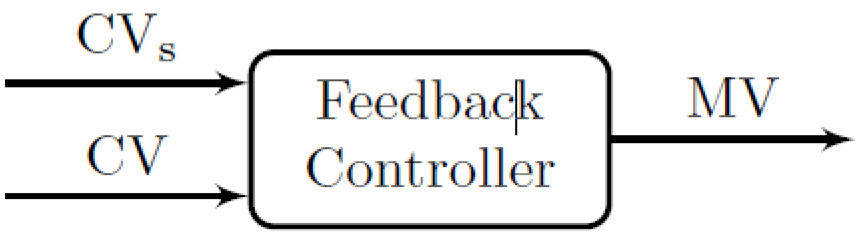
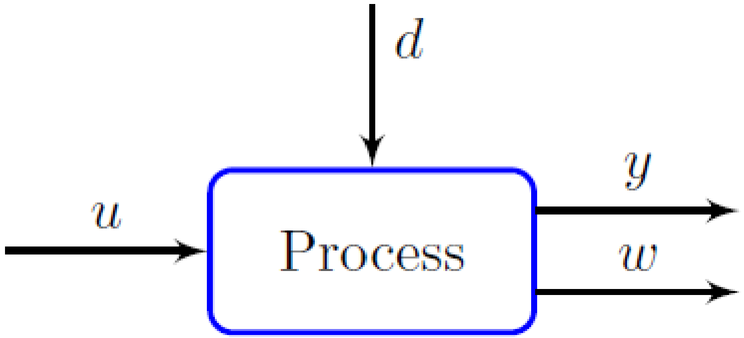
Part 1

- Basis, notation, PID control, feedback vs. feedforward
- Introduction to advanced regulatory control (ARC)
    - The three main inventions of process control

NOTATION and BLOCK DIAGRAMS



MV = manipulated variable (controller output)
CV = controlled variable (controller input)



u = process input (adjustable)
d = disturbance (non-adjustable)
y = process output (with setpoint y_s)
w = extra measured process variable (state x, y_2)

Combine:

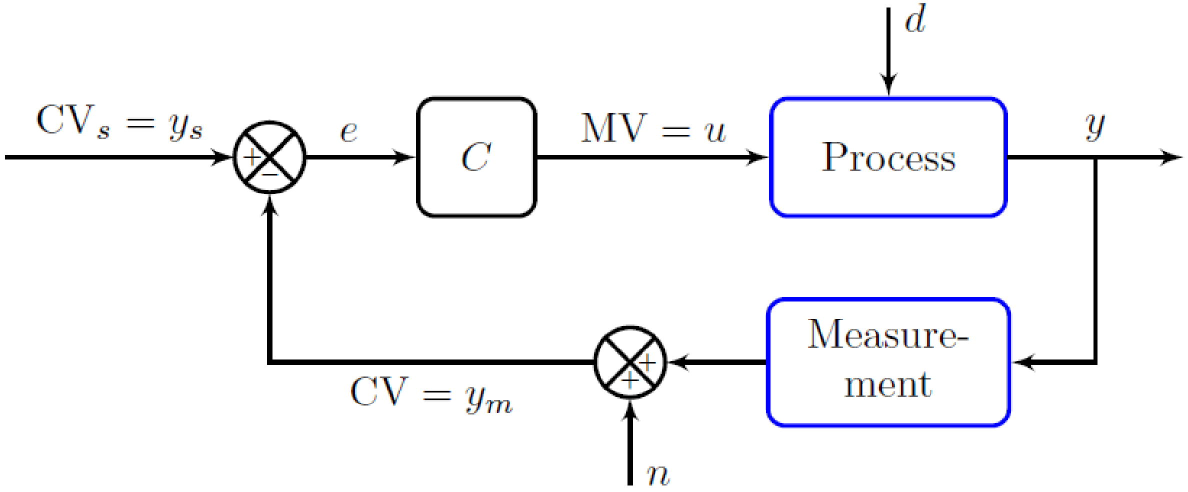


Figure 3: Block diagram of common "one degree-of-freedom" negative feedback control system.

$$e = y_s - y_m$$

Block diagrams: All lines are signals (information)

PID control

$$u(t) = K_c e(t) + K_c \tau_D \frac{de(t)}{dt} + \underbrace{\overbrace{\frac{K_c}{\tau_I} \int_{t_0}^{t} e(t')dt'}^{u_I} + u_0}_{\text{bias}=b}$$

$$K_c = \text{controller gain}$$

$$\tau_I = \text{integral time [s, min]}$$

$$\tau_D = \text{derivative time [s, min]}$$

- «You need a PhD to tune a PID»

AMERICAN CONTROL CONFERENCE
San Diego, California
June 6-8, 1984

IMPLICATIONS OF INTERNAL MODEL CONTROL FOR PID CONTROLLERS

Manfred Morari
Sigurd Skogestad

Daniel F. Rivera

California Institute of Technology
Department of Chemical Engineering
Pasadena, California 91125

University of Wisconsin
Department of Chemical Engineering
Madison, Wisconsin 53706

252 Ind. Eng. Chem. Process Des. Dev. **1986**, *25*, 252-265

Internal Model Control. 4. PID Controller Design

Daniel E. Rivera, Manfred Morari,* and Sigurd Skogestad

Chemical Engineering, 206-41, California Institute of Technology, Pasadena, California 91125

For a large number of single input-single output (SISO) models typically used in the process industries, the Internal Model Control (IMC) design procedure is shown to lead to PID controllers, occasionally augmented with a first-order lag. These PID controllers have as their only tuning parameter the closed-loop time constant or, equivalently, the closed-loop bandwidth. On-line adjustments are therefore much simpler than for general PID controllers. As a special case, PI- and PID-tuning rules for systems modeled by a first-order lag with dead time are derived analytically. The superiority of these rules in terms of both closed-loop performance and robustness is demonstrated.

Probably the best **simple** PID tuning rules in the world

Sigurd Skogestad*
Department of Chemical Engineering
Norwegian University of Science and Technology
N–7491 Trondheim Norway

Presented at AIChE Annual meeting, Reno, NV, USA, 04-09 Nov. 2001
Paper no. 276h ©Author
This version: November 19, 2001†

Abstract

The aim of this paper is to present analytic tuning rules which are as simple as possible and still result in a good closed-loop behavior. The starting point has been the IMC PID tuning rules of Rivera, Morari and Skogestad (1986) which have achieved widespread industrial acceptance. The integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, we start by approximating the process by a first-order plus delay processes (using the "half method"), and then use a single tuning rule. This is much simpler and appears to give controller tunings with comparable performance. All the tunings are derived analytically and are thus very suitable for teaching.

