Lynkurs Prosessregulering
(Crash course process control)

Sigurd Skogestad
Institutt for kjemisk prosessteknologi
(Department of Chemical Engineering)
Rom K4-211
skoge@ntnu.no

TKP4140. Process control .

In this course you must use both sides of your brain – and try to connect them in the end!

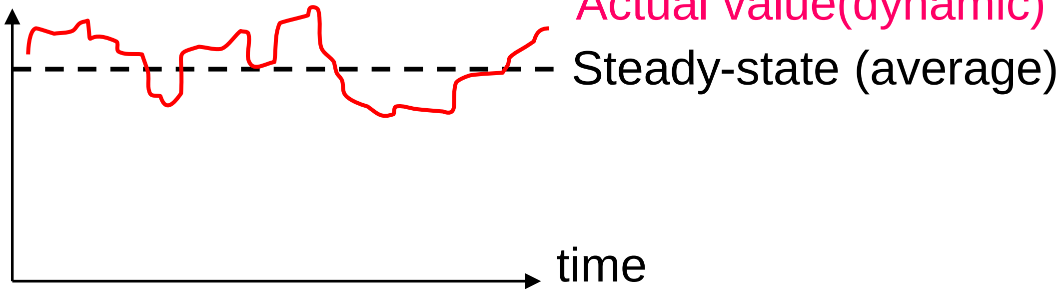
2. Then we switch to the left side:
Modelling - mathematics,
- Laplace - …….

1. We start towards the right side (first two weeks
Control structures – concepts- intuition

It will be a lot of work, but it will be fun, and the goal is that you in the end can design real control systems that work well!

Why control?

- Until now: *Design* of process. Assume steady-state
- Now: *Operation*



In practice never steady-state:

- Feed changes
- Startup
- Operator changes
- Failures
- …..

"Disturbances" (d's)

- **Control is needed to counteract disturbances – remain at steady state**

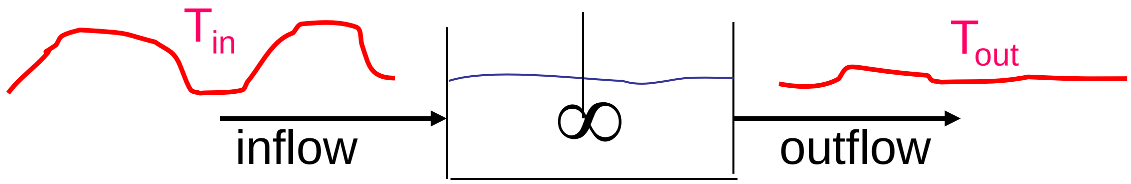
- 30% of investment costs are typically for instrumentation and control

Dealing with disturbances.

Three approaches:

1. Design process so it's insensitive to disturbances

- Example: Use buffer tank to dampen (smoothen) disturbances



2. Detect and remove the source of disturbances

- Sometimes called "statistical process control" (SPC)
  - But it's not control in our sense of the word
- Example: Detect and eliminate variations in feed (use better quality raw material)

3. Counteract disturbances using MV (this course).

MV = manipulated variable (usually valve

3. Counteract disturbances (this course).

Process control ("prosessregulering"):

Do something (usually manipulate valve)

to **counteract** the **effect** of the disturbances

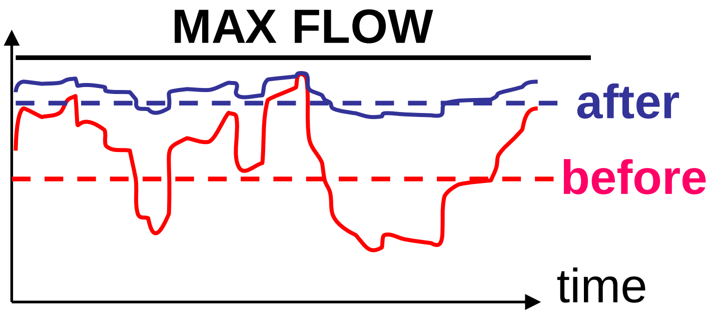
(a) Manual control: Need operator

(b) Automatic control: Need measurement + automatic valve + computer

Goals automatic control:

- Smaller variations
  - more consistent quality
  - More optimal ("squeeze and shift")
- Smaller losses (environment)
- Lower costs
- More production

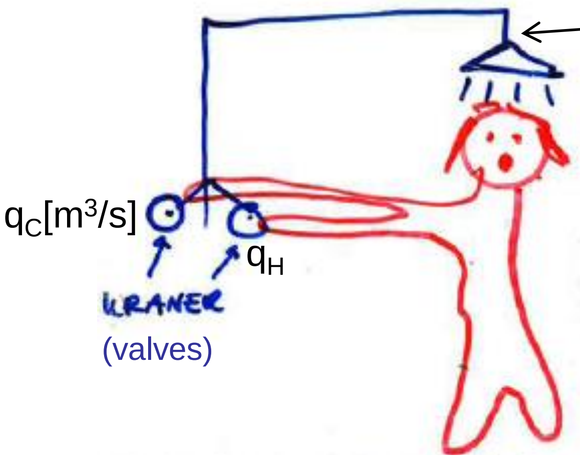
Industry: Still large potential for improvements!



By improving control and squeezing the variations we can shift the setpoint (average) closer to the constraint and increase production

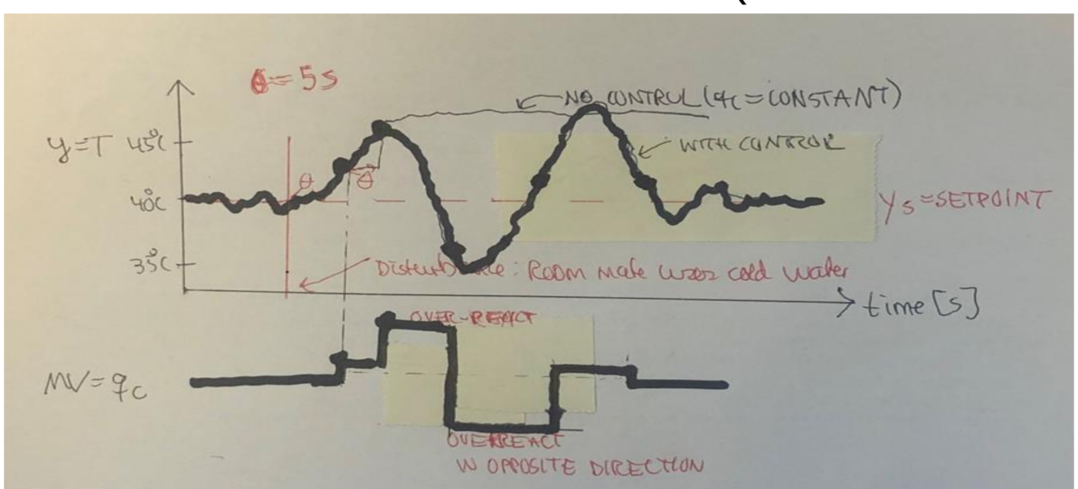
Example: Control of shower temperature

Temperature: 5s delay in pipe ($\theta = V/q$ = 100ml/20 ml/s = 5s)



T [K]
q [m3/s]
} want constant

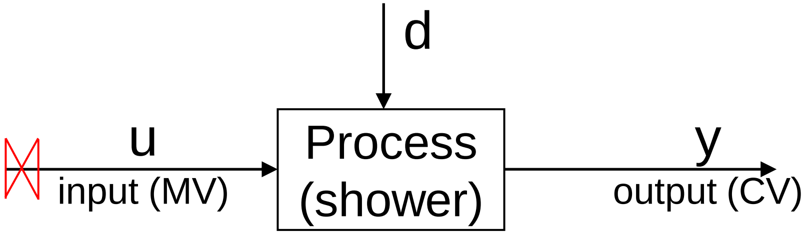
Disturbance: Room mate brushes teeth (uses cold water)



Time delay θ is the main enemy of feedback control

Time delay combined with over-reaction (too high gain): Can give instability

Classification of variables



Independent variables ("the cause"):

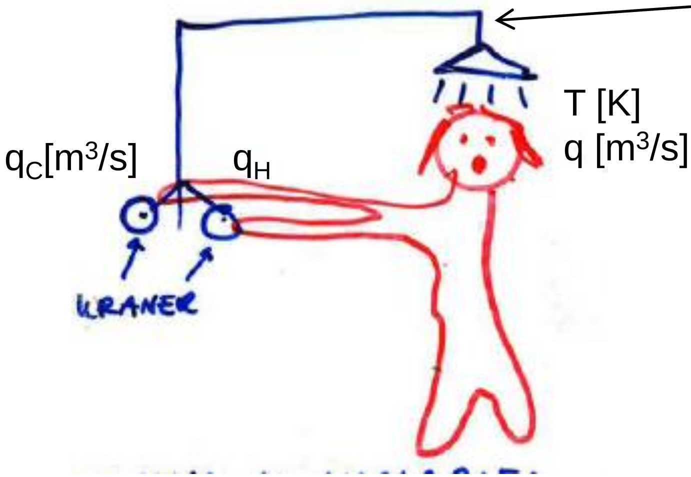
- **(a) Inputs (MV, u):** Variables we can adjust (valves)
- **(b) Disturbances (DV, d):** Variables outside our control

