

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea Magistrale in
Ingegneria Elettronica e Telecomunicazioni per l'Energia

Localizzazione e navigazione congiunta di droni

Elaborato in
Reti di Sensori Wireless

Candidato
NICOLA SPARNACCI
Matricola
784491

Relatore
Chiar.mo. Prof.
DAVIDE DARDARI

Correlatori
Dott.sa
ANNA GUERRA
Dott.
NICOLÒ DECARLI

III SESSIONE DI LAUREA
ANNO ACCADEMICO 2017-2018

Dedicato a tutti coloro che hanno creduto in me e che mi hanno supportato.

Siete la mia forza.

Sommario

In questa tesi vengono analizzati aspetti legati alla navigazione e localizzazione congiunta di Unmanned Aerial-Vehicle (UAV) per la ricerca della formazione ottimale in modo da minimizzare l'errore di stima della posizione di un target, situato in ambienti indoor piuttosto che outdoor. Attraverso l'uso di modelli d'osservazione realistici che tengono conto anche di eventuali scenari in cui vi sia un collegamento Non-Line of Sight (NLOS) tra target e UAV, vengono derivate le grandezze fondamentali legate al Cramér-Rao Lower Bound (CRLB) come la Fisher Information Matrix (FIM) e il Position Error Bound (PEB) per l'implementazione un algoritmo (centralizzato/distribuito) di ottimizzazione vincolata per il calcolo della formazione ottimale. Sono descritte metodologie e principi utili per creare un algoritmo di navigazione degli UAV tale che permetta di districarsi in ambienti caratterizzati dalla presenza di numerosi ostacoli (cluttered environments) e di evitare la collisione tra UAV, mantenendo condizioni di sicurezza. Nell'ipotesi in cui gli UAV siano in grado di comunicare e collaborare, vengono fornite le basi di teoria dei grafi per l'instaurazione della communication multi-hop rispettando i vincoli di massima latenza intrinseci alle applicazioni real-time quali la localizzazione di un target, aspetto non considerato in letteratura. Tramite simulazioni al calcolatore, sono stati ottenuti e discussi risultati al variare di un grande numero di parametri per studiare la prestazione dell'algoritmo in un vasto numero di circostanze. Viene effettuata una comparazione tra la versione centralizzata e distribuita dell'algoritmo fissando il numero di hops massimi tollerabili per rispettare vincoli di latenza e ipotizzando di sfruttare un buffer di memoria nel quale memorizzare le ultime informazioni ricevute da UAV posti a distanza troppo elevata. Il confronto viene effettuato tramite analisi di performance assoluta nella stima della posizione e attraverso la definizione di una metrica per il calcolo della percentuale e velocità di convergenza degli algoritmi. I risultati hanno mostrato come il numero di UAV che compongono la rete giochi un ruolo fondamentale in termini di velocità di convergenza fissata l'accuratezza di localizzazione desiderata anche in presenza di un approccio distribuito single hop. Infatti, il numero di UAV utilizzati incide nel miglioramento dell'accuratezza di localizzazione sia nella versione centralizzata che distribuita dell'algoritmo. Attraverso i risultati sono state ottenute importanti linee guida sul dimensionamento della rete al variare delle condizioni operative.

Abstract

In this Thesis joint navigation and localization algorithms with Unmanned Aerial Vehicles (UAV) are investigated. The purpose of the UAV network is to find the optimal formation minimizing the position estimation error of an indoor/outdoor target. Using realistic observation models, taking into account Non-Line of Sight (NLOS) connections due to obstacles, fundamental estimation limits based on the Cramér-Rao Lower Bound (CRLB) such as Fisher Information Matrix (FIM) and Position Error Bound (PEB) are calculated to design centralized/distributed navigation and localization algorithms. These algorithms also consider obstacle collision avoidance in order to use UAVs in cluttered environments and an inter-UAV safety distance to avoid collisions between UAVs. Assuming that each UAV is able to communicate and collaborate with other UAVs, fundamentals of graph theory are exploited to create a multi-hop network according to latency constraints of the real-time application. Note that latency constraints are not taken into account in literature. A rich numerical investigation in different scenarios and configurations is carried out using computer simulations. A comparison between centralized and distributed algorithms is made when the same number of UAVs is deployed and for different communication ranges. The analysis is focused on absolute localization performance and a newly defined metric to show the speed and the rate of convergence. Results show that the number of UAVs composing the dynamic network is crucial to achieve the desired accuracy with a good speed of convergence. In fact, the greater number of UAVs are deployed, the better localization accuracy is obtained, both in centralized and distributed algorithms. From these results, important network design guidelines are obtained.

Ringraziamenti

VORREI ringraziare il Prof. Davide Dardari e la Dott. Anna Guerra per avermi garantito un utile supporto e per avermi fornito preziosi consigli nella stesura della tesi, nei contenuti e nella forma, affinché il lavoro giungesse a compimento nel migliore dei modi. Vi ringrazio per avermi permesso di sottoporre ad una commissione di una conferenza internazionale il mio primo articolo scientifico come co-autore. Nonostante i suggerimenti ricevuti, tengo a precisare che la responsabilità di eventuali errori presenti in questa tesi è unicamente mia e non imputabile a loro.

Un ringraziamento va agli amici più cari che nei momenti più faticosi ed intensi mi sono stati d'aiuto nel distogliere l'attenzione e il pensiero dal lavoro. Tengo anche a ringraziare gli amici/colleghi universitari, in quanto ritengo che siano stati fondamentali nell'affrontare il percorso di studi. Grazie per le risate e i momenti vissuti insieme, li porterò sempre con me.

Infine, il ringraziamento più sentito va a Chiara e ai miei genitori. Solo voi potete sapere l'impegno che ho messo nell'affrontare la prova finale sono e felice di avervi avuto al mio fianco in ogni istante. Sono legato a voi più che a chiunque altro e l'affetto e il sostegno che mi avete garantito in questi anni universitari sono stati per me fondamentali. Per questo, è a voi che dedico questo mio lavoro.

Indice

Elenco delle figure	xiii
Elenco delle tabelle	xv
Lista dei simboli principali	xvii
Lista degli acronimi	xxi
1 Introduzione	1
1.1 Motivazioni e contributo della tesi	1
1.2 Tecnologie per la localizzazione indoor	3
1.2.1 Ultra-Wideband	4
1.2.2 Onde Millimetriche e Array di Antenne Massivi	5
1.2.3 Near-Field Electromagnetic Ranging	7
1.3 Metodi per la localizzazione	9
1.4 Filtri Bayesiani	11
1.5 Wireless Sensor Networks con UAV	14
1.6 Algoritmi di information seeking	16
1.7 Organizzazione del documento	19
2 Limiti fondamentali di localizzazione con UAV network	21
2.1 Modello centralizzato con varianza dell'errore di stima costante	22
2.2 Modello centralizzato con varianza dell'errore di stima dipen-	
dente dalla distanza	28
2.3 Modelli con bias	31
2.4 Modello di osservazione distribuito	34
3 Algoritmo di navigazione	39
3.1 Ottimizzazione non vincolata	41
3.1.1 Varianza dell'errore di stima costante e indipendente	
dalla distanza	41

3.2	Ottimizzazione vincolata	42
3.2.1	Metodo dei coefficienti di Lagrange	43
3.2.2	Metodo del gradiente proiettato	45
3.2.3	Collision avoidance	48
	Collision avoidance fra nodi	48
	Obstacles avoidance: Artificial Potential Field	51
	Obstacles avoidance: sensor model	55
	Controllo della velocità	60
3.2.4	Equazione di controllo vincolato	61
3.3	Algoritmo distribuito	65
3.3.1	Modelli di connettività	65
3.3.2	Fondamenti di teoria dei grafi	66
3.3.3	Equazione di controllo distribuito	69
4	Simulazioni con algoritmo centralizzato	73
4.1	Algoritmo di controllo con posizione del target nota	74
4.1.1	Ottimizzazione libera	74
4.1.2	Ottimizzazione vincolata con varianza d'errore di stima dipendente dalla distanza	79
4.1.3	Emettitore dinamico in ambiente outdoor e indoor	80
4.1.4	Performance dell'algoritmo al variare del numero di nodi	85
4.2	Algoritmo di controllo con stima della posizione dell'emettitore	88
4.2.1	Performance dell'algoritmo al variare della velocità dei nodi	92
4.2.2	Simulazione al variare del numero di nodi	94
4.3	Simulazioni in condizioni di Non-Line of Sight	97
5	Simulazioni con algoritmo distribuito	107
5.1	Singola realizzazione con algoritmo distribuito	108
5.2	Analisi statistica dell'algoritmo distribuito	111
5.2.1	Dipendenza dal communication range	114
5.2.2	Dipendenza dal numero massimo di hops	116
5.2.3	Dipendenza dal numero di UAV	117
6	Conclusioni	121
A	Collaborative Target-Localization and Information-based Control in Networks of UAVs	123
	Bibliografia	129

Elenco delle figure

1.1	Comparazione segnali Narrowband e Ultra-Wideband.	5
1.2	Rappresentazione delle onde dei campi \mathbf{E} , \mathbf{H} in condizioni di near e far field	7
1.3	Schema a blocchi degli stimatori di posizione.	9
1.4	Classificazione degli UAV.	14
2.1	Rappresentazione grafica del caso di un emettitore in una zona in cui non è disponibile il GPS e UAV situati all'esterno della zona che devono stimare la sua posizione.	22
2.2	Analisi della geometria legata alle posizioni dei nodi.	25
2.3	Istogramma della frequenza del valore di bias ottenuti in [48] con misure in ufficio.	31
3.1	Rappresentazione monodimensionale delle mosse di proiezione e ripristino.	47
3.2	Collision avoidance di un numero variabile di nodi e in diverse configurazioni iniziali.	50
3.3	Rappresentazione del potenziale di un ostacolo rettangolare.	53
3.4	Visualizzazione grafica delle equazioni dei vincoli di segmento.	53
3.5	Calcolo della violazione dei vincoli.	55
3.6	Rappresentazione del sensore.	56
3.7	Rappresentazione dei rettangoli creati tramite le intersezioni del sensore con l'ostacolo.	58
3.8	Funzione potenziale $\gamma_i(d_i(\mathbf{p}_i, \mathbf{p}_{c_i})) = \rho e^{-\tau(d_i(\mathbf{p}_i, \mathbf{p}_{c_i}) - d_0)}$ per il calcolo dei coefficienti di correzione con $\rho = 0.01$ e τ variabile.	59
3.9	Schema a blocchi per localizzazione e navigazione congiunta degli UAV tramite un algoritmo centralizzato.	64
3.10	Rappresentazione grafica dei modelli di coverage.	66
3.11	Collegamenti tra nodi per single-hop e multi-hop.	68
3.12	Schema a blocchi per localizzazione e navigazione congiunta dell' i -esimo UAV tramite un algoritmo distribuito.	71

4.1	Posizionamento degli UAV con varianza dell'errore di stima costante per $n = 4, 5$.	75
4.2	Evoluzione del PEB per 4 nodi inizialmente disposti in modo casuale attorno ad un valor medio.	77
4.3	PEB al variare del numero di UAV.	78
4.4	Formazione di 4 nodi disposti in linea in uno scenario in cui gli ostacoli sono noti e collision avoidance attivo.	81
4.5	Formazione dinamica degli UAV nel caso in cui l'emettitore si muova casualmente all'interno di un rettangolo.	83
4.6	Formazione dei nodi in condizioni di emettitore in movimento libero, dove k rappresentano il numero di iterazioni dell'algoritmo.	84
4.7	Confronto delle traiettorie e del PEB per un numero variabile di nodi.	87
4.8	Differenza dell'evoluzione del PEB per il caso in cui le coordinate dell'emettitore siano note e nel il caso siano stimate da uno stimatore ML.	90
4.9	PEB nel caso in cui vari σ_0 con posizione dell'emettitore nota e stimata.	91
4.10	Andamenti del PEB con posizione dell'emettitore nota e con posizione dell'emettitore stimata tramite stimatore ML.	93
4.11	Andamento del PEB al variare del numero dei nodi, con posizione dell'emettitore stimata.	94
4.12	Probabilità di convergenza all'aumentare dei nodi e per diverse distanze iniziali.	96
4.13	Andamento del PEB al variare del numero di nodi in presenza di regioni di bias dal valore noto.	99
4.14	Model mismatch.	101
4.15	Zona NLOS creata da un ostacolo: i nodi rossi sono LOS mentre quelli blu, che si trovano nella regione grigia, sono NLOS.	102
4.16	Percentuali di convergenza: comparazione LOS e NLOS.	104
4.17	PEB al variare del bias con distanza iniziale $d = 20\text{m}$ per diversi n dopo $k = 10000$ e $k = 50000$ iterazioni.	105
4.18	PEB al variare del bias per diverse d con $n = 10$ dopo $k = 10000$ e $k = 50000$ iterazioni.	105
5.1	Traiettorie e PEB di ciascun UAV per due connection ranges.	109
5.2	Scenario dove viene evidenziata la regione dedicata agli UAV per la partenza.	112
5.3	Performance e convergenza dell'algoritmo distribuito al variare di d_{hop} .	115
5.4	Andamento del PEB medio al variare del numero dei nodi n e del numero massimo di hops h_{max} con $d_{\text{hop}} = 20\text{m}$	116

5.5 Performance e convergenza dell'algoritmo distribuito al variare di n 119

Elenco delle tabelle

1.1	Risoluzione approssimativa di localizzazione ottenibile per alcune tecnologie standardizzate derivanti [15].	6
2.1	Performance degli schemi classici di LOS/NLOS detection. . .	34
4.1	Risultati di simulazione: minimo PEB e iterazione k^* al quale è stato raggiunto.	86
4.2	Simulazioni Monte Carlo per l'analisi della convergenza al variare della distanza e del numero dei nodi.	96

Lista dei simboli principali

Simbolo	Descrizione	Unità	Pagina
$\mathcal{A}^{(k)}$	Matrice adiacenze al time slot k .	-	67
$A_i(\mathbf{p}_t, \mathbf{p}_i)$	Coefficiente i -esimo che caratterizza la qualità del collegamento.	m^{-2}	26
α	Esponente path-loss.	-	28
$\beta_i^{(k)}$	Bias nella misura del nodo i -esimo al time slot k .	m	35
d_{hop}	Trasmission range UAV.	m	xii
$d_i(\mathbf{p}_t, \mathbf{p}_i)$	Distanza euclidea dall'emettitore del nodo i -esimo.	m	23
$d_{ij}^{(k)}$	Distanza coppia nodo i, j al time slot k .	m	65
$d_i^{(k)}$	Distanza tra il nodo i -esimo e il target al time slot k .	m	35
$d_j(\mathbf{p}_t, \mathbf{p}_j)$	Distanza euclidea dall'emettitore del nodo j -esimo.	m	24
$\mathcal{E}^{(k)}$	Set di collegamenti all'istante k del grafo $\mathcal{G}^{(k)}$.	-	66
$\epsilon_i^{(k)}$	Variabile aleatoria data dalla somma $v_i^{(k)} + \beta_i^{(k)}$ per il nodo i -esimo.	m	35
$\mathcal{G}^{(k)}$	Grafo composto dal set di vertici $\mathcal{V}^{(k)}$ e collegamenti $\mathcal{E}^{(k)}$ all'istante k .	-	66
\mathbf{g}	Vettore delle violazioni dei vincoli attivi.	-	46
$h_{ij}^{(k)}$	Numero di hop tra la coppia di nodi i, j al time slot k .	-	35

Simbolo	Descrizione	Unità	Pagina
h_{\max}	Massimo numero di hops.	-	114
$\mathbf{J}(\mathbf{p}_t; \mathbf{p})$	Fisher Information Matrix.	m^{-2}	23
$\mathbf{J}_i(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)})$	FIM calcolata dall' i -esimo nodo con possibili posizioni non aggiornate.	m^{-2}	36
$\Lambda(\mathbf{z} \mathbf{p}_t)$	Likelihood function.	-	23
$\Lambda_i(\mathbf{z}_i^{(k)} \mathbf{p}_{t_i}^{(k)})$	Likelihood function con possibili posizioni non aggiornate.	-	36
$\mathbf{M}(\theta_i)$	Matrice della geometria.	-	26
\mathbf{N}	Matrice dei gradienti dei vincoli attivi.	-	44
$v_i^{(k)}$	Rumore di stima nella misura del nodo i -esimo al time slot k .	m	35
\mathbf{P}	Matrice di proiezione.	-	45
\mathbf{p}	Vettore posizione di tutti i nodi.	m	23
$p_c(x, y)$	Potenziale artificiale circolare.	-	52
\mathbf{p}_{c_i}	Punto critico rilevato dal sensore.	m	57
$\text{PEB}(\mathbf{p}_t; \mathbf{p})$	Position Error Bound.	m	27
$\overline{\text{PEB}}_c^{(k)}$	PEB mediato sulle iterazioni di Monte Carlo per l'algoritmo centralizzato.	m	106
$\overline{\text{PEB}}_d^{(k)}$	PEB mediato sul numero di nodi e iterazioni di Monte Carlo per l'algoritmo distribuito.	m	112
$\text{PEB}_i(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)})$	PEB del nodo i -esimo per l'algoritmo distribuito	m	36
PEB_{goal}	Soglia per il calcolo della metrica $\text{SR}^{(k)}$.	m	111
$\text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p})$	PEB con stima della posizione del target	m	41
φ_i	Angolo creato tra l' i -esimo nodo e il punto più vicino dell'ostacolo.	rad	58
\mathbf{p}_i	Vettore posizione del nodo i -esimo.	m	23
$\mathbf{p}_i^{(k)}$	Posizione del nodo i -esimo al time slot k .	m	34

Simbolo	Descrizione	Unità	Pagina
$\mathbf{p}_j^{(k)}$	Posizione del nodo j -esimo al time slot k .	m	65
$p_r(x, y)$	Potenziale artificiale rettangolare.	-	54
\mathbf{p}_t	Vettore posizione dell'emettitore.	m	22
$\hat{\mathbf{p}}_t$	Stima della posizione del target.	m	40
$\mathbf{p}_{t_i}^{(k)}$	Posizione dell'emettitore allo step k utilizzata dal nodo i .	m	35
$\mathbf{p}_t^{(k)}$	Vettore posizione dell'emettitore al time slot k .	m	22
$\mathbf{q}_i^{(k)}$	Vettore delle posizioni conosciute al nodo i -esimo al time slot k .	m	36
$\mathcal{R}_i^{(k)}$	Set dei vicini del nodo i -esimo.	-	66
σ_0	Deviazione standard del rumore di stima ad un metro.	m	23
σ_b^2	Varianza introdotta dal bias.	m ²	32
σ_i^2	Varianza del nodo i -esimo.	m ²	23
$\sigma_i^{2(k)}$	Varianza del nodo i -esimo al time slot k .	m ²	22
$\text{SR}^{(k)}$	Metrica utilizzata nell'analisi dell'algoritmo distribuito.	-	111
θ_i	Angolo del nodo i -esimo rispetto le coordinate x, y .	rad	25
$\mathcal{V}^{(k)}$	Set di vertici all'istante k del grafo $\mathcal{G}^{(k)}$.	-	66
\mathbf{z}	Vettore dei segnali ricevuti dai nodi nel caso centralizzato.	m	22
$\mathbf{z}_i^{(k)}$	Vettore delle misure disponibili al nodo i -esimo al time slot k .	m	35

Lista degli acronimi

Acronimo	Descrizione	Pagina
AOA	Angle of Arrival	6
APF	Artificial Potential Field	51
AUV	Autonomous Underwater Vehicle	18
AWGN	Additive White Gaussian Noise	21
BS	Base Station	6
CDF	Cumulative Density Function	111
CRLB	Cramér-Rao Lower Bound	23
EKF	Extended Kalman Filter	13
ENAC	Ente Nazionale Aviazione Civile	16
FANET	Flying Ad-Hoc Network	15
FIM	Fisher Information Matrix	23
GDOP	Geometric Dilution of Precision	17
GPS	Global Positioning System	21
KF	Kalman Filter	11
LOS	Line of Sight	31
MAP	Maximum A Posteriori	14
ML	Maximum Likelihood	40
MMSE	Minimum Mean Square Error	14
NFER	Near-Field Electromagnetic Ranging	7
NLOS	Non-Line of Sight	32
PEB	Position Error Bound	27
PF	Particle Filter	11
RSS	Received Signal Strength	9
SAPR	Sistemi Aeromobili a Pilotaggio Remoto	16
SNR	Signal-Noise Ratio	4
SPAWC	Signal Processing Advances in Wireless Communications	123

Acronimo	Descrizione	Pagina
TDOA	Time Difference of Arrival	9
TOA	Time of Arrival	6
UAV	Unmanned Aerial Vehicle	20
UAV-BS	UAV-Base Station	16
UKF	Unscented Kalman Filter	13
ULA	Uniform Linear Array	10
UWB	Ultra-Wideband	28

Introduzione

1.1 Motivazioni e contributo della tesi

Lo sviluppo tecnologico ha permesso la diffusione di dispositivi tascabili di straordinaria potenza computazionale e in grado di fornire servizi personalizzati a ciascun utente in base alle sue esigenze. Tali servizi spesso richiedono la conoscenza della posizione dell'utente non solo in ambienti outdoor, dove è facilmente reperibile tramite *Global Positioning System* (GPS), ma anche in ambienti indoor. È allora emersa la necessità di approfondire lo studio della localizzazione indoor, potenzialmente in grado di rendere possibili una grande quantità di servizi e applicazioni. Una delle applicazioni più immediate è quella di fornire un servizio di navigazione simile a quella dei navigatori GPS. Questo servizio potrebbe essere in grado di fornire indicazioni per intraprendere il percorso più rapido per raggiungere un punto di interesse all'interno di un edificio e potrebbe tornare utile in edifici di servizio pubblico come ad esempio stazioni ferroviarie, ospedali, aeroporti o centri commerciali. Un altro esempio applicativo potrebbe essere a livello aziendale, per tracciare i movimenti di veicoli e macchinari mobili anche all'interno dei fabbricati. Questa applicazione garantirebbe un passo in avanti nella sicurezza sul lavoro. Basti pensare, ad esempio, alla possibilità di segnalare ad un lavoratore la presenza di un veicolo in avvicinamento e che si trova ad una distanza rischiosa, inferiore ad un valore minimo considerato di sicurezza. Un'applicazione particolarmente interessante è quella in cui vengono utilizzati dei dispositivi in ambiente outdoor, con posizioni note grazie al GPS, per localizzare un altro dispositivo situato in zone in cui la sua posizione non è nota, come si verifica se quest'ultimo si trova indoor. Infatti, quest'applicazione potrebbe rivelarsi molto utile in operazioni di salvataggio in quanto potrebbe, ad esempio, fornire un'assistenza tecnologica ai vigili del fuoco spesso costretti ad entrare in edifici pericolanti o in fiamme.

In letteratura, solitamente, quando si parla di localizzazione si definiscono

due tipologie di dispositivi:

- *nodi ancora*, sensori la cui posizione è nota a priori e tramite i quali vengono collezionate le misure;
- *nodi target*, i nodi le cui posizioni non sono note, oggetto della stima.

Il nodo target può essere attivo se dotato di un'unità di trasmissione per inviare dei segnali di probe verso i nodi ancora e in questo caso verrà chiamato anche *emettitore*, caso preso in esame in questa tesi. Al contrario, se sono i nodi ancora ad inviare i segnali di probe, il nodo target è considerato passivo e si usano tecniche di localizzazione in backscatter per stimarne la posizione. In questa tesi, si ipotizza di utilizzare sensori a bordo di dispositivi mobili come gli *Unmanned Aerial Vehicle* (UAV) per una serie di vantaggi, alcuni dei quali riportati in Sezione 1.5. Per questo motivo, la definizione di nodi ancora non può essere utilizzata perché può portare ad una visualizzazione non corretta del problema. Così, si è deciso di utilizzare la nomenclatura utilizzata in paper riguardanti algoritmi di flocking come [1] dove il sensore mobile viene definito *agente* o *nodo*.¹

Sfruttando le basi fornite da lavori precedenti su tematiche legate alla localizzazione di un target, lo scopo di questa tesi è quello di andare ad esplorare nel dettaglio gli elementi che influenzano l'accuratezza nella stima della posizione di un emettitore indoor. In particolare, attraverso una metrica legata alla *Fisher Information Matrix* (FIM) viene definito il *Position Error Bound* (PEB) che quantifica il minimo errore di posizionamento per lo stimatore ottimo e rappresenta un benchmark per il confronto con stimatori più pratici come, ad esempio, gli stimatori non-coerenti basati sull'energia del segnale ricevuto. Questa grandezza è strettamente dipendente dalla geometria degli UAV rispetto all'emettitore, per cui la formazione degli UAV risulta fondamentale ai fini della localizzazione del target. Come verrà brevemente accennato in Sezione 1.2-1.3, le tecnologie utilizzabili e le metodologie per effettuare la localizzazione sono molte. In questa tesi si è scelto di studiare un sistema di localizzazione basato su un'infrastruttura flessibile composta da UAV senza entrare nel dettaglio della tecnologia, focalizzando l'attenzione sulla geometria creata dagli UAV per ottenere l'accuratezza migliore. In letteratura si possono trovare degli articoli che descrivono diverse versioni di algoritmi di controllo per UAV, sia centralizzati che distribuiti. Soprattutto in quest'ultimo caso, tuttavia, non sono presenti lavori che tengano in considerazione la latenza della rete descritta dai sensori. Infatti, l'algoritmo di controllo distribuito deve tenere conto del raggio di copertura limitato per lo scambio di informazioni. Per questa ragione è sembrato interessante

¹Nel seguito del documento verranno utilizzate le parole nodo, drone e UAV come sinonimi, mentre il dispositivo da localizzare verrà nominato indistintamente con emettitore e target.

considerare anche reti di UAV che prevedano scambio di pacchetti multi-hop nel rispetto di limiti temporali stringenti.

Il contributo di questa tesi è quello di fornire le basi per la creazione di un algoritmo di controllo di droni tale da calcolare efficacemente la miglior formazione possibile, in presenza di ostacoli e regioni in cui il collegamento tra target e nodo sia in linea di vista o meno. Vengono effettuate delle simulazioni per tre modelli in cui: i) la qualità della misura non dipende dalla distanza e si ipotizza un collegamento sempre in linea di vista; ii) la qualità della misura dipende dalla distanza, dove la varianza dell'errore di misura dipende dalla distanza stessa, ipotizzando un collegamento in linea di vista; iii) la qualità della misura dipende dalla distanza considerando anche la possibilità di avere sensori oscurati da ostacoli, ovvero che sperimentano una condizione di *Non-Line of Sight* (NLOS) con il target. Vengono mostrati svariati risultati numerici per vari scenari di impiego per mostrare le differenze di prestazioni che si possono ottenere al variare di alcuni parametri. Nella maggior parte delle simulazioni vengono riportate le traiettorie seguite dai nodi per dare un'idea della geometria ottimale e vengono riportati gli andamenti dell'accuratezza di stima caratterizzata dal PEB. Sono effettuati degli studi della variazione di prestazione quando per la creazione della formazione viene utilizzata la posizione reale dell'emettitore e quella stimata. Per quest'ultimo caso è stata effettuata anche un'analisi di convergenza per studiare la dipendenza dalla distanza e del numero di droni. Infine, è stato implementato un algoritmo di navigazione distribuito per la localizzazione del target che non necessita di unità centrale ma in cui ogni UAV si comporta come tale per il calcolo della propria posizione ottimale. È stato fatto un confronto fra le prestazioni di algoritmo centralizzato e distribuito, al variare del raggio di copertura fornito agli UAV, al numero dei nodi e al numero di hop massimo per lo scambio di informazioni. Viene fatto uno studio legato alla convergenza dell'algoritmo distribuito basato su una nuova metrica che permette di evidenziare le caratteristiche richieste per ottimizzare il trade-off fra accurata localizzazione del target e velocità di convergenza.

1.2 Tecnologie per la localizzazione indoor

In questa sezione vengono riportate tre delle tecnologie più interessanti nell'ambito della localizzazione indoor/outdoor. Nella sezione seguente viene descritta la tecnologia *Ultra-Wideband* (UWB), attualmente la più utilizzata in questo ambito, mentre in Sezione 1.2.2 viene descritta la promettente tecnologia a onde millimetriche e massive antenna, recentemente proposta per applicazioni 5G. Infine si accenna brevemente la tecnologia *Near-Field Electromagnetic Ranging* (NFER) basata sullo sfasamento del campo elettrico e del campo magnetico in condizioni di campo vicino. La scelta di una tecnologia rispetto alle altre dipende principalmente dal tipo di applicazione.

1.2.1 Ultra-Wideband

Da quando il GPS è diventato largamente accessibile, anche per dispositivi elettronici di piccola dimensione, la localizzazione della posizione assoluta di un dispositivo ha creato la possibilità di molte applicazioni. Il punto debole della tecnologia GPS è la scarsa capacità di penetrare gli oggetti, rendendo la localizzazione indoor poco accurata con errori dell'ordine di metri [2–5]. Per sopperire a questa mancanza, sono stati effettuati molti studi in letteratura giungendo a buoni risultati sfruttando la “nuova” tecnologia UWB [6–10].

In realtà, la tecnologia UWB [11] venne scoperta da G. Marconi nel 1901, ma solo recentemente, grazie allo sviluppo dei microprocessori e alla velocità raggiunta di commutazione dei gate a semiconduttore negli ultimi anni, è stato possibile utilizzare la UWB per applicazioni commerciali.

I sistemi UWB sono caratterizzati da una bassissima potenza di trasmissione nell'ordine di $P_{t_{\text{UWB}}} = 500\mu\text{W}$ rispetto a $P_{t_{\text{Wi-Fi}}} = 1\text{W}$ della trasmissione Wi-Fi. Questa differenza nei livelli di potenza, permette ai segnali UWB di coesistere con altri tipologie di segnale nello stesso spettro di frequenze, in quanto gli apparati non UWB percepiranno il segnale generato come rumore di fondo.

Introducendo la quantità *Signal-Noise Ratio* (SNR), dalla formula di Shannon sulla capacità del canale in presenza di rumore gaussiano:

$$C = B \log_2(1 + \text{SNR}) \quad (1.1)$$

dove B è la larghezza di banda in Hz, a causa del basso valore di quest'ultimo in seguito alla piccola potenza di trasmissione, per ottenere una buona capacità di trasmissione è necessario utilizzare una banda molto grande. Infatti, i segnali UWB hanno banda assoluta $B \geq 500\text{MHz}$ o banda relativa $B_{\text{rel}} \geq 20\%$. La banda assoluta è ottenuta dalla differenza delle due frequenze nel punto di emissione -10dB :

$$B = f_h - f_l \quad (1.2)$$

dove si definiscono la frequenza maggiore f_h e la frequenza minore f_l . La banda relativa è data da:

$$D_{\text{rel}} = \frac{B}{f_c} \quad (1.3)$$

dove f_c è la frequenza centrale data dalla media aritmetica delle due precedenti.

Come accuratamente descritto in [7, 12] a causa della relazione inversa tra banda occupata e durata del segnale, i segnali UWB sono caratterizzati da una brevissima durata nel dominio dei tempi, dell'ordine dei nano-secondi. Questa peculiarità porta ad alcuni vantaggi che la rendono una tecnologia molto importante per applicazioni di localizzazione indoor. Infatti, si caratterizza per:

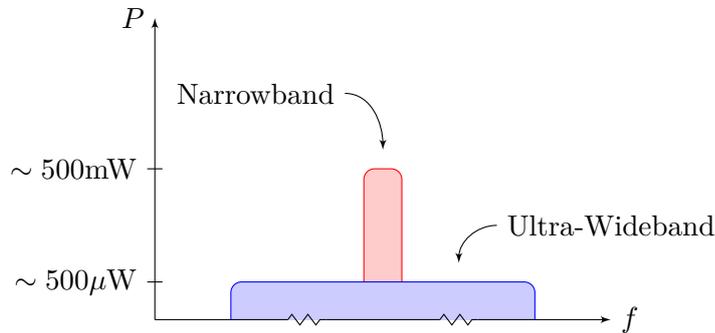


Figura 1.1: Comparazione segnali Narrowband e Ultra-Wideband.

- buona penetrazione degli ostacoli;
- maggior accuratezza nella stima della posizione, soprattutto indoor;
- alta velocità di trasmissione;
- basso costo e transceiver a bassa potenza.

La buona penetrazione degli ostacoli dei segnali UWB è data dal largo spettro occupato, che comprende componenti a bassa e ad alta frequenza. Inoltre, il grande spettro porta ad un'alta risoluzione temporale, che porta a buone performance di localizzazione e ad un'alta capacità di mitigazione del multipath.

1.2.2 Onde Millimetriche e Array di Antenne Massivi

La tecnologia ad onde millimetriche (*mm-Wave*) [13–15] è una tecnologia molto promettente per le future comunicazioni wireless e in rampa di lancio con la tecnologia 5G. Il motivo principale deriva dal fatto che le frequenze dei segnali utilizzati con questa tecnologia ricadono in una porzione dello spettro ancora non allocato, fuori dallo spazio saturo dai 700MHz ai 2.6GHz, quindi di grande interesse per i grandi provider di telefonia mobile. Si pensa che le onde millimetriche, la cui tecnologia è stata standardizzata in IEEE 802.11ad e IEEE 802.15.3c per 60GHz, possano portare ad ottenere comunicazioni wireless caratterizzate da un elevato data rate, bassa latenza e, a causa delle elevate frequenze che permettono di miniaturizzare le antenne che possono essere utilizzate in array massivi, una maggiore capacità di servizio. Questa tecnologia risulta essere promettente per i seguenti motivi principali [15, 16]:

- le onde millimetriche saranno utilizzate da molti dispositivi mobili 5G e se questi fossero in grado di localizzare un target non sarebbe necessario creare un'infrastruttura dedicata unicamente alla localizzazione ma sarebbe possibile sfruttare l'infrastruttura della rete telefonica. Le

Physical layer	Larghezza di banda	Risoluzione approssimativa
IEEE 802.11a/g	20MHz	15m
IEEE 802.11n	40MHz	7.5m
IEEE 802.11ac	< 160MHz	> 1.9m
UWB	> 500MHz	< 60cm
IEEE 802.11ad	> 2GHz	< 15cm

Tabella 1.1: Risoluzione approssimativa di localizzazione ottenibile per alcune tecnologie standardizzate derivanti [15].

onde millimetriche sono caratterizzate da una bassissima lunghezza d'onda quindi permettono di ridurre le dimensioni delle antenne alle *Base Station* (BS) che potrebbero essere quindi usate come singoli nodi;

- ii) a causa delle ridotte dimensioni delle BS che potrebbero essere in numero molto più elevato aumentando la densità, sarebbe più accurato tracciare il target andando ad analizzare a quale stazione base è collegato;
- iii) le onde millimetriche permettono di utilizzare delle antenne massive che presentano un'accuratezza nella stima *Angle of Arrival* (AOA) molto elevata;
- iv) l'elevata larghezza di banda porta un miglioramento nella stima del tempo di arrivo, rendendo più accurate le misure che legano la distanza tra due nodi ai parametri temporali come il *Time of Arrival* (TOA).

Come descritto dagli autori di [15, Tabella 1], riportata per comodità in Tabella 1.1, utilizzando la formula:

$$\lambda = \frac{c}{B} \quad (1.4)$$

è possibile fare una prima stima della precisione in metri λ raggiungibile per una tecnologia fissata la banda occupata B , dove c è la velocità della luce. Gli autori hanno utilizzato lo standard IEEE 802.11ad dove si utilizzano onde millimetriche di 60GHz e canale caratterizzato da 2.16GHz di banda per mostrare che una prima approssimazione della precisione raggiungibile è $\lambda \simeq 15\text{cm}$, anche meglio della UWB caratterizzata da banda $B = 500\text{MHz}$.

Gli autori di [14] hanno effettuato una campagna di misure nelle città di New York e Austin analizzando anche parametri come la penetrazione e riflessione del segnale trasmesso utilizzando le costruzioni presenti nelle città. Dai risultati di [14, Tabella 3], si evince una forte perdita di energia del

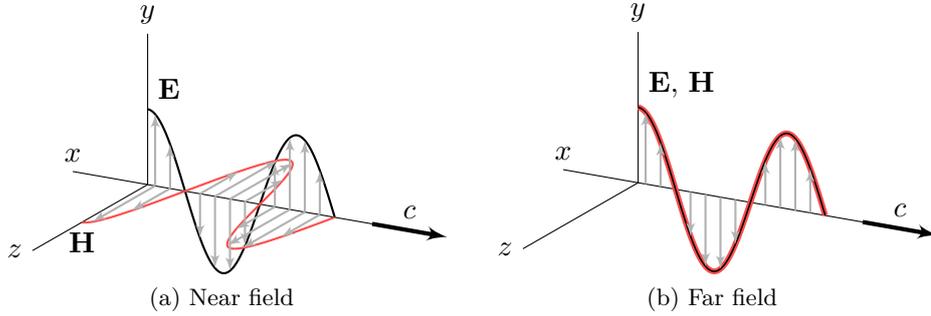


Figura 1.2: Rappresentazione delle onde dei campi \mathbf{E} , \mathbf{H} in condizioni di near e far field

segnale nella penetrazione degli edifici rendendo complessa la comunicazione outdoor-indoor. Gli autori spiegano, quindi, che il problema potrebbe essere arginato installando dei ripetitori all'interno degli edifici per propagare i segnali basati su onde millimetriche.

1.2.3 Near-Field Electromagnetic Ranging

Un'altra interessante tecnologia per la localizzazione indoor è il *Near-Field Electromagnetic Ranging* (NFER). Questa tecnologia si basa sulla struttura dell'onda elettromagnetica data dalla sovrapposizione di un'onda elettrica e una magnetica. È dimostrato in letteratura che quando ci si trova a distanza ravvicinata (*near field*) da una antenna (elettricamente) piccola, l'onda elettrica risulta sfasata di $\pi/2$ rispetto alla magnetica per poi tendere alla stessa fase man mano che la distanza aumenta fino ad arrivare in una situazione di *far field* dove le onde sono perfettamente in fase, come descritto graficamente in Figura 1.2.

