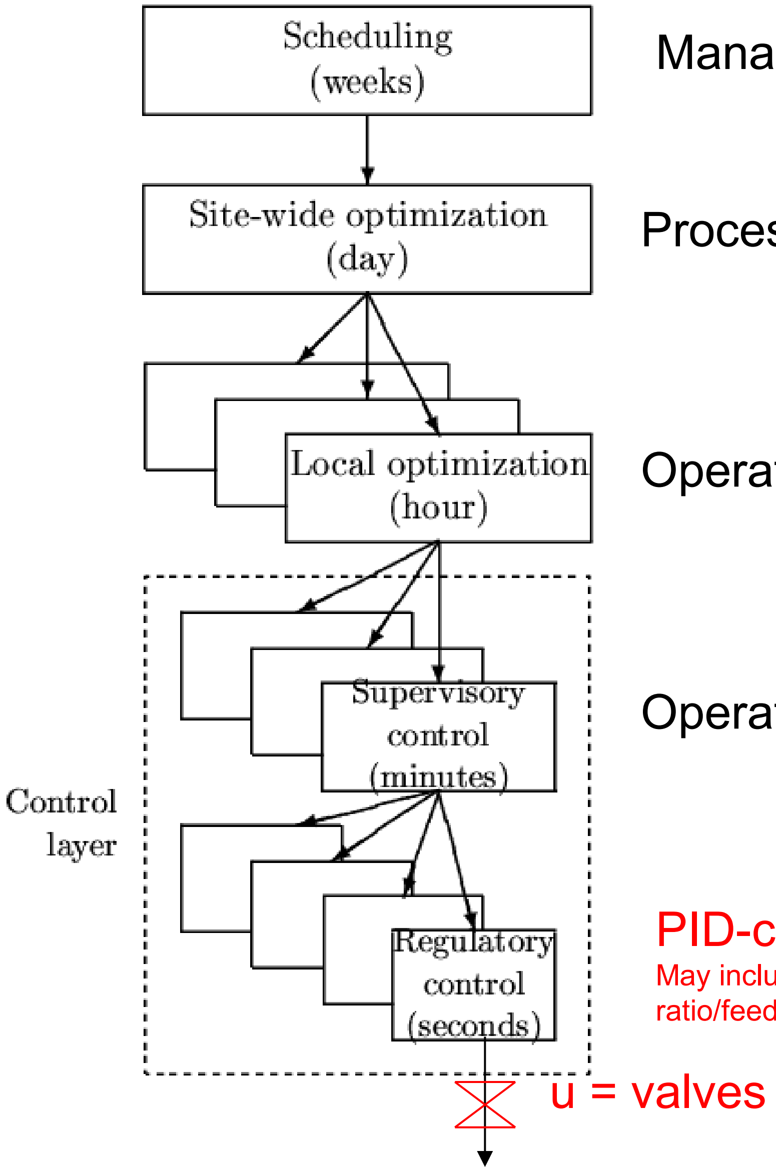
Supervisory control

Decomposition: vertical and horizontal
Constraint switching.

Practical operation: Hierarchical (cascade) structure based on time scale separation

NOTE: Control system is decomposed both
- Hierarhically (in time)
- Horizontally (in space)

Status industry:
- RTO is rarely used.
- MPC is used in the petrochemical and refining industry, but in general it is much less common than was expected when MPC «took off» around 1990
- ARC is common
- Manual control still common…



Manager

Process engineer

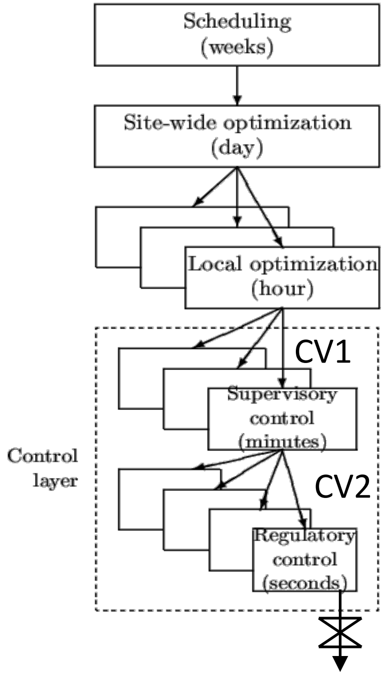
Operator/RTO

Operator/Advanced regulatory control (ARC)/MPC

PID-control
May include some ratio/feedforward and cascade control

Two fundamental ways of decomposing the controller

- Vertical (hierarchical; cascade)
- Based on time scale separation
- Decision: Selection of CVs that connect layers



- Horizontal (decentralized)
- Usually based on distance
- Decision: Pairing of MVs and CVs within layers

CV = controlled variable
MV = manipulated variable

Combine control and optimization into one layer?

EMPC: Economic model predictive "control"'

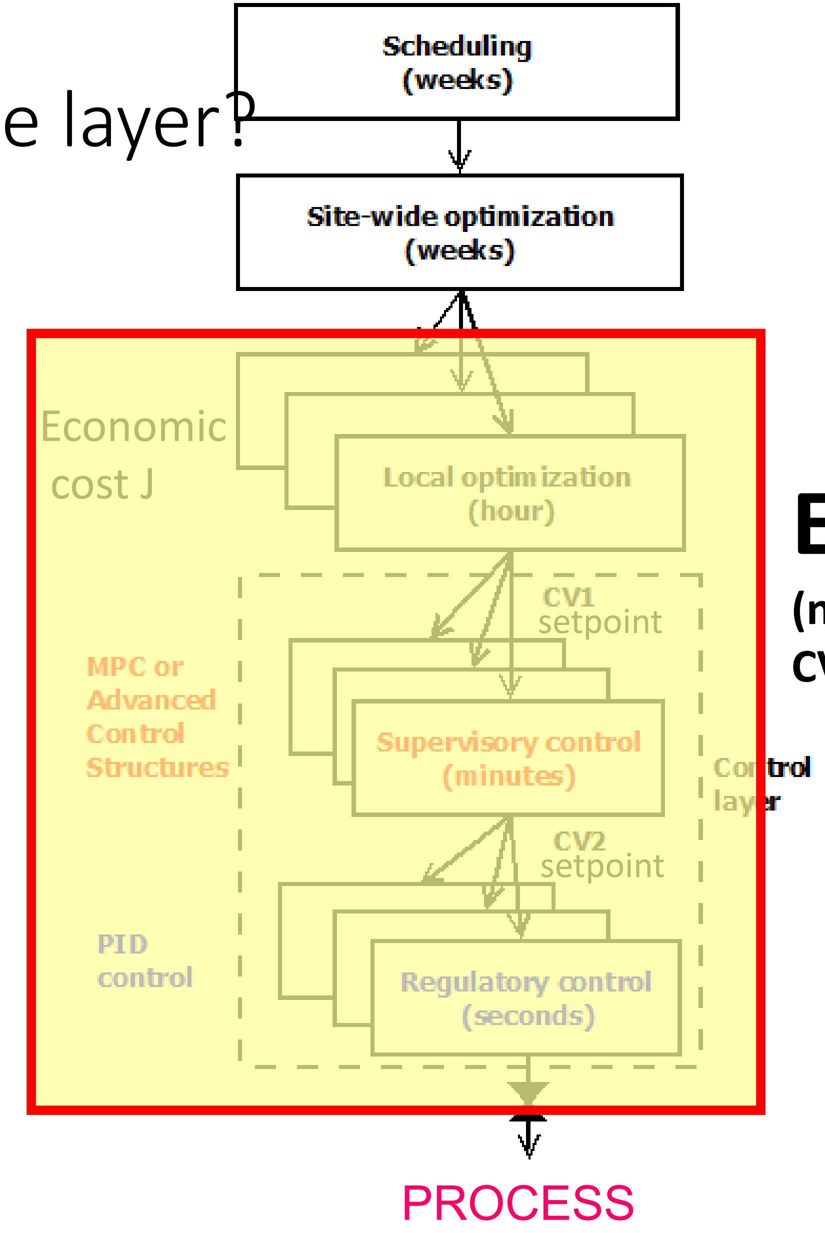
$J_{EMPC} = J + J_{control}$

Penalize input usage, $J_{control} = \Sigma \Delta u_i^2$

NO, combining layers is generally not a good idea!
(the good idea is to separate them!)

One layer (EMPC) is optimal theoreretically, but

- Need detailed dynamic model of everything
- Tuning difficult and indirect
- Slow! (or at least difficult to speed up parts of the control)
- Robustness poor
- Implementation and maintainance costly and time consuming



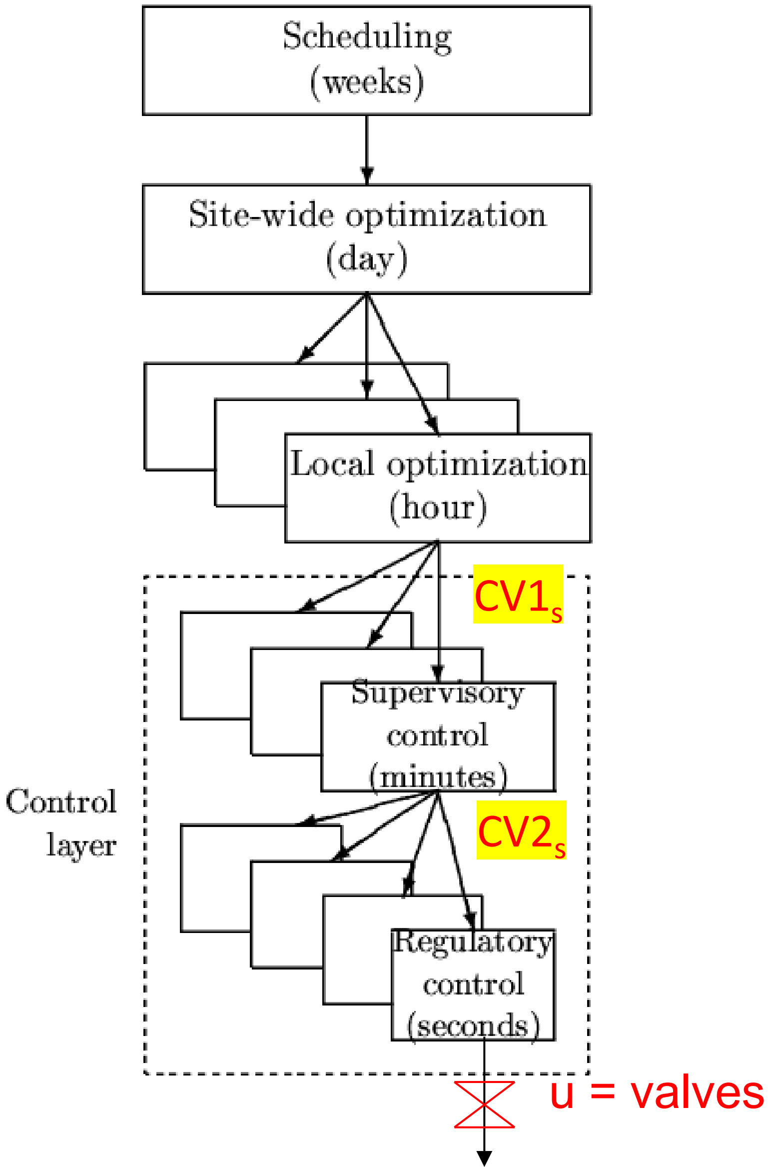
EMPC

(no setpoints, CV1, CV2)

Typical economic cost function:

J [$/s] = cost feed + cost energy – value products

What is the difference between optimization and control?



My definition:

Optimization:

- Minimizes economic cost

Control:

- Follow setpoints y_s

Cost functions in layers

RTO: Minimize economic cost

$J_\$$ = cost feed + cost energy – value products

Setpoint control

$J_{c1} = Q(y_1-y_{1s})^2 + R\Delta u_1^2$ (MPC)

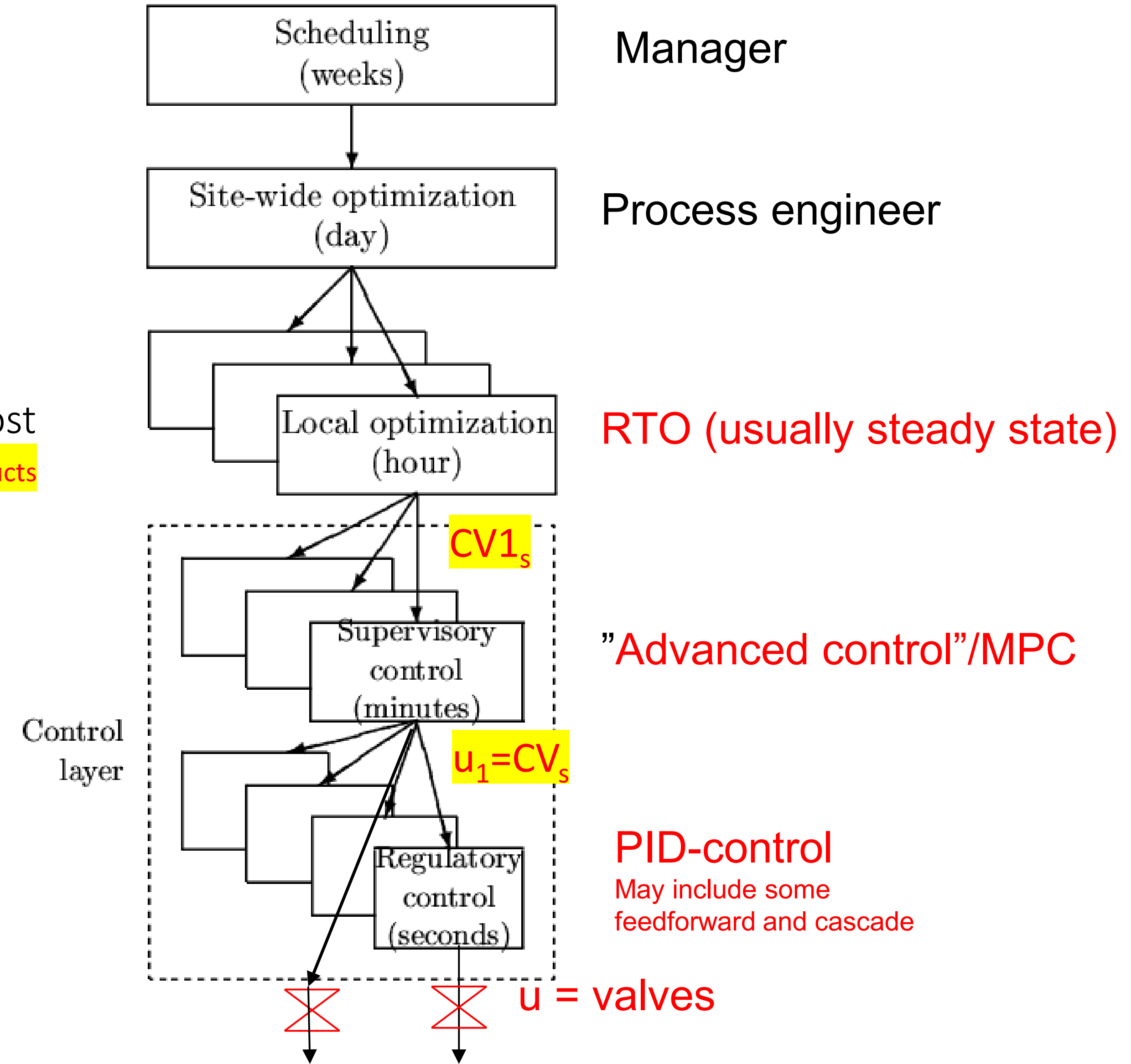
(+ look after other variables, Avoid constraints)

