



**UNIVERSITÀ DEGLI STUDI DI ROMA  
TOR VERGATA**

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

A.A. 2008/2009

Tesi di Laurea

Filtro Particellare per la localizzazione di un robot mobile  
dotato di sensori RFID

**RELATORE**

Ing. Francesco Martinelli

**CANDIDATO**

Alessio Mannisi

*Ringrazio...*

*l'ingegnere Francesco Martinelli, relatore della tesi,*

*che mi ha introdotto ad un così*

*appassionante argomento*

*la mia famiglia che mi è sempre stata accanto, in particolare*

*mia Madre, che mi ha sempre spronato ad andare avanti,*

*i miei zii Mauro e Mario, anche loro ingegneri, che mi hanno sostenuto*

*nell'affrontare gli esami più difficili*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Il problema della localizzazione globale</b>	<b>5</b>
2.1	Dinamica del robot . . . . .	5
2.2	I sensori RFID . . . . .	7
2.3	Il filtraggio . . . . .	8
<b>3</b>	<b>Sistemi RFID</b>	<b>10</b>
3.1	Gli RFID nella localizzazione . . . . .	10
3.1.1	RFID Tag o transponder . . . . .	11
3.1.2	RFID Reader . . . . .	12
3.2	Vantaggi degli RFID . . . . .	12
3.3	Difetti degli RFID . . . . .	13
3.4	Tipologie d'uso degli RFID . . . . .	14
<b>4</b>	<b>Filtro Particellare</b>	<b>15</b>
4.1	Introduzione al PF . . . . .	15
4.2	L'algoritmo del PF . . . . .	16
4.2.1	Schema dell'algoritmo . . . . .	17
4.2.2	Legenda dell'algoritmo . . . . .	18
4.3	Inizializzazione . . . . .	18
4.3.1	Discretizzazione . . . . .	19
4.3.2	Percorso . . . . .	19
4.4	Spostamento delle particelle . . . . .	20
4.5	Assegnazione delle probabilità . . . . .	21
4.6	Resampling . . . . .	21
4.7	Stima della posizione . . . . .	22
4.8	Varianti del PF . . . . .	23

<b>5</b>	<b>Risultati simulativi</b>	<b>24</b>
5.1	Errore di stima . . . . .	24
5.2	Simulazione ( $M_p=1000$ ) . . . . .	25
5.2.1	Ambiente e tags . . . . .	25
5.2.2	Il disturbo di lettura $\lambda$ . . . . .	25
5.2.3	Il robot . . . . .	26
5.2.4	L'indice d'errore J . . . . .	26
5.3	Simulazione al variare di $M_p$ . . . . .	27
5.3.1	Grafico degli errori $J_1$ e $J_2$ al variare di $M_p$ . . . . .	27
5.3.2	Complessità computazionale al variare di $M_p$ . . . . .	30
5.4	Simulazione al variare del numero di tag e $\rho_a$ . . . . .	30
5.4.1	$n_{tag} = 3, M_p = 1000, \lambda = 0$ . . . . .	32
5.4.2	$n_{tag} = 4, M_p = 1000, \lambda = 0$ . . . . .	34
5.4.3	$n_{tag} = 5, M_p = 1000, \lambda = 0$ . . . . .	36
5.4.4	Osservazioni . . . . .	38
5.5	Filtri a confronto . . . . .	39
<b>6</b>	<b>Conclusioni</b>	<b>40</b>

# Capitolo 1

## Introduzione

In questa tesi intendiamo mostrare i metodi di localizzazione di un robot a partire dalla misura odometrica e di quella sensoriale, dove per stima odometrica si intende una misura derivante dagli spostamenti delle ruote di un robot, mentre per stima sensoriale si intende la lettura tramite sensori, corrispondenti nel nostro caso agli RFID. Lo scopo dell'utilizzo combinato dei due metodi è legato alla minimizzazione dell'errore di localizzazione. La stima odometrica presenta sempre, infatti, un margine d'errore che non può essere quantificato e che ogni volta che la misurazione viene ripetuta non fa altro che sommarsi a quello della misura precedente accumulandosi. L'aumento di tale errore sulle misure comporta di conseguenza l'impossibilità di una localizzazione efficiente del robot. Per questo motivo la misura odometrica viene supportata dalla lettura da RFID. La fusione tra queste due misure può essere effettuata attraverso vari metodi, tra cui il filtro di Kalman e il filtro particellare.

Il filtro particellare (PF) è uno strumento molto preciso ma presenta una notevole complessità computazionale, come verrà evidenziato più avanti. Il filtro di Kalman viene utilizzato nella sua versione estesa (EKF) per il sistema non lineare preso in esame e risulta meno efficace come risultati, ma più efficiente dal punto di vista della velocità d'esecuzione. Per adattare le misure come ingressi per l'EKF vengono utilizzati vari metodi, che tuttavia vengono appena accennati nella presente trattazione (vedi Sezione 2.3) che approfondisce invece il funzionamento del PF. Nella Sezione 5.5 verranno messi a confronto i risultati del PF e dell'EKF (applicando i vari metodi).

Per studiare l'efficacia di questi filtri è stato creato un ambiente di simu-

lazione tramite MATLAB che crea il tragitto effettivo compiuto dal robot in una stanza sovrapponendovi quello ottenuto tramite localizzazione. L'ambiente in cui è stato fatto questo esperimento virtuale è costituito da una stanza di dimensioni note ( $4m \times 4m$ ) sul cui soffitto sono stati messi degli RFID tags. Il robot naviga nella stanza ed ha un'antenna rivolta verso l'alto che individua e riconosce i tags in cui si imbatte. Sono state fatte delle simulazioni per minimizzare l'errore  $J$  al variare del range dell'antenna e del numero di tags i cui risultati sono mostrati nella Sezione 5.4.

## Capitolo 2

# Il problema della localizzazione globale

Per localizzazione globale di un robot si intende la stima della posizione effettuata dal robot in base ai dati provenienti dall'ambiente esterno raccolti tramite appositi sensori, senza conoscere la posizione di partenza. Per ottenere la stima migliore vengono perciò "fusi" tramite un filtro i dati provenienti da più sensori. Questo problema risulta importante dal punto di vista pratico perché mira a fornire al robot la consapevolezza della propria posizione che potrebbe essere sfruttata in macchine più complesse per rispondere prontamente alle difficoltà incontrate lungo il tragitto.

### 2.1 Dinamica del robot

L'odometria è la stima della posizione di un robot a partire dai movimenti compiuti dalle sue ruote. Poiché il robot si muove nello spazio ci si aspetterebbe un modello di riferimento a tempo continuo, ma i movimenti del robot vengono tradotti dai dispositivi encoder collegati alle ruote in una serie di "passi" che dipendono dal tempo di campionamento utilizzato. Quindi possiamo descrivere la dinamica del robot tramite il seguente modello discreto:

$$\begin{cases} x_{k+1} = x_k + \frac{\delta_R \cdot u_{R,k} + \delta_L \cdot u_{L,k}}{2} \cos(\theta_k) \\ y_{k+1} = y_k + \frac{\delta_R \cdot u_{R,k} + \delta_L \cdot u_{L,k}}{2} \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \frac{\delta_R \cdot u_{R,k} - \delta_L \cdot u_{L,k}}{\Delta} \end{cases} \quad (2.1)$$

in cui:

- $[x, y, \theta]$  è la terna che indica ascissa, ordinata e orientamento del robot;
- $u_{R,k}$  e  $u_{L,k}$  sono gli spostamenti della ruota destra e sinistra;
- $\delta_R$  e  $\delta_L$  sono gli errori sistematici di ciascuna ruota, legati cioè a difetti di costruzione (ruote con diametro diverso);
- $\Delta$  è la distanza tra le ruote del robot.

Tuttavia gli spostamenti reali del robot non coincidono con quelli letti. Infatti è sempre presente un certo disturbo nelle letture da encoder. Per tale motivo gli spostamenti delle ruote letti sono sempre corrotti da errore:

$$u_{R,k}^e = u_{R,k} + n_{R,k}$$

$$u_{L,k}^e = u_{L,k} + n_{L,k}$$

dove  $u_{R,k}^e$  e  $u_{L,k}^e$  sono gli spostamenti letti delle ruote e  $n_{R,k}$  e  $n_{L,k}$  sono gli errori di lettura sulle ruote destra e sinistra supposti gaussiani e perciò definiti come:

$$n_{R,k} \sim \aleph(0, K_R \cdot |u_{R,k}|)$$

$$n_{L,k} \sim \aleph(0, K_L \cdot |u_{L,k}|)$$

dove il simbolo  $\aleph(0, K \cdot |u_k|)$  indica una distribuzione di probabilità gaussiana con media nulla e varianza  $\sigma^2 = K \cdot |u_k|$  (vedi [3] e [4]) e  $K_R$  e  $K_L$  sono due costanti d'errore odometriche delle ruote definite positive. La (2.1) letta dal robot diventa quindi:

$$\psi_{k+1}(x_{r,k}) := \begin{cases} x_{k+1} = x_k + \frac{\delta_R(u_{R,k} + n_{R,k}) + \delta_L(u_{L,k} + n_{L,k})}{2} \cos(\theta_k) \\ y_{k+1} = y_k + \frac{\delta_R(u_{R,k} + n_{R,k}) + \delta_L(u_{L,k} + n_{L,k})}{2} \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \frac{\delta_R(u_{R,k} + n_{R,k}) - \delta_L(u_{L,k} + n_{L,k})}{\Delta} \end{cases} \quad (2.2)$$

D'ora in avanti per comodità si indicherà con  $x_r = [x, y, \theta]^T$  la posa del robot.

## 2.2 I sensori RFID

I sensori RFID sono la tecnologia di supporto utilizzata per ovviare agli errori introdotti dalle letture encoder mostrati in (2.2). Tuttavia i soli RFID non sono sufficienti a stimare la posizione del robot in quanto permettono di individuare solo se il robot attraversa una determinata regione, non fornendo quindi la sua posizione esatta; ad esempio, in merito alla Figura 2.2, se il robot si trova in  $[1 ; 3.5]$  non riesce a distinguere la sua posizione rispetto ad un qualsiasi altro punto appartenente alla regione di tag  $[1 ; 3]$ .

Combinando agli RFID la stima odometrica, invece, il robot può utilizzare quest'ultima per orientarsi all'interno della regione in cui si trova.

Della tecnologia RFID e del suo funzionamento verrà parlato più ampiamente nel Capitolo 3.

