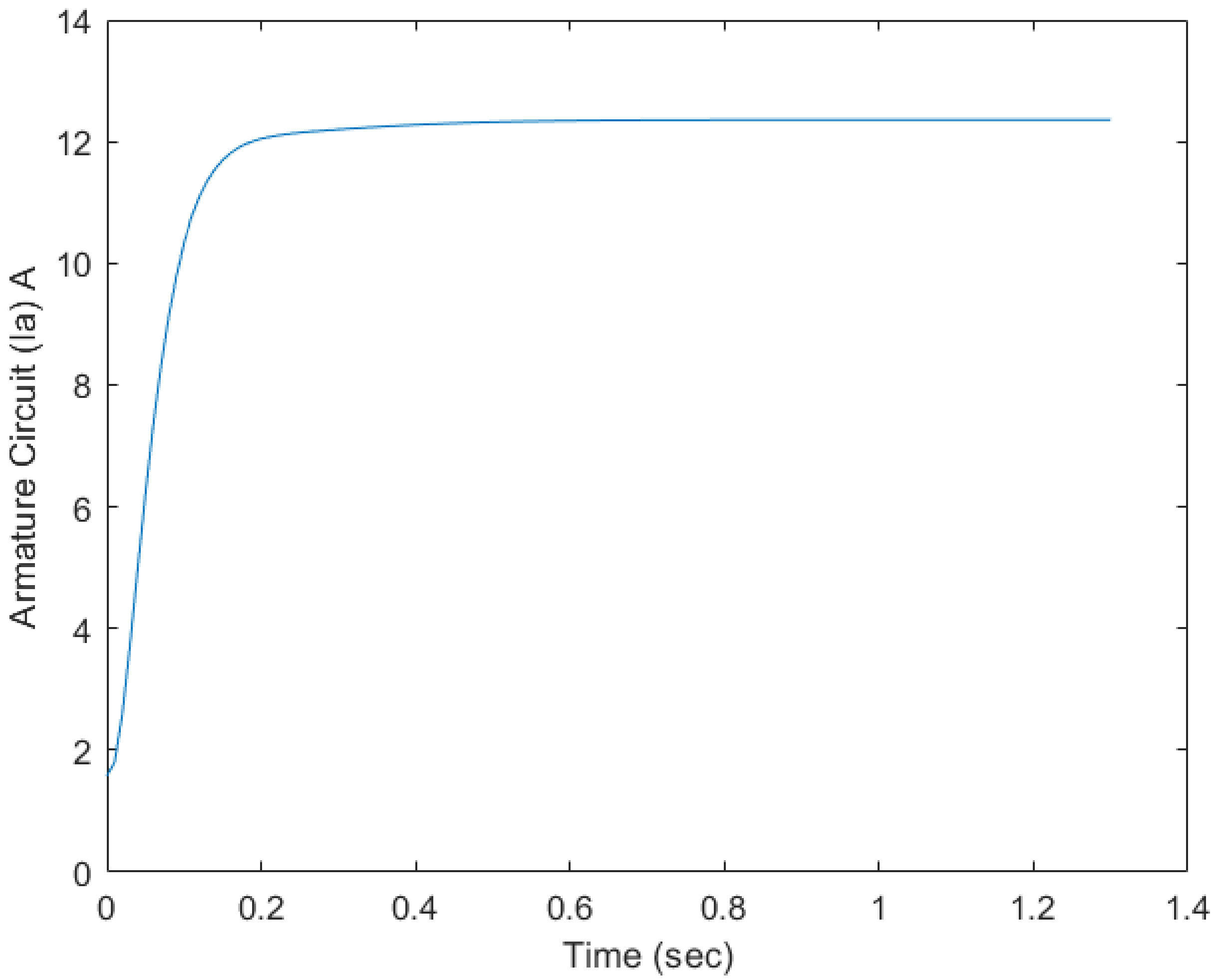
$$L_a \frac{dI_a}{dt} + R_a I_a + R_L I_L = E_g\left(I_f\right)$$

$$\Rightarrow 0.1\frac{dI_a}{dt} = E_g\left(I_f\right) - 1.5I_a - 17I_a \ [As\ I_L = I_a]$$

$$\Rightarrow \frac{dI_a}{dt} = \frac{E_g\left(I_f\right)}{0.1} - 185I_a$$

$$\Rightarrow \frac{dI_a}{dt} = f_2(t, I_f, I_a)$$


1.5
1
0.5
0
Field Current(if) (A)
0
0.2
0.4
0.6
0.8
1
1.2
Time (sec)


14
12
10
8
6
4
2
0
Armature Circuit (Ia) A
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Time (sec)

MATLAB CODE:

```
% Name: Redwan-Ul-Bari
% ID: 190021119
% Section: A
% Part 1: Modeling the OCC

clc
clear all
```

Open Circuit Characteristics

```
I=0.25:0.25:1.5
E=[60,120,180,210,225,230]
t=table(I',E','VariableNames',{'Field Current(if) (A)','Generated Voltage(Eg) (V)'})
```

Linearization of saturation-growth-rate equation

```
close all
c=1./I
e=1./E
n=1
I_s=zeros(n+1);
for i=1:(n+1)
    for j=1:(n+1)
    I_s(i,j)=(sum(c.^((i+j)-2)));
    end
end
E_s=zeros(n+1,1);
for i=1:(n+1)
    E_s(i,1)=(sum((c.^(i-1)).*e));
end
I_s
E_s
A=pinv(I_s)*E_s

Alpha=1/A(1,1)
Beta=A(2,1)*Alpha

plot(I,E,'o')
hold on;

y=(Alpha.*I)./(Beta+I)
plot(I,y)

Sr=sum((E-y).^2)
St=sum((E-mean(E)).^2)
r_squared=(St-Sr)/St
fprintf("Goodness of fit is %f percent",r_squared*100)
```

