# ANALYSIS OF A PMAC MOTOR DRIVE

APRIL 10, 2023

ARIJEET NATH
Purdue University
ECE 51018

# 1. Introduction

This project seeks to simulate a permanent-magnet AC motor drive using a sine-triangle modulator with third harmonic injection. It is intended to demonstrate that desired torque can be rapidly achieved within 10's of milliseconds, provided all relevant drive system parameters remain within their limits. To obtain the desired torque from an electrical system, the following torque equation is used

$$T_e = \frac{3}{2} \cdot \frac{P}{2} \left( \lambda'_m I_q + \left( L_d - L_q \right) I_q I_d \right)$$

Using Matlab fmincon programming solver, one must determine the optimal values for $I_{qs}$ and $I_{ds}$ currents to achieve this torque. Once the optimal $I_{qs}$ and $I_{ds}$ values have been identified, they can be applied to create $V_{qs}$ and $V_{ds}$ voltages. As $V_{qs}$ and $V_{ds}$ are sinusoidal voltages, an inverter is utilized to convert dc voltage from the car into $V_{qs}$ and $V_{ds}$ voltages. To guarantee the desired torque is within system limits, current and voltage constraints may be employed if needed. Once these values have been reached, optimal $I_{qs}$ and $I_{ds}$ values can be computed. The required AC voltages for these results are obtained by passing DC voltage from the car battery through an inverter before being sent directly to the motor generator to produce desired currents.

Minimize

$$\mathrm{I_s} = \sqrt{\mathrm{I_{qs}^2} + \mathrm{I_{ds}^2}}$$

Subject to the constraint

$$T_e = \frac{3}{2} \cdot \frac{P}{2} \left( \lambda'_m I_q + \left( L_d - L_q \right) I_q I_d \right) \tag{1}$$

where $P$ is the number of poles, $\lambda'_m$ is the magnetic flux linkage, $\mathrm{I_{qs}}$ and $\mathrm{I_{ds}}$ the quadrature and direct components of the stator current respectively.

The equivalent circuit/block diagram of the system is presented in Figure 1, with the filter, motor, and control parameters listed in Tables 1 through 3. The top-level simulation block diagram is displayed in Figure 2. The inverter switches and diodes are considered ideal, enabling the use of standard Simulink components to depict the inverter, as depicted in Figure 3. The current control subsystem is described in Figure 6.

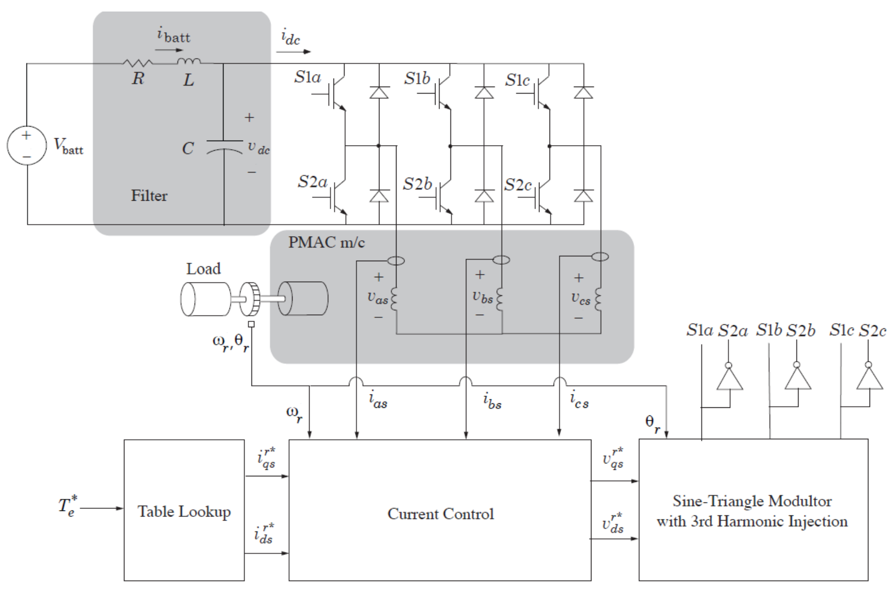


Figure 1 – Circuit diagram of the system

Table 1 – Source and Filter Parameters

| | |
|---|---|
| $V_{batt}$ | 400 V |
| $C$ | 2 mF |
| $L$ | 20 µH |
| $R$ | 0.01 Ω |

Table 2 – Motor Parameters

| | |
|---|---|
| $L_d$ | 2 mH |
| $L_q$ | 3.3 mH |
| $r_s$ | 0.02 Ω |
| $\lambda'_m$ | 0.2 V-s/rad |
| $P$ | 8 |
| $I_{max}$ | 225 A |

