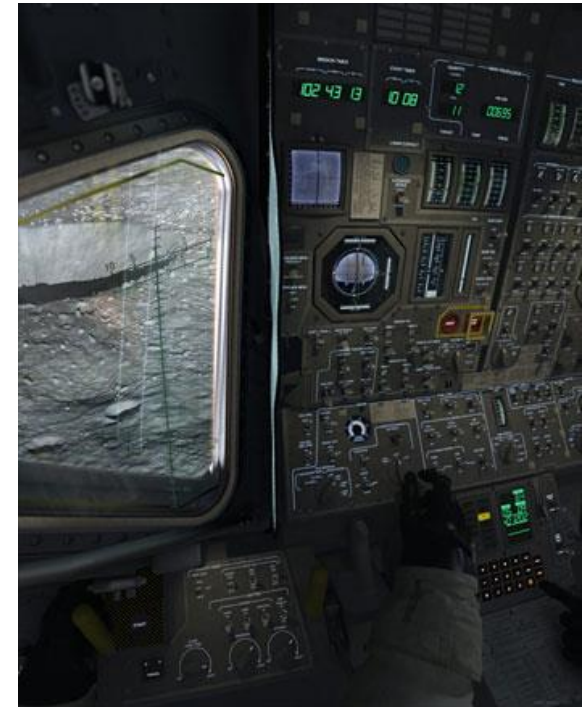
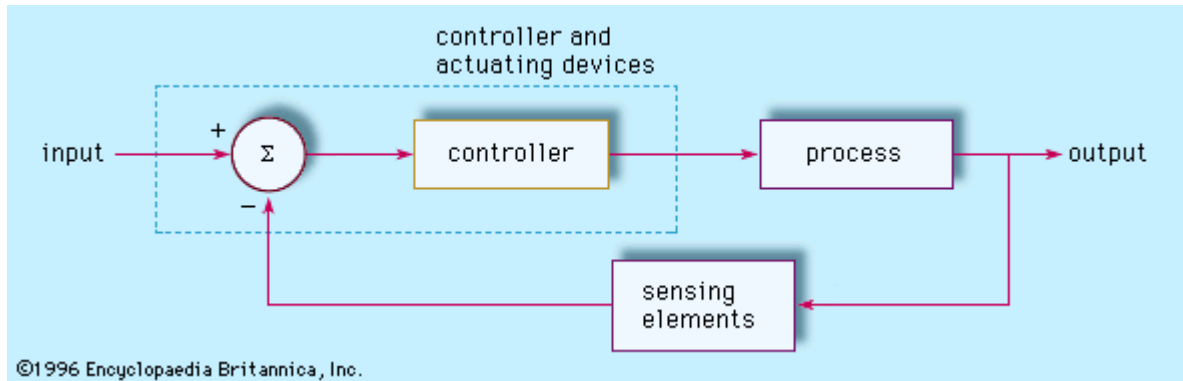
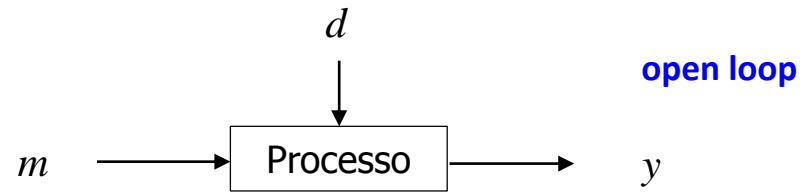


