



**UNIVERSITÀ DEGLI STUDI DI ROMA  
TOR VERGATA**

**FACOLTÀ DI INGEGNERIA**

**CORSO DI LAUREA IN INGEGNERIA  
DELL'AUTOMAZIONE**

**A.A. 2009/2010**

**Tesi di Laurea**

Risoluzione del problema dello SLAM  
mediante filtro di Kalman esteso

**RELATORE**

Prof. Francesco Martinelli

**CANDIDATO**

Benedetta Marchionne

*Ai miei Nonni,  
per essere ogni giorno  
dei maestri di Vita*

# Indice

<b>Ringraziamenti</b>	<b>1</b>
<b>Introduzione</b>	<b>2</b>
<b>1 Concetti Preliminari</b>	<b>5</b>
1.1 Il filtro di Kalman . . . . .	5
1.2 Il filtro di Kalman esteso (EKF) . . . . .	9
1.3 Definizione della cinematica del robot . . . . .	13
<b>2 Localizzazione mediante EKF</b>	<b>17</b>
2.1 Predizione . . . . .	17
2.2 Correzione . . . . .	19
2.3 Risultati simulativi . . . . .	22
<b>3 Introduzione al problema dello SLAM</b>	<b>24</b>
3.1 Lo SLAM posterior . . . . .	26
3.2 Notazioni dello SLAM . . . . .	27
<b>4 EKF SLAM con corrispondenze note</b>	<b>29</b>
4.1 Risultati simulativi . . . . .	34

<b>5</b>	<b>EKF SLAM con corrispondenze non note</b>	<b>39</b>
5.1	Risultati simulativi . . . . .	44
<b>6</b>	<b>Conclusioni e sviluppi futuri</b>	<b>47</b>
	Elenco delle figure	49
	Bibliografia	49

# Ringraziamenti

Vorrei cogliere questa opportunità per ringraziare tutte le persone che mi sono state accanto in questi anni. Fortunatamente sono molte e non riuscirò a citarle tutte. In primo luogo vorrei ringraziare i miei genitori, mia sorella e il mio piccolo fratellino per essermi stati vicini ed aver creduto in me anche quando non lo facevo io e per avermi sempre incoraggiato ad andare avanti. Sarebbe impossibile non ringraziare una persona per me davvero speciale, Francesco, per avermi teso la mano quando ne avevo bisogno ed aver gioito con me dei miei successi.

Le amiche di sempre, Michela ed Elisa, per essere state una presenza costante nella mia vita ed esserci oggi anche se arriveranno certamente in ritardo. In egual modo ringrazio anche Lucia e Alessio per essere sempre stati in grado di farmi sorridere.

Colgo l'occasione per ringraziare una persona che è da poco entrata nella mia vita, Antonio, per essere stato partecipe di questo mio piccolo traguardo nonostante la lontananza. Vorrei, infine, ringraziare il professor Francesco Martinelli per avermi dato la possibilità di avvicinarmi ad un argomento così affascinante e per i suoi preziosi consigli.

# Introduzione

Tra i maggiori problemi affrontati nella robotica mobile c'è quello di stimare la posizione del veicolo sfruttando le informazioni che si hanno sull'ambiente in cui evolve la sua posa, oppure quello di creare una mappa dell'ambiente a partire dalla conoscenza a priori del percorso effettuato dal veicolo. Tali problemi, noti nella letteratura come *localizzazione* e *mapping*, se sviluppati separatamente impediscono di rendere il veicolo completamente autonomo.

Il problema dello SLAM, acronimo per Simultaneous Localization And Mapping, è uno degli argomenti più studiati nel campo della robotica. Lo SLAM è essenzialmente la capacità di un veicolo di muoversi in un ambiente non noto, localizzando la propria posizione e contemporaneamente creando una mappa dell'ambiente circostante. Esistono molteplici approcci per la soluzione di questo problema. Storicamente, il primo è quello basato sull'utilizzo del filtro di Kalman e la sua estensione a sistemi non lineari noto in letteratura come EKF SLAM.

In questa tesi vengono proposte varie soluzioni al problema dello SLAM mediante filtro di Kalman esteso.

L'ambiente in cui si muove il veicolo è stato completamente virtualizzato con l'ausilio di Matlab, piattaforma scientifica largamente affermata nei campi ingegneristici per la sua flessibilità e potenza.

Si specifica, quindi, il percorso approssimativo che si vuole far seguire al robot, defi-

nendo in modo completamente arbitrario la posizione dei punti notevoli attraverso i quali il robot riesce a creare una mappa dell'ambiente.

In questo ambiente virtuale, modellando opportunamente la dinamica di moto del robot, così come l'odometria e le acquisizioni sensoriali, si sono potuti implementare diversi algoritmi basati su EKF SLAM.

Nello sviluppo del primo algoritmo, noto in letteratura con EKF SLAM CON CORRISPONDENZE NOT , si è fatta l'implicita assunzione che il robot conosca a priori la dimensione della mappa. In un primo momento si è anche supposto che i sensori di cui si suppone esso sia dotato, siano in grado di rilevare ad ogni passo tutti i landmark presenti nell'ambiente. In seguito sono state sviluppate anche due varianti di questo algoritmo ottenute associando un diverso modello per le acquisizioni sensoriali. In particolare, in una prima variante si suppone che il robot possa rilevare solo landmark che non distante più di una certa soglia, mentre nella seconda variante si è operata una limitazione anche sull'angolo di visione dei sensori (angolo di *bearing*).

Nel secondo algoritmo è stata implementata la soluzione al problema di EKF SLAM nel caso di CORRISPONDENZE NON NOTE. Tale algoritmo esegue l'aggiornamento della dimensione della mappa supponendo di vedere ad ogni passo un nuovo landmark. Per verificare se il landmark, di cui si è supposta l'esistenza, è effettivamente un nuovo landmark si applica in modo incrementale il criterio di massima verosomiglianza.

Nel primo capitolo sono introdotte le informazioni di base utili alla comprensione degli algoritmi che saranno presentati nei successivi capitoli. Nel secondo capitolo viene affrontato il problema di localizzazione fornendo un'algoritmo che risolve tale problema basato sull'utilizzo del filtro di Kalman esteso. Nel terzo capitolo viene effettuata un'introduzione al problema dello SLAM mettendone in evidenza le principali peculia-

rità.

Nel quarto capitolo viene analizzato il problema di SLAM sotto l'ipotesi di corrispondenze note. In particolare, vengono implementati diversi algoritmi, ognuno dei quali assume un modello differente per ciò che concerne le acquisizioni sensoristiche.

Nel quinto capitolo viene risolto il problema dello SLAM rimuovendo l'ipotesi di corrispondenze note .

Nell'ultimo capitolo vengono mostrate le conclusioni del lavoro di tesi e vengono dati degli spunti per gli sviluppi futuri di tale elaborato.

# Capitolo 1

## Concetti Preliminari

### 1.1 Il filtro di Kalman

Il filtro di Kalman è un algoritmo utilizzato per il filtraggio dei dati costruito su una media ragionata tra il prossimo valore predetto e il prossimo valore stimato.

Per la costruzione del suddetto filtro risulta necessario conoscere le seguenti componenti:

- una serie di misure effettuate sul sistema da stimare;
- la conoscenza di un modello matematico lineare descrittivo del sistema;
- il modello statistico dei rumori sulle misure.

Si consideri di voler effettuare una stima dello stato  $x \in \mathfrak{R}^n$  di un processo a tempo discreto governato dall'equazione alle differenze lineare stocastica

$$x_{k+1} = A_k x_k + B_k u_k + w_k \quad (1.1.1)$$

Nella (1.1.1) la matrice  $A \in \mathfrak{R}^{n \times n}$  lega lo stato all'istante  $k$  con lo stato all'istante  $k+1$  in condizioni di assenza di rumore di processo e supponendo che il sistema non sia sottoposto a nessuna sollecitazione in ingresso. La matrice  $B \in \mathfrak{R}^{n \times l}$  realizza un

legame tra l'ingresso  $u \in \mathfrak{R}^l$  e lo stato  $x$ .

La stima viene effettuata attraverso delle misurazioni successive  $z \in \mathfrak{R}^m$

$$z_k = H_k x_k + v_k \quad (1.1.2)$$

Nella (1.1.2) la matrice  $H \in \mathfrak{R}^{m \times n}$  permette di legare lo stato  $x$  alla misurazione effettuata all'istante  $k$   $z_k$ . Le variabili  $w_k$  e  $v_k$  rappresentano i rumori che affliggono rispettivamente il processo e le misure. Tali rumori si assumono essere indipendenti, bianchi e con distribuzione di probabilità normale

$$p(w) = N(0, Q) \quad (1.1.3)$$

$$p(v) = N(0, R) \quad (1.1.4)$$

Si indichi con  $\hat{x}^-$  la stima dello stato effettuata a priori e con  $\hat{x}$  la stima a posteriori eseguita a seguito della conoscenza della misurazione  $z_k$ .

Note queste due quantità si può definire l'errore di stima a priori e a posteriori come

$$e_k^- = x_k - \hat{x}^- \quad (1.1.5)$$

$$e_k = x_k - \hat{x} \quad (1.1.6)$$

La covarianza dell'errore a priori è quindi:

$$P_k^- = E \left[ e_k^- e_k^{-T} \right] \quad (1.1.7)$$

mentre la covarianza dell'errore a posteriori

$$P_k = E \left[ e_k e_k^T \right] \quad (1.1.8)$$

Derivando le equazioni del filtro di Kalman si procede ricercando un'equazione che computi una stima a posteriori  $\hat{x}$  come risultato di una combinazione lineare della

stima a priori  $\hat{x}^-$  e di una differenza pesata tra la misura attuale  $z_k$  e la sua previsione  $H_k \hat{x}^-$ .

La stima a posteriori risulta essere pari a

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (1.1.9)$$

Nella (1.1.9) la quantità  $(z_k - H_k \hat{x}_k^-)$  è detta *innovazione* o *residuo*. Essa quantifica la discrepanza tra la previsione della misurazione e la misurazione attuale.

