Model structuring and tuning brush-up

several controller structure
will be used... standard techniques
to tune controllers

- Modelling framework definitions
- A bit of practice
- Brush-up (and some complements) on controller tuning

tuning capabilities

MODELLICA
libraries > coursework > Tuning-brushup
Modelling-principles

code used during the lecture 3

Block-oriented and object-oriented (BO/OO) models

- As said we stick to 1st-principle models (based on dynamic balances) since we may need to analyse/simulate/optimise something that does not yet exist (thus, no data to identify e.g. black- or grey-box models).
- This does *not* mean that identification and estimation never come into play – think e.g. of adaptive control – but rather just that the matter does not fit in this course (except for a few words later on). ↳ use MIDA to tune controllers → here we wanna structure the scheme of control
- We need however to distinguish between block- and object-oriented models.
  - Block-oriented models: they are
    - are oriented or causal, → written with known boundaries condition, in mind
    - thus written having their boundary conditions in mind, ↲ (simulink I/O syst)
    - and connect to one another via inputs and outputs. [result of evolution of the syst.]
  - Object-oriented models
    - are non oriented or a-causal,
    - thus written independently of their boundary conditions,
    - and connect to one another via ports. comunicate by ports with one another
- Let us go through an introductory example, and then generalise.

BO (causal) and OO (a-causal) models

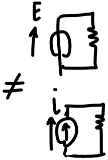
Important!

Introductory example

different models!

suppose model

- We need to model a resistor $\Rightarrow$ Ohm's law.
- We consider two cases:



1. the resistor is connected to a fixed voltage generator E, leading with obvious notation to the model fixing voltage

$$\begin{cases} V = E & \text{Voltage generator (boundary condition for the resistor)} \\ I = V/R & \text{Resistor} \end{cases}$$

2. the resistor is connected to a fixed current generator A, leading this time to

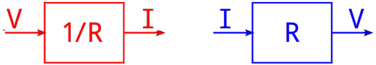
$$\begin{cases} I = A & \text{Current generator (boundary condition for the resistor)} \\ V = RI & \text{Resistor} \end{cases}$$

fixing current

depending on Resistor role inside, change the model usage

- Same component, different boundary conditions, different models,
- both oriented: in the former case V is an input and I an output, in the latter *vice versa*; the two BO resistor models are

the model of R role inside the equation NOT R model!



model used depending on R usage

with *input* and *output* connectors.

different model !

$\Downarrow$

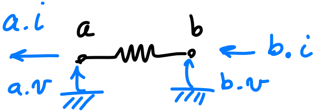
model of Resistor
independently from boundaries condition ? $\Rightarrow$ NO depends on our focus

every boundaries condition
calls for a different model
of the component!!

BO (causal) and OO (a-causal) models

Introductory example – a perspective change

phisical object



Different approach

- Now we take a different approach:
  - we identify *ports*, i.e., physical terminals characterising the interface exposed by the modelled component to the outside. In this case ports are the resistor's two pins (a and b to name them) each of which carries a voltage V and a current I;
  - we write the component's *constitutive equations*, that with obvious notation (current is taken positive when entering the pin) read

$$RES : \begin{cases} a.I + b.I &= 0 \\ a.V - b.V &= R\,a.I \end{cases}$$

  - we can do the same for the voltage and the current generators, obtaining (mind the current sign convention)

VOLTAGE Presubed　　CURRENT presuched

4 eq in 2 variables

$$VGEN : \begin{cases} a.I + b.I &= 0 \\ a.V - b.V &= E \end{cases} \qquad CGEN : \begin{cases} a.I + b.I &= 0 \\ a.I &= -A \end{cases}$$

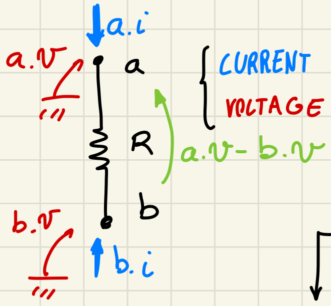
2 eq 4 var.

  - we can finally introduce a ground (with a single pin a), i.e.,

$$GND : a.V = 0.$$

1 eq, 2 var!

• **4 VARIABLES** $\Rightarrow \{ a.v \,, a.i \,, b.v \,, b.i \}$ (unconditionally true independently from connections)



⇑
physical object!
RESISTOR

ALWAYS :

(no electron losses)

$$\begin{cases} a.i + b.i = 0 \\ \underbrace{a.v - b.v} = R * a.i \end{cases}$$

role of R is to fix that relationship!

⇑
2 equations

(NOT closed) model... causality of model depends on other var...

{less eq than var}

{this eq is True unconditionally}

BO (causal) and OO (a-causal) models

Introductory example – exploiting OO modelling

- Doing so, the two addressed cases only differ for the generator equations:

Constitutive equations

$$\begin{aligned} RES.a.I + RES.b.I &= 0 \\ RES.a.V - RES.b.V &= RES.R * RES.a.I \end{aligned} \quad \text{res}$$

$$\begin{aligned} GEN.a.I + GEN.b.I &= 0 \\ GEN.a.V - GEN.b.V &= GEN.E \end{aligned} \quad \text{V gen} \qquad \begin{aligned} GEN.a.I + GEN.b.I &= 0 \\ GEN.a.I &= -GEN.A \end{aligned} \quad \text{C gen}$$

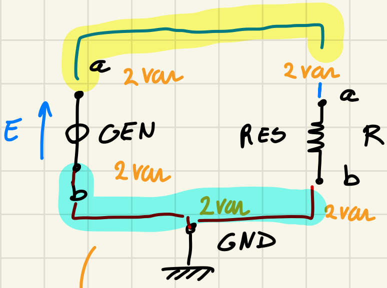
$$GND.a.V = 0 \quad \text{Gnd}$$

Connection equations

$$\begin{aligned} GEN.a.V &= RES.a.V \\ GEN.a.I + RES.a.I &= 0 \end{aligned} \quad \text{pins GEN.a and RES.a}$$

$$\begin{aligned} GEN.b.V &= GND.a.V \\ RES.b.V &= GND.a.V \\ GEN.b.I + RES.b.I + GND.a.I &= 0 \end{aligned} \quad \text{pins GEN.b, RES.b and GND.a}$$

taking more components ... GND, RES, GEN



constitutive equations

component name

$$
\begin{aligned}
RES.a.I + RES.b.I &= 0 && \text{res}\\
RES.a.V - RES.b.V &= RES.R * RES.a.I &&\\
GEN.a.I + GEN.b.I &= 0 && \text{V}\\
GEN.a.V - GEN.b.V &= GEN.E && \text{gen}\\
GND.a.V &= 0 && \text{Gnd}
\end{aligned}
$$

10 variables, 5 constitutive equations

"name component . pin . variable" (unambiguous var.name)
to access in univoque way a var.

⇓

IF I say:
CONNECTIONS:

connect (GEN.a, RES.a)
connect (GEN.b, GND.a)
connect (GND.a, RES.b)

connect statements

(eq. generated by connections)

connection sets { , }

⇓

2 more eq (without new var)

{ GEN.a.v = RES.a.v
GEN.a.i + RES.a.i = 0

{ GEN.b.v = GND.a.v
GND.a.v = RES.b.v

set
• equal voltage eq

Flow equation → [GEN.b.i + RES.b.i + GND.a.i = 0

|| 5 connection eq. ||

Pin Definition

a PIN contain:

- voltage v
- Flow current i

↳ {Modellica language} :

```
connector Pin
        Voltage v ;
   Flow current i ;
end pin;
```

all NON FLOW variables with same name are set to be equals while all Flow var are set to $\sum = 0$

attribute "Flow" to current...
when connecting 2 or more connector toghether

- 5 constitutive eq
- 5 connect eq

in 10 var → model closed !

even with different connections...
I don't need to change equations,
JUST one model general !

