CHAPTER 3

FUZZY PID SUPERVISED ONLINE RADIAL BASIS FUNCTION NEURAL NETWORK BASED SPEED CONTROLLER FOR BRUSHLESS DC MOTOR

3.1 INTRODUCTION

In this chapter, fuzzy PID supervised online Radial Basis Function Neural Network (RBFNN) based speed controller for Brushless DC motor is presented. The fuzzy PID controller is acting as supervisor for online RBFNN controller. The overall speed control system is created using MATLAB/ Simulink toolbox. Dynamic speed response for varying loading and varying set speed condition are analyzed for Brushless DC motor with conventional PID controller and fuzzy PID supervised online RBFNN controller. For effective comparison of controllers, the following control system performance parameters such as rise time, peak overshoot, recovery time, steady state error, integral of absolute error and integral of time multiplied absolute error of speed response are measured and analyzed for the above considered controllers.

3.2 DESIGN APPROACH OF FUZZY PID SUPERVISED ONLINE RADIAL BASIS FUNCTION NEURAL NETWORK CONTROLLER

The structure of fuzzy PID supervised online RBFNN based speed controller for Brushless DC motor is shown in Figure 3.1. Rotor position sensor and speed sensor used to measure the actual rotor position and the speed of the motor. The speed error (e) is obtained by comparing reference speed with actual speed. The rate of change of speed error (Δe) is produced by differentiating the speed error. The e and Δe are the input to the online RBFNN controller. This controller also receives the supervised error (e_s) by comparing the fuzzy PID supervised algorithm output (U_F) and output of the online RBFNN controller (U_a). Based on this supervised error, the
parameter of online RBFNN is updated in the network. The switching logic and pulse width modulation (PWM) inverter receives the signal from the controller and rotor position sensor. The switching logic circuit provides the PWM signal for inverter based on controller output and rotor position. The speed of the motor is controlled by controlling the DC bus voltage by means of triggering the switches in the PWM inverter.

Figure 3.1 Fuzzy PID supervised online RBFNN based speed controller for Brushless DC motor

3.3 DEVELOPMENT OF FUZZY PID SUPERVISED ALGORITHM

In this section, the development of fuzzy PID supervised algorithm is presented for online RBFNN. Figure 3.2 shows the basic block diagram for fuzzy supervised PID algorithm. In this figure, fuzzy tuner is providing the gain values for the PID controller with respect to the speed error and rate of change of error.
Figure 3.2 Block Diagram for fuzzy PID supervised algorithm

Fuzzy tuner for PID controller is modeled by mamdani fuzzy inference system. Figure 3.3 shows the structure of fuzzy tuner for PID controller. It has two inputs i.e., speed error (e) and the rate of change of error (Δe) and three outputs (K_p, K_i, and K_d). Each input has five membership functions with bell shape norm and outputs have five membership functions with gaussian norm.

Figure 3.3 The structure of fuzzy tuner for PID controller

Distribution of input membership function is shown in Figure 3.4. The universal bell shaped membership function is expressed by equation (3.1) as,
\[ f(x; a_1, b_1, c_1) = \frac{1}{1 + \left| \frac{x - c_1}{a_1} \right|^{2b_1}} \]  \tag{3.1}

Figure 3.4 Distribution of input membership functions for error (rpm) and rate of change of error (rev/s²)

The universal bell shaped function depends on three parameters \( a_1, b_1 \) and \( c_1 \) where ‘\( a_1 \)’ denotes the half width, ‘\( b_1 \)’ controls the slopes at the intersecting points and ‘\( c_1 \)’ determines the centre of the corresponding membership function. The input range for error and rate of change error are from -1500 to 1500 and distributed with five bell shaped membership functions denoted by Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive Big (PB).

Similarly each output is distributed with five gaussian shaped membership functions. Gaussian membership function has been used to ease the design task. The gaussian membership function has been widely used with mamdani fuzzy inference system and has shown good performance in many applications. Besides, the use of gaussian membership function ensures that the whole output space is covered by fuzzy rules and avoids the zero firing strength problems. The generalized gaussian function is expressed in the equation (3.2) as,

\[ f(x; \sigma_1, c_2) = e^{-\frac{(x-c_2)^2}{2\sigma_1^2}} \]  \tag{3.2}
Where, $c_2$ and $\sigma_1$ represents the center and width of the membership function. Five gaussian membership functions are denoted by Small (S), Medium (M), Big (B), Very Big (VB), Very-Very Big (VVB). The range of proportional gain is from 0 to 5, integral gain ranges from -1 to 1, and derivative gain ranges from -1 to 0. Totally 25 rules are created for fuzzy tuner and few rules are described in equation (3.3) as,

