CHAPTER 3

FAULT DETECTION SCHEMES FOR
THREE PHASE INDUCTION MOTOR

3.1 INTRODUCTION

Induction motors play a vital role in industries. Reliability of drive systems with these motors has a serious economical impact on the operation of industrial plants. Thermal overloading, degradation and aging of stator winding insulation often lead to inter turn faults. The inter turn fault in stator winding produces a large circulating current to flow in the shorted turns, which in turn leads to phase to ground fault. The ground fault causes irreversible damage to the machine core.

Detection of these faults in its incipient stage avoids further damage to the motor. This is achieved by extracting the parameters, which are sensitive to faults and continuous monitoring of these parameters results in fault indication. Inherent asymmetries of the motor, unbalanced supply voltage and measurement errors should be taken into account while extracting the fault indicator. The off-line insulation testing method can be used to detect the inter turn fault since this method is inconvenient and do not allow frequent testing of the insulation winding, online methods for detecting turn faults have been recently developed (Lee et al 2003, Bernieri et al 1996, Filippetti et al 1993, 1995, 1998 and Chow et al 1993). In industries motor failure during manufacturing process leads to excessive downtime expenses and this also motivates the online condition monitoring of induction machines.
The turn fault detection based on monitoring the current spectrum has been reported in Penman et al (1986, 1994) and Rankin (1995). The accurate sensing devices are required to monitor the current spectrum for fault signature and they are expensive. Therefore, this method is considered expensive for small machines.

The induction motor model derived in Tallam et al (2002) is useful in extracting fault signature for fault detection. But, it is very difficult to get accurate mathematical for induction motor including all nonlinearities and motor asymmetry. The supply voltage unbalance, inherent motor asymmetries and measurement errors cause the change in the fault indicator. Therefore, these effects must be compensated.

The neural network based turn fault detection method reported in Tallam et al (2003) attempted to detect the change in negative sequence current (I_{sn}) as fault indicator since the inter turn fault causes the unbalance in the motor which, in turn, introduces the negative sequence current. However, this method requires large amount of data to be stored for the given machine to avoid false detection.

Turn fault detection methodology proposed in Lee et al (2003) is immune to supply voltage unbalance. The effects due to inherent motor asymmetry are also considered but the fault detection scheme is not included.

In this chapter, three different fuzzy logic based fault detection schemes are studied. The first scheme is based on monitoring the line currents. Second scheme is based on monitoring the negative sequence component of line currents compensating the effect of asymmetries in the machine and unbalanced supply voltage. Third scheme is based on monitoring sequence component impedance, which is immune to supply voltage unbalance variation. The performances of these schemes are compared.
In this chapter, neural network fault detection scheme based monitoring the sequence component impedance of the machine is also presented. To improve the performance of conventional BP algorithm, the IWU scheme is presented for fault detection. To test the efficiency of the IWU scheme, line currents based fault detection scheme is studied. The performance of these algorithms is compared.

3.2 FUZZY FAULT DETECTION SCHEMES

In general, the stator of three phase induction motor is wound with balanced distributed winding. The inter turn short in the stator winding introduces the unbalance the stator line currents. The unbalanced line currents introduce the negative sequence current in the winding. This unbalance is also reflected in the symmetrical component impedance matrix of three phase machine. The winding unbalance can be detected if any one of these parameters is monitored. In this section, three different fault detection schemes based on monitoring the above said parameters are discussed to detect stator winding fault. For our study of fault, the winding study three phase induction motor is used, whose ratings and parameters are given in Table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>1.5kW</td>
</tr>
<tr>
<td>Slots</td>
<td>36</td>
</tr>
<tr>
<td>V, rated</td>
<td>415 V</td>
</tr>
<tr>
<td>Poles</td>
<td>4</td>
</tr>
<tr>
<td>Speed, rated</td>
<td>1440 rpm</td>
</tr>
<tr>
<td>Stator resistance per phase, $R_s$</td>
<td>11.2 $\Omega$</td>
</tr>
<tr>
<td>Rotor resistance referred to stator per phase, $R_r$</td>
<td>7.97 $\Omega$</td>
</tr>
<tr>
<td>Magnetizing inductance per phase, $L_{m}$</td>
<td>0.36 H</td>
</tr>
<tr>
<td>Stator and leakage inductances, $L_s=L_r$</td>
<td>0.056 H</td>
</tr>
</tbody>
</table>
3.2.1 Line currents based fault detection scheme

3.2.1.1 Problem definition

The turn fault in the stator winding creates the unbalanced line currents. Therefore, line currents are monitored to detect the turn faults. The fault detection is the mapping between the inputs and target outputs. The training samples for different faults are obtained by simulating the faults externally in the winding study motor. It is assumed that supply voltage is balanced.

To simulate stator winding turn faults, winding study induction motor whose ratings and parameters are given in Table 3.1 is used. The motor has the provision to create winding fault externally. The motor is connected through Variac to operate the motor at reduced voltage to limit the line current. Different faults are created and line currents are measured. The measured line currents for various faults are given in the Table 3.2.

Table 3.2 Line currents at 150V at no load with fault

<table>
<thead>
<tr>
<th>Condition $(F_c)$</th>
<th>Line current $I_a$ (Ampere)</th>
<th>Line current $I_b$ (Ampere)</th>
<th>Line current $I_c$ (Ampere)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>1 turn short</td>
<td>1.9</td>
<td>1.0</td>
<td>1.4</td>
</tr>
<tr>
<td>2 turn short</td>
<td>3.0</td>
<td>1.0</td>
<td>2.2</td>
</tr>
<tr>
<td>3 turn short</td>
<td>4.5</td>
<td>1.6</td>
<td>3.1</td>
</tr>
</tbody>
</table>
3.2.1.2 Proposed scheme

The schematic of the line currents based fault detection scheme is shown in Figure 3.1. The FFD monitors the line currents continuously to detect the winding unbalance due to inter turn short. The FFD is trained with line currents representing different fault conditions before using it for fault detection. To demonstrate this scheme, the winding study induction motor is used and experiment is conducted as described previously. The Table 3.2 is used as training data. With this data, the FFD is trained and tested in through computer simulation.

Figure 3.1 Schematic of line currents based fault detection scheme

3.2.1.3 Simulation of FFD

The FFD is constructed with three inputs and single output. The stator currents $I_a$, $I_b$ and $I_c$ are inputs of the fault detector and are used to estimate the fault codes representing different faults. The universe of discourse of line currents $I_a$, $I_b$ and $I_c$ are defined as
\[ I_a = \{ I_a / I_{al} \leq I_a \leq I_{au} \} \]
\[ I_b = \{ I_b / I_{bl} \leq I_b \leq I_{bu} \} \]
\[ I_c = \{ I_c / I_{cl} \leq I_c \leq I_{cu} \} \]

where, \( I_{al}, I_{bl}, I_{cl} \) are lower limits of currents \( I_a, I_b \) and \( I_c \) respectively.

\( I_{au}, I_{bu}, I_{cu} \) are upper limits of currents \( I_a, I_b \) and \( I_c \) respectively.

Three fuzzy sets are defined on each of input spaces corresponding to low, medium and high for each variable. The output space \( F_c \) of the fault detector is defined

\[ F_c = \{ F_c / 0 \leq F_c \leq 3 \} \]

Simulation is carried out using MATLAB. A triangular membership functions and LOM defuzzification method is used for simulation. The membership functions used for simulation are shown in Figure 3.2. The fuzzy rules are framed based on expert’s experience. The derived fuzzy rules are given in Table 3.3.

Table 3.3  Fuzzy rules for line currents based fault detection scheme

<table>
<thead>
<tr>
<th>( I_a, I_c )</th>
<th>LL</th>
<th>MM</th>
<th>HH</th>
<th>LM</th>
<th>ML</th>
<th>MH</th>
<th>HM</th>
<th>LH</th>
<th>HL</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>M</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>M</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

L: Low     M: Medium    H: High
Figure 3.2 Membership functions used for line currents based fault detection scheme

(a) Membership functions for line current, $I_a$

(b) Membership functions for line current, $I_b$
(c) Membership functions for line current, $I_c$

(d) Membership functions for output variable, $F_c$

Figure 3.2  (Continued)
3.2.1.4 Simulation results and discussion

The FFD is effectively tuned using the experimental data obtained from the test motor. The trained FFD is used for fault detection. The test results of FFD are given in Table 3.4.