Per spiegare il fenomeno di transizione di fase, è utile pensare l'antenna come un dipolo Hertziano composto da due cariche uguali ma di segno opposto, situate a distanza l [17–19] che forniscono un momento di dipolo dato da:

$$\mathbf{p} = p_0 T(t) \hat{\mathbf{z}} \quad (1.5)$$

dove la dipendenza dal tempo è data da $T(t)$. Il modulo del dipolo è dato da $p_0 = Ql$, con Q che rappresenta il valore assoluto della carica del dipolo. Utilizzando un sistema di riferimento polare a coordinate sferiche con vettori $(\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$ legate alle coordinate cartesiane da:

$$\begin{aligned} x &= r \cos \phi \sin \theta \\ y &= r \cos \phi \cos \theta \\ z &= r \sin \phi \end{aligned}$$

il campo elettrico di un dipolo Hertziano è dato da [17, 18, 20]:

$$\mathbf{E}(t) = \frac{p_0}{4\pi\epsilon_0 r^2} \left(\frac{T(t)}{r} + \frac{\dot{T}(t)}{c} \right) (2 \cos \theta \hat{\mathbf{r}} + \sin \theta \hat{\boldsymbol{\theta}}) + \frac{p_0 \ddot{T}(t) \sin \theta}{4\pi\epsilon_0 c^2 r} \hat{\boldsymbol{\theta}} \quad (1.6)$$

mentre quello magnetico è espresso da:

$$\mathbf{H}(t) = \frac{p_0}{4\pi r} \left(\frac{\dot{T}(t)}{r} + \frac{\ddot{T}(t)}{c} \right) \sin \theta \hat{\boldsymbol{\theta}}. \quad (1.7)$$

Attraverso l'elaborazione di queste equazioni, ponendo a zero i campi e valutando il comportamento dello zero-crossing, per un dipolo armonico con $T(t) = \omega \sin(\omega t)$, si ottengono le fasi per campo elettrico e magnetico rispettivamente:

$$\phi_{\mathbf{E}}(r) = -\frac{180}{\pi} \left[\frac{\omega r}{c} + \cot^{-1} \left(\frac{\omega r}{c} - \frac{c}{r\omega} \right) + n\pi \right] \quad (1.8)$$

$$\phi_{\mathbf{H}}(r) = -\frac{180}{\pi} \left[\frac{\omega r}{c} + \cot^{-1} \left(\frac{\omega r}{c} \right) + n\pi \right] \quad (1.9)$$

da cui è immediato ottenere la differenza di fase:

$$\phi_d(r) = \phi_{\mathbf{E}}(r) - \phi_{\mathbf{H}}(r) = -\frac{180}{\pi} \left[\cot^{-1} \left(\frac{\omega r}{c} - \frac{c}{r\omega} \right) - \cot^{-1} \left(\frac{\omega r}{c} \right) \right]. \quad (1.10)$$

La differenza di fase $\phi_d(r)$ è funzione della distanza r dal dipolo e si ha:

$$\phi_d(r) = \begin{cases} \frac{\pi}{2}, & r = 0 \\ 0, & r \rightarrow \infty. \end{cases}$$

Quindi, a causa della transizione da $\pi/2$ a 0 della differenza di fase tra $\mathbf{E}(t)$ e $\mathbf{H}(t)$, è possibile ottenere una buona precisione nella stima di ranging quando ci si trova in condizioni di campo vicino, al di sotto di metà della lunghezza d'onda λ . Per distanze superiori è necessario utilizzare strumenti più precisi per misurare il valore di $\phi_d(t)$, anche se per basse frequenze la distanza coperta tale che sia $r < \lambda/2$ è sufficientemente ampia.

A differenza delle altre tecnologie, il NFER permette di ottenere una precisione nell'ordine della decina dei centimetri a distanze fino a 70m utilizzando segnali narrowband e non necessitando di un preciso tempo di sincronizzazione. Operando a bassa frequenza, inoltre, la costruzione dei ricevitori può essere effettuata utilizzando componenti disponibili al grande commercio, a basso costo [19, 21].



Figura 1.3: Schema a blocchi degli stimatori di posizione.

1.3 Metodi per la localizzazione

La stima della posizione può essere fatta in modo diretto dal segnale ricevuto dal nodo (*direct positioning*) o in due passi (*two-step positioning*) dove il sistema prima stima dei parametri a partire dal segnale ricevuto e solo successivamente, a partire dalle stime, va a stimare la posizione come rappresentato da Figura 1.3 [22]. Quest'ultimo caso è in generale sub-ottimo, ma è molto meno complesso del *direct positioning* e per condizioni sufficientemente favorevoli presenta prestazioni simili. Per questo in pratica molti dei paper analizzati utilizzano questa tecnica di stima. Nel primo stadio della stima, i parametri stimati possono essere: la *Received Signal Strength* (RSS), AOA, TOA e/o *Time Difference of Arrival* (TDOA).

Questi parametri possono essere stimati con misurazioni *passive*, quando il segnale è generato in una posizione sconosciuta ad un tempo sconosciuto e ricevuto dai sensori senza che rispondano (*one-way*). Esistono anche tecniche con misurazioni *attive* dove si hanno segnali che viaggiano in entrambe le direzioni tra emettitore e sensori. Lo scambio di segnali può essere effettuato attraverso una risposta, come nel caso delle comunicazioni, ma anche tramite riflessioni come ad esempio nel caso di radar [23].

Un primo esempio di parametro stimabile viene descritto basandosi sul fatto che l'energia del segnale emesso dall'emettitore cambia con la distanza. Allora, la RSS è in grado di portare al nodo di sensing un'informazione sulla distanza tra il sensore e il target. Un fattore che incide sull'energia del segnale è il *path-loss*, che rappresenta la riduzione della potenza o energia del segnale che si propaga nello spazio. Un modello comune è dato da:

$$\bar{P}(d) = P_0 - 10\alpha \log_{10} \left(\frac{d}{d_0} \right) \quad (1.11)$$

dove $\bar{P}(d)$ è la potenza media a distanza d , α è l'esponente path-loss, P_0 è la potenza ricevuta a breve distanza d_0 . In ambienti wireless realistici, a causa dei meccanismi di propagazione come ad esempio riflessione e scattering, la

relazione tra distanza e potenza è molto complessa e questo porta ad una notevole fluttuazione della RSS, complicando molto un eventuale utilizzo di questo parametro per la stima di posizione. Per valutare le performance, si utilizza come metrica il *Cramér-Rao Lower Bound* (CRLB) che rappresenta la varianza dell'errore di stima unbiased e può essere usato come benchmark nel confronto con altri stimatori più pratici. Nel caso di misure RSS, il CRLB è dato da [24]:

$$\sigma_{\hat{d}} \geq \frac{\ln 10 \sigma_{sh} d}{10n} \quad (1.12)$$

dove \hat{d} rappresenta la stima unbiased della reale distanza d , σ_{sh} è la deviazione standard dello shadowing. Dalla (1.12) si nota come l'accuratezza del ranging decresce all'aumentare della distanza. Infine, il parametro RSS dipende fortemente dai parametri del canale, rendendo poco accurata la stima della posizione non appena le condizioni ambientali variano.

Un altro parametro collegato alla posizione è l'angolo creato tra emettitore e ricevitore. Comunemente, su un singolo nodo viene installato un array di antenne in modo da stimare l'AOA dal segnale ricevuto. L'idea su cui si basa questa stima di parametro è la differenza temporale con cui le antenne ricevono il segnale entrante, creando la possibilità di calcolare l'informazione sull'angolo di arrivo. Il CRLB per la varianza di uno stimatore AOA unbiased con una configurazione di N_a antenne *Uniform Linear Array* (ULA), ottenuta in [25], vale:

$$\sigma_{\hat{\theta}} \geq \frac{\sqrt{3}c}{\pi \beta l \cos \theta \sqrt{2N_a(N_a^2 - 1)\text{SNR}}} \quad (1.13)$$

dove θ è l'AOA, $\hat{\theta}$ è la sua stima, c la velocità della luce, SNR è il rapporto segnale rumore di ciascuna antenna che in questo caso è supposto uguale per tutte, l è lo spazio tra gli elementi dell'array, N_a è il numero di antenne e β è la banda effettiva. È importante notare che, a differenza della stima tramite RSS, l'accuratezza dello stimatore AOA cresce linearmente con la banda effettiva β : questo significa che la tecnologia Ultra-Wideband può facilitare una stima ad alta precisione. Al denominatore compare anche una dipendenza da N_a che permette di rendere più accurata la stima all'aumentare del numero di antenne, da cui si capisce l'importanza di array di antenne massive, fino a rendere impossibile la stima basata sulla direzione di arrivo per $N_a = 1$.

Oltre all'energia residua del segnale e all'angolo di arrivo, è possibile avere un'indicazione sulla posizione della sorgente del segnale andando ad analizzare il TOA. Per utilizzare il tempo di arrivo del segnale tra due nodi è necessario che questi siano ben allineati a livello di clock. Lo stimatore convenzionale TOA è caratterizzato da un correlatore o da un filtro adattato che dal segnale ricevuto:

$$r(t) = s(t - \tau) + n(t) \quad (1.14)$$

deve stimare il parametro τ con segnale sporcato da rumore gaussiano $n(t)$. Il problema è che nella pratica i segnali arrivano al ricevitore attraverso svariati percorsi a causa delle riflessioni. Per ottenere un'accurata stima di τ in un ambiente multipath, è necessario effettuare una detection del percorso diretto tramite degli algoritmi, che cercano di selezionare il primo segnale ricevuto invece che quello con intensità maggiore. Il CRLB per il segnale modellato come in (1.14) vale [6]:

$$\sigma_{\hat{\tau}} \geq \frac{1}{2\pi\beta\sqrt{2\text{SNR}}} \quad (1.15)$$

dalla quale è possibile notare come l'accuratezza di $\hat{\tau}$ aumenti con il rapporto segnale rumore e con la banda effettiva β .

Un altro parametro correlato alla posizione è la differenza dei tempi di arrivo TOA tra il nodo emettitore e due sensori che prende il nome di TDOA. Per effettuare questo tipo di misura è conveniente avere i nodi sincronizzati: in questo modo infatti non si avranno errori di offset e per effettuare la misura è sufficiente prendere i TOA dei due sensori. Il CRLB è derivabile dalla (1.15) e il miglioramento dipende dagli stessi parametri di cui sopra.

Ricapitolando, l'alta risoluzione temporale delle tecnologie UWB e *mm-waves* facilitano la stima dei parametri TOA, TDOA e AOA mentre il parametro RSS porta una stima molto grossolana della posizione. Un lato negativo della stima dell'angolo d'arrivo del segnale, però, è dato dalla necessità di utilizzare un array di antenne, complicando il ricevitore. Per queste ragioni, in letteratura si possono trovare molti lavori di positioning che sfruttano la stima TOA o TDOA. Nella tesi vengono analizzate delle misurazioni descritte dalla (2.2), dove il modello d'osservazione è legato a misure di distanza, proporzionali al TOA.

1.4 Filtri Bayesiani

A prescindere dalle tecnologie di misurazione scelte, gli algoritmi utilizzati per sfruttare le informazioni acquisite dai segnali si dividono in centralizzati e distribuiti (o decentralizzati). La differenza tra le due categorie è legata alla presenza di un'unità centrale di elaborazione nel caso centralizzato mentre non è presente in un algoritmo decentralizzato dove tutte le unità coinvolte si occupano di effettuare un'elaborazione.

Nel caso in cui si scelga di utilizzare un approccio decentralizzato [26], gli algoritmi sono incentrati su filtri Bayesiani, varianti del *Kalman Filter* (KF) e il più versatile *Particle Filter* (PF). Questi strumenti statistici permettono di giungere ad un metodo generale per sfruttare misure eterogenee, rumore e modelli di mobilità dell'utente. In [26], gli autori si concentrano sull'utilizzo della teoria Bayesiana che permette di modellare l'incertezza del sistema e il parametro di interesse combinando in modo ottimale la conoscenza a priori

e le informazioni ottenute dalle misurazioni. Chiamando $\mathbf{x}^{(k)}$ il vettore di stato e $\mathbf{y}^{(k)}$ il vettore contenente le misurazioni al tempo k , si vuole calcolare la probabilità avere l'intera sequenza di stati dall'istante iniziale all'istante k -esimo a partire dalle informazioni ottenute dalle misurazioni:

$$p(\mathbf{x}^{(0:k)}|\mathbf{y}^{(1:k)}) = \frac{p(\mathbf{y}^{(1:k)}|\mathbf{x}^{(0:k)})p(\mathbf{x}^{(0:k)})}{p(\mathbf{y}^{(1:k)})} \quad (1.16)$$

dove $\mathbf{x}^{(0:k)}$ rappresenta la sequenza di vettori di stato dall'istante iniziale all'istante k -esimo, $\mathbf{y}^{(1:k)}$ è la successione di tutte le misurazioni, $p(\mathbf{x}^{(0:k)})$ rappresenta tutta la conoscenza a priori, $p(\mathbf{y}^{(1:k)}|\mathbf{x}^{(0:k)})$ è il modello di percezione statistica e $p(\mathbf{y}^{(1:k)})$ è una costante di normalizzazione. Il problema nell'applicazione di questa equazione è la complessità computazionale che diventa ingestibile al crescere di k . Un'utile semplificazione per ridurre il costo computazionale, ampiamente utilizzata in letteratura, consiste nell'ipotizzare che lo stato $\mathbf{x}^{(0:k)}$ sia nella forma della sequenza di Markov del primo ordine così che diventi valida la relazione:

$$p(\mathbf{x}^{(k)}|\mathbf{x}^{(1:k-1)}, \mathbf{y}^{(1:k-1)}) = p(\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)}) \quad (1.17)$$

dalla quale si evince che lo stato all'iterazione k dipenda unicamente dallo stato $k-1$. Viene anche introdotta l'ipotesi di indipendenza della misura $\mathbf{y}^{(k)}$ dalle misure e stati precedenti, ottenendo:

$$p(\mathbf{y}^{(k)}|\mathbf{x}^{(1:k)}, \mathbf{y}^{(1:k-1)}) = p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}) \quad (1.18)$$

Queste assunzioni portano allo spazio degli stati secondo il modello Markoviano dove:

- $p(\mathbf{x}^{(0)})$ è l'informazione a priori allo step $k=0$;
- $p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)})$ è il modello che lega lo stato alla misura;
- $p(\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)})$ è la dinamica del modello.

Spesso, la descrizione dello spazio degli stati è data dal modello non lineare:

$$\mathbf{x}^{(k)} = \mathbf{g}(\mathbf{x}^{(k-1)}) + \mathbf{w}^{(k)} \quad (1.19)$$

$$\mathbf{y}^{(k)} = \mathbf{h}(\mathbf{x}^{(k)}) + \mathbf{v}^{(k)} \quad (1.20)$$

dove la funzione $\mathbf{g}(\cdot)$ modella la dinamica, la funzione $\mathbf{h}(\cdot)$ lega lo stato alla misura e $\mathbf{w}^{(k)}$ e $\mathbf{v}^{(k)}$ rappresentano rispettivamente rumore di processo e rumore di misura.

Per risolvere il problema di localizzazione, è sufficiente andare a calcolare la distribuzione $p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k)})$ che rappresenta la probabilità di avere lo stato $\mathbf{x}^{(k)}$ date le misure precedenti. Per il calcolo di questa quantità è richiesto un numero di iterazioni costante ed è basato sulla formulazione di filtro Bayesiano:

- i) *Inizializzazione*: allo step $k = 0$ viene associato a $p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k)})$ l'informazione a priori $p(\mathbf{x}^{(0)})$;
- ii) *Predizione*: sfruttando il modello della mobilità, alla k -esima iterazione si calcola:

$$p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k-1)}) = \int p(\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)}) p(\mathbf{x}^{(k-1)}|\mathbf{y}^{(1:k-1)}) d\mathbf{x}^{(k-1)}; \quad (1.21)$$

- iii) *Update*: si calcola lo stato a partire dalla misura $\mathbf{y}^{(k)}$:

$$p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k)}) = \frac{p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}) p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k-1)})}{\int p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}) p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k-1)}) d\mathbf{x}^{(k)}}. \quad (1.22)$$

Il problema è che il calcolo di queste ultime due equazioni è molto complesso salvo casi estremamente favorevoli nei quali i modelli diventano lineari. Negli altri casi si deve virare su approcci approssimati. Esistono tre metodi principali:

- i) *Extended Kalman Filter* (EKF), basato sulla linearizzazione delle funzioni non lineari del modello e viene assunto rumore Gaussiano;
- ii) *Unscented Kalman Filter* (UKF), nel quale viene eliminata la linearizzazione e si utilizza un approccio deterministico di campionamento per parametrizzare il valore medio e la covarianza del vettore di stato;
- iii) *Particle Filter* (PF), permette di eliminare il problema della non linearità e rumore non Gaussiano andando ad approssimare le funzioni densità di probabilità tramite un campionamento basato sulla pseudo-funzione delta di Dirac $\delta(\cdot)$:

$$p(\mathbf{x}^{(k)}|\mathbf{y}^{(1:k)}) \approx \sum_{m=1}^M w_m^{(k)} \delta(\mathbf{x}_m^{(k)} - \mathbf{x}^{(k)}) \quad (1.23)$$

così da trasformare gli integrali in sommatorie.

Una volta che la distribuzione marginale a posteriori dello stato corrente è stata calcolata, è possibile utilizzare dei criteri per effettuare la stima

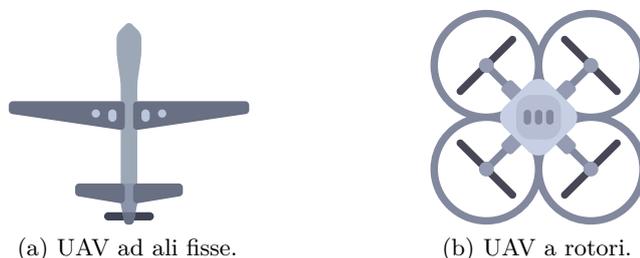


Figura 1.4: Classificazione degli UAV.

dello stato. I più largamente utilizzati sono il *Minimum Mean Square Error* (MMSE) e il *Maximum A Posteriori* (MAP), con quest'ultimo che sotto certe ipotesi si riduce nel semplice *Maximum Likelihood* (ML). Ad esempio, in [27] viene utilizzato un approccio distribuito in cui viene utilizzato il metodo Extended Kalman Filter. Invece, in [28] vengono descritti dei modelli distribuiti per la rappresentazione e lo studio di sistemi di localizzazione multi-target.

Un approccio differente da quello appena descritto è un algoritmo centralizzato. Supponendo di avere un'unità di elaborazione centrale verso la quale vengono convogliate tutte le misure ottenute dai nodi di sensing, essa è in grado di elaborare congiuntamente i dati per estrapolare le informazioni. In questo modo è possibile coordinare semplicemente un gruppo di sensori ottenendo una visione d'insieme di tutto il sistema. Questo approccio comporta l'utilizzo di sensori molto semplici che possono essere dotati semplicemente di schede di comunicazioni e con dei processori piccoli ed economici in quanto la richiesta computazionale non è impegnativa. Inoltre, questo approccio permette di semplificare l'algoritmo. I dettagli dell'implementazione dell'algoritmo nelle versioni centralizzata e distribuita sono stati concentrati in Capitolo 3.

1.5 Wireless Sensor Networks con UAV

Ipotizzando di avere un emettitore dinamico che deve essere localizzato, per mantenere una buona accuratezza, come si vedrà nel dettaglio nei prossimi capitoli, è necessario mantenere delle formazioni particolari ed è quindi necessario che i sensori siano in grado di muoversi. Una soluzione è quella di utilizzare gli UAV, anche conosciuti come droni, che grazie alla loro alta mobilità e basso costo, stanno trovando un vasto impiego nelle applicazioni moderne. Inoltre, grazie alla miniaturizzazione della sensoristica, è possibile trovare in commercio dei droni dotati di una buona quantità di sensori in grado di effettuare molte misurazioni.

Sin dalla sua prima apparizione certificata nel 1848, quando gli austriaci utilizzarono 200 palloni aerostatici carichi di esplosivo per bombardare una Venezia in rivolta contro l'impero Austro-Ungarico [29], gli UAV hanno storicamente trovato impiego prevalentemente in ambito militare [30] ma con la continua diminuzione del costo di produzione e della miniaturizzazione, al giorno d'oggi sono accessibili al pubblico.

Gli Unmanned Aerial Vehicle possono essere classificati essenzialmente in due categorie [31]: i) ad ali fisse (*fixed wings*), ii) ad ali rotanti (*rotary wings*). La prima tipologia di droni, raffigurata in Figura 1.4a, è caratterizzata da una elevata velocità ma richiede un costante movimento per mantenere quota -come avviene per gli aerei- e, per questo, non sono utilizzabili in applicazioni stazionarie. A causa della loro struttura aerodinamica, i droni ad ala fissa riescono a coprire lunghe distanze con un dispendio energetico relativamente basso grazie alla portanza generata dalla differenza di pressione che si presenta tra il ventre e il dorso del profilo alare. Al contrario, gli UAV ad ali rotanti utilizzano una tecnologia simile a quella degli elicotteri, ma, invece di avere un rotore principale per il sostentamento e l'avanzamento ed un rotore di coda per il contrasto al movimento rotatorio, i droni utilizzano coppie di rotori controrotanti per ottenere stabilità senza necessitare del rotore di coda. La configurazione più diffusa di drone ad ali rotanti è a quattro rotori, denominato quadricottero e raffigurato in Figura 1.4b, ma sono possibili anche configurazioni a più rotori. Gli rotary wings UAV sono caratterizzati da un decollo e atterraggio verticali e sono in grado di muoversi istantaneamente in tutte le direzioni ma anche restare fermi in un punto mantenendo la quota (*hovering*). Queste caratteristiche non sono ottenibili con i droni ad ala fissa, per cui la categoria ad ali rotanti è quella che più si adatta ad applicazioni di localizzazione. Infatti, i vantaggi principali ottenuti passando da una rete di sensori statica ad una composta da UAV ad ali rotanti sono la possibilità di formare una network flessibile definita e ri-configurabile in base alla qualità delle misure raccolte e la maggiore facilità nel creare collegamenti *Line of Sight* (LOS). Questo innovativo tipo di reti vengono definite *Flying Ad-Hoc Network* (FANET).

L'accesso a Unmanned Aerial Vehicle ha creato i presupposti per lo sviluppo di moltissime applicazioni. Un'applicazione molto calda riguarda l'utilizzo di droni per il supporto all'infrastruttura cellulare. Infatti, la comunicazione wireless supportata da UAV potrebbe permettere di fornire la connessione a dispositivi senza infrastruttura a causa di fattori ambientali proibitivi, in caso di calamità in cui l'infrastruttura presente venisse meno o in caso di infrastruttura in manutenzione. Alcune delle motivazioni principali della forza di questo approccio è la velocità di utilizzo con cui possono essere attivati e portati in una posizione consona, inoltre la dimensione è tale da rendere possibile l'utilizzo anche in ambienti urbani molto complessi.

Tutti gli aeromobili sono soggetti a codici della navigazione emessi dagli enti nazionali preposti al controllo del volo e a questo obbligo sono sottoposti

anche i droni, definiti come mezzi aerei la cui guida non è effettuata da un pilota a bordo. A causa della velocità di diffusione di Unmanned Aerial Vehicle negli ultimi anni, in molti paesi le norme per il controllo del volo di questi dispositivi non sono ancora state emesse. Uno dei paesi più preparati alla regolamentazione del volo degli UAV è l'Italia, dove nel dicembre 2013 l'Ente Nazionale Aviazione Civile (ENAC) ha emesso il regolamento attualmente alla seconda edizione, terzo emendamento [32]. In questo documento vengono definiti i requisiti da soddisfare per l'impegno di *Sistemi Aeromobili a Pilotaggio Remoto* (SAPR), composto da drone e strumenti necessari per il suo controllo, nell'ottica di regolamentare la sperimentazione nell'ottica di sviluppare questa nuova tecnologia. Questo regolamento è uno step essenziale verso la possibilità di fornire servizi tramite UAV, ponendo i principali requisiti che devono essere soddisfatti per ottenere l'autorizzazione al volo.

Un problema in questo tipo di rete eterogenea [33] è dato dalla struttura acerba dei protocolli di comunicazione che sono introdotti in [31]. Ad esempio in [34] vengono utilizzati degli UAV con a bordo delle schede in grado di implementare la funzione di stazioni radio base o BS. In particolare, gli autori hanno effettuato uno studio in cui descrivono il loro algoritmo di posizionamento degli *UAV-Base Station* (UAV-BS) in grado di ottimizzare il trade-off di consumo energetico e utenti serviti. Invece, in [35], gli autori effettuano uno studio su un algoritmo in grado di posizionare gli UAV-BS in presenza di un modello di canale più realistico rispetto i lavori precedenti.

1.6 Algoritmi di information seeking

Nella costruzione di un algoritmo di ottimizzazione, si rende necessario l'utilizzo di una funzione costo da minimizzare che rappresenti lo stato del sistema della configurazione all'iterazione attuale. Considerando che si utilizzano con degli stimatori, una metrica da considerare è quella fornita dal Cramér-Rao Lower Bound derivabile dalla FIM che quantifica la quantità di informazione sul parametro d'interesse a partire da misure rumorose.

Associando la FIM al simbolo \mathbf{J} , in letteratura sono state definite molte funzioni costo scalari $\psi(\mathbf{J})$. Le più utilizzate in letteratura in paper sul posizionamento sono dettagliatamente descritte in [36] e sono riassunte in seguito:

- D-optimality o criterio del determinante:

$$\psi(\mathbf{J}) = -\ln \det(\mathbf{J})$$

che minimizza il volume di incertezza nell'ellissoide della stima ed è invariante rispetto al cambio di scala dei parametri e trasformazioni lineari dell'output;

- E-optimality, criterio che dipende dal valore del minimo autovalore:

$$\psi(\mathbf{J}) = \lambda_{max}(\mathbf{J}^{-1})$$

dove l'operatore $\lambda_{max}(\cdot)$ è il massimo autovalore della matrice e questo criterio minimizza la lunghezza del massimo asse dello stesso ellissoide;

- A-optimality nel quale viene studiata la traccia:

$$\psi(\mathbf{J}) = \text{tr}(\mathbf{J}^{-1})$$

il quale elimina la varianza media delle stime;

- Sensitivity criterion nel quale si utilizza la traccia della matrice di informazione:

$$\psi(\mathbf{J}) = -\text{tr}(\mathbf{J})$$

spesso usato per la sua semplicità.

Nel seguito della sezione vengono riportati i principali riferimenti utilizzati durante la stesura di questo lavoro. La sezione si divide in due parti: la prima riguarda i paper principali presenti in letteratura sull'argomento della localizzazione e sul posizionamento ottimale per il tracking di un emettitore mentre nella seconda parte vengono descritti alcuni dei paper riguardanti problemi simili, caratterizzati da una funzione costo differente per i diversi obiettivi da conseguire.

Uno dei primi lavori sullo studio della posizione ottimale dei sensori per la localizzazione è [37] dove l'autore si è focalizzato su come la geometria del posizionamento di sensori vada ad incidere sulla prestazione della localizzazione dell'emettitore. È introdotto e descritto il fattore di prestazione *Geometric Dilution of Precision* (GDOP) e vengono riportate delle contour maps che mostrano come la disposizione ottimale sia quella che vede i sensori disposti come i vertici del poligono con centro nel target.

Successivamente, sono stati pubblicati [27, 38, 39] dove vengono studiate le strategie di coordinamento e il piazzamento ottimale di una rete di sensori mobili per il tracking di un target. In particolare, viene utilizzata come funzione costo il determinante della Fisher Information Matrix (D-optimality) e vengono calcolate le posizioni ottimali nel quale devono essere collocati i sensori. In questo paper viene sfruttato un risultato noto in letteratura, ovvero che la formazione ottimale dei sensori è quella che circonda il target in modo circolare descrivendo il poligono ad area maggiore il cui centro sia il nodo da localizzare, come discusso anche in [37]. Sfruttando questo risultato, gli autori hanno creato un algoritmo che forza i sensori a disporsi su una circonferenza centrata sul target.

In [40] gli autori hanno calcolato rigorosamente le posizioni dei sensori utilizzando gli angoli che essi formano rispetto al target. Utilizzando il

determinante della FIM (D-optimality), che è inversamente proporzionale all'incertezza della stima della posizione del target, viene mostrato come la posizione ottima dei sensori non sia unicamente quella descritta in [37] ma c'è la possibilità di disporre i sensori su una semicirconferenza centrata sul target. Un altro contributo di questo articolo consiste nelle considerazioni sul posizionamento nel quale i sensori non hanno stesso raggio di copertura.

Gli stessi autori hanno studiato in [41] il caso in cui la stima del parametro fosse di tipo TDOA. Dalle simulazioni è emerso che nel caso in cui venga utilizzata la differenza di tempo di arrivo per la stima della posizione la configurazione è indipendente dal sensor-range. Gli autori concludono che esistono più configurazioni ottimali, una delle quali è quella più volte riportata nel paper dove i sensori sono disposti in modo da formare angoli uguali tra loro.

Nell'articolo [42] vengono fatte delle analisi riguardanti la geometria ottima di emettitore e sensori misurando la potenza del segnale ricevuto RSS. Viene dimostrato rigorosamente per $n = 2, 3$ nodi che la disposizione su una circonferenza non è sempre la formazione ottima. Inoltre, si analizza anche la simmetria della formazione dei nodi rispetto all'emettitore, con i sensori che possono essere disposti su un semipiano rispetto la vera posizione del target. Gli autori si soffermano sull'unicità di questo lavoro alla data della pubblicazione e tutt'ora non è semplice trovare algoritmi di posizionamento che si basano sulla stima RSS a causa dei problemi delle fluttuazioni descritte in precedenza.

In [43] vengono effettuati degli studi sul posizionamento di dispositivi in grado di muoversi sott'acqua come gli *Autonomous Underwater Vehicle* (AUV) per il mapping dell'habitat marino, ispezione tubature ed altre applicazioni. Il contributo principale di questo lavoro, nel quale viene utilizzato la funzione costo D-optimality, è quello di analizzare la formazione per due modelli: i) posizione dell'emettitore nota con una piccola incertezza, ii) varianza della misura gaussiana dipendente dalla distanza. Una disposizione ottimale dei sensori è desiderabile anche in applicazioni di esplorazione terrestre con una destinazione fissata [44]. A causa del limitato range dei sensori, essi devono essere disposti ad una distanza adeguata. Un approccio frequentemente utilizzato, in quanto si dimostra un metodo molto valido, è la creazione di algoritmo Flocking [45]. La formazione dello stormo deve tener conto anche della disposizione ottimale dei sensori per ottenere una buona stima del target.

1.7 Organizzazione del documento

Il documento è organizzato come segue. Nel Capitolo 2 vengono derivati analiticamente il Cramér-Rao Lower Bound, la Fisher Information Matrix e il Position Error Bound a partire da modelli che descrivono l'informazione tra distanza tra nodo i -esimo ed emettitore presente all'interno del segnale ricevuto. In Sezione 2.1 viene utilizzato un modello molto semplice in cui viene ipotizzato che l'incertezza della misura non sia legata alla distanza tra nodo ed emettitore; In Sezione 2.2, invece, si utilizza un modello più realistico nel quale la qualità dell'informazione ricevuta è legata alla distanza: al crescere della distanza aumenta l'incertezza a causa della minor qualità del segnale ricevuto e questo effetto viene modellato in un aumento della varianza del rumore di stima. In Sezione 2.3 viene inserito nel modello la possibilità della presenza di ostacoli posti nella linea diretta tra nodo ed emettitore, rendendo la comunicazione NLOS. Infine, in Sezione 2.4 si deriva il modello utilizzato per l'implementazione dell'algoritmo di navigazione distribuito, considerando la possibilità che gli UAV sfruttino posizioni non aggiornate a causa dei vincoli temporali richiesti dall'applicazione.

Nel Capitolo 3 viene discusso l'algoritmo di controllo sviluppato per creare il posizionamento ottimo dei droni. In Sezione 3.1 si effettua lo studio di ottimizzazione non vincolata per l'applicazione al modello analizzato in Sezione 2.1. In Sezione 3.2 viene studiato nel dettaglio come è stato progettato l'algoritmo per ottenere la formazione migliore nel rispetto dei vincoli tramite il metodo del gradiente proiettato basato su moltiplicatori di Lagrange. Nella Sezione 3.2.3 vengono analizzati i possibili vincoli che un UAV deve rispettare come il mantenimento di una distanza minima dall'emettitore, mantenimento di una distanza di sicurezza tra UAV e anticollisione con ostacoli in traiettoria. In Sezione 3.2.4 vengono calcolate le quantità da fornire al simulatore per l'ottimizzazione della formazione per la localizzazione quando il modello è dipendente dalla distanza. In conclusione di capitolo, in Sezione 3.3 si descrivono i principi utili all'implementazione di un algoritmo di navigazione distribuito. In particolare, si introducono i principi relativi alla teoria dei grafi per la creazione della rete di collegamenti tra UAV, fornendo qualche esempio grafico e se ne deriva l'equazione di ottimizzazione.

Nel Capitolo 4 vengono riportati e discussi i risultati numerici ottenuti al variare di molti parametri utilizzando l'algoritmo di controllo centralizzato. Il capitolo viene diviso in tre sezioni: Sezione 4.1 contiene i risultati ottenuti inserendo la reale posizione dell'emettitore all'interno della formula del Position Error Bound mentre da Sezione 4.2 vengono utilizzate delle posizioni stimate da uno stimatore Maximum Likelihood per avvicinare maggiormente la simulazione ad uno scenario reale. In Sezione 4.3 vengono effettuate delle analisi nel caso in cui il modello di canale sia di tipo Non-Line of Sight, con studio della prestazione in termini di UAV e a livello di convergenza dell'algoritmo al variare di alcuni parametri.

Nel Capitolo 5 si descrivono i risultati ottenuti nel caso di algoritmo di navigazione e posizionamento distribuito. In Sezione 5.1 viene riportata una simulazione per mostrare il comportamento degli *Unmanned Aerial Vehicle* (UAV) nella singola realizzazione. In Sezione 5.2 vengono mostrati i risultati ottenuti tramite simulazioni Monte Carlo per analizzare il comportamento medio dell'algoritmo. In particolare vengono analizzate i) le differenze di performance al variare del communication range, dove viene effettuato anche un paragone con l'algoritmo centralizzato; ii) le differenze di prestazione al variare del numero massimo di hops per verificare quanto possa incidere il protocollo di comunicazione sulla localizzazione; iii) dipendenza della prestazione dal numero di nodi nel caso distribuito.

Infine, nel Capitolo 6 vengono tratte le conclusioni basate sui risultati ottenuti nei Capitoli 4 e 5.

Capitolo 2

Limiti fondamentali di localizzazione con UAV network

PRIMA di costruire l'algoritmo di localizzazione è necessario definire lo scenario e ricercare i limiti fondamentali dell'applicazione. L'applicazione analizzata consiste nella localizzazione di un nodo emettitore che per ipotesi emette un segnale radio con caratteristiche di frequenza e potenza tali da non avere problemi ad attraversare eventuali muri. Un esempio è dato dalla tecnologia *Near-Field Electromagnetic Ranging* (NFER), brevemente introdotta in Sezione 1.2.3, che sfrutta segnali a frequenza molto bassa (circa 1MHz) e condizioni di campo vicino per determinare la distanza fra il nodo emettitore e ricevitore. Si immagina che, per qualche ragione, il target non possa utilizzare il segnale *Global Positioning System* (GPS), come avviene nel caso in cui esso si trovi all'interno di un edificio. L'emettitore deve essere localizzato da uno stormo di nodi mobili situato all'esterno dell'area, come ad esempio *Unmanned Aerial Vehicle* (UAV), ciascuno dei quali si suppone conosca la propria posizione assoluta tramite il GPS. Questi nodi formano un'infrastruttura dinamica per la localizzazione dell'emettitore, molto più flessibile e veloce di una struttura fissa basata su sensori statici. Per semplicità, si suppone di utilizzare una tecnologia in grado di fornire direttamente la distanza tra emettitore e nodo, alla quale viene sovrapposta un'incertezza legata all'errore di stima e alla eventuale presenza di ostacoli. I limiti fondamentali vengono derivati sia per essere utilizzati in un algoritmo centralizzato che in un algoritmo distribuito.

In Sezione 2.1 si formalizza il problema matematico e si derivano le grandezze fondamentali per impostare l'algoritmo di posizionamento degli UAV centralizzato nel caso in cui la distanza non incida sulla qualità del segnale ricevuto dai droni e che la varianza del rumore di stima sia costante, nell'ipotesi di *Additive White Gaussian Noise* (AWGN). In Sezione 2.2 si introduce un modello di rumore di stima della distanza tale per cui l'incertezza sulla

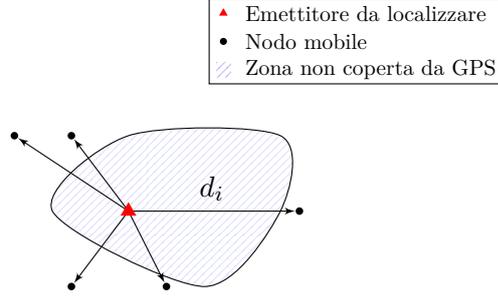


Figura 2.1: Rappresentazione grafica del caso di un emettitore in una zona in cui non è disponibile il GPS e UAV situati all'esterno della zona che devono stimare la sua posizione.

posizione del target aumenta al crescere della distanza tra emettitore e UAV per permettere all'algoritmo di simulare scenari più realistici, mantenendo l'ipotesi di un'unità di elaborazione centrale che calcola le posizioni ottimali degli UAV. In Sezione 2.3 si introduce il concetto di bias e si discutono le variazioni rispetto ai modelli precedenti, mentre in Sezione 2.4 si derivano i limiti fondamentali supponendo che siano gli UAV stessi ad effettuare l'elaborazione delle informazioni, utilizzando quindi un algoritmo distribuito.

2.1 Modello centralizzato con varianza dell'errore di stima costante

Dato un numero n di nodi mobili di cui sono note le posizioni, l'obiettivo è quello di localizzare un singolo emettitore, non coperto da GPS, situato in posizione non nota $\mathbf{p}_t^{(k)} = [x_t^{(k)}, y_t^{(k)}]^T$, dove k è lo slot temporale. Si ipotizza che la componente verticale della posizione sia nota: infatti i dispositivi mobili sono comandati, quindi sono in grado di andare all'altezza desiderata, mentre si suppone che il target nella regione non coperta da GPS sia dotato di un barometro in grado di fornire con notevole precisione la coordinata relativa all'altezza. Il problema affrontato, quindi, è 2D ma può essere facilmente esteso al caso 3D.

Si definisce con $\mathbf{z}^{(k)} = [z_1^{(k)}, \dots, z_i^{(k)}, \dots, z_n^{(k)}]^T$ il vettore contenente le misure disponibili collezionate dagli UAV all'istante k , fornito all'unità centrale:

$$\mathbf{z}^{(k)} = \mathbf{h}^{(k)}(\mathbf{p}_t^{(k)}) + \mathbf{v}^{(k)} \quad (2.1)$$

dove $\mathbf{h}^{(k)}(\mathbf{p}_t^{(k)})$ è la funzione che lega la posizione del target e l'osservazione mentre $\mathbf{v}^{(k)}$ è un vettore composto da variabili aleatorie Gaussiane a valor medio nullo e matrice covarianza $\mathbf{\Sigma} = \text{diag}(\dots, \sigma_i^2, \dots)$ che descrive il rumore di stima della distanza AWGN. Utilizzando un approccio centralizzato,

le misure degli UAV sono inviate a un'unità centrale che si occupa di stimare i segnali di controllo. Ipotizzando di non avere ritardi nella trasmissione, l'elaboratore centrale sarà a conoscenza delle posizioni dei nodi istante per istante. Quindi alla k -esima iterazione l'algoritmo utilizzerà unicamente informazioni aggiornate e per questa ragione si è scelto di non riportare l'apice relativo al fattore temporale delle informazioni durante la trattazione del caso centralizzato. Questa notazione verrà ripristinata quando si discuterà il caso distribuito, dove verrà introdotto il concetto di latenza e multi-hop.

