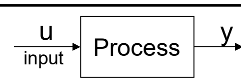
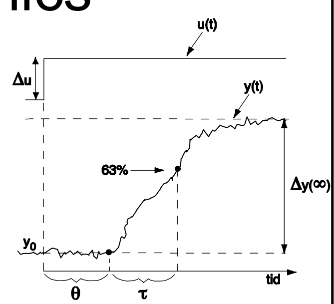
Dynamics and PID control

Sigurd Skogestad



Process dynamics



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

- Time constant τ: Often equal to residence time = V[m3]/q[m3/s] (but not always!)

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

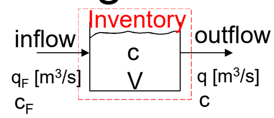
$$\text{Mass/energy [kg/s; J/s]:} \quad \frac{d}{dt}\text{Inventory} = \text{Inflow - Outflow}$$

$$\text{Component [mol/s]:} \quad \frac{d}{dt}\text{Inventory} = \text{Inflow - Outflow + Gen. by reaction}$$

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

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:

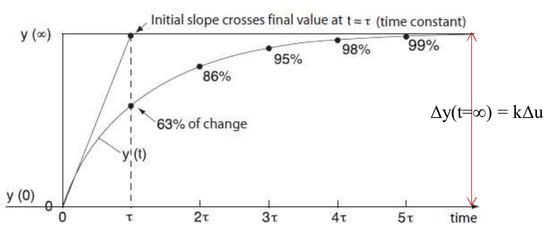
Mass $\frac{d(\rho V)}{dt} = \rho q_F - \rho q \quad [\text{kg/s}]. \quad \rho V \text{ constant} \Rightarrow q = q_F$

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

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

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



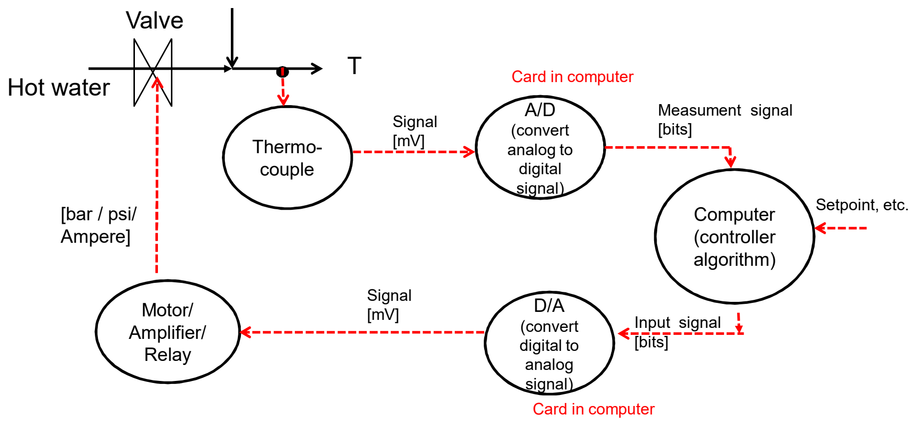
t/τ	$1-e^{-t/\tau}$	Value	Comment
0	$1-e^{0}=$	0	
0.1	$1-e^{-0.1}=$	0.095	
0.5	$1-e^{-0.5}=$	0.393	
1	$1-e^{-1}=$	0.632	63% of change is reached after time $t=\tau$
2	$1-e^{-2}=$	0.865	
3	$1-e^{-3}=$	0.950	
4	$1-e^{-4}=$	0.982	98% of change is reached after time $t=4\tau$
5	$1-e^{-5}=$	0.993	
∞	$1-e^{-\infty}=$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

Remember for first order response:
1. Starts increasing immediately (would reach new steady state after time τ if it kept going)
2. Reaches 63% of change after time τ.
3. Approaches new steady state exponentially (has for practical purposes reached new steady state after about 4τ)