# Introduzione al controllo in retroazione

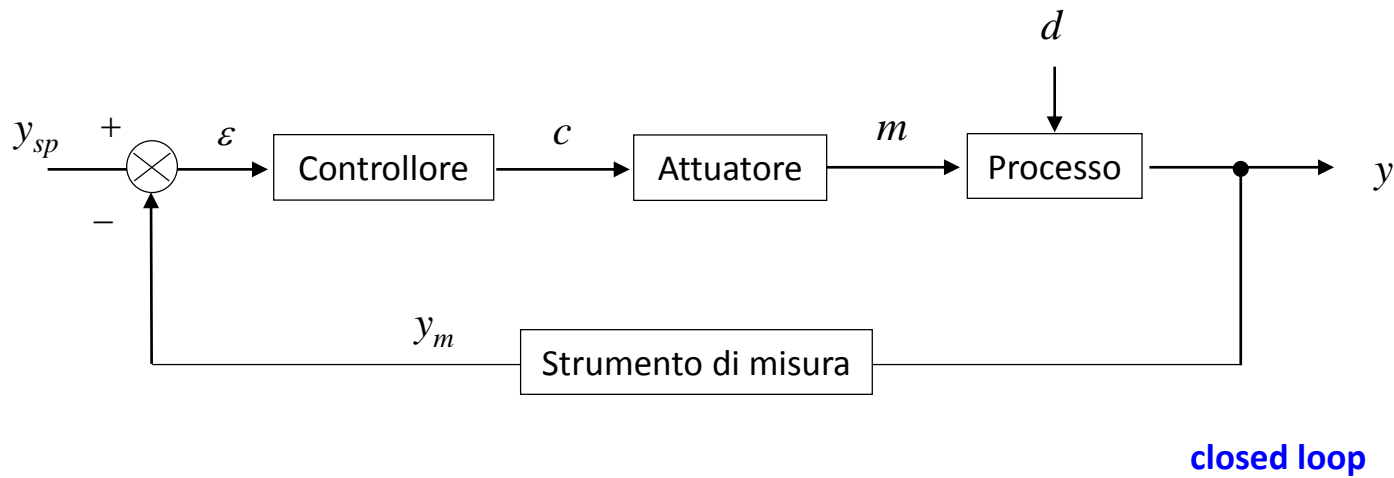


# Introduzione al controllo in retroazione

Dato il processo:



caratterizzato da un output  $y$ , da un disturbo  $d$  e da una possibile variabile manipolata  $m$ , la corrispondente **struttura a blocchi** del sistema di **controllo in retroazione** è:



# Introduzione al controllo in retroazione

In genere i disturbi  $d$  cambiano in modo imprevisto e l'obiettivo del sistema di controllo è mantenere la risposta del processo  $y$  ad un valore prestabilito  $y_{sp}$ .

Un sistema di controllo in retroazione funziona nel seguente modo:

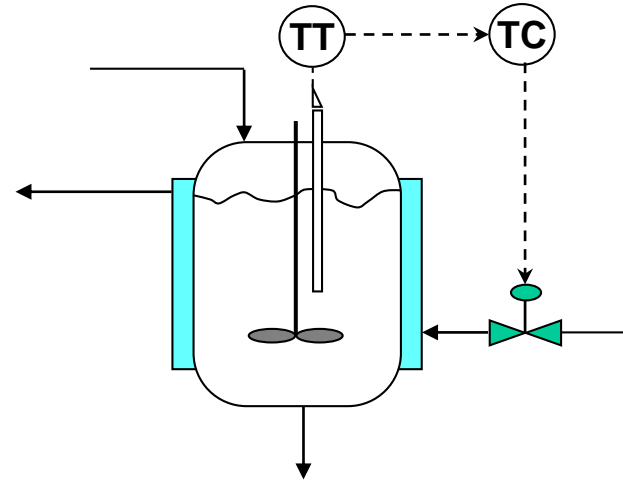
- misura il valore  $y_m$  della variabile di output  $y$  (e.g., pressione, livello, temperatura, composizione) tramite un opportuno strumento di misura;
- confronta il valore  $y_m$  con quello desiderato  $y_{sp}$  (i.e. setpoint) e quantifica l'errore  $\varepsilon = y_{sp} - y_m$ ;
- l'errore  $\varepsilon$  rappresenta il dato di input per il sistema di controllo, il quale propone come output il valore della variabile manipolata  $m$ ;
- il valore  $m$  viene fornito all'elemento finale della catena di controllo (solitamente una valvola) che attua l'azione di controllo sul processo.



# Introduzione al controllo in retroazione

I loop di controllo vengono spesso rappresentati con i seguenti acronimi:

- **FC** = flow control
- **PC** = pressure control
- **LC** = level control
- **TC** = temperature control
- **CC** = composition control



Analogamente si indicano con gli acronimi **LT**, **TT**, **PT**, **FT** e **CT** i dispositivi di misura e trasmissione dati (T = transmitter).

