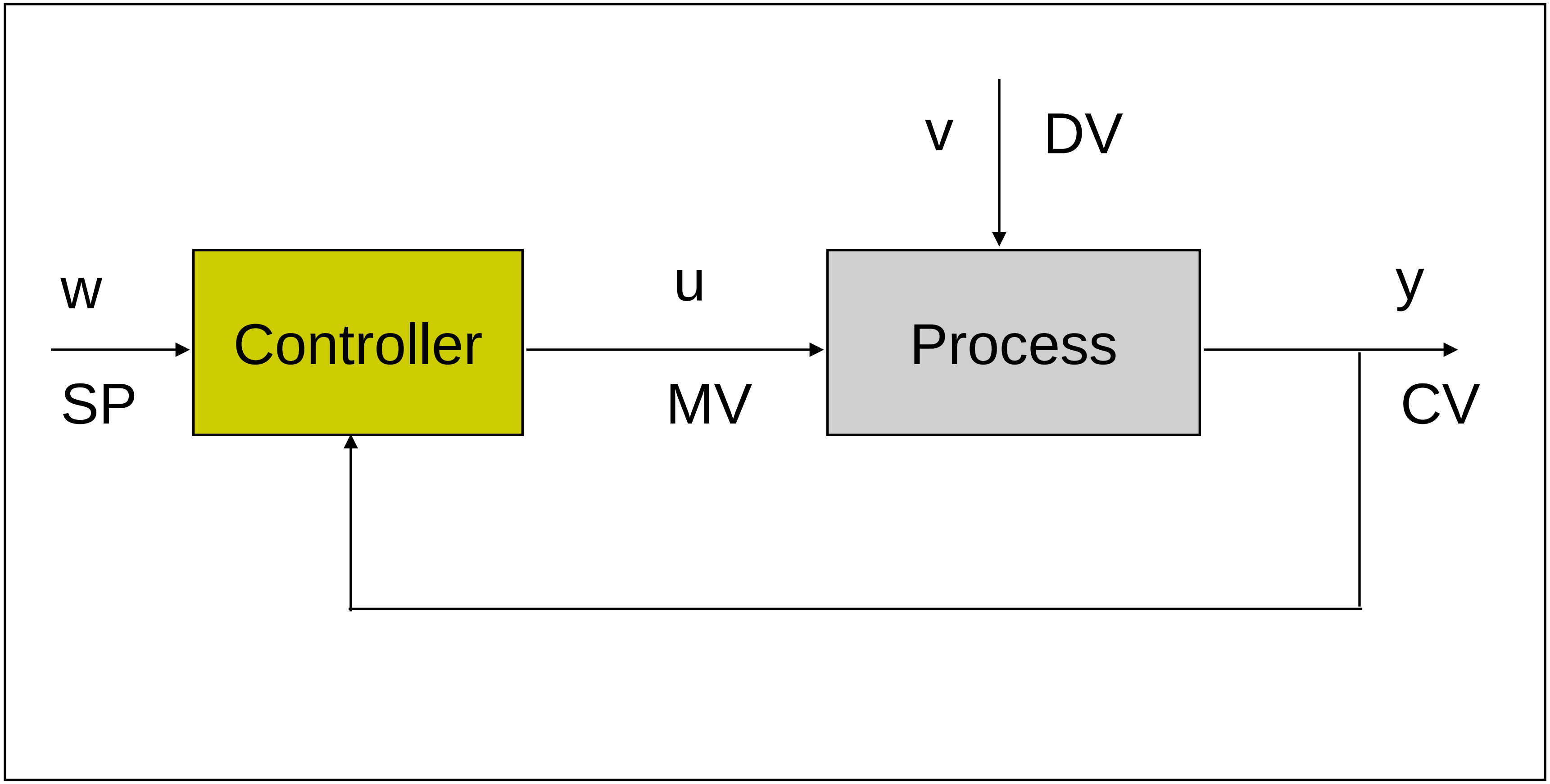
PID controllers and tuning

Prof. Cesar de Prada
Dpt. Systems Engineering and Automatic Control
University of Valladolid, Spain
prada@autom.uva.es
http://www.isa.cie.uva.es/~prada/

Outline

- PID controller
- Types of PID controllers
- Tuning criteria
- Automatic tuning

Control loop



The PID controller

$$e(t) = w(t) - y(t)$$

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int e(\tau)d\tau + T_d\frac{de}{dt}\right)$$

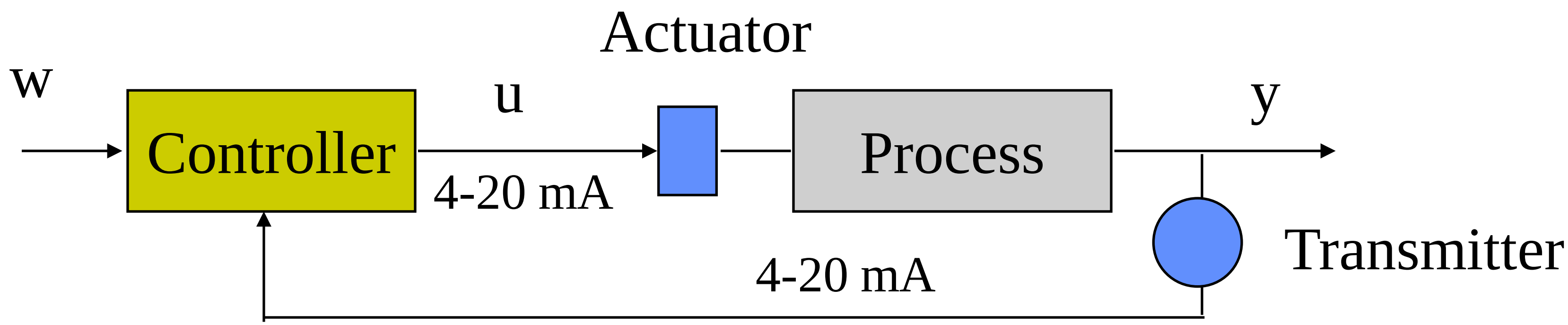
- **Signal based controller**, no explicit process knowledge is incorporated
- 3 tuning parameters K_p, T_i, T_d
- Many different implementations

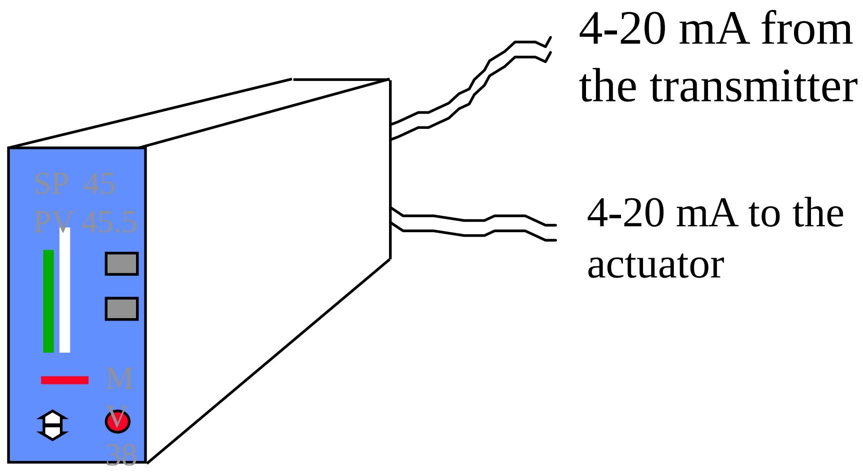
A bit of history

- ✓ 1911 – First application of a PID controller by Elmer Sperry.
- ✓ 1920 – First patent of a PI controller
- ✓ 1933 - Taylor Double-response plus Fulscope (Model 56R Fulscope) with adjustable P and I componenets
- ✓ 1925-1935: Widespread use of the PID in industry thanks to the action of instrumentation companies such as Foxboro and Taylor. 75.000 automatic controllers sold in the USA

1939 – First fully adjustable
commercial controller:
Fulscope 100
from Taylor Instruments

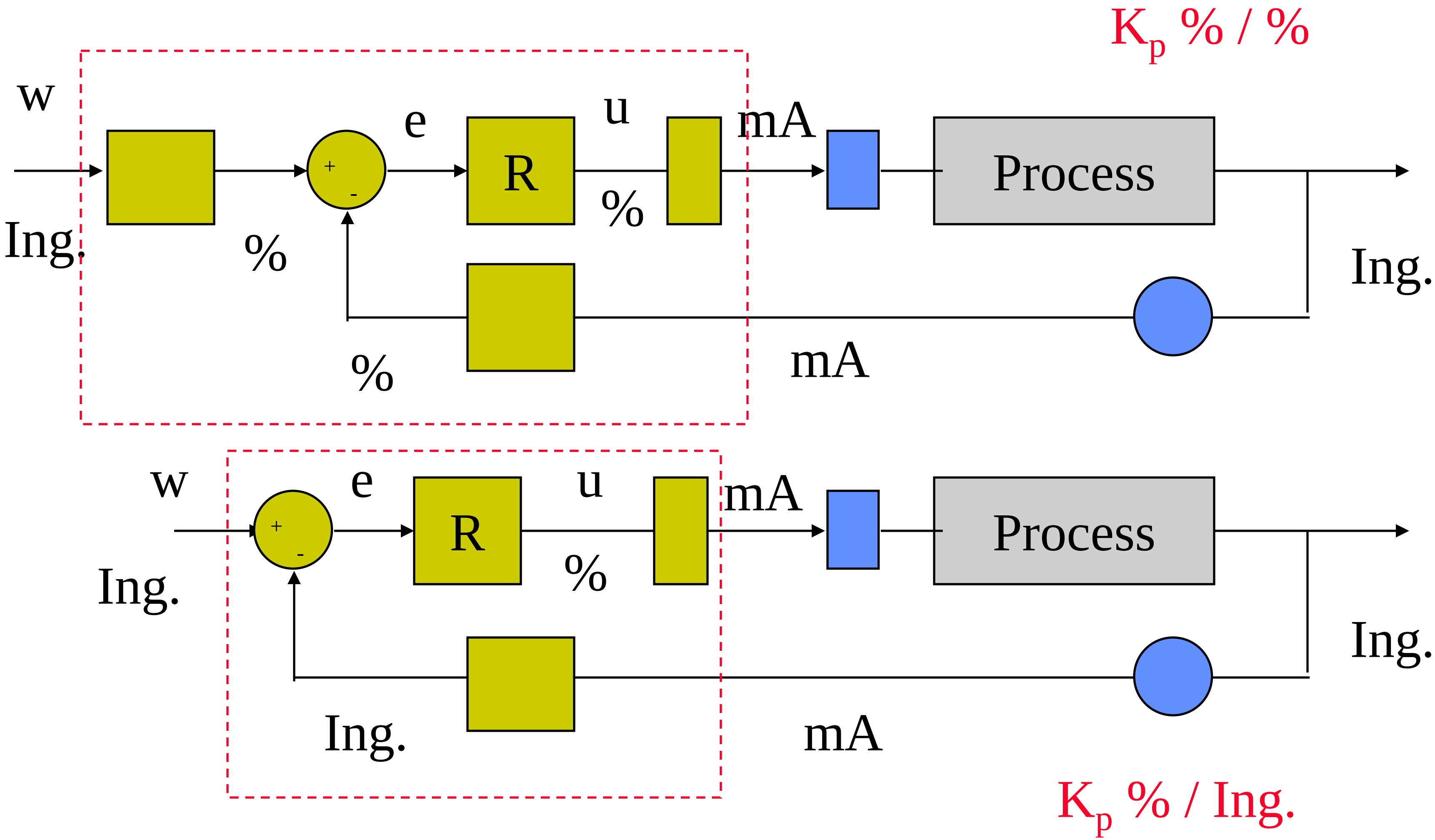
Normalised I/O signals





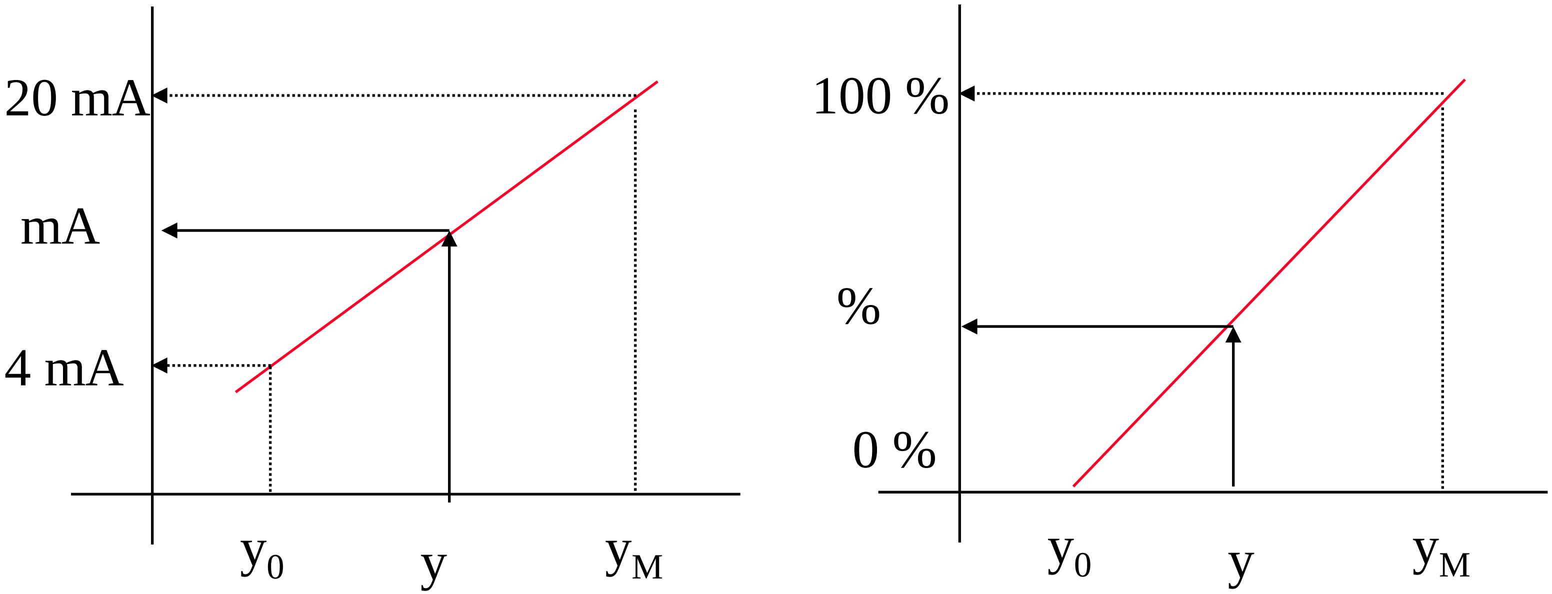
Loop controller

Two options



Kp % / %
w
e
u
mA
R
Process
Ing.
%
%
+
-
Ing.
%
mA
w
e
u
mA
R
Process
Ing.
%
+
-
Ing.
Ing.
mA
Kp % / Ing.

Conversion formulas y,w



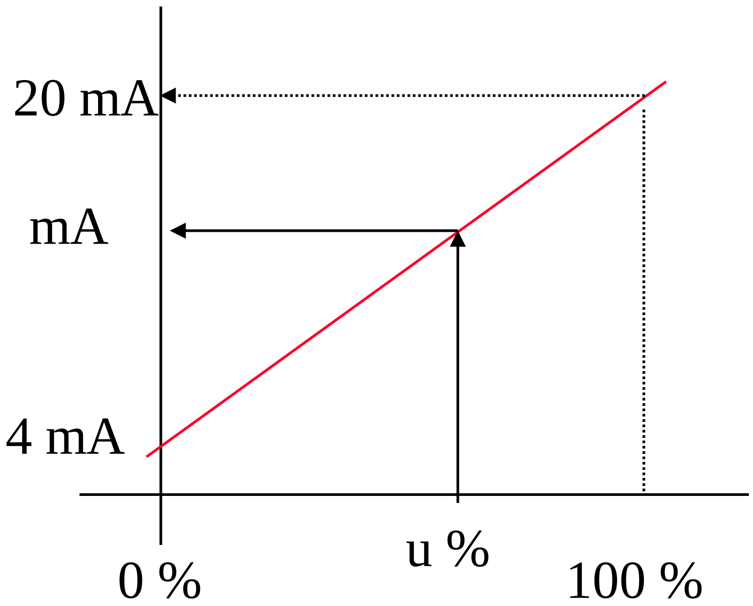
Span=y_M - y_0

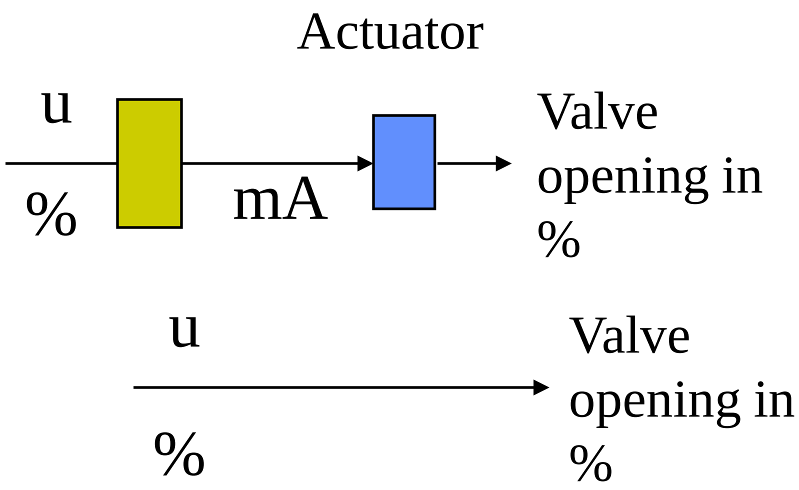
$$mA = \frac{16}{span}(y - y_0) + 4$$

$$\% = \frac{100}{span}(y - y_0)$$

$$\% = \frac{100}{16}(mA - 4)$$

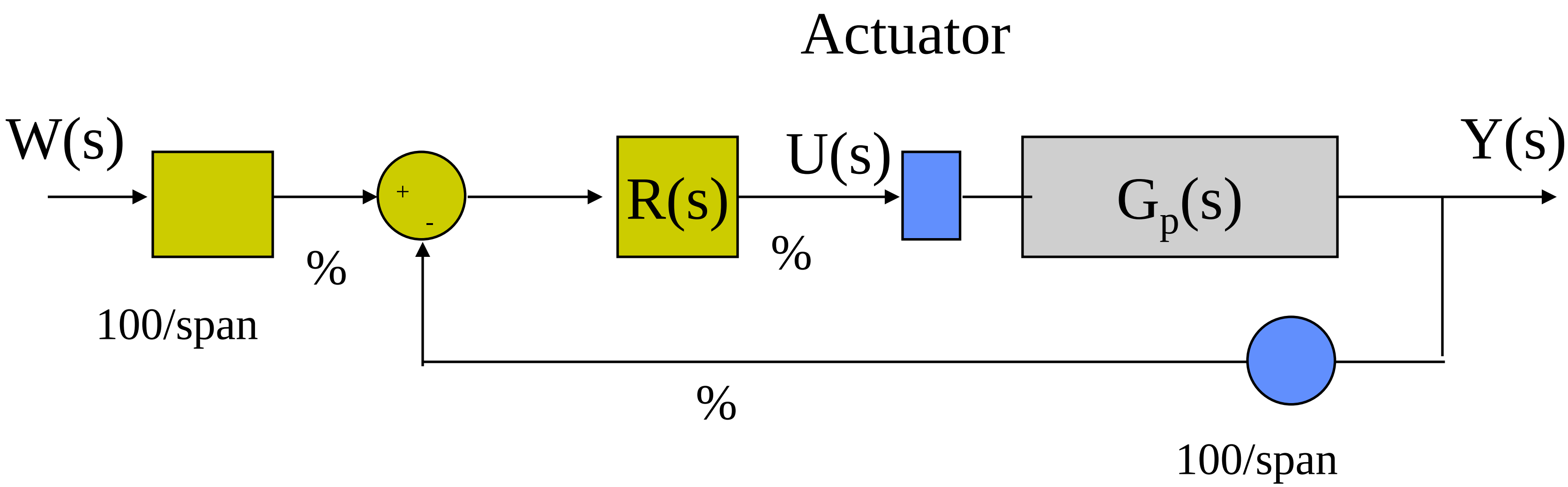
Conversion formulas u





$$mA = \frac{16}{100}\% + 4$$

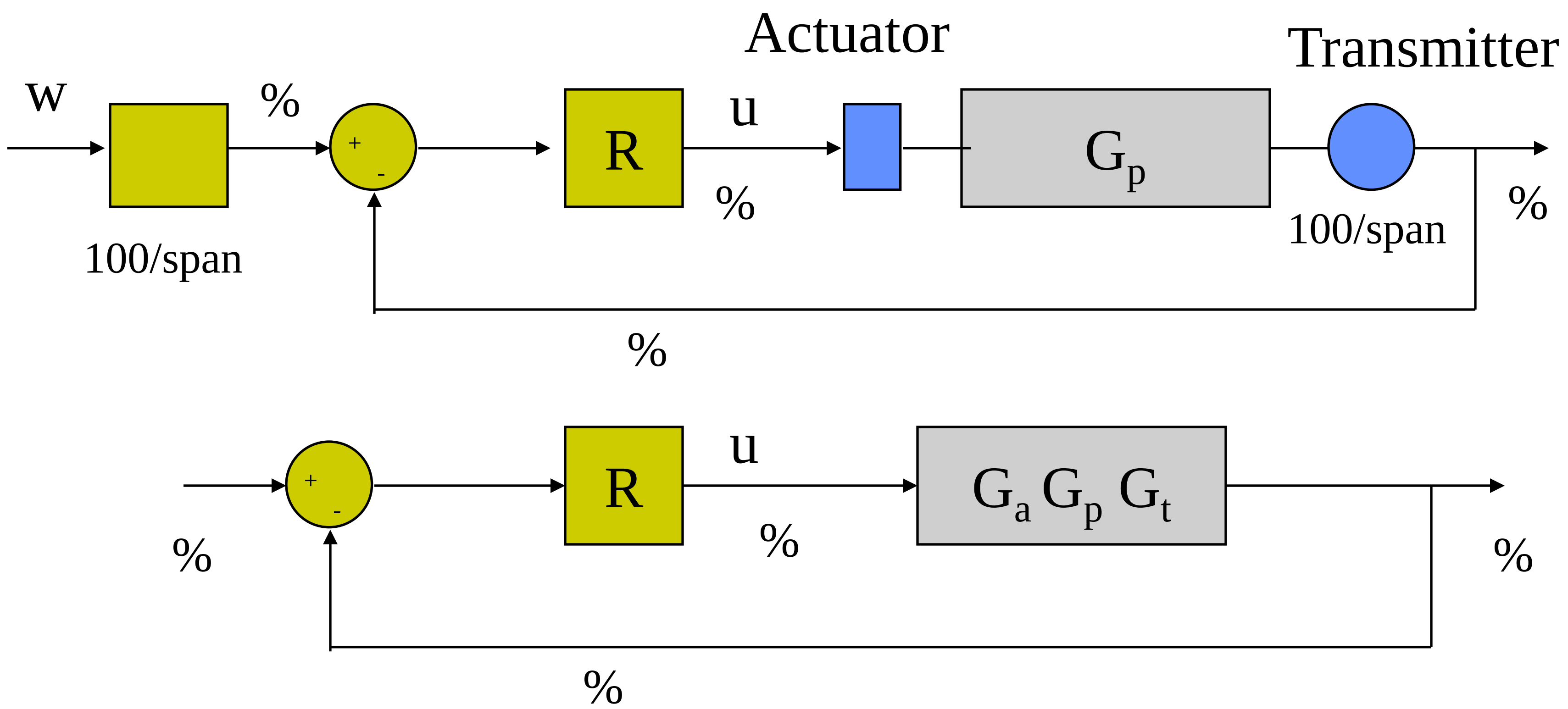
Units



Input and output regulator signals usually are expressed in terms of % of transmitter and actuator respectively

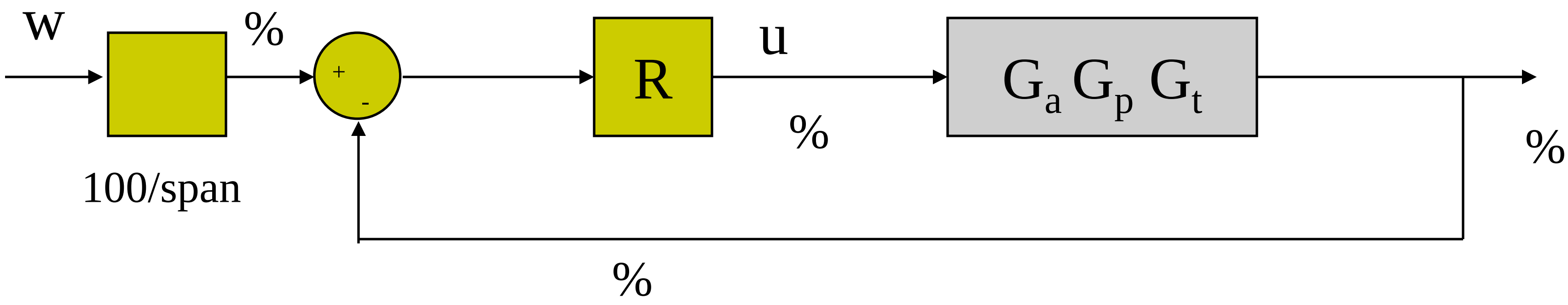
Conversion factors in the controller should correspond to the calibration of the instruments

Loop analysis



Dynamics of transmitter and actuator must be included if they are relevant

Loop analysis (1)



$$Y(s)\% = \frac{G_a G_p G_t R}{1 + G_a G_p G_t R} W(s)\% \qquad Y(s)\frac{100}{span} = \frac{G_a G_p G_t R}{1 + G_a G_p G_t R} W(s)\frac{100}{span}$$

G_a Ing'/%

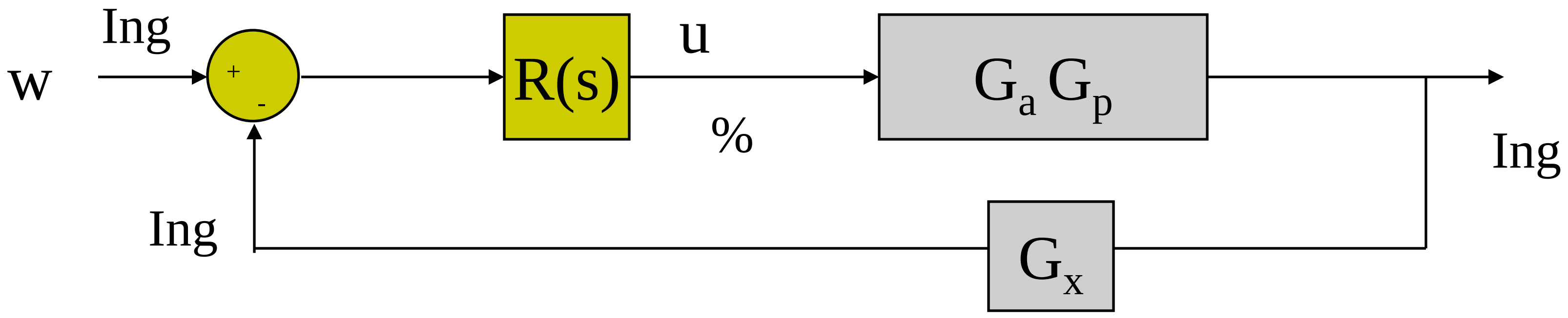
G_p Ing/Ing'

G_t %/Ing

R % / %

The output is the signal provided by the transmiter, measured variable, not the actual controlled variable of the process

Loop analysis (2)



$$Y(s) = \frac{G_a G_p R}{1 + G_a G_p G_x R} W(s)$$

G_a Ing'/%

G_p Ing/Ing'

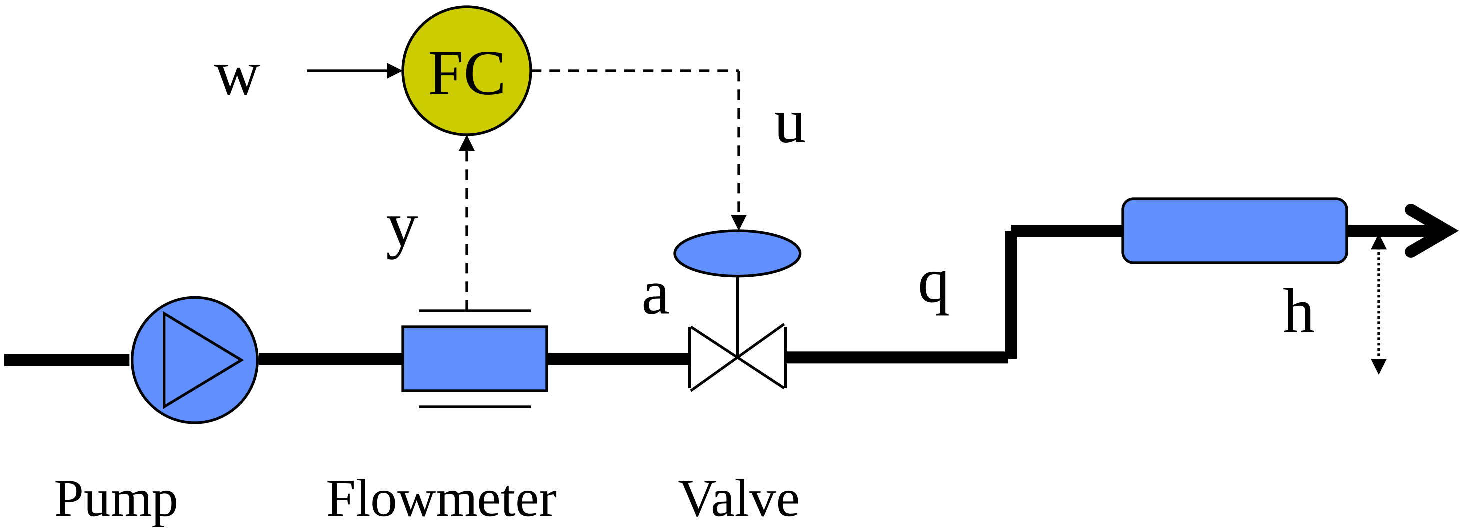
G_x adimensional

R % / Ing

The process output is the controlled variable

G_x has gain 1 and incorporates the transmitter dynamics

Flow control loop



Flowmeter: 0-50 m^3/h 4-20mA

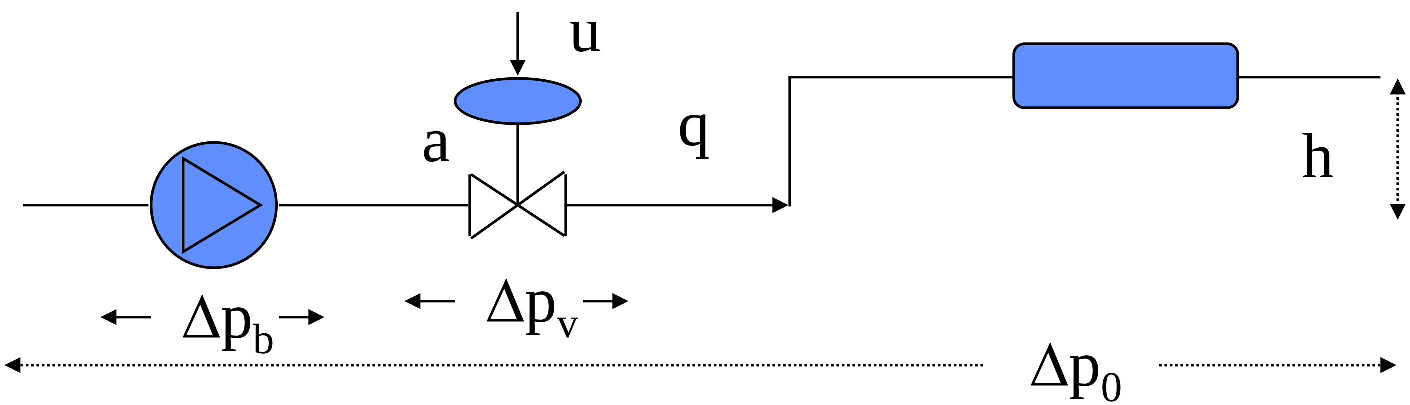
Model

$$\frac{d\,mv}{dt} = A(\Delta p_0 + \Delta p_b) - A\Delta p_v - AfL\rho v^2 - Ah\rho g$$

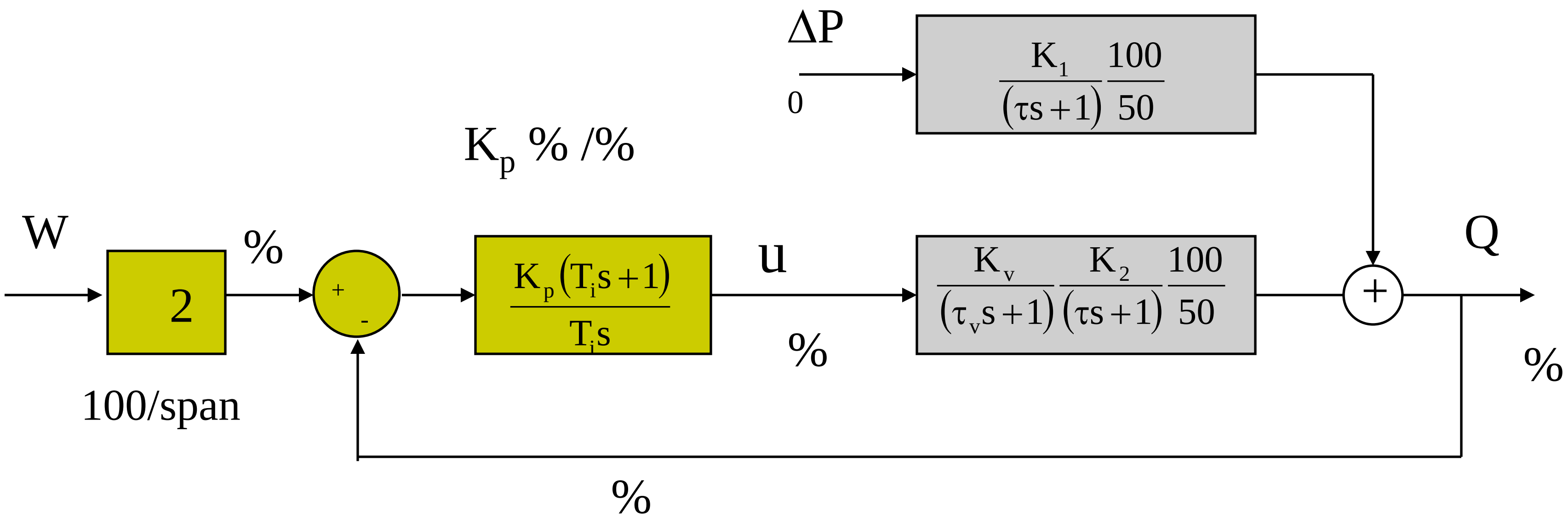
$$\Delta p_v = \frac{1}{a^2 C_v^2}\rho q^2 \qquad q = Av$$

$$\tau \frac{d\,\Delta q}{dt} + \Delta q = K_1 \Delta(\Delta p_0) + K_2 \Delta a$$

$$\tau_v \frac{d\,\Delta a}{dt} + \Delta a = K_v \Delta u$$



Block diagram



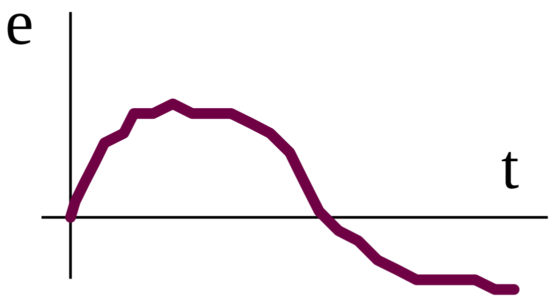
Transmitter dynamic is not considered

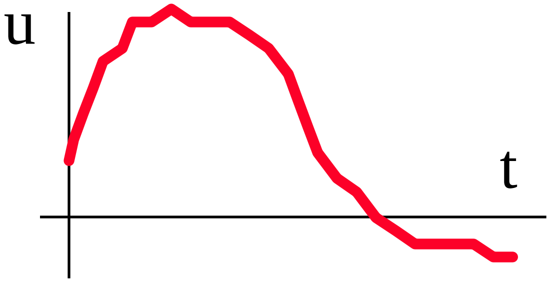
PID parameters

- K_p gain / Proportional term
  - – % span control / % span controlled variable
  - – Proportional band PB=100/ Kp
- T_i integral time / Integral term
  - – minutes o sg. (per repetition) (reset time)
  - – repetitions per min = 1/ Ti
- T_d derivative time / Derivative term
  - – minutes o sg.

Proportional controller P

$$u(t) = K_p e(t) + \text{bias}$$



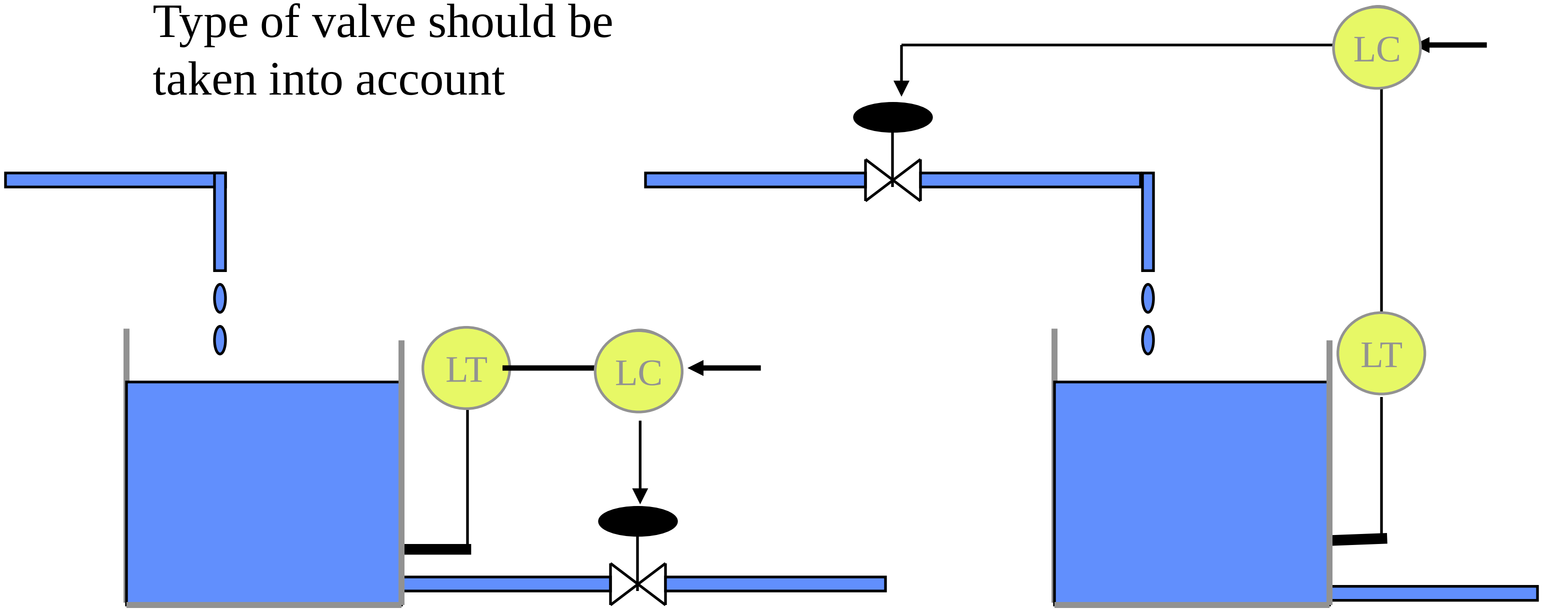


An error of x % creates an action of
K_p x % on the actuator

bias = manual reset (CV = SP)

Direct / Reverse Acting

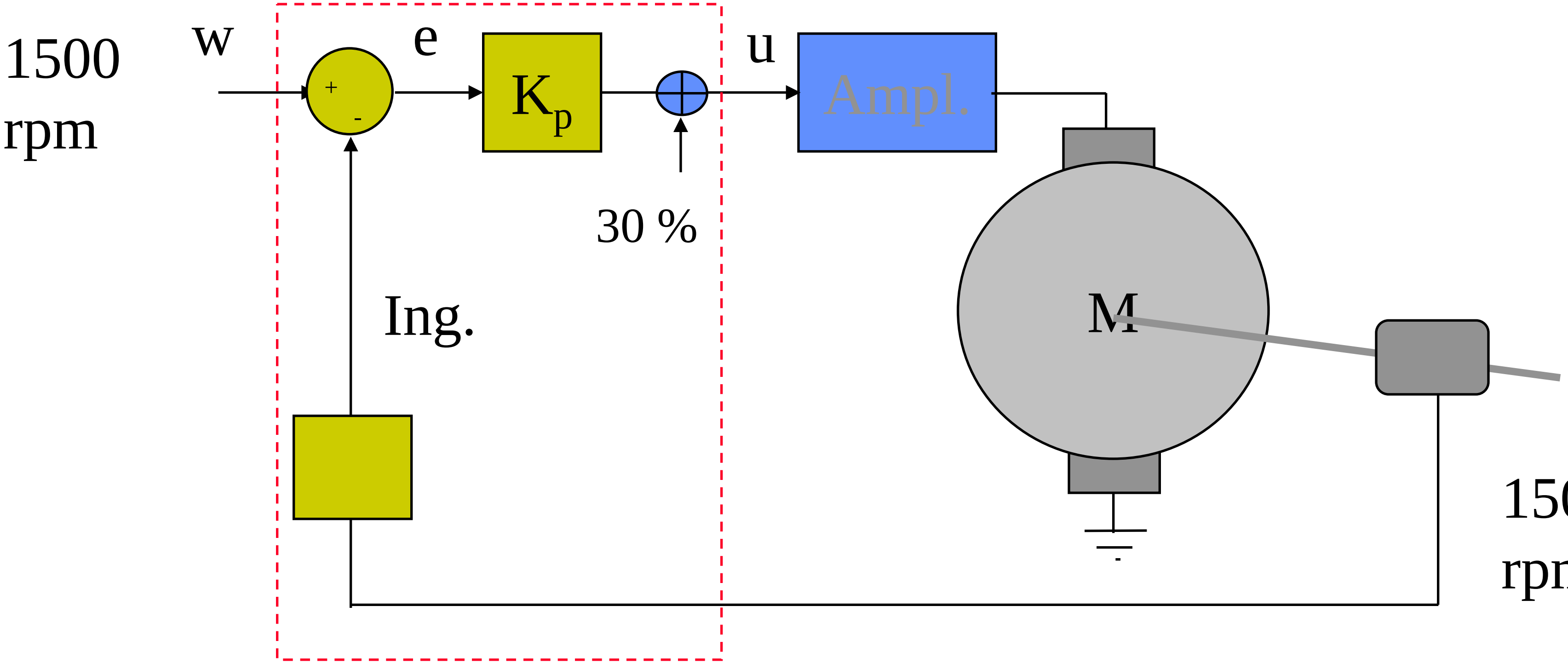
Type of valve should be taken into account



