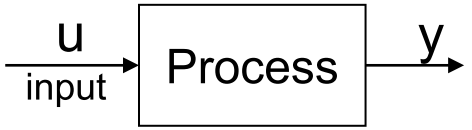
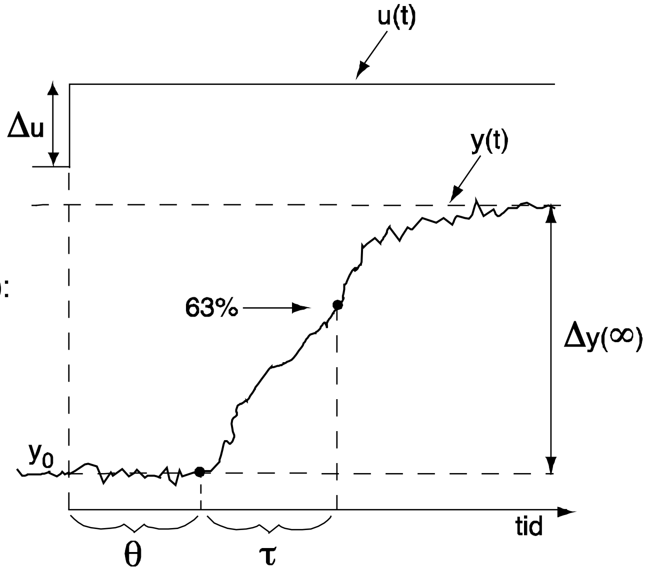
Dynamics and PID control

(part 2 of crash course)

Sigurd Skogestad



Process dynamics



- "Things take time"
- Step response (response of output y to a step in input u):
  - $k = \Delta y(\infty) / \Delta u$ – process gain
  - τ - process time constant (63%)
  - θ - process time delay

- Time constant τ: Sometimes equal to residence time = V[m3]/q[m3/s]

- Dynamic model: Can find τ (and k) from balance equations:

$$\boxed{\begin{array}{ll} \text{Mass/energy [kg/s; J/s]:} & \frac{d}{dt}\text{ Inventory} = \text{Inflow - Outflow} \\ \text{Component [mol/s]:} & \frac{d}{dt}\text{ Inventory} = \text{Inflow - Outflow + Gen. by reaction} \end{array}}$$

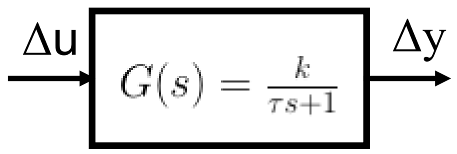
  - Then rearrange to match standard form of 1st order linear differential equation: $\boxed{\tau \frac{dy}{dt} = -y + ku}$

Response of linear first-order system

Standard form*: $\tau \frac{dy}{dt} = -y + ku,.$

Initially at rest (steady state): $y(0) = y_0.$

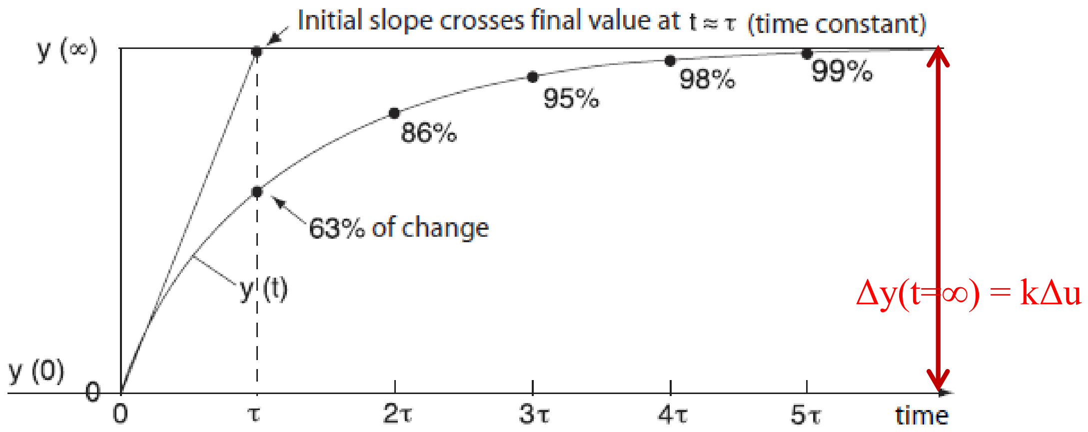
Make step in u at $t = 0$: Δu



Block diagram with transfer function for first-order process

Deviation variables:
$\Delta y = y - y_0$
$\Delta u = u - u_0$

Solution: $y(t) = y_0 + \left(1 - e^{-t/\tau}\right) \underbrace{k\Delta u}_{\Delta y(t=\infty)}$

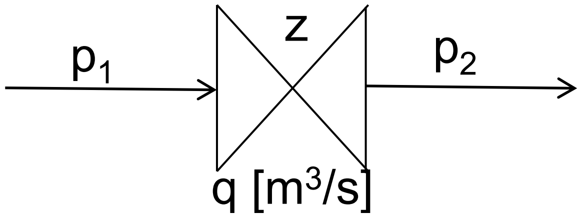


t/τ	$1 - e^{-t/\tau}$	Value	Comment
0	$1 - e^{0} =$	0	
0.1	$1 - e^{-0.1} =$	0.095	
0.5	$1 - e^{-0.5} =$	0.393	
1	$1 - e^{-1} =$	0.632	63% of change is reached after time $t = \tau$
2	$1 - e^{-2} =$	0.865	
3	$1 - e^{-3} =$	0.950	
4	$1 - e^{-4} =$	0.982	98% of change is reached after time $t = 4\tau$
5	$1 - e^{-5} =$	0.993	
∞	$1 - e^{-\infty} =$	1	

Remember for first order response:
1. Starts increasing immediately
2. Would reach new steady state after time ¿ if it kepy going with the same slope
3. Reaches 63% of change after time ¿
4. Approaches new steady state exponentially (has for practical purposes reached new steady state after about 4¿)

*A more general standard form for linear systems is the *state space* form (in deviation variables): $\frac{dx}{dt} = Ax + Bu, \quad y = Cx + Du, \quad x(0) = 0$
Our case: $A = -1/\tau, B = k/\tau, C = 1, D = 0$

More about valve equation

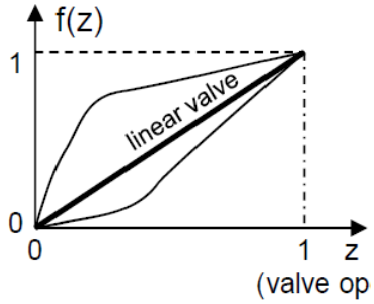


$$\text{Valve equation: } q[m^3/s] = C_v f(z)\sqrt{DP/\rho}$$

Linear valve: f(z)=z

$z \in [0,1]$ - valve opening (adjustable), $z = 0$: closed. $z = 1$: fully open.
$q[m^3/s]$ - volumetric flowrate
$DP = p_1 - p_2$ $[\mathrm{N/m^2}]$ - pressure drop over valve (Typical value: $DP = 0.1$ bar)

$C_v[m^2] = C_d A$ - valve coeffisient
C_d - dimensionless valve constant (Typical value: $C_d = 1$)
$A[m^2]$ = valve cross-sectional area
$f(z) \in [0,1]$ - valve characteristic (Linear valve has $f(z) = z$)
ρ $[\mathrm{kg/m^3}]$ - fluid density (e.g., 1000 $\mathrm{kg/m^3}$ for water; 1.19 $\mathrm{kg/m^3}$ for air at 1 bar/25C)



Derivation: From Bernoulli's equation for pressure drop (turbulent flow): $\Delta p_f = k\,\rho\, v^2$, $v = q/A$

$$\text{Comment. Mass flowrate: } w[kg/s] = \rho q = C_v f(z)\sqrt{\rho \cdot DP}$$

Even more about valve equation*

Valve (Figure 1.7b). A valve is a device that regulates the flow of substances (gases, liquids, slurries) by partially obstructing its passageways, resulting in a pressure drop. In a **control valve**, the flow can be adjusted by changing the valve position (z). The **valve equation** gives the dependency of flow on valve position and pressure drop. A typical valve equation for liquid flow is

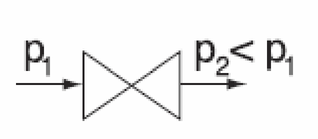
$$q = \underbrace{C_d f(z) A}_{C_v} \sqrt{\Delta p/\rho} \tag{1.8}$$

where q [m^3/s] is the volumetric flowrate, C_d (dimensionless in SI units) is the valve constant (relative capacity coefficient), z is the relative valve position (0 is fully closed and 1 is fully open), $f(z)$ is the valve characteristic (e.g., $f(z) = z$ for a linear valve), A [m^2] is the cross sectional area of the valve (at its inlet or outlet), $\Delta p = p_1 - p_2$ [N/m^2] is the pressure drop over the valve, and ρ [kg/m^3] is the fluid density. The mass flowrate is m [kg/s] $= \rho q$ and the flow velocity is v [m/s] $= q/A$ (at the valve inlet or outlet). A typical value for a control valve is $C_d \approx 1$ (see Example 9.2, page 244). $C_v = C_d f(z) A$ [m^2] is the **valve coefficient** (capacity coefficient), which depends on the valve opening. Note that the valve coefficient C_v' provided by the valve manufacturer, usually is the flow in gallons per minute (gpm) of cold water when the valve pressure drop is 1 psi, and to convert to SI units this value needs to be divided by 41625.

Exercise 1.6* *Prove that the expression for converting the manufacturer's valve coefficient C_v' to SI units is $C_v[m^2] = C_v'(\text{manufacturer})/41625$.*

A **choke (throttle) valve** is a valve where the primary objective is to reduce the pressure rather than to regulate flow.

A **Joule-Thompson** valve is a valve where the primary objective is to reduce the temperature of a non-ideal gas, by making use of the fact it requires energy to lower the pressure because of the attractive forces between the gas molecules (except at very high pressures).



