

Localizzazione tramite una rete di sensori *wireless* utilizzando tecniche non-parametriche.

Damiano Picco, Marco Prendin, Damiano Vanti

Abstract — Questo documento è la relazione finale del progetto svolto per il corso di Progettazione di Sistemi di Controllo tenuto dal prof. L. Schenato dell'Università di Padova.

In questo progetto viene sviluppato il problema della localizzazione di un nodo mobile tramite una rete di sensori wireless in modo distribuito e con tecniche non parametriche. Un approccio Bayesiano e in particolare l'utilizzo della teoria sui processi Gaussiani ha portato allo sviluppo di un algoritmo in grado di stimare la posizione di un nodo mobile grazie alla conoscenza della potenza del segnale radio ricevuto dallo stesso da parte di nodi fissi preposti nei due ambienti che sono stati analizzati: i laboratori dell'Istituto Nazionale di Fisica Nucleare (INFN) di Legnaro e i laboratori della sede di Ingegneria dell'Informazione (DEI) di Padova. L'algoritmo è stato poi riformulato sia in forma centralizzata che in forma decentralizzata, andando anche a sviluppare tecniche multi-hop.

Index Terms — Localizzazione, Bayesiano, Processo Gaussiano, inferenza, covarianza, WSN, multi-hop, connettività, centralizzato, decentralizzato.

I. INTRODUZIONE

L'obiettivo del progetto è la determinazione della posizione, all'interno di una rete di sensori wireless (Wireless Sensor Network, WSN), di un nodo mobile tramite la conoscenza della misura della potenza (Received Signal Strength, RSS) del segnale emesso da sensori fissi preposti nell'ambiente (nodi ancora) e grazie anche alla conoscenza

di un training set di misure di potenza-posizioni effettuate in precedenza.

Dal punto di vista teorico l'obiettivo è quello di estrapolare un modello continuo da un set di dati che sia un sottoinsieme dei dati sul quale vogliamo applicare il modello; si tratta perciò di un procedimento di tipo induttivo, in quanto si cerca di ricostruire un modello generale basandosi su un insieme limitato di dati sperimentali. Come sarà approfondito in dettaglio nel seguito, in questo progetto si farà uso della teoria sui processi Gaussiani, in quanto i calcoli richiesti per l'inferenza statistica diventano particolarmente semplici.

Questo progetto è nato dall'idea di migliorare i risultati ottenuti in precedenti lavori, in particolare con riferimento al progetto sviluppato nei laboratori dell'Istituto Nazionale di Fisica Nucleare (INFN) di Legnaro [3] nato dalla necessità di instaurare all'interno dei laboratori una rete che permettesse, in caso di emergenza, la localizzazione ed il tracking in tempo reale degli operatori di soccorso presenti nei locali colpiti, in modo da poterne guidare le mosse dall'esterno e facilitarli nell'intervento.

Il progetto è stato in seguito esteso ai laboratori al piano terra della sede di Ingegneria dell'informazione dell'Università di Padova (DEI), in particolare si è cercato di sfruttare tutta la rete già cablata presente nei laboratori nonostante fossero stati già intrapresi studi sulla connettività e sul numero minimo di sensori necessari per una buona prestazione (in termini di connettività) di tale rete [4]; studi che potranno essere ripresi in uno sviluppo futuro per la realizzazione di una rete efficiente e a minor costo.

Lo stato dell'arte nella localizzazione e tracking riporta, come sarà descritto in maggior dettaglio nella prossima sezione, svariati metodi e strategie per la soluzione di tale problema. I metodi utilizzati lungo tutto il progetto fanno uso, come già introdotto, di una mappa di potenza RSSI del segnale emesso e ricevuto rispettivamente dai nodi ancora e dal nodo mobile, diversificandosi a seconda della quantità di informazione utilizzata nella stima della posizione del nodo mobile da parte dei vari nodi ancora (saranno sviluppati algoritmi sia centralizzati che decentralizzati e in quest'ultimi si illustreranno anche tecniche multi-hop) e di come il nodo mobile "richiede" la propria posizione.

Questo documento è organizzato nel modo seguente. Nella Sezione II è contenuta una rassegna di alcuni metodi di localizzazione e *tracking* basati su reti wireless. La Sezione III è lasciata ai posteri.

II .STATO DELL'ARTE

Qualsiasi sistema di navigazione e di tracking ha come presupposto principale la localizzazione dell'oggetto di interesse; è bene ricordare che l'ambito al quale ci riferiremo d'ora in poi sono le WSN (*Wireless Sensor Network*) cioè reti senza fili formate da un insieme molto numeroso di dispositivi autonomi, detti nodi o mote, in grado di comunicare via radio tra loro e spesso dotati di sensori (come ad esempio per la temperatura e umidità) e in grado di instaurare delle connessioni ben distribuite e capillari con l'obiettivo di monitorare attentamente un'area territoriale.

Le WSN sono caratterizzate da un imponente numero di sensori, con una direzione del traffico dati non precisa; normalmente ogni nodo spedisce pacchetti a tutti i dispositivi in ascolto, e non ad alcuni soltanto.

Notevole importanza riveste l'aspetto energetico: utilizzandoli come sensori e posizionandoli anche in luoghi non facilmente o frequentemente raggiungibili, vengono spesso alimentati a batterie e diventa quindi

necessario assicurare loro un'autonomia discreta in base al servizio richiesto, per evitare onerosi tempi di sostituzione (nel caso il numero di sensori sia elevato) o lunghi spostamenti per raggiungerli tutti.

In sintesi possiamo elencare le caratteristiche fondamentali che distinguono una rete WSN:

- nodi di dimensioni estremamente contenute e dalle risorse di memoria e calcolo limitate;
- interazione con l'ambiente circostante tramite sensori;
- scambio di dati mediante rete wireless;
- lunga autonomia energetica
- contesti operativi eterogenei ed ostili, anche inaccessibili all'uomo;
- topologia dinamica di rete;
- funzionamento autonomo, non costantemente monitorato, e conseguente politica di gestione dei guasti.

Ritornando al nostro problema, in letteratura le tecniche di localizzazione sono classificate secondo svariati criteri, come ad esempio in base alla grandezza fisica utilizzata: onde radio, impulsi sonori, infrarossi; o in base al parametro che fornisce le informazioni tra i nodi, in particolare si distinguono:

- *Angle of Arrival AoA*: basato sulla stima dell'angolo relativo tra due nodi;
- *Time of Arrival ToA*: basato sul tempo impiegato dal segnale radio per propagarsi da un nodo ad un altro;
- *Time Difference of Arrival TDoA*: basato sull'intervallo di tempo tra l'arrivo di un segnale radio e un segnale ad ultrasuoni emesso da un nodo;
- *Received Signal Strength Indicator RSSI*: basato sulla potenza ricevuta-inviata dai diversi nodi.

Le prime tre tecniche presentate richiedono dispositivi specifici per la loro implementazione: ad esempio dispositivi ad ultrasuoni, dispositivi di sincronizzazione, telecamere ecc.

La loro realizzazione ha portato interessanti risultati, si ricordi come esempio il progetto Cricket [], dove per localizzare un nodo mobile il sistema misura la differenza fra il tempo di volo del segnale elettromagnetico ed il segnale ultrasonico prodotto dal nodo mobile.

La precisione ottenuta con questa tecnica di localizzazione è ragguardevole, in quanto compresa nell'ordine del centimetro, ma si è verificato che i componenti preposti alla generazione degli impulsi ultrasonici sono molto sensibili alle vibrazioni meccaniche e che quindi, l'inserimento di tali attuatori nei nodi mobili soggetti a spostamenti e a continue vibrazioni meccaniche, causa la rottura degli stessi con la conseguente inutilizzabilità del nodo mobile.

Questo e altri motivi legati al costo hanno provveduto ad assicurare ai metodi basati sulla RSSI una maggiore popolarità, malgrado i risultati ottenuti con tali dispositivi non siano altrettanto allettanti di quelli ottenuti con le già citate tecniche.

Le tecniche di localizzazione basate sull'RSSI si distinguono in due classi principali:

- modello di Canale: l'algoritmo stima la distanza relativa del nodo mobile dai nodi ancora e con tecniche di triangolazione individua la posizione dello stesso.
- mappa di Potenza RSSI: ogni nodo ancora crea una mappa di potenza che può essere ottenuta attraverso un modello analitico sulla propagazione del segnale radio oppure da una serie di misure di potenza raccolte in diverse posizioni (learning measure) così da permettere un confronto con la misura test. I principali sistemi sviluppati in questa categoria sono RADAR [] e MoteTrack [] che andiamo a descriveremo brevemente in seguito.

Il progetto RADAR [] opera in due fasi:

1. *fase off-line*, in cui si effettua una raccolta dati che descriva la distribuzione nell'ambiente della potenza del segnale emesso dai nodi ancora (*Received Signal Strength, RSS*);
2. *fase on-line*, in cui le misure rilevate vengono confrontate con i dati a priori ed elaborate per di determinare la posizione del nodo mobile.

La prima fase viene attuata in due modi: effettuando una serie di misurazioni sul campo oppure basandosi su un modello di propagazione del segnale radio. Nel secondo caso tipicamente il modello è il seguente:

$$P(d) = P(d_0) - 10n_p \log\left(\frac{d}{d_0}\right) - C \cdot WAF$$

dove nella formula $P(d_0)$ rappresenta il valore della potenza del segnale elettromagnetico che il nodo rileva ad una distanza d_0 dal nodo ancora che genera il segnale elettromagnetico. Tale modello prevede inoltre la stima sul campo di due parametri: n_p e WAF .

Il primo rappresenta il fattore secondo cui decresce la potenza del segnale elettromagnetico all'aumentare della distanza, mentre il secondo tiene conto della attenuazione di potenza del campo elettromagnetico causata dal passaggio delle onde attraverso i muri (*Wall Attenuation Factor*).

Anche in presenza di ambienti complessi in cui vi siano molte interferenze agenti sulla propagazione del segnale radio, il sistema RADAR permette di avere un notevole grado di accuratezza (lo scarto medio si aggira sui 2/3 metri).

Il progetto MoteTrack è un sistema di localizzazione decentralizzato, basato sulla misura della potenza del segnale elettromagnetico prodotto da alcuni nodi ancora. Per determinare la posizione di un nodo mobile il sistema utilizza una mappa, creata attraverso la raccolta di misure svolte a priori sul campo, che associa ogni punto del piano al valore di potenza del campo

elettromagnetico che un nodo mobile rilevarebbe in quella coordinata. MoteTrack non utilizza quindi alcun tipo di modello che legghi la potenza del segnale ricevuto(RSS, Received Signal Strength) ad un qualche valore di distanza.

L'insieme dei valori di RSS rilevati da un nodo mobile, unitamente all'identificativo (ID) del nodo ancora che ha generato il campo elettromagnetico a cui si riferiscono i dati, è detta signature. L'insieme di una signature e delle coordinate spaziali in cui essa è stata raccolta è detto invece reference signature. L'insieme delle reference signature raccolte sono memorizzate fra i nodi della rete in modo da minimizzare l'occupazione di memoria per il singolo nodo e contemporaneamente massimizzare la ridondanza dei dati.

Durante la fase di localizzazione, un nodo mobile M, dopo aver raccolto una propria signature, utilizza le reference signature memorizzate nei nodi ancora da esso rilevabili per stimare la propria posizione. La posizione del nodo è dunque determinata attraverso una media pesata delle distanze dalle coordinate contenute nelle reference signature utilizzate e, i pesi utilizzati nella media, sono proporzionali alle distanze fra la signature rilevata dal nodo M e quelle contenute nelle reference signature.

In figura è riportato un esempio di raccolta di una signature da parte di un nodo mobile M e l'invio di alcune reference signature da parte dei nodi ancora.

Sia RADAR che MoteTrack sono basati su un modello deterministico della mappa di potenza; nel progetto che andremo a descrivere in dettaglio nella prossima sezione includeremo anche dell'informazione relativa alla varianza delle misure rilevate, in questo modo la posizione del nodo mobile è ottenuta calcolando la maggiore probabilità di una posizione data la potenza ricevuta, quindi sfruttando un approccio di tipo Bayesiano.

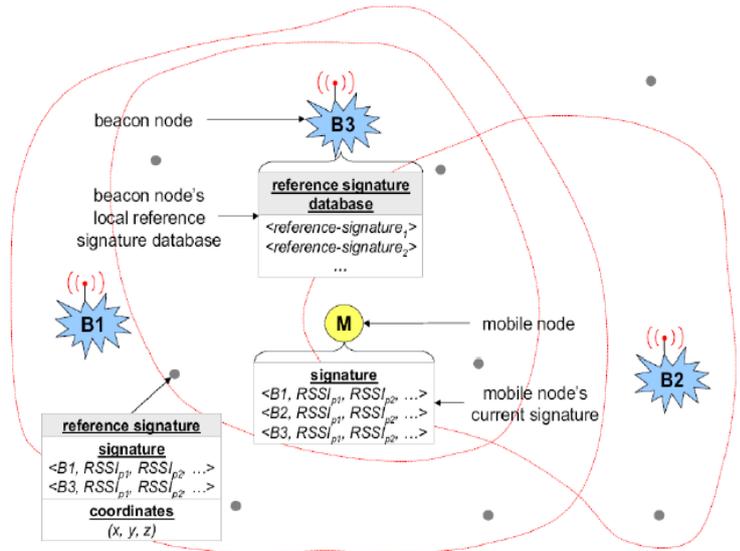


Figura 1.1: raccolta di una signature da parte di un nodo mobile M e l'invio di alcune reference signature da parte dei nodi ancora

III. REGRESSIONE E PROCESSI GAUSSIANI

• INTRODUZIONE

Dal punto di vista teorico, il problema affrontato in questo progetto è l'apprendimento di una mappa di ingresso-uscita da un set di dati empirici (training dataset). In letteratura il problema è stato largamente studiato, prendendo il nome di *regressione* nel caso in cui l'uscita sia continua, e *classificazione* in caso contrario.

Trattando con la regressione, il nostro obiettivo sarà quindi quello di estrapolare un modello continuo che legghi la potenza rilevata dal nodo mobile dai vari sensori ancora (input, x) alla posizione corrente dello stesso (uscita, y) in un sistema di riferimento noto. Chiamando D il numero di nodi ancora preposti nell'ambiente, l'obiettivo sarà quindi quello di ricavare una funzione $f(x)$ tale per cui

$$f(x) : \mathbb{R}^D \rightarrow \mathbb{R}^2$$

Per fare questo, ricavato sperimentalmente un set di misure ingresso-uscita, divideremo tale set in due insiemi: *training set* grazie al quale estrapoleremo la funzione $f(x)$ e *test set*

grazie al quale testeremo l'efficacia della funzione trovata.