\[
\text{rule 1: if } e \text{ is } NB \text{ and } \Delta e \text{ is } NB \text{ then } (K_p \text{ is } VVB)(K_i \text{ is } VVB)(K_d \text{ is } S) \\
\text{rule 25: if } e \text{ is } PB \text{ and } \Delta e \text{ is } PB \text{ then } (K_p \text{ is } S)(K_i \text{ is } S)(K_d \text{ is } VB)
\]

(3.3)

**Table 3.1 Rule base of fuzzy tuner**

<table>
<thead>
<tr>
<th></th>
<th>Rate of change of error ($\Delta e$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>Rate of change of error ($\Delta e$)</td>
</tr>
<tr>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>VVB</td>
</tr>
<tr>
<td>Z</td>
<td>VVB</td>
</tr>
<tr>
<td>PS</td>
<td>B</td>
</tr>
<tr>
<td>PB</td>
<td>M</td>
</tr>
</tbody>
</table>

| $K_i$          | Rate of change of error ($\Delta e$) |
| NB             | NB        | NS        | Z         | PS        | PB         |
| NS             | VVB       | VB        | VB        | B         | M          |
| Z              | B         | M         | S         | S         | S          |
| PS             | B         | B         | M         | S         | S          |
| PB             | M         | S         | S         | S         | S          |

| $K_d$          | Rate of change of error ($\Delta e$) |
| NB             | NB        | NS        | Z         | PS        | PB         |
| NS             | S         | S         | M         | M         | VB         |
| Z              | S         | S         | S         | S         | M          |
| PS             | M         | M         | M         | VB        | S          |
| PB             | S         | S         | S         | M         | VB         |

The overall rule base for fuzzy tuner is shown in Table 3.1. The fuzzy system utilizes the centroid defuzzification method. Centroid defuzzification provides the center of area under the curve. This is the most commonly used technique and it is very accurate. It is described by the equation (3.4) as,
The output of the fuzzy tuner is then multiplied with PID controller and it provides the supervised control signal \( U_F \) to the online RBFNN.

### 3.4 DEVELOPMENT OF ONLINE SUPERVISED RADIAL BASIS FUNCTION NEURAL NETWORK

In general, RBFNN combines the interpolation and approximation theory. The radial basis function neural network is similar to feed forward networks such as back propagation and multilayer perceptron. The radial basis function neural network aids in function approximation, classification, and modeling of dynamic systems.

![Architecture of four receptive field radial basis function neural network](image)

**Figure 3.5 Architecture of four receptive field radial basis function neural network**

In this section, development of online supervised radial basis function neural network is described. Figure 3.5 shows the architecture of four receptive field radial basis function neural network. The RBFNN consists of three layers with entirely different roles. The input layer is made up of nodes that connect the network to its environment. The second layer is the hidden layer of neurons. At the input of each neuron, the distance between the neuron center and the input vector is calculated. By applying the radial basis function (gaussian function) to this distance, the output of
the neuron is formed. The last layer is the output layer. It is linear and supplies the response of the network to the activation pattern.

The activation level of \( i^{th} \) receptive field unit (or hidden layer) is expressed in the equation (3.5) as,

\[
w_i = R_i(x) = R_i \left( \frac{\|x-u_i\|}{\sigma_i} \right), \quad i = 1, 2, \cdots, H1
\]  

(3.5)

Where, \( x \) is a multidimensional input vector, \( u_i \) is a vector with same dimension as \( x \), \( H1 \) is the number of radial basis functions or receptive field units, \( \sigma_i \) is spread width of the centre in \( i^{th} \) hidden layer, and \( R_i(x) \) is the \( i^{th} \) radial basis function with a single maximum at the origin. There are no connection weights between the input layer and the hidden layer. Typically, \( R_i(x) \) is a gaussian function and it is expressed in the equation (3.6) as,

\[
R_i(x) = \exp \left( -\frac{\|x-u_i\|^2}{2\sigma_i^2} \right)
\]  

(3.6)

Thus, the activation level of radial basis function \( w_i \) computed by the \( i^{th} \) hidden unit is maximum when the input vector \( x \) is at the centre \( u_i \) of that unit.