Exercise 1.6, page 26. *Prove that the expression for converting the manufacturer's valve coefficient to SI units is $C_v = C_v'/41625$.*

```
Solution

Cv':  manufacturer's valve capacity coefficient,
   Cv' = flow [gpm] cold water when dp=1psi
Valve equation (SI units)
   q [m3/s]  = Cv * sqrt(dp/rho) = Cv * sqrt(e5/14.5*e3)
where
   dp = 1 psi = (1/14.5) bar = (e5/14.5) N/m2
   rho = 1000 kg/m3
Convert to gpm
    Cv' = q [gpm] = q [m3/s] / 63.09 e-6 = Cv * (1/63.09 e-6) * sqrt(100/14.5) = Cv * 41625
```

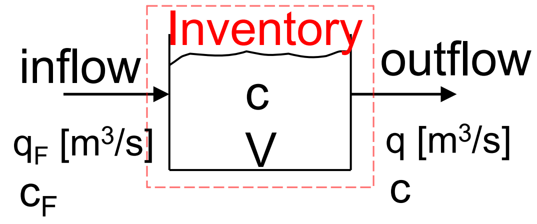
Conclusion: $C_v = C_v' \cdot 63.09e - 6\sqrt{14.5e-2} = C_v'/41625$

```
Comment:
The KV-value used by valve manufacturers in Europe needs to be divided by 36000. Proof:
    Kv = flow [m3/h] cold water when dp=1bar
    dp = 1 bar = e5 N/m2
Valve equation gives under these conditions:
   q [m3/s]  = Cv * sqrt(dp/rho) = Cv * sqrt(e5/e3)
Convert to m3/h
    Kv = q [m3/h] = q [m3/s] * 3600 = Cv * 3600 * sqrt(100) = Cv * 36000

Reference: B.L.Liptak (Editor), Instrument Engineers' Handbook,
4th Edition, CRC Taylor & Francis and ISA,
Volume II (Process control and optimization), p. 1051 (2006)
```

*From: S. Skogestad, *Chemical and Energy Process Engineering*, CRC Press, 2009

Example dynamic model: Concentration change in mixing tank



- Assume constant V [m^3]
- Assume constant density ρ [kg/m^3]
- Assume, c (in tank) = c (outflow) [mol A/m^3]
- Assume no reaction

	Mass balance	**Component balance**
Inflow	ρq_F [kg/s]	$c_F\ q_F$ [mol A /s]
Outflow	ρq [kg/s]	c q [mol A/s]
Inventory ("state variable")	ρV [kg]	c V [mol A]

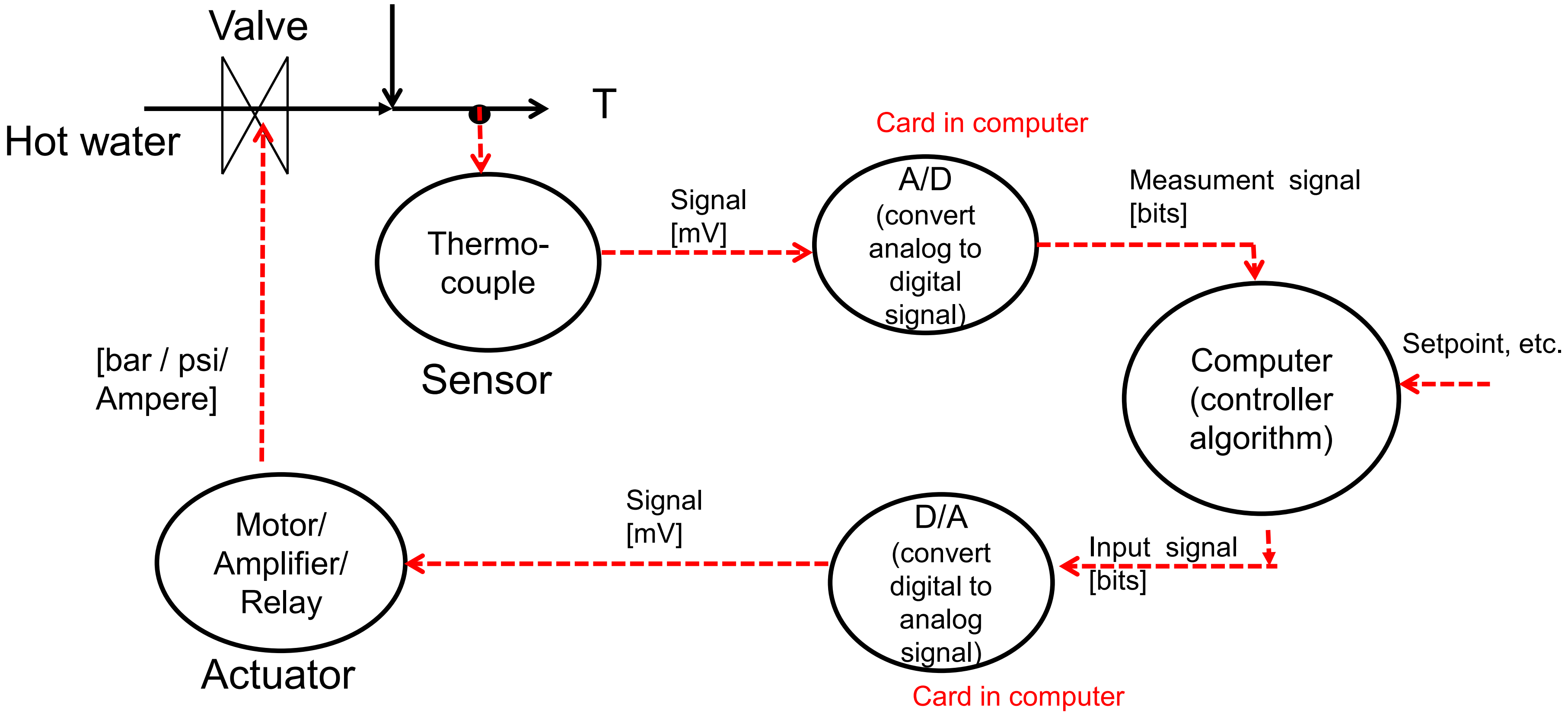
Balances:

Mass $\frac{d(\rho V)}{dt} = \rho q_F - \rho q \quad [\mathrm{kg/s}]. \quad \rho V \text{ constant} \Rightarrow q = q_F$

Component: $\frac{d(cV)}{dt} = c_F q_F - cq \quad [\mathrm{mol\ A/s}] \Rightarrow \underbrace{V/q}_{\tau} \frac{dc}{dt} = -c + \underbrace{1}_{k} \cdot c_F$

Feedback control

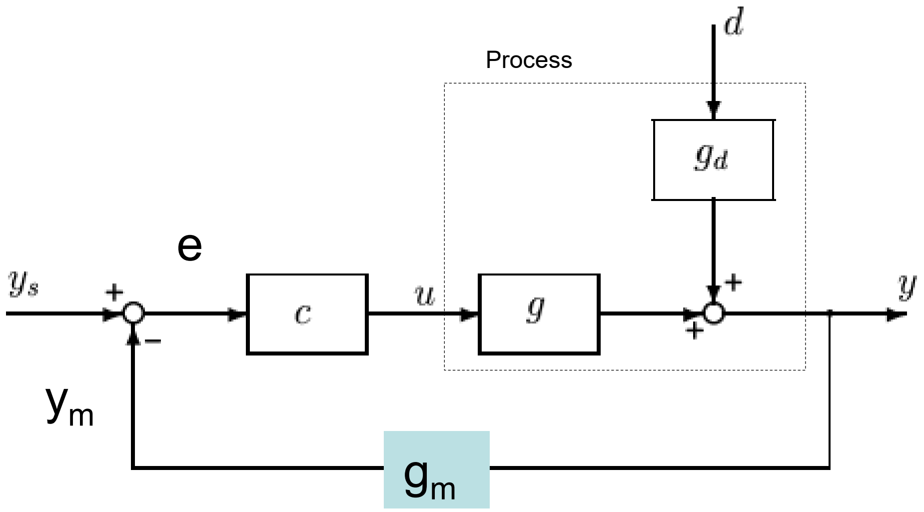
Control systems elements:



Measurements

- **Pressure** is usually the most robust, fast and cheap measurement
  - So we use it also for
    - Level (DP)
    - Flow (DP for venturi or orifice)
- **Temperature** is usually also robust and cheap
  - Thermocouple, Thermistors (resistors), infrared, etc.…
- **Composition** is often difficult
  - Gas chromotograph (GC) is best for low concentrations but it’s expensive and gives time delay (typical 5-10 minutes)

Block diagram



Block diagram of negative feedback control

Lines are signals ("information"):

y = controlled variable (CV)

y_m = measured CV

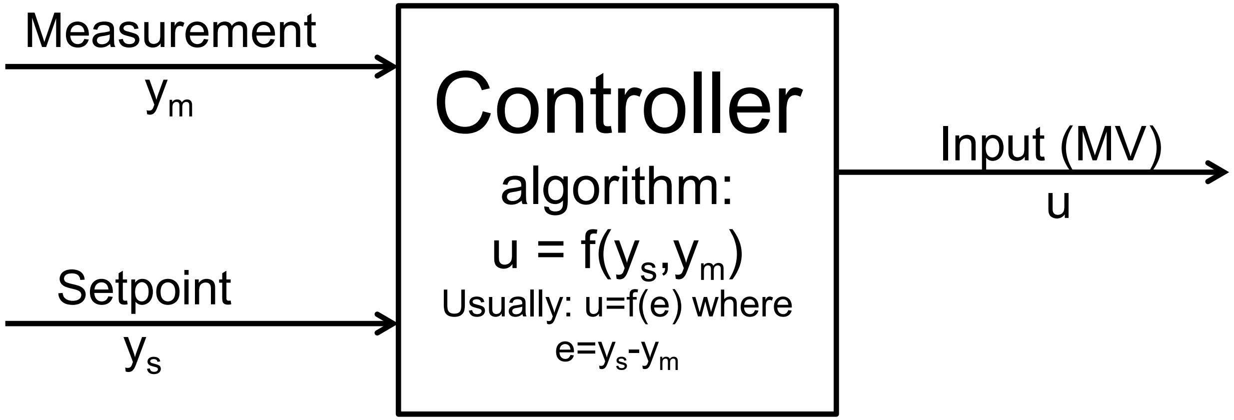
y_s = setpoint (SP)

$e = y_s - y_m$ = control error

u = manipulated variable (MV)

C = Feedback Controller = ?

Feedback controller

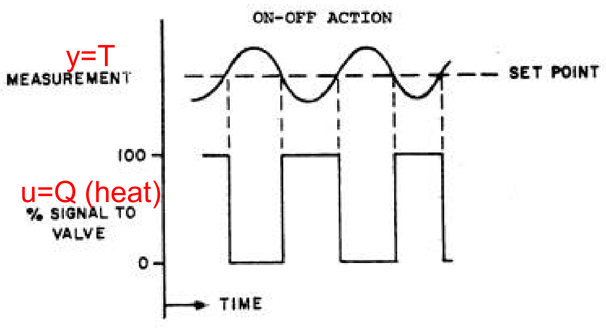


Simplest controller algorithm: On/off controller.

