

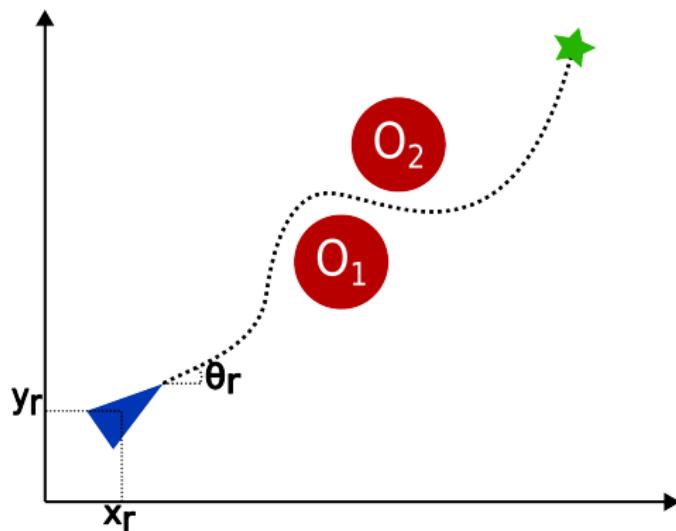
Path planning locale per robot mobili basato su switching tra potenziali artificiali

Flavio Maiorana

Definizione del Problema

Path planning locale

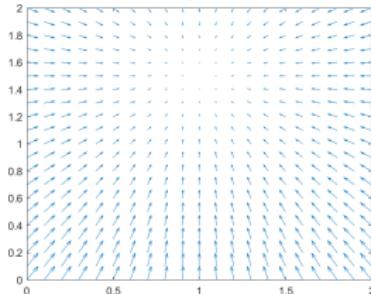
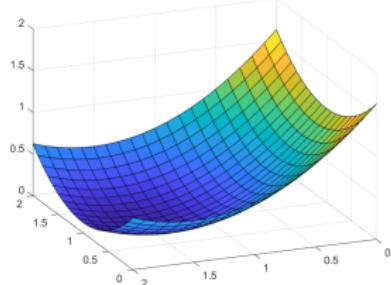
Guidare il robot mobile dalla sua **posizione iniziale** ad un **punto di goal**, evitando gli **ostacoli** non noti a priori



Potenziali artificiali

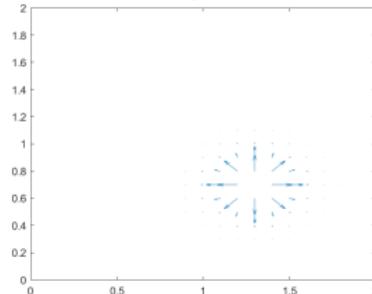
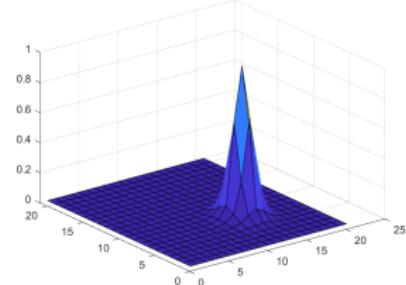
Metodo classico