# Hardware di sistema controllo in retroazione

**Processo:** apparecchiature quali serbatoi, scambiatori, reattori, colonne, separatori, compressori, pompe, ...

**Strumenti di misura e sensori:** termocoppie, misuratori di pressione, gascromatografi, analizzatori, ...

**Linee di trasmissione:** dal sensore al controllore e dal controllore all'attuatore sono di due tipi: pneumatiche (obsolete) o elettriche (analogiche o digitali)

**Controllore:** composto dal comparatore e dalla logica di controllo che determina il valore dell'azione di controllo

**Attuatore:** detto anche elemento finale di controllo, di solito una valvola o una pompa di regolazione. Spesso le valvole sono pneumatiche, ove lo stelo che regola il flusso è mosso da un diaframma modificato dalla pressione dell'aria.

Si hanno valvole *air-to-open* e valvole *air-to-close*.



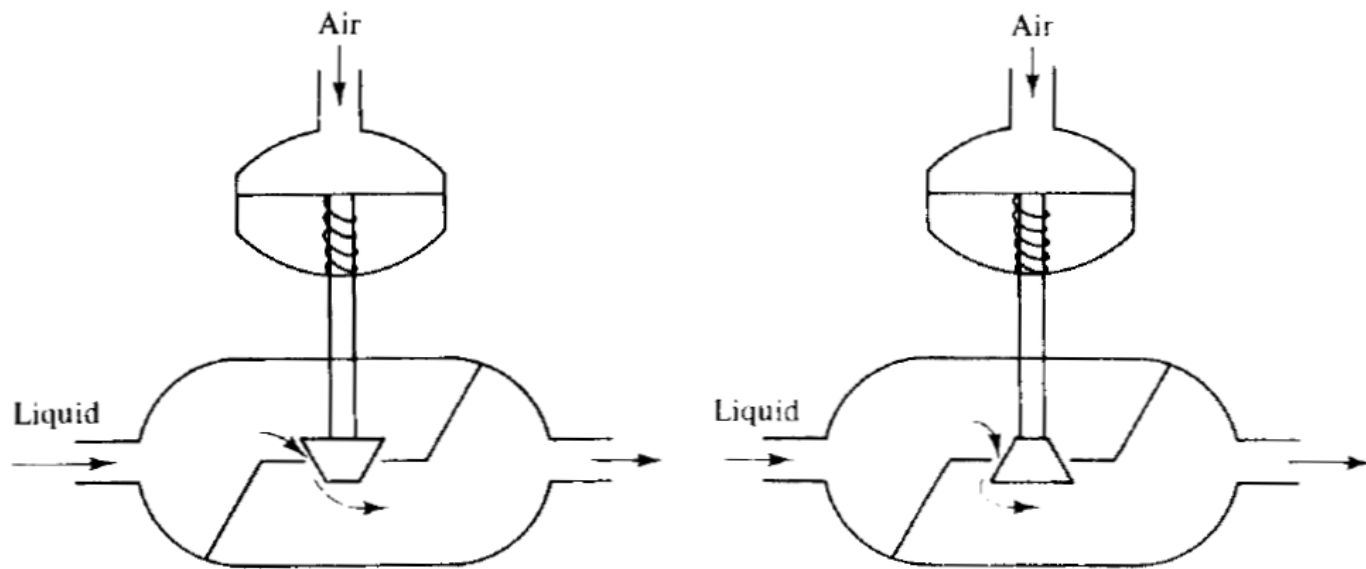
# Valvole pneumatiche

Nel caso di valvole air-to-close, se il **diaframma pneumatico** perde la pressurizzazione, esse risultano “fail open”

Al contrario, le valvole air-to-open risultano “fail closed”