Example: Common thermostat

Problem: Always cycles

FIGURE 5



Industry: Standard algorithm for SISO controllers: PID

Industry: Standard for interactive multivariable control: MPC (model predictive control)

PID controller

- Proportional control (P)

$$u = u_0 + K_c \overbrace{(y_s - y)}^{e}$$

Input change (u-u_0) *is proportional to control error e.*
K_c = proportional gain (tuning parameter)
u_0: = «bias»

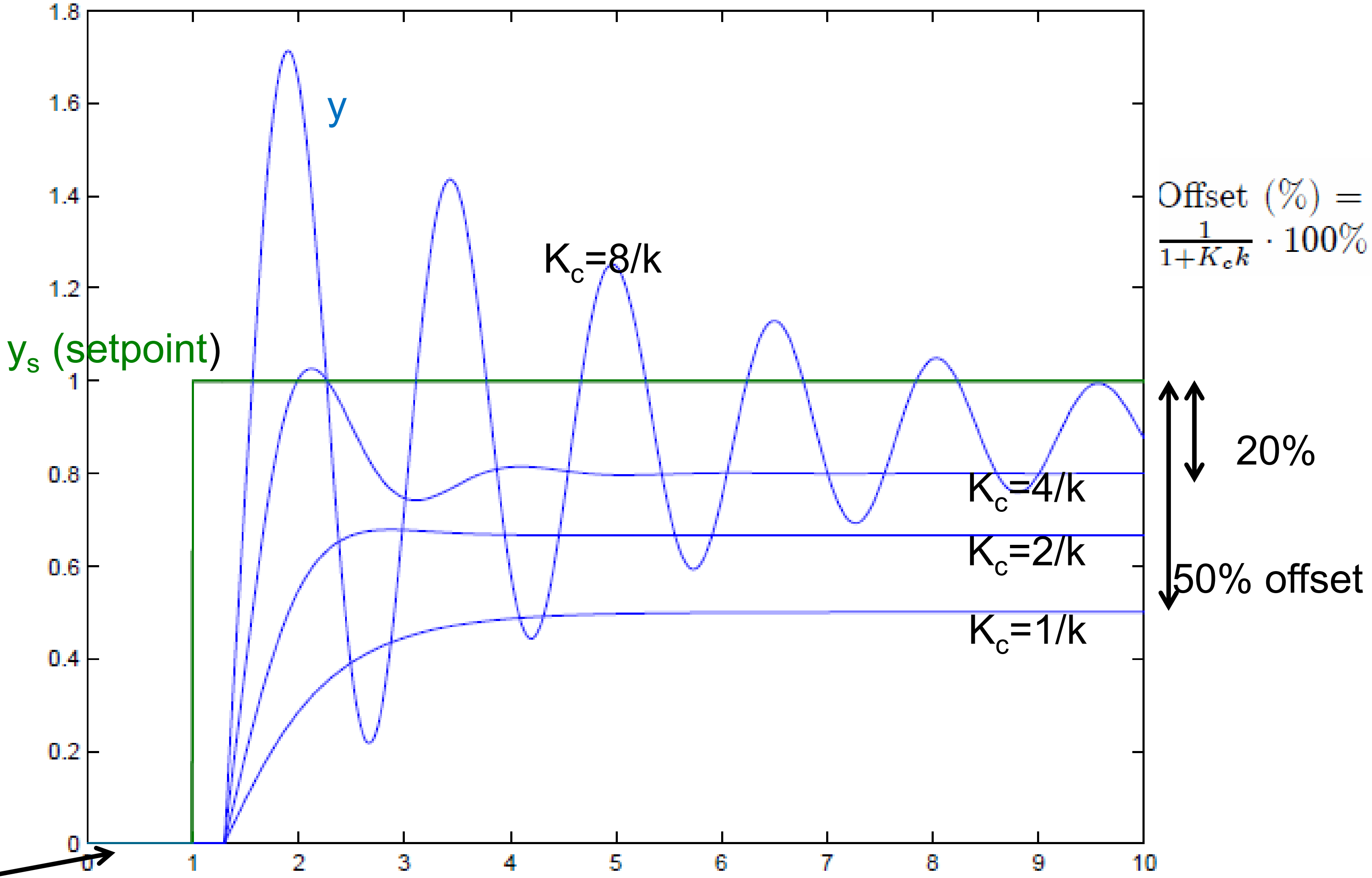
Problems proportional control:
1. Get steady-state offset (especially if K_c is small)

$$\text{Offset } (\%) = \frac{1}{1+K_c k} \cdot 100\%$$

k: process gain
K_c: controller gain

2. Oscillates if K_c is too large (can get instability)

P-control of typical process



Initially at steady-state ($y=y_s=0$). Change setpoint to $y_s=1$ at $t=1$.

Process has theta=0.3min and tau=1.5 min.

- Fix: Add Integral action (I)
- Get PI-control:

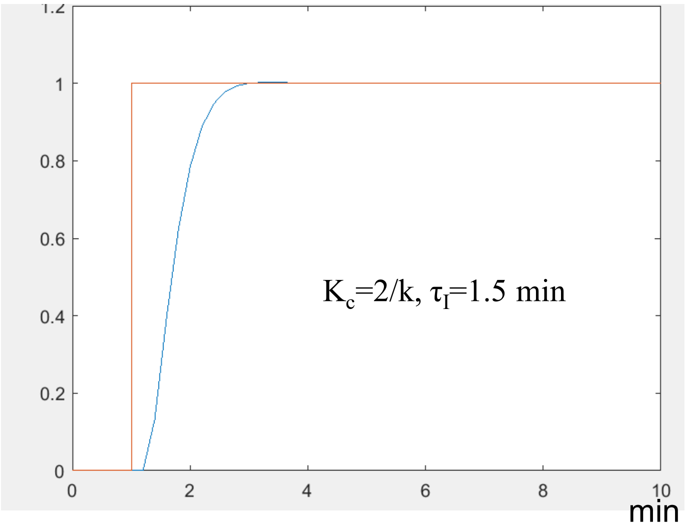
$$u(t) = u_0 + K_c e(t) + K_c \frac{\int_0^t e(t)dt}{\tau_I}$$

¿I = integral time (tuning parameter)
e = y_s – y (control error)

Note 1: Integral term will keep changing until e=0) No steady-state offset

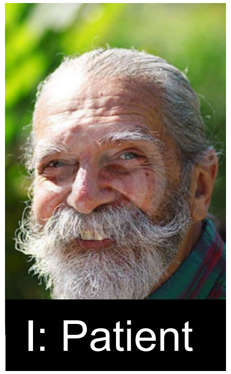
Note 2: Small integral time gives more effect! (so set ¿I = 99999 (large!) to turn off integral action)

Note 3: Integral action is also called «reset action» since it «resets» the bias.



Process has theta=0.3 min and tau=1.5 min.

Add also derivative action (D): Get PID controller



$$u(t) = u_0 + \underbrace{K_c[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D \frac{de(t)}{dt}]}_{\Delta u}$$

- P-part: MV (Δu) proportional to error
  - This is usually the main part of the controller!
- I-part: Add contribution proportional to integrated error.
  - Integral keeps changing as long as e≠0
  - -> Will eventually make e=0 (no steady-state offset!)
- Possible D-part: Add contribution proportional to change in (derivative of) error
  - Can improve control for high-order (S-shaped process response) and unstable processes, but sensitive to measurement noise

Many alternative PID parameterizations

This course:

$$u(t) = u_0 + K_c[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D \frac{de(t)}{dt}]$$

Alternative form :

$$u(t) = u_0 + Pe(t) + I\int_0^t e(t)dt + D\frac{de(t)}{dt}$$

$$P = K_c, \quad I = K_c/\tau_I, \quad D = K_c\tau_D$$

Also other:

Proportional band = $100/K_c$

Reset rate = $1/\tau_I$

Etc…..

NOTE: Always check the manual for your controller!

Comment 1: Often the D-action is not on the setpoint, so the D-term becomes $-\tau_D\, dy/dt$

Comment 2: This is the «ideal» PID. In this course we also use the series PID (SIMC-rule). For PI they are the same.

Digital implementation (practical in computer) of PID controller

Continuous (not possible in computer):

$$u(t) = \underbrace{u_0 + \frac{K_c}{\tau_I}\int_0^t e(t)dt}_{\bar{u}(t)} + K_c e(t) + K_c \tau_D \frac{de(t)}{dt}$$

where $\bar{u}(t)$ = bias term with integral action included

Introduce:

Δt= sampling time

k=current value (at time t)

$k-1$=previous value (at time $t-\Delta t$)

Discrete (digital) approximations :

$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{\Delta t}$

Integral action = «Reset» of bias:

$$\bar{u}_k = \bar{u}(t) \approx \bar{u}_{k-1} + \frac{K_c}{\tau_I} e_k \Delta t$$

Conclusion: Digital PID implementation

$$u_k = \bar{u}_k + K_c e_k + K_c \tau_D \frac{e_k - e_{k-1}}{\Delta t}$$

PID controller tuning

$$u(t) = u_0 + \underbrace{K_c[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D \frac{de(t)}{dt}]}_{\Delta u}$$

3 tuning parameters:

1. (Proportional) Controller Gain: K_c
2. Integral time: τ_I [s]
3. Derivative time: τ_D [s]

Want the system to be (TRADE-OFF!)

1. Fast intitially (K_c large, τ_D large)
2. Fast approach to steady state (τ_I small)
3. Robust / stable (OPPOSITE: K_c small, τ_I large)
4. Smooth use of inputs (OPPOSITE: K_c small, τ_D small)

Tuning of your PID controller
I. “Trial & error” approach (online)

(a) P-part: Increase controller gain (K_c) until the process starts oscillating or the input saturates

(b) Decrease the gain (~ factor 2)

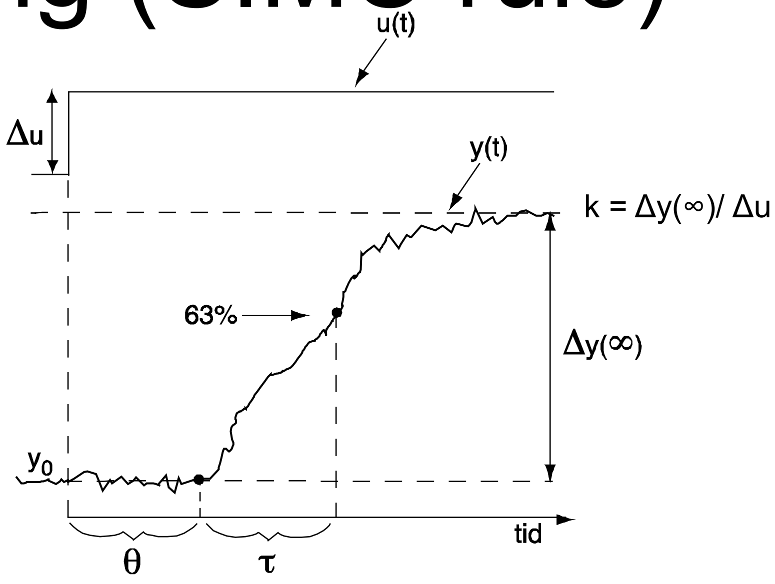
(c) I-part: Reduce the integral time (τ_I) until the process starts oscillating

(d) Increase a bit (~ factor 2)

(e) Possible D-part: Increase τ_D and see if there is any improvement

Very common approach,
BUT: Time consuming and does not give good tunings: NOT recommended

II. Model-based tuning (SIMC rule)



- From step response obtain
  - k = Δy(∞)/ Δu – process gain
  - τ - process time constant (63%)
  - θ - process time delay

- Proposed SIMC controller tunings

$$K_c = \frac{1}{k}\frac{\tau}{\tau_c + \theta}$$

$$\tau_I = \min(\tau, 4(\tau_c + \theta))$$

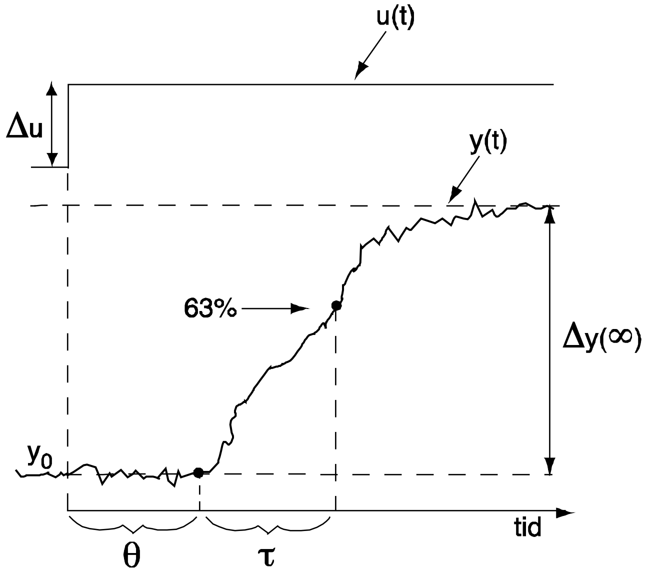
τ_c = desired response time with control (tuning parameter!).