Table 3 – Current Regulator Parameters

| | |
|---|---|
| $K_q$ | 2 Ω |
| $K_d$ | 2 Ω |

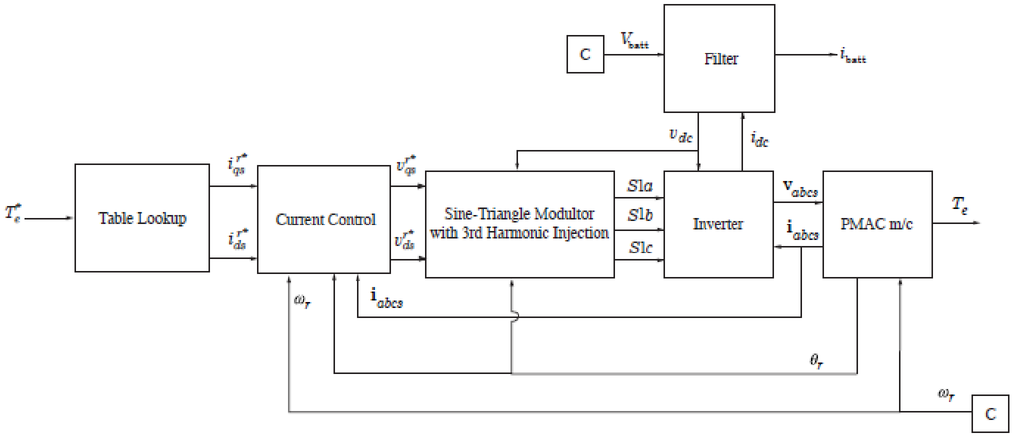


Figure 2 – Top level diagram of the system

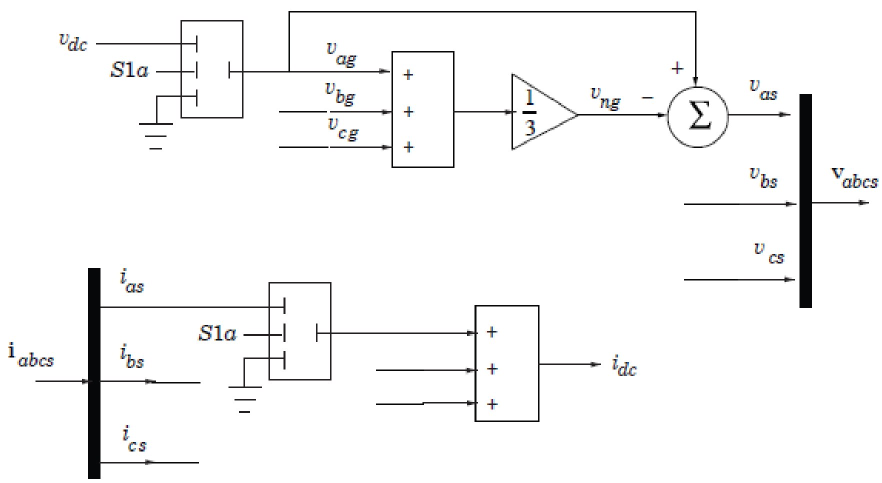

Figure 3 – Top level diagram of the Inverter subsytem

## 2. Simulink Model and Physics

This section will give a detailed overview on the development of the Simulink model, its important blocks and the pertinent equations used to develop the model. Figure. 5 shows the Simulink model of the entire PMAC architecture.

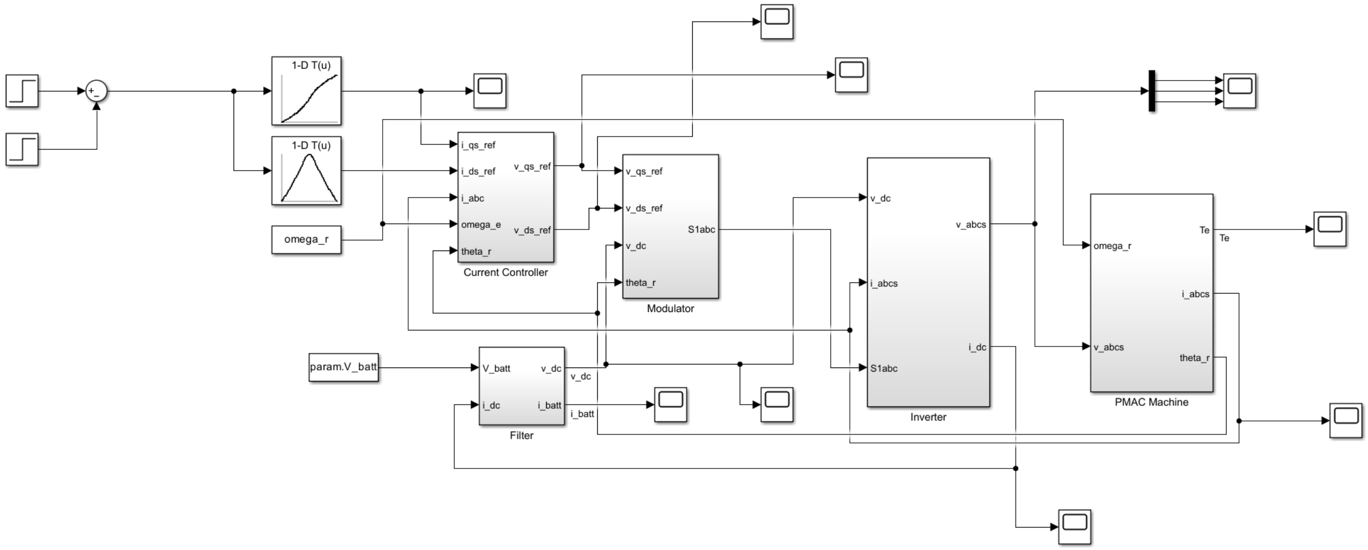


Figure 4 – Simulink block for the PMAC architecture

Figure 4 depicts a lookup table created to generate optimal currents, $I_{qs}$* and $I_{ds}$*, for the commanded torque. The current controller takes these currents , constant electrical speed $\omega_r$ and measures current $I_{abcs}$ as inputs. The current controller block then provides voltages required by sine triangle modulator which in turn generates inverter switching signals from these voltages. Figure 3 illustrates this process using switches $S_{1a}$, $S_{1b}$ and $S_{1c}$ along with the DC voltage $V_{dc}$ as they produce $V_{ag}$, $V_{bg}$ and $V_{cg}$ respectively; their average value results in $V_{ng}$. These equations are used to obtain $V_{as}$, $V_{bs}$, and $V_{cs}$

$$\begin{gathered} v_{ng} = \frac{1}{3}\left(v_{ag} + v_{bg} + v_{cg}\right) \\ v_{as} = v_{ng} - v_{ag} \\ v_{bs} = v_{ng} - v_{bg} \\ v_{cs} = v_{ng} - v_{cg} \end{gathered} \tag{2}$$

The outputs of the inverter are $V_{abcs}$ *and* $I_{dc}$. The PMAC machine block takes in the electrical speed $\omega_r$ and $V_{abcs}$ as inputs and calculates the desired torque $T_e$, $I_{abcs}$ and $\theta_r$ as outputs. The current controller block, modulator, inverter block and filter block are discussed below

