Part 6: Advanced/supervisory control layer

- Skogestad procedure for control structure design

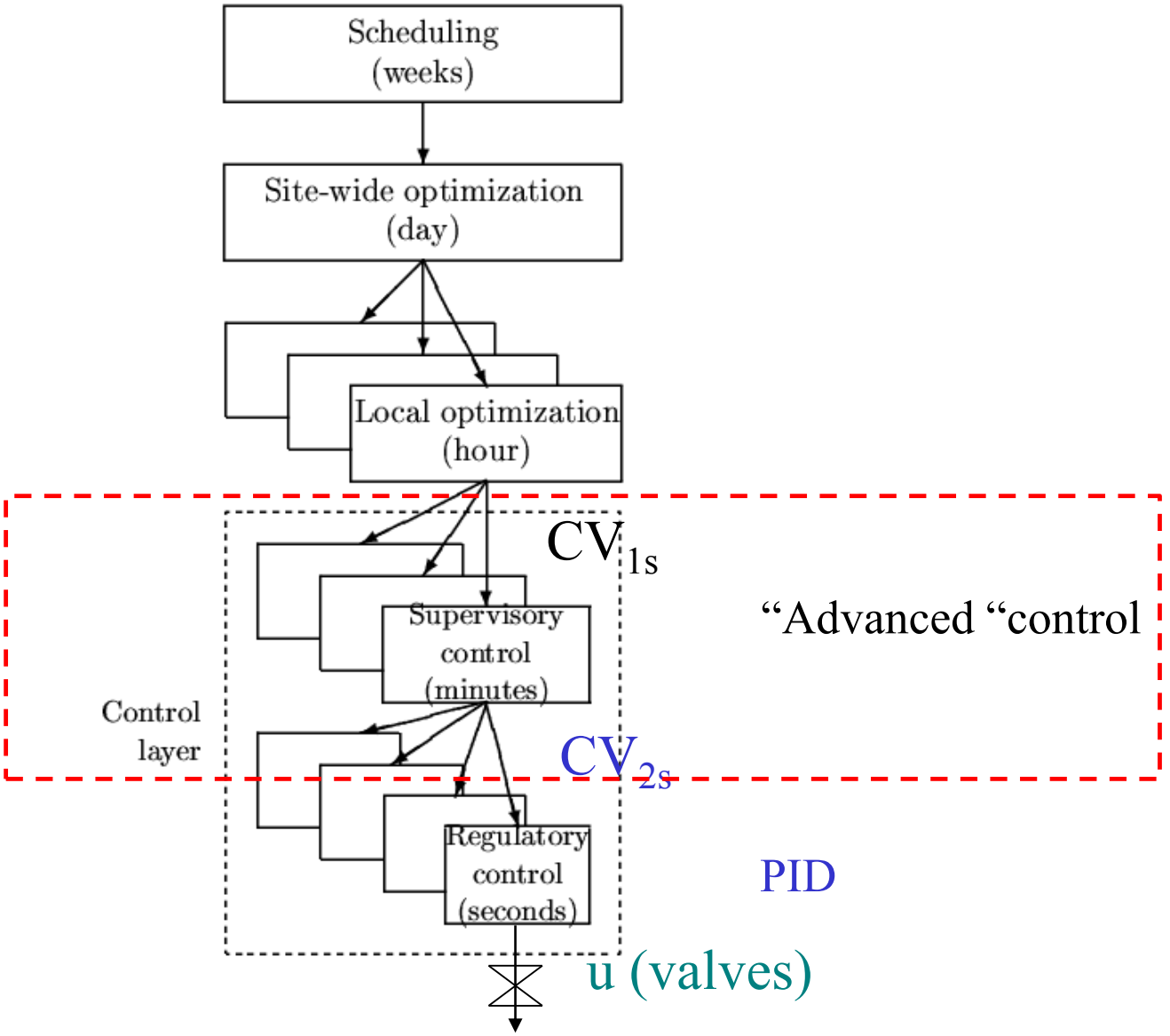
  I Top Down

  - Step S1: Define operational objective (cost) and constraints
  - Step S2: Identify degrees of freedom and optimize operation for disturbances
  - Step S3: Implementation of optimal operation
    - What to control ? (primary CV's)
    - Active constraints
    - Self-optimizing variables for unconstrained, c=Hy
  - Step S4: Where set the production rate? (Inventory control)

  II Bottom Up

  - Step S5: Regulatory control: What more to control (secondary CV's) ?
  - Step S6: Supervisory control
  - Step S7: Real-time optimization

Control layers



STEP S6. SUPERVISORY LAYER

Objectives of supervisory layer:

1. Perform "advanced" economic/coordination control tasks.

- Control primary variables CV1 at setpoint using as degrees of freedom (MV):
  - Setpoints to the regulatory layer (CV2s)
  - "unused" degrees of freedom (valves)
- Feedforward from disturbances
  - If helpful
- Make use of extra inputs
- Make use of extra measurements

2. Keep an eye on stabilizing layer

- Avoid saturation in stabilizing layer

3. Switch control structures (CV1) depending on operating region

- Active constraints
- self-optimizing variables

Implementation:

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

“Advanced control”
If single-loop feedback control (PID) alone is not good enough

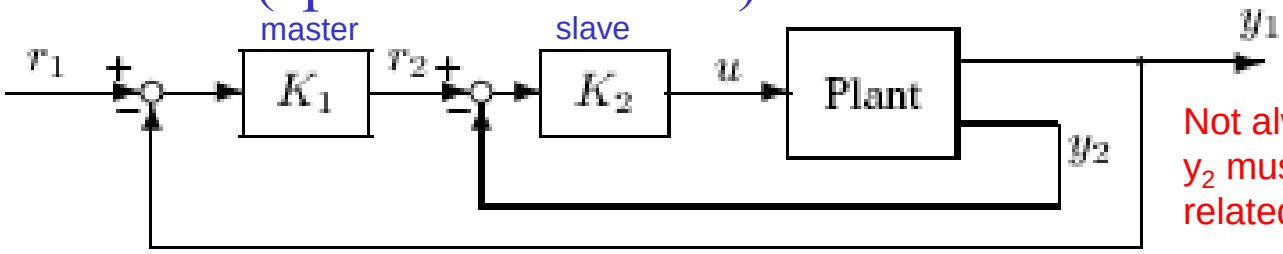
Design based on simple elements

1. Cascade control (measure and control internal variable, y2)
2. Feedforward control (measure disturbance, d)
    - Including ratio control
3. Multivariable control (decoupling)
    - For interactive process (where RGA is unfavorable)
4. Changes in active constraints
    - Selectors
    - Input resetting (valve position control)
    - Split range control

Cascade control

- Use cascade (with extra measurement* y_2) to improve control of y_1
- MV for one controller (master) is the setpoint to another (slave).

1. General case ("parallel cascade")



Not always helpful… y_2 must be closely related to y_1

(a) Extra measurements y_2 (conventional cascade control)

2. Special common case ("series cascade")

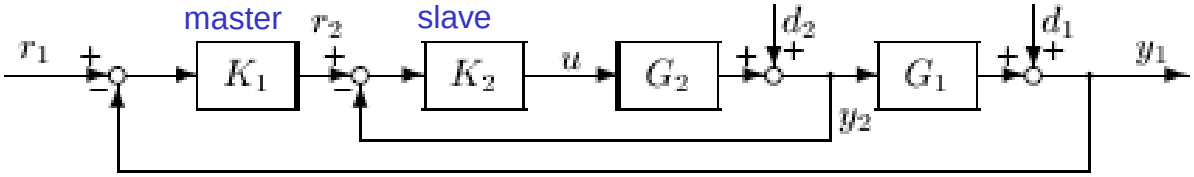


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

*Comment: Another approach that uses extra measurements to improve control is «**Full state feedback**».

Summary cascade control (for common case 2)

Use cascade (with extra measurement y_2) to improve control of y_1 when:

1. Effective delay for y_2 is small (compared to for y_1)
2. Important disturbances affect y_2
3. $G_2=G_vG_{p2}$ is nonlinear

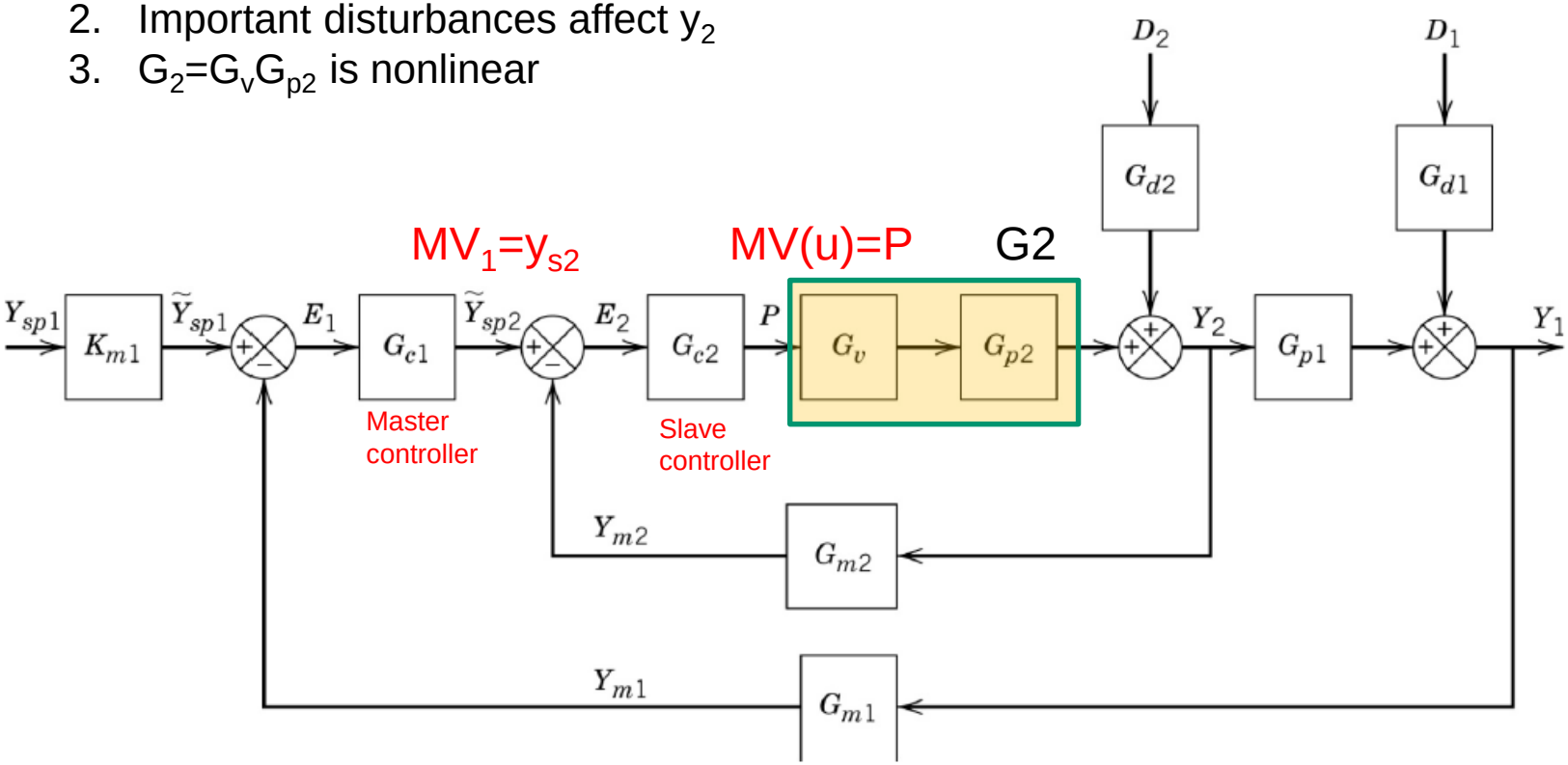


Figure 15.4

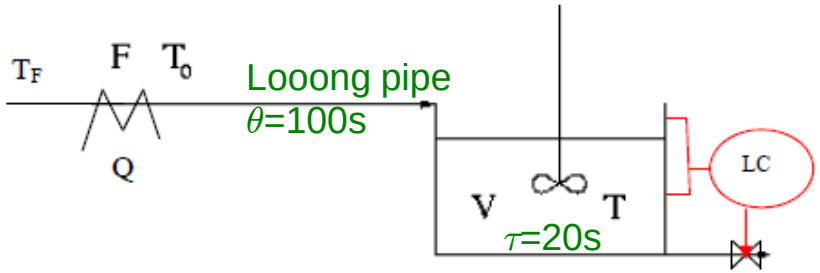
Common special case of “series cascade control” where $y_1 = g_{p1}\, y_2$.

Tuning for common "series cascade"

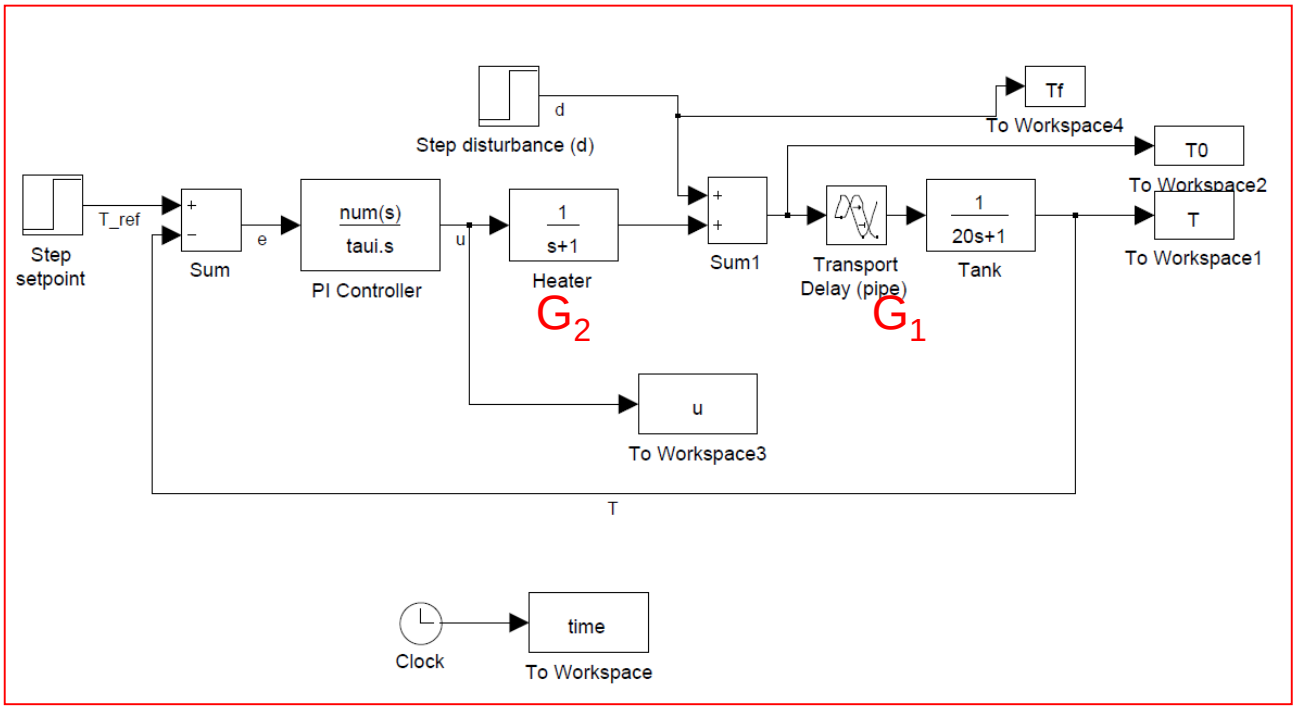
- First tune fast inner loop ("slave")
  - Design c2 based on model G2

- Then with slave closed, tune slower outer loop ("master"):
  - Design c1 based on model T2*G1
    - where T2 = G2 C2/(1+G2 C2) is closed-loop response from y2s to y2
    - With SIMC, $T_2 \approx e^{-\theta_2 s}/(\tau_{c2}\, s+1)$
      - Comment: Note that T2 has gain 1 provided C2 has integral action (independent of G2), which explains why cascade control counteracts nonlinearity in G2

Example: Similar to shower process



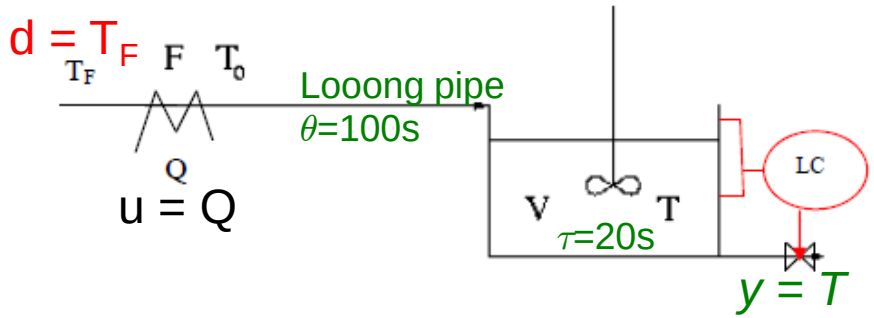
u = Q
y = T
d = T_F

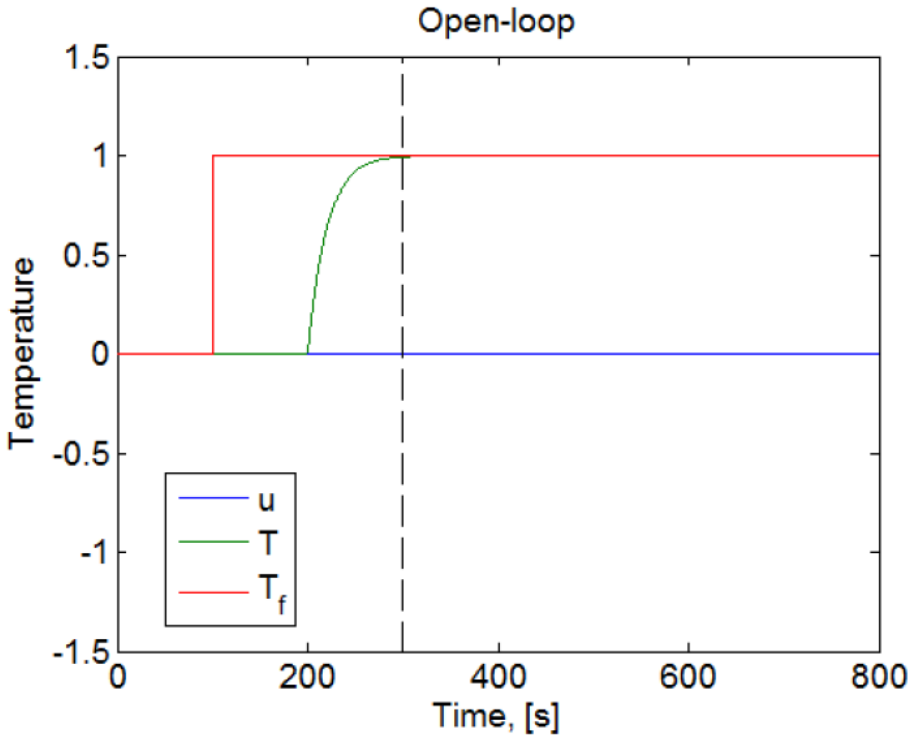


Simulink model: tunepid1_ex1

Note: level control not explicitly included in simulation (assume constant level)

Disturbance response with no control





u = Q
y = T
d = T_F

```
Kc=0; taui=9999; % no control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
```

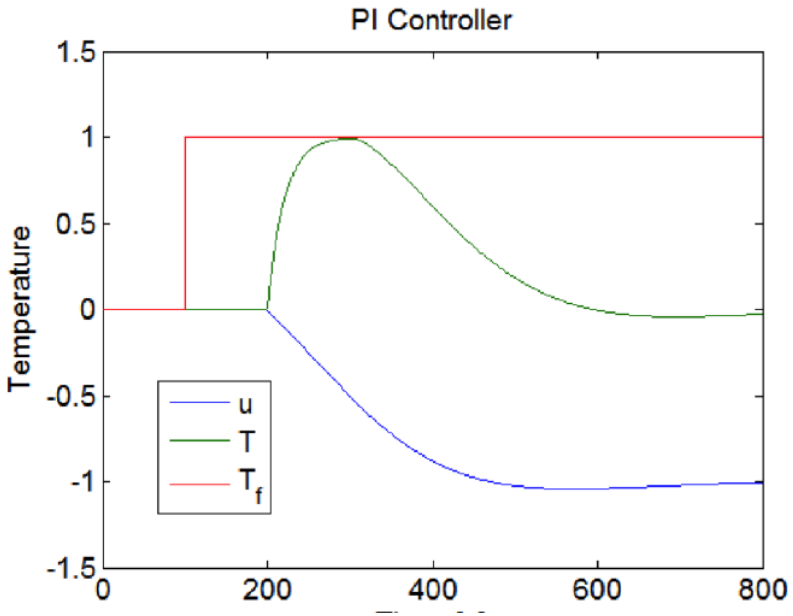
Without cascade: SIMC PI control

$G = G_1G_2 = \exp(-100s)/(20s+1)(s+1)$

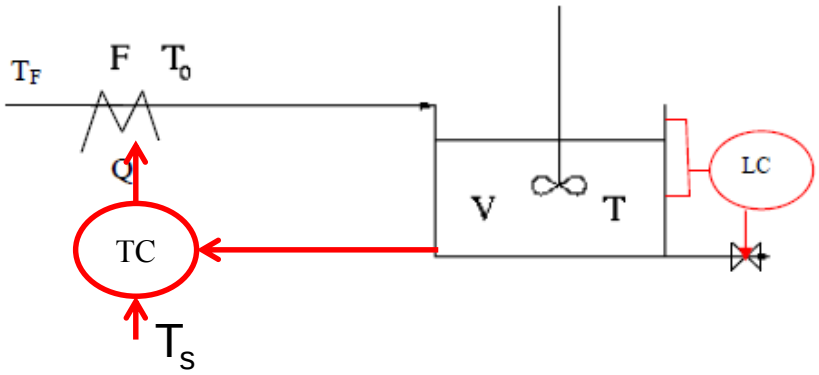
11) SIMC PI tuning rule with $\tau_c = \theta = 100$.

