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 = \( V[\text{m}^3]/q[\text{m}^3/\text{s}] \) (but not always!)
• Dynamic model: Can find \( \tau \) (and \( k \)) from balance equations:
  - Mass/energy [kg/s; J/s]: \( \frac{d}{dt} \) Inventory = Inflow - Outflow
  - Component [mol/s]: \( \frac{d}{dt} \) Inventory = Inflow - Outflow + Gen. by reaction
  - Rearrange to match standard form of 1st order linear differential equation: \( \tau \frac{dy}{dt} = -y + ku \)
Example dynamic model:
Concentration change in mixing tank

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

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

Balances:

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

Component:
\[
\frac{d(cV)}{dt} = c_F q_F - c q \quad [\text{mol A/s}] \Rightarrow V \left( \frac{d}{dt} \frac{c}{q} \right) = -c + \frac{1}{\tau} \cdot c_F
\]

Response of linear first-order system

Standard form*:
\[
\tau \frac{dy}{dt} = -y + ku
\]
Initially at rest (steady state): $y(0) = y_0$
Make step in $u$ at $t = 0$: $\Delta u$

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

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

* A more general transfer function for this system is the state-space form

\[
\begin{bmatrix}
\dot{x}
\end{bmatrix} =
\begin{bmatrix}
A & B
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix} +
\begin{bmatrix}
0
\end{bmatrix} u
\]
\[
\begin{bmatrix}
y
\end{bmatrix} =
\begin{bmatrix}
C
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix} +
\begin{bmatrix}
D
\end{bmatrix} u
\]

For case $A = -1/\tau$, $B = 1/\tau$, $C = 0$, $D = 0$.
Feedback control

Control systems elements:

Block diagram

Lines are signals ("information"):  
\( y \) = controlled variable (CV)  
\( y_m \) = measured CV  
\( y_s \) = setpoint (SP)  
\( e = y_s - y_m \) = control error  
\( u \) = manipulated variable (MV)  

\[ C = \text{Feedback Controller} = ? \]
Feedback controller

![Diagram of feedback 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)

---

**Hjelp til å velge varmeprodukter**

Vårt kjære lille land har i dag ett av de mest sommerfylte sommeren i mange år. Det er viktig å velge riktig oppvarming og termoteregulering.

**Ideell innenstemperatur:**


**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* \((u-u_0)\) 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 of typical process

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 \( e=0 \) \( \Rightarrow \) No steady-state offset

Note 2: Small integral time gives more effect!
(\( so \) 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 \( \Delta t \):»

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

Add also derivative action (D):
Get PID controller

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

• 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:
\[ u(t) = u_0 + P e(t) + I \int_0^t e(t)\,dt + D \frac{de(t)}{dt} \]
\[ P = K_c, \quad I = K_c/\tau_I, \quad D = K_c\tau_D \]

Also other:
- Proportional band = 100/K_p
- Reset rate = 1/τ_i

NOTE: Always check the manual for your controller!

Digital implementation (practical in computer) of PID controller

Continuous (not possible in computer):
\[ u(t) = u_0 + K_c \int_0^t \frac{\tau_I}{\tau} c(t)\,dt + K_c/\tau_D \frac{dc(t)}{dt} \]
\[ \tilde{u}(t) \]

where \( \tilde{u}(t) \) is the term with integral action included

Introduce:
- \( \Delta t \): sampling time
- \( k \): current value (at time \( t \))
- \( k-1 \): previous value (at time \( t - \Delta t \))

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

Conclusion: Digital PID implementation
\[ u_k = \tilde{u}_k + K_c e_k + K_c/\tau_D \frac{e_k - e_{k-1}}{\Delta t} \]
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
  - $k = \frac{\Delta y(\infty)}{\Delta u}$ – process gain
  - $\tau$ - process time constant (63%)
  - $\theta$ - process time delay

• Proposed SIMC controller tunings
  
  $K_C = \frac{1}{K} \frac{\tau}{\theta}$

  $\tau_f = \min(\tau, 4(\tau + \theta))$
  
  $\tau_f$ = desired response time with control (tuning parameter!).
  - Choose $\tau_c = \theta$ (delay) for “tight” control
  - Choose $\tau_c > \theta$ for smoother control (but $K_C \geq \frac{\Delta y_{max}}{\Delta y_{max}}$)

  $\tau_D$: normally 0 (may try $\tau_D = \tau_2 = 2$nd 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
  - \( k = \frac{\Delta y(\infty)}{\Delta u} = 10C / 1 \text{kW} = 10 \)
  - \( \tau = 0.4 \text{ min (time constant)} \)
  - \( \theta = 0.3 \text{ min (delay)} \)

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

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)
Comments tuning

1. Delay (θ) 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 \( \theta \)):

   \[
   k' = \frac{\Delta y}{\Delta t \cdot \Delta u}
   \]

   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

Simulink model: tunepid1_ex1

Note: level control not explicitly included in simulation (assume constant level)

Disturbance response with no control

Kc=0; tau_i=9999; % no control
Start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
**P-control**

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

\[K_c = 0.5; \tau_{ai} = 9999;\]  P-control

Start simulation (press green button)

\[plot(time, u, time, T, time, T_F), axis([0 800 -1.5 1.5])\]

**SIMC PI control**

1) SIMC PI tuning rule with \(\tau_i = 100\).

\[K_c = \frac{1}{\tau_i}; \tau_i = \frac{1}{\tau_i + \theta} = 20/200 = 0.1; \tau_i = \min(\tau_i, d/\tau_i + \theta) = 20\]

\[K_c = 0.1; \tau_{ai} = 20;\]  SIMC PI-control

Start simulation (press green button)

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

Measure also $T_c$: Cascade control is much better.

$K_c=0.119; taui=25; \%$ outer loop with $tauc=105$

$K_c2=0.1; taui2=1; \%$ inner loop with $tauc2=10$

Inner loop ($T_0$): tauc=10

Outer loop ($T$): tauc=105

sim('tunepid1_ex1_cascade') %start simulation
The experimental setup

This is the «Whistler»

\[ y = T \] (at top)
\[ u = Q \] (at bottom)

First we did a step-response experiment where \( u \) was increased from 0 to 1 (manual control). The temperature \( y = T \) increased from 20°C to 54°C (new steady state). This gives \( k = 68 \). The dynamics are quite slow because it takes time to heat up the glass. \( \theta = 5s, \tau = 120s \)

From this we obtained the model parameters and SIMC tunings (with \( \tau_c = \theta = 5s \))

We then put it into automatic and increased the setpoint to 70°C. The input (\( u = Q \)) increased immediately to max=1, and we should then have stopped the integration («ant windup») but we had forgotten to do this and this is why you can see that \( u = Q \) stayed at max=1 even after \( y = T \) has passed the setpoint... Not so good... but eventually we see that it was working well.

This can be confirmed by Ida who was the ONLY student who stayed behind to check how things went. Thanks, Ida!

The model. Step response: \( k = 68, \theta = 5s, \tau = 120s \)
The controller. SIMC (with \( \tau_c = \theta = 5s \)): \( K_c = 0.2, \tau_f = 40s \)
The closed-loop response

Ja, reguleringen virket etter hvert! - noe Ida kan bekrefte

\[ y = T \]
\[ y_s = T_a = 70 \text{C} \]

(The input is a bit noisy because of a noisy temperature measurement, but it works!)