CHAPTER 4

PID Controller Design

Introduction
Designing a PID control, or in fact any type of controller, must satisfy two requirements:

- Closed-loop stability
- Good performance

1. Stability Analysis

Stability is an important criterion of the overall system. The system is stable if all poles are negative, i.e. lie in the LHP in the complex domain.

In order for a feedback control loop to be stable, all of the roots of its characteristic equation must be either negative real numbers or complex numbers with negative real part.

![Stability Diagram]
Negative real root

Positive real root

Complex roots with negative real part

Complex roots with positive real part
1.0 Routh's Test
1.1 Direct substitution

Example 1.1: P-controller and first-order process

The characteristic equation is the denominator of the transfer function:

$$\tau s + 1 + k_c k_p = 0$$

Therefore, the closed-loop system is stable as long as $k_c > -1/k_p$

Since $k_c$ is always positive (or $k_c k_p > 0$), the stable region for $k_c$ is $[0, \infty)$

Example 1.2: PI-controller and first-order system

The characteristic equation is:

$$\tau I s^2 + (1 + k_p k_c) \tau I s + k_p k_c = 0$$

It has two roots:

$$p_1 = \frac{1}{2\tau I} \left( -(1 + k_c k_p) \tau I - \sqrt{(1 + k_c k_p)^2 \tau I^2 - 4\tau I k_c k_p} \right)$$

$$p_2 = \frac{1}{2\tau I} \left( -(1 + k_c k_p) \tau I + \sqrt{(1 + k_c k_p)^2 \tau I^2 - 4\tau I k_c k_p} \right)$$

Noting that the quantity under square root is always smaller than the term outside the square root, and since $\tau I > 0$ then the systems is stable as along as:

$$k_c > -1/k_p$$

Since $k_c$ is always positive (or $k_c k_p > 0$), the stable region for $k_c$ is $[0, \infty)$
Example 1.3: P-controller and second-order process

The characteristic equation is:

\[
1 + \frac{k_p}{\tau^2 s^2 + 2\tau\zeta s + 1} k_c = 0
\]

\[
\tau^2 s^2 + 2\tau\zeta s + 1 + k_c k_p = 0
\]

The system has two roots:

\[
p_1 = \frac{1}{2\tau^2} \left[ -2\tau\zeta + 2\sqrt{\tau^2\zeta^2 - \tau^2(1 + k_c k_p)} \right]
\]

\[
p_2 = \frac{1}{2\tau^2} \left[ -2\tau\zeta - 2\sqrt{\tau^2\zeta^2 - \tau^2(1 + k_c k_p)} \right]
\]

This means that the closed-loop system is stable for any value for \( k_c \) provided that \( k_c k_p > 0 \).

The above analysis indicated that the closed-loop system is stable for any value for \( k_c \) as long as the correct action is used (i.e. reverse or direct). However, this is only true for stable first-order and second order systems.

For unstable, or higher order, or time delay processes, there exist a range for stabilizing values for \( k_c \).

Example 1.4: P-controller with third order system

Consider the feedback system having the following elements:

\[
G_p = \frac{1}{5s + 1} \quad G_v = \frac{1}{2s + 1} \quad G_m = \frac{1}{s + 1}
\]

The overall transfer function for set point change is:

\[
\frac{y(s)}{r(s)} = \frac{G_c G_v G_p}{1 + G_c G_v G_p G_m}
\]
The characteristic equation is:

\[
1 + \frac{1}{5s + 1} + \frac{1}{2s + 1} + \frac{1}{s + 1} k_c = 0
\]

Therefore:

\[(5s + 1)(2s + 1)(s + 1) + k_c = 0\]

\[10s^3 + 17s^2 + 8s + 1 + k_c = 0\]

Using Routh test:

<table>
<thead>
<tr>
<th>(k_c)</th>
<th>(10)</th>
<th>(8)</th>
<th>(17)</th>
<th>(1 + k_c)</th>
<th>(7.4 - 0.59 k_c)</th>
<th>(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 + k_c)</td>
<td>(7.4 - 0.59 k_c)</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(7.4 - 0.59 k_c > 0 \Rightarrow 7.4/0.59 > k_c \Rightarrow k_c < 12.6\)

\(1 + k_c > 0 \Rightarrow k_c > -1\)

