Automatic control for energy transition

Advanced process control using simple elements* - «agile» production» allowing for moving constraints and enabling fast changes in production rate

Sigurd Skogestad

Department of Chemical Engineering
Norwegian University of Science and Technology (NTNU)
Trondheim

Salvador
12 Nov. 2024

*a.k.a. Advanced regulatory control (ARC)

Midnight or midday?

Geiranger fjord

Trondheim

Trondheim

Arctic circle

“The goal of my research is to develop simple yet rigorous methods to solve problems of engineering significance”

Sigurd Skogestad Professor

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), N7491 Trondheim, Norway

Start here...

- About me - CV - Powerpoint presentations - How to reach me - *Email: skoge@ntnu.no*
- Teaching: Courses - Master students - Project students
- Research: My Group - Research - Ph.D. students - Academic tree

"The overall goal of my research is to develop simple yet rigorous methods to solve problems of engineering significance"

"We want to find a self-optimizing control structure where close-to-optimalo operation under varying conditions is achieved with constant (or slowly varying) setpoints for the controlled variables (CVs). The aim is to move more of the burden of economic optimization from the slower time scale of the real-time optimization (RTO) layer to the faster setpoint control layer. More generally, the idea is to use the model (or sometimes data) off-line to find properties of the optimal solution suited for (simple) on-line feedback implementation"

"News"...

- 27 Nov. 2023: Welcome to the SUBPRO Symposium at the Britannia Hotel in Trondheim
- Aug. 2023: Tutorial review paper on "Advanced control using decomposition and simple elements". Published in Annual reviews in Control (2023). [paper] [tutorial workshop] [slides from Advanced process control course at NTNU]
- 05 Jan. 2023: Tutorial paper on "Transformed inputs for linearization, decoupling and feedforward control" published in JPC. [paper]
- 13 June 2022: Plenary talk on "Putting optimization into the control layer using the magic of feedback control", at ESCAPE-32 conference, Toulouse, France [slides]
- 08 Dec. 2021: Plenary talk on "Nonlinear input transformations for disturbance rejection, decoupling and linearization" at Control Conference of Africa (CCA 2021), Magaliesburg, South Africa (virtual) [video and slides]
- 27 Oct. 2021: Plenary talk on "Advanced process control - A newe look at the old" at the Brazilian Chemical Engineering Conference, COBEQ 2021, Gramado, Brazil (virtual) [slides]
- 13 Oct. 2021: Plenary talk on "Advanced process control" at the Mexican Control Conference, CNCA 2021 (virtual) [video and slides]
- Nov. 2019: Sigurd receives the "Computing in chemical engineering award from the American Institute of Chemical Engineering (Orlando, 12 Nov. 2019)
- June 2019: Best paper award at ESCAPE 2019 conference in Eindhoven, The Netherlands
- July 2018: PID-paper in JPC that verifies SIMC PI-rules and gives "Improved" SIMC PID-rules for processes with time delay (taud=theta/3)
- June 2018: Video of Sigurd giving lecture at ESCAPE-2018 in Graz on how to use classical advanced control for switching between active constraints
- Feb. 2017: Youtube vidoes of Sigurd giving lectures on PID control and Plantwide control (at University of Salamanca, Spain)
- 06-08 June 2016: IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS-2016), Trondheim, Norway.
- Videos and proceedings from DYCOPS-2016
- Aug 2014: Sigurd recieves IFAC Fellow Award in Cape Town
- 2014: Overview papers on "control structure design and "economic plantwide control"
- OLD NEWS

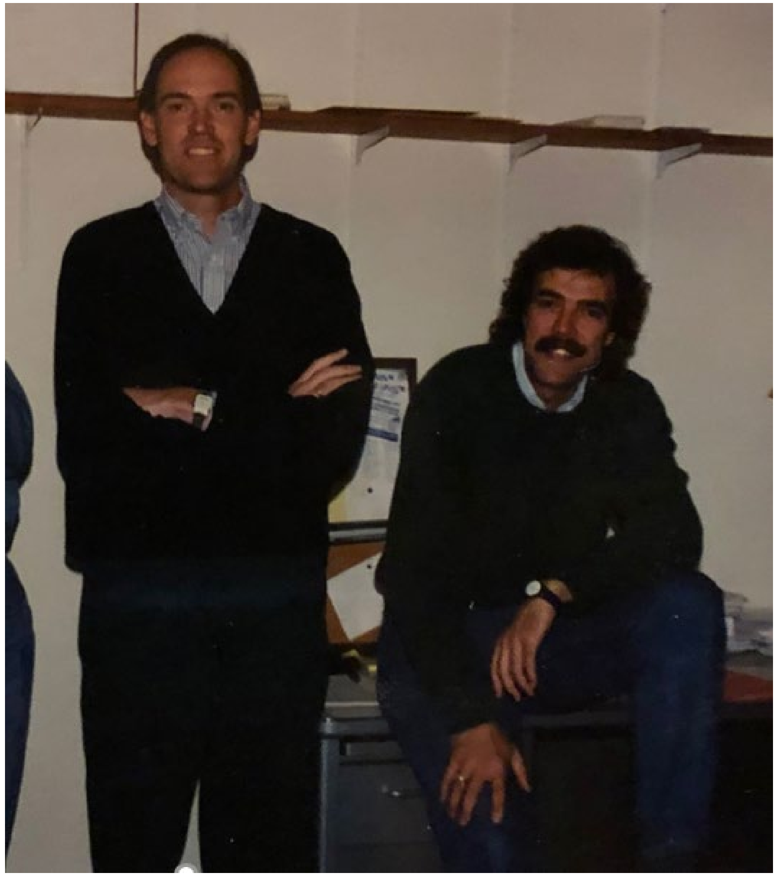
Books...

- Book: S. Skogestad and I. Postlethwaite: MULTIVARIABLE FEEDBACK CONTROL-Analysis and design. Wiley (1996; 2005)
- Book: S. Skogestad: CHEMICAL AND ENERGY PROCESS ENGINEERING CRC Press (Taylor&Francis Group) (Aug. 2008)
- Bok: S. Skogestad: PROSESSTEKNIKK- Masse- og energibalanser Tapir (2000; 2003; 2009).

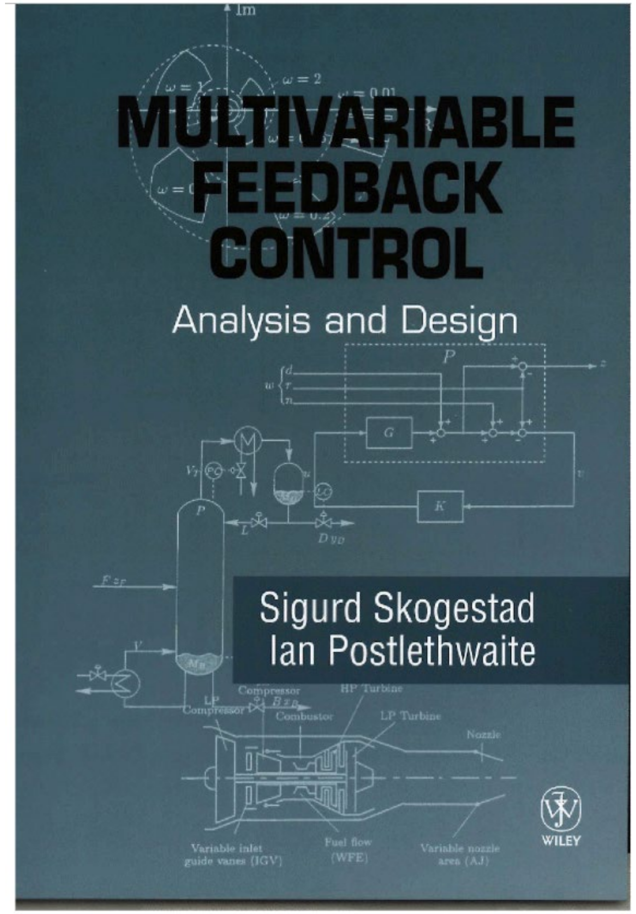
More information ...

- **Publications** from my Google scholar site
- **Download publications** from my official publication list .... or look HERE if you want to download our most recent and upublished work
- Proceedings from conferences - some of these may be difficult to obtain elsewhere
- Process control library - We have an extensive library for which Ivar has made a nice on-line search
- Photographs that I have collected from various events (maybe you are included...)
- International conferences - updated with irregular intervals
- SUBPRO (NTNU center on subsea production and processing) [Annual reports] [Internal]
- Nordic Process Control working group - in which we participate
- 5-year Master program in Chemical and Biochemical Engineering at NTNU (MTKJ) - Sigurd Skogestad is Program Leader 2019-2025.

Robust control



Berkeley, Dec. 1994



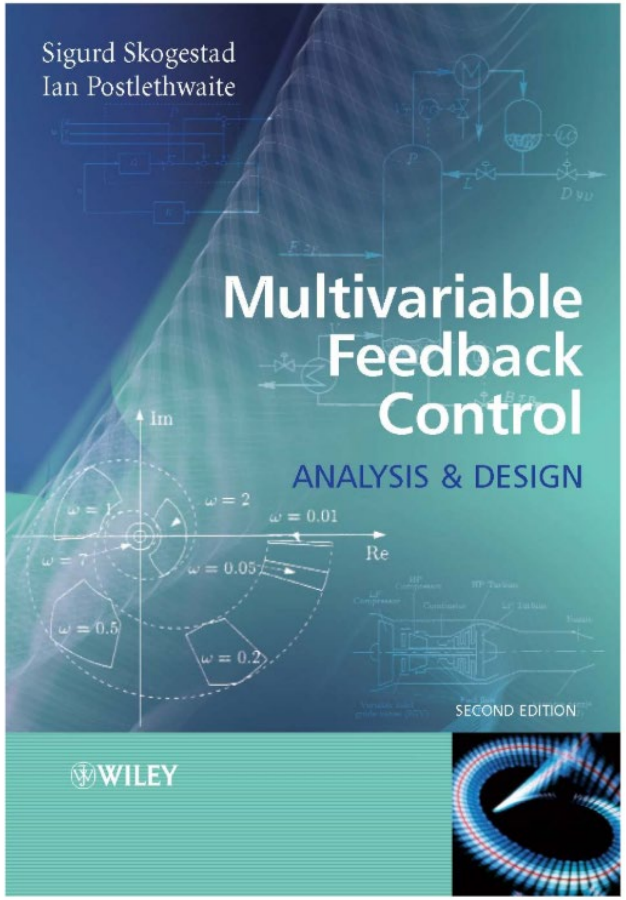
MULTIVARIABLE FEEDBACK CONTROL

Analysis and Design

Sigurd Skogestad
Ian Postlethwaite

WILEY

1996



Sigurd Skogestad
Ian Postlethwaite

Multivariable Feedback Control

ANALYSIS & DESIGN

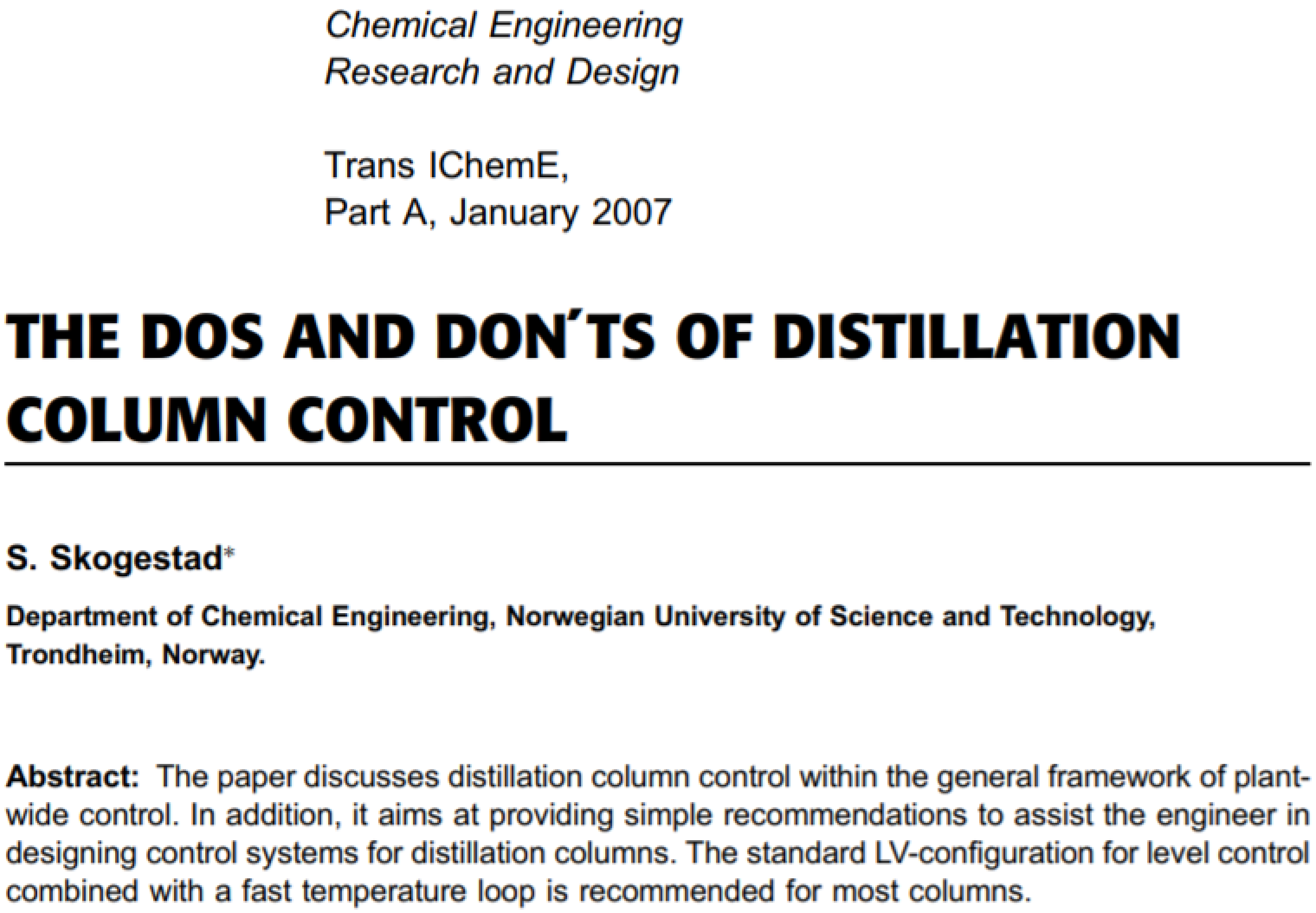
SECOND EDITION

WILEY

2005

Distillation

At home doing moonshine distillation (1979)



Chemical Engineering
Research and Design

Trans IChemE,
Part A, January 2007

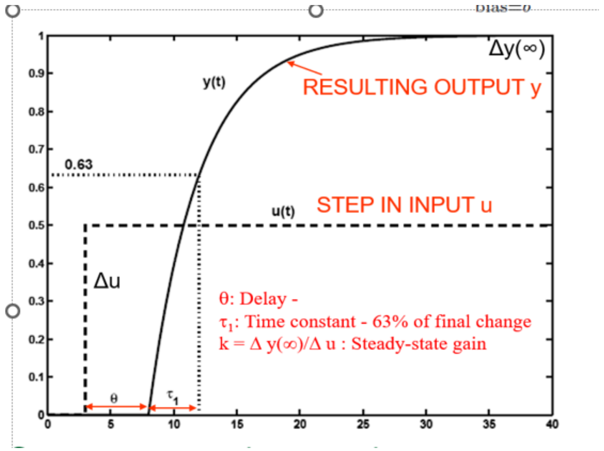
THE DOS AND DON´TS OF DISTILLATION COLUMN CONTROL

S. Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway.

Abstract: The paper discusses distillation column control within the general framework of plant-wide control. In addition, it aims at providing simple recommendations to assist the engineer in designing control systems for distillation columns. The standard LV-configuration for level control combined with a fast temperature loop is recommended for most columns.

SIMC* PID tuning rule (2001,2003)



$$g(s) = \frac{k}{(\tau_1 s+1)(\tau_2 s+1)} e^{-\theta s}$$

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta}$$

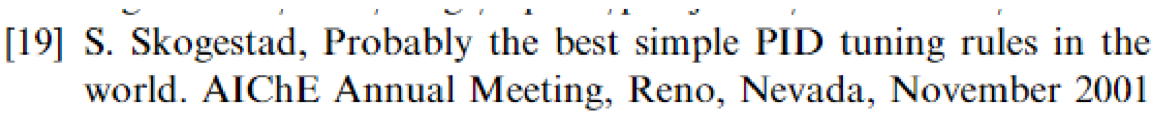
$$\tau_I = \min\{\tau_1, 4(\tau_c+\theta)\}$$

$$\tau_D = \tau_2$$

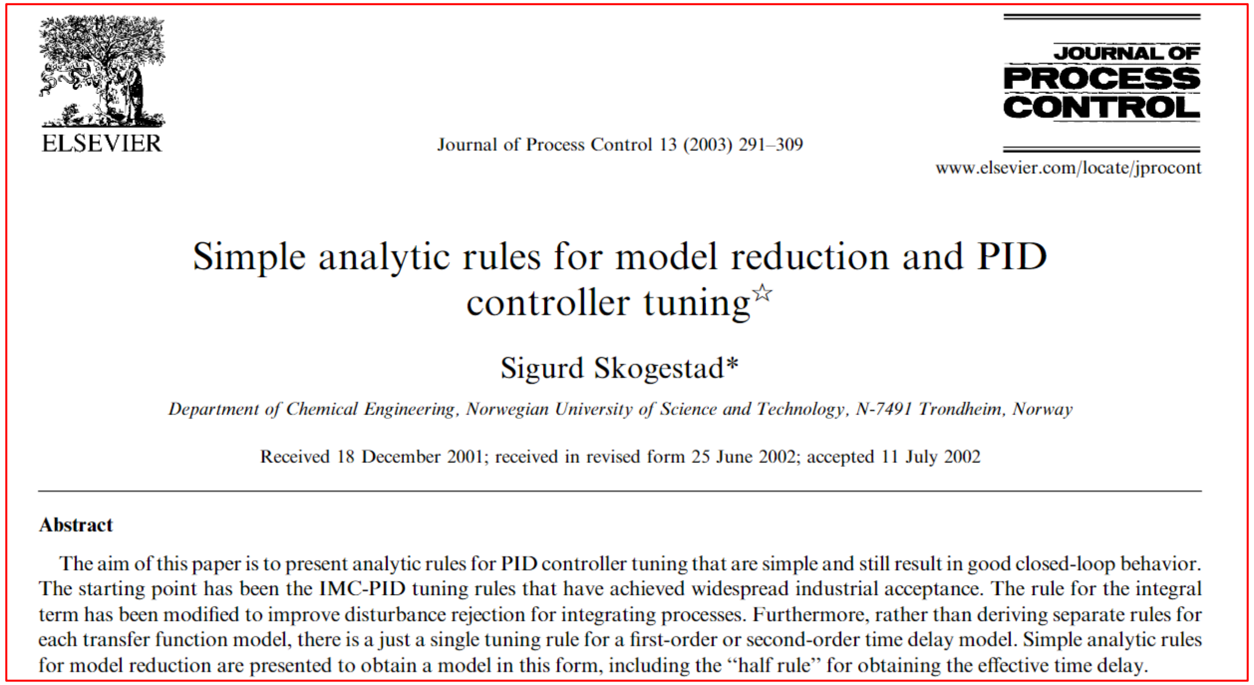
Tuning parameter:

$\tau_c \geq \theta$

$=\lambda$



[19] S. Skogestad, Probably the best simple PID tuning rules in the world. AIChE Annual Meeting, Reno, Nevada, November 2001



ELSEVIER

Journal of Process Control 13 (2003) 291–309

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Simple analytic rules for model reduction and PID controller tuning☆

Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Received 18 December 2001; received in revised form 25 June 2002; accepted 11 July 2002

Abstract

The aim of this paper is to present analytic rules for PID controller tuning that are simple and still result in good closed-loop behavior. The starting point has been the IMC-PID tuning rules that have achieved widespread industrial acceptance. The rule for the integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, there is a just a single tuning rule for a first-order or second-order time delay model. Simple analytic rules for model reduction are presented to obtain a model in this form, including the "half rule" for obtaining the effective time delay.

*SIMC = Simple/Skogestad IMC

Chemical Engineering

Sigurd Skogestad

PROSESSTEKNIKK

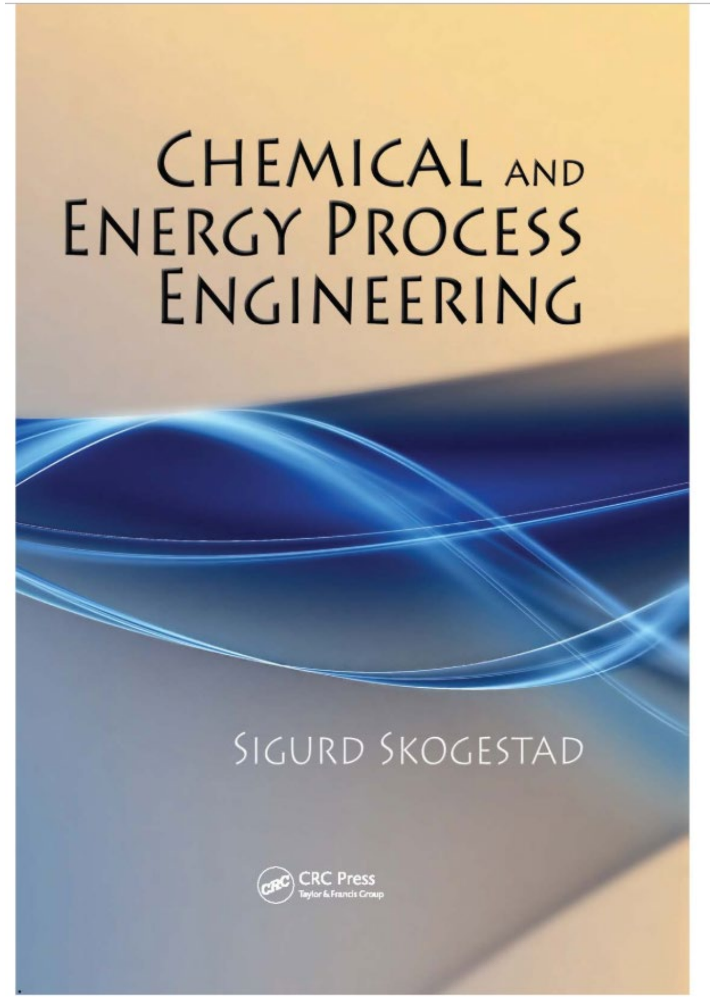
Masse- og energibalanser

2. utgave

tapir akademisk forlag

2000, 2003, 2009

2000

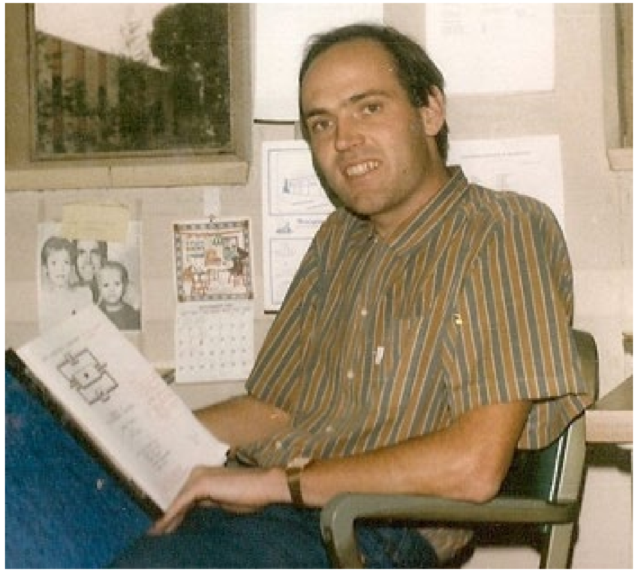


CHEMICAL AND ENERGY PROCESS ENGINEERING

SIGURD SKOGESTAD

CRC Press
Taylor & Francis Group

2009



Sigurd at Caltech (1984)

How we design a control system for a complete chemical plant?

- Where do we start?
- What should we control? and why?
- etc.
- etc.

AIChE

Economic Plantwide Control of the Ethyl Benzene Process

Rahul Jagtap, Ashok S Pathak, and Nitin Kaistha
Dept. of Chemical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016,Uttar Pradesh, India

DOI 10.1002/aic.13964

A1: Benzene
A2: Ethylene
B: Ethylbenzene (product)
C: Diethylbenzene (undersired, recycled to extinction)
$A1+A2 \rightarrow B$
$B + A2 \rightarrow C$
$C + A1 \rightarrow 2B$

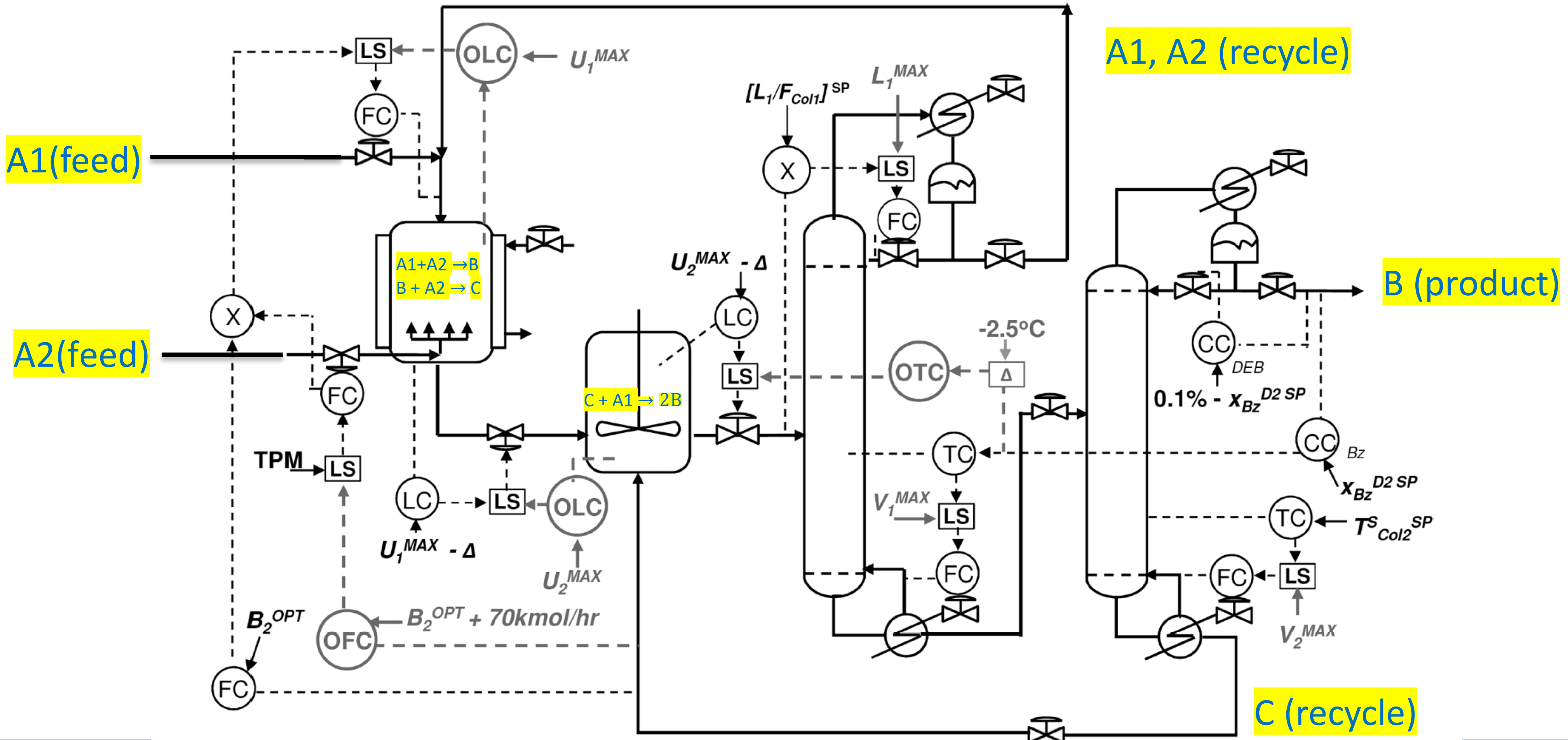


Figure 7. CS2 with overrides for handling equipment capacity constraints.

Control system structure*

Alan Foss ("Critique of chemical process control theory", AIChE Journal,1973):

The central issue to be resolved ... is the determination of control system structure.*

Which variables should be measured, which inputs should be manipulated and which links should be made between the two sets?

*Current terminology: Control system architecture

Main objectives of a control system

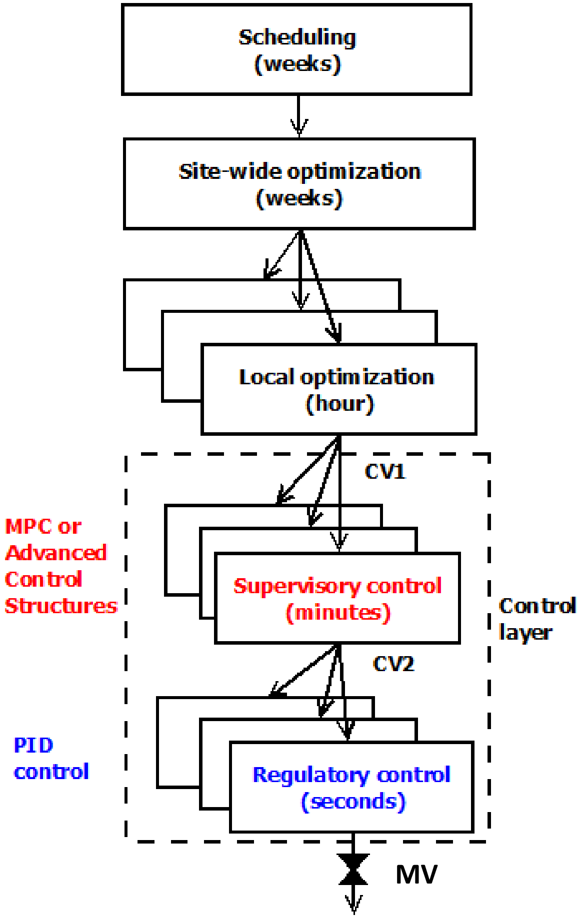
1. Economics: Implementation of acceptable (near-optimal) operation
2. Regulation: Stable operation

ARE THESE OBJECTIVES CONFLICTING?

- Usually NOT
  - Different time scales
    - Stabilization → fast time scale
  - Stabilization doesn't "use up" any degrees of freedom
    - Reference value (setpoint) available for layer above
    - But it "uses up" part of the time window (frequency range)

Two fundamental ways of decomposing the controller

- Vertical (hierarchical; cascade)
- Based on time scale separation
- Decision: Selection of CVs that connect layers



Scheduling
(weeks)

Site-wide optimization
(weeks)

Local optimization
(hour)

CV1

MPC or
Advanced
Control
Structures

Supervisory control
(minutes)

Control
layer

CV2

PID
control

Regulatory control
(seconds)

MV

PROCESS

- Horizontal (decentralized)
- Usually based on distance
- Decision: Pairing of MVs and CVs within layers

In addition: Decomposition of controller into smaller elements (blocks):
Feedforward element, nonlinear element, estimators (soft sensors), switching elements

CV = controlled variable

MV = manipulated variable

QUIZ

What are the three most important inventions of process control?

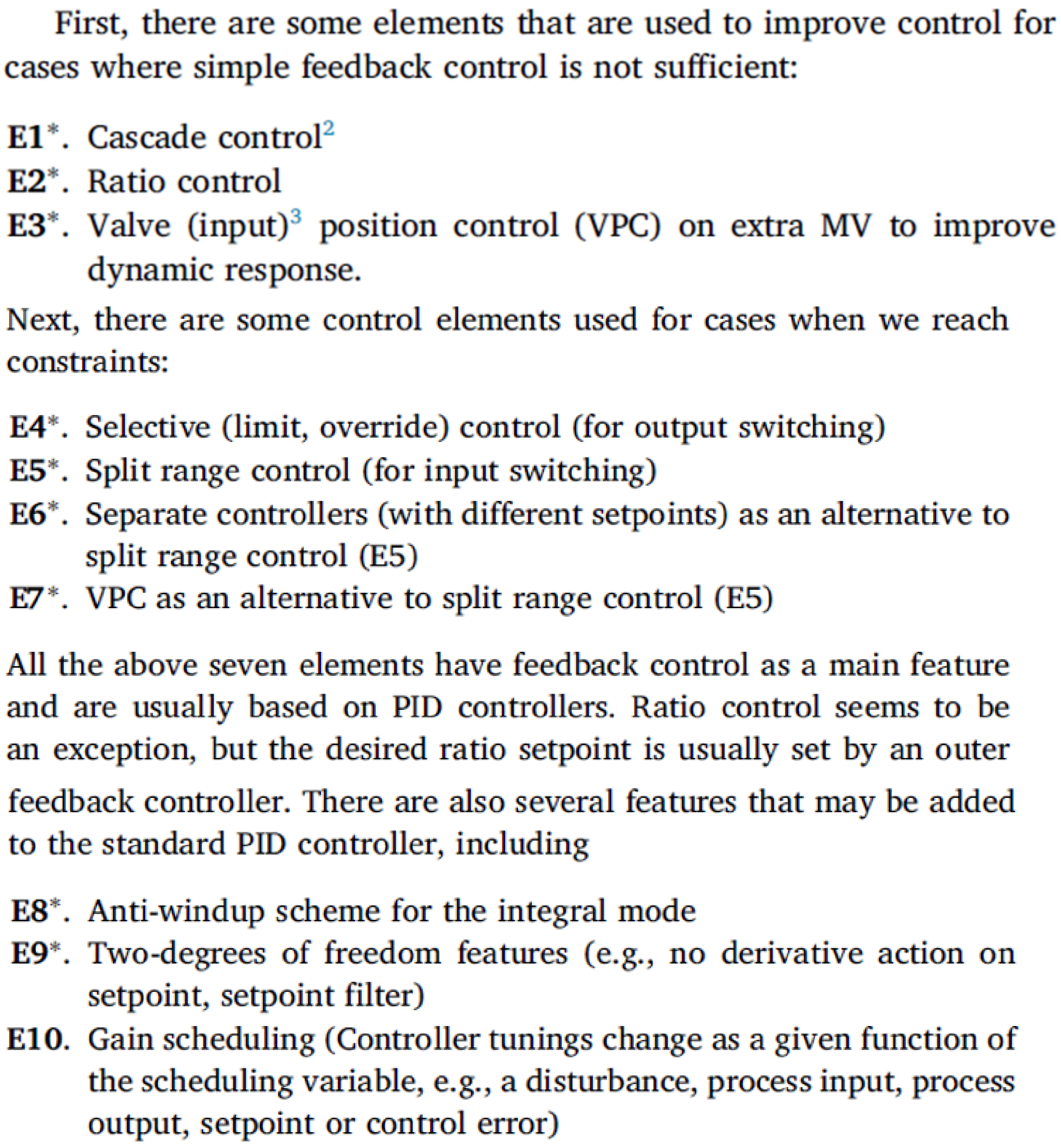
- Hint 1: According to Sigurd Skogestad
- Hint 2: All were in use around 1940

SOLUTION

1. PID controller, in particular, I-action
2. Cascade control
3. Ratio control

ARC: Standard Advanced control elements

Each element links a subset of inputs with a subset of outputs. Results in simple local design and tuning



First, there are some elements that are used to improve control for cases where simple feedback control is not sufficient:

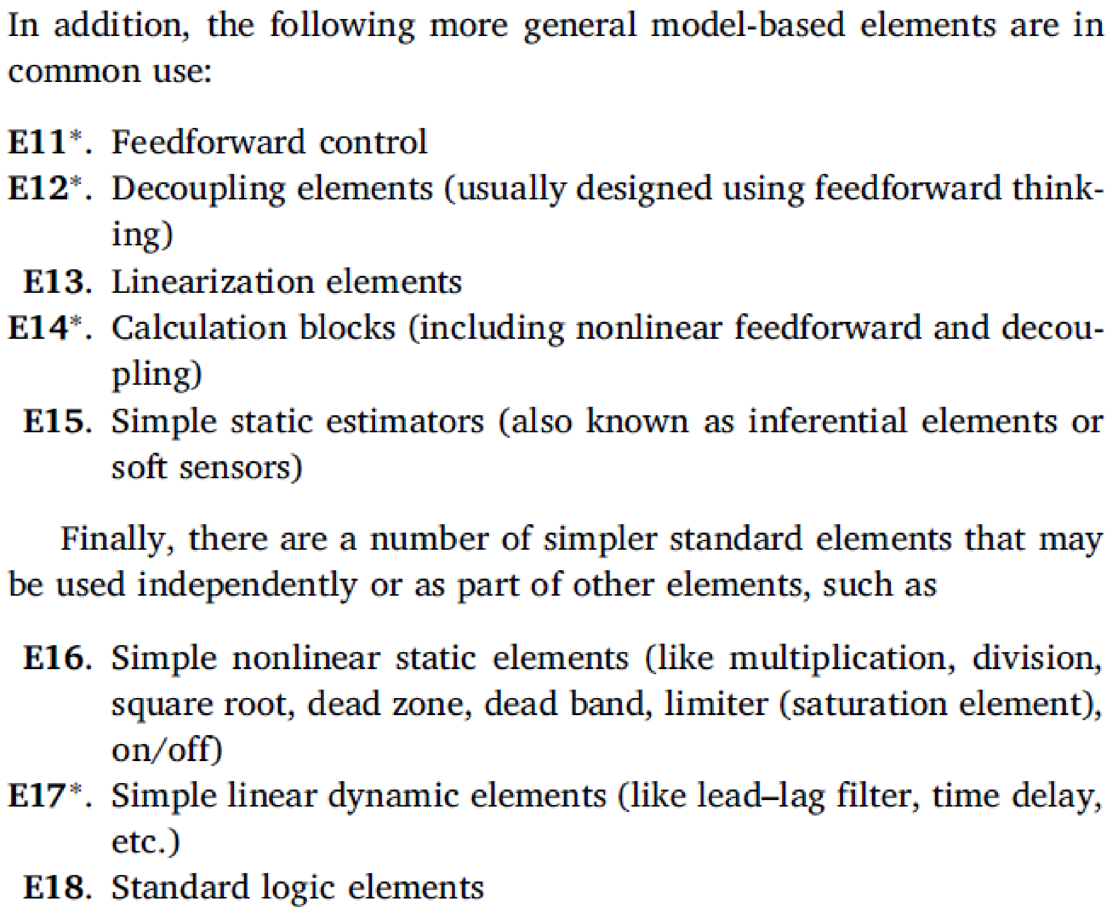
E1*. Cascade control[2]
E2*. Ratio control
E3*. Valve (input)[3] position control (VPC) on extra MV to improve dynamic response.

Next, there are some control elements used for cases when we reach constraints:

E4*. Selective (limit, override) control (for output switching)
E5*. Split range control (for input switching)
E6*. Separate controllers (with different setpoints) as an alternative to split range control (E5)
E7*. VPC as an alternative to split range control (E5)

All the above seven elements have feedback control as a main feature and are usually based on PID controllers. Ratio control seems to be an exception, but the desired ratio setpoint is usually set by an outer feedback controller. There are also several features that may be added to the standard PID controller, including

E8*. Anti-windup scheme for the integral mode
E9*. Two-degrees of freedom features (e.g., no derivative action on setpoint, setpoint filter)
E10. Gain scheduling (Controller tunings change as a given function of the scheduling variable, e.g., a disturbance, process input, process output, setpoint or control error)

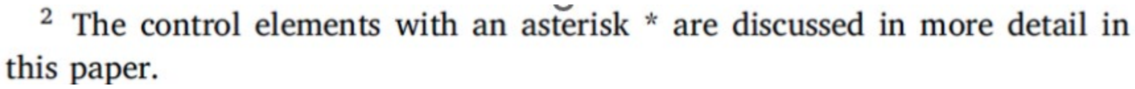


In addition, the following more general model-based elements are in common use:

E11*. Feedforward control
E12*. Decoupling elements (usually designed using feedforward thinking)
E13. Linearization elements
E14*. Calculation blocks (including nonlinear feedforward and decoupling)
E15. Simple static estimators (also known as inferential elements or soft sensors)

Finally, there are a number of simpler standard elements that may be used independently or as part of other elements, such as

E16. Simple nonlinear static elements (like multiplication, division, square root, dead zone, dead band, limiter (saturation element), on/off)
E17*. Simple linear dynamic elements (like lead–lag filter, time delay, etc.)
E18. Standard logic elements



[2] The control elements with an asterisk * are discussed in more detail in this paper.