Table 3.4 Simulation results of FFD for line currents based fault detection scheme

<table>
<thead>
<tr>
<th>Condition</th>
<th>Line current $I_a$ (Ampere)</th>
<th>Line current $I_b$ (Ampere)</th>
<th>Line current $I_c$ (Ampere)</th>
<th>Expected output ($F_c$)</th>
<th>Output of FFD</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 turn fault</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 turn short</td>
<td>1.9</td>
<td>1.0</td>
<td>1.4</td>
<td>1</td>
<td>0.99</td>
<td>1.0</td>
</tr>
<tr>
<td>2 turn short</td>
<td>3.0</td>
<td>1.0</td>
<td>2.2</td>
<td>2</td>
<td>2.07</td>
<td>3.5</td>
</tr>
<tr>
<td>3 turn short</td>
<td>4.5</td>
<td>1.6</td>
<td>3.1</td>
<td>3</td>
<td>2.91</td>
<td>4.0</td>
</tr>
<tr>
<td>Average error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.25</td>
</tr>
</tbody>
</table>

From the Table 3.4, it is found that the line currents based fault detection scheme is capable of detecting turn faults. Therefore this scheme can be used directly for online fault detection. However, this scheme is not immune to supply voltage variations. To overcome this drawback, negative sequence current based fault detection scheme is proposed in the next section.

3.2.2 $I_{sn}$ based fault detection scheme

The stator winding turn fault introduces the unbalance in the winding. The winding unbalance injects negative sequence component in the line currents of the machine. The negative sequence component of line
currents can be measured from machine and used as fault indicator. Even healthy machine possesses certain degree of asymmetry and causes the unbalanced line currents in the winding. This unbalance coupled with supply voltage unbalance may lead to false indication of fault. Hence, it is necessary to compensate these effects while extracting fault signature for fault detection.

### 3.2.2.1 Problem definition

An induction machine with stator winding turn faults on a single phase is shown in Figure 3.3. $I_f$ is the fault current in shorted turns and $\mu$ denotes the fraction of shorted turns. $S_2$ is the shorted turns. $X_s$ and $X_r$ are the per phase leakage reactance of stator and rotor windings. $R_s$ and $R_r$ are per phase stator and rotor resistance of stator and rotor windings and ‘s’ denotes slip. $R_o$ is the resistance equivalent to the core loss component of no load current. $X_m$ is the per phase magnetizing reactance.

![Figure 3.3 Stator winding with turn fault](image)

The inter turn fault in induction machine inject the negative sequence components in the line currents. The model derived in Tallam et al (2002) is suitably used here to validate the proposed scheme and experimental observations. The sequence component equivalent circuits of three phase induction machine with stator winding turn fault are shown in Figure 3.4 and Figure 3.5.
From the above equivalent circuits, the negative sequence component of line currents estimated has two components, one component due to the inter turn fault and other component due to asymmetries in the motor and unbalanced voltages. The induction machine posses certain degree of asymmetry even under healthy condition. To obtain exact fault signature, a test is conducted on the healthy induction machine and negative sequence component \( I_{sn} \) of line currents is estimated at rated balanced voltage. The estimated \( I_{sn} \) can be expressed as

\[
I_{sn} = k_1 V_{sp} + k_2 V_{sn}
\]

where, \( k_1 \) and \( k_2 \) are load dependent complex numbers;

\( V_{sp} \) is positive sequence component of line voltages;

\( V_{sn} \) is negative sequence component of line voltages.

\( k_1 V_{sp} \) is the current component due to asymmetry and \( k_2 V_{sn} \) is the current component due unbalanced voltages. Since, \( I_{sn} \) is measured at
balanced voltage, it will have only the current component contributed by the asymmetries in the machine. Then the estimated component of current is subtracted from the negative sequence component of currents with turn fault to extract the fault signature.

### 3.2.2.2 Proposed scheme

The schematic of the proposed scheme is shown in Figure 3.6. The proposed system estimates the parameters such as $V_{sn}$, $V_{sp}$, $I_{sn}$ and $I_{sp}$ continuously online.

![Figure 3.6 Schematic of $I_{sn}$ based fault detection scheme](image)

These parameters are used to estimate the fault severity and to provide compensation for unbalanced voltages and asymmetry in the machine. The compensation for supply voltage unbalance and motor
asymmetry is provided as discussed in the section 3.2.2.3. The FFD determines the motor condition based on the expert knowledge and the measured motor parameters. The design procedure for FFD is explained in the section 3.2.2.5.

3.2.2.3 Compensation

In general, the induction machine exhibits a certain degree of asymmetry. Therefore, the stator currents of the healthy machine may contain the negative sequence current even for balanced voltages. This component also depends on the slip (s). In order to compensate this effect, a look up table may be prepared for the healthy machine for various slips estimating the negative sequence component at balanced supply voltage.

3.2.2.3.1 Machine asymmetry compensation

The negative sequence component of current ($I_{sn}$) in general, can be expressed as

$$I_{sn} = k_1 V_{sp} + k_2 V_{sn} \tag{3.1}$$

where, $k_1$ and $k_2$ are load depended constants. $V_{sp}$ and $V_{sn}$ are the positive and negative sequence components of line voltages respectively. The first term in equation (3.1) represents the component of $I_{sn}$ due to asymmetry and the second term represents the component of $I_{sn}$ due to supply voltage unbalance. The $I_{sn}$ can be calculated for balanced voltage for healthy machine for various loads and can be stored as look up table for asymmetry compensation.

To illustrate the proposed method, the winding study induction motor is used whose ratings are shown in Table 3.1. The machine has 36 coils in the stator and the winding is designed for 4 poles. The machine had the provision to create turn fault externally. To study the load dependent machine
asymmetry, the machine is operated at reduced voltage to avoid damage to the winding. The motor line currents are measured at balanced supply voltage for different loads. The experimental setup for data acquisition is shown in Figure 3.7.

![Figure 3.7](image)

**Figure 3.7 The experimental set up for data acquisition**

Table 3.5 shows the experimental data obtained from the healthy machine under various load condition. It is observed that negative sequence component of line currents is increased with increase in load and even healthy machine draws unbalanced line currents due to asymmetry. Table 3.5 is used as a look up table to compensate the effect of asymmetry while designing the FFD

**Table 3.5 Line currents of the healthy machine at 150V for various loads**

<table>
<thead>
<tr>
<th>Load</th>
<th>Line current $I_a$ (Ampere)</th>
<th>Line current $I_b$ (Ampere)</th>
<th>Line current $I_c$ (Ampere)</th>
<th>$I_{sn}$ (Ampere)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Load</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>0.03</td>
</tr>
<tr>
<td>1/2 load</td>
<td>2.0</td>
<td>2.2</td>
<td>2.4</td>
<td>0.11</td>
</tr>
<tr>
<td>3/4 load</td>
<td>2.5</td>
<td>2.5</td>
<td>3.2</td>
<td>0.23</td>
</tr>
<tr>
<td>Full load</td>
<td>3.6</td>
<td>3.8</td>
<td>4.4</td>
<td>0.24</td>
</tr>
</tbody>
</table>
3.2.2.3.2 Unbalanced voltage compensation

To compensate the effect of unbalanced voltage on the fault signature, an experiment can be conducted on the healthy machine and $I_{sn}$ can be calculated for different balanced voltages for same slip. The values of $k_1$ and $k_2$ can be predicted from Equation (3.1) and the second term in Equation (3.1) indicating the component of $I_{sn}$ introduced by the unbalanced supply voltage. A look up table can be created for $k_1$ and $k_2$ for various values of slip. Knowing the values of $k_2$, $k_2 V_{sn}$ can be calculated at any supply voltage and this component can be subtracted from the measured $I_{sn}$ to obtain the exact fault current.

To study the effect of supply voltage unbalance, the test machine is started at no load at balanced supply voltage. Then, voltage unbalance is created using variac. Table 3.6 shows line currents of the healthy machine at unbalanced supply voltage. Similar procedure can be followed for various loads. Knowing the values of $V_{sn}$, $V_{sp}$, $I_{sn}$ and $I_{sp}$ for each load, $k_2$ can be calculated and then the compensation component can be calculated.

### Table 3.6 Line currents of healthy machine at no load with unbalanced supply voltage

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Line voltage $V_{ab}$ (V)</th>
<th>Line voltage $V_{bc}$ (V)</th>
<th>Line voltage $V_{ca}$ (V)</th>
<th>$V_{sp}$ (V)</th>
<th>$V_{sn}$ (V)</th>
<th>Line current $I_a$ (A)</th>
<th>Line current $I_b$ (A)</th>
<th>Line current $I_c$ (A)</th>
<th>$I_{sp}$ (A)</th>
<th>$I_{sn}$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>105</td>
<td>105</td>
<td>95</td>
<td>101.6</td>
<td>3.3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>0.56</td>
<td>0.20</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>130</td>
<td>126</td>
<td>128</td>
<td>1.45</td>
<td>0.9</td>
<td>0.8</td>
<td>0.9</td>
<td>0.96</td>
<td>0.33</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>150</td>
<td>147</td>
<td>149</td>
<td>0.99</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
<td>1.13</td>
<td>0.03</td>
</tr>
</tbody>
</table>

3.2.2.4 Implementation

The $I_{sn}$ based inter turn fault detection scheme can be implemented online with compensation for supply voltage unbalance and motor asymmetry. Two sets of data points of positive component of voltage ($V_{sp}$),
negative sequence component of voltage ($V_{sn}$) and negative sequence component of current ($I_{sn}$) must be obtained as a function of slip to estimate the negative sequence component of line currents due to unbalanced supply voltage. This is difficult to estimate online, since the supply voltage changes slowly in time by a small amount. So the implementation requires two stages for fault detection namely learning stage and detecting stage.