PID: Stabilize + avoid drift

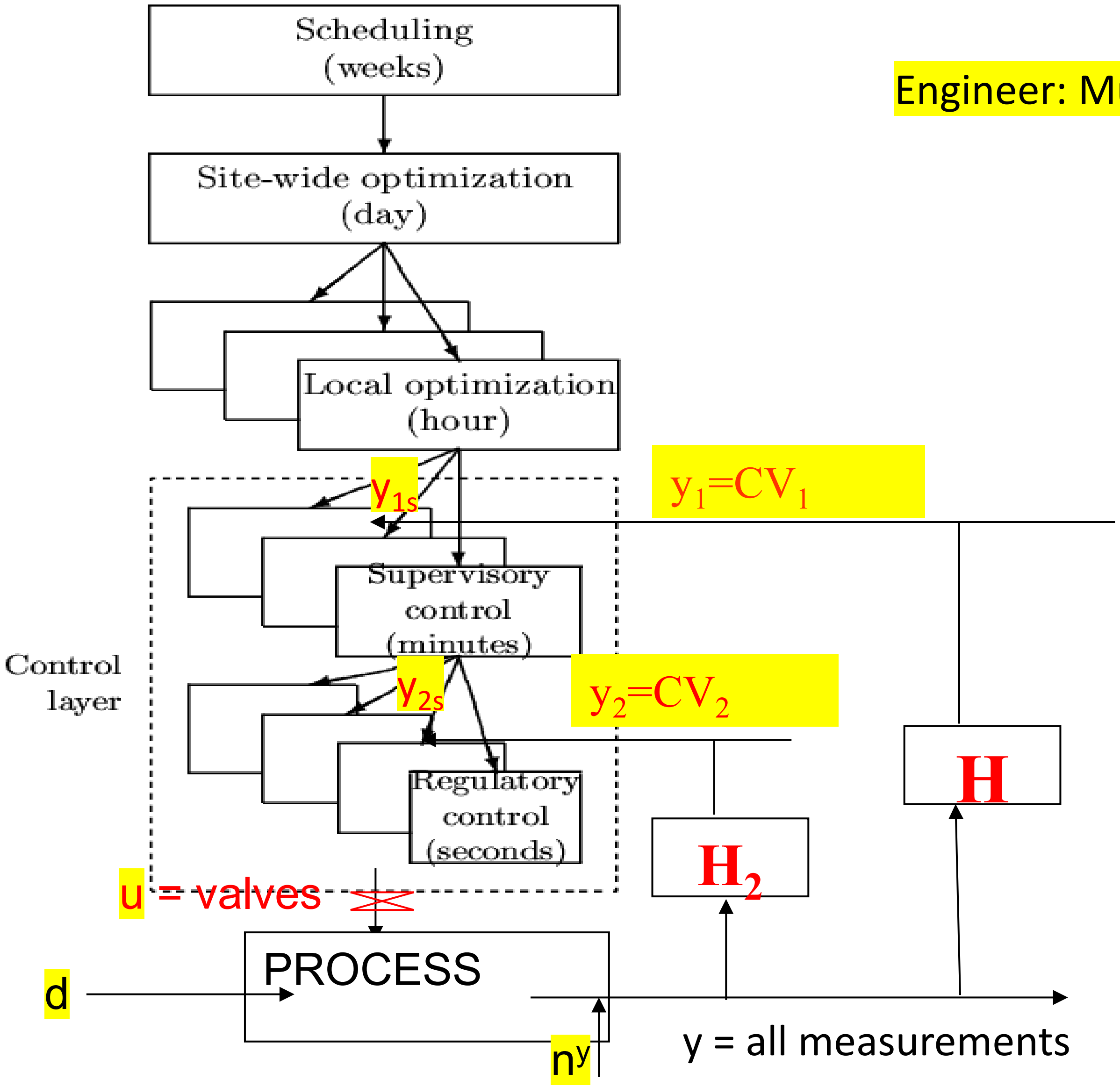
$J_{c2} = Q(y_2-y_{2s})^2 + R\Delta u^2$

+ look at Gain margin…

(or just use SIMC-rules!)



Use of data in layers (feedback)



Engineer: Must choose what to control (H and H_2)

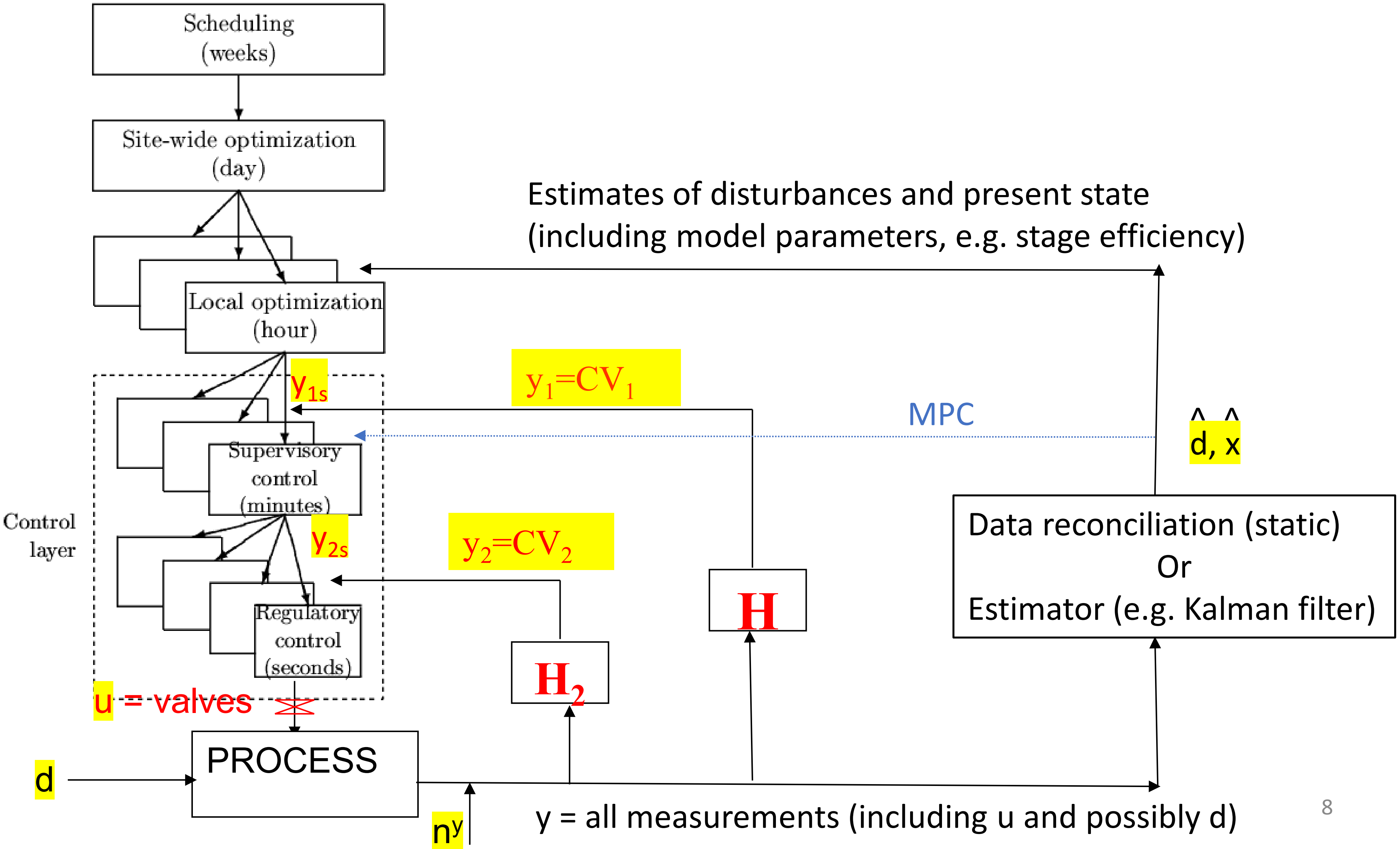
- H and H_2 are usually selection matrices

Typically:

- y_1=Hy = active constraints + «self-optimizing» variables
- $y_2=H_2y$ = drifting variables (levels, pressures, temperatures)

Use of data

Optimization layer: Needs model parameters and disturbances

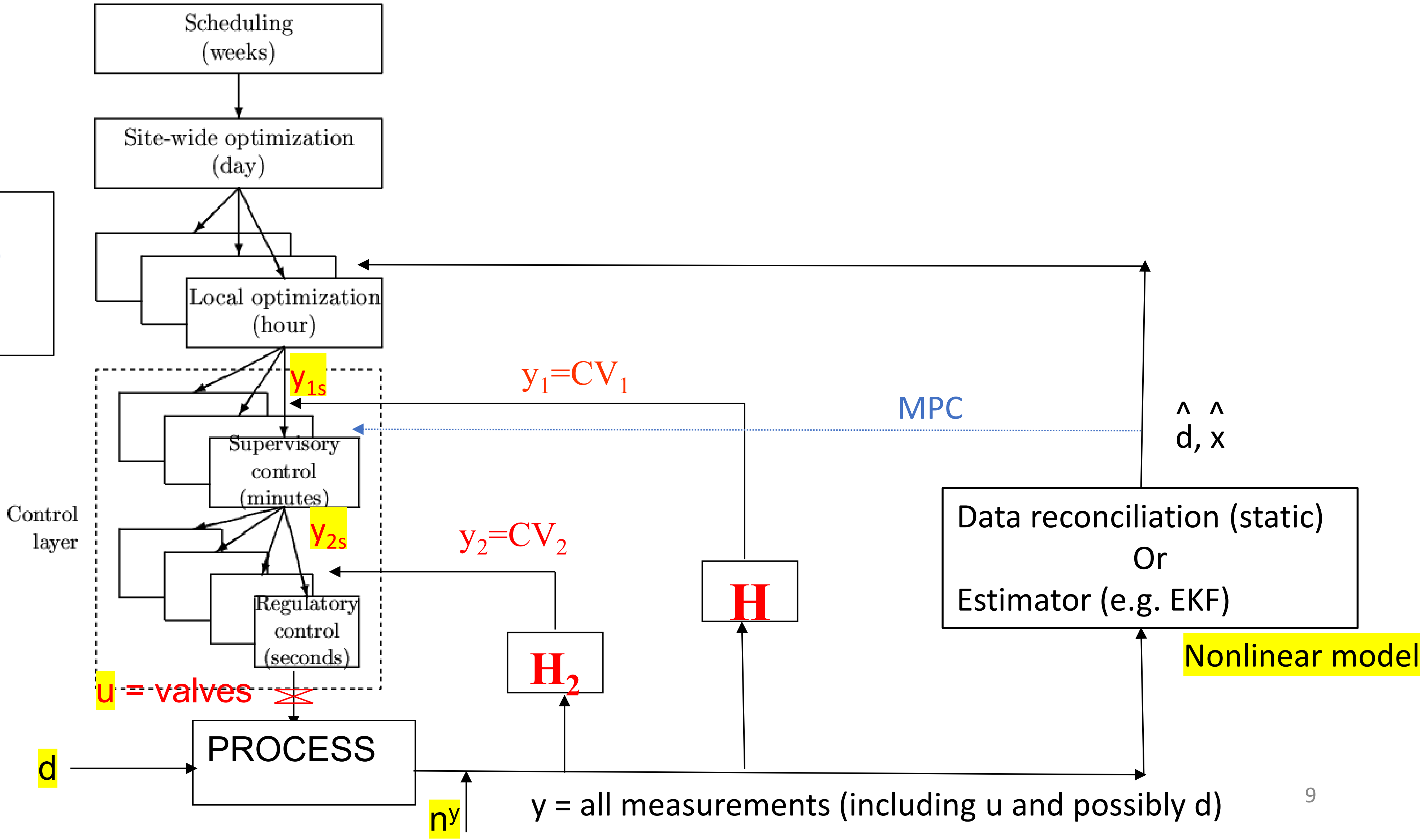


Use of models

RTO layer:
- Nonlinear model of whole process
- usually physical and static

MPC layer:
- Multivariable dynamic linear model for each unit
- usually from data