ARC = advanced reguklatory control

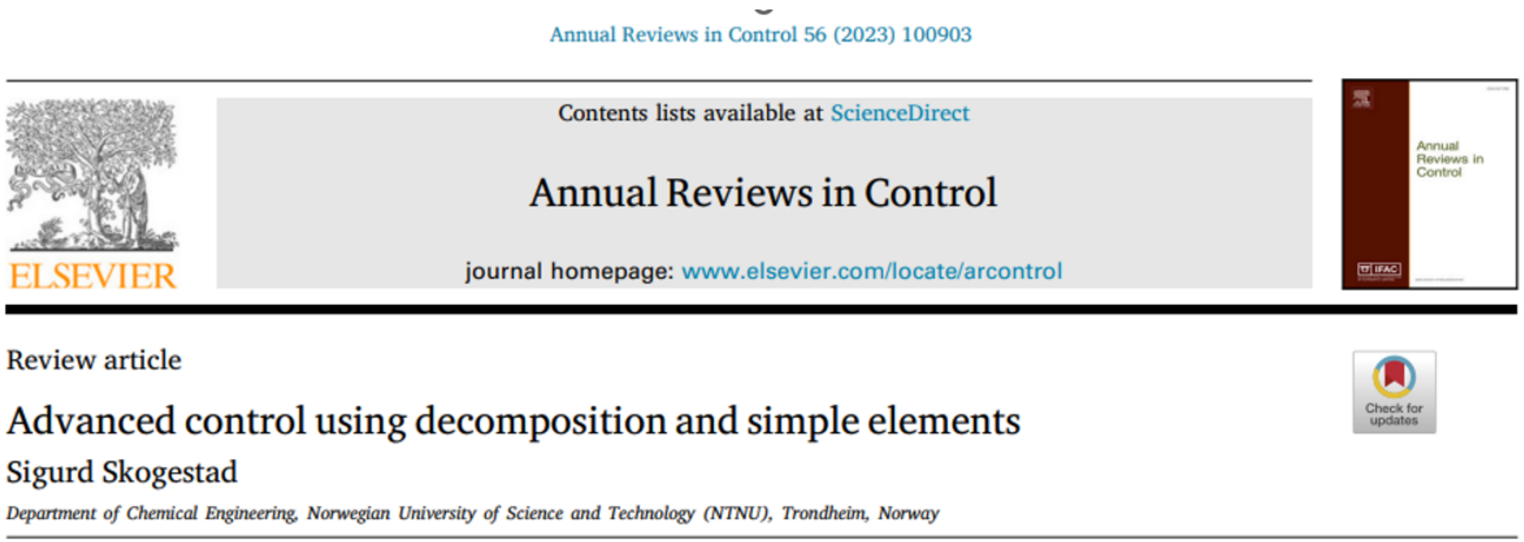
Sigurd Skogestad, "Advanced control using decomposition and simple elements". Annual Reviews in Control, vol. 56 (2023), Article 100903 (44 pages).

How design standard ARC elements?

- Industrial literature (e.g., Shinskey).

  Many nice ideas. But not systematic. Difficult to understand reasoning

- Academia: Very little work
  - I feel alone



Annual Reviews in Control 56 (2023) 100903

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

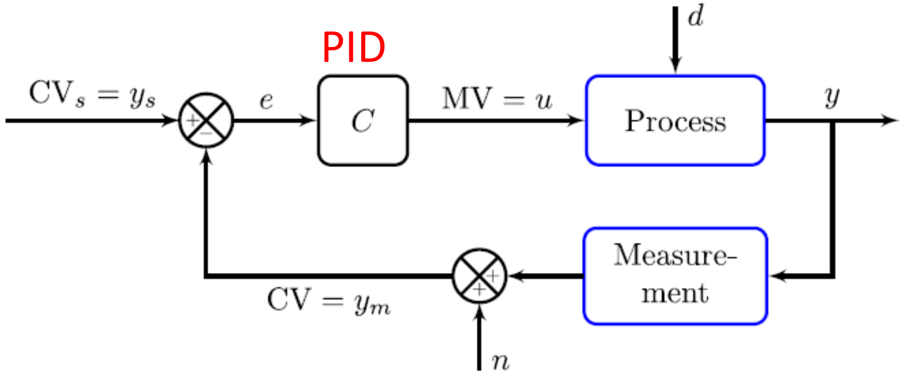
Review article

Advanced control using decomposition and simple elements

Sigurd Skogestad

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Most basic element: Single-loop PID control (E0)



MV-CV Pairing. Two main pairing rules:

1. ***"Pair-close rule" :*** *The MV should have a large, fast, and direct effect on the CV.*
2. ***"Input saturation rule":*** *Pair a MV that may saturate with a CV that can be .given up (when the MV saturates).*
   - *Exception: Have extra MV so we use MV-MV switching (e.g., split range control)*

Additional rule for interactive systems:

3. ***" RGA-rule".*** *Avoid pairing on negative steady-state RGA-element.*

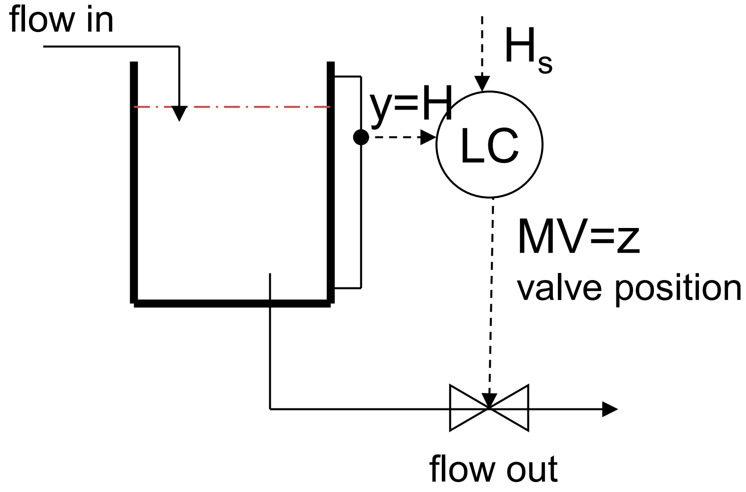
E1. Cascade control

y1=primary output (given setpoint)
y2=secondary output (adjustable setpoint)

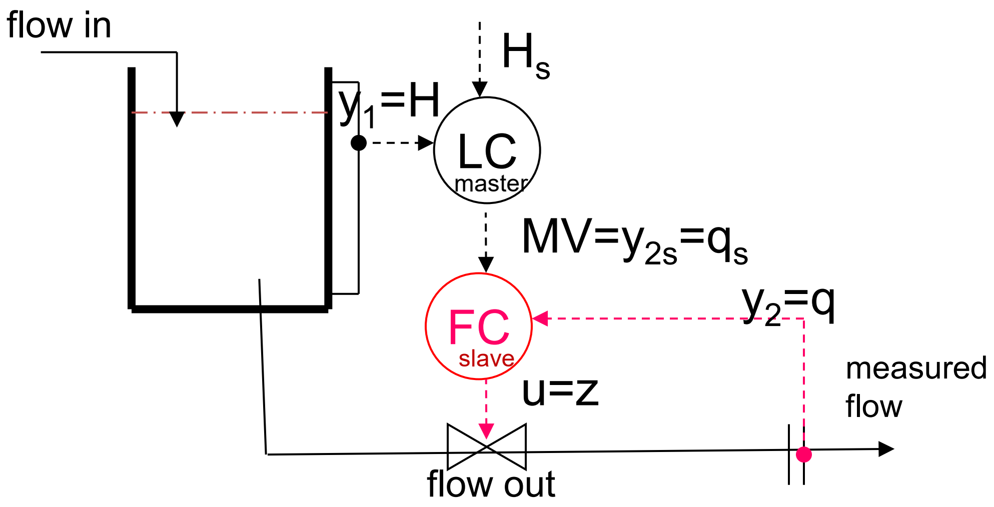
Idea: **make use of extra "local" output measurement (y_2)**
Implementation: Controller ("master") gives setpoint to another controller ("slave")

- **Example: Flow controller on valve (very common!)**

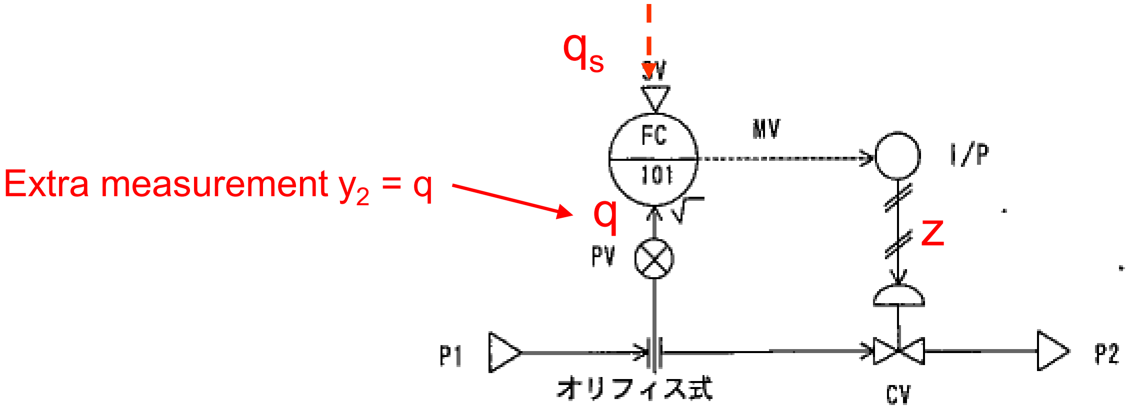
WITHOUT CASCADE



WITH CASCADE



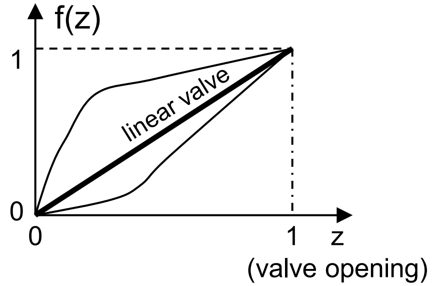
What are the benefits of adding a flow controller (inner cascade)?



$$\text{Flow rate: } q = C_v f(z)\sqrt{\frac{p_1-p_2}{\rho}} \qquad [\mathrm{m}^3/\mathrm{s}]$$

1. Counteracts nonlinearity in valve, f(z)
   - High gain in inner loop eliminates nonlinearity inside inner loop
   - With fast flow control we can assume q = q_s

2. Eliminates effect of disturbances in p1 and p2
   (FC reacts faster than outer level loop)



Tuning cascade control

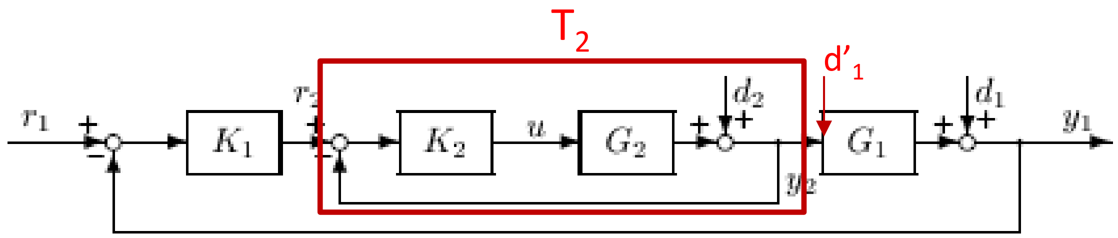


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

First tune fast inner controller K_2 ("slave")

Design K_2 based on model G_2

Select τ_{c2} based on effective delay in G_2

Nonlinearity: Gain variations (in G_2) translate into variations in actual time constant τ_{C2}

Then with slave closed, tune slower outer controller K_1 ("master"):

Transfer function for inner loop (from y_{2s} to y_2): $T_2 = G_2 K_2/(1+G_2 K_2)$

Design K_1 based on model $G_1' = T_2 * G_1$

Can often set $T_2=1$ if inner loop is fast!

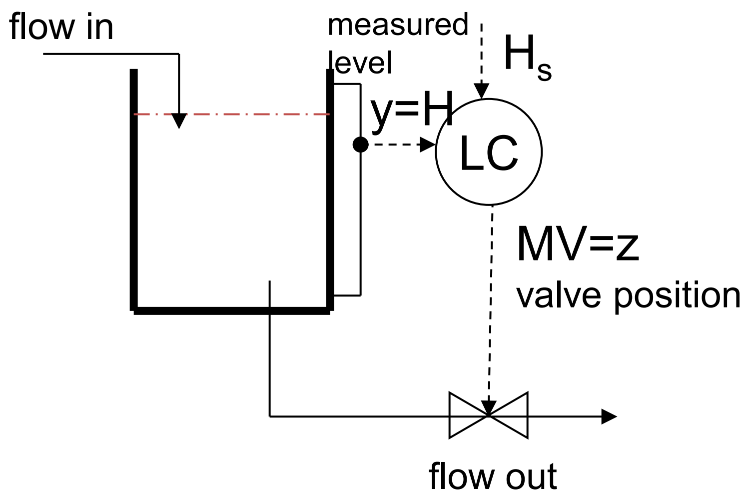
- Alternatively, $T_2 \approx e^{-\Theta_2 s}/(\tau_{c2}s+1) \approx e^{-(\Theta_2+\tau_{c2})s}$

Typical choice: $\tau_{c1} = \sigma\ \tau_{c2}$ where time scale separation $\sigma = 4\ to\ 10$.

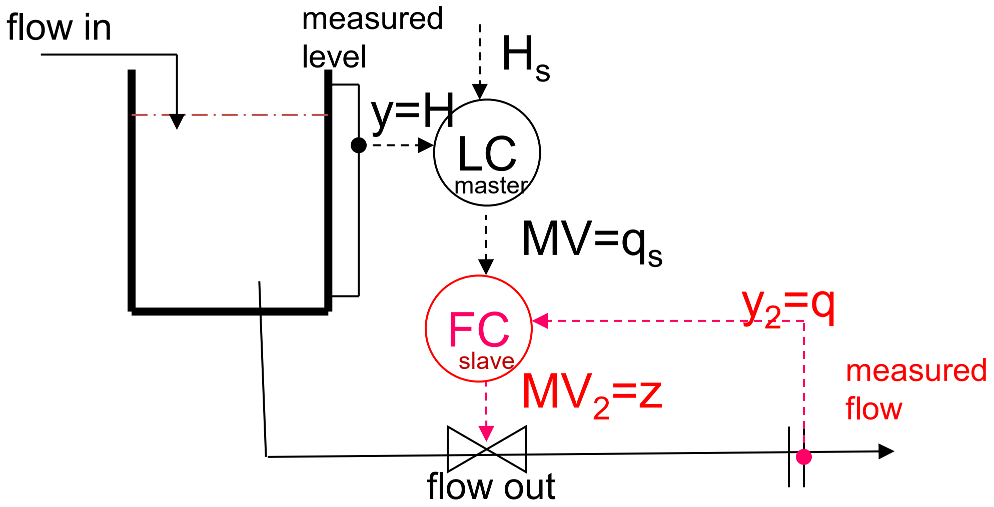
Linearization of valve using cascade control

- Benefits: 1. Local distrurbance rejection, 2. Linearization
- Does nonlinearity disappear?

WITHOUT CASCADE



WITH CASCADE (2 controllers)



No, it moves to the time constant for slave loop
– OK if we we have time scale separation between master and slave

Nonlinear valve with varying gain k_2: $G_2(s) = k_2(z) / (\tau_2 s+1)$

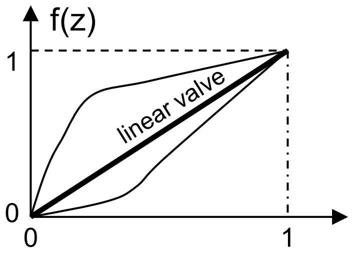
- Slave (flow) controller K_2: PI-controller with gain K_{c2} and integral time $\tau_I = \tau_2$ (SIMC-rule). Get

$$L_2 = K_2(s)G_2(s) = \frac{K_{c2}k_2}{\tau_2 s}$$

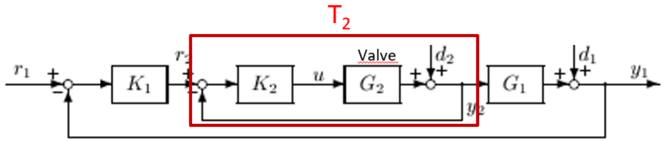
- With slave controller: Transfer function T_2 from y_{2s} to y_2 (as seen from master loop):

$$T_2 = L_2/(1+L_2) = 1/(\tau_{C2}\ s + 1), \text{ where } \tau_{C2} = \tau_2/(k_2\ K_{c2})$$

- Linearization: Gain for T_2 is always 1 (independent of k_2) because of intergal action in the inner (slave) loop
- But: Gain variation in k_2 (inner loop) translates into variation in closed-loop time constant τ_{C2}. This may effect the master loop



$k_2(z)$ = slope = df/dz



G_1T_2 = «Process» for tuning master controller K_1

Time scale separation is needed for cascade control to work well

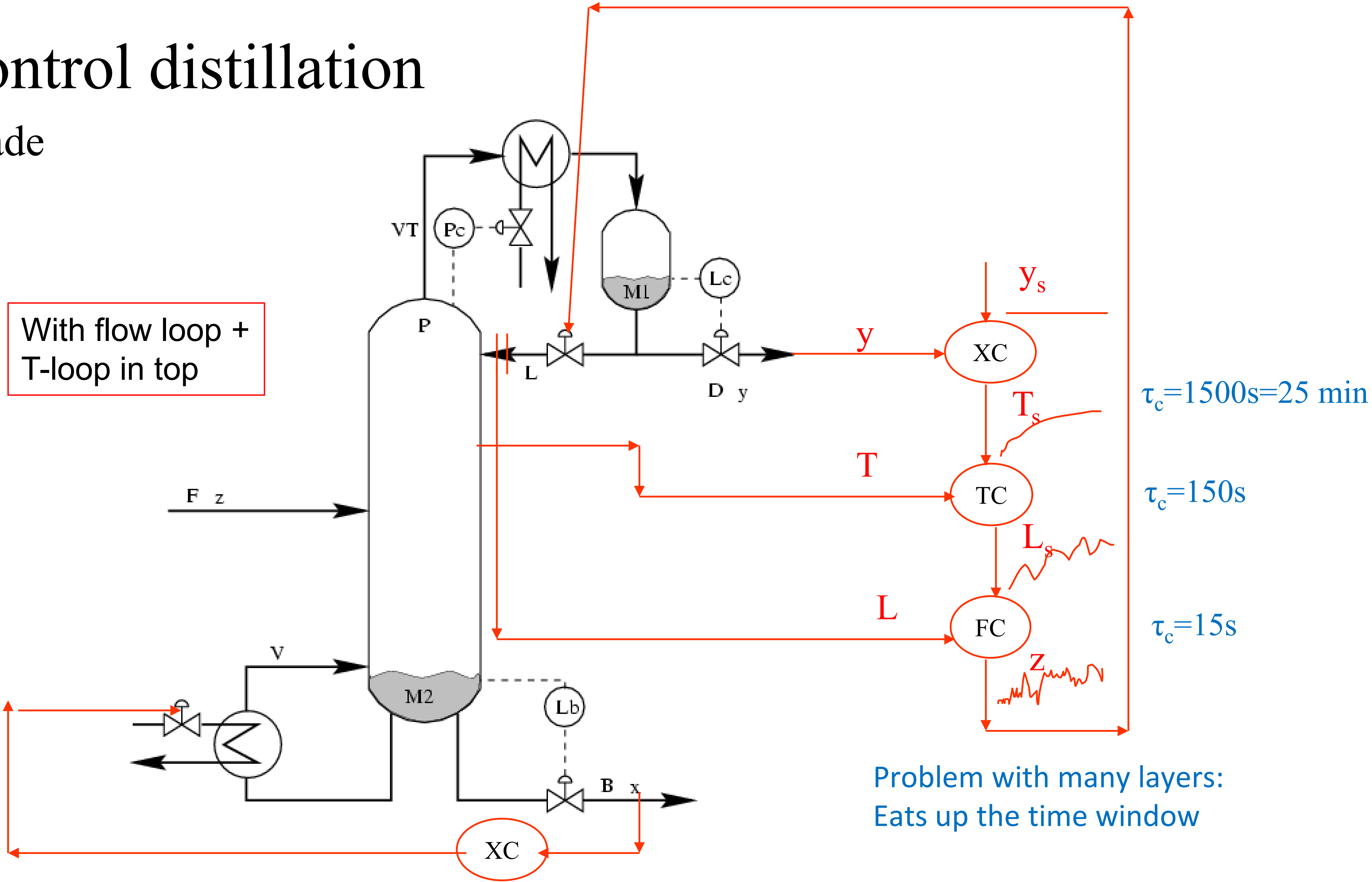
- Inner loop (slave) should be at least 4 times* faster than the outer loop (master)
  - This is to make the two loops (and tuning) independent.
  - Otherwise, the slave and master loops may start interacting
    - The fast slave loop is able to correct for local disturbances, but the outer loop does not «know» this and if it's too fast it may start «fighting» with the slave loop.
- Often recommend 10 times faster, $\sigma \equiv \frac{\tau_{c1}}{\tau_{c2}} = 10.$
  - A high σ is robust to gain variations (in both inner and outer loop)
  - The reason for the upper value ($\sigma = 10$) is to avoid that control gets too slow, especially if we have many layers

---

* Shinskey (Controlling multivariable processes, ISA, 1981, p.12)

Cascade control distillation

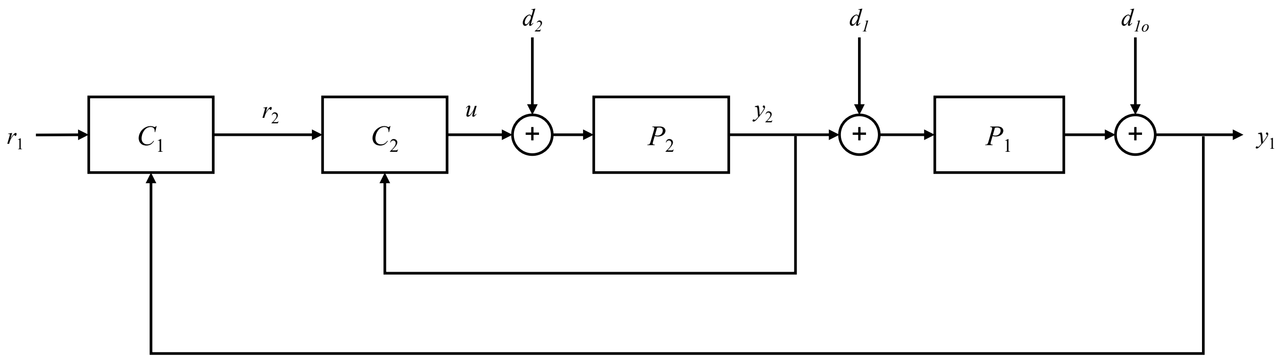
3 layers of cascade



Problem with many layers:
Eats up the time window

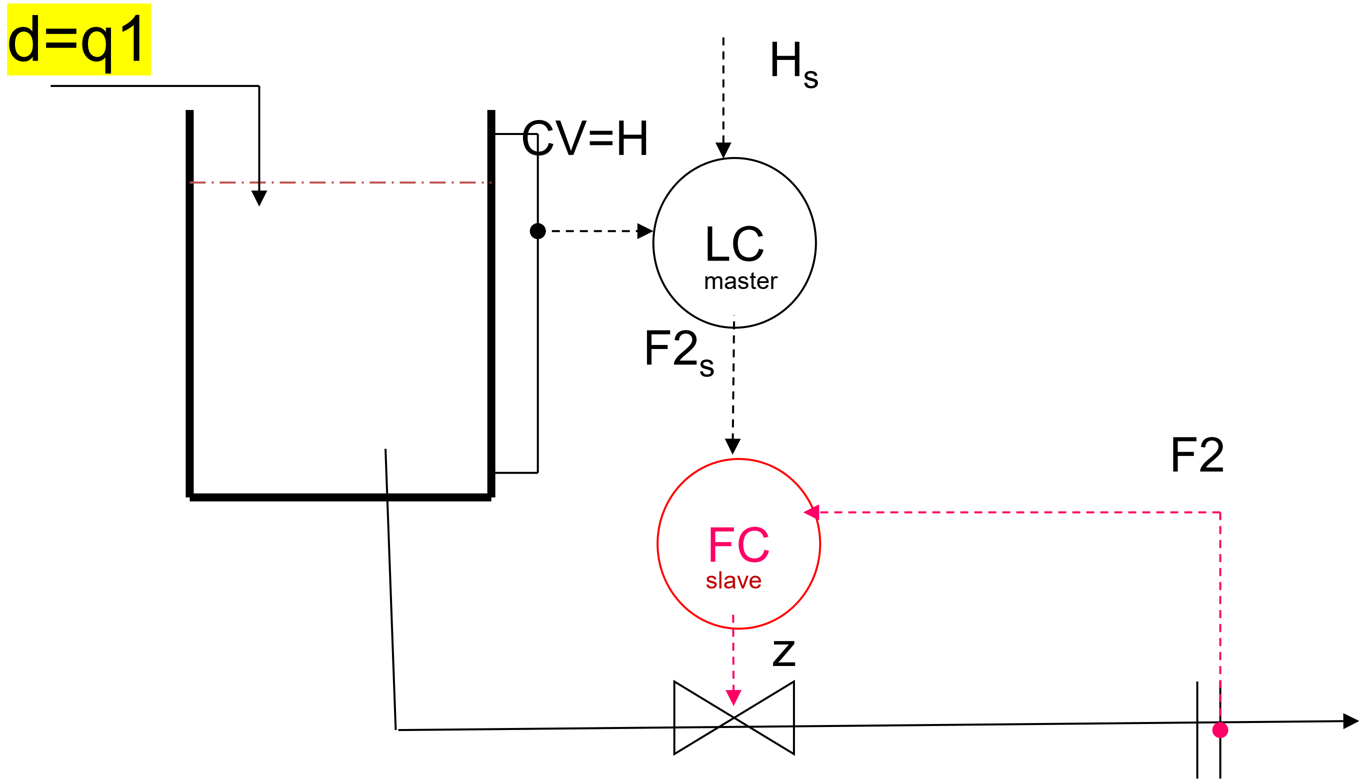
Cascade control block diagram

- Which disturbances motivate the use of cascade control?

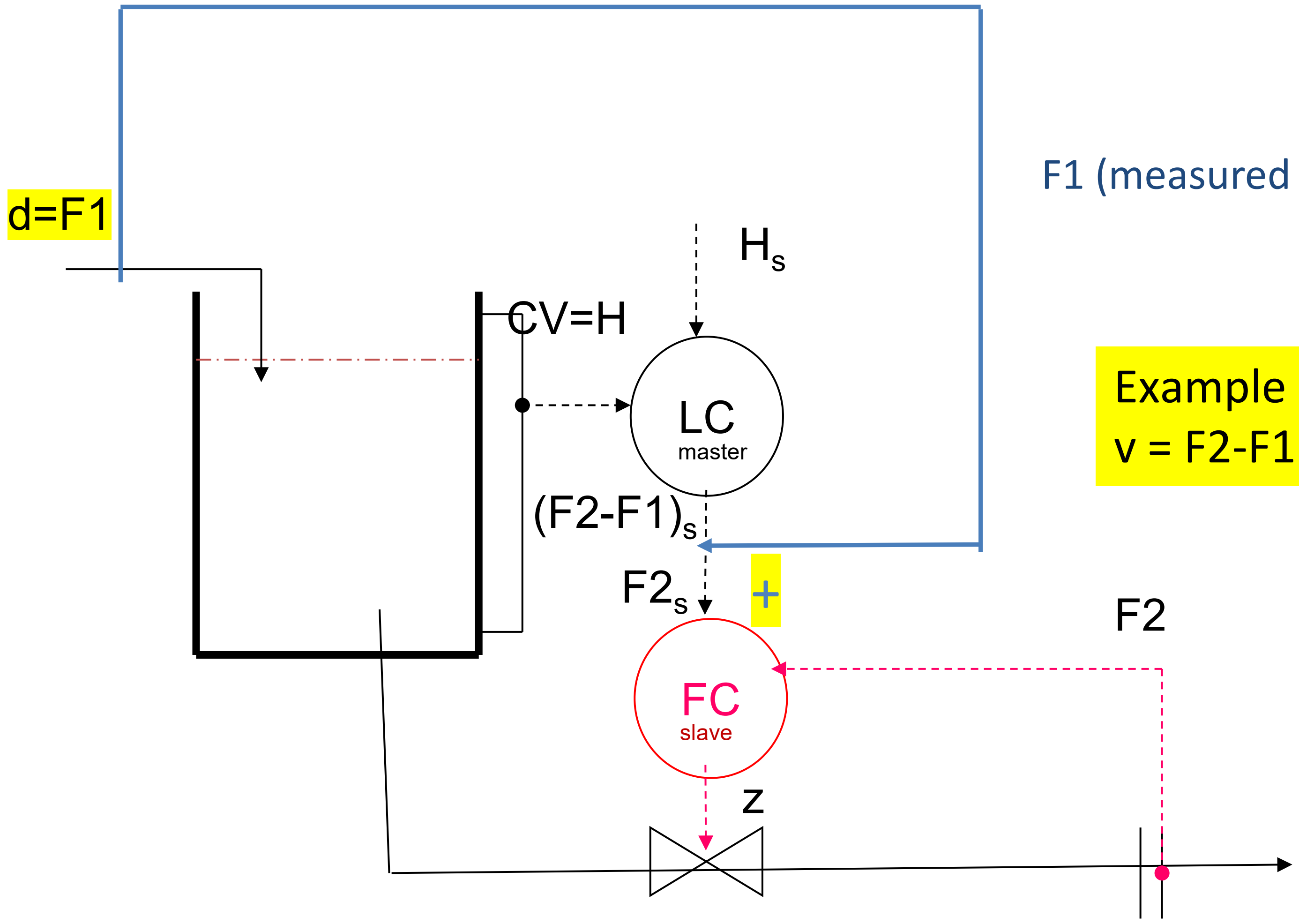


Answer: d_2

Quiz: How can we add feedforward?



Solution: How can we add feedforward?



F1 (measured flow disturbance)

Example of input transformation.
v = F2-F1 = u - d

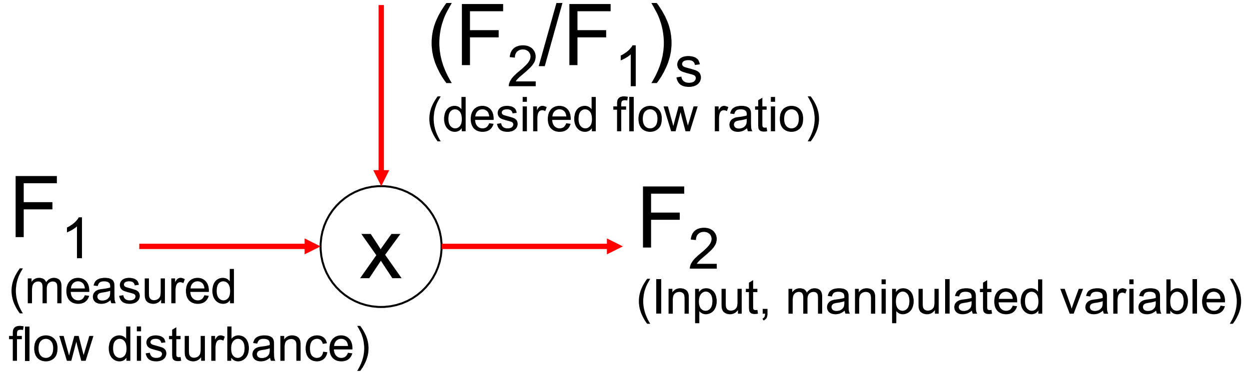
Ratio control

Special case of to feedforward, but don’t need model, just process insight.

Always use for mixing streams

- Note: Disturbance needs to be a flow (or more generally an extensive variable)

Use multiplication block (x):



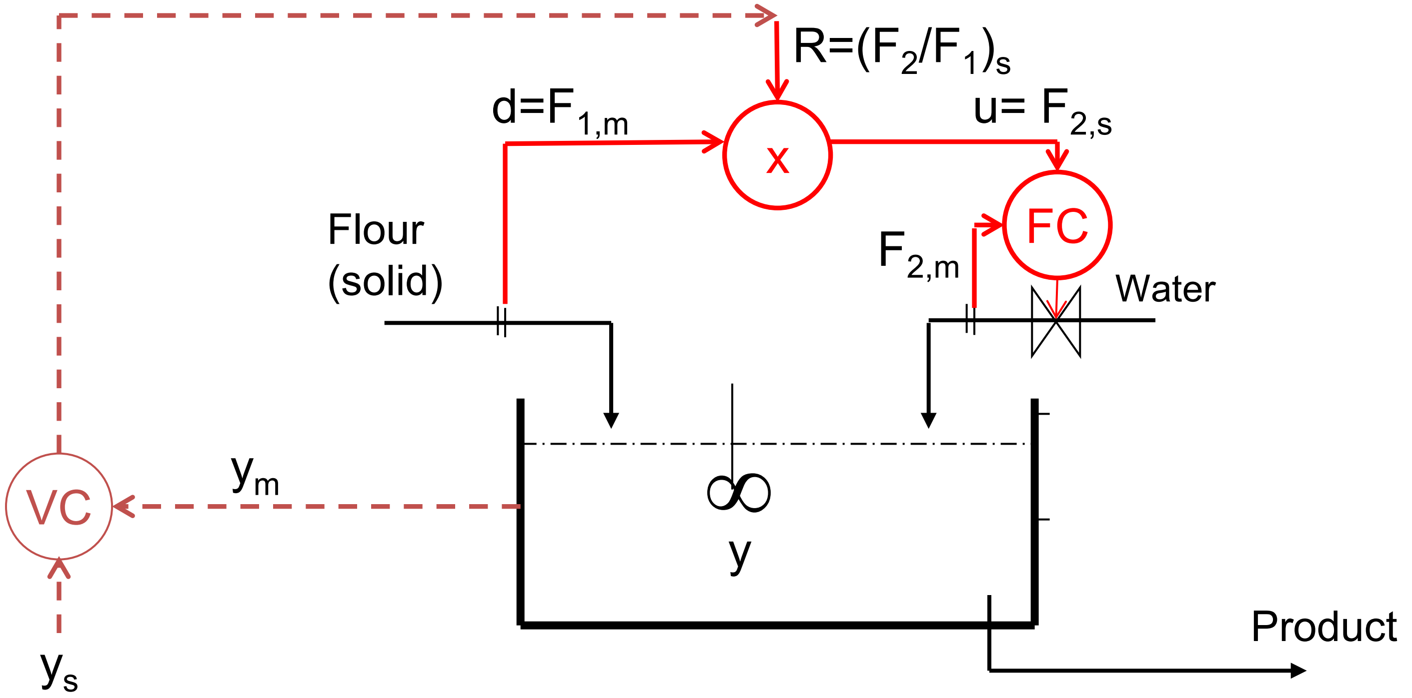
“Measure disturbance ($d=F_1$) and adjust input ($u=F_2$) such that ratio is at given value $(F_2/F_1)_s$”

Usually: Combine ratio (feedforward) with feedback

***Example cake baking**: Use recipe (ratio control = feedforward), but a good cook adjusts the ratio to get desired result (feedback)*

EXAMPLE: CAKE BAKING MIXING PROCESS

RATIO CONTROL with outer feedback (to adjust ratio setpoint)

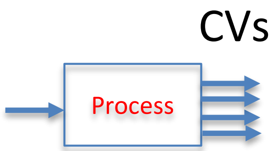


Feedback correction («trim»)

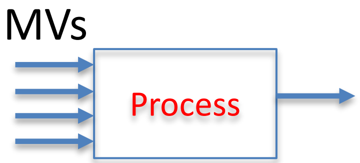
Want to control: Viscosity y [cP]
(or any intensive quality variable, like c, ρ or T)

Constraint switching (because it is optimal at steady state)

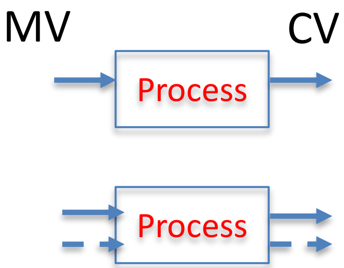
- CV-CV switching
  - Control one CV at a time



- MV-MV switching
  - Use one MV at a time



- MV-CV switching
  - MV saturates so must give up CV
  1. Simple («do nothing»)
  2. Complex (repairing of loops)



MV = manipulated variable
CV = controlled variable

MV-MV switching

- One CV, many MVs (to cover whole steady-state range because primary MV may saturate)*
- Use one MV at a time

Three alternatives:

Alt.1 Split-range control (SRC)

- Plus Generalized SRC (baton strategy)

Alt.2 Several controllers (one for each MV) with different setpoints for the single CV

Alt.3 Valve position control (VPC)

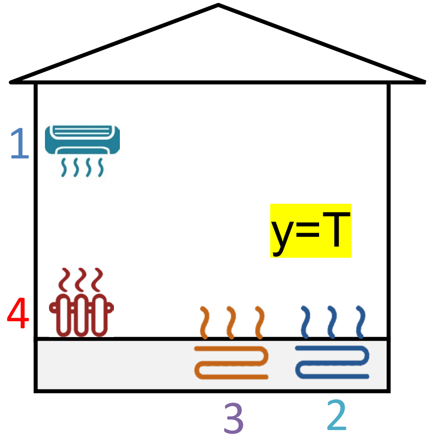
Which is best? It depends on the case!

***Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control, Adriana Reyes-Lua Cristina Zotica, Sigurd Skogestad, Adchem Conference, Shenyang, China. July 2018 ,**

Example MV-MV switching

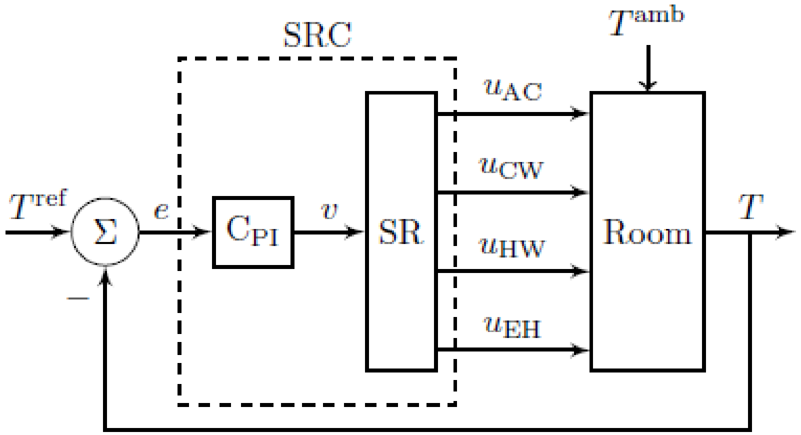
- Break and gas pedal in a car
- Use only one at a time,
- «manual split range control»

Example split range control: Room temperature with 4 MVs



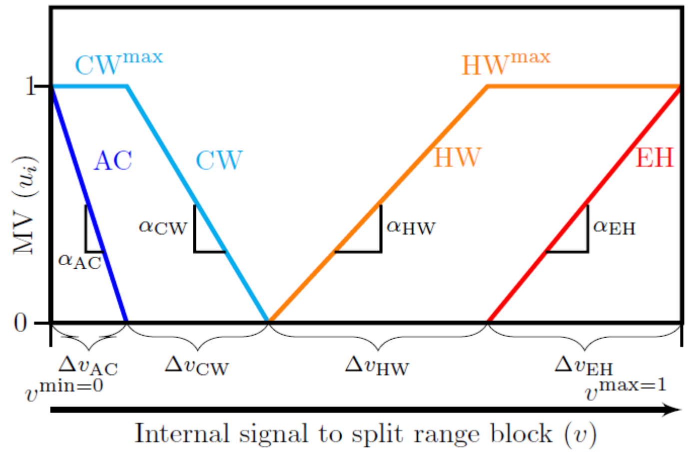
MVs (two for summer and two for winter):

1. AC (expensive cooling)
2. CW (cooling water, cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)

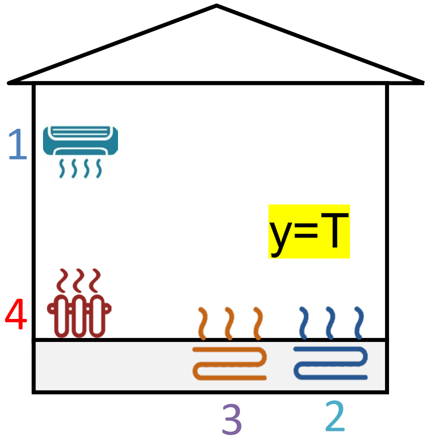


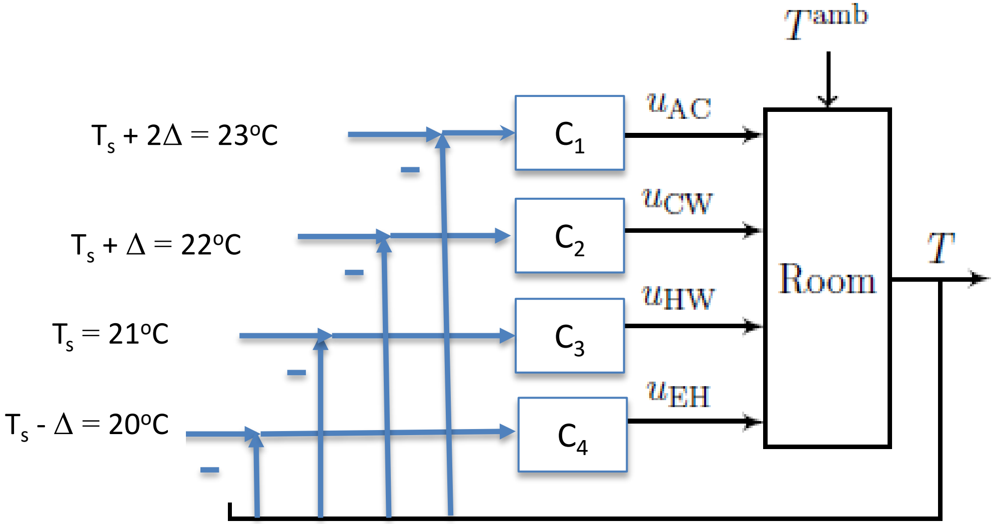
C_{PI} – same controller for all inputs (one integral time)
But get different gains by adjusting slopes α in SR-block

SR-block:



Alternative 2: Multipliple Controllers with different setpoints





Disadvantage (comfort):

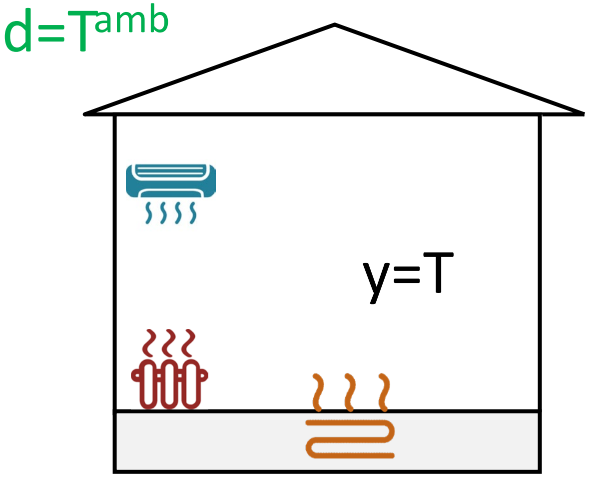
- Different setpoints

Advantage (economics) :

- Different setpoints (energy savings)

Simulation Room temperature

- Dashed lines: SRC (E5)
- Solid lines: Multiple controllers (E6)



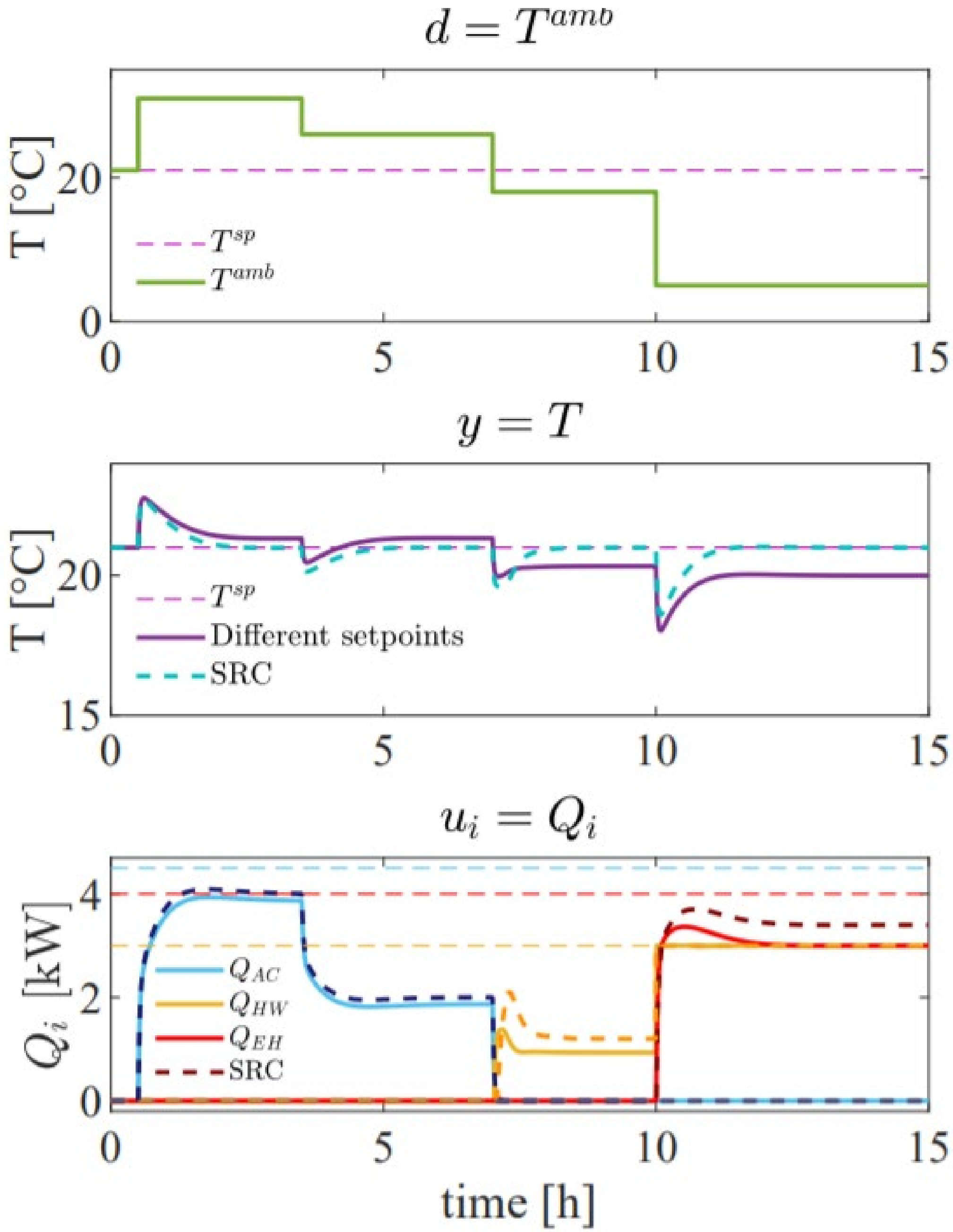
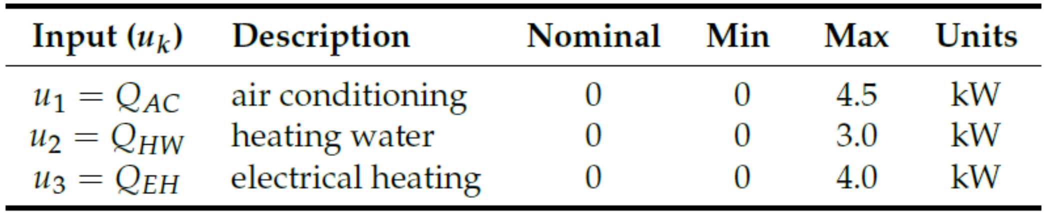


Table 1. Ranges for available inputs (u_k).



