CHAPTER 3
CONTROLLERS

This Chapter provides the basic fundamentals of various controllers and their operation and structure.

3.1 PID CONTROLLER

3.1.1 Introduction

Controller is a device which is inserted into the system to modify the error signal for better control action. PID Controller is one of the most widely used controllers for the purpose of controlling electrical drives. PID is made up of three main components:

- **P** – Proportional control
  The output varies based on how far we are from our target

- **I** – Integral control
  The output varies based on how long it's taking to get to target

- **D** – Derivative control
  The output varies based on the change in the error.

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs.
The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three term control: the proportional, the integral and derivative values, denoted P, I, and D. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.

3.1.2 Block Diagram

The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element shown in Figure 3.1.

![Block Diagram of PID Controller](image)

Figure 3.1 Block Diagram of PID Controller

P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. By
tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

### 3.1.3 PROPORTIONAL CONTROL

It has a varying output based on the error between current position and target position. It also has a “gain” value.

The formula for P is

\[ P_{OUT} = K_P * K_{ERR} \]  

(3.1)

The formula for \( K_{ERR} \) is

\[ K_{ERR} = \text{Target Point} - \text{Current Point} \]  

(3.2)

where,

- \( P_{OUT} \) is the result
- \( K_P \) is the gain
- \( K_{ERR} \) is the error

### 3.1.4 Integral Control

Integral (I) Control is similar to P control; however instead of the current error, it uses the integrated error. That is the sum of the error every cycle around. The longer it takes you to reach your target, the higher the integrated error becomes and the higher the output. This is most useful in completing a P based control. Using the above example of a turn, when the power level output by P control became low enough to stall, the integrated error starts to build up and keep the robot turning, as it approaches the target,
the P error continues to drop, and generally the I output will cause an 
overshoot and then drive it back. The formula for I control is:

\[ I_{OUT} = K_i \times I_{ERR} \]  \hspace{1cm} (3.3)

The formula for \( I_{ERR} \) is

\[ I_{ERR} = \text{Previous } I_{ERR} + K_{ERR} \]  \hspace{1cm} (3.4)

### 3.1.5 Derivative Control

Derivative control, D, sometimes called Delta control because it is 
actually driven by the change, or delta, of the \( K_{ERR} \). As such it can be used to 
react to sudden changes in error, and is good for maintaining a certain 
position or velocity on a closed loop system.

The formula for D is:

\[ D = D_{ERR} \times K_D \]  \hspace{1cm} (3.5)

The formula for \( D_{ERR} \) is:

\[ D_{ERR} = K_{ERR} - \text{Previous } K_{ERR} \]  \hspace{1cm} (3.6)

Derivative term slows the rate of change of the controller output 
and this effect is most noticeable close to the controller set point. Hence, 
derivative control is used to reduce the magnitude of the overshoot produced 
by the integral component and improve the combined controller-process 
stability.
3.2 FUZZY LOGIC

3.2.1 Introduction

Fuzzy logic is a logic having many values. The variables in fuzzy logic system may have any value in between 0 and 1 and hence this type of logic system is able to address the values of the variables those lies between completely truth and completely false. The variables are called linguistic variables and each linguistic variable is described by a membership function which has a certain degree of membership at a particular instance.

System based on fuzzy logic carries out the process of decision making by incorporation of human knowledge into the system. Fuzzy inference system is the major unit of a fuzzy logic system.

The decision making is an important part of the entire system. The fuzzy inference system formulates suitable rules and based on these rules the decisions are made. This whole process of decision making is mainly the combination of concepts of fuzzy set theory, fuzzy IF-THEN rules and fuzzy reasoning. The fuzzy inference system makes use of the IF-THEN statements and with the help of connectors present (such as OR and AND gate), necessary decision rules are constructed.

