Introduction to Control (034040)
lecture no. 12

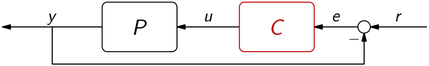
Leonid Mirkin

Faculty of Mechanical Engineering
Technion—IIT

Outline

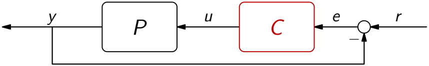
Outline

PID regulators



– P: $C(s) = k_{\mathrm{p}}$;

PID regulators



- P: $C(s) = k_{\mathrm{p}}$;
- PI:

$$C(s) = k_{\mathrm{p}}\left(1 + \frac{k_{\mathrm{i}}}{s}\right) = k_{\mathrm{p}}\left(1 + \frac{1}{\tau_{\mathrm{i}} s}\right),$$

with $\tau_{\mathrm{i}} = 1/k_{\mathrm{i}}$ called its reset time;

- PD:

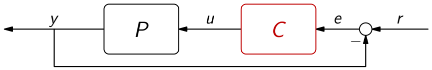
$$C(s) = k_{\mathrm{p}}(1 + \tau_{\mathrm{d}} s),$$

with τ_{d} called its derivative time;

- PID:

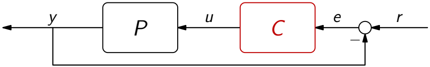
$$C(s) = k_{\mathrm{p}}\left(1 + \frac{1}{\tau_{\mathrm{i}} s} + \tau_{\mathrm{d}} s\right)$$

PID regulators



- P: $C(s) = k_{\mathrm{p}}$;
- PI:
$$C(s) = k_{\mathrm{p}}\left(1 + \frac{k_{\mathrm{i}}}{s}\right) = k_{\mathrm{p}}\left(1 + \frac{1}{\tau_{\mathrm{i}} s}\right),$$
with $\tau_{\mathrm{i}} = 1/k_{\mathrm{i}}$ called its reset time;
- PD:
$$C(s) = k_{\mathrm{p}}(1 + \tau_{\mathrm{d}} s),$$
with τ_{d} called its derivative time;
- PID:
$$C(s) = k_{\mathrm{p}}\left(1 + \frac{1}{\tau_{\mathrm{i}} s} + \tau_{\mathrm{d}} s\right) \quad \text{or}$$

PID regulators



- P: $C(s) = k_\mathrm{p}$;
- PI:

$$C(s) = k_\mathrm{p}\left(1 + \frac{k_\mathrm{i}}{s}\right) = k_\mathrm{p}\left(1 + \frac{1}{\tau_\mathrm{i} s}\right),$$

  with $\tau_\mathrm{i} = 1/k_\mathrm{i}$ called its reset time;
- PD:

$$C(s) = k_\mathrm{p}(1 + \tau_\mathrm{d} s),$$

  with τ_d called its derivative time;
- PID:

$$C(s) = k_\mathrm{p}\left(1 + \frac{1}{\tau_\mathrm{i} s} + \tau_\mathrm{d} s\right) \quad \text{or} \quad C(s) = k_\mathrm{p}\left(1 + \frac{1}{\tau_\mathrm{i} s}\right)\left(1 + \tau_\mathrm{d} s\right).$$

PID regulators: why

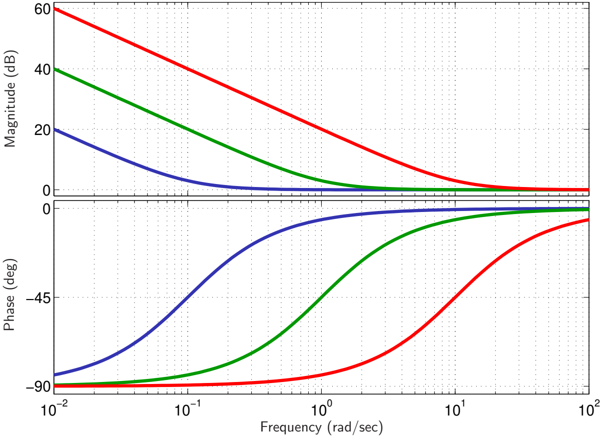
- Relatively simple structure
  - (relatively) easy to implement
  - (relatively) easy to tune

- Intuitively clear interpretation
  - the “P”-part exploits current knowledge of system behavior
  - the “I”-part exploits the past
  - the “D”-part attempts to exploit prediction of the future

- Vast practical experience

Frequency domain interpretation of PI controller

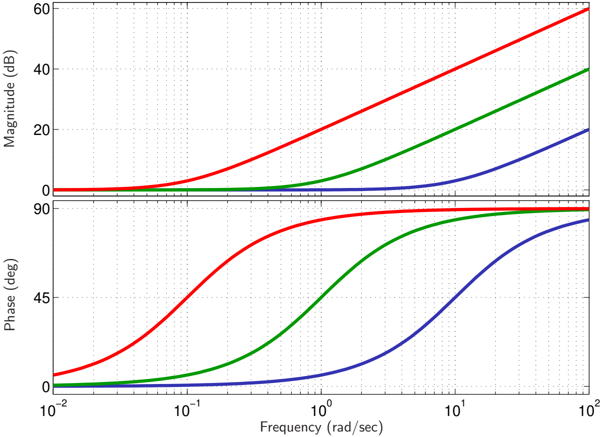
Bode diagram of $1 + \frac{1}{\tau_i s}$ for $\tau_i = 10, 1, 0.1$



- PI controller adds additional phase lag thus reducing stability margins
- to prevent this, τ_i should be such that $\omega_c \tau_i \gg 1$

Frequency domain interpretation of PD controller

Bode diagram of $1 + \tau_d s$ for $\tau_d = 0.1, 1, 10$



- PD controller adds a phase lead thus increasing stability margins
- to exploit this, τ_d should be such that $\tau_d \omega_c \not\ll 1$.

