Part 3:
Standard control elements
Constraint switching.

Standard Advanced control elements

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

Gives a decomposed control system:

- Each element links a subset of inputs with a subset of putputs
- Results in simple local tuning

What about the Smith Predictor?

Forget it!

Note that the Smith Predictor (Smith, 1957) is not included in the list of 18 control elements given in the Introduction, although it is a standard element in most industrial control systems to improve the control performance for processes with time delay. The reason why it is not included, is that PID control is usually a better solution, even for processes with a large time delay (Grimholt & Skogestad, 2018b; Ingimundarson & Hägglund, 2002). The exception is cases where the true time delay is known very accurately. There has been a myth that PID control works poorly for processes with delay, but this is not true (Grimholt & Skogestad, 2018b). The origin for the myth is probably that the Ziegler–Nichols PID tuning rules happen to work poorly for static processes with delay.

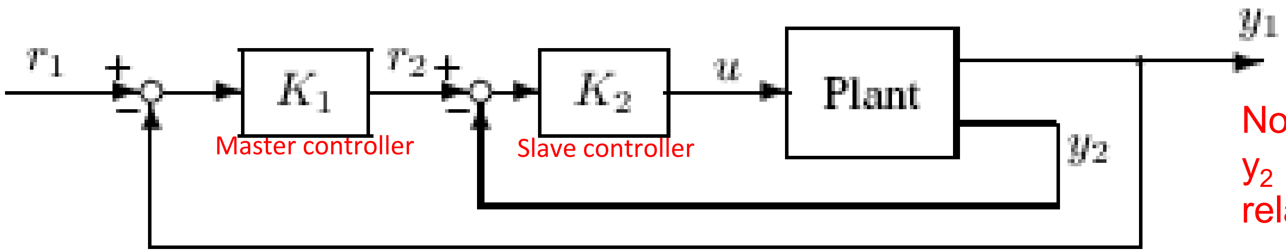
The Smith Predictor is based on using the process model in a predictive fashion, similar to how the model is used in internal model control (IMC) and model predictive control (MPC). With no model uncertainty this works well. However, if tuned a bit aggressively to get good nominal performance, the Smith Predictor (and thus also IMC and MPC) can be extremely sensitive to changes in the time delay, and even a *smaller* time delay can cause instability. When this sensitivity is taken into account, a PID controller is a better choice for first-order plus delay processes (Grimholt & Skogestad, 2018b).

Some standard advanced control elements in more detail

- E1-E18

E1. Cascade control

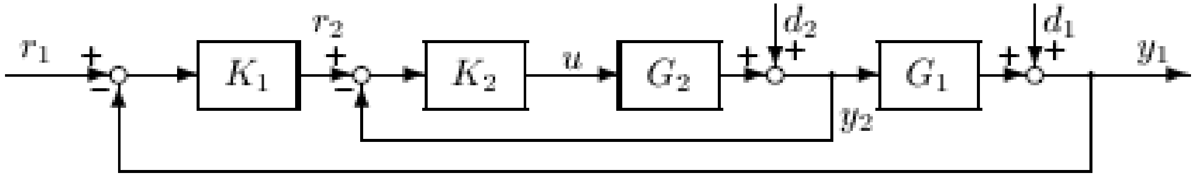
General case (“parallel cascade”)



Not always helpful…
y_2 must be closely related to y_1

(a) Extra measurements y_2 (conventional cascade control)

Special common case (“series cascade”)



Usually helpful…
Especially if slave K_2 is fast

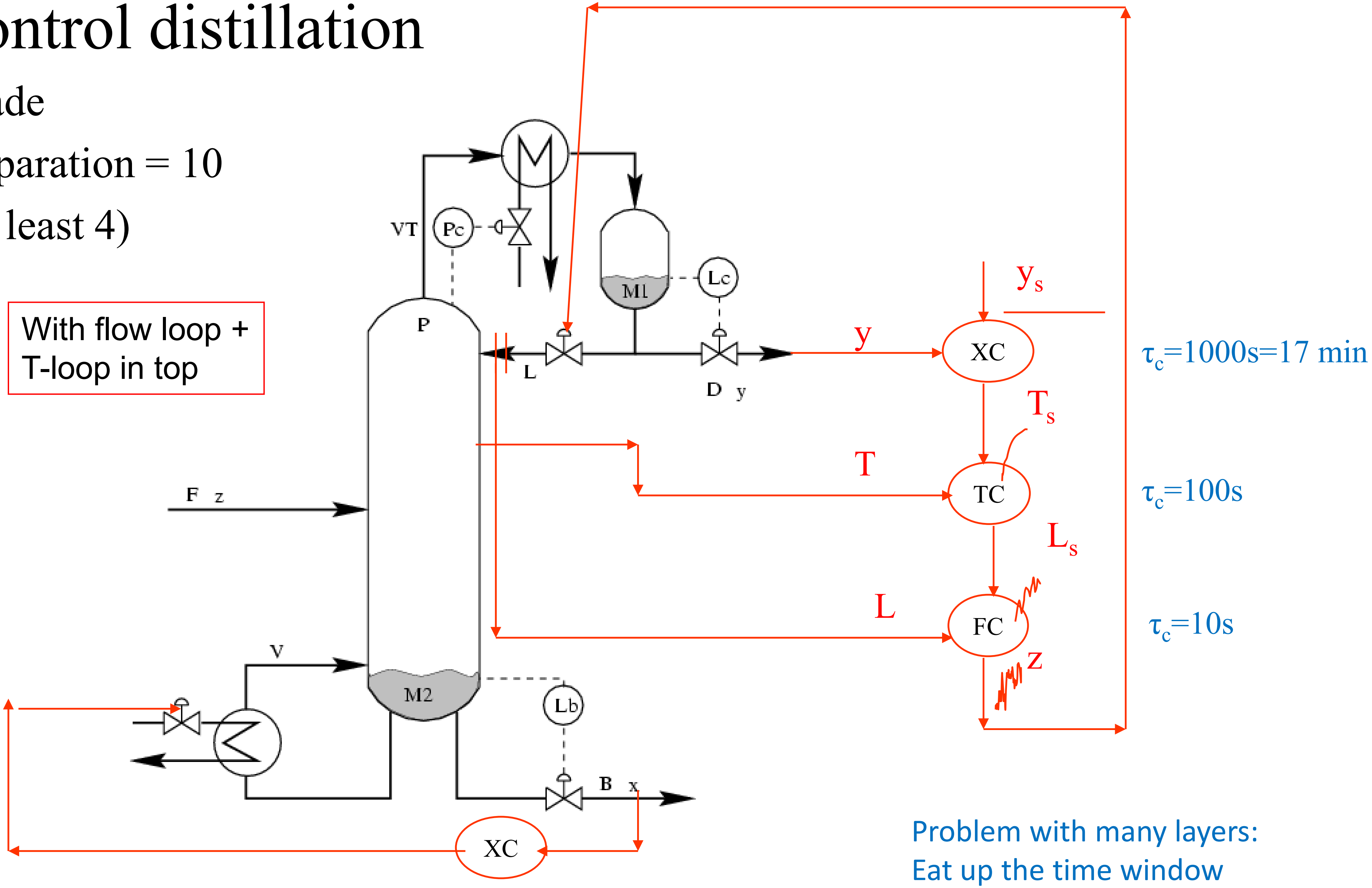
Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

Cascade control distillation

3 layers of cascade

- Time scale separation = 10

  (should be at least 4)



Problem with many layers:
Eat up the time window

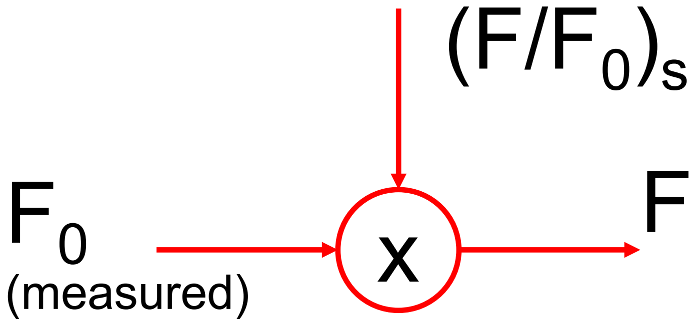
E2. Ratio control

Often viewed as special case of feedforward.

- BUT it doesn't need model
- Based on process insight: Scale all flows by same factor gives constant quality

Example: Process with two feeds F_0 (d) and F (u), where ratio should be constant.

Use* multiplication block**:



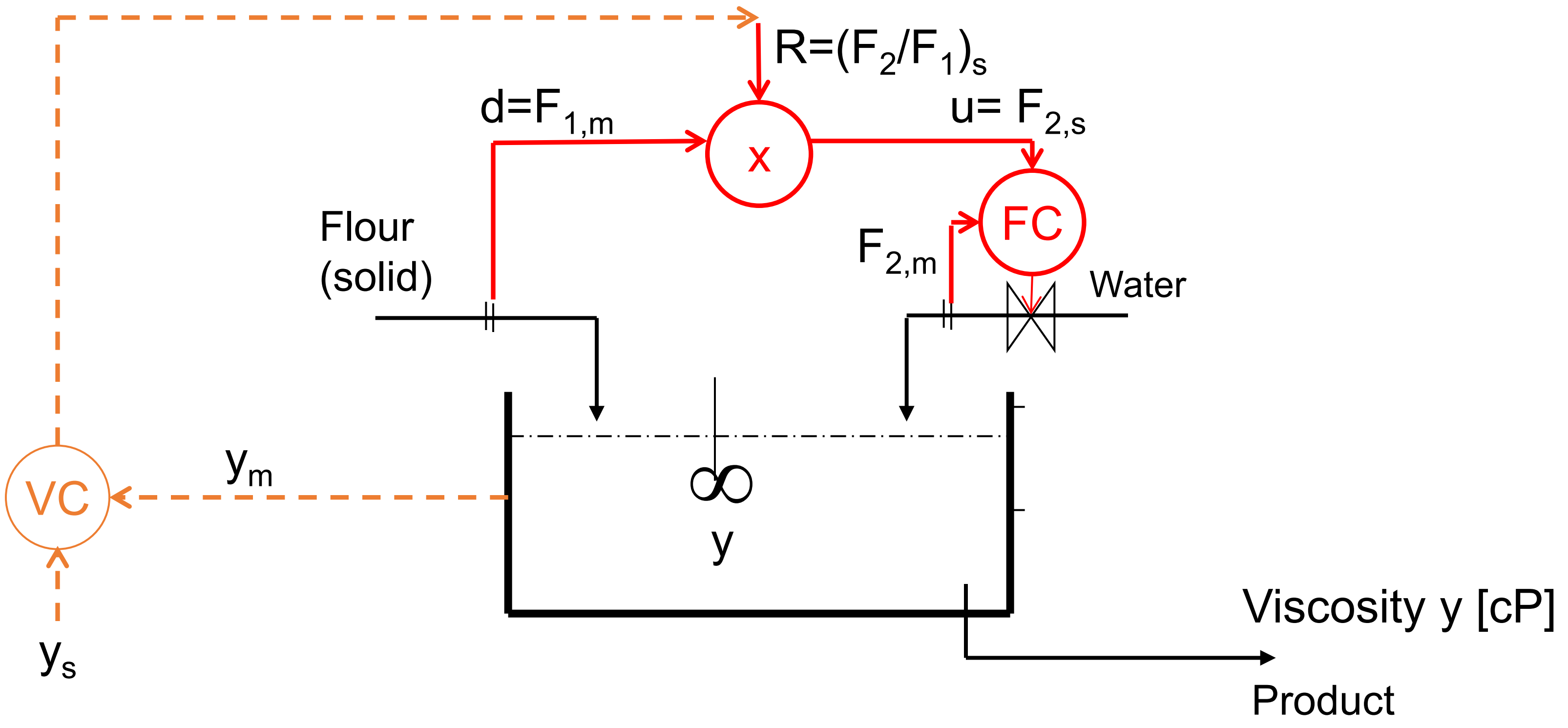
- F is usually setpoint to a flow controller
- Often $(F/F_0)_s$ is adjusted using feedback control in a cascade fashion.

* Don't use division element

** Multiplication block is sometimes called «ratio station» (bad name)

EXAMPLE: CAKE BAKING MIXING PROCESS

RATIO CONTROL with outer feedback trim (to adjust ratio setpoint)



Multiplication element (x) = Recipe (cook book) (feedforward)
VC = feedback correction of ratio setpoint

Ratio control

- Keep ratio R (between extensive variables) constant in order to keep property y constant
  - Feedforward: $R=u/d$
  - Decoupling: $R=u_1/u_2$
    - u,d: extensive variables
  - y: (any!) intensive variable
- Assumes that the «scaling property» holds
  - Based on physical insight
- Don’t really need a model (no inverse as in «normal» feedforward!)
  - Setpoint for R may be found by «feedback trim»
- Scaling property holds for mixing and equilibrium processes
    - Rato control is almost always used for mixing of reactants
  - Requires that <u>all</u> extensive variables are scaled by same amount
    - So does <u>not</u> hold for heat exchanger (since area A is constant) or non-equilibrium reactor (since volume V is constant)
    - L/F constant is <u>not</u> good for distillation column with saturated (max) heat input (V)
      - See presentation Tuesday (Bang and Skogestad)

Theoretical basis of ratio control

Ratio control: Theoretical basis and practical implementation

Sigurd Skogestad

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

DRAFT submitted to JPC on May 21, 2025

Abstract

Ratio control is the oldest control approach, dating back thousands of years (think of food recipes), but despite this, there exists no theoretical basis for its use. It is widely used in the process industry, in particular, for mixing processes and chemical rectors. It is sometimes viewed as a special case of feedforward control. However, feedforward control requires an explicit process model, but this is not needed for ratio control. Instead, ratio control is based on the physical insight that scaling all flows to keep constant flow ratios will result in constant product properties, and this scaling assumption is discussed in detail in the paper. Furthermore, the ratio setpoint may be set by an outer feedback loop, again without the need for a process model. The paper also discusses the practical implementation of ratio control, including dual ratio control for the case with saturation and cross-limiting control for keeping one component (typically oxygen) in excess during dynamic transients. Finally, it is shown that the multiplication trick proposed to avoid the limbo-effect for dual ratio control applies more generally to all split-range control solutions.