L'idea fondamentale, che svilupperemo inizialmente grazie all'inferenza bayesiana e che poi estenderemo grazie ai processi gaussiani, sta nell'assegnare ad ogni possibile funzione, che dato come ingresso la potenza rilevata dal nodo mobile dai vari sensori ancora dia in uscita la posizione corrente dello stesso, una probabilità a priori sulla base di assunzioni che verranno fatte sulle funzioni in gioco (ad esempio la regolarità) andando poi a modificare tale probabilità grazie al training set acquisito sperimentalmente. Appare evidente il problema della presenza di un insieme infinito di funzioni plausibili, problema che osserveremo in un primo momento parlando di inferenza bayesiana e che risolveremo in un secondo momento estendendo il nostro problema ai processi gaussiani.

Ricordando che un processo gaussiano non è altro che la generalizzazione della densità di probabilità Gaussiana, dove si trattano in questo caso non più scalari o vettori bensì funzioni, non è difficile a questo punto dare un'idea grafica della situazione che andremo a trattare; si pensi per semplicità ad un problema di regressione scalare, dove l'ingresso x produce l'uscita $f(x)$. In figura 1.1(a) sono rappresentate un numero finito di funzioni estratte casualmente dalla distribuzione a priori specificata da un particolare processo Gaussiano (si osservi come tale processo, attraverso la sua *covarianza*, favorisca funzioni regolari). La probabilità a priori rappresenta la nostra idea a priori sul tipo di funzioni che ci aspettiamo di osservare, assumendo che la varianza di tale probabilità non dipenda da x è evidente che la regione di incertezza (rappresentata dalla parte ombreggiata) è quella riportata in figura 1.2(a). Le cose cambiano se sono disponibili osservazioni di coppie di ingresso-uscita, ad esempio siano note. Ora vogliamo considerare solo funzioni che passano esattamente per i due punti specificati; tali punti insieme alla probabilità a priori danno luogo a quella che viene chiamata probabilità a posteriori, che porta i risultati visibili in figura 1.2(b).

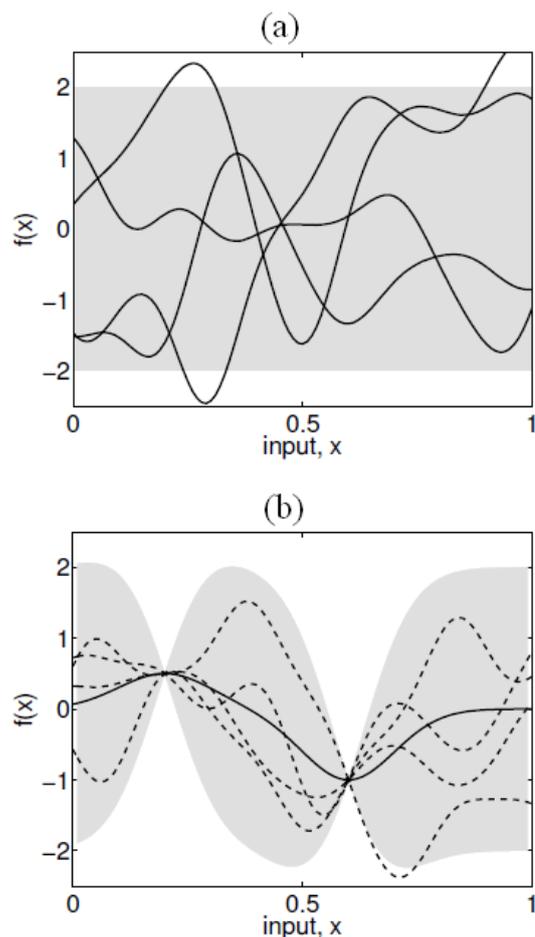


Figura 1.2: Il riquadro (a) mostra quattro funzioni campione estratte dalla probabilità a priori, il riquadro (b) mostra la situazione dopo l'osservazione di due datapoint.

Come studieremo approfonditamente nella parte dedicata ai processi Gaussiani, la specificazione della probabilità a priori è importante perché fissa le proprietà delle funzioni che considereremo per l'inferenza. Sopra abbiamo sfiorato l'argomento parlando della varianza, che comunque non è l'unica caratteristica che può essere specificata e manipolata. Molto importante è osservare che la regolarità e la stazionarietà delle funzioni in figura 1.1(a) è indotta dalla funzione *covarianza* del processo gaussiano generato; come si vedrà in seguito il problema di *learning* nei processi gaussiani si traduce esattamente nel problema di trovare proprietà adatte per le funzioni covarianza.

• **REGRESSIONE**

Supponiamo sia disponibile un training set \mathcal{D} di n osservazioni $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ dove \mathbf{x}_i è un vettore colonna di dimensione D rappresentante l'ingresso e y_i l'uscita. L'insieme X è formato dall'aggregazione degli n vettori \mathbf{x}_i , dunque è rappresentabile con una matrice $D \times n$, mentre gli output possono essere rappresentati dal vettore \mathbf{y} , così da poter scrivere $\mathcal{D} = (X, \mathbf{y})$.

Andiamo quindi ad analizzare il modello standard di regressione lineare attraverso un'analisi bayesiana:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}, \quad y = f(\mathbf{x}) + \varepsilon$$

dove \mathbf{x} è il vettore degli ingressi, \mathbf{w} è un vettore di pesi (parametri) del modello lineare, f è il valore della funzione e y è il valore osservato. Nel modello abbiamo assunto che il valore osservato y differisce dal valore della funzione $f(\mathbf{x})$ a causa di rumore additivo, che per tutta la trattazione assumeremo gaussiano a variabili indipendenti identicamente distribuite (i.i.d.) di media zero e varianza σ_n^2

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$$

L'assunzione fatta sul rumore insieme al modello utilizzato per la regressione danno origine a quella che in letteratura viene chiamata *verosimiglianza*, cioè la densità di probabilità delle osservazioni assumendo noti i parametri

$$\begin{aligned} p(\mathbf{y}|X, \mathbf{w}) &= \prod_{i=1}^n p(y_i|\mathbf{x}_i, \mathbf{w}) = \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma_n^2}\right) = \\ &= \frac{1}{(2\pi\sigma_n^2)^n} \exp\left(-\frac{1}{2\sigma_n^2} |\mathbf{y} - X^T \mathbf{w}|^2\right) = \\ &= \mathcal{N}(X^T \mathbf{w}, \sigma_n^2 I), \end{aligned}$$

dove $|\mathbf{z}|$ la norma euclidiana del vettore \mathbf{z} .

Nel formalismo Bayesiano siamo obbligati a specificare la probabilità a priori dei parametri, per il momento poniamo tale probabilità Gaussiana, con media nulla e matrice di covarianza Σ_p : $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$. Il ruolo e le proprietà di tale probabilità saranno discusse in dettaglio quando descriveremo i processi Gaussiani.

A questo punto, utilizzando la regola di Bayes (vedi A.3) possiamo calcolare la distribuzione a posteriori:

$$posteriori = \frac{priori \times verosimiglianza}{verosimiglianza marginalizzata}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$$

dove la costante di normalizzazione, nota come *verosimiglianza marginalizzata*, è indipendente dai pesi ed è data da:

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})d\mathbf{w}.$$

La probabilità a posteriori $p(\mathbf{w}|\mathbf{y}, X)$ unisce la verosimiglianza e la probabilità a priori, fornendo tutta la conoscenza che possiamo ricavare sui parametri. Tralasciando la costante di normalizzazione possiamo scrivere:

$$\begin{aligned} p(\mathbf{w}|X, \mathbf{y}) &\propto \\ &\propto \exp\left(-\frac{1}{2\sigma_n^2} (\mathbf{y} - X^T \mathbf{w})^T (\mathbf{y} - X^T \mathbf{w})\right) \exp\left(-\frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w}\right) \\ &\propto \exp\left(-\frac{1}{2} (\mathbf{w} - \bar{\mathbf{w}})^T \left(\frac{1}{2\sigma_n^2} X X^T + \Sigma_p^{-1}\right) (\mathbf{w} - \bar{\mathbf{w}})\right) \end{aligned}$$

$$\text{Dove } \bar{\mathbf{w}} = \sigma_n^{-2} (\sigma_n^{-2} X X^T + \Sigma_p^{-1})^{-1} X \mathbf{y}.$$

Si riconosce facilmente che la forma della distribuzione a posteriori è una Gaussiana con media $\bar{\mathbf{w}}$ e matrice di covarianza A^{-1}

$$p(\mathbf{w}|X, \mathbf{y}) = \mathcal{N}(\bar{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X \mathbf{y}, A^{-1})$$

$$\text{Dove } A = \sigma_n^{-2} X X^T + \Sigma_p^{-1}.$$

Arrivati a questo punto, per fare predizioni su un test case \mathbf{x}_* non ci resta che mediare su tutti i possibili valori dei parametri, pesati dalla loro probabilità a posteriori.

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})d\mathbf{w} = \\ &= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^T A^{-1}X\mathbf{y}, \mathbf{x}_*^T A^{-1}\mathbf{x}_*\right) \end{aligned}$$

Dove $f_* = f(\mathbf{x}_*)$. Si osservi che la distribuzione del predittore è ancora Gaussiana, con media data dalla media della probabilità a posteriori dei parametri moltiplicata dall'ingresso test, la varianza è una forma quadratica dell'ingresso test con la covarianza della probabilità a posteriori, mostrando che l'incertezza del predittore cresce all'aumentare del modulo del test in ingresso, come era facilmente immaginabile dal modello lineare in considerazione.

Il passo successivo di questa trattazione ha il fine di ovviare alla mancanza di flessibilità del modello lineare e renderlo adatto anche a funzioni non lineari. L'idea è quella di proiettare gli ingressi in uno spazio di funzioni per poi applicare il modello lineare in questo spazio piuttosto che sugli ingressi direttamente. Per esempio, un ingresso scalare x può essere proiettato sullo spazio delle potenze di x : $\phi(x) = (1, x, x^2, x^3, \dots)^T$ per implementare una regressione lineare. Fintanto che le proiezioni sono su funzioni indipendenti da \mathbf{w} il modello è ancora lineare nei parametri, e quindi analiticamente trattabile. Come già annunciato nell'introduzione, nasce il problema di come scegliere questa base di funzioni, come vedremo in seguito il formalismo Gaussiano risponderà a questa questione, per il momento assumiamo che tali funzioni siano date.

Usiamo quindi la mappa $\phi(\mathbf{x}): \mathbb{R}^D \rightarrow V$ per proiettare lo spazio degli ingressi di dimensione D in uno spazio di funzioni V di dimensione N , generato da una base di funzioni $\langle \phi_1, \dots, \phi_N \rangle$; sia quindi $\phi(X)$ la matrice che ha per colonne $\phi(\mathbf{x})$, dove \mathbf{x} appartiene al training set.

Il modello diventa:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w},$$

L'analisi di questo modello è identica a quella fatta in precedenza, basta sostituire a X con $\phi(X)$. La distribuzione del predittore diventa $p(f_*|\mathbf{x}_*, X, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^T A^{-1}X\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)\right)$

Dove $\phi = \phi(X)$ e $A = \sigma_n^2 \phi \phi^T + \Sigma_p^{-1}$.

Per fare predizioni usando questa equazione dobbiamo invertire la matrice A che ha dimensione $N \times N$, che può non essere del tutto conveniente se la dimensione dello spazio delle funzioni è grande. Ecco perché andremo a riscrivere l'equazione nel modo seguente:

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &\sim \mathcal{N}(\phi_*^T \Sigma_p \phi (K + \sigma_n^2 I)^{-1} \mathbf{y}, \\ &\phi_*^T \Sigma_p \phi_* - \phi_*^T \Sigma_p \phi (K + \sigma_n^2 I)^{-1} \phi^T \Sigma_p \phi_*) \end{aligned}$$

Dove $\phi_* = \phi(\mathbf{x}_*)$ e $K = \phi^T \Sigma_p \phi$.

Mostreremo ora l'equivalenza delle due equazioni: per quanto riguarda la media, con le definizioni date di A e K , abbiamo che:

$$\sigma_n^{-2} \phi (K + \sigma_n^2 I) = \sigma_n^{-2} \phi (\phi^T \Sigma_p \phi + \sigma_n^2 I) = A \Sigma_p \phi.$$

Ora moltiplicando a sinistra per A^{-1} e a destra per $(K + \sigma_n^2 I)^{-1}$ abbiamo che, come si voleva mostrare $\sigma_n^{-2} A^{-1} \phi = \Sigma_p \phi (K + \sigma_n^2 I)^{-1}$.

Per quanto riguarda la varianza è stato usato il lemma per l'inversione di matrice (cfr. A.9) con $Z^{-1} = \Sigma_p$, $W^{-1} = \sigma_n^2 I$, $V = U = \phi$.

Fondamentale è osservare che nell'equazione per il calcolo di $p(f_*|\mathbf{x}_*, X, \mathbf{y})$ gli ingressi contribuiscono sempre nella forma $\phi^T \Sigma_p \phi$, $\phi_*^T \Sigma_p \phi$ oppure $\phi_*^T \Sigma_p \phi_*$ che possiamo riassumere nella forma più concisa $\phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$ dove \mathbf{x} e \mathbf{x}' sono qui sia il *training* che i *test sets*.

Sia $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$. Per ragioni che saranno chiare nella prossimo paragrafo chiamiamo $k(\cdot, \cdot)$ funzione di covarianza, o come spesso si trova in letteratura *kernel*.

• PROCESSI GAUSSIANI

Una via più generale per ottenere i risultati ottenuti fin'ora è quella di considerare l'inferenza direttamente su uno spazio di funzioni. Infatti nel precedente paragrafo proiettando lo spazio degli ingressi X nello spazio V e prendendo $f(\mathbf{x}) = \phi(\mathbf{x}^T)\mathbf{w}$ abbiamo considerato la funzione $f(\mathbf{x})$ come una combinazione lineare di funzioni $\phi_1(\mathbf{x}), \dots, \phi_N(\mathbf{x})$, combinate secondo il vettore degli scalari \mathbf{w} che rappresenta il "peso" che viene attribuito a ciascuna funzione. Il procedimento visto necessita una base finita, e quindi di operare in uno spazio V di dimensione finita.

Utilizzando i processi gaussiani si può ottenere un modello equivalente al modello lineare visto proiettato questa volta su uno spazio V di dimensione infinita, questo perché il processo Gaussiano definisce una distribuzione di probabilità nello spazio delle funzioni. Tratteremo il problema della trattabilità di tale spazio infinito solo dopo aver dato una definizione formale di processo Gaussiano:

Definizione 1.1 Un processo Gaussiano è una collezione di variabili casuali, le quali hanno distribuzione congiunta Gaussiana.

Un processo gaussiano è completamente specificato dalla sua media e dalla sua funzione covarianza, si definiscono media $m(\mathbf{x})$ e covarianza $k(\mathbf{x}, \mathbf{x}')$ di un processo reale $f(\mathbf{x})$ come:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

Le quali ci permettono di scrivere il processo Gaussiano come

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

Nel nostro caso le variabili casuali del processo non sono altro che i valori della funzione $f(\mathbf{x})$ valutata in \mathbf{x} . In generale comunque la definizione di collezione di variabili casuali per un processo Gaussiano soddisfa alla richiesta di consistenza, infatti se ad esempio il \mathcal{GP} specifica che $(y_1, y_2) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, allora deve anche specificare

$y_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$; in altre parole, l'analisi di un set più grande di dati non cambia la distribuzione dell'insieme più piccolo. Ecco perché con i processi Gaussiani si riesce a trattare anche se idealmente si usa uno spazio infinito di funzioni.