1 Introduction

Hundreds, if not thousands, of papers have been written on tuning of PID controllers, and one must question the need for another one. The first justification is that PID controller is by far the most widely used control algorithm in the process industry, and that improvements in tuning of PID controllers will have a significant practical impact. The second justification is that the simple rules and insights presented in this paper may contribute to a significantly improved understanding into how the controller should be tuned.

ELSEVIER

Journal of Process Control 13 (2003) 291–309

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Simple analytic rules for model reduction and PID controller tuning☆

Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Received 18 December 2001; received in revised form 25 June 2002; accepted 11 July 2002

Abstract

The aim of this paper is to present analytic rules for PID controller tuning that are simple and still result in good closed-loop behavior. The starting point has been the IMC-PID tuning rules that have achieved widespread industrial acceptance. The rule for the integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, there is a just a single tuning rule for a first-order or second-order time delay model. Simple analytic rules for model reduction are presented to obtain a model in this form, including the "half rule" for obtaining the effective time delay.

Keywords: Process control; Feedback control; IMC; PI-control; Integrating process; Time delay

1. Introduction

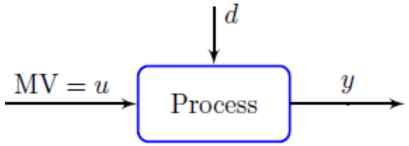
Although the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good values (settings) for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned. The tuning rules presented in this paper have developed mainly as a result of teaching this material, where there are several objectives:

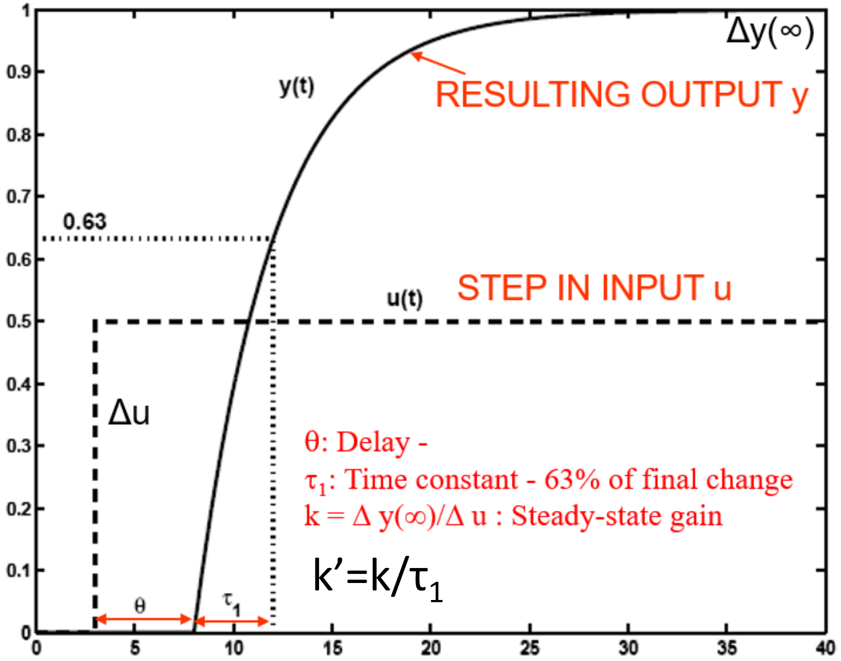
1. The tuning rules should be well motivated, and preferably model-based and analytically derived.
2. They should be simple and easy to memorize.
3. They should work well on a wide range of processes.

Step 2. Derive model-based controller settings. PI-settings result if we start from a first-order model, whereas PID-settings result from a second-order model.

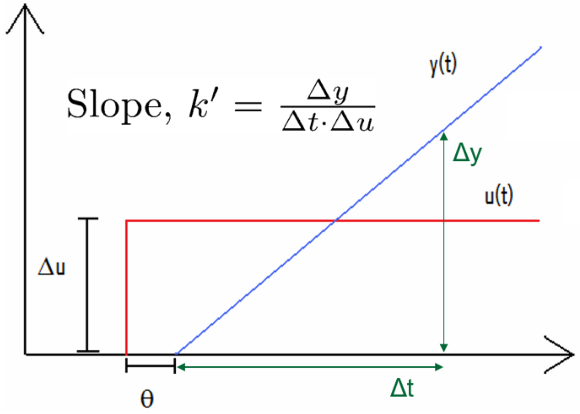
There has been previous work along these lines, including the classical paper by Ziegler amd Nichols [1], the IMC PID-tuning paper by Rivera et al. [2], and the closely related direct synthesis tuning rules in the book by Smith and Corripio [3]. The Ziegler–Nichols settings result in a very good disturbance response for integrating processes, but are otherwise known to result in rather aggressive settings [4,5], and also give poor performance for processes with a dominant delay. On the other hand, the analytically derived IMC-settings in [2] are known to result in a poor disturbance response for integrating processes (e.g., [6,7]), but are robust and

SIMC PID tuning rule





Step response integrating process



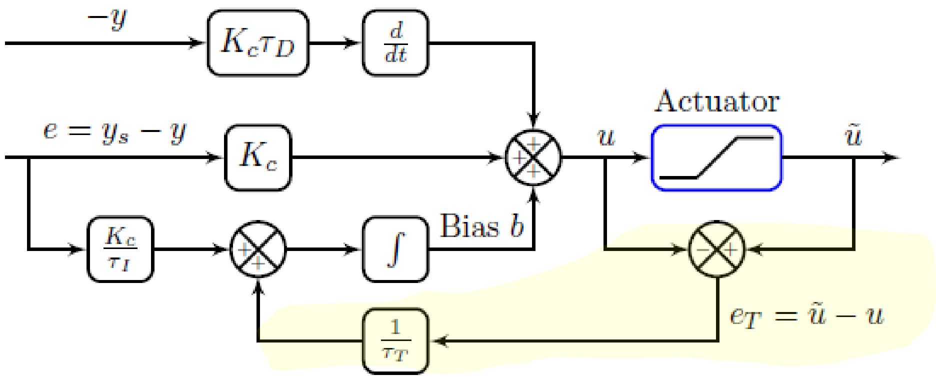


Figure 7: Recommended PID-controller implementation with anti-windup using tracking of the actual controller output ($\tilde{u}$), and without D-action on the setpoint. (Åström & Hägglund, 1988).