$$K_c = (1/k)\tau_1/(\tau_c + \theta) = 20/200 = 0.1; \tau_I = \min(\tau_1, 4(\tau_c + \theta) = 20$$

u = Q
y = T
d = T_F

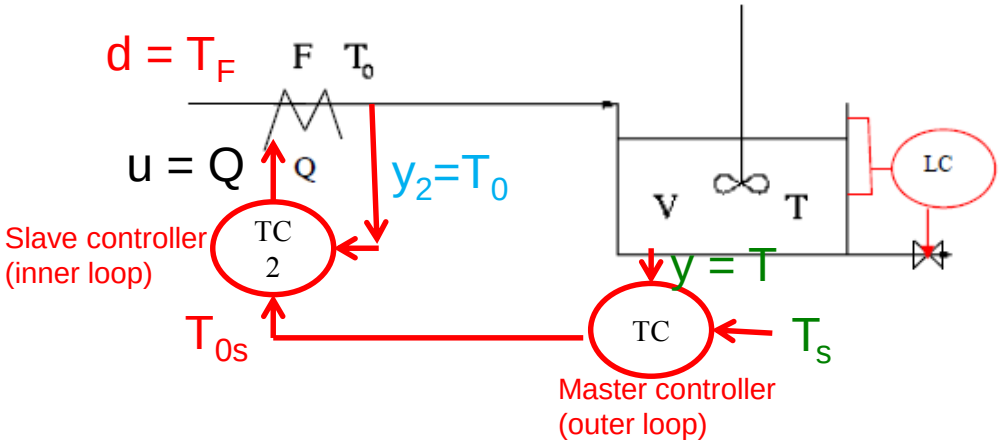


No offset

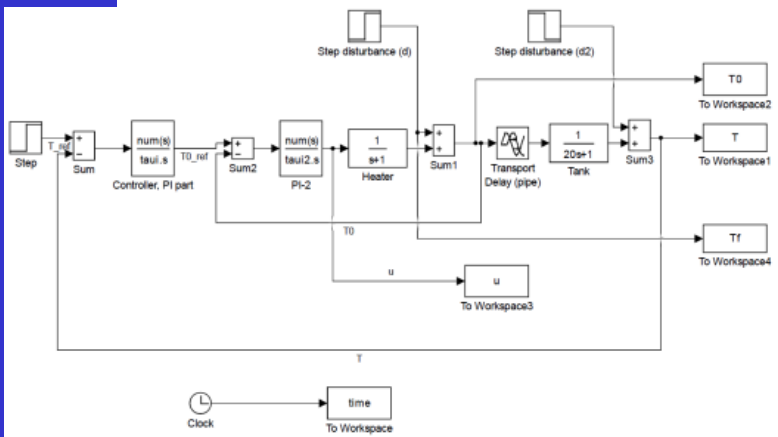


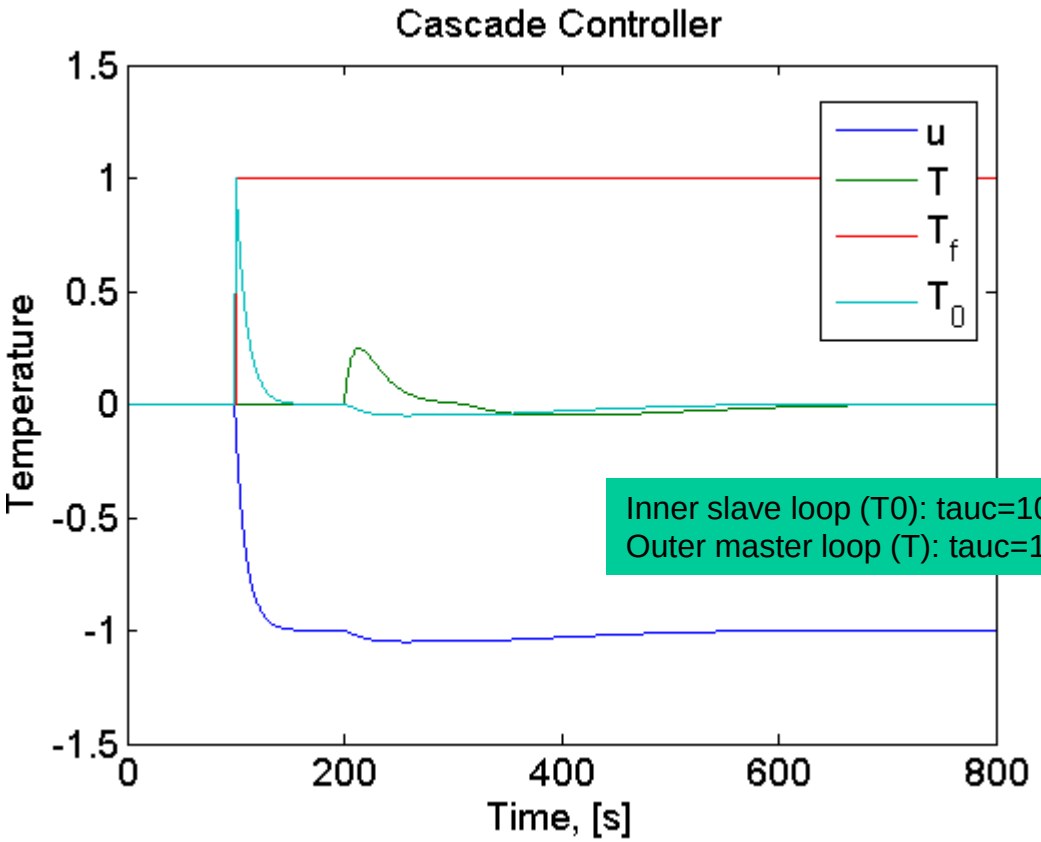

```
Kc=0.1; taui=20; % SIMC PI-control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
```

Measure also T_0: Cascade control is much better



G_1= exp(-100s)/(20s+1)
G_2=1/(s+1)





Inner slave loop (T0): tauc=10
Outer master loop (T): tauc=105

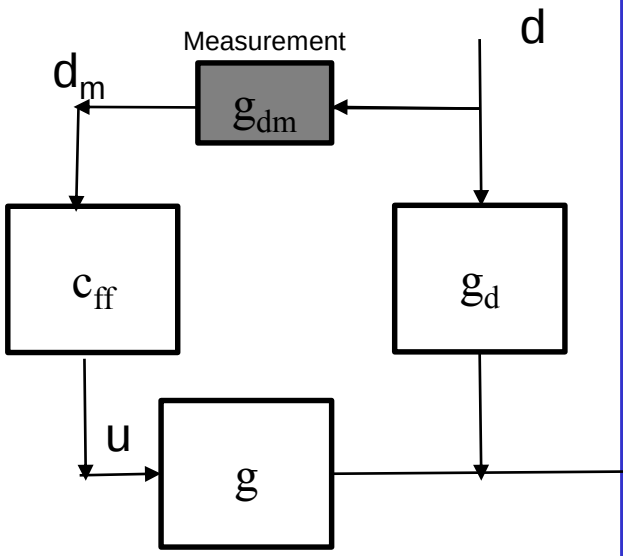
```
Kc2=0.1;taui2=1; % inner loop with tauc2=10
Kc=0.119; taui=25; % outer loop with tauc=105
sim('tunepid1_ex1_cascade') %start simulation
plot(time,u,time,T,time,Tf,time,T0), axis([0 800 -1.5 1.5])
```

Question: Will setpoint tracking for $y_1 = T_2$ be improved with cascade (in this case)?

- No, since there was essentially no dynamics in G2=1/(s+1), it is actually slightly worse (tauc increased from 100 to 105).

2. Feedforward control

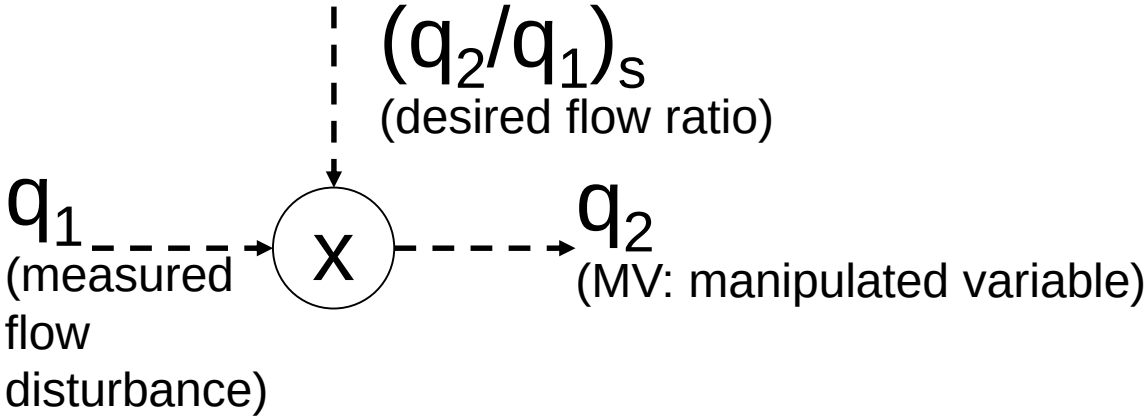
- See SIMC-slides



Ratio control (most common case of feedforward)

Example: Process with two feeds q_1(d) and q_2 (u), where ratio should be constant.

Use multiplication block (x):



"Measure disturbance ($d=q_1$) and adjust input ($u=q_2$) such that ratio is at given value $(q_2/q_1)_s$"

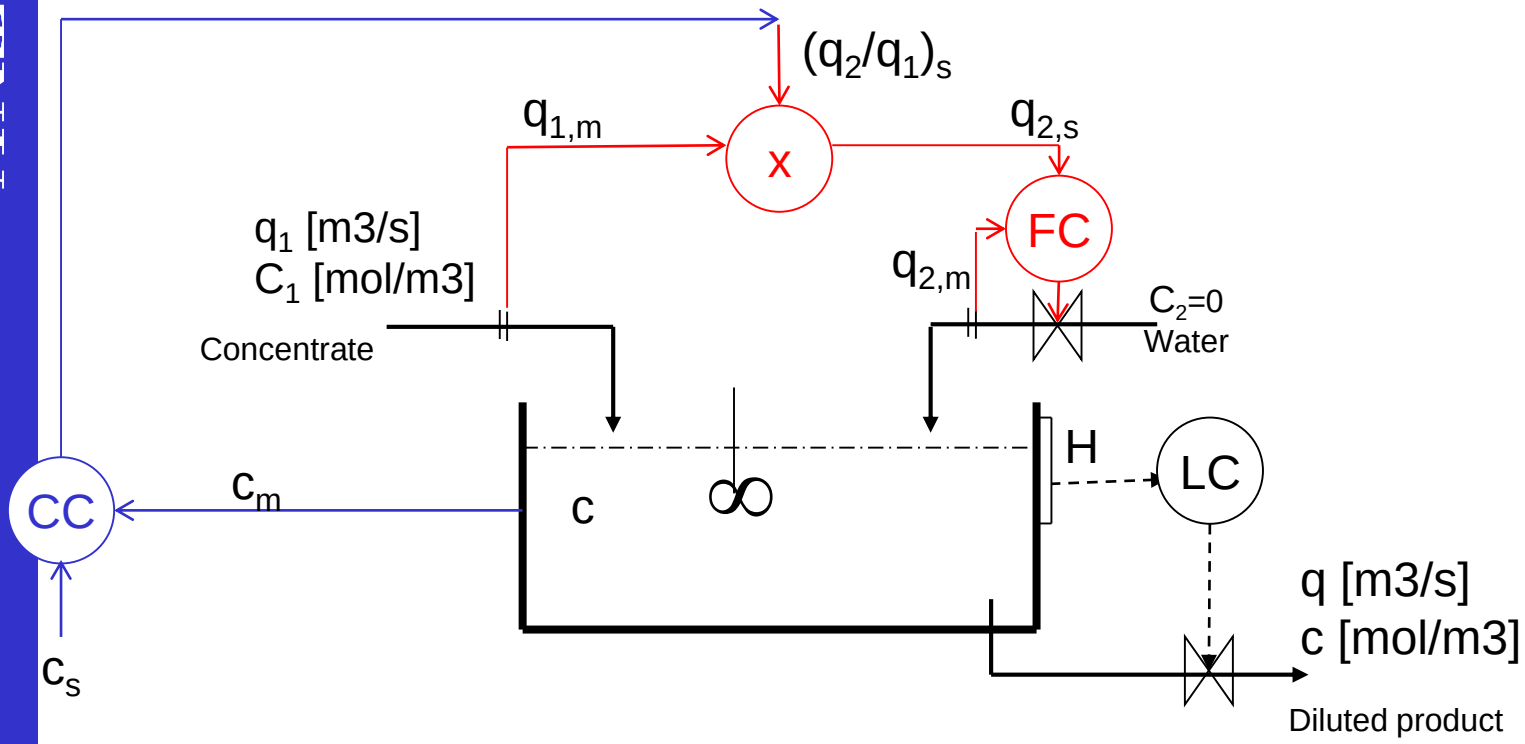
Usually: Combine ratio (feedforward) with feedback

- Adjust $(q1/q2)_s$ based on feedback from process, for example, composition controller.
  - This may be viewed as a special case of cascade control

  – ***Example cake baking***: *Use recipe (ratio control = feedforward), but adjust ratio if result is not as desired (feedback)*
  – ***Example evaporator:*** *Fix ratio q_H/q_F (and use feedback from T to fine tune ratio)*

EXAMPLE: MIXING PROCESS

RATIO CONTROL with outer cascade (to adjust ratio setpoint)



Potential problem for **outer** feedback loop (composition controller):

- Gain from MV = $(q2/q1)_s$ to CV=c will vary because of multiplication with q1,m.
- So if large variations in q1 are expected: Outer loop needs robust tunings (to get high gain margin; use large tauc in SIMC)

Ratio control

- It is simple
- Book has some strange suggestions, for example, Figure 14.5

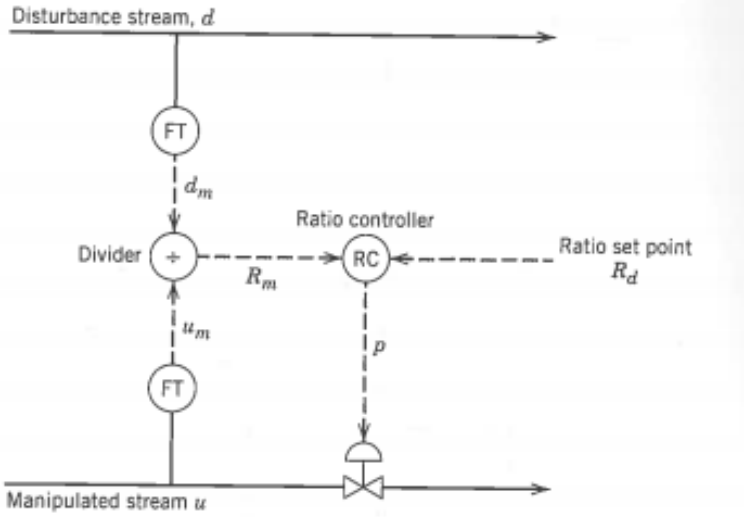


Figure 14.5 Ratio control, Method I.

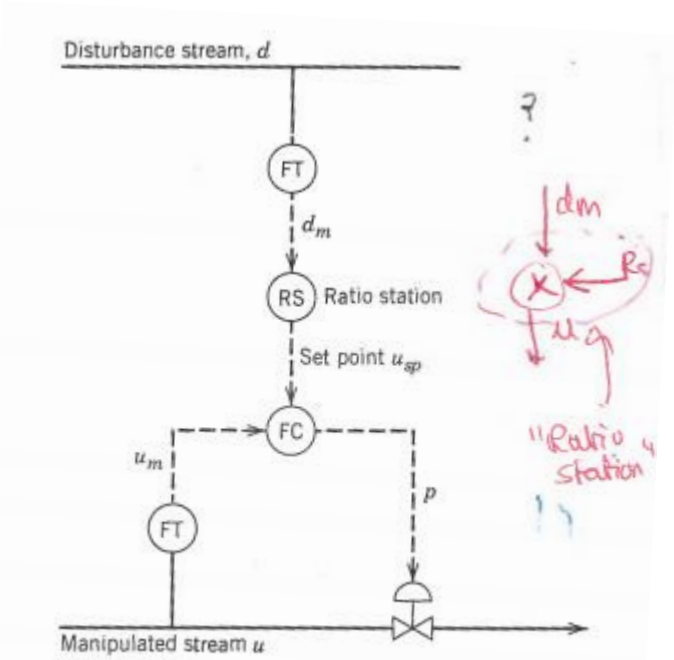


Figure 14.6 Ratio control, Method II.

Bad solution

Ok if implemented as shown in red

3. Multivariable control: If interactions cause poor performance

Possible solutions:

I. Adding fast loop to break interactions (cascade control)
II. Decoupling
III. MPC

Examples I: Breaking interactions with cascade

Example 1. Control of level and pressure in separator

- MVs: Valve positions for liquid and gas out
- Highly interactive
- Interactions an be avoided with cascade! How?

Example 2. Control of compositions in distillation column

- MVs: Reflux and heat input (boilup)
- Time delay on composition measurements
- Highly interactive
- Can to some extent be avoided with cascade (inner temperature loop)

Multivariable control

1. Single-loop control (decentralized)
2. Decoupling (similar to feedforward)
3. Model predictive control (MPC)

Single-loop control = Decentralized control

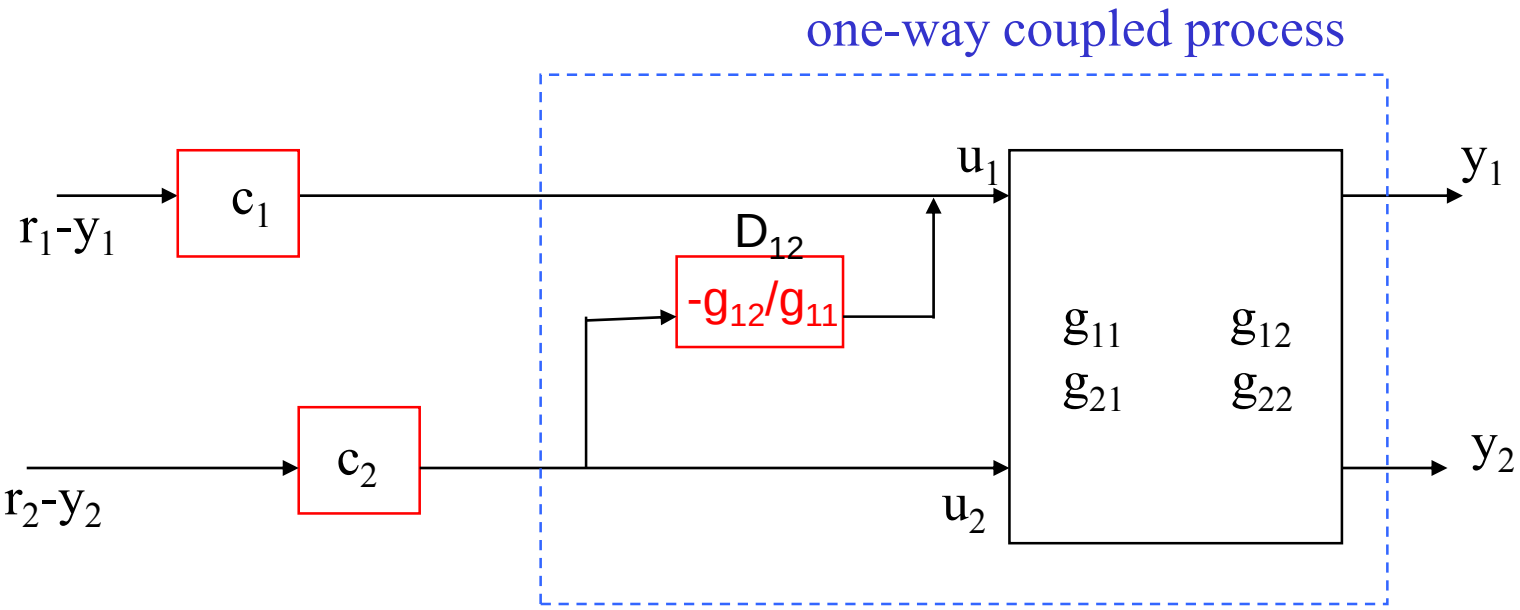
Use for: Noninteracting process and no change in active constraints

+ Tuning may be done on-line
+ No or minimal model requirements
+ Easy to fix and change
- Need to determine pairing
- Performance loss compared to multivariable control
- Complicated logic required for reconfiguration when active constraints move

Decentralized control tuning

- Independent design
  - Use when small interactions (Dynamic RGA close to I)
- Sequential design (similar to cascade)
  - Start with fast loop
  - NOTE: If close on negative steady-state RGA, system will go unstable of fast (inner) loop saturates
  - Sequential vs. independent design
    - + Generally better performance, but
    - - outer loop gets slow, and
    - - loops depend on each other

One-way Decoupling (improved control of y_1)



DERIVATION

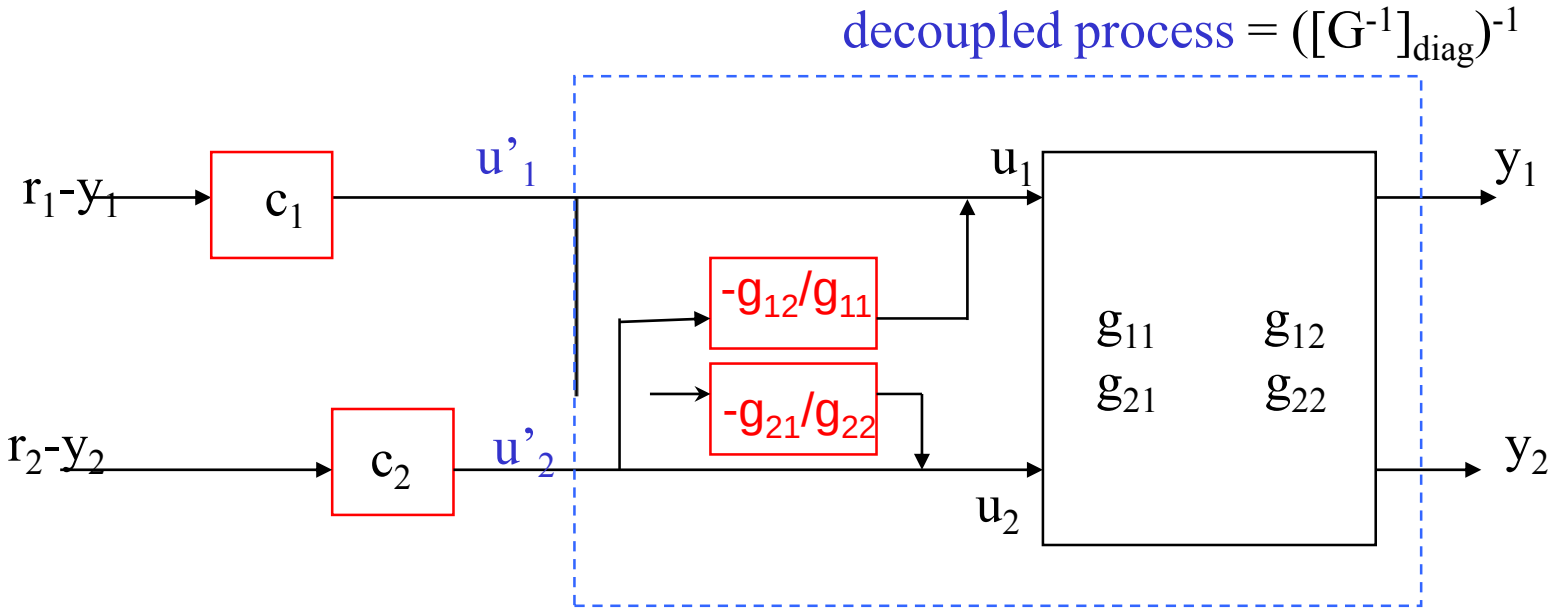
Process: $y1 = g11\ u1 + g12\ u2$ (1)

$y2 = g21\ u1 + g22\ u2$ (2)

Consider u2 as disturbance for control of y1.

