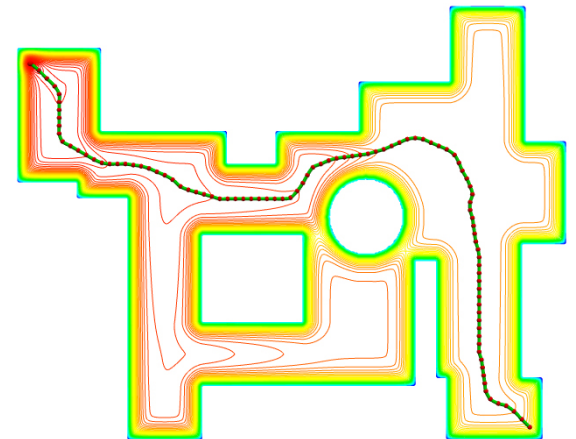


$$q = [x_r, y_r, \theta_r]^T \in \mathbb{R}^3$$

Odometria nei robot “DDD” Differential Dual Drive

Giovanni De Luca

www.delucagiovanni.com

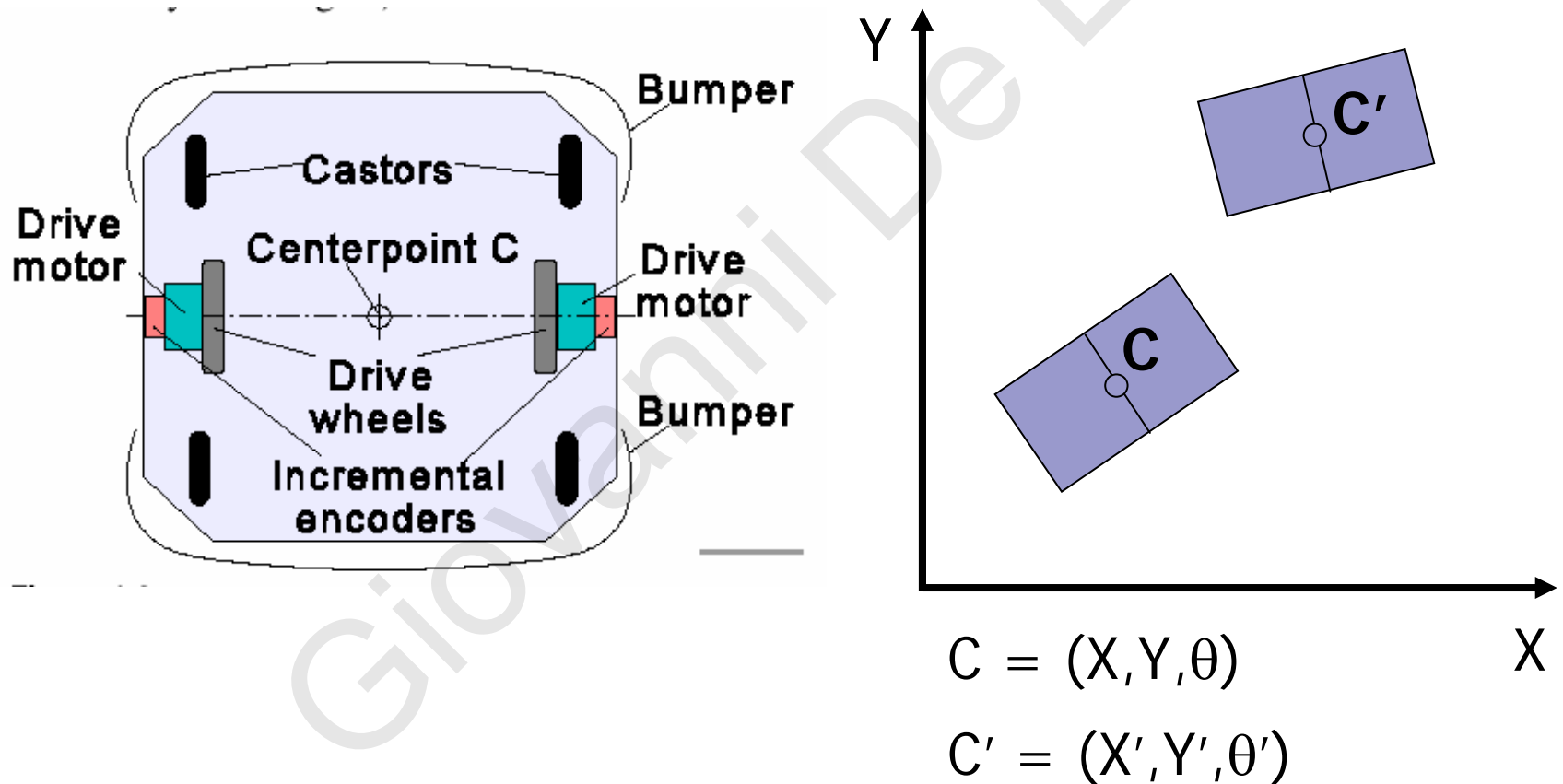


Odometria - Dead Reckoning

- Si basa sul calcolo dei giri effettuati dalle ruote del robot (misurata con gli encoder) durante gli spostamenti.
- Fornisce una buona accuratezza su piccoli spostamenti.
- L'errore commesso si accumula nel tempo con la distanza percorsa dal robot (scarsa precisione sulle lunghe distanze).
