

Sistemi di Controllo

Il Controllore Proporzionale-Integrale

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

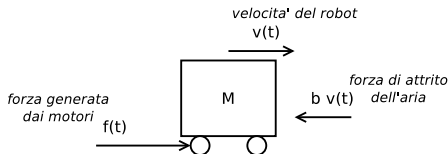
Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



Programmazione Sistemi Robotici

Controllo del Robot

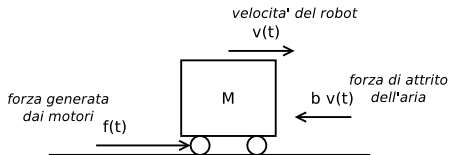


Nel sistema indicato, $f(t)$ è derivante dalla spinta dei motori, provocata a sua volta dall'applicazione di una certa tensione/corrente ai motori stessi.

Sappiamo che il sistema nel discreto è modellato come:

$$\begin{bmatrix} v(k+1) \\ p(k+1) \end{bmatrix} = \begin{bmatrix} -\frac{b\Delta t}{M} + 1 & 0 \\ \Delta t & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ p(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{M} \\ 0 \end{bmatrix} f(k)$$

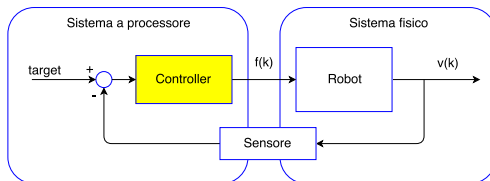
Il Problema del “Controllo”



Desideriamo trovare una $f(k)$ che consenta al robot di andare ad una velocità $\bar{v}$ specificata e di mantenerla

Il Problema del “Controllo”

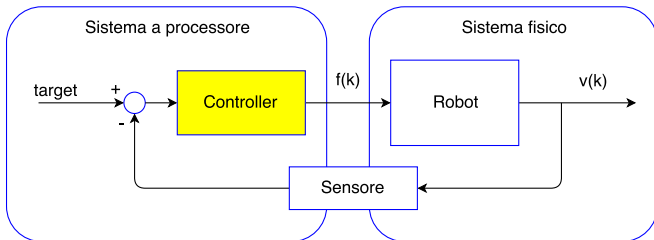
Desideriamo trovare una $f(k)$ che consenta al robot di andare ad una velocità $\bar{v}$ specificata e di mantenerla



Soluzione a Ciclo Chiuso

- Usiamo un **sensore di velocità** e confrontiamo il valore con il target
- Qualora ci sia una **differenza**, usiamo questa **differenza** per determinare un'appropriata $f(k)$

Il Problema del “Controllo”

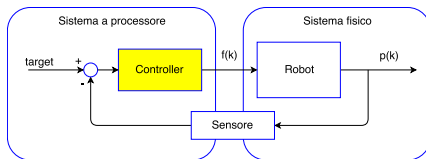


Possiamo usare anche in questo caso un **controllore proporzionale** ?

$$f(k) = K_P e(k) = K_P (\text{target} - v(k))$$

NO!

Rivediamo il Controllo in Posizione ...

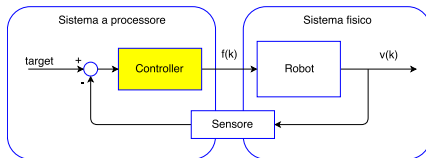


L'output “zero/non-zero” del controllore

$$f(k) = K_P \quad e(k) = K_P (target - p(k))$$

- Quando il target **è raggiunto**, $p(k) = target$ il sistema deve essere **fermo**
- Pertanto è corretto che, in tali condizioni, l'output del controllore sia 0
- Quando il target **non è raggiunto**, il sistema deve essere **azionato** (output non-zero)

Applichiamo al Controllo in Velocità ...

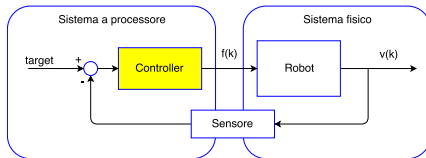


L'output "zero/non-zero" del controllore

$$f(k) = K_P e(k) = K_P (target - v(k))$$

- Quando il target **è raggiunto**, $v(k) = target$, il sistema deve essere **in moto a velocità costante**
- Ma se l'output del controllore è 0, il sistema **si fermerà**, pertanto un semplice controllo proporzionale **non va bene**
- Occorre un controllore che **continui a produrre una spinta** (costante) anche quando il target è stato raggiunto

Applichiamo al Controllo in Velocità ...



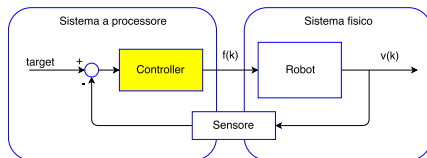
L'output “zero/non-zero” del controllore

- Occorre un controllore che **continui a produrre una spinta** (costante) anche quando il target è stato raggiunto
- **Ad errore zero, la spinta non deve variare**
- Allora mi serve un controllore che faccia **variare la spinta** in modo proporzionale all'errore, secondo un coefficiente Q

$$f(k) = f(k - 1) + Q e(k)$$

- Il controllore deve essere un “**accumulatore di errore**”, in altri termini, un **integratore**

Controllore Integratore



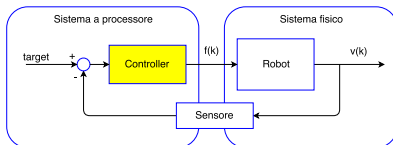
Controllore "I"

$$f(t) = K_I \int_0^t e(\tau) d\tau$$

Discretizzando:

$$f(k) = K_I \sum_{j=0}^k e(j) \Delta t$$

Controllore Proporzionale-Integrale



Controllore “PI”

Combinando il controllore proporzionale e il controllore integrale otteniamo:

$$f(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau$$

Discretizzando:

$$f(k) = K_P e(k) + K_I \sum_{j=0}^k e(j) \Delta t$$

Dove K_P e K_I sono le **costanti di controllo**

Sistemi di Controllo

Il Controllore Proporzionale-Integrale

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



Programmazione Sistemi Robotici