PID Control

Karl Johan Åström
Tore Hägglund

Department of Automatic Control, Lund University

September 23, 2020

PID Control

Theme: The most common controller.

Introduction

- PID control is widely used in all areas where control is applied
  Solves almost all control problems
  Often combined with other PID, feedforward, and nonlinear elements
- A PID controller is more than meets the eye
- The autotuning adventure (Tore+KJ)
  Telemetric, Eurotherm 1979
  Adaptive control and auto-tuning
  STU, patents, NAF (Sune Larsson) SDM20
  Satt Control, Alfa Laval Automation, ABB
  Fisher Control, Emerson 1979–
  Research and the PID books 1988, 1995, 2006, ?
  Interactive Learning Modules Guzman, Dormido http://aer.ual.es/ilm/
- Technology transitions
  Pneumatic, mechanical,electric, electronic, computer
- Modeling: the FOTD model $P(s) = \frac{K}{1+sT}e^{-sL}$

The Magic of Feedback

Feedback has some amazing properties, it can

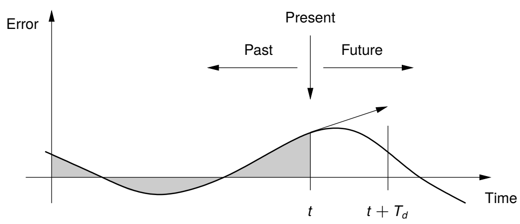
- make good systems from bad components,
- make a system insensitive to disturbances and component variations,
- stabilize an unstable system,
- create desired behavior, for example linear behavior from nonlinear components.

The major drawbacks are that

- feedback can cause instabilities
- sensor noise is fed into the system

PID control is a simple way to enjoy the Magic!

PID versus More Advanced Controllers



$$u(t) = k_p e + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt}, \qquad T_d = k_d / k_p$$

- PI does not predict
- PID predicts by linear extrapolation
- The derivative time T_d is the prediction horizon
- Advanced controllers predict using a mathematical model

The Amazing Property of Integral Action

Consider a PI controller

$$u = ke + k_i \int_0^t e(\tau)d\tau$$

Assume that all signals converge to constant values $e(t) \to e_0$, $u(t) \to u_0$ and that $\int_0^t (e(\tau) - e_0)d\tau$ converges, then e_0 must be zero.
Proof: Assume $e_0 \neq 0$, then

$$u(t) = ke_0 + k_i \int_0^t e(\tau)d\tau = ke_0 + k_i \int_0^t \big(e(\tau) - e_0\big)d\tau + k_i e_0 t$$

The left hand side converges to a constant and the left hand side does not converge to a constant unless $e_0 = 0$, futhermore

$$u(\infty) = k_i \int_0^\infty \big(e(\tau) - e_0\big)d\tau$$

A controller with integral action will always give the correct steady state provided that a steady state exists. It *adapts* to changing disturbances. Integral action is sometimes even called *adaptive*.

Entech Experience & Protuner Experiences

Bill Bialkowsk Entech - Canadian consulting company for pulp and paper industry *Average paper mill has 3000-5000 loops, 97% use PI the remaining 3% are PID, MPC, adaptive etc.*

- 50% works well, 25% ineffective, 25% dysfunctional

Major reasons why they don't work well

- Poor system design 20%
- Problems with valve, positioners, actuators 30%
- Bad tuning 30%

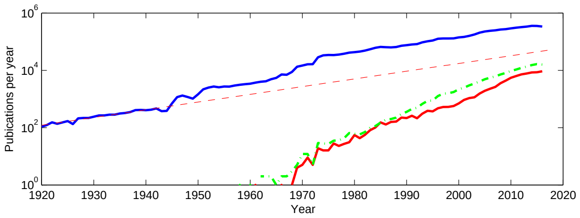
David Ender Techmation Control Engineering 1993 *Process Performance is not as good as you think.*

- More than 30% of installed controllers operate in manual
- More than 30% of the loops increase short term variability
- About 25% of the loops use default settings
- About 30% of the loops have equipment problems

Predictions about PID Control

- 1982: The ASEA Novatune Team 1982 (Novatune is a useful general digital control law with adaptation):
  PID Control will soon be obsolete
- 1989: Conference on Model Predictive Control:
  Using a PI controller is like driving a car only looking at the rear view mirror: It will soon be replaced by Model Predictive Control.
- 2002: Desborough and Miller (Honeywell):
  Based on a survey of over 11 000 controllers in the refining, chemicals and pulp and paper industries, 98% of regulatory controllers utilise PID feedback. The importance of PID controllers has not decreased with the adoption of advanced control, because advanced controllers act by changing the setpoints of PID controllers in a lower regulatory layer. The performance of the system depends critically on the behavior of the PID controllers
- 2016: Sun Li
  A recent investigation of 100 boiler-turbine units in the Guangdong Province in China showed 94.4% PI, 3.7% PID and 1.9% advanced controllers
- Similar studies in Japan and Germany

Publications in Scopus

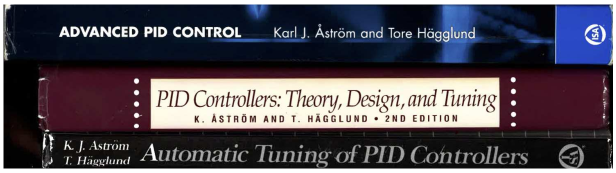


Number of publications by year for control (blue), PID (red) and model predictive control (green) from Scopus search for the words in title, abstract and keywords.

Tore – 40 Years of Collaboration

- Phd student 1978, PhD 1983; New Estimation Techniques for Adaptive Control
- Relay auto-tuning - patent 1983
- NAF 1985-89 - development of autotuners
- Back to the department at LTH 1989
- Three books

2007 Raymond D Molloy Award. Best selling book at ISA



Recent Student Project

- Kristian Soltesz 2013 On automation in Anesthesia
- Fredrik Bagge Carlson Projects: Optimization Julia programming
- Vanessa Romero PhD 2014 CPU Resource Management and Noise Filtering for PID Control
- Olof Garpinger PhD 2015 Analysis and Design of Software-Based Optimal PID Controllers
- Martin Hast PhD 2015 Design of Low-Order Controllers using Optimization Techniques
- Josefin Berner PhD 2017 Automatic Controller Tuning using Relay-based Model Identification
- Jonas Hansson and Magnus Svensson MS 2020 Next Generation Relay Autotuners Analysis and Implementation at ABB

PID Control

1. Introduction
2. The Controller
3. Stability
4. Performance and Robustness
5. Empirical Tuning Rules
6. Tuning based on Optimization
7. Relay Auto-tuning
8. Limitations of PID Control
9. Summary

Theme: The most common controller.

A PID Algorithm

A PID controller is much more than

$$u(t) = k_p e(t) + k_i \int_{t_0}^{t} e(\tau)d\tau + k_d \frac{de(t)}{dt}$$

We have to consider

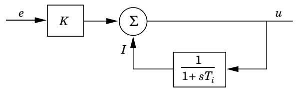
- Filtering
- Set point weigthing
- Actuator limitations
- Rate limitations
- Integrator Windup
- Mode switches
- Bumpless parameter changes
- Computer implementation

Dealing with these issues is a good introduction to practical aspects of any control algorithm.

Derivative and Integral Action from First Order Lag

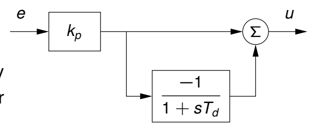
Integral action or automatic reset by positive feedback around a first order systems. We have

$$U = K\left(1 + \frac{1}{sT_i}\right)E$$



a PI controller!

Physical interpretation!!



Derivative action can be obtained by a parallel connection with a first order system. We have

$$U = k_p\left(1 - \frac{1}{1+sT_d}\right) = k_p \frac{sT_d}{1+sT_d} E$$

Is this how the body does it?

