

TITOLO DA INSERIRE

FRANCESCO ARGENTIERI^{*}

Università di Trento
francesco.argentieri@studenti.unitn.it

GIACOMO MAZZAGLIA[†]

Università di Trento
giacomo.mazzaglia@studenti.unitn.it

26 marzo 2018

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

^{*}ID: 183892

[†]ID: 123456

I. MODELLO DEL SISTEMA

i. Modello cinematico

Il robot è basato sul modello dell'uniciclo a trazione differenziale, la configurazione è completamente descritta da $\mathbf{q} = [x \ y \ \theta]^T$, dove (x, y) sono le coordinate cartesiane del punto di contatto con il suolo e θ è l'orientamento della ruota rispetto l'asse x . [1], come in figura 2. Il modello cinematico dell'uniciclo è descritto dall'equazioni (1):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (1)$$

Il robot ha le dimensioni riportate in tabella 1.

Tabella 1: Riepilogo dimensioni

dimensioni		
raggio ruote	[m]	0.07
interasse	[m]	0.30

Questo è equipaggiato con un sensore virtuale *lidar*, basato sul modello Hokuyo URG-04LX, collocato al centro della struttura in modo tale da evitare errori di offset, di seguito se ne riportano le caratteristiche, di cui adatte ad hoc per la simulazione. Come sensori propriocettivi presenta due encoder incrementali virtuali calettati sull'asse delle ruote, le caratteristiche di entrambi sono riportate in tabella 2. Una rappresentazione del robot è osservabile in figura 2.

Tabella 2: Specifiche sensori

specifiche lidar virtuale		
risoluzione angolare	[°]	0.36
angolo di scansione	[°]	180.00
massima distanza	[m]	4.00
minima distanza	[m]	0.02
risoluzione	[mm]	1.00
specifiche encoder virtuale		
risoluzione	$2 \cdot (\frac{\pi}{2600})$	

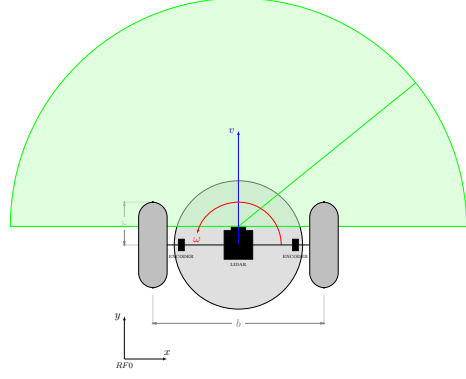


Figura 1: modello cinematico

ii. Pianificazione

La pianificazione del percorso per i robot è uno dei criteri importanti da prendere in considerazione per migliorare il livello di autonomia del robot. Nella pianificazione del percorso, la sicurezza è un problema importante che dovrebbe essere preso in considerazione al fine di garantire che un robot raggiunga la posizione target senza collisione con gli ostacoli circostanti. Inoltre, ci sono aspetti importanti che devono essere affrontati nella pianificazione del percorso; tempo computazionale, percorso ottimale e completezza. Uno dei metodi più diffusi per la pianificazione dei percorsi è il metodo *Campi Potenziali Artificiali*. Il metodo del potenziale è in grado di superare uno scenario sconosciuto, tenendo conto della realtà dell'ambiente corrente del movimento del robot. Due tipi di forze coinvolte nel potenziale metodo del campo; forza attrattiva generata da obiettivi e forza repulsiva generata da ostacoli, di conseguenza, il robot deve riprogrammare un nuovo percorso [2]. Utilizzando informazioni parziali sullo spazio di lavoro raccolte attraverso i sensori quindi le informazioni sensoriali integrate in una mappa secondo un paradigma *sense—plan—move*. Oppure utilizzare le informazioni sensoriali impiegate per pianificare moti secondo un paradigma *stimulus—response* (navigazione reattiva). Il robot è considerato come un punto sotto l'influenza dei campi prodotti da obiettivi e ostacoli nello spazio di ricerca. Le forze repulsive sono generate da ostacoli mentre la forza attrattiva è generata dagli obiettivi. Le forze risultanti (la somma di tutte

le forze) dei campi sul robot vengono utilizzate per determinare la direzione del movimento del robot e la velocità di spostamento evitando collisioni[3]. Tuttavia esistono svantaggi quali: **a)** situazione di stallo dovuta ai minimi locali; **b)** oscillazione in presenza di ostacoli; **c)** nessun passaggio tra ostacoli ravvicinati; **d)** oscillazioni in passaggi stretti[4]. Il robot viene considerato come punto $\mathbf{q} = (x \ y)^T$, in un piano cartesiano, attratto (potenziale U_{att}) dal punto obiettivo \mathbf{q}_g e respinto (potenziale U_{rep}) dagli ostacoli.