Dependent (output) variables ("the effect or result"):

- (c) **Primary outputs (CVs, y):** Variables we want to keep at a given setpoint
- (d) Internal variables in dynamic model ("states") (x)

MV = manipulated vartiable (input u)
CV = controlled variable (output y)
DV = disturbance variable (d)

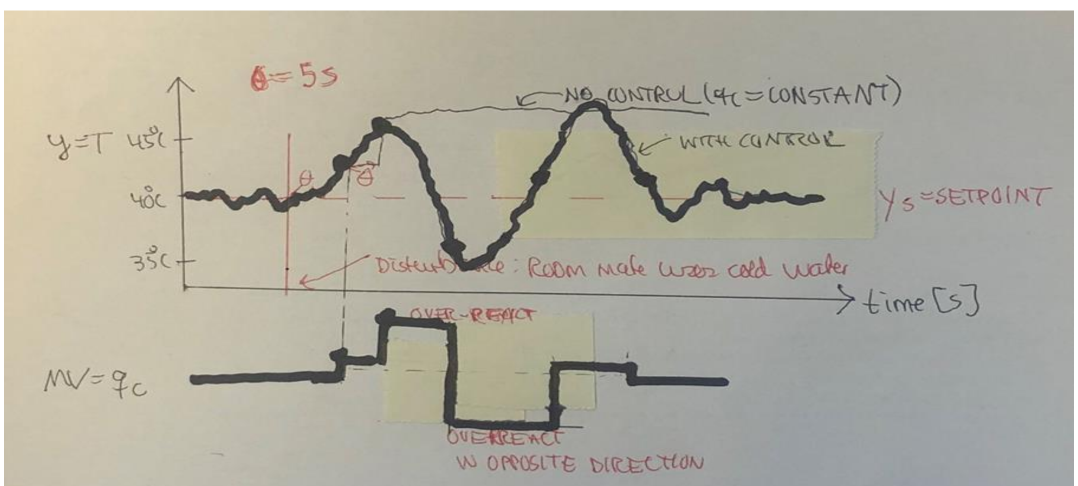
Classification of variables: Shower process



5s delay: $\theta = V/q$ = 100ml/20 ml/s = 5s

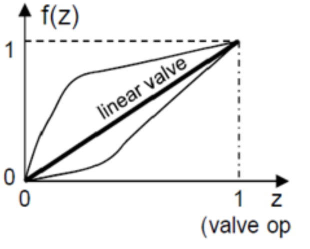
Control objective. MVs, CVs, DVs

1. Control objective
   - Keep temperature (y_1=T) a given setpoint
   - Keep total flow (y_2=q) at given setpoint
2. Classify variables
   - MVs (u) = q_H, q_C (strictly speaking, valve positions z_H, z_C)
   - CVs (y) = T, q
   - DVs (d) = q_H, q_C (strictly speaking, upstream pressures, p_H, p_C, which gives "uncontrolled" flow changes)



Inputs for control (MVs)

- Usually: Inputs (MVs) are valves.
  - Physical input is valve position (z), but we often simplify and say that flow (q) is input

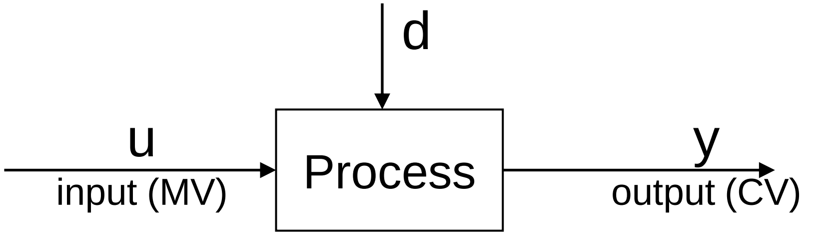


$$\text{Valve equation: } q[m^3/s] = C_v f(z)\sqrt{DP/\rho}$$

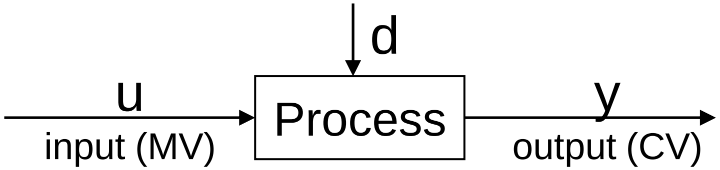
DP = pressure drop across valve [N/m^2]
f(z) = valve characteristic (Linear valve: f(z)=z)
C_v = valve constant [m^2]
ρ = fluid density [kg/m^3]

Control

- Use inputs (MVs, u) to counteract the effect of the disturbances (DVs, d) such that the outputs (CVs, y) are kept close to their setpoints (y_s)



Two fundamental control principles

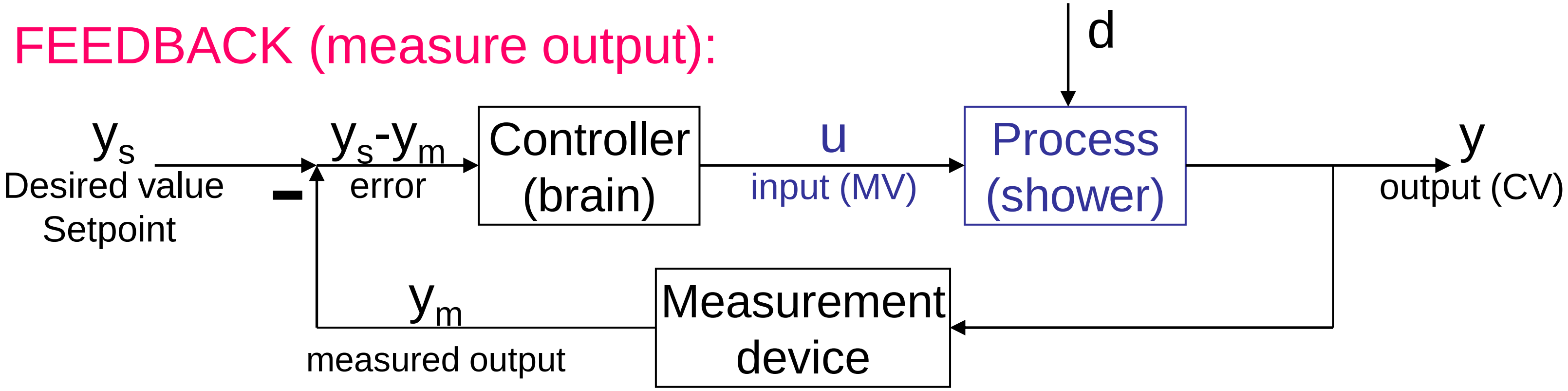


- **Feedback**: **Measure the result** (= controlled variable CV; output y) and keep adjusting the manipulated variable (MV; input u) until the results is OK
  - Example: Measure the temperature y=T (CV) and adjust the flow (u) of cold water (MV)

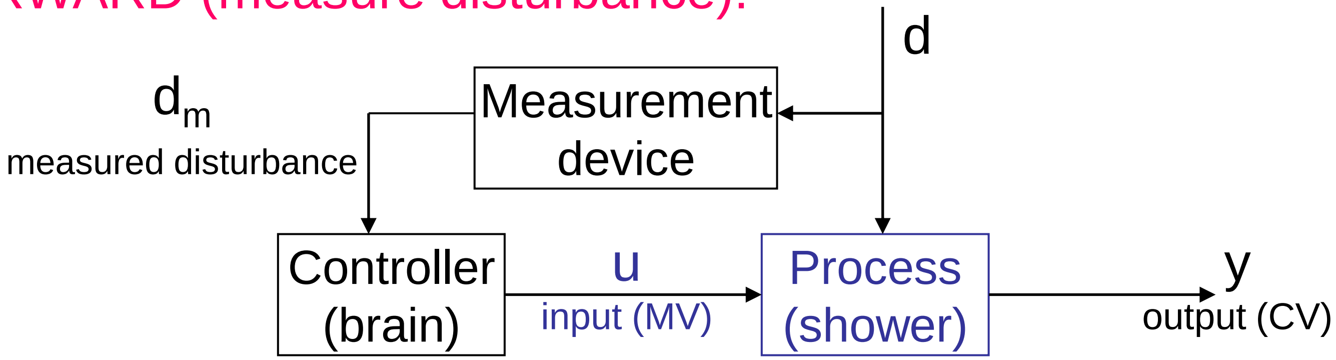
- **Feedforward**: **Measure the cause** (= disturbance d; DV) and based on a prediction (model!) make a "feedforward" adjustment of the MV (input u) to (hopefully) counteract its effect on the result (output y)
  - Feedforward control requires a good model!
  - Example: Room mate (disturbance d) says "I am tapping cold water" - and you know your friend so well (model) that you can make the correct increase in your cold water (MV) to counteract d.
  - NOT VERY REALISTIC FOR SHOWER EXAMPLE
  - BUT a good example of feedforward is coming in time to lecture!