Keywords: control architecture, control structure design, feedforward control, PID control, advanced regulatory control,

6. Conclusion

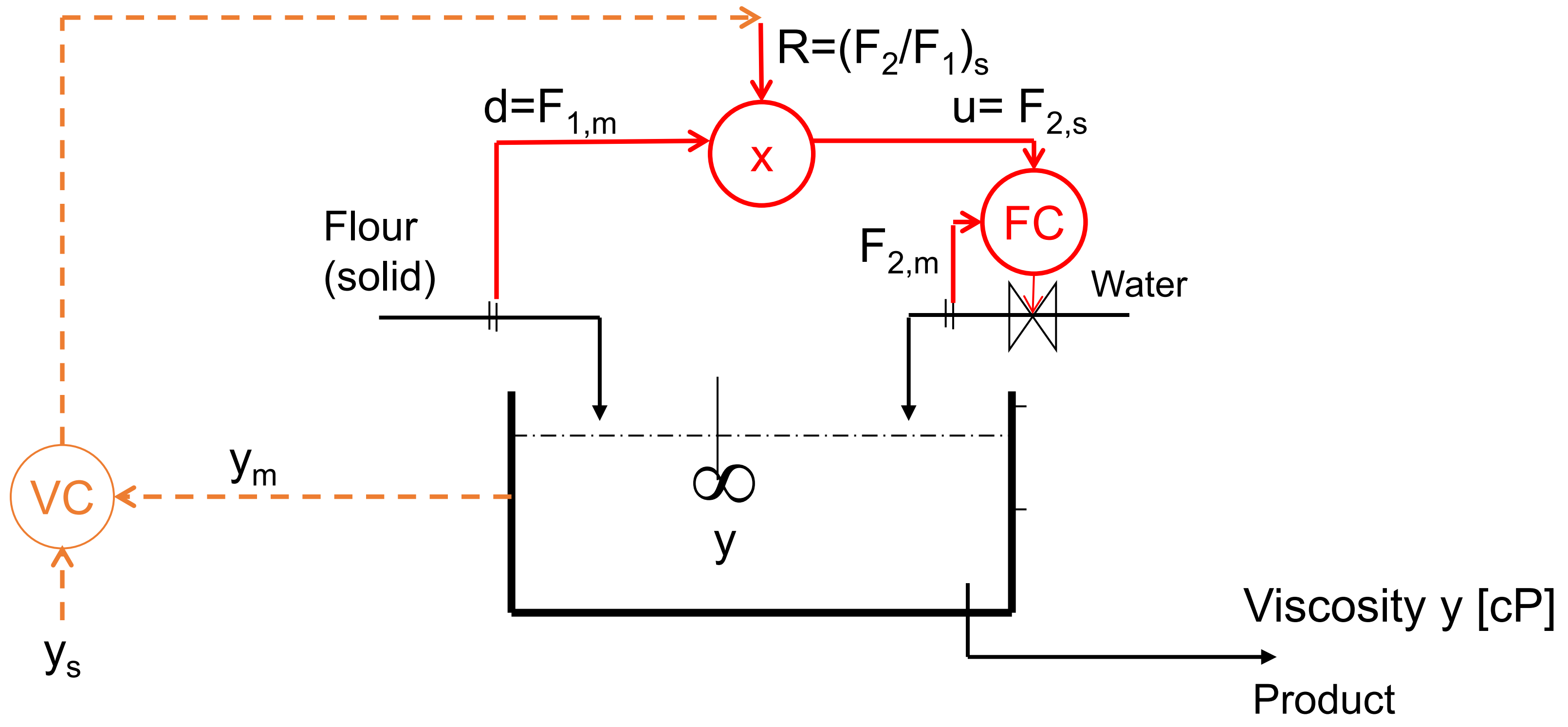
Ratio control is very simple to use and it gives nonlinear feedforward action without needing an explicit process model. It is almost always used for chemical processes to set the ratio of the reactant feed streams. Ratio control is sometimes viewed as a special case of feedforward control, but note that we do not need a model for the controlled property y for ratio control, whereas such a model is needed for feedforward control.

The theoretical basis for ratio control is the scaling assumption which says that we get the same steady-state solution if we increase all extensive variables (flows and heat rates) by the same factor compared to a basis. Similar to the use in thermodynamics, the scaling assumption holds for equilibrium systems with constant efficiencies.

The scaling assumption is formulated mathematically in (2). From this we derived the following rules for the use of ratio control:

- (R1) The controlled variable y is implicitly assumed to be an intensive variable, for example, composition, density, viscosity, taste or temperature.
- (R2) The system must satisfy the scaling assumption (2).
- (R3) Since all extensive variables must be scaled by the same factor k, there can only be one independent extensive disturbance variable. This variable is sometimes called the "basis", "wild variable", "master variable", "flow disturbance" or "throughput manipulator" (TPM).
- (R4) If the system has n independent extensive variables X_i, then from (2) we need to manipulate $n-1$ of these variables to keep the $n-1$ ratios constant (or more generally, $n-1$ dependent intensive variables y_i). For a change (disturbance) in the throughput (basis, wild flow), this will result in keeping *all* dependent intensive variables constant, including the controlled variable(s) y (at steady state).

LINEARITY OF RATIO CONTROL



Note : This way of implementing ratio control makes it easy to tune the outer feedback loop (CC: composition controller) because the gain from MV = R_s to CV=y does not depend on disturbance d=F_1.

Ratio control with feedforward: This implementation may be used if there are measured disturbances in feed quality

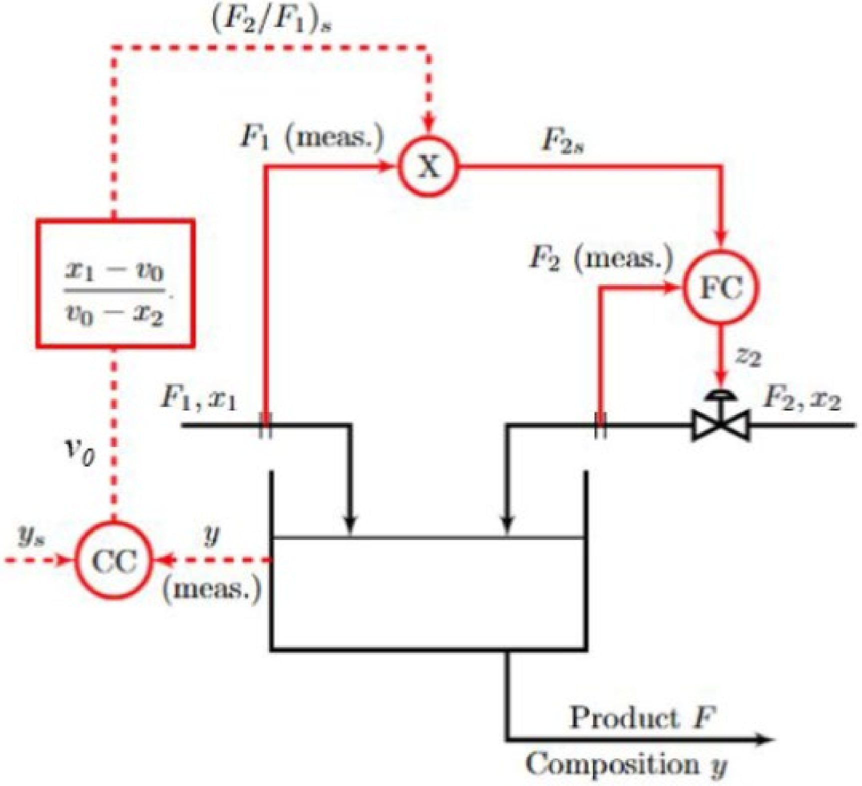


Figure 14: Improved ratio control scheme for mixing process using transformed input v_0. The feed mass fractions x_1 and x_2 that enter the computation block, need to be measured or estimated.

From the steady-state component material balance, we have that y is the weighted average of the feed fractions (recall (4))

$$y = f_0(u, d) = \frac{x_1 F_1 + x_2 F_2}{F_1 + F_2} \tag{12}$$

Note that $u = F_2$. The transformed input is defined as the right-hand side of this equation, $v_0 = f_0(u, d)$. Note that v_0 is the output from the feedback controller. Inverting (12), we find how the input $u = F_2$ depends on the transformed input v_0:

$$u = F_2 = f_0^{-1}(v_0, d) = \frac{x_1 - v_0}{v_0 - x_2} F_1 \tag{13}$$

Theoretical basis: Transformed input $v_0 = f_0(u, d)$ chosen equal to RHS of static model $y = f_0(u, d)$. Resulting model for outer controller (CC): $y = I\ v_0$ (linear, decoupled and perfect disturbance rejection!). Seems too simple to be true, but it works!

Valve position control (VPC)

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

Two different cases of VPC:

- E3. Extra dynamic MV for faster control
  - Both MVs are used all the time
- E7. “Split” VPC (extra static MV)
  - Alt.3 for MV-MV switching:
    - Need several MVs to cover whole range at steady state
    - We want to use one MV at a time

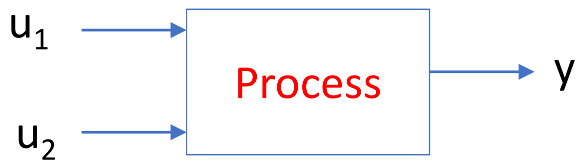
E3. VPC on extra dynamic input

u_2 = main input for steady-state control of CV

(but u_2 is poor for directly controlling y

- e.g. time delay or u_2 is on/off)

u_1 = extra dynamic input for fast control of y



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

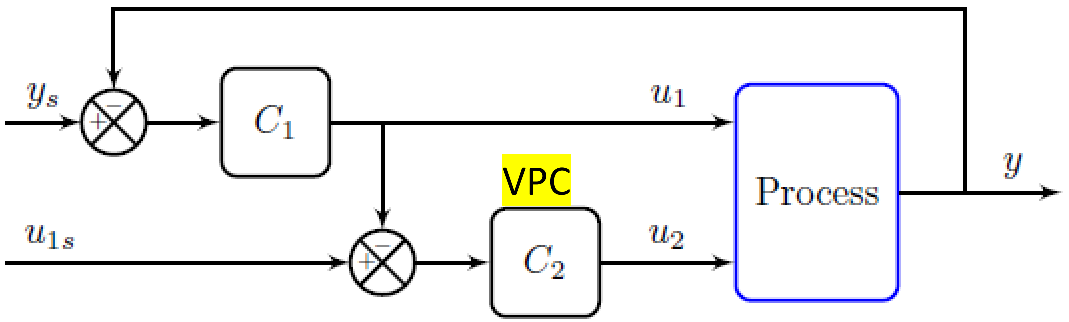


Figure 12: Valve (input) position control (VPC) for the case when an "extra" MV (u_1) is used to improve the dynamic response. A typical example is when u_1 is a small fast valve and u_2 is a large slower valve.
C_1 = fast controller for y using u_1.
C_2 = slow valve position controller for u_1 using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

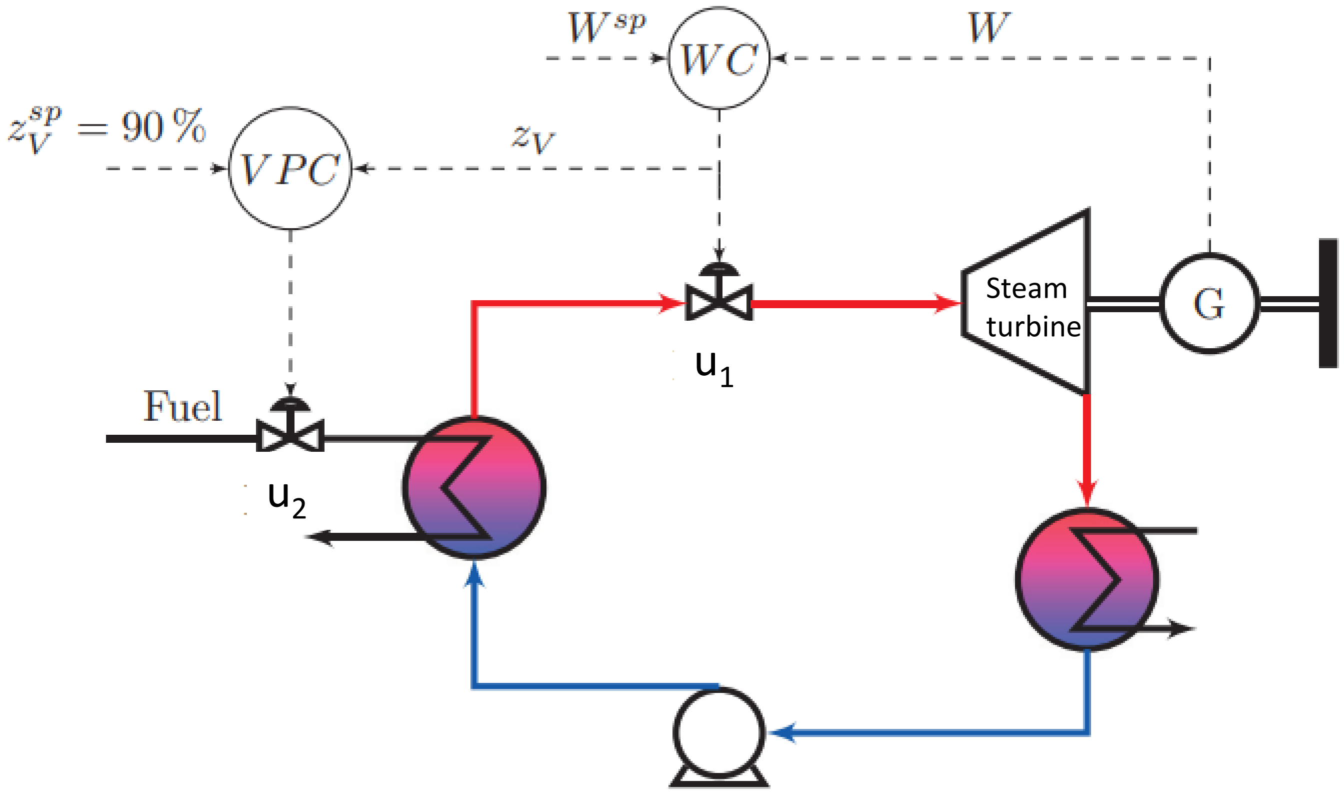
Example 1: Large (u_2) and small valve (u_1) (in parallell) for controlling total flowrate (y=F)

- The large valve (u_2) has a lot of stiction which gives oscillations if used alone for flow control
- The small valve (u_1) has less stiction and gives good flow control, but it's too small to use alone

Alternative term for dynamic VPC:

- Mid-ranging control (Sweden)

Example 2 VPC: Power plant control

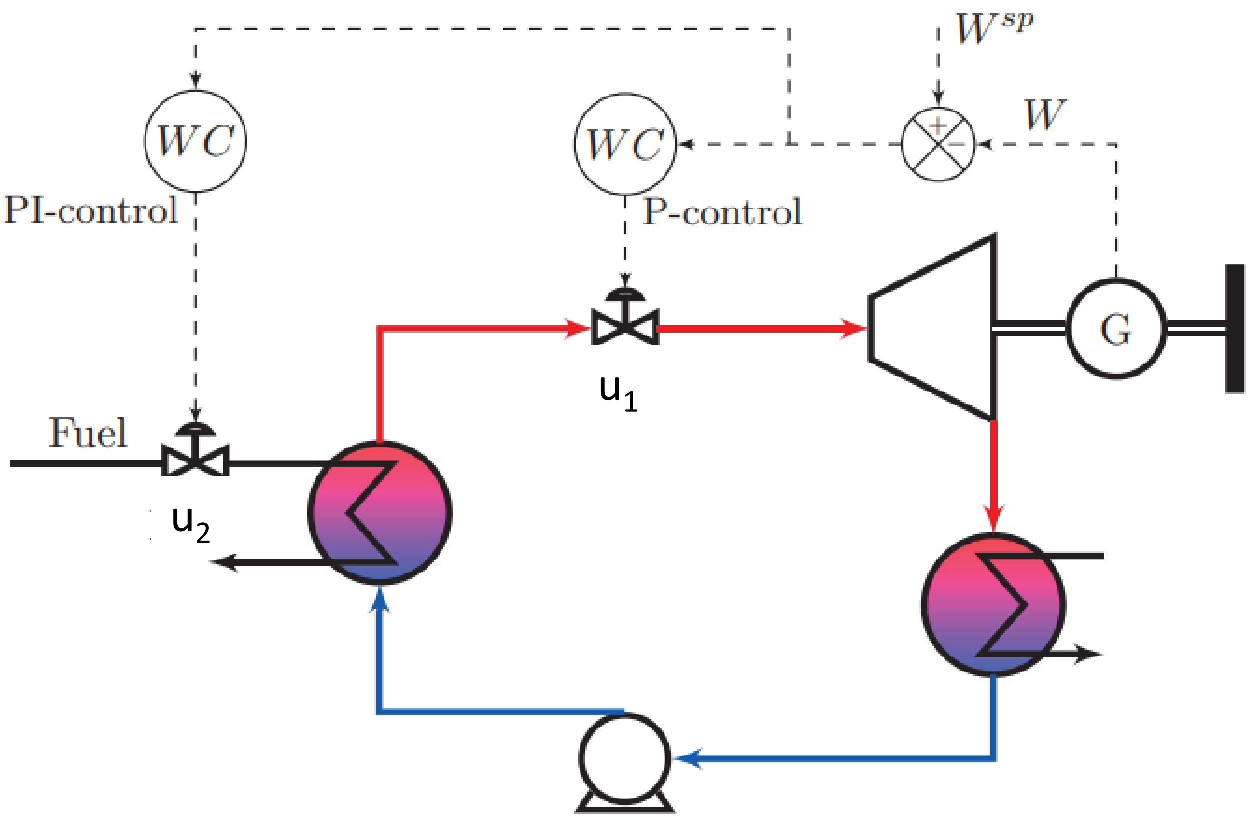


y = power W [MW]
u_1 = steam valve z (fast transient effect)
u_2 = Fuel (slow static effect), $y_2 = u_{1s}$ = 90%

- u_1 returns to u_{1s}=90% because of VPC
- Both controllers may be PI
- Need time scale separation: VPC slow

Optimal operation and control of heat to power cycles: A new perspective from a systematic plantwide control approach
Cristina Zotica , Lars O. Nord, Jenö Kovács , Sigurd Skogestad. Computers & Chemical Engineering,. Volume 141, 4 October 2020, 106995

Alternative to VPC: Parallel control



y = power W [MW]
u_1 = steam valve z (Fast P-control, $u_1=u_{10} + K_c\, e$)
u_2 = Fuel (Slow PI-control)

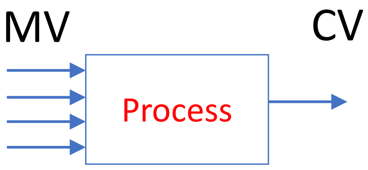
- u_1 returns to bias u_{01}=90% because PI controller gives e = W_s-W = 0 at steady state

Switching of MVs and CVs

Constraint switching
(because it is optimal at steady state)

A. MV-MV switching

- Use one MV at a time
- Three alternatives



B. CV-CV switching

- Control one CV at a time
- Always selector (or similar)

- ## MV-CV switching
  - MV saturates so must give up CV

C. Simple («do nothing»)

D. Complex (repairing of loops)

A. 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 main solutions for "selecting the right MV":

Alt.1: (Standard) Split-range control (one controller)

Alt.2 Split-parallel control (Many controllers with different setpoints)

Alt.3 (Split) Valve position control (VPC)

In addition: MPC

Which is best? It depends on the case!