Direct acting controller $K_p < 0$ Reverse acting controller $K_p > 0$

$u(t)=K_p(w-y)$ if *y* increases, then *u* decreases if K_p is positive

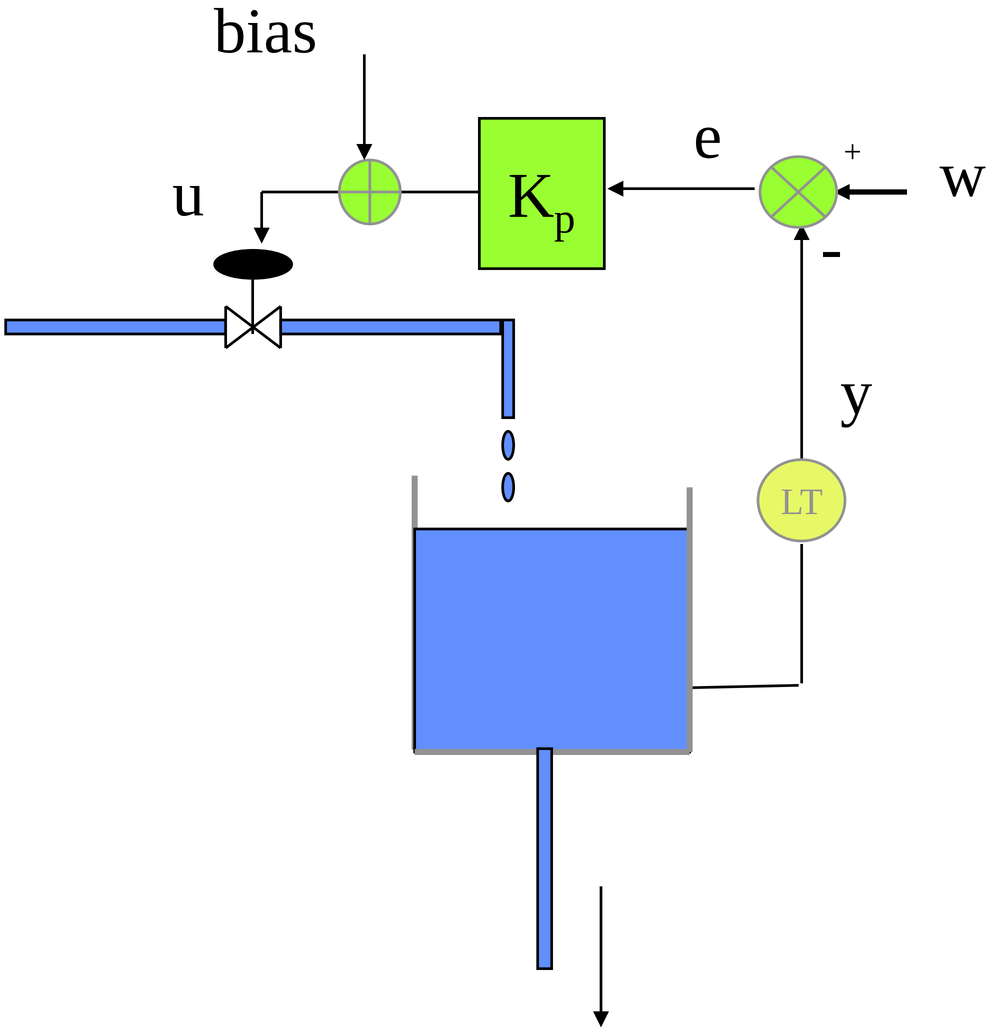
Proportional action



$u(t)=K_p\ e(t) + 30$

There is only an equilibrium point with zero error

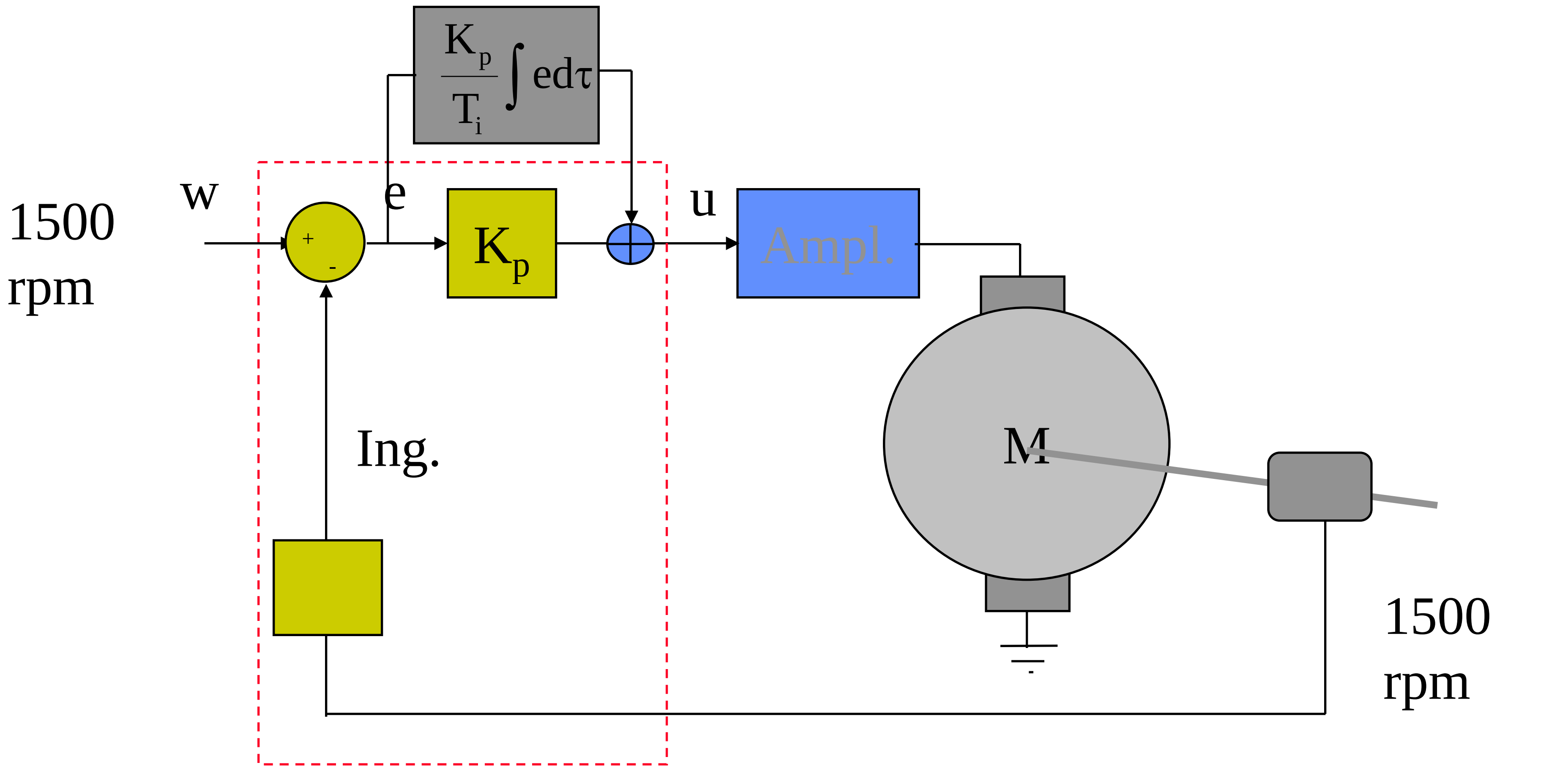
Proportional action



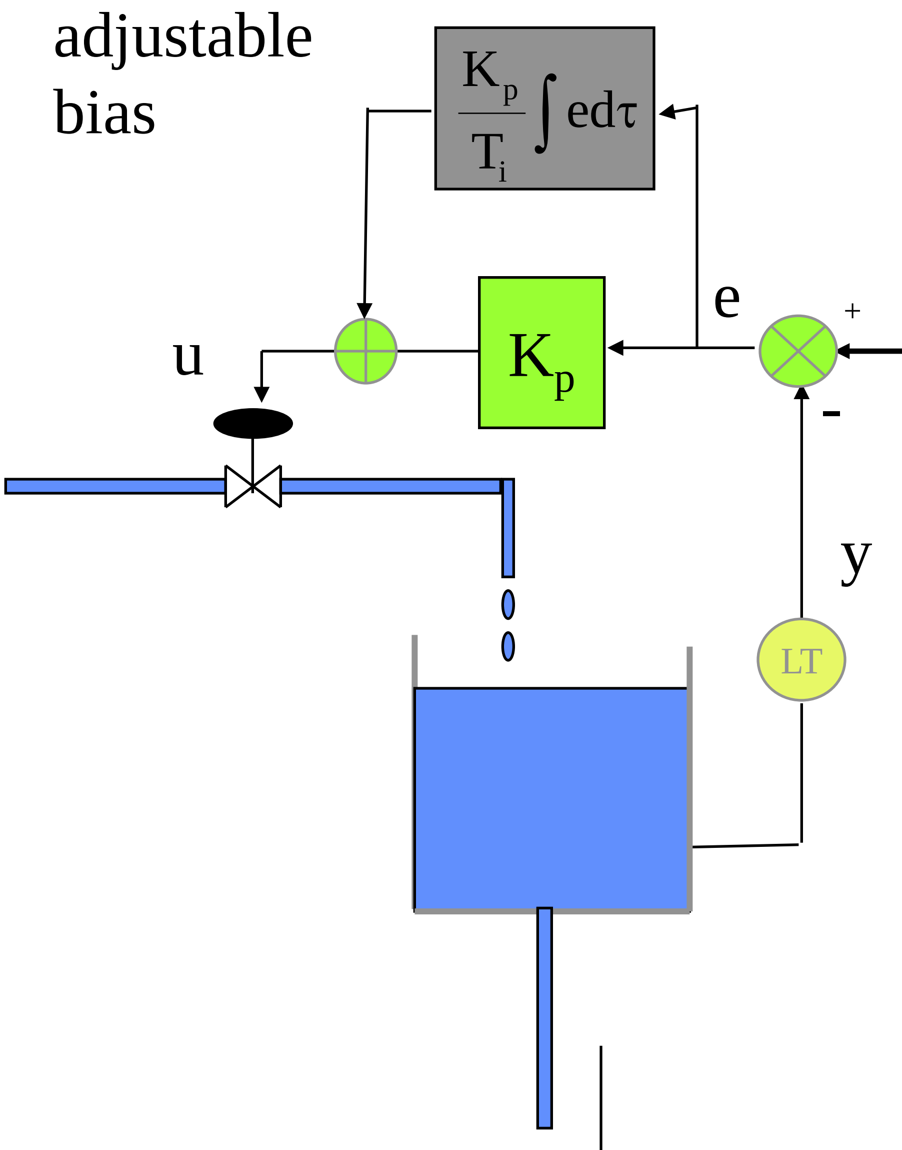
$$e(t) = w - y$$

$$u(t) = K_p\, e(t) + bias$$

Integral action



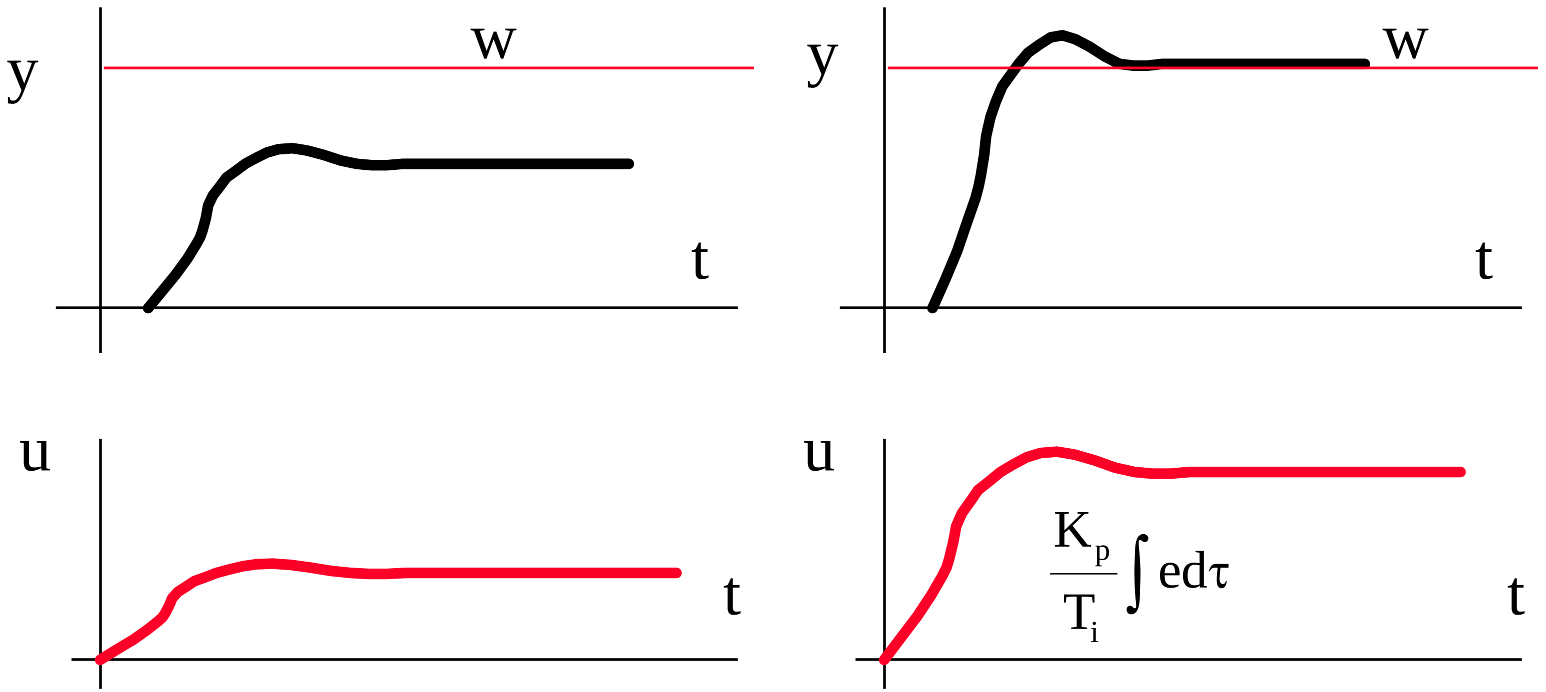
PI



$$e(t) = w - y$$

$$u(t)=K_p\, e(t) + bias$$

Integral action (automatic reset)

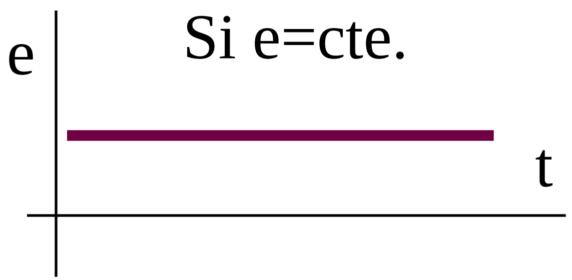


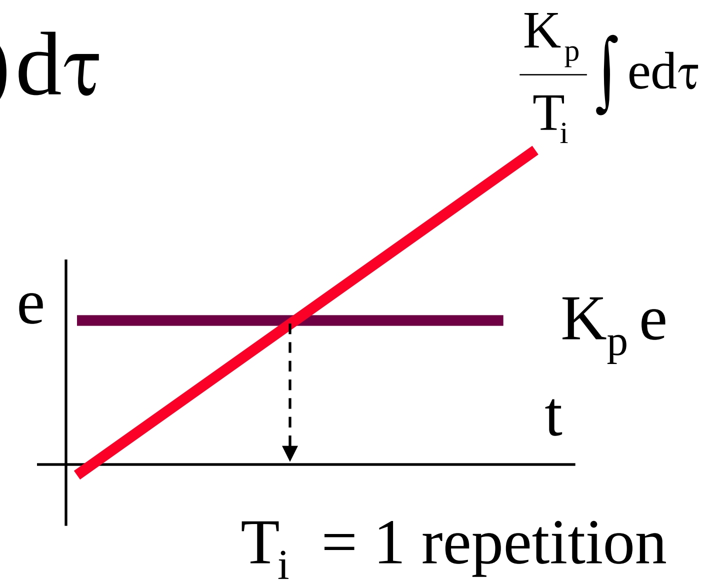
A P controller does not get steady zero error with self-regulated processes

The integral term changes continuously the control signal until the error is zero

Integral action

$$u(t) = \frac{K_p}{T_i}\int_0^t e(\tau)d\tau$$

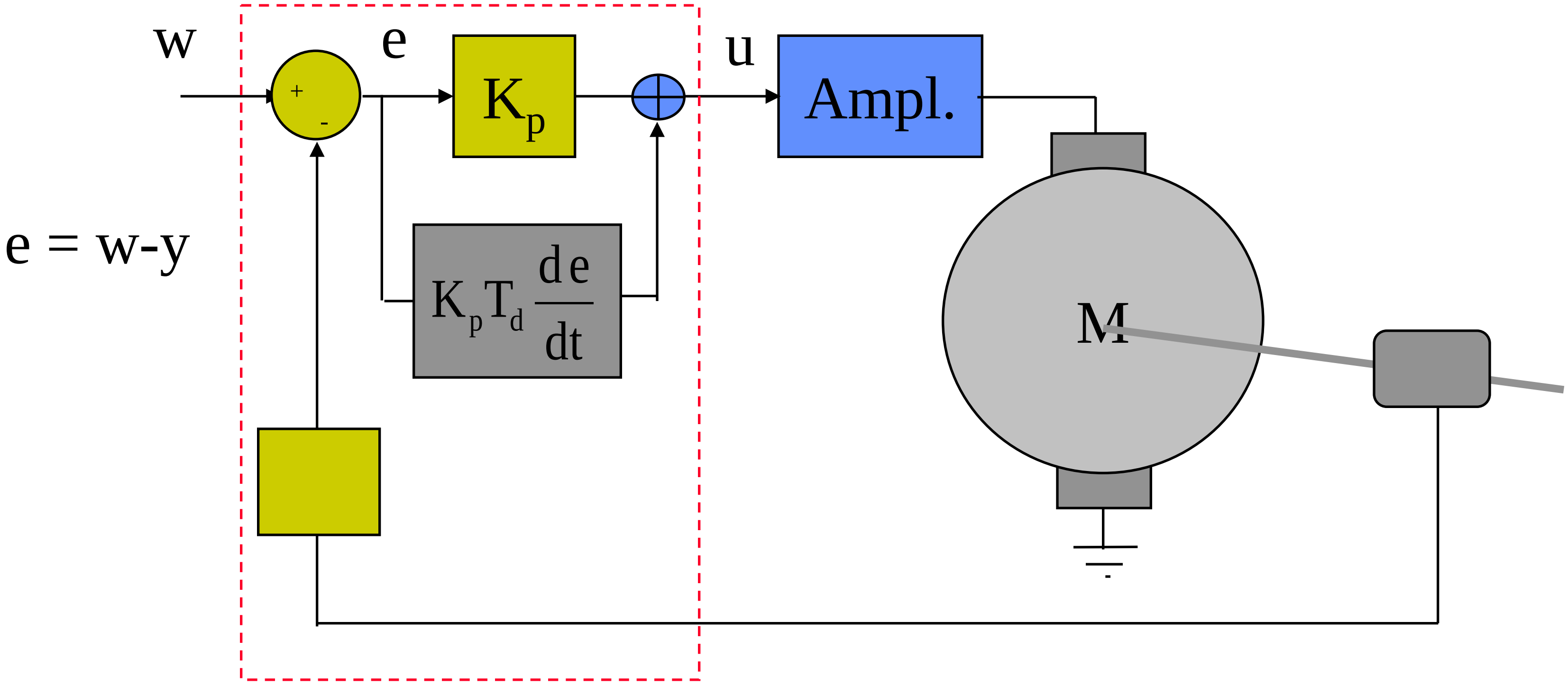




The integral action will equate the proportional one in T_i time units if e is constant (one repetition)

$$\frac{K_p}{T_i}\int e d\tau = \frac{K_p}{T_i} e t = K_p e \Rightarrow t = T_i$$

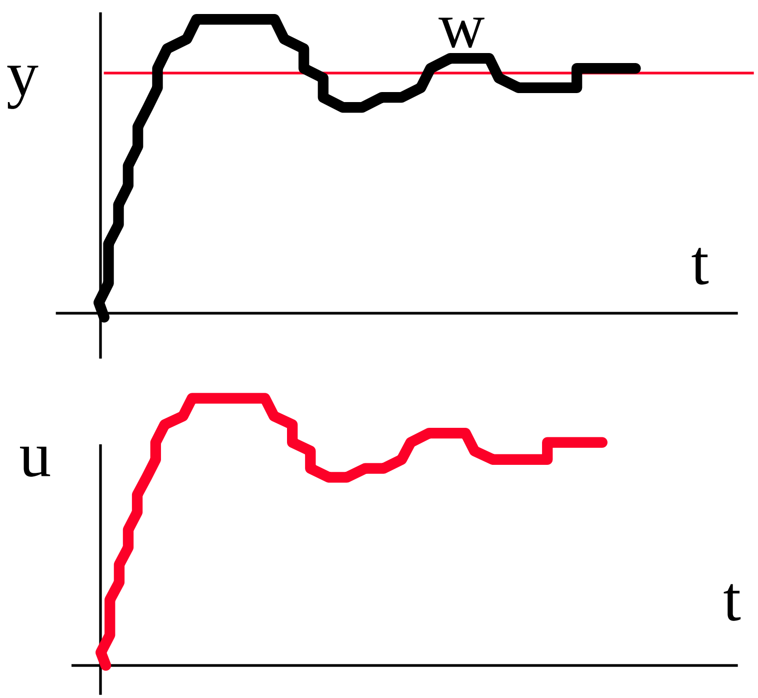
Derivative action



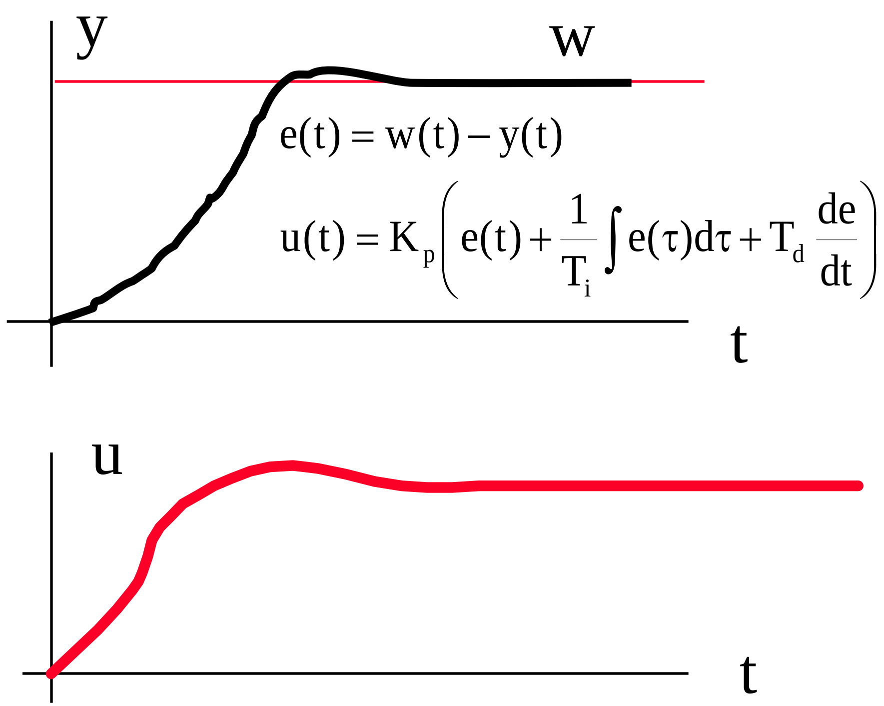
The derivative term will smooth sharp changes in the control signal due to fast changes in the error

$e = w - y$

Derivative action



A P controller tuned with high gain in order to get a fast process response can generate too strong u changes and oscillations



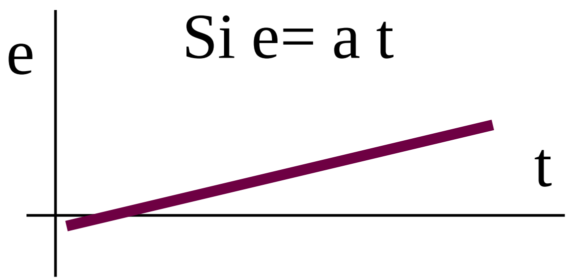
$$e(t) = w(t) - y(t)$$

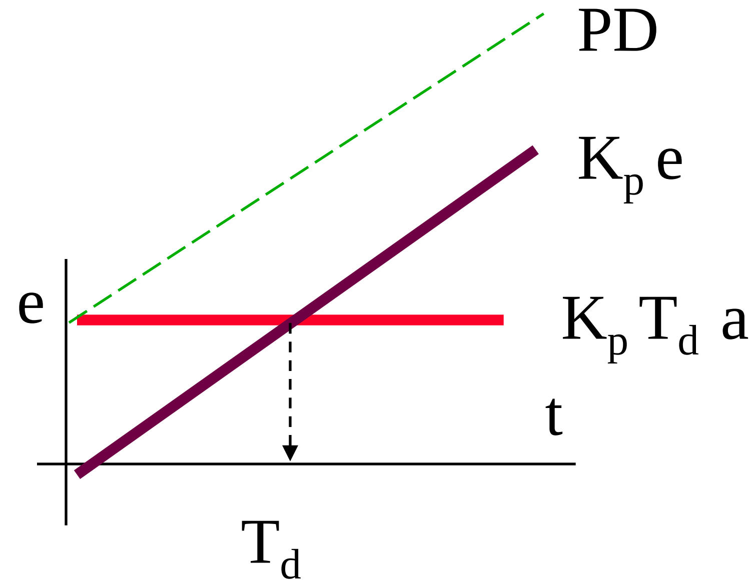
$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(\tau) d\tau + T_d \frac{de}{dt} \right)$$

If e decreases very fast, the derivative term will decrease u, avoiding oscillations

Derivative action

$$u(t) = K_p T_d \frac{de}{dt}$$



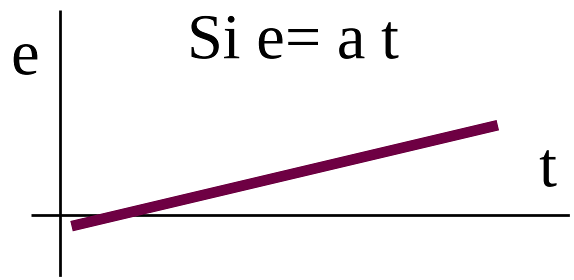


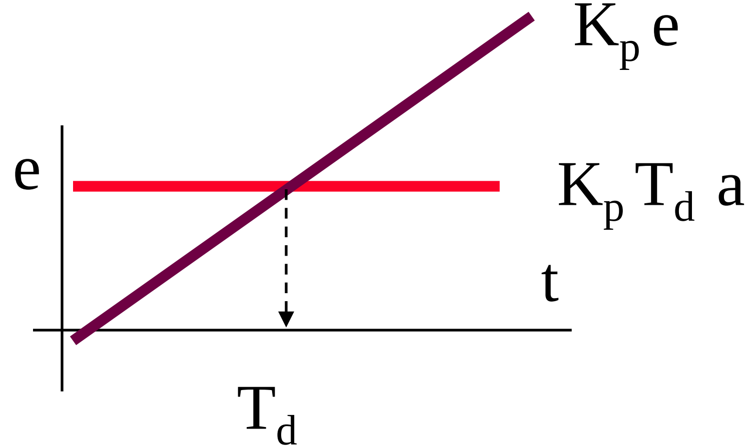
If e changes linearly, the derivative term will equate the proportional one after T_d time units

The derivative action has no influence in the steady state

Derivative action

$$u(t) = K_p T_d \frac{de}{dt}$$

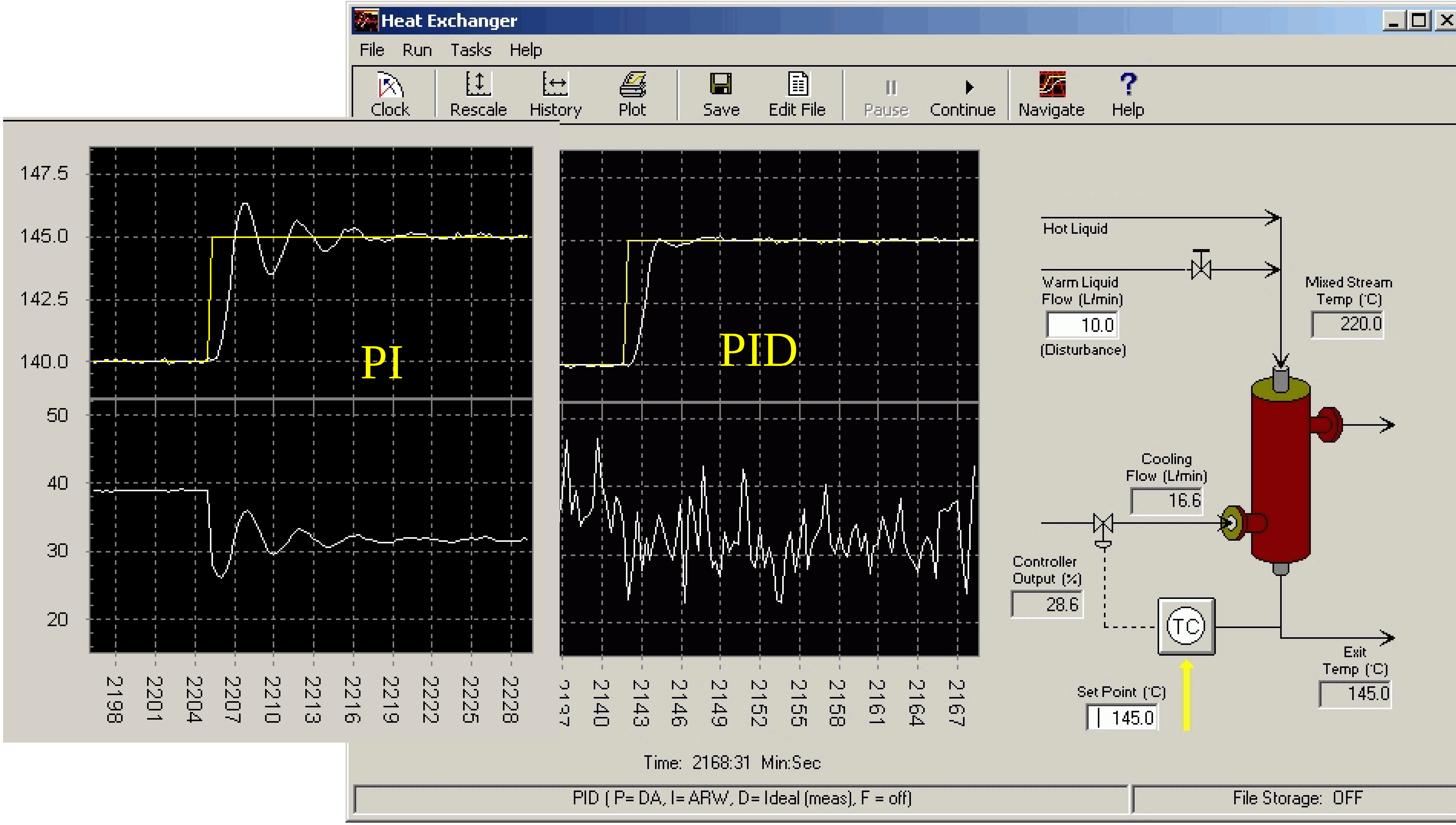




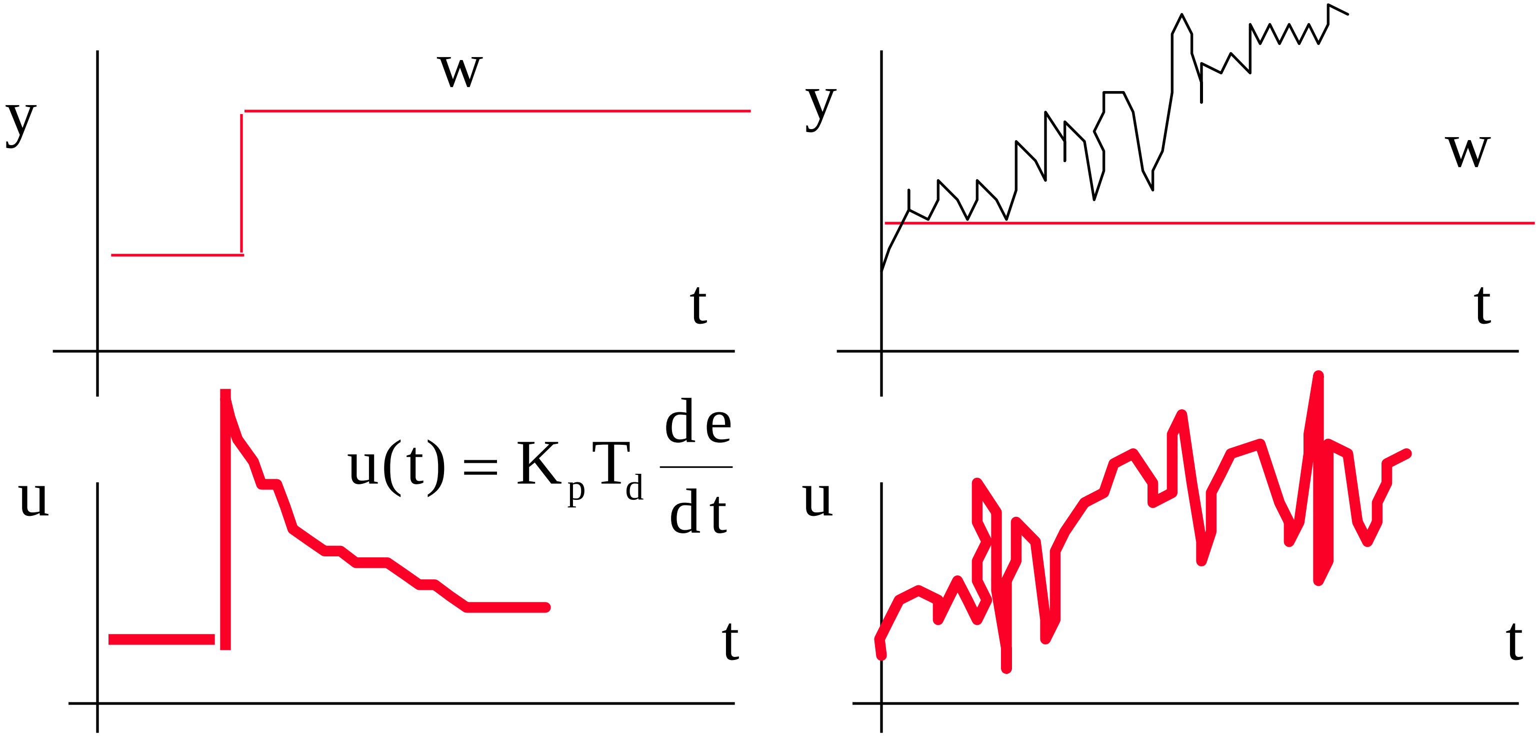
After T_d time units, derivative and proportional terms will be equal if e= a.t.

$$K_p T_d \frac{de}{dt} = K_p T_d a = K_p a t \Rightarrow t = T_d$$

Derivative action



Derivative action



$$u(t) = K_p T_d \frac{de}{dt}$$

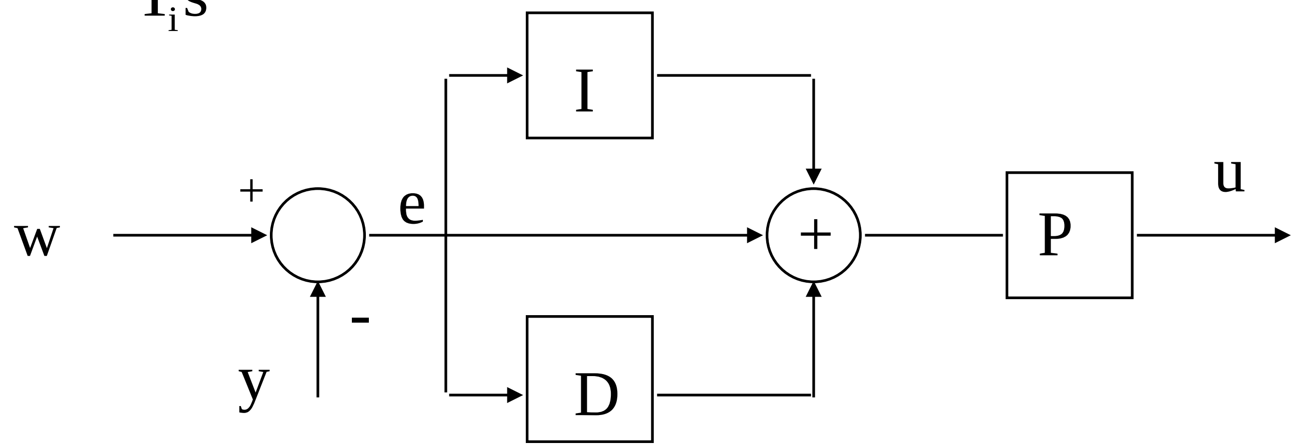
Sharp changes in w cause big changes in u at the time of change

Noisy process signals lead to fast changing control actions u

Ideal PID (non interactive)

$$u(t) = K_p (e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}) \qquad e(t) = w(t) - y(t)$$

$$U(s) = K_p \frac{T_i s + 1 + T_i T_d s^2}{T_i s} E(s)$$



It is not physically implementable

Very sensitive to noises

Real zeros for $T_i > 4T_d$

Real PID (non interactive)

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de_f}{dt}\right]$$

$$\frac{T_d}{N}\frac{de_f}{dt} + e_f = e(t) \text{ filter in the error } \quad E_f(s) = \frac{1}{\frac{T_d}{N}s + 1}E(s)$$

$$U(s) = K_p\left[1 + \frac{1}{Ts_i} + \frac{sT_d}{1 + sT_d/N}\right]E(s)$$

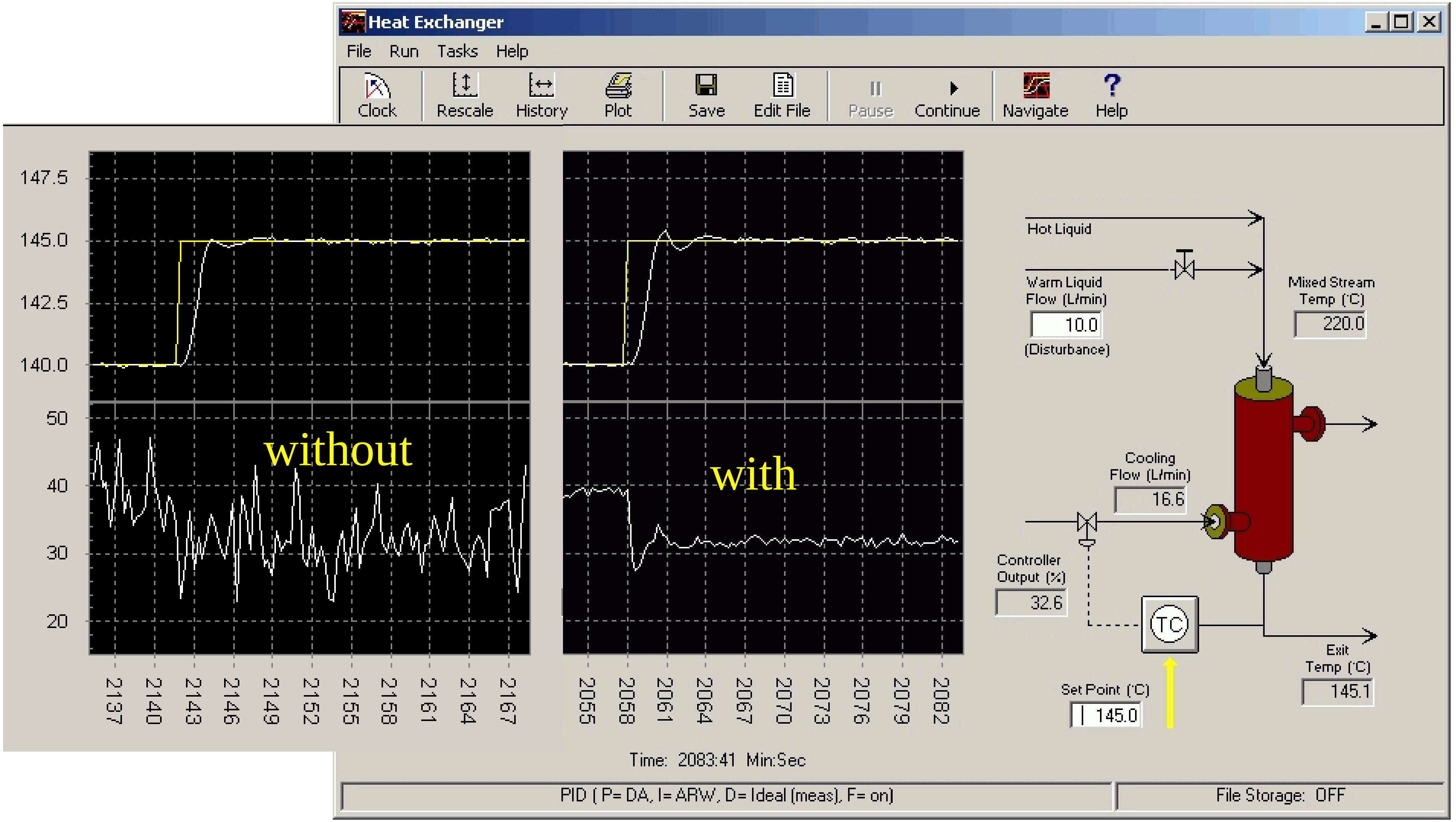
Physically implementable

Incorporates a filter in the derivative term

At high frequencies the maximum gain of the D term is $K_p N$

N : Maximum derivative gain. Typically N=10.

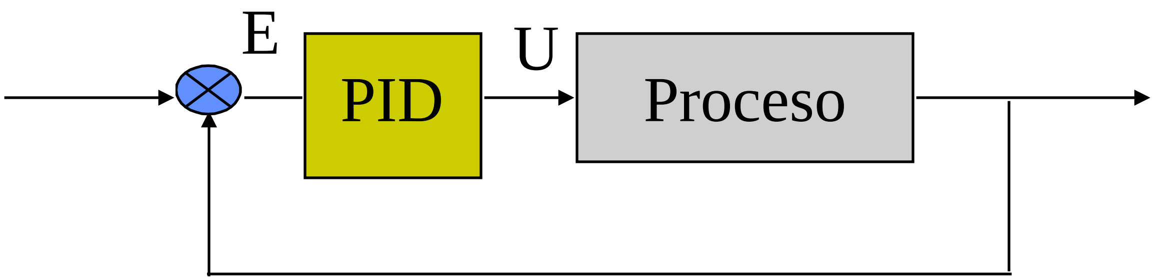
Effect of Filters



Non interactive PID

$$U(s) = \frac{K_p\left[0.1T_i T_d\ s^2 + (T_i + 1.1T_d)\,s + 1\right]}{T_i s\,(1 + 0.1T_d\ s)} E(s)$$

Position algorithm



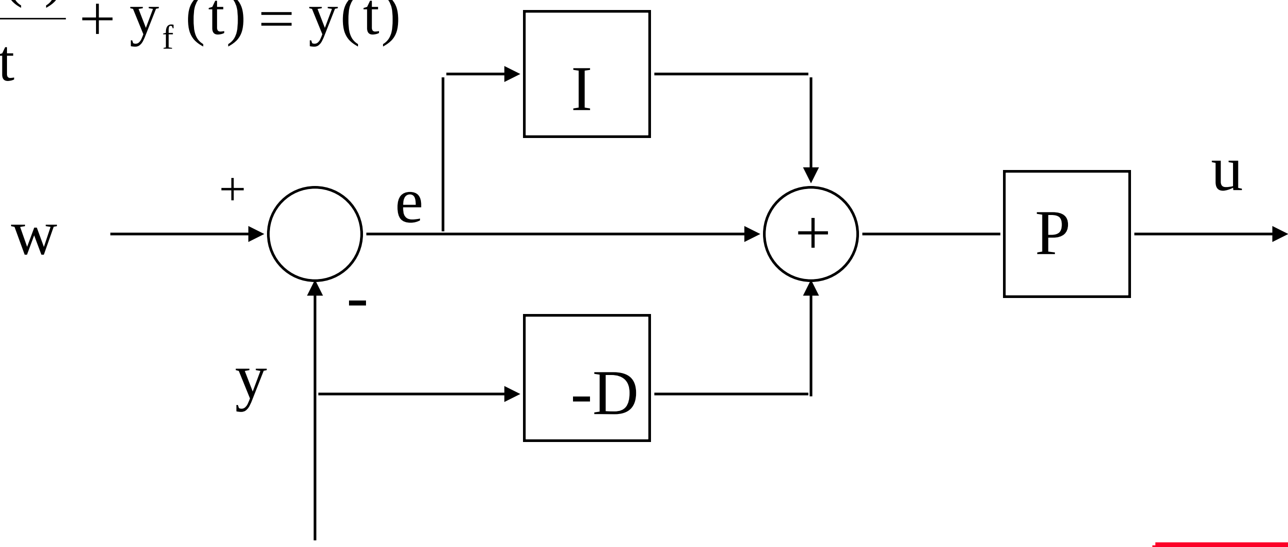
Speed algorithm: formulated in terms of the changes of u
Fits very well with incremental actuators such as
step motors, pulse driven actuators,…

PID (derivative action on y)

$$u(t) = K_p \left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau - T_d \frac{dy_f(t)}{dt}\right)$$

Honeywell type B

$$0.1T_d \frac{dy_f(t)}{dt} + y_f(t) = y(t)$$



Used in the DCS

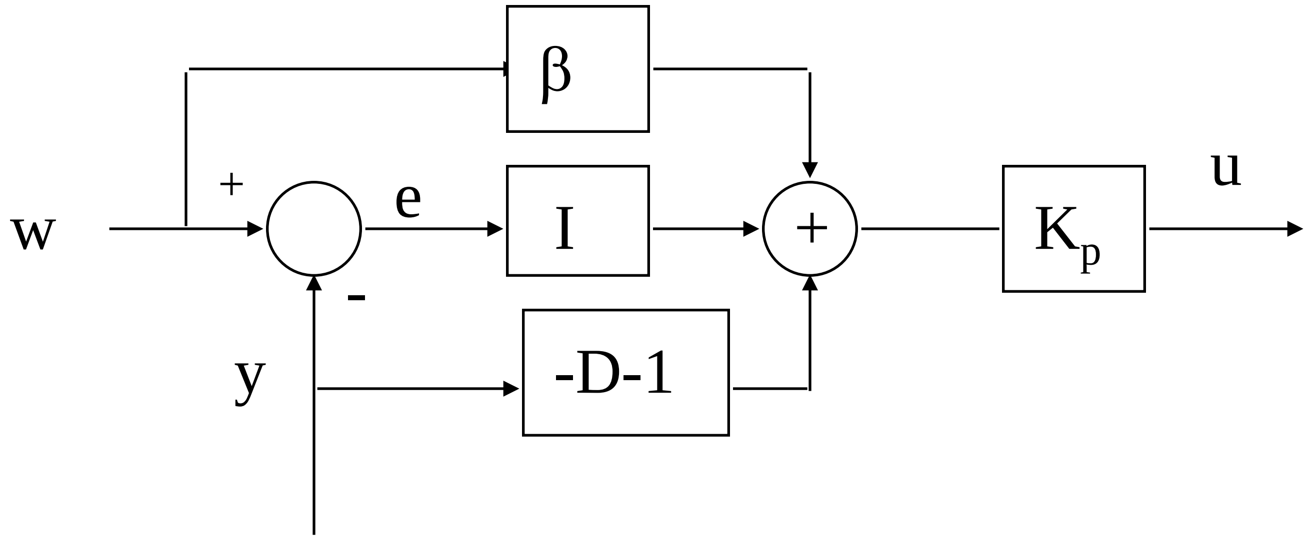
It avoids sharp changes in u when a step change is given to w

e = w -y

PID modified proportional action

$$u(t) = K_p \left[(\beta w(t) - y(t)) + \frac{1}{T_i} \int_0^t e(\tau) d\tau - T_d \frac{d y_f}{d t} \right]$$

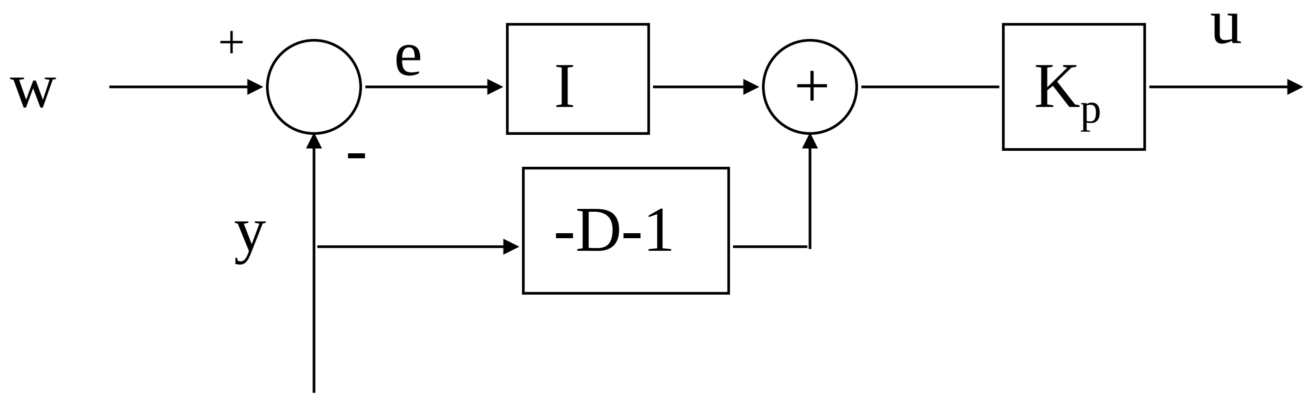
The factor β allows to have a certain independence when tuning
The controller against load or set point changes



PID modified proportional action

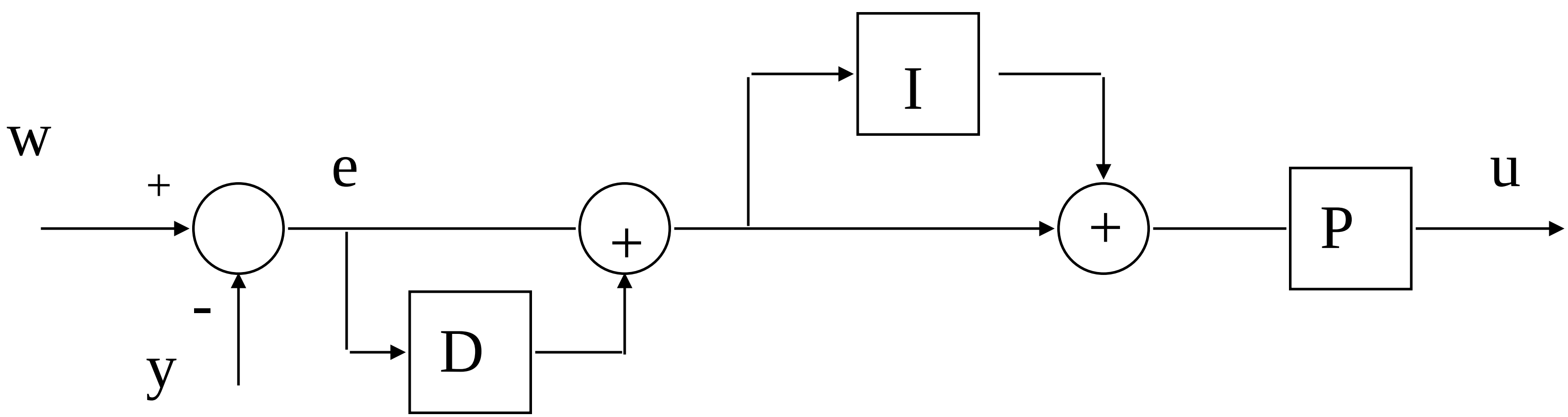
with $\beta = 0$ $$u(t) = K_p\left[(-y(t)) + \frac{1}{T_i}\int_0^t e(\tau)d\tau - T_d\frac{dy_f}{dt}\right]$$