Filtering

Filter only derivative part

$$C_{fb}(s) = k\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1+sT_f}\right) = k_p + \frac{k_i}{s} + \frac{k_d s}{1+sT_f}$$

Filter the measured signal (several advantages)

- Better noise attenuation and robustness due to high frequency roll-off
- Process dynamics can be augmented by filter and design can be made for an ideal PID

$$C_{fb}(s) = \frac{k_d s^2 + k_p s + k_i}{s(1+sT_f)} = k_i \frac{1 + sT_i + s^2 T_i T_d}{s(1+sT_f)}$$

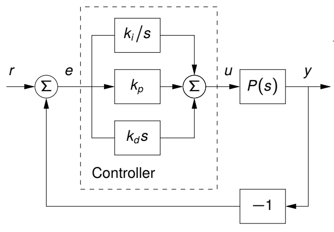
$$C_{fb}(s) = \frac{k_d s^2 + k_p s + k_i}{s(1+sT_f + s^2T_f^2/2)} = k_i \frac{1 + sT_i + s^2 T_i T_d}{s(1+sT_f + s^2T_f^2/2)}$$

High frequency rolloff improves robustness and noise sensitivity

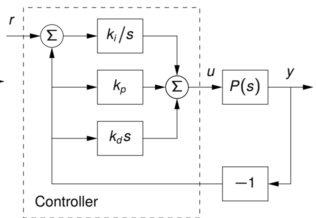
2DOF in PID Controllers

A 2DOF structure makes set-point response independent of disturbance response. Set-point weighting "Poor man's" 2DOF, allows a moderate adjustment of set point response through parameters b and c. Comment on practical controllers.

$$U(s) = k_p\big(bR(s) - Y(s)\big) + \frac{k_i}{s}\big(R(s) - Y(s)\big) + k_d s\big(cR(s) - Y(s)\big)$$



$b = 1 = 1$

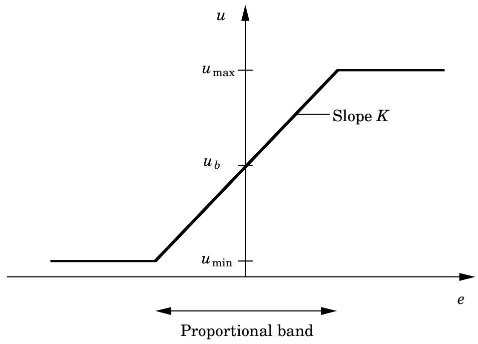


$b = c = 0$

The Proportional Controller - Proportional Band

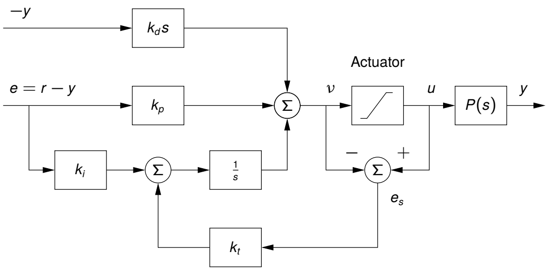
$$u = Ke + u_b, \qquad K \text{ gain}, \qquad u_b \text{ bias or reset}$$

The proportional band PB is the range where the output does not saturate, often given as percentage of error or measured signal.



Avoiding Windup

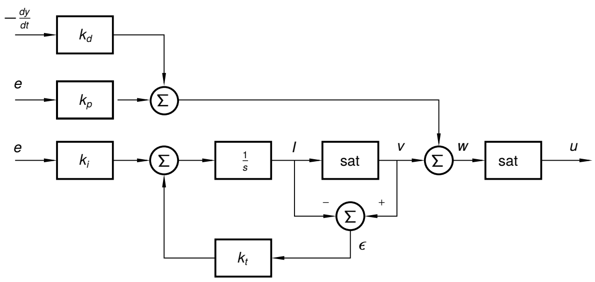
Feedback is broken when the actuator saturates?



A local feedback loop keeps integrator output close to the actuator limits. The gain k_t or the time constant $T_t = 1/k_t$ determines how quickly the integrator is reset. Intuitive Explanation - Cherchez l'erreur! Useful to replace k_t by a general transfer function.

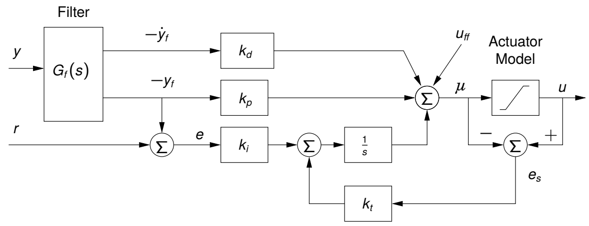
Dow Chemical Version of Anti-windup

Many process industries (also in Sweden) had their own control departments and they developed their own systems based on standard computers. Dow, Monsanto and Billerud were good examples.



The integrator is reset based on its output and not based on the nominal control signal as in previous scheme.

Dedicated Controller with Filtering and Antiwindup

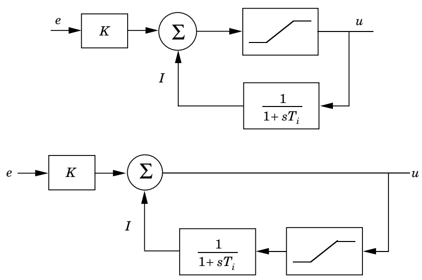


The filter (can be combined with antialias filter)

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -T_f^{-2} & -T_f^{-1} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ T_f^{-2} \end{bmatrix} y,$$

has the states $x_1 = y_f$ and $x_2 = dy_f/dt$. The filter thus gives filtered versions of the measured signal and its derivative. The second-order filter also provides good high-frequency roll-off.

Anti-windup in Series Implementation

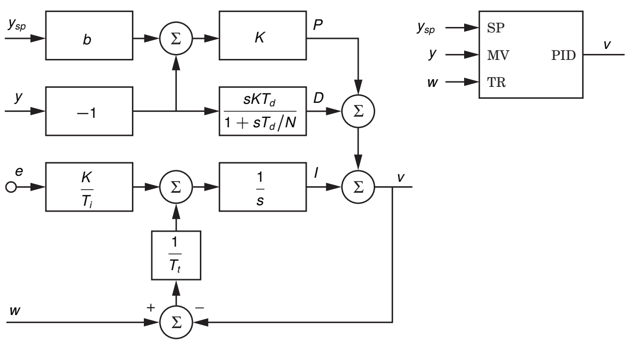


- These schemes are natural for pneumatic controllers
- Have been used by Foxboro (Invensys) for a long time
- Tracking time constant $T_t = T_i$

Manual and Automatic Control

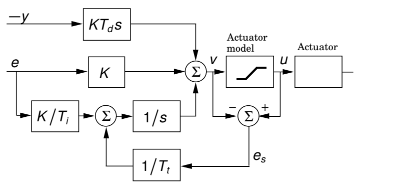
- Most controllers have several modes
  Manual/automatic
- In manual control the controllers output is adjusted manually by an operator often by increase/decrease buttons
- Mode switching is an important issue
- Switching transients should be avoided
- Easy to do if the same integrator is used for manual and automatic control

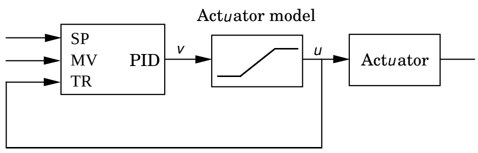
PID Controller with Tracking Mode



No tracking if $w = v$!

Anti-windup for Controller with Tracking Mode





- Notice that there is no tracking effect if $u = v$!
- The tracking input can be used in many other ways

Computer Implementation

Practically all control systems are today implemented using computers. We will briefly discuss some aspects of this.
AD and DA converters are needed to connect sensors and actuators to the computer. A clock is also needed to synchronize the operations. We will discuss

- Sampling and aliasing
- A basic algorithm
- Converting differential equations to difference equations
- Wordlength issues
- Bump-less parameter changes

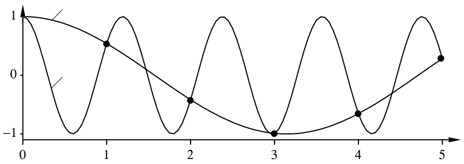
Basic Algorithm

The following operations are executed by the computer.

1. Wait for clock interrupt
2. Convert setpoint y_{sp} and process output y to numbers
3. Compute control signal u
4. Convert control signal to analog value
5. Update variables in control algorithm
6. Go to step 1

Desirable to make time between 1 and 4 as short as possible. Defer as much as possible of the computations to step 5.

Alias and Anti-aliasing Filters



- Nyquist frequency = (Sampling frequency)/2
- High frequencies may appear as low frequencies after sampling
- To represent a continuous signal uniquely from its samples the continuous signal cannot have frequencies above the Nyqyist frequency which which is half the sampling frequency
- Anti-aliasing filters that reduce the frequency content above the Nyquist frequency is essential.