Outline

Designing PID's

Historically, the design of PID controllers is dubbed their tuning. There are roughly two philosophies here:

- explicit model-based: starts with a process model, often of the form

$$P(s) = \frac{k}{\tau s + 1}\,\mathrm{e}^{-sh} \quad \text{or} \quad P(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)}\,\mathrm{e}^{-sh},$$

  and then chooses k_p, τ_i, and τ_d according to whatever algorithm;

- implicit model-based: chooses k_p, τ_i, and τ_d directly from outcomes of a (simple) experiment with the plant according to whatever algorithm
  - although frequently regarded as model-free, all such experiments implicitly assume that plant dynamics are “simple,” say that the gain and phase are monotonically decaying functions of ω

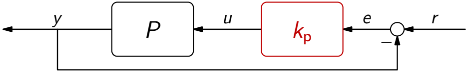
Ziegler-Nichols tuning

In 1942 John G. Ziegler & Nathaniel B. Nichols from Taylor Instrument Co. have published a paper with two methods of tuning PID controllers, based on

1. a closed-loop experiment with a proportional controller,
2. an open-loop step response experiment.

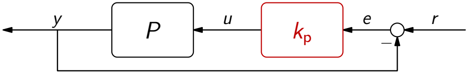
These methods were simple and efficient, soundly outperforming available solutions. They are still relevant today (although may no longer be widely used). We study the first of them below.

Ziegler-Nichols tuning: closed-loop experiment

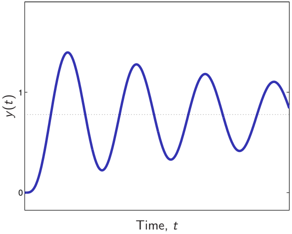


Increase k_p slowly until undamped oscillations arise in y:

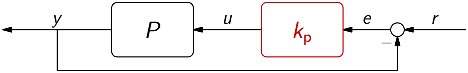
Ziegler-Nichols tuning: closed-loop experiment



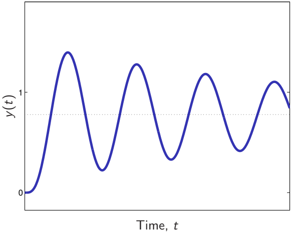
Increase k_p slowly until undamped oscillations arise in y:

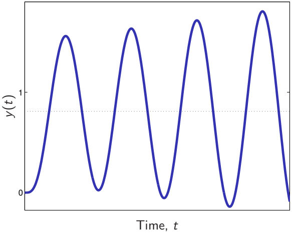


Ziegler-Nichols tuning: closed-loop experiment

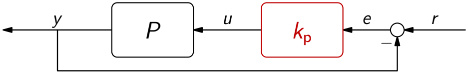


Increase k_p slowly until undamped oscillations arise in y:

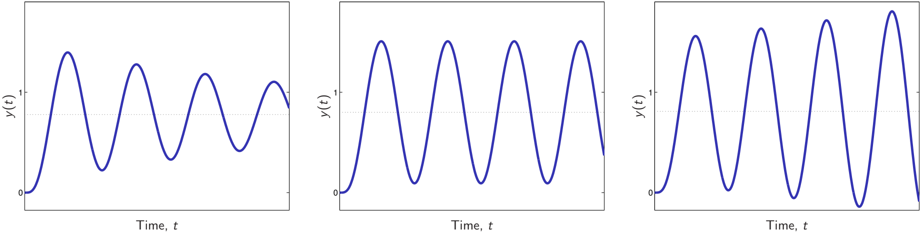




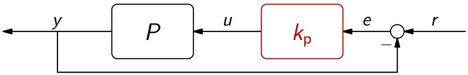
Ziegler-Nichols tuning: closed-loop experiment



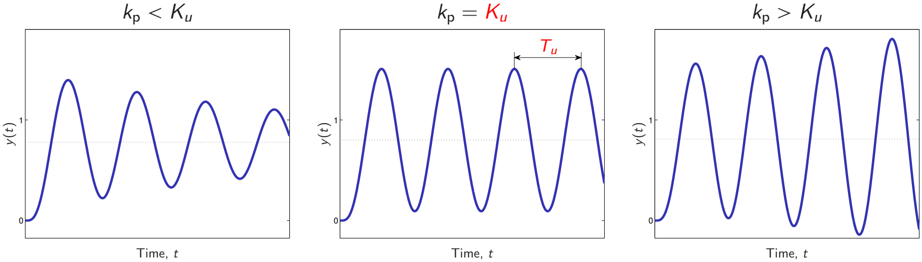
Increase k_p slowly until undamped oscillations arise in y:



Ziegler-Nichols tuning: closed-loop experiment



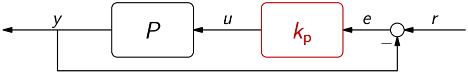
Increase k_p slowly until undamped oscillations arise in y:



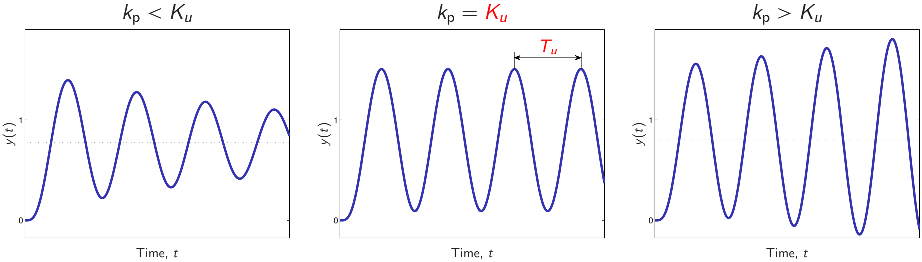
Denote the gain for which this happens as K_u and the oscillation period in steady state as T_u.

Remark: As a matter of fact, $T_u = 2\pi/\omega_\phi$ and $K_u = \mu_g = 1/|P(j\omega_\phi)|$.

Ziegler-Nichols tuning: closed-loop experiment



Increase k_p slowly until undamped oscillations arise in y:



Denote the gain for which this happens as K_u and the oscillation period in steady state as T_u.

Remark: As a matter of fact, $T_u = 2\pi/\omega_\phi$ and $K_u = \mu_\mathrm{g} = 1/|P(\mathrm{j}\omega_\phi)|$.

Ziegler-Nichols tuning: rules

	$k_{\rm p}$	$\tau_{\rm i}$	$\tau_{\rm d}$	
P	$0.5\,K_u$			renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.5\,\mathrm{e}^{-\mathrm{j}\pi}$ (thus, $\mu_{\rm g} = 2$)
PI	$0.45\,K_u$	$\frac{T_u}{1.2}$		renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.45\,\mathrm{e}^{-\mathrm{j}1.06\pi}$ (11° phase lag)
PID	$0.6\,K_u$	$\frac{T_u}{2}$	$\frac{T_u}{8}$	renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.66\,\mathrm{e}^{-\mathrm{j}0.86\pi}$ (25° phase lead) double zero at $s = -4/T_u$
PID[1]	$0.7\,K_u$	$\frac{T_u}{2.5}$	$\frac{T_u}{6.7}$	renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.8\,\mathrm{e}^{-\mathrm{j}0.84\pi}$ (28° phase lead) two zeros at $s = (-3.35 \pm \mathrm{j}2.351)/T_u$ ($\zeta \approx 0.82$ and $\omega_{\rm n} \approx 4.1/T_u$)

These values are to be used

– only as a starting point.

Typically, some (nontrivial) manual tuning of controller parameters required.

[1] D. Pessen rule; apparently, ZN rule affected by no longer existent hardware limitations.

Ziegler-Nichols tuning: rules

	k_p	τ_i	τ_d	
P	$0.5\,K_u$			renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.5\,\mathrm{e}^{-\mathrm{j}\pi}$ (thus, $\mu_\mathrm{g}=2$)
PI	$0.45\,K_u$	$\frac{T_u}{1.2}$		renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.45\,\mathrm{e}^{-\mathrm{j}1.06\pi}$ (11° phase lag)
PID	$0.6\,K_u$	$\frac{T_u}{2}$	$\frac{T_u}{8}$	renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.66\,\mathrm{e}^{-\mathrm{j}0.86\pi}$ (25° phase lead) double zero at $s=-4/T_u$
PID[1]	$0.7\,K_u$	$\frac{T_u}{2.5}$	$\frac{T_u}{6.7}$	renders $L(\mathrm{j}\frac{2\pi}{T_u}) = 0.8\,\mathrm{e}^{-\mathrm{j}0.84\pi}$ (28° phase lead) two zeros at $s=(-3.35\pm\mathrm{j}2.351)/T_u$ ($\zeta\approx 0.82$ and $\omega_\mathrm{n}\approx 4.1/T_u$)

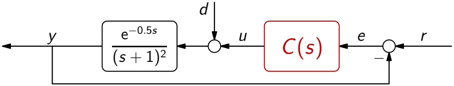
These values are to be used

– only as a starting point.

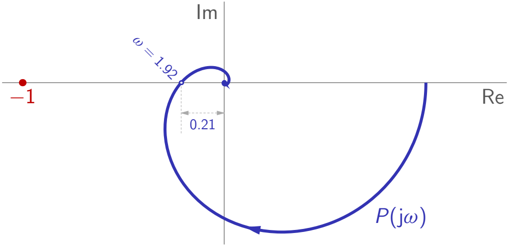
Typically, some (nontrivial) manual tuning of controller parameters required.

[1] D. Pessen rule; apparently, ZN rule affected by no longer existent hardware limitations.

Ziegler-Nichols tuning: example



Polar plot of the plant is:



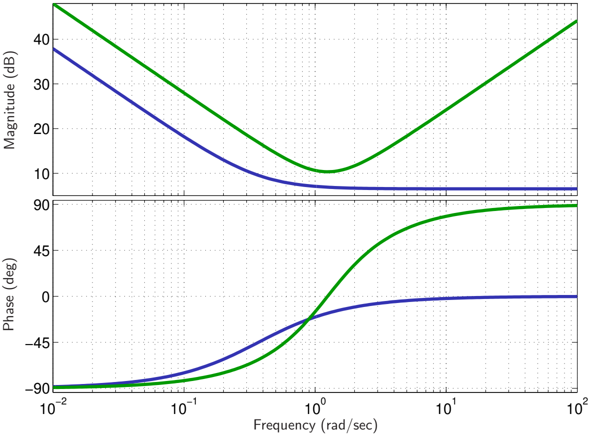
so we get: $K_u \approx 4.688$ and $T_u \approx 3.27$.