The output of an RBFNN can be computed by weighted average method. The final output is the weighted average of the value associated with each receptive field which is expressed in the equation (3.7) as,

\[
U_a(x) = \frac{\sum_{i=1}^{H} c_i w_i}{\sum_{i=1}^{H} w_i} = \frac{\sum_{i=1}^{H} c_i R_i(x)}{\sum_{i=1}^{H} R_i(x)}
\]  

(3.7)

Where, \( c_i \) is the weight of the output layer to the hidden layer. After the initialization of the initial centre (\( u_i \)) and spread of centre (\( \sigma_i \)) of the receptive field of the radial basis function neural networks, the whole architecture is adjusted through a further optimization procedure i.e., supervised learning using gradient descent and fuzzy PID supervised algorithms.
For developing the controller, the first step is to develop an objective function as given in the equation (3.8). The objective function is implemented using a gradient-descent procedure that represents a generalization of the least means squares algorithm. Least Mean Squares (LMS) algorithm is widely used to determine the transfer function of an unknown system. By using inputs and outputs of that system, the LMS algorithm is applied in an adaptive process based on the minimum mean squares error.

\[
E = \frac{1}{2} \sum_{p=1}^{N} (U_{Fp} - U_{Ap})^2
\]

(3.8)

Where, N is the number of training data, \(U_a\) and \(U_F\) are the actual and target output values respectively, if a network with differentiable active functions are considered, then the necessary condition for minimal error is that its derivatives with respect to the parameters centre \((u_i)\), spread width \((\sigma_i)\), and output weights \(c_i\) should vanish.

An iterative procedure for finding a solution to this problem is gradient descent. Here, the full parameter set \(U= (u_i, \sigma_i, c_i)\) is moved by a small distance \(\eta\) in the direction in which \(E\) decreases most rapidly, i.e. in the direction of the negative gradient \(-VE\) and it is expressed in the equation (3.9) as,

\[
U^{t+1} = U^t - \eta \nabla E(U^t)
\]

(3.9)

For the RBFNN, for the gaussian basis function, the adaption rules or the network parameters are expressed in the following equations (3.10) to (3.12) as,

\[
c_i(t + 1) = c_i(t) - \eta \frac{\partial E(t)}{\partial c_i(t)}
\]

(3.10)

\[
u_i(t + 1) = \nu_i(t) - \eta \frac{\partial E(t)}{\partial u_i(t)}
\]

(3.11)

\[
\sigma_i(t + 1) = \sigma_i(t) - \eta \frac{\partial E(t)}{\partial \sigma_i(t)}
\]

(3.12)

Where, \(t\) is the current iteration and \(\eta\) is the learning rate or step size of the learning algorithm.
3.5 SIMULATION RESULTS AND DISCUSSION

In order to validate the effectiveness of fuzzy PID supervised online RBFNN controller, simulink model is created with MATLAB/Simulink toolbox. Speed response is obtained for varying load conditions and varying set speed conditions of the Brushless DC motor with online RBFNN and conventional PID controller. Following performance parameter such as rise time, settling time, recovery time, peak overshoot, steady state error, integral of absolute error (IAE), and integral of time multiplied absolute error (ITAE) are measured and analyzed for above considered controllers. The specification for the Brushless DC motor are follows, nominal power-50 watts, rated current-2.5 amps, input voltage-28 V DC, rated speed-1500 rpm, rated torque- 0.38 Nm.

**Figure 3.6 Simulink model of fuzzy PID supervised algorithm**

Figure 3.6 shows a simulink model of fuzzy PID supervised algorithm and Figure 3.7 shows the simulink model of the fuzzy PID supervised online radial basis function neural network controller.
3.5.1 Speed response for varying load condition

In this section, simulated speed response of the Brushless DC motor for varying load conditions are described for two cases. For case 1, the load is varied from no load to full load at 0.2 Sec. For case 2, the load is varied from full load to no load at 0.4 Sec. Results of simulation of varying load condition with PID and fuzzy PID supervised online RBFNN controllers are shown in Figure 3.8. The control system performance parameters are measured and tabulated in Table 3.2.

From the simulation result of case1, it has been observed that the fuzzy PID supervised online RBFNN controller has reduced peak overshoot compared to the conventional PID controller. The steady state error with the fuzzy PID supervised online RBFNN controller has reduced drastically when compared with PID controller. The IAE and ITAE values of fuzzy PID supervised online RBFNN controller are lower than the PID controller with minimum oscillation. Smoothness of the control action with proposed controller is justified with minimum oscillation obtained. The
result also reveals that the disturbances are also compensated fairly well with the fuzzy PID supervised online RBFNN controller.