La matrice  $K \in \mathfrak{R}^{n \times m}$ , denominata guadagno di Kalman o blending factor, si determina in modo da rendere minima la covarianza dell'errore a posteriori (1.1.6). Una possibile forma del guadagno che minimizza la (1.1.8) è la seguente:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (1.1.10)$$

Analizzando la (1.1.10) si può notare che il guadagno di Kalman è direttamente proporzionale alla covarianza dell'errore di stima e inversamente proporzionale alla varianza dell'errore di misurazione.

Si può quindi concludere che:

- tanto maggiore è l'errore che si è commesso nella stima precedente e tanto è maggiore l'affinamento della stima attuale indotto dal filtraggio
- tanto meno le misurazioni si discostano dal valore reale tanto più sarà possibile ottenere stime precise.

Le equazioni del filtro di Kalman possono essere suddivise in due gruppi: equazioni di *time update*:

$$\hat{x}_k^- = A \hat{x}_{k-1} + B u_{k-1} \quad (1.1.11)$$

$$P_k^- = A P_{k-1} A^T + Q_k \quad (1.1.12)$$

ed equazioni di *measurement update*:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (1.1.13)$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-) \quad (1.1.14)$$

$$P_k = (I - K_k H_k) P_k^- \quad (1.1.15)$$

Le prime sono responsabili della previsione dello stato attuale e della covarianza dell'errore, mentre le seconde vengono impiegate per legare la misurazione con la stima a priori con l'obiettivo di ottenere una migliore stima a posteriori.

Le equazioni di *time update* possono essere, quindi, pensate come le equazioni di previsione mentre le altre rappresentano la fase di correzione.

## 1.2 Il filtro di Kalman esteso (EKF)

Il filtro di Kalman risulta essere lo stimatore ottimo nel caso in cui il sistema risulti essere governato da un'equazione differenziale lineare stocastica.

Sfortunatamente, le transizioni di stato e le misure sono raramente funzioni lineari e questo rende il filtro di Kalman non applicabile a tutti i principali problemi robotici. Nel *filtro di Kalman esteso (EKF)* è lecito che le probabilità di transizione di stato e le misure siano governate da funzioni non lineari semplicemente approssimandole a funzioni lineari tramite un processo di *linearizzazione*.

Le tecniche per la linearizzazione di una funzione non lineare sono molteplici, il filtro di Kalman esteso utilizza un metodo denominato (*first order*) *Taylor expansion*.

In pratica, il sistema viene linearizzato attorno alla stima ed alla covarianza corrente usando le derivate parziali delle funzioni di stato e di misura.

Si consideri, quindi, un processo a tempo discreto controllato da un'equazione alle differenze stocastica non lineare:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (1.2.1)$$

con misure  $z \in \mathfrak{R}^m$  governate dall'equazione:

$$z_k = h(x_k, v_k) \quad (1.2.2)$$

Per prima cosa si procede con la linearizzazione delle (1.2.1) e (1.2.2) intorno alla stima corrente ottenendo:

$$x_k \approx \tilde{x}_k + A_k(x_{k-1} - \hat{x}_{k-1} + W_k w_{k-1}) \quad (1.2.3)$$

$$z_k \approx \tilde{z}_k + H_k(x_k - \hat{x}_k + V_k v_k) \quad (1.2.4)$$

dove

- $x_k$  e  $z_k$  sono rispettivamente i vettori dello stato e della misura attuale;
- $\tilde{x}_k$  e  $\tilde{z}_k$  sono i vettori dello stato e della misura approssimata:

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (1.2.5)$$

$$\tilde{z}_k = h(\hat{x}_k, 0) \quad (1.2.6)$$

- $\hat{x}_k$  corrisponde alla stima a posteriori dello stato all'istante  $k$ ;
- $w_k$  e  $v_k$  sono delle variabili aleatorie che rappresentano rispettivamente i rumori di processo e di misura;
- $A_k$  è la matrice Jacobiana delle derivate parziali di  $f$  rispetto a  $x$ :

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0); \quad (1.2.7)$$

- $W_k$  è la matrice Jacobiana delle derivate parziali di  $f$  rispetto a  $w$ :

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0); \quad (1.2.8)$$

- $H_k$  è la matrice Jacobiana delle derivate parziali di  $h$  rispetto a  $x$ :

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0); \quad (1.2.9)$$

- $V_k$  è la matrice Jacobiana delle derivate parziali di  $h$  rispetto a  $v$ :

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0); \quad (1.2.10)$$

Si definisca l' errore di predizione nel seguente modo

$$\tilde{e}_{xk} = x_k - \tilde{x}_k \quad (1.2.11)$$

e l'errore di misura

$$\tilde{e}_{zk} = z_k - \tilde{z}_k \quad (1.2.12)$$

Poichè nella realtà non si conosce direttamente il valore reale di  $x_k$  e  $z_k$  le (1.2.13) e (1.2.12) vanno modificate come segue:

$$\tilde{e}_{xk} \approx \tilde{A}(x_{k-1} - \hat{x}_{k-1}) + \epsilon_k \quad (1.2.13)$$

$$\tilde{e}_{zk} = \tilde{H}\tilde{e}_{xk} + \eta_k \quad (1.2.14)$$

dove  $\epsilon_k$  e  $\eta_k$  sono due nuove variabili aleatorie aventi media nulla e matrici di covarianza rispettivamente pari a  $WQW^T$  e  $VRV^T$ , con  $Q$  e  $R$  matrici di covarianza dei processi aleatori  $w$  e  $v$ .

La stima a posteriori sarà pari a:

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \quad (1.2.15)$$

Anche l'algoritmo utilizzato nel filtro di Kalman esteso, così come nel filtro di Kalman, si può suddividere in due fasi:

### 1. Time update:

- vengono generate le proiezioni dello stato del sistema, a partire dalla conoscenza del modello e della stima al passo precedente;
- vengono aggiornati i parametri probabilistici di errore sulla misura dello stato.

### 2. Measurement update:

- vengono corrette le stime prodotte al passo precedente sulla base delle misure;
- vengono aggiornati i parametri probabilistici di errore sulla misura delle uscite.

Le equazioni del *time update* possono essere rappresentate come segue:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (1.2.16)$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (1.2.17)$$

mentre quelle di *measurement update*:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (1.2.18)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (1.2.19)$$

$$P_k = (I - K_k H_k) P_k^- \quad (1.2.20)$$

Un'importante caratteristica del filtro EKF è che la matrice Jacobiana  $H_k$  serve a propagare correttamente o ad amplificare solo le componenti rilevanti dell'informazione di misura.

## 1.3 Definizione della cinematica del robot

Per cinematica di un robot si intende l'insieme di relazioni ed equazioni che descrivono l'evoluzione della sua posizione nel tempo. Nel caso in esame si considera un robot mobile avente tre gradi di libertà ovvero traslazione sugli assi  $x$  e  $y$  e rotazione intorno all'asse  $z$ .

In tutti gli algoritmi presentati si è assunto che il robot si muova all'interno di un ambiente bidimensionale con coordinate  $(x, y)$  e orientazione  $\theta$ , pertanto si può concludere che la posizione del robot sia univocamente determinata da una tripla di coordinate  $(x, y, \theta)$ .

Un primo modello che ne descrive la dinamica è:

$$x_{t+1} = x_t + \delta\rho_t \cos \theta_t \quad (1.3.1)$$

$$y_{t+1} = y_t + \delta\rho_t \sin \theta_t \quad (1.3.2)$$

$$\theta_{t+1} = \theta_t + \delta\theta_t \quad (1.3.3)$$

dove  $\delta\rho_t$  e  $\delta\theta_t$  rappresentano rispettivamente lo spostamento e la rotazione del robot nell'istante di campionamento  $\delta t$ . Le simulazioni effettuate sono basate sull'ipotesi che il robot sia dotato di una cinematica differenziale in quanto tali robot, oltre ad essere quelli più diffusi, eliminano il problema dello sterzo, permettendo di definire le rotazioni con un moto inverso di una ruota rispetto all'altra. Pertanto, risulta necessario esprimere i parametri  $\delta\rho_t$  e  $\delta\theta_t$  in funzione degli spostamenti delle singole ruote, destra e sinistra. Dalle figura è immediato definire gli spostamenti come:

$$\delta\rho = \left(L + \frac{d}{2}\right)\delta\theta \quad (1.3.4)$$

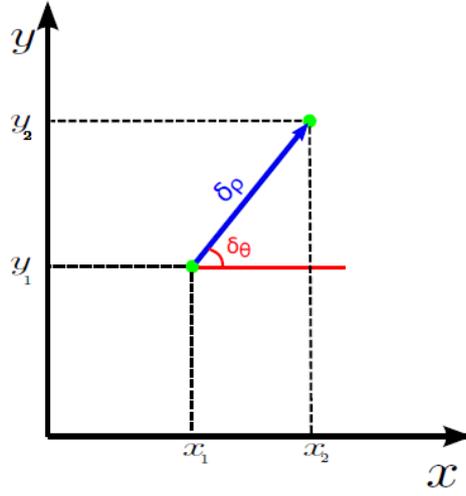


Figura 1.1: Rappresentazione di uno spostamento base del robot

$$\delta L = L\delta\theta \quad (1.3.5)$$

$$\delta R = (L + d)\delta\theta \quad (1.3.6)$$

Dividendo per 2 la somma di  $\delta R$  e  $\delta L$  e per  $d$  la loro differenza si ottengono le relazioni desiderate:

$$\frac{\delta R + \delta L}{2} = \frac{L\delta\theta + L\delta\theta + d\delta\theta}{2} = \left(L + \frac{d}{2}\right)\delta\theta = \delta\rho \quad (1.3.7)$$

$$\frac{\delta R - \delta L}{d} = \frac{L\delta\theta - L\delta\theta + d\delta\theta}{d} = \delta\theta \quad (1.3.8)$$