BO (causal) and OO (a-causal) models

phisic relationship to be highlighted...

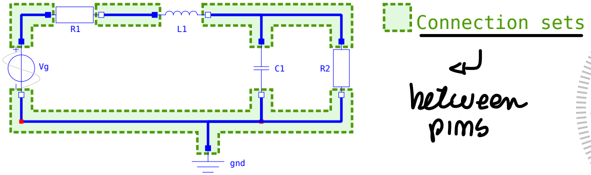
Introductory example – wrap-up

identify rate, Temp inside Thermal component: physical relationship between ports

- Summing up and abstracting:
  - component models are written independently of their connections;
  - they are neither oriented nor closed (having less equations than variables);
  - they interconnect via *ports*;
  - and the overall model is closed (thereby determining orientation) by joining

close model

- the (component-specific) *constitutive equations*
- and the (port-specific) *connection equations*,
- which happens connecting ports to form *connection sets*, as exemplified below.



between pins

BO (causal) and OO (a-causal) models

Introductory example – wrap-up

Joining mech shaft
same ϑ, $\Sigma_i F = 0$

- Ports (or connectors) carry variables, that can be
  - *effort* variables (defined as difference with respect to a reference, think of voltage or temperature), position...
  - or *flow* variables (defined as flowing through a boundary, think of current or power). → flow rate, force, torque
- Connecting N (two or more) ports into a *connection set* generates
  - $N-1$ equations per effort variable, stating that it is equal on all the connected ports,
  - and one equation per flow variable, stating that its sum over all connected ports is zero.

$(N-1)$
$+$
1 $\forall$ flow var

- Note: all of the above has a direct counterpart in OO modelling languages such as Modelica (more on this in due course).

"connect" : to specify this

BO and OO modelling

Distinctive features of both approaches

- Both types of models allow to *encapsulate* the model's behaviour with respect to its interface, hence permitting e.g. to scale the representation detail.
- BO models require the system to be oriented $\Rightarrow$ suitable for control components (block diagram elements) and *complete* controlled systems' models, i.e., for "plants completely built, with control signals and controlled variables already specified as inputs and outputs. control element → oriented system by definition!
- OO models do not require the system to be oriented $\Rightarrow$ suitable for individual plant components.
- Quite intuitively, we shall use a combination of the two model types.

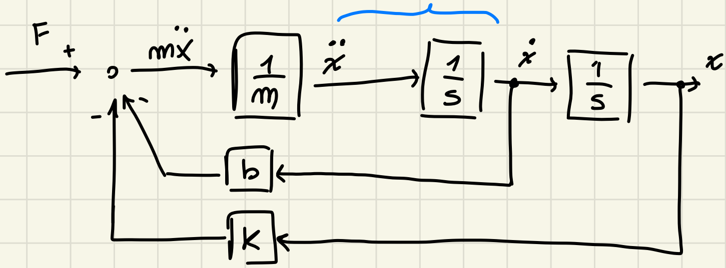
(Different things!)

ORIENTATION vs COMPUTATION FLOW

$$\left\{ m\ddot{x} = F - kx - h\dot{x} \right\} \text{ mech model}$$

$\Downarrow$

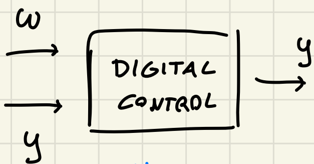
{ don't receive
compute and produces ... } → equation!



no a before / after ↑ JUST equations

(equation)

different meaning!



(Algoritms)

there is a certain workflow orientation

{ receive, compute, produce }

BO and OO modelling

Distinctive features of OO models

T: temperature is NOT a phisical quantity but a STATE index! (no phisical quantity)

- In OO models, ports are naturally associated with energy transfer.
- Consider the typical port with one effort and one flow variable: most such couples are keen to be related to power.

example

- Examples: (PIN)

  Effort var Flow var

  derivative in the loop

  - electric, obviously (voltage v, current i) $\Rightarrow vi =$ power;
  - mechanic, traslational (position x, force f) $\Rightarrow \dot{x}f =$ power, (note the derivative;)
  - mechanic, rotational (angle φ, torque τ) $\Rightarrow \dot{\varphi}\tau =$ power, note again the derivative;
  - thermal, conductive and convective (temperature T, power Q): note that here power is *not* the product of the two. state index NOT phisical quantity!
- This OO characteristic allows to create electric equivalent models, particularly when the flow variable is linearly related to a difference in the effort one, and to make such models modularly composable.

helps to solve

- Electric equivalents are often useful for a rapid interpretation of an interconnected system (we shall see some examples).

study electrical circuit to study our syst.

A multi-physics model

(electrical, thermal, and analogue control domains)

MODELLICA (Libraries > Tuning-brushup)

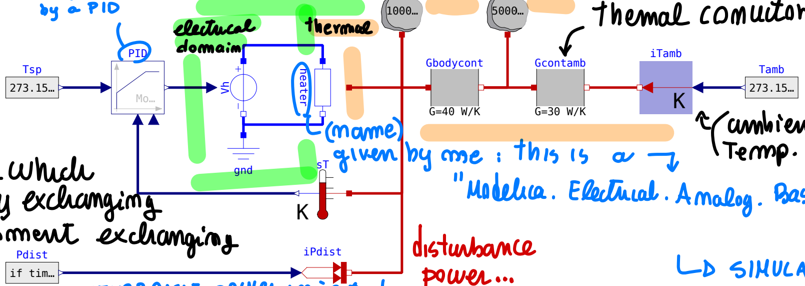
(OBJ ORIENTED)

elements of modellica STANDARD Library

controls Body Temp by a PID

[multi-phisic model]

{heat capacitor} containment



electrical domain

thermal

thermal conductor

(name) given by me : this is a "Modelica. Electrical. Analog. Basic. Resistor" obj

(ambient Temp.)

model of electric heater which heats a body exchanging with containment exchanging with ambient

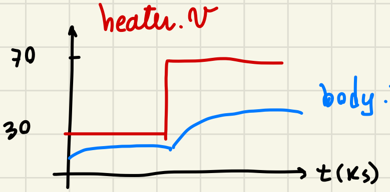
DISTURBANCE power injected to body

disturbance power...

↳ SIMULATION

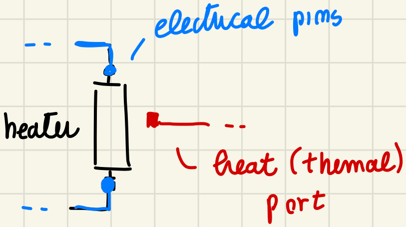
- Entirely assembled with components from the Modelica Standard Library (MSL for short).
- Observe the co-existence of BO and OO components.
- Check out `https://modelica.org/libraries`.

⇒ IF simulating on Modelica



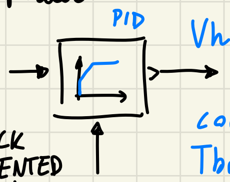
ecc.. all variable of interest plotted!

⇒ multiphisics model



With a PID controller which controls Body Temp acting on Resistor power

OBJECT ORIENTED (with connections, NOT I/O)

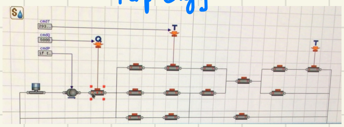


BLOCK ORIENTED model (I/O)

controls Tbody acting on power

⇓ we will see more complex model... with thousands of state var... where you can only see 2190 equations...

Topology



network of pipes

⇓

to model more complex system thermal network
↓ 2K equations!

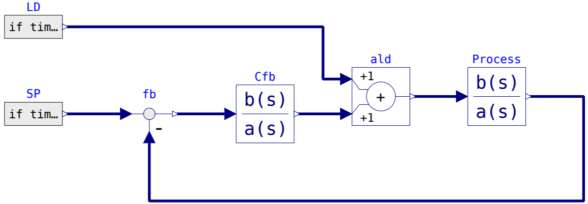
Block oriented model → NOT suitable for large scale system models
↓
for large scale syst ... Obj oriented!