Potenziale attrattivo U_a



$$-\nabla U_a$$

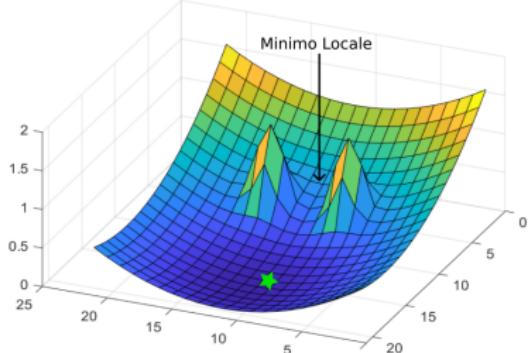
Potenziale repulsivo U_r



$$-\nabla U_r$$

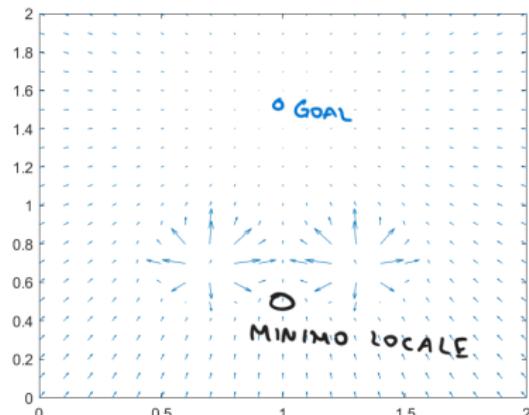
Potenziali artificiali

Metodo classico



Potenziale totale

$$U_{tot} = U_a + U_r$$



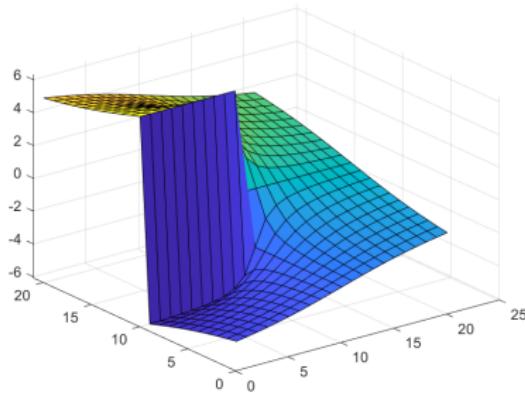
Antigradiente $-\nabla U_{tot}$
(Velocità di riferimento)

La somma di potenziali causa il rischio di minimo locale

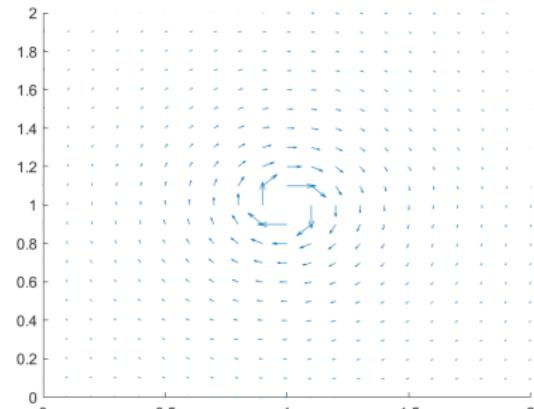
Potenziali artificiali

Potenziale bypassante

- Alternativa al potenziale repulsivo
- Antigradiente di forma circolare e non radiale
- Intensità aumenta con avvicinamento all'ostacolo