Effettuando delle stime di ranging, la funzione $\mathbf{h}(\mathbf{p}_t)$ che lega la posizione del target e l'osservazione misurabile, per l' i -esimo generico UAV, sarà data dalla distanza dal target al quale si sovrappone una componente legata all'incertezza nella stima. Così, la generica misura z_i può essere espressa da:

$$z_i = \|\mathbf{p}_t - \mathbf{p}_i\|_2 + v_i = d_i(\mathbf{p}_t, \mathbf{p}_i) + v_i \quad (2.2)$$

dove $v_i \sim \mathcal{N}(v_i; 0, \sigma_i^2)$ con varianza costante $\sigma_i^2 = \sigma_0^2$, definendo σ_0 come la deviazione standard alla distanza di riferimento $d_0 = 1\text{m}$ mentre $d_i(\mathbf{p}_t, \mathbf{p}_i) = \|\mathbf{p}_t - \mathbf{p}_i\|_2$ rappresenta la distanza euclidea tra l' i -esimo nodo mobile e l'emettitore. Essendo v_i una variabile aleatoria, anche la stima della distanza z_i è aleatorio ed è caratterizzato da una densità di probabilità il cui valor medio è rappresentato dalla distanza $d_i(\mathbf{p}_t, \mathbf{p}_i)$ data da:

$$f_i(z_i|\mathbf{p}_t) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(z_i - d_i(\mathbf{p}_t, \mathbf{p}_i))^2}{2\sigma_i^2}}. \quad (2.3)$$

Ipotizzando che le misure dei n droni raccolte nel vettore \mathbf{z} siano indipendenti, è possibile esprimere la likelihood function:

$$\Lambda(\mathbf{z}|\mathbf{p}_t) = \prod_{i=1}^n f_i(z_i|\mathbf{p}_t). \quad (2.4)$$

Per valutare l'efficacia di uno stimatore di posizione si utilizza come benchmark il *Cramér-Rao Lower Bound* (CRLB) definito nella forma vettoriale da [46]:

$$\mathbb{E}_{\mathbf{z}} \left[(\mathbf{p}_t - \hat{\mathbf{p}}_t)(\mathbf{p}_t - \hat{\mathbf{p}}_t)^T \right] \succeq \mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p}) \quad (2.5)$$

dove \mathbf{p}_t è la posizione dell'emettitore che si vuole stimare mentre $\hat{\mathbf{p}}_t$ è la sua stima, \mathbf{p} è il vettore contenente tutte le posizioni dei nodi e $\mathbf{J}(\mathbf{p}_t; \mathbf{p})$ è la *Fisher Information Matrix* (FIM) data da:

$$\mathbf{J}(\mathbf{p}_t; \mathbf{p}) = \mathbb{E}_{\mathbf{z}} \left\{ [\nabla_{\mathbf{p}_t} \ln \Lambda(\mathbf{z}|\mathbf{p}_t)] [\nabla_{\mathbf{p}_t} \ln \Lambda(\mathbf{z}|\mathbf{p}_t)]^T \right\} \quad (2.6)$$

mentre l'operazione $\mathbf{A} \succeq \mathbf{B}$ per due generiche matrici $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ significa che $\mathbf{a}^T \mathbf{A} \mathbf{a} \geq \mathbf{a}^T \mathbf{B} \mathbf{a}$ per ogni vettore $\mathbf{a} \in \mathbb{R}^n$. Il CRLB fornisce il limite teorico sulla varianza dell'errore di stima di tutti gli stimatori unbiased, quindi un

utile strumento per verificare la bontà di stimatori pratici. Il primo termine all'interno del valore atteso della (2.6) può essere riformulato sfruttando (2.4) e la proprietà del prodotto di logaritmi $\ln(\prod_i x_i) = \sum_i \ln(x_i)$, ottenendo:

$$\begin{aligned}\nabla_{\mathbf{p}_t} \ln \Lambda(\mathbf{z}|\mathbf{p}_t) &= \nabla_{\mathbf{p}_t} \ln \left(\prod_{i=1}^n f_i(z_i|\mathbf{p}_t) \right) = \\ &= \nabla_{\mathbf{p}_t} \sum_{i=1}^n \ln(f_i(z_i|\mathbf{p}_t)) = \\ &= \sum_{i=1}^n \frac{1}{f_i(z_i|\mathbf{p}_t)} \nabla_{\mathbf{p}_t} f_i(z_i|\mathbf{p}_t)\end{aligned}\quad (2.7)$$

Applicando la (2.7) al caso bidimensionale in esame, si ottiene il primo termine della (2.6):

$$\nabla_{\mathbf{p}_t} \ln \Lambda(\mathbf{z}|\mathbf{p}_t) = \sum_{i=1}^n \frac{1}{f_i(z_i|\mathbf{p}_t)} \left[\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t}, \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \right] \quad (2.8)$$

e, effettuando la trasposta della matrice, è possibile calcolare anche il secondo termine:

$$[\nabla_{\mathbf{p}_t} \ln \Lambda(\mathbf{z}|\mathbf{p}_t)]^T = \sum_{j=1}^n \frac{1}{f_j(z_j|\mathbf{p}_t)} \begin{bmatrix} \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial x_t} \\ \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial y_t} \end{bmatrix}. \quad (2.9)$$

Moltiplicando i termini ottenuti nella (2.8) e (2.9), si ottiene la matrice di Fisher:

$$\begin{aligned}\mathbf{J}(\mathbf{p}_t; \mathbf{p}) &= \mathbb{E}_{\mathbf{z}} \left\{ \sum_{i=1}^n \sum_{j=1}^n \frac{1}{f_i(z_i|\mathbf{p}_t) f_j(z_j|\mathbf{p}_t)} \times \right. \\ &\quad \left. \times \begin{bmatrix} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial x_t} & \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial y_t} \\ \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial x_t} & \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \frac{\partial f_j(z_j|\mathbf{p}_t)}{\partial y_t} \end{bmatrix} \right\}. \quad (2.10)\end{aligned}$$

Sfruttando l'indipendenza delle misure è possibile dimostrare che gli unici termini che contribuiscono sono per $i = j$. Infatti:

$$\mathbb{E}_{\mathbf{z}} \left\{ \frac{1}{f_i f_j} \frac{\partial f_i}{\partial x_t} \frac{\partial f_j}{\partial x_t} \right\} = \mathbb{E}_{\mathbf{z}} \left\{ \frac{1}{f_i} \frac{\partial f_i}{\partial x_t} \right\} \mathbb{E}_{\mathbf{z}} \left\{ \frac{1}{f_j} \frac{\partial f_j}{\partial x_t} \right\} \quad (2.11)$$

vale solo nell'ipotesi di indipendenza¹ e applicando ad un termine l'operatore valore medio:

$$\mathbb{E}_{\mathbf{z}} \left\{ \frac{1}{f_i} \frac{\partial f_i}{\partial x_t} \right\} = \int_{-\infty}^{\infty} \frac{1}{f_i} \frac{\partial f_i}{\partial x_t} f_i dz_i = \int_{-\infty}^{\infty} \frac{\partial f_i}{\partial x_t} dz_i. \quad (2.12)$$

¹Per non appesantire troppo la notazione, per questa breve dimostrazione, si è scelto di omettere la dipendenza della d_i dalle posizioni dei nodi che vengono ora rappresentate con $d_i = d_i(\mathbf{p}_t, \mathbf{p}_i)$ e $d_j = d_j(\mathbf{p}_t, \mathbf{p}_j)$. Lo stesso viene fatto per $f_i = f_i(z_i|\mathbf{p}_t)$ e $f_j = f_j(z_j|\mathbf{p}_t)$.

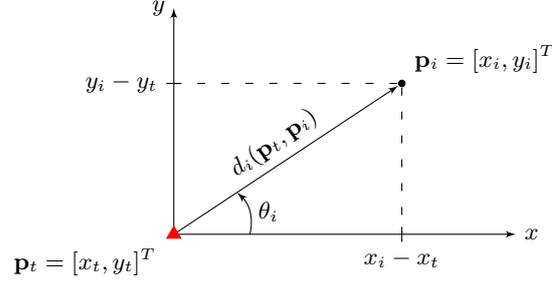


Figura 2.2: Analisi della geometria legata alle posizioni dei nodi.

Sotto ipotesi di regolarità della derivata parziale rispetto ad x_t ed y_t di f_i ed f_j , si possono scambiare gli operatori derivata e integrale:

$$\mathbb{E}_{\mathbf{z}} \left\{ \frac{1}{f_i} \frac{\partial f_i}{\partial x_t} \right\} = \frac{\partial}{\partial x_t} \int_{-\infty}^{\infty} f_i dz_i = 0 \quad (2.13)$$

dove per giungere al risultato finale si è sfruttato il secondo assioma della probabilità. La (2.13) mostra come gli unici valori non banali siano per $i = j$. Riscrivendo la (2.10), sfruttando la (2.13), si ottiene:

$$\mathbf{J}(\mathbf{p}_t; \mathbf{p}) = \mathbb{E}_{\mathbf{z}} \left\{ \sum_{i=1}^n \frac{1}{(f_i(z_i|\mathbf{p}_t))^2} \begin{bmatrix} \left(\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \right)^2 & \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \\ \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} & \left(\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \right)^2 \end{bmatrix} \right\} \quad (2.14)$$

ed è necessario calcolare le derivate al suo interno. Si sceglie di derivare unicamente rispetto a x_t perché, come sarà chiaro in seguito, si ottengono risultati analoghi anche rispetto ad y_t :

$$\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(z_i - \sqrt{\bar{x}^2 + \bar{y}^2})^2}{2\sigma_i^2}} \left(\frac{\sqrt{\bar{x}^2 + \bar{y}^2} - z_i}{\sigma_i^2} \cdot \frac{\bar{x}}{\sqrt{\bar{x}^2 + \bar{y}^2}} \right) \quad (2.15)$$

dove $\bar{x} = x_i - x_t$ e $\bar{y} = y_i - y_t$ sono utilizzate per compattare la notazione. L'ultimo termine tra parentesi corrisponde al coseno dell'angolo formato con l'asse orizzontale (o il seno dell'angolo formato con l'asse verticale), come si vede chiaramente da Figura 2.2. È quindi possibile riscrivere la (2.15) come:

$$\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(z_i - d_i(\mathbf{p}_t, \mathbf{p}_i))^2}{2\sigma_i^2}} \left(\frac{d_i(\mathbf{p}_t, \mathbf{p}_i) - z_i}{\sigma_i^2} \right) \cos \theta_i. \quad (2.16)$$

Va notato che nel caso di derivata rispetto ad y_t , al posto del coseno si ottiene il seno: per questo motivo si è ritenuto sufficiente mostrare il calcolo

della derivata solo rispetto ad x_t . Per comodità di notazione, introducendo la variabile $\xi_i = z_i - d_i(\mathbf{p}_t, \mathbf{p}_i)$, si definisce la funzione $g_i(\xi_i)$:

$$g_i(\xi_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{\xi_i^2}{2\sigma_i^2}} \left(\frac{-\xi_i}{\sigma_i^2} \right) \quad (2.17)$$

e si possono definire i valori delle derivate:

$$\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} = g_i(\xi_i) \cos \theta_i \quad (2.18)$$

$$\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} = g_i(\xi_i) \sin \theta_i. \quad (2.19)$$

Utilizzando queste ultime relazioni, la matrice di informazione di Fisher vale:

$$\mathbf{J}(\mathbf{p}_t; \mathbf{p}) = \mathbb{E}_{\mathbf{z}} \left\{ \sum_{i=1}^n \frac{g^2(\xi_i)}{(f_i(z_i|\mathbf{p}_t))^2} \begin{bmatrix} \cos^2 \theta_i & \cos \theta_i \sin \theta_i \\ \cos \theta_i \sin \theta_i & \sin^2 \theta_i \end{bmatrix} \right\} \quad (2.20)$$

ed è possibile definire la matrice che descrive la geometria dei nodi con $\mathbf{M}(\theta_i)$:

$$\mathbf{M}(\theta_i) = \begin{bmatrix} \cos^2 \theta_i & \cos \theta_i \sin \theta_i \\ \cos \theta_i \sin \theta_i & \sin^2 \theta_i \end{bmatrix}. \quad (2.21)$$

Osservando che $\mathbf{M}(\theta_i)$ non è aleatoria, il valore atteso si applica solo al primo termine della (2.20) definendo $A_i(\mathbf{p}_t, \mathbf{p}_i)$:

$$A_i(\mathbf{p}_t, \mathbf{p}_i) = \int_{-\infty}^{\infty} \frac{g^2(\xi_i)}{f_i(z_i|\mathbf{p}_t)} d\xi_i \quad (2.22)$$

che permette di riformulare la FIM in modo molto compatto:

$$\mathbf{J}(\mathbf{p}_t; \mathbf{p}) = \sum_{i=1}^n A_i(\mathbf{p}_t, \mathbf{p}_i) \mathbf{M}(\theta_i). \quad (2.23)$$

Sostituendo la (2.3) e la (2.17) nella (2.22) si ottiene un importante risultato:

$$A_i(\mathbf{p}_t, \mathbf{p}_i) = \frac{1}{\sqrt{2\pi}\sigma_i^5} \int_{-\infty}^{\infty} \xi_i^2 e^{-\frac{\xi_i^2}{2\sigma_i^2}} d\xi = \frac{1}{\sqrt{2\pi}\sigma_i^5} \sqrt{2\pi}\sigma_i^3 = \frac{1}{\sigma_i^2} \quad (2.24)$$

dove si vede come il valore sia costante, ricordando che, per ipotesi, $\sigma_i^2 = \sigma_0^2$. Calcolando l'inversa della matrice $\mathbf{J}(\mathbf{p}_t; \mathbf{p})$ si ottiene il CRLB²:

$$\mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p}) = \frac{1}{\det(\mathbf{J}(\mathbf{p}_t; \mathbf{p}))} \begin{bmatrix} \sum_{i=1}^n \frac{\sin^2 \theta_i}{\sigma_0^2} & -\sum_{i=1}^n \frac{\cos \theta_i \sin \theta_i}{\sigma_0^2} \\ -\sum_{i=1}^n \frac{\cos \theta_i \sin \theta_i}{\sigma_0^2} & \sum_{i=1}^n \frac{\cos^2 \theta_i}{\sigma_0^2} \end{bmatrix} \quad (2.25)$$

²Data la matrice $A = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ la sua inversa è data da $A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} D & -B \\ -C & A \end{pmatrix}$

dove:

$$\det(\mathbf{J}(\mathbf{p}_t; \mathbf{p})) = \frac{1}{\sigma_0^4} \left[\left(\sum_{i=1}^n \cos^2 \theta_i \right) \left(\sum_{i=1}^n \sin^2 \theta_i \right) - \left(\sum_{i=1}^n \cos \theta_i \sin \theta_i \right)^2 \right]$$

Introducendo la definizione di *Position Error Bound* (PEB) [46]:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\text{Tr}\{\mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p})\}} \quad (2.26)$$

si nota come sia necessario calcolare la traccia della matrice FIM inversa:

$$\begin{aligned} \text{Tr}\{\mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p})\} &= \frac{1}{\det(\mathbf{J}(\mathbf{p}_t; \mathbf{p}))\sigma_0^2} \left(\sum_{i=1}^n \cos^2 \theta_i + \sum_{i=1}^n \sin^2 \theta_i \right) = \\ &= \sigma_0^2 \frac{n}{\left(\sum_{i=1}^n \cos^2 \theta_i \right) \left(\sum_{i=1}^n \sin^2 \theta_i \right) - \left(\sum_{i=1}^n \cos \theta_i \sin \theta_i \right)^2} \end{aligned} \quad (2.27)$$

dalla quale è immediato calcolare il PEB:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sigma_0 \sqrt{\frac{n}{\sum_{i=1}^n \cos^2 \theta_i \sum_{i=1}^n \sin^2 \theta_i - \left(\sum_{i=1}^n \cos \theta_i \sin \theta_i \right)^2}}. \quad (2.28)$$

Ricordando che da considerazioni geometriche si ha:

$$c_i = \cos \theta_i = \frac{x_i - x_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)} \quad (2.29)$$

$$s_i = \sin \theta_i = \frac{y_i - y_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)} \quad (2.30)$$

è possibile riscrivere la (2.28) esplicitandone la dipendenza da \mathbf{p}_t e da \mathbf{p} :

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sigma_0 \sqrt{\frac{n}{\sum_{i=1}^n \left(\frac{x_i - x_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)} \right)^2 \sum_{i=1}^n \left(\frac{y_i - y_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)} \right)^2 - \left(\sum_{i=1}^n \frac{(x_i - x_t)(y_i - y_t)}{d_i^2(\mathbf{p}_t, \mathbf{p}_i)} \right)^2}}. \quad (2.31)$$

In letteratura, è frequente trovare la formulazione del PEB compatta ottenuta sostituendo i coefficienti geometrici c_i , s_i della (2.29) e della (2.30) che sostituiti nella (2.45) portano a:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sigma_0 \sqrt{\frac{n}{\left(\sum_i^n c_i^2 \right) \left(\sum_i^n s_i^2 \right) - \left(\sum_i^n c_i s_i \right)^2}}. \quad (2.32)$$

2.2 Modello centralizzato con varianza dell'errore di stima dipendente dalla distanza

Il caso analizzato in precedenza è una semplificazione utile per introdurre il problema e capire alcuni aspetti interessanti ma si distacca nettamente dalla realtà. Realisticamente, ci si aspetta che all'aumentare della distanza sia sempre più difficoltosa la stima della posizione dell'emettitore, ad esempio a causa dell'affievolimento della potenza del segnale ricevuto dagli UAV.

Per rendere più verosimile il modello, si introduce una dipendenza dalla distanza tra l'emettitore e l' i -esimo UAV della varianza dell'errore di stima, come in [47]:

$$\sigma_i^2(d_i(\mathbf{p}_t, \mathbf{p}_i)) = \sigma_0^2 d_i^\alpha(\mathbf{p}_t, \mathbf{p}_i) \quad (2.33)$$

dove α è l'esponente di path-loss che quantifica quanto rapidamente la varianza cresce con la distanza per la singola tratta del segnale e σ_0^2 è la varianza ad un metro; allora il rumore di stima sarà distribuito come:

$$\nu_i \sim \mathcal{N}(\nu_i; 0, \sigma_0^2 d_i^\alpha(\mathbf{p}_t, \mathbf{p}_i)). \quad (2.34)$$

Nonostante l'ampia presenza di questo modello in letteratura, per tecnologie recenti, come ad esempio la *Ultra-Wideband* (UWB), è molto conservativo: misurazioni effettuate mostrano come in linea di vista l'errore di localizzazione sia circa costante prima di crollare quando il rapporto segnale-rumore scende al di sotto di una soglia. Il modello utilizzato è ancora molto semplificato: ipotizzando di avere il nodo emettitore all'interno di un edificio, il segnale ricevuto da un UAV è attenuato e ritardato dai muri ed eventuali riflessioni. Questo non viene considerato in prima approssimazione, ma come descritto in [46] per avere una stima più verosimile a quella che si avrebbe in pratica è necessario considerare un bias positivo, quindi una stima abbondante della distanza. Una descrizione introduttiva del modello con bias è trattata in Sezione 4.3.

Analogamente a quanto fatto nel caso a varianza costante in Sezione 2.1, si formalizza il calcolo del PEB partendo dall'espressione della densità di probabilità:³

$$f_i(z_i|\mathbf{p}_t) = \frac{1}{\sqrt{2\pi}\sigma_i(d_i)} e^{-\frac{(z_i-d_i)^2}{2\sigma_i^2(d_i)}}. \quad (2.35)$$

³Nel seguito della trattazione, si utilizza una notazione che non esplicita la dipendenza della distanza dalle posizioni esprimendo $d_i = d_i(\mathbf{p}_t, \mathbf{p}_i)$ per rendere le formule più compatte ed eleganti.

Ipotizzando anche in questo caso che le misure siano tra loro indipendenti, si ottiene formalmente la FIM della (2.14), riportata per comodità:

$$\mathbf{J}(\mathbf{p}_t; \mathbf{p}) = \mathbb{E}_{\mathbf{z}} \left\{ \sum_{i=1}^n \frac{1}{(f_i(z_i|\mathbf{p}_t))^2} \begin{bmatrix} \left(\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \right)^2 & \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \\ \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} & \left(\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial y_t} \right)^2 \end{bmatrix} \right\} \quad (2.36)$$

dove la differenza rispetto al caso precedente risiede nelle derivate, più complesse a causa della dipendenza della varianza dalle coordinate x_t e y_t . Infatti, definendo la variabile ausiliaria $\xi_i = z_i - d_i$ per compattare la notazione, la derivata rispetto ad x_t vale:

$$\frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} = \frac{1}{\sqrt{2\pi}} \left[e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} \frac{\partial}{\partial x_t} \left(\frac{1}{\sigma_i(d_i)} \right) + \frac{1}{\sigma_i(d_i)} \frac{\partial}{\partial x_t} \left(e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} \right) \right]. \quad (2.37)$$

Calcolando singolarmente le derivate, ricordando che la distanza è espressa da $d_i = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}$ e la varianza dalla (2.33):

$$\frac{\partial}{\partial x_t} \frac{1}{\sigma_i} = \frac{\alpha \cos \theta_i}{2\sigma_i d_i} \quad (2.38)$$

$$\frac{\partial}{\partial x_t} e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} = -e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} \frac{\xi_i \cos \theta_i}{\sigma_i^2} \left(\frac{\alpha \xi_i}{2d_i} + 1 \right) \quad (2.39)$$

e sostituendo nella (2.37) si ottiene:

$$\begin{aligned} \frac{\partial f_i(z_i|\mathbf{p}_t)}{\partial x_t} &= \frac{\cos \theta_i}{\sqrt{2\pi}\sigma_i(d_i)} e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} \left(-\frac{\alpha \xi_i^2}{2d_i \sigma_i^2(d_i)} - \frac{\xi_i}{\sigma_i^2(d_i)} + \frac{\alpha}{2d_i} \right) = \\ &= f_i(z_i|\mathbf{p}_t) \left(-\frac{\alpha \xi_i^2}{2d_i \sigma_i^2(d_i)} - \frac{\xi_i}{\sigma_i^2(d_i)} + \frac{\alpha}{2d_i} \right) \cos \theta_i \end{aligned} \quad (2.40)$$

dove si vede la differenza rispetto al caso con varianza di errore di stima costante della (2.16). Da notare che per entrambi i modelli di varianza, compaiono gli stessi termini geometrici, per cui la matrice $\mathbf{M}(\theta_i)$ è la medesima. Si definisce nuovamente $g(\xi_i)$:

$$\begin{aligned} g(\xi_i) &= \frac{1}{\sqrt{2\pi}\sigma_i(d_i)} e^{-\frac{\xi_i^2}{2\sigma_i^2(d_i)}} \left(-\frac{\alpha \xi_i^2}{2d_i \sigma_i^2(d_i)} - \frac{\xi_i}{\sigma_i^2(d_i)} + \frac{\alpha}{2d_i} \right) = \\ &= f_i(z_i|\mathbf{p}_t) \left(-\frac{\alpha \xi_i^2}{2d_i \sigma_i^2(d_i)} - \frac{\xi_i}{\sigma_i^2(d_i)} + \frac{\alpha}{2d_i} \right) = \\ &= f_i(z_i|\mathbf{p}_t) (a\xi_i^2 + b\xi_i + c) \end{aligned} \quad (2.41)$$

nel quale si sono introdotte le quantità:

$$a = -\frac{\alpha}{2d_i \sigma_i^2(d_i)}, \quad b = -\frac{1}{\sigma_i^2(d_i)}, \quad c = \frac{\alpha}{2d_i},$$

utili per il calcolo del quadrato di $g(\xi_i)$ in modo compatto:

$$g^2(\xi_i) = (f_i(z_i|\mathbf{p}_t))^2 \left[a^2 \xi_i^4 + 2ab \xi_i^3 + (b^2 + 2ac) \xi_i^2 + 2bc \xi_i + c^2 \right]. \quad (2.42)$$

Si calcola il coefficiente A_i tramite la (2.22):

$$\begin{aligned} A_i &= \int_{-\infty}^{\infty} \frac{g^2(\xi_i)}{f_i(z_i|\mathbf{p}_t)} d\xi_i = \\ &= \int_{-\infty}^{\infty} f_i(z_i|\mathbf{p}_t) \left[a^2 \xi_i^4 + 2ab \xi_i^3 + (b^2 + 2ac) \xi_i^2 + 2bc \xi_i + c^2 \right] d\xi_i. \end{aligned} \quad (2.43)$$

Sfruttando la proprietà di linearità dell'operatore integrale e alcuni integrali notevoli:

$$\begin{aligned} \int_{-\infty}^{\infty} t e^{-\frac{t^2}{2\kappa^2}} dt &= 0 \\ \int_{-\infty}^{\infty} t^2 e^{-\frac{t^2}{2\kappa^2}} dt &= \kappa^3 \sqrt{2\pi} \\ \int_{-\infty}^{\infty} t^3 e^{-\frac{t^2}{2\kappa^2}} dt &= 0 \\ \int_{-\infty}^{\infty} t^4 e^{-\frac{t^2}{2\kappa^2}} dt &= 3\kappa^5 \sqrt{2\pi} \end{aligned}$$

sostituendo in a, b, c , si ha:

$$A_i = \frac{\alpha^2}{2d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha}. \quad (2.44)$$

Ponendo $\alpha = 0$, ovvero riconducendosi al caso a varianza indipendente dalla distanza, si ottiene esattamente il risultato della (2.24). È importante osservare che anche in questo caso la matrice $\mathbf{J}(\mathbf{p}_t; \mathbf{p})$ è ancora formalmente uguale alla (2.23).

Il calcolo del PEB, analogamente a quanto fatto in precedenza, è dato dalla radice quadrata della traccia di $\mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p})$ e si ottiene:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n A_i \cos^2 \theta_i \sum_{i=1}^n A_i \sin^2 \theta_i - (\sum_{i=1}^n \cos \theta_i \sin \theta_i)^2}}. \quad (2.45)$$

Questa formulazione del PEB è più generale: cambiando il modello della misura, l'equazione resta valida, ma cambiano i coefficienti A_i . Trasformando le funzioni trigonometriche, si ottiene la formulazione seguente:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n A_i \left(\frac{x_i - x_t}{d_i}\right)^2 \sum_{i=1}^n A_i \left(\frac{y_i - y_t}{d_i}\right)^2 - \left(\sum_{i=1}^n A_i \frac{(x_i - x_t)(y_i - y_t)}{d_i^2}\right)^2}}. \quad (2.46)$$

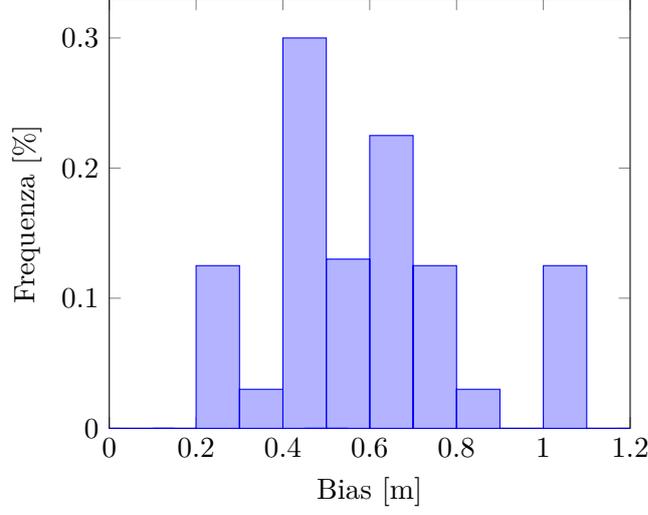


Figura 2.3: Istogramma della frequenza del valore di bias ottenuti in [48] con misure in ufficio.

Sfruttando la (2.29) e la (2.30), è possibile esprimere il PEB nella forma compatta:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\frac{\sum_i^n A_i}{(\sum_i^n A_i c_i^2) (\sum_i^n A_i s_i^2) - (\sum_i^n A_i c_i s_i)^2}} \quad (2.47)$$

che tornerà utile per effettuare un confronto fra formulazione del PEB per un algoritmo centralizzato e per la formulazione distribuita con network multi-hop con conseguenti ritardi di trasmissione.

2.3 Modelli con bias

Nella quasi totalità delle applicazioni, gli UAV dovranno localizzare l'emettitore in un ambiente in cui siano presenti degli ostacoli, che non permettono di sfruttare l'ipotesi di essere in linea di vista o *Line of Sight* (LOS). Questo obbliga a considerare la misura z_i come:

$$z_i = d_i(\mathbf{p}_t, \mathbf{p}_i) + v_i + \beta_i \quad (2.48)$$

dove $d_i(\mathbf{p}_t, \mathbf{p}_i)$ è la reale distanza tra il nodo i -esimo e l'emettitore, v_i è la variabile aleatoria gaussiana a valor medio nullo con varianza σ_i^2 che caratterizza l'incertezza della stima e β_i è una variabile aleatoria che caratterizza il bias della misura.

Nella maggior parte delle applicazioni non è realistico assumere una perfetta conoscenza del bias. Un'opzione più realistica è quella di considerare una certa caratterizzazione statistica derivata da misurazioni o indotte dalla

configurazione dell'ambiente. Oltre alla caratterizzazione dell'ambiente, si deve tenere presente che anche la tecnologia influenza la statistica della variabile aleatoria β_i . Come descritto in [46] e [48], per la tecnologia di ranging UWB, il bias sarà sempre non negativo e il suo valore dipende strettamente dall'ambiente. Ci si aspetta che prenda un range di valori più ampio in un ambiente con molti muri, macchine, come un tipico ufficio rispetto al valore in spazio aperto. Infatti da misure UWB fatte in un ufficio, si è notato che il valore del bias salta da un valore ad un altro in base alla posizione dell'emettitore all'interno dell'edificio, come in Figura 2.3. Quest'affermazione, però, può non essere valida se si effettuano delle misure con diverse tecnologie. Anche se una mappa dettagliata dell'ambiente può non essere disponibile, molte volte si può classificare l'ambiente a grandi linee. Effettuando delle campagne di misura in ambienti tipici come ufficio, magazzini, è possibile creare una libreria di istogrammi riportanti la frequenza con cui un valore di bias si verifica e questi istogrammi variano in base alla diversa classe di ambiente. Si può usare l'istogramma della classe d'ambiente più simile allo scenario in studio per approssimare la distribuzione di probabilità del bias. A causa della difficoltà nel trovare un buon modello analitico generale per la descrizione del bias, si è scelto di analizzare il comportamento in simulazione senza derivare analiticamente il PEB.

La misura del i -esimo UAV sarà data da:

$$z_i = d_i(\mathbf{p}_t, \mathbf{p}_i) + v_i + \beta_i = d_i(\mathbf{p}_t, \mathbf{p}_i) + \epsilon_i \quad (2.49)$$

dove $\epsilon_i = v_i + \beta_i$ è una variabile aleatoria caratterizzata da diverse statistiche in base alla presenza o meno del nodo i -esimo all'interno della b -esima regione "biased" \mathcal{B}_b . Ipotizzando, per semplicità, che il bias sia descritto da una variabile aleatoria Gaussiana, allora si ha $\beta_i \sim \mathcal{N}(\beta_i; \mu, \sigma_b^2)$, dove μ è il valor medio e σ_b^2 è la varianza. Sfruttando questa ipotesi, si ha:

$$\epsilon_i \sim \begin{cases} \mathcal{N}(\epsilon_i; \mu, \sigma_i^2), & \mathbf{p}_i \notin \mathcal{B}_b \\ \mathcal{N}(\epsilon_i; \mu, \sigma_i^2 + \sigma_b^2), & \mathbf{p}_i \in \mathcal{B}_b. \end{cases} \quad (2.50)$$

È evidente dalla (2.50) che quando un nodo si trova all'interno di una regione "biased", la sua misura sarà più incerta a causa della sovrapposizione delle due variabili aleatorie Gaussiane, che genera una nuova variabile aleatoria Gaussiana con valor medio dato dalla somma dei valori medi e varianza pari alla somma delle varianze. Questo evento si verifica quando un nodo non si trova in linea di vista, ad esempio per la presenza di ostacoli lungo la congiungente tra \mathbf{p}_t e \mathbf{p}_i . Per questa variazione del modello, risulta utile utilizzare degli algoritmi che permettano ai nodi di capire se si trovano in LOS o non in linea di vista *Non-Line of Sight* (NLOS) rispetto all'emettitore, per far calcolare le posizioni ottimali agli UAV nonostante la presenza di bias.

In letteratura sono presenti numerosi algoritmi che, analizzando la struttura del segnale, permettono di identificare con una certa probabilità di successo p_d se ci si trova in LOS o in NLOS. La maggior parte delle tecniche per l'identificazione LOS/NLOS proposte in letteratura sono basate sullo schema di detection binario. Esso si basa sull'estrazione di un certo numero di caratteristiche $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_N]^T$ dal segnale ricevuto e sull'applicazione del test di verosimiglianza:

$$\frac{p(\boldsymbol{\gamma}|\text{LOS})}{p(\boldsymbol{\gamma}|\text{NLOS})} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \frac{p(\text{NLOS})}{p(\text{LOS})} \quad (2.51)$$

dove $p(\boldsymbol{\gamma}, \text{LOS})$ e $p(\boldsymbol{\gamma}, \text{NLOS})$ sono le funzioni densità di probabilità congiunte del set $\boldsymbol{\gamma}$ sotto condizioni rispettivamente LOS e NLOS, $p(\text{LOS})$ e $p(\text{NLOS})$ sono le probabilità a priori e \mathcal{H}_0 e \mathcal{H}_1 sono rispettivamente l'ipotesi di essere in condizione LOS e NLOS. In molti casi, questo test può essere complicato da calcolare, quindi si utilizzano ipotesi semplificative.

Ogni algoritmo è strettamente legato alla caratteristica $\boldsymbol{\gamma}$ estratta dal segnale ricevuto. Un parametro spesso utilizzato è il valore quadratico medio della dispersione temporale dell'energia nel segnale τ_{rms} . Nel caso LOS, il percorso ad energia maggior è solitamente il primo a raggiungere il ricevitore mentre nel caso NLOS si possono ricevere prima degli echi con energia minore, risultando in una grande dispersione del ritardo. Sotto certe ipotesi, la (2.51) è equivalente a comparare $\boldsymbol{\gamma}$ ad una soglia λ_τ ottenuta dall'intersezione tra $p(\boldsymbol{\gamma}, \text{LOS})$ e $p(\boldsymbol{\gamma}, \text{NLOS})$. In questo caso, chiamando D la decisione, si ha:

$$D = \begin{cases} \text{LOS}, & \text{se } \tau_{\text{rms}} \leq \lambda_\tau \\ \text{NLOS}, & \text{se } \tau_{\text{rms}} > \lambda_\tau. \end{cases} \quad (2.52)$$

Un altro parametro preso in considerazione in letteratura è la curtosi κ . Le forme d'onda nel caso LOS portano un alto valore di κ e, fissata una soglia λ_κ , si decide:

$$D = \begin{cases} \text{LOS}, & \text{se } \kappa \geq \lambda_\kappa \\ \text{NLOS}, & \text{se } \kappa < \lambda_\kappa. \end{cases} \quad (2.53)$$

L'ultimo descritto tra i parametri considerati in letteratura è l'energia del segnale ricevuto ε_r . Intuitivamente, nel caso LOS si riceverà una energia maggiore rispetto al caso NLOS a causa dell'attenuazione dovuta ad ostacoli tra emettitore e ricevitore, quindi:

$$D = \begin{cases} \text{LOS}, & \text{se } \varepsilon_r \geq \lambda_\varepsilon \\ \text{NLOS}, & \text{se } \varepsilon_r < \lambda_\varepsilon. \end{cases} \quad (2.54)$$

In [49] sono riportate le prestazioni di algoritmi che calcolano la decisione D utilizzando le caratteristiche $\boldsymbol{\gamma}$ descritte in precedenza. I risultati si basano su utilizzando delle misure reali ottenute all'Università di Bologna, in un

Grandezza caratteristica	% Detection		Error rate
	$\mathcal{H}_0 \mathcal{H}_0$	$\mathcal{H}_1 \mathcal{H}_1$	
τ_{rms}	77%	79%	22%
κ	74%	77%	25%
$\tau_{\text{rms}}, \kappa$	88%	79%	17%
ε_r	96%	87%	9%

Tabella 2.1: Performance degli schemi classici di LOS/NLOS detection.

ufficio. Durante l'esperimento sono stati eliminati oggetti in movimento tra trasmittente e ricevente, quindi le attenuazioni dei segnali UWB sono stati soprattutto causati dalla presenza dei muri, dagli strumenti presenti nell'ufficio e dalle persone situate dietro ai dispositivi. Da Tabella 2.1, si vede come in uno scenario di questo tipo la percentuale di corretta detection è superiore al $p_d = 75\%$. Nella terza riga viene riportata la prestazione di uno schema che sfrutta congiuntamente le caratteristiche del segnale τ_{rms} e κ , risultando migliore rispetto agli algoritmi basati sui singoli parametri. L'ultima riga corrisponde al caso in cui viene analizzata l'energia del segnale: i risultati sono molto migliori rispetto gli altri metodi ma, come descritto in [49], questo può essere causato dall'ambiente in cui sono state effettuate le misure. Le simulazioni per il caso centralizzato verranno discusse nel dettaglio in Sezione 4.3, dove verranno analizzate le prestazioni in transitorio in presenza di algoritmi con diverse percentuali di detection.

2.4 Modello di osservazione distribuito

Nel caso di algoritmo distribuito, i nodi devono essere in grado di scambiarsi informazioni relative alle posizioni per calcolare accuratamente il PEB. A differenza della trattazione valida per l'algoritmo centralizzato, come si vedrà a breve, è importante specificare l'iterazione relativa a cui si riferisce l'informazione riguardante la posizione degli UAV.

A causa delle diverse informazioni che ciascun UAV possiede, è necessario andare a formalizzare nuovamente il PEB. A differenza del caso centralizzato dove tutti i nodi sono a conoscenza dello stesso PEB, inoltre, ogni nodo ne calcolerà uno diverso e dovrà calcolarsi in modo autonomo la miglior posizione da tenere per minimizzarlo.

Partendo dalle informazioni ricevute, lo scopo dell'algoritmo di navigazione e localizzazione implementato su ciascun UAV è quello di creare dei segnali di controllo in modo da minimizzare l'errore nella stima della posizione del target. Il modello di transizione degli UAV è:

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \mathbf{u}_i^{(k)} + \mathbf{w}_i^{(k)} \quad (2.55)$$

dove $\mathbf{w}_i^{(k)}$ è il rumore di transizione e $\mathbf{u}_i^{(k)} = [\Delta x_i^{(k)}, \Delta y_i^{(k)}]^T$ è il segnale di controllo.

Nel caso generale, definisce $\mathbf{z}_i^{(k)}$ il vettore contenente le misure disponibili all' i -esimo UAV al time slot k :

$$\mathbf{z}_i^{(k)} = \left[\dots, z_j^{(k-h_{ij}^{(k)}+1)}, \dots \right]^T = \mathbf{h}_i^{(k)}(\mathbf{p}_t^{(k)}) + \mathbf{v}_i^{(k)} \quad (2.56)$$

dove il generico elemento $z_j^{(k-h_{ij}^{(k)}+1)}$ è la misura del j -esimo UAV ritardato del numero di hops presenti tra la coppia di nodi i, j , $\mathbf{h}_i^{(k)}(\cdot)$ è la funzione che lega la posizione del target alla misura osservata, $\mathbf{v}_i^{(k)}$ è il vettore di variabili aleatorie Gaussiane a valor medio nullo e con matrice covarianza:

$$\Sigma_i^{(k)} = \text{diag} \left(\dots, \left(\sigma_j^{(k-h_{ij}^{(k)}+1)} \right)^2, \dots \right) \quad (2.57)$$

mentre $h_{ij}^{(k)}$ è il numero di hop tra la coppia di nodi i, j all'istante temporale k .⁴

Supponendo che siano effettuate stime di ranging, nel caso più generale di varianza dipendente dalla distanza e presenza di bias, l'osservazione dell' i -esimo UAV al time slot k la (2.56) diventa:

$$z_i^{(k)} = \left\| \mathbf{p}_i^{(k)} - \mathbf{p}_{t_i}^{(k)} \right\|_2 + v_i^{(k)} + \beta_i^{(k)} = d_i^{(k)} + \epsilon_i^{(k)} \quad (2.58)$$

dove $d_i^{(k)}$ è la distanza dall'emettitore, $v_i^{(k)} \sim \mathcal{N} \left(v_i^{(k)}; 0, \sigma_i^{(k)2} \right)$ è la variabile aleatoria che descrive il rumore mentre $\beta_i^{(k)}$ è modellata con:

$$\beta_i^{(k)} = \begin{cases} \mathcal{N} \left(\beta_i^{(k)}; \mu, \sigma_b^2 \right), & \text{se NLOS} \\ 0, & \text{se LOS} \end{cases} \quad (2.59)$$

ed è la variabile aleatoria che descrive il bias, per semplicità Gaussiano, al time slot k del nodo i -esimo. Infine, $\epsilon_i^{(k)}$ è la variabile aleatoria che unisce il bias e il rumore di misura e vale:

$$\epsilon_i^{(k)} = \begin{cases} \mathcal{N} \left(\epsilon_i^{(k)}; \mu, \sigma_i^{(k)2} + \sigma_b^2 \right), & \text{se NLOS} \\ \mathcal{N} \left(\epsilon_i^{(k)}; \mu, \sigma_i^{(k)2} \right), & \text{se LOS} \end{cases} \quad (2.60)$$

⁴Nel seguito della trattazione si utilizzerà la notazione compatta $h_{ij} = h_{ij}^{(k)}$.

