Advanced Controllers

The main limits of the SISO PID controller is the to simple algorithm. We can do it more complex:

1. PID controller

   External reset,

   Decoupling,

   Gain scheduling,

   autotuning

2. Controll loop

   (a) Cascade;

   (b) Selector (override);

   (c) Feedforward.

1 Cascade Control

Processes we discussed before were SISO, if object has additional I/O then

- General MIMO theory can be used

  we need accurate model of the process, design software - all that is not chip and takes a lot of time.

- Decompose the Object and Controller into parts and use simple SISO blocks.

Decomposition can be:

- Vertical: series or cascade;
- Horizontal: decentralized.

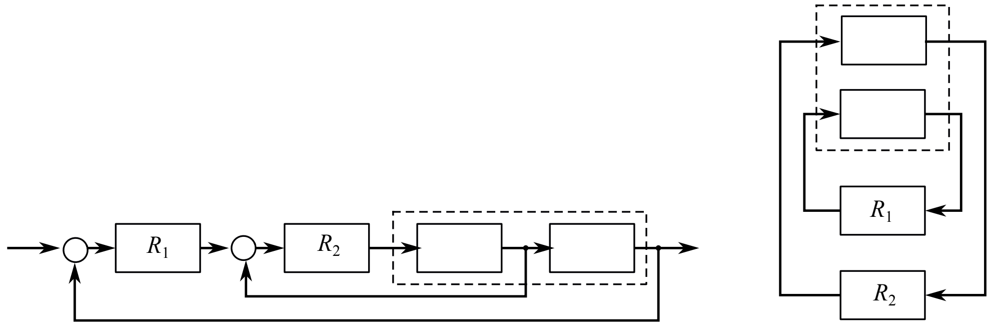
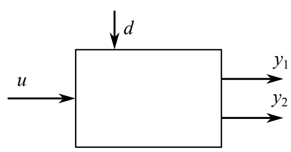


Figure 1: Decomposition of the system

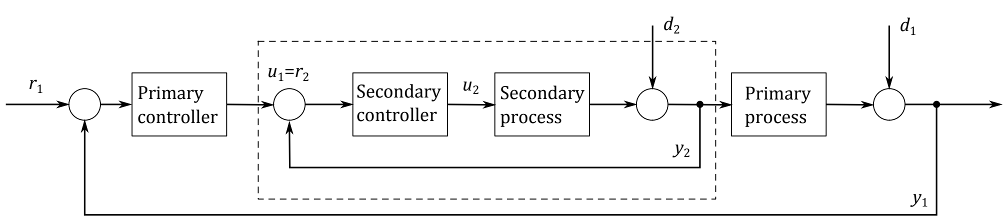
Additional information (outputs) about the object is used:



- to decrease disturbance,
- for linearization,
- for better dynamics.

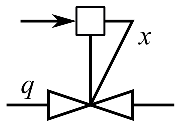
Additional output y_2 should be quick, then it can be used effectively.

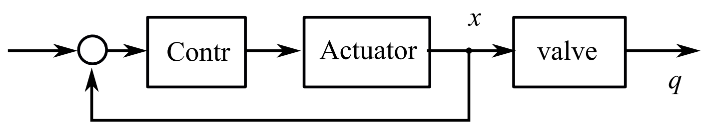
One way to improve the dynamic response to load changes is by using a secondary measurement point and a secondary controller . The purpose of the such kind of control is to ensure stability of the primary process variable by regulating a secondary in accordance with the needs of the first [2], [3].



Example 1 *Valve positioner*

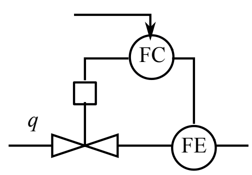
Valve positioner is an inner loop, where x - is the position and q - flow





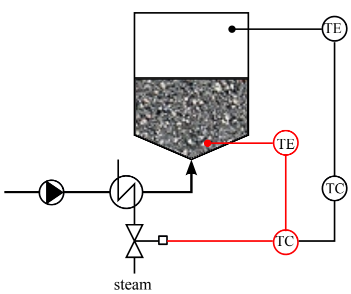
Example 2 *Flow*

Stabilization of the flow $q(P, x)$ *in the heat exchanger*



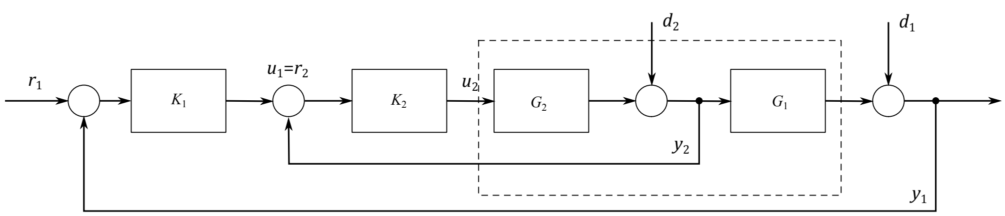
eliminates the influence of the pressure P*, measures and keeps the flow* q