![Figure 3.8 Speed response of Brushless DC motor for varying load condition](image)

**Table 3.2 Control system performance parameters for varying load conditions**

<table>
<thead>
<tr>
<th>Controller</th>
<th>Rise time (Sec)</th>
<th>Peak overshoot (%)</th>
<th>Steady State error (rpm)</th>
<th>IAE x $10^2$</th>
<th>ITAE x $10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 1</td>
</tr>
<tr>
<td>PID</td>
<td>0.04</td>
<td>-</td>
<td>2.6</td>
<td>1.46</td>
<td>12</td>
</tr>
<tr>
<td>Fuzzy PID supervised online RBFNN</td>
<td>0.035</td>
<td>-</td>
<td>1.5</td>
<td>0.33</td>
<td>0.5</td>
</tr>
</tbody>
</table>

For case 2, the overshoot in the speed response of the motor has been reduced from 1.46 % using PID controller to 0.33 % using fuzzy PID supervised online RBFNN controller. The steady state error of the speed response of the motor also has been reduced to minimum using fuzzy PID supervised online RBFNN controller. In addition, integral criteria, IAE and ITAE are considered because they
respectively reflect the transient and steady state characteristics of the speed control system. The values of error integral criteria obtained for both cases are provided in Table 3.2 which indicates that the error in fuzzy PID supervised online RBFNN controller is minimum compared to PID controller which shows the superiority of the online RBFNN controller.

3.5.2 Speed response for varying set speed condition

In this section, effectiveness of fuzzy PID supervised online RBFNN controller is described with varying set speed conditions. Two cases are considered. In case 1, the set speed is varied from 1500 rpm to 1000 rpm at 0.2 Sec and in case 2, the set speed is varied from 1000 rpm to 1500 rpm at 0.4 Sec. Result of simulation under varying set speed conditions under PID and fuzzy PID supervised online RBFNN controller is shown in Figure 3.9.

![Figure 3.9 Speed response of Brushless DC motor for varying set speed condition](image.png)

The comparison of the fuzzy PID supervised online RBFNN controller according to the performance measures used is given in Table 3.3. As a results obtained for both cases, fuzzy PID supervised online RBFNN controller improves the
transient performance for set speed variations. Peak overshoot and the steady state error are decreased. In addition, IAE, and ITAE values show the performance improvement capacity of the fuzzy PID supervised online RBFNN controller. With the fuzzy PID supervised online RBFNN controller, the set speed tracking performance is improved with small steady error and less overshoot as it compared to the PID control structure. From these analyses, it is evident that, all vital performances of the system are in favour of the fuzzy PID supervised online RBFNN controller.

Table 3.3 Control system performance parameters for varying set speed conditions

<table>
<thead>
<tr>
<th>Controller</th>
<th>Recovery time (Sec)</th>
<th>Peak overshoot (%)</th>
<th>Steady State error (rpm)</th>
<th>IAE x 10^2</th>
<th>ITAE x 10^4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 1</td>
<td>Case 2</td>
<td>Case 1</td>
</tr>
<tr>
<td>PID</td>
<td>0.30</td>
<td>0.50</td>
<td>3.5</td>
<td>3.2</td>
<td>15</td>
</tr>
<tr>
<td>Fuzzy PID supervised online RBFNN</td>
<td>0.22</td>
<td>0.43</td>
<td>2.0</td>
<td>1.5</td>
<td>8</td>
</tr>
</tbody>
</table>

3.6 SUMMARY

Fuzzy PID supervised online RBFNN based speed controller for Brushless DC motor has been presented. The overall control system has been created and simulated using MATLAB/Simulink. Effectiveness of the fuzzy PID supervised online RBFNN controller is analyzed and compared with conventional PID controller. The performance of the motor is simulated for different operating conditions with the proposed controller to suit the real time environment. In order to present a reasonable comparison, several performance measures such as peak overshoot, steady state error, rise time, recovery time, integral absolute error, and integral time absolute error are used. The results obtained from the simulations clearly reveal the drastic improvement on performance with proposed controller. It is ascertained that, the fuzzy PID supervised online RBFNN controller perform very well under all operating conditions than the conventional PID controller.