Part 4: More elements, more on switching

E5. Split-range control (SRC) (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.

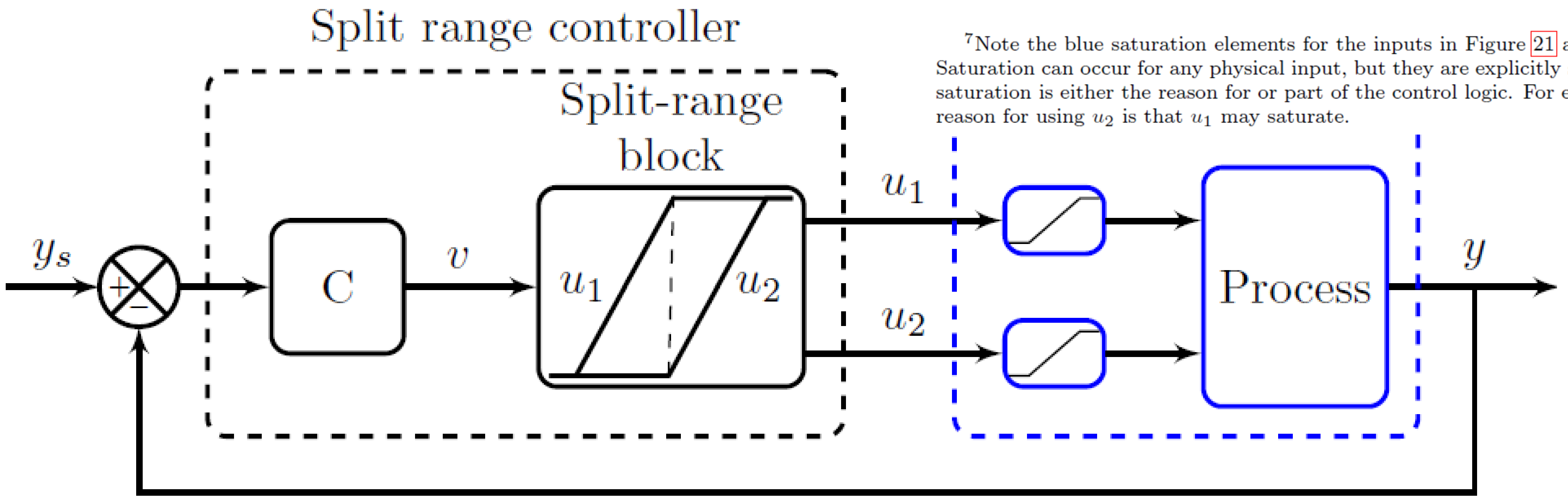


Figure 21: Split range control for MV-MV switching.

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!

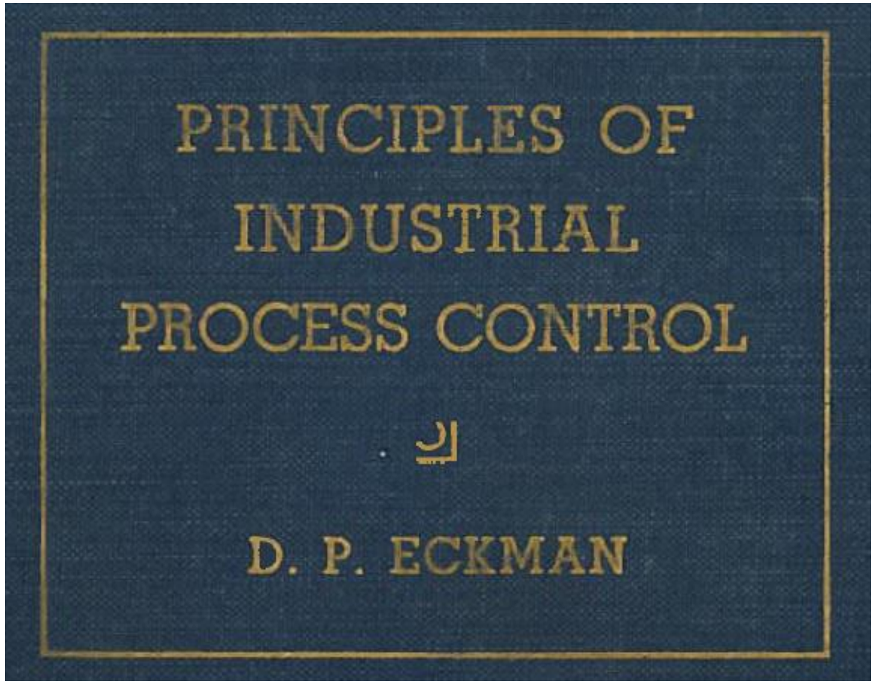
Disadvantages:

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)



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

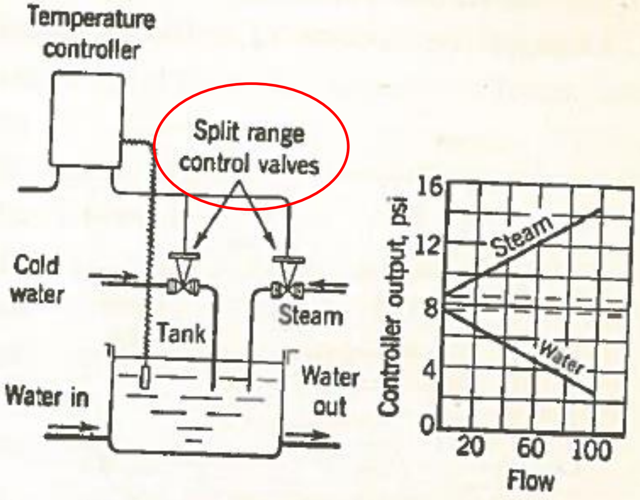
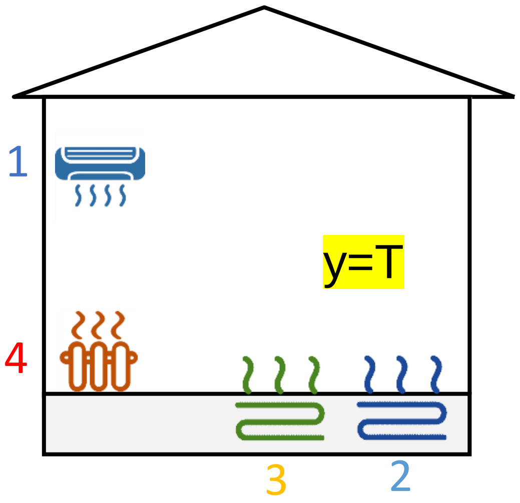


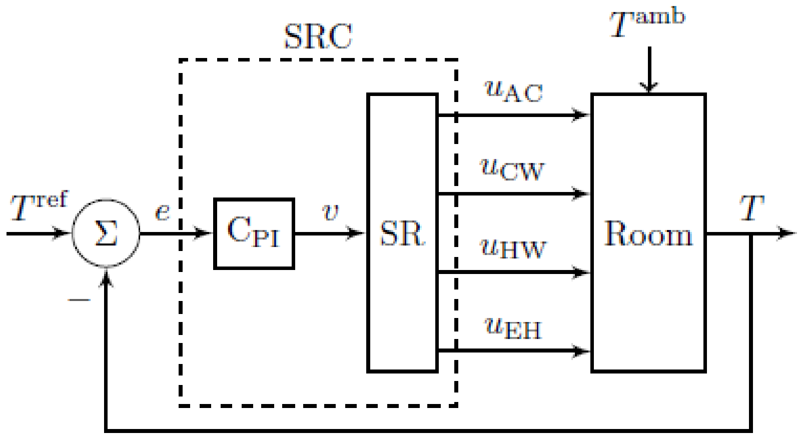
FIG. 10–12. Dual-Agent Control System for Adjusting Heating and Cooling of Bath.

Example split range control: Room temperature with 4 MVs

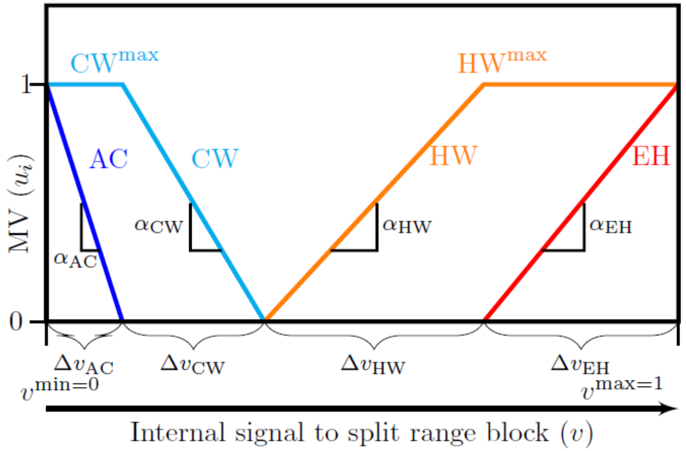


MVs (two for summer and two for winter):

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



SR-block:

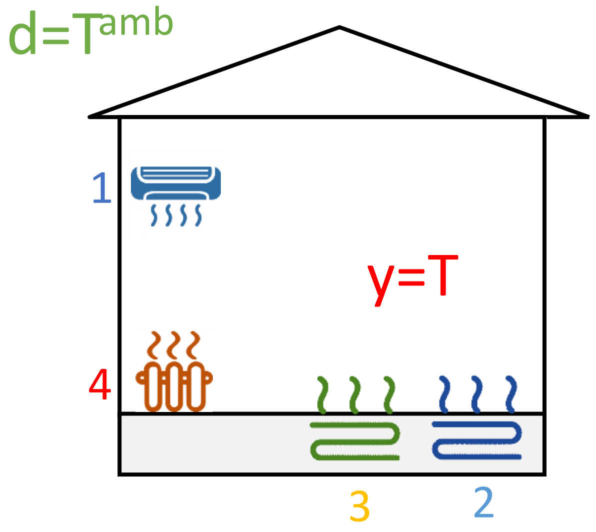


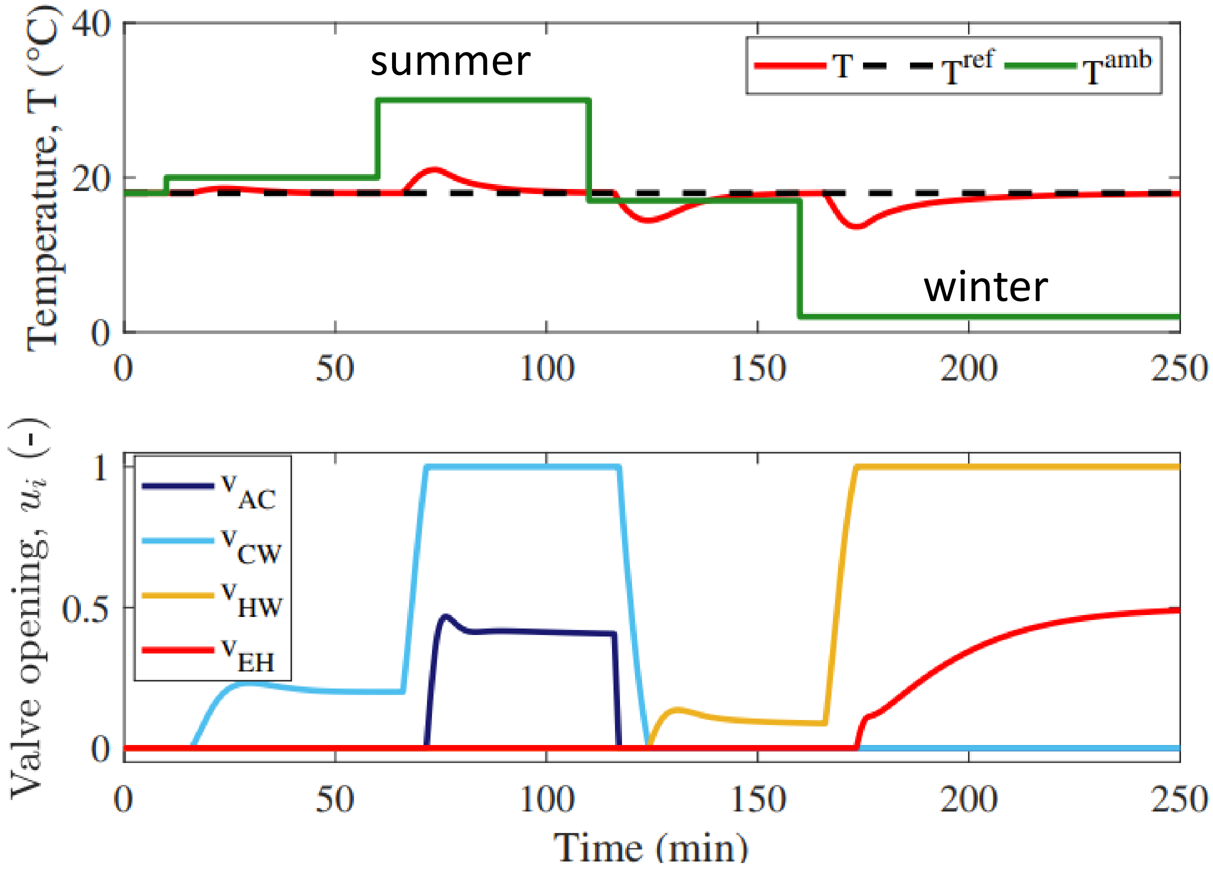
C_{PI} – same controller for all inputs (one integral time)

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

MV-MV switching

Simulation Split-range control (SRC)