PID-layer:
- Dynamic linear model for each loop
- usually from data.
- May use physical model for linearization, decoupling and feedforward
  - see: input transformation



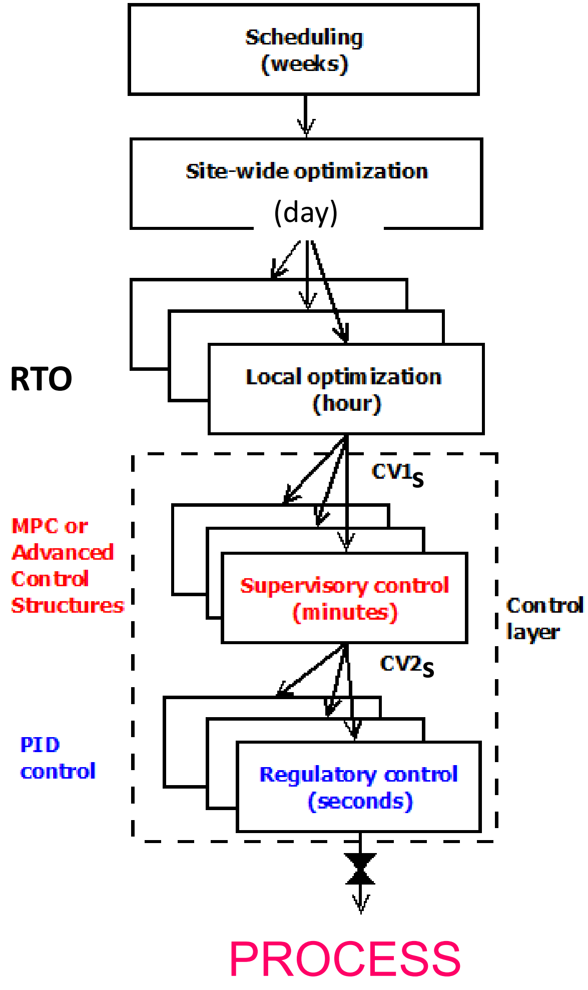
Is there a problem with model consistency between layers?

Quote from a recent paper I reviewed

- "One of the difficulties in practical implementations of classic Real-Time Optimization (RTO) strategy is the integration between optimization (RTO) and control layers (MPC), mainly due to the differences between the models used in each layer, which may result in unreachable setpoints coming from optimization to the control layer. In this context, Economic Model Predictive Control (EMPC) is a strategy where optimization and control problems are solved simultaneously."

- Is this likely to happen?

- **No, This is a myth** and no reason for choosing EMPC
- Truth: With integral action in the control layer (MPC), the process will go to the setpoints (y_{1s}=$CV1_s$) desired by the RTO layer, irrespective of any model error in the MPC layer
  - $J_{MPC} = Q(y_1 - y_{1s})^2 + R\Delta u_1^2$
- Of course, the setpoints from the RTO layer must correspond to a feasible steady state, but the model in the MPC layer does not affect this
- Of course, there may be economic losses dynamically, for example, dynamic constraints may mean it takes some time to reach the setpoints



Main objectives operation

1. **Economics**: Implementation of acceptable (near-optimal) operation

2. **Regulation:** Stable operation around given setpoint

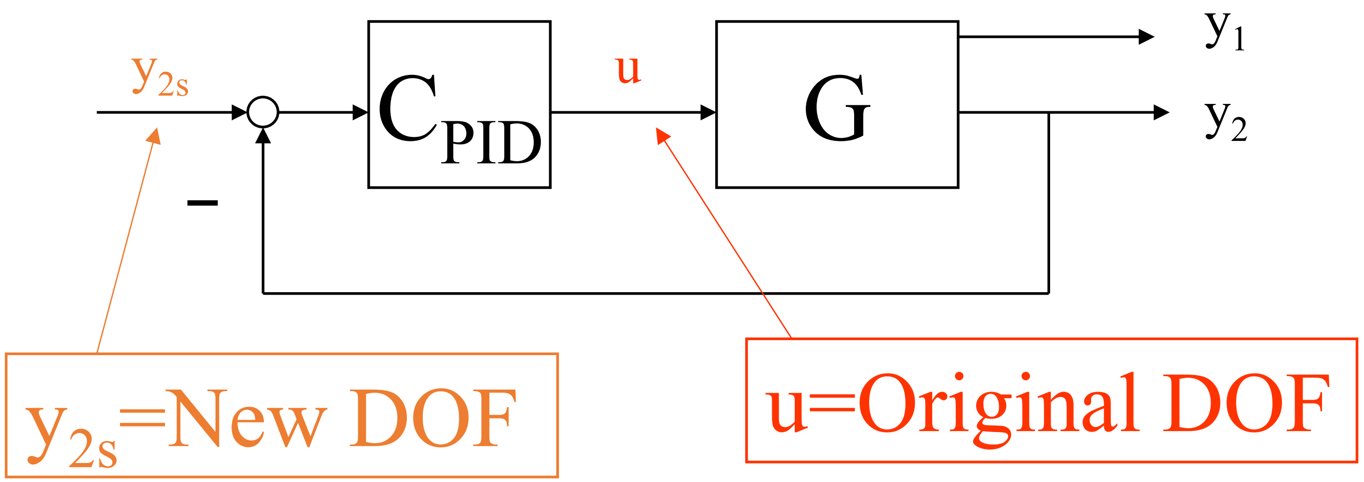
ARE THESE OBJECTIVES CONFLICTING?
IS THERE ANY LOSS IN ECONOMICS?

- Usually NOT
  - Different time scales
    - Stabilization fast time scale
  - Stabilization doesn't "use up" any degrees of freedom
    - Reference value (setpoint) available for layer above
    - But it "uses up" part of the time window

Hierarchical structure: Degrees of freedom unchanged

- No degrees of freedom lost as setpoints y_{2s} replace inputs u as new degrees of freedom for control of y_1

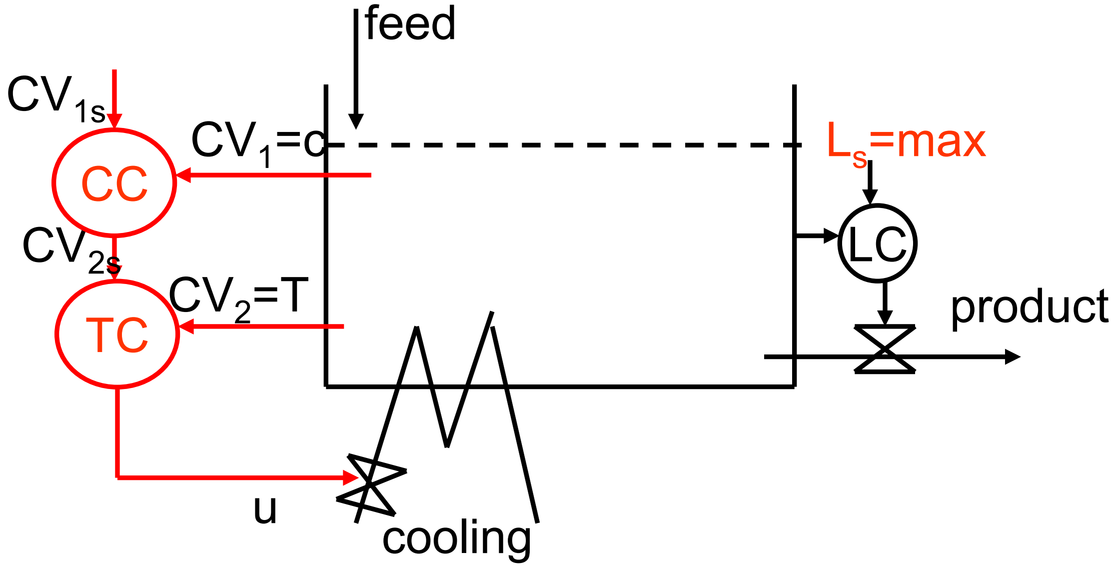
Cascade control:



Example: Exothermic reactor (unstable)

Want to control y_1=CV1=composition (+ level)

- u = cooling flow (F)
- CV_1 = composition (c)
- CV_2 = temperature (T)



Systematic procedure for economic process control

Start "top-down" with economics (steady state):

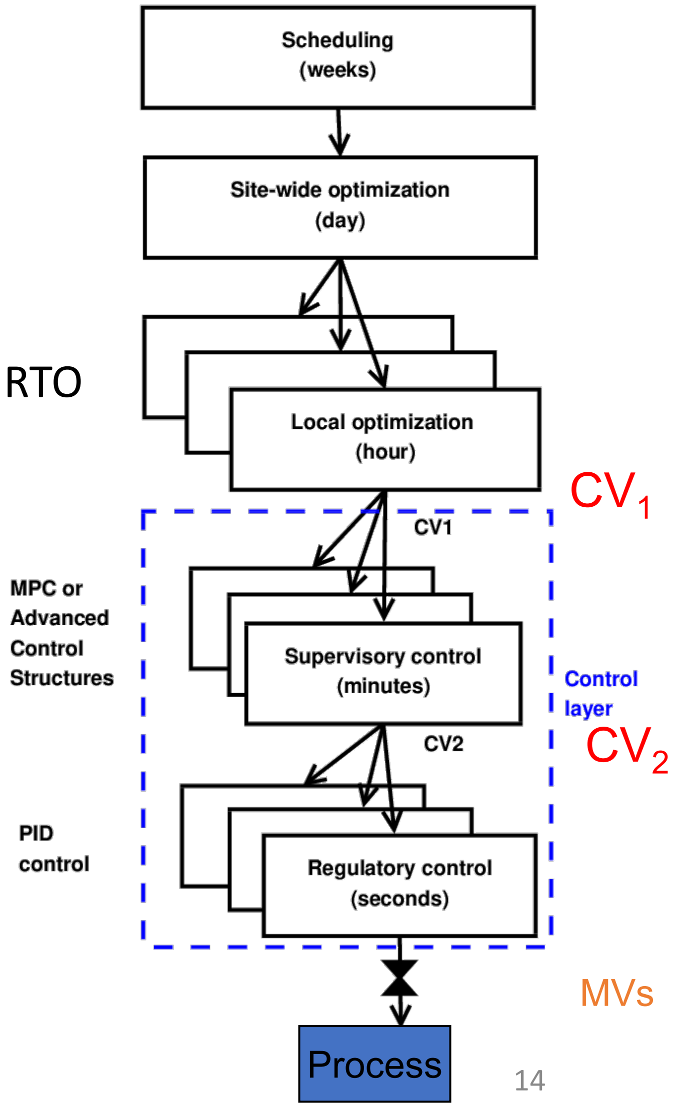
- Step 1: Define operational objectives (J) and constraints
- Step 2: Optimize steady-state operation
- Step 3: Decide what to control (CVs)
  - Step 3A: Identify active constraints = primary CV1.
  - Step 3B: Remaining unconstrained DOFs: Self-optimizing CV1 (find H)
- Step 4: Where do we set the throughput? TPM location

Then bottom-up design of control system (dynamics):

- Step 5: Regulatory control
  - Control variables to stop "drift" (sensitive temperatures, pressures, ...)
  - Inventory control radiating around TPM

Finally: Make link between "top-down" and "bottom up"

- Step 6: "Advanced/supervisory control"
  - Control economic CVs: Active constraints and self-optimizing variables
  - Look after variables in regulatory layer below (e.g., avoid saturation)
- **Step 7: Real-time optimization** (Do we need it?)



S. Skogestad, ``Control structure design for complete chemical plants'', *Computers and Chemical Engineering*, **28** (1-2), 219-234 (2004).

Step 3: Sigurd's rules for CV selection

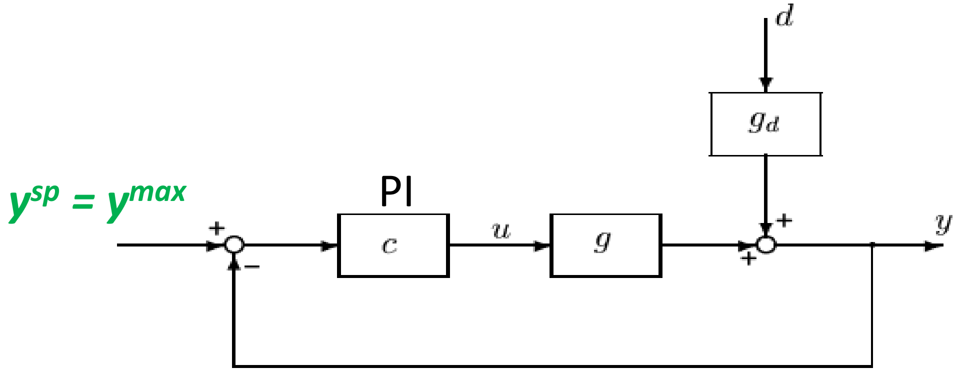
1. Always control active constraints! (almost always)
2. Purity constraint on expensive product always active (no overpurification):
   (a) "Avoid product give away" (e.g., sell water as expensive product)
   (b) Save energy (costs energy to overpurify)

Unconstrained optimum:

3. Look for "self-optimizing" variables. They should
   - Be sensitive to the MV
   - have close-to-constant optimal value
4. **NEVER try to control a variable that reaches max or min at the optimum**
   - In particular, never try to control directly the cost J
   - Assume we want to minimize J (e.g., J = V = energy) - and we make the stupid choice os selecting CV = V = J
     - Then setting J < Jmin: Gives infeasible operation (cannot meet constraints)
     - and setting J > Jmin: Forces us to be nonoptimal (which may require strange operation)

