

Practical Process Control

Examples from a chemical industry

Krister Forsman
2023-11-16

Many different types of processes to control

- Reactors
 - Tube reactors, tank reactors
 - Continuous, batch, semi-batch
- Heat exchangers
 - Liquid – liquid, steam - liquid
- Distillation columns
 - Continuous, batch
- Evaporators
- Vaporizers
- Crystallizers
- Centrifuges
- Filters
- Dryers
- Boilers
- ...

In order to optimize controls you need to have good process understanding.



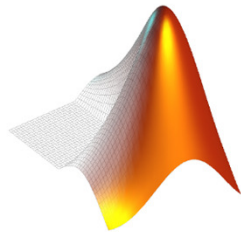
Typical tasks for the control group

Improve productivity by decreased variation and increased automation.

- Smarter control structures, e.g. feedforwards, mid-ranging, cascades, maximizing control, ratio-in-cascade, split-range, conditional control
 - PID control parameter tuning
 - Plant wide control
 - Introduce new controllers
 - Support in design and commissioning of new plants
-
- Vast majority of all work is done on existing plants.
 - "Control optimization"
 - Assumption: Minimal or no changes are to be made in equipment "No steel"

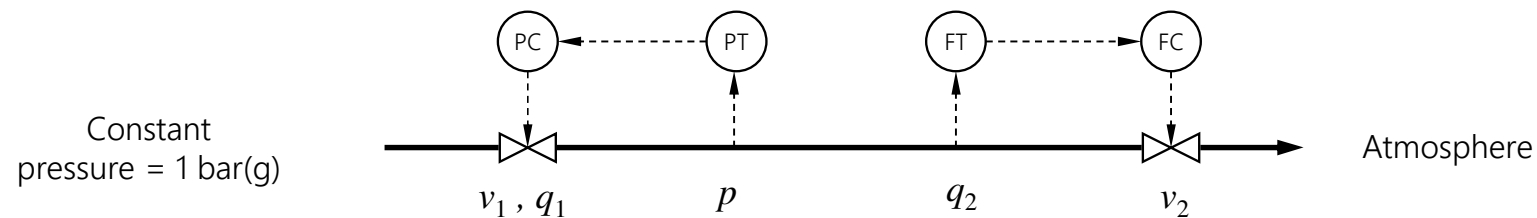
Tools and data sources

- The most important tools in the optimization work is Matlab, Python and TrendMiner.
- Most important data comes from a process historian (IP.21).
- We have developed our own libraries for
 - Data collection from process historian
 - Data analysis
 - Simulation of controllers and control structures
 - Identification
 - Assessment of control performance



Representation of control structures: P&ID (piping and instrumentation diagram)

Simple example: gas pipe

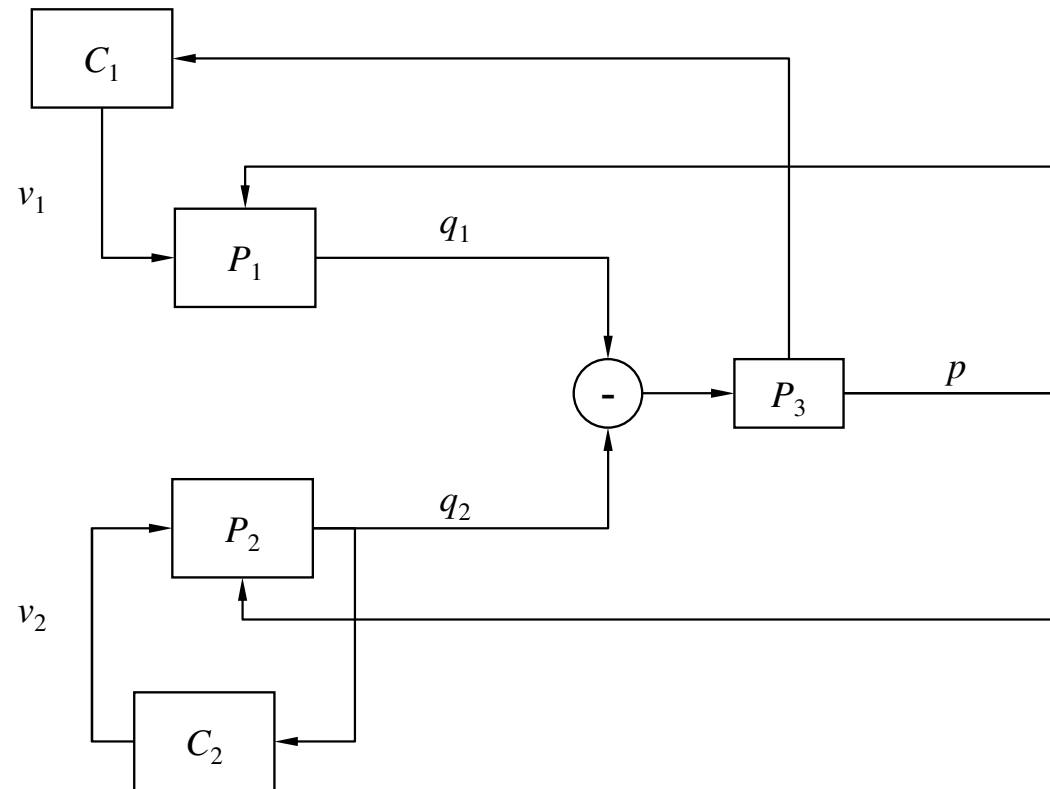


FC = flow controller, PC = pressure controller, LC = level ctrl, TC=temperature ctrl

FT = flow transmitter, etc

Why not use block diagrams, as in most control textbooks?

Block diagram for the same process



PID tuning

Practical PID control

- The four most important parameters in a PID-controller are
 - Gain
 - Integration time
 - PV filter time constant
 - Beta factor

$$u(s) = K_c \left\{ \beta r(s) - \frac{1}{1 + sT_f} y(s) + \frac{1}{sT_i} \left(r(s) - \frac{1}{1 + sT_f} y(s) \right) \right\}$$

- The PV filter prevents the controller from acting on noise that is not within control bandwidth anyway.
 - There is no other way to achieve this.
- Example from Perstorp plant: 182 PID-controllers
 - 101 of those have $T_f > 0$.
 - 6 of them have derivative action

Examples of "exotic" control parameters

- Anti-reset-windup parameters
 - ARW limits (hi and low)
 - ARW time constant
- Integration deadband
- Ramp rates (setpoint and controller output)
- Backcalculation parameters (cascade control)
 - What should happen to the master controller when the slave controller saturates
- Tracking options
- Alarms: high, low, highhigh, lowlow, deviation high, deviation low
- Which operating modes are allowed

Most common process model structures

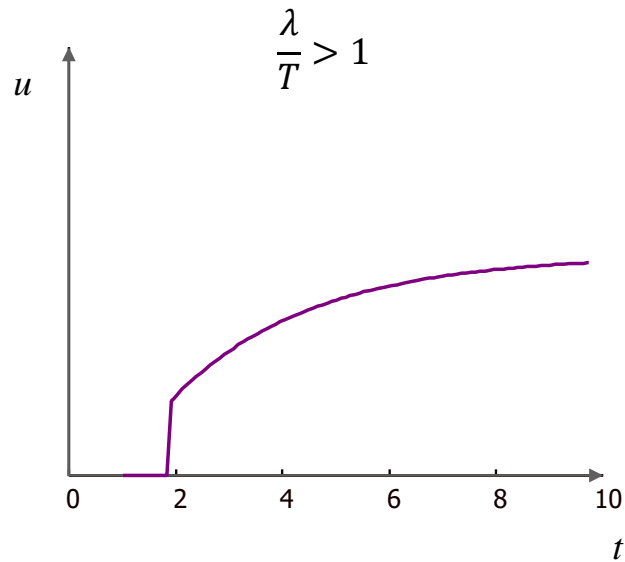
KLT	$\frac{Ke^{-sL}}{1 + sT}$	} ~80% of all loops
Integrating	$\frac{k_v e^{-sL}}{s(1 + sT)}$	
IPZ	$\frac{k_v(1 + sT_2)e^{-sL}}{s(1 + sT_1)}$	} ~15% of all loops
PPZ	$\frac{K(1 + sT_2)e^{-sL}}{(1 + sT_1)(1 + sT_3)}$	
P3	$\frac{Ke^{-sL}}{(1 + sT)^3}$	

The biggest challenge: model uncertainties

- Finding a decent process model (transfer functions etc) can be very hard
 - Trials are often needed
- Due to variation in process conditions we can expect process parameters to vary by up to a factor two.
- There are several thousands controllers, so we cannot monitor their performance individually.
- Too aggressive (unstable) control tuning causes much bigger problems than too defensive
- Conclusion: We need robust tuning.
 - Closed loop time constant often larger than process time constant (λ factor >1 , Ms around 1.2-1.4)

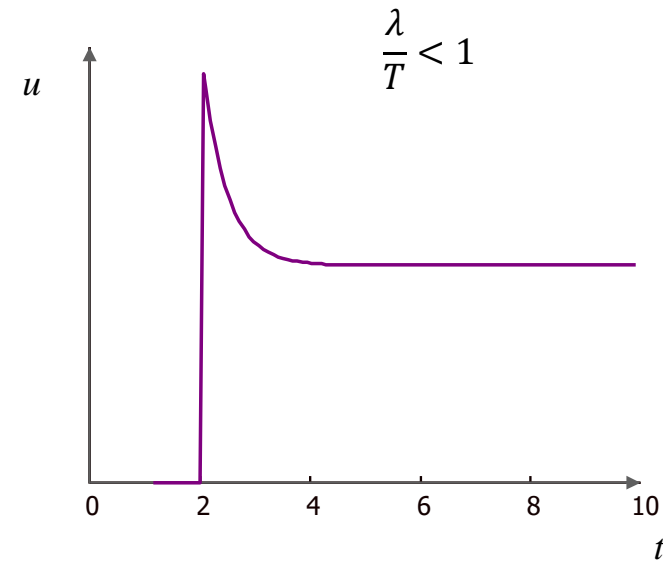
Controller output at setpoint change

Defensive tuning



"Lag action"

Aggressive tuning



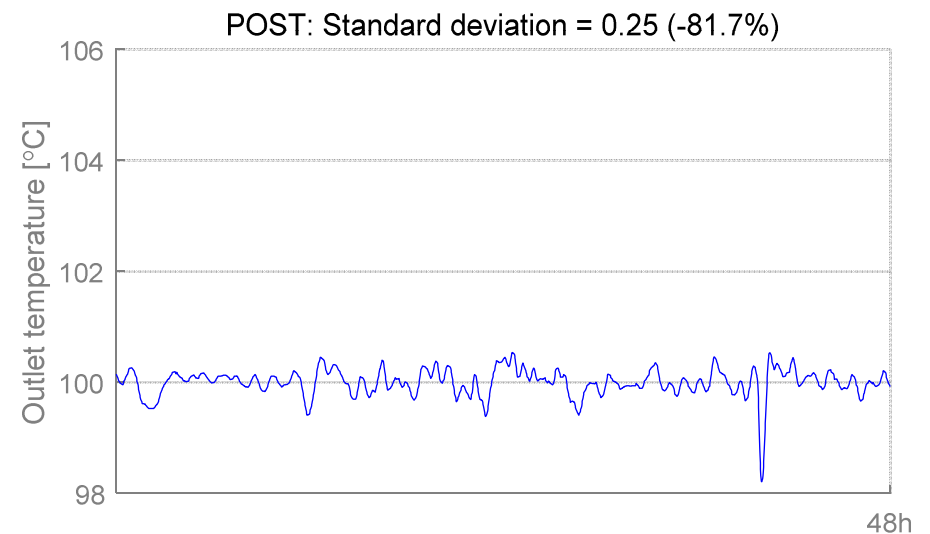
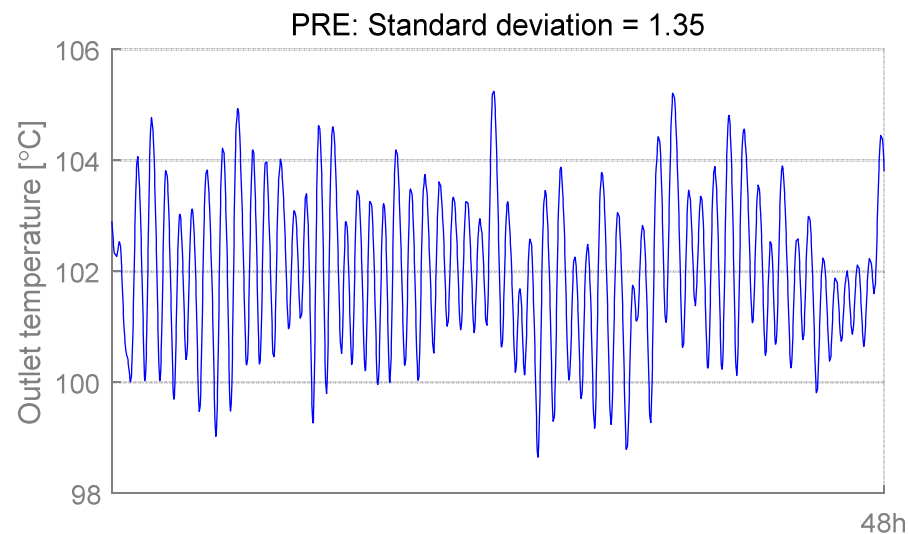
"Lead action"

PID tuning, done carefully, can take you a long way

- For a given process, there are many different structures that will all work.
- Often, choosing the right control structure is the best way to optimize operations.
 - (Topic for second half of the lecture)
- The drawback with that is that it requires reprogramming of the control system and training for operators.
 - DCS programmers is a scarce resource
 - Test run of the new paradigm is more challenging; DCS programmer needs to be available
- Surprisingly often, just tuning the PID controllers with great care can take you a long way

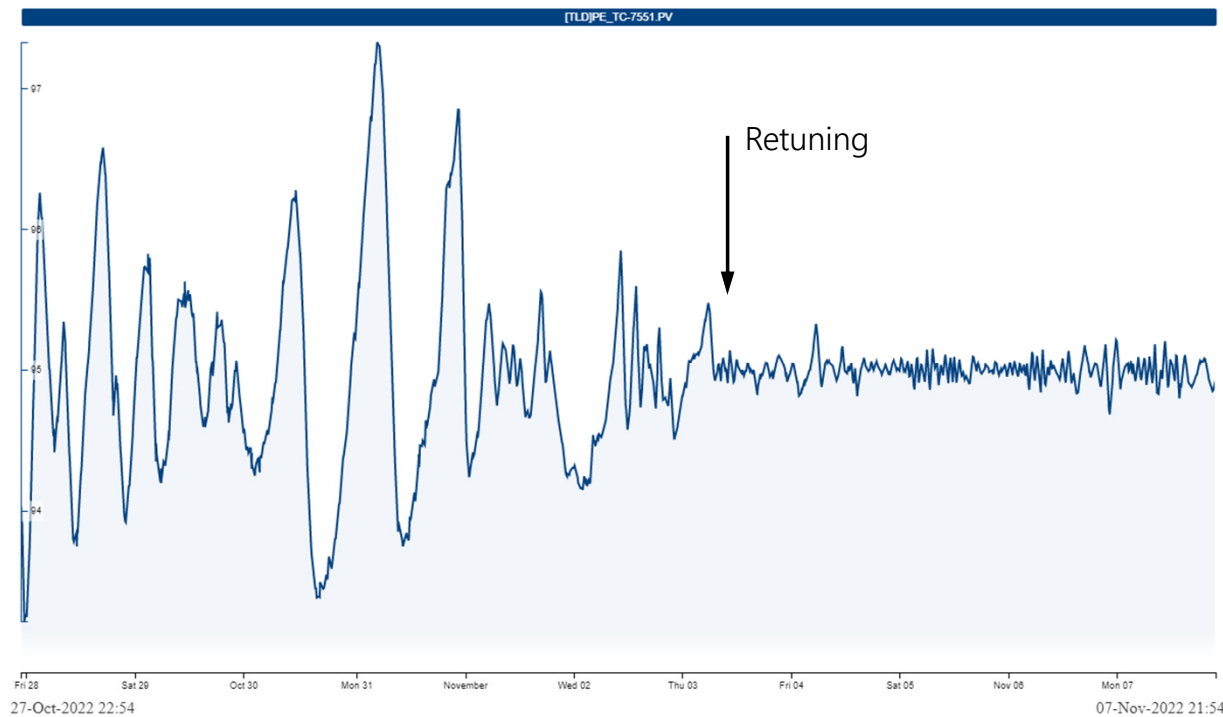
Example: Dryer temperature control

- Modeling on historical data, possible because the control output saturated intermittently
 - Integrating process, with time constant and deadtime
- Old controller parameters: Gain = 10, Integral time = 3600 s, Derivative time = 0, Feedforward gain = -10
- New parameters: Gain = 1.3, Integral time = 5300 s, Derivative time = 600 s, Feedforward gain = 0



Example: Feed vessel temperature control

- Model based on historical operations data: integration + dead time
- Controller gain changed from 3.33 to 70
- Integral time changed from 900 s to 2100 s



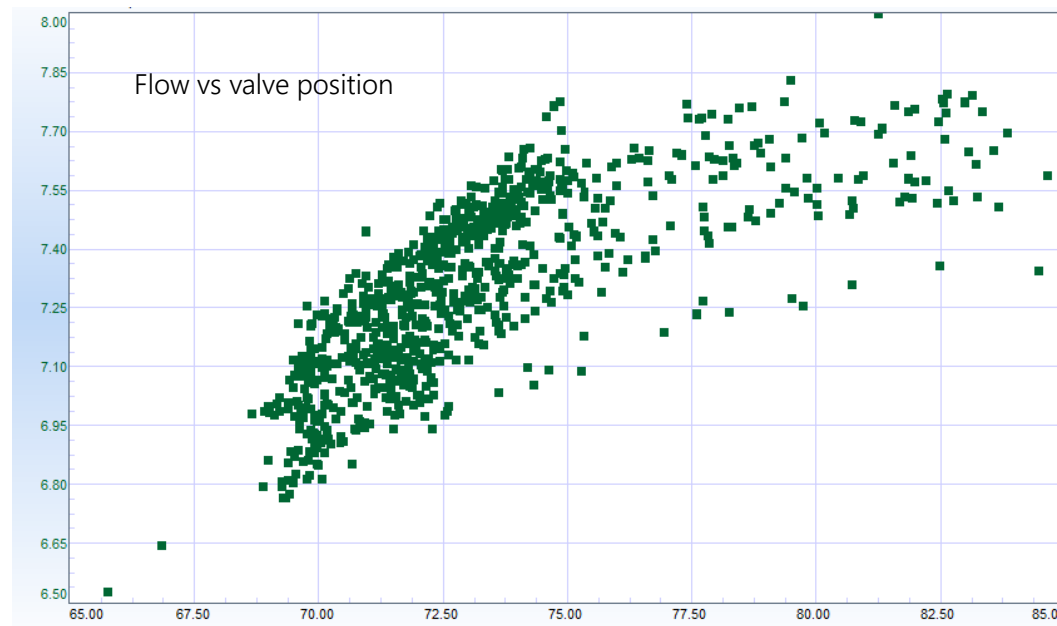


Basics

Feedback control and causality

Why we need process control

- Our processes are not perfect, and not perfectly predictable.
 - We need to compensate for disturbances



- If all process equipment, all raw material, all utilities etc were 100% known and predictable, we wouldn't need process control.
 - We wouldn't need any measurements either.