Think «feedforward»: Adjust u1 to make y1=0. (1) gives $u1 = -(g12/g11)\ u2$

Two-way Decoupling: Standard implementation (Seborg)

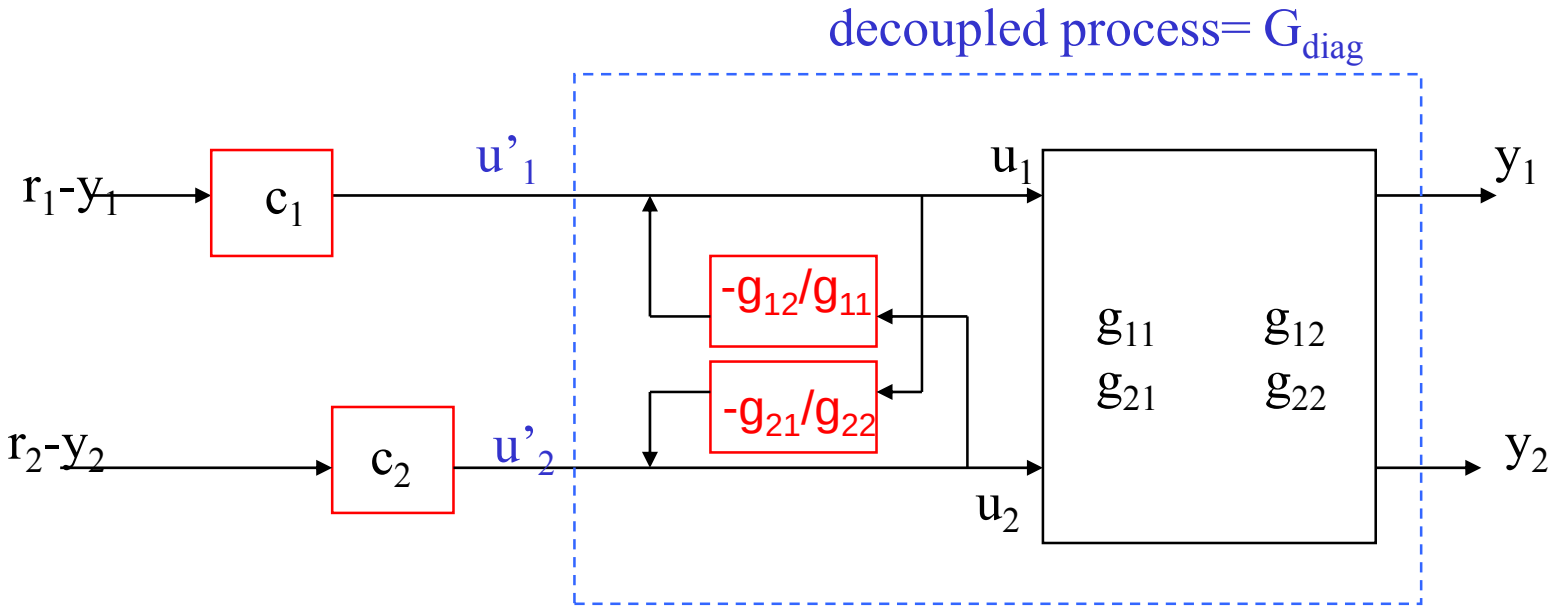


… but note that diagonal elements of decoupled process are different from G
Problem for tuning!

Process: y1 = g11 u1 + g12 u2
Decoupled process: y1 = (g11-g21*g12/g22) u1' + 0*u2'
Similar for y2.

Sigurd recommends this alternative!

Two-way Decoupling: «Inverted» implementation (Shinskey)



Advantages: (1) Decoupled process has same diagonal elements as G. Easy
(2) Handles input saturation! (if u1 and u2 are actual inputs)

Proof (1): $y_1 = g_{11} u_1 + g_{12} u_2$, where $u_1 = u_1' - (g_{12}/g_{11})u_2$.
Gives : $y_1 = g_{11} u'_1 + 0^* u_2'$
Similar: $y_2 = 0^* u_1' + g_{22} u_2'$

Pairing and decoupling

- To get ideal decoupling, offdiagonal elements should have smaller effective delay than the diagonal elements
- Thus, we should pair on elements with small effective delay ("pair close rule")
- Pairing on negative steady state RGA elements is not a problem if we use decoupling
  - Because negative RGA-elements are caused by interactions, which is what we are cancelling with decoupling

Advanced multivariable control with explicit constraint handling = MPC

Use for: Interacting process and changes in active constraints

+ Easy handling of feedforward control
+ Easy handling of changing constraints
  - no need for logic
  - smooth transition
- Requires multivariable dynamic model
- Tuning may be difficult
- Less transparent
- "Everything goes down at the same time"

Multivariable control: MPC versus decoupling

- Both MPC and decoupling require a multivariable process model
- MPC is usually preferred instead of decoupling because it can also handle feedforward control, nonsquare processes (cascade, input resetting) etc.

MPC = Model predictive control

4. Changing active contraints

Procedure for maintaining optimal operation when changing between active constraint regions

Step 1: Define all the constraints

Step 2: Identify relevant active constraint combinations and switches

Step 3: Propose a control structure for the nominal operating point.

Step 4: Propose switching schemes

Step 5: Design controllers for all cases (active constraint combinations)

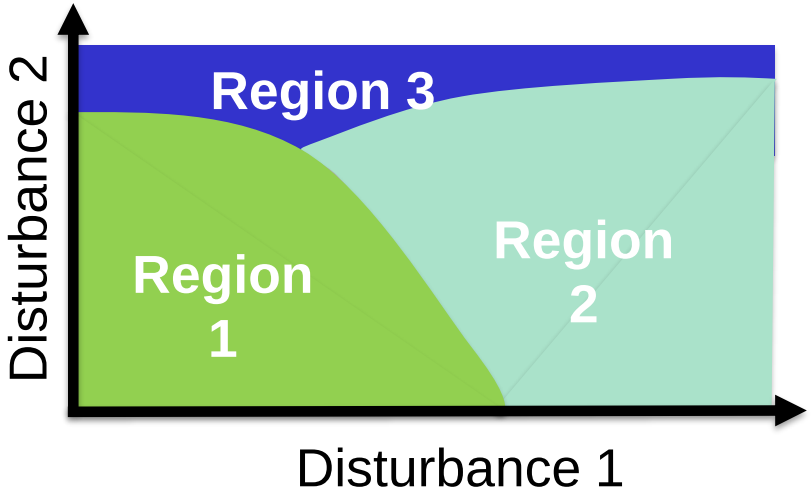
Step 1: Define all the constraints

- Identify the type of constraints
  - **Input constraints:**
    - Manipulated variables (MV)
      - e.g. valve fully open or fully closed.
  - **Output constraints:**
    - Controlled variables (CV)
      - e.g. maximum temperature or pressure.

- Unconstrained optimum:
  - Identify Self-optimizing variables.

Step 2: Identify relevant active constraint combinations and switches

- **Active constraints:**
  - variables that should optimally be kept at their limiting value.
- **Active constraint region:**
  - region in the disturbance space defined by which constraints are active within it.



Note: Not necessary to generate this diagram!

Step 2: Identify relevant active constraint combinations and switches

- **Maximum number of active constraint combinations (regions):**

  $n_r^{max} = 2^{nc}$

  nc: number of constraints

- **In practice, fewer regions**
  - Certain constraints are always active (reduces effective n_c)
  - Some combinations are infeasible:
    - Only n_u can be active at a given time
      - n_u = number of MVs (inputs)
    - Certain constraints combinations are not possible
      - *Cannot have both max* and *min* on the same variable (e.g. flow)
  - Certain regions are not reached by the assumed disturbance set
- **Furthemore, not all switches occur or are feasible**
  - Only neighboring switches need to be considered, that is, only one constraint is changing

Tool: Organize constraints in priority list

P1: MV inequality constraints (can never be violated)

P2: CV inequality constraints (may sometimes be given up)

P3: MV or CV equality constraints (may often be given up)

P4: Desired throughput (give up when reach bottleneck)

P5: Self-optimizing variables (can be given up)

A. Reyes-Lúa, C. Zotică, S. Skogestad, 2018. *Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control*. In: 10th ADCHEM. IFAC, Shenyang, China.

Step 3: Design control structure for normal operation

Try to follow **Input saturation pairing rule:**

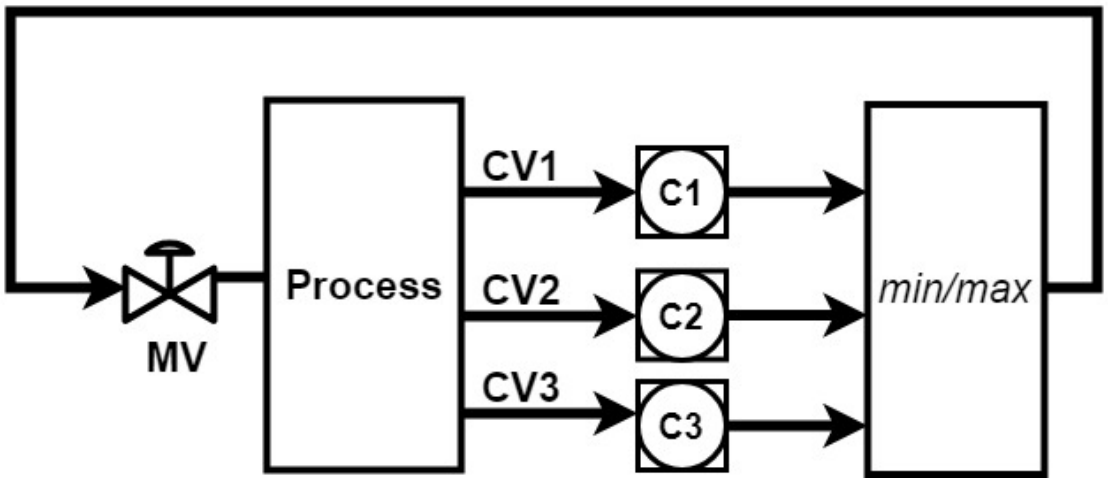
- Pair a manipulated variable (MV) that is likely to saturate
- With a low-priority controlled variable (CV) (can be given up)

Step 4. Design control structures for switching of active constraints

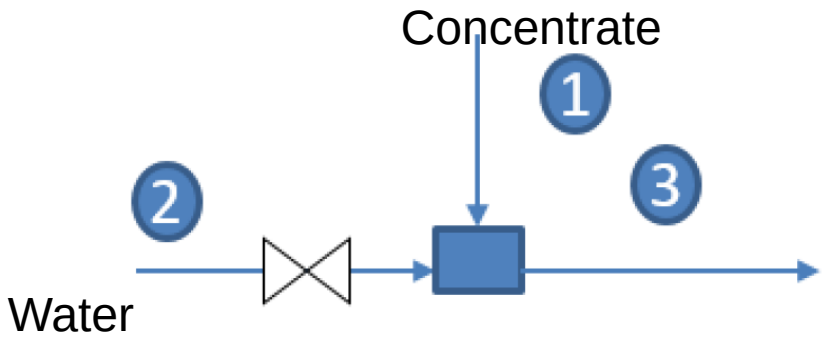
- Three main cases:
  1. Output to output (CV-CV) switching (SIMO)
  2. Input to input (MV-MV) switching (MISO)
  3. Input to output (MV-CV) switching

1. CV-CV switching (SIMO)

- MV is used to control CV1
- Problem: CV2 (so far uncontrolled) reaches constraint
- Solution: Use MV to control CV2 instead («give up» controlling CV1).
  - One MV used for several CVs (SIMO).
  - Design one controller for each CV.
  - Switching: Use *min/max* selector on controller output
  - Self-optimizing CV: Only track neighboring regions.



Problem 4. Mixing tank with changing control objective (40%)



CV1 = x_{S3} (sugar)
CV2 = x_{E3} (impurity)

You are mixing two streams. Stream 1 contains water (W), sugar (S) and some preservative (E). Your task is to mix the feed (stream 1) with pure water (stream 2) to get a product (stream 3) that satisfies:

Desired sugar content (want to keep product close to this value): $x_{s3} = 0.1$
Maximum E in product (required at all times): $x_{E3} \leq 0.001$

The water flow (F2) is limited to F2max

(a) Design a control system for the nominal case with little impurity E (X_{E3} small)

(b) Design a control system that can handle also the case when CV2 goes above the max. value of 0.001

(a)

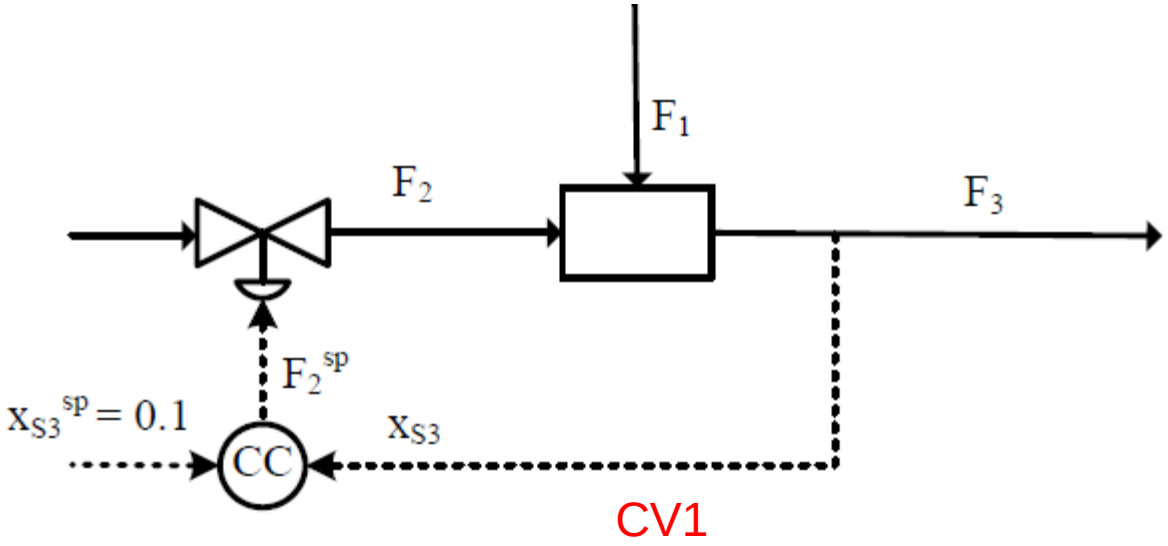


Figure 8: Control structure for the nominal point

This follows «input pairing rule»: Pair a MV that is likely to saturate (F2) with a CV that can be given up (x_{S3})

Important: Since we follow «input pairing rule» no special action is needed when F2 saturates (fully open valkve) except that controller CC should have anti windup

(b)

CV-CV switching using selector

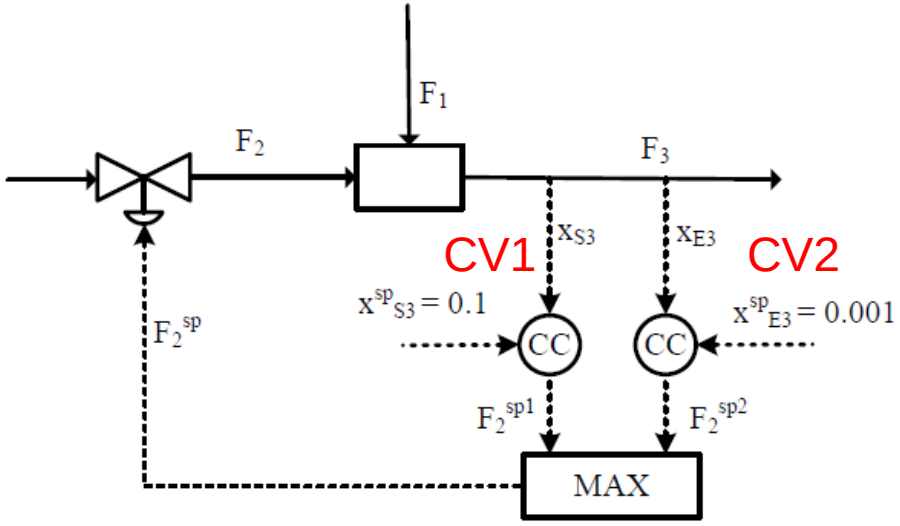


Figure 9: Control structure that handles both the nominal and the extreme cases

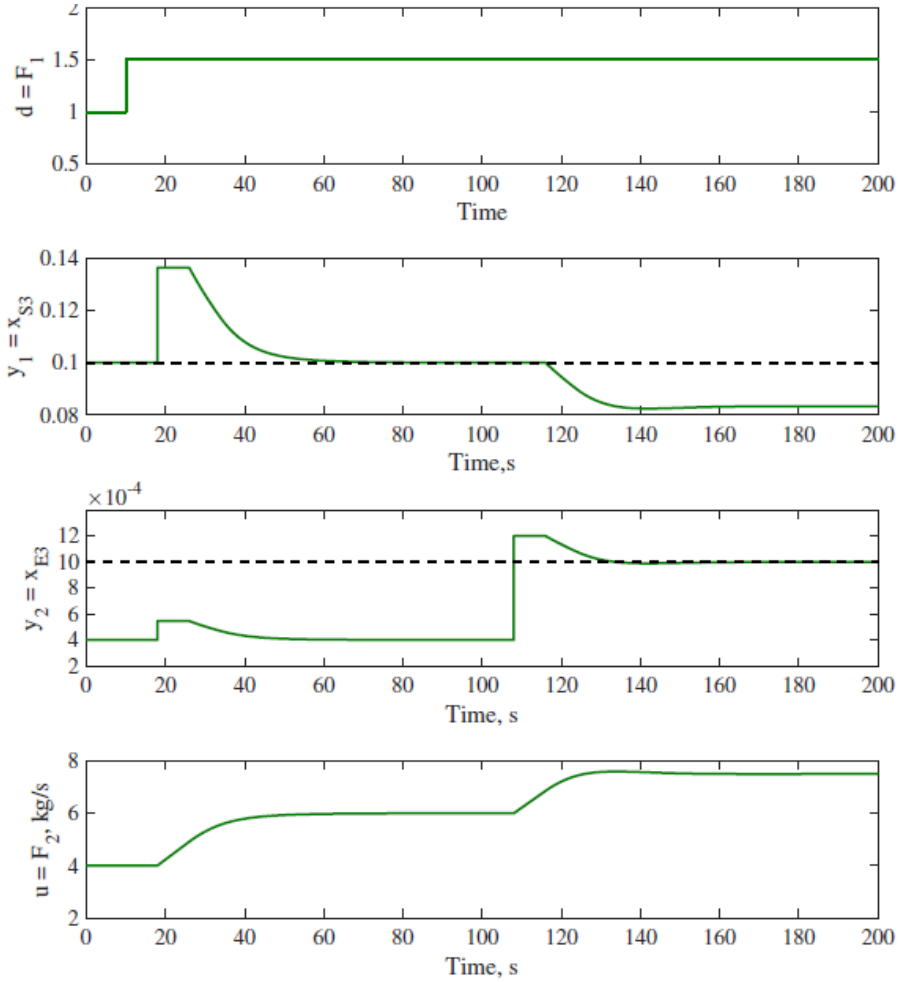
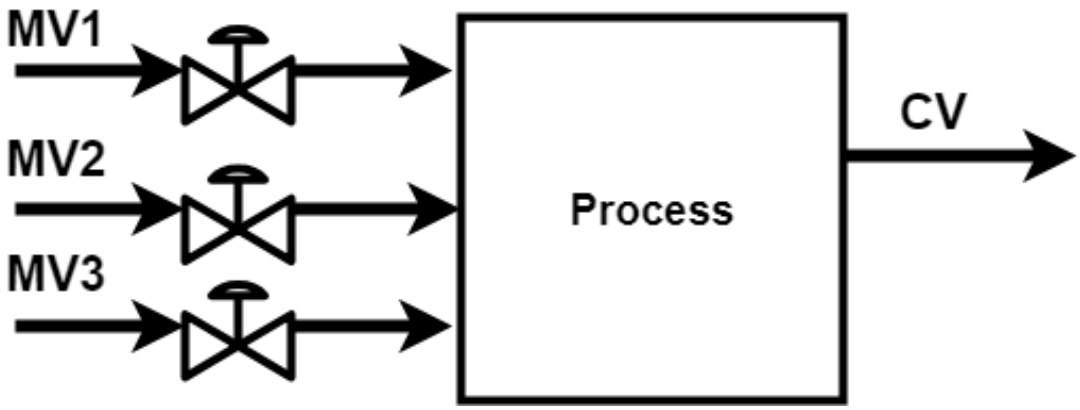


Figure 10: Simulation results for a step disturbance $F_1 = 1.5\,kg/s$ at $t = 10\,s$ and $x_{E1} = 0.006$ at $t = 100\,s$ (extreme case). The black dotted lines show the concentration specification for x_{S3} and x_{E3} respectively. In the normal case, the controller is controlling $y_1 = x_{S3}$ at $y_{1s} = 0.1$, while in the extreme case, the controller is controlling $y_2 = x_{E3}$ at $y_{2s} = 0.001$.

2. MV-MV switching (MISO)

- Problem: We are using MV1 to control CV, but MV1 saturates
- Solution: Let MV2 take over (re-pairing of MVs)

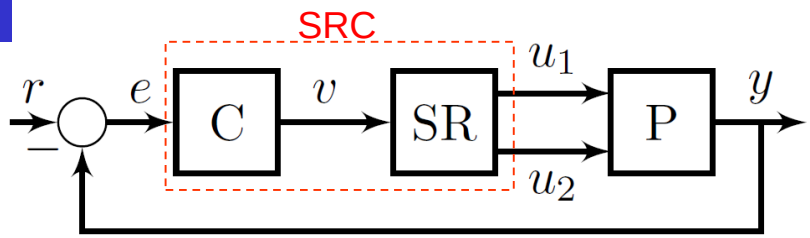
- Alternative MV-MV switching strategies:
  1. **Split-range control (SRC)**
  2. **Several controllers for same CV, but with different set points CV_s**
  3. **Input (valve) position control**



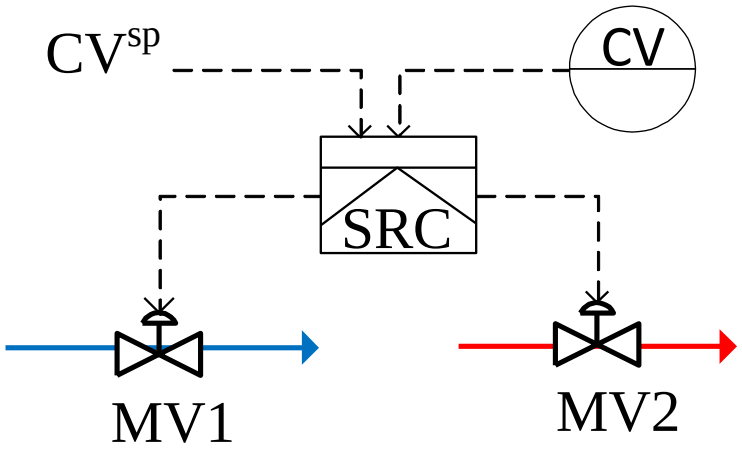
2. MV-MV switching

Alt. 1. Split-Range Control

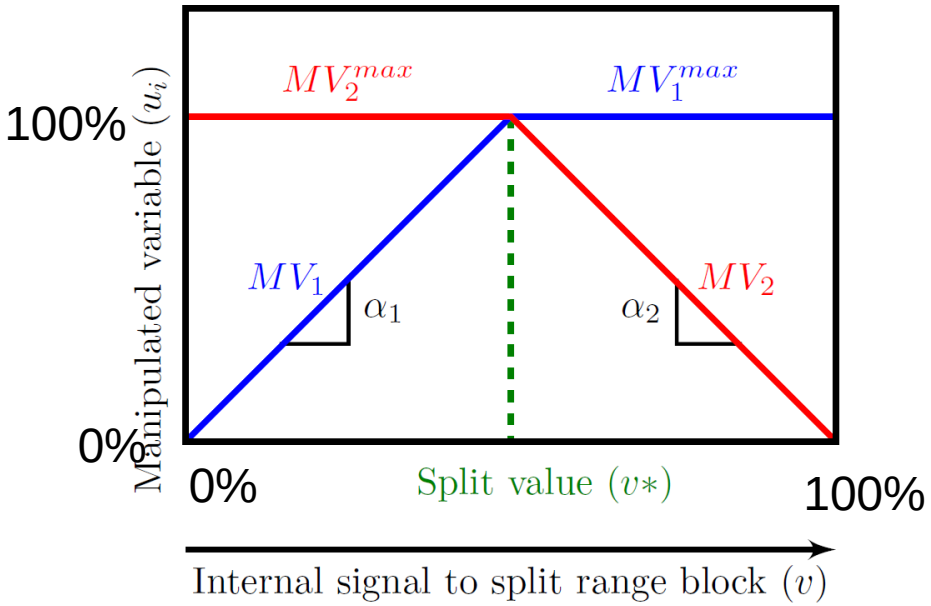
Block diagram:



Flowsheet:



Inside split range /SR) block:

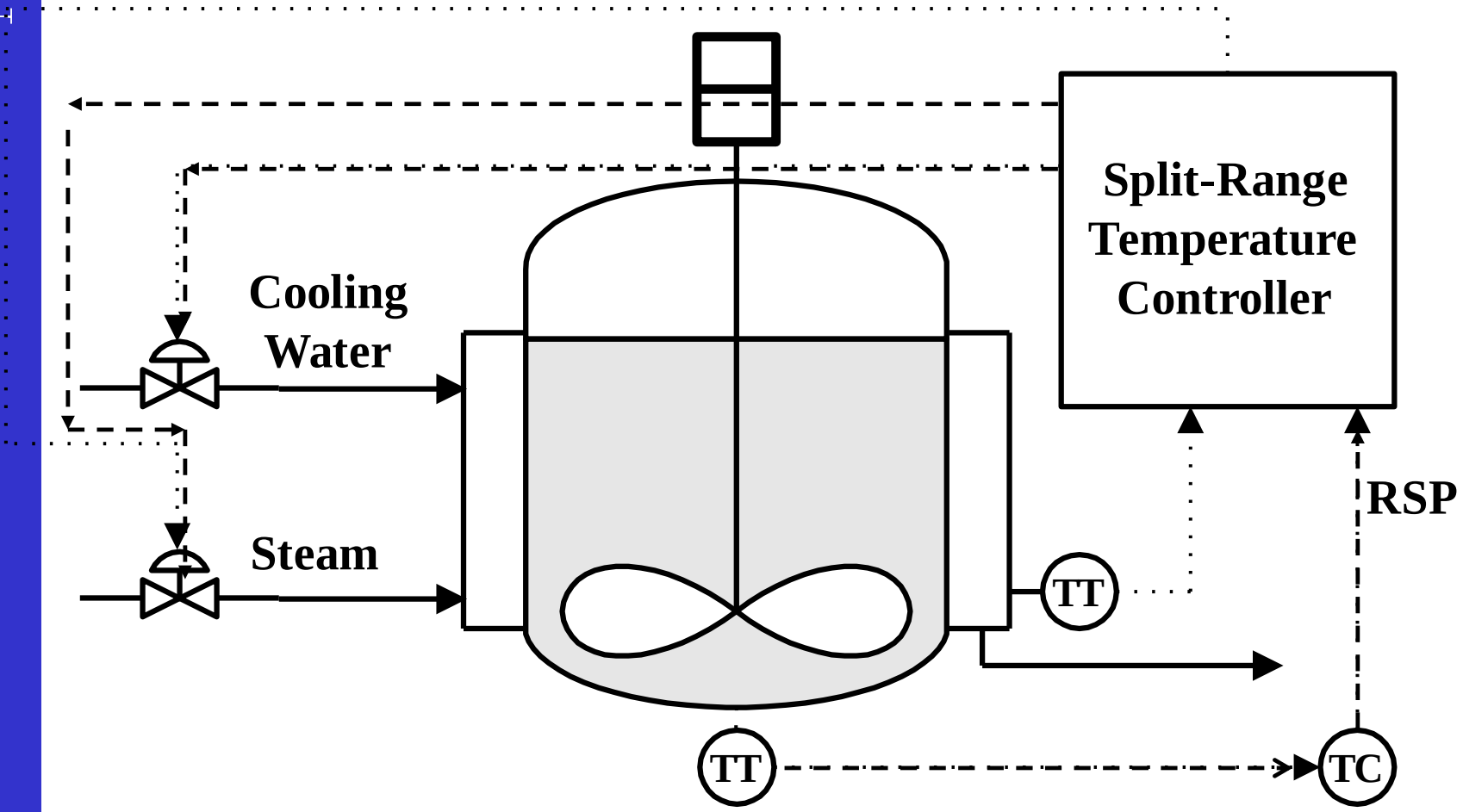


Note: One controller C is used for two MVs but v* is a design parameter which can be used to correct for different process gains

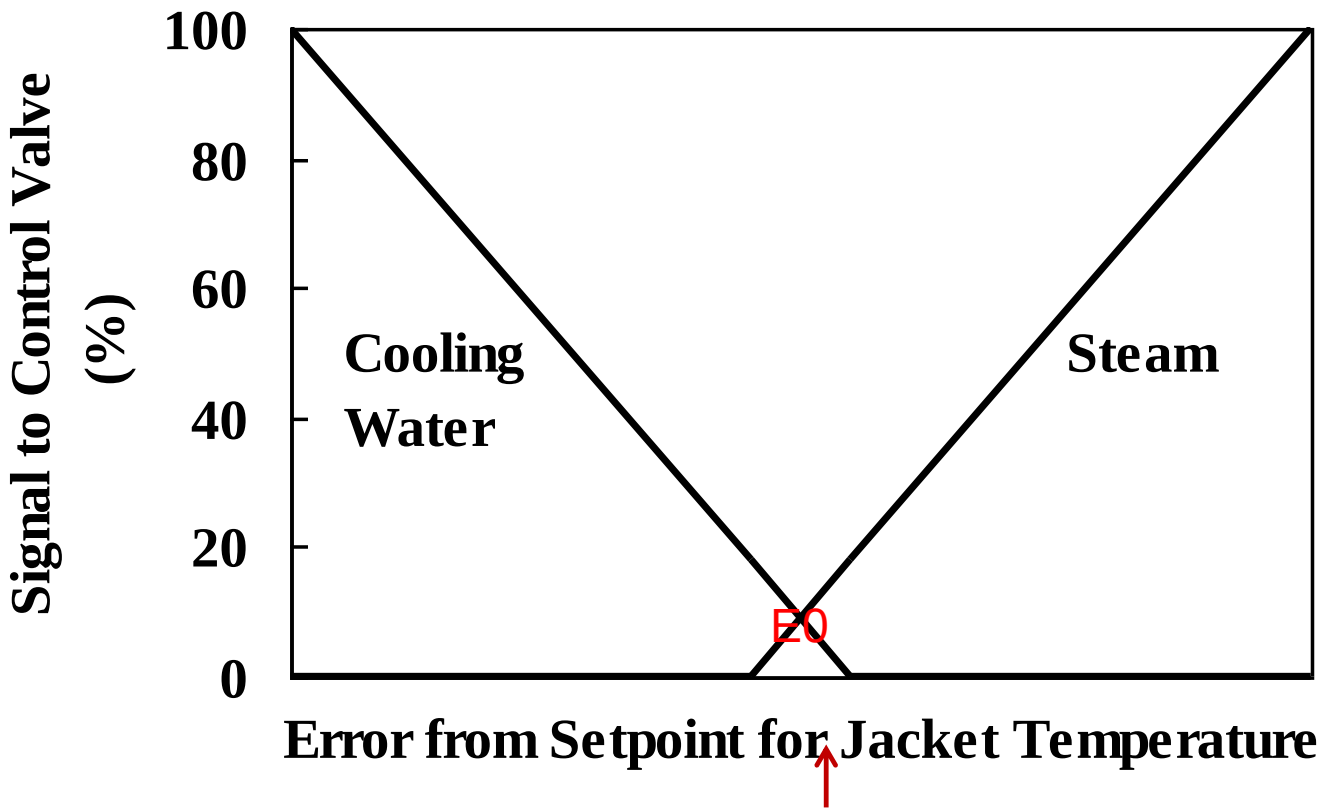
Example: MV-MV switching using Split Range Temperature Control

(Two MVs needed to cover whole range, one CV)

Split range control is used when we need to inputs to cover the whole output range (at steady state), for example, we need both heating and cooling in a house to control temperature.
The range is split so that only one input is active for control at a time



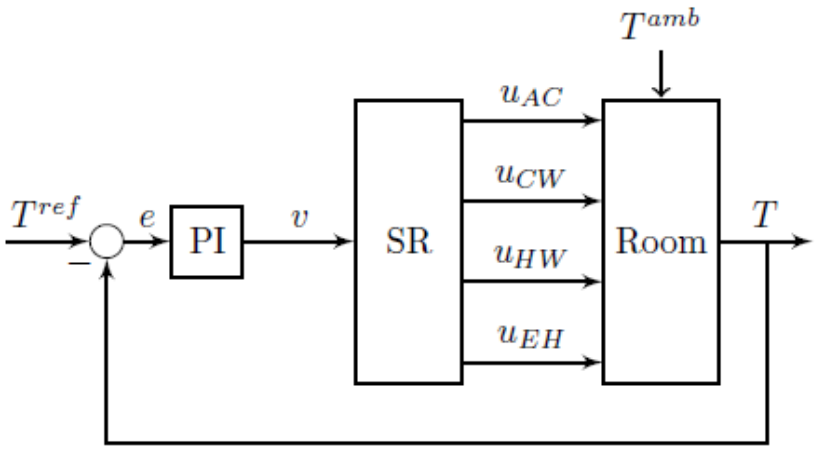
Split Range Temperature Control

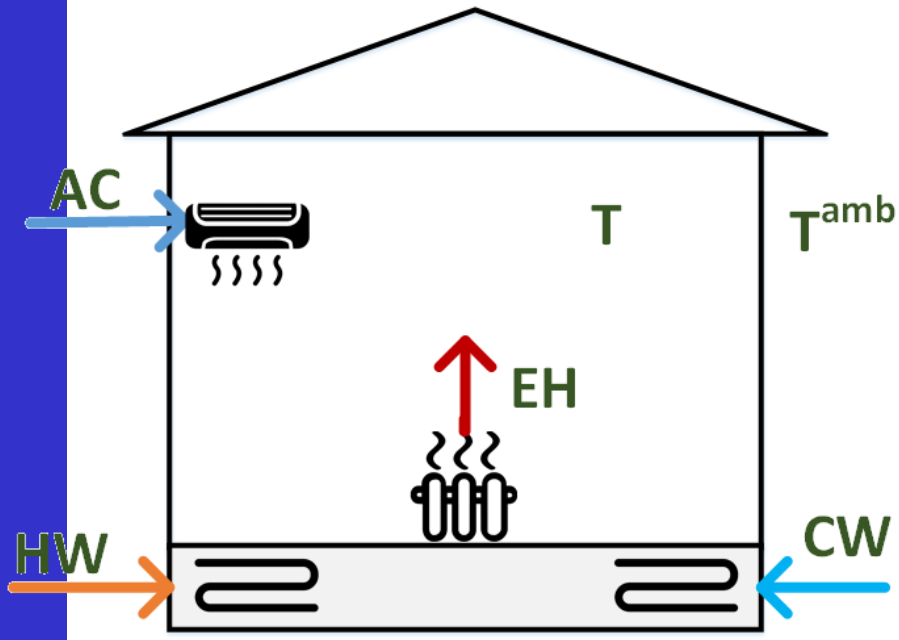


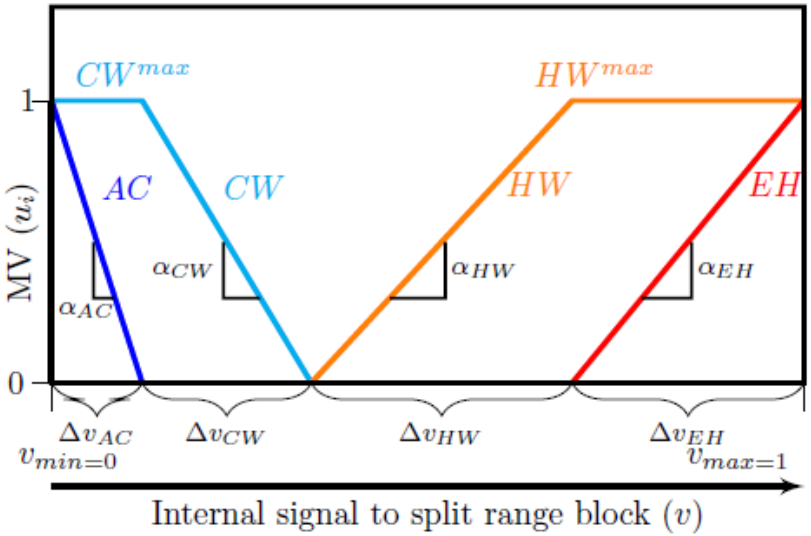
Note: may adjust the location er E0 to make process gains equal

A more complicated example using SRC

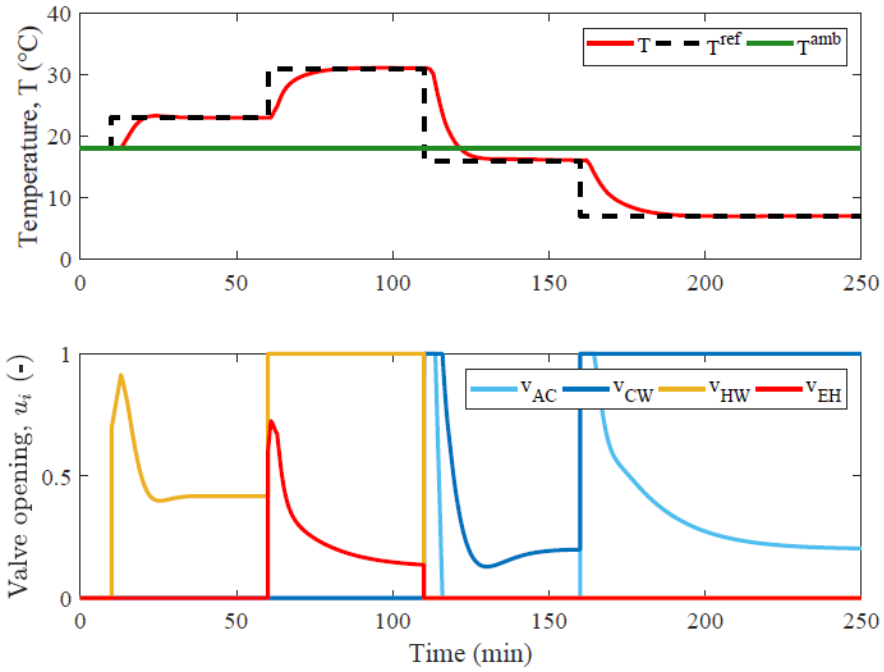
- Room temperature control
  - **4 MVs:** AC, CW, HW, EH
  - **1 CV:** room temperature (T)
    - $T_s = 21°C$

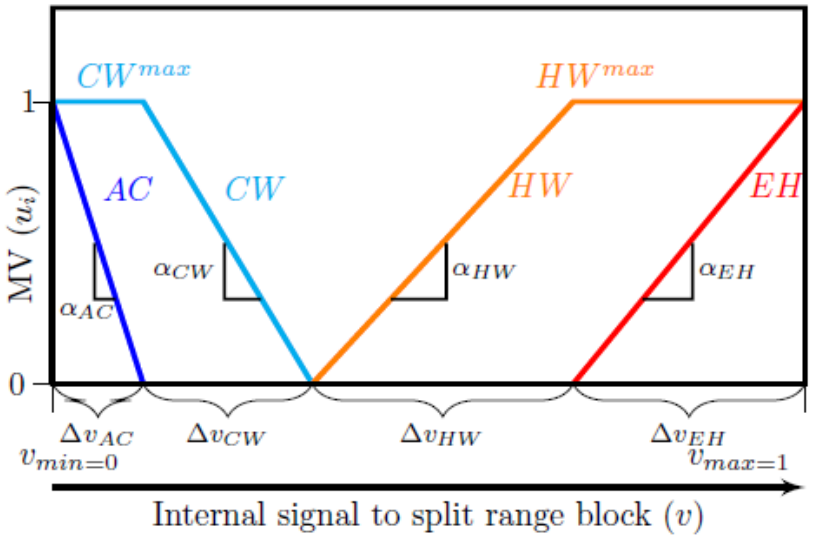






Simulation

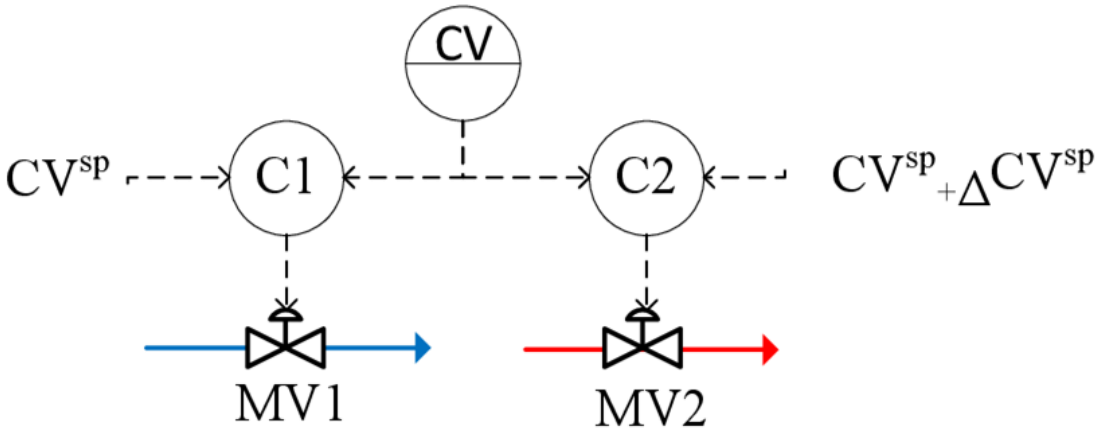

Temperature, T (°C)
T
T^ref
T^amb
Valve opening, u_i (-)
v_AC
v_CW
v_HW
v_EH
Time (min)


CW^{max}
HW^{max}
AC
CW
HW
EH
α_{AC}
α_{CW}
α_{HW}
α_{EH}
MV (u_i)
Δv_{AC}
Δv_{CW}
Δv_{HW}
Δv_{EH}
$v_{min=0}$
$v_{max=1}$
Internal signal to split range block (v)

MV-MV switching:

Alt. 2. Several controllers with different setpoints

- Two or more controllers with same CV
- Different setpoints ⇒ Only one controller active at a time.



Example

CV= room temperature

- MV1 = cooling
  CV^{sp} = 22°C
- MV2 = heating
  CV^{sp} = 20 + 2 = 20°C

Between 20°C and 22°C:
Temperature drifts with
Heating off (saturated) and
cooling off (saturated)

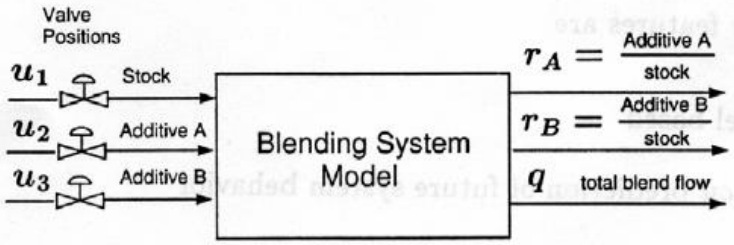
MV-MV switching
Alt. 3. Input (valve) position control (VPC)

- Keep the original loop (MV1-CV)
- Use MV2 to avoid saturation of MV1 (VPC)

- Advantage: No change in control of CV
- Disadvantages: Backoff in MV1 and control of MV1 can be slow

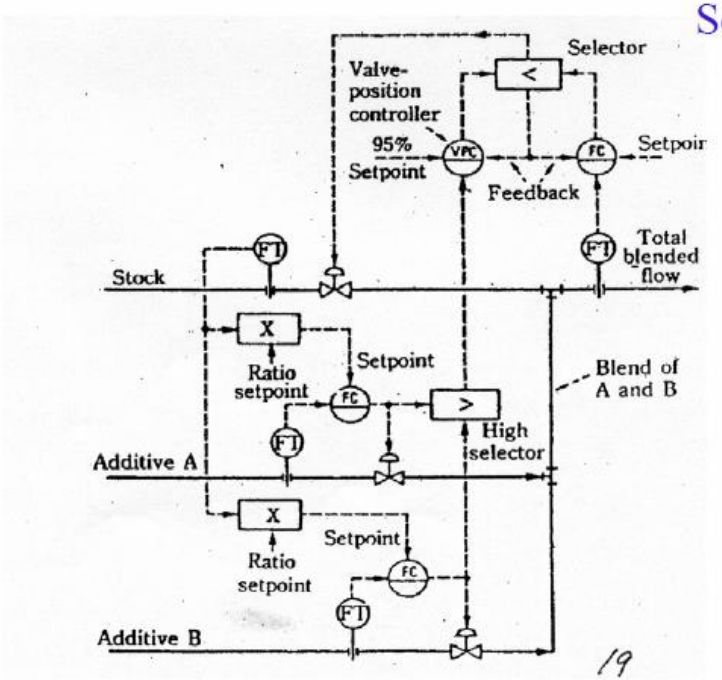
http://cepac.cheme.cmu.edu/pasilectures/lee/LecturenoteonMPC-JHL.pdf

Example 1: Blending System



- Control r_A and r_B
- Control q if possible
- Flowrates of additives are limited

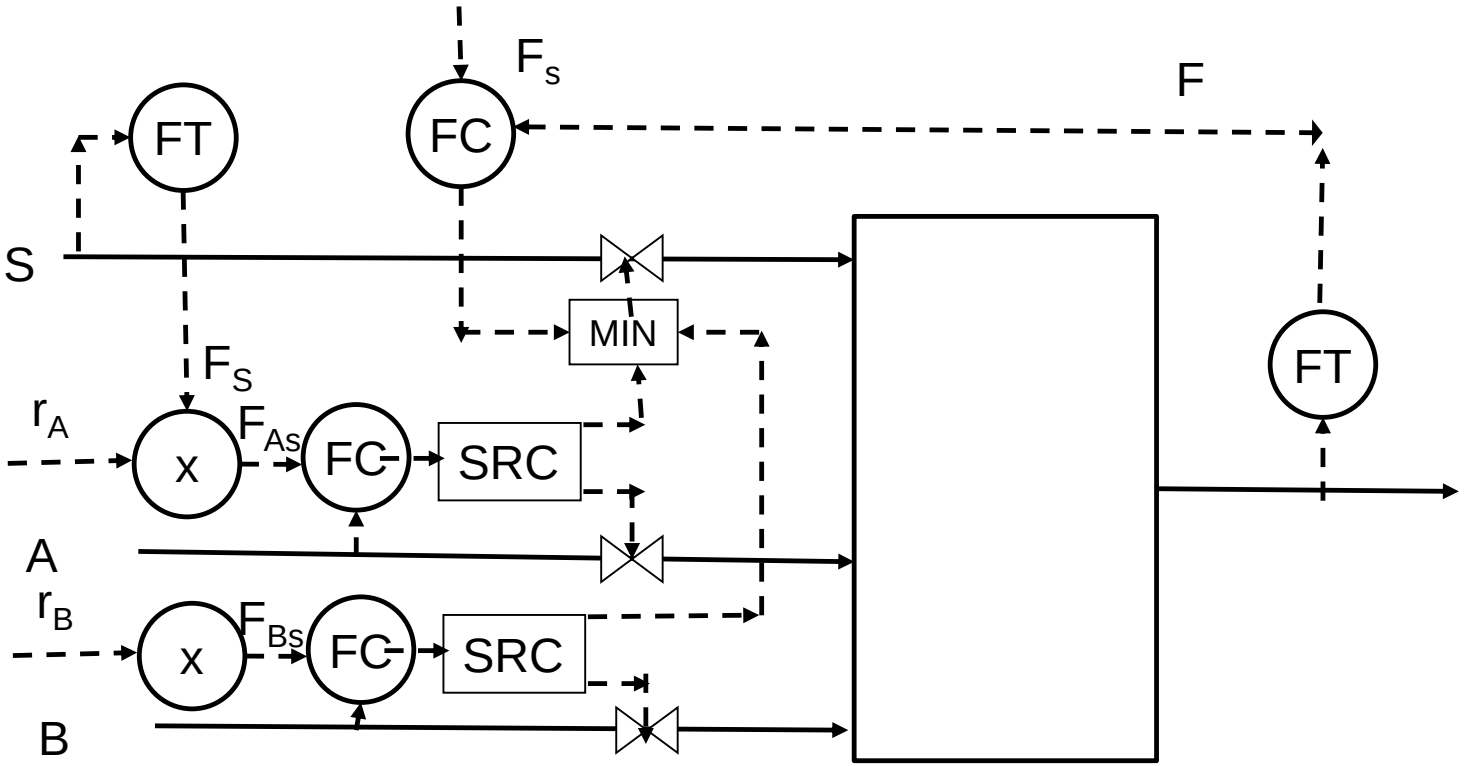
Classical Solution



Alt. 3. Valve position control:

- But get some back-off
- 95% of max. valve opening

Blending. Alt. 1. Split range control



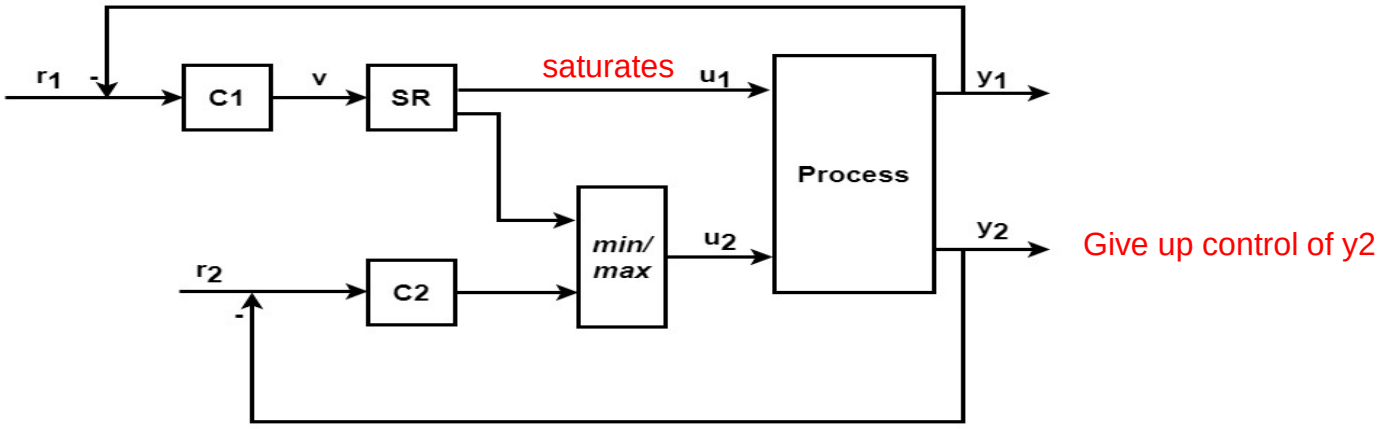
No back-off

3. MV-CV switching

- The MV that we are using to control an output (CV) saturates.