BLOCK DIAGRAMS

FEEDBACK (measure output):



FEEDFORWARD (measure disturbance):



- All lines: Signals (information)
- Blocks: controllers and process
- Do not confuse block diagram (lines are signals) with flowsheet (lines are flows); see below

FEEDBACK

+ Self-correcting with negative feedback (keeps adjusting until $y=y_s$ at steady state)
+ Do not need model (but must know process sign!)
- May give instability if controller overreacts 
- Need good and fast measurement of output

MAIN ENEMY OF FEEDBACK: TIME DELAY
(in process or in measurement of y)

FEEDFORWARD

+ Good when large time delay (in process or in measurement of y)
+ May react before damage is done
- Need good model
- Sensitive to changes and errors
- Works only for known and measured disturbances

USUALLY COMBINED WITH FEEDBACK

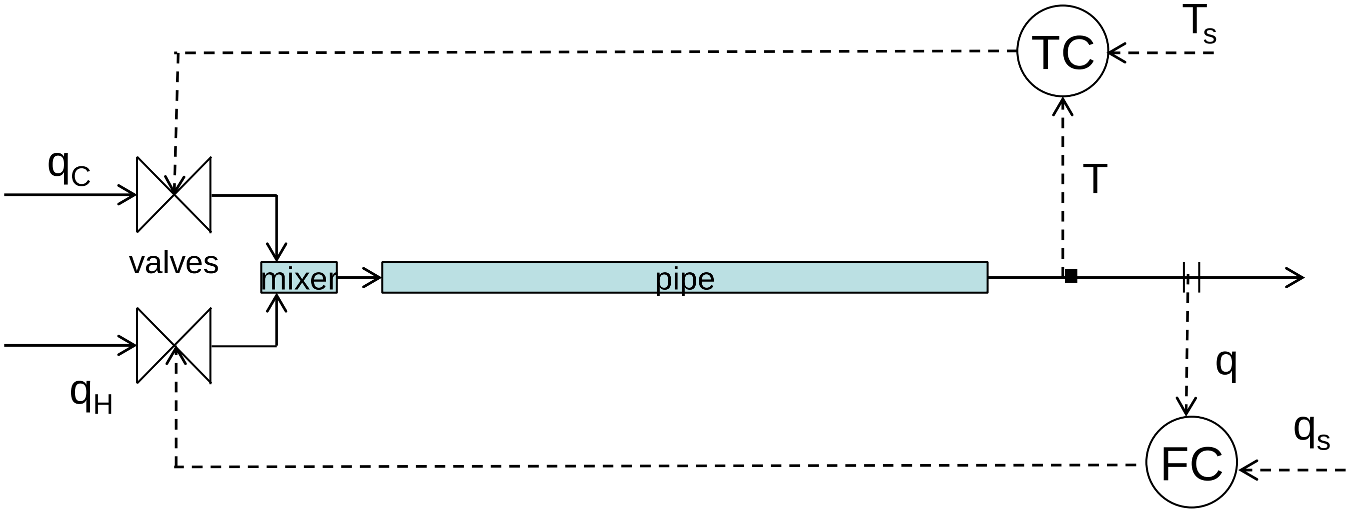
We use two kind of diagrams

- **Block diagram (information)**
  - Used by control engineers
- **Flowsheet (piping & instrumentation diagram, P&ID)**
  - Used by process engineers

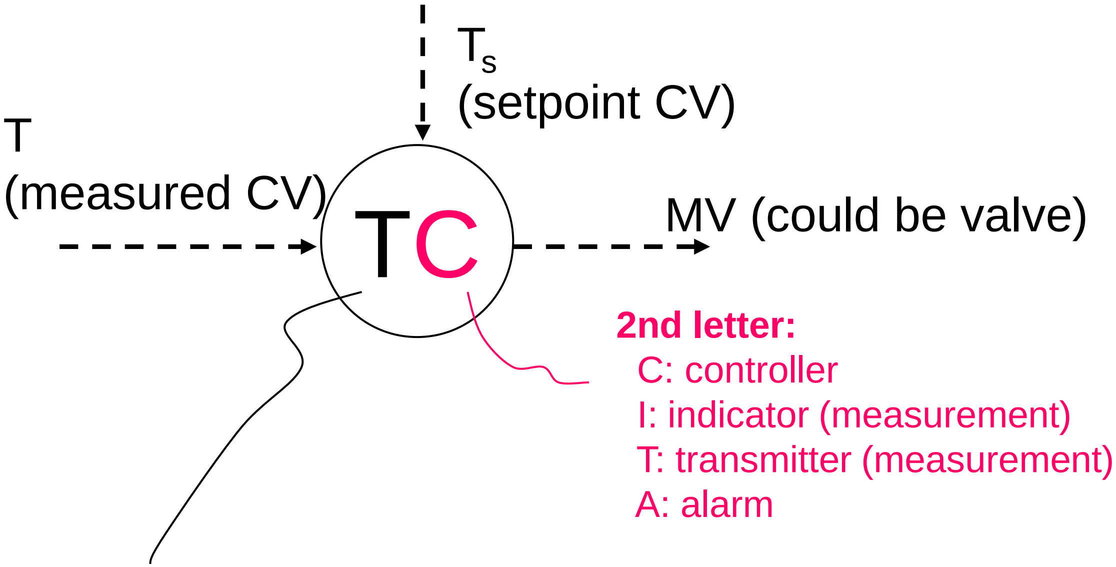
Piping and instrumentation diagram (P&ID) (flowsheet)

- Solid lines: mass flow (streams)
- Dashed lines: signals (control)

Example: Shower



Notation feedback controllers (P&ID)

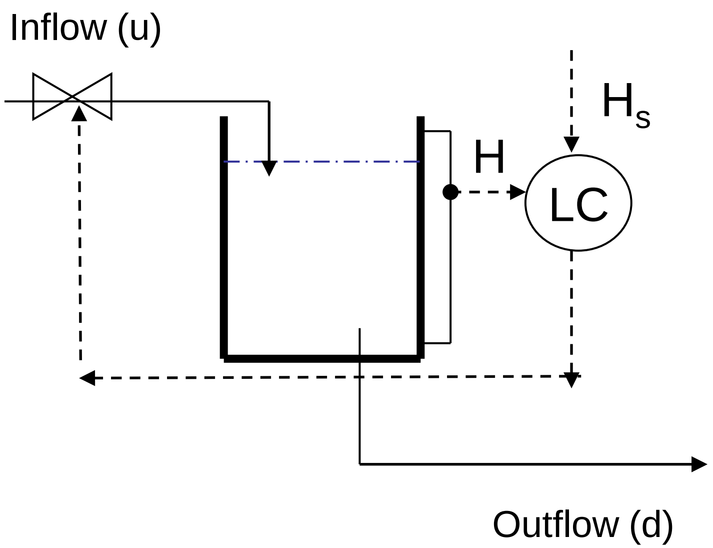


1st letter: Controlled variable (CV) = What we are trying to control (keep constant)

- **T: temperature**
- **F: flow**
- **L: level**
- **P: pressure**
- DP: differential pressure (Δp)
- A: Analyzer (composition)
- C: composition
- X: quality (composition)
- H: enthalpy/energy

Example flowsheet: Level control

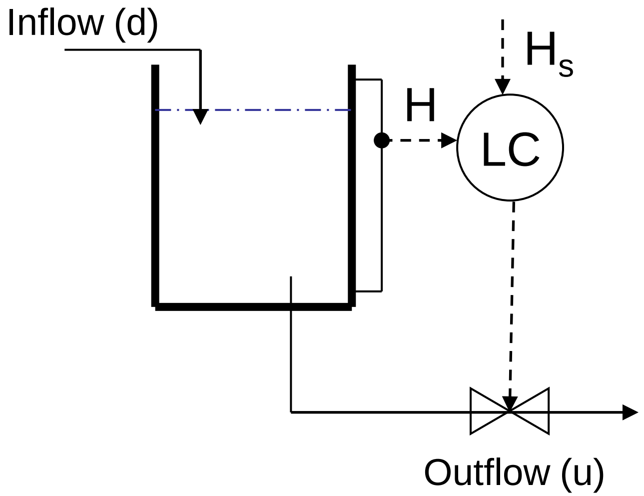
(with given outflow)



