PID is the future of Advanced Control

Sigurd Skogestad

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

IFAC PID’24- Conference
Almeria, Spain
12 June 2024

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

AMERICAN CONTROL CONFERENCE
San Diego, California
June 6-8, 1984

IMPLICATIONS OF INTERNAL MODEL CONTROL FOR PID CONTROLLERS

Manfred Morari
Sigurd Skogestad

California Institute of Technology
Department of Chemical Engineering
Pasadena, California 91125

Daniel F. Rivera

University of Wisconsin
Department of Chemical Engineering
Madison, Wisconsin 53706

252 *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 252–265

Internal Model Control. 4. PID Controller Design

Daniel E. Rivera, Manfred Morari,* and Sigurd Skogestad

Chemical Engineering, 206-41, California Institute of Technology, Pasadena, California 91125

For a large number of single input–single output (SISO) models typically used in the process industries, the Internal Model Control (IMC) design procedure is shown to lead to PID controllers, occasionally augmented with a first-order lag. These PID controllers have as their only tuning parameter the closed-loop time constant or, equivalently, the closed-loop bandwidth. On-line adjustments are therefore much simpler than for general PID controllers. As a special case, PI- and PID-tuning rules for systems modeled by a first-order lag with dead time are derived analytically. The superiority of these rules in terms of both closed-loop performance and robustness is demonstrated.

SIMC* PID tuning rule (2001,2003)

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

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

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

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

$$\tau_D = \tau_2$$

Tuning paramete

$\tau_c \geq \theta$

λ

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*

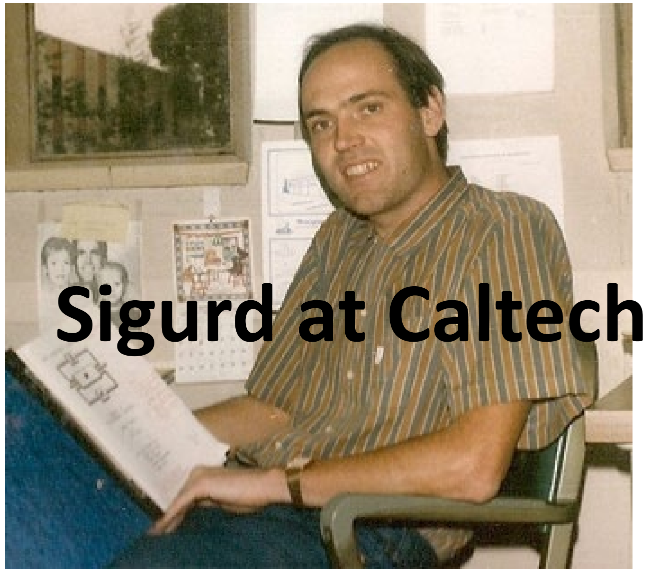
Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

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



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.

Control system structure*

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

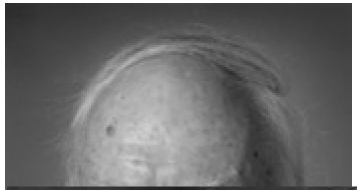
The central issue to be resolved ... is the determination of control system structure.*

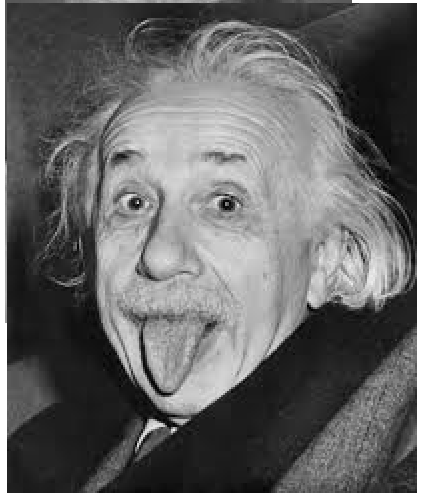
Which variables should be measured, which inputs should be manipulated and which links should be made between the two sets?

*Current terminology: Control system architecture

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?__

There is more than a suspicion that the work of a genius is needed here, for without it the control configuration problem will likely remain in a primitive, hazily stated and wholly unmanageable form.

The gap [between theory and practice] is present indeed, but contrary to the views of many, it is the theoretician who must close it.

*Current terminology: Control system architecture

Well, I'm not a genius, but I didn't give up.
I started on this in 1983. 40 years later:

Chemical Engineering **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]

Annual Reviews in Control 56 (2023) 100903

Contents lists available at ScienceDirect

Annual Reviews in Control

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

ELSEVIER

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

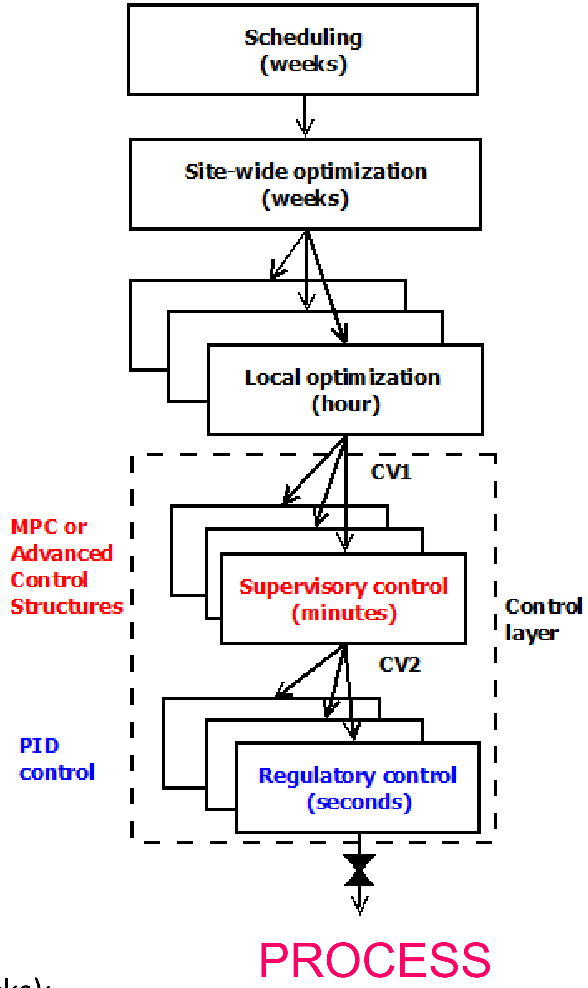
The paper explores the standard advanced control elements commonly used in industry for designing advanced control systems. These elements include cascade, ratio, feedforward, decoupling, selectors, split range, and more, collectively referred to as "advanced regulatory control" (ARC). Numerous examples 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

NTNU

Two fundamental ways of decomposing the controller

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



- 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

Time scale separation: Control* layers

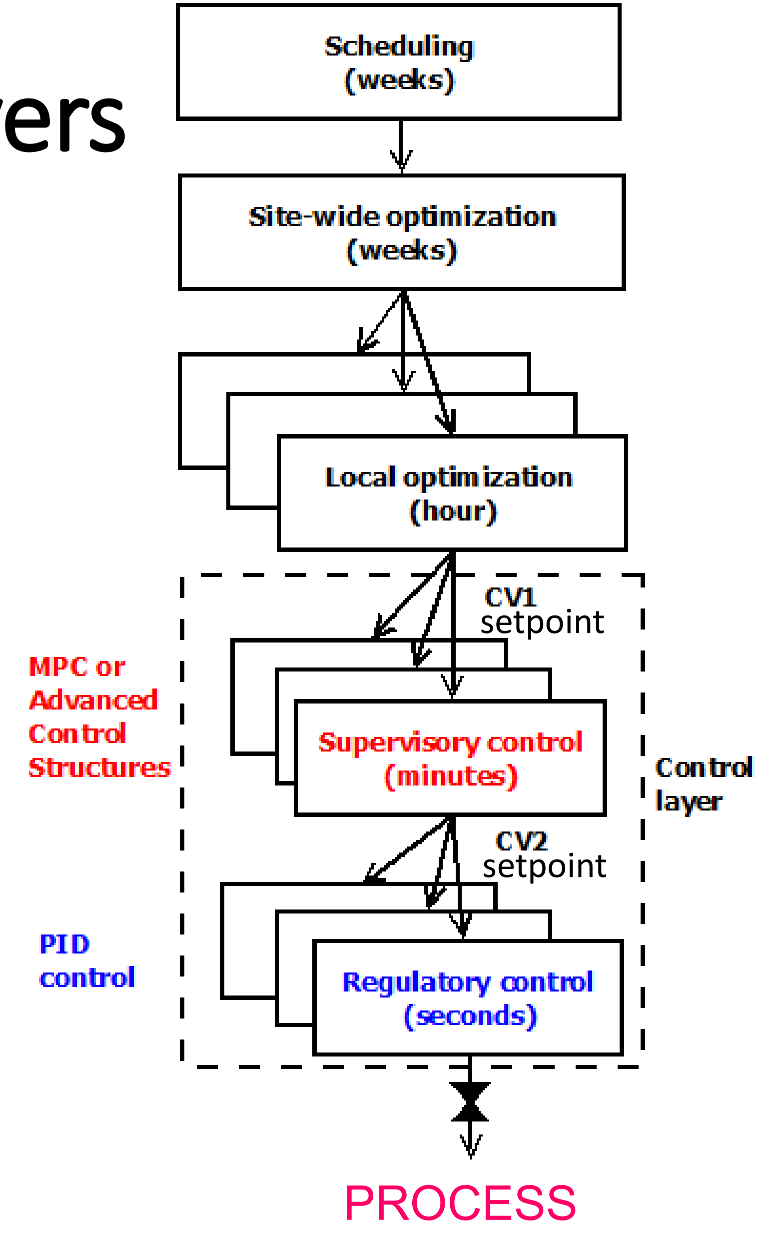
Two objectives for control: Stabilization and economics

- **Supervisory ("advanced") control layer**

  Tasks:
  - Follow set points for CV1 from economic optimization layer
  - Switch between active constraints (change CV1)
  - Look after regulatory layer (avoid that MVs saturate, etc.)

- **Regulatory control (PID layer):**
  - Stable operation (CV2)

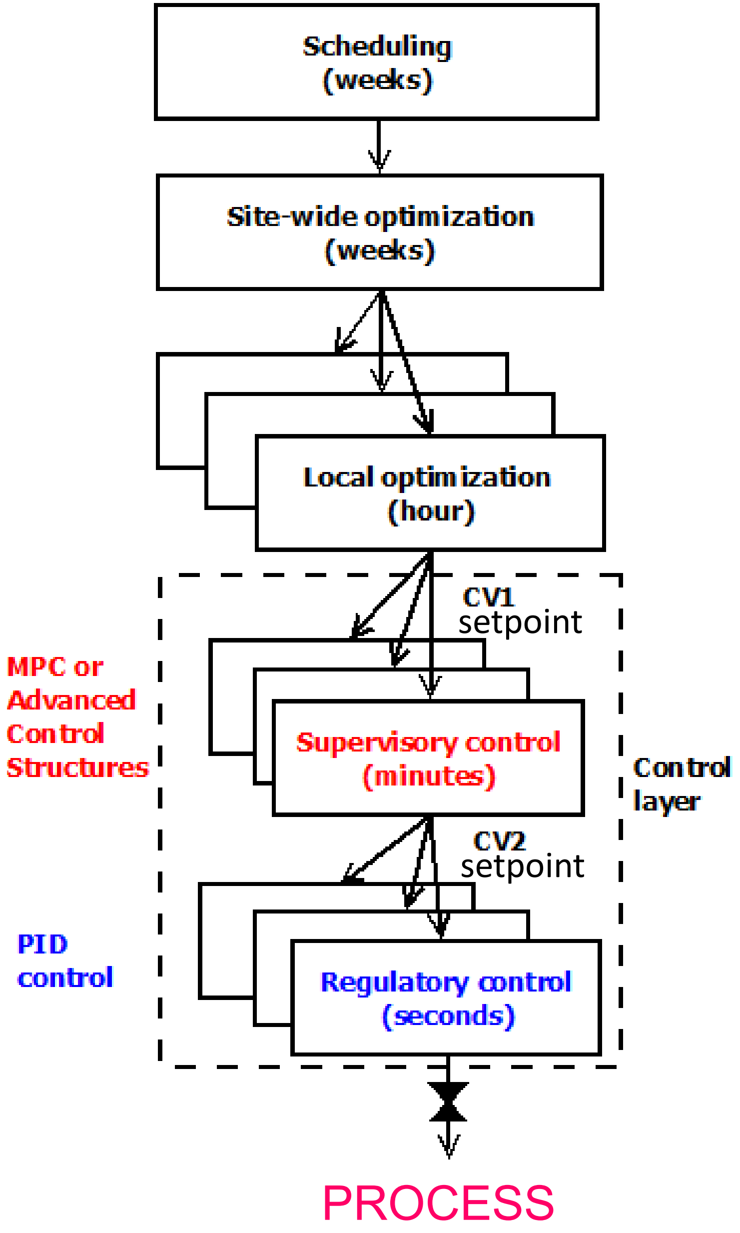
*My definition of «control» is that the objective is to track setpoints



CV = controlled variable

«Advanced» control

- Advanced: This is a relative term
- Usually used for anything than comes in addition to (or in top of) basic PID loops
- Mainly used in the «supervisory» control layer
- Two main options
  - **Standard «Advanced regulatory control» (ARC) elements**
    - Based on decomposing the control system
      - Cascade, feedforward, selectors, etc.
    - This option is preferred if it gives acceptable performance
  - **Model predictive control (MPC)**
    - Requires a lot more effort to implement and maintain
    - Use for interactive processes
    - Use with known information about future (use predictive capanulities)



Combine control and optimization into one layer?

EMPC: Economic model predictive "control"'

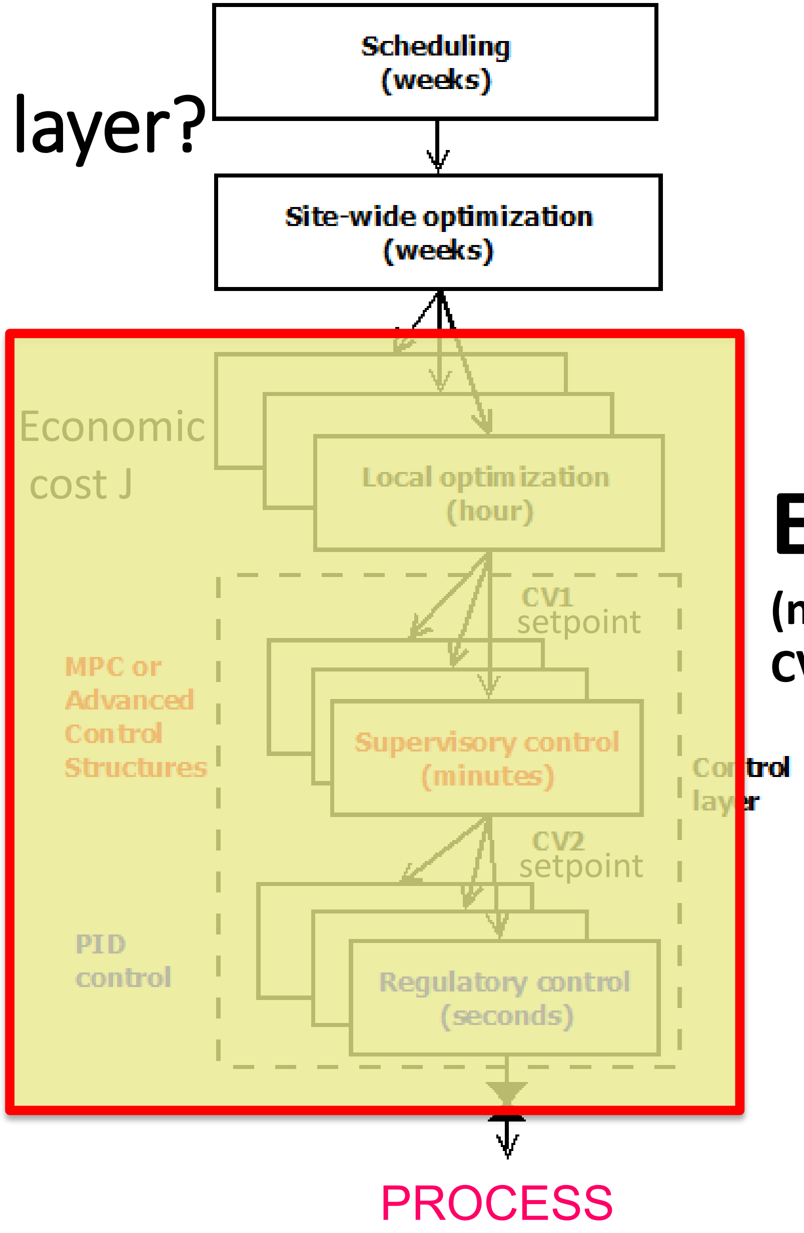
$J_{EMPC} = J + J_{control}$

Penalize input usage, $J_{control} = \Sigma \Delta u_i^2$

NO, combining layers is generally not a good idea!
(the good idea is to separate them!)

One layer (EMPC) is optimal theoreretically, but

- Need detailed dynamic model of everything
- Tuning difficult and indirect
- Slow! (or at least difficult to speed up parts of the control)
- Robustness poor
- Implementation and maintainance costly and time consuming



EMPC

(no setpoints, CV1, CV2)

Typical economic cost function:

J [$/s] = cost feed + cost energy – value products

CV = controlled variable

RTO = real-time optimization

What about «conventional» RTO and MPC?

- Yes, it's OK
- Both has been around for more than 50 years (since 1970s)
  - but the expected growth never came
- MPC is still used mostly in large-scale plants (petrochemical and refineries).
- MPC is far from replacing PID as some expected in the 1990s.

- But plants need to be run optimally:

  ⇒ Need something else than conventional RTO/MPC!

MPC = model predictive control
RTO = real-time optimization

Alternative solutions for advanced control

- Would like: Feedback solutions that can be implemented with minimum need for models

- **Machine learning?**
  - Requires a lot of data, not realistic for process control
  - And: Can only be implemented after the process has been in operation

- **"Classical advanced regukatory control" (ARC) based on single-loop PIDs?**
  - YES!
  - Extensively used by industry
  - Problem for engineers: Lack of design methods
    - Has been around since 1930's
    - But almost completely neglected by academic researchers
  - Main fundamental limitation: Based on single-loop (need to choose pairing)

ARC = Advanced regulatory control

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).