Cruise control: Optimization with PI-controller

$$\max y$$
$$\text{s.t.} \quad y \le y^{max}$$
$$u \le u^{max}$$



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

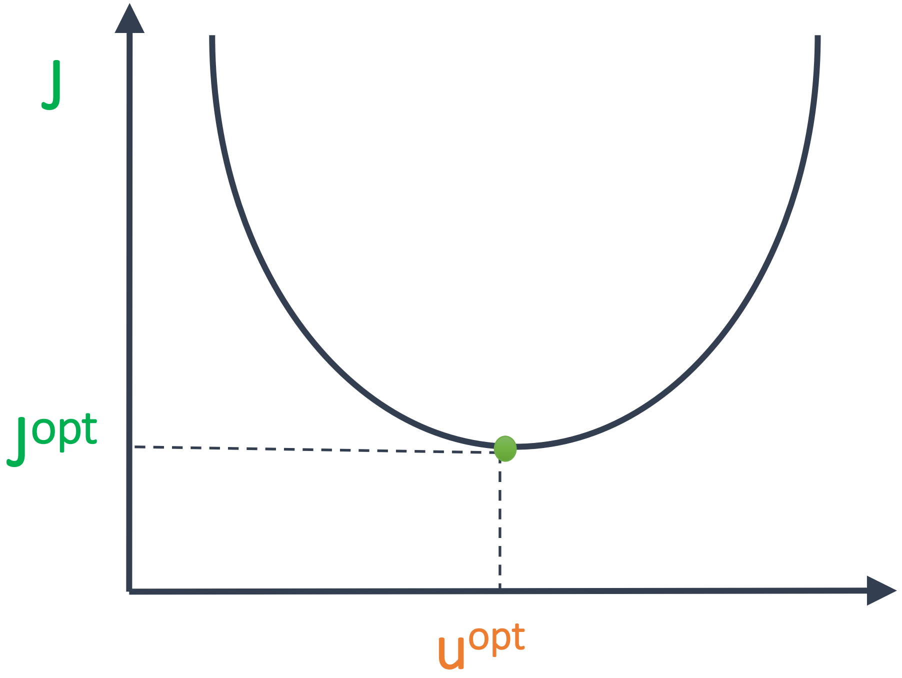
- **Optimal solution has two active constraint regions:**
  1. $y = y^{max}$ → **speed limit**
  2. $u = u^{max}$ → **max power**
- Note: Positive gain from MV (u) to CV (y)
- **Solved with PI-controller**
  - $y^{sp} = y^{max}$
  - Anti-windup: I-action is off when $u=u^{max}$

s.t. = subject to
y = CV = controlled variable

NTNU

The less obvious case: Unconstrained optimum

- u = unconstrained MV
- What to control? y=CV=?



Analytical solution:
$H = G^{yT}(YY^T)^{-1}$ where $Y = [FW_d \quad W_{n^y}]$

Step 4: Inventory control and TPM (later!)

Step 5: Design of regulatory control layer

Usually single-loop PID controllers

Choice of CVs (CV2):

- **CV2 = «drifting variables»**
  - Levels, pressures
  - Some temperatures
- CV2 may also include economic variables (CV1) that we want to control on a fast time scale
  - Hard constraints

Choice of MV2s and pairings (MV2-CV2):

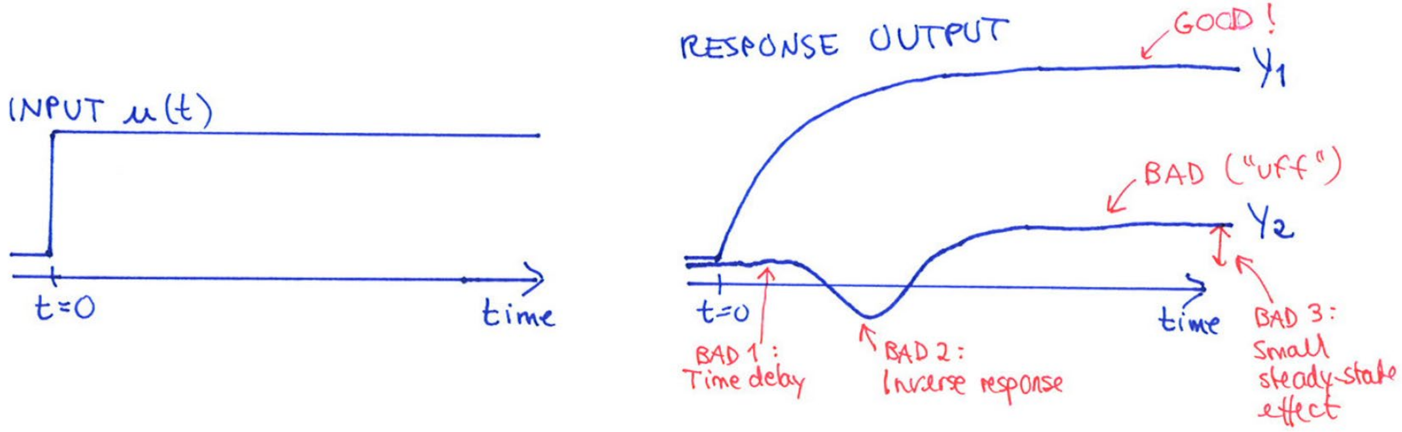
CV2
↕
MV2

1. **Main rule: "Pair close**". Want:
   - Large gain
   - Small delay
   - Small time constant
2. **Avoid MVs that may saturate in regulatory layer**
   - Otherwise, will need logic for re-pairing (MV-CV switching)
   - The exception is if you follow the ***Input saturation rule: "Pair a MV that may saturate with a CV that can be given up (when the MV saturates) "***
3. **Avoid pairing on negative steady-state RGA-elements**
   - It's possible, but then you must be sure that the loops are always working (no manual contriol or MV-saturation)

- May include cascade loops (flow control!) and some feedforward, decoupling, linearization

•

Main rule: “Pair close”



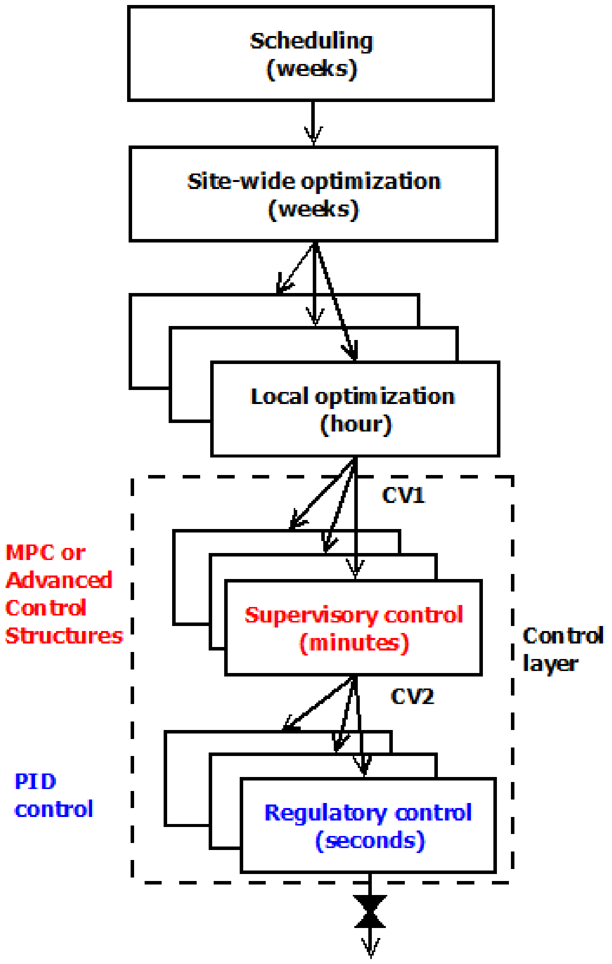
The response (from input to output) should be fast, large and in one direction.

Avoid time delay and inverse responses!

Objectives of regulatory control layer

1. Allow for manual operation
2. Simple decentralized (local) PID controllers that can be tuned on-line
3. Take care of "fast" control
4. Track setpoint changes from the layer above
5. Local disturbance rejection
6. Stabilization (mathematical sense)
7. Avoid "drift" (due to disturbances) so system stays in "linear region"
   - "stabilization" (practical sense)
8. Allow for "slow" control in layer above (supervisory control)
9. Make control problem easy as seen from layer above
10. Use "easy" and "robust" measurements (pressure, temperature)
11. Simple structure
12. Contribute to overall economic objective ("indirect" control)
13. Should not need to be changed during operation

Step 6: Design of Supervisory layer



Objectives 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 supervisory control layer:

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

QUIZ
What are the three most important inventions of process control?

- Hint 1: According to Sigurd Skogestad
- Hint 2: All were in use around 1940

SOLUTION

1. PID controller, in particular, I-action
2. Cascade control
3. Ratio control

The three main inventions of process control can only indirectly and with effort be implemented with MPC

1. Integral action with MPC: Need to add artificial integrating disturbance in estimator
   - ARC: Just add an integrator in the controller (use PID)

2. Cascade control with MPC: Need model for how u and d affect y_1 and y_2.
   - ARC: Just need to know that control of y_2 indirectly improves control of y_1

3. Ratio control with MPC: Need model for how u and d affect property y
   - ARC: Just need the insight that it is good for control of y to keep the ratio R=u/d constant

Because of this, MPC should be on top of a regulatory control layer with the setpoints for y_2 and R as MVs.

ARC: Standard Advanced control elements

Each element links a subset of inputs with a subset of outputs. Results in simple local design and tuning

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

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

Next, there are some control elements used for cases when we reach constraints:

E4*. Selective (limit, override) control (for output switching)
E5*. Split range control (for input switching)
E6*. Separate controllers (with different setpoints) as an alternative to split range control (E5)
E7*. VPC as an alternative to split range control (E5)

All the above seven elements have feedback control as a main feature and are usually based on PID controllers. Ratio control seems to be an exception, but the desired ratio setpoint is usually set by an outer feedback controller. There are also several features that may be added to the standard PID controller, including

E8*. Anti-windup scheme for the integral mode
E9*. Two-degrees of freedom features (e.g., no derivative action on setpoint, setpoint filter)
E10. Gain scheduling (Controller tunings change as a given function of the scheduling variable, e.g., a disturbance, process input, process output, setpoint or control error)

In addition, the following more general model-based elements are in common use:

E11*. Feedforward control
E12*. Decoupling elements (usually designed using feedforward thinking)
E13. Linearization elements
E14*. Calculation blocks (including nonlinear feedforward and decoupling)
E15. Simple static estimators (also known as inferential elements or soft sensors)

Finally, there are a number of simpler standard elements that may be used independently or as part of other elements, such as

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

[2] The control elements with an asterisk * are discussed in more detail in this paper.

Sigurd Skogestad, "Advanced control using decomposition and simple elements". Annual Reviews in Control, vol. 56 (2023), Article 100903 (44 pages).

JULY 2024

Annual Reviews in Control 56 (2023) 100903

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

ELSEVIER

Review article

Advanced control using decomposition and simple elements

Sigurd Skogestad

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

Check for updates

ARTICLE INFO

Keywords:
Control structure design
Feedforward control
Cascade control
PID control
Selective control
Override control
Time scale separation
Decentralized control
Distributed control
Horizontal decomposition
Hierarchical decomposition
Layered decomposition
Vertical decomposition
Network architectures

ABSTRACT

The paper explores the standard advanced control elements commonly used in industry for designing advance control systems. These elements include cascade, ratio, feedforward, decoupling, selectors, split range, an more, collectively referred to as "advanced regulatory control" (ARC). Numerous examples are provided, wi a particular focus on process control. The paper emphasizes the shortcomings of model-based optimizatio methods, such as model predictive control (MPC), and challenges the view that MPC can solve all contr problems, while ARC solutions are outdated, ad-hoc and difficult to understand. On the contrary, decomposin the control systems into simple ARC elements is very powerful and allows for designing control systems fo complex processes with only limited information. With the knowledge of the control elements presented i the paper, readers should be able to understand most industrial ARC solutions and propose alternatives an improvements. Furthermore, the paper calls for the academic community to enhance the teaching of AR methods and prioritize research efforts in developing theory and improving design method.

Contents

100%

8.2. *The harder problem: Control structure synthesis*

As a third approach, machine learning may prove to be useful. Machine learning has one of its main strength in pattern recognition, in a similar way to how the human brain works. I have observed over the years that some students, with only two weeks of example-based teaching, are able to suggest good process control solutions with feedback, cascade, and feedforward/ratio control for realistic problems, based on only a flowsheet and some fairly general statements about the control objectives. This is the basis for believing that machine learning (e.g., a tool similar to ChatGPT) may provide a good initial control structure, which may later be improved, either manually or by optimization. It is important that such a tool has a graphical interface, both for presenting the problem and for proposing and improving solutions.

Economic Plantwide Control of the Ethyl Benzene Process

Rahul Jagtap, Ashok S Pathak, and Nitin Kaistha
Dept. of Chemical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016,Uttar Pradesh, India

DOI 10.1002/aic.13964
Published online December 10, 2012 in Wiley Online Library (wileyonlinelibrary.com).