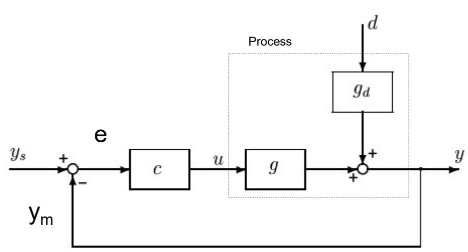
*A more general standard form for linear systems is the *state space* form (in deviation variables): $\frac{dx}{dt} = Ax + Bu, \quad y = Cx + Du, \quad x(0) = 0$
Our case: $A = -1/\tau, B = k/\tau, C = 1, D = 0$

Feedback control

Control systems elements:



Block diagram



Block diagram of negative feedback control

Lines are signals ("information"):

y = controlled variable (CV)

y_m = measured CV

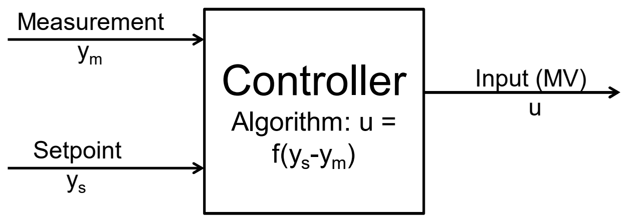
y_s = setpoint (SP)

$e = y_s - y_m$ = control error

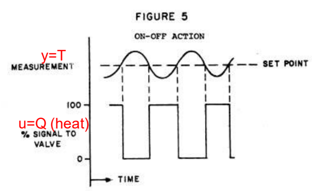
u = manipulated variable (MV)

C = Feedback Controller = ?

Feedback controller



Simplest controller algorithm: On/off controller.
Problem: cycles



Industry: Standard algorithm for SISO controllers: PID
Industry: Standard for multivariable control: MPC (model predictive control)

Hjelp til å velge varmeprodukter

Vårt kjære lille land her oppe i nord har tradisjonelt vært med mye snø og kulde - og er det fortsatt. Riktig valg av oppvarming er viktig både med tanke på strømregning og innemiljø.

Ideell innetemperatur

Vi har etter hvert vendt oss til en høy inne- temperatur, gjerne opp mot 25 grader. Vi tilbringer det meste av vår tid innendørs, og derfor er inneklimaet viktig. Undersøkelser har vist at mennesker trives og fungerer best når innetemperaturen er mellom 19 og 22 grader. Ved høyere temperaturer vil den relative luft- fuktigheten bli mindre, og luften føles tørrere. Vi bør også holde så jevn temperatur som mulig. Beha benytter elektroniske termostater med proporsjonalregulering på sine miljøovner. Det vil si at termostatene er meget nøyaktige med kort tid mellom inn- og utkobling av varme- elementene. De er faktisk så nøyaktige som +/- 0,3 grader ved 20°C.

Prinsippskisse for termostatregulering

Mechanical thermostat = On/Off-control (cycles)
Electronic controller (thermostat) = P-control (should give small offset)

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:

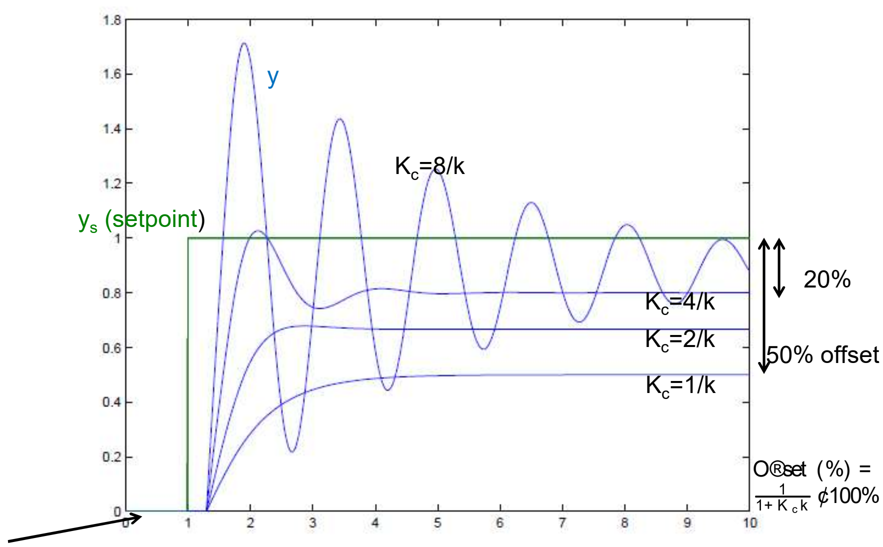
1. Get steady-state offset (especially if K_c is small)

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

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.

