CHAPTER 4

SENSORLESS SPEED CONTROL OF PMBLDC MOTOR

4.1 PRINCIPLE OF CONTROL LAWS

The control objective is to bring the speed of the PMBLDC motor to desired value by varying the voltage input, when there is variation in speed due to servo operation (set point change) as well as regulatory operation (disturbance rejection). Dixon and Leal (2002) reported that precise determination of the rotor angle is not necessary to control the speed of PMBLDC motor. Only the position of commutation points is required to achieve quasi-square current, with dead time periods of 60°. Hence, in the proposed sensorless speed control of PMBLDC motor, the position of the rotor is evaluated by measuring the phase currents.

The inverter supplies a quasi-square current, whose magnitude $I_{\text{MAX}}$ is proportional to the machine shaft torque. Thus, by controlling the phase currents, the torque and speed are adjusted. There are two ways to control the phase currents of a PMBLDC motor. Accordingly, the two control laws used in the present work are shown in Figure 4.1
Figure 4.1 Schematic Diagram of Control laws

In both the control laws, absolute values of the two of the three phase currents are rectified and the dc component corresponding to the amplitude $I_{\text{MAX}}$ of the original phase current is obtained. The circuit used to find $I_{\text{MAX}}$ from the original phase current is shown in Figure 4.2. The dc component corresponding to the amplitude $I_{\text{MAX}}$ is compared with a reference current $I_{\text{REF}}$. The error signal thus obtained is processed through a controller.

\[
\begin{align*}
R &= 1\, \text{K}\Omega \\
C &= 1000\ \mu\text{F} \\
D_1, D_4 &= \text{IN4007} \\
R_{\text{on}} &= 0.001\ \Omega \\
K_1 &= 0.001
\end{align*}
\]

Figure 4.2 Simulation circuit to find the amplitude $I_{\text{MAX}}$
4.2 DESIGN OF PI CONTROLLER

The DC input voltage is applied to the three phase inverter as in Figure 4.3. The gate signal for the inverter is obtained by sensing the position of the rotor. The simulation is carried out to study the variation of speed for the variation in DC input voltage.

4.2.1 Open loop response of VSI fed PMBLDC Motor

Figure 4.3 shows the simulation circuit used for carrying out open loop studies of VSI fed PMBLDC motor. The input is the DC voltage in volts and the output is speed in rpm.

![Open loop simulation circuit of VSI fed PMBLDC motor](image)

Figure 4.3  Open loop simulation circuit of VSI fed PMBLDC motor

Figure 4.4 shows the open loop speed response of the VSI fed PMBLDC motor. Figure 4.4(a) shows the DC voltage applied to the three phase inverter. Figure 4.4(b) shows the line to line voltage applied to the stator coil of PMBLDC motor. Figure 4.4(c) gives the response of stator phase current. Figure 4.4(d) shows the speed response. It is clearly seen in Speed curve that there is an overshoot and it resembles the second order response.
Figure 4.4  Open loop responses of PMBLDC motor for step change in Voltage Input
(a) Variation in DC voltage
(b) Variation of line to line voltage
(c) Variation of Stator phase current
(d) Variation of Speed
By using the open loop simulation results of input and output variables of PMBBLDC motor, its parameters were identified using system identification technique and are presented in Table 4.1.

**Table 4.1 PMBBLDC Motor Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>549.9</td>
</tr>
<tr>
<td>$K_m$</td>
<td>6.065</td>
</tr>
<tr>
<td>D</td>
<td>29.5</td>
</tr>
</tbody>
</table>

Thus, from the parameters in Table 4.1 transfer function of PMBBLDC motor is approximated as first order with dead time.

### 4.2.1.1 Velocity form of PI algorithm

In order to carryout closedloop studies the PI controller is represented in velocity form. Anti-reset windup, bumpless transfer, protection of system under computer failure are the inherent advantages of velocity form of PI control algorithm. For PI controller, the output of the controller is given by

$$u(t) = k_p e(t) + \frac{k_p}{T_i} \int e(t) \, dt + u_0$$

(4.1)

where $u(t)$ = controller output, $e(t)$ = error, $u_0$ = initial value of controller output $k_p$ = controller gain, $T_i$ = integral time constant.

$$U(kT) = k_p e(kT) + \frac{k_p}{T_i} \left[ \sum_{i=1}^{k} e(iT) \right] + U_0$$

(4.2)
After discretizing Equation (4.1),

\[ U(k+1)T = k_p e(k+1)T + \frac{k_p}{T_i} \left[ \sum_{t=1}^{k+1} e(\alpha T)T \right] + U_0 \]  

(4.3)

Similarly we can write

\[ U(k+1)T - U(kT) = k_p e(k+1)T - k_p e(kT) + \frac{k_p}{T_i} e(k+1)T \]  

(4.4)

Subtracting Equation (4.3) from (4.2)

\[ U(kT) - U(k-1)T = k_p [e(kT) - e(k-1)T] + \frac{k_p}{T_i} e(kT)T \]  

(4.5)

Equation (4.4) can also be written as follows

\[ \Delta u_k = k_p \left[ 1 + \frac{T}{T_i} \right] e_k - k_p e_{k-1} \]  

(4.6)

In the present work from the identified motor parameters it is clear that \( T_m >> T_d \). When the value of \( T_m >> T_d \), Oppelt (1951) proposed following tuning method for determining controller parameters. Accordingly, the controller parameters \( k_p \) and \( k_i \) are determined using Equations (4.7) and (4.8).

\[ k_p = \frac{1}{K_m} \left[ 0.77 \frac{T}{D} - 1 \right] \]  

(4.7)

\[ k_i = \frac{k_p}{T_i} \]  

(4.8)

where \( T_i = 3.32 \tau \)
The controller parameters thus obtained are presented in Table 4.2.

### Table 4.2 Controller Parameters

<table>
<thead>
<tr>
<th>Controller Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_p$</td>
<td>2.2</td>
</tr>
<tr>
<td>$k_i$</td>
<td>0.02246</td>
</tr>
</tbody>
</table>

In the present work speed control is carried out using sensorless method. Thus without using position sensors, the commutation points were identified from the stator phase currents. By controlling the stator current the speed and torque of the motor is controlled. Hence, it is required to design the current controller gain.

#### 4.2.1.2 Current Controller gain

In order to determine the gain of the current controller close attention was paid to speed and torque variations. The speed loop time constant is greater than the current loop time constant. Therefore, the speed loop time is assumed to be constant during each of the current control loop sampling interval. Hence, by assuming the settling time for current loop as 1ms the values of $k_{pc}$ and $k_{ic}$ are determined for a given PMBLDC motor specifications. The gain $k_{pc}$ is determined from Equation (4.9).

$$1\, ms = \frac{L'}{k_{pc}}$$  \hspace{1cm} (4.9)

where $L'$ is the inductance of the stator phase coil.