CLASSIFICATION OF VARIABLES FOR CONTROL (MV, CV, DV):

INPUT (u, MV): INFLOW
OUTPUT (y, CV): LEVEL
DISTURBANCE (d, DV): OUTFLOW

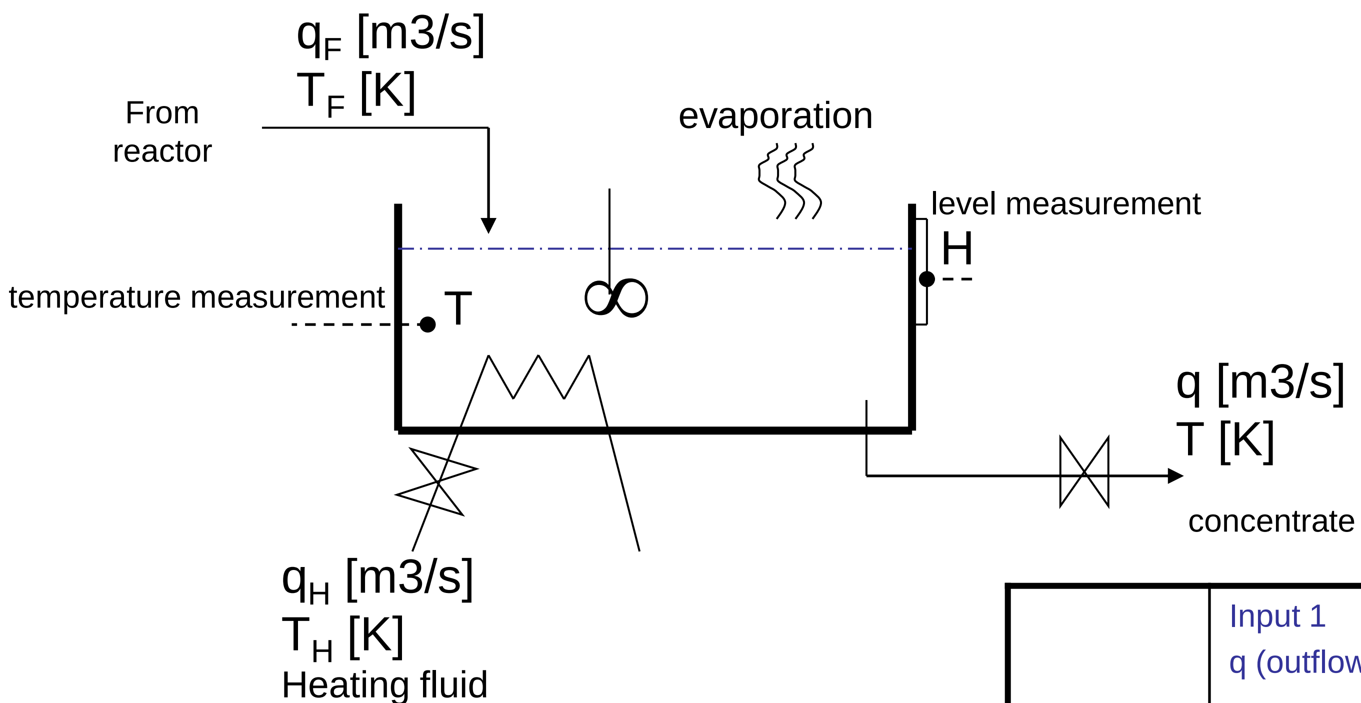
Level control when inflow is given (more common)



CLASSIFICATION OF VARIABLES FOR CONTROL (MV, CV, DV):

INPUT (u, MV): OUTFLOW (Input for control!)
OUTPUT (y, CV): LEVEL
DISTURBANCE (d, DV): INFLOW

Example: Evaporator with heating



	Input 1 q (outflow)	Input 2 q_H
Output 1 H	-	-
Output 2 T	0	+

Suggested pairings

1. Control objective
   - Keep level H at desired value
   - Keep temperature T at desired value
2. Classify variables (CVs, MVs, important DVs)
3. Qualitative Process matrix (+-0 from MVs to CVs) to help with pairings
4. Suggest pairings and put control loops on the flowsheet

Avoid pairing on zero gain

Most important control structures

1. Feedback control
2. Cascade control
3. Ratio control (special case of feedforward)

The last two are used if standard feedback is not good enough, typically because of delay in measurement of y.

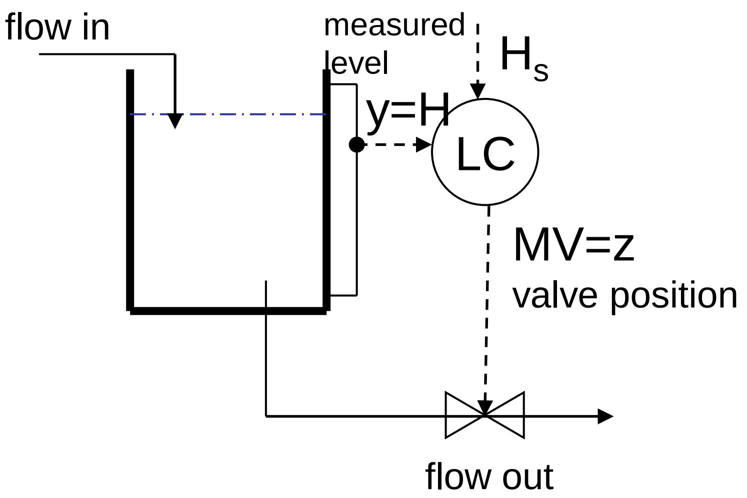
- Cascade: Use extra output measurement (y_2)
- Ratio/feedforward: Use disturbance measurement (d)

Cascade control

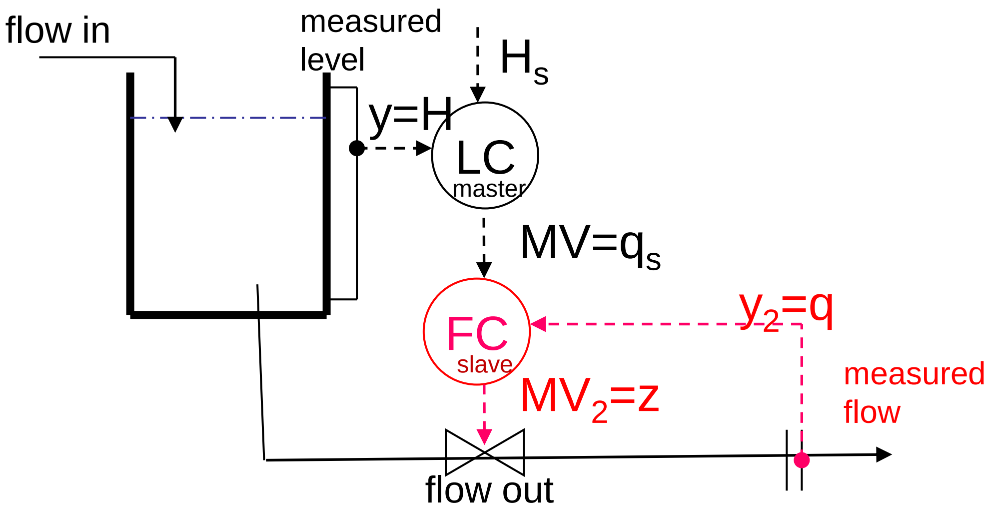
- **Controller ("master") gives setpoint to another controller ("slave")**
  - Without cascade: "Master" controller directly adjusts u (input, MV) to control y
  - With cascade: Local "slave" controller uses u to control "extra"/fast measurement (y_2). "Master" controller adjusts setpoint y_{2s}.

- **Example: Flow controller on valve (very common!)**
  - y = level H in tank (or could be temperature etc.)
  - u = valve position (z)
  - y_2 = flowrate q through valve

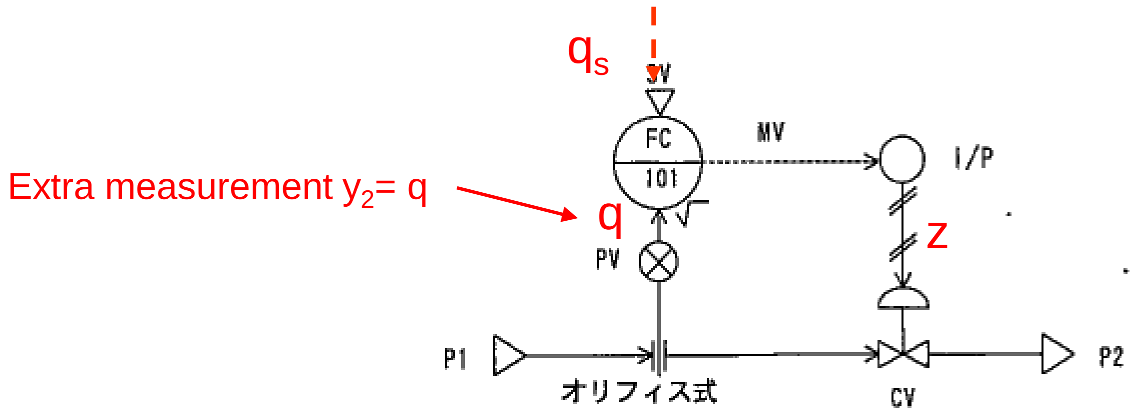
WITHOUT CASCADE



WITH CASCADE (2 controllers)



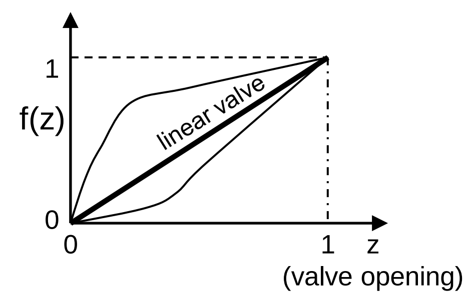
What are the benefits of adding a flow controller (inner cascade)?