Ziegler-Nichols tuning: example (contd)

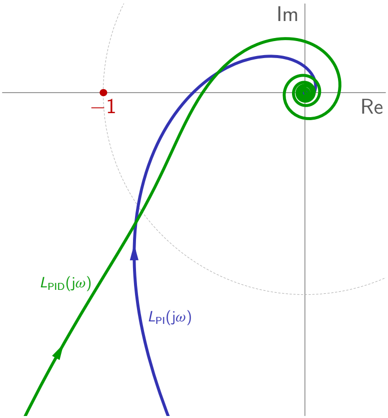
PI controller: $C_{\mathrm{PI}}(s) = 2.11\left(1 + \frac{1}{2.73s}\right)$

PID controller: PID controller: $C_{\mathrm{PID}}(s) = 3.28\left(1 + \frac{1}{1.31s} + 0.49s\right)$

Bode diagrams of C_{PI} and C_{PID}



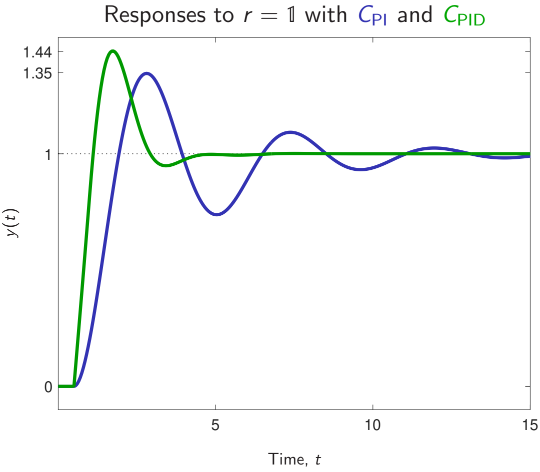
Ziegler-Nichols tuning: example (contd)



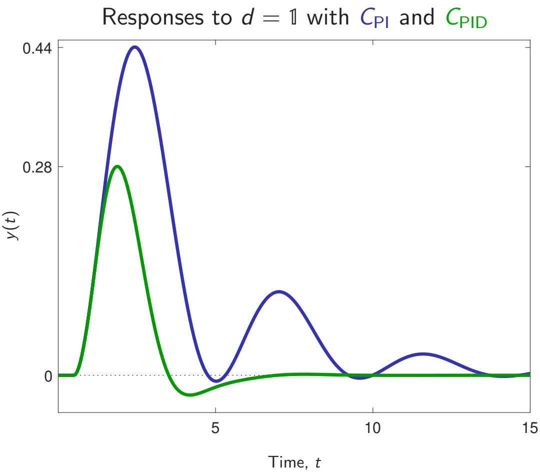
PI controller: $\omega_c = 1.11$, $\mu_g = 1.79$, $\mu_{ph} = 33.7^\circ$

PID controller: $\omega_c = 1.55$, $\mu_g = 1.97$, $\mu_{ph} = 36.3^\circ$

Ziegler-Nichols tuning: example (contd)



Ziegler-Nichols tuning: example (contd)

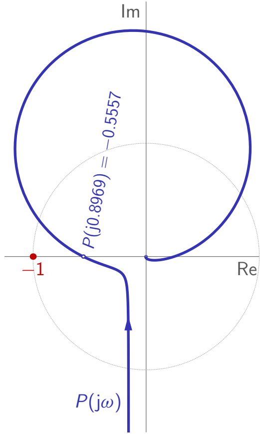


Ziegler-Nichols tuning *cum grano salis*

Let

$$P(s) = \frac{0.1e^{-1.5s}}{s(s^2 + 0.05s + 1)}$$

with



It has

$$\omega_\phi \approx 0.897 \quad \text{and} \quad |P(j\omega_\phi)| \approx 0.556,$$

so

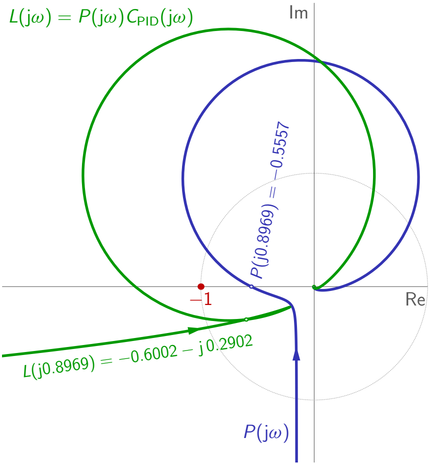
$$K_u \approx 1.8 \quad \text{and} \quad T_u \approx 7.$$

Hence,

$$C_{\text{PID}}(s) = 1.08\left(1 + \frac{1}{3.5s} + 0.875s\right) = \frac{0.31(1.75\,s + 1)^2}{s}.$$

Ziegler-Nichols tuning *cum grano salis* (contd)

The loop is then



which yields an unstable closed-loop system . . .

Where is it now?