# Valvole pneumatiche



**air-to-close**

**fail open**

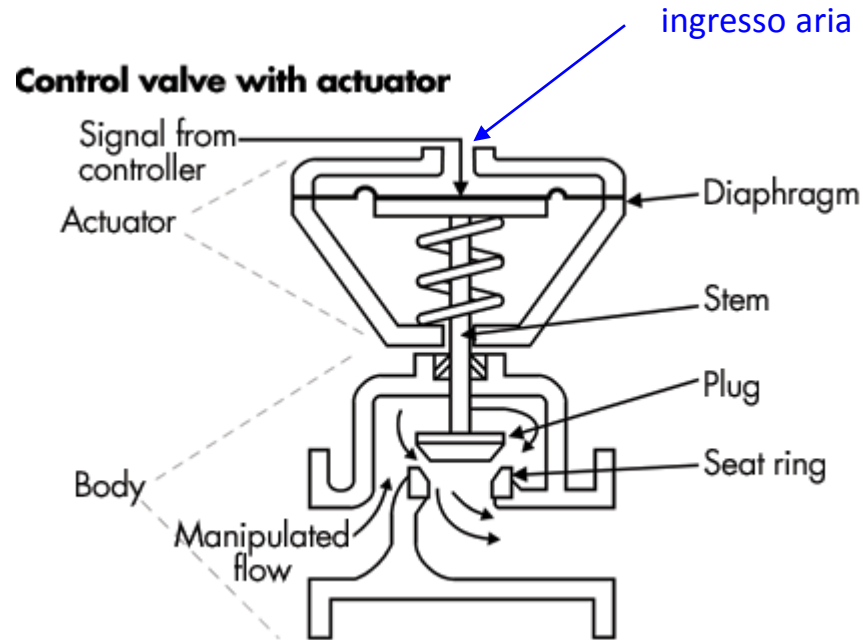
**air-to-open**

**fail closed**

Tratto da: Stephanopoulos, 1984