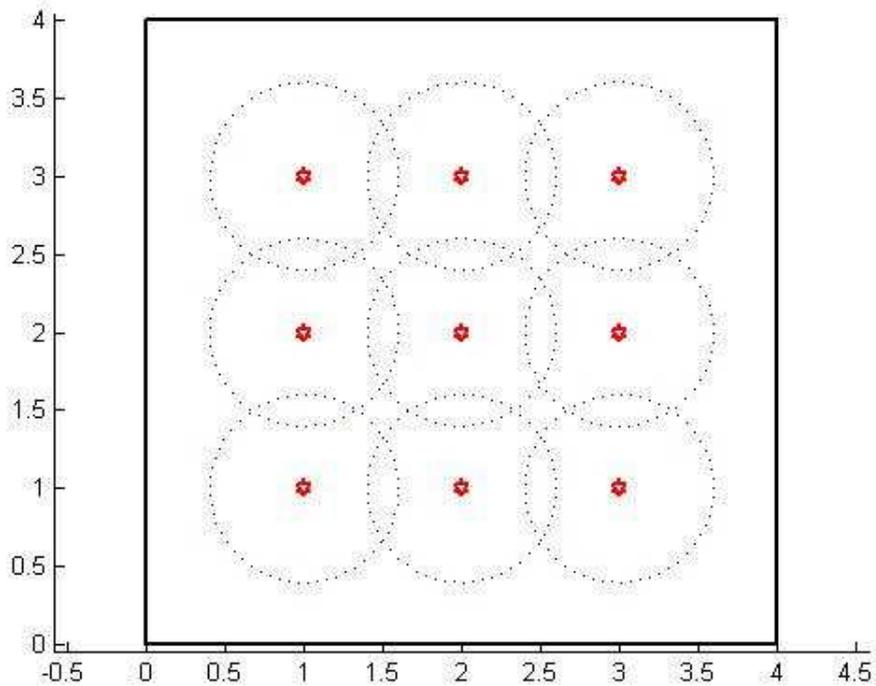


Figura 2.1: Esempio di stanza con 3 tags per lato e relative regioni (tratteggiate) per un raggio d'antenna di 60 cm

### 2.3 Il filtraggio

Per fondere insieme le misurazioni provenienti dall'odometria e dagli RFID si usano dei filtri matematici, tra cui il filtro di Kalman esteso e il filtro particellare. Questi filtri prendono in ingresso le varie misure per poi fornire in uscita una stima che il robot usa per localizzare la sua posizione nella stanza.

Il filtro di Kalman è un modello che fornisce una stima ottima nel caso di sistemi lineari perturbati dai disturbi gaussiani. Come si nota dalle (2.1) - (2.2) il nostro sistema è non lineare, e quindi si usa la versione "adattata" del filtro, definita filtro di Kalman esteso (EKF), che tuttavia non fornisce più una stima ottima. Il difetto del filtro di Kalman, e della sua versione estesa, è che necessita di segnali con probabilità di distribuzione gaussiana in ingresso, e quindi bisogna modificare i dati provenienti dagli RFID per

adattarli a questo filtro (le misure odometriche sono già gaussiane, come mostrato nella Sezione 4.3).

Il filtro particellare, presentato nel Capitolo 4, invece non necessita di ingressi particolari, tuttavia richiede un tempo di esecuzione maggiore rispetto all'EKF.

## Capitolo 3

# Sistemi RFID

RFID è l'acronimo di Radio Frequency Identification che sta ad indicare quindi un tipo di tecnologia che sfrutta segnali d'identificazione a radio frequenza (3Hz - 300GHz). È un tipo di tecnologia che si sta sviluppando molto negli ultimi anni perché presenta vantaggi notevoli rispetto a tecnologie precedenti. Gli RFID sono utilizzati su vasta scala in vari ambienti di lavoro. Esempi di RFID sono:

- Telepass
- Chip di riconoscimento impiantati negli animali
- Chip antitaccheggio

### 3.1 Gli RFID nella localizzazione

I sistemi RFID si basano sull'identificazione di entità fisiche che sono munite di transponder, gli RFID tags. Questi comunicano tramite antenna con l'RFID reader, ovvero l'unità in grado di leggere il tag. Grazie ad un'antenna ad esso collegata via cavo, il reader emette un campo elettromagnetico ad una determinata frequenza. La presenza del tag viene rilevata come perturbazione da parte del reader in quanto il campo elettromagnetico attiva il circuito del tag che modula il segnale e lo rinvia. Quando l'antenna capta tale segnale, il reader provvede a tradurlo in una sequenza di codice binario. Nel caso in cui, invece, non venga rilevato alcun tag, il reader rileva comunque il segnale riflesso ma privo di perturbazioni.

### 3.1.1 RFID Tag o transponder

I tags sono delle etichette di dimensioni ridotte (in genere 2-3 cm per lato).

Esistono vari tipi di tags più o meno complessi in base ai componenti utilizzati. Tutti sono costituiti da:

- un'antenna;
- un microcontrollore;
- una memoria;
- un involucro protettivo;
- una batteria (solo tags attivi);

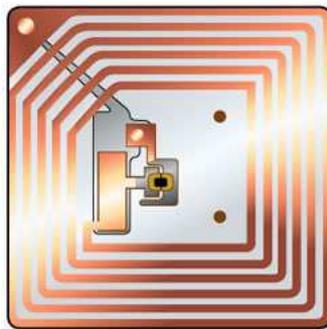


Figura 3.1: RFID Tag

L'antenna è un avvolgimento a spirale che, quando investita dal campo elettromagnetico del reader, genera una tensione ai suoi capi creando quindi un fenomeno d'accoppiamento con l'antenna del reader. Tale tensione attiva il tag dopo essere stata rettificata per mezzo di un apposito circuito raddrizzatore. A questo punto il transponder restituisce il segnale al reader modulandolo in ampiezza tramite una resistenza di carico in parallelo con l'antenna.

L'accoppiamento tra le antenne del reader e del tag può essere di vario tipo (*induttivo*, *backscatter*, *chiuso* o *elettrico*)<sup>1</sup> in base alle esigenze di frequenza (e quindi di distanza) e al tipo di alimentazione del tag. In particolare possiamo distinguere tra tags:

---

<sup>1</sup>Vedi [2]

- *passivi*: la tensione per accoppiamento è sufficiente ad alimentare il chip. Ovviamente all'aumentare della distanza il funzionamento del tag diminuisce. Tuttavia questi tag hanno una durata molto elevata e dimensioni e costi contenuti;
- *attivi*: sfruttano una batteria per alimentarsi, funzionando quindi sia da antenna ricevente che da trasmettente anche a diversi chilometri di distanza, a scapito di durata e costo.

La memoria dei tags può essere Read Only o Read/Write. Nel primo caso le viene impresso un identificativo al momento della produzione che non può essere cambiato. Nel caso Read/Write, invece, tale codice può essere modificato tramite il reader.

### 3.1.2 RFID Reader

Il reader è l'unità in grado di interagire con il transponder. Esso è munito di un'antenna con la quale invia e riceve il segnale, mentre l'unità di controllo lo analizza. I readers possono essere sia portatili che installabili su di un mezzo. È proprio quest'ultimo il caso preso in esame, in cui il reader è installato nel robot navigante.



Figura 3.2: RFID Reader aperto

## 3.2 Vantaggi degli RFID

Tra i vantaggi degli RFID, anche rispetto a tecnologie precedenti, si possono annoverare:

- nel caso di tags Read/Write i dati in esso contenuti possono essere aggiornati facilmente;
- non è necessaria la lettura a contatto (a differenza delle bande magnetiche);
- il tag non deve essere necessariamente visibile per la lettura (a differenza del codice a barre);
- lettura e scrittura possono essere effettuate anche in movimento;
- lettura di più tag contemporaneamente;
- prezzo accessibile.

### 3.3 Difetti degli RFID

Tra i fattori che ne limitano l'utilizzo troviamo:

- l'inquinamento elettromagnetico;
- sensibilità alla presenza di oggetti metallici;
- temperature critiche.

La presenza di fonti radio, piccoli elettrodomestici o altri transponders interferiscono, con i loro campi, sulla comunicazione tra reader e tag. Allo stesso modo gli oggetti metallici interferiscono sulla ricezione dell'antenna. Inoltre un tag in un recipiente di metallo non è rilevabile per effetto del fenomeno noto come *Gabbia di Faraday*<sup>2</sup>. In genere il funzionamento di un chip ha un range di temperatura piuttosto limitato e quindi questa tecnologia non può essere utilizzata a temperature troppo alte (oltre i 200°C) né troppo basse (al di sotto dei -80°C). Per quanto riguarda i costi di produzione bisogna dire che i tags hanno prezzi che variano da qualche centesimo a qualche decimo di euro. Per lo studio della localizzazione, quindi, il prezzo risulta più che accettabile, mentre se si vuole utilizzare questa tecnologia su larga scala, ad esempio per sostituire tecnologie più obsolete (vedi codice a barre e bande elettromagnetiche) i costi possono diventare eccessivamente alti.

---

<sup>2</sup>Questo principio afferma che in un conduttore le cariche elettriche tendono a disporsi all'esterno rendendo nullo il campo elettrico all'interno della "gabbia".

L'antenna, invece, ha un prezzo decisamente più elevato (intorno ai 500 euro), ma va sottolineato che l'antenna è una spesa da effettuare una sola volta e che può essere riutilizzata per altri esperimenti di localizzazione tramite RFID.

### 3.4 Tipologie d'uso degli RFID

Gli RFID sono una tecnologia recente di largo impiego. Un esempio dei diversi approcci a questa tecnologia nell'ambito della localizzazione è fornito da [5], in cui viene mostrato come gli RFID possano essere utilizzati essenzialmente in due modi: il primo approccio consiste nel disporre vari tags all'interno dell'ambiente (vedi [7]) che vengono rilevati quando il robot passa prossimità di esse; il secondo approccio consiste nell'utilizzo di un *reader* particolare in grado di determinare la distanza dal tag basandosi sul tempo di risposta (vedi [8]). Il primo metodo è quello utilizzato in questa trattazione e risulta decisamente più economico del secondo, necessitando di antenna e tags standard. Inoltre fornisce una buona stima della posizione quando abbinata all'odometria. Il secondo metodo, invece, permette una stima migliore dell'altro a fronte di una spesa maggiore soprattutto per l'*RFID reader*.

## Capitolo 4

# Filtro Particellare

### 4.1 Introduzione al PF

Il filtro particellare è un tipo di filtro bayesiano che sfrutta in maniera ricorsiva simulazioni Monte Carlo per calcolare la distribuzione di probabilità. I filtri bayesiani sono strumenti matematici che hanno come obiettivo quello di stimare lo stato di un sistema dinamico mediante misure sensoriali. Nel caso della localizzazione di un robot mobile, il sistema dinamico è rappresentato dall'insieme robot-ambiente, lo stato è la posizione del robot (composto da due coordinate cartesiane  $[x, y]$  che indicano la posizione nello spazio di stato e una coordinata angolare  $[\theta]$  che indica l'orientamento), e le misure possono includere letture odometriche e sensoriali. Inoltre i filtri bayesiani assumono che l'ambiente sia Markoviano, cioè i dati passati e futuri sono considerati indipendenti fra loro se è noto lo stato corrente. L'idea chiave della teoria bayesiana è determinare la funzione densità di probabilità a posteriori dello stato basata su tutte le informazioni disponibili (ovvero misure odometriche e sensori) per poi stimare lo stato del sistema. Tuttavia i filtri bayesiani sono di difficile implementazione, e per questo si ricorre ai metodi Monte Carlo.

