## Table of Contents



# EE362 HW#1

# NAME: *JOSEPH HENRY*

# STUDENT NUMBER: 123456

Change your .m file name to the following: name_surname_ID_hw1.m

```
% Please add axis names, legends, titles etc. in all your plots

% Use the already defined variable names whenever possible

% Examine the whole template before you start

% Delete the hints, guidelines etc. given in this template when you
 prepare
% your solution

% Note that, MATLAB trigonometric functions use radians, not degrees
% Indexes in MATLAB start at 1, not 0
```

Q.1)

```
%Locked Rotor Test
P_lock= 4000; %W
V_lock=34; %V
I_lock= 250; %A

% No Load Test
P_no= 4500; %W
V_no=400 ; % V
```

```
I_no= 8.5 ; %A

%Loss
Rot_Loss= 4500 ; %W

% Measurement
R_dc= 0.06; %ohm

%
V_1= 690; % V
P_1= 700000 ; %W
p=4; %number
f=50; % Hz
```

# A

```
R_1= (R_dc/2)*1.1; % ohm ( 1.1 skin effect constant ...
                       ... 50Hz
% Locked Rotor Test
R_2p= P_lock/(I_lock^2) -R_1; % ohm
X_1=sqrt((V_lock/I_lock)^2- (R_1+R_2p)^2)/2; %ohm
X_2p=X_1;
fprintf("R_1 is equal to %f. \nR_2' is equal %f. \nX_1 and X_2' are
 equal %f. \n", R_1,R_2p,X_1);

% No load Test
P_c= P_no-(Rot_Loss/3); %W
R_c= V_no^2/ P_c; %ohm
X_m= 1/(sqrt((I_no/V_no)^2- (1/R_c^2))); % ohm

fprintf("R_c is equal to %f. \nX_m is equal %f. \n", R_c,X_m);

R_1 is equal to 0.033000.
R_2' is equal 0.031000.
X_1 and X_2' are equal 0.060000.
R_c is equal to 53.333333.
X_m is equal 100.000000.
```

# B

```
n_s = 120*f/p;
display(n_s);


n_s =

        1500
```

# C

```
s_Tmax= R_2p/sqrt(R_1^2 +(X_1+X_2p)^2); % slip
```

```
display(s_Tmax);
```

```
s_Tmax =

    0.2491
```

# D

```
ws=2*pi*n_s/60;
V_1=V_1/sqrt(3);
T_max= (3/2)*V_1^2*(1/(ws*((R_1 + sqrt(R_1^2+(X_1+X_2p)^2)))));
display(T_max)
```

```
T_max =

   9.6248e+03
```