The PID Algorithm

The PID controller is described by:

$$U(s) = P(s) + I(s) + D(s)$$
$$P(s) = k\Big(bY_{sp}(s) - Y(s)\Big)$$
$$I(s) = k\frac{1}{sT_i}(Y_{sp}(s) - Y(s))$$
$$D(s) = -k\frac{sT_d}{1 + sT_d/N}Y(s)$$

Computers can only add and multiply, it cannot integrate or take derivatives. To obtain a programmable algorithm we must approximate. There are many ways to do this.
Introduce the times t_k when the clock ticks, assume that $t_k - t_{k-1} = h$, ,where h is the sampling period.

Proportional and Integral Action

$$p(t_k) = k * (by_{sp}(t_k) - y(t_k))$$

Integral part

$$i(t) = \frac{k}{T_i}\int^t e(s)ds$$

Differentiate

$$\frac{di}{dt} = \frac{k}{T_i}e(t)$$

Approximate the derivative by a difference

$$\frac{i(t_{k+1}) - i(t_k)}{h} = \frac{ke(t_k)}{T_i}$$

This equation can be written as

$$i(t_{k+1}) = i(t_k) + \frac{kh}{T_i}e(t_k)$$

Derivative Part

$$D(s) = -k\frac{sT_d}{1 + sT_d/N}Y(s)$$

Hence

$$(1 + sT_d/N)D(s) = -ksT_dY(s)$$

In time domain

$$d(t) + \frac{T_d}{N}\frac{dd}{dt} = -kT_d\frac{dy}{dt}$$

Approximate derivative by backward difference

$$d(t_k) + \frac{T_d}{N}\frac{d(t_k) - d(t_{k-1})}{h} = -kT_d\frac{y(t_k) - y(t_{k-1})}{h}$$

Derivative Part ...

$$d(t_k) + \frac{T_d}{N}\frac{d(t_k) - d(t_{k-1})}{h} = -kT_d\frac{y(t_k) - y(t_{k-1})}{h}$$

Hence

$$\Big(1 + \frac{T_d}{Nh}\Big)d(t_k) = \frac{T_d}{Nh}d(t_{k-1}) - \frac{kT_d}{h}\big(y(t_k) - y(t_{k-1})\big)$$

or

$$d(t_k) = \frac{T_d}{T_d + Nh}d(t_{k-1}) - \frac{kT_dN}{T_d + Nh}\big(y(t_k) - y(t_{k-1})\big)$$

Notice that the algorithm works well even if T_d is small, this is not the case if forward approximations are used.

Add Windup-protection

$$p(t_k) = k * (by_{sp}(t_k) - y(t_k))$$
$$d(t_k) = \frac{T_d}{T_d + Nh}\Big(d(t_{k-1}) - kN\big(y(t_k) - y(t_{k-1})\big)\Big)$$
$$v = p(t_k) + i(t_k) + d(t_k)$$
$$u(t_k) = \mathrm{sat}(v)$$
$$e(t_k) = y_{sp}(t_k) - y(t_k)$$
$$i(t_{k+1}) = i(t_k) + \frac{kh}{T_i}e(t_k) + \frac{kh}{T_r}(u - v)$$

- Useful to precompute parameters
- Make sure updating is done safely
- Organize the code right

Organize Computations

$$
\begin{aligned}
p(t_k) &= k * (by_{sp}(t_k) - y(t_k)) \\
e(t_k) &= y_{sp}(t_k) - y(t_k) \\
d(t_k) &= \frac{T_d}{T_d + Nh}\Big(d(t_{k-1}) - kN\big(y(t_k) - y(t_{k-1})\big)\Big) \\
v &= p(t_k) + i(t_k) + d(t_k) \\
u(t_k) &= \mathrm{sat}(v) \\
i(t_{k+1}) &= i(t_k) + \frac{kh}{T_i}e(t_k) + \frac{kh}{T_r}(u - v)
\end{aligned}
$$

- Useful to precompute parameters
- Make sure updating is done safely
- Organize the code right

Fix Point Implementation Word-length Issues

Over and under-flow
Consider updating of the integral part

$$i(t_{k+1}) = i(t_k) + \frac{kh}{T_i}e(t_k)$$

Example

- h=0.05 s
- T_i=5000 s
- k=1
- $\frac{kh}{T_i} = 10^{-5}$

If the error has 3 digits the integral need to be updated with 8 digits (28 bits) to avoid rounding off the errors!

Bump-less Parameter Changes

A PID controller is often switched between three modes: off, manual and automatic control. It is important that there are no switching transients.

- Notice the difference between

$$I = k_i(t)\int_0^t e(\tau)d\tau, \qquad I = \int_0^t k_i(\tau)e(\tau)d\tau$$

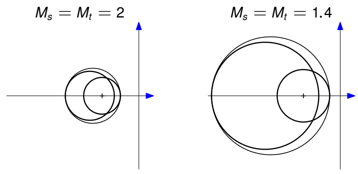
- Integration and multiplication with a time varying function do not commute!
- Some controllers require that you switch to manual mode to change parameters
- Problem is avoided by proper coding

```
"Compute controller coefficients
    p1=K*b                               "set-point gain
    p2=K+K*Td/(Tf+h)                      "PD gain
    p3=Tf/(Tf+h)                          "filter constant
    p4=K*Td*h/((Tf+h)*(Tf+h))              "derivative gain
    p5=K*h/Ti                            "integral gain
    p6=h/Tt                              "anti-windup gain
"Bumpless parameter changes
    I=I+Kold*(bold*ysp-y)-Knew*(bnew*ysp-y)
"Control algorithm
    adin(ysp)                            "read set point
    adin(y)                              "read process variable
    v=p1*ysp-p2*y+x+I                     "compute nominal output
    u=sat(v,ulow,uhigh)                   "saturate output
    daout(u)                             "set analog output
    x=p3*x+p4*y                          "update derivative
    I=I+p5*(ysp-y)+p6*(u-v)               "update integral
```

PID Control

Theme: The most common controller.

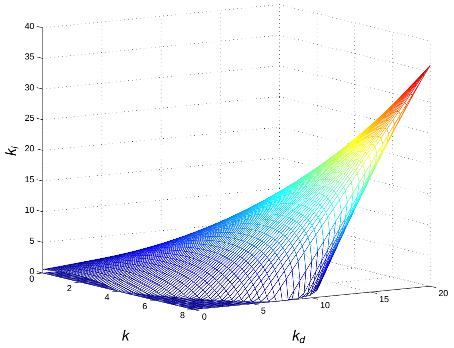
Circular Constraints on Sensitivities



Contour	Center	Radius
M_s	-1	$1/M_s$
M_t	$-\frac{M_t^2}{M_t^2-1}$	$\frac{M_t}{M_t^2-1}$
M_s, M_t	$-\frac{M_s(2M_t-1)-M_t+1}{2M_s(M_t-1)}$	$\frac{M_s+M_t-1}{2M_s(M_t-1)}$
$M_s = M_t = M$	$-\frac{2M^2-2M+1}{2M(M-1)}$	$\frac{2M-1}{2M(M-1)}$

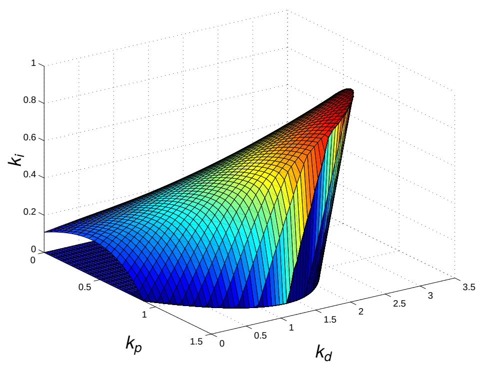
Stability Region for $P = (s+1)^{-4}$

– Derivative Cliff!



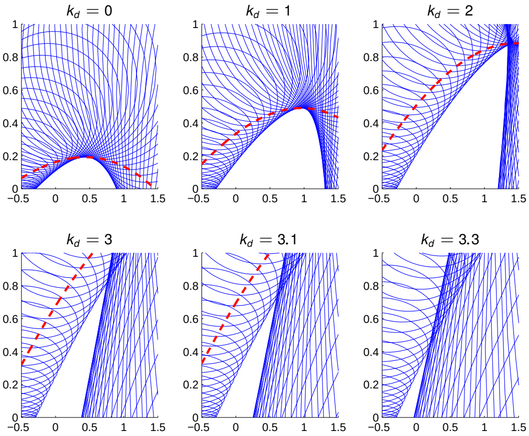
Explains why derivative action is difficult
Don't fall off the edge!