- Choose $\tau_c = \theta$ (delay) for "tight" control
- Choose $\tau_c > \theta$ for smoother control (but $K_c \geq \frac{\Delta u_{max}}{\Delta y_{max}}$)

τ_D : normally 0 (may try $\tau_D = \tau_2$= 2nd order time constant (e.g. response time measurement), but should then get new τ_1 and θ based on 2nd order response)

Example SIMC rule

- From step response
  - k = Δy(∞)/ Δu = 10C / 1 kW = 10
  - τ = 0.4 min (time constant)
  - θ = 0.3 min (delay)



- Proposed controller tunings

$$\text{Select } \tau_c = \theta = 0.3 \text{ min ("tight" control):}$$

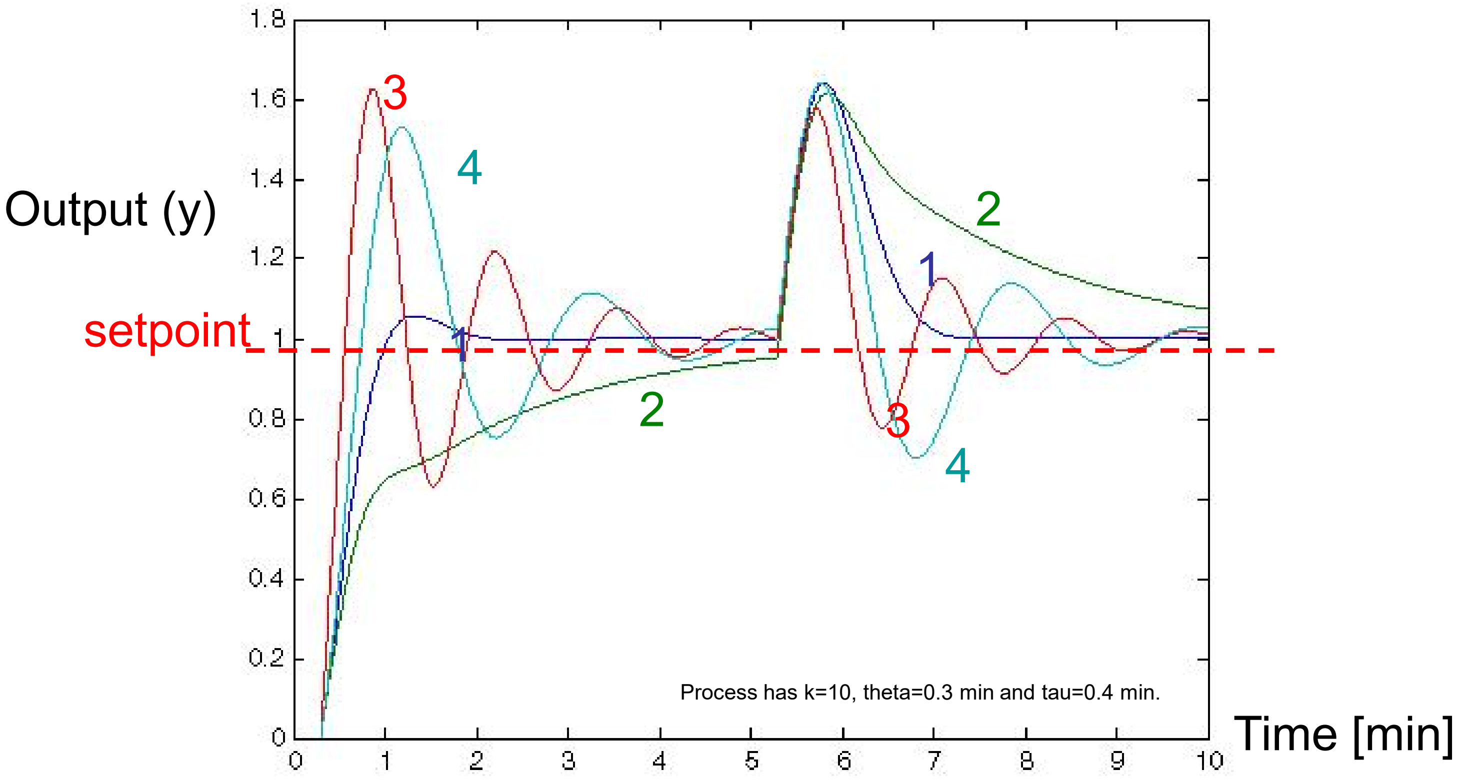
$$K_c = \frac{1}{k}\frac{\tau}{\tau_c+\theta} = \frac{1}{10}\frac{0.4}{0.3+0.3} = 0.067$$

$$\tau_I = \min(\underbrace{\tau}_{0.4}, 4\underbrace{(\tau_c+\theta)}_{0.3+0.3}) = \min(0.4, 2.4) = 0.4\text{min}$$

Simulation PID control

- Setpoint change at t=0 and input disturbance at t=5 min
  1. Well tuned (SIMC): Kc=0.07, taui=0.4min
  2. Too long integral time (Kc=0.07, taui=1 min) : settles slowly
  3. Too large gain (Kc=0.15, taui=0.4 min) – oscillates
  4. Too small integral time (Kc=0.07, taui=0.2 min) – oscillates
  5. Even more aggressive (Kc=0.12, taui=0.2 min) – unstable (not shown on figure)

  –



Comment: Can avoid the setpoint overshoot for curve 1 by adding derivative action (try taud=0.3/3=0.1 with series PID) but will be more sensitive to noise

Comments tuning

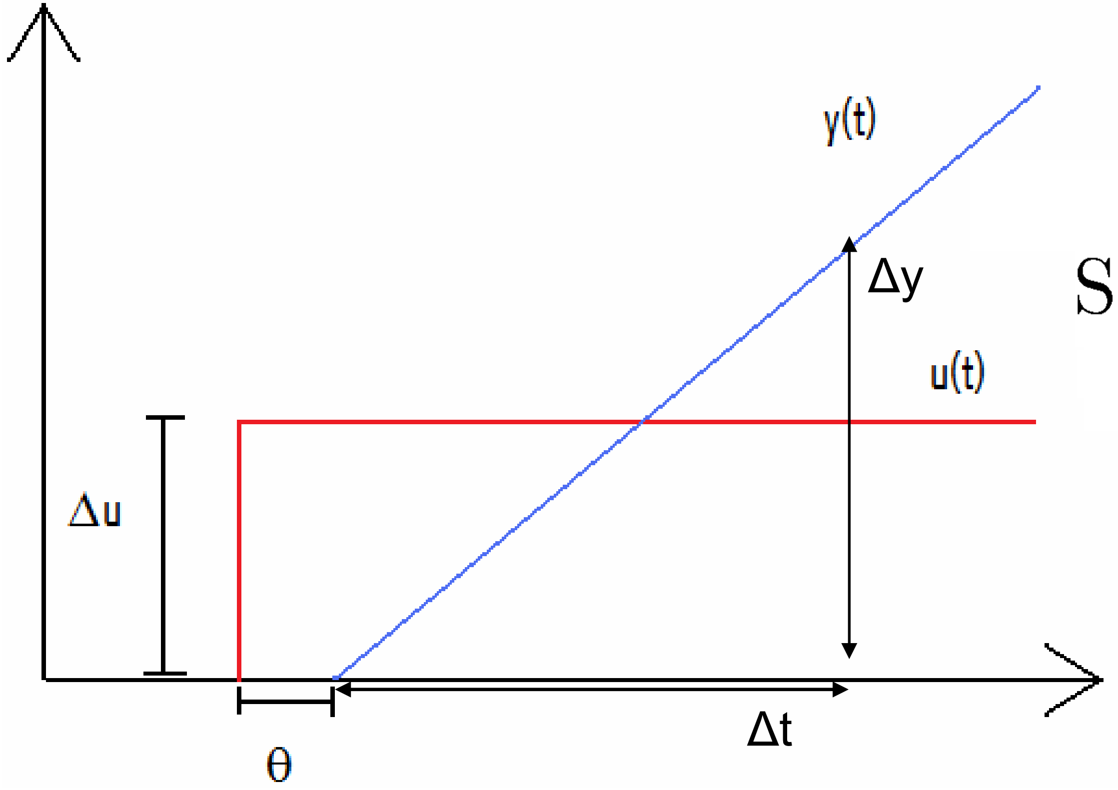
1. Delay (θ) is feedback control's worst enemy!
   - Try to reduce it, if possible. Rule: "Pair close"!

2. Common mistake: Wrong sign of controller!
   - Controller gain (K_c) should be such that controller *counteracts* changes in output
   - Need negative sign around the loop ("***negative feedback***")
   - Two ways of achieving this:

     - (Most control courses:) ***Use a negative sign in the feedback loop***. Then controller gain (K_c) should always have same sign as process gain (k)
     - (Most real control systems*:) ***Always use K_c positive*** and select between
       - "Reverse acting" in the normal case when the process gain (k) is positive
         - » because MV (u) should go down when CV (y) goes up (to get negative feedback), for example, when we use heat (u=Q) to control room temperature (y=T).
       - "Direct acting" when k is negative
       - Comment: This convention is common in process control (including Aspen/Hysys simulation software)
       - BUT WARNING: Be careful and read manual! Some use «direct» and «reverse» opposite!, e.g., wikipedia on PID control:
       - 2021: https://en.wikipedia.org/wiki/PID_controller

* Including Emerson, Honeywell, ABB, Yokogawa.