Some pitfalls in using process data for modeling

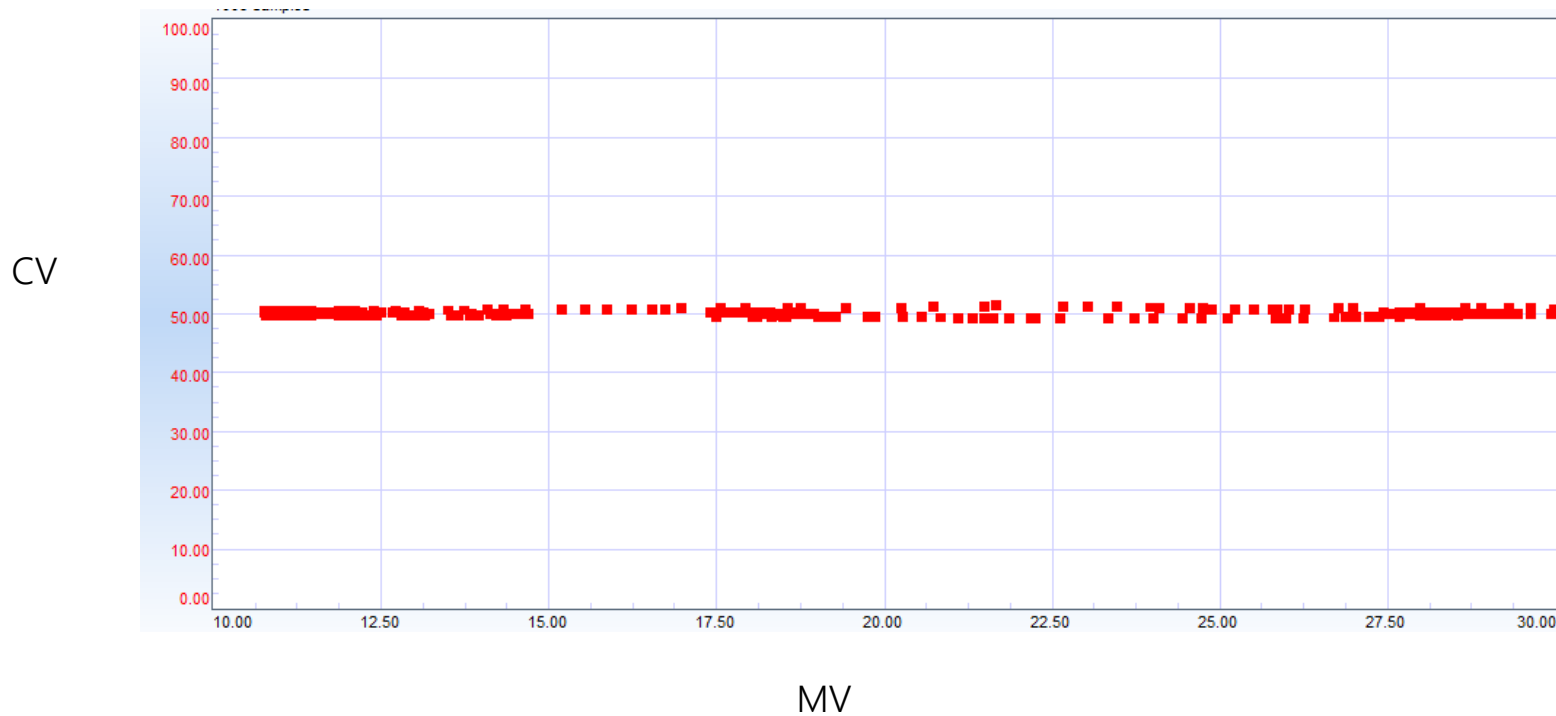
- When modeling a process, using historical data can be very tricky.
- There are some very common mistakes, explained by the examples below.
- The common theme is that feedback control “reverses causality”
 - In the process itself (open loop) the MV is the independent variable and the CV is the dependent variable.
 - For a feedback controlled process, the CV is the independent variable, and the MV is the dependent variable.

MV = manipulated variable (u)

CV = controlled variable (y)

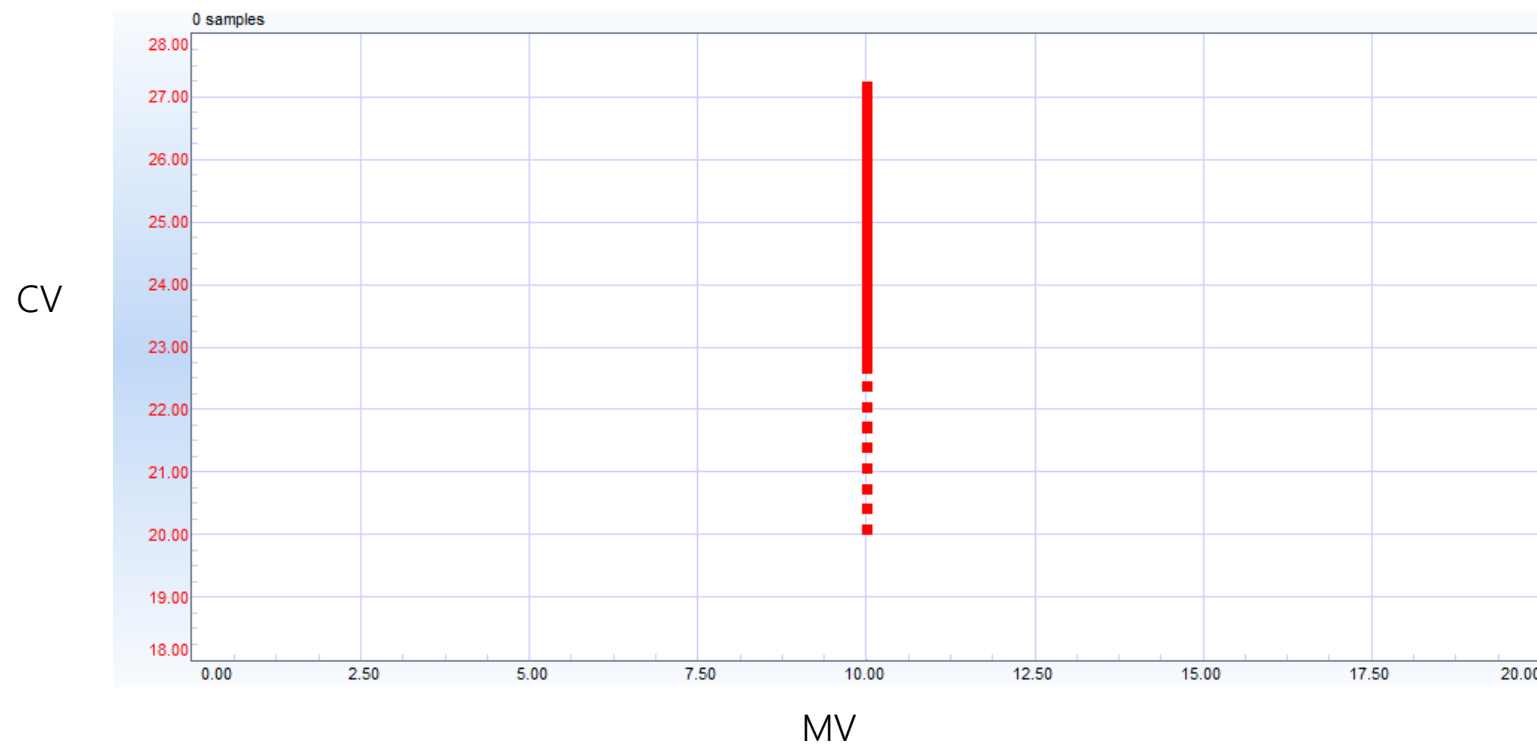
MV-CV plot; Example 1; Level controller

What can you say about the process, based on the data below?



MV-CV plot; Example 2; Temp controller

What can you say about the process, based on the data below?

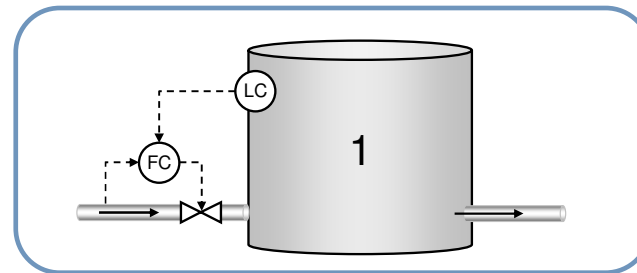
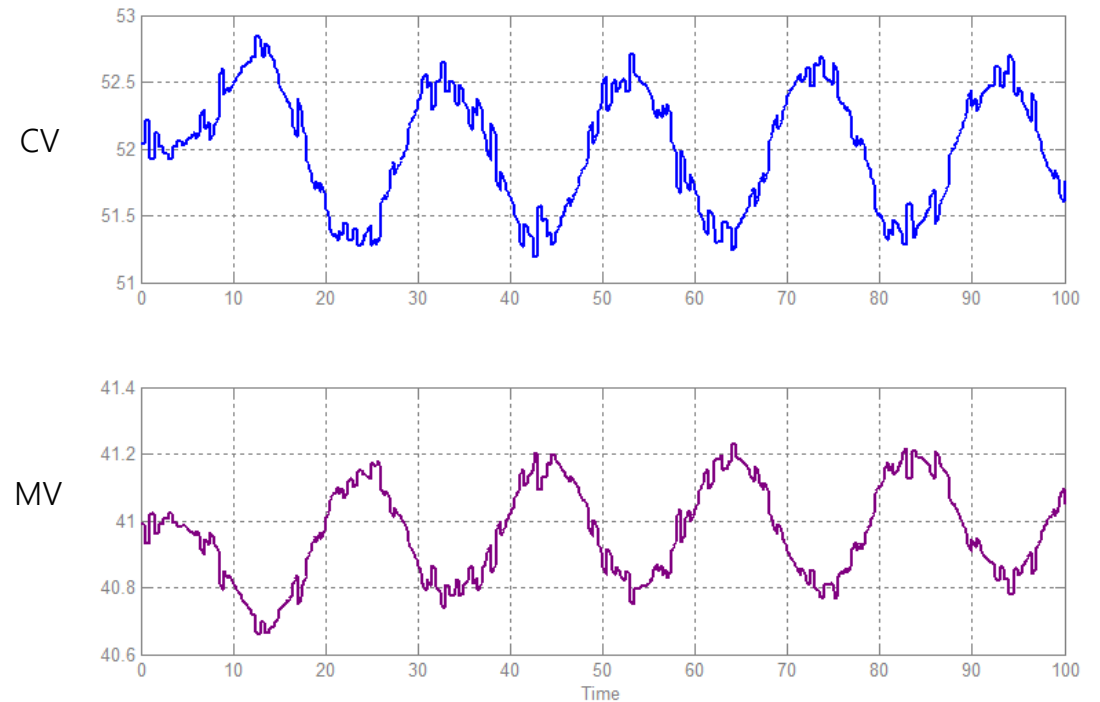


Determine MV action; Example 1

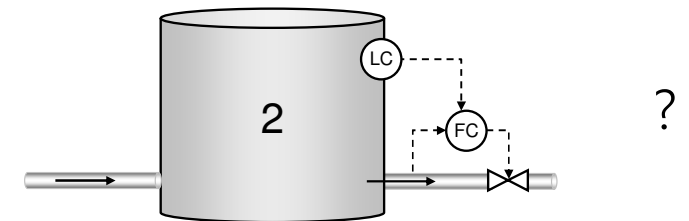
Data from a level control loop in Auto.

Determine if the controller manipulates the upstream or the downstream flow.

Setpoint is constant = 52.



OR



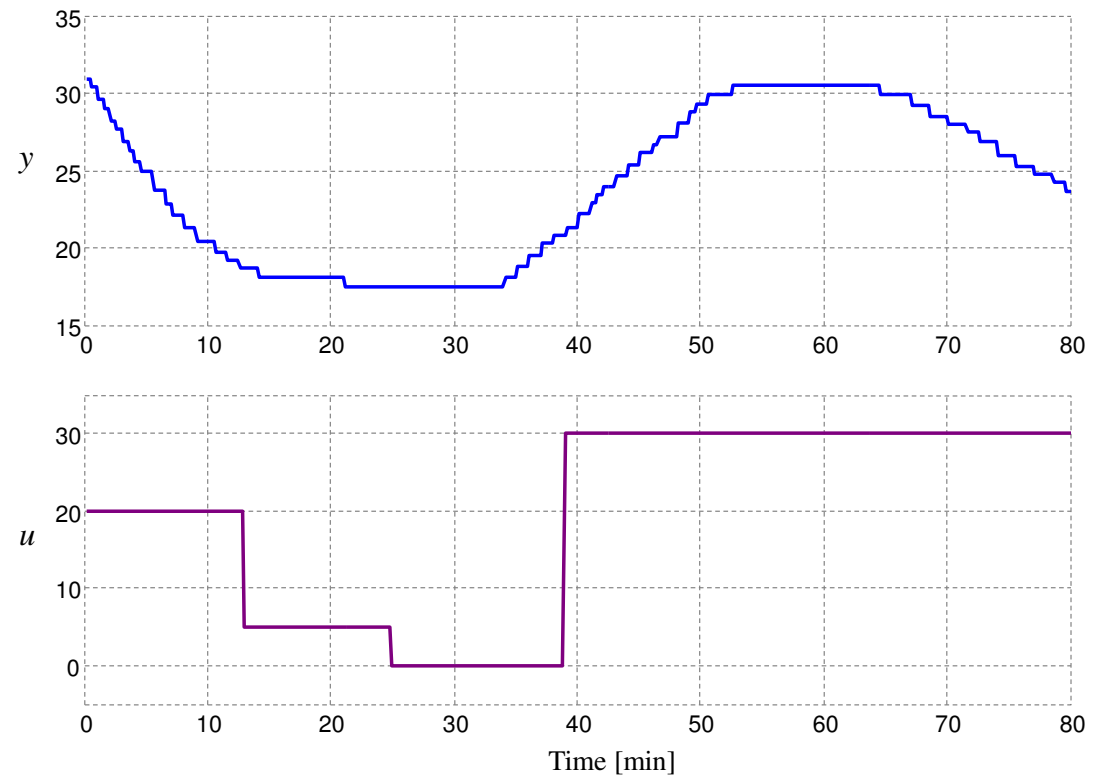
?

Careful when modeling on closed loop data

- Conclusion from these examples:
Normally, **you can't use historical data from normal operations to model the process.**
 - Not even with advanced methods like anything from machine learning
- Almost all variable dependencies known to plant staff and engineers are already used to some extent for controlling the process.
- You will not find the relevant correlations.
- You will find irrelevant correlations.

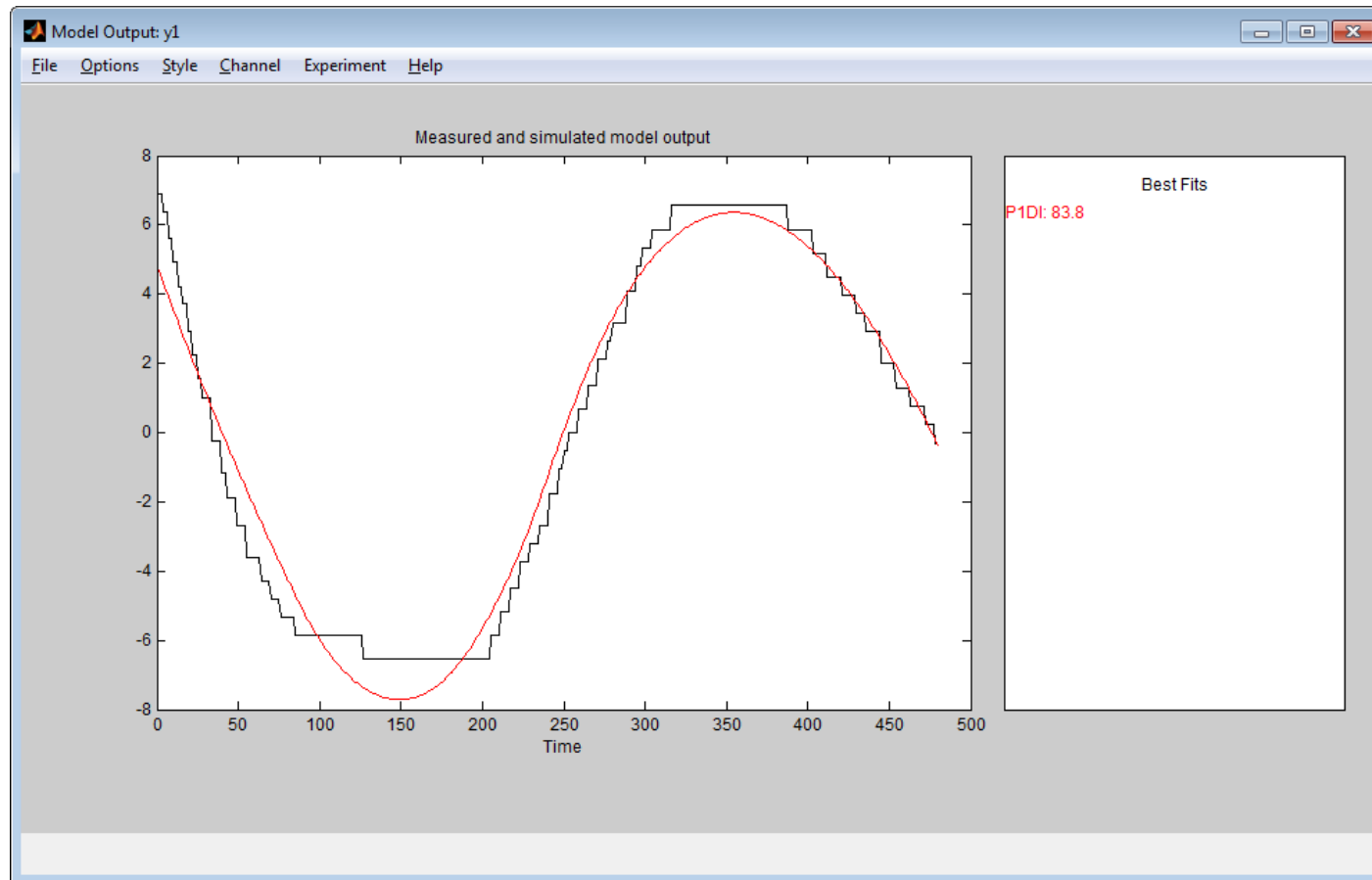
Real world example: is there a causal effect?

Based only on this data, can you tell if y can be controlled by manipulating u ?



$\frac{k_v}{s(1+sT)}e^{-sL}$ provides a very good model structure

Model fit in Matlab

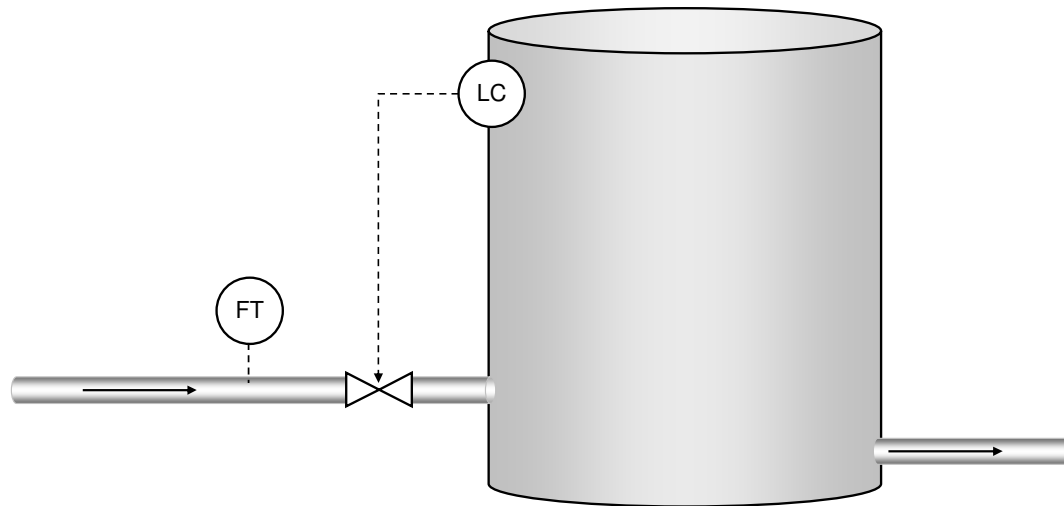




Control structures

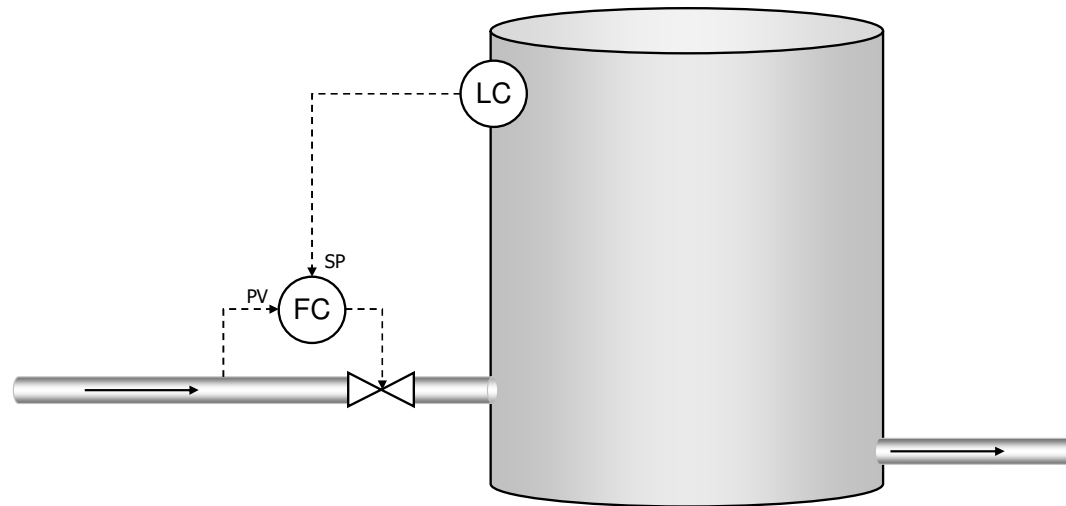
Some practical examples

Ex: Level control with improvement opportunity



- The level in the tank varies too much, because there are pressure variations in the line for the incoming flow.
- We can't tune the controller more aggressively - then it becomes unstable.
- Can we still improve control performance?