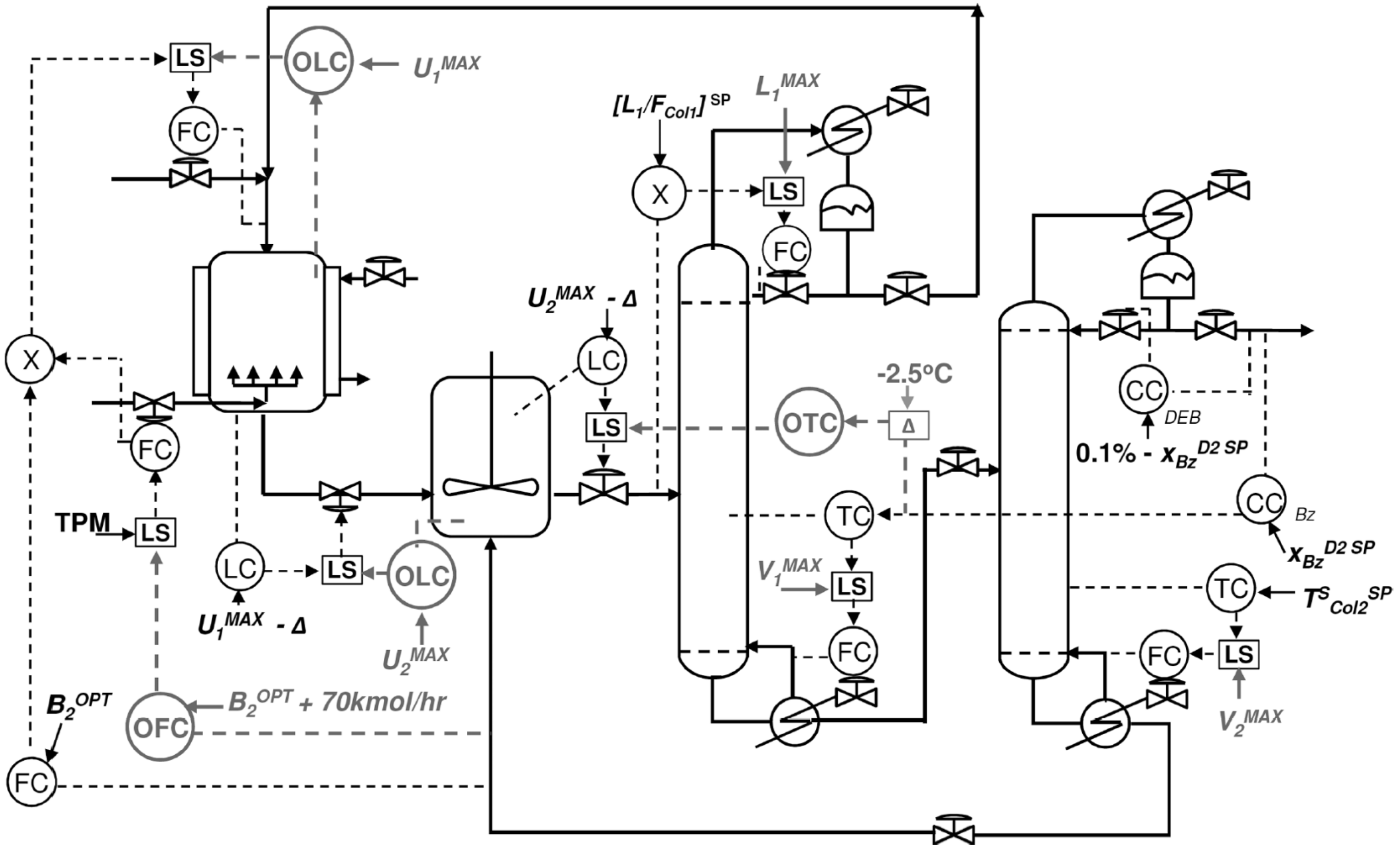
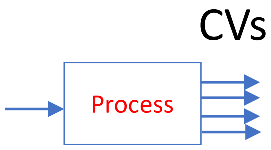


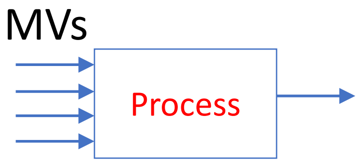
Figure 7. CS2 with overrides for handling equipment capacity constraints.

Constraint switching
(because it is optimal at steady state)

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

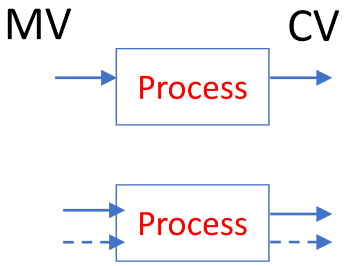


- MV-MV switching
  - Use one MV at a time



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

1. Simple («do nothing»)
2. Complex (repairing of loops)



MV-MV switching

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

Three solutions:

Alt.1 Split-range control (one controller) (E5)

Alt.2 Several controllers with different setpoints (E6)

Alt.3 Valve position control (E7)

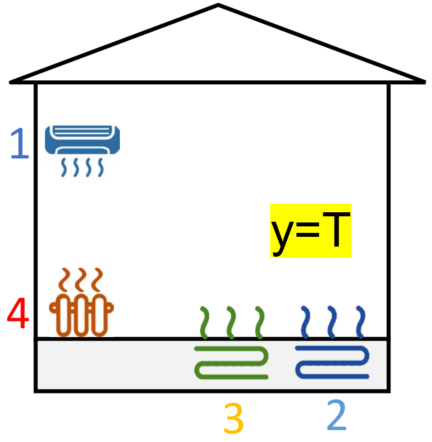
Which is best? It depends on the case!

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

Example MV-MV switching

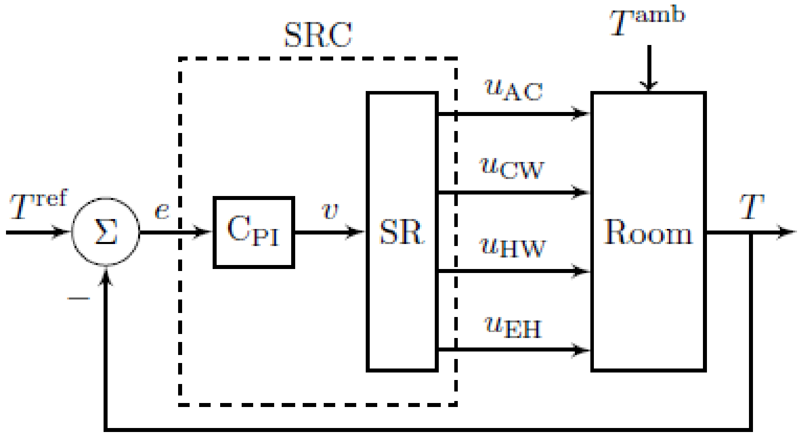
- Break and gas pedal in a car
- Use only one at a time,
- «manual split range control»

Example split range control (E5) : Room temperature with 4 MVs



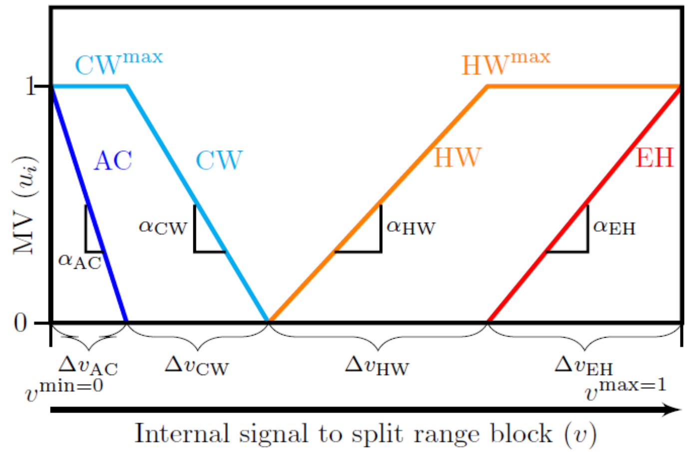
MVs (two for summer and two for winter):

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



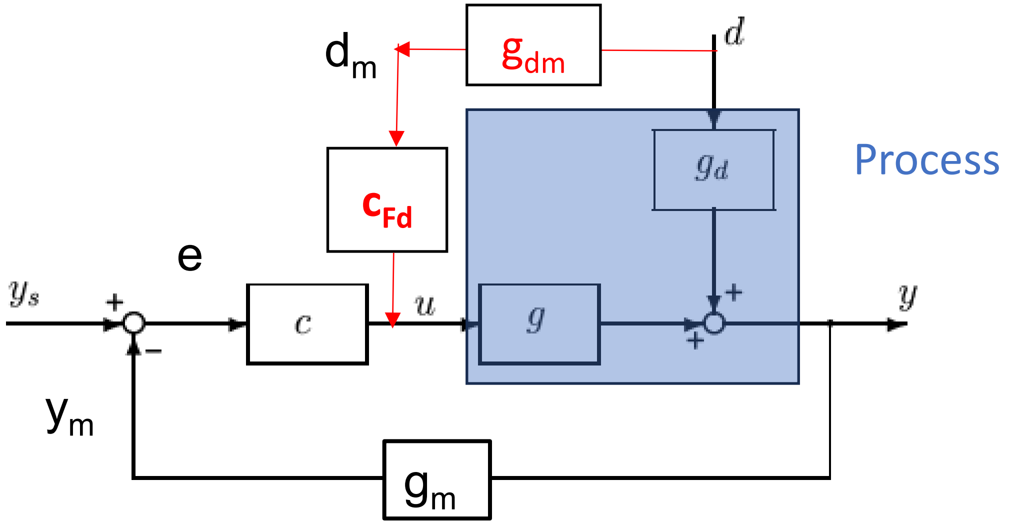
C_{PI} – same controller for all inputs (one integral time)
But get different gains by adjusting slopes α in SR-block

SR-block:



A little on feedforward control

Feedforward control: Measure disturbance (d)



Block diagram of feedforward control

c = Feedback controller

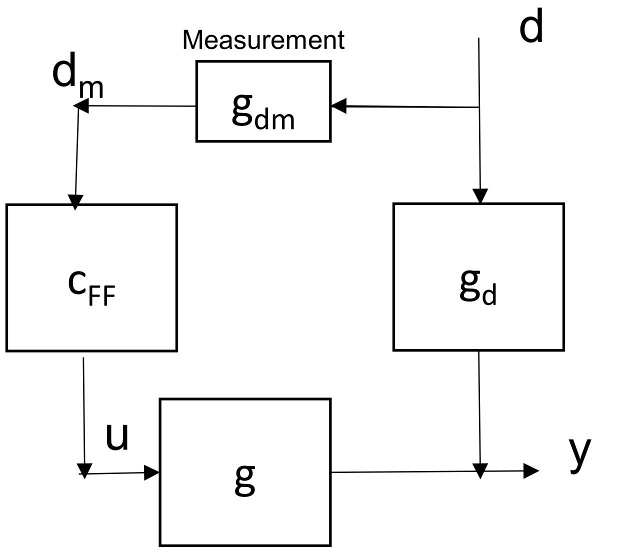
c_{Fd} = Feedforward controller.

Ideal, inverts process g: $c_{Fd} = g^{-1} g_d \, g_{dm}^{-1}$

Usually: Add feedforward when feedback alone is not good enough, for example, because of measurement delay in g_m

Details Feedforward control

- Model: $y = g\, u + g_d\, d$
- Measured disturbance: $d_m = g_{dm}\, d$
- Feedforward controller: $u = c_{FF}\, d_m$
- Get $y = (g\, c_{FF}\, g_{dm} + g_d)\, d$
- Ideal feedforward:
  - $y = 0 \Rightarrow c_{FF,\,ideal} = -g^{-1}\, g_d\, g_{dm}^{-1} = -\frac{g_d}{g_{dm}\, g}$
- In practice: $c_{FF}(s)$ must be realizable
  - Order pole polynomial ≥ order zero polynomial
  - No prediction allowed (θ cannot be negative)
  - Must avoid that c_{FF} has too high gain to avoid (to avoid aggressive input changes)
- Common simplification: $c_{FF} = k$ (static gain)
- General. Approximate $c_{FF,\,ideal}$ as :



$$c_{FF}(s) = k\,\frac{(T_1 s + 1)\,\ldots}{(\tau_1 s + 1)(\tau_2 s + 1)\,\ldots}\,e^{-\theta s}$$

where we must have at least as many τ's as T's

Example feedforward

$$y = gu + g_{d1}d_1 + g_{d2}d_2$$

Feedforward control: $u = c_{FF}d_m$
Ideal feedforward controller: $c_{FF} = -\frac{g_d}{g_{dm}g}$

Example (assume perfect measurements, $g_{dm} = 1$):
$g(s) = \frac{e^{-s}}{s(20s+1)}$
$g_{d1}(s) = \frac{1}{s}$
$g_{d2}(s) = \frac{e^{-s}}{s(20s+1)}$

Disturbance 1:
Ideal: $c_{FF1} = -(20s+1)e^{s}$ (has prediction + has more zeros than poles)
Actual: $c_{FF1} = -1 \cdot \frac{20s+1}{\tau s+1}$ where τ is tuning parameter
(smaller τ gives better control, but requires more input usage).

Comment: In the simulation we use $\tau = 2$ which is quite aggressive; $\tau = 20$ would give $c_{FF1} = -1$.

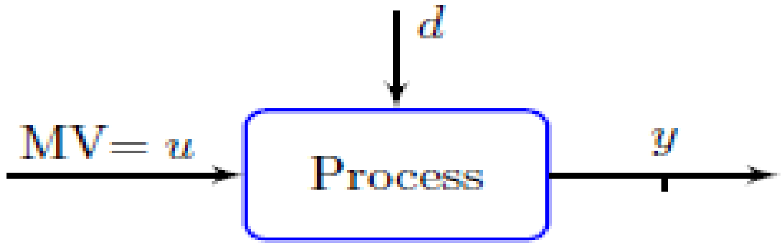
Disturbance 2:
Ideal: $c_{FF2} = -1$
Actual: $c_{FF2} = -1$

«Chicken factor»

Comment: In practice, one often sets the feedforward gain about 80% of the theoretical, that is, $c_{FF2} = -0.8$. This is to avoid that the feedforward controller overreacts, which may confuse the operators. It also makes the feedforward action more robust.

What is best? Feedback or feedforward?