Specificando una media $m(\mathbf{x})$ e una funzione covarianza $k(\mathbf{x}, \mathbf{x}')$ non si fa altro che assegnare una distribuzione di probabilità a priori sullo spazio delle funzioni; tornando ad esempio al nostro modello di regressione lineare $f(\mathbf{x}) = \phi(\mathbf{x}^T)\mathbf{w}$ con $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$ possiamo tradurre queste informazioni in un \mathcal{GP} con

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}] = 0,$$

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] &= \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}\mathbf{w}^T] \phi(\mathbf{x}') \\ &= \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}'). \end{aligned}$$

A questo punto si apre la questione che già era stata sollevata nell'introduzione sulla scelta delle funzioni covarianza, per il momento trascuriamo questo aspetto che riprenderemo più in dettaglio in seguito, andando ora a concentrarci su come incorporare tale conoscenza a priori con l'informazione che ci viene data dal *training set*. Consideriamo subito il caso in cui le misure siano affette da rumore Gaussiano \mathcal{E} con media nulla e varianza σ_n^2 ottenendo quindi $y = f(\mathbf{x}) + \mathcal{E}$. La distribuzione a priori diventa:

$$\text{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}$$

dove δ_{pq} è il delta di Kronecker. Risulta immediato scrivere ora la distribuzione congiunta a priori delle uscite del *training set* \mathbf{y} e delle uscite del *test set* \mathbf{f}_* :

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

Se ci sono n ingressi *training* e n_* ingressi *test* allora $K(X, X_*)$ denota la matrice $n \times n_*$ delle covarianze valutate per ogni coppia di ingressi *training-test*, stesso discorso vale per le altre matrici $K(X, X)$, $K(X_*, X_*)$ e $K(X_*, X)$.

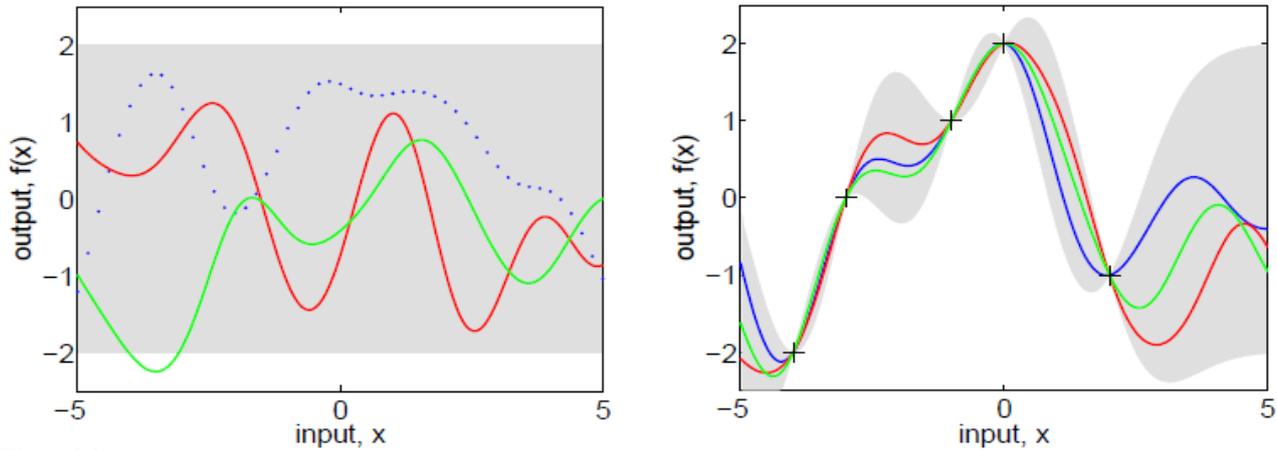


Figura 2.2: Il riquadro a sinistra mostra tre funzioni estratte casualmente dalla distribuzione del \mathcal{GP} a priori. Il riquadro a destra mostra tre funzioni estratte dalla probabilità a posteriori, cioè la probabilità a priori condizionata da cinque misure (in questo caso senza rumore aggiuntivo). In entrambe le figure l'area ombreggiata rappresenta l'area di confidenza del 95% rispettivamente per le due probabilità.

Per estrapolare la distribuzione a posteriori dobbiamo forzare la distribuzione a priori definita sopra a contenere solo quelle funzioni che non sono in contrasto con le osservazioni effettuate. Anche in questo caso una rappresentazione grafica viene utile, infatti osservando la Figura 2.2 a destra si potrebbe pensare di generare funzioni dalla probabilità a priori e poi scartare tutte quelle che non rispettano le osservazioni compiute, chiaramente questa strategia non sarebbe efficiente in senso computazionale. Per fortuna questa operazione in termini probabilistici è molto più semplice, infatti si tratta di *condizionare* la distribuzione di probabilità a priori con le osservazioni fatte, ottenendo (si veda A.2 per ulteriori dettagli):

$$\mathbf{f}_* | X, \mathbf{y}, X_* \sim (\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)), \text{ dove}$$

$$\begin{aligned} \bar{\mathbf{f}}_* &= K(X_*, X) [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y}, \\ \text{cov}(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \end{aligned}$$

Si osservi che la media del predittore è una combinazione lineare delle osservazioni, ecco perché spesso ci si riferisce con il termine *predittore lineare*. Ora assumendo per semplicità di avere un solo test input possiamo riscrivere tale funzione in un modo più standard in letteratura:

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

Dove $\boldsymbol{\alpha} = (K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}$.

Una implementazione pratica della regressione su processi Gaussiani (GPR) è mostrata nell'algoritmo 3.1.

input: X (inputs), \mathbf{y} (targets), k (covariance function), σ_n^2 (noise level), \mathbf{x}_* (test input)

2: $L := \text{cholesky}(K + \sigma_n^2 I)$
 $\boldsymbol{\alpha} := L^\top \setminus (L \setminus \mathbf{y})$

4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$
 $\mathbf{v} := L \setminus \mathbf{k}_*$ } predictive mean eq. (2.25)

6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$ } predictive variance eq. (2.26)

7: **return:** f_* (mean), $\mathbb{V}[f_*]$ (variance).

Algoritmo 3.1: Predizione per la regressione su un processo Gaussiano. L'implementazione richiede l'utilizzo della funzione di Cholesky per l'inversione di matrice. Nel caso di più test input vengono ripetute le righe 4-6. La complessità computazionale è $n^3/6$ per la decomposizione di Cholesky al punto 2 e $n^2/2$ per risolvere il sistema al punto 3.

IV. PARTE SPERIMENTALE

• APPARATO SPERIMENTALE

Nella presente sezione viene esposta una trattazione abbastanza dettagliata della strumentazione a disposizione, così da rendere più comprensibili le motivazioni che hanno portato, come sarà illustrato nella prossima sezione, allo sviluppo di algoritmi decentralizzati e di tecniche *multi-hop*.

Il nucleo della WSN è chiaramente rappresentato dai *nodii wireless* ivi presenti; per quanto riguarda il nostro esperimento faremo riferimento ai *Tmote-Sky*, nodii wireless prodotti della MoteIv corporation (www.moteiv.com); i quali possono essere sintetizzati come dispositivi a bassissima potenza dotati, come ben visibile dalla figura a lato, di diversi sensori ambientali.

Analizziamo in dettaglio le specifiche più interessanti del *Tmote-Sky*:

▪ Microprocessore

Prodotto dalla Texas Instruments, l'MSP430 è caratterizzato da 10kB di RAM, 48kB di flash ROM e un'architettura 16 bit RISC. Ha un oscillatore (DCO) che opera sopra gli 8 MHz e un oscillatore al quarzo utile alla calibrazione del primo. E' dotato ancora di 8 porte ADC interne e altrettante esterne, necessarie per leggere i valori dei sensori o dei livelli di batterie.

▪ Antenna

E' interna, posta parallelamente alla superficie della piastra, ma esiste la possibilità di montarne di esterne tramite il connettore apposito. Ha un range di copertura di 50 metri indoor e oltre 125 metri outdoor ma non è perfettamente omnidirezionale e l'irraggiamento può risentire della presenza o meno delle batterie e persino della posizione del mote (si ha un incremento delle prestazioni se posto in verticale piuttosto che disteso).

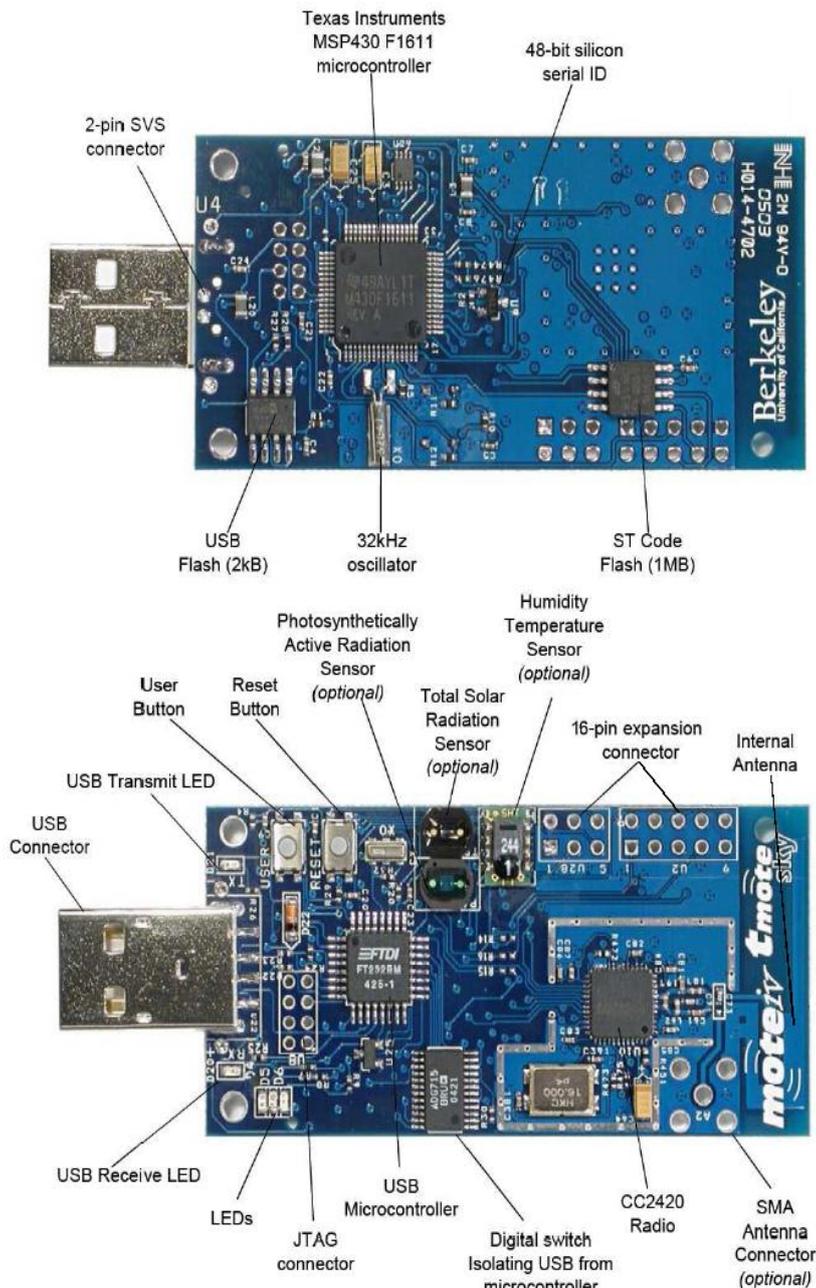


Figura 4.1: Raffigurazione delle due facce del nodo wireless *Tmote-Sky* prodotto dalla MoteIv corporation (www.moteiv.com).

▪ Interfaccia USB

Tmote-Sky utilizza un USB controller per gestire la comunicazione seriale col PC. Una volta inserito sulla relativa borchia verrà visualizzato come dispositivo nelle macchine Linux o OSX, e come COM in Windows; gli verrà inoltre assegnato un indirizzo identificativo univoco. Ad esempio in ambiente Linux col comando "motelist" si verifica quali dispositivi sono collegati al computer.

Non banale risulta la gestione del *Tmote-Sky* se si pensa alle caratteristiche principali che contraddistinguono tale nodo; ecco perché si giustifica lo studio sviluppato per l'implementazione di un sistema operativo di piccole dimensioni, che sia efficiente in termini di risorse impiegate e che occupi poca memoria, infatti, oltre al sistema operativo e al software con cui saranno programmati, verranno temporaneamente salvati in RAM anche i dati ascoltati via radio fino alla loro memorizzazione.

La soluzione attualmente più diffusa è costituita da *TinyOS* [?] [?], un sistema operativo *open source* rilasciato e sviluppato dall'Università di Berkeley (California) e scritto nel linguaggio di programmazione *nesC*. *TinyOS* è stato appositamente realizzato per le reti di sensori, è infatti caratterizzato da una architettura a moduli che sono caricabili singolarmente per permettere l'utilizzo delle sole risorse necessarie per l'applicazione, snellendo la dimensione dell'applicativo da memorizzare in RAM.

Il linguaggio di programmazione *nesC* (*Network embedded system C*) sopra introdotto è di fatto una rielaborazione del *C* basato però sulla guida ad eventi, appositamente pensato per realizzare i concetti e le strutture di esecuzione del *TinyOS*. Le caratteristiche e le strutture fondamentali del *nesC* sono trattate largamente in letteratura e non verranno qui ripetute, si rimanda a [?] per una trattazione esaustiva. Il codice *nesC* utilizzato per la programmazione di tutti i nodi fissi e del nodo mobile non è che un adattamento del codice sorgente utilizzato nel progetto WISE-WAI [?] e disponibile gratuitamente presso il sito <http://cariparo.dei.unipd.it/tools>.

- **APPONTAMENTO DEL PROGETTO**

Il primo obiettivo posto in questo progetto è stato il miglioramento dei risultati ottenuti in [?] presso i laboratori dell'Istituto Nazionale di Fisica Nucleare (INFN) di Legnaro; per questo si sono sfruttati tutti gli studi già fatti sull'analisi dell'ambiente e sulla posizione dei sensori andando a porre l'attenzione esclusivamente sullo studio di un algoritmo di

localizzazione più efficiente di quello presentato in [?].

Un ulteriore sviluppo è stato quello di strutturare l'algoritmo di localizzazione sia in forma centralizzata che in forma decentralizzata, dove nel primo caso si tratta di convogliare tutti i dati della WSN ad un elaboratore esterno, nel nostro caso il nodo mobile stesso, che svolge le operazioni richieste per il calcolo della posizione del nodo mobile; mentre nel secondo caso tutti i nodi fissi (*ancora*) della rete calcolano la posizione del nodo mobile, poi inviata allo stesso con diverse tecniche che tratteremo in dettaglio in seguito.