Robustness Region for $P = (s+1)^{-4}$ & $M_s \leq 1.4$

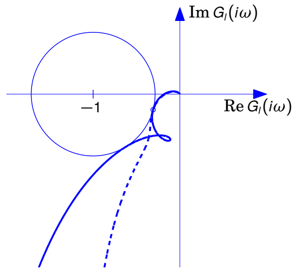


Compare with stability region

Projections on the k_p — k_i plane - Edge constraints

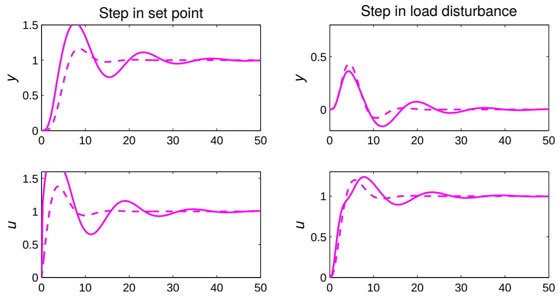


Edges Correspond to Cusps in the Nyquist Plot



Nyquist curve of the loop transfer function for PID control of the process $P(s) = 1/(s+1)^4$, with a controller having parameters $k_p = 0.925$, $k_i = 0.9$, and $k_d = 2.86$.
Cusps are avoided in this example by minimizing IAE instead (dashed curve) $k_p = 1.33$, $k_i = 0.63$, and $k_d = 1.78$

Time Responses



Process $P(s) = 1/(s+1)^4$, with controller having parameters $k_p = 0.925$, $k_i = 0.9$, and $k_d = 2.86$ (max k_i solid lines IAE=3.0) and $k_p = 1.33$, $k_i = 0.63$, and $k_d = 1.78$ (min IAE=2.2 dashed lines).
Damping ratios of zeros $\zeta = 0.16$ and 0.37.

PID Control

Theme: The most common controller.

Requirements

Disturbances
- Effect of feedback on disturbances
- Attenuate effects of load disturbances
- Moderate measurement noise injection

Robustness
- Reduce effects of process variations
- Reduce effects of modeling errors

Command signal response
- Follow command signals
- Architectures with two degrees of freedom (2DOF)

Tune for Load Disturbances - Shinskey 1993

"The user should not test the loop using set-point changes if the set point is to remain constant most of the time. To tune for fast recovery from load changes, a load disturbance should be simulated by stepping the controller output in manual, and then transferring to auto. For lag-dominant processes, the two responses are markedly different."

Process control: Tune k_p, k_i, k_d and T_f for load disturbances, measurement noise and robustness, then tune β, and γ for setpoint response (set point weighting)

$$u(t) = k_p\big(\beta r(t) - y_f(t)\big) + k_i \int_0^t \big(r(\tau) - y_f(\tau)\big)d\tau + k_d\Big(\gamma \frac{dr}{dt} - \frac{dy_f}{dt}\Big)$$

$$Y_f(s) = \frac{1}{1 + sT_f + s^2 T_f^2/2} Y(s)$$

Performance

Disturbance reduction by feedback

$$Y_{cl} = SY_{ol} = \frac{1}{1+PC} Y_{ol}$$

Load disturbance attenuation (typically low frequencies)

$$G_{yd} = \frac{P}{1+PC} \approx \frac{s}{k_i}, \qquad -G_{ud} = \frac{PC}{1+PC}$$

Measurement noise injection (typically high frequencies)

$$G_{xn} = \frac{PC}{1+PC}, \qquad -G_{un} = \frac{C}{1+PC} \approx C = G_f\Big(k_p + \frac{k_i}{s} + k_d s\Big)$$

Command signal following

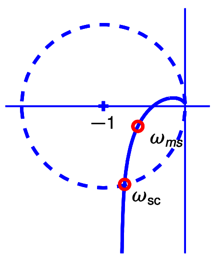
$$G_{xr} = \frac{PG_f(\gamma k_d s^2 + \beta k_p s + k_i)}{s + PG_f(k_d s^2 + k_p s + k_i)}, \; G_{ur} = \frac{G_f(\gamma k_d s^2 + \beta k_p s + k_i)}{s + PG_f(k_d s^2 + k_p s + k_i)}$$

Effects ofLoad Disturbances

Compare open and closed loop systems!

$$\frac{Y_{cl}}{Y_{ol}} = \frac{1}{1+PC} = S$$

Geometric interpretation: Disturbances with frequencies outside are reduced. Disturbances with frequencies inside the circle are amplified by feedback, the maximum amplification is M_s.
Disturbances with frequencies less than sensitivity crossover frequency ω_{sc} are reduced by feedback.

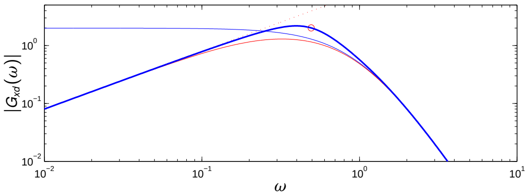


Load Disturbance Attenuation

Transfer function from load disturbance d to process outpur y ($P(0) = K$)

$$G_{yd} = \frac{P}{1+PC} = SP \approx \frac{1}{C} \approx sk_i, \qquad \text{low frequencies}$$
$$G_{yd} = \frac{P}{1+PC} = SP \approx P, \qquad \text{high frequencies}$$

$P = 2(s+1)^{-4}$ PI: $k_p = 0.5$, $k_i = 0.25$



Criteria IE and IAE

Traditionally the criteria

$$IE = \int_0^\infty e(t)dt, \quad IAE = \int_0^\infty |e(t)|dt, \quad IE2 = \int_0^\infty e^2(t)dt$$
$$ITAE = \int_0^\infty t\,|e(t)|dt, \quad QE = \int_0^\infty (e^2(t) + \rho u^2(t))dt$$

where e is the error for a unit step in the set point or the load disturbance have often been used to evaluate PID controllers

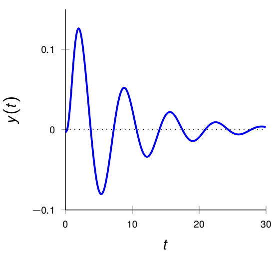
Notice that for a step u_0 in the load disturbance we have

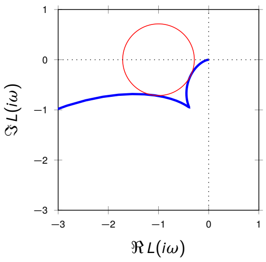
$$u(\infty) = k_i \int_0^\infty e(t)dt$$

For a unit step disturbance we have $u(\infty) = 1$ and hence $IE = 1/k_i$. If the responses are well damped we have $IE \approx IAE$ and integral gain is then a measure of load disturbance attenuation.

Advantages and Disadvantages with IE

Advantage: $IE = \frac{1}{k_i}$, the difficulty is that it gives poor damping in some cases





IE Curvature Constraint or IAE for $P = (s+1)^{-3}$

$C_{IE} = 3.31 + \frac{6.62}{s} + 6.26s$

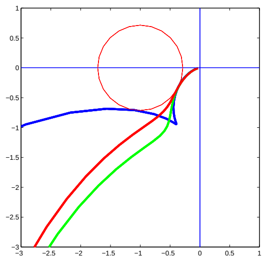
$IAE = 0.74$

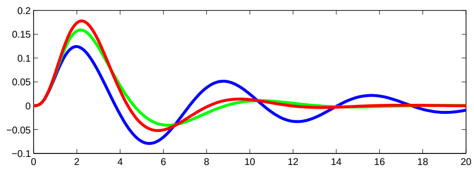
$C_\kappa = 3.61 + \frac{3.20}{s} + 3.34s$

$IAE = 0.57$

$C_{IAE} = 3.81 + \frac{3.33}{s} + 4.25s$

$IAE = 0.53$





Robustness

Gain and phase margins g_m and φ_m

Maximum sensitivities $M_s = \max_\omega |S(i\omega)|$, $M_t = \max_\omega |T(i\omega)|$

$$H = \frac{1}{1+PC}\begin{bmatrix} 1 & P \\ C & PC \end{bmatrix} = \begin{bmatrix} \frac{1}{1+PC} & \frac{P}{1+PC} \\ \frac{C}{1+PC} & \frac{PC}{1+PC} \end{bmatrix}$$