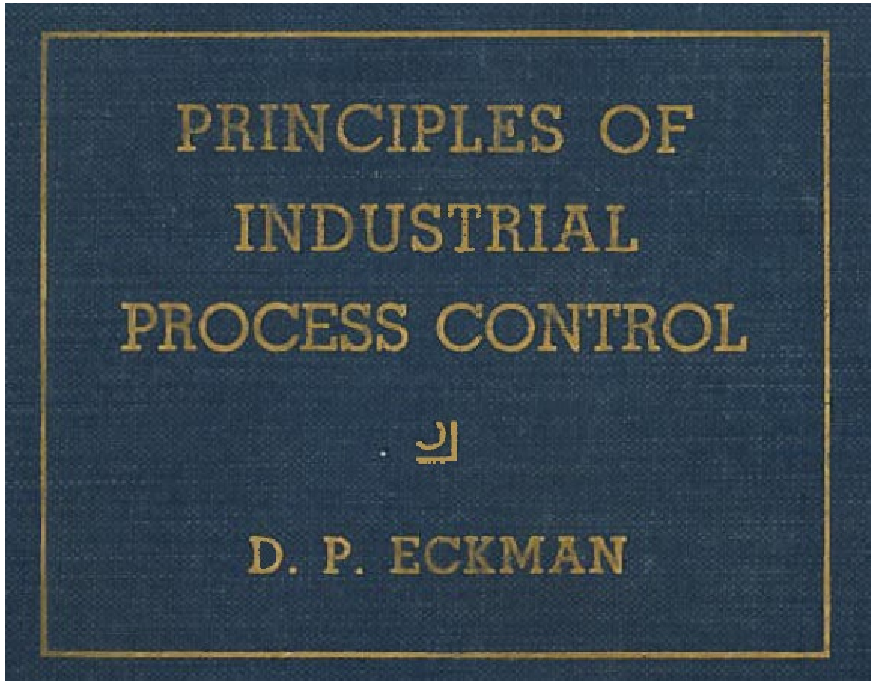
* Adriana Reyes-Lua Cristina Zotica, Sigurd Skogestad, «Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control,, Adchem Conference, Shenyang, China. July 2018 ,

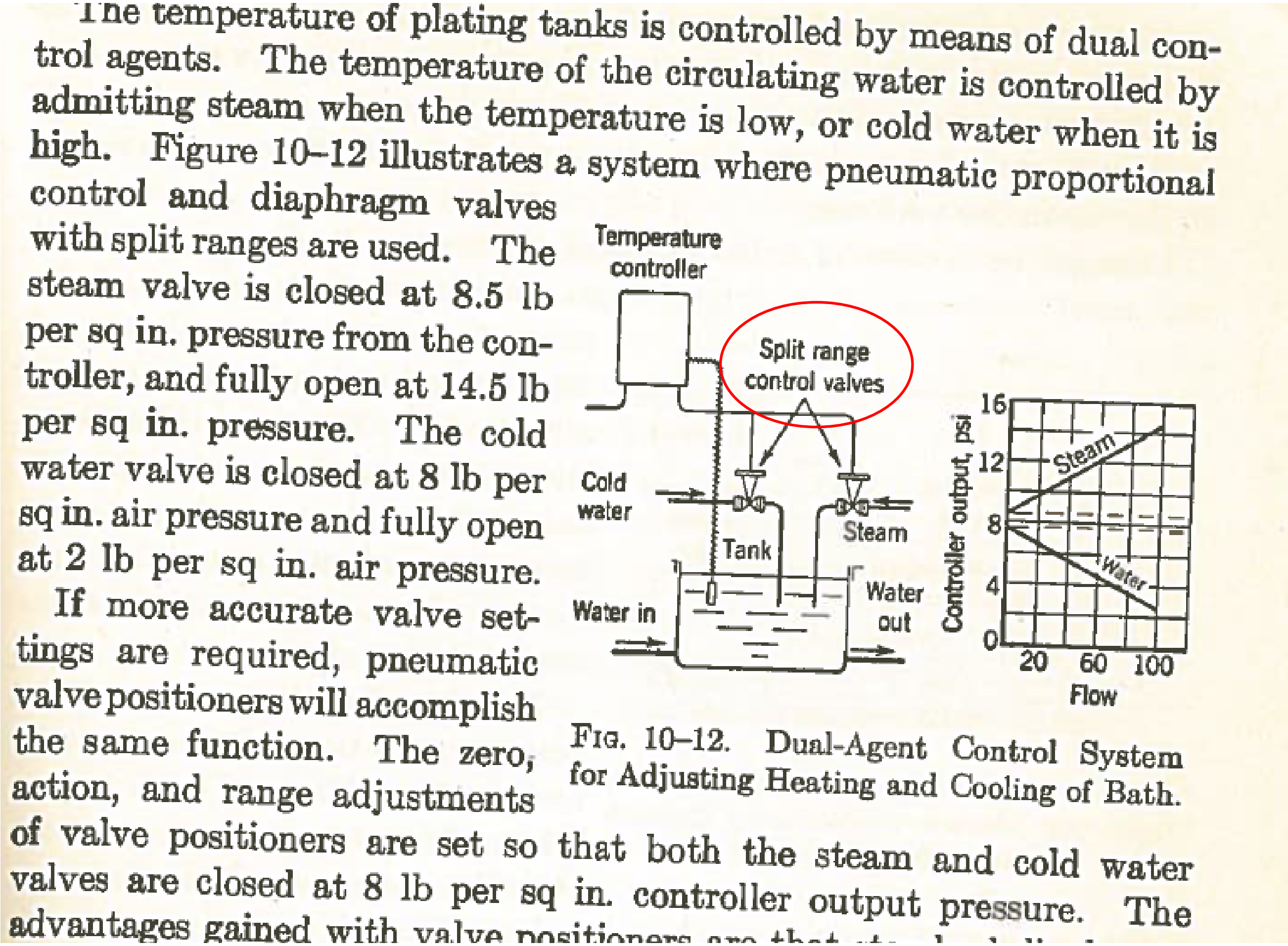
A. Reyes-Lúa and S. Skogestad. "Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy". Journal of Process Control 91 (2020)

MV-MV switching, Alt. 1

Split range control:

Donald Eckman (1945)





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

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

MV-MV switching, Alt. 1

Split-range control (SRC)

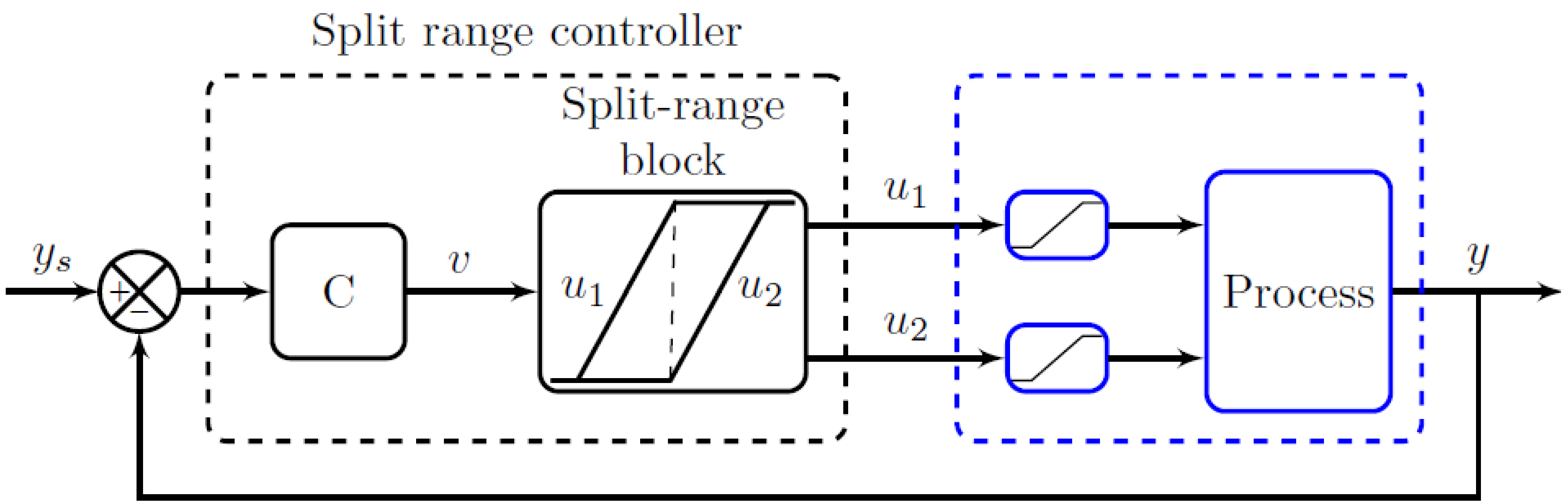


Figure 21: Split range control for MV-MV switching.

[7]Note the blue saturation elements for the inputs in Figure 21 and other block diagrams. Saturation can occur for any physical input, but they are explicitly shown for cases where the saturation is either the reason for or part of the control logic. For example, in Figure 21, the reason for using u_2 is that u_1 may saturate.

For MVs (u) that have same effect (same sign) on the output (y) (Fig. 21), we need to define the order in which the MVs will be used. This is done by the order in in the SR-block.

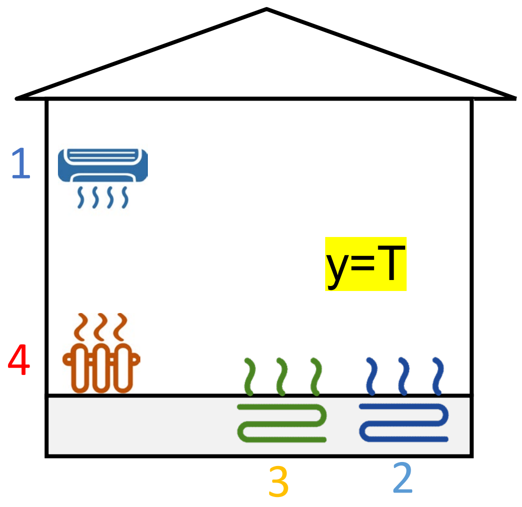
Example: With two heating sources, we need to decide which to use first (see next Example)

SRC is easy to understand and implement!

Disadvantages:

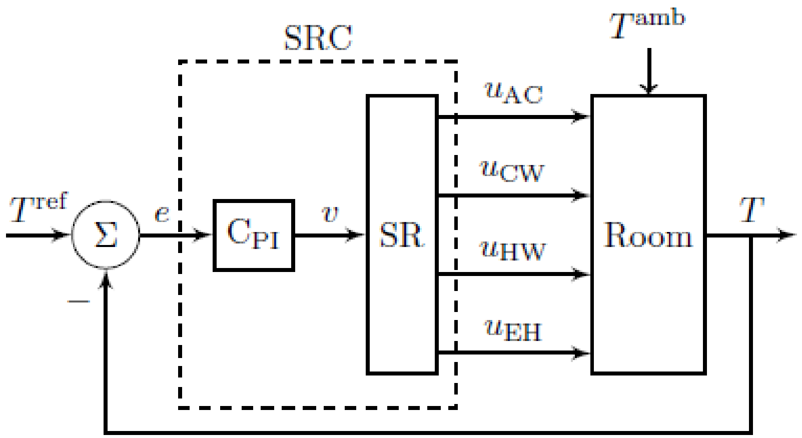
1. Only one controller ⇒ Same integral time for all inputs u_i (MVs)
   – Controller gains can be adjusted with slopes in SR-block!
2. Does not work well for cases where constraint values for u_i change

Example split range control: Room temperature with 4 MVs

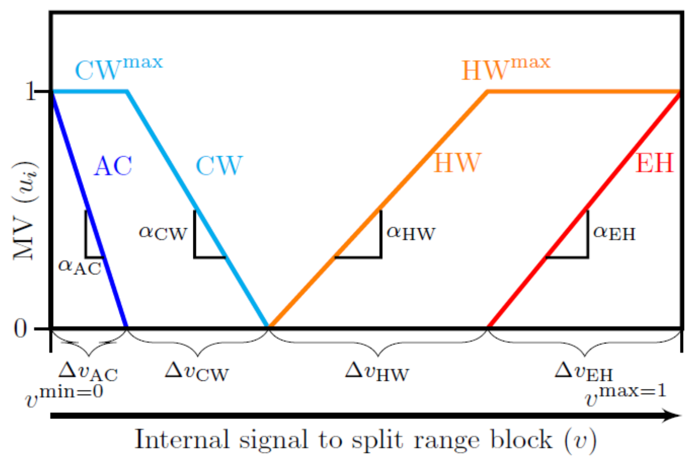


MVs (two for summer and two for winter):

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



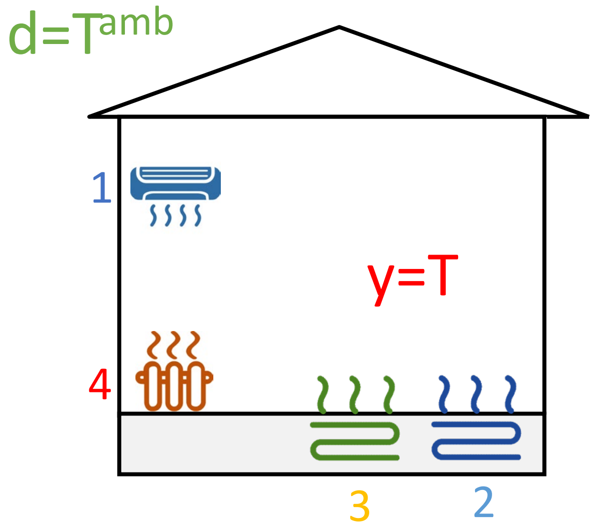
SR-block:

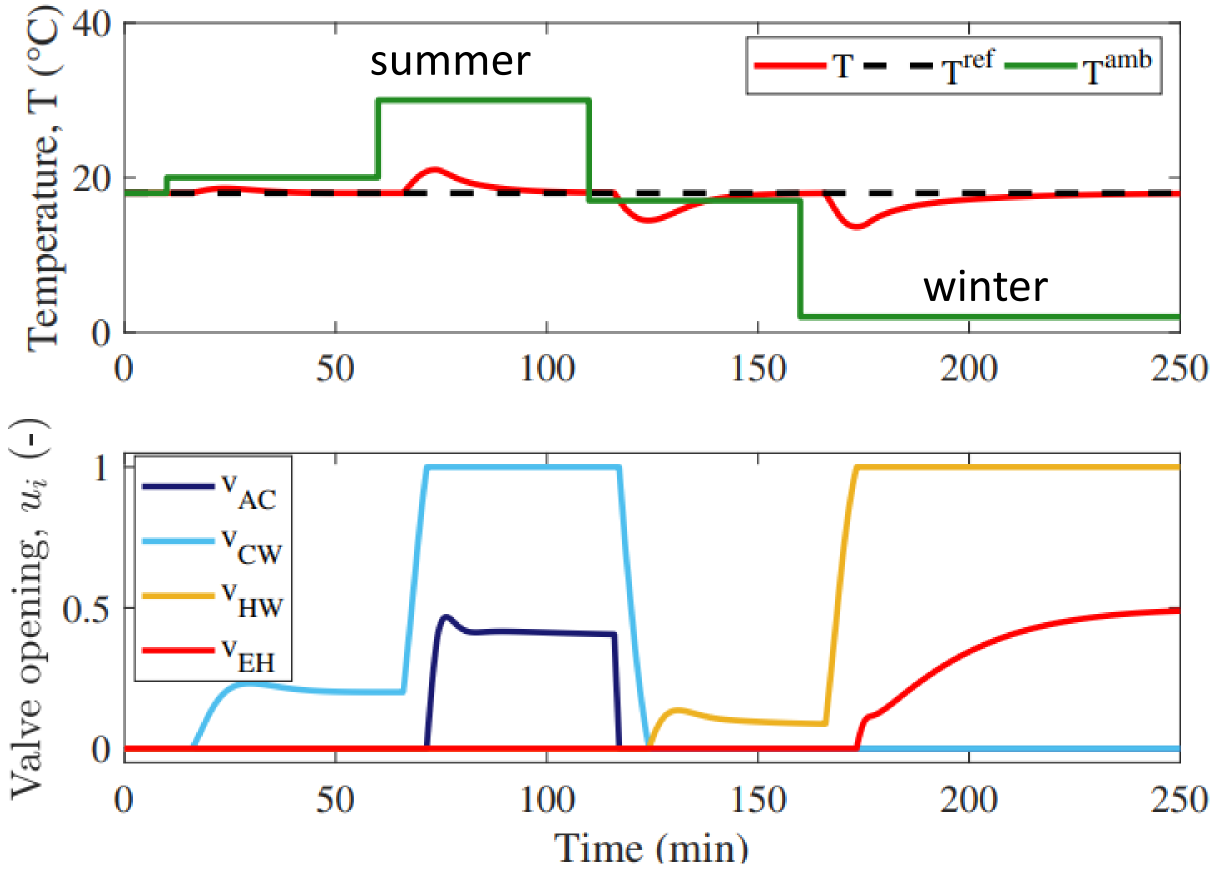


C_{PI} – same controller for all inputs (one integral time)
But get different gains by adjusting slopes α in SR-block

MV-MV switching

Simulation Split-range control (SRC)





MV-MV switching, Alt. 2

Split parallel control:

Separate controllers with different setpointsS

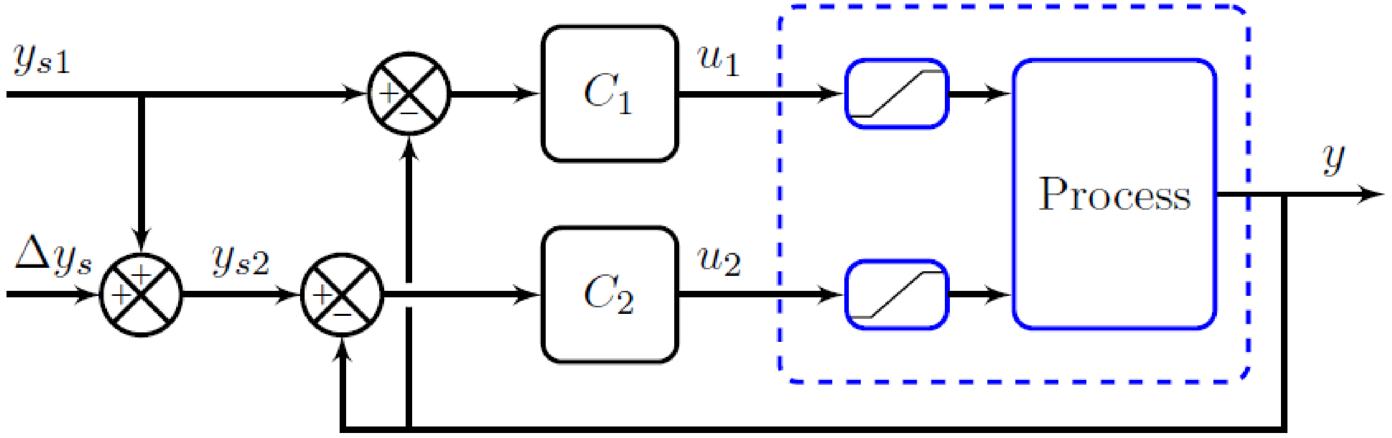


Figure 22: Separate controllers with different setpoints for MV-MV switching.

The setpoints $(y_{s1}, y_{s2}, \ldots)$ should in the same order as we want to use the MVs. The setpoint differences (e.g., $\Delta y_s = y_{s2} - y_{s1}$ in Fig. 22) should be large enough so that, in spite of disturbances and measurement noise for y, only one controller (and its associated MV) is active at a given time (with the other MVs at their relevant limits).

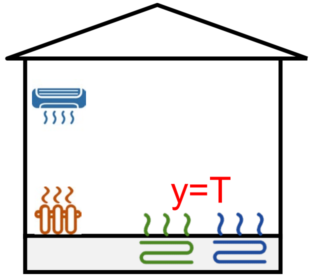
Advantages:

1. Simple to implement (no logic)
2. Controllers can be tuned independently (different integral times)
3. Switching by feedback: Do not need to know constraint values
   - Big advantage when switching point varies (complex MV-CV switching)

Disadvantages:

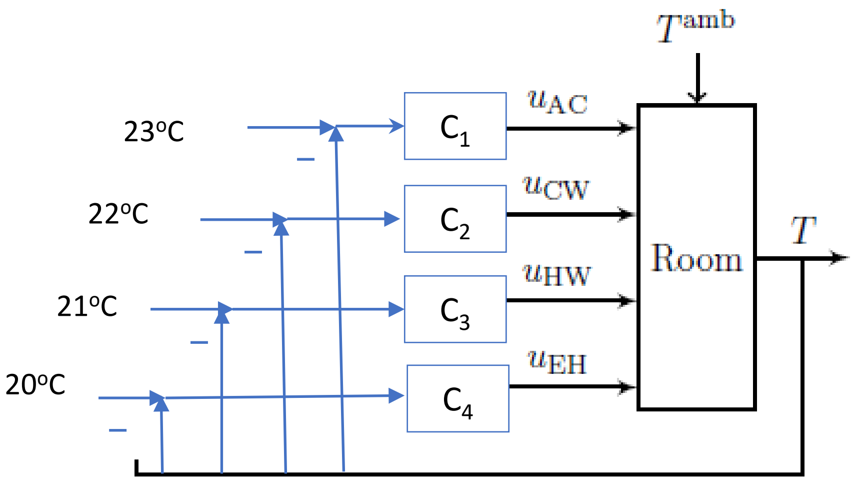
1. Temporary loose control during switching
2. Setpoint not constant
   - Can be an advantage!! (gives energy savings for room heating)

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)

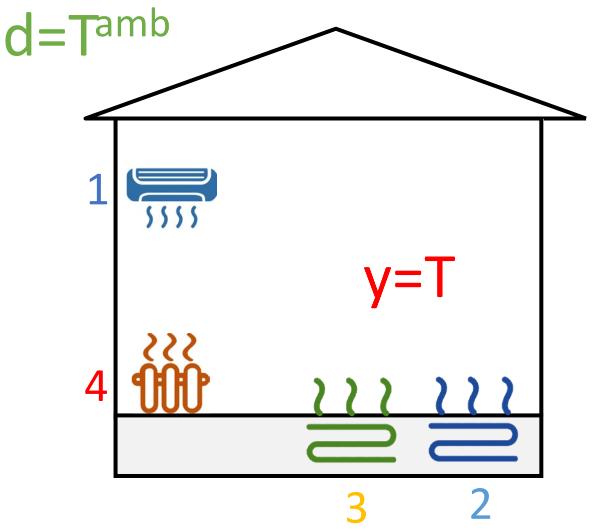
Alt. 2 for MV-MV switching. Split parallel control

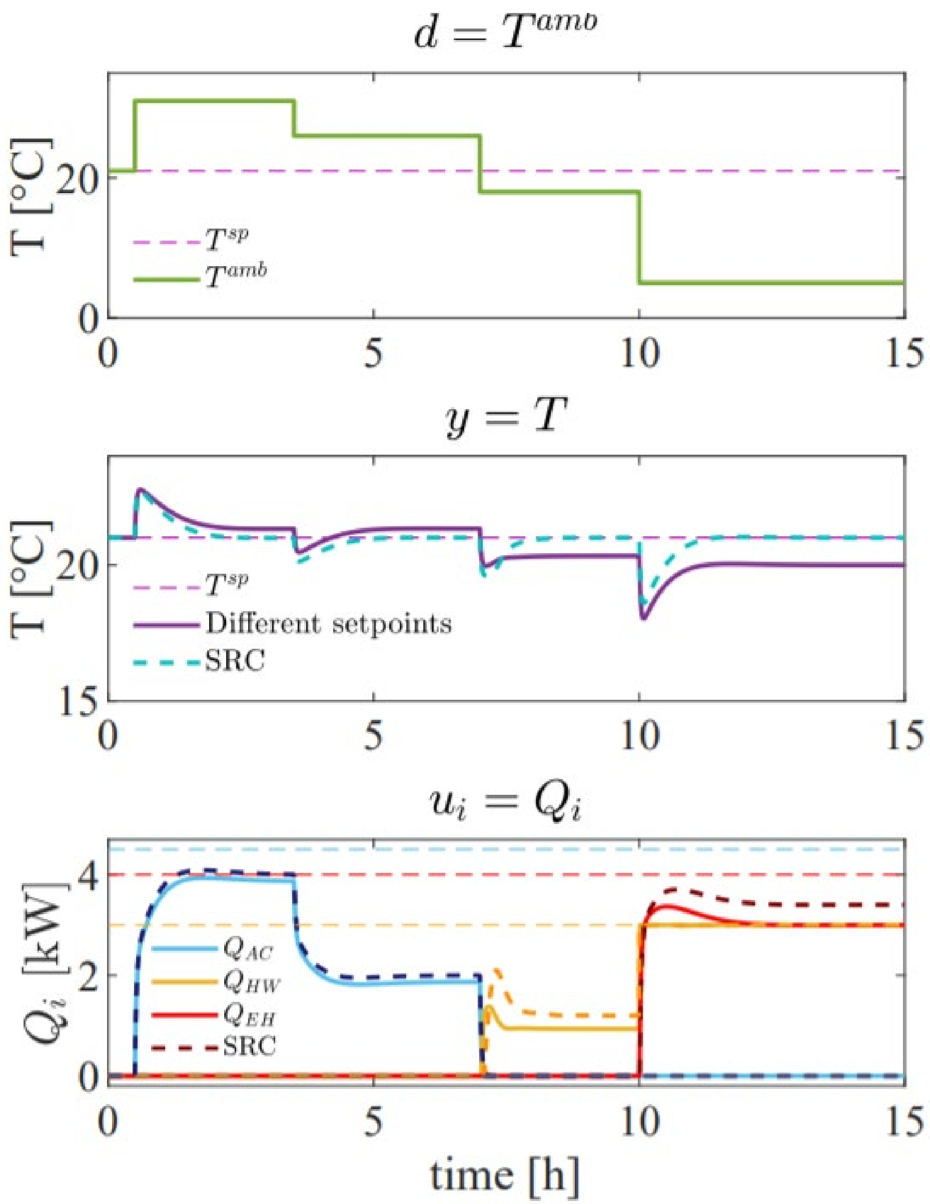


Disadvantage (comfort):
- Different setpoints

Advantage (economics) :
- Different setpoints (energy savings)

MV-MV switching, Alt. 2 vs. Alt. 1





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

Fix Split-parallel control: Outer cascade to avoid different setpoints

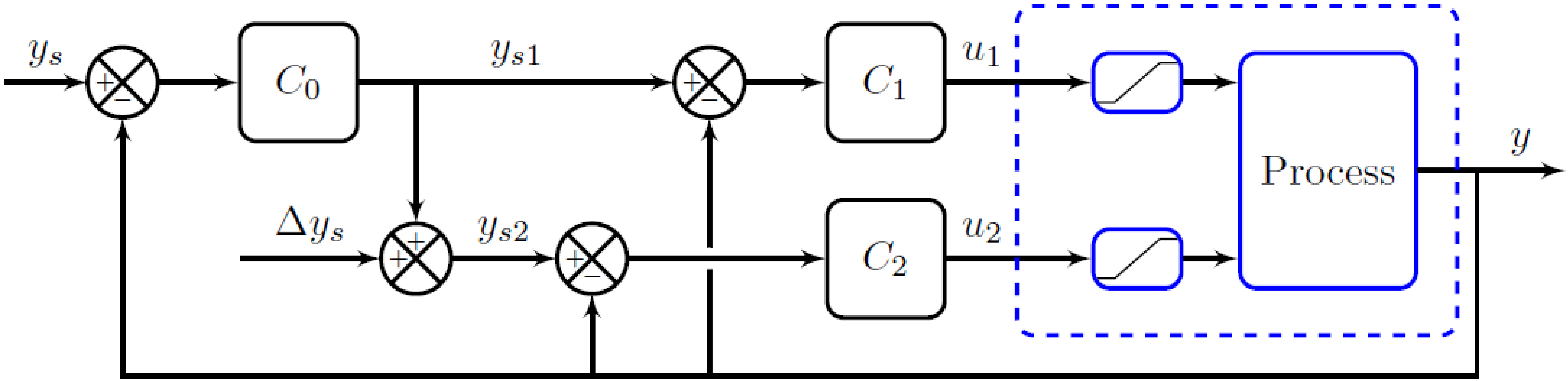


Figure 23: Separate controllers for MV-MV switching with outer resetting of setpoint. This is an extension of the scheme in Figure 22, with a slower outer controller C_0 that resets y_{1s} to keep a fixed setpoint $y = y_s$ at steady state.