$\int$ = integral = $\frac{1}{s}$ in Laplace domain
$\frac{d}{dt}$ = derivative = s in Laplace domain
K_c = controller gain
τ_I = integral time [s, min]
τ_D = derivative time [s, min]
τ_T = tracking time constant for anti-windup [s, min]
+ τ_F = filter time constant (on measurement y)

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

$$\tau_I = \min(\tau_1, 4(\tau_c + \theta))$$

$$\tau_D = \tau_2$$

Only one tuning parameter:
Closed-loop time constant:

$$\tau_c \geq \theta$$

(gives Gain Margin>3)

Filter time constant, $\tau_F \leq \frac{\tau_c}{2}$

Note: For integrating processes (with large time constant τ_1):

- Both K_c and τ_I depend on τ_c
- Example: Level control (often poorly tuned), $K_c\tau_I = 4/k'$

Motivation for better control: Squeeze and shift rule

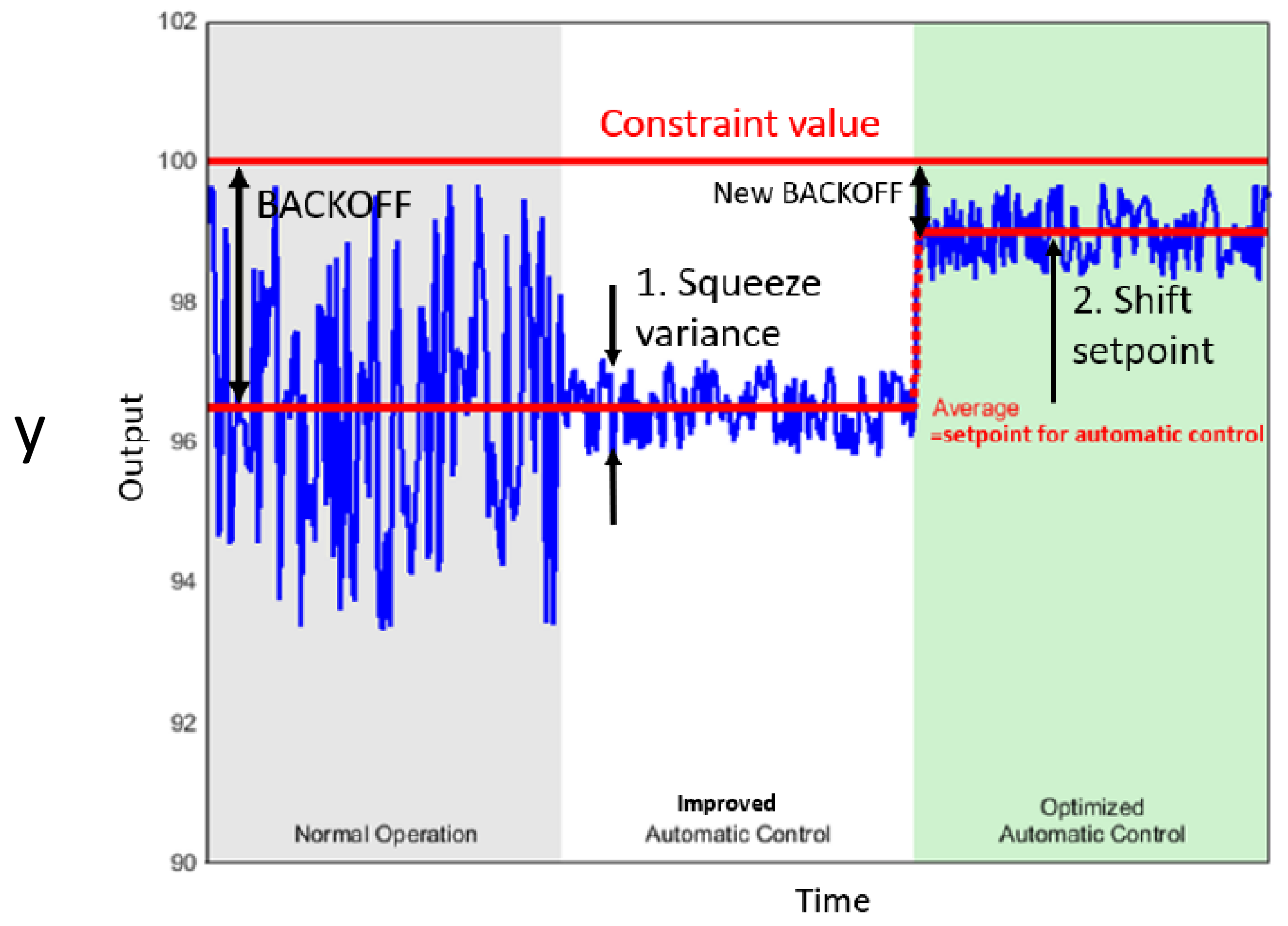
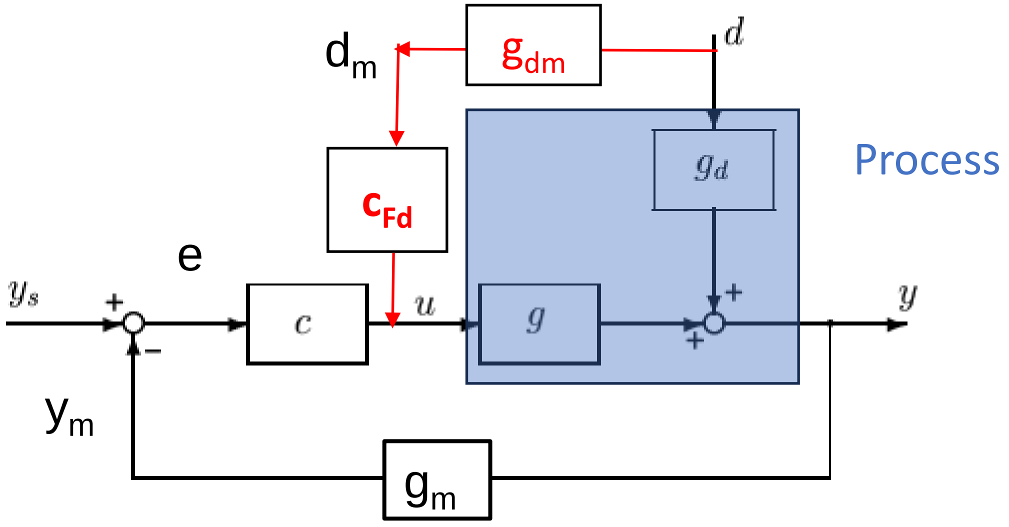


Figure 8: Squeeze and shift rule: Squeeze the variance by improving control and shift the setpoint closer to the constraint (i.e., reduce the backoff) to optimize the economics (Richalet et al., 1978).

