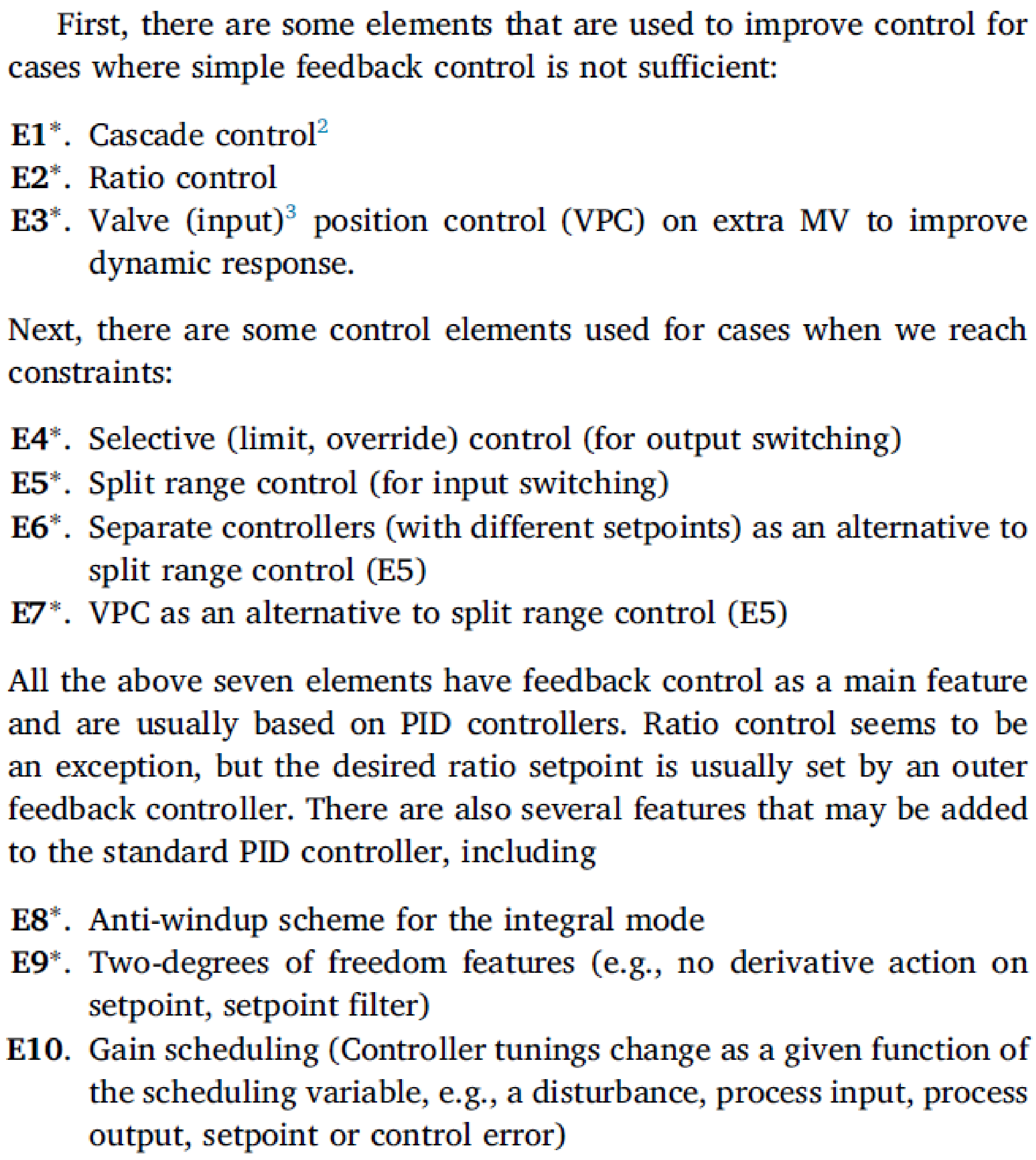
Part 3:
Standard control elements
Constraint switching.

Standard Advanced control elements



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

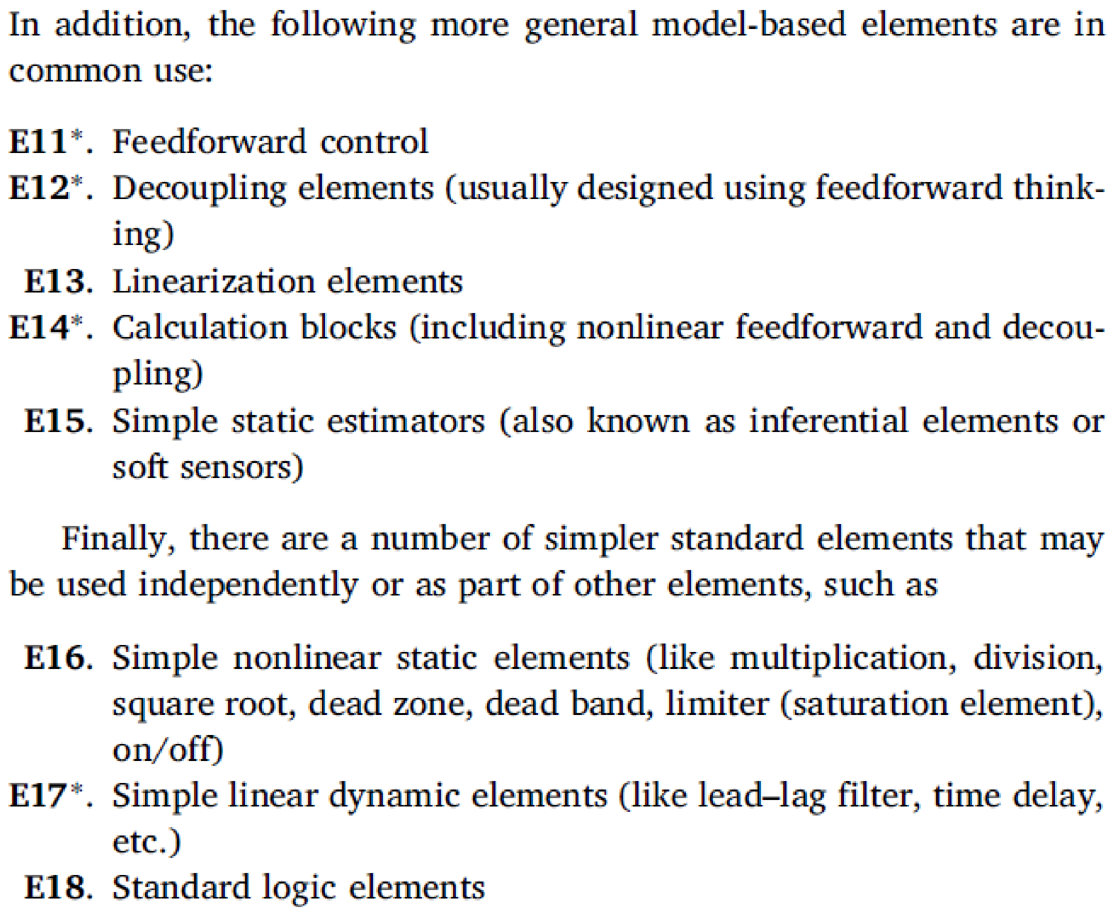
E1*. Cascade control[2]
E2*. Ratio control
E3*. Valve (input)[3] position control (VPC) on extra MV to improve dynamic response.

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

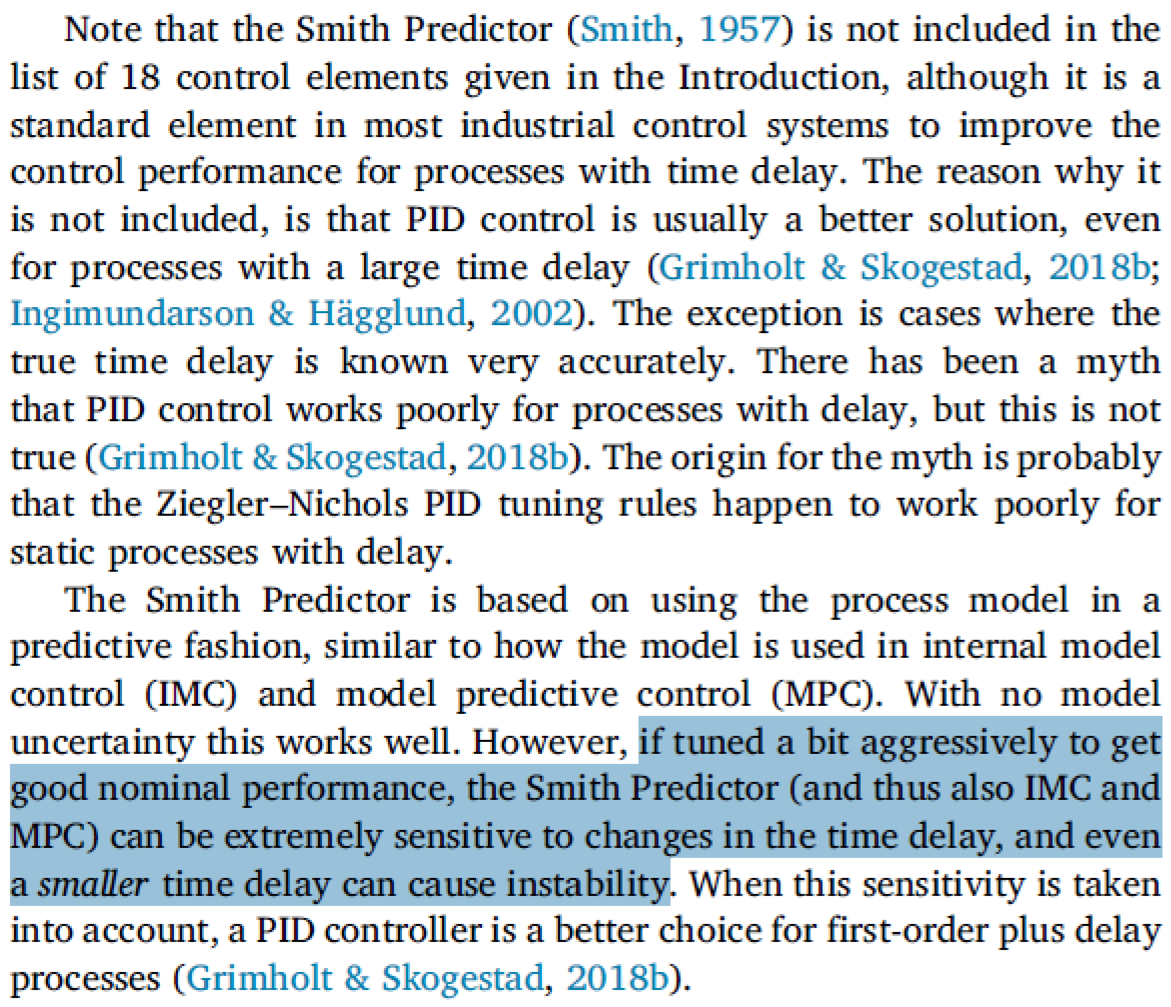
E16. Simple nonlinear static elements (like multiplication, division, square root, dead zone, dead band, limiter (saturation element), on/off)
E17*. Simple linear dynamic elements (like lead–lag filter, time delay, etc.)
E18. Standard logic elements

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!

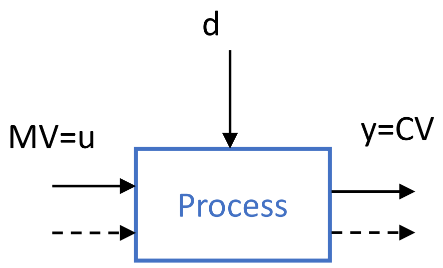


Note that the Smith Predictor (Smith, 1957) is not included in the list of 18 control elements given in the Introduction, although it is a standard element in most industrial control systems to improve the control performance for processes with time delay. The reason why it is not included, is that PID 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).

Introduction to pairing and switching

Most basic element: Single-loop PID control



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)*

Additional rule for interactive systems:

3. ***" RGA-rule"***
   - *Avoid pairing on negative steady-state RGA-element**.** Otherwise, the loop gain may change sign (for example, if the input saturates) and we get instability with integral action in the controller.*

Need to control active constraints
But active constraints may change during operation

Four cases:

- A. MV-MV switching
- B. CV-CV switching
- MV-CV switching
    - C. Simple (if we follow input saturation rule)
    - D. Complex (combine MV-MV and CV-CV)

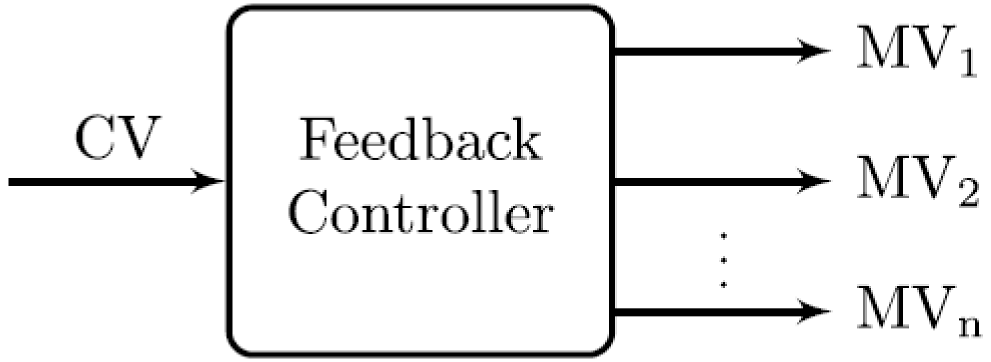


Fig. 5. MV-MV switching is used when we have multiple MVs to control one CV, but only one MV should be used at a time. The block "feedback controller" usually consists of several elements, for example, a controller and a split range block.

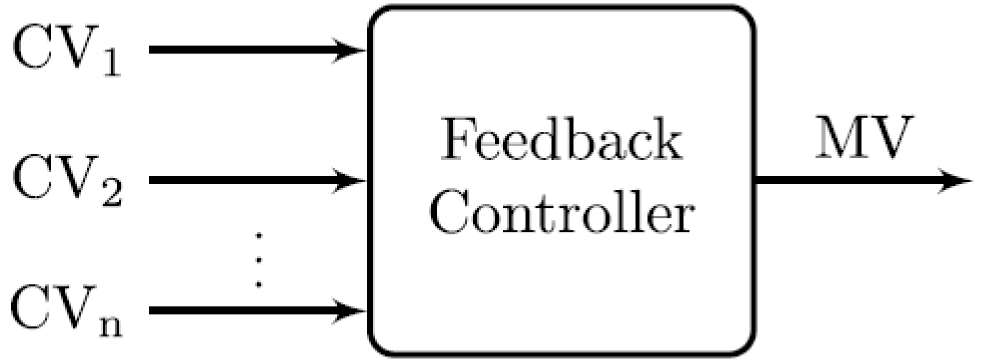


Fig. 6. CV-CV switching is used when we have one MV to control multiple CVs, but the MV should control only one CV at a time. The block "feedback controller" usually consists of several elements, typically several PID-controllers and a selector.

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 (SRC) (one controller)

Alt 1': Generalized SRC (many controllers)

Alt.2 Many controllers with different setpoints

Alt.3 Valve position control

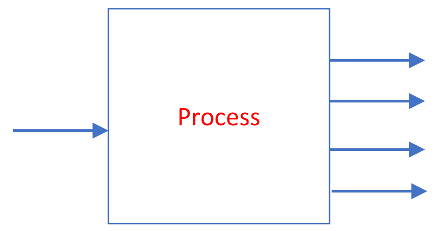
In addition: MPC

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)

B. CV-CV switching

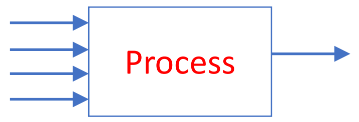


- One MV
- Many CVs, but control only one at a time
- Solution: Selector

The four cases in more detail

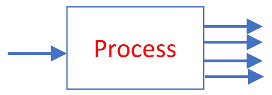
A. MV-MV switching (because MV may saturate)

- Need many MVs to cover whole steady-state range
- Use only one MV at a time
- Three options:
  - A1. Split-range control,
  - A2. Different setpoints,
  - A3. 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**



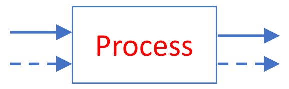
Plus the combination: MV-CV switching

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)



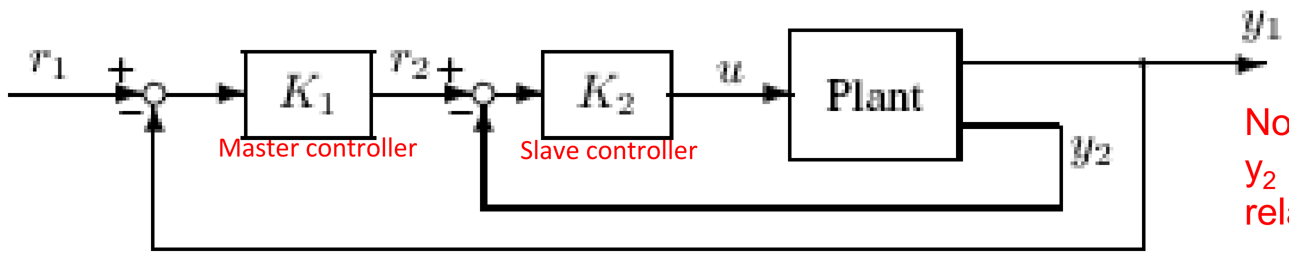
Note: we are here assuming that the constraints are not conflicting so that switching is possible

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")