Explicit model-based approaches seem to be dominant nowadays. Arguably, the most widely used plant model is the so-called "first-order+delay," like

$$P(s) = \frac{k}{\tau s + 1}\,\mathrm{e}^{-sh} \quad \text{or} \quad P(s) = \frac{k}{s}\,\mathrm{e}^{-sh}$$

for some $\tau > 0$. In some cases, when more inertial systems are considered, "second-order+delay" models of the form

$$P(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)}\,\mathrm{e}^{-sh} \quad \text{or} \quad P(s) = \frac{k}{s(\tau s + 1)}\,\mathrm{e}^{-sh}$$

may be picked.

Where is it now?

Explicit model-based approaches seem to be dominant nowadays. Arguably, the most widely used plant model is the so-called "first-order+delay," like

$$P(s) = \frac{k}{\tau s + 1}\,\mathrm{e}^{-sh} \quad \text{or} \quad P(s) = \frac{k}{s}\,\mathrm{e}^{-sh}$$

for some $\tau > 0$. In some cases, when more inertial systems are considered, "second-order+delay" models of the form

$$P(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)}\,\mathrm{e}^{-sh} \quad \text{or} \quad P(s) = \frac{k}{s(\tau s + 1)}\,\mathrm{e}^{-sh}$$

may be picked.

Where is it now? (contd)

Typical course of action:

1. identify parameters by fitting the model to system response
   - experiments may be carried out in open-loop or in closed-loop settings
   - may be based on step or frequency response characteristics

2. tune k_p, τ_i, τ_d to result in a "best" response for a given model
   - some methods are heuristic lookup tables

     e.g. the so-called SIMC rule sets

     $$k_\mathrm{p} = \frac{1}{k}\frac{\tau}{h+\tau_\mathrm{c}} \quad \text{and} \quad \tau_\mathrm{i} = \min\{\tau, 4(h+\tau_\mathrm{c})\},$$

     where τ_c is a tuning parameter (the closed-loop dominant time constant)
   - others use advanced optimization techniques to tune PID parameters (we may also employ brute force parametric search to minimize whatever cost function . . .)

First-order+delay richness: example

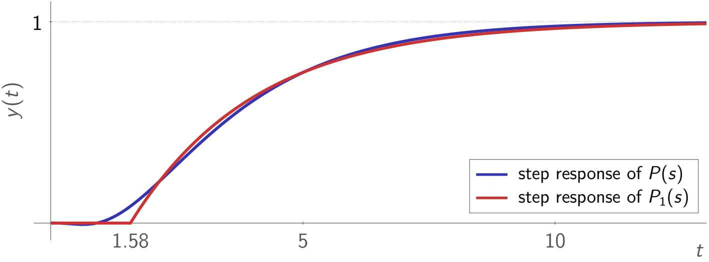
Let the "actual"

$$P(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}.$$

It may be approximated by

$$P_1(s) = \frac{1}{2.48s+1}\,\mathrm{e}^{-1.58s}$$

reasonably well:



Second-order+delay richness: example

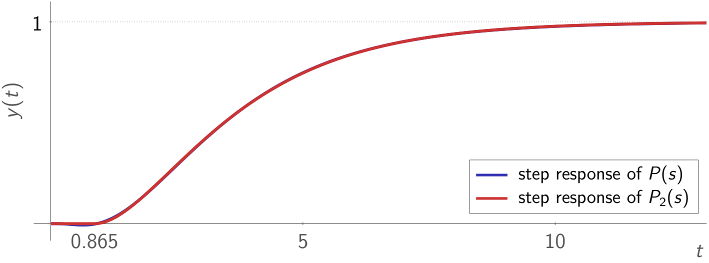
Let the “actual”

$$P(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}.$$

Its approximation by

$$P_2(s) = \frac{1}{(1.19s+1)(1.91s+1)}\,\mathrm{e}^{-0.865s}$$

is even better:



Outline

Industrial (PID) controllers

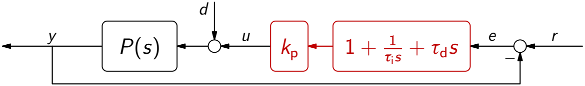
Tuning PID controllers

PID controller architectures and implementation

2-degrees-of-freedom controller configuration

Concluding remarks

"Natural" architecture



with the control signal

$$u(t) = k_\mathrm{p}\Big(e(t) + \frac{1}{\tau_\mathrm{i}}\int_0^t e(\theta)\mathrm{d}\theta + \tau_\mathrm{d}\dot{e}(t)\Big).$$

The transfer function of the controller,

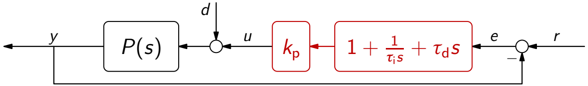
$$C(s) = k_\mathrm{p}\Big(1 + \frac{1}{\tau_\mathrm{i}s} + \tau_\mathrm{d}s\Big) = \frac{k_\mathrm{p}(\tau_\mathrm{d}\tau_\mathrm{i}s^2 + \tau_\mathrm{i}s + 1)}{\tau_\mathrm{i}s},$$

has zeros. It can be verified that unless canceled by plant poles,

– controller zeros are zeros of the transfer function $r \mapsto y$, i.e. of $T(s)$.

This might be problematic (zeros, especially dominant, complicate analysis and might contribute to increased overshoot).