“Split” VPC (E7)

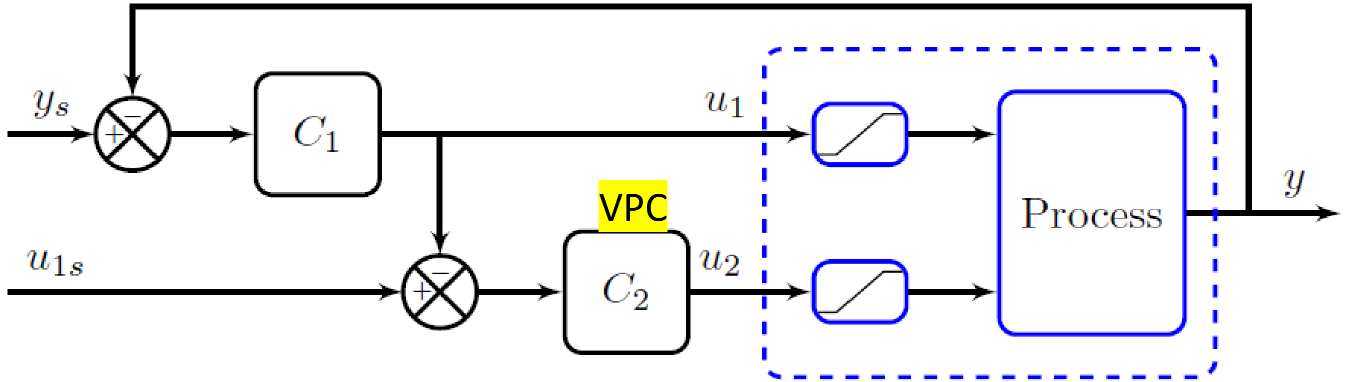


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

Use VPC for MV-MV switching when we always want to use u_1 to control y

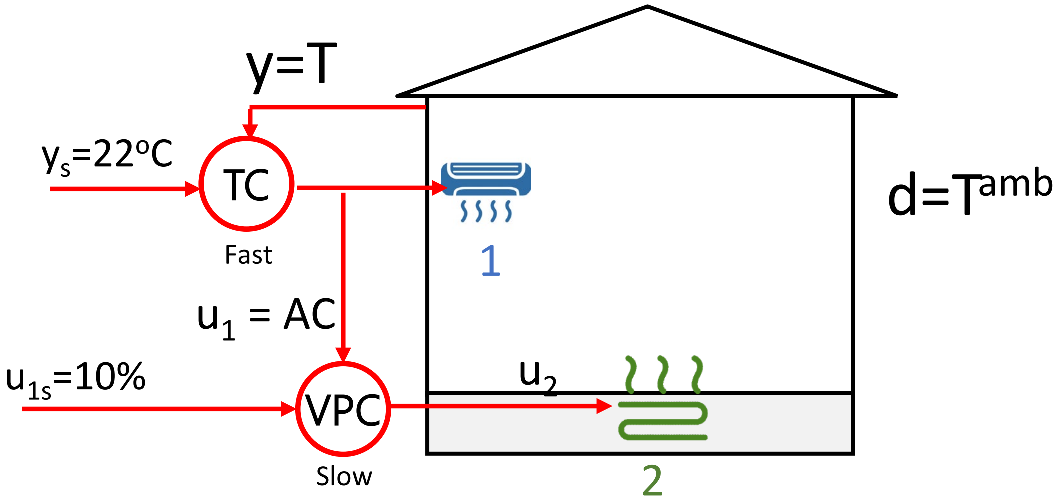
- For example, u_2 may only allow discrete changes (e.g., u_2=0,1,2,3,4)
- or dynamics for u_2 may be very slow

Disadvantages VPC for MV-MV switching:

1. We cannot let u_1 become fully saturated because then control of y is lost
   - This means that we cannot use the full range for u_1 (potential economic loss)

2. When u_2 is used, we need to keep using a ‘‘little’’ of u_1.
   - Example: If the two MVs (inputs) for temperature control are heating (u_1) and cooling (u_2), then we need to use both heating and cooling at the same time in the summer (when heating normally should be off).

Split VPC for MV-MV switching
Example: Room heating with fast cooling (AC) and slow heating



MVs:
1. u_1= AC (cooling with fan, fast)
2. u_2 = HW (hot water in floor, slow)

u_2= Hot water (VPC) is only used in winter (in the summer u_2=0%).

Advantage: Temperature is always controlled by fast cooling (u_1=AC)
Economic disadvantage: Cooling u_1 is used also in winter (about 10% load)

Beware: Two different applications of VPC (E3 and E7)

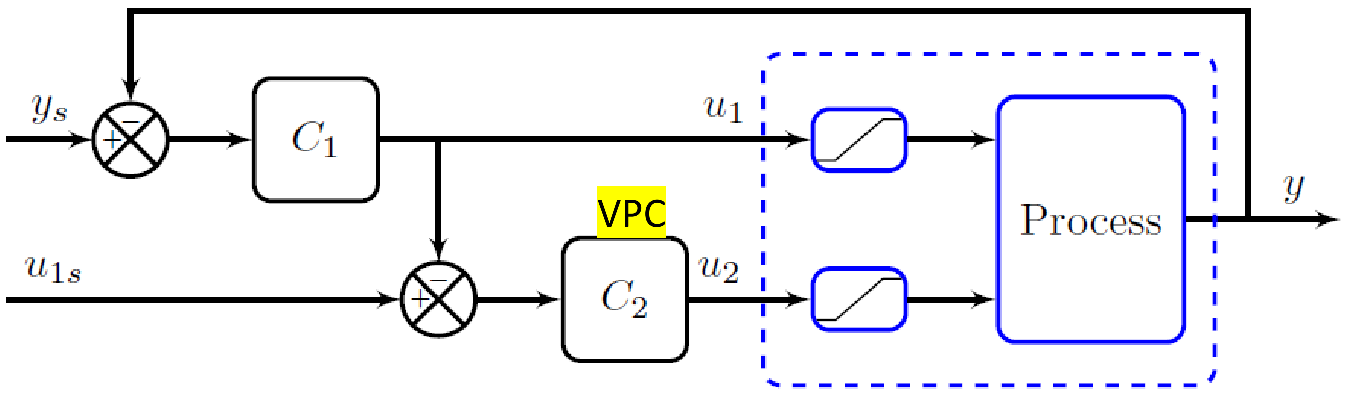


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

Same block diagram, except for the “need” for valve saturation
But their motivation is different!

- “Normal” VPC for improving dynamics (E3) – u_1 is the “extra MV” to get fast control
  - both inputs are used all the time
  - u_2 is the main steady-state input (and used all the time)
  - u_{1s} is typically 50% (mid-range)

I frequently see people confuse these two cases - which is very understandable!

- “Split” VPC for MV-MV switching (E7) – u_2 is the “extra MV” to avoid saturation for u_1
  - u_2 is only used when u_1 approaches saturation (MV-MV switching)
  - u_{1s} is typically close to the expected saturation limit (10% or 90%)

Summary 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.

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)

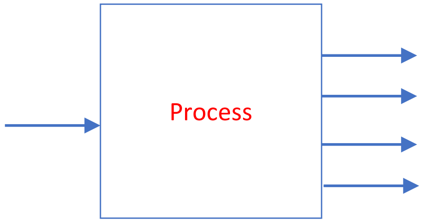
Alt.3 Valve position control (E7)

- 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 ,**

B. CV-CV switching



- One MV
- Many CVs, but control only one at a time
- Solution: One controller for each CV + Selector
- Often called "override"

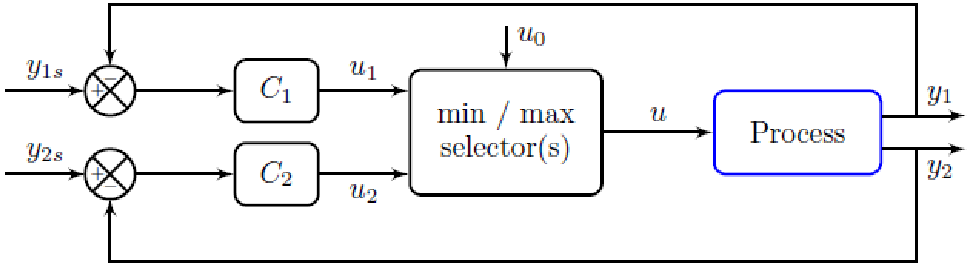
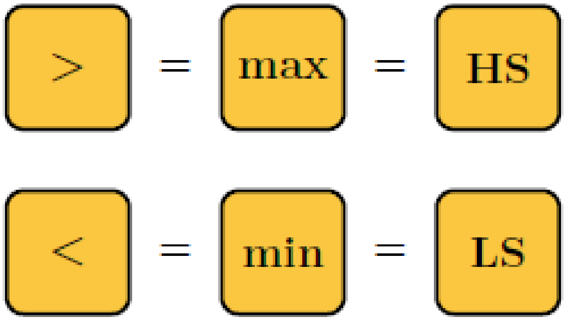


Figure 17: CV-CV switching with selector on MV (input u).

E4. Selector (for CV-CV switching*)

- Many CVs paired with one MV.
- But only one CV controlled at a time.
- Use: Max or Min selector

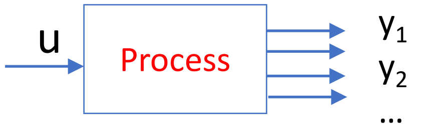


Note: Selectors are logic blocks

- Sometimes called “override”
  - But this term may be misleading
- Selector is generally on MV (compare output from many controllers)

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

Implementation selector



Alt. I (General). Several controllers (different CVs)

- Selector on MV (!!)
  - Must have anti windup in c1 and c2 !

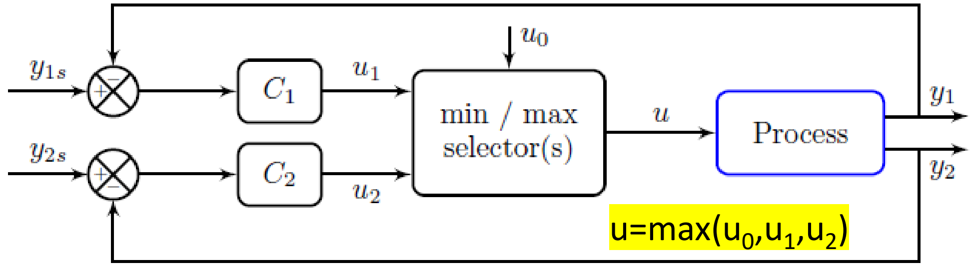
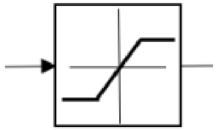


Figure 17: CV-CV switching with selector on MV (input u).

Alt. II (Less general) Controllers in cascade

- Selector on CV setpoint
- In this case: Selector may be replaced by saturation element (with y2s as the max) or min)



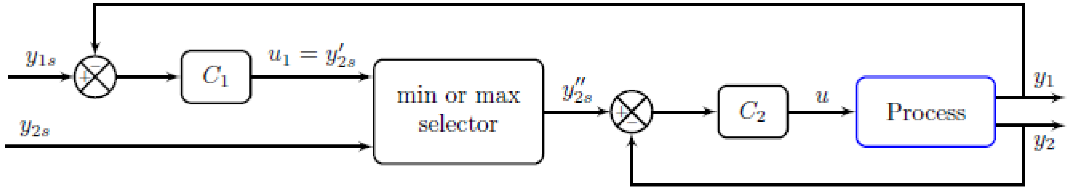
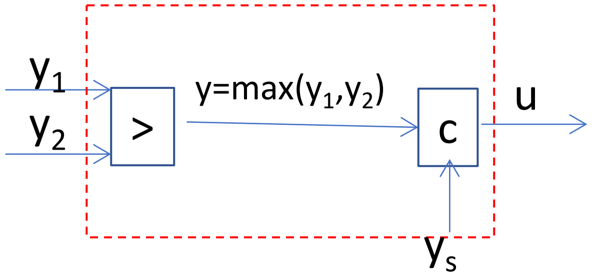


Figure 19: Alternative cascade CV-CV switching implementation with selector on the setpoint. In many cases, y_{1s} and y_{2s} are constraint limits.

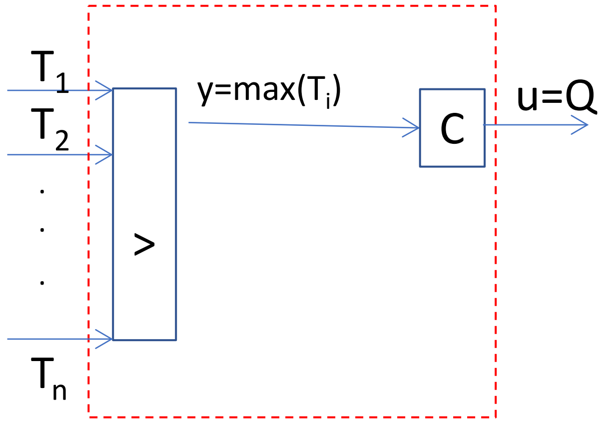
Alt. III (For special case where all CVs have same bound). One controller

- Selector is on CVs (Auctioneering)
- Also assumes that dynamics from u to y_1 and y_2 are similar; otherwise use Alt.I
- Example: Control hot-spot in reactor or furnace.



Example Alt. III

- Hot-spot control in reactor or furnace



- Comment: Could use General Alternative I (many controllers) for hot-spot control, with each temperature controller (c_1, c_2,…) computing the heat input (u_1=Q_1, u_2=Q_2, ….) and then select u = **min**(u_1, u_2, …), but it is more complicated.

Furnace control with safety constraint (Alt. I)

Input (MV)

u = Fuel gas flowrate

Output (CV)

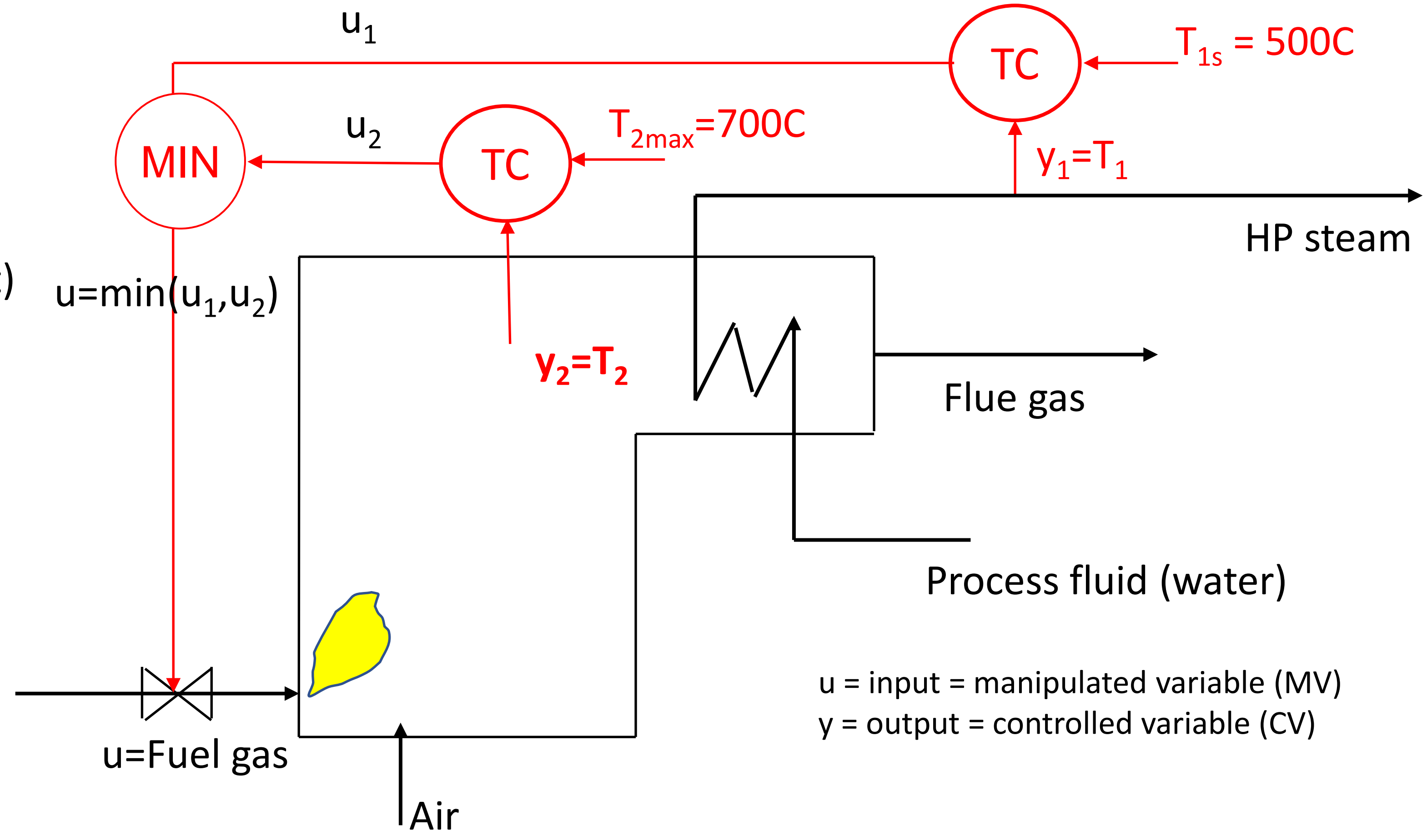
y_1 = process temperature T_1

(desired setpoint or max constraint)

y_2 = furnace temperature T_2

(max constraint)