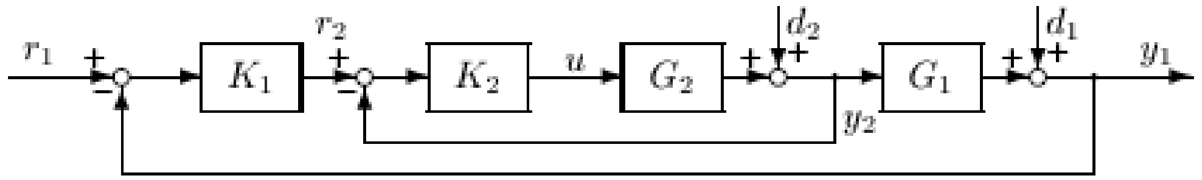
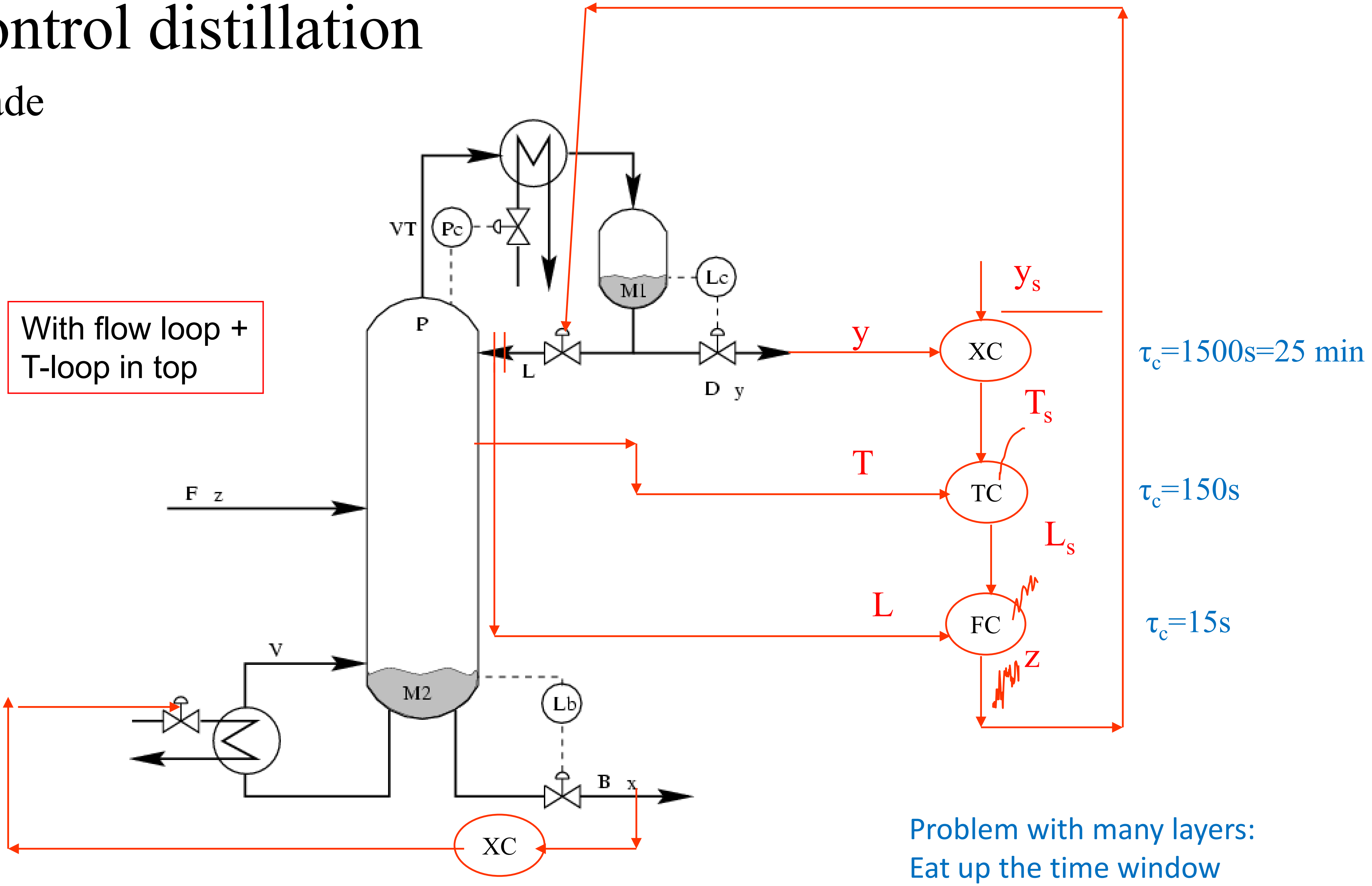


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

Cascade control distillation

3 layers of cascade



With flow loop + T-loop in top

τ_c=1500s=25 min

τ_c=150s

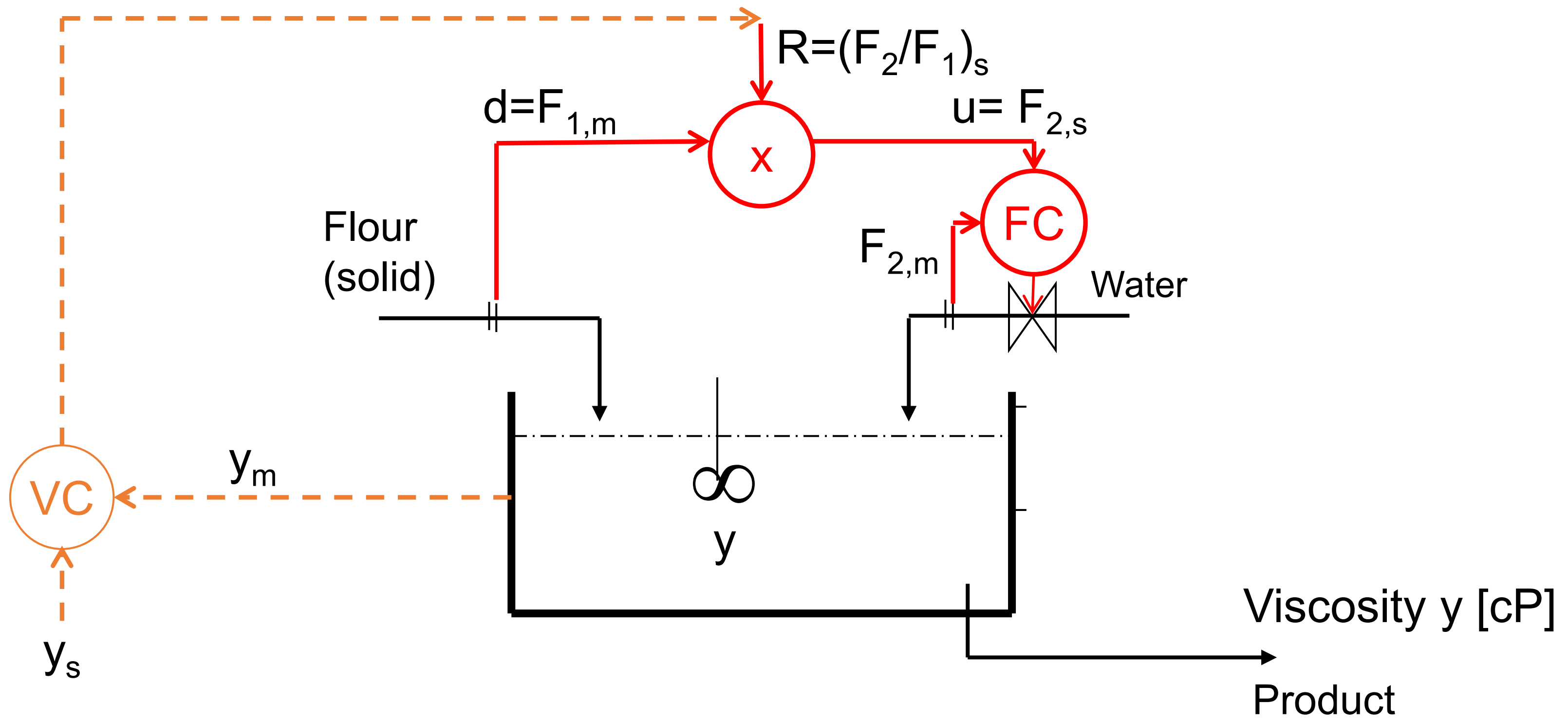
τ_c=15s

Problem with many layers: Eat up the time window

E2. Ratio control

EXAMPLE: CAKE BAKING MIXING PROCESS

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



Ratio control

- Avoid divisions in implementation! (avoid divide by 0)
- Process control textbooks has some bad/strange suggestions, for example, division (bad) and "ratio stations" (complex):

Seborg:

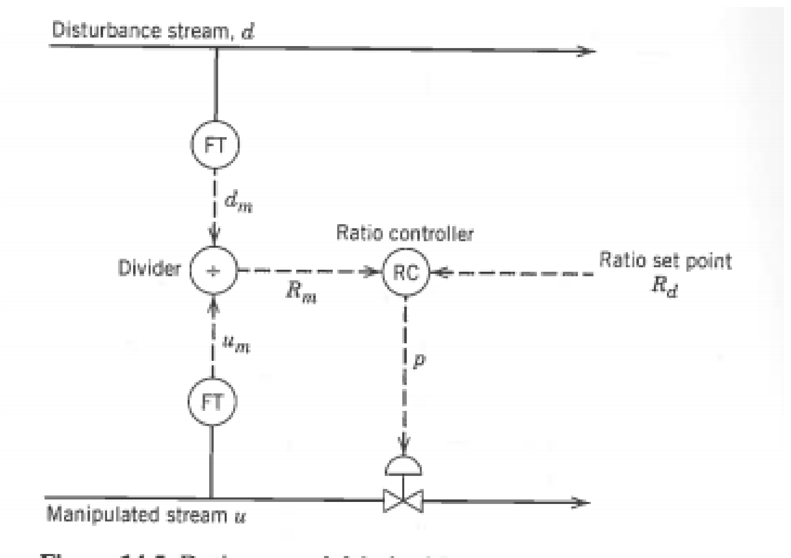


Figure 14.5 Ratio control, Method I.

Bad solution

Avoid divisions (divide by 0 if u =0, for example, at startup)

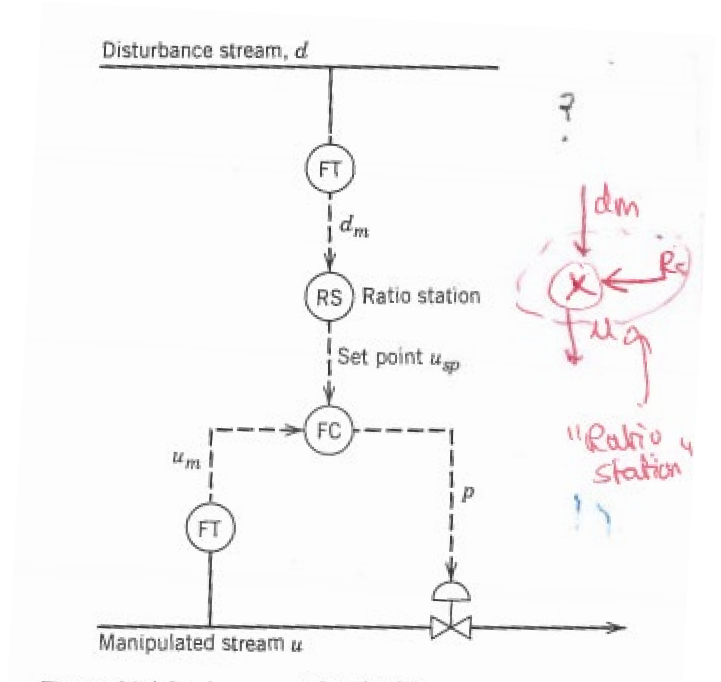


Figure 14.6 Ratio control, Method II.

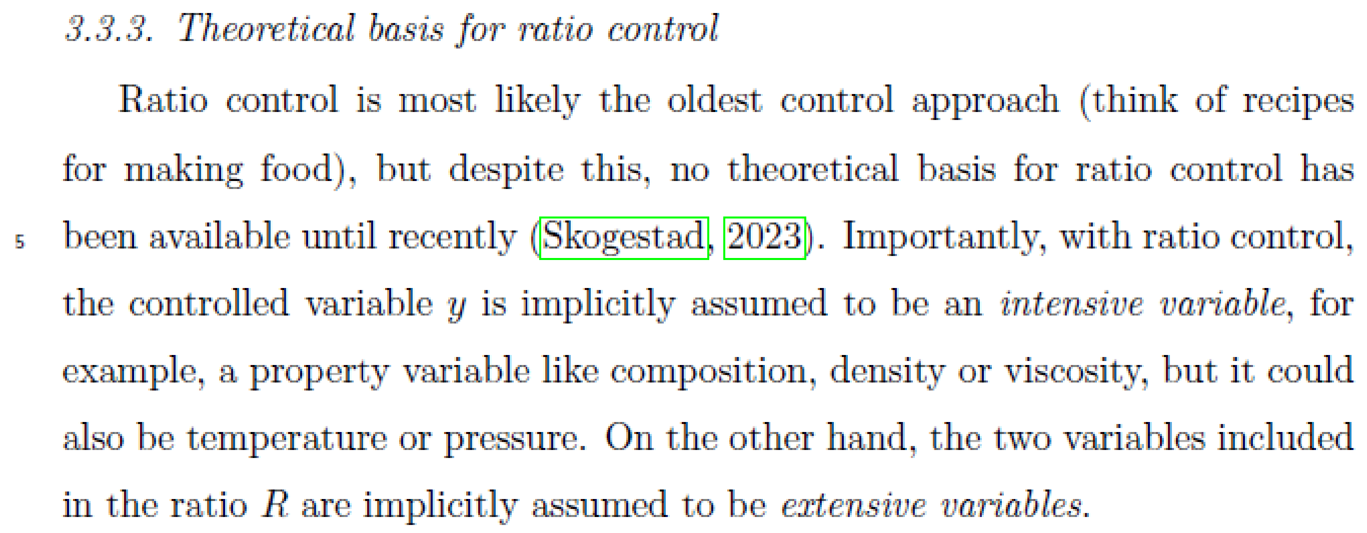
This is complicated. What is RS?

Ok if implemented as shown in red at right

Ratio control

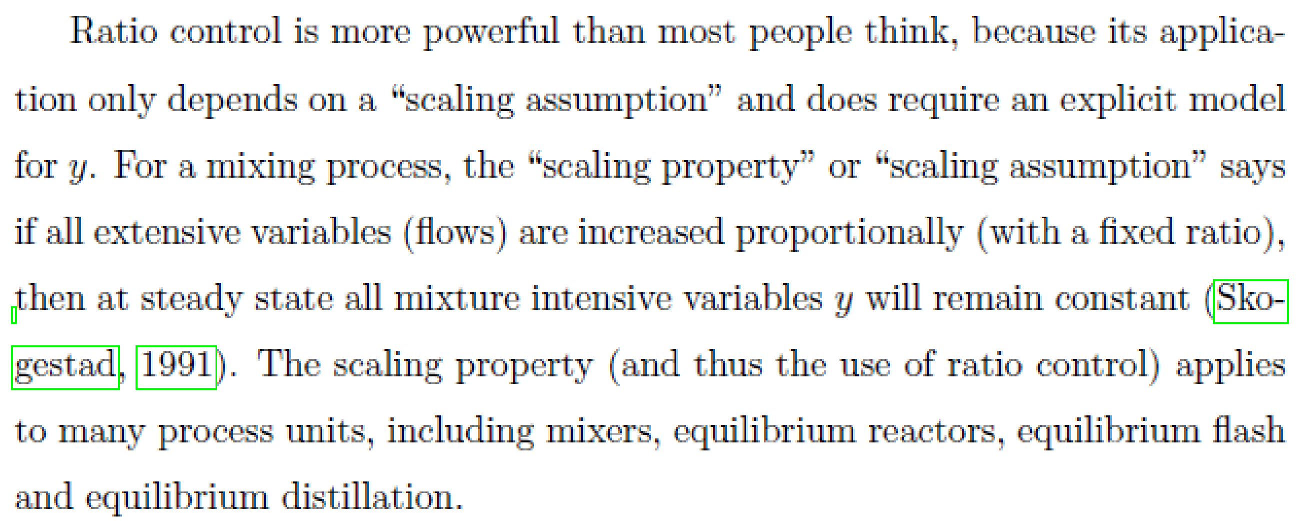
- Keep ratio R (between extensive variables) constant in order to keep property y constant
  - Feedforward: R=u/d
  - Decoupling: $R=u_1/u_2$
    - u,d: extensive variables
  - y: (any!) intensive variable
- Don’t really need a model (no inverse as in «normal» feedforward!)
- Assumes that «scaling property» holds
  - Based on physical insight
  - 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)

Theoretical basis of ratio control



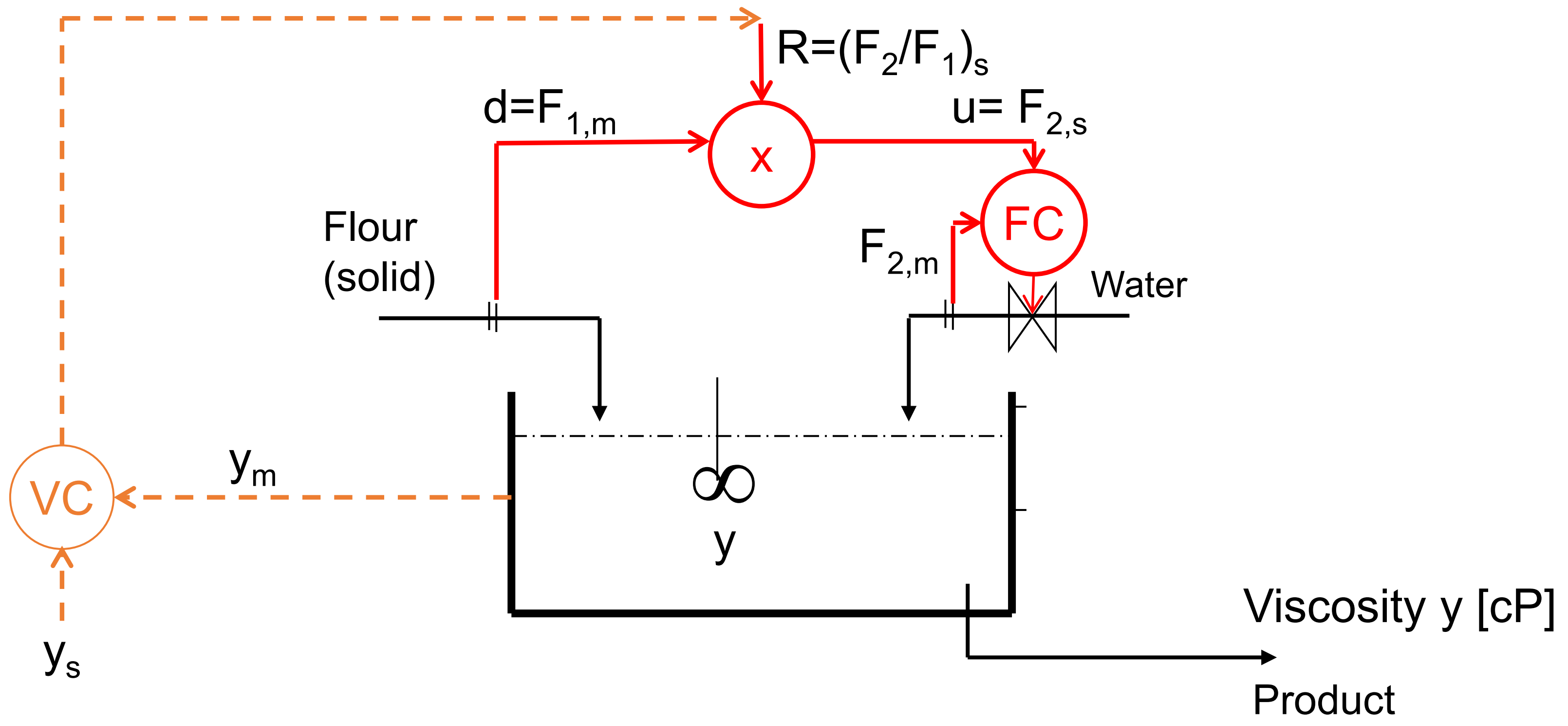
3.3.3. Theoretical basis for ratio control

Ratio control is most likely the oldest control approach (think of recipes for making food), but despite this, no theoretical basis for ratio control has been available until recently (Skogestad, 2023). Importantly, with ratio control, the controlled variable y is implicitly assumed to be an *intensive variable*, for example, a property variable like composition, density or viscosity, but it could also be temperature or pressure. On the other hand, the two variables included in the ratio R are implicitly assumed to be *extensive variables*.