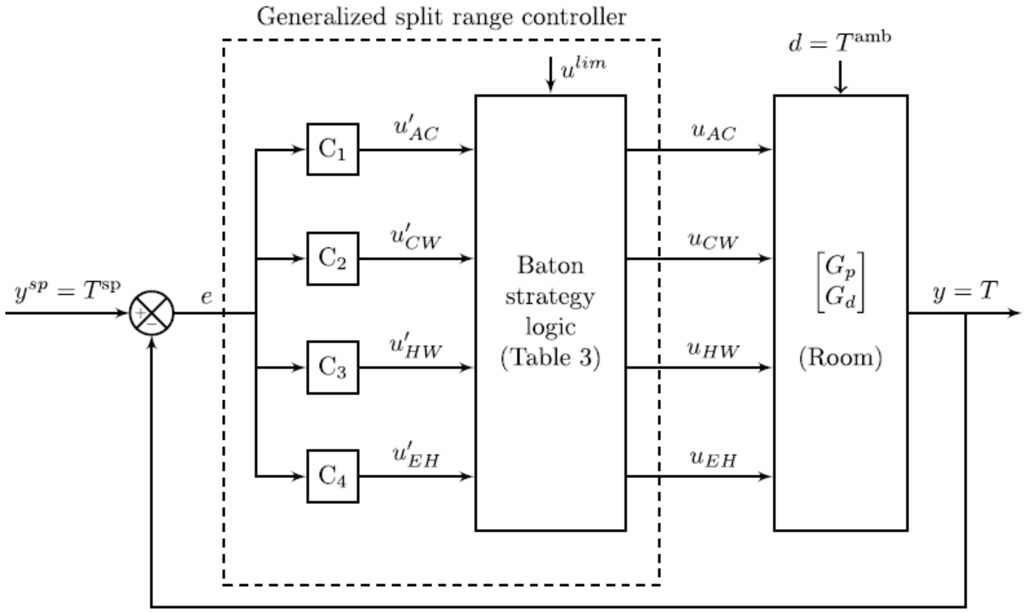




Disadvantages Standard Split-range control (SRC):

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)



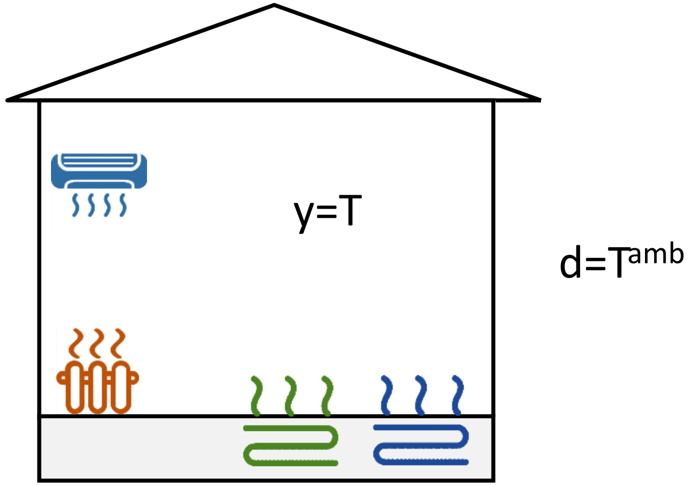


Table 3
Baton strategy logic for case study.

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 $u_3^0 = u_3^{max}$

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)

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.

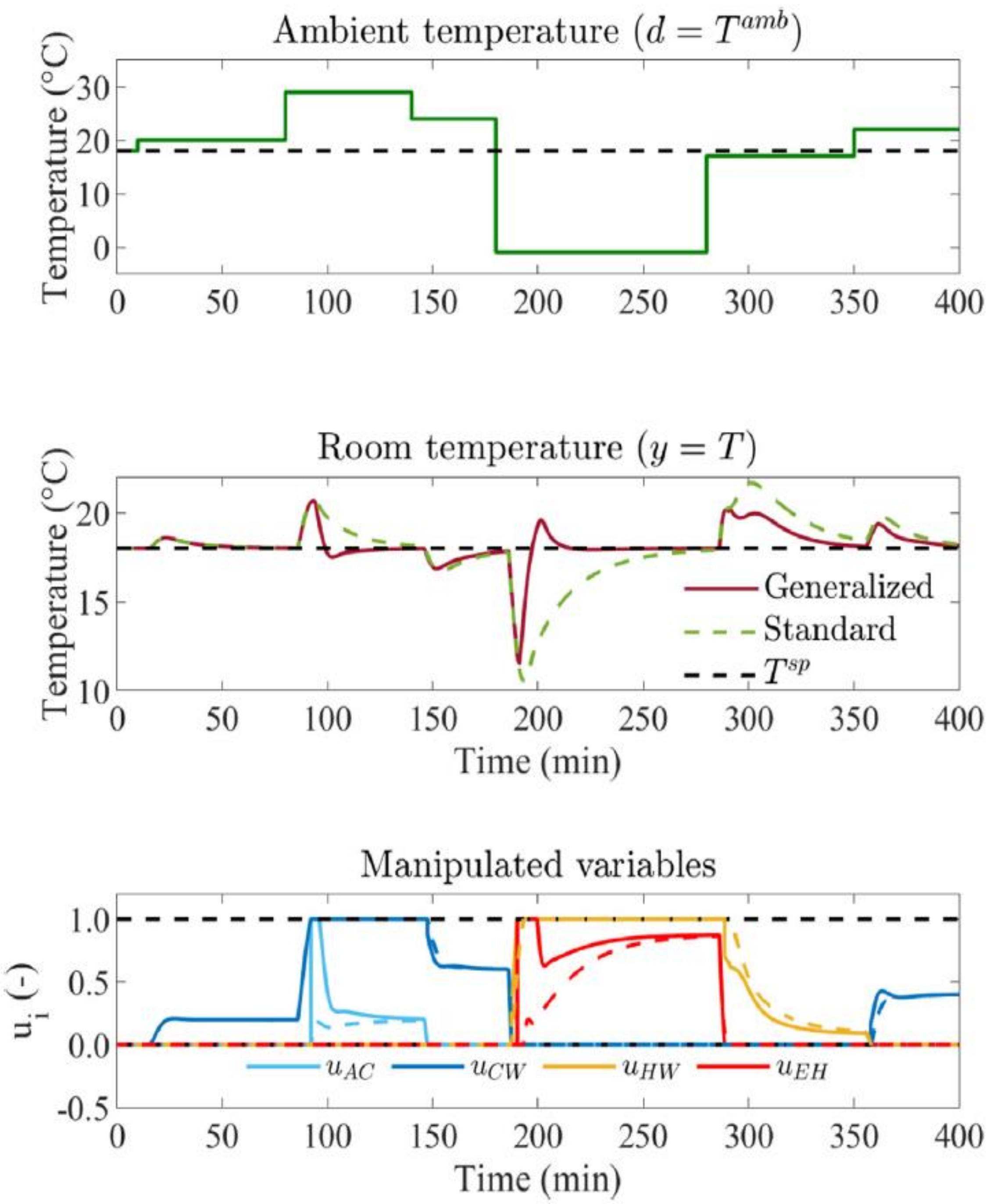
Value of u'_k	Active input (input with *baton*, u_k)			
	$u_1 = u_{AC}$	$u_2 = u_{CW}$	$u_3 = u_{HW}$	$u_4 = u_{EH}$
$u_k^{min} < u'_k < u_k^{max}$	Keep u_1 active $u_1 \leftarrow u'_1$ $u_2 \leftarrow u_2^{max}$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	Keep u_2 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u'_2$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	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

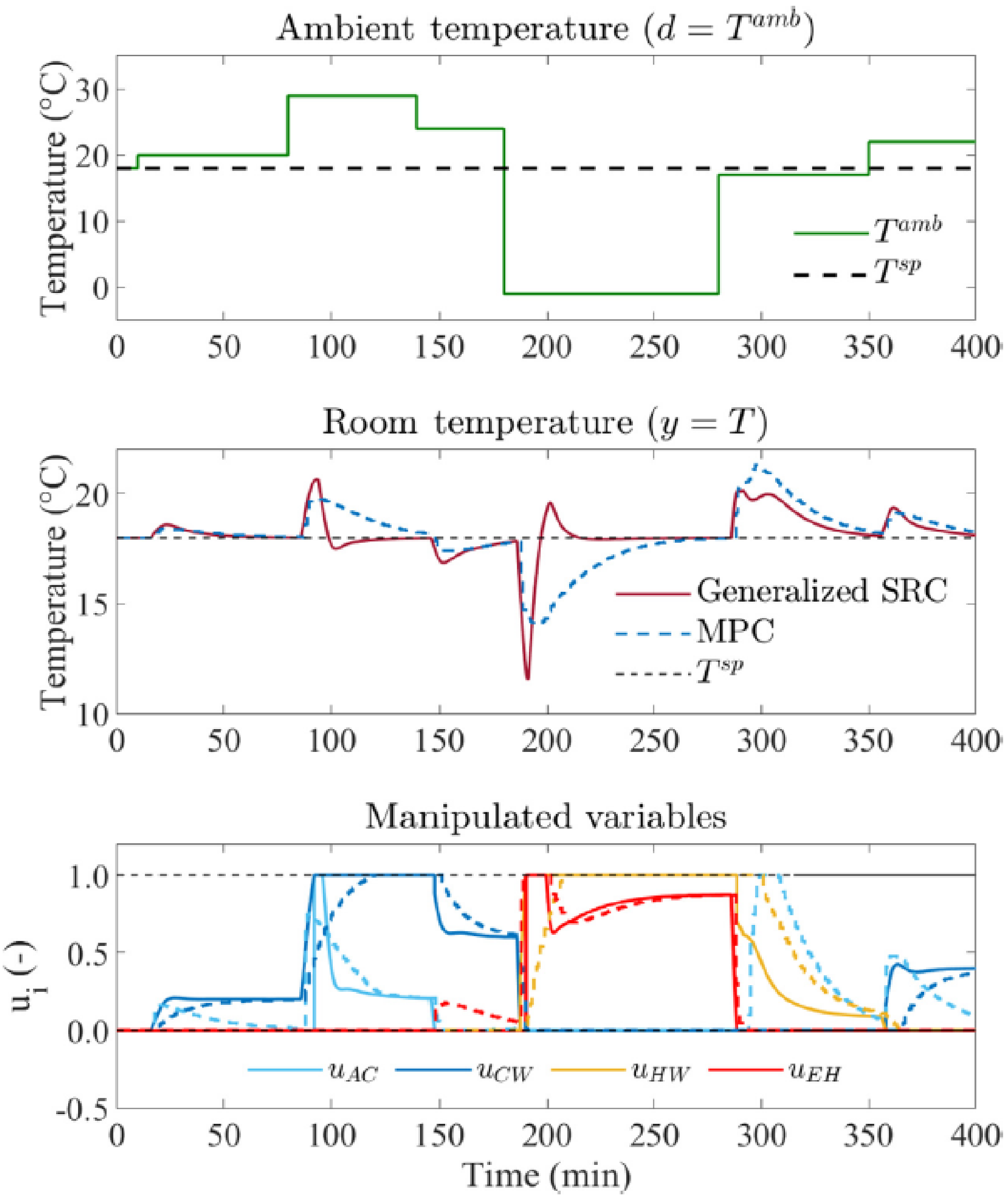
A. Reyes-Lúa and S. Skogestad. “Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy”. Journal of Process Control 91 (2020)

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



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)

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

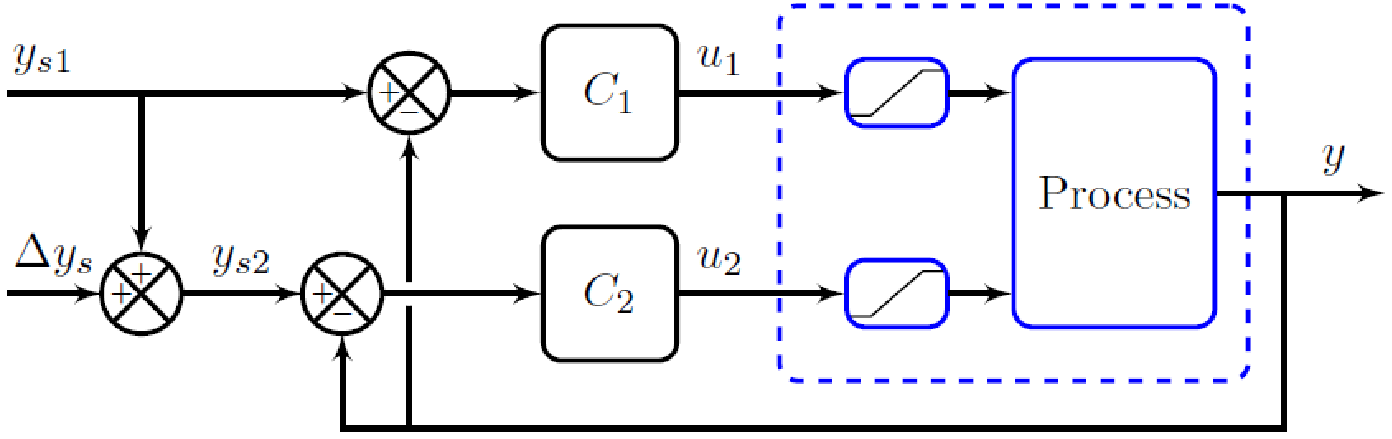


Figure 22: Separate controllers with different setpoints for MV-MV switching.

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).