## 2.1 Current Controller Block

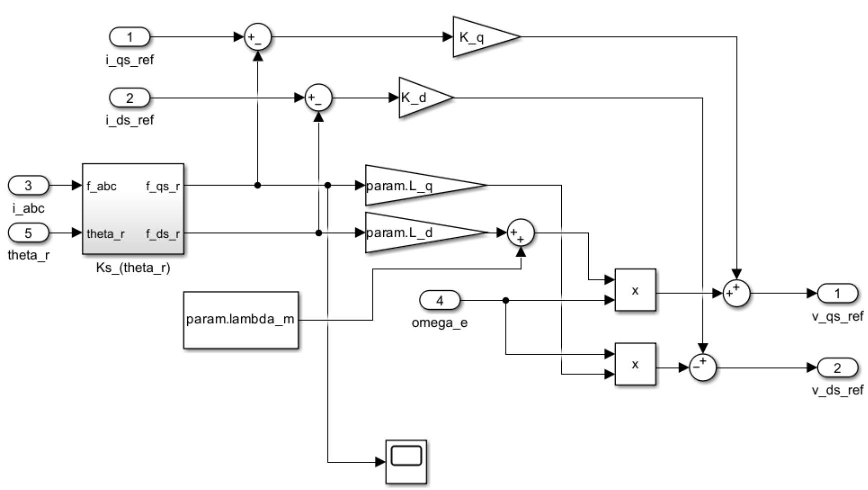


Figure 5 – Current controller subsystem in Simulink

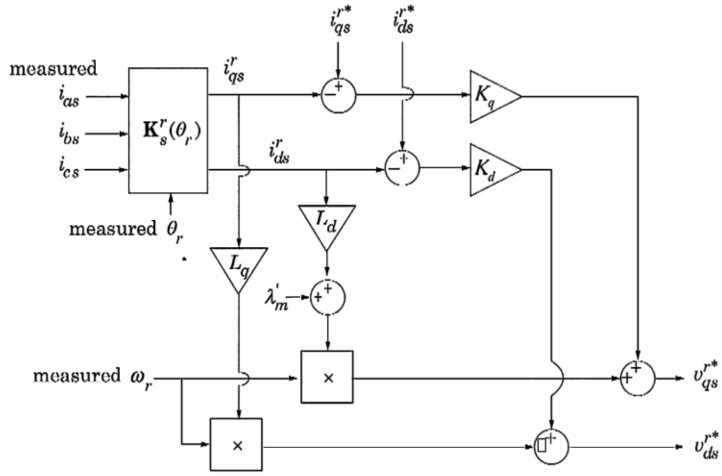


Figure 6 – Top level architecture of current controller subsystem

In the current control block, the measured currents from the stator are compared to the calculated currents $I_{qs}$* and $I_{ds}$*. The desired voltage $V_{qs}$ and $V_{ds}$ are obtained based on this comparison. Parks transformation is then applied to the measured currents $I_{as}$, $I_{bs}$, and $I_{cs}$ to obtain $I_{qs}$ and $I_{ds}$, which are dependent on the rotor position. The desired currents are compared to the measured currents in a feedback loop through the gains $K_q$ and $K_d$, which are kept equal for this project. The current control block outputs the desired $V_{qs}$ and $V_{ds}$. Figure 5 shows the current control block in Simulink and Figure 6 is a top-level diagram of the same.

$$V_{qs}^* = \omega_r L_d I_{ds}^* + \omega_r \lambda_m + K_q\left(I_{qs}^* - I_{qs}\right) = V_{qs} = R_s I_{qs}^* + L_q \frac{dI_{qs}}{dt} + \omega_r L_d I_{ds}^* + \omega_r \lambda_m \tag{3}$$

$$K_q\left(I_{qs}^* - I_{qs}\right) \approx R_s I_{qs}^* + L_q \frac{dI_{qs}}{dt} \tag{4}$$

If $K_q \gg R_s$, then $I_{qs} \approx I_{qs}^*$ In our case, $K_q = \mathrm{K_d} = 2$ and $R_s = 0.02\ \Omega$

## 2.2 Sine-triangle Modulator block

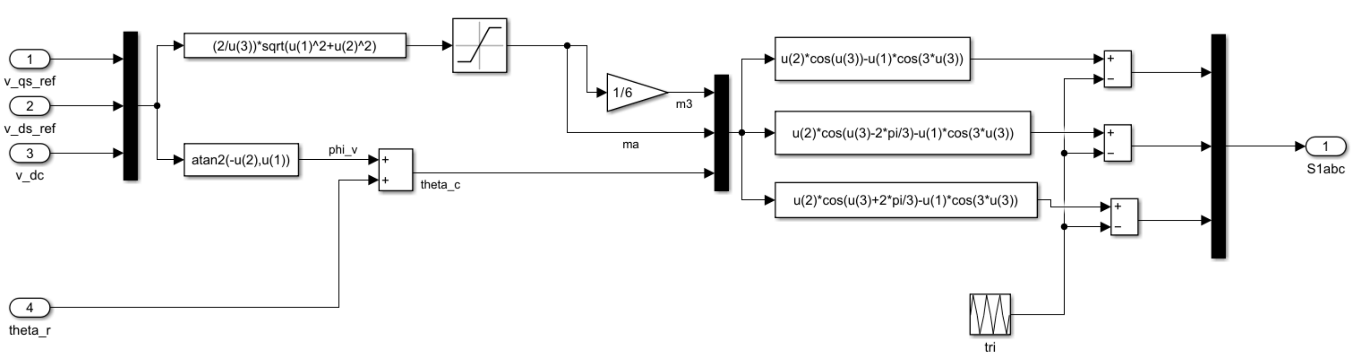


Figure 7 – Modulator subsystem in Simulink

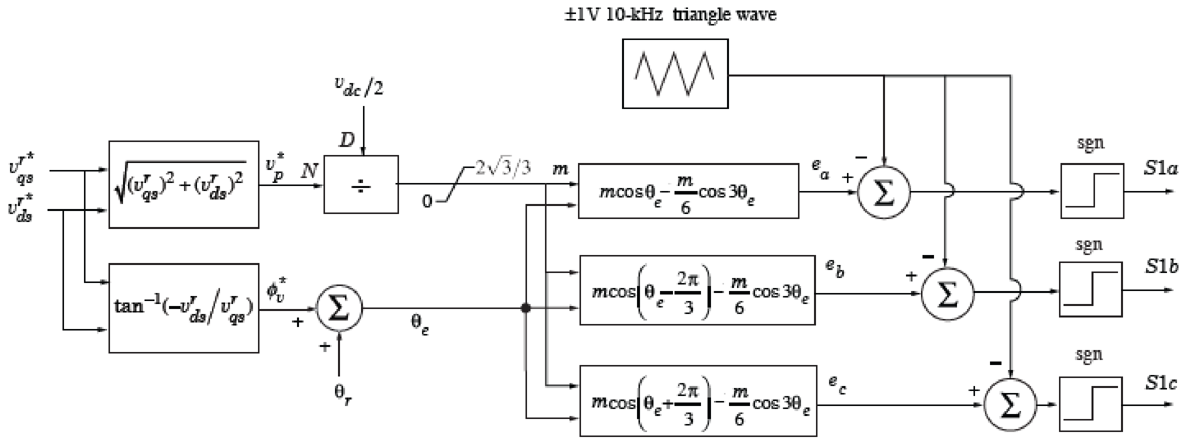


Figure 8 – Top level diagram of the modulator subsystem

A sine-triangle PWM modulator with third harmonic injection is used to convert the DC voltage $V_{dc}$ into three-phase AC line voltages $V_{an}$, $V_{bn}$, and $V_{cn}$ with higher amplitude. The commanded peak $V_p$ is obtained from the commanded $V_q$ and $V_d$ and is equal to $V_p = \sqrt{(V_d^2 + V_q^2)}$ and is compared to $V_{dc}/2$ The amplitude of the modulating signal $e_a$ or $e_b$ or $e_c$ is limited by $m$ and ranges between 0 and $m_{max} = 2\sqrt{3}/3$. The switching signals $S_{1a}$, $S_{1b}$, and $S_{1c}$ are obtained from the modulating signal. The frequency of the modulating signal can be calculated using the measured rotor position $\theta_r$ and $\phi_v = \arctan(V_{ds}/V_{qs})$. Figure 7 is the Simulink model of the sine-triangle modulator and figure 8 represents the top-level block diagram for the same.