Ratio control is more powerful than most people think, because its application only depends on a "scaling assumption" and does require an explicit model for y. For a mixing process, the "scaling property" or "scaling assumption" says if all extensive variables (flows) are increased proportionally (with a fixed ratio), then at steady state all mixture intensive variables y will remain constant (Skogestad, 1991). The scaling property (and thus the use of ratio control) applies to many process units, including mixers, equilibrium reactors, equilibrium flash and equilibrium distillation.

LINEARITY OF RATIO CONTROL when done correctly



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 $d=F_1$.

Proof of last statement

- "Note : <u>This way</u> of implementing ratio control makes it easy to tune the outer feedback loop (CC: composition controller) because the gain from MV = $(q2/q1)_s$ to CV=c does not depend on disturbances in q1."
- One may think that the last statement is fairly obvious, because we are talking about just scaling all flowrates by the same factor and then the composition c should remain constant. But actually, I wrote the following in 2021 (and earlier).

  WRONG: "Potential problem for outer feedback loop (CC: composition controller): Gain from MV = $(q2/q1)_s$ to CV=c will vary because of multiplication with $q_{1,m}$. So outer loop must have robust tunings to get high gain margin (large tauc)".
- In fact, it's opposite, there are less gain variations when the outer controller manipulates $(q_2/q_1)_s$ than when it manipulates q_2 directly
- ***Proof.*** The component balance gives: $CV=c=(c_1q_1 + c_2q_2)/(q_1+q_2)$
- We are here considering disturbances in q1, so assume that c1 and c2 are constant.
- We also assume that there is an outer loop so that c remains constant. From the component balance we see that c=constant implies that at as we change q1 (disturbance) we will have that q1/q2=constant and also that R_1=q1/(q1+q2) = constant.
- With no ratio control: The gain from MV=q_2 to CV=c is:
  - K = (c2-c1)q1/(q1+q2)^2 = (c2-c1)R1/(q1+q2)
  - From the above argument K = constant/(q1+q2) so the gain K will change with operation, which will be a problem for the outer feedback controller (CC). Actually, we find that K=infinity when q=q1+q2 goes to zero, so we may get instability in the outer feedback loop at low flowrates.
- With ratio control: The gain from MV=$(q_2/q_1)_s$ to CV=c is:
  - K_r = (c2-c1)q1^2/(q1+q2)^2 = (c2-c1) R_1^2
  - From the above argument we have that R1=constant so we get Kr= constant independent of the value of the disturbance (q1)! So the outer loop always has the same gain and there no reason to be careful about the tunings.
- Note: An alternative to ratio control is "standard" feedforward control where u = u_{FB} + u_{FF} (where FB is from the feedback controller CC and FF is from a feedforward controller from d=q1.) In this case we get the problem with process gain variation for the feedback controller CC). So ratio control is the best!
- But note that we should not always use ratio control for flow disturbances; it only holds if you are controlling temperature or composition (which are intensive variables). If you are controlling an extensive variable like total flow or level then you should add or subtract the disturbance. To the right an example:
- Challenge to myself (Sigurd): prove this more generally using theory of 1) ratio control and 2) input transformations.

Chapter 15

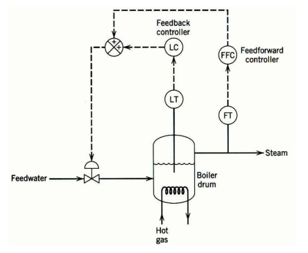


Figure 15.4 The feedfoward-feedback control of the boiler drum level.

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
  - MV-MV switching: Need several MVs to cover whole range at steady state
  - We want to use one MV at a time

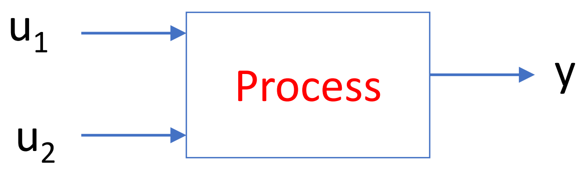
E3. VPC on extra dynamic input

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)

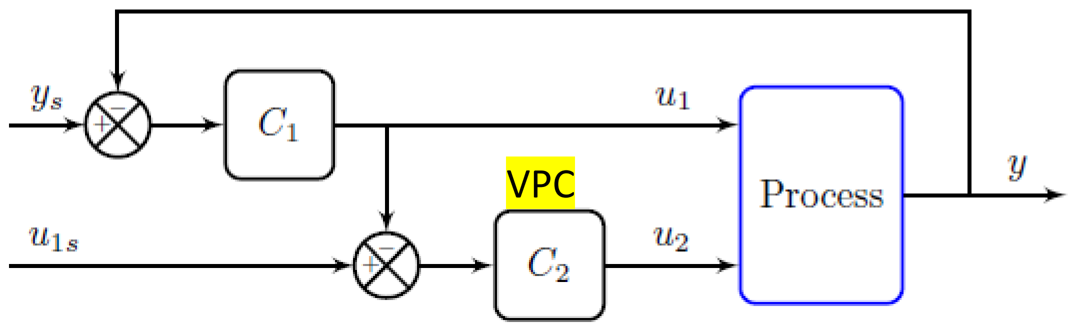


Figure 12: Valve (input) position control (VPC) for the case when an "extra" MV (u_1) is used to improve the dynamic response. A typical example is when u_1 is a small fast valve and u_2 is a large slower valve.
C_1 = fast controller for y using u_1.
C_2 = slow valve position controller for u_1 using 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

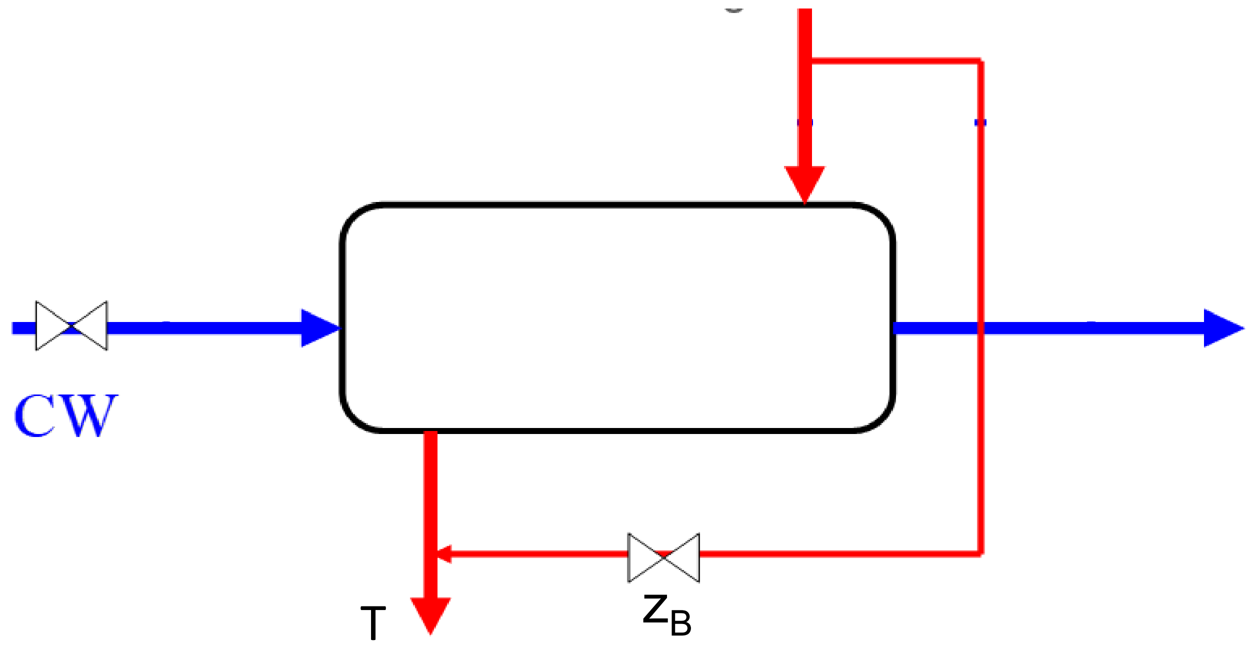
Example 2: Strong base (u_2) and weak base (u_1) for neutralizing acid (disturbance) to control y=pH

- Do pH change gradually (in two tanks) with the strong base (u_2) in the first tank and the weak base (u_1) in the last tank. u1 controls the pH in the last tank (y)

Alternative term for dynamic VPC:

- Mid-ranging control (Sweden)

Example: Heat exchanger with bypass

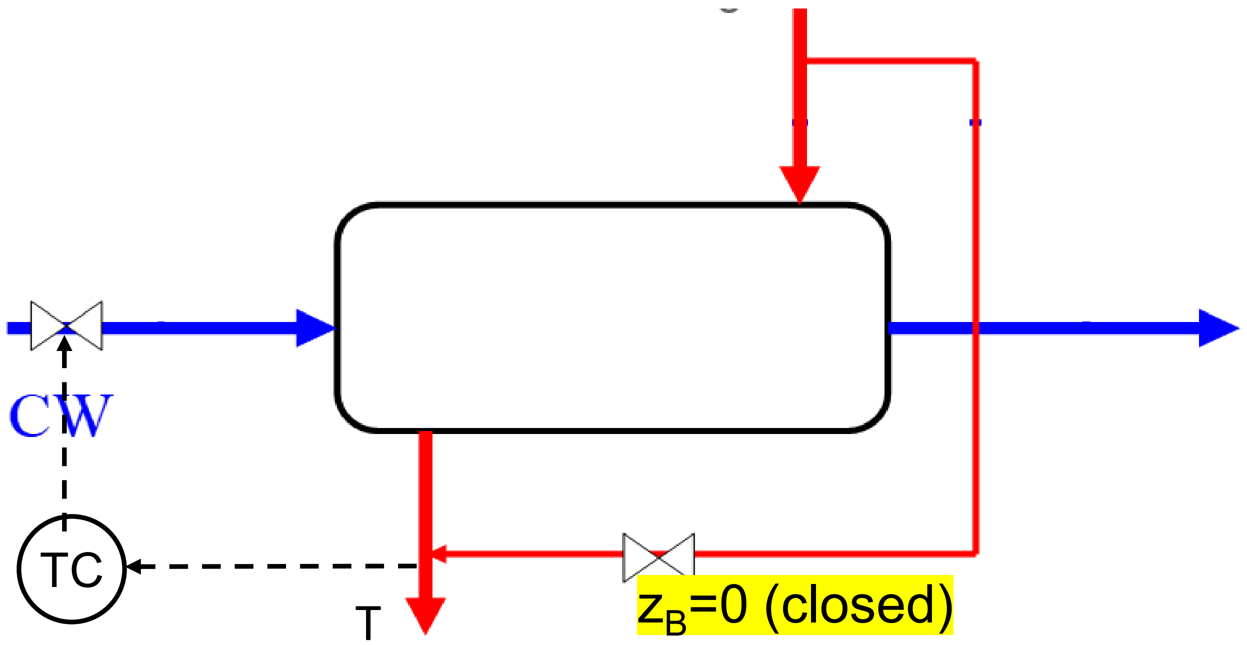


Want tight control of y=T.

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

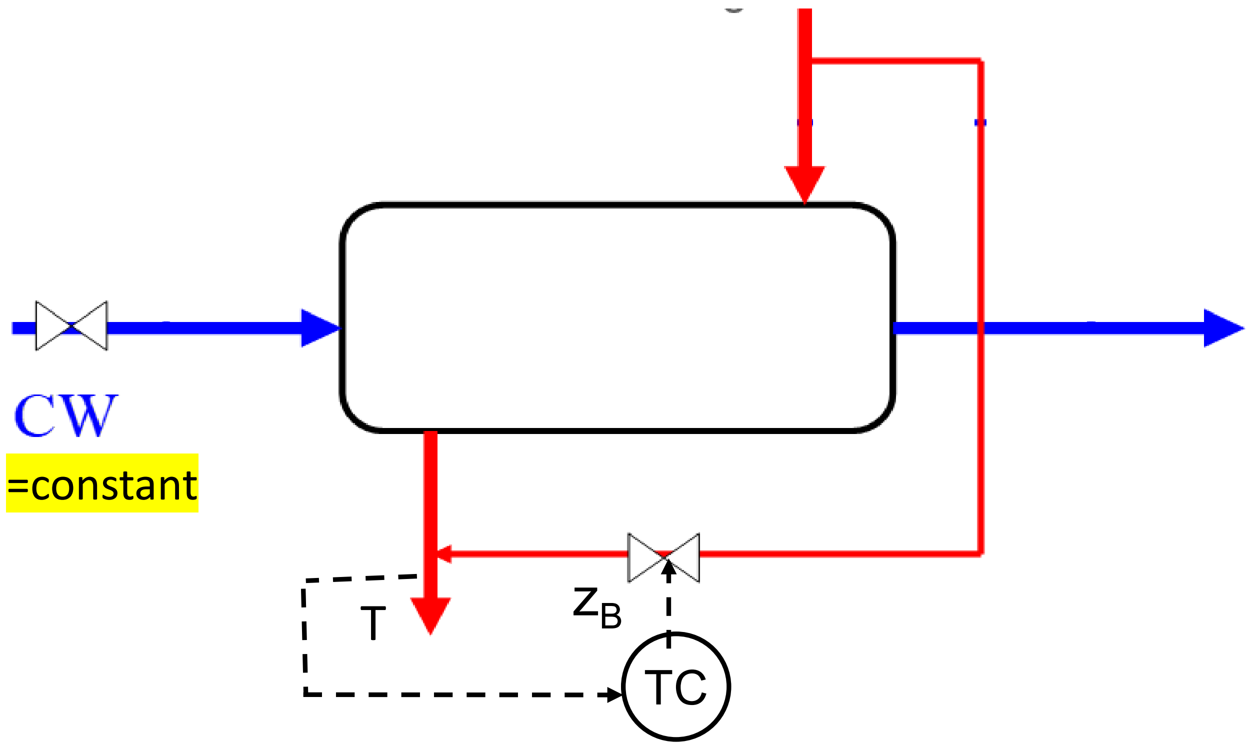
Proposed control structure?

Attempt 1. Use u_2=cooling water: TOO SLOW



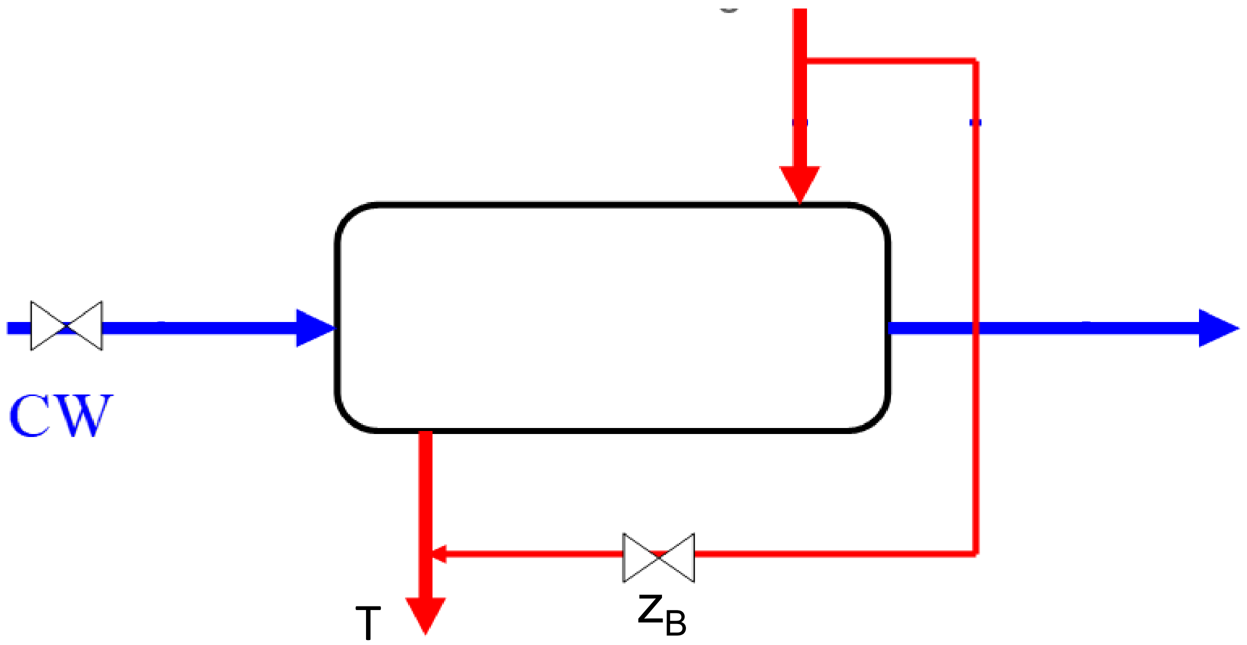
Attempt 2. Use $u_1=z_B$=bypass. SATURATES

(at z_B=0=closed if CW too small)



Advantage: Very fast response (no delay)
Problem: z_B is too small to cover whole range
+ not optimal to fix at large bypass (waste of CW)

What about VPC?

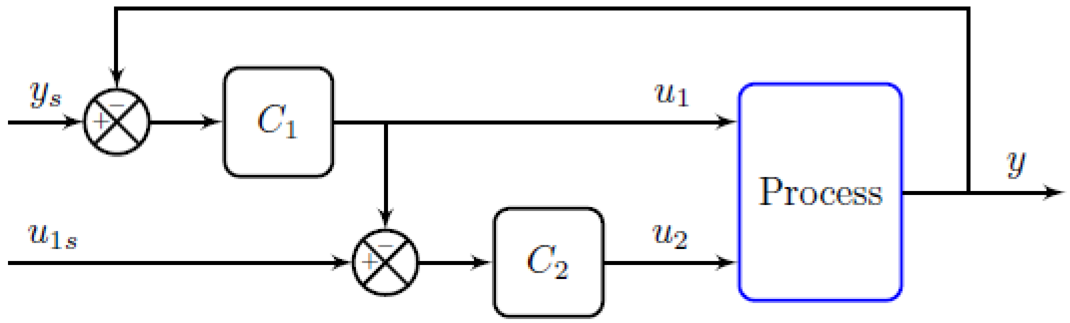


Want tight control of $y=T$.