3. Integrating («slow») process: If the response is not settling after approximately 10 times the desired closed-loop time constant (so ¿/(¿$_c$+θ) >4), then you can stop the experiment and approximate the response as an integrating process (with only two parameters, k' and μ):



$$\text{Slope, } k' = \frac{\Delta y}{\Delta t \cdot \Delta u}$$

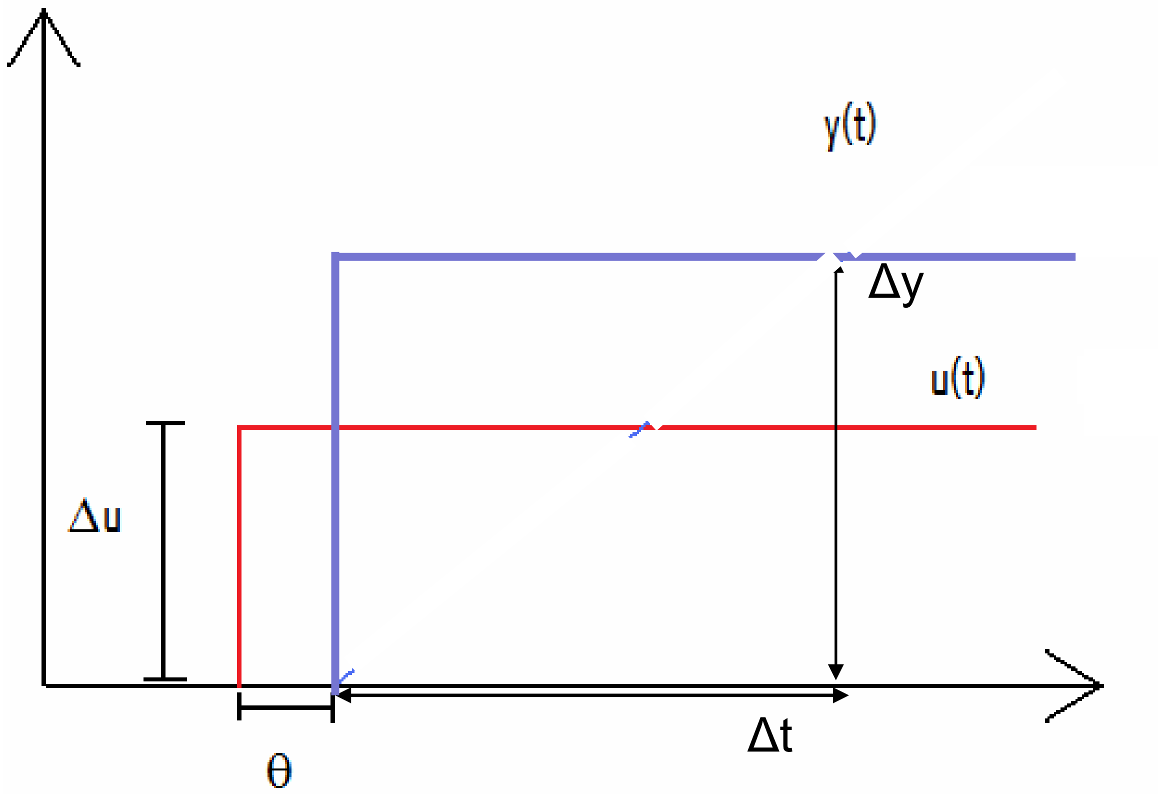
SIMC-settings (using $k' = k/\tau$):

$$K_c = \frac{1}{k'}\frac{1}{\tau_c+\theta}$$

$$\tau_I = 4(\tau_c + \theta)$$

Integrating processes require mainly P-action and are difficult to control manually

4. «Fast» process: If the response is fast compared to the desired closed-loop time constant (so ¿/(¿$_c$+θ) < 0.25, approximately), then you dom't really need so much P-action in the controller and you can approximate the response as a delay process (with only two parameters, k and μ):

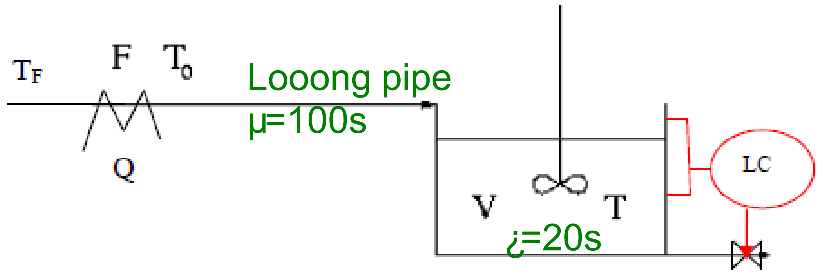


Gain,
k=Δy/Δu

May use pure I-controller with K$_I$=1/k(¿$_c$+θ).
Same as PI with K$_c$=¿/k(¿$_c$+θ) and ¿$_I$ =¿ for small ¿ (but then actual value of ¿ does not really matter).