"Natural" architecture



with the control signal

$$u(t) = k_p\Big(e(t) + \frac{1}{\tau_i}\int_0^t e(\theta)\mathrm{d}\theta + \tau_d \dot{e}(t)\Big).$$

The transfer function of the controller,

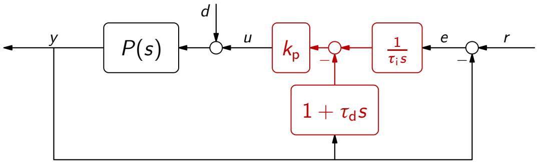
$$C(s) = k_p\Big(1 + \frac{1}{\tau_i s} + \tau_d s\Big) = \frac{k_p(\tau_d\tau_i s^2 + \tau_i s + 1)}{\tau_i s},$$

has zeros. It can be verified that unless canceled by plant poles,

- controller zeros are zeros of the transfer function $r \mapsto y$, i.e. of $T(s)$.

This might be problematic (zeros, especially dominant, complicate analysis and might contribute to increased overshoot).

Alternative architecture



with the control signal

$$u(t) = k_{\mathrm{p}}\Big(-y(t) + \frac{1}{\tau_{\mathrm{i}}}\int_0^t e(\theta)\mathrm{d}\theta - \tau_{\mathrm{d}}\dot{y}(t)\Big).$$

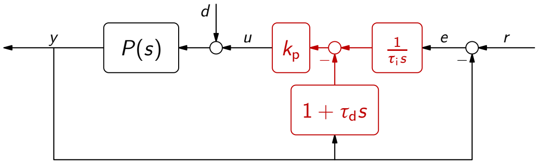
Now the transfer function of the system $r \mapsto y$,

$$T_c yr(s) = \frac{P(s)\,k_{\mathrm{p}}/(\tau_{\mathrm{i}}s)}{1 + P(s)C(s)},$$

has only zeros of the plant as its zeros, which might simplify matters. Note,

— disturbance response is not affected by this change of the architecture.

Alternative architecture



with the control signal

$$u(t) = k_{\mathrm{p}}\Bigl(-y(t) + \frac{1}{\tau_{\mathrm{i}}}\int_0^t e(\theta)\mathrm{d}\theta - \tau_{\mathrm{d}}\dot{y}(t)\Bigr).$$

Now the transfer function of the system $r \mapsto y$,

$$T_{\mathrm{c}}yr(s) = \frac{P(s)\,k_{\mathrm{p}}/(\tau_{\mathrm{i}}s)}{1 + P(s)C(s)},$$

has only zeros of the plant as its zeros, which might simplify matters. Note,

– disturbance response is not affected by this change of the architecture.

Alternative architecture: example

For example, let

$$P(s) = \frac{k}{s+a} \quad \text{and} \quad C(s) = k_p\Big(1 + \frac{1}{\tau_i s}\Big)$$

Then

$$T(s) = \frac{kk_p(\tau_i s + 1)}{\tau_i s^2 + \tau_i(a + kk_p)s + kk_p}$$

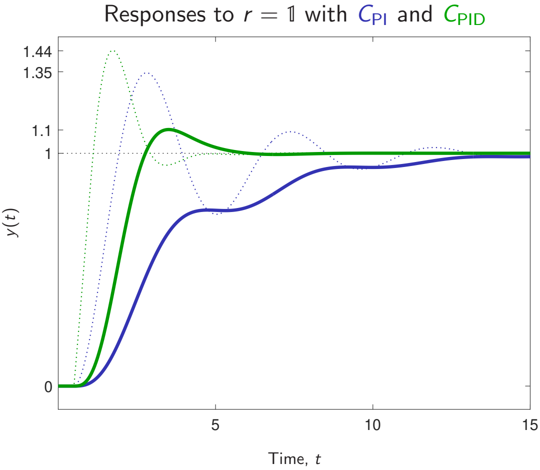
has a zero at $s = -1/\tau_i$ (unless $\tau_i = 1/a$). At the same time,

$$T_c yr(s) = \frac{kk_p}{\tau_i s^2 + \tau_i(a + kk_p)s + kk_p}$$

is a 2-order transfer function without zeros (hence, easier to understand).

Ziegler-Nichols tuning: example (contd)

The same tuning, but a different architecture:



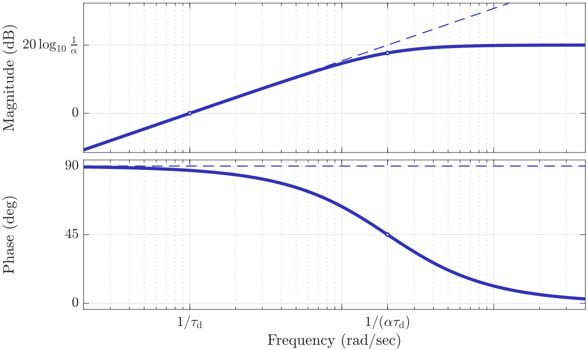
The disturbance response remains the same.