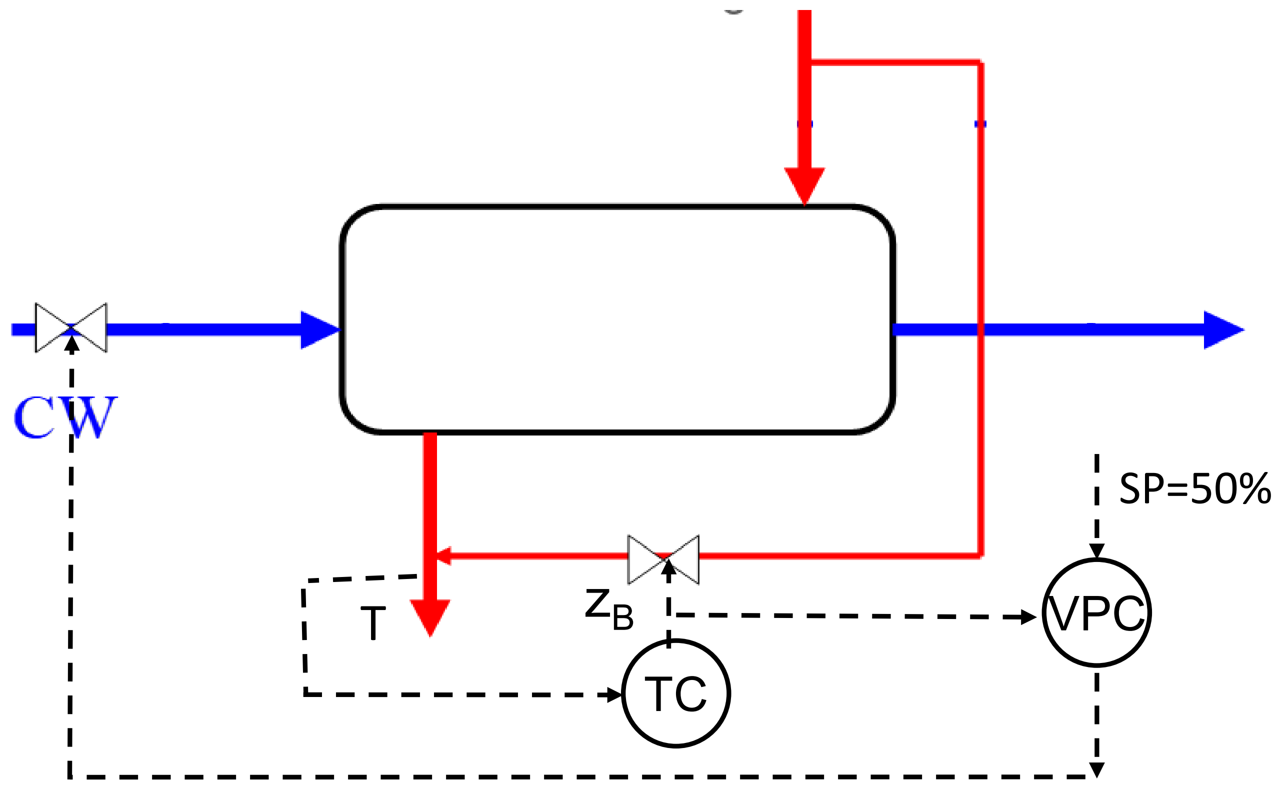
- $u_1= z_B$
- $u_2=CW$

Proposed control structure?

- Main control: $u_2=CW$
- Fast control: $u_1=z_B$



Attempt 3 (proposed): VPC



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

Comment on heat exchanger example

- The above example assumes that the flows on the two sides are «balanced» (mc_p for cooling water (CW) and hot flow (H) are not too different) such that both the bypass flow (u1) and CW flow (u2) have an effect on T (CV)

- There are two «unbalanced» cases:
  - If CW flow is small, then T-outCW will always approach T-inH, so from a total energy balance, the bypass will have almost zero *steady-state* effect on T.
  - If CW flow is large, then T-outH (before bypass mixing point) will always approach T-inCW, so CW will have almost zero effect on T. (*both* steady state and dynamically)

- This illustrates that heat exchanger may behave very nonlinearly, and a good control structure for one heat exchanger case, may not work well for another case

Alternative to VPC: Parallell control

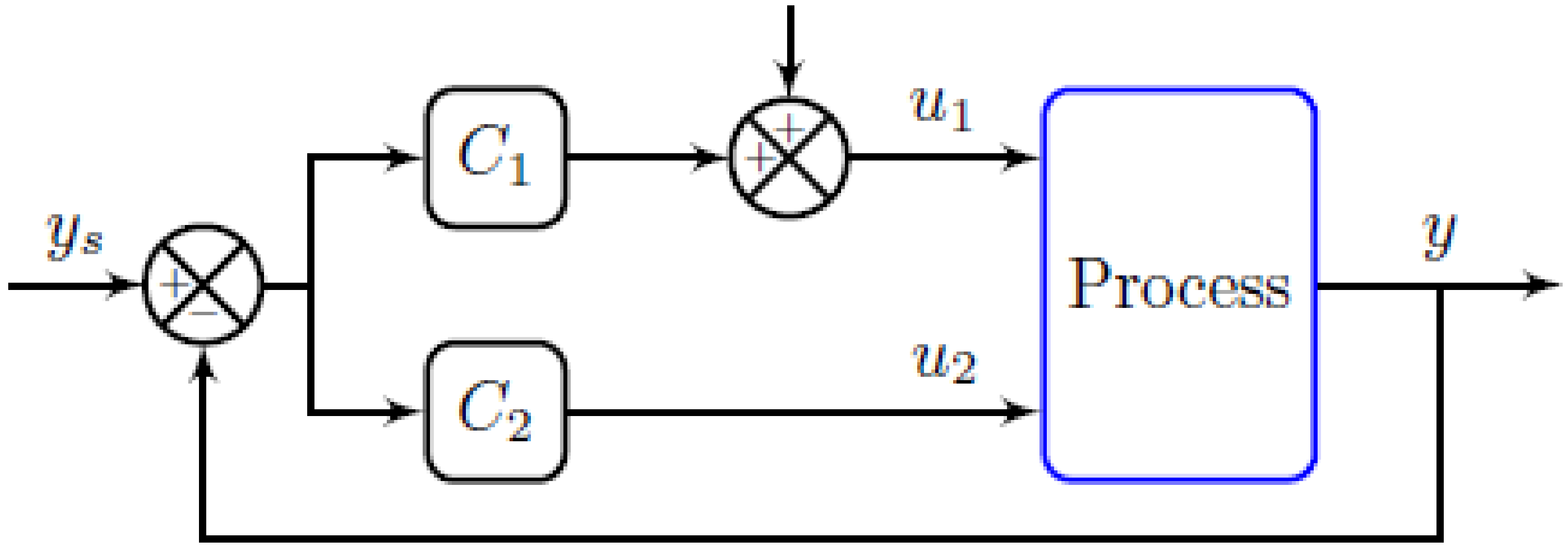


Figure 13: Parallel control to improve dynamic response - as an alternative to the VPC solution in Figure 12.
The "extra" MV (u_1) is used to improve the dynamic response, but at steady-state it is reset to u_{1s}. The loop with C_2 has more integral action and wins a steady state.

The advantage with valve position control compared to parallel control is that the two controllers in Figure 12 can be tuned independently (but C_1 must be tuned first) and that both controllers can have integral action. On the other hand, with some tuning effort, it may be easier to get good control performance for y with parallel control.

VPC with one MV: Stabilizing control with resetting of MV

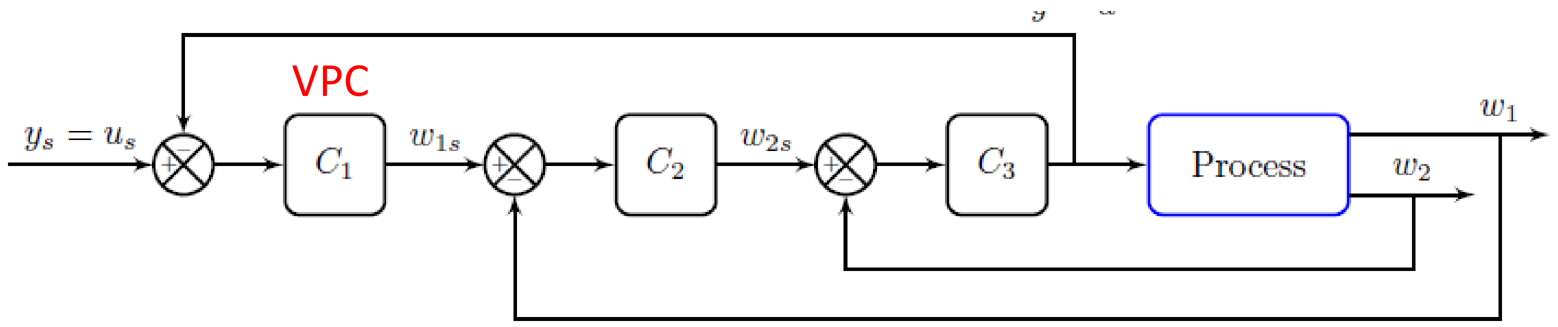
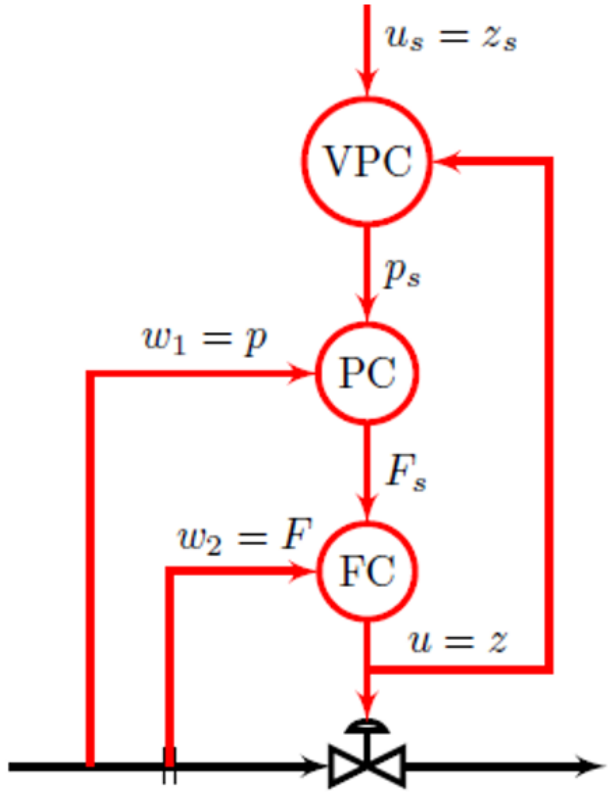


Figure 14: Stabilizing control of variable w_1 combined with valve position control (VPC) for u (=valve position) and inner flow controller ($w_2 = F$).
It corresponds to the flowsheet in Figure 15 with $w_1 = p$ (pressure), C_1 = outer VPC (slow), C_2 = stabilizing controller (fast), C_3 = inner flow controller (very fast).
Note that the process variables (w_1, w_2) have no fixed setpoint, so they are "floating".

Note: u is both an MV and a CV



Anti-slug control

Example: Anti-slug control

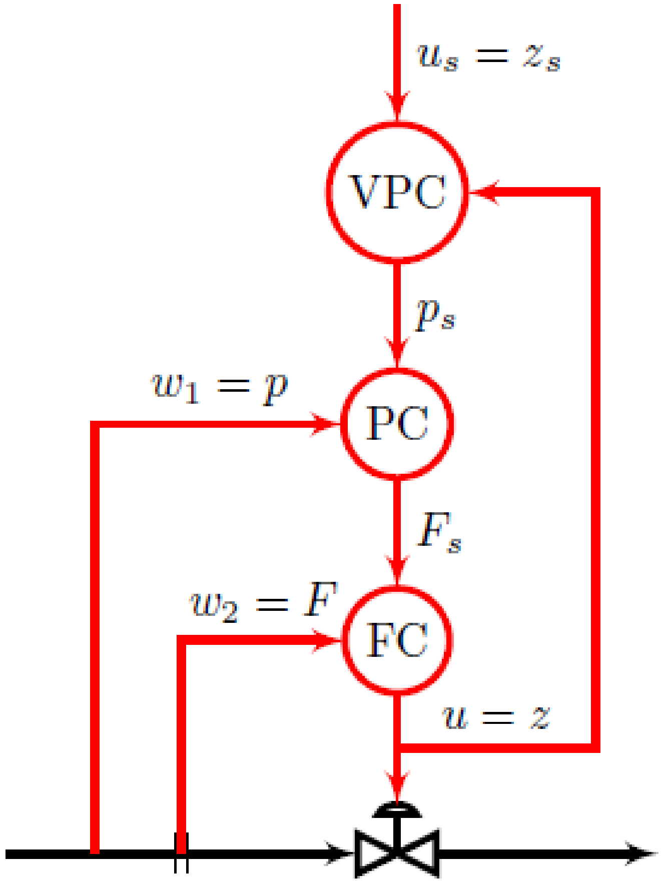


Figure 15: Anti-slug control where the pressure controller (PC) is used to stabilize a desired non-slugging flow regime. The inner flow controller (FC) (fast) provides linearization and disturbance rejection. The outer valve position controller (VPC) (slow) resets the valve position to its desired steady-state setpoint ($u_s = z_s$). It corresponds to the block diagram in Figure 14.

Note that this is a cascade control system, where we need at least a factor 4 (and preferably 10) between each layer. This implies that the outer VPC (C_1) must be at least 16 (and preferably 100) times slower than the inner flow controller (C_3). This may not be a problem for this application, because flow controllers can be tuned to be fast, with τ_c less than 10 seconds (Smuts, 2011).

Another more fundamental problem is that any unstable mode (RHP pole) in the process will appear as an unstable (RHP) zero as seen from the VPC (C_1) (Storkaas & Skogestad, 2004), which will limit the achievable speed (bandwidth) for resetting the valve to its desired position $u_s = z_s$.

Example: Stabilize bycycle

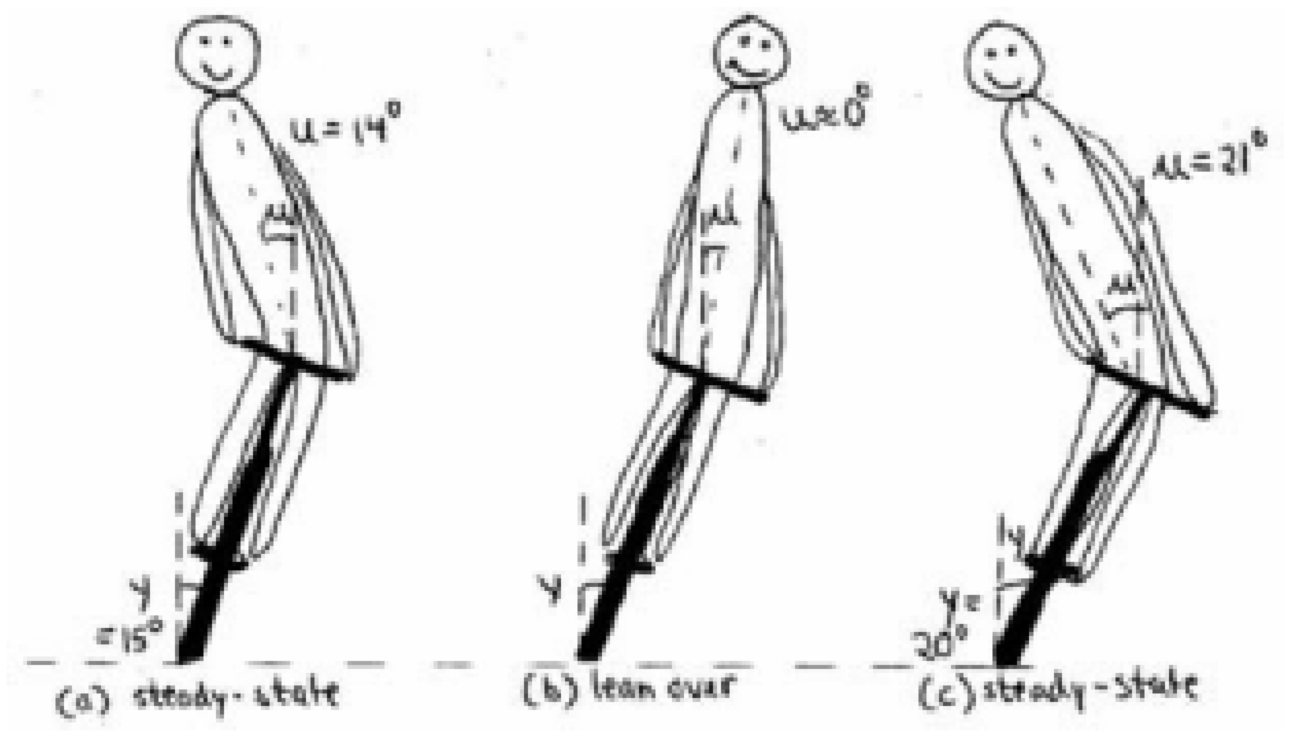


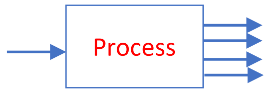
Fig. 2. Inverse response for a bicycle caused by an underlying instability

Consider Figure 2 where the aim is to tilt the bike from an initial angle $y = 15°$ (Fig. 2a) using your body (u) to an angle $y = 20°$ (Fig. 2c). Because of the inverse response, you first have to tilt your body in the direction of the tilt to start the movement (Fig. 2b). Eventually, you will have to move your body back to restore balance. This inverse response will be slower the greater the angle y, changing the angle while keeping balanced gets progressively slower as the tilting angle is increased.

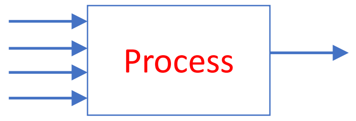
Espen Storkaas and Sigurd Skogestad, "Cascade Control of unstable systems with application to stabilization of slug flow", IFAC-symposium Adchem'2003, Hong Kong,

Constraint switching (because it is optimal at steady state)

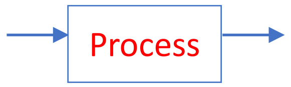
1. CV-CV switching
   - Control one CV at a time

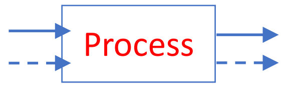


2. MV-MV switching
   - Use one MV at a time



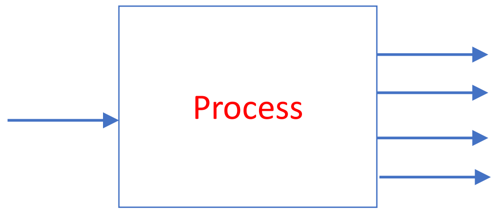
- MV-CV switching
  - MV saturates so must give up CV
  3. Simple («do nothing»)
  4. Complex (repairing of loops)

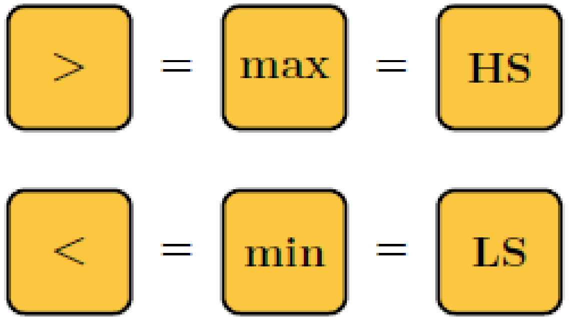




E4. Selector (for CV-CV switching*)

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





$$\boxed{>} = \boxed{\max} = \boxed{\text{HS}}$$

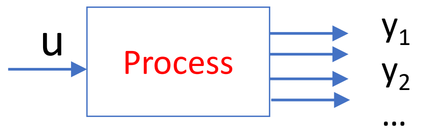
$$\boxed{<} = \boxed{\min} = \boxed{\text{LS}}$$

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 !

