Dynamics and PID control

Sigurd Skogestad

Process dynamics

• “Things take time”
• Step response: (response of output $y$ to step in input $u$):
  – $k = \Delta y(\infty)/\Delta u$ – process gain
  – $\tau$ - process time constant (63%)
  – $\theta$ - process time delay

• Time constant $\tau$: Often equal to residence time = $V[m^3]/q[m^3/s]$ (but not always!)

• Dynamic model: Can find $\tau$ (and $k$) from balance equations:

  • Rearrange to match standard form of 1st order linear differential equation:
Example dynamic model:
Concentration change in mixing tank

- Assume constant $V \text{ [m}^3\text{]}$
- Assume constant density $\rho \text{ [kg/m}^3\text{]}$
- Assume, $c$ (in tank) = $c$ (outflow) [mol A/m$^3$]
- Assume no reaction

<table>
<thead>
<tr>
<th>Inflow</th>
<th>Mass balance</th>
<th>Component balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho q_F$ [kg/s]</td>
<td>$c_F q_F$ [mol A/s]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outflow</th>
<th>Mass balance</th>
<th>Component balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho q$ [kg/s]</td>
<td>$c q$ [mol A/s]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inventory</th>
<th>Mass balance</th>
<th>Component balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho V$ [kg]</td>
<td>$c V$ [mol A]</td>
<td></td>
</tr>
</tbody>
</table>

Balances:

- Mass
- Component

\[
\frac{d(\rho V)}{dt} = \rho q_F - \rho q \quad \text{[kg/s]}, \quad \rho V \text{ constant } \Rightarrow q = q_F
\]

\[
\frac{d(cV)}{dt} = c_F q_F - cq \quad \text{[mol A/s]} \Rightarrow \frac{V}{q} \frac{dc}{dt} = -c + \frac{1}{k} c_F
\]

Response of linear first-order system

Standard form:\n\[
\frac{dy}{dt} = -y + ku, \quad y(0) = y_0.
\]

Make step in $u$ at $t = 0$

Solution: $y(t) = y_0 + \left(1 - e^{-t/\tau}\right) \frac{k\Delta y}{\Delta x(\infty)}$

Remember for first order response:
1. Starts increasing immediately (would reach new steady state after time $\tau$)
2. Reaches 63% of change after time $\tau$
3. Approaches new steady state exponentially (has for practical purposes reached new steady state after about 4 $\tau$)
Feedback control

Control systems elements:

- **Valve**
- **Hot water**
- **Thermocouple**
- **A/D (convert analog to digital signal)**
- **D/A (convert digital to analog signal)**
- **Computer (controller algorithm)**
- **Motor/Amplifier/Relay**
- **Card in computer**
- **Setpoint, etc.**

Block diagram of negative feedback control:

- $y$ = controlled variable (CV)
- $y_m$ = measured CV
- $y_s$ = setpoint (SP)
- $e = y_s - y_m = control\ error$
- $u = manipulated\ variable\ (MV)$

$C = Feedback\ Controller = ?$
Feedback controller

Controller
Algorithm: \( u = f(y_s - y_m) \)

Simplest controller algorithm: On/off controller.
Problem: cycles

Industry: Standard algorithm for SISO controllers: PID
Industry: Standard for multivariable control: MPC (model predictive control)

Mechanical thermostat = On/Off-control (cycles)
Electronic controller (thermostat) = P-control (should give small offset)
PID controller

• Proportional control (P)

\[ u = u_0 + K_c (y_s - y) \]

Input change is proportional to control error \( e \).

\( K_c \) = proportional gain (tuning parameter)
\( u_0 \) = «bias»

Problems proportional control:
1. Get steady-state offset (especially if \( K_c \) is small)

\[ \text{Offset (\%)} = \frac{1}{1 + K_c k} \cdot 100\% \]

\( k \): process gain
\( K_c \): controller gain

2. Oscillates if \( K_c \) is too large (can get instability)

P-control

Initially at steady-state (\( y = y_s = 0 \)). Change setpoint to \( y_s = 1 \) at \( t = 1 \).
• Fix: Add Integral action (I)
• Get PI-control:

\[ u(t) = u_0 + K_c e(t) + K_c \frac{\int_0^t e(t) dt}{\tau_I} \]

\( \tau_I \) = integral time (tuning parameter)
\( e = y_s - y \) (control error)

Note 1: Integral term will keep changing until \( e = 0 \) ⇒ No steady-state offset