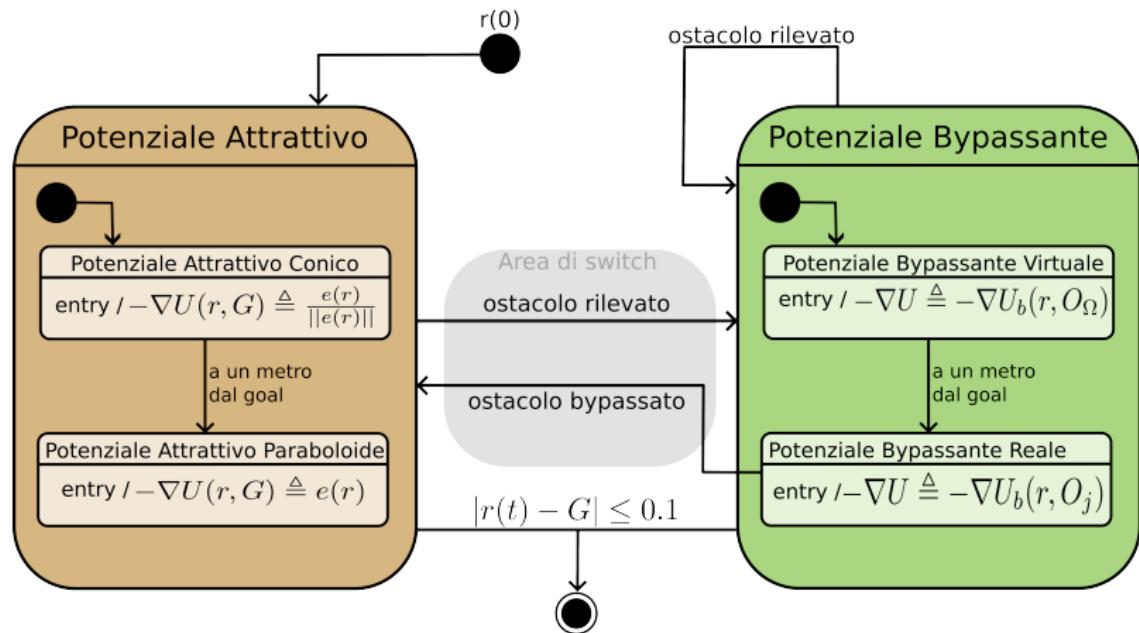
Potenziale totale U_b



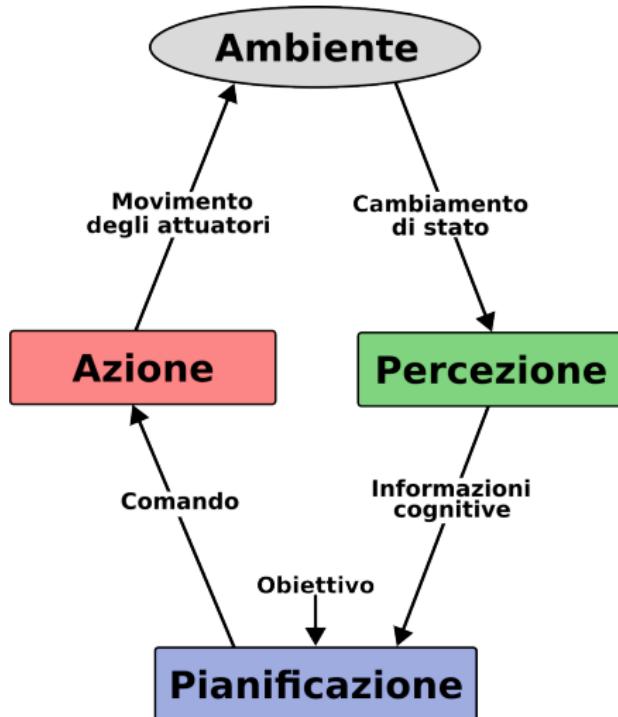
Antigradiente $-\nabla U_b$

Panoramica sull'algoritmo

Siano U_a e U_b il potenziale attrattivo e bypassante: in ogni t il robot è guidato da un solo potenziale tramite l'antigradiente $-\nabla U$