Example 3 *Dryer*



Disturbances:

- *Dampness;*
- *Air temperature;*
- *Steam temperature.*



$$u_2 = K_2 K_1 (r_1 - y_1) - K_2 y_2 \quad (1)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_1 \cdot G_2 \\ G_2 \end{bmatrix} \cdot u_2 \quad (2)$$

One strict requirement for cascade control in action: the slave loop must be faster-responding than the master loop. If this speed relationship is reversed control technique won't function. Cascading not only "shielding" load variables from primary controller, but also simplifies the dynamic characteristics of the process [3].

Cascade should be used:

1. Use it if G_1 is not non minimal phase system and disturbances d_2 significant.

2. Use it if G_2 is not very well known or nonlinear.

   Inner loop $L_2 = G_2 \cdot K_2 >>$ is accurate and fast.

   Feedback decreases the parameters deviations and inaccuracies ($\mathrm{d}T/T = S \cdot \mathrm{d}L/L$).

3. Use if on order to make system quicker.

   PID controller frequency margin until the frequency where phase shift of the object $= -180°$

Tuning

1. The secondary loop is tuned first, the primary loop is left in manual while the secondary loop is tuned.

2. After the secondary loop is tuned, the primary loop is tuned with the secondary loop in automatic.

When the loops is tuned, careful attention should be paid to the response to set point changes.

Without doubt cascade control can bring about substantial improvements in the quality of control. However, the benefits are critically dependant upon proper implementation. The control scheme should be designed to target specific disturbances [5, 4].

Benefits:

- compensates for specific disturbances at source;

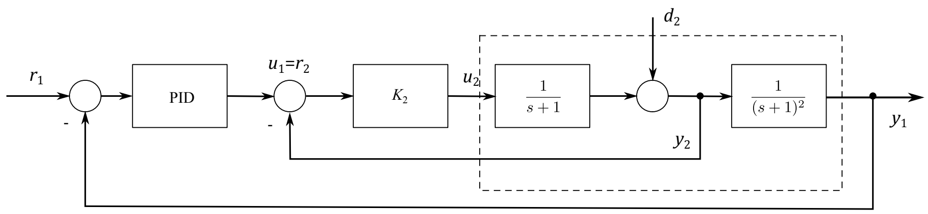
- improves dynamics;

- linearizes model.

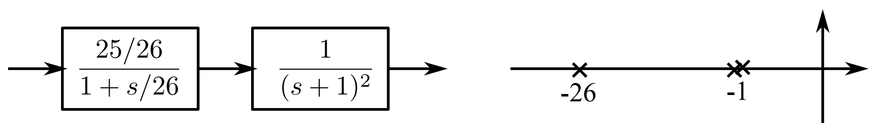
Example 4 *Triple pole*

$$W_o = \frac{1}{(s+1)^3}$$

Use cascade control:



Let's choose $K_2 = 25$*, then* $W_2 = \dfrac{K_2/(s+1)}{1 + K_2/(s+1)}$

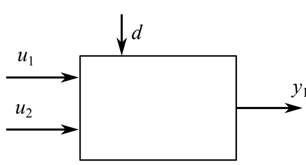


Primary loop:

- *transform into FOPDT* ($\tau = 0.5 + 1/26, T = 1.64$)
- *calculate PID parameters* ($K_p = \ldots, T_i = \ldots, T_d = \ldots$)

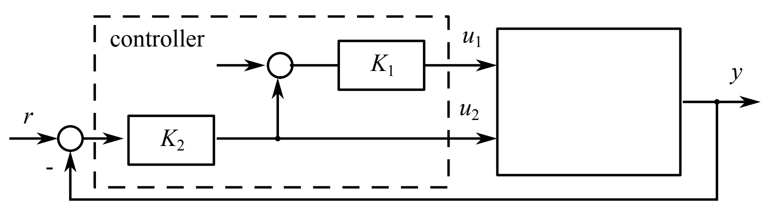
2 Mid ranging control - Additional inputs

Two inputs u_1, u_2 are needed to control y in order to increase accuracy and speed.



$$y = \begin{bmatrix} G_{11} & G_{12} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

mid ranging control



Fast channel K_2, slow and accurate channel K_1. $K_1 < 0$
In order to obtain the same direction of the channels u_1 and u_2, $u_2 \approx r_2 = 50\%$ - fast channel reacts on the change.