Solution: Control the flow too



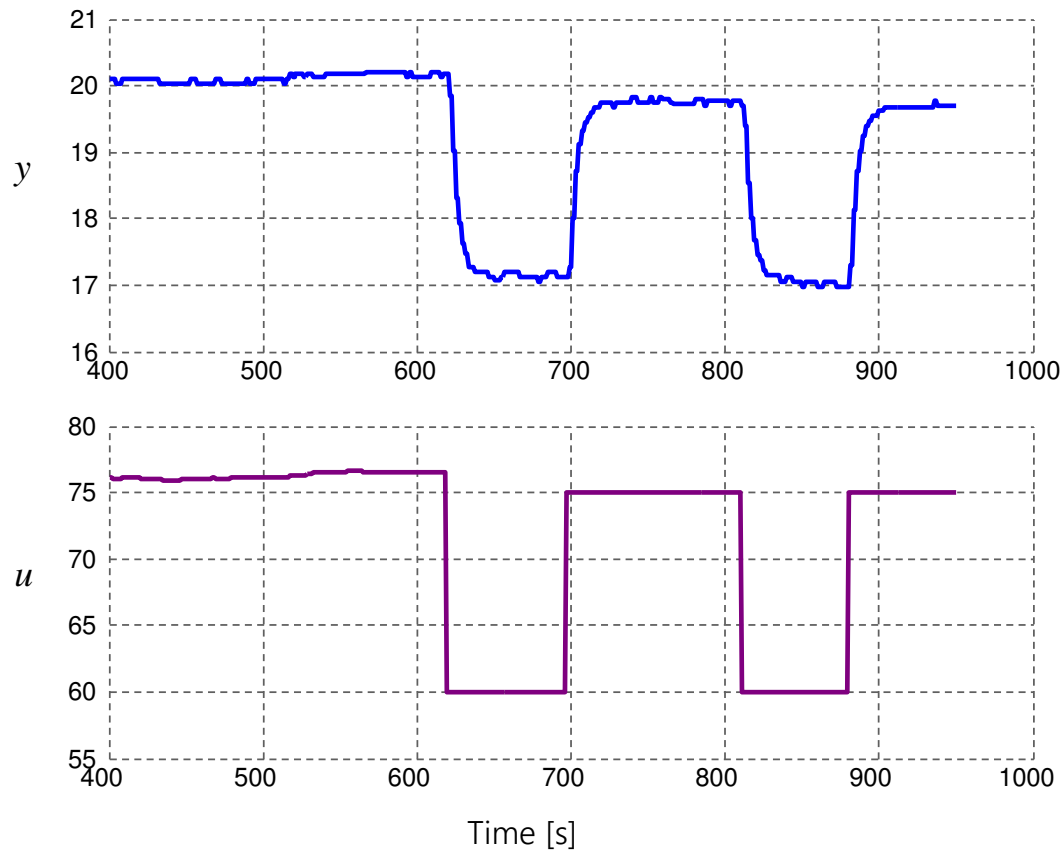
= Cascade control

Important: Motivate by disturbances

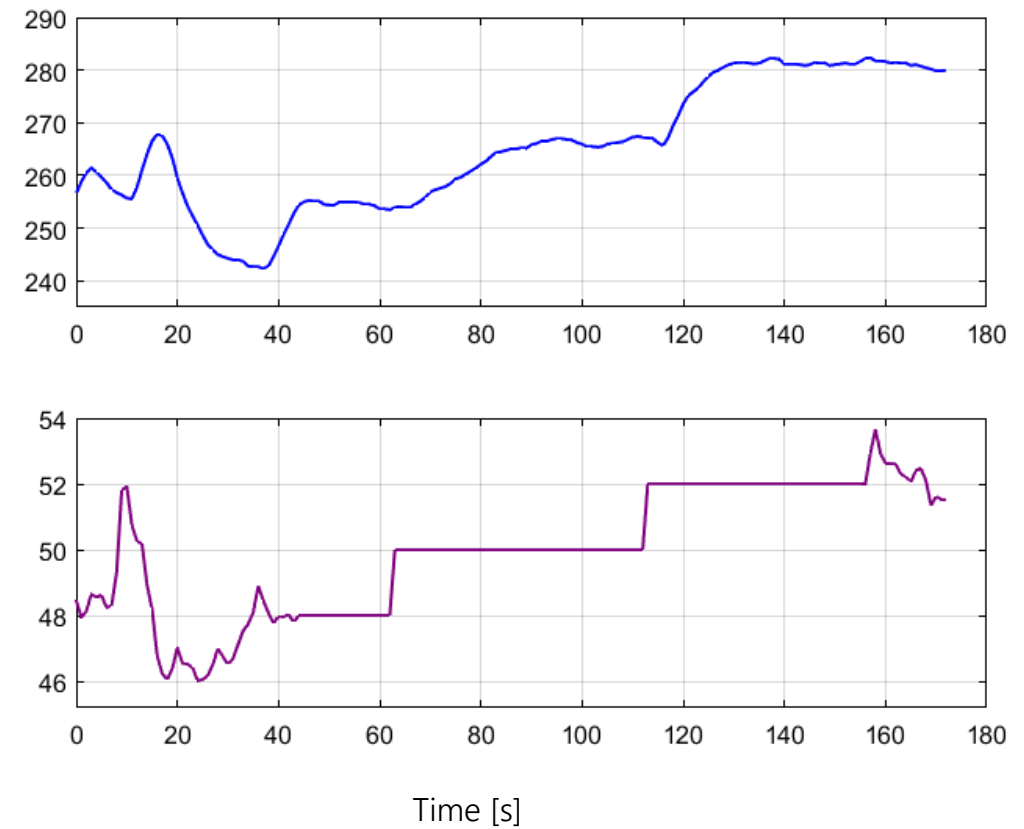
- For any control structure, the argument for using it should be based on which disturbances you have.
- In the case above:
 - If level variations were not due to varying feed pressure, then cascade control may not be the right solution.
- This holds for all process control:
 - The reason for using feedback control is disturbances.
- Process control deals with **real** plants.
- Example: A volumetric pump normally delivers a flow that is an exact, repeatable function of its input signal (frequency or current).
 - In that case, you should not use feedback flow control.

Flow step responses

Volumetric pump: Completely repeatable

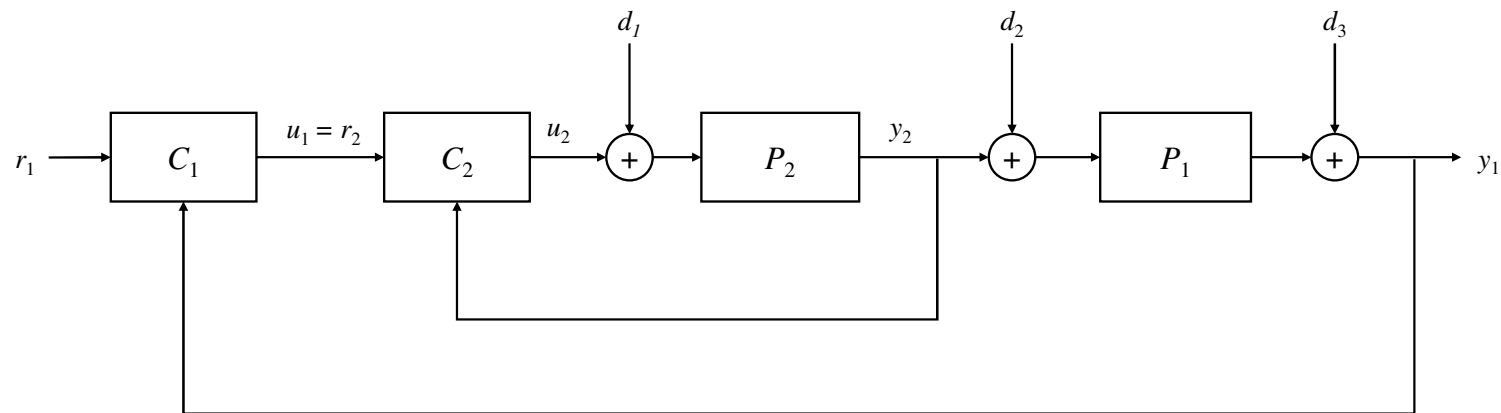


Centrifugal pump + control valve: Not repeatable



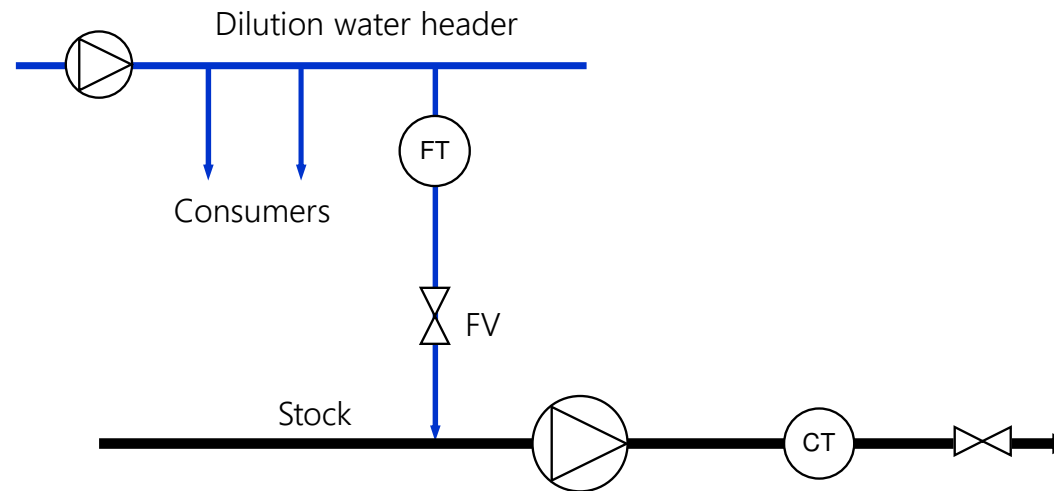
Cascade control block diagram

- Which disturbances motivate the use of cascade control?



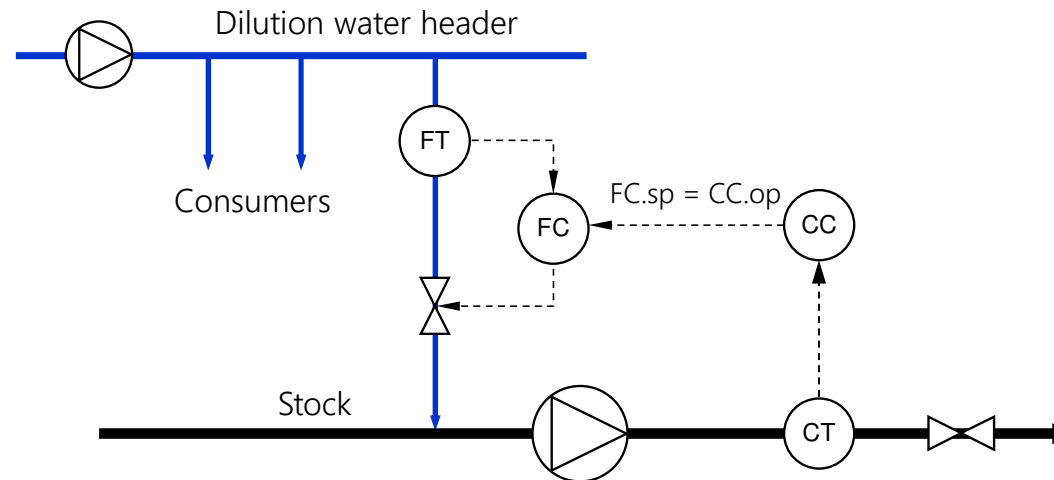
Answer: d_1

Example: Dilution process



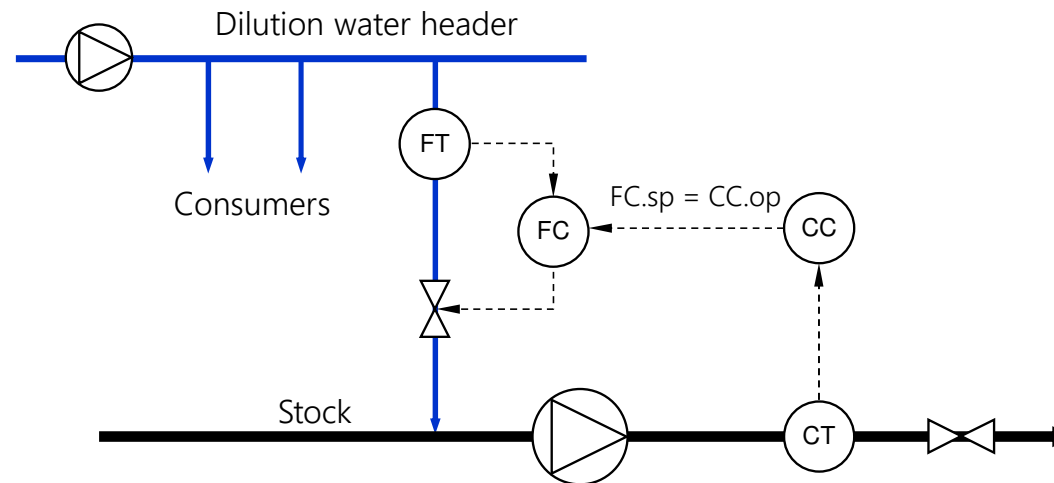
- Task: Control the concentration measured by the transmitter CT, by manipulating dilution water valve FV.
- How would you do that?

Structure for concentration control



- Concentration (consistency) control in cascade against dilution water flow.
- Questions:
 - Which disturbances/variations motivates the use of an FC?
 - Which disturbances/variations motivates the use of a CC?

Concentration control; Scenario analysis



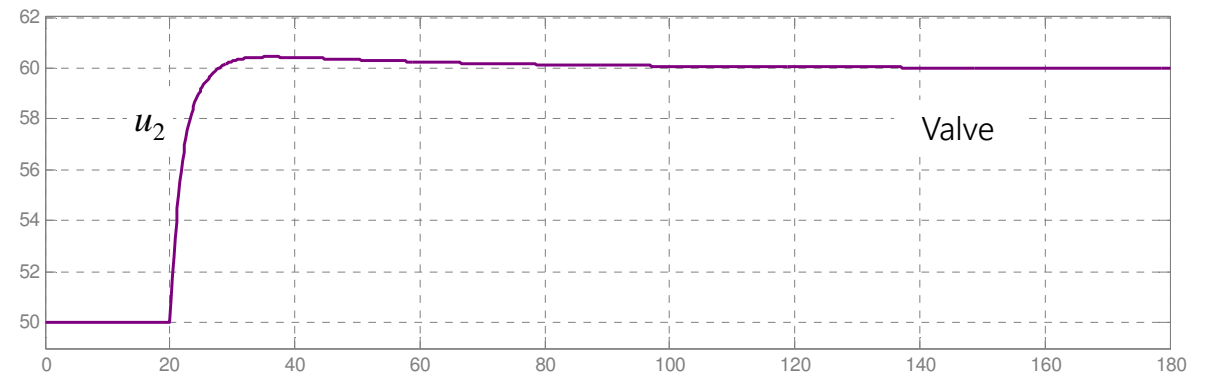
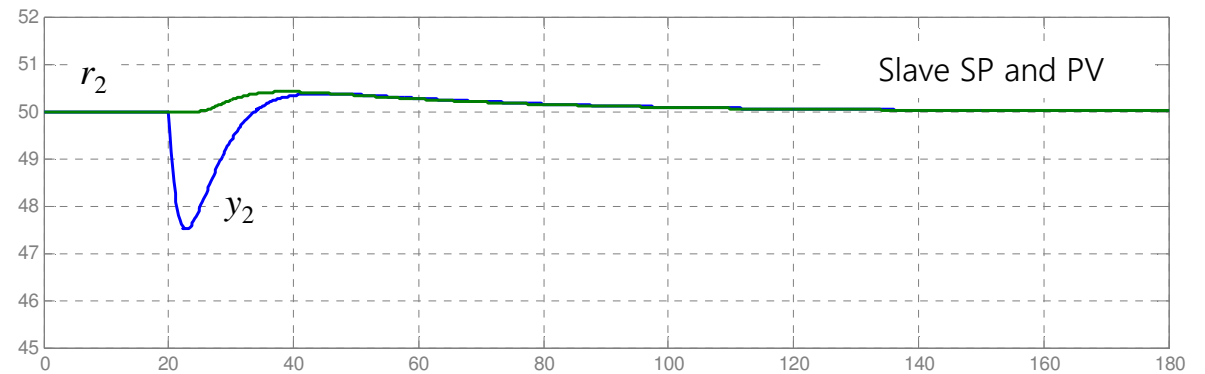
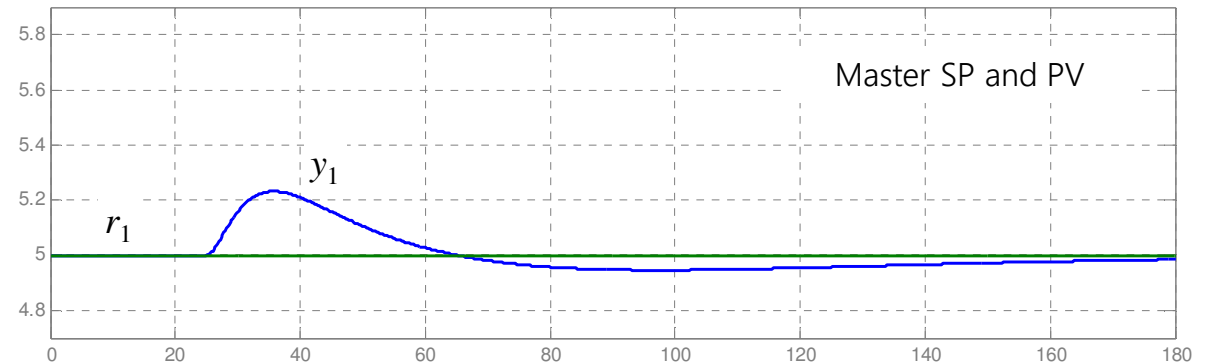
Scenario: There is a pressure drop in the dilution water header, because one of the other consumers suddenly increases its demand.

- We will now see how the cascade controller reacts to this, depending on the tuning of master and slave controller.

Slave controller much faster
than master.