Ricavando  $\delta R$  dalla (1.3.7) e sostituendola nella (1.3.8) è possibile ricavare gli spostamenti delle singole ruote eliminando la dipendenza dal parametro  $L$ :

$$\frac{\delta R + \delta L}{2} = \delta\rho \quad (1.3.9)$$

$$\delta R = 2\delta\rho - \delta L \quad (1.3.10)$$

$$\delta L = \delta\rho - \frac{d}{2}\delta\theta \quad (1.3.11)$$

$$\delta R = \delta\rho + \frac{d}{2}\delta\theta \quad (1.3.12)$$

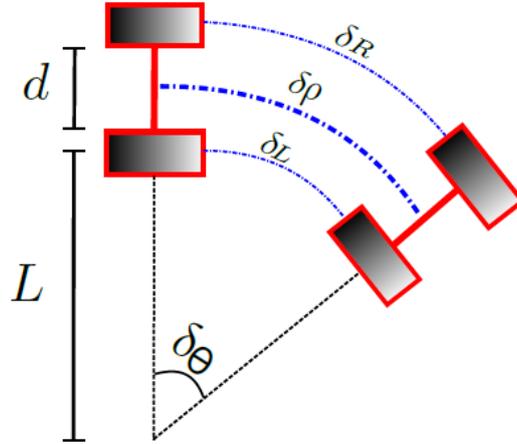


Figura 1.2: Relazione tra i movimenti delle singole ruote e lo spostamento totale del robot

Per ottenere delle stime odometriche, i motori devono essere dotati di encoder che siano in grado di memorizzare lo spostamento delle singole ruote. Introducendo, quindi, i valori di spostamento memorizzati dagli encoder e i relativi errori odometrici dovuti ad errori sistematici o casuali, si possono in ultima istanza definire gli spostamenti delle ruote come segue:

$$\delta R = \delta R^e + \eta_{R,t} \tag{1.3.13}$$

$$\delta L = \delta L^e + \eta_{L,t} \tag{1.3.14}$$

dove  $\delta R^e$  e  $\delta L^e$  rappresentano le distanze misurate dagli encoder mentre  $\eta_{R,t}$  e  $\eta_{L,t}$  rappresentano i disturbi che agiscono sulle misurazioni effettuate dagli encoder stessi. Come nei casi precedenti, anche in questo caso si modelleranno i disturbi come variabili gaussiane a media nulla:

$$\eta_{R,t} = N(0, K_R |\delta R_t^e|) \tag{1.3.15}$$

$$\eta_{L,t} = N(0, K_L |\delta L_t^e|) \tag{1.3.16}$$

con  $K_R$  e  $K_L$  costanti positive.

Sostituendo, infine, nelle (1.3.1), (1.3.2), (1.3.3) le quantità precedentemente calcolate si ottiene che la dinamica del robot può essere descritta nel seguente modo:

$$x_{t+1} = x_t + \frac{(\delta R_t^e + \eta_{R,t}) + (\delta L_t^e + \eta_{L,t})}{2} \cos \theta \quad (1.3.17)$$

$$y_{t+1} = y_t + \frac{(\delta R_t^e + \eta_{R,t}) + (\delta L_t^e + \eta_{L,t})}{2} \sin \theta \quad (1.3.18)$$

$$\theta_{t+1} = \theta_t + \frac{(\delta R_t^e + \eta_{R,t}) - (\delta L_t^e + \eta_{L,t})}{d} \quad (1.3.19)$$

# Capitolo 2

## Localizzazione mediante EKF

Il problema della localizzazione consiste nel ricavare la posizione corretta di un robot mobile a partire dalla conoscenza a priori della mappa dell'ambiente in cui il robot evolve la sua posa.

L'ambiente in questione è caratterizzato dalla presenza di punti notevoli denominati *landmark* che si trovano in posizioni note del piano e che possono essere di natura sia statica che dinamica.

La stima della posa del robot sarà effettuata con l'ausilio del filtro di Kalman esteso in quanto, sia il processo che descrive lo stato del robot, sia il modello utilizzato per le misure sono di natura non lineare.

Come sottolineato nel precedente capitolo, gli algoritmi che utilizzano il filtro di Kalman esteso si compongono di una fase di predizione e di una fase di stima.

### 2.1 Predizione

L'algoritmo di localizzazione tramite EKF utilizza il seguente modello di movimento per descrivere la dinamica del robot mobile:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \frac{u_{R,t}+u_{L,t}}{2} \cos \theta_t \\ \frac{u_{R,t}+u_{L,t}}{2} \sin \theta_t \\ \frac{u_{R,t}-u_{L,t}}{d} \end{pmatrix}$$

dove  $u_{R,t}$  e  $u_{L,t}$  rappresentano la distanza percorsa al tempo  $t$  da ruota destra e ruota sinistra rispettivamente e  $d$  corrisponde all'interasse, ovvero alla distanza tra le stesse.

Le quantità  $u_{R,t}$  e  $u_{L,t}$  sono relazionate alle letture encoder come segue:

$$u_{R,t} = u_{R,t}^e + n_{R,t} \quad (2.1.1)$$

$$u_{L,t} = u_{L,t}^e + n_{L,t} \quad (2.1.2)$$

Nella (2.1.1) e nella (2.1.2), i termini  $n_{R,t}$  e  $n_{L,t}$  corrispondono agli errori sistematici che affliggono gli encoder. Essi si suppongono essere due sequenze indipendenti, Gaussiane e a media nulla.

Nel caso in esame, si è supposto che i landmark siano di natura statica, pertanto la loro dinamica si può definire in modo molto semplice come segue:

$$x_{t+1}^i = x_t^i \quad (2.1.3)$$

$$y_{t+1}^i = y_t^i \quad (2.1.4)$$

dove  $x^i$  e  $y^i$  si devono considerare rispettivamente come le coordinate  $x$  e  $y$  dell' $i$ -esimo landmark in questione.

Si procede, quindi, alla linearizzazione dello stato definendo in primo luogo la matrice Jacobiana  $\mathcal{F}$  delle derivate parziali di primo ordine della dinamica rispetto allo stato:

$$\mathcal{F} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial x_t} & \frac{\partial x_{t+1}}{\partial y_t} & \frac{\partial x_{t+1}}{\partial \theta_t} \\ \frac{\partial y_{t+1}}{\partial x_t} & \frac{\partial y_{t+1}}{\partial y_t} & \frac{\partial y_{t+1}}{\partial \theta_t} \\ \frac{\partial \theta_{t+1}}{\partial x_t} & \frac{\partial \theta_{t+1}}{\partial y_t} & \frac{\partial \theta_{t+1}}{\partial \theta_t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{(u_{R,t}+u_{L,t})}{2} \sin \theta_t \\ 0 & 1 & \frac{(u_{R,t}+u_{L,t})}{2} \cos \theta_t \\ 0 & 0 & 1 \end{pmatrix}$$

e la matrice Jacobiana  $\mathcal{W}$  della dinamica rispetto al disturbo :

$$\mathcal{W} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial u_{R,t}} & \frac{\partial x_{t+1}}{\partial u_{L,t}} \\ \frac{\partial y_{t+1}}{\partial u_{R,t}} & \frac{\partial y_{t+1}}{\partial u_{L,t}} \\ \frac{\partial \theta_{t+1}}{\partial u_{R,t}} & \frac{\partial \theta_{t+1}}{\partial u_{L,t}} \end{pmatrix} = \begin{pmatrix} \frac{\cos \theta_t}{2} & \frac{\cos \theta_t}{2} \\ \frac{\sin \theta_t}{2} & \frac{\sin \theta_t}{2} \\ \frac{1}{d} & -\frac{1}{d} \end{pmatrix}$$

Si procede, dunque, con l'aggiornamento della matrice di covarianza:

$$\bar{\Sigma}_t = F\bar{\Sigma}_{t-1}F^T + WQW^T \quad (2.1.5)$$

dove  $Q$  rappresenta la matrice di covarianza dell'errore odometrico.

## 2.2 Correzione

Lo step relativo alla correzione richiede per prima cosa un modello di misure che, nel caso in esame, è il seguente:

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x)^2 + (y_i - y)^2} \\ \text{atan2}(y_i - y, x_i - x) - \theta \end{pmatrix}$$

dove  $x_i$  e  $y_i$  corrispondono alla posizione dell'  $i$ -esimo landmark.

Ovviamente la stima di tali quantità, come precisato precedentemente, è affetta da un rumore Gaussiano a media nulla caratterizzato dalla seguente matrice di covarianza:

$$\mathcal{R} = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$$

Si procede ora con la linearizzazione del modello delle misure definendo la matrice

Jacobiana  $\mathcal{H}_i$  nel seguente modo:

$$\mathcal{H}_i = \begin{pmatrix} \frac{\partial r_t^i}{\partial \hat{x}_t} & \frac{\partial r_t^i}{\partial \hat{y}_t} & \frac{\partial r_t^i}{\partial \hat{\theta}_t} \\ \frac{\partial \phi_t^i}{\partial \hat{x}_t} & \frac{\partial \phi_t^i}{\partial \hat{y}_t} & \frac{\partial \phi_t^i}{\partial \hat{\theta}_t} \end{pmatrix}$$

Per alleggerire la notazione si consideri:

$$q_i = (\hat{x} - x_i)^2 + (\hat{y} - y_i)^2$$

come la quantità che indica la distanza al quadrato tra la posizione del landmark  $i$ -esimo e quella stimata del robot. Nel caso in esame  $\mathcal{H}_i$  risulta essere quindi pari a:

$$\mathcal{H}_i = \begin{pmatrix} -\frac{x_i - \hat{x}}{\sqrt{q_i}} & -\frac{y_i - \hat{y}}{\sqrt{q_i}} & 0 \\ \frac{y_i - \hat{y}}{q_i} & -\frac{x_i - \hat{x}}{q_i} & -1 \end{pmatrix}$$

Dopo aver ottenuto l'espressione sopracitata si può procedere con il calcolo del guadagno di Kalman

$$K = \bar{\Sigma}_t H^T (H \bar{\Sigma}_t H^T + R)^{-1} \quad (2.2.1)$$

Indicando con  $\bar{\mu}_t$  e  $\bar{\Sigma}_t$  la media e la covarianza della stima effettuata si procede, quindi, con l'aggiornamento delle stesse come segue:

$$\bar{\mu}_t = \bar{\mu}_t + K(z_i - \hat{z}_i) \quad (2.2.2)$$

$$\bar{\Sigma}_t = (I - KH)\bar{\Sigma}_t \quad (2.2.3)$$

Nella (5.0.8), la quantità  $z_i - \hat{z}_i$  viene detta *innovation*. Essa quantifica la discrepanza tra la previsione della misurazione e la misurazione attuale e si può quindi considerare come il fattore di correzione che si deve apportare al fine di ottenere un andamento quanto più vicino a quello desiderato.