Example 5 *object* $G_{11} = G_{12} = 1 \quad y = u_1 + u_2, \quad r_2 = 0$
fast channel: $K_2 = 10/s$,
slow channel: $K_1 = -1/s$

$$\overbrace{W_a = \frac{10}{s} \cdot \left[1 - \frac{-1}{s}\right] = \frac{10(s+1)}{s^2}}^{\text{open loop}} \quad \overbrace{W_s = \frac{10s+10}{s^2+10s+10}}^{\text{closed loop}}$$

Features:

- $\lim_{t\to\infty} y = r, \quad \lim_{t\to\infty} u_2 = 0$! - woks on alternating signals
- used then system has one fast inaccurate input u_2 and slow, but accurate input u_1
- very often used with cascade controllers (boilers)

3 Override control

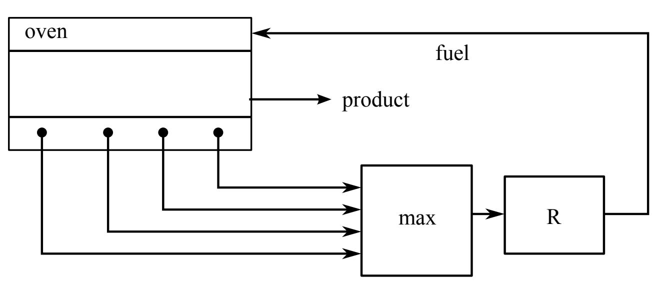
When there are more controlled variables than manipulated variables, a common solution to this problem is to use a selector to choose the appropriate process variable from among a number of available measurements. Selectors can be based on multiple measurement points, multiple final control elements, or multiple controllers, as discussed below. Selectors are used to improve the control system performance as well as to protect equipment from unsafe operating conditions [2].

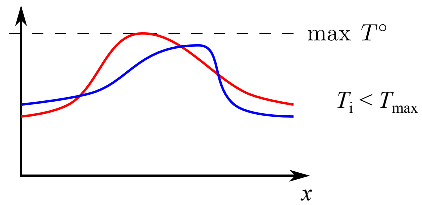
3.1 Inputs selector

Selection of inputs (criterion: max, min)

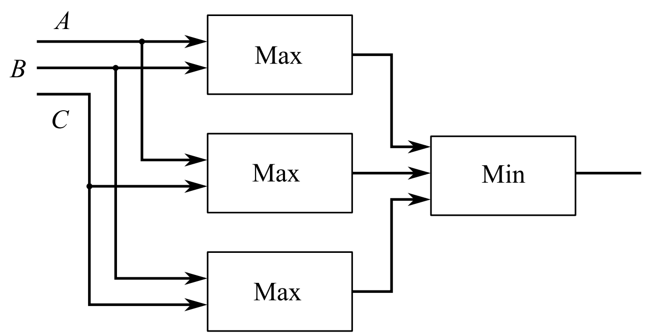
High-select, Low-select - one of the possible inputs

Example 6 *Oven Temperature*





Median - select - medium (of three)



Useful in situations where a redundant number of sensors is used to measure one process variable.

High - limit, Low - limit - one input, controller with signal limits

Generally used in cascade control strategies where the primary control output becomes the set point for the secondary controller. The master controller may compute a control law which, given as the set point for the slave controller, may result in unreachable or potentially unsafe operating conditions of the control loop.

Used in interlocking:

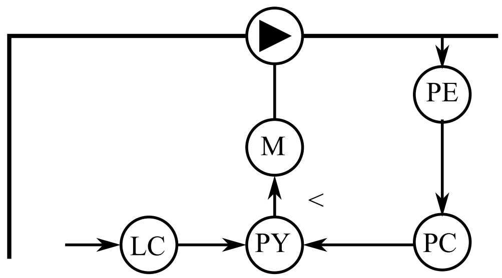
reaction on: alarms, load and failure.

3.2 Override

Change of the control algorithm, override into another control. The anti-reset windup feature is a type of override.

In certain circumstances (startup, shutdown, emergency) proceed differently.

