ADVANCED PROCESS CONTROL

Sigurd Skogestad

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

CNCA2021, 13 October 2021, Mexico (virtual)


Geiranger fjord


Trondheim


Norwegian Sea
Trondheim
Arctic circle
Sweden
Faroe Islands
Norway
Oslo
Stockholm
North Sea
Edinburgh
Copenhagen
Denmark
United Kingdom
Isle of Man
Dublin
Manchester
Liverpool
Ireland
Hamburg
Berlin
Poland
Amsterdam
Netherlands
London
Brussels
Cologne
Germany
Belgium
Frankfurt
Prague
Luxembourg
Czechia

About Sigurd Skogestad

- 1955: Born in Norway
- 1978: MS (Siv.ing.) in chemical engineering at NTNU
- 1979-1983: Worked at Norsk Hydro co. (process simulation)
- 1987: PhD from Caltech (supervisor: Manfred Morari)
- 1987-present: Professor of chemical engineering at NTNU
- 1994-95: Visiting Professor UC Berkeley
- 2001-02: Visiting Professor UC Santa Barbara
- 1999-2009: Head of ChE Department, NTNU
- 2015-..: Director SUBPRO (Subsea research center at NTNU)

Non-professional interests:

- **mountain skiing (cross country)**
- **orienteering (running around with a map)**
- **grouse hunting**

NTNU

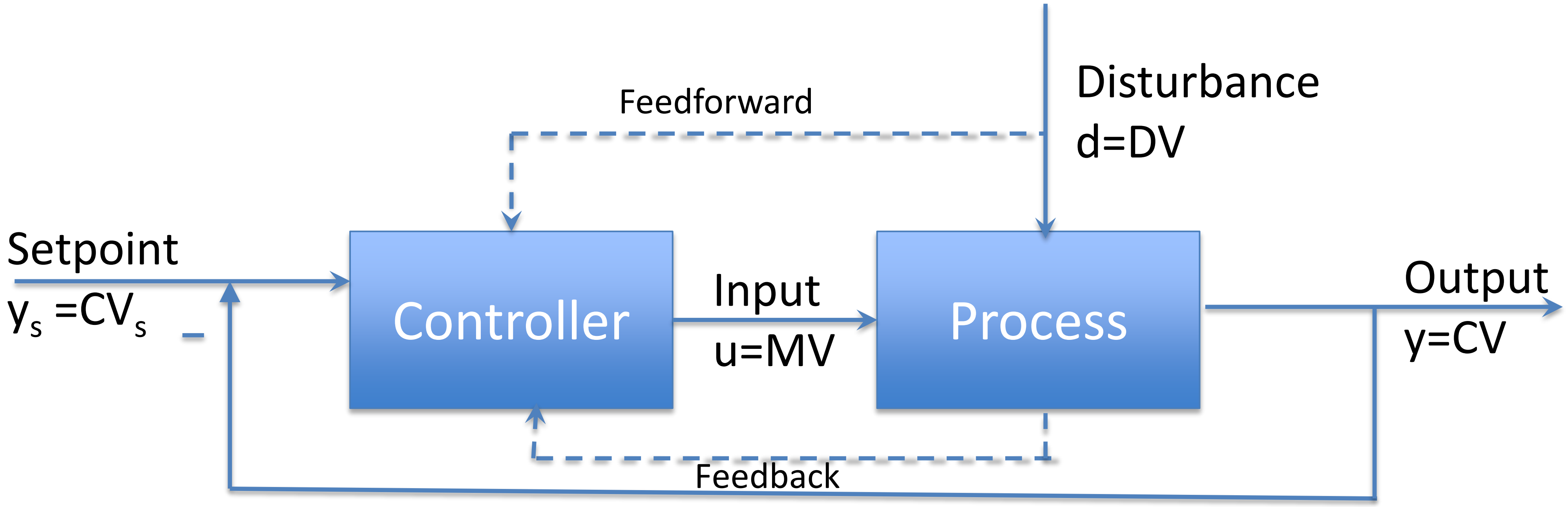
“The goal of my research is to develop simple yet rigorous methods to solve problems of engineering significance”

One example: SIMC PID tuning rules (Skogestad, JPC, 2003)
«Probably the best simple PID tuning rules in the world»

Outline

- Optimization and control
  - Control hierarchy
  - Want tight control of constraints
    - Lagrange multipliers
    - Selectors
- Classical advanced (process) control (APC)
  - Selectors
  - Feedforward, decoupling and linearization blocks
    - Input transformations (Feedback linearization)
  - Other elements
    - Split range control, valve position control, cascade
- Conclusion. MPC vs. classical APC

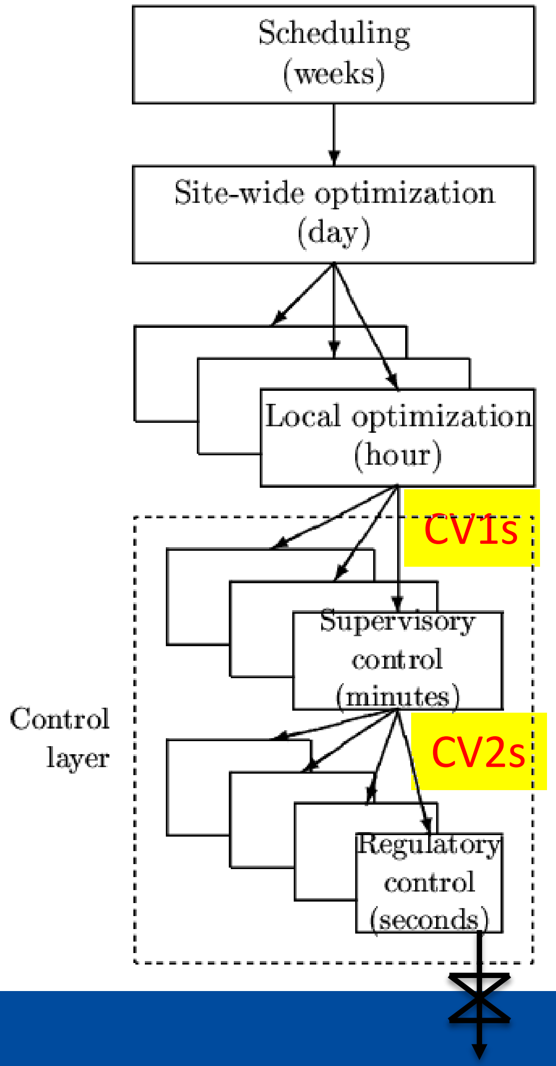
Main notation
Control: Given setpoints



This talk: Focus on feedback

MV = Manipulated Variable
CV = Controlled Variable

Process control: Hierarchical decision system



min J (economics)
hour

Setpoint control
(+ look after other variables)
minute

Stabilize + avoid drift
second

Manager

Process engineer

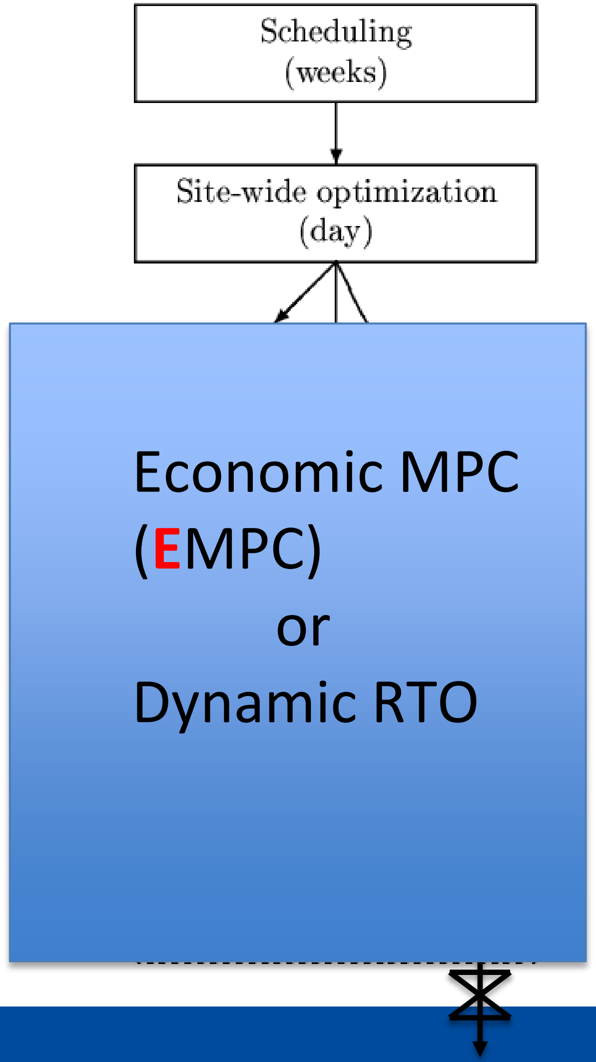
Operator/RTO (usually steady-state)

Supervisory control layer
"Advanced control"/MPC

PID-control

u (MV) = valves

Process control: Hierarchical decision system



Manager

Process engineer

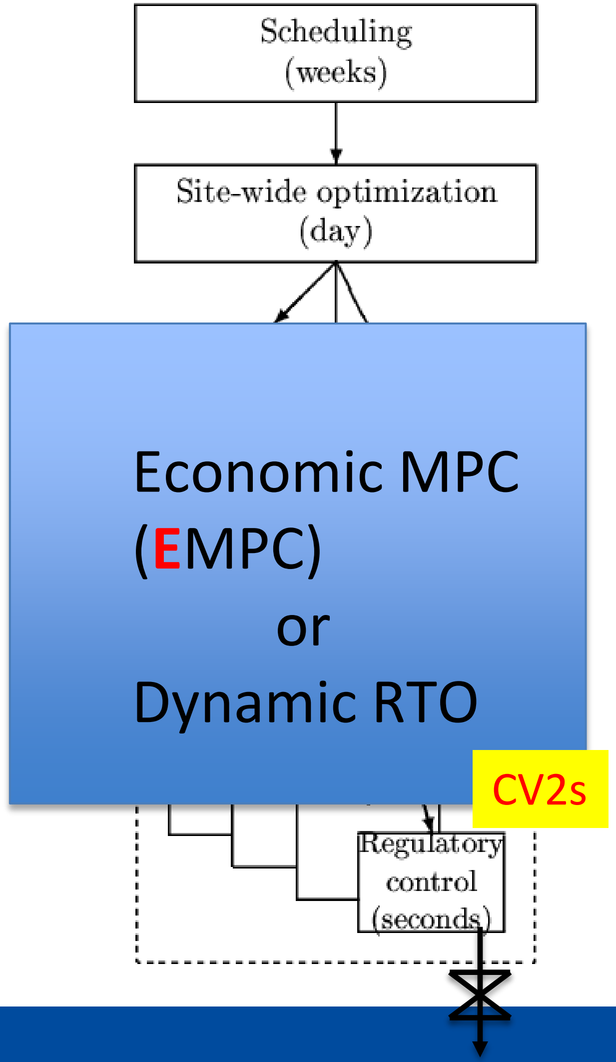
min J (economics)
hour

Setpoint control
(+ look after other variables)
minute

Stabilize + avoid drift
second

u (MV) = valves

Process control: Hierarchical decision system



Manager

Process engineer

min J (economics)

hour

Setpoint control

(+ look after other variables)

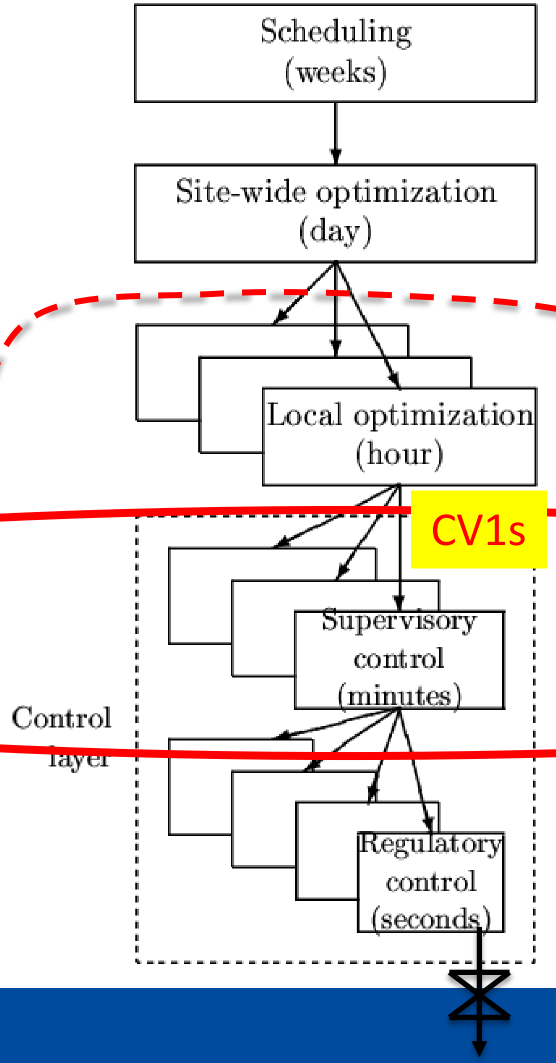
minute

Stabilize + avoid drift

second

u (MV) = valves

Process control: Hierarchical decision system



Manager

Process engineer

min J (economics)
hour

Operator/RTO (usually steady-state)

Focus of this talk

Setpoint control
(+ look after other variables)
minute

Supervisory control layer
"Advanced control"/MPC

Stabilize + avoid drift
second

PID-control

u = valves

How do we design the control system for a large process?

1. Consider economics

   $\min_u J(u,d)$ s.t. $g(u,d) \geq 0$

Translate into control: Find good controlled variables (CV1) to keep at fixed setpoints

CV1 = Active constraints (can change!)

