

Cinematica e Controllo di un robot mobile

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

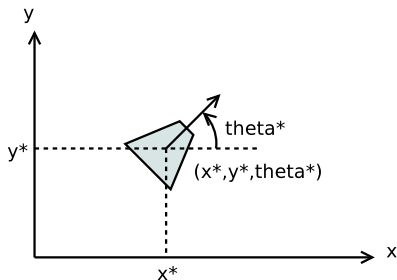
santoro@dmi.unict.it



Programmazione Sistemi Robotici

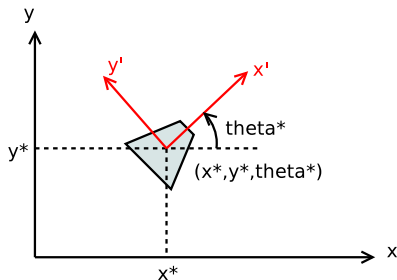
Modello Cinematico del Robot Mobile a Due Ruote

Cinematica di Robot Mobile su Ruote



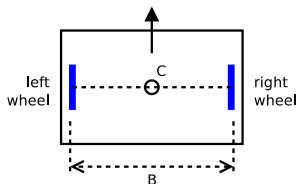
- La **posa** istantanea di un robot su un ambiente **bidimensionale** è caratterizzata da:
 - Posizione x
 - Posizione y
 - Orientamento θ (rispetto all'asse x)

Cinematica di Robot Mobile su Ruote



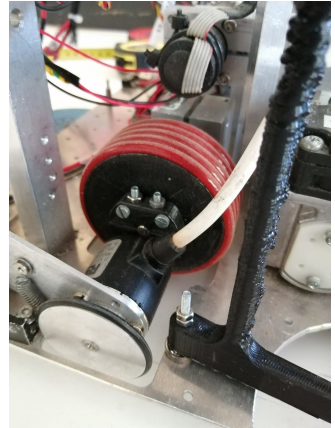
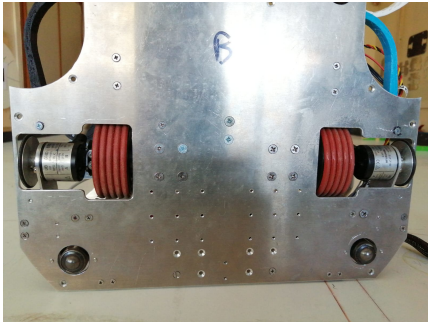
- Si considerano due sistemi di riferimento:
 - **Body system**, sistema di riferimento *locale* del robot (x', y')
 - **Earth system**, sistema di riferimento *globale* dell'ambiente (x, y)
- La trasformazione tra un sistema e l'altro avviene usando le normali formule di roto-traslazione

Modello di Robot Mobile a due ruote indipendenti



- Due ruote indipendenti poste ai lati del robot
- Il punto **C** è detto **centro di istantanea rotazione** ed è il riferimento per la posizione globale del robot
- I parametri di geometria per il modello cinematico sono:
 - r_L , raggio della ruota sinistra
 - r_R , raggio della ruota destra
 - n_L , numero di tick per giro dell'encoder sinistro
 - n_R , numero di tick per giro dell'encoder destro
 - B , distanza tra i punti di contatto delle due ruote (**wheelbase**)

Robot Mobile a due ruote indipendenti



Cinematica del Robot Mobile a due ruote indipendenti

- Parametri:
 - r_L , raggio della ruota sinistra
 - r_R , raggio della ruota destra
 - n_L , numero di tick per giro dell'encoder sinistro
 - n_R , numero di tick per giro dell'encoder destro
 - B , distanza tra i punti di contatto delle due ruote (**wheelbase**)
- Detti $\Delta tick_L$ e $\Delta tick_R$ il numero di **tick contati**, su ruota destra e sinistra, nell'intervallo di campionamento ΔT , abbiamo:

$$\Delta d_L = \frac{2\pi r_L}{n_L} \Delta tick_L \qquad \Delta d_R = \frac{2\pi r_R}{n_R} \Delta tick_R$$

$$v_L = \frac{\Delta d_L}{\Delta T} \qquad v_R = \frac{\Delta d_R}{\Delta T}$$

$$\Delta d = \frac{\Delta d_L + \Delta d_R}{2} \qquad \Delta \theta = \frac{\Delta d_R - \Delta d_L}{B}$$