I metodi Monte Carlo furono ideati negli anni '40 da *John von Neumann* e *Stanislaw Marcin Ulam* prendendo il nome del celebre casinò. Esistono varie tecniche tutte basate sul campionamento statistico e sulla stima finalizzate a determinare la soluzione del problema proposto. Il concetto di base è quello di approssimare un problema di integrazione continua con uno di integrazione discreta sfruttando appunto il campionamento. Scelti dei campioni casuali (nel nostro caso le particelle) la posizione di stima si ottiene come

somma delle probabilità di ogni campione, normalizzata al numero di campioni utilizzati (vedi Sezione 4.7). Per maggiori delucidazioni sugli approcci a modelli non lineari con segnali non gaussiani si rimanda a [11].

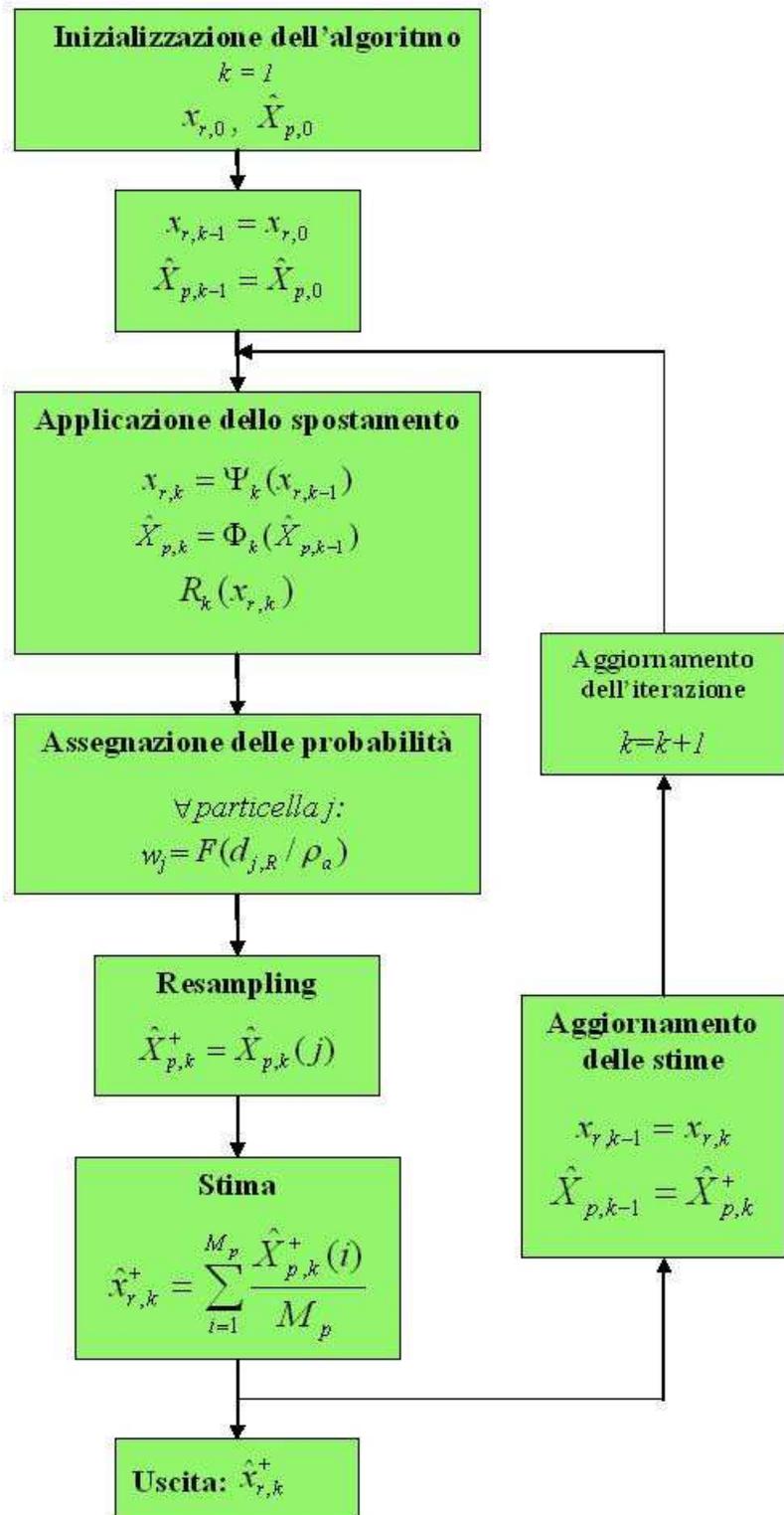
## 4.2 L'algoritmo del PF

Il filtro particellare (PF) si basa sulla ripetizione di una serie di operazioni ripetute ciclicamente. Inizialmente il filtro stima la posizione di partenza grazie agli RFID tags visti. Crea quindi un certo numero di “particelle” ovvero delle entità virtuali con diverse posizioni e orientamenti all’interno della stanza. A questo punto comincia il ciclo iterativo. Applicando alle particelle gli stessi movimenti del robot secondo la (2.2) le particelle subiranno gli stessi errori cui è soggetto il robot, in più avranno anche delle traiettorie diverse a causa dell’orientamento casuale. A questo punto tramite gli RFID viene individuata una regione di appartenenza della posizione del robot che permette di assegnare ad ogni particella una probabilità a seconda che cada o meno nella regione individuata. Le particelle con probabilità più alta vengono clonate per creare il nuovo gruppo di particelle da utilizzare all’iterazione successiva e vengono utilizzate per stimare la nuova posizione di partenza. Grazie a questo metodo, ad ogni ripetizione la posizione stimata sarà sempre più vicina a quella vera; tuttavia, dovendo ripetere ognuna di queste operazioni per ogni particella, il cui numero è dell’ordine delle migliaia, il filtro può risultare oneroso dal punto di vista dell’efficienza computazionale.

Viene di seguito presentato l’algoritmo che descrive il funzionamento del PF. Il simbolo “ $\hat{\cdot}$ ” sta ad indicare una stima che, se abbinata all’apice “ $+$ ”, indica la stima a posteriori.

## 4.2.1 Schema dell'algoritmo

Algorithm 4.1 Algoritmo del filtro particellare



### 4.2.2 Legenda dell'algoritmo

In merito all'algoritmo 4.2.1 viene di seguito proposta la legenda:

- $x_{r,k}$  è la posizione del robot al tempo  $k$ , in particolare  $x_{r,0}$  è la posizione di partenza;
- $\hat{X}_{p,k}$  è la matrice le cui colonne indicano la posizione di ogni particella al tempo  $k$  in particolare  $\hat{X}_{p,0}$  indica la posizione di partenza di ogni particella che viene assegnata casualmente;
- $\psi_k(x_{r,k-1})$  è l'applicazione della (2.2) alla posizione del robot in modo da calcolare il movimento al tempo  $k$ ;
- $\phi_k(\hat{X}_{p,k-1})$  è l'applicazione del moto ad ogni particella (vedi più avanti Sezione 4.4);
- $R_k(x_{r,k})$  è la regione RFID letta al passo  $k$ -esimo in base alla posizione del robot  $x_{r,k}$ ;
- $w_j$  è la probabilità (peso) assegnata alla  $j$ -esima particella tramite la funzione  $F$ ;
- $d_{j,R}$  è la distanza tra la  $j$ -esima particella e il centro della regione  $R$ ;
- $\rho_a$  è il raggio dell'antenna;
- $\hat{X}_{p,k}^+$  è l'insieme di particelle da utilizzare al passo successivo;
- $\hat{x}_{r,k}^+$  è la stima della posizione del robot successiva al resampling;
- $M_p$  è il numero di particelle utilizzate.

### 4.3 Inizializzazione

La posizione iniziale del robot viene stimata in base alla regione vista. Per individuare una generica regione RFID consideriamo innanzitutto il robot nella posizione  $x_r$ . Possiamo quindi costruire una stringa  $r_{RFID} = \{tag_{i_1}, \dots, tag_{i_n}\}$  di tutti i tags visibili dalla posizione  $x_r$ . La probabilità di vedere la stringa  $r_{RFID}$  da questa posizione è data dalla  $prob(r_{RFID} | x_r)$  a

posteriori (vedi [6]). D'altro canto la probabilità opposta, cioè che vedendo  $r_{RFID}$  il robot sia in  $x_r$ , è data da  $prob(x_r | r_{RFID})$ , secondo teoria<sup>1</sup>, ed ha una distribuzione uniforme. Per questo motivo il segnale proveniente dagli RFID deve essere modificato per potervi applicare l'EKF, dato che questo filtro necessita di segnali gaussiani in ingresso agendo su media e su covarianza.

La regione  $R_k$  è definita da tutte quelle posizioni  $x_r$  che vedono la stringa  $r_{RFID}$ , ovvero la  $prob(r_{RFID} | x_r) = 1$  se  $x_r \in R_k$ , 0 altrimenti.

### 4.3.1 Discretizzazione

Per superare il problema della ricezione degli RFID tags nel caso virtuale, nella simulazione la stanza è stata suddivisa primariamente in un numero prefissato di celle tutte delle stesse dimensioni. Ad ogni cella è stata poi abbinata una stringa di tags del tipo  $r_{RFID} = \{tag_{i_1}, \dots, tag_{i_n}\}$  in cui l'elemento  $j$ -esimo vale 1 nel caso di visibilità, 0 altrimenti. La visibilità di un tag è determinata dalla distanza tra il robot e il tag stesso: il tag risulta visibile nel caso in cui tale distanza risulti minore o uguale al raggio dell'antenna  $\rho_a$ . Applicando lo stesso ragionamento alla regione  $R_k$  se  $d_{j,R} \leq \rho_a$  (con  $d_{j,R}$  definito come sopra) allora il robot starà attraversando questa regione.

### 4.3.2 Percorso

Per motivi pratici nella simulazione è stato fatto seguire al robot un percorso prestabilito, rappresentato in Figura 4.3.2. Su questo percorso sono state fatte le varie simulazioni con i conseguenti risultati mostrati nel Capitolo 5.