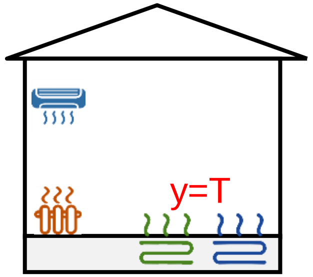
Advantages:

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:

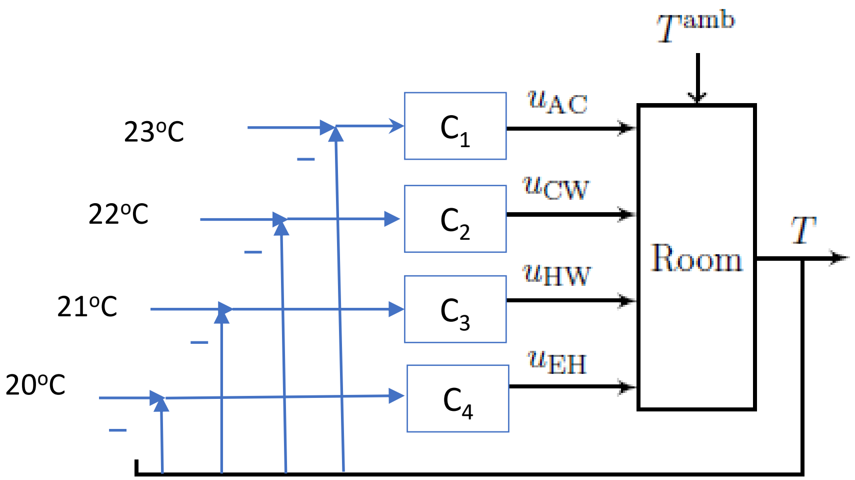
1. Temporary loose control during switching
2. Setpoint not constant
   • Can be an advantage (gives energy savings for room heating)

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

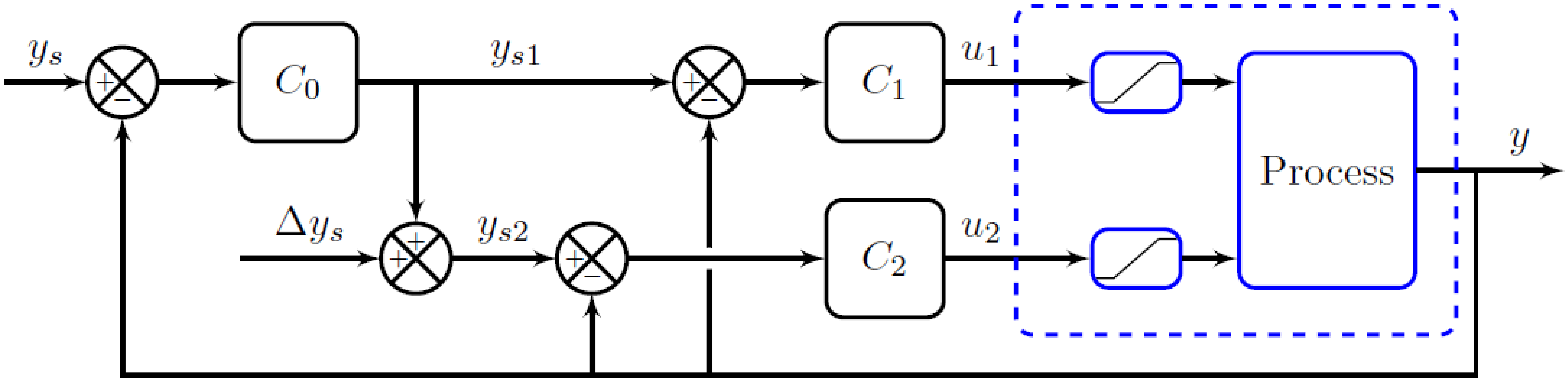


Figure 23: Separate controllers for MV-MV switching with outer resetting of setpoint. This is an extension of the scheme in Figure 22, with a slower outer controller C_0 that resets y_{1s} to keep a fixed setpoint $y = y_s$ at steady state.

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

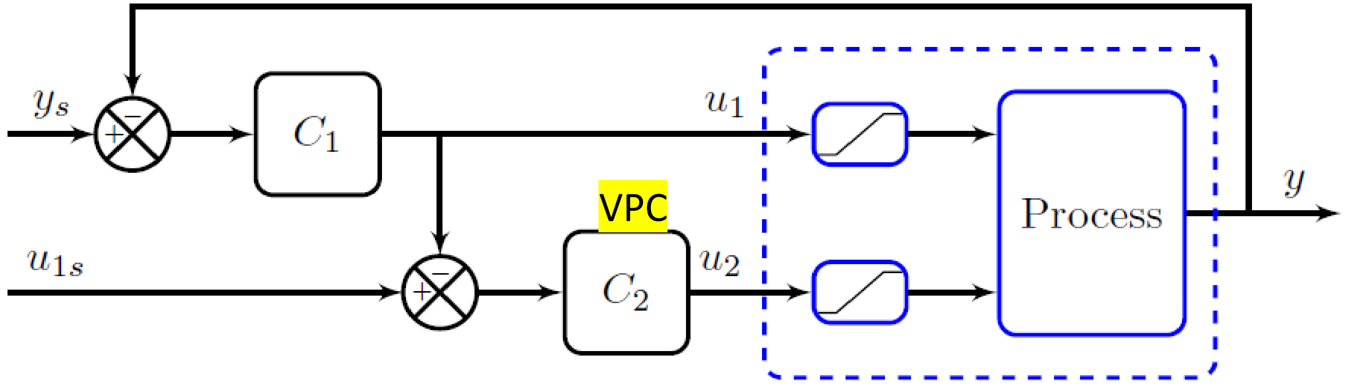


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

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)

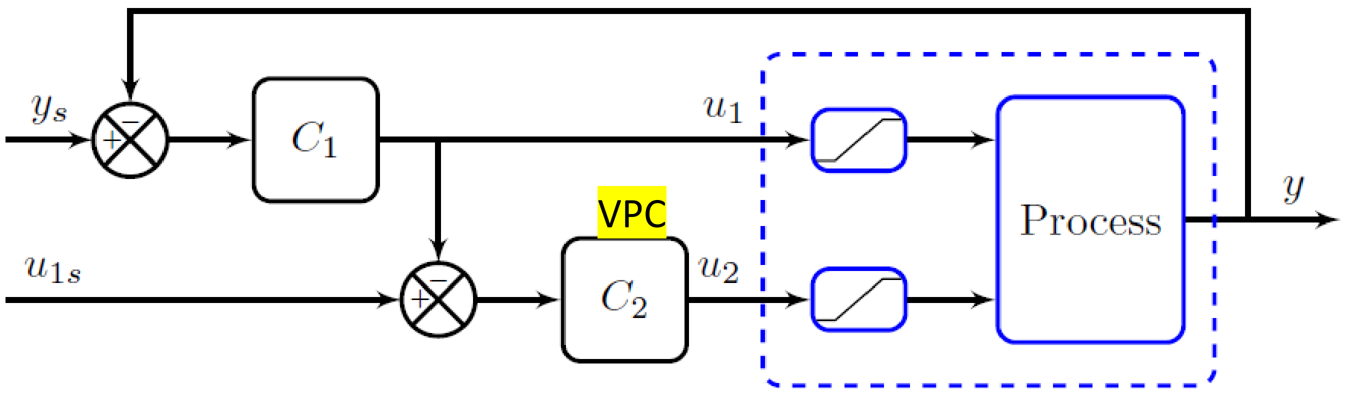


Figure 24: Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.

The VPC schemes in Figure 12 (E3 - VPC on dynamic input) and Figure 24 (E7) seem to be the same

- In fact, they are the same - except for the blue saturation elements - which tells that in Figure 24 (E7) the saturation has to be there for the structure to work as expected

But their behavior is very different!

- In Figure 12 (E3) both inputs are used all the time
  - u_1 is used to improve the dynamic response
  - u2 is the main steady-state input (and used all the time)
  - u_{1s} is typically 50% (mid-range)
- In Figure 24 (E7)
  - 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

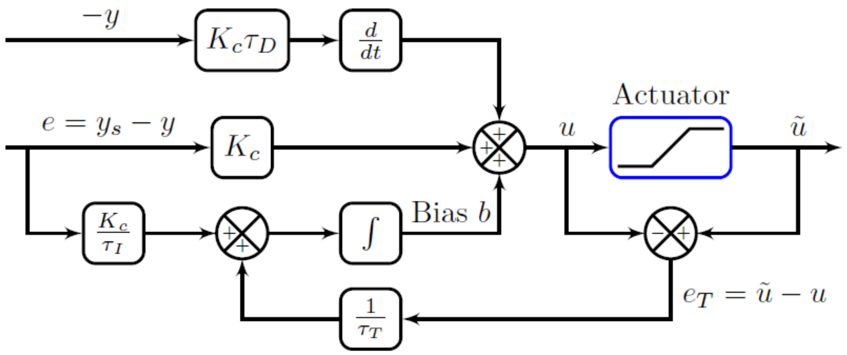


Figure 7: Recommended PID-controller implementation with anti-windup using tracking of the actual controller output ($\tilde{u}$), and without D-action on the setpoint. (Åström & Hägglund, 1988).

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

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

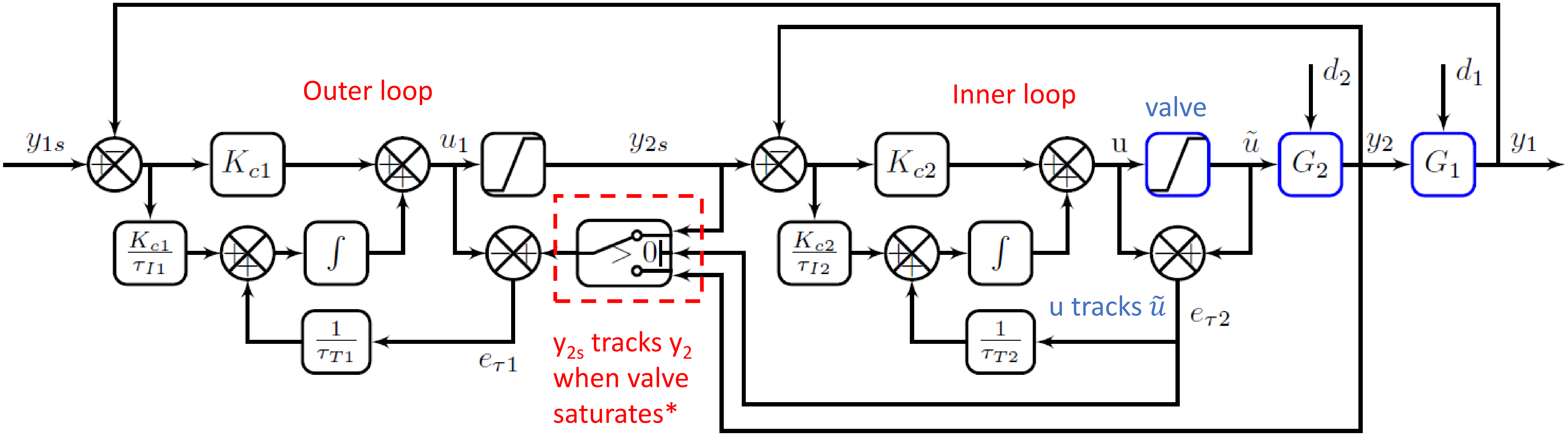


Figure 25: Cascade control with anti windup using the industrial switching approach (Leal et al., 2021).

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

E9. Two degrees-of-freedom control

- One degree-of freedom control: Controller uses $e=y_s-y$
- Two degrees-of freedom control: y and y_s used differently.
  - For example, no derivative action on setpoint,
  - More generally, setpoint filter F_s:

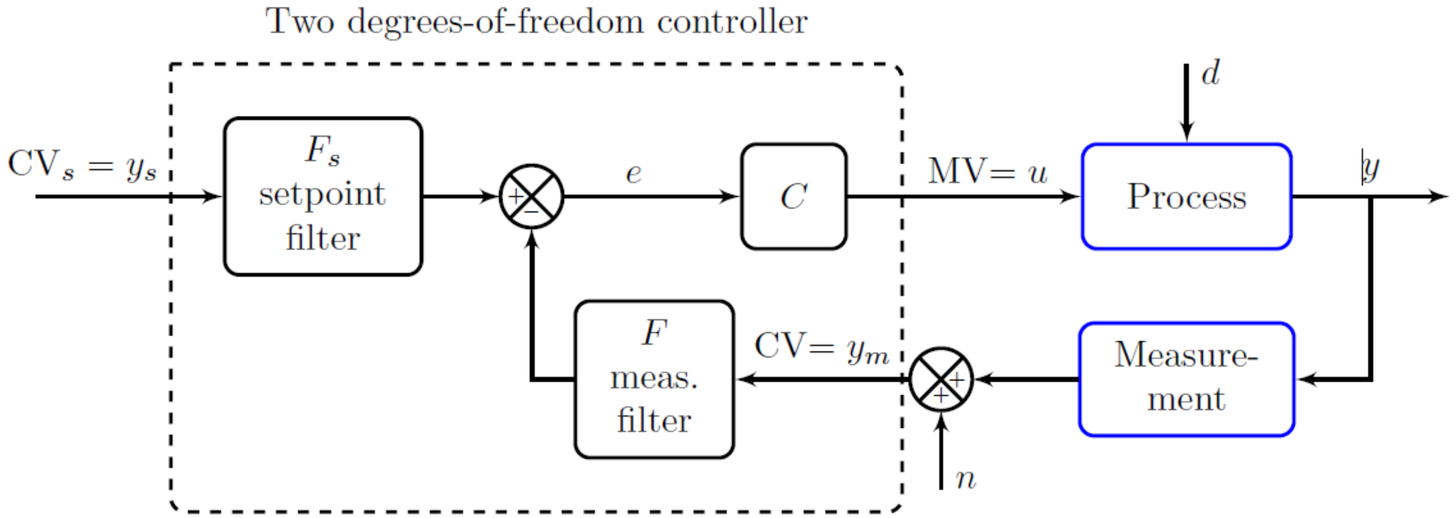
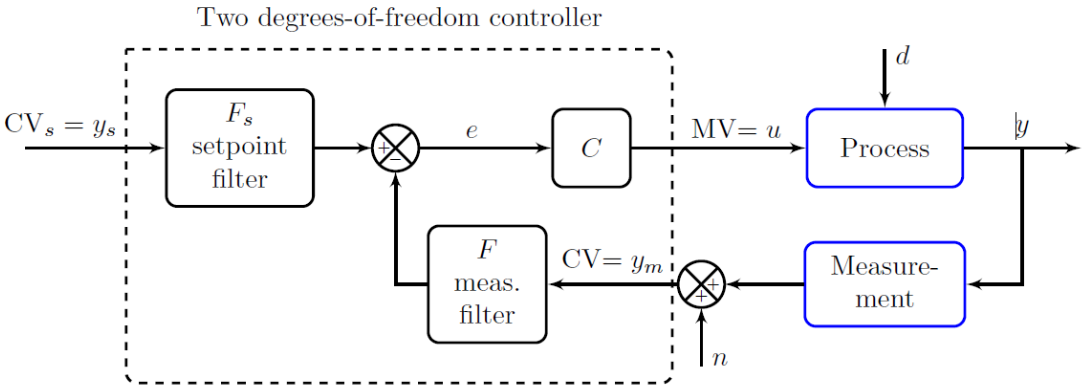


Figure A.41: Two degrees-of-freedom control system with setpoint filter F_s and measurement filter F. All blocks are possibly nonlinear.

Measurement filter F(s)



- Very common, especially with noisy measurements
  - Used also alone (without F_s)
  - Most common: First-order filter

$$F(s) = \frac{1}{\tau_F s + 1}$$

Recommended: $\tau_F \leq \frac{\tau_c}{2}$ (preferably smaller)

- τ_c: Closed-loop time constant (SIMC)

Here τ_F is the measurement filter time constant, and the inverse ($\omega_F = 1/\tau_F$) is known as the cutoff frequency. However, one should be careful about selecting a too large filter time constant τ_F as it acts as a effective delay as seen from the controller C.