I due metodi sono stati analizzati andando a confrontare le loro prestazioni sia in funzionamento ideale sia in caso di guasto di uno o più sensori. La tecnica decentralizzata ha visto lo sviluppo di due metodi per la sua realizzazione che sono stati in seguito messi a confronto: *single-hop* e *multi-hop*. Nel primo caso ogni nodo ancora stima la posizione del nodo mobile attraverso l'informazione (il segnale RSSI ricevuto dal nodo) sua e dei suoi vicini più prossimi (con il termine vicini, come si vedrà in seguito, si considera la distanza in metri tra due nodi). Nel secondo caso ogni nodo ancora stima la posizione del nodo mobile attraverso l'informazione sua e dei vicini dei vicini, si veda la Figura 4.1 per una chiarificazione di tale concetto.

Terminata la sperimentazione degli algoritmi sui dati relativi ai laboratori di Legnaro la seconda fase del progetto ha visto la raccolta di nuovi dati presso i laboratori della sede di Ingegneria dell'Informazione (DEI) di Padova. Per quanto riguarda la campagna di raccolta dei dati sono stati utilizzati tutti i sensori disponibili nella zona di interesse anche se in [?] sono stati eseguiti studi sulla connettività che garantivano una buona copertura dei laboratori con un numero minore di sensori. Anche in questa fase del progetto sono stati analizzati e messi a confronto algoritmi di tipo centralizzato e di tipo decentralizzato con le varianti *single-hop* e *multi-hop* già illustrate sopra.

• ANALISI METODI PROPOSTI

Per semplicità di trattazione si considera indipendente la posizione del nodo mobile lungo l'asse x dall'asse y, perciò vengono creati due modelli distinti, uno atto alla stima della posizione del nodo lungo l'asse x, l'altro per l'asse y.

$$y_{x_i} = f_x(x_i) : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$y_{y_i} = f_y(x_i) : \mathbb{R}^D \rightarrow \mathbb{R}$$

$i = 1, \dots, n$, (n è il numero di osservazioni).
 I training set saranno dunque i seguenti:

$$\mathcal{D}_x = (X, \mathbf{y}_x), \mathcal{D}_y = (X, \mathbf{y}_y)$$

dove l'insieme $X \in \mathbb{R}^{D \times n}$ è formato dall'aggregazione degli n vettori $\mathbf{x}_i \in \mathbb{R}^D$ (vettore delle potenze rilevate dal nodo mobile dai vari sensori ancora, di dimensione pari al numero di nodi ancora considerati per il modello), mentre i vettori $\mathbf{y}_x, \mathbf{y}_y \in \mathbb{R}^n$ sono costituiti dalla concatenazione delle posizioni **stimate** (scalari) del sensore mobile lungo l'asse x e y rispettivamente.

Come precedentemente dichiarato, sono stati sviluppati algoritmi in forma centralizzata e decentralizzata. I due metodi sono stati analizzati in funzionamento ideale e in caso di guasto di uno o più sensori. La scelta dei nodi guasti, nella simulazione dei vari metodi, è random, mentre è possibile impostare il numero dei nodi non funzionanti.

▪ Centralizzato

L'approccio centralizzato consiste nel convogliare tutti i dati della WSN ad un elaboratore esterno, nel nostro caso il nodo mobile stesso, che svolge le operazioni per il calcolo della propria posizione.

Tale metodo permette l'impiego di sensori fissi con potenza di calcolo molto ridotta, consentendo quindi l'implementazione di una rete wireless a basso costo. Per alcune applicazioni, comunque, non è disponibile un elaboratore centrale in grado di trattare, in tempo utile, tutta la mole dei dati acquisiti. Si

consideri ora una rete di sensori molto vasta dove tutte le misure devono essere convogliate in un singolo elaboratore, vi sarà perciò un alto consumo energetico presso l'unità centrale [].

Nel caso in esame, la dimensione D del vettore delle potenze rilevate dal nodo mobile è pari al numero totale N di sensori fissi presenti nell'ambiente, precisamente 12 nei laboratori del INFN di Legnaro e 62 nei laboratori della sede di Ingegneria dell'Informazione di Padova.

▪ Decentralizzato

L'architettura decentralizzata qui considerata consiste in un training set distribuito, cioè ad ogni sensore fisso viene associato un training set definito solo dalle potenze relative ai nodi ancora ad esso vicino (e dai vicini dei suoi vicini nel caso *multi hopes*) e non guasto.

Ogni sensore fisso, dotato di risorse hardware con potenze di calcolo sufficientemente elevate, è in grado di stimare la posizione del sensore mobile e di conoscere lo stato di integrità dei suoi vicini.

Successivamente il nodo mobile richiede la propria posizione mediante diverse tecniche discusse di seguito.

È bene definire ora la nozione di vicini:

Due nodi A,B si definiscono nodi vicini se, definito un cerchio di raggio r [m] e centro il nodo A, il nodo B risulta essere all'interno o tutt'al più sulla circonferenza di tale cerchio.

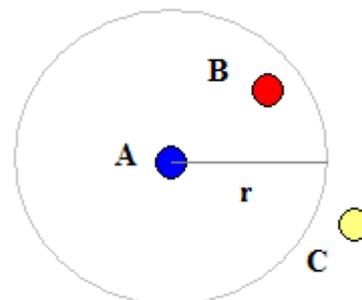


Figura 4.2: Raffigurazione nodi: la coppia A, B sono nodi vicini; A,C nodi non risultano essere nodi vicini.

Ulteriori tecniche di localizzazione distribuita sono discusse in [] dove viene presentato un approccio distribuito di tipo collaborativo.

Le principali motivazioni di un approccio decentralizzato, soprattutto per reti wireless di grandi dimensioni, sono le seguenti: un consumo efficiente dell'energia, distribuzione dei segnali acquisiti a più elaboratori, tempo di calcolo della posizione del nodo mobile ridotto rispetto all'approccio centralizzato in quanto appunto più processori lavorano in parallelo maneggiando una quantità ridotta di dati.

Di contro l'implementazione di tale rete riscontra un costo superiore rispetto ad una architettura centralizzata.

Nell'architettura decentralizzata perciò, saranno costruiti $2N$ modelli distinti:

$$f_{jx}, f_{jy}, j = 1, \dots, N$$

$$\mathcal{D}_{jx} = (X_j, \mathbf{y}_x), \mathcal{D}_{jy} = (X_j, \mathbf{y}_y)$$

dove l'insieme $X_j \in R^{D_j \times n}$ è formato dall'aggregazione degli n vettori $\mathbf{x}_i^{j \in R^{D_j}}$, vettore delle potenze rilevate dal nodo mobile dai sensori ancora j e dai suoi vicini (e dai vicini dei vicini nel caso *multi hopes*) non guasti.

I metodi proposti in questo progetto sono i seguenti:

1) Hope - Max RSSI:

Ciascun nodo ancora stima la posizione del nodo mobile utilizzando la propria informazione (RSSI ricevuto dal nodo mobile) e quella dei suoi vicini (1 hope). Successivamente il nodo mobile seleziona la stima della posizione in cui si trova dal sensore fisso dal quale ha ricevuto il segnale di maggior potenza.

Tale algoritmo è stato implementato nel file MATLAB® punto2_metodo1.m.

2) 1 Hope – Nearest node:

Come nel punto precedente, ad ogni sensore fisso viene associata un training set definito solo dalle potenze relative ai nodi ancora ad esso vicino. Di seguito scelti L nodi ancora in modo casuale, si procede ad una prima stima della posizione del nodo mobile mediante una media pesata delle stime ottenute dai L nodi fissi (il peso risulta proporzionale al RSSI ricevuto dal nodo mobile).

Infine, il sensore mobile seleziona la stima della posizione in cui si trova in un determinato istante dal nodo ancora ad esso più vicino (come posizione del nodo mobile viene utilizzata la stima calcolata al punto precedente).

Tale algoritmo è stato implementato nella funzione MATLAB® punto2_metodo2.m.

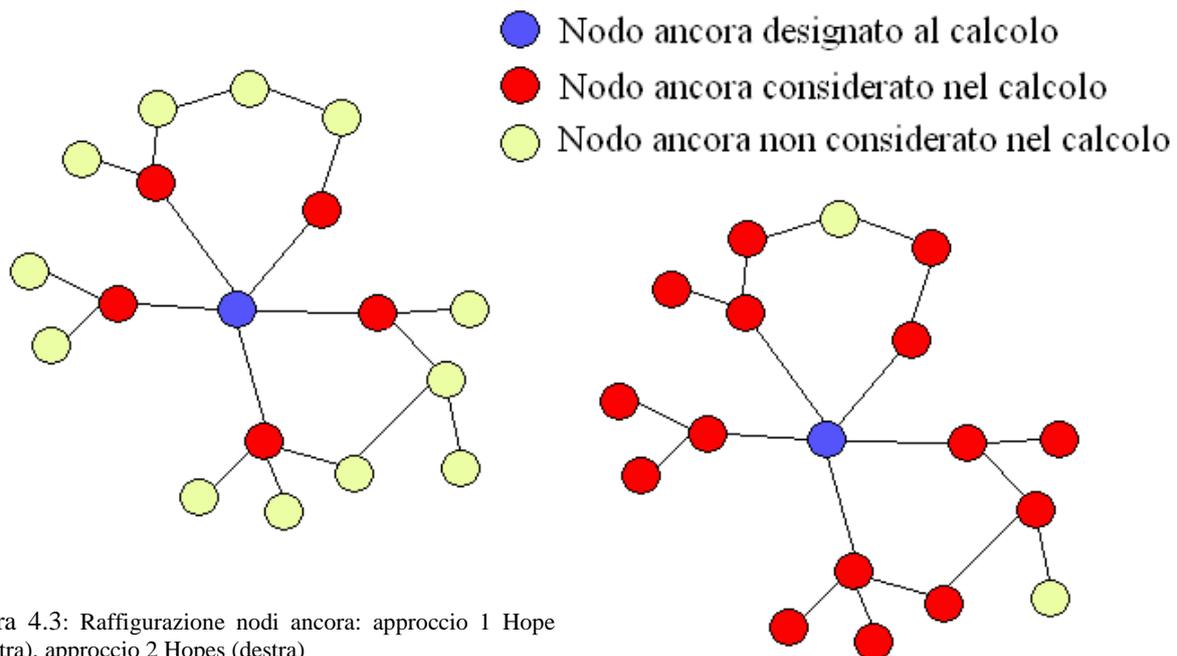


Figura 4.3: Raffigurazione nodi ancora: approccio 1 Hope (sinistra), approccio 2 Hopes (destra)

3) 2 Hops – Max RSSI:

Ogni nodo ancora stima la posizione del nodo mobile utilizzando la propria informazione, quella dei suoi vicini e le potenze relative ai sensori fissi vicini ai suoi vicini (si veda figura 4.2). Successivamente, come per il punto 1), il nodo mobile seleziona il sensore fisso dal quale ha ricevuto maggior potenza. Tale algoritmo è stato implementato nel file MATLAB® punto3_metodo1.m.

4) 2 Hops – Nearest node:

Il seguente metodo, evidentemente, risulta essere la concatenazione della prima parte del metodo 3) precedentemente illustrato e dalla seconda parte del punto 2).

Tale algoritmo è stato implementato nella funzione MATLAB® punto3_metodo2.m.

I quattro algoritmi sopracitati sono stati testati per differenti valori del raggio, il quale definisce i vicini di ogni nodo.

Per quanto riguarda l'ambiente nei laboratori del INFN di Legnaro e del DEI di Padova, i valori di r utilizzati sono rispettivamente i seguenti:

$$r = 10, 20, 30, 40 \text{ m}$$

$$r = 5, 15, 25, 35 \text{ m}$$

• IMPLEMENTAZIONE DELLA RETE WIRELESS

A) Propagazione del campo elettromagnetico

Malgrado una trattazione della propagazione del campo elettromagnetico nel vuoto sia relativamente semplice, si ricordi infatti che nel vuoto la direzione di propagazione dell'onda elettromagnetica è perpendicolare al piano identificato dalle direzioni delle due oscillazioni dei campi elettrico e magnetico con velocità di propagazione costante (velocità della luce) ed indipendente dalla velocità della sorgente, dalla direzione di propagazione, e dalla velocità dell'osservatore; una trattazione della propagazione nei mezzi materiali risulta molto più complessa, si pensi ad esempio che

la velocità di propagazione dipende dalle proprietà fisiche del mezzo.

I fenomeni di maggior rilevanza che intervengono nella propagazione di un'onda elettromagnetica in un mezzo materiale sono: *riflessione*, *diffrazione*, *scattering*, e *interferenza*. Ciascuno di questi fenomeni si verifica in misura direttamente o inversamente proporzionale al rapporto tra la lunghezza d'onda del campo elettromagnetico e le dimensioni dell'oggetto su cui lo stesso collide. Purtroppo nel protocollo utilizzato la frequenza del segnale trasmesso è compresa nell'intervallo [2400,2483.5] MHz e quindi la lunghezza d'onda del campo elettromagnetico generato è compresa fra 12.08 cm e 12.5 cm: dimensioni compatibili con la maggior parte degli oggetti di uso comune comportando l'insorgere di tutti i diversi disturbi elencati sopra.

Riflessione: Quando l'onda raggiunge un oggetto riflettente con un angolo i rispetto alla normale, essa modifica la propria traiettoria tornando indietro secondo un angolo $r = -i$. Un esempio è riportato in Fig. 4.4. è chiaro che, per la legge di conservazione dell'energia, la somma delle quantità di energia rispettivamente assorbita riflessa e trasmessa è uguale alla quantità di energia incidente.

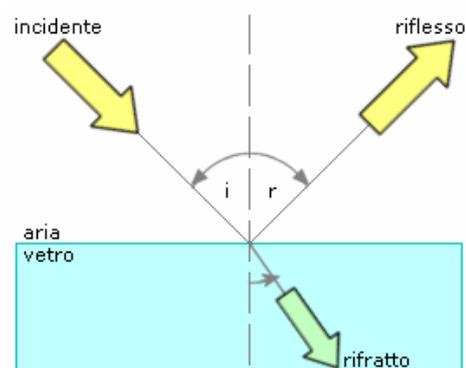


Figura 4.4: Riflessione di un'onda elettromagnetica.

La riflessione può avvenire *specularmente* (riflessione *speculare* o *regolare*) cioè in una unica (o quasi) direzione, oppure *diffusamente* (riflessione *diffusa*) cioè in varie direzioni, questo a seconda che la superficie dell'oggetto riflettente sia liscia o scabra, come riportata in Fig. 4.5.

