

UNIVERSITÀ DEGLI STUDI DI ROMA LA SAPIENZA  
FACOLTÀ DI INGEGNERIA

Tesi di Laurea Magistrale in  
INGEGNERIA DELLE COMUNICAZIONI

**ALGORITMI IBRIDI PER IL  
POSIZIONAMENTO INDOOR BASATI  
SU TECNOLOGIE WI-FI E RFID.**

Relatore  
Prof. Luca De Nardis

Candidato  
Luca Liparulo

Anno Accademico 2010/2011



---

# Indice

---

<b>1</b>	<b>Positioning Indoor: sistemi basati sul Wi-Fi</b>	<b>8</b>
1.1	Introduzione . . . . .	9
1.2	RSS-based Wi-Fi Indoor Localizer . . . . .	10
1.2.1	Nearest Neighbor in Signal Space (NNSS) . . . . .	10
1.2.2	Nearest Neighbor in Signal Space - Average (NNSS - AVG)	11
1.2.3	Smallest Polygon . . . . .	12
1.3	TOA-based Wi-Fi Indoor Localizer . . . . .	13
1.3.1	“RTT measurement using IEEE 802.11 data/ACK frames”	13
1.4	Confronto RSS-based vs TOA-based . . . . .	16
1.5	Bluetooth & Wi-Fi . . . . .	23
1.6	Conclusioni . . . . .	30

---

<b>2 Positioning Indoor: sistemi basati su RFID</b>	<b>31</b>
2.1 Introduzione . . . . .	32
2.2 Un primo accenno storico . . . . .	32
2.3 Componenti del sistema . . . . .	34
2.3.1 TAG . . . . .	35
2.3.2 READER . . . . .	43
2.4 Costi . . . . .	44
2.5 Positioning Indoor . . . . .	45
2.5.1 SpotON . . . . .	45
2.5.2 Landmarc . . . . .	48
2.6 Analisi progettuale. . . . .	52
2.6.1 Introduzione: Cosa e Come. . . . .	52
2.6.2 Soluzione 1 . . . . .	53
2.6.3 Soluzione 2 . . . . .	56
2.7 Conclusioni . . . . .	58
<b>3 Positioning Indoor: Ibridizzazione Wi-Fi e RFID</b>	<b>59</b>
3.1 Introduzione . . . . .	59
3.2 Il Progetto . . . . .	60
3.2.1 SPinV . . . . .	60
3.3 Algoritmi studiati . . . . .	61
3.3.1 Soluzione # 1: Wi-Fi . . . . .	61
3.3.2 Soluzione # 2: Wi-Fi + RFID . . . . .	63

INDICE	4
3.3.3 Soluzione # 3: RFID → Future Work . . . . .	64
3.4 Le nostre scelte . . . . .	65
<b>4 MatLab: Risultato SIMULAZIONI</b>	<b>72</b>
4.1 Introduzione . . . . .	73
4.2 Wi-Fi . . . . .	73
4.2.1 Free Space Model (Caso ideale) . . . . .	74
4.2.2 Multi wall Model: Variazione nr. Access Point in funzione del passo della griglia . . . . .	76
4.2.3 Multi wall Model: Variazione numero di Neighbors (K) in funzione del passo della griglia . . . . .	78
4.3 Wi-Fi + RFID . . . . .	80
4.3.1 Multi wall Model (Wi-Fi + RFID): Variazione nr. Access Point in funzione del passo della griglia . . . . .	80
4.3.2 Multi wall Model (Wi-Fi + RFID): Variazione numero di Neighbors (K) in funzione del passo della griglia . . . . .	82
<b>5 Conclusioni e Lavori Futuri</b>	<b>84</b>
<b>Bibliografia</b>	<b>87</b>
<b>A Codice Simulatore</b>	<b>94</b>

---

## Elenco delle figure

---

1.1	Algoritmo NNSS-AVG . . . . .	11
1.2	Algoritmo Smallest Polygon . . . . .	12
1.3	Esempio di Trilaterazione . . . . .	14
1.4	Calcolo del Round Time Robin ( $RTT$ ) . . . . .	15
1.5	Scalabilità TOA: Tempo speso per il calcolo di una pseudo-distanza	19
1.6	Scalabilità TOA - Richieste di ritrasmissione . . . . .	20
1.7	Funzione di Distribuzione Cumulativa (CDF) di errore nella stima della posizione . . . . .	22
1.8	Absolute Signal Strength percepito su un AP Bluetooth (RSS) .	27
1.9	Absolute Signal Strength percepito su un AP Wi-Fi (RSS) . . .	27
1.10	Signal Strength Difference tra 2 APs Bluetooth (SSD) . . . . .	28
1.11	Signal Strength Difference tra 2 APs Wi-Fi (SSD) . . . . .	28

---

1.12	Comparazione tra SSD e RSS confrontando gli algoritmi KNN e di Bayes . . . . .	29
2.1	Esempio di TAG passivo . . . . .	35
2.2	Grafico relativo ai costi della tecnologia RFID . . . . .	44
2.3	Risultati ottenuti con SpotON . . . . .	46
2.4	Errore percentile cumulativo del calcolo della distanza per $k \in (2, 5)$ . . . . .	49
2.5	Errore percentile cumulativo del calcolo della distanza durante il giorno e la notte. . . . .	50
2.6	Errore percentile cumulativo del calcolo della distanza al variare del numero dei readers da 3 a 4. . . . .	51
2.7	Esempio di partizionamento fisico per separare i reference tags 'c' ed 'f' dagli altri . . . . .	51
2.8	Positioning Indoor: Soluzione 1 . . . . .	53
2.9	Componenti del sistema (Soluzione 1) . . . . .	54
2.10	San Pietro in Vincoli: Mappa del primo piano . . . . .	55
3.1	Stima esplicativa percorso utente . . . . .	62
3.2	Range lettura RFID Reader nel nostro simulatore . . . . .	64
3.3	Posizione Range Reader nel nostro simulatore . . . . .	64
3.4	San Pietro in Vincoli: Primo Piano . . . . .	66
3.5	San Pietro in Vincoli: Mappa creata con Matlab . . . . .	67
3.6	San Pietro in Vincoli: Posizionamento Access Point . . . . .	68
3.7	San Pietro in Vincoli: Posizionamento Punti di Interesse . . . . .	69

---

4.1	Posizione dell'utente - Numerazione AP considerati . . . . .	74
4.2	Free Space Model: Passo della griglia: 1 . . . . .	75
4.3	Free Space Model: Passo della griglia: 2 . . . . .	75
4.4	Free Space Model: Passo della griglia: 10 . . . . .	75
4.5	Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 1 . . . .	76
4.6	Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 2 . . . .	77
4.7	Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 10 . . . .	77
4.8	Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 1	79
4.9	Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 5	79
4.10	Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 10 . . . . .	79
4.11	Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 3	81
4.12	Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 5	81
4.13	Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 10 . . . . .	81
4.14	Wi-Fi + RFID: Variazione K Neighbours da 1 a 10 - Passo della griglia: 5 . . . . .	83
4.15	Wi-Fi + RFID: Variazione K Neighbours da 1 a 10 - Passo della griglia: 10 . . . . .	83



---

## Positioning Indoor: sistemi basati sul Wi-Fi

---

### Indice

---

<b>1.1</b>	<b>Introduzione</b>	<b>9</b>
<b>1.2</b>	<b>RSS-based Wi-Fi Indoor Localizer</b>	<b>10</b>
1.2.1	Nearest Neighbor in Signal Space (NNSS)	10
1.2.2	Nearest Neighbor in Signal Space - Average (NNSS - AVG)	11
1.2.3	Smallest Polygon	12
<b>1.3</b>	<b>TOA-based Wi-Fi Indoor Localizer</b>	<b>13</b>
1.3.1	“RTT measurement using IEEE 802.11 data/ACK frames”	13
<b>1.4</b>	<b>Confronto RSS-based vs TOA-based</b>	<b>16</b>
<b>1.5</b>	<b>Bluetooth &amp; Wi-Fi</b>	<b>23</b>
<b>1.6</b>	<b>Conclusioni</b>	<b>30</b>

---

## 1.1 Introduzione

In questo lavoro vogliamo presentare uno studio riguardo il “positioning indoor” [Lassabe et al. [16]], cioè la possibilità di poter individuare e riconoscere un terminale mobile all’interno di spazi chiusi, in modo tale da poter interagire con lo stesso fornendo servizi di vario genere.

La prima precisazione da fare è che tale problematica non è attualmente così sviluppata, al contrario di quanto accade in ambiente outdoor in cui si approfitta dell’aiuto del GPS.

Sono state fatte molte ricerche circa lo sviluppo di architetture di servizi per sistemi di posizione, meno attenzione invece è stata dedicata al problema fondamentale e impegnativo della localizzazione e tracciamento degli utenti mobili, soprattutto in ambienti indoor. Le poche persone che hanno affrontato questo problema in genere lo hanno fatto nel contesto delle reti wireless infrarossi (IR). Inoltre, la rete di IR è spesso distribuita al solo scopo di individuazione di persone e non fornisce i dati tradizionali di servizi di rete. Per ovviare a queste limitazioni, ci siamo dunque occupati in modo particolare della tecnologia Wi-Fi (IEEE 802.11b standard) nella nostra ricerca.

Il nostro obiettivo è integrare i dati di funzionalità delle reti Wi-Fi con le capacità di rilevamento di posizione dell’utente di rete [Saha et al. [28]], accrescendo il valore di tali reti. Gli approcci al problema, anche nella stessa tecnologia, possono essere diversi e in particolare noi ci siamo soffermati su:

- RSS-based Wi-Fi indoor localizer [Vaupel et al. [31]];
- TOA-based Wi-Fi indoor localizer [Ciurana et al. [4]].

E’ importante prima di tutto esplicitare le caratteristiche di questi due approcci per poter poi fare un confronto accurato.

Uno dei primi sistemi RSS-based è sicuramente “RADAR” [Bahl and Padmanabhan [2]], che rappresenta un punto di riferimento per tutti i sistemi successivi, che è in grado di stimare la posizione dell’utente con un’accuratezza di qualche metro rispetto alla sua posizione reale, ciò ovviamente vuol dire che tale tecnologia può essere utilizzata da una grande classe di servizi.

## 1.2 RSS-based Wi-Fi Indoor Localizer

Un sistema basato su RSS, utilizza la tecnica del “fingerprinting” [Kaemarungsi and Krishnamurthy [15]] (impronte digitali) e può essere diviso in due fasi differenti. In una prima ‘off-line phase’ si crea una griglia immaginaria all’interno dell’ambiente che si vuole analizzare e su ogni intersezione  $j$  si registra il vettore  $M_j = \langle \mu_{j,1}, \mu_{j,2}, \dots, \mu_{j,N} \rangle$ , in cui  $\mu_{j,i}$  rappresenta una media di ‘x’ rilevazioni del valore dell’RSSi da ogni access point (APi). Si costruisce allora un database (radio-map) all’interno del quale ogni record contiene la tupla  $(x_j, y_j; M_j)$  e identifica la  $j$ -esima intersezione (posizione). Dopo di che abbiamo una seconda fase, la ‘tracking phase’, in cui con l’ingresso del (o dei) dispositivo mobile si effettua, con una serie di algoritmi, la ricerca della sua posizione.

Gli algoritmi che abbiamo analizzato sono quelli di tipo deterministico, tra cui i piu’ efficienti sono descritti qui di seguito.

### 1.2.1 NEAREST NEIGHBOR IN SIGNAL SPACE (NNSS)

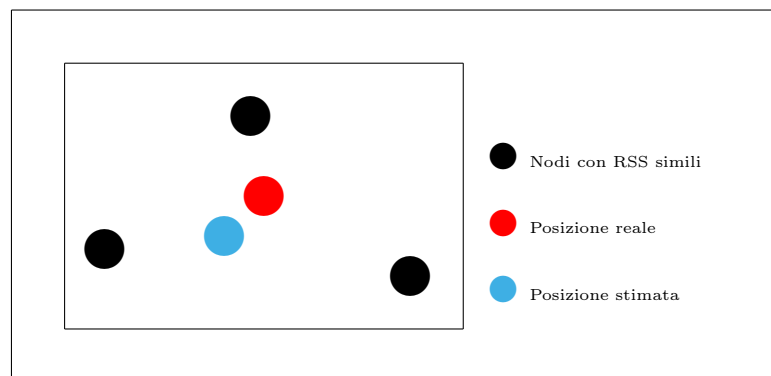
Questo metodo [Tran et al. [30]] computa la distanza tra il valore dell’RSS misurato dal dispositivo mobile e tutti gli altri valori di RSS nella radio-map. La posizione del dispositivo è stimata come la minima distanza trovata, che corrisponde alla piu’ piccola distanza euclidea. Questo vuol dire che dalla posizione dell’utente, dal terminale mobile, si registra un vettore  $R = \langle r_1, r_2, \dots, r_N \rangle$ ,

in cui  $r_i$  è l' $i$ -esimo valore RSSi proveniente dall'access point APi. A questo punto per il calcolo della piu' piccola distanza euclidea, si calcola:

$$D_j = \sqrt{\sum_{k=1}^N (r_i - \mu_{j,i})^2}$$

Tutti i valori scalari di  $D_j$  vengono poi ordinati in ordine crescente, quindi il primo valore del vettore risultante sarà quello che dovrebbe corrispondere alla posizione piu' vicina a quella reale del terminale mobile. Allora l'accuratezza dipenderà dalla grandezza della griglia.

### 1.2.2 NEAREST NEIGHBOR IN SIGNAL SPACE - AVERAGE (NNSS - AVG)



**Figura 1.1:** Algoritmo NNSS-AVG

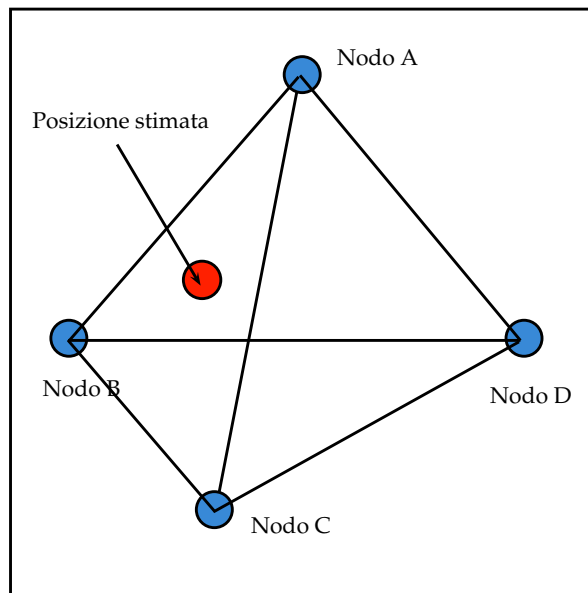
Questo algoritmo [Roxin et al. [27]] è simile al precedente, con la differenza sostanziale che invece di scegliere un solo vicino, sceglie i  $K$  nearest neighbours e computa quindi la posizione del dispositivo mobile facendo una media sulle coordinate degli stessi neighbours. La differenza sostanziale in termini di efficienza è che mentre nell'algoritmo precedente, l'accuratezza massima raggiungibile in ogni caso, dipendeva soltanto dalla tipologia della griglia, cioè

da quanto distanti fossero le posizioni da cui venivano immagazzinati tutti i dati per la creazione del database, facendo una media su diverse posizioni differenti, abbiamo una minore possibilità di errore con conseguente aumento di precisione nella localizzazione. In questo modo, trovati i  $K$  nearest neighbours, le coordinate del terminale sarebbero:

$$(x, y) = \frac{1}{K} \sum_{j=1}^K (x_j, y_j)$$

### 1.2.3 SMALLEST POLYGON

Questo metodo [Roxin et al. [27]] è molto simile al NNSS-SVG, viene citato in letteratura ma non garantisce prestazioni elevate. Dopo aver calcolato i  $K$  nearest neighbours, crea vari poligoni aventi come vertici esattamente i  $K$  punti scelti. Dopo di che viene stimata l'area di tutti e l'algoritmo sceglie il poligono con l'area più piccola, per poi definire la posizione del terminale mobile al centro dello stesso.



**Figura 1.2:** Algoritmo Smallest Polygon

### 1.3 TOA-based Wi-Fi Indoor Localizer

In un articolo del 2007 [Golden and Bateman [8]] vediamo che i sistemi di localizzazione basati su TOA non sono formalmente proposti né nella letteratura scientifica, né come prodotti da aziende private, perché questa tecnica ha ancora dei problemi che devono essere risolti. Soltanto la Intel recentemente ha pubblicato un interessante approccio basato su TOA, che tende ad aggiungere accuratezza allo standard 802.11v con modifiche dello strato fisico WLAN. Tutto ciò perché è difficile stimare con precisione le distanze con un approccio basato su TOA relativamente semplice, per la mancanza di precise timestamp nella trasmissione e nella ricezione di frame nello standard IEEE 802.11.

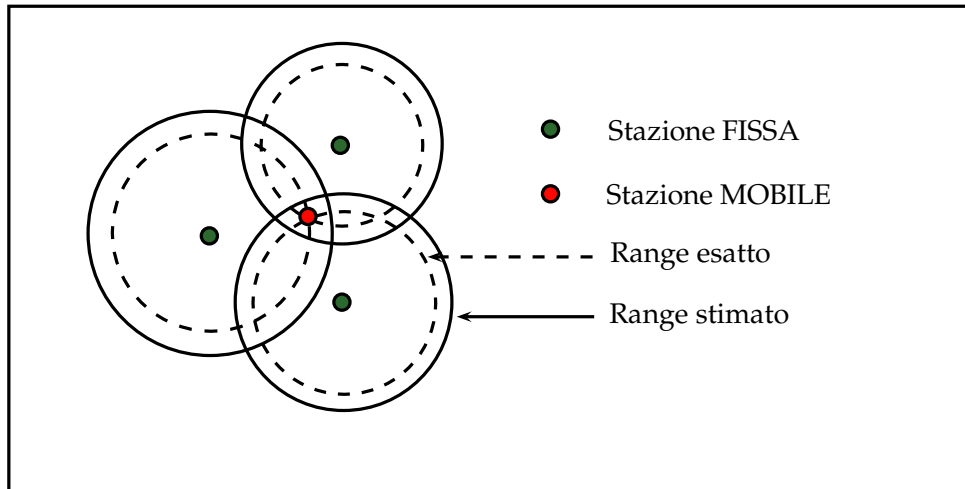
Sono stati pubblicati in ogni caso diverse soluzioni che trattano questo argomento (es. [Gunther and Hoene [9]], [Ciurana et al. [3]]), ma una soluzione fattibile e accurata, lavorando in un dispositivo Wi-Fi standard è ancora da trovare; per tal motivo noi non ci siamo soffermati sulla trattazione degli algoritmi utilizzati in tali modelli.

In ogni caso andiamo ad analizzare in maniera più precisa questo secondo approccio che resta comunque significativo, anche se ancora in via di sviluppo, per sfruttare il Wi-Fi per la localizzazione indoor e che si basa sulla misura della propagazione del ritardo (TOA) tra i diversi punti di accesso e un utente di telefonia mobile, utilizzando la tecnica della trilaterazione (Fig. 1.3). Una delle proposte più interessanti [Ciurana et al. [4]] a tal proposito, è stata presa come riferimento nei prossimi paragrafi.

#### 1.3.1 “RTT MEASUREMENT USING IEEE 802.11 DATA/ACK FRAMES”

Possiamo basarci sul lavoro appena citato [Ciurana et al. [4]] per esprimere in maniera dettagliata quali siano le caratteristiche del secondo approccio.

Dopo l’eliminazione delle tecniche TDOA (Time Difference of Arrive) a causa



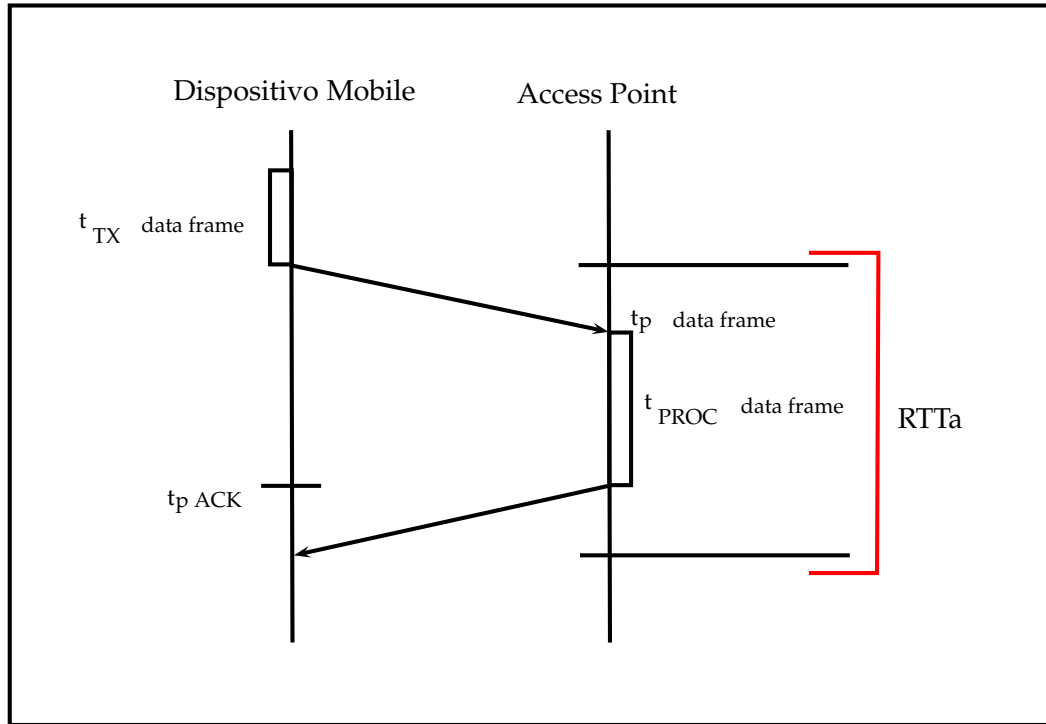
**Figura 1.3:** Esempio di Trilaterazione

della necessità di sincronizzazione tra i nodi, Ciurana et al. hanno implementato un metodo che ottiene i valori TOA mediante l'esecuzione di misurazioni dell'RTT (Round Trip Time) da un terminale mobile a un punto di accesso fisso, in modo da evitare la sincronizzazione tra i nodi della WLAN e quindi in modo da non incrementare la complessità e il costo della gestione del sistema.

RTT [Llombart et al. [18]] è il tempo speso da un segnale o un messaggio in viaggio da un trasmettitore a un ricevitore, con ritorno nuovamente al trasmettitore. Dal momento che si vuole sfruttare al massimo l'esistente infrastruttura di rete di comunicazione per stimare con precisione le distanze, vengono utilizzati nei test di Ciurana (e comunque in tutte le sperimentazioni a basso costo sul TOA) panorami standard IEEE 802.11 per misurare l'RTT, nello specifico: Data e ACK MAC frame.

Hanno quindi utilizzato l'orologio disponibile a 44 MHz ( $f_{clk}$ ) nella scheda WLAN del terminale mobile come contatore del tempo, in modo da raggiungere una risoluzione temporale di 22 ns.

La distanza 'a' tra un terminale mobile e un access point si è ottenuta



**Figura 1.4:** Calcolo del Round Time Robin ( $RTT$ )

dunque moltiplicando il TOA stimato, con la velocità della luce:

$$a = c * tp = c * TOA$$

Essendo  $RTT$  il tempo che il pacchetto impiega ad andare dal Tx al Rx e ritorno, allora scriviamo:

$$a = c * \left( \frac{\Delta RTT}{2} \right)$$

In base a quanto detto precedentemente, come si vede in Fig. 1.4, l' $RTT$  corrisponde al tempo trascorso tra l'invio del link-layer data frame e la conseguente ricezione del ACK frame sul MT, quindi:

$$RTT_a = (2 * tp) + t_{procdata\ frame}$$



Il  $\Delta RTT$  che ci interessa si ottiene sottraendo ad  $RTT_a$  il tempo che impiega il MAC a processare il data frame, quindi sia:

$$RTT_o = t_{proc\ data\ frame}$$

$$\Delta RTT = RTT_a - RTT_o$$

$$a = c * \left( \frac{RTT_a - RTT_o}{2} \right) * \left( \frac{1}{f_{clk}} \right) \quad \Longrightarrow \quad TOA = \frac{RTT_a - t_{proc\ data\ frame}}{2}$$

#### 1.4 Confronto RSS-based vs TOA-based

Un confronto tra queste due tecnologie, può essere fatto in base a diversi parametri:

1. il *COSTO* del sistema, che deve essere ragionevole, sia in termini economici, che di tempo e spazio;
2. l'*ACCURATEZZA*, che è sicuramente il parametro più importante;
3. la *PRECISIONE*, che misura la robustezza della tecnica di posizionamento;
4. la *SCALABILITA'*.

Una delle questioni più importanti, quella riguardante i costi, è molto incoraggiante in queste tecnologie in quanto, come dicevamo, esse si basano sull'architettura standard del Wi-Fi quindi non hanno bisogno di alcun intervento esterno di tipo hardware.

Il sistema RSS-based ad esempio può richiedere soltanto tre stazioni base preconfigurate; quello che risulta più difficile è il monitoraggio e la creazione del database.

Se viene utilizzato un modello empirico, c'è bisogno di misurazioni effettuate a ogni punto di riferimento.

Se devono essere utilizzati più modelli per tenere conto delle modifiche di ambiente, c'è bisogno di misurazioni effettuate a ogni punto di riferimento in ogni situazione ambientale.

Se è necessario utilizzare un modello analitico, devono essere ottenuti e analizzati disegni architettonici e c'è bisogno di analizzare fattori di attenuazione e misurare il modello applicato a ogni punto di riferimento.

Questo sicuramente non causa nessun tipo di peggioramento di tipo economico, però riporta un notevole peggioramento in termini di tempo.

Tutto questo non accade nei sistemi TOA-based in quanto non abbiamo bisogno di una prima fase di monitoraggio, però, come accennato in precedenza, in questo caso bisogna fare i conti con il problema della non-precisione dei clock dei router di tipo standard che vengono utilizzati (dobbiamo essere consapevoli che imprecisioni di appena 1 microsecondo introdurrebbero un errore di localizzazione dell'ordine dei 300 m.).

La conseguenza di quanto appena detto può essere vista in maniera più precisa andando ad analizzare la scalabilità dei due sistemi. Un risultato che sicuramente si vorrebbe ottenere è un sistema con una discreta scalabilità per un numero elevato di utenti poiché tecniche basate su TOA comportano la generazione di traffico di rete in ogni processo di calcolo della distanza, il numero di MTs che simultaneamente possono ottenere la propria stima di posizione mantenendo una qualità accettabile di servizio è limitato.

Più specificamente, è previsto che il tempo di risposta aumenta molto in situazioni di gran numero di MTs perché tenendo conto della natura del MAC IEEE 802.11 [art [1]], dobbiamo aspettarci di avere sicuramente un gran numero di collisioni, ritrasmissioni e individuazioni di sistema occupato. Allo stesso tempo, poiché in tali situazioni il raggiungimento delle distanze necessarie per

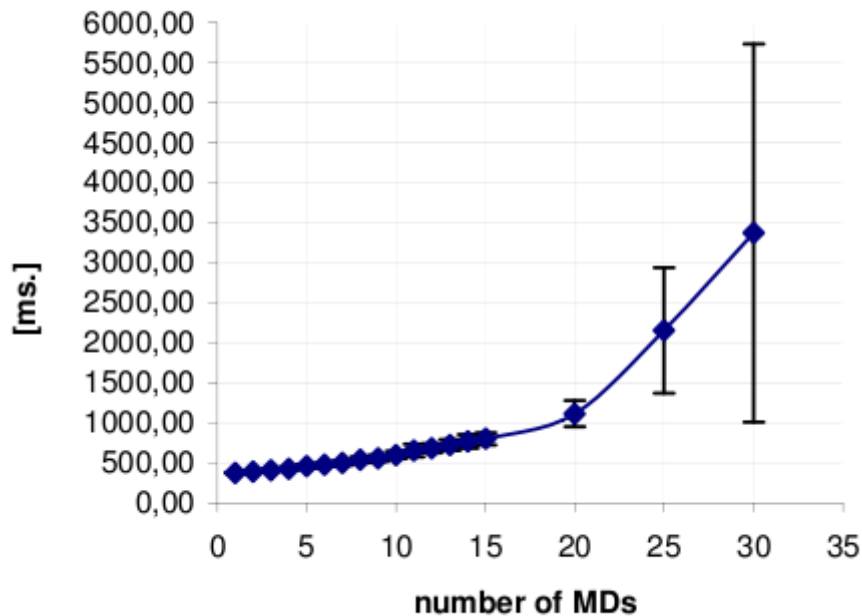
calcolare una posizione può essere ritardato, la precisione di posizionamento può anche essere compromessa soprattutto se la velocità della destinazione è elevata.