Cinematica del Robot Mobile a due ruote indipendenti

- Detti $\Delta tick_L$ e $\Delta tick_R$ il numero di **tick contati**, su ruota destra e sinistra, nell'intervallo di campionamento ΔT , abbiamo:

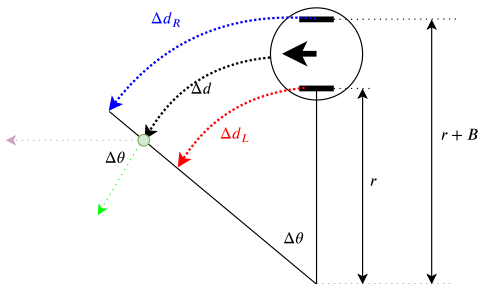
$$\Delta d_L = \frac{2\pi r_L}{n_L} \Delta tick_L \qquad \Delta d_R = \frac{2\pi r_R}{n_R} \Delta tick_R$$

$$v_L = \frac{\Delta d_L}{\Delta T} \qquad v_R = \frac{\Delta d_R}{\Delta T}$$

$$\Delta d = \frac{\Delta d_L + \Delta d_R}{2} \qquad \Delta \theta = \frac{\Delta d_R - \Delta d_L}{B}$$

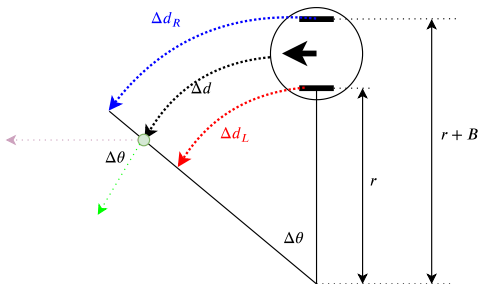
- Dove:
 - $\Delta d_L, \Delta d_R$, distanza percorsa (in ΔT) dalla ruota sinistra/destra
 - v_L, v_R , velocità ruota sinistra/destra
 - Δd , distanza percorsa
 - $\Delta \theta$, variazione dell'orientamento (in radianti)

Calcolo di Δd e $\Delta\theta$



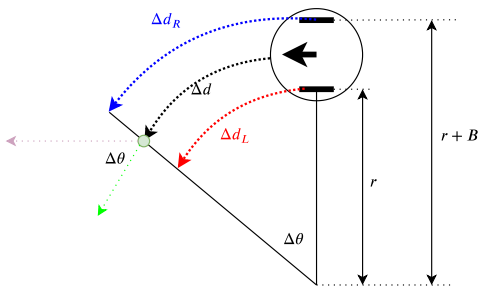
- Possiamo considerare che il moto del robot sia un insieme di **traiettorie circolari** infinitesime

Calcolo di Δd e $\Delta\theta$



- Data la traiettoria circolare in figura, abbiamo:
 - Δd_L , la distanza percorsa da ruota sinistra (arco di circonferenza)
 - Δd_R , la distanza percorsa da ruota destra (arco di circonferenza)
 - $\Delta\theta$, l'angolo relativo all'arco di circonferenza percorso da ruote la distanza percorsa da ruota destra
 - $\Delta\theta$, è anche la **variazione di orientamento** del robot
 - r , raggio dell'arco percorso da ruota sinistra
 - $r + B$, raggio dell'arco percorso da ruota destra

Calcolo di Δd e $\Delta\theta$



$$\Delta d_L = r\Delta\theta$$

$$\Delta d_R = (r + B)\Delta\theta$$

$$r = \frac{\Delta d_L}{\Delta\theta} \Rightarrow \Delta d_R = \left(\frac{\Delta d_L}{\Delta\theta} + B\right)\Delta\theta$$

$$\Delta d_R = \Delta d_L + B\Delta\theta \Rightarrow \Delta\theta = \frac{\Delta d_R - \Delta d_L}{B}$$

$$\Delta d = \frac{\Delta d_L + \Delta d_R}{2} \quad \Delta \theta = \frac{\Delta d_R - \Delta d_L}{B}$$