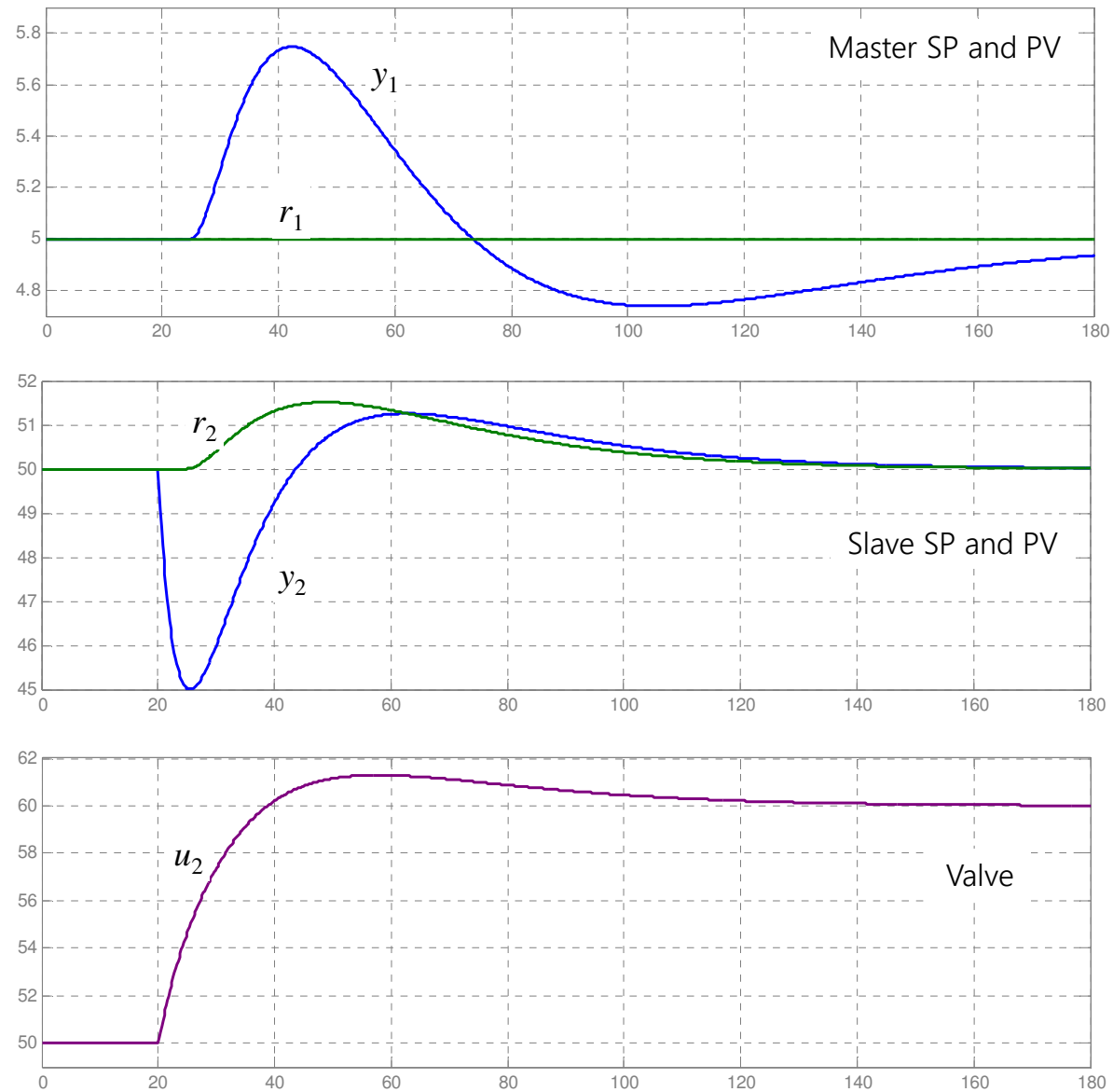
Rule of thumb for cascade control to work:
Time constant in slave
5-10 times faster than in master.

The disturbance is handled by
the slave controller before the master
controller reacts (in principle).



Difference between slave and master smaller than in previous slide.

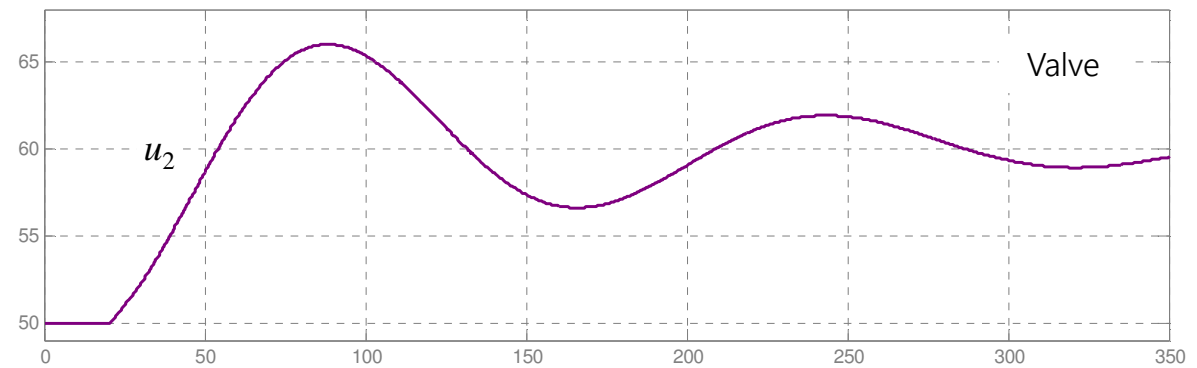
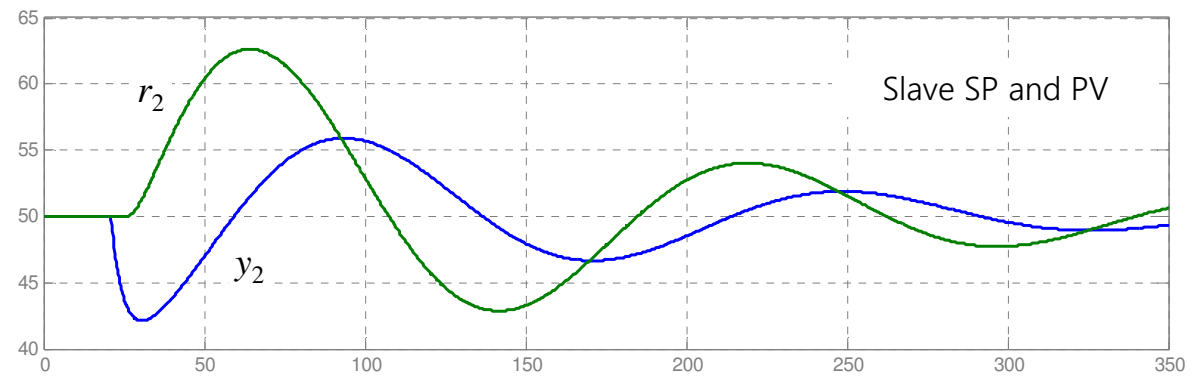
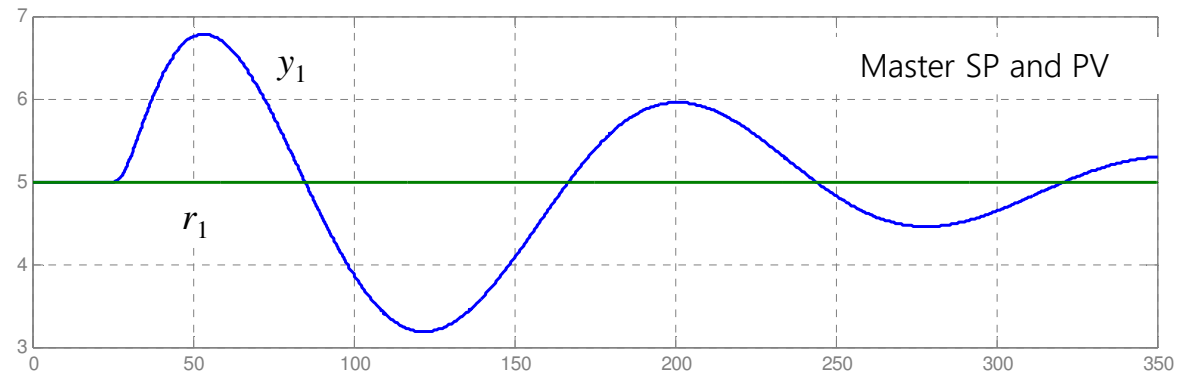
The disturbance is handled by the slave controller but the master controller also reacts, later on.



Master and slave controller almost equally fast.

The disturbance is thrown back and forth between master and slave.

Almost unstable.

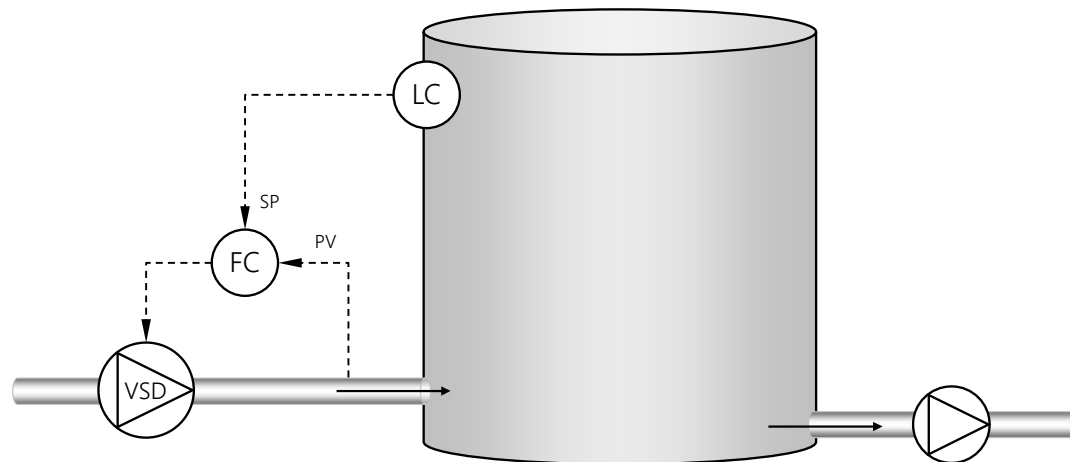


Cascade control application example Level

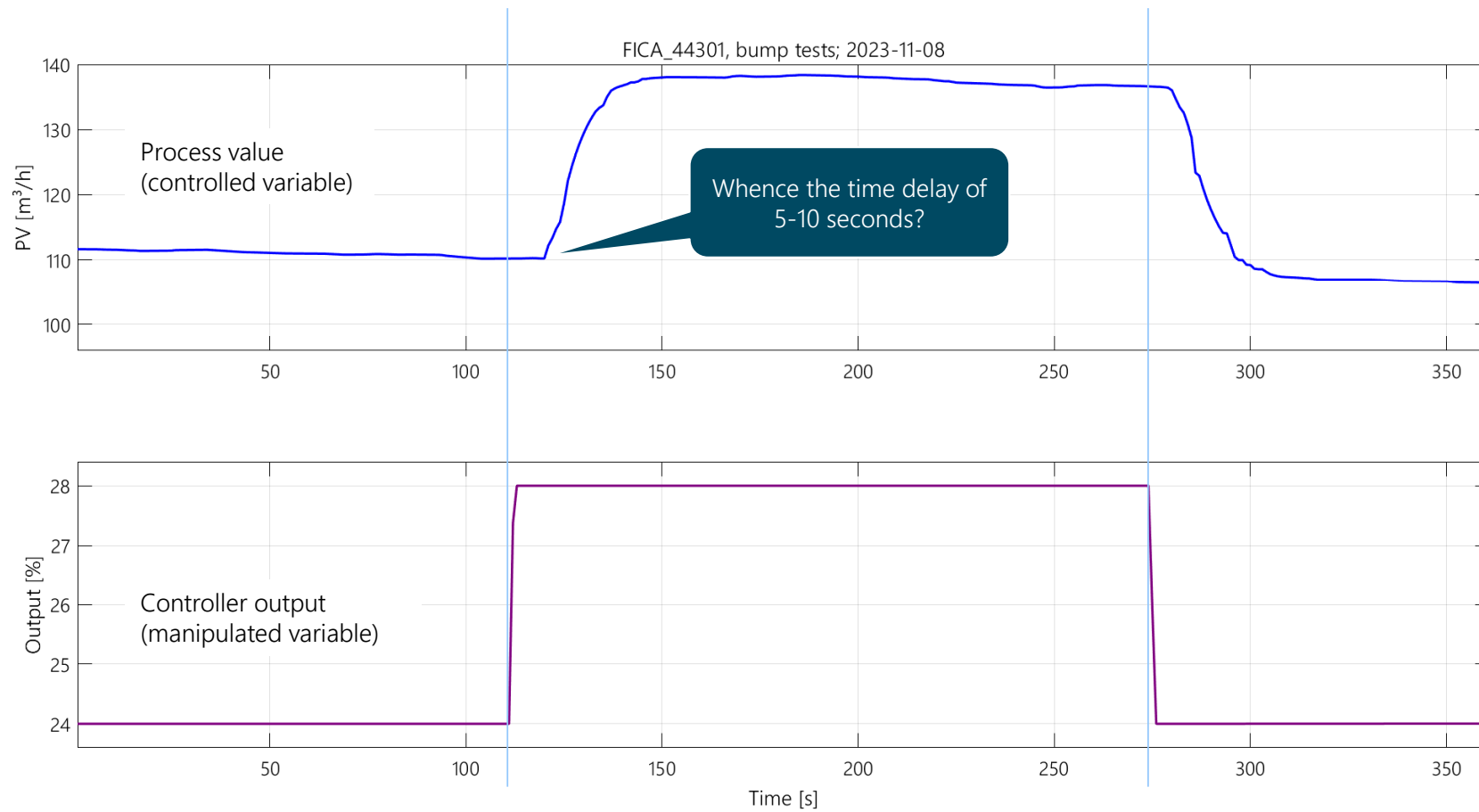
Background

- Control the level in a tank, by manipulating the feed to the tank, in cascade.
- Flow is controlled by manipulating a pump with variable speed drive, not a control valve

(This example is part of a bidirectional control example, coming in later slides.)

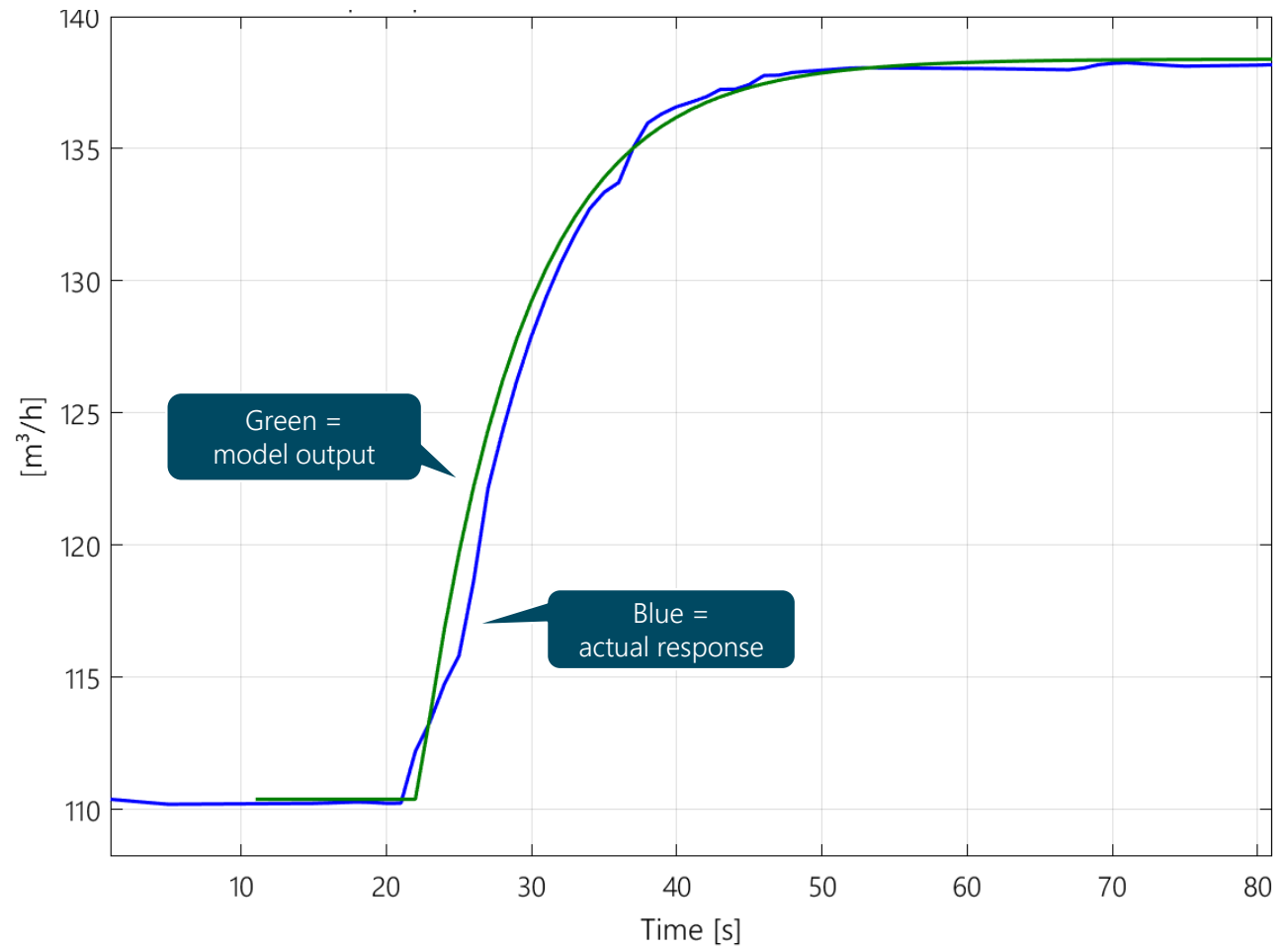


Bump tests for modeling the flow process



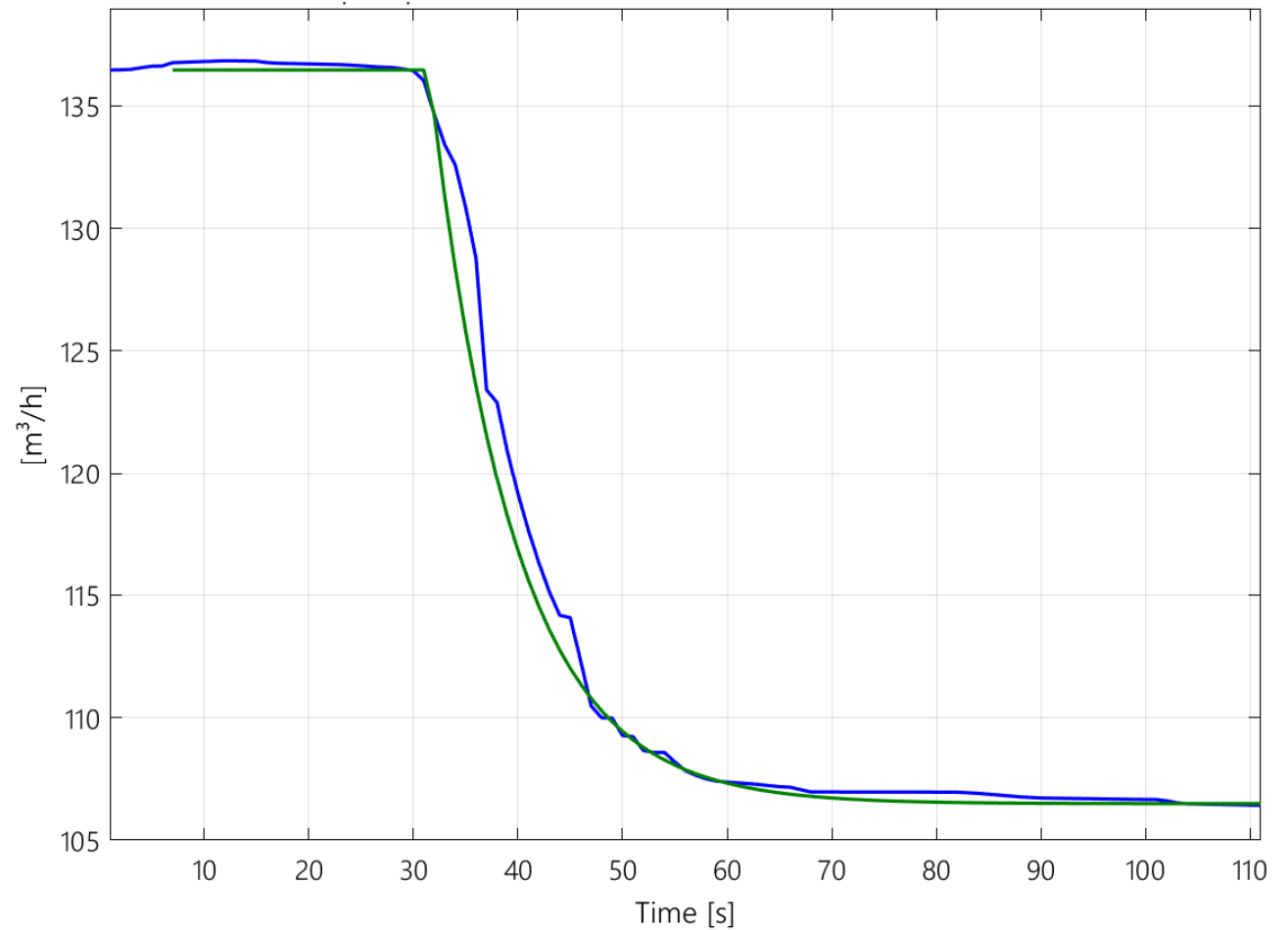
Process model for the first step response

- Process gain:
 - Engineering units: 7
 - Scaled: 2.19
- Time constant: 7 s
- Dead time: 10 s