King (2011) (page xii) writes in this respect: "Many engineers are guilty of installing excessive filtering to deal with noisy measurements. Often implemented only to make trends look better they introduce additional lag and can have a detrimental impact on controller performance." To reduce the effective delay (lag) introduced by filtering, Sigifredo Nino (personal email communication, 30 March 2023), who has extensive industrial experience, suggests using a second-order Butterworth filter,

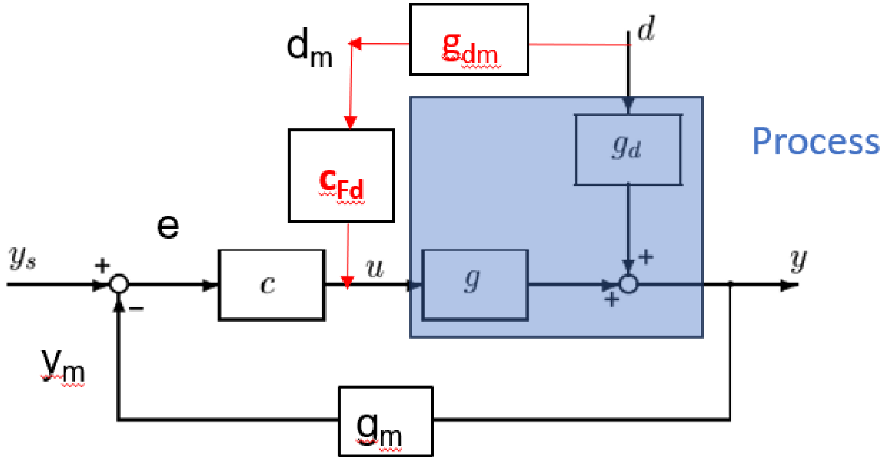
$$F(s) = \frac{1}{\tau_F^2 s^2 + 1.414\tau_F s + 1} \tag{A.4}$$

E10. Gain scheduling

- Very popular for PID within EE and ME, e.g., airplanes, automotive.
- Controller (PID) tunings change as a given function of the scheduling variable, e.g.,
  - disturbance d
  - process input u
  - process output y
  - setpoint y_s
  - control error $e=y_s-y$

E11. Feedforward control

Feedforward added to feedback (linear)



Feedforward (nonlinear) with setpoint v from feedback

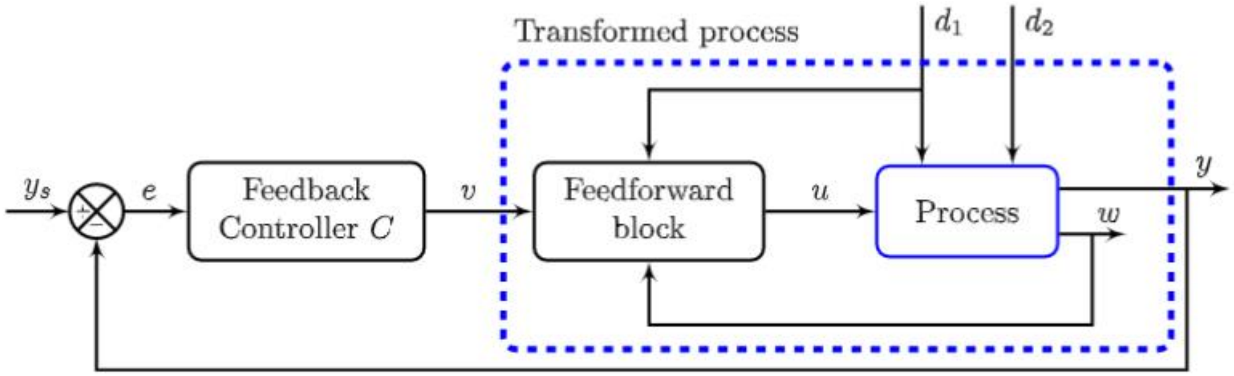


Fig. 27. Block diagram of control system with combined feedforward (often nonlinear) and feedback control (often linear). The outer feedback controller C uses the "transformed input" v to provide a feedback correction to the feedforward part.
Comment: The figure shows the possibility of treating a process measurement w in a feedfirward manner (like a measured disturbance), although strictly speaking this introduces feedback. Typically, w is a flow measurement. The main idea is that the Feedforward block is based on model inversion; e.g., see Fig. 29.

+ Many other Possible combinations

General: Feedforward block inverts process model

$$c_{Fd} = g^{-1} g_d \, g_{dm}^{-1}$$

E12. Decoupling

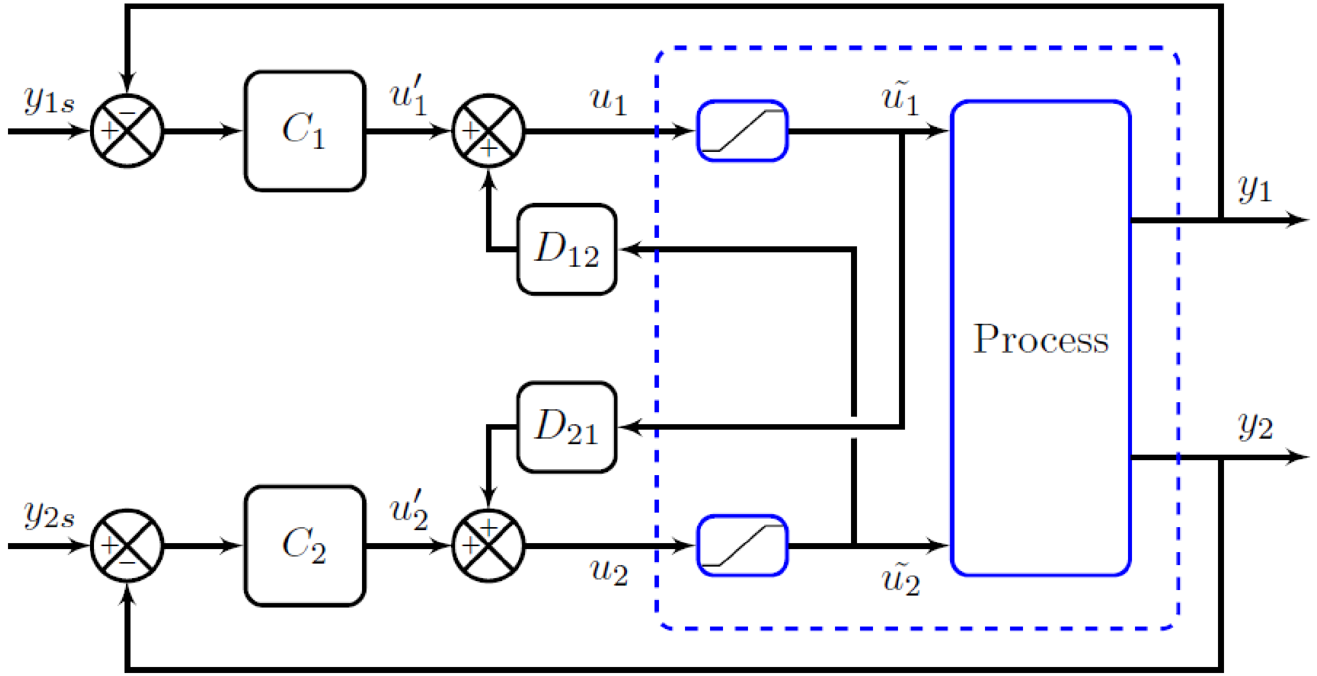


Figure 26: Linear decoupling with feedback (reverse) implementation of Shinskey (1979)

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \qquad D_{12} = -\frac{G_{12}}{G_{11}}, \qquad D_{21} = -\frac{G_{21}}{G_{22}}$$

To make the decoupling elements realizable, we need a larger (effective) delay in the off-diagonal elements than in the diagonal elements of G. This means that the "pair close" rule should be followed also when using decoupling. An alternative is to use static decoupling or partial (one-way) decoupling.

Note that Figure 26 uses the feedback decoupling scheme of Shinskey (1979) which is called inverted decoupling (Wade, 1997). Compared with the to the more common "feedforward" scheme (where the input to the decoupling elements is u' rather than $\tilde{u}$), the feedback decoupling scheme in Figure 26 has the following nice features (Shinskey, 1979):

1. With inverted decoupling, the model from the controller outputs (u') to the process outputs (y) becomes (assuming no model error) $y_1 = G_{11}u'_1$ and $y_2 = G_{22}u'_2$. Thus, the system, as seen from the controllers C_1 and C_2, is in addition to being decoupled (as expected), also identical to the original process (without decoupling). This simplifies both controller design and switching between manual and auto mode. In other words, the tuning of C_1 and C_2 can be based on the open loop models (G_{11} and G_{22}).

2. The inverted decoupling works also for cases with input saturation, because the actual inputs ($\tilde{u}$) are used as inputs to the decoupling elements.

Note that there is potential problem with internal instability with the inverted implementation because of the positive feedback loop $D_{12}D_{21}$ around the two decoupling elements. However, this will not be a problem if we can follow the "pair close" pairing rule. In terms of the relative gain array (RGA), we should avoid pairing on negative RGA-elements. To avoid the stability problem (and also for other reasons, for example, to avoid sensitivity to model uncertainty for strongly coupled processes) one may use one-way decoupling where one of the decoupling elements is zero. For example, if tight control of y_2 is not important, one may select $D_{21} = 0$.

The scheme in Figure 26 can easily be extended to 3x3 systems and higher

E13. Linearization elements

- Typically, logarithm or nonlinear feedforward blocks
- General approach: See Input transformations

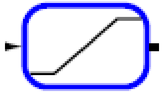
E14. Calculation block based on transformed input

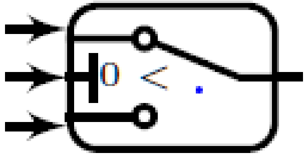
- SEE LATER

E11. Simple static estimators

- Inferential element
- Soft sensor

Additional standard elements

- E16. Simple nonlinear static elements
  - Multpliction
  - Division (avoid or at least be careful)
  - Square root
  - Dead zone
  - Dead band
  - Limiter (saturation element) 
  - On/off
- E17. Simple linear dynamic elements
  - Lead-lag filter
  - Time delay
  - … more…
- E18. Standard logic elements
  - If, then, else
  - Example: Select depending on sign of another signal:



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).

Standard advanced control elements studied in this paper.

Control element	Main use	Inputs	Outputs
E1. Cascade control Figs. 9 and 10	Linearization and local disturbance rejection	Outer master controller: • CV_{1s}-CV Inner controller: • CV_{2s}-CV_2	Outer master controller: • CV_{2s} Inner controller: • MV
E2. Ratio control Fig. 11	Feedforward or decoupling without model (assumes that scaling property holds)	• R (desired ratio) • DV or MV_1	• MV = R · DV, or • MV_2 = R · MV_1
E3. VPC on extra dynamic input Fig. 12	Use extra dynamic input MV_1 to improve dynamic response (because MV_2 alone is not acceptable). MV_1 setpoint is unconstrained (mid-range) and controlled all the time	• MV_{1s} - MV_1	• MV_2
E4. Selector Figs. 17, 18 and 19	CV-CV switching: Many CVs (CV_1, CV_2, ...) controlled by one MV	• MV_1 • MV_2, ... (generated by separate controllers for CV_1, CV_2, ...)	• MV = max/min (MV_1, MV_2, ...)
E5. Split-range control Figs. 21 and 23	MV-MV switching: One CV controlled by sequence of MVs (using only one controller)	• CV_s-CV	• MV_1 • MV_2, ...
E6. Separate controllers with different setpoints Fig. 22	MV-MV switching: One CV controlled by sequence of MVs (using individual controllers with different setpoints)	• CV_{s1} − CV • CV_{s2} - CV ...	• MV_1 • MV_2 ...
E7. VPC on main steady-state input Fig. 24	MV-MV switching: One CV controlled by main MV_1 with use of extra MV_2 to avoid saturation of MV_1. MV_1 setpoint is close to constraint and only controlled when needed	• MV_{1s} - MV_1	• MV_2
E9. Two degrees-of-freedom feedback controller Fig. A.41	Treat setpoint (CV_s) and measurement (CV) differently in controller C	• CV_s • CV	• MV
E11. Feedforward control Fig. A.42	Reduce effect of disturbance (using model from DV and MV to CV)	• DV	• MV
E12. Decoupling element Fig. 26	Reduce interactions (using model from MV_1 and MV_2 to CV)	• MV_1 • MV_2	• MV_2 • MV_1
E14. Calculation block based on transformed input Fig. 27	Static nonlinear feedforward, decoupling and linearization based on nonlinear model from MV, DV and w to CVv	• Transformed input = feedback trim (v) • DV (d) • Extra meas. (w)	• MV (u)

Another smart invention: Cross-limiting control

Industry also makes use of other smart solutions, which do not follow from the standard structures presented in this paper.

One example is cross-limiting control for combustion, where the objective is to mix air (A) and fuel (F) in a given ratio, but during dynamic transients, when there will be deviations from the given ratio, one should make sure that there is always excess of air. The scheme in Fig. 39 with a crossing min- and max-selector achieves this. It is widely used in industry and is mentioned in many industrial books (e.g., Liptak (1973), Nagy (1992) and Wade (2004)). The setpoint for the ratio, $(F_F/F_A)_s$, could be set by a feedback controller (not shown) which controls, for example, the remaining oxygen after the combustion.

The selectors in Fig. 39 are used to handle the dynamic (transient) case, so this is a somewhat rare case where the selectors are not performing a steady state CV-CV switch.