The choice of the switching frequency is guided by the limitations of the hardware. Higher the switching frequency, the lower will be the ripple in current. This results in smooth torque. Also, above 20 kHz, the switching
frequency starts being inaudible, hence lowers the sonic pollution. Hence a switching frequency of 20 kHz was chosen to calculate the sampling time. The gain $k_{ic}$ is calculated from Equation (4.10).

$$k_{ic} = \frac{k_p \times \text{sampling time}}{T_i}$$

(4.10)

Using the Equations (4.15) and (4.16) the current controller settings $k_{pc}$ and $k_{ic}$ for the given PMBLDC motor with stator resistance of 18.7 ohms and stator inductance of 0.02682H are calculated.

$$k_{pc} = 53.64$$

$$k_{ic} = 2.682$$

These controller parameters are used in current control loop to control the speed of PMBLDC motor.

4.3 VOLTAGE CONTROL BASED SPEED CONTROL OF PMBLDC MOTOR

4.3.1 Closed loop response using PI controller

The simulink block diagram for voltage control based speed control of PMBLDC motor using PI controller for set point tracking is shown in Figure 4.5. The dc component corresponding to the amplitude $I_{MAX}$ of the original phase current is compared with $I_{REF}$ and the error thus obtained is processed through PI controller. The output of the PI controller is applied to controlled voltage source to vary the input voltage applied to the inverter. The output of the inverter is fed as stator phase voltage. By controlling the stator phase voltage, the stator phase current can be controlled thereby controlling the speed of a motor.
Figure 4.5 Simulink Block Diagram of Voltage Control Based Speed Control of PMBLDC Motor Using PI Controller Scheme
The Stator phase currents are quasi-square currents as shown in Figure 4.6 (a). The commutation sequence is identified from the stator phase current as shown in Figure 4.6(b).

![Figure 4.6](image)

**Figure 4.6 (a) Three Phase Stator Currents of a PMBLDC Motor**

**Figure 4.6 (b) Commutation Sequence for a PMBLDC Motor**

The logical circuit to turn ON MOSFET is designed to identify the commutation point from the stator phase current based on the slope of \( I_{\text{MAX}} \) as shown in Figure 4.7. The sudden change in the phase back EMF will also produce a sudden change in the slope of \( I_{\text{MAX}} \) and this point indicates the instant of commutation. The electronic commutator outputs based on stator phase currents is presented in Table 4.3. When the stator current \( I_a \) is entering current and \( I_c \) is leaving current then the switches \( S_1 \) and \( S_6 \) are in ON condition. Thus, only two out of the three phases are in conduction state at any instant. Hence, by identifying the commutation points from two of the three phase stator current, the voltage applied to the stator coil varies thus controlling the speed of PMBLDC motor.
Table 4.3 Electronic Commutator Output Based on Stator Phase Currents

<table>
<thead>
<tr>
<th>$I_a$</th>
<th>$I_b$</th>
<th>$I_c$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>+1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The closed loop response of PMBLDC motor with PI controller for set point changes in speed is shown in Figure 4.8. The variation in actual speed of PMBLDC motor for changes in set values is shown. When the set speed is changed from 0 to 200 rpm and then to 1500 rpm, the motor speed also reaches the set value due to PI controller action.

The corresponding variation of line voltage is applied to the stator coil of PMBLDC motor as in Figure 4.8. At $t = 5.0$ seconds the motor speed
is increased from 200 rpm to 1500 rpm. It is clearly seen from Figure 4.8 that there is an increase in line voltage at $t = 5.0$ seconds due to controller action and the rotor reaches the speed 1500 rpm at $t = 6.3$ seconds.

Figure 4.8 Closedloop response of PMBLDC motor with PI scheme for setpoint Tracking

The design of conventional (PI) controller is based on the mathematical model with known parameters. The power electronic system models are often ill-defined and are subjected to parameter variation. Hence, speed control using FLC method is preferred. The FLC scheme does not need a mathematical model of the motor and it has the inherent ability to cope up with parameter variation. Hence, the FLC scheme is preferred over conventional PI controller for the speed control of PMBLDC motor.
4.3.2 Voltage Control Method using FLC Scheme

4.3.2.1 Fuzzy Logic Controller Design

Fuzzy logic uses simple rules to describe the system, rather than analytical equations, making it easy to implement. The block diagram of Fuzzy logic control scheme is shown in Figure 4.9. The principle components of Fuzzy logic controller include: Fuzzification, Rule base, Decision making logic and Defuzzification.

![Figure 4.9 Block Diagram of Fuzzy Logic Control scheme](image_url)

Fuzzification converts the input data into suitable linguistic values. For the FLC scheme based PMBLDC speed control, the error $e(t)$ and change of error $ce(t)$ are used as input variables. The error $e(t)$ ranging from $+5A$ to $-5A$ is converted into seven linguistic levels namely Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Similarly the second fuzzy input variable namely $ce(t)$ ranging from $-3.6A$ to $3.6A$ and the output variable speed $u(t)$ ranging from $-1500$ rpm to $1500$ rpm are converted into seven linguistic levels same as that used for $e(t)$.

The error is defined in Equation (4.11) as

$$e(k) = w(k) - y(k) \quad (4.11)$$
where $e(k)$ is the error

$w(k)$ is the Ref current

$y(k)$ is the Actual current

The change in error is defined in Equation (4.12)

$$ce(k) = e(k) - e(k-1)$$ (4.12)

A scale mapping is performed using the triangular membership function which transfers the range of input variables into corresponding universe of discourse. The elements of each of the sets are mapped on to the domain of corresponding linguistic variables.

Figure 4.10 Membership Function for error, change of error and Output Variable in FLC Scheme Based on Voltage Control Method
Care is taken to ensure that the cross point level for every membership function with degree of membership, is greater than zero. This means that every crisp value belongs to at least one membership value greater than zero. If this is not the case, no rule will fire and there will not be any control action. The membership function for input variables and output variable are shown in Figure 4.10.

Rule Base

The rule base is constructed using expert knowledge and experience. The rules are expressed in the form of following syntax:

IF < Fuzzy Preposition > THEN < Fuzzy Preposition >.

‘IF’ part is called antecedent. ‘THEN’ part is called consequent. In the present work $e(t)$, $ce(t)$ are antecedents and control command is the consequent. The combination is called the premise. The rules are generated heuristically from the response of the conventional controller. In the present work, 49 rules are derived from the analysis of trend obtained from the simulation results obtained from the PI controller scheme and are presented in Table 4.4 and 4.5.

