Part I

OVERVIEW OF PROCESS CONTROL

Contents

I OVERVIEW OF PROCESS CONTROL 1

1 BAISIC LOW-LEVEL CONTROL 4

1.1 BASIC IDEA OF FEEDBACK/FEEDFORWARD CONTROL 4

1.2 MOTIVATION- WHY(NEGATIVE) FEEDBACK CONTROL ? 9

1.3 ELEMENTS OF A FEEDBACK SYSTEM 13

1.4 PID FORMULATIONS 20

1.4.1 INTEGRAL(I) CONTROL 20

1.4.2 PROPORTIONAL(P) CONTROL 21

1.4.3 PROPORTIONAL INTEGRAL(PI) CONTROL 24

1.4.4 PROPORTIONAL INTEGRAL DERIVATIVE(PID) CONTROL . . 25

1.4.5 DIGITAL IMPLEMENTATION 27

1.5 QUANTITIVE PID TUNING METHODS 29

1.5.1 CONTINUOUS CYCLING METHOD 29

1.5.2 REACTION-CURVE-BASED METHOD 38

1.5.3 FOPDT-BASED TUNING RULES 41

1.5.4 DIRECT SYNTHESIS METHOD - IMC TUNING 46

1.6 PRACTICAL CONSIDERATIONS 50

2 MULTI-LOOP CONTROL AND FURTHER PRACTICAL ISSUES 55

2.1 FEEDFORWARD-FEEDBACK CONTROL 55

2.2 CASCADE CONTROL 57

2.3 OVERRIDE CONTROL 61

3 CONTROL OF MULTI-INPUT MULTI-OUTPUT(MIMO) PROCESSES 62

3.1 MIMO PROCESS ? 62

3.2 INTERACTION AND I/O PAIRING 64

3.2.1 INTERACTION 64

3.2.2 I/O PAIRING 67

3.3 DECOUPLING 70

3.4 CURRENT TRENDS 74

3.4.1 COMPUTER INTEGRATED PROCESS MANAGEMENT 74

3.4.2 MODEL-BASED APPROACHES 75

3.4.3 COMPUTING ENVIRONMENT 77

3.4.4 COMPUTER CONTROL SYSTEM 78

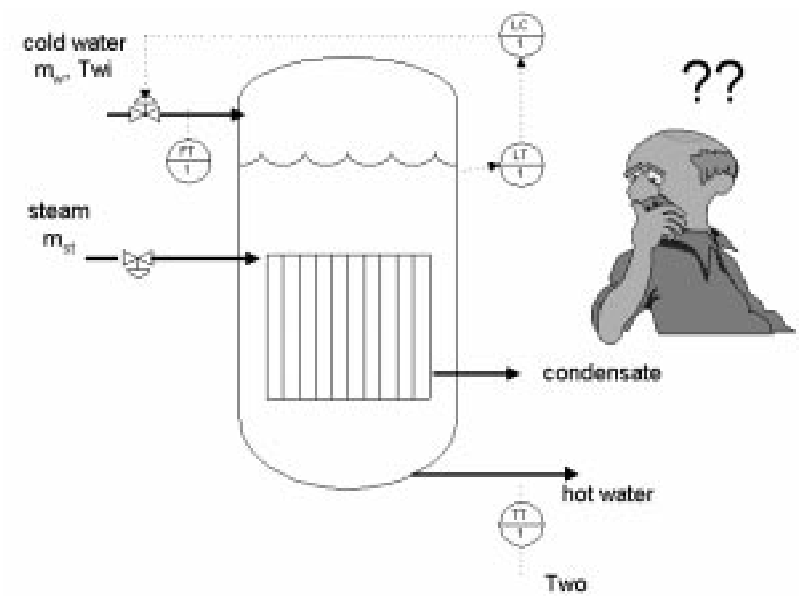
3.4.5 SMART INSTRUMENT AND FIELD BUS 79

Chapter 1

BAISIC LOW-LEVEL CONTROL

1.1 BASIC IDEA OF FEEDBACK/FEEDFORWARD CONTROL

Example Process



Control Objective keep the water outlet temp(T_{wo}) at a desired set point(T_{sp}).

Major Disturbance hot water demand(m_w)

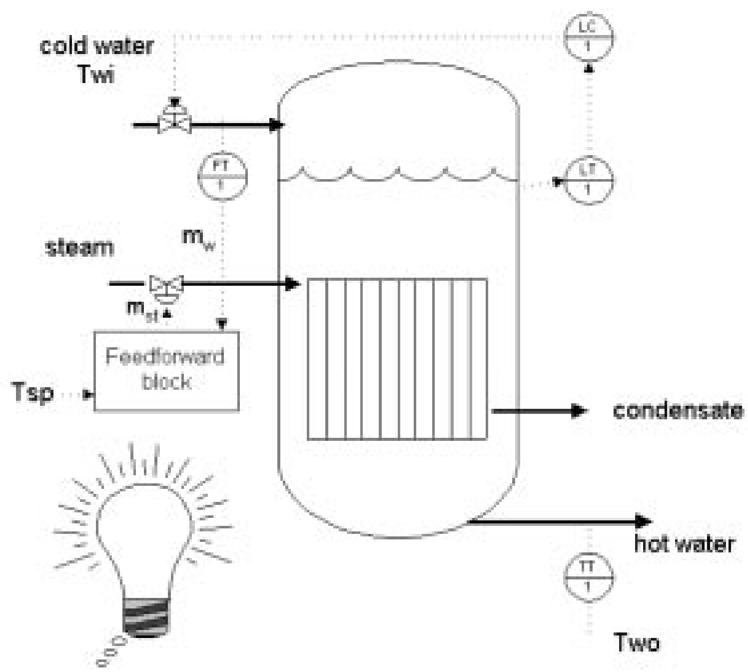
Manipulated Input steam flow rate(m_{st})

◇ **APPROACH 1 :** FEEDFORWARD CONTROL

- Measure m_w
- Calculate m_{st} needed to maintain T_{wo} for given m_w. For example,

$$\lambda_{st} m_{st} = m_w Cp_w (T_{wo} - T_{wi}) \;\rightarrow\; m_{st} = \frac{m_w Cp_w (T_{sp} - T_{wi})}{\lambda_{st}}$$

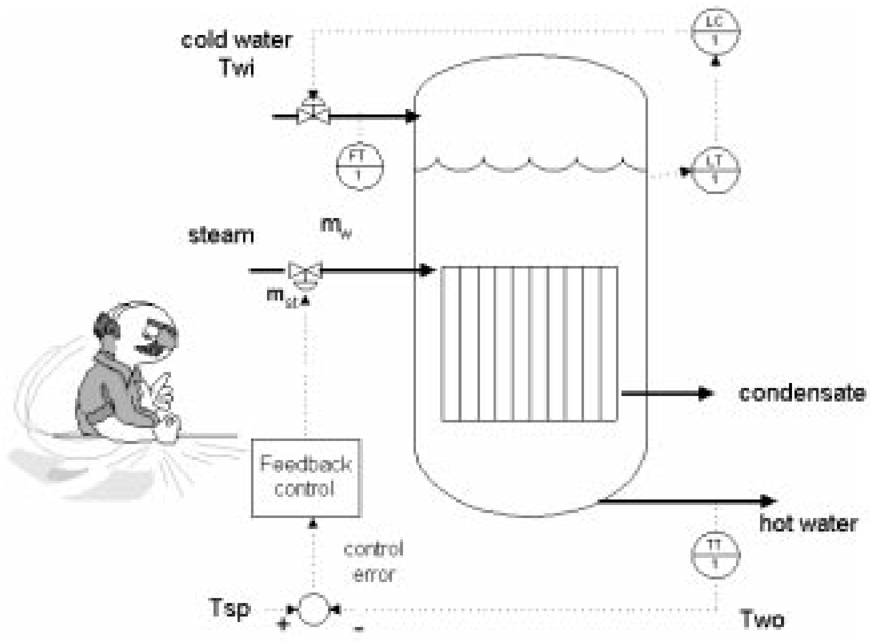
- Apply m_{st}.



No guarantee that $T_{wo} \to T_{sp}$ because of various *uncertainties* which come from the model description, measurements, and signal implemetation, etc.

◇ **APPROACH 2 :** FEEDBACK CONTROL

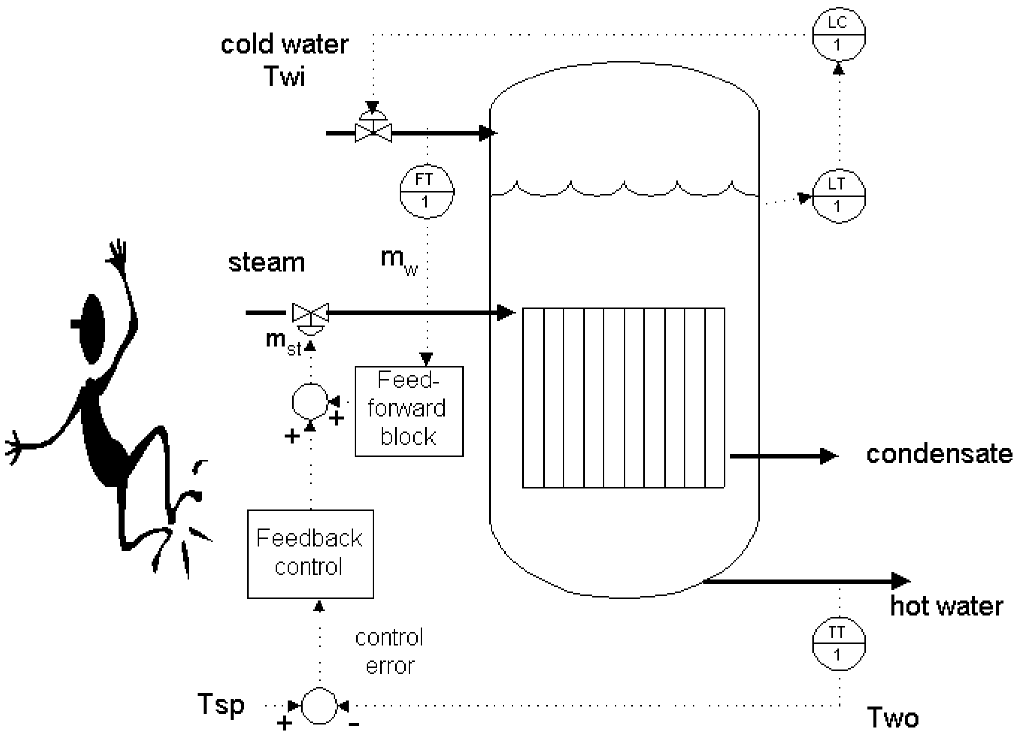
- Measure T_{wo}
- Compare with T_{sp}
- Take approriate action to elimiate the observed error.



T_{wo} can be steered to T_{sp}. But T_{wo} may undergo a long transient period, frequently with oscillation, due to the trial and error nature of the feedback action. If the feedback controller is designed based on a *process model*, however, the transient can be adjusted as desired (with some limitations).

◇ **APPROACH 3 :** FEEDFOWARD-FEEDBACK CONTROL

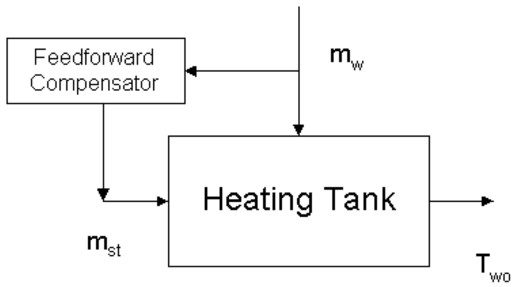
- Apply m_{st} from the feedforward block to the process.
- Provide additional corrective signal through feedback control when there is control error.



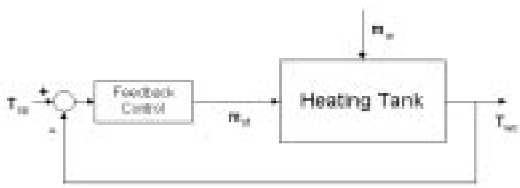
Better control performance can be expected than with feedback or feedforward-only control.

Block Diagram Representations

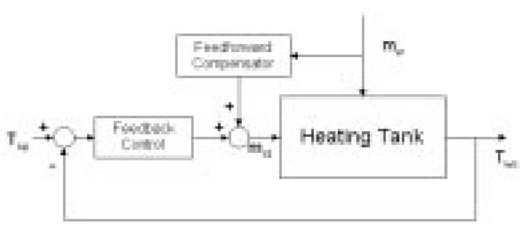
Feedforward Control



Feedback Control

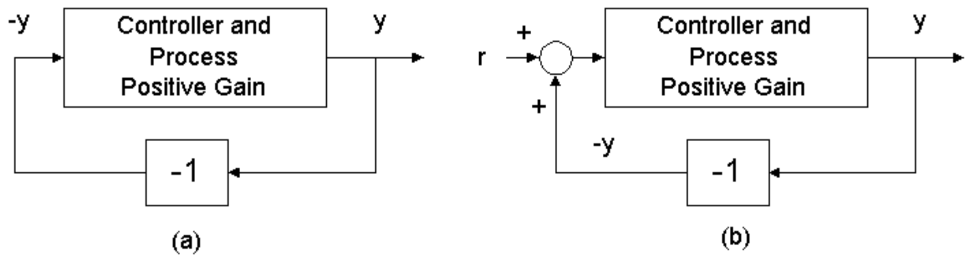


Feedforward-Feedback Control



1.2 MOTIVATION- WHY(NEGATIVE) FEEDBACK CONTROL ?

Some Comments on Negative Feedback



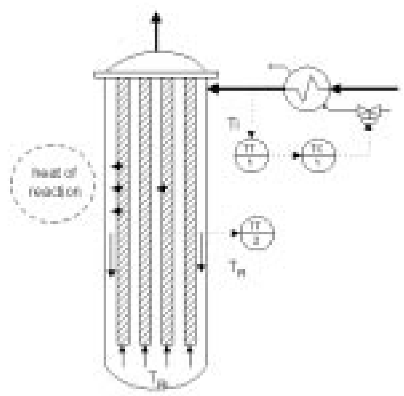
- In (a), as y deviates from zero, it is pulled back toward zero. Hence, negative feedback has self-stabilizing tendency.
- If an external r is put as in (b), y will tend to r.
- When the positive signal is fed back (positive feedback), it adds to itself and will tend to diverge.

What Can We Gain Through Feedback Control in Process Control Problems ?

1. **To steer the process variables to desired steady states**
   - Even when there is no disturbance, it is hard to manulally drive PVs to desired states.
   - With the aid of *integral action*, the feedback controller continues the corrective action until PVs reach their respective SPs.
2. **Disturbance Rejection**
   - Try to keep the PVs at their SPs against various disturbances.

3. **Stabilization**

- Some processes are intrinsically unstable.



Autothermal Reactor

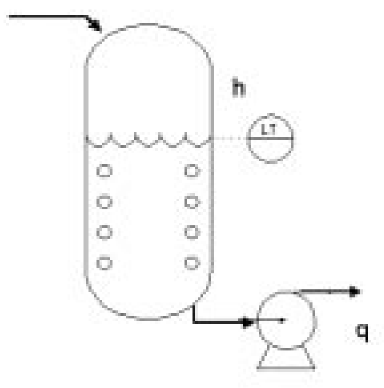
(a) When T_i is perturbed to increase, T_{Ri} is increased.

(b) This accelerates reaction rate and induces more heat of reaction.

(c) This again increases T_{Ri} and boosts up (b).

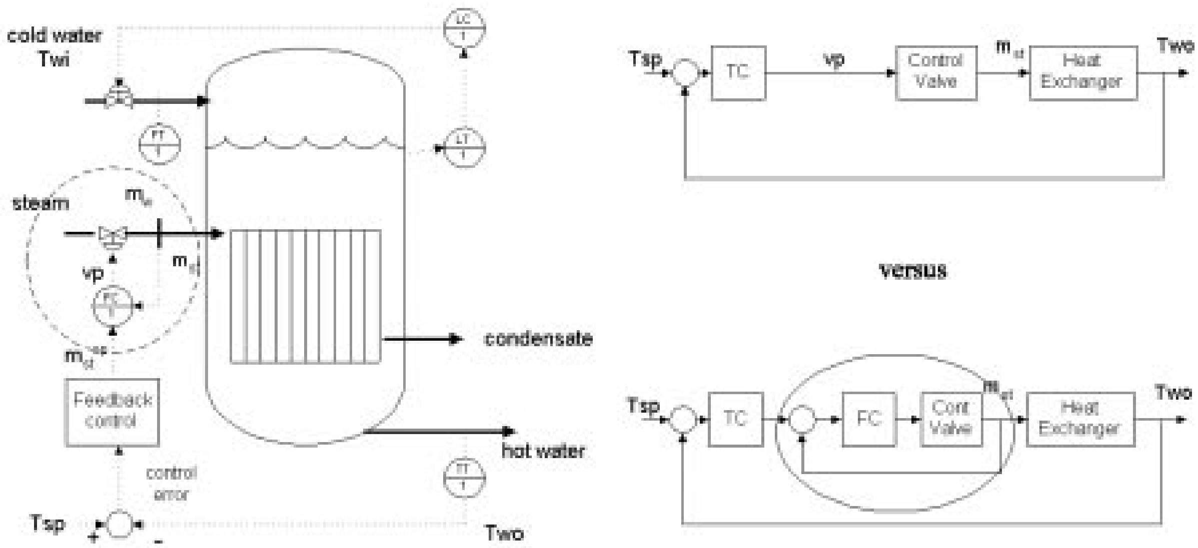
Positive feedback path exists between T_R and T_{Ri}.

- There are processes with integrating dynamics.



Surge Tank - Level is not self-stabilizable.

4. **Linearization**



- Usually, m_{st}(steam flow rate) changes nonlinearly with vp(valve position).
- When TC is configured to directly manipluate vp, the process seen by TC includes a nonlinear control valve block.
- In the cascade configuration, $m_{st} \approx m_{st}^{sp}$ if the slave controller is tightly tuned. Thus, the nonlinearity by the control valve block can be removed.