Deve essere detto che proposte di Wi-Fi che utilizzano metodi servendosi solo delle misurazioni di RSS evitano la necessità di accedere al MAC (cosa che invece viene fatta nei sistemi basati su TOA per la stima del RTT) e quindi si prevede di avere una migliore scalabilità.

Riportiamo allora degli studi fatti da Llombart, Ciurana, e Barceló-Arroyo [Llombart et al. [18]] che illustrano l'enorme peggioramento nei sistemi basati su TOA nel caso di crescita del numero di utenti. Nella Fig. 1.5 si può apprezzare una relazione lineare fra il tempo speso per ottenere una stima di una distanza e il numero di MDs nel sistema, con un basso ritmo di crescita in situazioni di meno di 20 utenti. Numericamente, in questo intervallo di numero di utenti la durata media della stima della distanza oscilla tra 400 ms (un solo utente in sala) e un secondo e la latenza di posizionamento non è maggiore di 3 secondi; infatti è possibile ottenere una posizione in meno di 1,5 secondi per situazioni di meno di 7 MTs. Tuttavia, in situazioni con più di 20 utenti la situazione peggiora notevolmente, raggiungendo più di 10 secondi di latenza per 30 dispositivi. Questa involuzione in corrispondenza della pendenza a 20 utenti nel sistema indica un punto di saturazione del traffico, in cui le prestazioni del sistema iniziano a peggiorare molto velocemente a causa di più collisioni, tentativi e difficoltà di accesso al MAC.

Quanto appena detto è corroborato con il grafico nella Fig. 1.6 in cui è raffigurata la crescita delle richieste di ritrasmissione man mano che cresce il numero di MTs. Da notare che da 20 MT in poi, il canale è congestionato e il numero di ritrasmissioni non può aumentare; la conseguenza è un inaccettabile crescita della latenza.

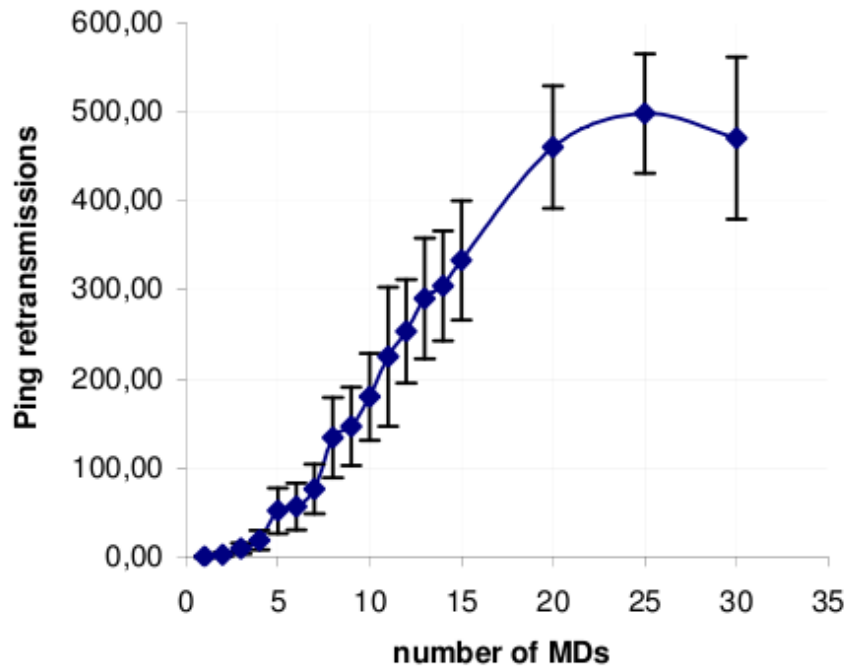
Tornando a quanto detto prima circa gli studi di Ciurana et al., è importante fare alcune considerazioni sul fatto che, essendo le misurazioni degli *RTT*



**Figura 1.5:** Scalabilità TOA: Tempo speso per il calcolo di una pseudo-distanza

affette da errori e imprecisioni, conviene sempre affidarsi ad un approccio di tipo probabilistico.

Dopo diverse analisi, questi studi hanno portato alla conclusione che un numero ragionevole di misurazioni per stimare con precisione la  $RTT$  è stato  $n = 300$ , effettuato in circa 1,5 secondi. Infine, al fine di ottenere una caratterizzazione statistica della precisione del sistema, la funzione di densità di probabilità (PDF) della stima della distanza è stata calcolata per normalizzare un istogramma empirico, ottenuto tenendo conto di 500 distanza stimate a una distanza fissa di 11 m, in situazione di LOS tra il punto di accesso e il terminale mobile. Confrontando la PDF risultante con note distribuzioni di probabilità infatti, si è accertato che quella che meglio si adatta era una distribuzione gaussiana con un valore medio di 11.12 m e varianza 0.84 m.



**Figura 1.6:** Scalabilità TOA - Richieste di ritrasmissione

La PDF è stata utilizzata nelle simulazioni di stima della posizione con due algoritmi (Non-Linear Least Squares & GPS Least-Squares) [Madsen [19]]. Le simulazioni si sono effettuate come segue:

- sono state introdotte le posizioni dei tre punti di accesso e la vera posizione del TM (che stava per essere stimato);
- il programma di simulazione ha calcolato le distanze esatte da ogni punto di accesso per la MT. Queste distanze sono state modificate utilizzando la distribuzione gaussiana già citata;
- un gran numero di stime della posizione delle Mts sono state eseguite con i due algoritmi citati, prendendo come input per ognuno una diversa combinazione di stime distanza dalle distribuzioni di probabilità che

variano dei punti di tre accesso. Così tutte le stime di posizione sono state ottenute considerando tutte le possibili combinazioni delle distanze stimate tenendo conto della variazione del modello. Dopo di che si sono calcolati gli errori di stima;

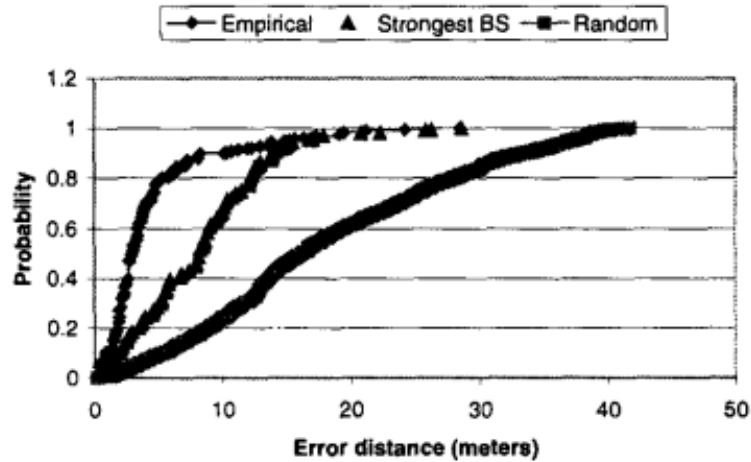
- infine, è stata calcolata la funzione di probabilità cumulativa (CDF) dell'errore di stima di posizione.

In uno scenario in cui il terminale mobile si trova all'interno del triangolo formato da tre access point (caso migliore), la precisione è superiore a 1,4 metri nel 66% dei casi. In una situazione in cui il terminale mobile non è all'interno del triangolo dei punti di accesso, ma sono correttamente distribuito (non allineati), la precisione è migliore di 1,8 m con una probabilità del 66%. Si può vedere anche che l'algoritmo Non-Linear Least Squares supera GPS Least-Squares in entrambi i casi. In conclusione, gli autori ottengono un'accuratezza di poco superiore al metro attraverso misurazioni TOA.

Per quanto riguarda invece i risultati ottenuti con la tecnologia RSS-based, andiamo a vedere che livello di accuratezza sono riusciti a raggiungere[Bahl and Padmanabhan [2]].

Per i test empirici si sono utilizzati i segnali ricevuti da 70 punti differenti, considerando le 4 direzioni di propagazione possibili (Nord, Sud, West, Ovest) quindi si sono considerate  $70(\text{posizioni}) * 4(\text{direzioni}) = 280$  combinazioni di posizione utente e direzione. Dopo di che è stata scelta una posizione con direzione random e poi è stata condotta una ricerca con l'algoritmo NNSS per poter confrontare le potenze ricevute dai restanti 69 punti della griglia ed effettuare una scelta cercando il punto con la minima distanza in potenza. Si sono confrontati il metodo empirico con altri due metodi: random selection e strongest base station selection . Con random selection, si stima la posizione dell'utente da uno dei 70 punti di prelievo in modo casuale, indipendentemente dalle informazioni SS. Con strongest base station selection, si individua la

posizione dell'utente in base alla stazione base che registra la potenza di segnale maggiore.



**Figura 1.7:** Funzione di Distribuzione Cumulativa (CDF) di errore nella stima della posizione

La Fig. 1.7 mostra la funzione di distribuzione cumulativa (CDF) della distanza di errore per tutti e tre i metodi, come possiamo facilmente vedere, il metodo empirico è quello che ci permette di avere una maggiore accuratezza.

Riportiamo nella Tab. 1.1 che segue, tutte le accuratezze facendo riferimento, nelle parentesi, alla percentuale peggiorativa degli altri metodi rispetto a quello puramente empirico. È importante considerare che l'accuratezza del primo metodo è di 2.94 metri, e tale valore è perfettamente assimilabile quindi alle dimensioni di una stanza.

L'utilizzo di un numero maggiore di punti di accesso sicuramente facilita il lavoro per un sistema di rilevamento della posizione, basato sulla creazione di un database, semplicemente perché aumenta la quantità di informazioni che possono essere ottenute con l'aggiunta di ogni punto di accesso. Il sistema sviluppato funziona quindi abbastanza bene con un numero minimo di punti

Metodo	25 <sup>th</sup> (metri)	50 <sup>th</sup> (metri)	75 <sup>th</sup> (metri)
<b>Empirical</b>	1.92	2.94	4.69
<b>Strongest</b>	4.54 (2.4x)	8.16 (2.8x)	11.5 (2.5x)
<b>Random</b>	10.35 (5.4x)	16.26 (5.5x)	25.63 (5.5x)

**Tabella 1.1:** Valori percentili dell'errore nel calcolo delle distanze

di accesso ed eccelle in prestazioni se utilizzato in un'infrastruttura con un gran numero di punti di accesso.

## 1.5 Bluetooth & Wi-Fi

Un interessante approccio è quello di Mahtab Hossain et al [Mahtab Hossain et al. [20]], in cui è stato studiato un sistema che mira a costruire sempre una radio-map, utilizza sempre il fingerprinting, ma sfruttando Wi-Fi con Bluetooth.

La scelta di combinare le due tecnologie è dovuto alla crescente diffusione dei loro punti di accesso, che si vorrebbero sfruttare al fine di aumentare le prestazioni del sistema. Per quanto riguarda la creazione del fingerprint, gli autori hanno selezionato la Signal Strength Difference (SSD), che è la differenza tra le potenze dei segnali che i due access point ricevono dallo stesso dispositivo mobile. Essi dimostrano sia analiticamente, sia sperimentalmente, che la SSD è più robusta rispetto RSS come localizzatore di posizione, perché essa non dipende dal dispositivo mobile che viene utilizzato.

In tale sistema è scelto un approccio basato sulla regressione lineare basata sull'interpolazione anziché un'indagine estenuante per creare il database, che



vuole affermare che un po' di punti di accesso buoni (ancore) sono meglio di molti punti di accesso con basso rendimento.

Hanno calcolato gli SSD utilizzando otto punti di accesso (quattro Bluetooth e quattro Wi-Fi) e considerando le coppie di punti di accesso della stessa tecnologia. I risultati indicano che la localizzazione basata solo su Bluetooth è più robusta rispetto al Wi-Fi e che le prestazioni dei quattro punti di accesso Bluetooth sono comparabile con il caso di tutti e 8 i punti di accesso sfruttati.

Tuttavia le prestazioni non sono così eccellenti: meno di quattro metri di distanza di errore con 80% di probabilità, ma vediamo nello specifico quanto è stato fatto.

Abbiamo già detto che un sistema di positioning indoor deve essere caratterizzato da un'elevata precisione, da una breve fase di preparazione, da un livello economico accettabile (preferibilmente utilizzando soluzioni hardware già presenti nel luogo da "mappare") e robustezza di fronte a tutte le condizioni precedentemente osservate. Questo lavoro mira a realizzare un sistema di localizzazione che possiede tutti questi requisiti. In futuro ci sarà una grossa crescita della ricerca di servizi di localizzazione per palmari e altri terminali mobili. Tuttavia, i palmari in generale presentano soluzioni hardware differenti, anche nella stessa tecnologia wireless. Di conseguenza un sistema di localizzazione che si basa esclusivamente sulla misurazione della potenza assoluta di segnale per definire la posizione non lavorerebbe bene. A prescindere che vengano utilizzati i valori di potenza del segnale di un dispositivo percepiti dall'access point per indicare le fingerprints della posizione del dispositivo, o che vengano utilizzati i valori di potenza percepiti dal dispositivo, le fingerprints possono differire significativamente in base all'hardware del dispositivo anche nelle stesse condizioni wireless. Questo può essere facilmente osservato nelle tecnologie wireless esistenti popolari, come il Bluetooth o Wi-Fi.

Ciò di cui ci occuperemo ora riguarda la robustezza del sistema; dimostriamo analiticamente che la differenza della potenza del segnale percepita sugli AP

fornisce una posizione più precisa rispetto ai valori di potenza assoluta trattati in RSS ad esempio.

Alcune opere precedenti hanno tentato di abbreviare la fase iniziale di un sistema di localizzazione. Si sostiene che, anziché eseguire un'indagine esaustiva per creare un database delle posizioni delle fingerprints che richiede costi di più elevati e manodopera, si potrebbe semplicemente raccogliere un numero limitato di letture. Haerberlen et al. [Haerberlen et al. [10]] raggiunge questo obiettivo suddividendo l'intera area in camere/cellule, limitando così il percorso secondo le stime per camera/livello. Al contrario, Li et al. [Li et al. [17]] tenta di completare il database utilizzando l'interpolazione di letture effettuate presso altri punti di monitoraggio.

Il lavoro fatto da Hossain et al. [Mahtab Hossain et al. [20]] ha adottato quest'ultimo approccio. Essi hanno ritenuto che basarsi sull'interpolazione può essere utile quando cambia l'ambiente o il programma di installazione. Normalmente in tali scenari i servizi di localizzazione possono essere sospesi durante l'attesa, per la creazione di un database appropriato che possa seguire il cambiamento. Questa procedura ha bisogno di tempo e di un intenso lavoro. L'amministratore del sistema di localizzazione può scegliere di continuare il servizio di previsione delle localizzazioni effettuando un'indagine grezza (tenendo alcuni campioni) nell'ambiente modificato o può completare i vuoti nel database con l'ausilio di tecniche basate su interpolazione. Il database può quindi essere ampliato in modo incrementale prendendo più campioni, fino a quando il sistema di localizzazione raggiunge una ragionevole accuratezza. Li et al. [Li et al. [17]] hanno utilizzato solo alcune linee guida intuitive per generare questi "falsi" punti di monitoraggio.

In questo documento, si sono serviti della regressione lineare ponderata al fine di ottenere un migliore adattamento per quei punti di monitoraggio fittizi sfruttando la somiglianza spaziale [Kaemarungsi and Krishnamurthy [13]] della distribuzione di potenza del segnale. Opere precedenti [Kaemarungsi and

Krishnamurthy [14]] hanno dimostrato che un aumento di AP non necessariamente aumenta la precisione, è importante soprattutto che gli access point siano “buoni” e in questo paper vengono descritti tali dispositivi come ancore.

Riportiamo i risultati ottenuti in base alle misurazioni fatte; si sono utilizzati due dispositivi differenti, un Laptop e un PDA, e si sono misurate le loro potenze di segnale sugli APs. Sono stati scelti 20 punti in maniera random e posizionato li quindi dei terminali per collezionare qualche campione sugli Aps per ogni dispositivo. Nelle Fig. 1.8 e Fig. 1.9 vediamo i valori RSS letti su un particolare AP mentre nelle Fig. 1.10 e Fig. 1.11 sono si evidenzia la differenza tra gli RSS visti tra due APs.

Si vede nella Fig. 1.8 e nella Fig. 1.9 che la potenza di segnale assoluta ricevuta su un certo AP è abbastanza differente per i due dispositivi. Questo ha ripercussioni sulla forma del fingerprint essendo abbastanza variabile quando sono usati durante il monitoraggio diversi dispositivi mobili. Al contrario SSD non soffre molto di questo problema, infatti garantisce un fingerprinting più robusto, come si vede nelle Fig. 1.10 e Fig. 1.11.

Come illustrato in Fig. 1.12 invece, si può vedere che un sistema di positioning basato su SSD batte in performance la controparte RSS nel caso di Bluetooth e Wi-Fi. Tuttavia si vede anche che Bluetooth SSD da' prestazioni migliori rispetto a Wi-Fi SSD. Ci sono 6 differenti coppie di 4 APs (per Bluetooth e Wi-Fi) in cui si possono dedurre 6 SSD per ogni posizione, ma in tutti gli esperimenti fatti si è visto che soltanto 3 valori sono sufficienti per avere simili performance per entrambi Bluetooth and Wi-Fi, che corrisponde ad un elemento in meno rispetto il proprio vettore degli RSS.

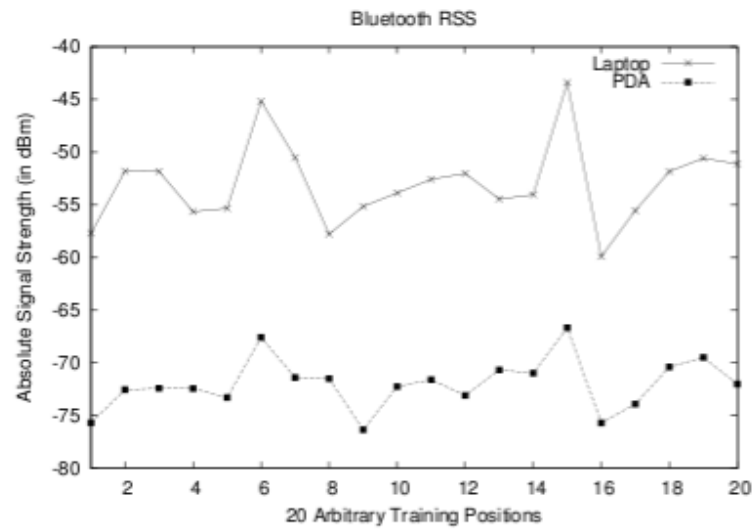


Figura 1.8: Absolute Signal Strength percepito su un AP Bluetooth (RSS)

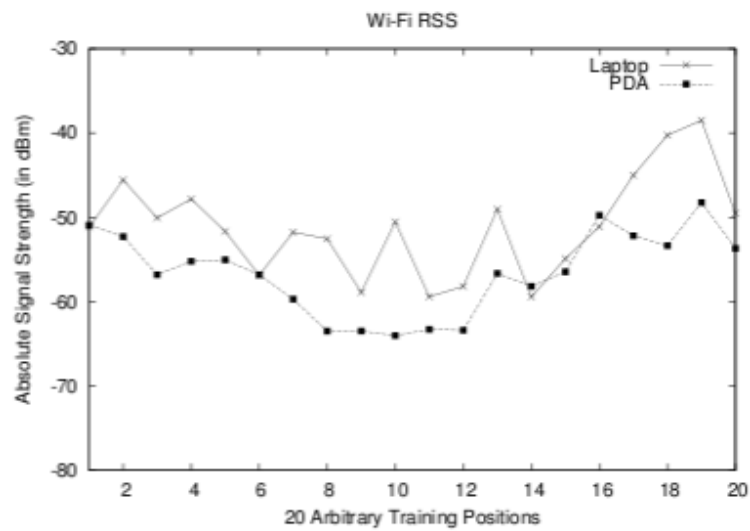


Figura 1.9: Absolute Signal Strength percepito su un AP Wi-Fi (RSS)

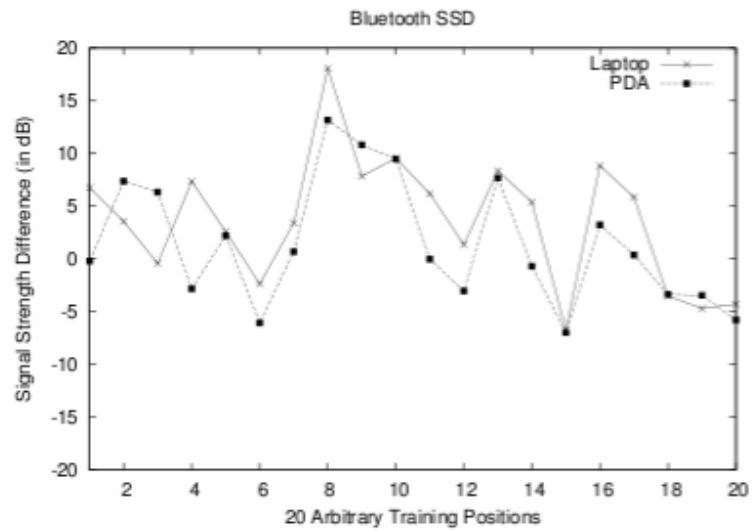


Figura 1.10: Signal Strength Difference tra 2 APs Bluetooth (SSD)

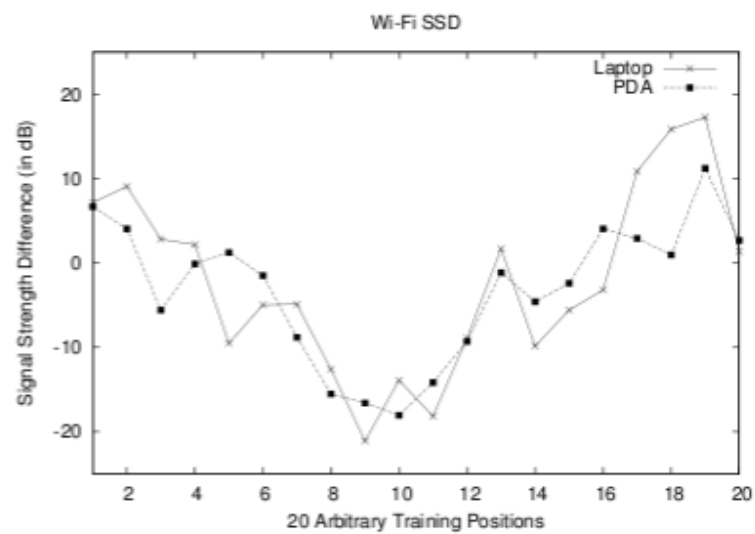
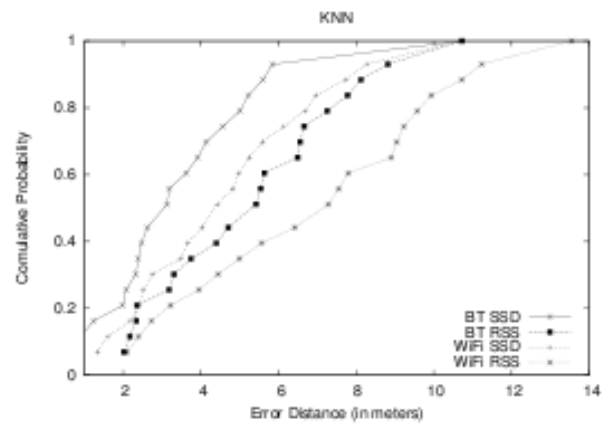
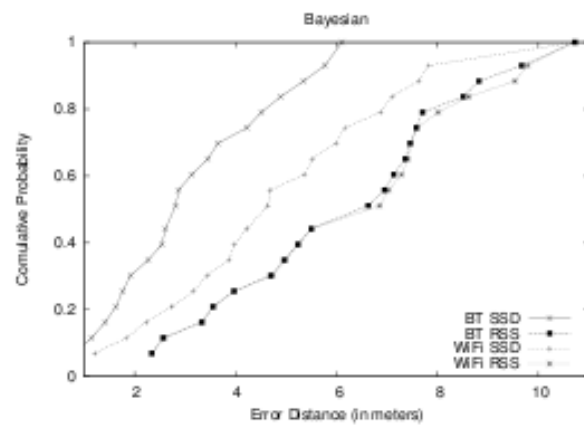


Figura 1.11: Signal Strength Difference tra 2 APs Wi-Fi (SSD)



(a) KNN's performance w.r.t. various location fingerprints



(b) Bayes' performance w.r.t. various location fingerprints

**Figura 1.12:** Comparazione tra SSD e RSS confrontando gli algoritmi KNN e di Bayes

## 1.6 Conclusioni

In base a questa ricca analisi fatta, come prima cosa viene da dire che dunque una possibilità di integrare la tecnologia bluetooth con Wi-Fi è ancora abbastanza lontana nonostante la maggiore accuratezza delle registrazioni SSD. Inoltre è ancora evidente che tra le due tecnologie RSS e TOA wi-fi based, quella che garantisce prestazioni migliori è sicuramente la prima e riportiamo brevemente i fattori che ci hanno orientato verso questa scelta.

1. Il fattore *COSTI* è l'unico che si avvicina al pareggio in quanto i costi economici delle tecnologie sono gli stessi, quello che varia è il costo computazionale, che risulta maggiore nei sistemi RSS a causa della prima fase di monitoraggio.
2. Per quanto riguarda l' *ACCURATEZZA* bisogna riconoscere che a causa delle imperfezioni dei clock dei normali dispositivi (abbiamo visto che un errore di 1 ms porta ad errori di localizzazione di circa 300 metri), TOA diventa praticamente quasi inutilizzabile senza un intervento tecnologico ad alto costo.
3. I dati ci hanno riportato anche grossi problemi riguardo la *SCALABILITÀ*, che vengono gestiti molto meglio dai sistemi RSS che non devono ogni volta stimare l'RTT.

Quello che farà parte delle nostre simulazioni, sarà dunque provare a migliorare il livello di precisione di un sistema RSS Wi-Fi based, che per quanto visto fino ad ora è ad un livello comunque abbastanza buono (2.94 metri).

---

## Positioning Indoor: sistemi basati su RFID

---

### Indice

---

<b>2.1</b>	<b>Introduzione . . . . .</b>	<b>32</b>
<b>2.2</b>	<b>Un primo accenno storico . . . . .</b>	<b>32</b>
<b>2.3</b>	<b>Componenti del sistema . . . . .</b>	<b>34</b>
2.3.1	TAG . . . . .	35
2.3.2	READER . . . . .	43
<b>2.4</b>	<b>Costi . . . . .</b>	<b>44</b>
<b>2.5</b>	<b>Positioning Indoor . . . . .</b>	<b>45</b>
2.5.1	SpotON . . . . .	45
2.5.2	Landmarc . . . . .	48
<b>2.6</b>	<b>Analisi progettuale. . . . .</b>	<b>52</b>
2.6.1	Introduzione: Cosa e Come. . . . .	52
2.6.2	Soluzione 1 . . . . .	53
2.6.3	Soluzione 2 . . . . .	56
<b>2.7</b>	<b>Conclusioni . . . . .</b>	<b>58</b>



---

## 2.1 Introduzione

Al contrario di quanto fatto in relazione alla tecnologia Wi-Fi, della quale non abbiamo dato alcuna informazione meramente tecnica, ma ci siamo interessati soltanto delle applicazioni riguardo il positioning, è interessante andare a introdurre con maggiori particolari quella che viene chiamata Radio Frequency Identification.

## 2.2 Un primo accenno storico

RFID (Radio Frequency IDentification) [P. Talone [22]] sta ad indicare la funzione di identificazione attraverso una trasmissione a radio frequenza. L'identificazione implica l'assegnazione di un'identità univoca ad un oggetto che consenta di distinguerlo in modo non ambiguo. Il fine principale di questa tecnologia, pertanto, è di assumere, da parte di un "identificatore", informazioni su oggetti, animali o persone identificati, per mezzo di piccoli apparati a radiofrequenza associati ai medesimi. L'assunzione di informazioni è relativa ad operazioni di ricerca, identificazione, selezione, localizzazione spaziale e tracciamento. Identificatore ed identificato comunicano mediante segnali a radio frequenza, quindi senza necessità di contatto fisico (a differenza, ad esempio, delle carte a banda magnetica) e senza che gli apparati siano né visibili (a differenza, ad esempio, dei codici a barre), né in visibilità reciproca (non-line-of-sight).