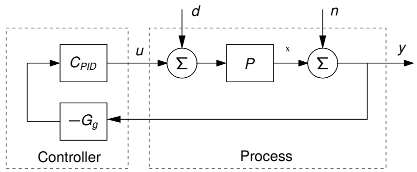
Dimensions! For SISO systems the $\mathcal{H}_\infty$ norm of G_s is

$$\gamma^2 = \max \frac{(1+|P|^2)(1+|C|^2)}{|1+PC|^2}$$

Scale process $P \to \alpha P$ and controller $C \to C/\alpha$, minimize with respect to α

$$\gamma = \max \frac{1+|PC|}{|1+PC|} = \max\Big(\Big|\frac{1}{1+PC}\Big| + \Big|\frac{PC}{1+PC}\Big|\Big) \le M_s + M_t$$

Measurement Noise Injection



Controller transfer function

$$G_f = \frac{1}{1 + sT_f + s^2T_f^2/2} \qquad C_{PID}(s) = k_p + \frac{k_i}{s} + k_d s, \qquad C = C_{PID}G_f$$

Transfer function from measurement noise n to control signal u

$$-G_{un}(s) = -\frac{C}{1+PC} = -SC \approx -\frac{s}{s+Kk_i} \times \frac{k_i + k_p s + k_d s^2}{s(1+sT_f + (sT_f)^2/2)}$$

Only controller parameters and $K = P(0)$

Stochastic Modeling of Measurement Noise

Measurement noise stationary with spectral density $\Phi(\omega)$

$$\sigma_u^2 = \int_{-\infty}^\infty |G_{un}(i\omega)|^2\Phi(\omega)d\omega, \quad \sigma_{y_f}^2 = \int_{-\infty}^\infty |G_f(i\omega)|^2\Phi(\omega)d\omega$$

$$G_{un}(s) \approx -\frac{k_i + k_p s + k_d s^2}{(s+Kk_i)(1+sT_f+(sT_f)^2/2)}$$

$$\sigma_u^2 \approx \pi\Big(\frac{k_i}{K} + \frac{k_p^2 - 2k_ik_d}{T_f} + 2\frac{k_d^2}{T_f^3}\Big)\Phi_0, \quad \sigma_{y_f}^2 = \frac{\pi}{T_f}\Phi_0$$

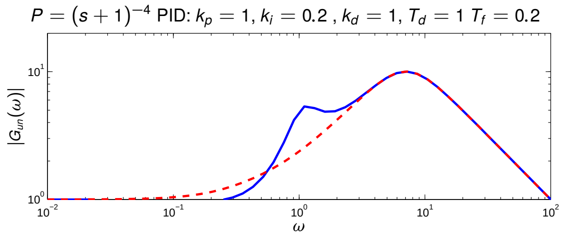
Noise gain $k_n = \sigma_u/\sigma_{y_f}$ and SDU (standard deviation of u with white measurement noise $\Phi_0 = 1$)

$$k_{nw} = \frac{\sigma_u}{\sigma_{y_f}} \approx \sqrt{\frac{k_iT_f}{K} + k_p^2 - 2k_ik_d + 2\frac{k_d^2}{T_f^2}}$$

$$\pi\Phi_0 = 1 \Rightarrow \sigma_u = SDU = \sqrt{\Big(\frac{k_i}{K} + \frac{k_p^2 - 2k_ik_d}{T_f} + 2\frac{k_d^2}{T_f^3}\Big)}$$

Measurement Noise Injection

$P = (s+1)^{-4}$ PID: $k_p = 1$, $k_i = 0.2$, $k_d = 1$, $T_d = 1$ $T_f = 0.2$

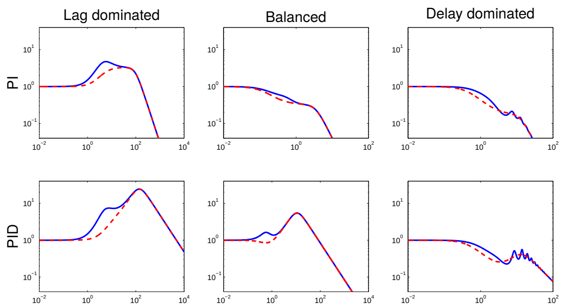


First order filter (dashed), second order filter (full)

$$-G_{un} = CS \approx \frac{k_ds^2 + k_ps + k_i}{s(1+sT_f+(sT_f)^2/2)} \times \frac{s}{s+Kk_i}$$

Peaks of G_{un} at ω_{ms} and at $\omega \approx \sqrt{2}/T_f$

Bode Plots of Noise Transfer Function G_{un}



- Validity of approximation (error in mid frequency range M_s peak)
- Differences PI/PID lag dominated/delay dominated

PID Control

Theme: The most common controller.

Empirical Tuning Rules

- When do you need rules?
- Why not model by physics or experiments and design a controller?
- Typical processes - essentially monotone - modeled by FOTD
- Ziegler-Nichols Tuning 1942 (for historical reasons)
- Lambda tuning - Common in pulp and paper industry
- SIMC - Skogestad: *Probably the best simple PID tuning rules in the world*
- Optimization, criteria and constraints
- AMIGO - Minimize IE, maiximze Integral gain subject to robustness constraint and edge constraint for PID
- MIAEO - Minimize IAE subject to robustness constraint (for local reasons and insight)
- How to get the models?

The FOTD Model - A Common Special Case

$$P(s) = \frac{K}{1+sT}e^{-sL}$$

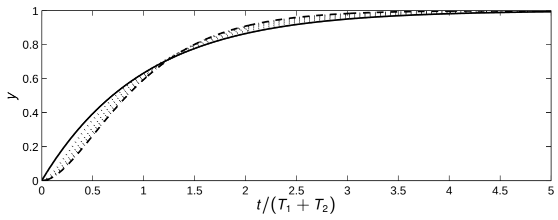
- *L time delay*, T time constant or *lag*
- Approximation of processes with (almost) monotone step responses
- Commonly used in process control and for PID tuning
- Performance limited by time delay $\omega_{gc}L < 1$. Useful to have a simple model that captures performance limitations
- Average residence time $T_{ar} = L + T$
- Delay ratio $\tau = L/T_{ar} = L/(L+T)$ $\quad 0 \le \tau \le 1$ is useful to classify dynamics

  Lag dominant: τ close to 0
  Balanced: τ around 0.5
  Delay dominant τ close to 1

A Difficulty in Step Response Modeling

Normalized step responses for

$$P(s) = \frac{1}{(1+sT_1)(1+sT_2)}, \qquad T_1/T_2 = 0, 0.1, \ldots 1$$



Difficult to estimate T_1 and T_2

Ziegler-Nichols Tuning - Commissioning

Process control scenario: You have a controller with adjustable parameters and a process. How do you find suitable values of the controller parameters? Ziegler-Nichols idea was to tune controller based on simple experiments on the process

- The step response method - open loop experiment

  Make an open loop step response (bump test)
  Pick out features of the step response and determine parameters from a table
- The frequency response method - closed loop

  Connect the controller change controller parameters, observe process behavior and adjust parmeters

The rules were developed by picking out typical process models, tuning controller by hand or simulation (MITs differential analyzer and pneumatic), and correlating controller parameters to process features

Assessment of Ziegler-Nichols Methods

Great simple idea: base tuning on simple process experiments,

- Published in 1942 in Trans. ASME 64 (1942) 759–768.
- Tremendously influential for establishing process control
- Slight modifications used extensively by controller manufacturers and process engineers
- The Million \$ question: What structure (series or parallel) did they use?