\(-1 < k_c < 12.6\)

Using direct substitution:

Set \(s = j\omega\)

\[10(j\omega)^3 + 17(j\omega)^2 + 8(j\omega) + 1 + k_c = 0\]

\[-10j \omega^3 - 17 \omega^2 + 8j\omega + 1 + k_c = 0\]

\[-10j \omega^3 + 8j\omega = 0 \Rightarrow \omega = 0, \ \omega = \pm 0.894\]

\[-17 \omega^2 + 1 + k_c = 0 \Rightarrow k_{cu} = 17 \omega^2 - 1\]

\[= 17(0)^2 - 1 = -1\]

\[= 17(0.894)^2 - 1 = 12.6\]

Chemical Engineering Department
King Saud University
2. Tuning of PID controllers

Controller tuning: selecting the best values for the adjustable parameters of a feedback controller

Adjustment of the PID settings should be performed to ensure some desired performance criteria:

1. Closed-loop system must be stable
2. Rapid, smooth response is obtained.
3. Offset is eliminated.
4. Specific overshoot, decay ratio, or rise time is obtained.
5. Excessive control action is avoided.
6. The control system is robust.

A number of alternative approaches for PID tuning are available:

1. Direct synthesis method
2. Internal model method
3. Minimum Error integral
4. Tuning relations based on quarter decay ratio
5. Frequency response techniques
6. Field tuning after installation

2.1 Tuning methods based on quarter decay ratio

2.1.1 Ultimate gain (continuous cycling)

Tuning Formula

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Proportional gain, $k_c$</th>
<th>Integral time, $\tau_i$</th>
<th>Derivative time, $\tau_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional only</td>
<td>P</td>
<td>$k_{cu}/2$</td>
<td>--</td>
</tr>
<tr>
<td>Proportional-integral</td>
<td>PI</td>
<td>$k_{cu}/2.2$</td>
<td>$T_u/1.2$</td>
</tr>
<tr>
<td>Proportional-integral-derivative</td>
<td>PID</td>
<td>$k_{cu}/1.7$</td>
<td>$T_u/2$</td>
</tr>
</tbody>
</table>
How to obtain ultimate gain:

(A) Experimental (watch out for saturation)
1. Switch the integral and derivative action off, i.e. set the integral time at maximum and derivative time to minimum.
2. With the controller in automatic, introduce small set point change and increase the controller gain till sustained oscillation is obtained.
3. From Figure (1) Record the ultimate gain and period.

(B) Modeling
1. Given the model transfer function and other loop elements, drive the overall transfer function using P-only controller.
2. Solve the characteristic equation using the direct substitution method described in Example 1.4 to determine the ultimate gain, $k_{cu}$ and ultimate frequency, $\omega$.
3. Calculate the period of oscillation, $T_u = \frac{2\pi}{\omega}$.

Advantages:
1. The experimental estimation of the ultimate gain does not require a process model, thus it is not affected by modeling error and is not limited to specific model type.
2. It provides considerable insight into the effect of all loop elements (process, instrumentation, etc).

Limitations:
Experimental
1. It is quite time-consuming and may be expensive because lost of productivity or poor product quality.
2. It may be objectionable since the process is pushed to the stability limit.

Experimental & modeled
3. It is not applicable to processes that are open-loop unstable.
4. Some simple processes do not have an ultimate gain such those which can be accurately modeled by first-order or second-order without time delay.

**Remarks on quarter decay ratio methods:**

1. Responses are often judged to be oscillatory with apparent overshoot for set point change.
2. The obtained tuning parameter values are not unique.
3. The tuning formula are empirical and should not be extrapolated beyond the range of \( \theta/\tau \) between 0.1 and 1.0.
4. The tuning formulas are based on FOPDT models.

**Autotuning (Relay feedback)**

Automatic tuning is an alternative to the continuous cycling. The method has the following advantages:

1. The system is forced to oscillate by a relay controller with a small amplitude.
2. Usually a single closed-loop experiment is sufficient.