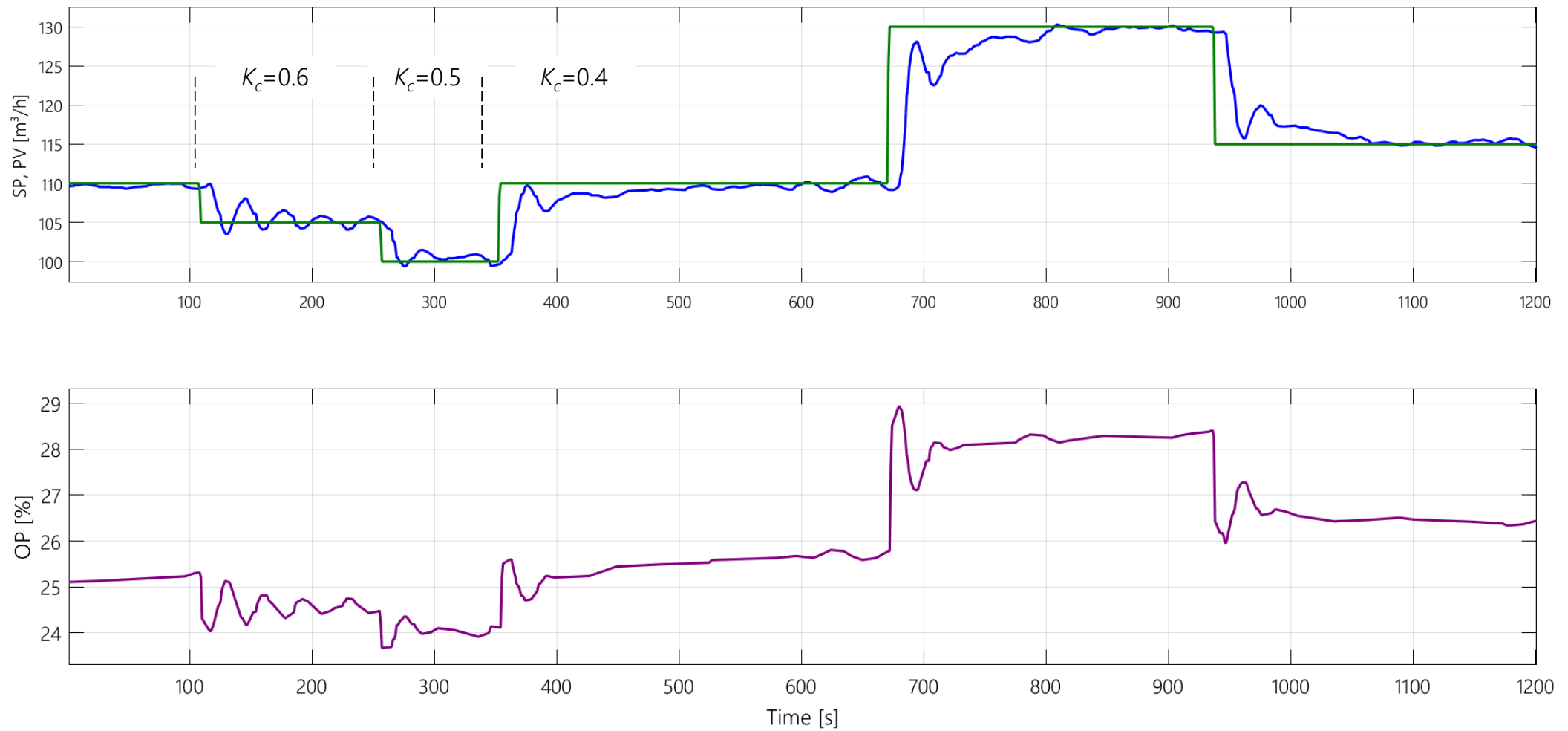


Process model: second step response

- Process gain:
 - In engineering units: 7.5
 - Scaled: 2.34
- Time constant: 8 s
- Dead time: 6 s

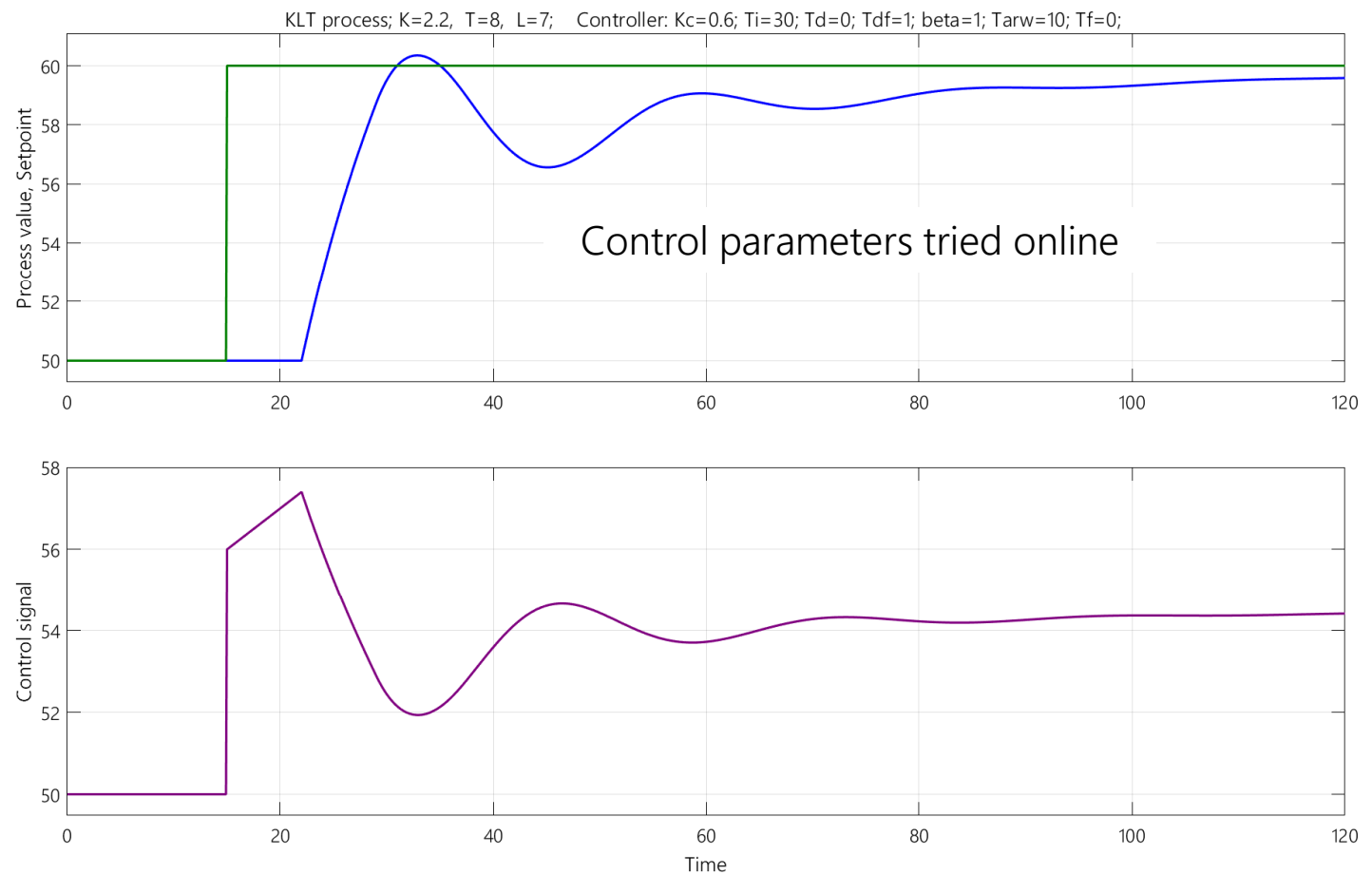


Closed loop behavior for different ctrl gains



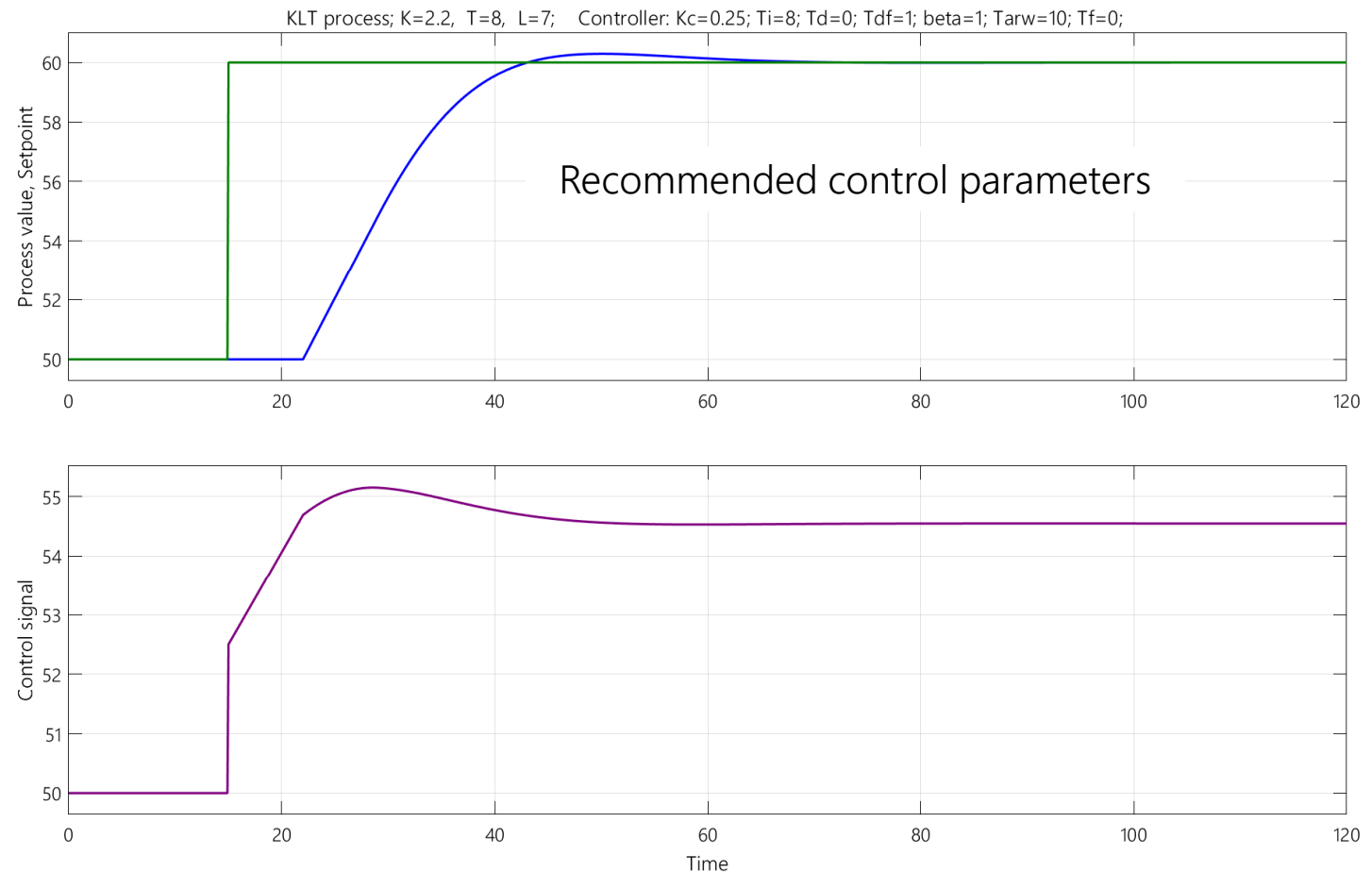
Simulated closed loop behavior

- These control parameters are not good.
- The integral time is too long and the gain is too high.
- The closed loop simulation confirms that the process model is good, though.

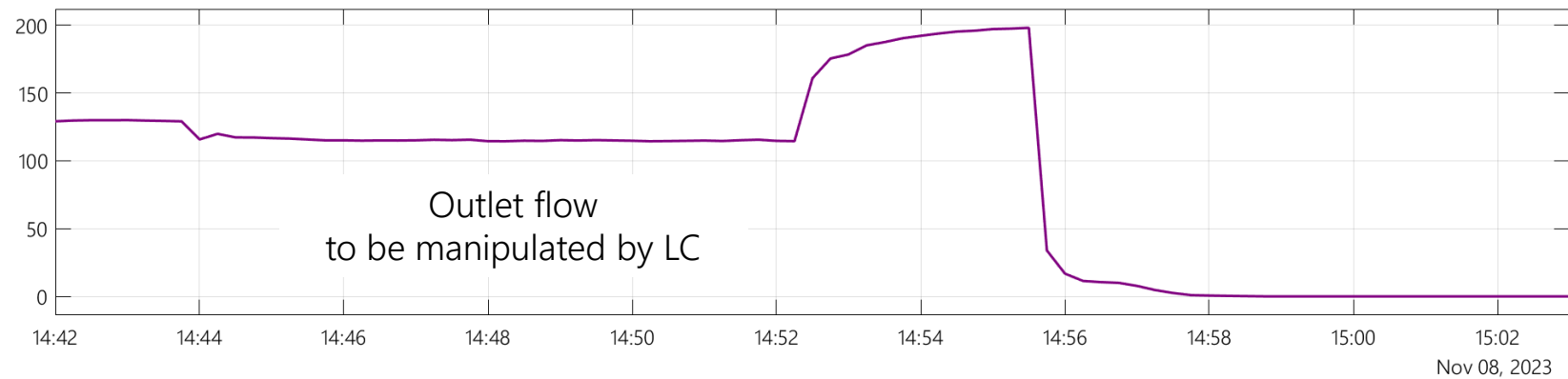
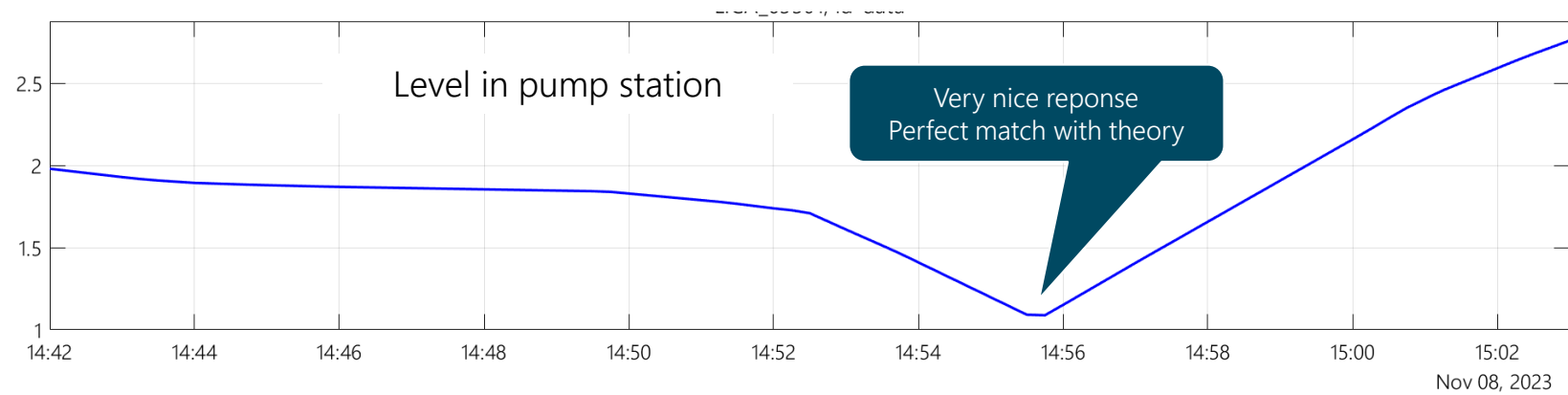


Recommended control parameters: simulation

- Tuning philosophy:
Long process deadtime means we cannot use aggressive tuning. However, this will be a slave controller, means that we cannot have a sluggish response.
- Compromise. Desired closed loop time constant: 8 s
- Lambda tuning
(= SIMC rule, in this case.)
Integral time = process time constant

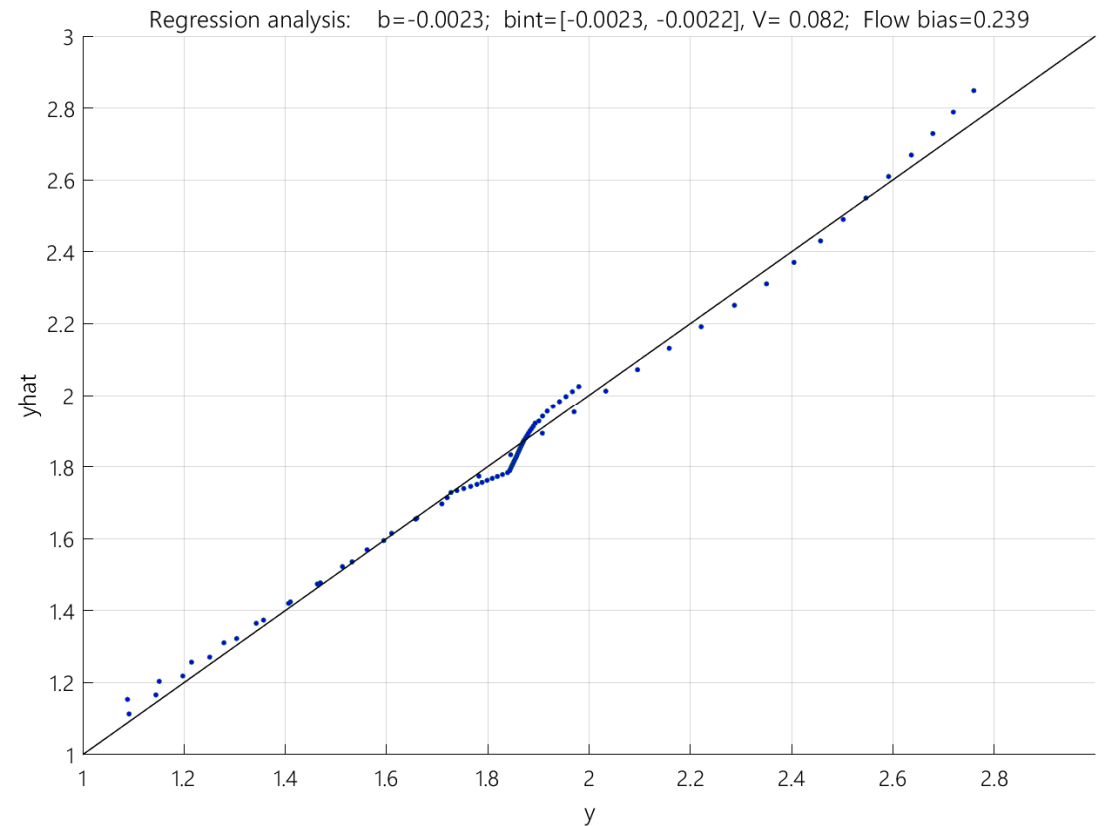


Level controller open loop bump tests



Modeling an integrating process (level)

- In most cases the only dynamics is an integration.
- The only thing to estimate in the process model is the coefficient to PV from integral of MV.
- Thus: a linear regression problem.
- To the right: estimated level vs actual level, for the data in the previous slide.