Honeywell type C



Series or Interactive PID

$$U(s) = K_{ps}\left(1 + \frac{1}{T_{is}s}\right)\left(\frac{1 + T_{ds}s}{1 + 0.1T_{ds}s}\right)E(s)$$



Series or Interactive PID

$$U(s) = K_{ps}(1 + \frac{1}{T_{is}s})(\frac{1 + T_{ds}s}{1 + 0.1T_{ds}s})E(s)$$

- Used in the old analog or loop controllers
- Equivalence tables between the parameters of series and parallel PID types

$F=1+T_{ds}/T_{is}$ $K_p = K_{ps} F$; $T_i = T_{is} F$; $T_d = T_{ds} / F$

$F_s = 0.5+(0.25-T_d/T_i)^{0.5}$ $K_{ps} = K_p F_s$; $T_{is} = T_i F_s$; $T_{d\,s} = T_d / F_s$

Full parallel PID

$$u(t) = K_{pp} e(t) + \frac{1}{T_{ip}} \int_0^t e(\tau) d\tau + T_{dp} \frac{de(t)}{dt}$$

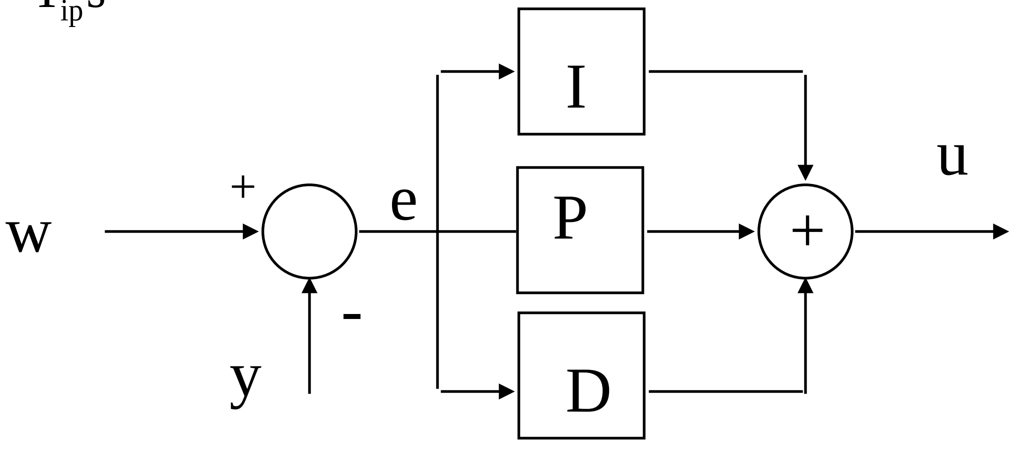
$$e(t) = w(t) - y(t)$$

$$U(s) = \frac{K_{pp} T_{ip} s + 1 + T_{ip} T_{dp} s^2}{T_{ip} s} E(s)$$

$$K_{pp} = K_p$$

$$\frac{1}{T_{ip}} = \frac{K_p}{T_i}$$

$$T_{dp} = K_p T_d$$



Non linear PID

The gain is modified, so that the action of the controller is stronger when the error is big and very smooth or zero when the error is small or there are noises, etc

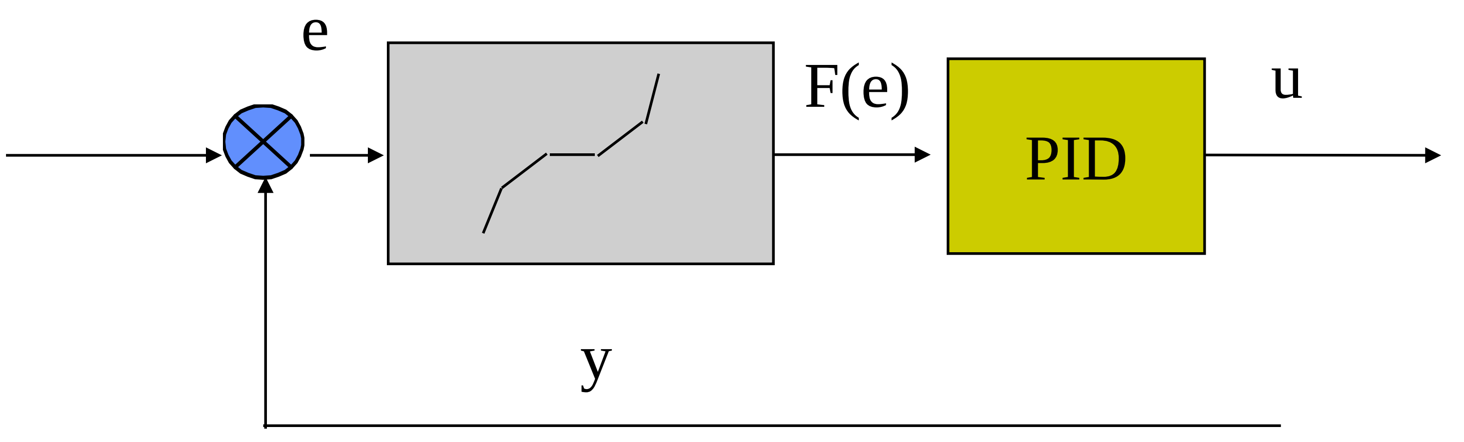
$$u(t) = K_p f(e)\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau - T_d \frac{dy_f}{dt} \right]$$

$f(e)$ function of the error, e.g.:

$f(e) = \alpha + (1-\alpha)e$ with, for instance, $\alpha = 0.1$

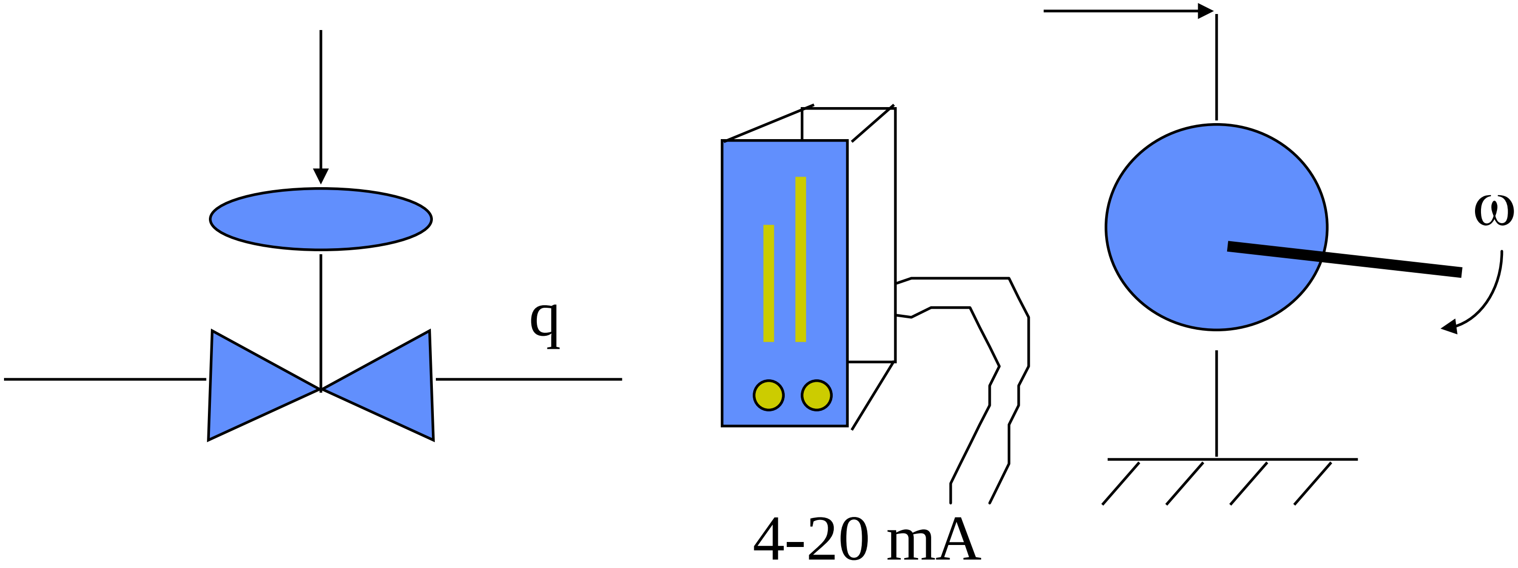
Non linear PID

f(e) Non linear function of the error
Dead zone around e=0
High gain for big |e|



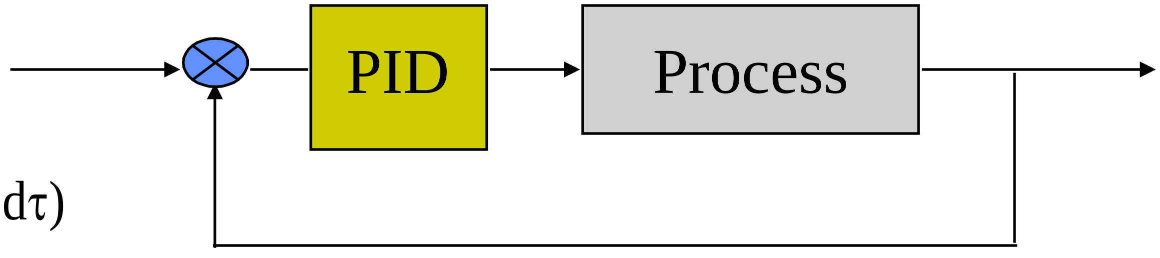
There are no changes in u when e is small, (e.g. noises)
Increases the control actions if e is big

Saturation in the instruments



All actuators and transmitters have a limited range of operation, with its signals been constrained to it (0 - 100 %)

Reset wind-up

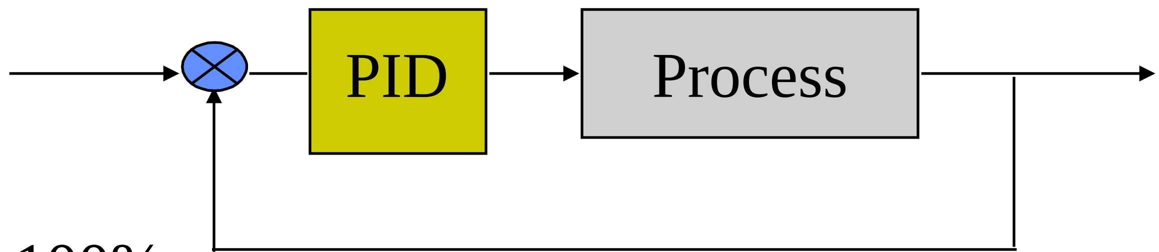


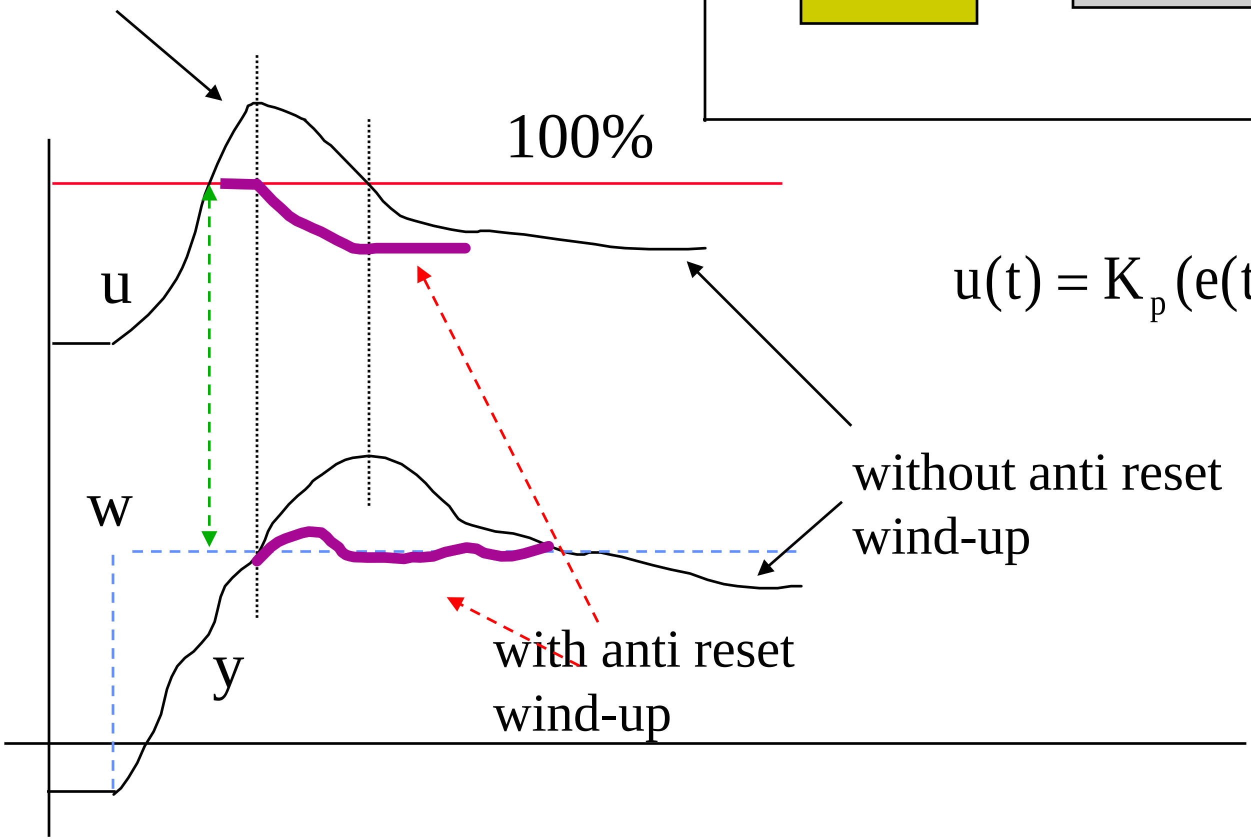
$$u(t) = K_p \left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)$$

Delay in the actuation of the controller output that appears when the value of the integral term exceed the allowable range of the manipulated variable.

The implementation of the so called anti wind-up systems, avoid the appearance of this phenomenon.

Reset wind-up



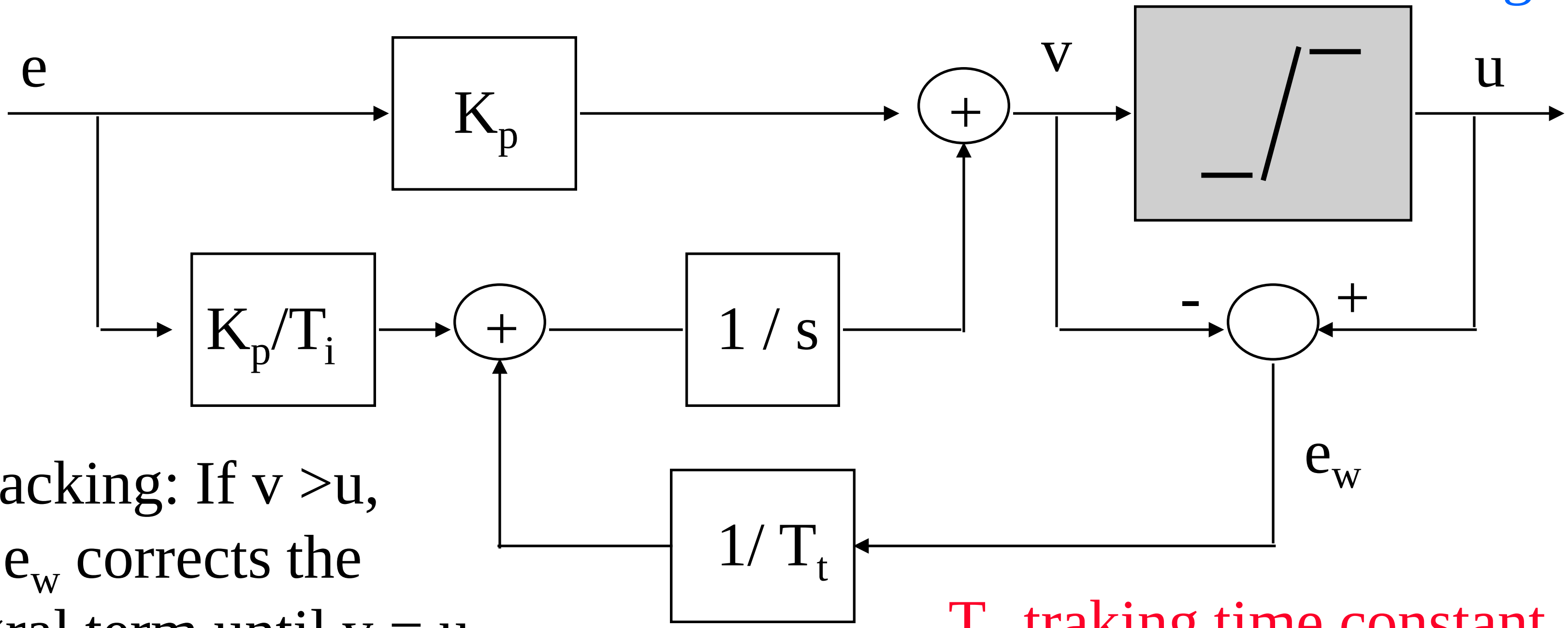


$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)$$

Anti-reset wind up

1. $$u(t) = K_p(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau)$$

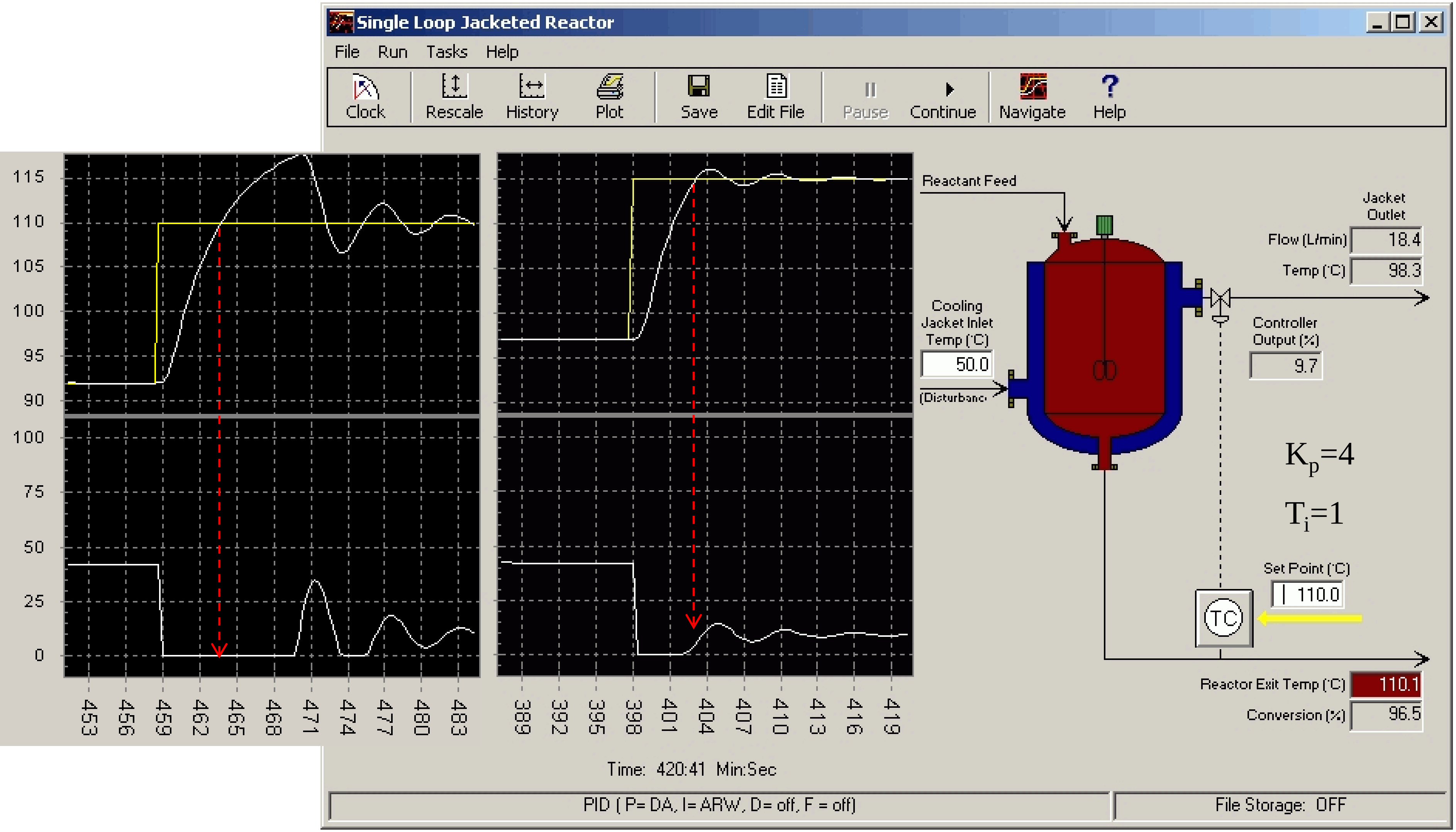
Key action: Stop the integration if the integral term exceeds the output range



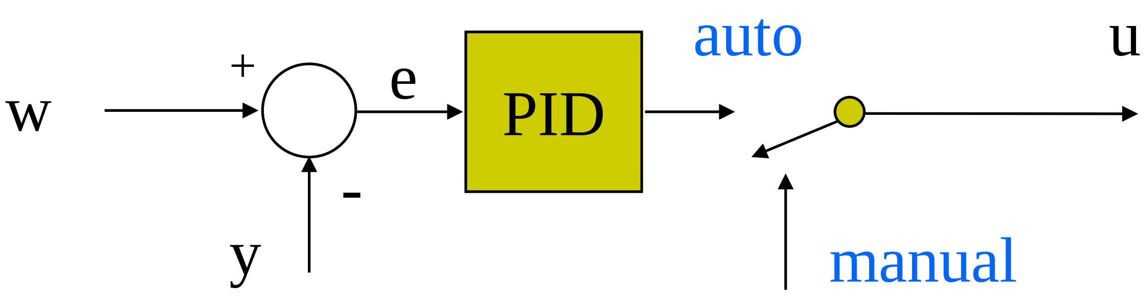
2. Tracking: If v >u, then e_w corrects the integral term until v = u

T_t traking time constant

Anti-reset windup



auto/man transfers



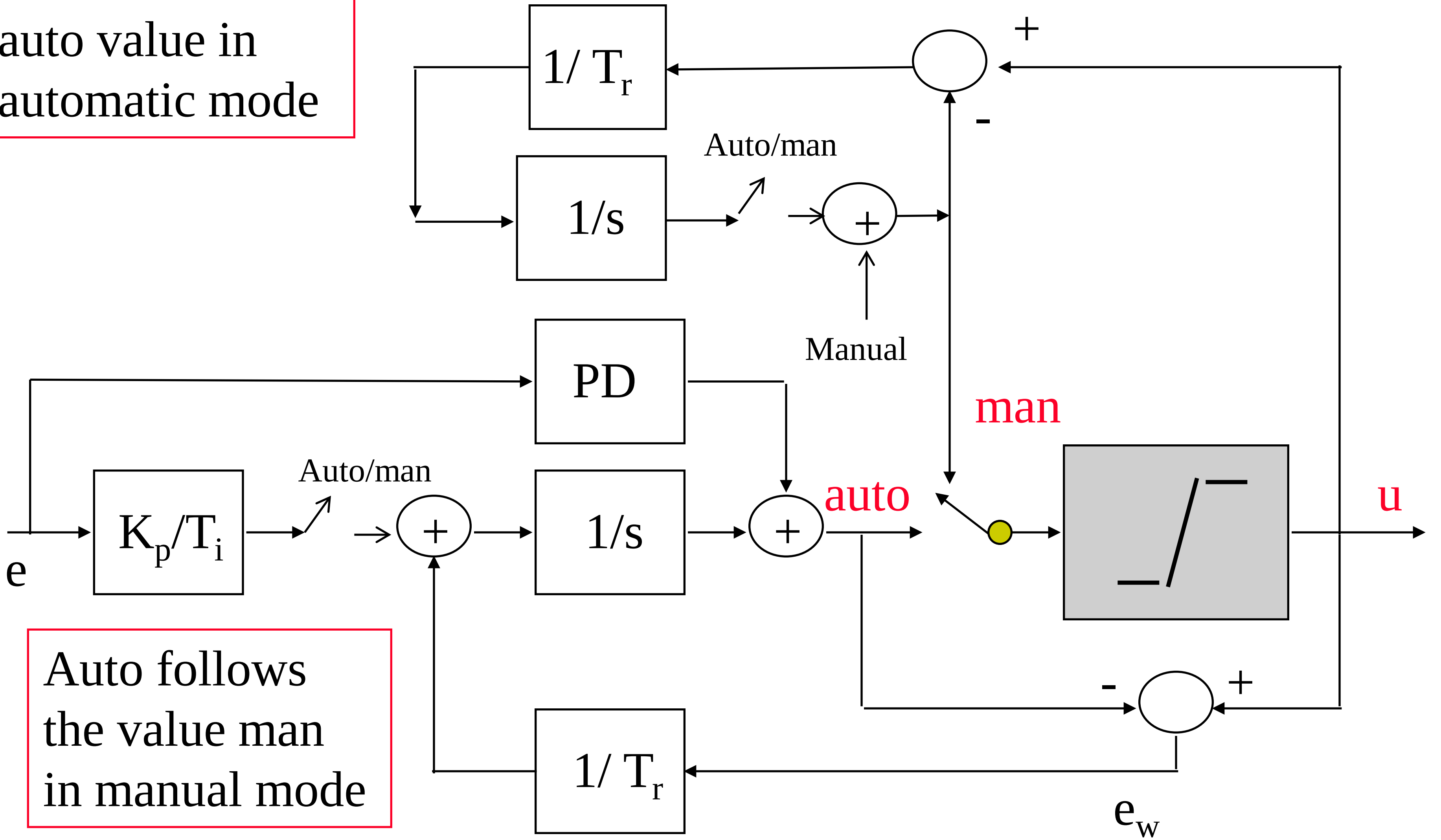
In a auto/man mode transfer u can suffer from strong changes

The controller should operate with smooth auto/man and man/auto transfers (bumpless)

Changing the value of a parameter should be made without strong output changes

Bumpless transfers

Man follows the auto value in automatic mode

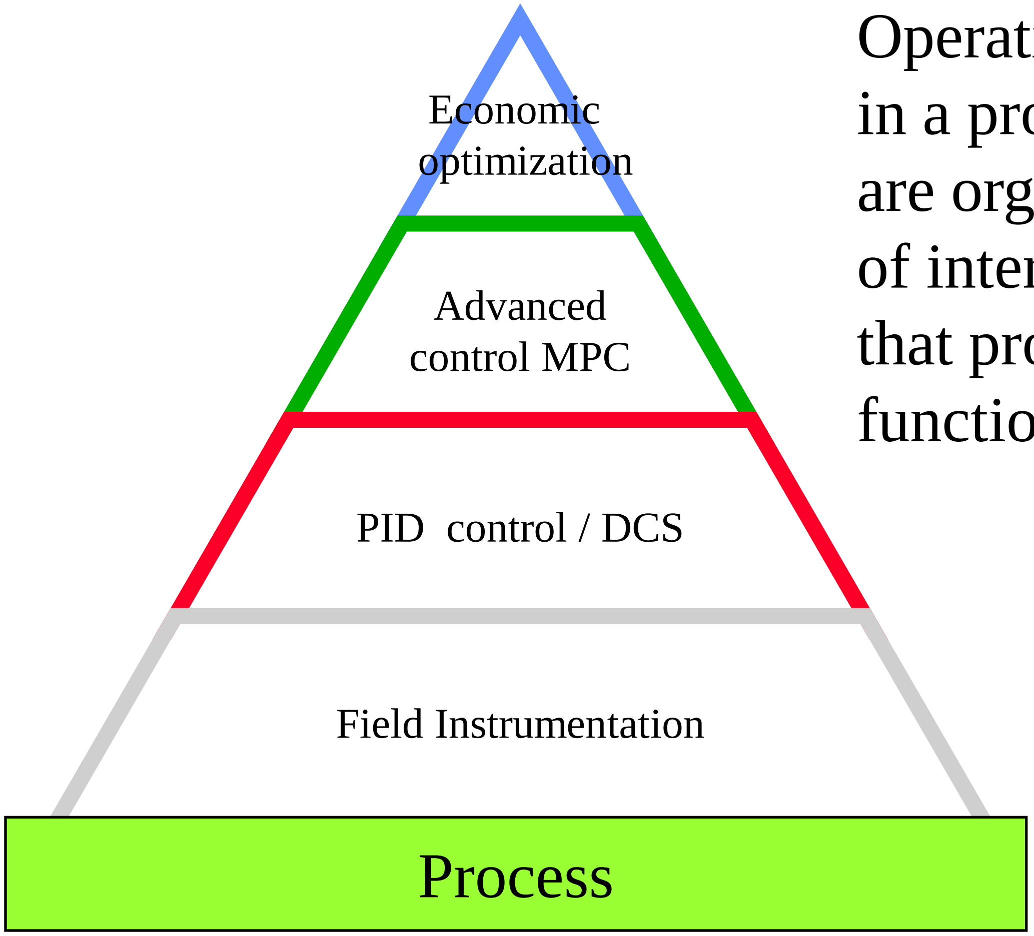


Auto follows the value man in manual mode

PID tuning

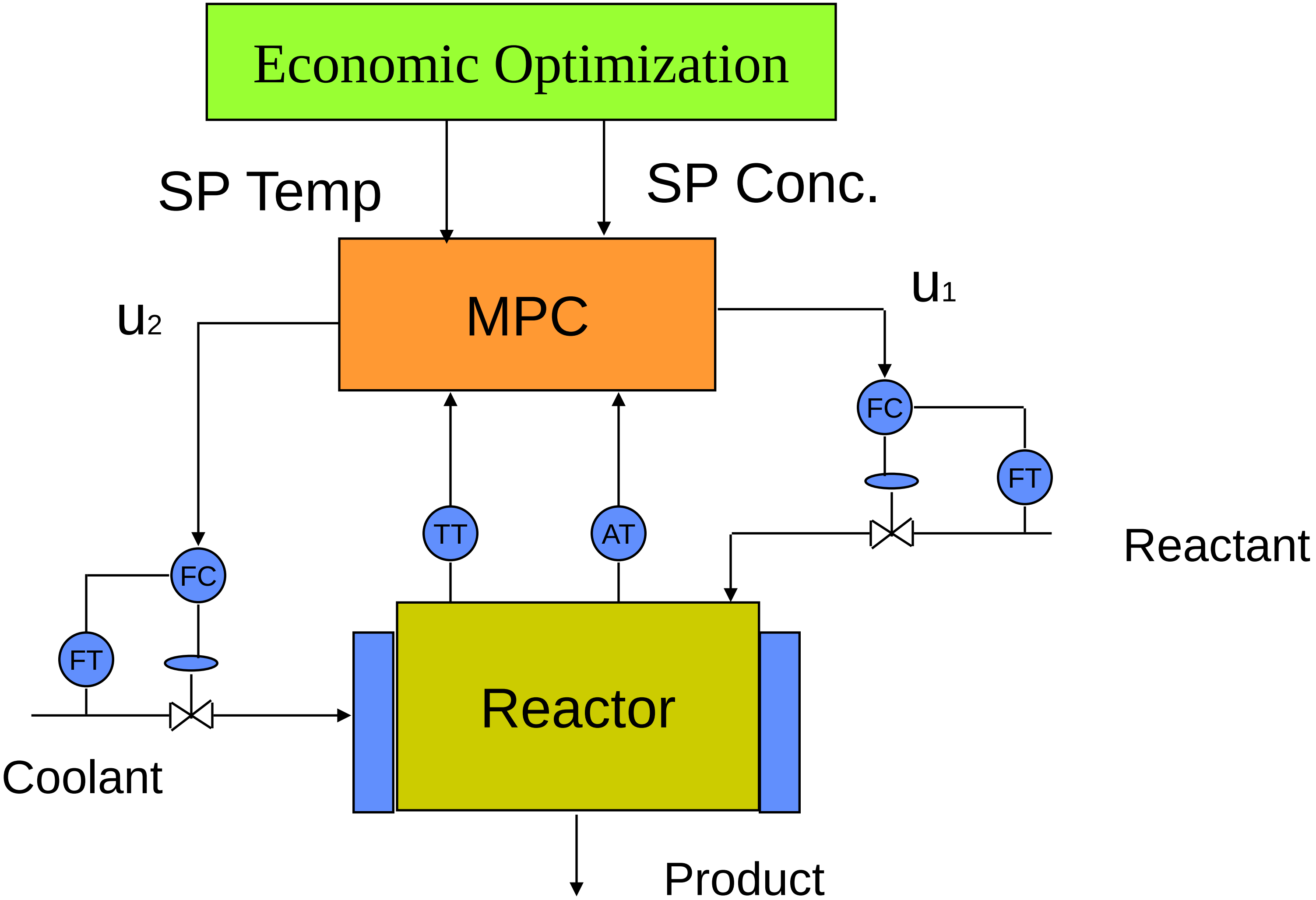
- Selection of the PID parameters in order to obtain an adequate closed loop behaviour
- K_p, T_i, T_d
- Other parameters: N,T_r, β, T,$_{,}$ constraints, ...
- Several methods + process knowledge
- Very important for an adequate operation of the factory

Control Pyramid

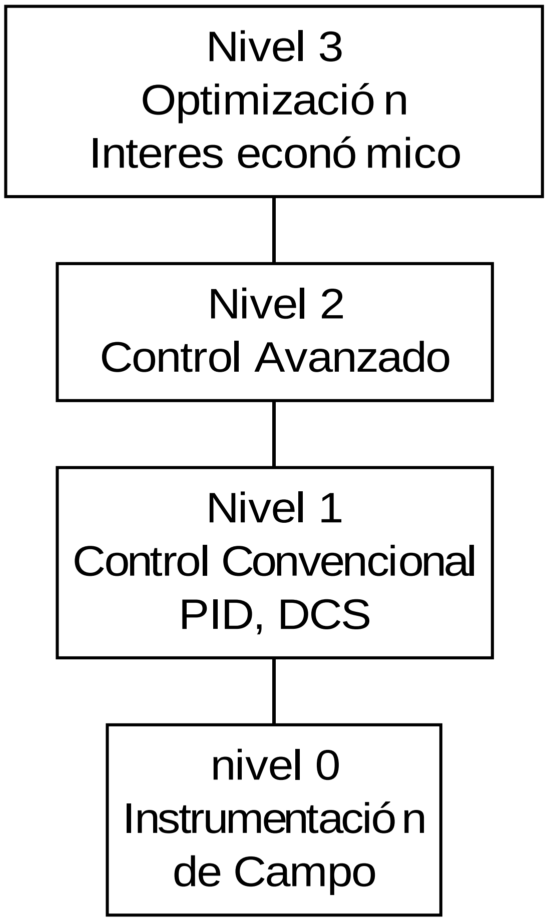


Operation and control in a process factory are organized in a set of interacting layers that provide different functionalities

Control Pyramid



Control Hierarchy



In order to implement solutions at one level, the lower ones must operate properly

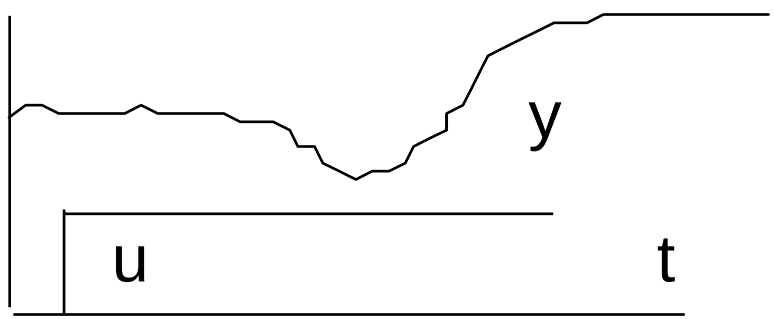
PID tuning is also important because implementing advanced control requires the correct functioning of the conventional PID controllers

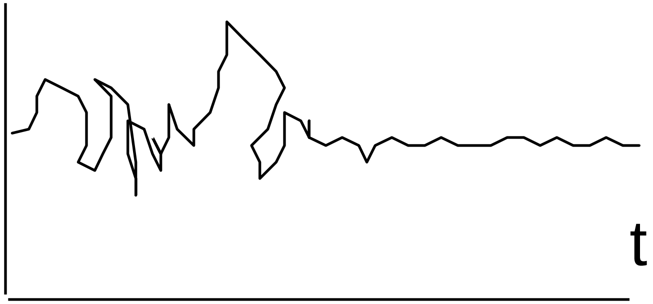
Control aims

- Safety: What can happen if the loop fails, or other associated variables fail?
- Impact: What other things are affected by this loop?
- Performance: What type of response can be achieved? How the loop will be affected by disturbances?
- Economy: How the functioning of the loop affects the economy of the process?
- Endurance: Which are the chances to fail?
- Price: How expensive is the instrumentation involved?

When using PID control?

- PID controllers work well with most of the single input single output (SISO) control problems (flow, pressure, speed, ...)
- Nevertheless, the PID may not be a good option when dealing with difficult dynamics or very demanding specifications:
  - » Significant delay   unstable systems
  - » Non minimum phase   minimum output variance





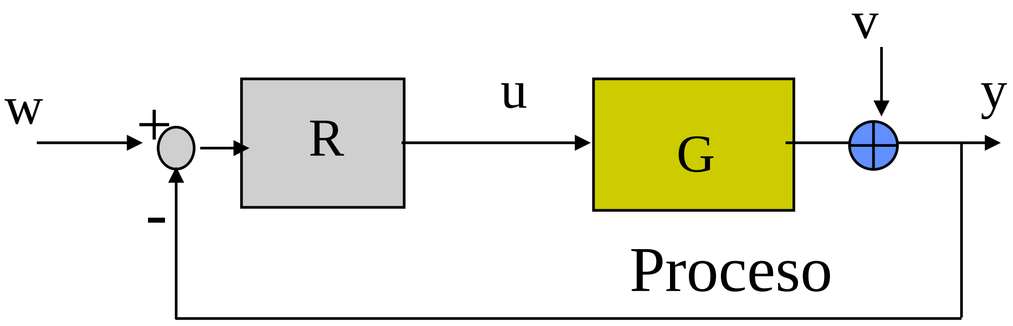
Tuning criteria

- ✓ Select the type of controller P, PI, PID, PD, type B, C.. or other controller (DMC, IMC,...)
- ✓ Tuning respect to set point or disturbance changes (w or v)
- ✓ Different control aims
- ✓ Do not forget the manipulated variable
- ✓ Robustness against changes in the process or the operating point

Controller types

- PID is the right choice in slow processes without a significant noise, such as temperature, concentration and, in some cases pressure.
- PI is the preferred choice most of the times
- P is used in processes with an integrator o where a zero steady state error is not important (e.g. internal loops in cascades).
- If the process have a significant delay use a Smith Predictor. Use MPC in multivariable, constraint or economic important process units.

Tuning: SP or disturbances?



$$y = \frac{GR}{1+GR} w + \frac{1}{1+GR} v$$

If the PID is tuned to obtain a good response against disturbances, then R is fixed and the dynamical response with respect to SP is also fixed. And viceversa.

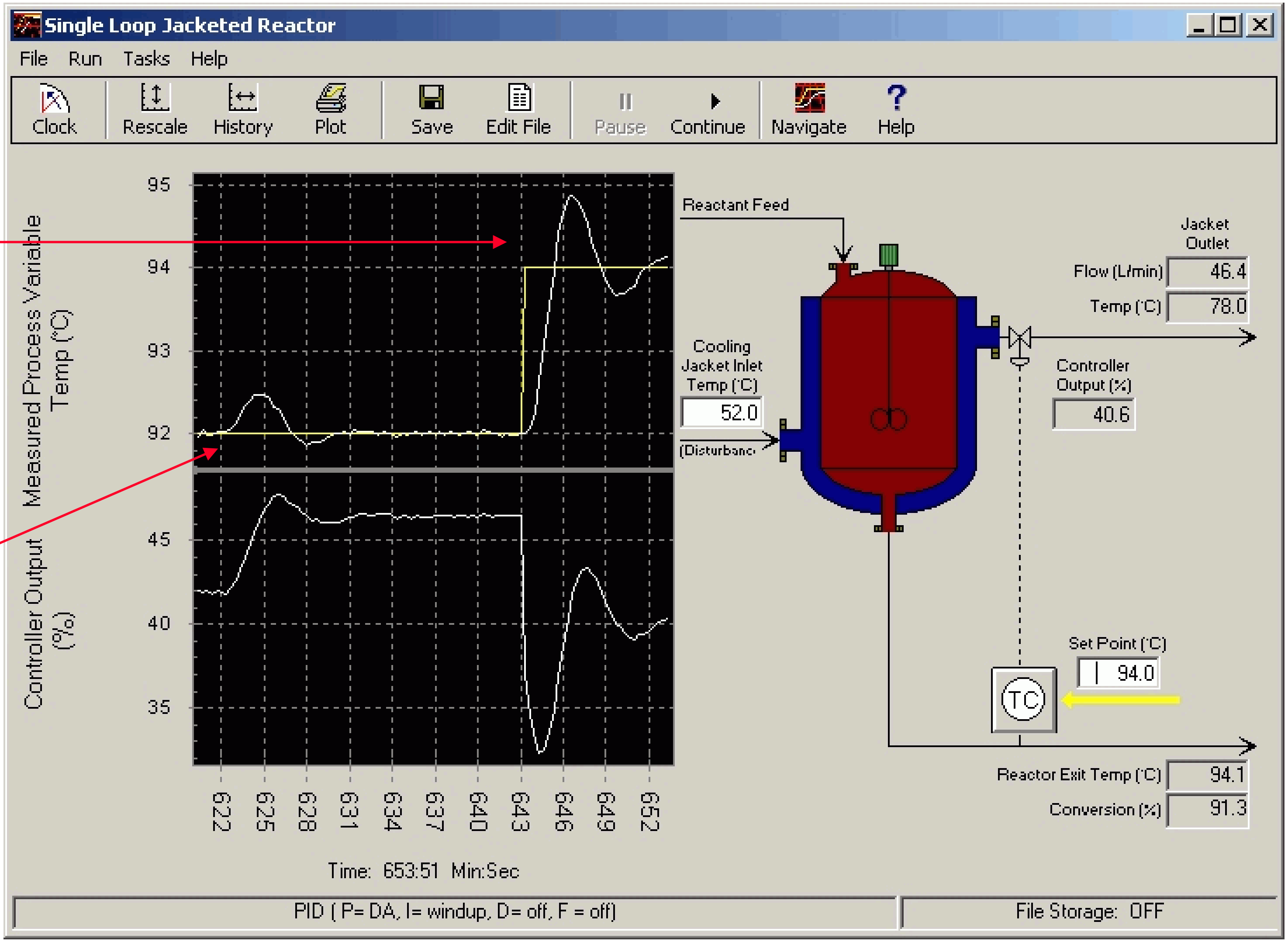
PID: a single degree of freedom

Disturbance / SP

Change in SP

Disturbance

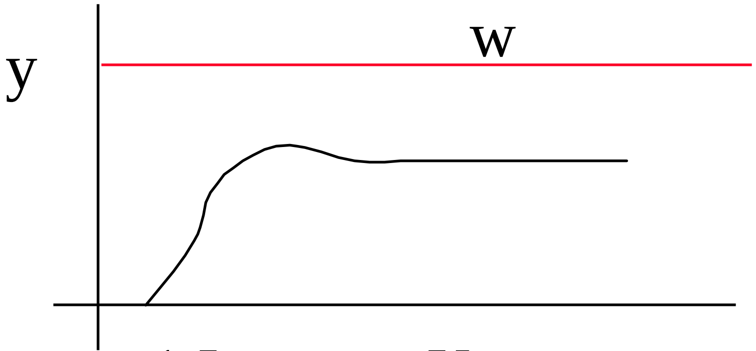
50 → 52



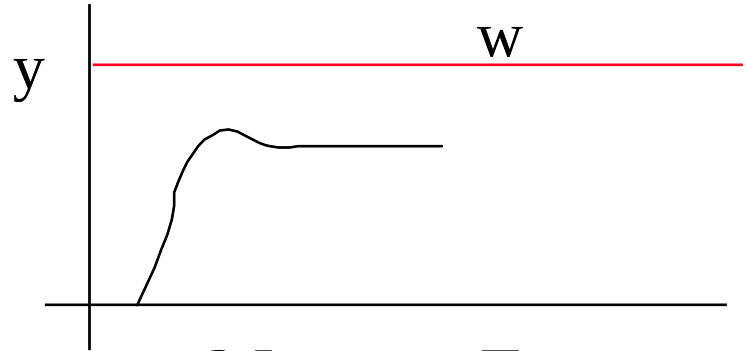
PID Tuning methods

- Trial and error methods
- Experiment based methods
  - Perform an experiment in order to estimate certain dynamic characteristics of the process
  - Compute the tuning parameters using tables or formulas as a function of the estimated dynamical characteristics of the process
- Model based analytical methods
- Automatic tuning methods

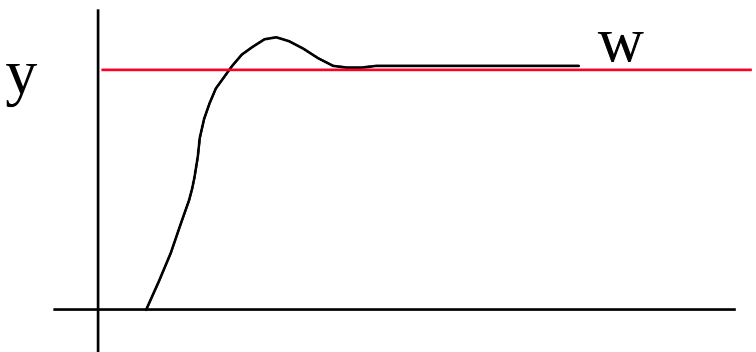
Trial and Error



1 Increase K_p



2 Increase T_d



3 Decrease T_i

Start from low K_p, and without integral or derivative actions

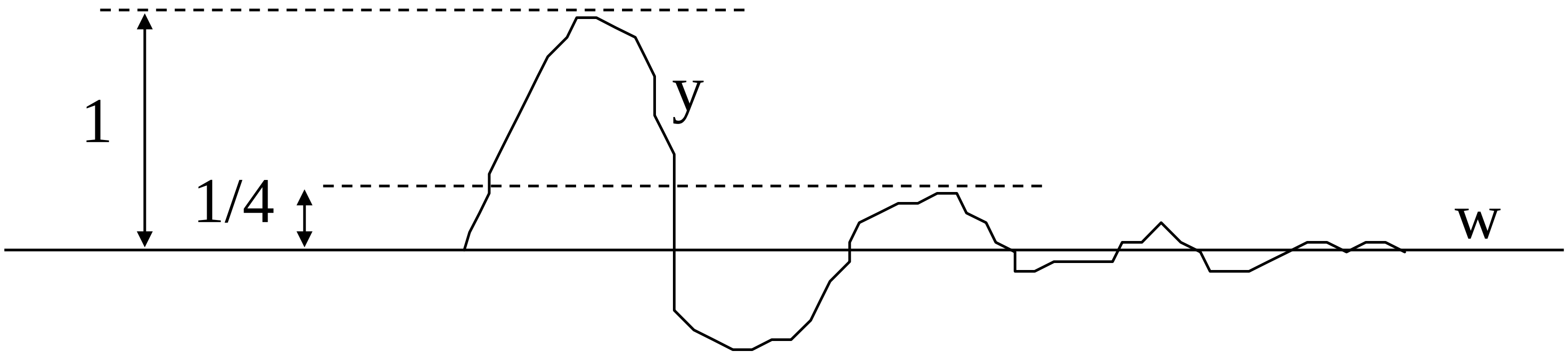
Increase K_p until a nice CV shape response is obtained without using excessive MV. Do not consider the steady state error