3.2.2 Fuzzy System

The basic Fuzzy inference system may take fuzzy inputs or crisp inputs depending upon the process and its outputs, in most of the cases, are fuzzy sets.
The fuzzy inference system in Figure 3.2 can be called as a pure fuzzy system due to the fact that it takes fuzzy sets as input and produces output that are fuzzy sets. The fuzzy rule base is the part responsible for storing all the rules of the system and hence it can also be called as the knowledge base of the fuzzy system. Fuzzy inference system is responsible for necessary decision making for producing a required output.

In general there are three main types of fuzzy inference systems such as:-

i) Mamdani model

ii) Sugeno model

iii) Tsukamoto model

There are also various defuzzification techniques such as:-

i) Mean of maximum method

ii) Centroid of area method

iii) Bisector of area method
In this work Mamdani fuzzification technique is used. There are two types of Mamdani fuzzy inference system such as, “min and max” and “product and max”. In this work the “min and max” Mamdani system is used. For this type of system, min and max operators are used for AND and OR methods respectively. Figure 3.3 explains the min and max Mamdani fuzzification technique. $\mu$ is the membership value for the linguistic variables $A_1, B_1, A_2, B_2, C_1, C_2$ and $C'$. 

Figure 3.3 Mamdani Fuzzy Inference system

The Fuzzy rules for the system are as follows:

1. If axis $A_1$ and $y$ belongs to $B_1$, then $z$ is $C_1$.
2. If axis $A_2$ and $y$ belongs to $B_2$, then $z$ is to $C_2$.

3.2.3 Fuzzy PID Controller

A fuzzy control system is a control system based on fuzzy logic and a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to
classical or digital logic, which operates on discrete values of either 1-true and 0- false. Fuzzy logic is widely used in machine control. The term itself inspires certain skepticism, sounding equivalent to "half-baked logic" or "bogus logic. Although genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

Fuzzy PID control method is a better method of controlling, to the complex and unclear model systems, it can give simple and effective control, Play fuzzy control robustness, good dynamic response, rising time, overstrike characteristics. Fuzzy Logic control (FLC) has proven effective for complex, non-linear and imprecisely defined processes for which standard model based control techniques are impractical or impossible.

3.2.4 Design of Fuzzy PID Controller

The Self-tuning fuzzy PID controller, which takes error "e" and rate of change-in-error "ec" as the input to the controller makes use of the fuzzy control rules to modify PID parameters on line. The self-tuning of the PID controller refers to finding the fuzzy relationship between the three parameters of PID, Kp, Ki, and Kd and "e" and "ec", and according to the principle of fuzzy control modifying the three parameters in order to meet different requirements for control parameters when "e" and "ec" are different and making the control object produce a good dynamic and static performance. Selecting the language variables of “e”, “ec”, KP, Ki and Kd is choosing even fuzzy values (NB,NM,NS,ZO,PS,PM,PB). The region of these variables, in this case, is taken to be {-3,-2,-1, 0, 1, 2, 3}. 
Here (NB, NM, NS, ZO, PS, PM, PB) is the set of linguistic values which respectively represent “negative big”, “negative medium”, “negative small”, “zero”, “positive small”, “positive medium” and “positive big”. The block diagram of a fuzzy self-tuning PID controller is shown in Figure 3.4.

![Fuzzy PID controller diagram](image)

**Figure 3.4 Fuzzy PID controller**

The set of linguistic rules is the essential part of a fuzzy controller. In many cases it’s easy to translate an expert’s experience into these rules and any number of such rules can be created to define the actions of the controller. In the designed fuzzy system, conventional fuzzy conditions and relations such as: “If e is A and ec is B, then Kp is C, Ki is D and Kd is E.” are used to create the fuzzy rule table which is shown in Table 3.1, 3.2 and 3.3.

