



Progetto di Robotica Mobile

Giorgio Ubbriaco 209899

Sommario

[Introduzione 3](#_Toc114162587)

[Robot Model 3](#_Toc114162588)

[Environment 3](#_Toc114162589)

[Path Planning 4](#_Toc114162590)

[Obstacles Shape 5](#_Toc114162591)

[Artificial Potential Fields 5](#_Toc114162592)

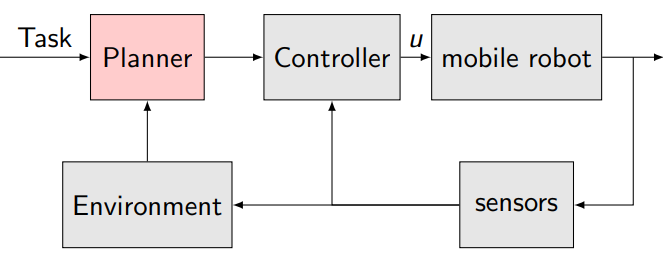
[Discrete Potential Fields 6](#_Toc114162593)

[Voronoi Diagrams 6](#_Toc114162594)

[Visibility Graphs 7](#_Toc114162595)

# Introduzione

L’obiettivo di questo progetto di robotica mobile è quello di progettare le principali tecniche di *path planning* e di controllo tenendo conto di un robot mobile di tipologia *uniciclo* e di ambiente composto di 6 ostacoli di varie dimensioni. Il seguente schema di controllo riassume i task principali per la generica progettazione di un robot mobile:

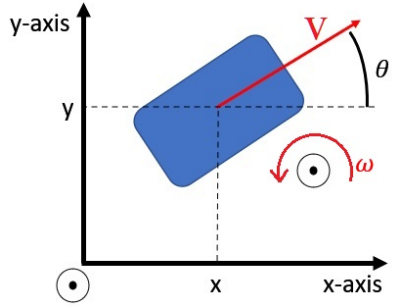


Schema di Controllo del Robot Mobile

# Robot Model

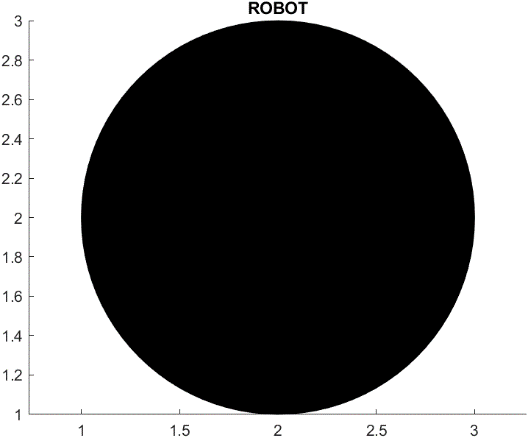
Un robot mobile di tipologia *uniciclo* è un veicolo avente una sola ruota orientabile. La configurazione di tale robot è descritta da , dove sono le coordinate cartesiane del punto di contatto della ruota con il suolo e è l’orientamento della ruota e, pertanto, del veicolo in questione rispetto all’asse x. Il vincolo del puro rotolamento della ruota è descritto dalla seguente relazione:

e indica che la velocità del punto di contatto è nulla nella direzione normale all’asse sagittale del veicolo. Pertanto, il modello del robot mobile di tipologia *uniciclo* risulta essere il seguente:

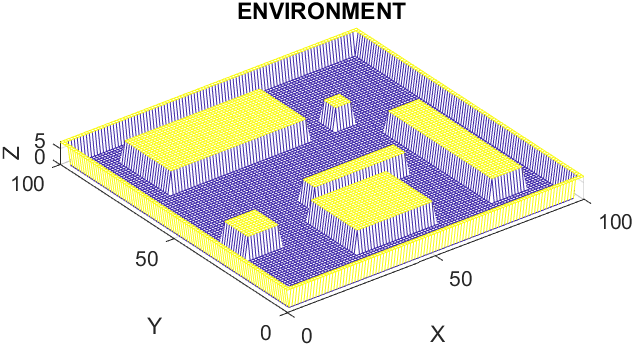


Unicycle Robot Model

dove è la velocità di “guida” (*driving velocity*) mentre è la velocità di sterzata (*steering velocity*).