Input (u_k)	Description	Nominal	Min	Max	Units
$u_1 = Q_{AC}$	air conditioning	0	0	4.5	kW
$u_2 = Q_{HW}$	heating water	0	0	3.0	kW
$u_3 = Q_{EH}$	electrical heating	0	0	4.0	kW

SRC = split range control

A Reyes-Lua, S Skogestad. Multiple-Input Single-Output Control for Extending the Steady-State Operating Range - Use of Controllers with Different Setpoints. Processes 7 (12), 941 (2019)

Summary MV-MV switching

- Need several MVs to cover whole <u>steady-state</u> range (because primary MV may saturate)*
- Note that we only want to use one MV at the time.

Alt.1 Split-range control (one controller) (E5)

  - Advantage: Easy to understand because SR-block shows clearly sequence of MVs
  - Disdvantages: (1) Need same tunings (integral time) for all MVs . (2) May not work well if MV-limits inside SR-block change with time, so: Not good for MV-CV switching

Alt.2 Several controllers with different setpoints (E6)

  - Advantages: 1. Simple to implement, do not need to keep track of MVs. 2. Can have independent tunings. .
  - Disadvantages: Temporary loss of control during switching. Setpoint varies (which can be turned into an advantage in some cases)

Alt.3 Valve position control (E7)

  - Advantage: Always use “primary” MV for control of CV (avoids repairing of loops)
  - Disadvantages: Gives some loss, because primary MV always must be used (cannot go to zero).

Which is best? It depends on the case!

***Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control, Adriana Reyes-Lua Cristina Zotica, Sigurd Skogestad, Adchem Conference, Shenyang, China. July 2018** ,

Example MV-MV switching: Pressure control

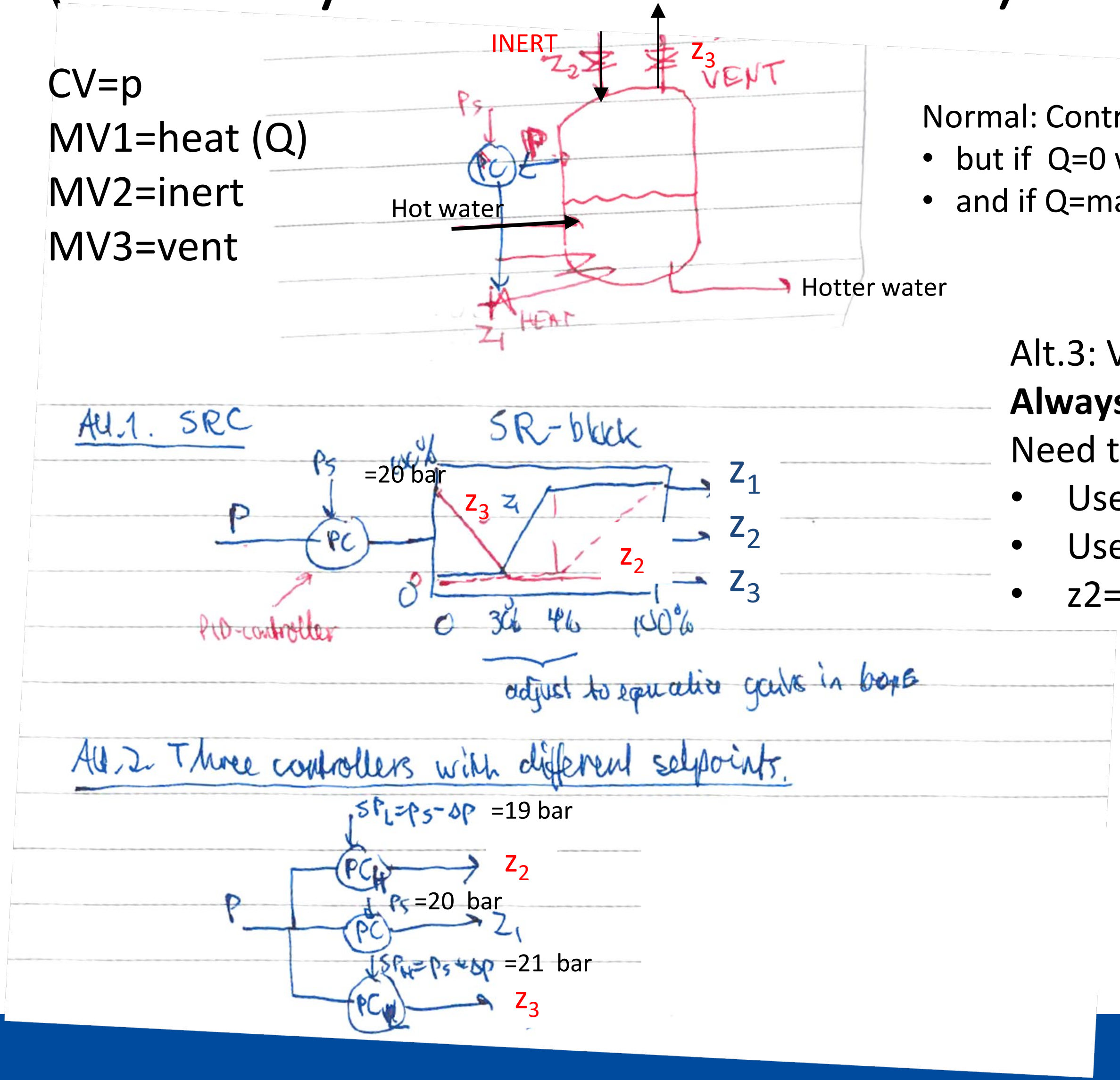
(Alt. 3 may be the best in this case)

CV=p

MV1=heat (Q)

MV2=inert

MV3=vent



Normal: Control CV=p using MV1=Q

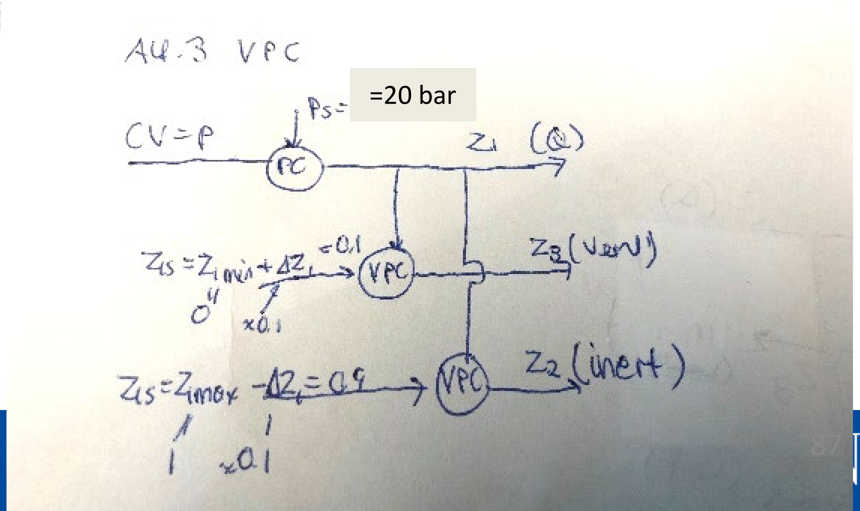
- but if Q=0 we must use MV3=vent
- and if Q=max we must use MV2=inert

Alt.3: VPC (z2 and z3 could here even be on/off valves)

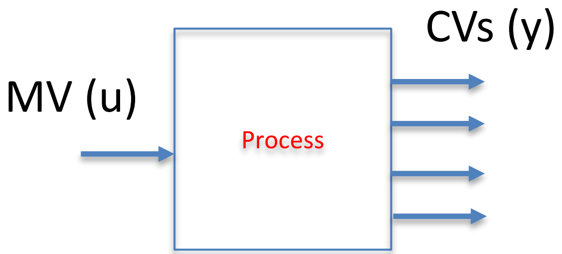
Always use Q (z1) to control p.

Need two VPC’s:

- Use vent (z3) to avoid Q small (z1=0.1)
- Use inert (z2) to avoid Q large (z1=0.9)
- z2=0 and z3=0 when 0.1<z1<0.9



CV-CV switching



- Only one input (MV) controls many outputs (CVs)
  - Typically caused by change in active constraint
- Always use MIN- or MAX-selector
  - Example: Control car speed (y_1) - but give up if too small distance (y_2) to car in front.

Example adaptive cruise control:
CV-CV switch followed by MV-MV switch

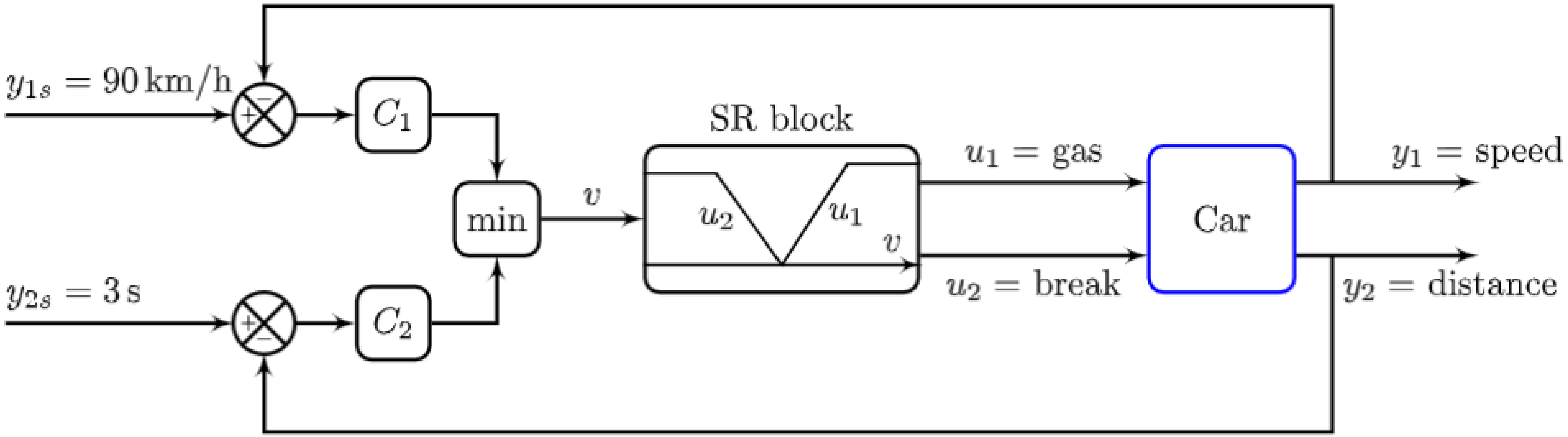
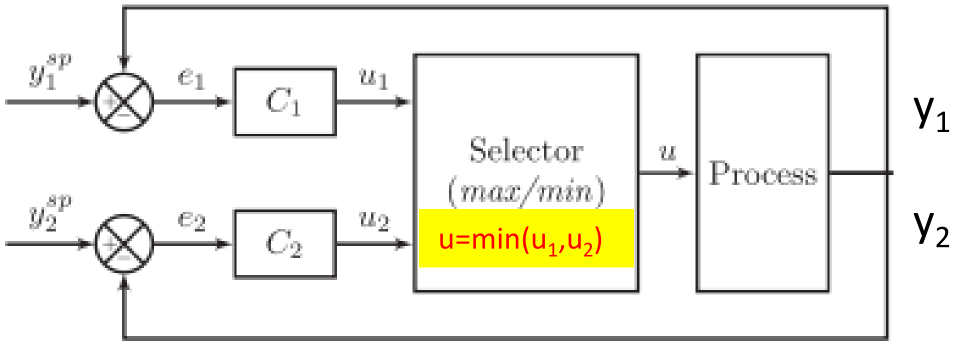


Fig. 31. Adaptive cruise control with selector and split range control.

Note: This is not Complex MV-CV switching, because then the order would be opposite.

Selector: One input (u), several outputs (y_1,y_2)



\> = MAX = HS

< = MIN = LS

- Many CVs paired with one MV, but only CV controlled at a time
- This requires output-output (CV-CV) switching: Use selector*
- Note: The selector is usually on the input u, even though the setpoint/constraint is on the output y
- Sometimes called "override"
  - OK name for temporary dynamic fix, but otherwise a bit misleading**
- Selectors work well, but require pairing each constraint with a given input (not always possible)

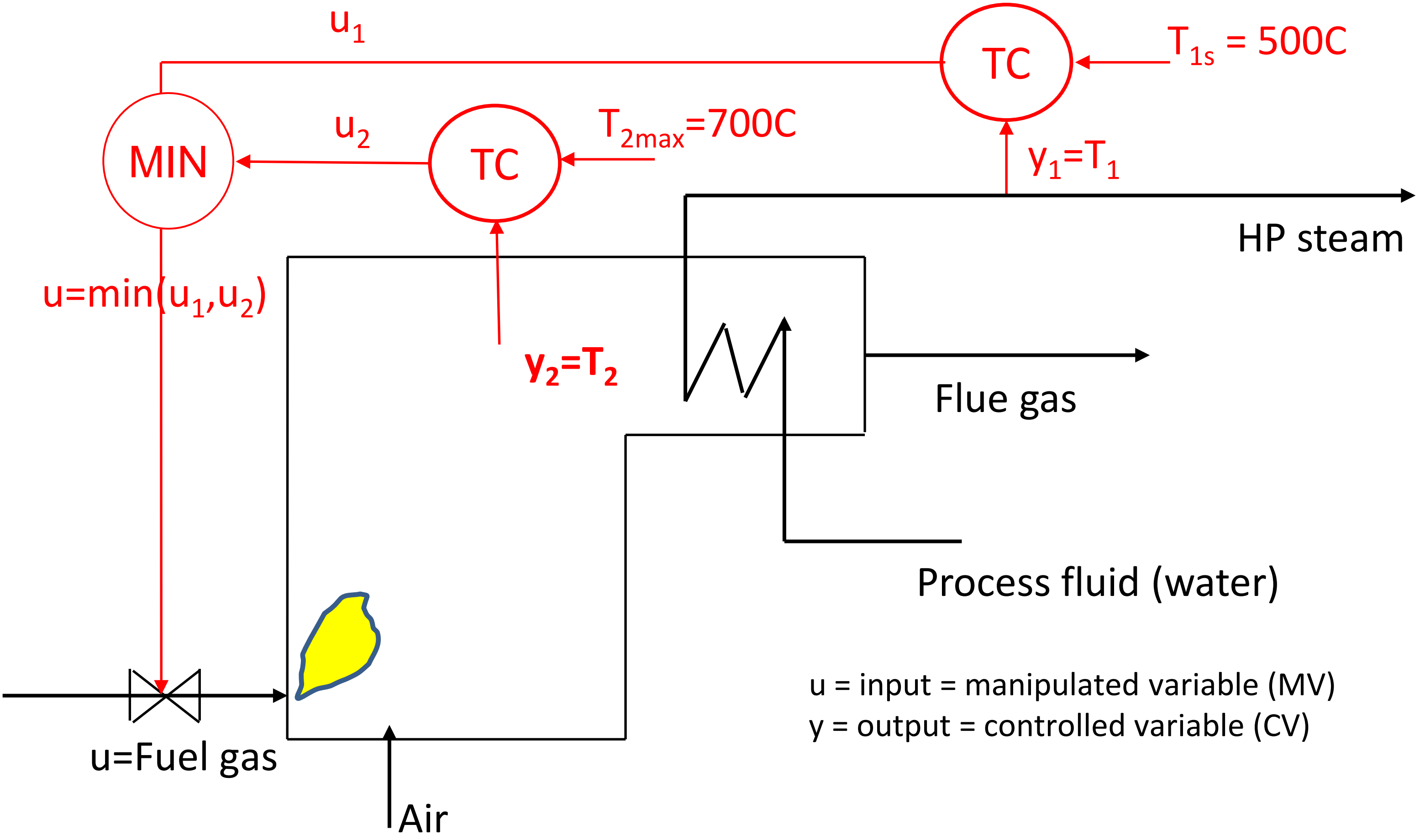
*Only option for CV-CV switching. Well, not quite true: Selectors may be implemented in other ways, for example, using «if-then»-logic.

** I prefer to use the term «override» for undesirable temporary (dynamic) switches, for example, to avoid overflowing a tank dynamically. Otherwise, it's CV-CV switching

Furnace control with safety constraint

Input (MV)
  u = Fuel gas flowrate
Output (CV)
  y_1 = process temperature T_1
    (desired setpoint or max constraint)
  y_2 = furnace temperature T_2
    (T2max= 700C)

Rule: Use min-selector for constraints that are satisfied with a small input



u = input = manipulated variable (MV)
y = output = controlled variable (CV)

Design of selector structure

Rule 1 (max or min selector)

- Use max-selector for constraints that are satisfied with a large input
- Use min-selector for constraints that are satisfied with a small input

Rule 2 (order of max and min selectors):

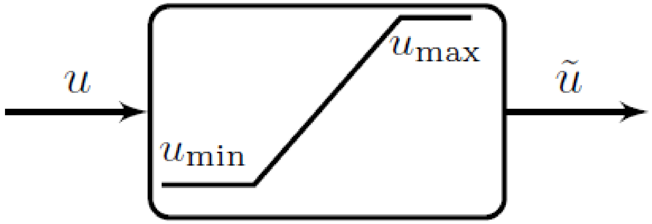
- If need both max and min selector: Potential infeasibility (conflict)
- Order does not matter if problem is feasible
- If infeasible: Put highest priority constraint at the end

“Systematic design of active constraint switching using selectors.” Dinesh Krishnamoorthy , Sigurd Skogestad. Computers & Chemical Engineering, Volume 143, (2020)
“Advanced control using decomposition and simple elements”. Sigurd Skogestad. Annual Reviews in Control, Volume 56, 100903 (2023)

Valves have “built-in” selectors

Rule 3 (a bit opposite of what you may guess)

- A closed valve (u_{min}=0) gives a “built-in” max-selector (to avoid negative flow)
- An open valve (u_{max}=1) gives a “built-in” min-selector
  - So: Not necessary to add these as selector blocks (but it will not be wrong).
  - The “built-in” selectors are never conflicting because cannot have closed and open at the same time
  - Another way to see this is to note that a valve works as a saturation element



Saturation element may be implemented in three other ways (equivalent because never conflict)

1. Min-selector followed by max-selector
2. Max-selector followed by min-selector
3. Mid-selector

$$\tilde{u} = \max(u_{min}, \min(u_{max}, u)) = \min(u_{max}, \max(u_{min}, u)) = \text{mid}(u_{min}, u, u_{max})$$

“Advanced control using decomposition and simple elements”. Sigurd Skogestad. Annual Reviews in Control, Volume 56, 100903 (2023)

MV-CV switching (because reach constraint on MV)

- **Simple CV-MV switching**
  - Don’t need to do anything if we followed the *Input saturation rule:*
  - *“Pair a MV that may saturate with a CV that can be given up (when the MV saturates)”*

Example «simple» MV-CV switching (no selector)

Anti-surge control (= min-constraint on F)

Minimize recycle (MV=z) subject to

CV= $F \geq F_{min}$

MV= z ≥ 0

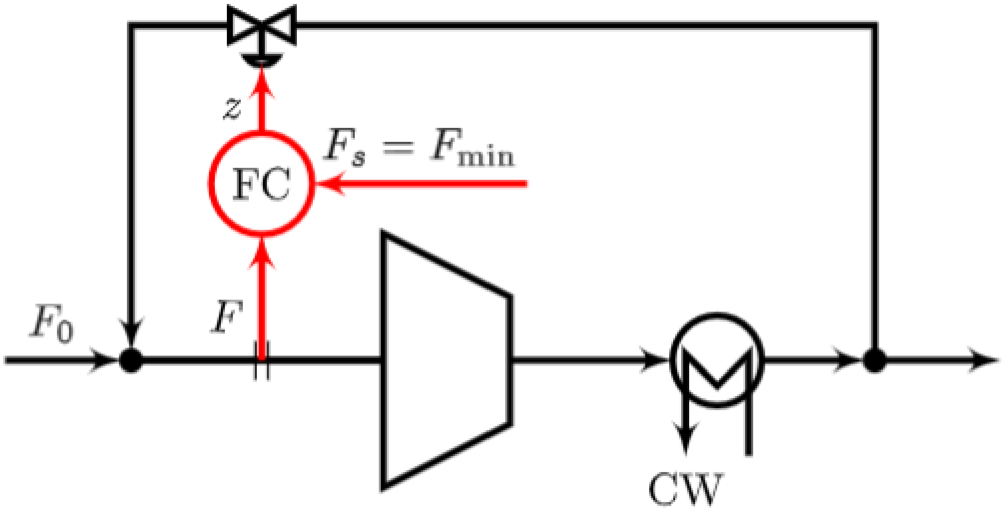
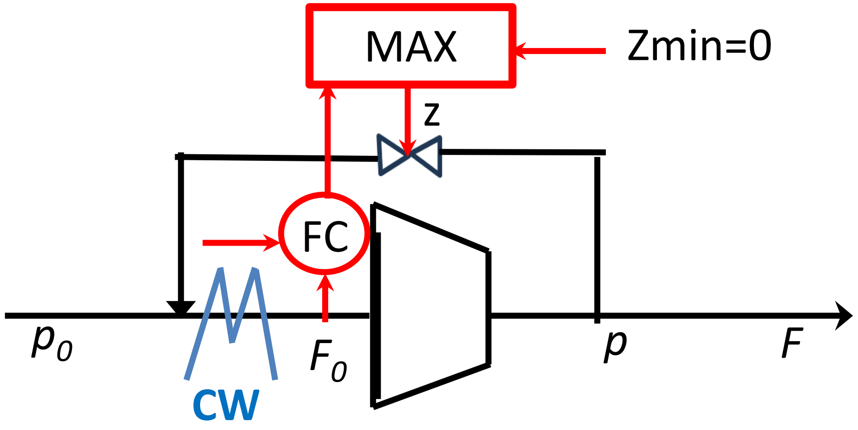


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding anti-windup to the controller. Note that the valve has a “built in” max selector.

- No selector required, because MV=z has a «built-in» max-selector at z=0.
- Generally: «Simple» MV-CV switching (with no selector) can be used if we satisfy the input saturation rule: «*Pair a MV that may saturate with a CV that can be given up (when the MV saturates at z=0)”*

QUIZ Compressor control

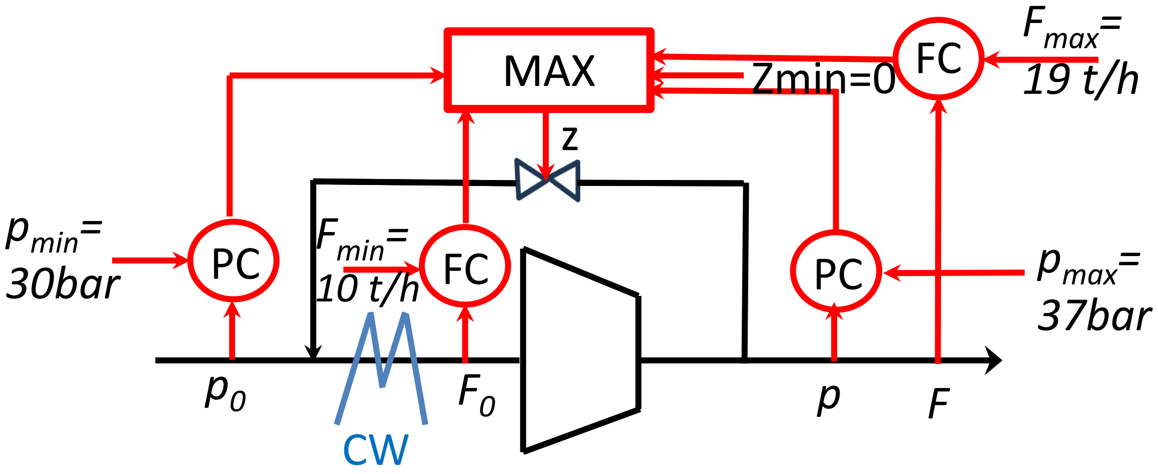


Suggest a solution which achieves

- $p < p_{max} = 37$ bar   *(max delivery pressure)*
- $P_0 > p_{min} = 30$ bar *(min. suction pressure)*
- $F < F_{max} = 19$ t/h   *(max. production rate)*
- $F_0 > F_{min} = 10$ t/h *(min. through compressor to avoid surge)*

All these 4 constraints are satisfied by a large z
-> MAX-selector

SOLUTION



Inventory control

- Very important decison for plantwide control:
  - Location of TPM

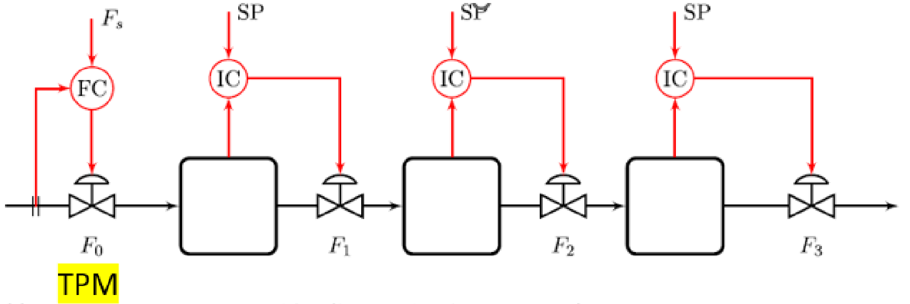
- TPM = Throughput manipulator

  = *Gas Pedal = Variable used for setting the throughput/production rate (for the entire process).*

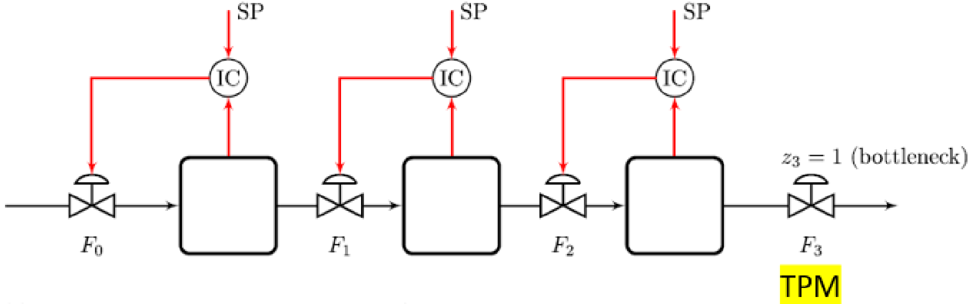
- ***Radiating rule:** Inventory control should be "radiating" around a given flow (TPM).*

Inventory control for units in series

Radiating rule:
Inventory control should be "radiating" around a given flow (TPM).

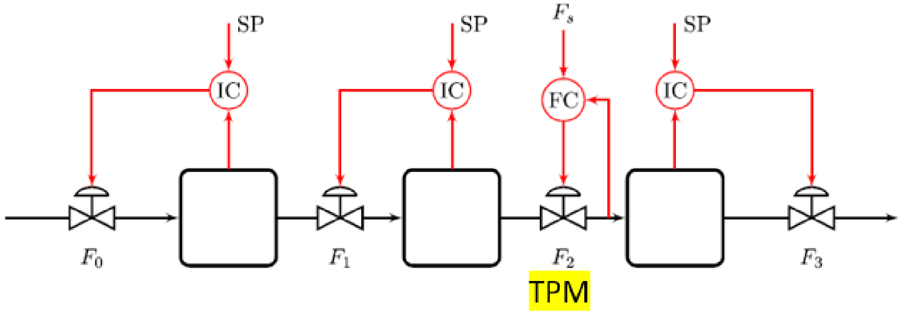


(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)

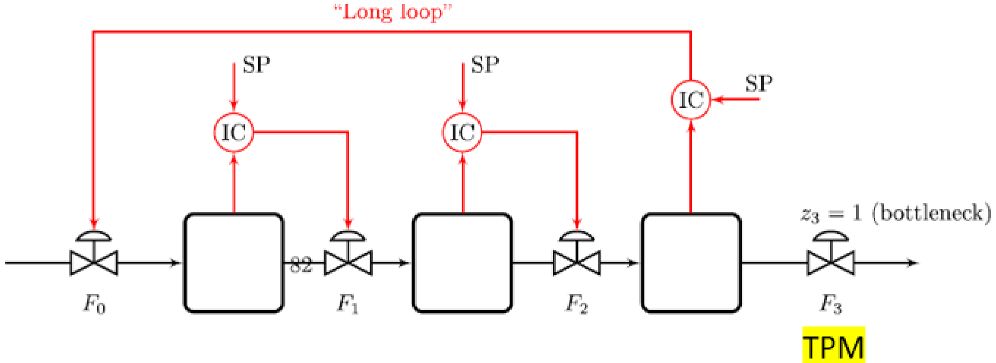


(b) Inventory control in opposite direction of flow (for given product flow, TPM= F_3)

Follows radiation rule



(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)



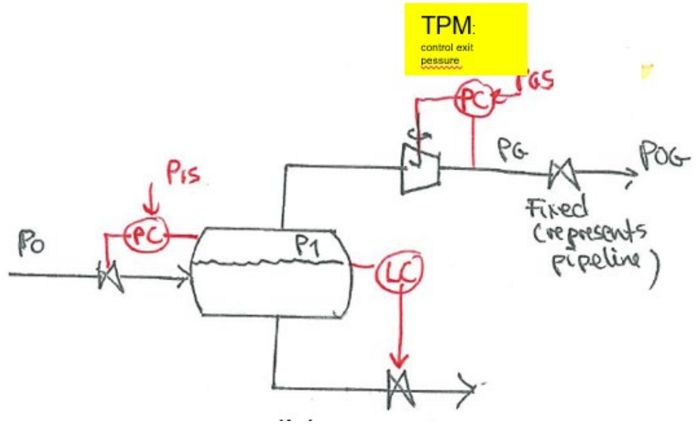
(d) Inventory control with undesired "long loop", not in accordance with the "radiation rule" (for given product flow, TPM= F_3)

Does NOT follow radiation rule

Rules for inventory control

Rules for inventory control

- **Rule 1.** Cannot control (set the flowrate) the same flow twice
- **Rule 2**. Follow the radiation rule whenever possible
- **Rule 3. (**which should never been broken): No inventory loop should cross the location of the TPM
- **Rule 4**. Controlling inlet or outlet pressure indirectly sets the flow (indirectly makes it a TPM)

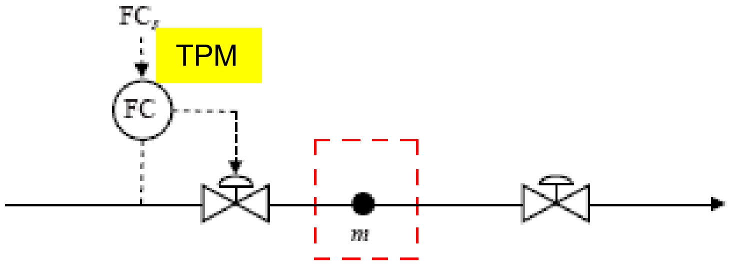


Rule 2. Controlling outlet pressure sets flow

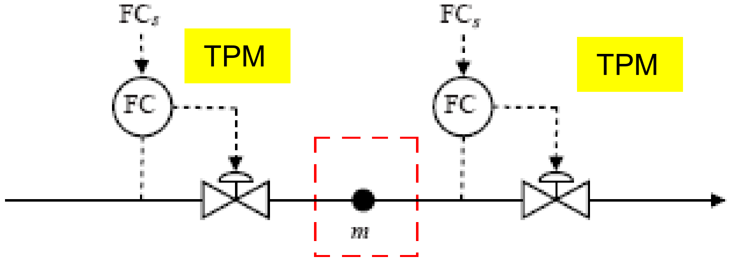
TPM = Variable used for setting the throughput/production rate (for the entire process).

QUIZ. Are these structures workable (consistent)? Yes or No?

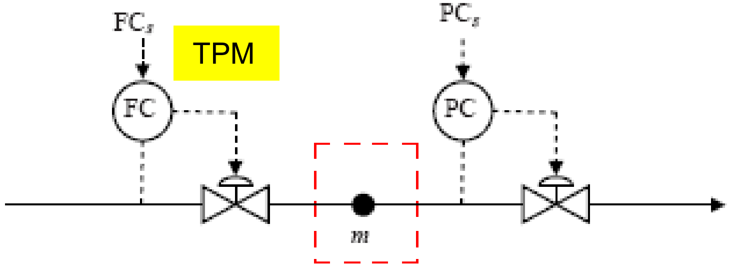
Hint: What happens to the mass holdup inside the red box? Is it self-regulated?



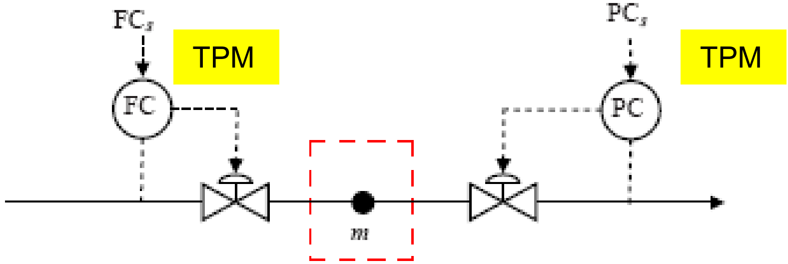
(a) [redacted]



(b) [redacted]

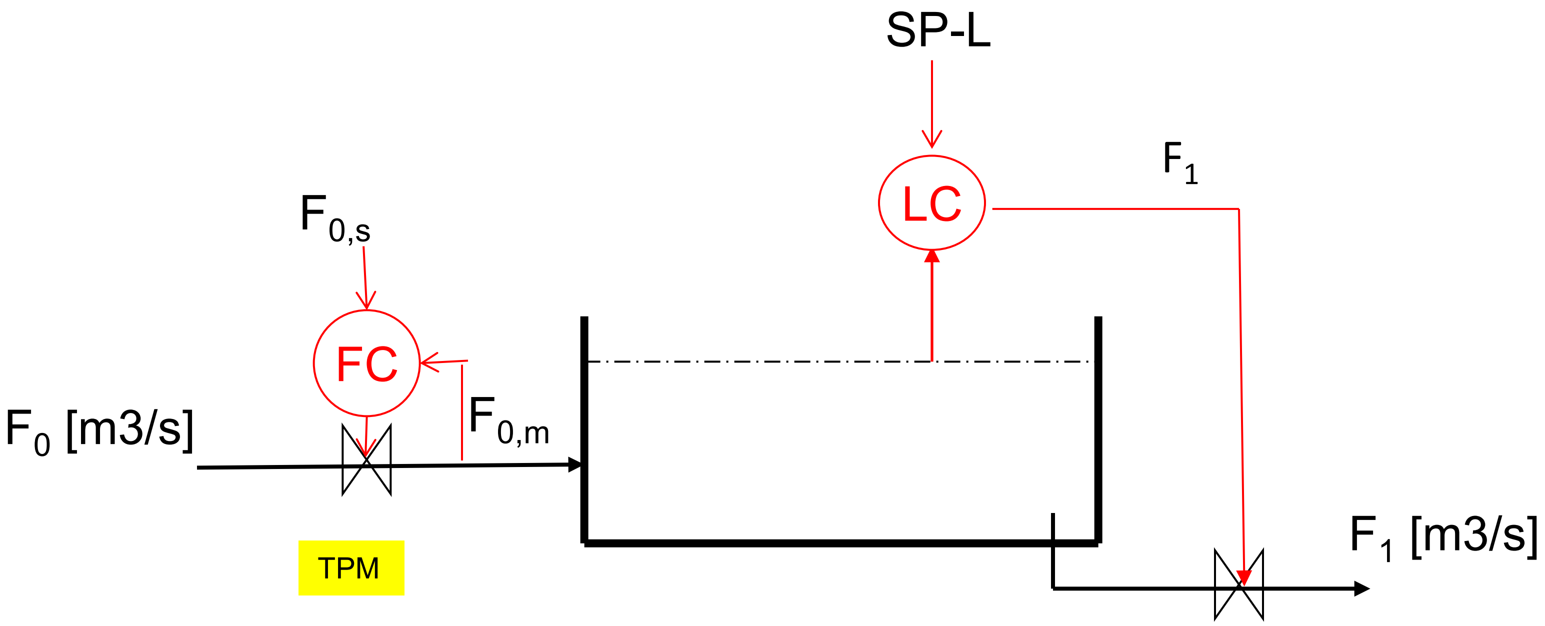


(c) [redacted]



(d) [redacted] ry on inventory m

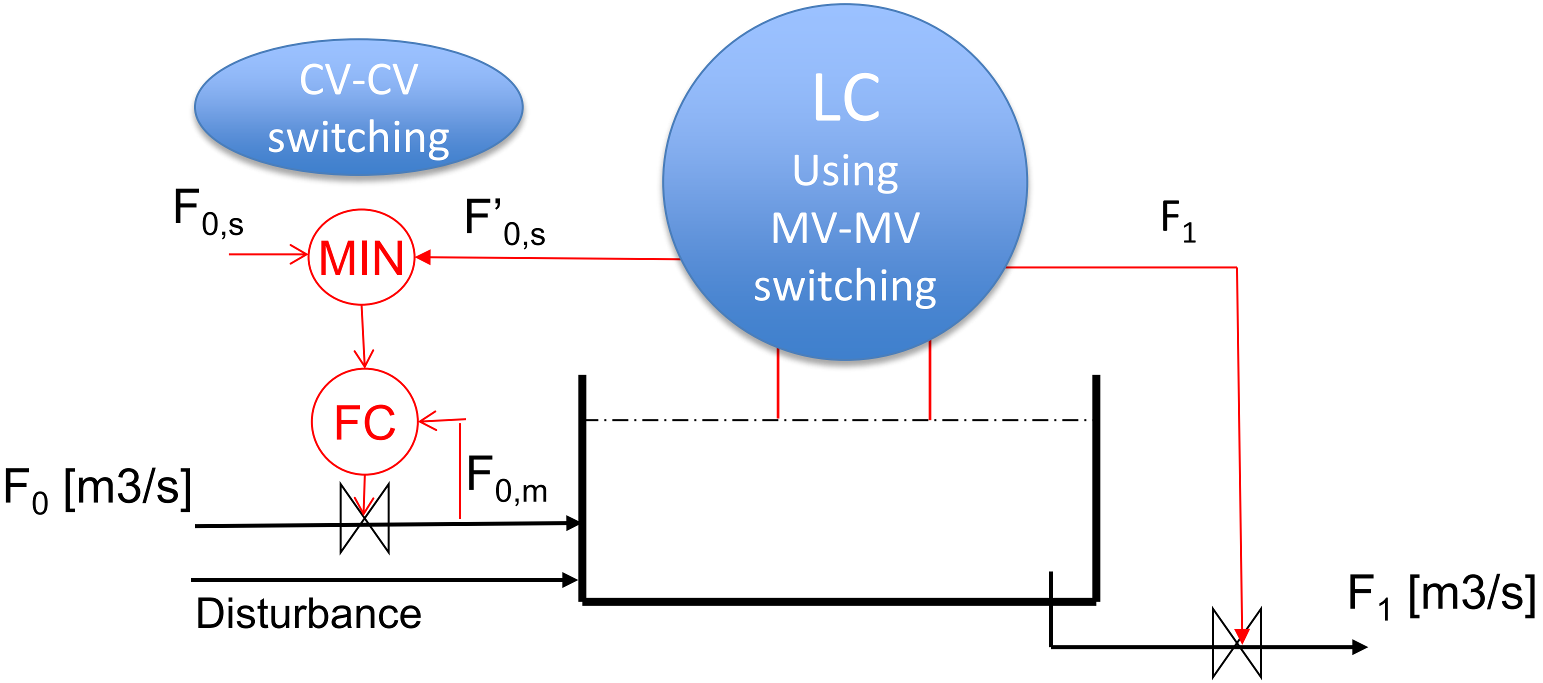
Example: Level control



What should we do if bottleneck at F1 (fully open valve, z1=1)?