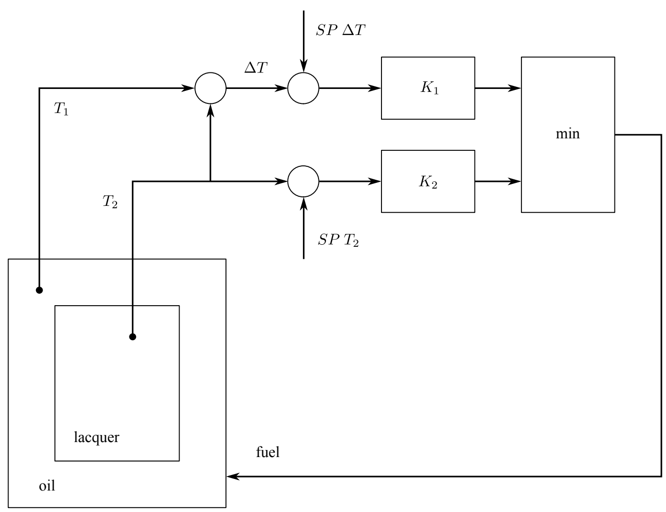
Example 7 *Water well pump*



keep pressure P
- *if level is high*

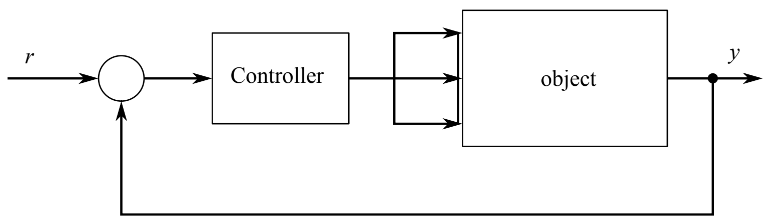
keep level
- *if level is low*

Example 8 *Resin heating (chemical reactor)*

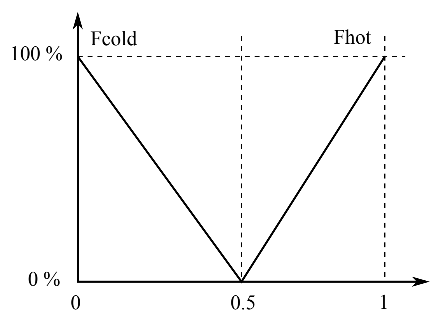


3.3 Split range control

Output into different places



temperature:	heating/cooling
neutralizing:	Acid/Alkaline
Multiple devices:	pumps, boilers, air compressors



4 Feedforward Control

By taking control action based on measured disturbances d rather than controlled variable error e , the controller can reject disturbances before they affect the controlled variable y.

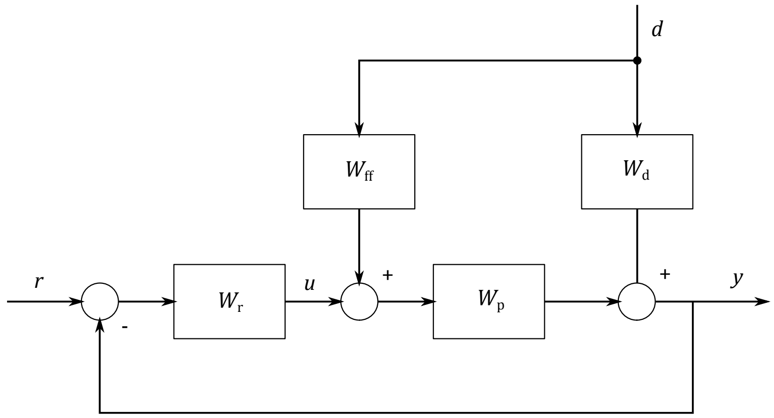
$$W_{ff} = -\frac{W_d}{W_p} \tag{3}$$

To calculate the control signal a mathematical models should be derived which describe the relations between control signal and output of the system ($u \rightarrow y$),and effect of disturbances on the controlled variable ($d \rightarrow y$). These models can be based on steady-state or dynamic analysis.

The performance of the feedforward controller depends on the accuracy of both models.

Feedforward W_{ff}:

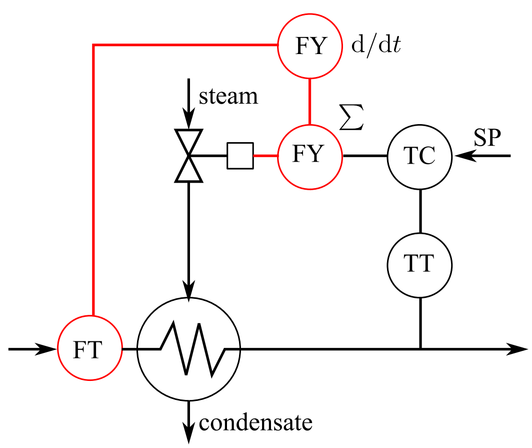
- Does not use output signal y;



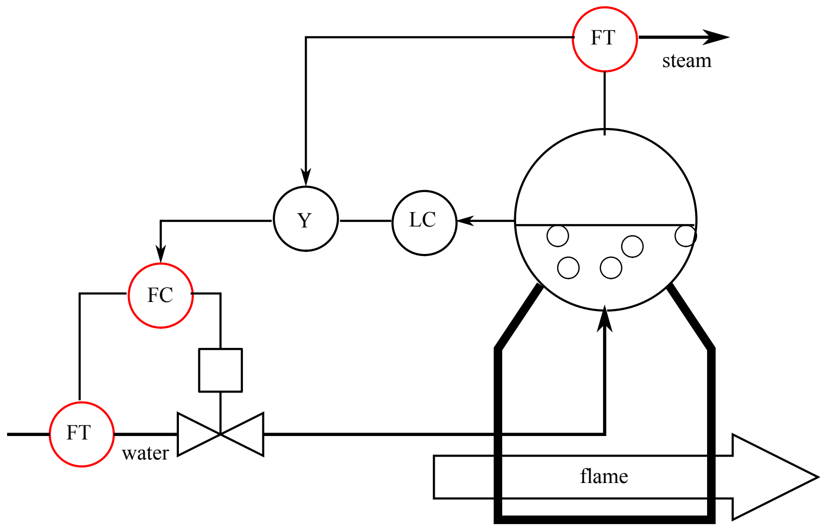
- Can be turned off and controller continues working;
- Does not work with PID algorithm.