CV1 = «Self-optimizing» variables (related to gradient $y = J_u = 0$)

2. Design control system to control CV1 under varying conditions (disturbances)

Usually two control layers:

- Supervisory control layer («advanced control»), CV1
- Stabilizing control layer («PID control»), CV2

«ADVANCED/SUPERVISORY CONTROL» LAYER

Objectives :

1. Control primary variables CV1 at setpoints computed by RTO
  - Tight control of active constraints is often desired for economic reasons.
- Feedforward from measured disturbances
  - If helpful
- Make use of extra inputs
- Make use of extra measurements

2. Switch control structures (CV1) depending on operating region
- Change in active constraints
- Identify "self-optimizing variables"

3. Keep an eye on stabilizing layer
- Avoid saturation in stabilizing layer, may require switching

Implementation:
- Alternative 1: Advanced control based on "simple elements" (decentralized control)
- Alternative 2: MPC

Academia: MPC

- MPC
  - General approch, but we need a dynamic model
  - MPC is usually implemented after some time of operation
  - Not all problems are easily formulated using MPC
  - Explicit control of changing active constraints requires additional logic
    - Industry uses two-stage MPC with steady-state feasibility based on constraint priority list

Research question: Alternative simpler solutions to MPC

- Would like: Feedback solutions that can be implemented without a detailed models

- Machine learning?
  - Requires a lot of data
  - Can only be implemented after the process has been in operation

- But we have "classical advanced control“ based on single-loop PIDs
  - Extensively used by industry
  - Problem for engineers: Lack of design methods
    - Has been around since 1940’s
    - But almost completely neglected by academic researchers
  - Main fundamental limitation: Based on single-loop (need to choose pairing)

“Classical Advanced control” using simple control elements

1. Cascade controllers
   - Have Extra output (state) measurements
2. Feedforward elements
   - Have measured disturbance
3. Decoupling elements
   - Have interactive process
4. Linearization elements / Adaptive gain
   - Have Nonlinear process
5. Split-range control (or multiple controllers or VPC)
   - Need extra inputs (MV) to handle all conditions (steady state) (MV-MV switch)
6. Valve position control (VPC) (Input resetting/Midranging control)
   - Have extra inputs dynamically
7. Selectors
   - Have changes in active constraints (CV-CV switch)

APC: Often static nonlinear «function block»
One unifying approach is «Transformed inputs» (similar to feedback linearization)

How design classical APC elements?

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

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

- Academia: Very little work

7. Change in CVs (active constraints) using selectors

Want tight control of constraints for economic reasons

$$\min_u J(u,d)$$
$$\text{s.t. } g(u,d) \geq 0$$

- Active constraint: $g_A=0$
- Tight control of g_A minimizes «back-off»

- How can we identify and control active constraints?

Example. Drive from a to B in shortest time

$J = T$

u = power (gas pedal)

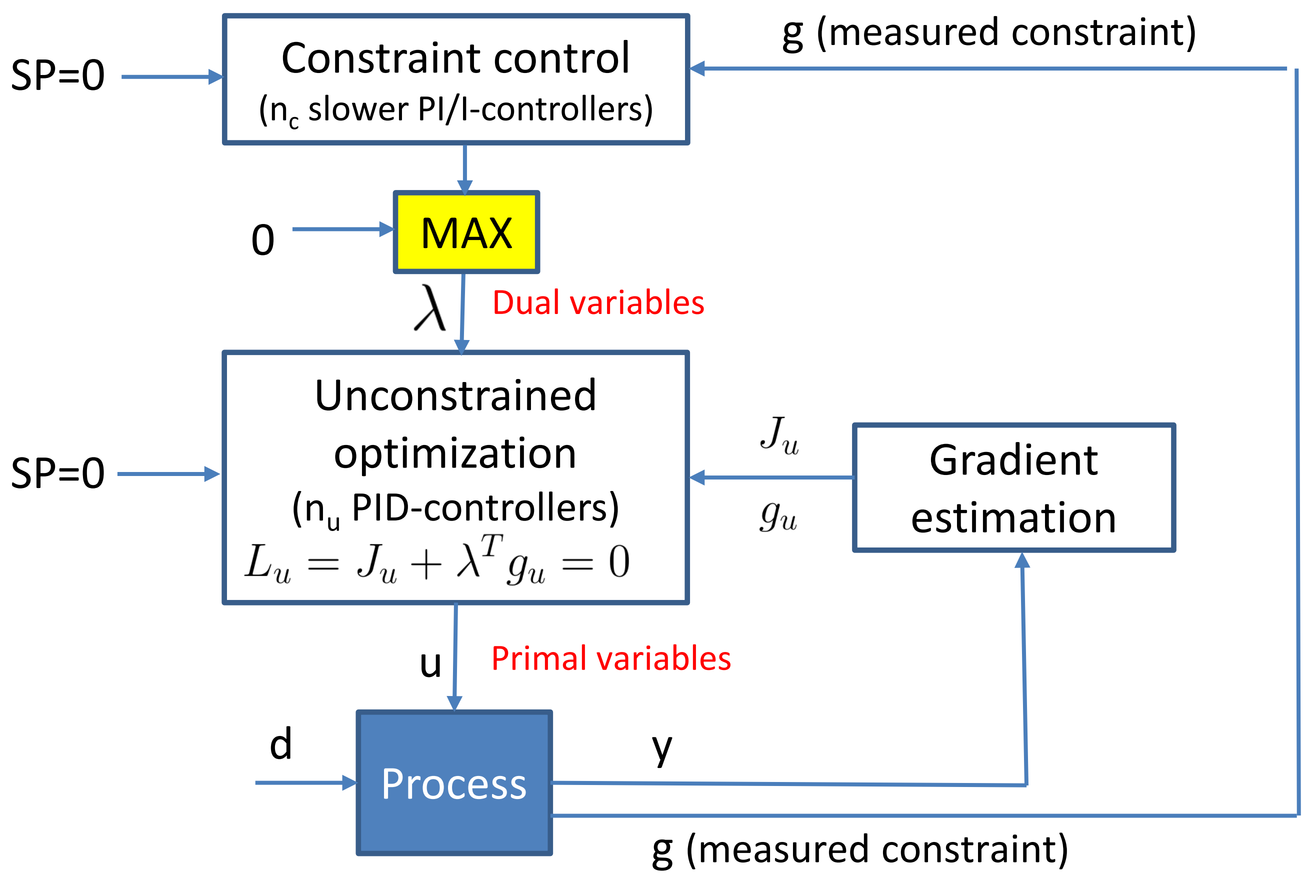
$g1$ = speed limit = $v_{max} - v$

$g2$ = safety limits

Straight road:

$g_A = g1$ (Speed limit)

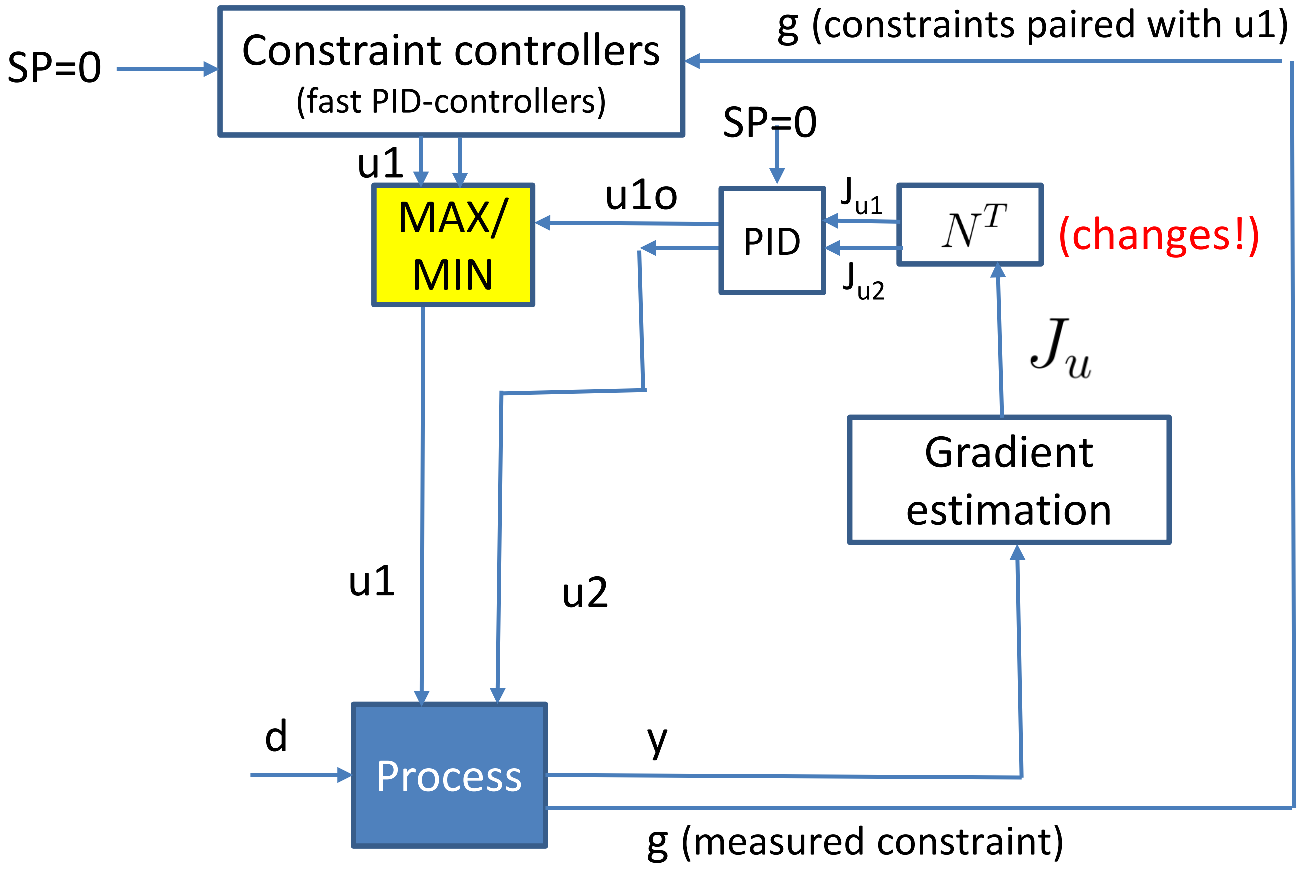
Feedback solution that automatically tracks active constraints by adjusting Lagrange multipliers (= shadow prices = dual variables) λ



Primal-dual feedback control.

- Makes use of «dual decomposition» of constrained optimization
- Selector on dual variables λ
- Problem: Constraint control using dual variables is on slow time scale

Alternative: Feedback solution with «direct» constraint control



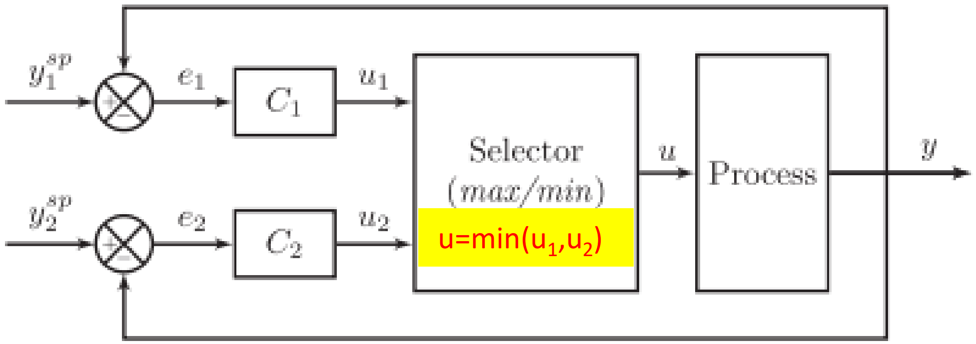
J_{u1}, J_{u2} = $N^T J_u$ reduced gradients
(«self-optimizing variables»)

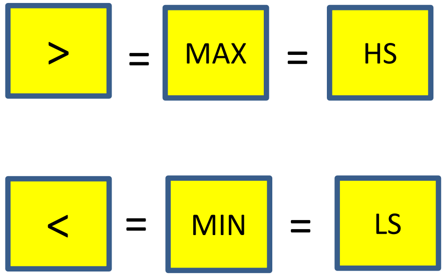
$$\boxed{\mathbf{N}^T \nabla_{\mathbf{u}} \mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{d})^T = 0}$$

- Selector on primal variables (inputs)
- Similar to selectors in APC

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

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





- Sometimes called "override"
- Selectors are used for output-output (CV-CV) switching
  - Example: Normally want to keep y1 at a setpoint,
    - but y2 (higher priority) must not exceed constraint.
    - With selector: When y2 reaches constraint, we give up control of y1.
    - More generally: $u_1=u_o$ is the optimal input value without constraint (can be given up)
  - Can have many constraints paired with same input y.
- Selectors work well, but require pairing each constraint with a given input (not always possible)

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 A. Maximize flow with pressure constraints

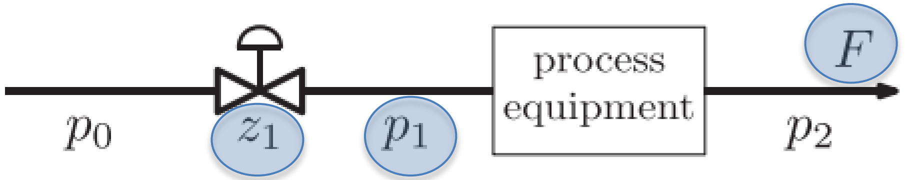


Fig. 6. Example 2: Flow through a pipe with one MV ($u = z_1$).