A partire dalla (2.58) è possibile calcolare la *distributed* log-likelihood function per l' i -esimo UAV al k -esimo time slot:

$$\Lambda_i \left(\mathbf{z}_i^{(k)} \middle| \mathbf{p}_{t_i}^{(k)} \right) = \ln f \left(z_i^{(k)} \middle| \mathbf{p}_{t_i}^{(k)} \right) + \sum_{\substack{j=1 \\ j \neq i}}^N \ln f \left(z_j^{(k-h_{ij}+1)} \middle| \mathbf{p}_{t_i}^{(k)} \right) \quad (2.61)$$

dove:

$$f \left(\mathbf{z}_i^{(k)} \middle| \mathbf{p}_{t_i}^{(k)} \right) = \mathcal{N} \left(\mathbf{z}_i^{(k)}; \mathbf{h}_i^{(k)}, \tilde{\Sigma}_i^{(k)} \right)$$

Sfruttando nuovamente il PEB come figura di merito per misurare l'accuratezza teorica nella localizzazione, ogni UAV lo dovrà calcolare utilizzando le posizioni che gli sono state inviate dai vicini con relativi ritardi dovuti agli hops. Il vettore delle posizioni note all' i -esimo drone viene definito con:

$$\mathbf{q}_i^{(k)} = \left[\dots, \mathbf{p}_j^{(k-h_{ij}+1)}, \dots \right]^T \quad (2.62)$$

che comprende la propria posizione $\mathbf{p}_i^{(k)}$ sempre aggiornata e le posizioni dei restanti UAV ritardati del numero di hop h_{ij} . Utilizzando $\mathbf{q}_i^{(k)}$, la definizione del PEB è:

$$\text{PEB}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sqrt{\text{Tr} \left\{ \mathbf{J}_i^{-1} \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) \right\}} \quad (2.63)$$

dove la Fisher Information Matrix $\mathbf{J}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right)$ è espressa da:

$$\mathbf{J}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \mathbb{E}_{\mathbf{z}_i^{(k)}} \left\{ \left[\nabla_{\mathbf{p}_{t_i}^{(k)}} \ln \Lambda_i \left(\mathbf{z}_i^{(k)} \middle| \mathbf{p}_{t_i}^{(k)} \right) \right] \left[\nabla_{\mathbf{p}_{t_i}^{(k)}} \ln \Lambda_i \left(\mathbf{z}_i^{(k)} \middle| \mathbf{p}_{t_i}^{(k)} \right) \right]^T \right\}. \quad (2.64)$$

Prendendo come modello quello caratterizzato da varianza di stima dipendente dalla distanza dal target $\sigma_i^{2(k)} = \sigma_0^2 d_i^\alpha(k)$, seguendo la trattazione di Sezione 2.2 è si ottiene nuovamente una formulazione di $\mathbf{J}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right)$ della (2.23):

$$\mathbf{J}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sum_{j=1}^n A_j^{(k)} \mathbf{M} \left(\theta_j^{(k)} \right) \quad (2.65)$$

dove la matrice $\mathbf{M} \left(\theta_j^{(k)} \right)$ e i coefficienti $A_j^{(k)}$ sono ancora dati dalla (2.21) e dalla (2.44) rispettivamente. A causa dei ritardi dovuti alla network multi-hop, è possibile formalizzare il PEB dell' i -esimo UAV, definendo con $\mathcal{C}_i^{(k)}$ il set di nodi ad esso connessi:

$$\text{PEB}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sqrt{\frac{A_i^{(k)} + \sum_{\substack{j \neq i \\ j \in \mathcal{C}_i^{(k)}}} A_j^{(k-h_{ij}+1)}}{S_{x,i}^{(k)} S_{y,i}^{(k)} - \left(S_{xy,i}^{(k)} \right)^2}} \quad (2.66)$$

dove, utilizzando le relazioni geometriche delle (2.29)-(2.30), si è definito:

$$S_{x,i}^{(k)} = A_i^{(k)} \left(c_i^{(k)} \right)^2 + \sum_{\substack{j \neq i \\ j \in \mathcal{C}_i^{(k)}}} A_j^{(k-h_{ij}+1)} \left(c_i^{(k-h_{ij}+1)} \right)^2 \quad (2.67)$$

$$S_{y,i}^{(k)} = A_i^{(k)} \left(s_i^{(k)} \right)^2 + \sum_{\substack{j \neq i \\ j \in \mathcal{C}_i^{(k)}}} A_j^{(k-h_{ij}+1)} \left(s_i^{(k-h_{ij}+1)} \right)^2 \quad (2.68)$$

$$S_{xy,i}^{(k)} = A_i^{(k)} c_i^{(k)} s_i^{(k)} + \sum_{\substack{j \neq i \\ j \in \mathcal{C}_i^{(k)}}} A_j^{(k-h_{ij}+1)} c_i^{(k-h_{ij}+1)} s_i^{(k-h_{ij}+1)}. \quad (2.69)$$

Ipotizzando di non utilizzare la memoria degli UAV per memorizzare le ultime posizioni note degli altri nodi, quando l' i -esimo drone non vede un collegamento col j -esimo (ad esempio a causa del limitato sensor coverage o perché sono stati superati gli hop massimi) dovrà calcolare il PEB senza usare le informazioni del j -esimo UAV. In questo modo, le prestazioni possono essere nettamente penalizzate in caso di UAV molto distante dagli altri, fino a portare alla divergenza quando esso è completamente isolato dagli altri. Per questo nelle sommatorie non è stato posto nessun apice, in quanto varia da caso a caso. Queste ultime considerazioni rappresentano la grande differenza rispetto al caso centralizzato che non soffre di questi problemi.

Se, invece, si utilizza un buffer nel quale ogni drone salva l'ultima posizione nota degli altri UAV, l' i -esimo drone è in grado di calcolare il PEB utilizzando n informazioni di posizione. Alla luce di questa considerazione, il PEB può essere formalizzato diversamente, con sommatorie che possono essere estese fino ad n :

$$\text{PEB}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sqrt{\frac{A_i^{(k)} + \sum_{j \neq i}^n A_j^{(k-h_{ij}+1)}}{S_{x,i}^{(k)} S_{y,i}^{(k)} - \left(S_{xy,i}^{(k)} \right)^2}} \quad (2.70)$$

dove:

$$S_{x,i}^{(k)} = A_i^{(k)} \left(c_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(c_i^{(k-h_{ij}+1)} \right)^2 \quad (2.71)$$

$$S_{y,i}^{(k)} = A_i^{(k)} \left(s_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(s_i^{(k-h_{ij}+1)} \right)^2 \quad (2.72)$$

$$S_{xy,i}^{(k)} = A_i^{(k)} c_i^{(k)} s_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} c_i^{(k-h_{ij}+1)} s_i^{(k-h_{ij}+1)}. \quad (2.73)$$

Capitolo 3

Algoritmo di navigazione

FORMALIZZATA l'espressione del Position Error Bound per diversi modelli, è necessario creare un algoritmo che trovi le migliori posizioni dei nodi mobili, in modo da minimizzarne il valore. Il problema di minimizzazione può essere descritto sinteticamente, come in [44], da:

$$\mathcal{P}_0 : \mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \sqrt{\text{Tr} \{ \mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p}) \}} = \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \quad (3.1)$$

dove \mathbf{p}^* rappresenta la matrice contenente l'insieme delle posizioni ottimali dei nodi calcolate dall'algoritmo di posizionamento mentre, utilizzando la notazione di Capitolo 2, \mathbf{p} corrisponde al vettore comprensivo delle posizioni di tutti i nodi. Ciascun nodo ha la possibilità di avere una posizione in tutto lo spazio bidimensionale \mathbb{R}^{2n} quindi questo primo caso corrisponde ad un problema di ottimizzazione senza vincoli. Il problema espresso dalla (3.1) può, tuttavia, essere modificato in modo da tenere in considerazione alcuni vincoli, come ad esempio il rispetto di una distanza minima tra i nodi. In quest'ultimo caso, il problema di ottimizzazione è espresso da:

$$\mathcal{P} : \mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathcal{U}} \sqrt{\text{Tr} \{ \mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p}) \}} = \arg \min_{\mathbf{p} \in \mathcal{U}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \quad (3.2)$$

dove per $\mathbf{p} \in \mathcal{U}$ si intende che le posizioni dei nodi devono appartenere al sotto-spazio delle soluzioni accettabili, dove la funzione viene minimizzata rispettando i vincoli imposti dal problema.

Vista la complessità di questa operazione e al fatto che i droni non possono collocarsi istantaneamente nella posizione ottimale, si utilizza un algoritmo iterativo, così da costruire per passi successivi la soluzione. Per semplicità, la scelta dell'algoritmo di minimizzazione ricade sul "metodo del gradiente". Infatti, dato un punto, il gradiente della funzione in quel punto fornisce come risultato la direzione verso il massimo della funzione stessa. Per ricercare il minimo, si prende l'antigradiente semplicemente utilizzando

la versione negata del gradiente, come espresso per il caso generico dalla seguente legge:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \varepsilon^{(k)} \odot \nabla_{\mathbf{x}} f(\mathbf{x}^{(k)}) \quad (3.3)$$

dove \mathbf{x} è un generico vettore delle coordinate di interesse, l'apice k indica lo step k -esimo, il simbolo \odot rappresenta il prodotto di Hadamard (elemento per elemento), $\varepsilon^{(k)}$ è un coefficiente moltiplicativo che indica il vettore delle velocità di discesa e che, in generale, può essere diverso per ogni agente. Questo vettore di coefficienti è critico: un valore troppo basso porta ad una convergenza molto lenta e ad un costo computazionale molto elevato; un valore troppo elevato può portare a risultati non corretti e, nei casi peggiori, alla divergenza del metodo. Inoltre, il metodo del gradiente non garantisce di trovare il minimo globale, ma tende al minimo locale più vicino alle posizioni iniziali, soprattutto quando si considerano basse velocità di discesa. In letteratura sono presenti una grande quantità algoritmi più avanzati per la ricerca del minimo globale come, ad esempio, il bubble-sort, l'algoritmo genetico e molti altri, ma non saranno presi in esame in questa tesi.

In applicazioni pratiche, inoltre, la posizione del target non è nota a priori quindi l'algoritmo non può sfruttarla per il calcolo del PEB e si rende necessario l'utilizzo di stimatori di posizione. Supponendo di avere uno stimatore, ad esempio *Maximum Likelihood* (ML), le coordinate possono essere espresse da:

$$\hat{\mathbf{p}}_t = \mathbf{p}_t^{(k)} + \boldsymbol{\xi}^{(k)}, \quad \text{con} \quad \boldsymbol{\xi}^{(k)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (3.4)$$

dove $\hat{\mathbf{p}}_t$ e $\mathbf{p}_t^{(k)}$ sono rispettivamente la posizione stimata e la posizione reale all'iterazione k , $\boldsymbol{\mu}$ rappresenta il vettore dei valori medi e $\boldsymbol{\Sigma}$ è la matrice covarianza del processo aleatorio Gaussiano bivariato. Supponendo che lo stimatore sia unbiased e che l'incertezza complessiva sia divisa allo stesso modo sulle coordinate x, y , caratterizzate da due processi aleatori Gaussiani ed indipendenti X, Y , ricordando che la varianza complessiva è strettamente legata al PEB e vale $\sigma^2 = \text{PEB}^2(\mathbf{p}_t; \mathbf{p})$, allora per la proprietà di somma delle varianze di processi Gaussiani indipendenti si ha:

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{bmatrix} = \begin{bmatrix} \frac{\text{PEB}^{(k-1)}}{\sqrt{2}} & 0 \\ 0 & \frac{\text{PEB}^{(k-1)}}{\sqrt{2}} \end{bmatrix}. \quad (3.5)$$

Il Capitolo 3 è strutturato in modo da affrontare in modo graduale l'implementazione dell'algoritmo. In Sezione 3.1 e Sezione 3.2 viene descritta rispettivamente l'ottimizzazione libera e la minimizzazione vincolata applicabile al caso centralizzato, mentre in Sezione 3.3 si discute come l'implementazione dell'algoritmo distribuito.

3.1 Ottimizzazione non vincolata

Analizzando il caso in cui si cerca la formazione per ridurre al massimo il Position Error Bound senza considerare alcun vincolo, si ricade nella (3.1). Questo caso è molto utile per capire gli aspetti principali degli algoritmi introdotti ma non è applicabile a casi pratici, dove i vincoli sono molti e non possono essere trascurati.

3.1.1 Varianza dell'errore di stima costante e indipendente dalla distanza

Applicando la (3.3) al caso in esame, dove la grandezza da minimizzare è il PEB e le variabili su cui è possibile agire sono le posizioni \mathbf{p} , l'equazione di ottimizzazione è:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \boldsymbol{\varepsilon}^{(k)} \odot \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p}) \quad (3.6)$$

dove $\mathbf{p}^{(k)}$ è il vettore contenente le posizioni correnti degli UAV mentre $\mathbf{p}^{(k+1)}$ sono le posizioni che devono raggiungere nell'intervallo temporale successivo, $\boldsymbol{\varepsilon}^{(k)}$ è un vettore di coefficienti moltiplicativi che, per semplicità, nel seguito della trattazione si assumerà costante e uguale per tutti gli UAV. Nel caso di problema bidimensionale e di varianza di stima costante, il Position Error Bound è dato dalla (2.31), riportata per comodità:

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sigma_0 \sqrt{\frac{n}{\sum_{i=1}^n \left(\frac{x_i - x_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)}\right)^2 \sum_{i=1}^n \left(\frac{y_i - y_t}{d_i(\mathbf{p}_t, \mathbf{p}_i)}\right)^2 - \left(\sum_{i=1}^n \frac{(x_i - x_t)(y_i - y_t)}{d_i^2(\mathbf{p}_t, \mathbf{p}_i)}\right)^2}} \quad (3.7)$$

Per affrontare il calcolo del gradiente, si sceglie di mostrare la trattazione per l' i -esimo UAV, derivando rispetto alle sue coordinate x_i, y_i . Quindi, omettendo lo step k , si deve calcolare:

$$\nabla_{\mathbf{p}_i} \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p}) = \left[\frac{\partial \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p})}{\partial x_i}, \frac{\partial \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p})}{\partial y_i} \right]. \quad (3.8)$$

Calcolando la derivata parziale rispetto ad x_i si ottiene:

$$\frac{\partial \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p})}{\partial x_i} = -\frac{1}{2} \frac{\text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p})}{D_{\text{PEB}}} \frac{\partial D_{\text{PEB}}}{\partial x_i} \quad (3.9)$$

dove D_{PEB} è il denominatore della (3.7). Nella derivata del denominatore, valutandola rispetto la sola posizione \mathbf{p}_i , alcune sommatorie vengono eliminate. Utilizzando una notazione compatta per indicare le differenze di coordinate tra il nodo i -esimo e l'emettitore ed omettendo le dipendenze da $\hat{\mathbf{p}}_t$, \mathbf{p}_i e \mathbf{p} ,

si ottiene:

$$\begin{aligned}\frac{\partial D_{\text{PEB}}}{\partial x_i} &= 2 \left[\frac{\bar{x} \bar{y}^2}{d_i^4} \sum_{i=1}^n \frac{\bar{y}^2}{d_i^2} - \frac{\bar{x} \bar{y}}{d_i^2} \sum_{i=1}^n \frac{\bar{x}^2}{d_i^2} - \frac{\bar{y}}{d_i^2} \left(1 - \frac{2\bar{x}^2}{d_i^2} \right) \sum_{i=1}^n \frac{\bar{x} \bar{y}}{d_i^2} \right] \\ &= 2 \left[\frac{\bar{x} \bar{y}^2}{d_i^4} \sum_{i=1}^n \frac{\bar{y}^2 - \bar{x}^2}{d_i^2} - \frac{\bar{y}}{d_i^2} \left(1 - \frac{2\bar{x}^2}{d_i^2} \right) \sum_{i=1}^n \frac{\bar{x} \bar{y}}{d_i^2} \right]\end{aligned}\quad (3.10)$$

dove $\bar{x} = x_i - \hat{x}_t$, $\bar{y} = y_i - \hat{y}_t$ e $d_i = \|\hat{\mathbf{p}}_t - \mathbf{p}_i\|_2$. Sostituendo il risultato appena ottenuto nella (3.9) si ha:

$$\frac{\partial \text{PEB}}{\partial x_i} = \frac{\text{PEB}}{D_{\text{PEB}}} \left[\frac{\bar{y}}{d_i^2} \left(1 - \frac{2\bar{x}^2}{d_i^2} \right) \sum_{i=1}^n \frac{\bar{x} \bar{y}}{d_i^2} - \frac{\bar{x} \bar{y}^2}{d_i^4} \sum_{i=1}^n \frac{\bar{y}^2 - \bar{x}^2}{d_i^2} \right]. \quad (3.11)$$

Procedendo allo stesso modo, la derivata rispetto la coordinata y_i vale:

$$\frac{\partial \text{PEB}}{\partial y_i} = \frac{\text{PEB}}{D_{\text{PEB}}} \left[\frac{\bar{x}}{d_i^2} \left(1 - \frac{2\bar{y}^2}{d_i^2} \right) \sum_{i=1}^n \frac{\bar{x} \bar{y}}{d_i^2} - \frac{\bar{y} \bar{x}^2}{d_i^4} \sum_{i=1}^n \frac{\bar{y}^2 - \bar{x}^2}{d_i^2} \right]. \quad (3.12)$$

Implementando la (3.3) al calcolatore, sfruttando la (3.11) e la (3.12), è possibile creare un algoritmo di controllo centralizzato non vincolato. I risultati di questo scenario sono discussi in Sezione 4.1.1.

3.2 Ottimizzazione vincolata

In pratica è poco realistico che un problema come quello in esame non sia soggetto a vincoli, che siano di posizione, di distanza, di velocità, o altri. Per questo motivo, si ottengono risultati più veritieri andando a studiare il problema espresso dalla (3.2) rispetto alla (3.1). Il problema sarà caratterizzato da vincoli che modificano lo spazio delle “feasible solutions” \mathcal{U} .

Nelle sezioni seguenti vengono introdotti i principi e i metodi utilizzabili per la risoluzione di un problema vincolato. In particolare, in Sezione 3.2.1 viene introdotto il metodo dei coefficienti di Lagrange che fornisce le basi per affrontare in Sezione 3.2.2 il metodo del gradiente proiettato, potente metodo per affrontare i vincoli legati all’applicazione oggetto della tesi. Successivamente si descrivono i vincoli in Sezione 3.2.3 per poi formulare l’equazione di controllo completa in Sezione 3.2.4.

3.2.1 Metodo dei coefficienti di Lagrange

Il problema che si considera in questa sezione è la minimizzazione di una funzione soggetta a vincoli di uguaglianza e disuguaglianza:

$$\min_{\substack{\text{t.c. } h_i(\mathbf{x})=0, i=1, \dots, n_e \\ g_j(\mathbf{x}) \geq 0, j=1, \dots, n_g}} f(\mathbf{x}) \quad (3.13)$$

dove \mathbf{x} è il vettore delle variabili, $h_i(\mathbf{x})$ è l' i -esimo vincolo di uguaglianza e $g_j(\mathbf{x})$ è il j -esimo vincolo di disuguaglianza. I vincoli dividono lo spazio delle soluzioni in due domini: i) il dominio nel quale i vincoli sono rispettati dove le soluzioni sono accettabili; ii) il dominio nel quale almeno un vincolo non è rispettato dove le soluzioni non sono accettabili.

In generale, la (3.13) può avere diversi minimi locali e solo in casi particolari si ha la certezza che esista un singolo minimo globale. Le condizioni necessarie per la presenza di un minimo in un problema vincolato sono ottenute utilizzando i moltiplicatori di Lagrange [50, 51]. Come primo caso, si considerino i soli vincoli di uguaglianza. Si definisce la funzione Lagrangiana:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{j=1}^{n_e} \lambda_j h_j(\mathbf{x}) \quad (3.14)$$

dove λ_j sono i moltiplicatori di Lagrange sconosciuti. Le condizioni necessarie per un punto stazionario sono:

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f(\mathbf{x})}{\partial x_i} - \sum_{j=1}^{n_e} \lambda_j \frac{\partial h_j}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (3.15)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = h_j(\mathbf{x}) = 0, \quad j = 1, \dots, n_e. \quad (3.16)$$

Queste condizioni si applicano solo in un punto regolare, dove i gradienti dei vincoli sono linearmente indipendenti. Se i gradienti sono linearmente dipendenti, significa che è possibile rimuovere delle constraints senza modificare la soluzione del problema. In un punto regolare, queste due equazioni rappresentano $n + n_e$ equazioni per n_e moltiplicatori e n coordinate del punto stazionario.

Lo scenario si complica se sono presenti dei vincoli di disuguaglianza. Per applicare il metodo dei moltiplicatori di Lagrange è necessario in primo luogo trasformare i vincoli di disuguaglianza in vincoli di uguaglianza aggiungendo delle variabili di appoggio. I vincoli possono essere scritti come:

$$g_j(\mathbf{x}) - t_j^2 = 0, \quad j = 1, \dots, n_g \quad (3.17)$$

dove t_j è una variabile che misura quanto sono distanti le j -esime constraints dall'essere critiche. La funzione Lagrangiana è:

$$\mathcal{L}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{j=1}^{n_g} \lambda_j (g_j(\mathbf{x}) - t_j^2) \quad (3.18)$$

e differenziandola rispetto \mathbf{x} , $\boldsymbol{\lambda}$ e \mathbf{t} si ha:

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f(\mathbf{x})}{\partial x_i} - \sum_{i=1}^{n_g} \lambda_i \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (3.19)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = -g_j(\mathbf{x}) + t_j^2 = 0, \quad j = 1, \dots, n_g \quad (3.20)$$

$$\frac{\partial \mathcal{L}}{\partial t_j} = 2\lambda_j t_j = 0, \quad j = 1, \dots, n_g. \quad (3.21)$$

Dalla (3.21) è immediato notare come per i vincoli non critici venga assegnato un moltiplicatore di Lagrange nullo. Inoltre la (3.19) e la (3.21) rappresentano le condizioni necessarie di un punto stazionario e regolare, che per il caso di vincoli di disuguaglianza è un punto in cui i gradienti dei vincoli *attivi* sono linearmente indipendenti. Queste ultime equazioni, chiamate condizioni Karush-Kuhn-Tucker, sono necessarie per avere un minimo [50, 51].

Se le condizioni Karush-Kuhn-Tucker sono soddisfatte ma è impossibile trovare una direzione con derivata negativa della funzione da minimizzare senza che vengano violati i vincoli, è comunque possibile muoversi in una direzione che sia tangente ai vincoli attivi e perpendicolare al gradiente (con pendenza nulla). Questo è espresso da:

$$\mathbf{s}^T \nabla f(\mathbf{x}) = \mathbf{s}^T \nabla g_j = 0, \quad j \in I_A \quad (3.22)$$

dove \mathbf{s} è la direzione in cui ci si vuole muovere e I_A è l'insieme delle constraints attive. L'effetto di questo movimento è quello di spostarsi in un punto che può ridurre la funzione in oggetto senza violare i vincoli.

Trovare il minimo direttamente dalle condizioni Karush-Kuhn-Tucker può essere difficoltoso. Partendo dalla (3.19) scritta in forma matriciale:

$$\nabla f(\mathbf{x}) - \mathbf{N}\boldsymbol{\lambda} = 0 \quad (3.23)$$

dove la matrice \mathbf{N} è definita da:

$$n_{ij} = \frac{\partial g_j}{\partial x_i}, \quad j = 1, \dots, r \text{ e } i = 1, \dots, n \quad (3.24)$$

considerando solo i moltiplicatori degli r vincoli attivi. Si assume che i gradienti dei vincoli siano indipendenti così che \mathbf{N} abbia rango pieno r . Utilizzando un approccio ai minimi quadrati per risolvere le equazioni, si definisce un vettore residuo \mathbf{u} :

$$\mathbf{u} = \mathbf{N}\boldsymbol{\lambda} - \nabla f(\mathbf{x}). \quad (3.25)$$

Una soluzione ai minimi quadrati della (3.23) minimizzerà la norma del vettore residuo:

$$\begin{aligned} \|\mathbf{u}\|^2 &= (\mathbf{N}\boldsymbol{\lambda} - \nabla f(\mathbf{x}))^T (\mathbf{N}\boldsymbol{\lambda} - \nabla f(\mathbf{x})) = \\ &= \boldsymbol{\lambda}^T \mathbf{N}^T \mathbf{N} \boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T \mathbf{N}^T \nabla f(\mathbf{x}) + \nabla f(\mathbf{x})^T \nabla f(\mathbf{x}) \end{aligned} \quad (3.26)$$

e differenziando rispetto al vettore dei moltiplicatori si ottiene:

$$2\mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} - 2\mathbf{N}^T\nabla f(\mathbf{x}) = 0 \quad (3.27)$$

da cui si può calcolare:

$$\boldsymbol{\lambda} = \left(\mathbf{N}^T\mathbf{N}\right)^{-1} \mathbf{N}^T\nabla f(\mathbf{x}) \quad (3.28)$$

che è la soluzione migliore nel senso dei minimi quadrati. Sostituendo la (3.28) nella (3.23) si arriva ad un importante risultato:

$$\left(\mathbf{I} - \mathbf{N} \left(\mathbf{N}^T\mathbf{N}\right)^{-1} \mathbf{N}^T\right) \nabla f(\mathbf{x}) = \mathbf{P}\nabla f(\mathbf{x}) = 0 \quad (3.29)$$

avendo definito la matrice proiezione \mathbf{P} con:

$$\mathbf{P} = \mathbf{I} - \mathbf{N} \left(\mathbf{N}^T\mathbf{N}\right)^{-1} \mathbf{N}^T. \quad (3.30)$$

La matrice \mathbf{P} se moltiplicata ad un arbitrario vettore lo proietta nello spazio tangente ai vincoli. In questo modo, il vettore \mathbf{x} viene proiettato nello spazio delle soluzioni accettabili e si ricerca un minimo locale al suo interno. Per far sì che il prodotto $\mathbf{N}^T\mathbf{N}$ sia invertibile, la matrice deve avere rango pieno e questo avviene solo se i vettori colonna contenenti i gradienti dei vincoli sono indipendenti tra loro. Da notare che in questo caso, la matrice di proiezione può essere espressa anche utilizzando la pseudo-inversa di Moore-Penrose definita da:

$$\mathbf{N}^\dagger = \left(\mathbf{N}^T\mathbf{N}\right)^{-1} \mathbf{N}^T \quad (3.31)$$

ottenendo:

$$\mathbf{P} = \mathbf{I} - \mathbf{N}\mathbf{N}^\dagger. \quad (3.32)$$

3.2.2 Metodo del gradiente proiettato

Il metodo del gradiente proiettato di Rosen [52, 53] si basa sulla proiezione dello spostamento su un sotto-spazio tangente ai vincoli attivi. Esaminando il metodo nel caso di vincoli lineari si deve affrontare il problema:

$$\begin{aligned} \min_{\text{t.c. } g_j(\mathbf{x}) \geq 0} \quad & f(\mathbf{x}) \\ \text{dove} \quad & g_j(\mathbf{x}) = \sum_{i=1}^n a_{ij}x_i - b_j \geq 0, \quad j = 1, \dots, n_g. \end{aligned} \quad (3.33)$$

I vincoli lineari possono essere espressi sinteticamente in forma matriciale da:

$$g_j(\mathbf{x}) = \mathbf{a}_j^T \mathbf{x} - b_j \geq 0 \quad (3.34)$$

dove il vettore \mathbf{x} è composto da n variabili, $g_j(\mathbf{x})$ è il j -esimo degli n_g vincoli. Supponendo di avere $r < n_g$ vincoli violati si possono definire le equazioni dei vincoli in forma matriciale:

$$\mathbf{g} = \mathbf{N}^T \mathbf{x} - \mathbf{b} = 0 \quad (3.35)$$

dove \mathbf{g} è il vettore dei vincoli attivi. La matrice \mathbf{N} di dimensioni $n \times r$ è composta da colonne che rappresentano i gradienti dei vincoli violati:

$$\mathbf{N} = \left[\nabla_{\mathbf{x}} g_1(\mathbf{x}); \nabla_{\mathbf{x}} g_2(\mathbf{x}); \dots; \nabla_{\mathbf{x}} g_r(\mathbf{x}) \right] \quad (3.36)$$

dove il gradiente è espresso come vettore colonna delle derivate parziali rispetto alle n variabili come per il generico vincolo attivo v :

$$\nabla g_v(\mathbf{x}) = \left[\frac{\partial g_v(\mathbf{x})}{\partial x_1}, \frac{\partial g_v(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial g_v(\mathbf{x})}{\partial x_n} \right]^T. \quad (3.37)$$

L'aggiornamento della soluzione con vincoli lineari è formalizzata con:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\varepsilon}^{(k)} \odot \mathbf{s} \quad (3.38)$$

e se sia $\mathbf{x}^{(k+1)}$ che $\mathbf{x}^{(k)}$ soddisfano la (3.35) allora si ha:

$$\mathbf{N}^T \mathbf{s} = 0 \quad (3.39)$$

e il problema può essere posto come:

$$\min_{\text{t.c. } \mathbf{N}^T \mathbf{s} = 0} \mathbf{s}^T \nabla f(\mathbf{x}). \quad (3.40)$$

Della funzione da minimizzare si costruisce la funzione Lagrangiana:

$$\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \mu) = \mathbf{s}^T \nabla f(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{N} \mathbf{s} - \mu (\mathbf{s}^T \mathbf{s} - 1) \quad (3.41)$$

dove la condizione per cui sia stazionaria è:

$$\frac{\partial \mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \mu)}{\partial \boldsymbol{\lambda}} = \nabla f(\mathbf{x}) - \mathbf{N} \boldsymbol{\lambda} - 2\mu \mathbf{s} = 0. \quad (3.42)$$

Premoltiplicando \mathbf{N}^T alla (3.42) e sfruttando la (3.39) si ottiene:

$$\boldsymbol{\lambda} = \left(\mathbf{N}^T \mathbf{N} \right)^{-1} \mathbf{N}^T \nabla f(\mathbf{x}). \quad (3.43)$$

Sostituendo il risultato appena ricavato nella (3.42), si ottiene la direzione \mathbf{s} :

$$\mathbf{s} = \frac{1}{2\mu} \left[\mathbf{I} - \mathbf{N} \left(\mathbf{N}^T \mathbf{N} \right)^{-1} \mathbf{N}^T \right] \nabla f(\mathbf{x}) = \frac{1}{2\mu} \mathbf{P} \nabla f(\mathbf{x}) \quad (3.44)$$

dove \mathbf{P} è la matrice proiezione definita nella (3.30). Considerando che si ricerca la direzione, è prassi trascurare il fattore moltiplicativo $1/2\mu$ e per cercare il minimo si prende solo la direzione $-\mathbf{s}$.

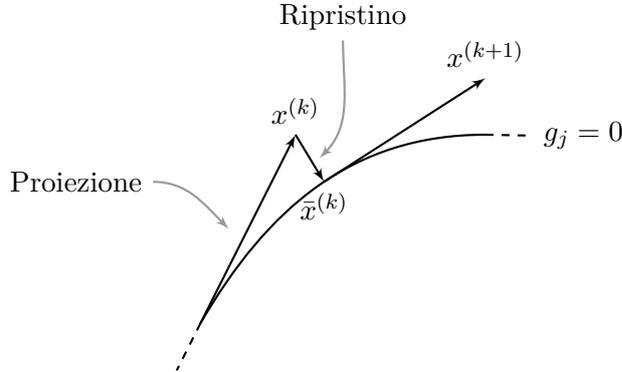


Figura 3.1: Rappresentazione monodimensionale delle mosse di proiezione e ripristino.

Il metodo del gradiente proiettato, però, è stato generalizzato da Rosen ai vincoli non lineari e, in questo caso, si linearizzano i vincoli nel punto $\mathbf{x}^{(k)}$. La difficoltà maggiore causata dalle non linearità è data dal fatto che ci si sposta nel sotto-spazio tangente ai vincoli che non li segue esattamente e per questo è necessario introdurre una mossa di ripristino (*restoration move*) che riporti la soluzione nei boundaries del vincolo. Si linearizzano i vincoli con l'approssimazione al primo ordine:

$$g_j(\mathbf{x}^{(k)}) \approx g_j(\mathbf{x}^{(k)}) + \nabla g_j^T (\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}). \quad (3.45)$$

La correzione $\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}$ ricercata nello spazio tangente, tale che si verifichi $\mathbf{P}(\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}) = 0$, è quella che riduce a zero $g_j(\mathbf{x})$. Si dimostra che la correzione ricercata vale:

$$\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)} = -\mathbf{N}(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{g}(\mathbf{x}^{(k)}). \quad (3.46)$$

La procedura per il calcolo della soluzione, alla luce di questi risultati, consiste nel ripetere iterativamente:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \varepsilon^{(k)} \odot \mathbf{s} + \bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)} = \\ &= \mathbf{x}^{(k)} - \varepsilon^{(k)} \odot \mathbf{P} \nabla f(\mathbf{x}) - \mathbf{N}(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{g}(\mathbf{x}^{(k)}). \end{aligned} \quad (3.47)$$

Graficamente è possibile immaginare di avere un comportamento dell'algoritmo per il caso monodimensionale come rappresentato in Figura 3.1.

Benché il metodo dei coefficienti di Lagrange e il metodo del gradiente proiettato di Rosen si comportino in maniera molto simile per vincoli lineari, la grande differenza si nota quando i vincoli non sono lineari. Infatti, la trattazione di cui sopra mostra come quest'ultimo metodo porti un notevole vantaggio in termini di semplicità di applicazione in caso di vincoli di

disuguaglianza non lineari. Considerando che i vincoli in esame sono di quest'ultimo tipo, si è scelto di adottare il metodo del gradiente proiettato per l'ottimizzazione vincolata implementata nell'algoritmo di navigazione degli UAV.

3.2.3 Collision avoidance

Nel corso di questa sezione vengono introdotti tutti i vincoli che possono presentarsi in applicazioni reali dell'algoritmo. In particolare, vengono analizzati nell'ordine: i) anti-collisione fra UAV; ii) obstacles avoidance per ostacoli di cui sono note a priori posizione e forma; iii) obstacles avoidance per ostacoli dei quali non si hanno informazioni a priori; iv) controllo della velocità.

Collision avoidance fra nodi

Nel caso di anti collisione fra droni, il problema è analogo alla (3.33), con singoli vincoli rappresentati da:

$$g_{ij}(\mathbf{p}_i, \mathbf{p}_j) = d_{ij}(\mathbf{p}_i, \mathbf{p}_j) - d_{min} \geq 0, \quad \forall i \neq j. \quad (3.48)$$

Supponendo di avere $r = V$ vincoli attivi e $2n$ variabili, corrispondenti a ciascuna coppia x_j, y_j , riguardante la posizione del j -esimo nodo, si può utilizzare una notazione matriciale dei soli vincoli di anticollisione fra UAV violati:

$$\mathbf{g}_{ca}(\mathbf{p}) = \mathbf{d}_{ca}(\mathbf{p}) - \mathbf{1} \cdot d_{min} \quad (3.49)$$

dove $\mathbf{d}_{ca}(\mathbf{p})$ è il vettore $V \times 1$ contenente tutte le distanze tra i nodi inferiori alla distanza di sicurezza e $\mathbf{1}$ è il vettore unitario, anch'esso di dimensioni $V \times 1$. Per il calcolo della matrice dei gradienti relativi al vincolo di anticollisione fra UAV \mathbf{N}_{ca} , si calcolano le derivate per ogni vincolo attivo. Supponendo che il vincolo coinvolga due nodi i, j , con coordinate rispettivamente $\mathbf{p}_i = [x_i, y_i]^T$, $\mathbf{p}_j = [x_j, y_j]^T$, esso è dato da:

$$g_{ij}(\mathbf{p}_i, \mathbf{p}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_{min} \geq 0 \quad (3.50)$$

e derivando rispetto una generica variabile x_u , si vede immediatamente che:

$$\frac{\partial g_{ij}(\mathbf{p}_i, \mathbf{p}_j)}{\partial x_u} = \begin{cases} \frac{x_i - x_j}{d_{ij}(\mathbf{p}_i, \mathbf{p}_j)}, & x_u = x_i \\ \frac{x_j - x_i}{d_{ij}(\mathbf{p}_i, \mathbf{p}_j)}, & x_u = x_j \\ 0, & \text{altrimenti.} \end{cases} \quad (3.51)$$

Dalla (3.51) si intuisce che la matrice \mathbf{N}_{ca} avrà molti elementi nulli. Questo è un potenziale problema: un gran numero di elementi nulli rende più semplice

ottenere una matrice a rango deficitario, quindi non invertibile. Inoltre, si rende necessario utilizzare la proprietà:

$$g_{ij}(\mathbf{p}_i, \mathbf{p}_j) = g_{ji}(\mathbf{p}_i, \mathbf{p}_j) \quad (3.52)$$

ipotizzando di avere una sorta di bidirezionalità del vincolo tra la coppia di nodi i, j . Da notare che nel caso in cui non sia sfruttata l'ipotesi della (3.52), la matrice \mathbf{N}_{ca} avrebbe contenuto almeno due colonne uguali ma di segno opposto, ricordando la (3.51), rendendo la matrice a rango deficitario e il prodotto $(\mathbf{N}_{ca}^T \mathbf{N}_{ca})^{-1}$ non invertibile.

Guardando Figura 3.2 si può vedere il comportamento tenuto dai nodi quando sono a distanza ravvicinata, con un vincolo di distanza mutua di $d_{min} = 3\text{m}$. Figura 3.2a e Figura 3.2b mostrano il comportamento dei nodi quando sono disposti a distanza ravvicinata lungo una linea retta, rispettivamente per il caso a 4 e 8 nodi. È interessante vedere che, come ci si aspetta, i nodi tendano ad allontanarsi mantenendo la linea, fino ad arrivare al punto in cui la distanza minima è rispettata per tutti. Un caso più rappresentativo di una configurazione transitoria è rappresentato da Figura 3.2c e Figura 3.2d, dove rispettivamente 4 e 8 nodi sono posizionati in modo causale ma a breve distanza. Per il primo caso si hanno dei movimenti sufficientemente lineari e dolci mentre nel secondo caso, a causa dell'elevata densità di nodi, i movimenti degli UAV più interni sono irregolari. In tutti i casi di Figura 3.2, però, i nodi si dispongono in posizioni tali da avere distanze mutue che rispettino il vincolo.