↙ hard to model in simulink

depending on my scope.. chose model

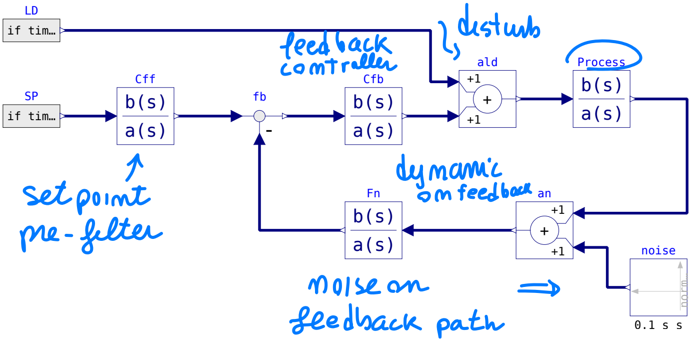
A basic control loop

other examples...



- Components from the MSL, package `Blocks`.

A more articulated control loop



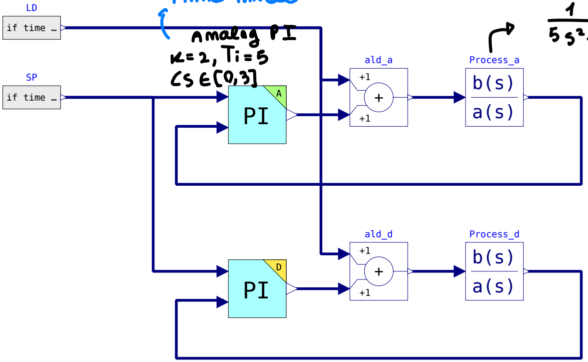
- Two-degree-of-freedom (2-dof) controller (more on this later).
- Measurement noise and analogue filter in the feedback path.

A simple PI control loop

with analogue and digital controller representation

Comparison between analog Digital controller

continuous time model



Analog PI
$K=2,\ Ti=5$
$Cs \in [0,3]$

$$\frac{1}{5s^2+6s+1}$$

(specify polinom. like matlab)

- Controllers from the course library.
- Play around with these models, create others, experiment, enjoy ☺

→ simulate on Modellica

with saturation on loop path

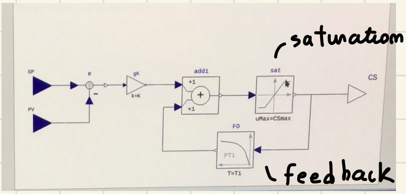
\+ digital representations by Algorithm

⇓

{ digital realization of system }

model

PI ANALOG



saturation path

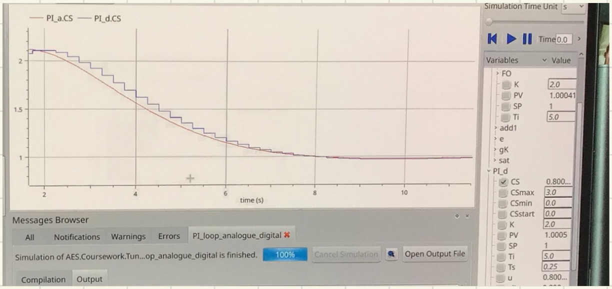
feedback

We use this blocks

PI DIGIDAL → algorithm ... sampling periodically ecc

simulate for $T_s = 0.25$

control signal → simulation on continuous time!



Foreword

Purpose of this treatise

Tuning brush-up

- In the sequel of the course we shall need to tune several controllers. NOT single loop
- In most cases, these will be of PI/PID type (albeit as part of larger schemes).
- We need "approximate but fast" procedures to do that.
- We therefore brush up the tuning matter with a deliberate operational focus.
- We also take the occasion of introducing some concepts (servo vs. regulatory tuning, 2-dof exploitation. *a priori* robustness quantification) to be used later on in the addressed control schemes...
- ...and of general validity also beyond the energy domain.

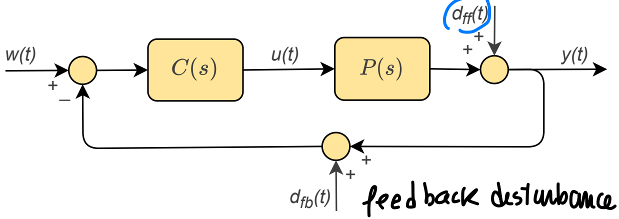
beyond energy systems

Foreword

The addressed *scenario* (in the Bode applicability hypotheses)

Loop

feed forward disturbance



feedback disturbance

- Loop $L(s)$, sensitivity $S(s)$, complementary sensitivity $T(s)$, control sensitivity $Q(s)$:

{loop T.F} $L(s) = \boxed{P(s)C(s)}$, ← adimensional! same IN/OUT dimension

↳ main T.F

$$S(s) = \boxed{\frac{1}{1+L(s)}} = \frac{Y(s)}{D_{ff}(s)}, \quad T(s) = \boxed{\frac{L(s)}{1+L(s)}} = \frac{Y(s)}{W(s)} = -\frac{Y(s)}{D_{fb}(s)},$$

$$Q(s) = \boxed{\frac{C(s)}{1+L(s)}} = \frac{U(s)}{W(s)} = -\frac{U(s)}{D_{ff}(s)} = -\frac{U(s)}{D_{fb}(s)}$$

↑ how control reacts!

$L(s)$ meaningfully ≥ 1 ...

$\uparrow$

loop T.F adimensional... same unit of measure on In/Out

$\Rightarrow$ so $L(s) \gtrless 1$ make sense

the same for $S(s)$ [adim], $T(s)$ [adim]

while $Q(s)$ dimensional! $\rightarrow$ IF $Q(s) > 1$

a change in unit of measure change all!

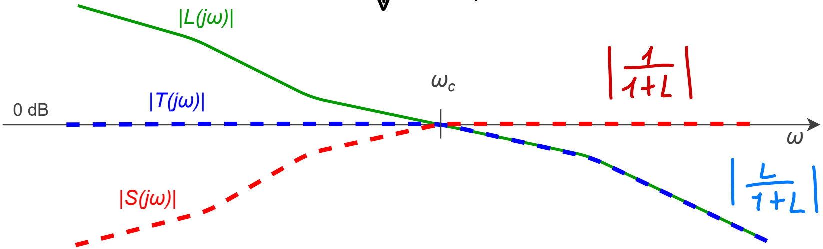
only $L(s)$ meaningfully $\gtrless 1$ $\leftarrow$

Foreword

Bode criterion applicability

The addressed *scenario* (in the Bode applicability hypotheses)

↓ T.F approx respect ω_c



- Approximations for the frequency responses of $S(s)$ and $T(s)$:

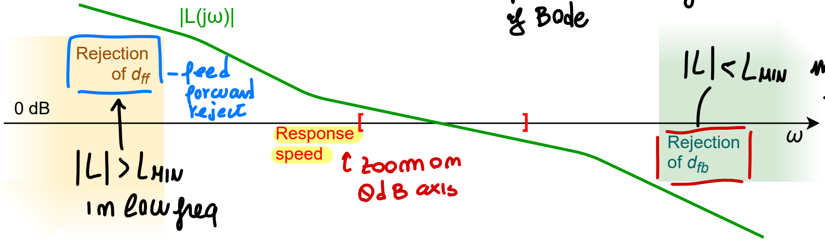
$$S(j\omega) \approx \begin{cases} 1/L(j\omega) & \omega < \omega_c \\ 1 & \omega > \omega_c \end{cases} \qquad T(j\omega) \approx \begin{cases} 1 & \omega < \omega_c \\ L(j\omega) & \omega > \omega_c \end{cases}$$

Foreword

The addressed *scenario* (in the Bode applicability hypotheses)

response speed constrains expressed through 0dB axis of Bode

⇓



- feed forward reject

|L| > L_MIN in low freq

↕ zoom on 0dB axis

|L| < L_MIN in high freq.