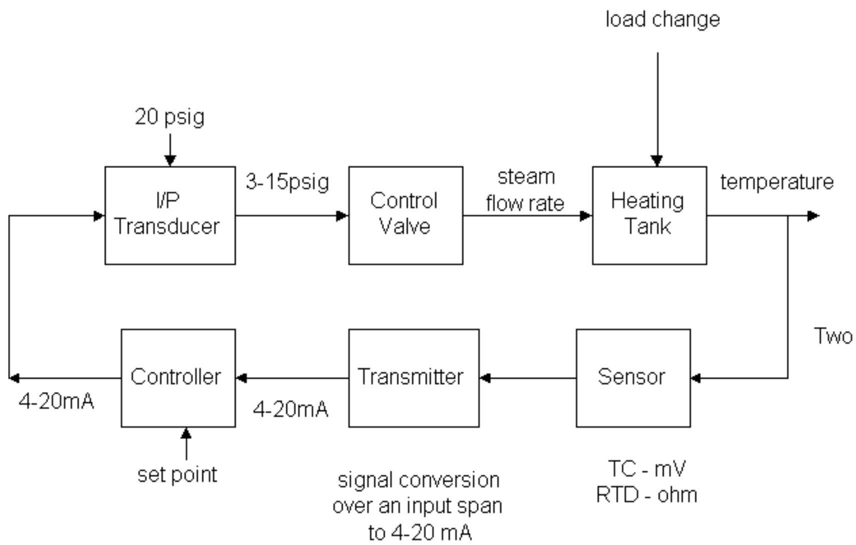
5. **Improving Dynamics**

- The dynamics of the slave loop in the cascade configuration can be adjusted to have a faster response than the control valve block has.
- Suppose T_{wo} is to be changed to another value. If the change in m_{st} is made manually, the settling time is set by the intrinsic process

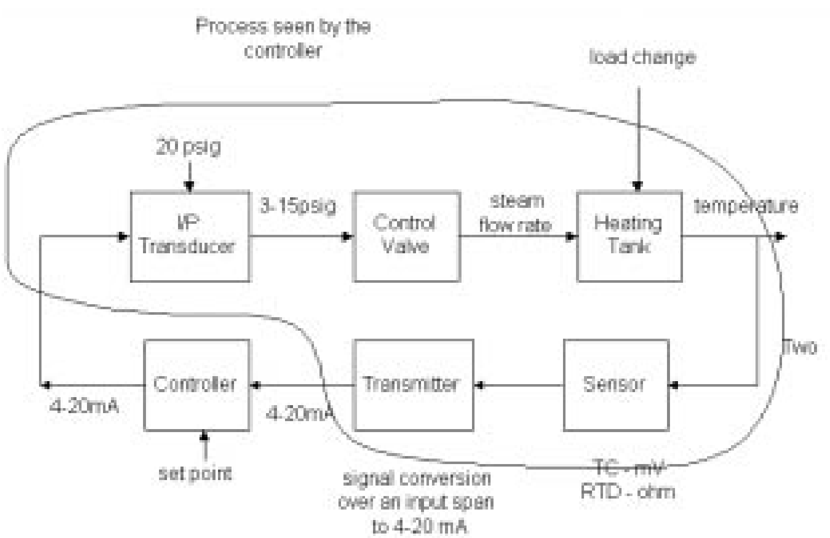
time constant. On the other hand, putting a feedback loop (TC) can speed up the response time.

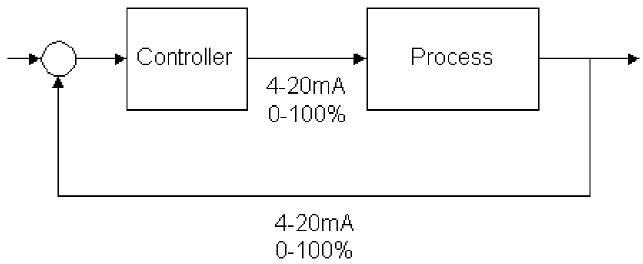
- Thus, TC in the cascade configuration controls a faster process than the one in the direct configuration.

1.3 ELEMENTS OF A FEEDBACK SYSTEM



Process seen by the Controller





Transmitter

- Conditioning the sensor signal
  - Ohm, mv, mA $\cdots$ to voltage
  - linearization
  - conversion of the voltage signal to 4-20 mA(0-100 %) standard signal.
- To transmit a signal over a long distance while minimizing corruption with electro-magnetic noise, the signal needs to be converted into a current form.
- 4 mA(0%) bias is for detection of a sensor failure.

I/P (Current to Pneumatic Pressure) Transducer

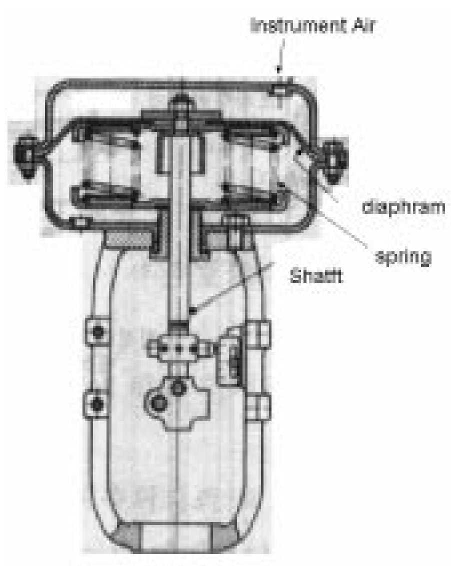
- Linearly convert 4-20 mA to 3-15 psig(0.2-1.0 Kgf/cm2) air pressure
- A separate device or embedded in a valve positioner.

Control Valve

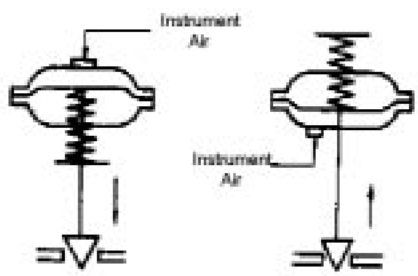
- Mostly pneumatic for safety reason

- Composed of an actuator and a valve body

- Actuator
  - ATO(air to open) and ATC types. Sometimes called NC(normally close)/NO or FC(fail close)/FO
  - Selection is based on what the fail-safe position is.
  - Depending the actuator type, sign of the process gain is reversed.



Mult-Spring Type Air-To-Close Actuator



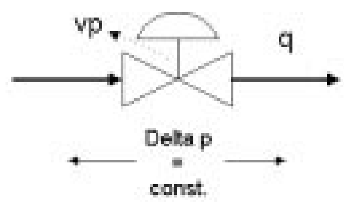
Air-to-Close Type Air-to-Open Type

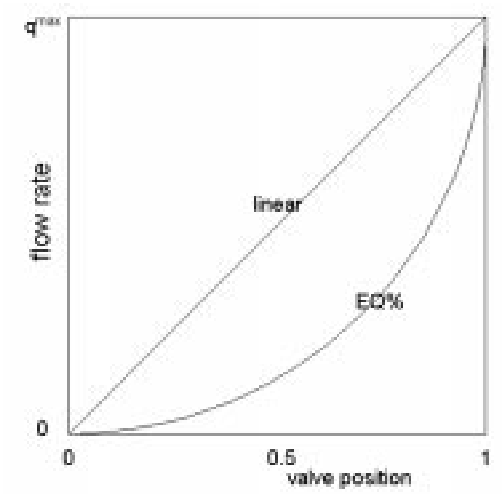
- Valve Body
  - According to intrinsic flow characteristics (Linear/Equal Percentage), leakage(single seated/double seated), critical flow characteristics, noise, etc, many different types of valves are designed.
  - Flow characteristics

$$q(\text{gal/min}) = C_v \times f(vp) \times \sqrt{\frac{\Delta P_v(\text{psig})}{\text{sp.gr}}}$$

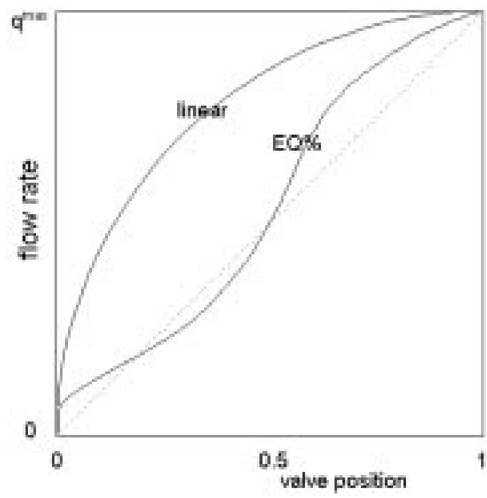
$$f(vp) = \begin{cases} vp & \Rightarrow \text{ linear} \\ \sqrt{vp} & \Rightarrow \text{ quick opening} \\ R^{vp-1} & \Rightarrow \text{ equal percentage} \end{cases}$$

Intrinsic flow characteristic





Installed flow characteristic

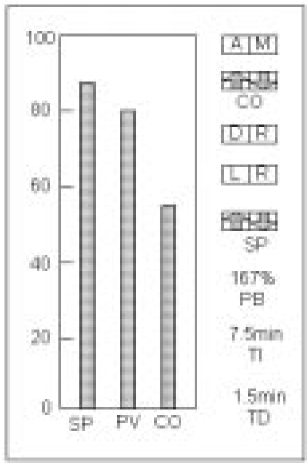


As q increases, Δp decreases.
A properly sized EQ-valve can show approximately linearized flow characteristics.

– Valve Size: C_v = [gallon water]/[min][psig]
Choose it to cover the needed flow range (with some additional room). A larger valve covers a larger range and reduces pumping cost, but tends to be more nonlinear and sensitive to pressure drop changes.

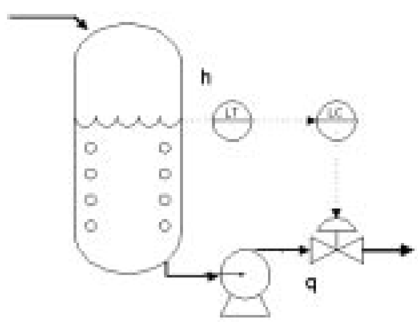
Controller

- Appearance



- A/M : Auto/Manual mode selection for CO adjustment

- D/R : Direct/Reverse mode selection
  Direct (Reverse) - CO increases(decreases) when PV increases. Negative (Positive) gain control.



ATO valve, select DIRECT.
ATC valve, select REVERSE.

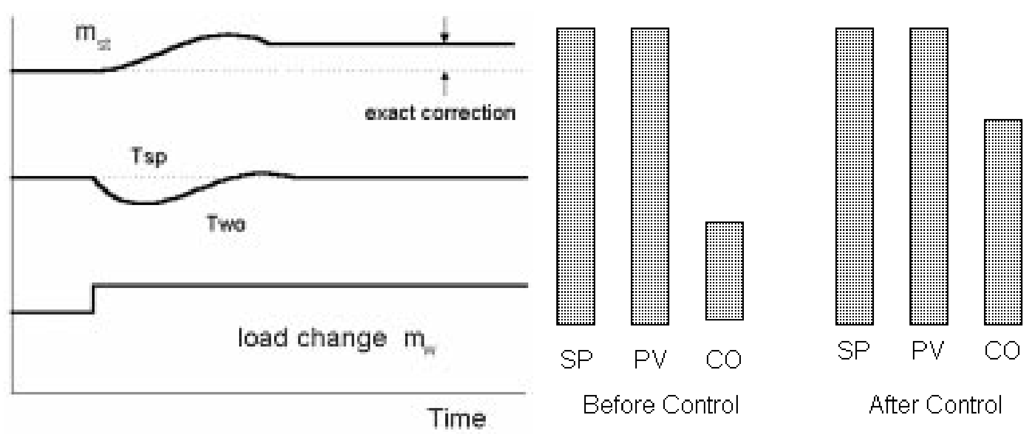
- L/R : Local SP/Remote SP selection
- PB, TI, TD : PID parameters
- In computer control, engineering units can be used for SP and PV instead of 0-100%.

1.4 PID FORMULATIONS

1.4.1 INTEGRAL(I) CONTROL

$$u(t) = u_{bias} + \frac{1}{T_I}\int_0^t e(\tau)d\tau = u_{bias} + \frac{1}{T_I}\int_0^t \left(r(\tau) - y(\tau)\right) d\tau$$

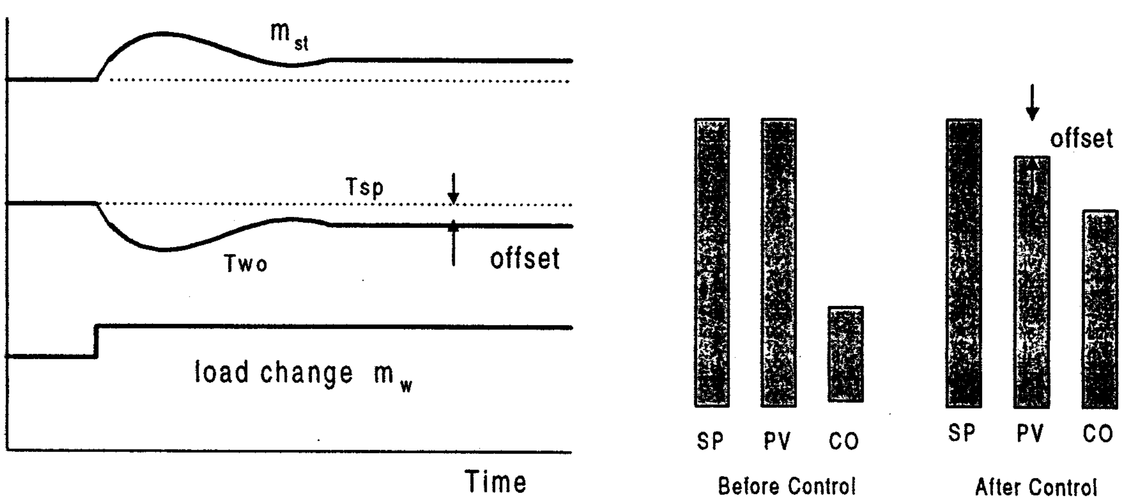
- u_{bias} is the manually set CO value before switching to AUTO mode.
- $t = 0$ represents the instant the AUTO mode is started.
- Corrective action based on accumulated control error.
- If control error persists, $u(t)$ continues to increase (decrease) until $e(t)$ is settled at zero.
- The I-mode ultimately finds a new u_{bias} for exact correction. In this sense, the I-mode is often called the RESET mode.
- When switching AUTO to MANUAL, $u(t)$ replaces u_{bias}. (Bumpless transfer)



1.4.2 PROPORTIONAL(P) CONTROL

$$u(t) = u_{bias} + K_c e(t) = u_{bias} + K_c\left(r(t) - y(t)\right)$$

- Corrective action proportional to the current control error. $\rightarrow$ Faster control action than I-mode



- For a load and/or set point change, exact correction cannot be made because u_{bias} is not reset. $\rightarrow$ OFFSET
- K_c : Proportional Gain (%/% or may have a Physical Unit depending on controllers)
- REVERSE mode $\leftrightarrow$ $K_C > 0$
  DIRECT mode $\leftrightarrow$ $K_C < 0$
- In industrial controllers, PB (Proportional Band) is dominantly used instead of K_c.

–

$$\mathrm{PB}(\%) = \frac{100}{K_c}, \qquad \Delta u(t) = \frac{100}{\mathrm{PB}}\, e(t)$$

PB represents the percent change of control error which induces 100% movement of CO.

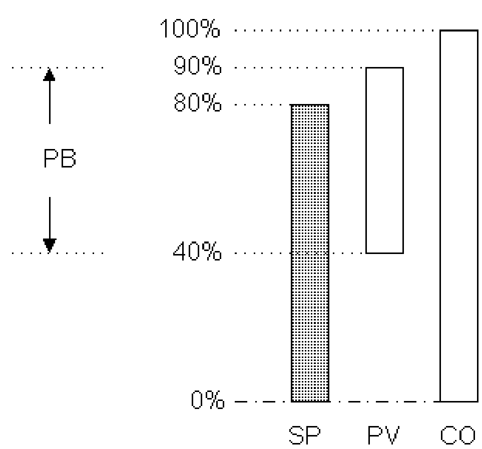
– Note that $0\% \le u(t), y(t) \le 100\%$.

Depending on r and u_{bias} settings, the controllable range of $y(t)$ which induces the maximum travel of CO may be different from PB.

The controllable range of $y(t)$ is not necessarily centered with respect to r.

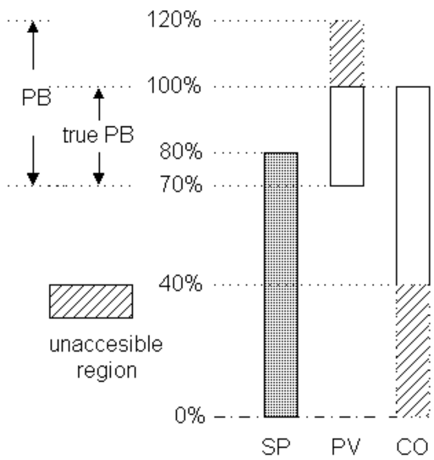
Example. 1 $\quad u(t) = 20 + (100/\mathrm{PB})(80 - y(t)), \quad \mathrm{PB} = 50\%$

$$\begin{aligned} \text{At } y(t) &= 90\%, u(t) = 0\% \\ \text{At } y(t) &= 40\%, u(t) = 100\% \end{aligned}$$



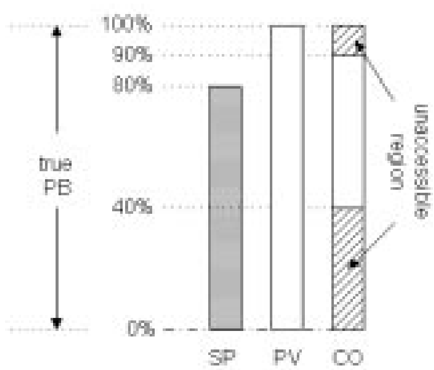
Example. 2 $u(t) = 80 + (100/\text{PB})(80 - y(t)), \quad \text{PB} = 50\%$

$$\begin{aligned} \text{At } y(t) &= 70\%, u(t) = 100\% \\ \text{At } y(t) &= 100\%, u(t) = 40\% \end{aligned}$$



Example. 3 $u(t) = 80 + (100/\text{PB})(20 - y(t)), \quad \text{PB} = 200\%$

$$\begin{aligned} \text{At } y(t) &= 0\%, u(t) = 90\% \\ \text{At } y(t) &= 100\%, u(t) = 40\% \end{aligned}$$

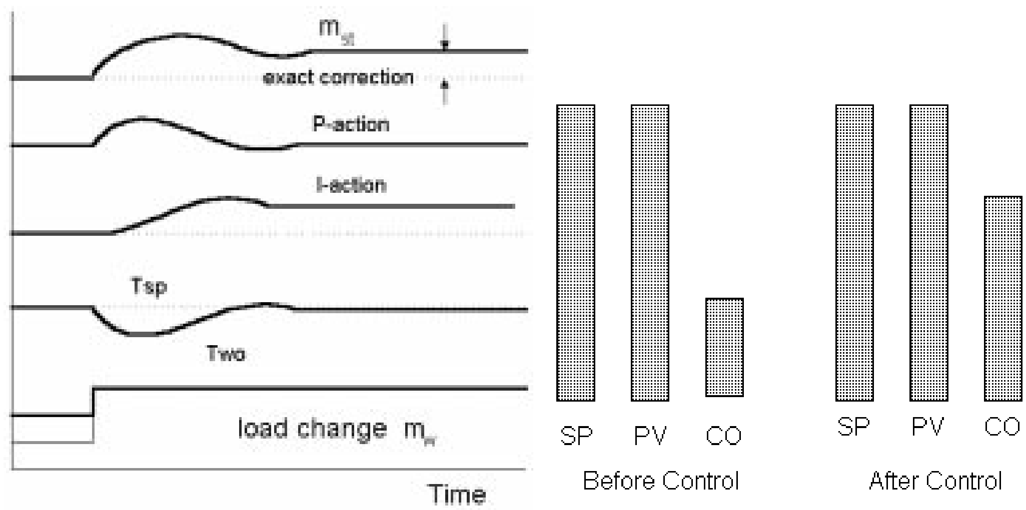


- The process is in control state only when PV is within the true PB. Outside the PB, CO is staturated (or PV itself is saturated.)