L'antenato degli RFID è comunemente riconosciuto nel sistema "Identification Friend or Foe (IFF)" sviluppato in Inghilterra durante la seconda guerra mondiale (1940). L'apparato a bordo degli aerei alleati, rispondeva, se inter-

rogato, identificando così gli aerei alleati e distinguendoli da quelli nemici. La tecnologia è poi evoluta in sistemi per seguire la rotta dei carri ferroviari, per l'automazione di processo nell'industria automobilistica, per la localizzazione del bestiame e degli animali selvatici, in agricoltura e nelle riserve naturali, per l'antitaccheggio nel commercio al minuto, per chiavi e documenti elettronici, ecc.

La diffusione dei sistemi RFID è avvenuta principalmente dagli anni '90 in poi; attualmente esistono numerose soluzioni commerciali a costi contenuti. Una tecnologia collaterale agli RFID e di grande rilevanza, è costituita dalle carte senza contatto che alle funzioni di identificazione aggiungono capacità di effettuare transazioni finanziarie in sicurezza. [fonte: libro bianco]

Molti pensano che l'RFID sia la tecnologia che permetterà di realizzare "l'Internet delle Cose" [RFID-Italia [24]], ovvero una grande rete in cui non soltanto le persone, ma anche gli oggetti saranno collegati fra di loro.

Per i ricercatori privati e le università, *l'RFID è una sfida*, in quanto nei prossimi anni ci sarà bisogno di progettare tag e lettori sempre più sensibili e intelligenti.

Per chi lavora negli ospedali, nelle pubbliche amministrazioni, nelle biblioteche, nella protezione civile, nell'esercito, *l'RFID è una promessa*. Questo perché grazie a questa tecnologia è possibile garantire più controllo, più sicurezza, più accuratezza nella gestione delle crisi.

Per i professionisti della supply chain, cioè per chi gestisce le merci, *l'RFID è uno strumento* utile oggi e indispensabile domani perché, insieme ad altre tecnologie, fornisce loro un controllo straordinario sui singoli prodotti. Con l'RFID possono ridurre i costi di gestione, aumentare l'efficienza dei magazzini, migliorare l'intera filiera.

Per i negozianti, *l'RFID è un'opportunità*; entro pochi anni permetterà di ridurre i furti nei punti vendita, ma soprattutto di migliorare la "shop ex-

perience”, ovvero il modo in cui i consumatori interagiscono con i prodotti che intendono acquistare.

Infine, per chi produce, integra e vende tecnologia, *l'RFID è uno dei mercati più promettenti del prossimo futuro*, perché i tag e le antenne, che sono i due elementi fondamentali di un sistema di identificazione a radiofrequenza, si stanno diffondendo in tutto il mondo (Italia compresa) con una rapidità sorprendente.

## 2.3 Componenti del sistema

La tecnologia RFID, che da qualche anno è stata oggetto di standardizzazioni ([EPCglobal-Inc. [6]], [RFID-Journal [25]]), si compone di tre elementi fondamentali:

- TAG  
⇒ Un transponder a radiofrequenza di piccole dimensioni costituito da un circuito integrato (chip) con funzioni di semplice logica di controllo, dotato di memoria, connesso ad un'antenna ed inserito in un contenitore o incorporato in una etichetta di carta, una Smart Card, una chiave. Il TAG permette la trasmissione di dati a corto raggio senza contatto fisico. Salvo eccezioni, i dati contenuti nella memoria del TAG sono limitati ad un codice univoco (identificativo).
- READER  
⇒ Un ricetrasmittitore controllato da un microprocessore ed usato per interrogare e ricevere le informazioni in risposta dai TAG.
- SISTEMA DI GESTIONE (Management system - Host system, etc...)  
⇒ Un sistema informativo che, quando esiste, è connesso in rete con i Reader. Tale sistema consente, a partire dai codici identificativi pro-



**Figura 2.1:** Esempio di TAG passivo

venienti dai TAG, di ricavare tutte le informazioni disponibili associate agli oggetti e di gestire tali informazioni per gli scopi dell'applicazione.

### 2.3.1 TAG

I TAG (chiamati anche Trasponder) vengono distinti, in primo luogo, per la gestione delle fonti energetiche. I TAG, infatti, possono essere divisi in due grandi famiglie:

- **Passivi**

In questo caso ricavano l'energia per il funzionamento dal segnale proveniente dal Reader; non possiedono un vero e proprio trasmettitore, ma reirradiano, modulandolo, il segnale trasmesso dal Reader e riflesso dalla propria antenna. Le distanze a cui possono operare sono, al massimo, dell'ordine di alcuni metri o di alcuni centimetri a seconda della frequenza operativa.

- **Attivi**

Sono alimentati da batterie. Incorporano ricevitore e trasmettitore come

i Reader. Possiedono memorie di dimensioni notevoli, spesso riscrivibili e possono contenere sensori. Le distanze a cui possono operare dipendono da trasmettitore e batterie, in genere sono, al massimo, dell'ordine di 200 metri.

I TAG passivi sono tipicamente dei dispositivi a basso costo e di piccole dimensioni che consentono di realizzare numerosi tipi di applicazioni. Essendo costituiti solamente da un'antenna (tipicamente stampata) e da un circuito integrato generalmente miniaturizzato, l'altezza dei TAG passivi può essere anche di poche centinaia di micron. I TAG, quindi, possono essere inseriti in carte di credito, etichette adesive, bottoni ed altri piccoli oggetti di plastica, fogli di carta, banconote e biglietti d'ingresso, generando così veri e propri oggetti "parlanti".

I TAG, inoltre, possono essere di tipo read-only o read-writable. Questi ultimi consentono, durante il loro uso, oltre alla lettura, anche la modifica o la riscrittura dell'informazione in essi memorizzata. In passato i TAG passivi erano principalmente di tipo read-only sia perché la fase di scrittura richiede la disponibilità di una quantità elevata di energia che si ricava con difficoltà dal segnale ricevuto, sia perché le memorie riscrivibili hanno un costo relativamente elevato. I TAG passivi riscrivibili sono comunque in rapida diffusione.

I TAG passivi non possono iniziare la comunicazione ma possono solamente essere interrogati. Per i TAG attivi o semi passivi, oltre alla maggior quantità di memoria ed alla funzione di riscrivibilità della stessa, l'evoluzione tecnologica ha consentito di aggiungere, in alcuni casi, funzioni che superano di gran lunga la pura identificazione. Si ricordano, ad esempio, le funzioni di radiolocalizzazione (RTLS - Real Time Location System - identificazione della posizione dell'oggetto che contiene l'RFID) o la misura di parametri ambientali attraverso sensori (temperatura, movimento, etc.).

Quando il TAG passa attraverso il campo elettromagnetico (EM) generato da un Reader, trasmette a quest'ultimo le proprie informazioni. Tipicamente un TAG passivo che riceve il segnale da un Reader usa l'energia del segnale medesimo per alimentare i propri circuiti interni e, di conseguenza, "svegliare" le proprie funzioni. Una volta che il TAG ha decodificato come corretto il segnale del Reader, gli risponde riflettendo, mediante la sua antenna, e modulando il campo emesso dal Reader. I protocolli di comunicazione tra Reader e TAG sono descritti in appositi standard.

Le informazioni che il TAG trasmette al Reader sono contenute in una certa quantità di memoria che ogni TAG contiene al suo interno. Le informazioni d'identificazione sono relative all'oggetto interrogato: tipicamente un numero di serie univoco, spesso una estensione dell'UPC (Universal Product Code) contenuto nel codice a barre ed altre informazioni (date di produzione, composizione dell'oggetto, etc.). Normalmente la quantità di dati contenuti in un RFID è piuttosto modesta (centinaia di byte o, al massimo qualche KByte). Ciò nonostante, la pervasività dell'uso dei TAG e di opportune tecniche a radiofrequenza che consentono di interrogare e ricevere risposte da tutti i TAG presenti in un particolare ambiente possono portare ad una "esplosione" della quantità di dati circolanti.

Vediamo nella Tab. 2.1 le principali caratteristiche che differenziano le due grandi famiglie di tag.

Senza soffermarci troppo su questi particolari, per le considerazioni circa il nostro progetto è necessario inserire alcune caratteristiche che riguardano il ruolo che ha la frequenza di funzionamento di questi dispositivi sull'utilizzo del dispositivo stesso.

Distinguiamo [Ibjesse [11]]:

1. Gamma di Frequenza LF (100-500 KHz)  
⇒ Frequenza Tipica: 125 KHz

	<b>Active TAG</b>	<b>Passive TAG</b>
<b>Tag Power Source</b>	Internal to tag	Energy transferred from the reader via RF
<b>Tag Battery</b>	Yes	No
<b>Avaiability of Tag Power</b>	Continuos	Only within field or reader
<b>Required Signal Strength from Reader to Tag</b>	Very Low	Very High (must power the tag)
<b>Avaiable Signal Strength from Tag to Reader</b>	High	Very Low
<b>Communication Range</b>	Long range (100m or more)	Short or very short range (3m or less)
<b>Sensor Capability</b>	Ability to continuously monitor and record sensor input; data/time stamp for sensor events	Ability to read and transfer sensor values only when tag is powered by reader; no date/time stamp
<b>Data Storage</b>	Large read/write data storage (128KB) with sophisticated data search and access capabilities avaiable	Small read/write data storage (e.g. 128 bytes)

**Tabella 2.1:** Caratteristiche dei TAGs

Abbastanza costoso specie per elevate quantità di pezzi. Le basse frequenze richiedono un'antenna più costosa. Poco sensibile alla degradazione delle performance ad opera di liquidi e metalli, ciò nonostante la distanza di lettura è molto breve

2. Gamma di Frequenza HF

⇒ Frequenza Tipica: 13.56 MHz

Meno costosi dei tag induttivi LF. Distanza di lettura e velocità di trasferimento dati abbastanza modesti, se paragonata ai tag UHF. Particolarmente indicati per le applicazioni che non richiedono lettura a grande distanza di numerosi tag.

3. Gamma di Frequenza UHF (852-926 MHz)

⇒ Frequenza Tipica: 865-868 Mhz EU - 915 Mhz USA

In elevate quantità, i tag UHF hanno la potenzialità di diventare meno costosi di quelli LF e HF grazie ai recenti miglioramenti nella progettazione dei relativi chip.

4. Gamma di Frequenza Microonde

⇒ Frequenza Tipica: 2.45 Ghz / 5.8 Ghz

Caratteristiche simili ai tag UHF ma con migliori prestazioni. Un problema di questa banda di frequenze è la più elevata sensibilità alla degradazione del segnale in presenza di liquidi o metalli in particolare. Offre un segnale molto più direzionale, ideale per alcune applicazioni

Nel particolare analizziamo UHF MEDIA che abbiamo scelto per un nostro eventuale progetto. L'evoluzione tecnologica dei semiconduttori, che ha



portato alla realizzazione di chip particolarmente parsimoniosi nel consumo energetico, ha consentito la realizzazione di etichette RFID operanti a questa frequenza e con distanza operativa decisamente più estesa di quanto non fosse consentito con LF ed HF. L'accoppiamento Reader-TAG avviene per via elettromagnetica, come nei tradizionali sistemi di radiocomunicazione. Una distanza operativa di 3 metri è ormai standard, ma sempre più spesso estendibile verso cinque e più metri. Grazie a questo l'UHF media è destinata sicuramente a confermarsi come la banda regina della logistica e, soprattutto, della gestione degli oggetti. Tuttavia alcune problematiche, ad oggi in via di risoluzione, ne hanno rallentato l'introduzione.

- *Frequenze operative:*

Usa, Europa e Asia si trovano a dover gestire frequenze diverse: le frequenze già occupate dalla telefonia cellulare, e quindi ormai immutabili, non consentono alle tre aree geografiche di utilizzare le stesse bande per le applicazioni RFID. Tuttavia i TAG passivi vengono spesso costruiti con accorgimenti che ne esaltano la capacità di rispondere a "larga banda", il che ne consente l'operatività su bande differenti (purché non troppo) al costo di un decadimento nelle prestazioni.

- *Standard di comunicazione:*

Sono necessari standard accettati dalla comunità internazionale per i protocolli di comunicazione tra Reader e TAG. Il problema è stato superato in tempi recenti dalla definizione del protocollo EPC "Class 1/Generation2" e dalla relativa inclusione nello standard ISO/IEC 18000-6 Type C avvenuta nel luglio del 2006.

- *Potenze in trasmissione (dei Reader) e larghezza delle bande di frequenza:*

In Usa ed in Europa esistono differenti limitazioni per la potenza massima emessa e, soprattutto, differenti larghezze della banda di frequenza UHF dedicata; ovvero esiste un maggior numero di canali sui quali i READER possono operare per interrogare i TAG. In logistica, questo si traduce (per gli USA) in un vantaggio competitivo (es. capacità di leggere rapidamente tutto il contenuto di una pallet) e di conseguenza un minore costo del servizio. Ciò avviene perché, potendo usufruire di un numero maggiore di canali, è possibile far operare contemporaneamente (nella stessa area) un maggior numero di Reader (vedere su LIBRO BIANCO algoritmo “Frequency Hopping”) inoltre ciascun Reader opera a potenza leggermente maggiore (maggiore distanza operativa).

A queste frequenze (accoppiamento elettromagnetico) la quantità di energia che il TAG raccoglie dal campo EM ricevuto, è funzione del diagramma di radiazione dell’antenna del TAG e della densità di potenza del campo EM presente nella posizione del TAG. Il diagramma di radiazione dell’antenna del TAG è correlato con la lunghezza d’onda del segnale e con le dimensioni fisiche dell’antenna medesima. La densità di potenza è correlata con i parametri tecnici del Reader (potenza, antenna).

Se consideriamo la frequenza di 900 MHz la lunghezza d’onda è circa 33 cm e un’antenna a 1/2 onda risulta lunga 16.5 cm, mentre un’antenna ad 1/4 d’onda risulta di circa 8,3 cm. Queste sono le dimensioni più probabili che ci si può aspettare per i TAG operanti in UHF media.

Essendo l’UHF media, grazie alla sua distanza operativa, la frequenza di elezione per la logistica, i rispettivi TAG dovrebbero arrivare, grazie all’esplosione attesa nei volumi di produzione, ad avere il costo più basso. La velocità di trasmissione risulta superiore a quella dei sistemi operanti a frequenze più basse. I sistemi, inoltre, sono in grado di gestire letture multiple contemporanee (anticollisione) arrivando alla lettura di più di 100 TAG al secondo. Sia le caratteristiche tecnologiche,

sia le caratteristiche dell'algoritmo di anticollisione (**devo specificare qualcosa di più riguardo tali caratteristiche???**) di TAG conformi alle specifiche EPC/ISO Class1/Gen2 dovrebbero consentire, in linea teorica, la lettura di 600 (in Europa) e 1.500 (in USA) TAG/s che si presentino contemporaneamente al lettore. Il supporto da parte dei fornitori di tecnologia si va facendo via via più consistente, con un sempre maggior numero di fornitori e con ampliamento della possibilità di scelta sia sulle capacità di memoria dei chip che sulla possibilità di avere TAG passivi, attivi o semipassivi. Quanto detto è ripreso e riportato nella Tab. 2.2.

Accoppiamento induttivo/campo M		Accoppiamento con campo EM (backscatter)	
LF	HF	UHF media	UHF alta
Sensibilità ad orientamento ortogonale dell'antenna			
<b>Molto sensibili</b> (mancate letture)		<b>Sensibili</b> (Attenuazione e possibili mancate letture)	
		<b>Poco sensibili</b>	
Velocità di trasferimento dei dati (TAG - READER)			
Minore	←————→		Maggiore
Velocità di lettura di gruppi di TAG			
Minore	←————→		Maggiore
Capacità di lettura di TAG in prossimità di metalli o superfici bagnate			
Migliore	←————→		Peggior
Assorbimento da parte di liquidi, ovvero Capacità di lettura di TAG all'interno degli animali o coperti da acqua o ghiaccio			
Migliore	←————→		Peggior
Dimensione dei TAG passivi			
Dipende dall'energia del captabile		Dipende da frequenza e tipo di antenna	
←————→		Grande	←————→
		Piccola	
Assorbimento di energia			
Minore	←————→		Maggiore

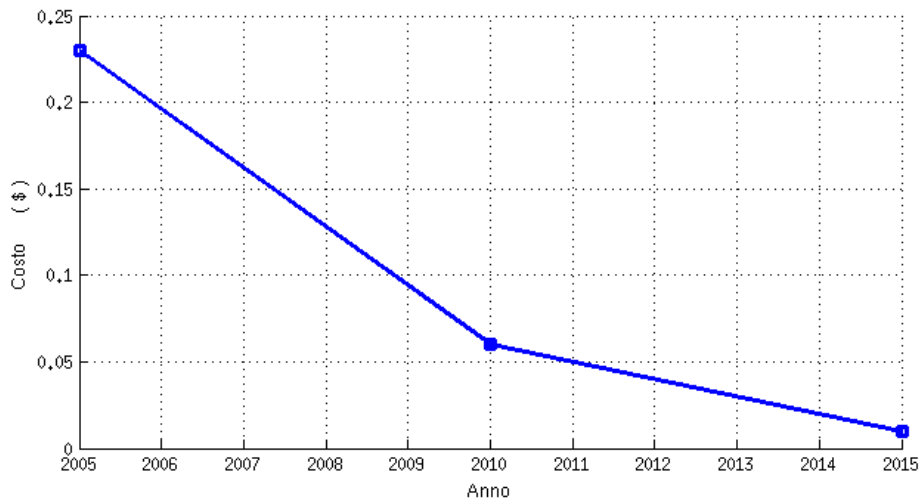
**Tabella 2.2:** Prestazioni dei TAGs

### 2.3.2 READER

Il Reader (chiamato anche “interrogator” o “controller” se distinto dalla sua antenna) è l’elemento che, nei sistemi RFID, consente di assumere le informazioni contenute nel TAG. Si tratta di un vero e proprio ricetrasmittitore, governato da un sistema di controllo e spesso connesso in rete con sistemi informatici di gestione per poter ricavare informazioni dall’identificativo trasmesso dai TAG. Questo, infatti, specie nei TAG passivi, è un semplice codice che ha però (a differenza dei codici a barre) la particolarità di essere univoco. Entrando quindi in un sistema informativo ed usando un codice univoco come chiave di ricerca, si possono ricavare dettagliate informazioni (anche aggiornate nel tempo) sul particolare oggetto a cui il TAG è associato.

I Reader per TAG attivi sono dei ricetrasmittitori controllati, che possono usare le più diverse tecniche a radiofrequenza. I TAG attivi, ad oggi, sono solo in piccola parte coperti da standard specifici. I Reader per TAG passivi (e semi passivi), invece, devono emettere segnali RF di tipo particolare, in grado di fornire al TAG anche l’energia necessaria per la risposta. Le tecniche di comunicazione e trasferimento dati utilizzate nei sistemi RFID sono molto diverse tra loro per applicazioni che possono andare dalla lettura, a pochi centimetri di distanza, di TAG passivi, fino a prevedere letture di TAG attivi a distanza di parecchie centinaia di metri.

Esistono Reader fissi (montati ad esempio sui portali di accesso ai magazzini, sui nastri trasportatori, negli scaffali, etc.) e Reader portatili (una sorta di “pistole elettroniche” esteticamente simili a quelle in uso per i codici a barre. Per quanto riguarda la diffusione, le installazioni di Reader fissi rappresentano (compresi quelli per smart card), ad oggi, più dell’80% del totale a livello mondiale. Per quanto riguarda l’uso delle frequenze, la maggior parte del mercato dei Reader, sia in termini di fatturato, sia in termini di numero di unità è rappresentato da Reader HF. Tuttavia si prevede che il mercato dei Reader UHF cresca molto più velocemente degli altri nel prossimo futuro.



**Figura 2.2:** Grafico relativo ai costi della tecnologia RFID

## 2.4 Costi

Per terminare la trattazione teorica, introduciamo ovviamente la questione relativa ai costi di questa tecnologia. Il costo dei TAG passivi è ritenuto il principale fattore abilitante per una diffusione massiva della tecnologia RFID nella catena di distribuzione. Per quanto riguarda i costi “minimi”, un traguardo importante è considerato il raggiungimento del costo di 5 cent/TAG per le “etichette intelligenti” da applicare ai singoli oggetti. Nella realtà, un’analisi dei costi dei TAG [Wyld [32]] porta a quanto illustrato in Fig. 2.2 in cui il grafico illustra una previsione sul costo dei TAG passivi a basso costo (“etichette intelligenti”) nei prossimi anni.

Il “mitico” traguardo di 5 cent/TAG sembrerebbe quindi raggiungibile proprio intorno al 2010, questo è effettivamente vero nel caso in cui venga però acquistata una quantità abbastanza grande di dispositivi. Nelle ipotesi di costo unitario di cui sopra le previsioni sulla diffusione dei TAG (della medesima fonte) parlano dell’inizio di una vera diffusione pervasiva appena dopo il 2010.

## 2.5 Positioning Indoor

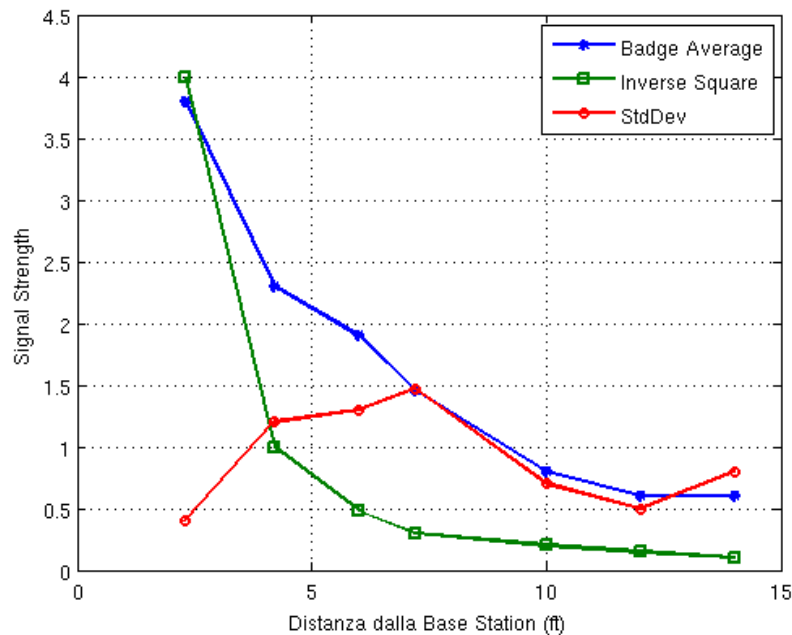
Una volta conosciuta la tecnologia RFID, il nostro obiettivo è quello di analizzare in che modo sarebbe possibile utilizzarla per i nostri scopi di *Positioning Indoor*. Per l'analisi delle prestazioni ci siamo basati sui due sistemi più importanti citati in letteratura: *SpotON* e *Landmarc*.

### 2.5.1 SPOTON

**SpotON** [Jeffrey [12]] è un sistema di “location sensing” in ambiente indoor 3D basato sulla potenza del segnale radio-frequenza (RF Signal Strength). Prima di addentrarci nel progetto è necessario fare una piccola distinzione tra le due definizioni di positioning e tracking (monitoraggio). I sistemi di posizionamento sono quelli che forniscono i mezzi per determinare la posizione e lasciare che sia l'utente stesso a calcolare la propria posizione attuale. I sistemi di inseguimento invece, sono quelli che controllano gli oggetti senza coinvolgere gli oggetti monitorati nel calcolo.

Molti sistemi sono un pò un ibrido di questi due tipi, sia per design che semplicemente nella configurazione, spesso nell'interesse della sicurezza e della privacy. In questo primo lavoro è stata applicata la “location sensing”, termine dato appunto a tali sistemi ibridi, compreso questo, dove la politica di manipolare i dati relativi all'ubicazione cerca di essere separata dal meccanismo di individuare effettivamente l'oggetto.

Nei test effettuati si sono serviti di dispositivi già presenti in commercio, la loro intenzione non era infatti quella di perdere tempo a costruire un nuovo hardware potendone trovare degli altri già pronti. Si sono serviti di una nota casa produttrice nel settore, “RFIDeas” [RFIDeas [26]] e dei suoi prodotti AIR ID: un “adjustable long range active ID badge, reader and software solution for desktop computers”.



**Figura 2.3:** Risultati ottenuti con SpotON

Le capacità di questi dispositivi di fornire e registrare effettivamente la giusta potenza di segnale sono state testate in un grande spazio chiuso all'interno del quale è stata posizionata una base-station nel punto centrale e 12 badge sistemati in cerchi concentrici di raggi noti; le misurazioni sono poi state effettuate spostando in postazioni differenti i vari badge. A scopo illustrativo riportiamo il grafico in Fig. 2.3 che rappresenta i risultati a cui sono arrivati Hightower, Borriello e Want, in cui “Badge Average” rappresenta il livello di potenza pervenuto al variare della distanza dei badge.

Senza scendere troppo nei particolari riguardo la tecnologia e l'hardware, quello che vogliamo illustrare sono l'implementazione e i risultati. L'algoritmo utilizzato è un “Aggregation Algorithm”[Jeffrey [12]] che lavora pressappoco come quello visto per i sistemi di positioning nel WI-Fi (NNSS) e che cerca di calcolare il Signal-Strength cercando di minimizzare l'errore relativo ai dati empirici.

Questo prototipo di sistema di localizzazione AIR ID può determinare la posizione di tag in modo semi-affidabile. Tuttavia, le limitazioni sono significative.

- In primo luogo, anche se il sistema è certamente adeguato per alcune applicazioni, l'accuratezza complessiva è molto più povera di quanto realmente desiderato: gli oggetti possono essere fissati nello spazio con un'accuratezza di non meno di 3 metri. Pertanto, per piccole stanze, questo sistema non è migliore di un sensore di movimento rudimentale. Particolare sicuramente non trascurabile è che nell'anno in cui sono stati fatti i test venivano sono stati utilizzati soltanto 2 bit di precisione per la memorizzazione dei Signal-Strength, le nuove generazioni di prodotti RFIDEas invece sono arrivati ora a 8-bit di precisione.
- Il secondo problema è la frequenza di misurazione. Poiché il protocollo AIR ID non è progettato per l'attività di rilevazione del percorso, anche utilizzando le API in modo efficiente su un server multithread richiederebbe ancora 10 a 20 secondi per prendere una misura di posizione da tutte le basestation. È chiaramente molto facile, anzi troppo facile, perdere significativi cambiamenti nella posizione dei tag con una frequenza di campionamento così lenta.

Il perché di tali limitazioni in questo progetto è molto facile da spiegare; ovviamente si sapeva già dall'inizio che i due bit di precisione non avrebbero portato a risultati eccelsi però questo progetto voleva essere soltanto un punto di partenza per lo sviluppo di nuovi hardware, quindi la scelta di quei dispositivi AIR ID è stata fatta esclusivamente per ragioni di costo.

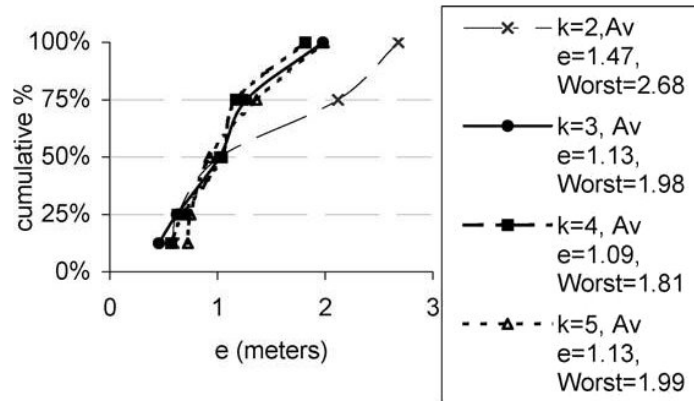


### 2.5.2 LANDMARC

Una più interessante proposta RFID-based resta invece **LANDMARC** [Ni et al. [21]] (**LocAtioN iDentification based on dynaMic Active Rfid Calibration**), un altro prototipo di sistema che ha come obiettivo quello di ridurre il numero di readers. Utilizza Tags e Readers di tipo attivo prodotti dalla “Spider System”; ogni tag ha un range di lettura di circa 45 metri (che possono però essere incrementati fino a circa 300 con antenne specifiche). In questo sistema i readers lavorano alla frequenza di 308 MHz e utilizzano un’interfaccia 802.11b per comunicare tra di loro. Viene presentato un algoritmo che utilizza appunto readers attivi e tag attivi, questi ultimi identificati come “reference tags” che determinano la posizione di tag passivi (tracking tags); in pratica quindi i reference tags servono come punti di riferimento all’interno del sistema. Questo approccio ha tre principali vantaggi:

1. Non aver bisogno di un numero troppo elevato di readers.
2. Essere un sistema facilmente adattabile alle variazioni ambientali in quanto le informazioni di riferimento sui tag attivi possono essere aggiornate dinamicamente.
3. La possibilità di avere informazioni sulla posizione più precise e affidabili (ovviamente è molto importante la collocazione dei reader e dei reference tags).