Advantages	Disadvantages
- acts before the effect of disturbance has been felt by the system	- requires direct measurement of all possible disturbances
- good for system with large time constant or dead time	- cannot cope with unmeasured disturbances
- does not introduce instability in the closed loop	- sensitive to process/model error

Example 9 *Control of a heat exchanger considering the flow*

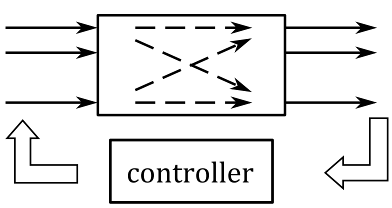


Example 10 *Water level in the boiler steam drum*



5 Multivariable Regulatory Control

I/O interaction.



Can be controlled if:

- Has not a strong connection;
- Is not very complex.

Also the problem is operating point (steady state, statics)

nonlinearities,

start-up, shut-down procedures.

Implementation of the MIMO control

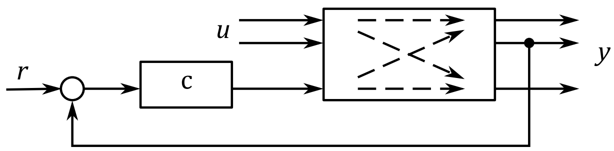
1. Multiloop control,

2. Decoupling,
3. Multi-dimensional control of MPC.

5.1 Multiloop Control

Works with several SISO controllers.

One input controls one output

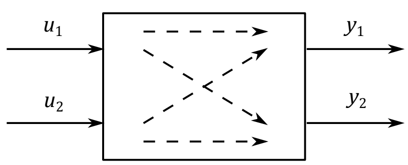


- m - inputs and outputs: together m controllers
- $m!$ - possible pairs

1. Choose pair with minimal impact on each other.
2. Tune the controllers.

Problem: however, loops are connected to each other.

Example 11 *{Glad, Ljung} Problems*



Identification gives transfer:

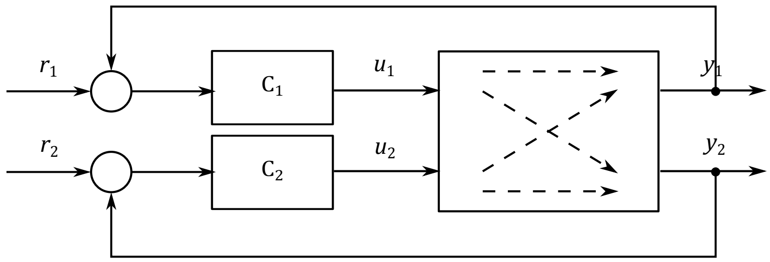
$$\begin{cases} y_1 = \dfrac{2}{s+1}u_1 + \dfrac{3}{s+2}u_2 \\ y_2 = \dfrac{1}{s+1}u_1 + \dfrac{1}{s+2}u_2 \end{cases}$$

Inputs and outputs, gains and time constants.
Step response.

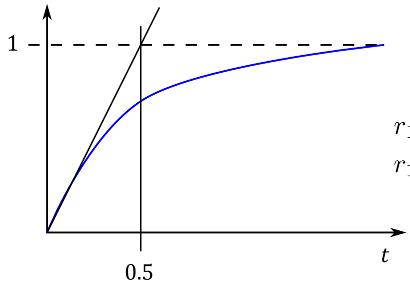
Let choose pairs (u_1, y_1), (u_2, y_2) and connect controllers C_1, C_2
Tune the PI controllers

$$(u_2 = 0) \rightarrow u_1 = \frac{K_1(s+1)}{s}(r_1 - y_1) \quad W_1 = \frac{2K_1}{s+2K_1} = \frac{2}{s+2}, \quad (K_1 = 1)$$

$$(u_1 = 0) \rightarrow u_2 = \frac{K_2(s+1)}{s}(r_2 - y_2) \quad W_2 = \frac{K_2}{s+K_2} = \frac{2}{s+2}, \quad (K_2 = 2)$$