Pure delay processes require mainly I-action and are easy to control manually

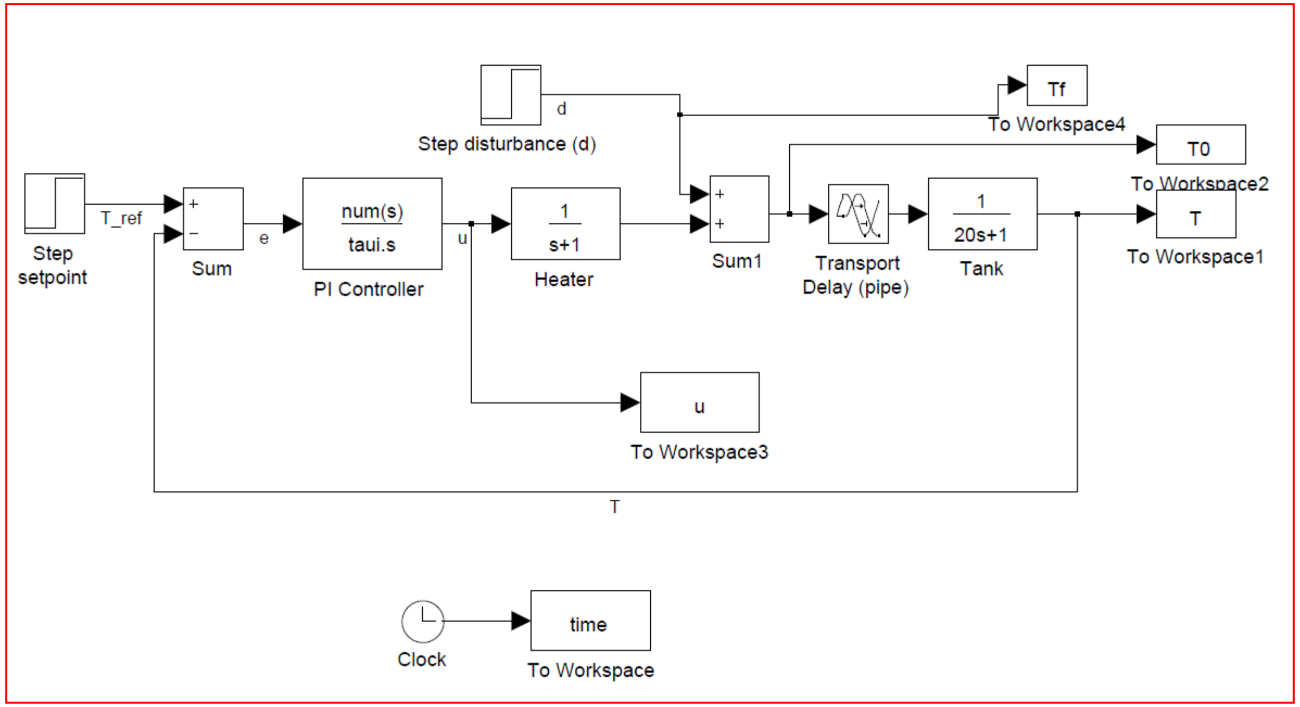
Example: Similar to shower process



u = Q

y = T

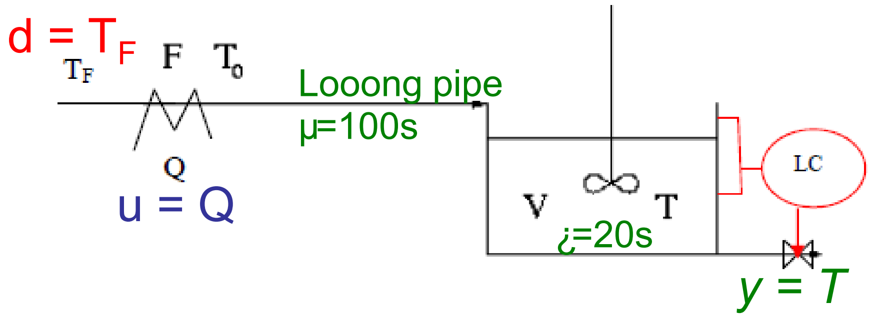
d = T_F

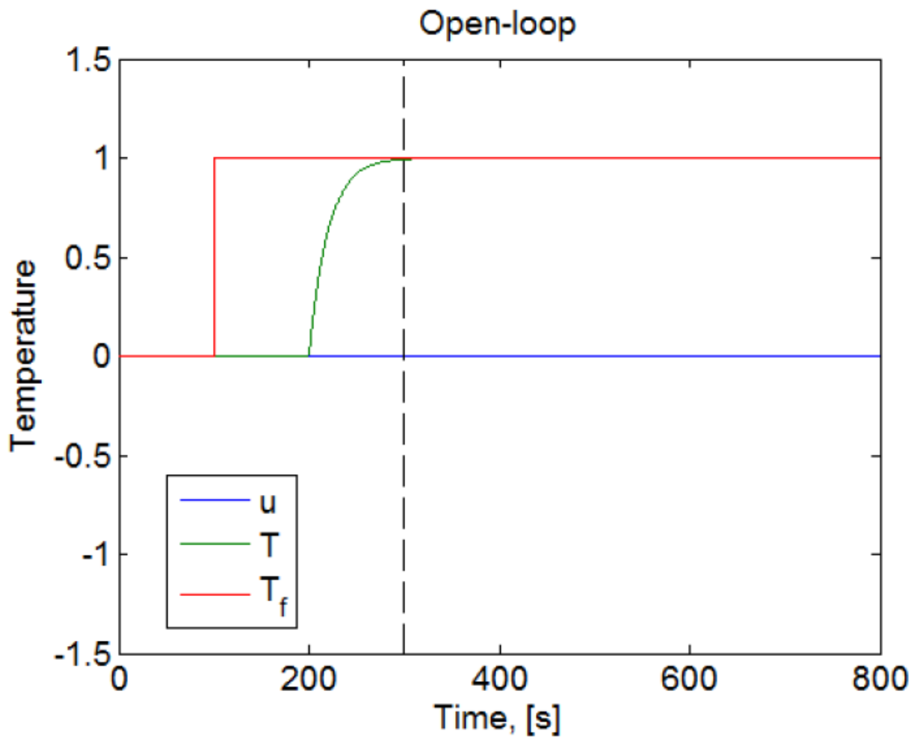


Simulink model: tunepid1_ex1

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

Disturbance response with no control





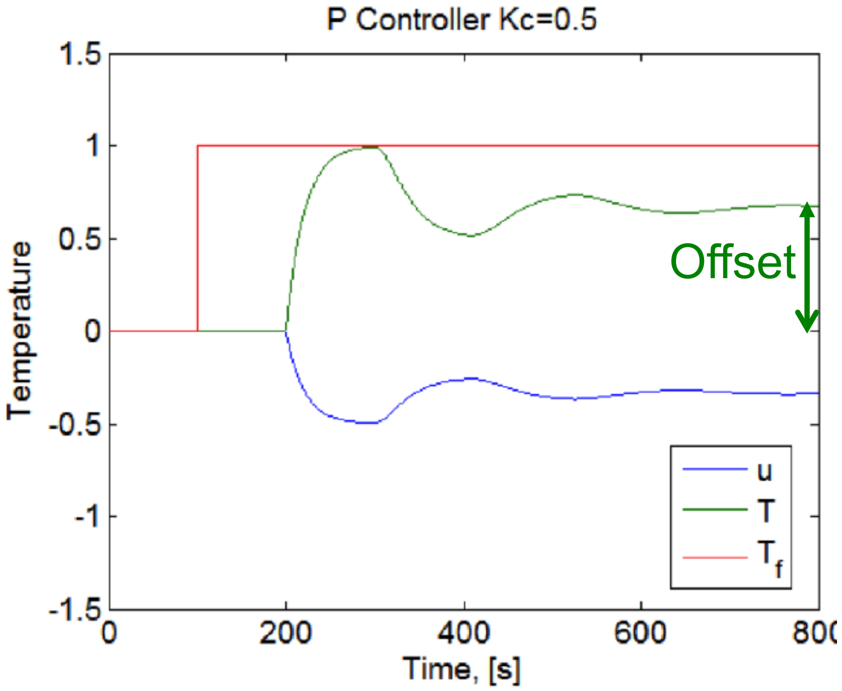
u = Q
y = T
d = T_F

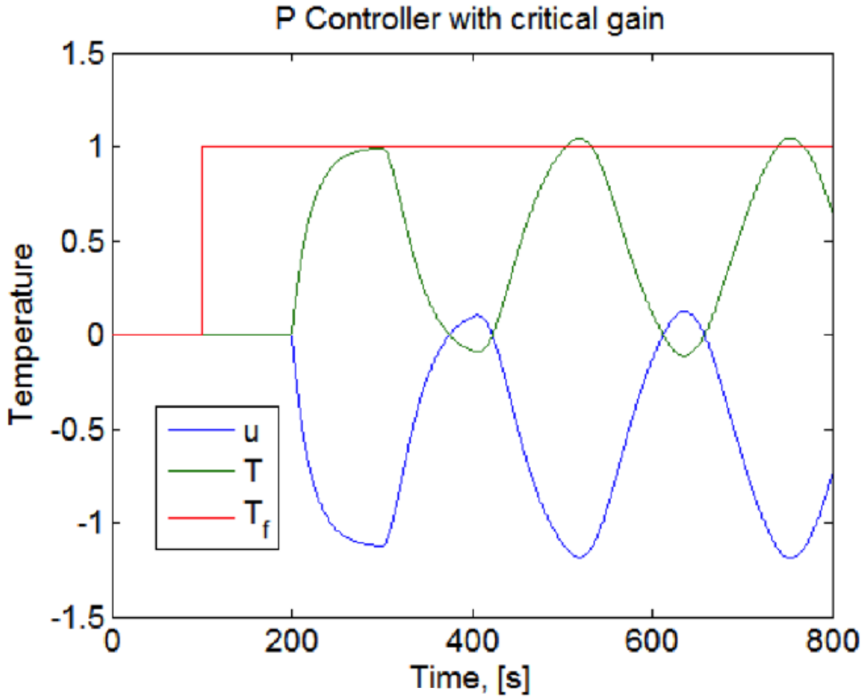
```
Kc=0; taui=9999; % no control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
```

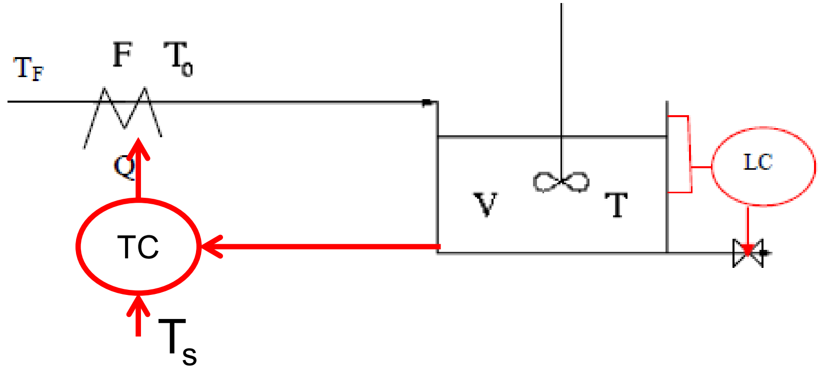
P-control

u = Q
y = T
d = T_F

and error we find it to be $K_c = 1.13$.







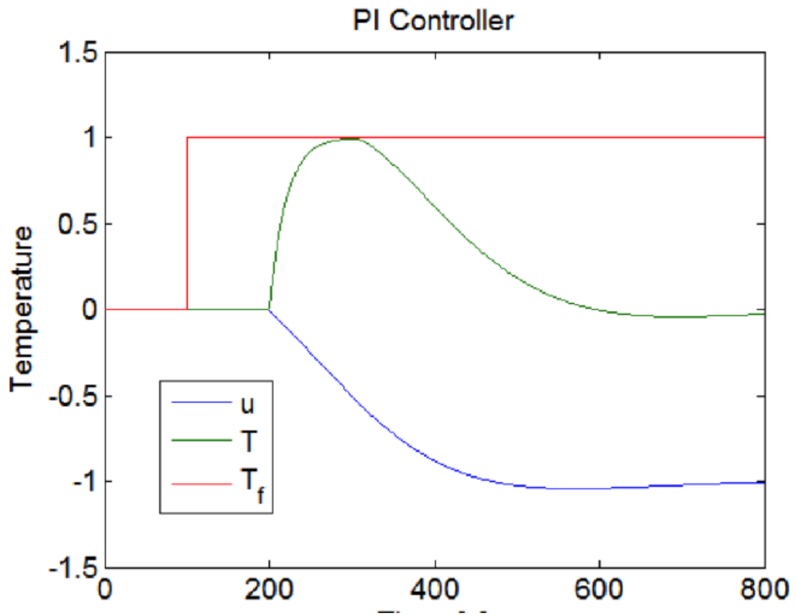

```
Kc=0.5; taui=9999; % P-control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
```

SIMC PI control

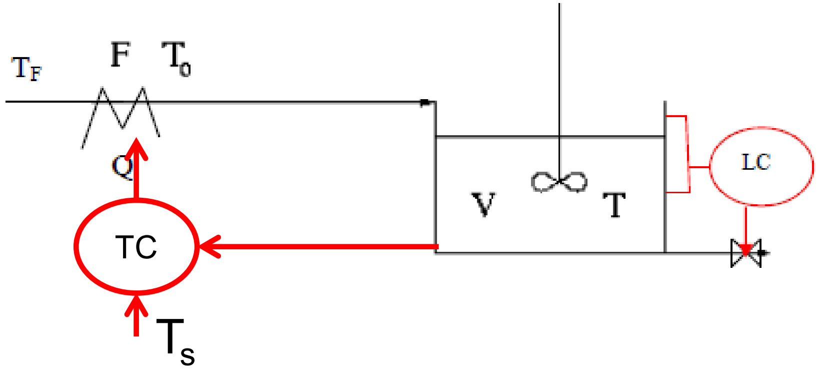
11) SIMC PI tuning rule with $\tau_c = \theta = 100$.

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

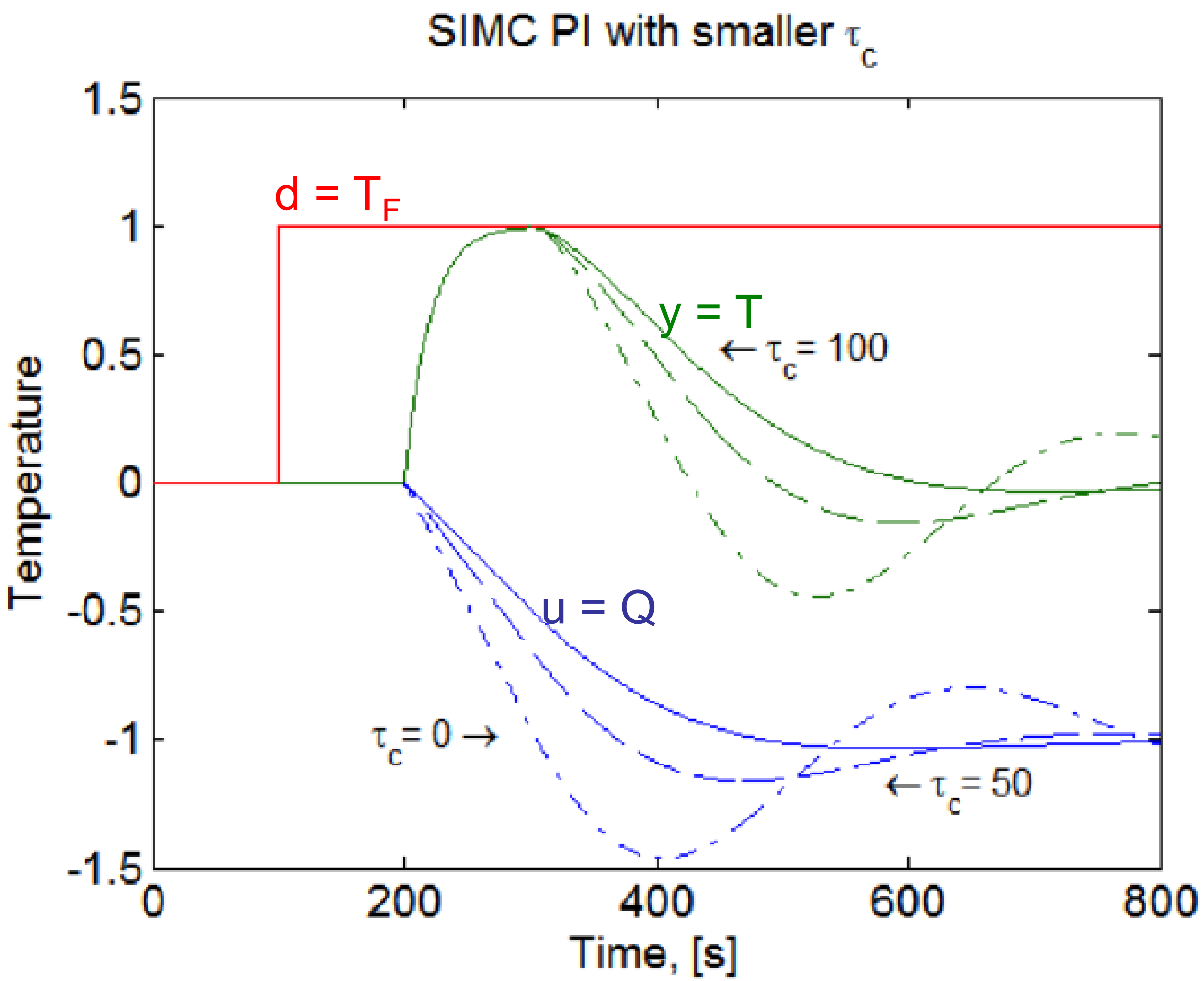
u = Q
y = T
d = T_F



No offset

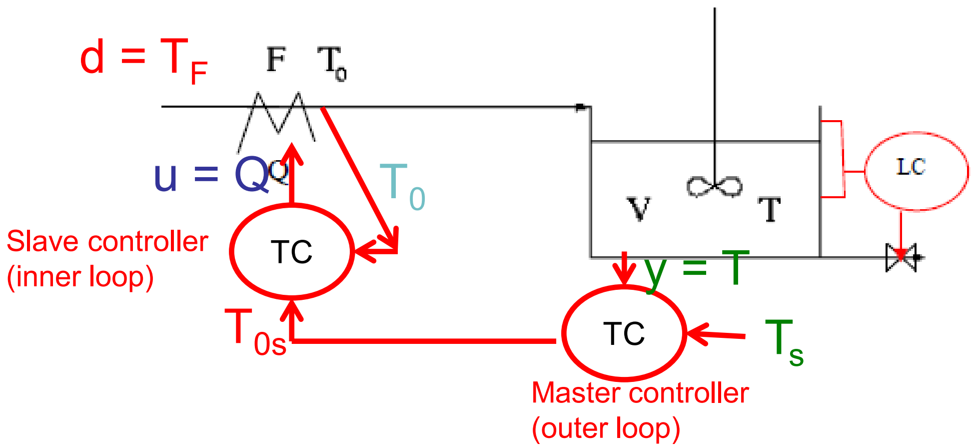


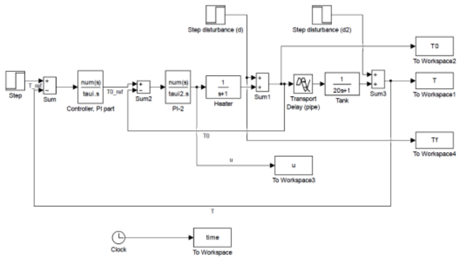
Kc=0.1; taui=20; % SIMC PI-control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])