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

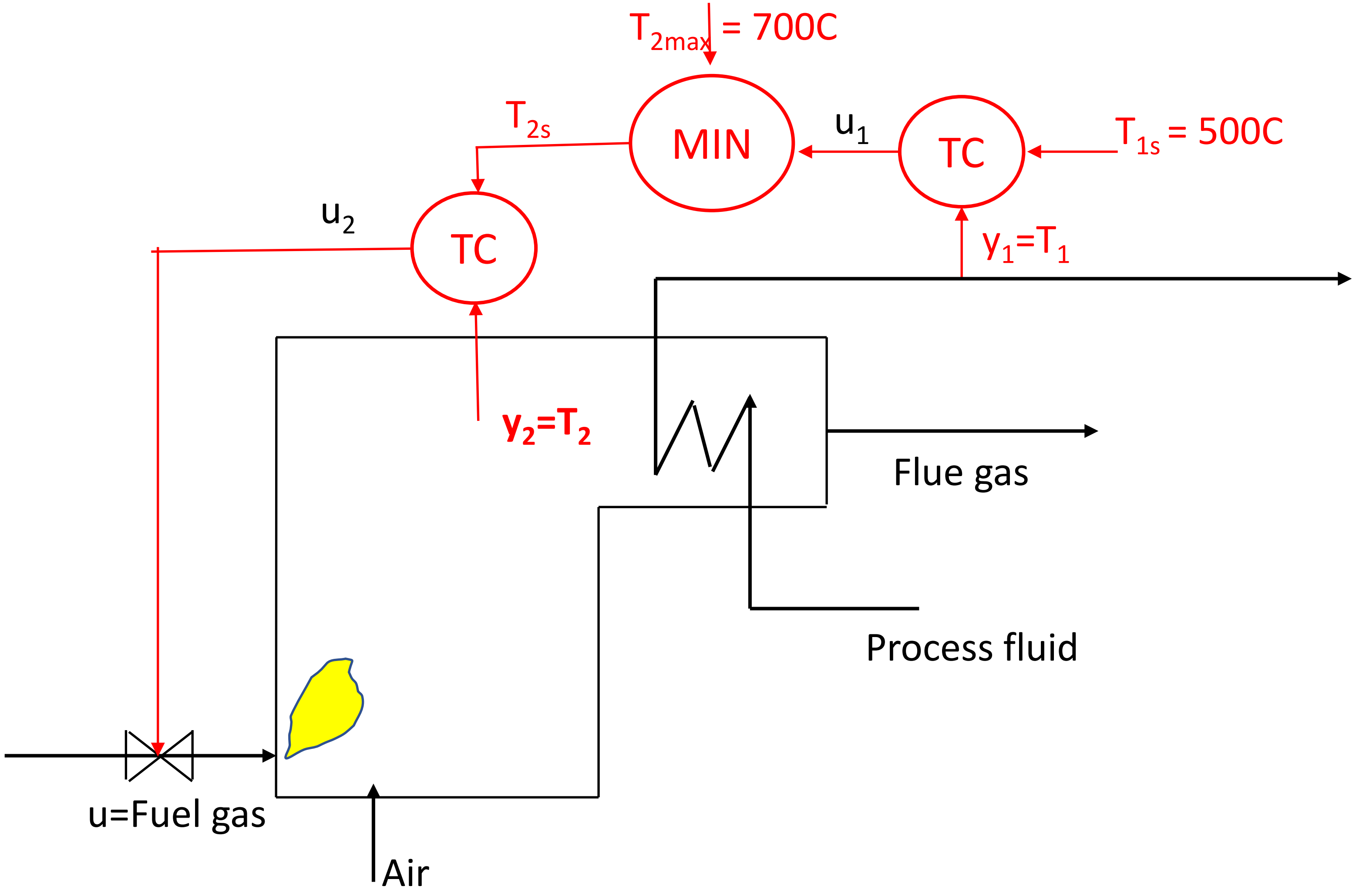


u = input = manipulated variable (MV)

y = output = controlled variable (CV)

Furnace control with cascade (Alt. II, selector on CV-sp)

Comparison
The cascade solution is less general but it may be better in this case.
Why better? Inner T2-loop is fast and always active and may improve control of T1.



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)

Rule 2 (order of selectors)

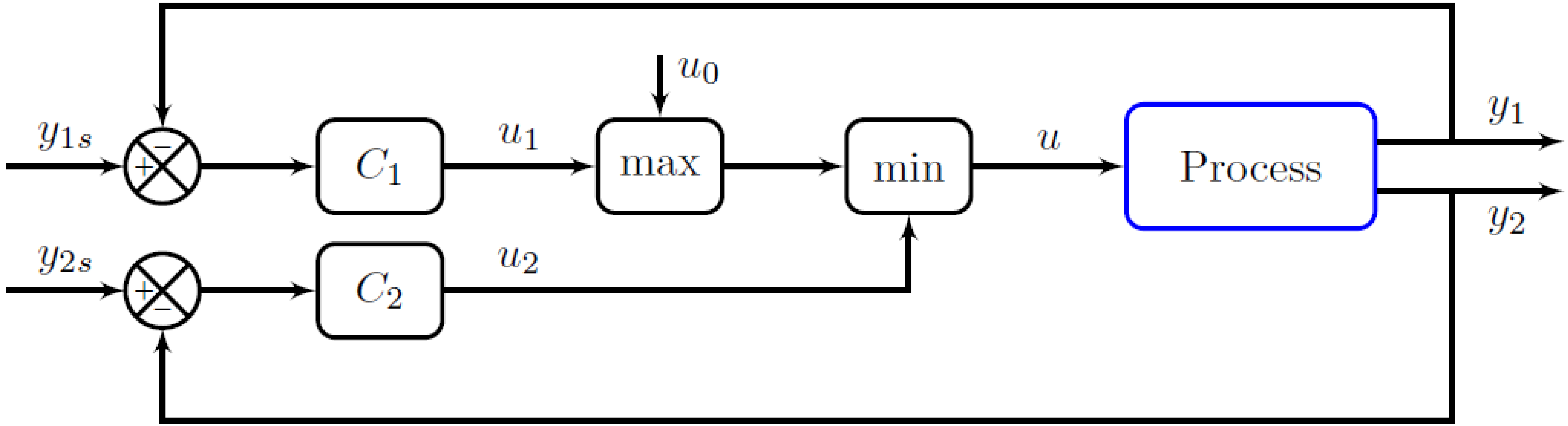


Figure 18: CV-CV switching for case with possibly conflicting constraints. In this case, constraint y_{1s} requires a max-selector and y_{2s}) requires a min-selector. The selector block corresponding to the most important constraint (here y_{2s}) should be at the end (Rule 2).

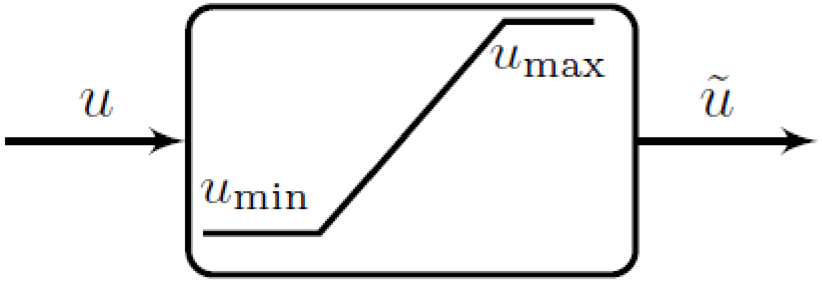
To understand the logic with selectors in series, start reading from the first selector. In this case, this is the max-selector: The constraint on y_1 is satisfied by a large value for u which requires a max-selector (Rule 1). u_0 is the desired input for cases when no constraints are encountered, but if y_1 reaches its constraint y_{1s}, then one gives up u_0. Next comes the min-selector: The constraint on y_2 is satisfied by a small value for u which requires a min-selector (Rule 1). If y_2 reaches its constraint y_{2s}, then one gives up controlling all previous variables (u_0 and y_1) since this selector is at the end (Rule 2). However, note that there is also a "hidden" max- and min- selector (Rule 3) at the end because of the possible saturation of u, so if the MV (input) saturates, then all variables (u_0, y_1, y_2) will be given up.

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).
- Another way to see this is to note that a valve works as a saturation element



The order of the "built-in" max- and min -selector in (8) does not matter because there is no possibility for conflict, as the two constraints (limits), u_{min} and u_{max}, cannot be active at the same time. However, in general, the order of the selectors does matter, and in cases of conflict, Rule 2 says that we should put the most important constraint at the end. Note that the "built-in" max- and min-selector

Question: Why doesn't order matter here?

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

Challenges selectors

- Standard approach requires pairing 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

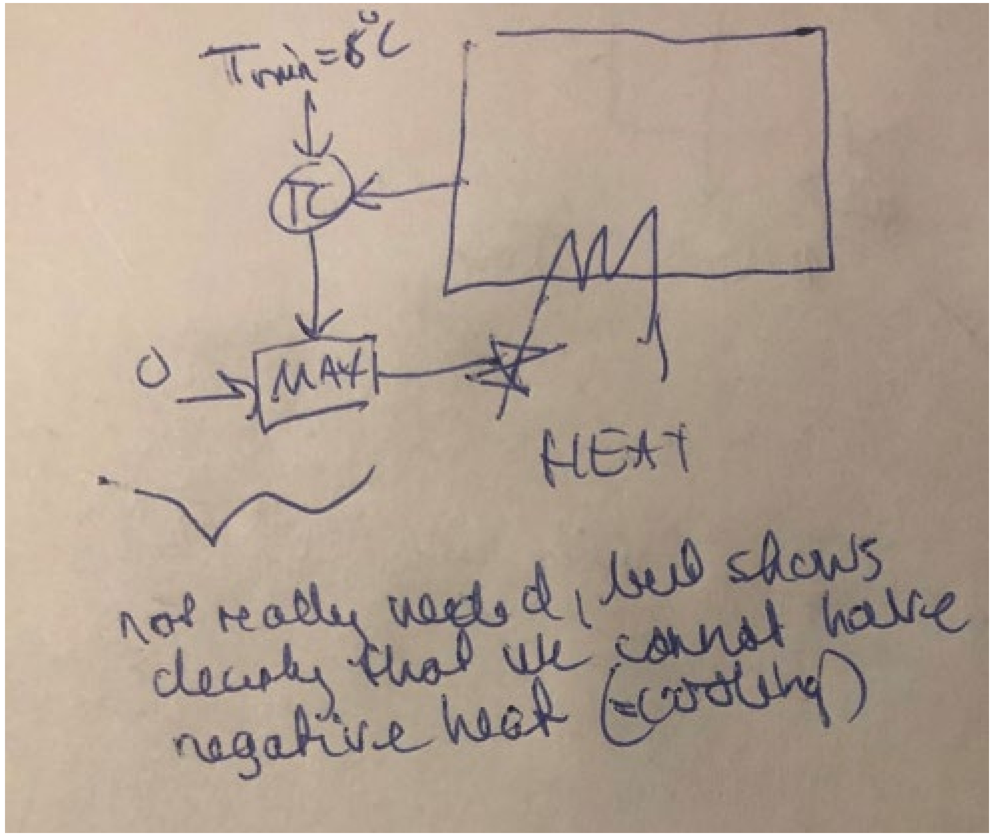
Avoid freezing in cabin

Minimize u (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.



- Actually, 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)*"

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

Minimize recycle (MV=z) subject to

CV= $F \geq F_{min}$

MV ≥ 0

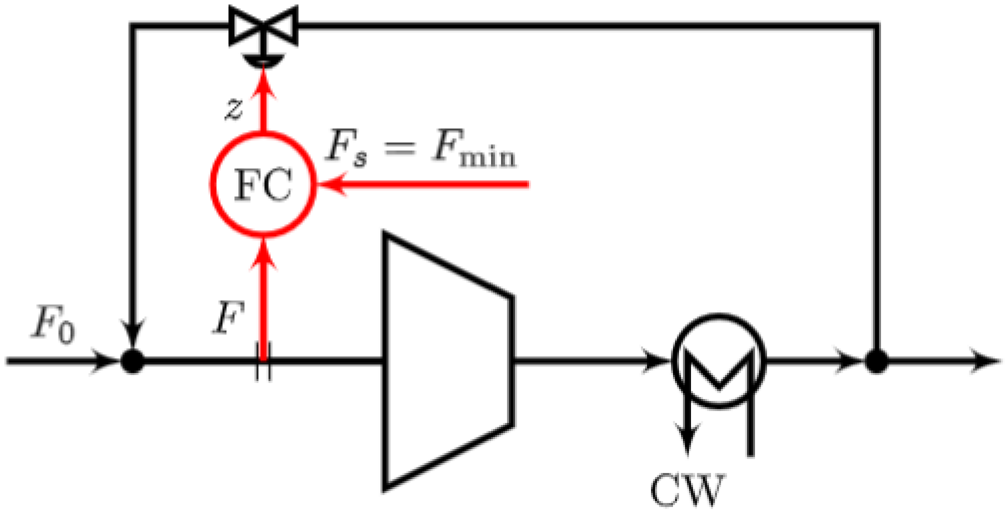


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding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