The period of oscillation is found from the response plot and the ultimate gain is found by:

\[
k_{cu} = \frac{4h}{a\pi}\]

![Figure 2: Auto-tuning cycling](image-url)
### 2.1.2 Reaction curve (step testing)

#### Ziegler-Nichols Formula

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Proportional gain, $k_c$</th>
<th>Integral time, $\tau_I$</th>
<th>Derivative time, $\tau_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional only</td>
<td>$P$</td>
<td>$\frac{1}{k}\left(\frac{\tau}{\theta}\right)$</td>
<td>--</td>
</tr>
<tr>
<td>Proportional-integral</td>
<td>$PI$</td>
<td>$0.9\left(\frac{\tau}{\theta}\right)$</td>
<td>$3.33\theta$</td>
</tr>
<tr>
<td>Proportional-integral-derivative</td>
<td>$PID$</td>
<td>$1.2\left(\frac{\tau}{\theta}\right)$</td>
<td>$2.0\theta$</td>
</tr>
</tbody>
</table>

#### Coon and Cohen Formula

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Proportional gain, $k_c$</th>
<th>Integral time, $\tau_I$</th>
<th>Derivative time, $\tau_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$\frac{1}{k}\theta [1 + \theta / 3\tau]$</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>$PI$</td>
<td>$\frac{1}{k}\theta [0.9 + \theta / 12\tau]$</td>
<td>$\frac{\theta [30 + 3(\theta / \tau)]}{9 + 20(\theta / \tau)}$</td>
<td>--</td>
</tr>
<tr>
<td>$PID$</td>
<td>$\frac{1}{k}\theta [(16\tau + 3\theta) / 12\tau]$</td>
<td>$\frac{\theta [32 + 6(\theta / \tau)]}{13 + 8(\theta / \tau)}$</td>
<td>$\frac{40}{11 + 2(\theta / \tau)}$</td>
</tr>
</tbody>
</table>

How to obtain the process parameters:

(A) **Step testing**

1. With the plant is in open-loop mode, step the process input by a small amount ($\Delta u$) and record the output response till it reaches new steady state.
2. From the obtained response for the output determine the process parameters, $k_p$, $\tau$, and $\theta$.

(B) **Modeling**

If a theoretical model for the process can be developed, then the given model equations can be linearized around specific operating condition. Use the linearized model to develop the transfer function and consequently the process parameters.
Limitations:

**Experimental**
1. The step test is performed under open-loop conditions. Thus, the results may be distorted if significant disturbance occur.
2. It may be difficult to determine the slope at the inflection point because of noisy measurement.

**Experimental & modeled**
3. The method is not recommended for open-loop unstable processes or processes that have open-loop oscillatory response.

### 2.2 Minimum Error Integrals

**Tuning Formula**

**Load:**
- PI: $k_p k_c = A(\theta/\tau)^B$
- I: $\tau/\tau_1 = A(\theta/\tau)^B$
- D: $\tau_d/\tau = A(\theta/\tau)^B$

**Setpoint:**
- PI: $k_p k_c = A(\theta/\tau)^B$
- I: $\tau/\tau_1 = A(\theta/\tau) + B(\theta/\tau)$
- D: $\tau_d/\tau = A(\theta/\tau)^B$

<table>
<thead>
<tr>
<th>Type of input</th>
<th>Type of controller</th>
<th>Mode</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>PI</td>
<td>P</td>
<td>0.859</td>
<td>-0.977</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.674</td>
<td>-0.680</td>
</tr>
<tr>
<td>Load</td>
<td>PID</td>
<td>P</td>
<td>1.357</td>
<td>-0.947</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.842</td>
<td>-0.738</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D</td>
<td>0.381</td>
<td>0.995</td>
</tr>
<tr>
<td>Set point</td>
<td>PI</td>
<td>P</td>
<td>0.586</td>
<td>-0.916</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>1.03</td>
<td>-0.165</td>
</tr>
<tr>
<td>Set point</td>
<td>PID</td>
<td>P</td>
<td>0.965</td>
<td>-0.850</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.796</td>
<td>-0.146</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D</td>
<td>0.308</td>
<td>0.929</td>
</tr>
</tbody>
</table>
Types of integral errors

1. Integral of the absolute value of the error (IAE)
   *Good for small errors that occur around the steady state*

\[ \text{IAE} = \int_{0}^{\infty} |e(t)| dt \]

2. Integral of the squared value of the error (ISE)
   *Good for large errors that occur at the beginning of the response*

\[ \text{ISE} = \int_{0}^{\infty} e(t)^2 dt \]