BUT poor execution

- Uses too little process information: only 2 parameters

  Step response method: a, L
  Frequency response method: T_u, K_u
- Basic design principle quarter amplitude damping is not robust, gives closed loop systems with too high sensitivity ($M_s > 3$) and too poor damping ($\zeta \approx 0.2$)

Lambda Tuning

Process model and desired command response

$$P(s) = \frac{K_p}{1+sT}e^{-sL}. \qquad G_{yy_{sp}} = \frac{1}{1+sT_{cl}}e^{-sL}.$$

The controller becomes

$$C(s) = P^{-1}(s)\frac{G_{yy_{sp}}(s)}{1 - G_{yy_{sp}}(s)} = \frac{1+sT}{K_p(1+sT_{cl} - e^{-sL})},$$

Cancellation of the process pole $s = -1/T$!! Approximations of e^{-sL} give PI and PID controllers, for example $e^{-sL} \approx 1 - sL$

$$C(s) = \frac{1+sT}{K_p(L+T_{cl})s} = \frac{T}{K_p(L+T_{cl})}\Big(1 + \frac{1}{sT}\Big)$$

PI controller with the parameters

$$k_p = \frac{1}{K_p}\frac{T}{L+T_{cl}}, \quad k_i = \frac{1}{K_p(L+T_{cl})}, \quad T_i = T.$$

Closed loop response time $T_{cl} = \lambda_f T$ is a design parameter, common choices $\lambda_f = 3$ (robust tuning), $\lambda_f \le 1$ aggressive tuning.

Lambda Tuning - Gang of Four

$$S = \frac{s(L+T_{cl})}{s(L+T_{cl})+e^{-sL}} \approx \frac{s(L+T_{cl})}{1+sT_{cl}}$$

$$PS = \frac{sK_p(L+T_{cl})}{\big(s(L+T_{cl})+e^{-sL}\big)(1+sT)}e^{-sL} \approx \frac{sK_p(L+T_{cl})}{(1+sT_{cl})(1+sT)}e^{-sL}$$

$$CS = \frac{s(T+T_{cl})(1+sT)}{\big(s(L+T_{cl})+e^{-sL}\big)(1+sT)} \approx \frac{(L+T_{cl})(1+sT)}{K(L+T_{cl})(1+sT_{cl})}$$

$$T = \frac{e^{-sL}}{s(L+T_{cl})+e^{-sL}} \approx \frac{1}{1+sT_{cl}}e^{-sL}.$$

- Very nice to have a tuning parameter T_{cl} with good physical interpretation
- Perhaps better to pick T_{cl} proportional to L
- Notice presence of canceled mode $s = -1/T$ in PS, very poor load disturbance response if $T_{cl} < T$

Skogestad SIMC

Process models

$$P_1(s) = \frac{K_p}{1+sT}e^{-sL}, \qquad P_2(s) = \frac{K_p}{(1+sT_1)(1+sT_2)}e^{-sL}.$$

Desired closed-loop transfer function

$$G_{yy_{sp}} = \frac{1}{1+sT_{cl}}e^{-sL}.$$

Hence

$$C(s) = \frac{1}{P} \times \frac{G_{yy_{sp}}}{1-G_{yy_{sp}}} = \frac{1+sT}{K_p(1+sT_{cl}-e^{-sL})} \approx \frac{1+sT}{sK_p(T_{cl}+L)}$$

typical choices of design parameter $T_{cl} = \lambda_f L$. Control law

$$k_p = \frac{1}{K_p}\frac{T}{L+T_{cl}}, \qquad T_i = \min\big(T, 4(T_{cl}+L)\big).$$

Fixes after lots of simulations SIMC++

$$k_p = \frac{1}{K_p}\frac{T+L/3}{L+T_{cl}}, \qquad T_i = \min\big(T+L/3, 4(T_{cl}+L)\big), \qquad T_{cl} = \lambda L.$$

Tore's One Third Rule "Tredjedels regeln"

- Make a unit step test
- Determine the static gain K_p and the time T_p for the process to reach 95% of its steady state value
- The controller parameters are

$$K = \frac{1}{3K_p}, \qquad T_i = \frac{T_P}{3} = T + \frac{L}{3}$$

Some Tuning Rules for PI Control

- Ziegler-Nichols step

$$k_p = \frac{0.9}{K_v L}, \qquad k_i = \frac{0.27}{K_v L^3}, \qquad T_i = L/0.3$$

- Ziegler-Nichols frequency

$$k_p = 0.45k_u, \qquad k_i = 0.54\frac{k_u}{T_u}, \qquad T_i = T_u/1.2$$

- Lambda Tuning - $T_{cl} = T, 2T, 3T$

$$k_p = \frac{T}{K(T_{cl}+L)}, \qquad k_i = \frac{1}{K(T_{cl}+L)}, \qquad T_i = T$$

- Skogestad SIMC Like Lambda but $T_i = min(T, 4(T_{cl}+L))$
- Skogestad SIMC+

$$k_p = \frac{T+L/3}{K(T_{cl}+L)}, \qquad T_i = min(T+L/3, 4(T_{cl}+L))$$

- Tore One Third Rule

$$k_p = \frac{1}{3K}, \qquad T_i = T + \frac{L}{3}$$

- AMIGO ($M_s, M_t = 1.4$)

$$k_p = \frac{0.15}{K} + \Big(0.35 - \frac{LT}{(L+T)^2}\Big)\frac{T}{KL}, \qquad T_i = 0.35L + \frac{13LT^2}{T^2+12LT+7L^2}$$

PID Control

1. Introduction
2. The Controller
3. Stability
4. Performance and Robustness
5. Empirical Tuning Rules
6. Tuning Based on Optimization
7. Relay Auto-tuning
8. Limitations of PID Control
9. Summary

Theme: The most common controller.

Tuning based on Optimization

A reasonable formulation of the design problem is to optimize performance subject to constraints on robustness and noise injection.

- Performance criteria IE or IAE for load disturbance attenuation
  - Small differences between IE and IAE for PI
  - Larger differences for PID because of derivative cliff use IAE
  - With IE it is necessary to use an *edge* constraint
- Constraints
  - Robustness M_s and M_t
  - Noise injection max $|G_{un}(i\omega)|$ or $||G_{un}||_2$
- Pick a class of representative processes
- Pick a design criterion: Maximize integral gain subject to constraints on robustness M_s and M_t MIGO (M-constrained Integral Gain Optimization)
- Relate controller parameters to FOTD model $Ke^{-sL}/(1+sT)$
- Rules for PI control, conservative rules for PID control
- Insight and understanding

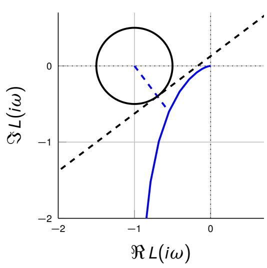
Solving the Optimization Problem

Boyd Hast Berhardsson

- Load disturbance attenuation IAE!
- Robustness M_s M_t
- Measurement noise SDU, k_n
- Loop transfer function

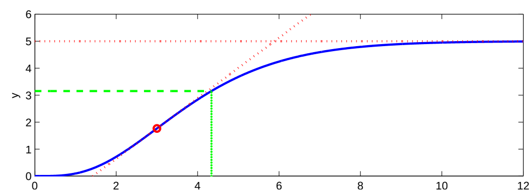
$$G_l = PG_f\Big(k_p + \frac{k_i}{s} + k_d s\Big)$$

- Convex optimization



How to Get the Models

Bump test



Relay feedback
Model reduction - Skogestads half rule
System identification
Modeling and control design should match

The Test Batch

$$P_1(s) = \frac{e^{-s}}{1+sT}, \qquad P_2(s) = \frac{e^{-s}}{(1+sT)^2}$$

$$P_3(s) = \frac{1}{(s+1)(1+sT)^2}, \qquad P_4(s) = \frac{1}{(s+1)^n}$$

$$P_5(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$$

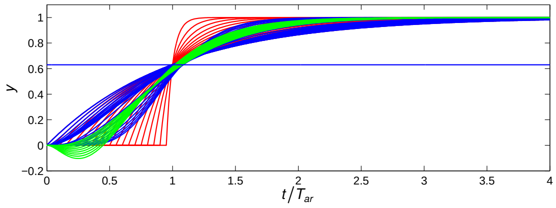
$$P_6(s) = \frac{1}{s(1+sT_1)}e^{-sL_1}, \quad T_1 + L_1 = 1$$

$$P_7(s) = \frac{T}{(1+sT)(1+sT_1)}e^{-sL_1}, \qquad T_1 + L_1 = 1$$

$$P_8(s) = \frac{1-\alpha s}{(s+1)^3}$$

$$P_9(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)}$$

Essentially Monotone Step Responses

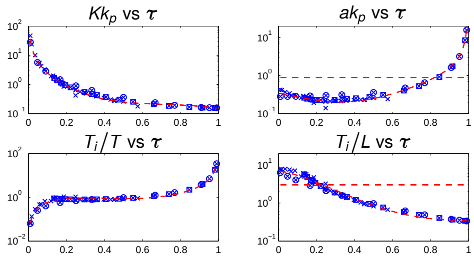


Step responses for test batch normalized by the average residence time $T_{ar} = \int tg(t)dt / \int g(t)dt = -P'(0)$. Empirical criterion for monotonicity

$$a = \frac{\int_0^\infty e(t)dt}{\int_0^\infty |e(t)|dt}, \qquad \text{essentially positive if} \quad a > 0.8$$