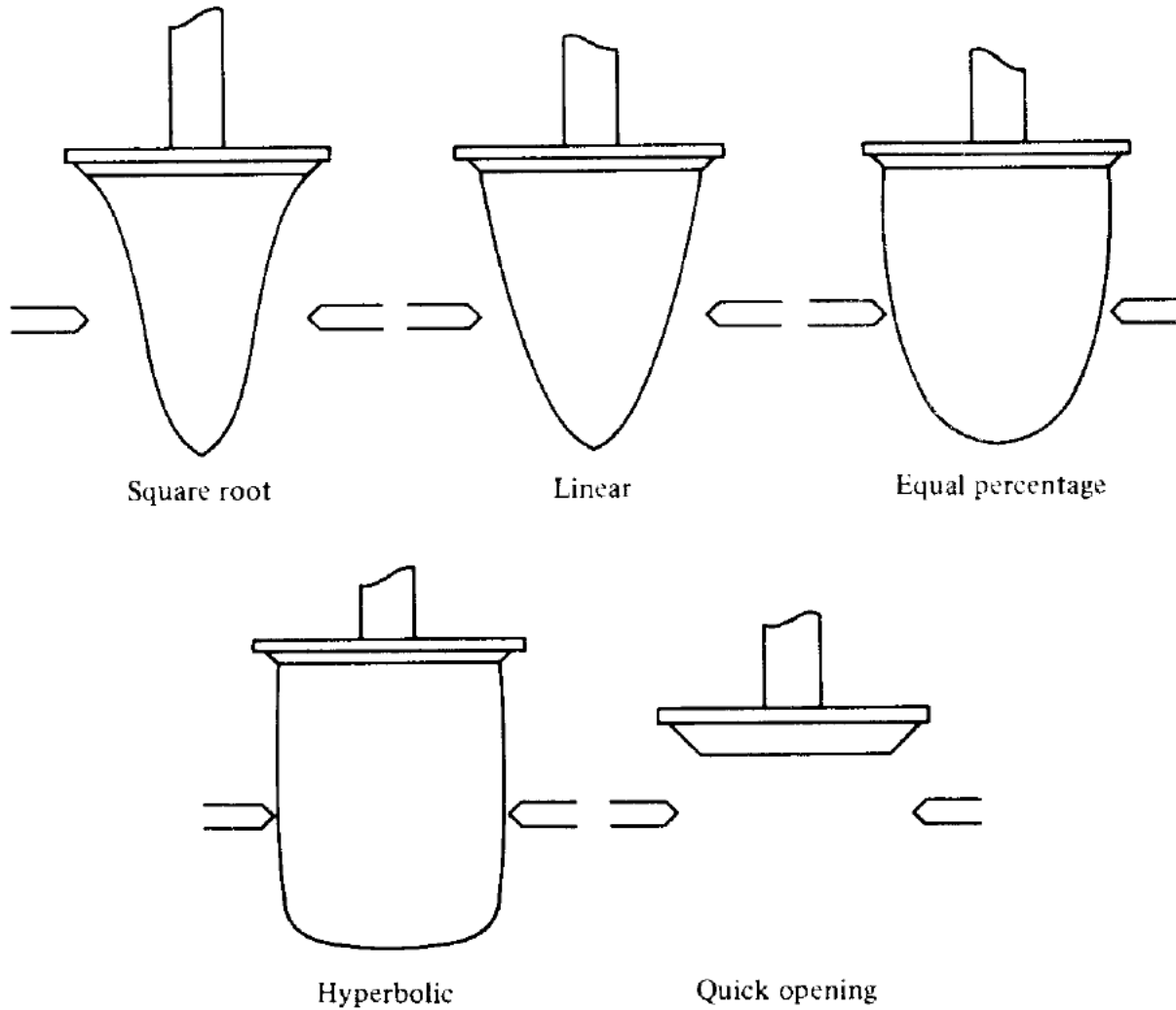
# Valvole pneumatiche

Esempio di valvola pneumatica air-to-close e quindi fail-open.