Supponendo di avere “n” RF readers, “m” Reference Tags e “u” Tracking Tags, in questo sistema viene utilizzato l’algoritmo NNSS già visto in precedenza nei sistemi Wi-Fi quindi, si definisce un Signal-Strength Vector per ogni tracking tag:  $S = (S_1, S_2, \dots, S_n)$  in cui  $S_i$  è la potenza registrata sul reader  $i$ -esimo, si definisce anche per ogni reference tag un vettore  $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ ,



**Figura 2.4:** Errore percentile cumulativo del calcolo della distanza per  $k \in (2, 5)$ .

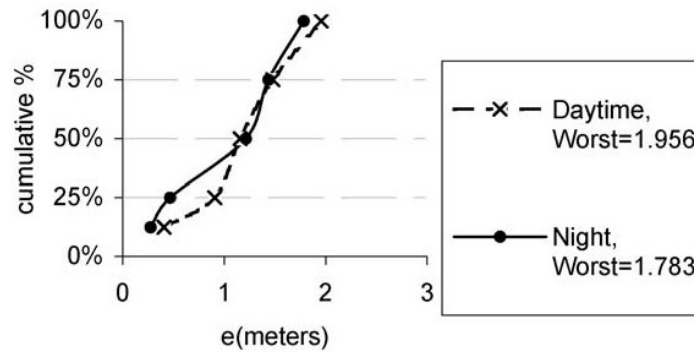
dopo di che per ogni tracking tag si calcola la distanza Euclidea:

$$E_j = \sqrt{\sum_{i=1}^n (\theta_i - S_i)^2} \quad \text{con } j \in (1, m).$$

Quindi dopo aver riportato tutti i valori nel vettore  $E = (E_1, E_2, \dots, E_m)$ , si scelgono le  $K$  distanze minime, che ovviamente rappresentano i “vicini” più appetibili, e a quel punto si approssima la posizione del tracking tag con la nota formula:

$$(x, y) = \left(\frac{1}{N}\right) * \left(\sum_{i=1}^K (x_i, y_i)\right).$$

Andiamo a riportare i risultati ottenuti da Lionel et al.[Ni et al. [21]]. Il loro esperimento si è avvalso di  $n = 4$  readers,  $m = 16$  reference tags e  $u = 8$  tracking tags. L’accuratezza e la precisione hanno subito delle variazioni in base a diverse specifiche come ad esempio il numero di “vicini” che venivano scelti, e dopo una serie di prove si è evinto che con  $k = 4$  neighbors (Fig. 2.4) si arriva ad avere un errore massimo di circa 1 metro, che è abbastanza incoraggiante in quanto supera anche i 2.37 metri di errore avuti con RADAR.



**Figura 2.5:** Errore percentile cumulativo del calcolo della distanza durante il giorno e la notte.

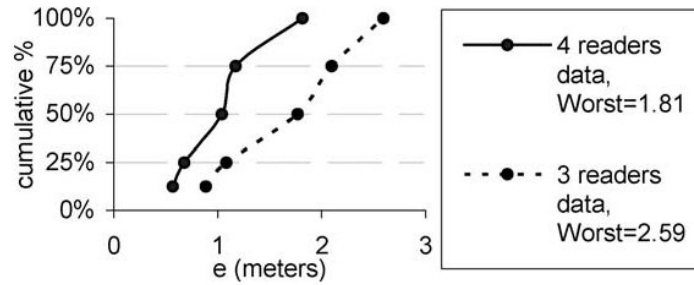
Diverse prove sono state fatte in diversi periodi della giornata e si è visto che durante il giorno in cui lo spazio adibito agli esperimenti (laboratorio) era maggiormente affollato, venivano registrati risultati peggiori (Fig. 2.5), quindi il numero di persone influenza le registrazioni delle potenze.

Anche il numero di readers (Fig. 2.6), come potevamo facilmente aspettarci, fa variare notevolmente l'errore nella localizzazione.

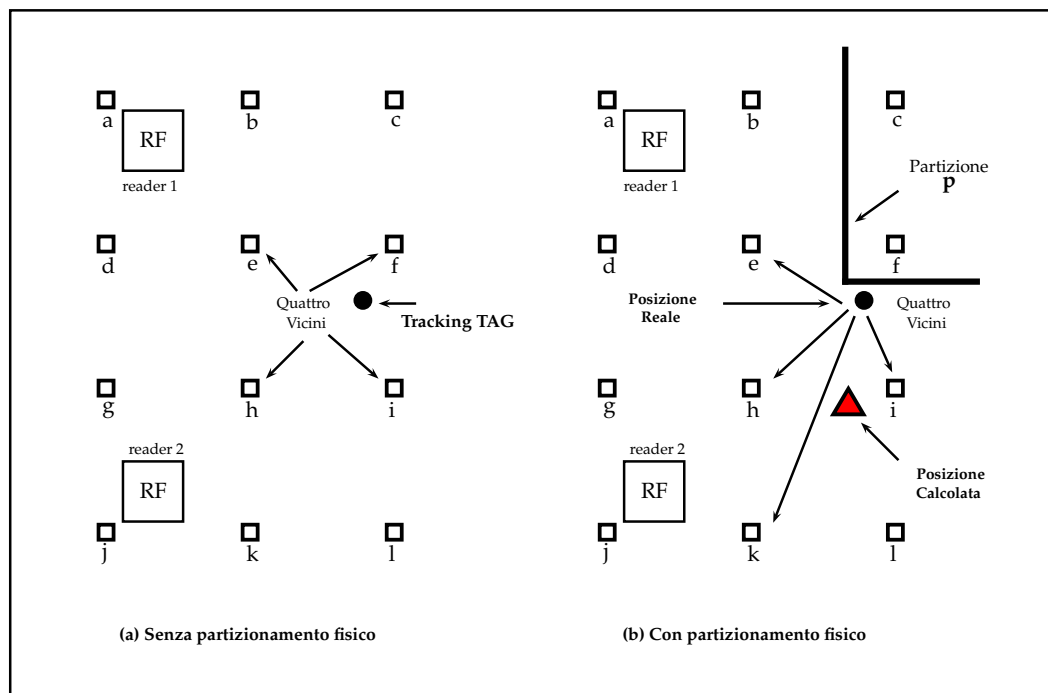
Infine come si vede in Fig. 2.7, lo spostamento dei reference tags può causare notevoli cambiamenti nei risultati, quindi la loro localizzazione deve essere oculata e precisa.

Dopo aver esposto i risultati ottenuti, aggiungiamo che questo prototipo ha 3 problemi importanti.

1. Nessuno degli attuali prodotti RFID calcolava direttamente il Signal Strength dei tag, quindi ogni reader impiegava circa 1 minuto per stimare tale potenza;
2. Il lungo periodo di latenza porta quindi a calcolare la posizione effettiva



**Figura 2.6:** Errore percentile cumulativo del calcolo della distanza al variare del numero dei readers da 3 a 4.



**Figura 2.7:** Esempio di partizionamento fisico per separare i reference tags 'c' ed 'f' dagli altri

del tracking in un momento successivo a quello reale;

3. La variazione del comportamento dei tags.

## 2.6 Analisi progettuale.

Dopo aver trattato in maniera sufficientemente ampia tutto ciò che riguarda la tecnologia “**R**adio **F**requency **I**Dentification” e gli studi fatti sulla stessa, un nostro obiettivo è stato quello di valutare un interessante progetto che potesse riguardare il *positioning indoor*, considerando fattori positivi e negativi, e ovviamente i costi di gestione.

### 2.6.1 INTRODUZIONE: COSA E COME.

La nostra idea è di creare un complesso ma funzionale sistema di positioning indoor che possa dare utili informazioni all’utente non soltanto per quanto riguarda la posizione (tracking real time).

Vogliamo provare a creare un sistema di POSITIONING INDOOR all’interno del primo piano della facoltà di ingegneria ‘de “La Sapienza” di Roma, sfruttando le caratteristiche della tecnologia RFID.

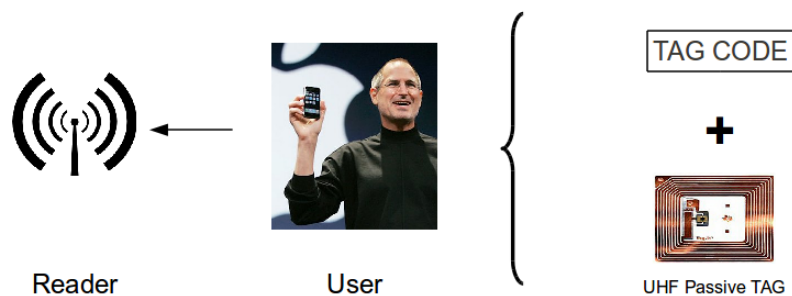
L’idea è di dare all’utente la possibilità di utilizzare il proprio dispositivo mobile (PC, smartphone, Iphone) per essere identificato, localizzato e per ricevere il maggior numero di informazioni possibili (Percorsi, uscite, WC, orario lezioni, impegni aule, etc...) senza ricorrere al GPS che, come detto più volte, in spazi chiusi non può essere utilizzato.

Dopo aver contattato diverse aziende che operano nel settore (FEIG Electronic, Wireless Dynamics, Motorola, Skyetec, etc...), in base alle nostre esigenze e per la comodità di ogni utente, abbiamo scelto di utilizzare per i nostri scopi dispositivi di tipo UHF in quanto noi vogliamo un range di qualche metro

per le nostre operazioni, dunque avere tali prestazioni utilizzando bande HF (13.56 MHz) avrebbe bisogno di tecnologie più avanzate che equivarrebbero a una crescita ancora maggiore del costo di produzione.

### 2.6.2 SOLUZIONE 1

Una **prima soluzione** (Fig. 2.8) sarebbe quella di posizionare un certo numero di READER all'interno del piano, dopo di che ogni utente all'ingresso della facoltà dovrebbe prendere un TAG, fotografare il codice stampato su di esso e conservare il circuito. In questo modo il Reader utilizzerebbe il circuito per la localizzazione dell'utente (tramite RFID) e il TAG code per comunicare con il dispositivo mobile (tramite Wi-Fi).



**Figura 2.8:** Positioning Indoor: Soluzione 1

Il nostro obiettivo è ovviamente quello di creare un sistema funzionale, ma che possa essere allo stesso modo un sistema a basso costo. Per questa ragione avevamo deciso di lavorare con i TAG di tipo passivo nelle bande UHF.

Allora, in base a quanto detto, i nostri dispositivi sarebbero:

- TAG Passivi UHF;

- READER MRU80-M2-U ReadWave Short Range UHF [RF-ID [23]](Fig. 2.9a);
  - Frequenza di lavoro: 860 - 960 MHz.
  - Alimentazione: via USB (5 Volt).
  - Range di lettura: fino a 200 cm.
  - Dimensioni: 93x62x27 mm.
  
- Antenne [RF-ID [23]](Fig. 2.9b);
  - 2 x ISC.ANT.U160/160-EU Antenna UHF.
  - Circular polarized Panel
  - Antenna. Gain: 4,0 dBic a 866 MHz.
  - Dimensioni: 162x162x16 mm.
  
- - Librerie SDK per ambiente MS Window.



(a) *Reader.*

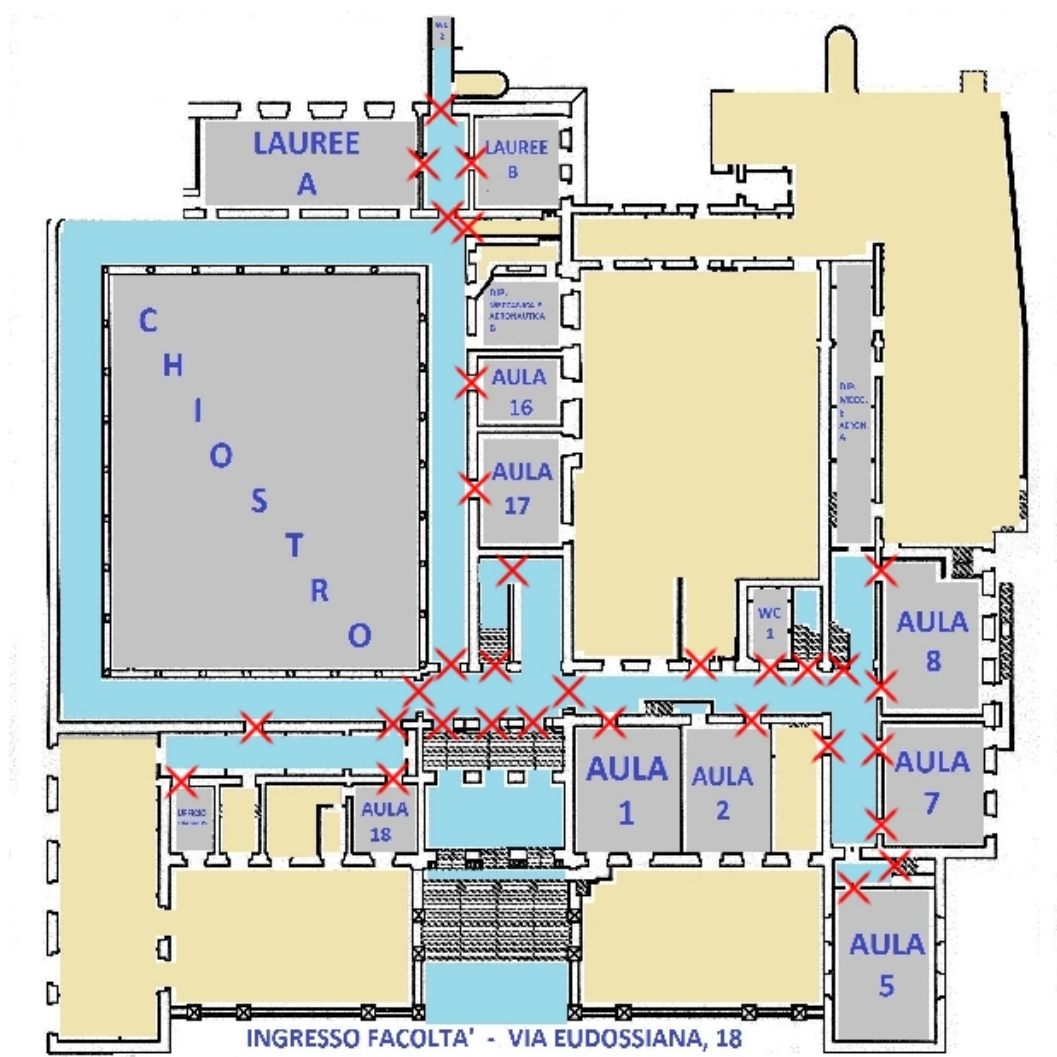


(b) *Antenna.*

**Figura 2.9:** Componenti del sistema (Soluzione 1)

Per poter effettuare un'analisi che ci permetta di valutare se il nostro sistema sia utilizzabile o meno, abbiamo dovuto fare un'attenta analisi dei *costi*. Per fare ciò è stata necessaria una stima del numero di dispositivi necessari, almeno per un primo test. Sulla mappa in Fig. 2.10 abbiamo provato a sistemare (con una X) un READER per ogni punto di interesse (ingressi, aule, scale, WC,

etc...); è importante avere un READER su ogni possibile “via di fuga” per non perdere mai il contatto con l’utente. Avremmo inoltre bisogno di un adattatore Wi-Fi su ogni reader in modo che questo possa comunicare con il dispositivo mobile corrispondente al TAG intercettato.



**Figura 2.10:** San Pietro in Vincoli: Mappa del primo piano

Il numero di Reader stimato è 32, ma avendo a disposizione 2 antenne per ogni reader a distanza massima di 6 metri tra loro, forse potremmo eliminarne



qualcuno. Allora il costo totale di ogni dispositivo, considerandone un numero pari a 30, si aggirerebbe intorno ai 650 € (Il costo dei TAG rientra in quello dei Reader) dunque il costo totale del progetto sarebbe di circa 25.000 € (iva compresa), come si vede nel prospetto in Tab. 2.3, quindi ancora troppo lontano dai nostri obiettivi.

<b>1 Reader + 2 Antenne</b>	€500
<b>Adattatore Wi-fi</b>	€100
<b>Cavi Collegamenti</b>	€50
<b>Costo singolo dispositivo</b>	€650

**Tabella 2.3:** Prospetto Costi: Soluzione 1

### 2.6.3 SOLUZIONE 2

A questo punto una **seconda soluzione** potrebbe essere quella di utilizzare un READER con un range maggiore e che ci possa dare la possibilità di posizionare le antenne a distanze maggiori (10 + 10 metri). Esistono dispositivi del genere, noi abbiamo analizzato l'MRU200-E LAN UHF Industrial Mid Range Controller [RF-ID [23]]; le caratteristiche tecniche (a parte il range di lettura che dipende dalla potenza di trasmissione e varia da 1 metro (0,1 W) a 3 metri (0,3 W)) sono le stesse del dispositivo precedente, con 2 differenze importanti, che riportiamo esplicitamente in Tab. 2.4:

<b>Reader</b>	€1000	⇒ <b>Circa il doppio</b>
<b>Dimensioni Antenna</b>	270x270x57 mm.	⇒ <b>Dim. Eccessiva</b>

**Tabella 2.4:** Prospetto Costi: Soluzione 2

In base alle considerazioni fatte circa la precisione da effettuare nel tracking non credo che il numero di Reader in questo caso non possa comunque scendere a 15 (la metà rispetto alla soluzione 1), quindi in ogni caso non avremmo mai una soluzione migliore della precedente.

## 2.7 Conclusioni

Abbiamo speso tante belle parole sulla tecnologia RFID, abbiamo analizzato diversi sistemi (SpotON e Landmarc), e siamo convinti che questa sia la soluzione migliore per un sistema basato sul positioning indoor, in quanto non risente ad esempio del fattore di disturbo e di interferenza dovucheto ad un maggior numero di utenti contemporanei o alle condizioni climatiche/ambientali che invece è presente nei precedenti sistemi wi-fi based. Detto ciò, è doveroso aggiungere che, per quanto abbiamo potuto vedere nei nostri studi e nelle nostre ricerche, il **costo** di un sistema del genere è ancora troppo elevato. E' vero che siamo arrivati ad un costo dei tag irrisori, però per avere un sistema complessivo da poter definire a basso costo c'è bisogno di aspettare ancora qualche sviluppo tecnologico.

A tal proposito la nostra idea riguardo il positioning indoor si è spostata in un'altra direzione e nel prossimo capitolo vedremo di cosa si tratta.

---

## Positioning Indoor: Ibridizzazione Wi-Fi e RFID

---

### Indice

---

<b>3.1</b>	<b>Introduzione . . . . .</b>	<b>59</b>
<b>3.2</b>	<b>Il Progetto . . . . .</b>	<b>60</b>
3.2.1	SPinV . . . . .	60
<b>3.3</b>	<b>Algoritmi studiati . . . . .</b>	<b>61</b>
3.3.1	Soluzione # 1: Wi-Fi . . . . .	61
3.3.2	Soluzione # 2: Wi-Fi + RFID . . . . .	63
3.3.3	Soluzione # 3: RFID → Future Work . . . . .	64
<b>3.4</b>	<b>Le nostre scelte . . . . .</b>	<b>65</b>

---

### 3.1 Introduzione

Dopo quanto spiegato nei precedenti capitoli riguardo lo studio fatto sulle possibili tecnologie utilizzabili per lo localizzazione indoor, era necessario ef-

fettuare dei test empirici per poter finalmente arrivare a dei risultati e poter quindi effettuare determinati confronti in base alle scelte fatte.

## 3.2 Il Progetto

Il sistema che abbiamo preso come riferimento per la nostra simulazione è RADAR, [Bahl and Padmanabhan [2]] (RSS-based Wi-Fi Indoor Localizer), già trattato nel capitolo 1. Bahl et al. hanno effettuato i loro test al secondo piano di un edificio di tre piani, su una superficie di 22.5m x 43.5m, per un totale di circa 980 mq. Sono stati posizionate tre Base Station (BS) all'interno del piano ognuna delle quali era costituita da un Pentium-based PC, con adattatore wireless. Come già anticipato, anche graficamente (Fig. 1.7), la precisione ottenuta in queste condizioni raggiunge un livello buono, siamo sui 2.94 m di errore nella stima della posizione, quindi con un ordine di grandezza assimilabile ad una stanza. Nei vari test fatti poi hanno considerato diverse variazioni sia nell'applicazione dell'algoritmo KNN che nello scegliere la modalità di acquisizione delle informazioni di potenza di segnale.

### 3.2.1 SPIN V

Il nostro lavoro rientra in un progetto chiamato SPinV, che è un acronimo che sta per San Pietro in Vincoli, la zona in cui sorge la facoltà di Ingegneria dell'università di Roma, "Sapienza" e dove appunto nasce l'idea stessa del progetto.

Tale progetto comincia proprio dalla necessità di colmare le lacune esistenti nei servizi di posizionamento e localizzazione in ambiente indoor, dove la tecnologia GPS perde di valore. L'intenzione di questo progetto dunque è di creare un sistema che possa guidare gli utenti all'interno di luoghi pubblici come università, ospedali, centri commerciali, aeroporti e qualsiasi altro posto

che non possa essere raggiunto dal GPS (oppure anche Google Maps, Google Earth, etc...).

### 3.3 Algoritmi studiati

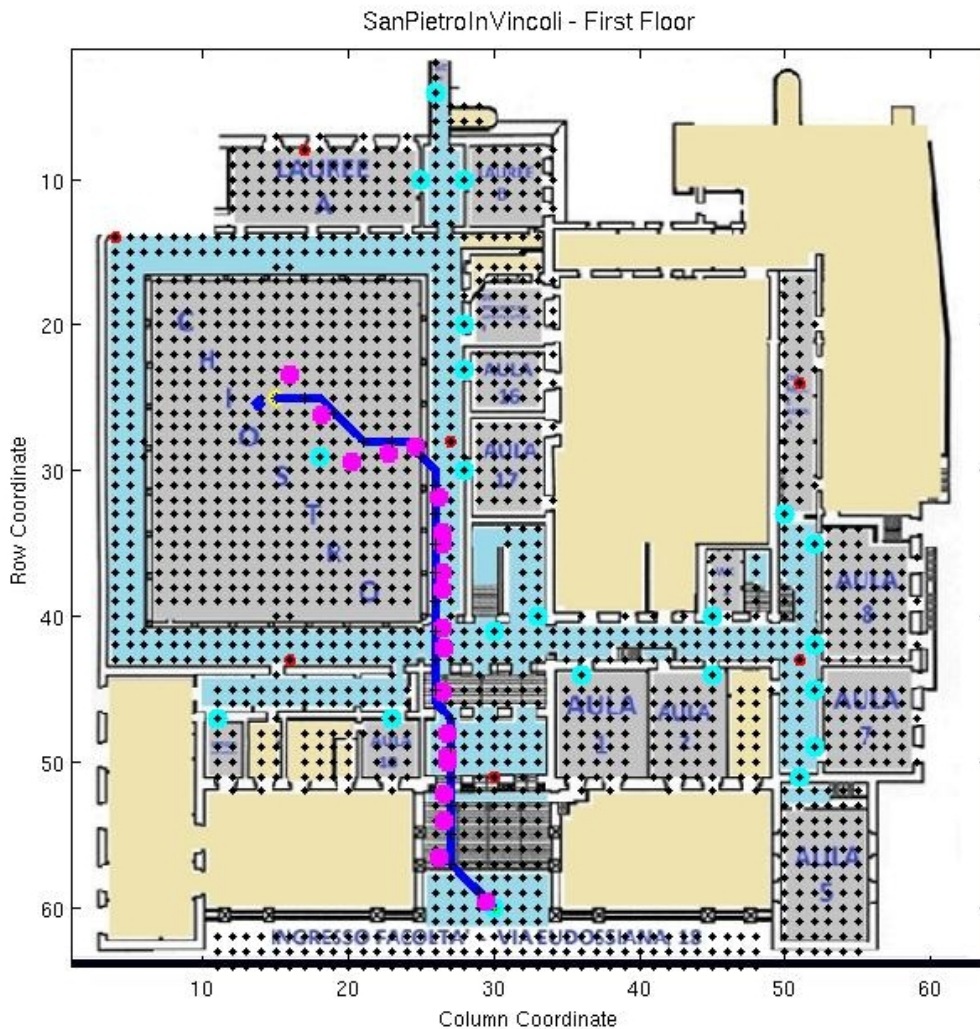
Prima di analizzare i risultati ottenuti con le varie simulazioni al variare dei parametri più importanti, per una maggiore facilità di lettura, crediamo sia importante visualizzare quale sia il nostro obiettivo primario, cioè quale sia la situazione reale della quale vogliamo massimizzare le prestazioni.

#### 3.3.1 SOLUZIONE # 1: WI-FI

Per questa prima soluzione ci siamo limitati ad impostare la griglia del fingerprinting con il passo più piccolo che permetteva la nostra implementazione, il numero di AP pari a 7; per l'algoritmo K-Nearest Neighbour [Roxin et al. [27]] abbiamo invece fissato  $K = 5$ . Abbiamo posizionato il nostro utente all'interno del chiostro (coordinate (15, 25)) e abbiamo calcolato il percorso minimo che avrebbe dovuto percorrere per raggiungere l'uscita della facoltà.

Per il calcolo del cammino minimo abbiamo scelto l'algoritmo di Dijkstra [Fuhao and Jiping [7]] e quindi ci siamo basati sulla teoria dei grafi.

In Fig. 3.1 vediamo la posizione iniziale dell'utente (in alto a sinistra) e la sua posizione finale (uscita della facoltà). La curva blu' rappresenta il percorso ideale calcolato al simulatore, quindi quello che noi vorremmo che il nostro "cliente" (utente) percorresse per raggiungere la sua destinazione nel minor tempo possibile. I pallini magenta invece sono la parte centrale del nostro sistema, rappresentano la stima della posizione dell'utente, che abbiamo calcolato durante il suo spostamento, cioè rappresentano il percorso che il nostro sistema riuscirebbe a monitorare in una situazione reale. Nel prossimo capitolo andremo a dare informazioni in più su quanto si possa migliorare o



**Figura 3.1:** Stima esplicativa percorso utente

peggiore questa stima, variando il passo della griglia del fingerprinting, il numero degli Access Point e il numero di  $K$  nell'algoritmo di localizzazione.

In base a quanto detto, come vedremo nel capitolo successivo dedicato a tutti i risultati ottenuti, in una situazione ideale, cioè applicando il modello di attenuazione nello spazio libero, abbiamo visto che al crescere del numero

degli access point, il valore dell'errore medio, dopo una serie di 1000 iterazioni, andava a sfiorare il minimo raggiungibile, in base al diverso passo della griglia. All'interno del codice infatti abbiamo calcolato quale fosse, a prescindere dall'algoritmo di stima della posizione, il minimo scarto calcolabile, considerando l'accuratezza che si voleva ottenere, in base sempre al passo della griglia, ed è stato importante verificare che riuscivamo effettivamente a raggiungerlo nel caso ideale.

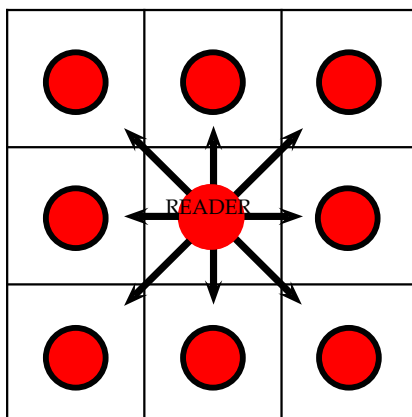
### 3.3.2 SOLUZIONE # 2: WI-FI + RFID

Nella seconda soluzione, dopo un'accurata analisi dei costi fatta nel capitolo 2, ci siamo posti il problema di considerare quale fosse il miglioramento che poteva apportare la tecnologia RFID, in aiuto al Wi-Fi.