Feedforward control: Measure disturbance (d)



Block diagram of feedforward control

c = Feedback controller

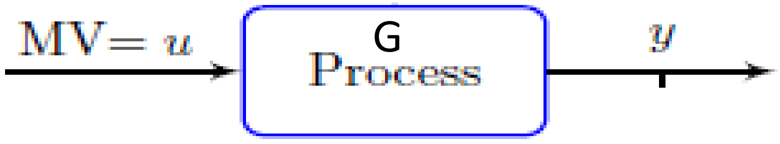
c_{Fd} = Feedforward controller.

Ideal, inverts process g: $c_{Fd} = g^{-1} g_d \, g_{dm}^{-1}$

Usually: Add feedforward when feedback alone is not good enough, for example, because of measurement delay in g_m

What is best? Feedback or feedforward?

Example: Feedback vs. feedforward for setpoint control of uncertain process



y = G(s) u

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3,\ \tau = 6 \qquad \text{(B.2)}$$

Desired response : $y = \dfrac{1}{\tau_c s + 1} y_s = \dfrac{1}{4s + 1} y_s$

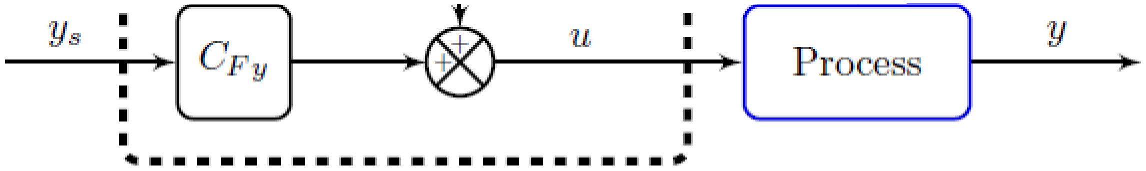


Figure A.42: Block diagram of feedforward control system with linear combination of feedforward from measured disturbance (d) and setpoint (y_s) (E14).

Feedforward solution. We use feedforward from the setpoint (Fig. A.42):

$$u = C_{Fy}(s) y_s$$

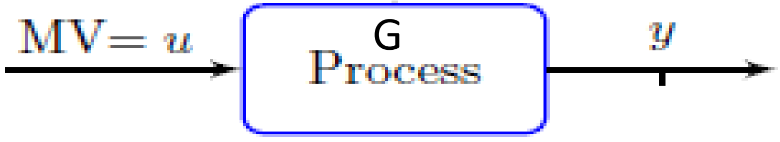
where we choose

$$C_{Fy}(s) = \frac{1}{\tau_c s + 1} G(s)^{-1} = \frac{1}{k} \frac{\tau s + 1}{\tau_c s + 1} = \frac{1}{3} \frac{6s + 1}{4s + 1} \qquad \text{(B.3)}$$

The output response becomes as desired,

$$y = \frac{1}{4s + 1} y_s \qquad \text{(B.4)}$$

Example: Feedback vs. feedforward for setpoint control of uncertain process



y = G(s) u

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3, \; \tau = 6 \qquad \text{(B.2)}$$

$$\text{Desired response}: \quad y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s$$

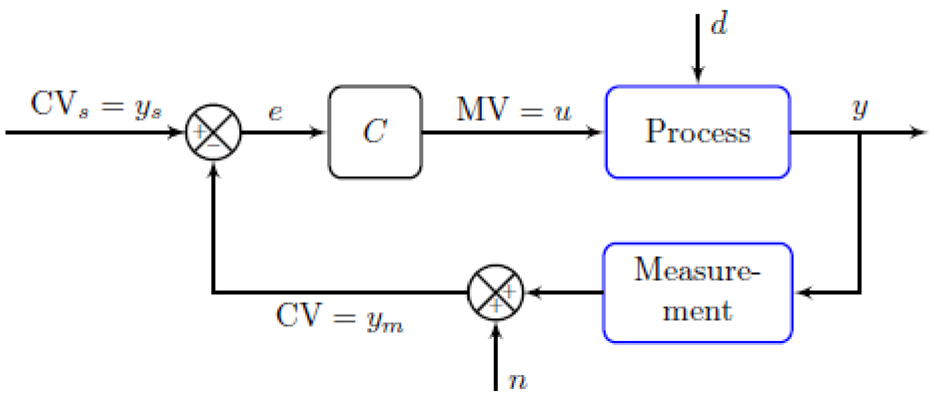


Figure 3: Block diagram of common "one degree-of-freedom" negative feedback control system.

Feedback solution. We use a one degree-of-freedom feedback controller (Fig. 3) acting on the error signal $e = y_s - y$:

$u = C(s)(y_s - y)$

We choose a PI-controller with $K_c = 0.5$ and $\tau_I = \tau = 6$ (using the SIMC PI-rule with $\tau_c = 4$, see Appendix C.2):

$$C(s) = K_c \left(1 + \frac{1}{\tau_I s}\right) = 0.5 \frac{6s + 1}{6s} \qquad \text{(B.5)}$$

Note that we have selected $\tau_I = \tau = 6$, which implies that the zero dynamics in the PI-controller C, cancel the pole dynamics of the process G. The closed-loop response becomes as desired:

$$y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s \qquad \text{(B.6)}$$

Proof. $y = T(s) y_s$ where $T = L/(1 + L)$ and $L = GC = kK_c/(\tau_I s) = 0.25/s$. So $T = \frac{0.25/s}{1+0.25/s} = \frac{1}{4s+1}$.

Thus, we have two fundamentally different solutions that give the same nominal response, both in terms of the process input $u(t)$ (not shown) and the process output $y(t)$ (black solid curve in Fig. B.43).