Fitting 2nd order polynomial

```
close all
n=2
I_s=zeros(n+1);
for i=1:(n+1)
    for j=1:(n+1)
    I_s(i,j)=(sum(I.^((i+j)-2)));
    end
end
E_s=zeros(n+1,1);
for i=1:(n+1)
    E_s(i,1)=(sum((I.^(i-1)).*E));
end
I_s
```

```
E_s
A1=pinv(I_s)*E_s

plot(I,E,'o')
hold on;
E_p=zeros(n+1,6);
for i=1:(n+1)
        E_p(i,:)=A1(i,1)*I.^(i-1);
end
Y=sum(E_p)
plot(I,Y)

Sr=sum((E-Y).^2)
St=sum((E-mean(E)).^2)
r_squared1=(St-Sr)/St
fprintf("Goodness of fit is %f percent",r_squared1*100)
```

Fitting 3rd order polynomial

```
close all
n=3
I_s=zeros(n+1);
for i=1:(n+1)
    for j=1:(n+1)
    I_s(i,j)=(sum(I.^((i+j)-2)));
    end
end
E_s=zeros(n+1,1);
for i=1:(n+1)
    E_s(i,1)=(sum((I.^(i-1)).*E));
end
I_s
E_s
A2=pinv(I_s)*E_s

plot(I,E,'x')
hold on;
E_p=zeros(n+1,6);
for i=1:(n+1)
        E_p(i,:)=A2(i,1)*I.^(i-1);
end
Y=sum(E_p)
plot(I,Y)

Sr=sum((E-Y).^2)
St=sum((E-mean(E)).^2)
r_squared2=(St-Sr)/St
fprintf("Goodness of fit is %f percent",r_squared2*100)
```

Fitting 4th order polynomial

```
close all
n=4
I_s=zeros(n+1);
for i=1:(n+1)
    for j=1:(n+1)
    I_s(i,j)=(sum(I.^((i+j)-2)));
    end
end
E_s=zeros(n+1,1);
for i=1:(n+1)
    E_s(i,1)=(sum((I.^(i-1)).*E));
end
I_s
E_s
```

```
A3=pinv(I_s)*E_s
plot(I,E,'*')
hold on;
E_p=zeros(n+1,6);
for i=1:(n+1)
        E_p(i,:)=A3(i,1)*I.^(i-1);
end
Y=sum(E_p)
plot(I,Y)

Sr=sum((E-Y).^2)
St=sum((E-mean(E)).^2)
r_squared3=(St-Sr)/St
fprintf("Goodness of fit is %f percent",r_squared3*100)
```

```
rstable=
table(r_squared*100,r_squared1*100,r_squared2*100,r_squared3*100,'VariableNames',{'Lineari
zation of saturation-growth-rate equation','2nd order polynomial','3rd order
polynomial','4th order polynomial'})
```

```
% Name: Redwan-Ul-Bari
% ID: 190021119
% Section: A
% Part 2: Solving ODEs using RK4
```

Generator Parameters

```
A = mod((119*263+71),1000) + 1359
Y= sum(num2str(A)-'0')
Vf=220
Rf= 150
Lf= Y
La= 0.1
Ra= 1.5
Rl=Y
```

OEDs

```
syms t I_f Ia
Eg(I_f)=A3(1,1)+A3(2,1)*I_f+A3(3,1)*(I_f^2)+A3(4,1)*(I_f^3)+A3(5,1)*(I_f^4)
f1(t,I_f,Ia)= Vf/Lf -(Rf*I_f)/Lf
f2(t,I_f,Ia)= Eg/La - (Ra*Ia)/La - (Ia*Rl)/La
```

Solve using RK4

```
format long
h=0.01
I_f=0
Ia=double(Eg(0))/(Ra+Rl)
i=1;
ea1=0;
ea2=0;

for t=0:h:1.3

k1_1=double(f1(t,I_f,Ia));


k1_2=double(f2(t,I_f,Ia));


k2_1=double(f1(t+h/2,I_f+k1_1*(h/2),Ia+k1_2*(h/2)));


k2_2=double(f2(t+h/2,I_f+k1_1*(h/2),Ia+k1_2*(h/2)));
```

```
k3_1=double(f1(t+h/2,I_f+k2_1*(h/2),Ia+k2_2*(h/2)));

k3_2=double(f2(t+h/2,I_f+k2_1*(h/2),Ia+k2_2*(h/2)));

k4_1=double(f1(t+h,I_f+k3_1*h,Ia+k3_2*h));

k4_2=double(f2(t+h,I_f+k3_1*h,Ia+k3_2*h));

I_f_n=I_f+(k1_1+2*k2_1+2*k3_1+k4_1)*(h/6);

Ian=Ia+(k1_2+2*k2_2+2*k3_2+k4_2)*(h/6);

ea1=((I_f_n-I_f)/I_f_n)*100;
ea2=((Ian-Ia)/Ian)*100;

T{i}=table(t,I_f,Ia,ea1,ea2,'VariableNames',{'Time (sec)','Field Current(if)
(A)','Armature Circuit (Ia) A','% for If','% for Ia'});

I_f=I_f_n;
Ia=Ian;

i=i+1;

end

d=vertcat(T{:})
```

Plotting growth of field current and armature current

```
plot(table2array(d(:,1)),table2array(d(:,2)))
plot(table2array(d(:,1)),table2array(d(:,3)))
```