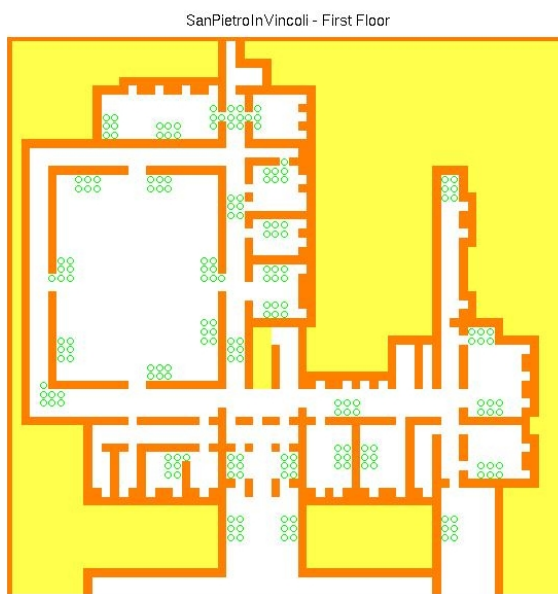
Quello che ci interessa è vedere quanto è possibile allargare il passo della griglia del fingerprinting aggiungendo gli RFID. Questo perchè, come vedremo, ovviamente con il passo della griglia strettissimo (1.3 m.) ovviamente i risultati sono ottimi, soltanto che in uno scenario reale, il costo di tale precisione, dal punto di vista del tempo impiegato per implementare la fase "off-line" sarebbe abbastanza alto, quindi quello che vogliamo è appunto un compromesso accettabile tra semplicità di creazione del database delle potenze e costo del sistema.

Abbiamo posizionato 31 reader UHF all'interno del nostro ambiente simulativo considerando l'analisi progettuale precedente, quindi reader di tipo UHF con range di lettura pari a 3 m. In base al grado di precisione massimo che ci permette la nostra mappa (matrice 64x64 dove  $1 = 1.3$  [m]), abbiamo considerato che ogni reader potesse "rilevare" l'utente, occupando al massimo 9 caselle (Fig. 3.2). Nella Fig. 3.3 invece, vediamo in che modo abbiamo posizionato i nostri RFID.





**Figura 3.2:** Range lettura RFID Reader nel nostro simulatore



**Figura 3.3:** Posizione Range Reader nel nostro simulatore

### 3.3.3 SOLUZIONE # 3: RFID → FUTURE WORK

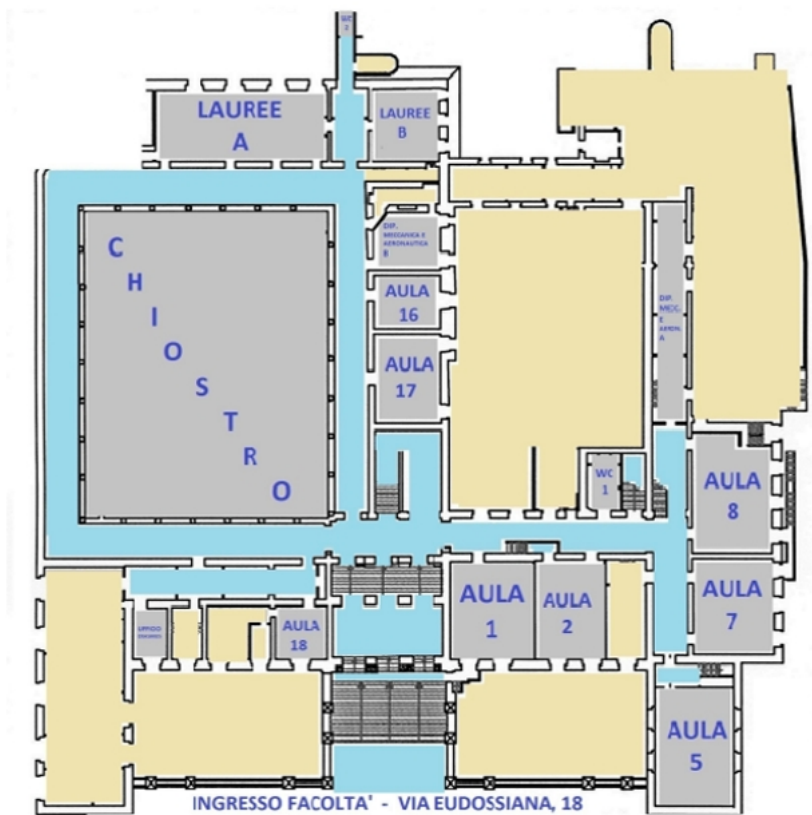
Quanto detto riguardo l'inserimento degli RFID nel progetto, riguarda sempre il calcolo della stima della posizione dell'utente. Sarebbe interessante, ed

è questo quello che faremo in un lavoro futuro, implementare una funzione di costo che oltre a calcolare la posizione dell'utente, ne possa seguire il cammino, quindi che riesca pesare, in base all'intervallo temporale considerato, l'importanza del passaggio nei pressi di un reader RFID, considerando quanto tempo è trascorso dall'ultimo rilevamento. Questo lo abbiamo già fatto con Dijkstra e abbiamo visto appunto che con un passo della griglia molto stretto, e con "1" solo utente nel sistema, il simulatore funziona molto bene. Tornando all'idea di allargare il più possibile la griglia del fingerprinting, il nostro intento è di provare a tenere conto di una variabile temporale.

In pratica se un utente, al tempo "t", si trova nei pressi di un determinato Reader, situato nella posizione "X", è ragionevole pensare che (muovendosi a piedi) nei successivi istanti temporali, il punto "X" debba essere ancora considerato un punto "vicino" all'utente; quindi se il sistema ottiene come posizione stimata una posizione molto distante dal punto "X" il simulatore dovrà riuscire a correggere l'errore e avvicinare la posizione dell'utente a quella reale. Per fare questo è importante però che il sistema riesca a dare molta importanza al variare del tempo perchè è in questo modo che si può riuscire a "pesare" l'importanza della rilevazione di un passaggio precedente nei pressi di un Reader RFID.

### 3.4 Le nostre scelte

Per i nostri test abbiamo scelto il primo piano della facoltà di Ingegneria, e come prima cosa abbiamo creato in MatLab una copia abbastanza fedele della mappa (Fig. 3.4) del nostro ambiente di lavoro, per poter rendere le simulazioni il più possibile reali. Come vediamo in Fig. 3.5, la mappa su cui abbiamo lavorato è una matrice M di dimensioni 64x64 all'interno della quale abbiamo definito con il valore 0 (in bianco) le zone calpestabili, quindi corridoi, aule, bagni, etc..., con il valore 1 (in arancio) tutti i muri separatori, con il

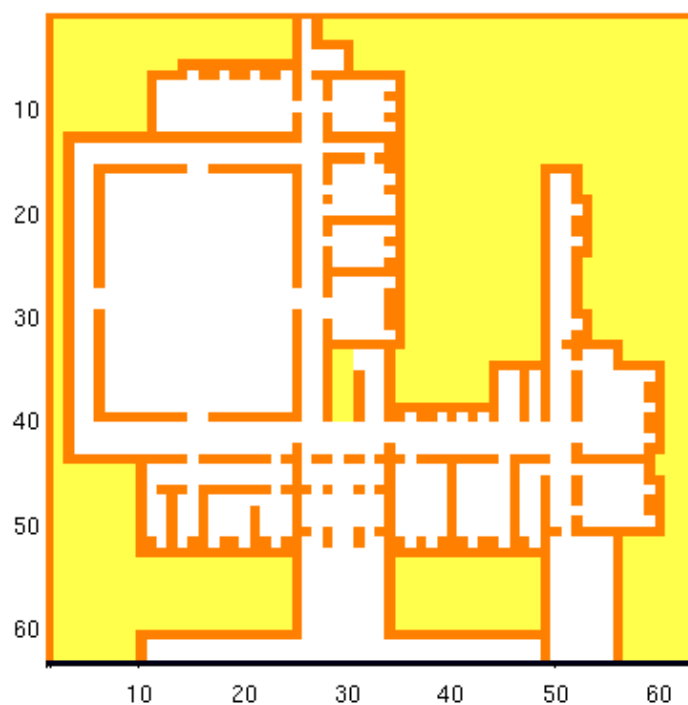


**Figura 3.4:** San Pietro in Vincoli: Primo Piano

colore 2 (giallo) tutte le parti che sicuramente non sono muri, ma che non fanno parte della mappa, come ad esempio le zone esterne e il cortile tra le palazzine. In tutte le simulazioni però abbiamo preferito “plottare” come sfondo la mappa reale, in modo da avere un risultato migliore.

Abbiamo posizionato come situazione iniziale 7 Access Point in posizione random (puntini rossi in Fig. 3.6), scegliendone cioè la posizione in base a quelle che secondo noi erano le esigenze dell’ambiente; poi durante le simulazioni abbiamo variato numero e posizione degli stessi.

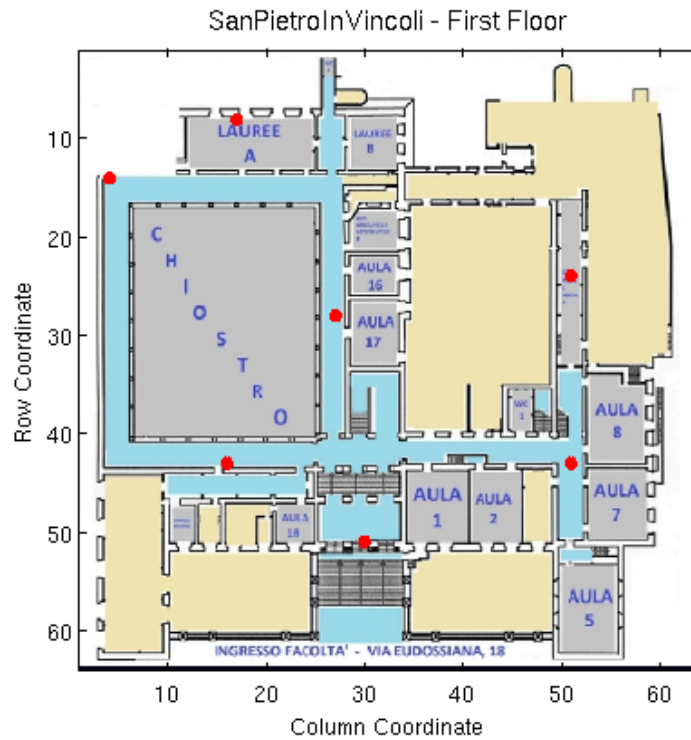
Quindi abbiamo fissato dei punti di interesse sulla mappa in modo da poter dare agli utenti del sistema tutte le informazioni aggiuntive che preferiscono. In



**Figura 3.5:** San Pietro in Vincoli: Mappa creata con Matlab

particolare nel nostro sistema a noi interessa sapere dove si trovi l'utente poter monitorare i suoi spostamenti con una certa precisione. Per fare ciò è stato sufficiente porre i "punti di interesse" (puntini celesti in Fig. 3.7) all'ingresso di tutte le zone raggiungibili sul piano. Non è importante sapere in quale punto preciso di una determinata stanza si trovi l'utente, per poter offrire il servizio che vogliamo, ci basta sapere che l'utente sia nell'aula  $x$  e che per raggiungere la posizione  $y$  debba seguire un determinato percorso che andremo a calcolare tra poco.

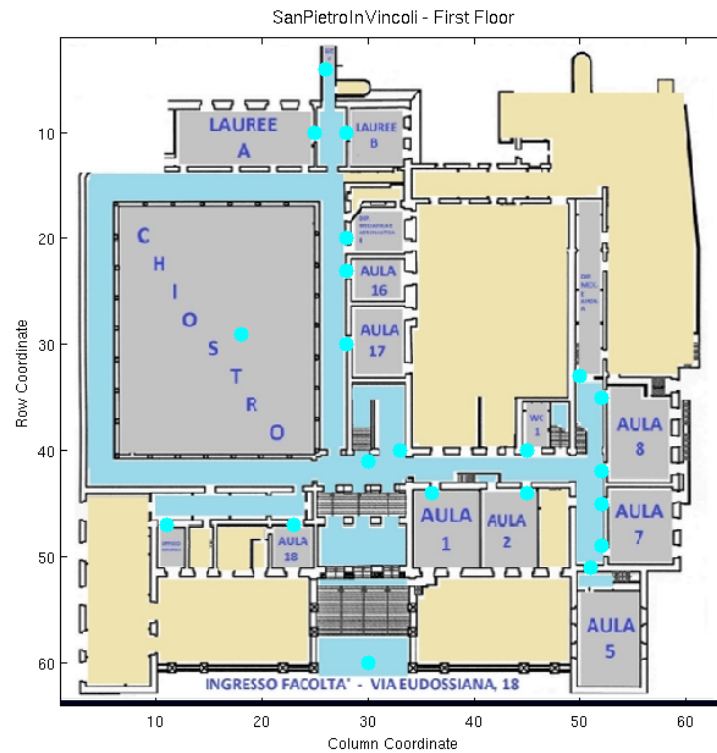
Dopo di che abbiamo creato la griglia per il fingerprinting, su tutti i punti calpestabili della matrice  $M$  ( $64 \times 64$ ) quindi abbiamo posizionato un punto della griglia su ognuno degli "zeri" della mappa, in modo da poter avere la risolu-



**Figura 3.6:** San Pietro in Vincoli: Posizionamento Access Point

zione massima; poi nelle simulazioni abbiamo provato a variare il passo della griglia per poter valutare quali fossero le differenze nei risultati ottenuti. In particolare questa analisi è molto importante perchè una volta fissata la griglia, bisogna effettuare quella che nel capitolo 1 chiamavamo “off-line phase”, cioè la misurazione di tutte le potenze ricevute da ogni punto della griglia su ogni access point. Abbiamo visto che nelle simulazioni questo è un procedimento davvero molto lungo in quanto tale operazione, in un’ipotetica realizzazione del sistema nella realtà, avrebbe bisogno di un tempo ancora più elevato in quanto bisognerebbe posizionarsi fisicamente su ogni punto e calcolare la potenza che viene percepita su ogni access point.

La parte più importante del simulatore risiede certamente nello studio e nel calcolo della matrice  $S$ , cioè della matrice che contiene in ogni elemento l’i-



**Figura 3.7:** San Pietro in Vincoli: Posizionamento Punti di Interesse

esimo valore di potenza ricevuto da ogni punto della griglia sul  $j$ -esimo access point. Per tale motivo abbiamo dovuto considerare le attenuazioni dovute alla variazione delle distanze e ai muri attraversati, quindi abbiamo utilizzato un modello di path loss chiamato “multi-wall model” [De Luca et al. [5]]. Questo modello stima la il path-loss calcolata ad una certa distanza tra Trasmittitore e Ricevitore con la formula:

$$L(d) = L_{os}(d) + M_w(dB)$$

dove  $L_{os}(d)$  vale:

$$L_{os}(d) = 10\gamma \log_{10}(d) + l_0, (dB)$$

$M_w$ Model Parameter	Descrizione
$l_c = 47.4$	Fattore costante (dB)
$l_1 = 3.8$	Attenuazione tra due muri di tipo “light” (dB); spessore (0, 20] cm
$l_2 = 3.9$	Attenuazione tra due muri di tipo “medium” (dB); spessore (20, 40] cm
$l_3 = 5.7$	Attenuazione tra due muri di tipo “heavy” (dB); spessore (40, 60] cm
$l_4 = 12.4$	Attenuazione tra due muri portanti (dB); spessore (60, 80] cm

**Tabella 3.1:** Lista e Descrizione dei parametri del “Multi Wall Model”

e  $\gamma$  è l’esponente del path-loss,  $d$  è la distanza LOS tra Tx e Rx (in metri),  $l_0$  è la componente fissa che di solito rappresenta la perdita media che si può misurare alla distanza di 1 metro dal Tx.  $M_w$  è invece la componente “multi-wall” che nel modello è espressa in questo modo:

$$M_w = l_c + \sum_{i=1}^I k_{wi} l_i + \sum_{n=1}^{N_d} \chi_n l_d + \sum_{n=1}^{N_{fd}} \lambda_n l_{fd} \text{ (dB)}$$

dove  $l_c$  è una costante,  $k_{wi}$  è il numero di muri penetrati di tipo  $i$ , con  $i = 1, 2, \dots, I$ ,  $N_d$  e  $N_{fd}$  sono rispettivamente il numero di porte standard o particolarmente spesse incontrate lungo il path e  $\chi_n$  e  $l_{fd}$  sono variabili che dipendono dal tipo di porta attraversata e che sono definite nel modello. Nella nostra implementazione, guardando la Tab. 3.1, in base al nostro ambiente di lavoro abbiamo deciso di considerare soltanto un tipo di muro  $l_2 = 3.9$  (dB) senza considerare il contributo dato dalla presenza di porte lungo il percorso.

Da questa trattazione teorica abbiamo implementato una parte dello script che ci restituiva esattamente una matrice, che contiene su ogni elemento il numero di muri attraversati da un ipotetico segnale in linea diretta tra tra-

smettitore (utente) e ricevitore (AP). Come dicevamo questa fase è appunto molto lunga, infatti come si vede nella parte iniziale del codice, che riportiamo in appendice, abbiamo effettuato una lunga simulazione una volta per tutte e poi abbiamo salvato i risultati, richiamandoli semplicemente durante le compilazioni.



---

## MatLab: Risultato SIMULAZIONI

---

### Indice

---

<b>4.1</b>	<b>Introduzione</b> . . . . .	<b>73</b>
<b>4.2</b>	<b>Wi-Fi</b> . . . . .	<b>73</b>
4.2.1	Free Space Model (Caso ideale) . . . . .	74
4.2.2	Multi wall Model: Variazione nr. Access Point in funzione del passo della griglia . . . . .	76
4.2.3	Multi wall Model: Variazione numero di Neighbors (K) in funzione del passo della griglia . . . . .	78
<b>4.3</b>	<b>Wi-Fi + RFID</b> . . . . .	<b>80</b>
4.3.1	Multi wall Model (Wi-Fi + RFID): Variazione nr. Access Point in funzione del passo della griglia . . . . .	80
4.3.2	Multi wall Model (Wi-Fi + RFID): Variazione nu- mero di Neighbors (K) in funzione del passo della griglia . . . . .	82

---

## 4.1 Introduzione

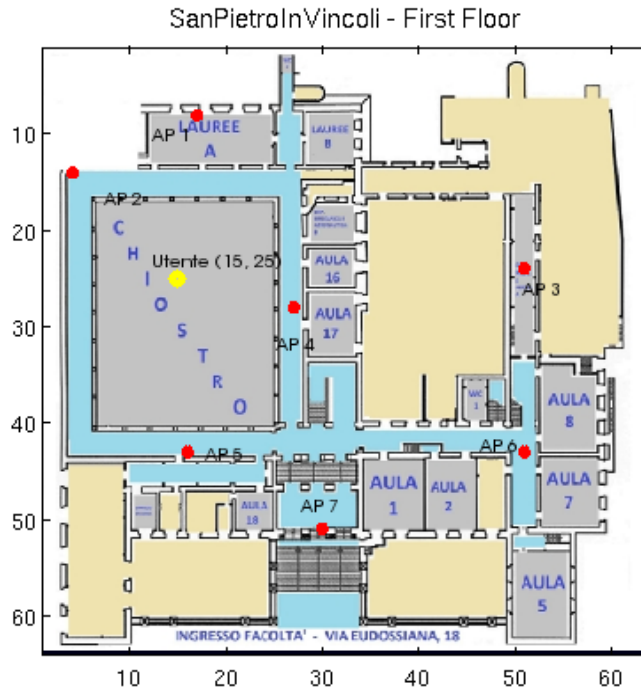
Questo capitolo sarà dedicato alla visualizzazione e ai commenti di tutte le nostre simulazioni, che abbiamo effettuato facendo variare il passo della griglia del fingerprinting, facendo variare il numero di Access Point e facendo variare il numero di Neighbors (K) all'interno dell'algoritmo di localizzazione K-Nearest Neighbors. Nella seconda parte invece vedremo cosa è successo con l'inserimento degli RFID.

## 4.2 Wi-Fi

In una prima serie di simulazioni abbiamo voluto studiare i risultati che si potevano ottenere facendo variare il numero di AP da 1 a 7, considerando il passo della griglia pari a 1, 2, 10 (nell'ambiente reale 1 punto della griglia equivale a 1.3 [m]). Quello che ci aspettavamo era ovviamente avere la situazione migliore nel caso di 7 APs e con un passo della griglia il più stretto possibile, in modo da ridurre la possibilità di errore.

Nonostante avessimo considerato sia l'attenuazione in potenza dovuta alla distanza, che l'attenuazione dovuta agli ostacoli incontrati, il nostro sistema non si poteva ancora considerare assimilabile alla realtà, in quanto dava ancora dei risultati con errori nulli, situazione che da un lato può essere un riscontro positivo, ma che in realtà non possiamo considerare accettabile, perchè ci troveremmo comunque in una situazione ideale non utile per i nostri scopi. Abbiamo inserito all'interno del calcolo del path-loss e quindi della matrice S (RSS - Received Signal Strength), il fattore di shadowing, in particolare abbiamo considerato il Log-Normal Shadowing Model [Szyszkowicz et al. [29]]. Questo modello è adatto in un ambiente di propagazione per uno scenario con mobilità e ostacoli; il valore di shadowing (fading lento) segue una distribuzione log-normale con una specificata deviazione standard. In generale, in ambiente

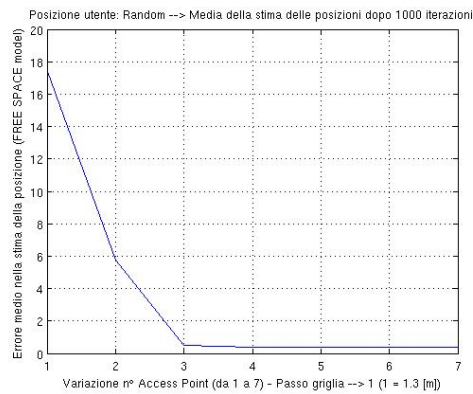
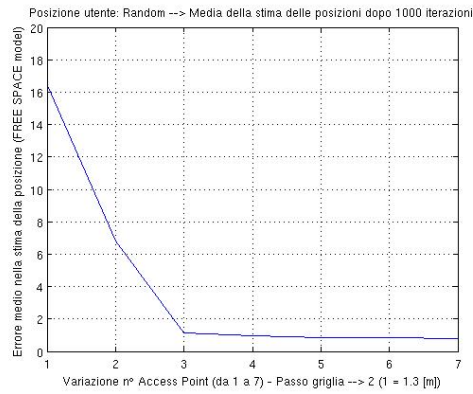
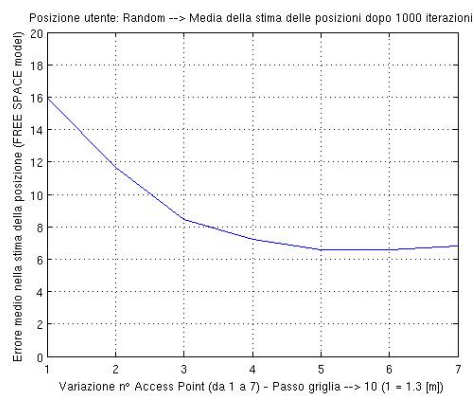
wireless, in base al numero di ostruzioni che sono presenti, possiamo inserire questo fenomeno con una deviazione standard  $\sigma$  che varia nel range di 4 e 12 dB.



**Figura 4.1:** Posizione dell'utente - Numerazione AP considerati

#### 4.2.1 FREE SPACE MODEL (CASO IDEALE)

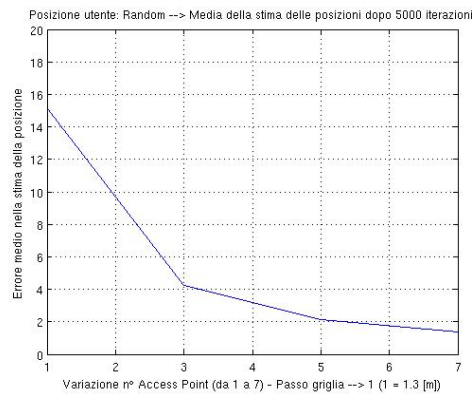
Come prima cosa riportiamo (Fig. 4.2, Fig. 4.3, Fig. 4.4) i risultati che abbiamo ottenuto considerando il modello di propagazione nello spazio libero, che simula una certa condizione di idealità. L'unico commento che facciamo è che all'aumentare del numero degli Access Point, dopo 1000 iterazioni, raggiungiamo il limite del minimo errore possibile in base all'accuratezza della griglia.

**Figura 4.2:** Free Space Model: Passo della griglia: 1**Figura 4.3:** Free Space Model: Passo della griglia: 2**Figura 4.4:** Free Space Model: Passo della griglia: 10

#### 4.2.2 MULTI WALL MODEL: VARIAZIONE NR. ACCESS POINT IN FUNZIONE DEL PASSO DELLA GRIGLIA

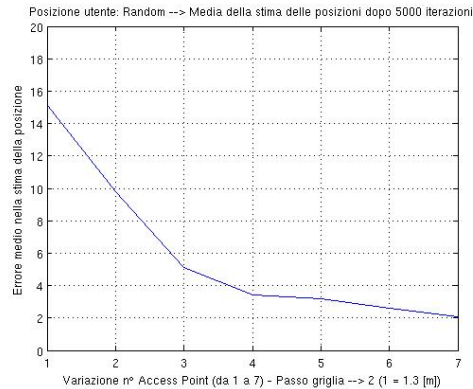
Riportiamo prima di tutto i grafici che abbiamo ottenuto dopo 5000 iterazioni (quindi 5000 posizioni random) in Fig. 4.5, Fig. 4.6, Fig. 4.7; Abbiamo preferito non creare un grafico unico per poter analizzare ogni risultato singolarmente.

Salta subito agli occhi che al crescere del numero degli AP abbiamo dei miglioramenti; nel caso di 1 AP in Fig. 4.5 c'è una grande differenza tra l'errore minimo (0.2) e l'errore massimo (addirittura 50.44), con una media di 15.1449, nel caso opposto invece di 7 APs, il range di errore varia da 0 a 15.9298, con una media di 1.404 quindi diminuendo il range dell'errore vuol dire che riusciamo ad essere allo stesso tempo precisi e accurati. E' importante annotare il grado di precisione così ottenuto, considerando la scala della nostra mappa: 1.404 pts. equivalgono a una precisione di **1.82 [m]**. Inoltre in base al funzionamento dell'algoritmo K-NN, è impossibile arrivare ad avere un errore minore di 0.5 pts (mezzo punto della griglia).

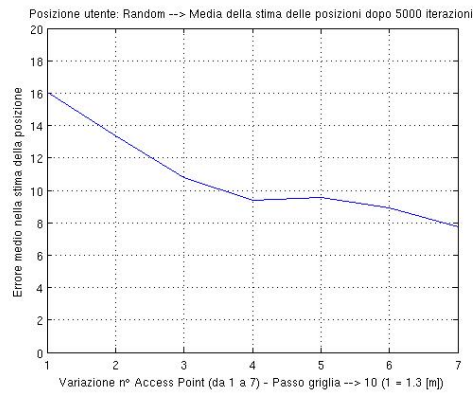


**Figura 4.5:** Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 1

Risultati simili li vediamo in Fig. 4.6 con la differenza che non arriviamo ad avere una precisione migliore, rispetto al caso precedente, con un passo della



**Figura 4.6:** Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 2



**Figura 4.7:** Wi-Fi: Variazione nr. AP da 1 a 7 - Passo della griglia: 10

griglia minore, a parità di numero di APs.

Allargando la griglia fino a un passo 10 (Fig. 4.7) come ci aspettavamo vediamo un lieve peggioramento, rispetto al caso analogo analizzato nel free space model. Vediamo che nella migliore delle ipotesi facciamo (7 APs) facciamo un errore di 7.73 pts. (circa 10 m.); è sicuramente un valore eccessivo, aspettiamo di vedere dunque se riusciamo ad avere dei miglioramenti con l'aiuto dei reader RFID.