- But what happens if the process changes?
  - Consider a gain change so that the model is wrong
    - Process gain from k=3 to k'=4.5

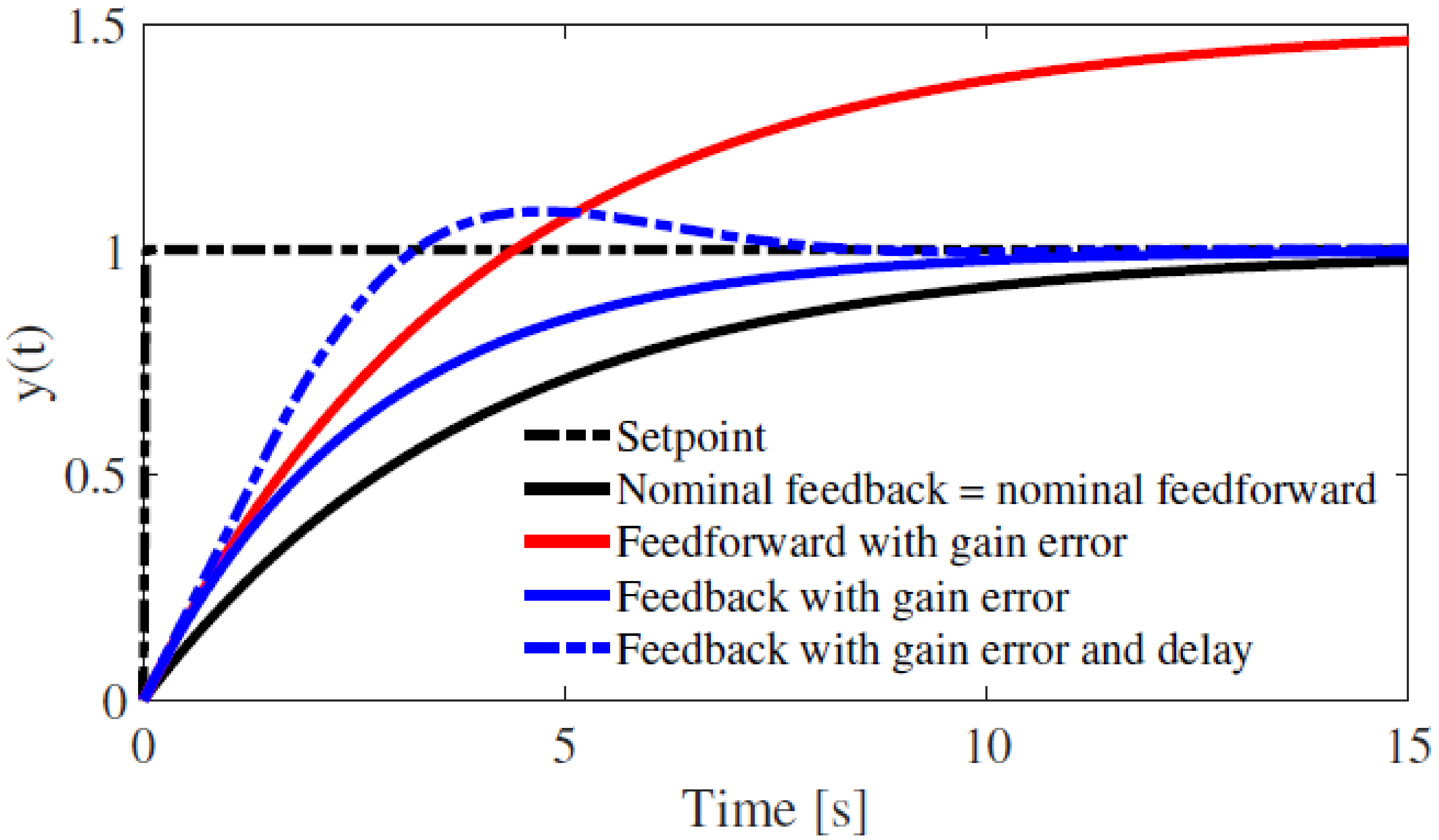


Figure B.43: Setpoint response for process (B.2) demonstrating the advantage of feedback control for handling model error.

Gain error (feedback and feedforward): From k=3 to k'=4.5

Time delay (feedback): From $\theta = 0\ to\ \theta = 1.5$

Introduction to Advanced regulatory control (ARC)

Uses Flowsheets

"Control" is to make active use of the inputs u *to counteract disturbances* d *such that the outputs* y *stay close to their desired setpoints* y_s.

The three main inventions of process control

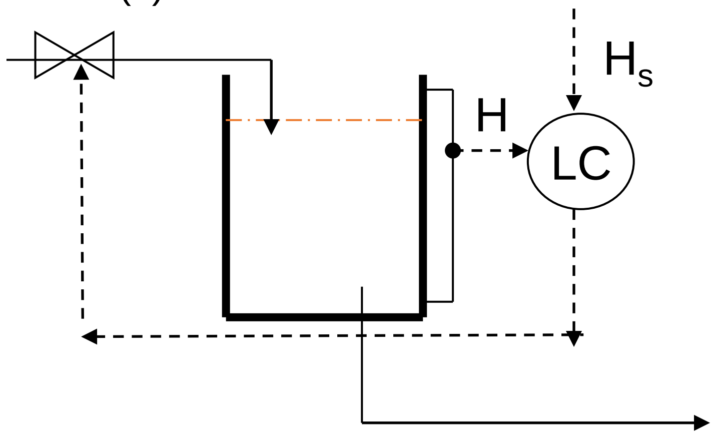
(became widely used in the process industry in the 1930s)

1. PID control, and in particular integral action ("bias reset)
2. Cascade control
3. Ratio control (special case of feedforward which needs no explicit model)

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

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

Flowsheet (P&ID) of feedback control
Example: Level control (with given outflow)



Solid lines: mass flow (streams)

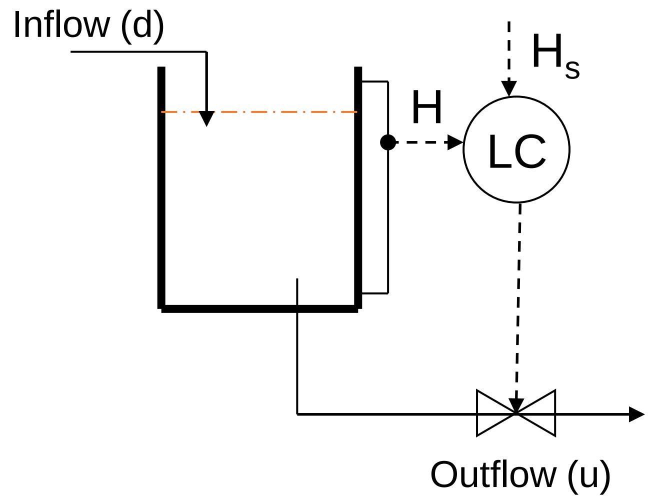
Dashed lines: signals (control)