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

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

τ_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 τ_I = 99999 (large!) to turn off integral action)

Note 3: Integral action is also called «reset action» since it «resets» the bias.
«Update bias u_0 at every Δt»: $u(t) = u_0(t) + K_c e(t)$
where $u_0(t) = \underbrace{u_0(t-\Delta t)}_{\text{old } u_0} + \frac{K_c \Delta t}{\tau_I} e(t)$

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 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/ τ_I

Etc.....

NOTE: Always check the manual for your controller!

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 :

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

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

Conclusion: Digital PID implementation

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

Ikke pensum sep.tek

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

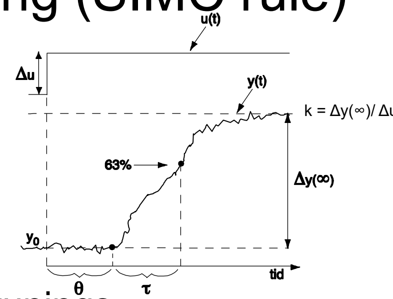
(d) Increase a bit (~ factor 2)

(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
  - $k = \Delta y(\infty) / \Delta u$ – process gain
  - τ - process time constant (63%)
  - θ - process time delay



- Proposed SIMC controller tunings

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

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

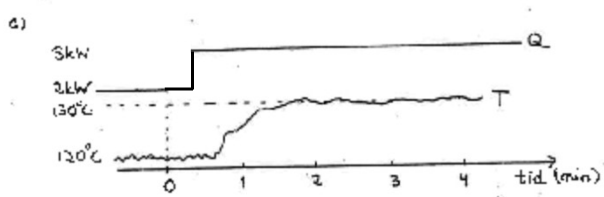
τ_c = desired response time with control (tuning parameter!).
- Choose $\tau_c = \theta$ (delay) for "tight" control
- Choose $\tau_c > \theta$ for smoother control (but $K_c \geq \frac{\Delta u_{max}}{\Delta y_{max}}$)

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)

Eksamen i fag 52320 Kjemiteknikk 2, 31. mai 1991

INNSTILLING ("TUNING")
AV PID - REGULATOR.

c)



Du har fått i oppgave å stille inn en regulator for å holde temperaturen (T) i en tank konstant ved bruk av elektrisk effekt (Q). Resultatet av et sprangresponsforsøk er vist i figuren.

a) Bestem prosessens dødtid (θ), tidskonstant (τ) og forsterkning (k).

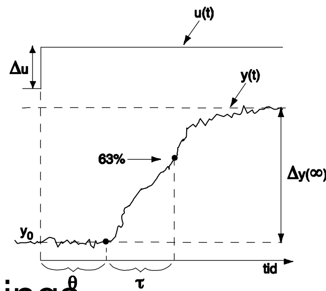
b) For slike prosesser med dødtid brukes ofte PID-regulatorer.

Hva kalles parametrene τ_I, τ_D, K_c? Bestem rimelige verdier i ditt tilfelle.

Kommentar: Det tar tiden $\tau + \theta$ før responsen når 63% av sin endelige verdi. Forsterkning: $k = \Delta T(t = \infty)/\Delta Q$.