Note 2: Small integral time gives more effect!
(\( \text{so set } \tau_I = 99999 \) (large!) to turn off integral action)

Note 3: Integral action is also called «reset action» since it «resets» the bias.
«Update bias \( u_0 \) at every \( \Delta t \):»

\[ u(t) = u_0(t) + K_c e(t) \]
where \( u_0(t) = u_0(t - \Delta t) + K_c \frac{\Delta t}{\tau_I} e(t) \)

Add also derivative action (D):
Get PID controller

\[ u(t) = u_0 + K_c [e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_I \frac{de(t)}{dt}] \]

• P-part: MV (\( \Delta u \)) proportional to error
  • This is usually the main part of the controller!

• I-part: Add contribution proportional to integrated error.
  • Integral keeps changing as long as \( e \neq 0 \)
  • \( \Rightarrow \) Will eventually make \( e = 0 \) (no steady-state offset!)

• Possible D-part: Add contribution proportional to change in (derivative of) error
  • Can improve control for high-order (S-shaped response) and unstable processes, but
  sensitive to measurement noise
Many alternative PID parameterizations

This course:
\[ u(t) = u_0 + K_c[e(t) + \frac{1}{\tau_I} \int_0^t e(t) \, dt + \tau_D \frac{de(t)}{dt}] \]

Alternative form:
\[ u(t) = u_0 + P e(t) + I \int_0^t e(t) \, dt + D \frac{de(t)}{dt} \]

Also other:
Proportional band = \( \frac{100}{K_c} \)
Reset rate = \( \frac{1}{\tau_I} \)
Etc…

NOTE: Always check the manual for your controller!

Digital implementation (practical in computer) of PID controller

Continuous (not possible in computer):
\[ u(t) = u_0 + K_c \left[ \int_0^t e(t) \, dt + K_I \frac{\int_0^t e(t) \, dt}{\tau_I} + K_D \frac{de(t)}{dt} \right] \]

where \( \tilde{u}(t) \) — bias term with integral action included

Introduce:
\[ \Delta t = \text{sampling time} \]
\[ k = \text{current value (at time } t \text{)} \]
\[ k - 1 = \text{previous value (at time } t - \Delta t \text{)} \]

Discrete (digital) approximations:
\[ \frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{\Delta t} \]
\[ \tilde{u}_k = \tilde{u}(t) \approx \tilde{u}_{k-1} + \frac{K_I e_k}{\tau_I} \Delta t \]

Conclusion: Digital PID implementation
\[ u_k = \tilde{u}_k + K_c e_k + K_D \frac{e_k - e_{k-1}}{\Delta t} \]
PID controller tuning

\[ u(t) = u_0 + K_c e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \frac{de(t)}{dt} \]

3 tuning parameters:
1. (Proportional) Controller Gain: \( K_c \)
2. Integral time: \( \tau_I \) [s]
3. Derivative time: \( \tau_D \) [s]

Want the system to be (TRADE-OFF!)
1. Fast initially (\( K_c \) large, \( \tau_D \) large)
2. Fast approach to steady state (\( \tau_I \) small)
3. Robust / stable (OPPOSITE: \( K_c \) small, \( \tau_I \) large)
4. Smooth use of inputs (OPPOSITE: \( K_c \) small, \( \tau_D \) small)

Tuning of your PID controller

I. “Trial & error” approach (online)

(a) P-part: Increase controller gain (\( K_c \)) until the process starts oscillating or the input saturates
(b) Decrease the gain (~ factor 2)
(c) I-part: Reduce the integral time (\( \tau_I \)) until the process starts oscillating
(d) Increase a bit (~ factor 2)
(e) Possible D-part: Increase \( \tau_D \) and see if there is any improvement

Very common approach,
BUT: Time consuming and does not give good tunings: NOT recommended
II. Model-based tuning (SIMC rule)

- From step response obtain
  - $k = \Delta y(\infty) / \Delta u$ – process gain
  - $\tau$ - process time constant (63%)
  - $\theta$ - process time delay

- Proposed SIMC controller tunings

$$K_c = \frac{1}{\tau_c + \theta}$$
$$\tau_f = \min(\tau, 4(\tau_c + \theta))$$

- Choose $\tau_c = \theta$ (delay) for "tight" control
- Choose $\tau_c > \theta$ for smoother control (but $K_c = \frac{\Delta y(\infty)}{\Delta u}$)