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

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

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

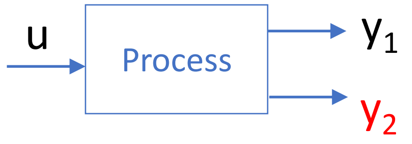
Level control when inflow is given (alternative input/output-pairing)



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

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

Invention 2. Cascade control



y1=primary output (given setpoint)
y2=secondary output (adjustable setpoint)

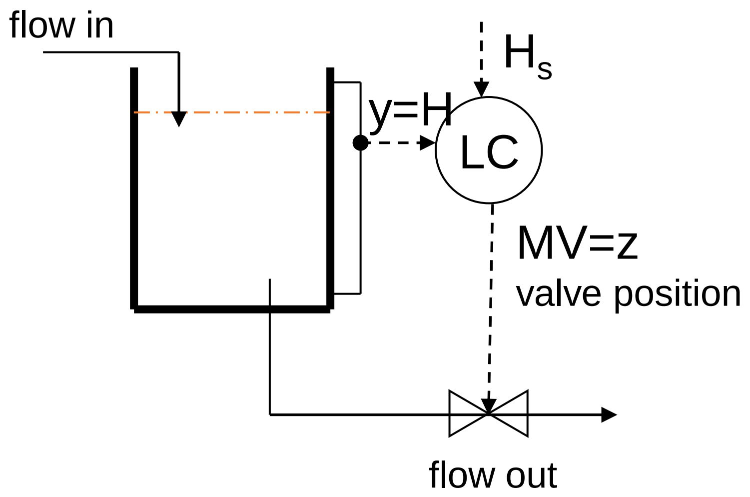
Idea: make use of extra "local" output measurement (y_2)

Implementation: Controller ("master") gives setpoint to another controller ("slave")

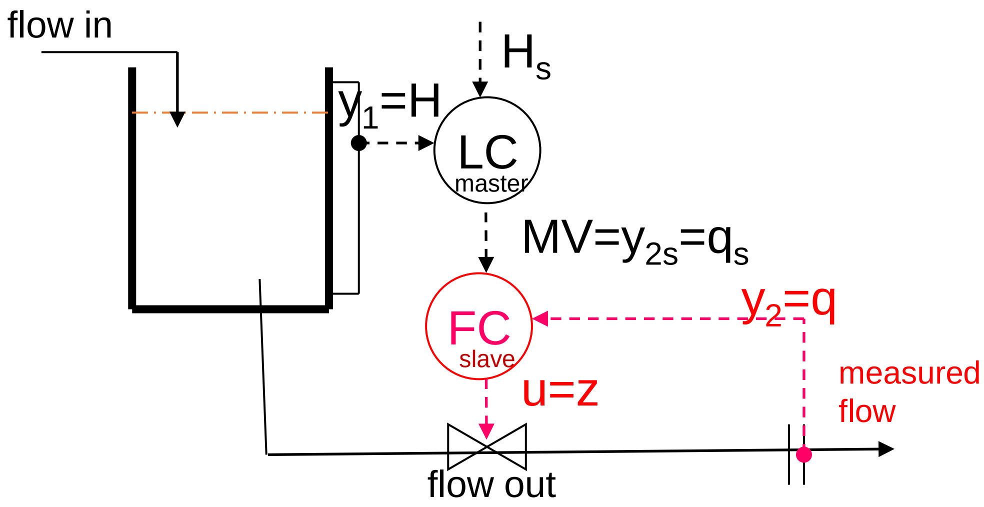
- Without cascade: "Master" controller directly adjusts u to control primary output y_1)
- With cascade: Local "slave" controller (fast) uses u to control "extra"/fast measurement (y_2). *
  "Master" (slow) controller adjusts setpoint y_{2s}.

- **Example: Flow controller on valve (very common!)**
  - y_1 = level H in tank
  - u = valve position (z)
  - y_2 = flowrate q through valve