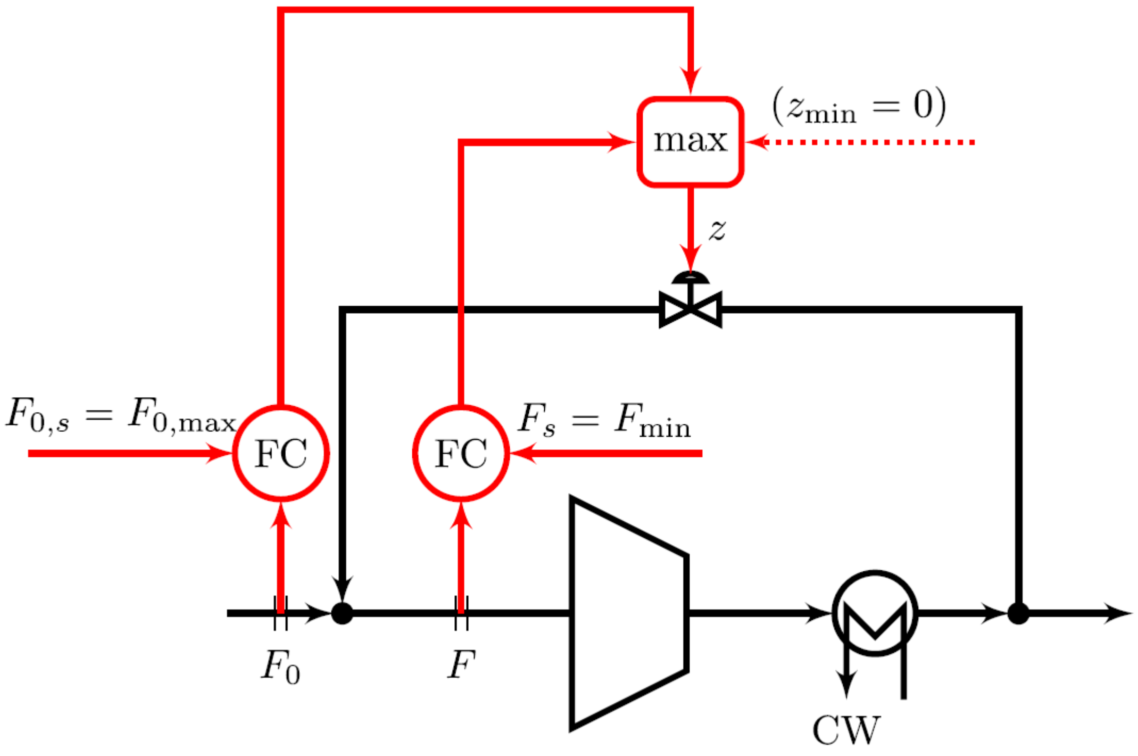
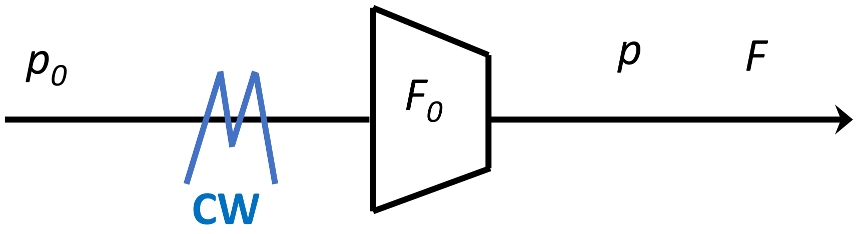


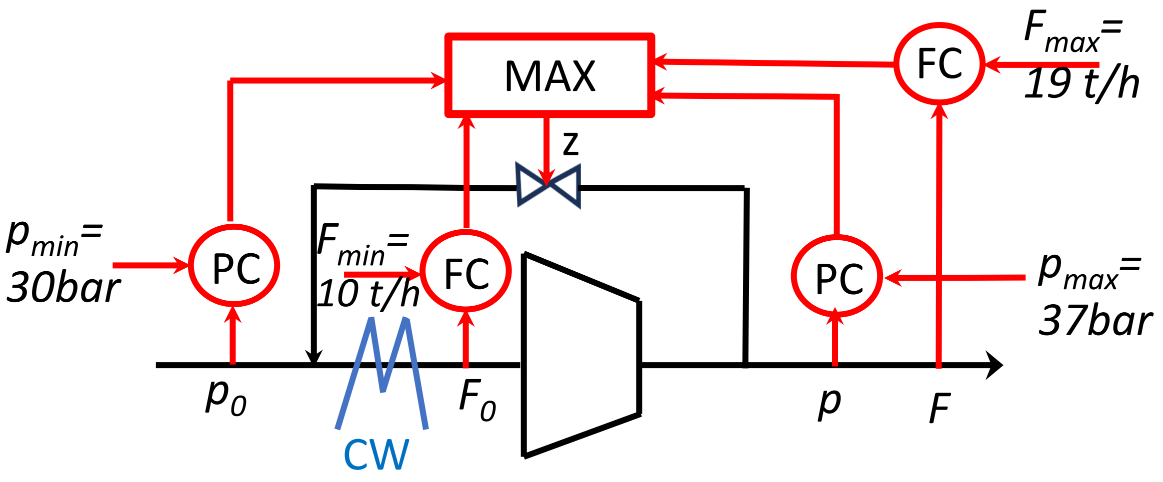
Fig. 33. Anti-surge compressor control with two CV constraints. This is an example of simple MV-CV-CV switching. MV = z, $CV_1 = F$, $CV_2 = F_0$ (all potentially active constraints).

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)*

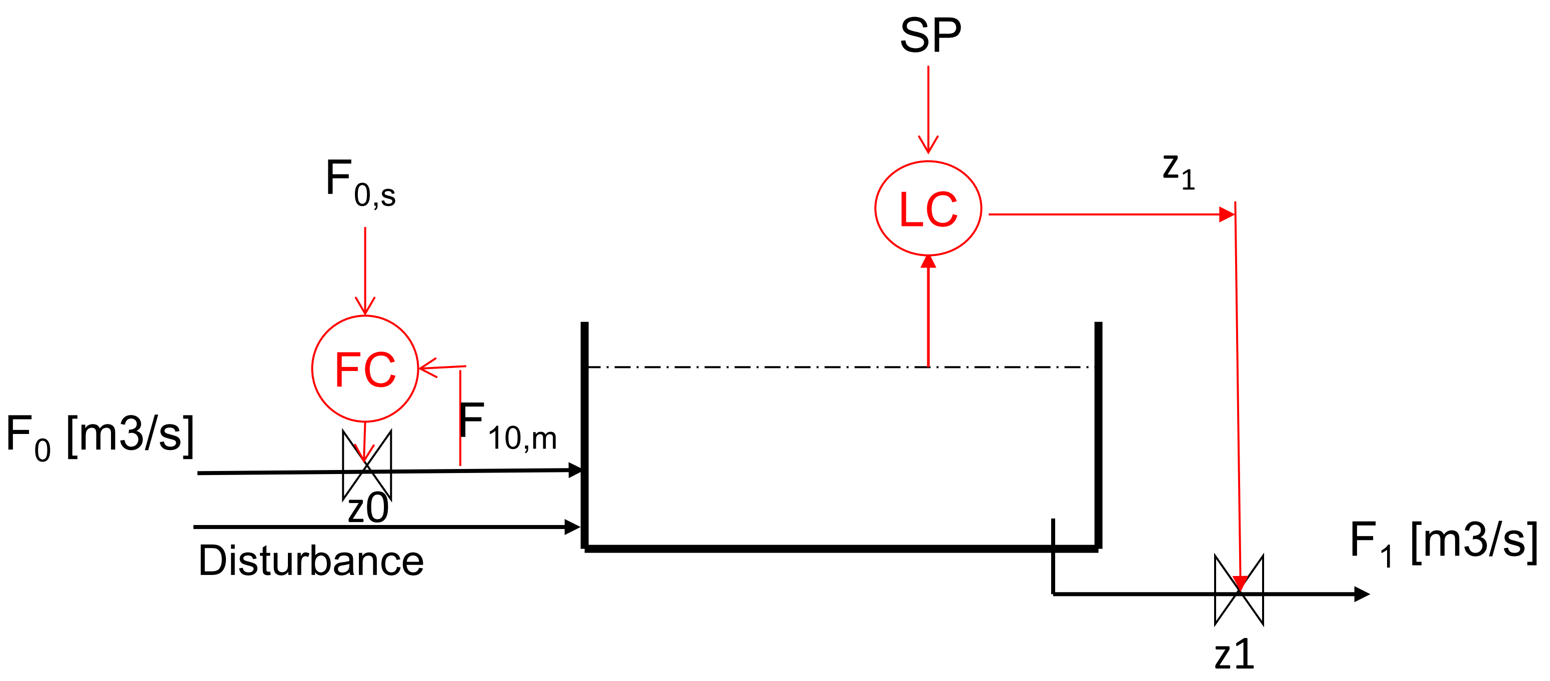


Rule CV-CV switching: Use max-selector for constraints that are satisfied by a large input (MV) (here: valve opening z)

D. Complex MV-CV switching

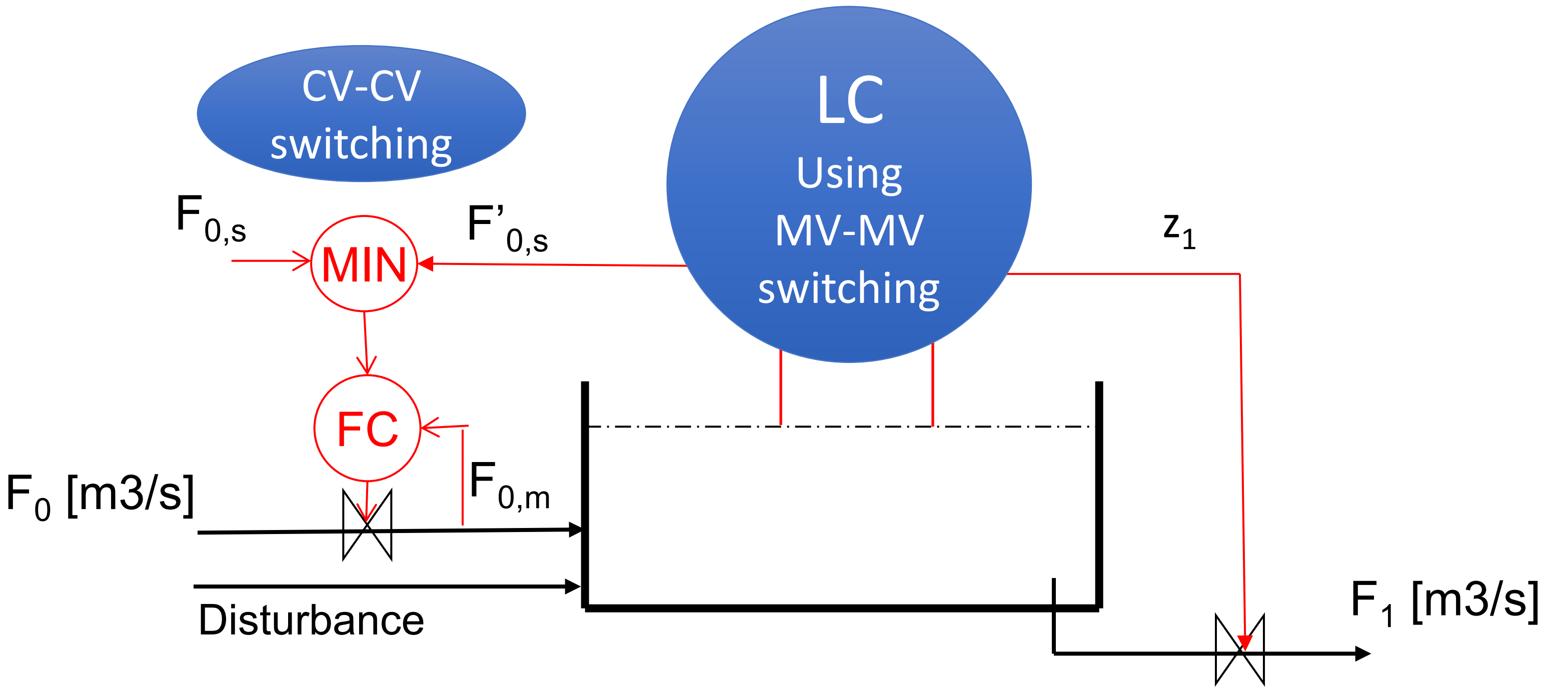
- Must combine MV-MV siwtching (3 options) with CV-CV swithing (selector)

Example : Level control



Problem: Lose level control of outflow valve saturate at fully open (z1=100%)

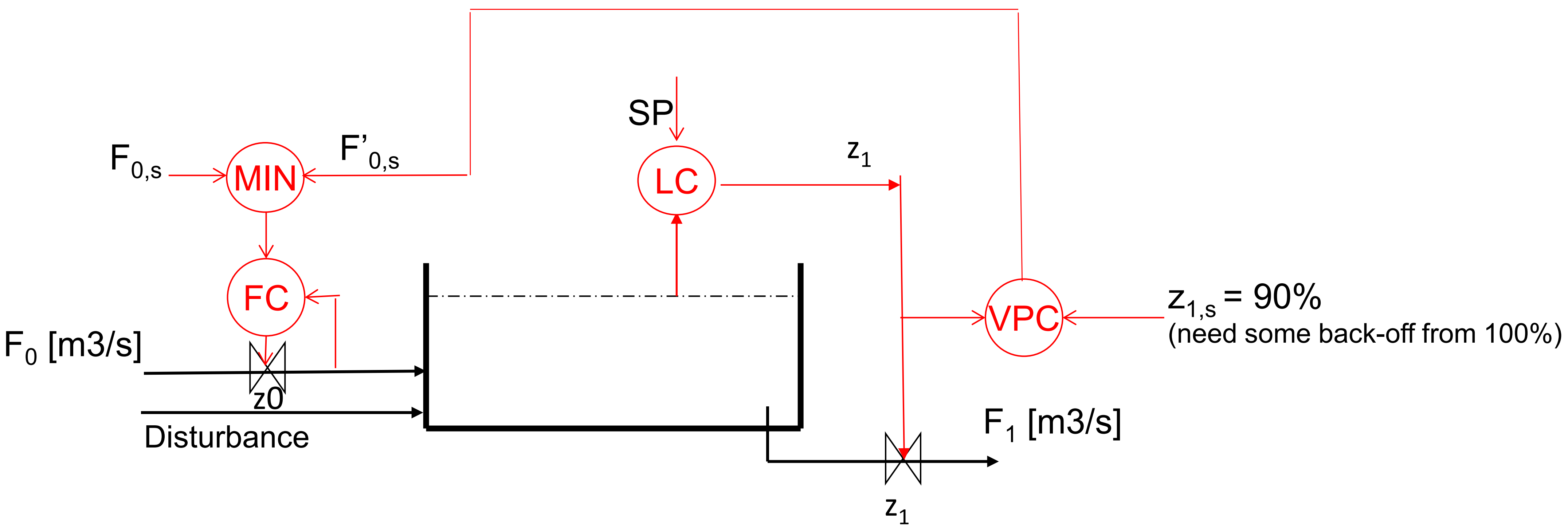
Solution: D. Complex MV-CV switching
(bidirectional inventory control)



Three alternatives for MV-MV switching

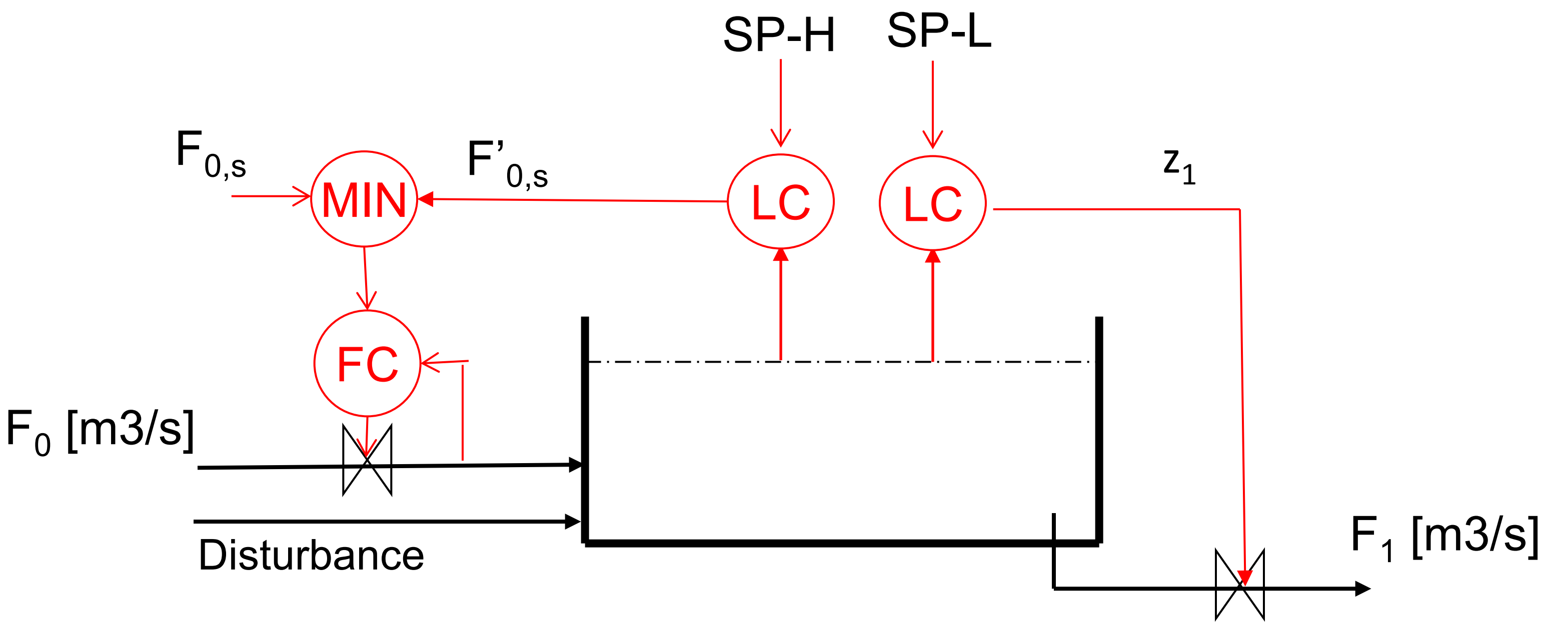
1. Split range control (problem since F_{0s} varies).
2. Split parallel control
3. VPC ("Long loop" for z1, backoff)