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \quad (2)$$

dove $U(q)$ potenziale artificiale; $U_{\text{att}}(q)$ campo attrattivo; $U_{\text{rep}}(q)$ campo repulsivo. La pianificazione avviene in modo incrementale: a ogni configurazione \mathbf{q} , la forza artificiale viene generata come (3)

$$\begin{aligned} F(q) &= -\nabla U(q) \\ &= -\nabla U_{\text{att}}(q) - U_{\text{rep}}(q) \\ F(q) &= F_{\text{att}}(q) + F_{\text{rep}}(q) \end{aligned} \quad (3)$$

dove $F(q)$: forza artificiale; $F_{\text{att}}(q)$: forza attrattiva; $F_{\text{rep}}(q)$: forza repulsiva. Il campo potenziale U_{att} tra robot e obiettivo viene descritto da (4) per trascinare il robot nell'area obiettivo.

$$\begin{aligned} U_{\text{att}}(q) &= \frac{1}{2} k_a (q - q_d)^2 \\ &= \frac{1}{2} k_a \rho_{\text{goal}}^2(q) \end{aligned} \quad (4)$$

dove k_a : coefficiente positivo per APF¹; q : posizione corrente del robot; q_d : posizione corrente dell'obiettivo. $\rho_{\text{goal}}(q) = \|q - q_d\|$ è una distanza euclidea dalla posizione del robot alla posizione dell'obiettivo. La forza attrattiva del robot è calcolata come gradiente negativo del potenziale campo[5]:

$$\begin{aligned} F_{\text{att}}(q) &= -\frac{1}{2} k_a \rho_{\text{goal}}^2(q) \\ F_{\text{att}}(q) &= -k_a (q - q_d) \end{aligned} \quad (5)$$

$F_{\text{att}}(q)$, nell'eq. (5), è un vettore diretto verso q_d con intensità linearmente proporzionale alla

distanza da q a q_d . Può essere scritto nelle sue componenti:

$$\begin{aligned} F_{\text{att}} - x(q) &= -k_a (x - x_d) \\ F_{\text{att}} - y(q) &= -k_a (y - y_d) \end{aligned} \quad (6)$$

Le equazioni (6) sono la forza attrattiva nelle direzioni x e y . Nella funzione potenziale, il robot deve essere respinto dagli ostacoli, ma se lontano da questi, il movimento non risente della loro influenza. La funzione potenziale di repulsione (7) è:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} k_b \left(\frac{1}{d(q)} - \frac{1}{d_0} \right)^2 & \text{se } d(q) \leq d_0 \\ 0 & \text{se } d(q) > d_0 \end{cases} \quad (7)$$

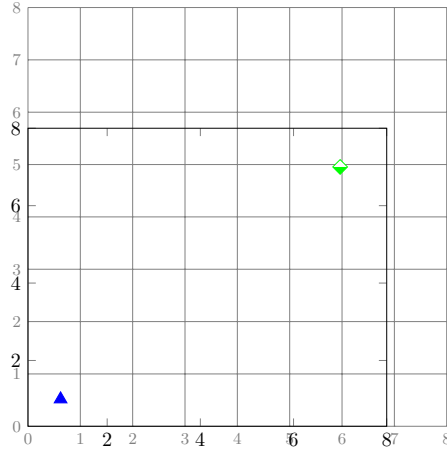


Figura 2: Modello cinematico

¹Artificial Potential Field

II. IMPLEMENTAZIONE

Per la realizzazione del progetto si è optato per un simulazione in ambiente MATLAB, si è così realizzato un ambiente virtuale 2D atto a ricostruire la pianta di un edificio come, ad esempio, uffici, capannoni, aule. Allo scopo è stata usata una generazione procedurale. Il vantaggio più evidente dei livelli generati proceduralmente è la loro varietà che portano ad ogni esecuzione, l'ambiente cambia. Ciò significa che i robot non possono apprendere le posizioni di oggetti e questo permette di testare l'affidabilità in casi sempre differenti. Un altro vantaggio comune a tutte le implementazioni della generazione procedurale è il tempo che si risparmia nello sviluppo. Nel nostro progetto, avremo un numero infinito di livelli unici. Se stessimo creando i nostri livelli manualmente, ciò sarebbe semplicemente impossibile. Saremmo limitati a forse una decina di livelli al massimo. L'utilizzo di una generazione procedurale come questa elimina questo carico di lavoro dagli sviluppatori, risparmiando tempo e aumenta la portata di ciò che è possibile. Tra gli svantaggi, si ricorda, che la generazione procedurale di per sé non è in alcun modo casuale. L'assenza di controllo è una mancanza comune della generazione procedurale in generale, ma forse non è più prevalente rispetto alla generazione di ambienti. Dato che, di solito gli ambienti sono realizzati a mano da designer. Lasciare questo lavoro a un algoritmo si traduce in una significativa perdita di controllo. Un'altra considerazione che deve essere presa in considerazione è la potenza di calcolo che è richiesta, nel nostro caso, si ha solo una matrice 2D di piccole dimensioni che deve essere generata. Tuttavia, mappe di più larga scala, avranno un costo computazionale diventa più significativo e deve essere preso in considerazione.[6]

i. Binary Space Partitioning

Il partizionamento dello spazio binario è un processo generico di divisione ricorsiva di una scena in due finché il partizionamento soddisfa uno o più requisiti. Può essere visto come una generalizzazione di altre strutture ad