WITHOUT CASCADE

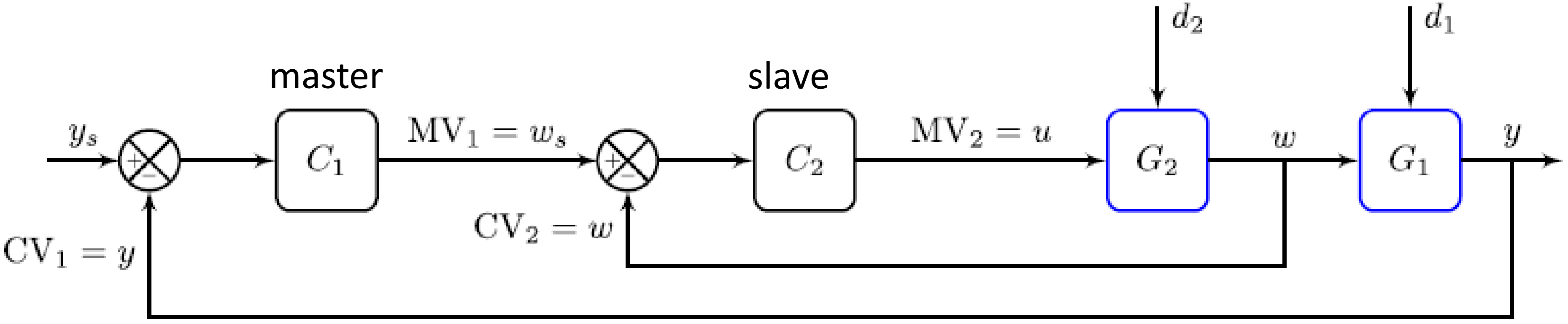


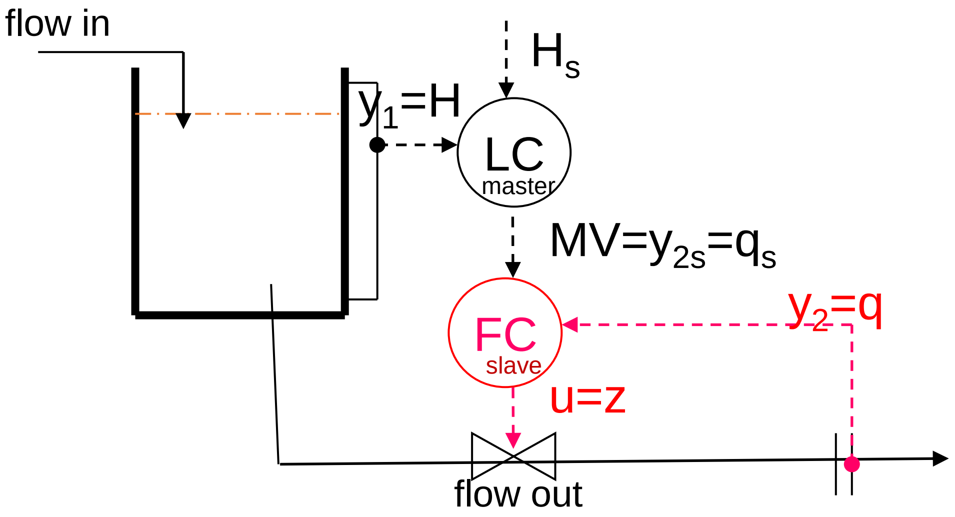
WITH CASCADE



*Comment: Another approach that uses extra measurements to improve control is «**Full state feedback».**

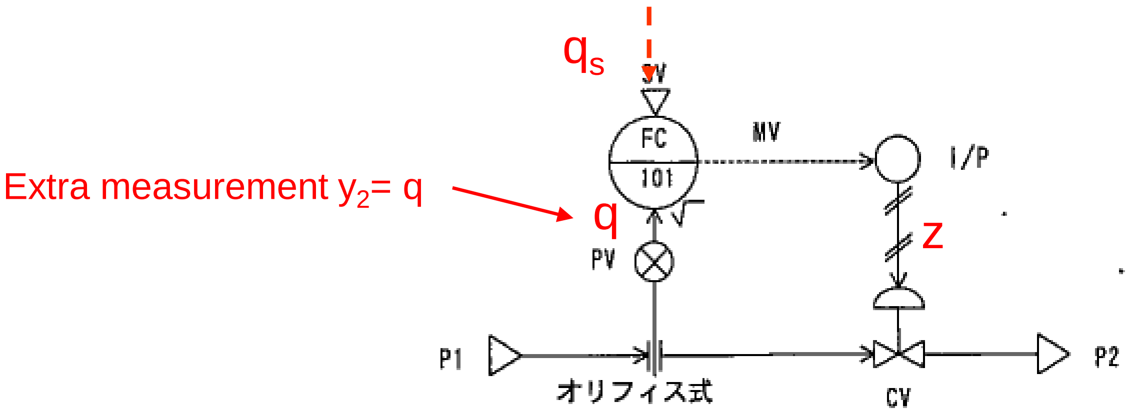
Block diagram of cascade control





$y = y_1 = H$ (measured level)
$u = z$ (valve position)
$w = y_2 = q$ (measured outflow)
C_1 = LC (P-controller)
C_2 = FC (I-controller)
$G_1 = k'/s$ (level is integrating)
G_2 = valve model (nonlinear, static)
d_1 = downstream pressure
d_2 = inflow

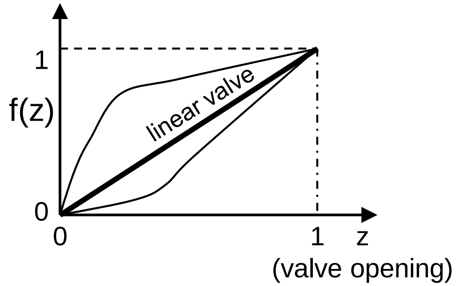
What are the benefits of adding a flow controller (slave=inner cascade)?



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

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

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



Time scale separation is needed for cascade control to work well

- Inner loop (slave) should be at least 4 times* faster than the outer loop (master)
- But normally recommend 10 times faster
    - tauc_master > 10 tauc_slave
- Otherwise, the slave and master loops may start interacting
    - The fast slave loop is able to correct for local disturbances, but the outer loop does not «know» this and if it’s too fast it may start «fighting» with the slave loop.
    - This may also result in stability problems

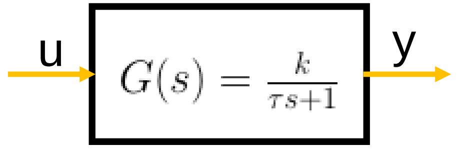
* Shinskey (Controlling multivariable processes, ISA, 1981, p.12)

Response of linear first-order system (with time constant τ)

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

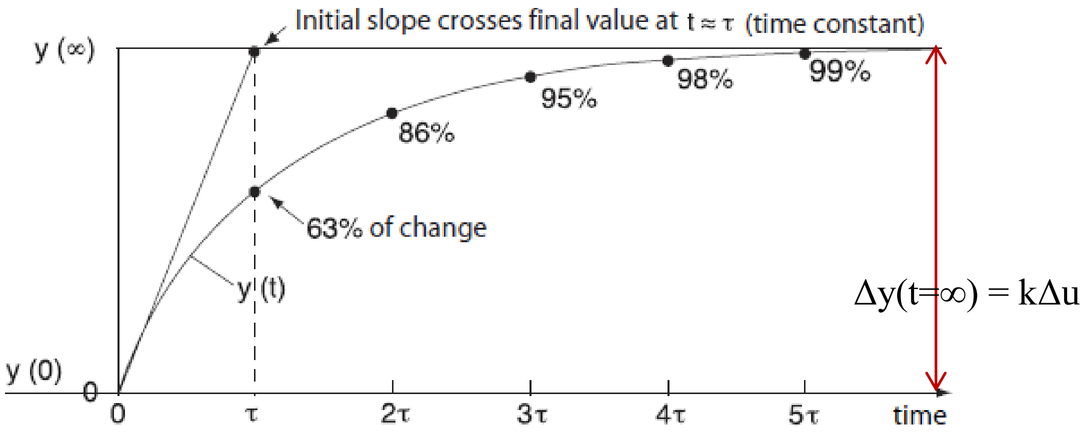
Initially at rest (steady state): $y(0) = y_0.$