3. Integral of the time-weighted absolute value of the error (ITAE)
   *Good for small errors that persist for long periods of time*

\[ \text{ITAE} = \int_{0}^{\infty} t|e(t)| dt \]

4. Integral of the time-weighted squared value of the error (ITSE)
   *Good for large errors that persist for long periods of time*

\[ \text{ITSE} = \int_{0}^{\infty} te(t)^2 dt \]
Remarks on minimum error integrals:
1. The obtained tuning parameter values are unique.
2. The criteria consider the performance of the whole response.
3. The tuning formulas are empirical and should not be extrapolated beyond the range of \((\theta/\tau)\) between 0.1 and 1.0.
4. The tuning formulas are based on FOPDT models.
5. The optimum tuning parameter values are different for servo and regulatory control objectives.
6. The optimum tuning parameters for disturbance is function of the type of disturbance.
7. The ITAE provide the most conservative tuning parameter values.
8. Tuning parameters obtained for disturbance rejection provide large overshoot when applied to set point change. While tuning parameters obtained for set point change may provide sluggish response when applied to disturbance rejection.

### 2.3 Direct Synthesis method

<table>
<thead>
<tr>
<th>Process Formula</th>
<th>mode</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_p/(\tau s+1))</td>
<td>PI</td>
<td>(k_c = \tau/k_p\tau_c) (\tau_1 = \tau)</td>
</tr>
<tr>
<td>(k_p/(\tau_1s+1)(\tau_2s+1))</td>
<td>PID</td>
<td>(k_c=(\tau_1+\tau_2)/k_p\tau_c) (\tau_1=(\tau_1+\tau_2)) (\tau_d=\tau_1\tau_2/(\tau_1+\tau_2))</td>
</tr>
<tr>
<td>(k_p e^{-\theta s}/(\tau s+1))</td>
<td>PI</td>
<td>(k_c = \tau/k_p(\tau_c+\theta)) (\tau_1 = \tau)</td>
</tr>
<tr>
<td></td>
<td>PID</td>
<td>(k_c = \tau/k_p(\tau_c+\theta)) (\tau_1 = \tau) (\tau_d=\theta/2)</td>
</tr>
</tbody>
</table>
**Design Procedure:**
The overall transfer function for set point change assuming $G_m = G_v = 1$:

$$G_{sp} = \frac{y}{r} = \frac{G_c G_p}{1 + G_c G_p}$$

Rearranging gives the design equation:

$$G_c = \frac{1}{G_p} \left[ \frac{(y/r)}{1 - (y/r)} \right] = \frac{1}{G_p} \left[ \frac{G_{sp}}{1 - G_{sp}} \right]$$

(1)

**Perfect control**

Consider the specification of perfect control, $G_{sp} = y/r = 1$

$$G_c = \frac{1}{G_p} \left[ \frac{1}{1} \right] = \frac{1}{G_p} 1$$

Therefore, perfect control is unachievable.

**Dahlin (first order) specification**

$$G_{sp} = \frac{1}{\tau_c s + 1}$$

$$G_c = \frac{1}{G_p} \left[ \frac{1}{\tau_c s + 1} \right] = \frac{1}{G_p} \frac{1}{\tau_c s}$$

(2)

$\tau_c$ is a tuning parameter that adjusts the speed of the response.

**Instantaneous process:** $G_p (s) = k_p$

$$G_c = \frac{1}{G_p} \frac{1}{\tau_c s} = \frac{1}{k_p \tau_c} \frac{1}{s}$$

This is a pure integral controller with $k_c = 1/k_p \tau_c$.
**First order process:**  \( G_p(s) = k_p/(\tau s + 1) \)

\[
G_c = \frac{1}{G_p} \frac{1}{\tau_c s} = \frac{\tau s + 1}{k_p \tau_c s} = \frac{\tau}{k_p \tau_c} (1 + \frac{1}{\tau s})
\]

*This is a PI controller with \( k_c = \tau/k_p \tau_c \) and \( \tau_I = \tau \)*