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

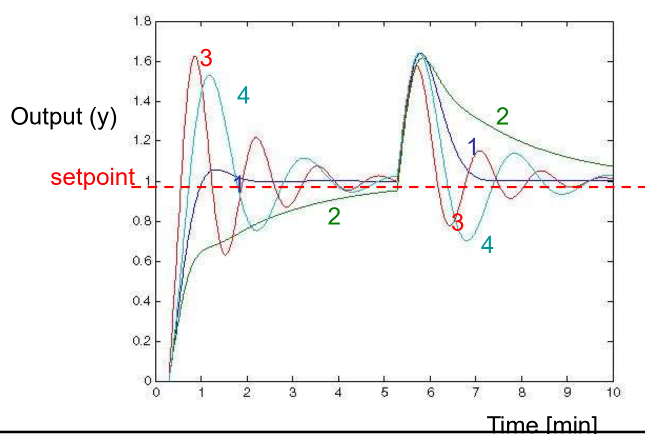
Select $\tau_c = \theta = 0.3$ min ("tight" control):

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

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

Simulation PID control

- Setpoint change at t=0 and 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)
  –



Comments tuning

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

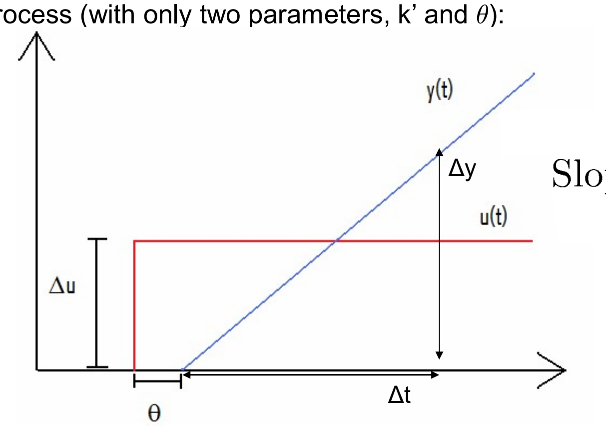
2. Common mistake: Wrong sign of controller!
   - Controller gain (K_c) should be such that controller *counteracts* changes in output
   - Need negative sign around the loop ("***negative feedback***")
   - Two ways of achieving this:

   - (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)
   - (Many real control systems:) ***Always use K_c positive*** and select between
     - "Reverse acting" when process gain (k) is positive
       - because MV (u) should go down when CV (y) goes up
     - "Direct acting" when k is negative

     - WARNING: Be careful and read manual! Some reverse these definitions (wikipedia used to do it, but I corrected it)
     - Oct. 2009: http://en.wikipedia.org/wiki/Controller_(control_theory)

3. Integrating («slow») process:

If the response is not settling after approximately 10 times the delay (so τ/θ is large), then you can stop the experiment and approximate the response as an integrating process (with only two parameters, k' and θ):



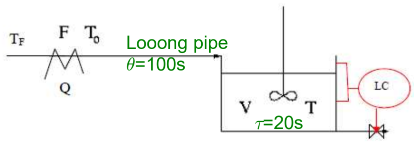
Slope, $k' = \frac{\Delta y}{\Delta t \cdot \Delta u}$

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

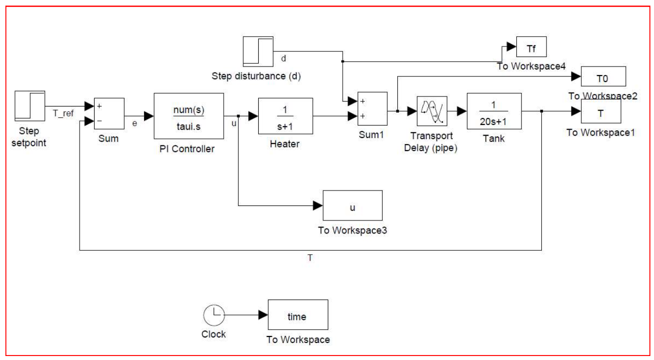
$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta}$