Increase a bit T_d and K_p in order to improve the response

Decrease T_i until the steady state error is cancelled in a sensible time

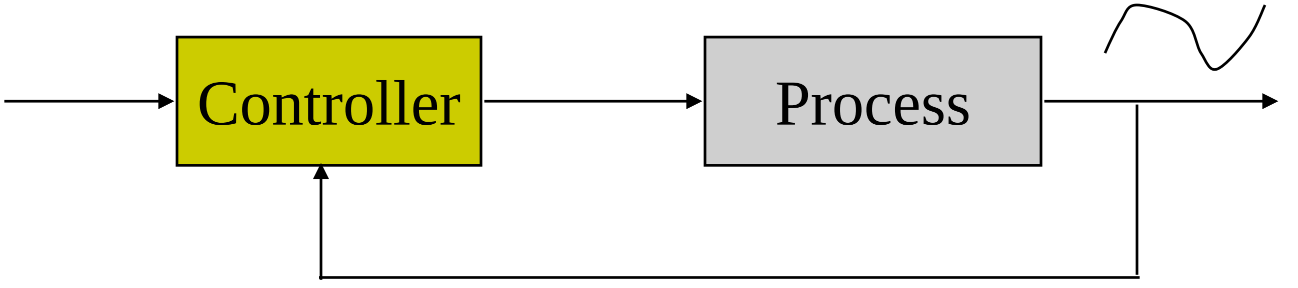
Ziegler-Nichols methods

- **Tuning criterion**: ¼ damping against disturbances (QDR)
- Empirically developed for series PID (1942)
- Two methods: Open and closed loop
- Can be applied when $0.15 < d/\tau < 0.6$ in monotonous processes
- Provide good starting values that can be fine tuned



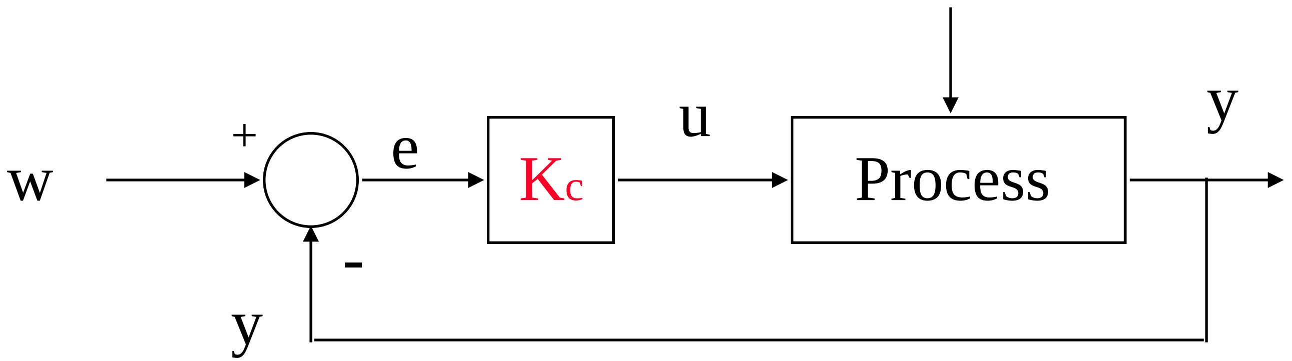
Open and closed loop methods

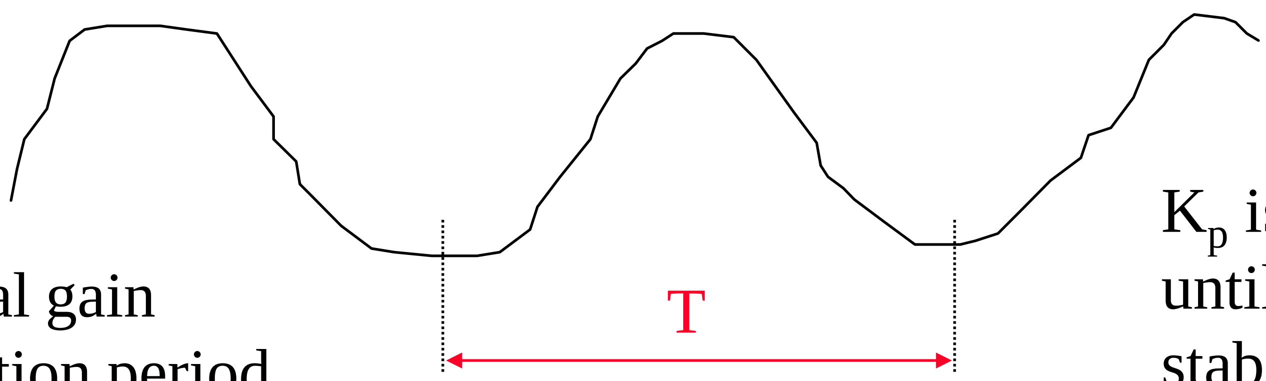
Closed loop experiment



Open loop experiment

Closed loop Ziegler-Nichols method





K_c critical gain
T oscillation period

K_p is increased until the stability limit is reached

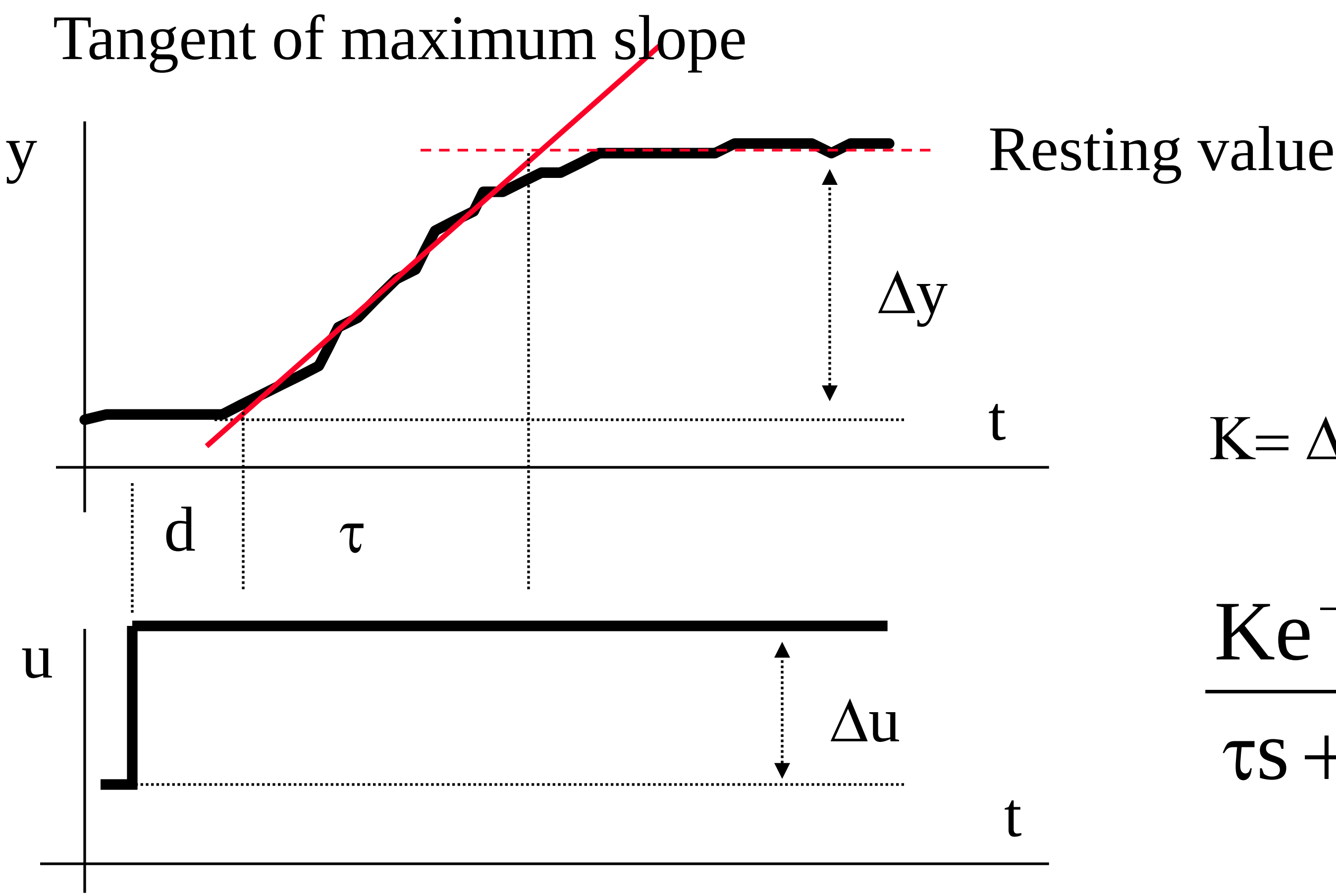
Closed loop Ziegler-Nichols tuning table

Type	**Gain K_p**	**Integral time**	**Derivative time**
P	0.5 K_c		
PI	0.45 K_c	T/1.2	
Parallel PID	0.75 K_c	T/1.6	T/10
Series PID	0.6 K_c	T/2	T/8

K_c critical gain T oscillation period

T_i and T_d in the same units as T

Step test Identification

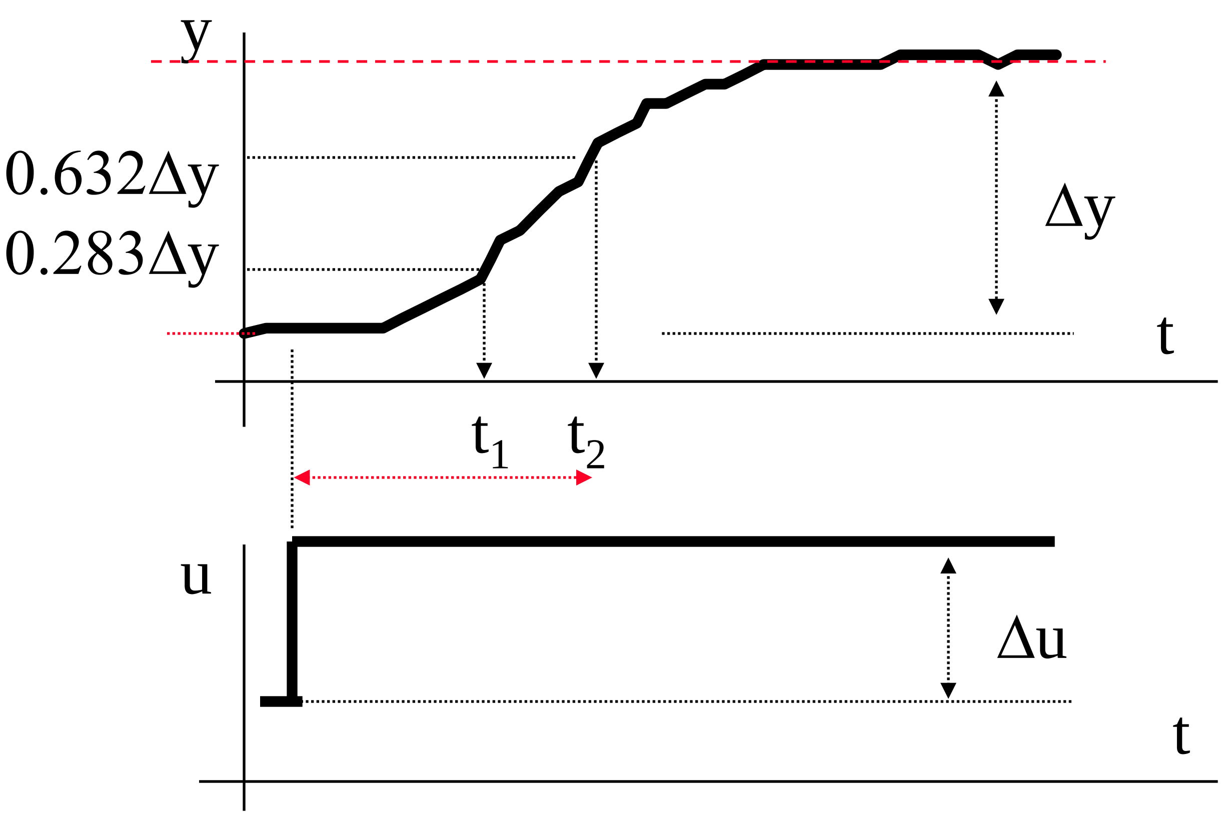


$$K = \Delta y / \Delta u$$

$$\frac{Ke^{-ds}}{\tau s + 1}$$

Adequate for Ziegler-Nichols

Step test Identification



$\tau = 1.5\,(t_2 - t_1)$

$d = t_2 - \tau$

$K = \Delta y / \Delta u$

$$\frac{Ke^{-ds}}{\tau s + 1}$$

Adequate for noisy systems

Open loop Ziegler-Nichols tuning table

Type	**Gain K_p**	**Integral time**	**Derivative time**
P	τ / (K d)		
PI	0.9τ /(K d)	3.33 d	
Series PID	1.2τ /(K d)	2 d	0.5 d

K process gain , d delay , τ time constant

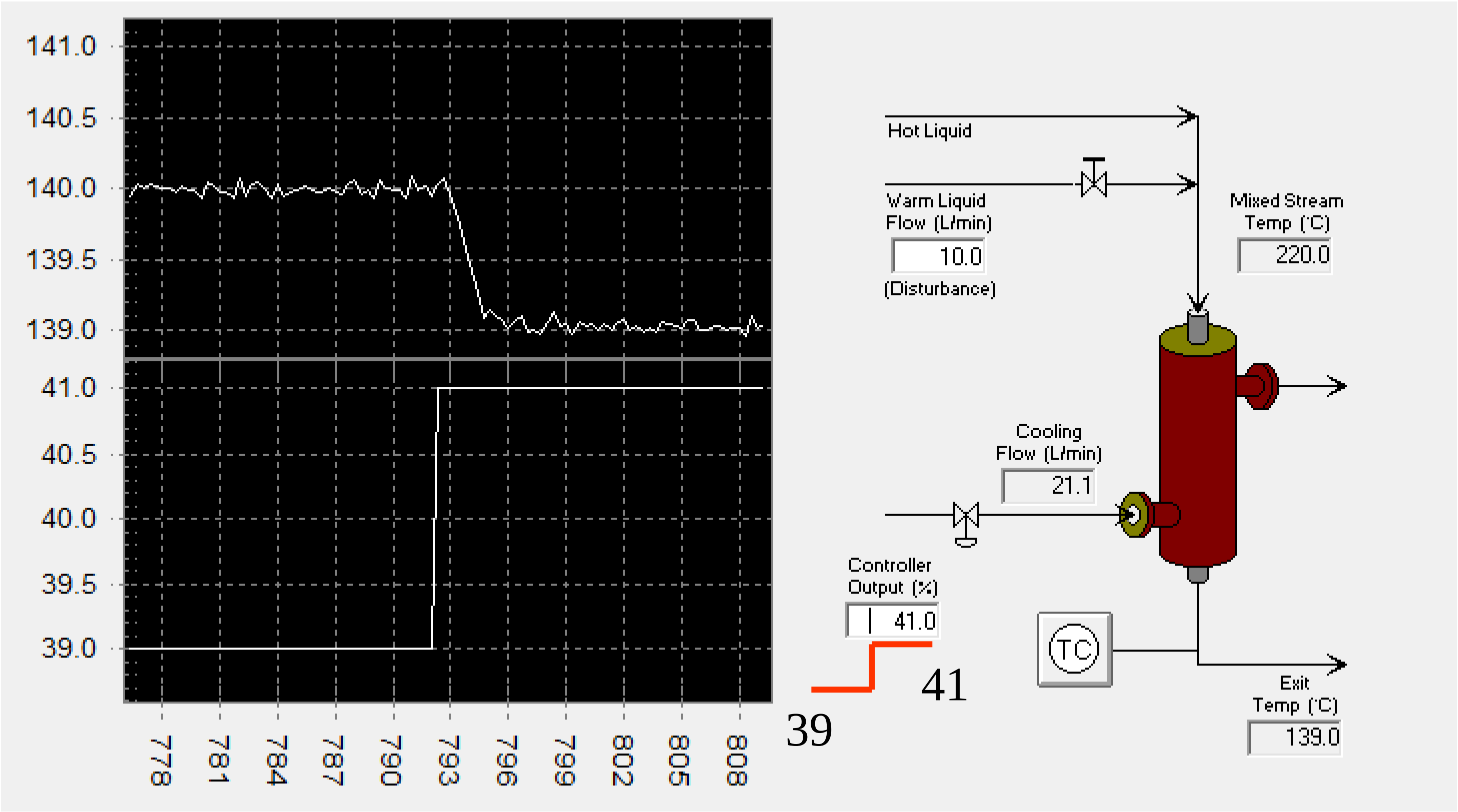
T_i and T_d in the same units as d

Notice that $T_i = 4 T_d$

When applied to digital controllers, increase d by half a sampling period

Heat exchanger

Open loop step test

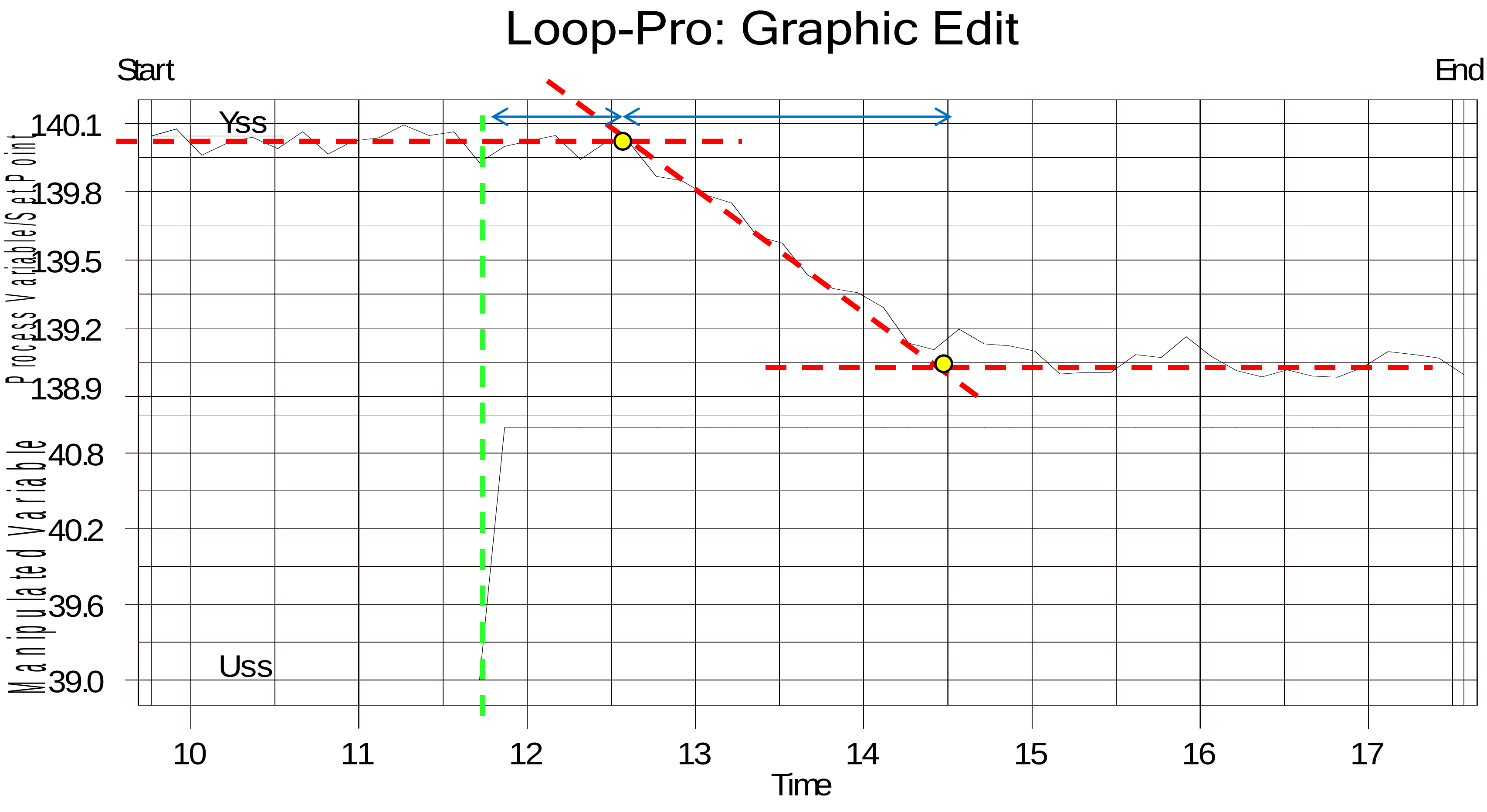


K = (139.05 -140) /2 = -0.475

d = 0.85   τ = 1.9

$$G(s) = \frac{-0.475e^{-0.85s}}{1.9s+1}$$

Difficulty of obtaining good models due to noise



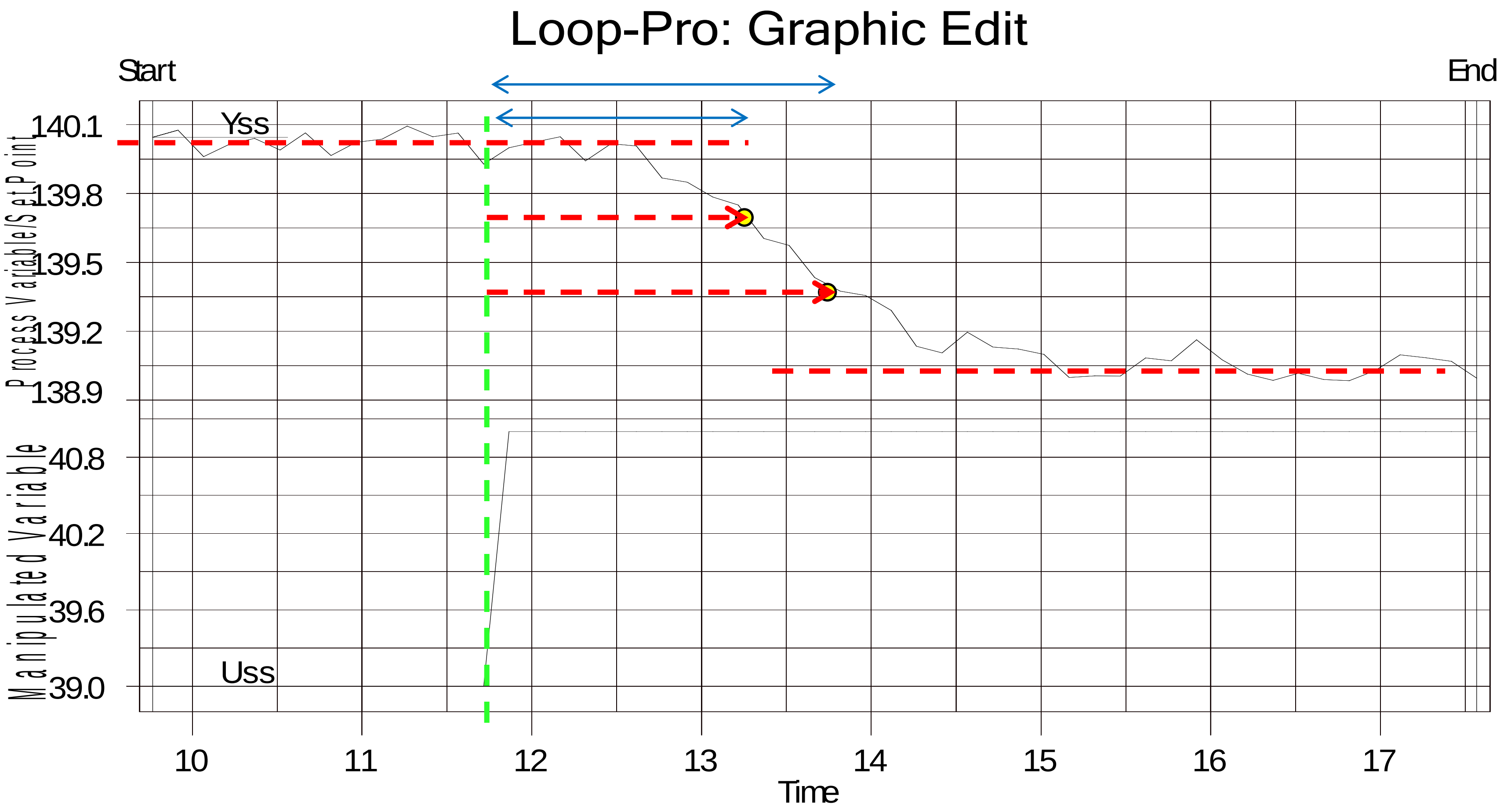
$K = (139.05 - 140) / 2 = -0.475$

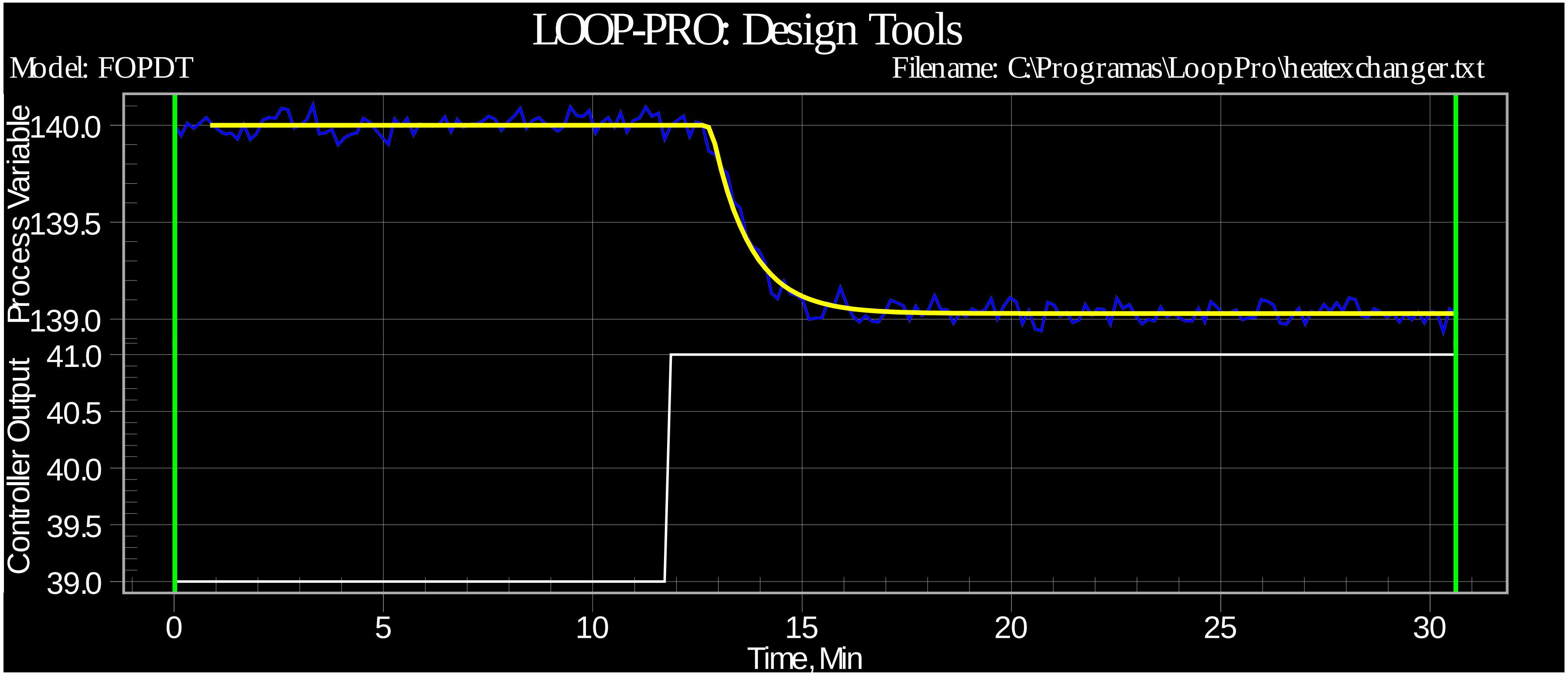
$d = 1.2 \quad \tau = 0.9$

$t_2 = 2.1$

$t_1 = 1.5$

$$G(s) = \frac{-0.475 e^{-1.2s}}{0.9s + 1}$$





$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s+1}$$

Least squares fit

Ziegler Nichols: designed for disturbance rejection

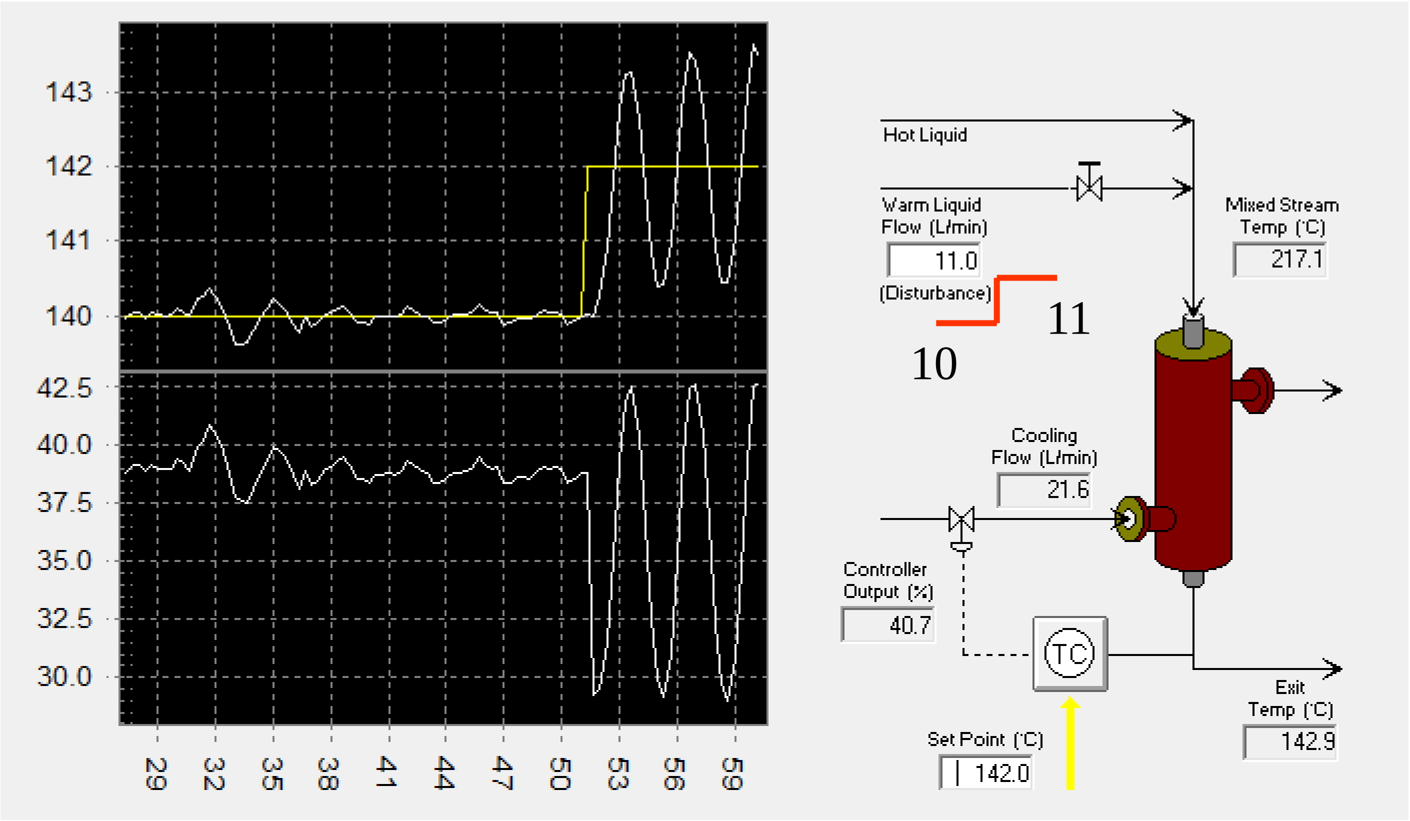
$$G(s) = \frac{-0.475e^{-0.85s}}{1.9s+1}$$

$K_p = 0.9\tau/(Kd) = -4.23$

$T_i = 3.333d = 2.83$

$d / \tau = 0.44$

Model not reliable



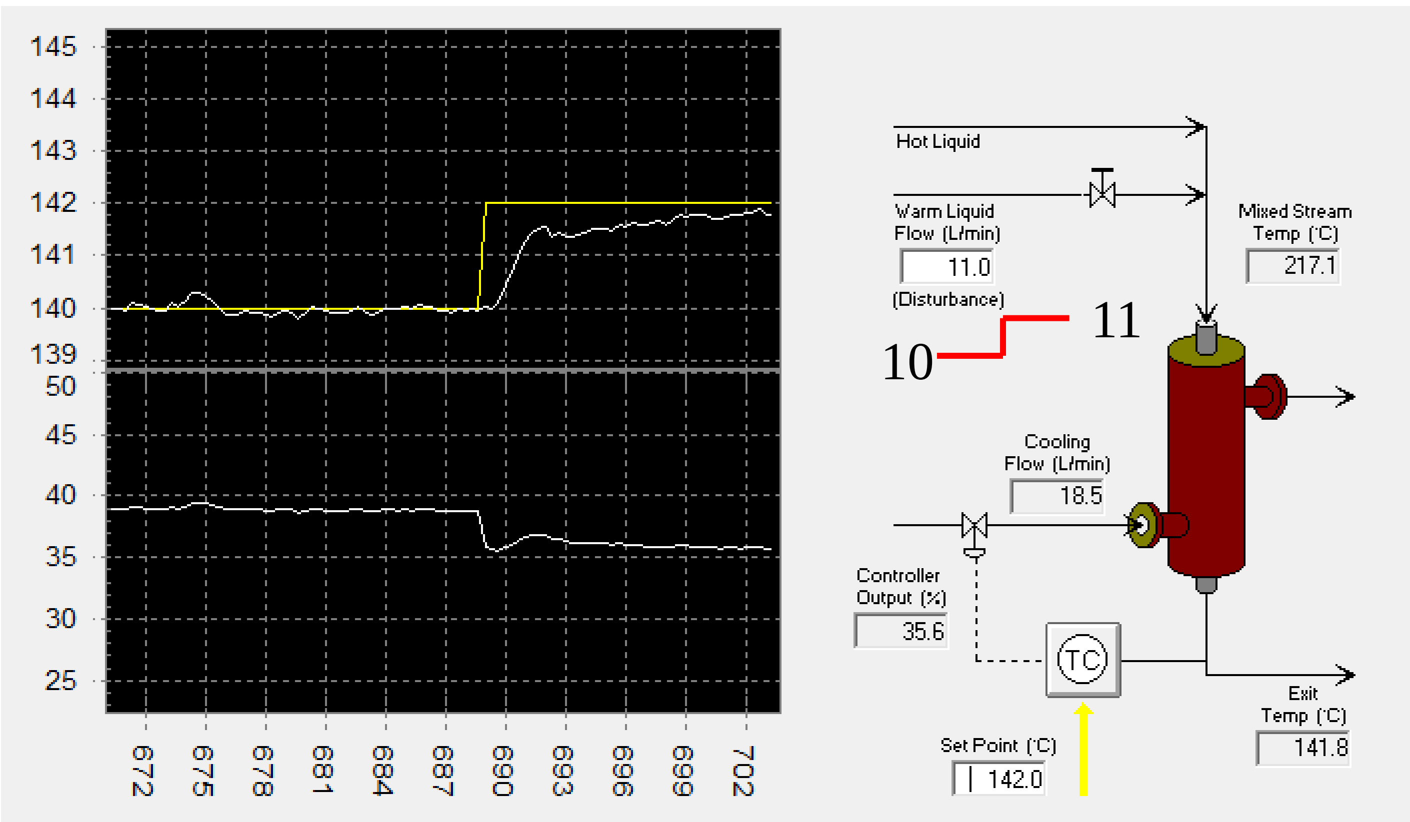
Ziegler Nichols: designed for disturbance rejection

$$G(s) = \frac{-0.475e^{-1.2s}}{0.9s+1}$$

$K_p = 0.9\tau/(Kd) = -1.42$

$T_i = 3.333d = 3.99$

$d/\tau = 1.33$ out of range



Ziegler Nichols: designed for disturbance rejection

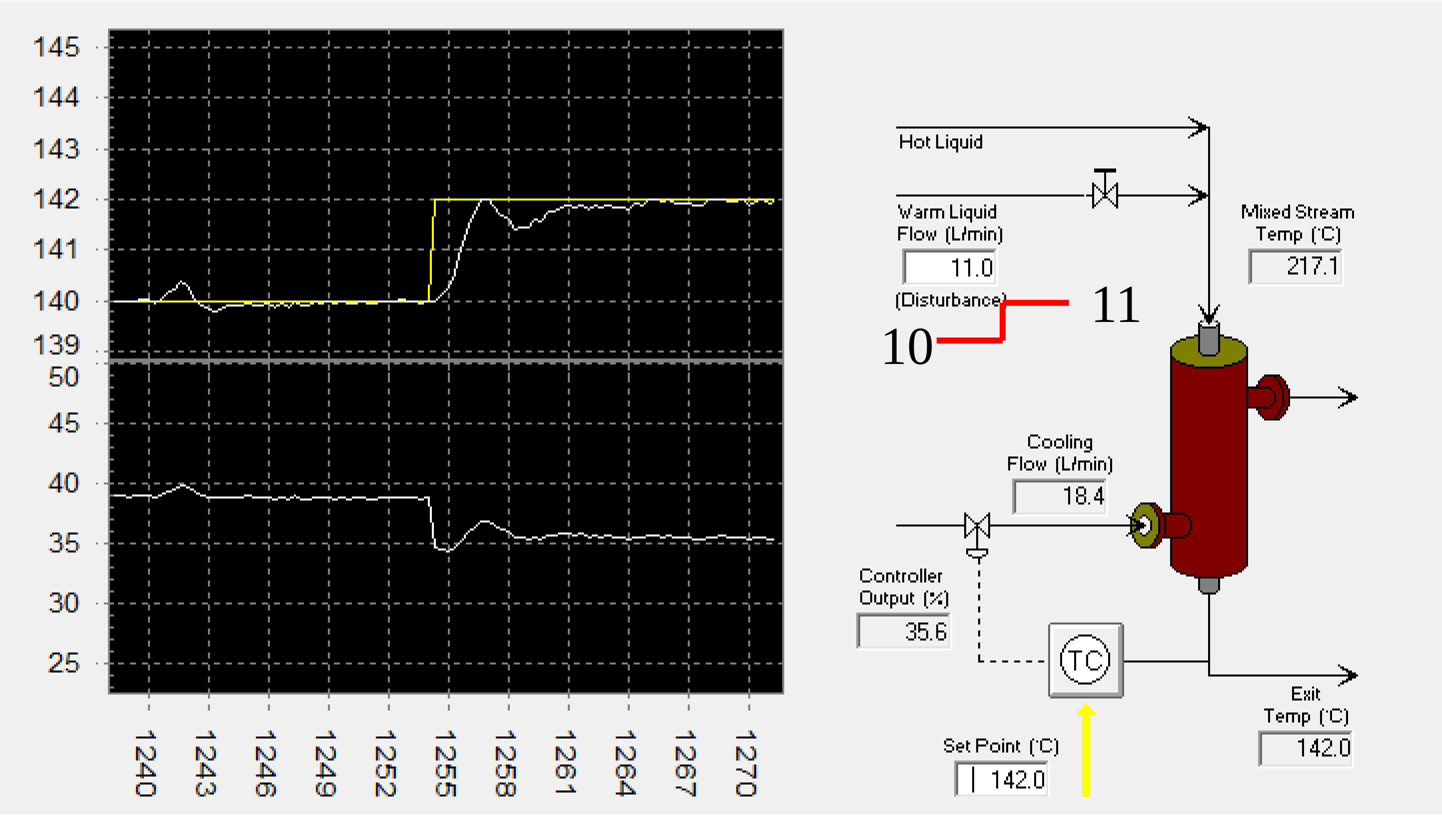
$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s+1}$$

$K_p = 0.9\tau/(Kd) = -1.92$

$T_i = 3.333d = 2.93$

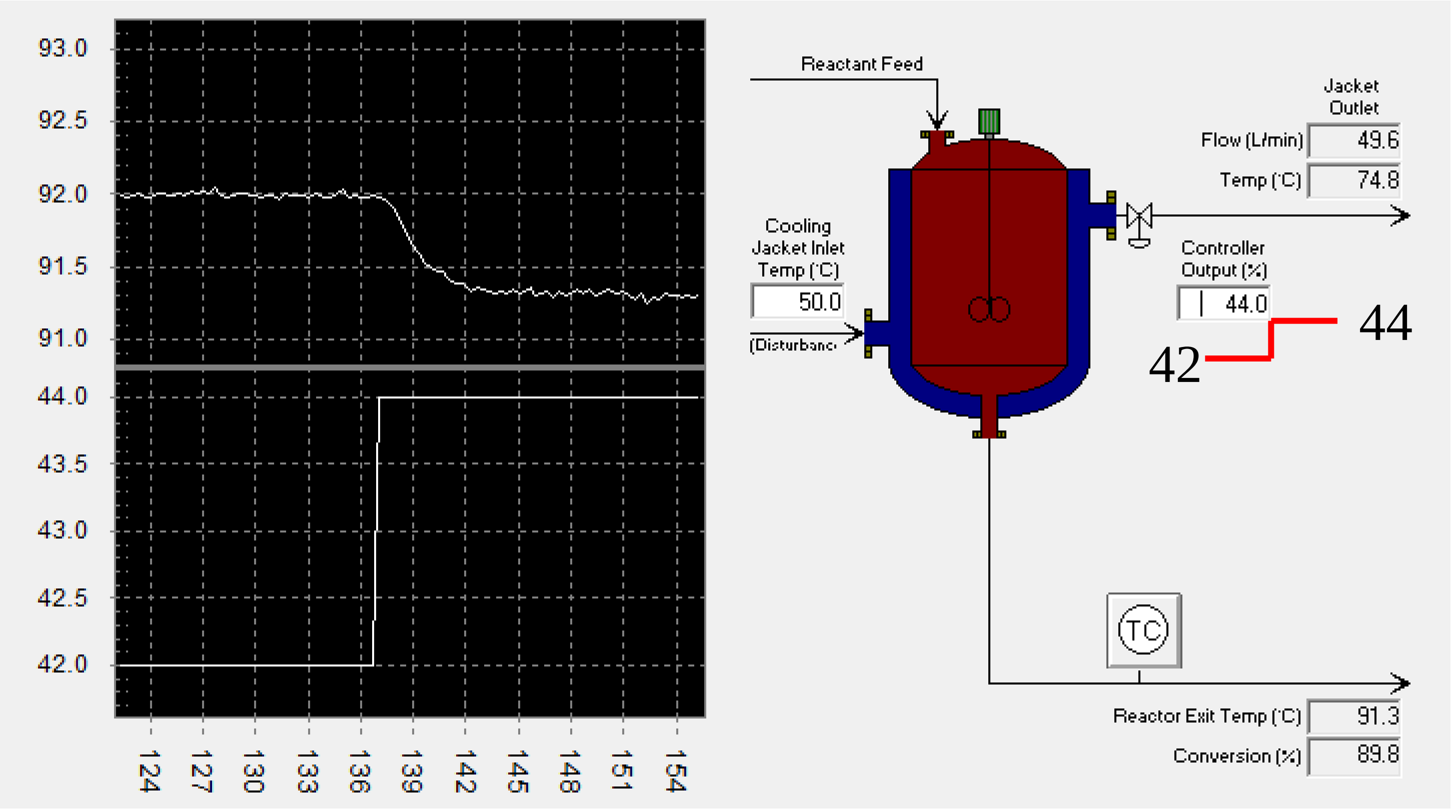
$d/\tau = 0.96$ out of range

Model reliable ,but out of the applicability range of the ZN table



Exothermic Reactor

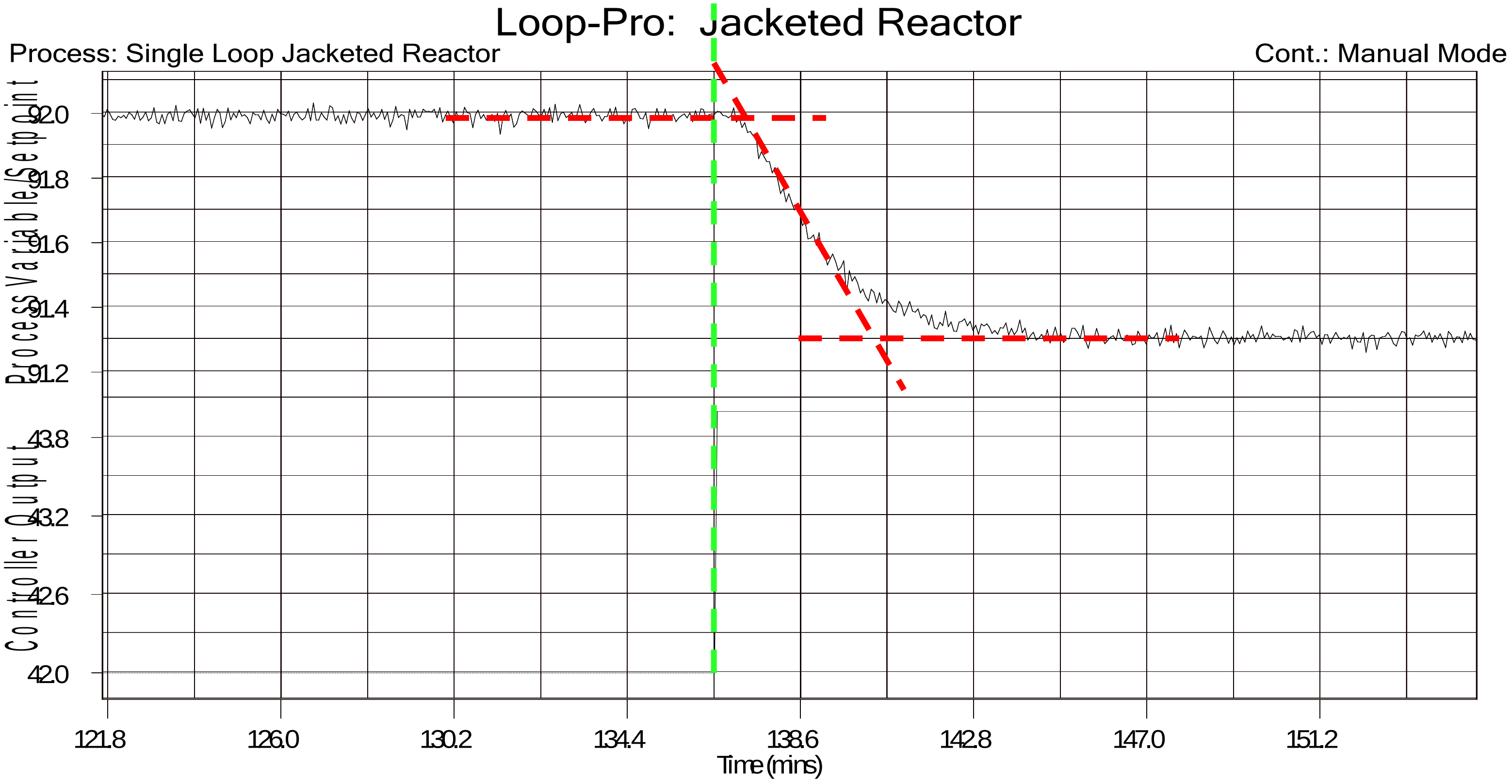
Open loop step test

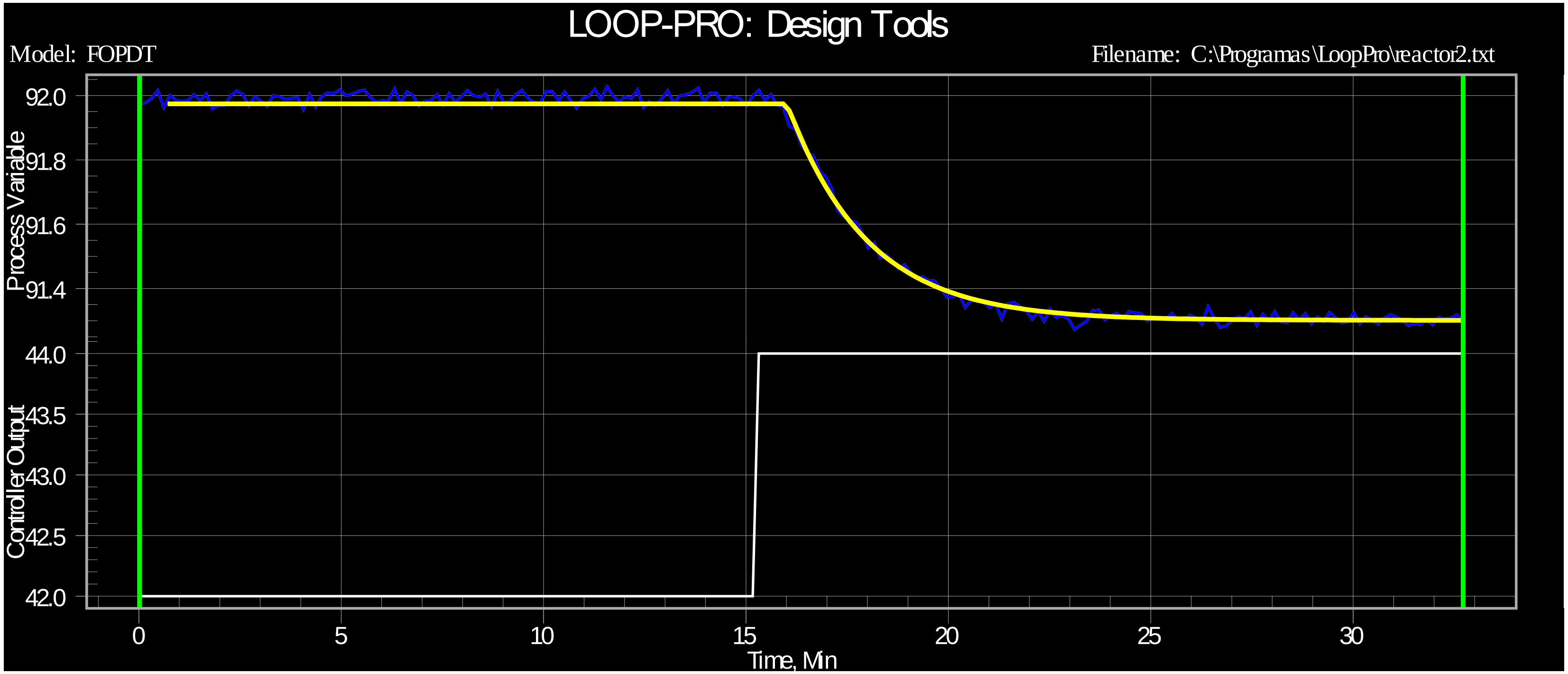


$$K = (91.3 - 92) / 2 = -0.35$$

$$d = 0.7 \quad \tau = 3$$

$$G(s) = \frac{-0.35 e^{-0.7s}}{3s + 1}$$





Least squares fit

$$G(s) = \frac{-0.336e^{-0.62s}}{1.98s+1}$$

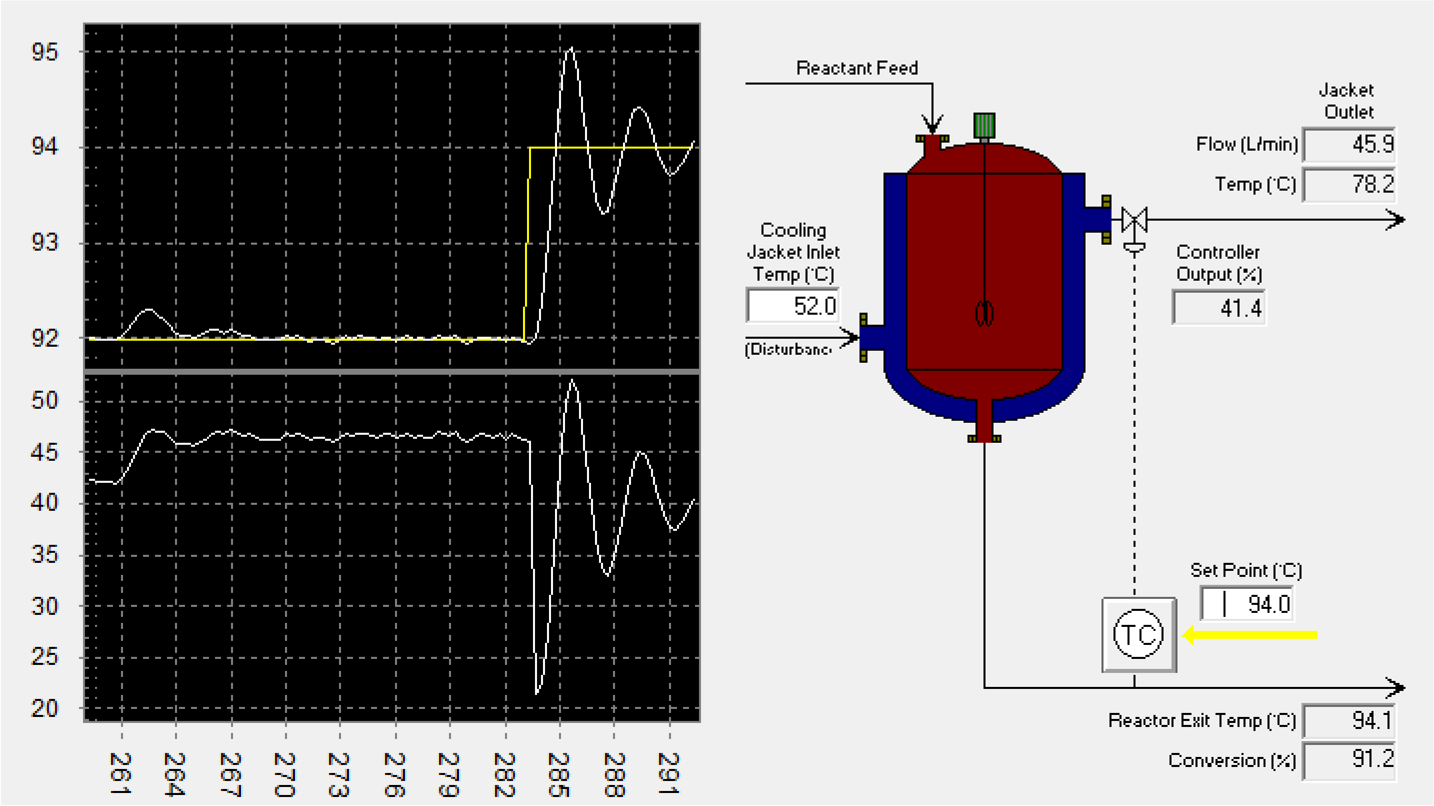
Ziegler Nichols: designed for disturbance rejection