Come si vedrà in seguito, alcuni scenari applicativi portano i nodi ad avvicinarsi all'emettitore per far sì che la misura sulla distanza sia di qualità maggiore. In questi casi, se l'emettitore non è all'interno di un ostacolo nel quale i nodi non possono entrare, è necessario configurare l'algoritmo di collision avoidance anche per far rispettare vincoli di distanza minima dal target. Anche in questo caso devono essere calcolate la matrice dei gradienti e la matrice di proiezione. Ipotizzando di mantenere la distanza minima d_{min_t} , utilizzando la distanza euclidea tra nodo i -esimo ed emettitore situato nel punto $\mathbf{p}_t = [x_t, y_t]^T$, i vincoli sono dati da:

$$g_{it}(\mathbf{p}_i, \mathbf{p}_t) = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2} - d_{min_t} \geq 0, \quad \forall i. \quad (3.53)$$

Come fatto in precedenza, supponendo di avere $r = V_t$ vincoli attivi, si crea un vettore $\mathbf{g}_t(\mathbf{p}, \mathbf{p}_t)$ rappresentativo della criticità dei vincoli non rispettati:

$$\mathbf{g}_t(\mathbf{p}, \mathbf{p}_t) = \mathbf{d}_t(\mathbf{p}, \mathbf{p}_t) - \mathbf{1} \cdot d_{min_t} \quad (3.54)$$

dove $\mathbf{d}_t(\mathbf{p}, \mathbf{p}_t)$ è il vettore di dimensioni $V_t \times 1$ contenente le distanze dall'emettitore dei soli nodi a distanza critica e $\mathbf{1}$ è il vettore unitario, anch'esso di dimensioni $V_t \times 1$. Il gradiente del generico vincolo $g_{it}(\mathbf{p}_i, \mathbf{p}_t)$ rispetto alle $2n$ coordinate dei nodi mobili sarà dato da un vettore di elementi nulli

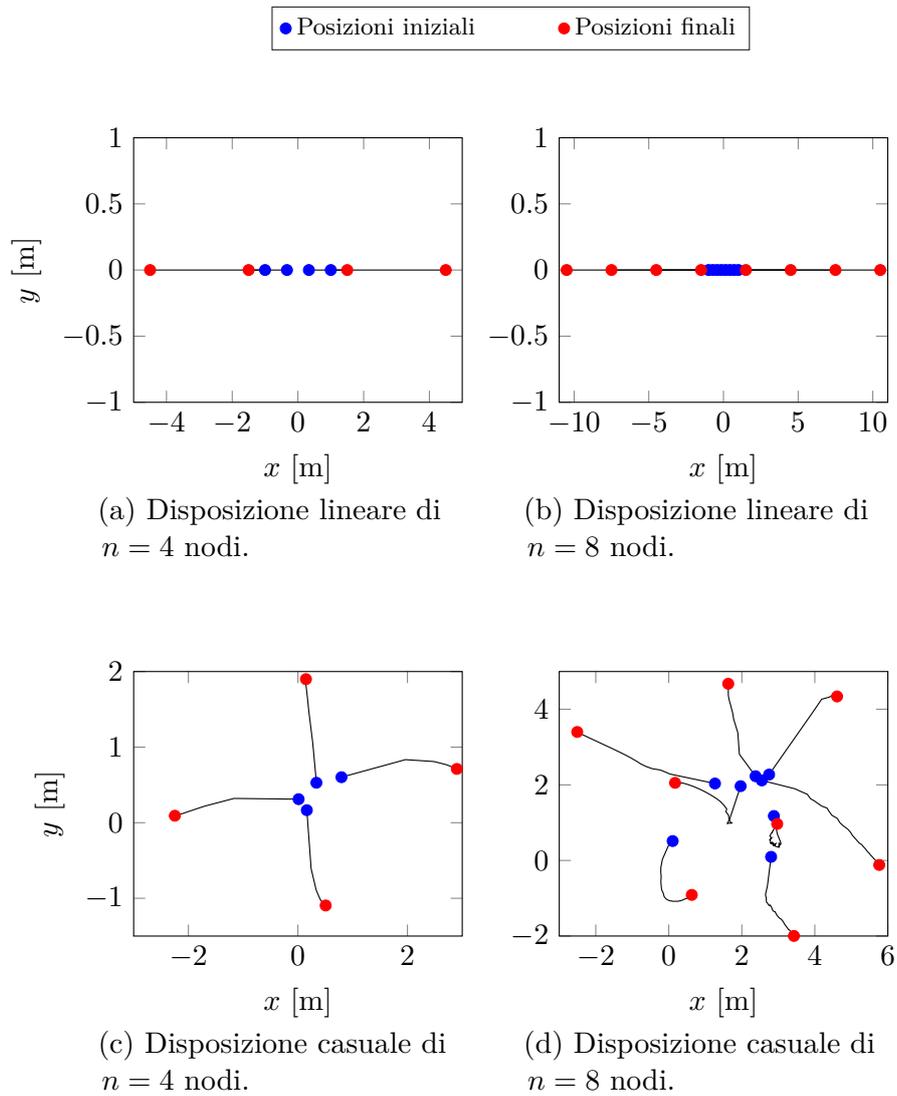


Figura 3.2: Collision avoidance di un numero variabile di nodi e in diverse configurazioni iniziali.

a parte le due righe corrispondenti alle coordinate x_i, y_i dove le derivate valgono rispettivamente:

$$\frac{\partial g_{it}(\mathbf{p}_i, \mathbf{p}_t)}{\partial x_i} = \frac{x_i - x_t}{d_{it}(\mathbf{p}_i, \mathbf{p}_t)} \quad (3.55)$$

$$\frac{\partial g_{it}(\mathbf{p}_i, \mathbf{p}_t)}{\partial y_i} = \frac{y_i - y_t}{d_{it}(\mathbf{p}_i, \mathbf{p}_t)}. \quad (3.56)$$

Non è necessario calcolare le derivate rispetto le coordinate dell'emittitore in quanto non sono variabili controllabili. Applicando alla (3.36) i vincoli di distanza minima dal target attivi, si ottiene la matrice dei gradienti \mathbf{N}_t ed è possibile calcolare la matrice di proiezione \mathbf{P}_t utilizzando la (3.30).

La versatilità del metodo del gradiente proiettato sta nel fatto che è possibile unire vincoli di diversa natura. Infatti, nel caso in cui siano attivi vincoli legati alla vicinanza tra due nodi e che ci siano dei nodi troppo vicini all'emittitore, è possibile concatenare le matrici dei gradienti e i vettori dei vincoli attivi. In generale, in presenza di C constraints attive, le matrici dei gradienti devono essere concatenate orizzontalmente mentre i vettori di violazione devono essere concatenati verticalmente:

$$\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2, \dots, \mathbf{N}_C] \quad (3.57)$$

che avrà dimensioni $2n \times \sum_{c=1}^C V_c$ e:

$$\mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \dots \\ \mathbf{g}_C \end{bmatrix} \quad (3.58)$$

di dimensioni $\sum_{c=1}^C V_c$. Utilizzando queste grandezze, è possibile calcolare la matrice di proiezione per tutti i vincoli attivi.

Obstacles avoidance: Artificial Potential Field

Per inserire all'interno dell'algoritmo di navigazione una funzione di obstacles avoidance, un formalismo matematico molto utile ed elegante è l'*Artificial Potential Field* (APF) [54–58] che permette di semplificare notevolmente l'espressione dei vincoli legati agli ostacoli. Anche se in letteratura è possibile trovare dei lavori che utilizzano unicamente l'APF [54–56, 59] per il path-planning, nell'implementazione dell'algoritmo di navigazione di UAV si è scelto di integrarlo all'interno del metodo del gradiente proiettato.

Esso si basa sulla creazione di campi potenziali attorno ad un ostacolo così che un UAV nelle sue vicinanze calcoli una posizione tale da portarsi a distanza di sicurezza. L'algoritmo introdotto da O. Khatib nel 1986 [57], prende spunto dalla definizione fisica di forze conservative $F(\mathbf{x})$, caratterizzate dall'importante proprietà di ammettere una funzione potenziale $U(\mathbf{x})$,

dove \mathbf{x} è il vettore delle coordinate. Infatti, sotto questa ipotesi, la forza è legata al potenziale tramite l'antigradiente:

$$F(\mathbf{x}) = -\nabla U(\mathbf{x}). \quad (3.59)$$

La forza risultante è data dalla forza attrattiva e dalla sommatoria delle forze repulsive corrispondenti al numero di ostacoli presenti:

$$F(\mathbf{x}) = F_{att}(\mathbf{x}) + \sum_{i=1}^{n_{obs}} F_{rep_i}(\mathbf{x}). \quad (3.60)$$

Nel caso di ricerca di una formazione ottimale, la forza attrattiva è rappresentata dalla direzione che i nodi intraprendono nella minimizzazione del PEB, mentre la forza repulsiva è rappresentata dalla derivata del potenziale degli ostacoli. In base alla definizione del potenziale, si possono ottenere diverse forme di ostacolo: in questo caso vengono analizzati i casi più semplici di ostacolo circolare e di ostacolo rettangolare. Questo algoritmo si presta ad essere utilizzato all'interno del metodo del gradiente proiettato: quando un vincolo è violato, viene inserito nel vettore \mathbf{g} e il corrispondente gradiente del potenziale viene inserito nella matrice \mathbf{N} , dalla quale viene calcolata la matrice di proiezione \mathbf{P} .

Nel caso di ostacolo circolare centrato in $[x_c, y_c]^T$, una funzione potenziale $p_c(x, y)$, nota in letteratura, è data da:

$$p_c(x, y) = \begin{cases} \frac{1}{2}k_0 \left(\frac{1}{\rho(x, y)} - \frac{1}{\rho_0} \right)^2, & \rho \leq \rho_0 \\ 0, & \rho > \rho_0 \end{cases} \quad (3.61)$$

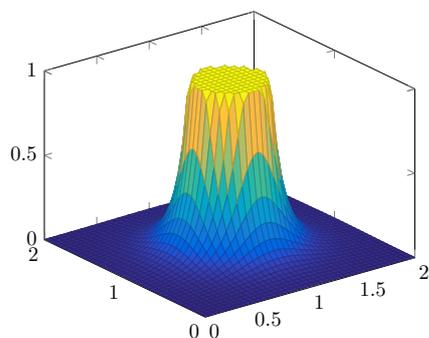
dove k_0 è una costante di scala positiva, $\rho(x, y)$ è la distanza euclidea del nodo i -esimo dal centro dell'ostacolo e ρ_0 rappresenta il raggio oltre al quale il nodo si può considerare come non critico rispetto all'ostacolo. Implementando la (3.61) al calcolatore, si ottiene il grafico di Figura 3.3a. Per il calcolo della direzione per cui l'UAV si porta in una condizione di sicurezza, è necessario calcolare il gradiente del potenziale $p_c(x, y)$ rispetto alle coordinate del nodo. Banalmente, si vede che:

$$\frac{\partial p_c(x, y)}{\partial x_i} = -k_0 \left(\frac{1}{\rho(x, y)} - \frac{1}{\rho_0} \right) \frac{x_i}{\rho(x, y)^3} \quad (3.62)$$

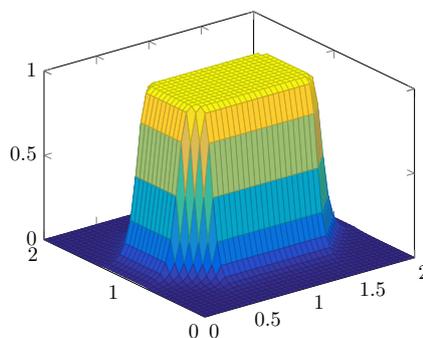
$$\frac{\partial p_c(x, y)}{\partial y_i} = -k_0 \left(\frac{1}{\rho(x, y)} - \frac{1}{\rho_0} \right) \frac{y_i}{\rho(x, y)^3}. \quad (3.63)$$

Per modellare un ostacolo che abbia la forma di un poligono, la funzione potenziale viene costruita a partire dal set di disequazioni che descrivono la regione in cui gli UAV possono giacere [60]:

$$g_i(x, y) \leq 0, \quad i = 1, \dots, I \quad (3.64)$$



(a) Rappresentazione del potenziale di un ostacolo circolare.



(b) Rappresentazione del potenziale di un ostacolo rettangolare.

Figura 3.3: Rappresentazione del potenziale di un ostacolo rettangolare.

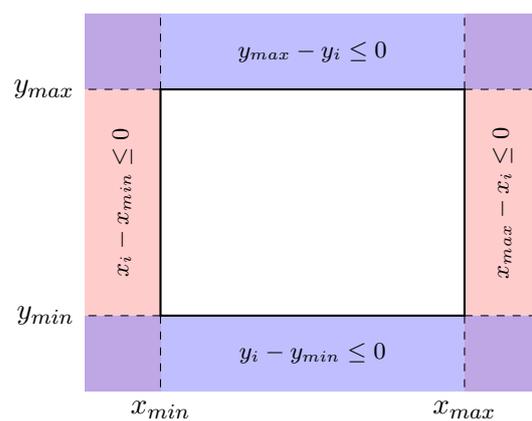


Figura 3.4: Visualizzazione grafica delle equazioni dei vincoli di segmento.

dove con $g_i(x, y)$ rappresenta una funzione lineare e I è il numero totale di segmenti. Si definisce la funzione scalare $f(x, y)$ data da:

$$f(x, y) = \sum_{i=1}^I g_i(x, y) + |g_i(x, y)| \quad (3.65)$$

che è caratterizzata dalla proprietà di essere nulla all'interno della regione proibita e di crescere linearmente con la distanza da essa. Allora è possibile definire la funzione potenziale $p_r(x, y)$:

$$p_r(x, y) = \frac{1}{\delta + f(x, y)} = \frac{1}{\delta + \sum_{i=1}^I (g_i(x, y) + |g_i(x, y)|)} \quad (3.66)$$

dove δ è una piccola costante positiva che permette di scalare il valore del potenziale. Infatti, il potenziale ha il suo massimo all'interno della regione delimitata dai segmenti ed è pari a δ^{-1} .

Per la costruzione del potenziale di un ostacolo rettangolare si formalizzano le $I = 4$ disuguaglianze che esprimono i vincoli:

$$g_i(x, y) \leq 0, \quad i = 1, \dots, 4$$

dove i quattro segmenti sono descritti dalle seguenti equazioni:

$$x_i - x_{min} \leq 0 \quad (3.67)$$

$$y_i - y_{min} \leq 0 \quad (3.68)$$

$$x_{max} - x_i \leq 0 \quad (3.69)$$

$$y_{max} - y_i \leq 0. \quad (3.70)$$

Le regioni descritte, sono rappresentate in Figura 3.4, dove si vede come l'ostacolo abbia forma rettangolare con dimensioni dipendenti dal valore di x_{max} , x_{min} , y_{max} e y_{min} . Implementando la (3.66) al calcolatore, sostituendo i vincoli delle (3.67)-(3.70), si ottiene il plot di Figura 3.3b.

Per definire l'effetto repulsivo che il potenziale porta al nodo mobile, si devono calcolare le derivate di $p_r(x, y)$. Derivando rispetto la coordinata x_i si ottiene:

$$\begin{aligned} \frac{\partial p_r(x, y)}{\partial x_i} &= -\frac{1}{\left(\delta + \sum_{i=1}^I (g_i(x, y) + |g_i(x, y)|)\right)^2} \cdot \frac{\partial (g_i(x) + |g_i(x)|)}{\partial x_i} = \\ &= -p_r(x, y)^2 \left(\frac{|x_i - x_{max}|}{x_i - x_{max}} - \frac{|x_{min} - x_i|}{x_{min} - x_i} \right) \end{aligned} \quad (3.71)$$

e analogamente rispetto a y_i si ha:

$$\frac{\partial p_r(x, y)}{\partial y_i} = -p_r(x, y)^2 \left(\frac{|y_i - y_{max}|}{y_i - y_{max}} - \frac{|y_{min} - y_i|}{y_{min} - y_i} \right). \quad (3.72)$$

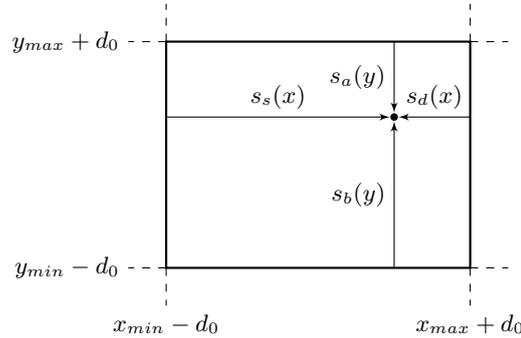


Figura 3.5: Calcolo della violazione dei vincoli.

Supponendo di utilizzare una distanza d_0 di sicurezza per ogni lato dell'ostacolo, restano valide le considerazioni fatte per ρ_0 . A differenza del caso circolare, dove quanto il vincolo sia stato violato è calcolabile facilmente tramite la distanza euclidea dal centro, per il vincolo rettangolare non è sufficiente. In particolare, è necessario definire:

$$s_a(y) = |y_i - y_{max} - d_0| \quad (3.73)$$

$$s_d(x) = |x_i - x_{max} - d_0| \quad (3.74)$$

$$s_b(y) = |y_{min} - y_i - d_0| \quad (3.75)$$

$$s_s(x) = |x_{min} - x_i - d_0| \quad (3.76)$$

dove $s_a(y)$ rappresenta la violazione del vincolo rispetto al segmento orizzontale superiore, $s_d(x)$ rispetto al segmento verticale destro, $s_b(x)$ rispetto al orizzontale basso e $s_s(y)$ rispetto al segmento verticale sinistro. Analizzando Figura 3.5, è facile intuire che non occorre considerare tutte le violazioni, ma è sufficiente considerare solo la coppia di violazioni verticale-orizzontale minime.

Una volta calcolate le derivate e le violazioni dei vincoli, si scrivono la matrice dei gradienti \mathbf{N}_p e il vettore dei vincoli attivi \mathbf{g}_p . Vengono successivamente integrate nell'algoritmo del gradiente proiettato utilizzando le (3.57)-(3.58) per la concatenazione.

Obstacles avoidance: sensor model

L'algoritmo dei potenziali artificiali, analizzato precedentemente, è utile per evitare facilmente ostacoli di cui è nota la posizione e la forma. Ad esempio, in Figura 4.4 vengono riportati i risultati di uno scenario in cui sono presenti unicamente ostacoli noti. Questa situazione può verificarsi quando i nodi mobili vengono impiegati per localizzare un emettitore collocato ad alta quota, sufficiente da non rendere pericolosi ostacoli sconosciuti come, ad esempio, veicoli e alberi. In questo scenario, quindi, gli unici ostacoli sono

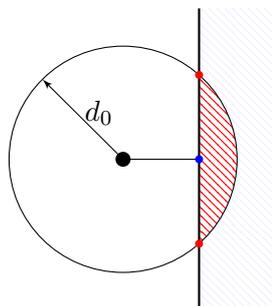


Figura 3.6: Rappresentazione del sensore.

rappresentati dagli edifici alti, dei quali è possibile conoscere a grandi linee la posizione e la topologia sfruttando informazioni a priori, come ad esempio quelle che si possono trovare nelle mappe elettroniche.

Il caso precedente rappresenta solo una piccola percentuale degli scenari che possono verificarsi in pratica. Infatti, i nodi mobili con molta probabilità dovranno muoversi nell'ambiente evitando anche degli ostacoli di cui non si è a conoscenza a priori. A tal proposito, è necessario che sia attivo un algoritmo per prevenire collisioni anche con questi oggetti sconosciuti. Per effettuare un sensing degli ostacoli nello spazio, è necessario che i droni siano dotati di sensori appositi. Nell'implementazione di un algoritmo di collision avoidance per ostacoli di cui non siano note posizione e forma, si considererà un sensore in grado di eseguire uno scan dell'ambiente a 360° , ad esempio con tecnologia laser. Per questo si può immaginare di descrivere una circonferenza centrata sul nodo con raggio d_0 rappresentante la distanza massima che il sensore può coprire nella rilevazione di ostacoli. In simulazione, si suppone che il sensore sia in grado di capire quali siano i punti in cui inizia a percepire l'ostacolo e la loro distanza, in modo da sfruttarli per calcolare dove si trova il punto più pericoloso dell'ostacolo nelle vicinanze. Un sensore così descritto è rappresentato in Figura 3.6, dov'è evidenziato il range del sensore e dove vengono evidenziati i punti rossi in cui viene percepito l'ostacolo e in blu il punto più vicino al nodo, quindi più critico per evitare la collisione. Viene anche evidenziata l'area in cui si ha la sovrapposizione tra l'ostacolo e la circonferenza di sensing.

Nel seguito di questa sezione si descriverà il modello di sensore per la creazione di un algoritmo di obstacle avoidance limitandosi al caso in cui gli ostacoli abbiano tutti forma rettangolare con lati paralleli agli assi cartesiani di riferimento. Questa semplificazione permette di simulare facilmente il comportamento del sensore, garantendo una simulazione a basso calcolo computazionale. Il punto critico e l'angolo d'impatto ottenuti dal sensore, vengono sfruttati per creare un comando da fornire ai nodi per correggere la traiettoria.

Il calcolo del punto critico, rappresentato dal punto dell'ostacolo più

vicino al nodo, viene fatto utilizzando i punti in cui la circonferenza del range del sensore e l'ostacolo si intersecano. In particolare si sfruttano i due punti estremi con coordinate $\mathbf{p}_{int_1} = [x_{int_1}, y_{int_1}]^T$ e $\mathbf{p}_{int_2} = [x_{int_2}, y_{int_2}]^T$, rappresentati in colore rosso in Figura 3.7. Le permutazioni delle coordinate di questi punti, vengono usate per creare i vertici di un rettangolo del quale è possibile calcolare la lunghezza di ciascun lato. I due lati vengono calcolati prendendo la differenza tra le coordinate massime e minime dei vertici:

$$l_x = \max(x_{int_1}, x_{int_2}) - \min(x_{int_1}, x_{int_2}) \quad (3.77)$$

$$l_y = \max(y_{int_1}, y_{int_2}) - \min(y_{int_1}, y_{int_2}) \quad (3.78)$$

e in base al valore che assumono si può calcolare facilmente il punto critico dell'ostacolo $\mathbf{p}_{c_i} = [x_c, y_c]^T$. Supponendo che il nodo si trovi nel punto $\mathbf{p}_i = [x_i, y_i]^T$, infatti, si possono distinguere vari casi:

- (a) se $l_x < l_y$ ci si trova nella situazione di Figura 3.7a, dove il nodo è situato lateralmente rispetto l'ostacolo, quindi il tratto più pericoloso dell'ostacolo è dato dal segmento verticale. Perciò, il punto critico avrà la stessa coordinata verticale del nodo, $y_c = y_i$. Il calcolo della coordinata orizzontale del punto critico è facilmente ottenibile calcolando le distanze tra il nodo e i quattro vertici del rettangolo ed andando a ricercare la distanza minima, alla quale è associata x_c . Quindi il punto critico, rappresentato in blu in Figura 3.7a, è dato da:

$$\mathbf{p}_{c_i} = \left[\arg \min_{x_{int_1}, x_{int_2}} d_i(\mathbf{p}_i, \mathbf{p}_{c_i}), y_i \right]^T. \quad (3.79)$$

- (b) nel caso in cui $l_x > l_y$, come raffigurato in Figura 3.7b, il nodo si trova sopra o sotto l'ostacolo rendendo il segmento critico orizzontale. Procedendo analogamente al caso precedente, il punto critico sarà dato da:

$$\mathbf{p}_{c_i} = \left[x_i, \arg \min_{y_{int_1}, y_{int_2}} d_i(\mathbf{p}_i, \mathbf{p}_{c_i}) \right]^T. \quad (3.80)$$

- (c) nel caso in cui i due lati siano approssimativamente uguali come in Figura 3.7c, significa che il nodo si trova nelle vicinanze di un vertice dell'ostacolo. Anche in questo caso, è necessario calcolare le distanze dai quattro vertici del rettangolo creato dalle intersezioni circonferenza-ostacolo, selezionando la coppia di coordinate che minimizzano $d_i(\mathbf{p}_i, \mathbf{p}_{c_i})$. Allora il punto critico sarà dato da:

$$\mathbf{p}_{c_i} = \left[\arg \min_{x_{int_1}, x_{int_2}} d_i(\mathbf{p}_i, \mathbf{p}_{c_i}), \arg \min_{y_{int_1}, y_{int_2}} d_i(\mathbf{p}_i, \mathbf{p}_{c_i}) \right]^T. \quad (3.81)$$

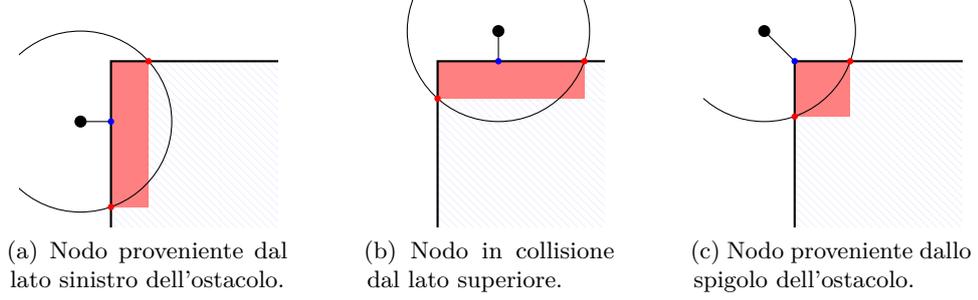


Figura 3.7: Rappresentazione dei rettangoli creati tramite le intersezioni del sensore con l'ostacolo.

Una volta ottenuto il punto critico \mathbf{p}_{c_i} , è possibile calcolare l'angolo d'impatto andando a valutare le differenze di coordinate col nodo per poi calcolare l'arcotangente. Considerando che il nodo può avere un angolo di impatto $\varphi_i \in [0^\circ, 360^\circ[$ è necessario definire l'operazione di arcotangente su quattro quadranti data da:

$$\arctan 2(x, y) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & x < 0, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & x < 0, y < 0 \\ +\frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \end{cases} \quad (3.82)$$

mentre non è definita nel caso in cui $x = y = 0$, ma non viene considerato in quanto questa condizione si verifica solo quando l'ostacolo è fuori dal range di sensing. Allora l'angolo di impatto sarà dato da:

$$\varphi_i = \arctan 2\left(\frac{y_i - y_c}{x_i - x_c}\right). \quad (3.83)$$

Una volta che la simulazione del sensore ha fornito il punto critico, la distanza da esso e l'angolo, si crea una legge di controllo per l'UAV in modo da correggere la traiettoria ed evitare la collisione con l'ostacolo. Per far sì che il movimento verso il punto critico sia proibito, si può comandare il drone per fargli assumere le posizioni:

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} - \gamma_i^{(k)}(d_i(\mathbf{p}_i, \mathbf{p}_{c_i}))\mathbf{u}_i^{(k)} \quad (3.84)$$

dove $\mathbf{u}_i^{(k)}$ è il comando di correzione che fa muovere il drone in direzione opposta all'ostacolo ed è dato da:

$$\mathbf{u}_i^{(k)} = \begin{bmatrix} \cos(\varphi_i) \\ \sin(\varphi_i) \end{bmatrix}. \quad (3.85)$$

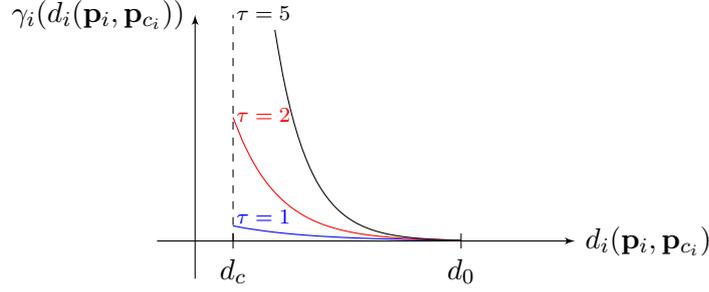


Figura 3.8: Funzione potenziale $\gamma_i(d_i(\mathbf{p}_i, \mathbf{p}_{c_i})) = \rho e^{-\tau(d_i(\mathbf{p}_i, \mathbf{p}_{c_i}) - d_0)}$ per il calcolo dei coefficienti di correzione con $\rho = 0.01$ e τ variabile.

Il coefficiente $\gamma_i^{(k)}$ permette di pesare l'intensità del comando di correzione. Nel caso più semplice, essa può essere considerata unitaria, avendo così una correzione costante senza considerare quanto il nodo sia vicino all'ostacolo.

Un altro approccio è quello di creare una funzione dalla quale calcolare i coefficienti in base a quanto il nodo è a rischio di collisione, basandosi sulla distanza dal punto critico. Ad esempio, si può utilizzare una funzione generatrice di tipo esponenziale:

$$\gamma_i^{(k)}(d_i(\mathbf{p}_i, \mathbf{p}_{c_i})) = \rho e^{-\tau(d_i(\mathbf{p}_i, \mathbf{p}_{c_i}) - d_0)} \quad (3.86)$$

dove ρ è una costante moltiplicativa piccola in modo da avere un effetto “cuscinato” quando l'ostacolo è di poco all'interno del range del sensore, mentre τ è un parametro che permette di rendere più veloce la crescita dell'esponenziale, quindi provoca un aumento dei coefficienti di correzione al diminuire della distanza tra nodo ed ostacolo. I coefficienti sono rappresentati in Figura 3.8 al variare del parametro τ : si vede come la funzione si comporti in modo da avere una leggera correzione quando l'ostacolo si trova a distanza sufficientemente sicura mentre la correzione diventa più intensa quando il nodo è molto vicino alla collisione. In simulazione si è visto che il calcolo dei coefficienti attraverso la funzione esponenziale permette di avere traiettorie complessivamente più lineari, presentando meno spikes nelle traiettorie dovuti a movimenti bruschi.

Analizzando il caso più generale con il dispiegamento di n nodi in uno spazio in cui vi siano n_{obs} ostacoli a distanza tale da essere percepiti dai sensori, i ragionamenti di cui sopra restano validi. In particolare, dato il vettore colonna delle posizioni dei nodi \mathbf{p} di dimensioni $2n$, si calcola la matrice dei punti critici \mathbf{p}_c di dimensione $2n \times n_{obs}$. Noti \mathbf{p} e \mathbf{p}_c , è immediato calcolare la matrice delle distanze $\mathbf{d}(\mathbf{p}, \mathbf{p}_c)$ e la matrice degli angoli d'impatto $\varphi(\mathbf{p}, \mathbf{p}_c)$, entrambe di dimensione $n \times n_{obs}$. Calcolando il vettore contenente i coefficienti peso $\gamma(\mathbf{d}(\mathbf{p}, \mathbf{p}_c))$, basandosi sulla distanza che ciascun nodo vede rispetto l'ostacolo più vicino ed associando alle coordinate dello stesso nodo il medesimo coefficiente. Quest'ultimo vettore ha dimensioni $2n$ e il

comando degli UAV è dato da:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \gamma^{(k)}(\mathbf{d}(\mathbf{p}, \mathbf{p}_c)) \odot \mathbf{u}^{(k)} \quad (3.87)$$

dove il simbolo \odot rappresenta il prodotto di Hadamard e il comando di correzione $\mathbf{u}^{(k)}$ è dato da:

$$\mathbf{u}^{(k)} = \left[u_{x_1}, u_{y_1}, u_{x_2}, u_{y_2}, \dots, u_{x_n}, u_{y_n} \right]^T \quad (3.88)$$

dove sono state introdotte le componenti delle correzioni:

$$\mathbf{u}_x^{(k)} = \begin{bmatrix} \sum_{t=1}^{n_{obs}} \cos \theta_{1,t} \\ \vdots \\ \sum_{t=1}^{n_{obs}} \cos \theta_{n,t} \end{bmatrix}, \quad \mathbf{u}_y^{(k)} = \begin{bmatrix} \sum_{t=1}^{n_{obs}} \sin \theta_{1,t} \\ \vdots \\ \sum_{t=1}^{n_{obs}} \sin \theta_{n,t} \end{bmatrix}. \quad (3.89)$$

Le sommatorie si rendono necessarie per sommare per riga i contributi che ciascun ostacolo porta al nodo i -esimo.

Un'alternativa elegante alla legge di comando della (3.88) è rappresentata dal metodo del gradiente proiettato. È infatti possibile sfruttare la distanza tra il nodo i -esimo e il punto critico \mathbf{p}_c fornita dal sensore per calcolare la matrice dei gradienti relativi ai vincoli di distanza percepiti dal “sensore” \mathbf{N}_o e il vettore della violazione dei vincoli \mathbf{g}_o . Note queste due quantità, effettuando una concatenazione come già descritto nelle (3.57) e (3.58), è possibile integrare il sensing del sensore al gradiente proiettato che calcolerà il comando di correzione. Nell'implementazione dell'algoritmo di collision avoidance per ostacoli non noti si è scelto di utilizzare questa soluzione, moltiplicando i coefficienti $\gamma(\mathbf{d}(\mathbf{p}, \mathbf{p}_c))$ alla matrice \mathbf{N}_o per ottenere un effetto smooth e una minor oscillazione degli UAV attorno agli ostacoli. È, infine, importante notare che l'utilizzo di coefficienti γ_i troppo elevati può portare a matrici “badly-scaled” rendendo l'operazione di inversione poco accurata. In questo caso, il metodo della proiezione del gradiente non può considerarsi sicuro, per cui è uno scenario da evitare.

Controllo della velocità

Per considerare i vincoli fisici legati alle prestazioni degli UAV, è necessario aggiungere un vincolo legato alla massima velocità possibile durante una singola iterazione. Questo vincolo viene implementato nel simulatore per renderlo il più fedele possibile alla realtà ma nella pratica è possibile ometterlo in quanto è intrinsecamente presente nella dinamica di spostamento dei droni.

Nel caso in cui lo spostamento abbia modulo maggiore del massimo consentito, che esso sia dovuto ad un limite di prestazione del drone o per scelta, l'algoritmo di controllo deve scalare le componenti della velocità.

Definendo il termine relativo al comando dell' i -esimo UAV ottenuto tramite l'algoritmo di navigazione con $\mathbf{u}_i^{(k)} = \left[u_{x_i}^{(k)} \hat{\mathbf{i}} + u_{y_i}^{(k)} \hat{\mathbf{j}} \right]$, dove $\hat{\mathbf{i}}$ e $\hat{\mathbf{j}}$ sono

i versori corrispondenti rispettivamente alle coordinate x e y , è possibile esprimere l'equazione di update con la (2.55):

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \mathbf{u}_i^{(k)} + \mathbf{w}_i^{(k)} \quad (3.90)$$

da cui è immediato vedere che:

$$\mathbf{u}_i^{(k)} = \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} - \mathbf{w}_i^{(k)} \quad (3.91)$$

corrisponde ad una velocità al quale si sovrappone un rumore di transizione. Per fare in modo che l'UAV si muova rispettando una velocità massima, si definisce un nuovo comando di update $\tilde{\mathbf{u}}_i^{(k)}$:

$$\tilde{\mathbf{u}}_i^{(k)} = \begin{cases} \mathbf{u}_i^{(k)}, & \text{se } |\mathbf{u}_i^{(k)}| \leq v_{lim_i} \\ \mathbf{u}_i^{(k)} \frac{v_{lim_i}}{|\mathbf{u}_i^{(k)}|}, & \text{se } |\mathbf{u}_i^{(k)}| > v_{lim_i}. \end{cases} \quad (3.92)$$

dove v_{lim_i} corrisponde alla massima velocità consentita all' i -esimo UAV, in generale diverso per ogni drone. In questo modo, sostituendo il nuovo comando di update alla (3.90) si garantisce che gli UAV si spostino con una velocità limitata in modulo, lasciando immutato l'angolo di spostamento.

Un altro possibile approccio per il controllo della velocità è quello di effettuare sempre l'operazione della (3.92) applicando:

$$\tilde{\mathbf{u}}_i^{(k)} = \mathbf{u}_i^{(k)} \frac{v_{lim_i}}{|\mathbf{u}_i^{(k)}|}. \quad (3.93)$$

ad ogni UAV. In questo caso, l'algoritmo modifica il suo comportamento portando tutti gli UAV a velocità massima, limitando l'effetto del rallentamento dovuto al miglioramento delle posizioni e al diminuire dell'intensità del gradiente della funzione da minimizzare. Nei risultati di simulazione riportati in Capitolo 4 e Capitolo 5 sono state utilizzate velocità limite costanti ed uguali per tutti gli UAV, ovvero $v_{lim_i} = v_{lim}$, a parte la sezione riguardante le simulazioni in scenari NLOS.

3.2.4 Equazione di controllo vincolato

Alla luce dei vincoli introdotti nelle sezioni precedenti, è ora possibile affrontare il modello a varianza di stima dipendente dalla distanza descritto in Sezione 2.2. Riportando per comodità la formulazione del PEB per questo scenario rappresentata dalla (2.46):

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n A_i \left(\frac{x_i - x_t}{d_i}\right)^2 + \sum_{i=1}^n A_i \left(\frac{y_i - y_t}{d_i}\right)^2 - \left(\sum_{i=1}^n A_i \frac{(x_i - x_t)(y_i - y_t)}{d_i^2}\right)^2}} \quad (3.94)$$

è facile intuire che, a causa della stretta dipendenza dalla distanza dei coefficienti A_i , gli UAV tendono muoversi verso l'emettitore per ridurre la distanza e migliorare la qualità della misura. Se non fossero presenti dei vincoli di collision avoidance, gli UAV mobili tenderebbero a collidere con l'emettitore o con l'edificio all'interno del quale è situato. Per questa ragione, il modello con varianza di stima dipendente dalla distanza richiede necessariamente un algoritmo di controllo vincolato. Utilizzando tutti i vincoli discussi nelle sezioni precedenti, la (3.2) valida in generale, per il caso in esame vale:

$$\begin{aligned} \mathbf{p}^* &= \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \text{PEB}(\mathbf{p}_t, \mathbf{p}) \\ \text{tale che} \quad & \begin{aligned} 1) \quad & d_{ij} \geq d_{min}, \quad i, j = 1, \dots, n, \quad i \neq j \\ 2) \quad & d_{it} \geq d_{mint}, \quad \forall i \\ 3) \quad & \mathcal{T}_i \cap \mathcal{O} = \emptyset, \quad \forall i \\ 4) \quad & \left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|_2 = v, \quad \forall i \end{aligned} \end{aligned} \quad (3.95)$$

dove:

- 1) corrisponde al vincolo di mantenimento della distanza minima tra UAV;
- 2) è il vincolo di distanza minima tra nodi e target;
- 3) implica che l'intersezione tra la traiettoria \mathcal{T}_i del nodo i -esimo e il set di coordinate degli ostacoli presenti nell'ambiente \mathcal{O} fornisca l'insieme vuoto;
- 4) è il controllo per far mantenere una velocità costante agli UAV.

Formalmente, la (3.95) può essere espressa dall'equazione vettoriale:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \boldsymbol{\varepsilon}^{(k)} \odot \left(\mathbf{P} \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \right) - \mathbf{N} \left(\mathbf{N}^T \mathbf{N}^{-1} \right) \mathbf{g} \left(\mathbf{p}^{(k)} \right) \quad (3.96)$$

dove si possono distinguere i termini:

- $\mathbf{P} \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\mathbf{p}_t; \mathbf{p})$ corrispondente alla proiezione nello spazio tangente ai vincoli del gradiente per la minimizzazione del PEB;
- $\mathbf{N} \left(\mathbf{N}^T \mathbf{N}^{-1} \right) \mathbf{g} \left(\mathbf{p}^{(k)} \right)$ rappresentante la restoration move dovuta ai vincoli non lineari in esame, dove \mathbf{N} è la matrice dei gradienti dei vincoli attivi, ottenuta concatenando i gradienti come nella (3.57), e $\mathbf{g} \left(\mathbf{p}^{(k)} \right)$ è il vettore contenente la violazione dei vincoli, ottenuto concatenando le singole violazioni tramite la (3.58).