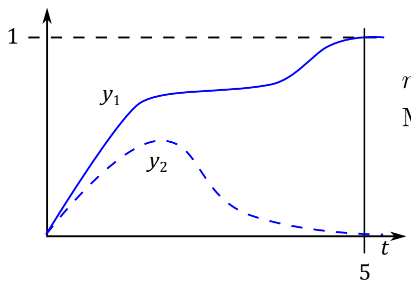
Step response:



- Separate work of the controllers

$r_1 = 1, \quad r_2 = 0 \ \rightarrow \ y_1$

$r_1 = 0, \quad r_2 = 1 \ \rightarrow \ y_2$



- Work of the controllers together

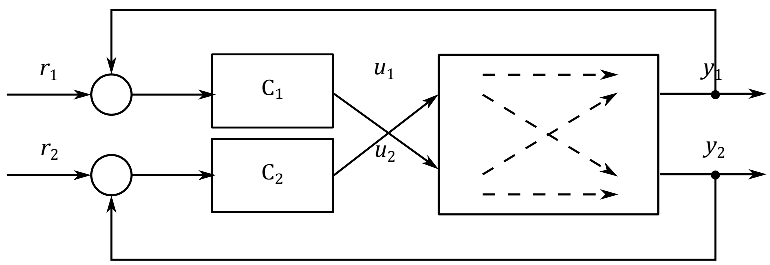
$r_1 = 1, \quad r_2 = 1 \ \rightarrow \ y_1, y_2$

More slow! 0.5 s $\rightarrow$ 5 s

Because gain is increased:

$K_1 = 4, \ K_2 = 8 \ \rightarrow$ unstable! (Separately controllers are stable.)

Change the pairs: inputs-outputs



Even greater difficulties:

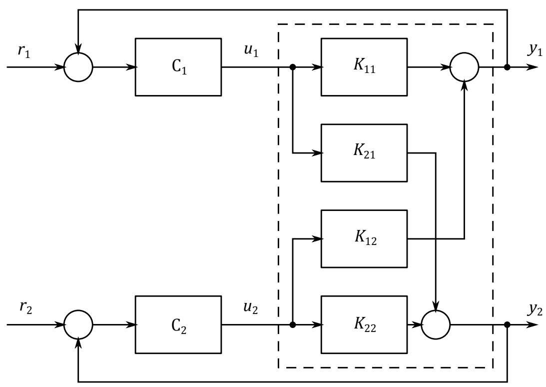
- One of the controller's gain $-K$

  Changes the controller into unstable. Controller cannot be disconnected!

Reason: MIMO tranfer function has RHZ, (however any of separate transfers do not have it by itself)

Input-output dynamic behavior [1]

Second order system ($n = 2$) has 3 loops $r_i \rightarrow y_i$ $(i = 1, 2)$



1. $C_1 \cdot K_{11}$
2. $C_2 \cdot K_{22}$
3. $C_1 \cdot K_{21} \cdot C_2 \cdot K_{12}$

$n = 3$ - 6 loops
Connections affect system work and stability

How to measure (characterize) loops relationship?
How to choose the input-output pairs for SISO regulator?

Sometimes it is clear, sometimes it is not.

A static gain matrix A, $[K_{ij}]$, $Y = A \cdot U$

$$\begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} K_{11} & \dots & K_{1n} \\ \dots & \dots & \dots \\ K_{n1} & \dots & K_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ \dots \\ u_n \end{bmatrix}$$

which describes the effect of input to output, does not suit well.

The relative Gain array [1]

Heuristic technique for possible interactive affects between several SISO loops proposed by Bristol. This approach requires only steady-state information and provides

1. A measure of process interactions.
2. A recommendations for the most effective pairing of the controlled and manipulated variables.

Relative gain λ_{ij} between input j and output i

$$\lambda_{ij} = \frac{\left(\frac{\partial y_i}{\partial u_j}\right)_{u_k \quad k\neq j} \qquad \text{open loop}}{\left(\frac{\partial y_i}{\partial u_j}\right)_{y_k \quad k\neq i} \qquad \text{closed loop}}$$

The RGA is a matrix that contains individual relative gain as elements

$$\text{RGA(A)}=\lambda_{ij}$$

In Matlab: `rga = K.*(inv(K)')`

Properties of RGA:

- Each row and each column sums to 1.0;
- Without dimension, do not depend on the units choice;
- Sensible to the errors of matrix K;
- If $K_{ij} = 0$ then $\lambda_{ij} = 0$.