---

<sup>1</sup>Vedi [1]

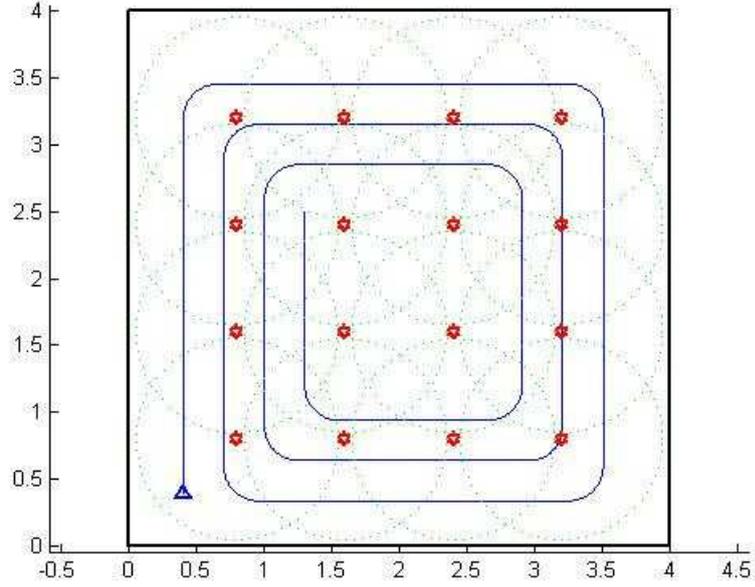


Figura 4.1: Percorso del robot utilizzato per la simulazione

#### 4.4 Spostamento delle particelle

Il movimento delle particelle è dettato dalla (2.2), dinamica del robot, con l'aggiunta di un certo errore sulle letture odometriche diverso rispetto a quello del robot, non essendo noto tale errore. In questo modo gli spostamenti delle particelle saranno casuali e in generale differenti rispetto a quelli del robot stesso. La dinamica delle particelle è perciò descritta da:

$$\phi_{k+1}(\hat{X}_{p,k}) := \begin{cases} \hat{X}_{k+1} = \hat{X}_k + \frac{U_{R,k}^p + U_{L,k}^p}{2} \cos(\hat{\Theta}_k) \\ \hat{Y}_{k+1} = \hat{Y}_k + \frac{U_{R,k}^p + U_{L,k}^p}{2} \sin(\hat{\Theta}_k) \\ \hat{\Theta}_{k+1} = \hat{\Theta}_k + \frac{U_{R,k}^p - U_{L,k}^p}{\Delta} \end{cases} \quad (4.1)$$

dove  $\Delta$  è la distanza tra le ruote del robot<sup>2</sup>,  $\hat{X}_{p,k} := [\hat{X}_k; \hat{Y}_k; \hat{\Theta}_k]^T$  è la terna di vettori che descrivono rispettivamente ascissa, ordinata e orientamento di ogni particella, mentre  $U_{R,k}^p$  e  $U_{L,k}^p$  sono le matrici contenenti gli spostamenti delle due ruote calcolati per ogni particella e così definiti:

<sup>2</sup>I valori delle costanti sono definite in ambito simulativo nel Capitolo 5

$$U_{R,k}^p = U_{R,k}^e + N_{R,k}^p$$

$$U_{L,k}^p = U_{L,k}^e + N_{L,k}^p$$

in cui:

- $U_k^e$  è una matrice con ogni elemento pari a  $u_k^e$  definito nella Sezione 2.1;
- $N_k^p \sim \mathfrak{N}(0, K \cdot |U_k^e|)$  matrice di cui ogni colonna contiene un vettore degli errori odometrici definito come nella Sezione 2.1.

## 4.5 Assegnazione delle probabilità

Ad ogni particella viene assegnata una certa probabilità legata alla distanza della particella dal centro della regione vista ( $d_{j,R}$ ). La funzione  $F$  di probabilità per la  $j$ -esima particella vale:

$$F_j(d_{j,R}/\rho_a) := \begin{cases} \frac{1}{2} E \left[ \left( \frac{d_{j,R}}{\rho_a} - 1 \right) \frac{\lambda}{\sqrt{2}} \right] & \Leftrightarrow r_{RFID}(tag_j) = 1 \\ 1 - \frac{1}{2} E \left[ \left( \frac{d_{j,R}}{\rho_a} - 1 \right) \frac{\lambda}{\sqrt{2}} \right] & \Leftrightarrow r_{RFID}(tag_j) = 0 \end{cases}$$

dove  $E[\star]$  è la funzione d'errore complementare<sup>3</sup> e  $\lambda$  è il disturbo di lettura quantificato.

## 4.6 Resampling

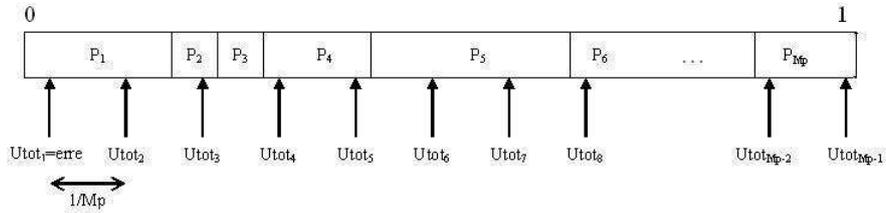
Il resampling, o ricampionamento, è la fase in cui viene ricreato il gruppo di particelle da utilizzare al passo successivo. Il metodo di scelta delle particelle può essere effettuato in vari modi sempre secondo il criterio di clonare le particelle con peso (i.e. probabilità) maggiore. Il metodo più intuitivo sarebbe quello di ordinare il vettore  $\hat{X}_{p,k}$  e, fissato un peso minimo  $w_{min}$ , creare tante copie di ogni particella  $j$  con peso  $w_j > w_{min}$  in proporzione al suo peso. Questo metodo ha una buona efficienza computazionale<sup>4</sup> ma risulta

<sup>3</sup>La generica  $E[x]$  è definita come:  $\frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt$

<sup>4</sup>Gli algoritmi di ordinamento più veloci (*mergesort*, *bubblesort*...) hanno un complessità che è dell'ordine di  $O(n \log n)$  (vedi [9])

poco efficace: non è semplice determinare un valore  $w_{min}$  soddisfacente col rischio di sceglierlo troppo elevato (verrebbero selezionate poche particelle) o troppo basso (troppe particelle scelte).

Un metodo che supera questi problemi è quello di seguito mostrato. Considerando il vettore  $\hat{X}_{p,k}$  normalizzato al numero di particelle possiamo immaginare il vettore come un segmento di lunghezza da 0 a 1. Oni particella occuperà una parte di questo segmento in proporzione al suo peso, come mostrato nella seguente figura:



Preso poi un valore a caso tra  $0 \div \frac{1}{M_p}$  (*erre* in figura) ad ogni iterazione ci si sposta lungo il segmento tramite il puntatore  $U_{tot}$  che a partire da *erre* somma il valore  $1/M_p$  per  $(M_p - 1)$  volte. Ogni volta che una particella viene toccata da  $U_{tot}$  una sua copia verrà inserita nella matrice  $\hat{X}_{p,k}^+$ . Di conseguenza le particelle con pesi maggiori saranno quelle più facilmente individuabili (perchè più vicine alla posizione vera del robot) e andranno a formare il nuovo gruppo.

## 4.7 Stima della posizione

A seguito del *resampling* viene creata la matrice  $\hat{X}_{p,k}^+$ , contenente la posizione di partenza di ogni particella del nuovo gruppo. A partire da questo vettore viene fatta la stima  $\hat{x}_{r,k}^+$  della posizione del robot dopo lo spostamento. Il valore di tale stima viene calcolato col metodo Monte Carlo, ed è dato dalla somma di tutte le particelle normalizzata al numero di particelle utilizzate  $M_p$ :

$$\hat{x}_{r,k}^+ = \sum_{i=1}^{M_p} \frac{\hat{X}_{p,k}^+(i)}{M_p}$$

## 4.8 Varianti del PF

A causa della complessità computazionale a livello temporale, il filtro particellare non può essere utilizzato su larga scala. Per ovviare a questo problema vengono a volte combinati più filtri di tipo diverso. Ad esempio in [10] vengono utilizzati sia il PF che l'EKF. La precisione del primo viene utilizzata per calcolare alcune variabili non facilmente approssimabili a modelli gaussiani. L'EKF, invece viene sfruttato grazie alla sua maggiore velocità per calcolare la maggior parte delle variabili o per rendere meno rumorose alcune variabili quali velocità e accelerazione.

Altri esempi di applicazioni del filtro particellare sono contenuti in [11] in cui vengono messi a confronto vari tipi di filtri, tra cui anche l'EKF, applicati a disturbi non lineari e non gaussiani.

## Capitolo 5

# Risultati simulativi

Per testare l'efficacia del filtro particellare sono state effettuate delle simulazioni mediante computer<sup>1</sup> in ambiente MATLAB.

### 5.1 Errore di stima

Per valutare gli effetti del filtro particellare ci si è avvalsi sia di un metodo grafico che di uno matematico. Graficamente, è stato sovrapposto al percorso reale compiuto dal robot (vedi Figura 4.3.2) il percorso tracciato secondo la stima  $\hat{x}_{r,k}$ , calcolata nella Sezione 4.7. Per maggiore chiarezza la stima è stata tracciata in rosso per la prima metà del percorso, in verde nella seconda metà.

Il metodo matematico consiste nel calcolo di un coefficiente d'errore  $J$  così definito:

$$J := \frac{1}{nPassi} \sum_{k=1}^{nPassi} \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \quad (5.1)$$

dove  $(x_k; y_k)$  rappresentano le coordinate del robot vere mentre  $(\hat{x}_k; \hat{y}_k)$  quelle stimate. Dalla (5.1) si vede che  $J$  non rappresenta altro che la media dello scostamento della stima, espresso in cm, rispetto alla posizione reale durante tutto il tragitto: minore è il valore di  $J$ , migliore è la stima effettuata e quindi migliore è il funzionamento del metodo.

---

<sup>1</sup>L'implementazione è stata effettuata su processore *Intel*<sup>®</sup> *Pentium*<sup>®</sup> M 760 (2.0 GHz, 533 MHz FSB, 2 MB L2 cache) e RAM da 1 GB

## 5.2 Simulazione ( $M_p=1000$ )

Vengono di seguito riportati i dati utilizzati nella simulazione principale che fa riferimento ad un robot che si muove come in Figura 4.3.2 e con un PF da 1000 particelle.

### 5.2.1 Ambiente e tags

L'ambiente è costituito da una stanza  $4m \times 4m$  sul cui soffitto sono stati disposti 16 tags, 4 per riga, equidistanti tra loro, così da formare un quadrato. L'ampiezza del raggio dell'antenna del robot  $\rho_a$  è stato fissato a 75 cm. Il numero di celle è stato fissato a 100 per lato.

### 5.2.2 Il disturbo di lettura $\lambda$

Il disturbo di lettura rappresenta l'interferenza sui dati letti dall'RFID *reader*. Il valore di  $\lambda \in [0; 1)$  deve essere specificato per il funzionamento del programma. Nel nostro caso è stato posto  $\lambda = 0$ , cioè assenza di disturbo. È stato sperimentalmente provato, tuttavia, che il funzionamento del programma migliora notevolmente inserendo un disturbo minimo, ma non nullo. Il motivo è legato al fatto che per  $\lambda = 0$  ad una particella in prossimità della regione  $R_{tag}$  viene assegnato un peso nullo e considerata, quindi, alla stregua di una particella dall'altra parte della stanza. Prendendo invece  $\lambda > 0$  viene inserita un'incertezza sui confini della regione cosicché una particella vicina a  $R_{tag}$  avrà un peso, per quanto minimo, comunque  $\neq 0$ . Per questo motivo all'interno dell'algoritmo del PF nel caso il disturbo lettura sia nullo, viene sostituito con  $\lambda_{modello} = 0.005$ .