Implementing the D part

The non-proper $\tau_d s$ is normally implemented as

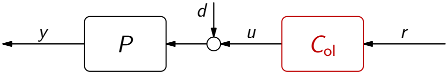
$$\frac{\tau_d s}{\alpha \tau_d s + 1}$$

with sufficiently small α (typically, $0.05 \le \alpha \le 0.3$):



Outline

Open-loop control: architecture



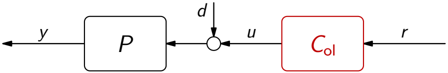
Signals of interest:

$$\boxed{y = PC_{ol}r + Pd} \quad \& \quad \boxed{u = C_{ol}r},$$

where

- r is a reference signal (requirements)
- d is a load disturbance

Open-loop control: strategy



Plant inversion with reference model:

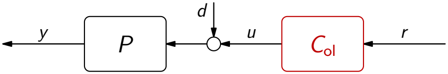
$$C = C_{\mathrm{ol}} = P^{-1} T_{\mathrm{ref}},$$

where stable T_{ref} should endeavor to have . . .

steady-state $|1 - T_{\mathrm{ref}}(\mathrm{j}\omega)| \ll 1$ at ω's where the spectrum of r dominates

transients
- dominant poles of $T_{\mathrm{ref}}(s)$ are in "good" regions
- sufficiently wide, but not too wide, bandwidth of $T_{\mathrm{ref}}(\mathrm{j}\omega)$
- no high resonant peaks of $|T_{\mathrm{ref}}(\mathrm{j}\omega)|$

Open-loop control: properties



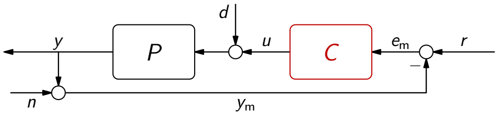
Open-loop architecture is

- ⌣̈ efficient in handling command following requirements
- ⌣̈ technically simple — both T_{ref} and C_{ol} are stable
  - all nonminimum-phase zeros of $P(s)$ as zeros of $T_{\mathrm{ref}}(s)$
  - pole excess of $T_{\mathrm{ref}}(s) \geq$ poles excess of $P(s)$ — unless $\dot{r}$, $\ddot{r}$, etc measurable

- ⌢̈ does not help in handling uncertainty
  - modeling inaccuracies in P
  - disturbances

- ⌢̈ inapplicable if the plant P is unstable

Closed-loop control: architecture (unity feedback)



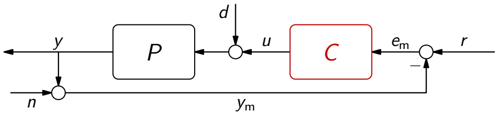
Signals of interest:

$$\boxed{y = Tr + T_d d - Tn} \quad \& \quad \boxed{u = T_c r - Td - T_c n},$$

where

- $T(s) = \dfrac{P(s)C(s)}{1 + P(s)C(s)}$ complementary sensitivity
- $T_d(s) = \dfrac{P(s)}{1 + P(s)C(s)}$ disturbance sensitivity
- $T_c(s) = \dfrac{C(s)}{1 + P(s)C(s)}$ control sensitivity
- $S(s) = 1 - T(s)$ sensitivity

Closed-loop control: strategy



Internally stabilizing C should endeavor to have . . .

Command following:

- $|1 - T(j\omega)| \ll 1$ at ω's where the spectrum of r dominates
- $T(j\omega)$ has sufficiently wide, but not too wide, bandwidth
- $T(j\omega)$ has no high resonance peaks

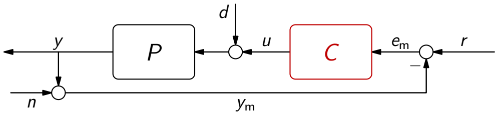
Disturbance attenuation:

- $|T_d(j\omega)| \ll 1$ at ω's where the spectrum of d dominates

Noise sensitivity:

- $|T(j\omega)| \ll 1$ & $|T_c(j\omega)| \not\gg 1$ at ω's where the spectrum of n dominates

Closed-loop control: properties



Closed-loop architecture is

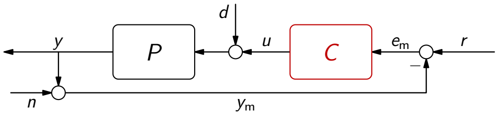
- efficient in stabilizing
- efficient in handling command following requirements
- efficient in attenuating disturbances

- technically nontrivial
  - high-low gain tradeoffs
  - crossover region acrobatics

partially because it has to address

- too many intrinsically conflicting goals.

Closed-loop control: properties



Closed-loop architecture is

- ☺ efficient in stabilizing
- ☺ efficient in handling command following requirements
- ☺ efficient in attenuating disturbances

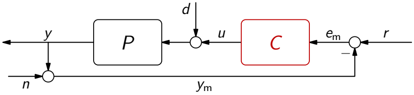
- ☹ technically nontrivial
  - high-low gain tradeoffs
  - crossover region acrobatics

partially because it has to address

- ☹ too many intrinsically conflicting goals.

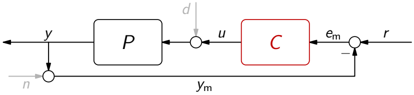
The best of both worlds

Handling uncertainty:

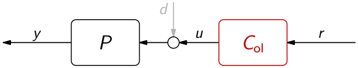


- no alternative to feedback
- no alternative architecture (C acts after n and before d)

Nominal command following:



vs.



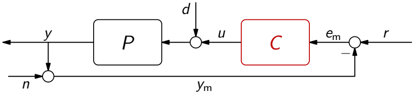
- no advantage of feedback
- open-loop design is simpler

Natural question:

- can we synergize open- and closed-loop architectures?

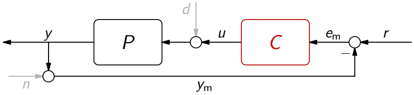
The best of both worlds

Handling uncertainty:

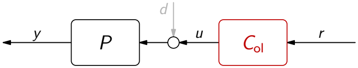


- no alternative to feedback
- no alternative architecture (C acts after n and before d)

Nominal command following:



vs.



