

Sistemi di Controllo

Controllo in Posizione con Controllore Proporzionale

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

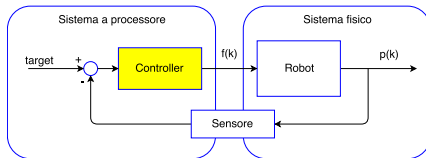
santoro@dmi.unict.it



Programmazione Sistemi Robotici

Controllo in Posizione (semplificato)

- Abbiamo già simulato un controllo in posizione usando un controllore P e lo schema seguente

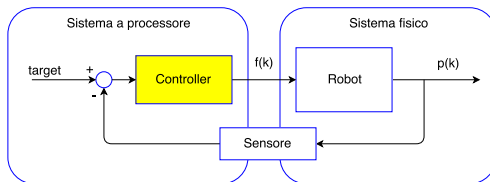


Schema Semplificato

- 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

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

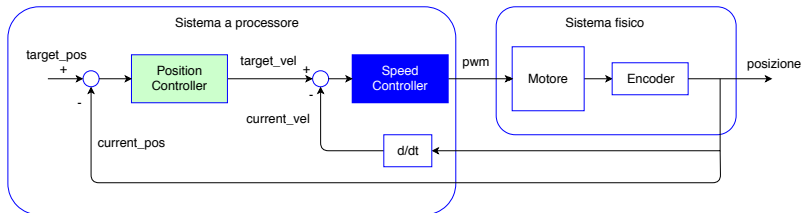
Controllo in Posizione (semplificato)



Problemi dello Schema Semplificato

- Tuttavia questo tipo di controllo **non ci consente** di imporre eventuali vincoli su **accelerazioni** o **decelerazioni**
- La **velocità** infatti **non è** tra le **variabili controllate**
- Occorre pertanto uno **schema di controllo** che consenta di controllare **sia la velocità che la posizione**

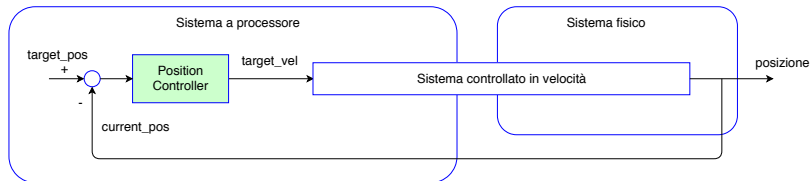
Controllo in Posizione



Uso del controllo in cascata

- Lo schema usato è quello dei **controllori in cascata**
- Un **controllore di posizione** più esterno (**outer loop**) che fornisce il **riferimento** (di velocità) a un **controllore di velocità** interno (**inner loop**)

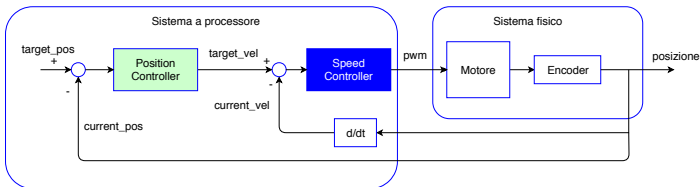
Controllo in Posizione



Controllo in velocità come “black box”

- In altri termini, possiamo considerare il **sistema di controllo in velocità** come una **black box** che garantisce il rispetto della velocità target
- Il **controllore di posizione** fornisce pertanto il **segnale di pilotaggio** per la “black box”

Controllo in Posizione



Algoritmo

while *True* **do**

On each ΔT ;

current_pos \leftarrow *read_encoder()*;

current_speed $\leftarrow \frac{\Delta \text{current_pos}}{\Delta t}$;

pos_error \leftarrow *target_pos* - *current_pos*;

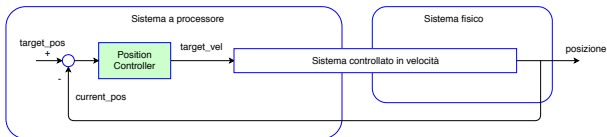
target_speed \leftarrow *position_controller*(*pos_error*);

speed_error \leftarrow *target_speed* - *current_speed*;

pwm \leftarrow *speed_controller*(*speed_error*);

drive_motor(*pwm*);

end



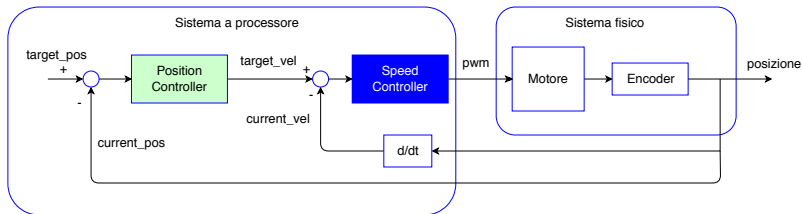
Tipologia del controllore in posizione

- Concettualmente: se “siamo lontani” una velocità grande, che diminuisce man mano che ci si avvicina
- Praticamente: usiamo un **controllore proporzionale**:

$$target_speed = K_P (target_pos - current_pos)$$

- Tuttavia, *oltre una certa velocità* non potremo andare
- Pertanto è necessario inserire una **saturatione**

Controllo in Posizione e Velocità



Sommario

- Controllo in velocità: controllore **PI + Saturazione**
- Controllo in posizione: controllore **P + Saturazione**

Sistemi di Controllo

Controllo in Posizione con 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