Example: Feedback vs. feedforward for setpoint control of uncertain process



y = G(s) u

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3, \ \tau = 6 \qquad \text{(B.2)}$$

$$\boxed{\text{Desired response :} \quad y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s}$$

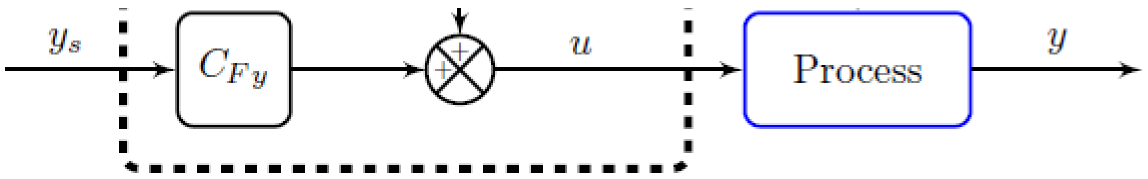


Figure A.42: Block diagram of feedforward control system with linear combination of feedforward from measured disturbance (d) and setpoint (y_s) (E14).

Feedforward solution. We use feedforward from the setpoint (Fig. A.42):

$$u = C_{F_y}(s) y_s$$

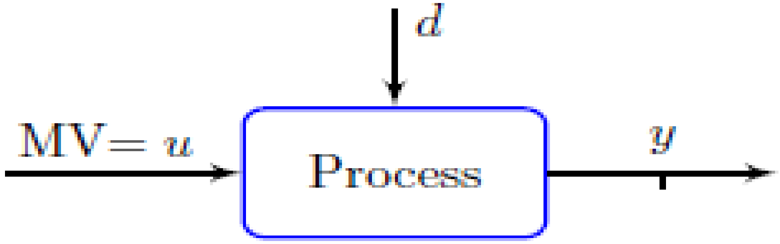
where we choose

$$C_{F_y}(s) = \frac{1}{\tau_c s + 1} G(s)^{-1} = \frac{1}{k} \frac{\tau s + 1}{\tau_c s + 1} = \frac{1}{3} \frac{6s + 1}{4s + 1} \qquad \text{(B.3)}$$

The output response becomes as desired,

$$y = \frac{1}{4s + 1} y_s \qquad \text{(B.4)}$$

Example: Feedback vs. feedforward for setpoint control of uncertain process



y = G(s) u

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3, \ \tau = 6 \qquad \text{(B.2)}$$

$$\boxed{\text{Desired response :} \quad y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s}$$

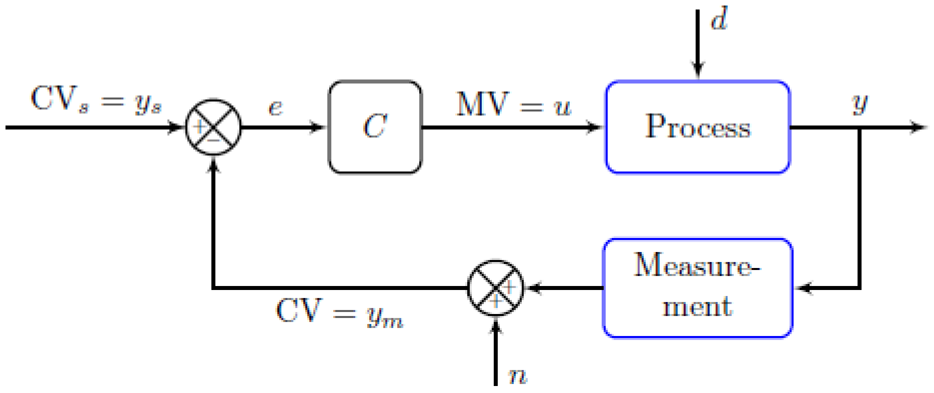


Figure 3: Block diagram of common "one degree-of-freedom" negative feedback control system.

Feedback solution. We use a one degree-of-freedom feedback controller (Fig. 3) acting on the error signal $e = y_s - y$:

$$u = C(s)(y_s - y)$$

We choose a PI-controller with $K_c = 0.5$ and $\tau_I = \tau = 6$ (using the SIMC PI-rule with $\tau_c = 4$, see Appendix C.2):

$$C(s) = K_c\left(1 + \frac{1}{\tau_I s}\right) = 0.5\frac{6s + 1}{6s} \qquad \text{(B.5)}$$

Note that we have selected $\tau_I = \tau = 6$, which implies that the zero dynamics in the PI-controller C, cancel the pole dynamics of the process G. The closed-loop response becomes as desired:

$$y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s \qquad \text{(B.6)}$$

Proof. $y = T(s)y_s$ where $T = L/(1 + L)$ and $L = GC = kK_c/(\tau_I s) = 0.25/s$. So $T = \frac{0.25/s}{1+0.25/s} = \frac{1}{4s+1}$.

Thus, we have two fundamentally different solutions that give the same nominal response, both in terms of the process input $u(t)$ (not shown) and the process output $y(t)$ (black solid curve in Fig. B.43).

- But what happens if the process changes?
  - Consider a gain change so that the model is wrong
    - Process gain from k=3 to k'=4.5

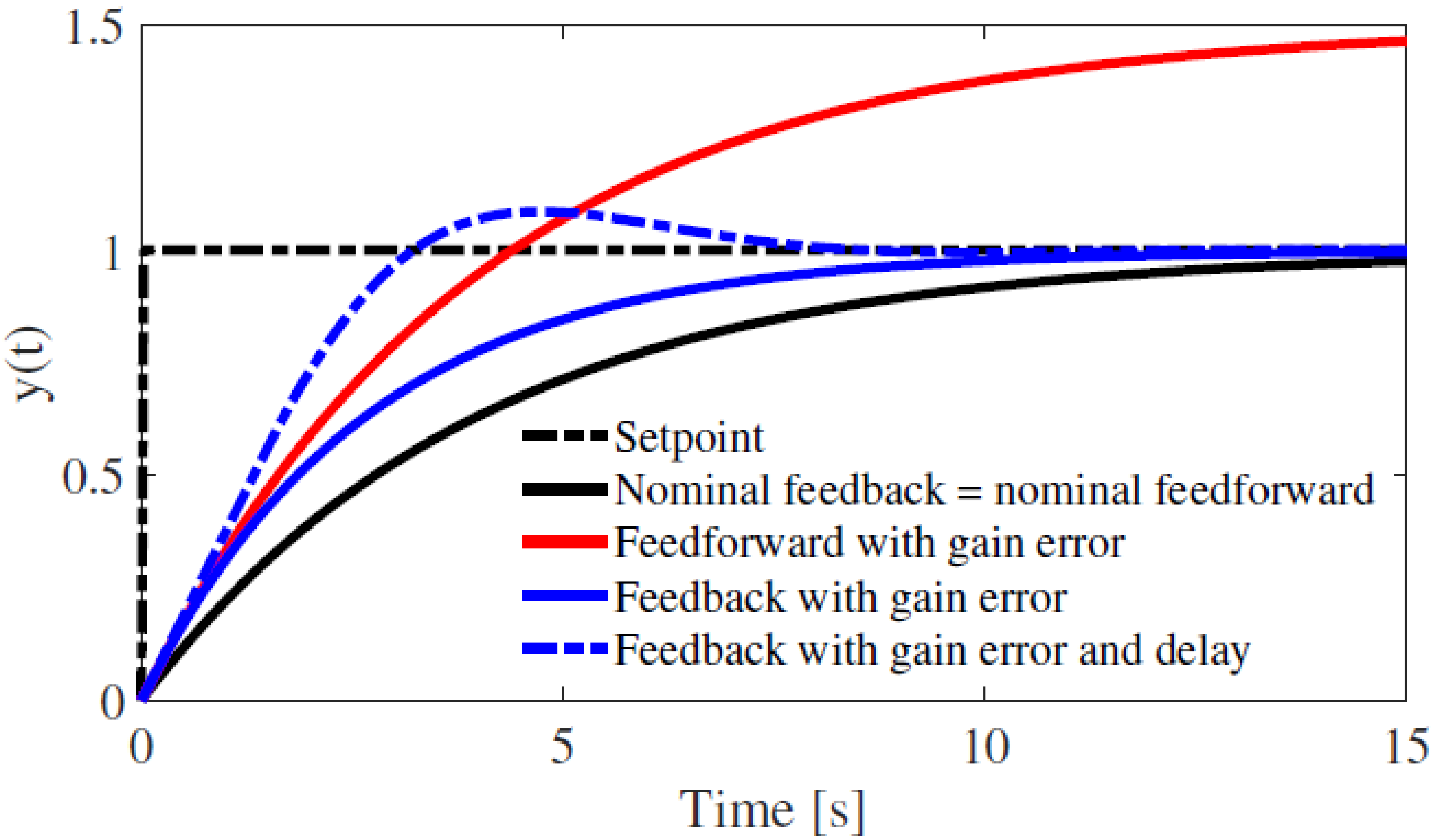


Figure B.43: Setpoint response for process (B.2) demonstrating the advantage of feedback control for handling model error.

Gain error (feedback and feedforward): From k=3 to k'=4.5

Time delay (feedback): From $\theta = 0$ to $\theta = 1.5$

Combine: Two degrees-of-freedom control

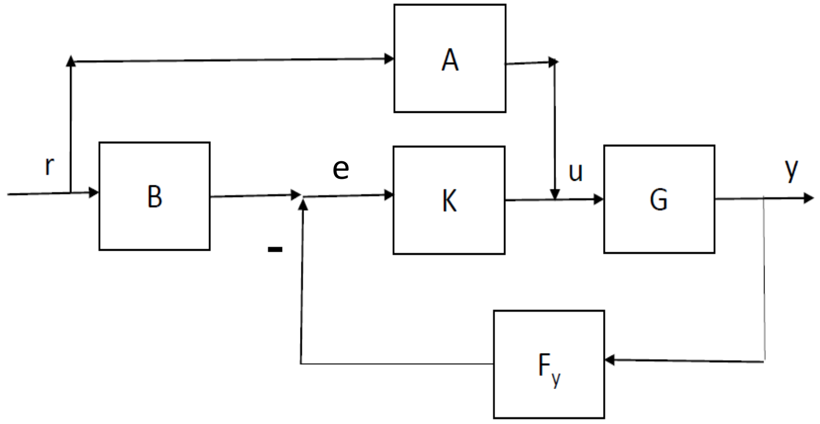


Fig. 3. Two degrees-of-freedom controller with feedforward controller A and prefilter B

- Typically, the feedforward block is $A = G_-^{-1}F_r$ where G_- is the invertible part of G.
  - A typical choice for the prefilter is $F_r = \frac{1}{\tau_r s+1}$
- We want to choose B such that A and K can be designed independently!!
  - Solution (Lang and Ham,1955): Choose $B = F_y GA$ so that transfer function from r to e is zero (with perfect model)!
  - The feedback will then only take action if the feedforward is not working as expected (due to model error).
  - We must have B(0) = I so that we will have no offset (y = r at steady state) even with model error for G
- The feedback controller K can be designed for disturbance rejection and robustness, e.g., using SIMC rules.

Introduction to switching:

Need to control active constraints

But active constraints may change during operation

Four cases:

- A. MV-MV switching
- B. CV-CV switching
- MV-CV switching
  - C. Simple (if we follow input saturation rule*). Example: car to airport
  - D. Complex (combine MV-MV and CV-CV)

**Input saturation rule: "Pair a MV that may saturate with a CV that can be given up (when the MV saturates) "*

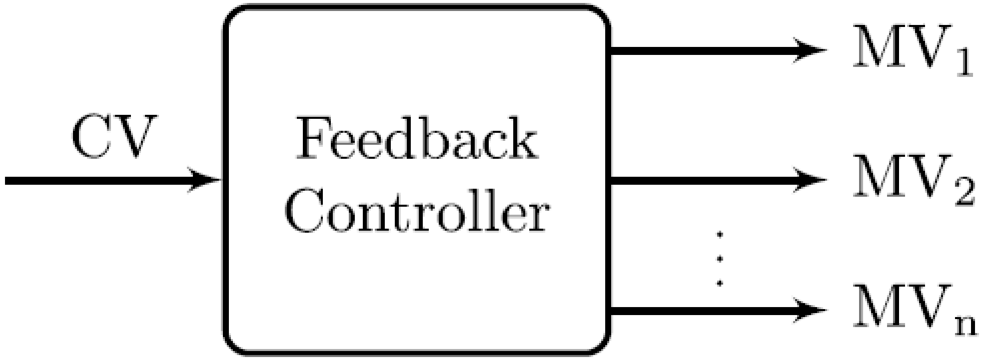


Fig. 5. MV-MV switching is used when we have multiple MVs to control one CV, but only one MV should be used at a time. The block "feedback controller" usually consists of several elements, for example, a controller and a split range block.

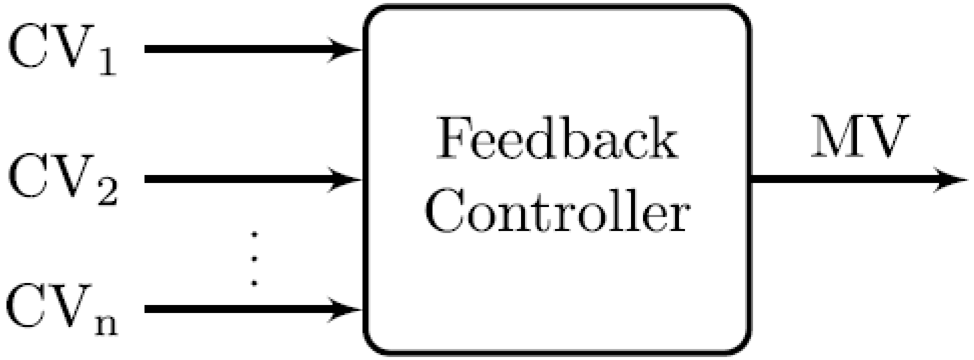


Fig. 6. CV-CV switching is used when we have one MV to control multiple CVs, but the MV should control only one CV at a time. The block "feedback controller" usually consists of several elements, typically several PID-controllers and a selector.