For the sine-triangle PWM,

$$
\begin{gathered}
V_{s,max} = \sqrt{3}/3 \cdot V_{dc} \\
V_{an} = e_a \cdot \frac{V_{dc}}{2} \\
V_{bn} = e_b \cdot \frac{V_{dc}}{2} \\
V_{cn} = e_c \cdot \frac{V_{dc}}{2} \\
e_a = m\cos(\theta_e) - \frac{m}{6}\cos(3\theta_e) \\
e_b = m\cos\left(\theta_e - \frac{2\pi}{3}\right) - \frac{m}{6}\cos(3\theta_e) \\
e_c = m\cos\left(\theta_e + \frac{2\pi}{3}\right) - \frac{m}{6}\cos(3\theta_e)
\end{gathered}
\tag{5}
$$

Therefore,

$$
\begin{gathered}
V_{an}(t) = \frac{(e_a+1)}{2} \cdot V_{dc} = \frac{\left(\frac{2V_p}{V_{dc}}\left[\cos(\theta_e) - \frac{1}{6}\cos(3\theta_e)\right] + 1\right)}{2} \cdot V_{dc} \\
V_{bn}(t) = \frac{(e_b+1)}{2} \cdot V_{dc} = \frac{\left(\frac{2V_p}{V_{dc}}\left[\cos\left(\theta_e - \frac{2\pi}{3}\right) - \frac{1}{6}\cos(3\theta_e)\right] + 1\right)}{2} \cdot V_{dc} \\
V_{cn}(t) = \frac{(e_c+1)}{2} \cdot V_{dc} = \frac{\left(\frac{2V_p}{V_{dc}}\left[\cos\left(\theta_e + \frac{2\pi}{3}\right) - \frac{1}{6}\cos(3\theta_e)\right] + 1\right)}{2} \cdot V_{dc} \\
V_{ng} = \frac{1}{3}(V_{an} + V_{bn} + V_{cn})
\end{gathered}
\tag{6}
$$

## 2.3 Voltage Controlled Inverter block

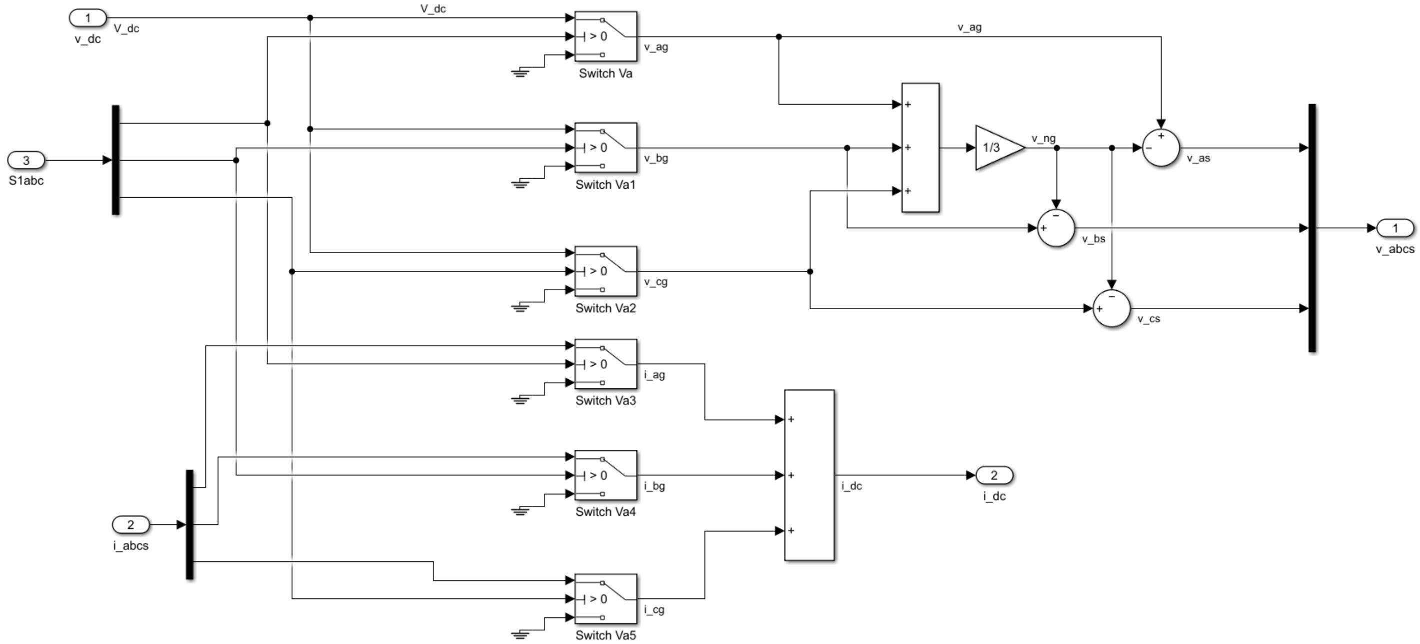


Figure 9 – Inverter subsystem in Simulink

After the switching is prepared for the inverter, the inputs vdc and switch $S_{1abc}$ are received by the inverter. When $V_{dc}$ passes through switch block $V_a$, $V_{ag}$ is obtained, taking values between $V_{dc}$ and zero. Similarly, $V_{bc}$ and $V_{cg}$ are obtained.

The vector $V_{abcs}$ is obtained by taking the multiplexer from the values of $v_{as}, v_{bs}, v_{cs}$

Similarly, the signal $I_{abcs}$ is taken as inputs by the inverter. After demuxing in MATLAB, the signals $I_{as}$, $I_{bs}$, and $I_{cs}$ are obtained. The currents $I_{ag}, I_{bg}, I_{cg}$ are obtained by passing these currents through the switch blocks $V_{a3}, V_{a4}, V_{a5}$ respectively. The output $I_{dc}$ is obtained by adding these currents. This means that when switch $V_{a3}$ is on, $I_{as}$ contributes to $I_{dc}$ Similarly, when switch $V_{a4}$ is on, $I_{bs}$ contributes to $I_{dc}$ and when switch $V_{a5}$ is on, $I_{cs}$ contributes to $I_{dc}$.