⇑ design, LOOP SHAPING

- Synthesis by loopshaping: determine type and/or gain of $C(s)$ based on steady-state requirements, set constraints as shown above, choose a suitable $L(s)$ avoiding critical cancellations and checking the phase margin, compute $C(s)$ dividing that $L(s)$ by $P(s)$. This, you know.
- We are looking for less powerful but simpler techniques.

given process, disturbance rejection, phase margin... you know how to draw Bode

Rapid PI/PID tuning for asymptotically stable overdamped processes

⇓ simpler tuning

(no oscillation..) / good approx if asymp. stable

- Idea: approximate the system to control with a 1st or 2nd-order one, possibly with delay (FOPDT/SOPDT, for First/Second Order Plus Dead Time).
- Check out the literature for the said models and PI/PID tuning; here we stick to the FOPDT structure

↳ FIRST ORDER PLUS DEAD TIME

$$\boxed{M(s) = \mu\frac{e^{-sD}}{1+sT},} \qquad T>0, D\geq 0.$$

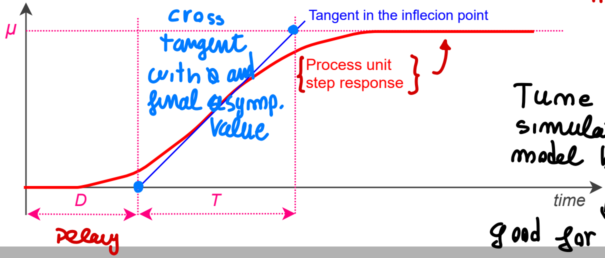
structure parameterized..

- For an overdamped system one can readily parametrise it from a step response. There are many methods for that, we see only the inflection point one:

method

plot step responce, take tangent, cross with 0 of value → TUNING!

OK for FIRST CUT



Tune the simulation model by IORD approx! ⇓ good for 1st cut

The simplest technique

Tune by cancellation

→ i see a step response like I ORD

dominantly I

(*) ⇒

- Dominantly first-order system, negligible delay ⇒ PI:

controller

$$M(s)=\frac{\mu}{1+sT},\ \text{desired } \omega_c \quad\Rightarrow\quad C(s)=\frac{\omega_c}{s}\frac{1+sT}{\mu}$$

→ $L(s)=\frac{\omega_c}{s}$

ω_c

↳ model

giving a phase margin $\varphi_m \approx 90°$ ($90°$ in nominal conditions, i.e., $M=P$).

- Dominantly second-order system, negligible delay ⇒ real PID:

2 zeros, 2 poles

$$M(s)=\frac{\mu}{(1+sT_1)(1+sT_2)},\ \text{desired } \omega_c \quad\Rightarrow\quad C(s)=\frac{\omega_c}{s}\frac{(1+sT_1)(1+sT_2)}{\mu\left(1+\frac{s}{\beta\omega_c}\right)}$$

2 time const model

I pole needed

giving in nominal conditions $\varphi_m = 90° - \arctan°(1/\beta)$.

pole on $-\beta\omega_c$, phase margin decreased

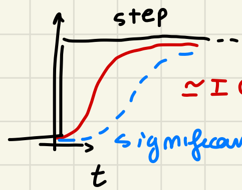
↓ depends on what value has β

- Can easily extend to models with delay by constraining φ_m.
- However...

↳ more general

Identify dominant order of syst to tune C(s) with model standard

Dominantly I ORD
(*)



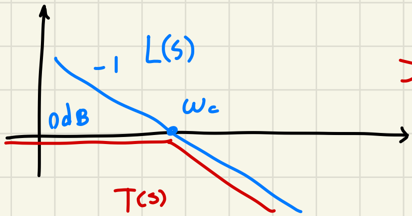
~ I ORD → dominant poles
(look the slowest one)

significantly II ORD (more poles) needed

"dominant poles" : slowest poles NOT cancelled

↓ with an $C(s)$

$$L = \frac{\omega_c}{s}$$



$$\frac{L}{1+L} = \frac{1}{1+s/\omega_c}$$

$\begin{cases} \varphi_m = 90° & \text{phase margin} \\ \omega_c \text{ as desired} \end{cases}$

Accounting for load disturbances

- Abstract from F.G. Shinskey, "Process control: as taught vs. as practiced", Industrial & Engineering Chemistry Research **41**(16), 2002, 3745–3750, DOI `https://doi.org/10.1021/ie010645n`

tune for disturbance rejections

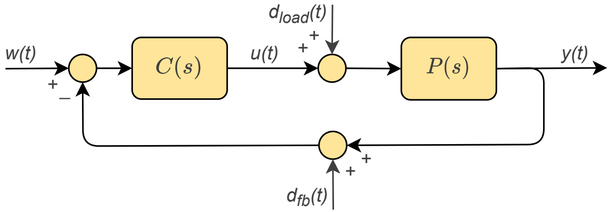
After 47 years of industrial experience and regular reviews of technical papers submitted to this and other journals, the author has discovered a number of common threads in papers written by academics that are at odds with industrial practice in process control. These indicators of the university-industry gap are so consistent over the years as to indicate that the gap is not closing. They include the study of loops with minimum-phase dynamics, overemphasis on set point response to the near exclusion of load regulation, omission of dynamics in the load path when covered at all, unrealistic economic considerations in objective functions, and a preference toward model-based control over PID control. Process models tend to be oversimplified, with little attention given to mass and energy balances and the bilinear models that they produce. Similarly, distributed dynamic models are rarely discussed, despite their predominance in mass and heat transfer. All this has led to research results left unused by industry and graduates left unprepared for industrial assignments.

– not always lin. model

$\Downarrow$

- Let us attempt to provide a small contribution toward filling the evidenced gap ☺

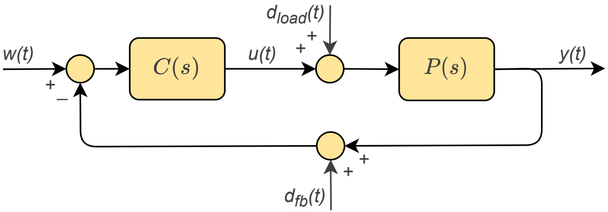
Accounting for load disturbances



- Load disturbances enter the loop at the process input. {summed to $u(t)$}
- Also called "matched" disturbances as homogeneous to u.
- Frequent in process control and in particular in energy systems (e.g., u is cooling power for a room and d_{load} a heat rate from inhabitants/equipment in that room, or u and d_{load} are produced & consumed power in frequency control, as we shall see).

many cases

Accounting for load disturbances



- Transfer function from d_{load} to y:

$$\boxed{G_{yd}(s) = \frac{P(s)}{1+L(s)}.} = \frac{y(t)}{d_{load}(t)}$$

- Remark 1: does not depend only on $L(s)$. ↴ you cannot realy only $L(s)$

  WARNING: $L(s)$ is adimensional (magnitude $\gtrless 1$ makes sense) while $C(s)$ and $P(s)$ in general are not, hence units do count. !

Accounting for load disturbances

mean ω_c

$|1+L| \simeq \sqrt{2}$

$|1+L|^2 \simeq 2$

trade-off...

$\frac{1}{1+L} + \frac{L}{1+L} = 1$

(not both can be small, effect limited enough)

ORTOGONAL $\simeq$ so

$|1+L| = \sqrt{2}$

$\measuredangle L \simeq -90^\circ$ (almost ortogonal)

- We know that $S(s) + T(s) = 1$
  $\Rightarrow$ feedback cannot reject d_{ff} and d_{fb} in the same band.
- Here we have $Q(s)G_{yd}(s) = L(s)/(1+L(s))^2$,
  $\Rightarrow$ to reject d_{load} by feedback in the vicinity of ω_c, where $|L| \approx 1$ (order of unity), $|Q|$ must be "large" in that band — i.e., we say, "feedback must be strong".