Input u = z_1
Want to maximize flow, J=-F:
Unconstrained: Optimal input is infinity:

$$u_0 = \infty$$

Optimization problem is:

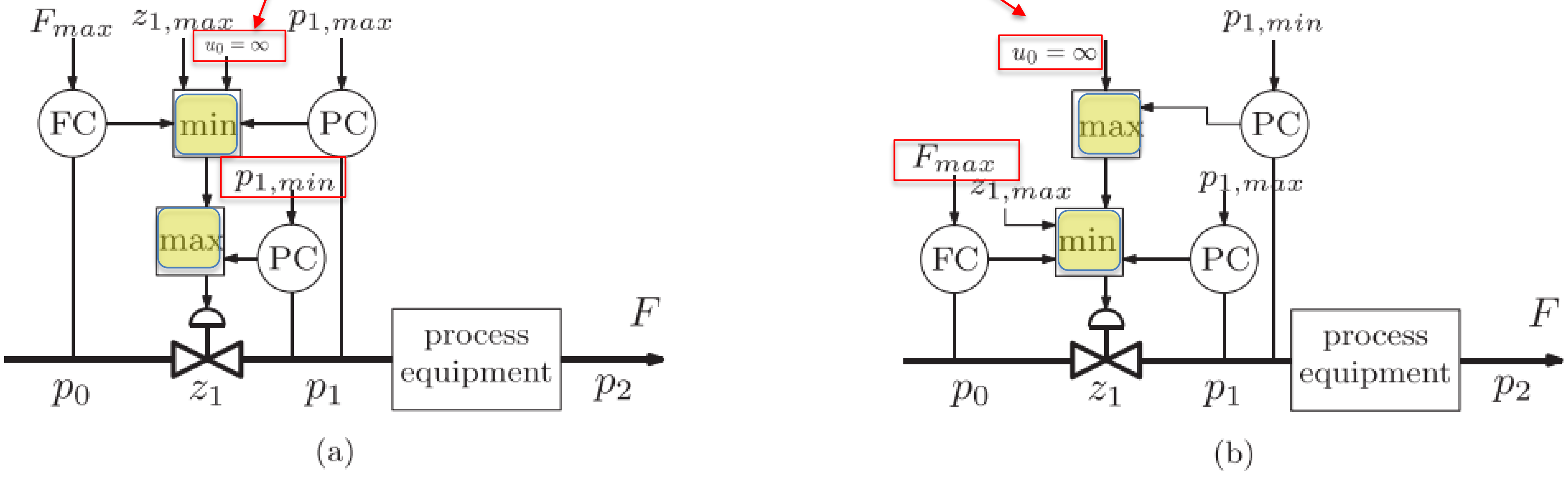
$$\max_{z_1} F$$

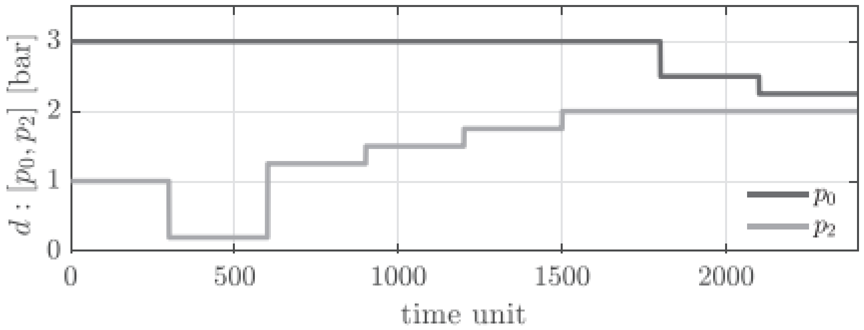
s.t.

$$\boxed{\begin{aligned} F &\leqslant F_{max} \\ p_1 &\leqslant p_{1,max} \\ p_1 &\geqslant p_{1,min} \\ z_1 &\leqslant z_{1,max} \end{aligned}} \tag{15}$$

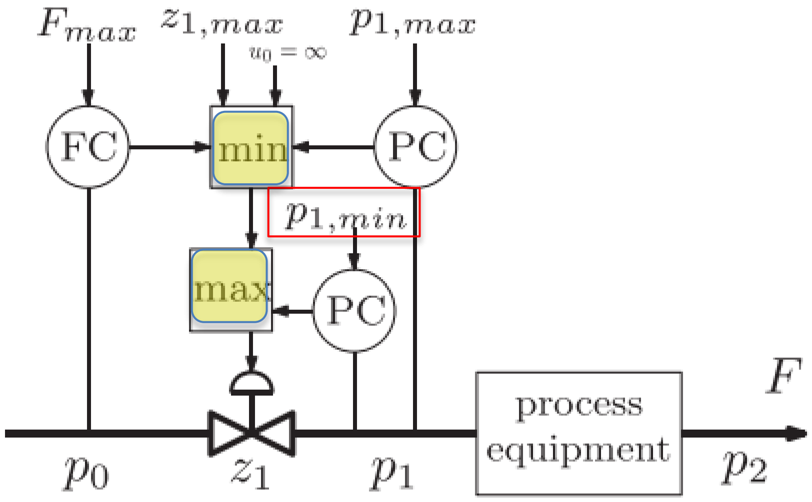
where $F_{max} = 10$ kg/s, $z_{1,max} = 1$, $p_{1,max} = 2.5$ bar, and $p_{1,min} = 1.5$ bar. Note that there are both max and min- constraints on p_1. De-

Desired input u_0 without constraints (can be given up) into first selector block

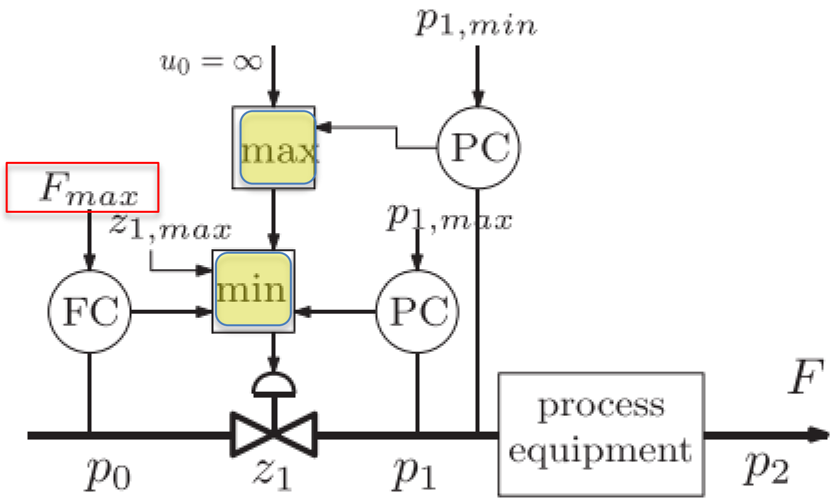




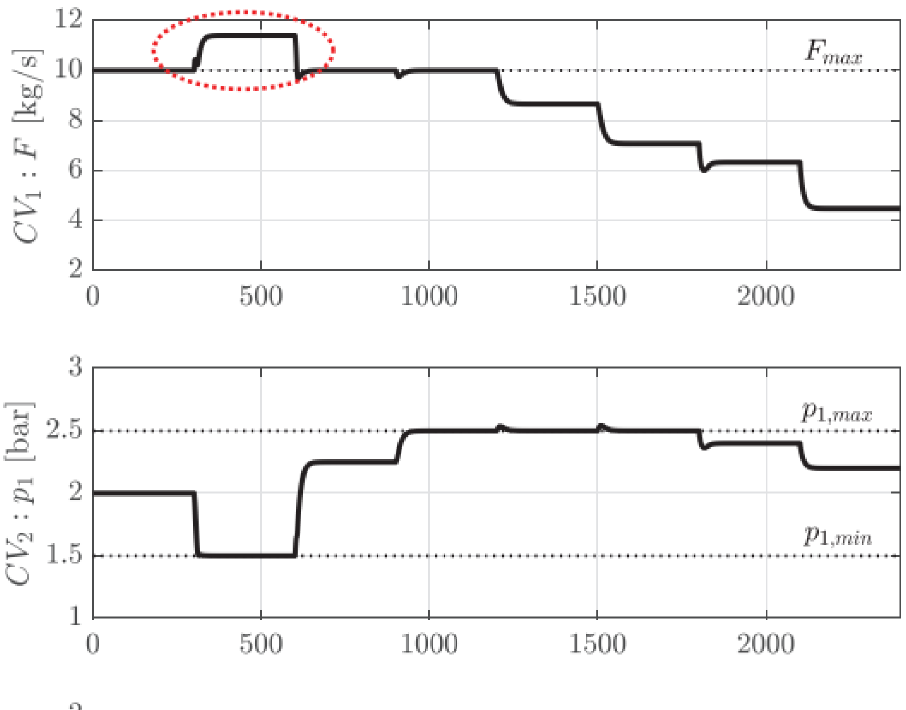
Disturbances in p_0 and p_2 (unmeasured)

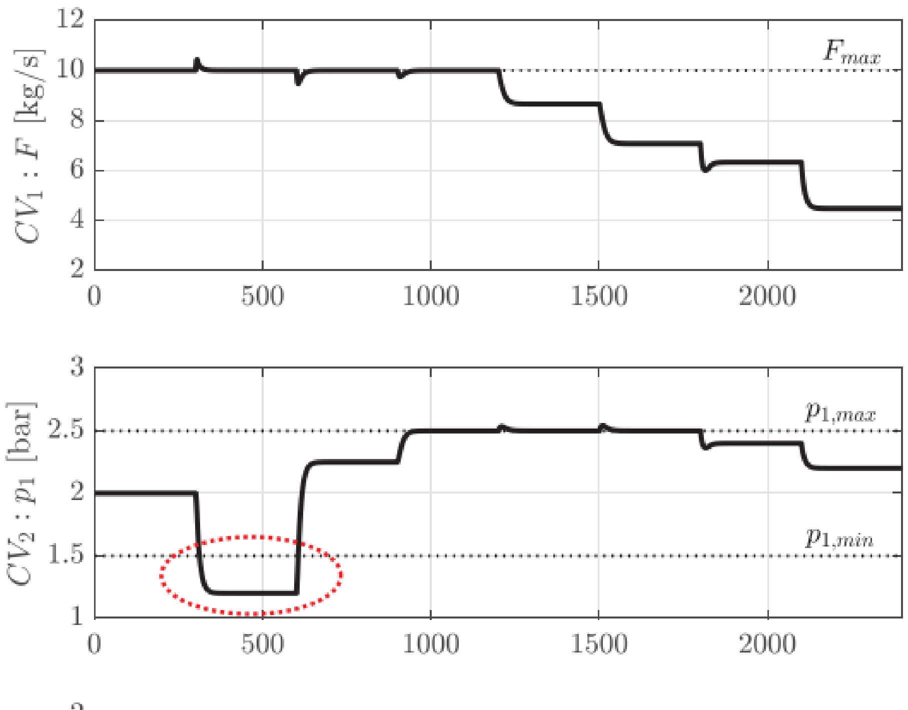


(a)



(b)

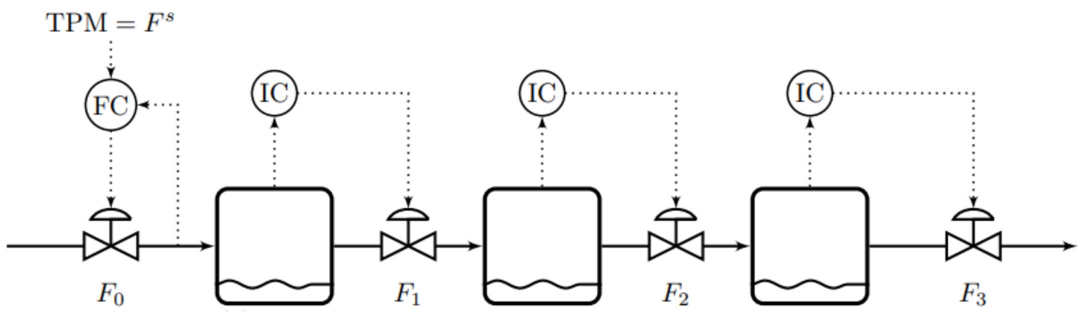




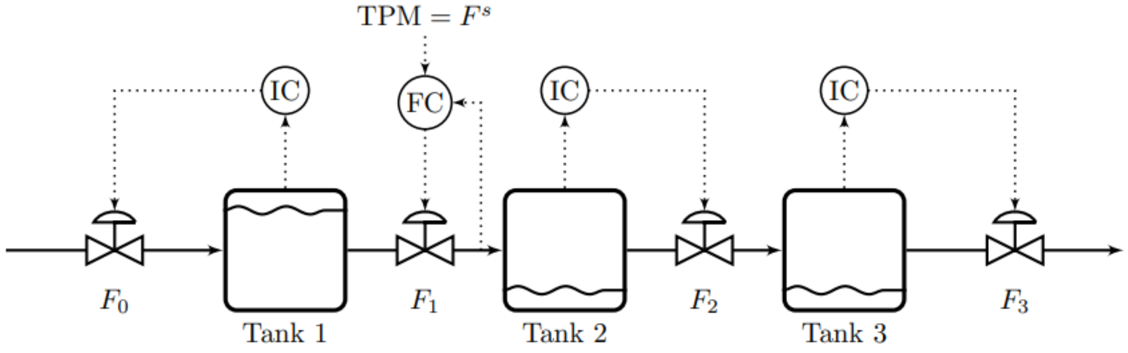
t>1800: u=zmax=1

Example B. Maximize production for serial process

- Constraints on levels (H,L)
- and valves/flows

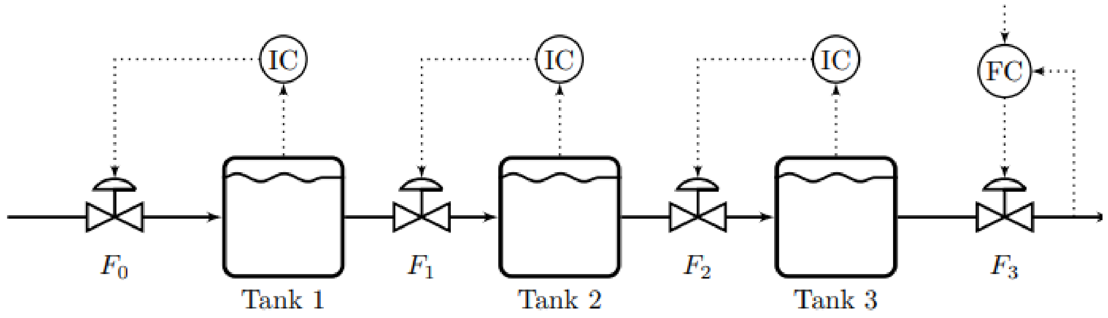


TPM = throughput manipulator
Typically at bottleneck («active constraint»)



(b) TPM at F_1. Inventory control radiating around the TPM.

