CHAPTER-3

CONVENTIONAL CONTROLLER

3.1 INTRODUCTION

Many industrial processes are controlled using conventional controllers like PI, PD, and PID etc. The PI controller is very popular because of their robust performance over a wide range of operating conditions and functional simplicity. The basic purpose of process control systems such as is two-fold: To manipulate the final control element in order to bring the process measurement to the set point whenever the set point is changed, and to hold the process measurement at the set point by manipulating the final control element. The control algorithm must be designed to quickly respond to changes in the set point (usually caused by operator action) and to changes in the loads (disturbances). The design of the control algorithm must also prevent the loop from becoming unstable, that is, from oscillating. There are varieties of control actions that are used, in order to achieve the desired response from the designed process satisfactorily and efficiently [45, 46].

The traditional and easiest approach to the controller design problem for non-linear systems involves linearising the modeling equations around a steady state and applying linear controller theory results. However, the controller performance will deteriorate as the process moves further away from the steady state around which it was
linearised. Therefore application of PI controller becomes more attractive for controlling tool wear.

### 3.2 PROPORTIONAL+ INTEGRAL CONTROLLER

It is well-known that, despite many sophisticated control theories and techniques that have been devised in the last decades, proportional+integral (PI) controllers are still the most adopted in practical cases. In fact, due to their simple structure, PI controllers are relatively easy to tune and their use is well understood by a great majority of industrial practitioners and automatic control designers. It is also well-known that PI controllers are particularly adequate for processes whose dynamics can be effectively modeled by a first or third order system.

**Proportional** - means a value varying relative to another value. The output of a proportional controller is relative to (or a function of) the difference between like temperature being controlled and the set point. Accordingly it is expressed as

\[
u(t) = K_p e(t) = K_p \{r(t) - y(t)\}
\]

In its pure form, the output of the controller is the error times the gain added to a constant known as “proportional gain”. 

Fig. 3.1 Block diagram for the proportional controller

Where the control signal is equal to $K_p e(t)$. The output (control signal) is equal to the error time the gain as shown in Fig. 3.2. One way to examine the response of a control algorithm is the open loop test. The evaluation is done by using an adjustable signal source as the process input and record the error (or process measurement) and the output [47, 48].

Fig. 3.2 Proportional Controller – error vs control signal

The proportional control produces an offset. Using the manual reset removes offset. The gain, however, cannot be made infinite. In most loops there is a limit to the amount of gain that can be used. If this limit is exceeded the loop will oscillate causing instability for the system.
The transfer function of a proportional controller can be derived trivially as:

\[ C(s) = k_c \]  \hspace{1cm} (3.2)

The main drawback of using a pure proportional controller is that it produces a steady-state offset (error). This motivates the addition of a bias (or reset) term \( u_b \) namely,

\[ u(t) = K_c e(t) + u_b \]  \hspace{1cm} (3.3)

The value of \( u_b \) can be fixed at a constant level \( (u_{\text{max}} + u_{\text{min}})/2 \) or can be adjusted manually until the steady-state error is reduced to zero. In commercial products the proportional gain is often replaced by the proportional band \( PB \), that is the range of error that causes a full range change of the control variable, \textit{i.e.,}

\[ PB = \frac{100}{K_c} \]  \hspace{1cm} (3.4)

\textbf{Acceleration} - is the derivative of velocity with respect to time.

\textbf{Integral} - is the opposite of a derivative. The integral of acceleration is velocity and the integral of velocity is distance.

The integral action is proportional to the integral of the control error, \textit{i.e.,} mathematically given by:

\[ u(t) = K_i \int_0^t e(\tau) d\tau, \]  \hspace{1cm} (3.5)
where $K_i$ is the integral gain. It appears that the integral action is causal of the control error. The $s$–domain transfer function of the integral action is given by:

$$C(s) = \frac{k}{s}$$  \hspace{1cm} (3.6)

The presence of a pole at the origin of the $s$–domain complex plane allows the reduction to zero of the steady-state error when a step reference signal is applied or a step load disturbance occurs. The correct value $u_b$ so that the steady-state error is zero through modifying the transfer function as:

$$C(s) = K_c \left( 1 + \frac{1}{sT_i} \right)$$  \hspace{1cm} (3.7)

For this reason the integral action is also often called automatic reset.

Thus, the use of a proportional action in conjunction to an integral action, i.e., performance a PI controller solves the main problems of the oscillatory response associated to a pure proportional controller. It must be kept in mind that when an integral action is present in shown Fig.3.3, the so-called integrator windup phenomenon might occur in the presence of saturation of the control variable [49].
By now, several schemes for adaptive tuning of PID controller have been proposed as reported in Astrom et al. (1988). Shahrokhi et al. (2000) compared four adaptive schemes for tuning of PID controller. The PI controller has been used in many industrial control systems for the past decades because of its simplicity and efficiency. Most commonly it is known as proportional+ Integral controller[50]. The form of this controller

\[ U(s) = K_c \left(1 + \frac{1}{T_i s}\right) e(s) \]  

(3.8)

Where, \(T_i\) is \((1/K_i)\) integral time constant or reset time.