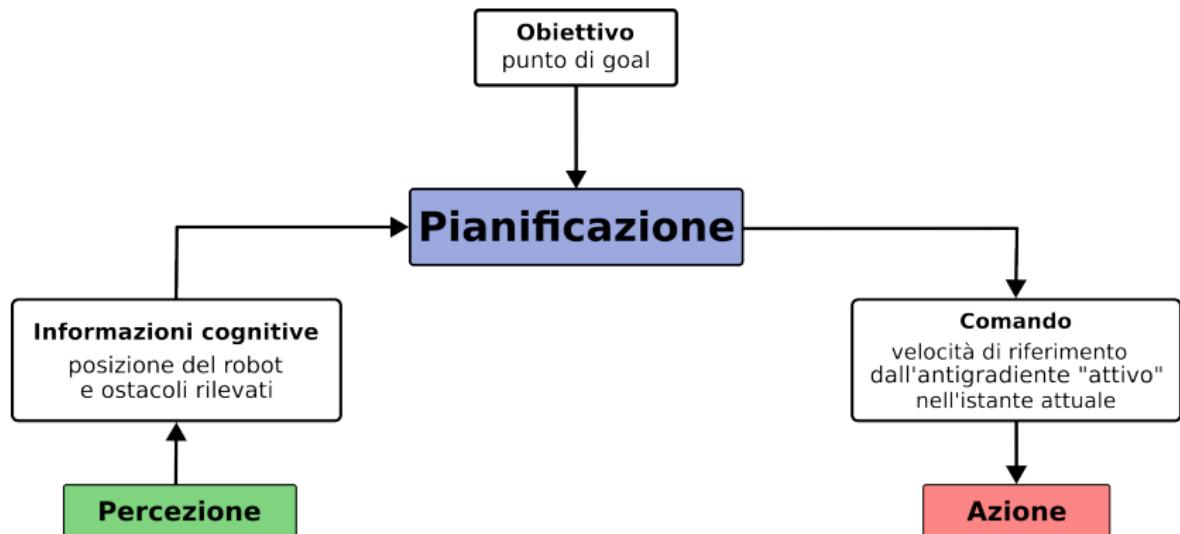
Architettura di navigazione



Entry point del modulo software

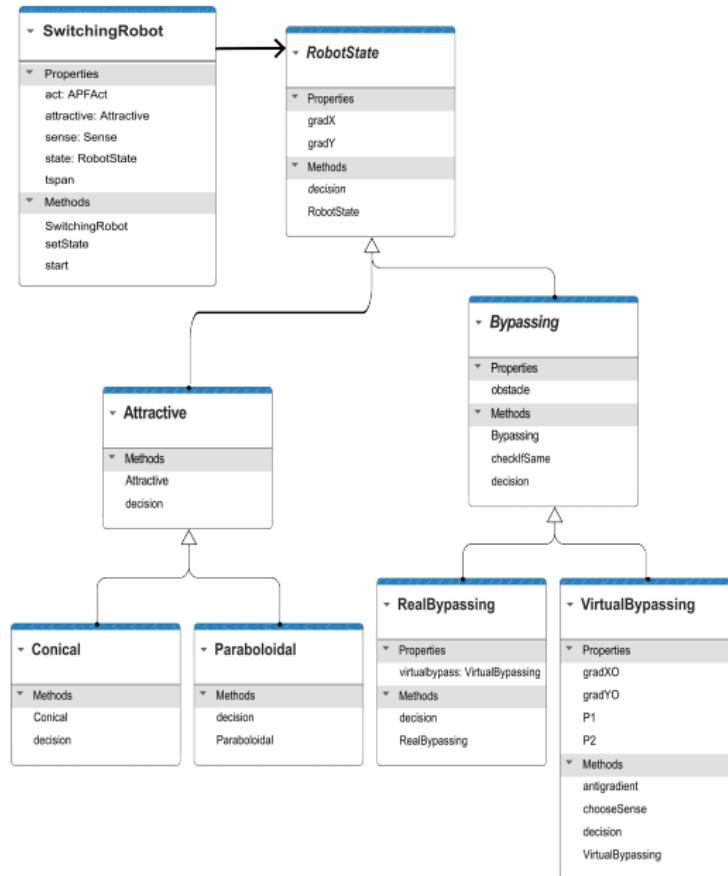
```
function start
    error = norm([xr ,yr] -G);
    while(error > 0.1)
        %Detected obstacle
        dObstacle = sense.scan();
        %New directive
        state.decision(dObstacle);
        %Commands to the actuators
        newPose = act.move(tspan);
        %Refreshing the error
        error = norm([xr ,yr] -G);
    end
end
```