($\downarrow |Q|$ high mean ω_c to have good rejection)

$|1+L|$ small, so to reject load disturb, $|Q|$ must be large on that band

- Moreover, we can approximately write

(control sensitivity)

$$Q(j\omega) \approx \begin{cases} 1/P(j\omega) & \omega < \omega_c \\ C(j\omega) & \omega > \omega_c \end{cases} \qquad \text{opposite} \qquad G_{yd}(j\omega) \approx \begin{cases} 1/C(j\omega) & \omega < \omega_c \\ P(j\omega) & \omega > \omega_c \end{cases}$$

which further helps relating the rejection of d_{load} to the concept of "feedback strength".

$$Q = \frac{C}{1+CP} \begin{cases} \nearrow \; 1/P & |CP| \gg 1 \quad (\omega < \omega_c) \\ \searrow \; C & |CP| \ll 1 \quad (\omega > \omega_c) \end{cases}$$

{control sensitivity} APPROX

Accounting for load disturbances

set-point tracking!
be carefule

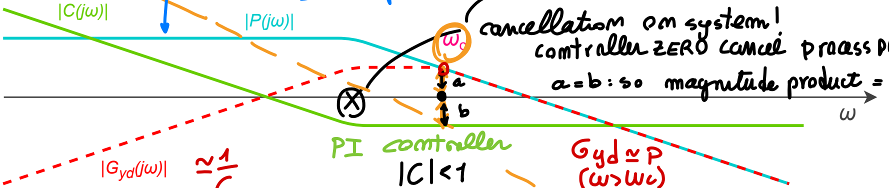
Weak feedback (illustrative example with 1st order process, PI controller)

↓ suppose process magnitude $|P(j\omega)|$ is this one:

I ORD process

$|P|>1$ (positive dB)

cancellation on system!
controller ZERO cancel process POLE
a=b: so magnitude product = 1



PI controller
$|C|<1$

$\simeq \frac{1}{C}$
$(\omega<\omega_c)$

$G_{yd} \simeq P$
$(\omega>\omega_c)$

$L(s) \simeq \frac{\omega_c}{s}$

consistent unit of measure!

- PI tuned by cancelling the process pole.
- The loop transfer function (not drawn for clarity) is ω_c/s, hence good set point tracking.

"WEAK"

is all adimensional NO unit of measure.. use normalized variable so general conclusion, NOT for each unit

(to ensure consistency) However, poor rejection of d_{load}.

↳ normalized values to check for different unit of measure
⇒ so we neglect unit of measures.. less conclusion on unit..

IF

$|CP| = 1$ which one if >1, and which <1

$\Rightarrow$ here

$|C| < 1$, $|P| > 1$ on that scheme

"weak" $\rightarrow$ I amplify the load disturbance !

(larger effect than cause !)

try to exclude load distub! set-point traking good but
disturb nor rejected

Accounting for load disturbances

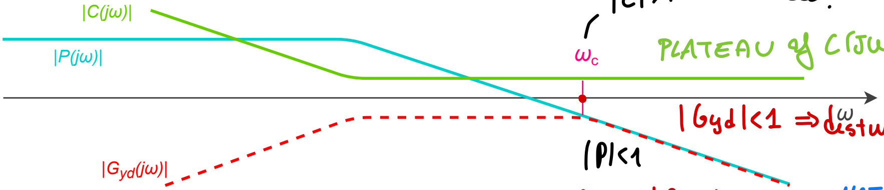
I have to modify ω_c

Strong feedback preserving cancellation tuning (same example)

⇓ to reject disturb with set point follow

@ ω_c ⇓ $|C|>1$ ⇒ apparently $|G_{yd}|<1$ rejecting disturbance well!

cancellation tuning! increasing $|C|$ gain



stability problem!

NOT always good!

- (*) The loop transfer function is still ω_c/s but had to increase ω_c.
- Set point response can be too nervous (curable with 2-dof)... → (too fast responce!)
- ...but worse, the high-frequency control sensitivity is large,
- hence measurement noise in d_{fb} is amplified on u (actuators are not happy with that, in general).

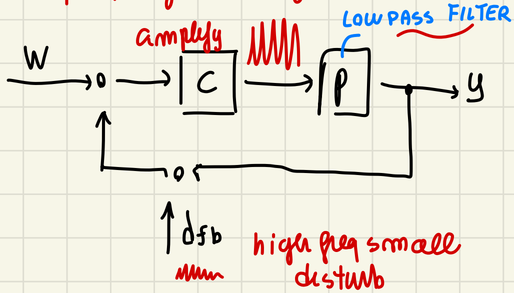
(*) PROBLEMS: possible stability problems because dominantly I ORD + unmodelled dynamic... set desired speed ω_c → try to regulate in freq where model is not well defined → issue stability

- set point responce may be too fast! not always good ... we wanna guarantee the responce request (NOT too fast) $\omega_{MIN} < \omega_c < \omega_{MAX}$ → IF too nervous you can pre-filter (low pass filter)

- control sensitivity Q @ high freq is large!

set low/$\underbrace{\text{high}}_{\text{not too fast}}$ limit on Responce time

|Q| high @ high freq



You don't see that on y(t) but you control badly u(t)

small useless movement of actuator! → (long run issue) reduce actuator lifetime

↓

in CONTROL tuning... carefully on actuation

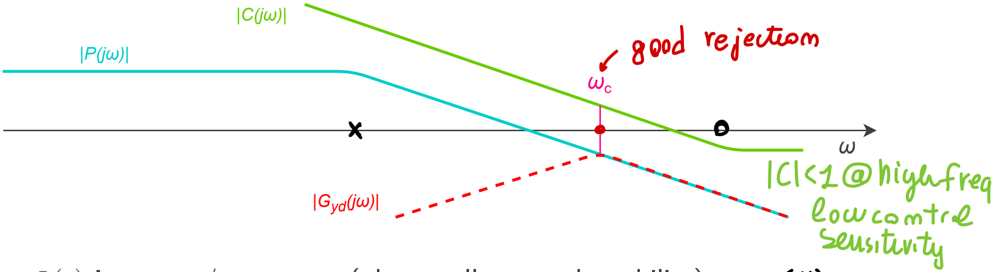
$$\left\{ \begin{array}{l} \text{IN CONTROL TUNING} \\ \text{look in } u(t) \end{array} \right\} !$$

↑ take care on actuator

Accounting for load disturbances

Strong feedback abandoning cancellation tuning (same example)

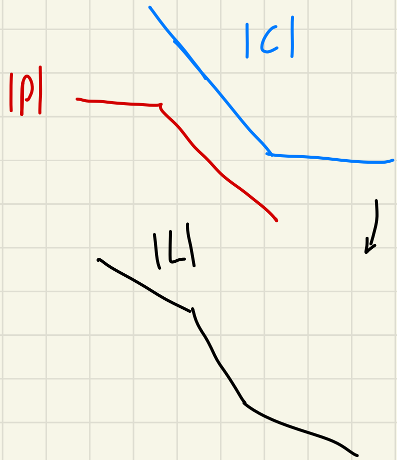
other tuning solution



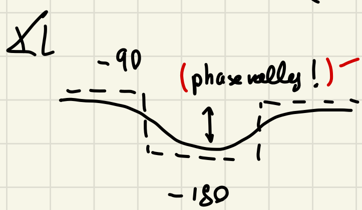
- Here $L(s)$ is not ω_c/s anymore (phase valley, watch stability). ⟹ (*)
- Set point response can be still recovered exploiting 2-dof...
- ...but this time the high-frequency control sensitivity is lower,
- and there is less/no *plateau* in $|G_{yd}(j\omega)|$, speeding up the rejection of d_{load} (more low-frequency content is cut).

more low freq content is cut away better by this scheme

(*)



logaritmic summ to obtain L(jw) Bode plot



carefully!