How to choose pairs of inputs and outputs

$\lambda < 0$	avoid such pairs
$0 < \lambda < 0.67$	works badly, decompose the system
$\lambda = 1$	the best solution, no effect from other loops
$0.67 < \lambda < 1.5$	realizable, insignificant effect by other loops
$2 < \lambda < 10$	dynamic decomposition is needed
$25 < \lambda$	works badly

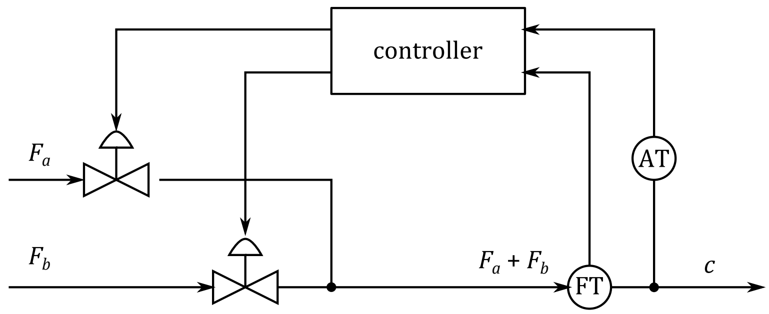
Example 12 *Blending system [1]*

To develop the control structure RGA is needed.

1. Steady-state model
2. Linearize it to find process gain K_i
3. Calculate the RGA on order to decide on variable pairing

The total blend flow rate is equal to the sum of two feed streams and the concentration are described as

$$\begin{aligned} F &= F_a + F_b \\ c &= c(F_a, F_b) = F_a/(F_a + F_b) \qquad \text{is nonlinear!} \end{aligned}$$



Linearization $c = c_0 + \Delta c,\ F_a = F_{a_o} + \Delta F_a,\ F_b = F_{b_0} + \Delta F_b$ into the form $\Delta c = K_1 \cdot \Delta F_a + K_2 \cdot \Delta F_b$

$$\Delta c = \frac{F_{b_0}}{(F_{a_0} + F_{b_0})^2}\Delta F_a + \frac{-F_{a_0}}{(F_{a_0} + F_{b_0})^2}\Delta F_b$$

process gain matrix is

$$\begin{bmatrix} c \\ F \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} F_a \\ F_b \end{bmatrix}$$

Do we have a non-zero determinant?

How to choose the control pairs?

- $c_0 = 5\% = 0.05$
- F_a - concentrate
- F_b - water

So $k_1 = 0.95$ and $k_2 = -0.05$

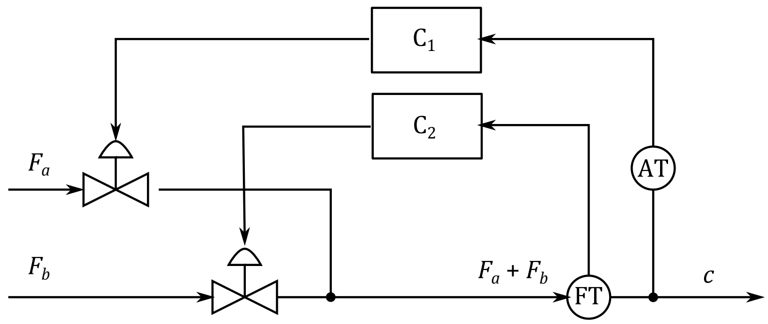
$$\texttt{rga=K.*(inv(K)')} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix},$$

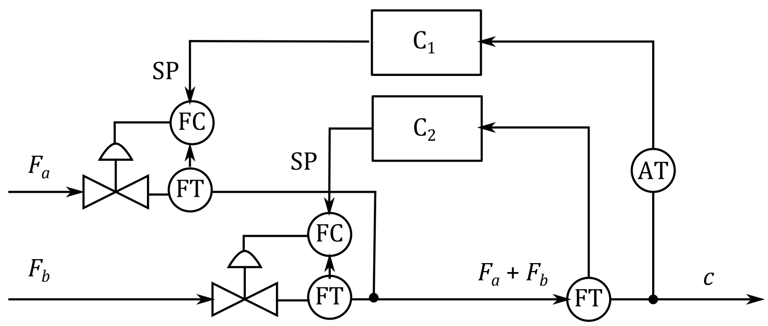
which indicates that $1 \to 1$ or blend concentration should be paired with concentrate flow $c \to F_a$ and $2 \to 2$ or blend total flow should be paired with water flow $F \to F_b$.

For the control of the flows F_a, F_b cascade control should be used.

If setpoint is changed for example $c = 0.95$, then pairs should be changed $c \to F_b$ and $F \to F_a$.

If the process interactions are significant, even the best multiloop control system cannot provide satisfactory control. In this situations we need to use multivariable control strategies such as decoupling or model predictive control.