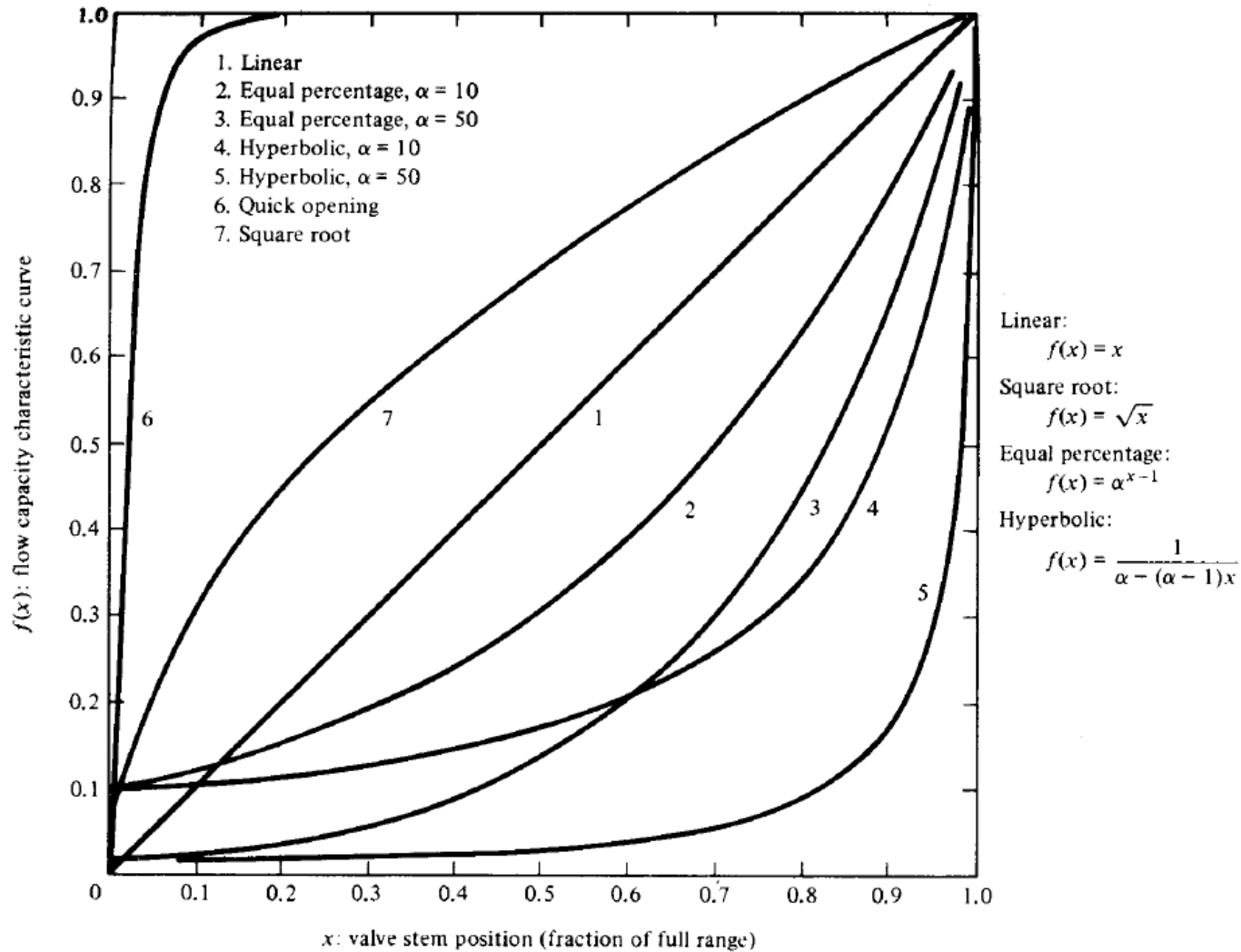


# Valvole pneumatiche: tipologie di otturatori



Tratto da: Stephanopoulos, 1984

# Valvole pneumatiche: equazioni caratteristiche



Tratto da: Stephanopoulos, 1984



# Tipologie di controllori in retroazione

Tra lo strumento di misura e l'attuatore si posiziona il controllore. Esso riceve il segnale misurato  $y_m(t)$  e dopo averlo confrontato con il valore di setpoint  $y_{sp}$  produce un segnale di attuazione  $c(t)$ .

L'input al controllore è l'errore  $\varepsilon(t) = y_{sp} - y_m(t)$  mentre l'output è  $c(t)$ .

Il segnale di output del controllore può essere:

- un segnale pneumatico (aria compressa)
- un segnale elettrico (controllore elettronico)

Esistono **tre tipologie di controllori in retroazione**:

- controllore proporzionale (P)
- controllore proporzionale-integrale (PI)
- controllore proporzionale-integrale-derivativo (PID)



# Controllore proporzionale

È caratterizzato dall'azione di controllo:

$$c(t) = K_c \varepsilon(t) + c_s$$

Ove:

- $K_c$  è il guadagno proporzionale del controllore,
- $c_s$  è il segnale di bias del controllore

Si definisce banda proporzionale **PB** la grandezza  $PB = 100/K_c$

In genere  $PB$  appartiene all'intervallo:  $1 \leq PB \leq 500$

Ne consegue che:  $0.2 \leq K_c \leq 100$

**N.B.:** più  $K_c$  è elevato maggiore è la sensibilità del controllore proporzionale rispetto all'errore  $\varepsilon$  rilevato.



# Controllore proporzionale-integrale

È caratterizzato dall'azione di controllo:

$$c(t) = K_c \varepsilon(t) + \frac{K_c}{\tau_I} \int_0^t \varepsilon(t) dt + c_s$$

Ove:

- $\tau_I$  è la costante di tempo integrale o tempo di azzeramento (reset time)