- L'informazione odometrica viene corretta utilizzando sistemi di localizzazione alternativi.

Odometria - Dead Reckoning

Esempio di calcolo dell'odometria (per piccoli spostamenti)



Odometria - Dead Reckoning

Supponiamo che dopo un certo intervallo (I) gli encoder della ruota sinistra e destra abbiano registrato rispettivamente un incremento del numero di impulsi pari a N_L e N_R .

Sia

$$C_m = \pi D/n C_e$$

dove

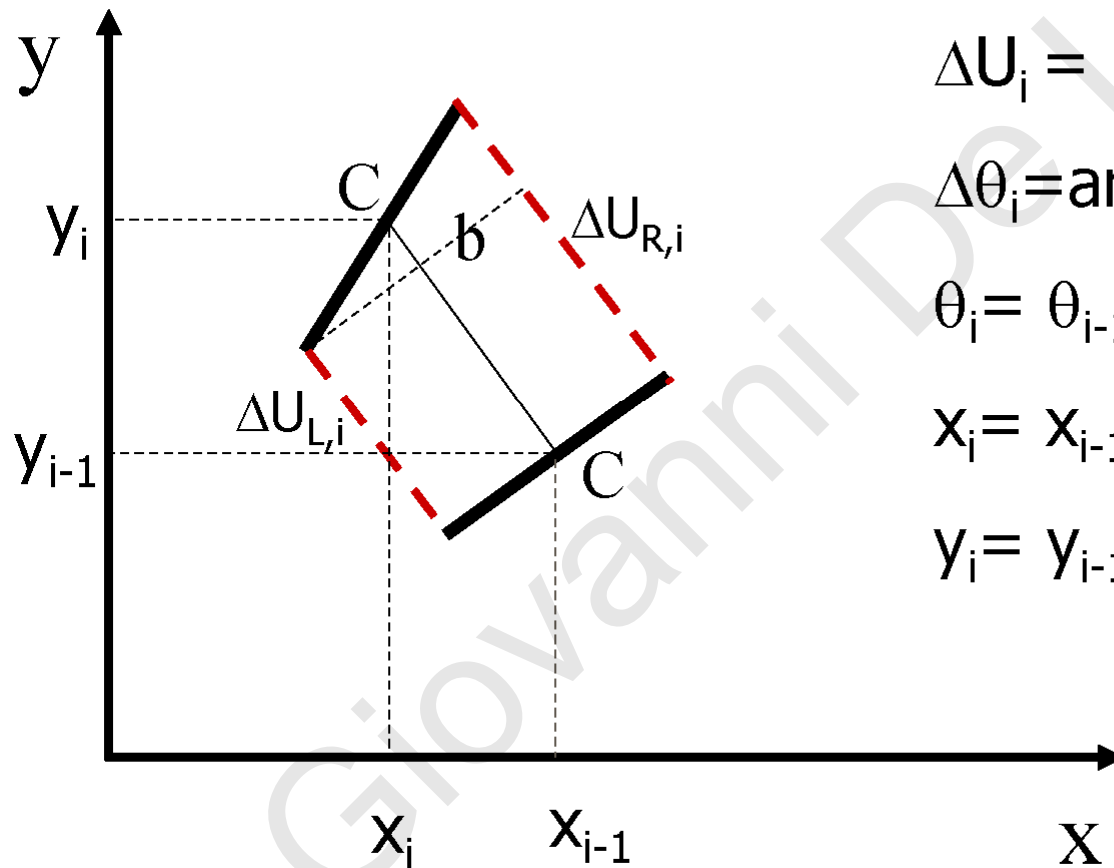
C_m = fattore di conversione che traduce gli impulsi lineari in distanza lineare effettuata delle ruote