### 2.4 Filter Block

All harmonics due to the switching in $I_{dc}$ would cause losses and heat in the battery. The filter's job is to smooth the battery current. The capacitor acts as a stabilizer to the six pack's input voltage $V_{dc}$.
The equations inside the filter can be written as follows:
Applying Kirchhoff's Voltage Law (KVL):

$$-V_{batt} + R \cdot I_{batt} + \frac{L_d d(i_{batt})}{dt} + V_{dc} = 0 \tag{7}$$

$$P \cdot I_{batt} = V_{batt} - V_{dc} - R \cdot I_{batt} \tag{8}$$

Applying Kirchhoff's Current Law (KCL):

$$C \cdot \frac{dV_{dc}}{dt} = I_{batt} - i_{dc} \tag{9}$$

$$P \cdot V_{dc} = \frac{1}{C}(I_{batt} - i_{dc}) \tag{10}$$

Here, $V_{batt}$ is the battery voltage, $R$ is the resistance, $L_d$ is the inductance, $i_{batt}$ is the battery current, $V_{dc}$ is the DC-link voltage, $P$ is the power, $C$ is the capacitance, and $i_{dc}$ is the DC current.

## 3. Results and Discussion

To determine and plot the optimal $I_{qs}$ and $I_{ds}$ as a function of desired torque, a Matlab script is written to solve the optimization problem in equation (1). The objective function is denoted by *objfun* and the constraint function is denoted by *confuneq*

Initialization script

```
global param

% source and filter parameters
param.V_batt = 400; % battery voltage in Volt (V)
param.C = 2e-3; % capacitance in Farad (F)
param.L = 20e-6; % inductance in Henry (H)
param.R = 0.01; % resistance in Ohm

% motor parameters
param.L_d = 2e-3; % d-axis inductance in Henry (H)
param.L_q = 3.3e-3; % q-axis inductance in Henry (H)
param.r_s = 0.02; % stator resistance in ohm
param.lambda_m = 0.2; % flux constant V-s/rad
param.P = 8; % number of poles

% Bonus question
param.Is_max = 225; % maximum current to be passed in Ampere (A)
param.Vs_max = sqrt(3)*param.V_batt/3;

% Current Control gains
K_q = 2; % in ohms
K_d = 2; % in ohms



% load lookup table data
load I_qs
load I_ds
load T_e
```

Objective function

```
function i_s = objfun(iqd) % i_s is output that depends on input variable iqd
%global param
% define function to minimize
i_s = sqrt(iqd(1)^2 + iqd(2)^2) ;
end
```

Constraint function

```
function [c,ceq] = confuneq(iqd) % c and ceq are output variables that depend on input variable iqd
global param
% define constraints
% no nonlinear inequality constraints
c = [];


% Nonlinear equality constraint defined by ceq

% it is defined as ceq(1) because the constraint
% equation is assigned to the first element of ceq
% in case multiple constraints are to be defined,
% they can be assigned as ceq(2), ceq(3), etc
ceq(1) = param.Te - 1.5*(param.P/2)*(param.lambda_m*iqd(1)+(param.L_d-param.L_q)*iqd(1)*iqd(2));
end
```

Main script

```
% Question 1

N= 100;

% use interior-point algorithm
T_e = linspace(-400,400,N);
options = optimoptions('fmincon','Algorithm','interior-point');

for i = 1:N
    param.Te = T_e(i);
    iqd = fmincon(@(iqd)objfun(iqd),[0;0],[],[],[],[],[],[],@(iqd)confuneq(iqd),options);
    i_qs(i) = iqd(1);
    i_ds(i) = iqd(2);
end
I_s = sqrt(i_qs.^2+i_ds.^2);
save i_qs i_qs;
save i_ds i_ds;
save T_e T_e;

% Plot
figure(1)
plot(T_e,i_qs,T_e,-i_ds,T_e,I_s,'LineWidth',1)
legend('I_{qs}','-I_{ds}' ,'I_s')
title('Optimal Currents vs Torque')
ylabel('Current (A)')
xlabel('Electric Torque (N.m)')

% Question 2

% define electrical rotor speed (w_rm = 500 rpm)
omega_r = 500 * 2*pi / 60 * 8/2; % rad/s

% For T_e = 400
V_qs = param.r_s*i_qs(100) + omega_r*param.L_d*i_ds(100) + omega_r*param.lambda_m
V_ds = param.r_s*i_ds(100) - omega_r*param.L_q*i_qs(100)
P_elec = 1.5*(V_qs*i_qs(100) + V_ds*i_ds(100))
I_batt = P_elec/param.V_batt

% For T_e = -400
V_qs_neg = param.r_s*i_qs(1) + omega_r*param.L_d*i_ds(1) + omega_r*param.lambda_m
V_ds_neg = param.r_s*i_ds(1) - omega_r*param.L_q*i_qs(1)
P_elec_neg = 1.5*(V_qs_neg*i_qs(1) + V_ds_neg*i_ds(1))
I_batt_neg = P_elec_neg/param.V_batt
```

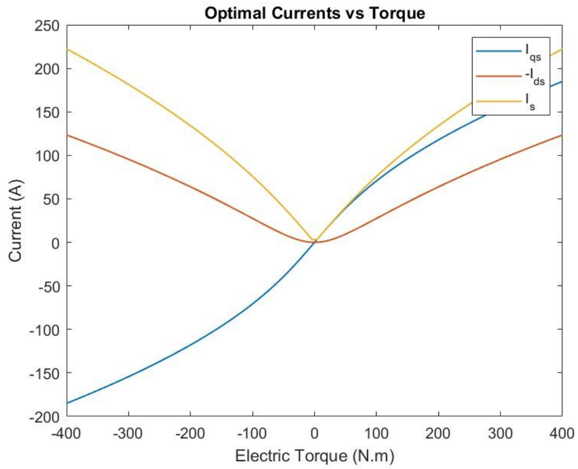


Figure 10- Optimal Currents v/s Electric Torque

Table 4 – Calculated values from Matlab for Te = 400 Nm and -400 Nm

| Variables | Values |
|---|---|
| $V_{qs}$ | -6.1034 V |
| $V_{ds}$ | -130.3086 V |
| $P_{elec}$ | 2.2427e+04 W |
| $I_{batt}$ | 56.0680 A |
| $V_{qs\_negative}$ | -13.5021 V |
| $V_{ds\_negative}$ | 125.3725 V |
| $P_{elec\_negative}$ | -1.9461e+04 W |
| $I_{batt\_negative}$ | -48.6518 A |

Figure 10 shows the optimal currents v/s electric torque figure generated after solving the optimization problem in Matlab. Table 4 shows the values of the variables calculated after the values of $I_{qs}$ and $I_{ds}$ are established. Assuming the mechanical rotor speed is 500 rpm, and the desired torque is 400 N.m, using Park's equation, the steady state values of $V_{qs}$ and $V_{ds}$ are calculated. The power supplied to the motor is denoted by $P_{elec}$. Knowing the battery voltage, the average steady state current is calculated and denoted by $I_{batt.}$. These calculations the repeated assuming the desired torque to be -400 Nm.