If both error and change in error are zero, then maintain the present control setting. If error is not zero but the output is approaching the satisfactory rate, then maintain the present control setting. If error is growing, then change the control signal depending on the magnitude and sign of error and change in error, to force error towards zero.
Table 4.4   Rule Base Matrix for Voltage Control Based Speed Control of PMBLDC Motor Using FLC Scheme

<table>
<thead>
<tr>
<th>e(k)/ ce(k)</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>Z</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

NB (Negative Big), NM (Negative Medium), NS (negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium), PB (positive Big).

The inference of Rule Base Matrix is given in Table 4.5

Table 4.5 Inference of Rule Base Matrix

<table>
<thead>
<tr>
<th>Zone</th>
<th>Rule Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>e(k) ≤ 0 and ce (k) ≤ 0 implies set point ≤ output. Therefore error is not self correcting. The magnitude of controller output changes based on the magnitude of e(k) and ce(k).</td>
<td></td>
</tr>
<tr>
<td>e(k) &lt; 0 and ce (k) &gt; 0 implies set point &lt; output. Therefore error is self correcting and change in output is almost zero, that is control variable remains at its present setting.</td>
<td></td>
</tr>
<tr>
<td>e(k) &gt; 0 and ce (k) &lt; 0 implies set point &gt; output. Therefore error is self-correcting and change in output is almost zero that is control variable remains at its present setting.</td>
<td></td>
</tr>
<tr>
<td>e(k) ≥ 0 and ce(k) ≥ 0 implies set point ≥ output. Therefore error is not self correcting. The magnitude of controller output changes based on the magnitude of e(k) and ce(k).</td>
<td></td>
</tr>
<tr>
<td>Both e(k) and ce(k) is closer to zero. The system is in steady state. The change in controller output is zero. That is the present control setting is maintained.</td>
<td></td>
</tr>
</tbody>
</table>
**Decision making logic**

The decision making logic, infers a system of rules through fuzzy operators namely ‘AND’ and ‘OR’ for finding ‘minimum’ and ‘maximum’ values respectively. In other words, ‘Max-Min’ criterion is used to combine the results to generate single truth value, which determines the outcome of the rules. Thus, the outcome of the decision making logic is the inferred fuzzy control action. In the present work, the single truth value is obtained using Max-Min criteria.

**Defuzzification**

The output of the rule base is converted into crisp value, using defuzzification module. The commonly used defuzzification methods are i) Max criteria ii) Mean of maximum iii) Center of Area method. The Defuzzification using Center of Area is considered for this application as it produces the results which are sensitive to all the rules and the output moves smoothly across the control surface which can reduce the wear and tear of final control element. The Table 4.6 shows the FLC design parameters for the voltage control based speed control of PMBLDC motor. The crisp output

$$z_o = \frac{\sum_{j=1}^{n} \mu_2(W_j)(W_j)}{\sum_{j=1}^{n} \mu_2(W_j)}$$

(4.13)

$\mu_2(W_j) = \text{Maximum value of the membership function corresponding to } j^{\text{th}} \text{ quantisation level.}$

where $j = 1 \text{ to } n \text{ is the number of quantisation levels}$

$W_j = \text{support value at which the membership function reaches maximum value.}$
Table 4.6 FLC Design Parameters for the Voltage Control Based Speed Control of PMBLDC Motor

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input variables</td>
<td>2</td>
</tr>
<tr>
<td>Output variable</td>
<td>1</td>
</tr>
<tr>
<td>linguistic levels</td>
<td>7</td>
</tr>
<tr>
<td>Rules</td>
<td>49</td>
</tr>
<tr>
<td>Membership Function</td>
<td>Triangular</td>
</tr>
<tr>
<td>Decision making logic</td>
<td>Max-Min criteria</td>
</tr>
<tr>
<td>Defuzzification</td>
<td>Centre of Area method</td>
</tr>
</tbody>
</table>

4.3.2.2 Closed loop simulation using FLC Scheme

After tuning FLC scheme by adjusting the membership function widths and the rule sets, its performance for set point change in speed is studied. The closed loop response of PMBLDC motor with FLC scheme for set point changes in speed is shown in Figure 4.11. The variation in actual speed of PMBLDC motor for changes in set values is shown. When the set speed is changed from 0 to 200 rpm and then to 1500 rpm, the motor speed also reaches the set value due to FLC action.

The corresponding variation of line voltage is applied to the stator coil of PMBLDC motor as in Figure 4.11. At $t = 5.0$ seconds the motor speed is increased from 200 rpm to 1500 rpm. It is clearly seen from Figure 4.11 that there is an increase in line voltage at $t = 5.0$ seconds due to controller action and the rotor reaches the speed 1500 rpm at $t = 6.0$ seconds.
Due to the presence of parameter variations in a PMBLDC motor, adaptation of PI control settings becomes necessary to obtain a reasonable behavior of a closed loop system. Moreover it is difficult to know the exact parameters of a PMBLDC motor with different mechanical loads. Hence, it becomes necessary to introduce the fuzzy algorithm to tune the PI controller thus forming a hybrid control scheme.

4.3.3 FUZZY TUNED PI CONTROLLER

4.3.3.1 Design of Fuzzy tuned PI controller parameters

In the Fuzzy tuned PI controller scheme based PMBLDC speed control, the error $e(t)$ and change of error $ce(t)$ are used as fuzzy input variables. These two fuzzy input variables and the output variable (speed)
are converted into seven linguistic levels namely Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). The symmetrical triangular membership function transfers the range of input variables into corresponding universe of discourse. The rule base is constructed using expert knowledge. The outcome of the decision making logic infers the fuzzy control action. The single truth value is obtained using Max-Min criteria. The defuzzified (crisp) value is obtained using Center of Area method.

The e(k), ce(k) and u(k) takes the values from the operating ranges 

[-ae, ae], [-ac, ac], [-au, au]. The normalised universe of discourse is given in Equation (4.19) for input variables and output variable as X1, X2 and Y respectively (Ronald R. Yager and Dimitar P. Filev, Theoretical Analysis of FLC)

\[ X_1 = [-a_c^*, a_c^*], X_2 = [-a_e^*, a_e^*], Y = [-a_u^*, a_u^*] \] (4.14)

To transform the operating ranges of the measured variables e(k) and ce(k) to the normalised universes X1 and X2 means to scale them with the scaling factors as in Equations (4.15) and (4.16).

\[ k_e = \frac{a_e^*}{ae} \] (4.15)

\[ k_c = \frac{a_c^*}{ac} \] (4.16)

Assuming linear mapping e*(k) and ce*(k) is given in Equations (4.17) and (4.18)

\[ e^*(k) = k_e e(k) \] (4.17)

\[ ce^*(k) = k_c ce(k) \] (4.18)
Defuzzified value $\Delta u^*(k)$, obtained by the application of the FLC algorithm, belongs to the normalised universe $Y = [-a_u^*, a_u^*]$ and related to the real change of control variable $\Delta u(k)$ from the operating range $[-a_u^*, a_u^*]$ through the scaling factor $k_u$ as in Equation (4.19).