5.2 Decoupling

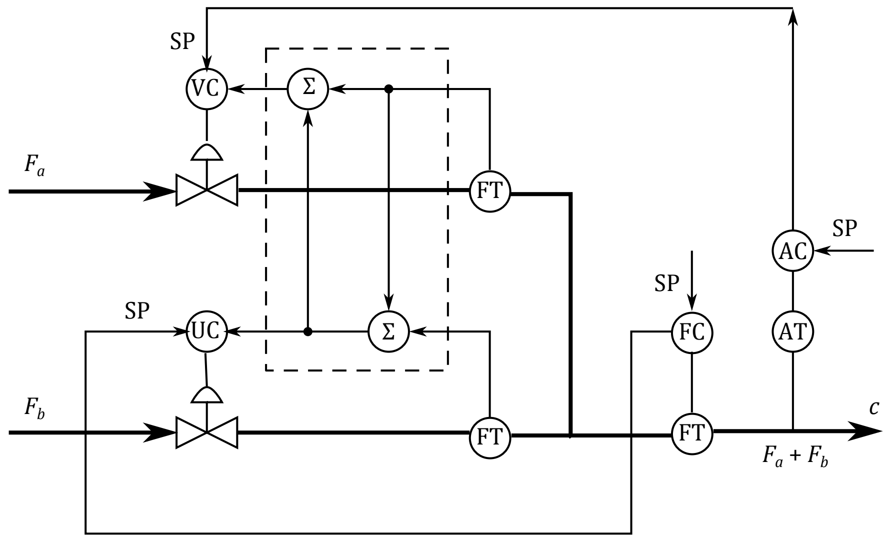
The idea of decoupling is to develop "synthetic" manipulated inputs that affect only one process output each [1]. That means in order to eliminate interactions we need achieve a diagonal transfer function matrix.

- Linear transformations or
- Change of the variables $V = F_a/(F_a + F_b), U = F_a + F_b$.

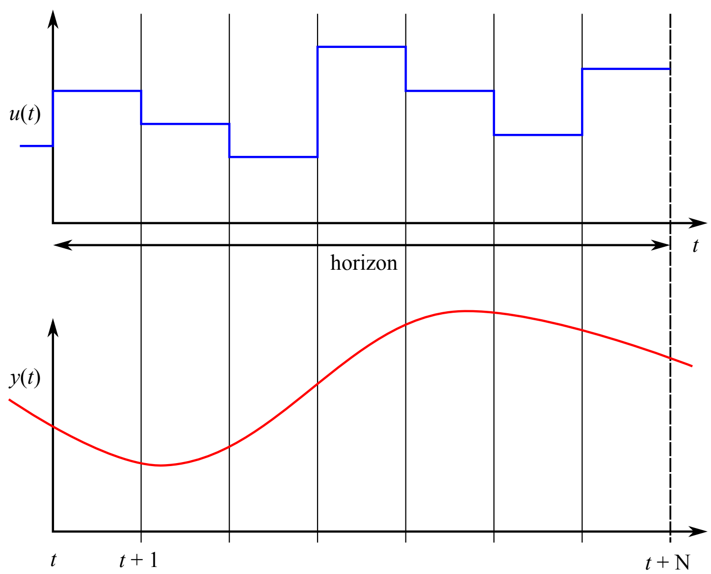
Redefining variables: sum, difference, ratio, etc.

Advantage	**Disadvantage**
independent SISO tuning parameters for each control loop	decoupler is inverse of the process transfer matrix
	decoupler will be unstable if RHP are present
	decopler extremely sensitive to model error

In practice, static decoupling instead of dynamic decoupling is used much more often. This is due to the fact that some dynamic versions may be not physically realizable or some dynamic properties are uncertain. In that case in decoupling equations set $s = 0$.



5.3 Model predictive Control (MPC)



Discrete time application. At each time step k optimization problem is solved. Objective function based on output predictions over a prediction horizon. Most MPC methods are based on step or impulse response models.

Dynamic matrix control (DMC)

In DMC the free response is the effect of the past control actions and the additive correction term, while forced response is the effect of the current and the future control moves.

1. Develop a discrete model with length N
2. Specify the prediction P and control M horizons. $N \geq P \geq M$
3. Specify the weighting on the control action.
4. All calculations assume deviation variable form.

Model length $N \approx t_{set}$ settling time to reach new steady state.
If prediction horizon is much longer than the control horizon, the control is less sensitive to model error.
If prediction horizon is much longer than control horizon control weighting often set to zero.

Bibliography

[1] B. Wayne Bequette, *Process Control: Modeling, Design, and Simulation*. Prentice Hall Professional, 2003.

[2] Don Green and Robert Perry, *Perry's Chemical Engineers' Handbook*, The McGraw-Hill Companies, Inc, 8-th Edition, 2008.

[3] Tony R. Kuphaldt, *Lessons In Industrial Instrumentation*. URL:`http://www.pacontrol.com/industrial-instrumentation.html`, 2012.

[4] Jonathan Love, *Process Automation Handbook: A Guide to Theory and Practice*. Springer-Verlag London Limited, 2007.

[5] John A. Shaw, *The PID Control Algorithm How it works, how to tune it, and how to use it*, Process Control Solutions, 2nd Edition, 2006.