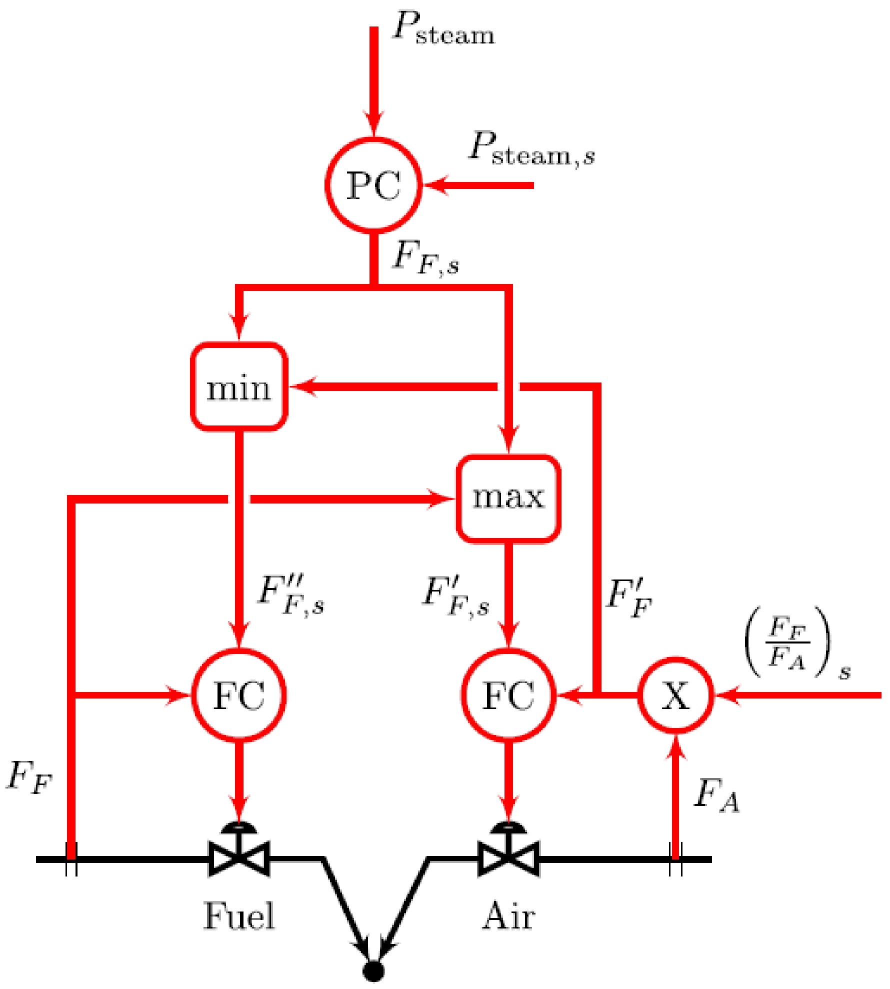


Fig. 39. Cross-limiting control for combustion where air (A) should always be in excess to fuel (F).

Change of active constraints. Four 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. 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 «repair loops»)

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

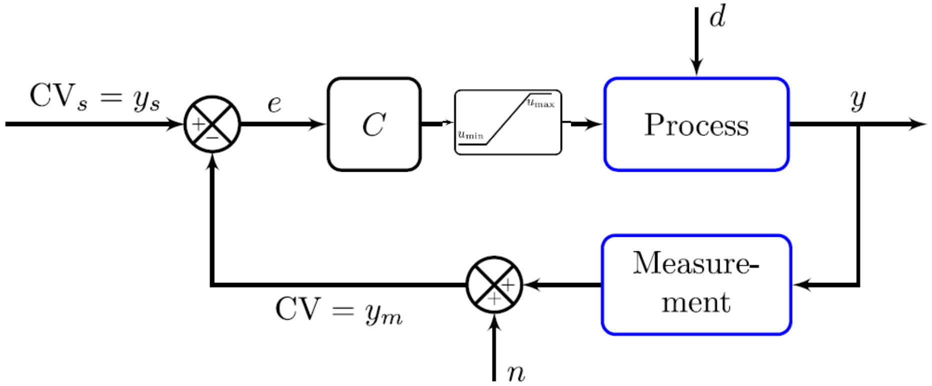
Adriana Reyes-Lua and Sigurd Skogestad, Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures, Ind.Eng.Chem.Res, 2020

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

C. Simple MV-CV switching

- When MV (u) saturates, we can give up the CV (y)
- If we follow the «input saturation rule» and pair these two variables then we have
  - Simple MV-CV switching
  - Don't need to do anything, except having anti-windup in c

- Many examples (that it works is not always so obvious!)
  - 1. Driving as fast as possible to the airport
    - u=power $\leq u_{max}$
    - y = speed $\leq y_{max}$
    - $y_s = y_{max}$ = 90 km/h
    - "If we reach max power, we must give up controlling y"
  - 2. Heating of cabin in the winter
    - u=power ≥ 0
    - y = temperature $\geq$ 8C
    - $y_s = y_{min}$ = 8C
    - "If we reach min. power (0), then it is hot outside - and there is no need to control y"
  - 3. Anti-surge control
    - u =bypass ≥ 0,
    - y = flowrate $\geq y_{min}$
    - $y_s = y_{min}$
    - "If we reach min. bypass (0), then the feedrate is it is larger than y_{min} - and there is no need to control y"



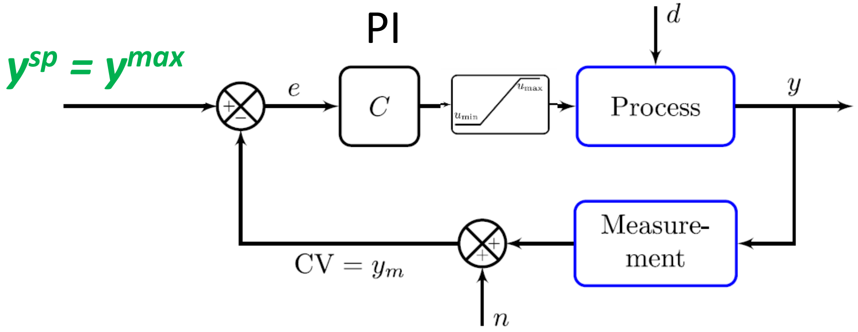
NTNU

Optimization with PI-controller

max y

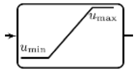
s.t. $y \leq y^{max}$

$u \leq u^{max}$



Example: Drive as fast as possible to airport (u=power, y=speed, y^{max} = 110 km/h)

- **Optimal solution has two active constraint regions:**
  1. $y = y^{max}$ → **speed limit**
  2. $u = u^{max}$ → **max power**



- **Solved with PI-controller**
  - $y^{sp} = y^{max}$
  - Anti-windup: I-action is off when $u=u^{max}$

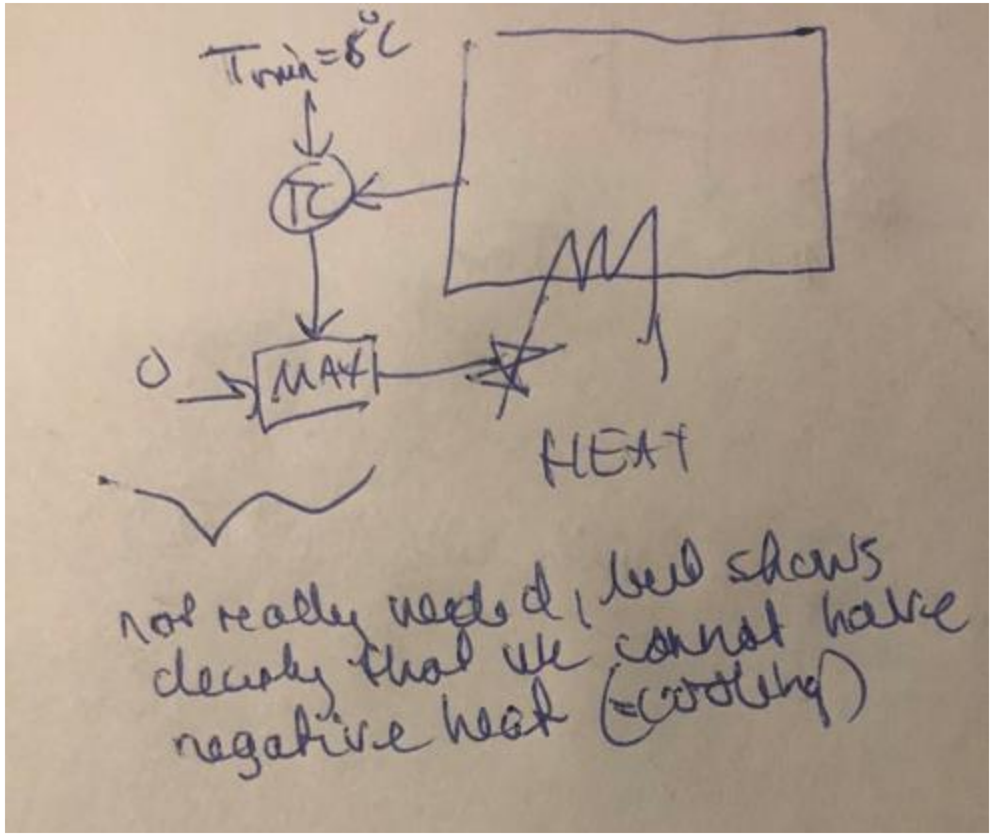
s.t. = subject to

y = CV = controlled variable

Avoid freezing in cabin

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

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



Anti-surge control

Keep minimum flow F_{min} for pump or compressor using recycle valve.

If the flow is large then the recycle valve will close (MV=0), but this does not matter as the constraint is over-satisfied.

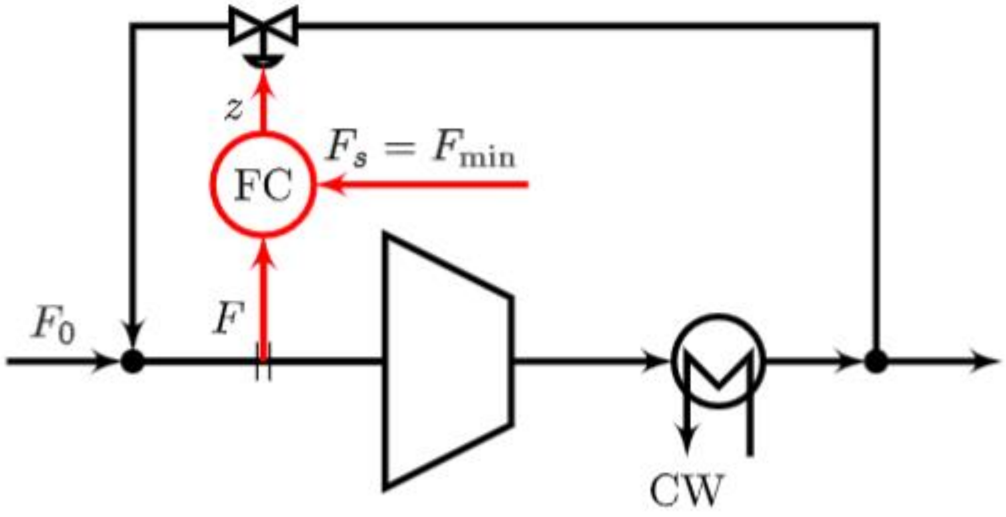


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding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

We satisfy the input saturation rule:
«When the MV (u) saturates, control of the CV (y) can be given up»

A. MV-MV switching

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

Alt.1 Split-range control (one controller)

- 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

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

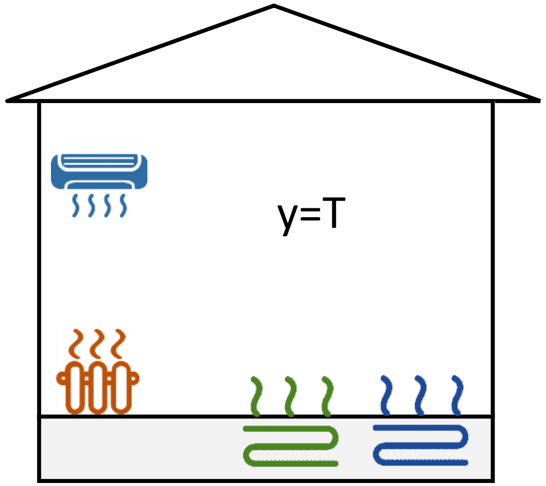
Alt.3 Valve position control

- 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 ,**

Summary Example: Room heating with 4 MVs



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

Alternatives (in addition to MPC):

1 Split range control (SP=22C)

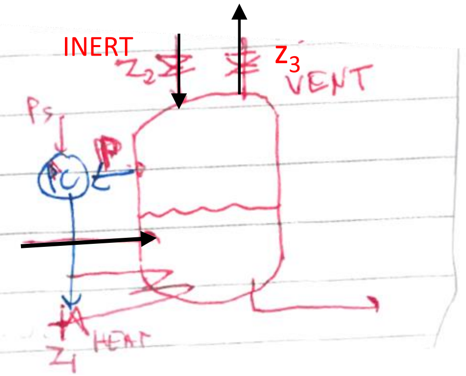
1G. Generalized split range control (SP=22C)

2. Controllers with different setpoint values (SP=24C, 23C, 22C, 21C)

3. Valve position control (not recommended here)
   - Use always MV=EH to control y=T with SP=22C
   - Combine with VPC: Control EH to 10% with the three other MVs (complicated…)

Example MV-MV switching: Pressure control
(Alt. 3 may be the best in this case)

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



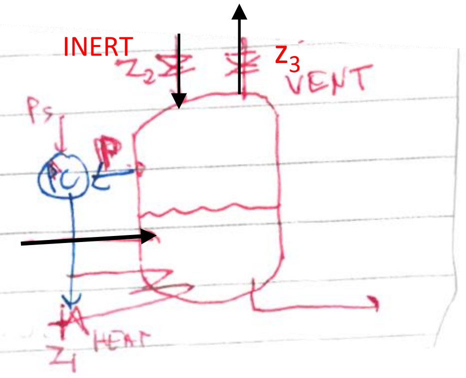
Normal: Control CV=p using MV1=Q
- but if Q=0 (because of too hot feed) we must use MV3=vent
- and if Q=max (becase of too cold feed) we must use MV2=inert

Example: Heating water to 220C. Control pressure at 20 bar.
«Inert» could be HP steam.