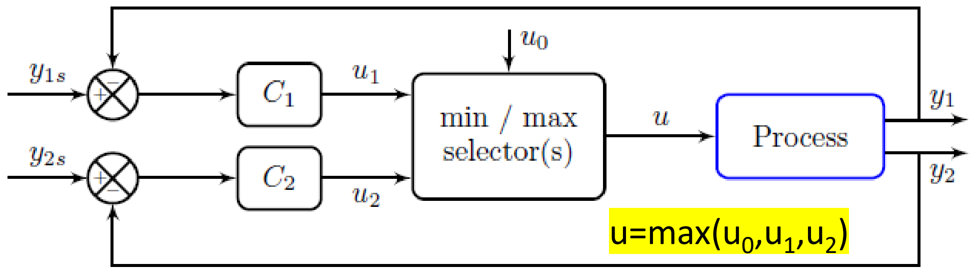
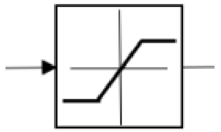


Figure 17: CV-CV switching with selector on MV (input u).

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)



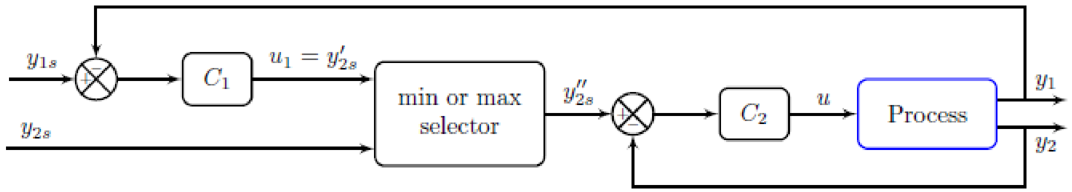
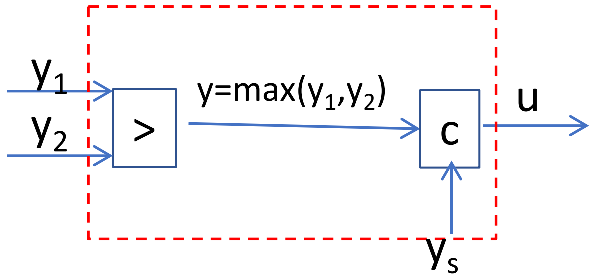


Figure 19: Alternative cascade CV-CV switching implementation with selector on the setpoint. In many cases, y_{1s} and y_{2s} are constraint limits.

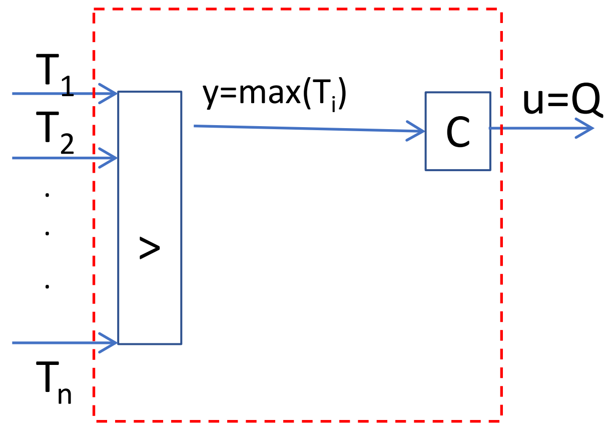
Alt. III (For special case where all CVs have same bound). One controller

- 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)

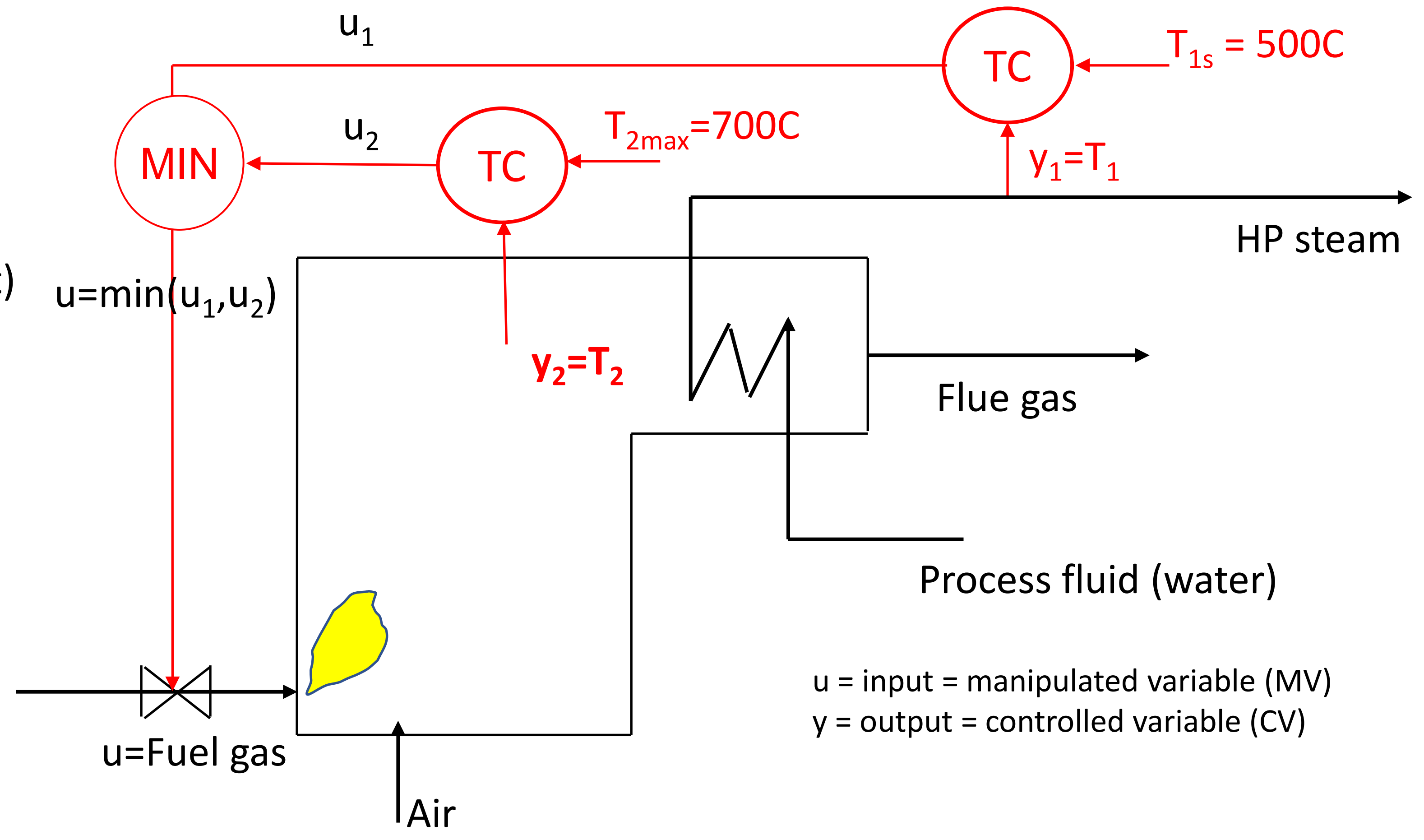
y_1 = process temperature T_1

(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

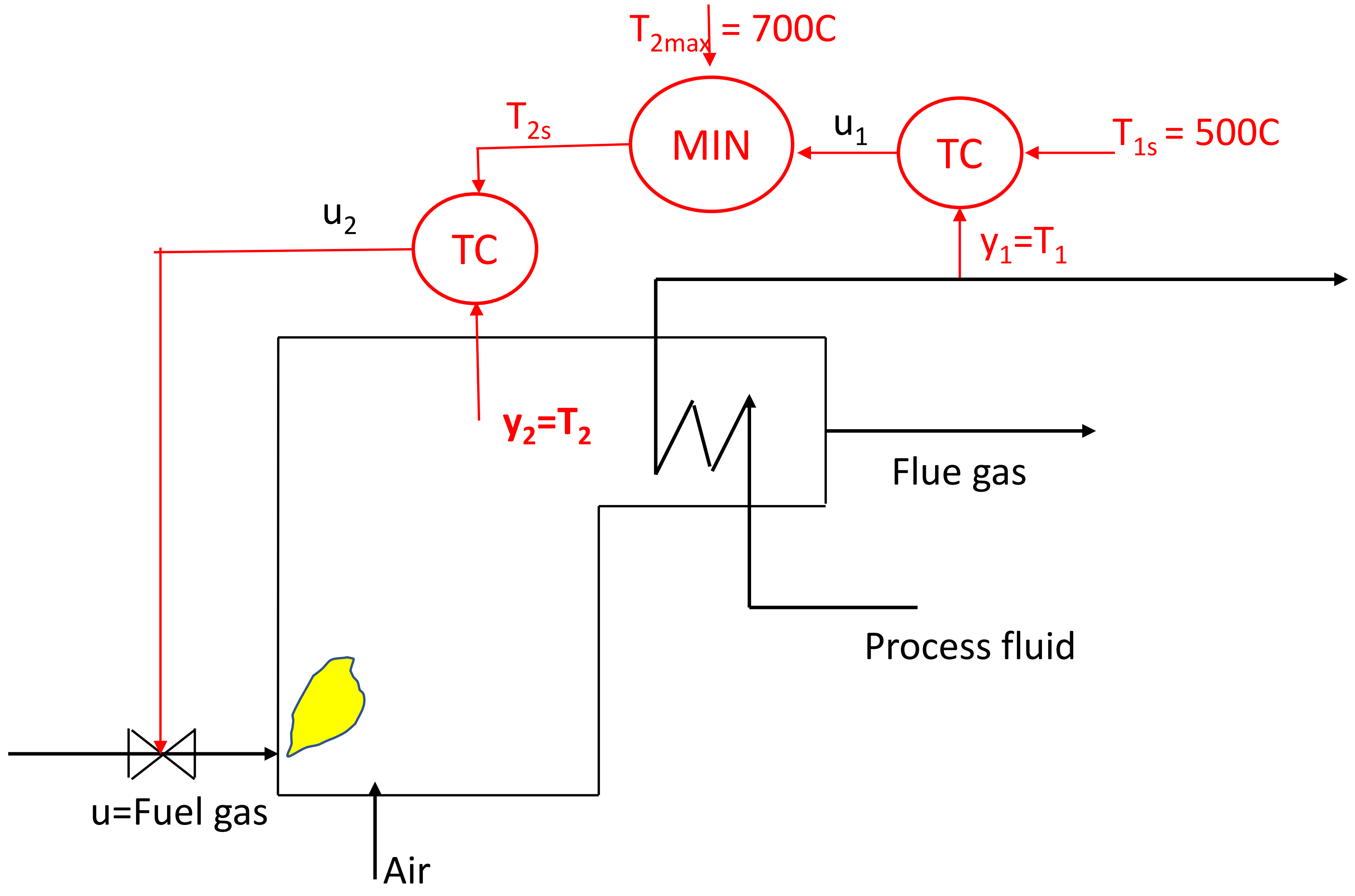


u = input = manipulated variable (MV)

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)

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

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

Minimize recycle (MV=z) subject to

CV= $F \geq F_{min}$

MV ≥ 0

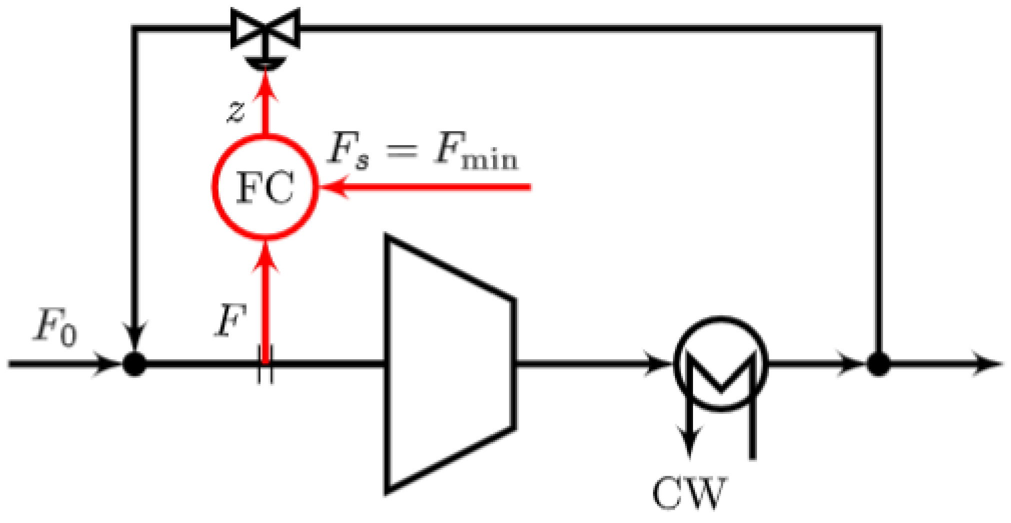


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding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

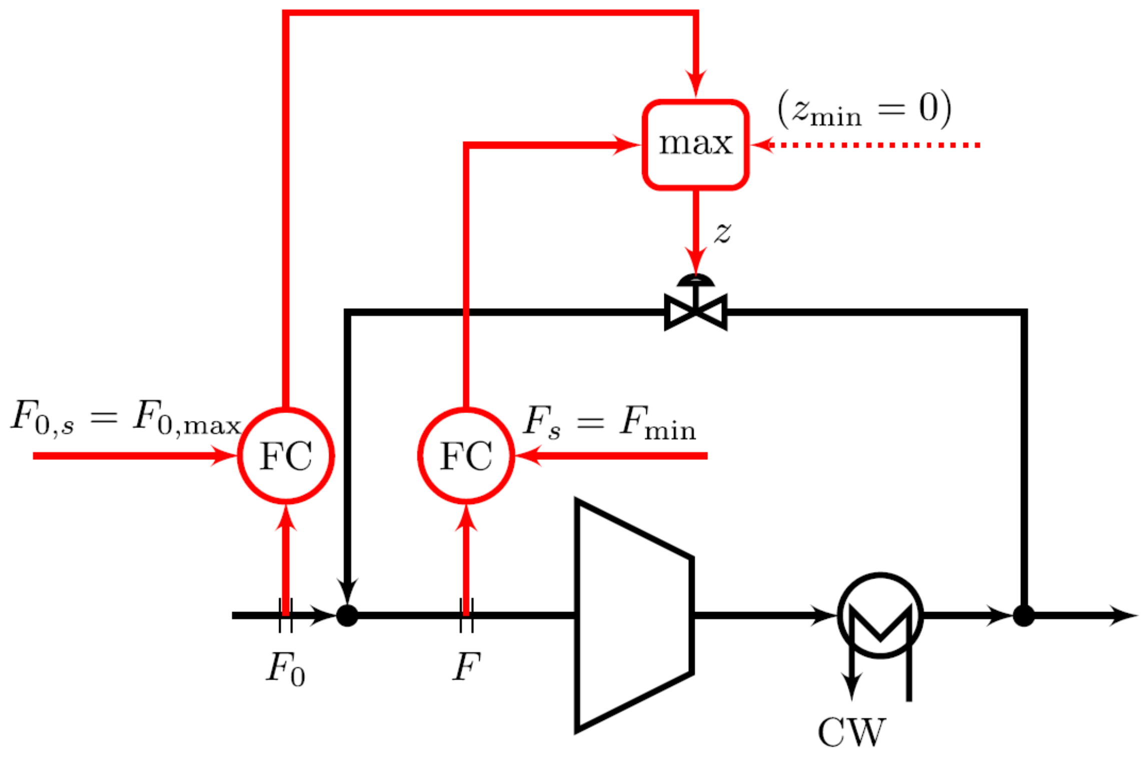
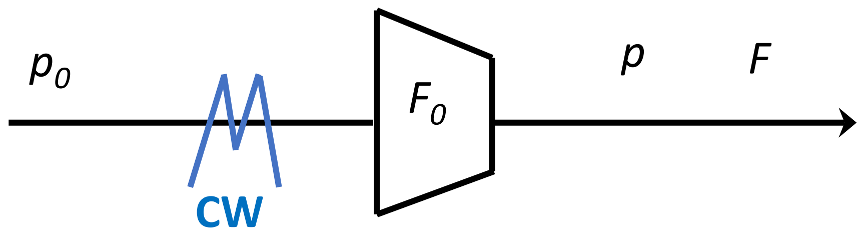
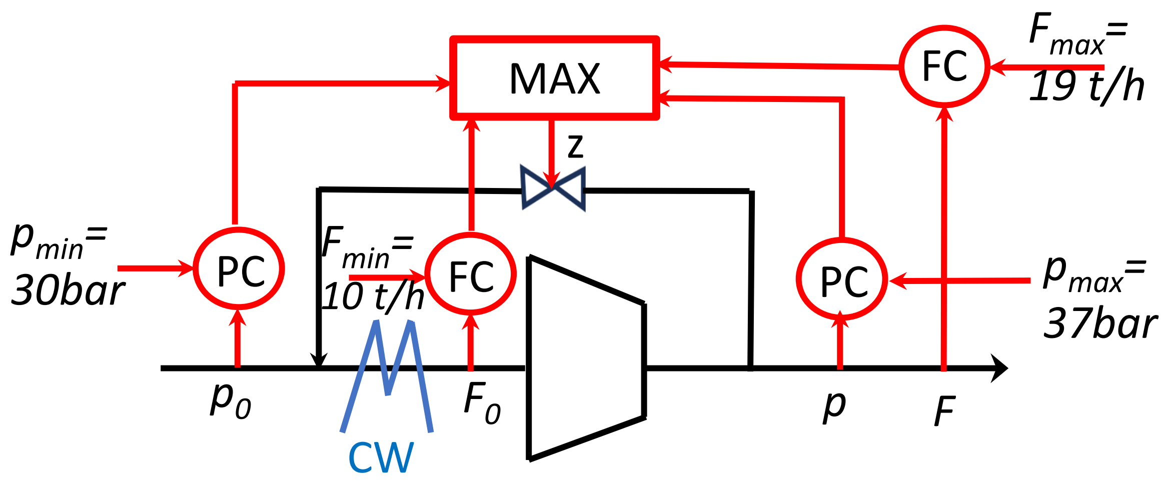


Fig. 33. Anti-surge compressor control with two CV constraints. This is an example of simple MV-CV-CV switching. MV = z, $CV_1 = F$, $CV_2 = F_0$ (all potentially active constraints).