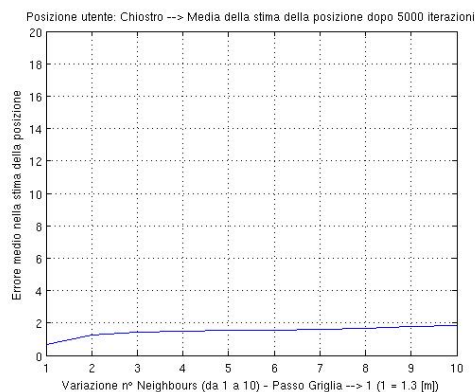
Numero APs	APs scelti
1	4
2	4 e 5
3	4, 5 e 6
4	4, 5, 6 e 7
5	2, 3, 4, 5 e 6
6	2, 3, 4, 5, 6 e 7
7	Tutti

**Tabella 4.1:** Configurazioni APs considerate nelle simulazioni

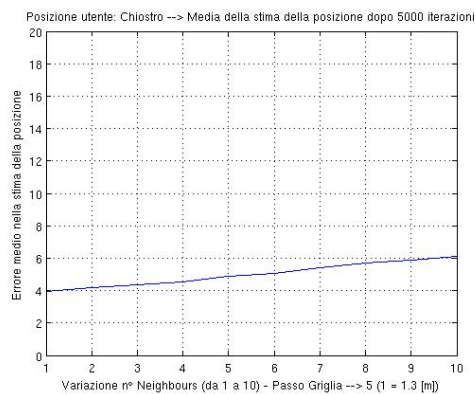
#### 4.2.3 MULTI WALL MODEL: VARIAZIONE NUMERO DI NEIGHBORS (K) IN FUNZIONE DEL PASSO DELLA GRIGLIA

Il numero di run considerate (5000) è lo stesso delle altre simulazioni, quindi abbiamo anche in questo caso una lunga serie di posizioni da poter stimare; valgono inoltre gli stessi accorgimenti fatti precedentemente riguardo la variazione del fattore di shadowing ( $\sigma$  da 4 a 12 dB). Riportiamo i grafici che abbiamo ottenuto nelle Fig. 4.8, Fig. 4.9 e Fig. 4.10.

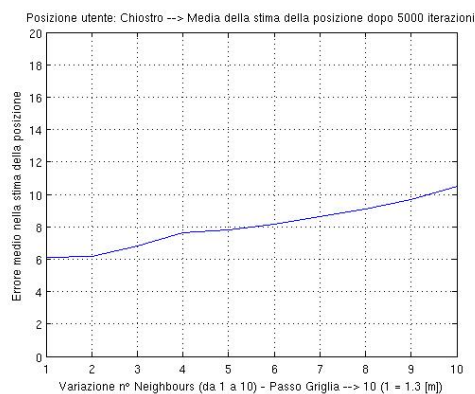
In tutti e tre i casi all'aumentare del numero di K considerati abbiamo un aumento dell'errore, questo è dovuto alla presenza dell'aleatorietà inserita con il fattore di shadowing. E' importante considerare però che, dai risultati ottenuti, avendo un numero minore di Neighbours magari abbiamo una precisione maggiore, quando non siamo in una posizione sfavorevole, però rischiamo di avere una poca accuratezza nei risultati, se veniamo ingannati nel calcolo delle potenze ricevute.



**Figura 4.8:** Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 1



**Figura 4.9:** Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 5



**Figura 4.10:** Wi-Fi: Variazione K Neighbours da 1 a 10 - Passo della griglia: 10



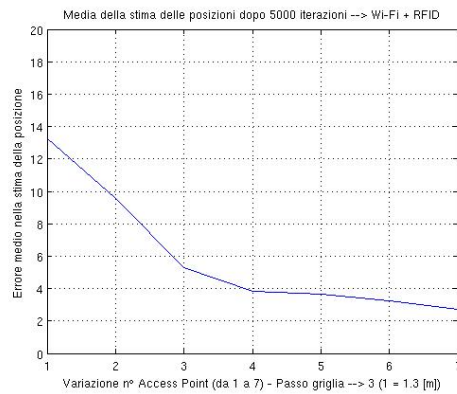
### 4.3 Wi-Fi + RFID

In base ai risultati ottenuti, come anticipato nel capitolo precedente, abbiamo provato ad inserire dei Reader RFID UHF nel nostro sistema e, modificando opportunamente il codice, abbiamo voluto analizzare i risultati che riuscivamo ad ottenere.

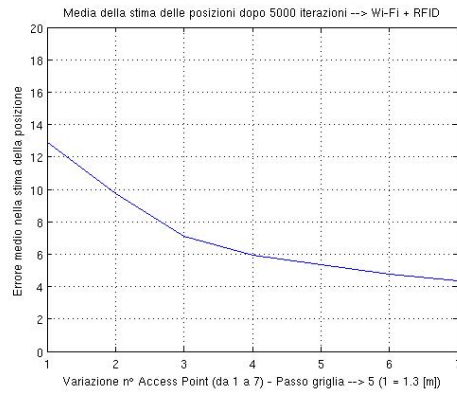
Nei precedenti scenari, il simulatore analizzava tutte potenze, calcolava le distanze e uclidee e calcolava la stima della posizione dell'utente facendo una media delle  $K$  posizioni (presunte) più vicine. Con l'inserimento dei Reader abbiamo dovuto modificare il simulatore: dunque deve continuare a lavorare considerando gli algoritmi di posizionamento, a meno che l'utente non si trovi nelle vicinanze (In Fig. 3.2 abbiamo descritto la precisione che potevamo ottenere sulla nostra mappa  $64 \times 64$ ) di un Reader  $k$ -esimo. Nel caso in cui fosse vera l'ultima ipotesi, il sistema non implementa più l'algoritmo K-NN ma calcola la stima della posizione dell'utente mediando su tutte le "n" coppie di coordinate dei punti raggiungibili dal  $k$ -esimo Reader.

#### 4.3.1 MULTI WALL MODEL (Wi-Fi + RFID): VARIAZIONE NR. ACCESS POINT IN FUNZIONE DEL PASSO DELLA GRIGLIA

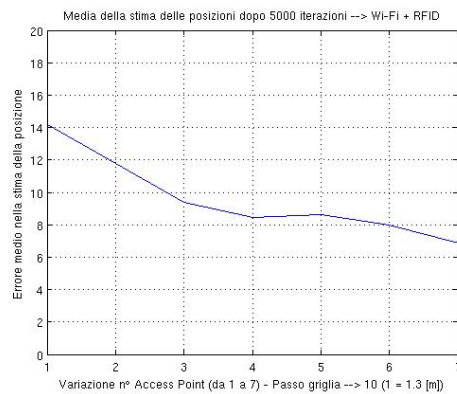
Partendo dunque dalle ipotesi fatte, abbiamo fatto variare il numero di APs da 1 a 7, allargando il passo della griglia ogni volta (3, 5 e 10), facendo compiere come sempre 5000 iterazioni ogni volta; i risultati grafici sono quelli di Fig. 4.11, Fig. 4.12 e Fig. 4.13. Dai grafici, come ci aspettavamo, nel caso di griglia molto stretta non abbiamo molti miglioramenti, rispetto al caso del solo Wi-Fi, perchè in ogni modo avevamo già una precisione molto buona mentre, invece, quando il passo della griglia è posto a 10, nello scenario migliore (7 APs), riusciamo a passare da un errore medio di circa 7.9 pts. (10.27 m.) a un errore medio di 6.8 pts (8.84 m), guadagnando quindi circa 1.5 m. di accuratezza.



**Figura 4.11:** Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 3



**Figura 4.12:** Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 5

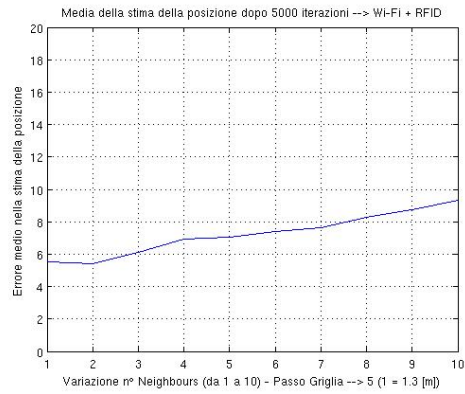


**Figura 4.13:** Wi-Fi + RFID: Variazione nr. AP da 1 a 7 - Passo della griglia: 10

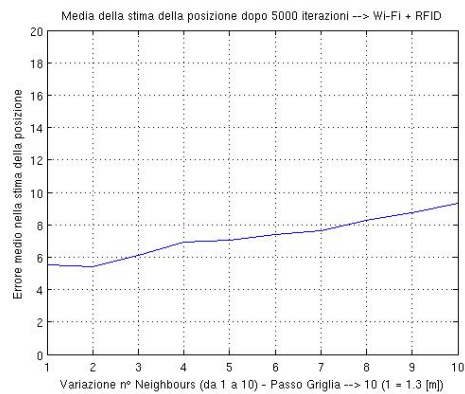
#### 4.3.2 MULTI WALL MODEL (Wi-Fi + RFID): VARIAZIONE NUMERO DI NEIGHBORS (K) IN FUNZIONE DEL PASSO DELLA GRIGLIA

Per questo scenario, in base a quanto avevamo riscontrato nella variazione degli APs, abbiamo considerato soltanto i casi di griglia a passo 5 (Fig. 4.14) e 10 (Fig. 4.15).

Quello che abbiamo notato è che con il passo della griglia posto a 5, quindi ancora “stretto”, non abbiamo miglioramenti, anzi abbiamo un andamento che è poco peggiore del caso del solo Wi-Fi; vediamo invece, anche se lieve, un miglioramento nel caso di griglia a passo 10, nel quale sia nella situazione peggiore ( $K = 10$ ), sia nella situazione migliore ( $K = 1$ ), abbiamo una diminuzione dell’errore medio di circa 1 punto (1.3 m.)



**Figura 4.14:** Wi-Fi + RFID: Variazione K Neighbours da 1 a 10 - Passo della griglia: 5



**Figura 4.15:** Wi-Fi + RFID: Variazione K Neighbours da 1 a 10 - Passo della griglia: 10

---

## Conclusioni e Lavori Futuri

---

A questo punto possiamo dire che in base alle nostre scelte iniziali, in base alle simulazioni e ai risultati ottenuti, sicuramente riusciamo ad ottenere i risultati migliori, con una precisione molto buona, anche di meno di 2 metri (abbiamo punti in cui raggiungiamo la precisione di 1.82 m) configurando l'algoritmo K-NN con 5 Neighbours e inserendo sicuramente almeno 4 Access Point.

Il confronto con "RADAR" di Bahl et al. (2000), il primo sistema di posizionamento RSS-based, è obbligatorio. Abbiamo visto che tale progetto ha portato ad una precisione molto buona, di 2.94 m.. Nelle loro simulazioni empiriche, Bahl et al., hanno considerato la direzione delle antenne dei terminali mobili, cosa che noi non abbiamo fatto, ma non hanno posto alcuna attenzione sul tracking. Noi siamo riusciti a raggiungere una precisione maggiore, riuscendo ad effettuare innumerevoli simulazioni variando il passo della griglia,

che loro invece hanno tenuto sempre fisso.

L'ibridizzazione che abbiamo ottenuto inserendo la seconda tecnologia nelle nostre simulazioni, è stata pensata per cercare di trovare un compromesso tra il costo in denaro e il costo computazionale del sistema.

E' interessante poter avere una griglia meno fitta, e riuscire a seguire l'utente con una buona precisione, anche se c'è ancora molto da lavorare in quanto è vero che siamo riusciti ad avere un miglioramento di qualche metro, però con un errore medio di 7-8 metri non possiamo ancora avere un sistema molto stabile. In ogni caso, anche se non siamo stati in grado di stimare quale possa essere il tempo di cui si avrebbe bisogno per implementare un database di potenze (fingerprinting) con un passo così stretto di circa 1.3 m., è comunque molto interessante aver potuto valutare con una media di molte simulazioni, e quindi su molte posizioni diverse, in un ambiente abbastanza ostile ricco di corridoi, muri e stanze, una stima media dell'errore inferiore ai 2 metri.

Come abbiamo anticipato, quello che abbiamo intenzione di fare è prima di tutto provare ad effettuare dei reali test nella nostra facoltà, magari servendoci della rete wireless della "Sapienza" e poi implementare un secondo simulatore che possa in qualche modo tenere conto del passare del tempo e quindi che possa tenere memoria delle posizioni passate che ha occupato l'utente, in modo da poter diminuire la possibilità di errore il più possibile.



---

## Bibliografia

---

- [1] Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Medium access control (mac) security enhancements. *IEEE Std 802.11i-2004*, 2004. (Citato a pagina **17**.)
- [2] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000. (Citato alle pagine **10, 21 and 60**.)
- [3] M. Ciurana, F. Barceló, and S. Cugno. Indoor tracking in wlan location with toa measurements. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access, MobiWac '06*, pages 121–125, New York, NY, USA, 2006. ACM. ISBN 1-



- 59593-488-X. doi: <http://doi.acm.org/10.1145/1164783.1164806>. URL <http://doi.acm.org/10.1145/1164783.1164806>. (Citato a pagina **13**.)
- [4] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. A ranging method with iee 802.11 data frames for indoor localization. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2092–2096, 2007. doi: 10.1109/WCNC.2007.392. (Citato alle pagine **9 and 13**.)
- [5] D. De Luca, F. Mazzenga, C. Monti, and M. Vari. Performance evaluation of indoor localization techniques based on rf power measurements from active or passive devices. *EURASIP J. Appl. Signal Process.*, 2006:160–160, January 2006. ISSN 1110-8657. doi: <http://dx.doi.org/10.1155/ASP/2006/74796>. URL <http://dx.doi.org/10.1155/ASP/2006/74796>. (Citato a pagina **69**.)
- [6] EPCglobal-Inc. Class 1 generation 2 uhf air interface protocol standard version 1.1.0. Website, 2008. <http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2110-standard-20071017.pdf>. (Citato a pagina **34**.)
- [7] Z. Fuhao and L. Jiping. An algorithm of shortest path based on dijkstra for huge data. *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, 4:244–247, 2009. doi: <http://doi.ieeecomputersociety.org/10.1109/FSKD.2009.848>. (Citato a pagina **61**.)
- [8] S. Golden and S. Bateman. Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging. *Mobile Computing, IEEE Transactions on*, 6(10):1185–1198, 2007. ISSN 1536-1233. doi: 10.1109/TMC.2007.1002. (Citato a pagina **13**.)
- [9] A. Gunther and C. Hoene. Measuring round trip times to determine the distance between wlan nodes. *NETWORKING 2005*, pages 768–779, 2005. URL [http://dx.doi.org/10.1007/11422778\\_62](http://dx.doi.org/10.1007/11422778_62). (Citato a pagina **13**.)

- [10] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 70–84, New York, NY, USA, 2004. ACM. ISBN 1-58113-868-7. doi: <http://doi.acm.org/10.1145/1023720.1023728>. URL <http://doi.acm.org/10.1145/1023720.1023728>. (Citato a pagina **25**.)
- [11] Ibiesse. Tipi di tag e frequenze di lavoro. Website. [http://www.ibiesse.it/RFID\\_tag5.html](http://www.ibiesse.it/RFID_tag5.html). (Citato a pagina **37**.)
- [12] Jeffrey. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. URL <http://seattle.intel-research.net/people/jhightower/pubs/hightower2000indoor/hightower2000indoor.pdf>. (Citato alle pagine **45 and 46**.)
- [13] K. Kaemarungsi and P. Krishnamurthy. Properties of indoor received signal strength for wlan location fingerprinting. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 14 – 23, 2004. doi: 10.1109/MOBIQ.2004.1331706. (Citato a pagina **25**.)
- [14] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1012 – 1022 vol.2, 2004. doi: 10.1109/INFCOM.2004.1356988. (Citato a pagina **26**.)
- [15] K. Kaemarungsi and P. Krishnamurthy. Properties of indoor received signal strength for wlan location fingerprinting. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 14 – 23, 2004. doi: 10.1109/MOBIQ.2004.1331706. (Citato a pagina **10**.)

- [16] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies. Indoor wi-fi positioning: techniques and systems. *Annals of Telecommunications*, 64: 651–664, 2009. ISSN 0003-4347. URL <http://dx.doi.org/10.1007/s12243-009-0122-1>. 10.1007/s12243-009-0122-1. (Citato a pagina **9**.)
- [17] B. Li, J. Salter, A. G. Dempster, and C. Rizos. Indoor positioning techniques based on wireless LAN. In *LAN, FIRST IEEE INTERNATIONAL CONFERENCE ON WIRELESS BROADBAND AND ULTRA WIDEBAND COMMUNICATIONS*, pages 13–16. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.1265>. (Citato a pagina **25**.)
- [18] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo. On the scalability of a novel wlan positioning system based on time of arrival measurements. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 15 –21, 2008. doi: 10.1109/WPNC.2008.4510352. (Citato alle pagine **14 and 18**.)
- [19] K. Madsen. Methods for non-linear least squares problems. Website, 2004. [http://www2.imm.dtu.dk/pubdb/views/edoc\\_download.php/3215/pdf/imm3215.pdf](http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3215/pdf/imm3215.pdf). (Citato a pagina **20**.)
- [20] A. Mahtab Hossain, H. N. Van, Y. Jin, and W.-S. Soh. Indoor localization using multiple wireless technologies. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1 –8, 2007. doi: 10.1109/MOBHOC.2007.4428622. (Citato alle pagine **23 and 25**.)
- [21] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. *Pervasive Computing and Communications, IEEE International Conference on*, 0:407–415, 2003. doi: 10.1109/PERCOM.2003.1192765. URL <http://dx.doi.org/10.1109/PERCOM.2003.1192765>. (Citato alle pagine **48 and 49**.)

- 
- [22] G. R. P. Talone. Rfid - fondamenti di una tecnologia silenziosamente pervasiva. Website, 2008. [www.gruppotia.it/Pdf/LibroBianco.pdf](http://www.gruppotia.it/Pdf/LibroBianco.pdf). (Citato a pagina **32**.)
- [23] RF-ID. Rfid for your business. Website. <http://www.rf-id.it/>. (Citato alle pagine **54 and 56**.)
- [24] RFID-Italia. Che cos'è l'rfid? Website, 22 Settembre 2009. <http://rfid.thebizloft.com>. (Citato a pagina **33**.)
- [25] RFID-Journal. A summary of rfid standards. Website. <http://www.rfidjournal.com/article/print/1335>. (Citato a pagina **34**.)
- [26] RFIDEas. Proximity card readers for personal identifier. Website. <http://www.rfideas.com/>. (Citato a pagina **45**.)
- [27] A. Roxin, J. Gaber, M. Wack, and A. Nait-Sidi-Moh. Survey of wireless geolocation techniques. In *Globecom Workshops, 2007 IEEE*, pages 1–9, 2007. doi: 10.1109/GLOCOMW.2007.4437809. (Citato alle pagine **11, 12 and 61**.)
- [28] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat. Location determination of a mobile device using ieee 802.11b access point signals. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1987–1992 vol.3, 2003. doi: 10.1109/WCNC.2003.1200692. (Citato a pagina **9**.)
- [29] S. Szyszkowicz, H. Yanikomeroglu, and J. Thompson. On the feasibility of wireless shadowing correlation models. *Vehicular Technology, IEEE Transactions on*, 59(9):4222–4236, 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2082006. (Citato a pagina **73**.)
- [30] Q. Tran, J. Tantra, C. H. Foh, A.-H. Tan, K. C. Yow, and D. Qiu. Wireless indoor positioning system with enhanced nearest neighbors in signal space

- 
- algorithm. In *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pages 1–5, 2006. doi: 10.1109/VTCF.2006.489. (Citato a pagina **10**.)
- [31] T. Vaupel, J. Seitz, F. Kiefer, S. Haimerl, and J. Thielecke. Wi-fi positioning: System considerations and device calibration. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–7, 2010. doi: 10.1109/IPIN.2010.5646207. (Citato a pagina **9**.)
- [32] D. C. Wyld. *RFID: The Right Frequency for Government*. IBM Center for Business of Government, Ottobre 2005. pp.32-33. (Citato a pagina **44**.)



APPENDICE A

---

**Codice Simulatore**

---

Alleghiamo il codice sorgente dello script nella forma generale; per le simulazioni abbiamo effettuato delle variazioni al codice implementando cicli differenti in base alle esigenze di studio.

```

clear all;
close all;
clc;

%%% Carico Workspace calcolato precedentemente %%%%

load('Nodi.mat'); % Carico i nodi della matrice M

load('Link.mat'); % Carico i link della matrice M

% Carico la matrice che contiene la misurazione di tutti i muri
% attraversati da un segnale che parte dal j-esimo punto della griglia e
% arriva all'i-esimo ACCESS POINT.

load('mat_num_wall_1_1.mat');
mat_num_wall_1_1 = mat_num_wall;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% IBRIDIZZAZIONE --> AGGIUNTA TECNOLOGIA R.F.I.D. <-- %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% VALORI DI INPUT %%%%%%%%%
% Eliminiamo il commento dalle seguenti due righe se vogliamo impostare
% manualmente la spaziatura della griglia del fingerprinting:
% w_x = input('Inserisci la variazione sulle x \n\n'); % variazione sulle x
% w_y = input('Inserisci la variazione sulle y \n\n'); % variazione sulle y

w_x = 1; % Spaziatura orizzontale griglia
w_y = 1; % Spaziatura verticale griglia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% POSITIONING E SMALLEST-PATH %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Versione CON CARICAMENTO WORKSPACE

% Versione con MATRICE M modificata e definitiva

%           % % % MULTI WALL CHANNEL MODE % % % %
%
% Generalmente se consideriamo un'area chiusa, non troppo estesa, non
% possiamo considerare il semplice modello di propagazione (path loss)
% applicato al caso ideale di free space e LOS per arrivare a calcolare le
% potenze ricevute. Abbiamo dunque bisogno di un modello differente che
% tenga conto della geometria dell'ambiente in considerazione.
% Cio' che facciamo in questo script e' considerare appunto il "multiwall
% path loss model, in accordo con la propagazione a 2.4 GHz (Wi-Fi).
% Questo modello e' stato sperimentato all'universita' Tor Vergata di Roma
% [Bibliografia: Performance Evaluation of Indoor... (De Luca)] e prevede
% il calcolo del path-loss come segue:
%
%                               L(d) = Los(d) + Mw(dB)
%
%
% Per prima cosa riportiamo la mappa del primo piano della facolta' di
% ingegneria di Roma, San Pietro in Vincoli:

% Dati iniziali
f = 2400000; % Frequenza Operativa (Hz)

```



```

c = 300000000; % Velocita' della luca (m/s)
lamda = c/f;

% LOG-NORMAL SHADOWING MODEL %%%%%%%%%%
%
% All'interno del modello per il path loss inseriamo anche il fattore di
% shadowing che di solito varia tra 4 e 12 dB. Piu' e' alto il numero di
% ostacoli nella propagazione e piu' tale valore aumenta. In base al nostro
% ambiente di lavoro
% All'interno della formula del path loss allora, genereremo ogni volta un
% numero casuale da 4 a 12 con la formula:
%
%
%           delta_shadowing = a + (b - a).*rand(1)
%           con:           a = 4
%                       b = 12
%
delta_min = 4; % [dB]
delta_max = 12; % [dB]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% FIRST FLOOR MAP

%
% San Pietro in Vincoli,
% Facolta' di Ingegneria, "La Sapienza", Roma
%

close all;
% create map grid:
M = zeros(64,64);

% LEGENDA MAPPA:
%
% 0 = Punti Calpestabili
% 1 = Muri
% 2 = Punti Calpestabili (quindi non muri) ma che non fanno parte della
%     mappa

% create obstacles:
for i = 1:64
    M(i,64)=1;
end
M(1,1:63)=1;
M(2,1)=1;M(2,2:24)=2;M(2,25)=1;M(2,27)=1;M(2,28:63)=2;
M(3,1)=1;M(3,2:24)=2;M(3,25)=1;M(3,27)=1;M(3,28:63)=2;
M(4,1)=1;M(4,2:24)=2;M(4,25)=1;M(4,27:30)=1;M(4,31:63)=2;
M(5,1)=1;M(5,2:24)=2;M(5,25)=1;M(5,30)=1;M(5,31:63)=2;
M(6,1)=1;M(6,2:13)=2;M(6,14:25)=1;M(6,30)=1;M(6,31:63)=2;
M(7,1)=1;M(7,2:10)=2;M(7,11:14)=1;M(7,16:17)=1;M(7,19:20)=1; ...
    M(7,22:23)=1;M(7,25)=1;M(7,27:35)=1;M(7,36:63)=2;
M(8,1)=1;M(8,2:10)=2;M(8,11)=1;M(8,25)=1;M(8,28)=1;M(8,35)=1;M(8,36:63)=2;
M(9,1)=1;M(9,2:10)=2;M(9,11)=1;M(9,25)=1;M(9,28)=1;M(9,34:35)=1; ...
    M(9,36:63)=2;
M(10,1)=1;M(10,2:10)=2;M(10,11)=1;M(10,35)=1;M(10,36:63)=2;
M(11,1)=1;M(11,2:10)=2;M(11,11)=1;M(11,25)=1;M(11,28)=1;M(11,34:35)=1; ...
    M(11,36:63)=2;
M(12,1)=1;M(12,2:10)=2;M(12,11)=1;M(12,25)=1;M(12,28)=1;M(12,35)=1; ...
    M(12,36:63)=2;
M(13,1)=1;M(13,2)=2;M(13,3:25)=1;M(13,28:35)=1;M(13,36:63)=2;
M(14,1)=1;M(14,2)=2;M(14,3)=1;M(14,34:35)=1;M(14,36:63)=2;
M(15,1)=1;M(15,2)=2;M(15,3)=1;M(15,28:31)=1;M(15,33:35)=1;M(15,36:63)=2;
M(16,1)=1;M(16,2)=2;M(16,3)=1;M(16,6:14)=1;M(16,17:25)=1;M(16,28)=1; ...
    M(16,34:35)=1;M(16,36:48)=2;M(16,49:52)=1;M(16,53:63)=2;
M(17,1)=1;M(17,2)=2;M(17,3)=1;M(17,6)=1;M(17,25)=1;M(17,28)=1; ...
    M(17,35)=1;M(17,36:48)=2;M(17,49)=1;M(17,52)=1;M(17,53:63)=2;