The learning stage can be performed for each individual machine prior to motor operation. In this stage, data needed for fault detection are measured and stored as the function of slip. In detection stage, the values of $V_{sp}$, $V_{sn}$, $I_{sp}$, $I_{sn}$ and slip can be continuously measured online. The compensation components are estimated from the stored data corresponding to the measured slip ($s$) and can be subtracted from the measured $I_{sn}$ to extract the exact fault signature. To demonstrate the proposed scheme, the test machine is used to simulate the turn faults. The machine is made to run at different voltage levels i.e. 150 V, 130 V and 100 V and then voltage unbalance is created using variac. The line currents and voltages are measured for different turn faults. Table 3.7 shows the unbalanced line voltages and currents of the test machine at no load for different turn faults at 150 V, 130 V and 100 V respectively. The sequence component of currents and voltages are calculated to estimate the fault severity. Since, the quantities, $V_{sn}$, $V_{sp}$, $I_{sn}$ and $I_{sp}$ contains all necessary information to compensate effect of unbalanced voltages and asymmetry in the machine, this scheme can be directly applied for online fault detection.
Table 3.7 Unbalanced line voltages and currents at no load for different turn faults

(a) at 150 V level

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Line voltage $V_{ab}$ (V)</th>
<th>Line voltage $V_{bc}$ (V)</th>
<th>Line voltage $V_{ca}$ (V)</th>
<th>$V_{sp}$ (V)</th>
<th>$V_{sn}$ (V)</th>
<th>Line current $I_a$ (A)</th>
<th>Line current $I_b$ (A)</th>
<th>Line current $I_c$ (A)</th>
<th>$I_{sp}$ (A)</th>
<th>$I_{sn}$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>0</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
<td>1.13</td>
<td>0.03</td>
</tr>
<tr>
<td>1 turn short</td>
<td>150</td>
<td>150</td>
<td>147</td>
<td>149</td>
<td>0.99</td>
<td>1.9</td>
<td>1.0</td>
<td>1.4</td>
<td>1.43</td>
<td>0.26</td>
</tr>
<tr>
<td>2 turn short</td>
<td>150</td>
<td>147</td>
<td>145</td>
<td>145</td>
<td>0.99</td>
<td>3.0</td>
<td>1.0</td>
<td>2.2</td>
<td>2.06</td>
<td>0.58</td>
</tr>
<tr>
<td>3 turn short</td>
<td>147</td>
<td>147</td>
<td>142</td>
<td>145</td>
<td>0.66</td>
<td>4.5</td>
<td>1.6</td>
<td>3.1</td>
<td>3.06</td>
<td>0.83</td>
</tr>
</tbody>
</table>

(b) at 130 V level

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Line voltage $V_{ab}$ (V)</th>
<th>Line voltage $V_{bc}$ (V)</th>
<th>Line voltage $V_{ca}$ (V)</th>
<th>$V_{sp}$ (V)</th>
<th>$V_{sn}$ (V)</th>
<th>Line current $I_a$ (A)</th>
<th>Line current $I_b$ (A)</th>
<th>Line current $I_c$ (A)</th>
<th>$I_{sp}$ (A)</th>
<th>$I_{sn}$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>0</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
<td>0.96</td>
<td>0.03</td>
</tr>
<tr>
<td>1 turn short</td>
<td>128</td>
<td>130</td>
<td>125</td>
<td>128</td>
<td>1.33</td>
<td>1.6</td>
<td>1.1</td>
<td>1.1</td>
<td>1.26</td>
<td>0.16</td>
</tr>
<tr>
<td>2 turn short</td>
<td>128</td>
<td>130</td>
<td>125</td>
<td>128</td>
<td>1.33</td>
<td>2.5</td>
<td>1.2</td>
<td>1.75</td>
<td>1.81</td>
<td>0.38</td>
</tr>
<tr>
<td>3 turn short</td>
<td>127</td>
<td>127</td>
<td>123</td>
<td>126</td>
<td>1.33</td>
<td>3.8</td>
<td>1.4</td>
<td>2.6</td>
<td>2.60</td>
<td>0.40</td>
</tr>
</tbody>
</table>

(c) at 100 V level

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Line voltage $V_{ab}$ (V)</th>
<th>Line voltage $V_{bc}$ (V)</th>
<th>Line voltage $V_{ca}$ (V)</th>
<th>$V_{sp}$ (V)</th>
<th>$V_{sn}$ (V)</th>
<th>Line current $I_a$ (A)</th>
<th>Line current $I_b$ (A)</th>
<th>Line current $I_c$ (A)</th>
<th>$I_{sp}$ (A)</th>
<th>$I_{sn}$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>0.56</td>
<td>0.07</td>
</tr>
<tr>
<td>1 turn short</td>
<td>100</td>
<td>102</td>
<td>95</td>
<td>99</td>
<td>1.96</td>
<td>1.0</td>
<td>0.5</td>
<td>1.1</td>
<td>0.86</td>
<td>0.17</td>
</tr>
<tr>
<td>2 turn short</td>
<td>95</td>
<td>100</td>
<td>90</td>
<td>95</td>
<td>2.90</td>
<td>1.6</td>
<td>0.6</td>
<td>1.5</td>
<td>1.23</td>
<td>0.26</td>
</tr>
<tr>
<td>3 turn short</td>
<td>95</td>
<td>99</td>
<td>90</td>
<td>95</td>
<td>2.6</td>
<td>2.6</td>
<td>1.1</td>
<td>2.1</td>
<td>1.93</td>
<td>0.45</td>
</tr>
</tbody>
</table>

3.2.2.5 The FFD

This section describes the design and simulation of FFD for fault detection. The fault detection process is the mapping the $V_{sn}$ and $I_{sn}$ to fault condition ($F_c$).
3.2.2.5.1 Simulation

The FFD is constructed with two inputs and single output. The $V_{sn}$ and $I_{sn}$ are inputs of the FFD and are used to estimate the fault condition ($F_c$). The universe of discourse of $V_{sn}$ and $I_{sn}$ are defined as

$$V_{sn} = \{V_{sn} / V_{sln} \leq V_{sn} \leq V_{snu}\}$$
$$I_{sn} = \{I_{sn} / I_{sln} \leq I_{sn} \leq I_{snu}\}$$

where, $V_{sln}$, and $I_{sln}$ are lower limits of $V_{sn}$ and $I_{sn}$ respectively.

$V_{snu}$, $I_{snu}$ are upper limits of $V_{sn}$ and $I_{sn}$ respectively

Five fuzzy sets are defined on each of input and output spaces corresponding to very low, low, medium, high and very high for each variable. The output space $I_{sn}$ of the FFD is defined as

$$F_c = \{F_c / 0 \leq F_c \leq 3\}$$, where, $F_c$ represents the fault condition.

Simulation is carried out using MATLAB. The LOM defuzzification method is used for simulation. The fuzzy rules are derived from experimental value and expert experience. The derived fuzzy rules are given in Table 3.8. The membership functions used for simulation are shown in Figure 3.8. During training phase, the experimental data are used to tune the FFD.

Table 3.8 Fuzzy rules for $I_{sn}$ based fault detection scheme

<table>
<thead>
<tr>
<th>$I_{sn}$</th>
<th>$V_{sn}$</th>
<th>VL: Very Low</th>
<th>L: Low</th>
<th>M: Medium</th>
<th>H: High</th>
<th>VH: Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL</td>
<td>0 (Very low)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>1 (Low)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>2 (Medium)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>3 (High)</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VH</td>
<td>4 (Very high)</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

VL: Very Low    L: Low    M: Medium    H: High    VH: Very High
Figure 3.8  Triangular membership functions used for $I_{sn}$ based fault detection scheme
Table 3.9 shows the training samples obtained from the test machine at no load. Similar data can be obtained from the machine at different loads and can be used to train the FFD for effective fault detection.

**Table 3.9 Training data for $I_{sn}$ based fault detection scheme**

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Fault code</th>
<th>$V_{sn}$ (V)</th>
<th>$I_{sn}$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>0</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td>1 turn short</td>
<td>1</td>
<td>0.99</td>
<td>0.26</td>
</tr>
<tr>
<td>2 turn short</td>
<td>2</td>
<td>0.99</td>
<td>0.58</td>
</tr>
<tr>
<td>3 turn short</td>
<td>3</td>
<td>0.66</td>
<td>0.83</td>
</tr>
</tbody>
</table>

In testing phase, the inputs are presented to the FFD after providing compensation to determine the fault condition.