Reset time is the time needed by the controller to repeat the initial proportional action change in its output. The integral action causes the controller output to change as long as error exists in the process output. Therefore, such a controller can eliminate even small errors. A PI controllers, thus needs special provisions to cope with the integral windup [51].

3.3 DESIGN OF PI CONTROLLER USING SYNTHESIS METHOD
Control system in which the output has an effect upon the input quantity in such a manner as to maintain the desired output value is called closed loop systems. The Flank wear model can be modified as closed loop system by providing a feedback. The provision of feedback automatically corrects the changes in output due to disturbance. Hence the closed loop system is also called automatic control system. The general block diagram of an automatic control system is shown in Fig. 3.4.

\[ \text{Input} \quad \rightarrow \quad \text{PI Controller} \quad \rightarrow \quad \text{Tool wear Sub-system} \quad \rightarrow \quad \text{Output} \]

*Fig.3.4 Block diagram of conventional controller*

It consists of an error detector, a controller, plant (Flank wear model) and feedback path elements [52].

The reference or input signal is nothing but cutting speed which corresponds to desired output. The feedback path element samples the output and converts it to a signal of same type as that of reference signal. The feedbacks signal is proportional to output signal. Here the output is flank wear which is to be controlled. The error signal generated by the error detector is the difference between reference signal and feedback signal. The controller modifies and amplifies the error signal to produce better control action. When the set-point is changed the
response of the process deviates and the controller tries to bring the output again close to the desired set point. In the present work a PI controller is implemented and controller parameters have an important effect on the response of the controlled process.

### 3.3.1 Control Loop Tuning

Once a Controlled process and associated control components are installed, this is referred to as control loop tuning. Before you set out to tune a control loop, you need to ensure the following [53].

- All sensor/transmitters must be calibrated to manufacturer’s specifications

- The end device must be properly sized for the process. This implies all parts of the system must be linearized to minimize variation in process gain over the normal operating range of the process.

For our purposes, the synthesis method is best, we will briefly outline by compare with the common trial-and-error method, Zeigler-Nichols continuous cycling method, Cohen-Coon method, these approaches to a process reaction method for PI control loop tuning methods, with following demerits [54].

- It is generally a time consuming method due to the number of trials necessary to achieve continuous cycling in each of the three modes.
If the process is noisy, it may be difficult to determine which part of the reaction curve has the steepest slope so as to construct your tangent line. In such cases, multiple trials may be necessary.

Continuous cycling is often objectionable because the process is pushed to the limits of its stability. As such, if any external disturbance enters the process while it is being tuned, the process could run away with a highly unstable response. Also, it is possible to damage equipment in some processes when operating at the stability limits.

If you do not take the time to ensure all elements of the process and control loop are properly configured, the time taken to tune the loop will likely be wasted.

Given the transfer function of the components of a feedback loop, synthesis the controller required to produce a specified closed-loop components other than the controller have been lumped into single block[69], \( G_p(s) \). From the Fig.3.2 the transfer function of the closed loop can be written as

![Fig. 3.5 Simplified Block Diagram of Conventional PI Controller](image)
\[
\frac{Y(s)}{Y_{sp}(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}
\]  

(3.9)

Substitute \(G_c(s) = 1\) and \(G_p(s) = \frac{K}{\tau s + 1}\)  

(3.10)

From this expression the controller transfer function is obtained as

\[
\frac{Y(s)}{Y_{sp}(s)} = \frac{k}{(\tau s + 1)}
\]

(3.11)

On simplification eqn (3.11), becomes

\[
\frac{Y(s)}{Y_{sp}(s)} = \frac{k}{(k+1)}
\]

(3.12)

\[
\frac{Y(s)}{Y_{sp}(s)} = \frac{K_p}{\tau c s + 1}
\]

(3.13)

This is a proportional + Integral (PI) Controller with tuning parameters

\[
k_c = \frac{\tau}{K_p \tau c}
\]

(3.14)

\[
T_i = \tau
\]

(3.15)

Where \(\tau_c\) is the time constant of the closed loop response and \(\tau\) is the process time constant. The PI controller settings are obtained by using synthesis method.

The smaller the value of \(\tau_c\) the faster will be the controller response so \(\tau_c\) provides convenient parameter to reach a compromise between fast
approach to set point and acceptable variation in the controller output that is in the manipulated variable. So $\tau_c$ is called performance adjustment parameter [55].