Simulation for the entire system

Process 1 (master)

Process type: Integrating

K1

0.23

T1

0

L1

0.1

Master controller

Kc1

1.7

+

-

Ti1

10

+

-

Td1

0

+

-

Tdf1

5

+

-

β_1

1

+

-

Tarw1

5

+

-

Tf1

0

+

-

Process 2 (slave)

Process type: KLT

K2

2.2

T2

0.13

L2

0.11

Slave controller

Kc2

0.25

+

-

Ti2

0.13

+

-

Td2

0

+

-

Tdf2

5

+

-

β_2

1

+

-

Tarw2

2

+

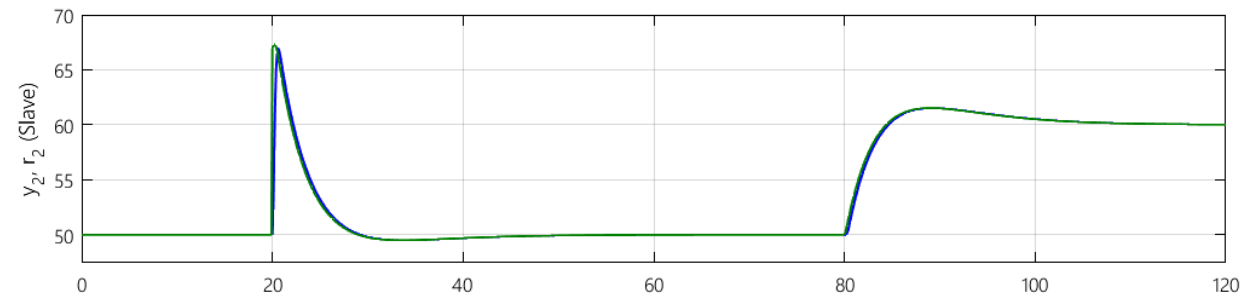
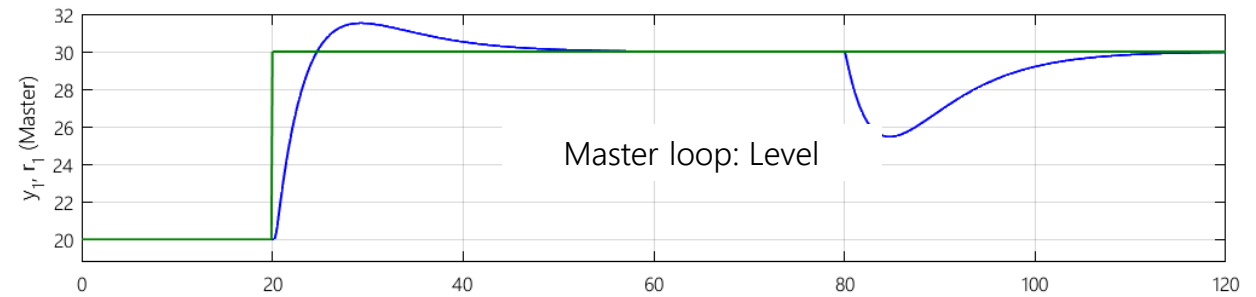
-

Tf2

0

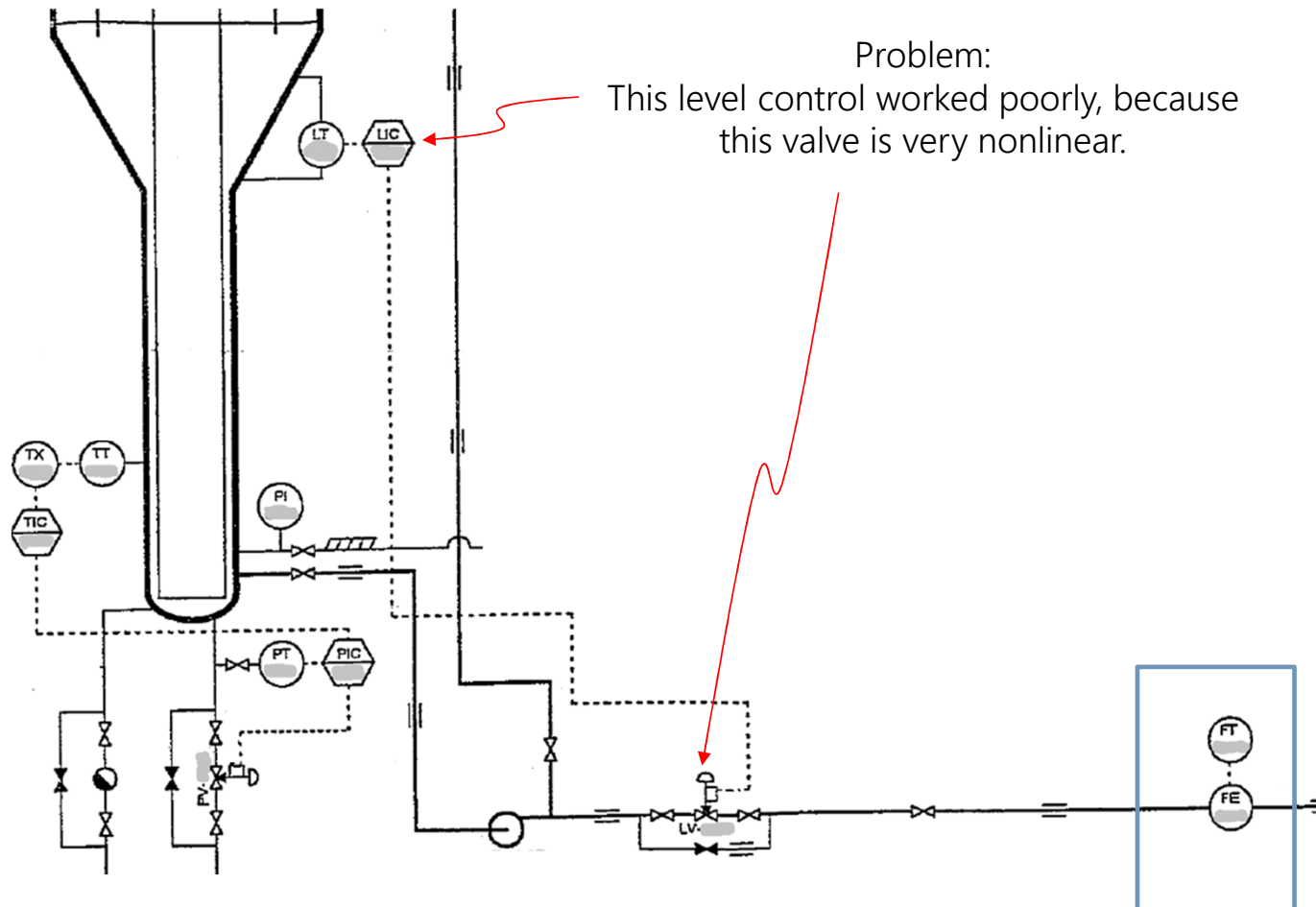
+

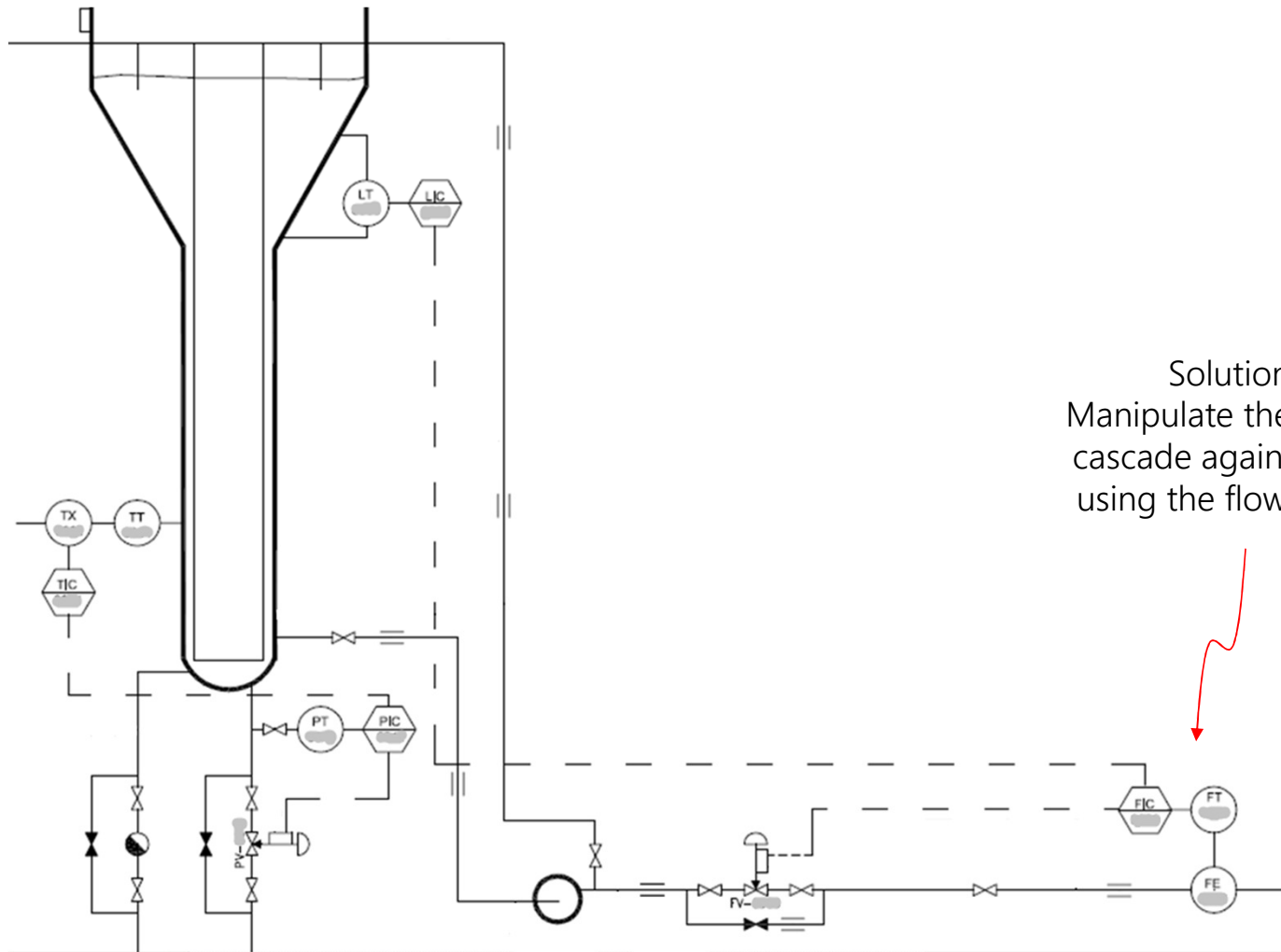
-



New example

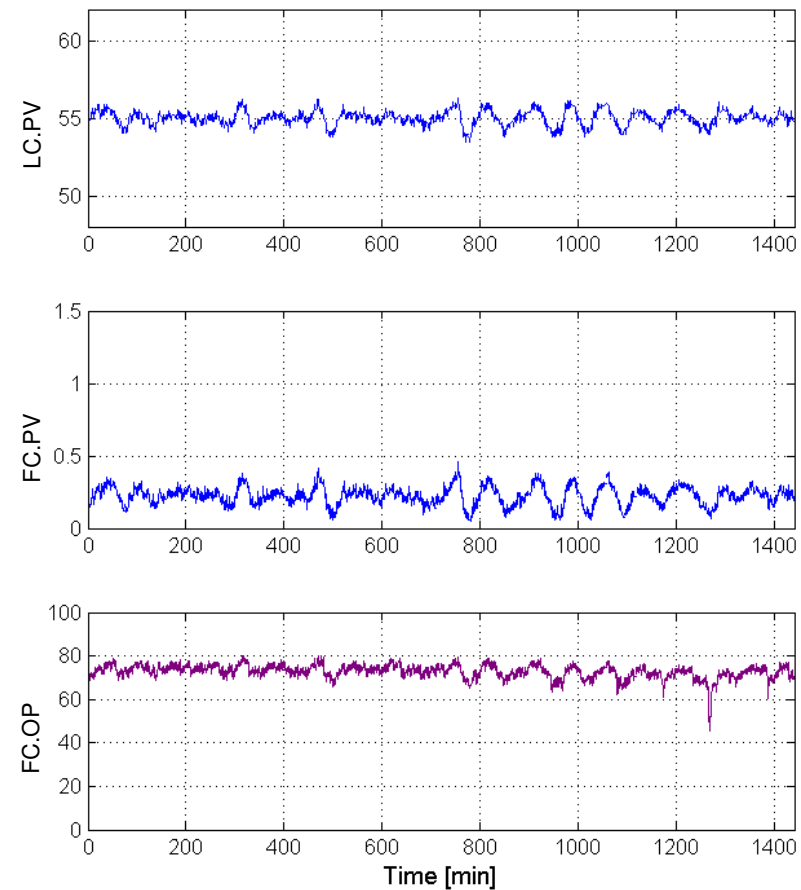
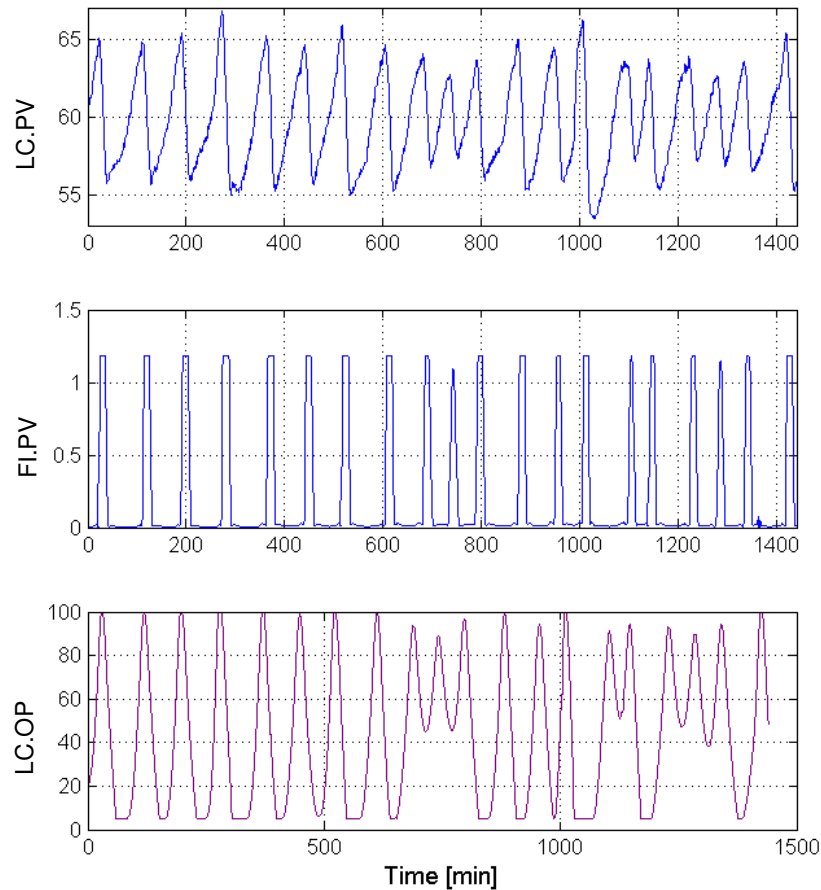
Evaporator with poor level control





Solution:
Manipulate the flow in
cascade against level,
using the flow meter.

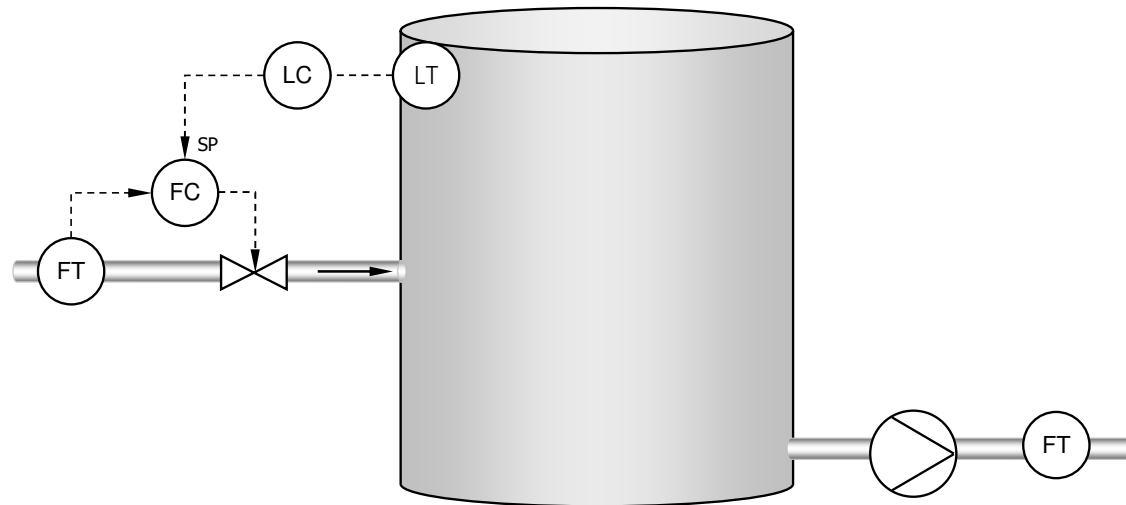
More stable level and smoother flow using cascade control



In this case, cascade control is used to linearize the slave process, rather than for disturbance rejection.

Ex: Level control with improvement opportunity

- The level in the tank varies too much because of variations in output flow.
- We can't tune the controller more aggressively - then it becomes unstable.
- How can we improve control performance?

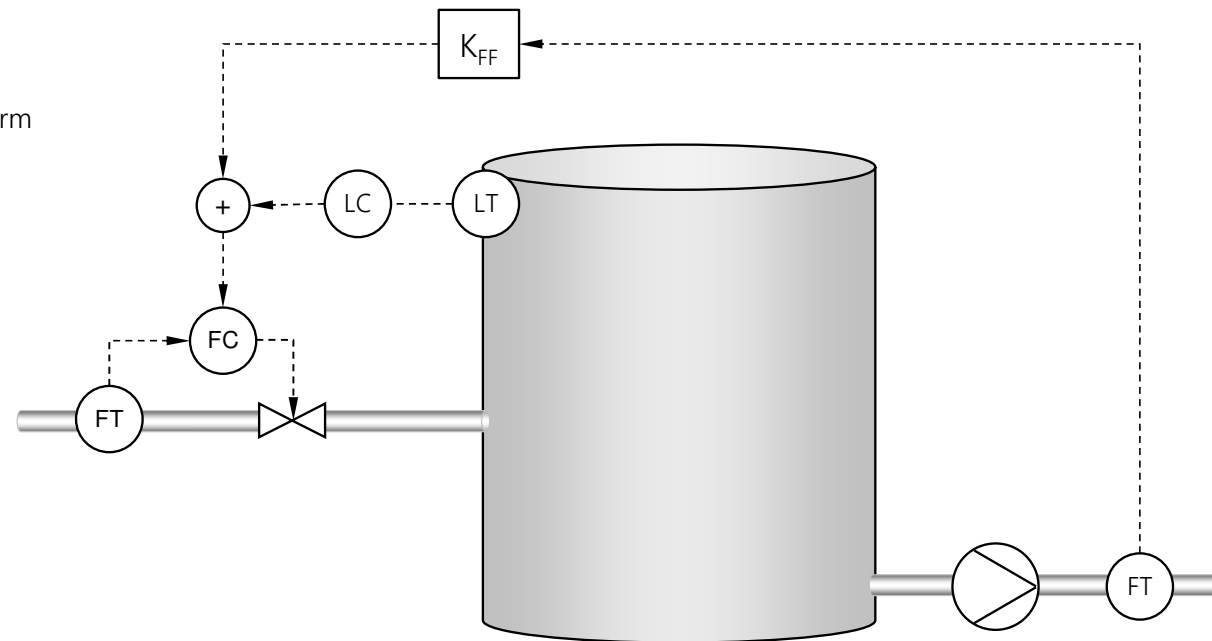


Feedforward: Give early information to the controller

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int_0^t e \right) + u_{FF}$$

Feedback term

Feedforward term

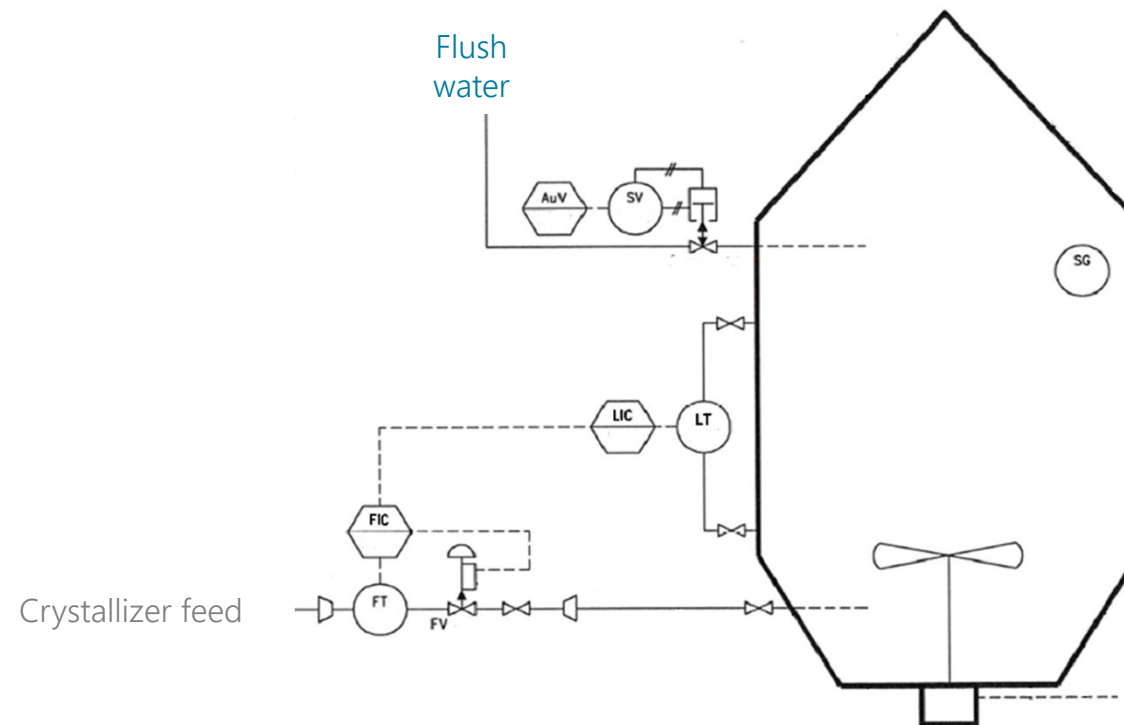


New example

Crystallizer: Level disturbance from wash sequence

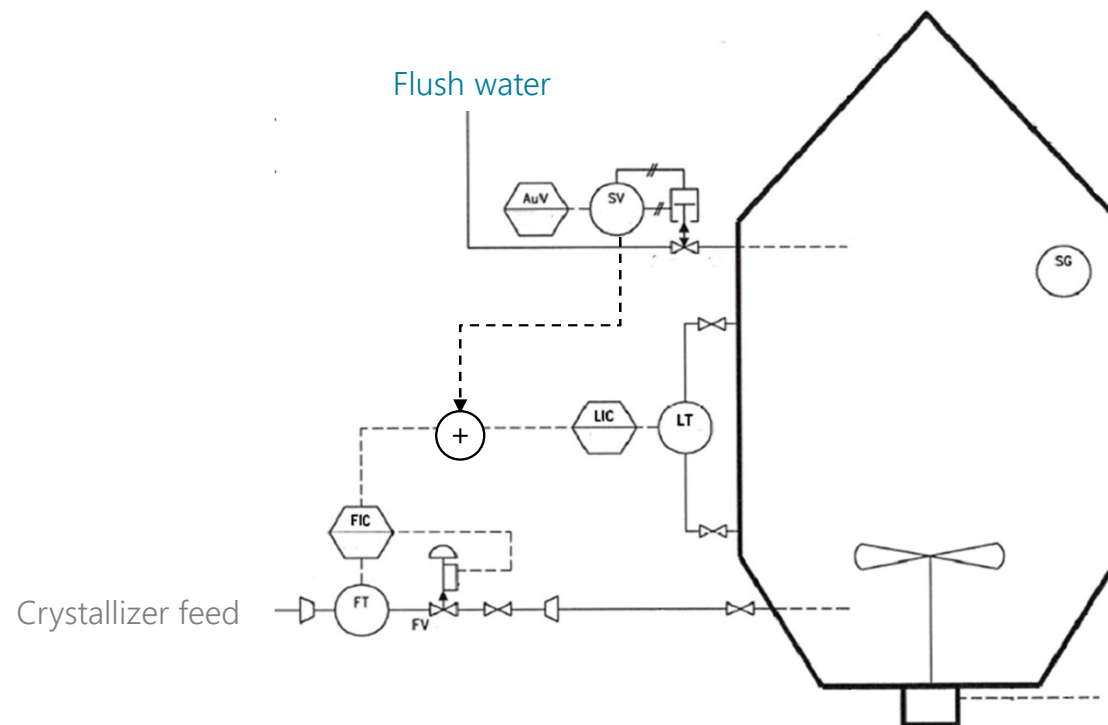
- A crystallizer is automatically flushed with water, once every second hour. The water flow is large enough to affect level.
- It's important to keep a steady level.