$$G(s) = \frac{-0.35e^{-0.7s}}{3s+1}$$

$K_p = 0.9\tau/(Kd) = -11.02$

$T_i = 3.333d = 2.33$

$d / \tau = 0.23$ ok



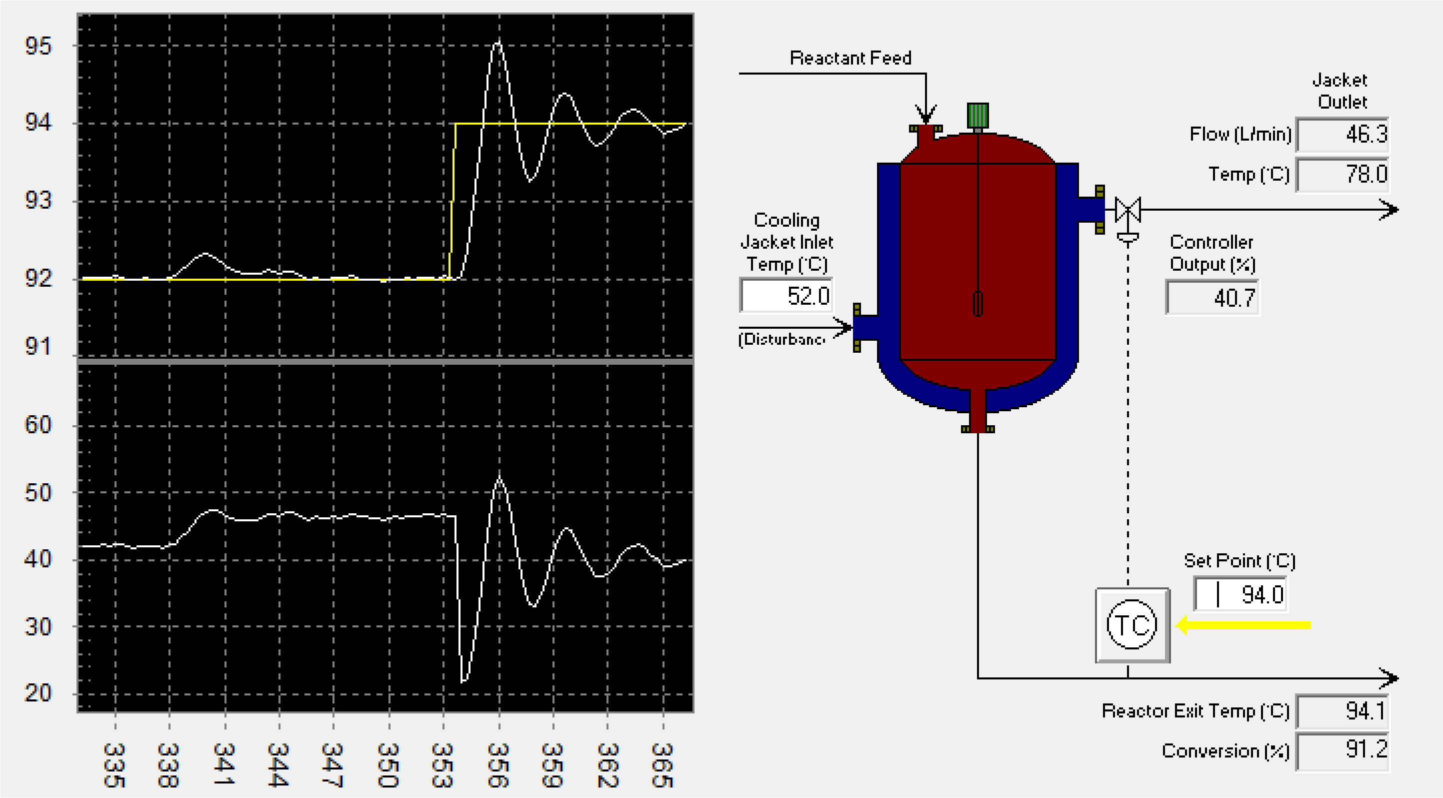
Ziegler Nichols: designed for disturbance rejection

$$G(s) = \frac{-0.336e^{-0.62s}}{1.98s+1}$$

$K_p = 0.9\tau/(Kd) = -8.55$

$T_i = 3.333d = 2.06$

$d / \tau = 0.31$ ok

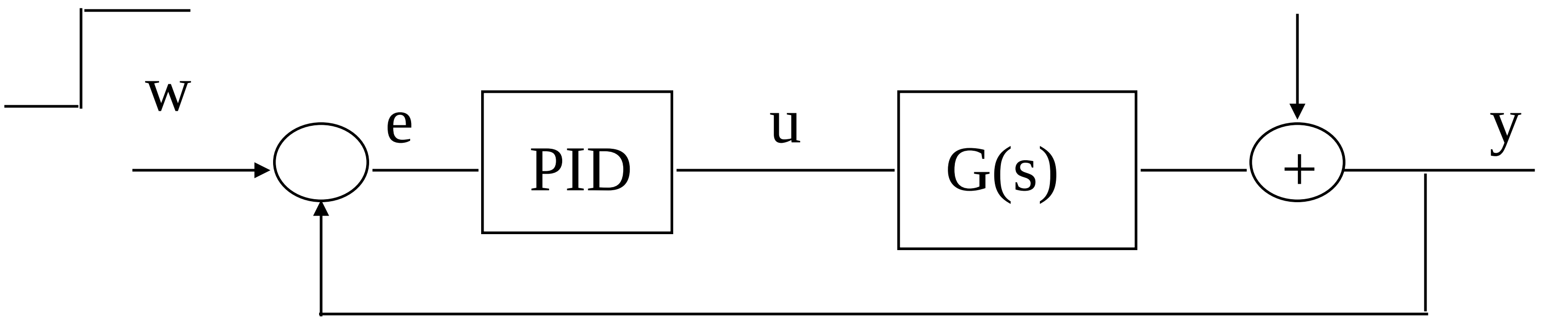


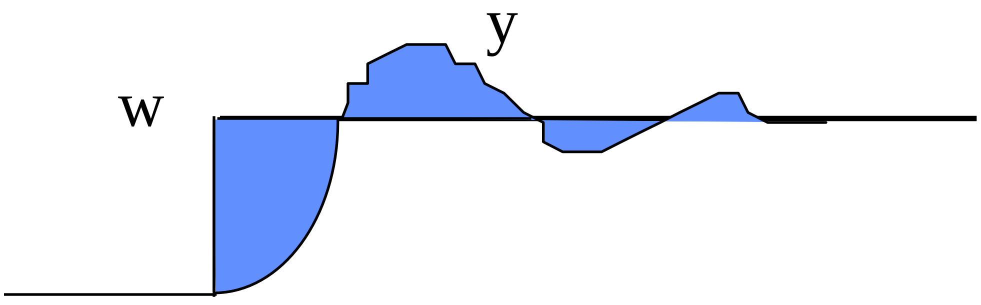
Cohen-Coon Tuning

Controller type	Gain K_c	Integral time T_i	Derivative time T_c
P	$\frac{\tau}{Kd}\left(1+\frac{d}{3\tau}\right)$		
PI	$\frac{\tau}{Kd}\left(0.9+\frac{d}{12\tau}\right)$	$d\frac{30+3d/\tau}{9+20d/\tau}$	
PID	$\frac{\tau}{Kd}\left(1.333+\frac{d}{4\tau}\right)$	$d\frac{32+6d/\tau}{13+8d/\tau}$	$d\frac{4}{11+2d/\tau}$

Same aims as Ziegler-Nichols. It provides better responses in processes with large time delays

Integral of the error minimization



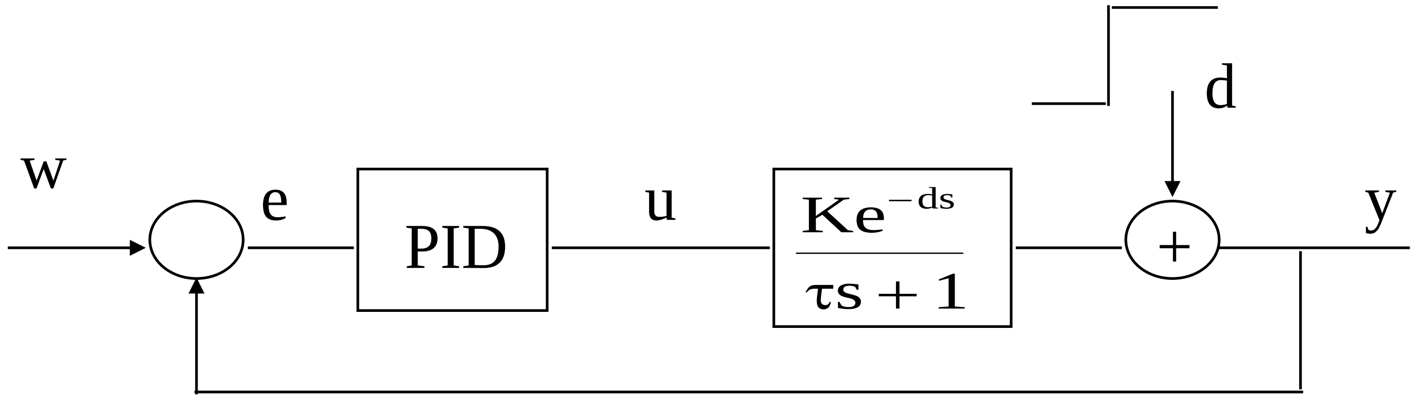


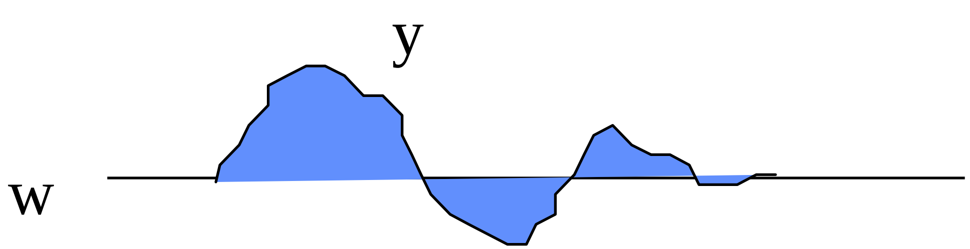
$$\min_{K_p, T_i, T_d} \int f(e(t))dt$$

Numerical minimization

error = f (K_p , T_i, T_d)

Integral of the error minimization





error = f (K_p , T_i, T_d)

$$\min_{K_p, T_i, T_d} \int |e(t)| dt \quad \text{MIAE}$$

$$\min_{K_p, T_i, T_d} \int e(t)^2 dt \quad \text{MISE}$$

$$\min_{K_p, T_i, T_d} \int |e(t)| t dt \quad \text{MITAE}$$

Lopez et al. tuning table

- Developed for Non interactive (parallel) PID (1967)
- For disturbance rejection
- Tuning criteria:

    Integral of the error minimization:

    MIAE |e|

    MISE e^2

    MITAE |e|t
- Based on First order plus delay model
- The tables provide the a and b parameters of the formulas
- Can be applied to monotonous processes with $0.1 < d / \tau < 1$

$$K_p K = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{\tau}{T_i} = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{T_d}{\tau} = a\left(\frac{d}{\tau}\right)^b$$

Lopez et al. tuning table

Parallel PI controllers

Criteria	Proportional	Integral	Derivative
MIAE	a=0.984 b=-0.986	a=0.608 b=-0.707	
MISE	a=1.305 b=-0.959	a=0.492 b=-0.739	
MITAE	a=0.859 b=-0.977	a=0.674 b=-0.68	

$$K_p K = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{\tau}{T_i} = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{T_d}{\tau} = a\left(\frac{d}{\tau}\right)^b$$

K in the same units as K_p

Disturbance rejection tuning

Can be used with monotonous processes with $0.1 < d / \tau < 1$

When applied to digital controllers, increase d by half a sampling period

Lopez et al. tuning table

Parallel PID controllers

Criteria	Proportional	Integral	Derivative
MIAE	a=1.435 b=-0.921	a=0.878 b=-0.749	a=0.482 b=1.137
MISE	a=1.495 b=-0.945	a=1.101 b=-0.771	a=0.560 b=1.006
MITAE	a=1.357 b=-0.947	a=0.842 b=-0.738	a=0.381 b=0.995

$$K_p K = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{\tau}{T_i} = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{T_d}{\tau} = a\left(\frac{d}{\tau}\right)^b$$

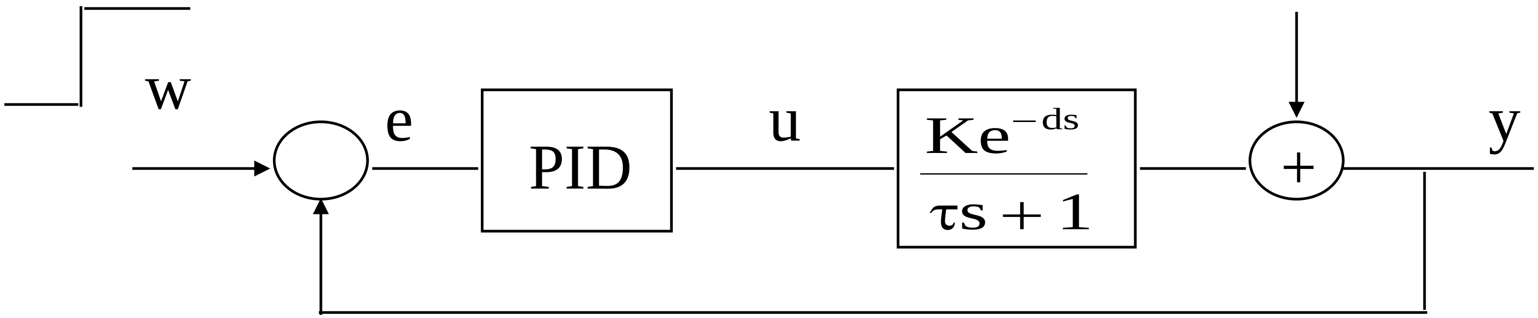
K in the same units as K_p

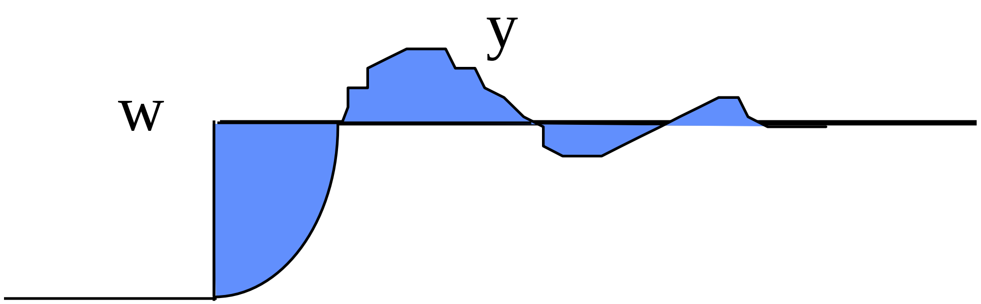
Disturbance rejection tuning

Can be used with monotonous processes with $0.1 < d / \tau < 1$

When applied to digital controllers, increase d by half a sampling period

Integral of the error minimization





$$\min_{K_p, T_i, T_d} \int |e(t)| dt \quad \text{MIAE}$$

$$\min_{K_p, T_i, T_d} \int e(t)^2 dt \quad \text{MISE}$$

$$\min_{K_p, T_i, T_d} \int |e(t)| t dt \quad \text{MITAE}$$

error = f (K_p , T_i, T_d)

Rovira et al. tuning table

- For non interactive (parallel) PI, PID (1969)
- For SP following
- Tuning criteria:

  Minimize the integral of the error:

  MIAE |e|

  MITAE |e|t
- Based on First order plus delay model
- The tables provide the a and b parameters of the formulas
- Can be applied to monotonous processes with $0.1 < d / \tau < 1$

$$K_p K = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{\tau}{T_i} = a\left(\frac{d}{\tau}\right) + b$$

$$\frac{T_d}{\tau} = a\left(\frac{d}{\tau}\right)^b$$

Rovira et al. tuning table

Parallel PI

Criteria	Proportional	Integral	Derivative
MIAE	a=0.758 b=-0.861	a=-0.323 b=1.020	
MITAE	a=0.586 b=-0.916	a=-0.165 b=1.030	
Parallel PID			
MIAE	a=1.086 b=-0.869	a=-0.130 b=0.740	a=0.348 b=0.914
MITAE	a=0.965 b=-0.855	a=-0.147 b=0.796	a=0.308 b=0.929

$$K_p K = a\left(\frac{d}{\tau}\right)^b$$

$$\frac{\tau}{T_i} = a\left(\frac{d}{\tau}\right) + b$$

$$\frac{T_d}{\tau} = a\left(\frac{d}{\tau}\right)^b$$

K in the same units as K_p

Set point following tuning

Can be used with monotonous processes with $0.1 < d / \tau < 1$

When applied to digital controllers, increase d by half a sampling period

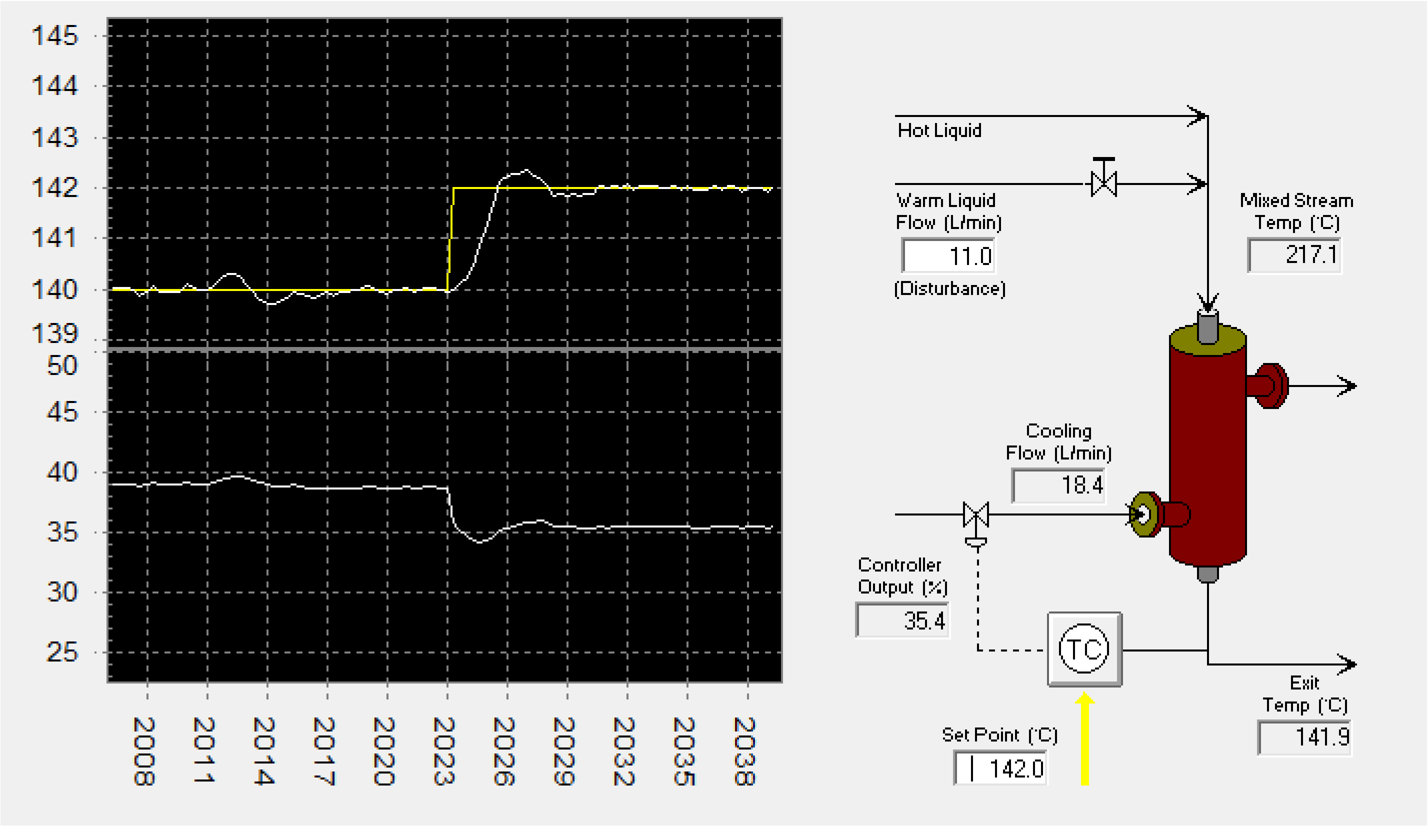
Rovira MIAE:
designed for set
point tracking

$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s+1}$$

$d / \tau = 0.96$
en rango

$$K_p(-0.485) = 0.586\left(\frac{0.88}{0.91}\right)^{-0.916}$$

$$\frac{0.91}{T_i} = -0.165\left(\frac{0.88}{0.91}\right) + 1.03$$



K_p = -1.246
T_i = 1.05

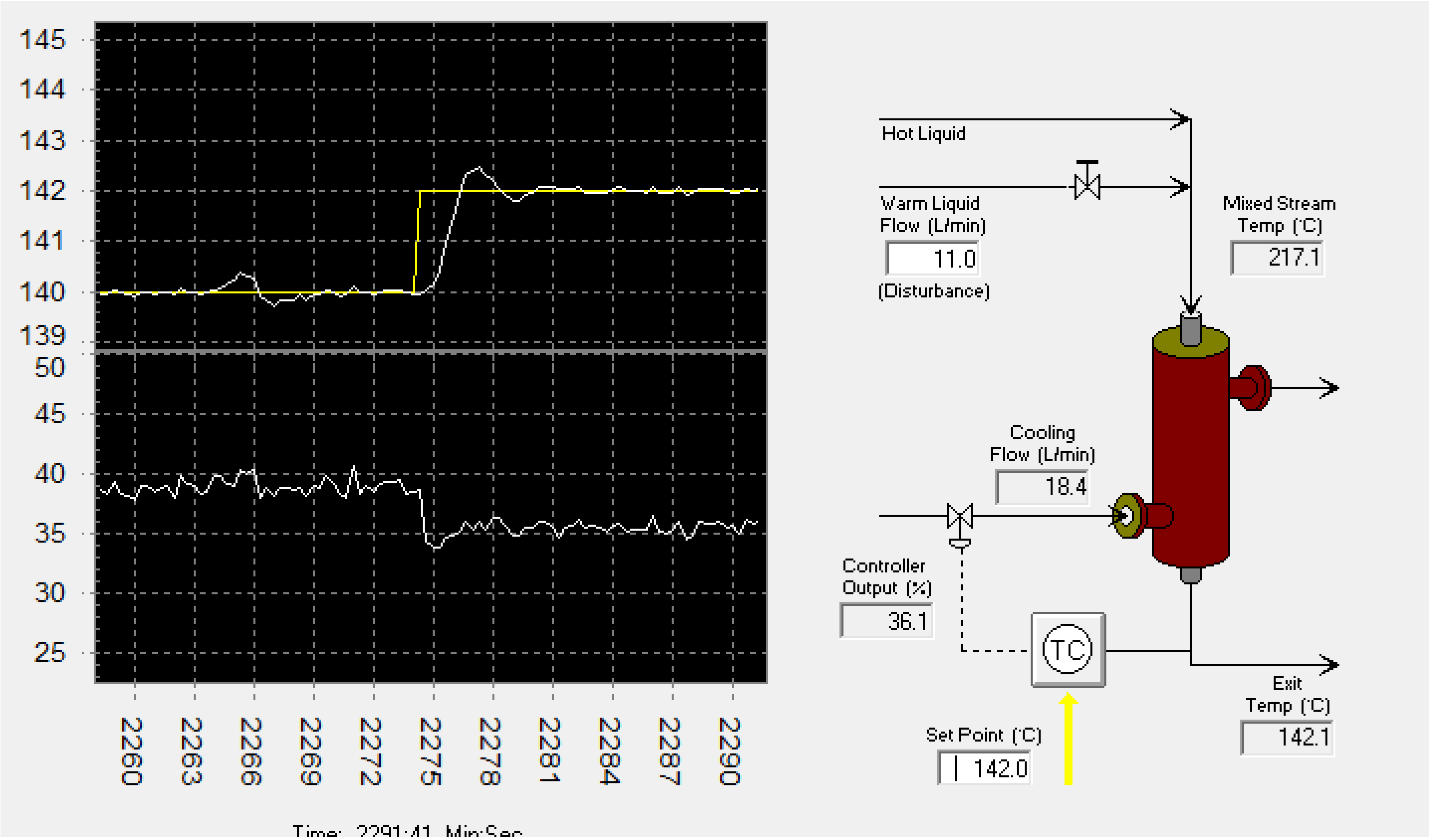
Rovira MIAE: designed for set point tracking

$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s+1}$$

$$K_p(-0.485) = 0.965\left(\frac{0.88}{0.91}\right)^{-0.855}$$

$$\frac{0.91}{T_i} = -0.147\left(\frac{0.88}{0.91}\right) + 0.796$$

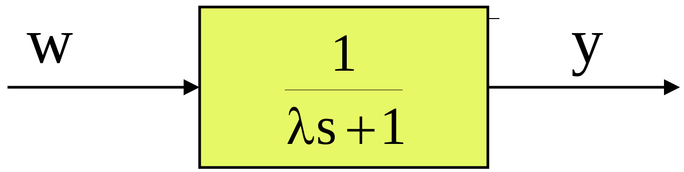
$$\frac{T_d}{0.91} = 0.308\left(\frac{0.88}{0.91}\right)^{0.929}$$



K_p = -2.04
T_i = 1.39
T_d = 0.27

λ Tuning

Type	K_p	T_i	λ recommended λ>0.2τ always
PI processes with integrator	$\frac{2\tau + d}{k(\lambda + d)^2}$	$2\lambda + d$	$\frac{\lambda}{d} > 1.7$
PI	$\frac{4\tau + d}{4K\lambda}$	$\tau + \frac{d}{4}$	$\frac{\lambda}{d} > 1.7$

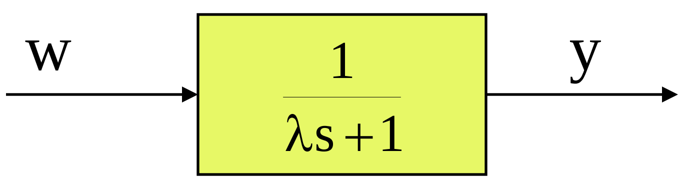


λ Desired closed loop time constant

"Lambda Tuning" refers to all tuning methods where the control loop speed of response is a selectable tuning parameter known as "Lambda". Some rules recommend values of λ higher than the open loop time constant

Rivera-Morari IMC

	Type	K_p	T_i	T_d	λ recommended $\lambda > 0.2\tau$ always
	PI	$\frac{\tau}{K(\lambda + d)}$	τ		$\frac{\lambda}{d} > 1.7$
	Improved PI	$\frac{2\tau + d}{2K\lambda}$	$\tau + \frac{d}{2}$		$\frac{\lambda}{d} > 1.7$
Parallel	PID with filter	$\frac{2\tau + d}{2K(\lambda + d)}$	$\tau + \frac{d}{2}$	$\frac{\tau d}{2\tau + d}$	$\frac{\lambda}{d} > 0.25$



λ Desired closed loop time constant

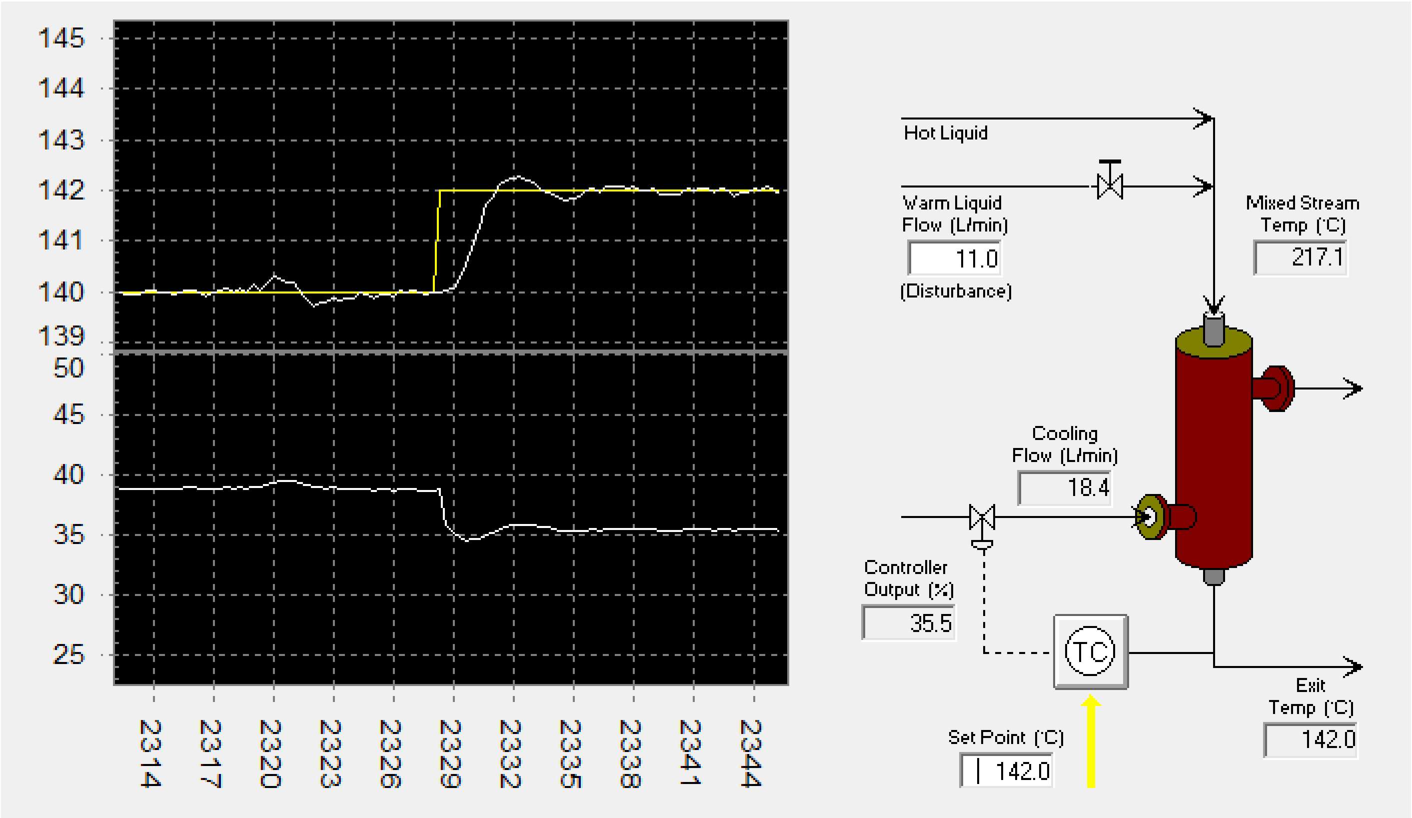
Practical $\lambda = \max(0.1\tau, 0.8d)$ conservative: $\max(0.5\tau, 4d)$

λ Tuning

$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s+1}$$

$$K_p = \frac{4\tau + d}{4K\lambda} \qquad T_i = \tau + \frac{d}{4}$$

$$\lambda / d = 2.27$$



$K_p = -1.16$

$T_i = 1.13$

Lambda tuning $\lambda = 2$

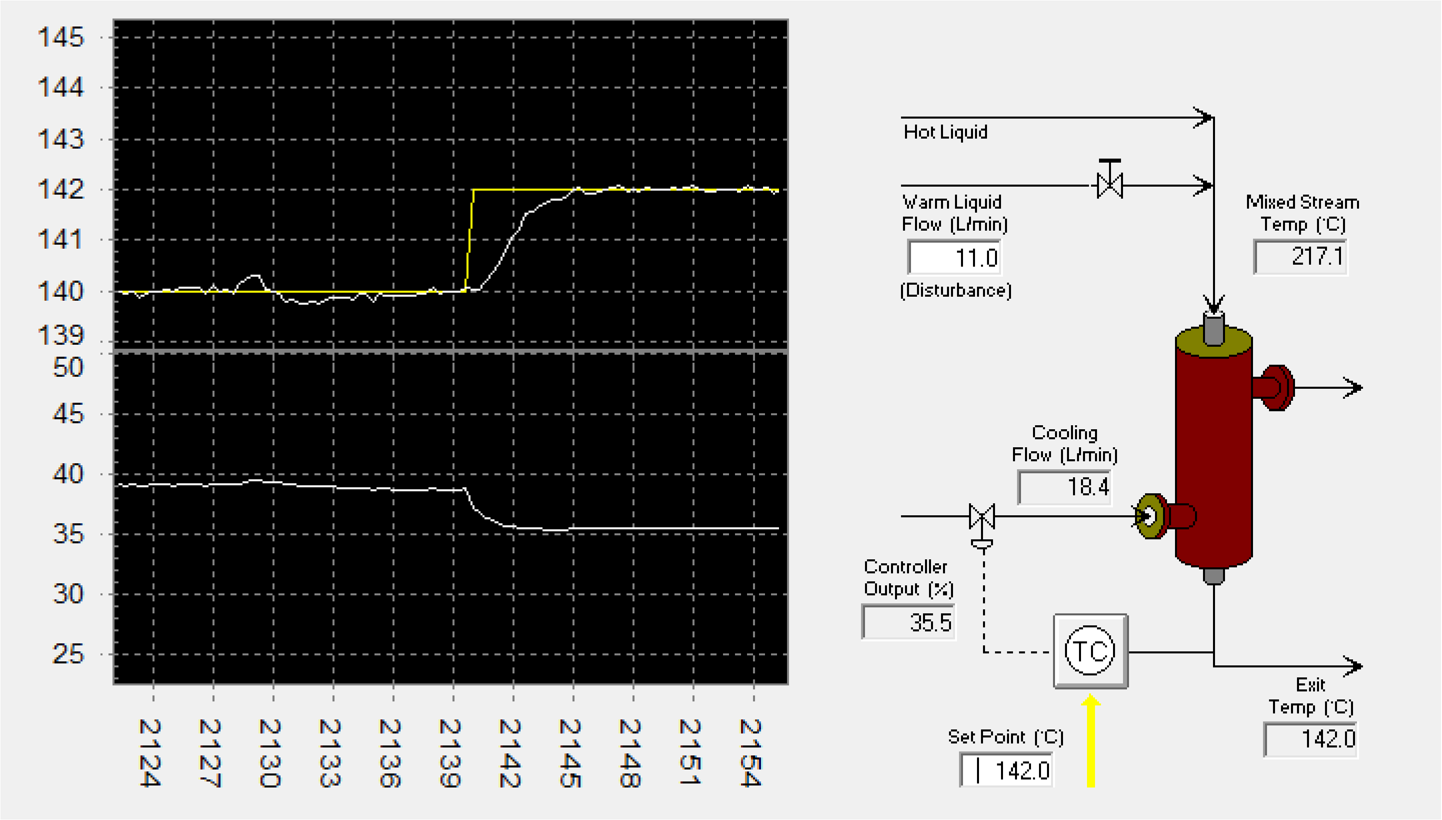
IMC Tuning

$$G(s) = \frac{-0.485e^{-0.88s}}{0.91s + 1}$$

$$K_p = \frac{\tau}{K(\lambda + d)}$$

$$T_i = \tau$$

$$\lambda / d = 2.27$$

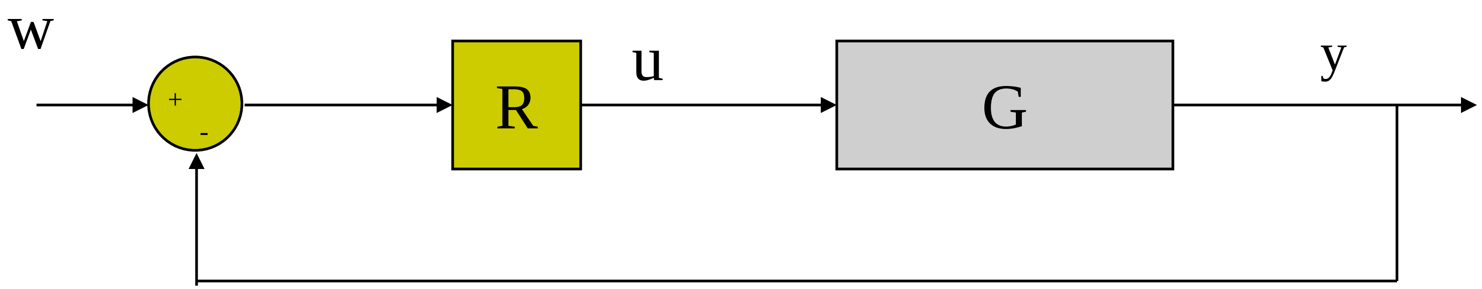


$K_p = -0.651$

$T_i = 0.91$

Lambda tuning $\lambda = 2$

Direct synthesis



$$Y(s) = \frac{GR}{1+GR} W(s)$$

M(s) = Desired closed loop TF

$$M(s) = \frac{GR}{1+GR}$$

$$R(s) = \frac{M(s)}{G(s)(1-M(s))}$$

Direct synthesis of PID controllers

Methodology:

- Start from a low order G(s)
- Choose the desired M(s) as a low order TF
- Compute R(s) and identify the corresponding PID parameters

$$R(s) = \frac{M(s)}{G(s)(1-M(s))} = \frac{\frac{1}{\lambda s+1}}{\frac{K}{s}\left(1-\frac{1}{\lambda s+1}\right)} = \frac{s}{K(\lambda s+1-1)} = \frac{1}{K\lambda}$$

$$M(s) = \frac{1}{\lambda s+1} \qquad G(s) = \frac{K}{s}$$

P controller with $K_p = 1/K\lambda$

Direct synthesis of PID controllers

If: $M(s) = \frac{1}{\lambda s + 1}$ $G(s) = \frac{K}{\tau s + 1}$

$$R(s) = \frac{M(s)}{G(s)(1 - M(s))} = \frac{\frac{1}{\lambda s + 1}}{\frac{K}{\tau s + 1}(1 - \frac{1}{\lambda s + 1})} = \frac{\tau s + 1}{K(\lambda s + 1 - 1)} = \frac{\tau s + 1}{K \lambda s} = \frac{\tau}{K\lambda} \frac{\tau s + 1}{\tau s}$$

$$PI = \frac{K_p (T_i s + 1)}{T_i s}$$

PI controller with

$K_p = \tau / K\lambda$ $T_i = \tau$

Direct synthesis of PID controllers

If: $$M(s) = \frac{1}{\lambda s + 1} \qquad G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

$$R(s) = \frac{M(s)}{G(s)(1 - M(s))} = \frac{\frac{1}{\lambda s + 1}}{\frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}(1 - \frac{1}{\lambda s + 1})} = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K(\lambda s + 1 - 1)} =$$

$$= \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K\lambda s} = \frac{(\tau_1 + \tau_2)}{K\lambda} \frac{(\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s + 1)}{(\tau_1 + \tau_2)s}$$

$$\text{PID ideal} = \frac{K_p (T_i T_d s^2 + T_i s + 1)}{T_i s}$$

PID controller with

$K_p = (\tau_1 + \tau_2)/K\lambda$

$T_i = \tau_1 + \tau_2 \quad T_d = \tau_1 \tau_2$

Direct synthesis of PID controllers

If: $M(s) = \frac{1}{\lambda s + 1}$ $\quad G(s) = \frac{K\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}$

$$R(s) = \frac{M(s)}{G(s)(1 - M(s))} = \frac{\frac{1}{\lambda s + 1}}{\frac{K\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}(1 - \frac{1}{\lambda s + 1})} = \frac{s^2 + 2\delta\omega_n s + \omega_n^2}{K\omega_n^2(\lambda s + 1 - 1)} =$$

$$= \frac{s^2 + 2\delta\omega_n s + \omega_n^2}{K\omega_n^2 \lambda s} = \frac{s^2/\omega_n^2 + (2\delta/\omega_n)s + 1}{K\lambda s} =$$

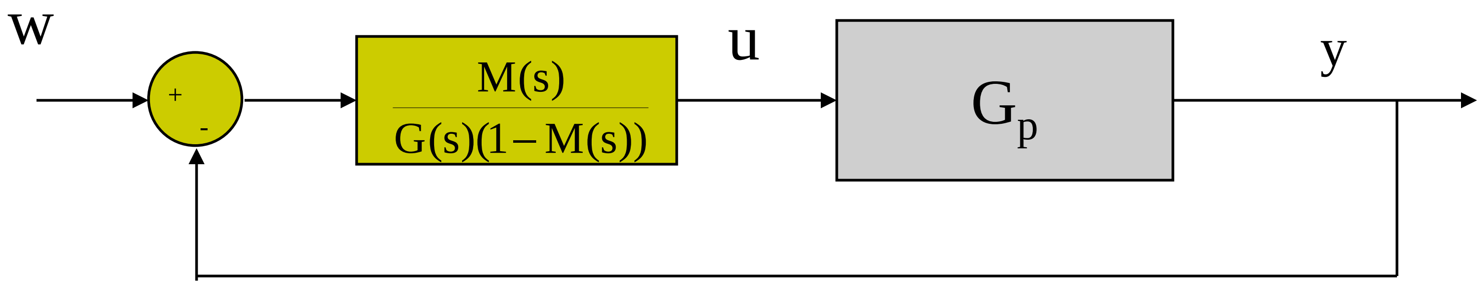
$$= \frac{2\delta}{\omega_n K\lambda} \frac{(2\delta/\omega_n)(1/2\delta\omega_n)s^2 + (2\delta/\omega_n)s + 1}{(2\delta/\omega_n)s}$$

PID controller with:

$$\text{PID ideal} = \frac{K_p(T_i T_d s^2 + T_i s + 1)}{T_i s}$$

$$K_p = \frac{2\delta}{\omega_n K\lambda} \quad T_i = \frac{2\delta}{\omega_n} \quad T_d = \frac{1}{2\delta\omega_n}$$

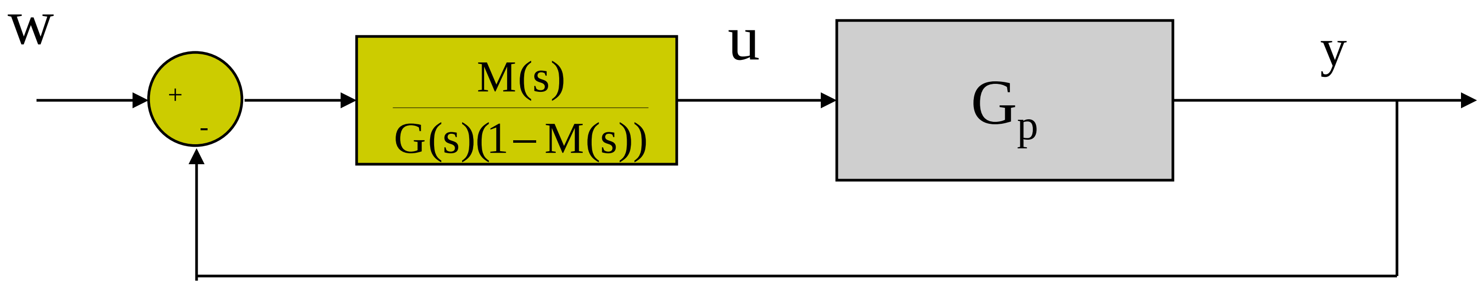
Can we choose M(s) arbitrarily?



$$\mathrm{CLTF} = \frac{G_p \dfrac{M}{G(1-M)}}{1 + G_p \dfrac{M}{G(1-M)}} = \frac{G_p M}{G(1-M) + G_p M} = \frac{G_p M}{G + (G_p - G)M}$$

- In order to answer this question we have to analyse both, the closed loop transfer function CLTF and the controller transfer function, taking into account that our model G(s) is always an approximation to the actual process TF, $G_p(s)$.