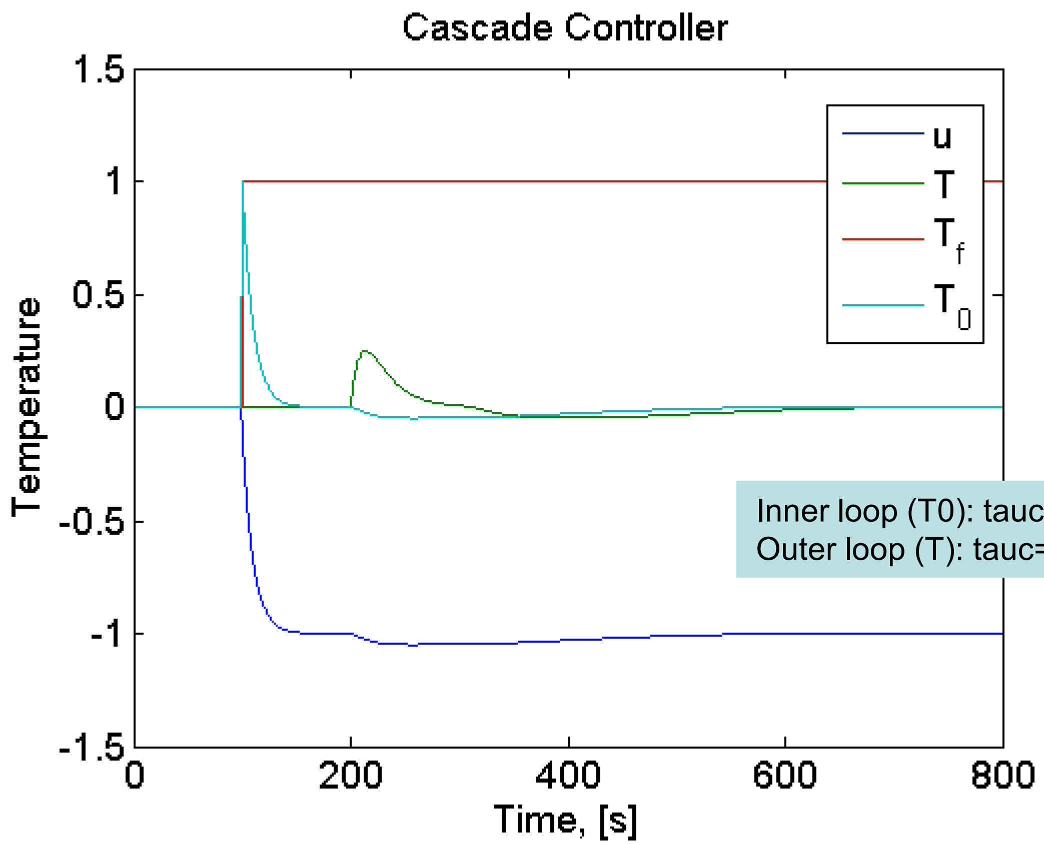
Recommend: ¿$_c$=delay µ=100s because it is more robust and gives no overshoot in u

Measure also T_0: Cascade control is much better





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



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

Lab: The experimental setup

This is the «Whistler»

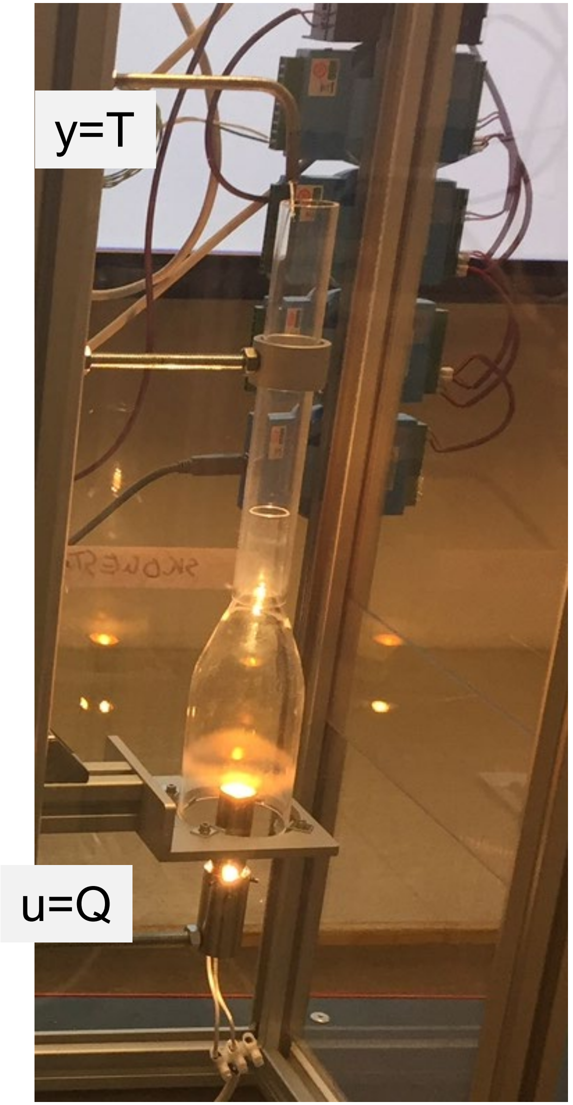
y=T [C] (at top)
u=Q [0-1] (at bottom)

First we did a step response experiment where u was increased from 0 to 1 (manual vontrol). The temperature y=T increased from 20C to 54C (new steady state). This gives k=68. The dynamics are quite slow because it takes time to heat up the glass. , θ=5s, τ =120s

From this we obtained the model parameters and SIMC tunings (with $\tau_c=\theta=5s$)

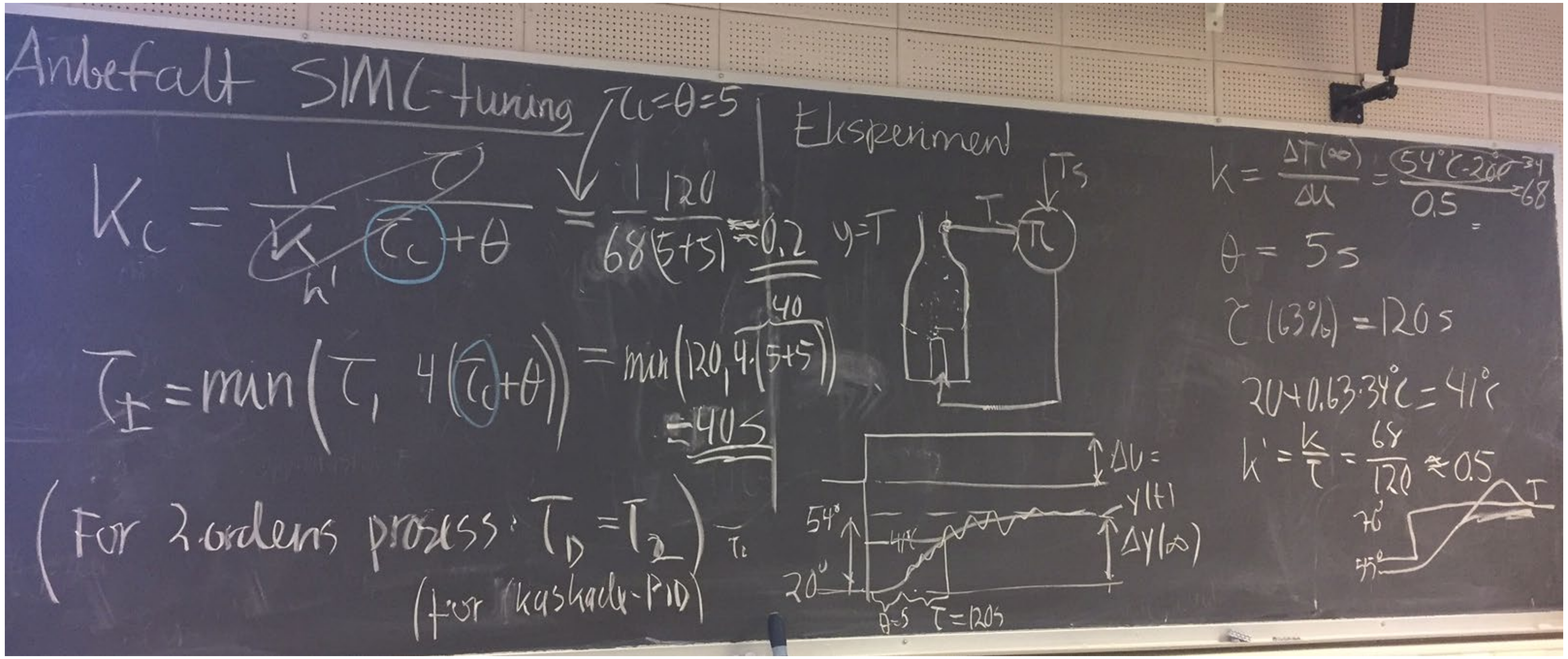
We then put it into automatic and increased the setpoint to 70C. The input (u=Q) increased immediately to max=1, and we should then have stopped the integration («anit windup») but we had forgotten to do this and this is why you can see that u=Q stayed at max=1 even after y=T has passed the setpoint…. Not so good… but eventually we see that it was working well.

Note: Need to use «anti-windup» to avoid that the integral action in the controller keeps increasing u when it actually has saturated.