- no advantage of feedback
- open-loop design is simpler

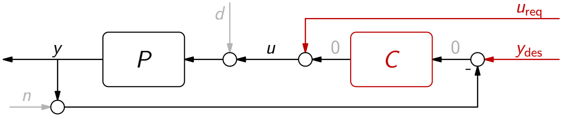
Natural question:

- can we synergize open- and closed-loop architectures?

Circumvent C in the nominal case

The idea is to

– negate the effect of C if everything behaves expectably.



Expectable behavior is expressed as two requirements to added signals:

1. $y_{\rm des}$ and $u_{\rm req}$ are bounded stability
2. $y_{\rm des} = Pu_{\rm req}$ consistency

In this case (remember, $T_cP = T$ and $S + T = 1$)

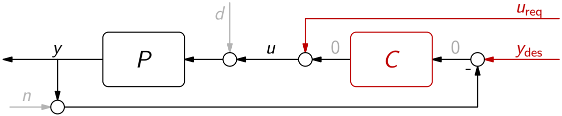
$$u = u_{\rm req} + C(y_{\rm des} - Pu) \quad \Longleftrightarrow \quad u = Su_{\rm req} + T_c y_{\rm des} = u_{\rm req}$$

and $y = Pu = y_{\rm des}$, *regardless* C (provided it is stabilizing, of course).

Circumvent C in the nominal case

The idea is to

– negate the effect of C if everything behaves expectably.



Expectable behavior is expressed as two requirements to added signals:

1. $y_{\rm des}$ and $u_{\rm req}$ are bounded stability
2. $y_{\rm des} = Pu_{\rm req}$ consistency

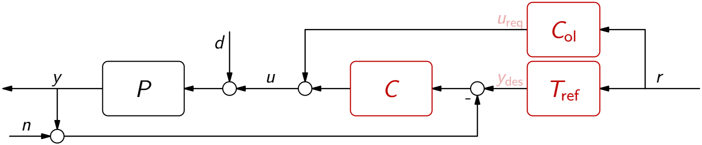
In this case (remember, $T_{\rm c}P = T$ and $S + T = 1$)

$$u = u_{\rm req} + C(y_{\rm des} - Pu) \quad \iff \quad u = Su_{\rm req} + T_{\rm c}y_{\rm des} = u_{\rm req}$$

and $y = Pu = y_{\rm des}$, *regardless* C (provided it is stabilizing, of course).

2DOF architecture (when a reference model is used)

If we take $y_{\mathrm{des}} = T_{\mathrm{ref}}\, r$ and $u_{\mathrm{req}} = C_{\mathrm{ol}}\, r$ for $C_{\mathrm{ol}} = P^{-1} T_{\mathrm{ref}}$ (consistent), then



and signals of interest are

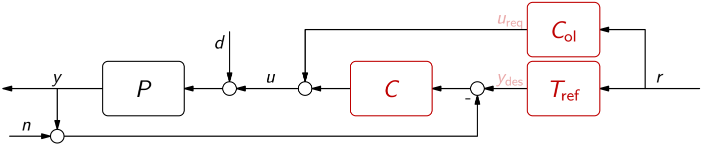
$$\boxed{y = T_{\mathrm{ref}}\, r + T_{\mathrm{d}}\, d - T n} \quad \& \quad \boxed{u = C_{\mathrm{ol}}\, r - T d - T_{\mathrm{c}}\, n},$$

This controller blends open- and closed-loop controller architectures, with a complete separation of

- nominal command response — shaped by C_{ol}
  standard open-loop design
- stabilization and handling uncertainty — shaped by C
  standard feedback design, just w/o taking the command response into account

2DOF architecture (when a reference model is used)

If we take $y_{\mathrm{des}} = T_{\mathrm{ref}} r$ and $u_{\mathrm{req}} = C_{\mathrm{ol}} r$ for $C_{\mathrm{ol}} = P^{-1} T_{\mathrm{ref}}$ (consistent), then



and signals of interest are

$$\boxed{y = T_{\mathrm{ref}} r + T_{\mathrm{d}} d - T n} \quad \& \quad \boxed{u = C_{\mathrm{ol}} r - T d - T_{\mathrm{c}} n},$$

This controller blends open- and closed-loop controller architectures, with a complete separation of

- nominal command response — shaped by C_{ol}

  standard open-loop design
- stabilization and handling uncertainty — shaped by C

  standard feedback design, just w/o taking the command response into account

Outline

Summary

Learned in this course,

- flavor of basic control ideas,

so you could communicate with control engineers in their native language.

A long and winding road to become a control engineer / R&D yourselves,

035188 Control Theory

035036 Control Systems Design

034406 Advanced Control Laboratory

036709 Sampled-Data Control Systems

036012 Linear Control Systems

036050 Nonlinear Control Systems

036013 Process Optimization

⋮

Summary

Learned in this course,

- flavor of basic control ideas,

so you could communicate with control engineers in their native language.

A long and winding road to become a control engineer / R&D yourselves,

035188 Control Theory

035036 Control Systems Design

034406 Advanced Control Laboratory

036709 Sampled-Data Control Systems

036012 Linear Control Systems

036050 Nonlinear Control Systems

036013 Process Optimization

⋮