Modulo di pianificazione



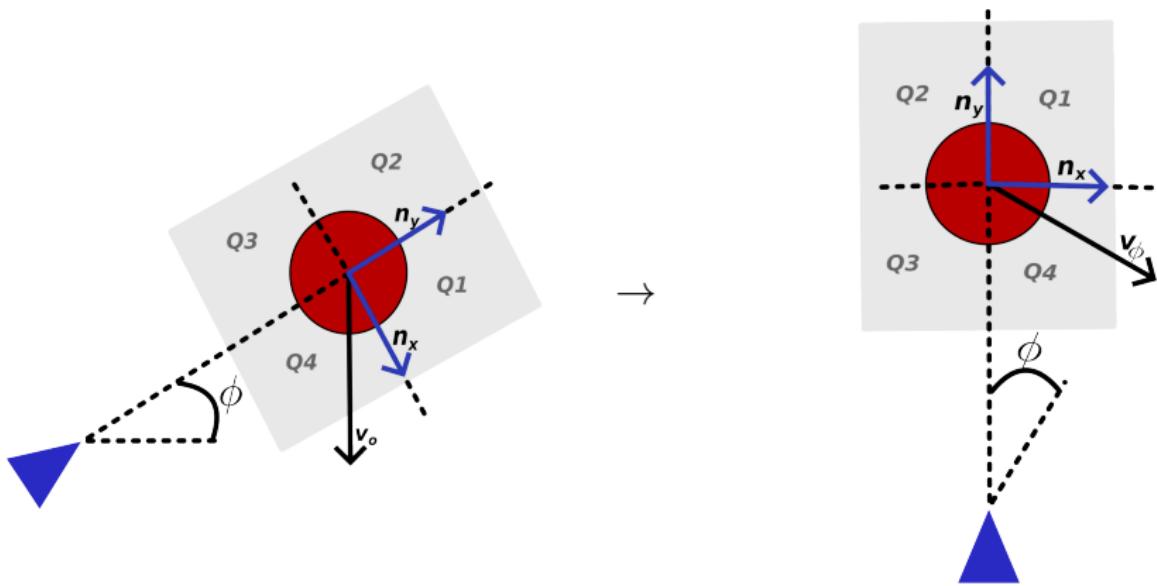
Modulo di pianificazione

Design pattern State



Modulo di pianificazione

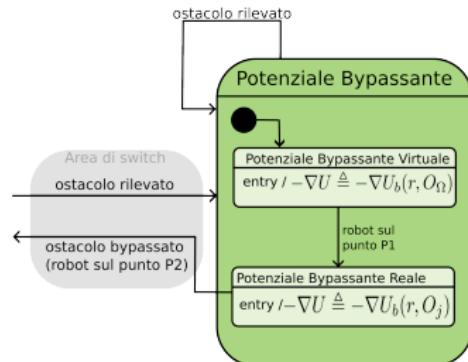
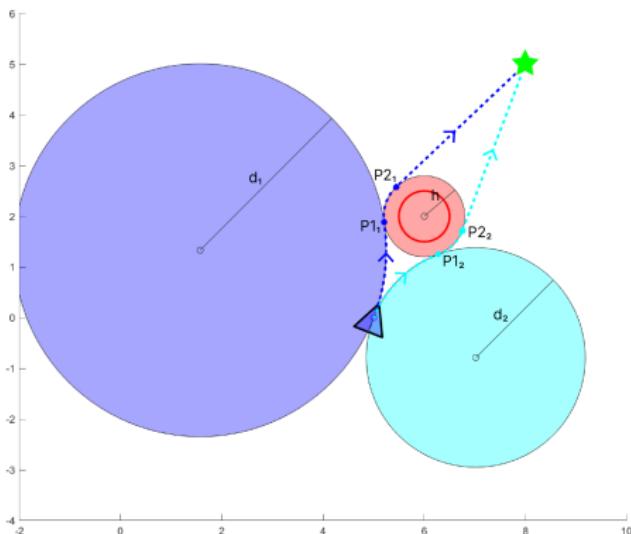
Switching : scelta del verso di bypassing



$$\begin{cases} \cos v_\phi \geq 0 \Rightarrow \text{orario} \\ \cos v_\phi < 0 \Rightarrow \text{antiorario} \end{cases}$$

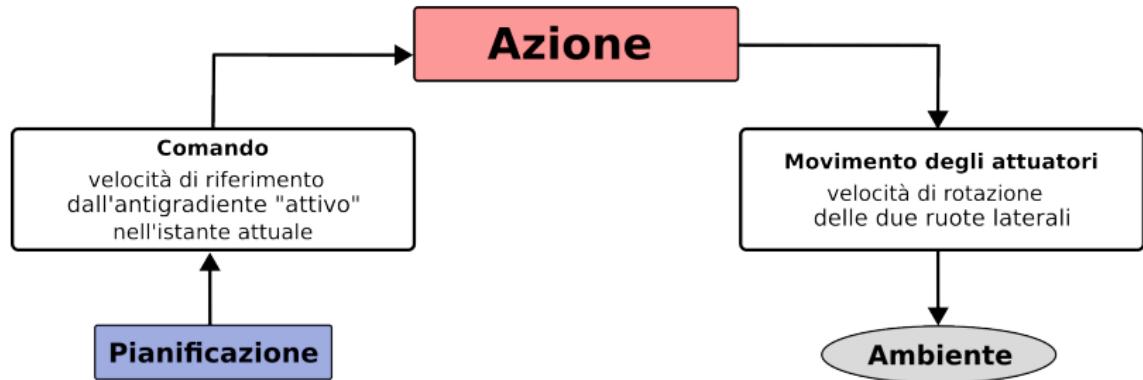
Modulo di pianificazione

Switching: assenza di discontinuità



- Ostacolo fermato in τ
- Ausilio di ostacolo virtuale
- Condizioni di tangenza sulle curve di livello del potenziale bypassante

Modulo di azione



Modulo di azione

Legge di controllo

- Obiettivo: ottenere dei comandi di velocità
- Nello specifico $v(t)$ velocità lineare e $\omega(t)$ velocità angolare
- Velocità di riferimento è $v_{\nabla}(t) \triangleq -\nabla U(r, G)$