## 2.3 Risultati simulativi

La simulazione è stata effettuata con l'ausilio di *Matlab*.

La scelta di tale ambiente è dovuta alla sua estrema flessibilità e potenza, che lo rendono un programma di sviluppo saldamente affermato nella comunità scientifica.

Come campione rappresentativo è stato analizzato un percorso abbastanza lungo che presenta un andamento non rettilineo e sono state scelte in modo arbitrario delle posizioni da assegnare ai landmark.

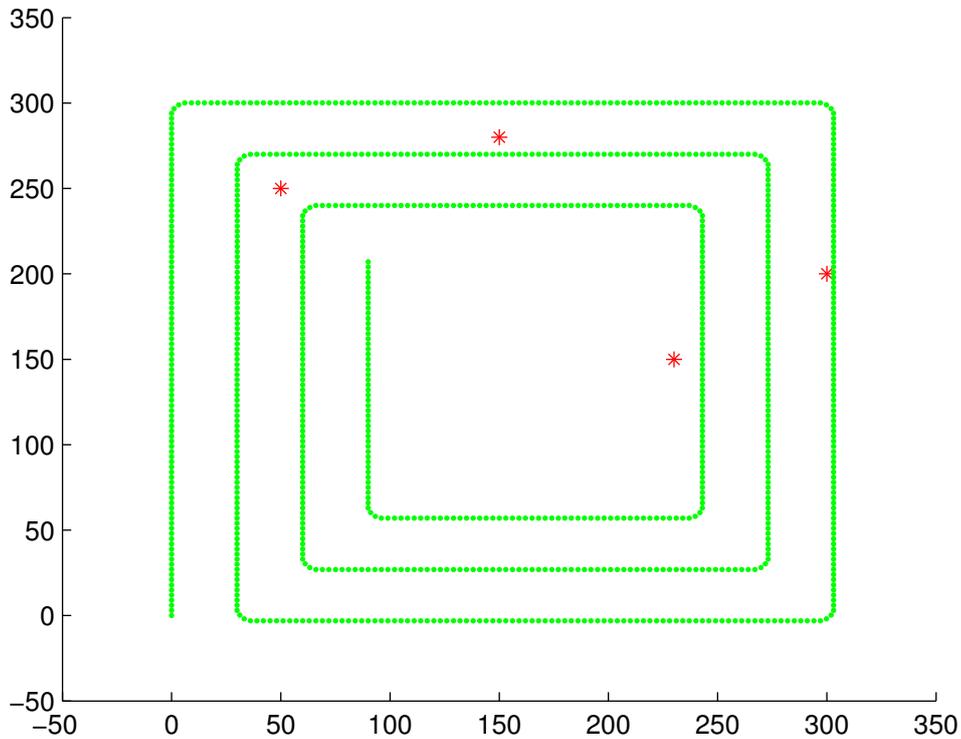


Figura 2.1: Percorso effettuato dal robot

L'integrazione dei dati odometrici porta ad un graduale allontanamento tra la posizione desiderata e quella effettiva del robot come si può notare dalla figura 2.2. Tale errore tenderà ad accumularsi: si può notare, quindi, che il discostamento tra

la traiettoria desiderata e quella dovuta all'odometria tendono a divergere sempre di più con il passare del tempo a causa del progressivo accumularsi di tale errore. L'uso

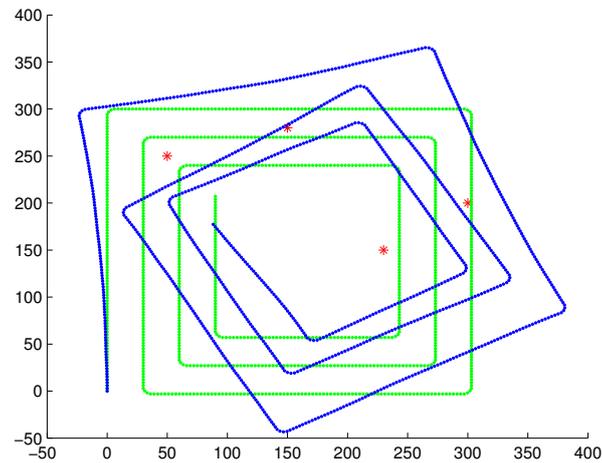


Figura 2.2: Percorso effettuato dal robot affetto da errore odometrico

del filtro di Kalman esteso annulla tale divergenza facendo convergere la traiettoria verso quella desiderata come si può notare dal percorso in nero nella figura 2.3.

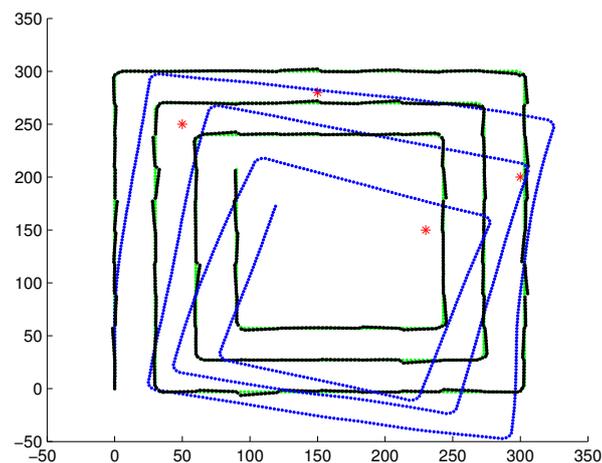


Figura 2.3: Percorso filtrato

## Capitolo 3

# Introduzione al problema dello SLAM

Il problema dello SLAM ha ragioni di esistere a causa della rumorosità che caratterizza qualsiasi applicazione ingegneristica.

Il robot effettua misure, affette da errore, sul suo moto e sull'ambiente circostante, sia per effettuare una stima corretta della sua posizione all'interno dell'ambiente non noto in cui è immerso, sia per generare una mappa affidabile.

Il problema dello SLAM è, quindi, un problema non banale: se il percorso del robot fosse conosciuto la creazione della mappa sarebbe un lavoro diretto, come pure se si conoscesse *a priori* la mappa da cui estrarre la stima della posizione; questa concorrenza di incertezze tra stima della posizione e stima dell'osservazione portano alla formulazione di soluzioni e metodi teorici che cercano di ovviare a questo inconveniente tramite approcci sostanzialmente differenti.

Da un punto di vista probabilistico, esistono due principali forme di SLAM di uguale importanza pratica.

Una di queste è conosciuta in letteratura come *online SLAM problem* poichè essa include solo la stima delle variabili che persistono al tempo  $t$ .

Molti degli algoritmi per la risoluzione del problema dello *online SLAM* sono di tipo

incrementale: essi scartano le misure e i controlli precedenti dopo averli processati.

Il secondo problema viene detto *full SLAM problem*: esso consiste nel calcolare il posterior sull'intero percorso, o *path*, effettuato dal robot al tempo  $t$ .

La sottile differenza tra i due diversi problemi, risiede nel diverso tipo di approccio utilizzato nell'implementazione degli algoritmi progettati per la risoluzione del problema.

La stima delle osservazioni e la stima della posa del robot sono problemi fortemente legati tra loro. Mentre il robot è in movimento esso accumula errore sulla sua posizione attuale, per cui il mondo percepito dai sensori è affetto dal proprio errore di misura e dalla incertezza della posa del robot. Oltretutto il movimento del robot è molto spesso affetto da errori sistematici, che si trasferiscono in modo evidente sulla costruzione della mappa stessa. In forma assolutamente generale è possibile affermare che l'errore sulla posa del robot è correlato con l'errore di costruzione della mappa. Non è quindi possibile effettuare una stima della mappa reale senza effettuare una stima della posizione del robot.

Ogni controllo ed ogni misurazione associata ad un modello di errore costituiscono un vincolo probabilistico. Ad esempio, il controllo vincola il robot rispetto a due pose successive, mentre l'osservazione vincola la posizione relativa del robot rispetto ad alcuni punti notevoli denominati *landmark* o *feature*, di natura statica o dinamica, posti in punti arbitrari del piano.

All'inizio tali relazioni costituiscono dei vincoli molto deboli ma, durante l'esecuzione del percorso, aumentano di rigidità.

### 3.1 Lo SLAM posterior

Lo SLAM posterior è una distribuzione probabilistica che costituisce una soluzione al problema dello SLAM definita come:

$$p(s_t, m | z^t, u^t) \quad (3.1.1)$$

Nella (3.1.1) si consideri  $s_t$  come la posa corrente del robot,  $m$  come l'insieme dei landmark presenti nell'ambiente considerato, che nel caso in esame sono delle entità puntuali. La probabilità *posterior* è condizionata da  $z^t$ , ovvero dall'insieme di tutte le misurazioni rilevate dai sensori e da  $u^t$  che indica l'insieme dei controlli effettuati. Il *posterior* permette non solo di ottenere una stima più accurata, sia sulla posa del robot e dei *landmark*, ma anche di quantificare l'incertezza di ogni quantità conosciuta, attraverso la reciproca correlazione di tali grandezze.

Inoltre, data la rumorosità degli ambienti dove trova largo sviluppo lo SLAM, definire quanto una soluzione sia possibile sembra di gran lunga la scelta più appropriata.

Il calcolo della (3.1.1) è effettuato in modo ottimo dal filtro di Bayes, ottenuto in modo ricorsivo sfruttando il *posterior* al tempo  $t-1$ :

$$p(s_t, \Theta | z^t, u^t) = \eta p(z_t, \theta_{n_t}) \int p(s_t | u_t, s_{t-1}) p(s_{t-1}, \theta | z^{t-1}, u^{t-1}) ds_{t-1} \quad (3.1.2)$$

Nella (3.1.2)  $\eta$  è una costante normalizzante che non dipende da  $s_t$  e da  $\Theta$ , verrà, pertanto, utilizzata nel seguito per indicare generiche costanti normalizzanti.

In generale non è possibile risolvere in forma chiusa la (3.1.2), ma è comunque possibile trovare una soluzione assumendo una forma particolare per il *posterior*.

Molte tecniche di stima, come il filtro di Kalman ed il filtro particellare, non sono

altro che un' approssimazione del filtro di Bayes.

## 3.2 Notazioni dello SLAM

Prima di descrivere gli algoritmi implementati, per una migliore comprensione del testo, si definisce la notazione utilizzata per indicare le varie grandezze in gioco.