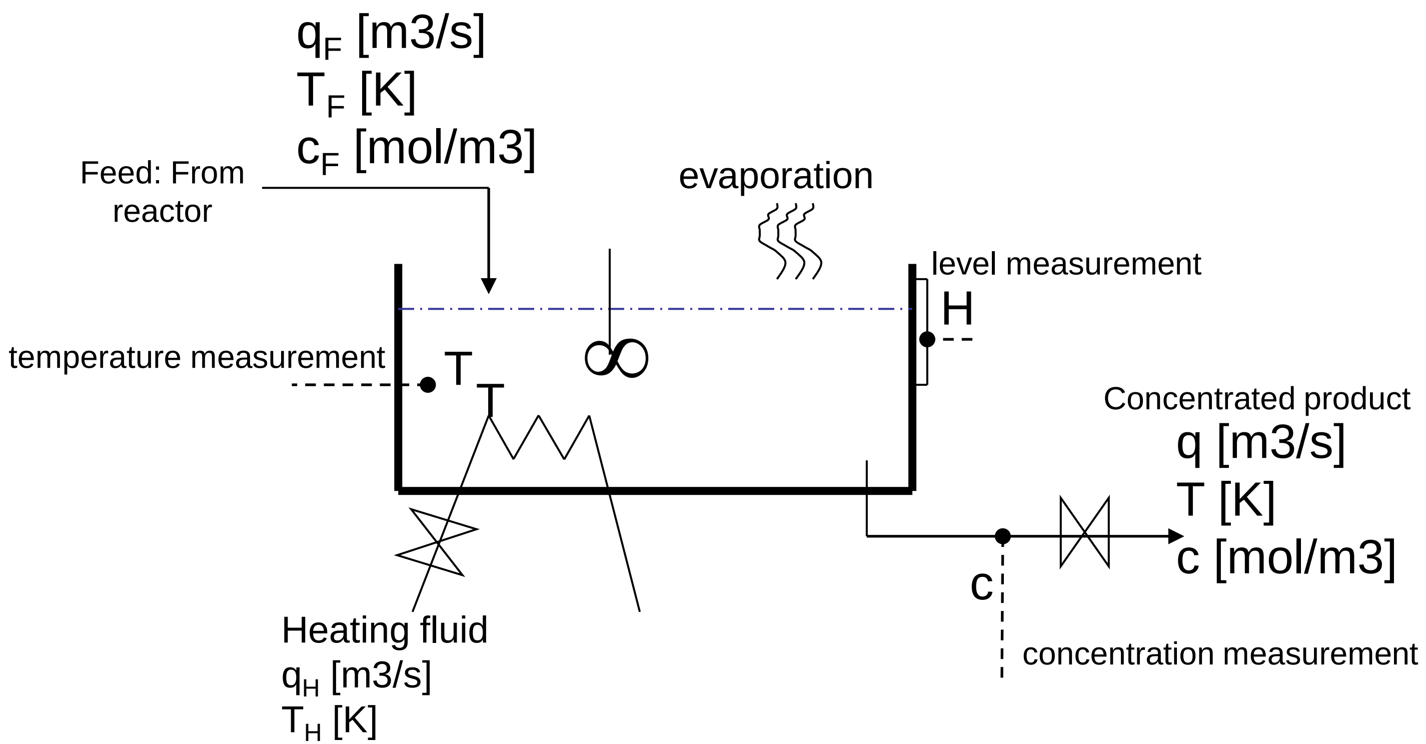
$$\text{Flow rate: } q = C_v f(z)\sqrt{\frac{p_1 - p_2}{\rho}} \quad [\mathrm{m}^3/\mathrm{s}]$$

1. Fast local control: Eliminates effect of disturbances in p_1 and p_2 (FC reacts faster than outer level loop)

2. Counteracts nonlinearity in valve, f(z)
   - With fast flow control we can assume $q = q_s$



Example: Evaporator with heating



Control objectives

- Keep level H at desired value
- NOW: Keep composition c at desired value

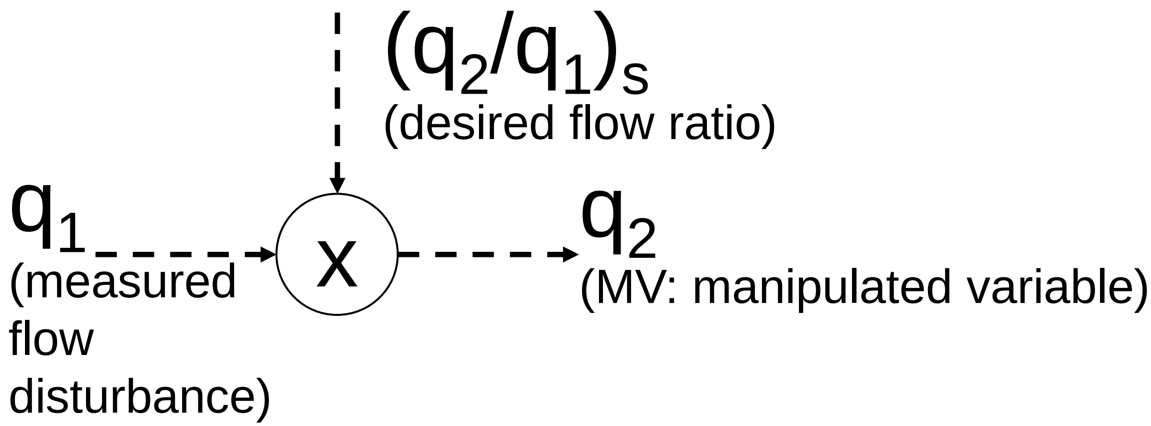
BUT: Composition measurement has large delay + unreliable

Suggest control structure based on cascade control

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 $(q_2/q_1)_s$"

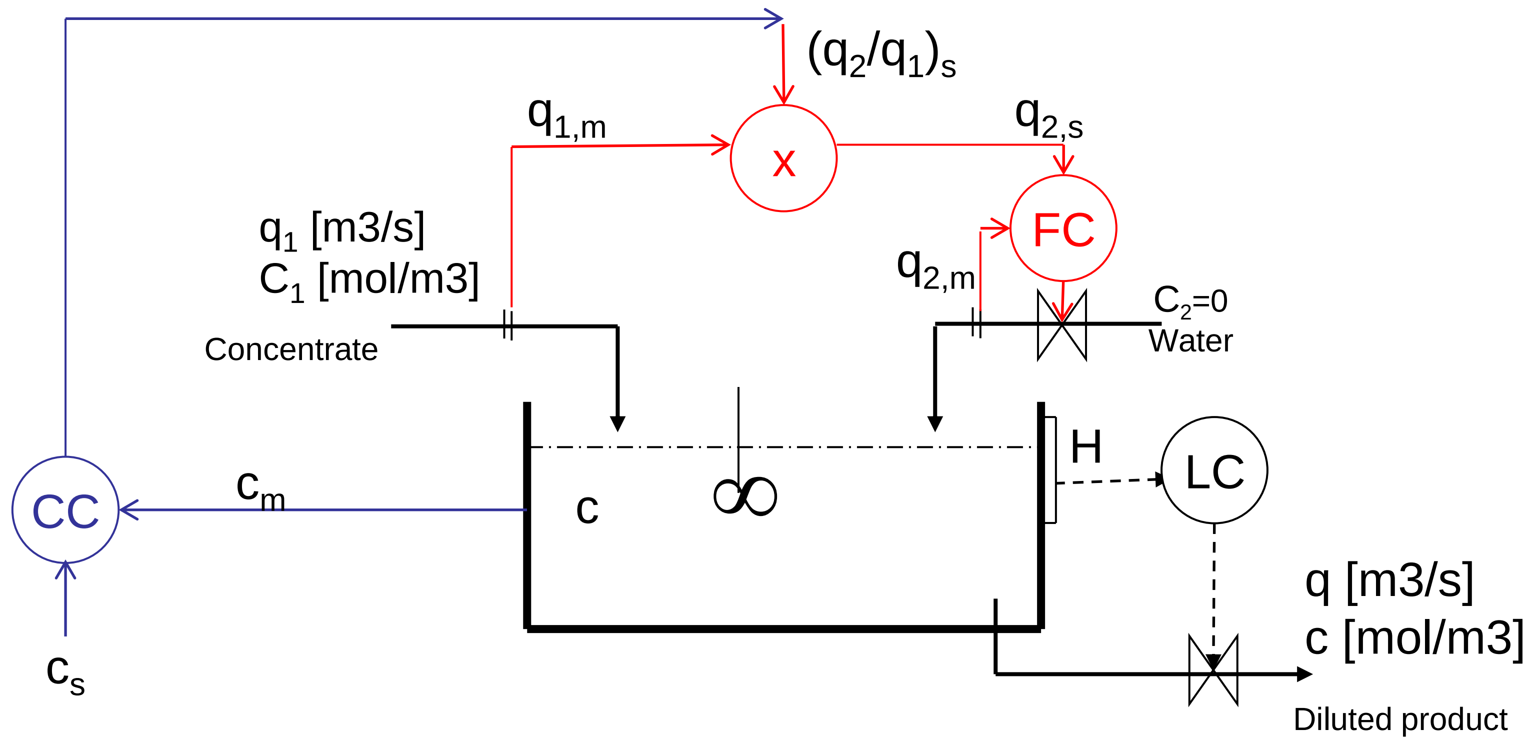
Usually: Combine ratio (feedforward) with feedback