**3.2.2.5.2 Results and discussion**

The FFD is tested with triangular membership function membership functions and LOM defuzzification method. The error is calculated as the difference between the experimental value and the output of FFD. Table 3.10 shows the accuracy of FFD for triangular membership functions.

**Table 3.10 Test results of $I_{sn}$ based fault detection scheme**

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>$V_{sn}$ (V)</th>
<th>$I_{sn}$ (A)</th>
<th>Fault code</th>
<th>Output of FFD</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>0</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 turn short</td>
<td>0.99</td>
<td>0.26</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2 turn short</td>
<td>0.99</td>
<td>0.58</td>
<td>2</td>
<td>1.99</td>
<td>0.5</td>
</tr>
<tr>
<td>3 turn short</td>
<td>0.66</td>
<td>0.83</td>
<td>3</td>
<td>2.90</td>
<td>3.0</td>
</tr>
<tr>
<td><strong>Average Error</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>1.0</strong></td>
</tr>
</tbody>
</table>


From the Table 3.10, it is found that negative sequence current based fault detection is an effective method for fault detection. Therefore, this scheme can be applied for online monitoring of electrical motors. However, this method requires large amount of data to compensate the effect supply voltage variation and motor asymmetry. To overcome this drawback, $Z_{np}$ based fault detection scheme is presented in the next section.

3.2.3 $Z_{np}$ based fault detection scheme

The steady state sequence component voltage equation of a general three phase ac machine is given by (Krause 1987)

$$
\begin{bmatrix}
\tilde{V}_p \\
\tilde{V}_n \\
\tilde{V}_z
\end{bmatrix} =
\begin{bmatrix}
Z_{pp} & Z_{pn} & Z_{pz} \\
Z_{np} & Z_{nn} & Z_{nz} \\
Z_{zp} & Z_{zn} & Z_{zz}
\end{bmatrix} \begin{bmatrix}
\tilde{I}_p \\
\tilde{I}_n \\
\tilde{I}_z
\end{bmatrix}
$$

(3.2)

where, subscripts p, n, and z represent positive, negative, and zero sequence components of the voltage and current, and $Z_{ij}$ represent the impedance of the i sequence due to the j sequence. For an ideal induction machine, the off-diagonal terms of the impedance matrix in Equation (3.2), $Z_{pn}$, $Z_{pz}$, $Z_{np}$, $Z_{nz}$, $Z_{zp}$ and $Z_{zn}$ are zero. However, in actual machines, inherent asymmetries in the motor cause the off-diagonal terms to be nonzero and they are slip dependent.

From Equation (3.2), the negative sequence steady state voltage equation ($I_z = 0$) with inherent asymmetries is given by

$$
V_n = Z_{np}I_p + Z_{nn}I_n
$$

(3.3)

For a healthy motor with inherent asymmetry, $Z_{np}$ is a small slip dependent non-zero term. When a turn fault occurs, the value of $Z_{np}$ changes.
Therefore, stator winding turn faults can be detected by continuously monitoring $Z_{np}$ and comparing it with the value of $Z_{np}$ evaluated for a healthy motor, $Z_{np0}$, under the same slip condition. The change in $Z_{np}$ is used as a turn fault indicator and is given by

$$\Delta Z_{np} = Z_{np} - Z_{np0}$$  \hspace{1cm} (3.4)

To demonstrate this scheme, the winding study induction motor is used. The ratings of the test machine are given in Table 3.1.

### 3.2.3.1 Calculation of $Z_{np0}$ for healthy machine

To compensate the motor asymmetry, $Z_{np0}$ is calculated for a healthy machine. To calculate the value of $Z_{np0}$ at a particular slip, $s$, two data sets are required since the Equation (3.3) contains two unknown terms $Z_{np}$ and $Z_{nn}$.

The $Z_{np0}$ can be calculated using following equation:

$$Z_{np0} = \frac{\tilde{I}_n \cdot \tilde{V}_n \cdot \tilde{I}_n \cdot \tilde{V}_n}{\tilde{I}_{p1} \cdot \tilde{I}_{n2} - \tilde{I}_{p2} \cdot \tilde{I}_{n1}}$$  \hspace{1cm} (3.5)

where subscripts 1 and 2 represent the first and second sets of reading. Figure 3.9 shows the experimental set up used for calculating $Z_{np0}$. Normally one reading is taken under balanced condition (with switch’s’ closed) and another reading under unbalanced condition (with switch ‘S’ opened) maintaining same slip. Then the value of $Z_{np}$ is calculated for a healthy motor ($Z_{np0}$) using Equation (3.5). Table 3.11 shows experimental values of $Z_{np0}$ for different slips.

From the Table 3.11, the impedance of healthy machine, $Z_{np0}$ is calculated using the Equation (3.5). In the similar way, $Z_{np0}$ can be calculated from no load slip to rated slip. The values of $I_n$, $V_p$, $V_n$ and $Z_{np0}$ are stored as the function of slip.
3.2.3.2 Calculation of $Z_{np}$ under fault conditions

To calculate the $Z_{np}$ of the machine under fault condition, the winding study motor can be used. But, it is not possible to simulate turn faults more than four steps. This will increase the stator current beyond the rated value. Therefore, induction machine turn fault model (Tallam et al 2002) is used for calculating the $Z_{np}$ of faulty machine. The expression for $Z_{np}$ for induction machine is given by

$$Z_{np} = \frac{-Y_{np}Z_pZ_n}{1 + Y_{np}(Z_p + Z_n)}$$  \hspace{1cm} (3.6)

where, $Z_p$ and $Z_n$ are the ideal positive and negative sequence impedances and $Y_{ij}$ is the admittance of i sequence due to the j sequence. The expression for $Y_{np}$ is given as
\[ Y_{np} = Y_{pn} = \frac{\mu^2}{3(R_f + \mu (R_s + j\omega L_{ls}))} = \frac{\mu^2}{3(R_f + \mu Z_n)} \]  

(3.7)

where, \( Y_p \) and \( Y_n \) are the ideal positive and negative sequence admittances and \( Z_n \) is the ideal zero sequence impedance. The input electrical frequency is denoted by, \( \omega_e \). \( R_s \) and \( L_{ls} \) represents the stator resistance and leakage inductance respectively. \( R_f \) is the resistance in the faulted loop and \( \mu \) is the fraction of shorted turns. The value of \( Z_{np} \) increases with fault severity (increase in \( \mu \)) and it is slip-dependent.

From the Equation (3.6) and Equation (3.7), the values of \( Z_{np} \) for faulty machine for different values of slip can be calculated from the equivalent circuit of the induction motor.

The values of \( Z_{np} \) for different fault conditions at a slip, \( s = 0.034 \) are calculated and shown in Figure 3.10.

![Figure 3.10 The values of \( Z_{np} \) and \( \Delta Z_{np} \) for different ‘\( \mu \)’ at \( s = 0.034 \)](image-url)
3.2.3.3 Calculation of turn fault indicator, $\Delta Z_{np}$

From the calculated values of $Z_{np}$ of the machine for healthy and faulty conditions, the value of $\Delta Z_{np}$ can be calculated by using Equation (3.4) for any slip, $s$. It is very difficult to calculate the value of $Z_{np}$ online because it is difficult to get two sets of data at the same slip ‘$s$’. Therefore, one set of reading at slip ‘$s$’ can be noted. The other set of data corresponding to the same slip can be calculated from the stored data using linear interpolation technique. The Figure 3.10 shows the values of fault indicator, $\Delta Z_{np}$ at slip= 0.034 for various values of $\mu$.

3.2.3.4 Fuzzy fault detection scheme

The schematic of fuzzy fault detection scheme is shown in Figure 3.11. The line voltages and line currents measured from the three phase squirrel cage induction motor are used to calculate the sequence component impedance term $Z_{np}$. The calculated value of $Z_{np}$ at a particular slip ‘$s$’ is compared with the stored value of $Z_{np0}$ at the same slip. The difference between them ($\Delta Z_{np}$) is fed to FFD as fault indicator to analyze the fault condition.

Figure 3.11 Schematic of fuzzy fault detection scheme

The problem of $Z_{np}$ based turn fault detection is the difficulty of calculating $\Delta Z_{np}$ online. To calculate $Z_{np0}$, two sets of data points must be obtained as a function of slip when the supply unbalance is different from one
another. To overcome this, $Z_{np0}$ is calculated prior to motor operation and stored in memory as the function of slip. The next section describes the design and testing of FFD in simulation environment.

### 3.2.3.4.1 Simulation of FFD

The fault detection process is the mapping between $\Delta Z_{np}$ and fraction of shorted turns, $\mu$. The fuzzy model is constructed with single input and single output.