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

Example: Similar to shower process



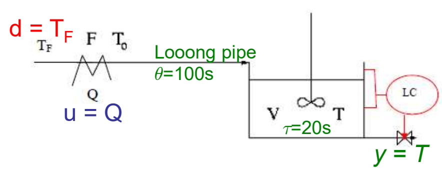
u = Q
y = T
d = T_F

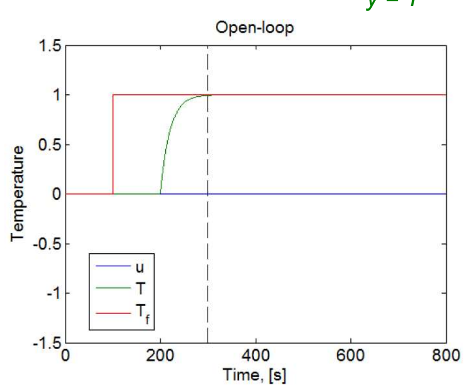


Simulink model: tunepid1_ex1

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

Disturbance response with no control





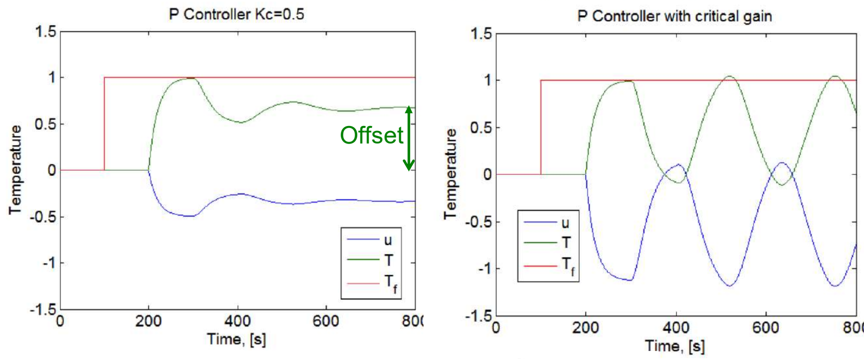
u = Q
y = *T*
d = T_F

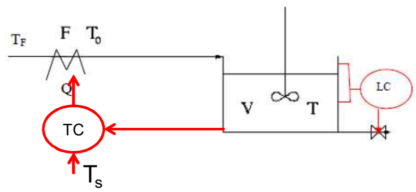
```
Kc=0; taui=9999; % no control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])
```

P-control

u = Q
y = T
d = T_F

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





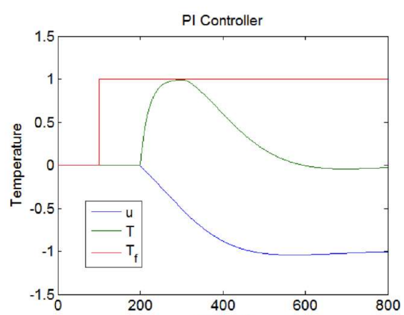
Kc=0.5; taui=9999; % P-control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])

SIMC PI control

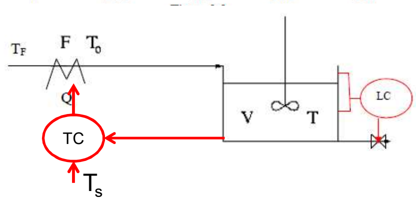
u = Q
y = T
d = T_F

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

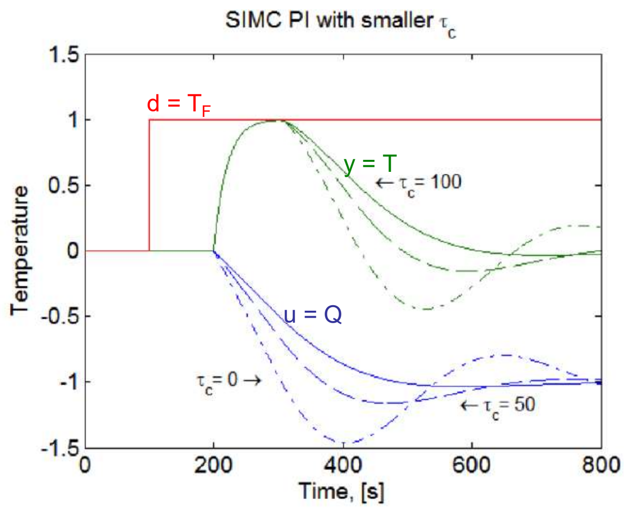
$$K_c = (1/k)\tau_1/(\tau_c + \theta) = 20/200 = 0.1; \tau_I = \min(\tau_1, 4(\tau_c + \theta) = 20$$