1. *Input saturation pairing rule* was followed (SISO): Do nothing.
   - Why? Because control of this CV should be given up anyway.
2. *Input saturation pairing rule* was **not** followed (MISO):
   - The CV that should be given up (y2) is then controlled by another MV (u2)
   - Implement an MV (u1)-MV (u2) switching strategy to maintain control of y1.

e.g. split range control with a min/max selector.



3. MV-CV switching. Example

Example: Control outlet temperature and flow of heat exchanger

High priority CV:

$y_1=T_H=T_H^{sp}$

Low priority CV:

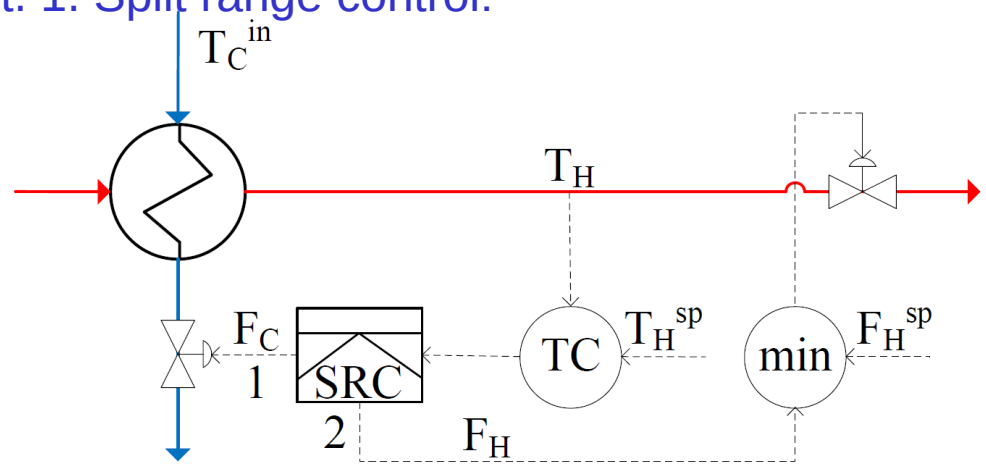
$y_2= F_H=F_H^{sp}$

Available MVs:

$u_1=z_C$, $u_2=z_H$ Both valves may saturate at max

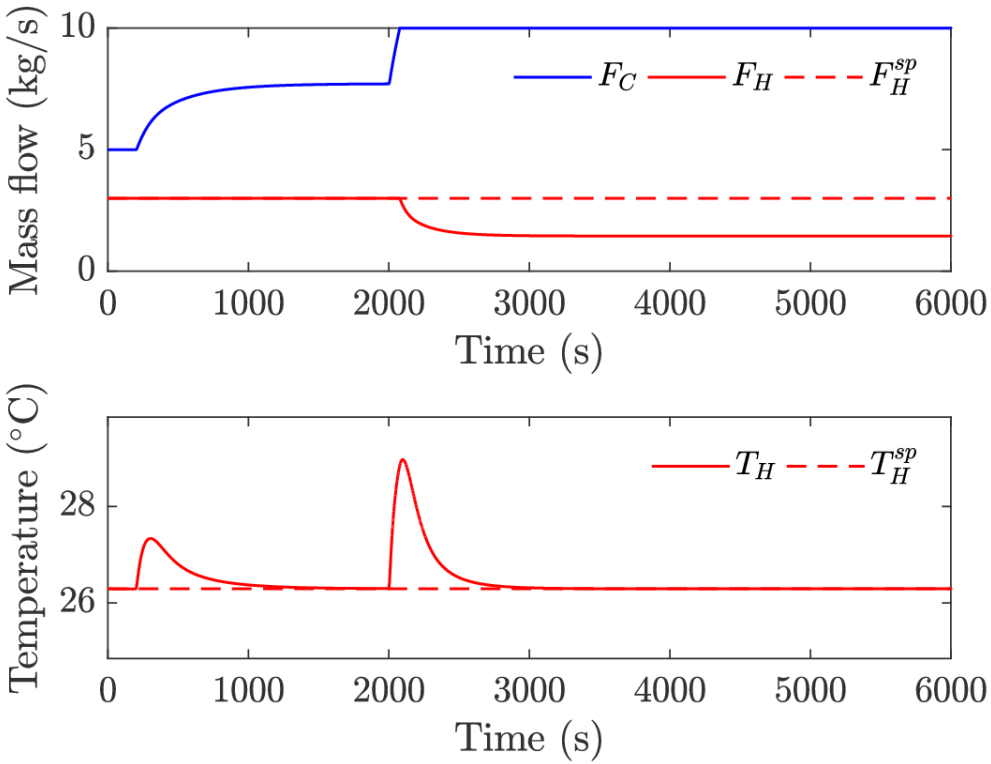
Disturbance: T_C^{in}

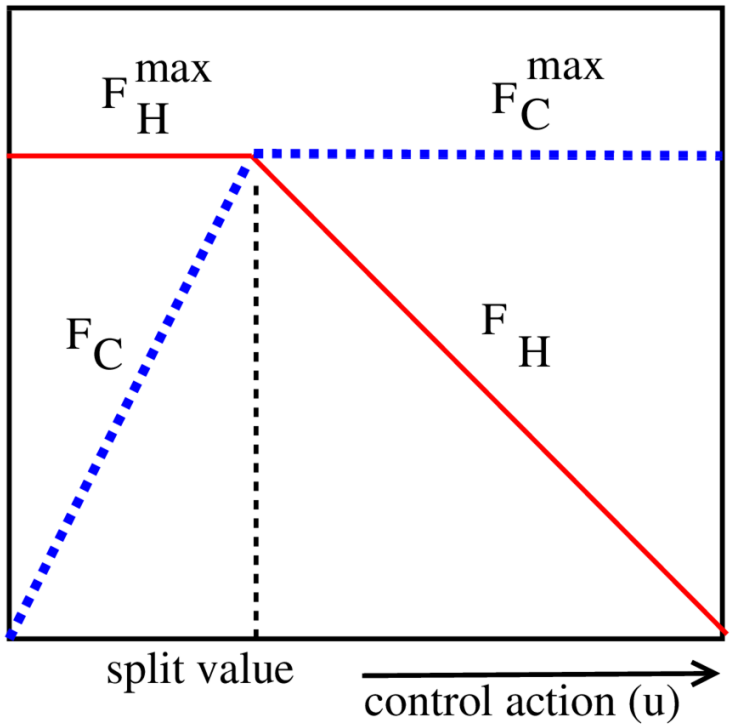
Alt. 1. Split range control:



A. Reyes-Lúa, C. Zotică, S. Skogestad, 2018. *Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control*. In: 10th ADCHEM. IFAC, Shenyang, China.

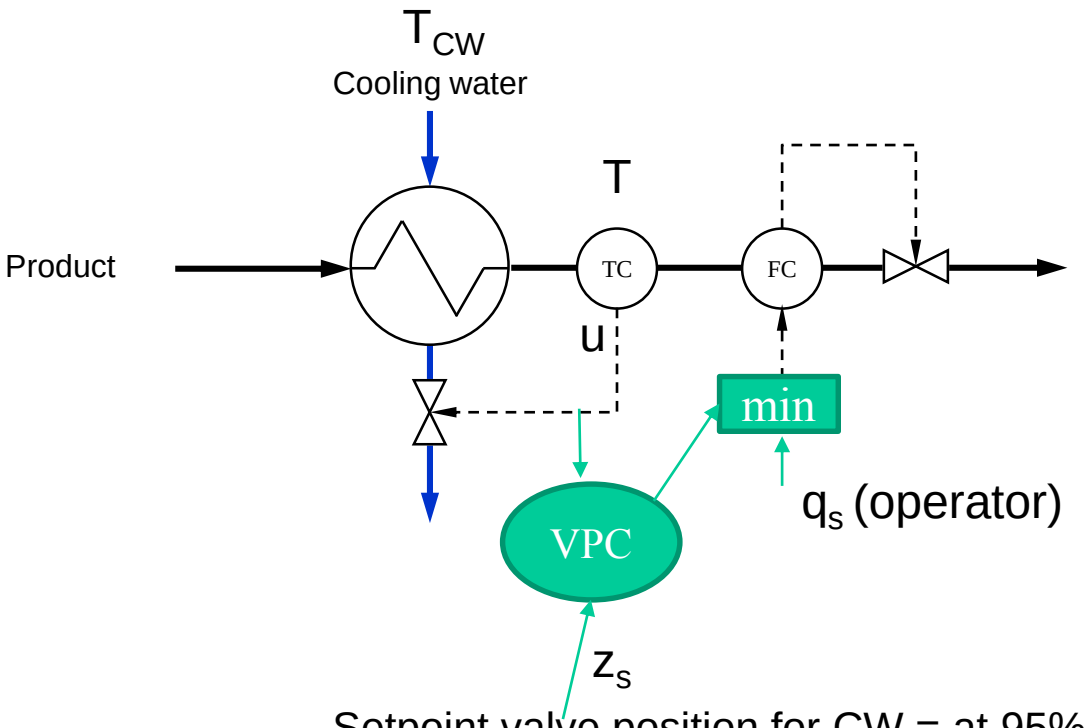
Simulations split range control….





A. Reyes-Lúa, C. Zotică, S. Skogestad, 2018. *Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control*. In: 10th ADCHEM. IFAC, Shenyang, China.

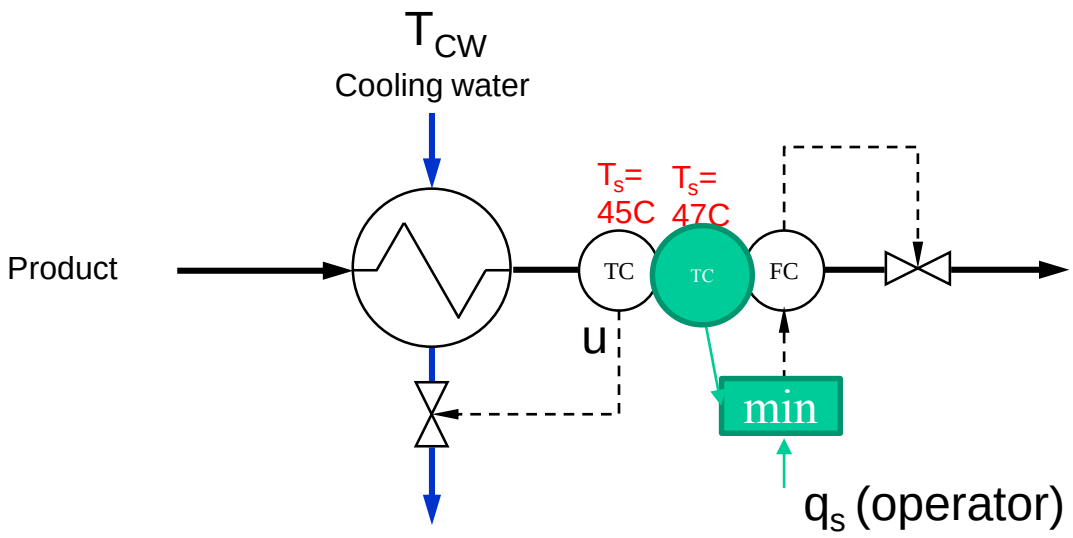
Alt.3 «valve position control» (suboptimal)



Setpoint valve position for CW = at 95% of max. opening (so some backoff)

Comment: 1. Has the advantage of not changing the controller for T.
2. With q_s large (infeasible setpoint) one can have q=qmax (as long as q_{CW} < max)
3. But cannot quite achieve fully open CW valve (so some loss = backoff)

Alt. 3. Two temperature controllers with different setpoints (slightly suboptimal).



Comment: 1. Similar to SRC, but avoids the SRC block.
2. Advantage: Two separate controllers
3. Disadvantage: Temperature a bit higher when we reach CW-constraint

Rules

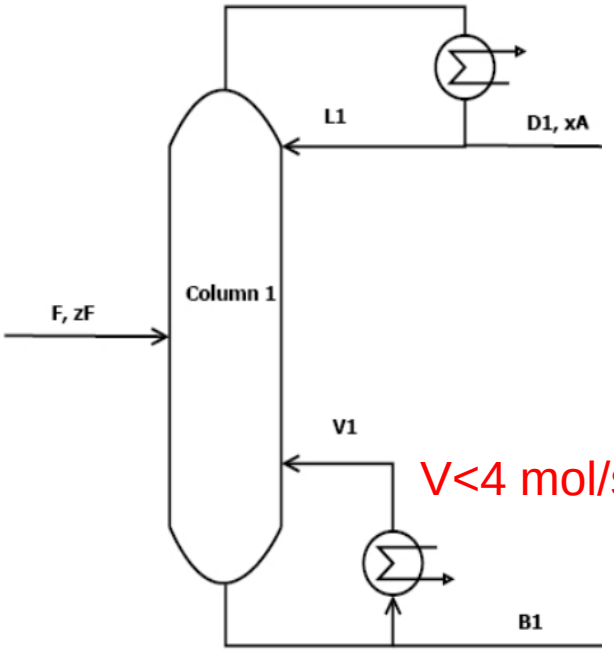
- Control actice constraints
- MV that may saturate should be paired with CV that may be given up
  - Alternative interpretation of rule:
    - Pair high-priority controlled variable (y, CV) (cannot be given up) with a manipulated variable (u, MV) that is not likely to saturate.
- TPM should be located close to bottleneck
  - Reason: Avoid «long loop» (and resulting backoff) when we have max. throughput
  - Bottleneck: Last constraint to be reached as we increase throughput
- *Arrange the inventory control loops (for level, pressures, etc.) around the TPM location according to the radiation rule (Georgakis)*
- *Select "sensitive/drifting" variables as controlled variables CV_2 for regulatory control.*

More rules

- Never control the cost function J (at fixed value)
  - May give infeasibility and certainly non-optimal operation
- Never do inventory control across TPM-location
  - Corresponds to pairing on zero

Distillation example

- Separate components A (light) and B (heavy)
- Cost (J) = - Profit = p_F F + p_VV – p_DD – p_BB
- Prices: p_F=p_D=1 $/mol, p_B=2 $/mol, Energy p_V= 0-0.2 $/mol (varies)
- With given feed and pressures: 2 steady-state DOFs.
- 3 constraints
  - Product purities (D,B) > 95%
  - capacity constraint on V



x_A>95%

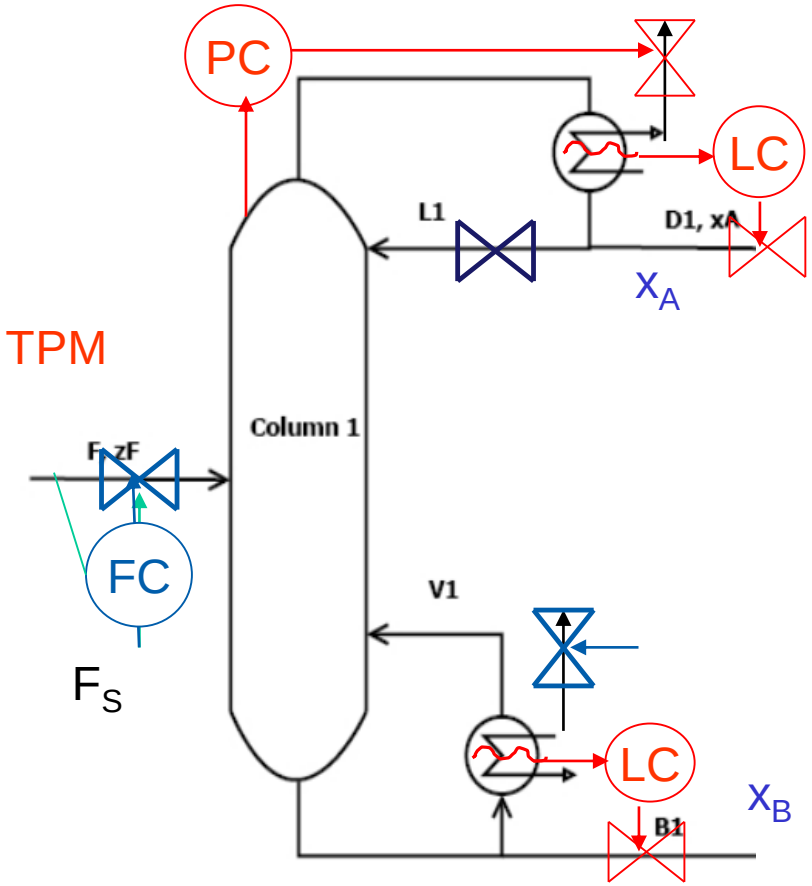
V<4 mol/s

xB>95%
Valueable product

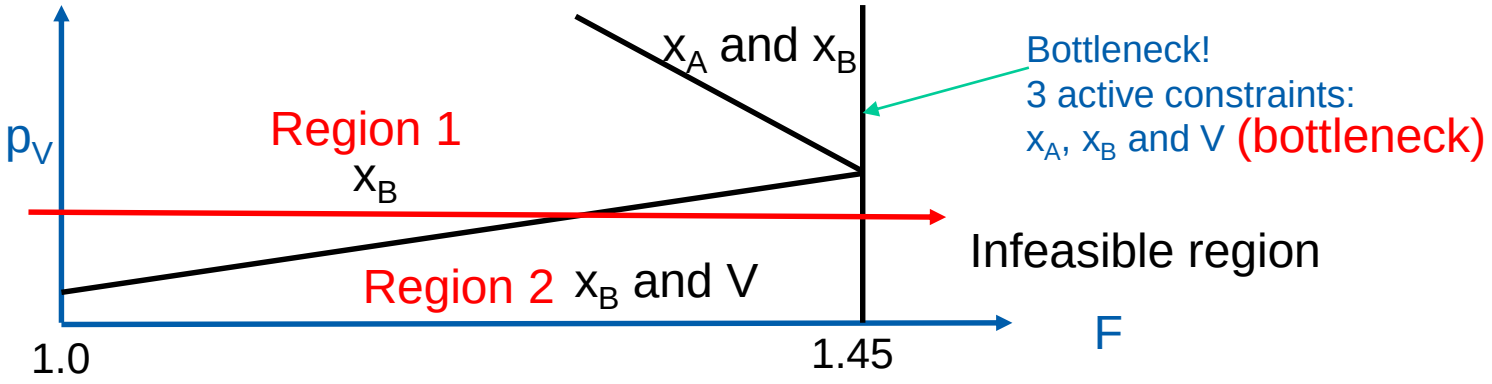
«Avoid product give-away» -> Valueable product constraint always active -> xB=95%

DOF = Degree Of Freedom

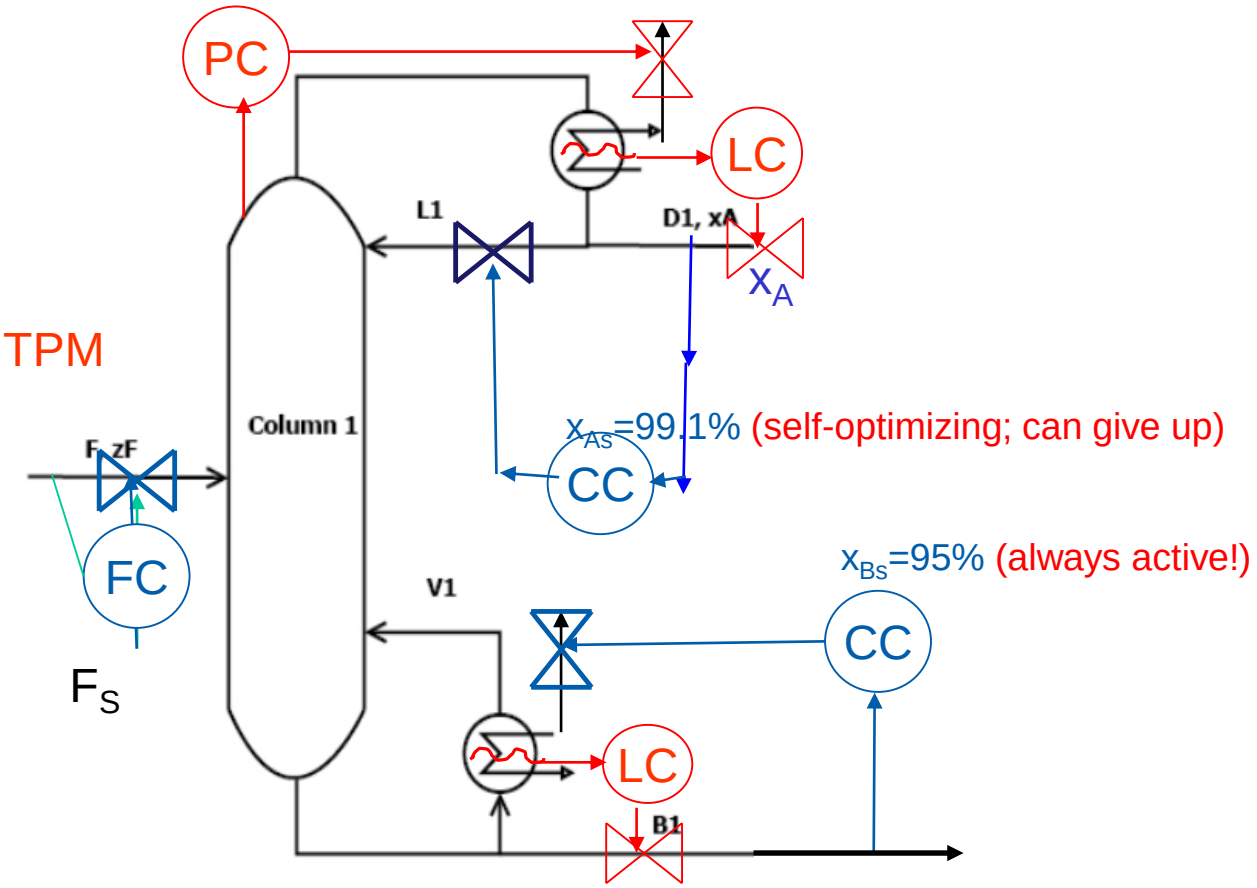
Regulatory control of levels and pressure



