

Sistemi di Controllo

Il Controllore Proporzionale

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

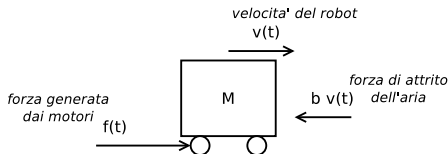
Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



Programmazione Sistemi Robotici

Controllo del Robot

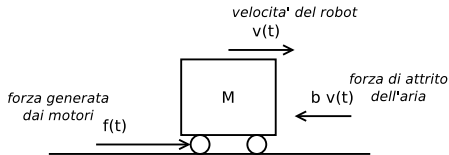


Nel sistema indicato, $f(t)$ è derivante dalla spinta dei motori, provocata a sua volta dall'applicazione di una certa tensione/corrente ai motori stessi.

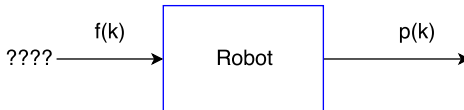
Sappiamo che il sistema nel discreto è modellato come:

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

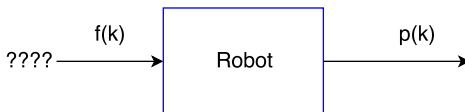
Il Problema del “Controllo”



Desideriamo trovare una $f(k)$ che consenta al robot di raggiungere una posizione $\bar{p}$ specificata; una volta che la posizione è raggiunta, essa deve essere mantenuta



Il Problema del “Controllo”



Desideriamo trovare una $f(k)$ che consenta al robot di raggiungere una posizione $\bar{p}$ specificata; una volta che la posizione è raggiunta, essa deve essere mantenuta

Soluzione 1: Ciclo Aperto

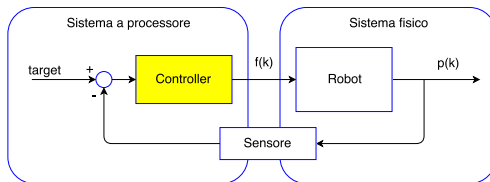
Invertiamo analiticamente il modello e determiniamo la $f(k)$ a partire dalla $\bar{p}$

Non funziona!

- Non è detto che il modello sia (facilmente) invertibile
- Il modello è sempre un'approssimazione della realtà

Il Problema del “Controllo”

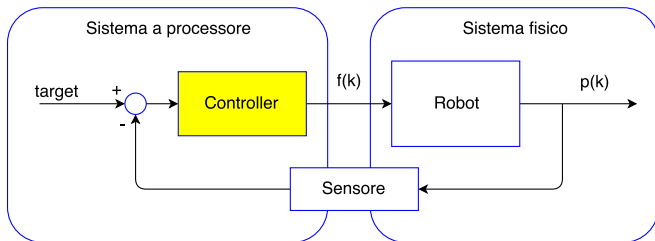
Desideriamo trovare una $f(k)$ che consenta al robot di raggiungere una posizione $\bar{p}$ specificata; una volta che la posizione è raggiunta, essa deve essere mantenuta



Soluzione 2: Ciclo Chiuso

- **Leggiamo** con un sensore la posizione **effettiva**
- **Confrontiamola** con il nostro valore desiderato $\bar{p}$ (target)
- Qualora ci sia una **differenza**, usiamo questa **differenza** per determinare un'appropriata $f(k)$
- Ripetiamo questi tre step **continuamente**, o meglio, **periodicamente**

Il Problema del “Controllo”



```
doPositionControl(target)
```

```
while True do
```

```
    On each  $\Delta T$ ;
```

```
    current  $\leftarrow$  read_position_by_sensor();
```

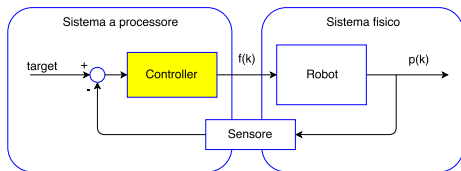
```
    error  $\leftarrow$  target - current;
```

```
    f  $\leftarrow$  controller(error);
```

```
    drive_motors(f);
```

```
end
```

Il Problema del “Controllo”



doPositionControl

while *True* **do**

On each ΔT ;

current $\leftarrow$

read_position_by_sensor();

error $\leftarrow$ *target* $-$ *current*;

f $\leftarrow$ *controller*(*error*);

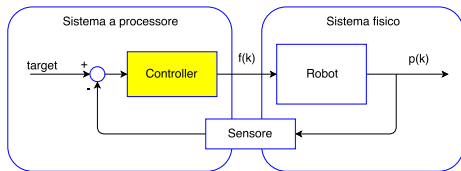
drive_motors(*f*);

end

Funziona!!!!

- Non si basa su un modello teorico ma usa un **dato vero** letto da un sensore
- Non richiede la conoscenza del modello teorico e funziona **su tutti i sistemi**
- Agisce **continuamente** sul sistema e legge l'effetto delle azioni
- In caso di variazioni della posizione **interviene subito** riportandola al target

Il Problema del “Controllo”



doPositionControl

while *True* **do**

 On each ΔT ;

current $\leftarrow$

read_position_by_sensor();

error $\leftarrow$ *target* – *current*;

f $\leftarrow$ **controller**(*error*);

drive_motors(*f*);

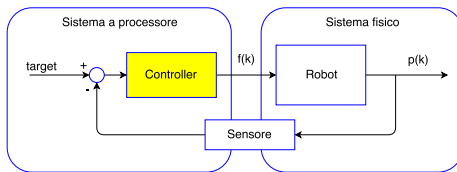
end

Problematiche da risolvere

- La complessità si sposta però dal modello alla **progettazione del blocco di controllo**
- Il controller è comunque un **sistema dinamico** che occorre **sintetizzare** (progettare) e successivamente implementare
- **Com'è fatto il controller?**

Esempio di controllo in posizione

Controllo in Posizione (semplificato)



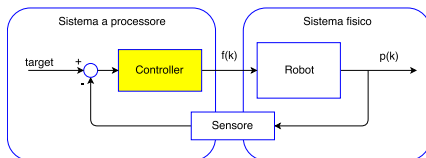
Soluzione 1

- Se l'errore è diverso da zero, usiamo una $f(k)$ costante (positiva o negativa a seconda del segno dell'errore)
- Quando l'errore diventa 0, applichiamo $f(k) = 0$, così fermiamo il sistema

Non Funziona!!

- Il sistema (robot) ha una **dinamica**: quando lo stimolo è 0 il sistema **non si ferma istantaneamente!**
- Il target verrà "superato" provocando dunque una "retromarcia"
- Il comportamento a regime sarà **oscillatorio**

Controllo in Posizione (semplificato)



Soluzione 2

- Per cercare di rispettare la dinamica, usiamo una $f(k)$ **proporzionale all'errore**
- Se l'errore è grande, lo stimolo applicato sarà grande, facendo sì che il target venga "avvicinato prima possibile"
- Se siamo in prossimità del target applichiamo un stimolo piccolo

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

Funziona!!

- ... ma occorre "calcolare" o "scegliere" il valore di K_P

Sistemi di Controllo

Il Controllore Proporzionale

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



Programmazione Sistemi Robotici