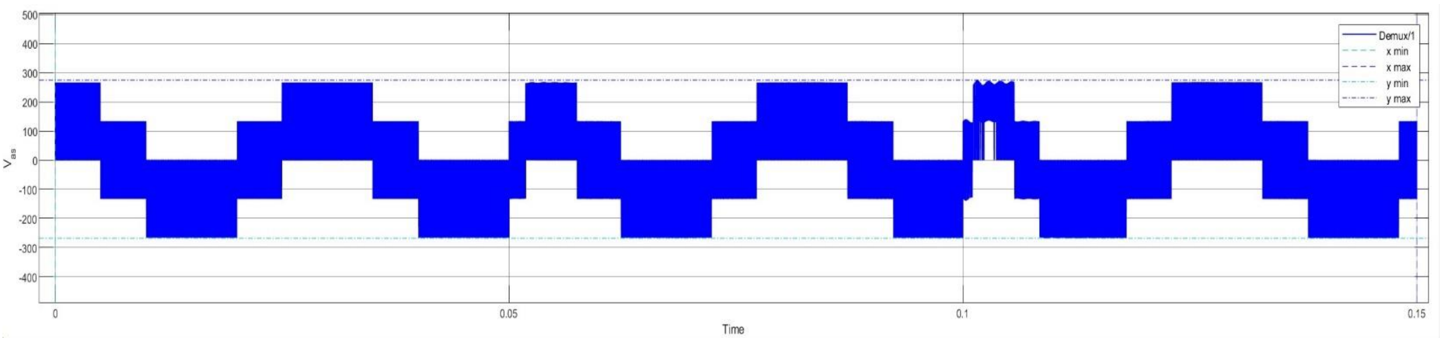


Figure 11- Vas v/s time

Figure 11 shows the plot for the stator voltage as a function of time. The minimum, maximum and mean values of the voltage are -268.6 V, 275 V and -6.934 V respectively. $V_{as}(t) = V_{qs}\cos(\theta_r) + V_{ds}\sin(\theta_r)$.

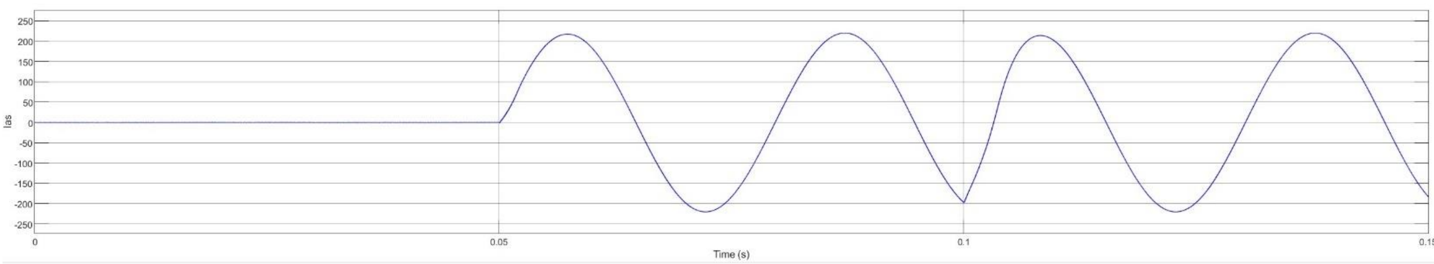


Figure 12- $I_{as}$ v/s time

Figure 12 shows the plot for the stator current as a function of time. The minimum, maximum and mean values for the currents are -220.6 A, 220.7 A and 14.89 A. $I_{\mathrm{as}}(\mathrm{t}) = I_{\mathrm{qs}}\cos(\theta_{\mathrm{r}}) + I_{\mathrm{ds}}\sin(\theta_{\mathrm{r}})$. The calculated values for $I_{\mathrm{as}}$ are 222.35 A. The peak values of $I_{\mathrm{as}}$ are close to the calculated values.

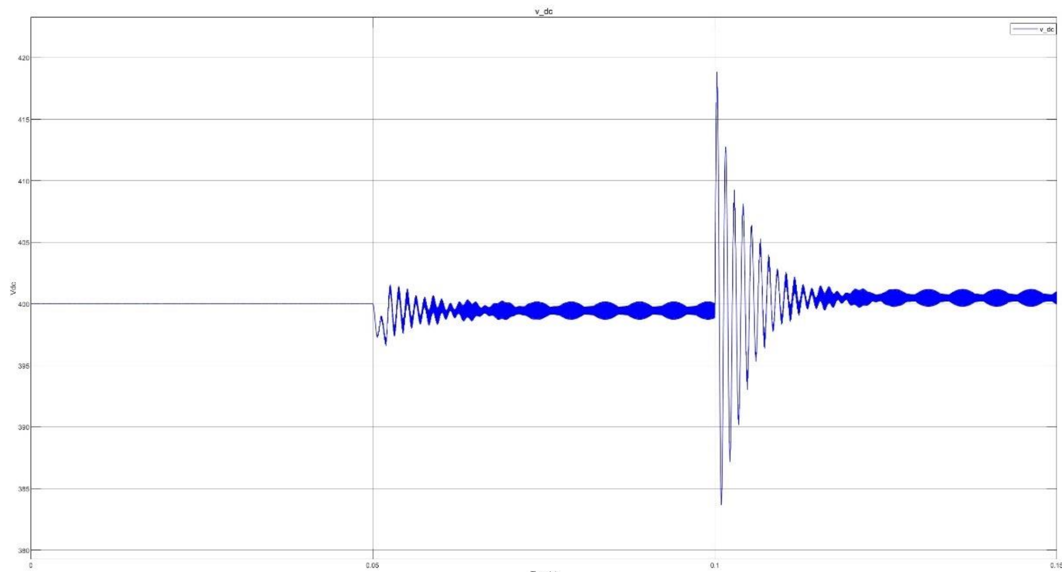


Figure 13- $V_{dc}$ v/s time

Figure 13 shows the simulated plot for $V_{dc}$ as a function of time. The minimum, maximum and average values for Vdc are 383.6 V, 418.9 V and 400 V.