Process/model cancellation



$$CLTF = \frac{G_p M}{G + (G_p - G)M} =$$

$$= \frac{N_p M}{D_p \left[G + (G_p - G)M\right]}$$

If, due to the modelling errors, there is no cancellation between G_p and G, then the unstable process poles may appear in the closed loop TF !

$$R(s) = \frac{M(s)}{G(s)(1 - M(s))} =$$

$$= \frac{e^{sd} D(s) M(s)}{N(s)(1 - M(s))}$$

Non-minimum phase systems give unstable controllers!

Models with delays will lead to use future values of e in the controller!

Selecting M(s)

As the open loop zeros and delays must be present in the closed loop response, we should incorporate to M(s) these elements. Then $M(s) = M_0(s)N(s)e^{-ds}$, where $M_o(s)$ is chosen stable, and it is possible to obtain a feasible and stable controller as:

$$R(s) = \frac{e^{sd} D(s) M(s)}{N(s)(1 - M(s))} = \frac{e^{sd} D(s)\left[M_0(s) N(s) e^{-sd}\right]}{N(s)(1 - M_0(s) N(s) e^{-sd})} = \frac{D(s) M_0(s)}{(1 - M_0(s) N(s) e^{-sd})}$$

The order of $M_0(s)$ can be selected to obtain a proper R(s)

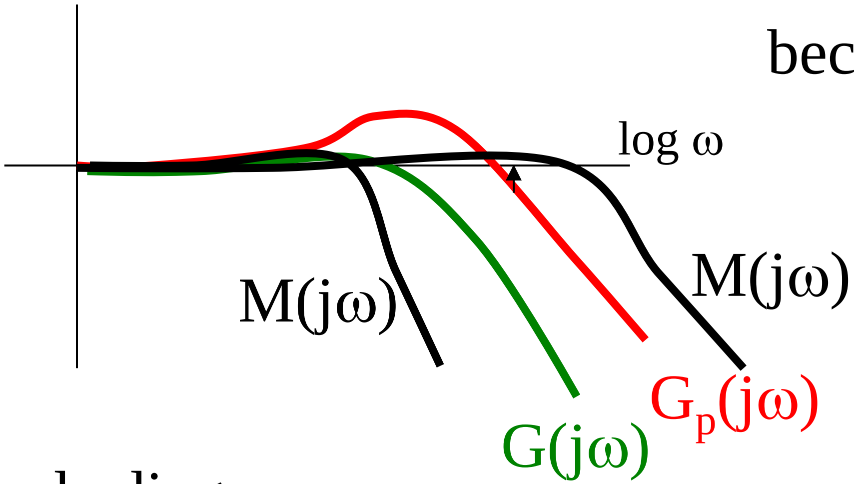
$$R(s) = \frac{D(s) M_0(s)}{(1 - M_0(s) N(s) e^{-sd})} = \frac{D(s) M_{0N}(s)}{(M_{0D}(s) - M_{0N}(s) N(s) e^{-sd})}$$

Where M_{0D} refers to the denominator of $M_0(s)$

Selecting M(s)

$$CLTF = \frac{G_p M}{G + (G_p - G)M}$$

The effect of modelling errors $G_p - G$ in a certain range of frecuencies can be attenuated if M(s) (that is, $M_0(s)$) is chosen small enough in that range, because then $(G_p - G)\, M_0 \rightarrow 0$.



Bode diagrams

Slowing down the desired closed loop response, that is, increasing λ, improves robustness

FOPD

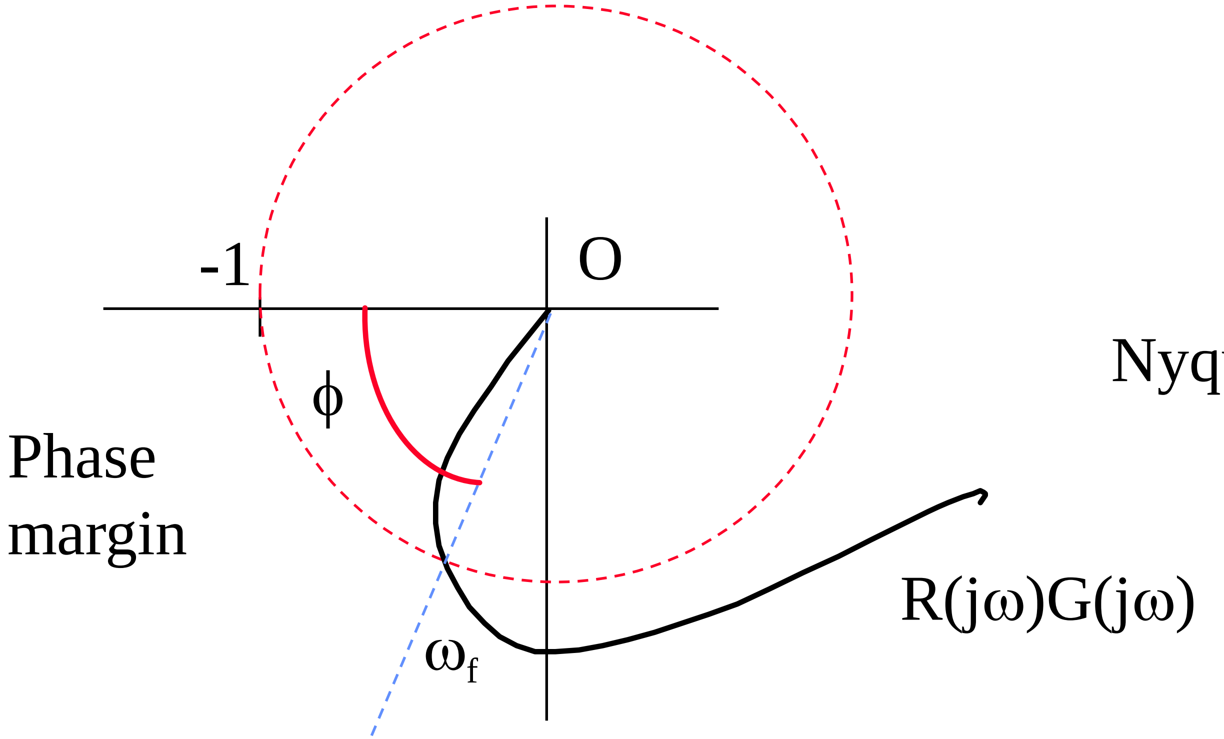
If: $M(s) = \frac{e^{-sd}}{\lambda s + 1}$ $G(s) = \frac{Ke^{-sd}}{\tau s + 1}$

$$R(s) = \frac{M(s)}{G(s)(1 - M(s))} = \frac{\frac{e^{-sd}}{\lambda s + 1}}{\frac{Ke^{-sd}}{\tau s + 1}(1 - \frac{e^{-sd}}{\lambda s + 1})} = \frac{\tau s + 1}{K(\lambda s + 1 - e^{-sd})}$$

Which is not a PI controller

$$PI = \frac{K_p (T_i s + 1)}{T_i s}$$

Phase margin

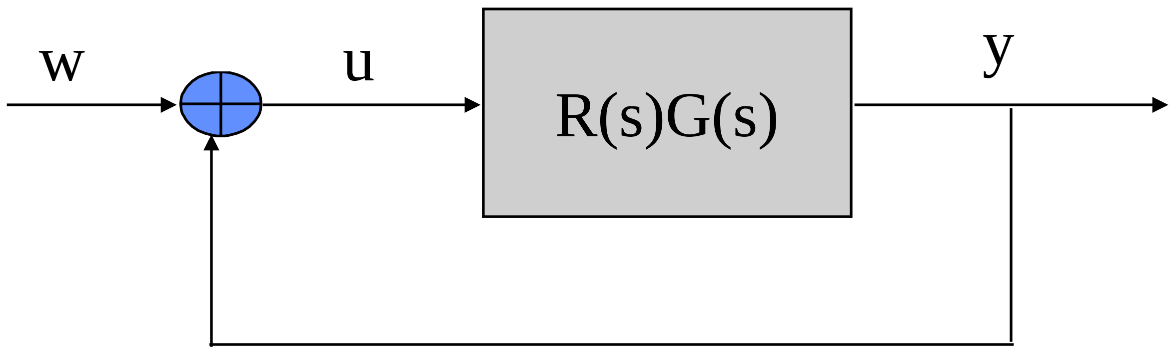


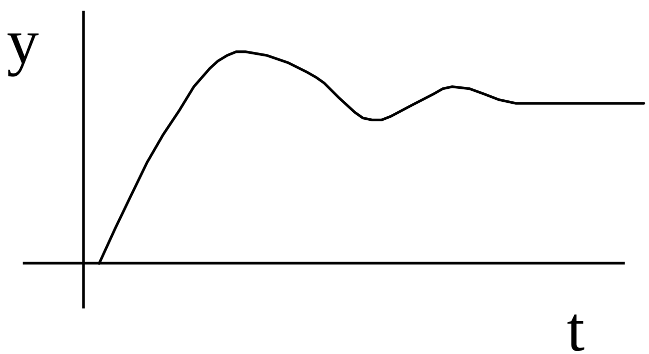
Nyquist Diagram

ω_f highest frequency at which $\left| R(j\omega_f)G(j\omega_f) \right| = 1$

ϕ angle at which $\arg(R(j\omega_f)G(j\omega_f)) = -\pi + \phi$

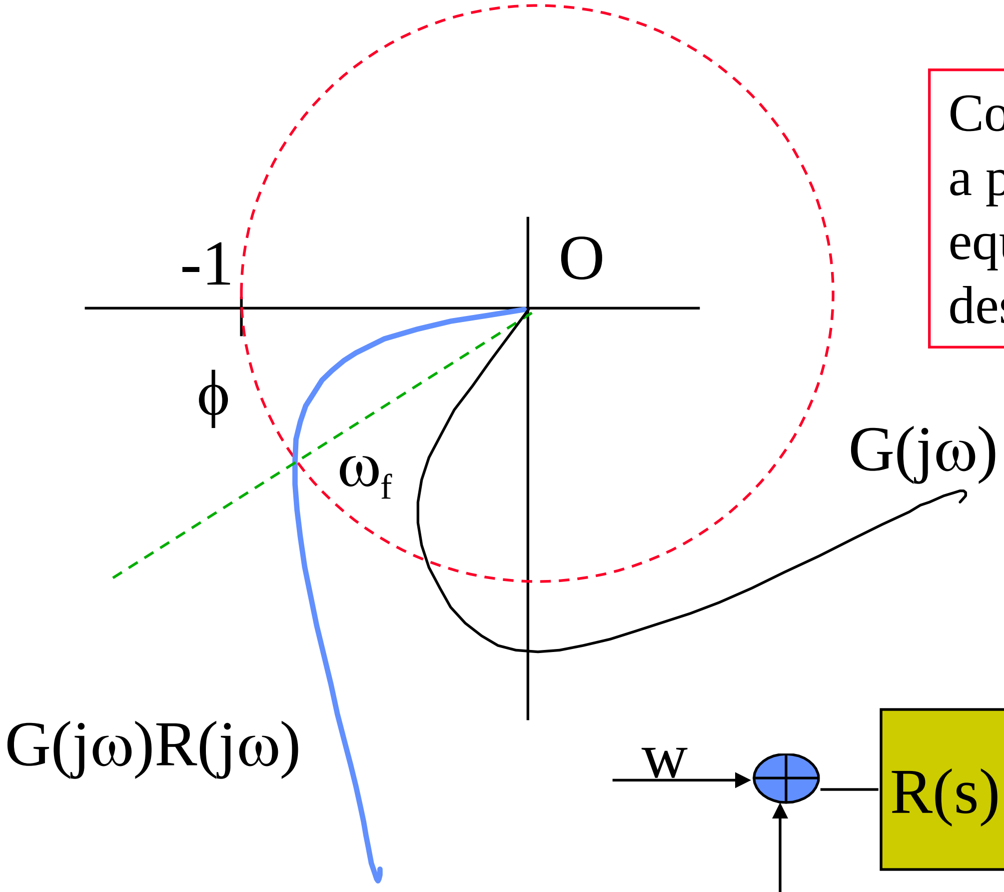
Phase margin



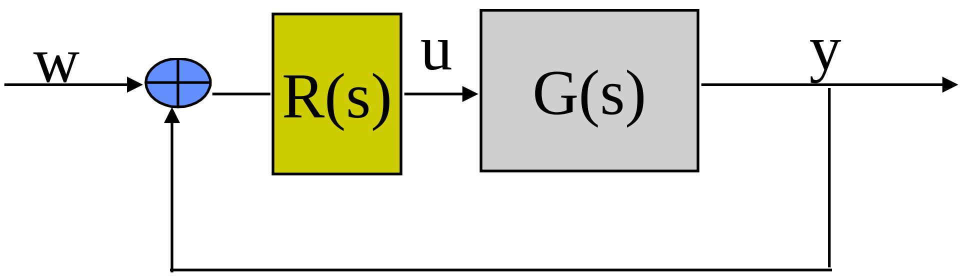


The phase margin ϕ is related to the overshoot and stability
The frequency ω_f is related to the speed of response

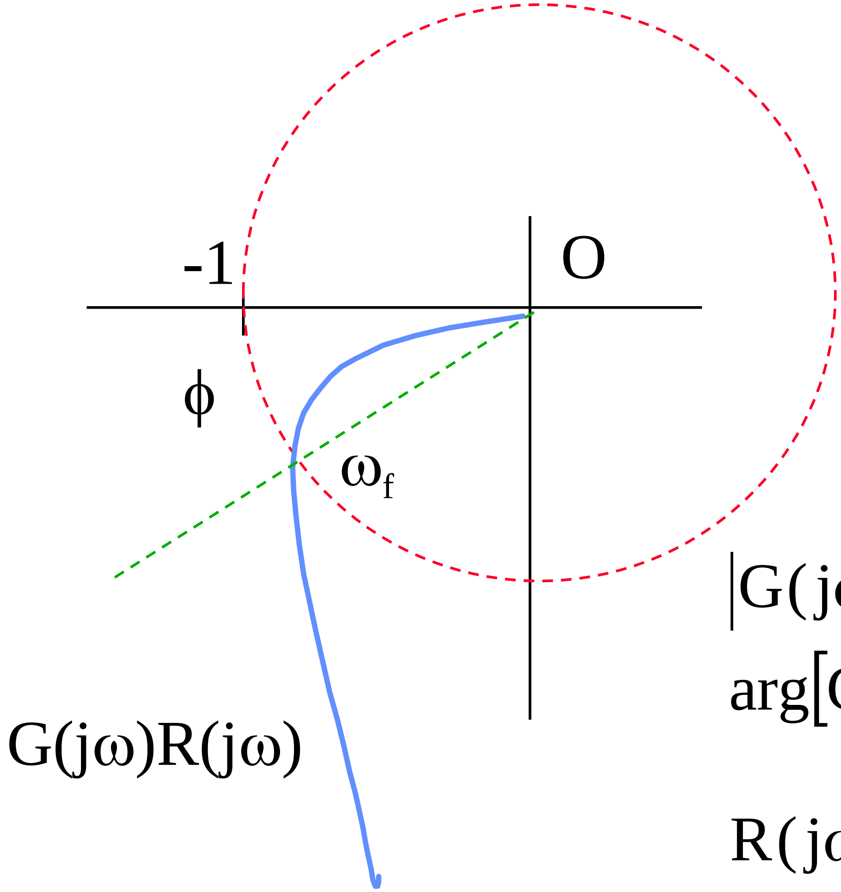
Design with the phase margin



Compute R(s) in order to obtain a phase margin of G(jω)R(jω) equal to the desired ϕ at the desired frequency ω_f



PID design with phase margin specifications



$$|G(j\omega_f)R(j\omega_f)| = 1$$

$$\arg[G(j\omega_f)R(j\omega_f)] = -\pi + \phi$$

$$R(j\omega) = K_p\left[1 + \frac{1}{T_i j\omega} + \frac{T_d j\omega}{1 + 0.1 T_d j\omega}\right]$$

PID design with phase margin specifications

$$\left|G(j\omega_f)R(j\omega_f)\right| = 1$$

$$\arg\left[G(j\omega_f)R(j\omega_f)\right] = -\pi + \phi$$

$$R(j\omega) = K_p\left[1 + \frac{1}{T_i j\omega} + \frac{T_d j\omega}{1 + 0.1T_d j\omega}\right]$$

$$K_p\left|1 + \frac{1}{T_i j\omega_f} + \frac{T_d j\omega_f}{1 + 0.1T_d j\omega_f}\right| = \frac{1}{\left|G(j\omega_f)\right|}$$

$$\arg\left[1 + \frac{1}{T_i j\omega_f} + \frac{T_d j\omega_f}{1 + 0.1T_d j\omega_f}\right] = -\pi + \phi - \arg\left[G(j\omega_f)\right]$$

$$T_d = \alpha T_i \quad \text{con } \alpha = 0......0.25$$

- Two equations and three unknowns: K_p , T_i, T_d
- ω_f and ϕ should be specified
- The solution only exists for a range of values
- Only a point of the Nyquist diagram is required!

PI design with PM specifications

$$K_p \left| 1 + \frac{1}{T_i j\omega_f} \right| = \frac{1}{|G(j\omega_f)|}$$

$$\arg\left[1 + \frac{1}{T_i j\omega_f} \right] = -\pi + \phi - \arg[G(j\omega_f)]$$

$$\arg\left[1 + \frac{1}{T_i j\omega_f} \right] = \arg\left[1 - j\frac{1}{T_i \omega_f} \right] =$$

$$= -\operatorname{arctg}\frac{1}{T_i\omega_f} = -\theta$$

$$\left| 1 + \frac{1}{T_i j\omega_f} \right| = \sqrt{1 + \left(\frac{1}{T_i\omega_f} \right)^2} =$$

$$= \sqrt{1 + \operatorname{tg}^2\theta} = \sec\theta$$

$$\theta = \pi - \phi + \arg[G(j\omega_f)]$$

$$T_i = \frac{1}{\omega_f \operatorname{tg}\theta}$$

$$K_p = \frac{\cos\theta}{|G(j\omega_f)|}$$

PD design with PM specifications

$$K_p \left| 1 + \frac{T_d\, j\omega_f}{1 + 0.1 T_d\, j\omega_f} \right| = \frac{1}{|G(j\omega_f)|}$$

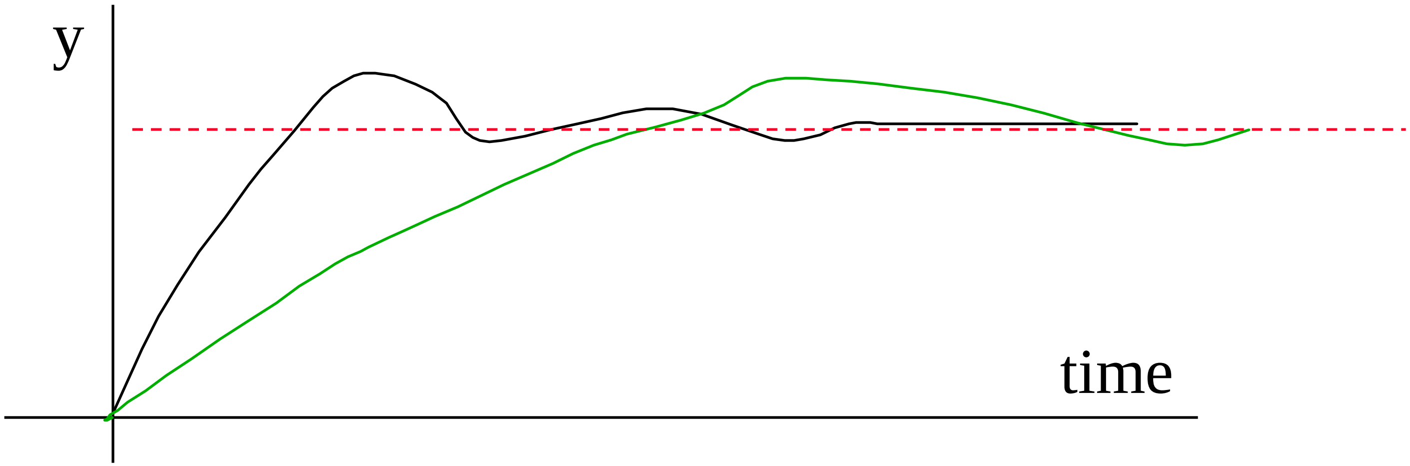
$$\arg\left[1 + \frac{T_d\, j\omega_f}{1 + 0.1 T_d\, j\omega_f} \right] = -\pi + \phi - \arg[G(j\omega_f)]$$

$$K_p = \left[|G(j\omega_f)| \sqrt{1 + \left(\frac{T_d \omega_f}{1 + 0.1 T_d \omega_f} \right)^2} \right]^{-1}$$

$$T_d = \frac{-1 + \sqrt{1 - 0.44\, \mathrm{tg}\,\theta}}{0.22 \omega_f\, \mathrm{tg}\,\theta}$$

$$\theta = \pi - \phi - \arg(G(j\omega_f))$$

Controller design with phase margin specifications



The overshoot decreases with ϕ

Larger values of ω_f give faster responses and more active control signals

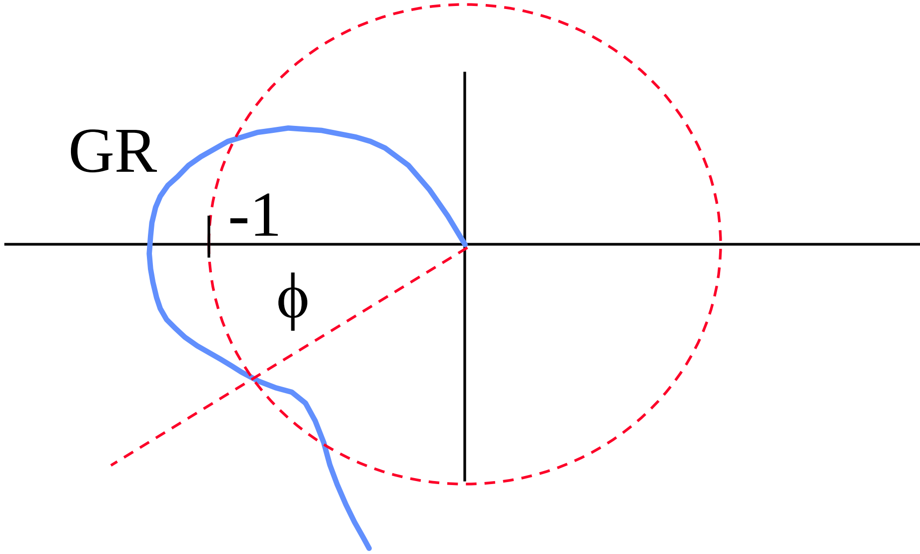
Controller design with phase margin specifications

The fulfilment of the equations:

$$\left|G(j\omega_f)R(j\omega_f)\right| = 1$$

$$\arg\left[G(j\omega_f)R(j\omega_f)\right] = -\pi + \phi$$

Does not guarantee the closed loop stability!

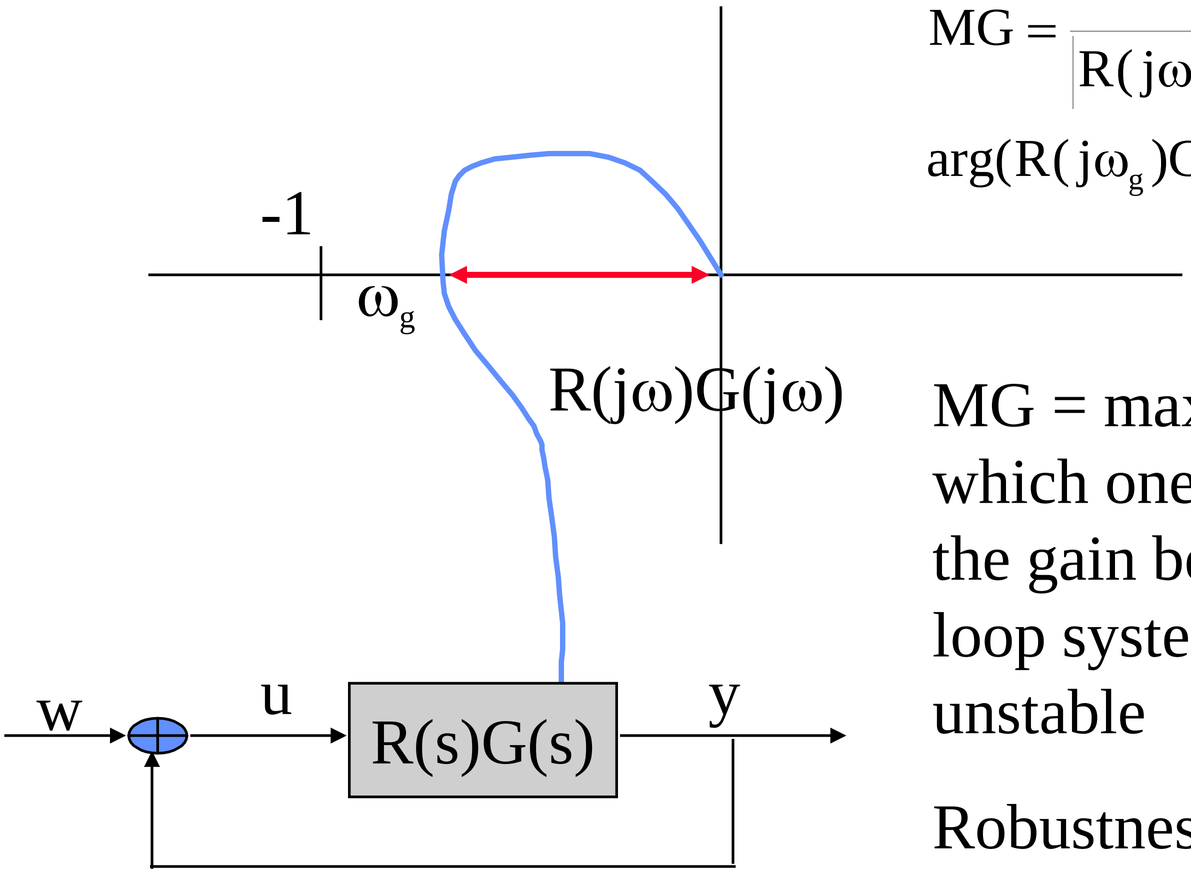


Notice that the PM is defined as a function of the highest frequency satisfying |GR| = 1, but several solutions are possible

Gain Margin

$$MG = \frac{1}{\left| R(j\omega_g)G(j\omega_g) \right|}$$

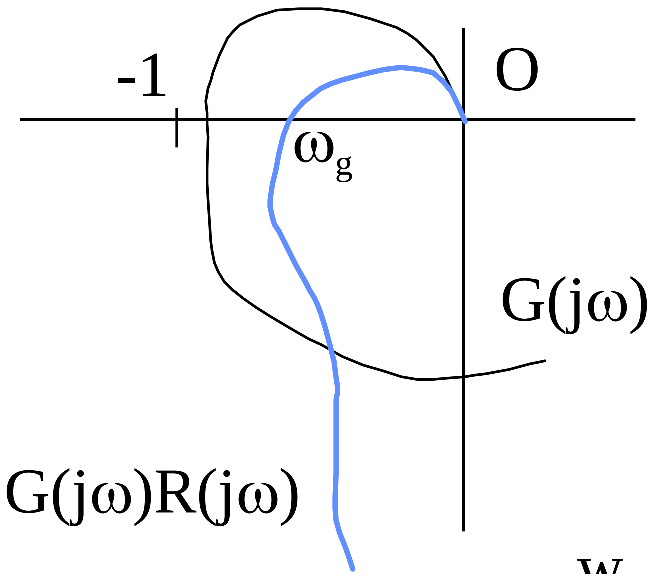
$$\arg(R(j\omega_g)G(j\omega_g)) = -\pi$$



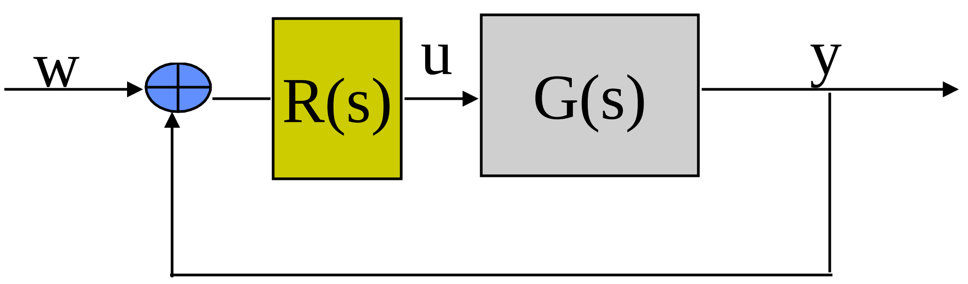
MG = maximum factor by which one can increment the gain before the closed loop system becames unstable

Robustness measurement

Controller design using the GM



Compute R(s) in order to obtain a given gain margin M at a frequency ω_g



Controller design using the GM

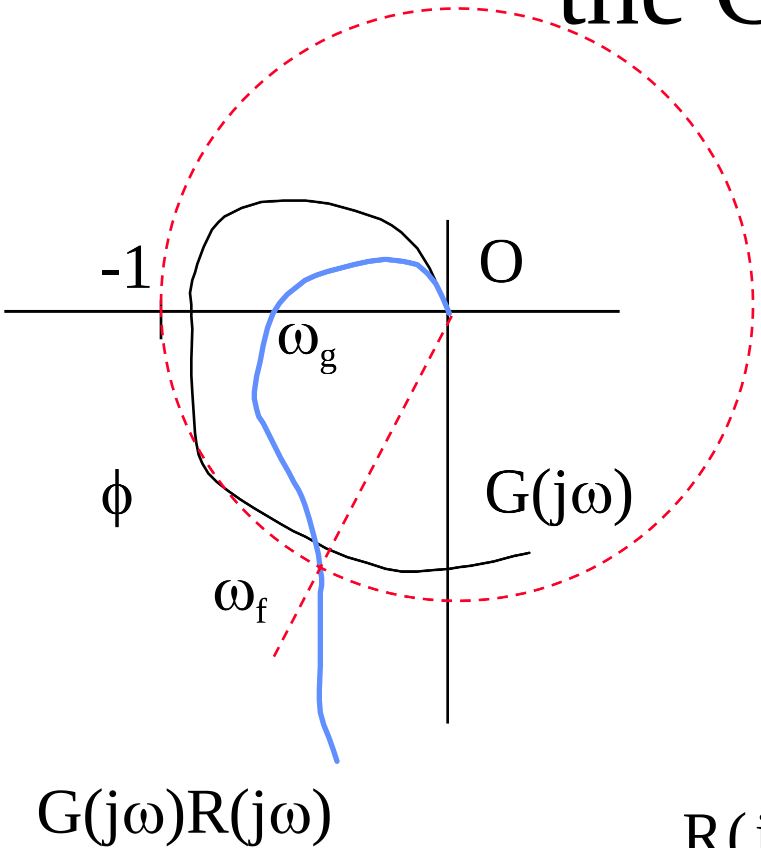
$$\left| G(j\omega_g)R(j\omega_g) \right| = \frac{1}{M_g}$$

$$\arg\left[G(j\omega_g)R(j\omega_g) \right] = -\pi$$

$$R(j\omega) = K_p \left[1 + \frac{1}{T_i j\omega} + \frac{T_d j\omega}{1 + 0.1 T_d j\omega} \right]$$

Same design problems as with the PM

Controller design with the PM and the GM



$$\left|G(j\omega_g)R(j\omega_g)\right| = \frac{1}{M_g}$$

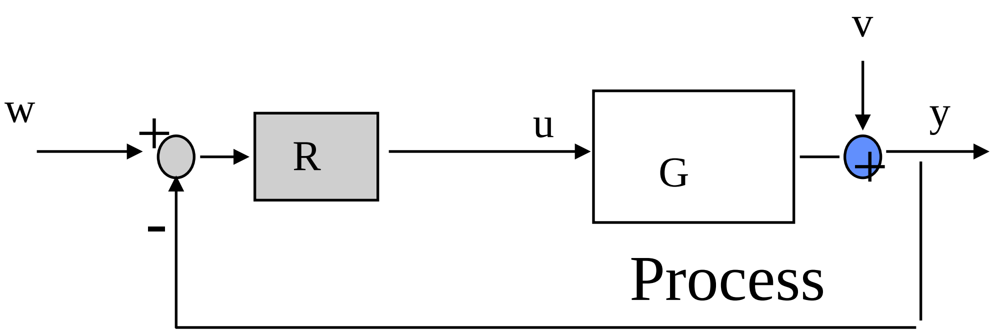
$$\arg\left[G(j\omega_g)R(j\omega_g)\right] = -\pi$$

$$\left|G(j\omega_f)R(j\omega_f)\right| = 1$$

$$\arg\left[G(j\omega_f)R(j\omega_f)\right] = -\pi + \phi$$

$$R(j\omega) = K_p\left[1 + \frac{1}{T_i j\omega} + \frac{T_d j\omega}{1 + 0.1 T_d j\omega}\right]$$

Four transfer functions



$$y = \frac{GR}{1+GR} w + \frac{1}{1+GR} v$$

S_{wy} S_{vy}

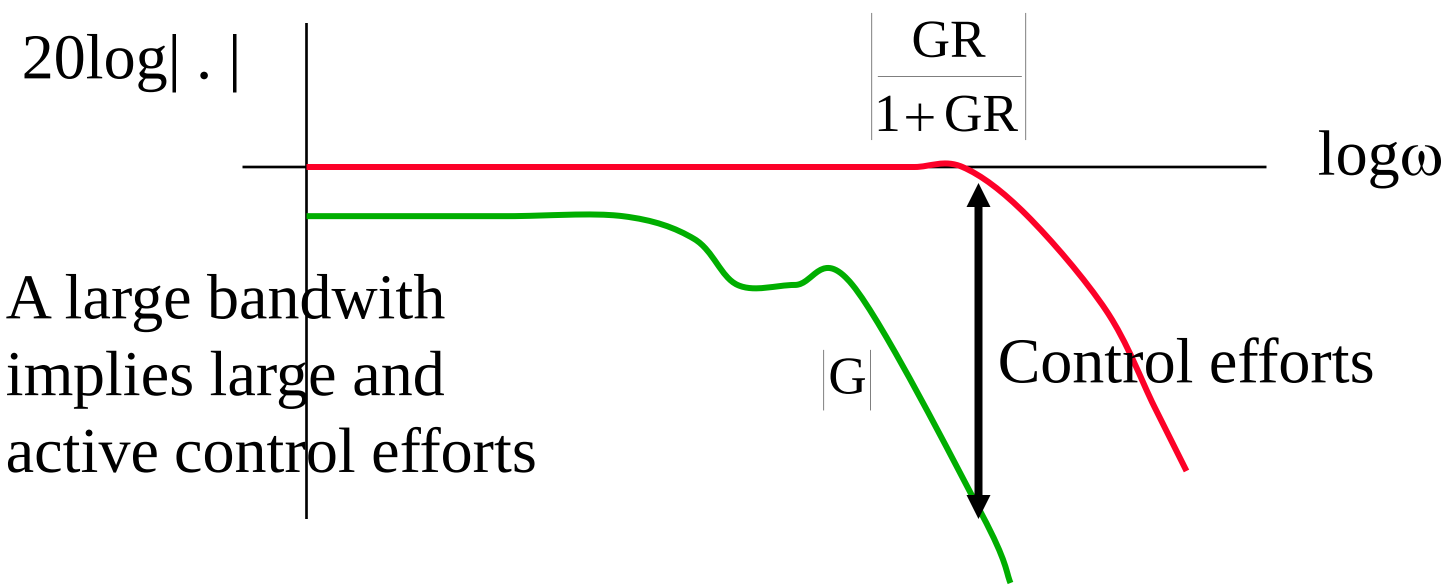
$$u = \frac{R}{1+GR} w - \frac{R}{1+GR} v$$

S_{wu} S_{vu}

Control efforts

$$S_{uw} = \frac{GR}{1+GR} = G\,\frac{R}{1+GR} = G\,S_{uw}$$

$$20\log\left|\frac{GR(j\omega)}{1+GR(j\omega)}\right| - 20\log\left|G(j\omega)\right| = 20\log\left|\frac{R(j\omega)}{1+GR(j\omega)}\right|$$



Disturbance rejection

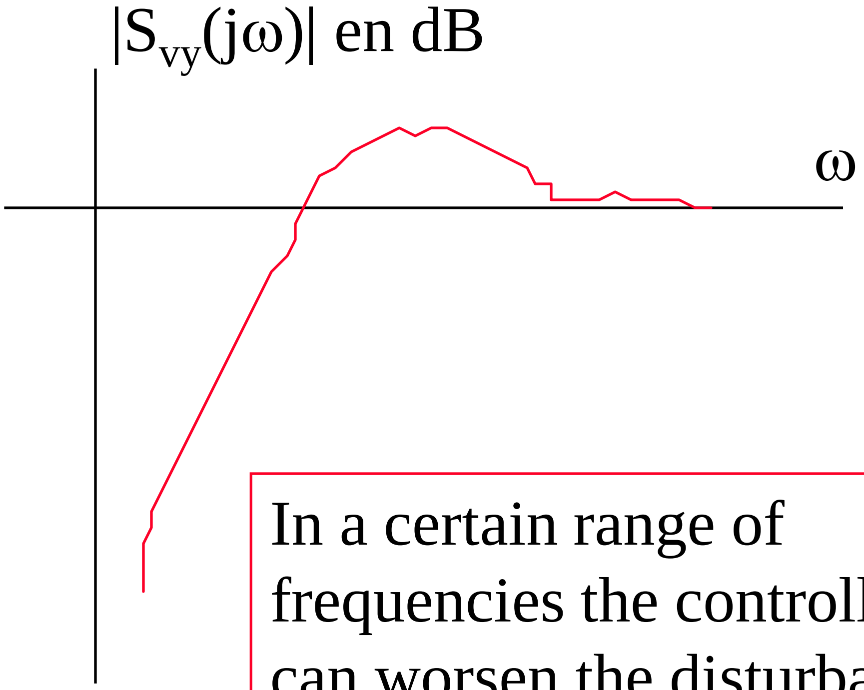
$$S_{vy} = \frac{1}{1+GR} =$$

$$= \frac{1}{1+G(j\omega)R(j\omega)}$$

$\text{if } R \text{ has integral action}$

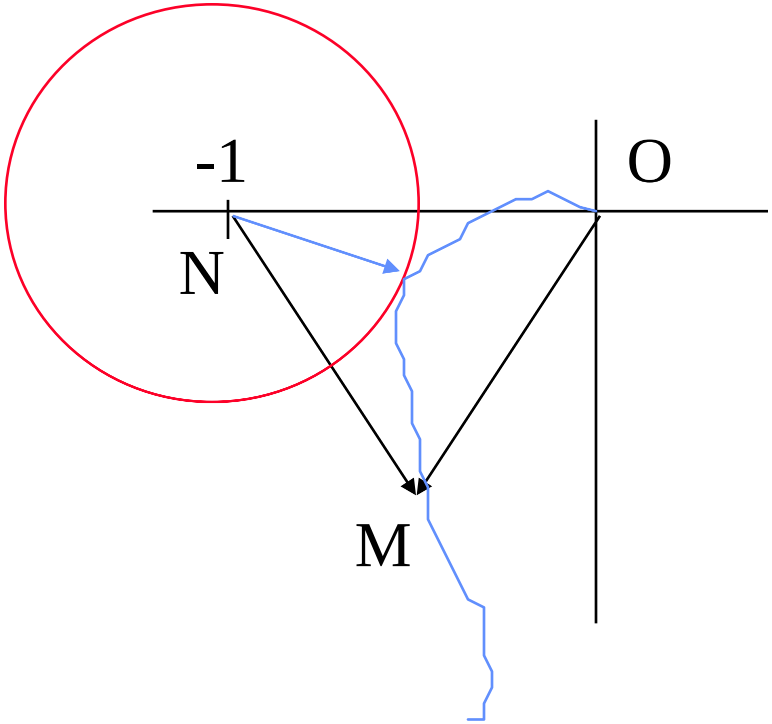
$\text{if} \quad \omega \to 0 \quad \text{then} \quad S_{vy} \to 0$

$\text{if} \quad \omega \to \infty \quad \text{then } S_{vy} \to 1$



In a certain range of frequencies the controller can worsen the disturbance rejection.
It is important to minimize the maximum $|S_{vy}(j\omega)|$

Modulus margin



Nyquist diagram

$$\overline{-1} + \overline{NM} = \overline{OM} = G(j\omega)R(j\omega)$$

$$\left|\overline{NM}\right| = |1 + GR| = \left|S_{vy}^{-1}\right|$$

Modulus margin = min |NM|

$$\min |NM| = \left(\max \left|S_{vy}(j\omega)\right|\right)^{-1}$$

$$= \left\|S_{vy}(j\omega)\right\|_{\infty}^{-1}$$

A larger modulus margin improves the disturbance rejection

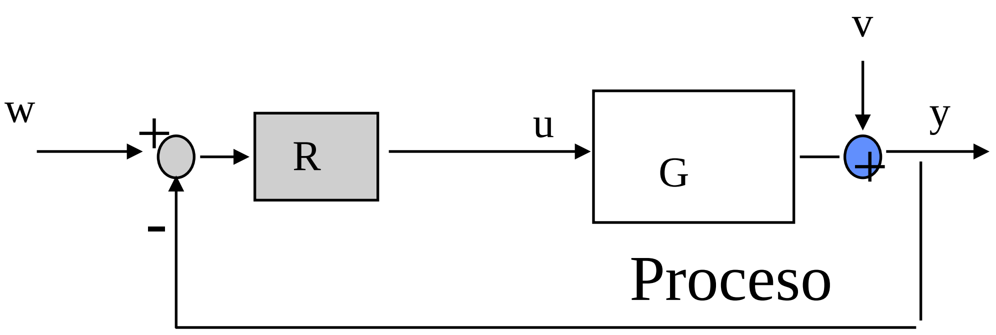
Controller design with the modulus margin

$$\max_{K_p, T_i, T_d} \quad \min_{\omega} \left| 1 + G(j\omega) R(j\omega) \right|$$

$$R(j\omega) = K_p \left[1 + \frac{1}{T_i j\omega} + \frac{T_d j\omega}{1 + 0.1 T_d j\omega} \right]$$

Max min optimization oriented to disturbance rejection

Robustness

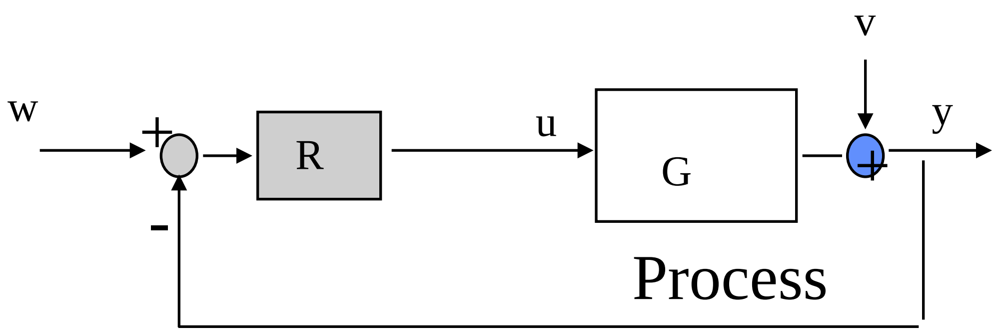


$$y = \frac{GR}{1+GR} w + \frac{1}{1+GR} v$$

How the closed loop dynamics changes when the process parameters varies?