$\tau_0$: normally 0 (may try $\tau_0 = \tau_2$ 2nd order time constant (e.g. response time measurement), but should then get new $\tau_1$ and $\theta$ based on 2nd order response)
Example SIMC rule

• From step response
  – $k = \Delta y(\infty) / \Delta u = 10 \text{C} / 1 \text{kW} = 10$
  – $\tau = 0.4 \text{ min} \text{ (time constant)}$
  – $\theta = 0.3 \text{ min} \text{ (delay)}$

• Proposed controller tunings

  Select $\tau_c = \theta = 0.3 \text{ min} \text{ ("tight" control)}$:
  
  \[
  K_c = \frac{1}{k \tau_c + \theta} = \frac{1}{10 \cdot \frac{0.4}{0.3+0.3}} = 0.067
  \]
  
  \[
  \tau_I = \min \left( \frac{\tau}{0.4}, 4 \left( \frac{\tau_c + \theta}{0.3+0.3} \right) \right) = \min(0.4, 2.4) = 0.4 \text{ min}
  \]

Simulation PID control

• Setpoint change at $t=0$ and disturbance at $t=5 \text{ min}$
  1. Well tuned (SIMC): $K_c=0.07, \tau_{ai}=0.4 \text{ min}$
  2. Too long integral time ($K_c=0.07, \tau_{ai}=1 \text{ min}$): settles slowly
  3. Too large gain ($K_c=0.15, \tau_{ai}=0.4 \text{ min}$) – oscillates
  4. Too small integral time ($K_c=0.07, \tau_{ai}=0.2 \text{ min}$) – oscillates
  5. Even more aggressive ($K_c=0.12, \tau_{ai}=0.2 \text{ min}$) – unstable (not shown on figure)
1. **Delay** ($\theta$) is feedback control’s worst enemy!
   - Try to reduce it, if possible. Rule: “Pair close”!

2. **Common mistake:** Wrong sign of controller!
   - Controller gain ($K_c$) should be such that controller counteracts changes in output
   - Need negative sign around the loop ("negative feedback")
   - Two ways of achieving this:
     - (Most control courses:) Use a negative sign in the feedback loop. Then controller gain ($K_c$) should always have same sign as process gain ($k$)
     - (Many real control systems:) Always use $K_c$, positive and select between
       - “Reverse acting” when process gain ($k$) is positive
         - because MV ($u$) should go down when CV ($y$) goes up
       - “Direct acting” when $k$ is negative
   - WARNING: Be careful and read manual! Some reverse these definitions (wikipedia used to do it, but I corrected it)

3. **Integrating («slow») process:** If the response is not settling after approximately 10 times the delay (so $\tau/\theta$ is large), then you can stop the experiment and approximate the response as an integrating process (with only two parameters, $k'$ and $\theta$):

   ![Integrating process diagram]

   **Slope,** $k' = \frac{\Delta y}{\Delta t \cdot \Delta u}$

   **SIMC-settings (using $k' = k/\tau$):**
   
   \[
   K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta} \\
   \tau_I = 4 (\tau_c + \theta)
   \]
Example: Similar to shower process

\[ u = Q \]
\[ y = T \]
\[ d = T_F \]

Disturbance response with no control

\[ u = Q \]
\[ y = T \]
\[ d = T_F \]
**P-control**

\[ u = Q \]
\[ y = T \]
\[ d = T_F \]

And we find it to be \( K_c = 1.13 \)

\[ P \text{-control} \]

\[ K_c = 0.5; \quad \tau_{ai} = 9999; \]

Start simulation (press green button)

\[ \text{plot}(\text{time}, \text{u}, \text{time}, \text{T}, \text{time}, \text{T_f}), \quad \text{axis}([0 \ 800 \ -1.5 \ 1.5]) \]

**SIMC PI control**

\[ u = Q \]
\[ y = T \]
\[ d = T_F \]

No offset

\[ \text{SIMC PI control} \]

\[ K_c = 0.1; \quad \tau_{ai} = 20; \]

Start simulation (press green button)

\[ \text{plot}(\text{time}, \text{u}, \text{time}, \text{T}, \text{time}, \text{T_f}), \quad \text{axis}([0 \ 800 \ -1.5 \ 1.5]) \]
Recommend: $\tau_c =$ delay $\neq 100s$ because it is more robust and gives no overshoot in $u$

Measure also $T_d$: Cascade control is much better