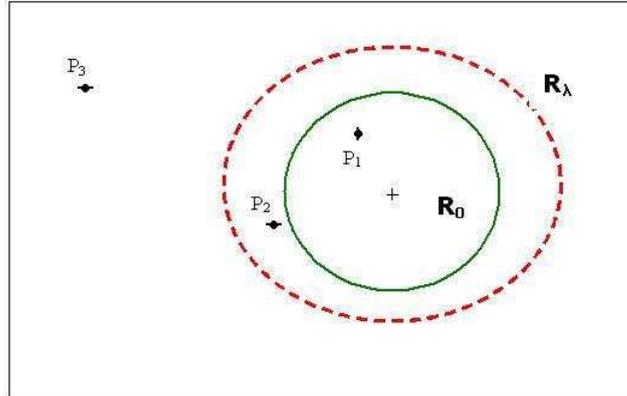


Figura 5.1: Regione tag per  $\lambda = 0$  (regione in verde) e  $\lambda > 0$  (regione tratteggiata in rosso). Se  $\lambda = 0$ ,  $P_2$  e  $P_3$  hanno lo stesso peso (0) nonostante  $P_2$  sia più vicina alla regione. Per  $\lambda > 0$ , a  $P_2$  viene assegnato un peso  $w_2 > 0$ .  $P_1$  ha sempre peso  $> 0$ .

### 5.2.3 Il robot

Trattandosi di una simulazione l'unica misura fisica che ci si è preoccupati di fissare è la distanza tra le due ruote:  $\Delta = 10$  cm. Il numero di passi encoder letti per compiere il tragitto è  $nPassi = 3000$ .

Le costanti degli errori odometrici delle ruote  $K_R$  e  $K_L$  sono state poste pari a  $1 \times 10^{-7}$  m, mentre le ruote sono state considerate perfette ponendo  $\delta_R$  e  $\delta_L$  pari ad 1.

### 5.2.4 L'indice d'errore J

Nel programma simulativo vengono calcolati due coefficienti d'errore, uno relativo all'intero percorso ( $J_1$ ), la alla sola seconda parte ( $J_2$ ). La stima relativa alla Figura 5.2.4 ha un errore  $J_1 = 3.1801$ ; si nota come il PF sia particolarmente efficace dalla seconda metà in poi per la quale  $J_2 = 1.5311$ . Come illustrato nei paragrafi precedenti, infatti, ad ogni passo vengono selezionate le particelle più vicine alla posizione reale. Inizialmente le particelle sono sparse per tutta la stanza, quindi la stima è molto approssimativa. Andando avanti lungo il tragitto, invece, l'errore si riduce fino a che la posizione stimata va quasi a coincidere con quella reale.

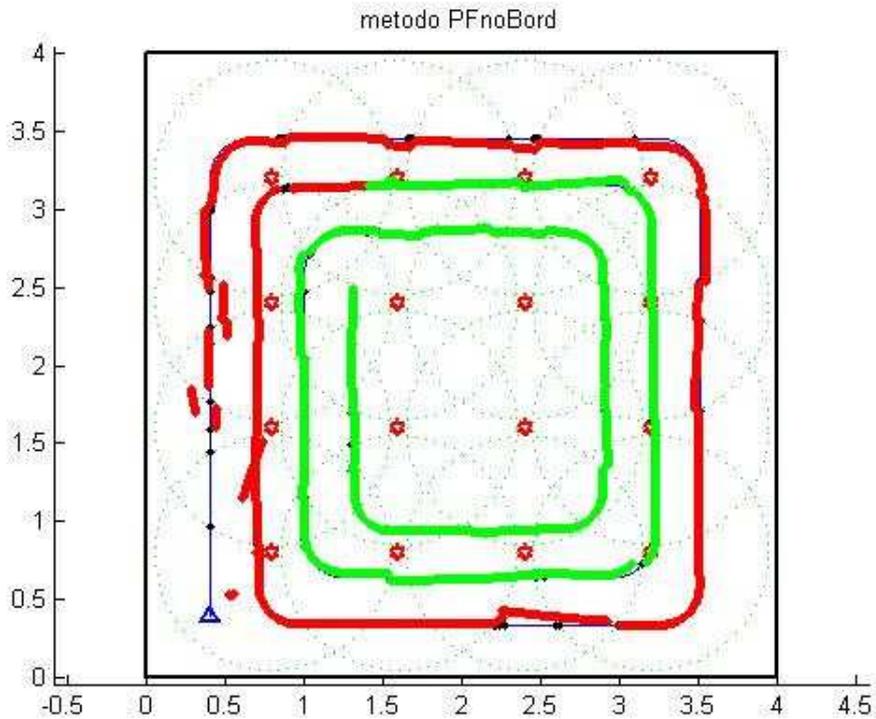


Figura 5.2: Stima tramite PF. L'errore su tutto il percorso ha un indice d'errore  $J_1 = 3.1801$ ; per la seconda metà (in verde)  $J_2 = 1.5311$ .

### 5.3 Simulazione al variare di $M_p$

Vengono ora mostrati graficamente i risultati delle simulazioni al variare del numero di particelle. Il modello preso in considerazione è il medesimo della Sezione 5.2 con la differenza di far variare  $M_p$  tra  $1000 \div 5000$ . Nella prima sottosezione si mostrerà l'andamento degli indici d'errore  $J_1$  e  $J_2$  all'aumentare di  $M_p$ , nella seconda il tempo impiegato.

#### 5.3.1 Grafico degli errori $J_1$ e $J_2$ al variare di $M_p$

Il seguente grafico mostra l'andamento dell'errore al variare del numero di particelle. I valori di  $J_1$  e  $J_2$  risultano praticamente costanti a parte qualche picco legato agli errori casuali introdotti nelle varie misure. Un numero di particelle superiore alle migliaia non influenza particolarmente, quindi, l'efficacia del metodo.

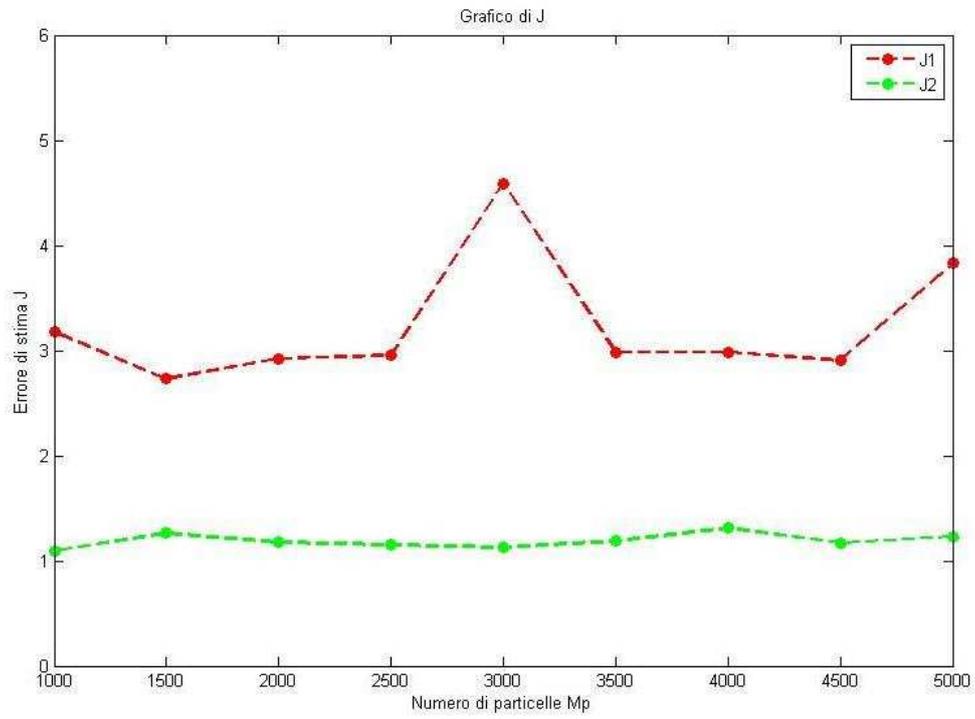


Figura 5.3: Andamento degli errori di stima  $J_1$  e  $J_2$  per  $M_p = 1000 \div 5000$

$M_p < 1000$

Per  $M_p$  inferiore a 1000 la stima effettuata dipende molto dalle pose date alle singole particelle. L'errore totale sarà basso se casualmente le particelle finiranno vicine alla regione (nel qual caso si ottengono paradossalmente anche buoni risultati), altrimenti si raggiungono valori molto alti di  $J$ . Il numero insufficiente di particelle fa sì che la stima sia eccessivamente frutto del caso, mentre per  $M_p$  alto la stima è dettata dalle particelle stesse, che riescono a battere la casualità con cui sono state create.

Di seguito viene riportato uno dei grafici creati dalle simulazioni e che mostra palesemente l'aleatorietà della stima calcolata.

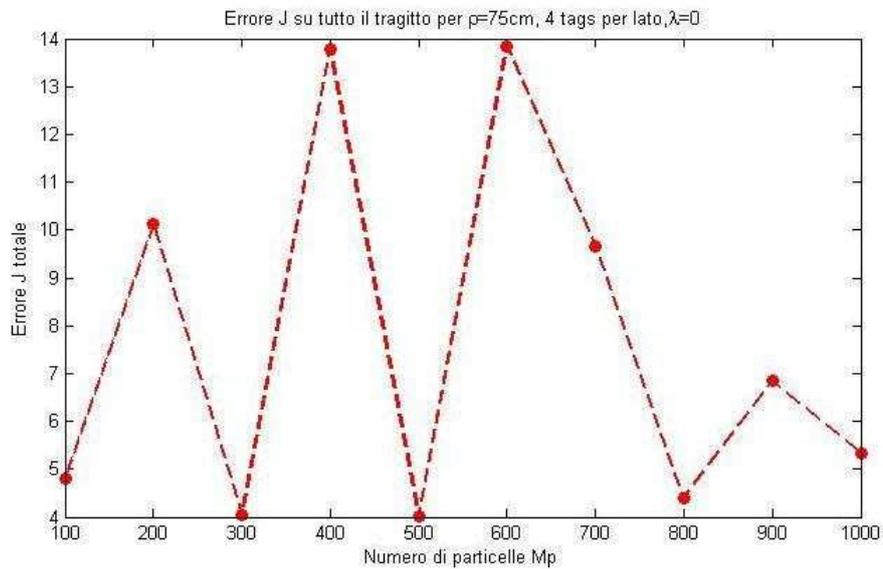


Figura 5.4: Errore totale  $J_1$  per valori di  $M_p < 1000$

### 5.3.2 Complessità computazionale al variare di $M_p$

Discorso diverso va fatto invece per il tempo impiegato dall'algoritmo. Come mostra il grafico in Figura 5.3.2, il tempo impiegato dall'algoritmo per 5000 particelle è praticamente cinque volte quello impiegato per 1000: c'è un rapporto costante di proporzionalità diretta per cui  $T \propto M_p$ .