### 3.2.3.4.2 Training

Mamdani type of fuzzy inference system is used. Input variable, $\Delta Z_{np}$ is classified into four membership functions namely Very High, High, Medium and Low ranging from 0 to 2. The output variable $\mu$ is classified into four regions namely Very High, High, Medium, and Low ranging from 0 to 25.

The fuzzy rules are derived based on experimental values. Table 3.12 shows the rules framed for training the FFD. The data shown in Figure 3.10 are used as training data to illustrate the application of fuzzy system for fault detection. The FFD is trained with triangular membership functions and LOM defuzzification method. The membership functions used for training are shown in Figure 3.12.

**Table 3.12 Fuzzy rules for training the FFD**

<table>
<thead>
<tr>
<th>$\Delta Z_{np}$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High</td>
<td>Very High</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
3.2.3.4.3 Test results and discussion

Once the FFD is trained with experimental data, the FFD can be tested for fault detection. To study the effectiveness of fuzzy fault detection scheme, the FFD is tested for different values of input, \( \Delta Z_{np} \). The deviation of FFD output from the actual value is calculated as error. The test results are shown in Table 3.13.
Table 3.13 Simulation results of Z_{np} based fault detection scheme

<table>
<thead>
<tr>
<th>Expected Output μ (%)</th>
<th>Triangular</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Output of FFD</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Average Error</td>
<td></td>
</tr>
</tbody>
</table>

From the Table 3.13, it is found that Z_{np} based fault detection scheme is capable of monitoring the stator winding fault very closely. This scheme has the advantage of using single parameter to represent the fault condition. The fault indicator is immune to supply voltage variations and motor asymmetry.

3.2.4 Comparison of fault detection schemes

The performance of the three different fuzzy fault detection schemes for three phase induction is analysed quantitatively in terms of the average error. Table 3.14 shows the performance of three different fault detection schemes.

Table 3.14 Comparison of fault detection schemes

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Fault detection schemes</th>
<th>Number inputs</th>
<th>Number of test patterns</th>
<th>Average error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Line current based scheme</td>
<td>3</td>
<td>4</td>
<td>2.25</td>
</tr>
<tr>
<td>2</td>
<td>I_{sn} based scheme</td>
<td>2</td>
<td>4</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>Z_{np} based scheme</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
From the Table 3.14, it is inferred that the accuracy of \( Z_{np} \) based fuzzy fault detection scheme is comparable with other two schemes. Therefore, this scheme can be applied for online monitoring directly. This scheme has the following advantages

- The fault indicator is immune to supply voltage variations, motor asymmetry and measurement errors.
- This scheme utilizes one parameter as fault indicator for fault detection. Therefore, it is easy to monitor.
- This scheme does not require the large data to provide compensation for fault indicator.

Therefore, the performance of \( Z_{np} \) based fault detection scheme is also investigated using neural network because the neural network has very good learning capability. The next section describes the neural network based fault detection scheme for three phase induction motor.

### 3.3 NEURAL NETWORK BASED FAULT DETECTION SCHEME

#### 3.3.1 \( Z_{np} \) based fault detection scheme

This section describes the \( Z_{np} \) based fault detection scheme for three phase induction motor using neural network. The \( Z_{np} \) based fault detection scheme is previously described in section 3.2.3. The same scheme is implemented using neural network. Then, the performance of NNFD is analysed.
3.3.1.1 The structure of neural network for fault detection

The neural network is constructed with one input layer, one hidden layer and output layer. The neural network is trained by adjusting the numerical value of the weights between each unit. Once the neuron is trained, the network weights will contain the nonlinearity of the desired mapping, so that difficulties in the mathematical modeling can be avoided. The BP training algorithm is used to adjust the numerical values of the weights and the internal threshold of each neuron.

The network is trained initially by selecting small random weights and internal threshold and then presenting all training data. The connection weights and thresholds are adjusted after every training sample is presented to the network, until the error is reduced to acceptable value. Figure 3.13 shows the structure of the conventional BP neural network for fault detection. The network is constructed with one input layer, one hidden layer and output layer. There are eight neurons in the hidden layer, one neuron in output layer and one neuron in the output layer. The sigmoid function is used as activation function for all layers. The bias value is set to +1.

![Figure 3.13 The structure of neural network for fault detection](image-url)
The input to the neural network is $Z_{np}$. The output is the fraction of shorted turns ($\mu$). The condition of the winding is calculated in terms of fraction of shorted turns. For effective fault detection, the neural network requires training.

### 3.3.1.2 Training phase

In training phase, neural network learns the negative sequence impedance of a healthy machine under different load conditions including the effect of inherent asymmetries. The line voltages and currents are measured and the positive and negative sequence components of line currents and the positive and the negative sequence components of line voltages are calculated. The value of $Z_{np}$ is calculated and stored as the function of slip.

![Diagram of Training process of NNFD](image)

**Figure 3.14 Training process of NNFD**
The network is trained with various values of $Z_{np}$ for various loads. The training process of NNFD is shown in Figure 3.14. Training data for NNFD is shown in Figure 3.10. In addition to this, the data shown in Figure 3.15 are also used for training.

![Figure 3.15 Training data for the neural network](image)

**Figure 3.15 Training data for the neural network**

There are various steps involved in training the neural network and these steps are explained as shown in Figure 3.16.

### 3.3.1.3 Monitoring phase

In the monitoring phase, the trained NNFD is used to estimate the fault severity in terms of fraction of shorted turns ($\mu$). As stated previously, the value of $Z_{np}$ is calculated online from the presently measured set of data and the stored data that corresponds to the same slip. Then the neural network estimates the output in terms of fraction of shorted turns, which indicates the winding condition.
3.3.1.4 Results and discussion

In induction machine, the value of $Z_{np}$ is slip dependent. From the Figure 3.10, it is clear that, during fault condition at a particular slip, the magnitude of $Z_{np}$ increases as the fraction of shorted turns increases. Therefore, the magnitude of $Z_{np}$ is used for training and testing the neural network. To train the neural network, eighty training data and twenty testing data are used.

The BP algorithm is used for training. The bipolar sigmoid function activation function is used. Figure 3.17 shows the error Vs epoch characteristic of NNFD.
The NNFD is tested with the experimental data obtained from the winding study machine. The test results are given in Table 3.15.

Table 3.15 Test results of NNFD for $Z_{\text{np}}$ based fault detection scheme

<table>
<thead>
<tr>
<th>Expected Output $\mu$ (%)</th>
<th>Output of NNFD</th>
<th>(%)Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Average Error</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
From the Table 3.15, it is found that the NNFD is capable of detecting the stator winding fault effectively. The performance of neural network based fault detection scheme is comparable with fuzzy based fault detection scheme.

To avoid false indication of fault, NNFD can be trained with different data corresponding to different fault conditions. This requires the more training time. To minimize the training time of the conventional BP algorithm, the modified BP algorithm can be used. To demonstrate modified BP algorithm for fault detection, the line currents based scheme is considered. The next section describes the application of modified BP algorithm for motor fault detection.

3.3.2 Line currents based fault detection scheme

3.3.2.1 Introduction

Invariably the conventional BP networks have been used in many engineering applications because of their learning and generalization capability. The conventional BP network gets struck in local minima. Therefore the training time is increased (Fausett 1994). Yamamota et al (2000) and Van Ooyen (1992) have suggested the modifications in learning algorithm to improve the convergence. In Somasundareswari et al (2001), the modified BP algorithm was presented for solving XOR problem. This algorithm has used the technique called IWU scheme. This scheme updates weight as soon as error term is calculated. In this thesis, a modified BP network is used for motor fault detection. The performance of this algorithm for fault detection is compared with conventional BP algorithm.
3.3.2.2 Description of fault detection problem

The turn fault in the stator winding creates the unbalanced line currents. Therefore, line currents can be monitored to detect the turn faults. The fault detection process is the mapping between the normalized inputs and target outputs, which represents the different fault conditions. The training samples for different faults are obtained by simulating the faults externally in the winding study test motor.

3.3.2.3 Fault simulation

To study the efficiency of modified BP algorithm, the different experimental data is obtained from winding study motor. The ratings of the test motor are given in Table 3.1. The motor has the provision to create winding fault externally. The motor is connected through Variac to operate the motor at reduced voltage to limit the line current. Different faults are created and line currents are measured. The measured line currents for various faults are given in the Table 3.16. The Table 3.16 is used as training samples for the neural network.