Feedback loops

NOT only tuned for set-up tracking

But for → stability + disturbance rejection !

(for example in ROBOTICS Feed Forward not used!)

disturbance rej speed increase because more low freq content cut away

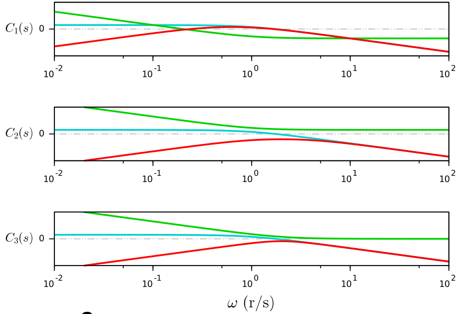
output spectrum : input * |TF| → low freq better cut away by this scheme ↴

so faster disturbance rejection

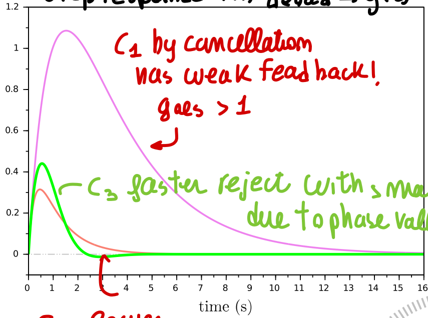
Accounting for load disturbances

Bode plots and y responses to a d_{load} step – a numeric example

comparison of the three solutions!



step responce on $d_{load} \rightarrow y(t)$



C_1 by cancellation has weak feed back! goes > 1

C_3 faster reject with small oscillation due to phase valley

C_2 lower but same shape of C_1! less gain, scaled

due to phase valley... less stability

$$P(s) = \frac{2}{1+s};$$

$$C_1(s) = 0.2\left(1+\frac{1}{s}\right), \quad C_2(s) = 2\left(1+\frac{1}{s}\right), \quad C_3(s) = 1+\frac{1}{0.5s}.$$

cancellation scheme → weak feedback | cancellation with high gain | ↑ rejection faster!

usually we see
mostly tune by cancellation + proper design

↓

now we should
look better on
consequences of controller
design! better reasoning
on rejection and so one...

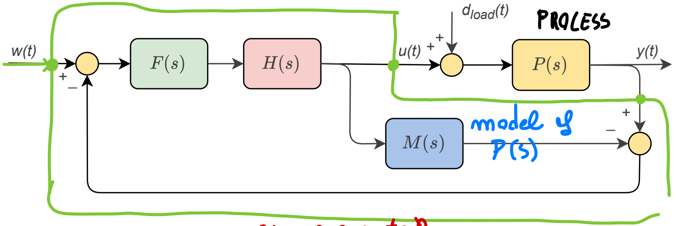
cancellation tuning
+ pushing tuning capability

A more formal and powerful technique

{Formalize this techniques in General procedures}

Internal Model Control scheme

↳ knowing process P(s) by a model M(s)



"internal model" because M(s) inside C(s)

⇑

(inside the computer)

CONTROLLER

W → u

y

- In nominal conditions ($M = P$, $d_{load} = 0$) the loop is open, hence

asymp goes to 0

(perfect model)

no disturb

$$\boxed{\frac{Y(s)}{W(s)} = P(s)H(s)F(s).}$$

|no loop!|

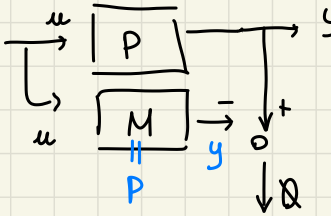
(neglecting initial state that will go to 0)

- And if in addition one could set $H(s) = P(s)^{-1}$ $= M(s)^{-1}$

prescribe the dynamic

$$\boxed{\frac{Y(s)}{W(s)} = F(s)}$$

↳ prescribe set point dynamic by controller variable nominal condition $P = M$ $d_{load} = 0$



no feedback signal → (open loop)

no uncertainty, no disturbance ⇒ useless feedback!

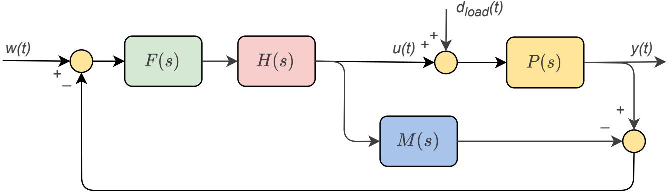
$$\frac{Y}{W} = PHF \underset{P=M}{=} MHF$$

A more formal and powerful technique

Internal Model Control scheme

assign dynamic with carefulness ← If P contain zeros >0 NOT invertible!

If P has rel deg = 2 also F should have rel deg ≥ 2



{realization issues}

$P = \frac{1}{(1+sT_1)(1+sT_2)}$

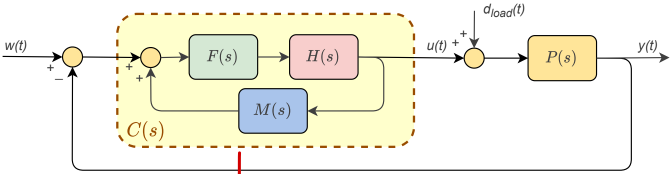
P^{-1} not realizable!

- Elegant proof that
  - in the absence of model error and disturbances feedback is useless (the loop opens spontaneously), (IN THAT hypotesis)
  - one can assign the $w \to y$ dynamics, but with a relative degree at least equal to that of P (otherwise HF is not realisable) → If problem in realization
  - and preserving in that dynamics possible non minimum-phase parts of P (otherwise, causality violation and/or critical cancellations).

A more formal and powerful technique

Internal Model Control (IMC) PI/PID tuning

⇓ manipulating previous scheme!



- The IMC scheme is equivalent to a standard feedback one with

$$C(s) = \frac{H(s)F(s)}{1-H(s)F(s)M(s)}.$$

- This is useful for specialising the scheme, e.g. to PI/PID controllers and FOPDT (asymptotically stable) models.

(IOAD)

A more formal and powerful technique

Internal Model Control (IMC) PI/PID tuning

H(s) inverse of what is meaningful to invert on M(s)

H·F must be Realizable overall |F|≃1 gain

- Let us take

delay

F@ least one pole for realizability

$$M(s)=\mu\frac{e^{-sD}}{1+sT},\quad H(s)=\frac{1+sT}{\mu},\quad F(s)=\frac{1}{1+s\boxed{\lambda}}.$$

λ: desired close loop time constant

- This corresponds to zero steady-state error for constant inputs as $F(0)=1$.
- Parameter λ is readily interpreted as the desired closed-loop (dominant) time constant (easy to relate to a response speed request).
- Doing so we get computing C(s)... NOT rational..with a delay

$$C(s)=\frac{\frac{1+sT}{\mu}\frac{1}{1+s\lambda}}{1-\frac{1+sT}{\mu}\frac{1}{1+s\lambda}\mu\frac{e^{-sD}}{1+sT}}=\frac{1}{\mu}\frac{1+sT}{1+s\lambda-e^{-sD}}.$$

delay!

NOT Rational

padet approx!