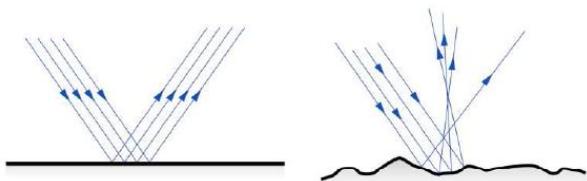


Figura 4.5: Riflessione su superficie liscia e su superficie scabra.

Diffrazione: La *diffrazione*, invece, si manifesta se un'onda incontra un ostacolo le cui *dimensioni sono comparabili o minori* rispetto alla propria lunghezza d'onda. Ad esempio, l'apertura può essere costituita da un foro circolare o rettangolare praticato in uno schermo assorbente per le onde in esame, un ostacolo da un filo, un disco assorbente o più in generale da un qualsiasi oggetto. Nel caso in cui questa colpisca una fenditura tra due corpi impenetrabili, dopo averla attraversata, si crea una superficie di onde semi-circolari, all'incirca aventi la stessa intensità della prima, che si propagano in ogni direzione come riportato in Fig. 4.6

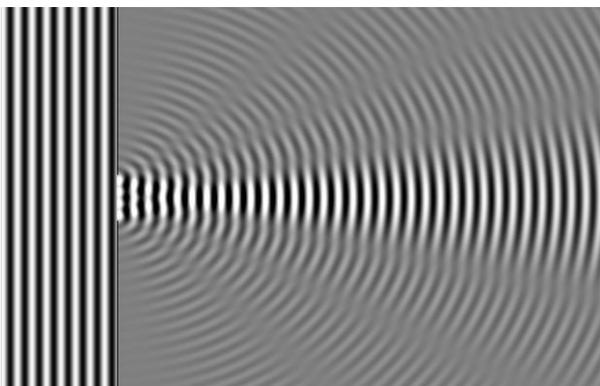


Figura 4.6: Caso particolare di Diffrazione.

Scattering: Il fenomeno dello *scattering* è dovuto alla natura corpuscolare del campo elettromagnetico e si manifesta in presenza di oggetti impenetrabili aventi dimensioni paragonabili a quelli della lunghezza dell'onda. In questo caso l'energia dell'onda che insiste su tali oggetti viene disseminata (dall'inglese *scatter*) in moltissime direzioni che nei casi reali sono imprevedibili e dipendono dalla lunghezza d'onda del campo incidente (vedi Fig. 4.7). Un esempio molto comune di scattering della luce è dato dal colore blu del cielo: la luce (bianca) del sole incide sull'atmosfera terrestre la quale

diffonde con più facilità le frequenze più alte (ovvero più vicine all'ultravioletto); di conseguenza, mentre la luce bianca ci arriva direttamente se guardiamo dritti nel sole, la luce blu diffusa ci sembra provenire da tutte le direzioni

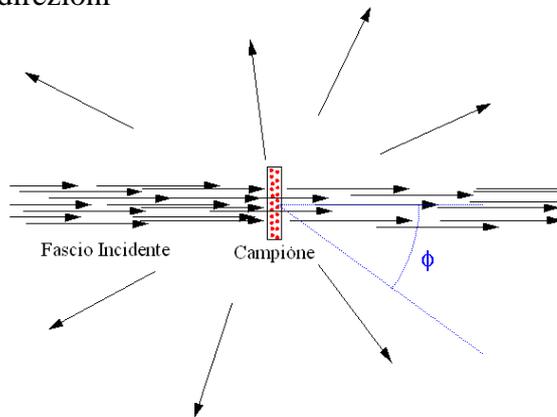


Figura 4.7: Caso particolare di Scattering su un campione.

Interferenza: Il fenomeno è dovuto alla sovrapposizione, in un punto dello spazio, di due o più onde. Quello che si osserva è che l'intensità dell'onda risultante in quel punto può essere diversa rispetto alla somma delle intensità associate ad ogni singola onda di partenza; in particolare, essa può variare tra un minimo, in corrispondenza del quale non si osserva alcun fenomeno ondulatorio, ed un massimo coincidente con la somma delle intensità. Il termine viene usualmente utilizzato per parlare di interferenza tra due onde coerenti, dove con coerenti si intendono onde, di norma provenienti dalla stessa sorgente, che presentano una differenza di fase costante nel tempo in qualsiasi punto dello spazio.

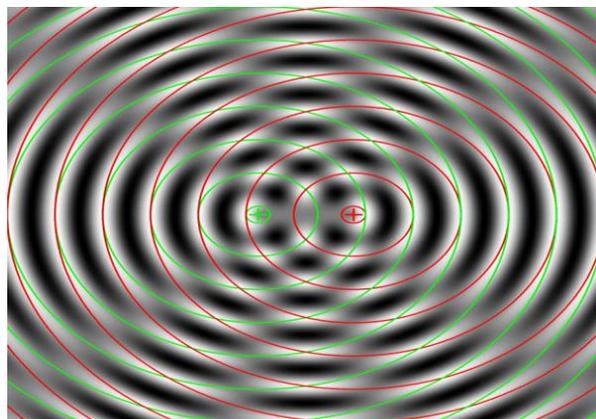


Figura 4.8: Interferenza prodotta da due sorgenti puntiformi coerenti. Si noti l'alternanza di frange chiare e frange scure.

Illustrati i problemi legati alla propagazione dell'onda elettromagnetica non ci resta che approfondire gli ambienti in cui sono stati sviluppati e testati gli algoritmi proposti.

B) Analisi dei laboratori di Legnaro

I laboratori in cui sono stati fatti gli studi di localizzazione si trovano al piano interrato: la larghezza totale dell'ambiente è pari a 37 m e la lunghezza pari a 31, per un'area totale di 1150 mq circa. L'altezza del soffitto misura mediamente 3.10 m, ma è da evidenziare la presenza di tubature che riducono l'effettiva altezza di lavoro di circa 2m. La figura 4.9 riporta una mappa 2D dei laboratori.

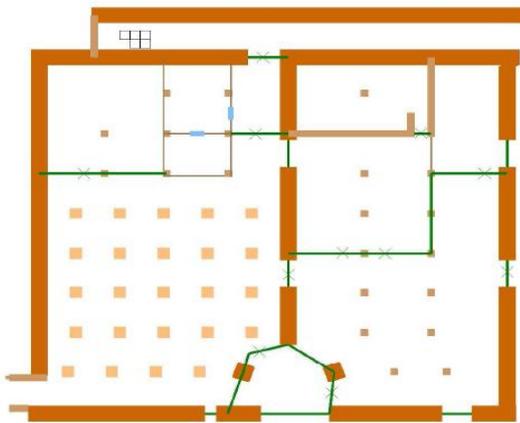


Figura 4.9: Piantina 2D dei laboratori sotterranei di Legnaro.

La disposizione dei nodi ancora è stata studiata con l'ausilio di un software specifico basato su tecniche geometriche che ha portato alla decisione di utilizzare 12 nodi ancora per una buona copertura dell'ambiente, come riportato in figura 4.10.

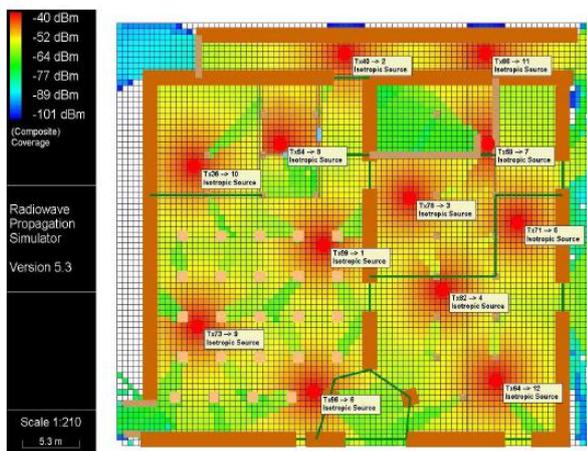


Figura 4.10: Figura di potenza del segnale RSSI nei laboratori, in rosso sono presentati i nodi ancora.

Gli algoritmi proposti in questo progetto utilizzano un set di misure (che come già annunciato metà è andato a costituire il *training set* e l'altra metà il *test set*) ricavato in [] e rappresentato in figura 4.11.

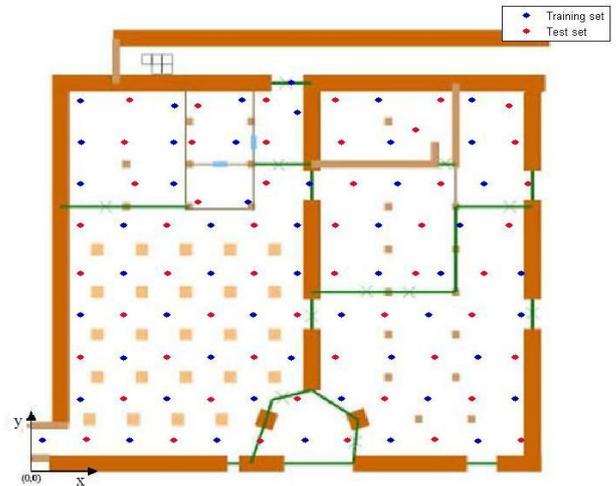


Figura 4.11: Piantina 2D dei laboratori sotterranei di Legnaro con indicati il *training set* (punti blu) e il *test set* (punti rossi).

C) Analisi dei laboratori di Padova

Per quanto riguarda i laboratori di Ingegneria dell'informazione di Padova, si tratta di un edificio con una lunghezza di 36 m e una larghezza di 14.5 m per un'area totale di 550 mq circa. Come già annunciato più volte sebbene fossero stati eseguiti studi sulla connettività il progetto ha visto l'utilizzo di tutti i 62 sensori disponibili, rappresentati in Figura 4.12

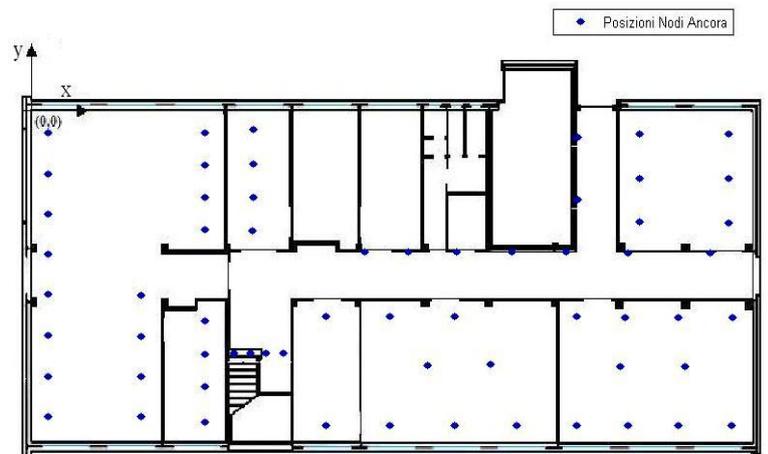


Figura 4.12: Piantina dei laboratori del dipartimento di ingegneria dell'informazione di Padova con segnati (punti blu) i nodi ancora.

Come per quanto fatto a Legnaro gli algoritmi proposti in questo progetto per i laboratori di Padova utilizzano un set di misure (che, come prima, metà è andato a costituire il *training set* e l'altra metà il *test set*) per un totale di 207 misurazioni, come rappresentato in figura 4.13.

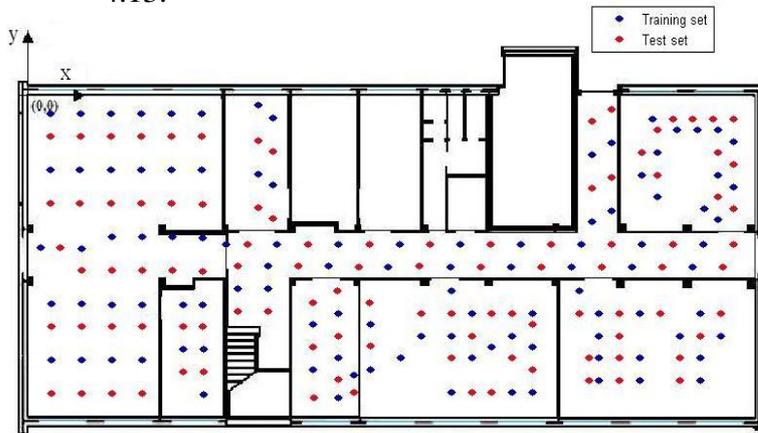


Figura 4.13: Piantina 2D dei laboratori del dipartimento di ingegneria dell'informazione di Padova, con indicati il *training set* (punti blu) e il *test set* (punti rossi).

V. ANALISI SIMULAZIONI

Ci si presta ora ad analizzare le prestazioni degli algoritmi centralizzato e decentralizzati implementati.

Il primo passo fatto è stato decidere quale tra le varie funzioni covarianza proposte in IV fosse la più indicata al nostro problema.

A tal fine si sono costruiti più modelli, ciascuno implementato da una funzione covarianza differente, e si è confrontato l'errore medio delle relative predizioni delle varie posizioni del sensore mobile.

Per ogni modello considerato, un punto fondamentale era la scelta iniziale dei parametri del kernel dai quali si ottengono quelli definitivi tramite la massimizzazione della verosimiglianza marginalizzata; difatti si ha fatto uso di uno script, creato da C.E. Rasmussen e C.K.I. Williams, che calcola un punto di massimo per approssimazioni successive dati dei valori iniziali. Per essere sicuri di aver trovato un massimo assoluto e non locale si ha ripetuto la procedura di inizializzazione più volte.

In seguito a più prove effettuate la nostra scelta è ricaduta sulla *Matérn* ($\nu = 3/2$) per quanto riguarda funzioni covarianze a numero

di parametri fissi, mentre sulla *SE* per le covarianze in cui il numero di parametri è dipendente dal dominio della funzione $f(\mathbf{x})$ corrispondente da determinare.

L'approccio centralizzato, come si è visto, implica la costruzione di un unico modello di predizione $f_x(\mathbf{x})$ per l'asse x e uno $f_y(\mathbf{x})$ per l'asse y , implementati nell'elaboratore centrale.

Viceversa l'algoritmo decentralizzato implica $2N$ modelli distinti, dove N rappresenta il numero dei sensori della rete. Per tali motivi verranno riportati solo i valori dei parametri relativi all'approccio centralizzato.