Disturbances: Temporary bottelenecks (max-constraints) for F0, F1, F2 or F3

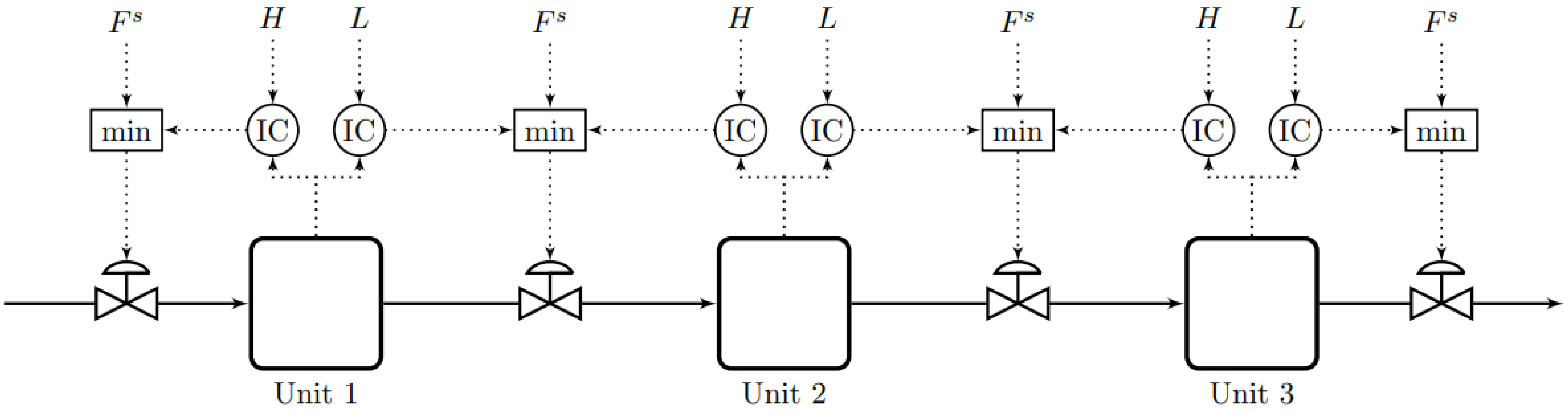


(d) TPM at F_3. Inventory control in direction opposite of flow.

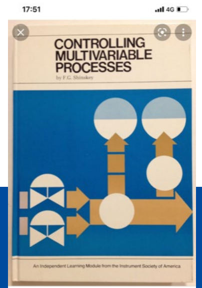
Example B. Very smart selector strategy: **Bidirectional inventory control**

Reconfigures automatically with optimal buffer management!!

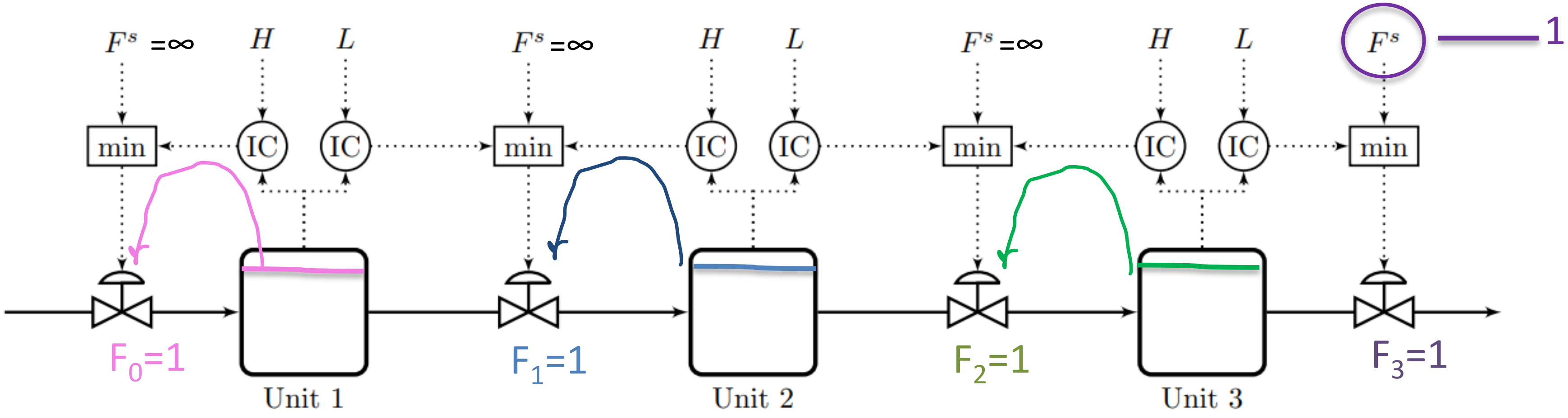
Max flow:
$F^s=\infty$



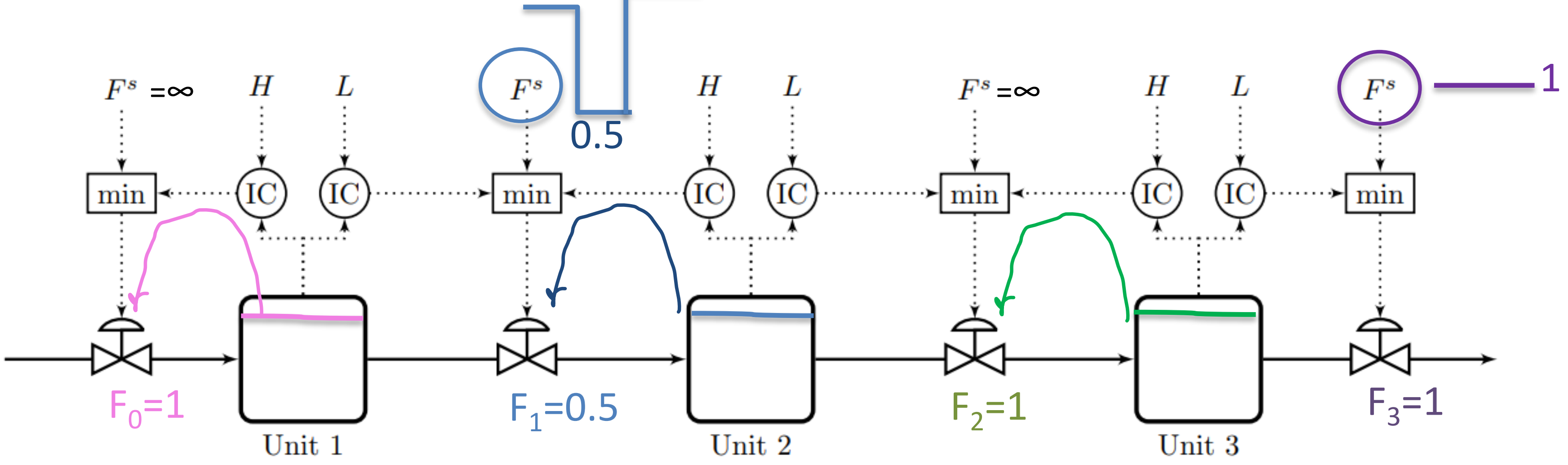
F.G. Shinskey, «Controlling multivariable processes», ISA, 1981
C. Zotica, S. Skogestad and K. Forsman, Comp. Chem. Eng, 2021



NTNU



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

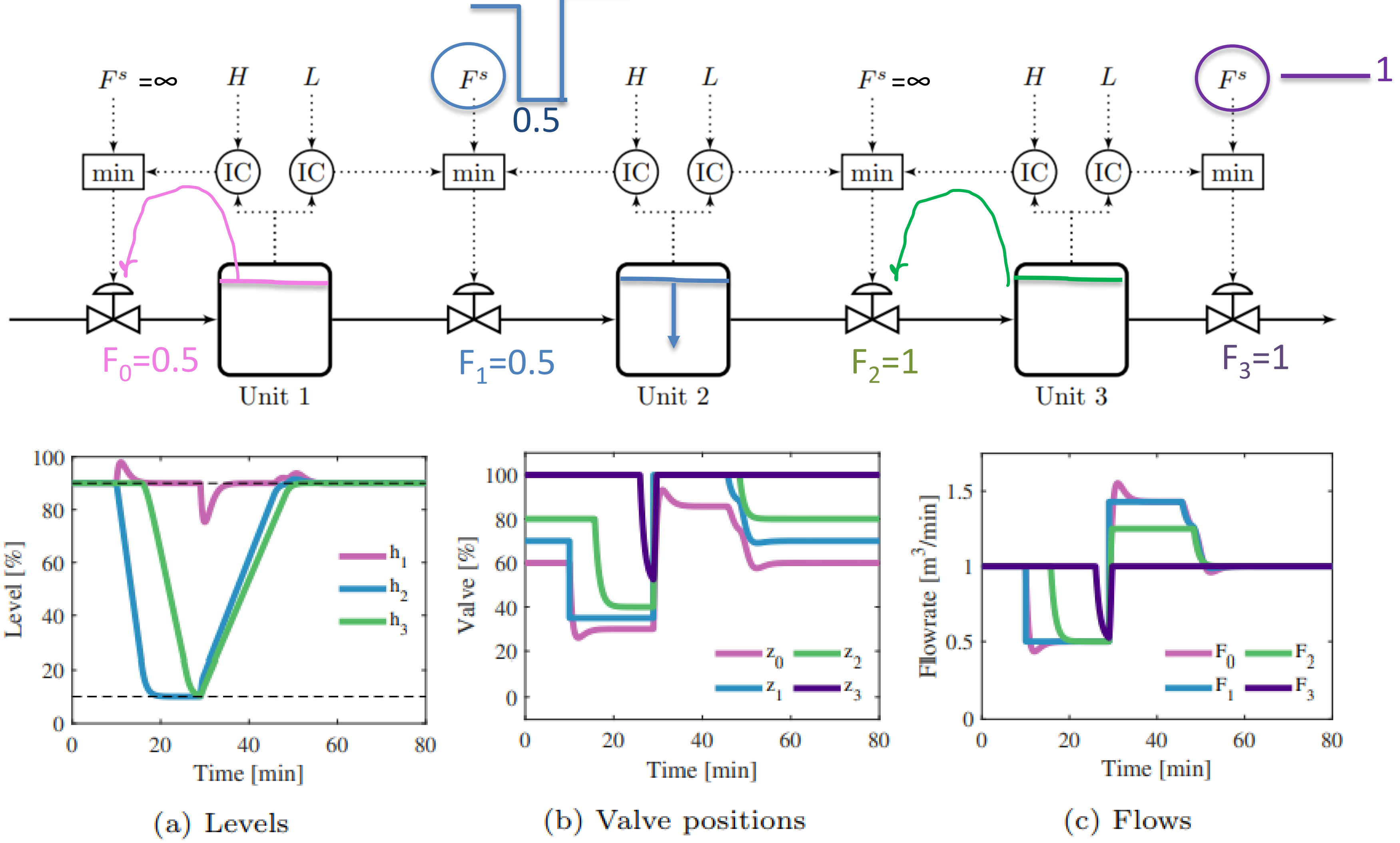


Figure 12: Simulation of a 19 min temporary bottleneck in flow F_1 for the control structures in Fig. 3d with the TPM downstream of the bottleneck.

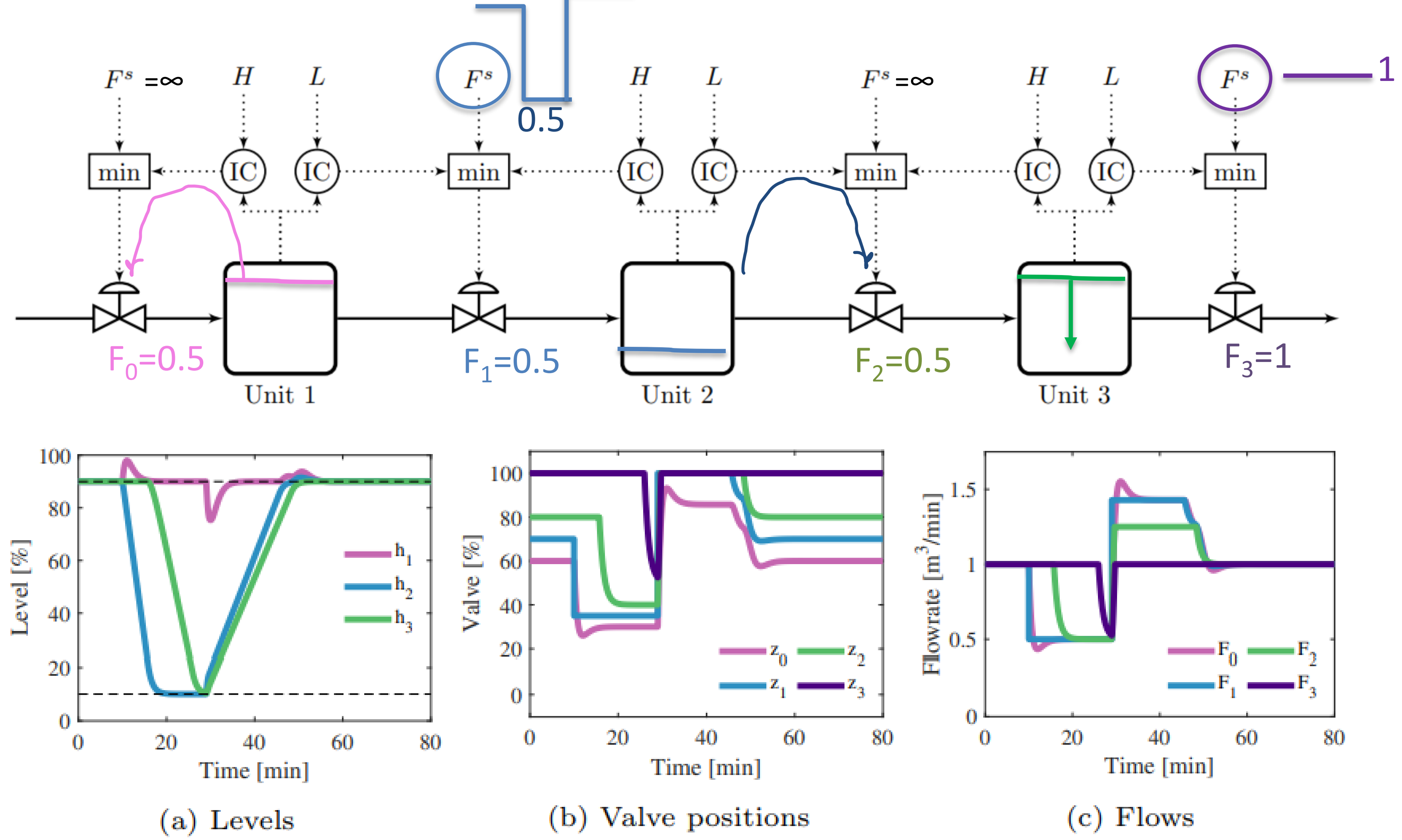


Figure 12: Simulation of a 19 min temporary bottleneck in flow F_1 for the control structures in Fig. 3d with the TPM downstream of the bottleneck.