Common ARC elements

Journal of Process Control 122 (2023) 113–133


Journal of Process Control

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

ELSEVIER

Review

Transformed inputs for linearization, decoupling and feedforward control

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

E1. Cascade control

- Have Extra output (state) measurements

E2. Ratio control

- "Feedforward" for mixing process

E12. Decoupling elements

- Have interactive process

E13. Linearization elements / Adaptive gain

- Have Nonlinear process

Often static nonlinear «function block»

One unifying approach is «Transformed inputs» (similar to feedback linearization)

E5-E7. Split-range control (or multiple controllers or VPC)

- Need extra inputs (MV) to handle all conditions (steady state) (MV-MV switching)

E3. Valve position control (VPC) (Input resetting/Midranging control)

- Have extra inputs dynamically

E4. Selectors

- Have changes in active constraints (CV-CV switching)

ARC = Advanced regulatory control
VPC = Valve position control

CV = Controlled variable
MV = manipulated variable

How design classical APC 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

ELSEVIER

Review article

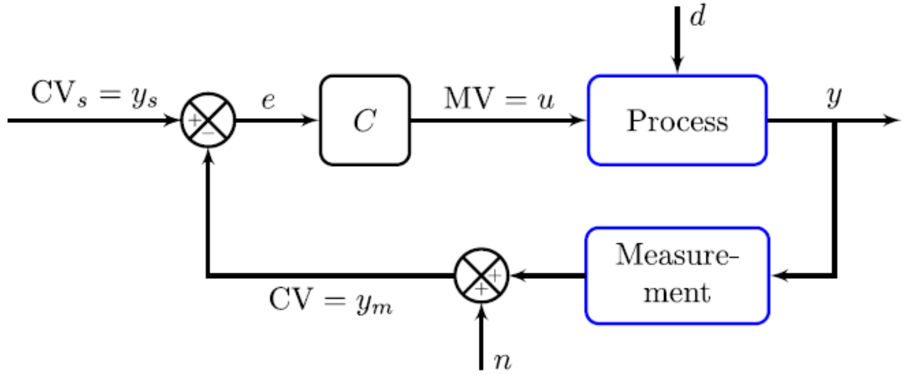
Advanced control using decomposition and simple elements

Sigurd Skogestad

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

NTNU

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)*

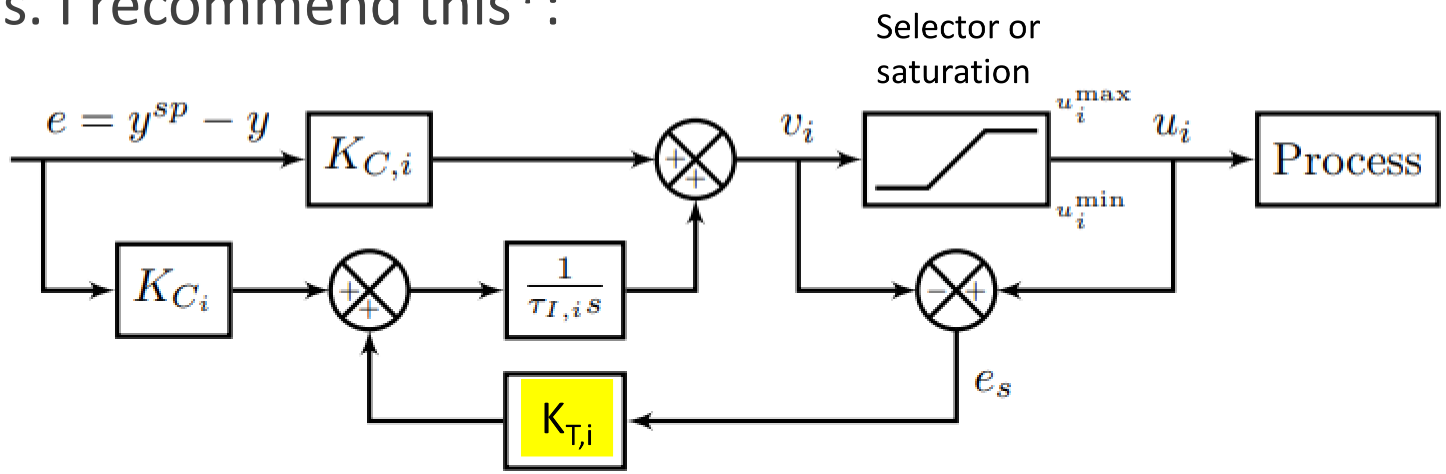
Additional rule for interactive systems:

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

NTNU

E8. Anti-windup

- All controllers with I action need anti-windup to «stop integration» during periods when the controller output (v_i) is not affecting the process:
  - Controller is disconnected (e.g., because of selector)
  - Physical MV u_i is saturated
- Many approaches. I recommend this*:



Anti-windup using back-calculation*. Typical choice for tracking constant, K_T=1

*Åström, K. J., & Hägglund, T. (1988). *Automatic tuning of PID controllers*. ISA.

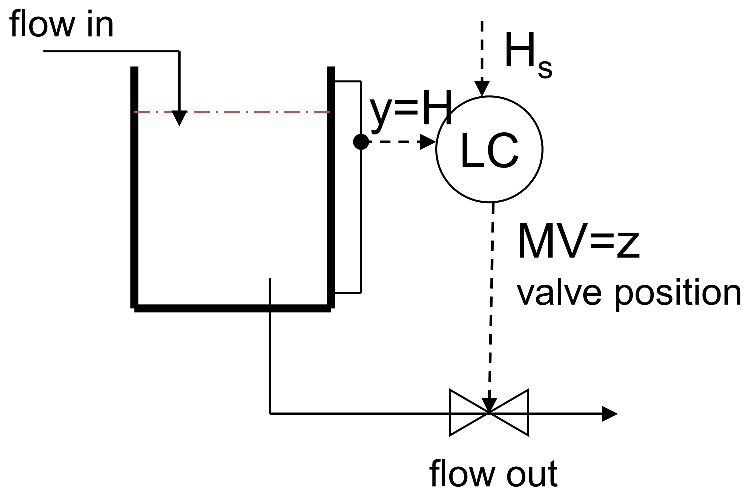
E1. Cascade control

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

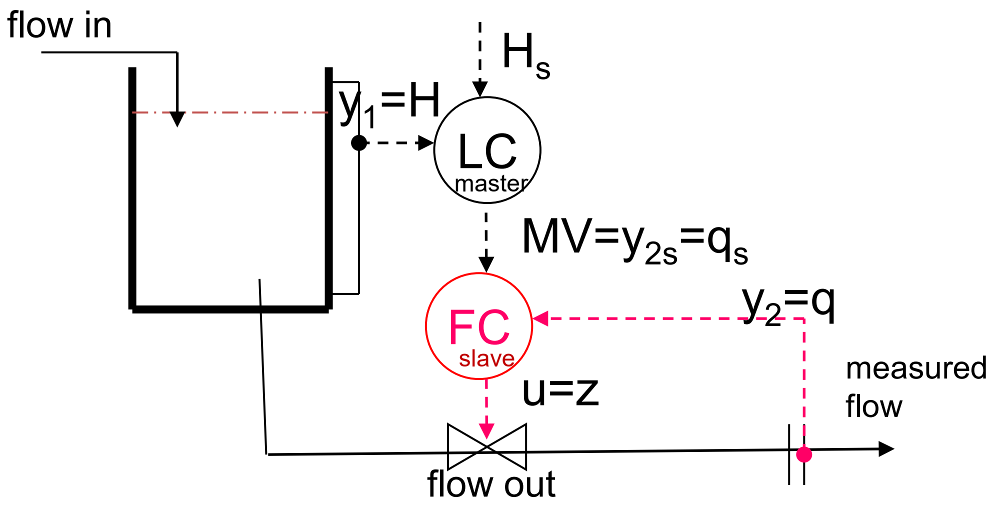
Idea: make use of extra "local" output measurement (y_2)
Implementation: Controller ("master") gives setpoint to another controller ("slave")

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

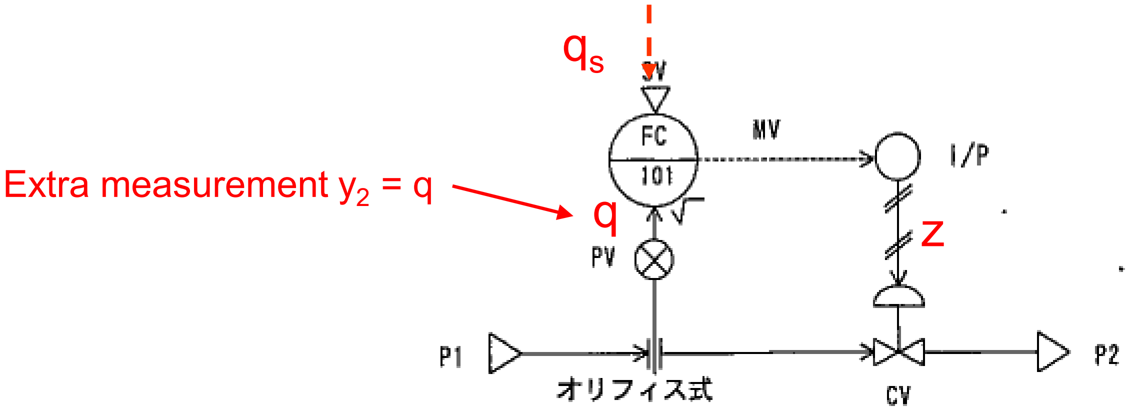
WITHOUT CASCADE



WITH CASCADE



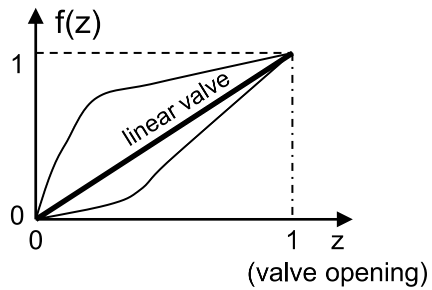
What are the benefits of adding a flow controller (inner cascade)?



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

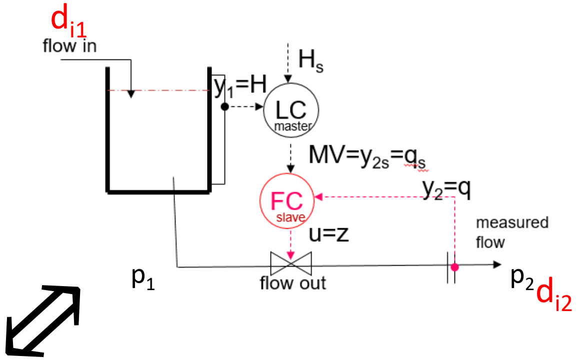
1. Counteracts nonlinearity in valve, f(z)
   - High gain in inner loop eliminates nonlinearity inside inner loop
   - With fast flow control we can assume $q = q_s$

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



NTNU

Block diagram flow controller (inner cascade)



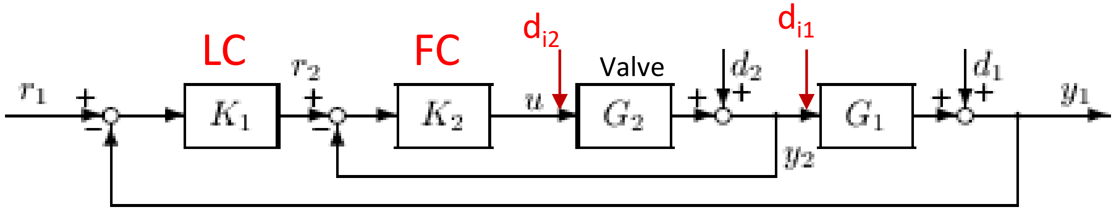
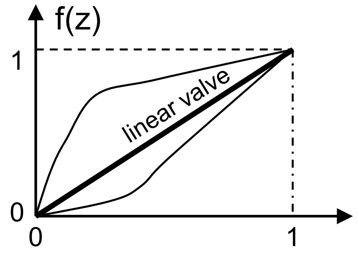


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

Example: Level control with slave flow controller:

u = z (valve position, flow out)
y_1 = H
y_2 = q
d_{11} = flow in
d_{i2} = p_1-p_2

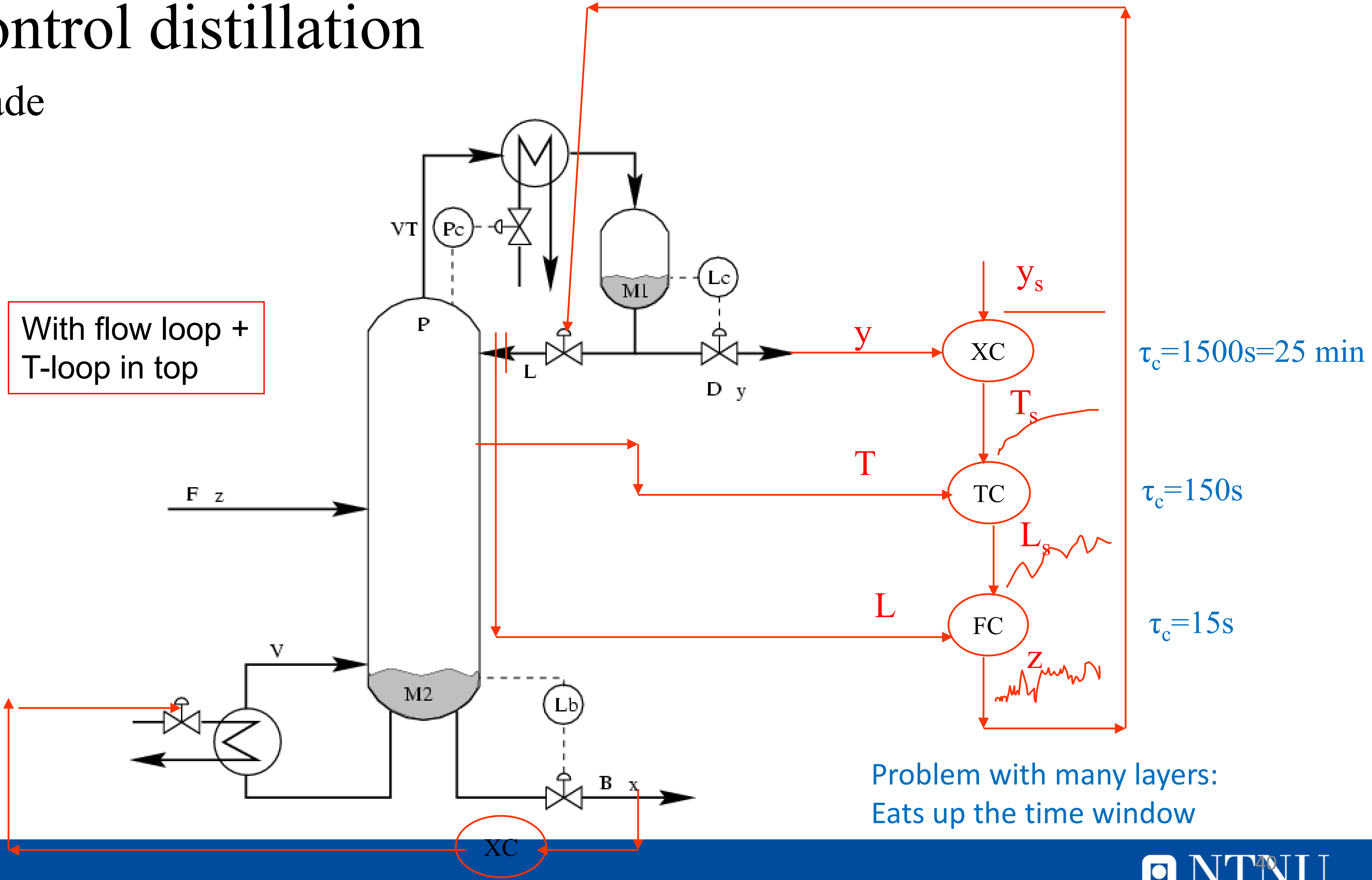
Transfer functions:
$G_2 = k(z)/(\tau s+1)$ where $k(z) = dq/dz$ (nonlinear!)
$G_1 = -1/(As)$
K_1 = Level controller (master)
K_2 = Flow controller (slave)



k(z) = slope df/dz

Cascade control distillation

3 layers of cascade



Problem with many layers:
Eats up the time window

Shinskey (1967)

The principal advantages of cascade control are these:

1. Disturbances arising within the secondary loop are corrected by the secondary controller before they can influence the primary variable.

2. Phase lag existing in the secondary part of the process is reduced measurably by the secondary loop. This improves the speed of response of the primary loop.

3. Gain variations in the secondary part of the process are overcome within its own loop.

4. The secondary loop permits an exact manipulation of the flow of mass or energy by the primary controller.

Special common case ("series cascade")

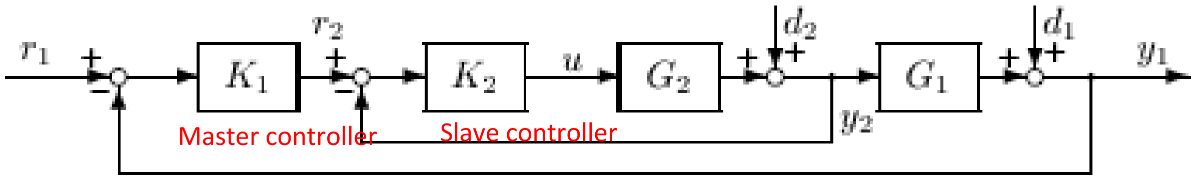
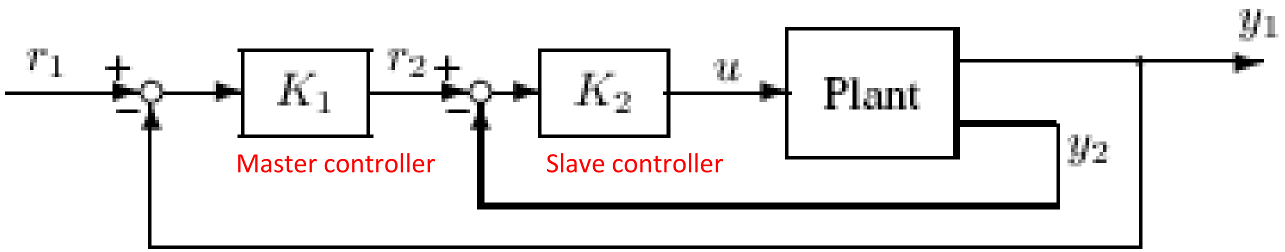


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

Always helpful

General case ("parallel cascade")



(a) Extra measurements y_2 (conventional cascade control)

Not always helpful

Tuning cascade control

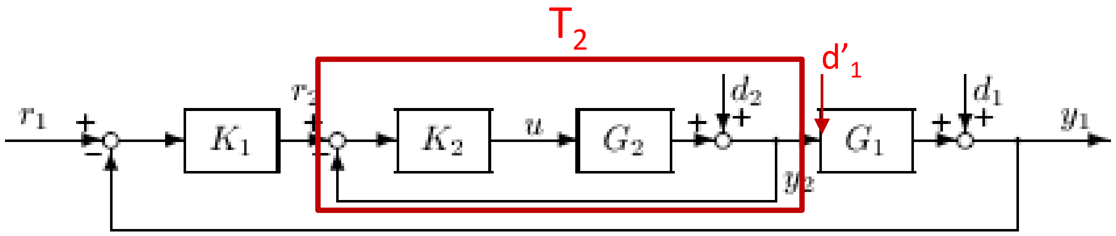


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

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.

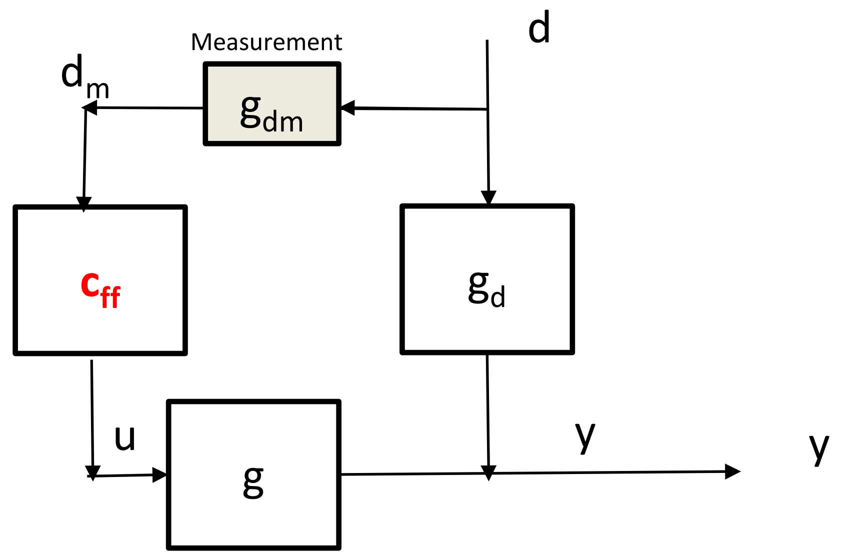
NTNU

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 (σ =10) is to avoid that control gets too slow, especially if we have many layers

E11. Feedforward (FF) control

Mainly: For disturbances where feedback control is not good enough.