**Second order process:**  \( G_p(s) = k_p/(\tau_1 s + 1)(\tau_2 s + 1) \)

\[
G_c = \frac{1}{G_p} \frac{1}{\tau_c s} = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k_p \tau_c s} = \frac{\tau_1 + \tau_2}{k_p \tau_c} \left( 1 + \frac{1}{(\tau_1 + \tau_2) s} + \frac{\tau_1 \tau_2}{(\tau_1 + \tau_2) s} \right)
\]

*This is a PID controller with \( k_c = (\tau_1 + \tau_2)/k_p \tau_c \), \( \tau_I = (\tau_1 + \tau_2) \), \( \tau_d = \tau_1 \tau_2/(\tau_1 + \tau_2) \)*

**First order process with time delay:**  \( G_p(s) = k_p e^{-\theta s}/(\tau s + 1) \)

\[
G_c = \frac{1}{G_p} \frac{1}{\tau_c s} = \frac{\tau}{k_p \tau_c} \left( 1 + \frac{1}{\tau s} \right) e^{-\theta s}
\]

This controller is **unrealizable** because it contains knowledge of the future.

*Remedy:* Modify the desired specification to include time delay as follows:

\[
G_{sp} = \frac{e^{-\theta s}}{\tau_c s + 1}
\]

*Note:* This requires a prior **knowledge** of the dead time.

The design equation becomes:
Approximating the time delay either by first-order Taylor series or by long division gives:

\[ e^{-\theta s} \approx 1 - \theta s \]

Inserting the approximation of the time delay in the design equation gives:

\[
G_c = \frac{1}{G_p} \frac{e^{-\theta s}}{(\tau_c + \theta)s}
\]  

(3)

Then for FOPDT, we get:

\[
G_c = \frac{\tau s + 1}{k_p (\tau_c + \theta)s} = \frac{\tau}{k_p(\tau_c + \theta)} \left( 1 + \frac{1}{\tau s} \right)
\]

This is a PI controller with, \( k_c = \tau/k_p(\tau_c + \theta), \tau_l = \tau \)

Note: the design equation (3) can have different form depending on the type of approximation of the time delay. Thus, \( G_c \) can produce PID controller (See Smith and Corripio)

Note: The time delay sets an upper limit to the value of \( k_c \), e.g. \( k_c^{\text{max}} = \tau/k_p(\theta) \)

Remarks on the direct synthesis method:
1. It depends heavily on the model type.
2. It requires model inversion, which may cause problem for non-minimum phase processes.
3. The tuning parameters are function of \( \tau_c \).
4. The parameter \( \tau_c \) can be further optimized by IAE or 5% overshoot specifications.
• The parameter $\tau_c$ is advantageous because it allows engineer to achieve a specific response by adjusting on parameter.

• The parameter $\tau_c$ is dis-advantageous because the formula does not provide a ball-park.

### 2.4 Internal Model Control method

The IMC design is conceptually similar to the direct synthesis, except of two differences:

1. It explicitly takes into account model uncertainty.
2. It allows designer to trade-off control system performance against control robustness.

**Remarks:**
The IMC design is based on zero-pole cancellation, therefore:

1. It is not recommended for open-loop unstable processes.
2. The response to load disturbance may be poor if the canceled poles are slow.

**Note:** IMC is a control algorithm by itself, but it can be used to derive tuning formula for PID controllers.

**Tuning Formula:**

<table>
<thead>
<tr>
<th>Model</th>
<th>$k_c$</th>
<th>$\tau_1$</th>
<th>$\tau_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{k_p}{s}$</td>
<td>$1$</td>
<td>$\tau$</td>
<td>$\tau$</td>
</tr>
<tr>
<td>$\frac{k_p}{\tau_1}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$\tau_1$</td>
<td>$\tau_1$</td>
</tr>
<tr>
<td>$\frac{k_p}{\tau_2}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$\tau_2$</td>
<td>$\tau_2$</td>
</tr>
<tr>
<td>$\frac{k_p}{(\tau_1s + 1)(\tau_2s + 1)}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$\tau_1\tau_2$</td>
<td>$\tau_1\tau_2$</td>
</tr>
<tr>
<td>$\frac{k_p}{(\tau^2s^2 + 2\tau\zeta s + 1)}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$2\zeta\tau$</td>
<td>$2\zeta\tau$</td>
</tr>
<tr>
<td>$\frac{k_p}{s(\tau s + 1)}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$\tau$</td>
<td>$\tau$</td>
</tr>
<tr>
<td>$\frac{k_p}{\tau(c + \beta)}$</td>
<td>$\frac{k_p}{\tau}$</td>
<td>$2\zeta\tau$</td>
<td>$2\zeta\tau$</td>
</tr>
</tbody>
</table>