A more formal and powerful technique

Internal Model Control (IMC) PI/PID tuning

$e^{-sD} = 1-sD$ (1,0)

- Now, replace the delay term with the (1,0) Padé approximation $1-sD$:

{approx delay!}

$$C(s)=\frac{1}{\mu}\frac{1+sT}{\cancel{1}+s\lambda-\cancel{1}+sD}=\frac{1+sT}{s\mu(\lambda+D)} \quad \Rightarrow \quad \text{PI. controller}$$

- Instead, use the (1,1) Padé $\frac{1-sD/2}{1+sD/2}$:

$$C(s)=\frac{1}{\mu}\frac{1+sT}{1+s\lambda-\frac{1-sD/2}{1+sD/2}}=\ldots=\frac{(1+sT)(1+sD/2)}{s\mu(\lambda+D)\left(1+s\frac{\lambda D}{2(\lambda+D)}\right)} \quad \Rightarrow \quad \text{real PID.}$$

- For the 1-dof ISA PI

$$\boxed{C(s)=K\left(1+\frac{1}{sT_i}\right)}$$

tuning formula

we readily get the IMC tuning formulæ $K=\frac{T}{\mu(\lambda+D)}$, $T_i=T$.

A more formal and powerful technique

Internal Model Control (IMC) PI/PID tuning

PID standard form

- For the 1-dof ISA PID

$$\boxed{C(s) = K\left(1+\frac{1}{sT_i}+\frac{sT_d}{1+s\frac{T_d}{N}}\right)}$$

the resulting IMC tuning formulæ – in sequence – are

$$T_i = T+\frac{D^2}{2(\lambda+D)}, \quad K = \frac{T_i}{\mu(\lambda+D)},$$

$$N = \frac{T(\lambda+D)}{\lambda T_i}-1, \quad T_d = \frac{\lambda DN}{2(\lambda+D)}.$$

A more formal and powerful technique

F) desired loop

"Internal Model Control" (IMC) PI/PID tuning

↳ controller $C(s)$ contains the model $M(s)$ inside it!

periodic model update

- Easy to employ (check it out *by hand*).
- Useful for rapid adaptation: keep F, update M and C updates as a consequence.
- Capable of accommodating for a delay in the model.
- Capable of tuning also the "derivative pole" of the real PID. Tuning an ideal PID and then setting N to a "default high" value can displace its zeroes to undesired positions; recall also that the high-frequency Q magnitude is $K(1+N)$. carefull!

powerfull but based on cancellation → possible problems

- However still operating by cancellation, with the known shortcomings as for the rejection of $\underbrace{d_{load}}_{\text{unless... (*)}}$. ⇒

we focus on tuning / modelling respect experiment consider PID ideal form

$$P + I + D$$

$$C(s) = K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + \frac{sT_d}{N}}\right)$$

typically N setted to avoid control sensitivity issue

+additional pole

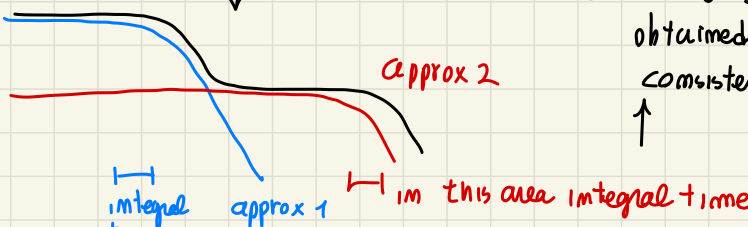
tuning without N... if we set N higher later → displace controller zeros

(numerator will contain N)

roots converging ! ⇒ can do Bad → good to tune N properly

tuning by cancellation using approx.. model error but good tuning..
MISMATCH on process control

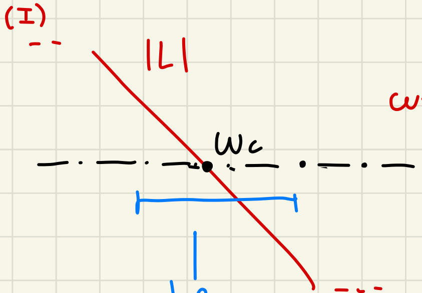
(*) d load rejection...
IF you have a process



cancellation tuning C(s) with this approx...

tuning by cancellation obtained with consistency ...

not @ low freq...



model MUST be accurate near Wc !

wrong on (I), (II) don't care!

depending on where I want to set Wc I chose between approx 1 or 2 of the process

A more formal and powerful technique

S-IMC PI tuning

Alternative tech!

to avoid plateau we moove |C(s)| ZERO @ higher freq. respect process pole

- Remember the example? We pushed the PI zero to a higher frequency w.r.t. the model pole.
- That is, we attempted to make T_i as small as possible compatibly with the specifications. → stability degree specification! (Carefully about phase valley)
- S-IMC rule (we can limit the scope to the PI case):

$$K = \frac{T}{\mu(\lambda + D)}, \quad T_i = \min(T, 4(\lambda + D)).$$

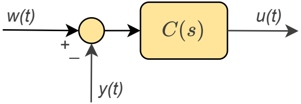
usual gain particular Ti formula

- Same interpretation for λ.
- Check it out for d_{load} rejection in comparison to standard IMC.

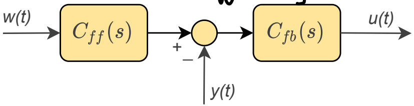
One- and two-degree-of-freedom (1-dof and 2-dof) controllers

more formally ↓ you can use 1dof controller (because W→u, y→u T.F) same dynamic

1DOF



2DOF $\frac{u}{W} \neq \frac{u}{y}$



way that control signal responce to W or y are the same!

- 1-dof (left): $\frac{U(s)}{W(s)} = -\frac{U(s)}{Y(s)} = C(s)$; cannot assign the two independently. (except for sign)
- 2-dof (right): $\frac{U(s)}{W(s)} = C_{fb}(s)C_{ff}(s)$, $\frac{U(s)}{Y(s)} = C_{fb}(s)$; can assign the two independently.
- In the 2-dof case $C_{ff}(s)$ (feed foreward) acts as a set point filter, hence must be asymptotically stable and with unity gain (equal signal to set point)
- Apparently, only $C_{fb}(s)$ influences the loop stability.

  [tuned for stability

Regulatory *versus* servo tuning

in general

(ON LITERATURE) distinction on possible TUNING

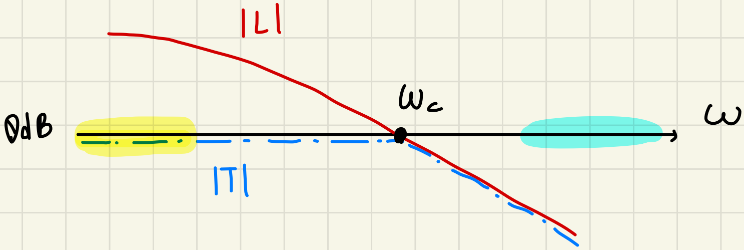
reactive controller to reject but with possible instability issue..

- Regulatory tuning
  primary purpose is disturbance rejection, strong feedback is almost invariantly required, only C_{fb} is involved (stability/rejection tradeoff).
- Servo tuning
  primary purpose is set point tracking, both C_{fb} and C_{ff} are involved.

some rule relevant! IMPLEMENTING IT..

- There are tuning rules for C_{fb} to privilege either objective.
- It is common practice to first tune C_{fb} for stability and disturbance rejection, then use C_{ff} to recover set point tracking if required.
- Remember that feedback kills uncertainty only up to ω_c, hence beware if C_{ff} needs to accelerate the system beyond (unless your process model is guaranteed good enough in that band).

• feedback kills uncertainty up to ω_c



if you want:

• set-point responce "slower than ω_c"
(time constant of set-point follow slow than ω_c)

you need ⇓ LOW-PASS (respect ω_c)
set point prefilter, cutting in this band
↓
(where $\frac{C_{fb} P}{1 + C_{fb} P} \simeq 1$)

be aware when speed up

BUT: • set-point responce "Faster than ω_c",
requires to enhance High Freq (with respect to ω_c)
components in the set point, and here
↓
feedback does NOT reduce uncertainty in $P(s)$
(on a range where the process can be unkn.)