1.4.3 PROPORTIONAL INTEGRAL(PI) CONTROL

$$u(t) = u_{bias} + K_c \left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau \right]$$

- Immediate action by P-mode PLUS offset elimination by I-mode



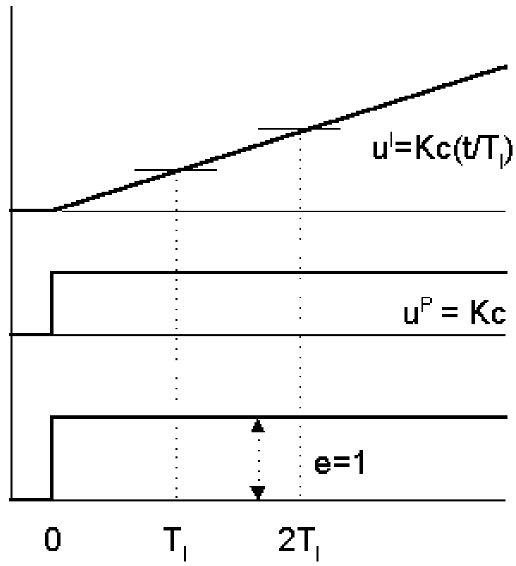
- T_I : Integral time (min/repeat)

Consider the situation where $e(t) = 1$ persists.

$$\begin{aligned} u^P(t) &= K_c e(t) = K_c \\ u^I(t) &= \frac{K_c}{T_I}\int_0^t e(\tau)d\tau = K_c\frac{t}{T_I} \end{aligned}$$

$$\Rightarrow \qquad u^I(T_I) = K_c = u^P, \quad u^I(2T_I) = 2K_c = 2u^P, \cdots$$

I-mode *repeats* u^P every T_I minutes.



- Sometimes, Reset Rate R_I defined by R_I(repeat/min) $= 1/T_I$ is used in industrial controllers.

1.4.4 PROPORTIONAL INTEGRAL DERIVATIVE(PID) CONTROL

$$u(t) = u_{bias} + K_c \left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau + T_D \frac{de(t)}{dt} \right]$$

- $de(t)/dt$ detects how fast the error increases(or decreases).
- D-mode can much imporve the control performance by anticipating the future trend of control error (in a linear fashion) and taking a compensating action in advance.

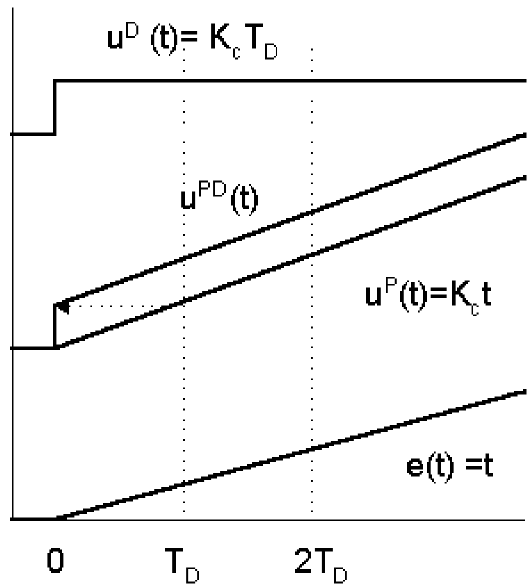
- T_D: Derivative time (min)

  Consider the situation where $e(t) = t$ (linearly increases).

$$\begin{aligned} u^P(t) &= K_c e(t) = K_c t \\ u^D(t) &= K_c T_D \frac{de(t)}{dt} = K_c T_D \end{aligned}$$

$$\Rightarrow \quad u^D(t) = u^P(T_D)$$

  or $u^{PD}(t)$ acts T_D(min) ahead of $u^P(t)$.



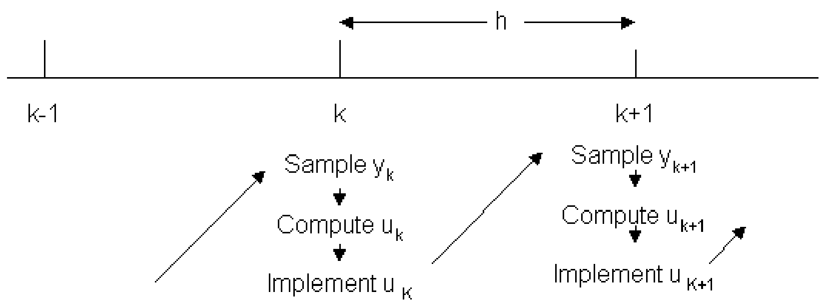
  In this sense, D-mode is sometimes called a *preact mode* and T_D is called the preact time.

- D-mode can also serve to reduce oscillation, thereby allowing use of higher gains.

- D-mode is sensitive to high frequency noise in the control error. Because of this trouble, not employed for flow rate, pressure, and level control where the measurements usually have significant process noise. Mostly used for temperature control.

1.4.5 DIGITAL IMPLEMENTATION

Digital control is run according to the following sequence:



Position Form Digital PID Algorithm

At the instance $t = kh$,

$$
\begin{aligned}
e(t) &= e_k \\
\int_0^t e(\tau)d\tau &\approx \sum_{i=1}^{k} e_i h \\
\frac{de(t)}{dt} &\approx \frac{e_k - e_{k-1}}{h}
\end{aligned}
$$

Using the above approximations, we can derive a digital PID control algorithm

$$
u_k = u_{bias} + K_c \left(e_k + \frac{h}{T_I} \sum_{i=1}^{k} e_i + \frac{T_D}{h} [e_k - e_{k-1}] \right)
$$

- The above is called the *position form* because the positional value of u_k is directly computed.
- Required to store e_k and $\sum_{i=1}^{k} e_i$ in the memory.
- Inconvenient to handle the *reset windup* problem.

Velocity Form Digital PID Algorithm

The position from algorithm at $t = (k-1)h$ is

$$u_{k-1} = u_{bias} + K_c \left(e_{k-1} + \frac{h}{T_I} \sum_{i=1}^{k-1} e_i + \frac{T_D}{h} [e_{k-1} - e_{k-2}] \right)$$

Subtracting this from u_k yields

$$u_k = u_{k-1} + K_c \left([e_k - e_{k-1}] + \frac{h}{T_I} e_k + \frac{T_D}{h} [e_k - 2e_{k-1} + e_{k-2}] \right)$$

- It is called the *velocity form* algorithm since only $\Delta u_k = u_k - u_{k-1}$ is computed at each sampling instant.
- Can effectively and conveniently handle the *reset windup* problem.

- As $h \to 0$, digital PID $\to$ continuous PID.

In single loop controllers or local control units of DCS, 100-500 msec of h is usually used ($\approx$ continuous controller)

In computer implementations, $h \approx$ (time constant)/(10∼20) is sufficient for most applications.

1.5 QUANTITIVE PID TUNING METHODS

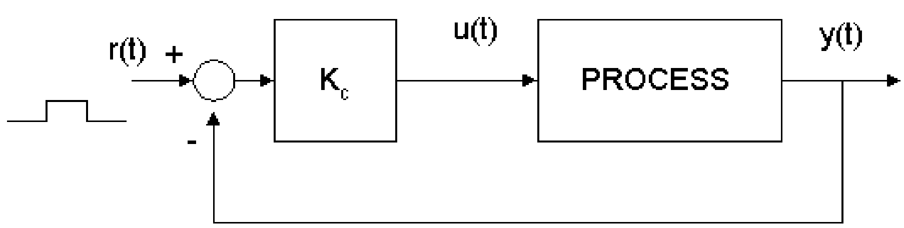
- Tuning PID parameters is not a trivial task in general.
- Various tuning methods have been proposed for different model descriptions and performance criteria.

1.5.1 CONTINUOUS CYCLING METHOD

Frequently called Ziegler-Nichols method since it was first proposed by Ziegler and Nichols (1942). Also referred to as loop tuning or the ultimate sensitivity method.

Procedure:

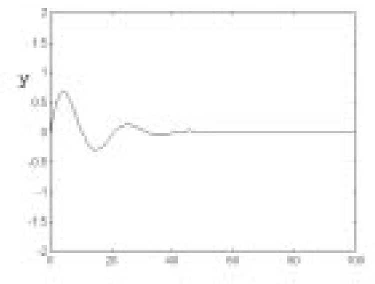
step1 Under P-control, set K_c at a low value. Be sure to choose the right (direct/reverse) mode.



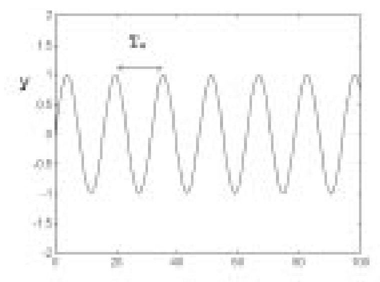
step 2 Increase K_c slowly and monitor $y(t)$ whether it shows oscillating response.
If $y(t)$ does not respond to K_c change, apply a short period of small pulse input on $r(t)$.

step 3 Increase K_c until $y(t)$ shows continuous cycling. Let K_u be K_c at this condition. Also let T_u be the period of oscillation under this condition.

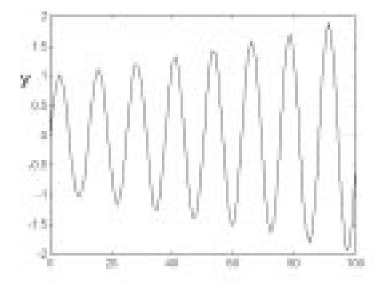
$K_c < K_u$



$K_c = K_u$



$K_c > K_u$



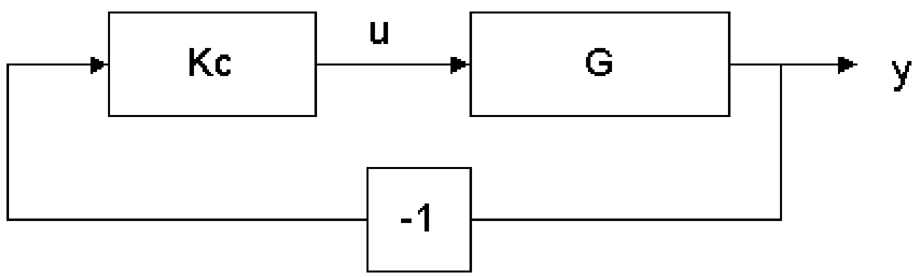
step 4 Calculate and implement PID parameters using the the Ziegler-Nichols tuning tables:

Ziegler-Nichols Controller Settings

Controller	K_c	T_I	T_D
P	$0.5K_u$	$--$	$--$
PI	$0.45K_u$	$T_u/1.2$	$--$
PID	$0.6K_u$	$T_u/2$	$T_u/8$

Remarks :

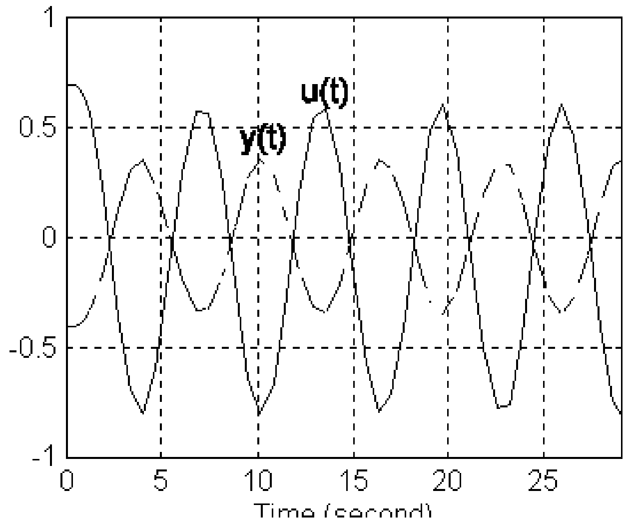
- We call
  - K_u ultimate gain
  - T_u ultimate period ($\omega_{co} = 2\pi/T_u$ critical frequency)
- Ziegler-Nichols tuning is based on the process characteristic at a single point where the closed-loop under P-control shows continuous cycling.



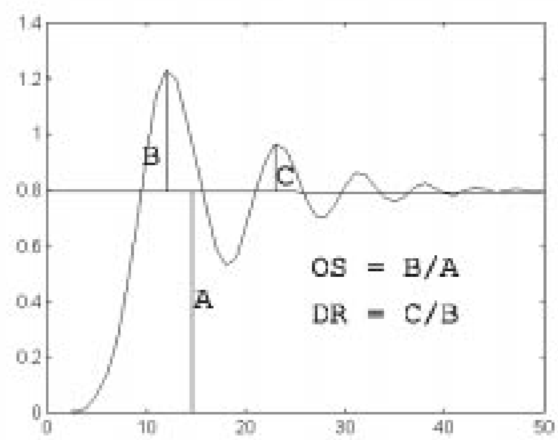
At ω_{co},

$$|G_c K_c|_{\omega_{co}} = 1$$

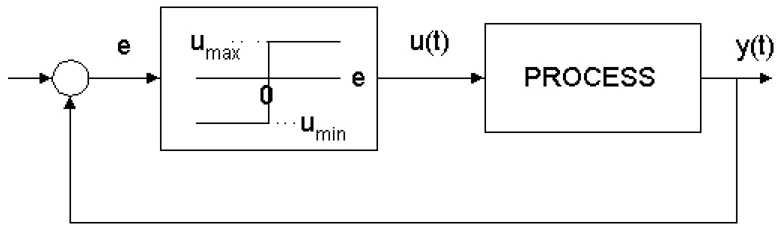
$y(t)$ is 180^o (phase lag) behind $u(t)$.



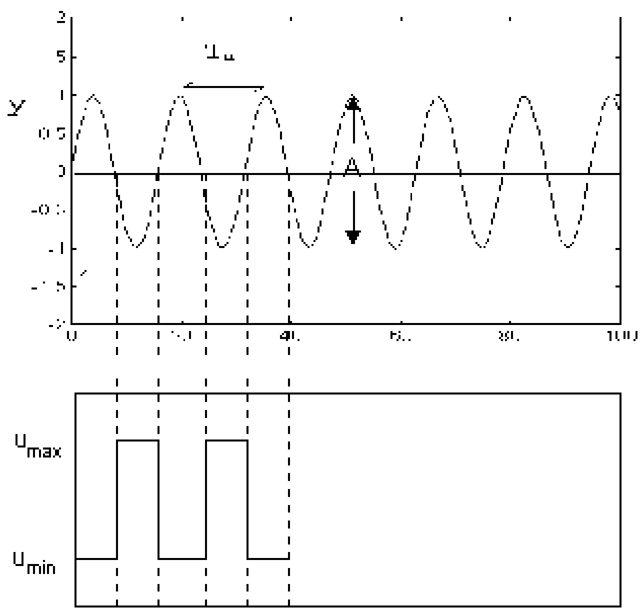
- K_u is the largest K_c for closed-loop stability under P-control. When $K_c = 0.5K_u$ under P-control, the closed-loop approximately shows 1/4 (Quarter) decay ratio response. This roughly gives 50% overshoot.



- An alternative way to emperically find K_u and T_u is to use relay feedack control (sometimes called bang-bang control).