Chemical Engineering Department
King Saud University
Design Procedure:
The IMC design is based on the simplified block diagram:

![Figure 4: Classical Feedback Control](image1)

From Figure (4) and for set point change:

\[ y = \frac{G_c G}{1 + G_c G} r \]  \hspace{1cm} (4)

From Figure (5) and for set point change:

(Click on bookmark how1)

\[ y = \frac{G_c^* G}{1 + G_c^* (G - \bar{G})} r \]  \hspace{1cm} (5)

Comparing the last two equations yields:

(Click on bookmark how2)

\[ G_c = \frac{G_c^*}{1 - G_c^* \bar{G}} \]  \hspace{1cm} (6)
The design of $G^*_c$ is performed in three steps:

**Step 1:** The process model is factored as

$$\tilde{G} = \tilde{G}_+ \tilde{G}_-$$

Where $G_+$ contains any time delay and RHP zeros and it should have a unity steady state gain.

**Step 2:** The controller is specified as:

$$G^*_c = \frac{1}{G_-(\tau_c s + 1)^r} = \frac{1}{\tilde{G}_-} f$$

**Step 3:** convert $G^*_c$ into $G_c$ using equation (6).

- If the model is perfect, i.e. $\tilde{G} = G$, then equation (5) reduces to equation (4).
- $f$ is the IMC filter with tuning parameter $\tau_c$. Parameter $r$ is a positive integer that is selected so that $G^*_c$ is proper.

**Example: First order process:** $G_p(s) = k_p/(\tau s + 1)$

$$G_+ = 1 \quad G_- = \frac{k_p}{\tau s + 1}$$

$$G^*_c = \frac{1}{G_-} f$$

From Eq. (6):
\[ G_c = \frac{1}{G_- f} \left( 1 - f \frac{1}{G_-} \right) = \frac{1}{G_- f} \left( 1 - G_+ f \right) \]

Let \( f = \frac{1}{\tau_c s + 1} \); \( r = 1 \) is enough to make \( G_c^* \) proper.

\[ G_c = \frac{1}{G_-} f \left( 1 - \frac{1}{\tau_c s + 1} \right) = \frac{1}{G_-} f \frac{\tau_c s + 1}{\tau_c s} = \frac{1}{G_-} \frac{1}{\tau_c s + 1} \frac{\tau_c s + 1}{\tau_c s} \]

\[ G_c = \frac{\tau s + 1}{k_p \tau_c s} = \frac{\tau}{k_p \tau_c} \left( 1 + \frac{1}{\tau s} \right) \]

Therefore, comparing the last equation with that of a PI controller gives:

\[ k_c = \frac{\tau}{k_p \tau_c} \]

\[ \tau_f = \tau \]
2.5 Frequency response methods

2.5.1 Gain and phase margins
2.5.2 Pole placement

Remarks
- They are based on graphical representation of the closed-loop system in frequency-domain.
- Require knowledge of the model transfer function.
- Pole placement, complex models can lead to complex controllers.
3. Tuning of PID for nonlinear processes

Some Chemical processes have varying dynamics. The process gain and time constant vary with operating condition. Therefore, a PID controller with variable settings is recommended for such processes.

For example, the storage tank has the following parameters:

\[ k = R = \frac{2\sqrt{h_s}}{\beta}, \quad \tau = AR = 2A\frac{\sqrt{h_s}}{\beta} \]

For the heated storage tank:

\[ k = \frac{1}{F_s \rho C_p}, \quad \tau = \frac{V}{F_s} \]

For the CSTR with second order reaction:

\[ k = \frac{F}{F + 2VkC_{As}}, \quad \tau = \frac{V}{F + 2VkC_{As}} \]

The available tuning methods that tackle nonlinearities are:

- Gain scheduling
- Nonlinear PID
- Adaptive Self tuning
- Fuzzy logic tuning

3.1 Gain scheduling

Gain scheduling is mechanism to compensate for the changing dynamics of the process by adapting the process gain.