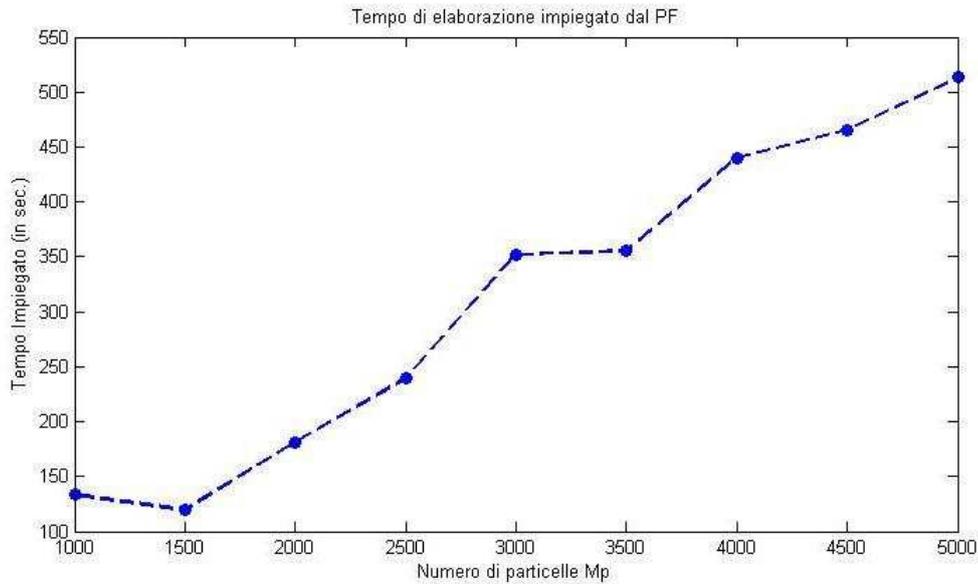


Figura 5.5: Grafico del tempo in secondi impiegato dal PF per l'elaborazione della stima per  $M_p = 1000 \div 5000$

### 5.4 Simulazione al variare del numero di tag e $\rho_a$

Vengono di seguito proposte delle simulazioni che calcolano l'indice d'errore  $J$  al variare del raggio dell'antenna  $\rho_a$  per valori compresi tra i 50 e i 100 cm. Le simulazioni sono state fatte, inoltre, variando il numero di tag in modo che, disponendoli sempre a riempire un quadrato, risultino 3, 4 o 5 per ogni lato. Per tutto il resto del capitolo con  $n_{tags}$  si intenderà il numero di tag per ogni lato del quadrato ( il numero totale di tags è ovviamente  $n_{tags}^2$ ). Le caratteristiche del modello utilizzato sono le stesse descritte nella Sezione 5.2 (con  $M_p = 1000$  e  $\lambda = 0.2$ ), a parte il numero di tags e il raggio dell'antenna.

<sup>2</sup>Vedi Sottosezione 5.2.2

Accanto a questi grafici viene inoltre mostrato l'andamento della deviazione standard ( $\sigma$ ) al variare del raggio dell'antenna con sovrapposta l'area media. La deviazione standard (o scarto quadratico medio) rappresenta la dispersione dei dati attorno al valore atteso  $\bar{x}$ , media aritmetica dei dati. Il valore di  $\sigma$  rispetto agli elementi di un vettore  $x$  con media  $\bar{x}$ , è dato da:

$$\sigma_x = \sqrt{\sum_{i=1}^n \frac{[x(i) - \bar{x}]^2}{n}}$$

Nel nostro caso è interessante osservare come variano le regioni RFID all'aumentare del raggio dell'antenna e del numero di tags. Ad ogni regione di tags viene assegnata una densità di probabilità gaussiana in modo da stimarne la dimensione. Utilizziamo perciò la deviazione standard<sup>3</sup> per dare un'idea sull'ampiezza delle regioni rispetto alla regione media  $\bar{A}_R$ . Il valore di  $\sigma$  ci fornisce una stima sull'omogeneità delle regioni, cioè di quanto la loro superficie si discosta mediamente rispetto ad  $\bar{A}_R$ . Più il valore di  $\sigma$  è basso rispetto all'area media, più le regioni hanno superfici di dimensione uniforme ed  $\bar{A}_R$  ne è una buona stima. Al contrario, se  $\sigma$  è alta, vuol dire che le superfici delle regioni saranno molto eterogenee, cioè alcune molto piccole altre molto più grandi rispetto all'area media. .

---

<sup>3</sup>Si ricorda che la curva a "campana" della funzione di probabilità raccoglie il 68% dei casi nell'intervallo  $[\mu - \sigma; \mu + \sigma]$  con  $\mu$  valore medio, e il 95% nell'intervallo  $[\mu - 2\sigma; \mu + 2\sigma]$

### 5.4.1 $n_{tag} = 3, M_p = 1000, \lambda = 0$

#### Deviazione Standard e Area Media

Per  $\rho_a = 50$ , in cui  $\sigma = 0$  e  $\bar{A} = 484$  (non mostrato in figura). In questo caso, infatti, la distanza tra i tags risulta maggiore del raggio dell'antenna, quindi le regioni non si intersecano.

La Figura 5.4.1 mostra che un'altra  $\sigma$  rilevante si ha per  $\rho_a = 60$ . La deviazione standard suggerisce infatti che la maggior parte delle superfici è contenuta entro una volta e mezzo l'area media. Per altri valori di  $\rho_a$ , invece le regioni assumono dimensioni oltre il doppio rispetto a quella media. Ciò è legato al crearsi di regioni molto piccole, frutto dell'intersezioni delle regioni principali, ovvero delle circonferenze create prendendo ogni tag come centro.

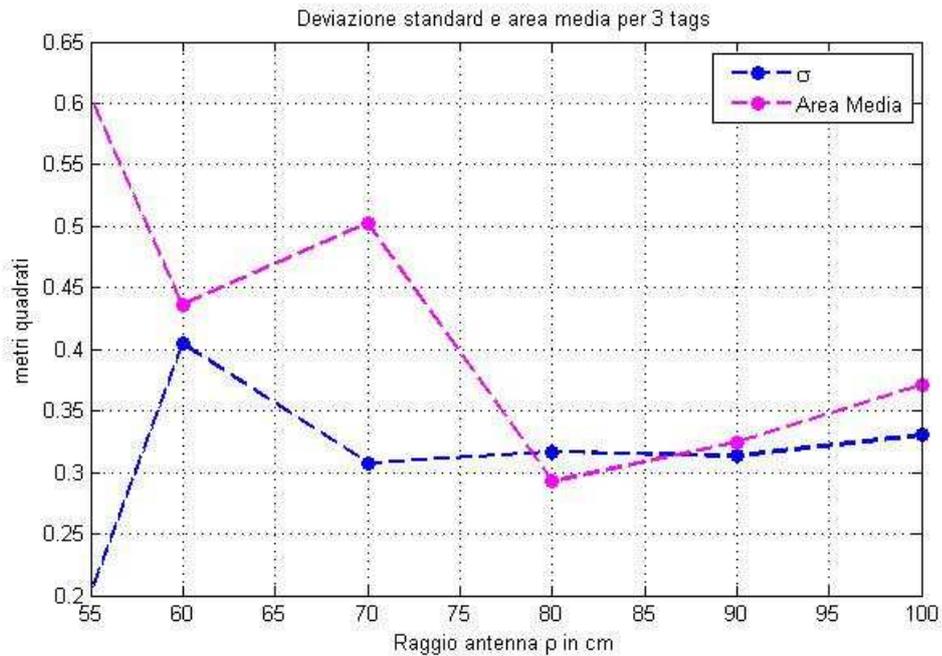


Figura 5.6: Ingrandimento del grafico della deviazione standard e dell'area media per  $n_{tag} = 3$ ; per  $\rho_a = 50$ ,  $\sigma = 0$  e  $\bar{A} = 484$

### Errore J

Dal grafico sottostante si nota come all'aumentare del raggio d'antenna l'errore tende a diminuire. In particolare se ci interessa un errore praticamente basso e costante<sup>4</sup> durante il tragitto conviene scegliere  $\rho_a = 90$ , mentre se vogliamo l'errore minimo alla fine del percorso scegliamo  $\rho_a = 100$ . Confrontando questi valori con la deviazione standard e l'area media notiamo che in entrambi i casi le regioni hanno dimensioni piuttosto variabili (nel secondo superiori al doppio di quella media). Quindi sembra che il filtro funzioni meglio per regioni eterogenee.

È importante notare che **il valore dell'errore è comunque alto** ( $J_1 > 5$ ) e quindi il PF, a queste condizioni non dà risultati soddisfacenti.

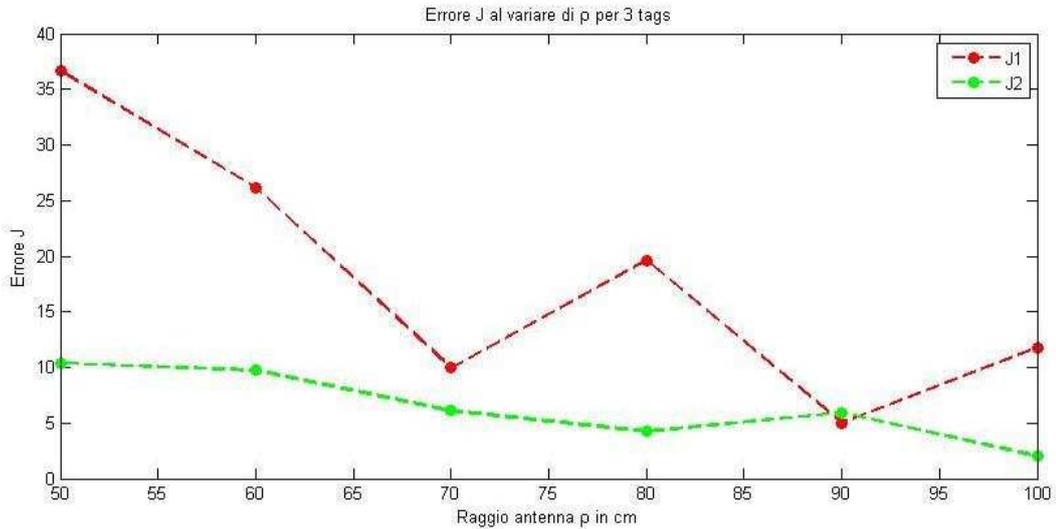


Figura 5.7: Grafico degli errori  $J_1$  e  $J_2$  per  $n_{tag} = 3$

<sup>4</sup>È lecito supporre che, poiché i valori finale e totale dell'errore sono simili, ad ogni passo del percorso l'errore rimanga più o meno costante poiché la stima tende a migliorare ad ogni passo

### 5.4.2 $n_{tag} = 4, M_p = 1000, \lambda = 0$

#### Deviazione Standard e Area Media

Dal grafico sottostante si nota che la maggiore eterogeneità delle regioni si ha al crescere del raggio. Per  $\rho_a = 50$  si ha un valore di  $\sigma < \bar{A}_R$  ma comunque alto che comporterà la presenza di regioni grandi e piccole.