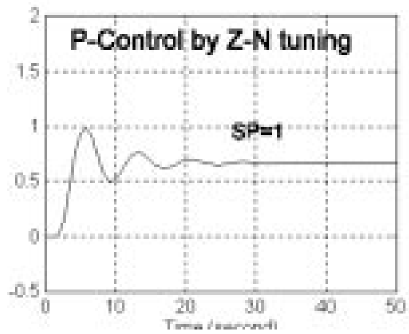
Under relay feedback, the following repsonse is obtained:

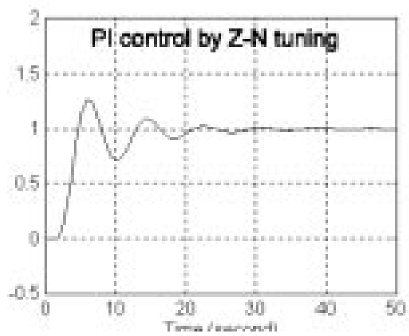


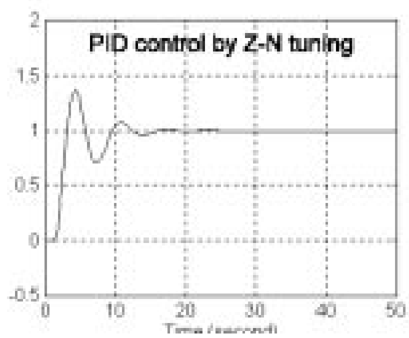
$$K_u = \frac{4(u_{max} - u_{min})}{\pi A}, \quad T_u : \quad \text{from the repsonse}$$

Typical Responses of Z-N Tuning

$$\text{Process:} \quad G(s) = \frac{1}{(s+1)^4}$$





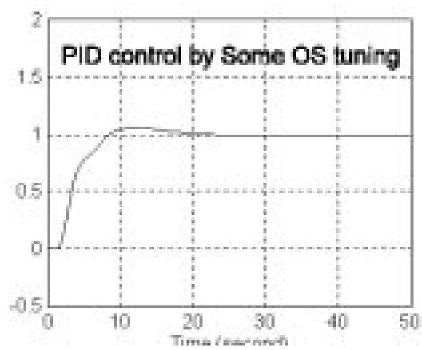


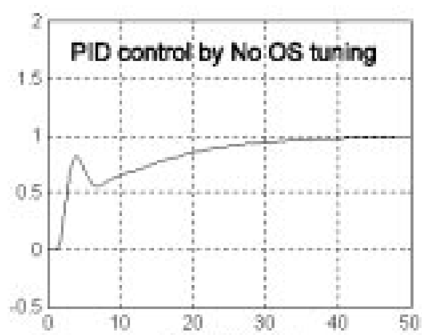
- Since 50% overshoot is considered too oscillatory in chemical process control, the following modified Ziegler-Nichols settings have been proposed for PID contollers:

Modified Ziegler-Nichols Settings

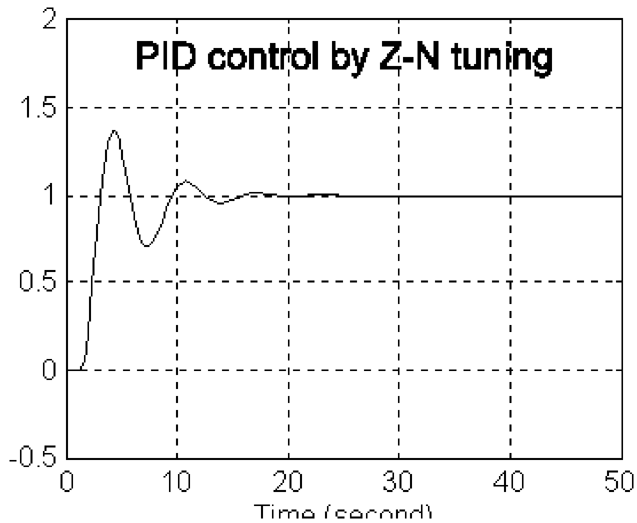
	K_c	T_I	T_D
Original(1/4 decay ratio)	$0.6K_u$	$T_u/2$	$T_u/8$
Some Overshoot	$0.33K_u$	$T_u/2$	$T_u/3$
No Overshoot	$0.2T_u$	T_u	$T_u/3$

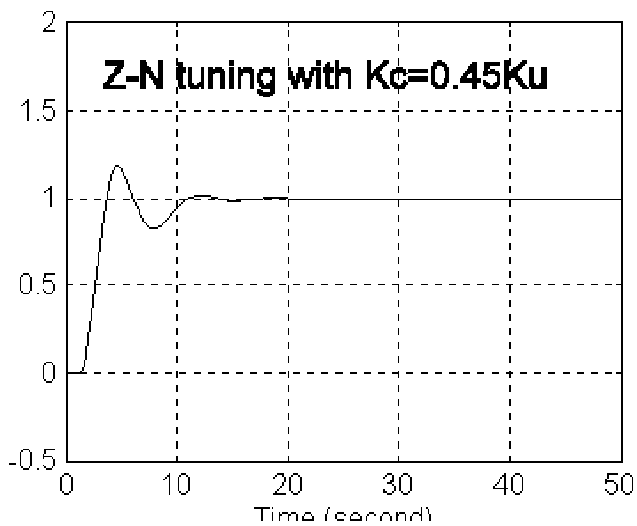
$$\text{Process:} \quad G(s) = \frac{1}{(s+1)^4}$$

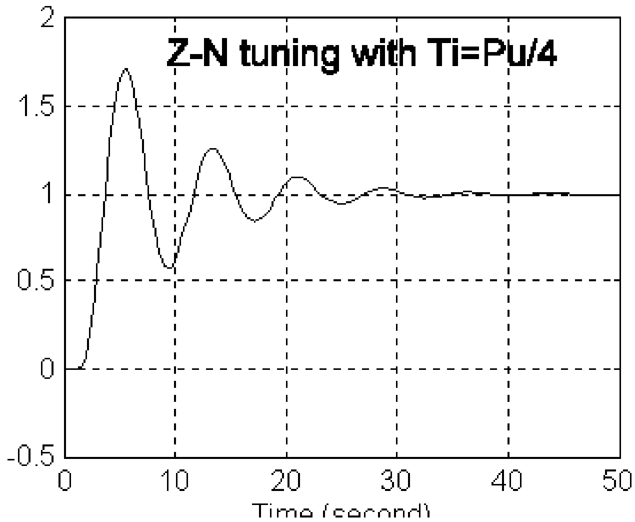


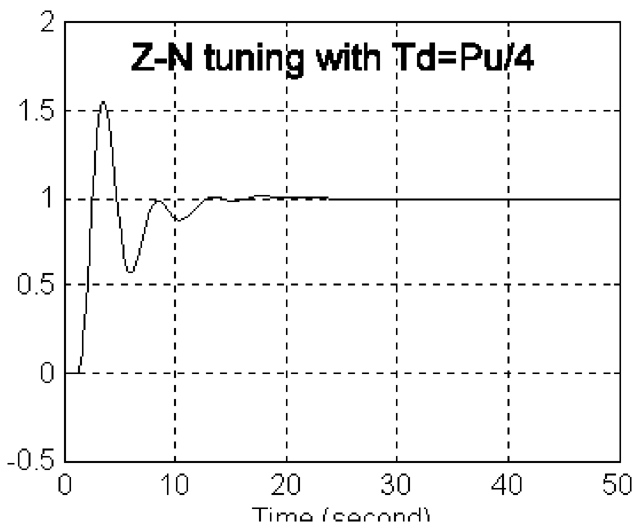


Effects of Tuning Parameters: $G(s) = 1/(s+1)^4$

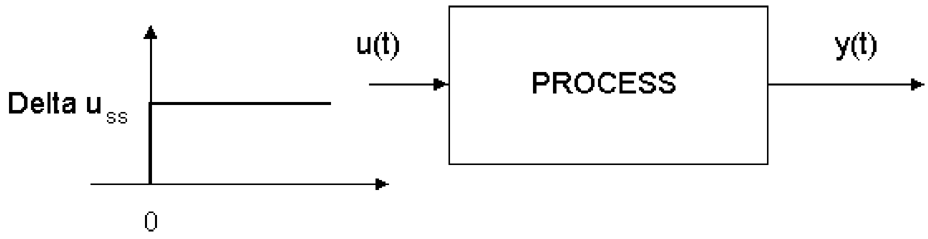




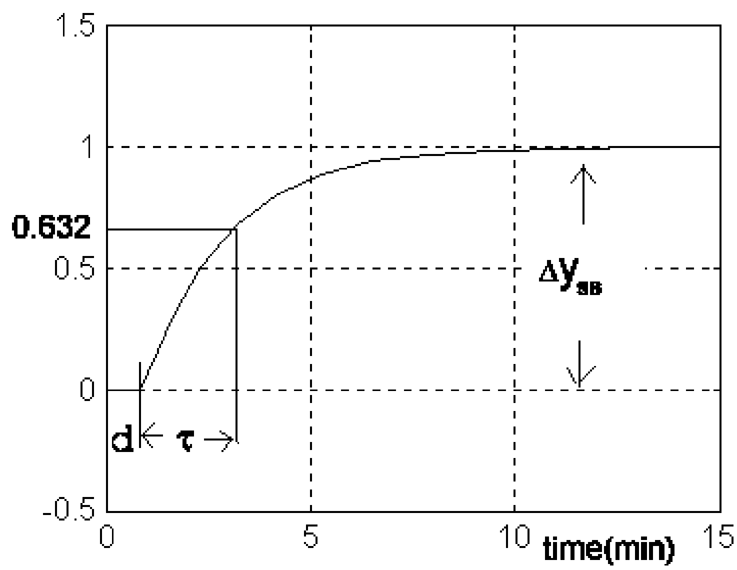

Z-N tuning with Ti=Pu/4
2
1.5
1
0.5
0
-0.5
0
10
20
30
40
50
Time (second)


Z-N tuning with Td=Pu/4
2
1.5
1
0.5
0
-0.5
0
10
20
30
40
50
Time (second)

1.5.2 REACTION-CURVE-BASED METHOD



Not all but many SISO(single-input single-output) chemical processes show step responses which can be well approximated by that of the **First-Order Plus Dead Time(FOPDT)** process.



The FOPDT process is represented by three parameters

- K_p steady state gain defined by $\Delta y_{ss}/\Delta u_{ss}$
- d dead time (min), no response during this period

- τ time constant, represents the speed of the process dynamics.

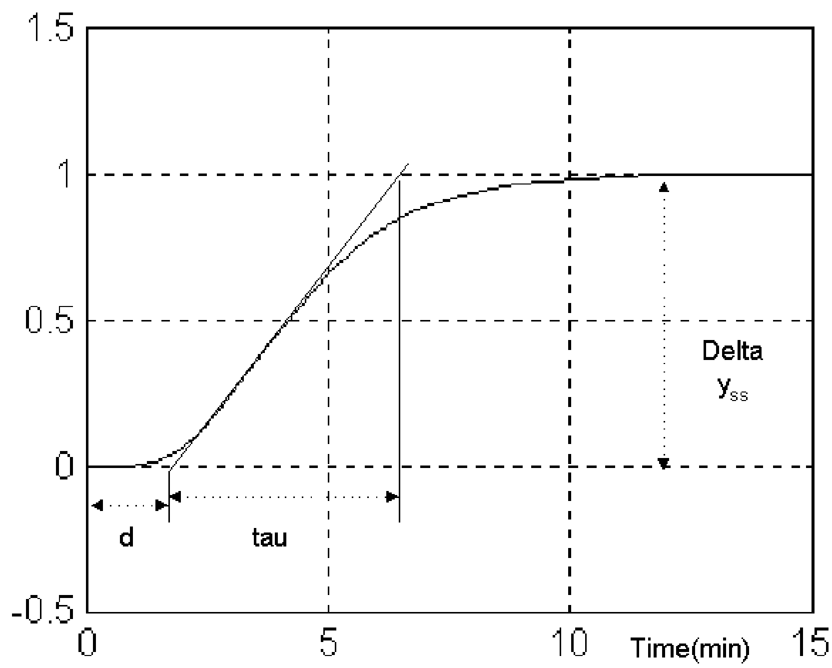
$$G(s) = \frac{K_p e^{-ds}}{\tau s + 1}$$

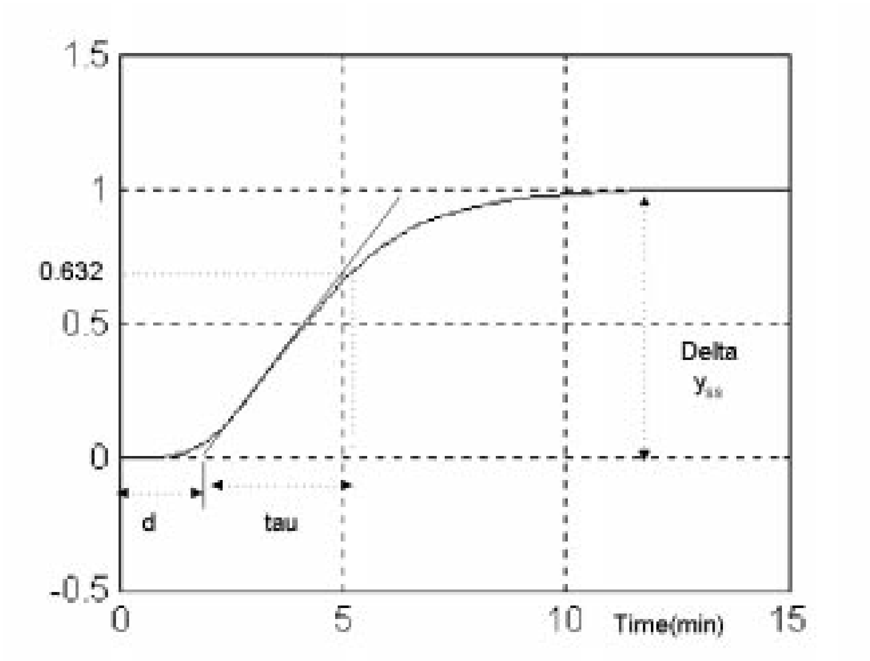
Procedure

step 1 Wait until the process is settled at the desired set point.

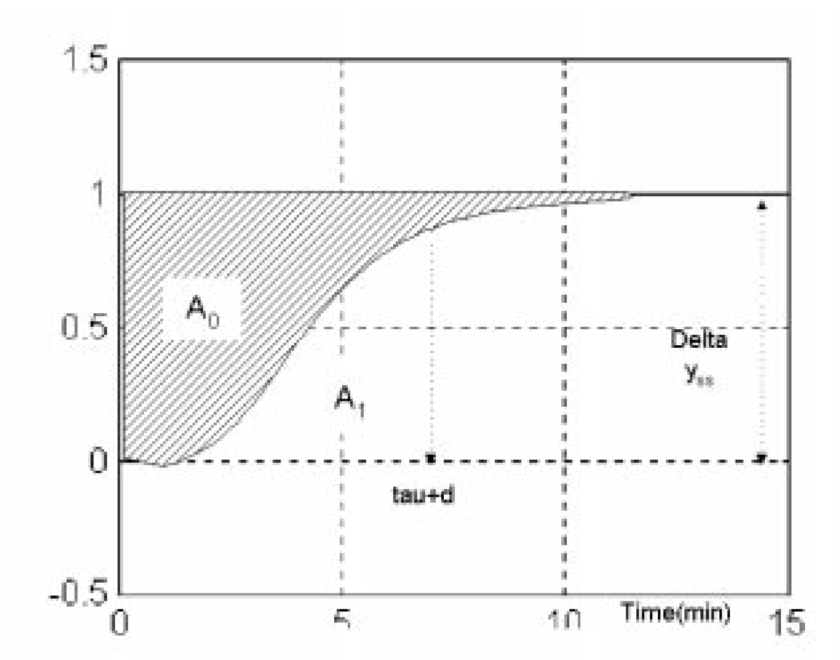
step 2 Switch the A/M toggle to the manual position and increase the CO ($u(t)$) by Δu_{ss} stepwise.

step 3 Record the output reponse and find an approximate FOPDT model using one of the following methods:





- Drawing a tangent is apt to include significant error, especially when the measurement is noisy. To avoid this trouble, the following method is recommended:



1. Obtain $K_p = \Delta y_{ss}/\Delta u_{ss}$.

2. Estimate the area A_0.

3. Let $\tau + d = A_0/K_p$ and estimate the area A_1.

4. Then $\tau = 2.782 A_1/K_p$ and $d = A_0/K_p - \tau$

step 4 Once a FOPDT model is obtained, PID setting can be done based on a tuning rule in the next subsection.

Popular tuning rules are Quater-decay ratio setting and Integral error criterion-based setting.

1.5.3 FOPDT-BASED TUNING RULES

• The following PID tuning rules are applicable for FOPDT processes with $0.1 < d/\tau < 1$.

1/4 Decay Ratio Settings

• Z-N tuning for the FOPDT model.

Controller	K_c	T_I	T_D
P	$(\tau/K_p d)$	$--$	$--$
PI	$0.9(\tau/K_p d)$	$3.33d$	$--$
PID	$1.2(\tau/K_p d)$	$2.0d$	$0.5d$

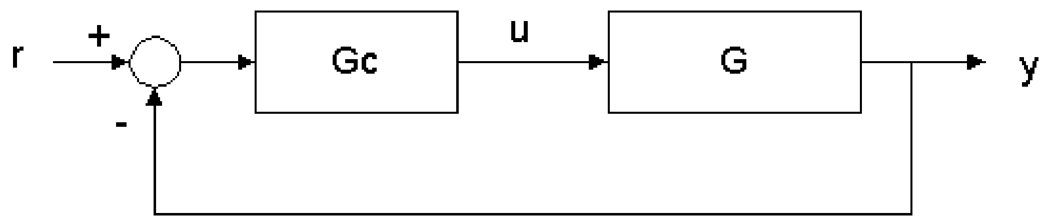
Integral Error Criteria-based Settings

- PID parameters which minimizes one of the following error-integration criteria:

$$\mathrm{IAE} = \int_0^\infty |e(t)|dt$$

$$\mathrm{ISE} = \int_0^\infty e^2(t)dt$$

$$\mathrm{ITAE} = \int_0^\infty t|e(t)|dt$$



IAE Tuning Relations

Type of Input	Controller	Mode	A	B
Load	PI	P	0.984	-0.986
		I	0.608	-0.707
Load	PID	P	1.435	-0.921
		I	0.878	-0.749
		D	0.482	1.137
Set Point	PI	P	0.758	-0.861
		I	1.02^b	-0.323
Set Point	PID	P	1.086	-0.869
		I	0.740^b	-0.130^b
		D	0.348	0.914

ISE Tuning Relations

Type of Input	Controller	Mode	A	B
Load	PI	P	1.305	-0.959
		I	0.492	-0.739
Load	PID	P	1.495	-0.945
		I	1.101	-0.771
		D	0.56	1.006
Set Point	PI	P	-	-
		I	-	-
Set Point	PID	P	-	-
		I	-	-
		D	-	-

ITAE Tuning Relations

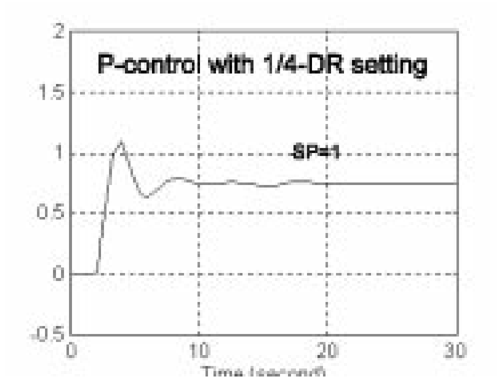
Type of Input	Controller	Mode	A	B
Load	PI	P	0.859	-0.977
		I	0.674	-0.680
Load	PID	P	1.357	-0.947
		I	0.842	-0.738
		D	0.381	0.995
Set Point	PI	P	0.586	-0.916
		I	1.03^b	-0.165
Set Point	PID	P	0.965	-0.855
		I	0.796^b	-0.147^b
		D	0.308	0.929

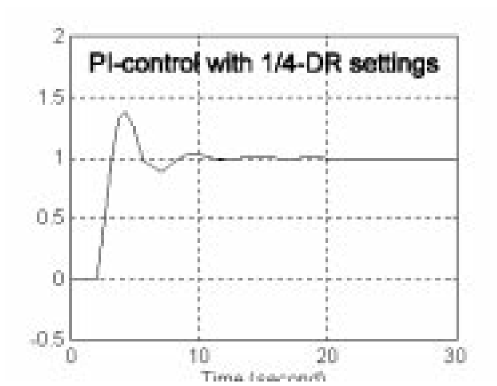
Design relation: $Y = A(d/\tau)^B$ where $Y = KK_c$ for P-mode, τ/τ_I for I-mode, and τ_D/τ for D-mode.