Exploiting 2-dof controllers

An example

(use "Modelica" model to test this)

⇑

- Process:

$$P(s) = \frac{1}{(1+10s)(1+s)}.$$

- Weak and strong feedback:

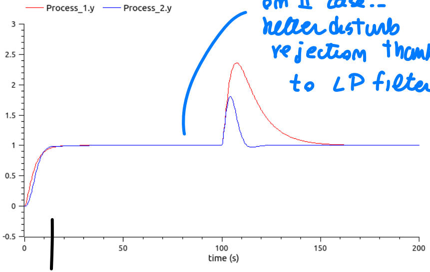
$$C_{fb1} = 2\left(1+\frac{1}{10s}\right), C_{fb2} = 4\left(1+\frac{1}{4s}\right).$$

- Feedforward (set point prefilter):

$$C_{ff1} = 1, \quad C_{fb2} = \frac{1}{1+5s}.$$

- Set point and load disturbance unit step: observe the controlled variable response.

on II case... better disturb rejection thanks to LP filter



same responce approx... with different load disturbance responce

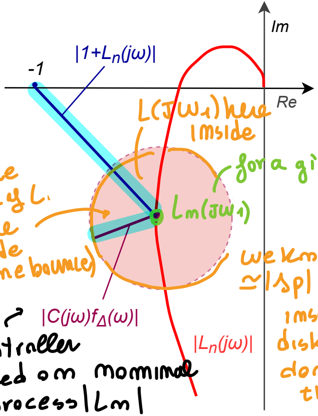
Commonly on our system we re-tune basing on local experim. (re-tune some parameter)

take step-response → (compute data)

Quantifying robustness *a priori*

A sufficient condition for robust stability (in the Bode hypotheses)

ROBUST TUNING ?



$L(j\omega_1)$ here inside

true value of L is here inside (some boundle)

$L_n(j\omega_1)$

for a given ω_1

we know $\simeq |\Delta_P| < f_\Delta(\omega)$ inside a disk (we don't know the phase)

controller tuned on nominal process $|L_n|$

- Nominal loop $L_n(j\omega)$ does not encircle -1.
- Additive model error: $P(s) = M(s) + \Delta_P(s)$.
- Loop transfer function: $L(s) = (M(s) + \Delta_P(s))C(s)$.
- That is, w.r.t. nominal, $L(s) = L_n(s) + \Delta_P(s)C(s)$.
- Error magnitude bound: $|\Delta_P(j\omega)| < f_\Delta(\omega)\ \forall\omega$.
- Sufficient condition: $|C(j\omega)f_\Delta(\omega)| < |1 + L_n(j\omega)|\ \forall\omega$.
- Nominal control sensitivity: $Q_n(s) = \frac{C(s)}{1+L_n(s)}$.

considering Q_n T.F

- Summing up, the condition is

$$\boxed{\left|f_\Delta(\omega)\right| < \frac{1}{|Q_n(j\omega)|}}\ \forall\omega.$$

computing $|Q_n|$ provide you the max accettable model error $\forall\omega$ (overbound)

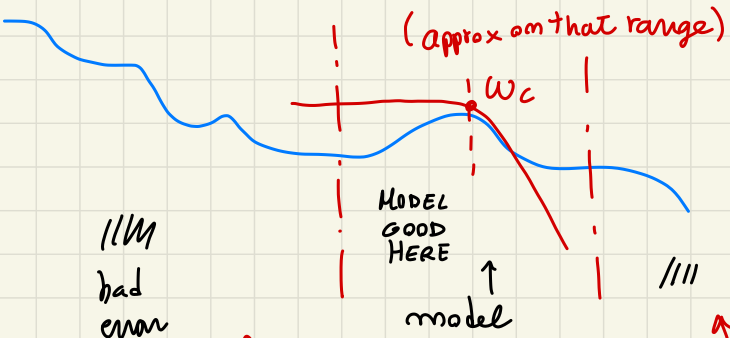
The inverse of the nominal control sensitivity provides a magnitude bound for the acceptable additive model error.

PROPERTY is "ROBUST" IF once guaranteed nominally is
also guarantee for some uncertainty
on the process
↓
guarantee of small Gain change → STILL stable

tuning on locally info : only 1 experiment... you don't have
the idea of variability of process (ubiquous)

(tuning - time limited)
⇓
- how to quantify Robustness a priori?

(phase margin is an indicator of Robustness)
↓
taking a complex freq responce model

local approx
↑


↑
model
errors significant
outside that range

But how about errors here

Is φ_m enough to give guarantees?

- nominal process model M
- true process : $P = M + \underbrace{\Delta P}_{\text{unknown}}$ additive model error

- nominal loop T.F $L_m = CM$
- true loop T.F $L = C(M+\Delta P) = L_m + C\Delta P$

obviously: controller must stabilize nominal process

↓

- on Bode hyp. for Nyquist theorem... Nyquist plot of L_m don't encircle $-1 + j0$

(sufficient condition) for STABILITY robustness

IF $|C(j\omega) f_\Delta(\omega)| < |1 + L_m(j\omega)| \ \forall \omega$

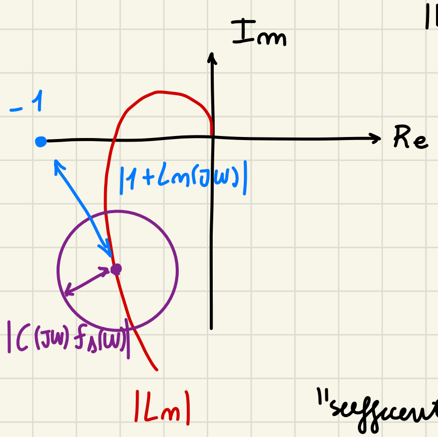
⇓

for sure the disk does NOT touch (−1) point!

not encircle −1

⇓

model error ΔP don't destroy stability



"sufficient" because the circle may be larger but the phase such that always inside..

- NOT Necessary condition

↓ it could transpass −1 but maybe not because phase move away −1

Example

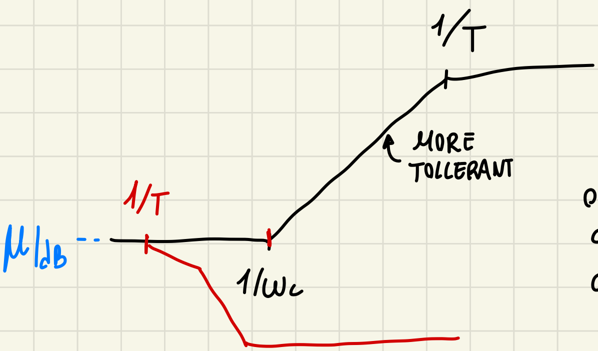
given $M(s) = \frac{\mu}{1+sT}$ $\quad C(s) = \frac{\omega_c}{s}\,\frac{1+sT}{\mu}$ cancellation C(s)

$$Q_m = \frac{C(s)}{1+L(s)} = \frac{\omega_c}{s}\,\frac{1+sT}{\mu}\,\frac{s\cdot 1}{s\cancel{1}+\frac{\omega_c}{\cancel{s}}} =$$

$$= \frac{\cancel{\omega_c}}{\cancel{s}}\,\frac{1+sT}{\mu}\,\frac{\cancel{s}/\cancel{\omega_c}}{1+s/\omega_c} = \frac{1}{\mu}\,\frac{1+sT}{1+s/\omega_c}$$

$$\left[\frac{1}{Q_m} = \mu\,\frac{1+s/\omega_c}{1+s/T}\right]$$ overbound of accetpatble $f_s(\omega)$

inversion... done as...



{ IT can be like this
or like that }

overbound of acceptable f_s... accept an HIGHER MODEL ERROR
↓
you must stay under that inversion

↓ so...

when tuning C(s) account also possible uncertainty on the process (variation)
↓
in that way desired ... (techniques to ensure robustness)