Example MV-MV switching: Pressure control

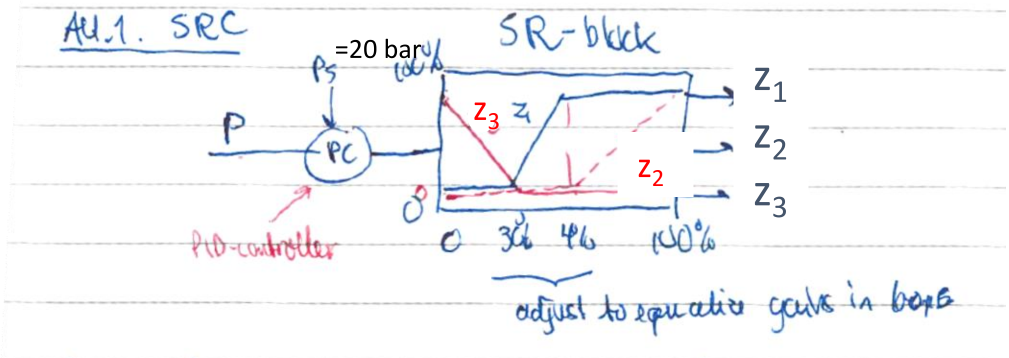
(Alt. 3 may be the best in this case)

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



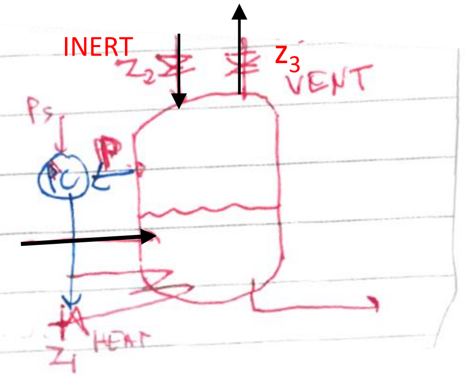
Normal: Control CV=p using MV1=Q

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

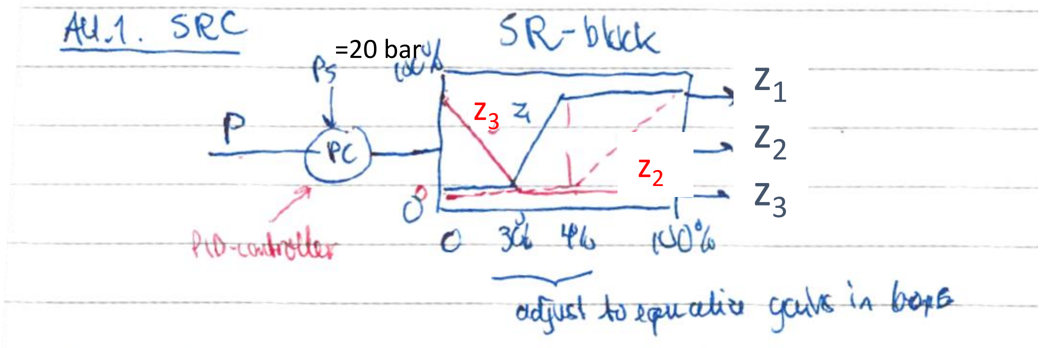


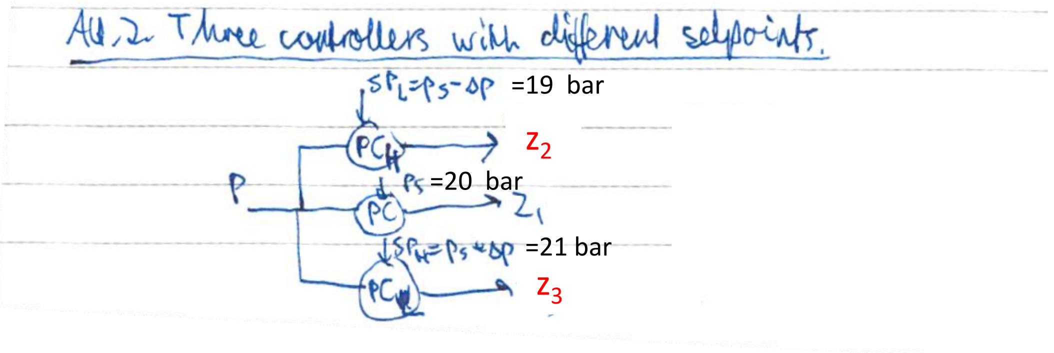
Example MV-MV switching: Pressure control
(Alt. 3 may be the best in this case)

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



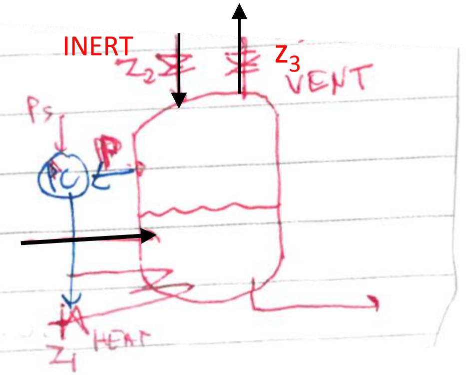
Normal: Control CV=p using MV1=Q
- but if Q=0 we must use MV3=vent
- and if Q=max we must use MV2=inert



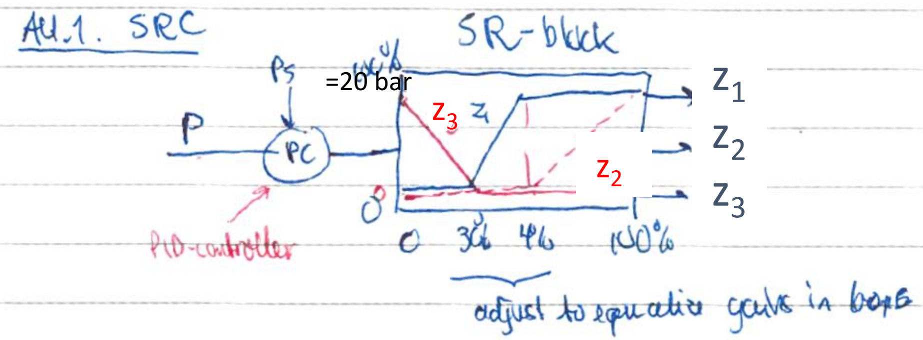


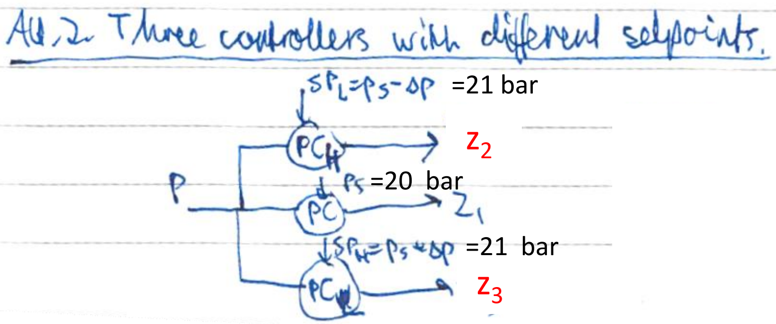
Example MV-MV switching: Pressure control
(Alt. 3 may be the best in this case)

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

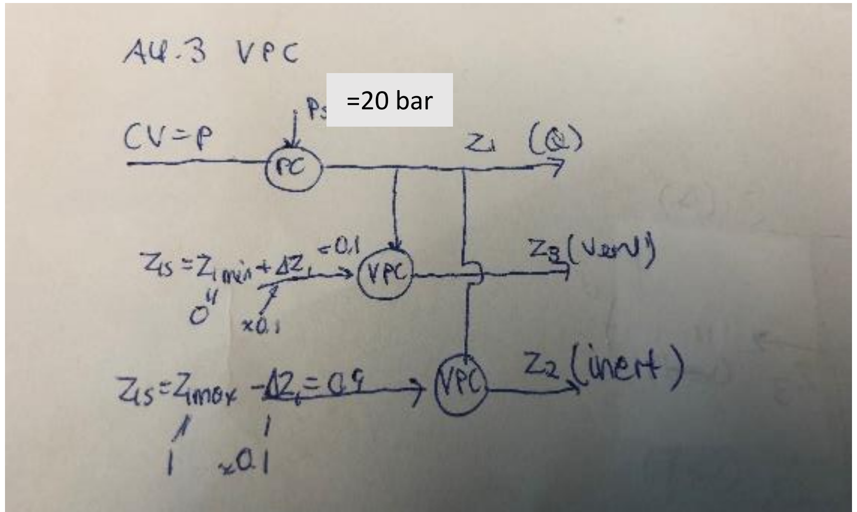


Normal: Control CV=p using MV1=Q
- but if Q=0 we must use MV3=vent
- and if Q=max we must use MV2=inert





Alt.3: VPC (z2 and z3 could here even be on/off valves)
Always use Q (z1) to control p.
Need two VPC's:
- Use vent (z3) to avoid Q small (z1=0.1)
- Use inert (z2) to avoid Q large (z1=0.9)
- z2=0 and z3=0 when 0.1<z1<0.9



B. CV-CV switching

- Use selector

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

MV = z ≥ 0
CV_1 = F $\geq F_{min}$
CV_2 = $F_0 \leq F_{0,max}$

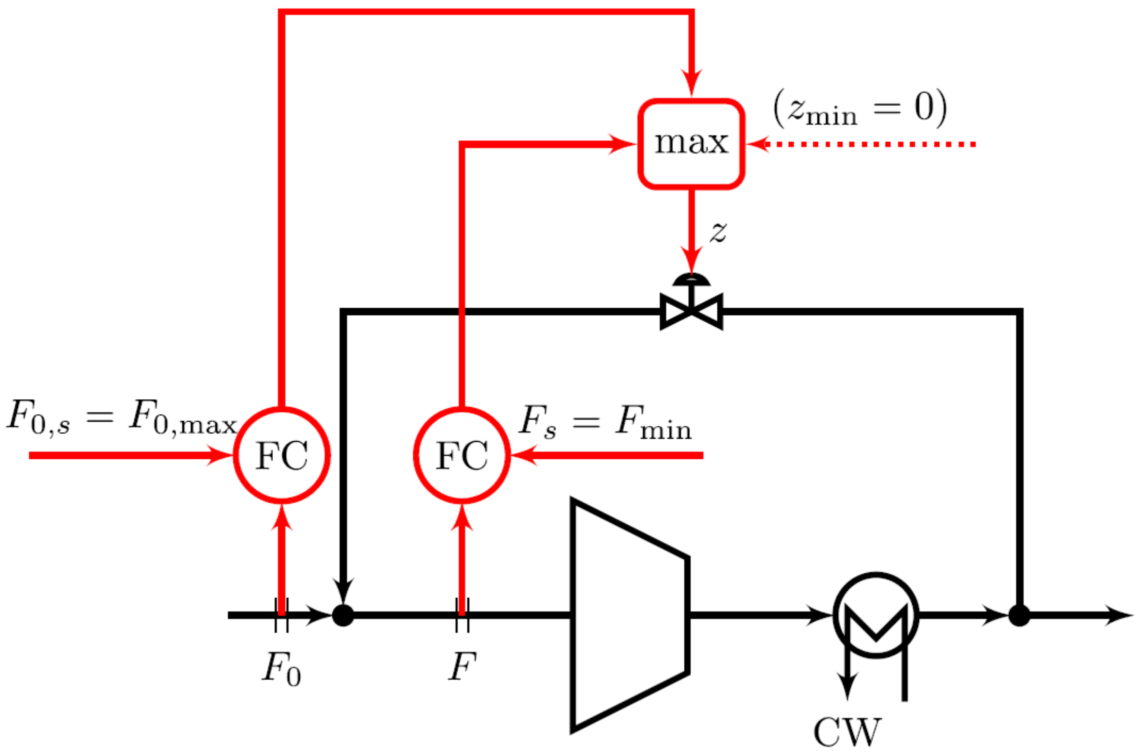


Fig. 33. Anti-surge compressor control with two CV constraints. This is an example of simple MV-CV-CV switching. MV = z, $CV_1 = F$, $CV_2 = F_0$ (all potentially active constraints).

Example. Maximize flow with pressure constraints

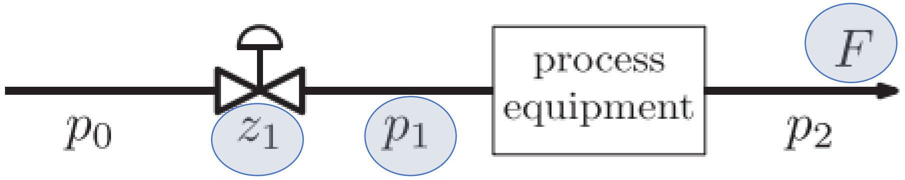


Fig. 6. Example 2: Flow through a pipe with one MV ($u = z_1$).

Input u = z_1

Want to maximize flow, J=-F:

Optimization problem is:

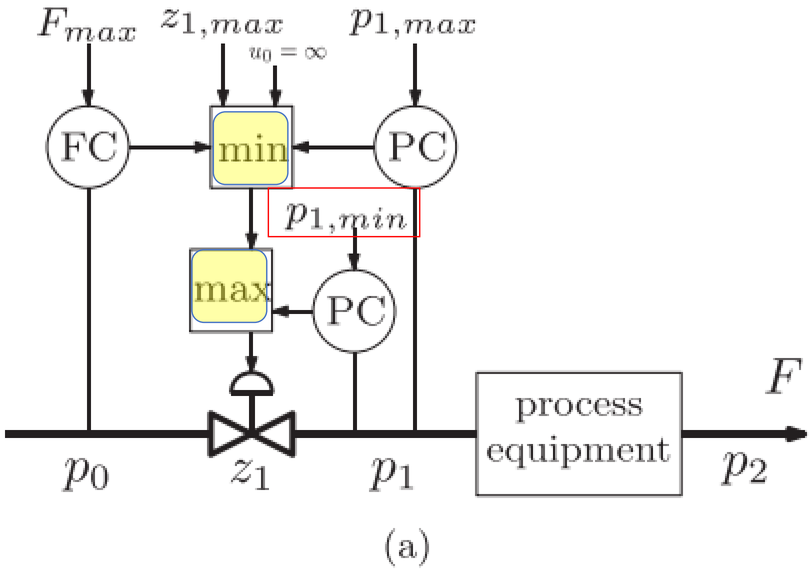
$$\max_{z_1} F$$

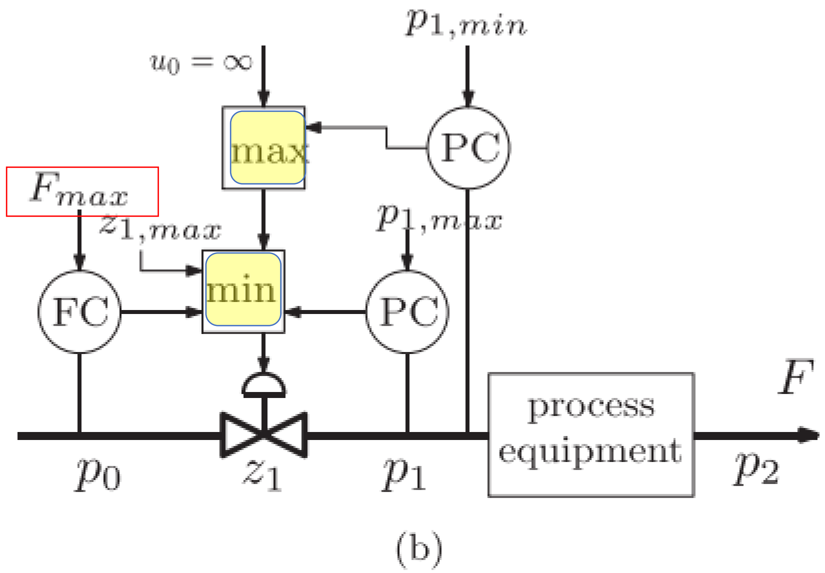
s.t.