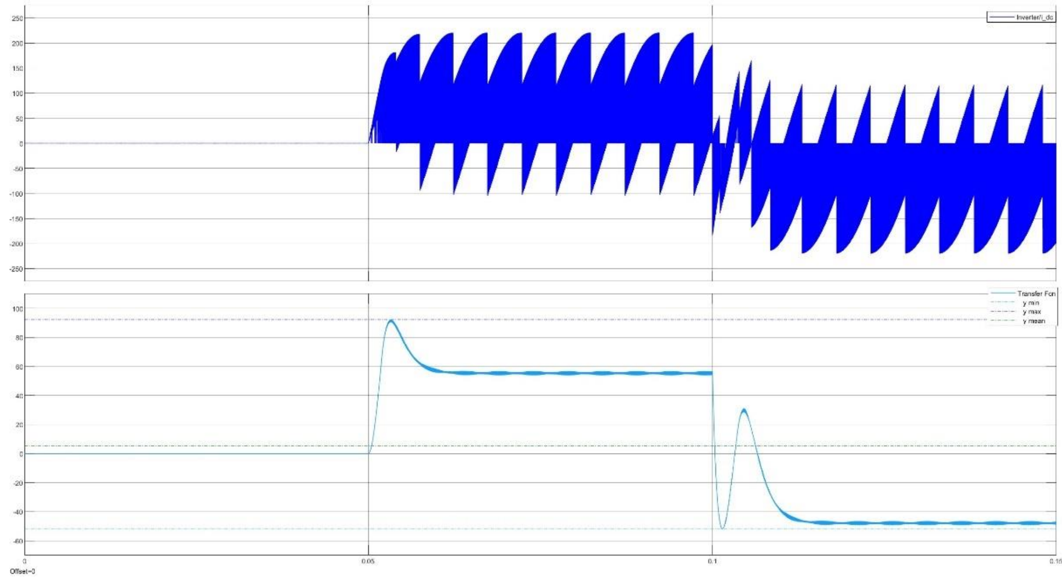

Figure 14- Idc v/s time

Figure 14 plots the DC current $I_{dc}$ v/s time. A transfer function block is used as a filter with time constant of 0.001 to smooth out $I_{dc}$ to find the average easily. The mean value is 5.412 A and the minimum and maximum current values are -51.99 A and 92.22 A respectively.

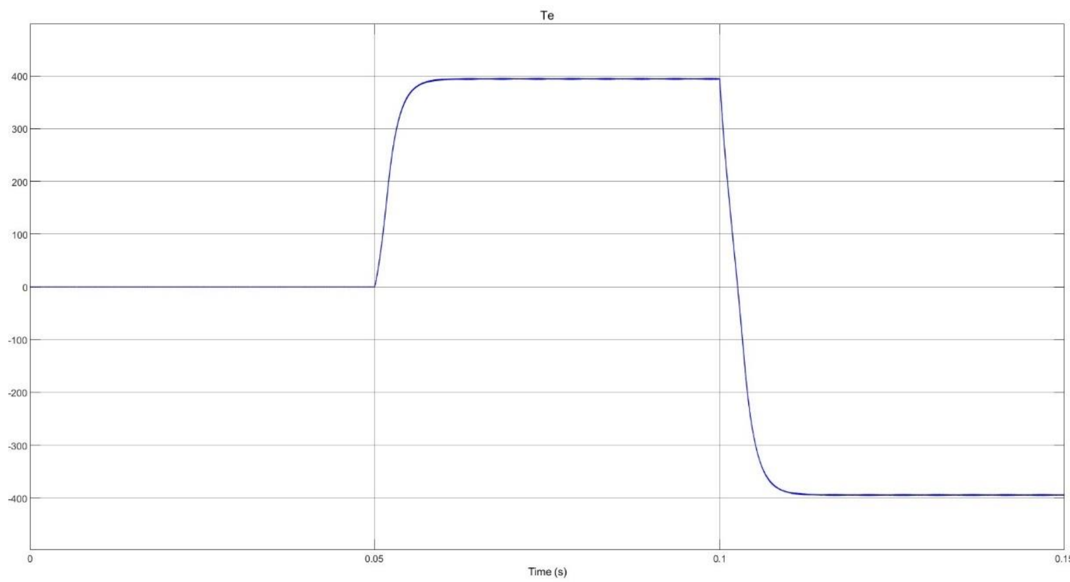


Figure 15- Te ( electrical torque) v/s time

As seen from figure 15, the Torque reaches its maximum value of 400 Nm in approximately 0.06 seconds or 60 milliseconds. Therefore, it can be shown that a permanent-magnet AC motor drive using a sine-triangle modulator with third harmonic injection can achieve the desired torque within 10’s of milliseconds.

## 4. Bonus

In Matlab, the following optimization problem to maximize the torque output of the PMAC is solved using the fmincon solver:

Maximize $$T_e = \frac{3}{2} \cdot \frac{P}{2}\left(\lambda'_m I_{qs} + \left(L_d - L_q\right) I_{qs} I_{ds}\right)$$

where $P$ is the number of poles, $\lambda'_m$ is the magnetic flux linkage, $I_{qs}$ and $I_{ds}$ the quadrature and direct components of the stator current respectively

The optimization problem is subject to two constraints: a current constraint and a voltage constraint. The current is given by

$\sqrt{I_{qs}^2 + I_{ds}^2} - I_{s,max} \leq 0$ where $I_{s,max}$ is the maximum allowable current. The voltage constraint is given by

$\sqrt{V_{qs}^2 + V_{ds}^2} - V_{s,max} \leq 0$ where $V_{qs}$ and $V_{ds}$ are the quadrature and direct components of the stator voltage, respectively, and $V_{s,max}$ is the maximum allowable voltage.

where $V_{qs} = R_s I_{qs} + \omega_r \lambda_m' + \omega_r L_d I_{ds}$ and $V_{ds} = R_s I_{ds} - \omega_r L_q I_{qs}$ where $R_s$ is the stator resistance, and $\omega_r$ is the electrical angular speed.