Example: Level control. Complex MV-CV switching

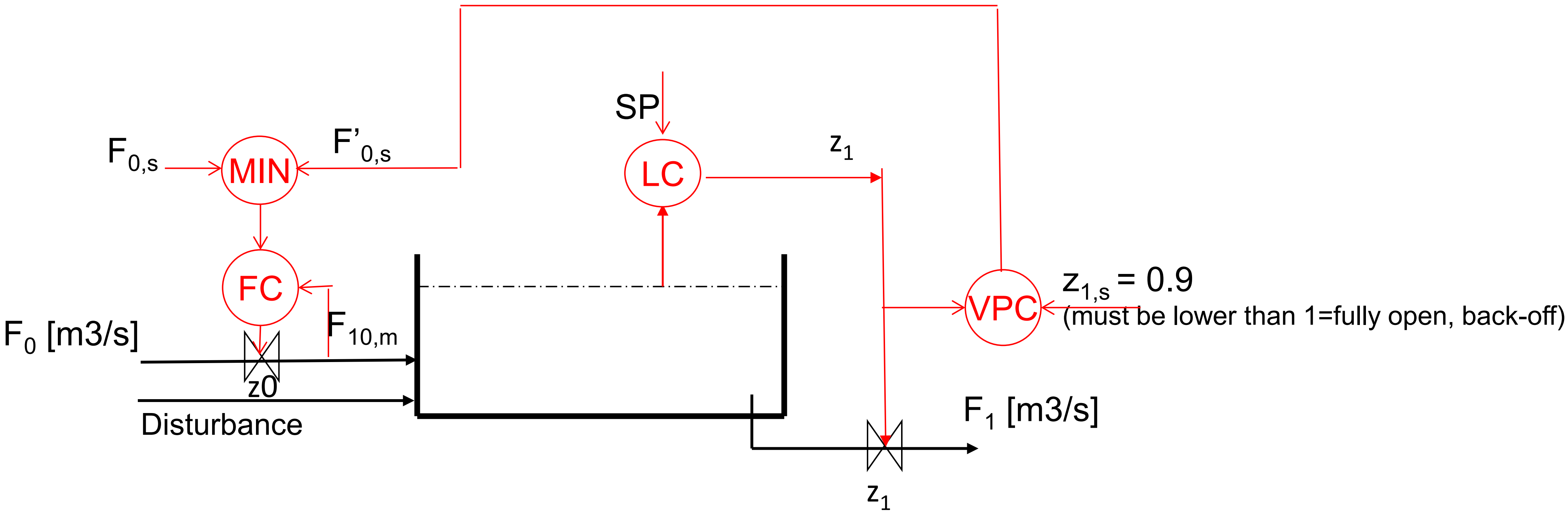
“Bidirectional inventory control” (Shinskey, 1981)



Three alternsatives for MV-MV switching

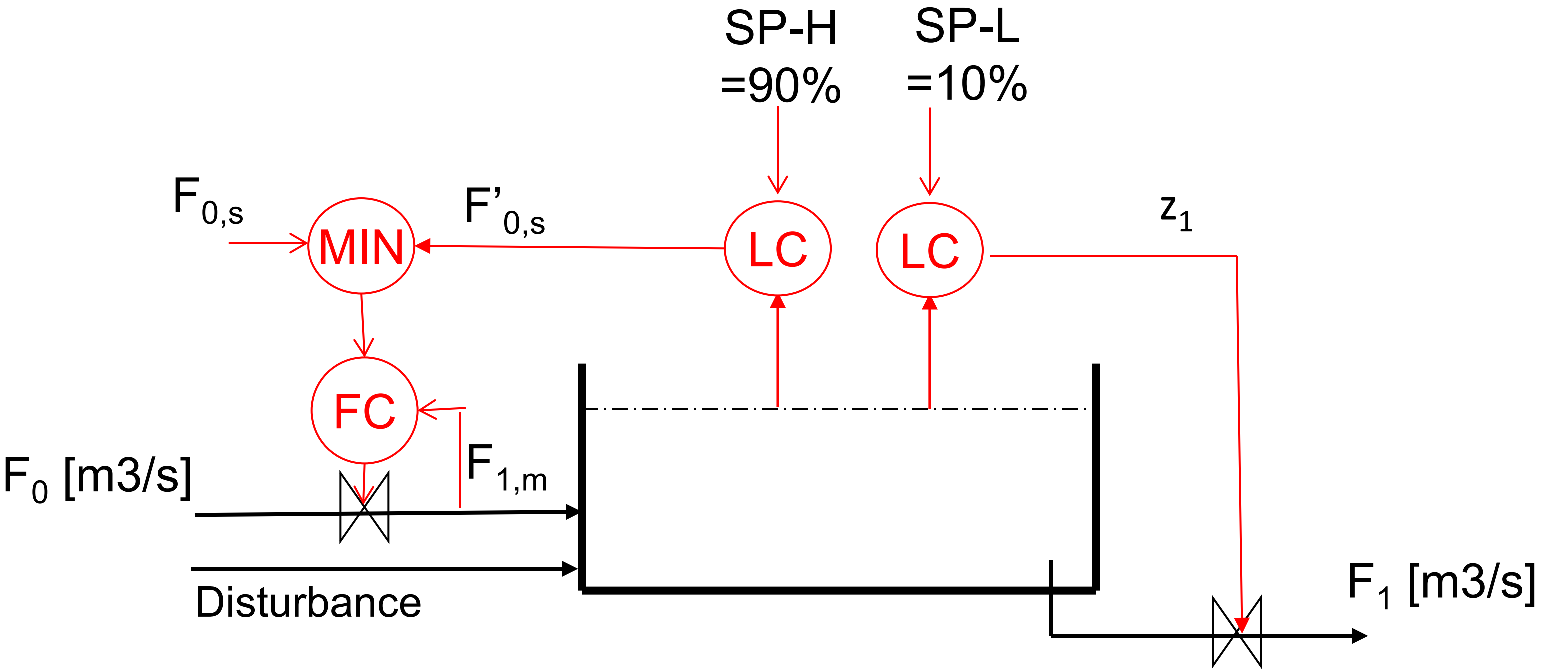
1. SRC (problem since F_{0s} varies)
2. Two controllers
3. VPC (“Long loop” for F1)

Alt. 3 MV-MV switching: VPC



VPC: “reduce inflow (F_0) if outflow valve (z_1) approaches fully open”

Alt. 2 MV-MV switching: Two controllers (recommended)

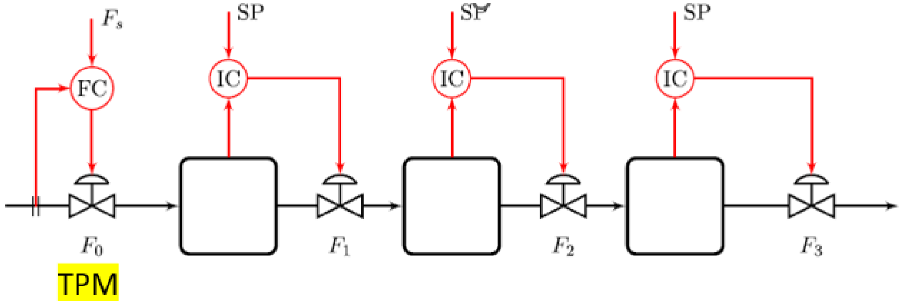


SP-L = low level setpoint
SP-H = high level setpoint

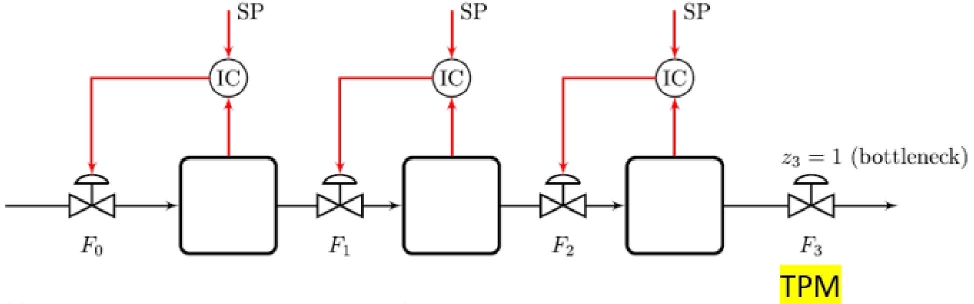
Extra benefit: Use of two setpoints is good for using buffer dynamically!!

Inventory control for units in series

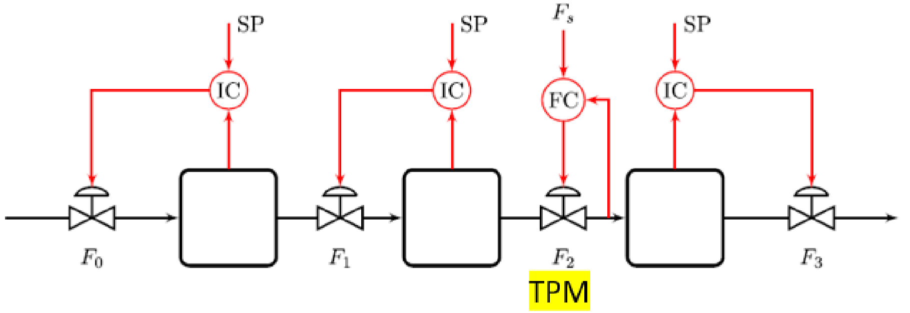
Radiating rule:
Inventory control should be "radiating" around a given flow (TPM).



(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)



(b) Inventory control in opposite direction of flow (for given product flow, TPM= F_3)



(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)

Follows radiation rule

Need to reconfigure inventory loops if TPM moves

Generalization of bidirectional inventory control

Reconfigures TPM automatically with optimal buffer management!!

Maximize throughput:

$F_s = \infty$

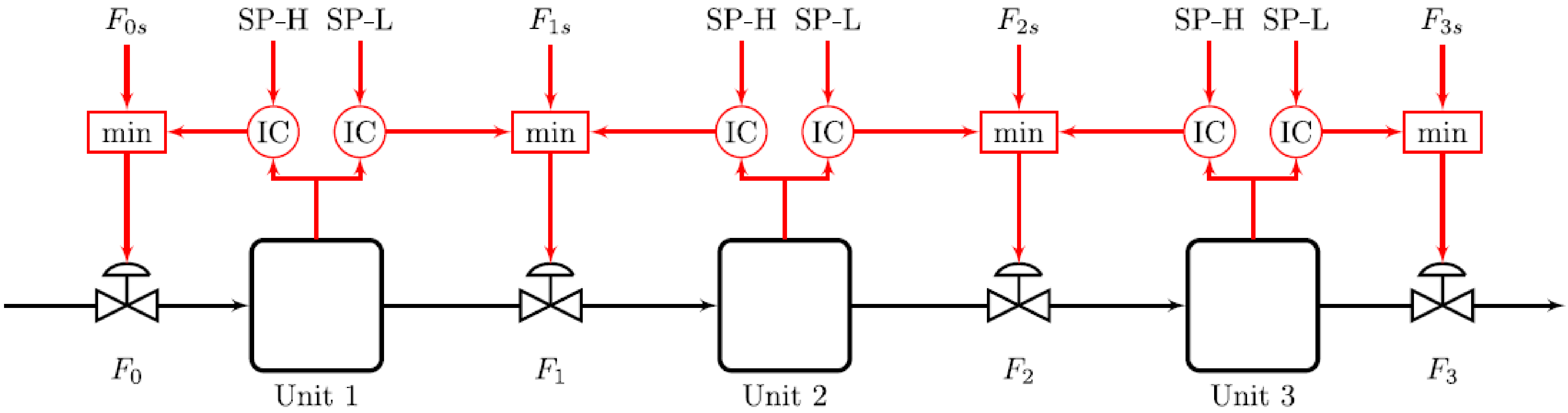


Fig. 38. Bidirectional inventory control scheme for automatic reconfiguration of loops (in accordance with the radiation rule) and maximizing throughput. Shinskey (1981) Zotica et al. (2022).

SP-H and SP-L are high and low inventory setpoints, with typical values 90% and 10%.

Strictly speaking, with setpoints on (maximum) flows ($F_{i,s}$), the four valves should have slave flow controllers (not shown). However, one may instead have setpoints on valve positions (replace $F_{i,s}$ by $z_{i,s}$), and then flow controllers are not needed.

F.G. Shinskey, «Controlling multivariable processes», ISA, 1981, Ch.3

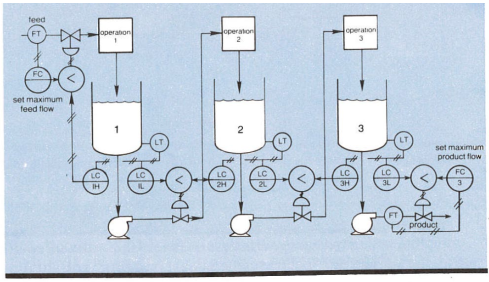
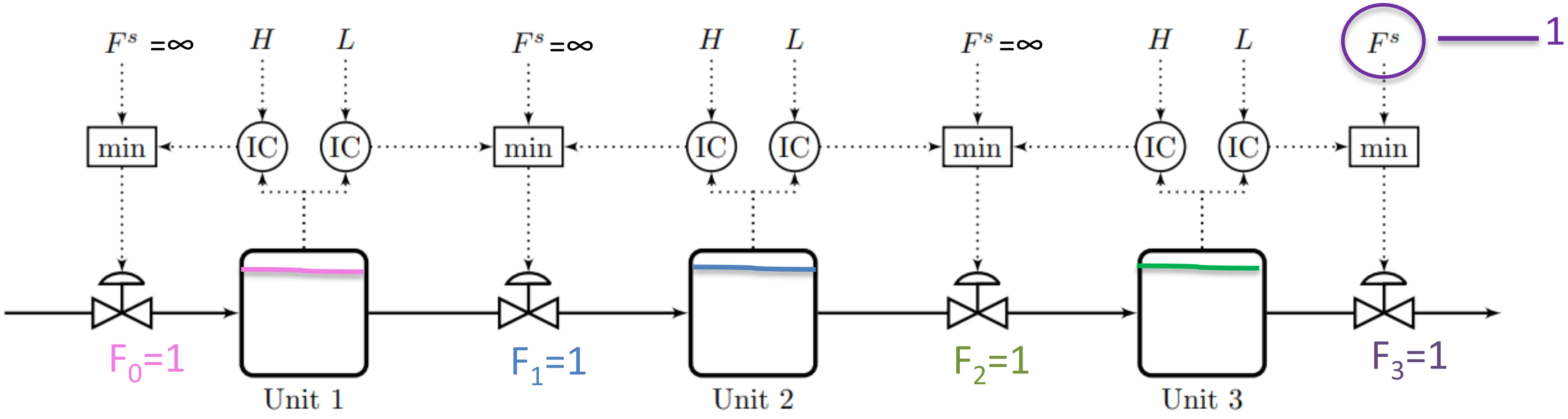
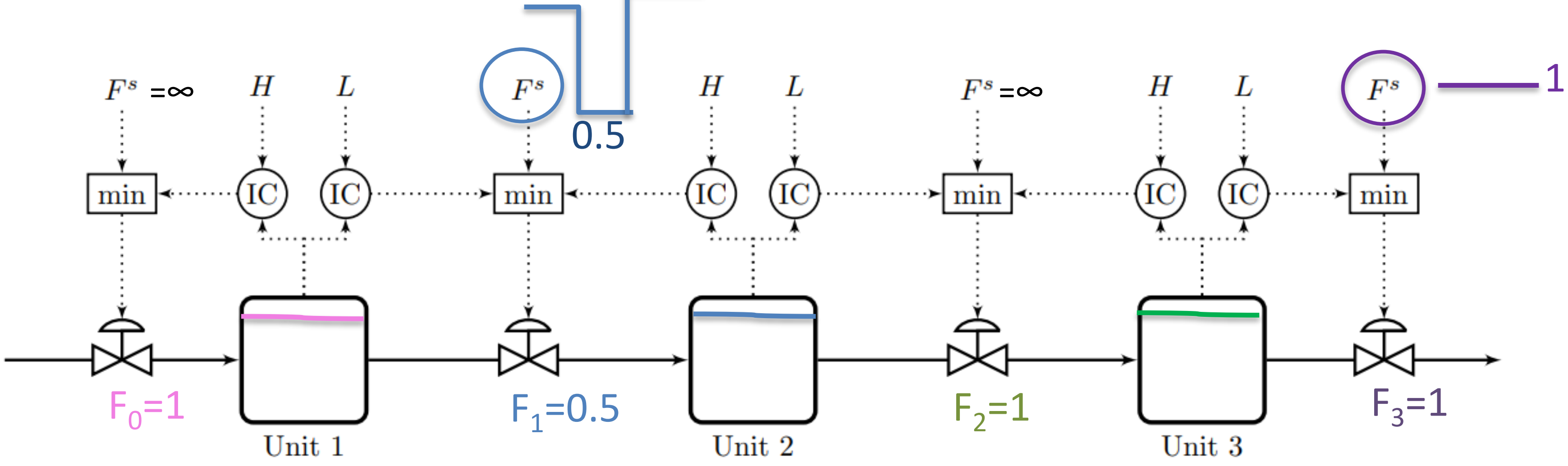


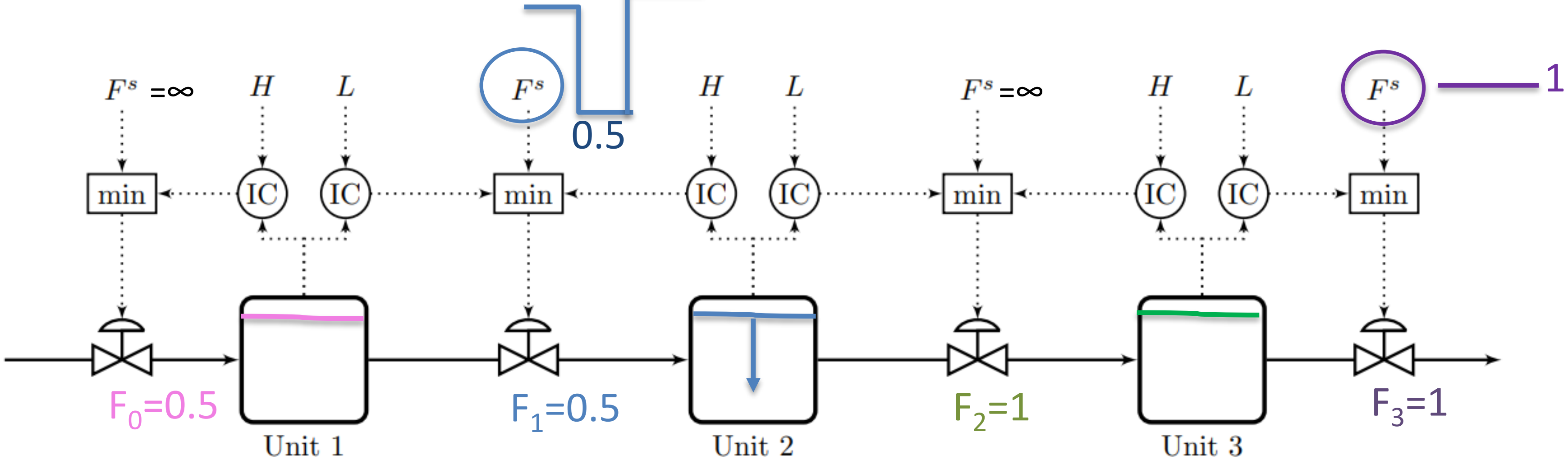
Fig. 3-7. Production rate can be set at either end of the process or constrained at any intermediate point without loss of inventory control.

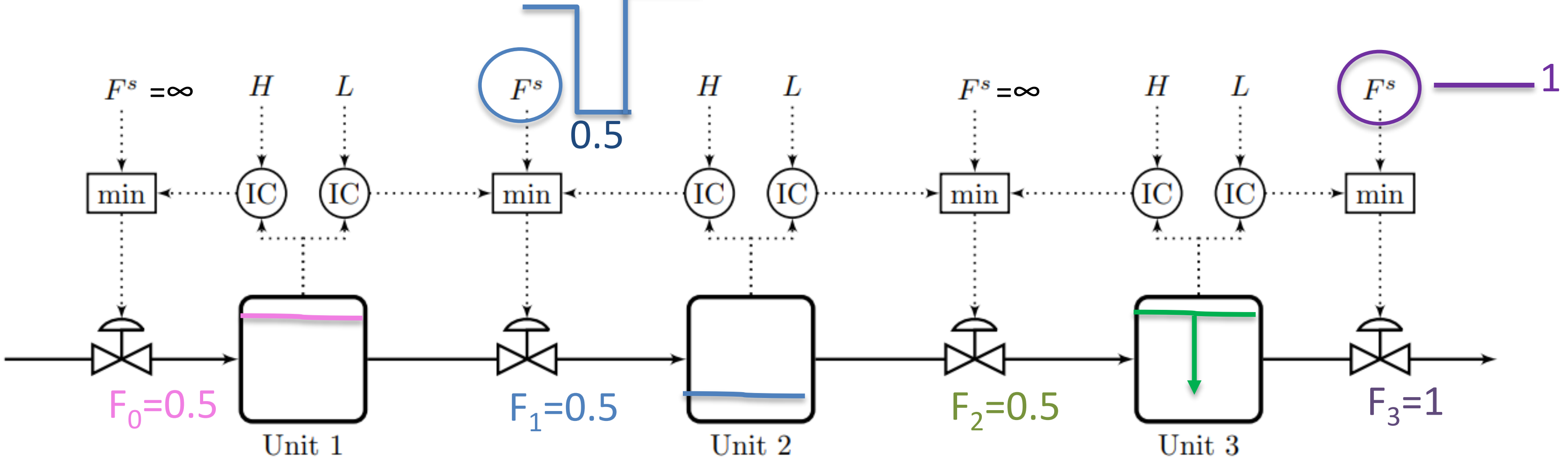
Cristina Zotica, Krister Forsman, Sigurd Skogestad ,»Bidirectional inventory control with optimal use of intermediate storage», Computers and chemical engineering, 2022

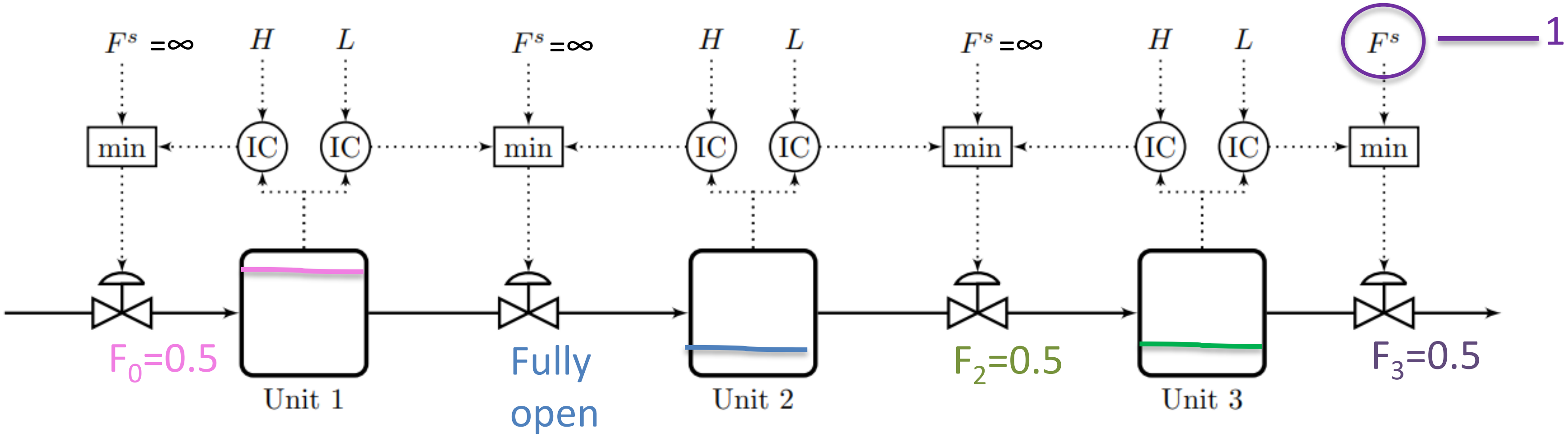

F^s =∞
H
L
F^s =∞
H
L
F^s =∞
H
L
F^s
1
min
IC
IC
min
IC
IC
min
IC
IC
min
F_0=1
Unit 1
F_1=1
Unit 2
F_2=1
Unit 3
F_3=1



F^s =∞
H
L
F^s
0.5
H
L
F^s =∞
H
L
F^s
1
min
IC
IC
min
IC
IC
min
IC
IC
min
F_0=1
F_1=0.5
F_2=1
F_3=1
Unit 1
Unit 2
Unit 3


F^s =∞
H
L
F^s
0.5
H
L
F^s =∞
H
L
F^s
1
min
IC
IC
min
IC
IC
min
IC
IC
min
F_0=0.5
F_1=0.5
F_2=1
F_3=1
Unit 1
Unit 2
Unit 3


F^s =∞
H
L
F^s
0.5
H
L
F^s =∞
H
L
F^s
1
min
IC
IC
min
IC
IC
min
IC
IC
min
F_0=0.5
Unit 1
F_1=0.5
Unit 2
F_2=0.5
Unit 3
F_3=1



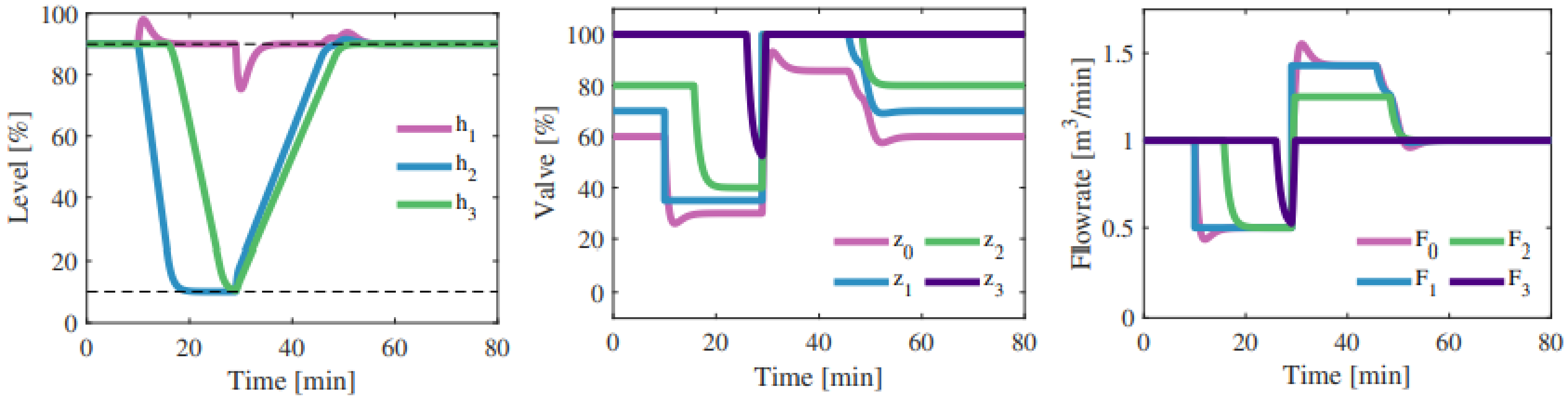
1

F_0=0.5

Fully open

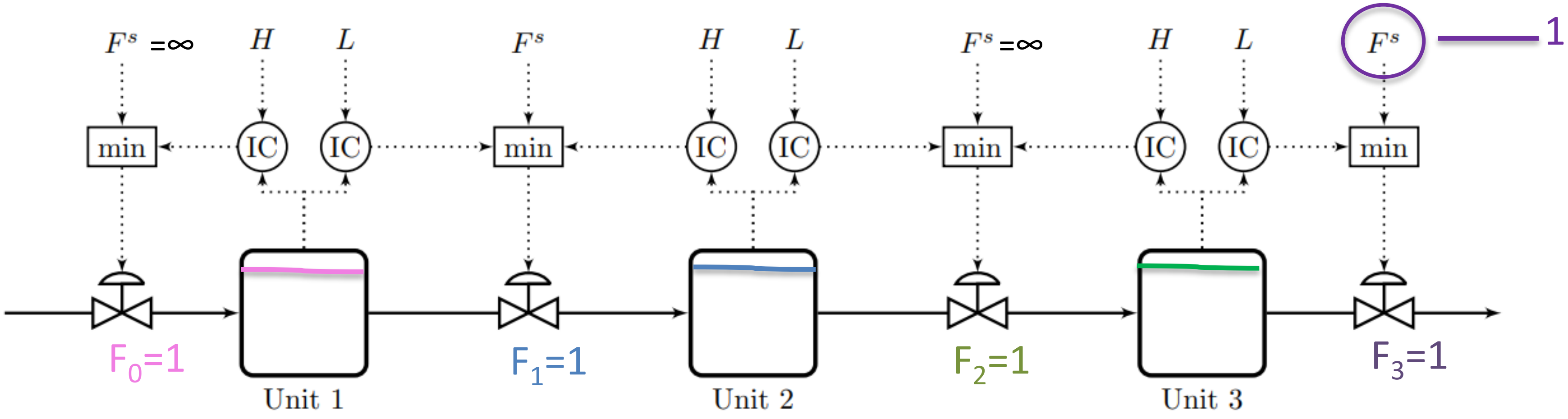
F_2=0.5

F_3=0.5



(a) Levels (b) Valve positions (c) Flows

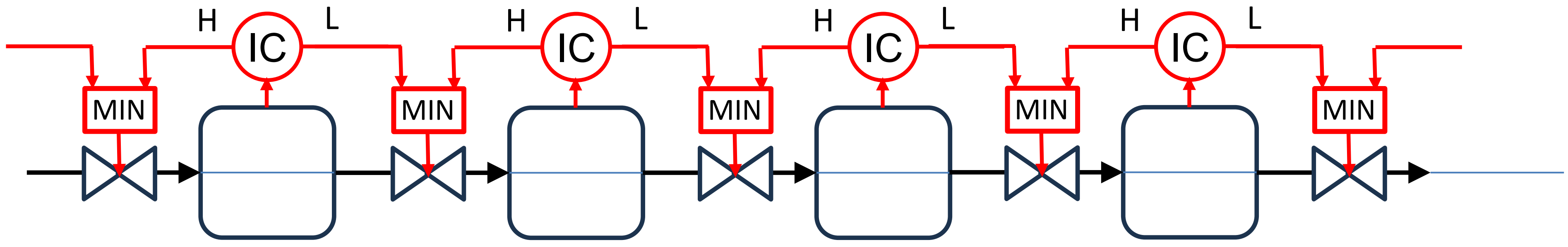
Figure 12: Simulation of a 19 min temporary bottleneck in flow F_1 for the control structures in Fig. 3d with the TPM downstream of the bottleneck.

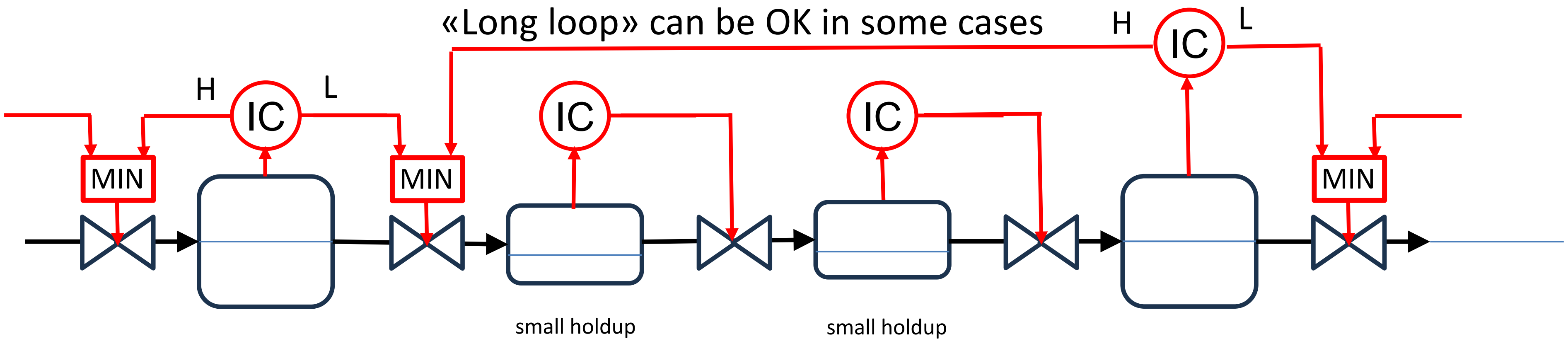


Challenge: Can MPC be made to do his? Optimally reconfigure loops and find optimal buffer?

- Yes, possible with standard setpoint-based MPC if we use
    - Trick: All flow setpoints = infinity (unachievable setpoint)
- What about Economic MPC? Cannot do it easily; may try scenario-MPC

Don’t need bidirectional control on all units



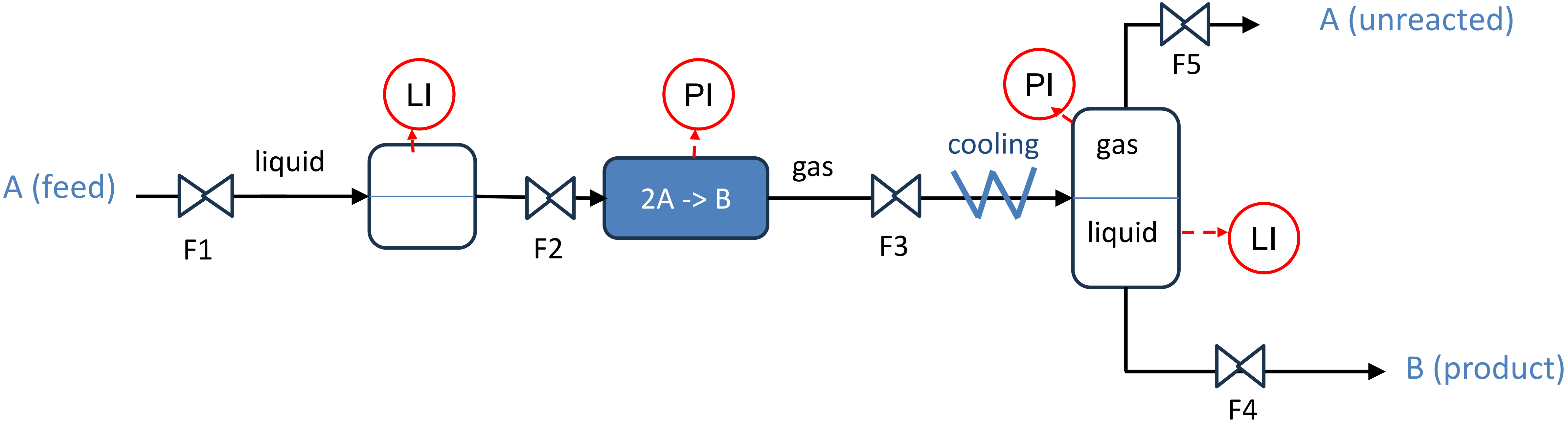

«Long loop» can be OK in some cases

Important insight

- Many problems: Optimal steady-state solution always at constraints
- In this case optimization layer may not be needed
  - if we can identify the active constraints and control them using selectors

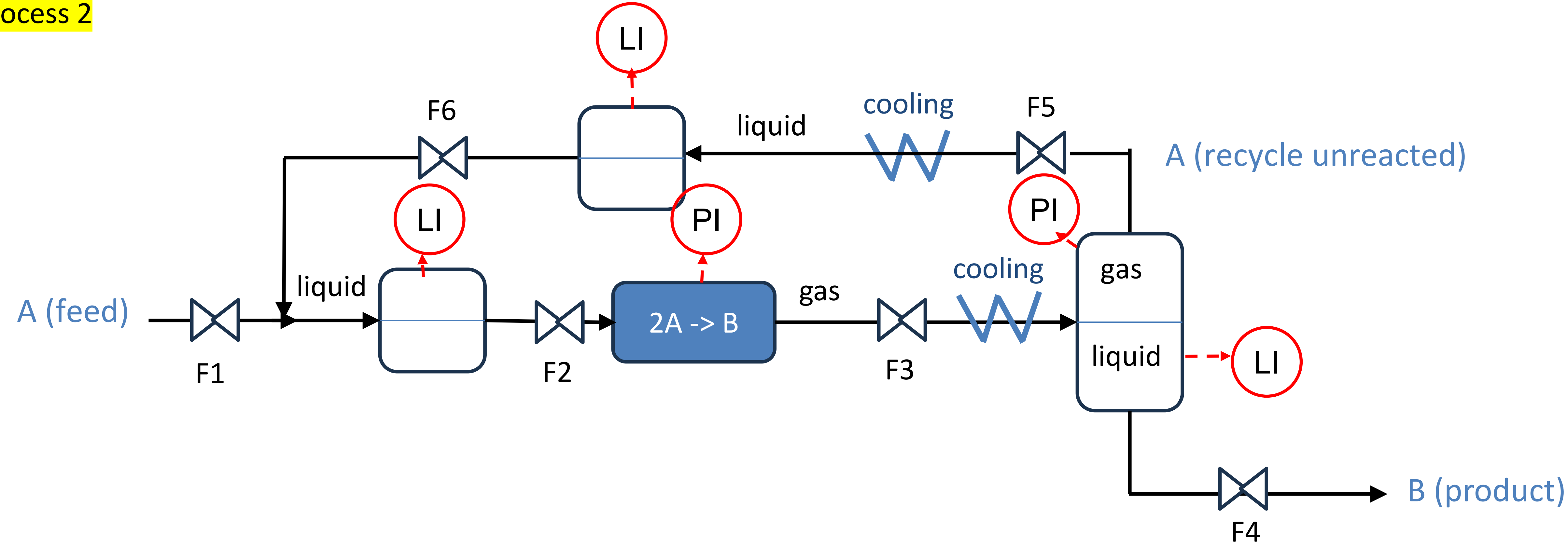
Control of chemical processes with recycle