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)

Rule 2 (order of selectors)

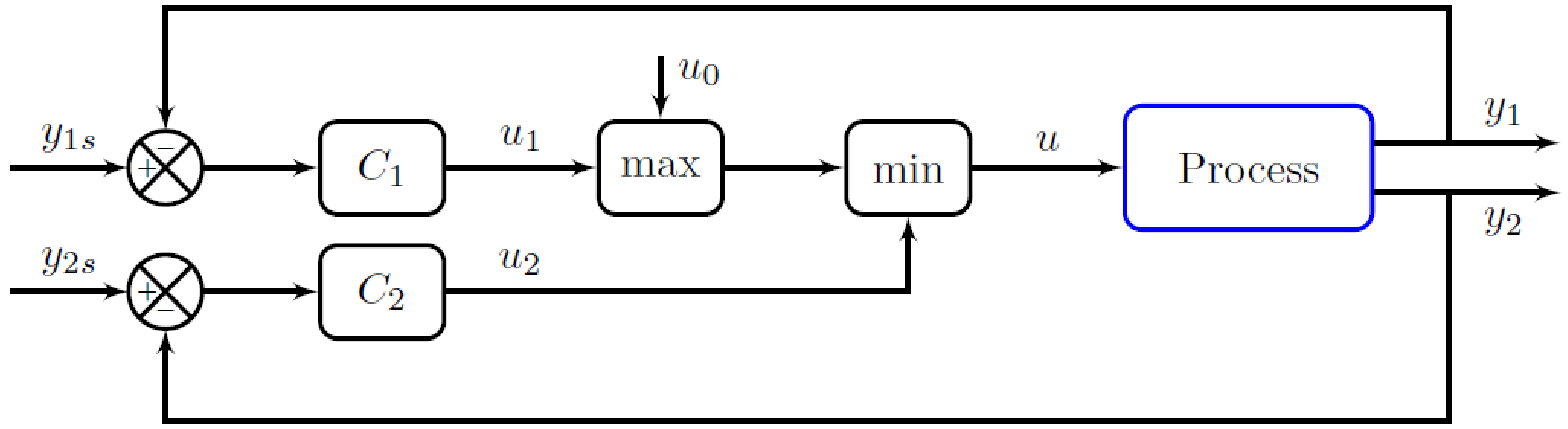


Figure 18: CV-CV switching for case with possibly conflicting constraints. In this case, constraint y_{1s} requires a max-selector and y_{2s}) requires a min-selector. The selector block corresponding to the most important constraint (here y_{2s}) should be at the end (Rule 2).

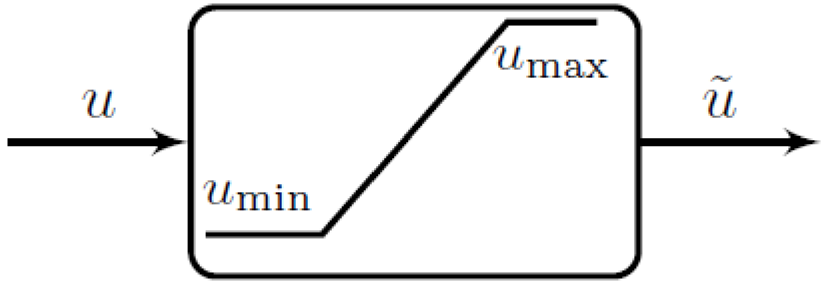
To understand the logic with selectors in series, start reading from the first selector. In this case, this is the max-selector: The constraint on y_1 is satisfied by a large value for u which requires a max-selector (Rule 1). u_0 is the desired input for cases when no constraints are encountered, but 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.

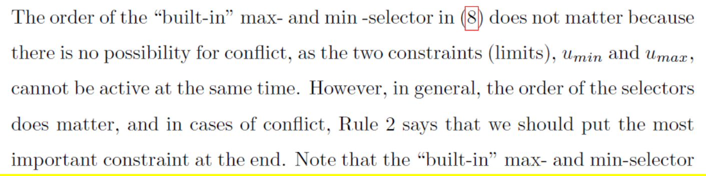
Valves have "built-in" selectors

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?

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

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

Part 4: More elements, more on switching

E5. Split-range control (SRC) (for MV-MV switching)

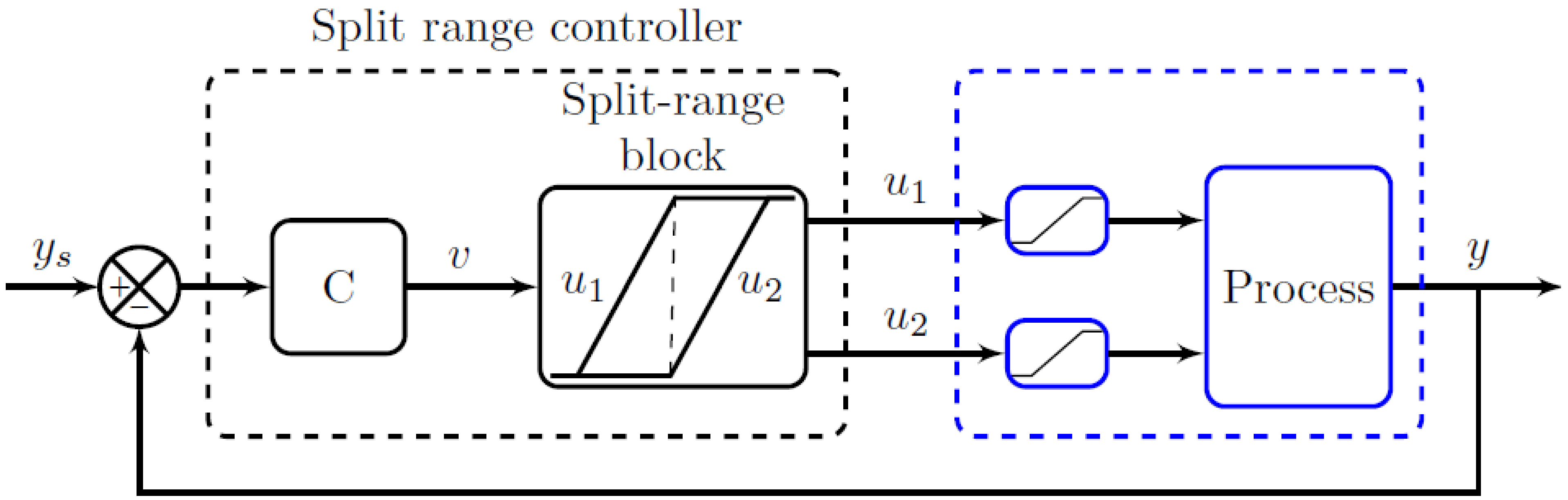


Figure 21: Split range control for MV-MV switching.

[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)

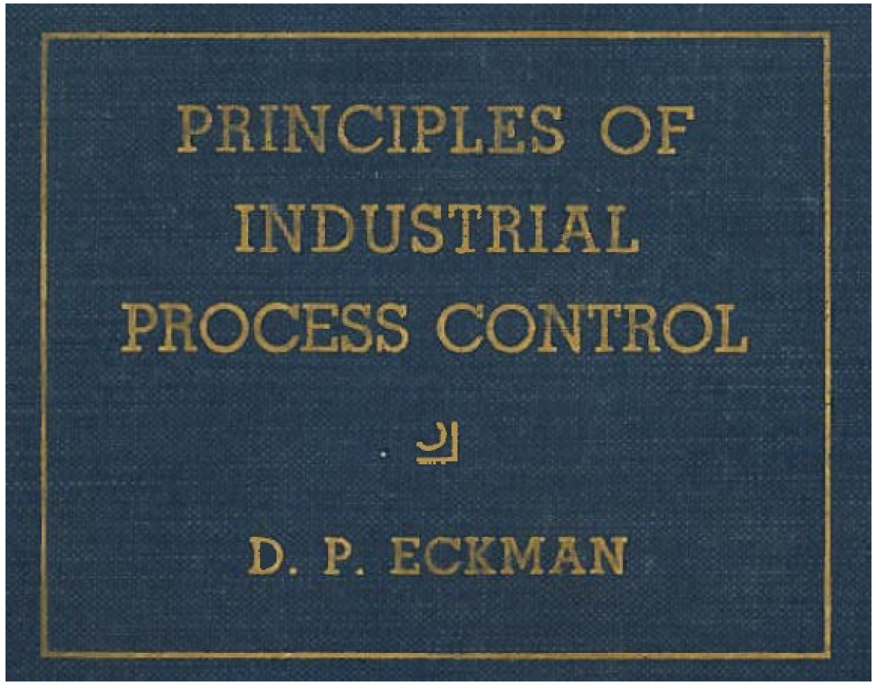
SRC is easy to understand and implement!

Disadvantages:

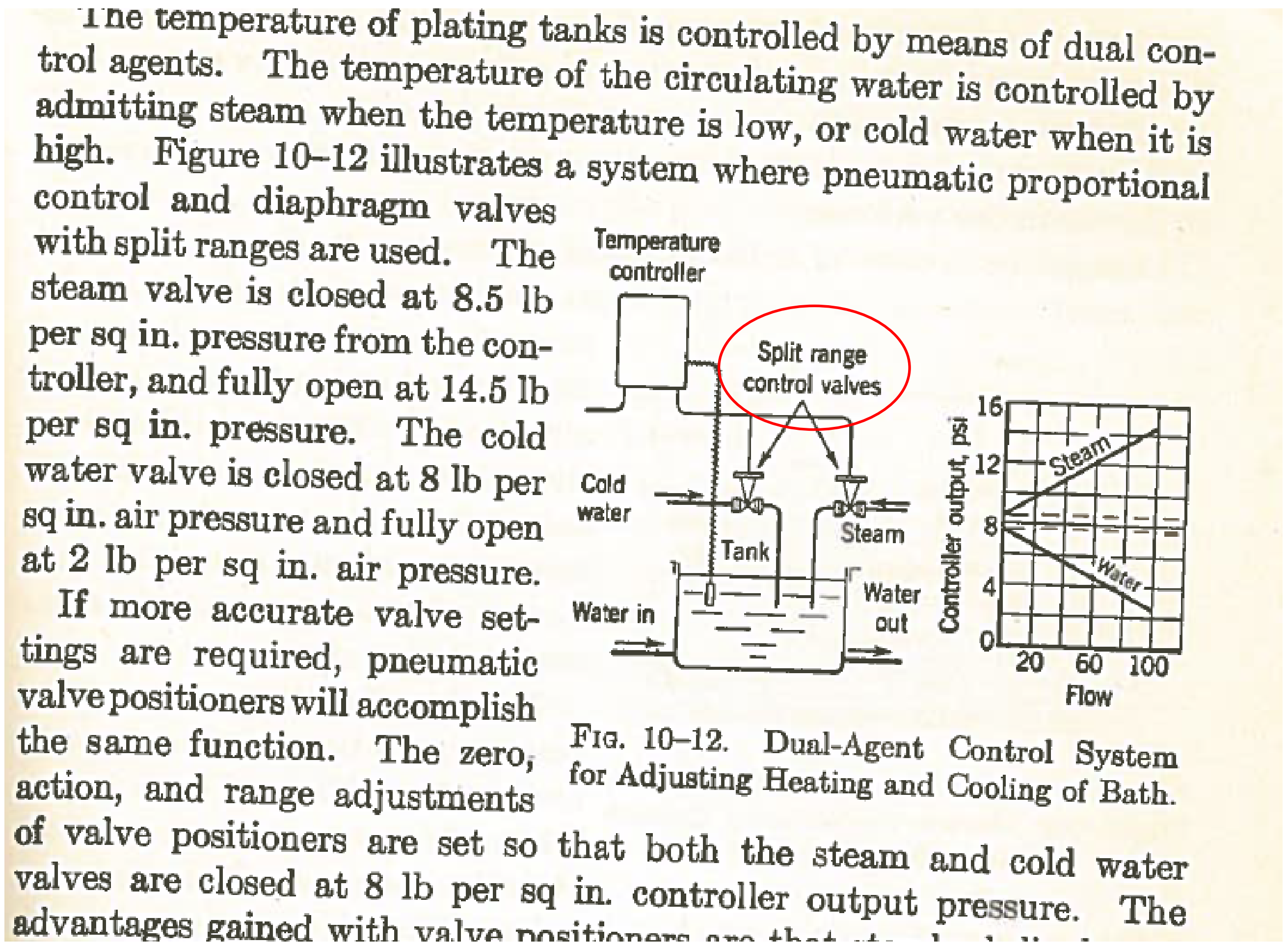
1. Only one controller ⇒ Same integral time for all inputs u_i (MVs)
   – Controller gains can be adjusted with slopes in SR-block!
2. Does not work well for cases where constraint values for u_i change

MV-MV switching

Split range control:
Donald Eckman (1945)


PRINCIPLES OF INDUSTRIAL PROCESS CONTROL

D. P. ECKMAN


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

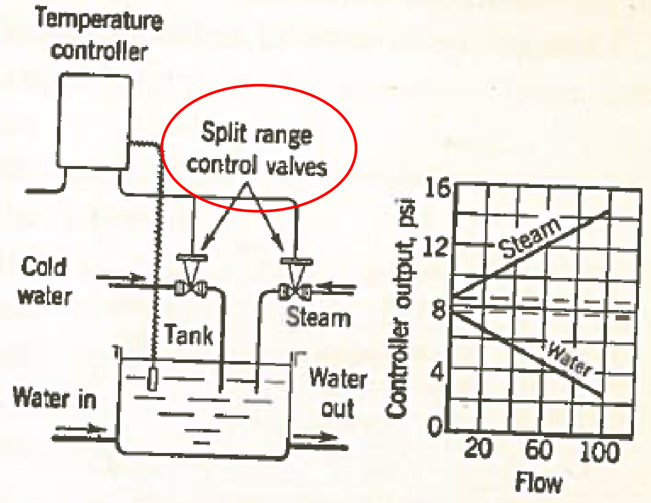
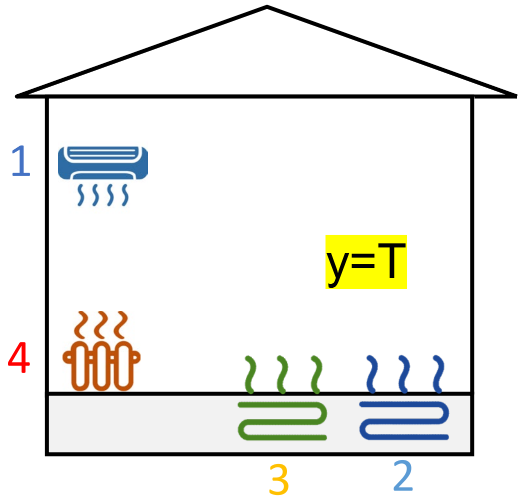


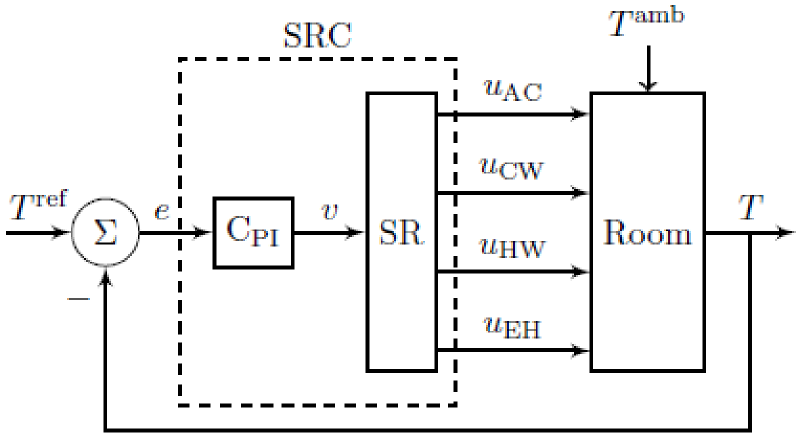
Fig. 10–12. Dual-Agent Control System for Adjusting Heating and Cooling of Bath.

Example split range control: Room temperature with 4 MVs

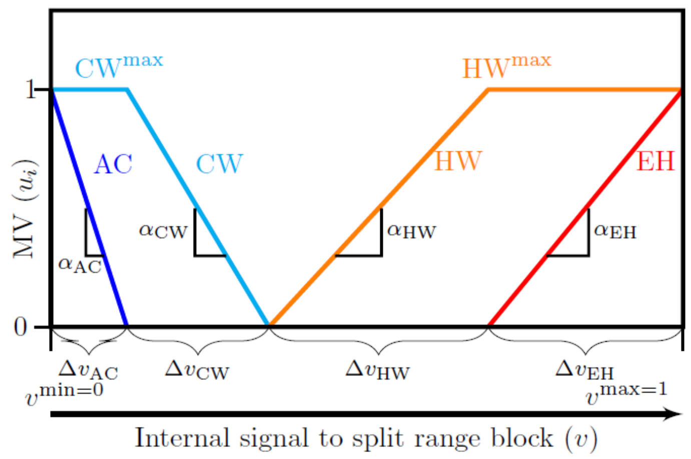


MVs (two for summer and two for winter):

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



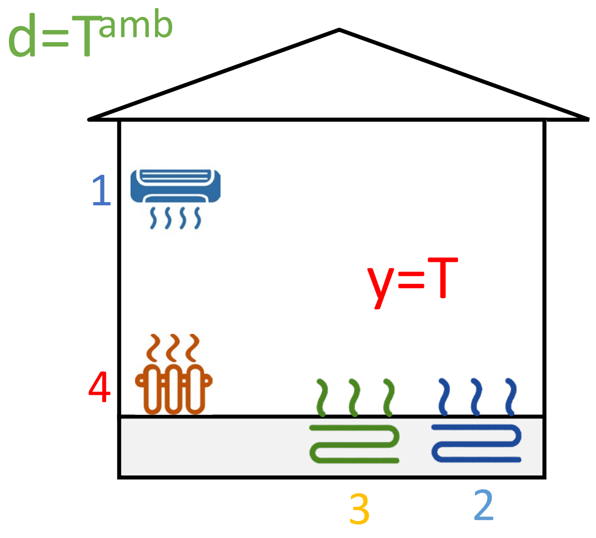
SR-block:

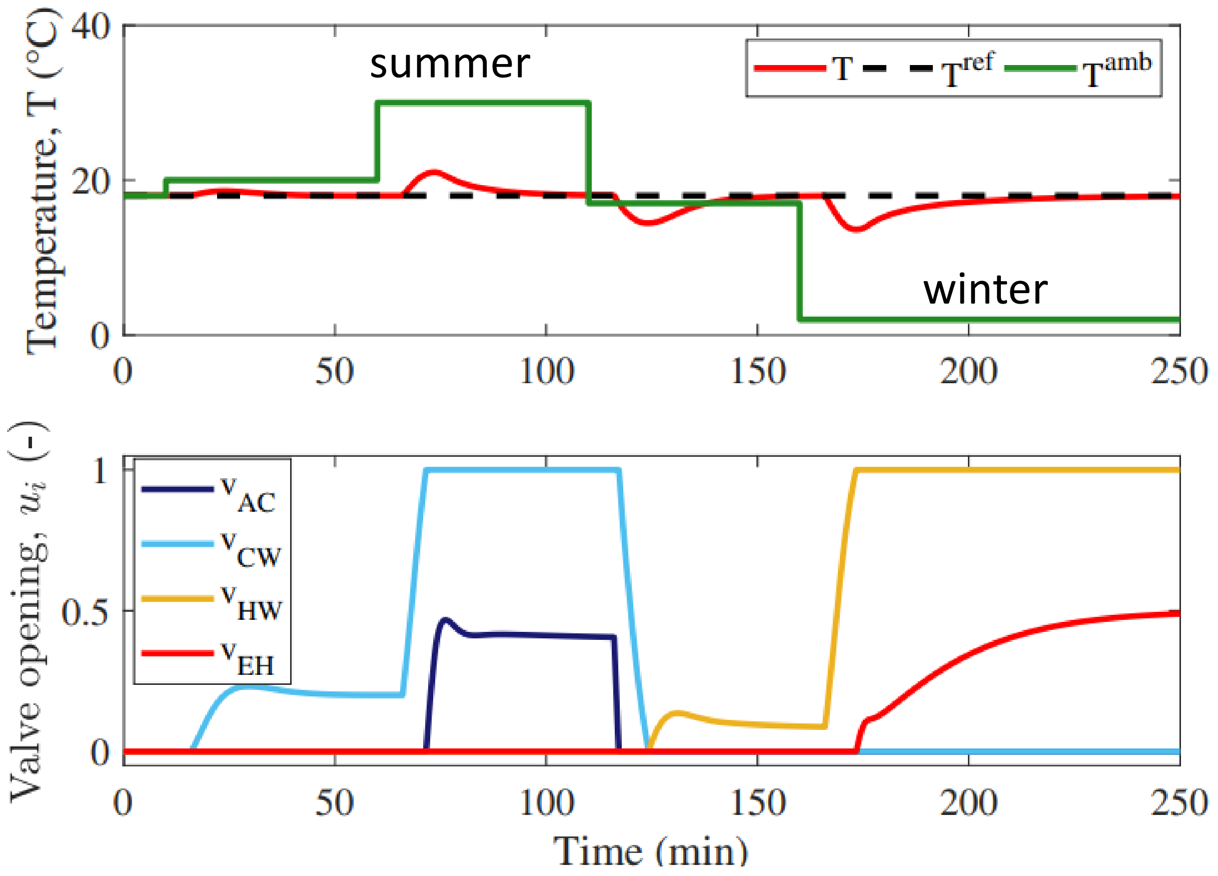


C_{PI} – same controller for all inputs (one integral time)

But get different gains by adjusting slopes α in SR-block

Simulation Split-range control (SRC)



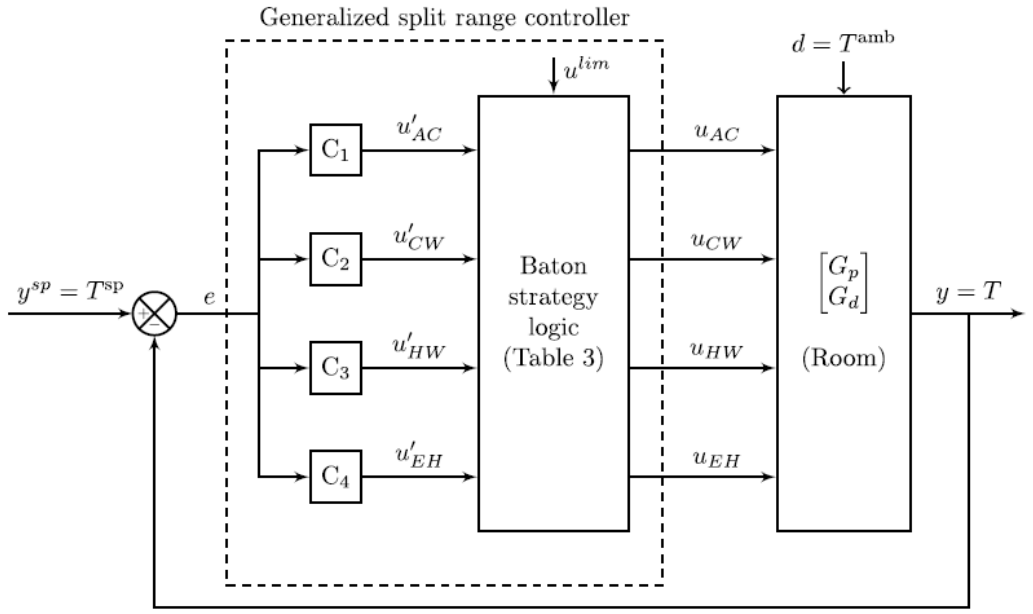


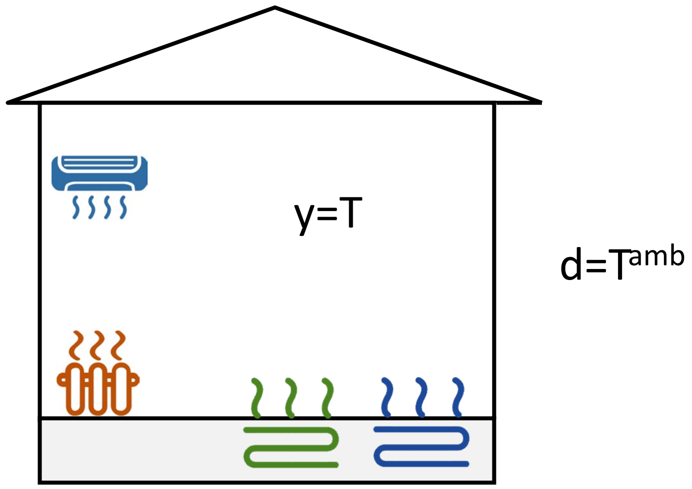
MV-MV switching

Disadvantages Standard Split-range control (SRC):

1. Must use same integral/derivative time for all MVs
2. Does not work well when constraint values change (SR-block problem)

Alternative: Generalized SRC (Baton strategy: multiple independent controllers)





All four controllers need anti-windup

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)

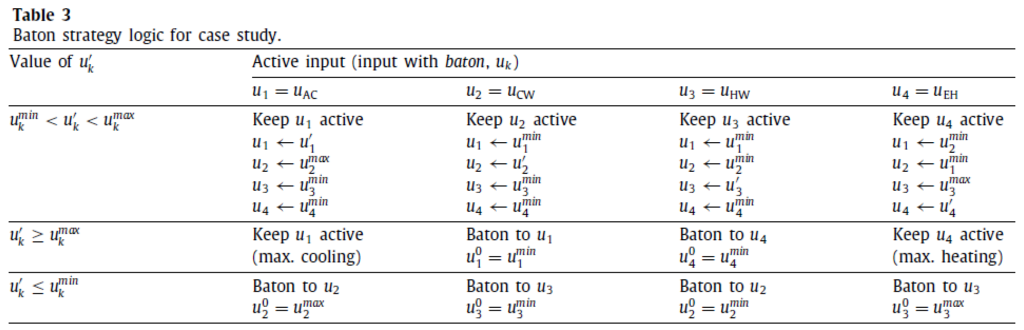


Table 3
Baton strategy logic for case study.

Value of u'_k	Active input (input with *baton*, u_k)			
	$u_1 = u_{AC}$	$u_2 = u_{CW}$	$u_3 = u_{HW}$	$u_4 = u_{EH}$
$u_k^{min} < u'_k < u_k^{max}$	Keep u_1 active $u_1 \leftarrow u'_1$ $u_2 \leftarrow u_2^{max}$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	Keep u_2 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u'_2$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	Keep u_3 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u_2^{min}$ $u_3 \leftarrow u'_3$ $u_4 \leftarrow u_4^{min}$	Keep u_4 active $u_1 \leftarrow u_2^{min}$ $u_2 \leftarrow u_1^{min}$ $u_3 \leftarrow u_3^{max}$ $u_4 \leftarrow u'_4$
$u'_k \geq u_k^{max}$	Keep u_1 active (max. cooling)	Baton to u_1 $u_1^0 = u_1^{min}$	Baton to u_4 $u_4^0 = u_4^{min}$	Keep u_4 active (max. heating)
$u'_k \leq u_k^{min}$	Baton to u_2 $u_2^0 = u_2^{max}$	Baton to u_3 $u_3^0 = u_3^{min}$	Baton to u_2 $u_2^0 = u_2^{min}$	Baton to u_3 $u_3^0 = u_3^{max}$

MV-MV switching

Disadvantages Alt. 1A. Standard Split-range control (SRC):

1. Must use same integral/derivative time for all MVs
2. Does not work well when constraint values change (SR-block problem)

Alt. 1B. Generalized SRC (Baton strategy: multiple independent controllers)

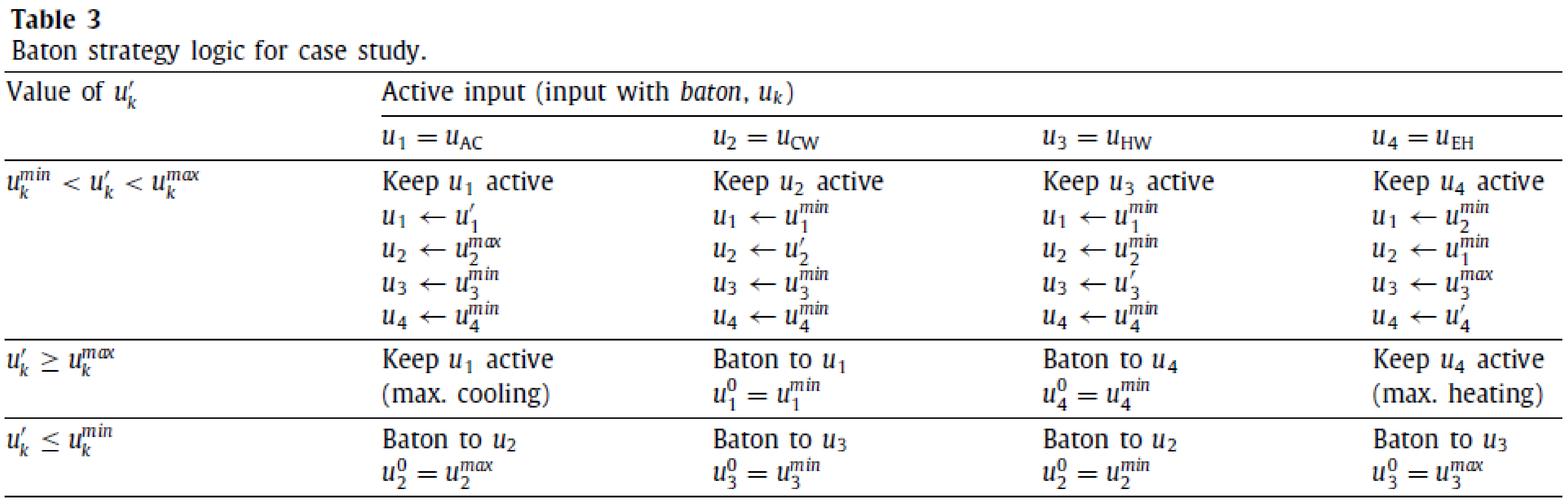


Table 3
Baton strategy logic for case study.

Value of u'_k	Active input (input with *baton*, u_k)			
	$u_1 = u_{AC}$	$u_2 = u_{CW}$	$u_3 = u_{HW}$	$u_4 = u_{EH}$
$u_k^{min} < u'_k < u_k^{max}$	Keep u_1 active $u_1 \leftarrow u'_1$ $u_2 \leftarrow u_2^{max}$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	Keep u_2 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u'_2$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	Keep u_3 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u_2^{min}$ $u_3 \leftarrow u'_3$ $u_4 \leftarrow u_4^{min}$	Keep u_4 active $u_1 \leftarrow u_2^{min}$ $u_2 \leftarrow u_1^{min}$ $u_3 \leftarrow u_3^{max}$ $u_4 \leftarrow u'_4$
$u'_k \geq u_k^{max}$	Keep u_1 active (max. cooling)	Baton to u_1 $u_1^0 = u_1^{min}$	Baton to u_4 $u_4^0 = u_4^{min}$	Keep u_4 active (max. heating)
$u'_k \leq u_k^{min}$	Baton to u_2 $u_2^0 = u_2^{max}$	Baton to u_3 $u_3^0 = u_3^{min}$	Baton to u_2 $u_2^0 = u_2^{min}$	Baton to u_3 $u_3^0 = u_3^{max}$

ıb

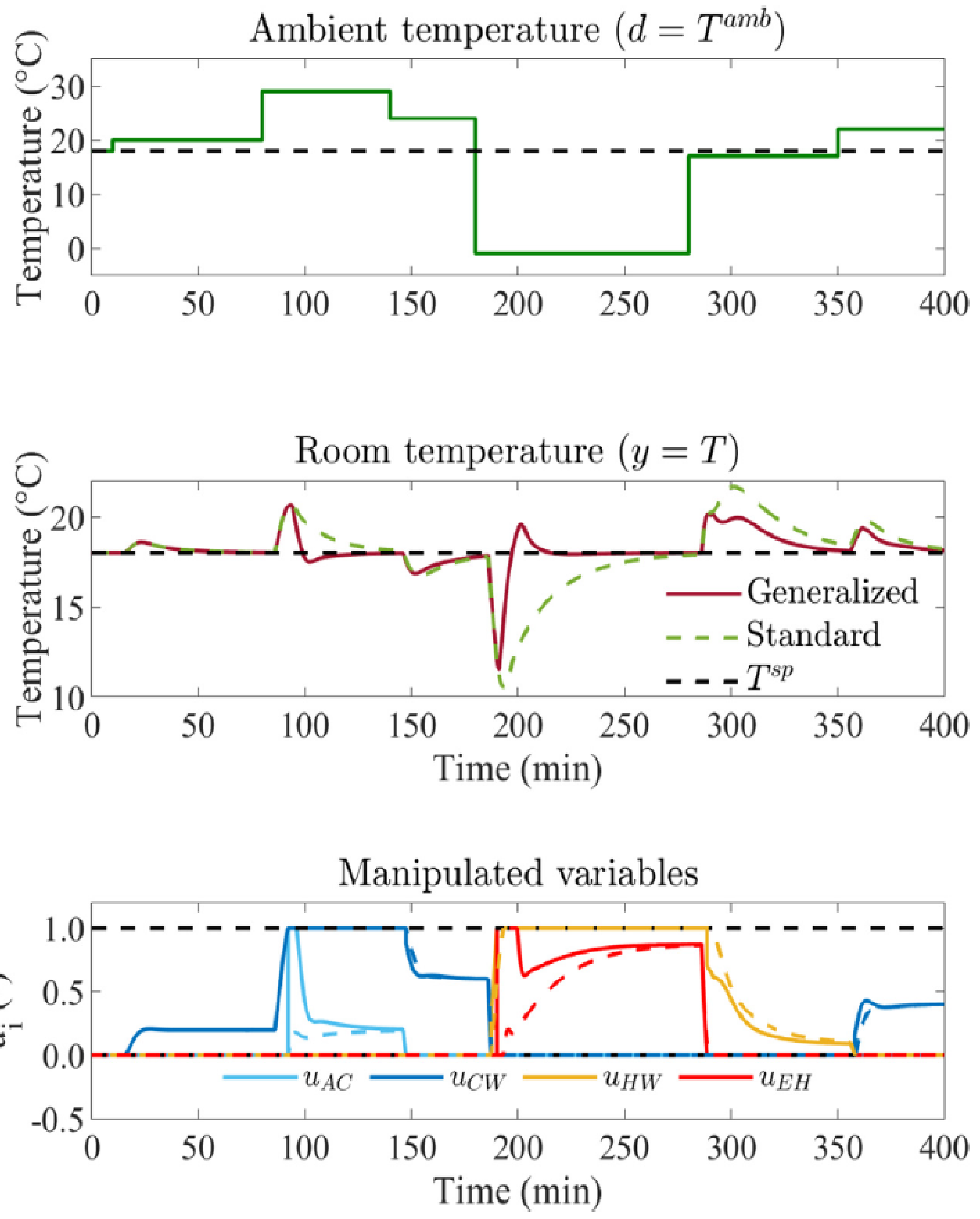
All four controllers need anti-windup

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)

Comparison of standard and generalized SRC

Generalized split range control:

- Different (smaller) integral times for each input
- Gives faster settling for most inputs



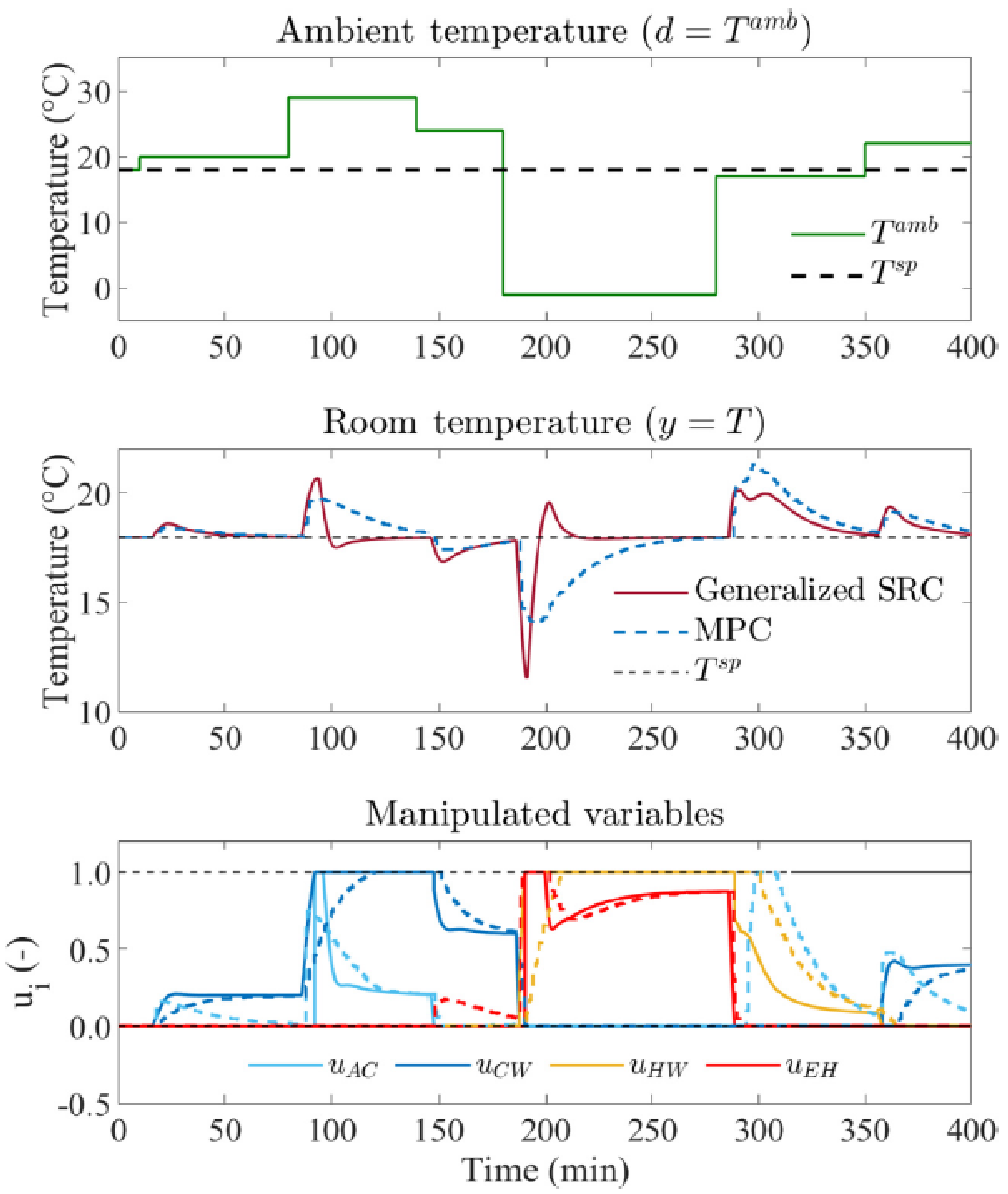
What about Model Predictive control (MPC)?