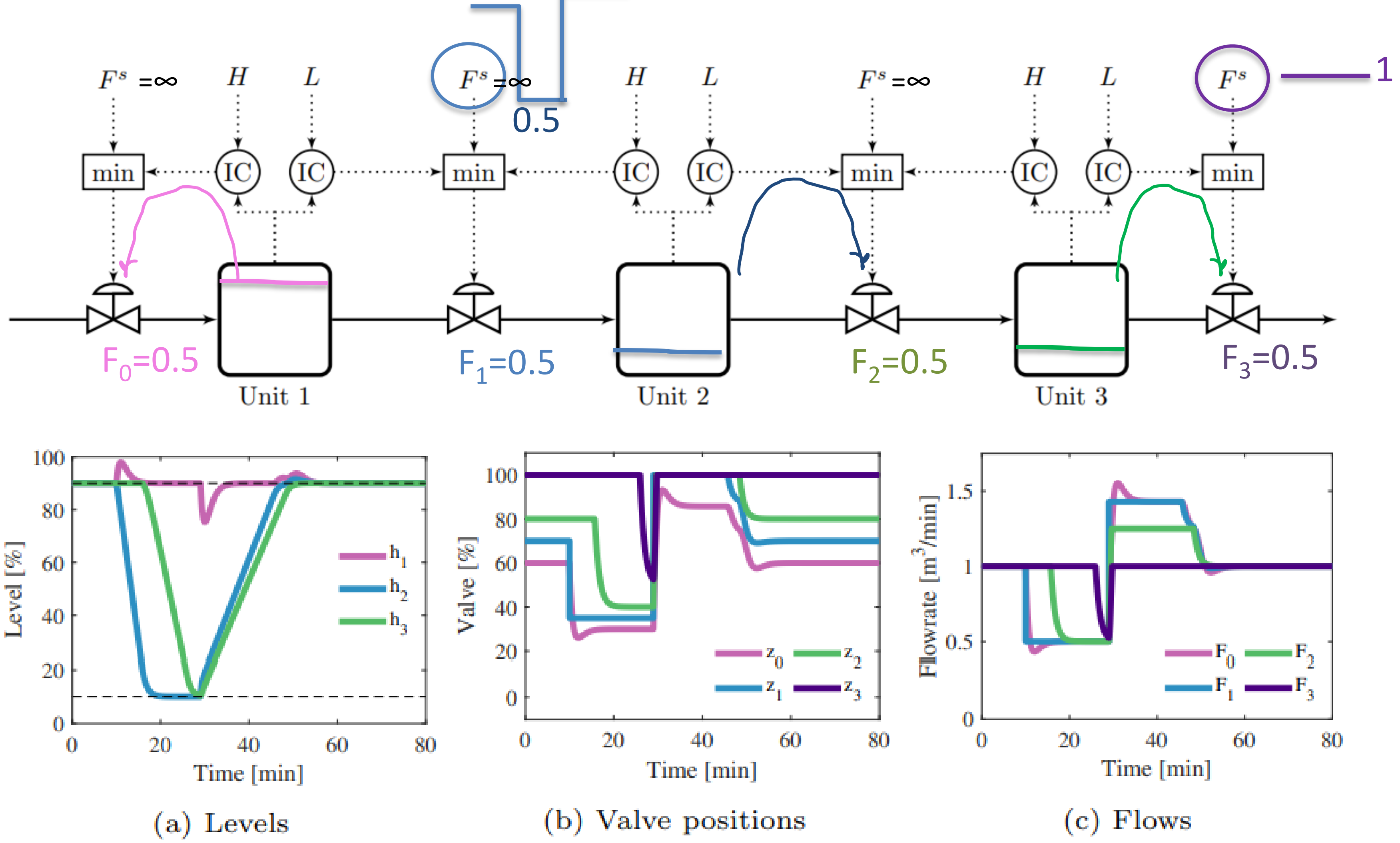
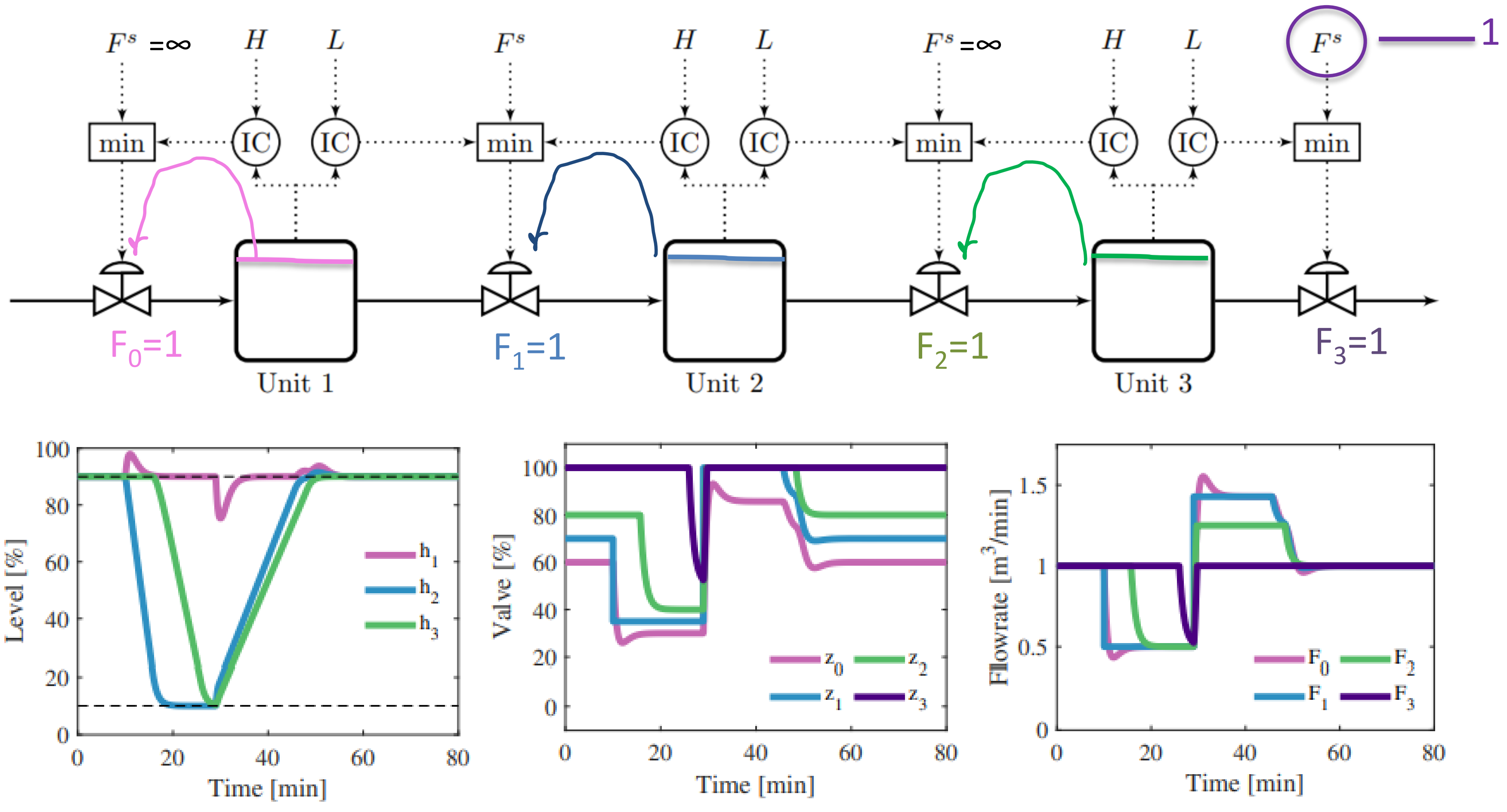


Figure 12: Simulation of a 19 min temporary bottleneck in flow F_1 for the control structures in Fig. 3d with the TPM downstream of the bottleneck.



(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? I doubt it. We tried.

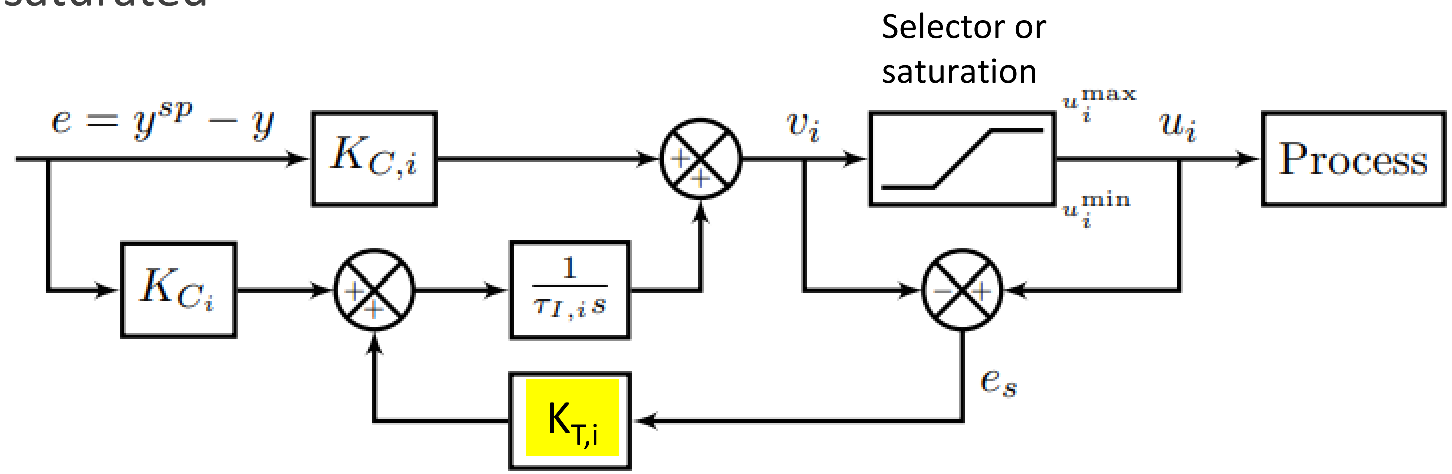
NTNU

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 (Examples A and B).

Anti-windup

- All the controllers shown need anti-windup to «stop integration» during periods when the control action (v_i) is not affecting the process:
  - Controller is disconnected (because of selector)
  - Physical MV u_i is saturated



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

Challenges selector design

- Standard approach requires pairing of each active constraint with a single input
  - May not be possible in complex cases
- Stability analysis of switched systems is still an open problem
  - Undesired switching may be avoided in many ways:
    - Filtering of measurement
    - Tuning of anti-windup scheme
    - Minimum time between switching
    - Minimum input change

"Classical Advanced control" using simple control elements

1. Cascade controllers
- Have Extra output (state) measurements (similar to state feedback)

2. Feedforward elements
- Have measured disturbance

3. Decoupling elements
- Have interactive process

4. Linearization elements / Adaptive gain
- Have Nonlinear process

APC: Often static nonlinear «function block»
One unifying approach is «Transformed inputs» (similar to feedback linearization)

5. Split-range control (or multiple controllers or VPC)
- Need extra inputs (MV) to handle all conditions (steady state) (MV-MV switch)

6. Valve position control (VPC) (Input resetting/Midranging control)
- Have extra inputs dynamically

7. Selectors
- Have changes in active constraints (CV-CV switch)

1. Cascade control

- Use extra measurement y_2 to improve control of primary output y_1
- Cascade: MV for outer primary controller is the setpoint r2 to inner secondary controller
- Fast inner loop: Eliminate d2 and nonlinearity in G2
  - Proof: Ideally y2 = r2

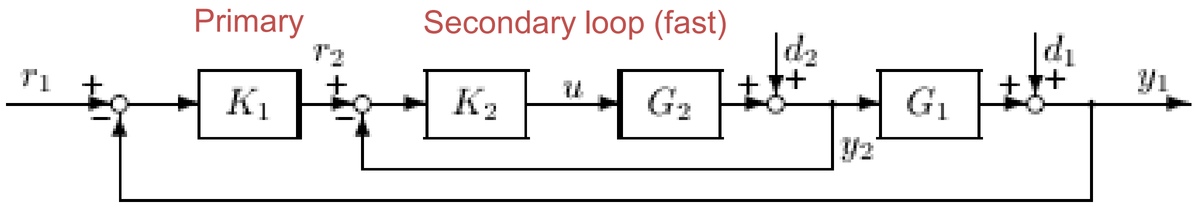


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

Design: First design fast inner/secondary controller (can be just P-control)
Then design slower outer/primary controller (PI-controller)
Closed-loop time constant for outer loop at least 4 times larger (slower)

An alternative approach that uses extra measurements to improve control is «**Full state feedback».**

NTNU

2,3,4. Feedforward, Decoupling, Linearization

- Unified approach (for static case): «Transformed» inputs v (and outputs)*

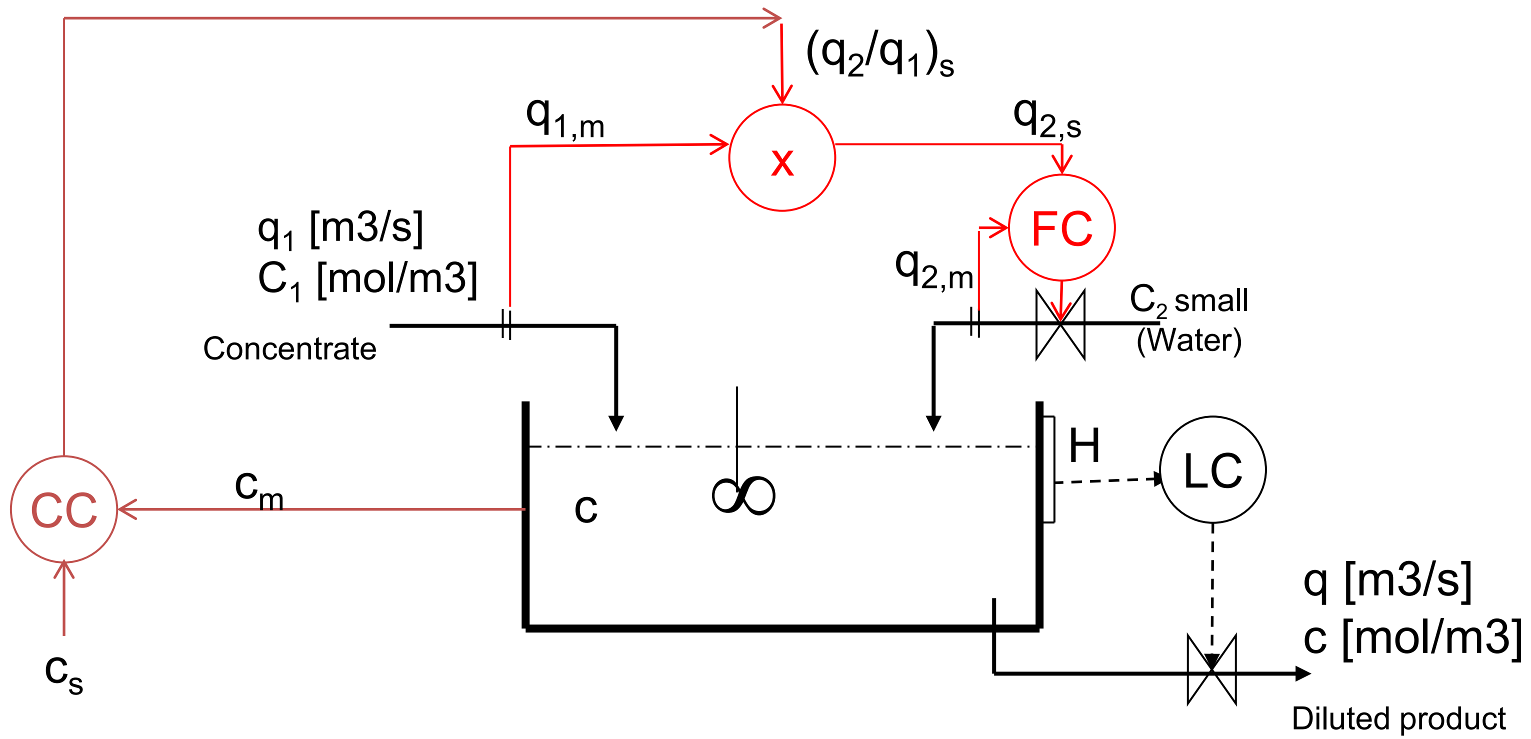
- Greg Shinskey (1981): “There is no need to be limited to single measurable or manipulable variables. If a more meaningful variable happens to be a mathematical combination of two or more measurable or manipulable variables, there is no reason why it cannot be used.”