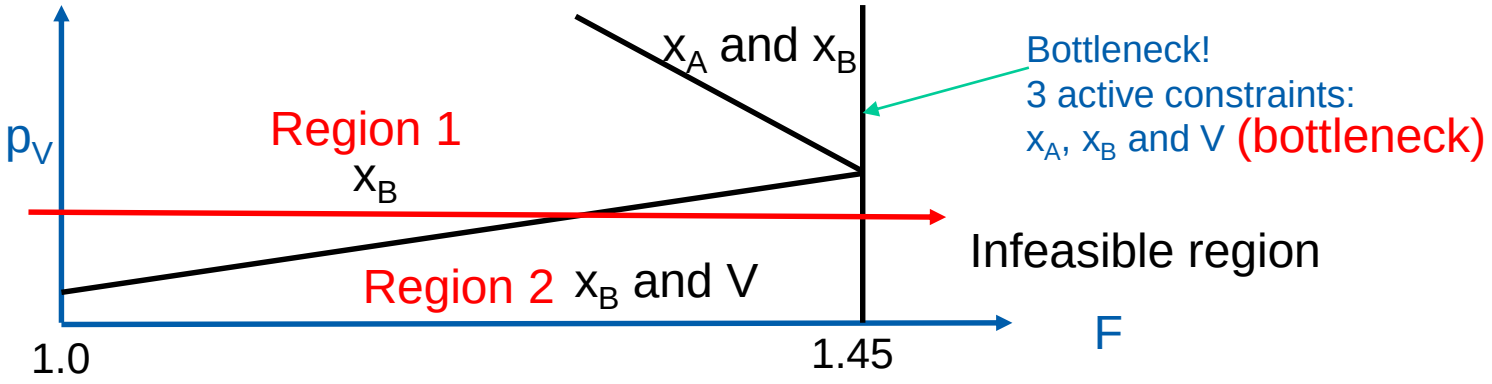
Three active constraint regions (+ bottleneck):



Region 1

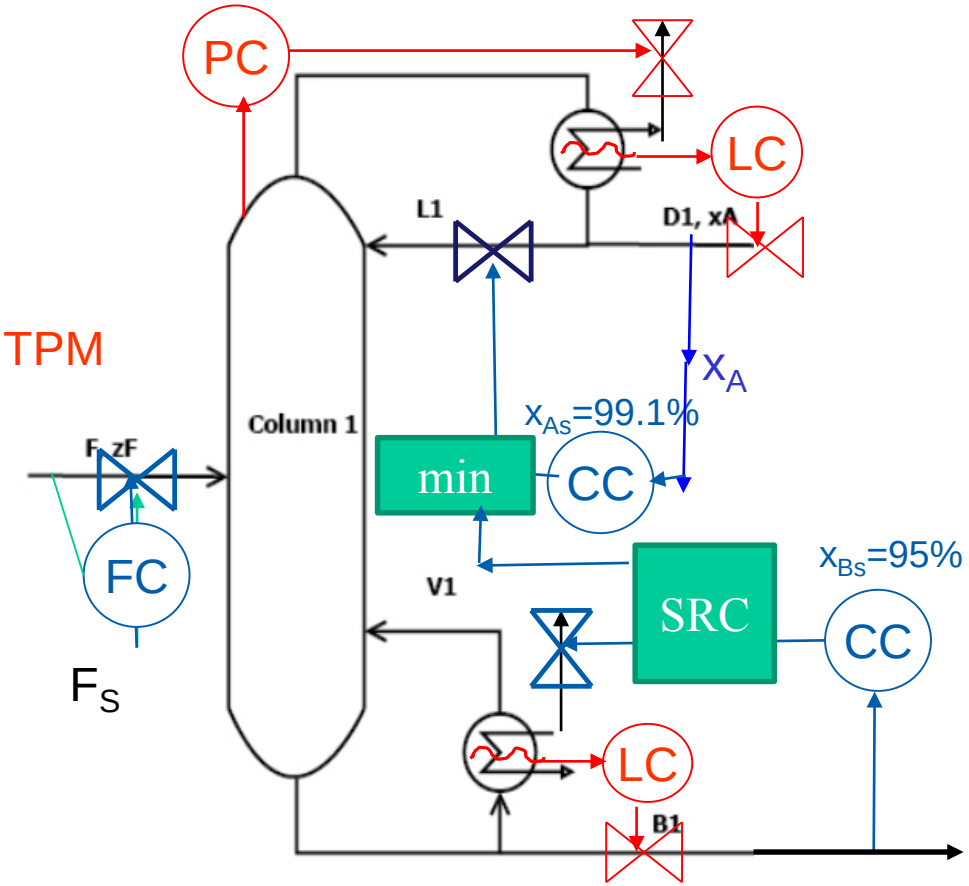


Three active constraint regions (+ bottleneck):

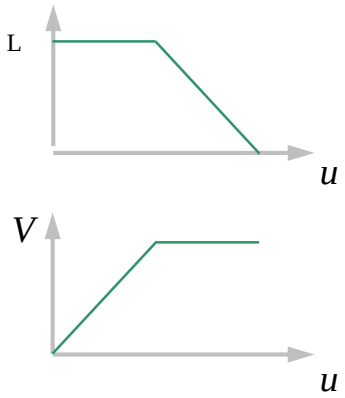


Region 1+2

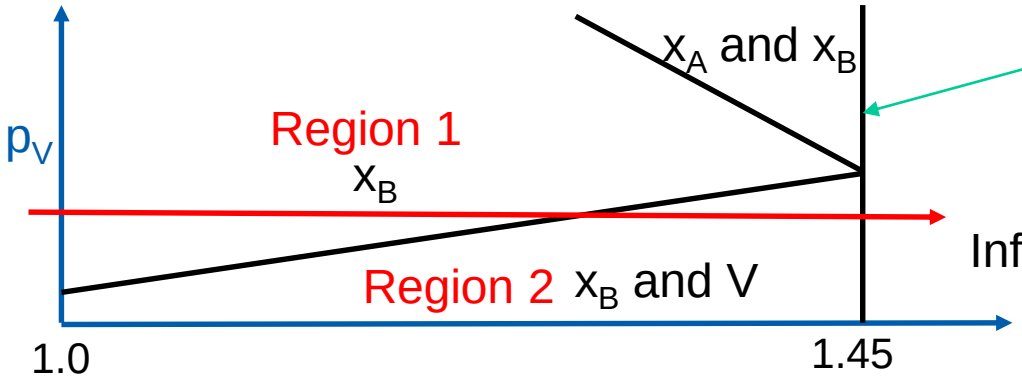
TPM



SRC:



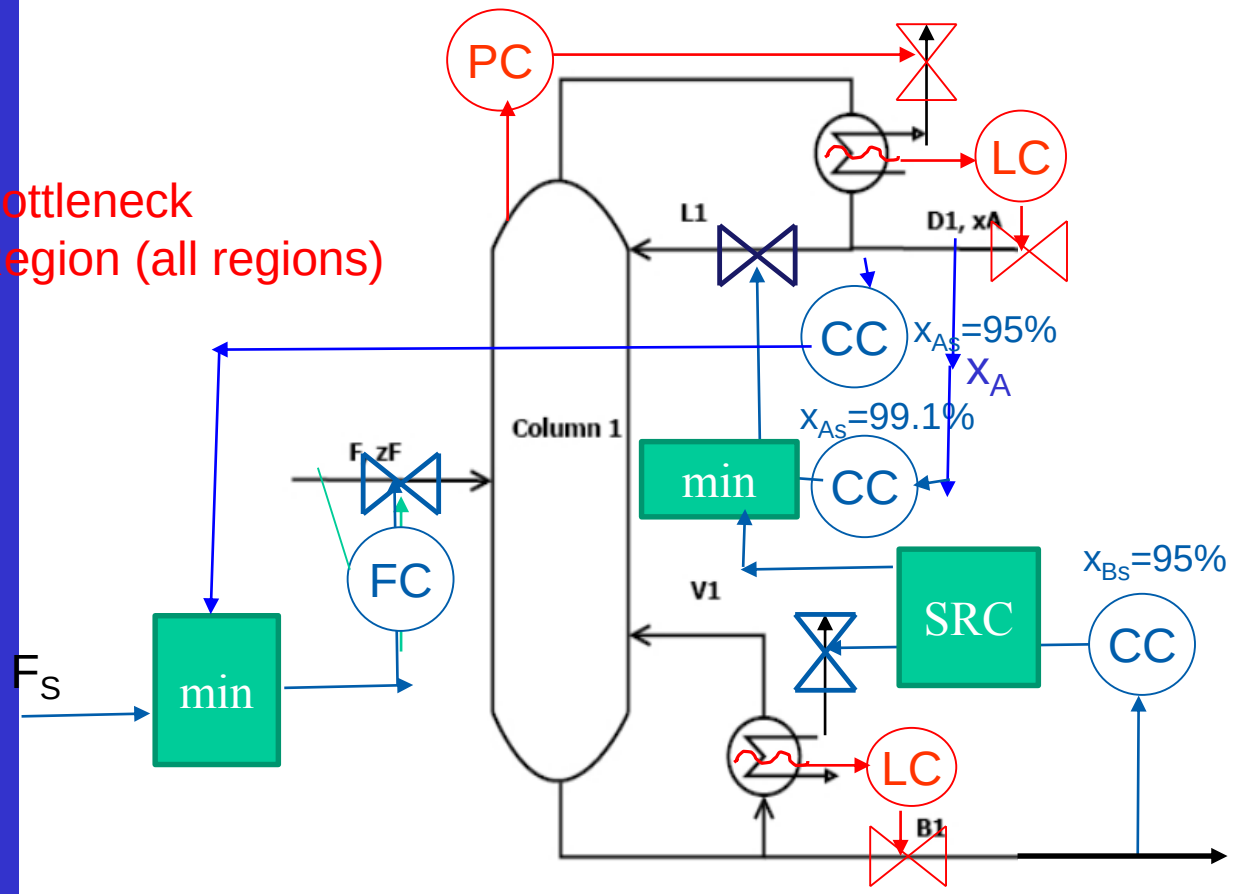
Three active constraint regions (+ bottleneck):

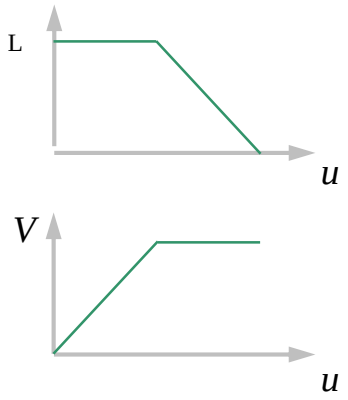


Bottleneck!
3 active constraints:
x_A, x_B and V (bottleneck)

Infeasible region

Bottleneck
Region (all regions)





Alternatives

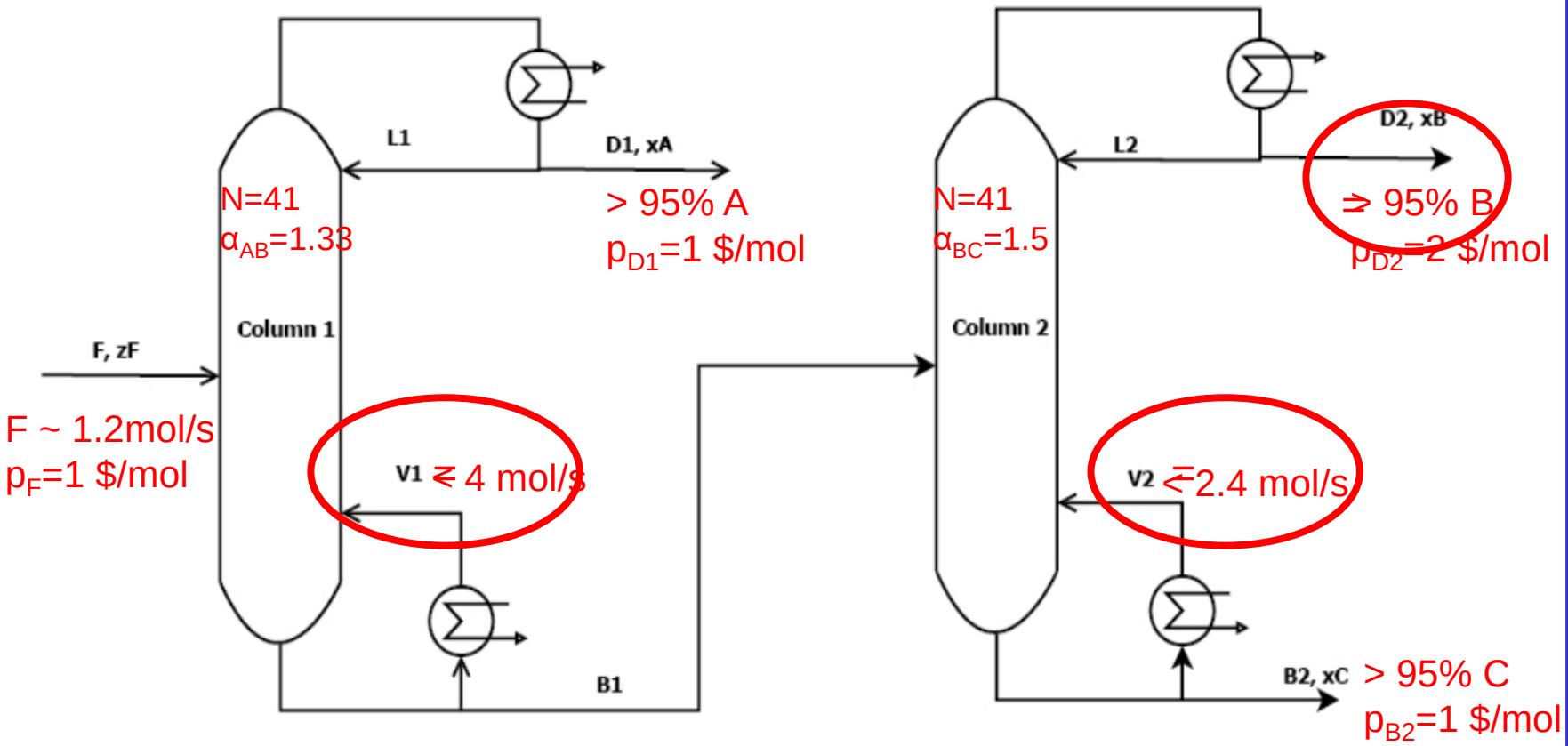
- Recerse pairing (pair on negative RGA)
- Valve position control (use L to avoid V saturating)

Problem

- Price changes (pv). Must be handled by «feed forward» since they do not affect the process
- That is, optimal «overpurification» setpoint (which is 99.1% in this example), depends on price pv.
- At some point it becomes 95% and we have the «xA xB»-region! So all regions are handled by a single structure

Operation of Distillation columns in series

- Cost (J) = - Profit = $p_F F + p_V(V_1+V_2) - p_{D1}D_1 - p_{D2}D_2 - p_{B2}B_2$
- Prices: $p_F=p_{D1}=P_{B2}=1$ \$/mol, $p_{D2}=2$ \$/mol, Energy p_V= 0-0.2 \$/mol (varies)
- With given feed and pressures: 4 steady-state DOFs.
- Here: 5 constraints (3 products > 95% + 2 capacity constraints on V)

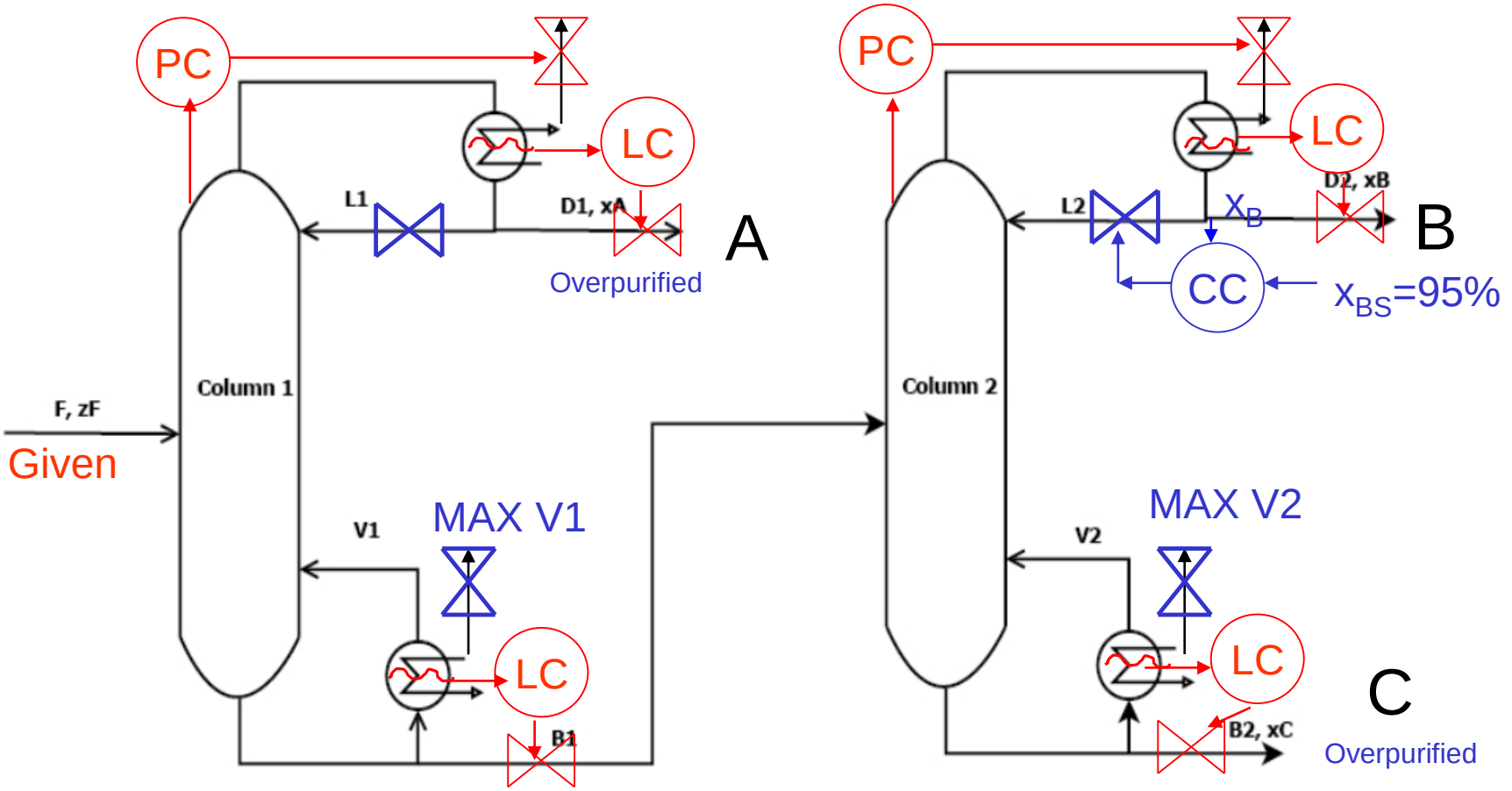


QUIZ: What are the expected active constraints?
1. Always. 2. For low energy prices.

DOF = Degree Of Freedom
Ref.: M.G. Jacobsen and S. Skogestad (2011)

Example.

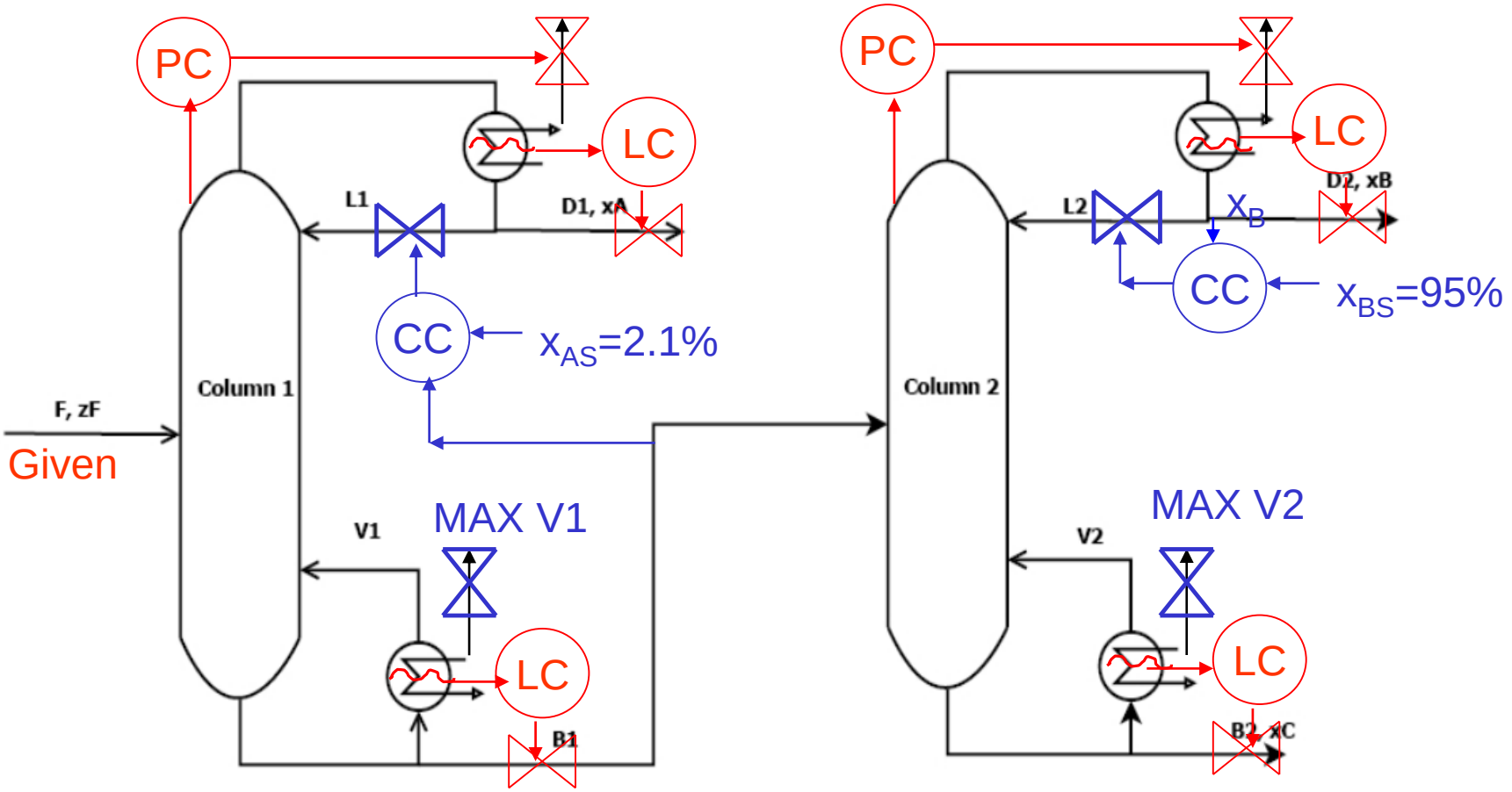
Control of Distillation columns. Cheap energy



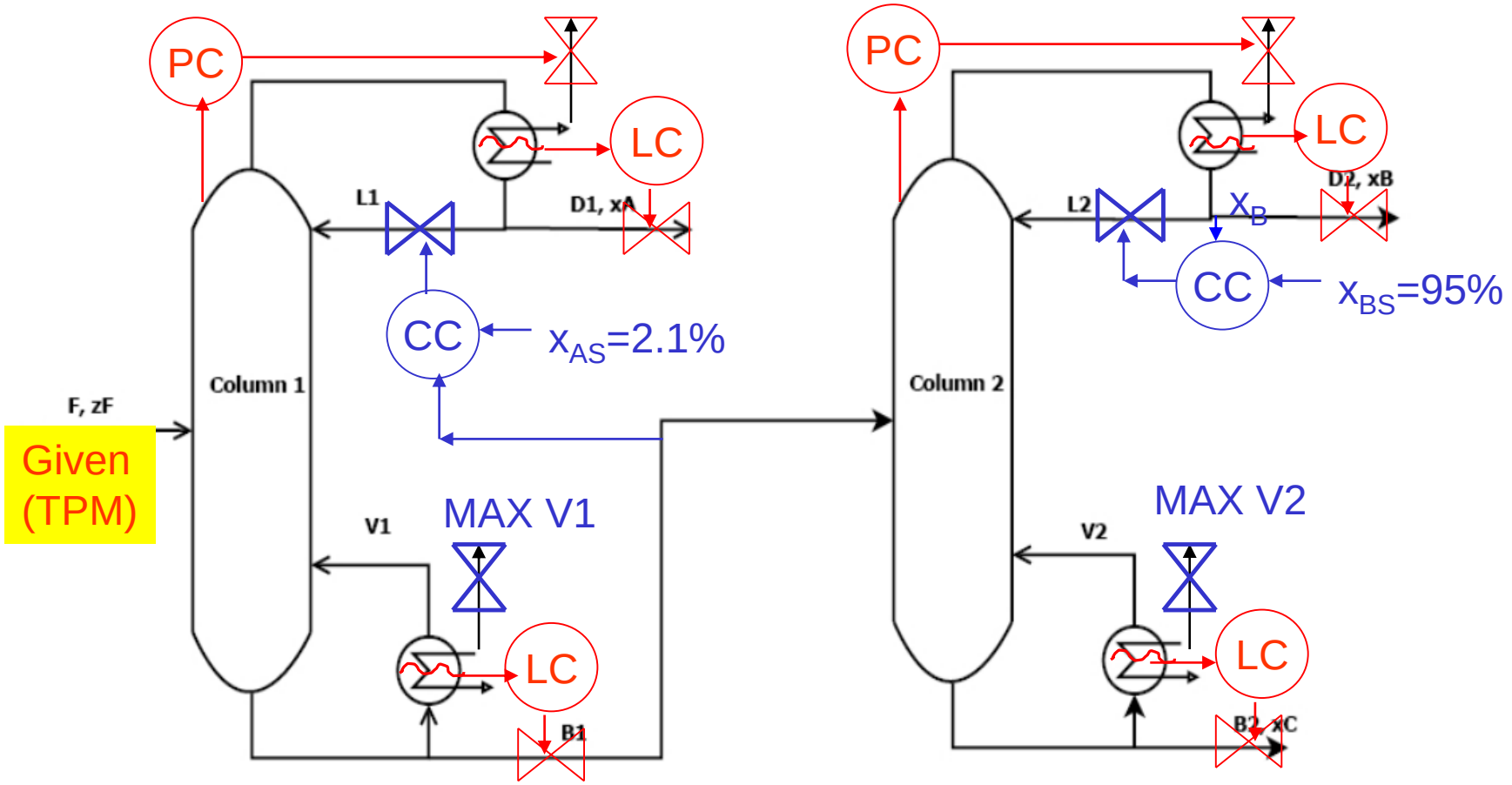
Overpurified: To avoid loss of valuable product B

Solution.