Sensibility $\dfrac{\frac{\partial T}{T}}{\frac{\partial G}{G}} = \dfrac{G}{T}\dfrac{\partial T}{\partial G}$ $\qquad T = \dfrac{GR}{1+GR}$

Robust design



$$y = \frac{GR}{1+GR} w + \frac{1}{1+GR} v$$
$$= Tw + Sv$$

$$\frac{G}{T}\frac{\partial}{\partial G}\left[\frac{GR}{1+GR}\right] = \frac{G(1+GR)}{GR}\frac{(1+GR)R - GRR}{(1+GR)^2} = \frac{1+GR}{R}\frac{R}{(1+GR)^2} = \frac{1}{(1+GR)} = S_{vy}$$

$$\frac{G}{S}\frac{\partial}{\partial G}\left[\frac{1}{1+GR}\right] = \frac{G(1+GR)}{1}\frac{(-R)}{(1+GR)^2} = \frac{-GR}{(1+GR)} = -T$$

Sensibility function S_{vy} = sensibility with respect to changes in G

It is important to minimize the errors in the range of frequencies where the sensibility respect to w or v is higher

Automatic tuning methods

Most of the commercial controllers incorporate some methods for automatic tuning (most of them autotuning)
Only in a few cases we find real adaptive control

Autotuning: The tuning procedure starts under operator demand

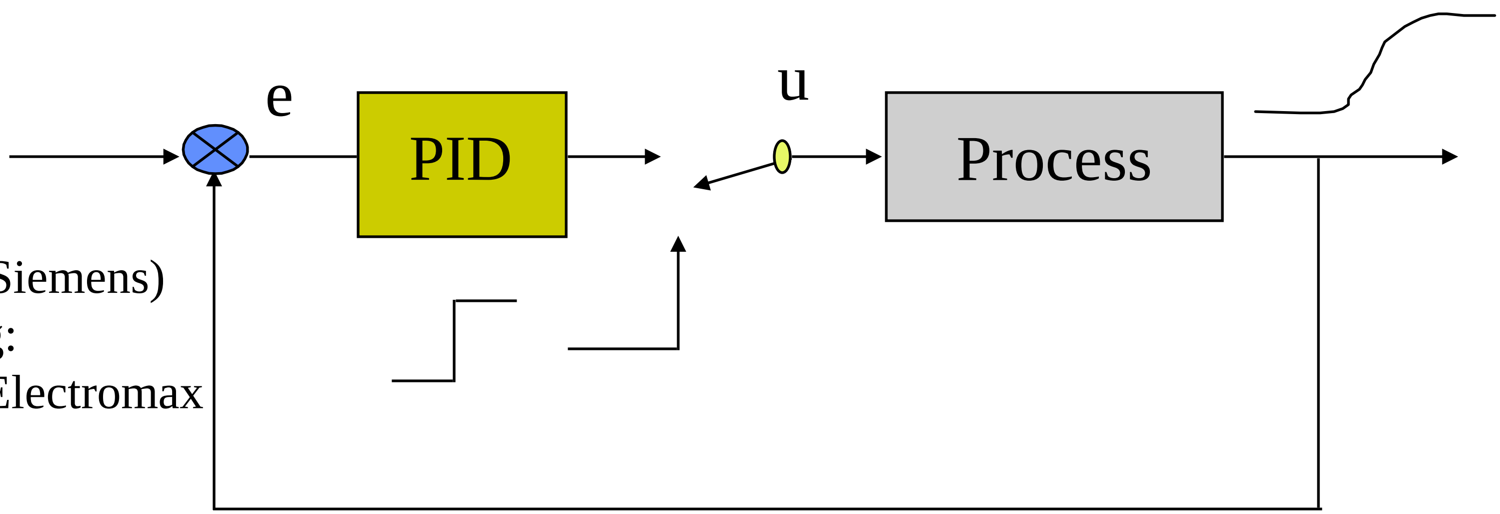
Adaptive control: The automatic tuner continuously identifies the process dynamics and readjust the controller parameters if there is any change

Automatic tuning methods

- Step response
- Relay’s method
- Closed loop response identification (Exact)
- Adaptive control
- Gain scheduling

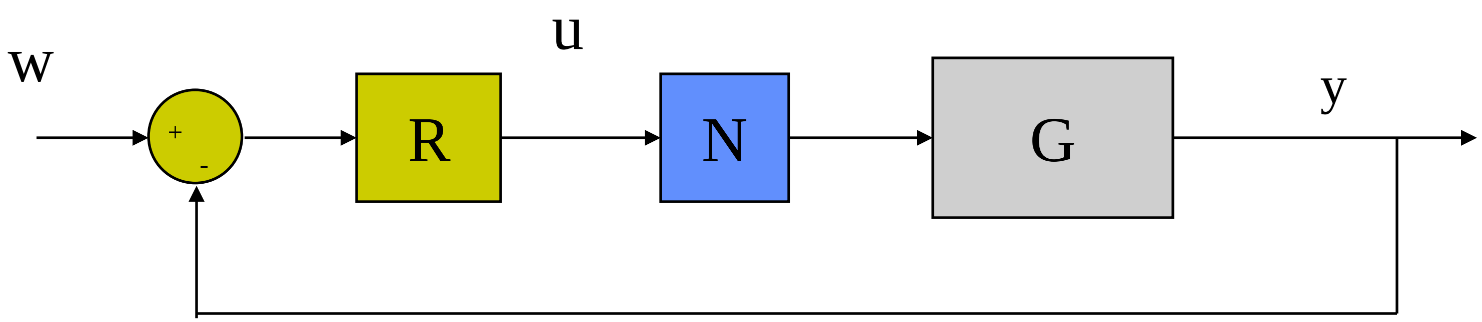
Step response

When the autotuning function is activated, the controller is turned into manual mode, then, it generates a step in order to identify a first order plus delay model from which the controller parameters are obtained using tuning tables.



SIPART (Siemens)
Pre-tuning:
EXACT, Electromax

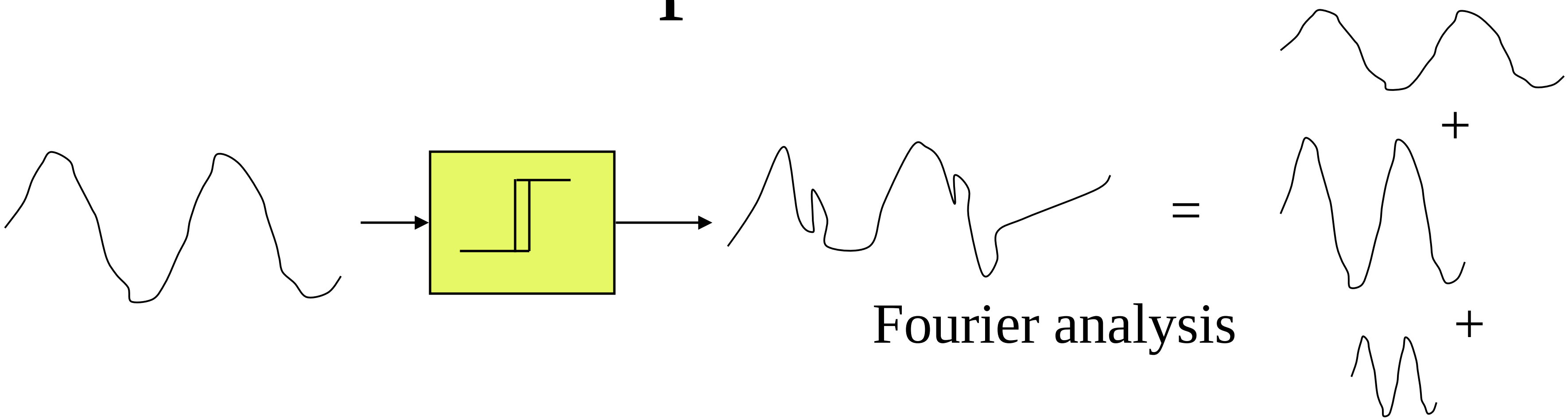
System analysis with a non linear block



N: descriptive function: linear approximation of the non-linear element: relay, saturation, hysteresis, etc.

Characteristic equation: $1+GRN = 0$

Descriptive function



How to compute the frequency response of a non-linear element:

1 Feed the system with a sinusoidal signal of frequency ω

2 Compute the first harmonic of the system output

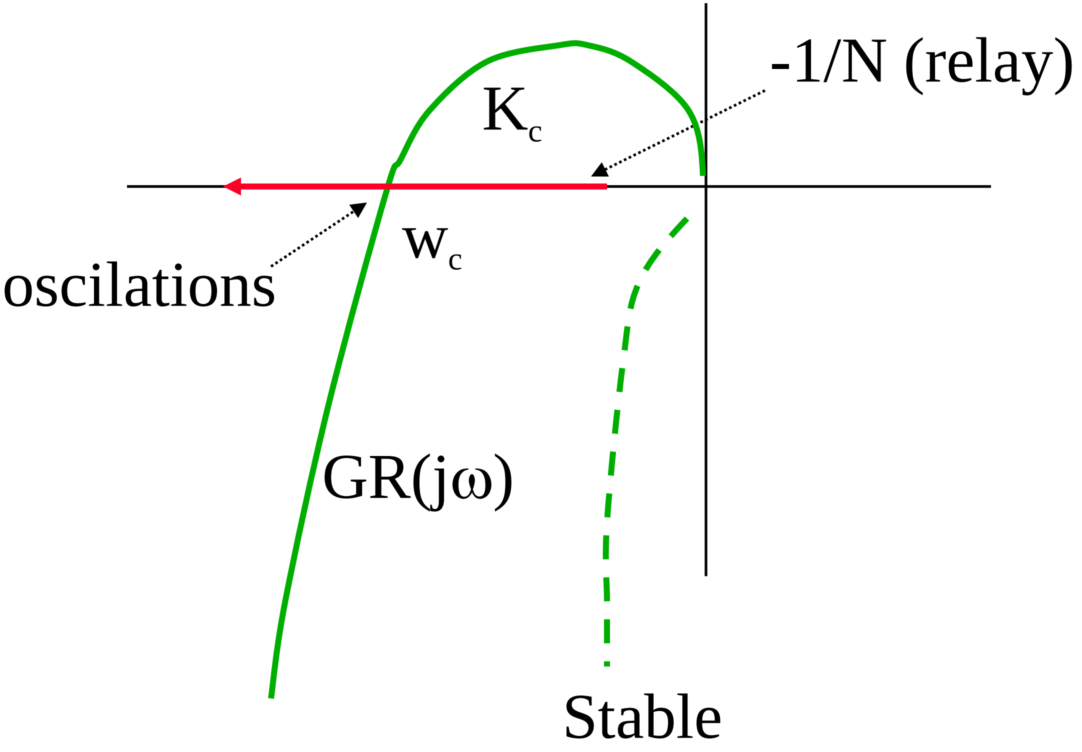
3 compute the gain and phase shift with respect to the first harmonic

System analysis with a non linear block

1+GRN = 0

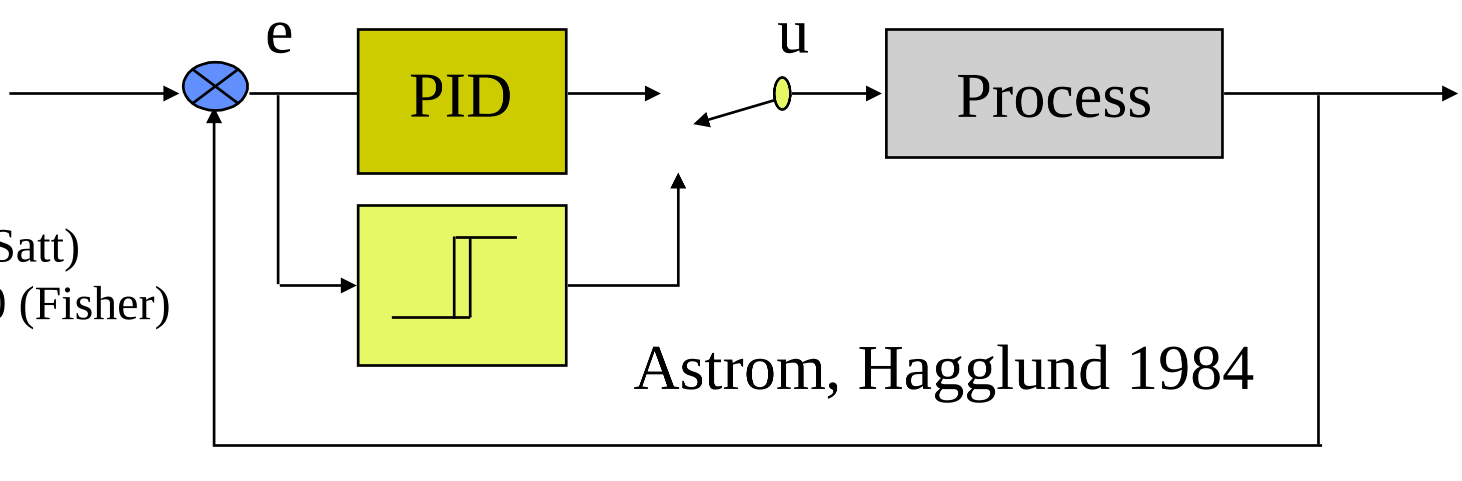
GR = -1/N

In the Nyquist diagram analysis, the -1/N plot plays the same role as the -1 point in linear systems



The relay method

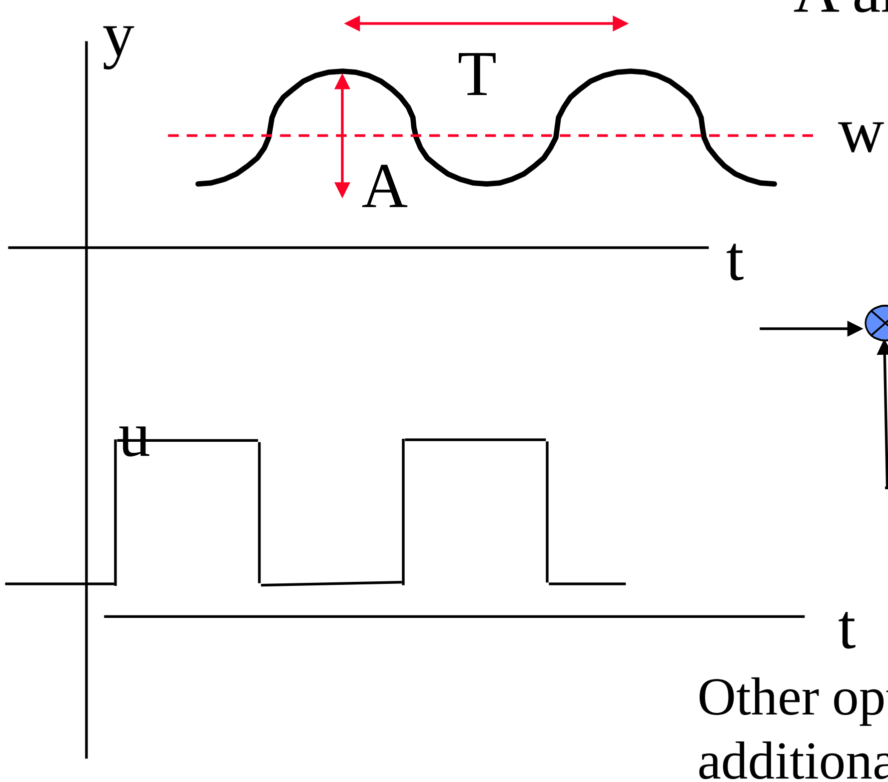
When the autotuning function is activated, there is a switching from the PID to a relay controller that creates controlled oscillations in the process which are used to identify some of its dynamic characteristics

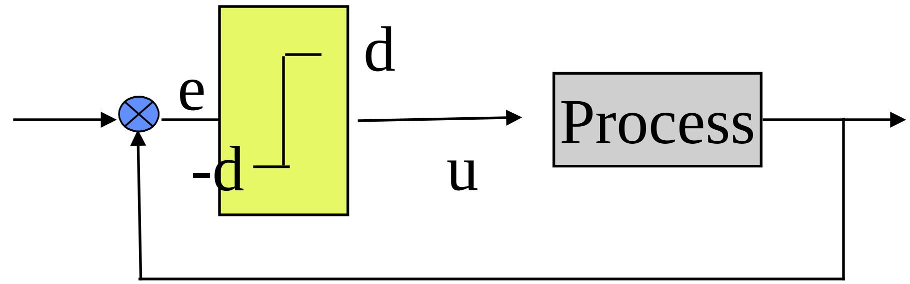


ECA40 (Satt)
DPR9000 (Fisher)

The relay method

T oscillation period
A amplitude of the first harmonic

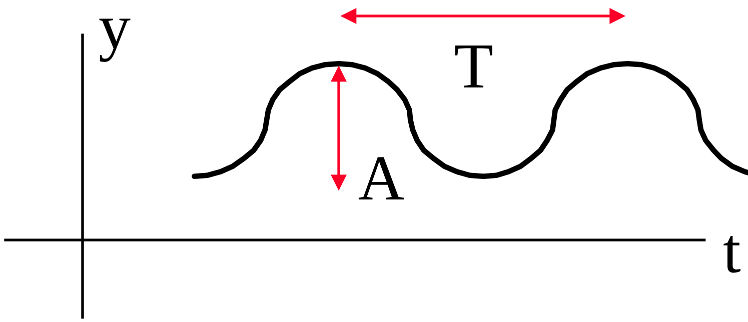


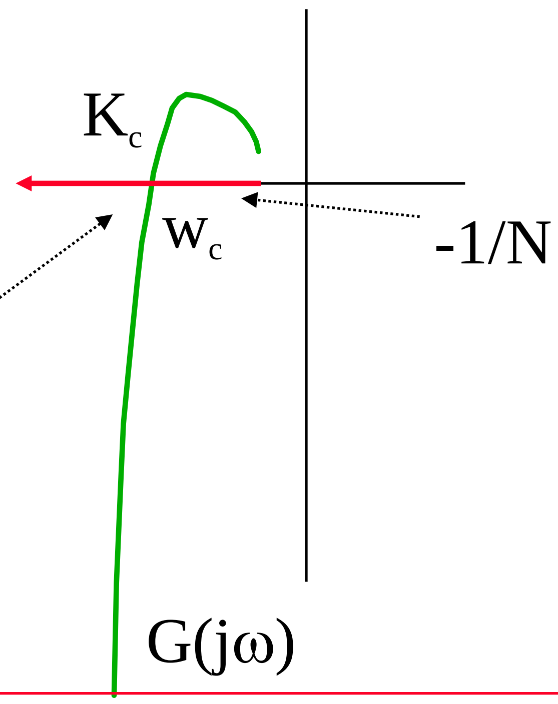


Other options: relay with hysteresis or additional loops are added in order to force the generation of oscillations

The relay method

1+GN=0   N relay descriptive function





$$K_c = \frac{4d}{\pi A}$$

$$\omega_c = \frac{2\pi}{T}$$

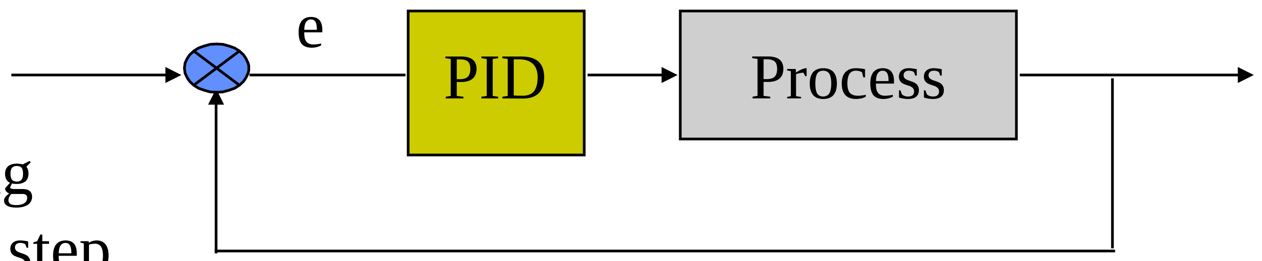
The oscillations allow to identify a point of the Nyquist diagram

Then the controller is tuning with the phase margin method

The Exact method

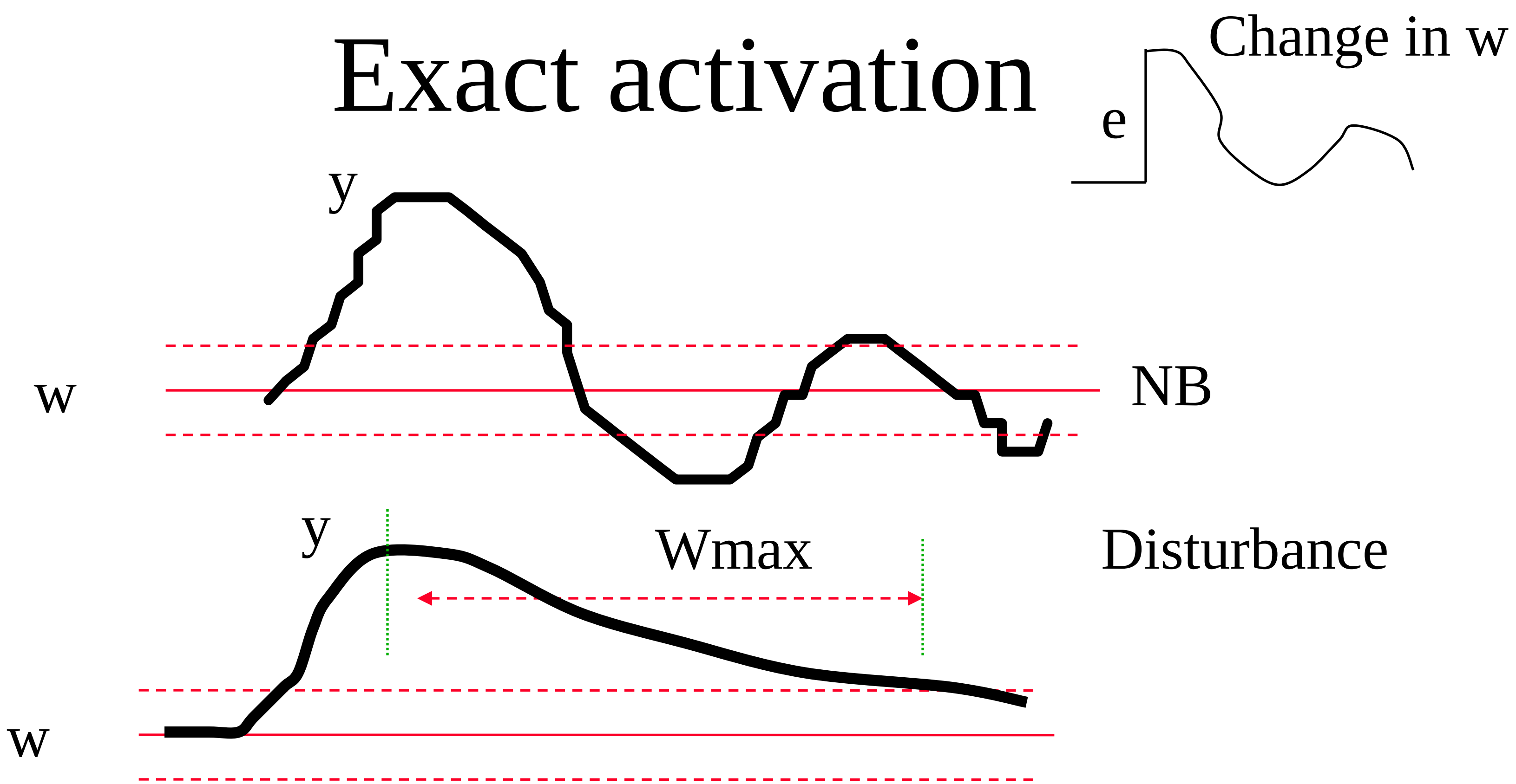
EXact Adaptive Controller Tuning (Foxboro)

- ✓Continuous closed loop tuning
- ✓If the error exceeds a range, then a process identification procedure based on pattern recognition is started
- ✓The controller computes the new tuning in real time using modified Ziegler-Nichols tables plus some rules
- ✓The desired dynamics is specified in terms of overshoot and damping



Pretuning with the step method

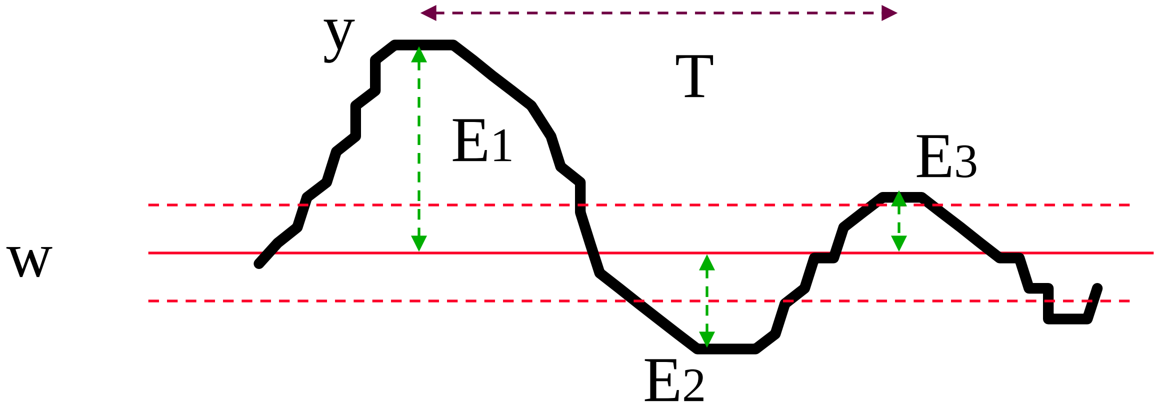
Exact activation



The procedure is activated automatically if the error is outside the error band NB and the second pick appears before Wmax sg. after the first one

If no second pick appears before Wmax, the process is considered a overdamped one

Exact



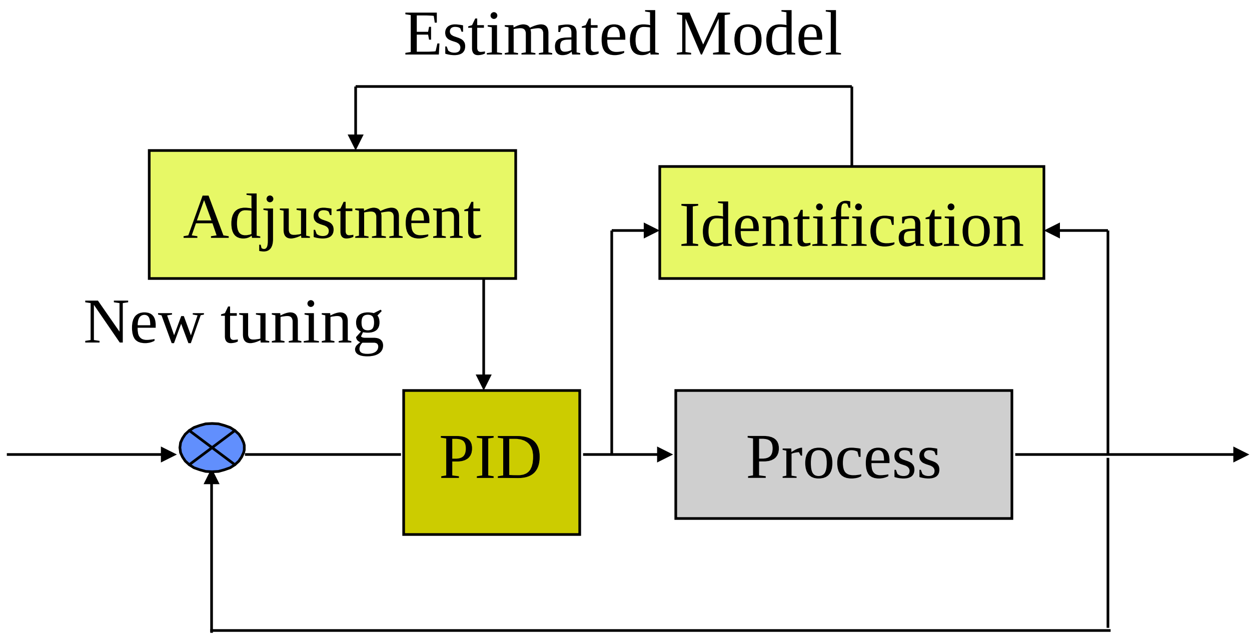
When the tuning procedure is activated, the exact measure thee picks E_1, E_2, E_3 as well as its times of ocurrence and uses them to estimate a process model with:

$$\text{damping} = \frac{E_3 - E_2}{E_1 - E_2} \quad \text{overshoot} = \frac{E_2}{E_1}$$

Or an overdamped process model

Then modified Ziegler-Nichols tuning rules are applied

Adaptive Control



External excitation for identification or conditional activation
The adjustment is activated with a larger temporal scale
Controller supervision / Stability

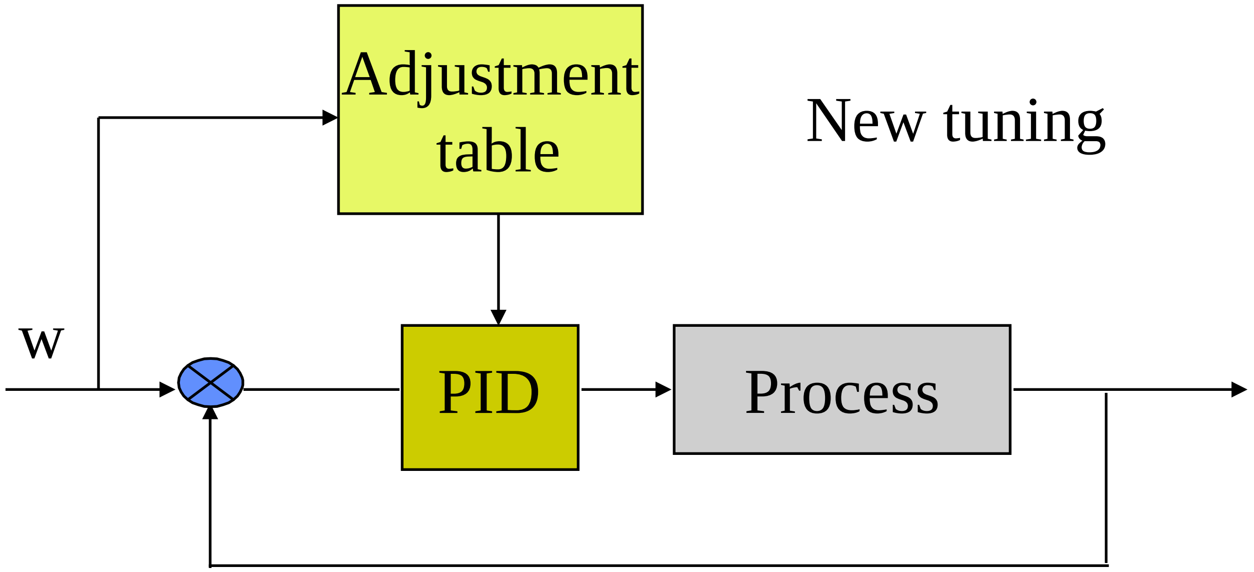
Adaptive PID

Electromax
Firstloop (First Control)
Identification of a two pole model
PID tuning by pole assignment

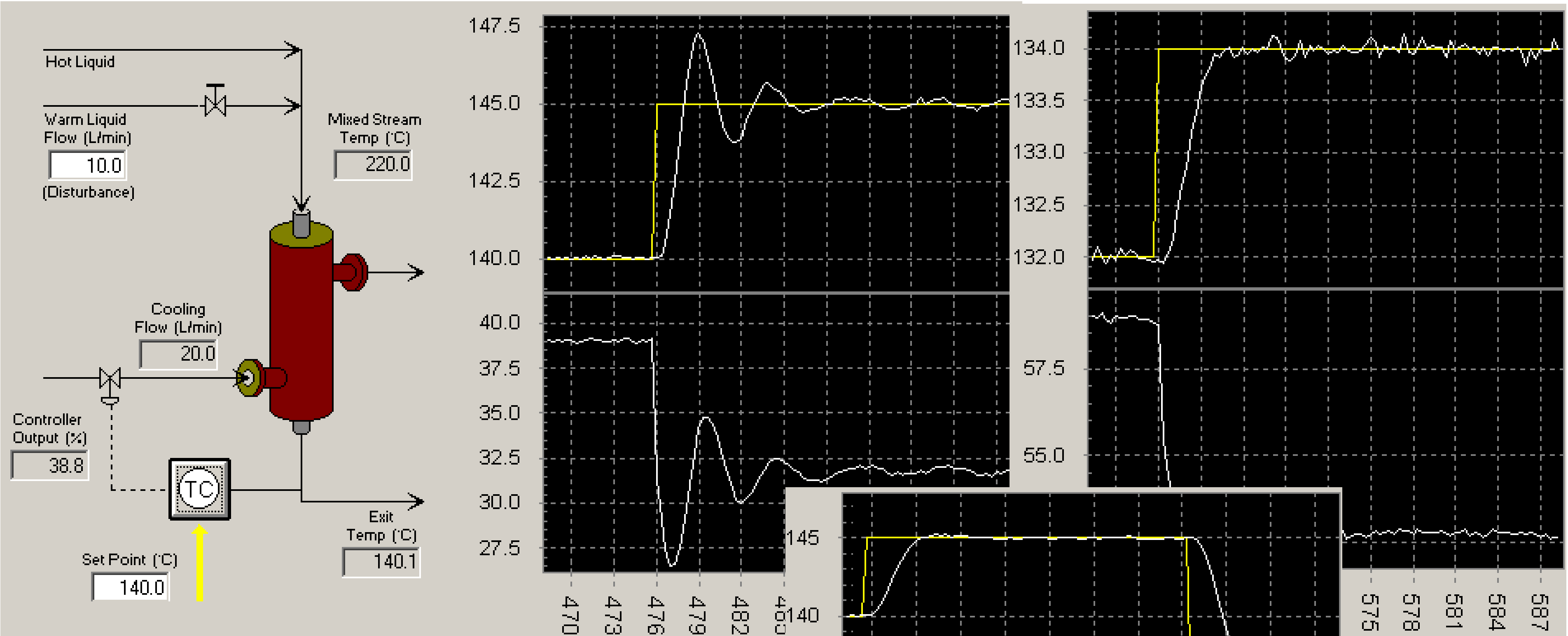
Novatune (ABB)
Recursive identification
Tuning by minimum variance control

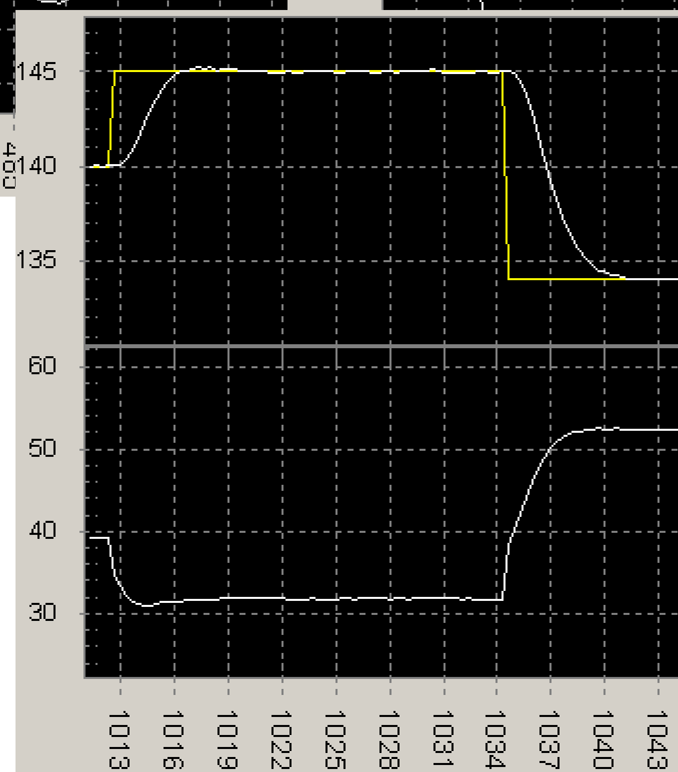
Wittenmark (1979) Cameron-Seborg (1983)
Radke-Isermann (1987)   Vega/Prada (1987)

Gain scheduling

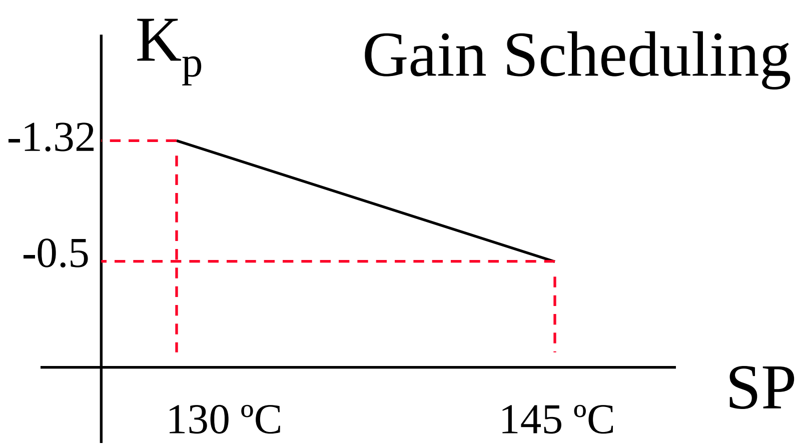


The controller parameters are adjusted using a pre-computed table function of some operating condition: e.g. the set point value

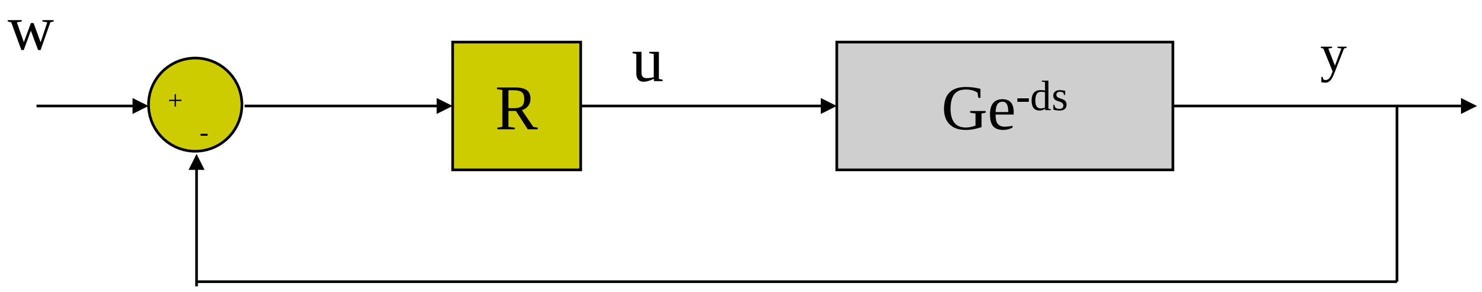




Gain Scheduling



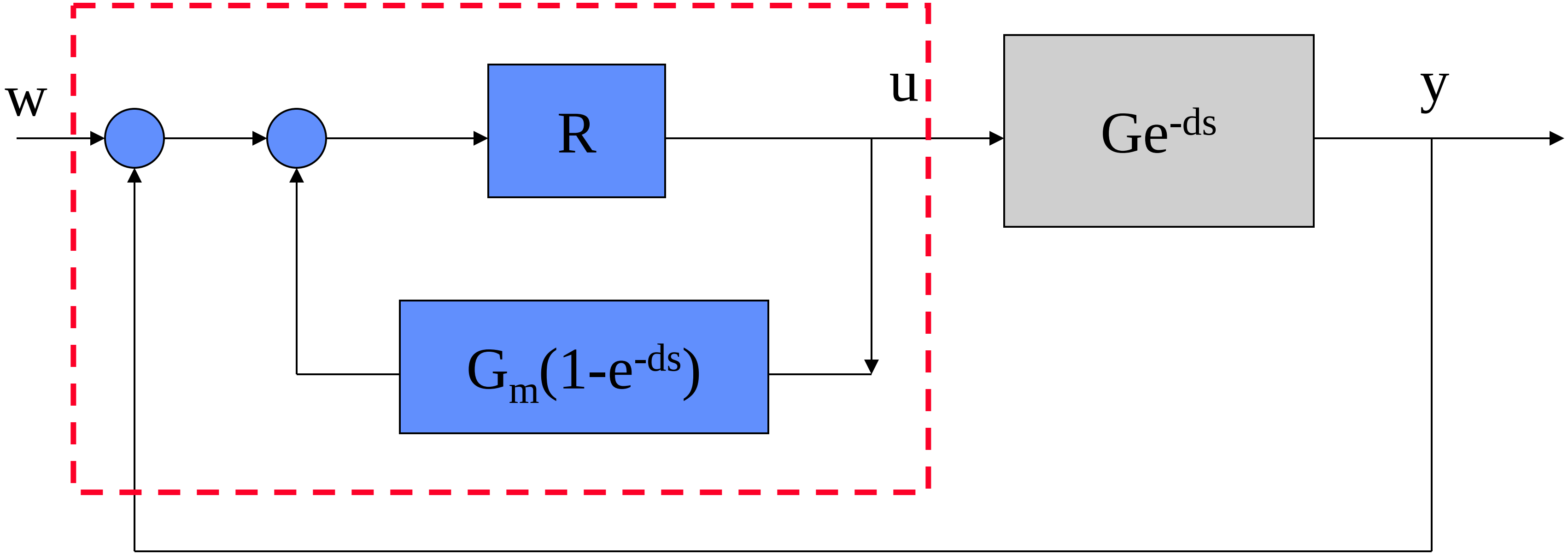
Systems with delay



If the delay is higher than the process time constant, the system is difficult to tune.

The Smith predictor is a controller that improves the time response of this type of processes. It needs to know the model Ge^{-ds}

Delays: Smith Predictor



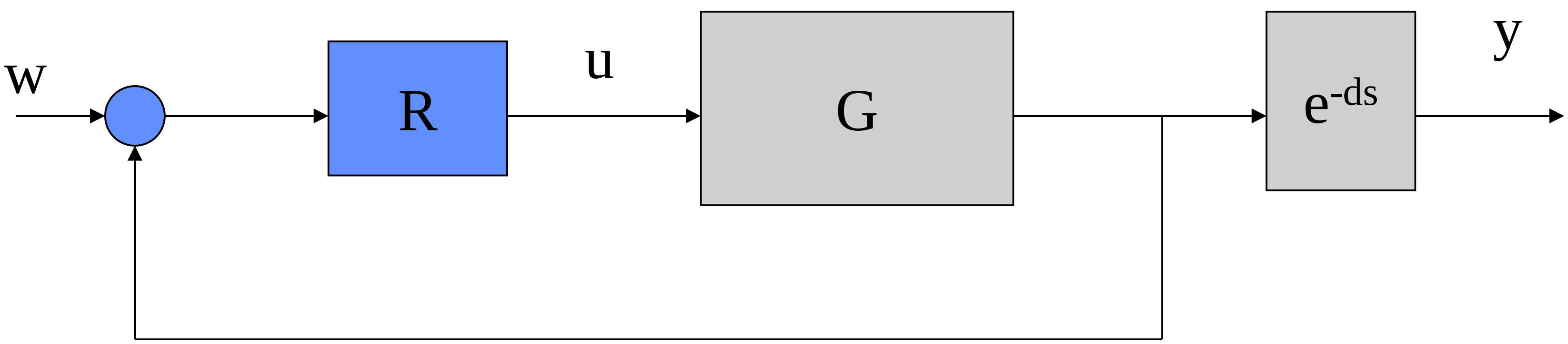
$$y = Ge^{-ds}u = Ge^{-ds}R\left[w - y - G_m(1-e^{-ds})u\right] =$$

$$= Ge^{-ds}R\left[w - Ge^{-ds}u - G_m(1-e^{-ds})u\right]$$

$$\text{si } G = G_m \qquad y = Ge^{-ds}R\left[w - Gu\right]$$

Smith Predictor

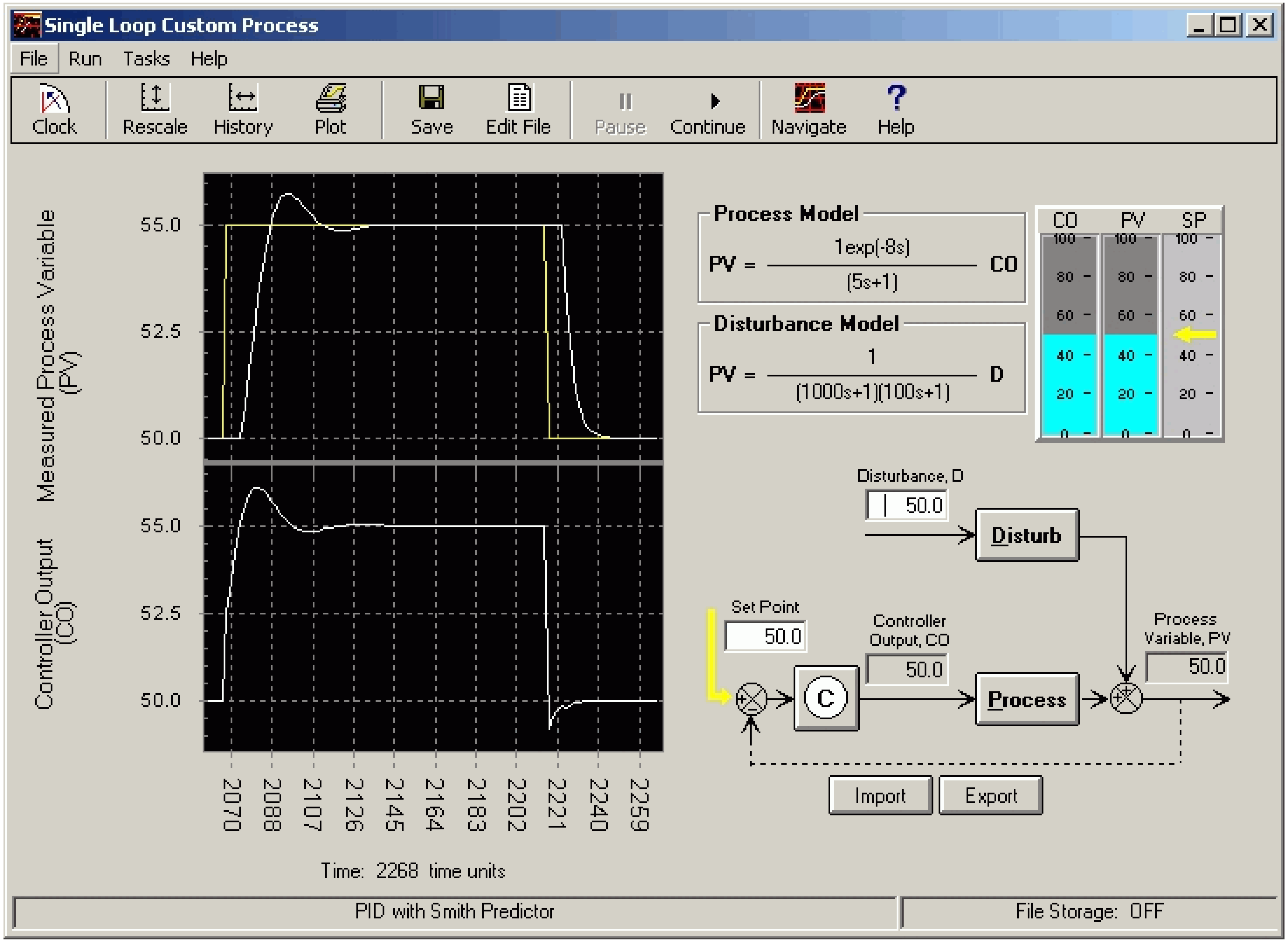
$$y = e^{-ds} GR\left[w - Gu\right]$$



Equivalent diagram

R can be tuned as if there were no delay

Smith Predictor



$K_p = 0.4$

$T_i = 5$

with Smith predictor

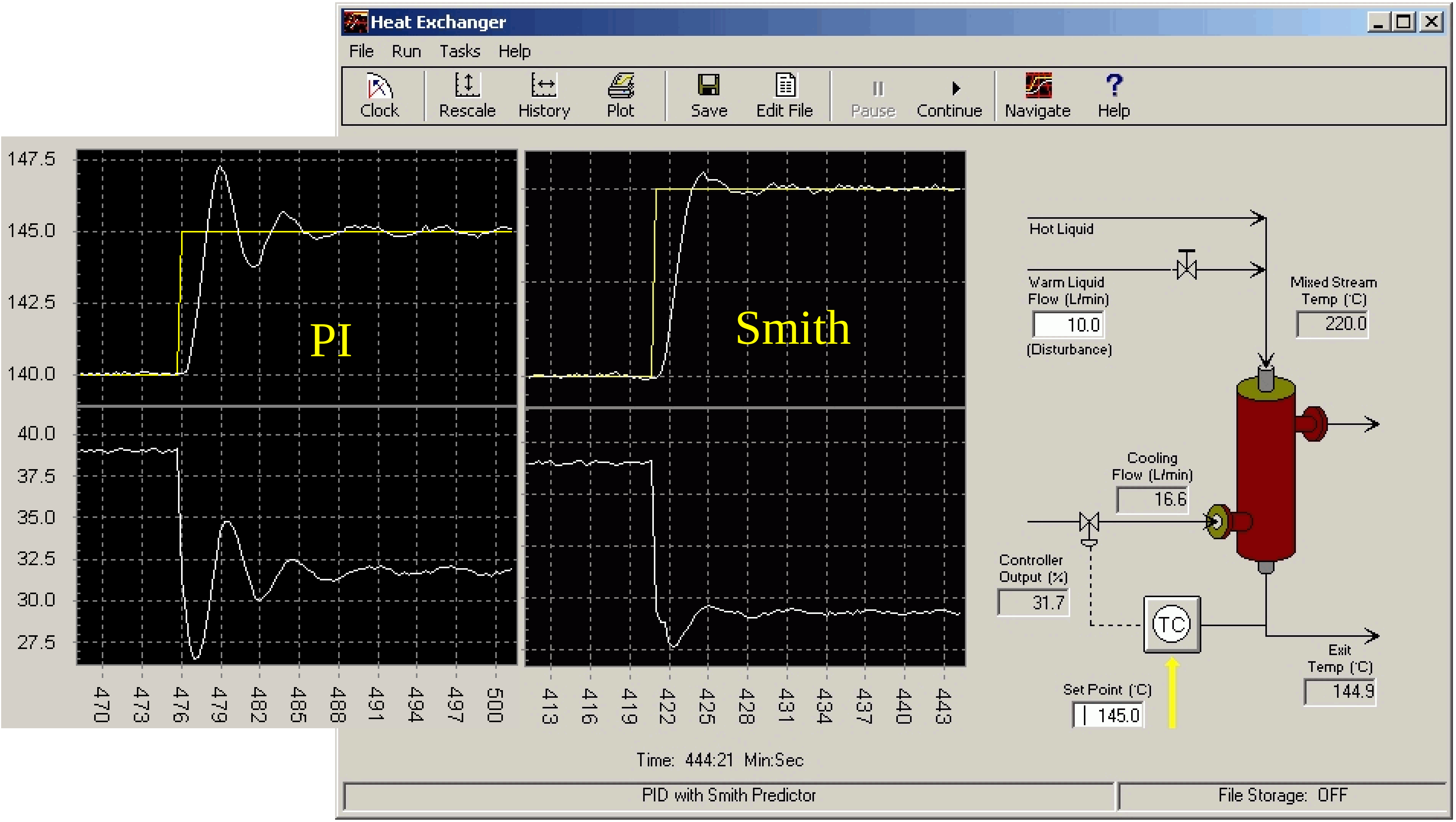
$K_p = 1.2$

$T_i = 5$

$$\frac{-0.46e^{-0.87s}}{0.96s+1}$$

$$K_p = -1.32,\ T_i = 0.96$$

Smith Predictor



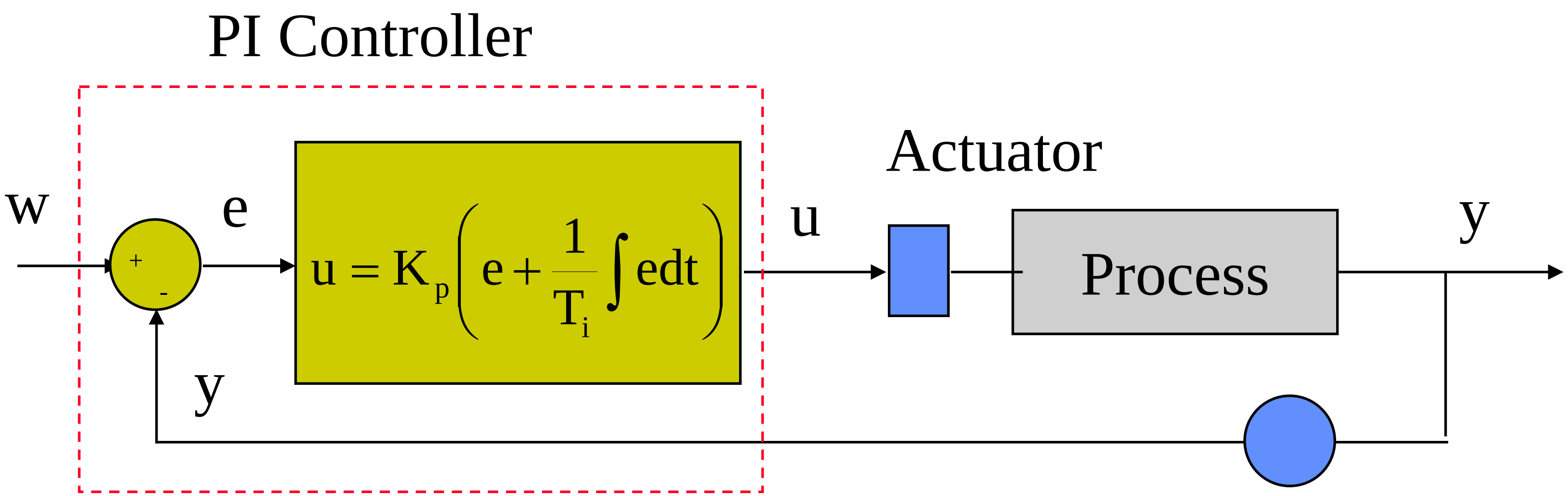
The PID controller

$$e(t) = w(t) - y(t)$$

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int e(\tau)d\tau + T_d\frac{de}{dt}\right)$$