La posa corrente del robot è denominata  $s_t$ . Nei robot planari, ovvero quelli che operano in mondi bidimensionali, essa è descritta da una terna di coordinate. Tre, infatti, è il minimo numero di variabili che bisogna utilizzare per descrivere in modo univoco la posizione di un oggetto in un piano.

La suddetta terna di coordinate sarà pertanto costituita dalla coppia di coordinate cartesiane  $(x, y)$ , che ne definiscono la posizione assoluta nel piano di lavoro, e l'orientamento  $\theta$ .

La traiettoria completa, o *path*, è data dall'insieme di tutte le pose occupate dal robot durante il suo percorso che si suppone essere composto da  $n$  step ed è denominata  $s^t$ :

$$s^t = \{s_1, s_2, \dots, s_n\} \quad (3.2.1)$$

In questo elaborato si è assunto che le *feature* dell'ambiente siano delle entità puntuali immobili in numero pari a  $N$ . Tali landmark possono essere considerati come una rappresentazione sintetica di quelle caratteristiche dell'ambiente che sono evidenti ai sensori.

L'insieme  $\Theta$  degli  $N$  landmark verrà indicato come segue:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_N\} \quad (3.2.2)$$

Durante l'esplorazione dell'ambiente, il robot riceve delle informazioni sui suoi spostamenti. Qualsiasi sia la natura di queste informazioni, esse vengono comunemente

denominate *controlli*:

$$u^t = \{u_1, u_2, \dots, u_t\} \quad (3.2.3)$$

Nel caso in esame, tali informazioni si suppongono essere restituite da encoder incrementali posti sull'asse delle ruote. Le misurazioni effettuate dal robot, comunemente, vengono definite con una coppia di variabili che descrivono, rispettivamente, la distanza e l'orientamento del robot relative alla posizione dei landmark. L'insieme delle misurazione è definito da:

$$z^t = \{z_1, z_2, \dots, z_t\} \quad (3.2.4)$$

# Capitolo 4

## EKF SLAM con corrispondenze note

Storicamente il primo algoritmo per la soluzione del problema dello SLAM si basa sull'uso del filtro di Kalman esteso.

La mappa, nell' EKF SLAM è detta *feature-based* ovvero essa è composta da un insieme di entità notevoli denominate *landmark*.

Come ogni altro algoritmo basato sull'utilizzo del filtro di Kalman esteso, anche lo SLAM basato su EKF necessita di assumere che esso sia caratterizzato da un rumore sia per quanto concerne il moto del robot, sia per le osservazioni.

Occorre, inoltre, che l'incertezza nella stima del *posterior* sia relativamente bassa poichè, altrimenti, la linearizzazione tende ad introdurre errori intollerabili.

Lo sviluppo dell' algoritmo per la risoluzione dello SLAM con corrispondenze note presenta notevoli parallelismi con quello implementato per risolvere il problema legato alla localizzazione.

La sostanziale differenza risiede nel fatto l'algoritmo dello SLAM si prefigge di stimare, oltre alla posa del robot, anche le coordinate di tutti i landmark presenti nell'ambiente.

Questo rende necessario introdurre tali coordinate nel vettore di stato:

$$\chi = \left( x \ y \ \theta \ x_1 \ y_1 \ \dots \ \dots \ x_N \ y_N \right)^T$$

dove  $x, y, \theta$  rappresentano le coordinate del robot al tempo  $t$  mentre le variabili  $x_i$  e  $y_i$  denotano le coordinate dell' $i$ -esimo landmark stimate al tempo  $t$ . La dimensione di tale vettore è pari a  $3+2N$ , dove  $N$  indica il numero di landmark considerati.

In questo elaborato, si è assunto che il robot si muova all'interno di un ambiente bi-dimensionale ed il suo moto verrà descritto attraverso le coordinate  $x, y$  (traslazioni sull'asse  $x$  e sull'asse  $y$ ) e attraverso variabile  $\theta$  che ne descrive l'orientamento.

Dal momento che molti dei concetti che verranno illustrati in seguito faranno riferimento al modello della cinematica preso in considerazione, si riportano in seguito le equazioni che lo caratterizzano sia per quanto riguarda la dinamica del robot che quella dei landmark:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ x_{1,t+1} \\ y_{1,t+1} \\ \dots \\ \dots \\ x_{N,t+1} \\ y_{N,t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \\ x_{1,t} \\ y_{1,t} \\ \dots \\ \dots \\ x_{N,t} \\ y_{N,t} \end{pmatrix} + \begin{pmatrix} \frac{u_{R,t}^e + u_{L,t}^e}{2} \cos \theta_t \\ \frac{u_{R,t}^e + u_{L,t}^e}{2} \sin \theta_t \\ \frac{u_{R,t}^e - u_{L,t}^e}{d} \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \\ 0 \end{pmatrix}$$

Si noti che nello scrivere le precedenti equazioni, attraverso le quantità  $u_{R,t}$  e  $u_{L,t}$  si è tenuto conto dell'errore gaussiano a media nulla che si suppone affligga la dinamica del sistema.

Come già precisato precedentemente, il modello di movimento è approssimato ad una funzione lineare grazie ad una *first order Taylor expansion*. Vengono quindi create delle matrici Jacobiane che permettono la realizzazione di tale linearizzazione.

Si definisce, in primo luogo, la matrice Jacobiana della dinamica rispetto allo stato:

$$\mathcal{F} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial x_t} & \frac{\partial x_{t+1}}{\partial y_t} & \frac{\partial x_{t+1}}{\partial \theta_t} \\ \frac{\partial y_{t+1}}{\partial x_t} & \frac{\partial y_{t+1}}{\partial y_t} & \frac{\partial y_{t+1}}{\partial \theta_t} \\ \frac{\partial \theta_{t+1}}{\partial x_t} & \frac{\partial \theta_{t+1}}{\partial y_t} & \frac{\partial \theta_{t+1}}{\partial \theta_t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{-(u_{R,t}+u_{L,t})}{2} \sin \theta_t \\ 0 & 1 & \frac{(u_{R,t}+u_{L,t})}{2} \cos \theta_t \\ 0 & 0 & 1 \end{pmatrix}$$

e si denota con  $W$  la matrice Jacobiana della dinamica rispetto al disturbo:

$$\mathcal{W} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial u_{R,t}} & \frac{\partial x_{t+1}}{\partial u_{L,t}} \\ \frac{\partial y_{t+1}}{\partial u_{R,t}} & \frac{\partial y_{t+1}}{\partial u_{L,t}} \\ \frac{\partial \theta_{t+1}}{\partial u_{R,t}} & \frac{\partial \theta_{t+1}}{\partial u_{L,t}} \end{pmatrix} = \begin{pmatrix} \frac{\cos \theta_t}{2} & \frac{\cos \theta_t}{2} \\ \frac{\sin \theta_t}{2} & \frac{\sin \theta_t}{2} \\ \frac{1}{d} & -\frac{1}{d} \end{pmatrix}$$

Attraverso la definizione di tali matrici è possibile ottenere come risultato la media e la covarianza della posizione stimata al tempo  $t$  dopo aver effettuato il caricamento dei controlli  $u_t$  ma prima di aver incorporato le misure  $z_t$ .

Il modello di misure è composto da una coppia di grandezze: distanza e orientamento. Esse vengono calcolate tra la posizione stimata del landmark e la posa stimata del robot.

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(\hat{x}_i - x)^2 - (\hat{y}_i - y)^2} \\ \text{atan2}(\hat{y}_i - y, \hat{x}_i - x) - \theta \end{pmatrix}$$

dove con  $\hat{x}_i$  e  $\hat{y}_i$  si indica la posizione stimata dell'  $i$ -esimo landmark.

Ovviamente la stima della distanza tra la posa del robot e i punti notevoli è affetta da un rumore Gaussiano a media nulla caratterizzato dalla seguente matrice di covarianza:

$$\mathcal{R} = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$$

Data la natura non lineare del modello delle misure, si procede con una linearizzazione definendo la matrice Jacobiana  $H$  ottenuta derivando le misure rispetto allo stato, ricordando che lo stato non è composto solo dalle coordinate che descrivono la posa del robot, bensì anche dalle coordinate dei landmark considerati.

$$\mathcal{H} = \begin{pmatrix} \frac{\partial r_t^i}{\partial \hat{x}_t} & \frac{\partial r_t^i}{\partial \hat{y}_t} & \frac{\partial r_t^i}{\partial \hat{\theta}_t} & \frac{\partial r_t^i}{\partial \hat{x}_i} & \frac{\partial r_t^i}{\partial \hat{y}_i} \\ \frac{\partial \phi_t^i}{\partial \hat{x}_t} & \frac{\partial \phi_t^i}{\partial \hat{y}_t} & \frac{\partial \phi_t^i}{\partial \hat{\theta}_t} & \frac{\partial \phi_t^i}{\partial \hat{x}_i} & \frac{\partial \phi_t^i}{\partial \hat{y}_i} \end{pmatrix}$$

Data la complessità di scrittura si consideri:

$$q = (\hat{x} - \hat{x}_i)^2 + (\hat{y} - \hat{y}_i)^2$$

come la quantità che denota la distanza al quadrato tra la posizione stimata del landmark  $i$ -esimo e quella stimata del robot. Nel caso in esame  $H$  risulta essere quindi pari a:

$$\mathcal{H} = \begin{pmatrix} \frac{\hat{x} - \hat{x}_i}{\sqrt{q}} & \frac{\hat{y} - \hat{y}_i}{\sqrt{q}} & 0 & -\frac{\hat{x} - \hat{x}_i}{\sqrt{q}} & -\frac{\hat{y} - \hat{y}_i}{\sqrt{q}} \\ \frac{\hat{y} - \hat{y}_i}{q} & \frac{\hat{x} - \hat{x}_i}{q} & -1 & -\frac{\hat{y} - \hat{y}_i}{q} & -\frac{\hat{x} - \hat{x}_i}{q} \end{pmatrix}$$

Con la linearizzazione del modello delle misure e della dinamica del sistema si conclude la fase di predizione.