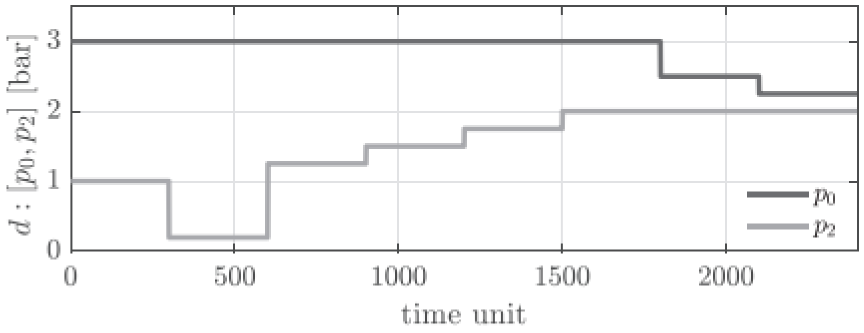
$$\begin{aligned} F &\leqslant F_{max} \\ p_1 &\leqslant p_{1,max} \\ p_1 &\geqslant p_{1,min} \\ z_1 &\leqslant z_{1,max} \end{aligned} \qquad (15)$$

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

B. CV-CV switching

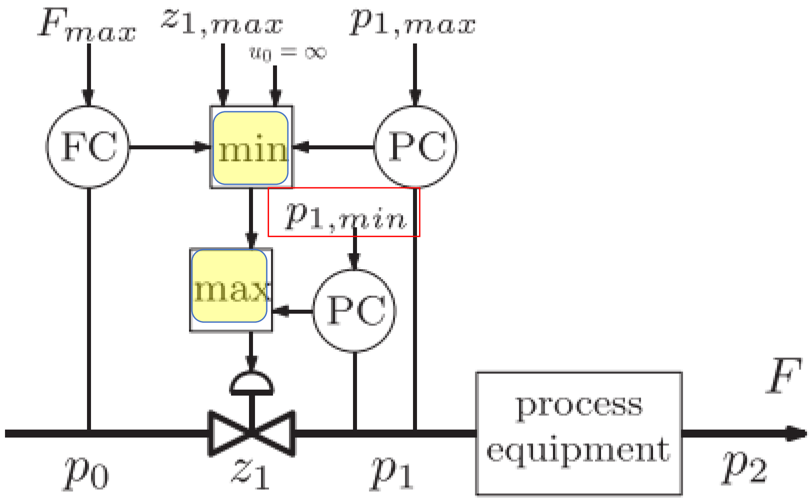




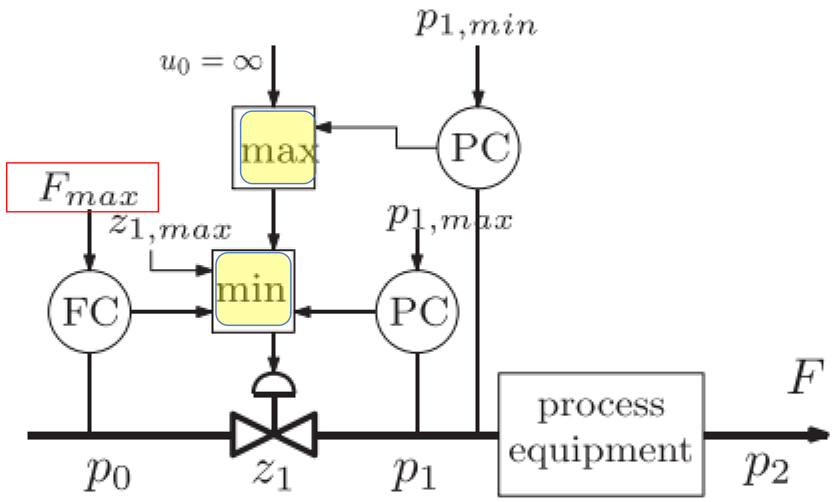


Disturbances in p_0 and p_2 (unmeasured)

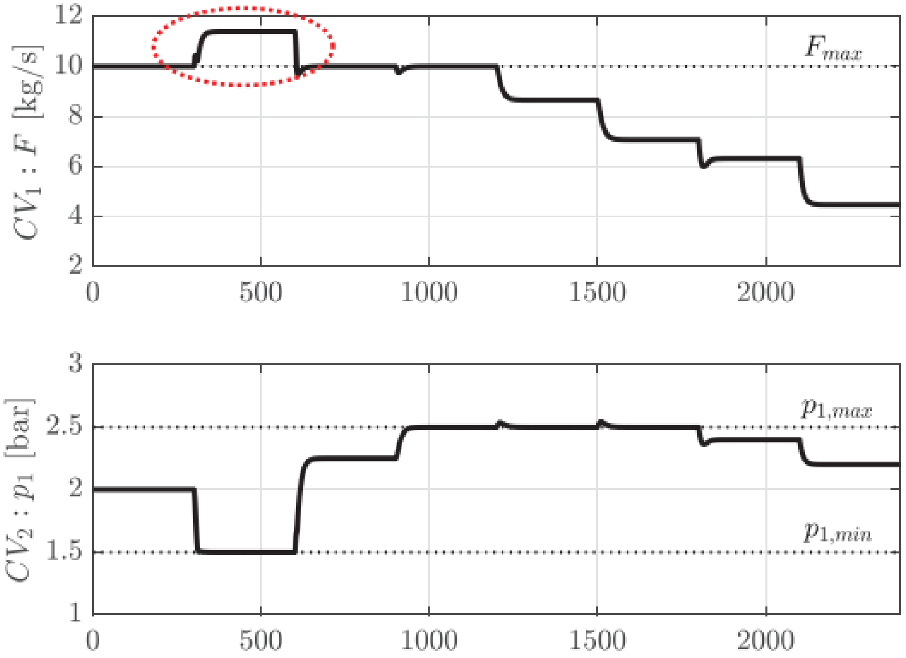
B. CV-CV switching

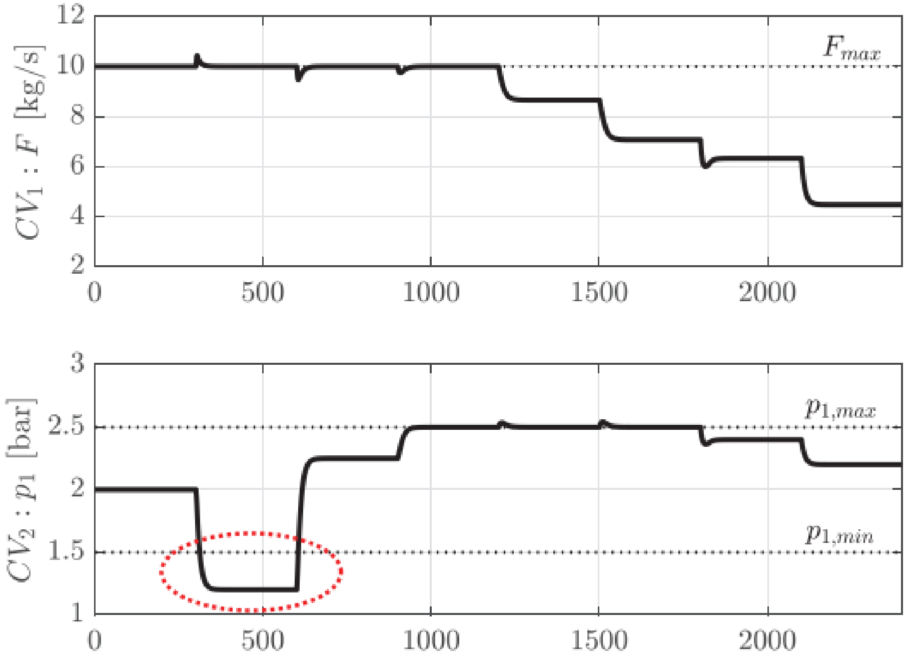


(a)



(b)





t>1800: u=zmax=1

MV-CV switching (because of constraint on MV)

- **C. Simple MV-CV switching**
  - Don’t need to do anything if we paired the MV with a CV that can be given up
- **D. Complex MV-CV switching**
  - This is a repairing of loops
  - Need to combine MV-MV switching with CV-CV-switching
- The CV-CV switching always uses a selector
- As usual, there are three alternatives for the MV-MV switching:
  1. Split range control (block /\): Has problems because limits may change
  2. Different setpoints for level. Actually, may have additional advantages
  3. Valve position control («long loop»). avoids repairing.

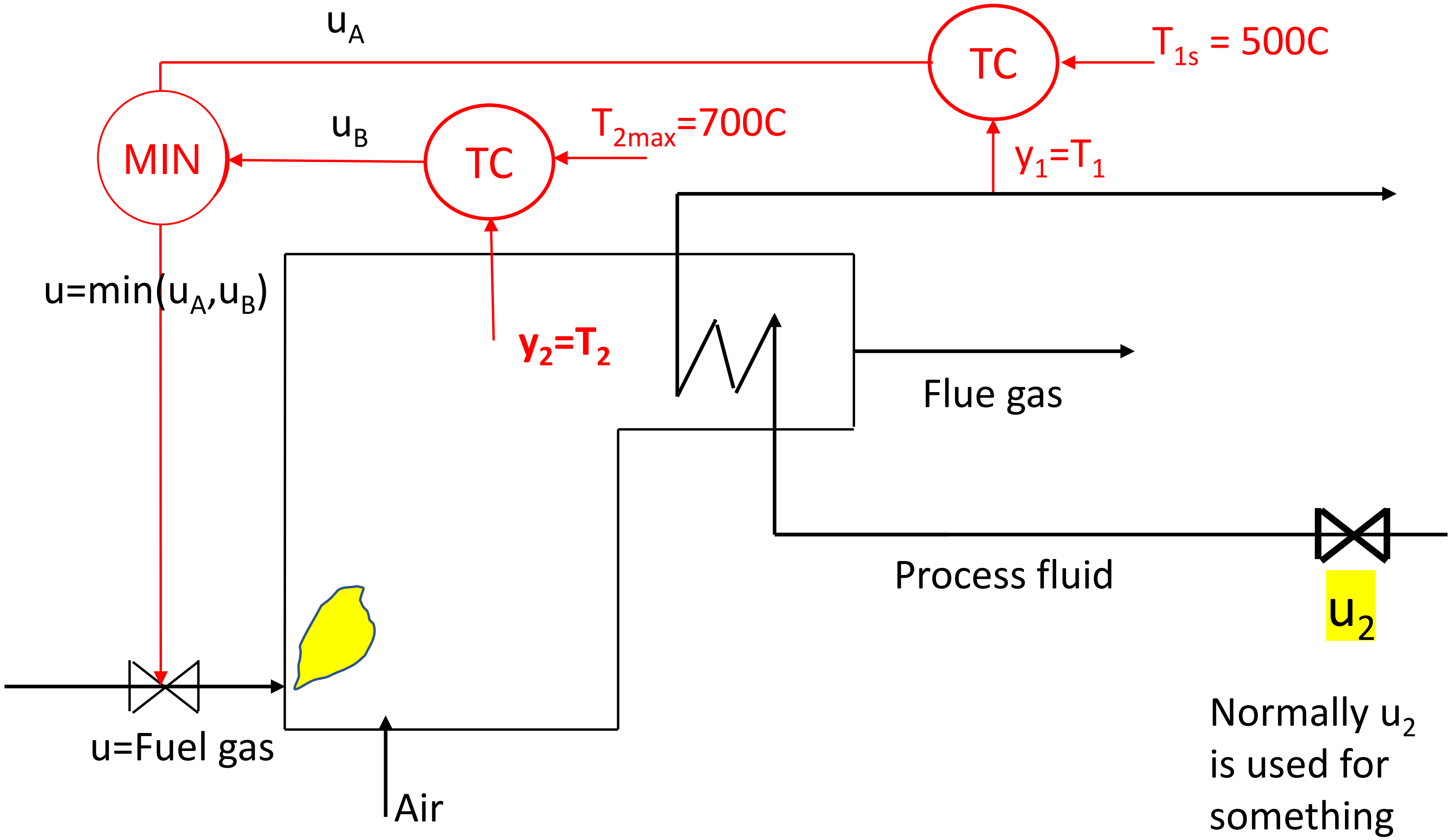
Furnace control : Cannot give up control of $y_1=T_1$. What to do?