# E

```
T_st= (3*V_1^2*R_2p)/(ws*((R_1+R_2p)^2+ (X_1+X_2p)^2));
display(T_st)
```

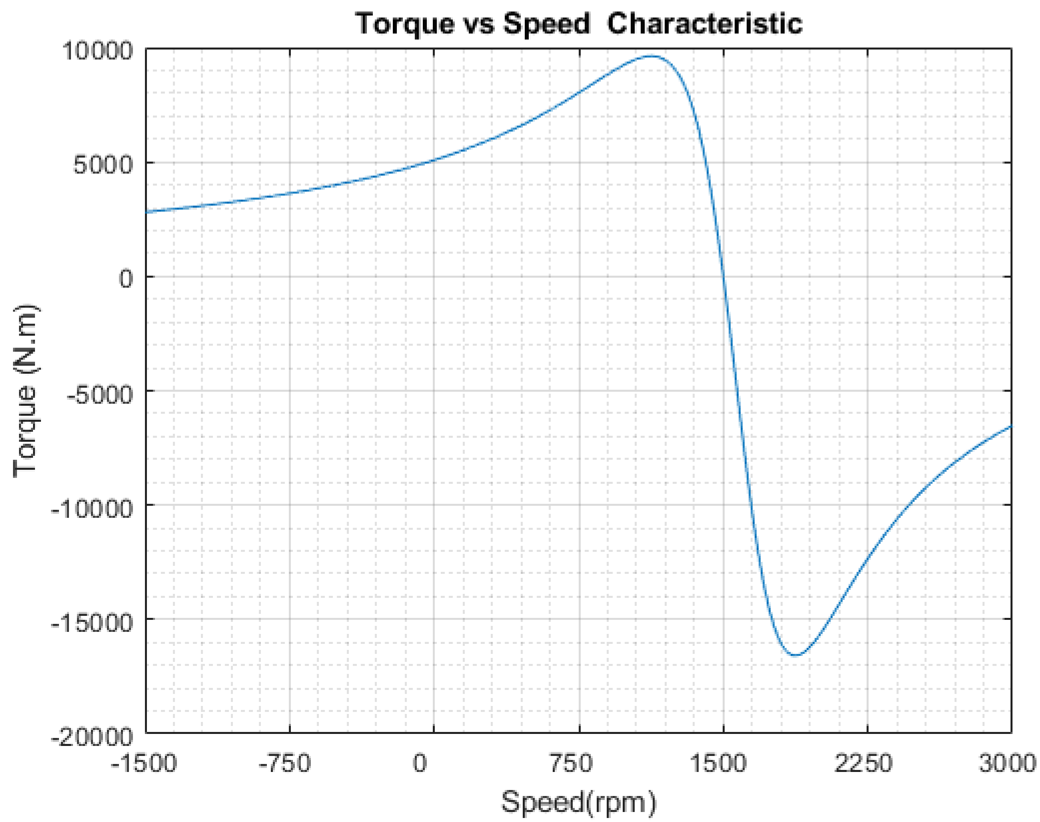
```
T_st =

   5.0800e+03
```

# F

```
s = -1:0.0001:2;
n_r= (1-s)*n_s;
T_mech= (3*V_1^2*R_2p)./(ws.*s.*((R_1 +R_2p./s).^2 +(X_1+X_2p)^2));

a=figure(1);
plot(s,T_mech);
xt = get(gca, 'XTick');
xtl = linspace(3000, -1500,numel(xt));
set(gca, 'XTick',xt, 'XTickLabel',xtl);
set(gca,'xdir','reverse');
title('Torque vs Speed  Characteristic' );
curtick = get(gca, 'YTick');
ylabel('Torque (N.m)');
xlabel('Speed(rpm)')
set(gca, 'YTickLabel', cellstr(num2str(curtick(:))));
grid on;
grid minor;
```




# G

What happens to the maximum torque when the terminal voltage is increased?

```
fprintf('Increases\n');

%What happens to the maximum torque when external resistances are
 connected to the rotor windings?
fprintf('No change\n');

% What happens to the starting torque when the terminal voltage is
 increased?
fprintf('Increases\n');

% What happens to the starting torque when external resistances are
 connected to the rotor windings?
fprintf('Increases\n');
```

```
Increases
No change
Increases
Increases
```

# h

```
s1=0.05;
```

```
T_load= (3*V_1^2*R_2p)./(ws.*s1.*((R_1 +R_2p./s1).^2 +(X_1+X_2p)^2));
n_r1= n_s*(1-s1); % rpm

display(T_load);
display(n_r1);
```

*T_load =*

*4.2630e+03*

*n_r1 =*

*1425*

```
w_r1= n_r1*2*pi/60; % rad/sec
P_mech= T_load*w_r1;
P_out= P_mech-Rot_Loss;
I_2p=sqrt((P_mech*s1)/(3*R_2p*(1-s1)));
P_cur= 3*I_2p^2*R_2p;
P_cus= 3*I_2p^2*R_1;
Pg=P_mech+P_cur;
Pc= 3*V_1^2/R_c;
Pin= Pc+P_cus+Pg;
Eff= (P_out/Pin)*100;

fprintf('Efficiency is equal to %f percentage\n', Eff);
fprintf('Input Power : %f W \n', Pin);
fprintf('Output Power : %f W \n', P_out);
fprintf('Cupper Loss in Stator Side : %f W \n', P_cus);
fprintf('Core Loss: %f W \n', Pc);
fprintf('Air Gap Power: %f W \n', Pg);
fprintf('Cupper Loss in Rotor Side: %f W \n', P_cur);
fprintf('Mechanical Power: %f W \n', P_mech);
fprintf('Rotational Loss: %f W \n', Rot_Loss);
```

*Efficiency is equal to 88.441610 percentage*
*Input Power : 714205.805330 W*
*Output Power : 631655.114800 W*
*Cupper Loss in Stator Side : 35641.967383 W*
*Core Loss: 8926.875000 W*
*Air Gap Power: 669636.962948 W*
*Cupper Loss in Rotor Side: 33481.848147 W*
*Mechanical Power: 636155.114800 W*
*Rotational Loss: 4500.000000 W*

# After you finished

Run the following command from Matlab terminal (command window) Generate a report of your .m file as pdf and ONLY upload the PDF file to ODTUClass.

```
% publish('name_surname_ID_hw1.m','pdf')
```

*Published with MATLAB® R2018a*