A. MV-MV switching

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

Three main solutions for "selecting the right MV":

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

Alt 1': Generalized SRC (many controllers)

Alt.2 Many controllers with different setpoints

Alt.3 Valve position control

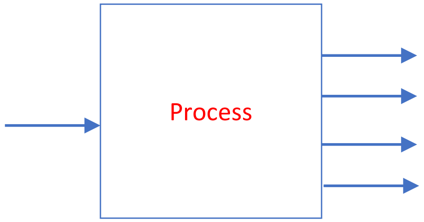
In addition: MPC

Which is best? It depends on the case!

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

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

B. CV-CV switching



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

The four switching cases in more detail

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

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

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

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

Plus the combination: MV-CV switching

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

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

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

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

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

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

Design of selector structure

Rule 1 (max or min selector)

- Use max-selector for constraints that are satisfied with a large input
- Use min-selector for constraints that are satisfied with a small input

Rule 2 (order of max and min selectors):

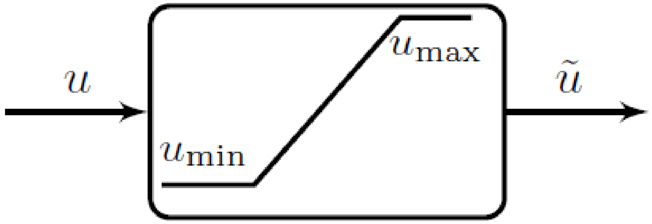
- If need both max and min selector: Potential infeasibility (conflict)
- Order does not matter if problem is feasible
- If infeasible: Put highest priority constraint at the end

"Systematic design of active constraint switching using selectors." Dinesh Krishnamoorthy , Sigurd Skogestad. Computers & Chemical Engineering, Volume 143, (2020)
"Advanced control using decomposition and simple elements". Sigurd Skogestad. Annual Reviews in Control, Volume 56, 100903 (2023)

Valves have “built-in” selectors

Rule 3 (a bit opposite of what you may guess)

- A closed valve (u_{min}=0) gives a “built-in” max-selector (to avoid negative flow)
- An open valve (u_{max}=1) gives a “built-in” min-selector
  - So: Not necessary to add these as selector blocks (but it will not be wrong).
  - The “built-in” selectors are never conflicting because cannot have closed and open at the same time
  - Another way to see this is to note that a valve works as a saturation element

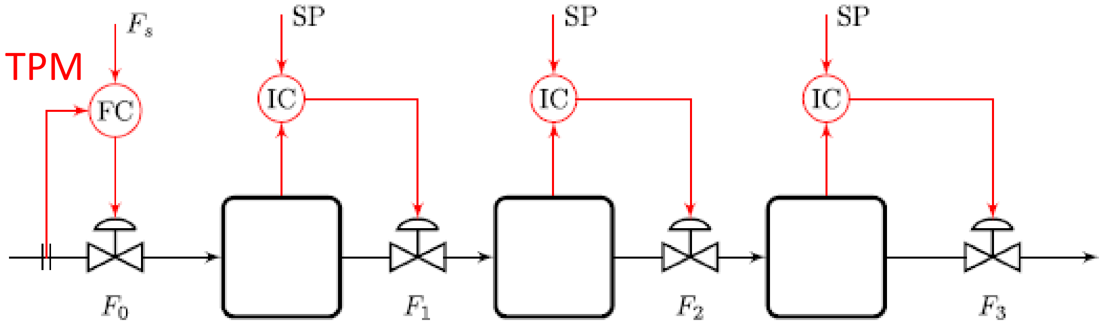


Saturation element may be implemented in three other ways (equivalent because never conflict)

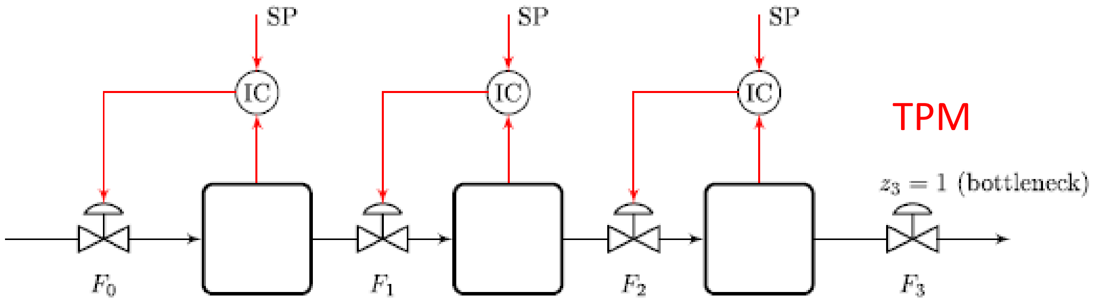
1. Min-selector followed by max-selector
2. Max-selector followed by min-selector
3. Mid-selector

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

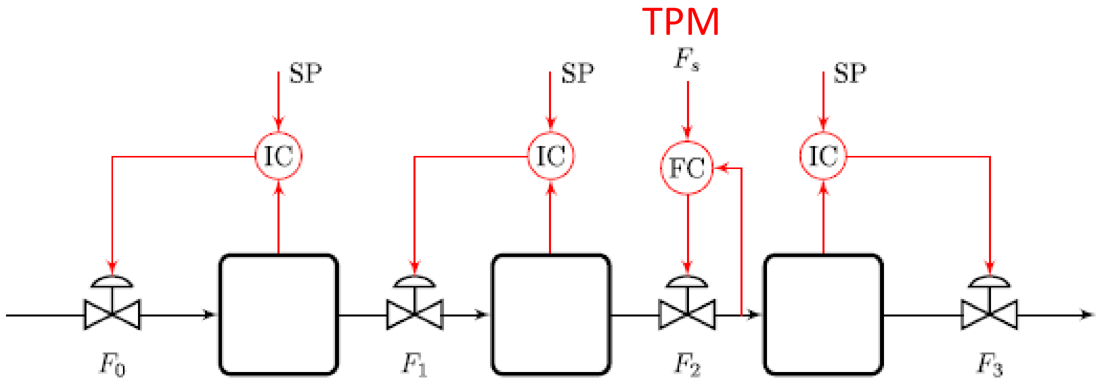
“Advanced control using decomposition and simple elements”. Sigurd Skogestad. Annual Reviews in Control, Volume 56, 100903 (2023)



(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)



(b) Inventory control in opposite direction of flow (for given product flow, TPM= F_3)

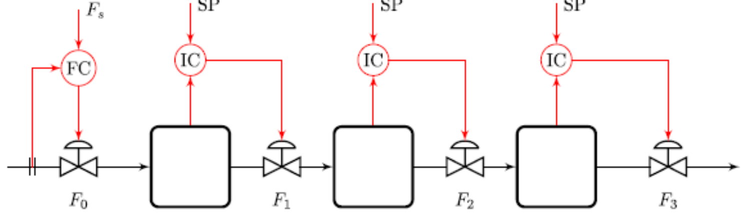


(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)

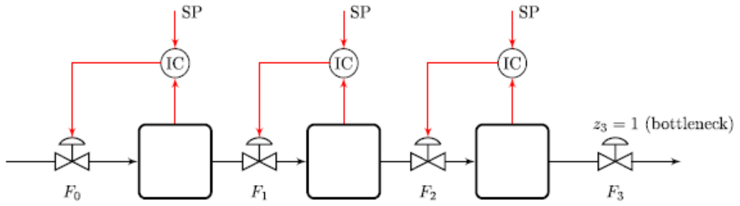
Radiation rule *Inventory control should be "radiating" around a given flow (TPM),*

TPM = Gas Pedal = Variable used for setting the throughput/production rate (for the entire process).

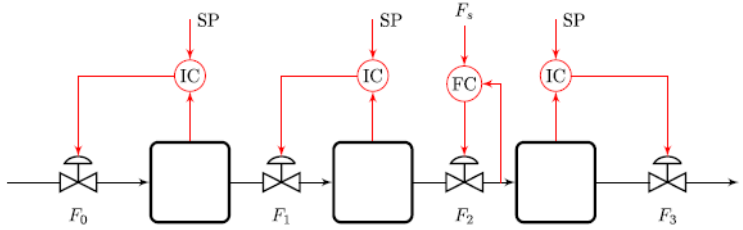
Bidirectional inventory control



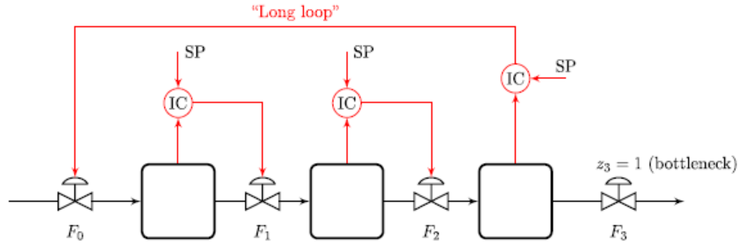
(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)



(b) Inventory control in opposite direction of flow (for given product flow, TPM= F_3)



(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)



(d) Inventory control with undesired "long loop", not in accordance with the "radiation rule" (for given product flow, TPM= F_3)

Fig. 35. Inventory control for units in series. Cases (a), (b) and (c) are in accordance with the "radiation rule".

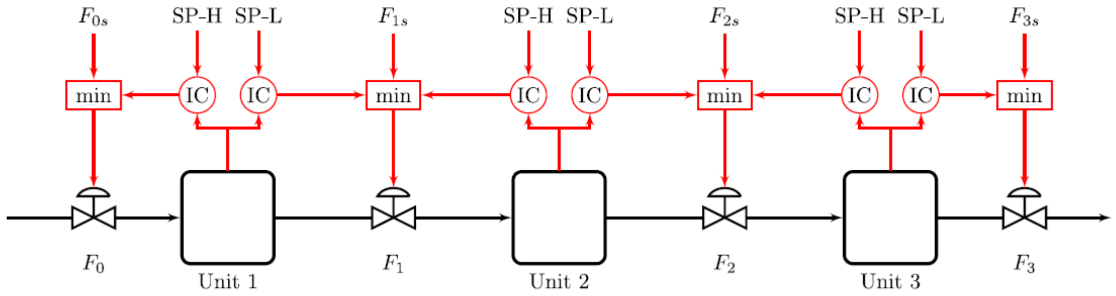


Fig. 36. Bidirectional inventory control scheme for automatic reconfiguration of loops (in accordance with the radiation rule) and maximizing throughput (Shinskey, 1981) (Zotică et al., 2022).
SP-H and SP-L are high and low inventory setpoints, with typical values 90% and 10%. Strictly speaking, since there are setpoints on the (maximum) flows ($F_{i,s}$), the four valves should have slave flow controllers (not shown). However, one may instead have setpoints on valve positions (replace $F_{i,s}$ by $z_{i,s}$), and then flow controllers are not needed.

Economic Plantwide Control of the Ethyl Benzene Process

Rahul Jagtap, Ashok S Pathak, and Nitin Kaistha
Dept. of Chemical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016,Uttar Pradesh, India

DOI 10.1002/aic.13964
Published online December 10, 2012 in Wiley Online Library (wileyonlinelibrary.com).