\[
k_u = \frac{a_u^*}{a_u^*} \tag{4.19}
\]

By applying the normalised value of the input and output values $e^*(k)$, $ce^*(k)$ and $\Delta u^*(k)$ of the input and output variables, we get

\[
\Delta u^*(k) = k_p ce^*(k) + k_i e^*(k) \tag{4.20}
\]

Using the scaling factor, the Equation (4.20) becomes

\[
\frac{\Delta u(k)}{k_u} = k_c ce(k)k_c + k_e e(k)k_e \tag{4.21}
\]

Simplifying the Equation (4.21)

\[
\Delta u(k) = k_p ce(k) (k_c k_u) + k_i e(k)( k_c k_u) \tag{4.22}
\]

Thus FLC is approximated as PI tuned FLC controller as given in Equation (4.23)

\[
\Delta u(k) = K_p ce(k) + K_i e(k) \tag{4.23}
\]

Where $K_p = k_p k_c k_u$

$K_i = k_i k_c k_u$

The Table 4.7 gives the scaling factor for Fuzzy tuned – PI controller.
i. Increasing or decreasing of the scaling factor $k_c$ causes increase or decrease of the parameter $K_p$.

ii. Increasing or decreasing of the scaling factor $k_c$ causes increase or decrease of the parameter $K_i$.

iii. Scaling factor $k_u$ determines the gain of the PI controller.

**Table 4.7  Scaling factors for Fuzzy tuned PI controller**

<table>
<thead>
<tr>
<th></th>
<th>$k_c$</th>
<th>$k_c$</th>
<th>$k_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/5</td>
<td>1/3.6</td>
<td>1/1500</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.12 shows the Membership functions of the normalised fuzzy sets associated with the term sets of the error, its change and Output control Variable with the scaling factors $K_c=1/5$, $K_c=1/3.6$ and $K_u=1/1500$ respectively.

![Membership functions](image)

**Figure 4.12** Membership functions of the normalised fuzzy sets associated with the error, its change and Output control Variable in Fuzzy tuned PI controller
4.3.3.2 Closed loop simulation using Fuzzy tuned PI control scheme

The closed loop response of PMBLDC motor with Fuzzy tuned PI controller for set point changes in speed is shown in Figure 4.13. The variation in actual speed of PMBLDC motor for changes in set values is shown. When the set speed is slowly changed from 0 to 200 rpm and then to 1500 rpm, the motor speed also reaches the set value due to PI controller action.

The corresponding variation of line voltage applied to the stator coil of PMBLDC motor is as in Figure 4.13. At $t = 5.0$ seconds the motor speed is increased from 200 rpm to 1500 rpm. It is clearly seen from Figure 4.13 that there is an increase in line voltage at $t = 5.0$ seconds due to controller action and the rotor reaches the speed 1500 rpm at $t = 5.5$ seconds.

![Figure 4.13 Closed loop response of PMBLDC motor with Fuzzy tuned PI controller scheme for set point Tracking](image)
The simulation results of stator phase current using Fuzzy tuned PI controller is shown in Figure 4.14. The stator currents are quasi square waves with a displacement of 120°.

![Figure 4.14 Three Phase Stator Current of PMBLDC Motor Based on Voltage Control Method Using Fuzzy Tuned PI Controller](image)

The phase back emf is trapezoidal in nature as shown in Figure 4.15 and is the function of the speed and rotor position angle. The phase back EMF are displaced by 120° with each other.

![Figure 4.15 Three Phase Stator Back EMF of PMBLDC Motor using Fuzzy Tuned PI Controller](image)
The Table 4.8 gives the comparison of various performance measures based on voltage control method using PI, FLC and Fuzzy Tuned PI controller scheme. It is clear that the Fuzzy Tuned PI controller shows improved performance in all aspects compared to conventional PI controller and FLC schemes.

**Table 4.8 Performance Measures Based on Voltage Control Method Using PI, FLC and Fuzzy Tuned PI Controller**

<table>
<thead>
<tr>
<th>Performance measures</th>
<th>PI</th>
<th>FLC</th>
<th>Fuzzy Tuned PI Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling time (Ts) in seconds</td>
<td>6.3</td>
<td>6.0</td>
<td>5.5</td>
</tr>
<tr>
<td>RMS current in amperes</td>
<td>3.5</td>
<td>3.76</td>
<td>3.98</td>
</tr>
<tr>
<td>IAE</td>
<td>42.79</td>
<td>28.39</td>
<td>10.49</td>
</tr>
<tr>
<td>ISE</td>
<td>334</td>
<td>107</td>
<td>29.35</td>
</tr>
<tr>
<td>THD in %</td>
<td>0.5747</td>
<td>0.4173</td>
<td>0.3825</td>
</tr>
</tbody>
</table>

The voltage control based speed control of PMBLDC motor method requires a variable or controlled voltage source, which increases the complexity of the circuit. Hence, Gate control based speed control of PMBLDC motor (PWM switching control) is preferred.

4.4 **GATE CONTROL BASED SPEED CONTROL OF PMBLDC MOTOR**

4.4.1 **Gate Control Method using PI Controller**

In this method, the variable voltage to the stator coil is obtained by varying the duty cycle of PWM signal. The output voltage of the inverter is controlled by means of gate control. The PWM signal applied to the gate terminal of inverter is varied to maintain the required speed.
The dc component corresponding to the amplitude $I_{MAX}$ of the original phase current is compared with $I_{REF}$ and the obtained error is processed through PI controller. The output of the PI controller changes the duty cycle of the PWM signal. The inverter receives the gate pulses from the PWM Gate block. The output of the inverter is fed as stator phase voltage. By controlling the stator phase voltage, the stator phase current varies, thus controlling the speed of a motor.

Similar to the voltage control method, the position sensing for a PMBLDC motor needs to detect six positions, which determine the commutation points. The commutation sequence is identified from the stator phase current. The PWM gate control block receives two inputs, switching signal from decoder and duty cycle generated from the PI controller block. Based on the controller output the width of the pulses changes, which in turn varies the stator voltage of the PMBLDC motor. The motor parameters are identified from the open loop simulation results of input and output variables of PMBLDC motors. The controller parameters $k_p$ and $k_i$ were identified using system identification technique.