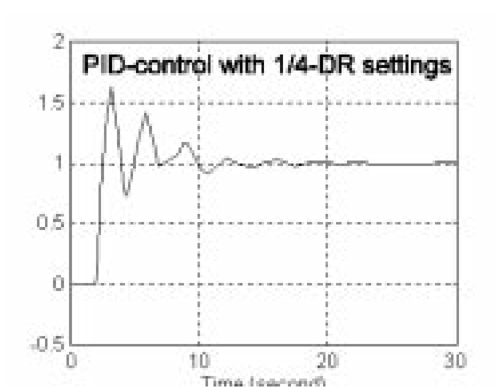
[b] For set-point change, the design relation for I-mode is $\tau/\tau_I = A + B(d/\tau)$.

Performance of 1/4-Decay Ratio Tuning

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



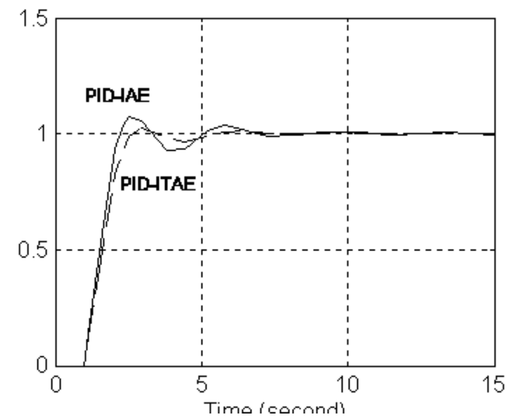




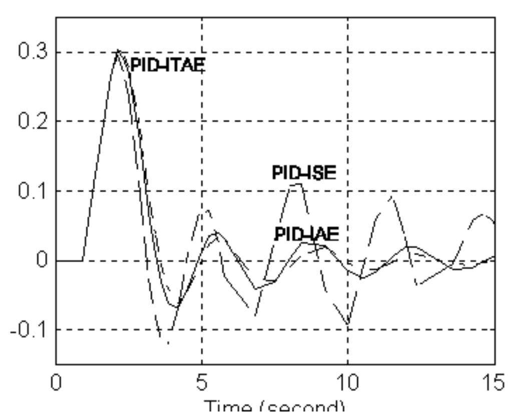
Performance of Integral Error Criteria-based Tuning

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

set-point change



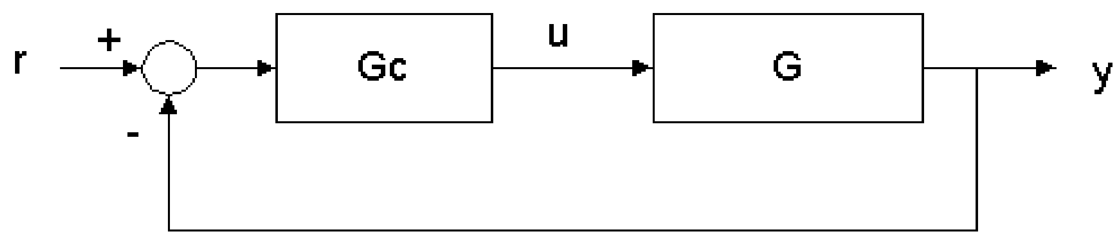
load change



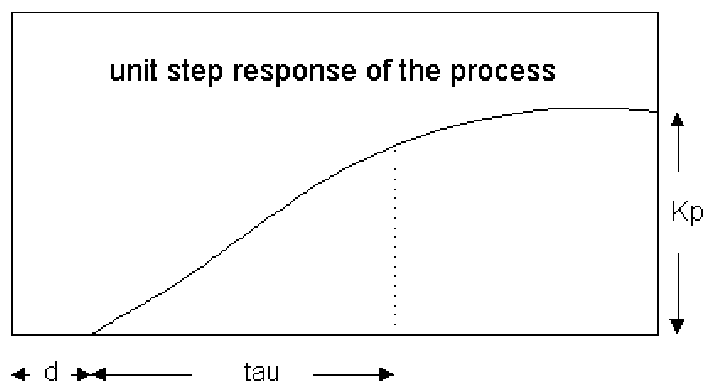
1.5.4 DIRECT SYNTHESIS METHOD - IMC TUNING

- In the direct synthesis method, a desired closed-loop response is specified first for a given process model, and then the controller which satisfies the specification is determined.
- IMC(internal model control) deals with virtually the same problem but from somewhat different point of view.

Consider a closed-loop system.

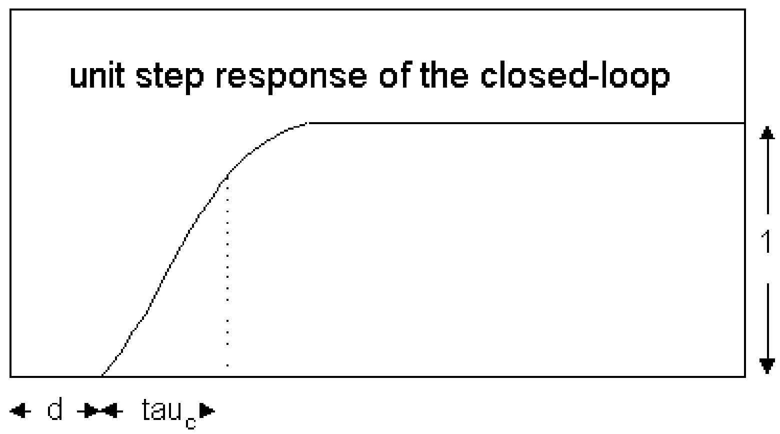


Let the unit step response of the process look like



$$y(s) = G(s)u(s) = \frac{K_p e^{-ds}}{\tau s + 1} u(s)$$

We want to design a controller which gives the following response to unit step change in set point.



$$y(s) = G_{cl}(s)r(s) = \frac{e^{-ds}}{\tau_c s + 1} r(s)$$

- Since the process has time delay of d(min), the closed-loop response should have at least d(min) of time delay.
- For zero offset error, the steady state gain of the closed-loop system is specified as unity.

G_c which satisfies the above objective ?

- From the block diagram, we have

$$y(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} r(s) = G_{cl}(s)r(s) \qquad (*)$$

Solving for G_c gives

$$G_c(s) = \frac{1}{G(s)} \frac{G_{cl}(s)}{1 - G_{cl}(s)} = \frac{(\tau s + 1)/K_p}{\tau_c s - e^{-ds} + 1} \qquad --- \qquad (**)$$

When d is small,

$$e^{-ds} \approx 1 - ds$$

Therefore,

$$G_c(s) \approx \frac{(\tau s + 1)/K_p}{(\tau_c s + d)s} = \left(\frac{\tau / K_p}{\tau_c + d} \right) \left\{ 1 + \frac{1}{\tau s} \right\} \qquad -- \quad \text{PI-type}$$

Remarks :

- Simple and easy, but requires a model (of fairly good quality).
- τ_c is *de facto* a tuning parameter. As τ_c gets small, faster speed closed-loop response (to set point change) but more sensitive to measurement noise as well as model error. Note that tuning is reduced to a *single* parameter. Hence it is somewhere between random tuning (two or three parameters) and using PID tuning table (no parameter).
- Using (*), we can design $G_c(s)$ for arbitrary $G(s)$ and $G_{cl}(s)$ (but some restriction applies on the choice of G_{cl}). In this case, a more general type of controller other than PID is obtained. For example, (**) is in fact a Smith predictor equation.
- The following controller settings was derived for different process models based on the IMC-tuining rule.

IMC-based PID Controller Settings

Type of Model	K_cK_p	τ_I	τ_D
$\frac{K_p}{\tau s+1}$	$\frac{\tau}{\tau_c}$	τ	
$\frac{K_p}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{\tau_1+\tau_2}{\tau_c}$	$\tau_1+\tau_2$	$\frac{\tau_1\tau_2}{\tau_1+\tau_2}$
$\frac{K_p}{\tau^2 s^2+2\zeta\tau s+1}$	$\frac{2\zeta\tau}{\tau_c}$	$2\zeta\tau$	$\frac{\tau}{2\zeta}$
$\frac{K_p(-\beta s+1)}{\tau^2 s^2+2\zeta\tau+1}, \beta > 1$	$\frac{2\zeta\tau}{\tau_c+\beta}$	$2\zeta\tau$	$\frac{\tau}{2\zeta}$
$\frac{K_p}{s}$	$\frac{1}{\tau_c}$		
$\frac{K_p}{s(\tau s+1)}$	$\frac{1}{\tau_c}$		τ

1.6 PRACTICAL CONSIDERATIONS

In commercial PID controllers, many modifications are made from the standard form. Some of them are

- remove derivative kick, sometimes together with proportional kick
- suppress noise sensitivity of the D-mode
- anti-reset windup
- bumpless transfer during Auto-Manual switching
- bumpless parameter change
- nonlinear compensation
- etc.

Removing Kicks

- During *regulation*, SP is kept constant in most time and changed rarely. When SP is changed, a surge signal is produced by the derivative mode which acts as a "shock" on the process. We call this phenomenon *derivative kick*.
  To remove this harmful effect, derivative is not taken on SP in most commercial PID controllers.

$$\frac{de(t)}{dt} = \frac{d}{dt}(r(t) - y(t)) \quad \rightarrow \quad -\frac{dy(t)}{dt}$$

  PID controllers with the above modification is denoted as PI-D controllers.

- A similar phenomenon can be said for P-mode. In order to mitigate *proportional kick*, some commercial controllers modifies the P-mode as follows:

$$u^P(t) = K_c[\alpha r(t) - y(t)] \quad \text{where} \;\; 0 \leq \alpha \leq 1$$

Controllers with P and D modifications are denoted as I-PD controllers.

Filtered D-mode

- For suppression of high frequency noise sensitivity, the D-mode is further modified to to take derivative on a (low-pass) filtered output signal

$$-\frac{dy(t)}{dt} \quad \rightarrow \quad -\frac{dy^F(t)}{dt} \;\; \text{where} \;\; a\frac{dy^F(t)}{dt} + y^F(t) = y(t) \;\;\; a \approx 3\tau_D \sim 10\tau_D$$

With such modification, the high frequency gain of the D-mode is bounded not to exceed some limit.

Bumpless A/M Transfer

$$u(t) = \underbrace{u_{bias} + \frac{K_c}{T_I}\int_0^t e(\tau)d\tau}_{u_{bias}(t)} + K_c\left[e(t) - T_D\frac{dy^F(t)}{dt}\right]$$

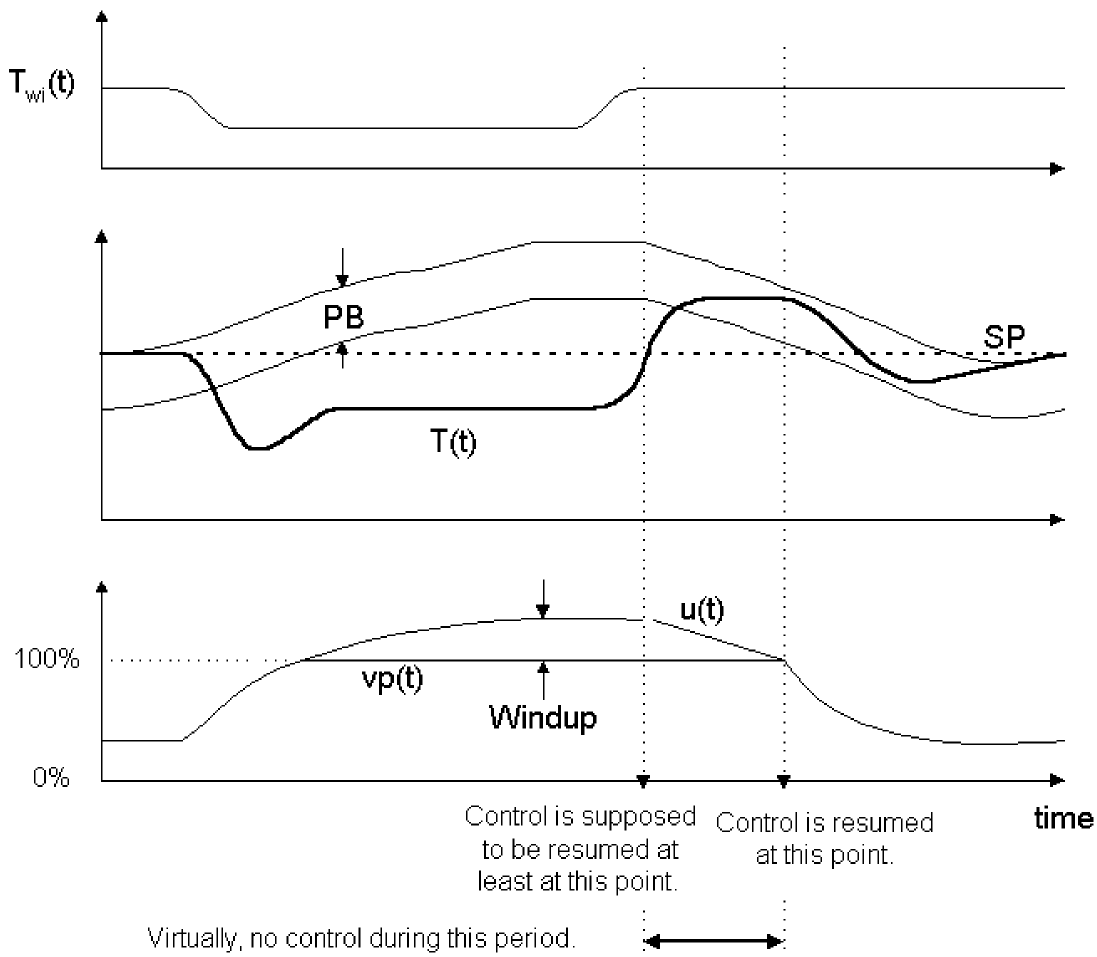
$u_{bias}(t) \quad \Rightarrow \quad u_{bias}$ at Auto to Manual switching

Bumpless Parameter Change

Even when $e(t) = 0$, adjusting K_c and/or T_I modifies the I-mode value, which in turn changes $u_{bias}(t)$.

To avoid this trouble, a special algorithm needs to be introduced in the I-mode calculation.

Anti-Windup

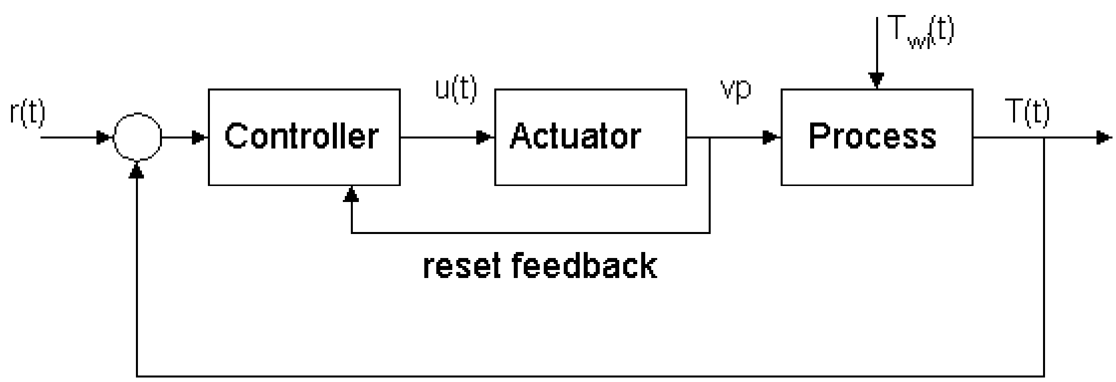


- Suppose that the actuator has saturated at 100% (0 %). Without knowing the actuator saturation, the I-mode continues error integration.
  $\rightarrow$ Integral WINDUP !!

- Once windup occurs, the actuator does not return to its normal range until the windup is removed by opposite-signed control error over a certain period.
  Control is lost for this period.

  To prevent the I-mode from being wound up, the controller needs to monitor $vp(t)$. If $vp(t)$ is observed to be stuck at a saturation limit, the controller stops integration until opposite-signed control error enters.

  This function is called **Anti-Windup**.



The anti-windup is a very important concept not only in feedback control with actuator saturation but also in various multi-loop control techniques.