- Model: $y = g\,u + g_d\,d$
- Measured disturbance: $d_m = g_{dm}\,d$
- Feedforward controller: $u = c_{FF}\,d_m$
- Get $y = (g\,c_{FF}\,g_{dm} + g_d)\,d$
- Ideal feedforward controller:

  $y = 0\,d$ -> $c_{FF,ideal} = -(g_d / (g_{dm}\,g))$

  - But often not realizable

  – Common simplification is to use static FF: $c_{FF} = k$

  – General. Approximate $c_{FF,ideal}$ by $c_{FF}(s) = k\frac{(T_1 s+1)\cdots}{(\tau_1 s+1)(\tau_2 s+1)\cdots}e^{-\theta s}$

  where must have at least as many τ's as T's

NTNU

Main problem feedforward: Sensitive to model error

- "If process gain increases by more than a factor 2, then ideal feedforward control is worse than no control"
- Why? Overcompensate in wrong direction
  - Proof: $y = gu + g_d\, d$ where $u = c_{FF}\, g_{dm}\, d$
  - Response with feedforward controller:

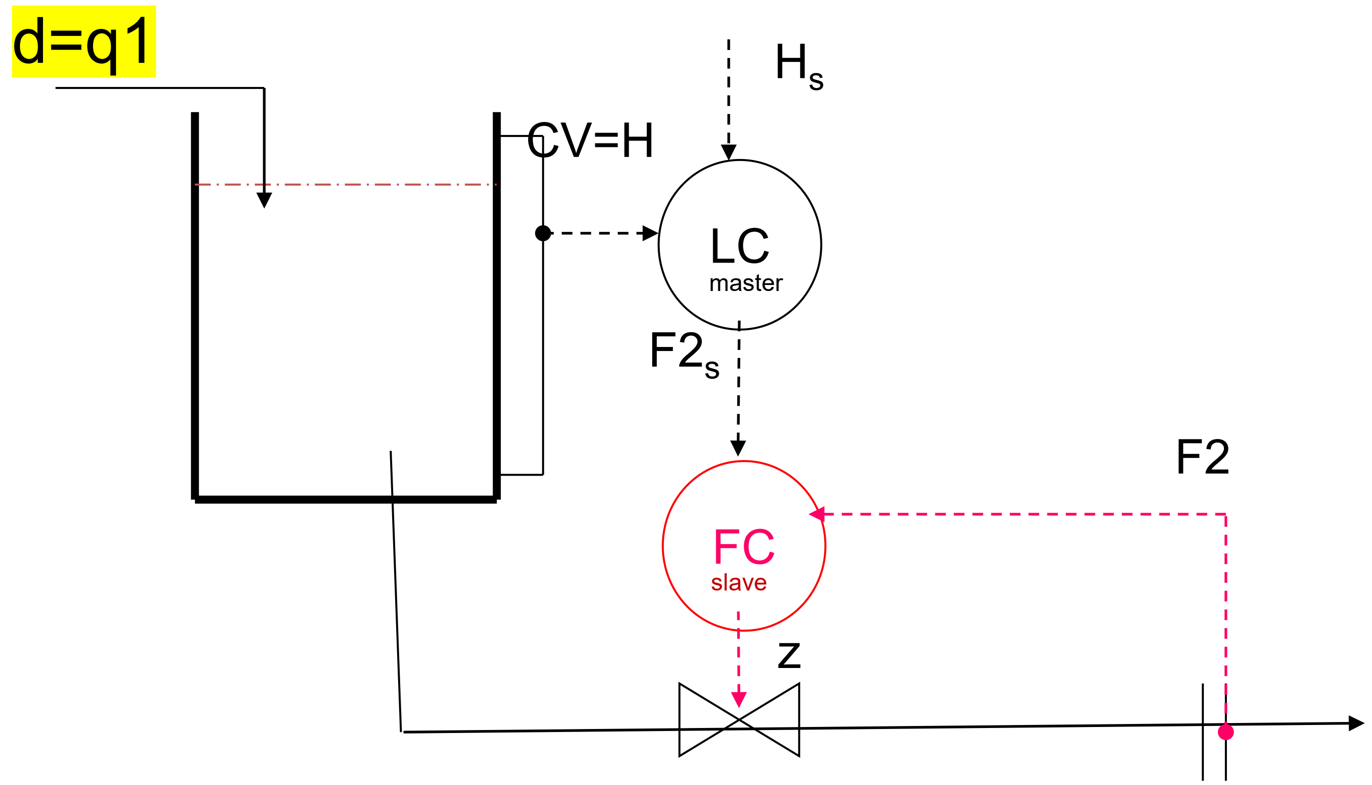
  $y = (g\, c_{FF}\, g_{dm} + g_d)\, d$
  - Ideal: Use $c_{FF,ideal} = -\, g_d/g\, g_{dm}$. Gives $y = (-g_d + g_d)\, d = 0\, d$
  - But note that g is $c_{FF,ideal}$ is a model
  - Real: If the real process gain (g) has increased by a factor x then

  $y = (-x g_d + g_d)\, d = (-x+1)\, g_d\, d$

  For $x>2$: $|-x+1|>1$ (worse than no control)….

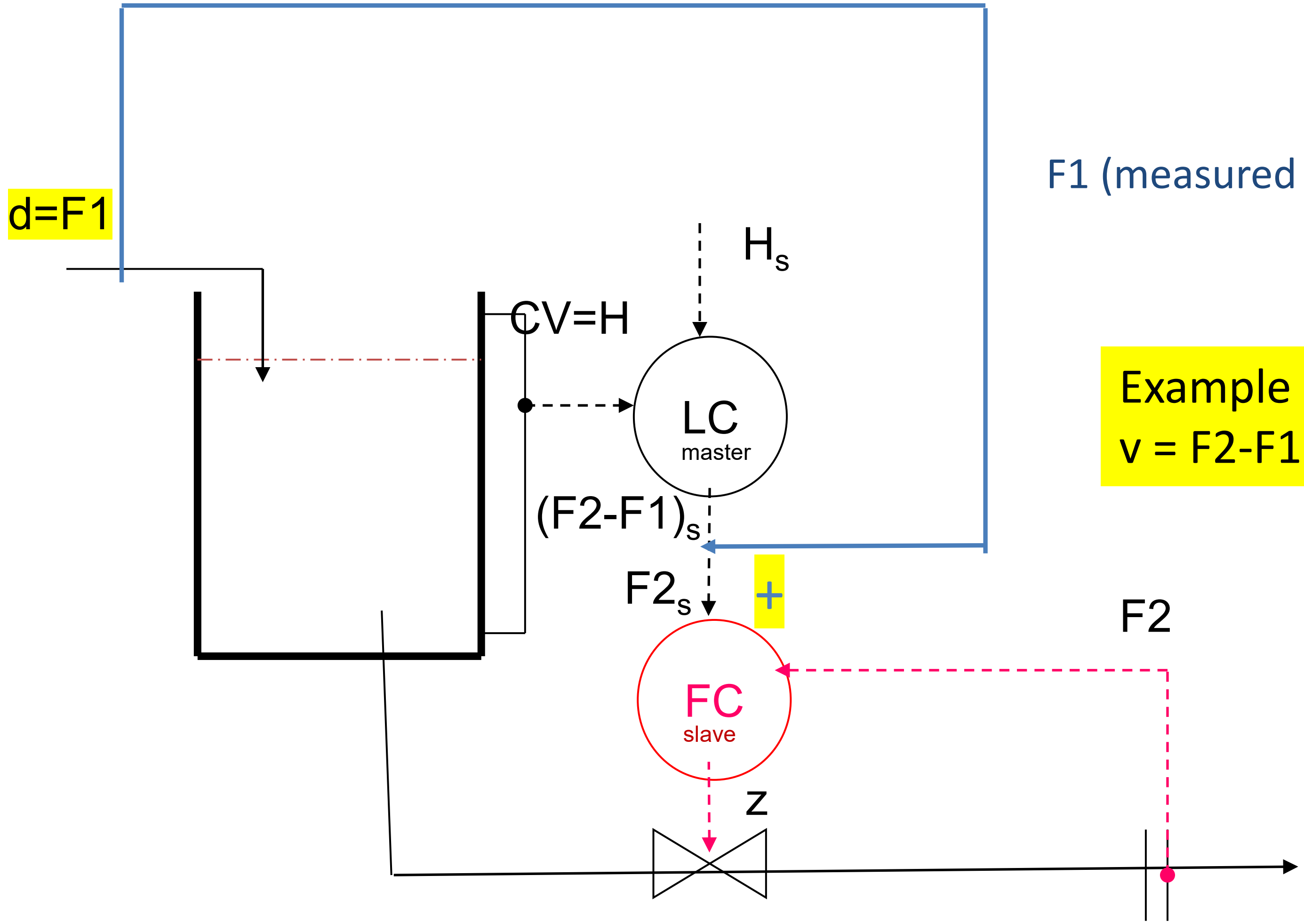
NTNU

Quiz: How can we add feedforward?



NTNU

Solution: How can we add feedforward?



F1 (measured flow disturbance)

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

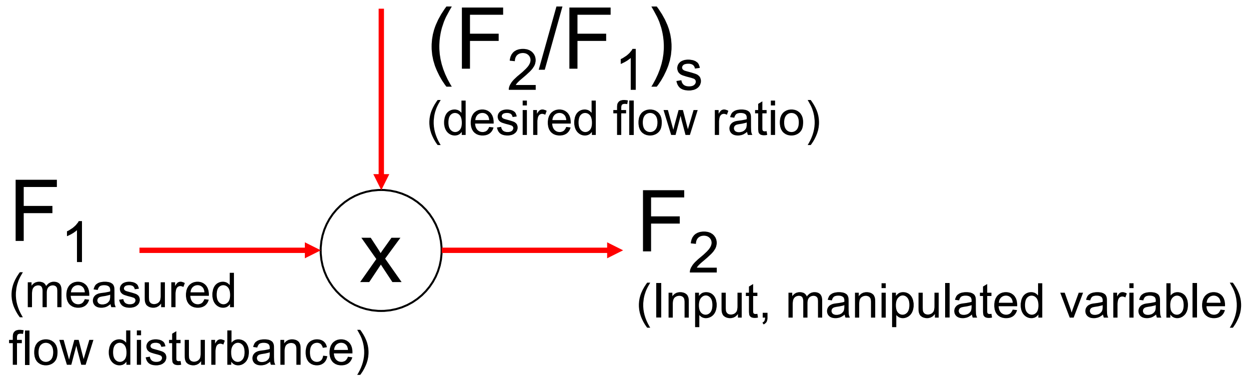
NTNU

E2. 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):



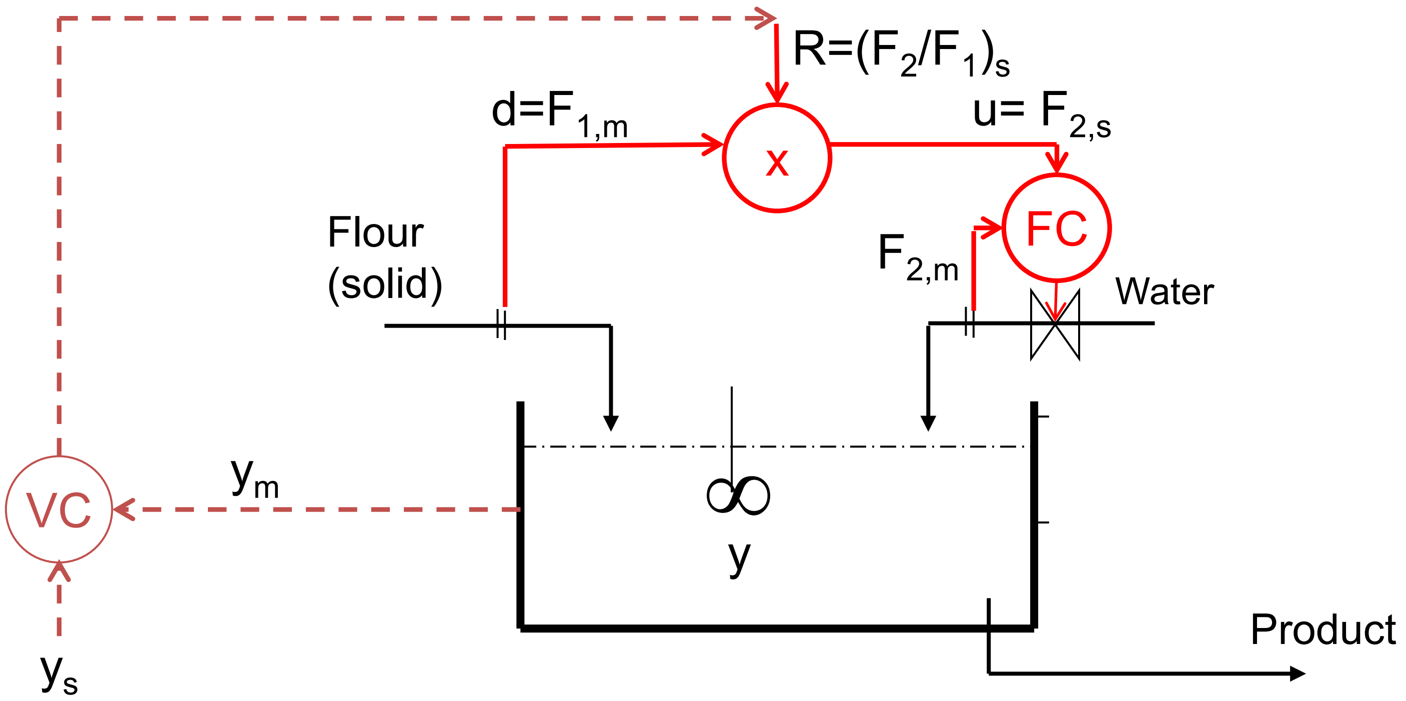
"Measure disturbance ($d=F_1$) and adjust input ($u=F_2$) such that ratio is at given value $(F_2/F_1)_s$"

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)

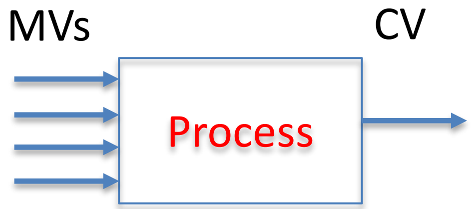


Feedback correction («trim»)

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

Valve position control (VPC)

Have extra MV (input): One CV, many MVs



Two different cases of VPC:

- E3. Have extra <u>dynamic</u> MV
  - Both MVs are used all the time
- E7. Have extra <u>static</u> MV
  - May use VPC for MV-MV switching: see later

MV = manipulated variable
CV = controlled variable

E3. VPC for extra dynamic **input**

u_2 = main input for steady-state control of CV

u_1 = extra dynamic input for fast control of y

3.4. Input (valve) position control (VPC) to improve the dynamic response (E3)

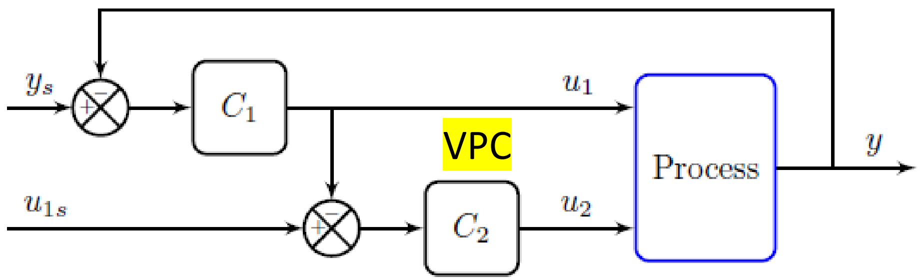


Figure 12: Valve (input) position control (VPC) for the case when an "extra" MV (u_1) is used to improve the dynamic response. A typical example is when u_1 is a small fast valve and u_2 is a large slower valve.
C_1 = fast controller for y using u_1.
C_2 = slow valve position controller for u_1 using u_2 (always operating).
u_{1s} = steady-state resting value for u_1 (typically in mid range. e.g. 50%).

Example 1: Large (u_2) and small valve (u_1)

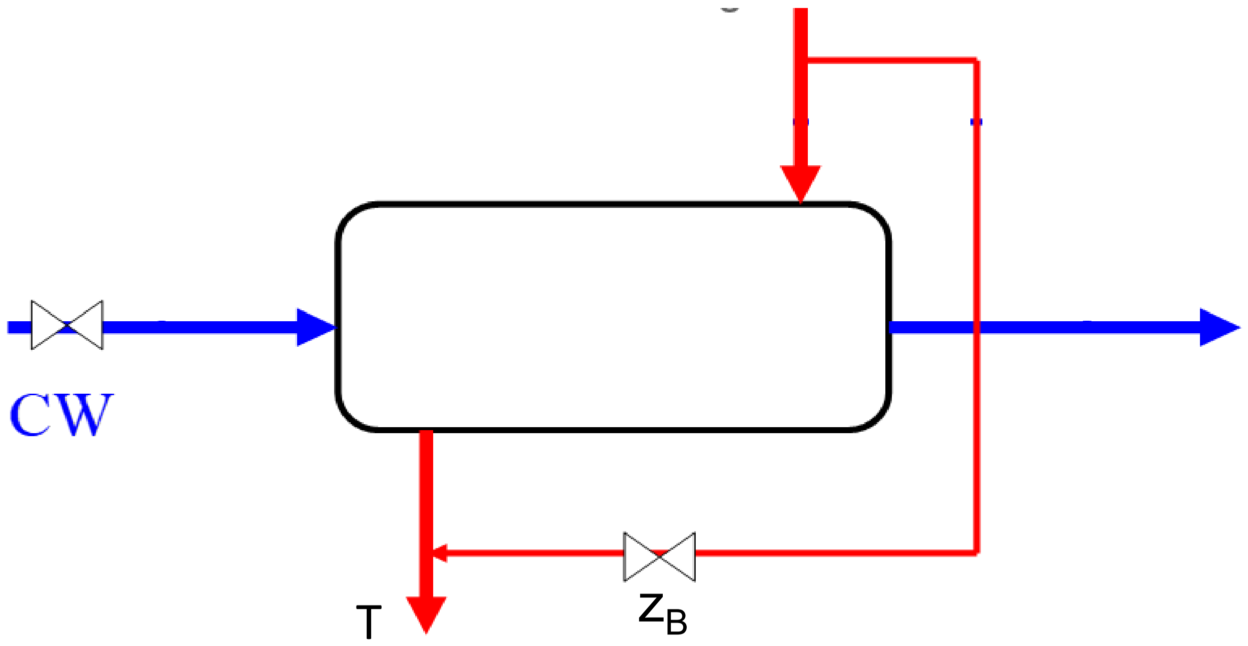
Example 2: Strong base (u_2) and weak base (u_1) for neutralizing acid (disturbance) to control y=pH

Alternative term for dynamic VPC:

- Mid-ranging control (Sweden)

NTNU

Example: Heat exchanger with bypass

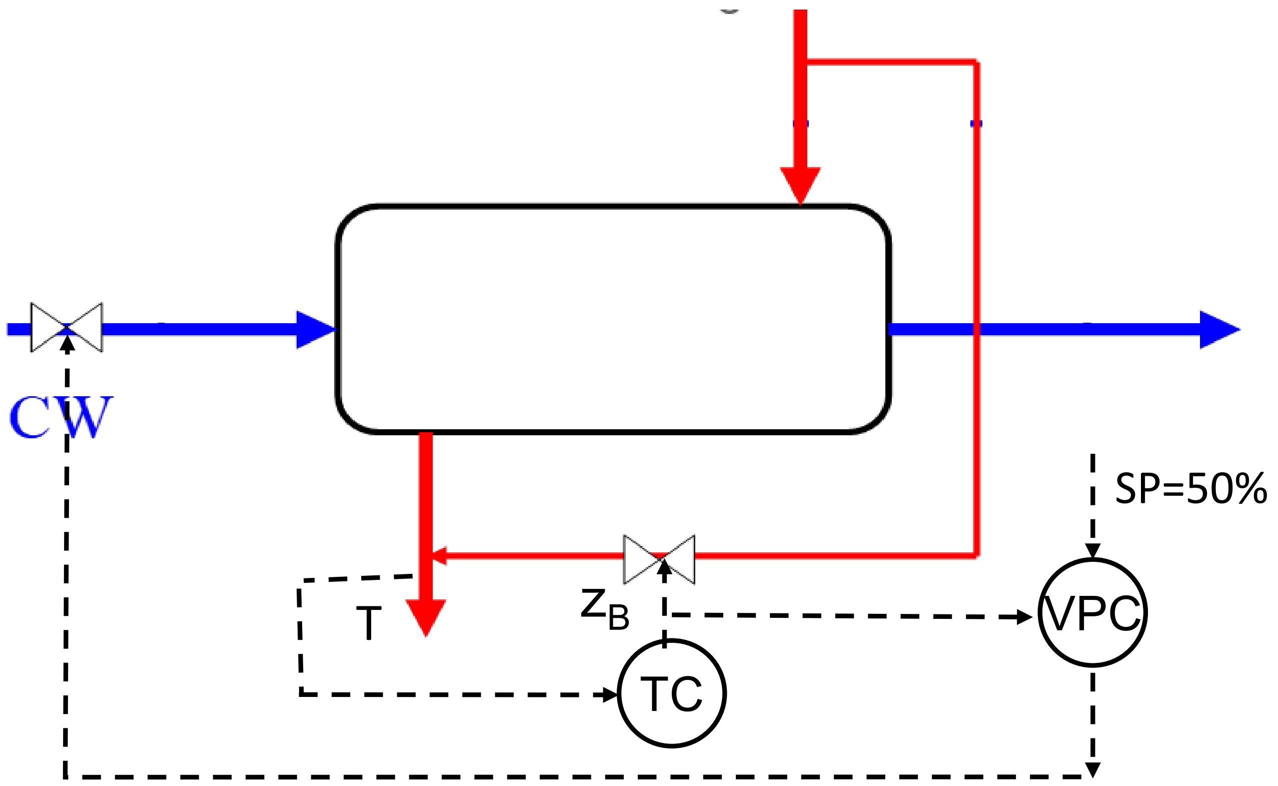


Want tight control of y=T.