Process 1



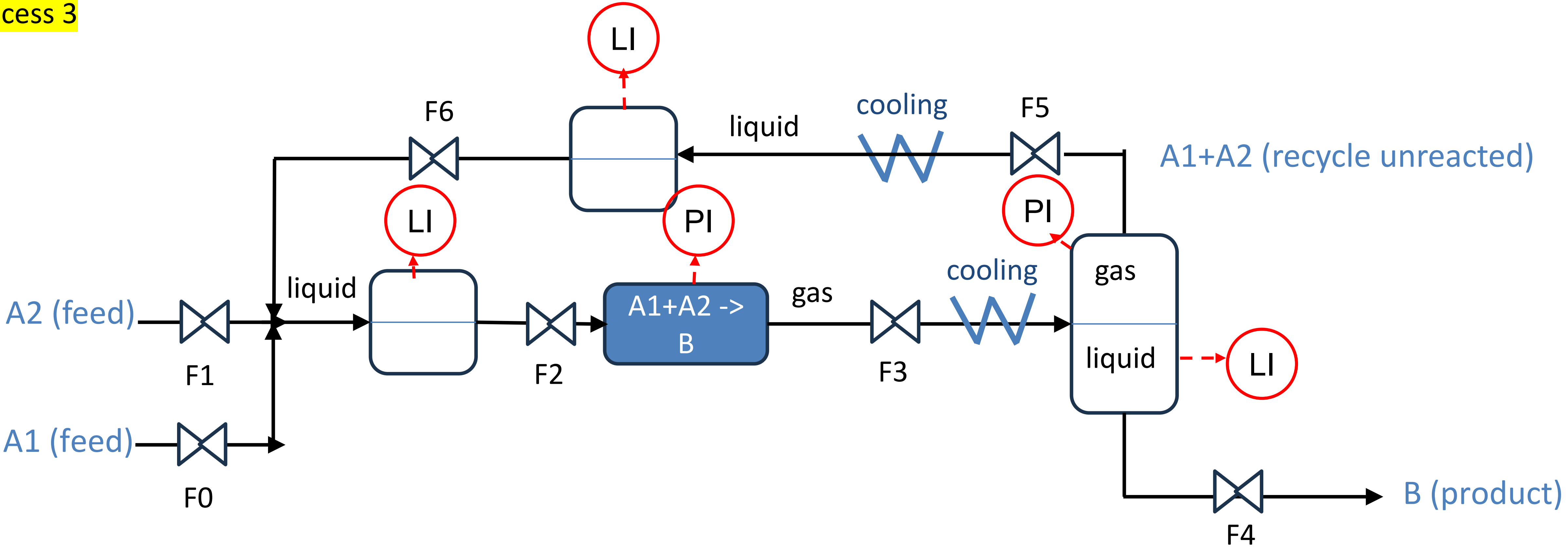
Exothermic reaction

Process 2

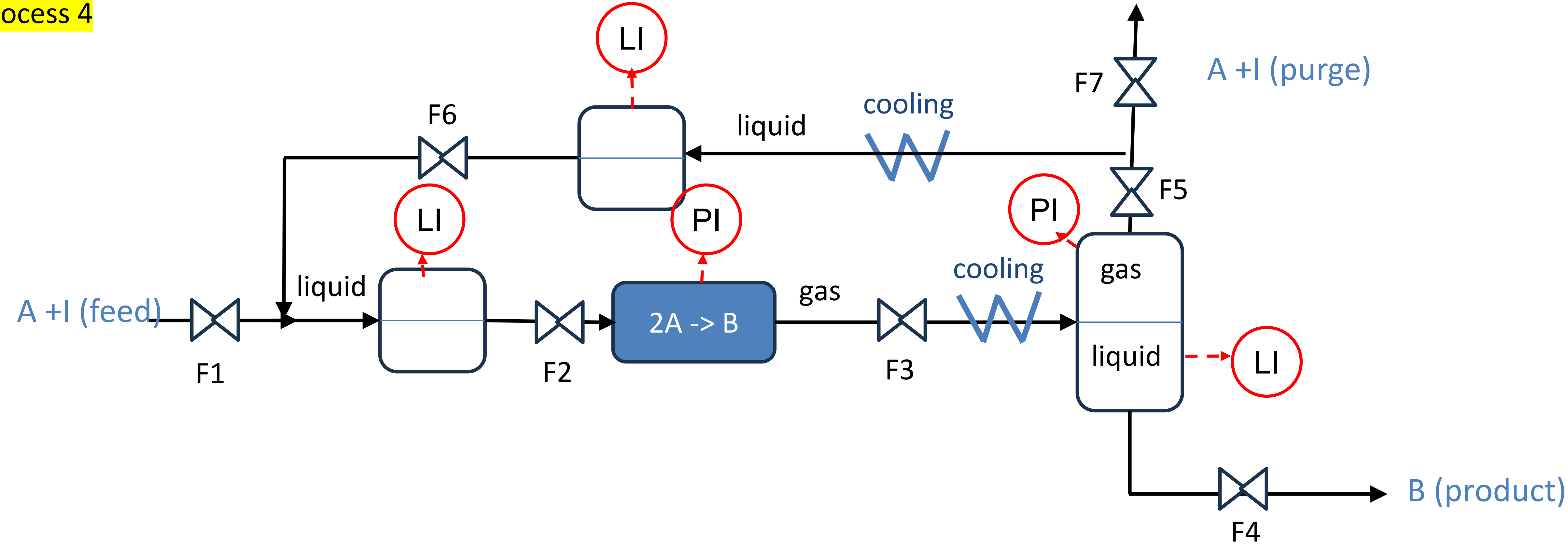


Comment: Valve F5 may not be necessary. Could use valve on cooling instead

Process 3

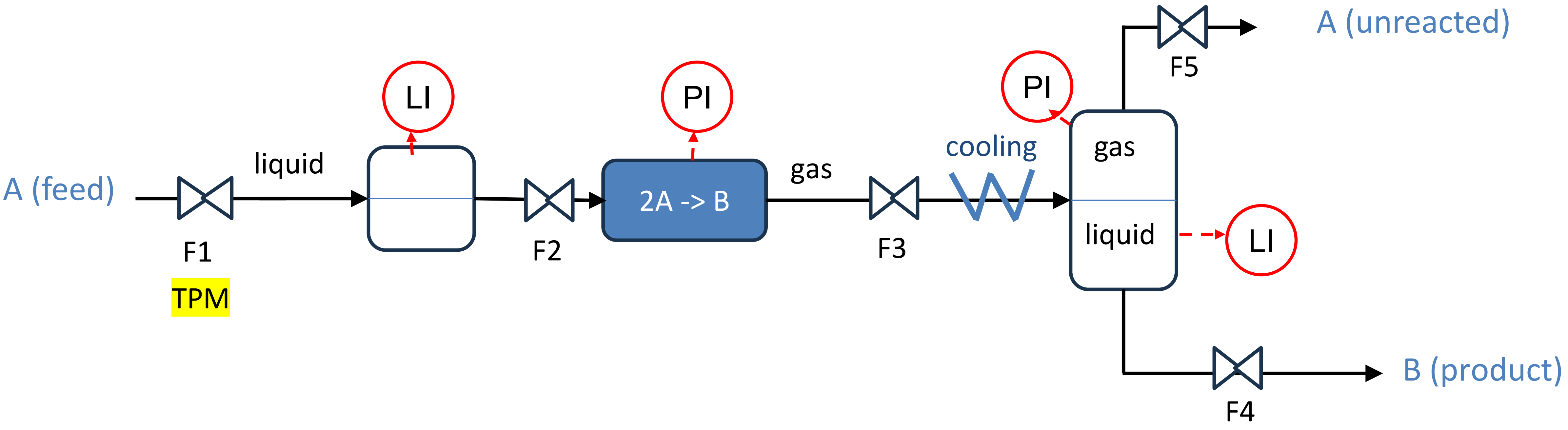

LI
F6
liquid
cooling
F5
A1+A2 (recycle unreacted)
LI
PI
PI
liquid
cooling
gas
gas
A2 (feed)
A1+A2 -> B
F1
F2
F3
liquid
LI
A1 (feed)
F0
B (product)
F4

Process 4


LI
F7
A +I (purge)
cooling
F6
liquid
F5
PI
LI
PI
liquid
gas
cooling
gas
A +I (feed)
2A -> B
liquid
LI
F1
F2
F3
F4
B (product)

Process 1

Control

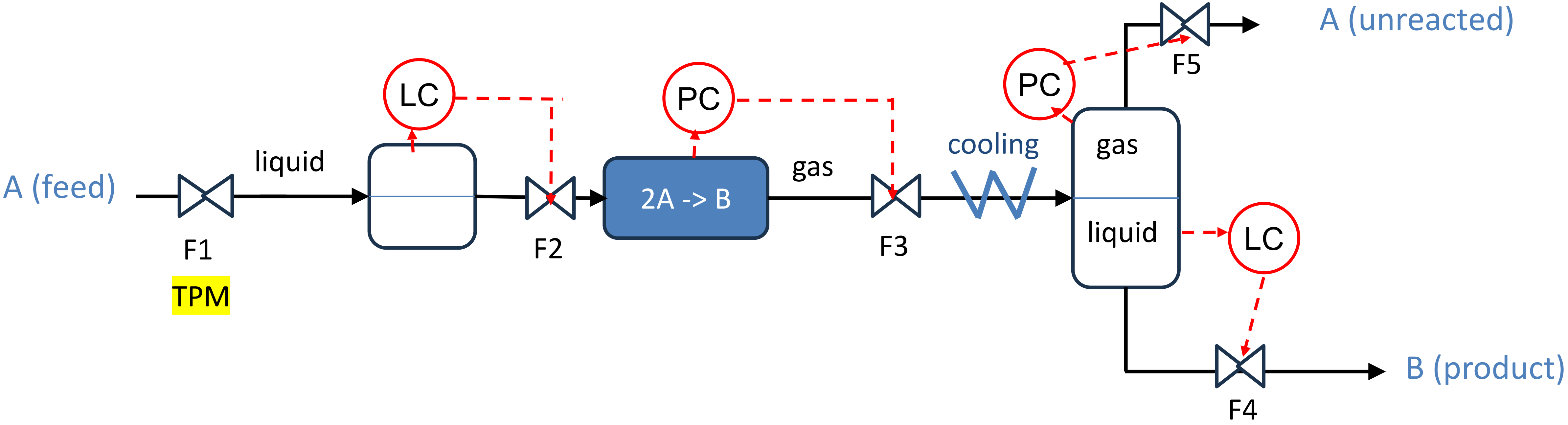


TPM

Exothermic reaction

Process 1

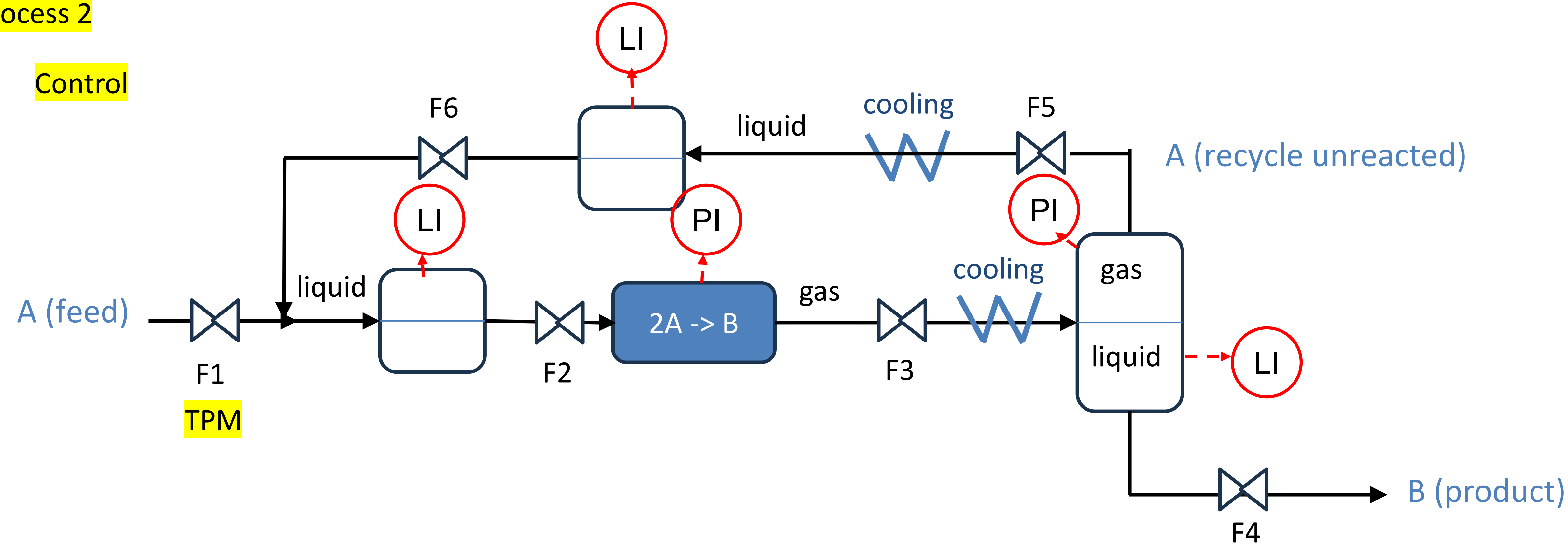
Control



Exothermic reaction

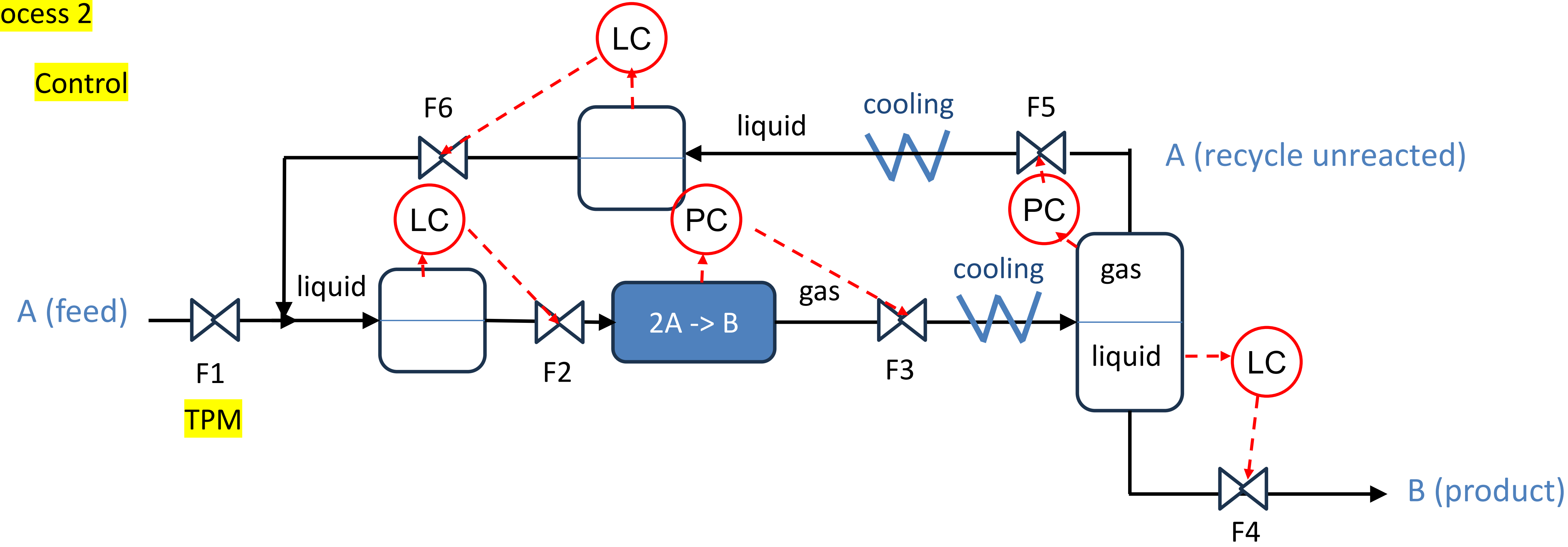
Process 2

Control



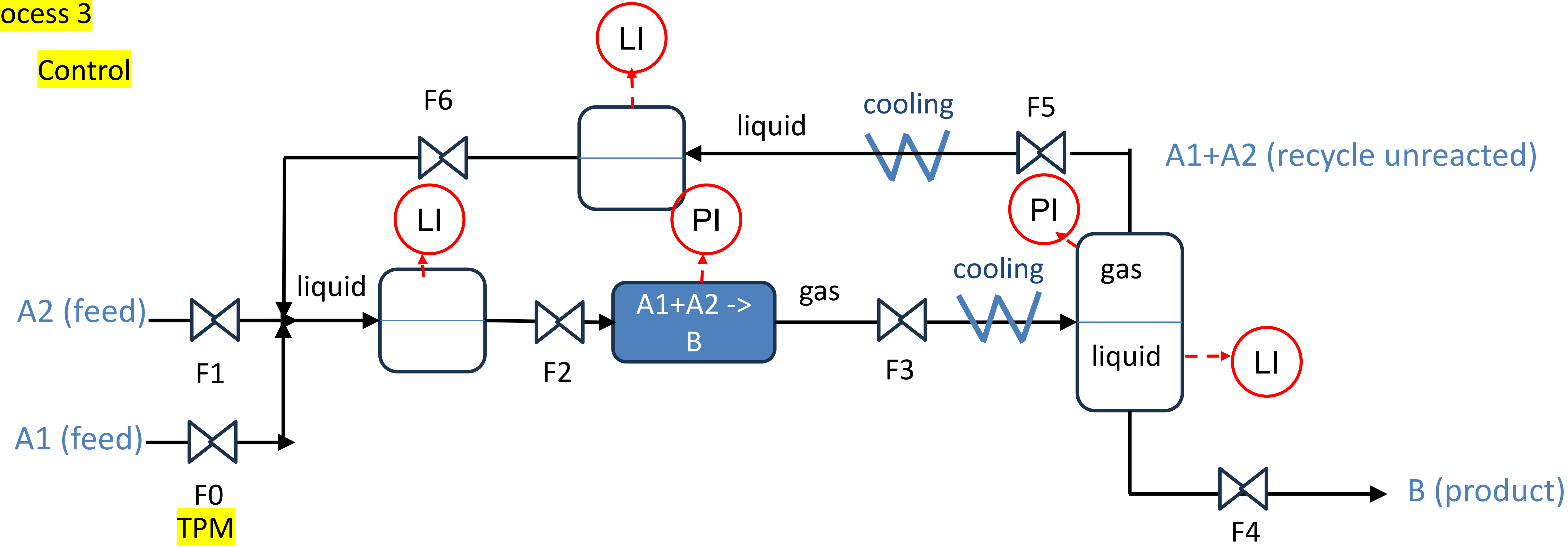
Process 2

Control



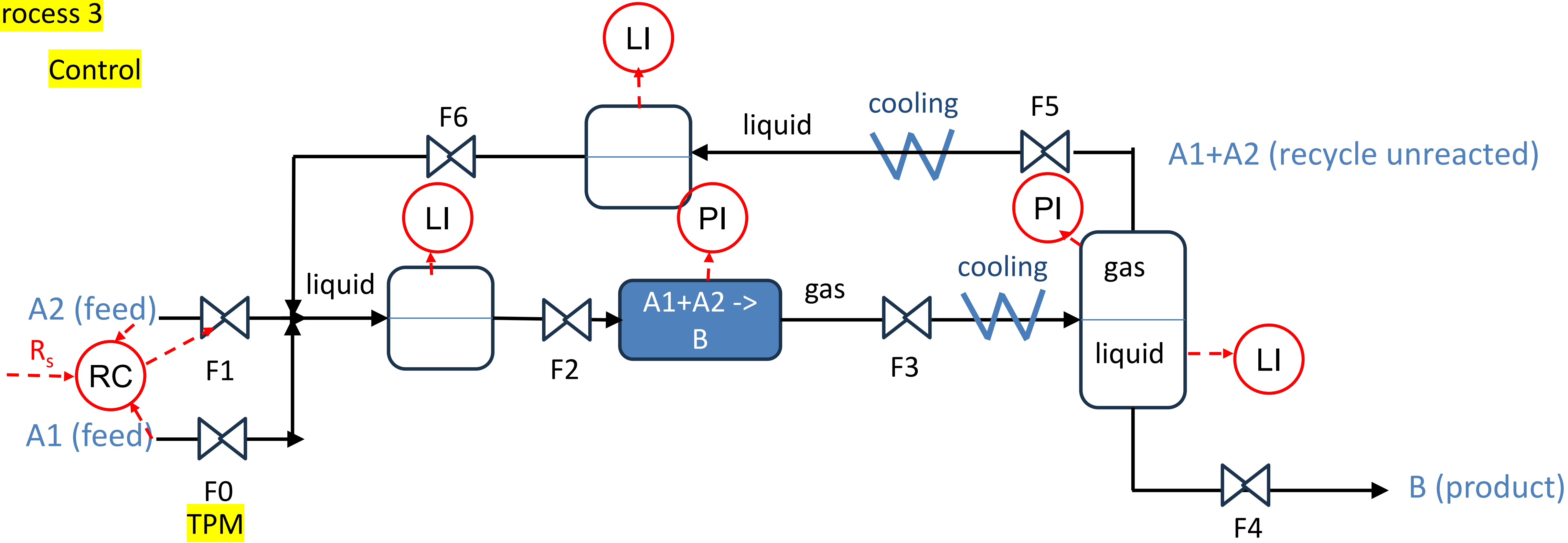
Process 3

Control



Process 3

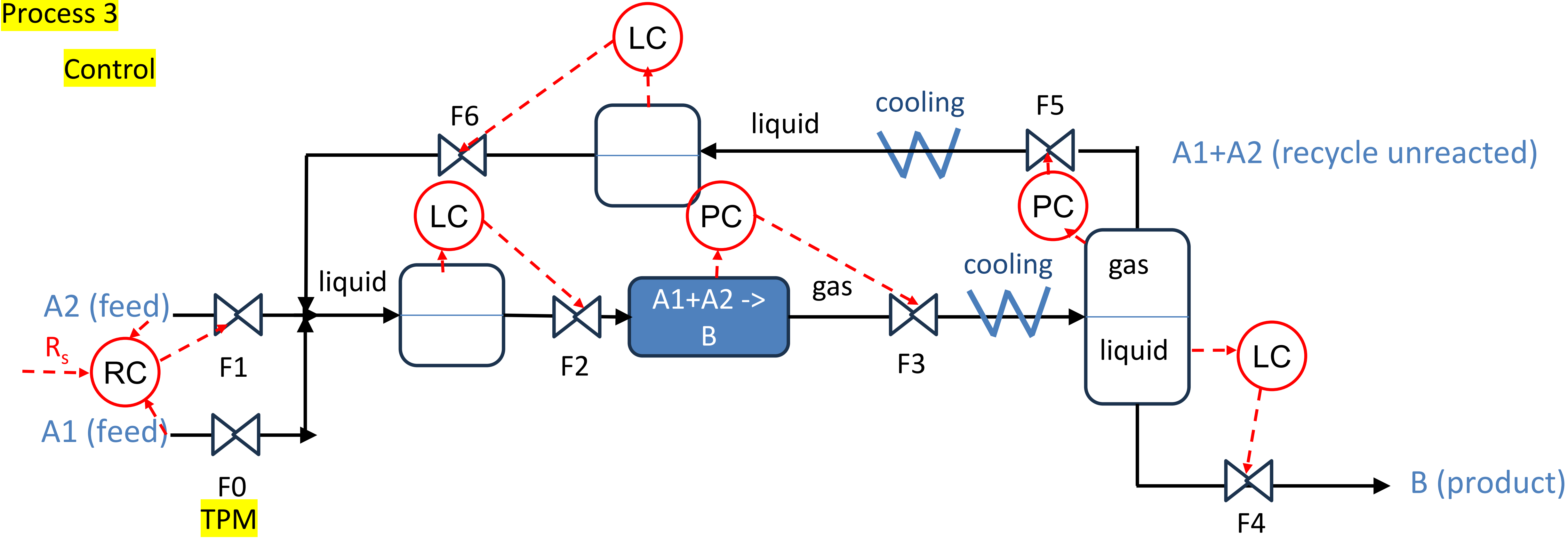
Control



The ratio control can be done in different ways.
It requires two flow measurements (F0, F1)
One of the flows is the TPM

Process 3

Control

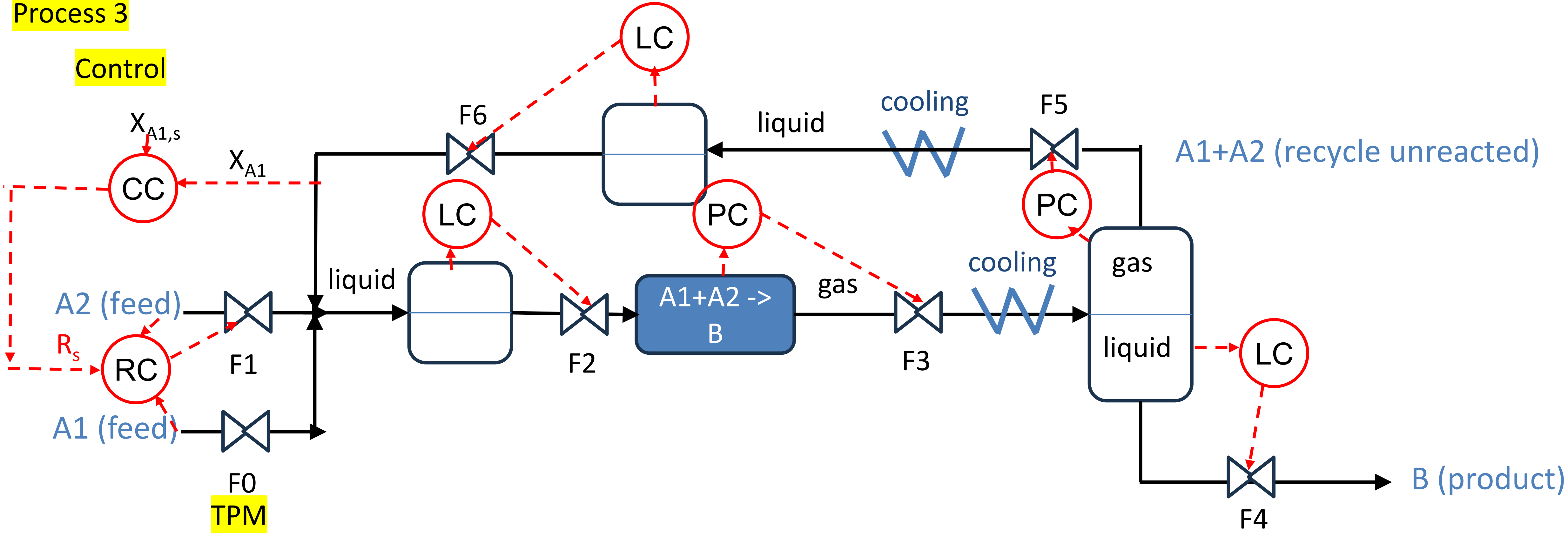


Will this work?

No, it’s not possible to feed exactly the same amount of A1 and A2 without feedback correction

Process 3

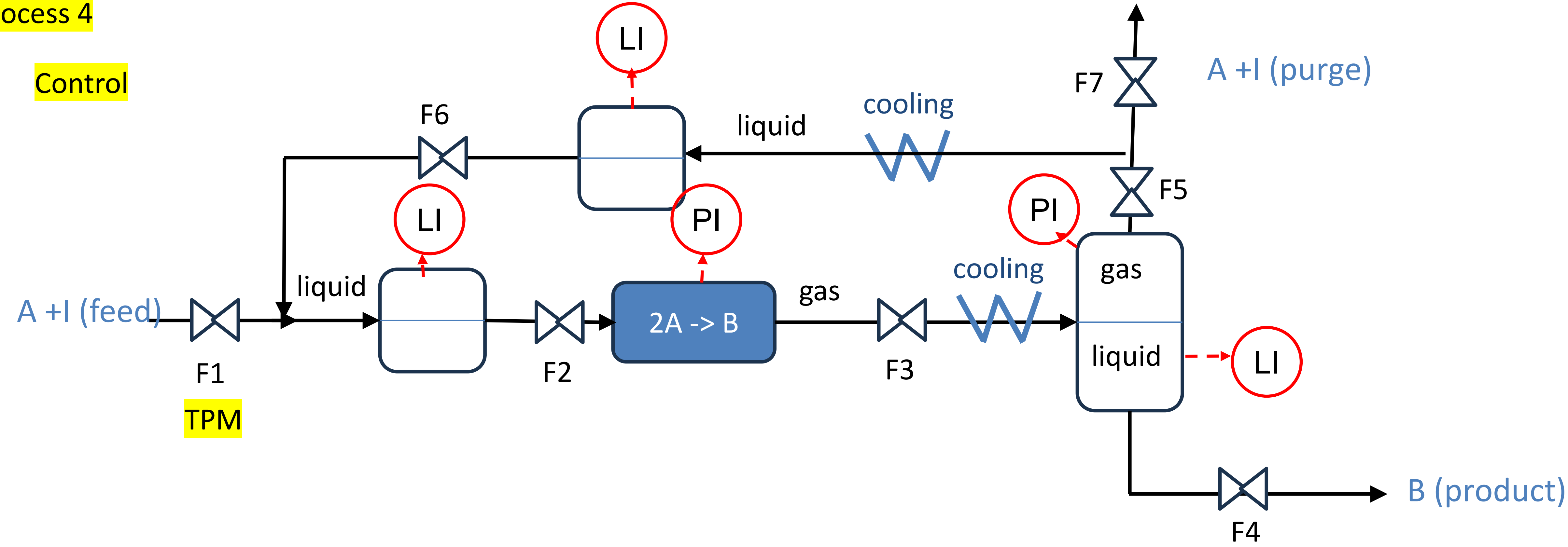
Control



With composition control of A1 (or A2).
This works!

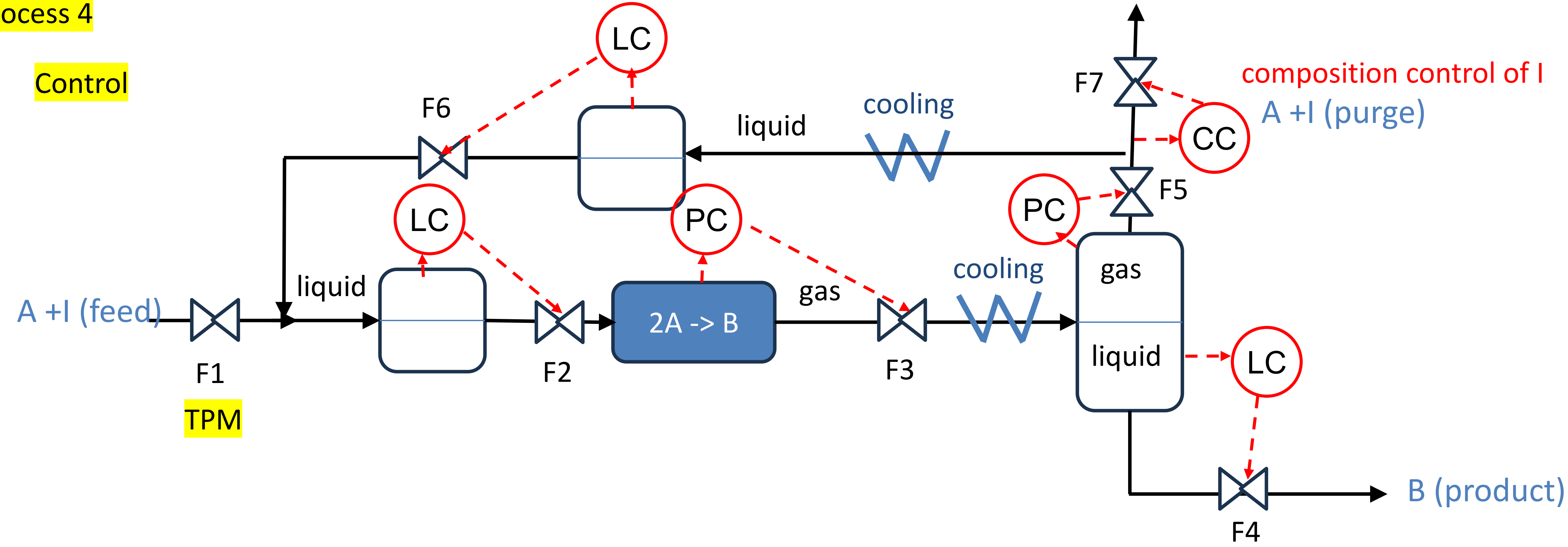
Process 4

Control



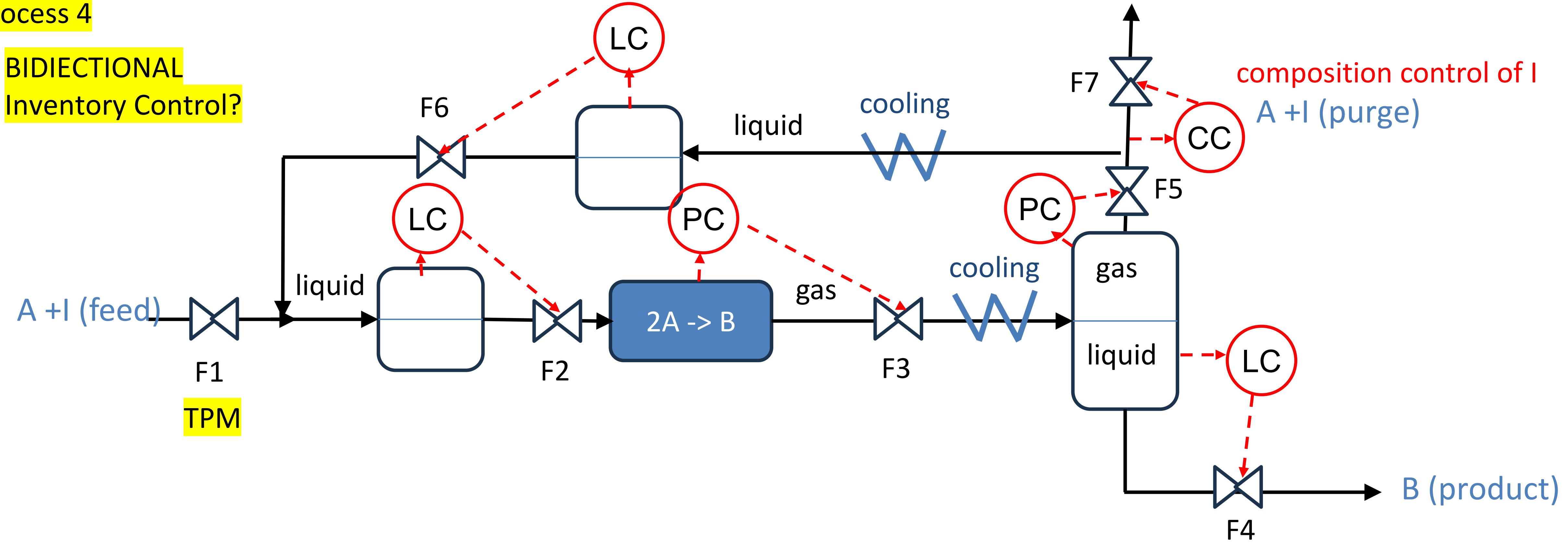
Process 4

Control



Process 4

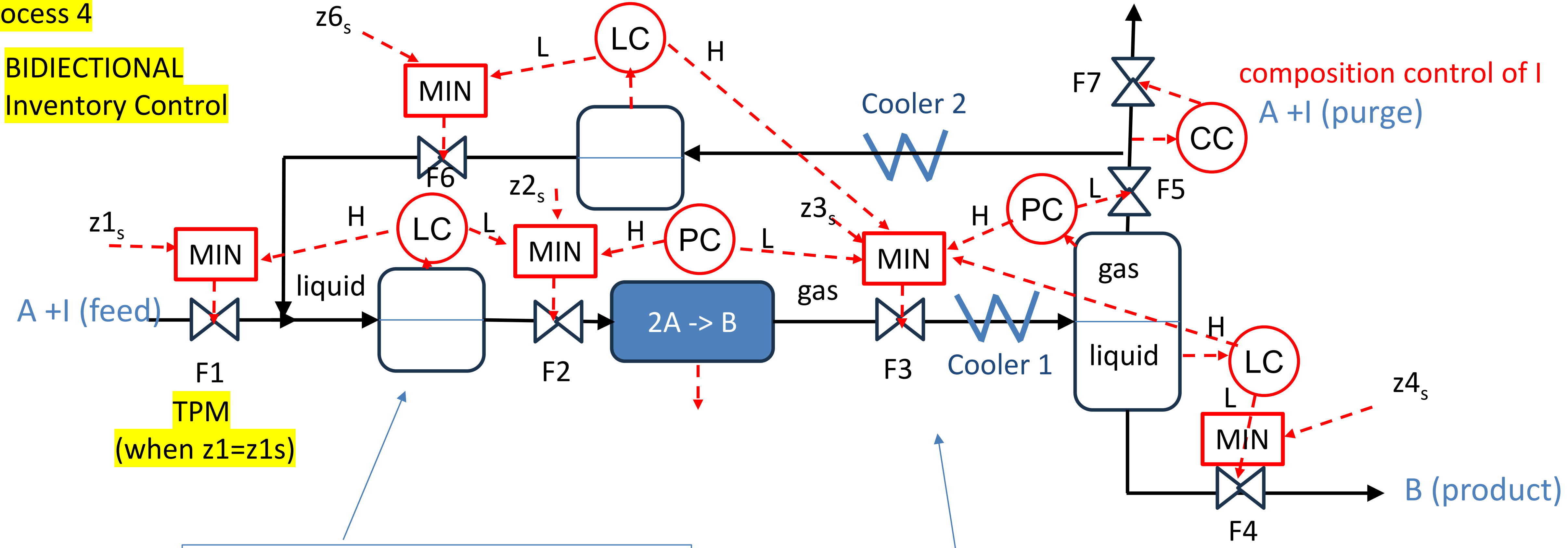
BIDIECTIONAL
Inventory Control?



TPM

Process 4

BIDIECTIONAL
Inventory Control



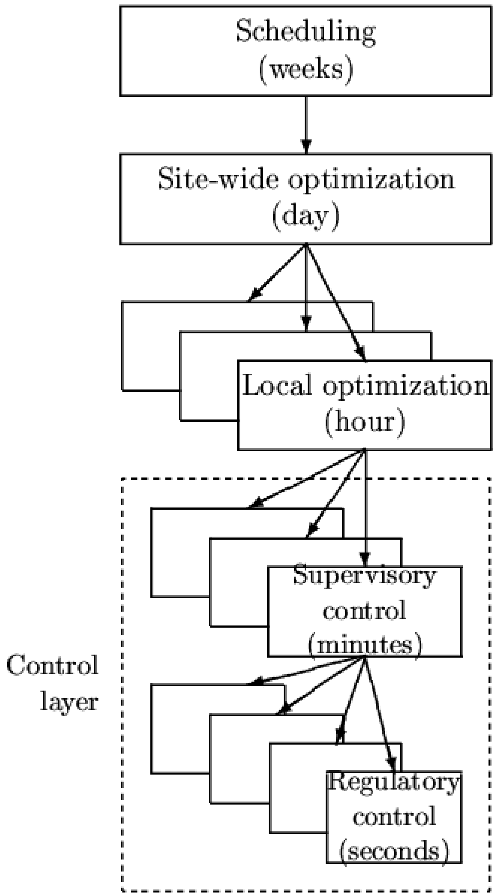
This LC is two controllers which both control level.

- The one with outflow F2 as the MV has a Low level setpoint
- The one with inflow F1 as the MV has a High level setpoint

3 H-setpoints go to this MIN-selector:
Reduce F3 if

1. too high pressure (cooler 2 max),
2. too much gas (high level) (F6 limiting)
3. too much liquid (high level) (F4 limiting)

Optimal operation and control objectives: What should we control?



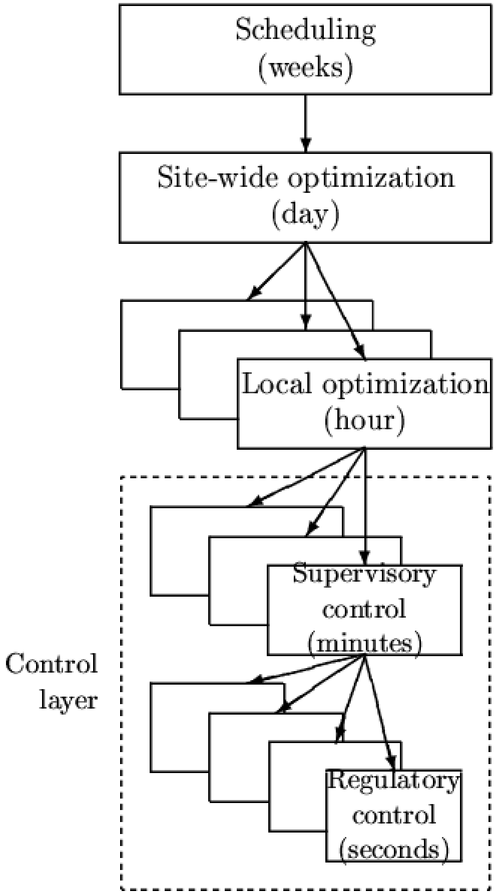
CV$_1$ (economics)

CV$_2$ (stabilization)

Skogestad procedure for control structure design:

I. Top Down (analysis)
  - Step S1: Define operational objective (cost) and constraints
  - Step S2: Identify degrees of freedom and optimize operation for disturbances
  - Step S3: Implementation of optimal operation
    - What to control? (CV1) (self-optimizing control)
  - Step S4: Where set the production rate (TPM)? (Inventory control)

II. Bottom Up (design)
  - Step S5: Regulatory control: What more to control (CV2)?
  - Step S6: Supervisory control
  - Step S7: Real-time optimization



CV_1 (economics)

CV_2 (stabilization

TPM = Throughput manipulator

Step S1. Define optimal operation (economics)

- What are the ultimate goals of the operation?
- Typical cost function*:

J = cost feed + cost energy – value products

*No need to include fixed costs (capital costs, operators, maintainance) at ”our” time scale (hours)

Note: J=-P where P= Operational profit

Example Step 1: distillation column

- Distillation at steady state with given p and F: N=2 DOFs, e.g. L and V (u)
- **Cost to be minimized (economics)**

$$J = -P \text{ where } P = p_D D \underbrace{+\, p_B B}_{\text{value products}} - \underset{\text{cost feed}}{p_F F} - \overset{\text{cost energy (heating + cooling)}}{p_V V}$$

- **Constraints**

  Purity D: For example, $x_{D,\text{ impurity}} \leq \max$

  Purity B: For example, $x_{B,\text{ impurity}} \leq \max$

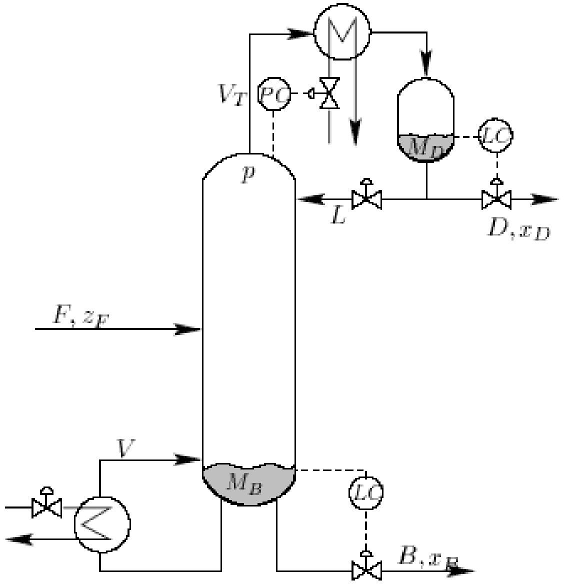
  Flow constraints: $\min \leq D, B, L \text{ etc.} \leq \max$

  Column capacity (flooding): $V \leq V_{max}$, etc.