Improvement suggestions?



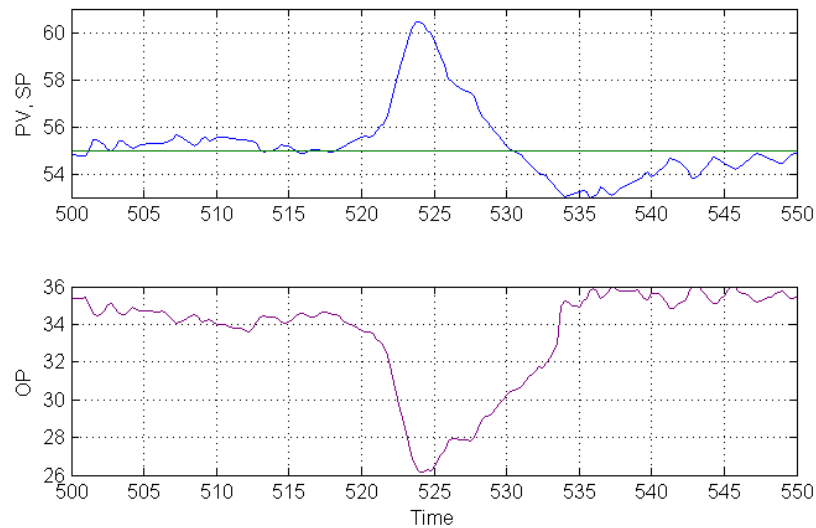
Crystallizer: Feedforward reduces level variations

- A feedforward from on-off-valve to level controller reduces level variations.
- Thus, it's ok to make a FF from a discrete (binary) variable.

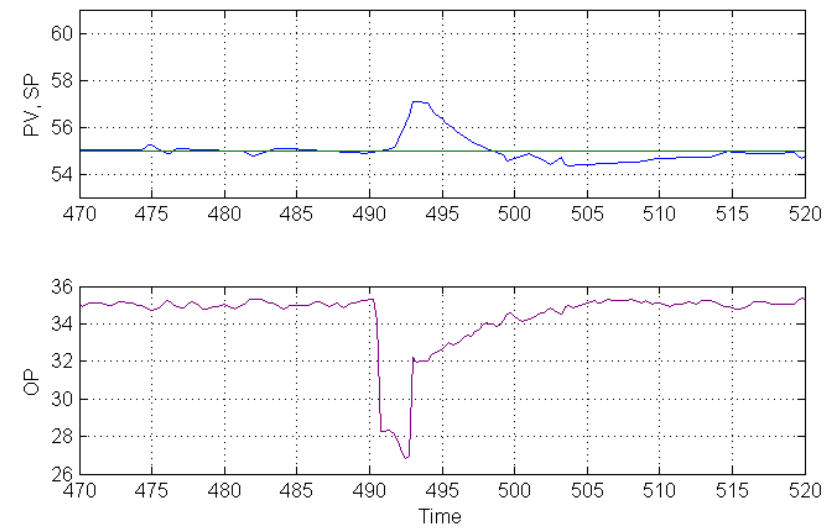


Smaller level variations in crystallizer

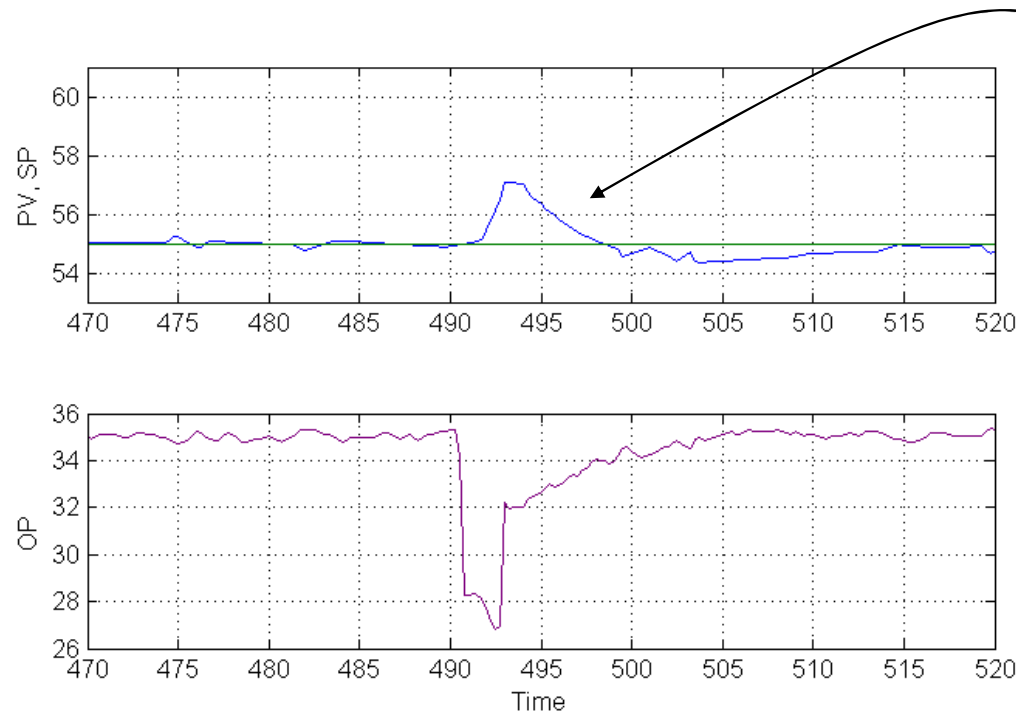
Without feedforward from on-off valve



With feedforward



Why don't we do the full compensation?



There is a residual level disturbance, because the feedforward gain used was only 0.8 of that calculated theoretically.

Why didn't we use the full gain for compensating?



Bidirectional control

An application example

Application background

- We will connect to a new source for process water, in addition to the existing one. (Site: Stenungsund)
 - Motivated by sustainability: re-use waste water
- Control challenge: Buffer management
 - The bottle neck of the entire system is sometimes in waster water supply, sometimes in plant consumption and sometimes in purification (membrane unit).
 - How do we control buffer levels?
 - It is out of the question to let operators control the levels manually, because that would be too much work.

New source: Strävliden



Plant cooling towers

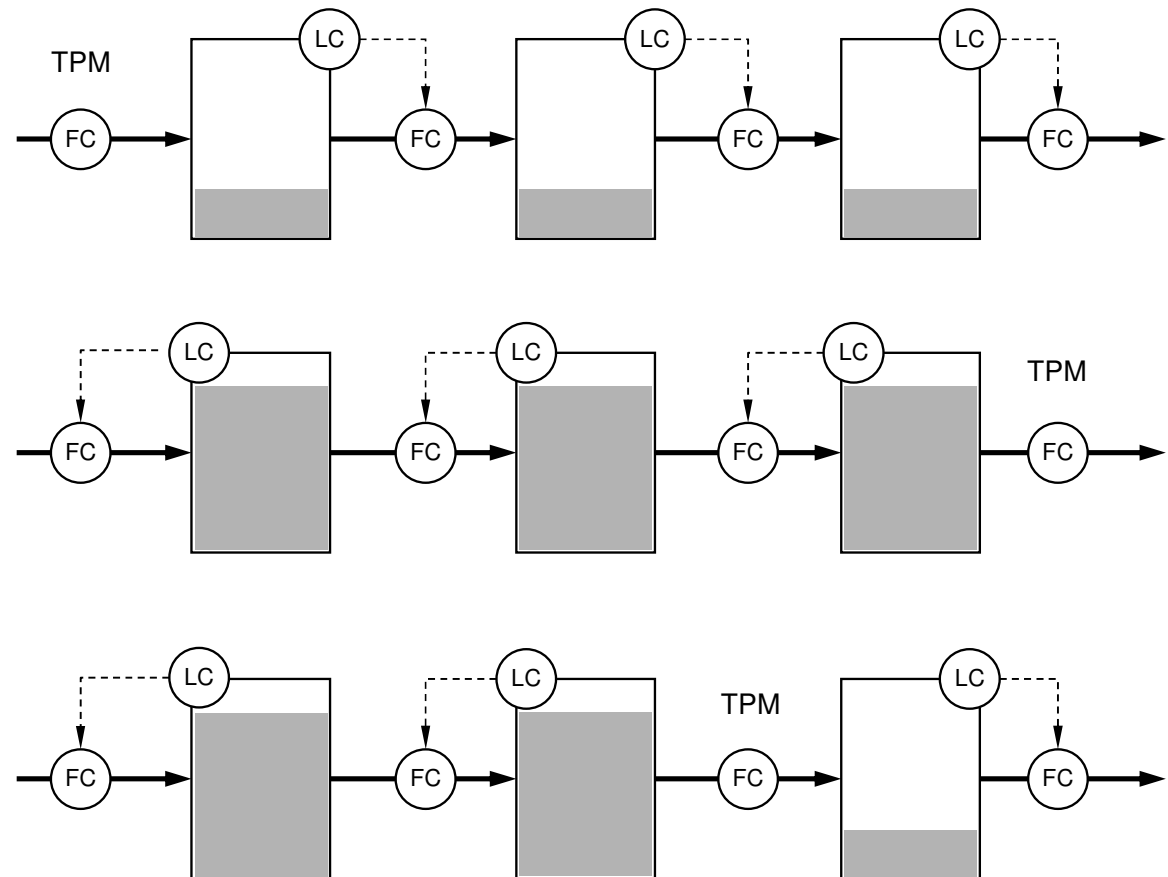


Existing source: Hällungen



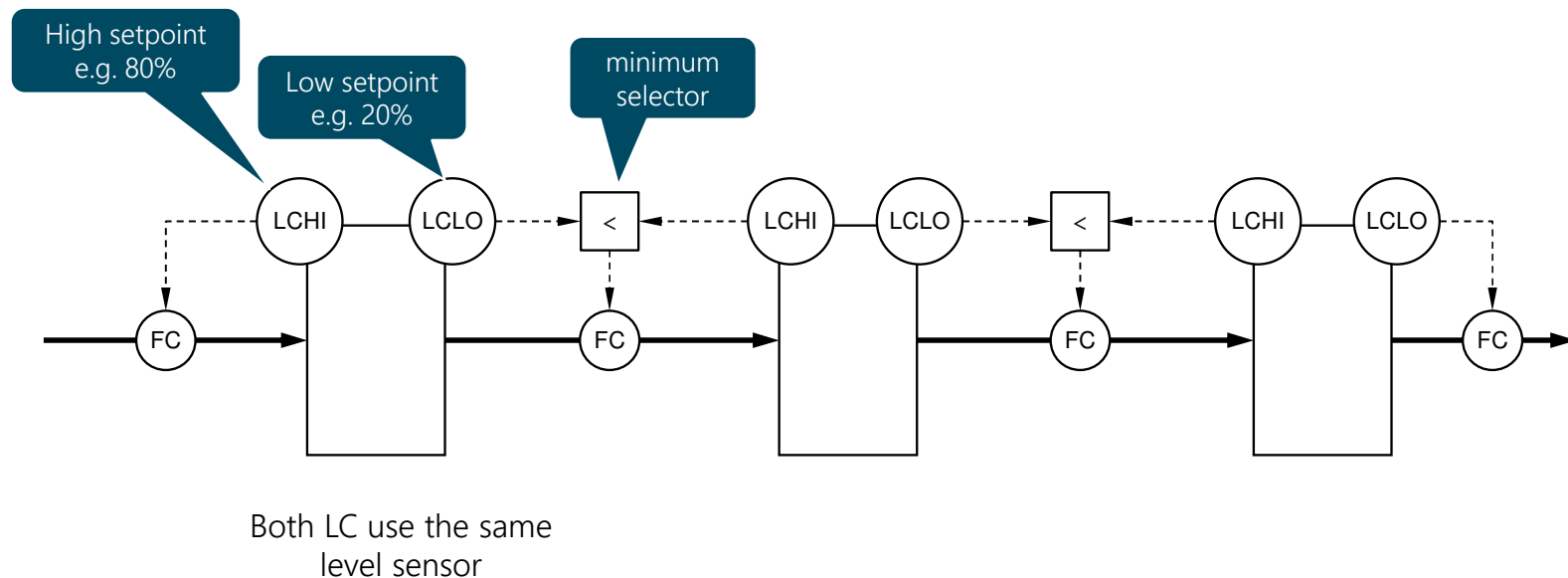
Recap: Bidirectional control

In a chain of inventories you may need different directionality of the level controllers depending on where the bottle neck (or throughput manipulator) is.



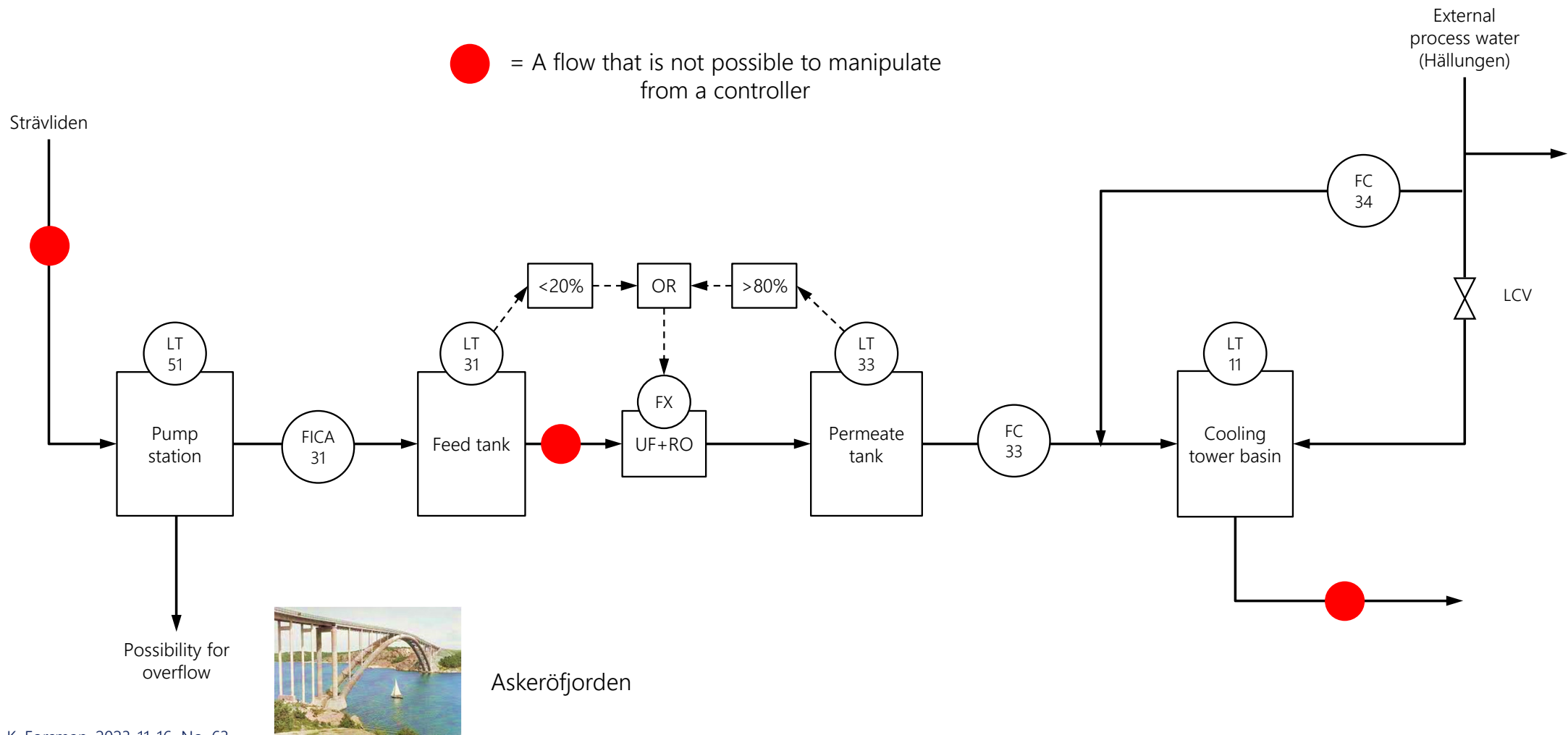
Continued recap

- A bidirectional control scheme handles all the cases automatically.
- The control structure reconfigures itself based on the position of the bottle neck.