- $u_1=z_B$ (bypass)
- u_2=CW

Proposed control structure?

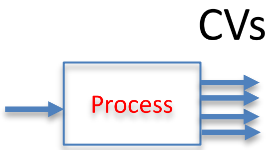
Example: VPC for heat exchanger



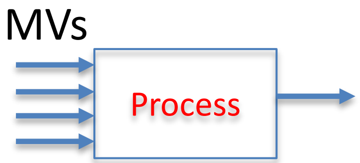
- Fast control of y: $u_1 = z_B$
- Main control (VPC): u_2=CW (slow loop)
- Need time scale separation between the two loops

Constraint switching
(because it is optimal at steady state)

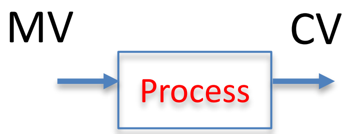
- CV-CV switching
  - Control one CV at a time



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

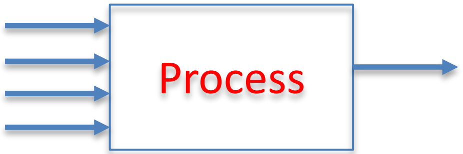


- MV-CV switching
  - MV saturates so must give up CV
  1. Simple («do nothing»)
  2. Complex (repairing of loops)



MV = manipulated variable
CV = controlled variable

MV-MV switching



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

Three solutions:

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

Alt.2 Several controllers with different setpoints (E6)

Alt.3 Valve position control (E7)

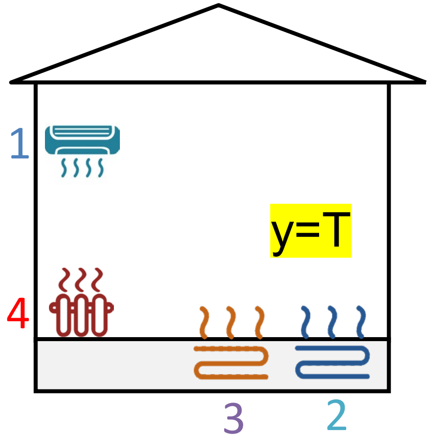
Which is best? It depends on the case!

***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

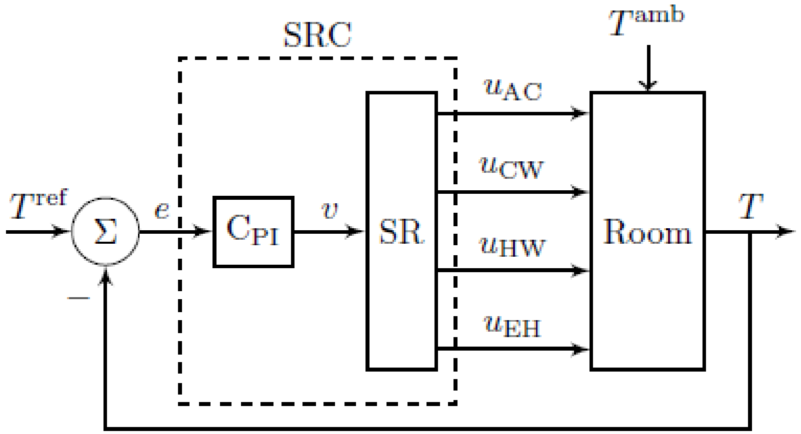
- Break and gas pedal in a car
- Use only one at a time,
- «manual split range control»

Example split range control (E5) : Room temperature with 4 MVs

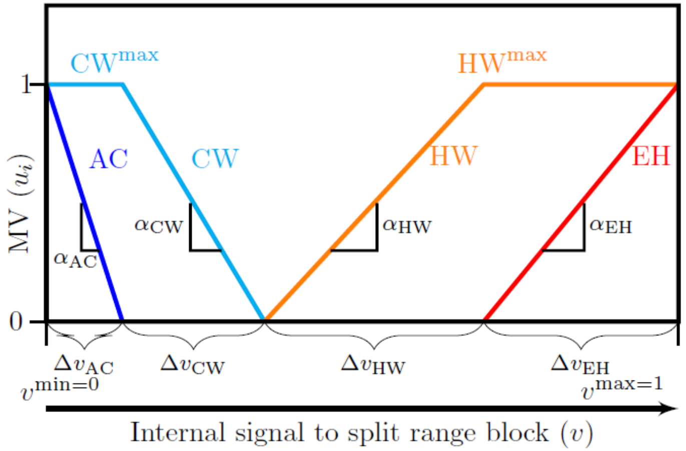


MVs (two for summer and two for winter):

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

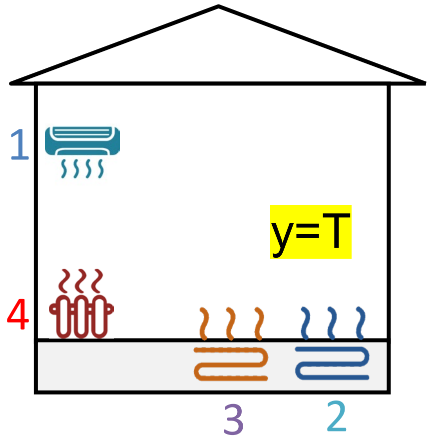


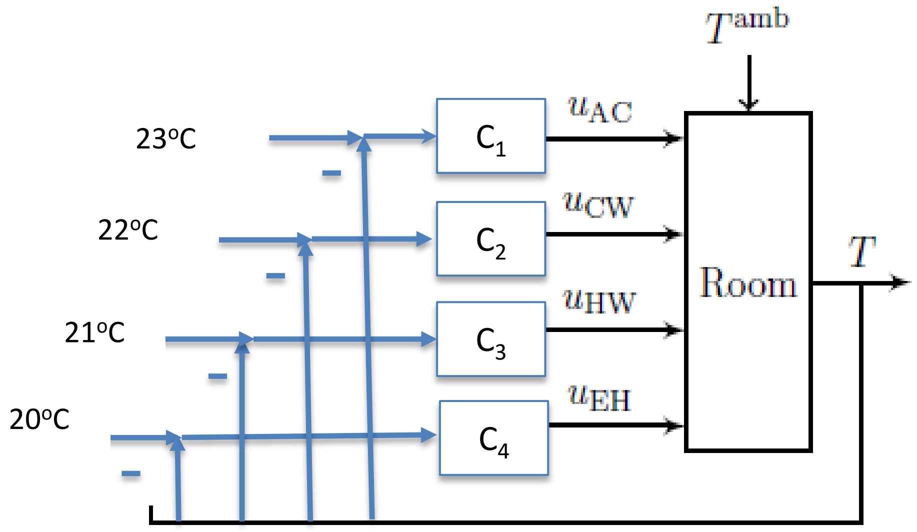
SR-block:



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

Alternative: Multipliple Controllers with different setpoints (E6)





Disadvantage (comfort):

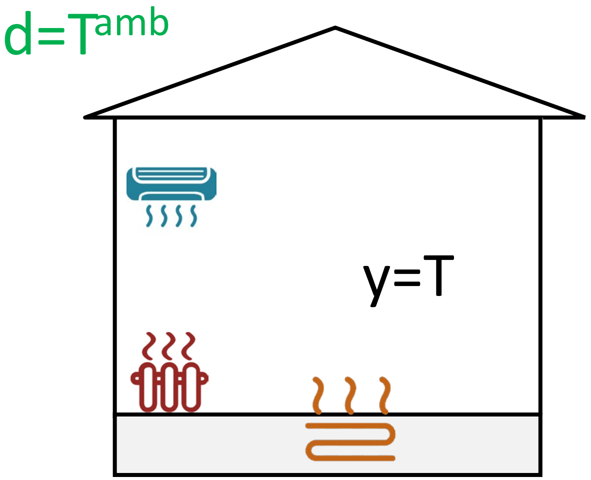
- Different setpoints

Advantage (economics) :

- Different setpoints (energy savings)

Simulation Room temperature

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



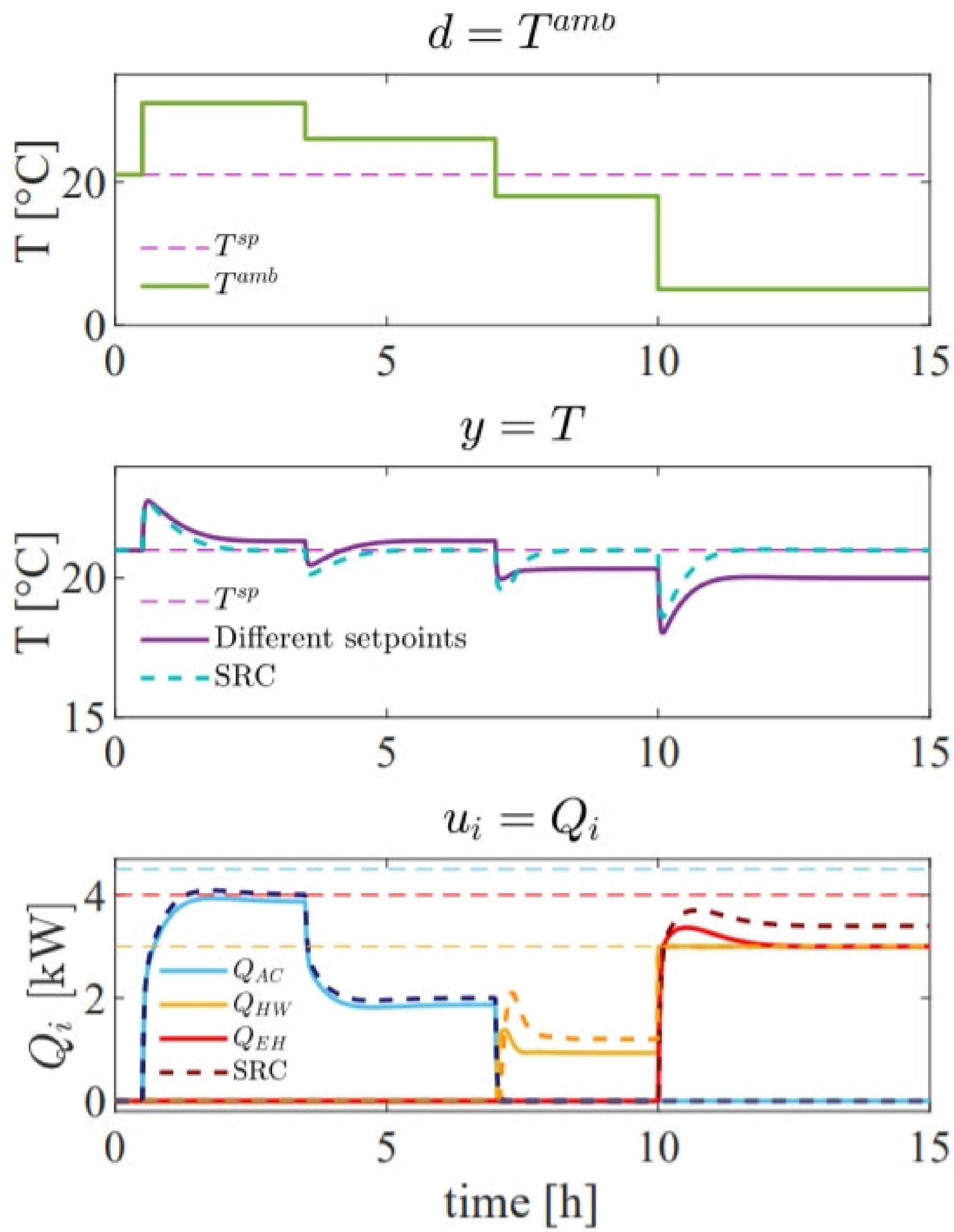


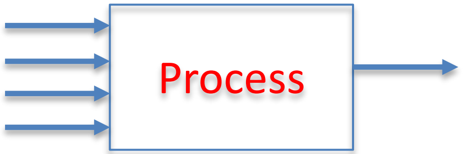
Table 1. Ranges for available inputs (u_k).

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)

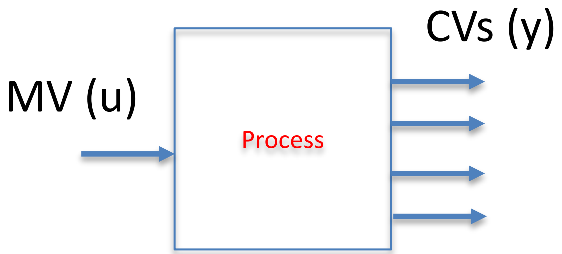
Alt.3 Valve position control (E7)

- 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** ,

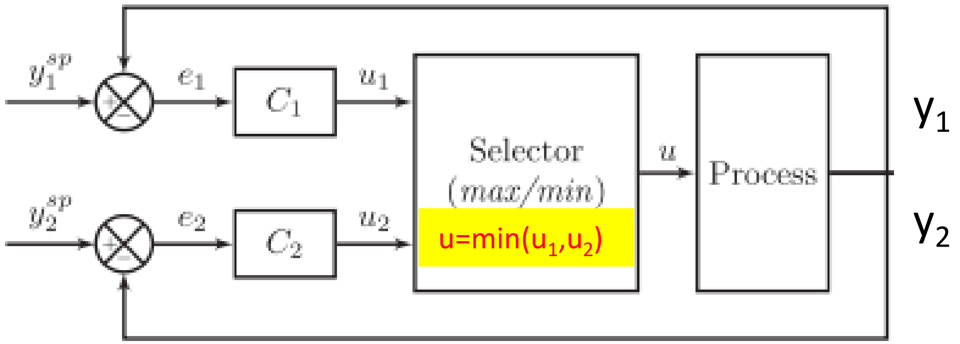
CV-CV switching

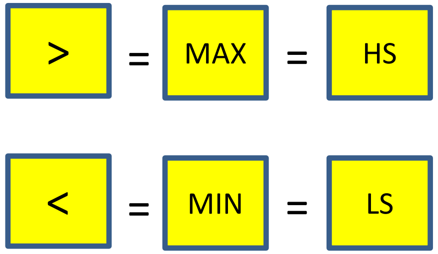


- Only one input (MV) controls many outputs (CVs)
  - Typically caused by change in active constraint
  - Example: Control car speed (y_1) - but give up if too small distance (y_2) to car in front.

- Use max- or min-selectors (E4)

E4. Selector: One input (u), several outputs (y_1,y_2)





- Note: The selector is 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 are used for output-output (CV-CV) switching
- Selectors work well, but require pairing each constraint with a given input (not always possible)

NTNU

Furnace control with safety constraint

Input (MV)

u = Fuel gas flowrate

Output (CV)

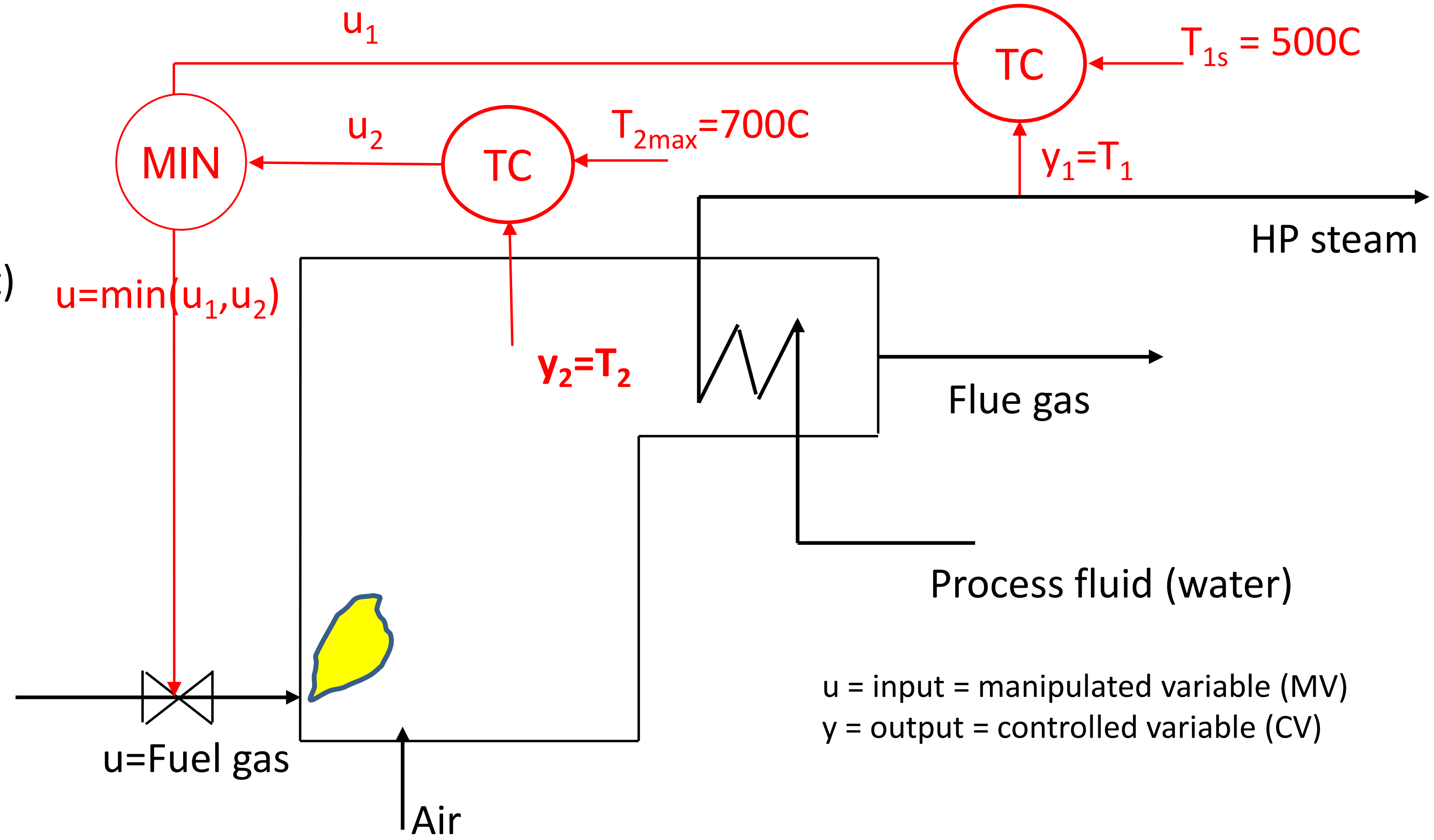
y_1 = process temperature T_1

(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
- 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)

Example. Maximize flow with pressure constraints

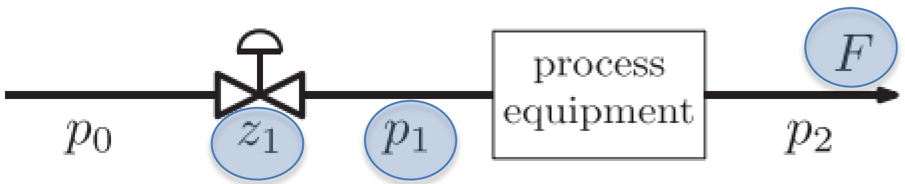


Fig. 6. Example 2: Flow through a pipe with one MV ($u = z_1$).