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)

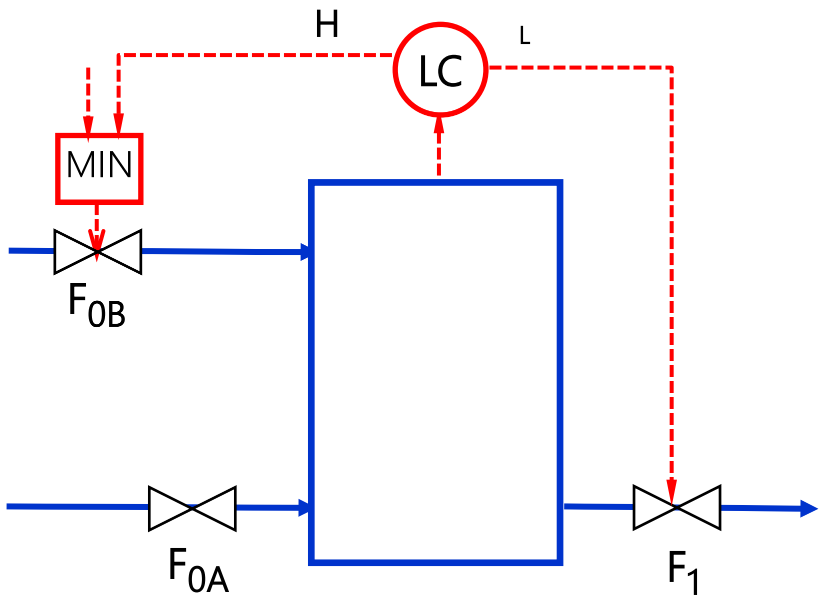


SP-L = low level setpoint = 50 % (or lower)
SP-H = high level setpoint = 60 % (or higher)

In addition: Use of two setpoints is good for using buffer dynamically!!

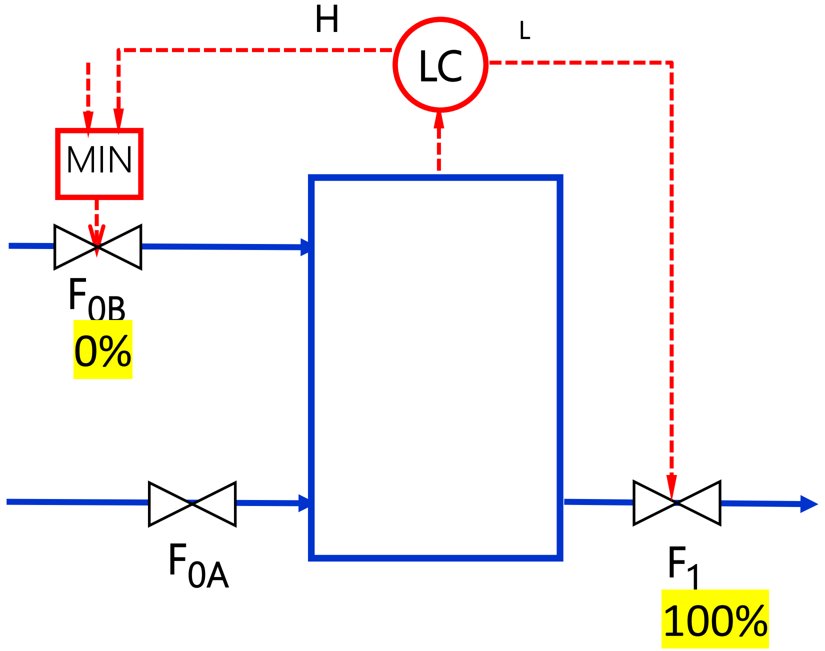
- Use low setpoint when level is controlled by product (outflow): Have room for feed if outflow stops temporarily.
- Use high setpoint when level is controlled by feed (inflow): Can keep producing if inflow stops temporarily.

Simplified representation



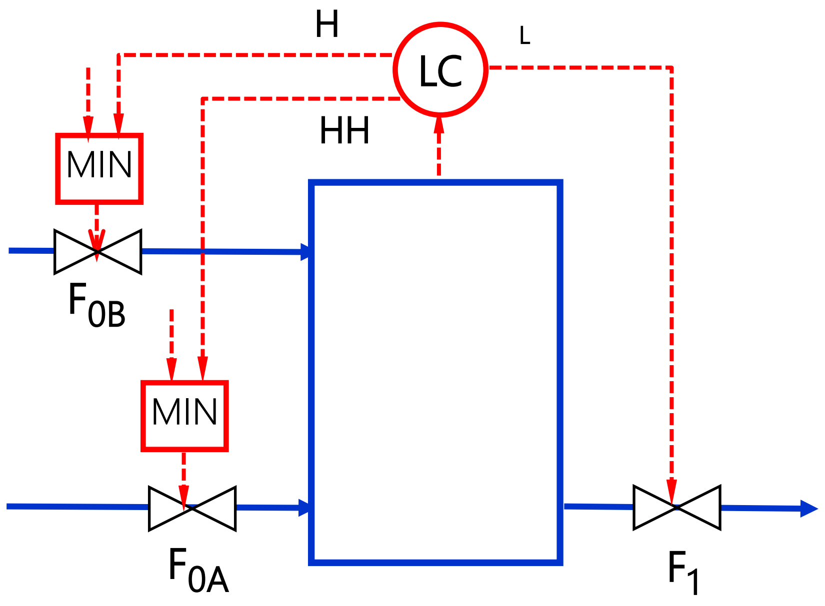
LC: Two controllers with different setpoints

What should we do if also F_{0B} saturates (at fully open)?



Level is at H=80% but keeps rising

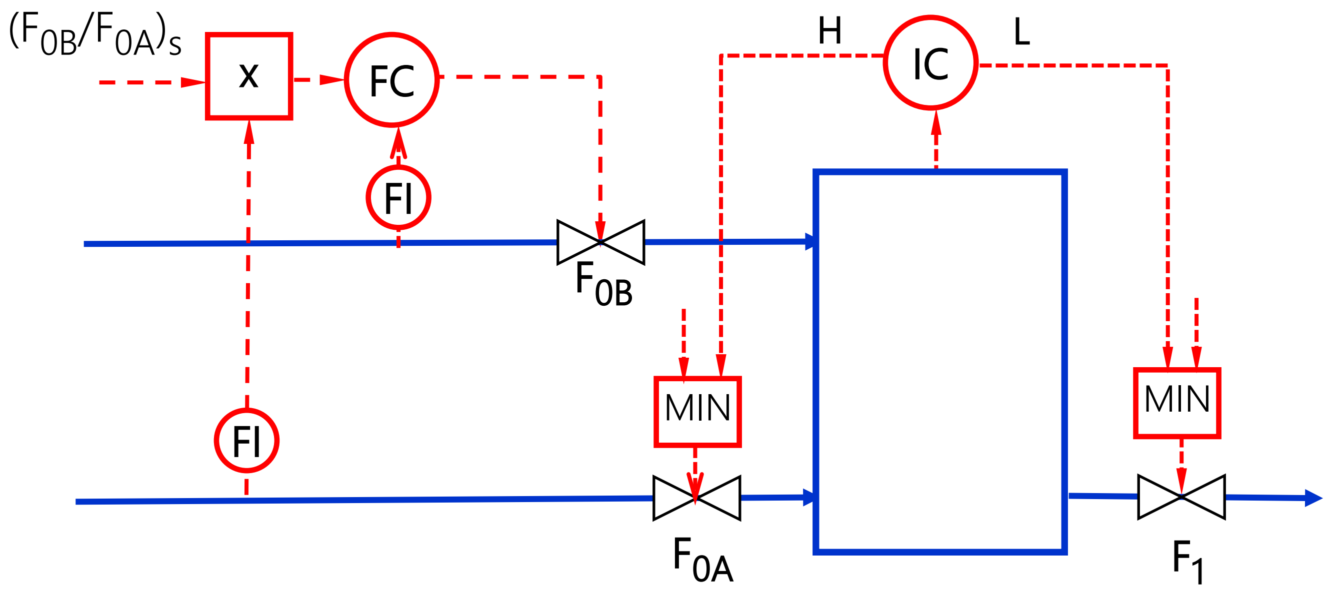
Solition: Another override (MV-CV switching)!



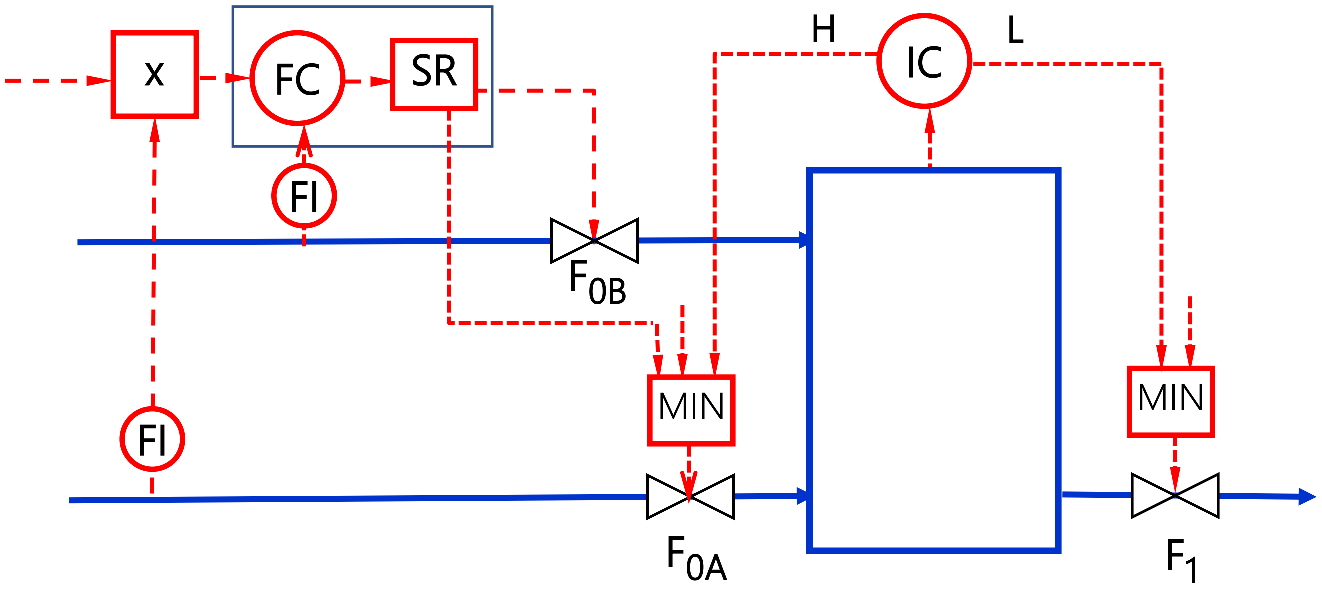
L=50%
H=80%
HH=90%

LC: Three controllers with different setpoints
When level reaches HH=90% we reduce F_{0A}

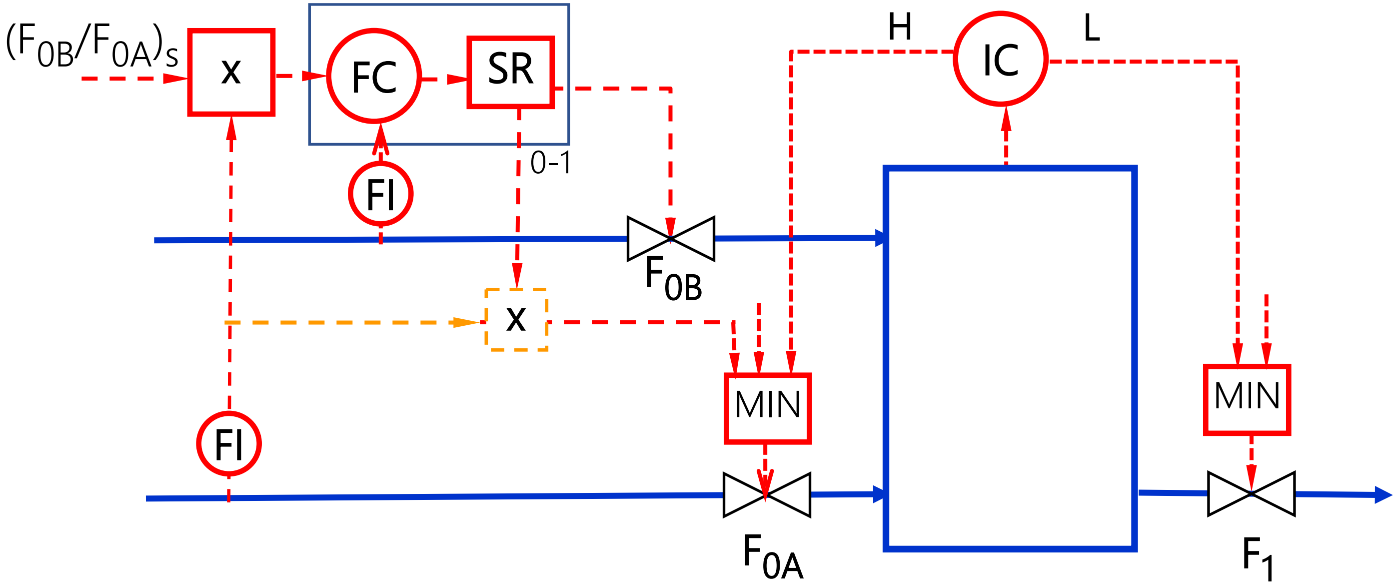
Alernative: Ratio control (don't need HH)



Add «dual» ratio control (if keeping ratio has top priority)



Trick to avoid delay (limbo) during transition



Summary: The four constraint switching cases

A. MV-MV switching (because MV may saturate)

- Need many MVs to cover whole steady-state range
- Use only one MV at a time
- Three options:

  A1. Split-range control,

  A2. Split-parallel control,

  A3. (Split) Valve position control (VPC)

B. CV-CV switching (because we may reach new CV constraint)

- Must select between CVs
- One option: Many controllers with Max-or min-**selector**

Plus the combination: MV-CV switching

C. **Simple** MV-CV switching: CV can be given up

- We followed «input saturation rule»
- Don't need to do anything (except anti-windup in controller)

D. **Complex** MV-CV switching: CV cannot be given up (need to «re-pair loops»)

- Must combine MV-MV switching (three options) with CV-CV switching (selector)

Note: we are here assuming that the constraints are not conflicting so that switching is possible

Adriana Reyes-Lua and Sigurd Skogestad, Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures, Ind.Eng.Chem.Res, 2020