PID Controller Tuning

1. Introduction
2. Model-based PID tuning methods
3. On-line PID controller tuning
4. PID tuning guidelines and troubleshooting
5. Simulink example
Introduction

- **PID control law**
  
  \[ p(t) = \bar{p} + K_c \left[ e(t) + \frac{1}{\tau_I} \int_0^t e(t^*) dt^* + \tau_D \frac{de(t)}{dt} \right] \]

- **Transfer function**
  
  \[
  \frac{P'(s)}{E(s)} = G_c(s) = K_c \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right)
  \]

- **Controller tuning**
  
  » Need to select PID parameters \((K_c, \tau_I, \tau_D)\) that yield “good” closed-loop response to disturbances and setpoint changes
  
  » Only know: \(KK_c > 0, \tau_I > 0, \tau_D > 0\)
  
  » Trial-and-error tuning difficult and time consuming
  
  » Need methods to determine good initial parameter values
  
  » Refine parameter values by trial-and-error fine tuning
Impact of Controller Tuning

\( G_p(s) = G_d(s) = \frac{e^{-4s}}{20s + 1} \)
Closed-Loop Performance Criteria

1. Stability of closed-loop system
2. Minimization of disturbance effects
3. Rapid, smooth tracking of setpoint changes
4. Elimination of steady-state offset
5. Avoidance of excessive control action
6. Robustness to changes in process conditions
7. Robustness to errors in process model
Model-Based PID Controller Tuning

- Often based on first-order-plus-time-delay model

\[
\frac{Y(s)}{U(s)} = G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}
\]

- Integral error criteria

- Integral of absolute error
  \[
  IAE = \int_0^\infty |e(t)| \, dt
  \]

- Integral of squared error
  \[
  ISE = \int_0^\infty e^2(t) \, dt
  \]

- Integral of time weighted absolute error
  \[
  ITAE = \int_0^\infty t \cdot |e(t)| \, dt
  \]
## ITAE Tuning Rules

<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>Disturbance</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>Disturbance</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(^b)</td>
<td>−0.165(^b)</td>
</tr>
<tr>
<td>Set point</td>
<td>PID</td>
<td>P</td>
<td>0.965</td>
<td>−0.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I</td>
<td>0.796(^b)</td>
<td>−0.1465(^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D</td>
<td>0.308</td>
<td>0.929</td>
</tr>
</tbody>
</table>

\(^a\) Design relation: \(Y = A(\theta/\tau)^B\) where \(Y = KK_c\) for the proportional mode, \(\tau/\tau_I\) for the integral mode, and \(\tau_D/\tau\) for the derivative mode.

\(^b\) For set-point changes, the design relation for the integral mode is \(\tau/\tau_I = A + B(\theta/\tau)\).
General Trends

- The controller gain ($K_c$) is inversely proportional to the process gain ($K$).
- $K_c$ decreases as the ratio of the time delay ($\theta$) and the dominant process time ($\tau$) constant increases.
- Both the integral time ($\tau_I$) and the derivative time ($\tau_D$) increase as $\theta/\tau$ increases.
- A reasonable initial guess for the derivative mode is $\tau_D = 0.25\tau_I$.
- $K_c$ decreases as integral control is added to a proportional controller. $K_c$ increases as derivative control is added to a PI controller.
ITAE Controller Tuning Example

- **Process model**

\[
\frac{Y(s)}{U(s)} = G(s) = \frac{1.54e^{-1.07s}}{5.93s + 1}
\]

- **ITAE tuning for disturbance (more aggressive)**

\[
K_c = \frac{1}{K} A\left(\frac{\theta}{\tau}\right)^B = \frac{1}{1.54} \cdot 0.859 \left(\frac{1.07}{5.93}\right)^{-0.977} = 2.97
\]

\[
\tau_I = \frac{\tau}{A(\theta/\tau)^B} = \frac{5.93}{0.674(1.07/5.93)^{-0.680}} = 2.75
\]

- **ITAE tuning for setpoint (less aggressive)**

\[
K_c = \frac{1}{K} A\left(\frac{\theta}{\tau}\right)^B = 1.83
\]

\[
\tau_I = \frac{\tau}{A + B(\theta/\tau)} = 5.93
\]
On-Line PID Controller Tuning