Input u = z_1
Want to maximize flow, J=-F:

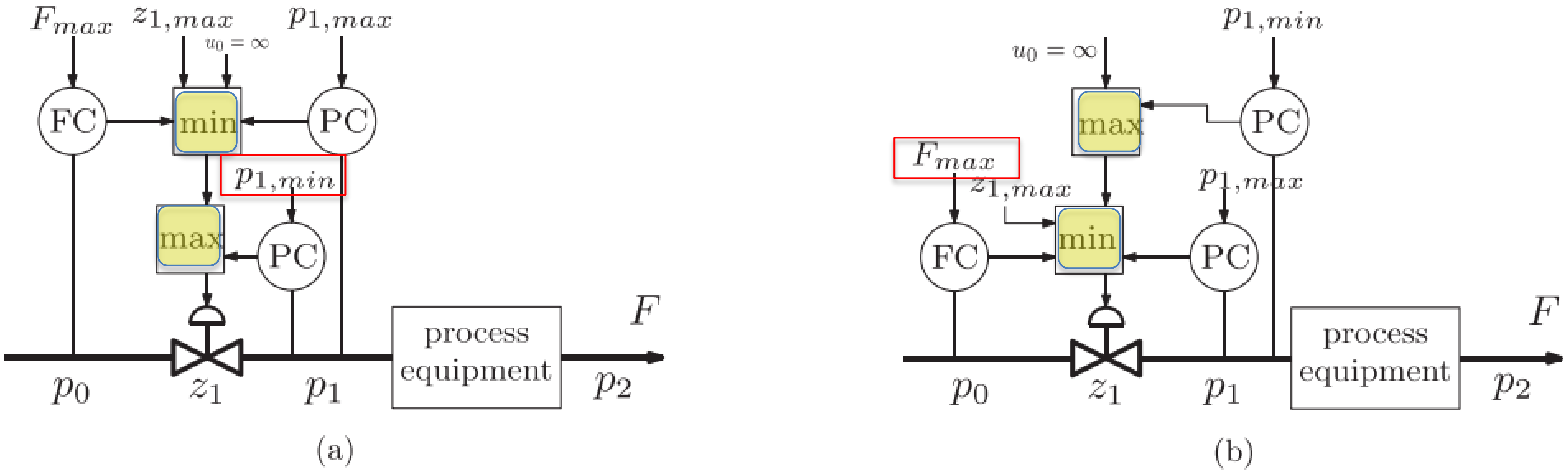
Optimization problem is:

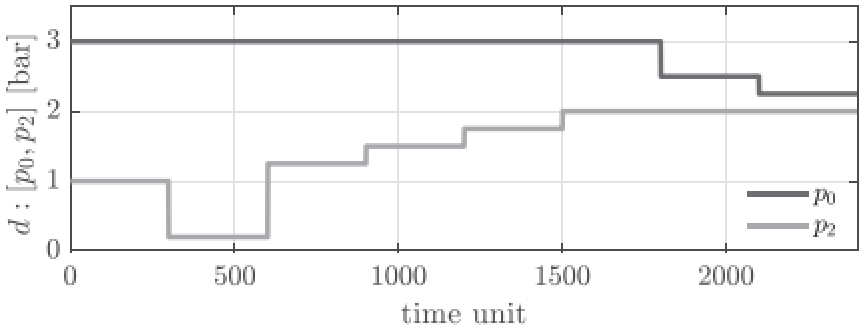
$\max_{z_1} F$ — Satisfied by

s.t.

$F \leqslant F_{max}$	Small u	(15)
$p_1 \leqslant p_{1,max}$	Small u	
$p_1 \geqslant p_{1,min}$	Large u	Possible conflict
$z_1 \leqslant z_{1,max}$	-	

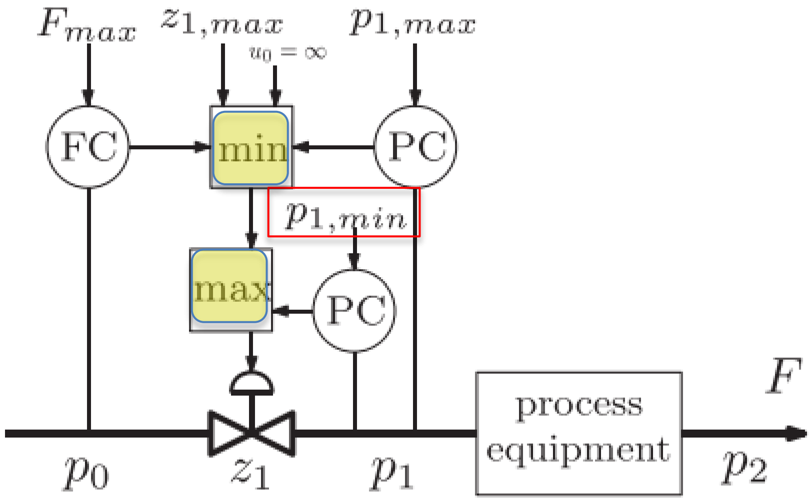
where F_{max} =10 kg/s, $z_{1,max} = 1$, $p_{1,max} = 2.5$ bar, and $p_{1,min} = 1.5$ bar. Note that there are both max and min- constraints on p_1. De-



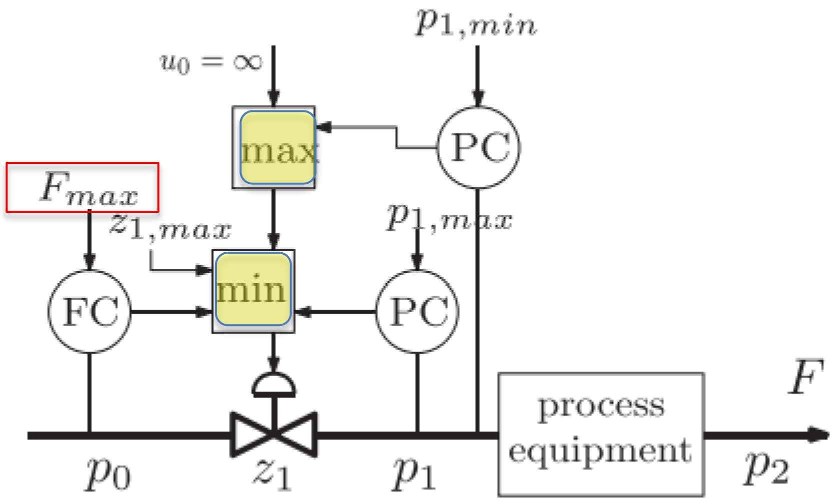


Disturbances in p_0 and p_2 (unmeasured)

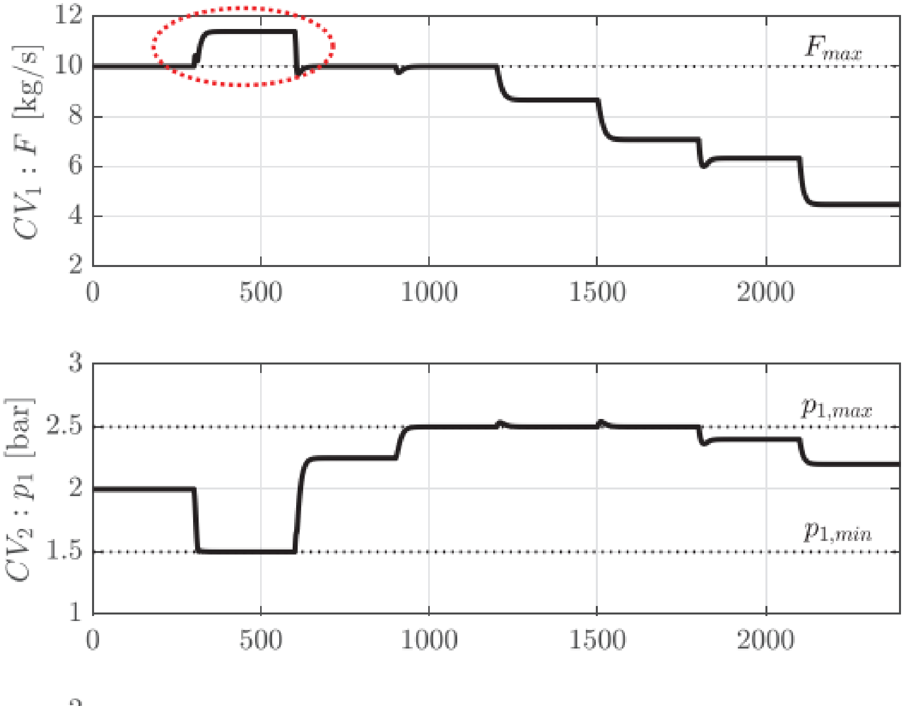
NTNU

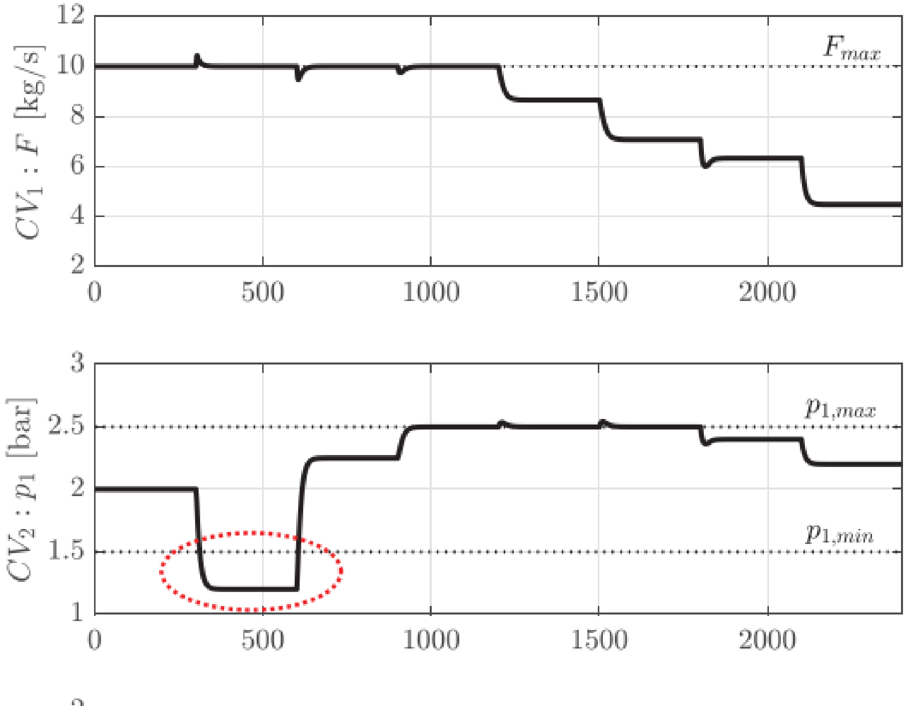


(a)



(b)



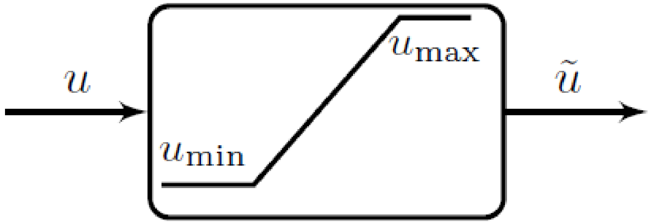


t>1800: u=zmax=1

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 ways (equivalent because never conflict)
1. Min-selector followed by max-selector
2. Max-selector followed by min-selector
3. Mid-selector

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

NTNU

MV-CV switching (because reach constraint on MV)

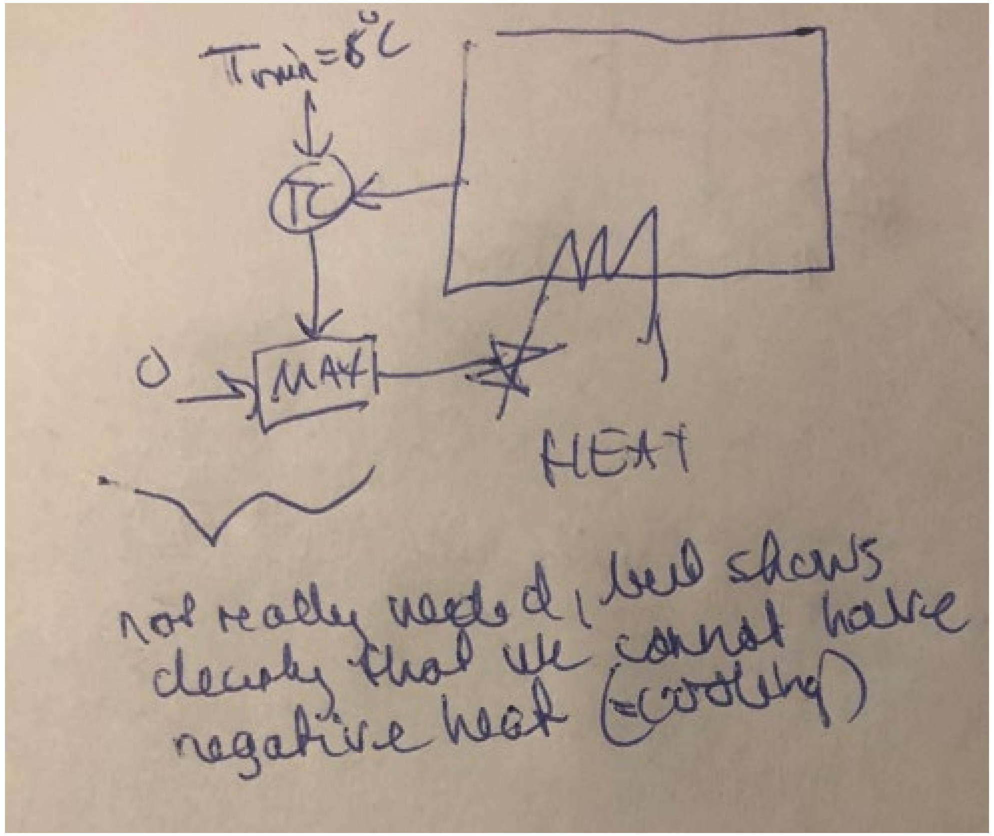
- **Simple CV-MV switching**
  - Don’t need to do anything if we followed the *Input saturation rule:*
  - *“Pair a MV that may saturate with a CV that can be given up (when the MV saturates)”*

Example: Avoid freezing in cabin

$$\text{Minimize } u \text{ (heating), subject to}$$
$$T \geq T_{min}$$
$$u \geq 0$$

Keep CV=T>T_{min} = 8C in cabin in winter by using MV=heating

If it's hot outside (>8C), then the heat will go to zero (MV=Q=0), but this does not matter as the constraint is over-satisfied.



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

Minimize recycle (MV=z) subject to

CV= $F \geq F_{min}$

MV ≥ 0

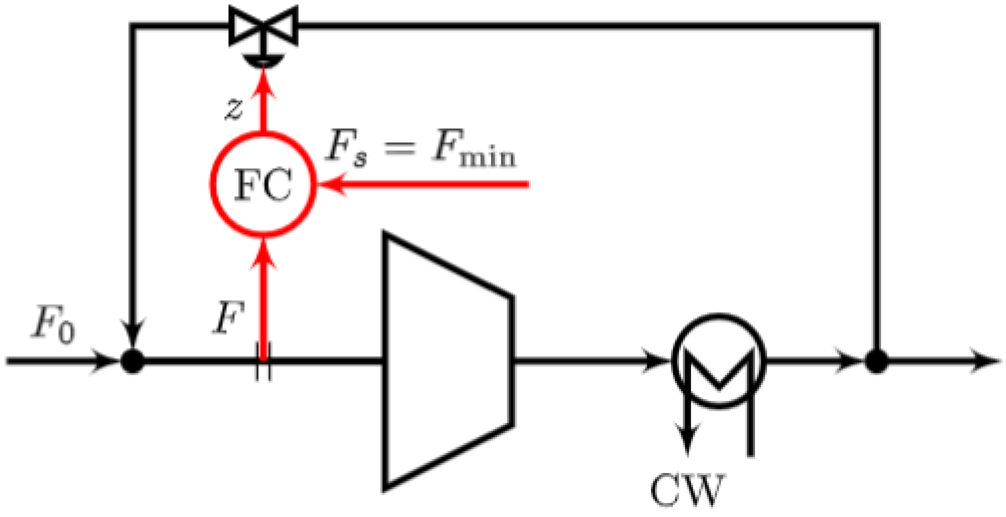


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding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

Example: Compressor with max-constraint on F_0
(in addition to the min-constraint on F)

Minimize u (recycle), subject to

$u = z \geq 0$

$CV_1 = F \geq F_{min}$

$CV_2 = F_0 \leq F_{0,max}$

- Both constraints are satisfied by a large z
  ⇒ Max-selector for CV-CV
- When we reach MV-constraint (z=0) both constraints are oversatisfied
  ⇒ Simple MV-CV switching

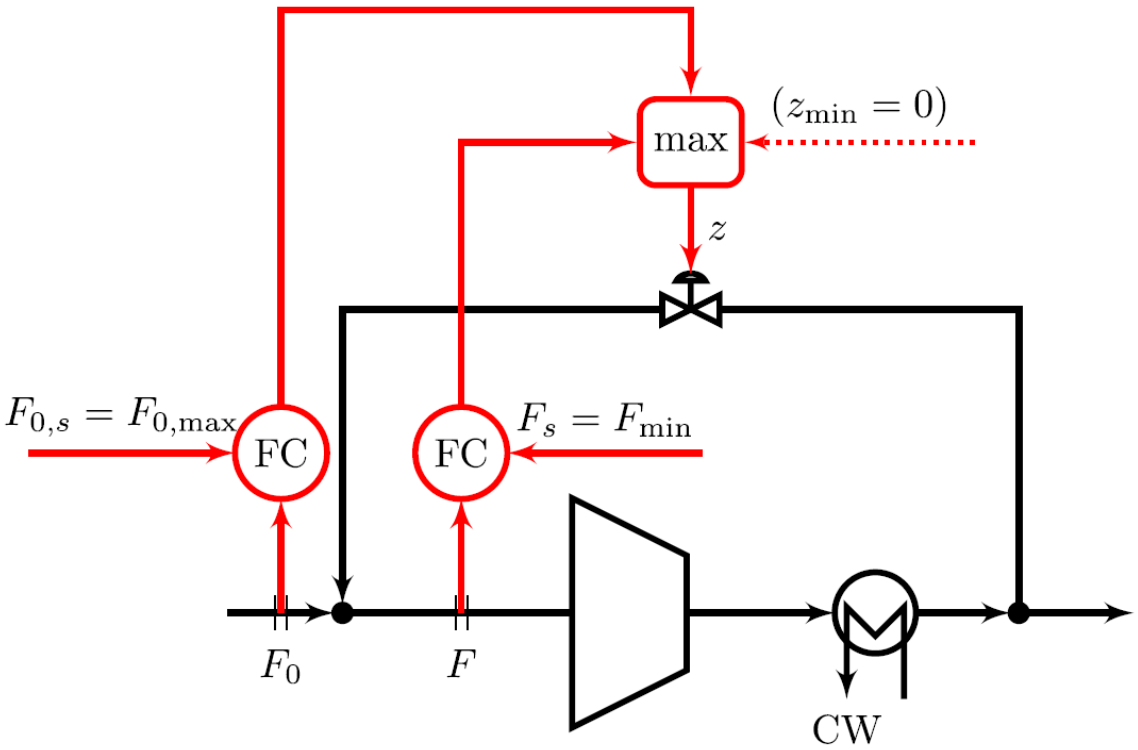
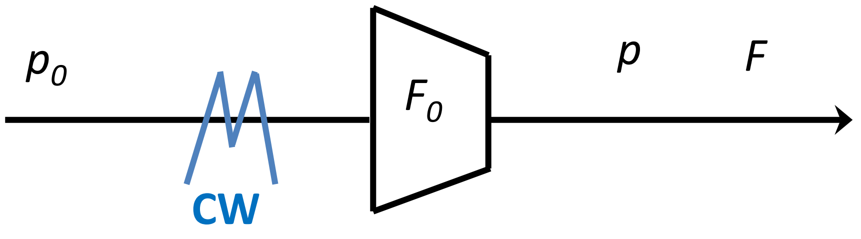


Fig. 33. Anti-surge compressor control with two CV constraints. This is an example of simple MV-CV-CV switching. MV $= z$, $CV_1 = F$, $CV_2 = F_0$ (all potentially active constraints).