- Sulla base delle relazioni precedenti, l'aggiornamento della posa si ottiene con le seguenti relazioni:

$$x = x + \Delta d \cos\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$y = y + \Delta d \sin\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$\theta = \theta + \Delta \theta$$

$$\Delta d = \frac{\Delta d_L + \Delta d_R}{2}$$

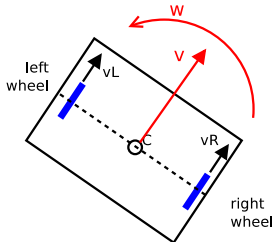
$$\Delta \theta = \frac{\Delta d_R - \Delta d_L}{B}$$

$$x = x + \Delta d \cos\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$y = y + \Delta d \sin\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$\theta = \theta + \Delta \theta$$

- Le relazioni precedenti, note come **formule per l'odometria**, si determinano tramite l'**integrazione dell'equazione del moto**
- Tale integrazione (che non può essere effettuata analiticamente) è effettuata **numericamente** e pertanto contiene un **errore che si propaga in modo additivo** nella stima della posa (x, y, θ)
- La posa stimata (x, y, θ) è pertanto soggetta ad un errore non nullo e non eliminabile che cresce con l'andare del tempo/distanza percorsa

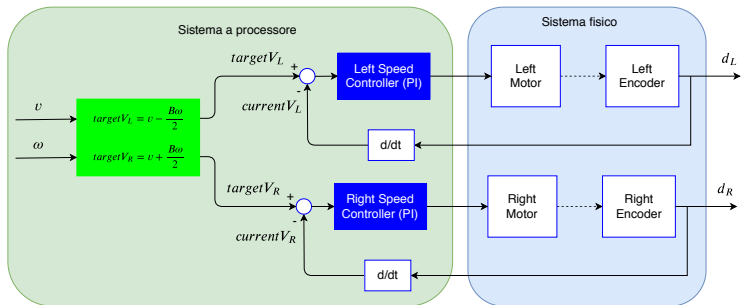


- La **velocità** è invece rappresentata secondo due possibili sistemi di riferimento
 - (v_L, v_R) , velocità istantanea delle rotaie sinistra e destra
 - (v, ω) , velocità istantanea di **traslazione** e **rotazione** del punto C
- Tra i due sistemi di riferimento esiste la seguente trasformazione:

$$\begin{aligned} v &= \frac{v_R + v_L}{2} & \omega &= \frac{v_R - v_L}{B} \\ v_L &= v - \frac{B\omega}{2} & v_R &= v + \frac{B\omega}{2} \end{aligned}$$

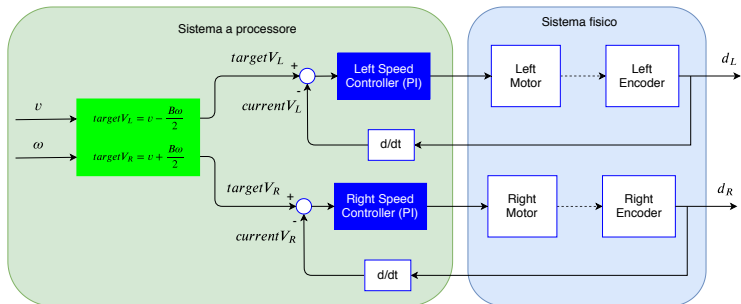
Controllo del Robot Mobile a Due Ruote

Modello di Controllo in Velocità



- Per il **controllo del moto**, il primo step è l'uso di due **controlli in velocità indipendenti** sulle due ruote
- Tali controlli sono poi pilotati da un blocco che converte (v, ω) desiderati in (V_L, V_R)
- Attraverso il pilotaggio di (v, ω) è possibile implementare differenti movimenti

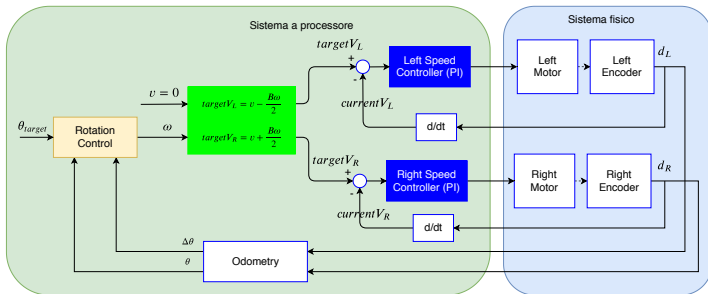
Controllo del Moto



- Se $v = 0$ e ω è impostata pilotata esternamente, il moto è **rotazione sul posto**
- Se $\omega = 0$ e v è pilotata, il moto è **rettilineo in direzione θ** (orientamento corrente del robot)
- Se entrambe v e ω sono **pilotate**, il moto è **curvilineo**