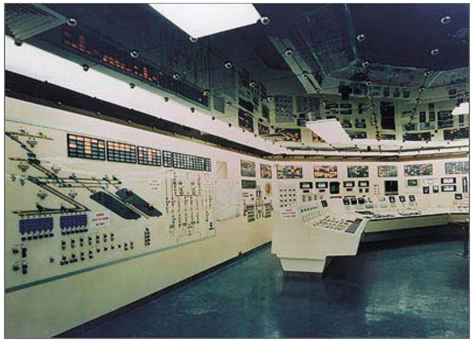
- **Signal based controller**, no explicit process knowledge is incorporated
- 3 tuning parameters K_p, T_i, T_d
- Many different implementations

Implementation



Transmitter

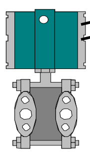
Loop controller



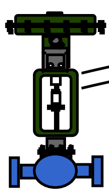
Panel mounting, PLC,…

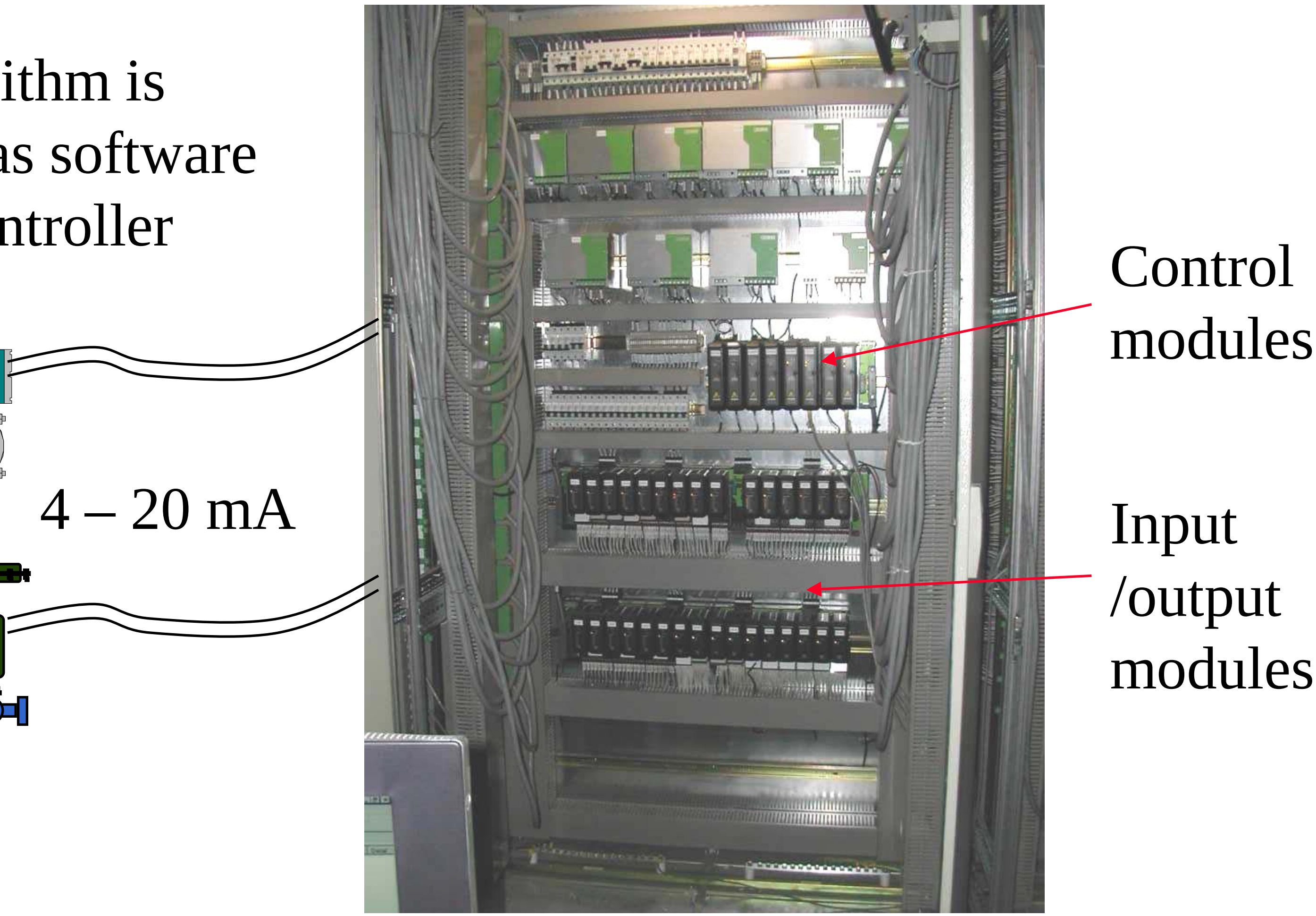
Implementation

The PID algorithm is implemented as software in the DCS controller modules



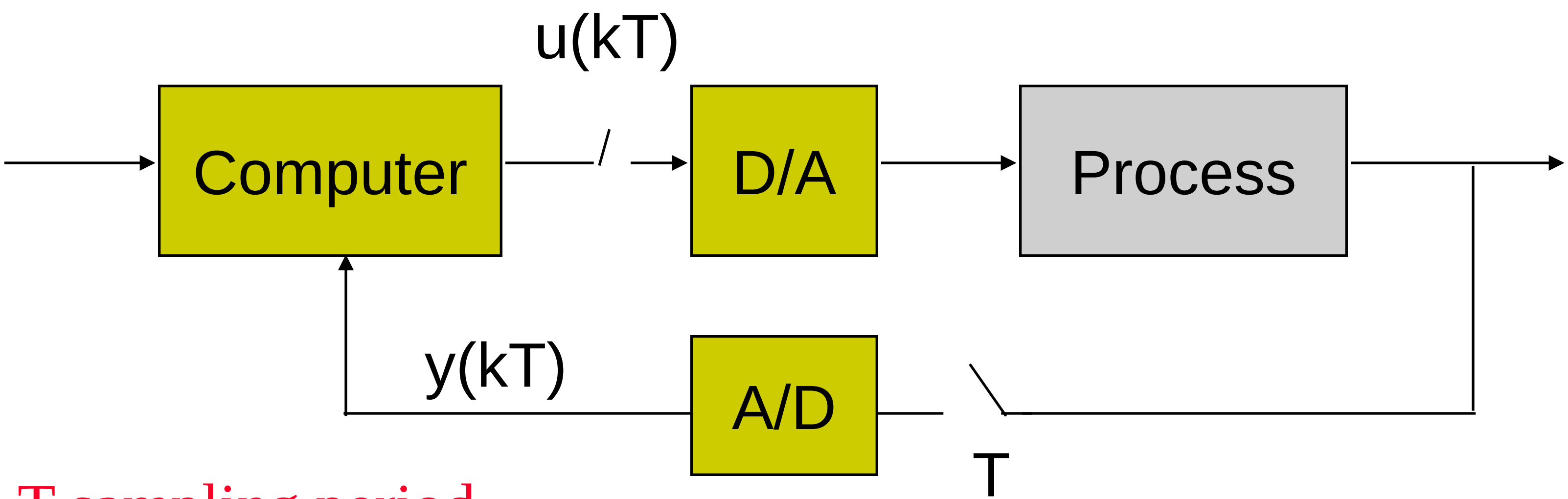
4 – 20 mA





Control wardrobe

Digital Control



T sampling period

T should be chosen according to the process dynamics, as well as considering numerical problems in integration and differentiation.

Integration: $T \cong 0.1 \ldots 0.3\, T_i$ Differenciation. $T \cong 0.2 \ldots 0.6\, T_d / N$

Accuracy in the measurement depends also on the D/A converter

Higher precision in the internal computations than the one of D/A

Discretizing PID controllers

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de}{dt} \right)$$

Rectangular approximation

$$u(t) \approx K_p\left(e(t) + \frac{1}{T_i}\sum_{i=1}^{t} e(iT)T + T_d\frac{e(t)-e(t-T)}{T} \right)$$

$$u(t-T) \approx K_p\left(e(t-T) + \frac{1}{T_i}\sum_{i=1}^{t-T} e(iT)T + T_d\frac{e(t-T)-e(t-2T)}{T} \right)$$

$$u(t) - u(t-T) = K_p\left(e(t) - e(t-T) + \frac{T}{T_i}e(t) + T_d\frac{e(t)-2e(t-T)+e(t-2T)}{T} \right)$$

$$u(t) = u(t-T) + g_0 e(t) + g_1 e(t-T) + g_2 e(t-2T)$$

$$g_0 = K_p\left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right) \quad g_1 = K_p\left(-1 - \frac{2T_d}{T}\right) \quad g_2 = K_p\frac{T_d}{T}$$

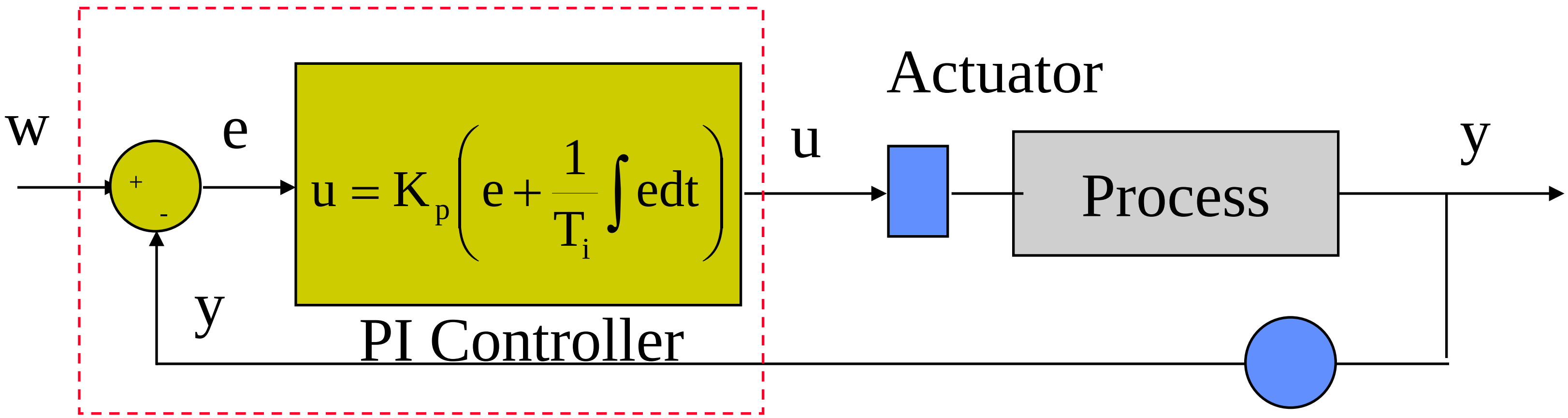
Digital PID

$$e(t) = w(t) - y(t)$$

$$u(t) = u(t-1) + g_0 e(t) + g_1 e(t-1) + g_2 e(t-2)$$

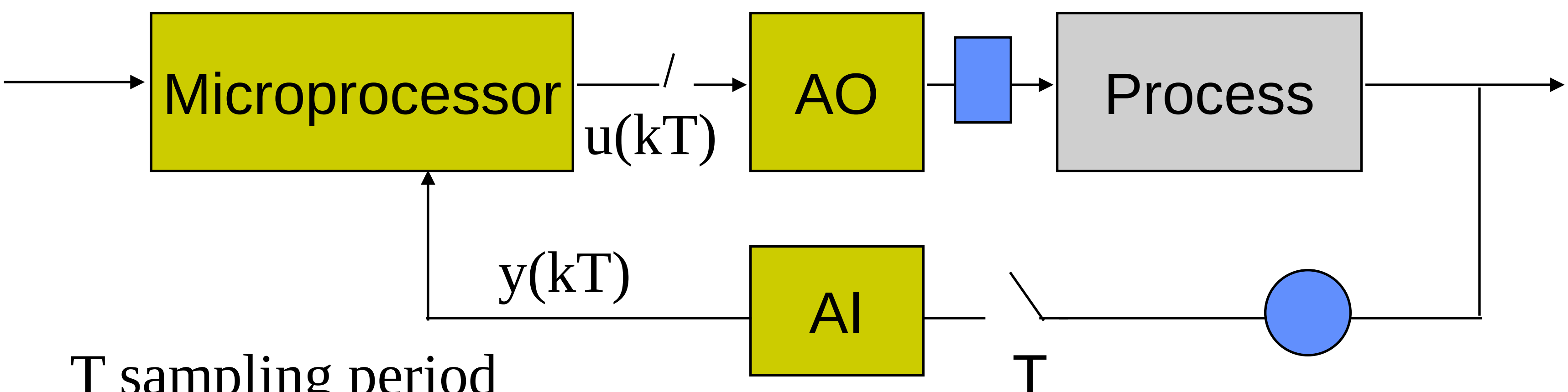
- Many formulas for discretization
- Microprocesor based controller with many auxiliary functions
- Sampling time T very often fixed in the range 100...200 msg

Implementation



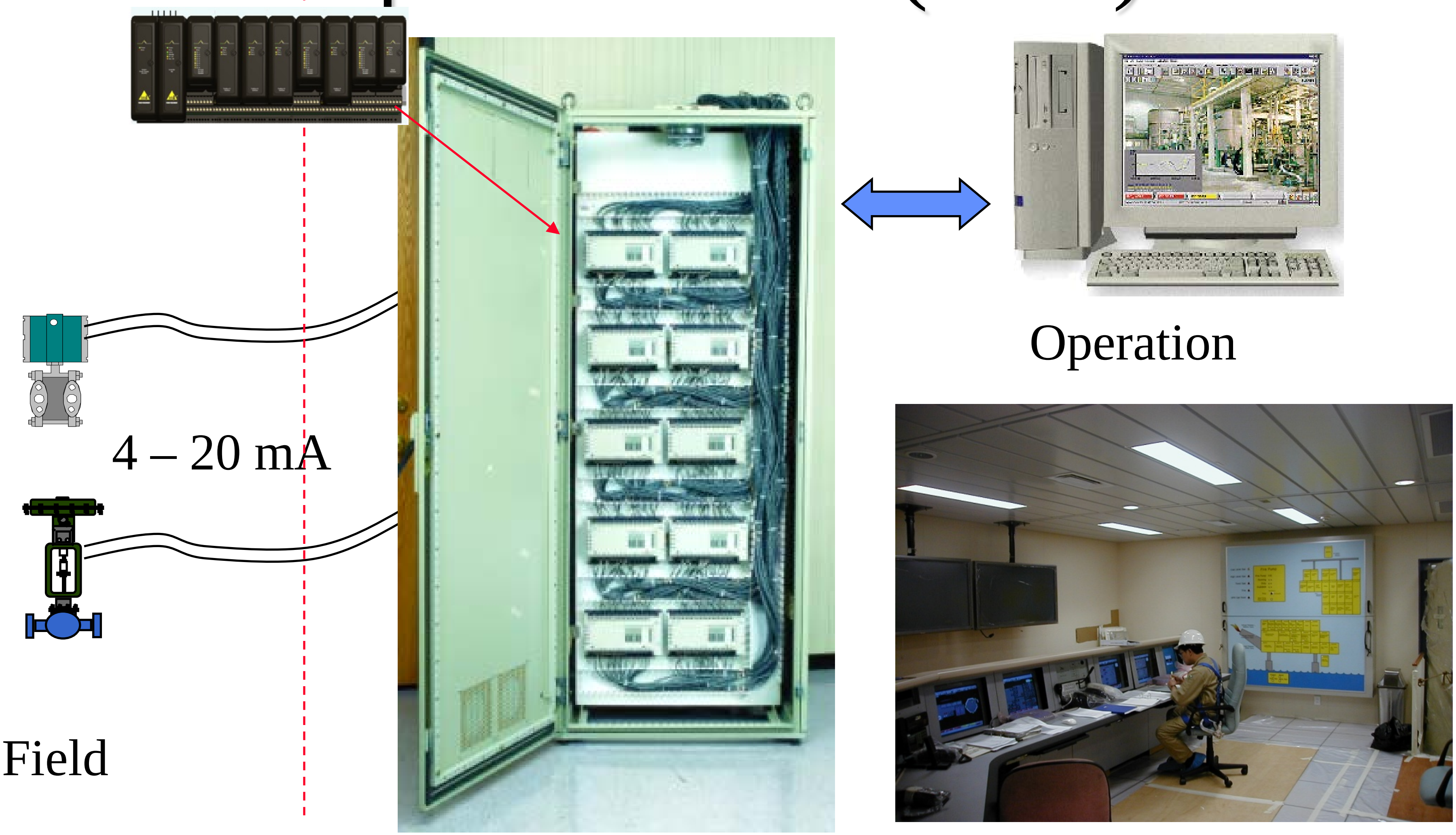
$$u(kT) = u((k-1)T) + g_0 e(kT) + g_1 e((k-1)T)$$

Discrete formulation

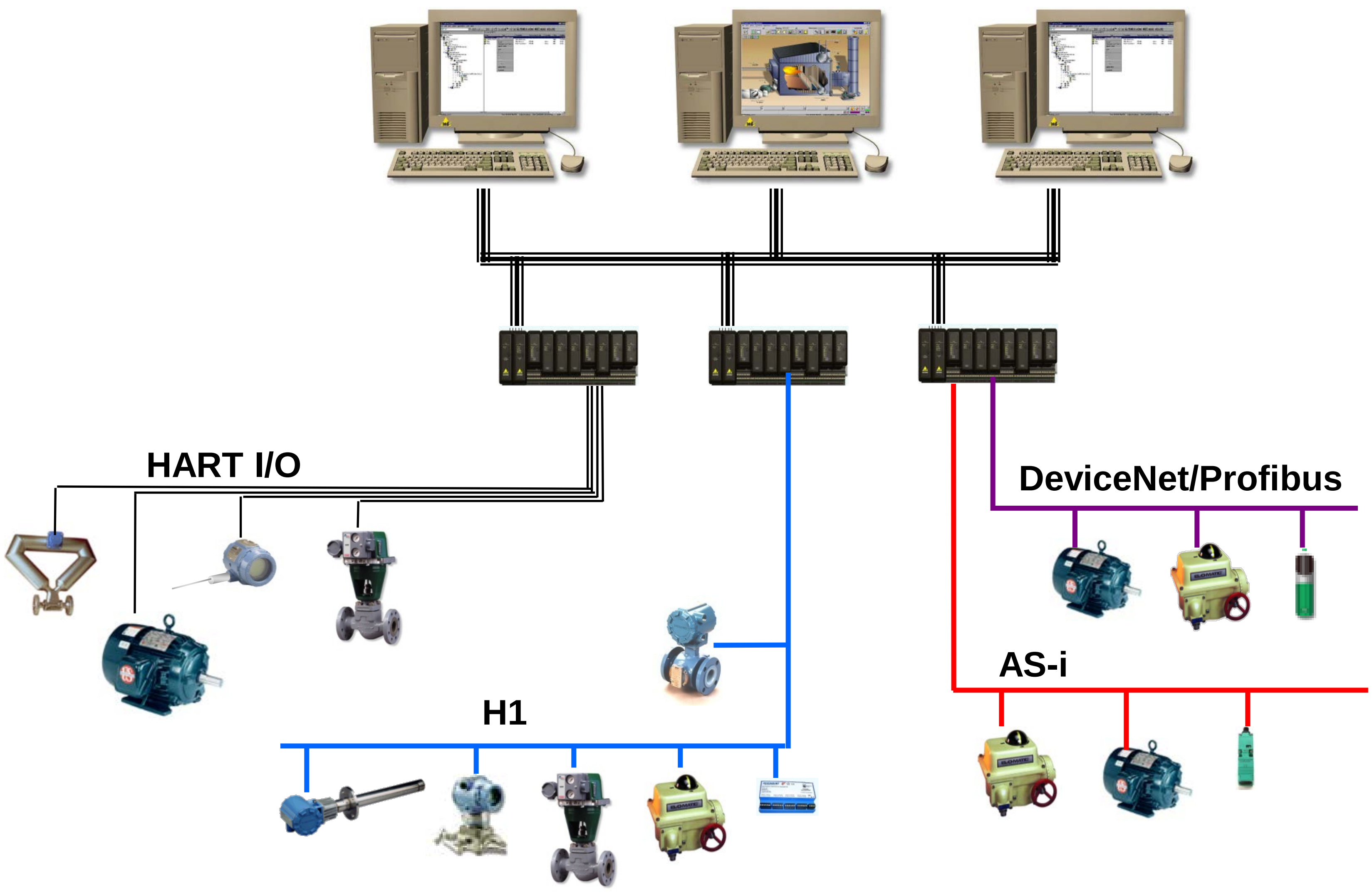


T sampling period

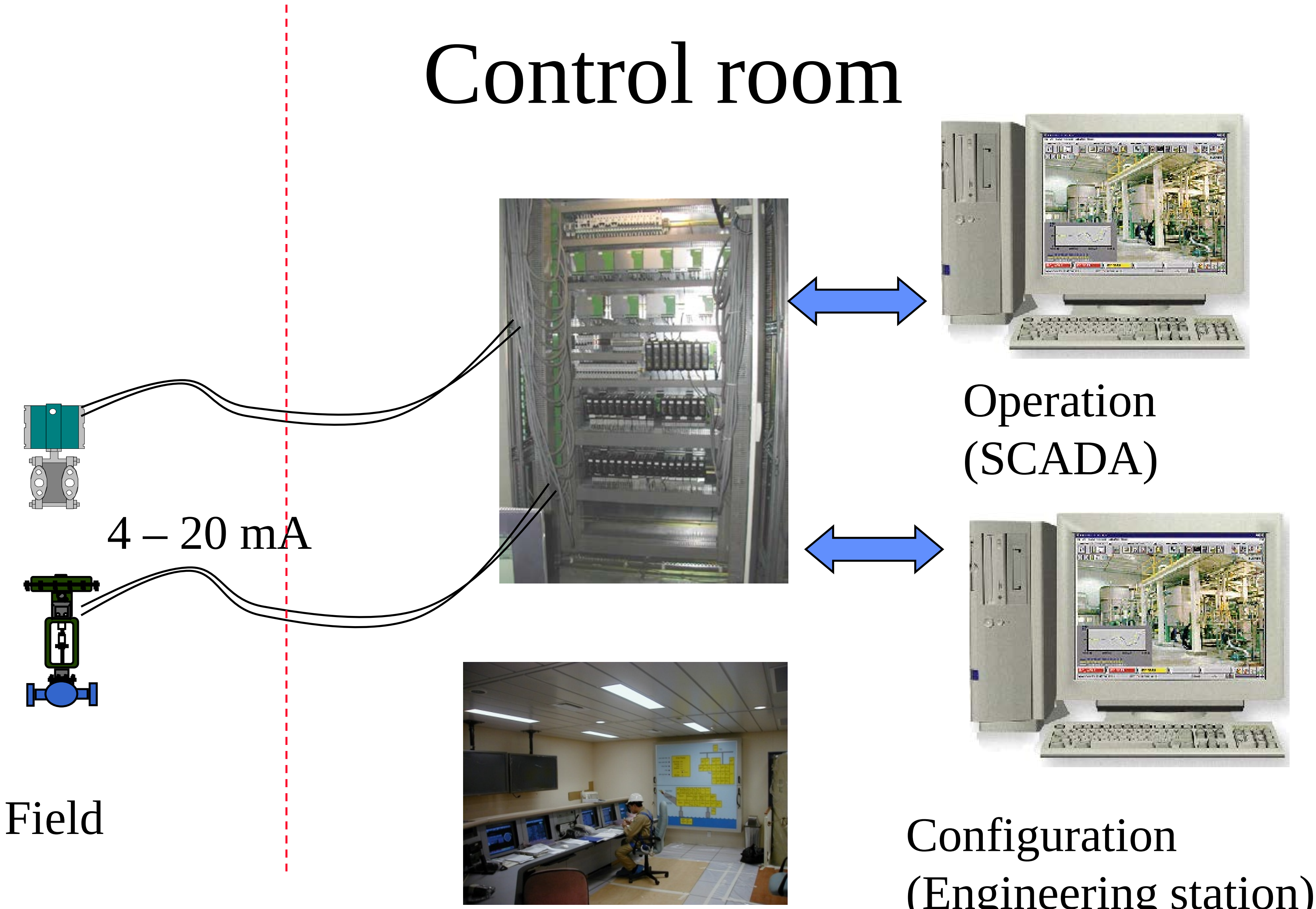
Implementation (DCS)



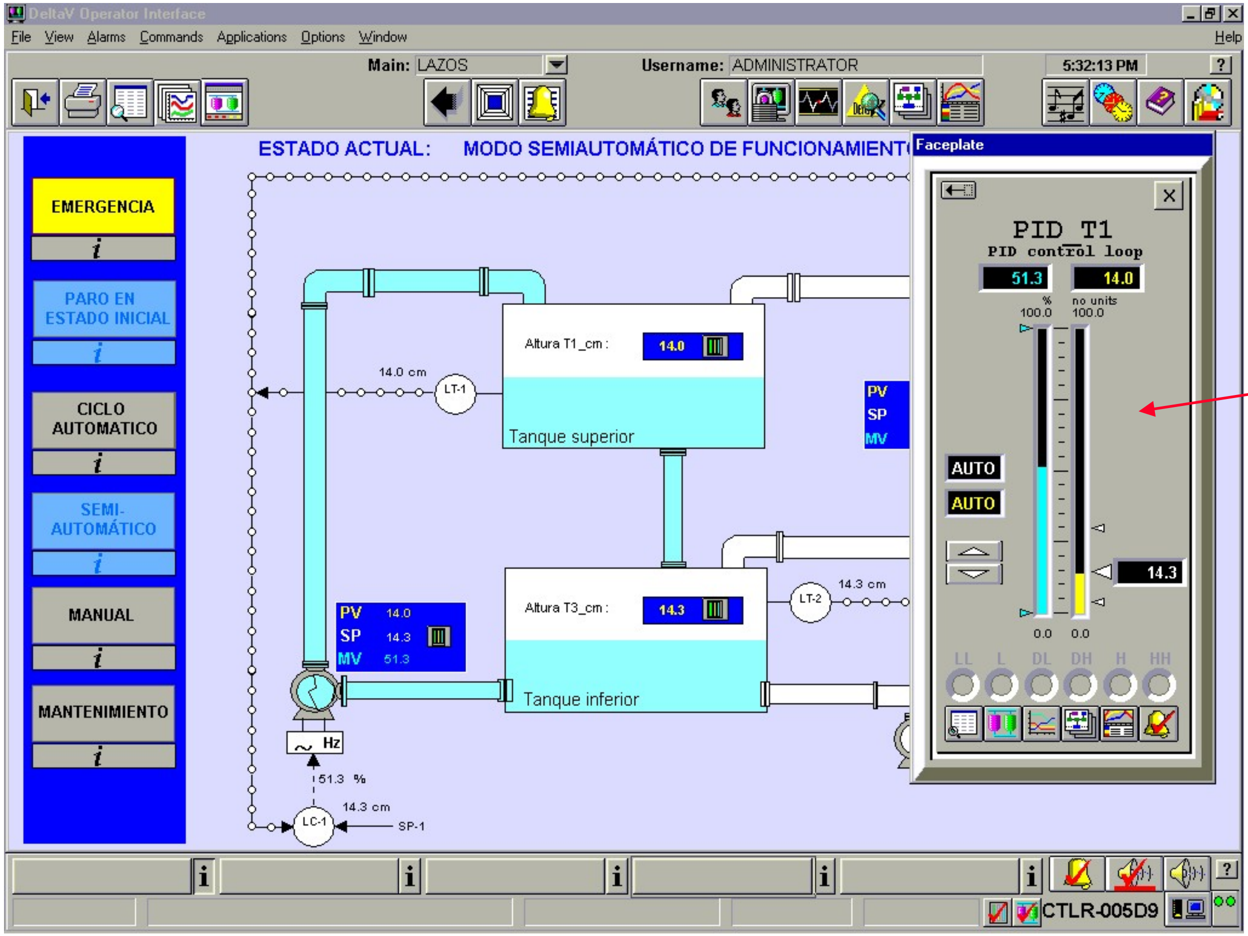
Architectures



Control room



Operation

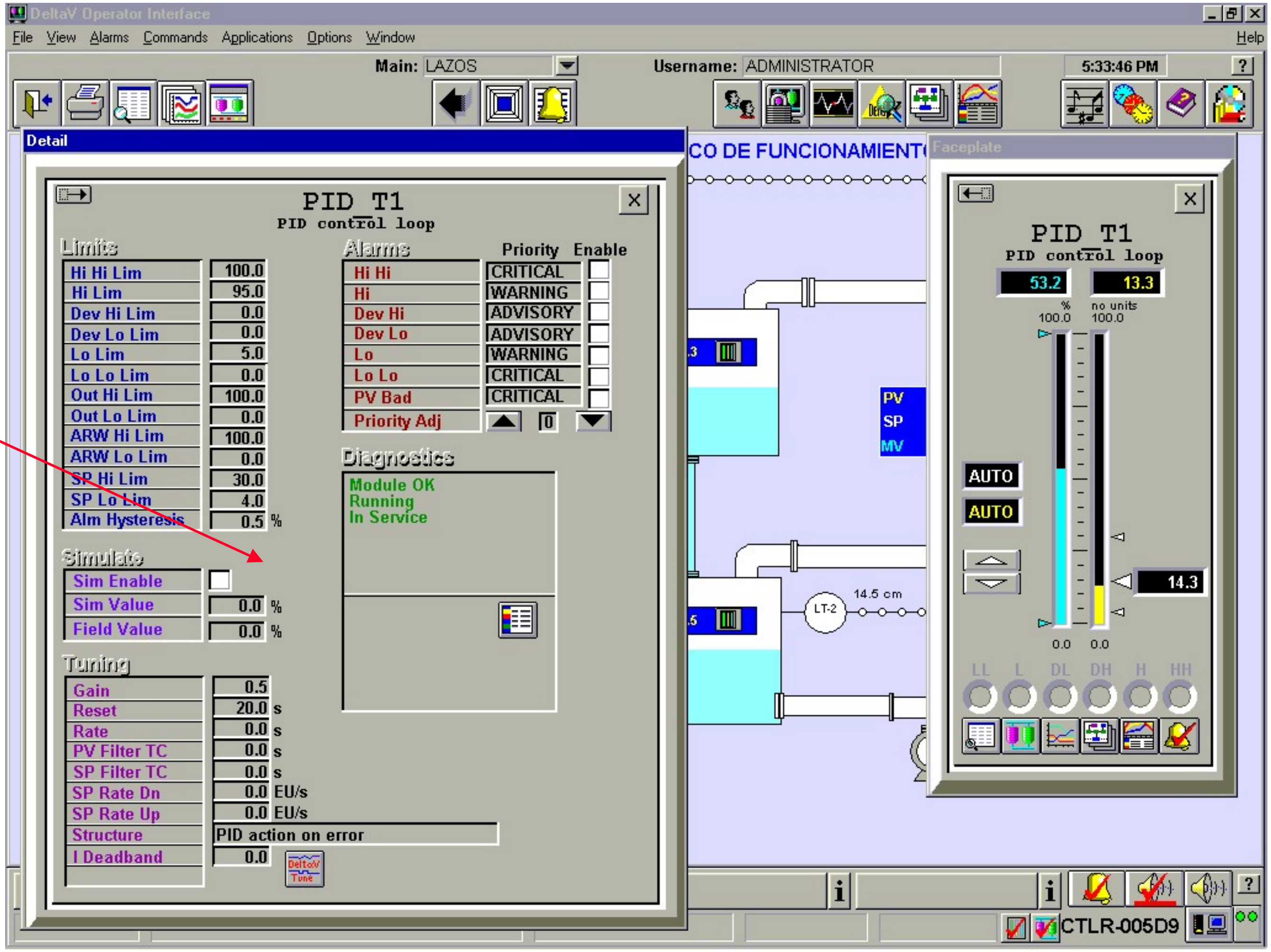


Typical PID face

Typical operator screen

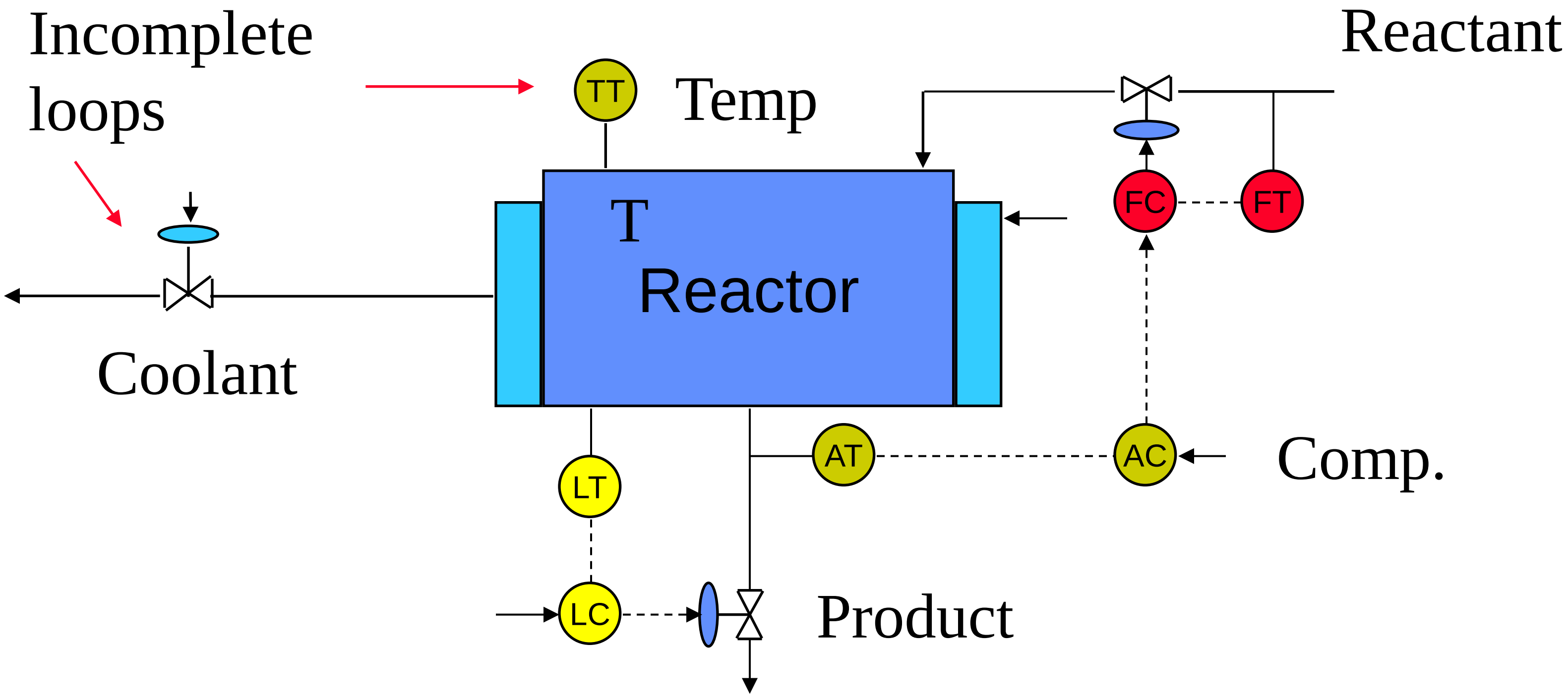
Configuration

Forms with configuration parameters



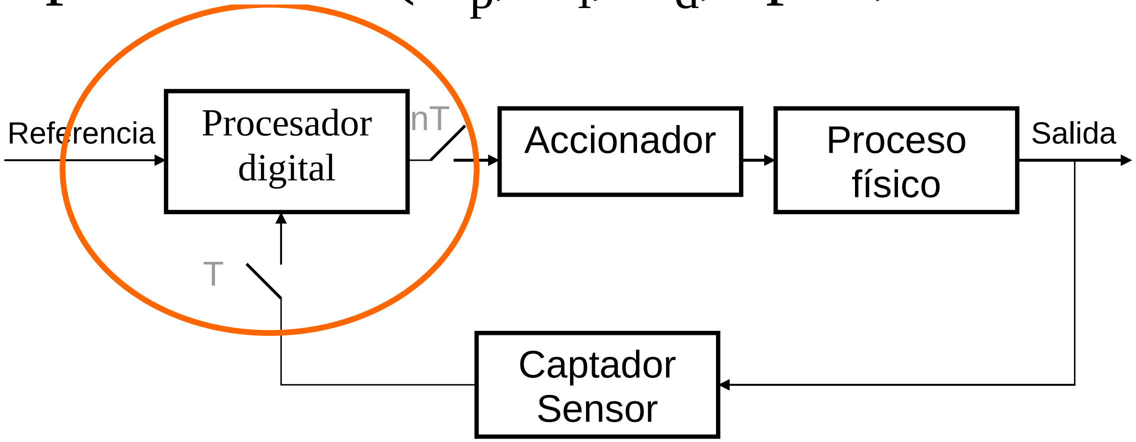
Java – Regula / Configuration

- A control system is a set of interconnected loops

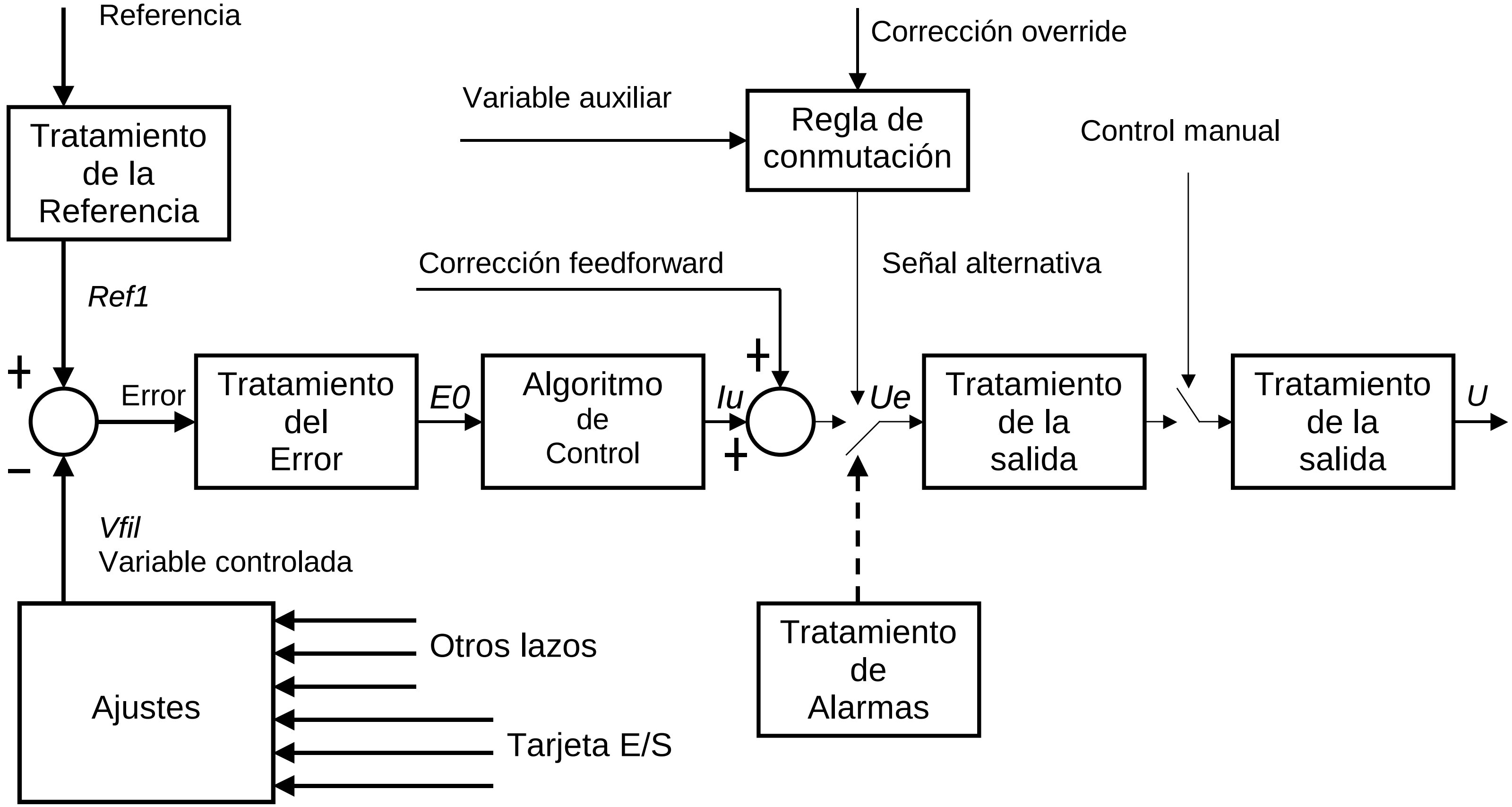


Java – Regula / Configuration

- For each loop one should specify:
  - Which are its inputs and outputs (w, y, u)
  - How the loop is connected to other loops (cascade, single loop,…)
  - Its parameters (K_p, T_i, T_d, span, constraints,…)



Java – Regula / Control loop



Configuration file

```
# Periodo-Basico-Muestreo(sg)    Tpo-Graficas(min)    Per-Muestras-Hist(sg)
0.2                                     5                    1
#NOMBRE LAZO            CODIGO
           Nivel        L01
#CABLE-ENTRADA       CABLE-SALIDA   BLOQUE-ENTRADA     V0      V1     V2
          1             0                 L01          0       0      0
#TIPO-AJUSTE        AJ0      AJ1      AJ2       PERIODO-MUESTREO(sg)
       1             0        20          0            1
#TIPO-REGULADOR   MODO(adaptativo)     AUTOMATICO
          1       1                     0
#REFERENCIA-INICIAL    CONTROL-INICIAL
             20          0
#SPAN-MEDIDA   INCREMENTO-MAXIMO-MEDIDA        FACTOR-FILTRADO
       100        10                             0
#CONTROL-MIN      CONTROL-MAX  INCREMENTO-MAXIMO-CONTROL
     0              100          10
#Kp            Ti       Td      G0         G1        G2
   5           0        0       0           0        0
#TIPO-REFERENCIA                      Ccr
             0           0
#TIPO-ERROR      cce1     cce2  cce3
       0          0       0      0
#NUMERO-FEED-FORWARD    LAZOS-DE-DONDE-VIENEN
             0
#TIPO-VALVULA      Tcv    Ccv
        0            0
#TRATAMIENTO-ALARMA      Pala     Vinf     Vsup   Varer    Halar
          0                0        0         0      0        0
#ESCALA-INF    ESCALA-SUP   TIEMPO-GUARDAR-DATOS-GRAFICAS(en periodos basicos
        0       100              5
```

Tuning in DCS

There are applications to help in the automatic or manual tuning in the DCS