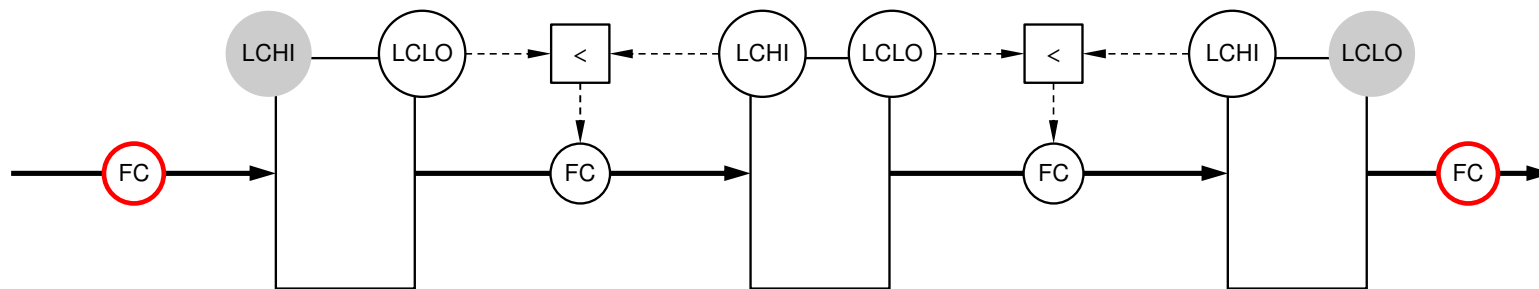
Process, without controls

● = A flow that is not possible to manipulate from a controller

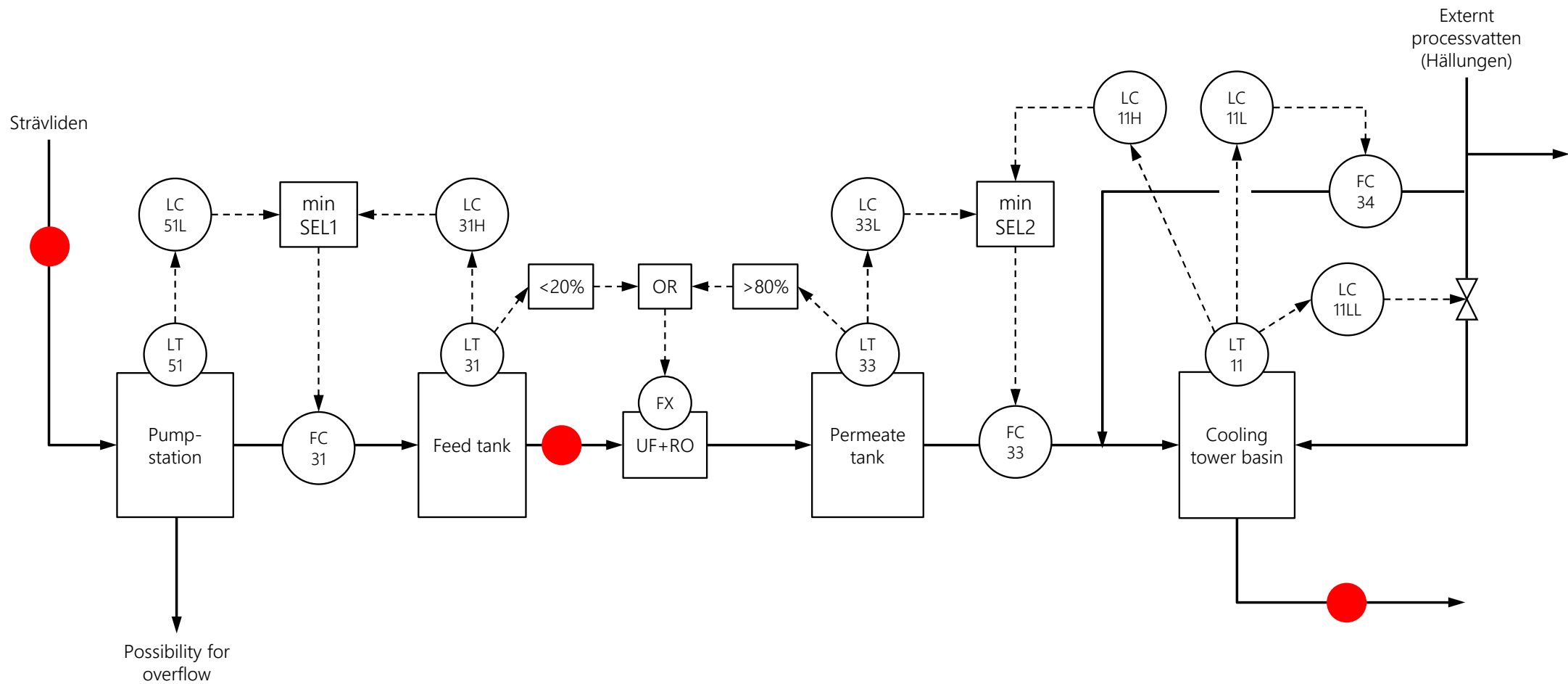


Comparison to simplified topology

- In this application, the level controllers cannot manipulate enough flows to control all levels.
- We need an overflow, and additional feed flow



Control solution implemented: Bidirectional control

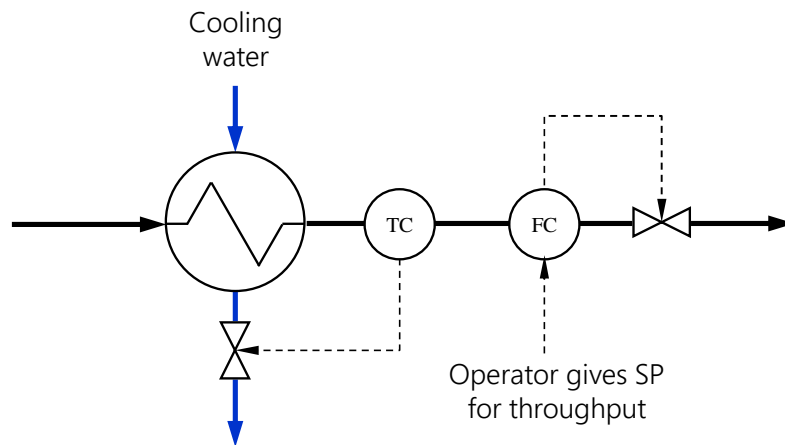


Heat exchanger control: Maximizing throughput

Maximizing throughput in a HEX (cooler)

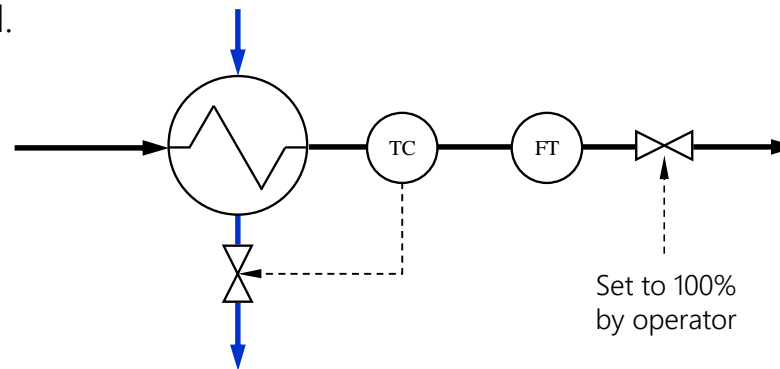
Temperature must be held at a given setpoint, e.g. 45 degrees.

"Traditional" structure:

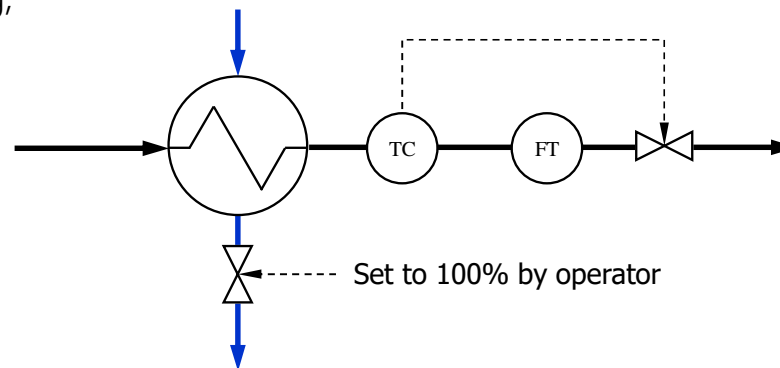


Maximizing throughput in a cooler; cont'd

If the flow valve is the limiting factor, then this structure should be used.

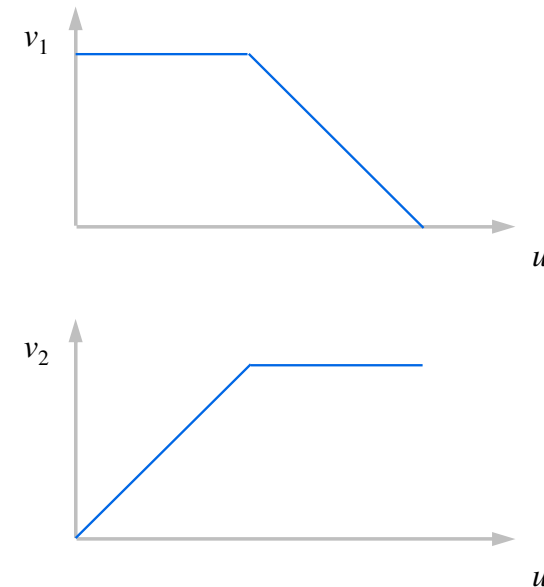
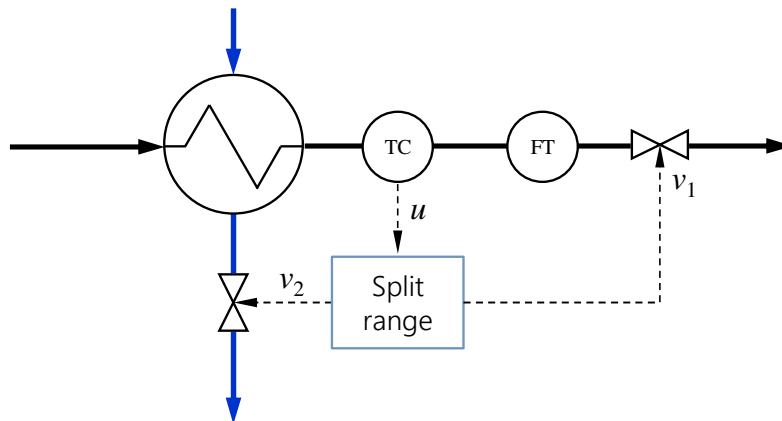


If the cooling water valve is limiting, then use this structure.



Maximizing throughput in a cooler; cont'd

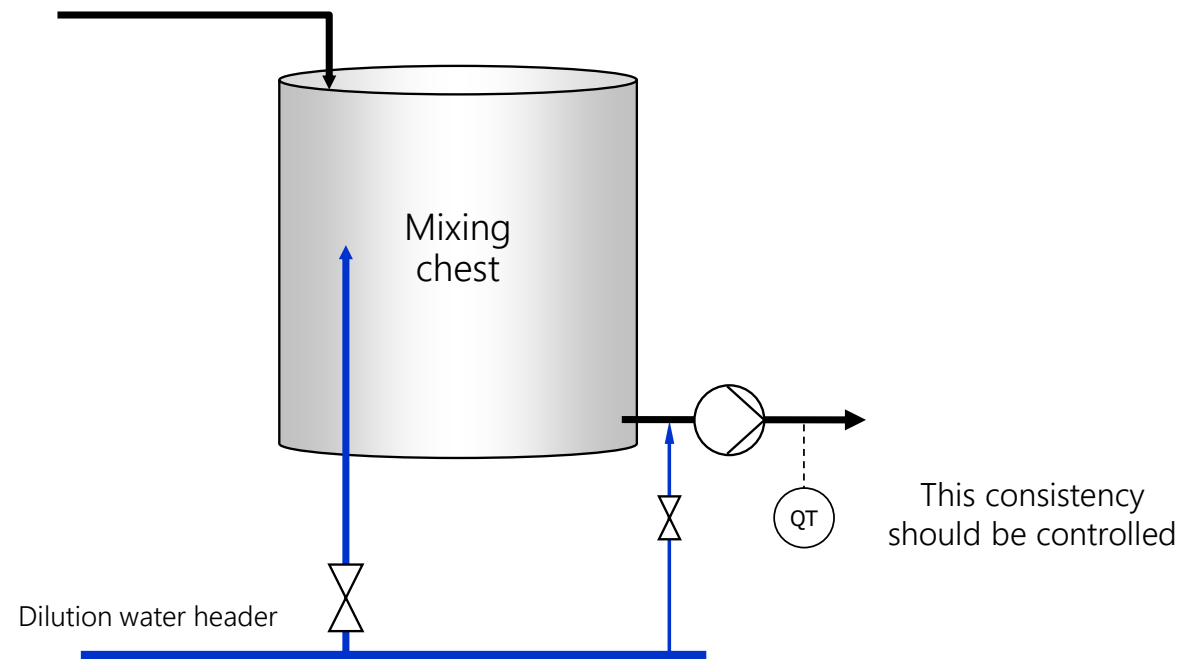
- It is not uncommon that sometimes the flow valve and sometimes the cw valve is limiting, e.g. due to cooling water temperature variations.
- There is a structure that handles both cases...



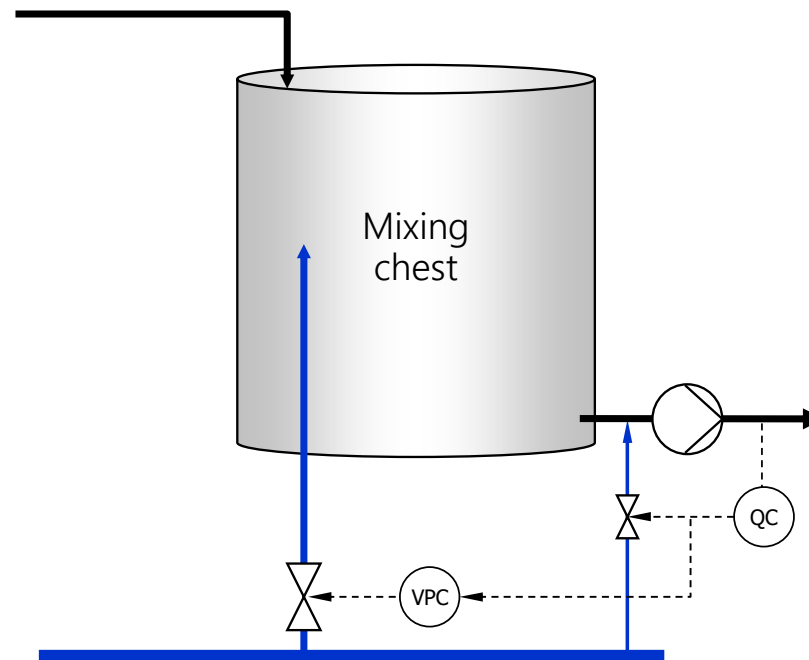
New example



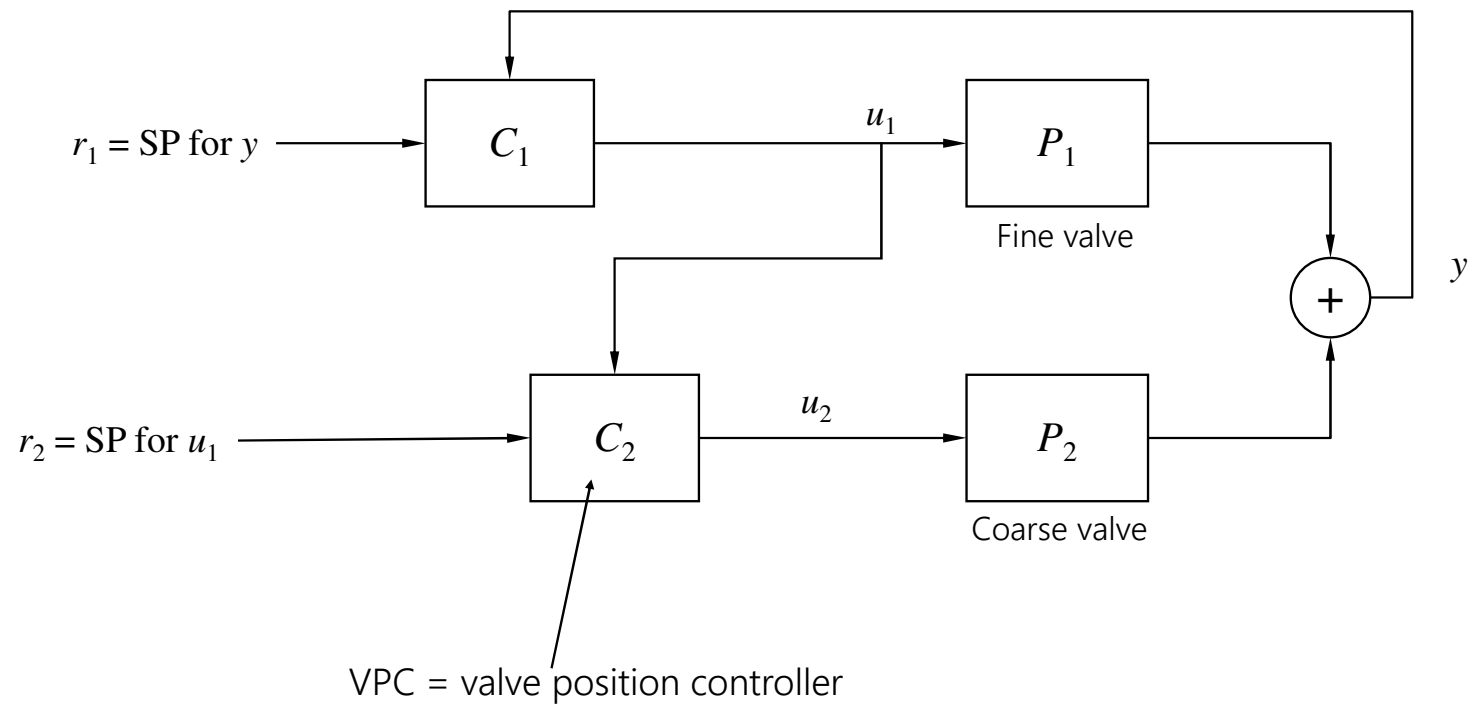
Two step dilution: How to use the extra DoF?



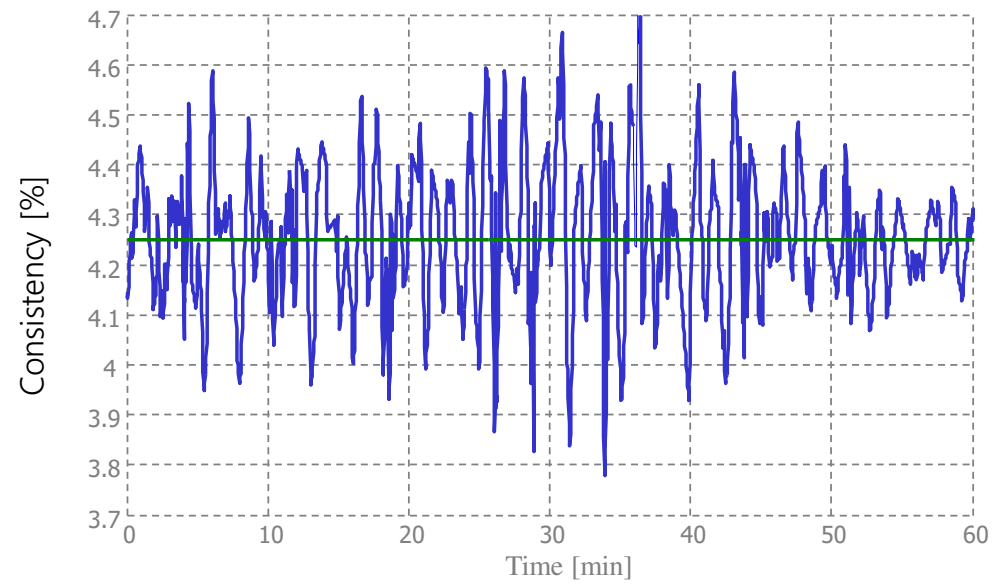
Two step dilution = Typical case for mid-ranging



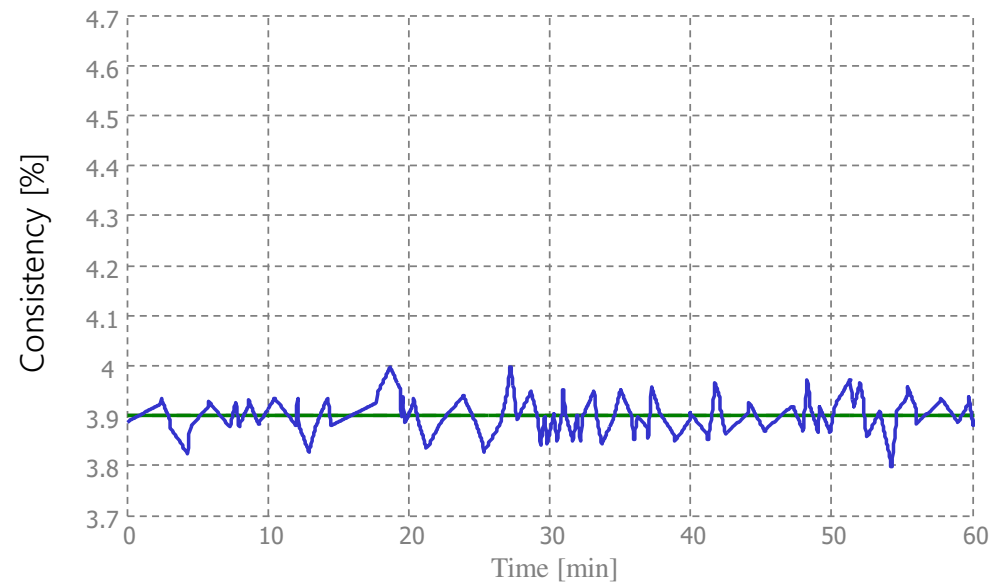
Block diagram for VPC: Give setpoint for u_1



Before



After





Perstorp