- Often a process model is not available for controller tuning
- Need to determine tuning parameters directly from plant data
- Continuous cycling method
  - Utilizing a proportional controller, find the $K_c$ value that produces sustained oscillations. This is the ultimate gain $K_{cu}$.
  - Determine the period of the sustained oscillations. This is the ultimate period $P_u$.
  - $K_c$ and $P_u$ can also be found from an available transfer function model by direct substitution.
Tuning Rules Based on Ultimate Gain

<table>
<thead>
<tr>
<th>Ziegler-Nichols</th>
<th>$K_c$</th>
<th>$\tau_I$</th>
<th>$\tau_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$0.5K_{cu}$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PI</td>
<td>$0.45K_{cu}$</td>
<td>$P_u/1.2$</td>
<td>—</td>
</tr>
<tr>
<td>PID</td>
<td>$0.6K_{cu}$</td>
<td>$P_u/2$</td>
<td>$P_u/8$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tyreus-Luyben $\dagger$</th>
<th>$K_c$</th>
<th>$\tau_I$</th>
<th>$\tau_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>$0.31K_{cu}$</td>
<td>$2.2P_u$</td>
<td>—</td>
</tr>
<tr>
<td>PID</td>
<td>$0.45K_{cu}$</td>
<td>$2.2P_u$</td>
<td>$P_u/6.3$</td>
</tr>
</tbody>
</table>

$\dagger$ Luyben and Luyben (1997).
Shortcomings of Continuous Cycling Method

- Time consuming for processes with large time constants
- Process pushed to stability limit
- Ultimate gain does not exist for first- and second-order systems without time delays
- Generally not applicable to integrating and open-loop unstable systems because stability is achieved only for intermediate $K_c$ values
- First two shortcomings can be overcome using relay autotuning method to determine $K_c$ and $P_u$ (see text)
Step Test Method

- Model structure

\[ Y(s) = \frac{Ke^{-\theta s}}{\tau s + 1} U(s) \quad K = \frac{\Delta y}{\Delta u} \]
Step Test Method cont.

- Model structure
  \[ Y(s) = \frac{Ke^{-\theta s}}{\tau s + 1} U(s) \quad K = \frac{\Delta y}{\Delta u} \]

- Calculation of \( \theta \) and \( \tau \)
  » Determine times when output has reached 35.3\% (\( t_{35} \)) and 85.3\% (\( t_{85} \))
  » Calculate \( \theta \) and \( \tau \)

  \[ \theta = 1.3t_{35} - 0.29t_{85} \quad \tau = 0.67(t_{85} - t_{35}) \]

- Then use tuning method for FOTPD models
Shortcomings of Step Test Method

- Experimental test performed under open-loop conditions
- Method is not applicable to open-loop unstable processes
- Step response can be sensitive to the direction and magnitude of the step change if the process is nonlinear
Guidelines for Common Control Loops

- Flow rate control – aggressively tuned PI controller due to fast dynamics and few disturbances
- Liquid level control – conservatively tuned P or PI controller depending on whether offset-free performance is required
- Gas pressure control – moderately tuned PI controller with little integral action due to small time constants and stability concerns
- Temperature control – PID controllers with application dependent tuning
- Composition control – PID or more advanced controller due to sampling delay and measurement noise
Troubleshooting Control Loops

- Surveys indicate that as many as 35% of industrial controllers are placed in manual at any given time.

- Determine if control problem is attributable to:
  - Transducer – degradation or complete failure
  - Controller – poor tuning or inadequate design
  - Final control element – control valve stiction
  - Process – change in operating condition

- Troubleshooting procedure
  - Collect data on control system performance
  - Determine if operating conditions have changed
  - Check individual control system components
  - Retune or redesign the controller
Simulink Example: continuous_cycling.mdl

\[ G(s) = \frac{2e^{-s}}{(10s + 1)(5s + 1)} \]
Determining the Ultimate Gain

![Graph showing output vs time for different values of Kc]
PID Controller Tuning

- $K_c = 7.88$, $P_u = 11.66$
- Ziegler-Nichols tuning rules

$$K_c = 0.6K_{cu} = 4.73 \quad \tau_I = \frac{P_u}{2} = 5.83 \quad \tau_D = \frac{P_u}{8} = 1.45$$