LS = MIN

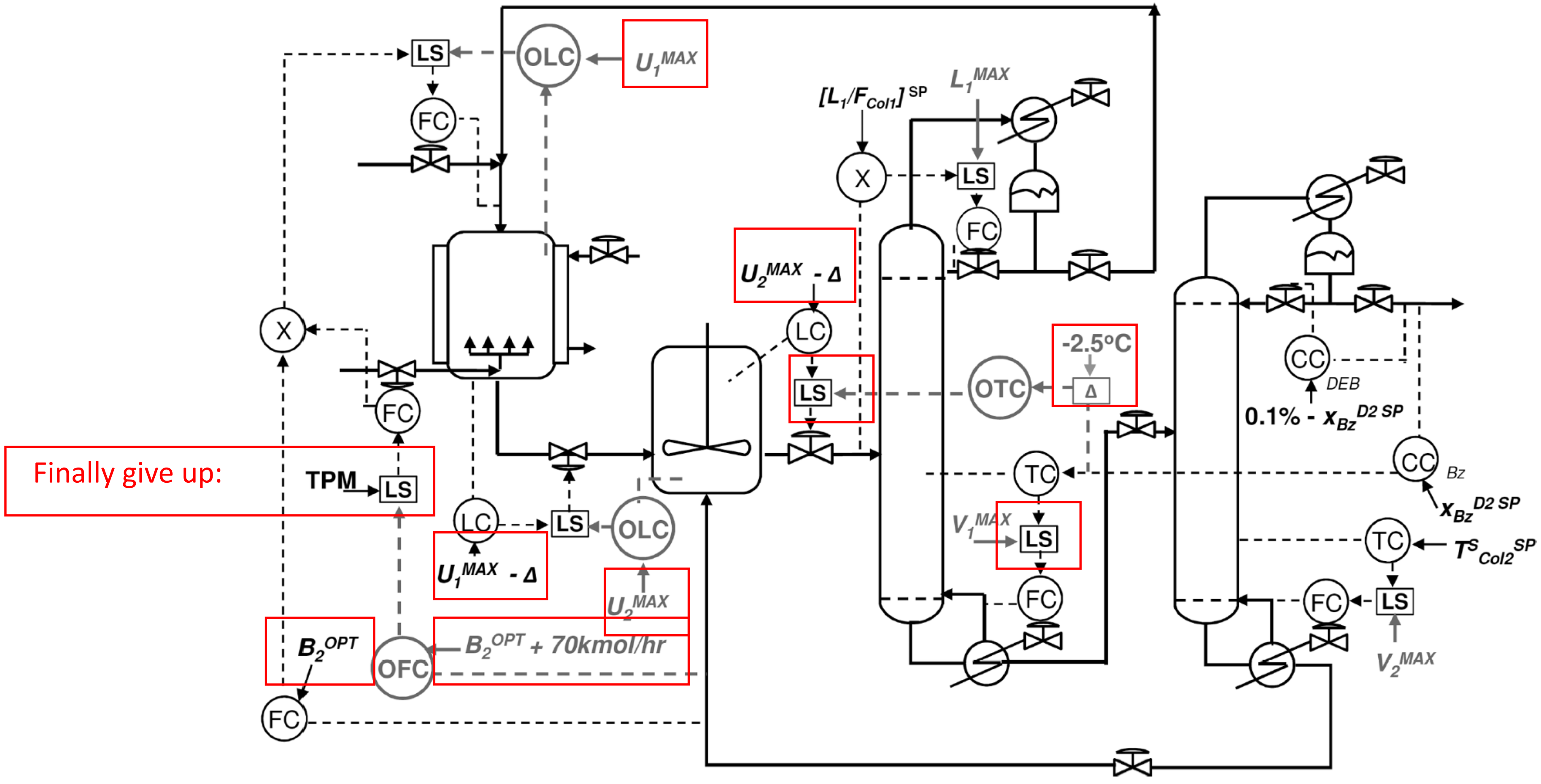
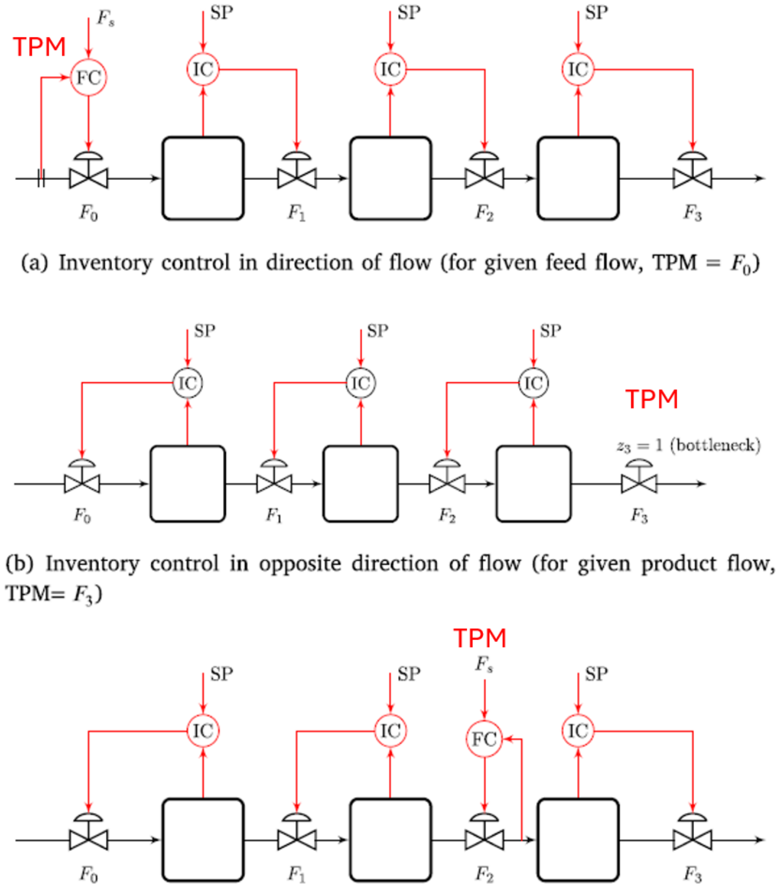


Figure 7. CS2 with overrides for handling equipment capacity constraints.

Rules for inventory control

Rules for inventory control

- **Rule 1.** Cannot control (set the flowrate) the same flow twice
- **Rule 2**. Controlling inlet or outlet pressure indirectly sets the flow (indirectly makes it a TPM)
- **Rule 3**. Follow the radiation rule whenever possible
- ***Radiation rule*** *(actually more a strong recommendation): Inventory control should be "radiating" around a given flow (TPM), that is, it should be in the direction of flow downstream the TPM and it should opposite the direction of flow upstream the TPM.*
  - *Ref:* (Aske & Skogestad, 2009; Buckley, 1964; Price et al., 1994)
- Breaking the radiation rule results in a "long loop", that is, a control loop that only works when other loops are closed
- **Rule 4 (**which should never been broken): No inventory loop should cross the location of the TPM
  - Ref: Not sure, but I have seen it stated



(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)

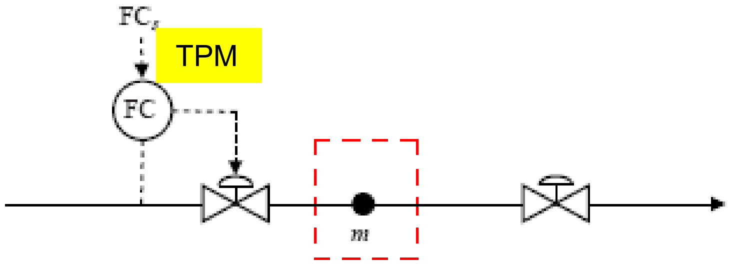
(b) Inventory control in opposite direction of flow (for given product flow, TPM= F_3)

(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)

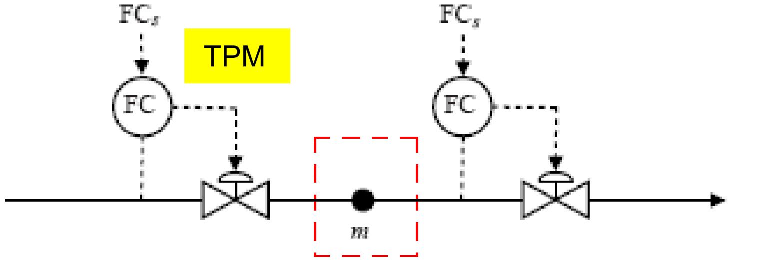
TPM = Variable used for setting the throughput/production rate (for the entire process).

QUIZ. Are these structures workable (consistent)? Yes or No?

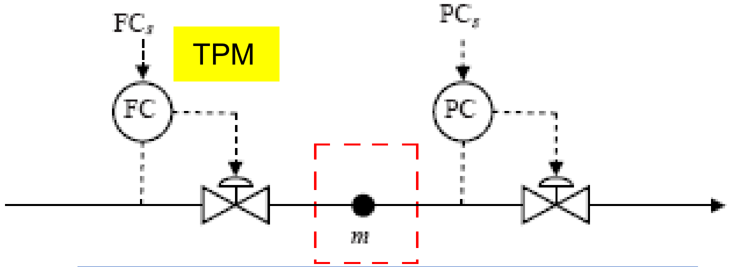
Hint: What happens to the mass holdup inside the red box? Is it self-regulated?



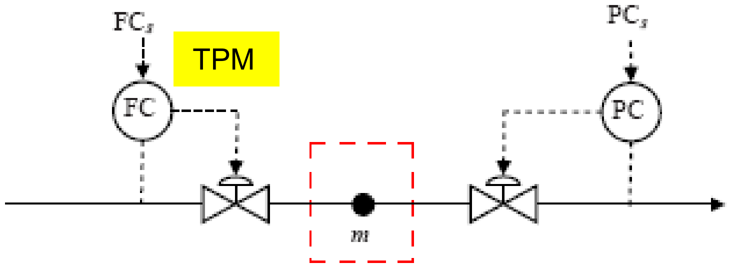
(a) ([redacted]



(b) [redacted]

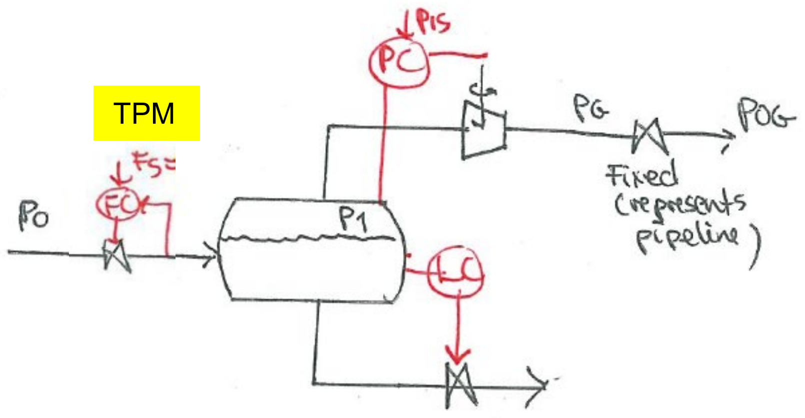


(c) ([redacted]

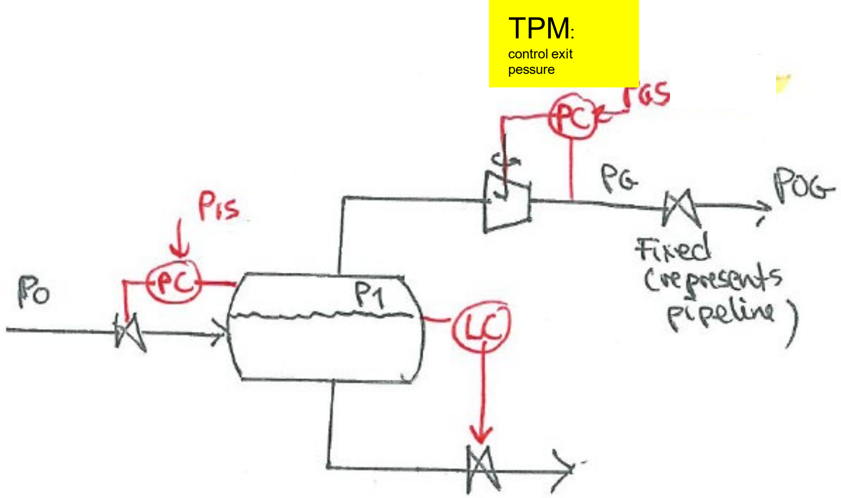


(d) [redacted] ry on inventory m

Quiz 2. Gas-liquid separator.
Where is TPM? Consistent (One is not)?



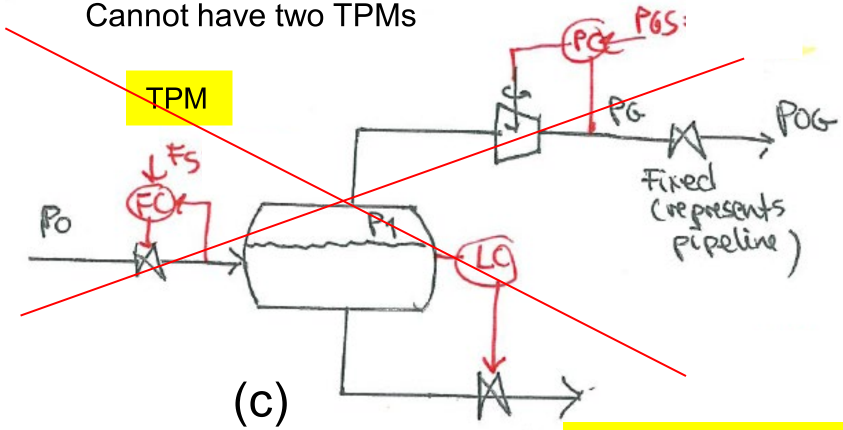
(a)



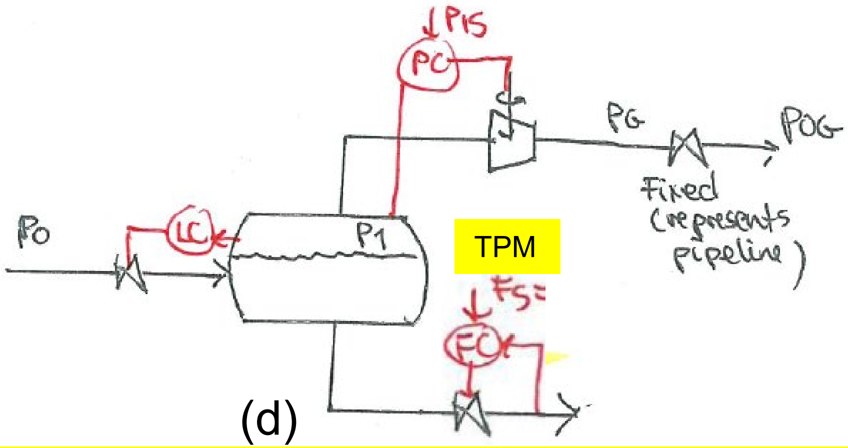
(b)

Doesn't follow radiation rule +
Cannot have two TPMs

TPM



(c)



(d)

Case (a): Given feedrate. Could alternatively set p_0
Cases (b) and (c): Gas production limiting
Case (d): Liquid production limiting

Rule: Setting in-pressure p_0 sets inflow = TPM at inlet or inlet direction (no cases above)
Setting out-pressure p_G sets outflow = TPM at outlet or outlket direction (offdiagonal two cases)

Advanced regulatory control (ARC)

- Using simple standard elements

Standard Advanced control elements

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

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

Next, there are some control elements used for cases when we reach constraints:

E4*. Selective (limit, override) control (for output switching)
E5*. Split range control (for input switching)
E6*. Separate controllers (with different setpoints) as an alternative to split range control (E5)
E7*. VPC as an alternative to split range control (E5)

All the above seven elements have feedback control as a main feature and are usually based on PID controllers. Ratio control seems to be an exception, but the desired ratio setpoint is usually set by an outer feedback controller. There are also several features that may be added to the standard PID controller, including

E8*. Anti-windup scheme for the integral mode
E9*. Two-degrees of freedom features (e.g., no derivative action on setpoint, setpoint filter)
E10. Gain scheduling (Controller tunings change as a given function of the scheduling variable, e.g., a disturbance, process input, process output, setpoint or control error)

In addition, the following more general model-based elements are in common use:

E11*. Feedforward control
E12*. Decoupling elements (usually designed using feedforward thinking)
E13. Linearization elements
E14*. Calculation blocks (including nonlinear feedforward and decoupling)
E15. Simple static estimators (also known as inferential elements or soft sensors)

Finally, there are a number of simpler standard elements that may be used independently or as part of other elements, such as

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

Gives a decomposed control system:

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