In questo progetto è stato considerato un robot mobile di tipologia *uniciclo* di raggio pari a 1 e di forma circolare ai fini dell’implementazione. Infatti, la condizione ideale e teorica era quella di considerarlo di raggio pari a 0.5, cioè puntiforme.

# Environment

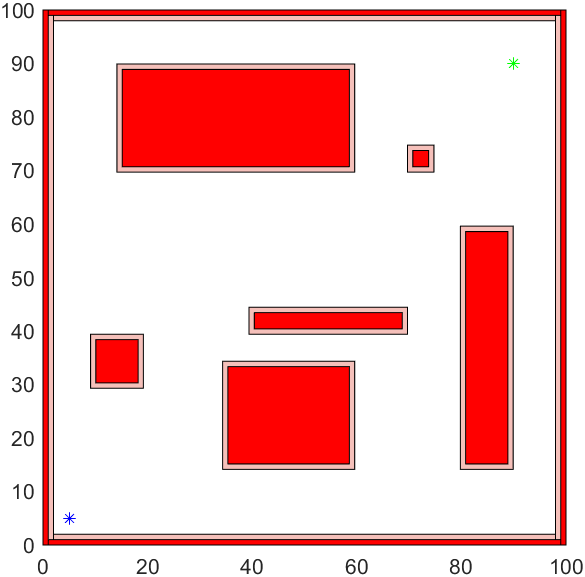
L’environment considerato prevede una larghezza ed una lunghezza rispettivamente pari a 100. Pertanto, la griglia binaria corrispondente in cui sono stati aggiunti gli ostacoli prevede un numero di righe di righe e di colonne pari al numero sopra indicato. I muri che delimitano l’ambiente sono stati considerati come ostacoli di forma mista (sia rettangolare che quadrata). Nello specifico tale griglia presenta valori pari a 0 o 1. Il valore 0 corrisponde ad una posizione (cella della matrice) libera mentre il valore 1 corrisponde ad una posizione occupata dagli ostacoli.

3D Environment Representation

# Path Planning

La simulazione del path planning viene effettuata tramite lo script “*path\_planning\_exec(environment).m*”. Sono state implementate ben 4 tecniche di path planning. Ogni tecnica restituisce in output la traiettoria generata dagli algoritmi corrispondenti. Inoltre, è stata considerata a priori un “ingrandimento” degli ostacoli che viene descritto nel seguente sotto-paragrafo.

## Obstacles Shape



Environment with Obstacle Management

Prevede di circoscrivere la forma dell'ostacolo con una forma nota al fine di evitare eventuali impatti del robot contro l’ostacolo mentre che segue le traiettorie progettate. In questo caso specifico essendo ogni ostacolo una forma nota (rettangolo o quadrato), lo si ingrandisce di 1 unità, cioè pari al raggio del robot mobile in questione. Nello specifico la parte degli ostacoli colorata di rosso indica l’ostacolo vero e proprio mentre quella colorata di rosso chiaro indica l’enlargement considerato. Inoltre, sono stati considerati i punti di *start* e *goal* aventi rispettivamente coordinate:

Pertanto, il robot mobile in questione partirà dalla posizione di *goal* e in base alla traiettoria progettata e generata dovrà giungere nella posizione di *goal* specificata.

## Artificial Potential Fields

Questa tecnica consiste nel far muovere il robot mobile nello spazio delle configurazioni sotto l’azione di un campo potenziale ottenuto come sovrapposizione di un potenziale attrattivo verso la destinazione e di un campo repulsivo dalla regione dei -ostacoli. La pianificazione avviene per iterazioni: ad ogni configurazione del robot, la forza artificiale generata dal potenziale viene generata come l’antigradiente del potenziale, e indica la direzione del moto localmente più promettente.