```

```

M(18,1)=1;M(18,2)=2;M(18,3)=1;M(18,6)=1;M(18,25)=1;M(18,25)=1; ...
  M(18,34:35)=1;M(18,36:48)=2;M(18,49)=1;M(18,52)=1;M(18,53:63)=2;
M(19,1)=1;M(19,2)=2;M(19,3)=1;M(19,6)=1;M(19,25)=1;M(19,28)=1; ...
  M(19,35)=1;M(19,36:48)=2;M(19,49)=1;M(19,52)=1;M(19,53)=1; ...
  M(19,54:63)=2;
M(20,1)=1;M(20,2)=2;M(20,3)=1;M(20,6)=1;M(20,25)=1;M(20,35)=1; ...
  M(20,36:48)=2;M(20,49)=1;M(20,53)=1;M(20,54:63)=2;
M(21,1)=1;M(21,2)=2;M(21,3)=1;M(21,6)=1;M(21,25)=1;M(21,28:35)=1; ...
  M(21,36:48)=2;M(21,49)=1;M(21,52:53)=1;M(21,54:63)=2;
M(22,1)=1;M(22,2)=2;M(22,3)=1;M(22,6)=1;M(22,25)=1;M(22,28)=1; ...
  M(22,35)=1;M(22,36:48)=2;M(22,49)=1;M(22,52:53)=1;M(22,54:63)=2;
M(23,1)=1;M(23,2)=2;M(23,3)=1;M(23,6)=1;M(23,25)=1;M(23,34:35)=1; ...
  M(23,36:48)=2;M(23,49)=1;M(23,53)=1;M(23,54:63)=2;
M(24,1)=1;M(24,2)=2;M(24,3)=1;M(24,6)=1;M(24,25)=1;M(24,28)=1; ...
  M(24,35)=1;M(24,36:48)=2;M(24,49)=1;M(24,52:53)=1;M(24,54:63)=2;
M(25,1)=1;M(25,2)=2;M(25,3)=1;M(25,6)=1;M(25,25)=1;M(25,28)=1; ...
  M(25,34:35)=1;M(25,35)=1;M(25,36:48)=2;M(25,49)=1;M(25,52)=1; ...
  M(25,53:63)=2;
M(26,1)=1;M(26,2)=2;M(26,3)=1;M(26,6)=1;M(26,25)=1;M(26,28:35)=1; ...
  M(26,36:48)=2;M(26,49)=1;M(26,52)=1;M(26,53:63)=2;
M(27,1)=1;M(27,2)=2;M(27,3)=1;M(27,6)=1;M(27,25)=1;M(27,28)=1; ...
  M(27,35)=1;M(27,35)=1;M(27,36:48)=2;M(27,49)=1;M(27,52)=1; ...
  M(27,53:63)=2;
M(28,1)=1;M(28,2)=2;M(28,3)=1;M(28,28)=1;M(28,34:35)=1;M(28,36:48)=2; ...
  M(28,49)=1;M(28,52)=1;M(28,53:63)=2;
M(29,1)=1;M(29,2)=2;M(29,3)=1;M(29,34:35)=1;M(29,36:48)=2;M(29,49)=1; ...
  M(29,52)=1;M(29,53:63)=2;
M(30,1)=1;M(30,2)=2;M(30,3)=1;M(30,6)=1;M(30,25)=1;M(30,34:35)=1; ...
  M(30,36:48)=2;M(30,49)=1;M(30,52:53)=1;M(30,54:63)=2;
M(31,1)=1;M(31,2)=2;M(31,3)=1;M(31,6)=1;M(31,25)=1;M(31,28)=1; ...
  M(31,34:35)=1;M(31,36:48)=2;M(31,49)=1;M(31,53)=1;M(31,54:63)=2;
M(32,1)=1;M(32,2)=2;M(32,3)=1;M(32,6)=1;M(32,25)=1;M(32,28)=1; ...
  M(32,35)=1;M(32,36:48)=2;M(32,49)=1;M(32,52:53)=1;M(32,54:63)=2;
M(33,1)=1;M(33,2)=2;M(33,3)=1;M(33,6)=1;M(33,25)=1;M(33,28:35)=1; ...
  M(33,36:48)=2;M(33,49)=1;M(33,51:56)=1;M(33,57:63)=2;
M(34,1)=1;M(34,2)=2;M(34,3)=1;M(34,6)=1;M(34,25)=1;M(34,28)=1; ...
  M(34,34)=1;M(34,35:48)=2;M(34,49)=1;M(34,52)=1;M(34,56)=1; ...
  M(34,57:63)=2;
M(35,1)=1;M(35,2)=2;M(35,3)=1;M(35,6)=1;M(35,25)=1;M(35,28)=1; ...
  M(35,34)=1;M(35,35:43)=2;M(35,44:49)=1;M(35,56:60)=1;M(35,61:63)=2;
M(36,1)=1;M(36,2)=2;M(36,3)=1;M(36,6)=1;M(36,25)=1;M(36,28)=1; ...
  M(36,31)=1;M(36,34)=1;M(36,35:43)=2;M(36,44)=1;M(36,47)=1; ...
  M(36,49)=1;M(36,52)=1;M(36,60)=1;M(36,61:63)=2;
M(37,1)=1;M(37,2)=2;M(37,3)=1;M(37,6)=1;M(37,25)=1;M(37,28)=1; ...
  M(37,31)=1;M(37,34)=1;M(37,35:43)=2;M(37,44)=1;M(37,47)=1; ...
  M(37,49)=1;M(37,52)=1;M(37,59:60)=1;M(37,61:63)=2;
M(38,1)=1;M(38,2)=2;M(38,3)=1;M(38,6)=1;M(38,25)=1;M(38,28)=1; ...
  M(38,31)=1;M(38,34)=1;M(38,35:43)=2;M(38,44)=1;M(38,47)=1; ...
  M(38,49)=1;M(38,52)=1;M(38,59:60)=1;M(38,61:63)=2;
M(39,1)=1;M(39,2)=2;M(39,3)=1;M(39,6)=1;M(39,25)=1;M(39,28)=1; ...
  M(39,31)=1;M(39,34:44)=1;M(39,47)=1;M(39,49)=1;M(39,52)=1; ...
  M(39,60)=1;M(39,61:63)=2;
M(40,1)=1;M(40,2)=2;M(40,3)=1;M(40,6:14)=1;M(40,17:25)=1;M(40,28)=1; ...
  M(40,31)=1;M(40,34:35)=1;M(40,37:38)=1;M(40,40)=1;M(40,42)=1; ...
  M(40,44)=1;M(40,47)=1;M(40,49)=1;M(40,52)=1;M(40,59:60)=1; ...
  M(40,61:63)=2;
M(41,1)=1;M(41,2)=2;M(41,3)=1;M(41,59:60)=1;M(41,61:63)=2;
M(42,1)=1;M(42,2)=2;M(42,3)=1;M(42,60)=1;M(42,61:63)=2;
M(43,1)=1;M(43,2)=2;M(43,3)=1;M(43,25)=1;M(43,34)=1;M(43,52)=1; ...
  M(43,59:60)=1;M(43,61:63)=2;
M(44,1)=1;M(44,2)=2;M(44,3)=1;M(44,4:14)=1;M(44,16:22)=1;M(44,24:25)=1; ...
  M(44,27:28)=1;M(44,30:31)=1;M(44,33:35)=1;M(44,37:41)=1; ...
  M(44,42:44)=1;M(44,46:49)=1;M(44,52:59)=1;M(44,60:63)=2;

```

```

for i=34:40; M(i,29:30)=2; end
for i=45:64; M(i,1)=1; end

```





```

Aula7a = [52 45];
Nodo_Aula7a = find((Nodi(:,3)==Aula7a(1)) & (Nodi(:,2)==Aula7a(2)));

Aula7b = [52 49];
Nodo_Aula7b = find((Nodi(:,3)==Aula7b(1)) & (Nodi(:,2)==Aula7b(2)));

Aula8a = [52 42];
Nodo_Aula8a = find((Nodi(:,3)==Aula8a(1)) & (Nodi(:,2)==Aula8a(2)));

Aula8b = [52 35];
Nodo_Aula8b = find((Nodi(:,3)==Aula8b(1)) & (Nodi(:,2)==Aula8b(2)));

Dip_Meccanica_e_Aeronautica_a = [50 33];
Nodo_Dip_Meccanica_e_Aeronautica_a = find((Nodi(:,3)== ...
    Dip_Meccanica_e_Aeronautica_a(1)) & (Nodi(:,2)== ...
    Dip_Meccanica_e_Aeronautica_a(2)));

Aula18 = [23 47];
Nodo_Aula18 = find((Nodi(:,3)==Aula18(1)) & (Nodi(:,2)==Aula18(2)));

Ufficio_Erasmus = [11 47];
Nodo_Ufficio_Erasmus = find((Nodi(:,3)==Ufficio_Erasmus(1)) & ...
    (Nodi(:,2)==Ufficio_Erasmus(2)));

Chiostro = [18 29];
Nodo_Chiostro = find((Nodi(:,3)==Chiostro(1)) & (Nodi(:,2)==Chiostro(2)));

Aula17 = [28 30];
Nodo_Aula17 = find((Nodi(:,3)==Aula17(1)) & (Nodi(:,2)==Aula17(2)));

Aula16 = [28 23];
Nodo_Aula16 = find((Nodi(:,3)==Aula16(1)) & (Nodi(:,2)==Aula16(2)));

Dip_Meccanica_e_Aeronautica_b = [28 20];
Nodo_Dip_Meccanica_e_Aeronautica_b = find((Nodi(:,3)== ...
    Dip_Meccanica_e_Aeronautica_b(1)) & (Nodi(:,2)== ...
    Dip_Meccanica_e_Aeronautica_b(2)));

Sala_Chiostro = [25 10];
Nodo_Sala_Chiostro = find((Nodi(:,3)==Sala_Chiostro(1)) & ...
    (Nodi(:,2)==Sala_Chiostro(2)));

Saletta_Chiostro = [28 10];
Nodo_Saletta_Chiostro = find((Nodi(:,3)==Saletta_Chiostro(1)) & ...
    (Nodi(:,2)==Saletta_Chiostro(2)));

WC_Ingresso = [45 40];
Nodo_WC_Ingresso = find((Nodi(:,3)==WC_Ingresso(1)) & ...
    (Nodi(:,2)==WC_Ingresso(2)));

WC_Chiostro = [26 4];
Nodo_WC_Chiostro = find((Nodi(:,3)==WC_Chiostro(1)) & ...
    (Nodi(:,2)==WC_Chiostro(2)));

Mappa_Nodi_Punti_Interesse = [Nodo_Ingresso; Nodo_Uscita; ...
    Nodo_Portineria; Nodo_Secondo_Piano;Nodo_Aula1;...
    Nodo_Aula2;Nodo_Aula5;Nodo_Aula7a;Nodo_Aula7b;Nodo_Aula8a; ...
    Nodo_Aula8b;Nodo_Dip_Meccanica_e_Aeronautica_a;...
    Nodo_Aula18;Nodo_Ufficio_Erasmus;Nodo_Chiostro;Nodo_Aula17; ...
    Nodo_Aula16;Nodo_Dip_Meccanica_e_Aeronautica_b;...
    Nodo_Sala_Chiostro;Nodo_Saletta_Chiostro;Nodo_WC_Ingresso; ...
    Nodo_WC_Chiostro];

Mappa_Punti_Interesse = [Ingresso; Uscita;Portineria;Secondo_Piano; ...

```

```

Aula1;Aula2;Aula5;Aula7a;Aula7b;Aula8a;Aula8b; ...
Dip_Meccanica_e_Aeronautica_a;Aula18;Ufficio_Erasmus;Chiostro; ...
Aula17;Aula16;Dip_Meccanica_e_Aeronautica_b;Sala_Chiostro; ...
Saletta_Chiostro;WC_Ingresso;WC_Chiostro];

plot(Mappa_Punti_Interesse(:,1), ...
      Mappa_Punti_Interesse(:,2),'co','linewidth',6);

%%%%%%%%%% Inizio variazione punto di arrivo
%
% Se vogliamo variare il punto di arrivo ogni volta che compiliamo lo
% script, bisogna eliminare la spunta "commento" dalle righe che seguono

% fprintf('Digita il codice del posto che vuoi raggiungere:\n');
% fprintf(['Ingresso: cod. %d \nUscita: cod. %d\nPortineria: cod. ' ...
%         '%d\nSecondo_Piano: cod. %d\nAula1: cod. %d\nAula2: cod. ' ...
%         '%d\nAula5: cod. %d\nAula7a: cod. %d\nAula7b: cod. %d\nAula8a: ' ...
%         '%d\nAula8b: cod. %d\nDip_Meccanica_e_Aeronautica_a: cod. ' ...
%         '%d\nAula18: cod. %d\nUfficio_Erasmus: cod. %d\nChiostro: cod. ' ...
%         '%d\nAula17: cod. %d\nAula16: cod. ' ...
%         '%d\nDip_Meccanica_e_Aeronautica_b: cod. %d\nSala_Chiostro: ' ...
%         '%d\nSaletta_Chiostro: cod. %d\nWC_Ingresso: cod. ' ...
%         '%d\nWC_Chiostro: cod. %d\n'],Nodo_Ingresso, Nodo_Uscita, ...
%         Nodo_Portineria, Nodo_Secondo_Piano, Nodo_Aula1, Nodo_Aula2, ...
%         Nodo_Aula5, Nodo_Aula7a, Nodo_Aula7b, Nodo_Aula8a, Nodo_Aula8b, ...
%         Nodo_Dip_Meccanica_e_Aeronautica_a, Nodo_Aula18, ...
%         Nodo_Ufficio_Erasmus, Nodo_Chiostro, Nodo_Aula17, Nodo_Aula16, ...
%         Nodo_Dip_Meccanica_e_Aeronautica_b, Nodo_Sala_Chiostro, ...
%         Nodo_Saletta_Chiostro, Nodo_WC_Ingresso, Nodo_WC_Chiostro);
nodo_arrivo = 1470;


```

```

%         N2 = Nodi(id_temp,1);
%         vect_temp = [id N1 N2];
%     end
%     Link = [Link; vect_temp];
% end
% vect_temp = [];
%
%% NODO NORD-EST %%
% Se non siamo nello spigolo in alto a Dx
% if ((x - 1) >= 1) && ((y+1) <= length(M(1,:)));
%     if (M(x-1,y+1))==0;           % Se il punto in alto a Dx e' un nodo
%         id = id + 1;             % Genera ID Link
%                                   % Cerca nella matrice Nodi
%         id_temp = find(Nodi(:,2)==(x-1) & ...
%             Nodi(:,3)==(y+1),1,'first');
%         N1 = i;
%         N2 = Nodi(id_temp,1);
%         vect_temp = [id N1 N2];
%     end
%     Link = [Link; vect_temp];
% end
% vect_temp = [];
%
%% NODO NORD %%
% if (x - 1 >= 1);                % Se non siamo sul bordo in alto
%     if (M(x-1,y))==0;           % Se il punto in alto e' un nodo
%         id = id + 1;           % Genera ID Link
%                                   % Cerca nella matrice Nodi:
%         id_temp = find(Nodi(:,2)==(x-1) & Nodi(:,3)==(y),1,'first');
%         N1 = i;
%         N2 = Nodi(id_temp,1);
%         vect_temp = [id N1 N2];
%     end
%     Link = [Link; vect_temp];
% end
% vect_temp = [];
%
%% NODO NORD-OVEST %%
% Se non siamo nello spigolo in alto a Sx
% if ((x - 1)>= 1) && ((y-1) >= 1);
%     if (M(x-1,y-1))==0;        % Se il punto in alto a Sx e' un nodo
%         id = id + 1;          % Genera ID Link
%                                   % Cerca nella matrice Nodi:
%         id_temp = find(Nodi(:,2)==(x-1) & ...
%             Nodi(:,3)==(y-1),1,'first');
%         N1 = i;
%         N2 = Nodi(id_temp,1);
%         vect_temp = [id N1 N2];
%     end
%     Link = [Link; vect_temp];
% end
% vect_temp = [];
%
%% NODO OVEST %%
% if ((y - 1)>= 1)                % Se non siamo sul bordo Sx
%     if (M(x,y-1))==0;          % Se il punto a Sx e' un nodo
%         id = id + 1;          % Genera ID Link
%                                   %Cerca nella matrice Nodi:
%         id_temp = find(Nodi(:,2)==(x) & Nodi(:,3)==(y-1),1,'first');
%         N1 = i;
%         N2 = Nodi(id_temp,1);
%         vect_temp = [id N1 N2];
%     end
%     Link = [Link; vect_temp];
% end
% vect_temp = [];

```

```

%%
%% NODO SUD-OVEST %%
% Se non siamo nello spigolo in basso a Sx
if ((x + 1) <= length(M(:,1))) && ((y-1) >= 1)
    if (M(x+1,y-1))==0; % Se il punto in basso a Sx e' un nodo
        id = id + 1; % Genera ID Link
        % Cerca nella matrice Nodi:
        id_temp = find(Nodi(:,2)==(x+1) & ...
            Nodi(:,3)==(y-1),1,'first');
        N1 = i;
        N2 = Nodi(id_temp,1);
        vect_temp = [id N1 N2];
    end
    Link = [Link; vect_temp];
end
vect_temp = [];

%% NODO SUD %%
% Se non siamo sul bordo in basso
if ((x + 1) <= length(M(:,1)));
    if (M(x+1,y))==0; % Se il punto in basso e' un nodo
        id = id + 1; % Genera ID Link
        % Cerca nella matrice Nodi:
        id_temp = find(Nodi(:,2)==(x+1) & Nodi(:,3)==(y),1,'first');
        N1 = i;
        N2 = Nodi(id_temp,1);
        vect_temp = [id N1 N2];
    end
    Link = [Link; vect_temp];
end
vect_temp = [];

%% NODO SUD-OVEST %%
% Se non siamo nello spigolo in basso a Dx
if ((x + 1) <= length(M(:,1)) && (y + 1) <= length(M(1,:)));
    if (M(x+1,y+1))==0; % Se il punto in basso a Dx e' un nodo
        id = id + 1; % Genera ID Link
        % Cerca nella matrice Nodi:
        id_temp = find(Nodi(:,2)==(x+1) & Nodi(:,3)==(y+1),1,'first');
        N1 = i;
        N2 = Nodi(id_temp,1);
        vect_temp = [id N1 N2];
    end
    Link = [Link; vect_temp];
end
vect_temp = [];

%% end

%% Introduciamo gli ACCESS POINT %%
% Introduciamo 7 access point in posizione random

ap_x = [51 16 30 27 17 51 4]; % Coordinate APs asse delle x
ap_y = [43 43 51 28 8 24 14]; % Coordinate APs asse delle y

ap = [ap_x; ap_y]; % Matrice APs
ap = ap';

% Posizioniamo gli APs nella griglia 64x64
AP_matrix = zeros(64,64);
AP_matrix(43,51)=1;AP_matrix(43,16)=1;AP_matrix(51,30)=1; ...
    AP_matrix(28,27)=1;AP_matrix(8,17)=1; ...
    AP_matrix(24,51)=1;AP_matrix(14,4)=1;

% AP e' la matrice degli APs ordinati per ID
AP = crea_nodi_da_matrice_mod(AP_matrix);

```



```

n_ap = length(ap_x);
plot(ap(:,1),ap(:,2), 'r*', 'linewidth', 3);

hold on

%% Introduciamo la griglia SCALABILE per il fingerprinting

% Se il passo della griglia per il fingerprinting e' uguale al passo della
% griglia dei nodi (cioe' il minimo possibile nel nostro sistema), assegnamo
% le stesse coordinate: fingerprint = Nodi.

if (w_x == 1) && (w_y == 1)

    fingerprint = Nodi;           % In queste condizioni, la griglia
    fingerprint_x = Nodi(:,3);    % del fingerprinting coincide con
    fingerprint_y = Nodi(:,2);    % quella dei nodi

end

count_griglia = 0;
i = 1;
j = 1;

% Questo ciclo viene utilizzato soltanto per calcolare la griglia a maglie
% piu' larghe di 1x1

if (w_x ~= 1) || (w_y ~=1);
% Fino a quando non arrivo alla fine della griglia scorrendola
% verso il basso
while (i<=length(M(:,1)))
% Fino a quando non arrivo alla fine della griglia scorrendola
% verso destra
while (j<=length(M(:,1)))
% Se trovo un punto "calpestabile" lo assegno ad un punto
% della griglia
if M(i,j)==0;
count_griglia = count_griglia + 1;
fingerprint_x(count_griglia) = j;
fingerprint_y(count_griglia) = i;
end

% Se trovo un punto "non calpestabile" o un "muro"
if (M(i,j)==1);
% Se non mi trovo sul bordo destro
if (j~=length(M(:,1)));
if (M(i,j+1))==0;
% Controllo che il punto successivo sia
% calpestabile, se la risposta e' si, lo assegno
% ad un punto della griglia
count_griglia = count_griglia + 1;
fingerprint_x(count_griglia) = j+1;
fingerprint_y(count_griglia) = i;
end

% Se non mi trovo sul bordo sinistro
if (j~=1)
if (M(i,j-1))==0;
% Controllo che il punto precedente sia
% calpestabile, se la risposta e' si, lo assegno
% ad un punto della griglia
count_griglia = count_griglia + 1;
fingerprint_x(count_griglia) = j-1;
fingerprint_y(count_griglia) = i;
end
end
end
end
end

```

```

        end
    end
end
    j = j + w_x;
end
    i = i + w_y;
    j = 1;
end

%%%%%%%%%% Creo la matrice Nx3 della griglia --> fingerprint %%%%%%%%%%

fingerprint(:,1) = 1:1:count_griglia;
fingerprint(:,3) = fingerprint_x;
fingerprint(:,2) = fingerprint_y;

mat_num_wall = zeros(n_ap, length(fingerprint_x));

    for h = 1:length(fingerprint_x)
        mat_num_wall_temp = find((Nodi(:,2)==fingerprint(h,2)) ...
            & (Nodi(:,3)==fingerprint(h,3)));
        mat_num_wall(:,h) = mat_num_wall_1_1(:,mat_num_wall_temp);
    end

end % Fine del ciclo che considera la maglia
    % della griglia superiore al passo 1

%%%%%%%%%%
%% PLOT DELLA GRIGLIA %%
% A prescindere dalla spaziatura nella griglia, la disegniamo sulla mappa.
% Nell'ultima versione del codice questo comando sara' commentato in quanto
% la visione della griglia sulla mappa impedisce la visualizzazione esatta
% della stima della posizione dell'utente e dei suoi eventuali spostamenti.

plot(fingerprint(:,3),fingerprint(:,2),'k.');
```

% Plot della griglia

```

%% Fingerprinting con pathloss
%
% Calcoliamo la distanza degli ACCESS POINT da ogni punto della griglia
% del fingerprinting. la matrice "distanze".
%
% size(distanze) = length(fingerprint_x) x n_ap
%

% Inizializzazione a zero matrice delle distanze
distanze = zeros(n_ap,length(fingerprint_x));
for i = 1:n_ap
    for j = 1:length(fingerprint_x)
        distanze(i,j)=calcoladistanza(fingerprint(j,2), ...
            fingerprint(j,3),AP(i,2),AP(i,3));
        if distanze(i,j) == 0
            distanze(i,j) = 1; % Condizione necessaria in quanto nel calcolo
                % del pathloss abbiamo un logaritmo, e quando
                % nel variare il passo della griglia un punto
                % della stessa coincide con un AP, avremo una
                % distanza nulla e quindi un valore del
                % logaritmo pari ad infinito.
        end
    end
end
end

%% MATRICE Los_d %%
% Fattore Costante
lc = -47.4; % [dB]
```

```

% Attenuazione che si ha oltrepassando un muro di (20-40) cm.
l2 = 3.9; % [dB]

% lc = 0; % Se vogliamo testare il sistema mettiamo a zero lc ed l2
% l2 = 0;

alpha = 2; % Esponente del path_loss --> Impostiamo a 2

% La matrice Los_d rappresenta tutte le perdite in potenza soltanto in
% funzione della distanza tra Tx e Rx, e ci servira' nell'implementazione
% del modello multi wall.

Los_d = 10*alpha*log(distanze);

%% MURI ATTRAVERSATI
% Generiamo la matrice mat_num_wall (n_ap x length(fingerprint_x))
% all'interno della quale calcoliamo in ogni elemento il numero di muri
% attraversati andando dall'i-esimo AP verso il j-esimo punto della griglia

% Come vediamo anche questa parte di codice e' commentata. Abbiamo
% calcolato quanti muri vengono oltrepassati da un segnale che parta
% dalj-esimo punto della griglia e arrivi all'i-esimo ACCESS POINT.
% E' una procedura molto lunga, come accenato per la costruzione delle
% matrici di nodi e link, per questo abbiamo effettuato una simulazione e
% poi abbiamo salvato i risultati ottenuti. All'inizio dello script e'
% infatti caricata una matrice "mat_num_wall" che contiene appunto tutti
% i nostri risultati.
%
% size(mat_num_wall) = num_nodi x n_ap
%

% passo_retta = 100;
%
% Questo vettore (x_M) ci serve per entrare nella matrice M:
% x_M = linspace(1,64,64);

% num_wall = 0; % Contatore del numero dei muri a zero
% a=0;
% d_min=10^4; % Pongo la distanza minima ad un valore inaccessibile per
% % poter effettuare i controlli all'interno dello script
% for i = 1:n_ap
%
%     for j = 1:length(fingerprint_x)
%
%         count_wall_temp=zeros(size(M,1),size(M,2));
%
%         x0 = fingerprint(j,3); % coordinata x del j-esimo punto
%                                % della griglia
%         y0 = fingerprint(j,2); % coordinata y del j-esimo punto
%                                % della griglia
%         x1 = AP(i,3);          % coordinata x dell' i-esimo access point
%         y1 = AP(i,2);          % coordinata y dell' i-esimo access point
%
%         passo_retta_temp = calcoladistanza(x0,y0,x1,y1);
%         passo_retta = (ceil(10*passo_retta_temp));
%
%         % Crea due vettori x e y con le coordinate della retta passate
%         % per i punti (x0,y0) e (x1,y1):
%         [x y] = crea_retta(x0,y0,x1,y1,passo_retta);
%         plot(x,y)
%
%         for w = 1:passo_retta % Per ogni punto della retta (x,y)
%
%             x_confronto = x(w);
%             y_confronto = y(w);
%             plot(x_confronto,y_confronto,'go','linewidth',4)

```

```

% %
% pause
%
% Questa serie di condizioni e' necessaria er poter
% individuare la regione all'interno della quale cadono i
% punti da analizzare. In base al coefficiene angolare della
% retta in questione, ci aspettiamo infatti 4 possibili
% scenari (commentiamo soltanto il primo dei 4):
%
% if (x(1)>x(passo_retta) && y(1)>y(passo_retta))
% for q = ceil(x(passo_retta)):ceil(x(1))
%     for z = ceil(y(passo_retta)):ceil(y(1))
%         % Distanza tra punto della griglia appartenente alla
%         % regione di intgeresse e w-esimo punto della retta
%         % x-y:
%         prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
%             x_confronto, y_confronto);
%
%         % Se la distanza calcolata e' minore della precedente
%         if prova_distanza < d_min
%
%             % Aggiorna il valore di d_min
%             d_min = prova_distanza;
%
%             x_ok = q; % Aggiorna la coordinata x
%                 % corrispondente a d_min
%
%             y_ok = z; % Aggiorna la coordinata y
%                 % corrispondente a d_min
%
%         end
%     end
% end
%
% Se sulla matrice M (mappa) il punto (x_ok,y_ok)
% corrisponde ad un 1 e se + la prima volta che entriamo su
% questo punto della mappa
% if (M(y_ok,x_ok)==1) && (count_wall_temp(y_ok,x_ok)==0)
%
%     num_wall = num_wall + 1; % Aggiorno il contatore
%         % dei muri aggiungendone 1
%     count_wall_temp(y_ok,x_ok)=1; % Avverto che questo muro
%         % e' stato conteggiato per
%         % non incorrere nell'
%         % errore di contare n
%         % volte lo stesso ostacolo
%
% end
% %
% plot(x_ok,y_ok,'go','linewidth',4)
% d_min=10^4;
% end
%
% if (x(1)>x(passo_retta) && y(1)<y(passo_retta))
% for q = ceil(x(passo_retta)):ceil(x(1))
%     for z = ceil(y(1)):ceil(y(passo_retta))
%         prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
%             x_confronto, y_confronto);
%         if prova_distanza < d_min
%             d_min = prova_distanza;
%             x_ok = q;
%             y_ok = z;
%         end
%     end
% end
%
% if (M(y_ok,x_ok)==1) && (count_wall_temp(y_ok,x_ok)==0)
%     num_wall = num_wall + 1;
%     count_wall_temp(y_ok,x_ok)=1;
%
% end

```

```

% %
%     plot(x_ok,y_ok,'go','linewidth',4)
%     d_min=10^4;
%     end
%
%     if (x(1)<x(passo_retta) && y(1)>y(passo_retta))
%     for q = ceil(x(1)):ceil(x(passo_retta))
%         for z = ceil(y(passo_retta)):ceil(y(1))
%             prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
%                 x_confronto, y_confronto);
%             if prova_distanza < d_min
%                 d_min = prova_distanza;
%                 x_ok = q;
%                 y_ok = z;
%             end
%         end
%     end
%     end
%     if (M(y_ok,x_ok)==1) && (count_wall_temp(y_ok,x_ok)==0)
%         num_wall = num_wall + 1;
%         count_wall_temp(y_ok,x_ok)=1;
%     end
% %
%     plot(x_ok,y_ok,'go','linewidth',4)
%     d_min=10^4;
%     end
%
%     if (x(1)<x(passo_retta) && y(1)<y(passo_retta))
%     for q = ceil(x(1)):ceil(x(passo_retta))
%         for z = ceil(y(1)):ceil(y(passo_retta))
%             prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
%                 x_confronto, y_confronto);
%             if prova_distanza < d_min
%                 d_min = prova_distanza;
%                 x_ok = q;
%                 y_ok = z;
%             end
%         end
%     end
%     end
%     if (M(y_ok,x_ok)==1) && (count_wall_temp(y_ok,x_ok)==0)
%         num_wall = num_wall + 1;
%         count_wall_temp(y_ok,x_ok)=1;
%     end
% %
%     plot(x_ok,y_ok,'go','linewidth',4)
%     d_min=10^4;
%     end
%
%     end
%     % Matrice contenente il numero di muri attraversati in ogni
%     % percorso
%     mat_num_wall(i,j) = num_wall;
%     num_wall = 0;
%
%     end
%     a=a+1;
%     fprintf('Percentuale calcolo matrice mat_num_wall: %d/7\n',a);
% end
%
%     x_ok = 0;
%     y_ok = 0;
%     fprintf('\n\n Matrice mat_num_wall --> OK!! \n\n');
%
%% Applichiamo definitivamente il modello MULTI-WALL
%
% L(d) = Los(d) + Mw(dB)
% Mw(dB) = lc + sum(num_wall*l2)
% lc contiene anche il valore di lo