**Table 3.1 Fuzzy Rules For \( \Delta Kp \)**

<table>
<thead>
<tr>
<th>e</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZO</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>PB</td>
<td>PB</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>NM</td>
<td>PB</td>
<td>PB</td>
<td>PM</td>
<td>PS</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
</tr>
<tr>
<td>NS</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZO</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
</tr>
<tr>
<td>PM</td>
<td>PS</td>
<td>ZO</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NB</td>
</tr>
<tr>
<td>PB</td>
<td>ZO</td>
<td>ZO</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NB</td>
</tr>
</tbody>
</table>
Table 3.2 Fuzzy Rules For ΔKi

<table>
<thead>
<tr>
<th>e≤ec</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZO</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>ZO</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>ZO</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>ZO</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>PS</td>
<td>NM</td>
<td>NS</td>
<td>ZO</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PM</td>
<td>ZO</td>
<td>ZO</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>ZO</td>
<td>ZO</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

Table 3.3 Fuzzy Rules For ΔKd

<table>
<thead>
<tr>
<th>e≤ec</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZO</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>PS</td>
<td>NS</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>PS</td>
</tr>
<tr>
<td>NM</td>
<td>PS</td>
<td>NS</td>
<td>NB</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>ZO</td>
</tr>
<tr>
<td>NS</td>
<td>ZO</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>ZO</td>
</tr>
<tr>
<td>ZO</td>
<td>ZO</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZO</td>
</tr>
<tr>
<td>PS</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>PM</td>
<td>PB</td>
<td>NS</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>PB</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
</tr>
</tbody>
</table>

3.3 NEURAL NETWORK

3.3.1 Introduction

The field of neural networks covers a very broad area. It would be impossible in a short time to discuss all types of neural networks. Instead, it is concentrated on the most common neural network architecture in the multilayer perceptron.
3.3.2 Neuron Model

The multilayer perceptron neural network is built up of simple components. A single input neuron, which will extend to multiple inputs. It will next stack these neurons together to produce layers. Finally, cascade the layers together to form the network.

3.3.3 Single-Input Neuron

The scalar input p is multiplied by the scalar weight w to form \( w_p \), one of the terms that is sent to the summer. The other input, 1 is multiplied by a bias b and then passed to the summer. The summer output n, often referred to as the net input, goes into a transfer function f, which produces the scalar neuron output a. Note that w and b are both adjustable scalar parameters of the neuron. Typically the transfer function is chosen by the designer and then the parameters w and b will be adjusted by some learning rule so that the neuron input/output relationship meets some specific goal.

3.3.4 Multiple-Input Neuron

Typically, a neuron has more than one input. The individual inputs \( p_1, p_2, \ldots, p_R \) are each weighted by corresponding elements \( w_{1-1}, w_{1-2}, \ldots, w_{1-R} \) of the weight matrix W. The first index indicates the particular neuron destination for that weight. The second index indicates the source of the signal fed to the neuron.

3.3.5 Neuro Fuzzy Controller

Consider a multi-input, single-output dynamic system whose states at any instant can be defined by “n” variables \( X_1, X_2, \ldots, X_n \). The control action that derives the system to a desired state can be described by a well known concept of “if-then” rules, where input variables are first transformed
into their respective linguistic variables, also called fuzzification. Then, conjunction of these rules, called inference process, determines the linguistic value for the output. This linguistic value of the output also called fuzzified output is then converted to a crisp value by using defuzzification scheme. All rules in this architecture are evaluated in parallel to generate the final output fuzzy set, which is then defuzzified to get the crisp output value.

**Constraint 1**: The fuzzification process uses the triangular membership function.

**Constraint 2**: The width of a fuzzy set extends to the peak value of each adjacent fuzzy set and vice versa. The sum of the membership values over the interval between two adjacent sets will be one. Therefore, the sum of all membership values over the universe of discourse at any instant for a control variable will always be equal to one. This constraint is commonly referred to as fuzzy partitioning.

**Constraint 3**: The defuzzification method used is the modified center of area method. This method is similar to obtaining a weighted average of all possible output values.