In genere:  $0.1 \leq \tau_I \leq 50$  min

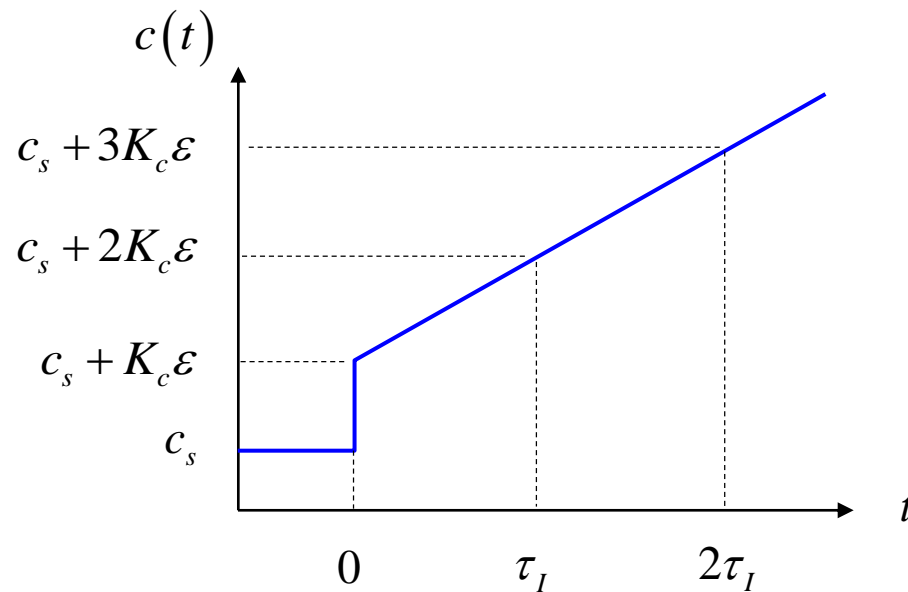


# Controllore proporzionale-integrale

Cerchiamo ora di dare una spiegazione del perché  $\tau_I$  venga detto tempo di azzeramento.

**Hp:** si supponga che l'errore cambi al tempo 0 con un gradino di ampiezza costante  $\varepsilon$ .

L'azione del controllore PI risulta essere:



# Controllore proporzionale-integrale

All'inizio ( $t=0$ ) la risposta del controllore è data dal solo termine proporzionale e risulta essere:  $K_c \varepsilon$ .

Dopo un tempo  $\tau_I$  il termine integrale contribuisce per:

$$\frac{K_c}{\tau_I} \int_0^{\tau_I} \varepsilon(t) dt = \frac{K_c}{\tau_I} \int_0^{\tau_I} \varepsilon dt = \frac{K_c}{\tau_I} \varepsilon \tau_I = K_c \varepsilon$$

Tale contributo è uguale a quello del termine proporzionale.

Si può affermare che ad ogni tempo  $\tau_I$ , sotto l'ipotesi di errore costante nel tempo, l'azione integrale ripete quella del termine proporzionale.

**N.B.:** l'azione integrale prosegue fintantoché esiste anche un minimo errore  $\varepsilon$  tra la variabile controllata ed il suo valore di setpoint.

**N.B.:** un controllore PI è in grado di eliminare anche piccoli errori.



# Integral windup

Come detto in precedenza, il controllore PI continua a fornire una risposta non nulla fintantoché l'errore  $\varepsilon$  è diverso da zero.

Succede di frequente che gli errori non vengano eliminati velocemente, sicché l'azione integrale continua ad accrescere il suo contributo finché l'azione di controllo raggiunge la  **saturazione**  che consiste ad esempio nell'avere una  valvola completamente aperta o completamente chiusa .

Questa condizione di saturazione della risposta integrale è detta:  **integral windup** .

Per esempio: l'integral windup può accadere nel corso di conduzione del processo in modalità manuale durante le fasi di avvio o di spegnimento. Quando il processo torna ad operare in modalità automatica, l'azione di controllo resta saturata e la risposta del sistema subisce ampi  *overshoot* .

Per questi motivi  l'azione integrale  di un controllore PI deve essere  saltuariamente azzerata  per dimenticare la storia passata e ripartire da una condizione operativa consona.





# Controllore proporzionale-integrale-derivativo

È caratterizzato dall'azione di controllo:

$$c(t) = K_c \varepsilon(t) + \frac{K_c}{\tau_I} \int_0^t \varepsilon(t) dt + K_c \tau_D \frac{d\varepsilon}{dt} + c_s$$

Ove:

- $\tau_D$  è la costante di tempo derivativa

Il controllore PID è in grado di anticipare quanto varrà l'errore nell'immediato futuro tramite il termine derivativo.

Due sono gli svantaggi introdotti dal termine derivativo:

- se il sistema risponde con un errore costante non nullo la derivata è nulla e quindi il termine derivativo non ha azione pratica;
- se la risposta del sistema è caratterizzata da rumore con un errore praticamente nullo, l'azione derivativa può essere comunque elevata o addirittura ingente e quindi introdurre disturbi e possibili instabilità.