- Motivation: ratio control

*«Transformed Manipulated Variables for Linearization, Decoupling and Perfect Disturbance Rejection», C Zotica, N Alsop, S Skogestad, IFAC World Congress, 2020

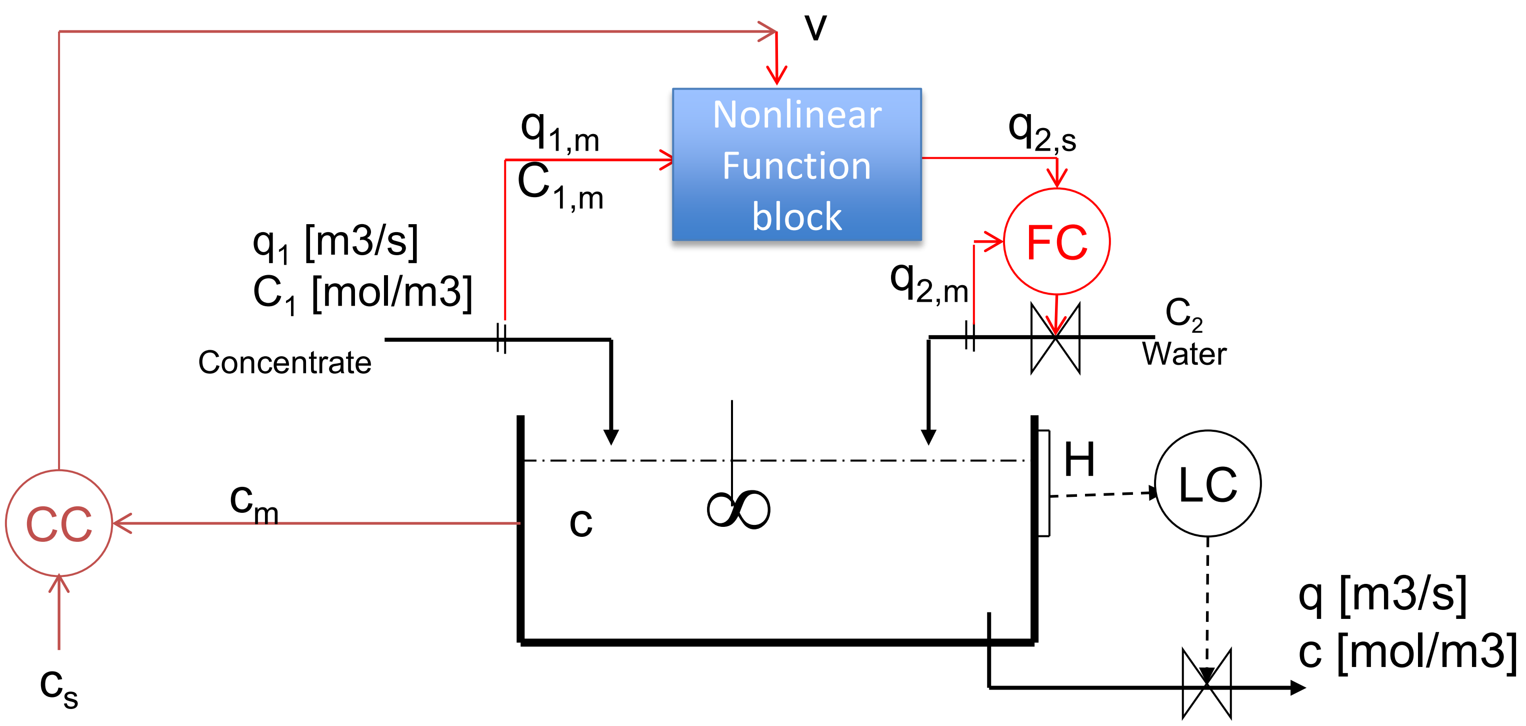
MOTIVATING EXAMPLE 1: MIXING PROCESS

RATIO CONTROL with outer cascade (to adjust ratio setpoint)



- Input u=q2.
- Transformed input v = q2/q1
- Gives “perfect” feedforward control for d=q1
- Potential problem for **outer** feedback loop (composition controller, y=c):
  - Gain from MV = $(q2/q1)_s$ to CV=c will vary because of multiplication with q1,m
  - Outer loop must handle disturbances in c1 and c2

Improved RATIO CONTROL = Ideal TRANSFORMED INPUT from static model



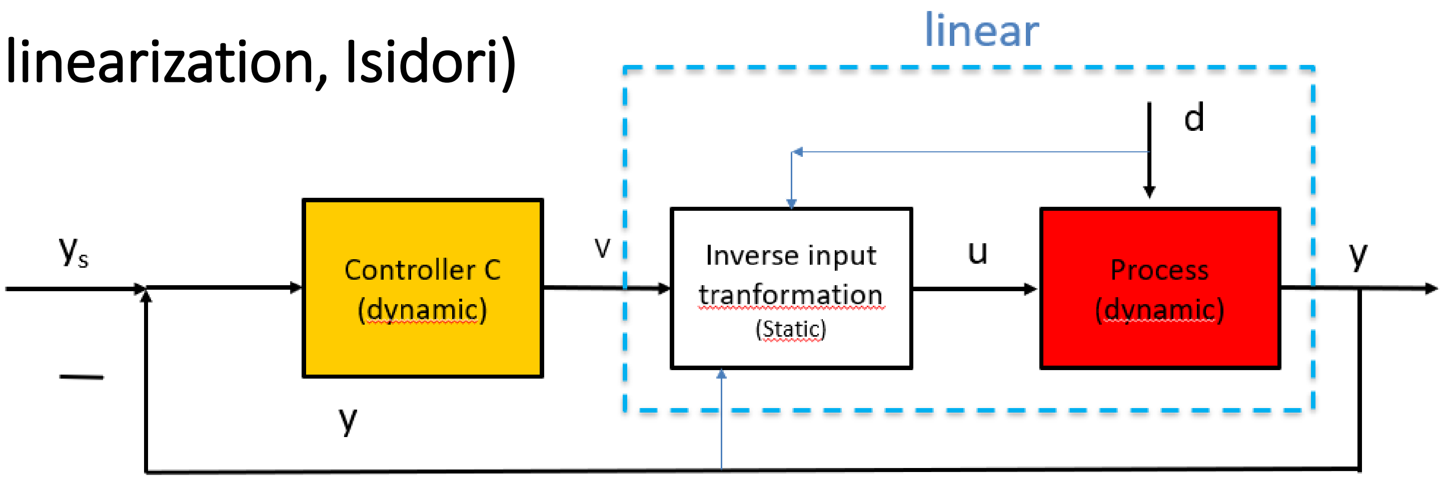
- Transformed input v = Right-Hand-Side of static model for y=c:

  $c = \underbrace{(q1\ c1 + q2\ c2)/(q1+q2)}_{=v}$

  Solve with respect to input q2.

  Get **"nonlinear function block":** **q2 = q1*(c1-v)/(v –c2)**
- Gives "perfect" feedforward control for disturbances in q1, c1 and c2.
- and also gives linear response (y=v) for controller CC

Dynamic case (~ feedback linearization, Isidori)



- Nonlinear dynamic system (process)

  $$\frac{dy}{dt} = f(y, u, d)$$

- Introduce transformed inputs

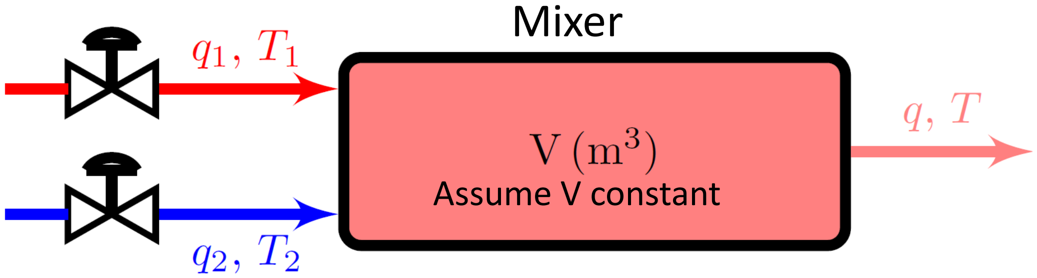
  $$v = f(y, u, d) - Ay$$

- New transformed system is linear, first-order, decoupled if A is diagonal and independent of disturbances!

  $$\frac{dy}{dt} = Ay + v$$

- Static input calculation **(inverse input transformation):** u is solution to: $f(y, u, d) - Ay = v$
- v can be set by conventional linear controller C (PI)
  - Ideally: Don't need to change v, but in practice need C to handle uncertainty
- **Assumptions**
  - Measure all disturbances (d)
  - **Low-order model with no. states (y) = no. inputs (u)**
  - The solution to the **static inverse problem** exists and satisfies certain properties.

Example 2. Mix hot (1) and cold (2) water, y=[q T]



Mass balance: $q = q_1 + q_2$ (static equation for y_1=q)

$$v_0 = q_1 + q_2$$

Energy balance: $\frac{dT}{dt} = \frac{q_1}{V}(T_1 - T) + \frac{q_2}{V}(T_2 - T)$ (dynamic equation for dy_2/dt)

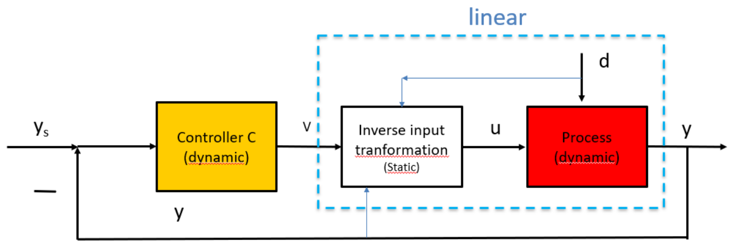
$$v = \frac{q_1}{V}(T_1 - T) + \frac{q_2}{V}(T_2 - T) - AT$$

New transformed inputs: v_0 and v.

Inverse transformation (with u_1=q_1 and u_2=q_2):

$$q_1 = \frac{V(v+AT)}{T_1 - T_2} - v_0(T_2 - T)$$

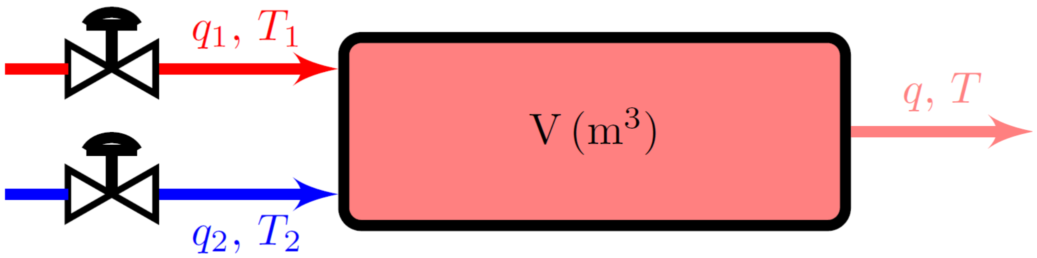
$$q_2 = v_0 - q_1$$



Tuning parameter, A = -(q/V)* (nominal)

Resulting Transformed system

y_1=q, y_2=T

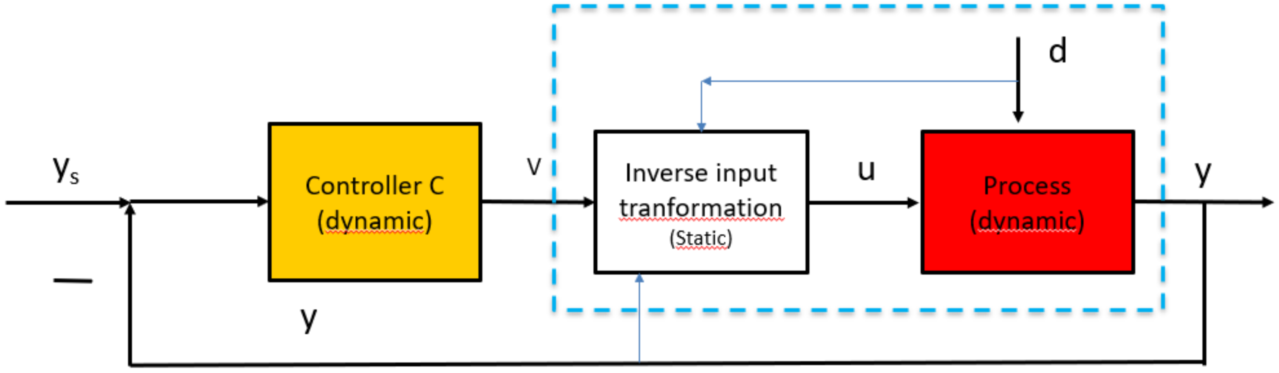


$$y_1 = v_0$$

$$\frac{dy_2}{dt} = Ay_2 + v$$

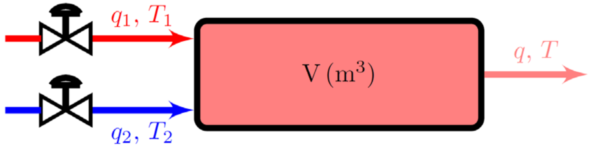
Transformed system

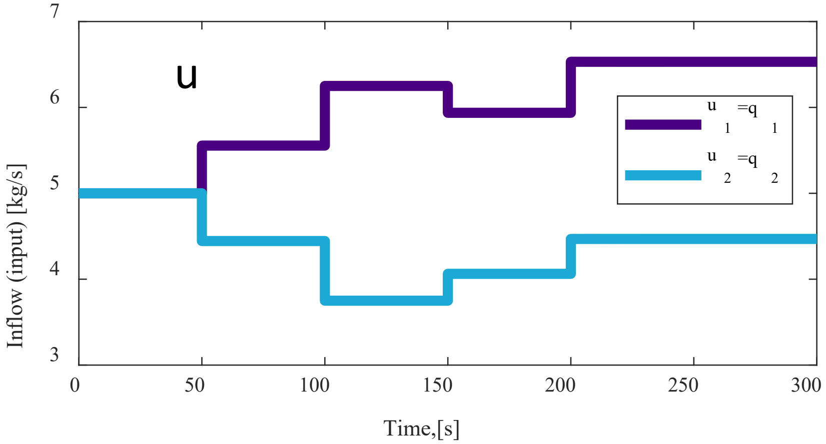
Linear, decoupled and independent of disturbances