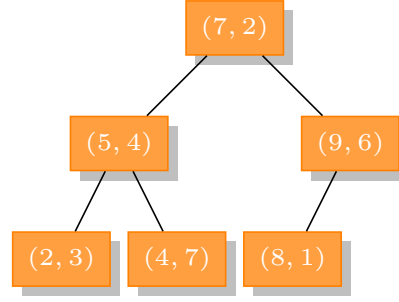


Figura 3: risultato dell'albero $k-d$.

albero spaziale, uno in cui gli iperpiani che suddividono lo spazio possono avere qualsiasi orientamento, piuttosto che essere allineati con gli assi delle coordinate.[7] L'albero $k-d$ è un albero binario in cui ogni nodo è un punto k -dimensionale. Ogni nodo non foglia può essere pensato come generare implicitamente un iperpiano scisso che divide lo spazio in due parti, note come semispazi. I punti a sinistra di questo iperpiano sono rappresentati dal sotto-albero sinistro di quel nodo e i punti a destra dell'iperpiano sono rappresentati dal sotto-albero destro. La direzione dell'iperpiano viene scelta nel modo seguente: ogni nodo dell'albero è associato a una delle dimensioni k , con l'iperpiano perpendicolare all'asse di quella dimensione.[8]

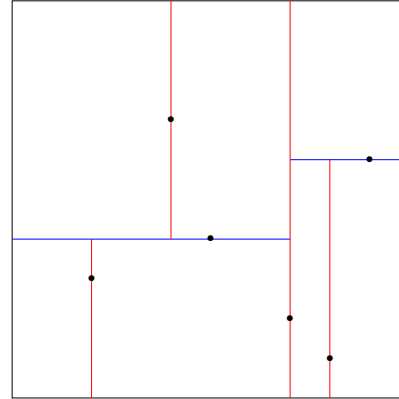


Figura 4: decomposizione per il set di punti.

ii. Esecuzione

Una volta generato lo scenario la simulazione inizia con il posizionamento di uno o più robot all'interno della mappa come condizione iniziale e assegnato un obiettivo da raggiun-

gere. Questi non conoscono l'ambiente in cui si trovano e quindi tentano di raggiungere l'obiettivo nella maniera più rapida impostando un' traiettoria rettilinea determinata come norma tra due punti. La pianificazione della traiettoria viene modificata in base alle rilevazioni effettuate dal lidar in tal modo essi evitano l'ostacolo perché repulsi.

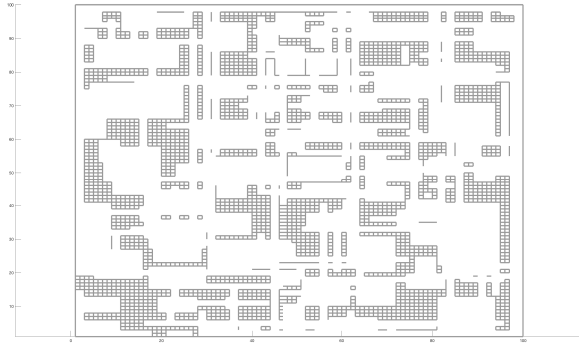


Figura 5: esempio di scenario procedurale 100×100

- [1] B. Siciliano, *Robotica. Modellistica, pianificazione e controllo*, ser. Collana di istruzione scientifica. Serie di automatica. McGraw-Hill Companies, 2008. [Online]. Available: <https://books.google.it/books?id=HdaVPQAACAAJ>
- [2] C. K. M. C. K. A. N. H. Sabudin E. N, Omar. R, “Potential field methods and their inherent approaches for path planning,” *ARPJ Journal of Engineering and Applied Sciences*, vol. 11, no. 18, SEPTEMBER 2016.
- [3] X. Xu, C. Li, and Y. Zhao, “Air traffic rerouting planning based on the improved artificial potential field model,” in *2010 Chinese Control and Decision Conference*, May 2010, pp. 1444–1449.
- [4] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 1398–1404 vol.2.
- [5] G. Li, A. Yamashita, H. Asama, and Y. Tamura, “An efficient improved artificial potential field based regression search method for robot path planning,” in *2012 IEEE International Conference on Mechatronics and Automation*, Aug 2012, pp. 1227–1232.
- [6] D. Green, *Procedural Content Generation for C++ Game Development*. Packt Publishing, 2016. [Online]. Available: <https://books.google.it/books?id=EQYcDAAAQBAJ>
- [7] W. contributors, “Binary space partitioning — wikipedia, the free encyclopedia,” 2018, [Online; accessed 14-March-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Binary_space_partitioning&oldid=828843135
- [8] —, “K-d tree — wikipedia, the free encyclopedia,” 2018, [Online; accessed 14-March-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-d_tree&oldid=829846868