# Appendice: sistema di guida dell'Apollo 11

We often hear that the on-board Apollo Guidance Computer (AGC) had less power than a modern digital watch; but it all depends on what we mean by 'power'. In its way, it was one of the most capable computers ever invented. It absorbed data from a complex gyroscopic inertial navigation system, allied to an optical star telescope and two radar range finders. It also mediated between the astronauts and the thrusters and rocket engines that drove the ship through space. Apollo's Command Module (Columbia) and Lunar Module (Eagle) had an AGC each, but it's the latter's that has attracted the most attention over the 40 years since it took Neil Armstrong and Buzz Aldrin from lunar orbit to their landing on the Moon's surface.

Translating the AGC's capacity into modern computing parlance can be misleading, but its magnetic core ROM stored the entire guidance programme in the equivalent of about 36 kilobytes. Each tiny ring-shaped core acted as a miniature transformer. Signals from wires running through a core were interpreted as a '1' while those running past it registered a '0'. Up to 64 wires could be threaded in or around a core.

Nothing was 'stored' in ROM when the computer was switched off, but once activated, it booted up within less than a second. It was a totally 'hard-wired' system, because the software was encoded as patterns of wiring, snaking in and out of the little ring-shaped cores, that could not be overwritten or erased.

Raytheon, the company assigned to build the AGC, was located in a region of eastern Massachusetts that had been renowned since the 19th century for its textile industries. Raytheon made clever use of these local skills. The ROM's many thousands of wires were threaded through the cores by middle-aged women with nimble fingers and a painstaking ability to work from coded knitting patterns.

The computer processor itself depended on a relatively untried device: the integrated circuit (IC). The first examples were invented in 1958 by Jack Kilby of Texas Instruments. A year later, Robert Noyce of Fairchild Semiconductors (and later, founder of Intel) refined the process by putting all the components on a chip of silicon and connecting with copper lines.

It's a cliché that Apollo kick-started the microchip revolution, but there's some truth here. NASA bought up 60 per cent of America's entire output of ICs in the early 1960s, temporarily shoring up an industry for which few other markets yet existed.

The AGC's processor ran on just 5,600 ICs, all in the form of NOR logic gates. NASA insisted that only one gate design be used throughout, so that quality inspectors could check them more easily. This sounds limiting, but by stringing a sufficient number of NORs together in the right way, all the required logic functions could be carried out. The AGC's designers delivered a very capable computer, despite its modest 2K of RAM.



# Appendice: sistema di guida dell'Apollo 11

After Apollo, the distinction blurred between the people who operate complex flight systems, and the systems themselves. **Apollo treated the human as an embedded component within the feedback and control logic of the spacecraft as a whole**; but this meant finding a practical 'man-machine interface.'

It would have been impossible to land without the computer. The LM was balanced on a plume of engine thrust from a single nozzle, like an upright pencil poised precariously on a fingertip.

A pistol-grip 'translation controller' enabled Armstrong to steer the descent engine's nozzle, which was pivoted on gimbals that were nudged by electromechanical actuators, but those were under the AGC's control. Armstrong could not have maintained the balance of an unstable vehicle without having his steering commands refined, at one-tenth second intervals, by the computer.

As the Apollo lunar landings came to an end in the early 1970s, a team at NASA's Dryden Flight Research Centre in California wondered if AGC-derived systems might help create a better partnership between pilots and their aircraft, just as they had between astronauts and lunar landers.

On 25 May 1972, test pilot Gary Krier flew a modified F-8 Crusader jet fighter, knowing that for the first time in history, his joystick commands did not feed directly to the aircraft's hydraulic actuators and flight control surfaces, but first were verified and adjusted by an electronic mind: an AGC, suddenly made available because three Apollo lunar missions, 18, 19, and 20, had been cancelled.

Fly-by-wire is now commonplace, both in high-performance jet fighters and aboard commercial airliners, where a smooth, safe ride, rather than extreme agility, is the aim.



# Riferimenti

- Stephanopoulos G., “Chemical Process Control. An Introduction to Theory and Practice”, Prentice-Hall, Englewood Cliff, (1984)