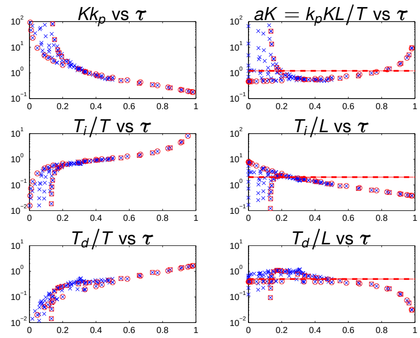
Positive systems is a research issue (Sontag)

PI Control: Minimize IAE $M = 1.4$ - Correlation with FOTD parameters



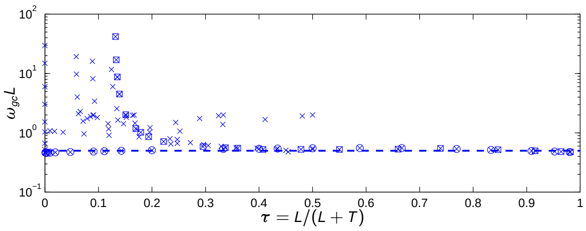
PI Control can be based on an FOTD model

PID Control: Minimize IAE, M_s, $M_t \leq 1.4$



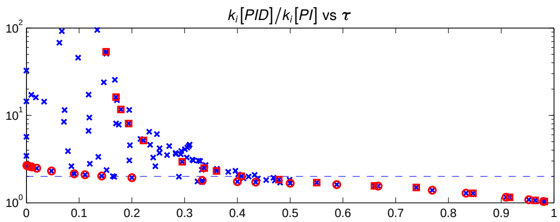
- Tuning rules based on FOTD can be found for $\tau > 0.3$
- More complex models required for lag dominated dynamics
- Limiting cases $\frac{K}{1+sT}e^{-sL}$ and $\frac{K}{(1+sT/2)^2}e^{-sL}$

An Observation



- Compare with fundamental limit due to time delay $\omega_{sc}L < \frac{2(M_s-1)}{M_s} \approx 0.57$
- Close to limit for P_1 (red circles) for all τ
- Close to limit for whole batch for $\tau > 0.3$
- Reason for large variability for small τ is that the FOTD model overestimates L for lag dominated systems, high order dynamics approximated by time delay

Benefit of Derivative Action



- Derivative action gives small benefits for processes with delay dominated dynamics (derivative is a poor predictor for systems which are dominated by time delay)
- Derivative action doubles performance for $\tau = 0.5$
- Significant may be possible for small τ, but better modeling may be required, notice difference between P_1 (red circles) and P_2 (red squares)
- Processes with small τ are easy to control and admit very high gains. In practice the admissible gains are limited by sensor noise. A PI controller will often work well.

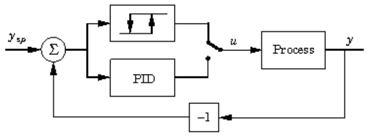
Summary

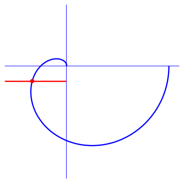
- Processes with essentially monotone step responses
- The FOTD model gives insight
- Realize difference between lag and delay dominated dynamics τ
- PI is sufficient for processes with delay dominated dynamics
- Advantage of derivative action increases with decreasing τ
- Derivative action doubles performance for $\tau = 0$.
- Derivative action may give significant improvement for processes with lag dominated dynamics but more complex models may be useful
- Processes with small τ admit high controller gains and performance may be limited by noise injection, a PI controller may then be sufficient
- AMIGO and Skogestad SIMC+ are reasonable rules
- Modeling is essential

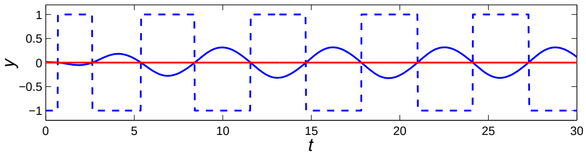
PID Control

Theme: The most common controller.

Relay Auto-tuning



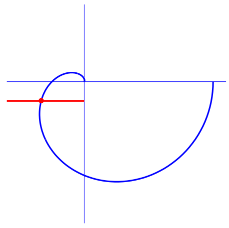




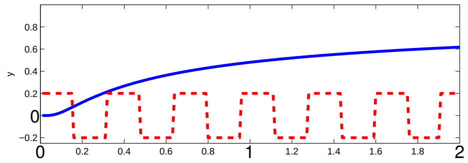
Relay feedback creats oscillation at ω_{180}!
Automation of ZN frequency response method modified ZN tuning rules

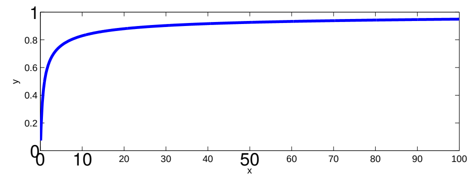
Practical Details

- Bring process to equilibrium
- Measure noise level
- Compute hysteresis width
- Initiate relay
- Monitor each half period
- Change relay amplitude automatically
- Check for steady state
- Compute controller parameters
- Resume PID control



Short Experiment Time $G(s) = exp(-\sqrt{s})$





Extreme but not unusual case!

Commercial Autotuners

- One-button autotuning
- Three settings: fast, slow, delay dominated
- Automatic generation of gain schedules
- Adaptation of feedback gains
- Adaptation of feedforward gain
- Many versions
  - Single loop controllers
  - DCS systems
- Robust
- Excellent industrial experience
- Large numbers

ECA 600

Industrial Impact

Functions

- Automatic tuning AT
- Automatic generation of gain scheduling GC
- Adaptive feedback AFB and adaptive feedforward AFF

Sample of products

- NAF Controls SDM 20 - 1984 DCS AT, GS
- SattControl ECA 40 - 1986 SLC AT, GS
- Satt Control ECA 04 - 1988 SLC AT
- Alfa Laval Automation Alert 50 - 1988 DCS AT, GS
- Satt Control SattCon31 - 1988 PLC AT, GS
- Satt Control ECA 400 -1988 2LC AT, GS, AFB, AFF
- Fisher Control DPR 900 - 1988 SLC
- Satt Control SattLine - 1989 DCS AT, GS, AFB, AFF
- Emerson Delta V - 1999 DCS AT, GS, AFB, AF
- ABB 800xA - 2004 DCS AT, GS, AFB, AFF

Properties of Relay Auto-tuning

- Safe for stable systems
- Close to industrial practice
  - Easy to explain similar to Ziegler-Nichols tuning
- Little prior information. Relay amplitude
- One-button tuning
- Automatic generation of test signal
  - Injects much energy at ω_{180} with no prior knowledge of ω_{180}
  - Easy to modify for signal injection at other frequencies
- Good industrial experience for more than 25 years. Many patents are running out.
- Good for pre-tuning of adaptive controllers
- Still room for improvement
  - Exploit advances in computing
  - Exploit understanding of modeling and controller design

A Millon Dollar Question
Classify all linear systems which have stable limit cycles under relay feedback!

PID Control

1. Introduction
2. The Controller
3. Stability
4. Performance and Robustness
5. Empirical Tuning Rules
6. Tuning based on Optimization
7. Relay Auto-tuning
8. Limitations of PID Control
9. Summary

Theme: The most common controller.