A Simple but practical windup algorithm is

$$u(t) = u_{bias} + \frac{1}{T_t}\int_0^t (vp(\tau) - u(\tau))d\tau + K_c\left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau + T_D\frac{de(t)}{dt}\right]$$

The equivalent discrete form is

$$\Delta u_k = \frac{h}{T_t}(vp_k - u_{k-1}) + K_c\left([e_k - e_{k-1}] + \frac{h}{T_I}e_k + \frac{T_D}{h}[e_k - 2e_{k-1} + e_{k-2}]\right)$$

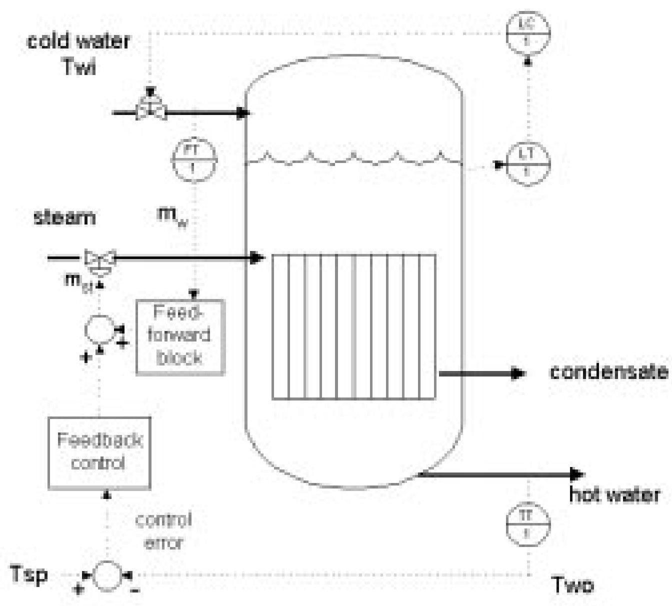
where T_t is called *tracking time constant.*

Chapter 2

MULTI-LOOP CONTROL AND FURTHER PRACTICAL ISSUES

2.1 FEEDFORWARD-FEEDBACK CONTROL

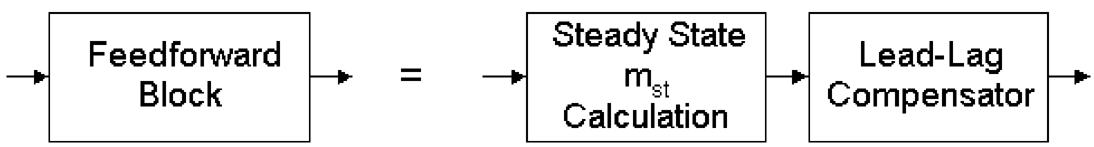
Revisit of the Heating Tank Process

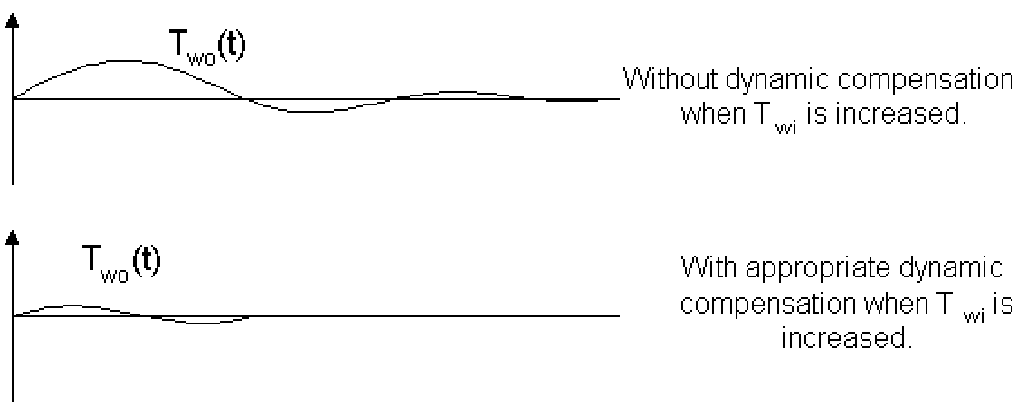


- In section 1.1, we considered feedforward control based on steady-state compensation.

$$\lambda_{st} m_{st} = m_w C p_w (T_{wo} - T_{wi}) \ \rightarrow \ m_{st} = \frac{m_w C p_w (T_{sp} - T_{wi})}{\lambda_{st}}$$

- When the time lag between T_{wi} and T_{wo} is significantly different from that of m_{st} and T_{wo}, it is necessary to introduce a dynamic compensation in addition.
  The **lead-lag** compensator is used in industrial controllers for this purpose.

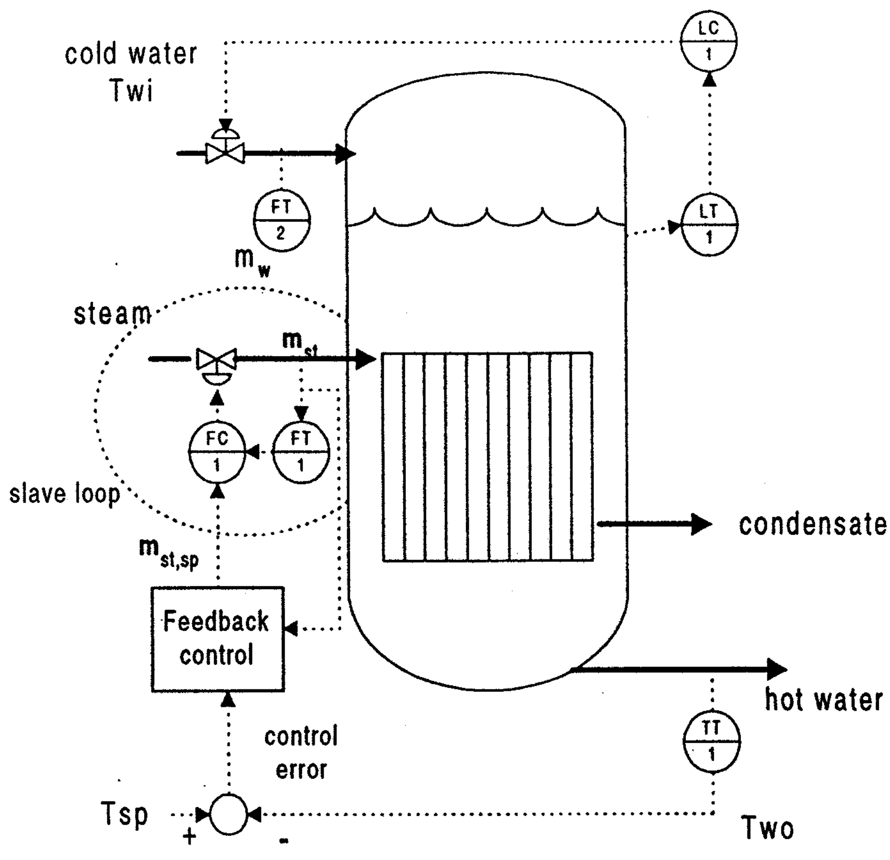


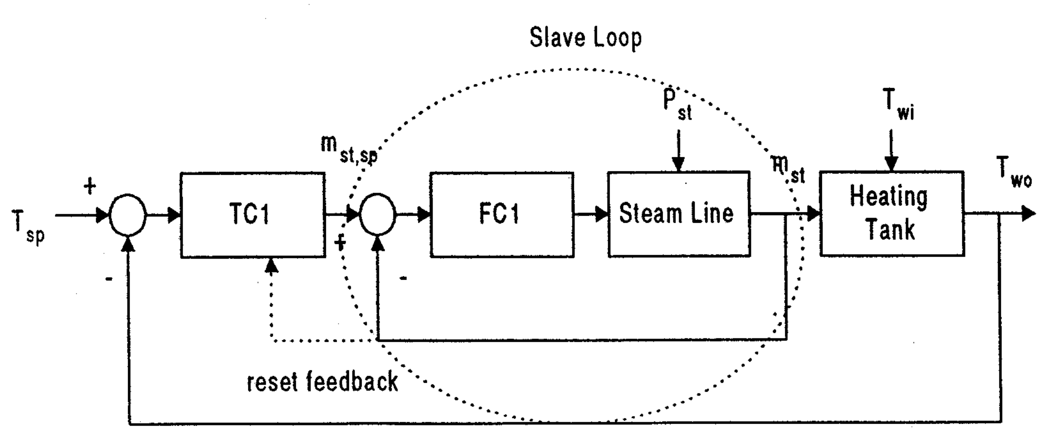


- When appropriately designed, feedforward control can considerably improve the overall control performace. In this case, feedback control plays only a trimming role.

- Feedforward control can be designed for *measurable disturbances* only.

2.2 CASCADE CONTROL





Why cascade control ?

- Reject disturbance in the slave process (P_{st} - steam pressure) before it affects the main process variable (T_{wo}).
- Linearize the slave process.
- Improve the dynamics of the slave process.

Tips for Implementation

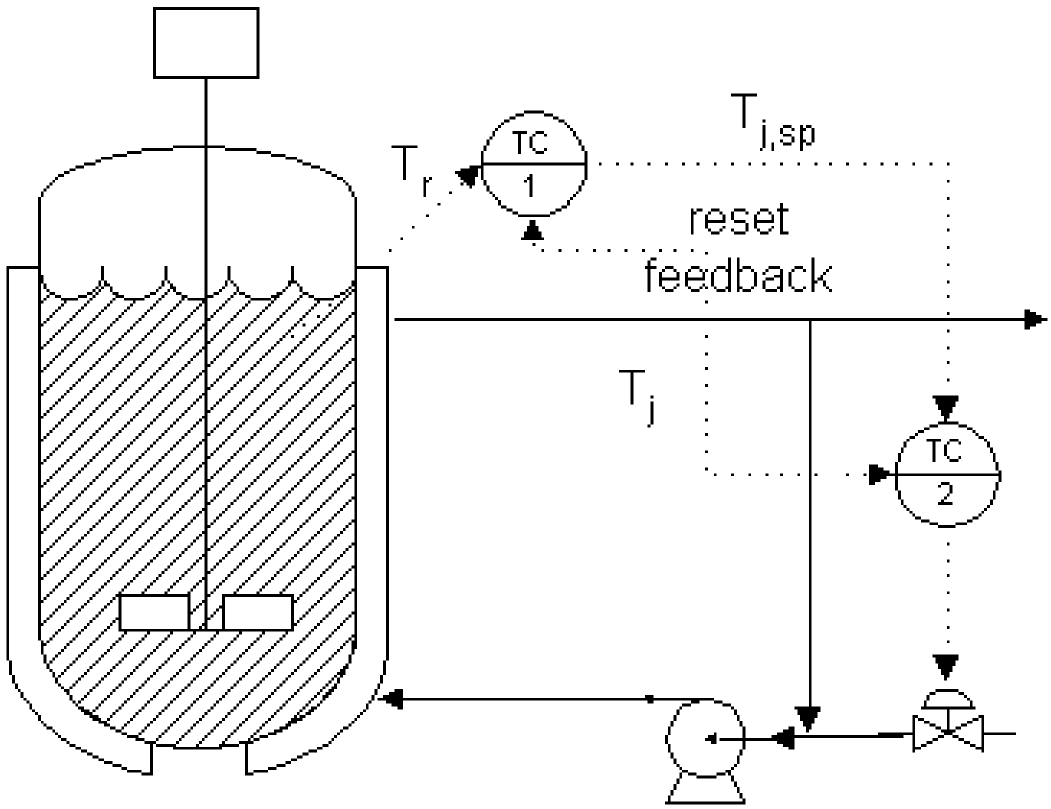
- Satisfactory performance can be expected when the slave process is at least three times as fast as the master process in terms of time constant.

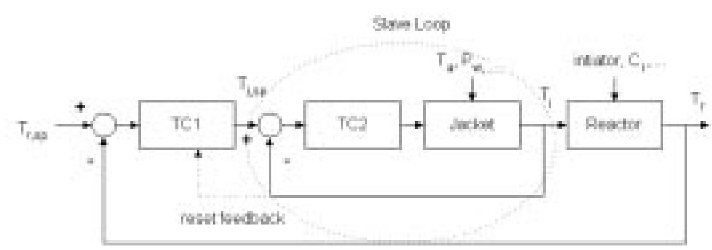
  When the master process has a time constant similar to or shorter than that of the slave process, there is little incentive for cascade control.

  Flow rate is frequently subject to disturbance in supply pressure and varies nonlinearly with valve position. But the process itself is very fast.
  $\Rightarrow$ an ideal target for cascade control.
- Since the output from the slave process is not the major process variable to control, it is not necessary to use I-mode in the slave controller. By this reason, P-control is usually employed for the slave controller.
- Reset feedback is required for output tracking as well as anti-reset windup.

More Examples

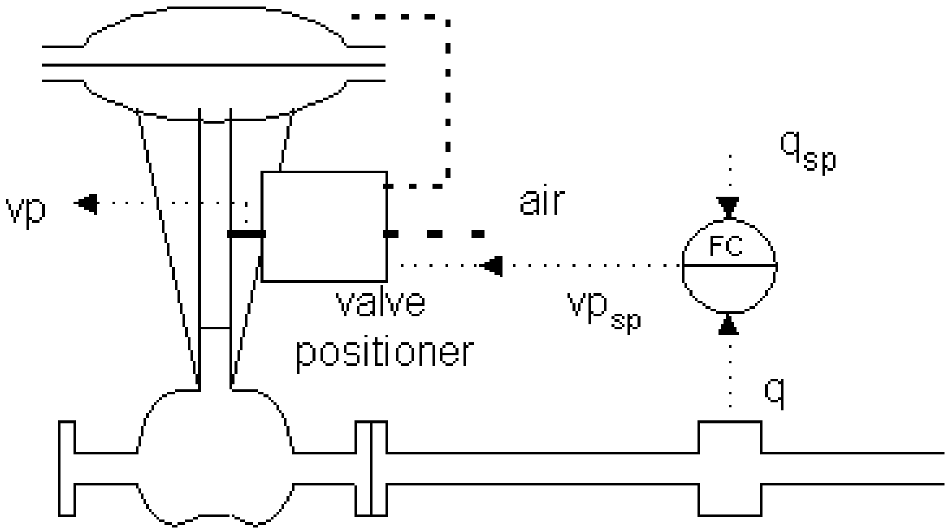
Batch Reactor

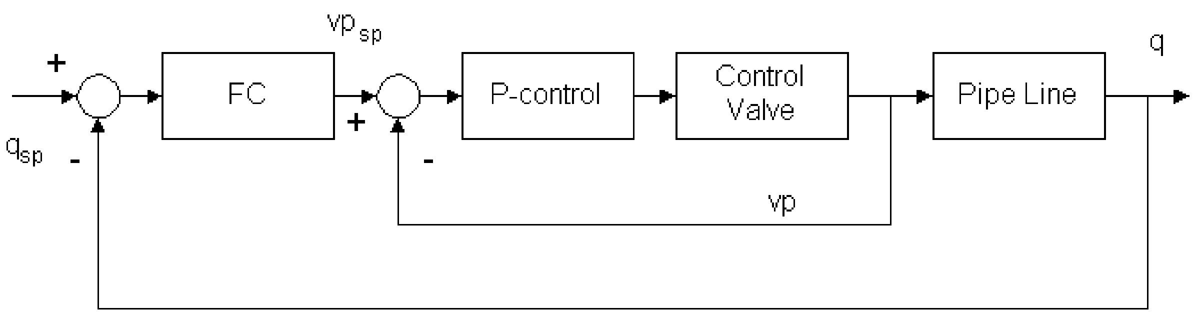




Since the jacket has a much smaller volume than the reactor, the batch reactor is considered a good target for cascade control from the viewpoint of dynamics distribution.

Flow Rate Control with a Control Valve with Positioner

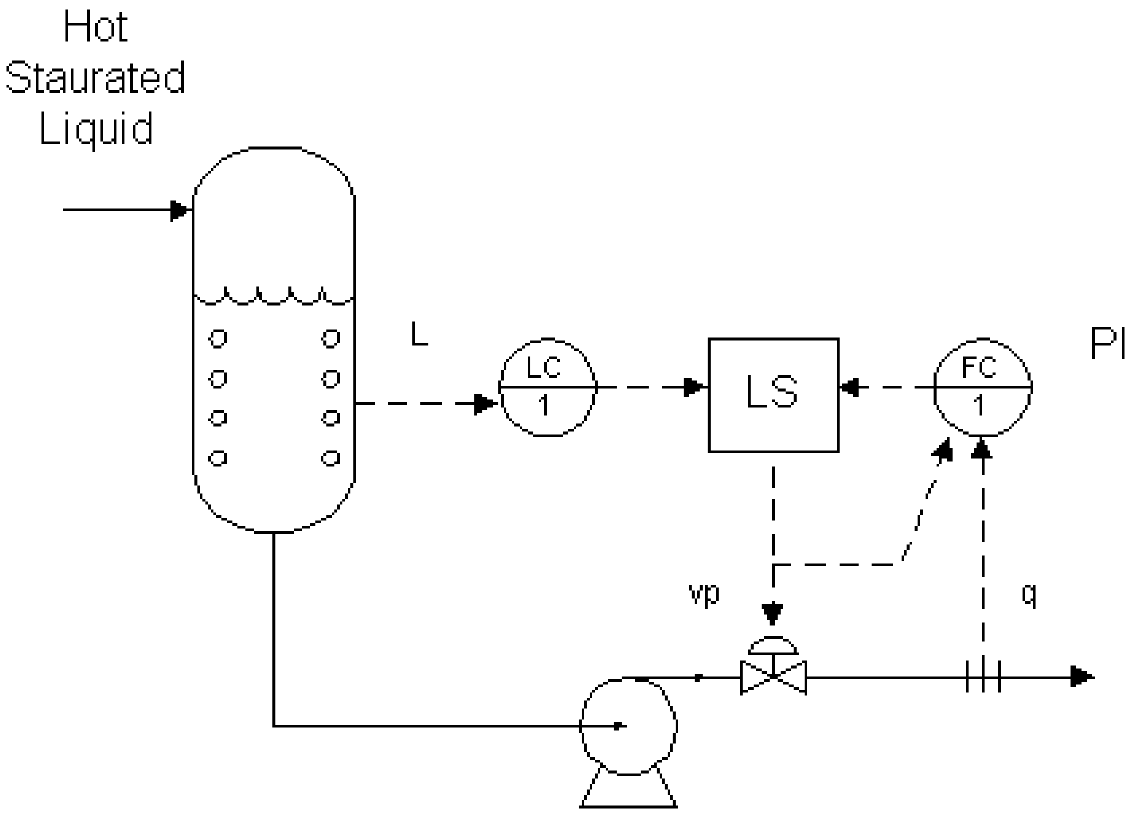




The control valve usually has a longer time constant than the pipeline itself. Therefore, FC cannot be tightly tuned.

By this reason, the positioner-control valve is not recommended for flow rate control where the process dynamics is fast compared with the valve dynamics.

2.3 OVERRIDE CONTROL



- It is assumed that only FC has an I-mode.
- Under normal condition, q is required to be kept constant.
- When L is lowered below a certain level, LC overrides the control loop and starts level control.
- When L is returned to the normal range, FC takes over the control.
- To prevent the integral in FC from being wound up when LC overrides the actuator, reset feedback is needed.