▪ Laboratori INFN di Legnaro:

| Parametri funzione covarianza <i>Matérn</i> approccio CENTRALIZZATO | | | | |
|--|-----------------|----------------|-----------------|----------------|
| | Asse x | | Asse y | |
| | Valori iniziali | Valori trovati | Valori iniziali | Valori trovati |
| l | 5 | 110.425 | 5 | 94.284 |
| σ_f^2 | 5 | 9.837 | 5 | 10.744 |
| σ_n^2 | 5 | 1.244 | 5 | 1.549 |

| Parametri funzione covarianza <i>SE</i> approccio CENTRALIZZATO | | | | |
|--|-----------------|----------------|-----------------|----------------|
| | Asse x | | Asse y | |
| | Valori iniziali | Valori trovati | Valori iniziali | Valori trovati |
| l_1 | 3 | ? | 3 | ? |
| l_2 | 3 | ? | 3 | ? |
| l_3 | 3 | ? | 3 | ? |
| l_4 | 3 | | 3 | |
| l_5 | 3 | | 3 | |
| l_6 | 3 | | 3 | |
| l_7 | 3 | | 3 | |
| l_8 | 3 | | 3 | |
| l_9 | 3 | | 3 | |
| l_{10} | 3 | | 3 | |
| l_{11} | 3 | | 3 | |
| l_{12} | 3 | | 3 | |
| σ_f^2 | 3 | | 3 | |
| σ_n^2 | 3 | | 3 | |

Determinati i valori dei parametri, si è pronti a testare l’algoritmo centralizzato e i quattro approcci decentralizzati utilizzando i dati raccolti nella matrice test_X relativo alle potenze ricevute dal nodo mobile dai vari sensori fissi test_Y_x e test_Y_y definiti dalle

posizioni assunte dal sensore mobile nell’ambiente lungo l’asse x e y rispettivamente.

I metodi decentralizzati, inoltre, vengono confrontati per vari valori del raggio r il quale definisce i vicini di ciascun nodo.

Matérn:

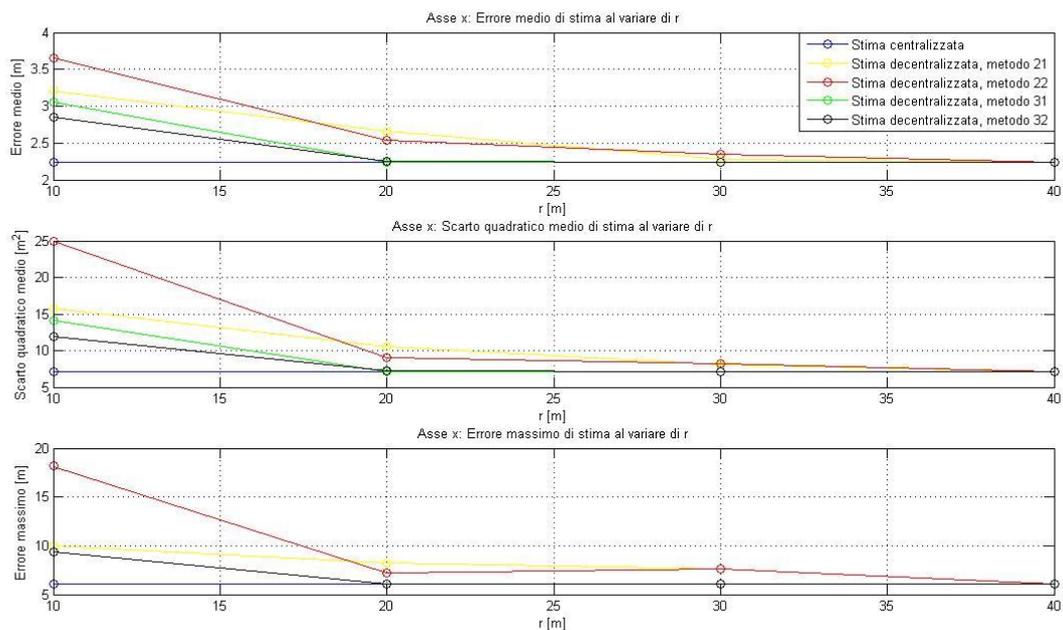


Figura .. : Prestazioni metodi proposti lungo l’asse x

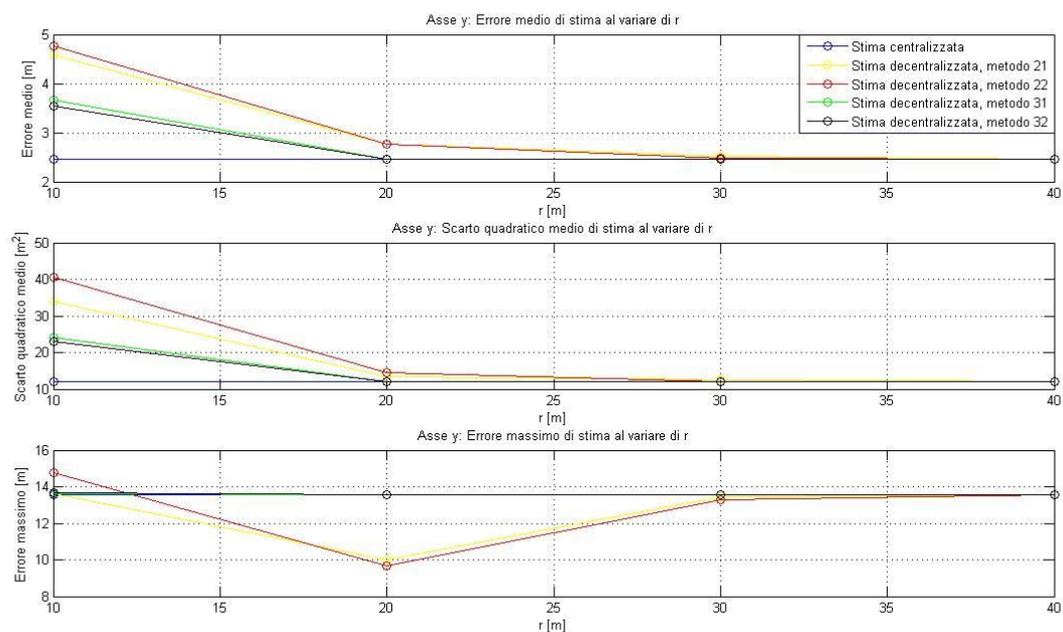


Figura .. : Prestazioni metodi proposti lungo l’asse y

Square Exponential (SE):

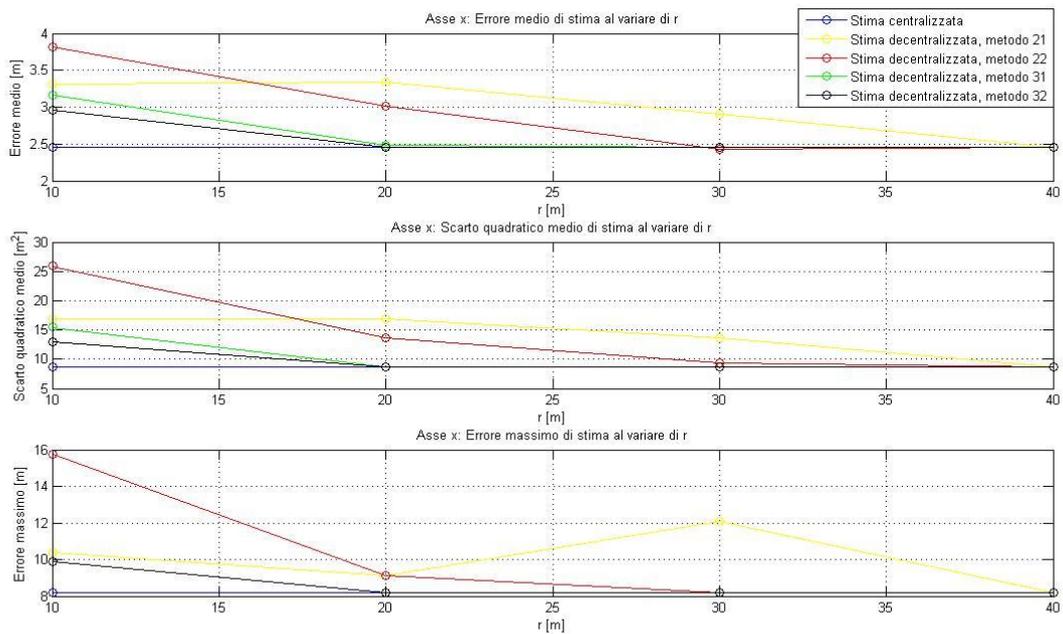


Figura .. : Prestazioni metodi proposti lungo l'asse x

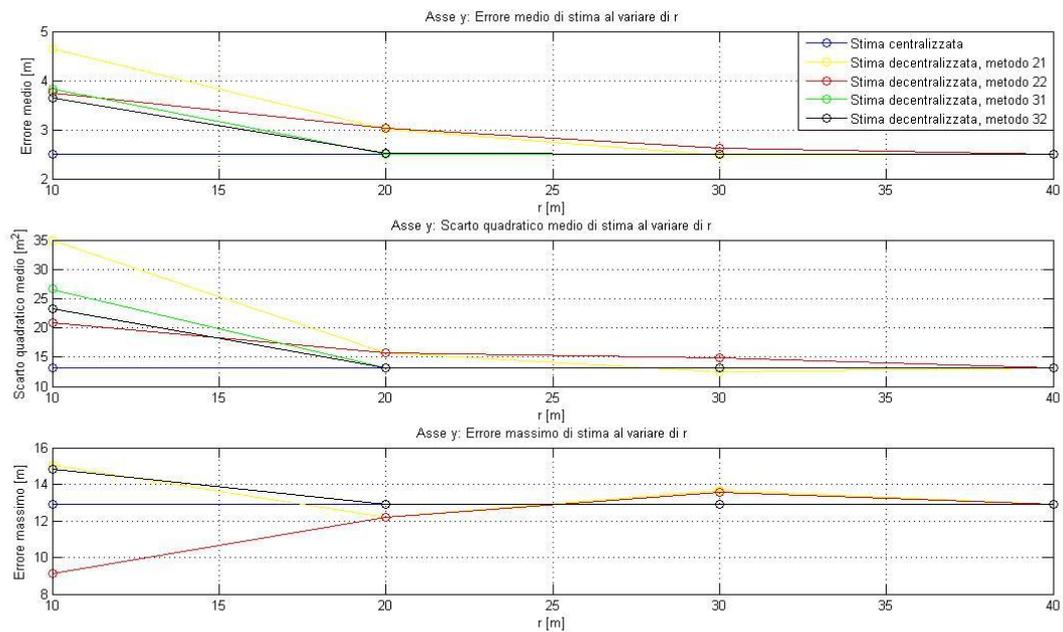


Figura .. : Prestazioni metodi proposti lungo l'asse y

Le simulazioni proposte sono state effettuate nel caso di una situazione ideale, cioè tutti i sensori della rete sono funzionanti. Ora si considera il caso in cui qualche nodo sia guasto: tale ipotesi è importante considerarla al fine di testare la robustezza degli algoritmi proposti. Il numero dei nodi guasti deciso nel caso dei laboratori di Legnaro è pari a due

(circa il 16.5% della rete) mentre la scelta dei sensori non funzionanti è stata di tipo casuale. I nodi guasti nella simulazione sono risultati il numero 10 e 11.

Le prestazioni dei metodi implementati sono apprezzabili dalle figure seguenti:

Matérn:

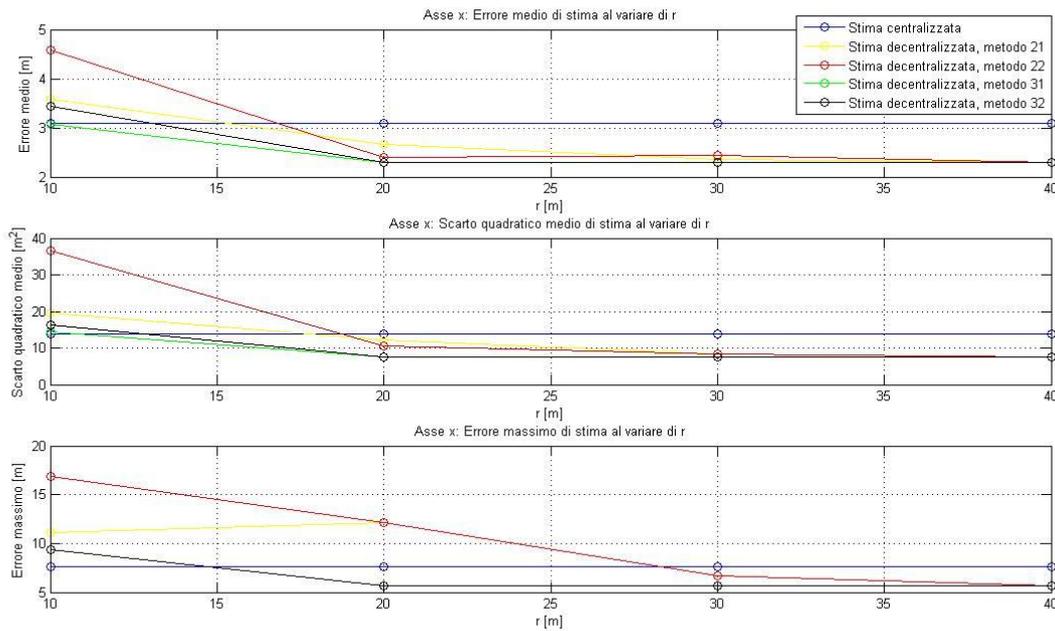


Figura .. : Prestazioni metodi proposti lungo l'asse x

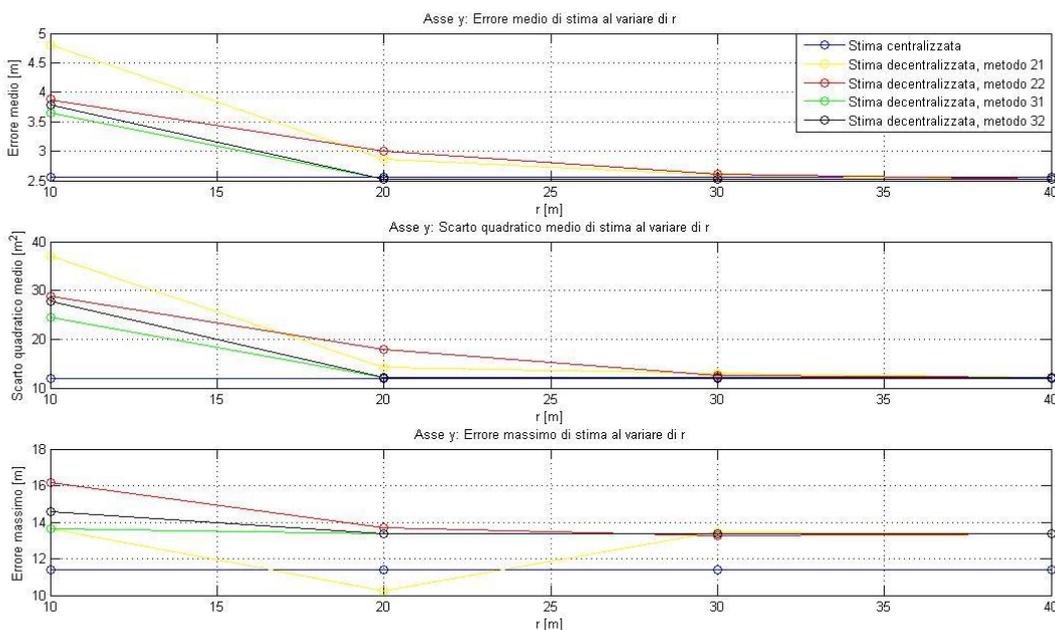


Figura .. : Prestazioni metodi proposti lungo l'asse y

Square Exponential (SE):