The reference dc link current at $k^{th}$ instant is given by Equation(4.24)

$$I_{REF}(k) = I_{REF}(k-1) + k_p(e_k - e_{k-1}) + k_i e_k$$  \hspace{1cm} (4.24)

Where $e_k$ and $e_{k-1}$ are the errors at $k^{th}$ and $k-1$ instants

$I_{REF}(k)$ and $I_{REF}(k-1)$ are the dc link reference currents at $k^{th}$ and $k-1$ instants, $k_p$ and $k_i$ are the proportional and Integral gains.
4.4.1.1 Determination of Duty cycle

PWM is used to control the VSI feeding the motor phase currents. The PWM duty cycle is determined by using the Equation

\[
\text{Duty Cycle}(k) = \text{Duty Cycle}(k-1) + k_{pc}(I_{REF}(k) - I_{MAX}(k)) \quad (4.25)
\]

Where \( \text{Duty Cycle}(k) \) is the on duration of a PWM period.

\( I_{MAX}(k) \) is the measured dc link current.

\( k_{pc} \) is the gain of the current controller.

The PWM unit generates PWM signal with a duty cycle as found in Equation to control the desired gate current of an inverter. By controlling the gate current the stator phase voltage and hence the speed of the motor is controlled.

4.4.1.2 Closed loop simulation using PI

The simulink block diagram of gate control based speed control of PMBLDC motor using PI/FLC controller scheme is shown in Figure 4.16.
Figure 4.16 Simulink Block Diagram of Gate Control Based Speed Control of PMBLDC Motor Using PI Controller/ FLC Schemes
The motor speed variation to a change in set speed using PI controller is shown in Figure 4.17. The response shows the starting performance as well as the response with a step change in reference load. Under no load condition the reference speed is initially set at 550 rpm. Consequently the speed is increased to 700rpm and then to 1000rpm and decreased to 600rpm. Using PI controller the motor reaches the reference speed at 0.4 seconds with overshot and undershoot.

![Figure 4.17 Variation of motor speed for changes in set point using PI controller](image1.png)

The corresponding phase current variation to a change in actual speed is shown in Figure 4.18. The magnitude of phase current variation is more when there is a large change in set speed.

![Figure 4.18 Variation of phase current for changes in set point using PI controller](image2.png)
The corresponding electromagnetic torque variation to a change in actual speed is shown in Figure 4.19. When the set speed is decreased from 1000 rpm to 600 rpm the variation in torque is in negative direction as shown in Figure 4.19.

![Figure 4.19 Variation of electromagnetic torque for changes in set point using PI controller](image)

Figure 4.19 Variation of electromagnetic torque for changes in set point using PI controller

Figure 4.20 shows the speed response of the PMBLDC drive for changes in set point with constant load torque of 5N-m using PI controller. There is an overshoot in the speed response and it takes more settling time to reach the actual speed.

![Figure 4.20 Variation of motor speed for changes in set point with constant load using PI controller](image)

Figure 4.20 Variation of motor speed for changes in set point with constant load using PI controller
The PMBLDC drive speed response for step changes in the load torque using PI controller for a rated speed of 800 rpm is shown in Figure 4.21. At $t = 1.5$ second, a load torque of 0.5 N-m is applied to the motor shaft and removed at $t = 2.5$ seconds. The variation in speed due to the applied load torque is reflected using PI controller.

![Figure 4.21 Variation of motor speed to the Step change in load torque using PI controller](image)

The electromagnetic torque variation for step changes in the load torque for a rated speed of 800 rpm is shown in Figure 4.22. At $t = 1.5$ second, a load torque of 0.5 N-m is applied to the motor shaft and removed at $t = 2.5$ seconds. The variation in electromagnetic torque due to the applied load torque is shown in Figure 4.22.

![Figure 4.22 Variation of electromagnetic torque to the Step change in load torque using PI controller](image)
Figure 4.23 shows the variation of speed in the absence of load disturbance and with load disturbance at time $t = 8$ seconds. Due to PI controller action the PMBLDC motor is capable of settling to rated speed even with load disturbance.

![Figure 4.23 Comparison of Variation in speed with load disturbance based on Gate control method using PI Controller Scheme](image)

Referring to the drawbacks of conventional PI controller, FLC scheme is designed for gate control based speed control of PMBLDC motor.

4.4.2 Gate Control Method using FLC Scheme

The reference current $I_{\text{REF}}$ and $I_{\text{MAX}}$ from the motor are compared, to obtain the error and change in error. These errors are considered as two input linguistic variable and the firing angle is taken as output variable. For convenience, the inputs and the output of the FLC were scaled with three different coefficients. These scaling factors can be constant or variable, and play an important role for the FLC design in order to achieve a good behavior in both transient and steady state. In this work these scaling factors are assumed as constant value.
Fuzzification

Seven membership functions with overlap, of triangular shape and equal width, are used for each input variable, so that a 49 rule base is created. The seven linguistic variables used as input and output variables are Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM) and Positive Large (PL) are used in this work.

Triangular membership function is assigned for input and output variables defined in different universe of discourses as in Figure 4.24. Rule Base Matrix for Gate Control Based Speed Control of PMBLDC Motor Using FLC Scheme is given in Table 4.9

Figure 4.24 Membership Function for error, change of error and Output Variable (firing angle) in FLC Scheme Based on Gate Control Method
The error is defined in Equation (4.26) as

\[ e(k) = y(k) - w(k) \]  \hspace{1cm} (4.26)

Where

- \( e(k) \) is the error
- \( w(k) \) is the Ref current
- \( y(k) \) is the Actual current

The change in error is defined in Equation (4.27)

\[ ce(k) = e(k) - e(k-1) \]  \hspace{1cm} (4.27)

Table 4.9 Rule Base Matrix for Gate Control Based Speed Control of PMBLDC Motor Using FLC Scheme

<table>
<thead>
<tr>
<th>( e(k)/ce(k) )</th>
<th>PL</th>
<th>PM</th>
<th>PS</th>
<th>Z</th>
<th>NS</th>
<th>NM</th>
<th>NL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
</tr>
<tr>
<td>PM</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>NL</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>Z</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>NS</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
</tr>
<tr>
<td>NL</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

**Decision making logic**

‘Max-Min’ criterion is used to combine the results to generate single truth value, which determines the outcome of the rules.

**Defuzzification**

The Defuzzification using Center of Area is considered for this application as it produces the results which are sensitive to all the rules. FLC design parameters for the gate control based speed control of PMBLDC motor is given in Table 4.10.
Table 4.10  FLC Design Parameters for the Gate Control Based Speed Control of PMBLDC Motor

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input variables</td>
<td>2</td>
</tr>
<tr>
<td>Output variable</td>
<td>1</td>
</tr>
<tr>
<td>linguistic levels</td>
<td>7</td>
</tr>
<tr>
<td>Rules</td>
<td>49</td>
</tr>
<tr>
<td>Membership Function</td>
<td>Triangular</td>
</tr>
<tr>
<td>Decision making logic</td>
<td>Max-Min criteria</td>
</tr>
<tr>
<td>Defuzzification</td>
<td>Centre of Area method</td>
</tr>
</tbody>
</table>

4.4.2.1  Closed loop simulation using FLC Scheme

The closed loop simulation results of stator phase current and phase back EMF using Gate controlled based FLC scheme is shown in Figure 4.25. The phase back- EMF is in phase with stator phase current. In order to get maximum efficiency from the motor, the commutation should take place when the current in a stator winding is in phase with the back EMF of the same winding. The maximum efficiency is obtained only when the phase back EMF and stator current are in phase as in Figure 4.25.