Chapter 3

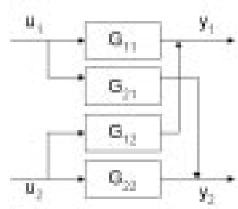
CONTROL OF MULTI-INPUT MULTI-OUTPUT(MIMO) PROCESSES

3.1 MIMO PROCESS ?

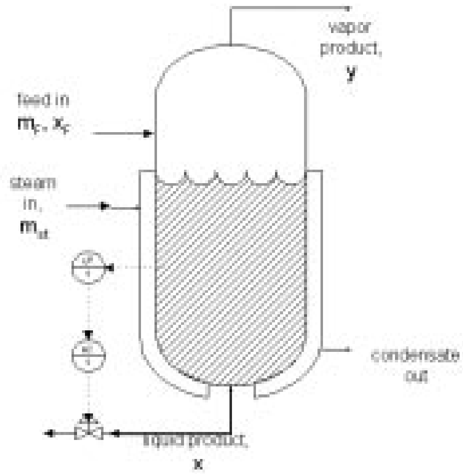
Many chemical processes have MIMO characteristics.

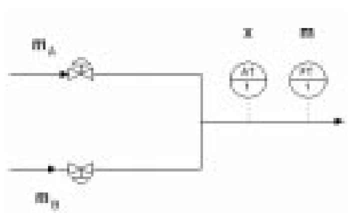
Each manipulated variable influences two or more process variables simultaneously.

This trait gives rise to difficulties in controller design, which has not been observed in SISO control.



The followings are some of the MIMO process examples:





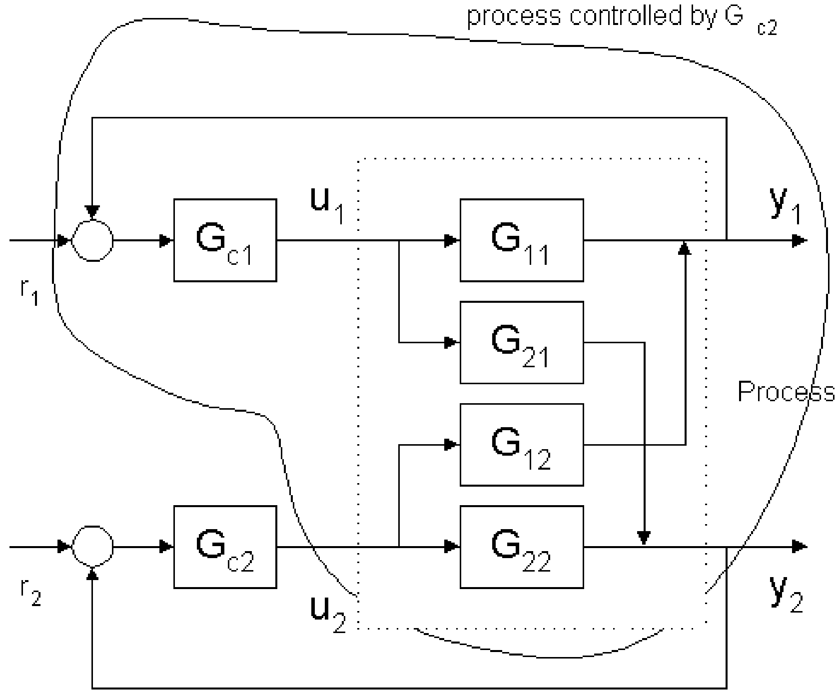
Crude unit

Distialltion columns

. . .

3.2 INTERACTION AND I/O PAIRING

3.2.1 INTERACTION



- When G_{c1} is open, the process that G_{c2} controls is G_{22}.
- When G_{c1} is closed, however, the process controlled by G_{c2} becomes

$$\bar{G}_{22} = G_{22} - \frac{G_{21}G_{c1}G_{12}}{1 + G_{c1}G_{11}}$$

  If both G_{12} and G_{21} are not zero, $\bar{G}_{22}$ varies with G_{c1}.

  If G_{c1} is adjusted, G_{c2} should be retuned, too.

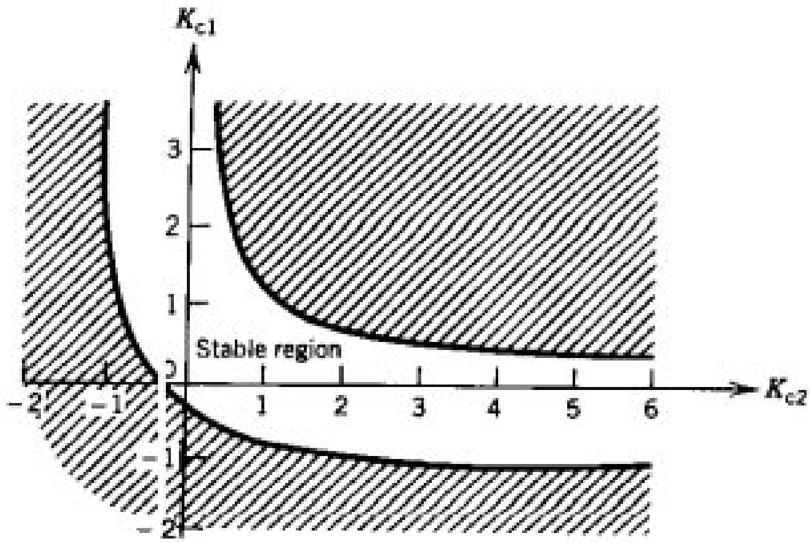
- Same thing can be said for G_{c1}.
- The above probelm is caused by the *interaction* through G_{21} and G_{12}.

Ex. Consider the following process:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{2}{10s+1} & \frac{1.5}{s+1} \\ \frac{1.5}{s+1} & \frac{2}{10s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Assume that P-control is used for both G_{c1} and G_{c2}.

- When only one of y_1 and y_2 is under control, the controller gain can have value (as far as it is positive) without causing stability problem.
- When both control loops are closed, stability is attained for the controller gains in a limited region shown below. Both gains cannot be increased simultaneously.

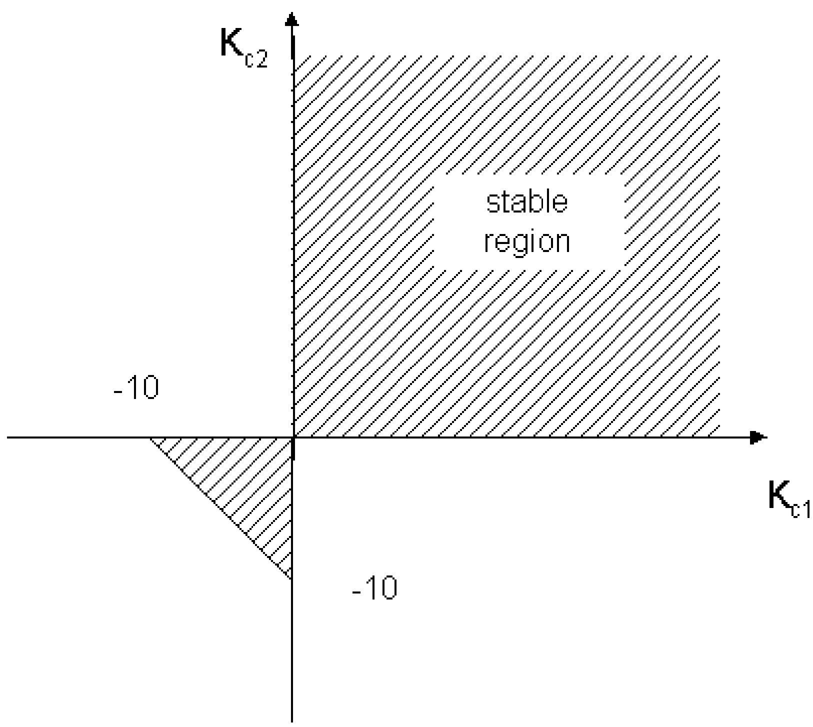


Ex. This time, we consider the process

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{2}{10s+1} & \frac{1.5}{s+1} \\ -\frac{1.5}{s+1} & \frac{2}{10s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Again, it is assumed that P-control is used for both G_{c1} and G_{c2}.

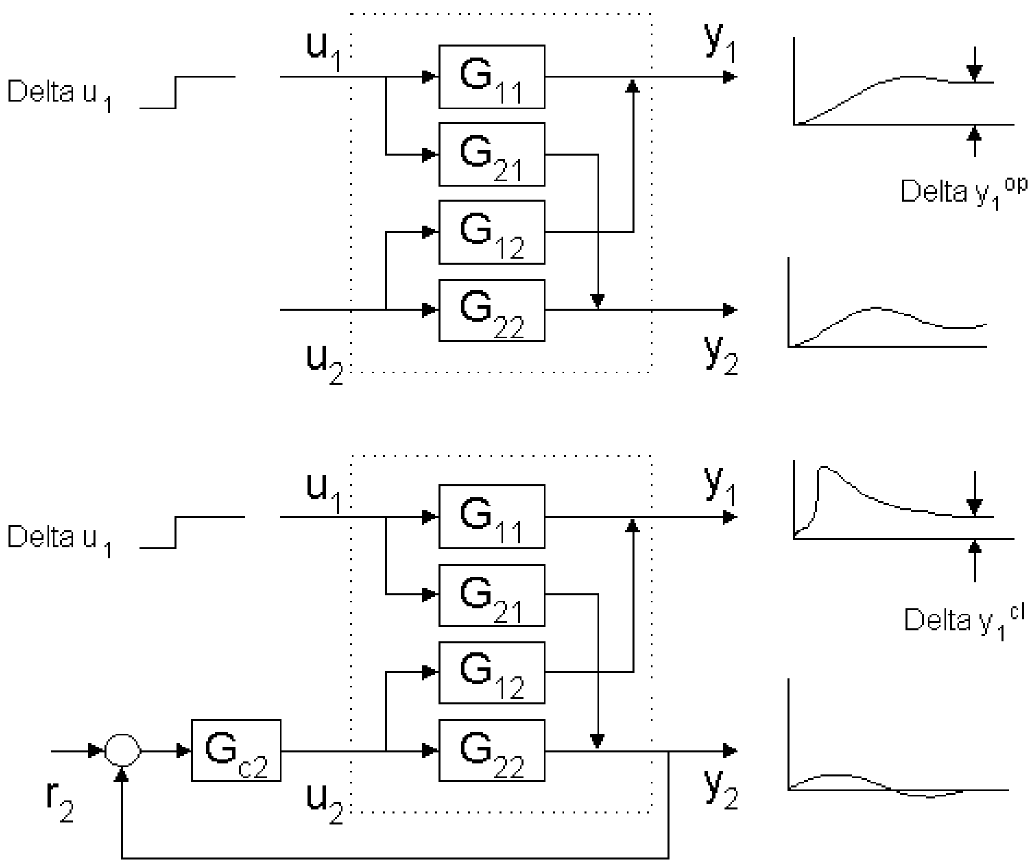
- The stability region is changed to



3.2.2 I/O PAIRING

- Suppose that we want to control a MIMO process using single-loop controllers (in a decentralized fashion).
- Then the pairing will be the primary question. $u_1 - y_1/u_2 - y_2$ or $u_1 - y_2/u_2 - y_1$?
- From the previous considerations, the judgement for proper pairing can be made based on an interaction measure.
- How to measure interaction ? Many different ways, but the most widely used one is the *relative gain* by Bristol(1968).

Relative Gain



Definition :

$$\lambda_{11} = \frac{\Delta y_1^{op}/\Delta u_1}{\Delta y_1^{cl}/\Delta u_1}$$

$$= \frac{\text{gain between } u_1 \text{ and } y_1 \text{ when all other loops are open}}{\text{gain between } u_1 \text{ and } y_1 \text{ when all other loops are closed}}$$

- The relative gain is usually defined under steady state conditions.
- $\Lambda = [\lambda_{ij}]$ is called the *Relative Gain Array*.

Interpretations:

- Obviously, $\lambda_{11} = 1$ when G_{12} and/or G_{21} is zero. $\Rightarrow$ No interaction, $u_1 - y_1$ pair is decoupled from other loops.
- $\lambda_{11} = 0$ when $G_{11} = 0$ $\Rightarrow$ No coupling between u_1 and y_1; y_1 should be paired with u_2 for a 2×2 process.
- $\lambda_{11} > 1$ $\Rightarrow$ The gain is increased when other loops are closed.
- $\lambda_{11} < 1$ $\Rightarrow$ Sign of the gain is reversed when other loops are closed.
- $\lambda_{11} \gg 1$ or $\lambda_{11} \ll 1$ implies that the system has serious interaction. $\rightarrow$ SISO pairing has limitations. $\rightarrow$ should rely on MIMO control.

Properties:

- For an $n \times n$ process, $\sum_i \lambda_{ij} = \sum_j \lambda_{ij} = 1$.
  For a 2×2 process,

$$\lambda_{11} = \frac{1}{1 - K_{12}K_{21}/K_{11}K_{22}}, \quad \lambda_{22} = \lambda_{11}, \quad \lambda_{12} = \lambda_{21} = 1 - \lambda_{11}$$

- The relative gain can be directly computed from the steady state gain matrix of the process.

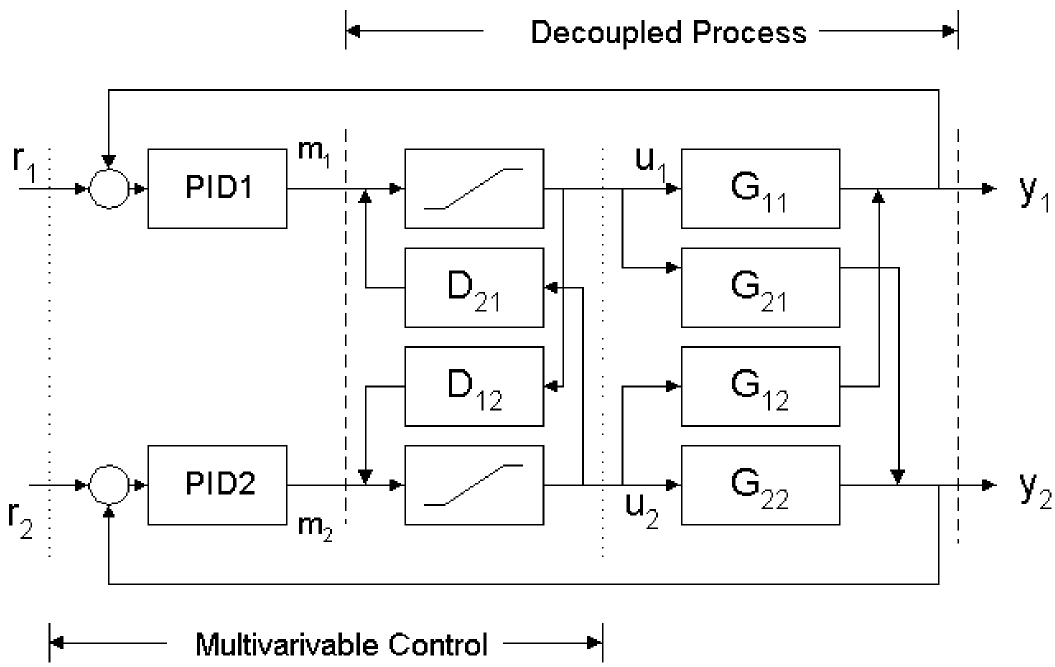
  Let $\mathbf{K}$ be the steady state gain matrix of a process.
  Then,

$$\Lambda = \mathbf{K} \bigotimes \mathbf{K}^{-T}$$

  where $\otimes$ denotes the element-by-element multiplication.

3.3 DECOUPLING

- When λ_{11} is far from one for a 2×2 process, decentralized control has limitations. $\Rightarrow$ Multivariable Control
- One of the classical multivariable control techniques is *decoupling control.*



- **Decoupled Process**

  If we neglect the input saturation blocks,

$$\begin{aligned} y_1 &= (G_{11} + G_{12}D_{12})m_1 + (G_{12} + G_{11}D_{21})m_2 \\ y_2 &= (G_{21} + G_{22}D_{12})m_1 + (G_{22} + G_{21}D_{21})m_2 \end{aligned}$$

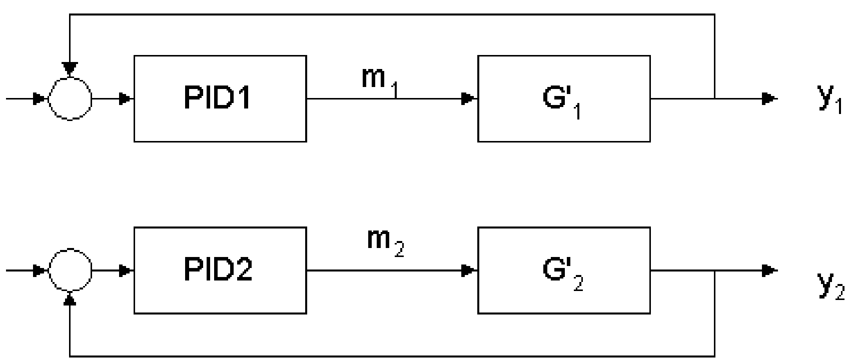
Let

$$\begin{aligned} D_{21} &= -\frac{G_{12}}{G_{11}} \\ D_{12} &= -\frac{G_{21}}{G_{22}} \end{aligned}$$

Then

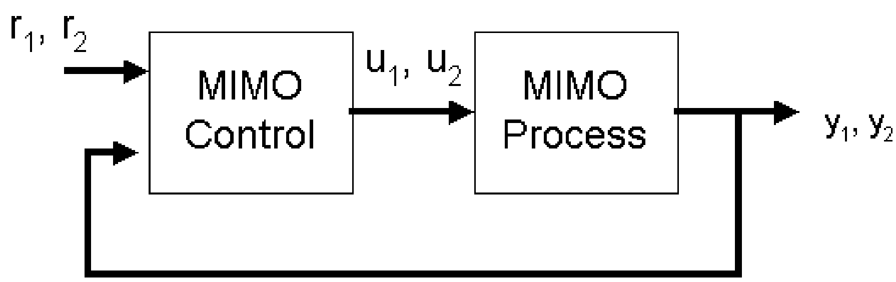
$$\begin{aligned} y_1 &= \left(G_{11} - \frac{G_{12}G_{21}}{G_{22}}\right) m_1 = G_1' m_1 \\ y_2 &= \left(G_{22} - \frac{G_{21}G_{12}}{G_{11}}\right) m_2 = G_2' m_2 \end{aligned}$$

The relative gain for the decoupled process, λ_{11d}, is one.