Dopo aver ottenuto le precedenti grandezze, si può procedere con il calcolo del guadagno di Kalman:

$$K = \bar{\Sigma}_t H^T (H \bar{\Sigma}_t H^T + R)^{-1} \quad (4.0.1)$$

Indicando con  $\mu_t$  e  $\Sigma_t$  la media e la covarianza della stima effettuata si procede, quindi, con l'aggiornamento delle stesse come segue:

$$\bar{\mu}_t = \bar{\mu}_t + K(z_i - \hat{z}_i) \quad (4.0.2)$$

$$\bar{\Sigma}_t = (I - KH)\bar{\Sigma}_t \quad (4.0.3)$$

Nella (4.0.2), la quantità  $z_i - \hat{z}_i$  viene detta *innovation*. Tale grandezza permette di procedere ad una quantizzazione dell'errore che si è commesso e di quantificare il fattore di correzione che si deve apportare per ottenere un andamento desiderato.

## 4.1 Risultati simulativi

A proposito della risoluzione del problema dello SLAM con corrispondenze note, sono stati messi a punto tre differenti algoritmi che si differenziano tra di loro per le potenzialità che si sono supposte per i sensori.

In tutti e tre i casi considerati si è imposto al robot mobile un percorso vario che descrive una sorta di spirale all'interno del piano. Nel particolare sono stati sviluppati:

- un algoritmo in cui tutti i landmark presenti nell'ambiente sono sempre rilevabili dai sensori;
- un algoritmo in cui i sensori possono valutare la posizione di tutti e soli quei landmark che rispettano una particolare condizione sulla distanza;
- un algoritmo in cui sono visti solo quei landmark che rispettano delle particolari condizioni sia per quanto riguarda la loro posizione relativa al robot sia per quel che concerne l'orientamento.

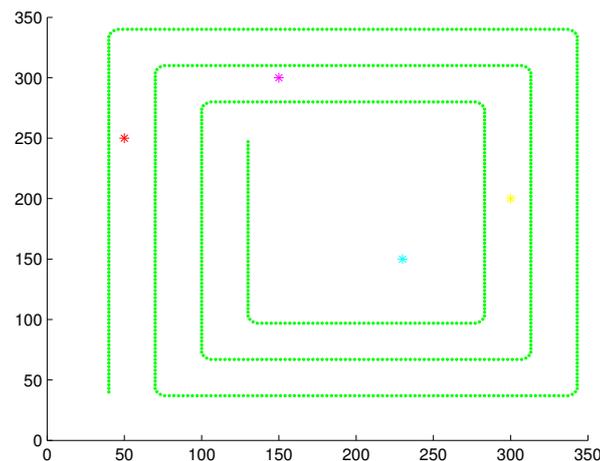


Figura 4.1: Percorso effettuato dal robot

I landmark sono posti in punti arbitrari dello spazio di lavoro del robot come si può notare dalla figura 4.1.

L'errore odometrico che affligge le misure fa sì che, in assenza di filtro, la traiettoria si discosti sempre di più dal percorso desiderato.

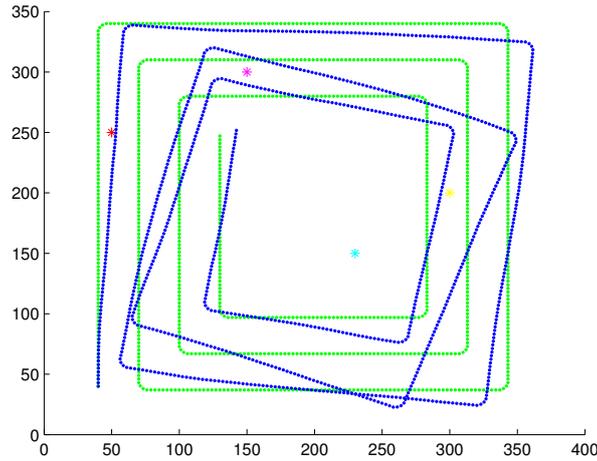


Figura 4.2: Percorso affetto da rumore odometrico

Nell'implementazione di tutti gli algoritmi proposti la stima della posizione dell'  $i$ -esimo landmark è stata effettuata attraverso la definizione di due quantità che indicano rispettivamente la distanza e l'orientamento tra la posizione effettiva del landmark e quella dove si troverebbe il robot in assenza di errore odometrico:

$$rhoReale = \sqrt{(x_i - x)^2 + (y_i - y)^2} + n_R \quad (4.1.1)$$

$$phiReale = atan2(y_i - y, x_i - x) - \theta + n_\varphi \quad (4.1.2)$$

Attraverso tali quantità possiamo stimare le coordinate  $x$ ,  $y$  del landmark nel modo seguente:

$$\hat{x}_i = \bar{x} + rhoReale_i \cdot \cos(phiReale_i + \bar{\theta}); \quad (4.1.3)$$

$$\hat{y}_i = \bar{y} + rhoReale_i \cdot \sin(phiReale_i + \bar{\theta}); \quad (4.1.4)$$

dove  $\bar{x}$ ,  $\bar{y}$  e  $\bar{\theta}$  indicano la posa e l'orientamento del robot considerando l'errore odometrico. Si possono quindi definire le seguenti quantità:

$$rhoAtteso = \sqrt{(\hat{x}_i - \hat{x})^2 + (\hat{y}_i - \hat{y})^2} \quad (4.1.5)$$

$$phiAtteso = atan2(\hat{y}_i - \hat{y}, \hat{x}_i - \hat{x}) - \hat{\theta} \quad (4.1.6)$$

Nella (4.1.6)  $\hat{x}$ ,  $\hat{y}$  e  $\hat{\theta}$  rappresentano le coordinate stimate della posa del robot. Le quantità  $rhoAtteso$  e  $phiAtteso$  rappresentano quindi la distanza e l'orientamento del landmark stimato rispetto alla posa ideale del robot.

L'innovation si può quindi definire come il discostamento tra le suddette due quantità:

$$rInnovation_i = rhoReale_i - rhoAtteso_i \quad (4.1.7)$$

$$phiInnovation_i = phiReale_i - phiAtteso_i \quad (4.1.8)$$

Nel primo algoritmo, i sensori di cui si suppone essere dotato il robot sono in grado, come detto precedentemente, di rilevare ad ogni passo tutti i landmark presenti nell'ambiente. La stima del percorso del robot e la stima dei punti notevoli ottenuti tramite questo approccio sono riportati nella figura 4.4: Si noti che la stima dei

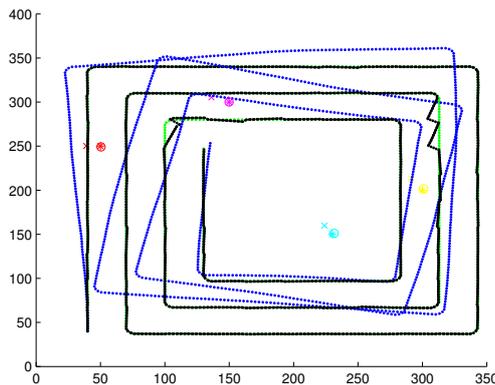


Figura 4.3: Percorso filtrato sotto l'ipotesi di visibilità di tutti i landmark

landmark è caratterizzata da tre simboli, in particolare:

- si è indicato con il simbolo ‘\*’ il punto dello spazio di lavoro del robot assegnato all’  $i$ -esimo landmark
- ‘+’ indica la posizione stimata dell’  $i$ -esimo landmark al primo passo
- con ‘o’ la stima dell’  $i$ -esimo landmark all’ultimo passo

Nel secondo algoritmo si è fatta, come detto precedentemente, l’assunzione che si possano rilevare solo i landmark che si trovano al di sotto di una distanza massima, ovvero quei landmark che rispettano la seguente relazione:

$$rhoReale_i \leq rhoMAX \quad (4.1.9)$$

Occorre notare che, in questo caso, la dimensione del vettore di stato non è costante in quanto, ad esempio, il robot potrebbe ad un istante rilevare un solo landmark mentre in seguito potrebbero essere tre i landmark che sono entrati nella sua visibilità.

A tal proposito, sono state definite delle matrici che permettono quindi, a partire dai risultati dell’iterazione precedente, di adattare le dimensioni delle matrici per permettere l’analisi corretta nel caso al tempo  $t$ . Tali matrici hanno la caratteristica di avere un numero di colonne pari alla dimensione massima del vettore di stato (nel caso in esame 11) e di avere un numero di righe pari a  $3+2N$  dove  $N$  indica il numero di landmark che rispettano la condizione (4.1.9).

I risultati ottenuti sulla stima della posizione del robot e dei landmark sono riportati nella figura 4.4. Nell’implementazione del terzo ed ultimo algoritmo si prevede che i sensori riescano a vedere solo quei landmark che rispettano sia una condizione sulla distanza, analoga a quella in (4.1.9), sia una condizione sull’angolo  $\phi$ . Tale condizione è stata considerata essere del seguente tipo:

$$phiReale_i \leq phiMAX \quad (4.1.10)$$

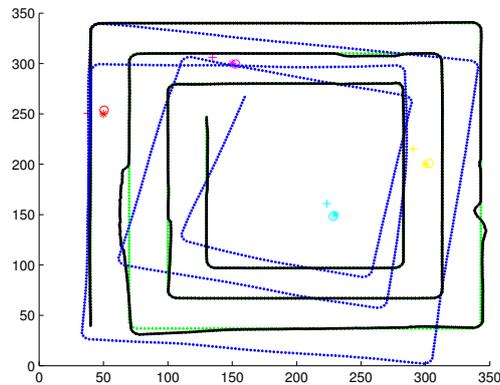


Figura 4.4: Percorso filtrato sotto l'ipotesi di limitazione sensoristica sulla distanza

Anche in questo caso, come nell'implementazione del secondo algoritmo, la dimensione del vettore di stato e conseguentemente tutte le matrici che caratterizzano la simulazione non rimangono invariate durante tutta l'esecuzione. Questo rende necessario la definizione di matrici, analoghe a quelle descritte nel caso precedente, che siano in grado di aggiornare correttamente senza perdita di dati la dimensione del vettore di stato e tutte le altre matrici coinvolte.

La stima della posa del robot e delle coordinate dei landmark risulta, anche in questo caso, soddisfacente come si può notare dalla figura 4.5.