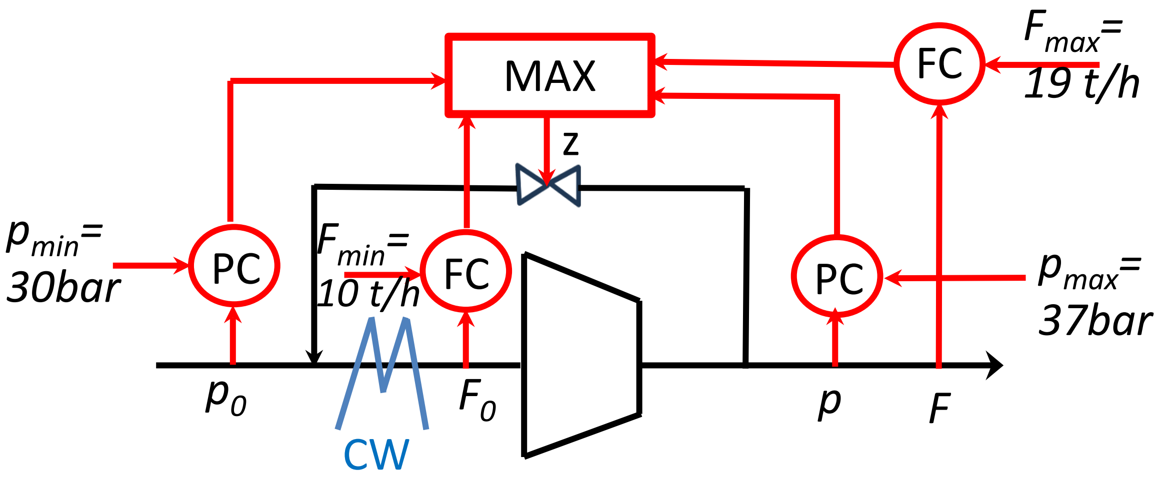
NTNU

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)



Rule CV-CV switching: Use max-selector for constraints that are satisfied by a large input (MV) (here: valve opening z)

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)"*

Complex MV-CV switching

- Didn't follow input saturation rule
- This is a repairing of loops
- Need to combine MV-MV switching with CV-CV-switching
  - The CV-CV switching always uses a selector
  - As usual, there are three alternatives for the MV-MV switching:
    1. Split range control (block /\): Has problems because limits may change
    2. Several controllers with different setpoints (often the best for MV-CV switching)
    3. Valve position control (Gives «long loop» but avoids repairing).

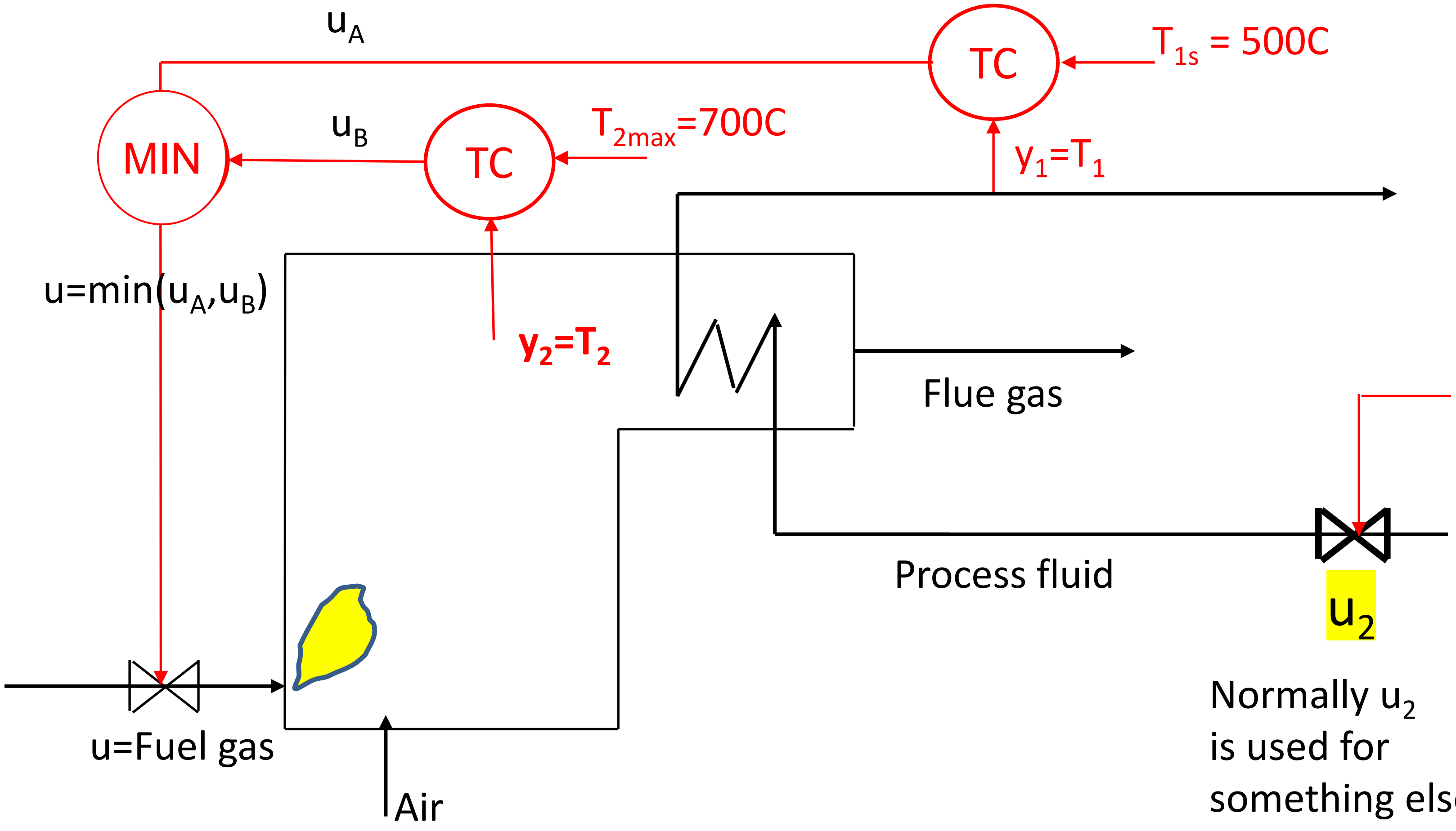
Furnace control : Cannot give up control of y_1=T_1. What to do?

Inputs (MV)

- u = Fuel gas flowrate
- u_2 = Process flowrate

Output (CV)

- y_1 = process temperature T_1 (with desired setpoint)



Normally u_2 is used for something else

Cannot give up controlling T_1

Solution: Cut back on process feed (u_2) when T_1 drops too low

Inputs (MV)

- u = Fuel gas flowrate
- u_2 = Process flowrate

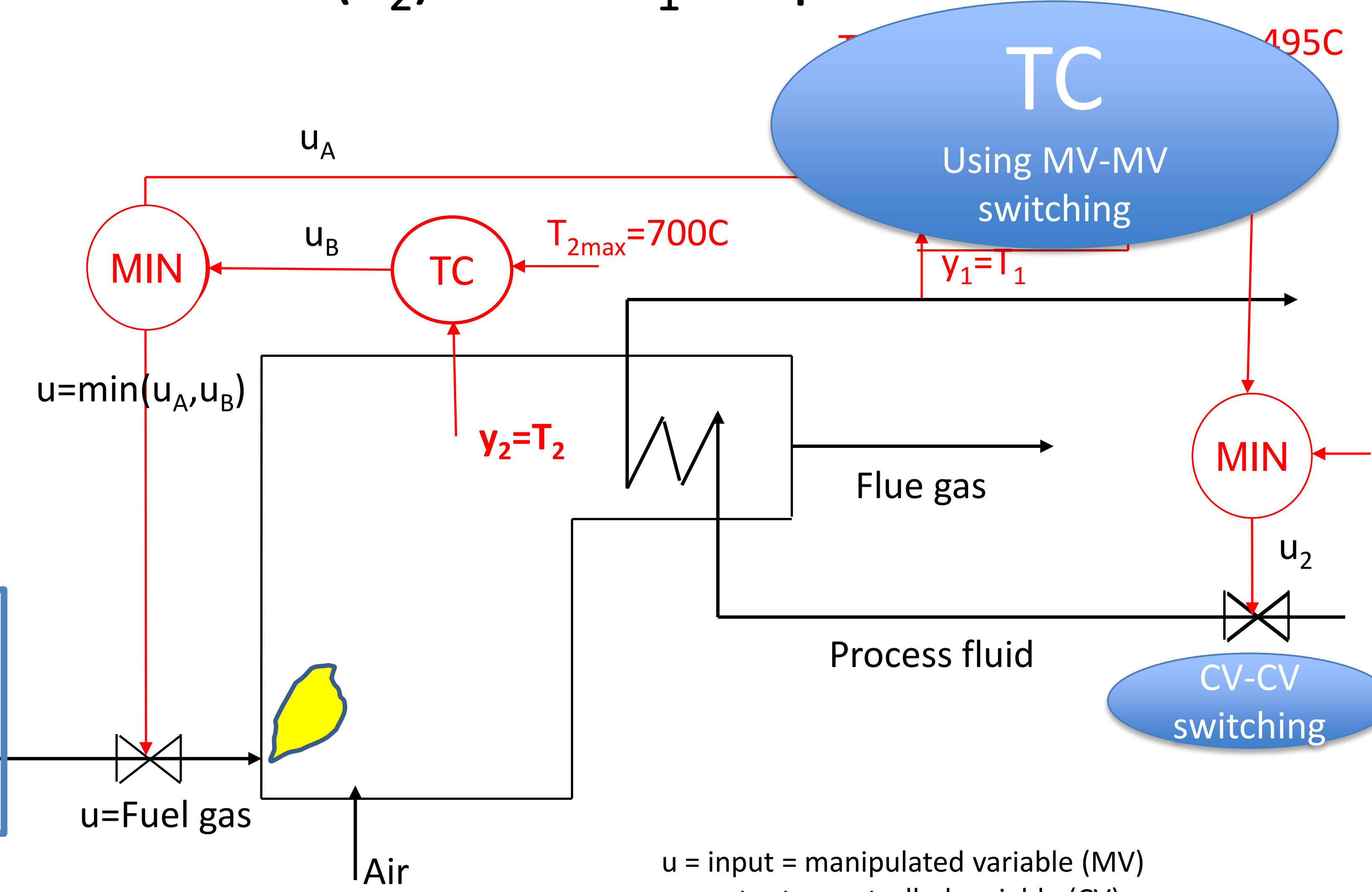
Output (CV)

- y_1 = process temperature (with desired setpoint)

Note: Standard Split Range Control (Alt. 1) is not good here for MV-MV swiitching.
Could be two reasons for too little fuel

- Fuel is cut back by override (safety)
- Fuel at max,

So don’t know limit for MV1 to use in SRC-block.



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

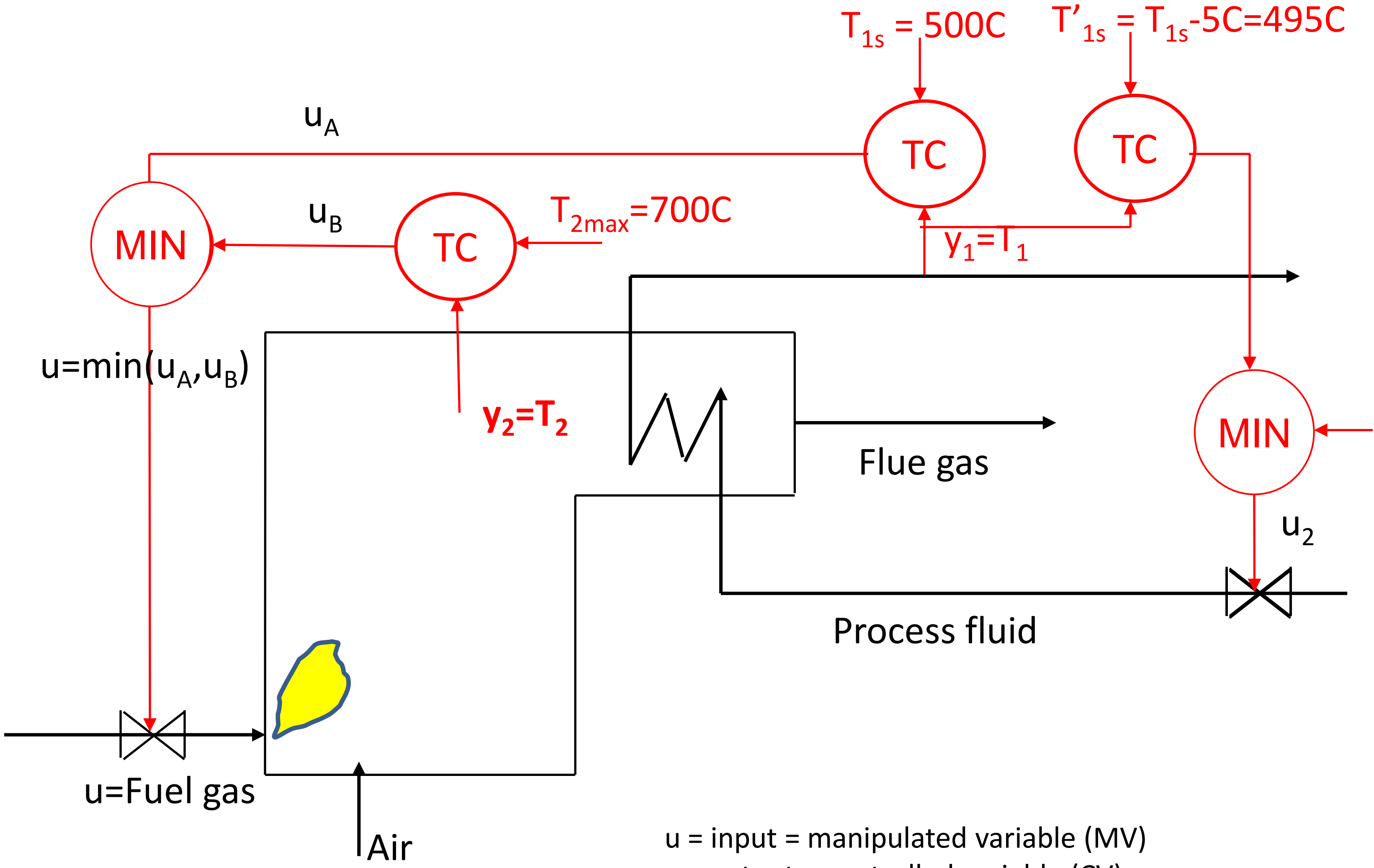
Use Alt. 2: Two controllers

Inputs (MV)

- u = Fuel gas flowrate
- u2 = Process flowrate

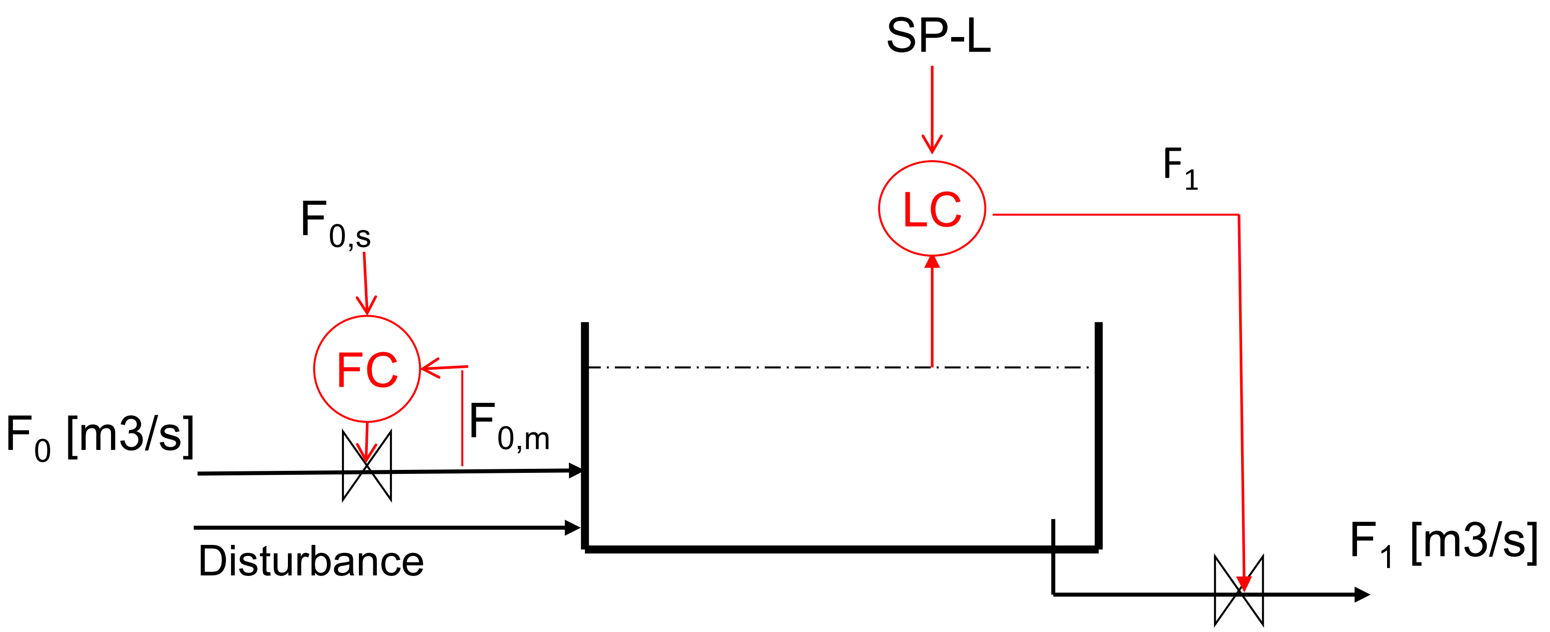
Output (CV)