Il calcolo del gradiente, pur mantenendo una forma simile, è diverso rispetto al caso in cui la varianza della stima delle distanze sia costante e non dipenda dalla distanza. Omettendo per comodità di notazione le dipendenze

dei fattori da \mathbf{p}_i e \mathbf{p}_t , ma ricordando che sia numeratore che denominatore del PEB dipendono dalla coordinata di derivazione, la derivata rispetto la coordinata x_i è data da:

$$\begin{aligned} \frac{\partial \text{PEB}}{\partial x_i} &= \frac{1}{2\text{PEB}} \frac{\partial}{\partial x_i} \left(\frac{\text{N}_{\text{PEB}}}{\text{D}_{\text{PEB}}} \right) = \\ &= \frac{1}{2\text{PEB}} \left[\frac{1}{\text{D}_{\text{PEB}}} \frac{\partial \text{N}_{\text{PEB}}}{\partial x_i} + \text{N}_{\text{PEB}} \frac{\partial}{\partial x_i} \left(\frac{1}{\text{D}_{\text{PEB}}} \right) \right] = \\ &= \frac{1}{2\text{PEB}} \frac{1}{\text{D}_{\text{PEB}}} \left(\frac{\partial \text{N}_{\text{PEB}}}{\partial x_i} - \text{PEB}^2 \frac{\partial \text{D}_{\text{PEB}}}{\partial x_i} \right). \end{aligned} \quad (3.97)$$

La derivata del numeratore viene calcolata utilizzando l'espressione della (2.44) dei coefficienti A_i , riportata per comodità:

$$A_i = \frac{\alpha^2}{2d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha} \quad (3.98)$$

e vale:

$$\frac{\partial \text{N}_{\text{PEB}}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\sum_{i=1}^n A_i \right) = \frac{\partial A_i}{\partial x_i} = -\frac{\alpha \bar{x}}{d_i^2} \left(\frac{\alpha}{d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha} \right) \quad (3.99)$$

dove $\bar{x} = x_i - \hat{x}_t$, $\bar{y} = y_i - \hat{y}_t$ e $d_i = \|\hat{\mathbf{p}}_t - \mathbf{p}_i\|_2$. Dalla (3.99), si può facilmente verificare che per $\alpha = 0$, quindi nel caso in cui la varianza non dipenda dalla distanza ($A_i = 1/\sigma_0^2$), la derivata del numeratore si annulla ottenendo esattamente l'espressione della (3.9). La derivata del denominatore della (2.46) è data da:

$$\frac{\partial \text{D}_{\text{PEB}}}{\partial x_i} = \text{D}_{1x} \left(\sum_{i=1}^n A_i \frac{\bar{y}^2}{d_i^2} \right) + \text{D}_{2x} \left(\sum_{i=1}^n A_i \frac{\bar{x}^2}{d_i^2} \right) - \text{D}_{3x} \quad (3.100)$$

dove i termini D_{1x} , D_{2x} e D_{3x} corrispondono alle derivate dei singoli componenti del denominatore e valgono:

$$\text{D}_{1x} = \frac{\partial}{\partial x_i} \left(\sum_{i=1}^n A_i \frac{\bar{x}^2}{d_i^2} \right) = \frac{\bar{x}}{d_i^2} \left(\bar{x} \frac{\partial A_i}{\partial x_i} + 2A_i \frac{\bar{y}^2}{d_i^2} \right) \quad (3.101)$$

$$\text{D}_{2x} = \frac{\partial}{\partial x_i} \left(\sum_{i=1}^n A_i \frac{\bar{y}^2}{d_i^2} \right) = \frac{\bar{y}^2}{d_i^2} \left(\frac{\partial A_i}{\partial x_i} - 2A_i \frac{\bar{x}}{d_i^2} \right) \quad (3.102)$$

$$\text{D}_{3x} = \frac{\partial}{\partial x_i} \left[\left(\sum_{i=1}^n A_i \frac{\bar{x}\bar{y}}{d_i^2} \right)^2 \right] = 2 \left(\sum_{i=1}^n A_i \frac{\bar{x}\bar{y}}{d_i^2} \right) \frac{\bar{y}}{d_i^2} \left[\bar{x} \frac{\partial A_i}{\partial x_i} + \frac{A_i}{d_i^2} (\bar{y}^2 - \bar{x}^2) \right]. \quad (3.103)$$

Anche all'interno dell'algoritmo, si è scelto di utilizzare queste formulazioni senza andare ad espandere la derivata $\partial A_i / \partial x_i$, in quanto essa viene già calcolata per la derivata del numeratore e riutilizzarla per la derivata del

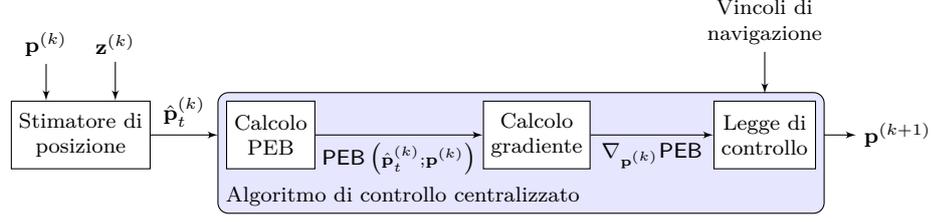


Figura 3.9: Schema a blocchi per localizzazione e navigazione congiunta degli UAV tramite un algoritmo centralizzato.

denominatore non porta ad un ulteriore costo computazionale. Sostituendo le ultime equazioni nella (3.100) si ottiene:

$$\begin{aligned} \frac{\partial D_{\text{PEB}}}{\partial x_i} &= \frac{\bar{x}}{d_i^2} \left(\bar{x} \frac{\partial A_i}{\partial x_i} + 2A_i \frac{\bar{y}^2}{d_i^2} \right) \left(\sum_{i=1}^n A_i \frac{\bar{y}^2}{d_i^2} \right) + \\ &+ \frac{\bar{y}^2}{d_i^2} \left(\frac{\partial A_i}{\partial x_i} - 2A_i \frac{\bar{x}}{d_i^2} \right) \left(\sum_{i=1}^n A_i \frac{\bar{x}^2}{d_i^2} \right) - \\ &- 2 \left(\sum_{i=1}^n A_i \frac{\bar{x}\bar{y}}{d_i^2} \right) \frac{\bar{y}}{d_i^2} \left[\bar{x} \frac{\partial A_i}{\partial x_i} + \frac{A_i}{d_i^2} (\bar{y}^2 - \bar{x}^2) \right]. \end{aligned} \quad (3.104)$$

Allo stesso modo si ottiene la derivata del PEB rispetto la coordinata y_i che è formalmente uguale alla (3.97) ma la derivata del numeratore è data da:

$$\frac{\partial N_{\text{PEB}}}{\partial y_i} = \frac{\partial}{\partial y_i} \left(\sum_{i=1}^n A_i \right) = \frac{\partial A_i}{\partial y_i} = -\frac{\alpha \bar{y}}{d_i^2} \left(\frac{\alpha}{d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha} \right) \quad (3.105)$$

mentre la derivata del denominatore vale:

$$\begin{aligned} \frac{\partial D_{\text{PEB}}}{\partial y_i} &= \frac{\bar{x}^2}{d_i^2} \left(\frac{\partial A_i}{\partial y_i} - 2A_i \frac{\bar{y}}{d_i^2} \right) \left(\sum_{i=1}^n A_i \frac{\bar{y}^2}{d_i^2} \right) + \\ &+ \frac{\bar{y}}{d_i^2} \left(\bar{y} \frac{\partial A_i}{\partial y_i} + 2A_i \frac{\bar{x}^2}{d_i^2} \right) \left(\sum_{i=1}^n A_i \frac{\bar{x}^2}{d_i^2} \right) - \\ &- 2 \left(\sum_{i=1}^n A_i \frac{\bar{x}\bar{y}}{d_i^2} \right) \frac{\bar{x}}{d_i^2} \left[\bar{y} \frac{\partial A_i}{\partial y_i} + \frac{A_i}{d_i^2} (\bar{x}^2 - \bar{y}^2) \right]. \end{aligned} \quad (3.106)$$

L'algoritmo, nella sua versione centralizzata, è riportato in Figura 3.9 sotto forma di schema a blocchi. Viene evidenziato il blocco di stima della posizione del target posto a monte del macro-blocco dell'algoritmo di controllo.

3.3 Algoritmo distribuito

Si considera una rete dinamica composta da n nodi mobili e ci si riferisce all' i -esimo drone situato alla posizione $\mathbf{p}_i^{(k)} = [x_i^{(k)}, y_i^{(k)}]^T$ assunta nota per tutti i time-slot $k = 1, \dots, K$, grazie al GPS. A differenza del caso centralizzato, dove le posizioni dei nodi erano note a tutti, questa posizione è nota solo al nodo i -esimo. Ponendo vincoli di latenza, si ipotizza che la posizione $\mathbf{p}_i^{(k)}$ sia nota a tutti i gli UAV raggiungibili con un single-hop.

In Sezione 3.3.1 vengono descritti gli aspetti legati al modello adottato per la simulazione del raggio di copertura degli UAV, in Sezione 3.3.2 si forniscono le basi di teoria dei grafi sfruttate per lo studio di un algoritmo per l'instaurazione dei collegamenti tra i nodi mentre in Sezione 3.3.3 viene espressa l'equazione di ottimizzazione per l'algoritmo di posizionamento distribuito per il modello a varianza di errore di stima dipendente dalla distanza.

3.3.1 Modelli di connettività

Per simulare la capacità e la qualità di sensing vengono utilizzate dei modelli matematici di *sensor coverage*. Essi sono dei modelli astratti che cercano di quantificare quanto un sensore sia in grado di percepire un fenomeno fisico come una trasmissione di un'onda elettromagnetica, ovvero quanto un sensore copra una regione. Nella maggior parte dei casi, questi modelli sono matematicamente formalizzati in funzioni di copertura a partire da distanze ed angoli. In letteratura viene fatta una classificazione basata su modelli:

- i) deterministici, come in Figura 3.10a, che si basano unicamente su aspetti geometrici;
- ii) statistici, rappresentati in Figura 3.10b, più complessi e che considerano anche il rumore.

Il modello più semplice, che verrà utilizzato nel seguito della tesi, è deterministico definito *modello Booleano a disco*. Definendo la distanza tra due UAV con $d_{ij}^{(k)} = \|\mathbf{p}_i^{(k)} - \mathbf{p}_j^{(k)}\|_2$, la probabilità che due nodi siano connessi, supponendo che tutti abbiano lo stesso *transmission range* d_{hop} , è data da:

$$C(d_{ij}^{(k)}(\mathbf{p}_i^{(k)}, \mathbf{p}_j^{(k)}), d_{\text{hop}}) = \begin{cases} 1, & d_{ij}^{(k)}(\mathbf{p}_i^{(k)}, \mathbf{p}_j^{(k)}) \leq d_{\text{hop}} \\ 0, & d_{ij}^{(k)}(\mathbf{p}_i^{(k)}, \mathbf{p}_j^{(k)}) > d_{\text{hop}}. \end{cases} \quad (3.107)$$

Questo modello non tiene in considerazione gli angoli, in quanto si suppone che la copertura di ogni singolo nodo sia omnidirezionale, così che il j -esimo UAV all'interno del disco sia connesso all' i -esimo UAV a prescindere

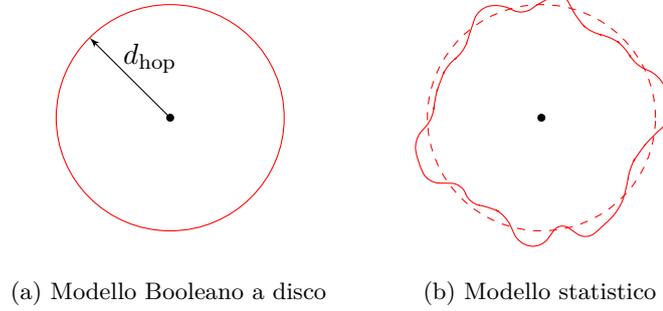


Figura 3.10: Rappresentazione grafica dei modelli di coverage.

dall'angolo formato. In letteratura sono presenti molti altri modelli più aderenti alla realtà, analizzati nel dettaglio in [61, 62].

Utilizzando questo semplice modello, il raggio del disco vale d_{hop} e si indica con:

$$\mathcal{R}_i^{(k)} = \left\{ \mathbf{p}_j^{(k)} : \left\| \mathbf{p}_i^{(k)} - \mathbf{p}_j^{(k)} \right\|_2 \leq d_{\text{hop}}, j \neq i \right\} \quad (3.108)$$

il set di nodi raggiungibili con un single-hop (*neighbors*) dall' i -esimo UAV al time slot k . Per tutti gli altri nodi posti a distanza maggiore, a causa dei vincoli di latenza, le posizioni degli UAV sono ottenute tramite comunicazione multi-hop con un ritardo di $h_{ij}^{(k)}$ time slot, dove $h_{ij}^{(k)}$ è il numero di hop tra la coppia di nodi i, j all'iterazione k . A causa di questo scambio di informazioni effettuate in un tempo finito, c'è la possibilità che alcune posizioni siano parzialmente non aggiornate (out-of-date).

3.3.2 Fondamenti di teoria dei grafi

Per l'instaurazione della rete di collegamenti, si rende necessario implementare un algoritmo in grado di verificare le distanze mutue tra i nodi così da classificare il tipo di link e uno strumento molto utile per questo scopo è la teoria dei grafi [63, 64].

Un grafo $\mathcal{G}^{(k)} (\mathcal{V}^{(k)}, \mathcal{E}^{(k)})$ consiste in un set non vuoto di elementi chiamati vertici, ai quali sono associate le posizioni degli UAV:

$$\mathcal{V}^{(k)} = \left\{ v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)} \right\} = \left\{ \mathbf{p}_1^{(k)}, \dots, \mathbf{p}_i^{(k)}, \dots, \mathbf{p}_n^{(k)} \right\} \quad (3.109)$$

e un set finito e non vuoto $\mathcal{E}^{(k)} (\mathcal{V}^{(k)})$ di collegamenti (*edges*) tra coppie non ordinate di elementi distinti di $\mathcal{V}^{(k)}$:¹

$$\mathcal{E}^{(k)} = \left\{ \dots, \left(v_u^{(k)}, v_w^{(k)} \right), \dots \right\}. \quad (3.110)$$

¹Nel seguito si utilizzerà le notazioni compatte $\mathcal{G}^{(k)} (\mathcal{V}^{(k)}, \mathcal{E}^{(k)}) = \mathcal{G}^{(k)}$ e $\mathcal{E}^{(k)} (\mathcal{V}^{(k)}) = \mathcal{E}^{(k)}$.

Dati i vertici, è possibile generare una famiglia di $\mathcal{E}^{(k)}$ creando link tra nodi diversi basandosi sul criterio legato all'applicazione in esame.

Due vertici $v_u^{(k)}$, $v_w^{(k)}$ del grafo $\mathcal{G}^{(k)}$ si definiscono *adiacenti* se esiste un collegamento tra di loro, ovvero se all'interno del set $\mathcal{E}^{(k)}$ è presente la coppia $(v_u^{(k)}, v_w^{(k)})$. Per analizzare l'adiacenza fra nodi, si definisce la matrice adiacenza $\mathcal{A}^{(k)} \in \mathbb{R}^{n \times n}$ dove, nel caso in cui il collegamento sia bidirezionale e non vengano applicati dei pesi ai links, per la coppia di nodi u , w , si ha:

- $a_{uw}^{(k)} = 1$ se sono connessi (adiacenti);
- $a_{uw}^{(k)} = 0$ se non sono connessi (non sono adiacenti).

Una matrice così definita ha proprietà di simmetria rispetto alla diagonale principale, che viene posta nulla per non tenere conto del collegamento tra il nodo e se stesso, esprimendo formalmente $a_{uu}^{(k)} = 0$. Considerando il caso multi-hop, per tenere traccia del numero di hop necessari per un collegamento, si definiscono h_{\max} matrici di adiacenza $\mathcal{A}_1^{(k)}$, $\mathcal{A}_2^{(k)}$, \dots , $\mathcal{A}_{h_{\max}}^{(k)}$, corrispondenti all' h -esimo salto dove h_{\max} è il massimo numero di hop consentiti e k è il time slot corrispondente. Selezionando unicamente il link che percorrono meno distanza per collegare la generica coppia u , w , le matrici di adiacenza non hanno coefficienti $a_{uw}^{(k)} = 1$ sovrapposti, per cui la matrice adiacenza complessiva può essere ottenuta sommando le singole matrici:

$$\mathcal{A}^{(k)} = \sum_{h=1}^{h_{\max}} \mathcal{A}_h^{(k)} \quad (3.111)$$

e sarà anch'essa caratterizzata unicamente da coefficienti $a_{uw} = \{0, 1\}$. Per semplicità, non essendo l'oggetto principale di questa tesi, l'algoritmo di instaurazione della rete implementato può essere impostato per creare network scegliendo un numero massimo di hop $h_{\max} \in [1, 2, 3]$.

È possibile quantificare il numero di collegamenti attivi per ogni vertice andando a definire la matrice grado del grafo:

$$\mathcal{D}^{(k)} = \text{diag} \left(d_{\mathcal{D}1}^{(k)}, d_{\mathcal{D}2}^{(k)}, \dots, d_{\mathcal{D}n}^{(k)} \right) \quad (3.112)$$

dove il termine corrispondente al vertice $v_u^{(k)}$ vale

$$d_{\mathcal{D}u}^{(k)} = \sum_{\substack{w=1 \\ w \neq u}}^n a_{uw}^{(k)}, \quad \forall u \quad (3.113)$$

e rappresenta il numero di collegamenti dell' u -esimo nodo, ottenibile dalla matrice adiacenza sommando l' u -esima riga/colonna.

Per valutare la connettività di un grafo $\mathcal{G}^{(k)}$, dalle due precedenti matrici è possibile derivare la matrice Laplaciana, data da:

$$\mathcal{L}^{(k)} = \mathcal{D}^{(k)} - \mathcal{A}^{(k)} \quad (3.114)$$

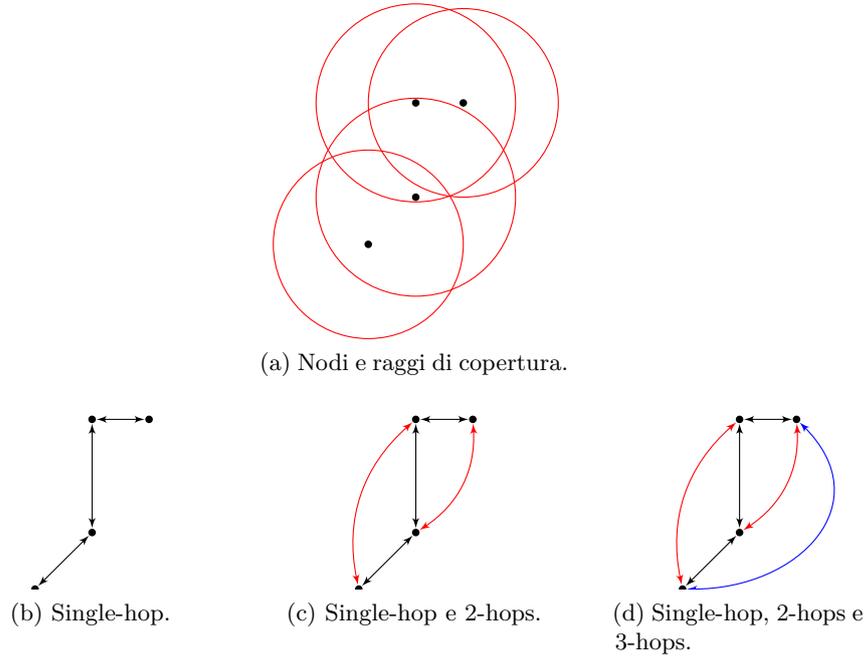


Figura 3.11: Collegamenti tra nodi per single-hop e multi-hop.

e il generico coefficiente l_{uw} vale:

$$l_{uw} = \begin{cases} d_{\mathcal{D}u}^{(k)}, & \text{se } u = w \\ -1, & \text{se } u \text{ e } w \text{ sono adiacenti} \\ 0, & \text{altrimenti.} \end{cases} \quad (3.115)$$

Siccome la matrice $\mathcal{L}^{(k)}$ è caratterizzata dalla proprietà di essere simmetrica e semi-definita positiva, utilizzando la caratterizzazione variazionale degli autovalori in termini del quoziente di Rayleigh [65–67], è possibile dimostrare che gli autovalori di $\mathcal{L}^{(k)}$ sono non-negativi. Il set degli autovalori λ_i è solitamente chiamato *spettro* di $\mathcal{L}^{(k)}$ o *spettro associato al grafo* $\mathcal{G}^{(k)}$, e la proprietà di non-negatività è formalmente espressa da:

$$0 = \lambda_1 \leq \lambda_1 \leq \dots \leq \lambda_n \quad (3.116)$$

Si dimostra che lo spettro di \mathcal{L} ha p autovalori nulli che corrispondono al numero di sotto-grafi disconnessi. Quindi, se $\mathcal{G}^{(k)}$ è connesso allora la molteplicità dell'autovalore nullo dovrà essere $p = 1$ e:

$$\lambda_1 = 0, \quad \lambda_2 > 0 \quad (3.117)$$

Applicando la teoria dei grafi ad una network di UAV, assegnando a ciascuno di essi un transmission range (*range assignment*) per il riconoscimento degli altri nodi per la creazione dei link, si rientra nel caso di grafo

di comunicazione o *communication graph*. Ipotizzando di utilizzare il modello Booleano a disco descritto in Sezione 3.3.1, supponendo per semplicità $d_{\text{hop}_i} = d_{\text{hop}}$ per ogni UAV, si rientra nel caso in cui il range assignment è d_{hop} -omogeneo. In generale, nello scenario in cui d_{hop} sia diverso da nodo a nodo, tutti i nodi avranno matrici adiacenze diverse in base al proprio transmission range e (3.111) deve essere calcolata da ogni UAV:

$$\mathcal{A}_i^{(k)} = \sum_{h=1}^{h_{max}} \mathcal{A}_{i_h}^{(k)} \quad (3.118)$$

con risultati che tenderanno ad uniformarsi all'aumentare di d_{hop} . Basandosi su questo modello, tutti i nodi collocati all'interno del cerchio di raggio d_{hop} centrato nell' i -esimo nodo possono effettuare uno scambio single-hop di informazioni e la matrice adiacenza per un hop è data da (3.108):

$$\mathcal{A}_{i_1}^{(k)} = \mathcal{R}_i^{(k)} = \left\{ \mathbf{p}_j^{(k)} : \left\| \mathbf{p}_i^{(k)} - \mathbf{p}_j^{(k)} \right\|_2 \leq d_{\text{hop}}, j \neq i \right\}. \quad (3.119)$$

Con questo modello, disponendo alcuni nodi in modo casuale, si ottiene la rappresentazione grafica di Figura 3.11a, dove sono rappresentati in rosso i cerchi di raggio d_{hop} per tutti i nodi. In Figura 3.11b viene rappresentata la rete in esempio considerando i soli collegamenti single-hop. In questo modo la rete non è completamente connessa, per cui sarebbe necessario aumentare il numero di hop per fare in modo che la connettività sia completa. Si mostra, infatti, in Figura 3.11c la rete con collegamenti diretti e 2-hops, dove si vede che il numero di nodi raggiunti aumenta. Infine, in Figura 3.11d, si mostra la rete con collegamenti fino a 3-hop. Così facendo, tutti i nodi siano in grado di raggiungere tutte le destinazioni possibili, rendendo la rete completamente connessa.

3.3.3 Equazione di controllo distribuito

Ipotizzando di utilizzare il modello a varianza di errore di stima dipendente dalla distanza calcolato sfruttando la memorizzazione delle ultime posizioni note all' i -esimo UAV, la formulazione del PEB per il caso distribuito è dato da (2.70):

$$\text{PEB}_i \left(\mathbf{p}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sqrt{\frac{A_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)}}{S_{x,i}^{(k)} S_{y,i}^{(k)} - \left(S_{xy,i}^{(k)} \right)^2}} \quad (3.120)$$

dove:

$$S_{x,i}^{(k)} = A_i^{(k)} \left(c_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(c_i^{(k-h_{ij}+1)} \right)^2 \quad (3.121)$$

$$S_{y,i}^{(k)} = A_i^{(k)} \left(s_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(s_i^{(k-h_{ij}+1)} \right)^2 \quad (3.122)$$

$$S_{xy,i}^{(k)} = A_i^{(k)} c_i^{(k)} s_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} c_i^{(k-h_{ij}+1)} s_i^{(k-h_{ij}+1)} \quad (3.123)$$

e $\mathbf{q}_i^{(k)}$ è il vettore contenente le posizioni degli UAV note all' i -esimo nodo. La maggior differenza rispetto alla formulazione centralizzata risiede nel fatto che ciascun UAV deve calcolare il PEB, che in generale è differente per tutti, basandosi su posizioni che possono essere ritardate a causa della latenza. Utilizzando questa equazione risolutiva, ogni UAV utilizza posizioni relative a $h_{ij} - 1$ slot temporali precedenti, per tenere conto della propagazione delle informazioni limitate ad un solo salto per time slot per vincoli di latenza.

Applicando tutti i vincoli discussi nelle sezioni precedenti, (3.95) valida per il caso centralizzato, viene modificata:

$$\begin{aligned} \mathbf{p}_i^* &= \arg \min_{\mathbf{p}_i \in \mathbb{R}^2} \text{PEB}_i \left(\hat{\mathbf{p}}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) \\ \text{tale che} \quad & 1) \ d_{ij} \geq d_{min}, \ j = 1, \dots, n, \ j \neq i \\ & 2) \ d_{it} \geq d_{min_t} \\ & 3) \ \mathcal{T}_i \cap \mathcal{O} = \emptyset \\ & 4) \ \left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|_2 = v \end{aligned} \quad (3.124)$$

dove la notazione $\hat{\mathbf{p}}_{t_i}^{(k)}$ indica che l' i -esimo UAV utilizza una propria stima della posizione del target che, in generale, è diversa per tutti i nodi impiegati. Formalmente, (3.124) può essere espressa dall'equazione vettoriale:

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} - \varepsilon_i^{(k)} \left(\mathbf{P} \nabla_{\mathbf{q}_i^{(k)}} \text{PEB}_i \left(\hat{\mathbf{p}}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) \right) - \mathbf{N} \left(\mathbf{N}^T \mathbf{N}^{-1} \right) \mathbf{g} \left(\mathbf{p}_i^{(k)} \right) \quad (3.125)$$

dove oltre al termine legato al gradiente e alla sua proiezione nello spazio delle “feasible solutions”, si distingue il termine per il rispetto dei vincoli.² Considerando che, a differenza del caso centralizzato, l'algoritmo distribuito caricato sul i -esimo UAV calcola unicamente il controllo della sua posizione, la restoration move del gradiente proiettato viene calcolata utilizzando unicamente $\mathbf{p}_i^{(k)}$. Analiticamente, il gradiente del PEB percepito dall' i -esimo UAV

²Notare che sono state utilizzate notazioni semplificative $\mathbf{P} = \mathbf{P}_i^{(k)}$ e $\mathbf{N} = \mathbf{N}_i^{(k)}$.

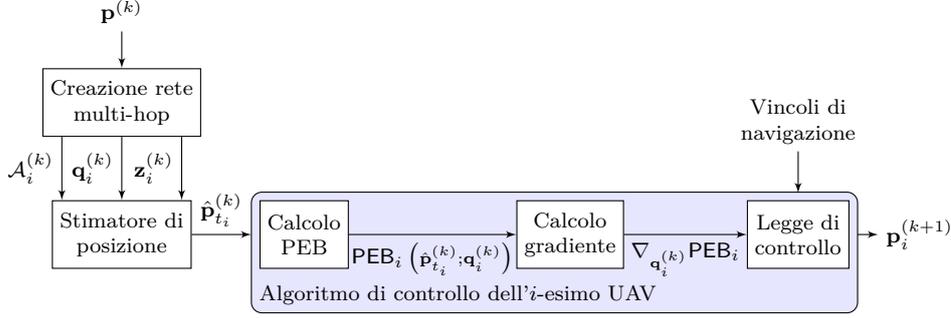


Figura 3.12: Schema a blocchi per localizzazione e navigazione congiunta dell' i -esimo UAV tramite un algoritmo distribuito.

può essere calcolato utilizzando le (3.99)-(3.106) di Sezione 3.2.4, sostituendo ciclicamente per $j = 1, \dots, n$, a \bar{x} , \bar{y} :

$$\bar{x} = x_j^{(k-h_{ij}+1)} - \hat{x}_t \quad (3.126)$$

$$\bar{y} = y_j^{(k-h_{ij}+1)} - \hat{y}_t. \quad (3.127)$$

In Figura 3.12 viene rappresentato lo schema a blocchi dell'algoritmo di navigazione distribuito che viene utilizzato da ciascun UAV. In simulazione, a partire dalle posizioni degli UAV, attraverso il primo blocco, viene creata la configurazione dei collegamenti quantificando il numero di hop per raggiungere una destinazione. Questo blocco fornisce al simulatore le posizioni, con relativi ritardi dovuti a multi-hop, che ciascun UAV riceve. Si noti, inoltre, che anche il vettore $\mathbf{z}_i^{(k)}$ contiene informazioni ritardate, di conseguenza lo stimatore di posizione si basa su misure di distanza potenzialmente out-of-date, fornendo una prestazione peggiore. In seguito, dopo il blocco di stima di posizione del target, viene evidenziato in azzurro il blocco dedicato all'algoritmo di controllo di posizione applicato sul generico UAV.

Capitolo 4

Simulazioni con algoritmo centralizzato

ALL'INTERNO di questo capitolo vengono riportate delle simulazioni per valutare il comportamento dell'algoritmo nella versione centralizzata in varie circostanze. Considerando la natura iterativa dell'algoritmo, si è scelto di utilizzare come criterio di stop un numero massimo K di iterazioni. Il valore di K è stato scelto accuratamente da caso a caso per garantire la convergenza alle posizioni ottimali degli UAV.

Il capitolo è stato suddiviso in tre sezioni principali. Nella prima sezione viene discusso il comportamento dell'algoritmo utilizzando per il calcolo del PEB la posizione esatta del target. In particolare, vengono effettuate delle simulazioni di ottimizzazione libera e ottimizzazione vincolata dove sarà descritto nel dettaglio ogni aspetto relativo al rispetto dei vincoli introdotti in Capitolo 3. In Sezione 4.2 vengono effettuate delle simulazioni ipotizzando di aver stimato la posizione dell'emettitore tramite uno stimatore Maximum Likelihood. Questo stimatore, benché sia asintoticamente efficiente, presenta sempre un errore non nullo per cui il calcolo del Position Error Bound si basa su informazioni non perfette, come avviene nei casi pratici. Infine, in Sezione 4.3 vengono riportate delle simulazioni in scenari in cui il link dei nodi possa essere anche *Non-Line of Sight* (NLOS). Per descrivere nel migliore dei modi il comportamento della infrastruttura dinamica composta dagli UAV, la trattazione è strutturata in modo graduale: i) inizialmente vengono riportati esempi di funzionamento per mostrare come l'algoritmo si comporta introducendo sequenzialmente i vincoli descritti in Sezione 3.2, discutendo il comportamento qualitativo del sistema; ii) successivamente vengono discussi esempi con algoritmo completo al variare di alcuni parametri per analizzare quantitativamente le performance nei vari scenari, fornendo i dati utilizzati in simulazione.

4.1 Algoritmo di controllo con posizione del target nota

In questa sezione vengono riportati dei risultati di simulazione ottenuti nell'ipotesi in cui si verifichi $\hat{\mathbf{p}}_t = \mathbf{p}_t$.

4.1.1 Ottimizzazione libera

Le prime simulazioni riportate sono state ottenute utilizzando un modello a varianza dell'errore di stima costante $\sigma_i^2 = \sigma_0^2$ in assenza di vincoli, risolvendo il problema di ottimizzazione libera della (3.1):

$$\mathcal{P}_0 : \mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \sqrt{\text{Tr} \{ \mathbf{J}^{-1}(\mathbf{p}_t; \mathbf{p}) \}} = \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \quad (4.1)$$

che al calcolatore può essere risolto iterativamente utilizzando:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \boldsymbol{\varepsilon}^{(k)} \odot \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \quad (4.2)$$

dove l'espressione del PEB è definito dalla (2.28):

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sigma_0 \sqrt{\frac{n}{\sum_{i=1}^n \cos^2 \theta_i \sum_{i=1}^n \sin^2 \theta_i - (\sum_{i=1}^n \cos \theta_i \sin \theta_i)^2}}. \quad (4.3)$$

In uno scenario di questo tipo, è importante notare che grazie al modello di rumore di stima indipendente dalla distanza dei nodi dal target, gli UAV si posizionano basandosi unicamente su fattori geometrici, non essendo presenti informazioni legate alla qualità del link.

In Figura 4.1 sono riportati alcuni esempi delle traiettorie seguite dagli UAV ottenute minimizzando il PEB della (2.28). Per la rappresentazione dei grafici di Figura 4.1 si è simulato l'algoritmo di posizionamento in uno scenario caratterizzato da $\sigma_0 = 1\text{m}$ e per rendere più semplice la visualizzazione della formazione degli UAV è stata rappresentata la circonferenza centrata sull'emettitore, con raggio dato dalla media delle distanze degli UAV rispetto alla posizione \mathbf{p}_t . È molto interessante vedere come la posizione ottima ottenuta segua un arco di circonferenza, con i nodi che asintoticamente si dispongono in modo equidistante tra loro e alla stessa distanza dal target. Come si può notare dalla (4.3), è presente una simmetria che permette di ottenere lo stesso risultato sia con θ_i che con $\theta_i + \pi$ grazie alle seguenti relazioni trigonometriche:

$$\begin{aligned} \cos^2(\theta_i + \pi) &= \cos^2(\theta_i) \\ \sin^2(\theta_i + \pi) &= \sin^2(\theta_i) \\ \sin(\theta_i + \pi) \cos(\theta_i + \pi) &= \sin(\theta_i) \cos(\theta_i) \end{aligned}$$

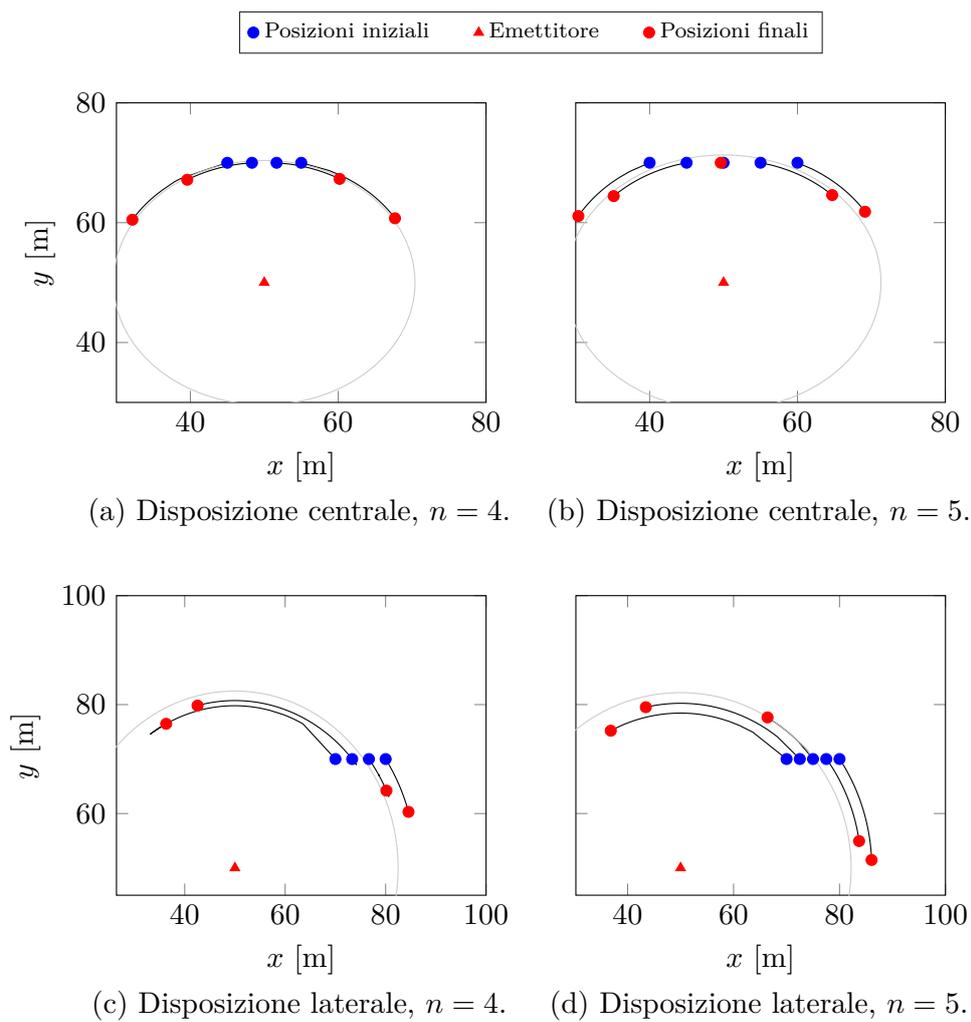


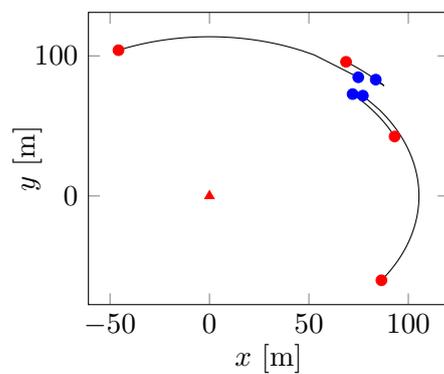
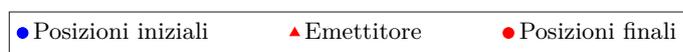
Figura 4.1: Posizionamento degli UAV con varianza dell'errore di stima costante per $n = 4, 5$.

che sostituite nella (4.3) portano esattamente alla stessa espressione. In Figura 4.1 sono riportati esempi riguardanti due posizioni iniziali di partenza: si nota sperimentalmente come una posizione iniziale favorevole porti ad una formazione che si dispone quasi perfettamente su un semicerchio centrato sull'emettitore, con droni a distanze simili, come nel caso di Figura 4.1a e Figura 4.1b. Partendo da posizioni iniziali più sfavorevoli, gli UAV, oltre ad impiegare più tempo per posizionarsi, creano una formazione tendente ad un cerchio in modo più irregolare. Un'altra importante considerazione da fare è che, come ci si aspettava, aumentando i numeri di droni, le performance aumentano: guardando le differenze fra le formazioni per 4 e 5 UAV, è facile notare, almeno qualitativamente, che in quest'ultimo caso si ha un miglioramento nella descrizione del semicerchio.

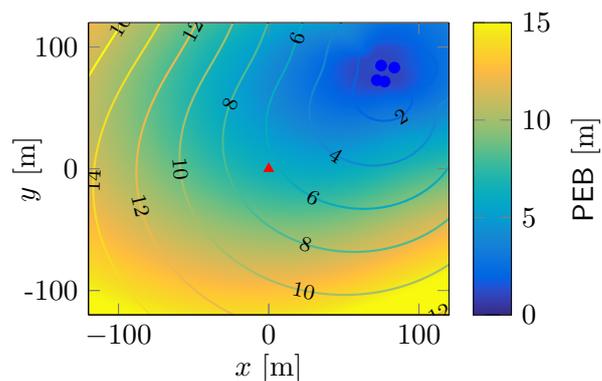
In Figura 4.2 sono riportati gli andamenti del Position Error Bound per la configurazione iniziale e per la formazione ottenuta attraverso l'algoritmo di controllo in assenza di vincoli. In Figura 4.2a sono riportate le traiettorie percorse da $n = 4$ UAV posizionati in modo casuale attorno ad un valor medio: si può vedere nuovamente come i nodi tendano ad disporsi su un arco di circonferenza il cui centro corrisponde alla posizione del target. Figura 4.2b mostra il valore del PEB nei punti dello spazio di simulazione per la configurazione iniziale, dove il colore più scuro rappresenta un valore basso del PEB. Come ci si aspettava, la zona all'interno dei 4 nodi ha bassa incertezza sulla posizione mentre nella zona dell'emettitore il PEB si alza molto velocemente. Invece, Figura 4.2c mostra il PEB dopo aver applicato l'algoritmo di posizionamento. È facile osservare come tutta la zona dell'emettitore sia caratterizzata da un basso valore di PEB, con linee di livello tutte comprese tra 1 e 2 metri, dimostrando come effettivamente l'algoritmo di posizionamento disponga gli UAV secondo la miglior geometria per minimizzare il Position Error Bound.