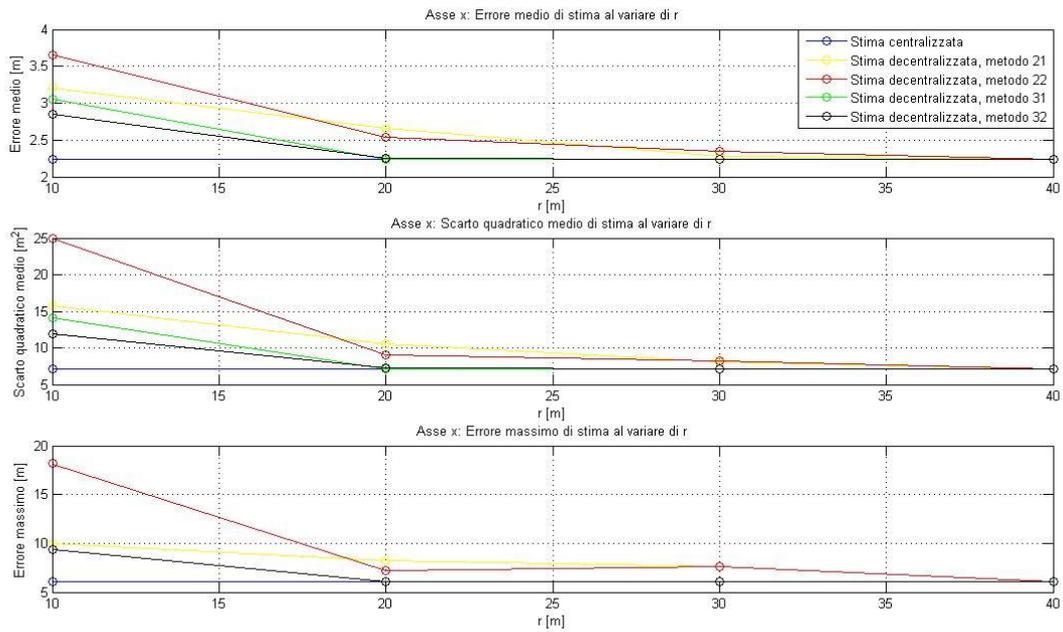


Figura .. : Prestazioni metodi proposti lungo l'asse x (DA SOSTITUIRE FIGURA)

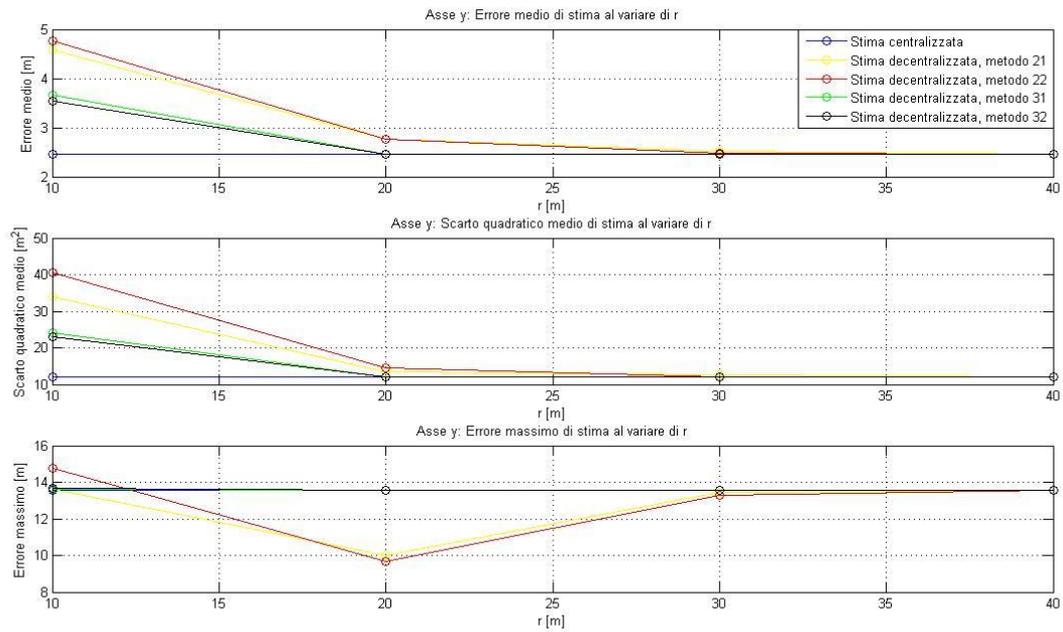


Figura .. : Prestazioni metodi proposti lungo l'asse y (DA SOSTITUIRE FIGURA)

▪ Laboratori DEI di Padova:

La rete WSN dei laboratori di Padova è costituita da 62 sensori, perciò per ogni asse di riferimento è necessario implementare una funzione $f(\mathbf{x})$ nell'elaboratore centrale nel caso di approccio centralizzato, mentre una $f(\mathbf{x})$ distinta in ogni nodo della rete per il metodo decentralizzato.

Di seguito vengono riportati i valori dei parametri della funzione covarianza trovati nel caso centralizzato.

| Parametri funzione covarianza <i>Matérn</i> approccio CENTRALIZZATO | | | | |
|--|-----------------|----------------|-----------------|----------------|
| | Asse x | | Asse y | |
| | Valori iniziali | Valori trovati | Valori iniziali | Valori trovati |
| l | 5 | 297.194 | 5 | 345.549 |
| σ^2_f | 5 | 9.205 | 5 | 7.418 |
| σ^2_n | 5 | 1.881 | 5 | 1.338 |

| Parametri funzione covarianza <i>SE</i> approccio CENTRALIZZATO | | | | |
|--|-----------------|----------------|-----------------|----------------|
| | Asse x | | Asse y | |
| | Valori iniziali | Valori trovati | Valori iniziali | Valori trovati |
| l_1 | 6 | ? | 6 | ? |
| l_2 | 6 | ? | 6 | ? |
| l_3 | 6 | ? | 6 | ? |
| l_4 | 6 | | 6 | |
| l_5 | 6 | | 6 | |
| l_6 | 6 | | 6 | |
| l_7 | 6 | | 6 | |
| l_8 | 6 | | 6 | |
| l_9 | 6 | | 6 | |
| l_{10} | 6 | | 6 | |
| l_{11} | 6 | | 6 | |
| l_{12} | 6 | | 6 | |
| l_{13} | 6 | | 6 | |
| l_{14} | 6 | | 6 | |
| l_{15} | 6 | | 6 | |
| l_{16} | 6 | | 6 | |
| l_{17} | 6 | | 6 | |
| l_{18} | 6 | | 6 | |
| l_{19} | 6 | | 6 | |

| | | | | |
|--------------|---|--|---|--|
| | 6 | | 6 | |
| l_{20} | 6 | | 6 | |
| l_{21} | 6 | | 6 | |
| l_{22} | 6 | | 6 | |
| l_{23} | 6 | | 6 | |
| l_{24} | 6 | | 6 | |
| l_{25} | 6 | | 6 | |
| l_{26} | 6 | | 6 | |
| l_{27} | 6 | | 6 | |
| l_{28} | 6 | | 6 | |
| l_{29} | 6 | | 6 | |
| l_{30} | 6 | | 6 | |
| l_{31} | 6 | | 6 | |
| l_{32} | 6 | | 6 | |
| l_{33} | 6 | | 6 | |
| l_{34} | 6 | | 6 | |
| l_{35} | 6 | | 6 | |
| l_{36} | 6 | | 6 | |
| l_{37} | 6 | | 6 | |
| l_{38} | 6 | | 6 | |
| l_{39} | 6 | | 6 | |
| l_{40} | 6 | | 6 | |
| l_{41} | 6 | | 6 | |
| l_{42} | 6 | | 6 | |
| l_{43} | 6 | | 6 | |
| l_{44} | 6 | | 6 | |
| l_{45} | 6 | | 6 | |
| l_{46} | 6 | | 6 | |
| l_{47} | 6 | | 6 | |
| l_{48} | 6 | | 6 | |
| l_{49} | 6 | | 6 | |
| l_{50} | 6 | | 6 | |
| l_{51} | 6 | | 6 | |
| l_{52} | 6 | | 6 | |
| l_{53} | 6 | | 6 | |
| l_{54} | 6 | | 6 | |
| l_{55} | 6 | | 6 | |
| l_{56} | 6 | | 6 | |
| l_{57} | 6 | | 6 | |
| l_{58} | 6 | | 6 | |
| l_{59} | 6 | | 6 | |
| l_{60} | 6 | | 6 | |
| l_{61} | 6 | | 6 | |
| l_{62} | 6 | | 6 | |
| σ^2_f | 6 | | 6 | |
| σ^2_n | 6 | | 6 | |

Le prestazioni dei metodi implementati sono visibili dalle figure seguenti:

Matérn:

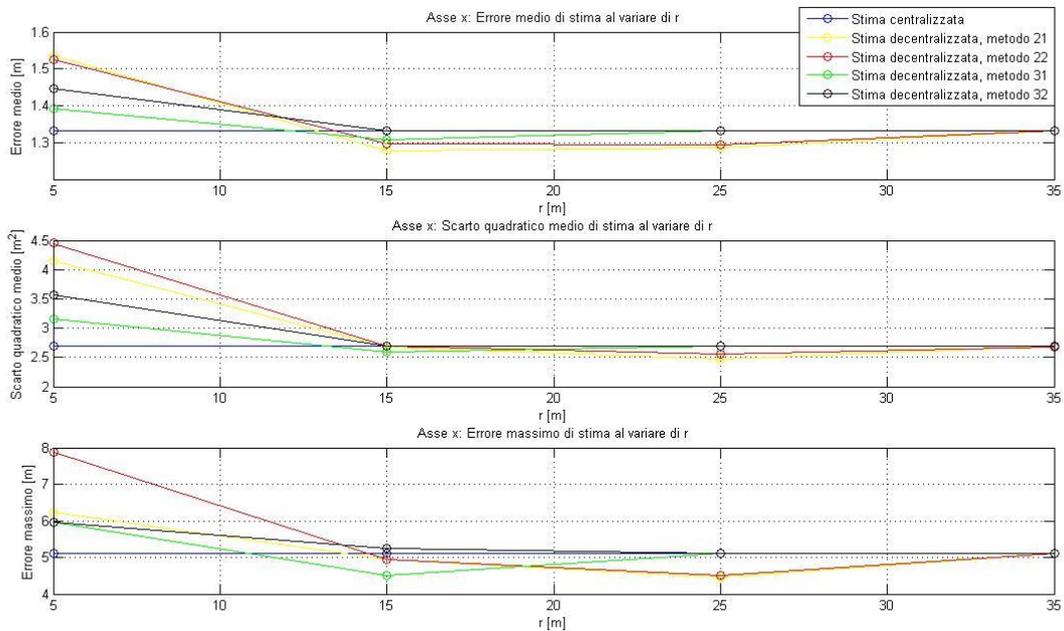


Figura .. : Prestazioni metodi proposti lungo l'asse x

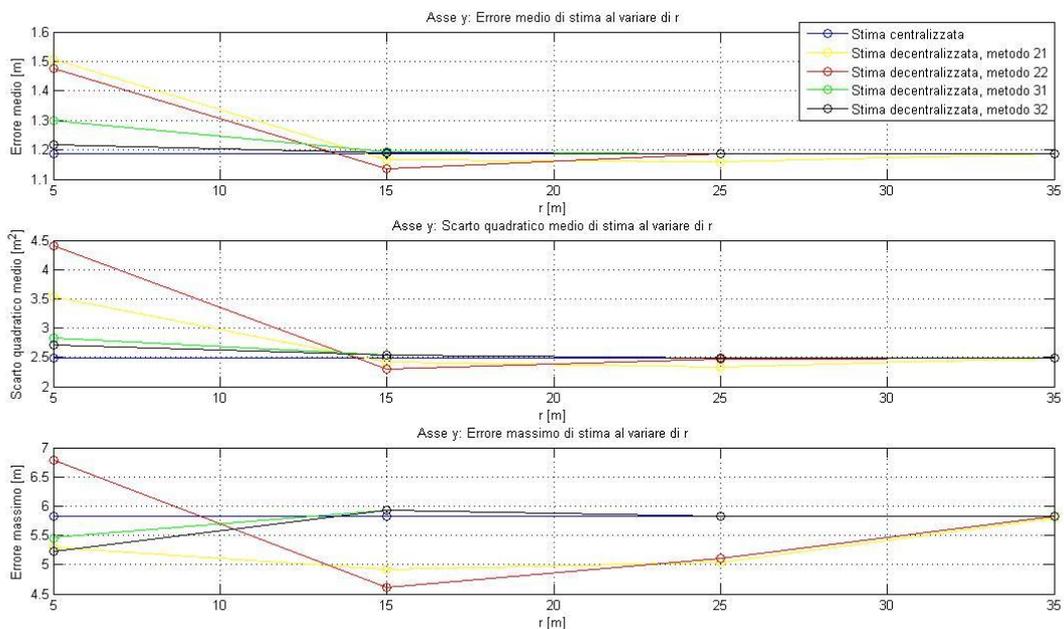


Figura ... : Prestazioni metodi proposti lungo l'asse y

Square Exponential (SE):