- y_1 = process temperature (with desired setpoint)



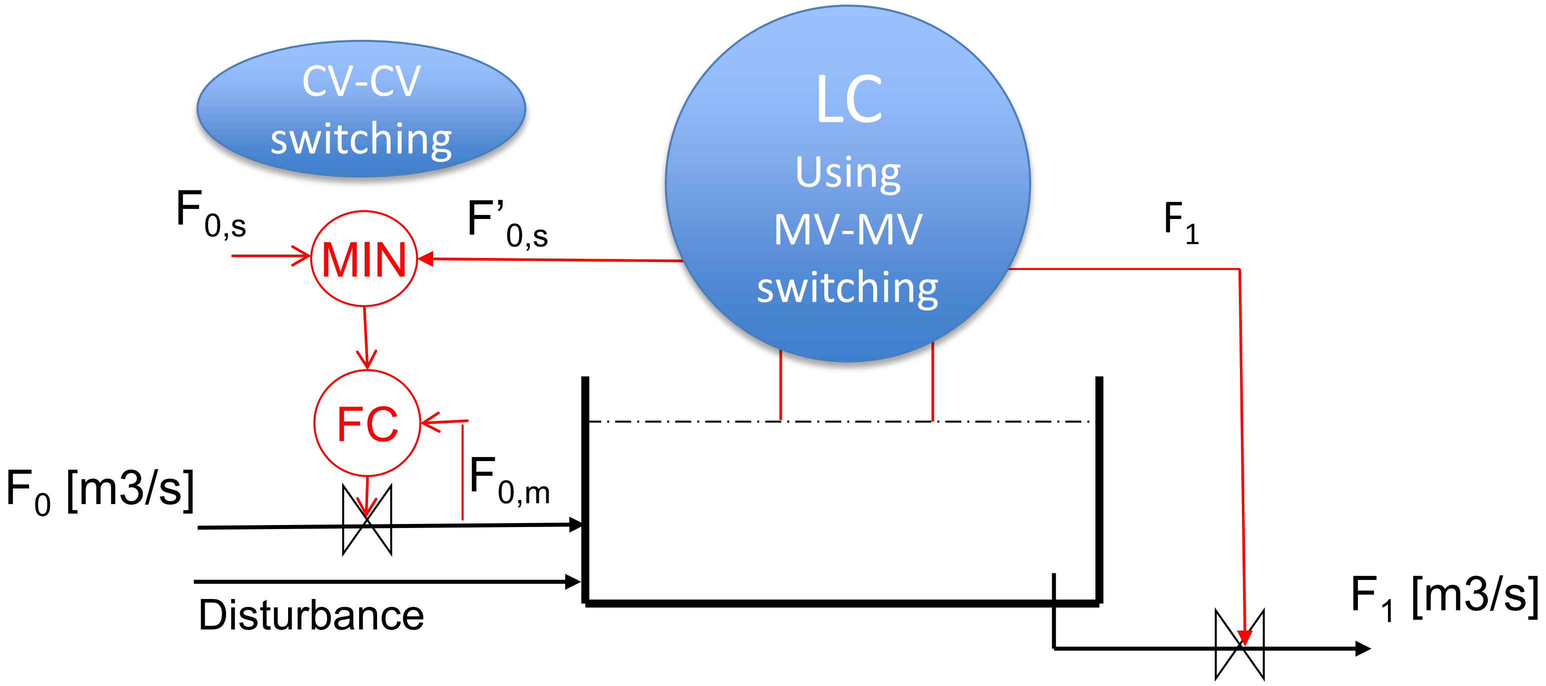
u = input = manipulated variable (MV)
y = output = controlled variable (CV)

Example: Level control



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

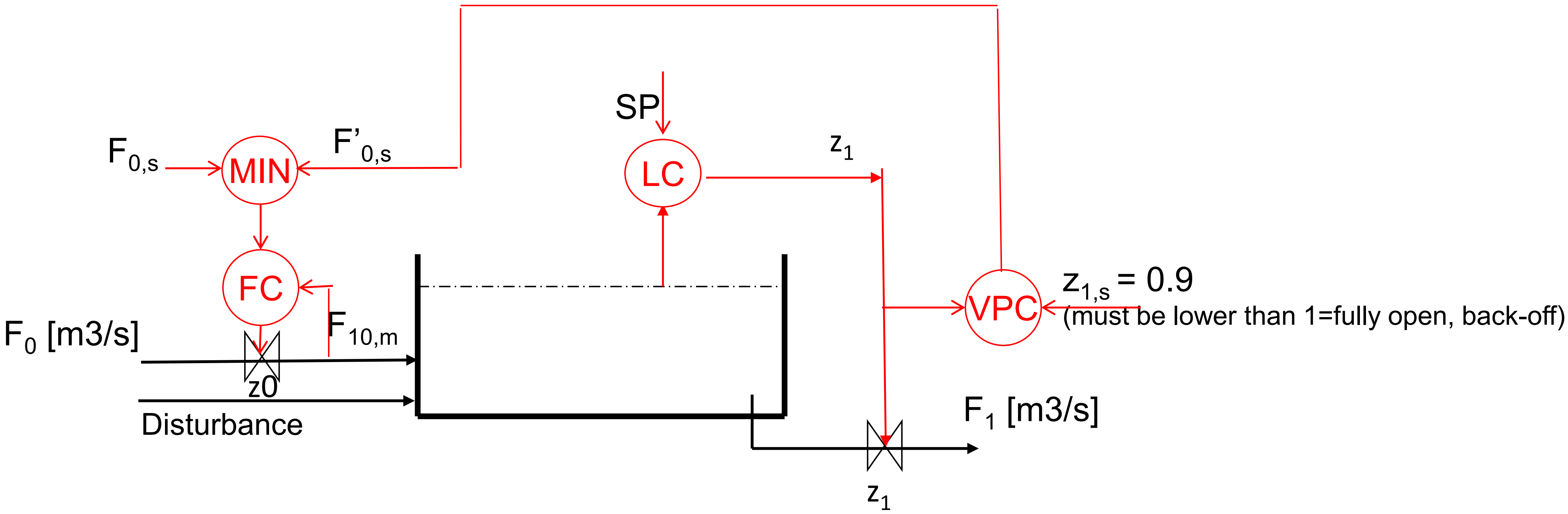
"Bidirectional inventory control"



Three alternsatives for MV-MV switching

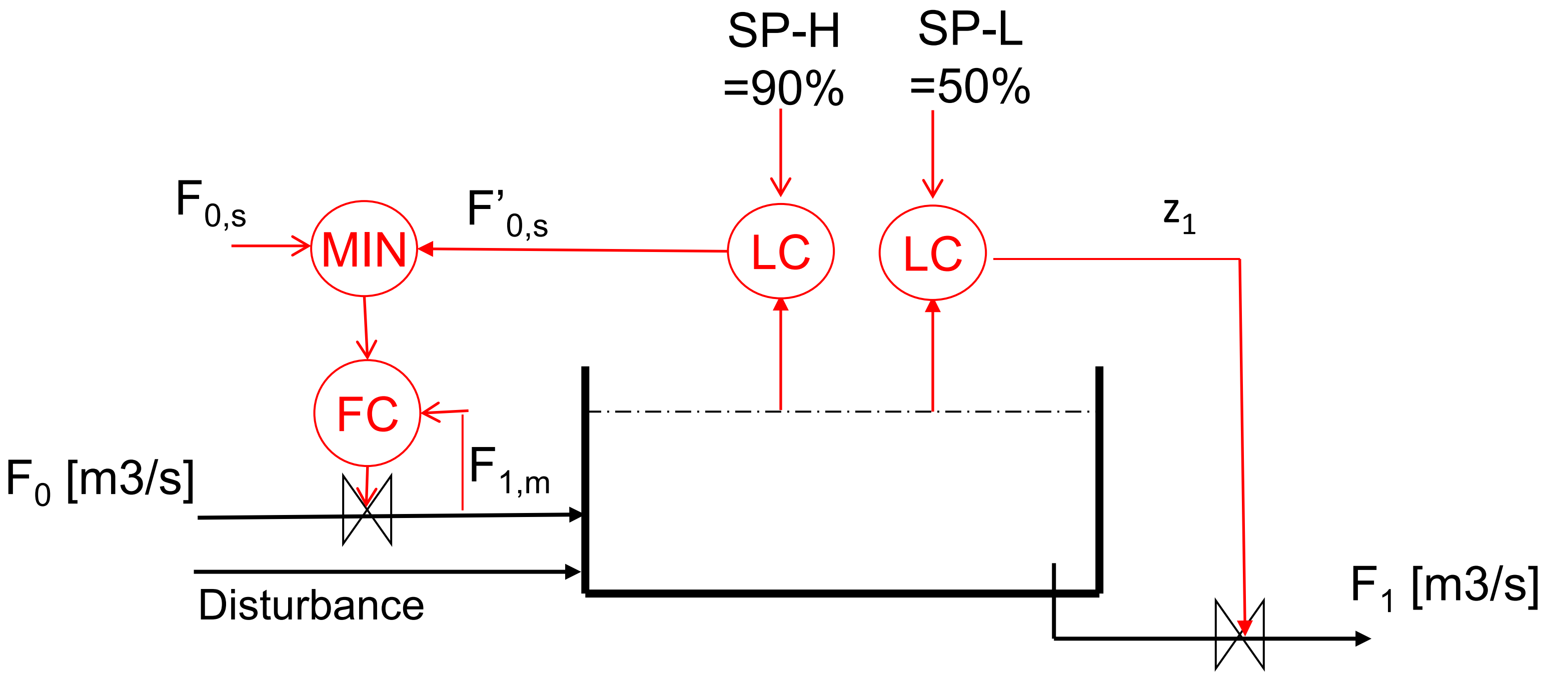
1. SRC (problem since F_{0s} varies)
2. Two controllers
3. VPC ("Long loop" for F1)

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)



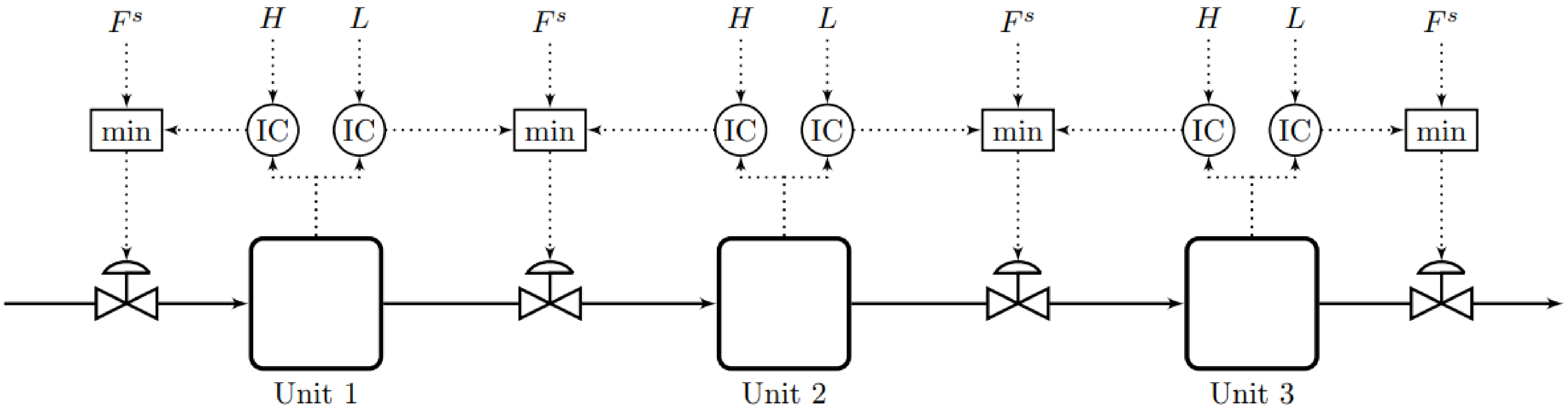
SP-L = low level setpoint
SP-H = high level setpoint

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

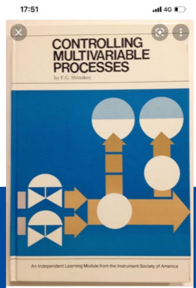
Inventories in series . Very smart selector strategy based on **Bidirectional inventory control**

Reconfigures automatically with optimal buffer management!!

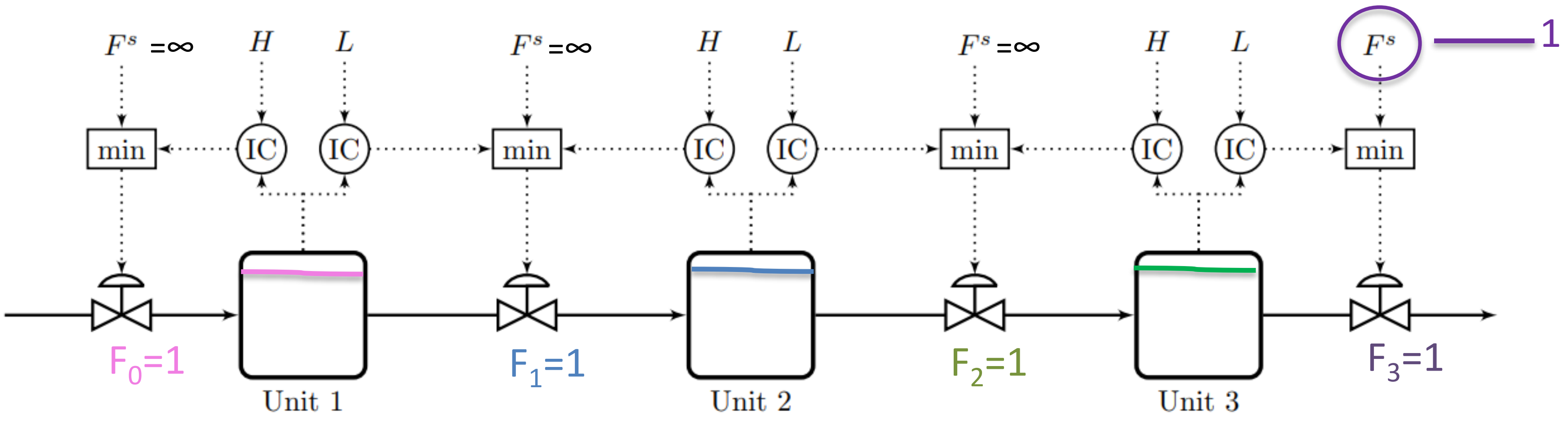
Max flow:
$F^s=\infty$



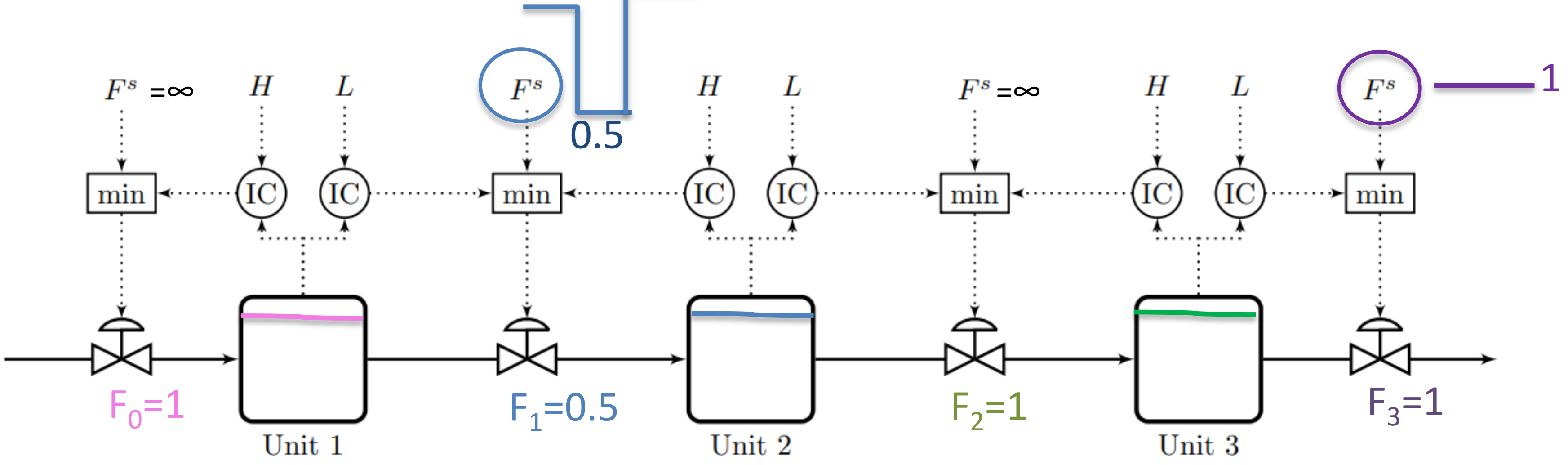
F.G. Shinskey, «Controlling multivariable processes», ISA, 1981
C. Zotica, S. Skogestad and K. Forsman, Comp. Chem. Eng, 2021