  Pressure: 1) p given (d) 2) p free (u): $p_{min} \leq p \leq p_{max}$

  Feed: 1) F given (d) 2) F free (u): $F \leq F_{max}$

- Optimal operation: Minimize J with respect to steady-state DOFs (u)

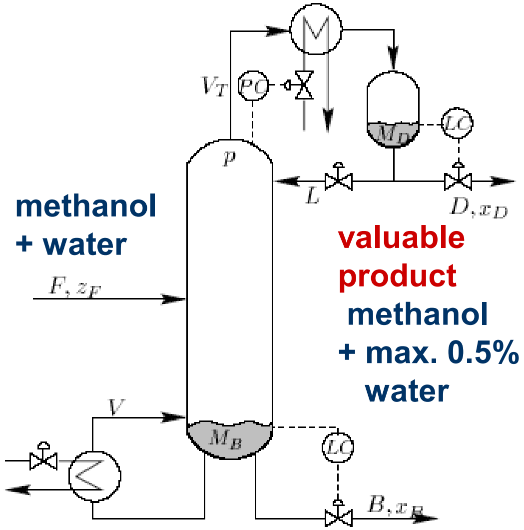


Steps S2/S3. Distillation: expected active constraints

- **Both products (D, B) generally have purity specs**
- **Valuable product: Purity spec. always active**
  - Reason: Amount of valuable product (D or B) should always be maximized
    - Avoid product “give-away” (“Sell water as methanol”)
    - Also saves energy

Control implications:

1. ALWAYS Control valuable product at spec. (active constraint)
2. May overpurify (not control) cheap product
   - And then maybe V=Vmax is active constraint to get max. overpurification



methanol + water

valuable product
methanol + max. 0.5% water

cheap product (byproduct)
water + max. 2% methanol

Example bidirectional inventory control

AIChE

Economic Plantwide Control of the Ethyl Benzene Process

Rahul Jagtap, Ashok S Pathak, and Nitin Kaistha
Dept. of Chemical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016,Uttar Pradesh, India

DOI 10.1002/aic.13964

A1: Benzene
A2: Ethylene
B: Ethylbenzene (product)
C: Diethylbenzene (undersired)

A1+A2 →B
B + A2 → C
C + A1 → 2B

LS = MIN

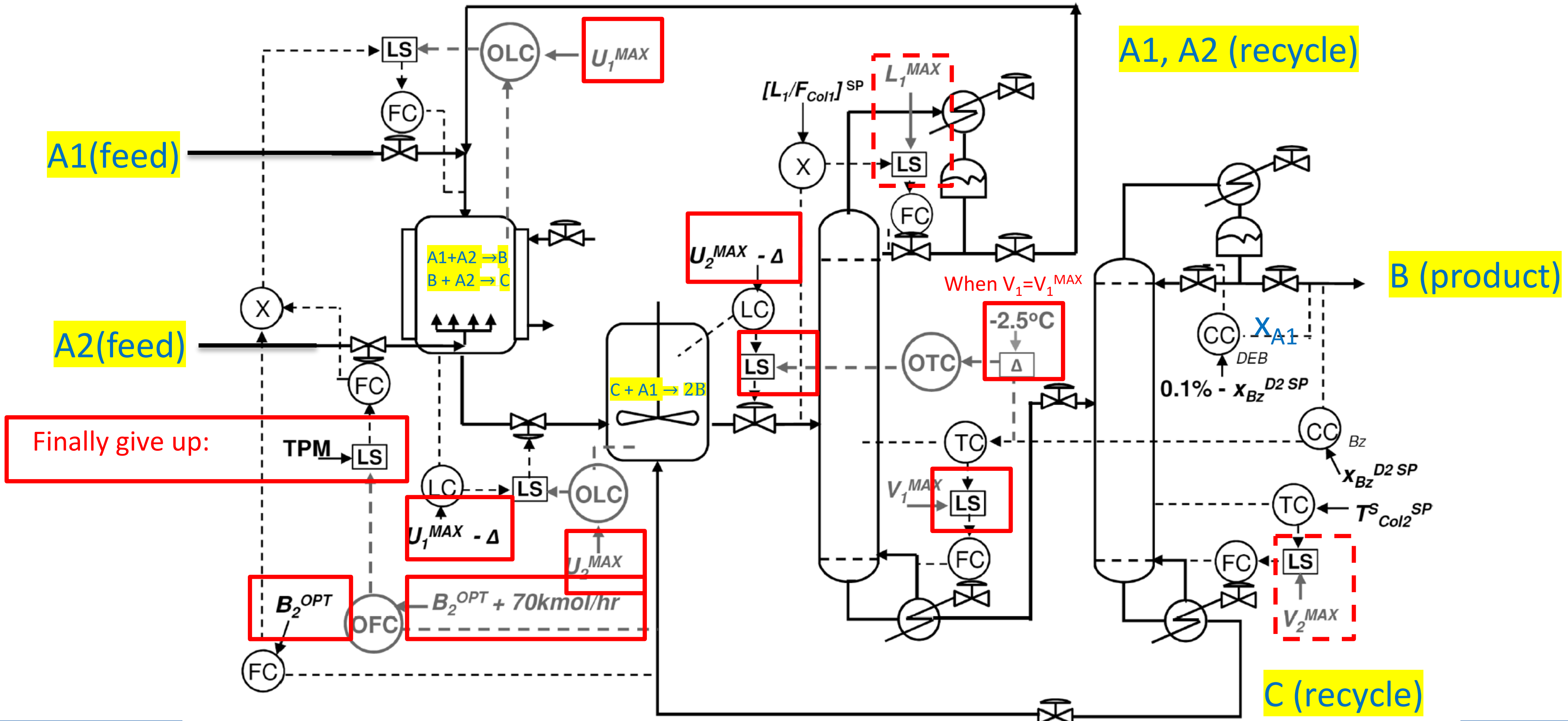
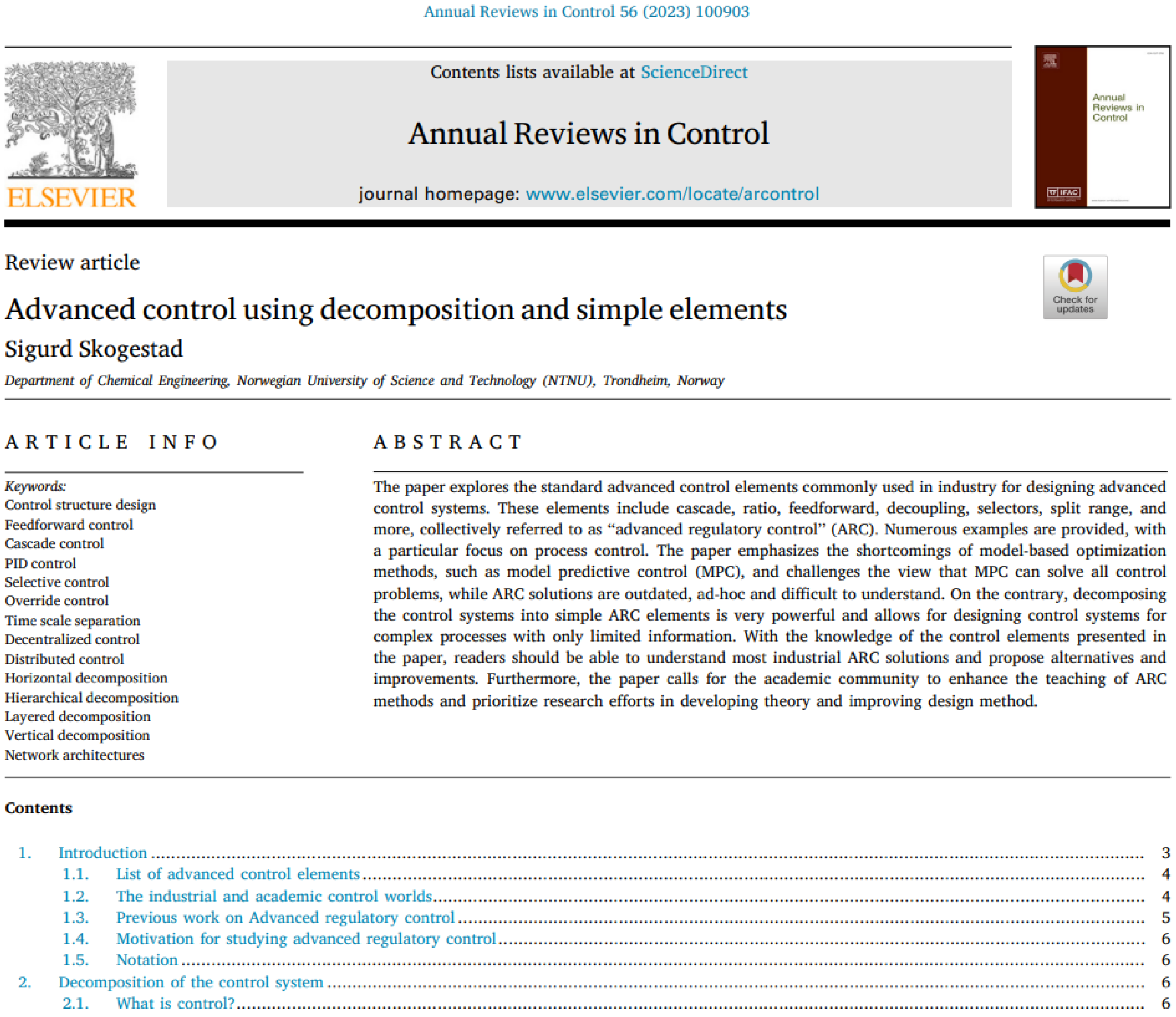


Figure 7. CS2 with overrides for handling equipment capacity constraints.

Conclusion Advanced process control (APC)

- Classical APC, aka «Advanced regulatory control» (ARC) or «Advanced PID»:
  - Works very well in many cases
  - Optimization by feedback (active constraint switching)
  - Need to pair input and output.
    - Advantage: The engineer can specify directly the solution
    - Problem: Unique pairing may not be possible for complex cases
  - Need model only for parts of the process (for tuning)
  - Challenge: Need better teaching and design methods

- MPC may be better (and simpler) for more complex multivariable cases
  - But MPC may not work on all problems (Bidirectional inventorycontrol)
  - Main challenge: Need dynamic model for whole process
  - Other challenge: Tuning may be difficult



Annual Reviews in Control 56 (2023) 100903

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Advanced control using decomposition and simple elements

Sigurd Skogestad

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Keywords:
Control structure design
Feedforward control
Cascade control
PID control
Selective control
Override control
Time scale separation
Decentralized control
Distributed control
Horizontal decomposition
Hierarchical decomposition
Layered decomposition
Vertical decomposition
Network architectures

ABSTRACT

The paper explores the standard advanced control elements commonly used in industry for designing advanced control systems. These elements include cascade, ratio, feedforward, decoupling, selectors, split range, and more, collectively referred to as "advanced regulatory control" (ARC). Numerous examples are provided, with a particular focus on process control. The paper emphasizes the shortcomings of model-based optimization methods, such as model predictive control (MPC), and challenges the view that MPC can solve all control problems, while ARC solutions are outdated, ad-hoc and difficult to understand. On the contrary, decomposing the control systems into simple ARC elements is very powerful and allows for designing control systems for complex processes with only limited information. With the knowledge of the control elements presented in the paper, readers should be able to understand most industrial ARC solutions and propose alternatives and improvements. Furthermore, the paper calls for the academic community to enhance the teaching of ARC methods and prioritize research efforts in developing theory and improving design method.

Contents

P8 – Extra if time

Tranformed inputs

Briefly on pro and cons of MPC

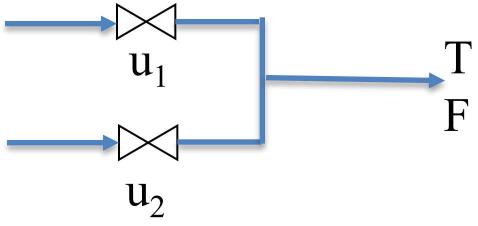
RTO

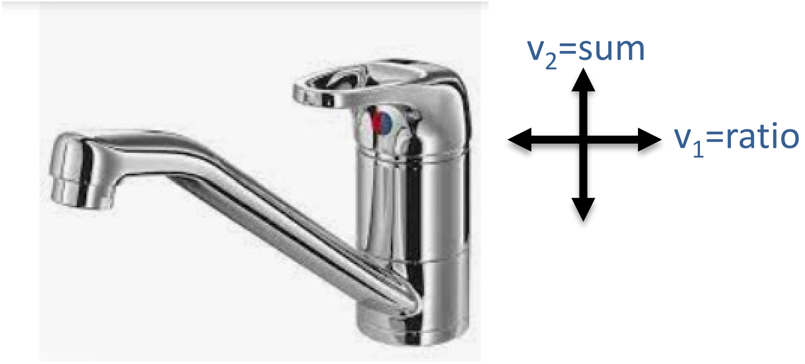
ESC

Nonlinear feedforward, decoupling and linearization

- Transformed inputs: Extremely simple and effective way of achieving feedforward, decoupling and linearization

Example decoupling: Mixing of hot (u_1) and cold (u_2) water

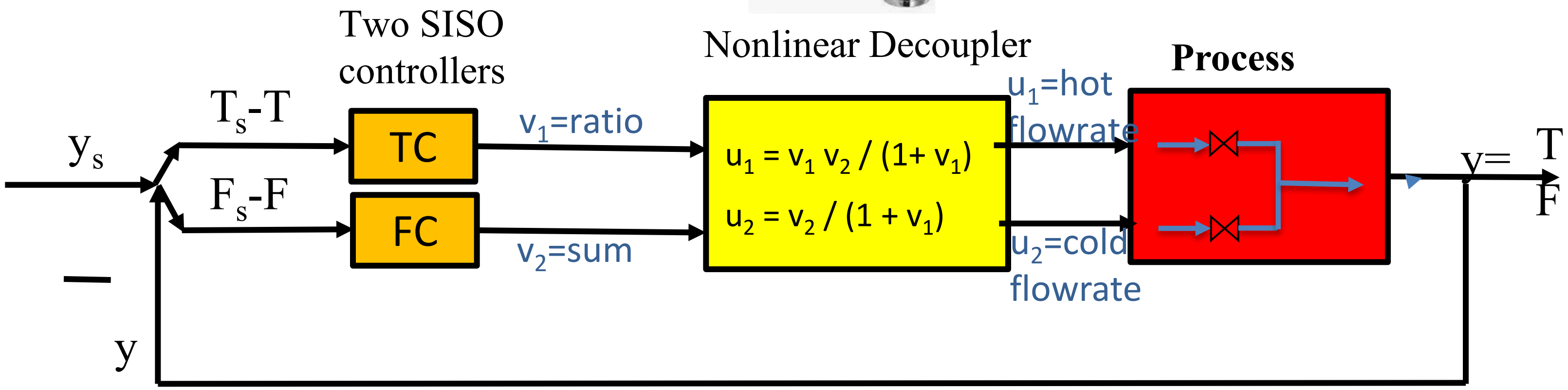




- Want to control
  - y_1 = Temperature T
  - y_2 = total flow F
- Inputs, u=flowrates
- May use two SISO PI-controllers
  - TC
  - FC
- Insight: Get decoupled response with transformed inputs
  - TC sets flow ratio, $v_1 = u_1/u_2$
  - FC sets flow sum, $v_2 = u_1 + u_2$
- Decoupler: Need «static calculation block» to solve for inputs

$$u_1 = v_1 v_2 / (1+ v_1)$$

$$u_2 = v_2 / (1 + v_1)$$



Pairings:

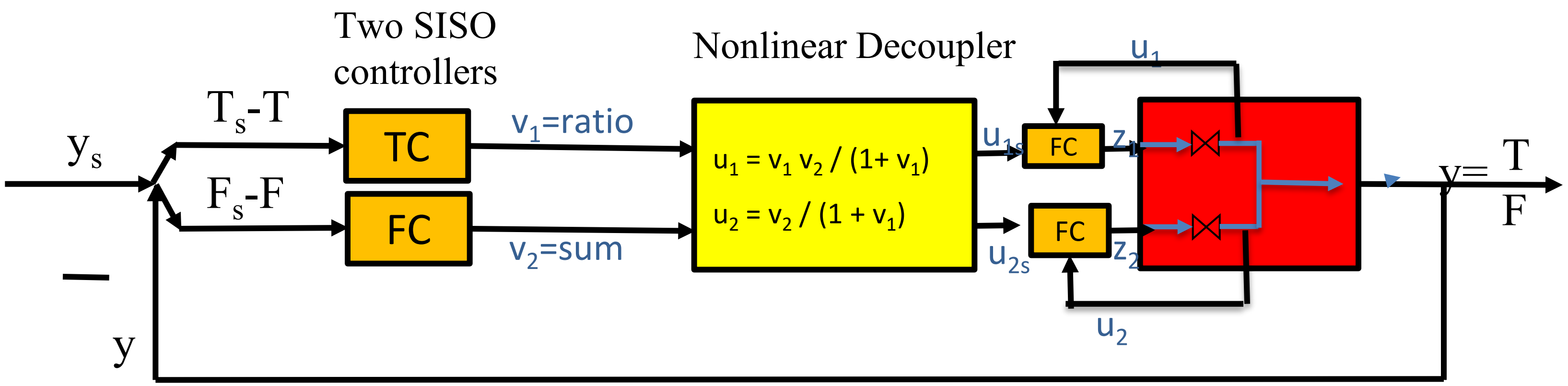
- T – v_1
- F – v_2

No interactions for setpoint change

Note:

- In practice u=valve position (z)
- So must add two flow controllers
  - These generate inverse by feedback

In practice must add two slave flow controllers



v = transformed inputs
u = flowrates
z = valve positions

Decoupler with feedforward:

$$q_h = \frac{v_2(v_1 - T_c)}{T_h - T_c}$$

$$q_c = v_2 - q_h$$

Feedforward (and decoupling) control

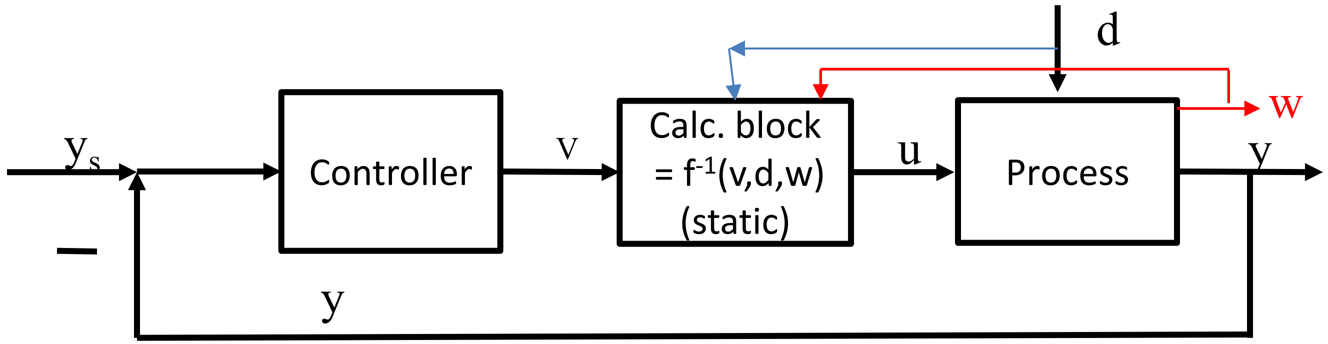
- Feedforward control relies on model
- as opposed to feedback which relies mostly on data

- Feedback control: Linear model is often OK
- Feedforward control: Much less likely that linear model is OK because of process changes and disturbances
- Here: Nonlinear feedforward control using Input transformations based on static process model

Input transformations

General approach: Combined Nonlinear decoupling, feedforward and linearization using Transformed Inputs *

- Generalization: Introduce transformed input v and use Nonlinear calculation block



Genaral Method*:

Steady-state model: $y = f(u,d,w)$

Select transformed input: $v = f(u,d,w)$ («right-hand side» of model)

Calculation block: Invert for given v: $u = f^{-1}(v,d,w)$ (may be replaced by slave v-controller)

w=dependent variable (flow, temperature), but treated as measured disturbance

w-variables may be used to simplify model

Transformed system becomes: $y=I\ v$ («decoupled, linear, indepedent of d»)

Note: To simplify often use only «parts» of f(u,d,w) as v (because of unknown parameters etc.)

*Zotica, Alsop and Skogestad. 2020 IFAC World Congress

Example: Combined nonlinear decoupling and feedforward. Mixing of hot and cold water

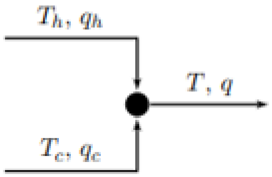
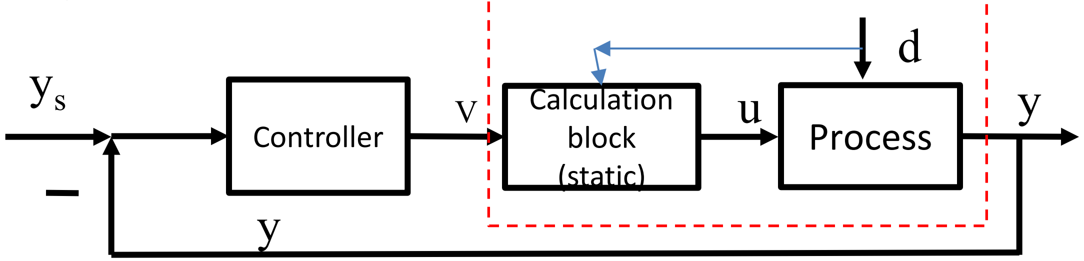


Figure 1: Mixer system



$$u = \begin{pmatrix} q_h \\ q_c \end{pmatrix}$$

$$d = \begin{pmatrix} T_h \\ T_c \end{pmatrix}$$

$$y = \begin{pmatrix} T \\ q \end{pmatrix}$$

Steady-state model written as y=f(u,d):

$$T = \frac{q_h T_h + q_c T_c}{q_h + q_c}$$

$$q = q_c + q_h$$

Select transformed inputs as right hand side, v = f

$$v_1 = \frac{q_h T_h + q_c T_c}{q_h + q_c} \quad (1)$$ Generalized ratio

$$v_2 = q_c + q_h \quad (2)$$

Model from v to y (red box) is then decoupled and with perfect disturbance rejection:

$$T = v_1$$

$$q = v_2$$

- Can then use two single-loop PI controllers for T and q!
  - These controllers are needed to correct for model errors and unmeasured disturbances
- Note that v_1 used to control T is a generalized ratio, but it includes also feedforward from Tc and Th.

Implementation (calculation block) : Solve (1) and (2) with respect to u=(qc qh):

Decoupler with feedforward:

$$q_h = \frac{v_2(v_1 - T_c)}{T_h - T_c}$$

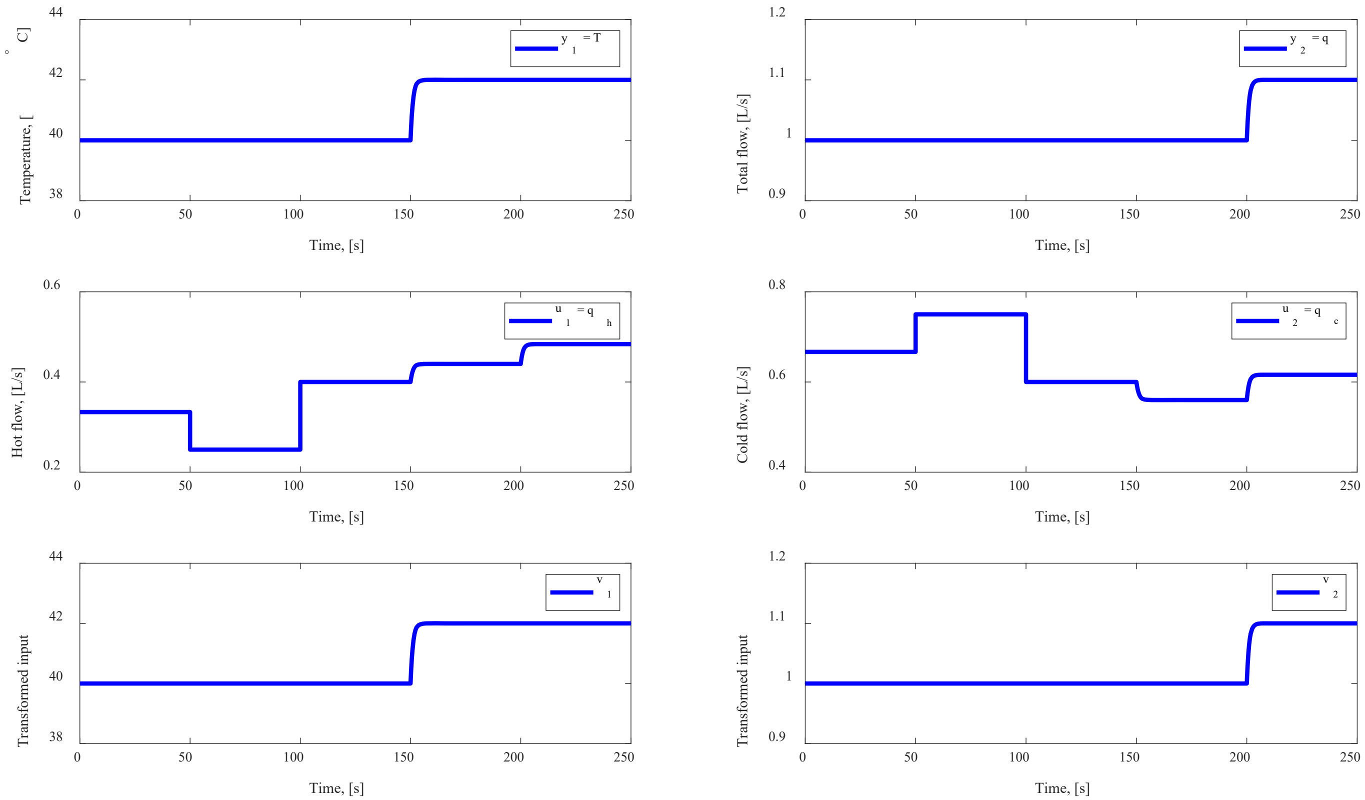
$$q_c = v_2 - q_h$$

Transformed MVs for decupling, linearization and disturbance rejection

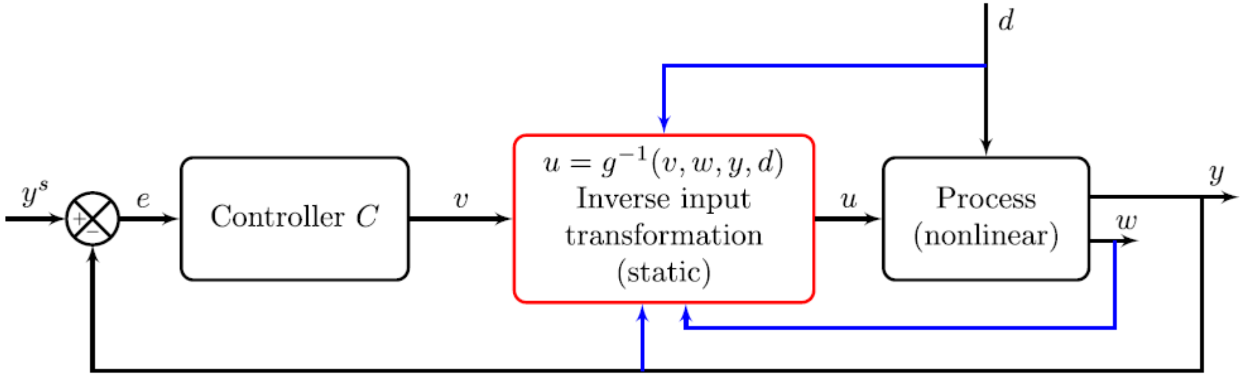
Mixing of hot and cold water (static process)
New system: $T=v_1$ and $q=v_2$

Outer loop: Two I-controllers with $\tau_C = 1$ s

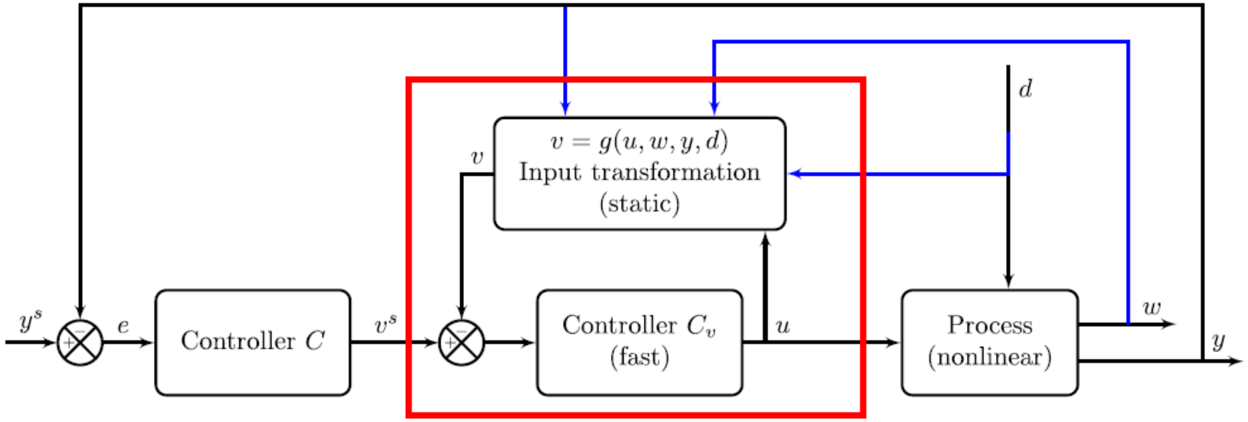
1. T_h: 60->70 °C at t = 50 s
2. T_c: 30->20 °C at t = 100 s
3. T_h^s: 40->42 °C at t = 150 s
4. q^s: 1->1.1 L/s at t = 200 s



Alternative B: Calculation block solved by feedback (using fast slave controller C_v)

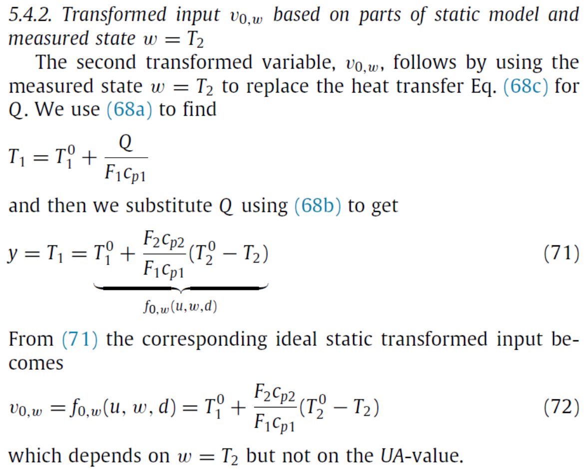


(a) Alternative A. Model-based implementation of transformed input $v = g(u, w, y, d)$. The physical input $u = g^{-1}(v, w, y, d)$ is generated by a static (algebraic) calculation block which inverts the transformed input model equations. The model-based implementation generates the exact inverse for the case with no model error.



(b) Alternative B. Feedback implementation of transformed input $v = g(u, w, y, d)$ using cascade control with a slave v-controller. The computed value of v is driven to its setpoint v_s by the inner (slave) feedback controller C_v which generates the physical input u. This implementation generates an approximate inverse.

Example: Power control



5.4.2. Transformed input $v_{0,w}$ based on parts of static model and measured state $w = T_2$

The second transformed variable, $v_{0,w}$, follows by using the measured state $w = T_2$ to replace the heat transfer Eq. (68c) for Q. We use (68a) to find

$$T_1 = T_1^0 + \frac{Q}{F_1 c_{p1}}$$

and then we substitute Q using (68b) to get

$$y = T_1 = \underbrace{T_1^0 + \frac{F_2 c_{p2}}{F_1 c_{p1}}(T_2^0 - T_2)}_{f_{0,w}(u,w,d)} \qquad (71)$$

From (71) the corresponding ideal static transformed input becomes

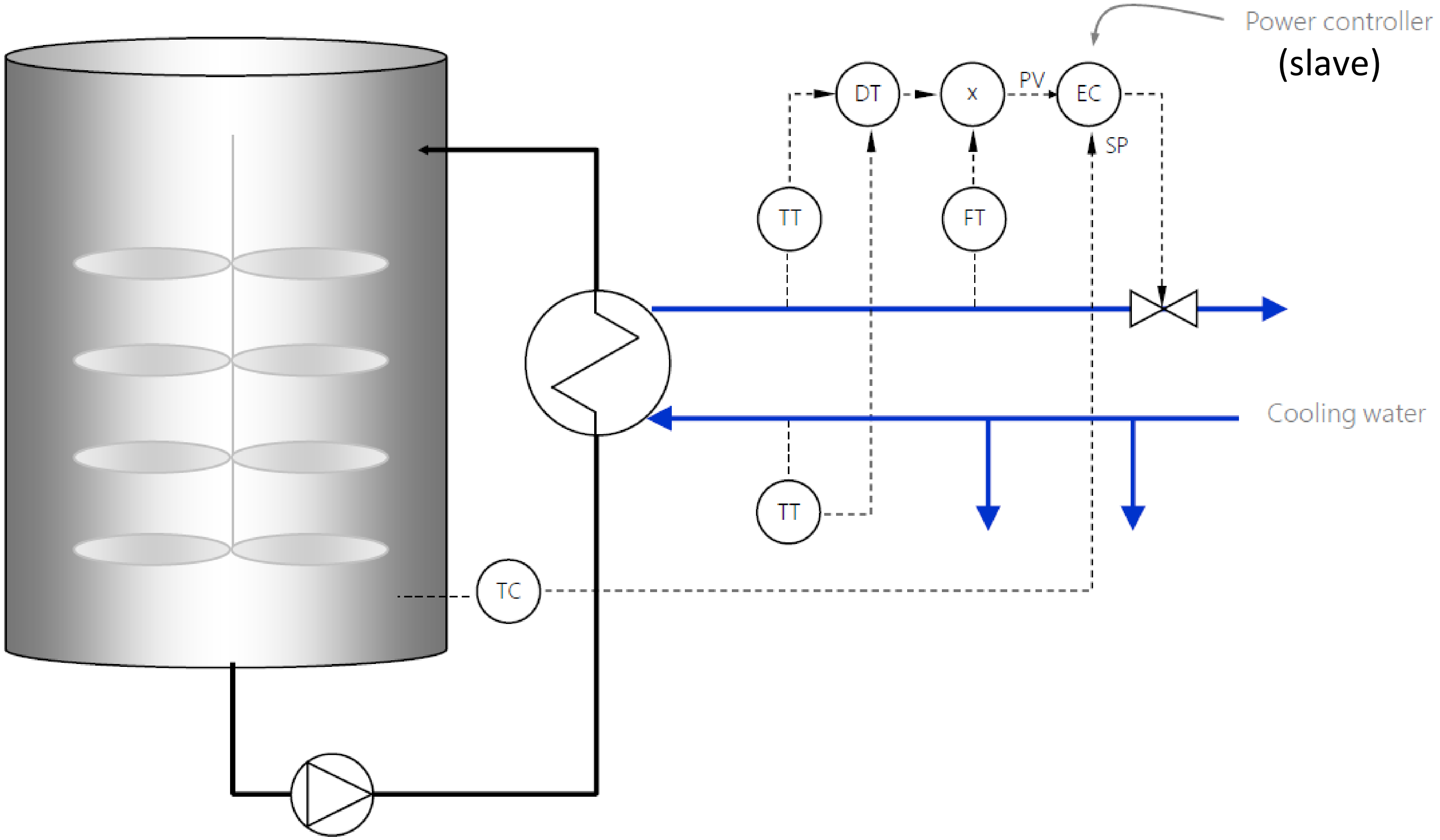
$$v_{0,w} = f_{0,w}(u, w, d) = T_1^0 + \frac{F_2 c_{p2}}{F_1 c_{p1}}(T_2^0 - T_2) \qquad (72)$$

which depends on $w = T_2$ but not on the *UA*-value.

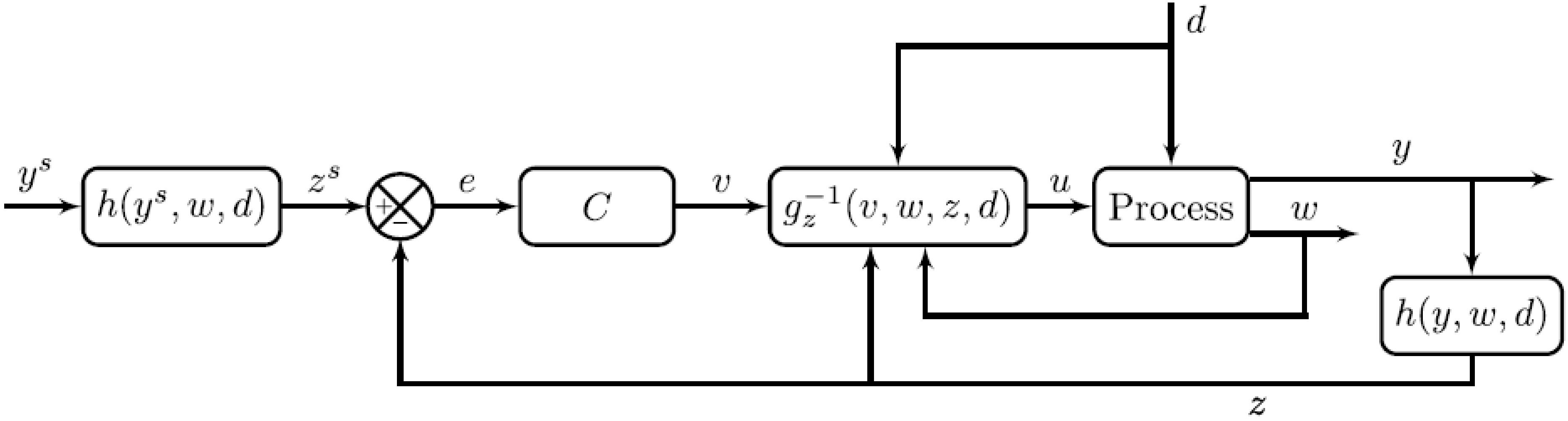
In practice (Perstorp) use only part of this:

$v=F_2(T_2^0 – T_2)$

New control structure: Power control



Also: Transformed outputs z



(a) General implementation of transformed output z

- No fundamental advantage, but can simplify input transformation
- For example, y=T, z=H (enthalpy)

More on transformed inputs

Journal of Process Control 122 (2023) 113–133

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Review

Transformed inputs for linearization, decoupling and feedforward control

Sigurd Skogestad [a,*], Cristina Zotică [a], Nicholas Alsop [b]

[a] Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491, Trondheim, Norway
[b] Senior Process Control Engineer, Borealis AB, Stenungsund, Sweden

MPC and RTO

What about MPC?

- First industrial use in the 1970s
- Became common in the refining and petrochemical industry in the 1980s
- In the 1990s a bright future was predicted for MPC in all process industries (chemical, thermal power, …)
- 30 years later: We know that this did not happen
- Why? First, the performance benefits of MPC compared to ARC are often minor (if any)
- In addition, MPC has some limitations
    1. Expensive to obtain model
    2. Does not easily handle integral action, cascade and ratio control
    3. Normally, cannot be used at startup (so need ARC anyway)
    4. Can be difficult to tune. Difficult to incorporate fast control tasks (because of centralized approach)
    5. Computations can be slow
    6. Robustness (e.g., gain margin) handled indirectly
- Advantages of MPC
    1. Very good for interactive multivariable dynamic processes
    2. Coordinates feedforward and feedback
    3. Coordinates use of many inputs
    4. Makes use of information about future disturbances, setpoints and prices (predictive capabilities of MPC)
    5. Can handle nonlinear dynamic processes (nonlinear MPC)
- What about constraints
    - Not really a major advantage with MPC; can be handled well also with ARC

MPC = model predictive control
ARC = advanced regulatory control

7.6.7. *Summary of MPC shortcomings*

Some shortcomings of MPC are listed below, in the expected order of importance as seen from the user's point of view:

1. MPC requires a "full" dynamic model involving all variables to be used by the controller. Obtaining and maintaining such a model is costly.
2. MPC can handle only indirectly and with significant effort from the control engineer (designer), the three main inventions of process control; namely integral control, ratio control and cascade control (see above).
3. Since a dynamic model is usually not available at the startup of a new process plant, we need initially a simpler control system, typically based on advanced regulatory control elements. MPC will then only be considered if the performance of this initial control system is not satisfactory.
4. It is often difficult to tune MPC (e.g., by choosing weights or sometimes adjusting the model) to give the engineer the desired response. In particular, since the control of all variables is optimized simultaneously, it may be difficult to obtain a solution that combines fast and slow control in the desired way. For example, it may be difficult to tune MPC to have fast feedforward control for disturbances because it may affect negatively the robustness of the feedback part (Pawlowski et al., 2012).
5. The solution of the online optimization problem is complex and time-consuming for large problems.
6. Robustness to model uncertainty is handled in an ad hoc manner, for example, through the use of the input weight R. On the other hand, with the SIMC PID rules, there is a direct relationship between the tuning parameter τ_c and robustness margins, such as the gain, phase and delay margin Grimholt and Skogestad (2012), e.g., see (C.13) for the gain margin.

7.6.8. *Summary of MPC advantages*

The above limitations of MPC, for example, with respect to integral action, cascade control and ratio control, do not imply that MPC will not be an effective solution in many cases. On the contrary, MPC should definitely be in the toolbox of the control engineer. First, standard ratio and cascade control elements can be put into the fast regulatory layer and the setpoints to these elements become the MVs for MPC. More importantly, MPC is usually better (both in terms of performance and simplicity) than advanced regulatory control (ARC) for:

1. Multivariable processes with (strong) dynamic interactions.
2. Pure feedforward control and coordination of feedforward and feedback control.
3. Cases where we want to dynamically coordinate the use of many inputs (MVs) to control one CV.
4. Cases where future information is available, for example, about future disturbances, setpoint changes, constraints or prices.
5. Nonlinear dynamic processes (nonlinear MPC).