Il secondo picco ( $\rho_a = 90$ ) è legato alla creazione di regioni intersecanti più piccole rispetto a quelle principali. Per gli altri valori la stima dell'area media non è soddisfacente essendo lo scarto quadratico medio addirittura superiore all'area media stessa.

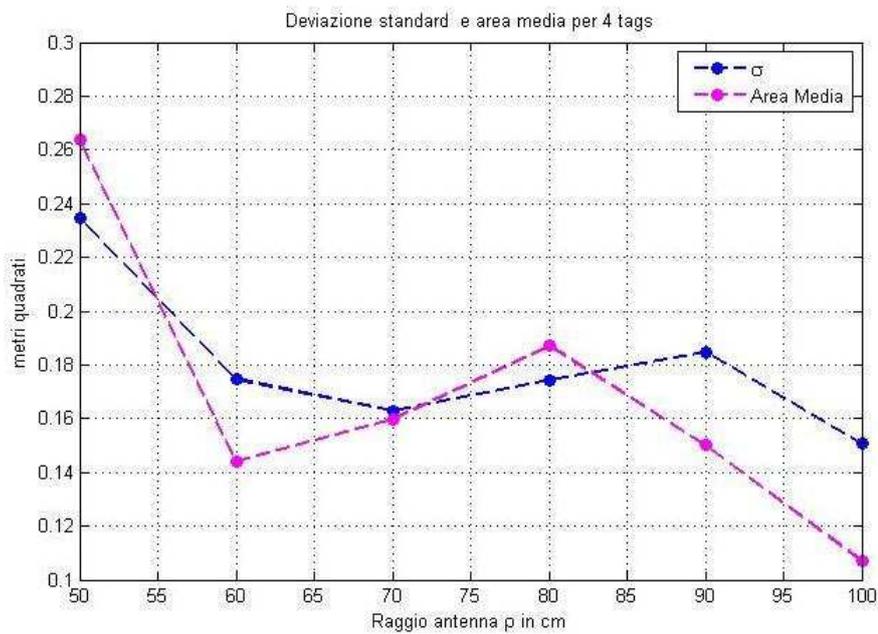


Figura 5.8: Grafico della deviazione standard e dell'area media per  $n_{tag} = 4$

**Errore J**

Dalla figura si nota come l'errore si riduce drasticamente all'aumentare del raggio d'antenna fino a diventare ottimo nel caso di massimo raggio sia per  $J_1$  che per  $J_2$  in cui l'errore su tutto il tragitto si riduce a 2, mentre quello sulla seconda metà raggiunge quasi l'unità. Confrontando con la deviazione standard notiamo che l'errore minimo si ottiene in prossimità di una pessima stima dell'area media, ovvero nel caso di regioni con differenti dimensioni.

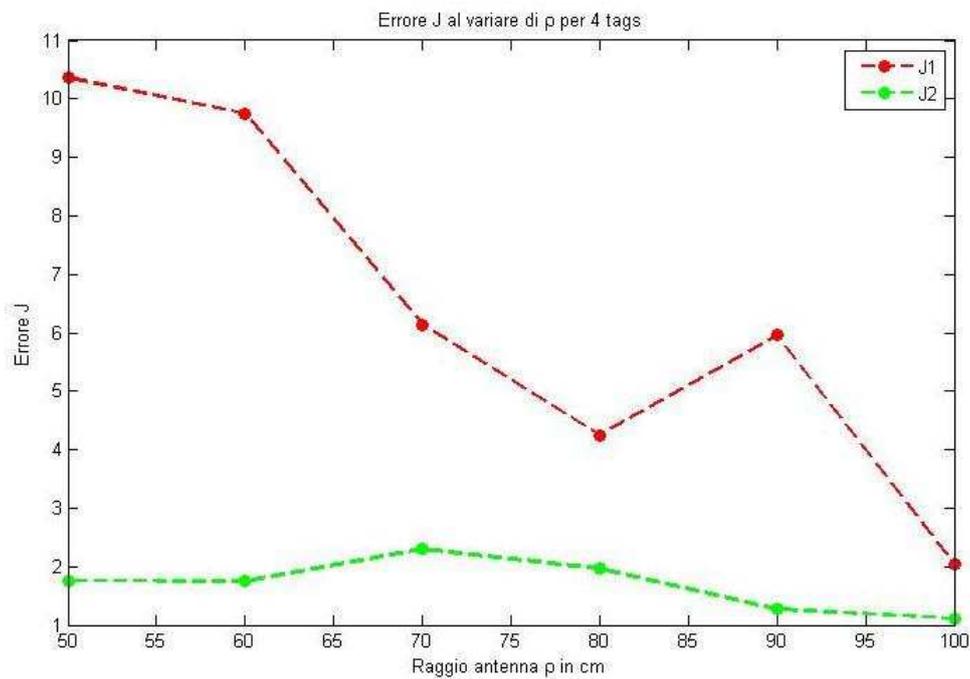
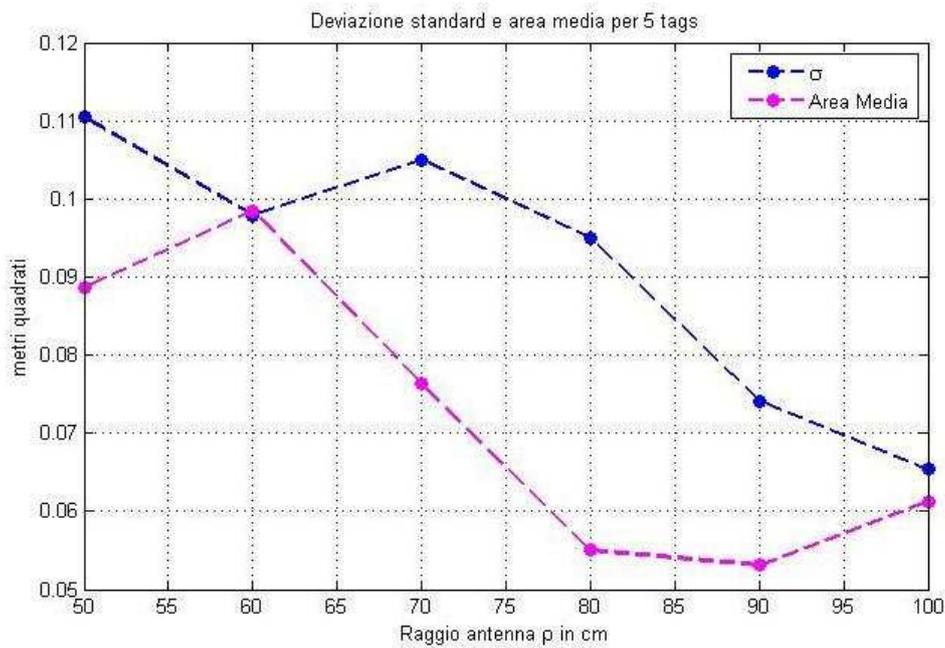


Figura 5.9: Grafico degli errori  $J_1$  e  $J_2$  per  $n_{tag} = 4$

**5.4.3**  $n_{tag} = 5, M_p = 1000, \lambda = 0$ 

**Deviazione Standard e Area Media** La deviazione standard risulta troppo elevata affinché l'area media dia una buona stima della dimensione delle aree (nel caso migliore la deviazione standard ha un valore prossimo all'area media). La massima incertezza si ha per  $\rho_a = 80$ , in cui  $\sigma$  è quasi due volte  $\bar{A}$ .

Figura 5.10: Grafico della deviazione standard e dell'area media per  $n_{tag} = 5$

**Errore J**

In questo caso l'errore finale  $J_2$  risulta sempre molto buono e prossimo all'unità. Come nel caso precedente, inoltre, il punto in cui si hanno valori ottimi di  $J_1$  (in questo caso proprio il migliore delle simulazioni) e di  $J_2$  sono per  $\rho_a = 80$ , punto in cui l'area media è la peggiore stimata.

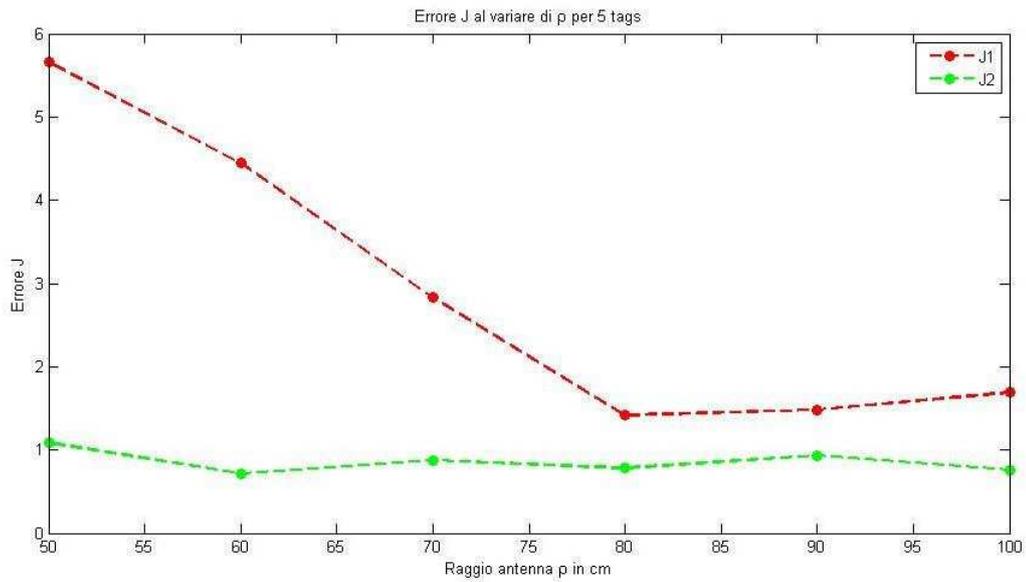


Figura 5.11: Grafico degli errori  $J_1$  e  $J_2$  per  $n_{tag} = 5$

#### 5.4.4 Osservazioni

Dai dati riportati dai grafici risulta interessante osservare che nel caso di  $n_{tag} = 4$  e  $n_{tag} = 5$  errori minimi sono stati ottenuti per raggi ampi rispetto alla media  $e$ , soprattutto per valori di  $\sigma$  molto elevati in rapporto all'area media. L'aumentare del raggio influisce nel creare regioni di varie grandezze: quanto più una regione è piccola tanto più risulta precisa la stima della posizione quando il robot vi passa attraverso. Le regioni più piccole si trovano verso il centro dove più circonferenze si intersecano. Per come è stato disegnato il percorso, il robot passa proprio attraverso queste regioni fornendo perciò una stima più accurata che non passando lungo i bordi dove le regioni sono più ampie. Il nodo della questione resta perciò determinare una relazione tra il raggio dell'antenna e il numero di tags affinché sia minimo l'indice d'errore. La deviazione standard influisce sì su  $J$ , definendo la variazione delle regioni in media, ma non è sufficiente a determinare un legame tra numero di tags e raggio d'antenna grazie al quale si troverebbero le dimensioni ideali delle regioni per minimizzare efficacemente l'errore.