Table 3.16 Training data for IWU scheme

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>Line current $I_R$ (A)</th>
<th>Line current $I_Y$ (A)</th>
<th>Line current $I_B$ (A)</th>
<th>Status of the winding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.60</td>
<td>2.60</td>
<td>2.50</td>
<td>Normal</td>
</tr>
<tr>
<td>2</td>
<td>3.00</td>
<td>2.70</td>
<td>2.40</td>
<td>One turn short in R phase winding</td>
</tr>
<tr>
<td>3</td>
<td>2.50</td>
<td>3.00</td>
<td>2.70</td>
<td>One turn short in Y phase winding</td>
</tr>
<tr>
<td>4</td>
<td>2.60</td>
<td>2.70</td>
<td>3.00</td>
<td>One turn short in B phase winding</td>
</tr>
<tr>
<td>5</td>
<td>3.90</td>
<td>3.00</td>
<td>3.80</td>
<td>One turn short in R &amp; B phase winding</td>
</tr>
<tr>
<td>6</td>
<td>3.00</td>
<td>3.90</td>
<td>3.90</td>
<td>One turn short in Y &amp; B phase winding</td>
</tr>
<tr>
<td>7</td>
<td>3.90</td>
<td>3.90</td>
<td>2.90</td>
<td>One turn short in R &amp; Y phase winding</td>
</tr>
<tr>
<td>8</td>
<td>4.20</td>
<td>4.00</td>
<td>3.80</td>
<td>One turn short in R,Y &amp; B phase winding</td>
</tr>
</tbody>
</table>
3.3.2.4 The structure of FFNN

The structure of FFNN for fault detection is constructed with three input neurons, five neurons in hidden layer and four neurons in output layer as shown in the Figure 3.18. Since the training and testing algorithms are implemented using ‘C’ language, the number of neurons, activation functions and number of hidden layers can be changed according to the requirements.

3.3.2.5 Modified BP algorithm for fault diagnosis

The network shown in Figure 3.18 is used for training and testing the modified BP algorithm. When the network is trained with conventional BP algorithm, the weights are updated either by batch updation method or by
per epoch updation method. In the modified BP algorithm, the weights that are connected from the hidden layer to the output layer are updated immediately so that the error correction factor for the hidden neurons includes the effect of updated weights. This algorithm improves the stabilization of the network. The training and testing programs are implemented using ‘C’ language. The modified BP algorithm for fault detection is given below. The nomenclature for this algorithm is already defined in chapter 1.

3.3.2.5.1 Training algorithm

Step1: The weights are initialized randomly between –0.5 to +0.5 and normalized.

Step2: While the sum squared error is greater than the tolerance, do steps 3 to 8.

Step3: For each training pair, do steps 4 to 7.

Step4: Compute the output signal for the hidden units \(( j = 1,2, \ldots, p)\)

\[
z_{inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{jk} \\
z_j = f(z_{inj})
\]

Step5: Compute the output signal for the output units \(( k = 1,2,\ldots, m)\)

\[
y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk} \\
y_k = f(y_{ink})
\]

Step 6: Compute

\[
w_{jk} = \alpha \delta_k z_j + \mu_i \Delta w_{jk} \text{ (old)}
\]

where, \(\delta_k = (t_k - y_k) f'(y_{ink}), \ k = 1,2, \ldots, m\)

\[
w_{jk} \text{ (new)} = w_{jk} \text{ (old)} + \Delta w_{jk}
\]
Step 7: Compute

\[ \delta_j = \sum \delta_k w_{jk} f'(z_{inj}) \quad j = 1,2,\ldots,p \]

\[ \Delta v_{ij} = \alpha \delta_j x_i + \mu_i \Delta v_{ij} \text{(old)} \]

\[ v_{ij} \text{(new)} = v_{ij} \text{(old)} + \Delta v_{ij} \]

Step 8: Test for Stopping Condition.

The weights from hidden to output layer are updated as soon as the change in weight is calculated. This updated weight is utilized in the error term, which is back propagated to the subsequent layer.

3.3.2.5.2 Simulation results and discussion

The bias neurons for both the hidden and the output neurons are set to 1. The initial weights are randomised between -0.5 to +0.5 and normalized. The hidden neurons are activated by binary sigmoid functions and the output neurons are activated by binary sigmoid functions.

Table 3.16 is used as training data for the network. The line currents are taken as inputs and binary codes are taken as output, which represents the corresponding fault condition. The fault detection process is the mapping between the motor line currents and fault codes. The input samples are normalised between 0 and 1. The network is trained with random initial weights, learning rate and momentum factor. The learning rate and momentum factor are modified for effective convergence. The network is trained with modified BP algorithm until the sum-squared error is equal to 0.01. The network is converged for learning rate, \( \alpha = 0.35 \) and momentum factor, \( \mu_i = 0.9 \). 1000 sets weight samples are used for training. Once network learned the input patterns, the final weights are stored and used for testing. The testing results are given in Table 3.17. The effectiveness of network training can be improved by increasing the number of hidden layers/neurons.
Table 3.17  Simulation results of modified BP algorithm

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>Line current $I_R$ (A)</th>
<th>Line current $I_Y$ (A)</th>
<th>Line current $I_B$ (A)</th>
<th>Line current $I_R$ (A)</th>
<th>Status of the winding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.60</td>
<td>2.60</td>
<td>2.50</td>
<td>0000</td>
<td>Normal</td>
</tr>
<tr>
<td>2</td>
<td>3.00</td>
<td>2.70</td>
<td>2.40</td>
<td>0100</td>
<td>One turn short in R phase winding</td>
</tr>
<tr>
<td>3</td>
<td>2.50</td>
<td>3.00</td>
<td>2.70</td>
<td>0101</td>
<td>One turn short in Y phase winding</td>
</tr>
<tr>
<td>4</td>
<td>2.60</td>
<td>2.70</td>
<td>3.00</td>
<td>0110</td>
<td>One turn short in B phase winding</td>
</tr>
<tr>
<td>5</td>
<td>3.90</td>
<td>3.00</td>
<td>3.80</td>
<td>0111</td>
<td>One turn short in R &amp; B phase winding</td>
</tr>
<tr>
<td>6</td>
<td>3.00</td>
<td>3.90</td>
<td>3.90</td>
<td>1000</td>
<td>One turn short in Y &amp; B phase winding</td>
</tr>
<tr>
<td>7</td>
<td>3.90</td>
<td>3.90</td>
<td>2.90</td>
<td>1001</td>
<td>One turn short in R &amp; Y phase winding</td>
</tr>
<tr>
<td>8</td>
<td>4.20</td>
<td>4.00</td>
<td>3.80</td>
<td>1010</td>
<td>One turn short in R, Y &amp; B phase winding</td>
</tr>
</tbody>
</table>

The modified BP algorithm is compared with the conventional BP algorithm in terms of number of epochs, time and number of failures. The network is trained with modified BP algorithm for different values of momentum factor and learning rate and compared with conventional BP algorithm. The training results of modified BP algorithm with binary sigmoid activation functions are given in Table 3.18. The error Vs epoch characteristics for BP and modified BP algorithms for binary sigmoid activation functions are shown in Figure 3.19.

The modified BP algorithm is also tested with tangential activation functions in hidden and output neurons and compared with conventional BP algorithm. Table 3.19 shows the test results for tangential activation functions. The error Vs epoch characteristics for BP and modified BP algorithms for tangential activation functions are shown in Figure 3.20.
Table 3.18  Comparison of modified BP and BP algorithms with binary sigmoid activation functions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of epochs</th>
<th>Time in ms</th>
<th>Number of failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional BP</td>
<td>722</td>
<td>79.43</td>
<td>32</td>
</tr>
<tr>
<td>Modified BP</td>
<td>688</td>
<td>75.63</td>
<td>14</td>
</tr>
</tbody>
</table>

Figure 3.19  Error Vs epoch characteristics for binary sigmoid activation functions
Table 3.19  Comparison of modified BP and BP algorithms with tangential activation functions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of epochs</th>
<th>Time in ms</th>
<th>Number of failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional BP</td>
<td>429</td>
<td>47.17</td>
<td>11</td>
</tr>
<tr>
<td>Modified BP</td>
<td>406</td>
<td>44.65</td>
<td>01</td>
</tr>
</tbody>
</table>

From the simulation results, it is inferred that the performance of the modified BP algorithm in learning the fault detection process is comparable with conventional BP procedure.
3.4 IMPLEMENTATION OF FAULT DETECTION SCHEME

A prototype model is developed to implement the $Z_{np}$ based fault detection scheme. The $Z_{np}$ based fault detection scheme is implemented experimentally using neural network. The schematic of neural network based fault detection scheme is shown in Figure 3.21. The parameters and ratings of

![Figure 3.21 The schematic of neural network based fault detection scheme](image-url)
the test motor are shown in Table 3.1. The motor is connected to the three phase supply through variable resistor. The operation of the NNFD can be classified into training phase and the monitoring phase. In the training phase, the neural network is trained with the stored set of data. The weights of the network are stored once the training is completed.