### Objective equation

```
function Torque = maxtorque(iqd)
global param

% define function to minimize
% maximizing torque equivalent to minimizing negative torque
Torque = -1.5*param.P/2*(iqd(1)*param.lambda_m + (param.L_d - param.L_q)*iqd(1)*iqd(2));
end
```

### Constraint equation

```
function [c,ceq] = myconstraint(iqd)
global param

% define constraints
v_q = param.r_s*iqd(1) + param.w_r*(param.lambda_m + param.L_d*iqd(2));
v_d = param.r_s*iqd(2) - param.w_r*param.L_q*iqd(1);

c(1) = sqrt(iqd(1)^2 + iqd(2)^2) - param.Is_max;
c(2) = sqrt(v_q^2 + v_d^2) - param.Vs_max;

ceq = []; % no equality constraints

end
```

## Main equation

```
% use interior point algorithm
N = 100;
omega_r = linspace(0,2000,N);
options = optimoptions('fmincon', 'Algorithm','interior-point');

for i = 1:N
    param.w_r = omega_r(i);
    iqd_forMaxT = fmincon(@(iqd_forMaxT)maxtorque(iqd_forMaxT),...
    [0;0],[],[],[],[],[],[],...
    @(iqd_forMaxT)myconstraint(iqd_forMaxT),options);
    Iqd_forMaxT(i,:) = iqd_forMaxT;
    Te_max(i,:) = -maxtorque(Iqd_forMaxT(i,:));

end

% Current calculation for maximum torque
Iq_forMaxT = Iqd_forMaxT(:,1);
Id_forMaxT = Iqd_forMaxT(:,2);
Is_forMaxT = sqrt(Iq_forMaxT.^2+Id_forMaxT.^2);

% Voltage calculation for maximum torque
Vq_forMaxT = param.r_s*Iq_forMaxT + param.w_r*(param.lambda_m + param.L_d*Id_forMaxT);
Vd_forMaxT = param.r_s*Id_forMaxT - param.w_r*param.L_q*Iq_forMaxT;
Vs_forMaxT = sqrt(Vq_forMaxT.^2+Vd_forMaxT.^2);

% Power calculation for maximum torque
```

```
P_forMaxT = (1.5*(Vq_forMaxT.*Iq_forMaxT + Vd_forMaxT.*Id_forMaxT))*0.001;

% plot torque vs mechanical speed
figure(1)
plot(omega_r,Te_max,LineWidth=1)
legend('Te,max')
title('Maximum Torque versus Mechanical speed')
ylabel('Maximum Torque in N-m')
xlabel('Mechanical Speed in rad/s')

% plot currents vs mechanical speed
figure(2)
plot(omega_r,Is_forMaxT,omega_r,Iq_forMaxT,omega_r,-Id_forMaxT,LineWidth=1)
legend('Is','Iq','-Id')
title('Currents needed for Maximum Torque versus Mechanical speed')
xlabel('Mechanical Speed in rad/s')
ylabel('Currents in Amperes')

% plot voltage vs mechanical speed
figure(3)
plot(omega_r,Vs_forMaxT,LineWidth=1)
legend('Vs')
title('Voltage needed for Maximum Torque versus Mechanical speed')
xlabel('Mechanical Speed in rad/s')
ylabel('Voltage in Volts')

% plot power vs mechanical speed
figure(4)
plot(omega_r,P_forMaxT,LineWidth=1)
legend('Pelec')
title('Power needed for Maximum Torque versus Mechanical speed')
xlabel('Mechanical Speed in rad/s')
ylabel('Power in kW')
```

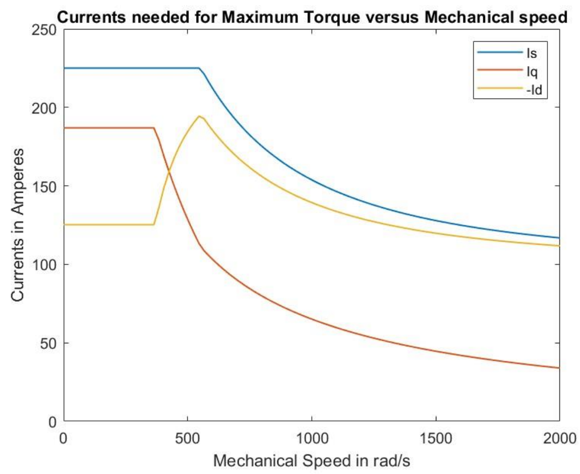


Figure 16

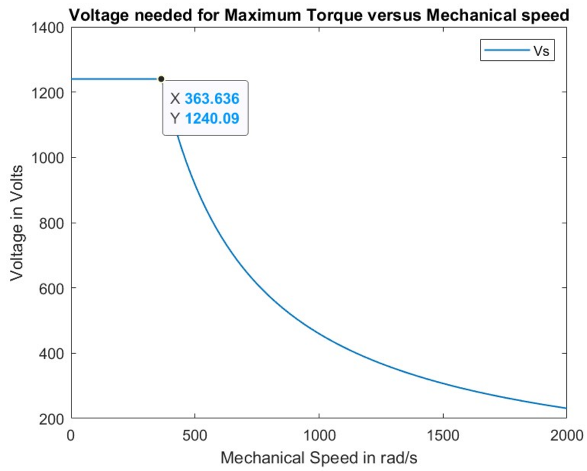


Figure 17

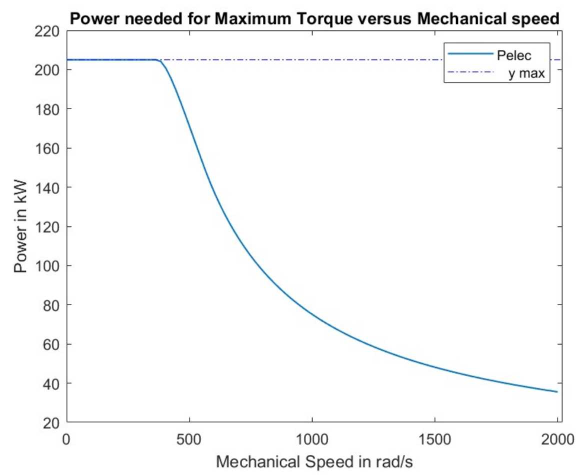


Figure 18

The currents needed for maximum torque vs mechanical speed is plotted in Figure 17. Figure 18 shows the voltage v/s mechanical speed plot. For a rotor speed of 363.636 rad/sec, voltage constraint starts to limit the maximum torque. Figure 18 shows the power needed for maximum torque v/s mechanical speed. The maximum power is 205 kW.