Inputs (MV)

- u = Fuel gas flowrate
- u_2 = Process flowrate

Output (CV)

- y_1 = process temperature T_1 (with desired setpoint)



Normally u_2 is used for something else

Cannot give up controlling T_1

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

Inputs (MV)

- u = Fuel gas flowrate
- u_2 = Process flowrate

Output (CV)

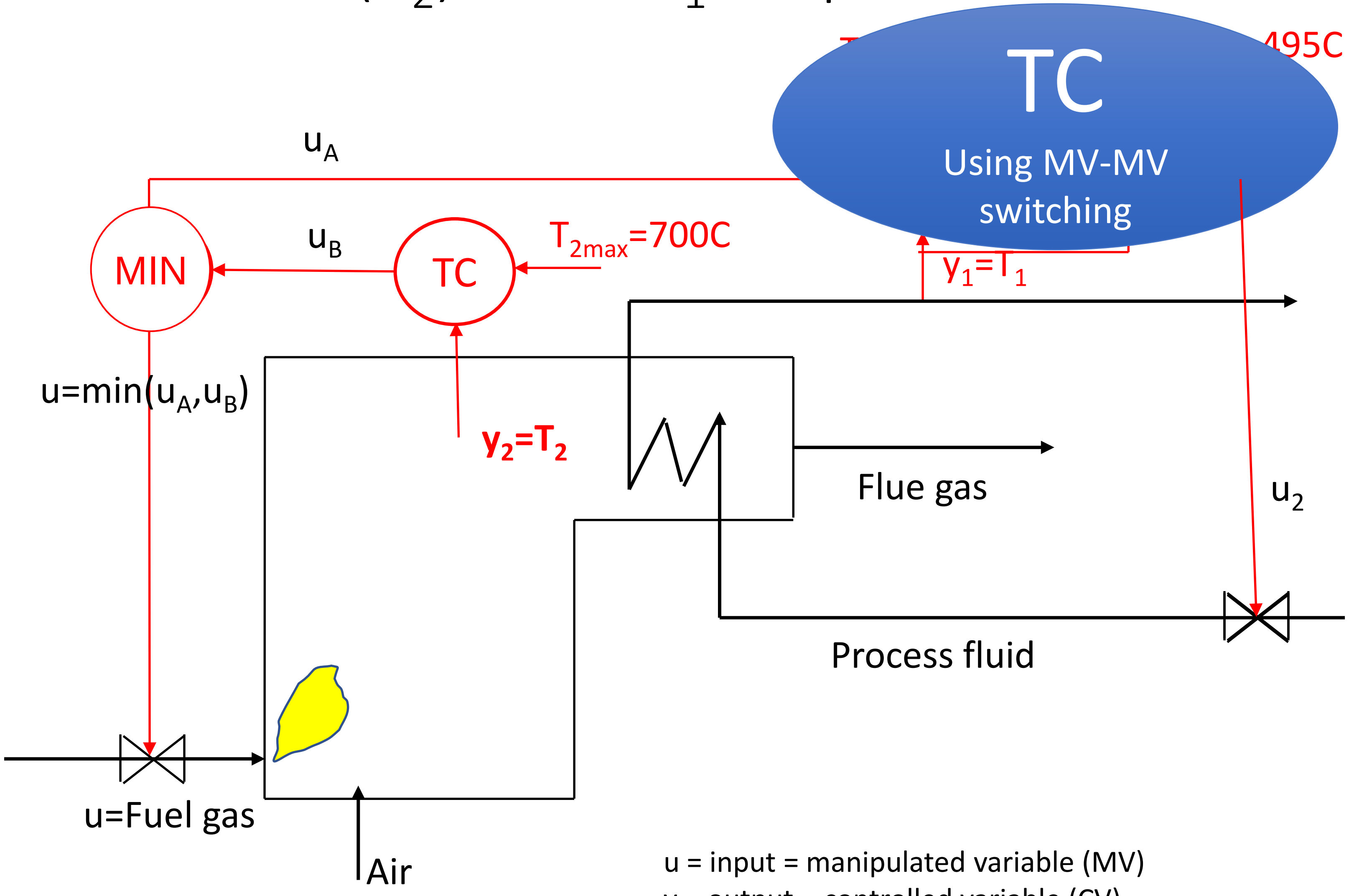
- y_1 = process temperature (with desired setpoint)

Note: Standard Split Range Control (Alt. 1) is not good here.

Could be two reasons for too little fuel

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

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



u = input = manipulated variable (MV)

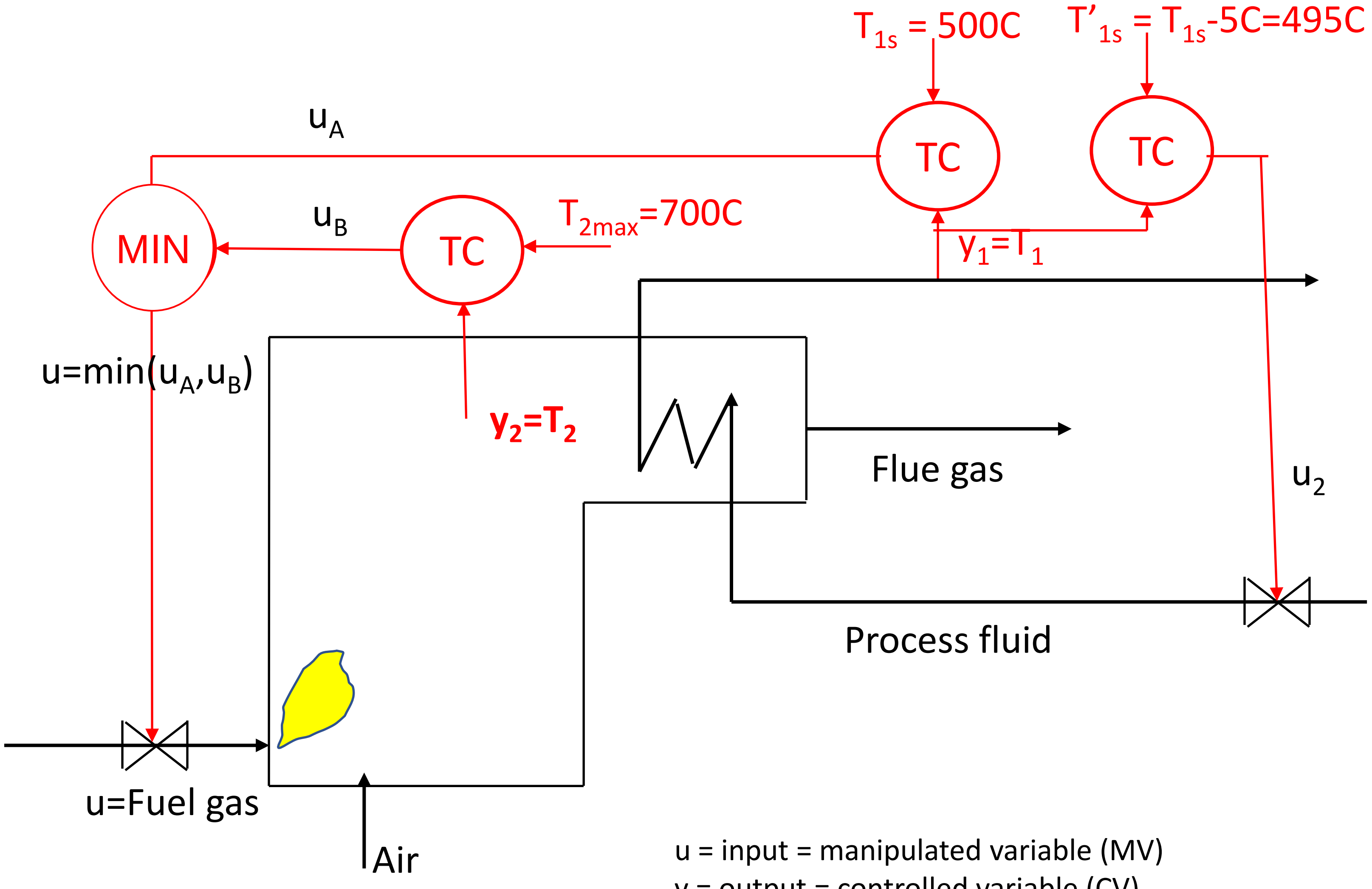
y = output = controlled variable (CV)

Use Alt. 2: Two controllers

Inputs (MV)
- u = Fuel gas flowrate
- u2 = Process flowrate

Output (CV)
- y_1 = process temperature (with desired setpoint)

- Solution: Two controllers with different setpoints



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

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

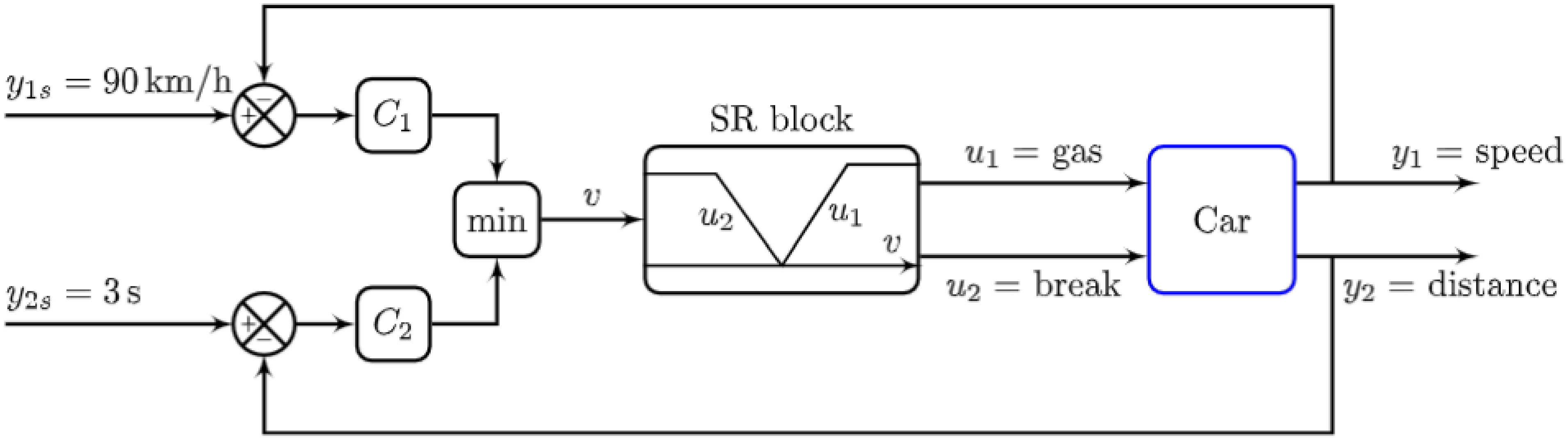


Fig. 31. Adaptive cruise control with selector and split range control.

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

Examples: Important insight

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

Comment on need for rules

- The human brain (at least mine) has problems in analyzing even quite simple cases
- Two «simple» cases are:
  - choice of max- and min- selectors
  - how to get consistent inventory control
- I frequently need to og back to the «selector rules» or the «radiation rule» to get this right.

Quiz. Is this OK?

Cristina: I am looking at a control solution using selectors for keeping the pressure within constraints, while maximizing the valve opening. See figure This is for the valve before a steam turbine. This should work OK, right? Of course, if pmax is reached while the valve is fully open, the turbine bypass will have to open.

Answer:
Rule 1. Yes, the rule is to use a max-selector for a constraint which is satisfied with a large input. And since the pressure is measured upstream, the pressure will get lower if we increase the valve opening, making it easier to satisfy the pmax-constraint. So yes, this is OK.

Rule 1. Similar for the min-block with pmin.

Rule 2. Since you have two constraints on the same variable, you cannot have infeasibility so the order of the min. and max-blocks doesn't matter for pmin and pmax.

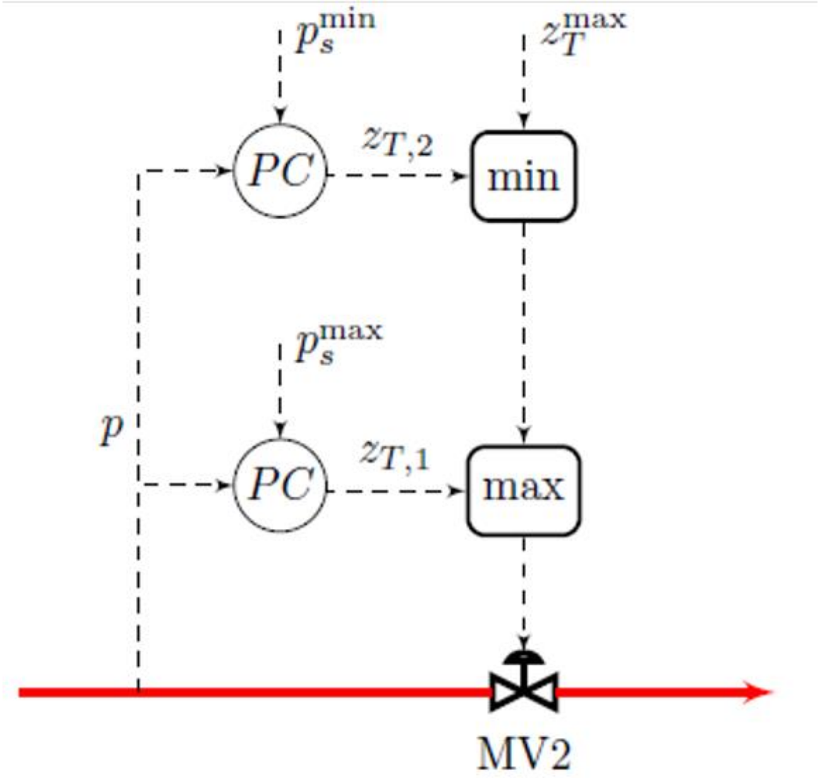
Rule 2. Yes, the desired value uo=zmax should always enter the first block.

Conclusion: yes, it works.

BUT….
Comment 1: But note that there is also a "hidden" min-selector just before the valve because of the valve which has zmax. And also a "hidden" max-selector because of zmin (a fully closed valve). These constraints may be inconsistent with the pressure constraints.

Comment 2: Since the order of the two selectors does not matter in this case, one may instead use the "equivalent" alternative with the max-block first. But we then see clearly that the constraint on pmax will never be activated, because ztmax is large. I guess this makes sense since you want to have the valve as open as possible, so then the you will always be at the pmin-constraint or have a fully open valve. So you can cut the pmax-constraint (and thus the max-selector) as you anyway want to open the valve as much as possible.

In addition, you can also cut the min-selector because there is already a "hidden" min-selector with zmax. (On the other hand, it will not be wrong to keep them.)



Final conclusion: Yes, it works, but it's much too complicated.

- **All what is shown can be replaced by a pressure controller (PC) with setpoint p^{min}.**

Challenges selectors

- Standard approach requires pairing of each active constraint with a single input
  - May not be possible in complex cases
  - See part 6 (RTO/feedback-based RTO)
- 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

Systematic design of advanced regulatory control (ARC)

- First design simple control system for nominal operation
  - With single-loop PID control we need to make pairing between inputs (MVs) and outputs (CVs):
  - Should try to follow two rules
    1. «Pair close rule» (for dynamics).
    2. «Input saturation rule»:

Then: design of switching schemes

- Make a list of possible new contraints that may be encountered (because of disturbances, parameter changes, price changes)
- Reach constraint on new CV
  - Simplest: Find an unused input (simple MV-CV switching)
  - Otherwise: CV-CV switching using selector (may involve giving up a CV-constraint or a self-optimizing CV)
- Reach constraint on MV (which is used to control a CV)
  - Simplest (If we followed input saturation rule):
    - Can give ip controlling the CV (Simple MV-CV switching)
    - Don't ned to do anything
  - Otherwise (if we cannot give up controlling CV)
    - Simplest: Find an unused input
      - MV-MV switching
    - Otherwise: Pair with a MV that already controls another CV
      - Complex MV-CV switching
      - Must combine MV-MV and CV-CV switching
- Is this always possible? No, pairing inputs and outputs may be impossible with many constraints.
- May then instead use RTO or feedback-RTO
- Maybe MPC?