Limitations of PID Control

PID control is simple and useful but there are limitations

- Multivariable and strongly coupled systems
- Complicated dynamics
- Large parameter variations
  - Robust design
  - Gainscheduling and adaptation
- Difficult compromises between load disturbance attenuation and measurement noise injection

Complex Controllers

Complex controllers can be built bottom up by combining

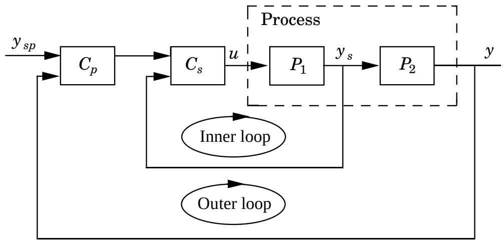
- PID Controllers
- Nonliner elements
- Logic
- Observers

Using control principles such as

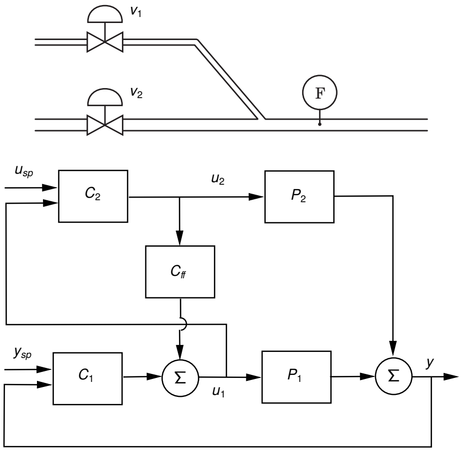
- Cascade control
- Mid-ranging and Split-ranging
- Selector control
- Ratio control

to deal with more complicated control problems.
Such solutions become very complicated for systems with many inputs, outputs and constraints on control variables and state variables. Model predictive control is often a viable substitute.

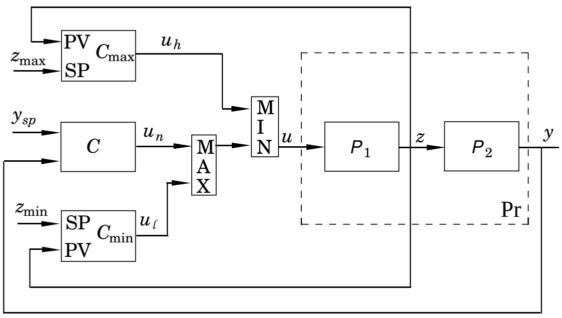
Cascade Control - Many Sensors



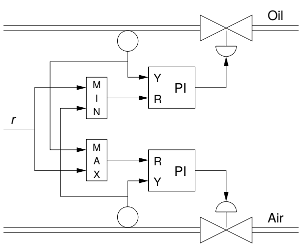
Midrange Control - Many Actuators



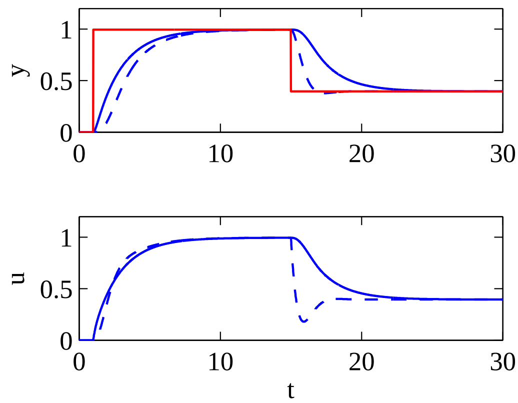
Selector Control - Equipment protection



Selector Control - Safe Operation



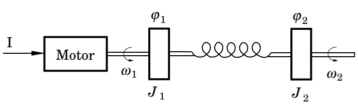
Selector Control - Safe Operation 2



Full line air, dashed line oil

Complicated Dynamics

- Any stable system can be controlled by an integrating controller if performance requirements are modest
- PI control and systems with first order dynamics
- PID control and systems with second order dynamics
- States are the variables required to account for storage of mass, energy and momentum



Transfer function (physical meaning of approximation)

$$P(s) = \frac{0.045s + 0.45}{s^2(s^2 + 0.1s + 1)} \approx \frac{0.45}{s^2}$$

PID Control

With an ideal PID controller and the approximate model the loop transfer function is

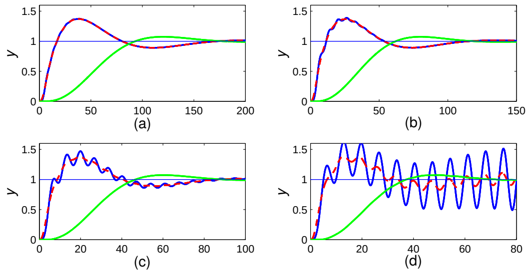
$$L(s) = \frac{0.45(k_d s^2 + k_p s + k_i)}{s^3}$$

We will add high frequency roll-off later. Closed loop characteristic polynomial

$$s^3 + 0.45k_d s^2 + 0.45k_p s + 0.45k_i = s^3 + 2\omega_c s^2 + 2\omega_c^2 s + \omega_c^3$$
$$(s + \omega_c)(s^2 + \omega_c s + \omega_c^2), \qquad \text{Butterworth}$$

The approximation is valid if ω_c small (say $\omega_c < 0.1\omega_0$. Increasing ω_c leads to instability. The bandwidth and the performance $k_i = \omega_c^3/0.45$ are limited.

PID Control ...

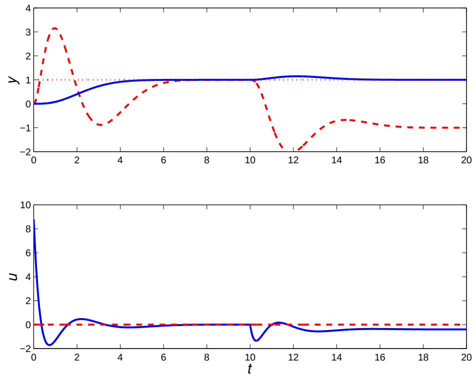


$\omega_c/\omega_0 =$ a) 0.04, b) 0.06, c) 0.08 d) 0.1

φ_1 blue, φ_2 red, setpoint weighting green

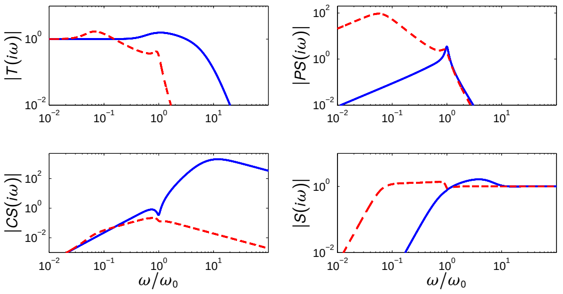
With low bandwidth controller the inertias move together

Observer and State Feedback



φ_1 blue, φ_2 red

Comparison PID SFB - GoF



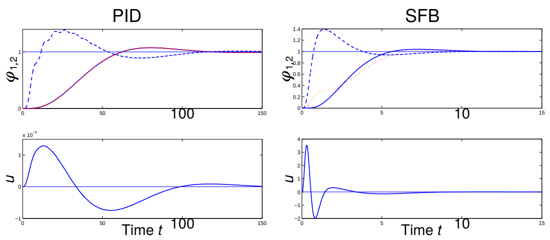
PID is designed for $\omega_c = 0.06\omega_0$

PID red dashed SFB blue

Notice orders of magnitude

SFB requires high quality low noise sensors

Comparison PID SFB Command Response

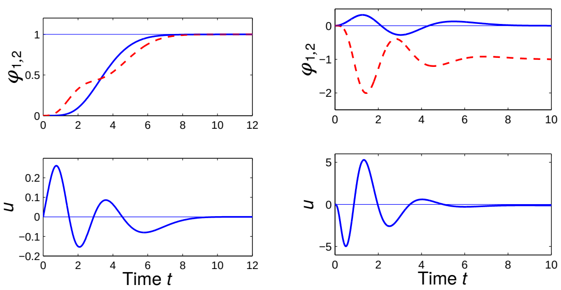


notice time scales and control signal amplitudes!

SFB gives ten times faster response

φ_1 red dotted, φ_2 blue solid, dashed without 2DOF

Set Point and Load Disturbance Response SFBI



φ_1 red dotted, φ_2 blue solid

Explain behavior of inertias!

PID Control

1. Introduction
2. The Controller
3. Performance and Robustness
4. Empirical Tuning Rules
5. Tuning based on Optimization
6. Relay Auto-tuning
7. Limitations of PID Control
8. Summary

Theme: The most common controller.

Summary

- A simple and useful controller
- Much tradition and legacy
- Many things to consider: set point weighting, filtering, windup protection, mode switching and tracking modes
- Many design methods relative time delay τ is important to classify
- Good models can be obtained by relay feedback
- Next generation auto-tuners are almost here
- There are processes where PID can be outperformed significantly
  - Multivariable systems and constraints
  - Oscillatory systems
- The Million dollar question: Find all linear systems that give a stable oscillation under relay feedback!