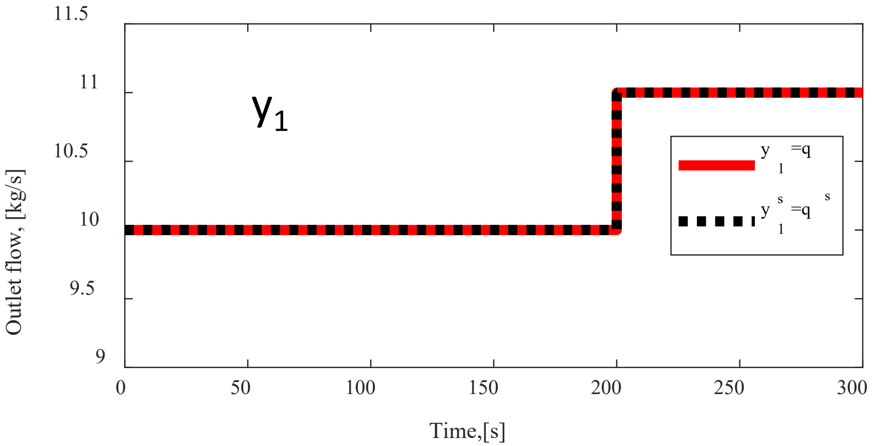


Simulation responses with transformation only.

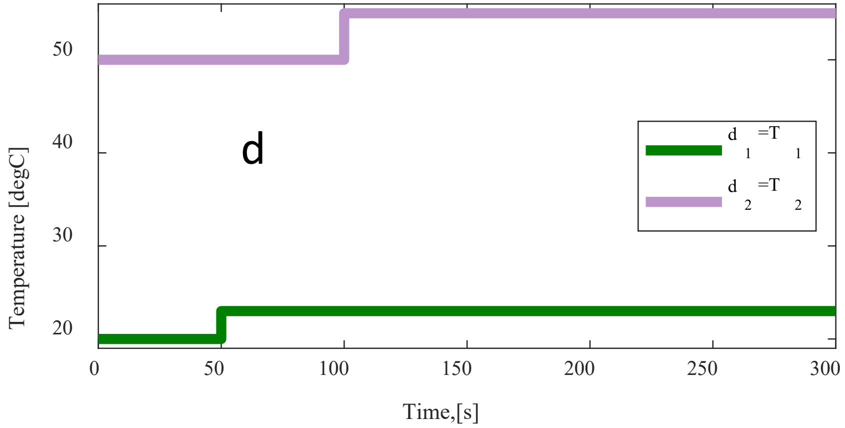
-> Perfect disturbance rejection and decoupling

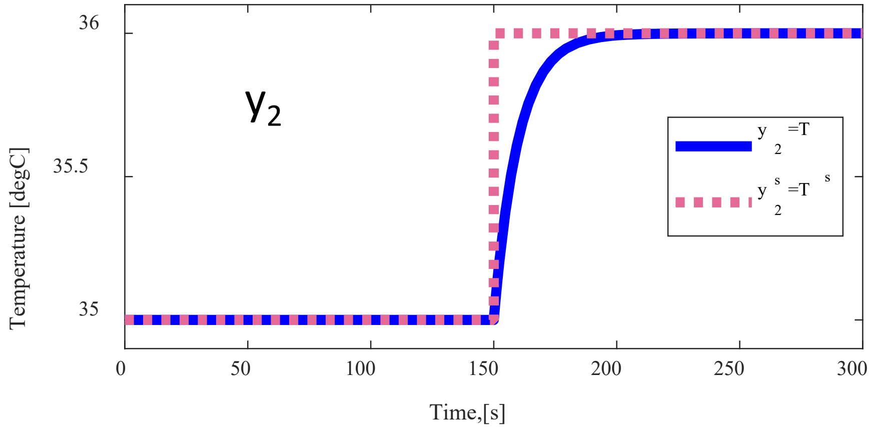






1. d_1=T_1: 20 ->22 °C at t = 50 s
2. d_2=T_2: 50 ->55 °C at t = 100 s
3. y_{2s}=T_s: 35 ->36 °C at t = 150 s
4. y_{1s}=q_s: 10 ->11 kg/s at t = 200 s





1,2: v_0 and v constant
3: Step in v
4: step in v_0

Use of extra MV (inputs): One CV, many MVs

Two different cases:

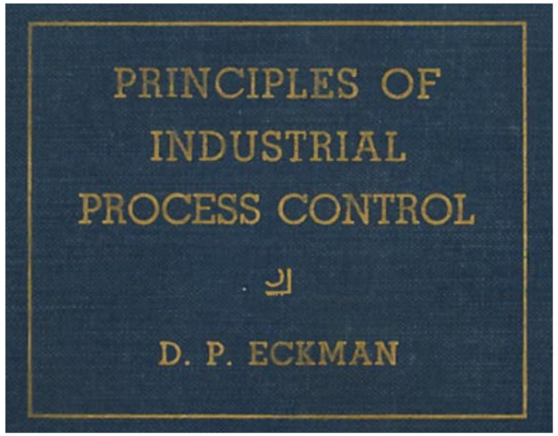
5. MV-MV switching: Need several MVs to cover whole range at <u>steady state</u>
   - We want to use one MV at a time

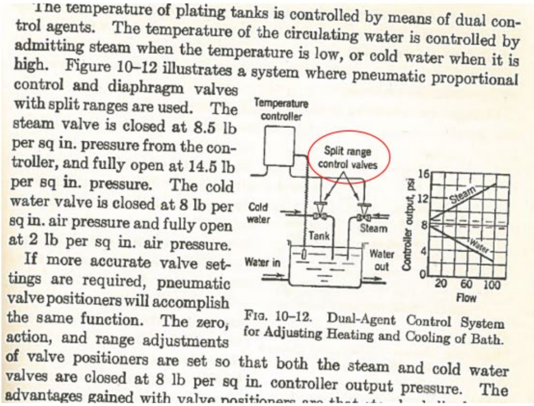
6. Have extra <u>dynamic</u> MV
   - Both MVs are used all the time

5. Extra inputs (MV) to handle all conditions (steady state)

1. Split-range control

Eckman, D.P. (1945). Principles of industrial control, pp.204-207. John Wiley & Sons, New York.





The temperature of plating tanks is controlled by means of dual control agents. The temperature of the circulating water is controlled by admitting steam when the temperature is low, or cold water when it is high. Figure 10–12 illustrates a system where pneumatic proportional control and diaphragm valves with split ranges are used. The steam valve is closed at 8.5 lb per sq in. pressure from the controller, and fully open at 14.5 lb per sq in. pressure. The cold water valve is closed at 8 lb per sq in. air pressure and fully open at 2 lb per sq in. air pressure.

If more accurate valve settings are required, pneumatic valve positioners will accomplish the same function. The zero, action, and range adjustments of valve positioners are set so that both the steam and cold water valves are closed at 8 lb per sq in. controller output pressure. The advantages gained with valve positioners are that ...

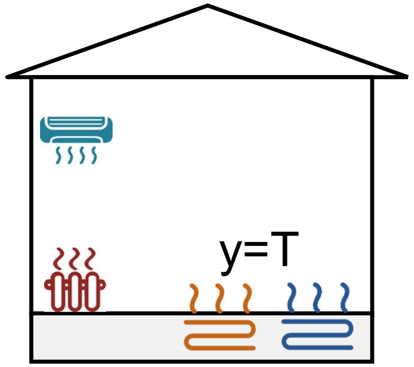
FIG. 10–12. Dual-Agent Control System for Adjusting Heating and Cooling of Bath.

Alternatives to Split-range control*:

2. Multiple controllers with different setpoints
3. Valve position control

* A. Reyes-Lua and S. Skogestad, «Systematic design of active constraint switching using classical advanced control structures», Ind.Eng.Chem.Res, Vol. 59, 2229-2241 (2020)

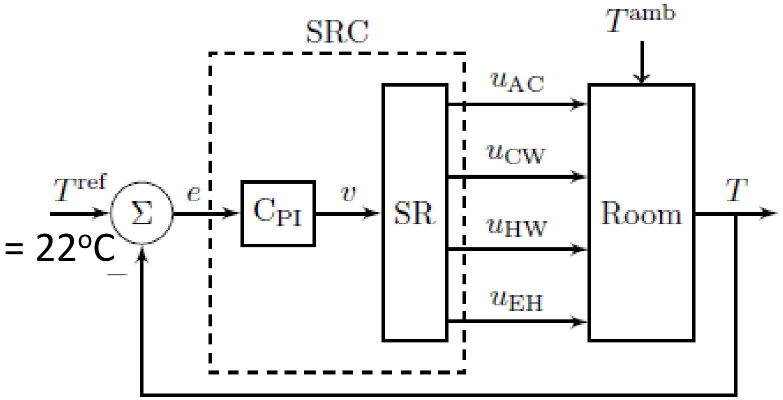
Example: Room heating with one CV (T) and 4 MVs

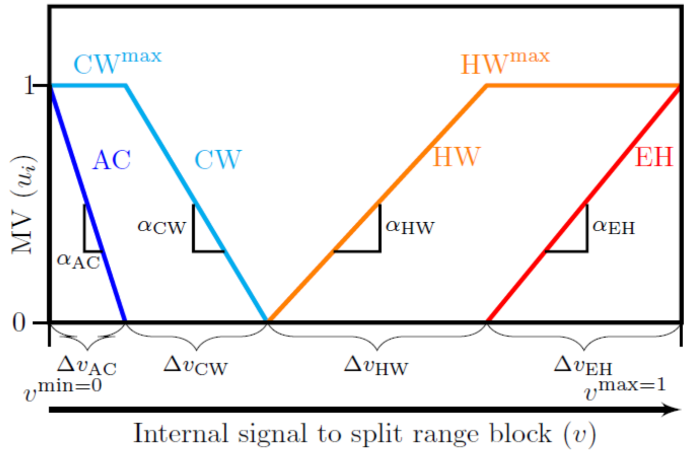


MVs (two for summer and two for winter):

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

Alt. 1 Split-range control (SRC).

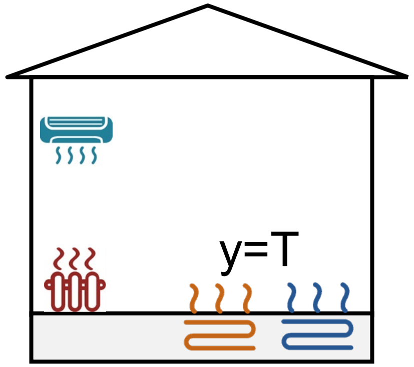




Note: may adjust the location of split (x-axis) to make loop gains equal.

Disadvantage SRC: 1. Must use same integral time for all MVs
2. Does not work well for cases where constraint values change

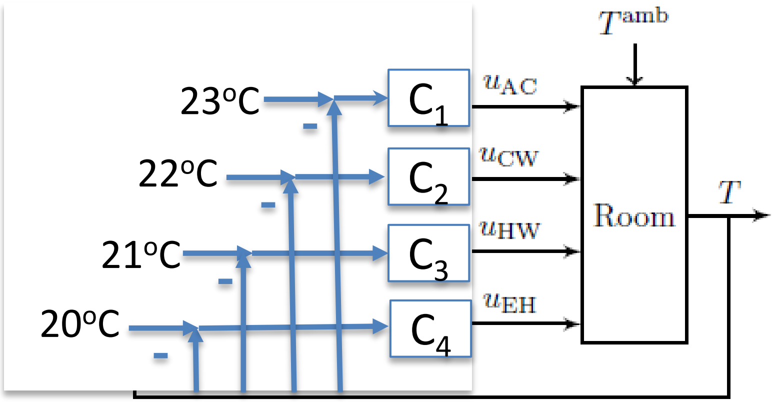
Example: Room heating with one CV (T) and 4 MVs



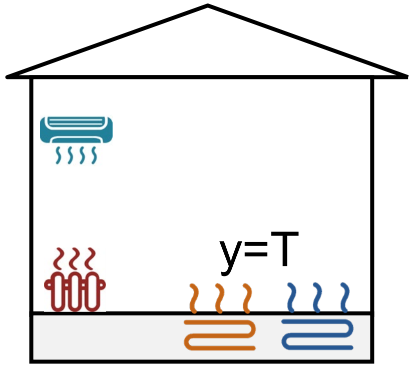
MVs (two for summer and two for winter):