Make step in u at $t = 0$: Δu



Block diagram with transfer function for first-order process

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	

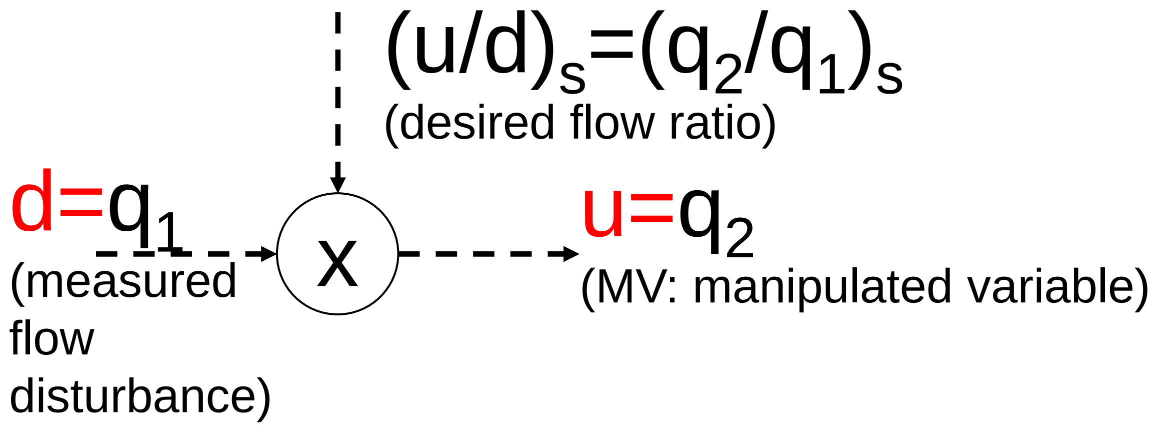
Convergence rate (of inner loop):

- 63% after 1τ
- 98% after 4τ (recommended lower limit)
- To be safe (process changes): 10 τ

Invention 3. Ratio control

Example: Process with two feeds $d=q_1$ and $u=q_2$, where ratio should be constant.

Use multiplication block (x):

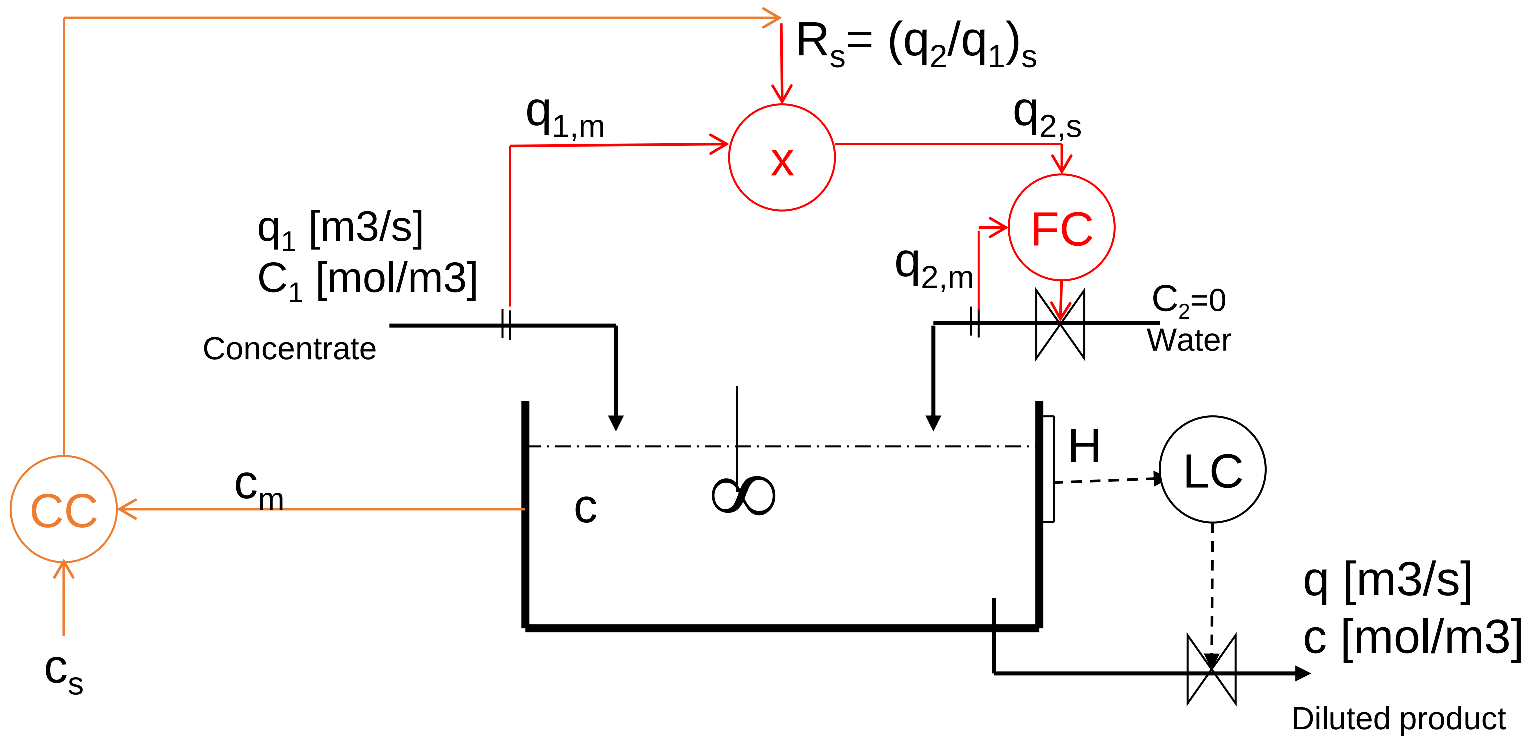


Usually: Combine ratio (feedforward) with feedback

- Adjust $(q_1/q_2)_s$ based on feedback from process, for example, composition controller.
    - Maybe be viewed a special case of cascade control

- ***Example mixing process (cake baking)****: Use recipe initially (ratio control = feedforward), but adjust ratio if result is not as desired (feedback)*

EXAMPLE: RATIO CONTROL FOR MIXING PROCESS



CC: outer “cascade” = feedback trim” (correction) of ratio setpoint

Later: Will see that this is a special case of **input transformation:**

Transformed input (as seen from feedback controller CC) is $v = (q_2/q_1)_s$

The three main inventions of process control can only indirectly and with effort be implemented with MPC

1. Integral action with MPC: Need to add artificial integrating disturbance in estimator
   - ARC: Just add an integrator in the controller (use PID)

2. Cascade control with MPC: Need model for how u and d affect y_1 and y_2.
   - ARC: Just need to know that control of y_2 indirectly improves control of y_1

3. Ratio control with MPC: Need model for how u and d affect property y
   - ARC: Just need the insight that it is good for control of y to keep the ratio R=u/d constant

Because of this, MPC should be on top of a regulatory control layer with the setpoints for y_2 and R as MVs.