![Figure 4.25  Phase back- EMF and Stator Current Variation in PMBLDC Motor Based on Gate Control Method Using FLC Scheme](image-url)
The closed loop response of PMBLDC motor with FLC scheme for set point changes in speed is shown in Figure 4.26. The response shows the starting performance as well as the response with a step change in reference load. Under no load condition the reference speed is initially set at 550 rpm. Consequently the speed is increased to 700 rpm and then to 1000 rpm and decreased to 600 rpm. Using FLC scheme the motor reaches reference speed within 0.2 seconds without any overshoot/undershoot and with zero steady state error.

Figure 4.26  Variation of motor speed for changes in set point using FLC Scheme

The corresponding phase current variation for changes in set point is shown in Figure 4.27. It is observed that the magnitude of phase current varies when there is a variation in set speed.
The corresponding electromagnetic torque variation to a change in set point is shown in Figure 4.28. When the set speed is decreased from 1000rpm to 600rpm the variation in torque is in negative direction.

Figure 4.28 Variation of electromagnetic torque for changes in set point using FLC Scheme
Figure 4.29 shows the speed response of the PMBLDC drive for changes in set point with constant load torque of 5N-m using FLC scheme. The response with FLC controller is capable of following the reference speed with zero steady state error and no overshoot.

![Graph showing speed response with FLC scheme](image)

Figure 4.29 Variation of motor speed for changes in set point with constant load using FLC Scheme

The PMBLDC drive speed response for step changes in the load torque using FLC for a rated speed of 800 rpm is shown in Figure 4.30. At $t = 1.5$ second, a load torque of 0.5 N-m is applied to the motor shaft and removed at $t = 2.5$ seconds. The speed response in Figure 4.30 shows that the actual speed does not change much due to the disturbance with FLC scheme.
The electromagnetic torque variation for step changes in the load torque for a rated speed of 800 rpm using FLC is shown in Figure 4.31. At $t = 1.5$ second, a load torque of 0.5 N-m is applied to the motor shaft and removed at $t = 2.5$ seconds. The oscillation in electromagnetic torque due to applied load torque during transition period is negligible.
The stator current variation for step changes in the load torque using FLC is to the load applied at \( t = 1.5 \) seconds and reduces to its original shown in Figure 4.32. The stator current reaches the new value corresponding value when the load is removed at \( t = 2.5 \) seconds.

![Figure 4.32 Stator phase current variation to the Step change in load torque using FLC Scheme](image1)

Figure 4.32 Stator phase current variation to the Step change in load torque using FLC Scheme

Figure 4.33 shows the comparison of variation in speed using PI and FLC schemes in the presence of load disturbance. The FLC scheme takes 3.4 seconds to settle at rated speed whereas PI Controller requires 4.2 seconds. The Load torque of 8.3 N-m is given as disturbance at \( t = 8 \) seconds. The PI controller takes longer time than FLC scheme to settle at rated speed.

![Figure 4.33 Comparison of Variation in Speed with load Disturbance Based on Gate Control Method Using PI and FLC Scheme](image2)
Figure 4.34 shows the torque response of FLC and PI controller schemes. At the time of starting, in FLC scheme the torque increases up to nearly 42 N-m and stabilizes rapidly when the motor reaches the reference value. In PI controller the torque increases only upto 33.5N-m and takes larger time to stabilize when compared to the FLC scheme.

![Graph showing torque response of FLC and PI controller schemes](image)

**Figure 4.34  Comparison of Variation in Torque in the Absence of Load Disturbance Based on Gate Control Method Using PI and FLC Scheme**

For the ac machine drive application, full utilization of the DC bus voltage is extremely important in order to achieve the maximum output torque under all operating conditions. In this aspect compared to any other PWM method for voltage source inverter, the PWM method based voltage space vectors results in excellent DC bus utilization. Moreover as compared to sinePWM method, the current ripple in steady state operation can be minimised using SVM method. Hence, in the present work the speed of the motor is controlled using the best modulation strategy known as SVM method. With SVM the performance of motor is improved because it
eliminates all the lower order harmonics in the output voltage of the inverter. The performance of the motor can be further improved by eliminating the current harmonics in the stator current of the motor. When the machine is on load, the load neutral is isolated, which causes interaction among phases. Since, interaction was not considered; stator current harmonics could not be reduced in the Gate control method. In order to overcome the above drawbacks, the SVM method is proposed.

### 4.4.3 Space Vector Modulation (SVM)

SVM is a digital modulating technique, where the objective is to generate PWM output in such a way that the average voltages follow the sinusoidal three phase command voltages with a minimum amount of harmonic distortion. This is done in each sampling period by properly selecting the switch states of the inverter and by the computation of the appropriate time period for each state. The SVM for a three phase voltage source inverter is obtained by sampling the reference vector at the fixed sampling frequency. The eight possible switching combinations of the switching network are mapped into an orthogonal plane, resulting in six non-zero vectors and two zero vectors. The six non-zero switching vectors form a hexagon as shown in Figure 4.35.

![Figure 4.35 Representations of Six Non-Zero Switching Vectors](image)
In order to implement the SVPWM, the voltage equations in the a-b-c reference frame can be transformed into the stationary d-q reference frame that consists of the horizontal (direct) and vertical (quadrature) axes. There are eight possible combinations of ON and OFF patterns for the upper power switches. The ON and OFF states of the lower power devices are opposite to the upper one and hence are easily determined, once the states of the upper power switches are determined.

### 4.4.3.1 Modulation Procedure

The four steps involved to perform the Space Vector Modulation are as follows:

1. The reference signals for phase A, B and C are mapped into the orthogonal d-q co-ordinated and are represented by reference vector $V_{\text{ref}}$.

2. Switching vectors are selected, including non-zero and zero vectors to synthesize the reference vector $V_{\text{ref}}$ for one switching cycle.

3. The time durations for all selected switching vectors are calculated by a simple trigonometric algorithm. The objective is to make the averaged switching vector in one switching cycle equal to the reference vector $V_{\text{ref}}$.

4. The switching vectors are sequenced and applied to the switching network.

### 4.4.3.2 Modulation Algorithm

The various steps involved in the modulation algorithm is as follows:

1. Read three phase reference voltages ($V_a$, $V_b$, $V_c$).
2. Obtain three-phase to two-phase transformation (a, b, c→d, q).