- Adjust $(q_1/q_2)_s$ based on feedback from process, for example, composition controller.
  - This is 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 y=c to fine tune ratio)*

EXAMPLE: MIXING PROCESS

RATIO CONTROL with outer cascade (to adjust ratio setpoint)



Procedure for design of control system

1. Define control objective (why control?)
2. Classify variables
   - MVs (u)
   - Disturbances (d)
   - CVs (y)

   \+ measurements
3. Process description
   - Flow sheet
   - Process matrix

     - Qualitative: with 0, +, -, (+)*, (-)*
     - Quantitative: transfer matrix (see later)
4. Control structure
   - Feedforward / feedback
   - Pairing of variables (avoid pairing on 0!)
   - Cascade loops (MV from one controller (master) is setpoint for another (slave))
   - Put on process & instrumentation diagram (P&ID)
5. Control algorithm
   - On/off
   - PID (proportional-integral-derivative)
   - Model based (MPC)
6. Implementation
   - Today: Normally computer + connect measurements and valves (actuators)

Process matrix

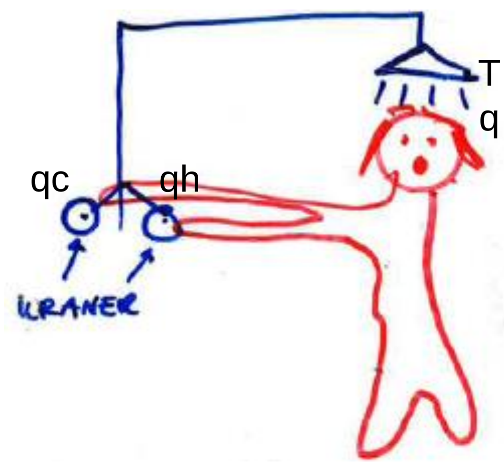
	Input 1	input2
Output 1	+	-
Output 2	0	+

Process engineer (YOU):
- Responsible for items 1- 4
- The most important is process understanding

*(has some effect, but too small for control)

Example: Shower

1. Define control objective (why control?)
   - CVs: Control temperature T and flow q
2. Classify variables
   - MVs (u): qc, qh
   - Disturbances (d): Focus on main
   - CVs (y): T, q
3. Process description
   - Flow sheet
   - Process matrix
4. Control structure
   - Pairing of variables (Alt.1, Alt.2)
   - Multivariable (Alt 3)



	Input 1 qc	Input2 qh
Output 1 T	-	+
Output 2 q	+	+

In this case the process matrix has no 0's ⇒ Interactive, so pairing is not obvious!

Multivariable control ("decoupling") is used in practice:
One handle (up/down) for total flow (qh+qc), one (left/right) for ratio (qh/qc)

3x3 pairing example

Pairing: Choose one pairing from each row/column. Avoid pairing on 0's

Inputs

Outputs		u_1	u_2	u_3
	y_1	(+)	+	+
	y_2	0	+	(-)
	y_3	0	(+)	0

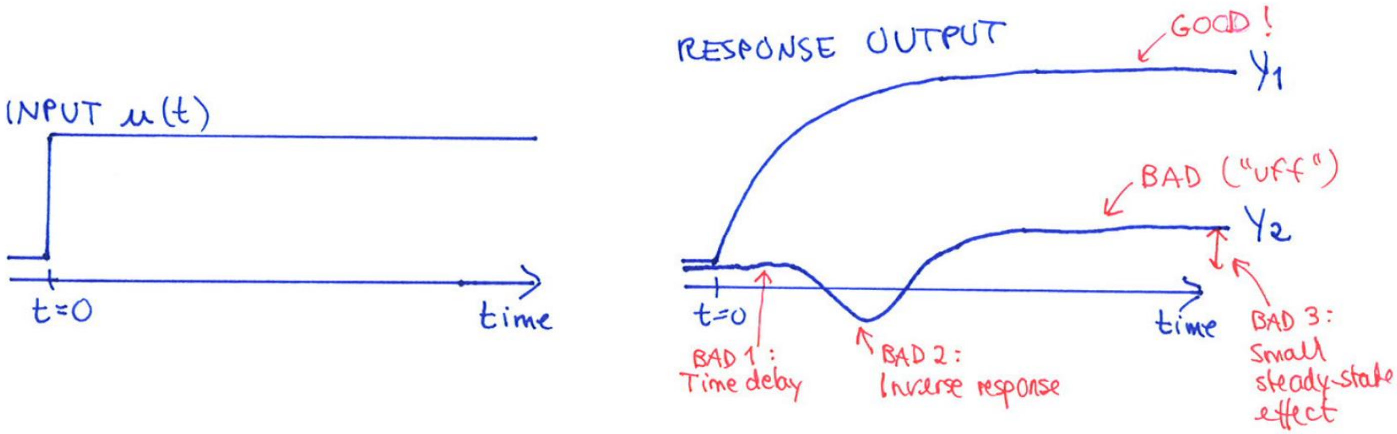
Conclusion:

$$y_1 \leftrightarrow u_1$$
$$y_2 \leftrightarrow u_3$$
$$y_3 \leftrightarrow u_2$$