Considerando Figura 4.3 è possibile vedere i diversi livelli di PEB nel caso in cui ci siano $n = 4$ e $n = 6$ droni. Si osserva che, come già accennato in precedenza, il PEB migliora all'aumentare del numero di nodi coinvolti nel posizionamento. Infatti, riferendosi alla (4.3), all'aumentare del numero di UAV il numeratore aumenta ma il denominatore aumenta con un ordine superiore portando ad una diminuzione complessiva del PEB. Utilizzando la stessa scala, Figura 4.3a ha valori di Position Error Bound maggiori, quindi più chiari, rispetto Figura 4.3b dove si hanno valori al di sotto dell'unità, ottenendo una prova quantitativa del miglioramento della performance con un numero di UAV più elevato. Infatti, analiticamente, ipotizzando varianza dell'errore di stima costante, il PEB nel caso ottimo è dato da:

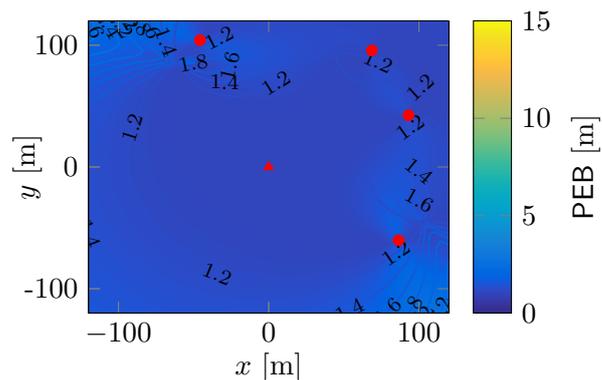
$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \frac{\sigma_0}{\sqrt{n}}. \quad (4.4)$$



(a) Posizionamento dei nodi.



(b) PEB in metri per la configurazione iniziale.



(c) PEB in metri per la configurazione finale.

Figura 4.2: Evoluzione del PEB per 4 nodi inizialmente disposti in modo casuale attorno ad un valor medio.

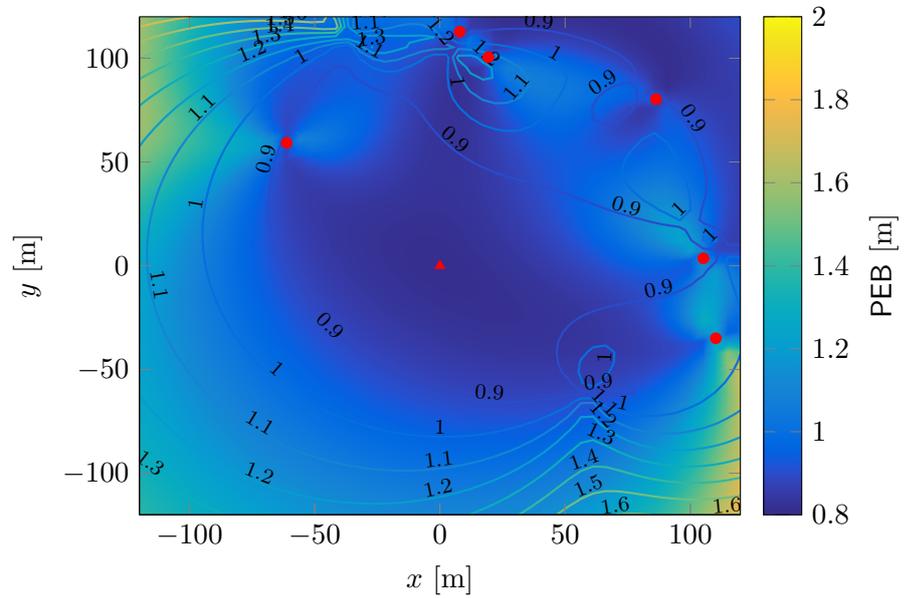
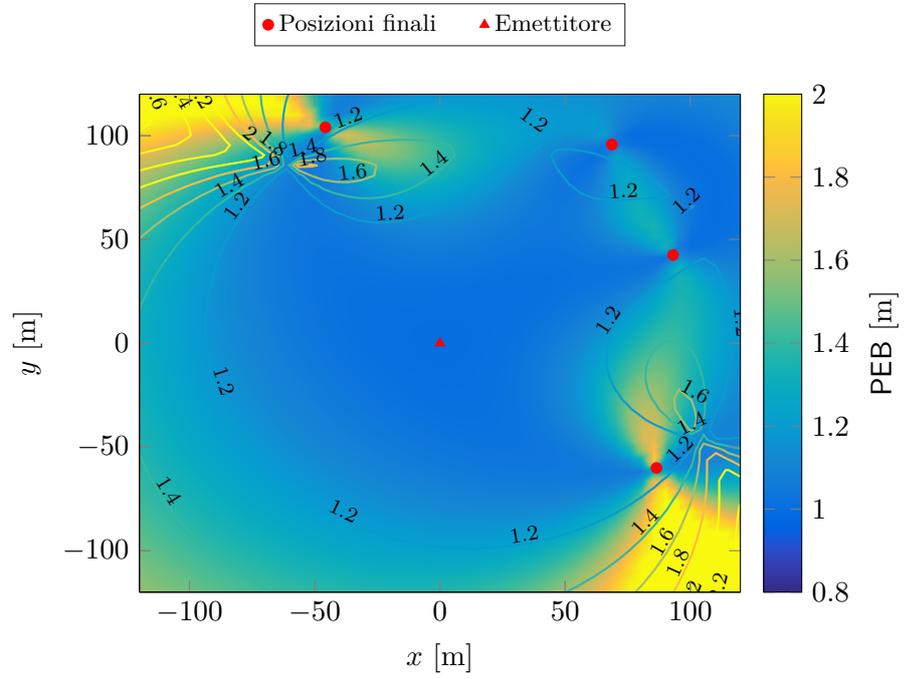


Figura 4.3: PEB al variare del numero di UAV.

4.1.2 Ottimizzazione vincolata con varianza d'errore di stima dipendente dalla distanza

Nelle simulazioni che seguono è stato utilizzato il modello con varianza dell'errore di stima dipendente dalla distanza, più realistico rispetto al caso in cui la qualità della misura prescindere dalla distanza. In questo scenario, il problema di ottimizzazione vincolata è espresso dalla (3.95) supponendo $\hat{\mathbf{p}}_t = \mathbf{p}_t$:

$$\begin{aligned} \mathbf{p}^* &= \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \text{PEB}(\mathbf{p}_t, \mathbf{p}) \\ \text{tale che} \quad & 1) \ d_{ij} \geq d_{min}, \ i, j = 1, \dots, n, \ i \neq j \\ & 2) \ d_{it} \geq d_{min_t}, \ \forall i \\ & 3) \ \mathcal{T}_i \cap \mathcal{O} = \emptyset, \ \forall i \\ & 4) \ \left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|_2 = v, \ \forall i \end{aligned} \quad (4.5)$$

e può essere risolto iterativamente utilizzando la (3.96):

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \boldsymbol{\varepsilon}^{(k)} \odot \left(\mathbf{P} \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\mathbf{p}_t; \mathbf{p}) \right) - \mathbf{N} \left(\mathbf{N}^T \mathbf{N}^{-1} \right) \mathbf{g} \left(\mathbf{p}^{(k)} \right) \quad (4.6)$$

dove il PEB è descritto dalla (2.46):

$$\text{PEB}(\mathbf{p}_t; \mathbf{p}) = \sqrt{\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n A_i \left(\frac{x_i - x_t}{d_i} \right)^2 \sum_{i=1}^n A_i \left(\frac{y_i - y_t}{d_i} \right)^2 - \left(\sum_{i=1}^n A_i \frac{(x_i - x_t)(y_i - y_t)}{d_i^2} \right)^2}} \quad (4.7)$$

In Figura 4.4 è riportato l'esempio di quattro nodi disposti in linea che devono localizzare il target situato all'interno di un edificio, con ostacoli rappresentati dagli edifici nelle vicinanze. Questo scenario è quello che si riscontra quando si vuole localizzare un emettitore in uno spazio urbano. Infatti, immaginando di avere il target da localizzare in un grattacielo e portando gli UAV alla stessa altezza dell'emettitore, tutti gli ostacoli a bassa quota possono essere considerati come non pericolosi. Le traiettorie di Figura 4.4 sono state ottenute utilizzando il metodo del potenziale artificiale supponendo note le forme e le posizioni degli ostacoli. Per evidenziare la distanza minima che i droni devono mantenere dall'ostacolo, in Figura 4.4 sono state rappresentate le aree proibite con una linea tratteggiata. Per mostrare meglio l'andamento del PEB, a differenza delle altre simulazioni presenti in questo capitolo, si è scelto di utilizzare uno scenario con una scala ridotta, così da accentuare ogni piccola oscillazione.

È interessante notare come i droni si portino, compatibilmente alla distanza di sicurezza che ciascun UAV deve mantenere dagli ostacoli, il più possibile vicino all'emettitore. Questo risulta coerente se si analizzano i coefficienti A_i della (2.44) in quanto questi tendono ad aumentare al diminuire della distanza. Così facendo il numeratore della (2.46) aumenta linearmente,

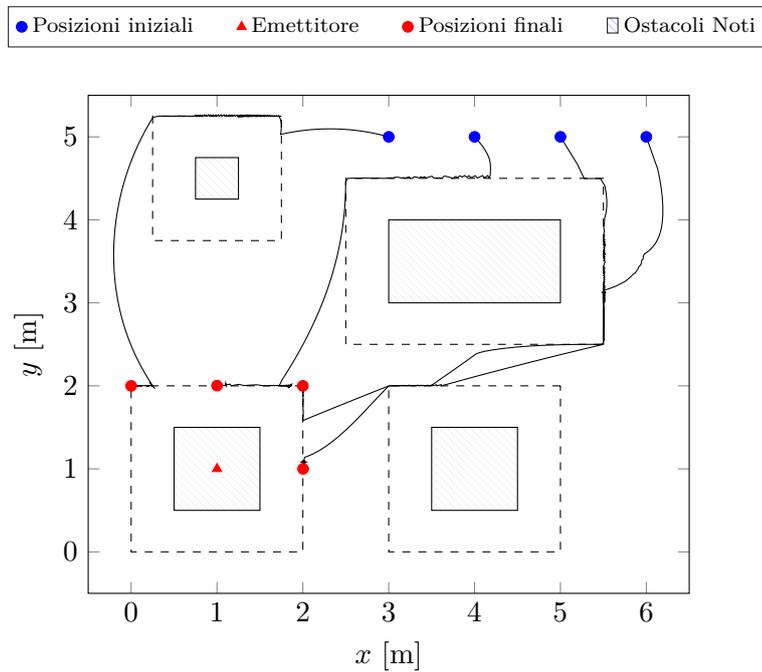
ma il denominatore aumenta con un ordine superiore, portando ad un valore complessivo di PEB minore al diminuire della distanza. Come per il modello a varianza dell'errore di stima costante, anche in questo caso i nodi si posizionano su una semicirconferenza a causa della simmetria che anche in questo caso viene mantenuta. I coefficienti A_i , infatti, dipendono dalla distanza ma non dalla direzione, per cui il valore rimane lo stesso su tutta la circonferenza di raggio pari alla distanza tra emettitore e nodo mobile. Quando un nodo approccia un ostacolo si hanno delle fasi in cui il Position Error Bound non migliora, perché non può seguire la traiettoria ottima per raggiungere la collocazione nella formazione migliore. Questo effetto è chiaramente visibile in Figura 4.4b: dove i tratti rettilinei dell'andamento del PEB sono causati da un temporaneo impedimento causato dagli ostacoli, che non permettono agli UAV di percorrere la traiettoria ottimale per ottenere una miglior stima di posizione del target. Una volta raggiunte le posizioni ottimali, i nodi tendono ad avere delle piccole oscillazioni che si ripercuotono sul grafico del PEB di Figura 4.4b con dei piccolissimi spikes.

Risultati simili sono stati ottenuti per ostacoli sconosciuti che sono stati rilevati dal sensore a bordo del drone, ma questo caso verrà discusso più dettagliatamente nelle simulazioni in cui il target sia outdoor.

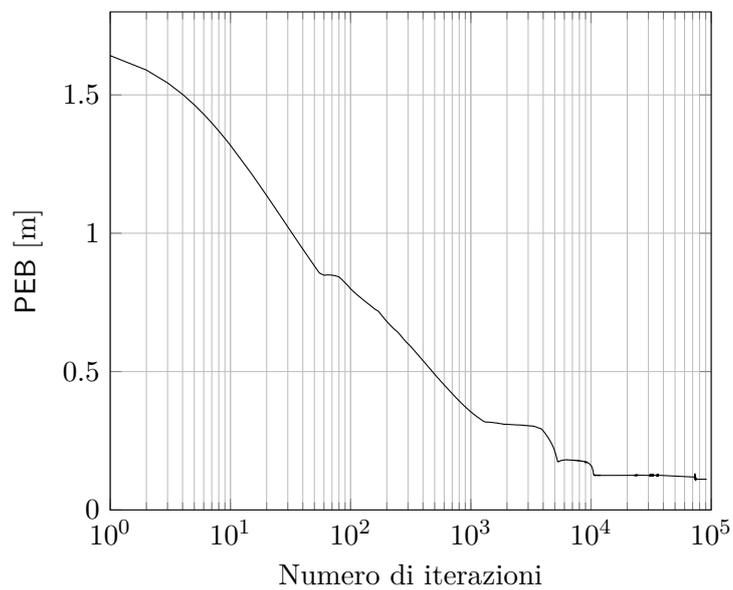
4.1.3 Emettitore dinamico in ambiente outdoor e indoor

Sinora, sono stati riportati esempi di funzionamento considerando il problema di localizzazione di un emettitore statico, supponendo cioè che non fosse mobile. In realtà, nella maggior parte delle possibili applicazioni pratiche questo non si verifica, soprattutto se si immagina di dover tracciare un dispositivo fornito ad un essere umano in condizioni di emergenza. Per simulare il movimento dell'emettitore è possibile utilizzare un random walk, vincolando il target a restare all'interno di un'area che per semplicità viene considerata di forma rettangolare per simulare l'edificio in cui si trova o lasciandolo libero di muoversi nell'ambiente. Lo spostamento dell'emettitore è irregolare in quanto l'algoritmo di movimento è basato sullo stato delle variabili di posizione x_t, y_t e delle componenti della velocità \dot{x}_t, \dot{y}_t e sulla generazione di campioni aleatori correlati.

In Figura 4.5 viene riportato un esempio dell'algoritmo di posizionamento in condizioni dinamiche, supponendo che l'emettitore possa spostarsi con una velocità molto inferiore rispetto a quella con cui possono spostarsi i nodi in modo da non avere delle formazioni transitorie in cui il Position Error Bound non sia minimizzato. In Figura 4.5a si mostra la formazione dei nodi con la posizione iniziale dell'emettitore posto al centro dell'edificio, come rappresentato negli esempi precedenti. Diverso è il caso di Figura 4.5b dove l'emettitore compie un percorso casuale che lo fa tendere nell'angolo alto destro dell'edificio: si può notare come il rettangolo tratteggiato sia contornato da molte linee solide rappresentanti le traiettorie percorse dai



(a) Posizionamento dei nodi.



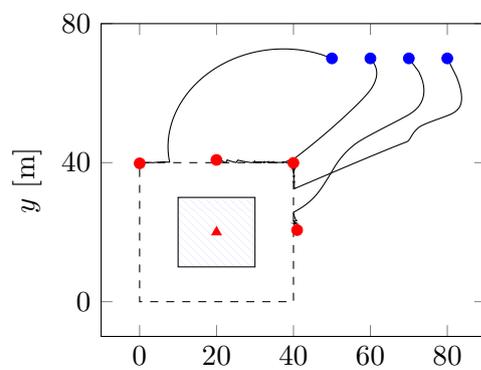
(b) Andamento PEB.

Figura 4.4: Formazione di 4 nodi disposti in linea in uno scenario in cui gli ostacoli sono noti e collision avoidance attivo.

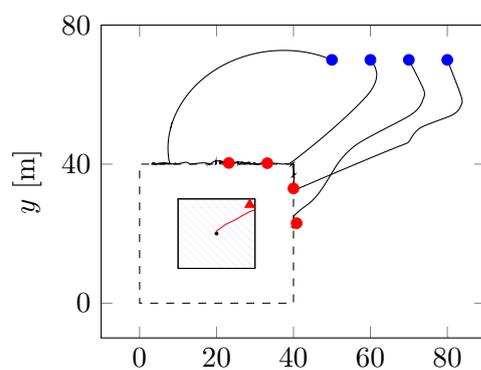
nodi. È ben visibile come i nodi traslino per descrivere la semicirconferenza. Dopo un certo numero di iterazioni, il random walk porta l'emettitore a sinistra dell'edificio come riportato in Figura 4.5c, dopo aver percorso la parte superiore dell'area a lui concessa. A causa della traiettoria che ha percorso il target, gli UAV hanno modificato le proprie posizioni per seguirlo lungo la parte superiore del rettangolo tratteggiato rappresentante la distanza minima dall'edificio.

In conclusione, una volta che i nodi sono arrivati vicino all'emettitore, mantenendo la distanza di sicurezza dall'edificio, si spostano per restare il più possibile vicini all'emettitore, mantenendo la distanza mutua di sicurezza per evitare collisioni, formando costantemente una semicirconferenza.

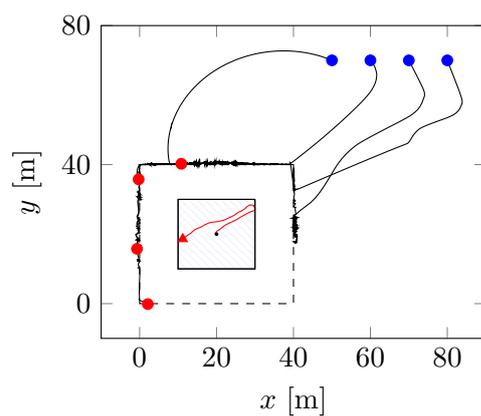
Un altro interessante caso è rappresentato dalla localizzazione di un emettitore che si trova in un ambiente libero da ostacoli e che esso stesso non sia circondato da un ostacolo. Si immagini di avere un emettitore all'interno di una grande regione libera, all'interno della quale siano presenti solo gli UAV, per introdurre un primo esempio di tracking di un target situato in ambiente outdoor, utile, ad esempio, per attuare tecniche di information seeking, nel caso in cui il target non possa utilizzare il GPS. Focalizzando l'attenzione su Figura 4.6, si distinguono sei frame, riportati ogni numero di iterazioni costante e pari a $\Delta = 10000$, in modo da riuscire facilmente a vedere l'evoluzione del sistema. Ciascun frame rappresenta la configurazione istantanea che gli UAV assumono all'iterazione k in base alla posizione dell'emettitore, in grado di muoversi senza restrizioni nello spazio bidimensionale. Lo spostamento dell'emettitore è progettato in modo che questo avvenga molto più lentamente della massima velocità con cui possono muoversi i nodi, per permettere agli UAV di raggiungere una condizione di quasi-regime in cui la formazione è tale da garantire il minimo del Position Error Bound. In Figura 4.6b le condizioni sono quasi come nel caso con emettitore statico: ricordando le differenti velocità permesse ai nodi e all'emettitore, gli UAV si posizionano descrivendo una semicirconferenza prima che l'emettitore abbia compiuto uno spostamento rilevante. Dopo altre Δ iterazioni, si vede da Figura 4.6c come la formazione si adatti per seguire l'emettitore descrivendo praticamente le medesime traiettorie. Studiando i restanti frame, si vede come i UAV tendano a spostarsi solidalmente con l'emettitore, adattando la formazione alla posizione variabile del target da localizzare ma mantenendo sempre una semicirconferenza di raggio pari alla distanza minima che i nodi devono tenere dall'emettitore.



(a) Emettitore centrale.



(b) Emettitore a destra.



(c) Emettitore a sinistra.

Figura 4.5: Formazione dinamica degli UAV nel caso in cui l'emettitore si muova casualmente all'interno di un rettangolo.

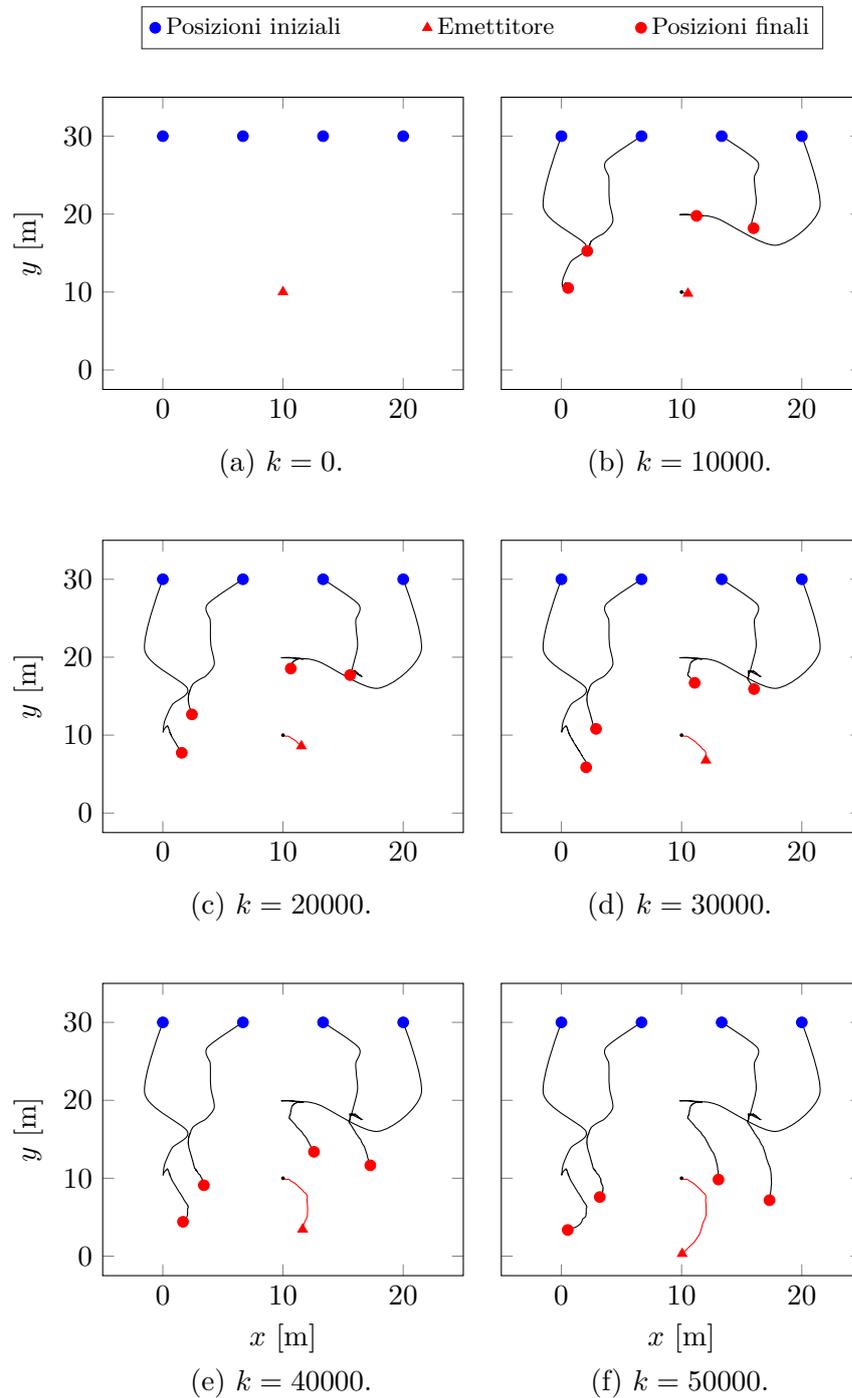


Figura 4.6: Formazione dei nodi in condizioni di emettitore in movimento libero, dove k rappresentano il numero di iterazioni dell'algoritmo.

4.1.4 Performance dell'algoritmo al variare del numero di nodi

Per effettuare un confronto delle prestazioni dell'algoritmo di controllo al variare del numero di nodi, è stato eseguito un test con $n = 3, 4, 5, 6$ nodi collocati in uno scenario in cui gli ostacoli siano non noti a priori ed emettitore libero. I nodi vengono fatti partire linearmente nello stesso spazio per tutti i casi, quindi all'aumentare del numero di nodi n le distanze mutue tra i nodi diminuiscono, innescando il collision avoidance. La descrizione dei dati in ingresso della simulazione verranno forniti utilizzando la sintassi della versione 2017a di MATLAB. Le posizioni iniziali rispettivamente di emettitore e nodi sono:

```
pt0=[0,0]
n=[3,4,5,6]
x0=linspace(55,60,n(i))
y0=150*ones(1,n(i))
p0=[x0' y0']
```

dove la posizione dell'emettitore, in questo scenario statico, è costante. Supponendo di utilizzare uno stimatore Maximum Likelihood per la stima della distanza, è possibile derivare il Cramér-Rao Bound ottenendo il limite della varianza sfruttando la formula di [8, Eq. (16)]:

$$\sigma_0^2 \geq \frac{c^2}{8\pi^2\beta^2\text{SNR}} \quad (4.8)$$

dove c è la velocità della luce, β è chiamata *banda effettiva* e SNR è il rapporto segnale-rumore. Approssimando con $\beta^2 \approx W^2$, dove W è la banda del segnale, si ottiene:

$$\sigma_0 \approx \sqrt{\frac{c^2}{8\pi^2W^2\text{SNR}}} \quad (4.9)$$

e ipotizzando di utilizzare un segnale con banda $W = 100\text{MHz}$ e $\text{SNR} = 10\text{dB}$ si ottiene $\sigma_0 \approx 0.1\text{m}$. In simulazione, quindi, sono stati utilizzati valori di σ_0 e di α costanti, pari a:

```
sigma0=0.1
alpha=2.
```

Non avendo ostacoli la cui posizione e forma siano noti, è stato fondamentale l'utilizzo del modello di sensore al quale è stato assegnato un raggio di copertura pari a $d_{\text{obs}}=5$ e parametro τ per il calcolo dei coefficienti esponenziali $\text{tau}=5$. Inoltre, è stata utilizzata la distanza $d_{\text{agents}}=5$ anche per la distanza minima che i nodi devono reciprocamente mantenere mentre è stata utilizzata una distanza minima dall'emettitore pari a $d_{\text{emitter}}=15$. Questo parametro deve essere sufficientemente alto per far sì che i nodi riescano a

Numero di nodi n	PEB	Iterazione PEB minimo k^*
3	0.1705m	2349
4	0.1538m	3522
5	0.1310m	3897
6	0.1239m	6239

Tabella 4.1: Risultati di simulazione: minimo PEB e iterazione k^* al quale è stato raggiunto.

rispettare il vincolo di distanza minima reciproca, di distanza minima dall'emettitore e posizionarsi correttamente. Gli ostacoli sono stati descritti fissando il centro e calcolando i vertici tramite le altezze e le lunghezze fornite dall'utente, utilizzando:

$$x_{max}, x_{min} = cr_x \pm \frac{l}{2}$$

$$y_{max}, y_{min} = cr_y \pm \frac{h}{2}$$

per ogni ostacolo. In questa simulazione, gli ostacoli sono stati creati a partire da:

```
cr=[-40,100;60,100;80,40;5,50]
lung=[40,20,20,20]
alt=[40,40,40,20].
```

Nella simulazione seguente la velocità è stata mantenuta costante e settata a $v=1$ e un numero di iterazioni massimo di $K=1e4$.

Analizzando le traiettorie riportate in Figura 4.7, si può vedere come all'aumentare del numero dei nodi aumentino le possibilità che partano da posizioni che non rispettano i vincoli di distanza mutua. Prendendo i casi estremi con $n = 3$ di Figura 4.7a e $n = 6$ di Figura 4.7d, si vede come le traiettorie di quest'ultimo siano più complesse, con più curve, per evitare collisioni. All'aumentare dei nodi la zona attorno all'emettitore diventa più popolata. Ricordando che i nodi devono mantenere una distanza di sicurezza dall'emettitore pari a $d_emitter=15$, si nota come i nodi si dispongano su un arco di circonferenza corrispondente alla minima distanza a loro concessa oltre al quale si attiverebbero dei vincoli.

Studiando le curve di Figura 4.7e, si possono vedere gli andamenti del Position Error Bound al variare del numero di nodi coinvolti. Il grafico è strutturato in modo tale da mostrare l'andamento per ogni iterazione di simulazione e viene riportato un ingrandimento dei valori a regime per mostrare meglio quale sia la differenza nel risultato finale. A differenza di quello che si vede dalle traiettorie, in Figura 4.7e è possibile vedere che a

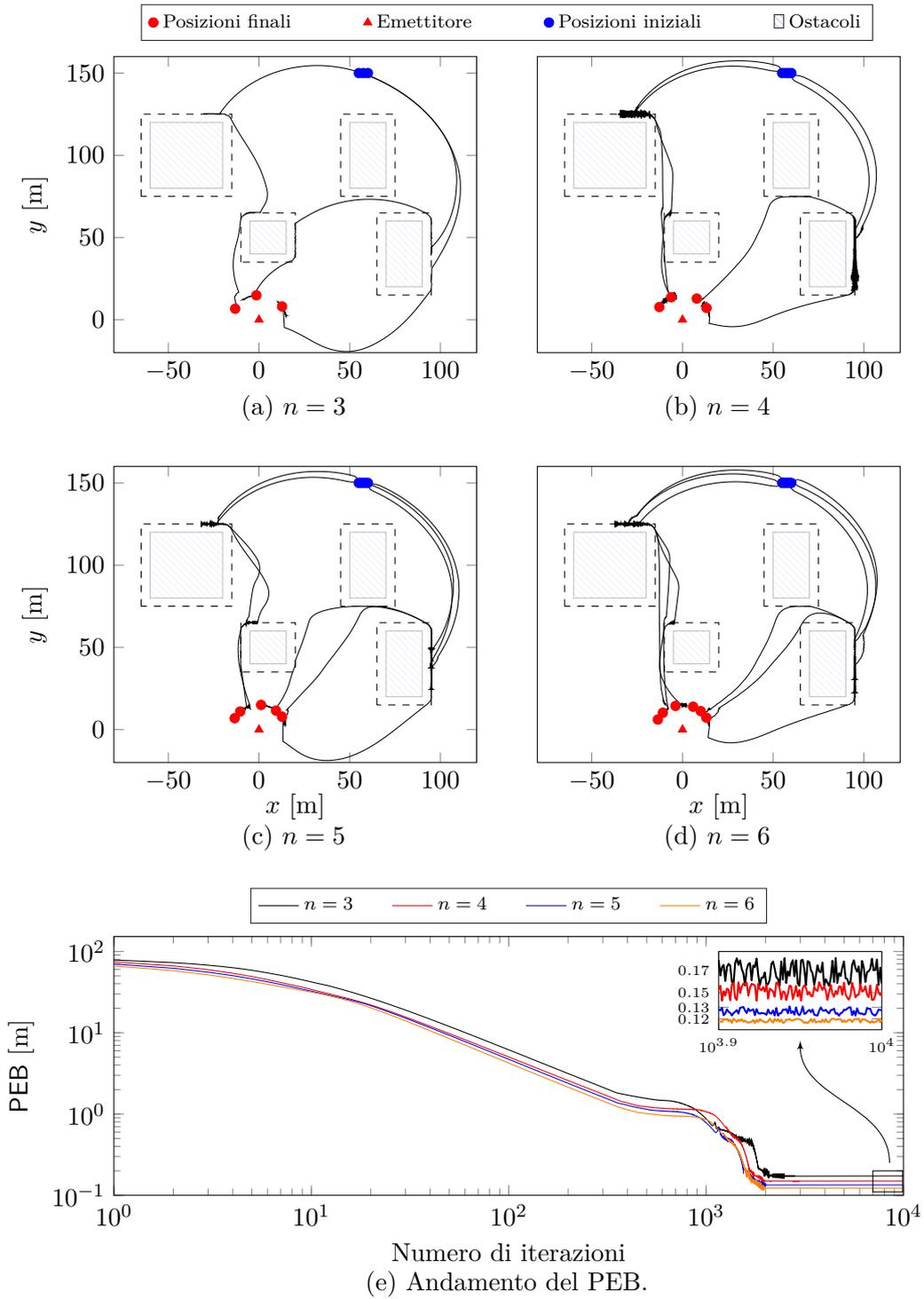


Figura 4.7: Confronto delle traiettorie e del PEB per un numero variabile di nodi.

regime si hanno delle oscillazioni di PEB, causate da piccoli spostamenti che hanno gli UAV attorno alle posizioni ottimali. Si nota come il valore del PEB sia decrescente al crescere del numero dei nodi e che anche l'oscillazione dei nodi tende a calare con l'aumentare di n . Dai risultati di simulazione, si vede che il miglioramento maggiore si ottiene nel passaggio da $n = 3$ a $n = 4$, per poi avere un miglioramento sempre più basso all'aumentare di molto il numero di nodi.

Un altro aspetto evidenziato da Figura 4.7e è che con più nodi il Position Error Bound decresce più rapidamente ma allo stesso tempo sono necessarie più iterazioni per portare in posizione tutti i nodi, quindi si raggiunge il risultato di regime più lentamente rispetto al caso con pochi UAV coinvolti. I risultati ottenuti in simulazione sono riassunti in Tabella 4.1, dov'è stato riportato il numero di nodi n , il minimo valore di PEB assunto e l'iterazione k^* in cui nodi si sono disposti in modo ottimale. L'iterazione k^* è stata calcolata andando a ricercare il time slot in cui si è verificata la prima occorrenza del valore di regime del PEB. Il trend del minimo PEB è già stato discusso a partire dal grafico di Figura 4.7e, mentre dall'ultima colonna di Tabella 4.1 è chiaro come la convergenza dell'algoritmo si abbia più tardi all'aumentare del numero di nodi. Questo implica un maggior tempo e un maggior costo computazionale all'aumentare del numero di nodi.

Apparentemente, in uno scenario di questo tipo, queste considerazioni possono far pensare che aumentare il numero di nodi n possa essere uno spreco di risorse visto l'esiguo miglioramento che si ha a livello di PEB. In realtà, un importante miglioramento c'è e verrà discusso in Sezione 4.2, dove verrà discusso anche il concetto di convergenza dell'algoritmo.

4.2 **Algoritmo di controllo con stima della posizione dell'emettitore**

Nelle simulazioni precedenti, si è supposto di essere a conoscenza della posizione dell'emettitore. In pratica, questo non avviene in quanto lo scopo dei nodi è proprio quello di localizzare l'emettitore, quindi calcolare la posizione in cui è situato. Anziché fare riferimento ad un particolare stimatore di posizione, si è deciso di emulare la prestazione dello stimatore utilizzando come riferimento la prestazione di uno stimatore efficiente, con prestazione uguale al PEB. Pertanto le stime di posizione sono state generate come Gaussiane 2D con media uguale al valore esatto di posizione e deviazione standard data dal PEB, come nella (3.5). Questo approccio rende l'analisi "agnostica" rispetto ad una particolare scelta dello stimatore seppur ottimistica quindi da utilizzarsi come benchmark per stimatori reali. Asintoticamente, in condizioni di elevato SNR, uno stimatore efficiente è il ML e nel seguito si

ipotizzerà di essere in queste condizioni. Così la (4.5) diventa:

$$\begin{aligned}
\mathbf{p}^* &= \arg \min_{\mathbf{p} \in \mathbb{R}^{2n}} \text{PEB}(\hat{\mathbf{p}}_t, \mathbf{p}) \\
\text{tale che} \quad & 1) \ d_{ij} \geq d_{min}, \ i, j = 1, \dots, n, \ i \neq j \\
& 2) \ d_{it} \geq d_{min_t}, \ \forall i \\
& 3) \ \mathcal{T}_i \cap \mathcal{O} = \emptyset, \ \forall i \\
& 4) \ \left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|_2 = v, \ \forall i
\end{aligned} \tag{4.10}$$

e può essere risolta iterativamente utilizzando:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \boldsymbol{\varepsilon}^{(k)} \odot \left(\mathbf{P} \nabla_{\mathbf{p}^{(k)}} \text{PEB}(\hat{\mathbf{p}}_t; \mathbf{p}) \right) - \mathbf{N} \left(\mathbf{N}^T \mathbf{N}^{-1} \right) \mathbf{g} \left(\mathbf{p}^{(k)} \right). \tag{4.11}$$

Utilizzando le coordinate stimate per il calcolo del PEB della k -esima iterazione, si ottengono dei valori che possono essere distorti dall'incertezza della stima. In Figura 4.8 è riportato l'andamento del PEB nel caso in cui siano utilizzate le reali coordinate dell'emettitore e nel caso in cui si utilizzino le coordinate stimate. La simulazione è stata lanciata utilizzando un numero di nodi $n = 3$, posti in linea al di sopra dell'emettitore, in un ambiente caratterizzato da un rumore di stima pari a $\sigma_0 = 0.01\text{m}$, esponente path-loss $\alpha = 2$ e assegnando agli UAV una velocità $v = 0.01\text{m/step}$. Per impostare questi parametri, utilizzando MATLAB 2017a, si inseriscono le istruzioni:

```

pt0=[0,0]
n=3
x0=linspace(-10,10,n)
y0=150*ones(1,n)
p0=[x0',y0']
sigma0=0.01
alpha=2
v=0.01
d_agents=5
d_emitter=15
K=1e4

```

Da Figura 4.8 si evince come l'andamento del PEB all'aumentare del numero di iterazioni sia diverso per i due casi in esame. Nel caso in cui le posizioni siano esatte, l'andamento è caratterizzato da una discesa con una bassa componente oscillatoria. Nel caso più realistico di posizioni stimate, l'andamento del PEB è pressoché simile al caso precedente ma caratterizzato da una forte componente oscillatoria che tende a rendere meno regolare la curva. Il maggior effetto oscillatorio si vede anche a regime: la curva nera è molto più spessa rispetto alla curva rossa e questo sta a significare che i nodi hanno un'oscillazione più ampia attorno alla posizione ottima nel caso di coordinate stimate rispetto al caso con posizione nota.