Comparison of Generalized SRC and MPC

Responses
MPC: Similar response to standard SRC
MPC: Faster initially, uses several input simultanously
MPC: Slower settling

Disadvantage MPC:

- Complex: Requires full dynamic model
- Does not use on input at a time



E6. Separate controllers with different setpoints
(for MV-MV switching)

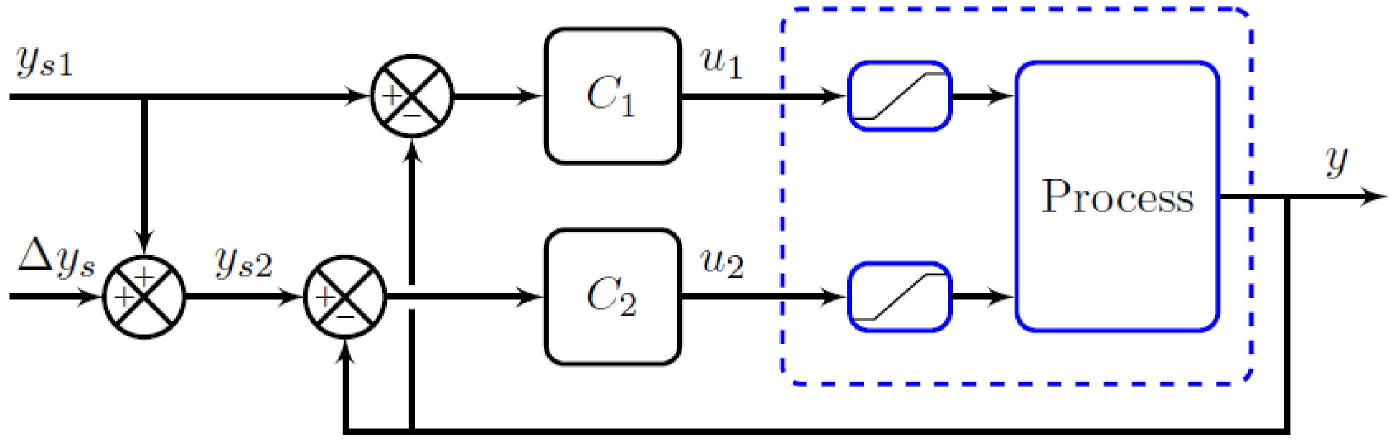


Figure 22: Separate controllers with different setpoints for MV-MV switching.

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).

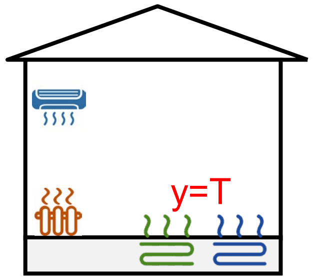
Advantages:

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)

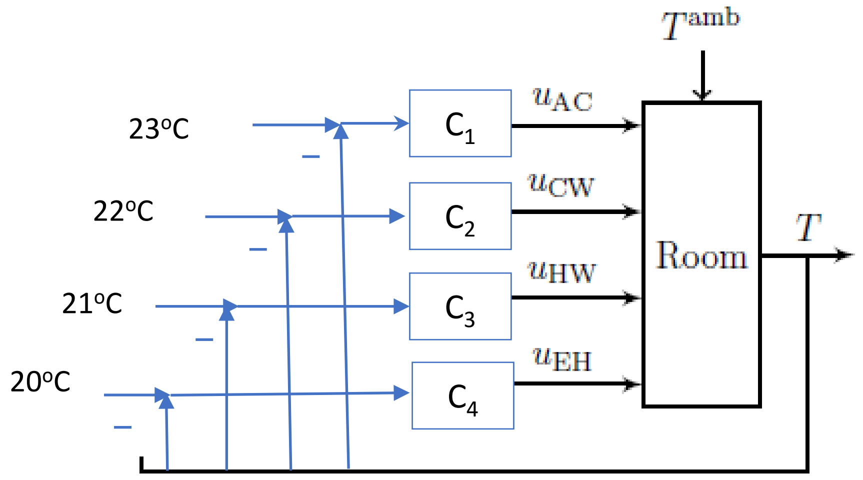
Example: Room heating with one CV (T) and 4 MVs



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. A2 for MV-MV switching. Multiple controllers with different setpoints



Disadvantage (comfort):

- Different setpoints

Advantage (economics) :

- Different setpoints (energy savings)

Want separate controllers. Fixes that avoid using different setpoints

Alt.1: «Baton strategy» (a bit complicated).

Alt.2 (simpler, but gives temporary setpoint change at MV-MV switch): Introduce a (slow) outer cascade (master controller) that resets the setpoint of the active controller to y_s, while maintaining the setpoint distances

Fix: Outer cascade to avoid different setpoints

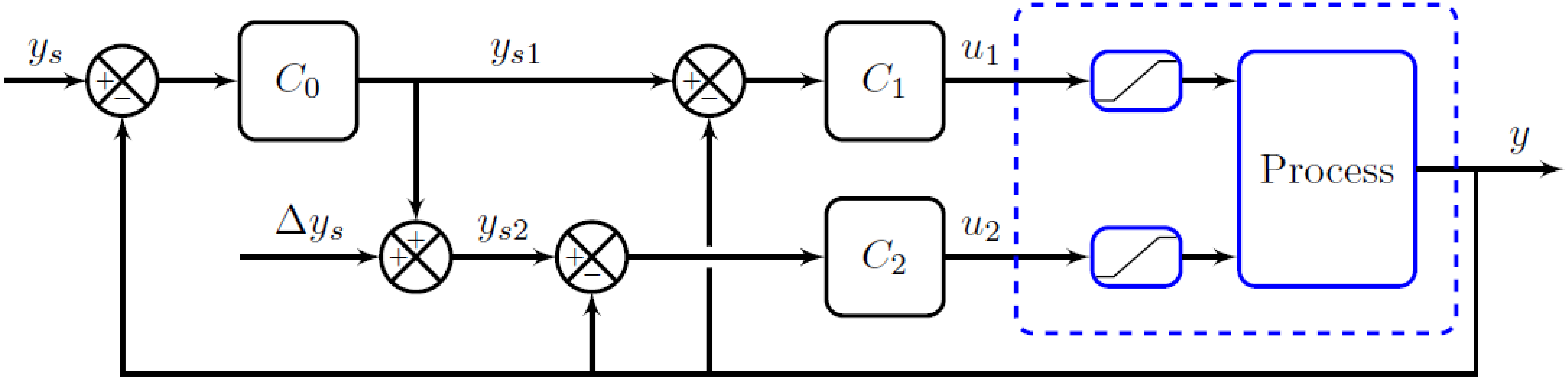


Figure 23: Separate controllers for MV-MV switching with outer resetting of setpoint. This is an extension of the scheme in Figure 22, with a slower outer controller C_0 that resets y_{1s} to keep a fixed setpoint $y = y_s$ at steady state.

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 ,**

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

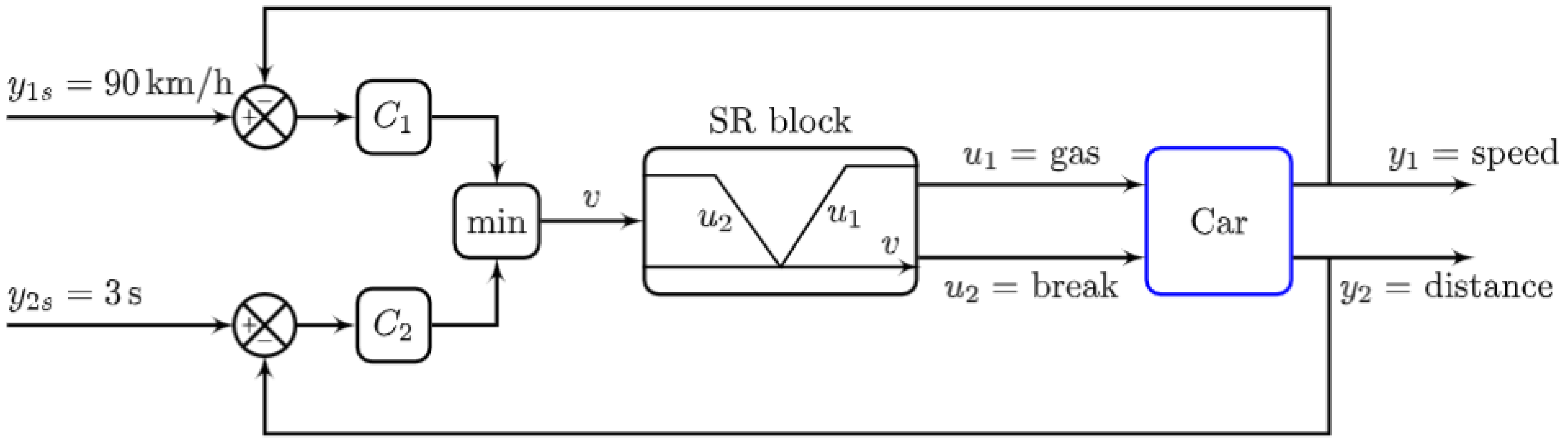


Fig. 31. Adaptive cruise control with selector and split range control.

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

E7. VPC on main steady-state input (for MV-MV switching)

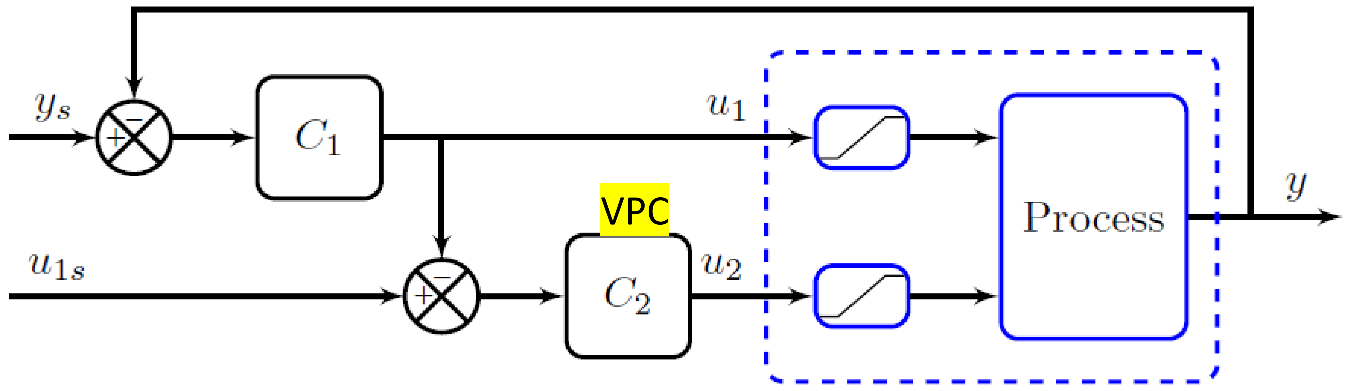


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

Use E7 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 E7:

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).

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

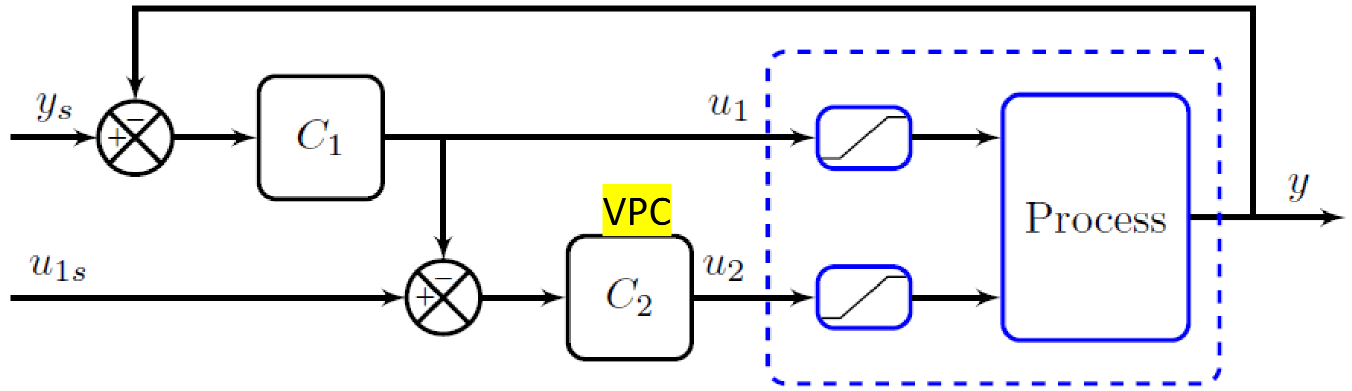


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

Same block diagram, except for the "need" for valve saturation
But their behavior is very different!

- Improve dynamics (E3) – should not have valve saturation
  - both inputs are used all the time
  - u_1 is used to improve the dynamic response
  - u_2 is the main steady-state input (and used all the time)
  - u_{1s} is typically 50% (mid-range)
- MV-MV switching (E7) - designed to avoid valve saturation
  - u_1 is the main input
  - u_2 is only used when u_1 approaches saturation (for MV-MV switching)
  - u_{1s} is typically close to the expected saturation constraint (10% or 90%)

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

E8. Anti-windup for the integral mode

$$u(t) = K_c e(t) + K_c\tau_D \frac{de(t)}{dt} + \underbrace{\overbrace{\frac{K_c}{\tau_I}\int_{t_0}^{t} e(t')dt'}^{u_I} + u_0}_{\text{bias}=b} \qquad \text{(C.1)}$$

Appendix C.6.1. Simple anti-windup schemes

Many industrial anti-windup schemes exist. The simplest is to limit u in (C.1) to be within specified bounds (by updating u_0), or to limit the bias $b = u_0 + u_I$ to be within specified bounds (also by updating u_0). These two options have the advantage that one does not need a measurement of the actual applied input value ($\tilde{u}$), and for most loops these simple anti-windup approaches suffice (Smith, 2010) (page 21).

Appendix C.6.2. Anti-windup using external reset

A better and also common anti-windup scheme is "external reset" (e.g., Wade (2004) Smith (2010)) which originates from Shinskey. This scheme is found in most industrial control systems and it uses the "trick" of realizing

Appendix C.6.3. Recommended: Anti-windup with tracking

The "external reset" solution is a special case of the further improved "tracking" scheme in Figure 7 which is recommended by Åström & Hägglund (1988). The tracking scheme (sometimes referred to as the "back-calculation" scheme (Åström & Hägglund, 2006)) has a very useful additional design parameter, namely the tracking time constant τ_T, which tells how fast the controller output u tracks the actual applied value $\tilde{u}$. This makes it possible to handle more general cases in a good way, e.g., switching of CVs. In the simpler "exter

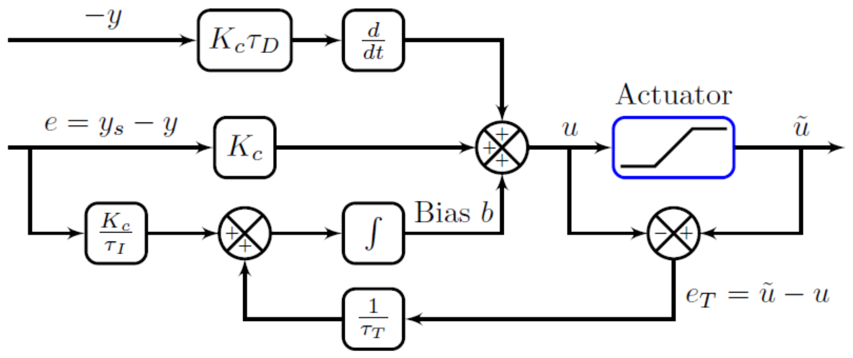


Figure 7: Recommended PID-controller implementation with anti-windup using tracking of the actual controller output ($\tilde{u}$), and without D-action on the setpoint. (Åström & Hägglund, 1988).

$$u(t) = K_c e(t) + K_c\tau_D \frac{de(t)}{dt} + \underbrace{\overbrace{\int_{\hat{t}=t_0}^{t}\left(\frac{K_c}{\tau_I}e(\hat{t}) + \frac{1}{\tau_T}e_T(\hat{t})\right)d\hat{t}}^{u_I(t)} + u_0}_{\text{bias}=b} \qquad \text{(C.14)}$$

to choose the tracking time equal to the integral time ($\tau_T = \tau_I$). With this value, we get at steady state that the output from the integral part (u_I) is such that the bias b is equal to the constraint value, $b = u_{lim}$. To derive this, note that with

Anti-windup with cascade control

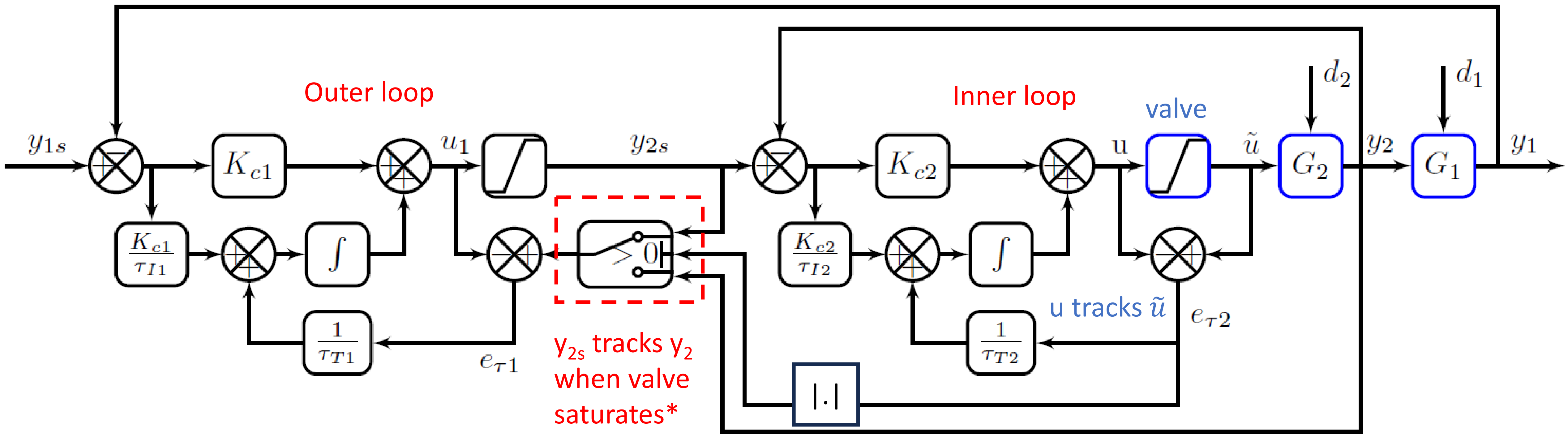


Figure 25: Cascade control with anti windup using the industrial switching approach (Leal et al., 2021).

* Normally, it's opposite: y_2 is tracking y_{2s}.