Il potenziale repulsivo viene considerato affinché il robot, sotto l’azione della forza attrattiva, non entri in collisione con gli ostacoli. L’idea di base considerata è stata quella di considerare ogni ostacolo come una componente convessa tale da definire un corrispondente potenziale repulsivo come:

dove è la configurazione del robot, , , è la distanza dalla componente convessa ed rappresenta il raggio di influenza di . Il potenziale si annulla al di fuori del raggio di influenza mentre tende ad infinito quanto più grande è . Per quanto riguarda il calcolo di , esso è stato appena definito come . Esso, infatti, viene ottenuto tramite la funzione nativa *bwdist* presente in MATLAB. Tale function calcola la distanza euclidea tale che risulta essere la distanza tra la generica cella della matrice e la cella diversa da zero più vicina. In questa maniera si riesce ad ottenere le distanze i-esime tra una generica configurazione del robot e la prossima che dovrà assumere. Tale calcolo è stato assegnato all’interno dello script con la variabile *Rho*. Inoltre, la variabile è identificata nello script dalla variabile *Eta*. In questo caso è stato scelto un e un che risultano essere le scelte tipiche in fase di implementazione del potenziale repulsivo. Tali assunzioni implementative provengono dalla FIRAS Function che Oussama Khatib propose nel 1980 con la pubblicazione dell’articolo “*Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*” per il *The International Journal of Robotics Research* che riporto qui di seguito:

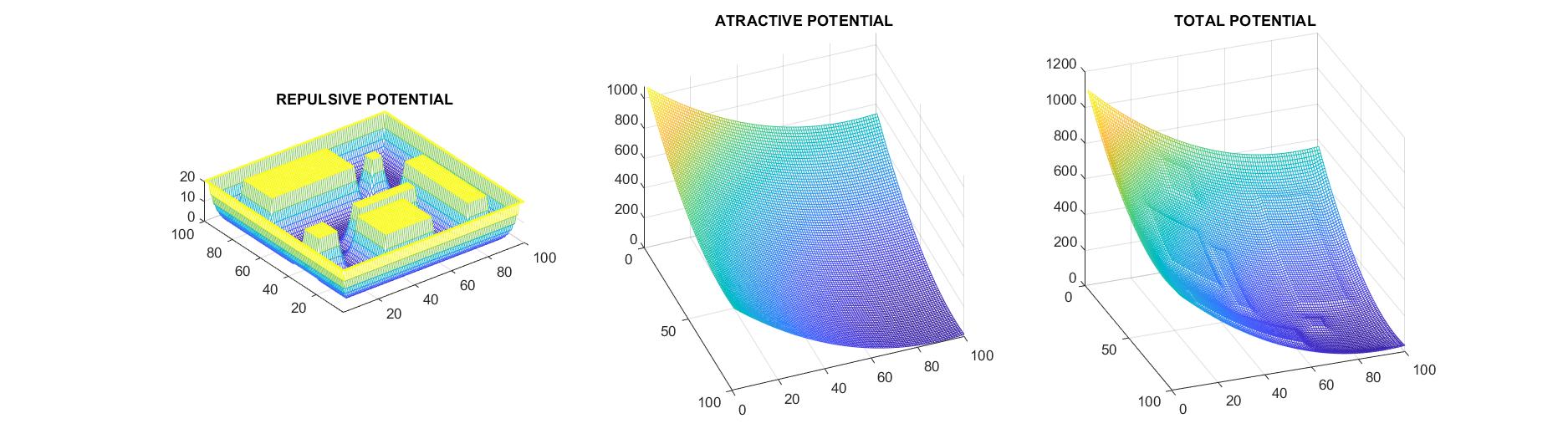
Infatti, come lo stesso Oussama Khatib afferma, rappresenta la minima distanza dall’ostacolo .

Il potenziale attrattivo, invece, è stato definito nella seguente maniera:

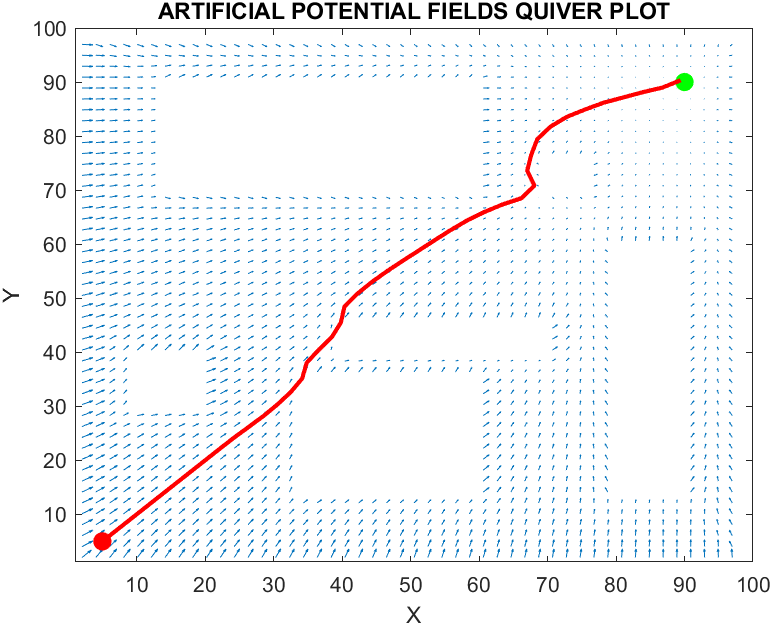
dove è la generica posizione, è la posizione target “g” (tratto sempre dalla pubblicazione di Oussama Khatib). In questo caso il coefficiente è stato considerato pari a ed assegnato alla variabile *xi*. Tale coefficiente viene utilizzato per regolare la forza del potenziale attrattivo. Per valori troppo bassi, cioè per un denominatore che tende sempre di più a valori più grandi comporta una forza attrattiva bassa tale che il robot mobile incontrerà maggiore difficoltà a raggiungere la posizione target. Invece, per un coefficiente sempre maggiore e, quindi, per un denominatore sempre minore, corrisponde una forza attrattiva sempre più grande tale che il robot mobile incontrerà minore difficoltà a raggiungere la posizione obiettivo. Tanto è vero che per valori troppo grandi di forza attrattiva, il robot potrebbe essere così tanto “attratto” che nel caso peggiore potrebbe passare attraverso gli stessi ostacoli.

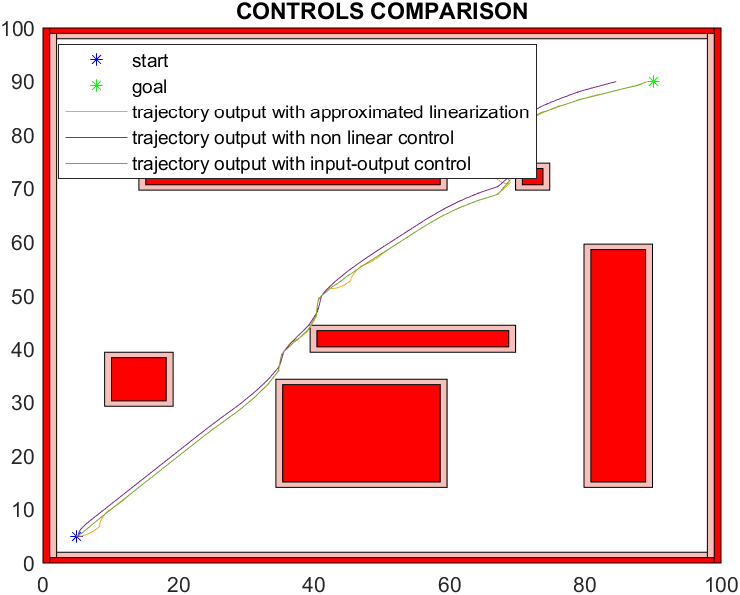
Il potenziale totale, pertanto, sarà pari alla soma del potenziale repulsivo e di quello attrattivo:

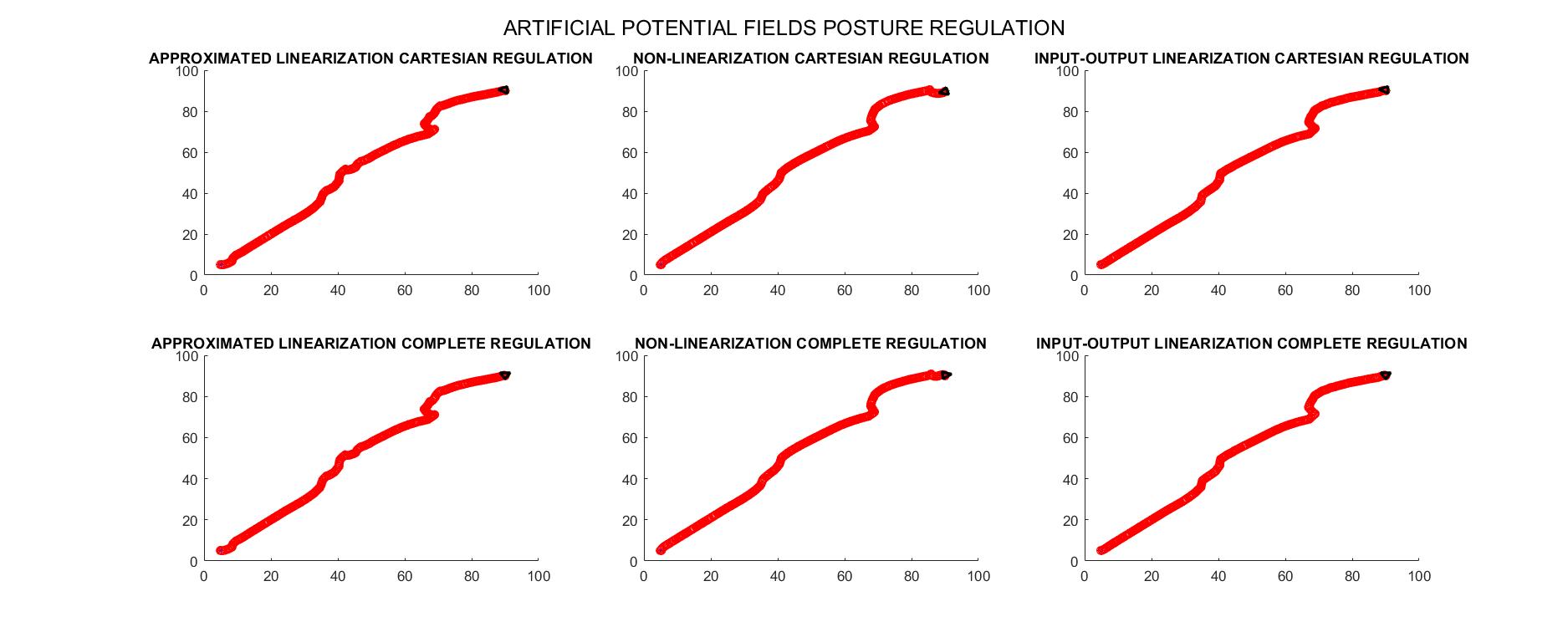
Viene riportato qui di seguito il plot grafico del potenziale repulsivo, attrattivo e totale:



Si può notare come il potenziale repulsivo rappresenti alla perfezione l’ambiente considerato. Questo perché geometricamente più il robot si avvicina all’ostacolo generico più il potenziale repulsivo aumenta. Infatti, la colorazione giallo-blu sta ad indicare che nelle regioni “libere”, cioè dove il robot ha libero passaggio, presenta un colore blu scuro mentre nelle regioni occupate dagli ostacoli è presente un colore che gradualmente sfuma verso il giallo intenso. Ovviamente anche i muri, essendo considerati come ostacoli, presenteranno le stesse caratteristiche appena descritte. Per quanto riguarda, invece, il potenziale attrattivo, essendo caratterizzato da una forza attrattiva relativamente media, il paraboloide risulta essere relativamente marcato. Tanto è vero che il potenziale totale si può notare come esso presenti praticamente soltanto il potenziale attrattivo con piccoli cenni di potenziale repulsivo. Questo è dovuto al fatto che la forza repulsiva scelta risulta essere parecchio superiore rispetto alla forza attrattiva ed essendo indirettamente proporzionali l’uno cerca di contrastare l’altro.

Dopo che si sono definite approfonditamente le premesse e i dettagli implementativi relativi ai potenziali utilizzati, è possibile generare la traiettoria utilizzando un algoritmo che calcola, sulla base del gradiente del potenziale totale considerato, per iterazioni la direzione e, quindi, la prossima posizione del robot mobile, all’interno dell’environment considerato. Il plot seguente illustra come l’uniciclo, partendo dalla posizione di start, riesce a raggiungere la posizione di goal evitando gli ostacoli presenti all’interno dell’ambiente.





## Discrete Potential Fields

## Voronoi Diagrams

## Visibility Graphs