- **Multivariable Control**

$$
\begin{aligned}
u_1 &= m_1 + D_{21}m_2 \\
u_2 &= m_2 + D_{12}m_1
\end{aligned}
$$

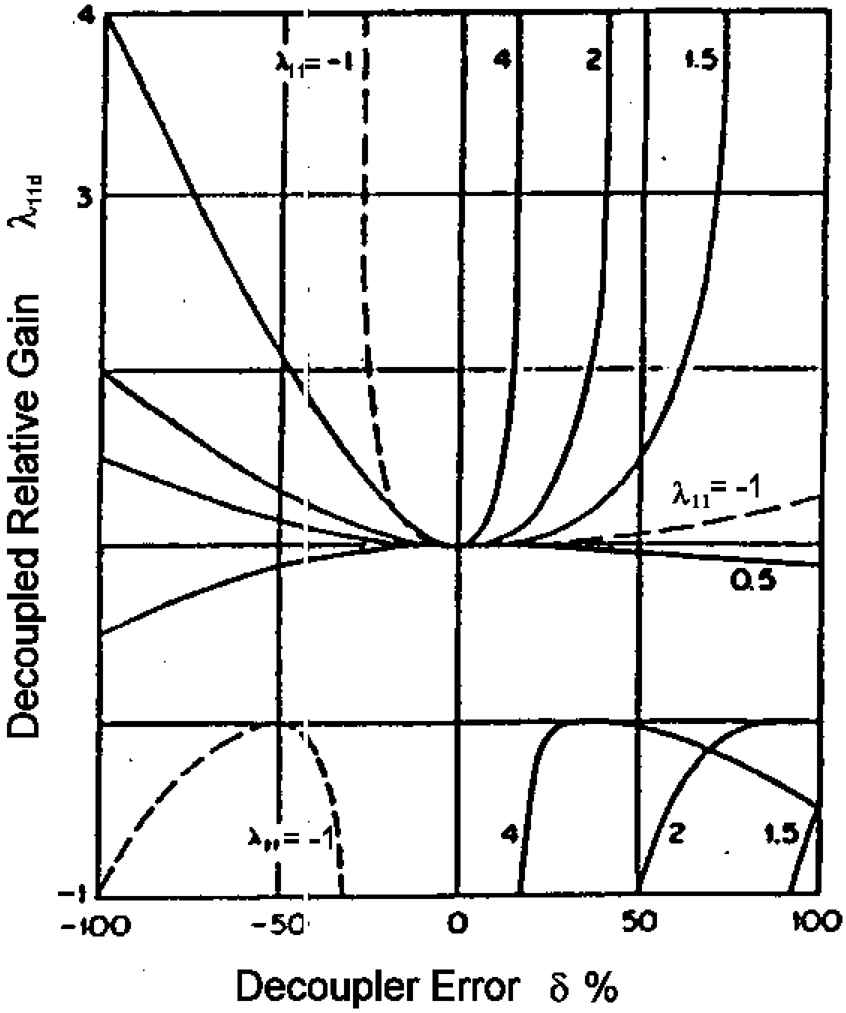


Remarks :

- A decisive drawback of decoupling control is that it is sensitive to model error.
  - For processes with $\lambda_{11} \approx 1$, $\lambda_{11d} \approx 1$.
  - As λ_{11} deviates from one, λ_{11d} also deviates from one.

  Let

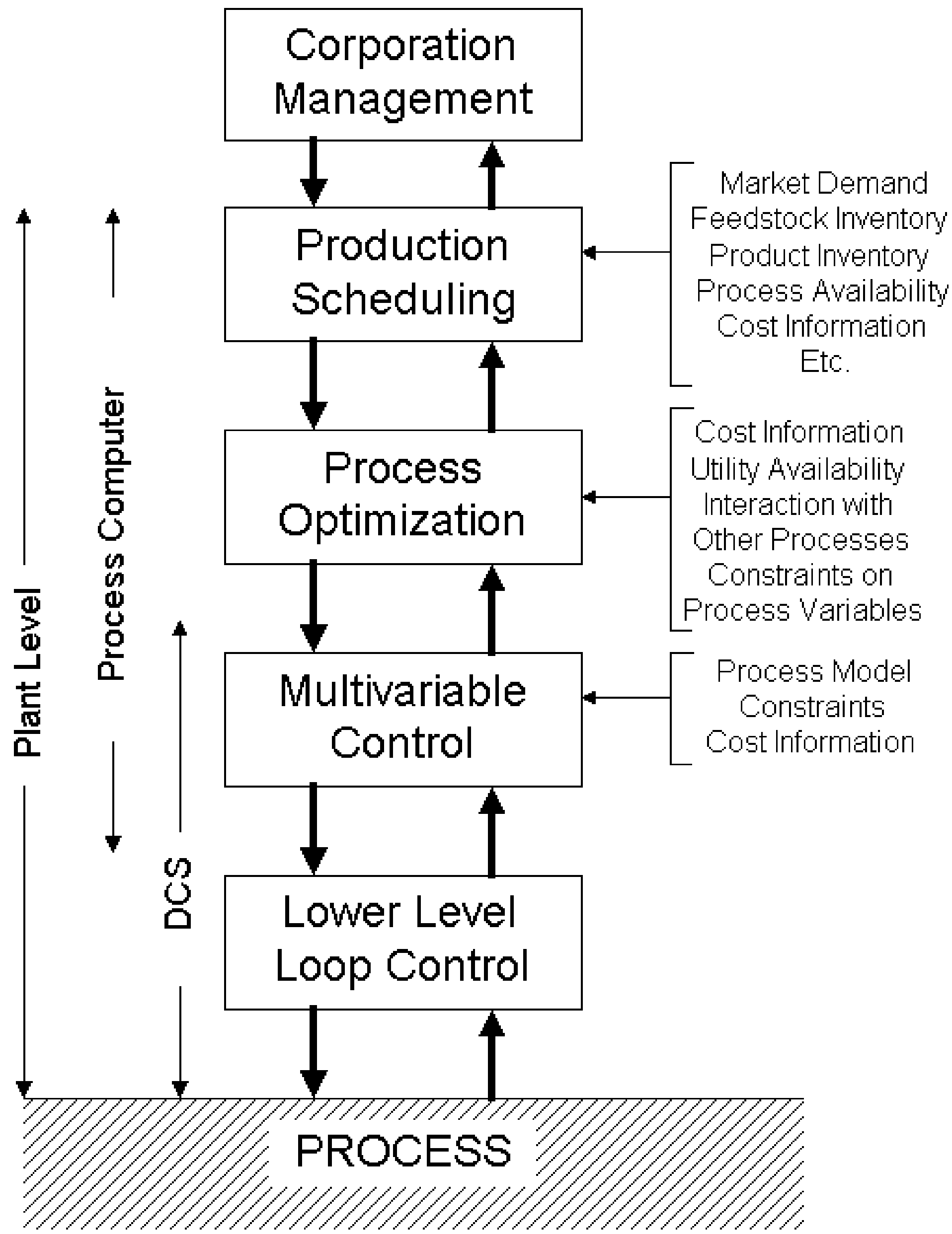
$$
\begin{aligned}
D_{21} &= (1+\delta)\left(-\frac{G_{12}}{G_{11}}\right) \\
D_{12} &= (1+\delta)\left(-\frac{G_{21}}{G_{22}}\right)
\end{aligned}
$$



- The decoupler can be designed at the output.
- To design decoupling control, *Process Model* is required !
  Why not try other MIMO control techniques ?

3.4 CURRENT TRENDS

3.4.1 COMPUTER INTEGRATED PROCESS MANAGEMENT



3.4.2 MODEL-BASED APPROACHES

- At every level of the previous CIM hierachy possibly except low-level loop control and corporation management, *process models* play very important roles.

- **Multivariable Control**

  – In multivariable control, modelless approaches such as decentralized PID has limitations in achievable performance. *Model-based approach* is indeed mandatory.

  – In SISO control, too, controllers designed based on a process model always shows better performance than modelless ones. Especially, for processes with long time-delay and/or inverse response, performance of PID control appears very poor.

  – In industries, economically optimal operating conditions are frequently given at intersetions of process constraints. For this reason, consitraint handling is an important part of multivariable control. This problem is most effectively handled with model-based approaches.

  – *Model Predictive Control* is a convenient and effefctive vehicle to embrace all the above requirements in a single framework and has now become the industrial standard for multivariable control.

- **On-line Optimization**

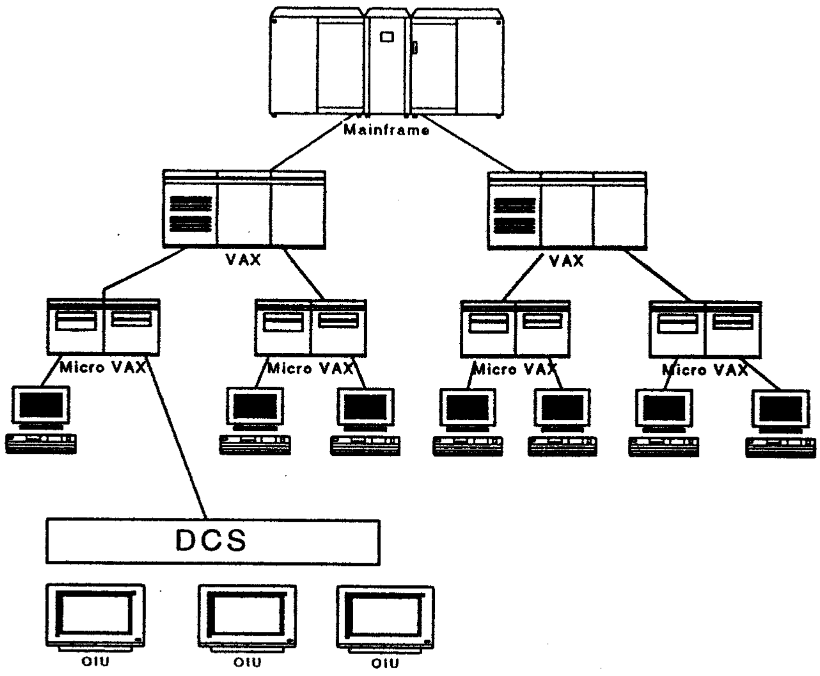
  - Dependable on-line process optimization has been a long-cherished desire in the process industries.

  - Success of on-line optimization definitely relies on the quality of the process model.

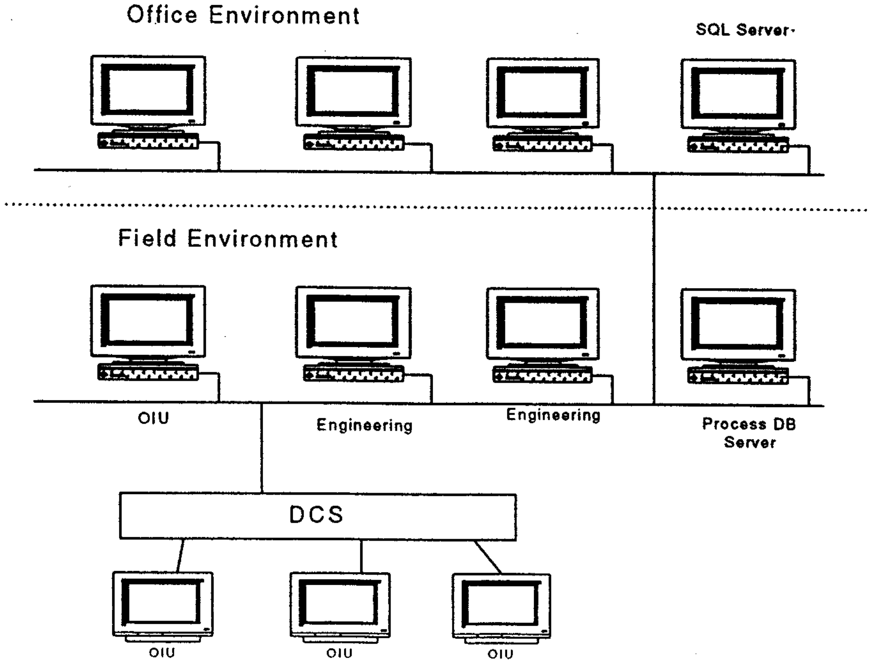
  - Thanks to recent advances in system identification and parameter estimation techniques, creation of a reliable model tuned to an industrial process is now much easier than before. As a consequence, on-line optimization is becoming a practical gear which makes existing processes more profitable.

3.4.3 COMPUTING ENVIRONMENT

Past : vertical hierchy

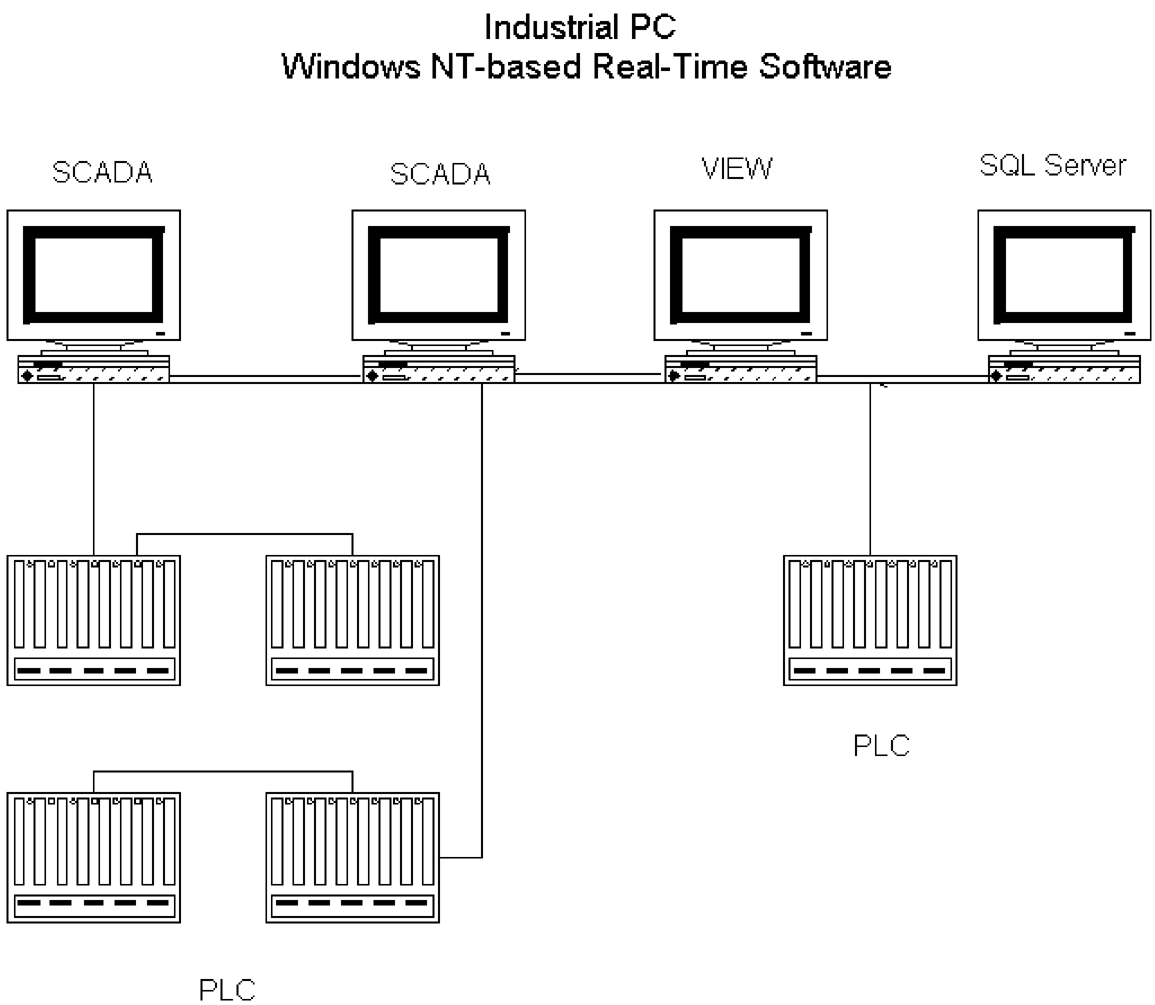


Present : Networked server-client



3.4.4 COMPUTER CONTROL SYSTEM

- From proprietary systems to open architectured systems
- Popularization of PC-PLC systems

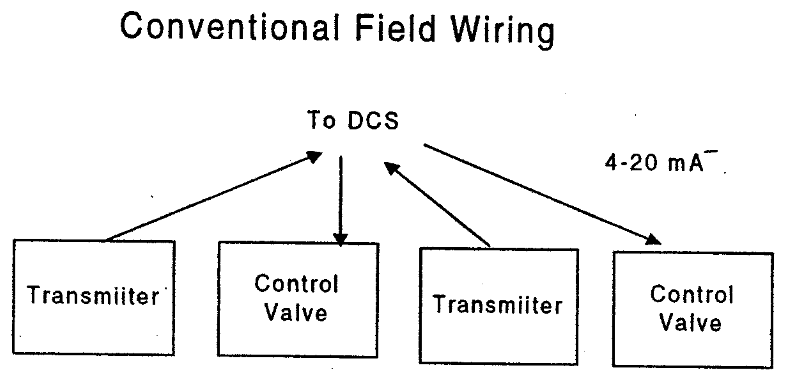


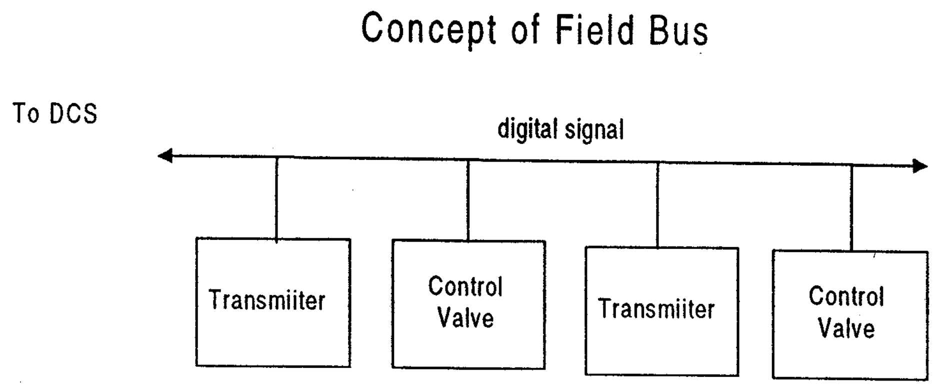
3.4.5 SMART INSTRUMENT AND FIELD BUS

Smart Transmiter

- Includes a CPU and additional ancillary sensors
- Self-diagnosis, self-calibration, linearization, enhanced accuracy by digital signal processing

Field bus





- With the field-bus
  - Wiring cost can be greatly reduced.
  - diaglog between the field instruments and the computer control system is possible.