No offset

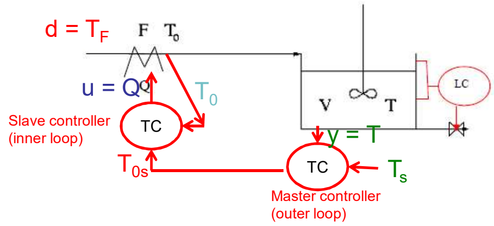


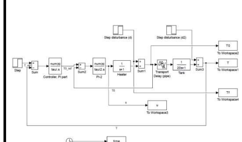
Kc=0.1; taui=20; % SIMC PI-control
%start simulation (press green button)
plot(time,u,time,T,time,Tf), axis([0 800 -1.5 1.5])

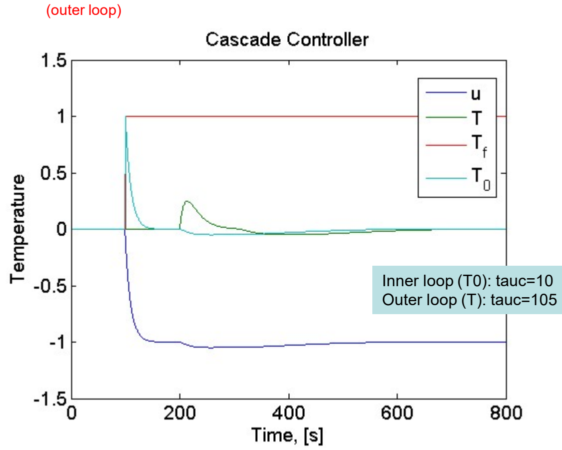


Recommend: τ_c=delay θ=100s because it is more robust and gives no overshoot in u

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])
```

K5, 11 Nov. 2016

The experimental setup

This is the «Whistler»

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

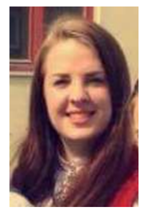
First we did a step response experiment where u was increased from 0 to 1 (manual vontrol). The temperature y=T increased from 20C to 54C (new steady state). This gives k=68. The dynamics are quite slow because it takes time to heat up the glass. , θ=5s, τ =120s

From this we obtained the model parameters and SIMC tunings (with τ_c=θ=5s)

We then put it into automatic andf 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 oassed the setpoint…. Not so good… but eventually we see that itr was working well.

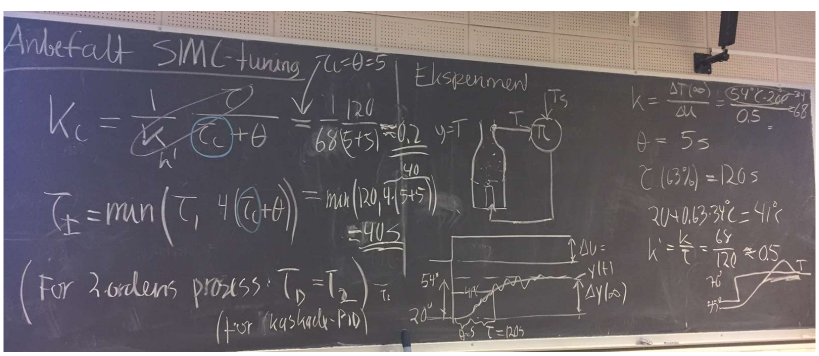
This can be confirmed by Ida who was the ONLY student who stayed behind to check how things went. Thanks, Ida!

Thanks to Tamal Das



Thanks to Ida

The model. Step response: k=68, θ=5s, τ =120s
The controller. SIMC (with τ_c=θ=5s): K_c=0.2, τ_I=40s



The closed-loop response

Ja, reguleringen virket etter hvert! - noe Ida kan bekrefte