D = diametro nominale delle ruote

C_e = risoluzione degli encoder

n = Rapporto di riduzione tra motore (dove è posizionato l'encoder) e ruota

Odometria - Dead Reckoning



$$\Delta U_{L/R,i} = C_m N_{L/R,i}$$

$$\Delta U_i = (\Delta U_{L,i} + \Delta U_{R,i})/2$$

$$\Delta \theta_i = \arctg(\Delta U_{R,i} - \Delta U_{L,i})/b$$

$$\theta_i = \theta_{i-1} + \Delta \theta_i$$

$$x_i = x_{i-1} + \Delta U_i \cos \theta_i$$

$$y_i = y_{i-1} + \Delta U_i \sin \theta_i$$

Odometria - Dead Reckoning

Possiamo calcolare la **distanza** percorsa dalle ruote sinistra e destra,

$\Delta U_{L,i}$
e $\Delta U_{R,i}$, come

$$\Delta U_{L/R,i} = C_m N_{L/R,i}$$

Allora, la distanza percorsa dal centro del robot, ΔU_i risulta essere:

$$\Delta U_i = (\Delta U_{L,i} + \Delta U_{R,i})/2$$

Mentre l'angolo di orientazione del robot risulta essere incrementato di

$$\Delta \theta_i = (\Delta U_{R,i} - \Delta U_{L,i})/b$$

Dove (**b**) è la distanza tra le due ruote della base (idealmente misurata come la distanza tra i punti di contatto delle le ruote con il pavimento)

Odometria - Dead Reckoning

La nuova posizione del robot risulta essere:

$$\theta_i = \theta_{i-1} + \Delta\theta_i$$

$$x_i = x_{i-1} + \Delta U_i \cos\theta_i$$

$$y_i = y_{i-1} + \Delta U_i \sin\theta_i$$

Dove $(x_{i-1}, y_{i-1}, \theta_{i-1})$ era la posizione nello spazio del centro del robot c.

Calcolo spostamento lineare

--- Calcolo spostamento lineare delle ruote Cm -----

$$Cm = (\text{Pi} * \text{diametro_ruote}) / (\text{riduzione} * \text{impulsi_encoder})$$

$$Cm = (3.1415926 * 100) / (43 * 400)$$

$$Cm = 314.15926 / 17200$$

$$Cm = 0,018265073 \quad \text{per ogni impulso encoder}$$

$$\text{Inverso} \quad 1 / 0,018265073 = 54.74$$

Calcoli odometrici

(Timer 10 mSec o meno)

Differenza(1) = (Val_Encoder(1) - Val_Encoder_Old(1))

Differenza(2) = (Val_Encoder(2) - Val_Encoder_Old(2))

Delta(1) = Differenza(1) / Spostamento_lineare_ruota

Delta(2) = Differenza(2) / Spostamento_lineare_ruota

Delta_U = (Delta(1) + Delta(2)) / 2

Delta_Theta = (Delta(2) - Delta(1)) / Interasse_ruote

Theta = Theta + Delta_Theta

Xpos = Xpos_old + Delta_U * Cos(Theta)

Ypos = Ypos_old + Delta_U * Sin(Theta)

Angolo = Theta * 180 / PI

Xpos_old = Xpos

Ypos_old = Ypos

Val_Encoder_Old(1) = Val_Encoder(1)

Val_Encoder_Old(2) = Val_Encoder(2)