Control of Distillation columns. Cheap energy

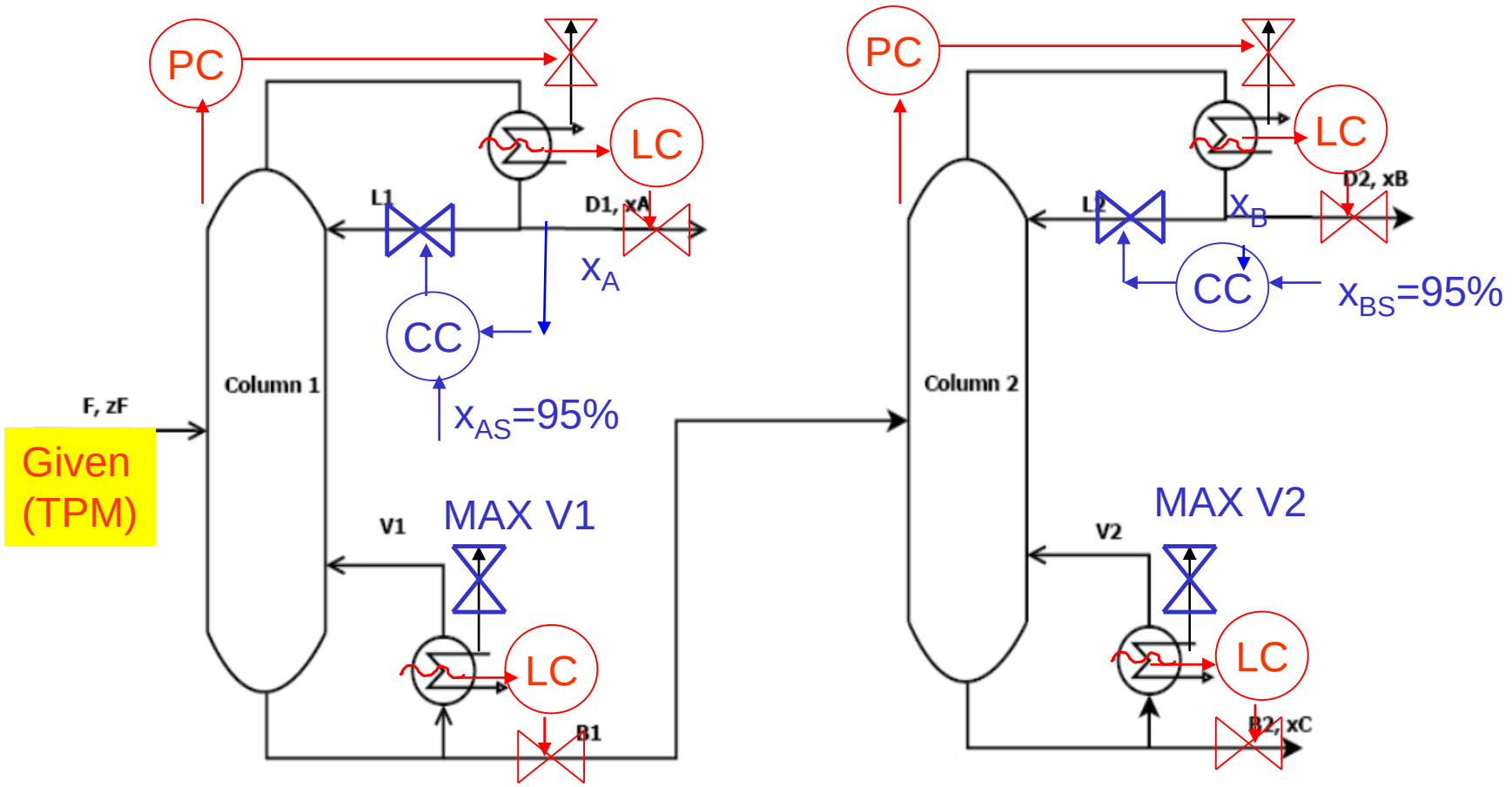


What happens if we increase the federate?

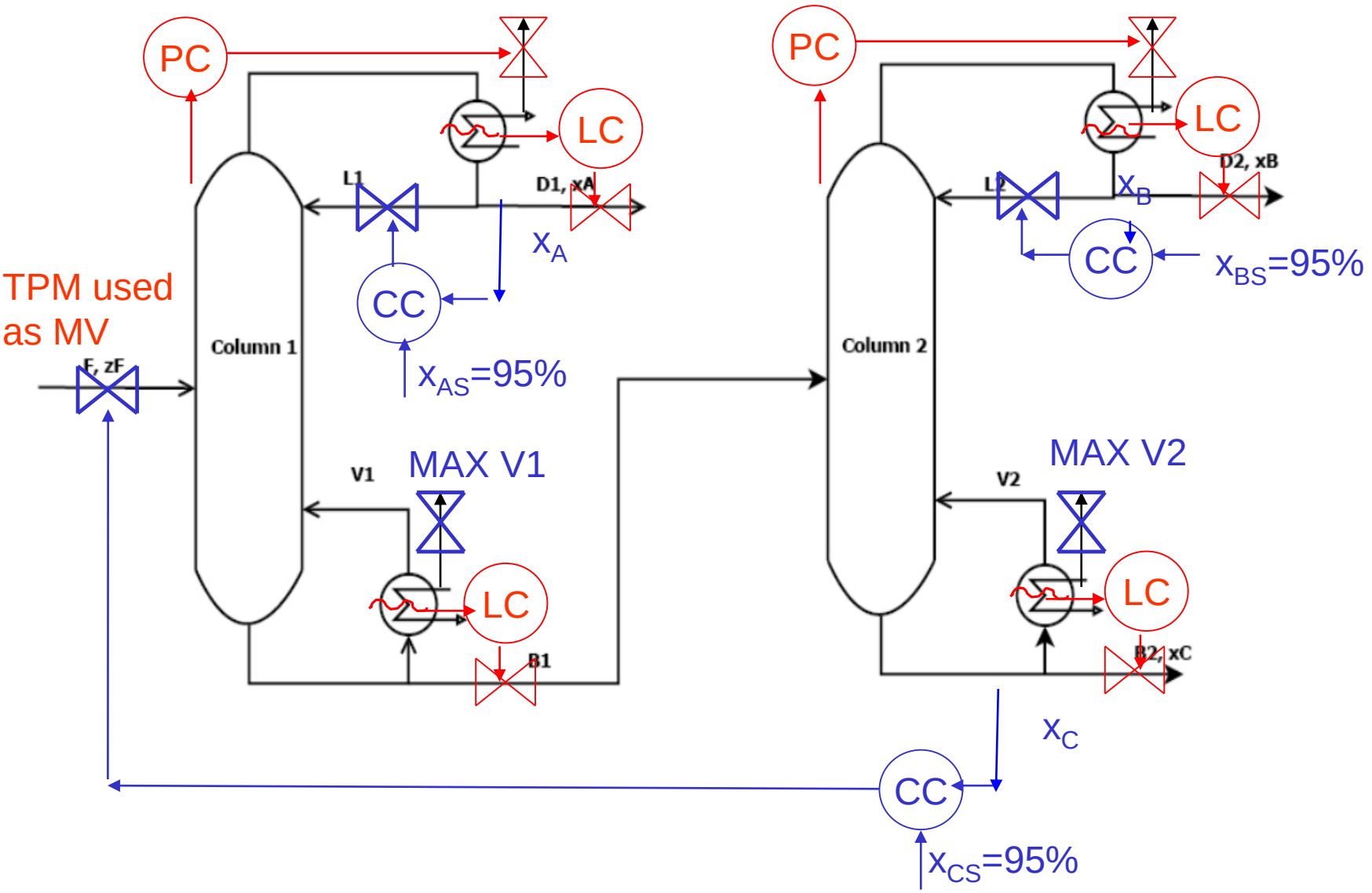


Is this control structure still OK??

Increase federate: Reach x_A-constraint

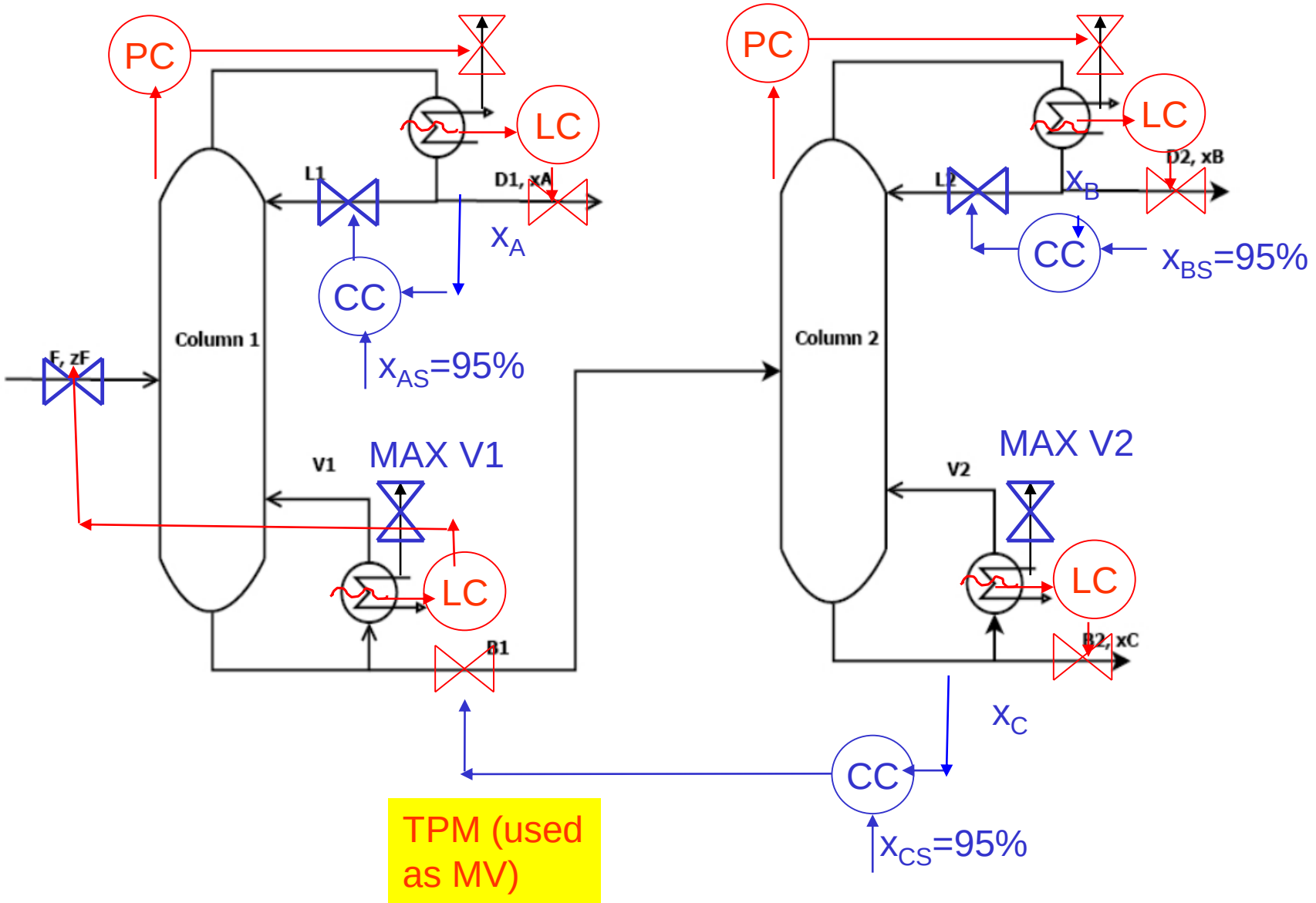


Increase federate further: Reach also x_C-constraint (Bottleneck)

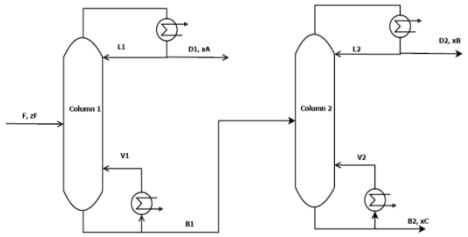


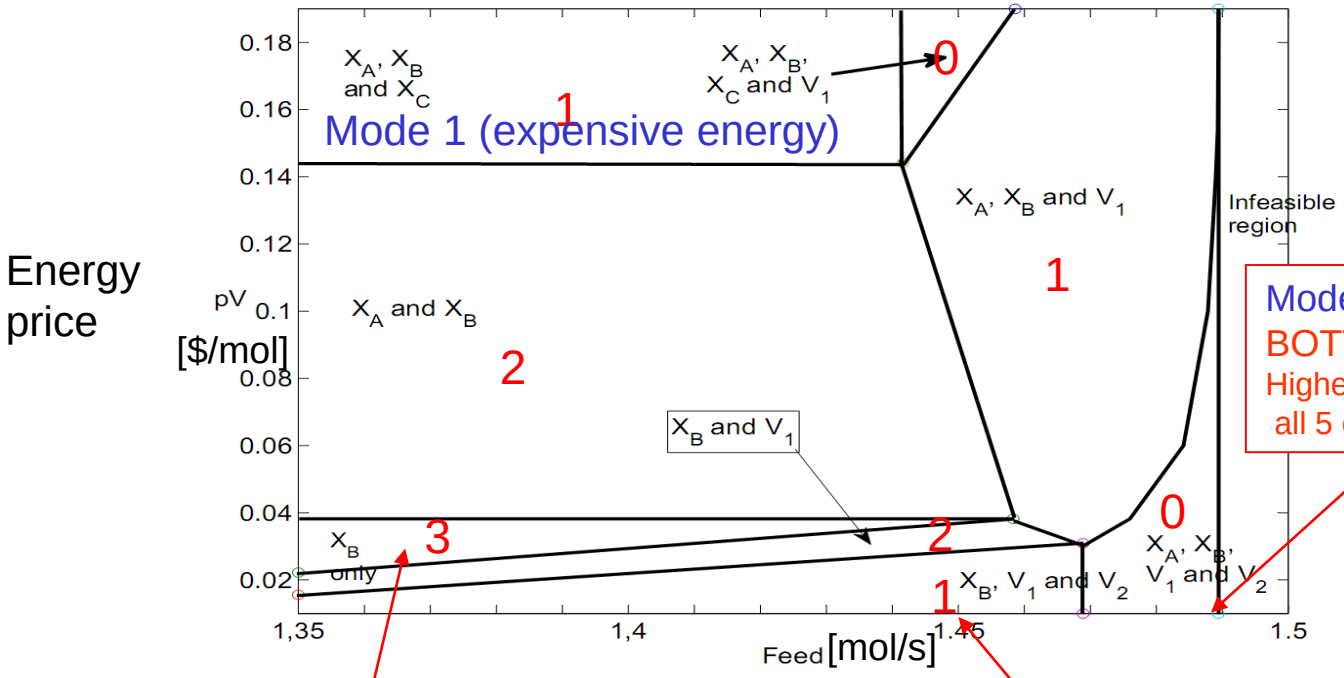
Move TPM

Move TPM to F2 (closer to bottleneck) and rearrange level loop



Active constraint regions for two distillation columns in series





Mode 2: operate at BOTTLENECK. F=1,49
Higher F infeasible because all 5 constraints reached

Mode 1, Cheap energy: 3 active constraints -> 1 remaining unconstrained DOF (L_1) -> Need to find 1 additional CVs ("self-optimizing")

More expensive energy: Only 1 active constraint (xB) ->3 remaining unconstrained DOFs -> Need to find 3 additional CVs ("self-optimizing")

CV = Controlled Variable

How many active constraints regions?

- Maximum: 2^{n_c}

  n_c = number of constraints

Distillation
$n_c = 5$
$2^5 = 32$

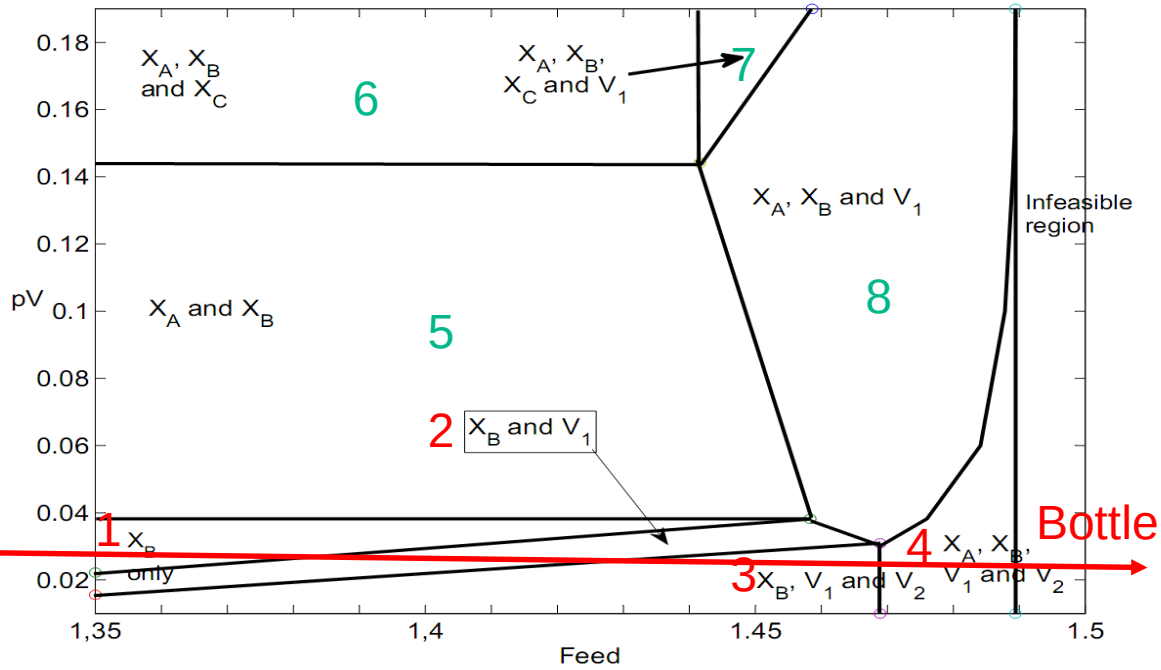
BUT there are usually fewer in practice

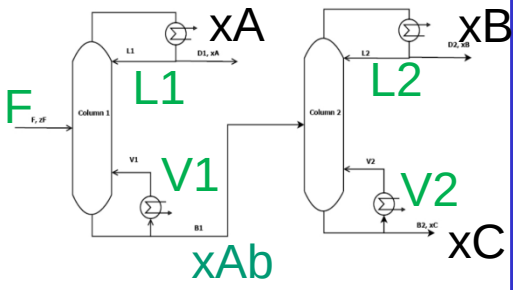
- Certain constraints are always active (reduces effective n_c)
- Only n_u can be active at a given time

  n_u = number of MVs (inputs)
- Certain constraints combinations are not possibe
  - For example, max and min on the same variable (e.g. flow)
- Certain regions are not reached by the assumed disturbance set

x_B always active
2^4 = 16
-1 = 15

In practice = 8





CV regions with suggested pairings

MV	1	2	3	4	Bottleneck (5 constraints)
F	F	«	«	«	xC=0.95 (min.select)
L1	xA=0.991	XAb=0.023 (SRC+min.select)*	«	xA=0.95 (max.select)	«
V1	XAb=0.023	V1=max	«	«	«
L2	xB=0.95	«	«	«	«
V2	xC=0.993	«	V2=max	«	«