```

```

%% MATRICE S %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La matrice S e' effettivamente la matrice che contiene tutti gli L(d)
% del modello multi wall:
%
% Los_d = Los_d';
% size(S) = numero punti della griglia x numero di access point

% Prima di calcolare la matrice S generiamo una matrice della stassa
% dimensione ma che contenga il fattore di SHADOWING:

delta_shadowing_S = delta_min + (delta_max - delta_min) ...
    .*rand(length(fingerprint_x),n_ap);
% L_d = Distanza in potenza (dB) tra l'i-esimo AP e il j-esimo punto della
% griglia:
L_d = Los_d + lc + (mat_num_wall*12);
L_d = L_d';

S = L_d + delta_shadowing_S; % Matrice S degli RSS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Distanze di ogni utente dagli Access Point

dist_utenti = zeros(n_ap,utenti); % Matrice delle distanze
for i = 1:n_ap
    % Con questo ciclo calcoliamo la distanza degli utenti da tutti i punti
    % della griglia:
    for j = 1:utenti

        dist_utenti(i,j)=calcoladistanza(utente_y(j), ...
            utente_x(j),AP(i,2),AP(i,3));
    end
end

Los_d_utente = 10*alpha*log(dist_utenti);
Los_d_utente = Los_d_utente';

%% Matrice delle potenze ricevute sugli Access Point da ogni utente

% Stesso ciclo precedente, molto piu' veloce in quanto il numero di utenti
% e' minore del numero di punti sulla griglia

% Questa parte commentata invece calcola il numero di muri che
% oltrepassa il segnale che parte dalla posizione dell'utente e arriva su
% tutti gli ACCESS POINT. Dato che nel nostro modello le posizioni
% dell'utente che andiamo a stimare appartengono sicuramente a coordinate
% della matrice dei nodi, questo risultato si puo' ottenere in maniera
% diversa come si vede alla fine della parte commentata.

% num_wall_utente = 0; % Contatore del numero dei muri a zero
% d_min=10^4; % Pondo la distanza minima ad un valore inaccessibile per
% % poter effettuare i controlli all'interno dello script
% b=0;
% q=0;
% z=0;
% for i = 1:n_ap
%
%     for j = 1:utenti
%
%         count_wall_utente_temp = zeros(size(M,1),size(M,2));
%
%         x0_utente = utente_x(j); % coordinata x del j-esimo utente
%         y0_utente = utente_y(j); % coordinata y del j-esimo utente
%         x1 = AP(i,3); % coordinata x dell' i-esimo
% % access point
%         y1 = AP(i,2); % coordinata y dell' i-esimo

```

```

% access point
%
% passo_retta_temp_utente = calcoladistanza(x0_utente, ...
% y0_utente,x1,y1);
% passo_retta = (ceil(16*passo_retta_temp_utente));
%
% Crea due vettori x e y con le coordinate della retta passate
% per i punti (x0,y0) e (x1,y1):
% [x_user y_user] = crea_retta(x0_utente, ...
% y0_utente,x1,y1,passo_retta);
% plot(x_user,y_user)
%
% for w = 1:passo_retta % Per ogni punto della retta (x,y)
% x_confronto = x_user(w);
% y_confronto = y_user(w);
% % plot(x_confronto,y_confronto,'go','linewidth',4)
% % pause
%
% % Questa serie di condizioni e' necessaria per poter individuare la
% % regione all'interno della quale cadono i punti da analizzare. In base
% % al coefficiente angolare della retta in questione, ci aspettiamo infatti
% % 4 possibili scenari (commentiamo soltanto il primo dei 4):
%
% if (x_user(1)>x_user(passo_retta) && ...
% y_user(1)>y_user(passo_retta))
% for q = ceil(x_user(passo_retta)):ceil(x_user(1))
% for z = ceil(y_user(passo_retta)):ceil(y_user(1))
%
% % Distanza tra punto della griglia appartenente alla
% % regione di interesse e w-esimo
% % punto della retta x-y:
% prova_distanza = calcoladistanza(x_M(q), ...
% x_M(z), x_confronto, y_confronto);
%
% % Se la distanza calcolata e' minore della precedente
% if prova_distanza < d_min
% % Aggiorna il valore di d_min
% d_min = prova_distanza;
%
% % Aggiorna la coordinata x
% % corrispondente a d_min:
% x_ok = q;
%
% % Aggiorna la coordinata y
% % corrispondente a d_min
% y_ok = z;
% end
% end
% end
% % Se sulla matrice M (mappa) il punto (x_ok,y_ok)
% % corrisponde ad un 1 e se + la prima volta che entriamo su
% % questo punto della mappa:
% if (M(y_ok,x_ok)==1) && ...
% (count_wall_utente_temp(y_ok,x_ok)==0)
%
% % Aggiorno il contatore dei muri aggiungendone uno
% num_wall_utente = num_wall_utente + 1;
%
% % Avverto che questo muro e' stato conteggiato per non
% % incorrere nell'errore di contare n volte lo stesso
% % ostacolo
% count_wall_utente_temp(y_ok,x_ok) = 1;
% end
% % plot(x_ok,y_ok,'go','linewidth',4)
% % pause
% d_min=10^4;

```

```

end
%
%
%
if (x_user(1)>x_user(passo_retta) ...
    && y_user(1)<y_user(passo_retta))
for q = ceil(x_user(passo_retta)):ceil(x_user(1))
    for z = ceil(y_user(1)):ceil(y_user(passo_retta))
        prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
            x_confronto, y_confronto);
        if prova_distanza < d_min
            d_min = prova_distanza;
            x_ok = q;
            y_ok = z;
        end
    end
end
end
% Se sulla matrice M (mappa) il punto (x_ok,y_ok)
% corrisponde ad un 1 e se + la prima volta che entriamo su
% questo punto della mappa:
if (M(y_ok,x_ok)==1) && ...
    (count_wall_utente_temp(y_ok,x_ok)==0)

    % Aggiorno il contatore dei muri aggiungendone uno
    num_wall_utente = num_wall_utente + 1;

    % Avverto che questo muro e' stato conteggiato per non
    % incorrere nell'errore di contare n volte lo stesso
    % ostacolo
    count_wall_utente_temp(y_ok,x_ok)=1;
end
% %
% %
plot(x_ok,y_ok,'go','linewidth',4)
pause
d_min=10^4;
end

if (x_user(1)<x_user(passo_retta) && ...
    y_user(1)>y_user(passo_retta))
for q = ceil(x_user(1)):ceil(x_user(passo_retta))
    for z = ceil(y_user(passo_retta)):ceil(y_user(1))
        prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
            x_confronto, y_confronto);
        if prova_distanza < d_min
            d_min = prova_distanza;
            x_ok = q;
            y_ok = z;
        end
    end
end
end
end
% Se sulla matrice M (mappa) il punto (x_ok,y_ok)
% corrisponde ad un 1 e se + la prima volta che entriamo su
% questo punto della mappa:
if (M(y_ok,x_ok)==1) && ...
    (count_wall_utente_temp(y_ok,x_ok)==0)

    % Aggiorno il contatore dei muri aggiungendone uno
    num_wall_utente = num_wall_utente + 1;

    % Avverto che questo muro e' stato conteggiato per non
    % incorrere nell'errore di contare n volte lo stesso
    % ostacolo
    count_wall_utente_temp(y_ok,x_ok)=1;
end
% %
% %
plot(x_ok,y_ok,'go','linewidth',4)
pause

```



```

%           d_min=10^4;
%           end
%
%           if (x_user(1)<x_user(passo_retta) && ...
%               y_user(1)<y_user(passo_retta))
%           for q = ceil(x_user(1)):ceil(x_user(passo_retta))
%               for z = ceil(y_user(1)):ceil(y_user(passo_retta))
%                   prova_distanza = calcoladistanza(x_M(q), x_M(z), ...
%                       x_confronto, y_confronto);
%                   if prova_distanza < d_min
%                       d_min = prova_distanza;
%                       x_ok = q;
%                       y_ok = z;
%                   end
%               end
%           end
%           end
%           % Se sulla matrice M (mappa) il punto (x_ok,y_ok)
%           % corrisponde ad un 1 e se + la prima volta che entriamo su
%           % questo punto della mappa:
%           if (M(y_ok,x_ok)==1) && ...
%               (count_wall_utente_temp(y_ok,x_ok)==0)
%
%               % Aggiorno il contatore dei muri aggiungendone uno
%               num_wall_utente = num_wall_utente + 1;
%
%               % Avverto che questo muro e' stato conteggiato per non
%               % incorrere nell'errore di contare n volte lo stesso
%               % ostacolo
%               count_wall_utente_temp(y_ok,x_ok)=1;
%           end
%           %
%           plot(x_ok,y_ok,'go','linewidth',4)
%           %
%           pause
%           d_min=10^4;
%           end
%       end
%
%       % Matrice contenente il numero di muri attraversati in ogni
%       % percorso:
%       mat_num_wall_utente(i,j) = num_wall_utente;
%       num_wall_utente = 0;
%   end
%   b=b+1;
%   fprintf('Percentuale calcolo matrice mat_num_wall_utente: %d/7\n',b);
% end
%   fprintf('\n\n Matrice mat_num_wall_utente --> OK!! \n\n');

%Calcolo MURI ATTRAVERSATI UTENTE

mat_num_wall_utente_temp = find((Nodi(:,2)==utente_y) & ...
    (Nodi(:,3)==utente_x));
mat_num_wall_utente = mat_num_wall_1_1(:,mat_num_wall_utente_temp);
Los_d_utente = Los_d_utente';
%% MATRICE OMEGA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% size(omega) = numero utenti x numero access point
delta_shadowing_omega = delta_min + ...
    (delta_max-delta_min).*rand(utenti,n_ap);

% L_d_omega = Distanza in potenza (dB) tra l'i-esimo AP e il j-esimo utente
L_d_omega = Los_d_utente + lc + (mat_num_wall_utente*l2);
L_d_omega = L_d_omega';
omega = L_d_omega + delta_shadowing_omega;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calcolo le minime distanze Euclidee

```

```

% Dobbiamo confrontare le righe della matrice S con le colonne della
% matrice omega. E' importante considerare che stiamo parlando di PathLoss,
% quindi, piu' il valore sara' elevato, piu' distanti saranno gli utenti
% dall'access point.

% Dimensioni della matrice S --> [length(x) X n_ap]
% Dimensioni della matrice omega --> [utenti X n_ap]

% [D] = Min_Dist_Euclidea_mod(num_utenti, ...
%   num_punti_griglia, Matrice_RSS,Matrice_omega)

D = Min_Dist_Euclidea_mod(utenti, length(fingerprint_x), S, omega);
D = D';

% Dimensione della matrice D --> [length(x) X utenti]

%% Posizionamento READER RFID %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Per cercare di migliorare il nostro sistema, prima di applicare
% l'algoritmo KNN, aggiungo una seri (31) di READER UHF RFID,
% posizionandoli come se fossimo nella realt  , quindi, a ridosso dei muri,
% in quanto abbiamo bisogno di un'alimentazione di corrente elettrica per il
% loro funzionamento.

RFID_matrix = [1 11 12;1 10 12;1 10 13;1 11 13;1 12 12;1 12 13;
2 12 19;2 11 18;2 11 19;2 11 20;2 12 18;2 12 20;
3 10 25;3 9 24;3 9 26;3 10 24;3 10 26;3 11 24;3 11 26;
4 10 28;4 9 27;4 9 29;4 10 27;4 10 29;4 11 27;4 11 29;
5 17 10;5 17 9;5 17 11;5 18 9;5 18 10;5 18 11;
6 17 18;6 17 17;6 17 19;6 18 17;6 18 18;6 18 19;
7 16 31;7 15 32;7 16 30;7 16 32;7 17 30;7 17 31;7 17 32;
8 20 26;8 19 26;8 19 27;8 20 27;8 21 26;8 21 27;
9 22 31;9 22 30;9 22 32;9 23 30;9 23 31;9 23 32;
10 18 50; 10 17 50; 10 17 51; 10 18 51; 10 19 50; 10 19 51;
11 27 7; 11 26 7; 11 26 8; 11 27 8; 11 28 6; 11 28 7; 11 28 8;
12 27 24; 12 26 23; 12 26 24; 12 27 23; 12 28 23; 12 28 24; 12 28 25;
13 27 31; 13 27 30; 13 27 32; 13 28 30; 13 28 31; 13 28 32;
14 32 31; 14 31 30; 14 31 31; 14 31 32; 14 32 30; 14 32 32;
15 34 24; 15 33 23; 15 33 24; 15 34 23; 15 35 24; 15 35 23;
16 36 27; 16 35 26; 16 35 27; 16 36 26; 16 37 26; 16 37 27;
17 34 54; 17 34 53; 17 34 55; 17 35 53; 17 35 54; 17 35 55;
18 36 7; 18 35 7; 18 35 8; 18 36 8; 18 37 7; 18 37 8; 19 39 18;
19 38 17; 19 38 18; 19 38 19; 19 39 17; 19 39 19;
20 41 6; 20 40 5; 20 41 5; 20 41 7; 20 42 5; 20 42 6; 20 42 7;
21 43 39; 21 42 38; 21 42 39; 21 42 40; 21 43 38; 21 43 40;
22 43 55; 22 42 54; 22 42 55; 22 42 56; 22 43 54; 22 43 56;
23 49 20; 23 48 19; 23 48 20; 23 48 21; 23 49 19; 23 50 19; 23 50 20;
24 49 26; 24 48 26; 24 48 27; 24 49 27; 24 50 26; 24 50 27;
25 49 33; 25 48 32; 25 48 33; 25 49 32; 25 50 32; 25 50 33;
26 48 39; 26 47 38; 26 47 39; 26 48 38; 26 49 38; 26 49 39;
27 48 41; 27 47 41; 27 47 42; 27 48 42; 27 49 41; 27 49 42;
28 50 55; 28 49 54; 28 49 55; 28 49 56; 28 50 54; 28 50 56;
29 56 26; 29 55 26; 29 55 27; 29 56 27; 29 57 26; 29 57 27;
30 56 33; 30 55 32; 30 55 33; 30 56 32; 30 57 32; 30 57 33;
31 56 50; 31 55 50; 31 55 51; 31 56 51; 31 57 50; 31 57 51];

plot(RFID_matrix(:,3),RFID_matrix(:,2), 'go');

target_rfid_ok = find(RFID_matrix(:,2) == utente_y ...
& RFID_matrix(:,3) == utente_x);

if isempty(target_rfid_ok) == 0;
vettore_neighbour_pos = find(RFID_matrix(:,1) == ...
RFID_matrix(target_rfid_ok,1));
n_neighbour = length(vettore_neighbour_pos);
stima_pos_y = (1/n_neighbour)*sum(RFID_matrix(min...

```

```

        (vettore_neighbour_pos):max(vettore_neighbour_pos),2));
    stima_pos_x = (1/n_neighbour)*sum(RFID_matrix(min...
        (vettore_neighbour_pos):max(vettore_neighbour_pos),3));
    rfid = rfid + 1;

end

%% Applichiamo l'algoritmo KNN: K Nearest Neighbour
if isempty(target_rfid_ok) == 1;
K = 5;

for i = 1:utenti

    vettore_neighbour = K_NN_multi_wall_temp(D(:,i), K);

    for j = 1:length(vettore_neighbour)
        k_y(j) = fingerprint_x(vettore_neighbour(j));
        k_x(j) = fingerprint_y(vettore_neighbour(j));
    end
    k_x;
    k_y;
    stima_pos_y(i) = (1/length(vettore_neighbour))*(sum(k_y));
    stima_pos_x(i) = (1/length(vettore_neighbour))*(sum(k_x));
    errore_pos_iniziale(qw) = calcoladistanza(utente_x, ...
        utente_y, stima_pos_y(i),stima_pos_x(i));

    k_x=[];
    k_y=[];
end
end
plot(stima_pos_y,stima_pos_x,'bx','linewidth', 5);
% legend('Griglia','Access Point','Utenti','Posizione stimata degli
% utenti','location','BestOutside');

end

%% Calcolo SMALLEST PATH %%
[dist,path] = dijkstra(Nodi,Link,nodo_partenza,nodo_arrivo);

for n = 2:length(path)
    plot(Nodi(path(n-1:n),3),Nodi(path(n-1:n),2),'b-','linewidth',4);
end

plot(Nodi(nodo_partenza,3),Nodi(nodo_partenza,2),'b*')
plot(Nodi(nodo_arrivo,3),Nodi(nodo_arrivo,2),'g+')

% end %
%% TRACKING

% Consideriamo di seguire l'utente dalla posizione di partenza a quella
% finale, facendo l'ipotesi che vengano percorsi 2 nodi/secondo, quindi
% nella nostra stima faremo un controllo ogni due nodi del path.

limite = length(path);
count = 3;
i_temp = 1;
while count<=limite
    posizione_ideale_x = Nodi(path(count),3);
    posizione_ideale_y = Nodi(path(count),2);
    % Distanze della posizione intermedia dagli Access Point
    dist_posizioni_relative = zeros(1,n_ap); % Vettore delle distanze
    %Calcolo MURI ATTRAVERSATI UTENTE
    Los_d_utente_relativo_temp = find((Nodi(:,2)==posizione_ideale_y) & ...
        (Nodi(:,3)==posizione_ideale_x));

```

```

if (w_x ~= 1) || (w_y ~= 1)
    load('Los_d_1_1.mat');
    Los_d_utente_relativo = Los_d_1_1(:,Los_d_utente_relativo_temp);
end
if (w_x == 1) && (w_y == 1)
Los_d_utente_relativo = Los_d(:,Los_d_utente_relativo_temp);
end
mnw_utente_relativo_temp = find((Nodi(:,2)==Nodi(path(count),2)) & ...
    (Nodi(:,3)==Nodi(path(count),3)));
mnw_utente_relativo = mat_num_wall_1_1(:,mnw_utente_relativo_temp);

%% MATRICE OMEGA_intermedio%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta_shadowing_omega_relativo = delta_min + ...
    (delta_max-delta_min).*rand(utenti,n_ap);
% L_d_omega_relativo = Distanza in potenza (dB) tra l'i-esimo AP e il
% j-esimo passaggiointermedio dell'utente
L_d_omega_relativo = Los_d_utente_relativo + ...
    lc + (mnw_utente_relativo*l2);
L_d_omega_relativo = L_d_omega_relativo';
omega_relativo = L_d_omega_relativo + delta_shadowing_omega_relativo;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Calcolo le distanze euclidee
D_intermedio = Min_Dist_Euclidea_mod(1, ...
    length(fingerprint_x), S, omega_relativo);
% D_intermedio = D_intermedio';

% Appliciamo l'algorithmo KNN: K Nearest Neighbour

vettore_neighbour_intermedio = K_NN_multi_wall_temp(D_intermedio, K);

for n = 1:length(vettore_neighbour_intermedio)
    k_intermedio_x(n) = fingerprint_x(vettore_neighbour_intermedio(n));
    k_intermedio_y(n) = fingerprint_y(vettore_neighbour_intermedio(n));
end

k_intermedio_x;
k_intermedio_y;
stima_pos_x_intermedio = (1/length(vettore_neighbour_intermedio)) ...
    *(sum(k_intermedio_y));
stima_pos_y_intermedio = (1/length(vettore_neighbour_intermedio)) ...
    *(sum(k_intermedio_x));

errore_pos_intermedia(i_temp) = calcoladistanza(posizione_ideale_x, ...
    posizione_ideale_y, stima_pos_y_intermedio(i), ...
    stima_pos_x_intermedio(i));
i_temp = i_temp + 1;

k_intermedio_x=0;
k_intermedio_y=0;

plot(stima_pos_y_intermedio,stima_pos_x_intermedio,'m+', 'linewidth',9);
plot(posizione_ideale_x,posizione_ideale_y,'k+');

pause
count = count+2;

end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FINE CODICE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

RIPORTIAMO LE FUNZIONI PIU' IMPORTANTI CHE ABBIAMO UTILIZZATO ALL'INTERNO DEL CODICE:

```
1) function [neighbour] = K_NN_multi_wall_temp(dist_euclidean, K)
```

```
% Funzione che prende in input il vettore D delle distanze euclidee  
% e un intero K che rappresenta il numero di neighbours da considerare,  
% e che restituisce in output un vettore di lunghezza pari a K con le  
% posizioni dei nodi più vicini al punto di interesse.
```

```
neighbour = zeros(1,K);  
M = max(dist_euclidean);  
for j = 1:K  
  
    pos_minimo_temp = find(dist_euclidean == min(dist_euclidean));  
    if length(pos_minimo_temp) == 1;  
        neighbour(j) = pos_minimo_temp;  
    end  
    if length(pos_minimo_temp) ~=1;  
        neighbour(j) = pos_minimo_temp(1);  
    end  
    dist_euclidean(pos_minimo_temp) = M;  
  
end
```

```
2) function [dist,path] = dijkstra(nodes,segments,start_id,finish_id)
```

```
% Algoritmo che abbiamo utilizzato per il calcolo dello shortest path.
```

```
%DIJKSTRA Calculates the shortest distance and path between points on a map  
% using Dijkstra's Shortest Path Algorithm  
%  
% [DIST, PATH] = DIJKSTRA(NODES, SEGMENTS, SID, FID)  
% Calculates the shortest distance and path between start and finish nodes  
SID and FID  
%  
% [DIST, PATH] = DIJKSTRA(NODES, SEGMENTS, SID)  
% Calculates the shortest distances and paths from the starting node SID to  
all  
% other nodes in the map  
%  
% Note:  
% DIJKSTRA is set up so that an example is created if no inputs are  
provided,  
% but ignores the example and just processes the inputs if they are  
given.  
%  
% Inputs:  
% NODES should be an Nx3 or Nx4 matrix with the format [ID X Y] or [ID X Y  
Z]  
% where ID is an integer, and X, Y, Z are cartesian position  
coordinates)  
% SEGMENTS should be an Mx3 matrix with the format [ID N1 N2]  
% where ID is an integer, and N1, N2 correspond to node IDs from NODES  
list  
% such that there is an [undirected] edge/segment between node N1 and  
node N2  
% SID should be an integer in the node ID list corresponding with the  
starting node  
% FID (optional) should be an integer in the node ID list corresponding  
with the finish  
%
```

```

% Outputs:
%   DIST is the shortest Euclidean distance
%   If FID was specified, DIST will be a 1x1 double representing the
shortest
%   Euclidean distance between SID and FID along the map segments. DIST
will have
%   a value of INF if there are no segments connecting SID and FID.
%   If FID was not specified, DIST will be a 1xN vector representing the
shortest
%   Euclidean distance between SID and all other nodes on the map. DIST
will have
%   a value of INF for any nodes that cannot be reached along segments
of the map.
%   PATH is a list of nodes containing the shortest route
%   If FID was specified, PATH will be a 1xP vector of node IDs from SID
to FID.
%   NAN will be returned if there are no segments connecting SID to FID.
%   If FID was not specified, PATH will be a 1xN cell of vectors
representing the
%   shortest route from SID to all other nodes on the map. PATH will
have a value
%   of NAN for any nodes that cannot be reached along the segments of
the map.
%
% Example:
%   dijkstra; % calculates shortest path and distance between two nodes
%             % on a map of randomly generated nodes and segments
%
% Example:
%   nodes = [(1:10); 100*rand(2,10)]';
%   segments = [(1:17); floor(1:0.5:9); ceil(2:0.5:10)]';
%   figure; plot(nodes(:,2), nodes(:,3),'k.');
```

```

%   hold on;
%   for s = 1:17
%       if (s <= 10) text(nodes(s,2),nodes(s,3),[' ' num2str(s)]); end
%       plot(nodes(segments(s,2:3)',2),nodes(segments(s,2:3)',3),'k');
```

```

%   end
%   [d, p] = dijkstra(nodes, segments, 1, 10)
%   for n = 2:length(p)
%       plot(nodes(p(n-1:n),2),nodes(p(n-1:n),3),'r-.','linewidth',2);
%   end
%   hold off;
%
% Author: Joseph Kirk
% Email: jdkirk630 at gmail dot com
% Release: 1.3
% Release Date: 5/18/07

if (nargin < 3) % SETUP
    % (GENERATE RANDOM EXAMPLE OF NODES AND SEGMENTS IF NOT GIVEN AS INPUTS)
    % Create a random set of nodes/vertices, and connect some of them with
    % edges/segments. Then graph the resulting map.
    num_nodes = 40; L = 100; max_seg_length = 30; ids = (1:num_nodes)';
    nodes = [ids L*rand(num_nodes,2)]; % create random nodes
    h = figure; plot(nodes(:,2),nodes(:,3),'k.') % plot the nodes
    text(nodes(num_nodes,2),nodes(num_nodes,3),...
        [' ' num2str(ids(num_nodes))],'Color','b','FontWeight','b')
    hold on
    num_segs = 0; segments = zeros(num_nodes*(num_nodes-1)/2,3);
    for i = 1:num_nodes-1 % create edges between some of the nodes
        text(nodes(i,2),nodes(i,3),[' '
num2str(ids(i))],'Color','b','FontWeight','b')
        for j = i+1:num_nodes
            d = sqrt(sum((nodes(i,2:3) - nodes(j,2:3)).^2));
            if and(d < max_seg_length,rand < 0.6)
                plot([nodes(i,2) nodes(j,2)],[nodes(i,3) nodes(j,3)],'k.-')
```

```

        % add this link to the segments list
        num_segs = num_segs + 1;
        segments(num_segs,:) = [num_segs nodes(i,1) nodes(j,1)];
    end
end
end
segments(num_segs+1:num_nodes*(num_nodes-1)/2,:) = [];
axis([0 L 0 L])
% Calculate Shortest Path Using Dijkstra's Algorithm
% Get random starting/ending nodes, compute the shortest distance and path.
start_id = ceil(num_nodes*rand); disp(['start id = ' num2str(start_id)]);
finish_id = ceil(num_nodes*rand); disp(['finish id = '
num2str(finish_id)]);
[distance,path] = dijkstra(nodes,segments,start_id,finish_id);
disp(['distance = ' num2str(distance)]); disp(['path = [' num2str(path)
']']);
% If a Shortest Path exists, Plot it on the Map.
figure(h)
for k = 2:length(path)
    m = find(nodes(:,1) == path(k-1));
    n = find(nodes(:,1) == path(k));
    plot([nodes(m,2) nodes(n,2)], [nodes(m,3)
nodes(n,3)], 'ro-', 'LineWidth', 2);
end
title(['Shortest Distance from ' num2str(start_id) ' to ' ...
num2str(finish_id) ' = ' num2str(distance)])
hold off

else
%-----
% MAIN FUNCTION - DIJKSTRA'S ALGORITHM

% initializations
node_ids = nodes(:,1);
[num_map_pts,cols] = size(nodes);
table = sparse(num_map_pts,2);
shortest_distance = Inf(num_map_pts,1);
settled = zeros(num_map_pts,1);
path = num2cell(NaN(num_map_pts,1));
col = 2;
pidx = find(start_id == node_ids);
shortest_distance(pidx) = 0;
table(pidx,col) = 0;
settled(pidx) = 1;
path(pidx) = {start_id};
if (nargin < 4) % compute shortest path for all nodes
    while_cmd = 'sum(~settled) > 0';
else % terminate algorithm early
    while_cmd = 'settled(zz) == 0';
    zz = find(finish_id == node_ids);
end
while eval(while_cmd)
    % update the table
    table(:,col-1) = table(:,col);
    table(pidx,col) = 0;
    % find neighboring nodes in the segments list
    neighbor_ids = [segments(node_ids(pidx) == segments(:,2),3);
segments(node_ids(pidx) == segments(:,3),2)];
    % calculate the distances to the neighboring nodes and keep track of
the paths
    for k = 1:length(neighbor_ids)
        cidx = find(neighbor_ids(k) == node_ids);
        if ~settled(cidx)
            d = sqrt(sum((nodes(pidx,2:cols) - nodes(cidx,2:cols)).^2));
            if (table(cidx,col-1) == 0) || ...
                (table(cidx,col-1) > (table(pidx,col-1) + d))

```

```

        table(cidx,col) = table(pidx,col-1) + d;
        tmp_path = path(pidx);
        path(cidx) = {[tmp_path{1} neighbor_ids(k)]};
    else
        table(cidx,col) = table(cidx,col-1);
    end
end
end
% find the minimum non-zero value in the table and save it
nidx = find(table(:,col));
ndx = find(table(nidx,col) == min(table(nidx,col)));
if isempty(ndx)
    break
else
    pidx = nidx(ndx(1));
    shortest_distance(pidx) = table(pidx,col);
    settled(pidx) = 1;
end
end
if (nargin < 4) % return the distance and path arrays for all of the nodes
    dist = shortest_distance';
    path = path';
else % return the distance and path for the ending node
    dist = shortest_distance(zz);
    path = path(zz);
    path = path{1};
end
end
end

```