### 3.4 DEVELOPMENT OF PI CONTROLLER

The non-linear flank wear is modeled using mathematical method based on state-space analysis approach and the corresponding non-linear model details are obtained in chapter 2. The objective of modeling is to obtain the simplest mathematical description that adequacy predicts the response of the physical system to all anticipated inputs, there are different methods for tuning PI controller parameters $K_p$ and $T_i$ are designed based on a step change in cutting speed is given and the open loop response for tool wear is shown in Fig.3.6.

The optimal controller settings are then found after evaluation of the minimum values of ISE, IAE and percentage of peak overshoot, from table6.1 lists the PI controller settings for the chosen for non-linear flank wear model. The modeled of non-linear flank wear model of physical mathematical relationship developed. And PI control is developed based on tuning. Error in the output of flank wear is the surface roughness and cutting speed are respectively the output and input of the PI controller. Simulation studies are carried out on the mathematical model of the
non-linear flank wear model. Three major steps are considered in
performing the dynamic simulation of the process.

1. Developments of the mathematical model of the non-linear flank
wear.

2. Solution of the non-linear flank wear model and

3. Analysis of the simulation results.

There are different methods for tuning PI controller. A step change in
cutting speed is given and the open loop response for non-linear flank
wear is as shown in Fig.2.4.

The transfer function of the model is given by

\[ G(s) = \frac{K}{\tau s + 1} \]  \hspace{1cm} (3.16)

\[ \tau = 63.2\% \text{ of maximum value corresponding to time axis value} \]

K= \frac{\text{Change in output}}{\text{Change in input}}

Therefore, the transfer function of the model is given by

\[ G(s) = \frac{0.0041}{4.85s + 1} \]  \hspace{1cm} (3.17)

PI controller settings using synthesis method are given by

\[ K_c = \frac{\tau}{k\tau_c} \]  \hspace{1cm} (3.18)

\[ T_i = \tau \]  \hspace{1cm} (3.19)

The typical PI controller parameters are shown in Table 3.1, from the
results obtained using the synthesized method. Table 3.1 PI controller
parameters for Tool wear model
3.5 SIMULATION RESULTS

The dynamic non-linear flank wear models developed can be used for simulation. In the simulation of conventional controller, the cutting conditions were selected such that flank wear become dominant. The cutting speed was changed between 180-220 m/min and the simulation results obtained using PI controller is shown in Fig. 3.6 to 3.7.

<table>
<thead>
<tr>
<th>$K_c$</th>
<th>250.983</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_i$</td>
<td>51.856</td>
</tr>
</tbody>
</table>

Fig. 3.6(a) Response of Flank Wear under PI Controller { SP =0.06}

The Fig.3.6 (a) shows evaluation values are settling time = 300 Sec. , % of peack over shoot value= 34.16% and $IAE = 2.106$, $ISE =0.9211$ for the closed loop controller system response with the input value of Set Point = 0.06 of flank wear model.
The Fig.3.6(b) is shows the input value of the cutting speed varies in between 180-220 (m/min), which is used as input to the flank wear model of the plant in the closed loop controller system.

The Fig.3.7 (a) shows the evaluation values (i) settling time is 400 sec., (ii)% of over shoot = 20%, and (iii)ISE =0.12, (iv)IAE=5.78 for the closed loop control system response based on input reference as 0.1.
The Fig. 3.7 (b) shows the cutting speed value for the range 125 - 270 m/min instead of 180 – 220 m/min due to the input value of the flank wear of S.P = 0.1 because of the sudden variation from S.P = 0.06 however cutting speed is maintained as 180m/min after few seconds as shown in the same figure, which is used to give the input for the model of closed loop PI controller system.

The open loop response of non-linear flank wear model is obtained from the mathematical model of the flank wear equations (2.1) to (2.13) and (2.14) to (2.15), and the transfer function of the open loop response is derived as eqn (3.17) which indicates first order transfer function and it is used to design the PI controller by using the controller tuning methods to obtain the linear PI controller.

The figure 2.4 shows that the operating point varies linearly from 0 to 0.06 mm and non linearly from 0.06 to 0.12 mm , these non-linear
variation is converted to the linear process plant by using a PI controller, if this conversion fails then the non-linear will be converted into linear by using non-conventional controllers.

In this chapter theoretical aspects of conventional controller (PIC) along with simulation results and evaluation have been explained, however, to implement and develop high performance control system using PI controller when the plant's dynamic characteristics are poorly known and large unpredictable variations occur in the flank wear model. Therefore many difficulties exist in controlling time-varying and highly non-linear plan like Flank wear model. To overcome above difficulties, the new approach using ANN in control field has been introduced significantly in recent years. ANN is an interconnection of a no., of artificial neurons that simulates a biological neurons system. It has ability to approximate an arbitrary function mapping and can achieve the higher degree of fault tolerance. ANN has been successfully introduced to tune the PID control parameters like $k_c$, $k_i$, and $k_d$ values and also fuzzy controller techniques are suggested which are described in the following chapter.