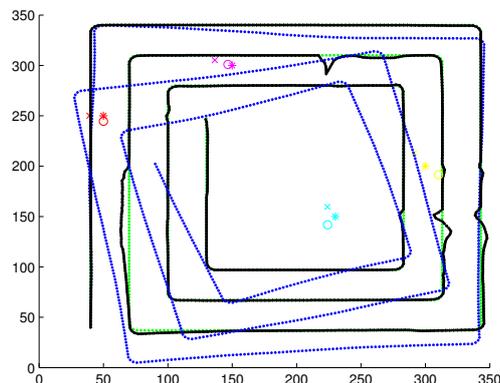


Figura 4.5: Percorso filtrato con limitazione sensoristica sulla distanza e sull'orientamento

## Capitolo 5

# EKF SLAM con corrispondenze non note

Lo SLAM tramite filtro di Kalman esteso nel caso di corrispondenze note è ora esteso nel caso più generale di EKF SLAM che usa in modo incrementale il criterio di massima verosimiglianza per determinare le corrispondenze.

Una delle principali differenze con il caso in cui le corrispondenze sono conosciute risiede nel fatto che negli input dell'algoritmo manca l'etichetta identificativa del landmark anche se è presente la dimensione temporanea della mappa che verrà indicata con  $N_t$ . Il modello di movimento assegnato al robot è analogo a quello visto nel caso precedente ma, per maggiore chiarezza, si riporta di seguito:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \frac{u_{R,t}^e + u_{L,t}^e}{2} \cos \theta_t \\ \frac{u_{R,t}^e + u_{L,t}^e}{2} \sin \theta_t \\ \frac{u_{R,t}^e - u_{L,t}^e}{d} \end{pmatrix}$$

Si procede comunque alla linearizzazione di tale modello attraverso la definizione delle matrici Jacobiane  $F$  della dinamica rispetto allo stato:

$$\mathcal{F} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial x_t} & \frac{\partial x_{t+1}}{\partial y_t} & \frac{\partial x_{t+1}}{\partial \theta_t} \\ \frac{\partial y_{t+1}}{\partial x_t} & \frac{\partial y_{t+1}}{\partial y_t} & \frac{\partial y_{t+1}}{\partial \theta_t} \\ \frac{\partial \theta_{t+1}}{\partial x_t} & \frac{\partial \theta_{t+1}}{\partial y_t} & \frac{\partial \theta_{t+1}}{\partial \theta_t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{-(u_{R,t}+u_{L,t})}{2} \sin \theta_t \\ 0 & 1 & \frac{(u_{R,t}+u_{L,t})}{2} \cos \theta_t \\ 0 & 0 & 1 \end{pmatrix}$$

e  $W$  della dinamica rispetto al disturbo:

$$\mathcal{W} = \begin{pmatrix} \frac{\partial x_{t+1}}{\partial u_{R,t}} & \frac{\partial x_{t+1}}{\partial u_{L,t}} \\ \frac{\partial y_{t+1}}{\partial u_{R,t}} & \frac{\partial y_{t+1}}{\partial u_{L,t}} \\ \frac{\partial \theta_{t+1}}{\partial u_{R,t}} & \frac{\partial \theta_{t+1}}{\partial u_{L,t}} \end{pmatrix} = \begin{pmatrix} \frac{\cos \theta_t}{2} & \frac{\cos \theta_t}{2} \\ \frac{\sin \theta_t}{2} & \frac{\sin \theta_t}{2} \\ \frac{1}{d} & -\frac{1}{d} \end{pmatrix}$$

Si procede dunque all'aggiornamento della matrice di covarianza:

$$\bar{\Sigma}_t = F \Sigma_{t-1} F^T + W Q W^T \quad (5.0.1)$$

dove  $Q$  rappresenta la matrice di covarianza dell'errore odometrico:

$$Q = \begin{pmatrix} K_R |u_R^e| & 0 \\ 0 & K_L |u_L^e| \end{pmatrix}$$

Il caricamento delle misure, invece, è completamente differente.

L'algoritmo, in primo luogo, ipotizza di vedere un nuovo landmark con indice  $N_t + 1$ .

Tale indice supera di uno la dimensione della mappa calcolata fino a quel momento.

La posizione del nuovo landmark è calcolata a partire dalla conoscenza della stima sulla posa del robot e delle rilevazioni da parte dei sensori della distanza e dell'orientamento, *bearing*, del nuovo landmark rispetto alla posa:

$$\begin{pmatrix} \hat{x}_{N_t+1} \\ \hat{y}_{N_t+1} \end{pmatrix} = \begin{pmatrix} \hat{x}_t \\ \hat{y}_t \end{pmatrix} + \begin{pmatrix} r^i \cos(\phi^i + \hat{\theta}) \\ r^i \sin(\phi^i + \hat{\theta}) \end{pmatrix}$$

In seguito viene effettuato, per ogni landmark catalogato precedentemente nella mappa ed anche per quello di cui si ipotizza l'esistenza, il calcolo di diverse grandezze. In particolare, si indichi con  $\delta_k$  il vettore le cui due componenti corrispondono alla differenza delle coordinate del  $k$ -esimo landmark e le coordinate della posa del robot calcolate rispettivamente lungo l'asse  $x$  e  $y$ :

$$\delta_k = \begin{pmatrix} \delta_{k,x} \\ \delta_{k,y} \end{pmatrix} = \begin{pmatrix} \hat{x}_k - \hat{x} \\ \hat{y}_k - \hat{y} \end{pmatrix}$$

Si può, quindi, definire con  $q_k$  la quantità che corrisponde alla distanza al quadrato tra la posa del robot e il  $k$ -esimo landmark nel seguente modo:

$$q_k = \delta_k^T \delta_k \quad (5.0.2)$$

La definizione di tali grandezze ci permette di definire il vettore delle misure in modo più compatto.

$$\dagger_k = \begin{pmatrix} \sqrt{q_k} \\ \text{atan2}(\delta_{k,y}, \delta_{k,x}) \end{pmatrix}$$

Data la natura non lineare del modello delle misure si procede ad una linearizzazione definendo la matrice Jacobiana  $H_k$  delle derivate delle misure rispetto allo stato relative al  $k$ -esimo landmark considerato:

$$\mathcal{H}_k = \begin{pmatrix} -\frac{\delta_{k,x}}{\sqrt{q_k}} & -\frac{\delta_{k,y}}{\sqrt{q_k}} & 0 & \frac{\delta_{k,x}}{\sqrt{q_k}} & \frac{\delta_{k,y}}{\sqrt{q_k}} \\ \frac{\delta_{k,y}}{q_k} & -\frac{\delta_{k,x}}{q_k} & -1 & -\frac{\delta_{k,y}}{q_k} & \frac{\delta_{k,x}}{q_k} \end{pmatrix}$$

In seguito si calcola per ogni landmark (incluso quello ipotizzato) la seguente quantità

che prende il nome di *distanza di Mahalanobis* :

$$\pi_k = (z_t^i - \hat{z}_t^k)^T (H_t^k \bar{\Sigma}_t [H_t^k]^T + R)^{-1} (z_t^i - \hat{z}_t^k) \quad (5.0.3)$$

dove  $R$  indica la matrice di covarianza dell'errore delle misure supposto essere Gaussiano a media nulla.

Il criterio di massima verosimiglianza permette di stabilire se il landmark ipotizzato è realmente una nuova feature da aggiungere alla mappa incrementandone di uno la dimensione oppure no.

Per effettuare ciò si definisce in primo luogo una soglia per la creazione di un nuovo landmark:

$$\pi_{N_t+1} = \alpha \quad (5.0.4)$$

Un nuovo landmark viene creato se la distanza di Mahalanobis di tutti i landmark esistenti nella mappa supera il valore  $\alpha$ . Si può ora definitivamente applicare il criterio di massima verosimiglianza:

$$j(i) = \operatorname{argmin}_k \pi_k \quad (5.0.5)$$

$$N_t = \max \{N_t, j(i)\} \quad (5.0.6)$$

In pratica, attraverso il calcolo delle quantità definite in (5.0.5) e (5.0.6), si verifica se le osservazioni effettuate possono essere o meno associate a uno dei landmark già visti; se questo non è possibile viene incrementata di uno la dimensione della mappa e si può procedere al calcolo della correzione dopo aver adattato le dimensioni delle matrici coinvolte alla dimensione attuale della mappa.

Si procede quindi al calcolo del Kalman Gain:

$$K = \bar{\Sigma}_t H^T (H \bar{\Sigma}_t H^T + R)^{-1} \quad (5.0.7)$$

ed al conseguente aggiornamento della media e della covarianza attraverso le quali si esprime la posa del robot e la posizione di tutti i landmark presenti attualmente nella mappa:

$$\bar{\mu}_t = \bar{\mu}_t + K(z_i - \hat{z}_i) \quad (5.0.8)$$

$$\bar{\Sigma}_t = (I - KH)\bar{\Sigma}_t \quad (5.0.9)$$

## 5.1 Risultati simulativi

Per effettuare la simulazione è stato preso in considerazione un percorso piuttosto lungo e vario come si può notare dalla figura successiva. A differenza del caso di

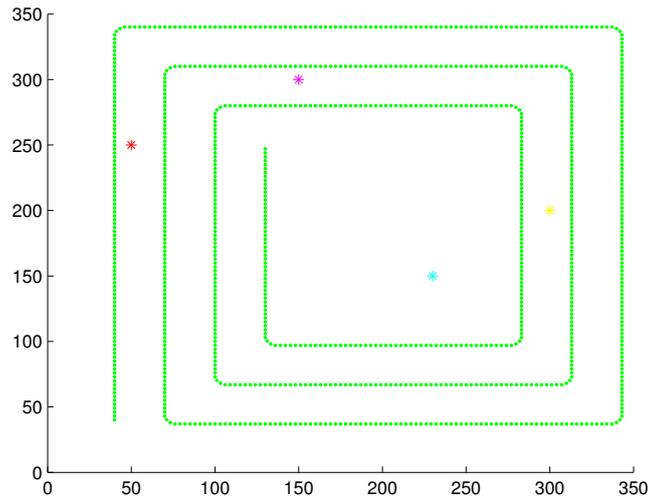


Figura 5.1: Percorso effettuato dal robot

corrispondenze note, non vi sono landmark nel piano di lavoro del robot in quanto all' inizio la dimensione della mappa è nulla.