Rotazioni

Rotazioni



- Il blocco **Rotation Control** implementa i possibili algoritmi di controllo delle rotazioni
- Le rotazioni possono essere **assolute** (raggiungimento di un θ_{target} specificato) o **relative** (rotazione complessiva un angolo θ_{target} specificato)

Controllo Proporzionale con Saturazione

```
while True do
  On each  $\Delta T$ ;
   $\theta_{error} \leftarrow \theta_{target} \ominus \theta$ ;
   $\omega_{target} \leftarrow K_{P\_ROT} \cdot \theta_{error}$ ;
   $\omega_{target} \leftarrow \text{saturate}(\omega_{target}, \omega_{max})$ ;
end
```

Controllo con Profilo di Velocità

```
while True do
  On each  $\Delta T$ ;
   $\omega_{target} \leftarrow \text{profile\_angle\_controller}(\theta_{target}, \theta, \omega)$ ;
end
```

Rotazioni, angoli e differenze

- Nell'usare gli angoli, occorre considerare che la loro aritmetica è **modulare**
- Ad esempio, i valori (in gradi) **350** e **-10** si riferiscono allo **stesso angolo**
- Tuttavia, il **differente segno** implica un comportamento diverso del controllore che genererà velocità **positive** o **negative**
- Nel calcolo degli angoli, è pertanto opportuno **normalizzare** sempre il risultato, riconducendolo all'intervallo $[-\pi, \pi]$ (resp. $[-180, 180]$)
- Esempio: consideriamo $\theta = 170^\circ$ e $\theta_{target} = -170^\circ$:
 - **Non normalizzato:** $\theta_{error} = \theta_{target} - \theta = -340^\circ \Rightarrow$ velocità negativa, rotazione in senso **orario**
 - **Normalizzato:** $\theta_{error} = \theta_{target} - \theta = 20^\circ \Rightarrow$ velocità positiva, rotazione in senso **antiorario**

- Si utilizza il $\Delta\theta$ proveniente dal modulo di odometria per aggiornare l'angolo percorso dall'inizio del moto θ'
- θ' costituisce il valore corrente da confrontare con θ_{target}
- Si applica uno dei possibili controlli in posizione sugli angoli

Controllo Proporzionale con Saturazione

while *True* **do**

 On each ΔT ;

$\theta' \leftarrow \theta' + \Delta\theta$;

$\theta_{error} \leftarrow \theta_{target} - \theta'$;

$\omega_{target} \leftarrow K_{P_ROT} \cdot \theta_{error}$;

$\omega_{target} \leftarrow \text{saturate}(\omega_{target}, \omega_{max})$;

end

Spostamenti

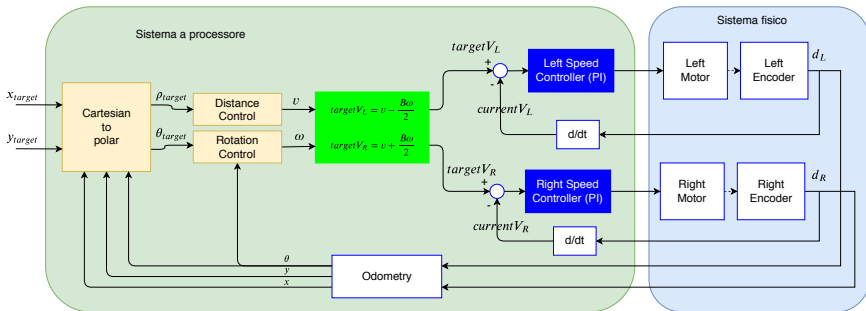
Controllo Polare (raggiungimento punto)

- Desideriamo raggiungere un punto specifico del piano di coordinate (x_{target}, y_{target})
- Trattiamo il problema in coordinate polari considerando il robot come **origine**

$$\begin{aligned}\rho_{target} &= \sqrt{(x_{robot} - x_{target})^2 + (y_{robot} - y_{target})^2} \\ \theta_{target} &= \arctan \frac{y_{target} - y_{robot}}{x_{target} - x_{robot}}\end{aligned}$$