In the monitoring phase, the online values of current, voltage and speed are monitored. The LCD displays voltage, current and speed continuously. These values are also available at the serial port of PC. The sequence components are calculated using MATLAB program. The calculated data are given as input to the neural network and the output is obtained in terms of fraction of shorted turns.

The micro-controller and the LCDs require a power supply of +5 volts. The operational amplifier requires a dual power supply of +12 volts and –12 volts. With suitable voltage regulators, the power supply unit is designed.

The line voltages and currents are measured using potential transformers and current transformers respectively. The output of the potential transformer is then given to the inverting amplifier, which acts as precision rectifier. The output is then converted into the positive value using inverting amplifier. The measurement of current is similar to voltage measurement except that the output of current transformer is converted into ac voltage by connecting a shunt resistor across the current transformer.

The speed of the induction motor is sensed using proximity sensor. From the proximity sensor, the time taken for one revolution is obtained and thereby speed in terms of rpm is calculated.

These parameters are then fed to micro-controller. The analog values are converted into digital values by the in-built A/D converter of 10-bit
resolution. The analog values are sampled at the rate of 0.2 microseconds. The micro-controller is programmed using ‘C’ Language and is compiled using Hi-tech C. The parameters are displayed and the data are also made available at the serial port of PC by means of RS232 interface. The serial data are taken as input. The MATLAB processes the serial data and produce the fraction of shorted turns as output.

3.4.1 Hardware description

The power supply is the basic building block for the entire hardware unit.

Figure 3.22 5V Power supply circuit

Figure 3.23 12V Dual power supply circuit
Figure 3.22 shows the 5V power supply circuit and Figure 3.23 shows the 12V dual power supply circuit. Power supply unit consists of a step down transformer, 230V/15-0-15V and 230V/0-9V used to perform the step down operation. The current rating of transformer is 1.5A.

A bridge rectifier configuration is used with IN 4007 diodes. The output from the rectifier is filtered. Filtering unit consists of capacitor of 1000µf/25V to reduce ripples and a capacitor of 10µf/25V to maintain constant voltage. Voltage regulator IC7805 is used for +5V dc supply. IC7812 and 7912 is used for +12V and -12V dc supply respectively. The specification of power supply is given below.

Voltage drop across the diodes = 2*0.7 = 1.4V.
Input voltage = 15V

Without capacitor
\[ V_{avg} = (15-1.4) \text{ V} \]
\[ V = 13.6 \text{ V} \]

With capacitor
\[ V = V_{avg}*1.11 \]
\[ V = 15.06 \text{ V} \]

Input voltage = 9V

Without capacitor
\[ V_{avg} = (9-1.4) \text{ V} \]
\[ V = 7.6 \text{ V} \]

With capacitor
\[ V = V_{avg}*1.11 \]
\[ V = 8.83 \text{ V} \]

With 7812 voltage regulator: \( V_o = +12 \text{ V} \)
With 7912 voltage regulator: \( V_o = -12 \text{ V} \)
With 7805 voltage regulator: \( V_o = +5 \text{ V} \)

**3.4.1.1 Measurement of voltage**

The circuit diagram for voltage measurement is shown in Figure 3.24. Three potential transformers are connected across three phase supply, which is to be monitored. The potential transformer is rated for
230V/6V. The ac output voltage of the potential transformer is rectified, filtered and converted into pure dc by using precision rectifier.

![Figure 3.24 Voltage measuring circuit](image)

In the precision rectifier circuit, A is an inverting rectifier. The output from A is added to the original input signal in B (summing mixer). Polarity change of $E_{in}$ results in no output at $E_1$ due to the rectification. $E_{in}$ feeds B through a 20K ohms resistor and $E_1$ feeds B through a 10K ohms resistor.

The net effect of this scaling is that, for equal amplitudes of $E_{in}$ and $E_1$. $E_1$ will produce twice as much current flow into the summing point. This fact is used as an advantage here, as the negative alteration of $E_1$ produces twice the input current of half the amplitude, which $E_1$ alone would generate due to the subtraction of $E_{in}$. It is the equivalent of having $E_1$ feed through a 20 K ohms input resistor and having $E_{in}$ non-existing during this half cycle and it results in a positive output at B.
3.4.1.2 Measurement of current

The current measuring circuit is shown in Figure 3.25. The current transformers are connected in series with the motor. The current transformer is rated at 5A/150mA. The shunt resistor of 56 ohm is connected across the current transformer. The voltage across the resistor is proportional to the current flowing through the primary of the current transformer. This voltage is fed to the rectifier unit.

![Current measuring circuit](image)

**Figure 3.25 Current measuring circuit**

3.4.1.3 Measurement of speed

The speed sensing circuit is shown in Figure 3.26. The proximity sensor is used for speed measurement. The main advantage of proximity sensor is simple in construction, less maintenance, easy to calibrate and inexpensive. A magnetic pickup sensor is placed near the strip, which is fixed on the rotor whose speed is to be measured.
When the rotor rotates, an electromotive force is induced in the pick-up coil due to the change in reluctance of air gap between pick-up coil and strip. The output will be in pulses and the time difference between the pulses are calculated and then speed is calculated.

3.4.1.4 Peripheral Interface Controller (PIC)

The sensed signals are digitalized using the in-built A/D converter in the micro-controller. The digital value is used by the PIC for the fault detection. Then the digital value is converted into equivalent numeric value and it is displayed in the LCD. These data are also interfaced to PC.

The PIC interface diagram is shown in Figure 3.27. The motor parameters are interfaced to the PIC via port. The port assignment is shown in Table 3.20.
Figure 3.27 PIC interface diagram

Table 3.20 Port assignment of PIC 16F877

<table>
<thead>
<tr>
<th>Pin</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>R phase voltage (V₁)</td>
</tr>
<tr>
<td>RA1</td>
<td>R phase current (I₁)</td>
</tr>
<tr>
<td>RA2</td>
<td>Y phase voltage (V₂)</td>
</tr>
<tr>
<td>RA3</td>
<td>Y phase current (I₂)</td>
</tr>
<tr>
<td>RA4</td>
<td>B phase voltage (V₃)</td>
</tr>
<tr>
<td>RA5</td>
<td>B phase current (I₃)</td>
</tr>
<tr>
<td>RB0</td>
<td>Speed(ω)</td>
</tr>
<tr>
<td>RC0-RC2</td>
<td>LCD control bit</td>
</tr>
<tr>
<td>RC6,RC7</td>
<td>PC Interface</td>
</tr>
<tr>
<td>RD0-RD7</td>
<td>LCD Data bit</td>
</tr>
</tbody>
</table>
3.4.1.5 RS232 communication interface

The most common communication interface for short distance is RS232. RS232 defines a serial communication for one device to one computer communication port, with speeds up to 19,200 baud. Typically 7 or 8 bits (on/off) signal is transmitted to represent a character or digit.

![Figure 3.28 System Interface IC - Max-232](image)

The IC diagram of MAX 232 is shown in Figure 3.28. Each of the two transmitters is a CMOS inverter with +10V power supply. The input is TTL and CMOS compatible with a logic threshold of about 26% of \( V_{cc} \). If an unused transmitter section is left unconnected, the internal 400K\( \Omega \) pull up resistor connected between the transistor input and \( V_{cc} \) will pull the input high forming the unused transistor output low. The open circuit output voltage swing is guaranteed to meet the RS232 specification. The slew rate at output is limited to 30V / \( \mu \)s at \( V_{cc} = 0V \). The outputs are short circuit protected and can be short circuited to ground indefinitely.
**Receiver section**

The two receivers fully confirm to RS232 specifications. Their input impedance is 3KΩ either with or without 5V power applied and their switching threshold is within +3V of RS232 specification. To ensure compatibility with either RS232 IIP or TTL/CMOS input, the MAX232 receivers have VIL of 0.8V and VIH of 2.4V. The receivers have 0.5V of hysteresis to improve noise rejection. The TTL/CMOS compatible output of receiver will be low whenever the RS232 input is greater than 2.4V. The receiver output will be high when input is floating or driven between +0.8V and −30V.

The specification of MAX232 is given below.

\[ V_{cc} = 6V \quad V+ = 12V \quad V- = 12V \]

**Input voltage**

\[ T_{1in}, T_{2in} = -0.3 \text{ to } (V_{cc} + 0.3V) \]
\[ R_{1in}, R_{2in} = +30V \text{ or } -30V \]

**Output voltage**

\[ T_{1out}, T_{2out} = ((V+) + 0.3V) \text{ to } ((V-) + 0.3V) \]
\[ R_{1out}, R_{2out} = -0.3V \text{ to } (Vcc + 0.3V) \]

Power dissipation = 375 mW

Output resistance = 300Ω
3.4.2 Software description

3.4.2.1 Algorithm for programming PIC

Microcontroller has many built in features to support any application. The PIC 16F877 micro-controller is used for this application. Some of the features of PIC micro-controller are given below:

1. Improved architecture – Harvard Architecture
2. Three built-in timers
3. Built-in A/D converters
4. Separate program and data memory
5. High frequency operation
6. Flexible programming
7. Instruction pipelining
8. Five I/O programmable ports
9. Easy interfacing facilities

Harvard Architecture has a program memory and a data memory, which are accessed from separate buses. This improves bandwidth over traditional Van Neumann architecture. These separate buses allow one instruction to execute while the next instruction is fetched. The main advantage of using the PIC micro-controller is the reduced instruction set. When an instruction is well designed and highly orthogonal (symmetric), few instructions are required to perform the needed task.