La stima della posizione dei landmark avviene in un modo differente rispetto a quello precedentemente visto. Infatti, è possibile creare un nuovo landmark se la Mahalanobis distance di tutti i landmark esistenti nella mappa supera il valore  $\alpha$  definito, nel caso in esame in accordo con l'errore random che affligge la stima sulla posa del robot e sulle osservazioni.

Ad ogni passo l'algoritmo tenta di aumentare di uno la dimensione della mappa attuale sulla base delle osservazioni effettuate dai sensori che si è supposto riescano a rilevare quei landmark che rispettano la seguente relazione:

$$RhoReale_i \leq RhoMax \quad (5.1.1)$$

Per comprendere se il landmark che cerca di stimare è effettivamente un ‘nuovo’ landmark applica il criterio della massima verosimiglianza come descritto precedentemente.

I landmark sono dei punti notevoli definiti precedentemente nello spazio di lavoro del robot. Per effettuare quindi la stima della loro posizione e l’aggiornamento della mappa sono stati creati due vettori:

- un vettore di dimensione pari al numero di landmark presenti (nel caso in esame 4) le cui componenti sono pari a 1 se e solo se il landmark corrispondente rispetta la relazione (5.1.1);
- un vettore, anch’ esso di dimensione pari al numero di landmark presenti, che permette di stabilire l’ordine di inserimento dei landmark nella mappa.

Come detto precedentemente, l’ EKF SLAM nel caso di corrispondenze non note opera applicando in modo incrementale il criterio di massima verosimiglianza. Risulta chiaro, pertanto, che la dimensione della mappa può aumentare, ad ogni passo, al massimo di un’ unità.

Non è quindi possibile che la mappa al passo  $k$ , aumenti di due unità nonostante più landmark soddisfino la condizione (5.1.1).

Al fine di ovviare a tale situazione, nel caso in cui si verifichi la situazione di cui sopra, si sceglie il landmark più vicino.

Dalla figura successiva, si può notare che il cammino reale (in verde) e quello stimato sono quasi sempre sovrapposti e lo stesso per ciò che concerne la stima finale dei landmark (indicata con il simbolo ‘o’).

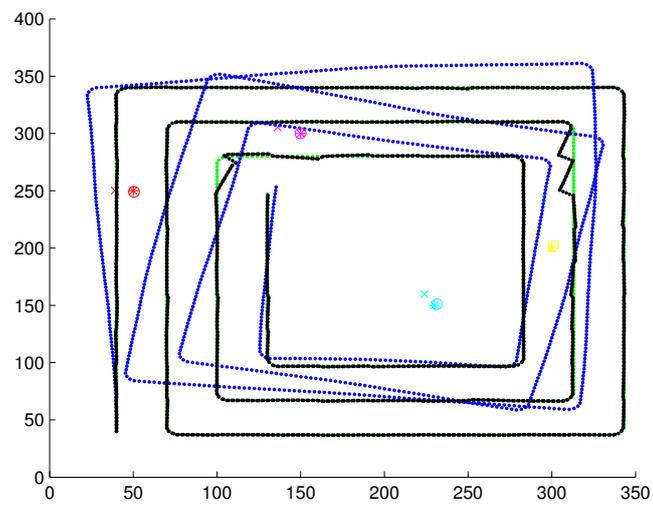


Figura 5.2: Percorso filtrato nel caso di corrispondenze non note

# Capitolo 6

## Conclusioni e sviluppi futuri

Lo scopo del lavoro di tesi è stato quello di implementare degli algoritmi basati sull'utilizzo del filtro di Kalman esteso, in grado di effettuare con successo la stima della posizione sia del robot che dei landmark presenti nell'ambiente.

Sono stati presentati due diversi scenari: il caso in cui la dimensione della mappa è nota a priori e quello in cui non lo è.

In entrambi i casi si sono ottenuti risultati piuttosto soddisfacenti sia per la stima della posa del robot sia per la costruzione della mappa.

Data la complessità dell'EKF SLAM ( $O(n^2)$ ) non è possibile operare in questo modo in ambienti in cui il numero di punti notevoli da riconoscere è estremamente elevato. Risultati simulativi dimostrano, infatti, che algoritmi di questo tipo applicabili in ambienti con un numero di landmark inferiori al centinaio. Altro punto debole dell'EKF SLAM risiede nel fatto che gli errori che affliggono il processo e la misura non devono essere troppo elevati, in quanto questi potrebbero introdurre errori molto gravi nella fase di linearizzazione.

L'applicazione della tecnica di SLAM mostrata in questa tesi può essere sicuramente ampliata affiancando all'algoritmo che risolve il problema della stima, un algoritmo

di pianificazione della traiettoria (path planning), che, una volta generata la mappa, permetta di trovare la traiettoria ottima per raggiungere più punti nella mappa.

Data la natura non sperimentale della tesi un ovvio sviluppo è quello di trasferire tutti gli algoritmi presentati in un contesto reale al fine di verificarne l'efficacia.

A proposito della complessità dell'algoritmo, un'ultima estensione potrebbe essere il superamento dell' EKF SLAM implementando tecniche che siano in grado di operare in modo efficiente in ambienti molto densi di landmark. Una di queste tecniche è quella nota in letteratura come *FastSLAM*.

# Elenco delle figure

1.1	Rappresentazione di uno spostamento base del robot . . . . .	14
1.2	Relazione tra i movimenti delle singole ruote e lo spostamento totale del robot . . . . .	15
2.1	Percorso effettuato dal robot . . . . .	22
2.2	Percorso effettuato dal robot affetto da errore odometrico . . . . .	23
2.3	Percorso filtrato . . . . .	23
4.1	Percorso effettuato dal robot . . . . .	34
4.2	Percorso affetto da rumore odometrico . . . . .	35
4.3	Percorso filtrato sotto l'ipotesi di visibilità di tutti i landmark . . . . .	36
4.4	Percorso filtrato sotto l'ipotesi di limitazione sensoristica sulla distanza	38
4.5	Percorso filtrato con limitazione sensoristica sulla distanza e sull'orientamento . . . . .	38
5.1	Percorso effettuato dal robot . . . . .	44
5.2	Percorso filtrato nel caso di corrispondenze non note . . . . .	46

# Bibliografia

- [1] S.Thrun - W.Burgard - D.Fox, "*Probabilistic Robotics*", the MIT Press,2005.
- [2] E.Di Giampaolo - F.Martinelli, "*Robot localization by sparse and passive RFID tags*", ISIE, 2010.
- [3] J.A. Castellanos - J.D. Tard?s - G. Schmidt, "*Building a global map of the environment of a mobile robot: The importance of correlations*", in Proc. IEEE Int. Conf. Robot. Automat., 1997, pp. 1053-1059.
- [4] Greg Welch - Gary Bishop, "*An Introduction to the Kalman Filter*", Department of Computer Science University of North Carolina at Chapel Hill, 2004.
- [5] R. Smith - P. Cheesman, "*On the representation of spatial uncertainty*", Int. J. Robot. Res., vol. 5, no. 4, pp. 56-68, 1987.
- [6] H.F. Durrant - Whyte, "*Uncertain geometry in robotics*", IEEE Trans. Robot. Automat., vol. 4, no. 1, pp. 23-31, 1988.
- [7] Hugh Durrant - Whyte and Tim Bailey, "*Simultaneous Localization and Mapping: Tutorial*", IEEE Robotics and Automation Magazine, 2006.
- [8] N. Ayache - O. Faugeras, "*Building, registrating, and fusing noisy visual maps*", Int. J. Robot. Res., vol. 7, no. 6, pp. 45-65, 1988.

- 
- [9] J. Crowley, “*World modeling and position estimation for a mobile robot using ultra-sonic ranging*”, in Proc. IEEE Int. Conf. Robot. Automat., 1989, pp. 674-681.
- [10] R. Chatila - J.P. Laumond, “*Position referencing and consistent world modeling for mobile robots*”, in Proc. IEEE Int. Conf. Robot. Automat., 1985, pp. 138-143.
- [11] S. Thrun, D. Fox, and W. Burgard, “*A probabilistic approach to concurrent mapping and localization for mobile robots*”, Mach. Learning, vol. 31, no. 1, pp. 29-53, 1998.
- [12] R. Smith - M. Self - P. Cheeseman, “*Estimating uncertain spatial relationships in robotics*”, in Autonomous Robot Vehicles, I.J. Cox and G.T. Wilfon, Eds. New York: Springer-Verlag, pp. 167-193, 1990.
- [13] H. Durrant-Whyte - D. Rye - E. Nebot, “*Localisation of automatic guided vehicles*”, in Robotics Research: The 7th International Symposium (ISRR’95), G. Giralt and G. Hirzinger, Eds. New York: Springer Verlag, pp. 613-625, 1996.
- [14] M. Csorba - H.F. Durrant-Whyte, “*A new approach to simultaneous localisation and map building*”, in Proc. SPIE Aerosense, Orlando, FL, 1996.
- [15] J.J. Leonard - H.J.S. Feder, “*A computational efficient method for large-scale concurrent mapping and localisation*”, in Robotics Research, The Ninth International Symposium (ISRR’99), J. Hollerbach and D. Koditscheck, Eds. New York: Springer-Verlag, pp. 169-176, 2000.
- [16] J.A. Castellanos - J.M. Martnez - J. Neira - J.D. Tard?s, “*Experiments in multi-sensor mobile robot localization and map building*”, in Proc. 3rd IFAC Sym. Intell. Auton. Vehicles, 1998, pp. 173-178.

- 
- [17] J. Guivant - E.M. Nebot - S. Baiker, “*Localization and map building using laser range sensors in outdoor applications*”, J. Robot. Syst., vol. 17, no. 10, pp. 565-583, 2000.
- [18] S.B. Williams - P. Newman - G. Dissanayake - H.F. Durrant-Whyte, “*Autonomous underwater simultaneous localisation and map building*”, in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), San Francisco, CA, Apr. 2000, pp. 1793-1798.
- [19] J. Hollerbach - D. Koditscheck, Eds., *Robotics Research, The Ninth International Symposium (ISRR'99)*, New York: Springer-Verlag, 2000.
- [20] M. Csorba, “*Simultaneous Localisation and Map Building*”, Ph.D. dissertation, Univ. Oxford, 1997.