The handling of constraints is often claimed to be a special advantage of MPC, but it can it most cases also be handled well by ARC (using selectors, split-range control solutions, anti-windup, etc.). Actually, for the Tennessee Eastman Challenge Process, Ricker (1996) found that ARC (using decentralized PID control) was better than MPC. Ricker (1996) writes in the abstract: "There appears to be little, if any, advantage to the use of NMPC (nonlinear MPC) in this application. In particular, the decentralized strategy does a better job of handling constraints – an area in which NMPC is reputed to excel". In the discussion section he adds: "The reason is that the TE problem has too many competing goals and special cases to be dealt with in a conventional MPC formulation".

Optimal operation and constraints switching

- We have presented effective decentralized approaches for constraint switching (MV-MV, CV-CV, MV-CV).
  - Optimal in many cases, but not in general
  - For example, may not be able to cover cases with more than one unconstrained region ⇒ More than one self-optimizing variable

- An alternative is model-based RTO, usually based on static model

RTO = Real-time optimization

Economic real-time optimization(RTO)
Alternative RTO approaches:

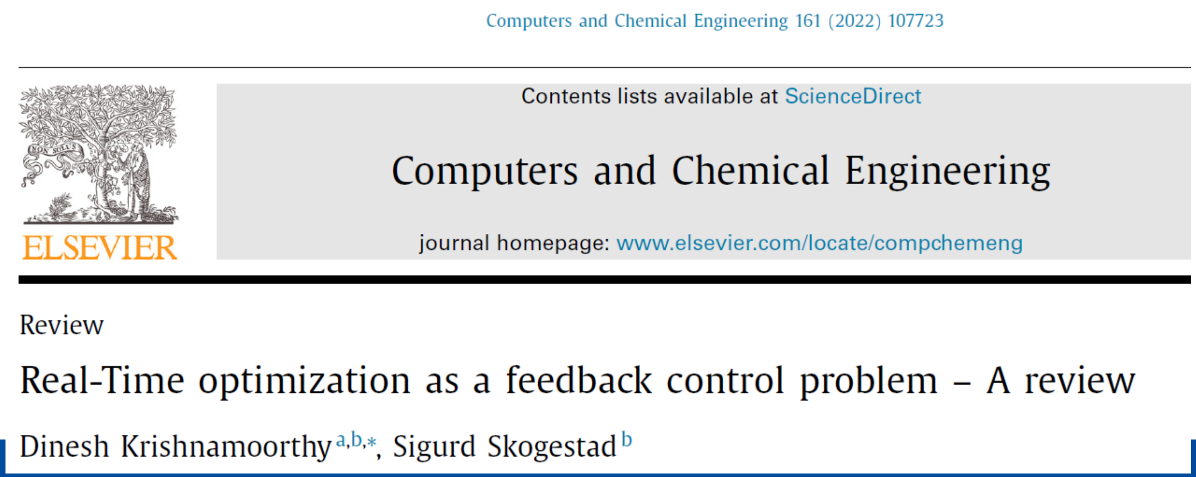
Model-based

I. Separate RTO layer (online dynamic or steady-state optimization)

II. Feedback-optimizing control (put optimization into control layer)
  - Alt.1. (Most general): Based on dual decomposition (iterate on Lagrange multipliers λ)
  - Alt.2 (Tighter constraint control): Region-based with reduced gradient

Data-based

III. Hill-climbing methods = Extremum-seeking control (model free. But need to measure cost J)



Computers and Chemical Engineering 161 (2022) 107723

ELSEVIER

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Review

Real-Time optimization as a feedback control problem – A review

Dinesh Krishnamoorthy[a,b,*], Sigurd Skogestad[b]

I. Conventional (commercial) steady-state RTO

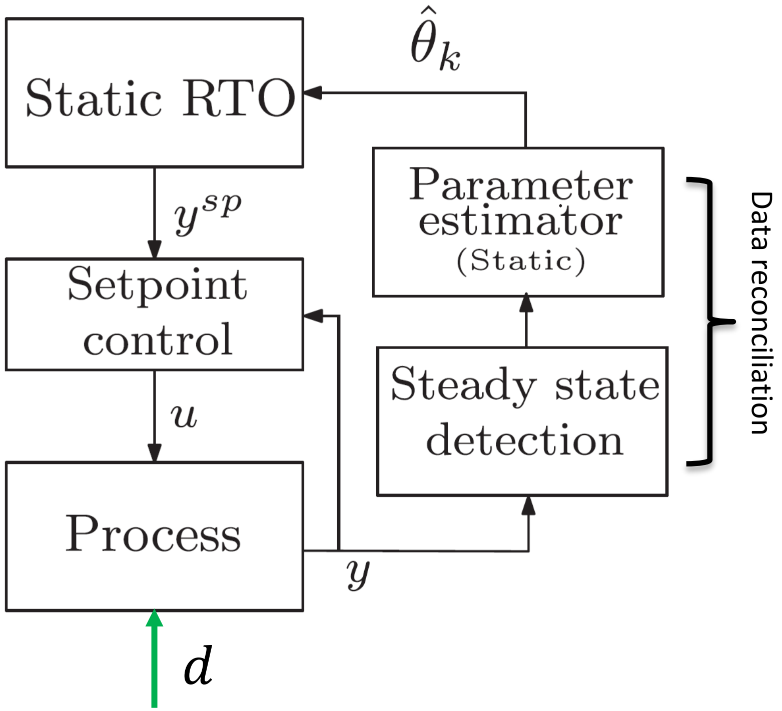
Fairly common in refining and petrochemical industy.

Two-step approach:

Step 1. “Data reconciliation”:

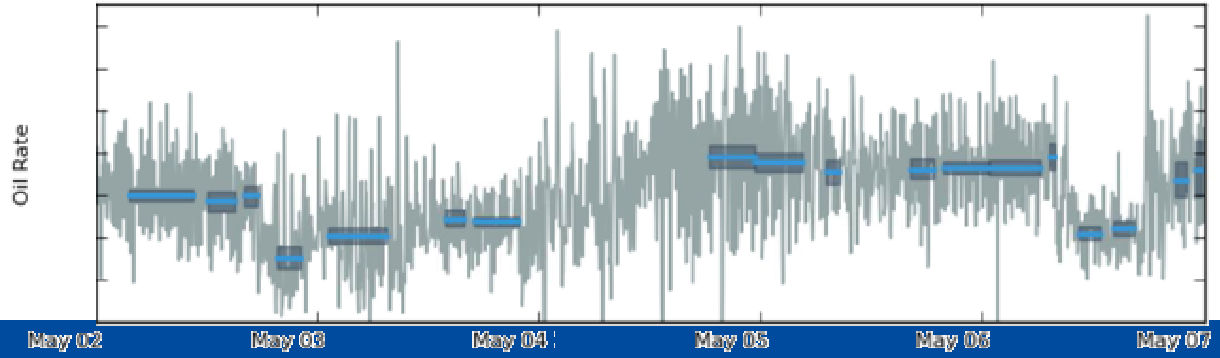
- Steady-state detection
- Update estimate of d: model parameters, disturbances (feed), constraints

Step 2. Re-optimize to find new optimal steady state



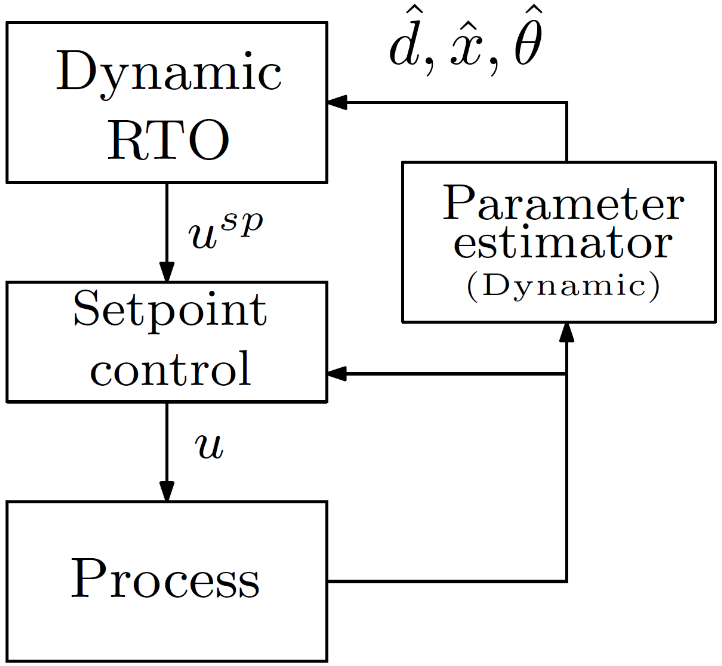
Steady-state wait time

- Transient measurements cannot be used → system must "settle"
- Large chunks of data discarded
- Steady state detection issues
  - Erroneously accept transient data
  - Non-stationary drifts



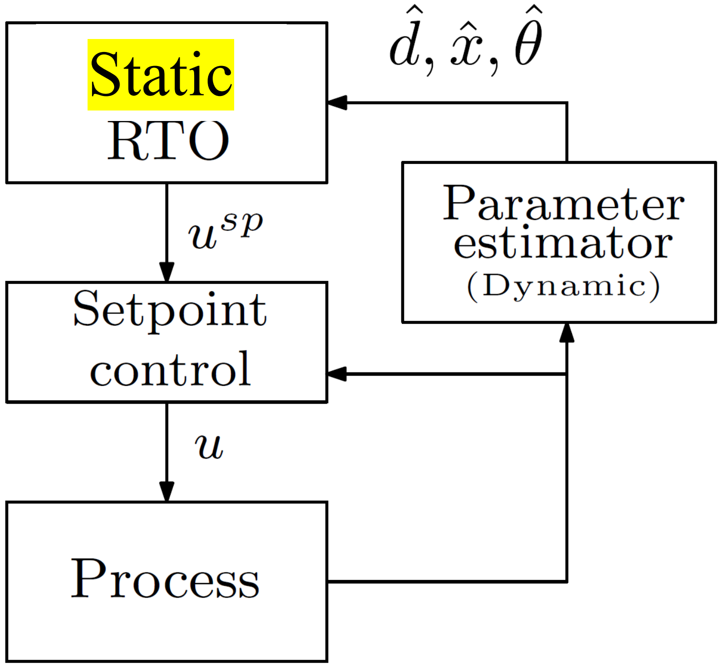
How to avoid steady state wait time?

1. Dynamic RTO = EMPC



How to avoid steady state wait time?

2. Hybrid RTO



RTO problem

Steady-state RTO (used in Hybrid RTO):

$$\min_{x,u} J(x,d,u)$$

s.t.:

$$\begin{gathered} 0 = F(x,d,u) \\ 0 = h(x,d,u) \\ g(x,d,u) \leq 0 \end{gathered}$$

Dynamic RTO ≡ (Economic) nonlinear MPC **:**

$$\min_{x(t),u(t)} \int_{t_0}^{t_f} J\big(x(t),d(t),u(t)\big)\,dt$$

s.t.:

$$\begin{gathered} \dot{x}(t) = F\big(x(t),d(t),u(t)\big) \\ 0 = h\big(x(t),d(t),u(t)\big) \\ g\big(x(t),d(t),u(t)\big) \leq 0 \\ x(t_0) = \hat{x}_0 \end{gathered}$$

Now we calculate not only an optimal point, but an optimal trajectory!'

BUT Much more complex that static RTO, and may not give much economic benefit

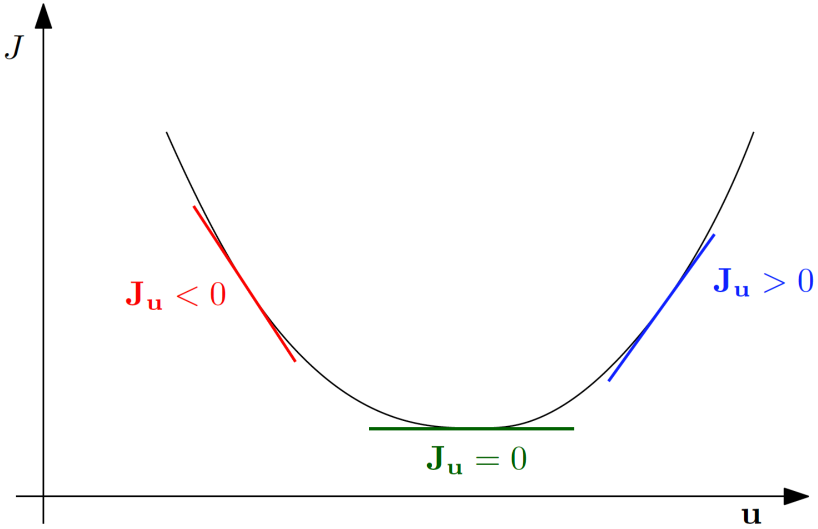
II. Feedback RTO (unconstrained case)

«Solving RTO-problem using PI control»

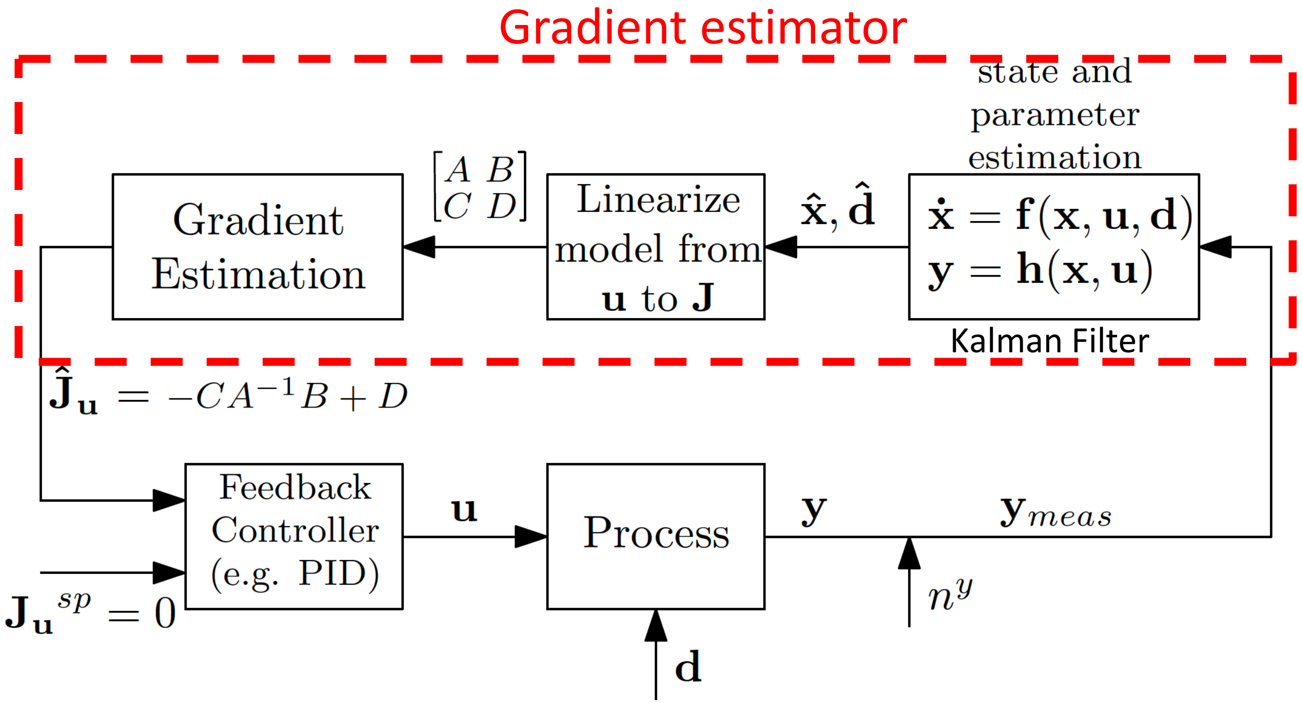
Unconstrained optimization.

Necessary condition of optimality (NCO):

- Gradient of cost function = 0
- $J_u \equiv \frac{\partial J}{\partial u} \equiv \nabla_u J = 0$



IIA. Feedback RTO (unconstrained case)



Linearize the dynamic model

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \\ J &= \mathbf{g}(\mathbf{x}, \mathbf{u})\end{aligned} \quad \Rightarrow \quad \begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ J &= C\mathbf{x} + D\mathbf{u}\end{aligned}$$

$$A = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad B = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

$$C = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad D = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

Trick, set $\dot{x} = 0, to\ get\ estimate\ of\ static\ gradient:$

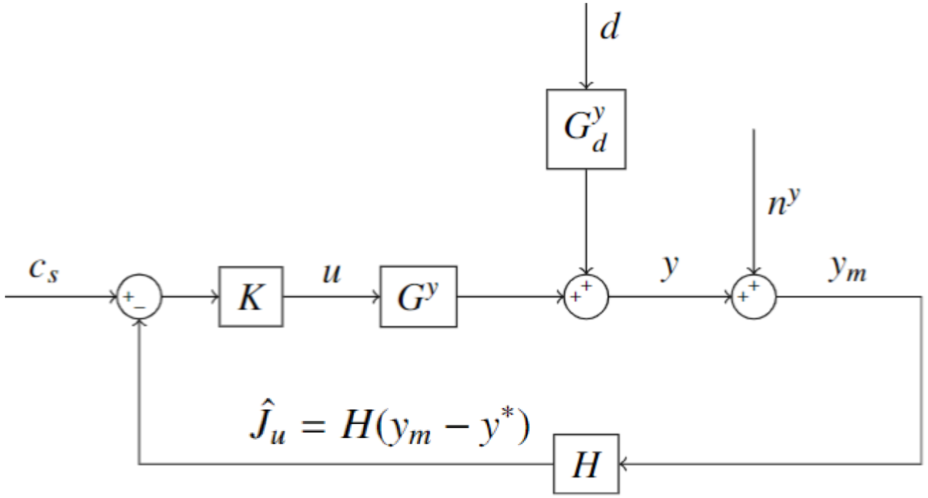
$$J = \underbrace{\left(-CA^{-1}B + D\right)}_{\hat{\mathbf{J}}_\mathbf{u}}\mathbf{u}$$

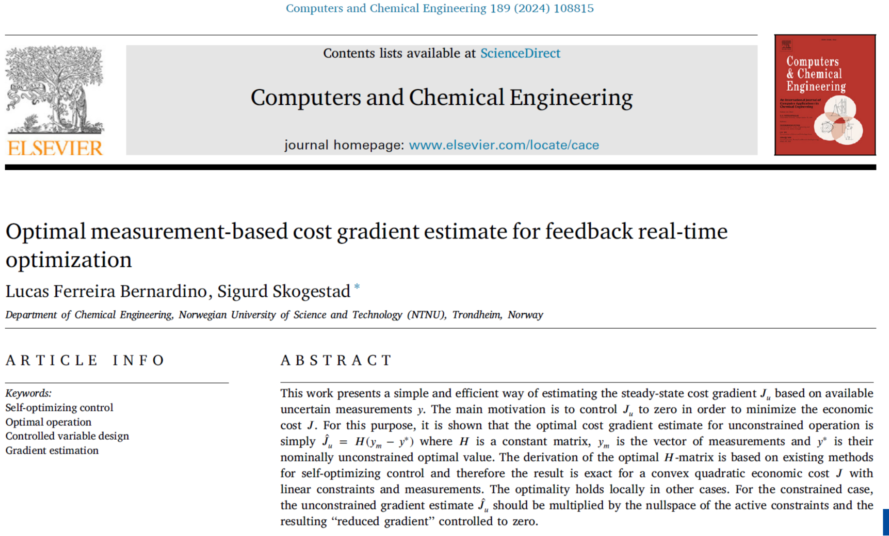
Note: This is one simple way of doing the gradient estimation, but needfs dynamic model (Kalman Filter)

D Krishnamoorthy, E Jahanshahi, S Skogestad. Feedback Real-Time Optimization Strategy Using a Novel Steady-state Gradient Estimate and Transient Measurements. Industrial & Engineering Chemistry Research, 2019

Here is another Static gradient estimation:

Based on self-optimizing control. Very simple and works well!




Computers and Chemical Engineering 189 (2024) 108815

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Optimal measurement-based cost gradient estimate for feedback real-time optimization

Lucas Ferreira Bernardino, Sigurd Skogestad *

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Keywords:
Self-optimizing control
Optimal operation
Controlled variable design
Gradient estimation

ABSTRACT

This work presents a simple and efficient way of estimating the steady-state cost gradient J_u based on available uncertain measurements y. The main motivation is to control J_u to zero in order to minimize the economic cost J. For this purpose, it is shown that the optimal cost gradient estimate for unconstrained operation is simply $\hat{J}_u = H(y_m - y^*)$ where H is a constant matrix, y_m is the vector of measurements and y^* is their nominally unconstrained optimal value. The derivation of the optimal H-matrix is based on existing methods for self-optimizing control and therefore the result is exact for a convex quadratic economic cost J with linear constraints and measurements. The optimality holds locally in other cases. For the constrained case, the unconstrained gradient estimate $\hat{J}_u$ should be multiplied by the nullspace of the active constraints and the resulting "reduced gradient" controlled to zero.

From «exact local method» of self-optimizing control:

$$H^J = J_{uu}\left[G^{yT}\left(\tilde{F}\tilde{F}^T\right)^{-1}G^y\right]^{-1}G^{yT}\left(\tilde{F}\tilde{F}^T\right)^{-1}$$

$$\text{where } \tilde{F} = [FW_d \quad W_{n^y}] \text{ and } F = \frac{dy^{opt}}{dd} = G_d^y - G^y J_{uu}^{-1} J_{ud}.$$

- Bernardino and Skogestad, Optimal measurement-based cost gradient estimate for real-time optimization, Comp. Chem. Engng., 2024

With constraints

Constrained optimization problem

$$\min_{\mathbf{u}} \; J\,(\mathbf{u}, \mathbf{y}, \mathbf{d})$$

$$\text{s.t.} \quad \mathbf{g}\,(\mathbf{u}, \mathbf{y}, \mathbf{d}) \le 0$$

Solution: Turn into unconstrained optimization problem using Lagrange multipliers

$$\mathcal{L}(\mathbf{u}, \mathbf{y}, \mathbf{d}, \lambda) = J\,(\mathbf{u}, \mathbf{y}, \mathbf{d}) + \lambda^{T}\mathbf{g}\,(\mathbf{u}, \mathbf{y})$$

$\min_{u,\lambda}$ L

u = primal variables = inputs

λ ≥ 0 = dual variables = Lagrange multipliers = shadow prices

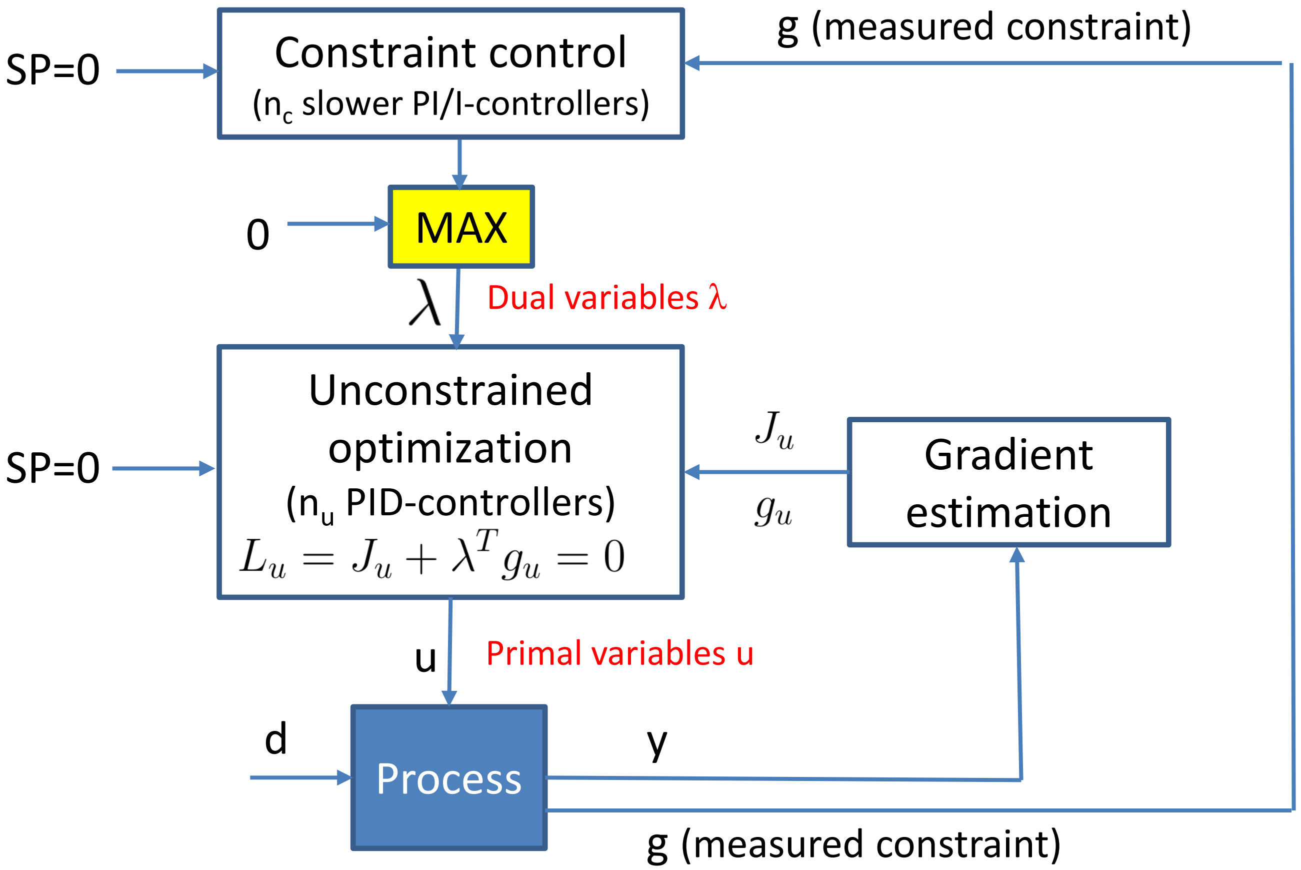
Necessary conditions of optimality (KKT-conditions)

$$\nabla_u \mathcal{L} = 0, \quad \lambda \ge 0, \quad g \cdot \lambda = 0$$

(complementary condition)

$$\boxed{L_u = J_u + \lambda^T g_u = 0}$$

A. Primal-dual control based on KKT conditions: Feedback solution that automatically tracks active constraints by adjusting Lagrange multipliers (= shadow prices = dual variables) λ



SP=0

Constraint control

(n_c slower PI/I-controllers)

g (measured constraint)

0

MAX

λ

Dual variables λ

SP=0

Unconstrained optimization

(n_u PID-controllers)

$L_u = J_u + \lambda^T g_u = 0$

J_u

g_u

Gradient estimation

u

Primal variables u

d

Process

y

g (measured constraint)

KKT: $L_u = J_u + \lambda^T g_u = 0$

Inequality constraints: $\lambda \geq 0$

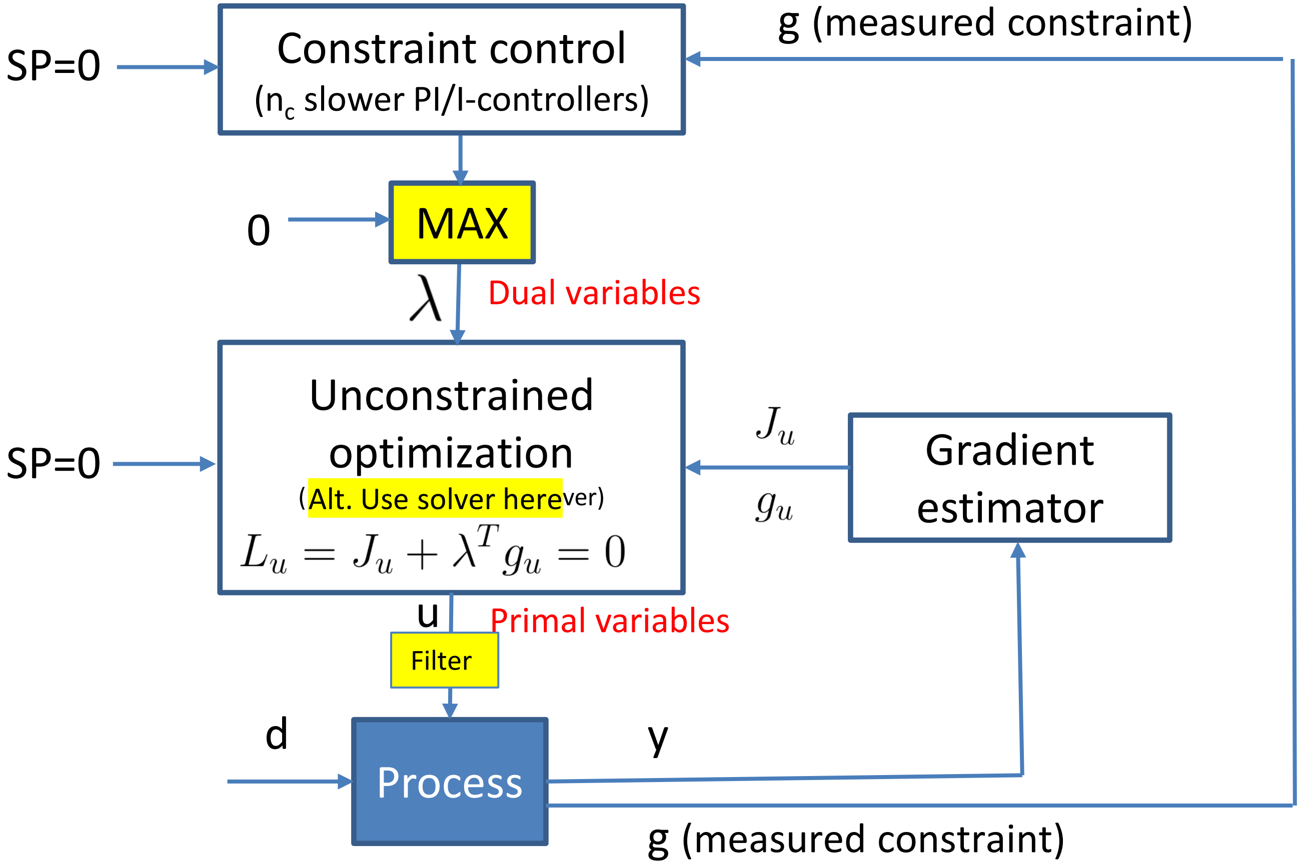
Primal-dual feedback control.

- Makes use of «dual decomposition» of KKT conditions
- Selector on dual variables λ
- Problem: Constraint control using dual variables is on slow time scale (upper layer)
  - Can be fixed using override at bottom of hiearchy (Dirza)
- Problem 2: Single-loop PID control in lower layer (L_u=0) may not be possible for coupled processes so may need to use Solver.

- D. Krishnamoorthy, A distributed feedback-based online process optimization framework for optimal resource sharing, J. Process Co
- R. Dirza and S. Skogestad . Primal–dual feedback-optimizing control with override for real-time optimization. J. Process Control, Vol. 138 (2024), 103208.

Alternative: Dual composition with optimization/solver for computing u (primal variables)

- May need to add filter to avoid instability



g (measured constraint)

SP=0

Constraint control

(n_c slower PI/I-controllers)

0

MAX

λ

Dual variables

SP=0

Unconstrained optimization

(Alt. Use solver herever)

$L_u = J_u + \lambda^T g_u = 0$

J_u

g_u

Gradient estimator

u

Primal variables

Filter

d

Process

y

g (measured constraint)

Alternative: Direct control of constraints

KKT: $L_u = J_u + \lambda^T g_u = 0$

Introduce N: $N^T g_u = 0$

Control

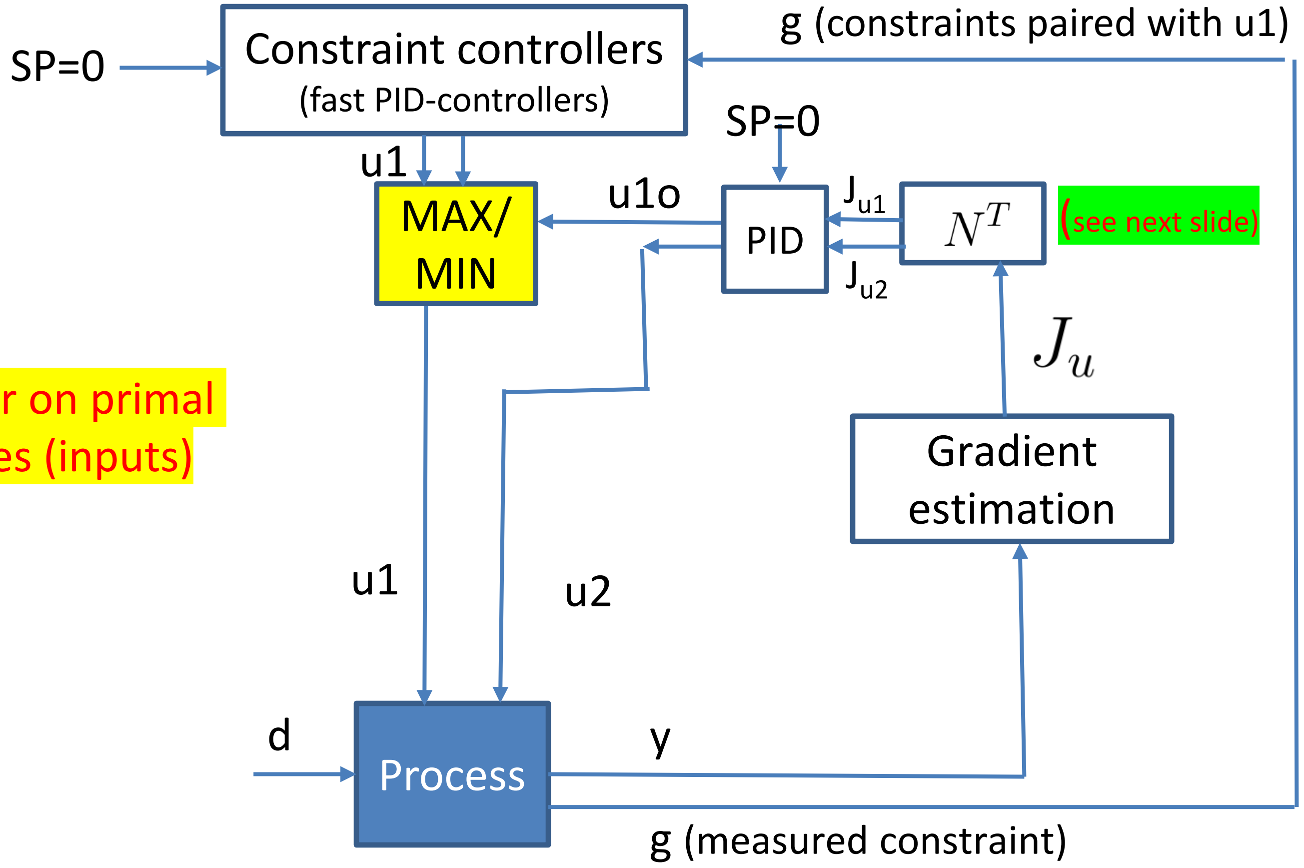
1. Active constraints $g_A = 0$.
2. Reduced gradient $N_A{}^T J_u = 0$
   - for the remaining inbconstrained degrees of freedom
   - «self-optimizing variables»

Seems easy. But how do we handle changes in constraints?

- Because g_A and N_A varies
- Originally, I thought we need a new control structure (with pairings) in each region

- Jaschke and Skogestad, «Optimal controlled variables for¨ polynomial systems». S., J. Process Control, 2012
- D. Krishnamoorthy and S. Skogestad, «Online Process Optimization with Active Constraint Set Changes using Simple Control Structure», I&EC Res., 2019

B. Region-based feedback solution with «direct» constraint control



SP=0

Constraint controllers
(fast PID-controllers)

g (constraints paired with u1)

u1

SP=0

MAX/
MIN

u1o

PID

J_{u1}

J_{u2}

N^T

(see next slide)

J_u

Gradient estimation

u1

u2

d

Process

y

g (measured constraint)

- Selector on primal variables (inputs)

KKT: $L_u = J_u + \lambda^T g_u = 0$

Introduce N: $N^T g_u = 0$

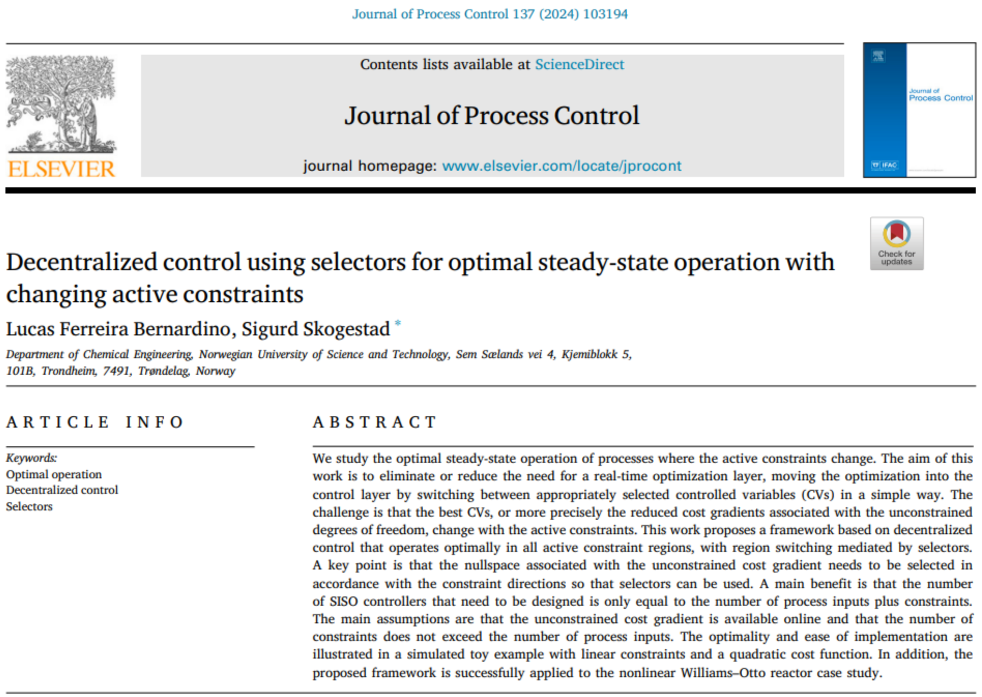
- Selector on primal variables (inputs)
- Similar to selectors in ARC
- Limitation: need to pair each constraint with an input u, may not work if many constraints

- Jaschke and Skogestad, «Optimal controlled variables for¨ polynomial systems». S., J. Process Control, 2012
- D. Krishnamoorthy and S. Skogestad, «Online Process Optimization with Active Constraint Set Changes using Simple Control Structure», I&EC Res., 2019
- L. Bernadino and S. Skogestad, Decentralized control using selectors for optimal steady-state operation with active constraints, J. Proc. Control, 2024

Feedback RTO with constraints

Assume: Have at least as many inputs as constraints

Can them have fixed pairings between constraints and unconstrained CVs! (with N is fixed)



Journal of Process Control 137 (2024) 103194

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Decentralized control using selectors for optimal steady-state operation with changing active constraints

Lucas Ferreira Bernardino, Sigurd Skogestad *

Department of Chemical Engineering, Norwegian University of Science and Technology, Sem Sælands vei 4, Kjemiblokk 5, 101B, Trondheim, 7491, Trøndelag, Norway

ARTICLE INFO

Keywords:
Optimal operation
Decentralized control
Selectors

ABSTRACT

We study the optimal steady-state operation of processes where the active constraints change. The aim of this work is to eliminate or reduce the need for a real-time optimization layer, moving the optimization into the control layer by switching between appropriately selected controlled variables (CVs) in a simple way. The challenge is that the best CVs, or more precisely the reduced cost gradients associated with the unconstrained degrees of freedom, change with the active constraints. This work proposes a framework based on decentralized control that operates optimally in all active constraint regions, with region switching mediated by selectors. A key point is that the nullspace associated with the unconstrained cost gradient needs to be selected in accordance with the constraint directions so that selectors can be used. A main benefit is that the number of SISO controllers that need to be designed is only equal to the number of process inputs plus constraints. The main assumptions are that the unconstrained cost gradient is available online and that the number of constraints does not exceed the number of process inputs. The optimality and ease of implementation are illustrated in a simulated toy example with linear constraints and a quadratic cost function. In addition, the proposed framework is successfully applied to the nonlinear Williams–Otto reactor case study.