3. Calculate absolute values of $V_d$, $V_q$ and arc-tangent ($V_d/V_q$).

4. Identify the sector in which the reference voltage vector lies.

5. Select the switching vectors corresponding to the identified sector.

6. The switching times are calculated depending on the output voltage vector magnitude.

7. Sequence the switching vectors as given by the sequencing scheme (symmetrical).

8. Control signals are applied for each phase of the switching network.

9. Output is obtained at the load terminals of the voltage source inverter.

The relationship between the switching variable vector $(a, b, c)^t$ and the line-to-line voltage vectors $(V_{ab}, V_{bc}, V_{ca})^t$ is given in Equation (4.28)

$$
\begin{bmatrix}
V_{ab} \\
V_{bc} \\
V_{ca}
\end{bmatrix} = V_{dc} \begin{bmatrix}
1 & -1 & 0 \\
0 & 1 & -1 \\
-1 & 0 & 1
\end{bmatrix} \begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
$$  \hspace{1cm} (4.28)

Also, the relationship between the switching variable vector $(a, b, c)^t$ and the phase voltage vector $(V_{an}, V_{bn}, V_{cn})^t$ can be expressed as given in Equation (4.29).

$$
\begin{bmatrix}
V_{an} \\
V_{bn} \\
V_{cn}
\end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix}
2 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix} \begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
$$  \hspace{1cm} (4.29)
There are six modes of operation in a cycle and the duration of each mode is 60°. The switches are numbered in the sequence of gating them (612, 123, 234, 345, 456 and 561). The line to neutral and line to line voltages in terms of dc-link voltage $V_{dc}$ obtained for the eight switching vectors using Equations (4.28) and (4.29) are presented in Table 4.11.

### Table 4.11 Voltages across the switching vectors

<table>
<thead>
<tr>
<th>Voltage Vector</th>
<th>Switching Vectors</th>
<th>Line to Neutral Voltage</th>
<th>Line to line Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>1 0 0</td>
<td>$V_{an}$</td>
<td>2/3 -1/3 -1/3</td>
</tr>
<tr>
<td>$V_2$</td>
<td>1 1 0</td>
<td>$V_{an}$</td>
<td>1/3 1/3 -2/3</td>
</tr>
<tr>
<td>$V_3$</td>
<td>0 1 0</td>
<td>$V_{an}$</td>
<td>-1/3 2/3 -1/3</td>
</tr>
<tr>
<td>$V_4$</td>
<td>0 1 1</td>
<td>$V_{an}$</td>
<td>-2/3 1/3 1/3</td>
</tr>
<tr>
<td>$V_5$</td>
<td>0 0 1</td>
<td>$V_{an}$</td>
<td>-1/3 -1/3 2/3</td>
</tr>
<tr>
<td>$V_6$</td>
<td>1 0 1</td>
<td>$V_{an}$</td>
<td>1/3 -2/3 1/3</td>
</tr>
</tbody>
</table>

#### 4.4.3.3 Mathematical Modelling of $V_d, V_q$ for a PMBLDC Motor

Various steps involved in developing $V_d, V_q$ for PMBLDC motor from $V_{an}, V_{bn}, V_{cn}$ are presented below.

**Step 1:** Determine $V_d, V_q, V_{ref}$ and angle ($\alpha$) as shown in Figure 4.36.

![Figure 4.36 Representation of $V_d, V_q, V_{ref}$ from $V_{an}, V_{bn}, V_{cn}$](image-url)
\[
V_d = V_{an} - V_{bn} \cos 60^\circ - V_{cn} \cos 60^\circ \\
= V_{an} - 1/2 V_{bn} - 1/2 V_{cn}
\]

(4.30)

\[
V_q = 0 + V_{bn} \cos 30^\circ - V_{cn} \cos 30^\circ \\
= \sqrt{3}/2 V_{bn} - \sqrt{3}/2 V_{cn}
\]

(4.31)

From the above Equations (4.30) and (4.31), the relationship between direct voltage \(V_d\) and Quadrature voltage \(V_q\) with line to neutral voltage is given in Equation (4.32)

\[
\begin{bmatrix}
V_d \\
V_q
\end{bmatrix}
= \frac{2}{3}
\begin{bmatrix}
1 & -1/2 & -1/2 \\
0 & \sqrt{3}/2 & -\sqrt{3}/2
\end{bmatrix}
\begin{bmatrix}
V_{an} \\
V_{bn} \\
V_{cn}
\end{bmatrix}
\]

(4.32)

\[
|V_{\text{ref}}| = V_d^2 + V_q^2
\]

(4.33)

\[
\alpha = \tan^{-1}\left[\frac{V_q}{V_d}\right]
\]

(4.34)

**Step 2:** Determine the time duration \(T_1, T_2\) and \(T_0\)

![Figure 4.37 Reference Vector Realization at Sector1](image)
The switching time duration can be calculated using the Figure 4.37.

Switching time duration at Sector1 is obtained as follows

\[
\int_{0}^{T_1} V_{\text{ref}} \, dt = \int_{0}^{T_1} V_1 \, dt + \int_{0}^{T_1+T_2} V_2 \, dt + \int_{T_1+T_2}^{T_2} V_0 \, dt \tag{4.35}
\]

\[
T_z \, V_{\text{ref}} = T_1 \, V_1 + T_2 \, V_2 \tag{4.36}
\]

\[
T_z \, V_{\text{ref}} \left[ \begin{array}{c} \cos(\alpha) \\ \sin(\alpha) \end{array} \right] = T_1 \frac{2}{3} V_{dc} \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] + T_2 \frac{2}{3} V_{dc} \left[ \begin{array}{c} \cos(\pi/3) \\ \sin(\pi/3) \end{array} \right] \tag{4.37}
\]

(where, \(0 \leq \alpha \leq 60\))

\[
T_1 = T_z \cdot a \cdot \frac{\sin(\pi/3 - \alpha)}{\sin(\pi/3)} \tag{4.38}
\]

\[
T_2 = T_z \cdot a \cdot \frac{\sin(\alpha)}{\sin(\pi/3)} \tag{4.39}
\]

\[
T_0 = T_z - (T_1 + T_2) \tag{4.40}
\]

where \(T_z = 1/f_s\) and

\[
a = \frac{|V_{\text{ref}}|}{(2/3)V_{dc}} \tag{4.41}
\]

**Step 3:** Determine the switching time of each transistor (S₁ to S₆)

The model thus formulated is used to carry out simulation studies in Simulink platform. The switching sequence thus for the lower and upper thyristors obtained is listed in Table 4.12. The null time has been conveniently
distributed between $V_0$ and $V_7$ vectors to describe the symmetrical pulse pattern to give minimal output harmonics. By comparing the stationary frame d-q components of the reference voltage vector, the sector where the reference vector to be located is identified. Using the d-q components of reference vector and the DC link voltage information, the effective time $T_1, T_2$ are calculated. Using the corresponding sector information, the actual switching time for each inverter is generated from the combination of the effective times and zero sequence time.