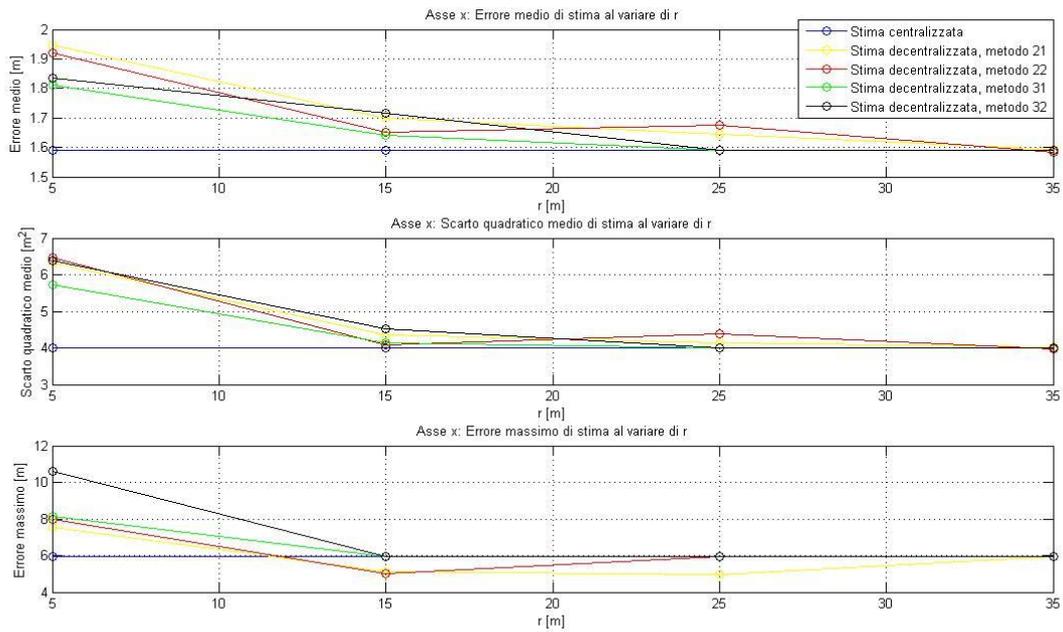


Figura .. : Prestazioni metodi proposti lungo l'asse x

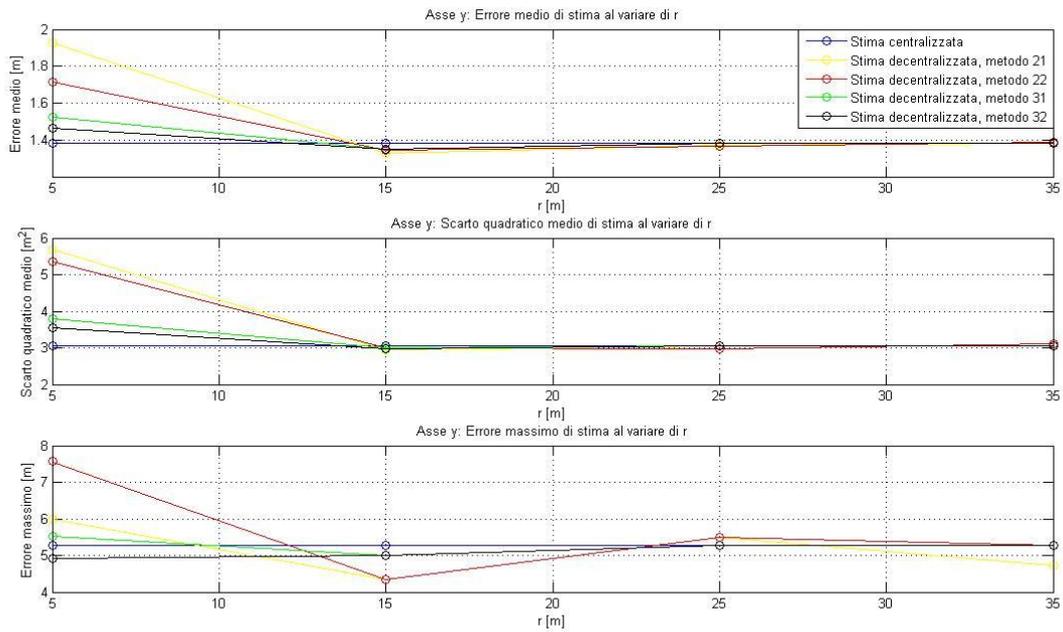


Figura .. : Prestazioni metodi proposti lungo l'asse y

Ci si presta ora ad analizzare la robustezza degli algoritmi in caso di malfunzionamento di alcuni nodi della rete. Il numero dei nodi guasti scelto è pari a 10 (circa il 16% della rete), mentre i sensori guasti, determinati in

modo aleatorio, risultano i numeri 18, 34, 61, 45, 52, 27, 30, 35, 17, 47.

I grafici delle performance degli algoritmi sono i seguenti:

Matérn:

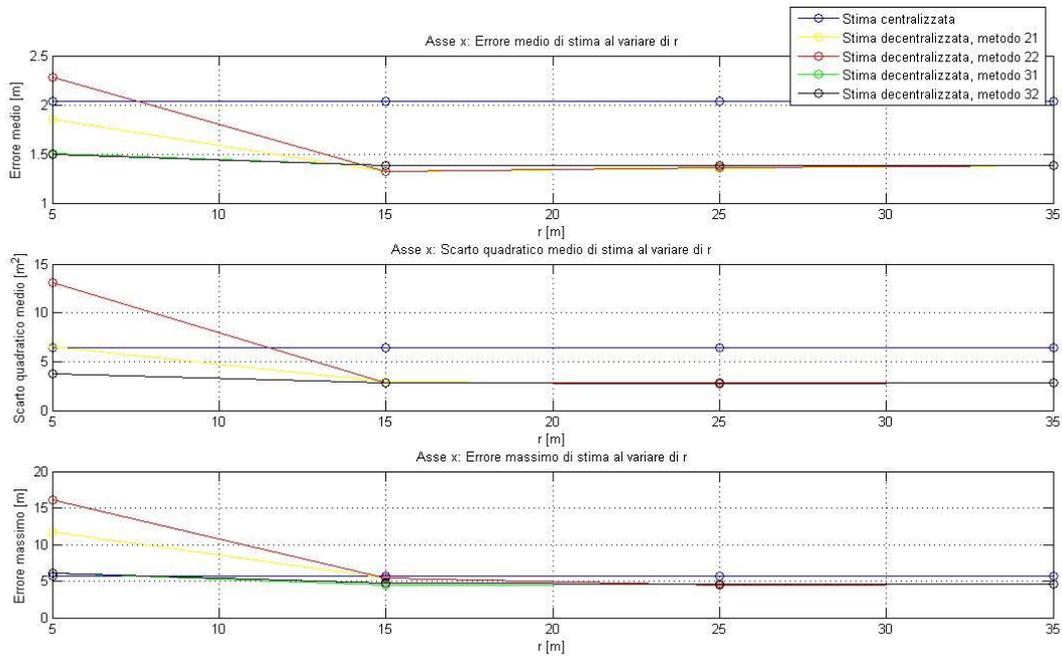


Figura .. : Prestazioni metodi proposti lungo l'asse x

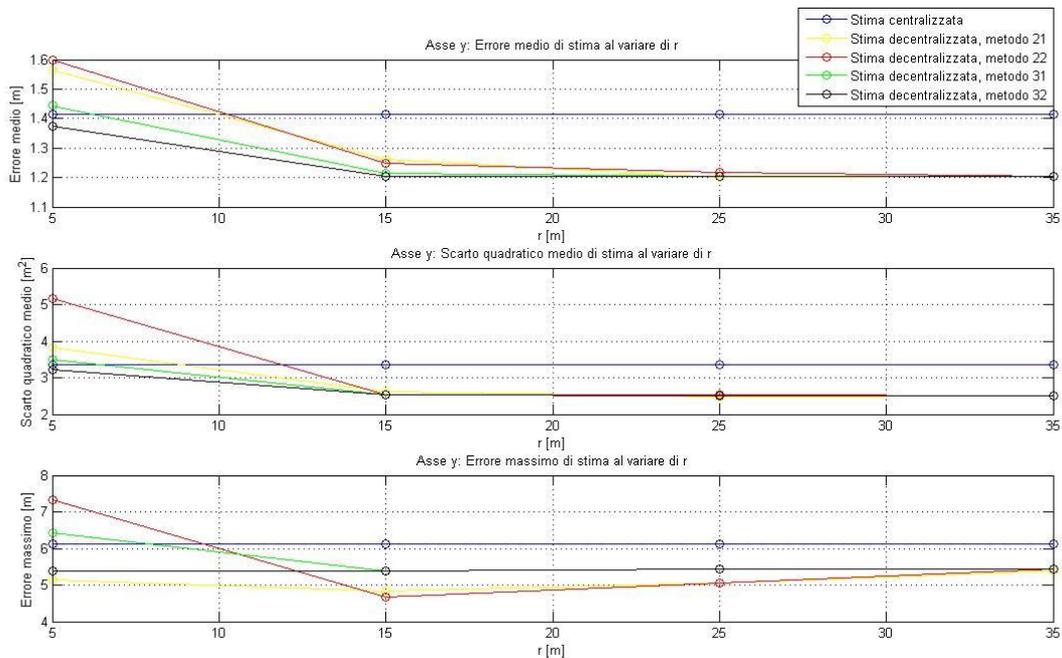


Figura .. : Prestazioni metodi proposti lungo l'asse y

Square Exponential (SE):

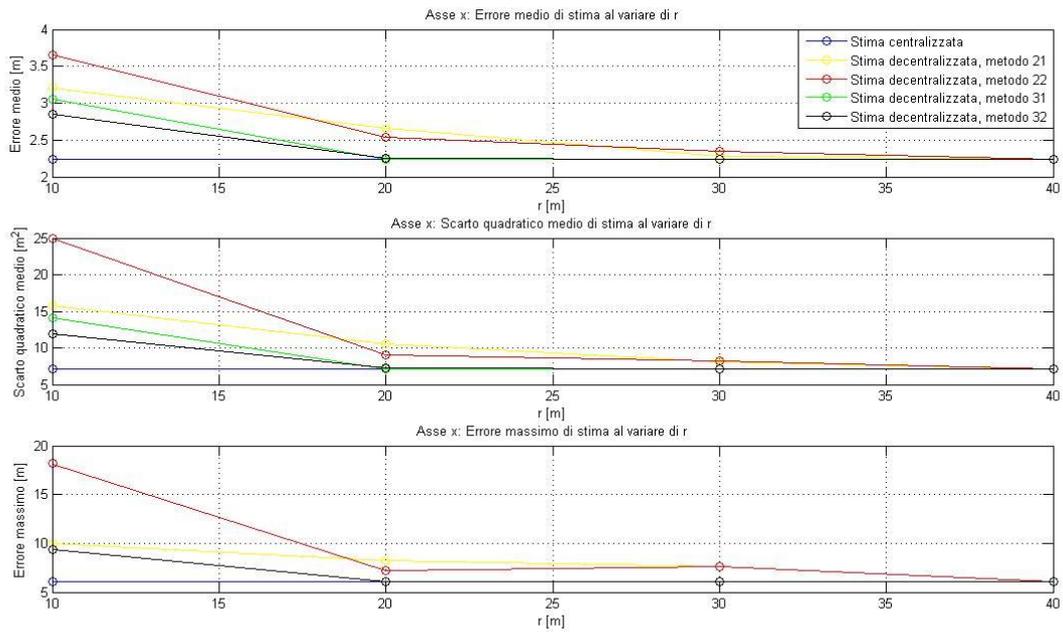


Figura .. : Prestazioni metodi proposti lungo l'asse x (DA SOSTITUIRE FIGURA)

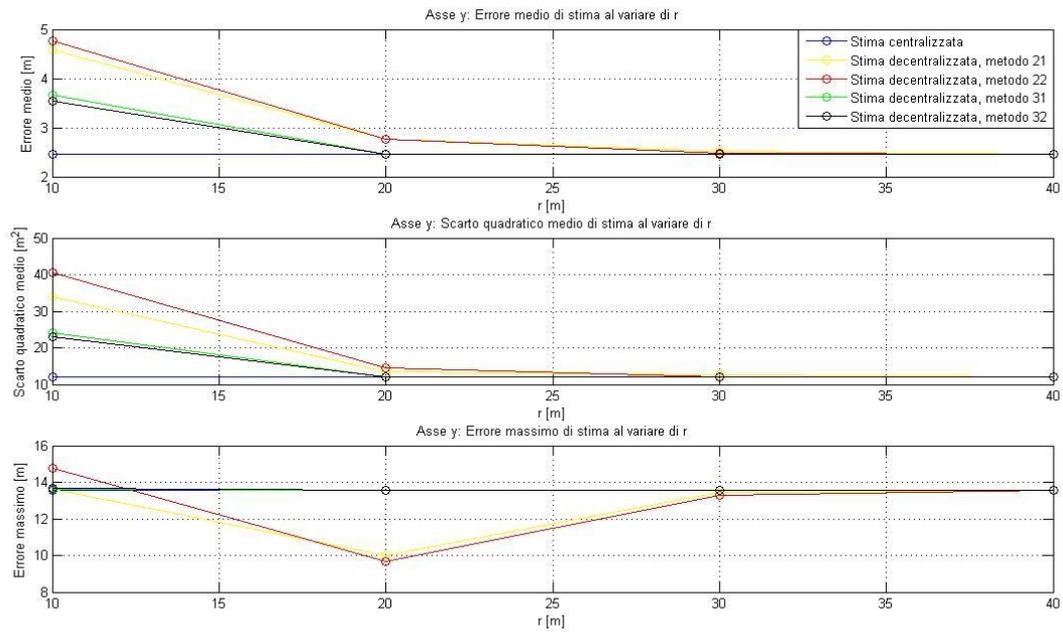


Figura .. : Prestazioni metodi proposti lungo l'asse y (DA SOSTITUIRE FIGURA)

VI. SVILUPPI FUTURI

In questa sezione tratteremo brevemente gli sviluppi che possono completare il progetto sviluppato, è sottointeso che qualsiasi innovazione e miglioramento dovranno inglobare un lavoro di ingegnerizzazione, così da rendere più snelli i calcoli e permettere una commercializzazione del sistema.

Innanzitutto si potrebbe sviluppare un ulteriore algoritmo che grazie al progetto presentato implementi il tracking del nodo mobile, magari con l'utilizzo di un Filtro di Kalman o altre tecniche avanzate di controllo; in secondo luogo si potrebbe ottimizzare l'invio di pacchetti dai nodi ancora al nodo mobile così da ritagliare capacità di calcolo che potrebbe essere utilizzata per permettere una comunicazione e interazione tra i nodi: si ricordi ad esempio che nelle tecniche decentralizzate abbiamo definito il concetto di "vicini", nelle sperimentazioni si è considerato vicino un nodo che in termini di distanza (in metri) fosse al di sotto di una soglia stabilita mentre con questa capacità di calcolo si potrebbe provare a definire vicini due nodi se la potenza che i due si inviano/ricevono sta sopra una certa soglia, definizione che ha anche più senso logico rispetto a quella utilizzata.

Un'altra idea basata su questa ulteriore capacità di calcolo è un controllo preventivo tra i sensori onde evitare spostamenti degli stessi: si pensi ad esempio ad un terremoto o in generale al crollo di una superficie che abbia vincolati dei nodi ancora, è chiaro che se questi rimangono in grado di comunicare e di fornire dati l'informazione che daranno al nodo mobile sarà più dannosa che utile nel contesto di determinare la sua posizione. Ecco perché ogni nodo ancora dovrebbe essere in grado di sapere la potenza ricevuta dai nodi ancora presenti nell'ambiente, così da rilevare anomalie di questo tipo.

In questo progetto si è sempre parlato di localizzazione di un nodo mobile, ma pensando al primo obiettivo di questo progetto, cioè ad uno strumento di supporto per una squadra di soccorso, è chiaro che si dovrà estendere il lavoro a più nodi ancora. Più in dettaglio, pensando ad una squadra di

pompieri che entra in un edificio, è facile immaginare che il movimento di queste persone non sarà casuale, ma si muoveranno tutti insieme mantenendo sempre almeno un contatto visivo tra di loro. Ecco perché si potrebbe introdurre l'idea di supportare l'algoritmo qui studiato con algoritmi di tipo collaborativo, utilizzando in qualche modo da approfondire anche l'informazione visiva che ogni pompiere ha sugli altri.