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

Alt. 2. Multiple controllers with different setpoints



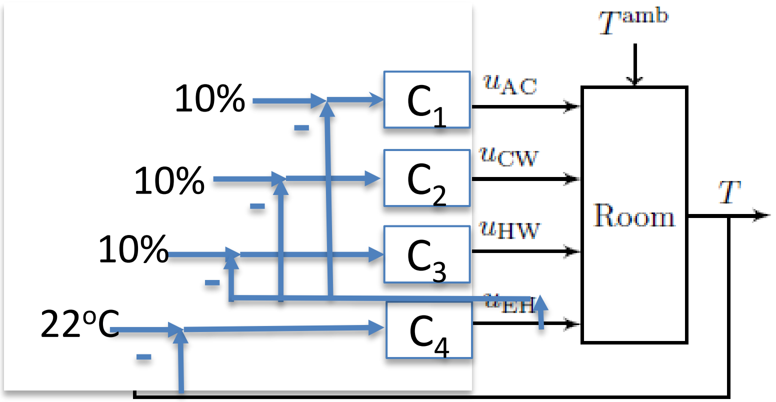
Example: Room heating with one CV (T) and 4 MVs



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)

Alt. 3. Input resetting (VPC)



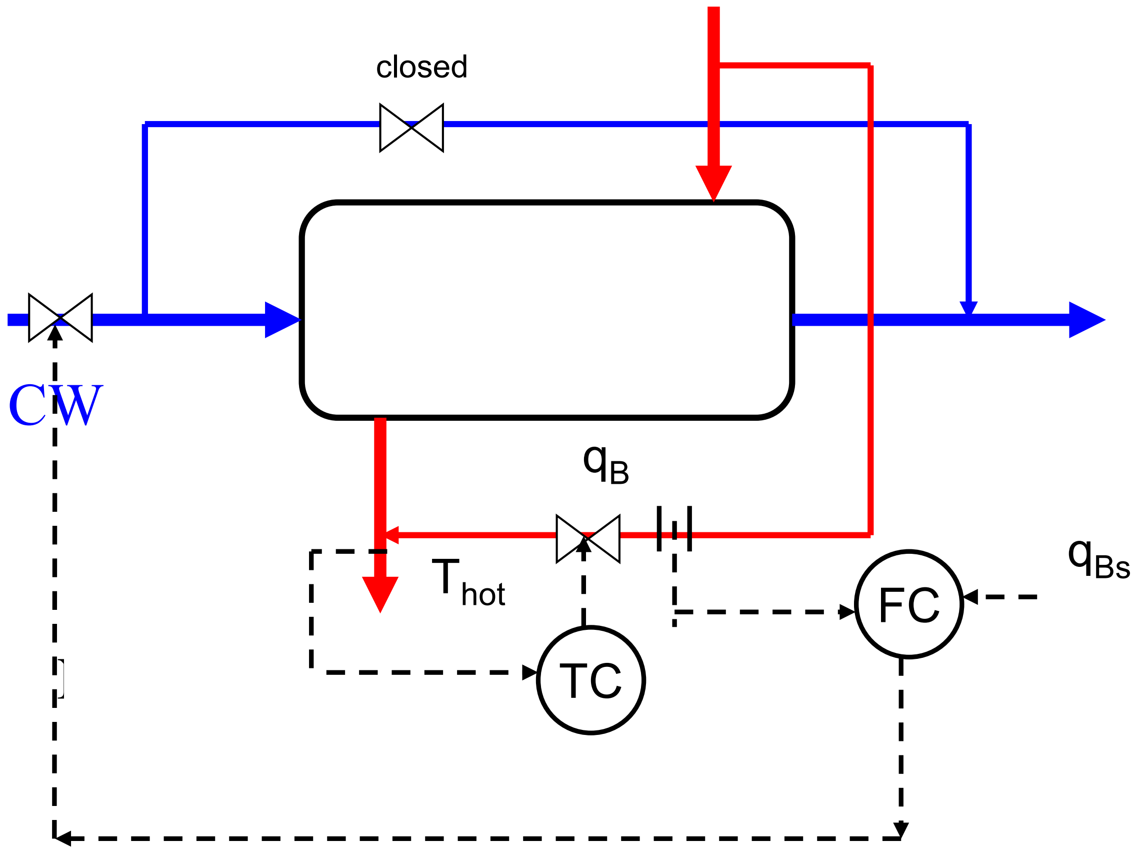
Summary MV-MV switching

- Use Alt.1 (split range control) for cases where the MV ranges (max and min values) are fixed.
  - Advantage: Easy to understand, because SR-block shows clearly sequence of MVs
- Use Alt. 2 (controllers for different setpoints) for cases where MV ranges vary
  - Advantage: Easier to implement than SRC and can have different controller tunings
- Use Alt. 3 (input resetting) for cases where CV (y) should always be controlled by same MV
  - But gives some economic loss

6. Extra inputs (MVs) dynamically: Valve position control (VPC)

- This VPC is active all the time with the aim «resetting» or «midranging» the input.

EXAMPLE. Use of bypass (extra input) for fast control



TC: Fast control of CV=T_{hot} using the “dynamic” MV2=q_B
FC: Resets MV2=q_B to its setpoint (q_{bs}) (e.g. 5%) using MV1=CW

When use MPC?

When conventional APC performs poorly or becomes complex

- Cases with many changing constraints (where we cannot assign one input to each constraint)
- Interactive process
- Know future disturbances and setpoint changes (predictive capability)

Conclusion Advanced process control

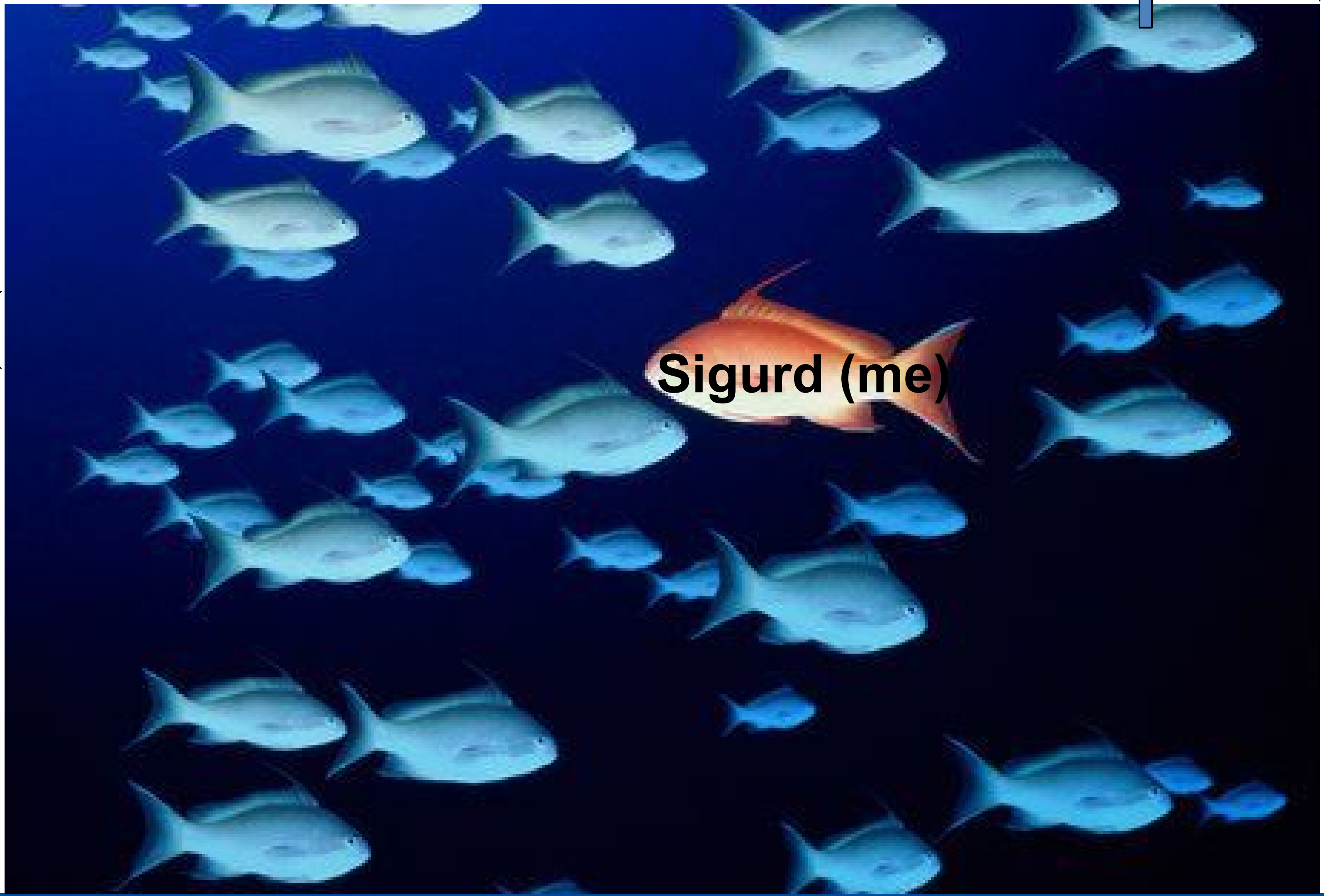
- Conventional APC works very well in may cases
  - Optimization by feedback (active constraint switching)
  - Need to pair input and output.
    - Advantage: The engineer can specify directly the solution
    - Problem: Unique pairing may not be possible for complex cases
  - Need model only for parts 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

Academic process control community fish pond

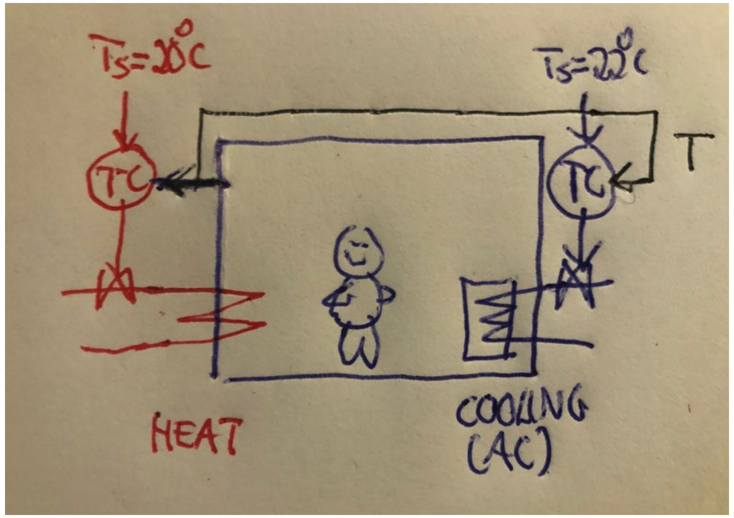
Optimal centralized Solution (**EMPC**)

Simple solutions that work (selector, PID++)



Extra

Alt. 2. Several controllers with different setpoints



Example

CV=y= room temperature

- MV1 = u1 = cooling
  CV^{spH} = 22°C (summer)
- MV2 = u2 = heating
  CV^{spL} = 20 - 2 = 20°C (winter)

Between 20°C and 22°C:
Temperature drifts with both inputs saturated:
- heating off, u1=0
- cooling off, u2=0

Need different setpoints to
- Avoid using cooling and heating at the same time
- avoid fighting between controllers

Note: The order of when to use each MV is determined by the setpoints. This also works if we have to two or more MVs with the same sign, for example, if we have two heating sources: hot water (as shown in figure) and more expensive electric heat. Then we could add a temperature controller for the electic heat with an even lower setpoint, for example, Ts=18C.

Alt. 3. Input (valve) position control (VPC)

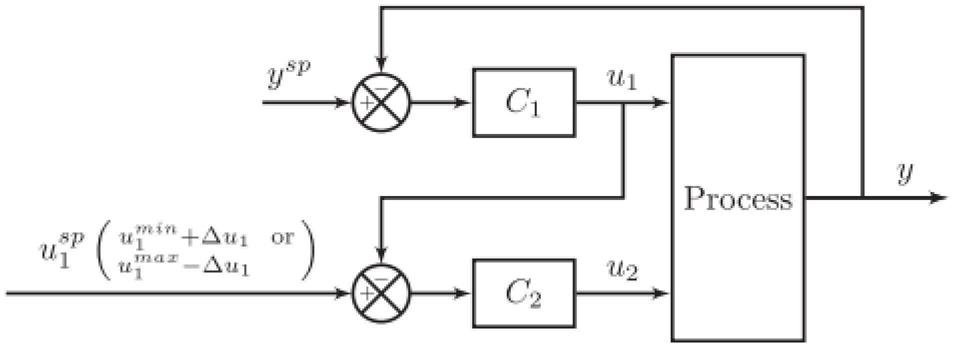


Figure 6. MV to MV constraint switching using input (valve) position control.

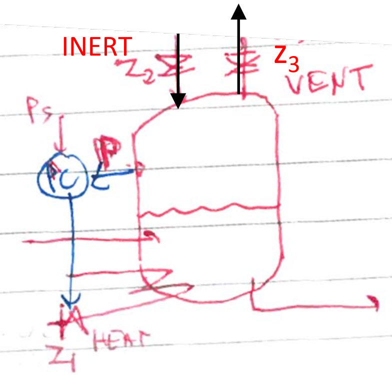
- Keep the original loop (u1-y)
- Use u2 to avoid saturation of u1 (VPC)

- Advantage: Always use u1 for control of y
- Disadvantages: Always use u1 (cannot put at constraint -> economic loss)

Important: Loop C2 with u2 is not active all the time, only when u1 is close to saturation

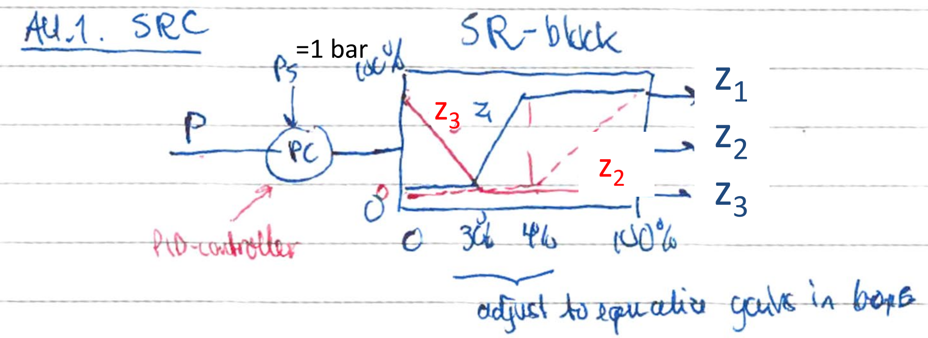
Example MV-MV switching: Pressure control

CV=p
MV1=heat (Q)
MV2=inert
MV3=vent



Normally: 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
Always use Q (z1) to control p.
Need two VPC's:

- Use vent (z3) to avoid Q=0
- Use inert (z2) to avoid Q=max