Table 4.12 Switching sequence for lower and upper Thyristors

<table>
<thead>
<tr>
<th>Sector</th>
<th>Upper Switches($S_1, S_3, S_5$)</th>
<th>Lower Switches($S_4, S_6, S_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_1 = T_1 + T_2 + T_0/2$</td>
<td>$S_4 = T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_2 + T_0/2$</td>
<td>$S_6 = T_1 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_0/2$</td>
<td>$S_2 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td>2</td>
<td>$S_1 = T_1 + T_0/2$</td>
<td>$S_4 = T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_1 + T_2 + T_0/2$</td>
<td>$S_6 = T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_0/2$</td>
<td>$S_2 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td>3</td>
<td>$S_1 = T_0/2$</td>
<td>$S_4 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_1 + T_2 + T_0/2$</td>
<td>$S_6 = T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_2 + T_0/2$</td>
<td>$S_2 = T_1 + T_0/2$</td>
</tr>
<tr>
<td>4</td>
<td>$S_1 = T_0/2$</td>
<td>$S_4 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_1 + T_0/2$</td>
<td>$S_6 = T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_1 + T_2 + T_0/2$</td>
<td>$S_2 = T_0/2$</td>
</tr>
<tr>
<td>5</td>
<td>$S_1 = T_2 + T_0/2$</td>
<td>$S_4 = T_1 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_0/2$</td>
<td>$S_6 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_1 + T_2 + T_0/2$</td>
<td>$S_2 = T_0/2$</td>
</tr>
<tr>
<td>6</td>
<td>$S_1 = T_1 + T_2 + T_0/2$</td>
<td>$S_4 = T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = T_0/2$</td>
<td>$S_6 = T_1 + T_2 + T_0/2$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = T_1 + T_0/2$</td>
<td>$S_2 = T_2 + T_0/2$</td>
</tr>
</tbody>
</table>
4.4.3.4   Speed Control of SVM Inverter fed PMBLDC Motor

The block diagram for the closed loop sensorless speed control of PMBLDC motor using SVM technique is shown in Figure 4.38. The MOSFETs are used as switching devices to vary the output frequency of the inverter is varied. The applied voltage to the motor is varied linearly with the supply frequency to maintain the flux constant.

![Simulink Diagram for the Speed Control of SVM Inverter Fed PMBLDC Motor](image)

Figure 4.38   Simulink Diagram for the Speed Control of SVM Inverter Fed PMBLDC Motor

For more clarity, the details of SVM block are shown in Figure 4.39. This block gives the sector information and the actual switching time for each phase of the inverter.
Figure 4.39  Simulink Diagram for Determining Sector and Switching Duration in SVM Block

The Figure 4.40 shows the gate driving pulses of MOSFETs. The Figure 4.41 shows the stator phase currents. There are no spikes in the current output, which shows that there is significant reduction in the harmonic content.

Figure 4.40  Gate Driving Pulses to MOSFETs of the Inverter
The Figure 4.42 shows the simulated output of electromagnetic torque. The torque reaches to nearly 0.5 N-m when the motor starts. It stabilizes rapidly nearly to zero when the motor speed reaches the rated value. The nominal torque is applied at t = 0 second. Due to the controller action DC bus voltage increases to produce the required electric torque. The initial current is high and later decreases during the acceleration to the nominal speed. When the nominal torque is applied, the stator current increases to maintain the nominal speed.

Figure 4.42  Variation of Electromagnetic Torque with SVM Inverter Fed PMBLDC Motor
However, the motor's inertia prevents this noise from appearing in the motor's speed. The rotor reaches the rated speed in 3 seconds. Figure 4.43 shows the speed variation obtained using SVM scheme. It is observed that the amount of harmonic content is less compared to PI and FLC based on voltage control and Gate control method.

![Figure 4.43 Variation of Speed Based on SVM Method](image1)

The comparison of variation in electromagnetic torque for SVM, FLC and PI controller fed PMBLDC motor is shown in Figure 4.44. From the torque characteristic it is clearly seen amount of ripple content presence in PI controller is comparatively more than FLC and SVM technique.

![Figure 4.44 Comparison of Variation in Electromagnetic Torque for SVM, FLC and PI Controller fed PMBLDC Motor](image2)
The amount ripple content in torque for the SVM, FLC and PI controller is given in Figure 4.45. The amount of ripple content presence using PI controller is roughly four times greater than SVM technique.

![Figure 4.45 Comparison of Ripple Content in Torque for SVM, FLC and PI Controller Fed PMBLDC Motor](image)

From the quantitative comparison parameter in Table 4.13 it is clear that the SVM scheme provides better performance compared to other schemes. The settling time for the PMBLDC motor to settle at rated speed using SVM is only 3 seconds. The amount of power transferred to the motor is larger, which indicates lesser power loss in the inverter and more efficient use of supply voltage.

**Table 4.13  Quantitative Analysis of Performance Measures of Various Controllers in Sensorless Method**

<table>
<thead>
<tr>
<th>Performance Measures</th>
<th>PI Control</th>
<th>Fuzzy Logic Control</th>
<th>Space Vector Modulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling Time Ts in Second</td>
<td>4.2</td>
<td>3.4</td>
<td>3.0</td>
</tr>
<tr>
<td>RMS values of current in Amphere</td>
<td>4.625</td>
<td>4.886</td>
<td>7.35</td>
</tr>
<tr>
<td>IAE</td>
<td>69.74</td>
<td>11.55</td>
<td>0.335</td>
</tr>
<tr>
<td>ISE</td>
<td>150</td>
<td>33.5</td>
<td>0.8002</td>
</tr>
<tr>
<td>THD in %</td>
<td>3.123</td>
<td>1.434</td>
<td>0.2897</td>
</tr>
</tbody>
</table>
4.5 SUMMARY

In SVPWM, switching harmonics are suppressed to a large extent by the low-pass characteristic of the machine inductances and by the inertia of the mechanical system. The three phase inverter has lower harmonic distortion both in phase voltage and current. The amount of harmonic distortion in the speed is less compared with voltage control method using PI and FLC schemes and gate control method using PI and FLC schemes. The RMS line current is more in SVM technique. From the FFT analysis it is also observed that SVM generates less harmonic distortion in the output voltage and more efficient use of supply voltage and hence results in improved motor performance.

In the subsequent chapter the experimental investigation of speed control in the presence of sensor with PI control and sensorless control with FLC scheme are developed and presented.