Rules for pairing of variables and choice of control structure

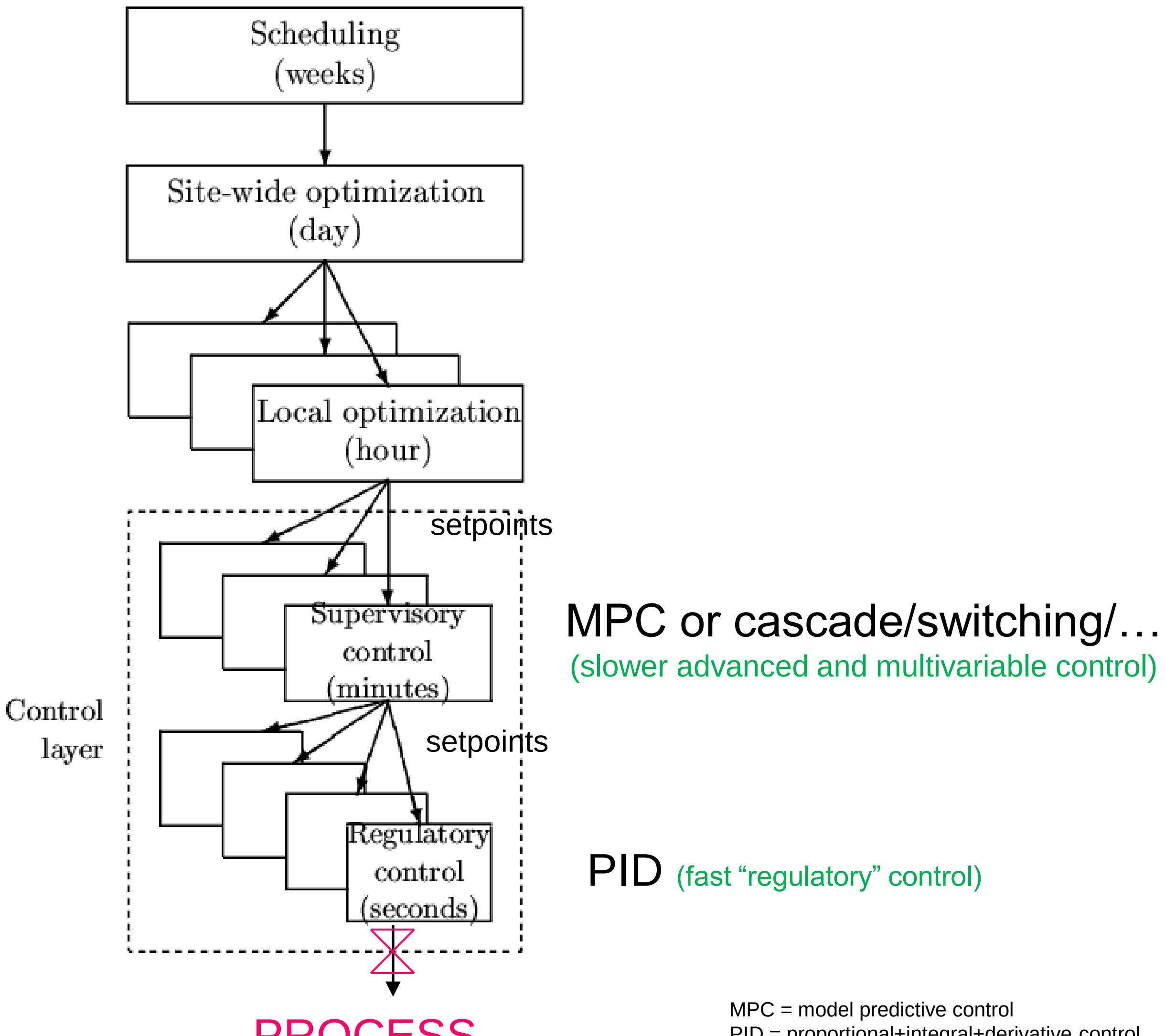
Main rule: "Pair close"

1. The response (from input to output) should be fast, large and in one direction. *Avoid pairing on 0!* Avoid dead time and inverse responses!



2. The input (MV) should preferably affect only one output (to avoid interaction between the loops; want some 0's in the process matrix)
3. Try to avoid input saturation (valve fully open or closed) in "basic" control loops for level and pressure
4. The measurement of the output y should be fast and accurate. It should be located close to the input (MV) and to important disturbances.
   - Use extra measurements y2 and cascade control if this is not satisfied
5. The system should be simple
   - Avoid too many feedforward and cascade loops
6. "Obvious" loops (for example, for level and pressure) should be closed first, before you spend too much time on deriving process matrices etc.

Control hierarchy based on "time scale separation"



MPC = model predictive control
PID = proportional+integral+derivative control

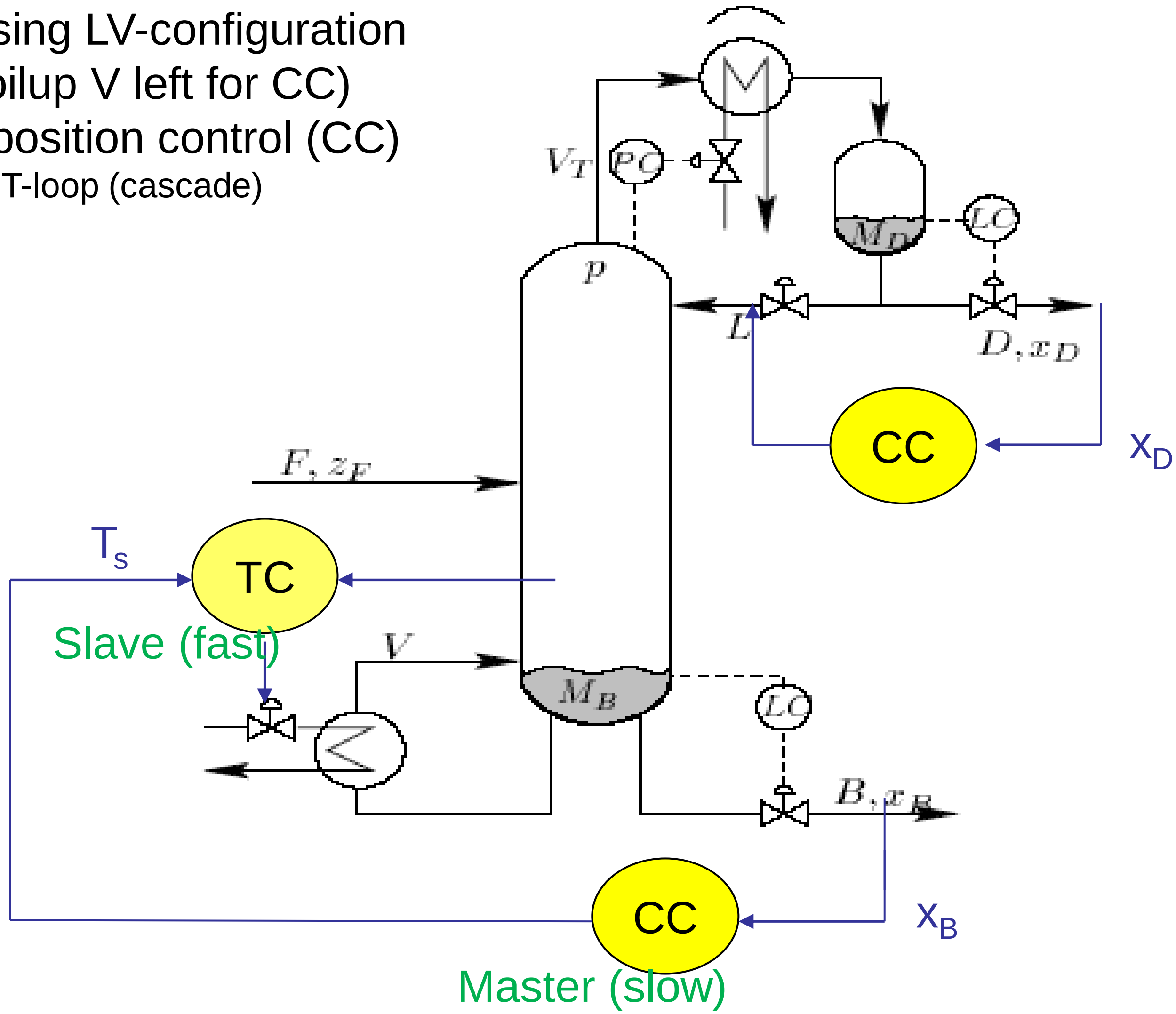
Example: Distillation

- Here: Given feed (i.e., feedrate is disturbance)

1. Objective: “Stabilize” column + keep compositions in top and bottom constant
   - But compositions measurements are delayed + unreliable
2. Classify variables
3. Process description
   - Flowsheet
   - Process matrix
4. Control structure: Stabilize column “profile” using sensitive temperature measurement.

Typical distillation control:

- Level control using LV-configuration (reflux L and boilup V left for CC)
- Two-point composition control (CC)
  - with inner T-loop (cascade)



Inventory control rule (TPM)

- Inventory control: Usually control of level and pressure
- Throughput manipulator (TPM): Where the operator sets production rate
- Usually at the feed, but better place: Close to throughput bottleneck
- **Rule** for level and pressure control («to keep things flowing»): Inventory control must be radiating around TPM

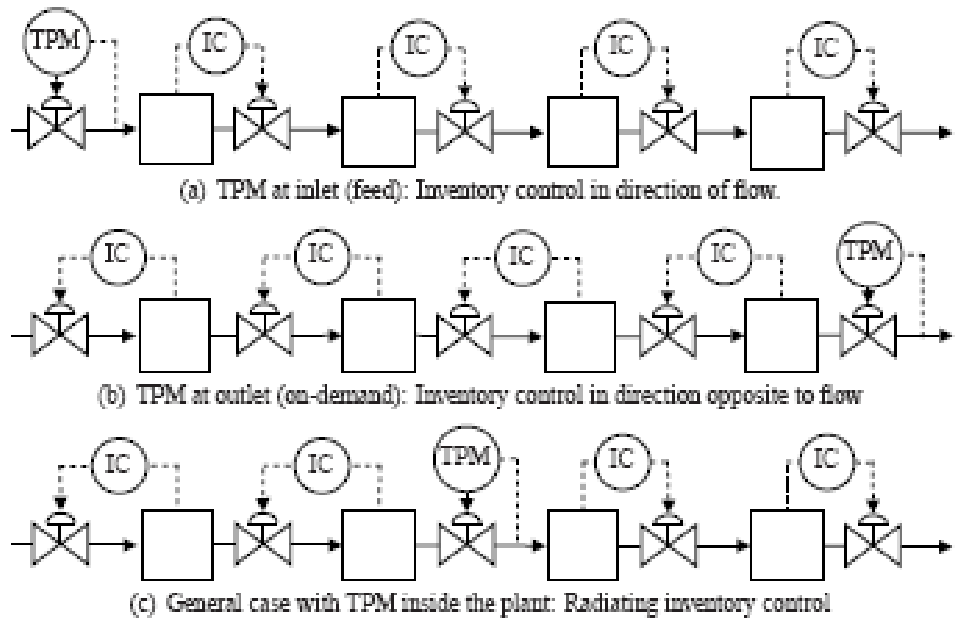
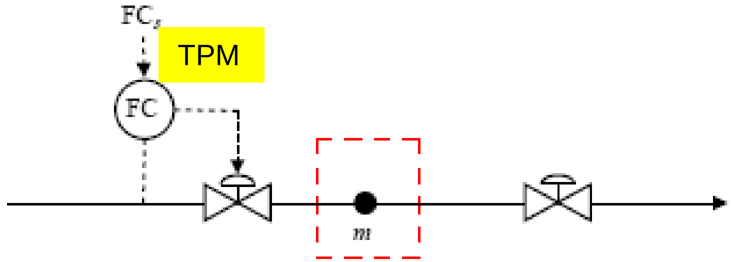