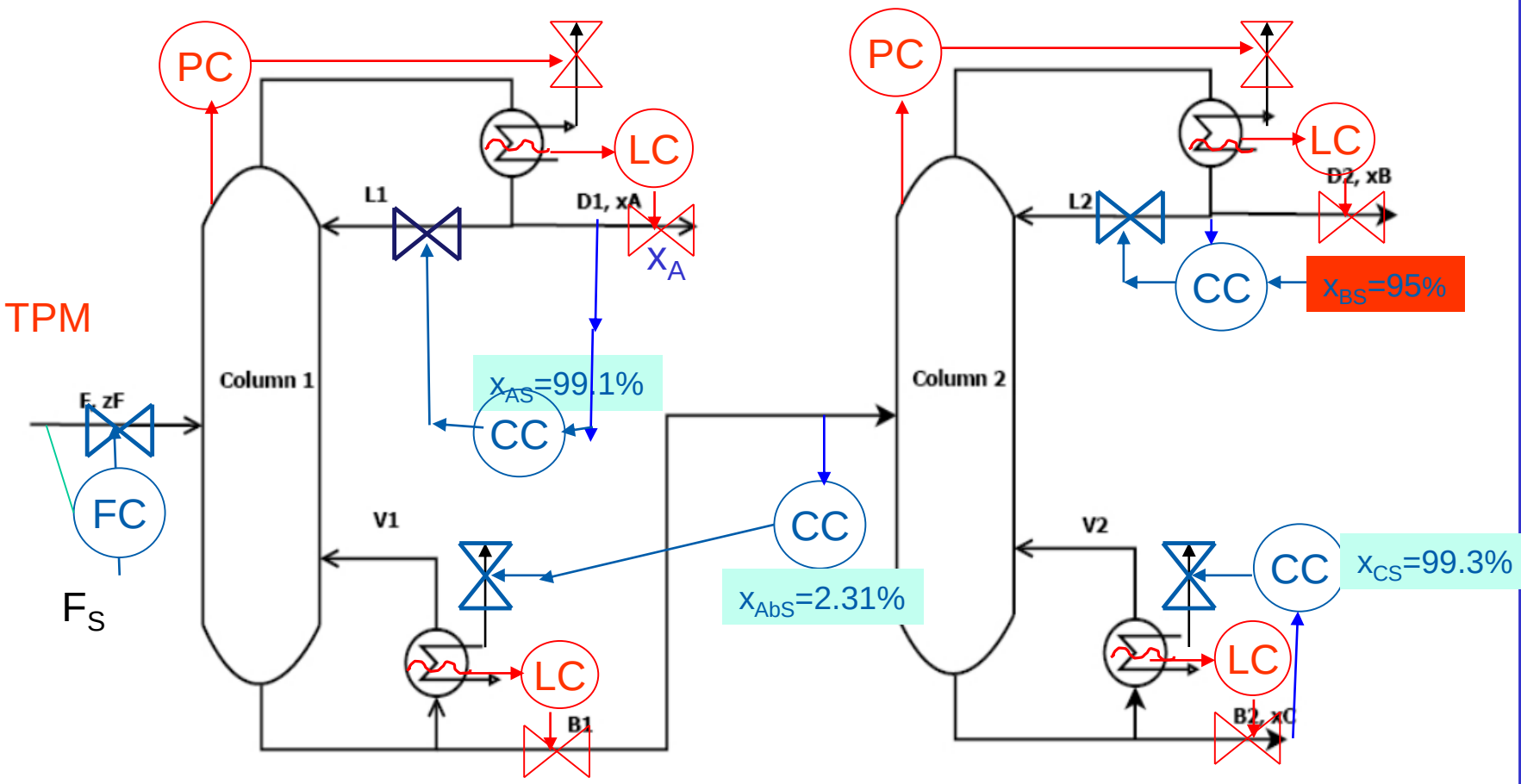
Blue: unconstrained optimal values (depend on energy price)

*Could avoid with reverse pairing in region 1 (pair on negative RGA)

Constraints: xA, xC, V1, V2 (xB always active

1. No constraints (1)
2. xA (5) (handled OK, happens when energy is expensive so xAopt reaches 95%)
3. xC (handled OK, happens if column 2 is shorter and energy expensive so xCopt reaches 95%)
4. V1 (2)
5. V2 (handled OK, happens if column 2 is shorter)
6. xA, xC (6) (handled OK)
7. xA, V1 (8) (NOT handled, two constraints in column 1, so will be bottleneck!)
8. xA, V2 (handled OK, happens if column 2 is shorter)
9. xC, V1 (handled OK)
10. xC, V2 (NOT handled, three constraints in column 2, so must use column 1 to control xC)
11. V1, V2 (3)
12. xA, xC, V1 (7) (NOT handled; bottleneck; see region 7)
13. xA, V1, V2 (4)
14. xc, V1, V2 (NOT handled, see region 10)
15. xA, xB, xC, V1, V2 (Bottleneck)

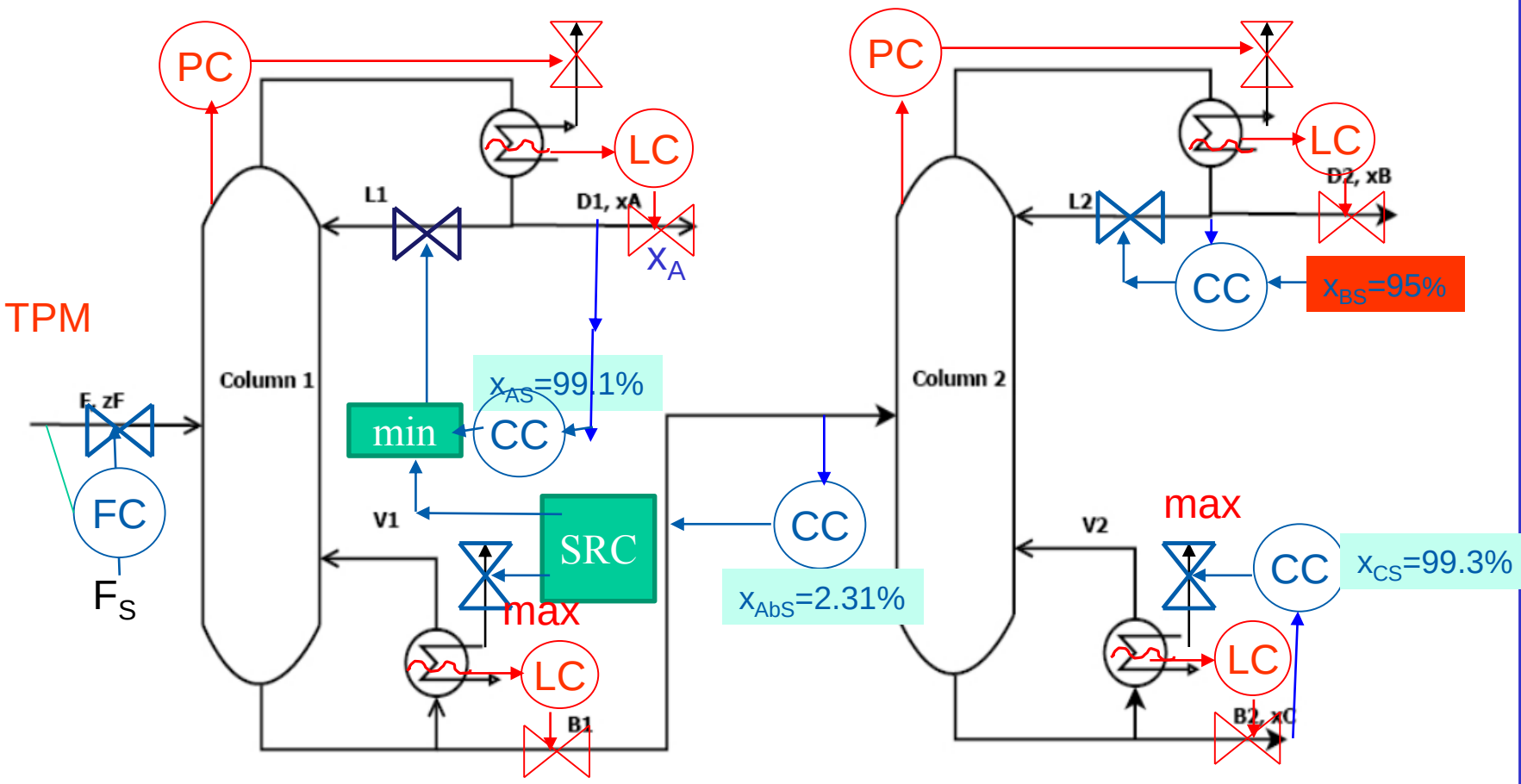
Solution for low federate (region 1; intermediate energy prices)



Self-optimizing variable: Can be given up. Optimal value depends on product prices

Active constraint (because B is valuable product)

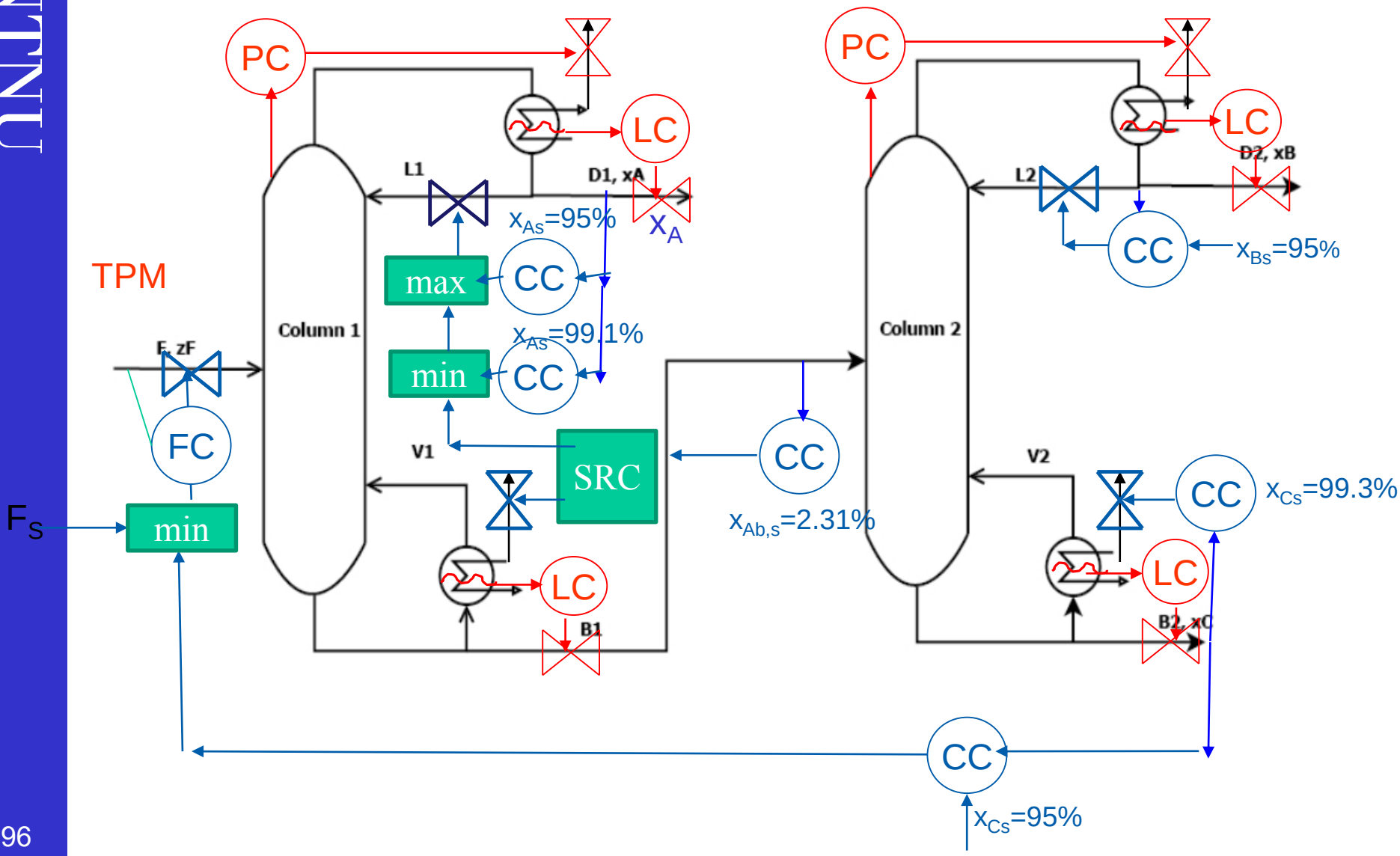
Solution for higher federate (region 2)



Self-optimizing variable: Can be given up

Active constraint

Solution for all regions (including regions 3 and 4)



Comment: Use of extra inputs

Two different cases

1. **Have extra dynamic inputs (degrees of freedom)**

   Cascade implementation: "Input resetting to ideal resting value"

   Example: Heat exchanger with extra bypass

   Also known as: Midranging control, valve position control

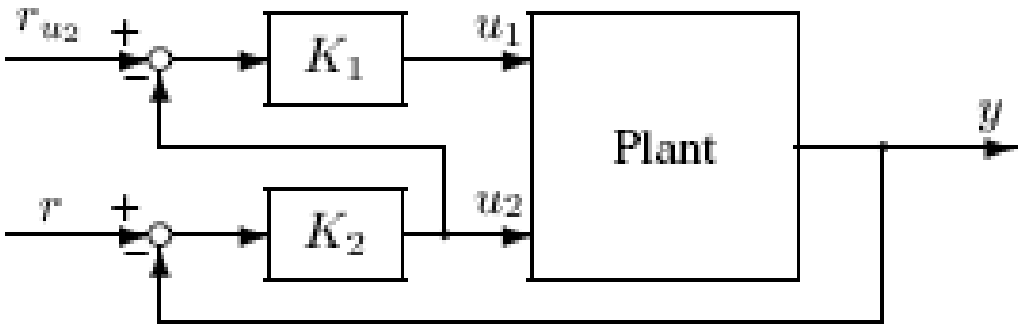
2. Need several inputs to cover whole range (because primary input may saturate) (steady-state)

   Split-range control

   Example 1: Control of room temperature using AC (summer), heater (winter), fireplace (winter cold)

   Example 2: Pressure control using purge and inert feed (distillation)

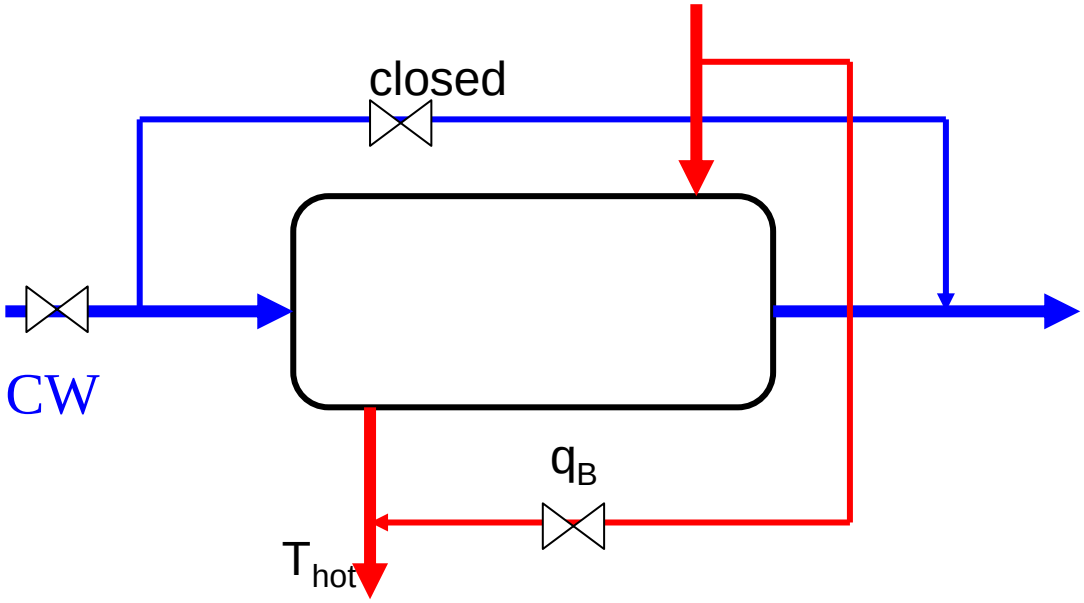
Extra inputs, dynamically



(b) Extra inputs u_2 (input resetting)

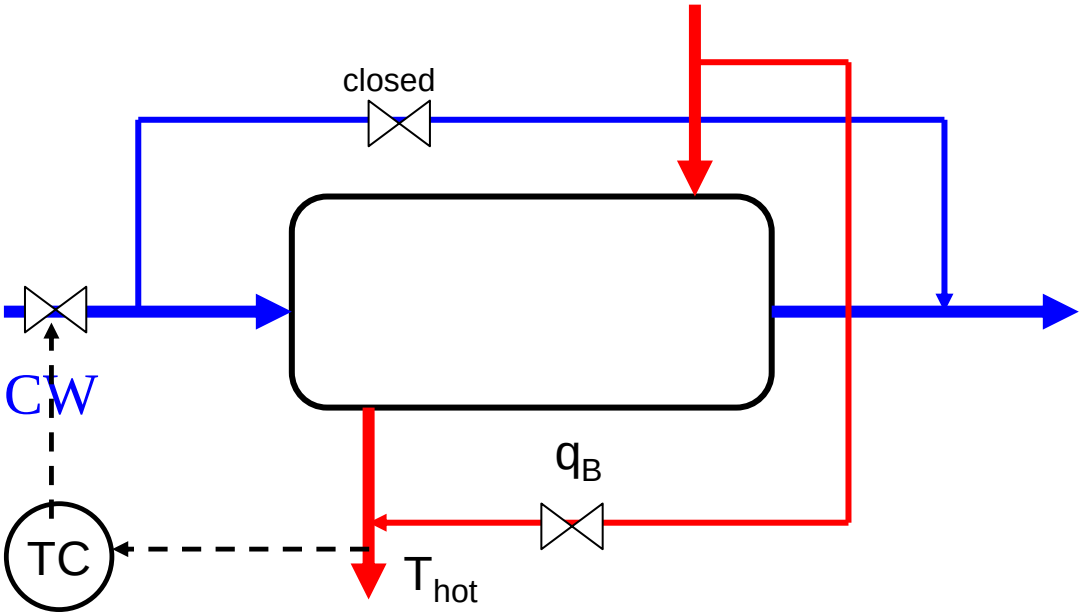
- Exercise: Explain how “valve position control” fits into this framework. As en example consider a heat exchanger with bypass

QUIZ: Heat exchanger with bypass



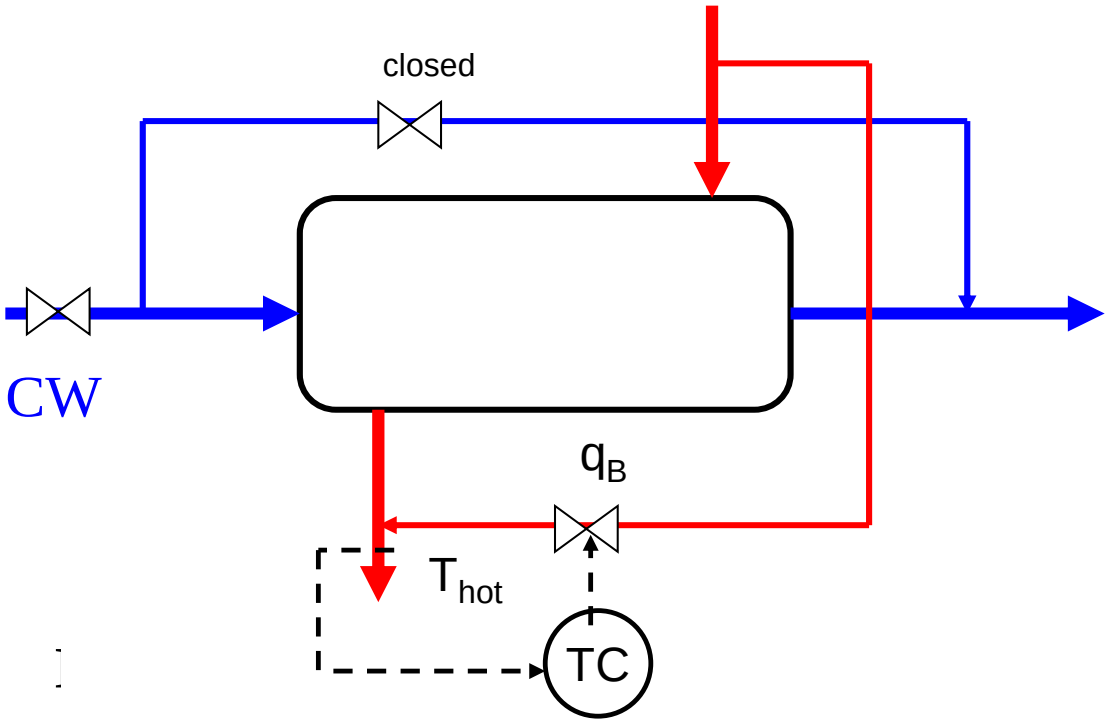
- Want tight control of T_{hot}
- Primary input: CW
- Secondary input: q_B
- Proposed control structure?

Alternative 1



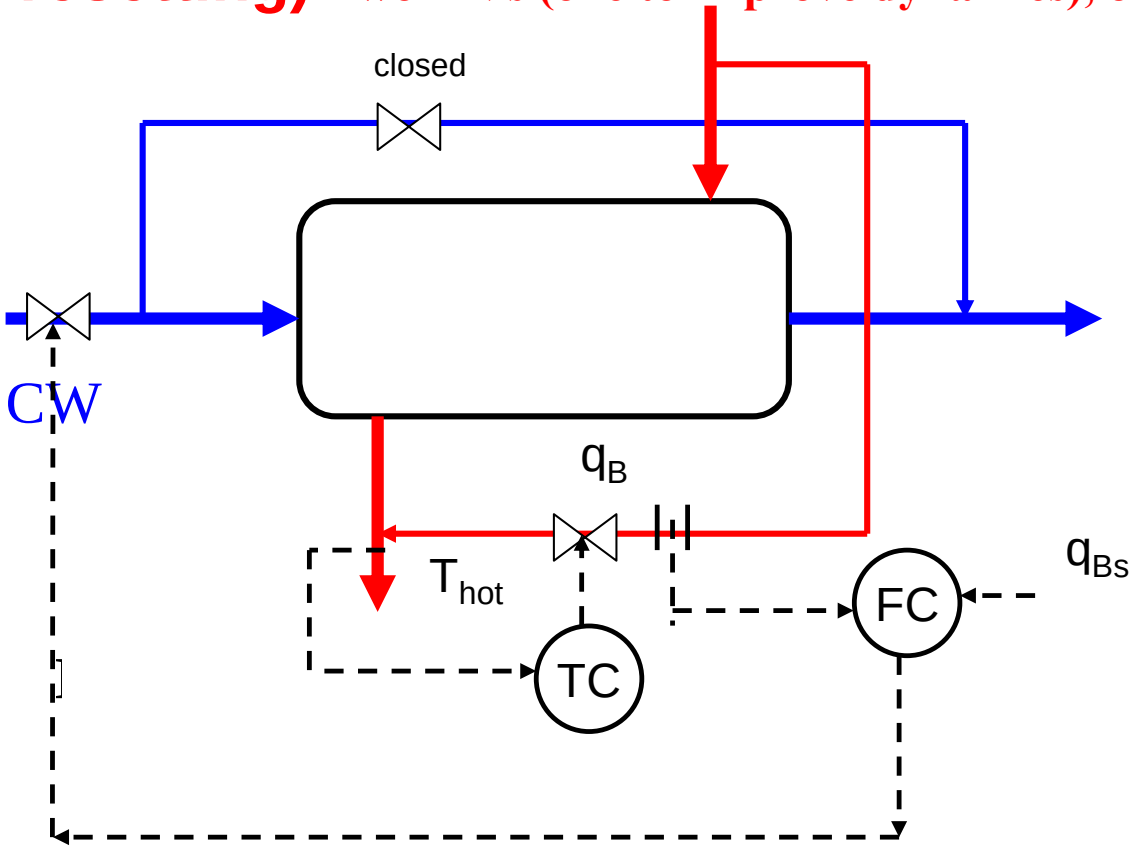
Use primary input CW: TOO SLOW

Alternative 2



Use “dynamic” input q_B
Advantage: Very fast response (no delay)
Problem: q_B is too small to cover whole range
+ has small steady-state effect

Alternative 3: **Valve position control (input resetting)** Two MVs (one to improve dynamics), one CV



TC: Gives fast control of T_{hot} using the "dynamic" input q_B
FC: Resets q_B to its setpoint (IRV) (e.g. 5%) using the "primary" input CW

IRV = ideal resting value

Also called: "valve position control" (Shinskey) and "midranging control" (Sweden)

Outline

- Skogestad procedure for control structure design

I Top Down

- Step S1: Define operational objective (cost) and constraints
- Step S2: Identify degrees of freedom and optimize operation for disturbances
- Step S3: Implementation of optimal operation
  - What to control ? (primary CV's) (self-optimizing control)
- Step S4: Where set the production rate? (Inventory control)

II Bottom Up

- Step S5: Regulatory control: What more to control (secondary CV's) ?
- Step S6: Supervisory control
- Step S7: Real-time optimization

Step S7. Optimization layer (RTO)

- *Purpose:* Identify active constraints and compute optimal setpoints (to be implemented by supervisory control layer)
- *Main structural issue:* Do we need RTO? (or is process self-optimizing)

- RTO not needed when
  - Can "easily" identify change in active constraints (operating region)
  - For each operating region there exists self-optimizing variables