Imponendo

$$v(t) = M_v \cos(\theta_{\nabla}(t) - \theta_r(t)) \quad (1)$$

$$\omega(t) = K_{\omega}(\theta_{\nabla}(t) - \theta_r(t)) \quad (2)$$

dove $M_v = \|v_{\nabla}(t)\|$, $\theta_{\nabla} = \angle v_{\nabla}(t)$ e

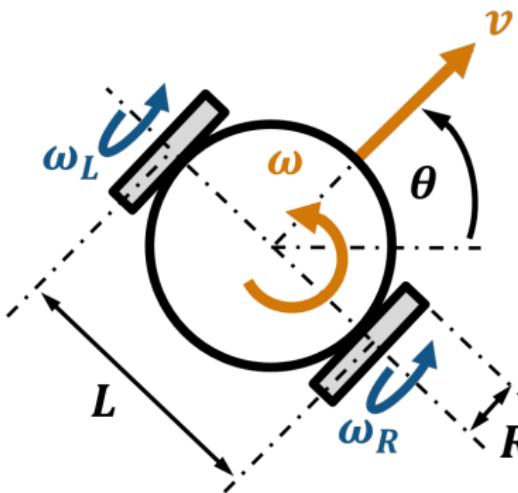
$$K_{\omega}(t) =$$

$$\begin{cases} \frac{\dot{\theta}_{\nabla}(t) + K_c |\theta_{\nabla}(t) - \theta_r(t)|^{\nu} \cdot sign(\theta_{\nabla}(t) - \theta_r(t))}{\theta_{\nabla}(t) - \theta_r(t)} & |\theta_{\nabla}(t) - \theta_r(t)| \geq \xi \\ 0 & \text{altrimenti} \end{cases}$$

Modulo di azione

Modello cinematico

$$\begin{cases} \dot{x}(t) = v(t) \cos(\theta(t)) \\ \dot{y}(t) = v(t) \sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \end{cases} \Rightarrow \begin{cases} \dot{x}(t) = \frac{R}{2} (\omega_R(t) + \omega_L(t)) \cos(\theta(t)) \\ \dot{y}(t) = \frac{R}{2} (\omega_R(t) + \omega_L(t)) \sin(\theta(t)) \\ \dot{\theta}(t) = \frac{R}{L} (\omega_R(t) - \omega_L(t)) \end{cases}$$



Dettaglio: velocità delle ruote

$$\begin{bmatrix} \omega_R(t) \\ \omega_L(t) \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix}^{-1} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$$

Modulo di percezione

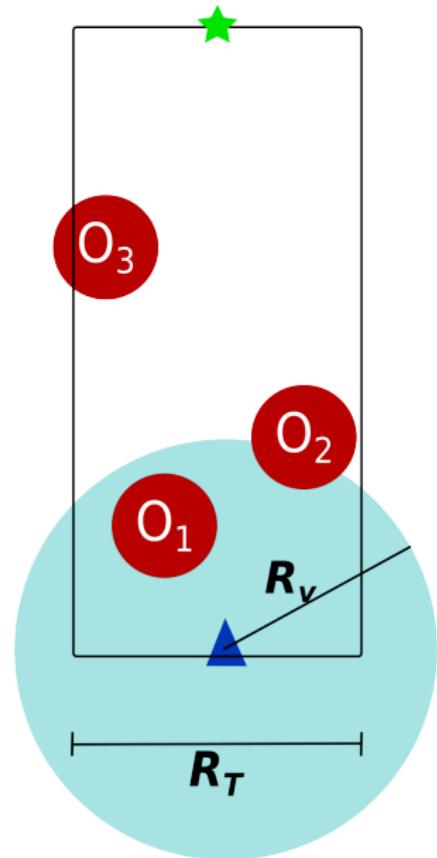


Modulo di percezione

- Raggio di visione R_v
- Tubo verso il goal di larghezza R_T
- Localizzazione sul centro dell'ostacolo



L'ostacolo da bypassare é quello, se c'è, che si trova nel tubo ad una distanza dal robot minore di tutti gli altri ostacoli nel tubo



Risultati

Tre ostacoli in movimento

Risultati

Cinque ostacoli in movimento

Risultati

Tredici ostacoli fissi

Risultati

Rischio minimo locale