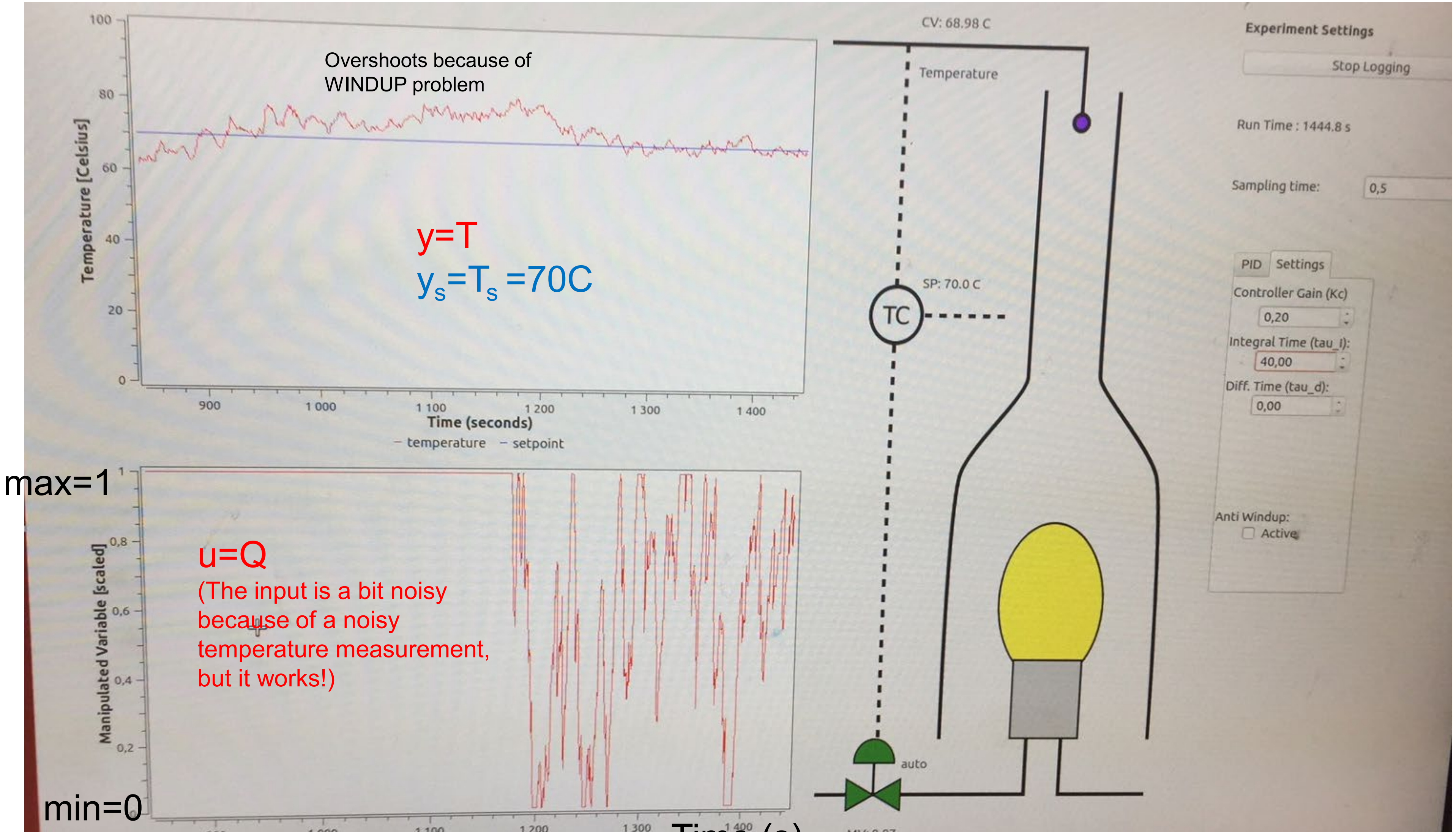


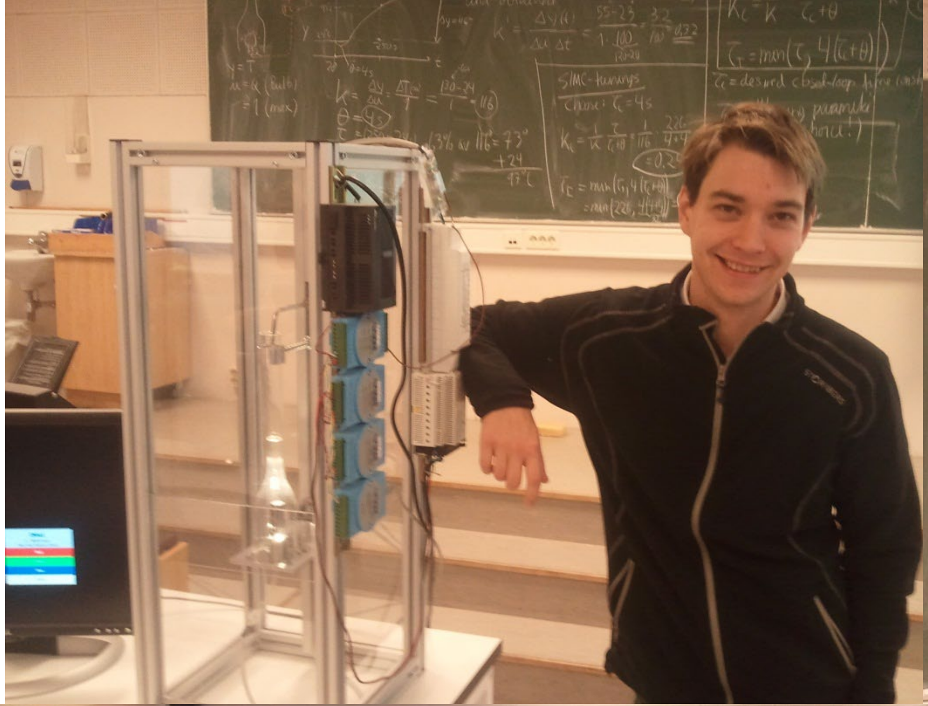
The model. Step response: $k=68$, $\theta=5s$, $\tau=120s$

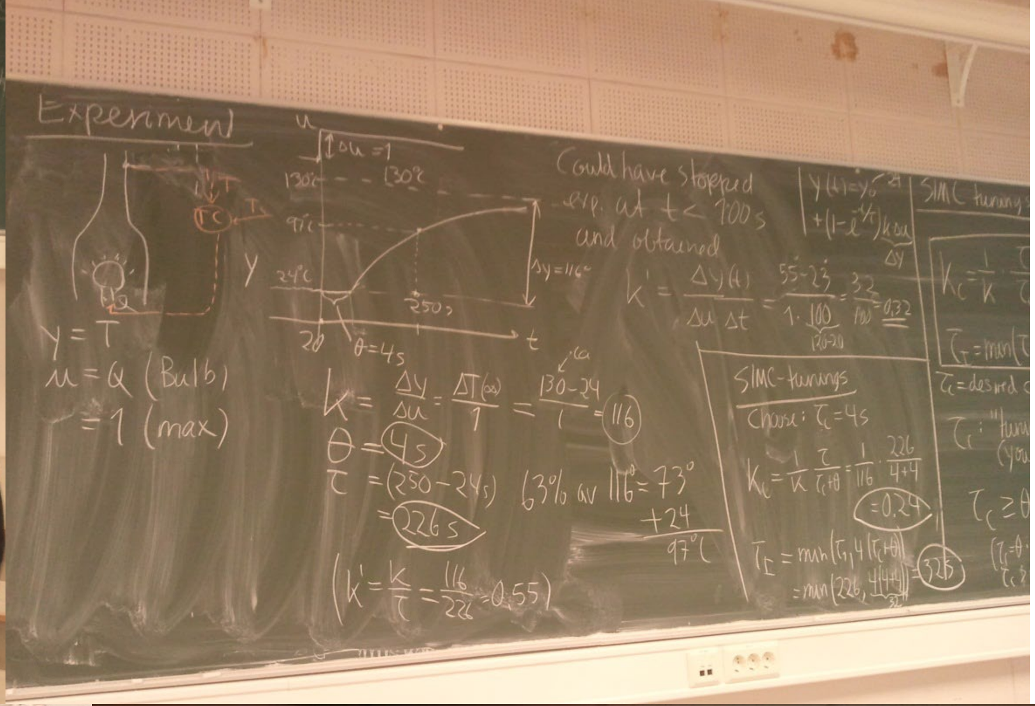
The controller. SIMC (with $\tau_c=\theta=5s$): $K_c=0.2$, $\tau_I=40s$

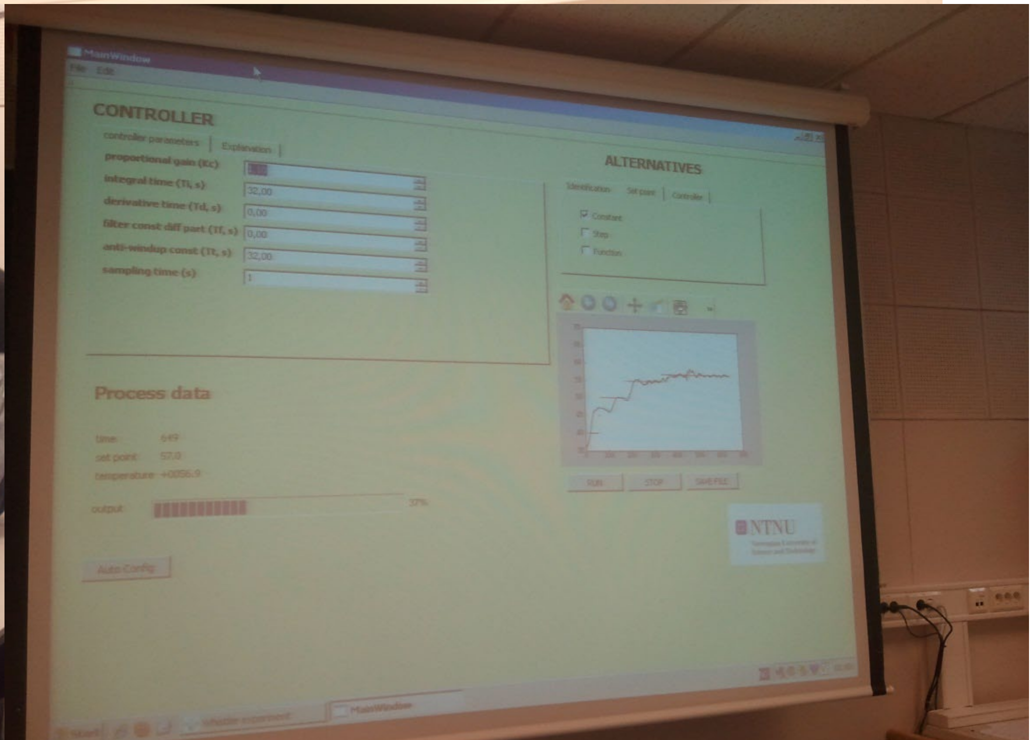


The closed-loop response






Experiment
Could have stopped exp. at t < 100 s and obtained
SIMC-tunings
Chose: τc = 4s
y = T
u = Q (Bulb)
= 1 (max)
63% av 116 = 73°
+24
97°C


CONTROLLER
ALTERNATIVES
Process data
NTNU