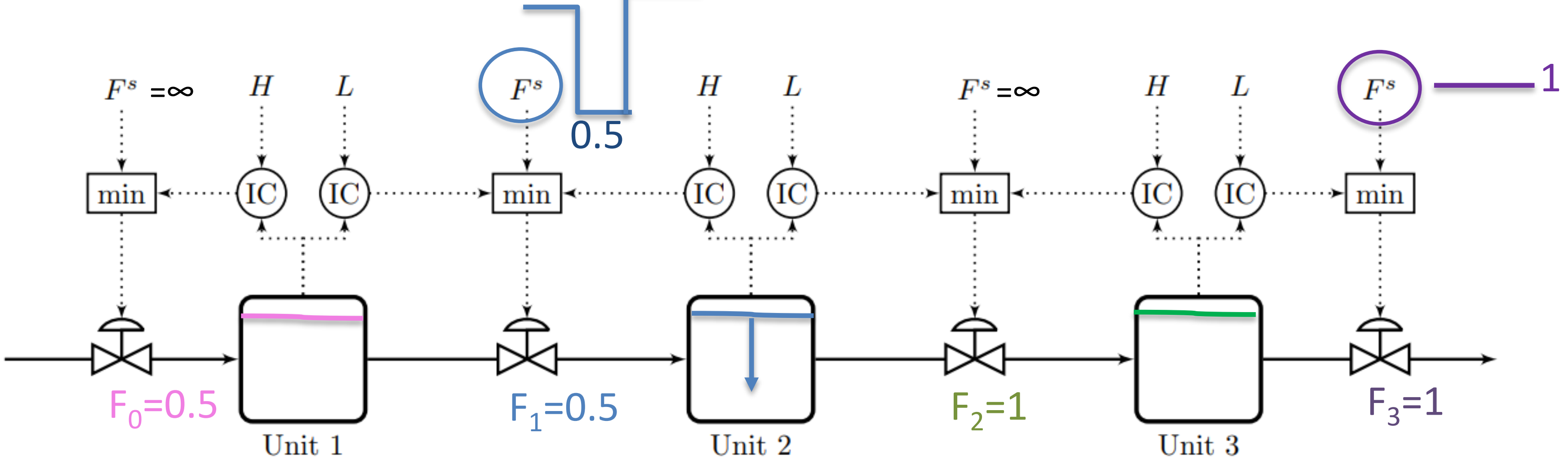


NTNU

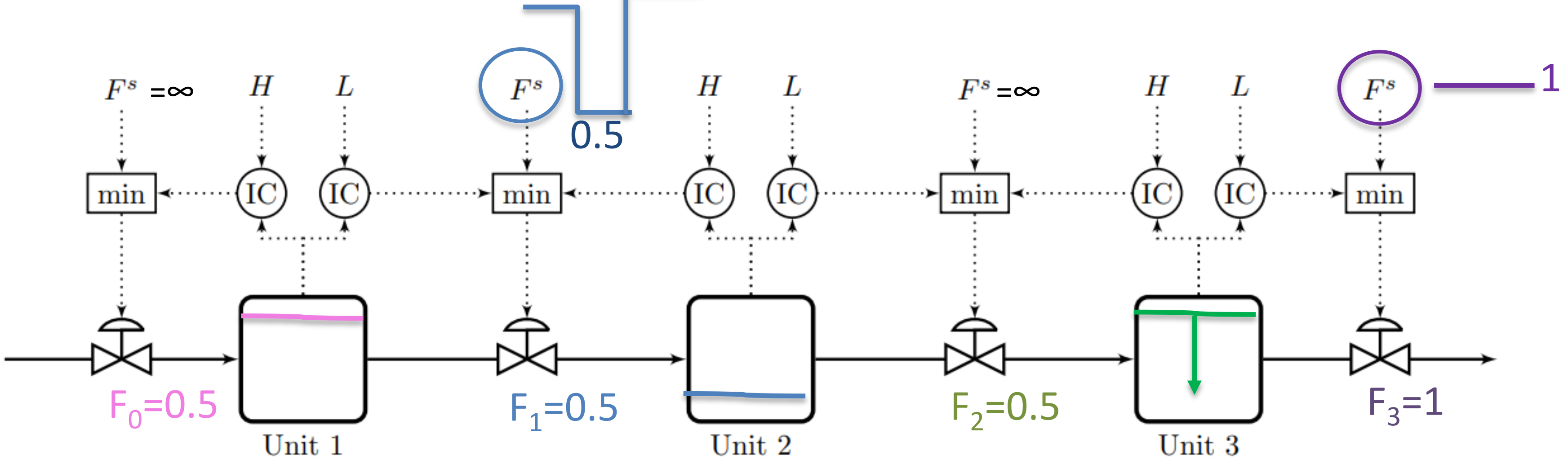


$F^s = \infty$
H
L
$F^s = \infty$
H
L
$F^s = \infty$
H
L
F^s
1
min
IC
IC
min
IC
IC
min
IC
IC
min
$F_0=1$
$F_1=1$
$F_2=1$
$F_3=1$
Unit 1
Unit 2
Unit 3


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



$F^s = \infty$
H
L
F^s
0.5
H
L
$F^s = \infty$
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
$F_2=1$
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
F0=0.5
Unit 1
F1=0.5
Unit 2
F2=0.5
Unit 3
F3=1

NTNU

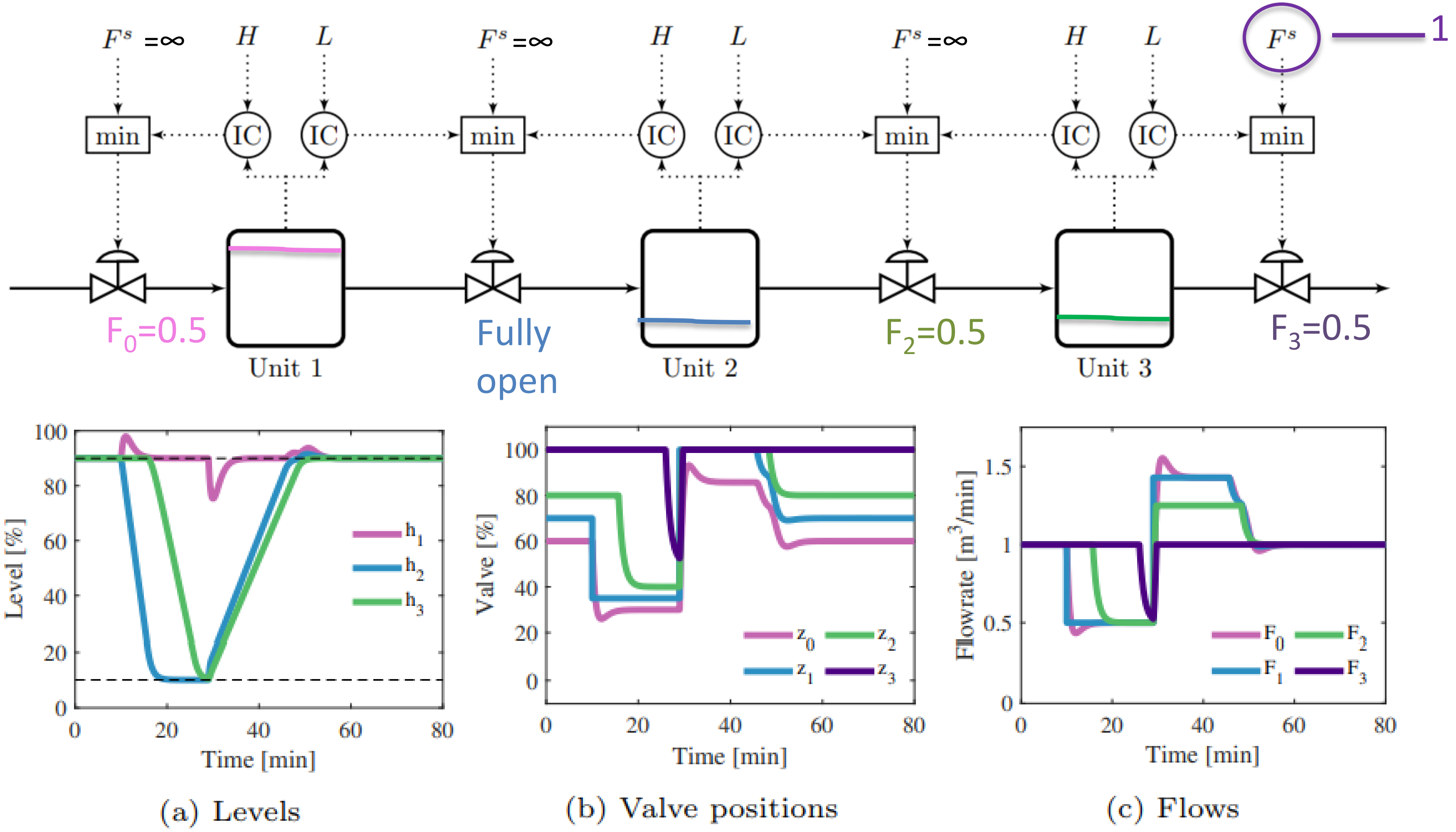
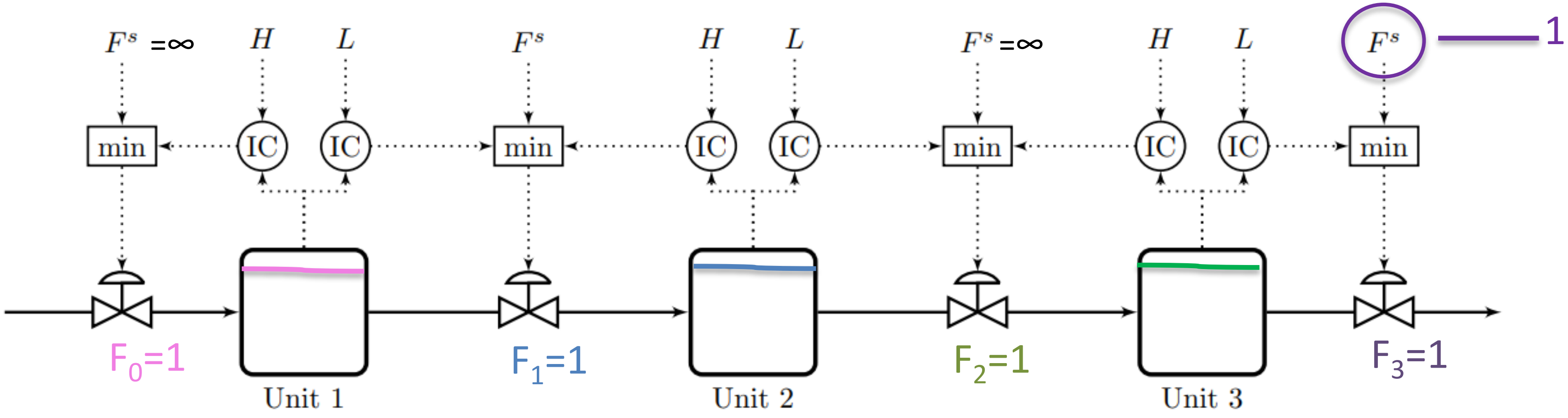


Figure 12: Simulation of a 19 min temporary bottleneck in flow F_1 for the control structures in Fig. 3d with the TPM downstream of the bottleneck.



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

Industrial application (Sweden)

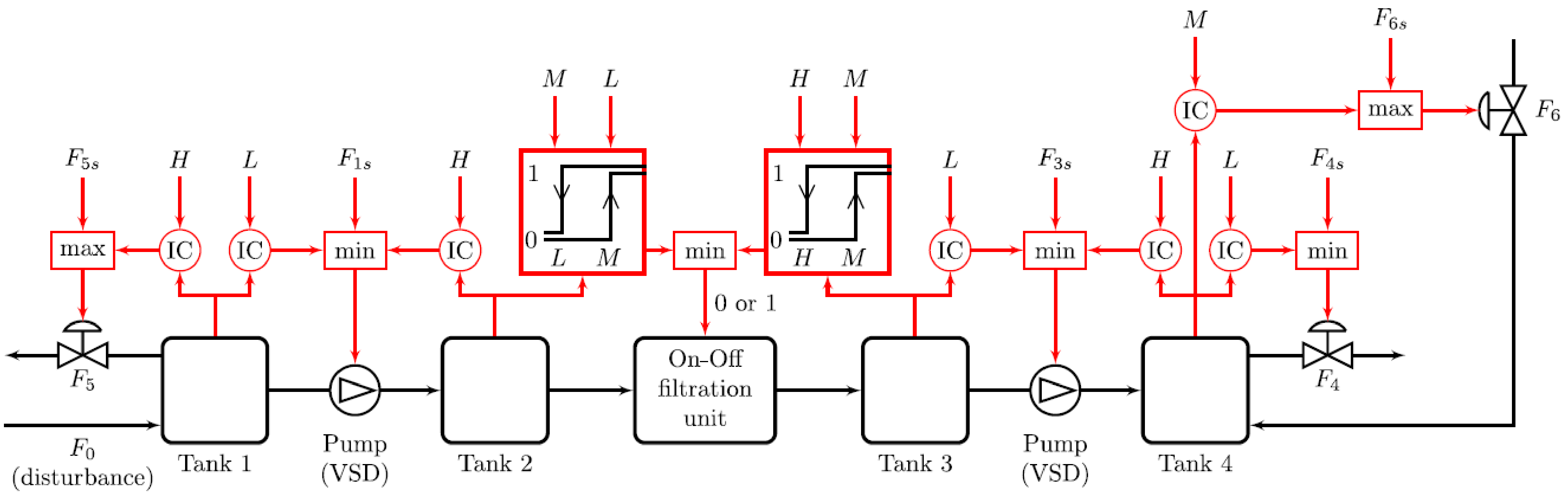


Fig. 38. Bidirectional inventory control structure for industrial plant with on/off (1/0) control of filtration unit.

H, L and M are inventory setpoints with typical values 90%, 10% and 50%.

If it is desirable to set a flowrate (F_s) somewhere in the system, then flow controllers must be added at this location.

Extension . **Bidirectional inventory control with minimum flow for F_2**

Max flow: $F_s = \infty$
L = 10%,
M_L = 40%,
M_H = 60%
H = 90%.

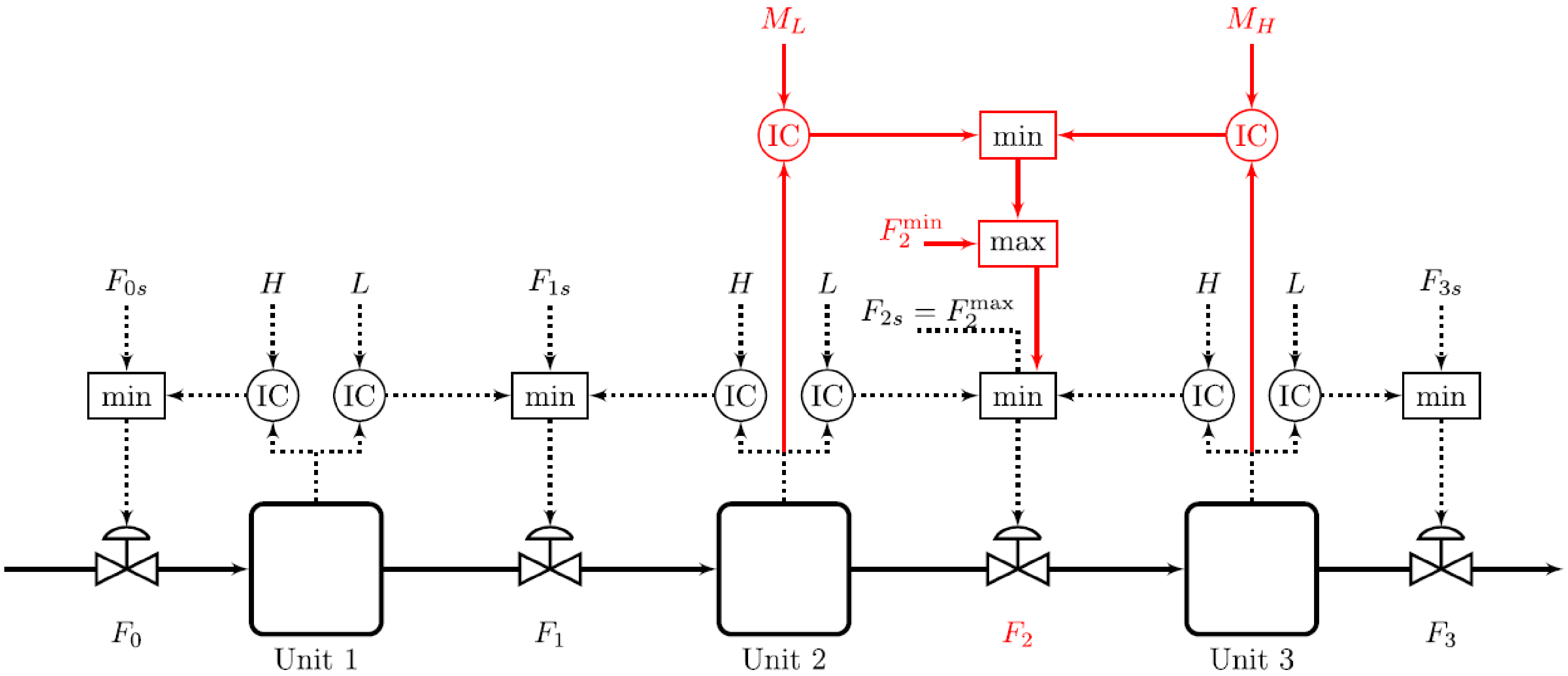


Fig. 37. Bidirectional inventory control scheme for maximizing throughput (dashed black lines) while attempting to satisfy minimum flow constraint on F_2 (red lines). H, L, M_L and M_H are inventory setpoints.

The control structure in Fig. 37 may easily be dismissed as being too complicated so MPC should be used instead. At first this seems reasonable, but a closer analysis shows that MPC may not be able to solve the problem (Bernardino & Skogestad, 2023).[8] Besides, is the control structure in Fig. 37 really that complicated? Of course, it is a matter of how much time one is willing to put into understanding and studying such structures. Traditionally, people in academia have dismissed almost any industrial structure with selectors to be ad hoc and difficult to understand, but this view should be challenged.

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

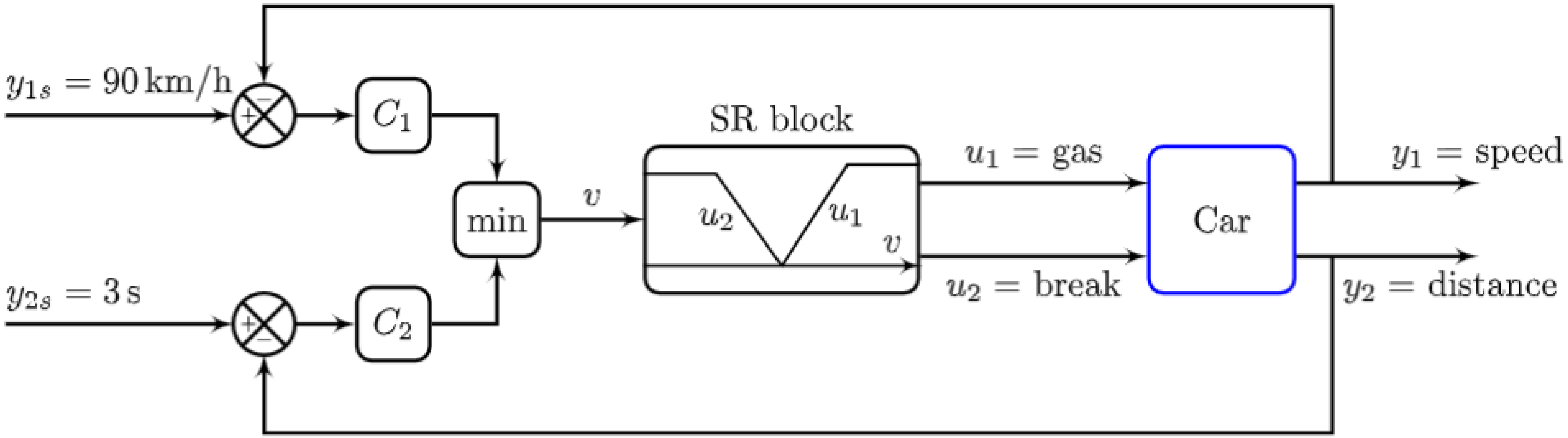


Fig. 31. Adaptive cruise control with selector and split range control.

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

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

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

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)
11. A concise list or library of special (smart) control structures (inventions) that solve specific control problems, for example, cross-limiting control

What about research on PID tuning? Except for the problem of "default tunings", PID tuning has probably received enough academic attention. One exception may be oscillating systems, but these are rare in process control provided robust tunings have been used in the lower-layer control loops. In addition, both for unstable and oscillating processes, a better approach may be to use cascade control on top of a fast inner P- or PD-controller which stabilizes or removes oscillations (see footnote 4). In summary, "PID control" researchers are recommended to switch their attention to "advanced PID control", that is, the interconnection of the PID controller with the other advanced control elements.

In summary, ‘‘PID control’’ researchers are recommended to switch their attention to ‘‘advanced PID control’’, that is, the interconnection of the PID controller with the other advanced control elements.

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

Academic process control community fish pond

Optimal centralized Solution (**EMPC**)

Simple solutions that work (**ARC** = PID++)