PIC micro-controller contains an 8-bit Arithmetic Logic Unit (ALU) and an 8-bit working register. It performs arithmetic and boolean functions between the data in the working register and any register file. Depending on the instruction executed, the ALU may affect the values of the carry, digit carry and zero bits in the status register. PIC micro-controller consists of a Central Processing Unit (CPU), which is responsible for using the information
in the program memory to control the operation of the device. The CPU works in conjunction with the ALU to execute the arithmetic and logical operations. The CPU controls the program memory address bus, data memory address bus and provides access to stack register.

The program memory has a 13-bit program counter capable of addressing $8K \times 14$ program memory space. The width of the program memory bus is 14 bits. The data memory is made up of Special Function Register (SFR) and the General Purpose Register (GPR). The SFR controls the operation of the device, while GPR provides space for data storage and scratch pad operation. The data memory is portioned into 4 banks. Each bank contains general-purpose and special function registers. Switching between these banks requires the configuring of RP0 and RP1 bits in the STATUS register. The IRP bit in the STATUS register is used for indirect addressing. Table 3.21 shows the bits that have to be configured for bank selection.

### Table 3.21  Bank selection in PIC 16F877

<table>
<thead>
<tr>
<th>Accessed Bank</th>
<th>Direct (RP1: RP0)</th>
<th>Indirect (IRP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

The A/D converter allows the conversion of an analog input signal to the corresponding 8-bit digital number. The A/D converter has eight analog inputs. The output of the sample and hold is the input to the converter. The result is obtained by successive approximation. The A/D module has three registers.
They are

1. A/D result register (ADRES)
2. A/D control register0 (ADCON0) and
3. A/D control register1 (ADCON1)

The PIC micro-controller has three timers namely Timer0, Timer1 and Timer2. Timer1 is a 16bit timer and counter, consisting of two 8-bit registers. It can be turned on and off using the TMR1ON control bit. Timer2 is an 8-bit timer with a prescaler, a postscaler and period register. Timer0 is an 8-bit timer/counter, which gets incremented either on the rising or the falling edge of the clock pulse.

The algorithm for micro-controller program is as follows:

Step 1: Start the program.
Step 2: Assign the I/O ports depending upon the requirement.
Step 3: Initialize the ADCON registers.
Step 4: Enable the global interrupt and peripheral interrupt.
Step 5: The voltage measured in each of the three phase is multiplied by suitable value to obtain the original value.
Step 6: Similar procedure is repeated for current.
Step 7: Pulses are counted and the speed is calculated.
Step 8: The values of voltage, current and speed are given to the LCD display.
Step 9: The measured values are given to the serial port of PC.
Step 10: The PC takes the data and calculates the fault.
Step 11: If the severity of the fault is more, the fault has to be cleared to reset the program.
Step 12: Else step 5 to step 10 is repeated continuously.
3.4.2.2 Simulation of NN using MATLAB

MATLAB is a software package, which performs numeric computation and visualization. MATLAB is widely used, because of its matrix/vector notation and it is convenient to experiment with NN. Many of the important features of NN become apparent only for large-scale problems, which are computationally intensive and not feasible for hand calculations. With MATLAB, NN can be implemented and large-scale problems can be tested conveniently.

The algorithm for programming in MATLAB is as follows:

Step 1: Define the structure of the network.
Step 2: Initialize the weights and biases of the network.
Step 3: Provide the network with inputs and targets.
Step 4: State the number of epochs, learning rate and the error tolerance.
Step 5: Train the network for the input and the corresponding targets.
Step 6: Check for the error convergence else repeat step 5 till the error converges to the specified value.
Step 7: Check whether number of epochs is reached. Stop training if the specified number of epochs is reached else continue step 4 to step 6.
Step 8: Get the input data from the serial port.
Step 9: Calculate the negative sequence impedance.
Step 10: Provide negative sequence impedance and slip as inputs to the network.
Step 11: Display the fraction of shorted turns.
3.4.3 Experimental Results

The experimental setup for the implementation of fault detection scheme is shown in Figure 3.29. The test machine is tested for different load conditions and values of line currents, line voltages, sequence impedance and fraction of shorted turns for the test machine are given in Table 3.22.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Slip ( 's' )</th>
<th>Line voltage ( V_{kk} ) (V)</th>
<th>Line voltage ( V_{kr} ) (V)</th>
<th>Line Current ( I_k ) (A)</th>
<th>Line Current ( I_y ) (A)</th>
<th>Line Current ( I_b ) (A)</th>
<th>Impedance, ( Z_{np} ) (ohm)</th>
<th>Fraction of shorted turns ( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.045</td>
<td>238</td>
<td>239</td>
<td>237</td>
<td>0.8</td>
<td>0.9</td>
<td>-0.1340 + 0.7352i</td>
<td>0.0053</td>
</tr>
<tr>
<td>2</td>
<td>0.045</td>
<td>239</td>
<td>237</td>
<td>238</td>
<td>0.8</td>
<td>0.9</td>
<td>0.3686 + 0.1432i</td>
<td>-0.0149</td>
</tr>
<tr>
<td>3</td>
<td>0.045</td>
<td>237</td>
<td>237</td>
<td>238</td>
<td>0.8</td>
<td>0.9</td>
<td>-0.3954 + 0.0039i</td>
<td>0.0155</td>
</tr>
<tr>
<td>4</td>
<td>0.054</td>
<td>239</td>
<td>236</td>
<td>240</td>
<td>1.1</td>
<td>1.2</td>
<td>0.2791 – 0.4502i</td>
<td>-0.0119</td>
</tr>
<tr>
<td>5</td>
<td>0.071</td>
<td>239</td>
<td>236</td>
<td>241</td>
<td>1.6</td>
<td>1.7</td>
<td>0.1362 – 0.4970i</td>
<td>-0.0072</td>
</tr>
<tr>
<td>6</td>
<td>0.057</td>
<td>240</td>
<td>235</td>
<td>240</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9259 – 1.6038i</td>
<td>-0.0419</td>
</tr>
<tr>
<td>7</td>
<td>0.070</td>
<td>238</td>
<td>237</td>
<td>241</td>
<td>1.5</td>
<td>1.6</td>
<td>-0.1219 – 0.4057i</td>
<td>0.0037</td>
</tr>
<tr>
<td>8</td>
<td>0.080</td>
<td>238</td>
<td>239</td>
<td>238</td>
<td>1.9</td>
<td>2.0</td>
<td>-0.1786 + 0.3093i</td>
<td>0.0057</td>
</tr>
<tr>
<td>9</td>
<td>0.100</td>
<td>239</td>
<td>237</td>
<td>238</td>
<td>2.4</td>
<td>2.5</td>
<td>0.0505 + 0.0875i</td>
<td>-0.0031</td>
</tr>
</tbody>
</table>

The negative value of \( \mu \) and the value of \( \mu < 0.005 \) indicate that there are no shorted turns in the stator winding. The value of \( \mu > 0.005 \) indicates the presence of shorted turns. As the severity of the fault increases, the value of shorted turns (\( \mu \)) increases. Therefore, this scheme can be directly applied to induction motor for online fault detection.
Figure 3.29 The experimental set up for neural network based fault detection scheme
3.5 SUMMARY

In this chapter, three different fault detection schemes are demonstrated using fuzzy logic. The performance of these schemes is compared in terms of accuracy in fault detection. From the simulation results, it is found that the performance of impedance based fault detection scheme is comparable with other two schemes. This scheme is immune to supply voltage variations, motor asymmetry and measurements errors.

The impedance based fault detection scheme is demonstrated using neural network and the performance of this scheme is analyzed in terms of average error. From the simulation results, it is found that the performance of neural network based fault detection scheme is comparable with fuzzy based fault detection scheme. However, the NNFD requires large amount of data to be stored to avoid fault indication. This requires more training time.

To minimize the training time, the modified BP training algorithm is presented. To test the efficiency of the modified BP training algorithm, the line currents based fault detection scheme is studied. The performance of the conventional BP algorithm and modified BP algorithm is compared in terms of training time and number of epochs. From the test results, it is found that the performance of modified BP algorithm is comparable with conventional BP algorithm.

A prototype type model is developed to implement the impedance based fault detection scheme. The performance of the impedance based fault detection scheme is verified experimentally.