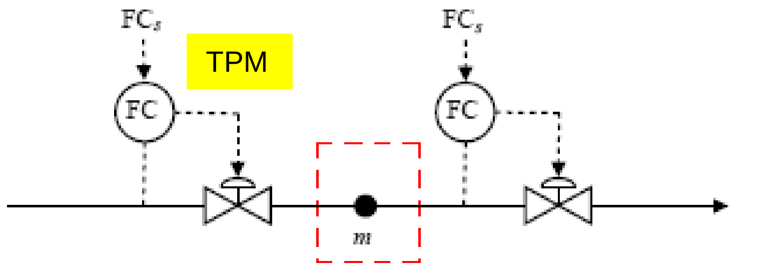
Figure 2.6: Self-consistency requires a radiating inventory control around a fixed flow (TPM)

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

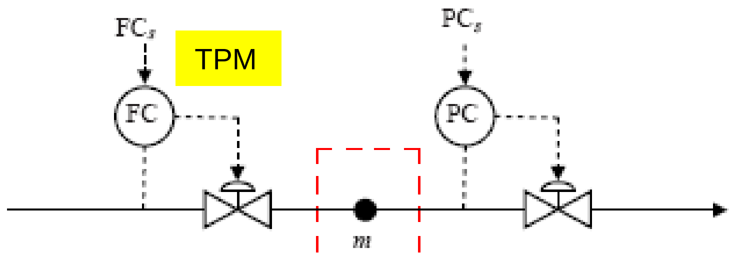
Hint: What happens to the mass holdup inside the red box? Is it self-regulated?



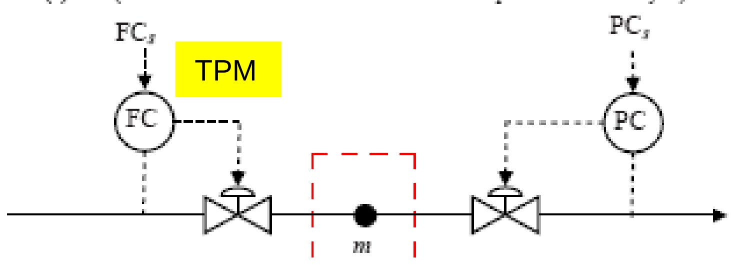
(a) (



(b)

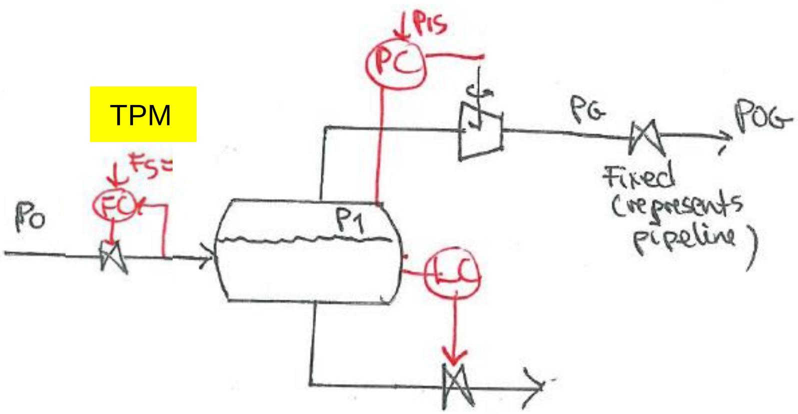


(c) (

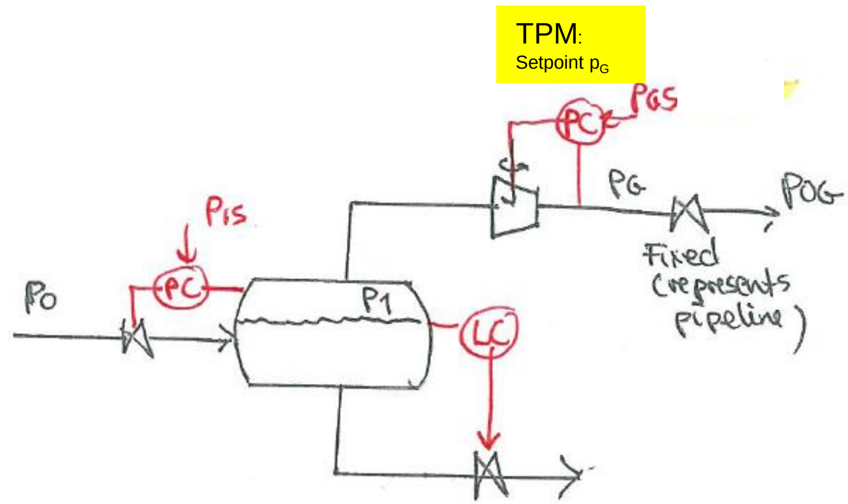


(d) ry on in-
ventory m

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

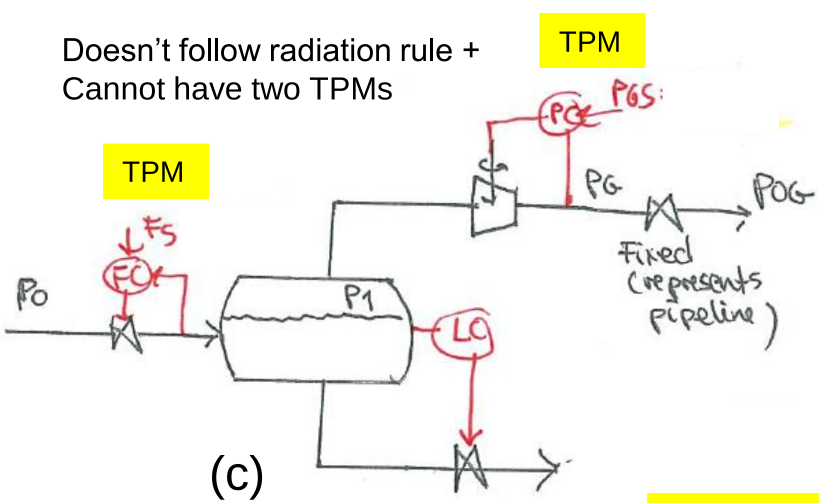


(a)

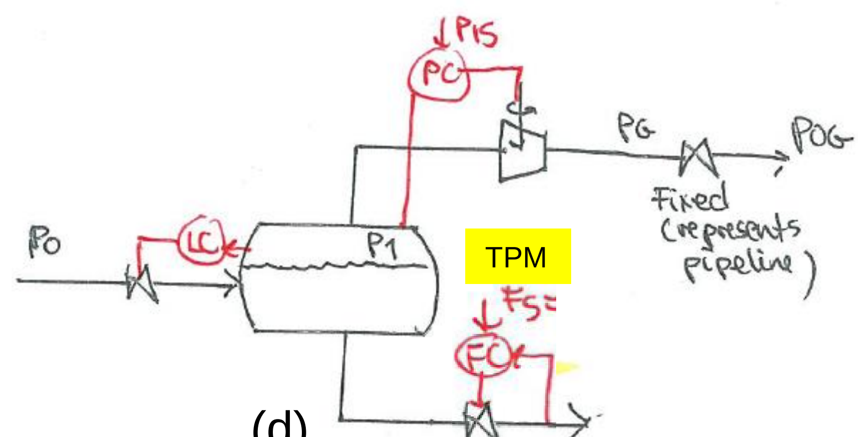


(b)

Doesn't follow radiation rule +
Cannot have two TPMs



(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
Setting out-pressure p_G sets outflow = TPM at outlet (left two cases)