L. Bernadino and S. Skogestad, Decentralized control using selectors for optimal steady-state operation with active constraints, J. Proc. Control, 2024

Journal of Process Control 137 (2024) 103194

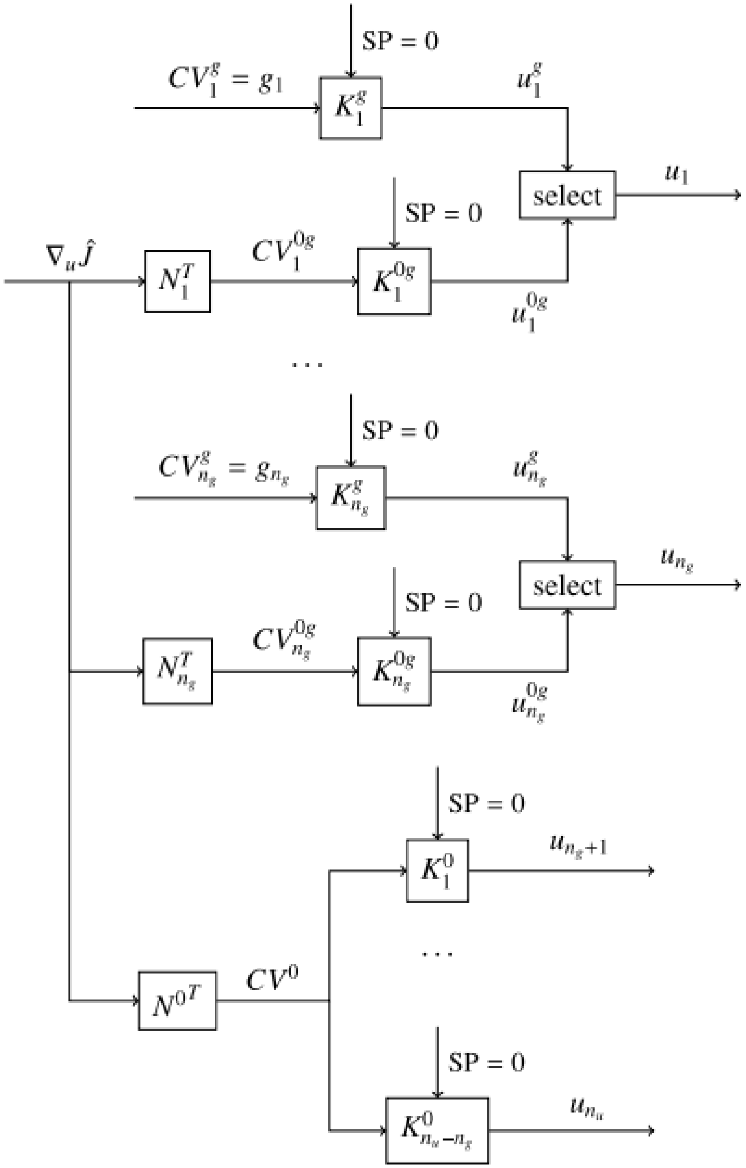


Fig. 3. Decentralized control structure for optimal operation according to Theorem 2. The "select" blocks are usually max or min selectors (see Theorem 3).

Feedback RTO WITH CONSTRAINTS

C. Region-based MPC with switching of cost function (for general case)

Standard MPC with fixed CVs: Not optimal

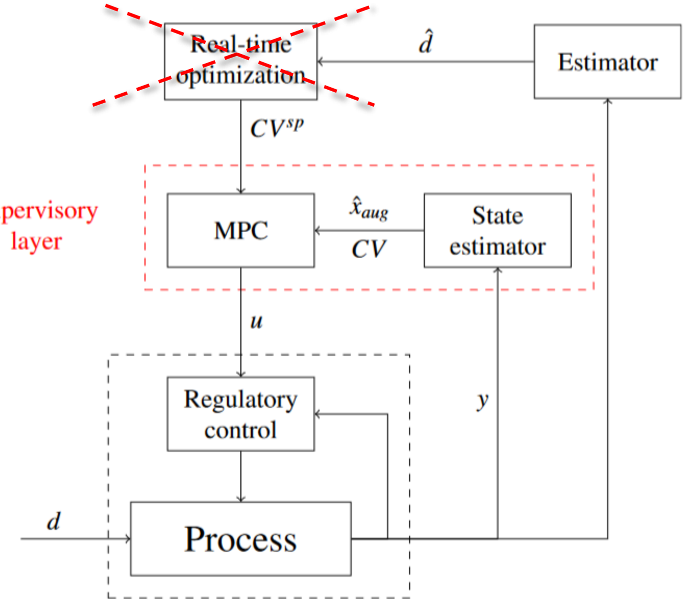


Figure 1: Typical hierarchical control structure with standard setpoint-tracking MPC in the supervisory layer. The cost function for the RTO layer is J^{ec} and the cost function for the MPC layer is J^{MPC}. With no RTO layer (and thus constant setpoints CV^{sp}), this structure is not economically optimal when there are changes in the active constraints. For smaller applications, the state estimator may be used also as the RTO estimator.

$$J^{MPC} = \sum_{k=1}^{N} \|CV_k - CV^{sp}\|_Q^2 + \|\Delta u_k\|_R^2$$

Proposed: With changing cost (switched CVs)

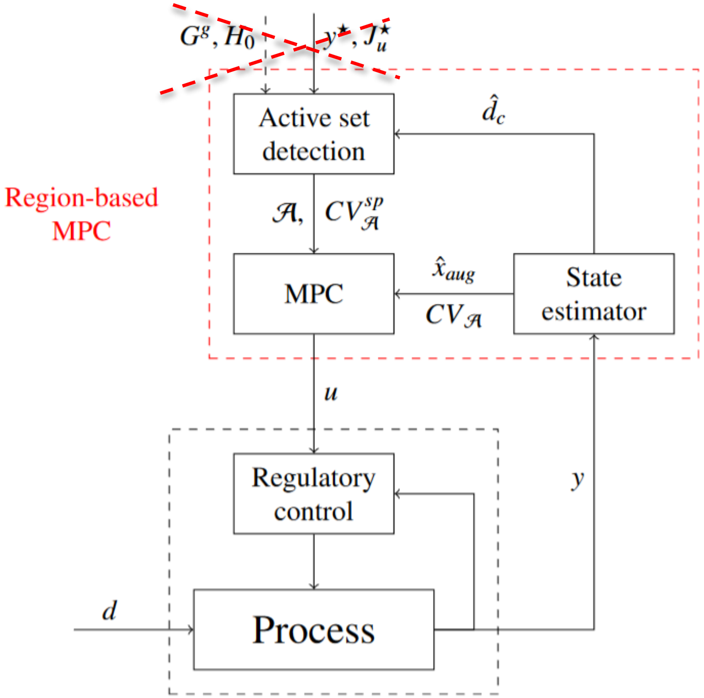


Figure 2: Proposed region-based MPC structure with active set detection and change in controlled variables. The possible updates from an upper RTO layer (y^*, J_u^* etc.) are not considered in the present work. Even with no RTO layer (and thus with constant setpoints $CV_{\mathcal{A}}^{sp}$, see (14) and (15), in each active constraint region), this structure is potentially economically optimal when there are changes in the active constraints.

$$J_{\mathcal{A}}^{MPC} = \sum_{k=1}^{N} \|CV_{\mathcal{A}} - CV_{\mathcal{A}}^{sp}\|_{Q_{\mathcal{A}}}^2 + \|\Delta u_k\|_{R_{\mathcal{A}}}^2$$

$$CV_{\mathcal{A}} = \begin{bmatrix} g_{\mathcal{A}} \\ c_{\mathcal{A}} \end{bmatrix} = \begin{bmatrix} g_{\mathcal{A}} \\ N_{\mathcal{A}}^T H_0 y \end{bmatrix} \tag{14}$$

$$H_0 = \begin{bmatrix} J_{uu} & J_{ud} \end{bmatrix} \begin{bmatrix} G^y & G_d^y \end{bmatrix}^\dagger$$

$$H^J = J_{uu} \left[G^{yT} \left(\tilde{F}\tilde{F}^T \right)^{-1} G^y \right]^{-1} G^{yT} \left(\tilde{F}\tilde{F}^T \right)^{-1}$$

- Bernardino and Skogestad, Optimal switching of MPC cost function for changing active constraints. J. Proc. Control, 2024

Model-free optimization:
Extremum Seeking Control (ESC) based on measuring cost J

$$\min_{u} J(u,d)$$

Why ESC?

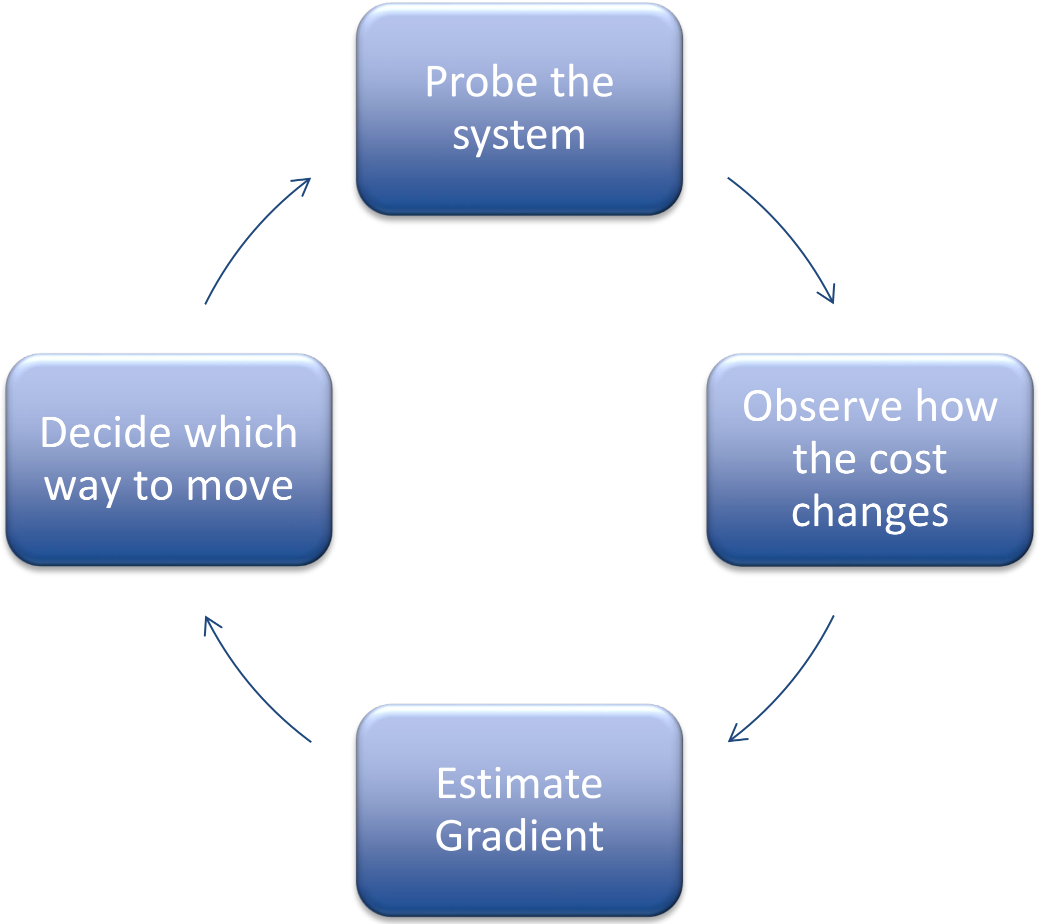
- Expensive to obtain model (for J): Use data-based ESC instead of model-based RTO
- May also be used on top of RTO
  - «Adapt» setpoint for J_u (to a nonzero bias value) to correct for model error
  - Aka «modifier adaptation»

Main problems with ESC:

- Cost function J often not measured
  - For chemical process $J = p_F F - p_P P - p_Q Q$
  - need model (!) to estimate flows F, P and utility Q
- Very slow. Typically 100 times slower than process dynamics

Data-based optimization: “Hill-climbing” / “Extremum seeking control”

Drive gradient J_u=dJ/du to zero.

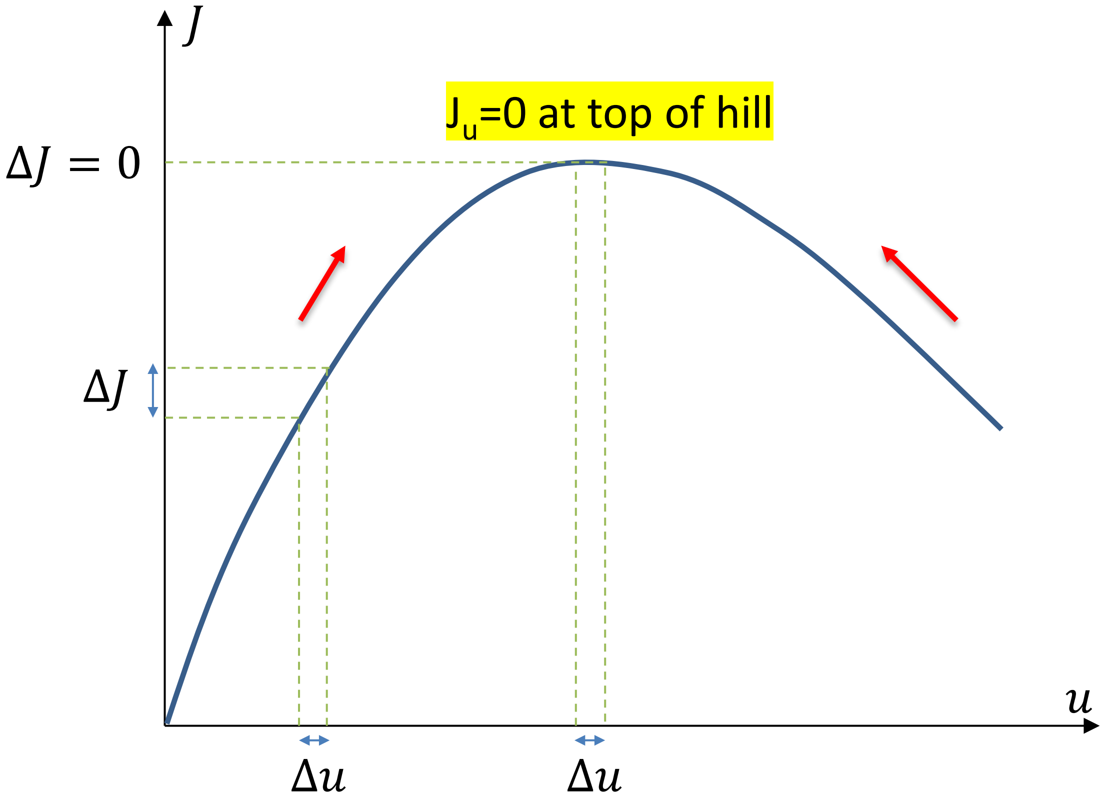


Probe the system

Observe how the cost changes

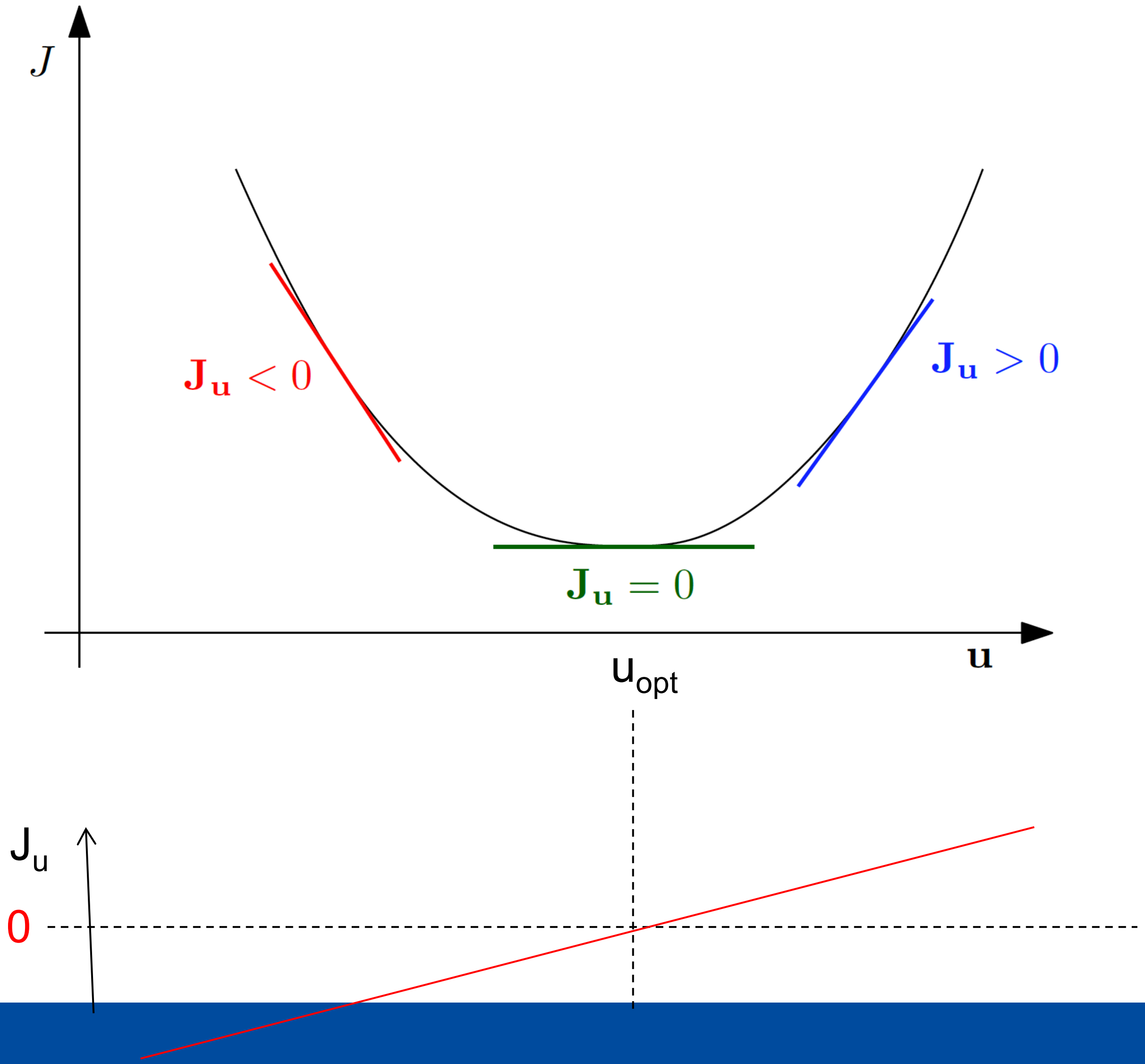
Estimate Gradient

Decide which way to move



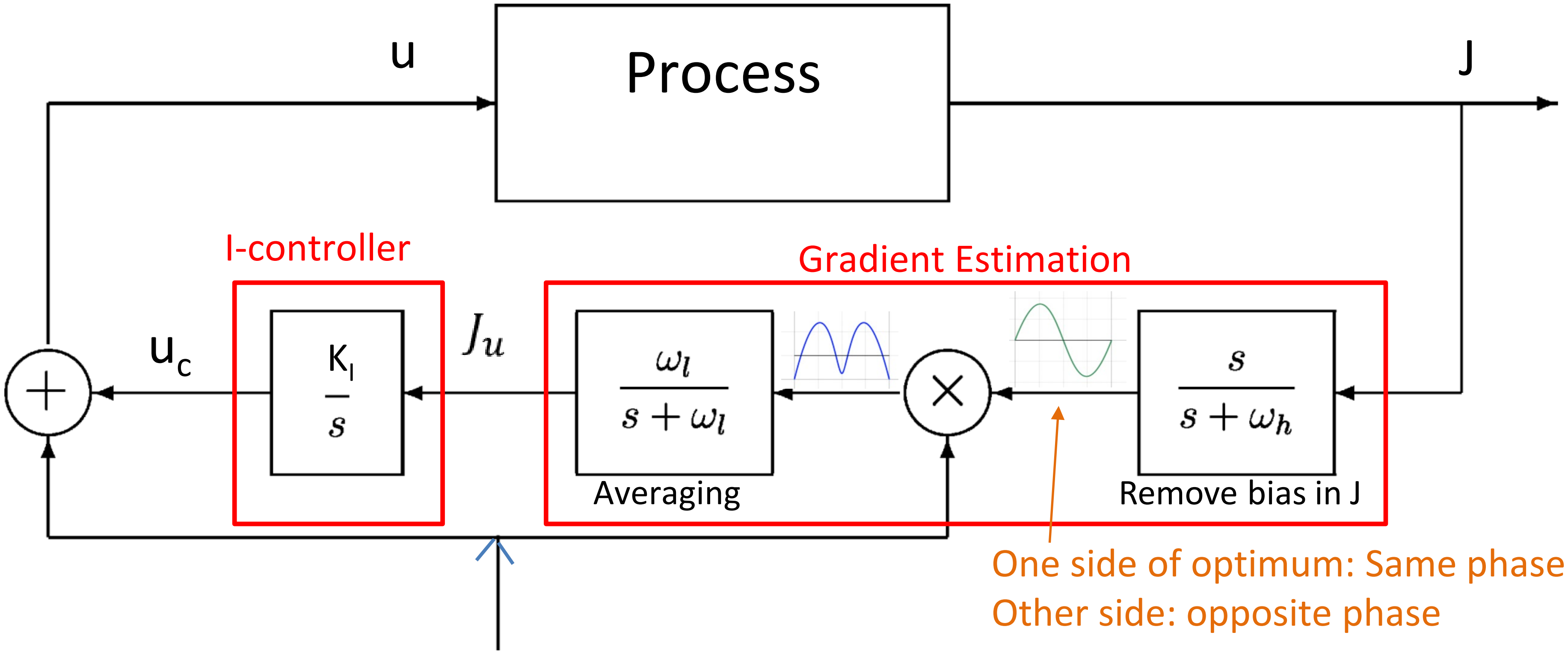
J_u=0 at top of hill

Equivalent: Minimize cost J (go to bottom of valley)

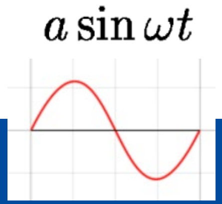


- Optimal setpoint: $J_u=0$
- If Hessian J_{uu} is constant:
  - J_u as a function of u is a straight line with slope J_{uu}
- Nice properties for feedback control of J_u
- No dynamics: Pure I-controller optimal
  - SIMC-rule: $K_I = 1/(J_{uu}\ \tau_c)$

Classical Extremum seeking control using sinusoids



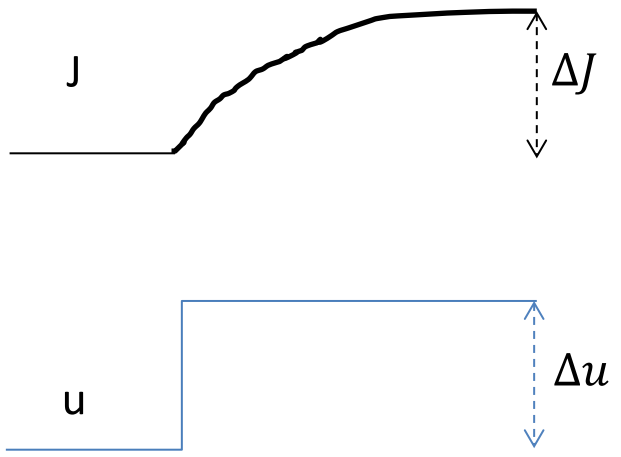
One side of optimum: Same phase
Other side: opposite phase



Multiplication trick: Draper & Li (1951)
Theory: Krstic & Wang (Automatica, 2000)

- **Simple to implement (don’t need computer),** but
- Prohibitively slow convergence for systems with slow dynamics
- Typically 100 times slower than the system dynamics !

More common today: Estimate Steady-state gradient using discrete perturbations (steps)



$$J_u = \frac{\Delta J}{\Delta u}$$

Usually only one input. Simplest: step change in u:

- Hill climbing control (Shinskey, 1967)
- Evolutionary operation (EVOP) (1960’s)
- NCO tracking (Francois & Bonvin, 2007)
- “Peturb and observe” = Maximum power point tracking (MPPT) (2010’s).

More advanced variants which may also be applied to multivariable systems

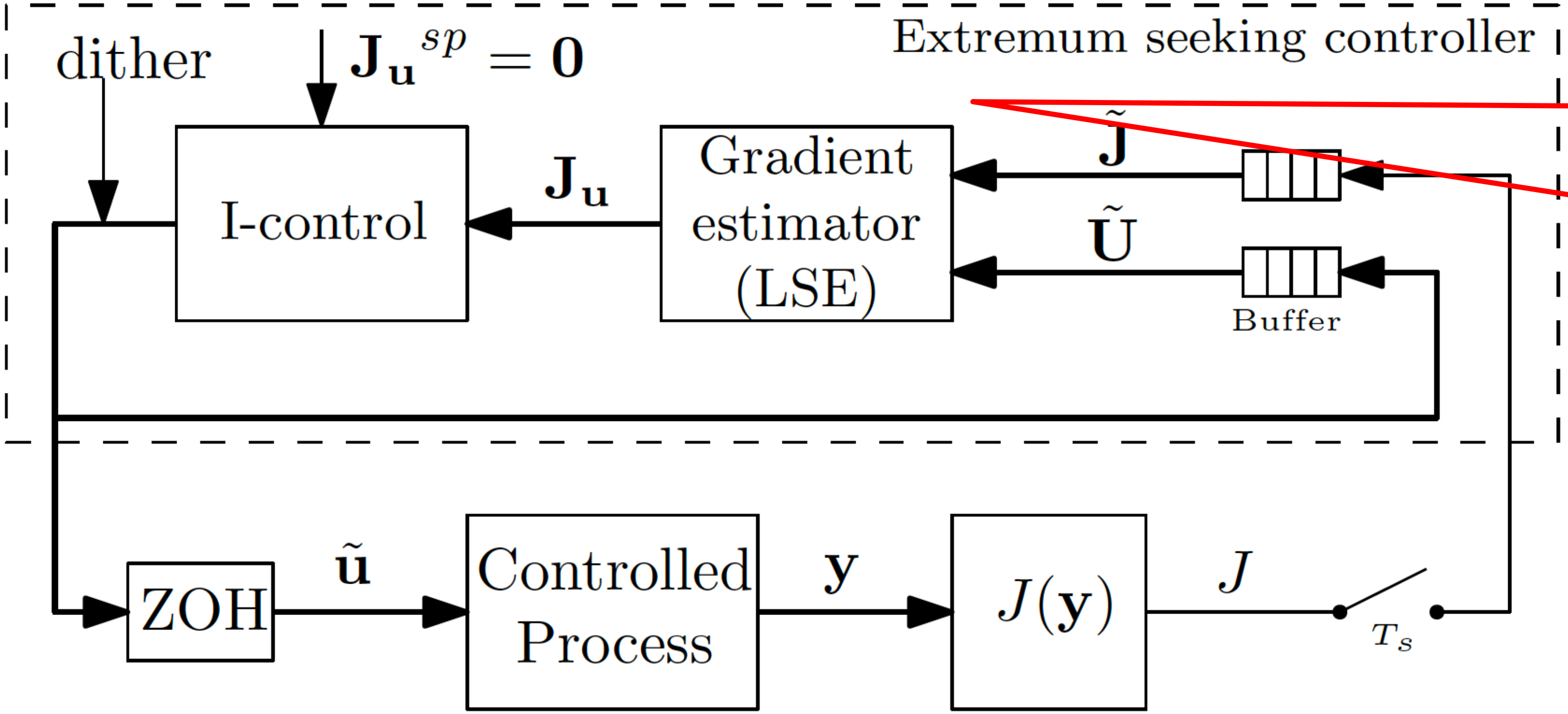
- **Least squares estimation**
- Fast Fourier transform (Dinesh Krishnamoorthy)

To avoid waiting for steady state

- Fitting of data to ARX model (difficult to make robust)

Note: Assumes steady state -> samling (step) time > 3-10 time process time constant

Least square Extremum seeking control



LSE: Fit a linear model

$$J = \mathbf{J}_{\tilde{\mathbf{u}}}{}^{\mathsf{T}} \tilde{\mathbf{u}} + m$$

Using least squares fit

$$\mathbf{Y} = [J_k, J_{k-1}, \ldots, J_{k-N+1}]^T$$

$$\mathbf{U} = [\mathbf{u}_k, \ldots, \mathbf{u}_{k-N+1}]^T$$

$$\theta = [\hat{\mathbf{J}}_{\mathbf{u}}^T, m]^T$$

$$\hat{\theta} = \arg\min_{\theta} \|\mathbf{Y} - \Phi^T \theta\|_2^2$$

to which the analytical solution is given by

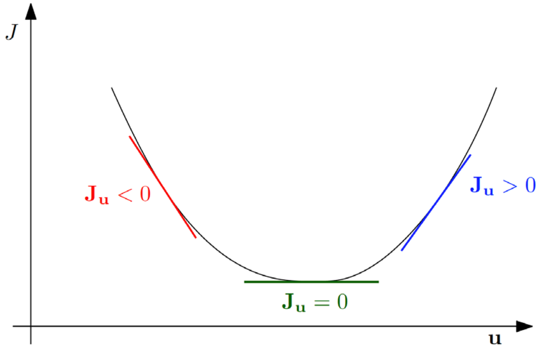
$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{Y}$$

213

Note: Assumes no dynamics -> samling time > 3-10 time process constant

Hunnekens et al. (2011, 2014)

Summary extremum seeking control



Idea: Estimate the cost gradient J_u from data and drive it to zero

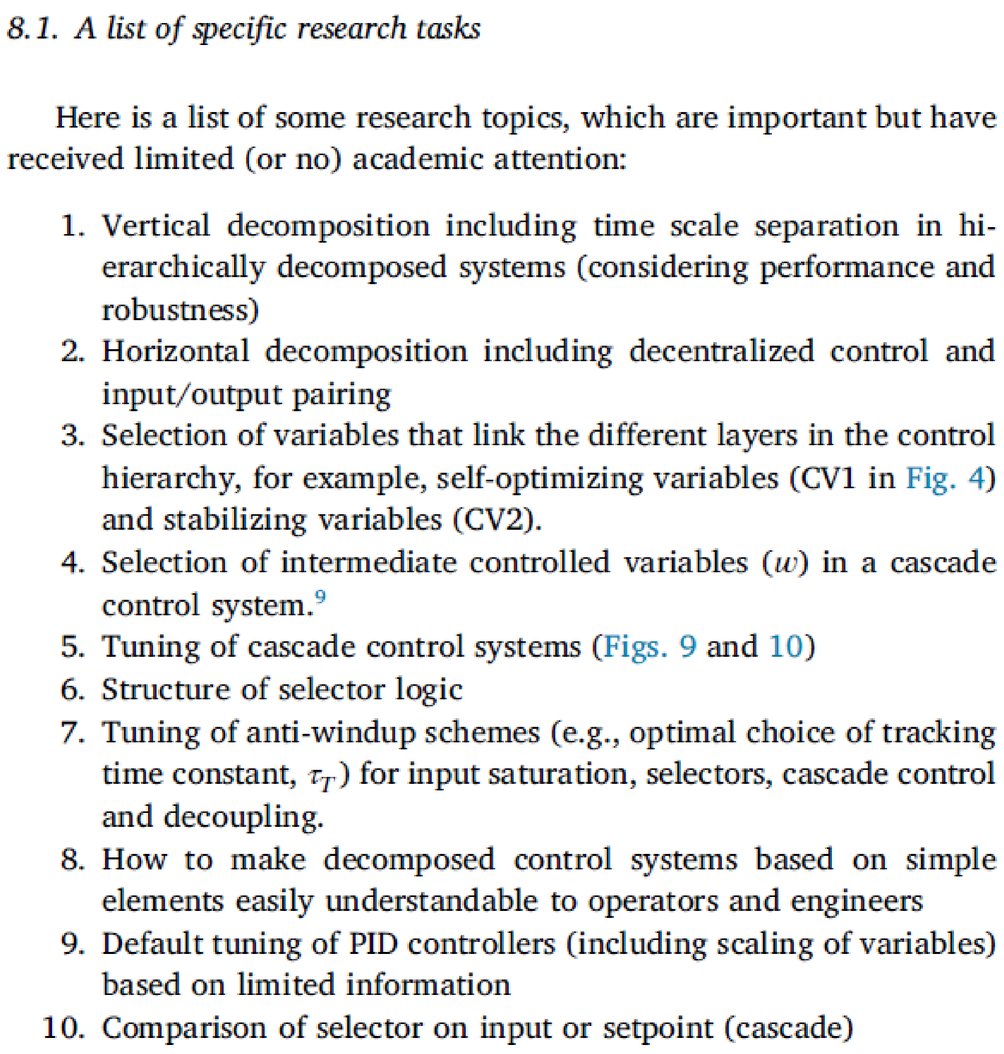
- Common to all methods:
  - Need measurement of cost J
  - Must wait for steady state (except ARX method which fails frequently)
  - Must assume no «fast» disturbances (while optimizing)

Algorithm needs two layers on top of process:

1. Optimization layer (slowest): Drive J_u to zero (may use I-controller)
2. Lower estimation layer: Estimate the local gradient J_u using data
   - Must wait for the process to reach steady state

- Need time scale separation between layers.
  - At best this means that the optimization needs to be 10 times slower than the process.
  - Often it needs to be 100 times slower.
- **Useful for fast processes with settling time a few seconds**
- Not useful for many chemical processes where time constant typically are several minutes
  - 10 minutes * 100 = 1000 minutes = 16 hours
  - Unllikely with 16 hours without disturbance

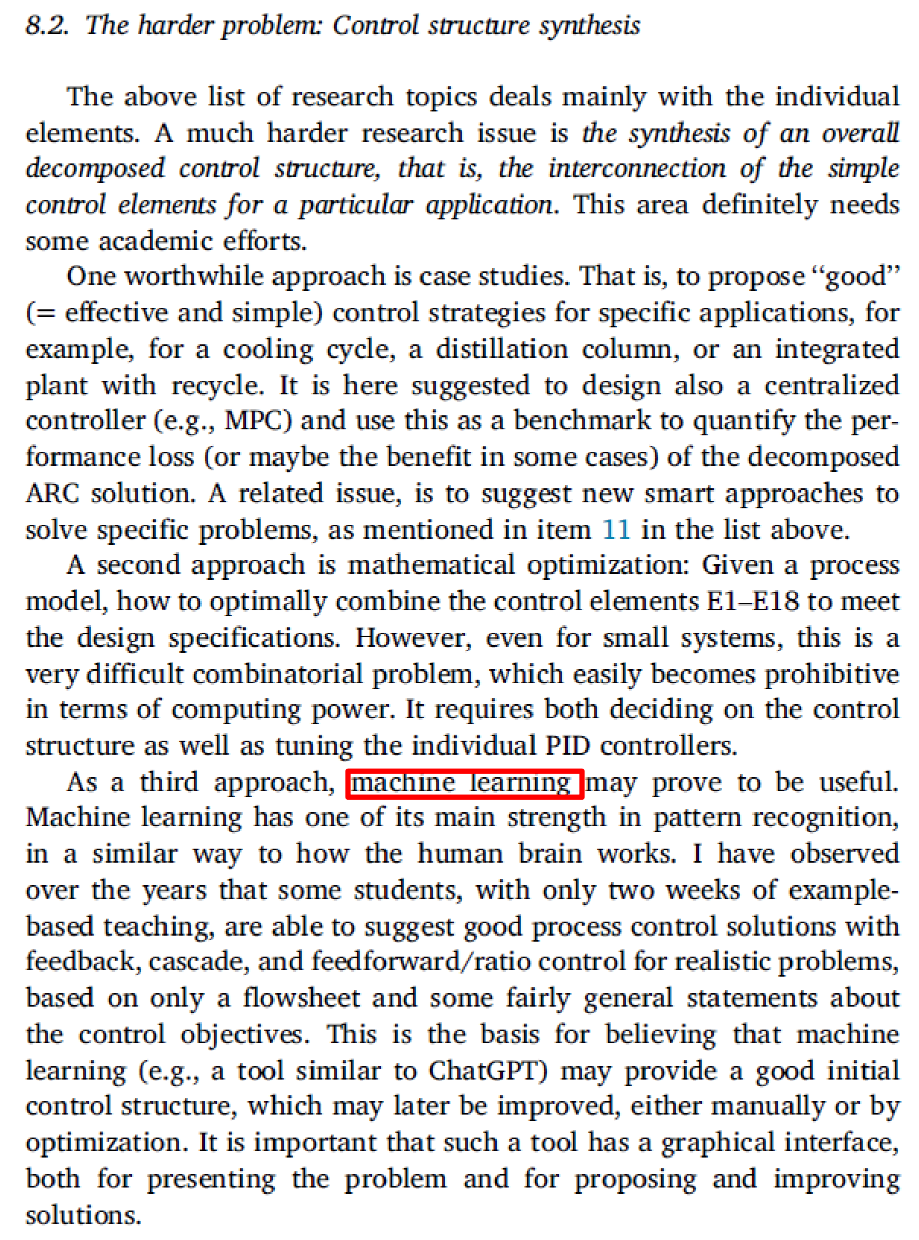
ARC: Research tasks



8.1. A list of specific research tasks

Here is a list of some research topics, which are important but have received limited (or no) academic attention:

1. Vertical decomposition including time scale separation in hierarchically decomposed systems (considering performance and robustness)
2. Horizontal decomposition including decentralized control and input/output pairing
3. Selection of variables that link the different layers in the control hierarchy, for example, self-optimizing variables (CV1 in Fig. 4) and stabilizing variables (CV2).
4. Selection of intermediate controlled variables (w) in a cascade control system.[9]
5. Tuning of cascade control systems (Figs. 9 and 10)
6. Structure of selector logic
7. Tuning of anti-windup schemes (e.g., optimal choice of tracking time constant, τ_T) for input saturation, selectors, cascade control and decoupling.
8. How to make decomposed control systems based on simple elements easily understandable to operators and engineers
9. Default tuning of PID controllers (including scaling of variables) based on limited information
10. Comparison of selector on input or setpoint (cascade)



8.2. The harder problem: Control structure synthesis

The above list of research topics deals mainly with the individual elements. A much harder research issue is *the synthesis of an overall decomposed control structure, that is, the interconnection of the simple control elements for a particular application*. This area definitely needs some academic efforts.

One worthwhile approach is case studies. That is, to propose “good” (= effective and simple) control strategies for specific applications, for example, for a cooling cycle, a distillation column, or an integrated plant with recycle. It is here suggested to design also a centralized controller (e.g., MPC) and use this as a benchmark to quantify the performance loss (or maybe the benefit in some cases) of the decomposed ARC solution. A related issue, is to suggest new smart approaches to solve specific problems, as mentioned in item 11 in the list above.

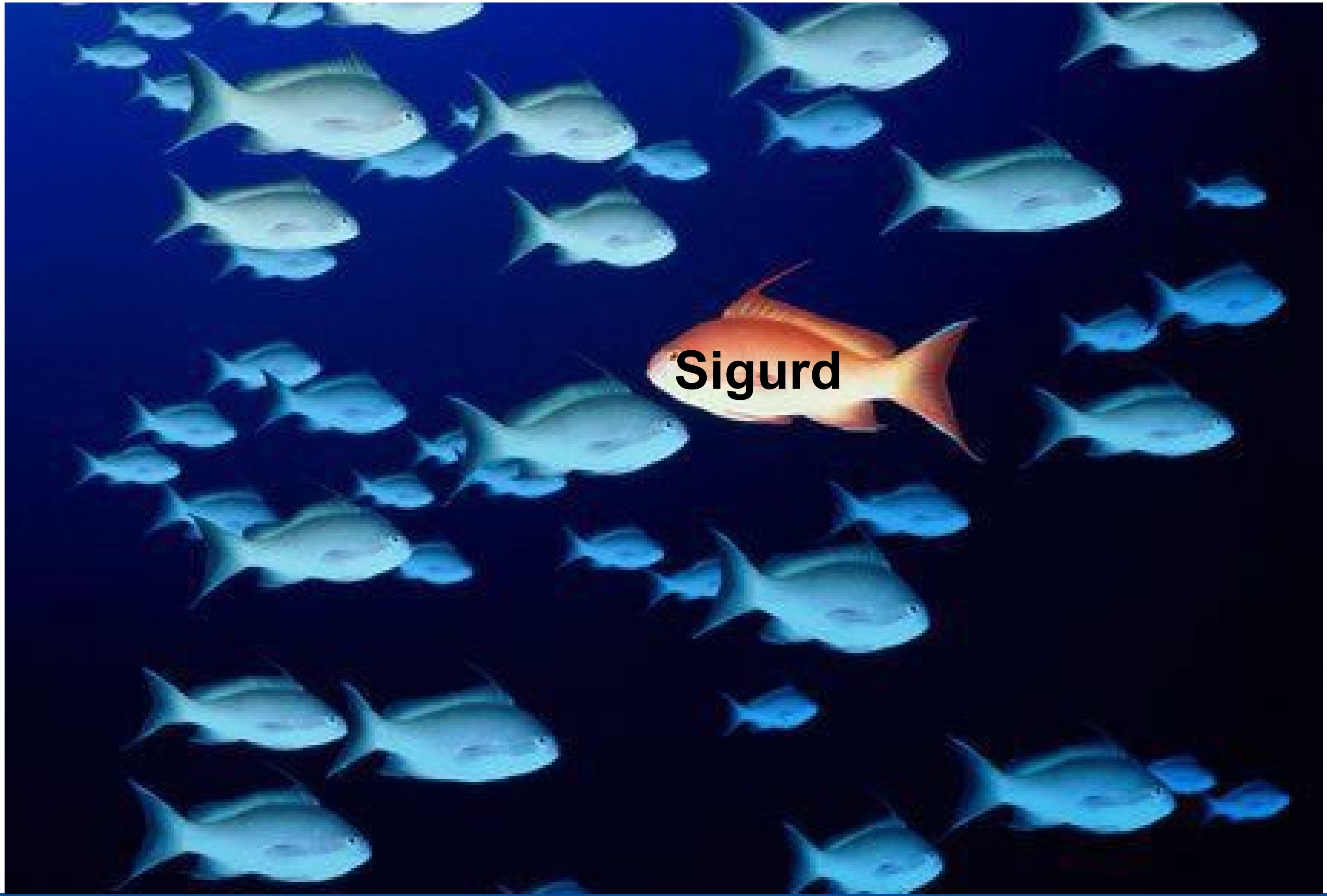
A second approach is mathematical optimization: Given a process model, how to optimally combine the control elements E1–E18 to meet the design specifications. However, even for small systems, this is a very difficult combinatorial problem, which easily becomes prohibitive in terms of computing power. It requires both deciding on the control structure as well as tuning the individual PID controllers.

As a third approach, machine learning may prove to be useful. Machine learning has one of its main strength in pattern recognition, in a similar way to how the human brain works. I have observed over the years that some students, with only two weeks of example-based teaching, are able to suggest good process control solutions with feedback, cascade, and feedforward/ratio control for realistic problems, based on only a flowsheet and some fairly general statements about the control objectives. This is the basis for believing that machine learning (e.g., a tool similar to ChatGPT) may provide a good initial control structure, which may later be improved, either manually or by optimization. It is important that such a tool has a graphical interface, both for presenting the problem and for proposing and improving solutions.

Present Academic control community fish pond

Complex optimal centralized
Solution (**EMPC, FL**)

Simple solutions
that work (ARC, PID)



FL = feedback linearization

Future Academic control community fish pond

Complex optimal centralized Solution (**EMPC, FL**)

Simple solutions that work (SRC,PID)