The overall gain is \( k_p k_c \), if we want to keep the overall gain constant at \( k_{co}k_{po} \), then observe that this would require to change \( k_c \) according to:

\[ k_c = \frac{k_{co}k_{po}}{k_p(t)} \]
The varying process gain can be determined from the process measurements as shown in the Figure.

![Gain Scheduling Control Scheme](image)

**Figure 6: Gain Scheduling Control Scheme**

Another common way of GS is to predetermine a set of controller tuning parameters for each region of operating space and then apply this schedule of appropriate controller parameters for each part of phase space. One of which can be given as an interpolation of two given extreme values for the controller gain

\[
k_c(t) = (1 - q) k_c^u + q k_c^l
\]

where \( k_c^u \) and \( k_c^l \) are the upper and lower values for the controller gain respectively and \( q \) is an interpolation parameter, which can be given as:

\[
q = \exp (-\kappa |e(t)|)
\]

where \( \kappa \) is a design parameter. In this case, the scheme is also known as fuzzy gain scheduling (FGS).

**Remarks**
1. Gain scheduling is comparable to feed-forward compensation. There is no feedback to compensate for incorrect adaptation.
3.2 Nonlinear PID

\[ k_c(t) = k_{co} \left( 1 + b|e(t)| \right) \]

where \( k_{co} \) is controller gain at zero error, \( e \) is the absolute value for the error and \( b \) is an adjustable parameter.

Remarks:
- Provide good set point response over broad range of operating levels with changing process gain.
- For large error the system may even be unstable, but instability is in the direction of driving the loop rapidly back to stable set point region.
- The low gain at the set point reduces the effect of noise.

3.3 Adaptive self tuning PID

The mechanism of the STC is shown in Figure 7.

![Block diagram for Self-Tuning Controller](image)

The PID parameters can be computed each sampling time from the following relations:

\[ k_c = \frac{(1 - q)(2a_2 + a_1)}{b_1} \]
\[ \tau_I = -T \frac{2a_2 + a_1}{1 + a_1 + a_2} \]

\[ \tau_d = -T \frac{a_2}{2a_2 + a_1} \]

Where \( q \) is an adjustable parameter, \( T \) is the sampling time and the rest are the parameters of the discrete model:

\[ y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) \]

The model represents a second-order system and its parameters are determined online by the recursive least squares method.

- The model can be modified to fit most types of processes and can include time delay.
- For first-order models, \( a_2 \) becomes zero and a PI control may be obtained.
- The model parameters can be estimated by straightforward multiple regression.
- However, the obtained controller is sensitive to the accuracy of the estimation method and to the adjustable parameters of the estimator.
4. Tuning of PID for unstable processes

See References
How to 1:

\[ e = R - (c - \tilde{c}) \]

\[ u = G^*_c e \]

\[ c = Gu = GG^*_c e \quad \Rightarrow \quad e = \frac{1}{GG^*_c} \]

\[ \tilde{c} = \tilde{G}u = \tilde{G}G^*_c e = \frac{\tilde{G}}{G} c \]

\[ \therefore \quad \frac{1}{GG^*_c} c = R - (c - \frac{\tilde{G}}{G} c) \]

\[ c = GG^*_c R - GG^*_c c + G^*_c \tilde{G} c \]

\[ c(1 + G^*_c (G - \tilde{G})) = GG^*_c R \]

\[ c = \frac{GG^*_c}{1 + G^*_c (G - \tilde{G})} R \]
How to 2:

Comparing Eq. (4) and Eq. (5):

\[
\frac{GG_c}{1 + GG_c} = \frac{GG_c^*}{1 + G_c^* (G - \tilde{G})}
\]

\[
G_c \left[1 + G_c^* (G - \tilde{G})\right] = G_c^* + GG_c G_c^*
\]

\[
G_c + GG_c G_c^* - \tilde{G} G_c G_c^* = G_c^* + GG_c G_c^*
\]

\[
G_c = G_c^* (1 + G_c \tilde{G}) \quad \Rightarrow \quad G_c^* = \frac{G_c}{1 + G_c \tilde{G}}
\]

\[
G_c (1 - G_c^* \tilde{G}) = G_c^* \quad \Rightarrow \quad G_c = \frac{G_c^*}{1 - G_c^* \tilde{G}}
\]