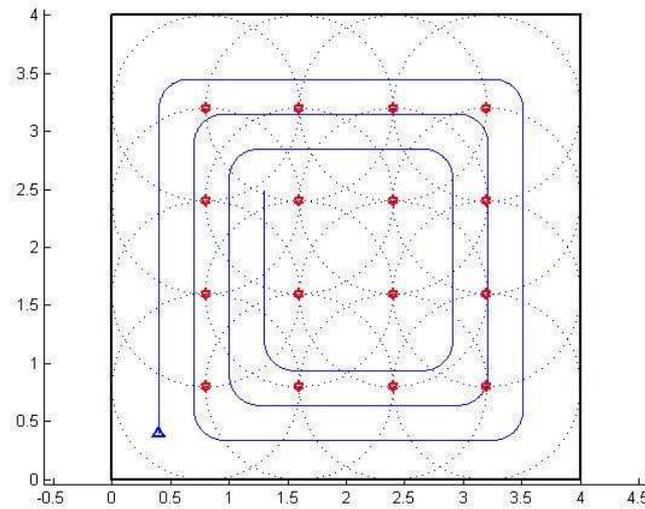


Figura 5.12: Percorso attraverso regioni RFID per  $n_{tag} = 5$ ,  $\rho_a = 80$

## 5.5 Filtri a confronto

Vengono ora messi a confronto il filtro PF, con  $M_p = 1000$ , e l'EKF sfruttando un metodo d'approssimazione degli ingressi a disturbi gaussiani basato covarianza delle regioni RFID<sup>5</sup>. Il grafico in Figura 5.5 mostra il valore dell'errore totale  $J_1$  per un disturbo di lettura ( $\lambda$ ) nullo e 4 tags per ogni lato.

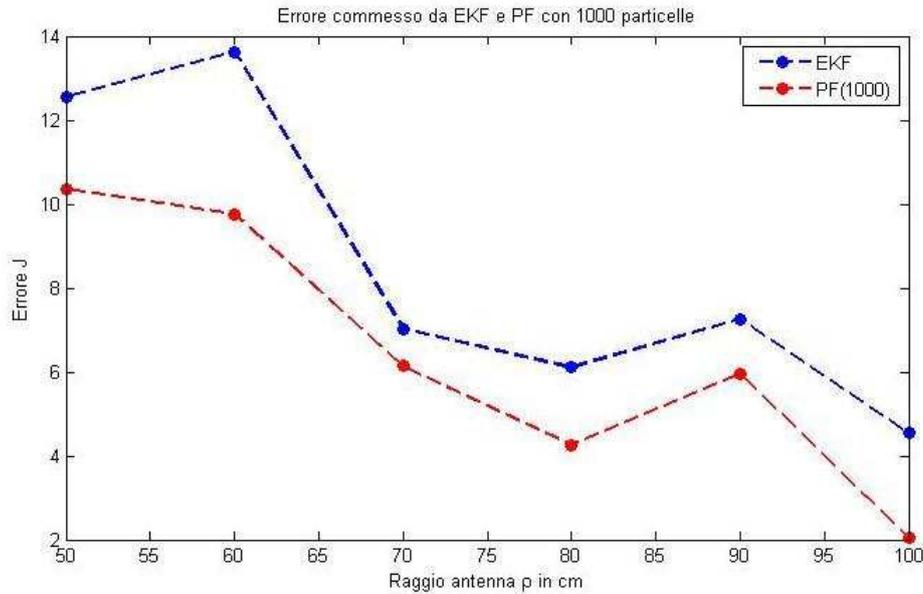


Figura 5.13: Grafico dell'errore  $J_2$  (relativo alla seconda metà del percorso) dei filtri PF ( $M_p = 1000$ ) ed EKF con metodi met1noBord e met1Bord, al variare del raggio  $\rho_a$  con specifiche  $\lambda = 0$  e  $n_{tag} = 4$ .

Dal grafico si nota l'efficacia del filtro particellare rispetto a quello di Kalman in particolare per valori bassi di  $\rho_a$ . Tuttavia il filtro di Kalman è nettamente più veloce nell'esecuzione. Il tempo d'esecuzione cresce all'aumentare del raggio dell'antenna per entrambi i filtri. Tuttavia mentre per l'EKF il tempo impiegato ad ogni iterazione è intorno al secondo se non inferiore, per il PF il tempo è superiore al minuto, fino ad arrivare oltre i 3 minuti per i valori di  $\rho_a$  maggiori. Per la sproporzione presente tra tali valori si è deciso di non riportare né rappresentare graficamente il tempo impiegato dai vari metodi.

<sup>5</sup>Per maggiori delucidazioni su questi metodi si veda [12]

## Capitolo 6

# Conclusioni

In questa tesi è stata trattata la localizzazione globale di un robot in grado di rilevare degli RFID tags disposti sul soffitto della stanza in cui si muove. Un esempio pratico è quello di un robot dotato di sensori di temperatura che si muove in una stanza piena di elaboratori e che ha il compito di controllare che la loro temperatura non superi certi valori. Se il robot deve seguire un certo percorso, deve necessariamente essere in grado di localizzarsi, in modo da poter correggere la traiettoria, nel caso di deviazioni dal tragitto predefinito, ma anche per essere in grado di identificare il luogo, la posizione, in cui magari venga rilevata una temperatura troppo elevata.

La localizzazione risulta perciò un problema importante nell'ambito della robotica. È stato mostrato inoltre come la sola odometria (i dati provenienti dalle letture encoder) non sia sufficiente da sola ai fini della localizzazione a causa di un errore cumulativo che rende la stima della posizione del tutto fallace dopo un certo numero di misurazioni. A questa misurazione deve essere perciò affiancata una stima che sfrutta sensori di prossimità quali laser o sensori ottici. Gli RFID rappresentano una valida alternativa rispetto ad altri sensori proponendo una stima efficace quando abbinata all'odometria e un'economicità nei costi che altri sensori non offrono. Per integrare i dati provenienti dagli encoders delle ruote con quelli derivanti dalla ricezione degli RFID si usano dei filtri matematici tra cui il filtro particellare (PF) mostrato in questa tesi. Tale filtro si basa sul metodo Monte Carlo, ovvero sulla campionatura dell'ambiente creando delle particelle in diverse posizioni, cui viene fornito lo stesso moto applicato al robot secondo le letture odometriche. Scegliendo poi le particelle che ad ogni passo si trovano più vicine alla regione

RFID vista, si crea il gruppo di particelle da utilizzare al passo successivo clonando quelle scelte in numero proporzionale alla loro probabilità di stare nella regione vista.

Sono stati poi mostrati i risultati delle simulazioni al variare di certi parametri sia dell'ambiente che della tecnologia usata. A partire da un percorso fissato è stata mostrata la stima calcolata dal PF per determinare la posizione del robot, che è risultata soddisfacente già prima della metà del tragitto. Introdotto, inoltre, un coefficiente d'errore di stima  $J$  si sono potuti apprezzare numericamente, oltre che graficamente, gli effetti di questo filtro. Messa a confronto con l'EKF il filtro particellare è nettamente più preciso presentando valori d'errore anche inferiori al centimetro. Nota dolente del PF rimane però la complessità computazionale che risulta molto più elevata di altri filtri come l'EKF, in particolare all'aumentare del numero di campioni (i.e. particelle) utilizzate. Per questo motivo si tende a combinare ad esempio l'EKF al PF lasciando il calcolo della stima a quest'ultimo dopo aver scremato le variabili e aver affidato i conti più lunghi al filtro di Kalman esteso (vedi [10]).

Tramite altre simulazioni, variando il raggio dell'antenna e il numero di tags utilizzati, si è cercato di determinare una relazione tra la deviazione standard delle regioni RFID create in rapporto con l'area media, e l'errore di stima  $J$ . Dai conti effettuati risulta che per regioni di dimensioni differenti il filtro funziona meglio piuttosto che con regioni di superfici più uniformi. Inoltre le prestazioni del filtro migliorano all'aumentare del numero di tags e per un'ampiezza del raggio d'antenna elevato (intorno agli 80 cm) .

In definitiva il filtro particellare risulta ottimale dal punto di vista dell'efficacia, ma non altrettanto buono per quanto riguarda la velocità d'esecuzione. Nel caso di numerosi stati da stimare, conviene affiancargli filtri più veloci come lo stesso filtro di Kalman esteso lasciando le stime in cui risulta rilevante la precisione al filtro particellare.

# Bibliografia

- [1] R. Aliberti, E. Di Giampaolo e G. Marrocco. **A model to estimate the RFID read-region in real environment.** *Proc. of the European Wireless Technology Conference, Amsterdam, October 2008.*
- [2] F. Picotto. **Progetto e realizzazione di un lettore/scrittore general purpose per tag RFID a 125 kHz.** *Tesi di Laurea Specialistica in Ingegneria Elettronica, Università di Udine.*
- [3] G. Galati, M. Naldi, G. Pavan. **Teoria dei fenomeni aleatori: fondamenti e prime applicazioni ai modelli stocastici, Volume I.** *Texmat, 2002.*
- [4] G. Galati, M. Naldi, G. Pavan. **Teoria dei fenomeni aleatori: Modelli stocastici e Metodi Statistici, Volume II.** *Texmat, 2002.*
- [5] Y. Zhou, W Liu e P.Huang. **Laser-activated RFID-based Indoor Localization System for Mobile Robots.** *IEEE International Conference on Robotics and Automation (ICRA), April 2007, Rome, Italy.*
- [6] D. Hahnel, W. Burgard, D. Fox, K. Fishkin e M. Philipose. **Mapping and Localization with RFID Technology.** *IEEE International Conference on Robotics and Automation (ICRA), 2004.*
- [7] S. Han, H. Lim e J. Lee. **An efficient localization scheme for a different-drive mobile robot based on RFID system.** *IEEE Trans. Industrial Electronics, Dec. 2007.*
- [8] M. Kim, T. Takeuchi e N.Y. Chong. **A 3-axis Orthogonal Antenna for Indoor Localization.** *In First International Workshop on Networked Sensing Systems, Vol 50, NO. 2, pp. 56-62, 2004.*

- [9] C. Demetrescu, I. Finocchi e G. Italiano. **Algoritmi e strutture dati**. *McGraw-Hill, 2004*.
- [10] S. Bittante. **Ricostruzione dell'angolo di rollio di una motocicletta tramite filtro particellare e sensori GPS/INS**. *Tesi di Laurea in Ingegneria Informatica, Università degli Studi di Padova, 28 Aprile 2006*.
- [11] S. Arulampalam, S. Maskell, N. Gordon e T. Clapp. **A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking**. *IEEE Transaction on Signal Processing, pagg. 174-188, Feb. 2002*.
- [12] M. Boccadoro, F. Martinelli e S. Pagnottelli. **Constrained and quantized Kalman filtering for an RFID robot localization problem**. *Submitted to Automatics Robots*.