- La coppia $(\rho_{target}, \theta_{target})$ viene utilizzata come target in due controlli di posizione, per determinare, rispettivamente v e ω

Controllo Polare (raggiungimento punto)



Controllo Polare (raggiungimento punto)

Pseudo-codice

while *True* **do**

 On each ΔT ;

$position_error \leftarrow \sqrt{(x_{target} - x)^2 + (y_{target} - y)^2}$;

$v_{target} \leftarrow position_controller(position_error)$;

$\theta_{target} \leftarrow atan2(y_{target} - y, x_{target} - x)$;

$\theta_{error} \leftarrow \theta_{target} \ominus \theta$;

$\omega_{target} \leftarrow angle_controller(\theta_{error})$;

end

Inseguimento di una traiettoria generica (virtual robot)

- Desideriamo muoverci su una traiettoria generica
- Specifichiamo la **legge oraria** $\tilde{x}(t), \tilde{y}(t)$ delle coordinate della traiettoria che un ipotetico robot (virtual robot) sta seguendo
- Discretizziamo la legge: $\tilde{x}(k\Delta T), \tilde{y}(k\Delta T)$
- Utilizziamo i punti $\tilde{x}(k\Delta T), \tilde{y}(k\Delta T)$ come **target temporanei** e applichiamo il **controllo polare**
- Il robot reale procederà via via “inseguendo” il robot virtuale agganciando la sua traiettoria ed arrivando all'eventuale punto finale

Pseudo-codice

while *True* **do**

 On each ΔT ;

$x_p \leftarrow \tilde{x}(k\Delta t)$;

$y_p \leftarrow \tilde{y}(k\Delta t)$;

$k \leftarrow k + 1$;

$position_error \leftarrow \sqrt{(x_p - x)^2 + (y_p - y)^2}$;

$v_{target} \leftarrow position_controller(position_error)$;

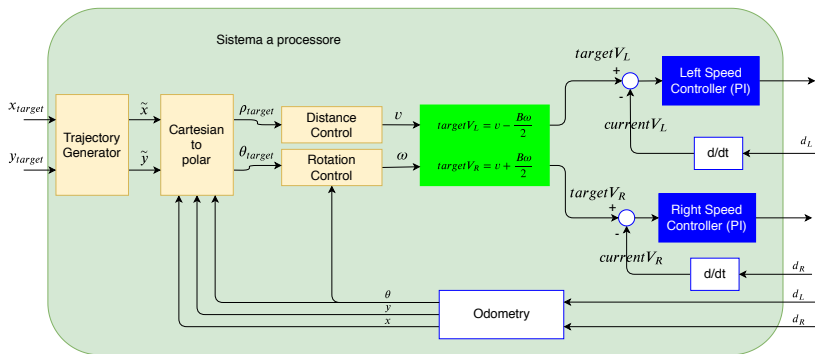
$\theta_{target} \leftarrow atan2(y_p - y, x_p - x)$;

$\theta_{error} \leftarrow \theta_{target} \ominus \theta$;

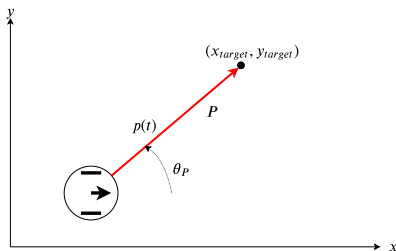
$\omega_{target} \leftarrow angle_controller(\theta_{error})$;

end

Inseguimento di una traiettoria generica (virtual robot)



Raggiungimento punto con virtual robot



- All'inizio del moto, si determina la traiettoria rettilinea P dalla posizione di partenza del robot al punto target (x_{target}, y_{target})
- Lungo P si fa muovere **unidimensionalmente** un punto la cui posizione nel tempo sarà uno scalare $p(t)$
- Si determinano, istante per istante, le coordinate:

$$\tilde{x}(t) = p(t) \cos \theta_P$$

$$\tilde{y}(t) = p(t) \sin \theta_P$$

Cinematica e Controllo di un robot mobile

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory

Dipartimento di Matematica e Informatica - Università di Catania, Italy

santoro@dmi.unict.it



Programmazione Sistemi Robotici