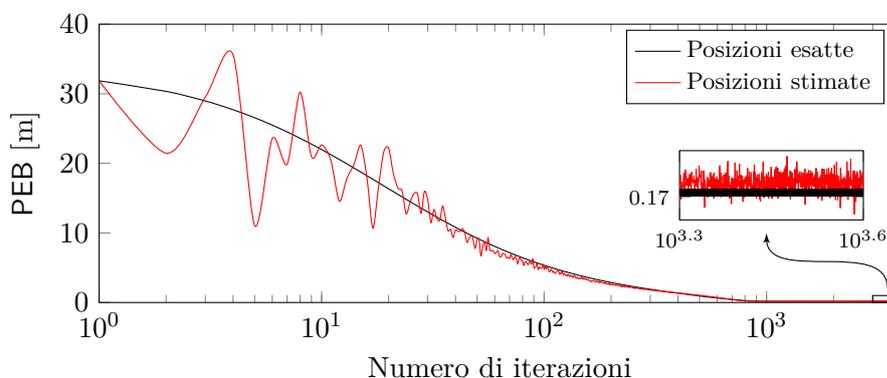
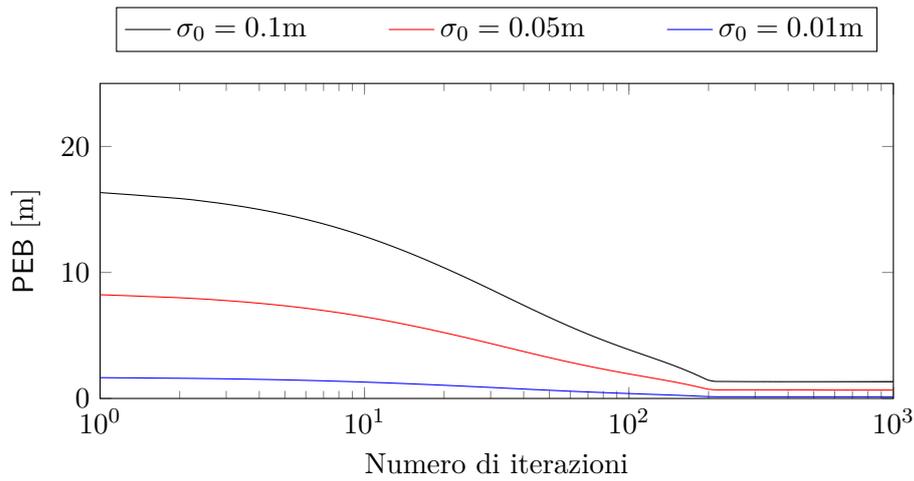
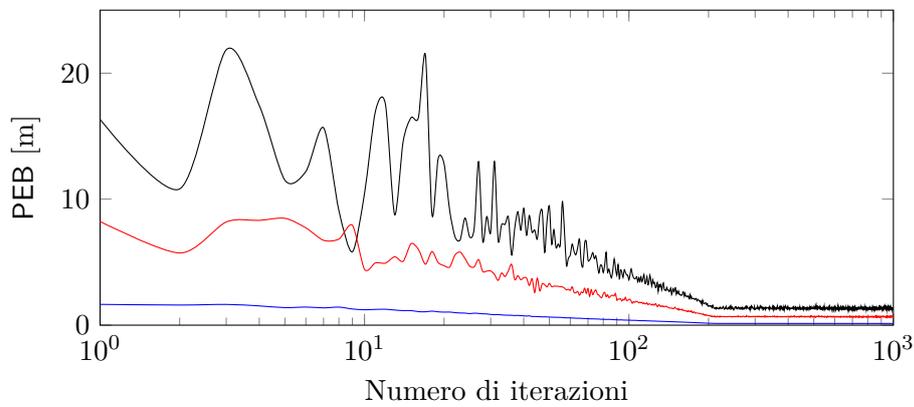


Figura 4.8: Differenza dell'evoluzione del PEB per il caso in cui le coordinate dell'emettitore siano note e nel il caso siano stimate da uno stimatore ML.

Le traiettorie degli UAV nel caso in cui la posizione sia stata stimata è molto simile a quella del caso con coordinate esatte: l'unica differenza è rappresentata da qualche piccola irregolarità nelle curve. Per questa piccola differenza si è scelto di non riportare le traiettorie delle simulazioni.

In Figura 4.9 sono riportate delle simulazioni al variare dell'intensità del rumore di stima della distanza caratterizzato nel simulatore da $\sigma_0 = 0.1\text{m}, 0.05\text{m}, 0.01\text{m}$. Partendo dal caso di coordinate note dell'emettitore di Figura 4.9a si vede, com'era facilmente ipotizzabile, che l'andamento del Position Error Bound ha bassa oscillazione e tende ad appiattirsi verso lo zero al diminuire del rumore di stima.

Molto interessanti sono i risultati di Figura 4.9b nel caso di posizioni stimate. A rumori di bassa entità, le curve sono molto simili a quelle di Figura 4.9a anche se si vedono delle leggere irregolarità. Invece, all'aumentare del rumore di stima della posizione, le curve iniziano ad essere decisamente irregolari, con oscillazioni via via sempre più intense a causa della peggior stima della posizione dell'emettitore. Nel caso in cui il rumore di stima sia molto elevato, l'algoritmo tende ad essere meno stabile e si rischia di non convergere ad una soluzione valida. Questo effetto è causato da un grande errore nella stima della posizione dell'emettitore che causa un cattivo posizionamento degli UAV che al passo successivo avranno un valore di PEB molto elevato portando a peggiorare ulteriormente la stima e ad errare maggiormente il posizionamento, entrando in un loop che porta all'errore l'algoritmo.

(a) PEB al variare di σ_0 con posizione dell'emettitore nota.(b) PEB al variare di σ_0 con posizione dell'emettitore stimata.**Figura 4.9:** PEB nel caso in cui vari σ_0 con posizione dell'emettitore nota e stimata.

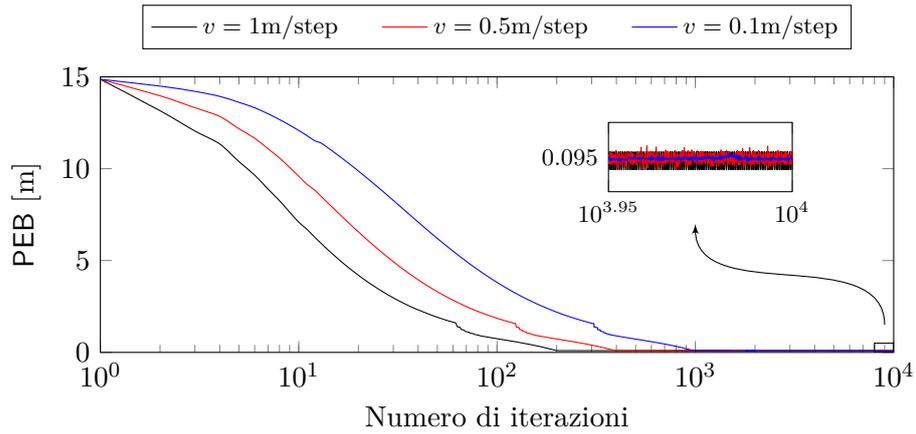
4.2.1 Performance dell'algoritmo al variare della velocità dei nodi

Una importante simulazione riguarda le diverse velocità che possono essere raggiunte dagli UAV per singolo step di processo. Utilizzando la stima della posizione dell'emettitore e le stesse condizioni delle simulazioni di Sezione 4.2 ma fornendo tre diverse velocità $v = 1\text{m/step}$, 0.5m/step , 0.1m/step , utilizzando il codice:

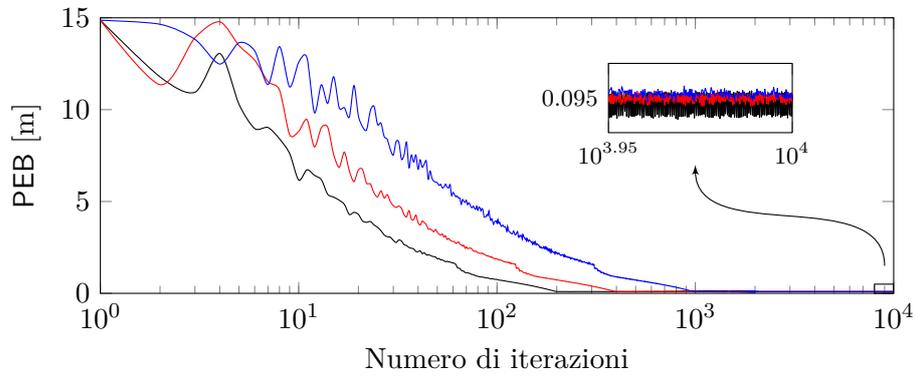
```
pt0=[0,0]
n=5
x0=linspace(-10,10,n)
y0=150*ones(1,n)
p0=[x0',y0']
sigma0=0.01
alpha=2
v=[1,0.5,0.1]
d_agents=5
d_emitter=15
K=1e4
```

si ottengono i risultati di Figura 4.10. In Figura 4.10a si vede l'andamento del Position Error Bound al variare della velocità utilizzando la posizione esatta dell'emettitore. Dalle curve si vede come utilizzando una velocità elevata i nodi arrivino prima ad un buon valore di PEB rappresentato dal ginocchio che si viene a creare quando gli UAV arrivano alla minima distanza consentita dall'emettitore. Il problema di una elevata velocità è ben visibile nel riquadro che mostra nel dettaglio quello che si verifica a regime. Dalla finestra si vede che l'algoritmo con elevata velocità presenta una maggior oscillazione rispetto alle simulazioni a bassa velocità, portando ad un maggior movimento attorno alle posizioni ottime.

In Figura 4.10b, invece, si simulano le diverse velocità utilizzando le coordinate stimate. Come nei casi precedenti, stimando la posizione si è soggetti a forti oscillazioni causate dall'errore sulle coordinate stimate. A regime si nota un leggero aumento nell'oscillazione anche a bassa velocità rispetto al caso precedente, causato dalla sovrapposizione di oscillazioni legate alla velocità con cui i nodi oscillano attorno alle posizioni ottimali e all'errore di stima della posizione del target.

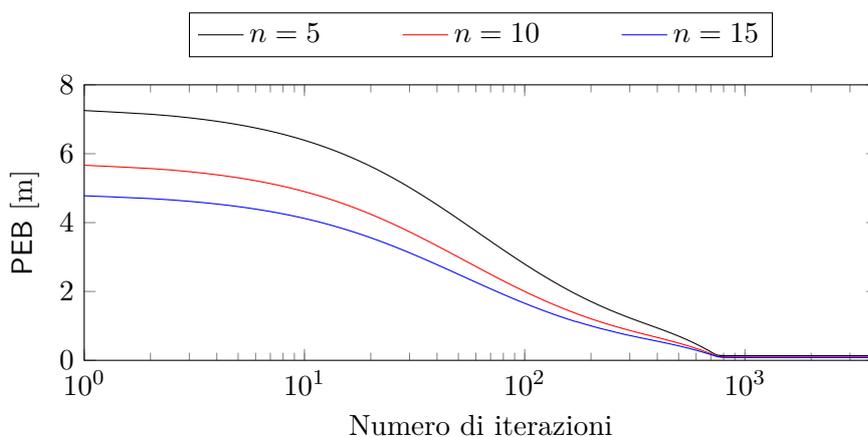


(a) Simulazione del PEB al variare della velocità massima basandosi sulle coordinate esatte dell'emettitore.

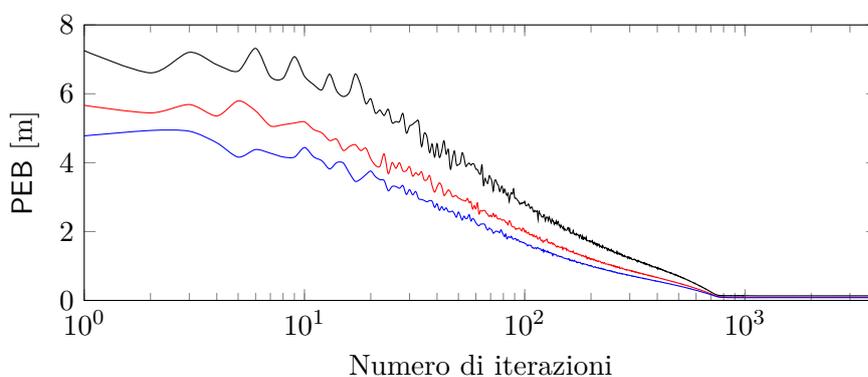


(b) Simulazione del PEB al variare della velocità massima basandosi sulla stima ML della posizione dell'emettitore.

Figura 4.10: Andamenti del PEB con posizione dell'emettitore nota e con posizione dell'emettitore stimata tramite stimatore ML.



(a) Andamento PEB calcolato con posizione del target esatta.



(b) Andamento PEB calcolato con posizione del target stimata.

Figura 4.11: Andamento del PEB al variare del numero dei nodi, con posizione dell'emettitore stimata.

4.2.2 Simulazione al variare del numero di nodi

Come già fatto per il calcolo del PEB utilizzando le coordinate esatte dell'emettitore in Figura 4.7, è interessante vedere come l'algoritmo di posizionamento si comporti all'aumentare del numero di UAV coinvolti utilizzando la stima di posizione. Si utilizza una velocità pari a $v = 0.01\text{m/step}$, rumore di stima pari a $\sigma_0 = 0.01\text{m}$, esponente path-loss $\alpha = 2$ e un numero $n = 3, 4, 5$, di nodi in linea. I dati vengono inseriti su MATLAB 2017a utilizzando le seguenti istruzioni:

```
pt0=[0,0]
n=[5,10,15]
x0=linspace(-20,20,n(i))
y0=150*ones(1,n(i))
```

```

p0=[x0',y0']
sigma0=0.01
alpha=2
v=0.01
d_agents=5
d_emitter=15
K=1e4

```

che permettono di ottenere gli andamenti riportati in Figura 4.11. All'aumentare del numero di nodi n , il PEB ha valori più bassi già dalle prime iterazioni, ma tende a decrescere più lentamente rispetto al caso in cui i nodi coinvolti siano pochi. Infatti, si conferma che la velocità con cui il Position Error Bound raggiunge valori intorno al minimo è maggiore nel caso con n basso.

Il principale guadagno nell'utilizzare un gran numero di UAV è l'aumento della probabilità di convergenza dell'algoritmo di controllo. Nel caso di algoritmo centralizzato, ogni simulazione si cataloga come convergente quando non è stato riscontrato nessun errore nel calcolo del PEB. Questa definizione viene rivisitata nel Capitolo 5 a causa dell'algoritmo distinto per ogni UAV.

Nei risultati di simulazione precedenti, all'aumentare del rumore di stima della posizione del target si possono incontrare dei casi in cui il PEB è talmente elevato da stimare una posizione molto errata, non permettendo all'algoritmo di calcolare i comandi corretti da fornire ai droni. Oltre al rumore di stima, il problema della non convergenza viene amplificato anche dalla distanza iniziale dei nodi dall'emettitore. Qualitativamente, il miglioramento nella robustezza dell'algoritmo può essere dedotto dal diminuire dell'intensità delle oscillazioni all'aumentare del numero di UAV, come ben visibile in Figura 4.11b.

Per lo studio della percentuale di convergenza dell'algoritmo, fissato uno scenario, a causa della variabile aleatoria Gaussiana, è necessario effettuare una simulazione Monte Carlo. I risultati di Tabella 4.2 sono stati calcolati lanciando $N_{MC} = 100$ simulazioni per scenario, facendo variare il numero di nodi n e la distanza iniziale dei nodi. Sono invece stati mantenuti costanti i valori di rumore di stima $\sigma_0 = 0.1\text{m}$, l'esponente path-loss $\alpha = 2$, velocità $v = 0.1\text{m/step}$ e distanze di sicurezza. Ricapitolando, i parametri impostati sono:

```

pt0=[0,0]
n=[2,3,4,5,6,7,8,9,10,15,20,30]
d=[15,20,25,30]
x0=linspace(-2.5,2.5,n(i))
y0=d(j)*ones(1,n(i))
p0=[x0',y0']
sigma0=0.1
alpha=2

```

n	Percentuale di convergenza			
	$d = 15m$	$d = 20m$	$d = 25m$	$d = 30m$
2	76%	23%	1%	0%
3	75%	17%	0%	0%
4	86%	23%	2%	0%
5	92%	43%	3%	0%
6	96%	56%	9%	0%
7	99%	75%	13%	0%
8	100%	79%	44%	6%
9	99%	96%	57%	15%
10	100%	96%	58%	18%
15	100%	99%	95%	53%
20	100%	100%	98%	80%
30	100%	100%	99%	95%

Tabella 4.2: Simulazioni Monte Carlo per l'analisi della convergenza al variare della distanza e del numero dei nodi.

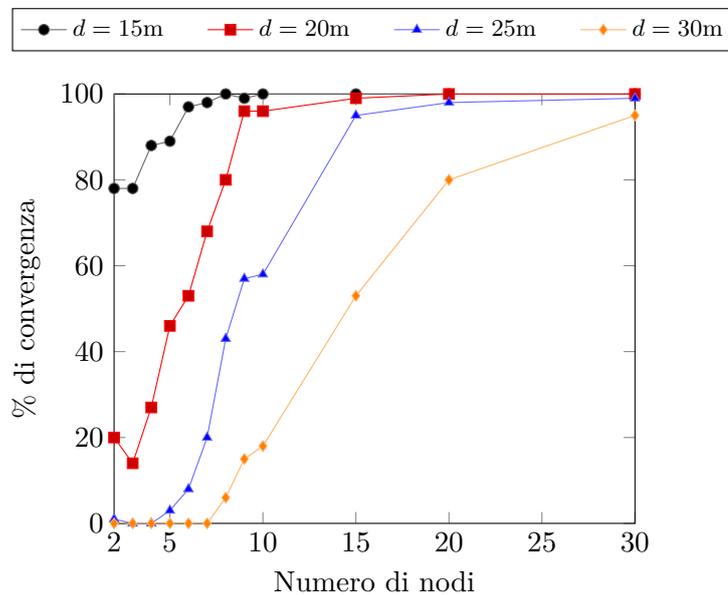


Figura 4.12: Probabilità di convergenza all'aumentare dei nodi e per diverse distanze iniziali.

```
v=0.1
d_agents=1
d_emitter=5
K=1e4
Nmc=100.
```

Dai risultati di Tabella 4.2 è possibile ricavare il plot di Figura 4.12 che permette facilmente di vedere che all'aumentare del numero dei nodi aumenta la percentuale di casi in cui l'algoritmo converge, permettendo ai droni di raggiungere le posizioni ottimali. Man mano che la distanza iniziale aumenta, la percentuale di convergenza diminuisce, a parità di numero di nodi. Questo permette di concludere che il maggiore vantaggio nell'aumentare il numero di nodi n , come accennato in precedenza, è quindi quello di rendere più facilmente raggiungibile la convergenza.

È, infatti, consigliabile aumentare il numero di nodi n quando le posizioni iniziali dei nodi siano ad elevata distanza da target e/o in presenza di forte rumore di stima.

4.3 Simulazioni in condizioni di Non-Line of Sight (NLOS)

Nell'intento di rendere il simulatore sempre più compatibile con un'applicazione pratica, si effettuano delle simulazioni creando delle regioni nel quale si ha un aumento dell'incertezza della stima della posizione come descritto in Sezione 2.3 a causa di condizioni NLOS. Nelle prime simulazioni si è scelto di ridurre lo scenario per mostrare in modo più efficace le differenze di performance che porta l'introduzione del bias nel modello di osservazione. Successivamente viene effettuato uno studio sulla convergenza variando il numero di nodi e la distanza iniziale degli UAV. Inoltre, per mostrare gli effetti causati dalle condizioni NLOS, si è scelto di utilizzare un vincolo di limitazione superiore sulla velocità, non mantenendola costante.

Nella seguente simulazione si ipotizza che l'algoritmo abbia una probabilità di detection dello stato del canale ideale $p_d = 100\%$. In questo modo, l'algoritmo di controllo è in grado di modificare i pesi A_i per rendere meno influenti le misure affette da condizioni NLOS. Per capire gli effetti che le condizioni NLOS hanno sul posizionamento dei nodi per la localizzazione dell'emettitore, si suppone di utilizzare uno stimatore ML per la stima della posizioni, in uno scenario senza ostacoli e con regioni NLOS rettangolari, create utilizzando istruzioni analoghe a quelle usate per la creazione degli ostacoli in alcune delle simulazioni precedenti. Ad esempio, per generare la zona NLOS della seguente simulazione, sono stati utilizzati i comandi:

```
cr_nlos=[5,10]
lung_h_nlos=10
```

```
alt_nlos=4
bias=0.5
```

dove `cr_nlos` è il vettore contenente il centro del rettangolo, `lungh_nlos` e `alt_nlos` sono rispettivamente la lunghezza e l'altezza della regione rettangolare e `bias` è il valore che assume σ_b^2 all'interno della zona NLOS. La simulazione viene effettuata posizionando inizialmente i nodi a $d = 15\text{m}$, con un basso rumore di stima pari a $\sigma_0 = 0.01\text{m}$ e per $n = 3, 4, 5$, così da vedere gli effetti del bias al variare del numero di nodi. I dati inseriti nel simulatore sono:

```
pt0=[0,0]
n=[3,4,5]
x0=linspace(-2.5,2.5,n(i))
y0=15*ones(1,n(i))
p0=[x0',y0']
sigma0=0.01
alpha=2
v=0.1
d_agents=1
d_emitter=5
K=1e4.
```

In Figura 4.13 si possono osservare i risultati ottenuti con i dati di cui sopra. Si è scelto di riportare solo la componente transitoria dell'andamento del Position Error Bound in quanto la regione di bias è stata posta a circa $d_b = 11\text{m}$ dall'emettitore e perché dell'andamento a regime al variare del numero di nodi si è già discusso in Sezione 4.2.2. L'effetto della regione di bias si riconosce dai salti repentini che possono essere facilmente distinti nelle curve di Figura 4.13. All'aumentare del numero dei nodi n , si nota come il salto di PEB tenda a diminuire. Questo effetto porta a dedurre che all'aumentare del numero di nodi l'effetto del bias sia più debole, rendendo la stima della posizione dell'emettitore più insensibile al bias. Infatti, maggiore è n , maggiore sarà la probabilità che ci siano nodi al di fuori della regione NLOS, quindi PEB migliore. Ricordando che si suppone di utilizzare un algoritmo di detection NLOS con probabilità di successo $p_d = 100\%$, l'algoritmo di controllo è in grado di calcolare la derivata del Position Error Bound utilizzando la varianza corretta per posizionare correttamente i nodi. Infatti, si riscontra un rallentamento dei nodi nella regione di bias che viene però compensato da una velocizzazione degli altri UAV che vanno a migliorare la prestazione di localizzazione. Quando un nodo entra nell'area NLOS, tutti gli altri modificano le loro traiettorie per compensare la misura caratterizzata da forte incertezza.

Ovviamente, l'ipotesi di avere un algoritmo di NLOS detection ideale è molto forte. Per questo motivo, si è effettuata un'analisi dei diversi comportamenti in presenza e in assenza dell'algoritmo. Questo significa calcolare le

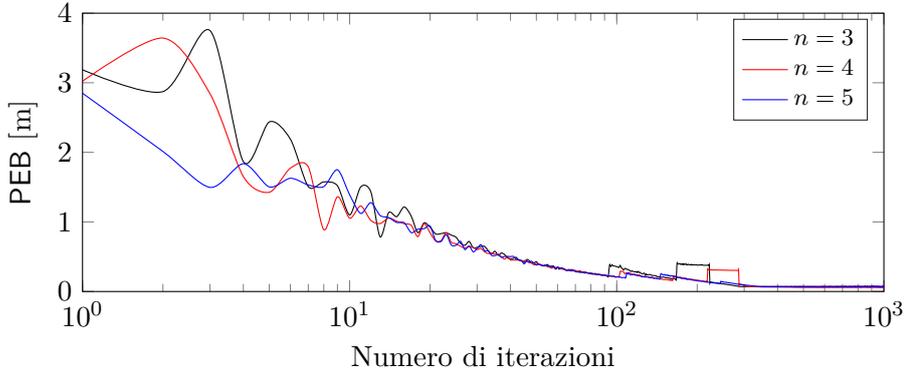


Figura 4.13: Andamento del PEB al variare del numero di nodi in presenza di regioni di bias dal valore noto.

posizioni dei nodi utilizzando rispettivamente il valore $\sigma^2 = \sigma_i^2 + \sigma_b^2$ per il caso in cui sia nota la dispersione aggiuntiva causata dalla condizione NLOS e con $\sigma^2 = \sigma_i^2$ in assenza di algoritmo di detection.

Per studiare la differenza di performance in questi due differenti casi, si mantengono le stesse impostazioni della simulazione precedente per il caso $n = 3$, andando però a spostare la regione NLOS, per far sì che solo un nodo vi entri:

```
cr_nlos=[8,10]
lungh_nlos=5
alt_nlos=10
bias=0.5.
```

Una simulazione di questo tipo, fornisce i risultati di Figura 4.14 dove sono riportate le traiettorie nel caso in cui si abbia l'algoritmo di detection NLOS attivo ed ideale (Figura 4.14a), nel caso con algoritmo non attivo (Figura 4.14b) e viene mostrato l'andamento del PEB di entrambi i casi in Figura 4.14c. In Figura 4.14a si visualizzano le traiettorie descritte utilizzando un algoritmo di detection NLOS ideale. Questo algoritmo permette di muovere i nodi esterni alla regione NLOS in modo lineare, perché il coefficiente peso A_i viene reso minore dalla peggiore qualità del link, caratterizzato da una varianza pari a $\sigma^2 = \sigma_i^2 + \sigma_b^2$:

$$A_i = \frac{\alpha^2}{2d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha + \sigma_b^2}. \quad (4.12)$$

Questo non avviene nel caso in cui l'algoritmo non ci sia come in Figura 4.14b, dove si vede come i nodi fuori dalla regione NLOS subiscano delle brusche variazioni di traiettoria dovuti al repentino aumento del Position Error Bound. I nodi, non sapendo che uno di loro si trova in una regione sfavorevole, continuano a calcolare le posizioni utilizzando i coefficienti A_i

ottenuti con $\sigma^2 = \sigma_i^2$:

$$A_i = \frac{\alpha^2}{2d_i^2} + \frac{1}{\sigma_0^2 d_i^\alpha}. \quad (4.13)$$

La differenza di performance dei due casi si può trovare in Figura 4.14c. Ciò che si riscontra è che nel caso in cui venga riconosciuta la regione NLOS, il tempo che i nodi impiegano a limitare il problema e portarsi al livello naturale del PEB è molto minore rispetto al caso in cui non vi sia l'algoritmo di detection. Questo effetto è ben visibile: la curva nera, rappresentante il caso con algoritmo ideale, presenta un picco positivo improvviso dovuto all'entrata del nodo nella zona NLOS. Benché questo UAV resti del tempo nella zona NLOS, gli altri due nodi si avvicinano all'emettitore mettendosi ben in formazione, portando a limitare velocemente il problema della condizione di Non-Line of Sight. Nella curva rossa rappresentante la performance in assenza di algoritmo di detection NLOS, invece, si ha uno stallo ad un livello di PEB alto per un numero di iterazioni maggiore. Oltre al fatto che il terzo nodo percorre più spazio all'interno della regione NLOS, anche i restanti due nodi si comportano diversamente dal primo caso. Infatti, descrivono delle curve irregolari, avvicinandosi più lentamente all'emettitore e diminuendo in modo minore il Position Error Bound.

Le curve ottenute in Figura 4.14c sono state ottenute nel caso in cui la probabilità di detection NLOS dell'algoritmo sia rispettivamente pari a $p_d = 100\%$ e a $p_d = 0\%$. Il primo caso significa che l'algoritmo è infallibile, cosa impossibile in pratica. Per ottenere curve più realistiche, è necessario modificare la probabilità di detection e renderla pari ad alcuni valori garantiti da tecnologie esistenti, ad esempio basandosi sulle prestazioni riportate in Tabella 2.1. Per limitare la probabilità di successo, è sufficiente creare nel simulatore una variabile aleatoria uniformemente distribuita e paragonare il suo valore a una soglia pari alla prestazione dell'algoritmo:

- quando il numero casuale generato ω è al di sotto della soglia di prestazione $\omega \leq p_d$, si suppone che l'algoritmo abbia funzionato correttamente e si calcolano le nuove posizioni utilizzando la varianza $\sigma^2 = \sigma_i^2 + \sigma_b^2$ per il calcolo dei coefficienti A_i , come nella (4.12);
- quando il numero casuale è superiore alla soglia di prestazione $\omega > p_d$, l'algoritmo di detection ha fallito e le nuove posizioni vengono calcolate tramite coefficienti A_i con varianza $\sigma^2 = \sigma_i^2$, come se il nodo non si trovasse all'interno della regione NLOS;

L'andamento del PEB, quindi, viene modificato al variare della performance p_d dell'algoritmo di NLOS detection. I risultati numerici ottenuti sono riportati in Figura 4.14d nel caso di $p_d = 100\%$, 75% , 50% , 0% , dove il primo ed ultimo caso corrispondono rispettivamente all'algoritmo ideale ed algoritmo assente, le cui curve sono esattamente quelle di Figura 4.14c che sono state riportate per comodità di confronto. Da Figura 4.14d si vede come al

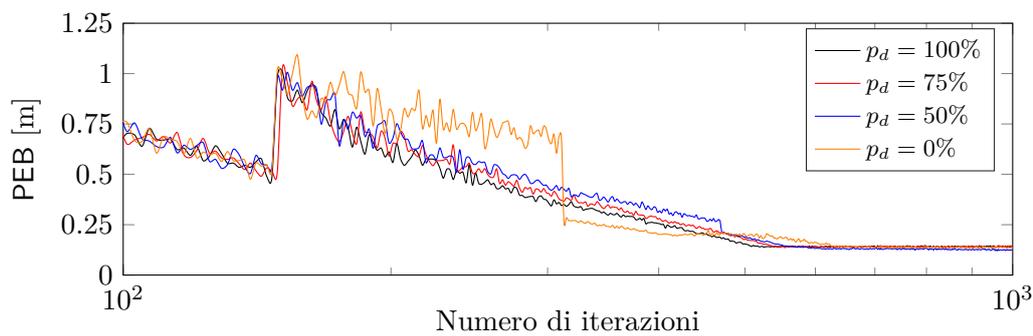
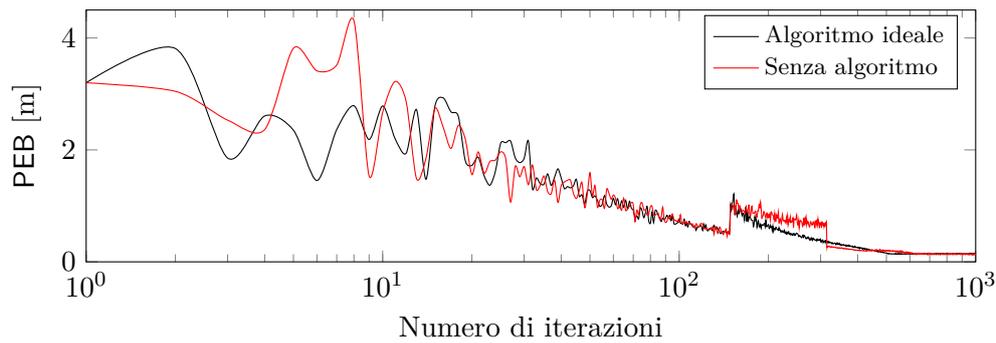
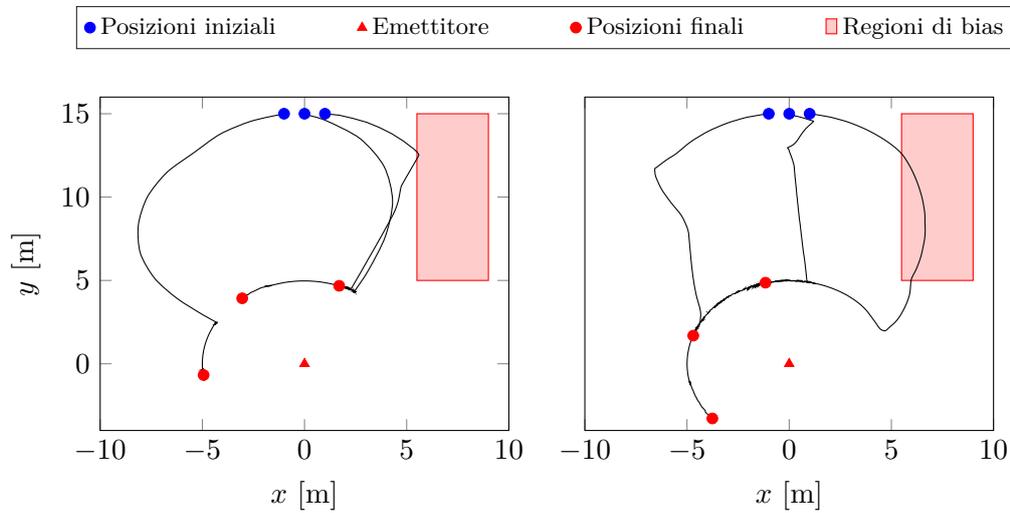


Figura 4.14: Model mismatch.

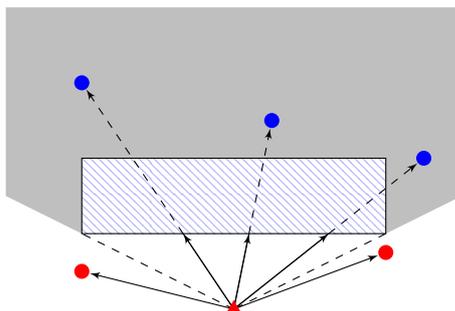


Figura 4.15: Zona NLOS creata da un ostacolo: i nodi rossi sono LOS mentre quelli blu, che si trovano nella regione grigia, sono NLOS.

diminuire della percentuale di detection il valore del PEB è complessivamente peggiore per un numero di iterazioni crescente. Questo sta ad indicare che l'effetto della condizione NLOS tende ad essere più intenso rispetto al caso di algoritmo di detection ideale, com'era ipotizzabile. Si nota però una leggera differenza nei primi casi, corrispondenti ad una buona percentuale di successo. La grande differenza, invece, si presenta se ci si focalizza sulle curve con algoritmo per confrontarle con quella senza algoritmo. Il valore del PEB in quest'ultimo caso è molto più elevato rispetto gli altri casi, anche non ideali. Questo risultato, permette di concludere che la presenza di un algoritmo di NLOS detection permette all'algoritmo di posizionamento di essere più insensibile al NLOS e per ottenere un buon miglioramento è sufficiente utilizzare algoritmi poco complessi con percentuale di detection p_d anche del 50%.

Nelle simulazioni appena mostrate, si è supposto di avere una regione rettangolare in cui un nodo al suo interno riceva una misura affetta da condizioni NLOS a causa della comunicazione NLOS. In uno scenario pratico questo tipo di condizione avviene quando un UAV si trova in una zona d'ombra, causata ad esempio da un ostacolo interposto tra esso e l'emettitore. Una descrizione grafica di uno scenario più realistico è raffigurato in Figura 4.15 dove viene rappresentata in grigio la regione Non-Line of Sight creata dall'ostacolo e dove i nodi blu riceveranno una misura affetta da bias a causa delle condizioni NLOS mentre i nodi rossi, essendo Line of Sight, non avranno questo problema.

Per simulare questo scenario, si è resa necessaria la costruzione di una semplice funzione il cui compito è quello di tracciare la retta congiungente tra l'emettitore e l' i -esimo nodo verificando che non venga incontrato un ostacolo. In questo modo il simulatore è in grado di analizzare i collegamenti con l'emettitore e discriminare quale nodo si trovi in condizione LOS o NLOS.

Chiaramente, l'effetto che porta questa concezione rispetto alla precedente è quello di creare zone NLOS molto grandi che coprono gran parte dello

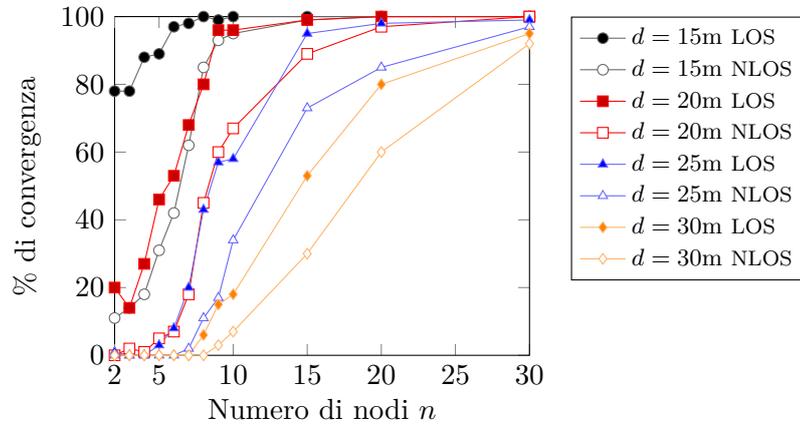
spazio. Di conseguenza il valore del Position Error Bound potrà facilmente raggiungere dei valori più elevati rispetto a quelli che ci si aspetta quando si va ad analizzare una geometria ottimale, peggiorando nettamente la prestazione nella localizzazione del target. In Figura 4.13 e Figura 4.14 gli UAV entrano nelle regioni NLOS nella fase finale del posizionamento, mentre nelle fasi iniziali del processo vengono sfruttate delle misure non affette da bias, quindi favorevoli. Purtroppo, però, più un UAV si trova lontano dall'obiettivo più è probabile che ci siano degli ostacoli che non permettono una comunicazione NLOS. La fase iniziale è anche quella più critica a livello di convergenza perché quando vengono attivati i droni, questi possono trovarsi in una formazione tale da rendere pessimo il PEB portando ad una stima della posizione del target molto sbagliata, innescando un loop che porta alla non convergenza dell'algoritmo.

Per verificare gli effetti delle condizioni NLOS sulla convergenza, sono state effettuate le stesse simulazioni Monte Carlo fatte per tracciare il grafico di Figura 4.12, aggiungendo il dato riguardante alla dispersione aggiuntiva che si verifica quando il collegamento è NLOS. Nello scenario simulato vengono disposti un numero n di nodi in linea posti a diverse distanze iniziali d dal target da localizzare, supponendo una dispersione aggiuntiva nella stima della posizione portato dalla regione Non-Line of Sight σ_b . Per la creazione della regione NLOS viene posto un ostacolo di forma rettangolare a $d_{obs} = 10\text{m}$ dall'emettitore, in modo che inizialmente tutti i nodi ricevano una misura NLOS ma che a regime riescano a portarsi a ridosso dell'emettitore portandosi in uno scenario LOS.

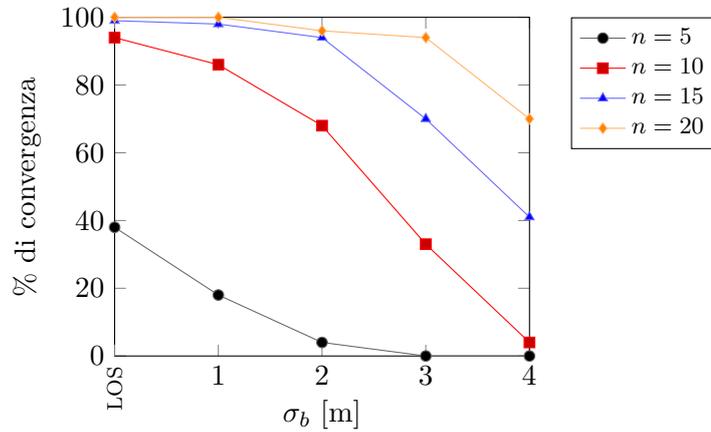
In Figura 4.16 sono riportati i risultati delle simulazioni di convergenza. In Figura 4.16a vengono riportate le percentuali di convergenza al variare di n e d supponendo una dispersione aggiuntiva costante $\sigma_b = 2\text{m}$ e le percentuali di Figura 4.12 riguardanti il caso LOS. A causa delle condizioni NLOS, le percentuali di convergenza vengono decisamente abbassate quando il numero di nodi è basso, mentre l'effetto è minore quando gli UAV utilizzati aumentano. Dai risultati numerici è interessante notare che la probabilità di convergenza viene peggiorata maggiormente dal bias quando le distanze di partenza sono più basse, mentre l'effetto negativo è meno incisivo all'aumentare della distanza. Questi risultati suggeriscono che la distanza iniziale degli UAV è il parametro più critico a livello di percentuale di convergenza.

Le curve in Figura 4.16b mettono in relazione la percentuale di convergenza al variare del numero di nodi e della varianza legata alle condizioni NLOS σ_b ad una distanza iniziale fissata a $d = 20\text{m}$. I risultati mostrano come l'aumentare della dispersione NLOS corrisponda ad una perdita in termini di percentuale di convergenza dell'algoritmo. È interessante notare come la perdita di performance venga ridotta quando il numero di nodi n aumenta. Questo fattore suggerisce di usare un maggior numero di nodi quando il bias è elevato.

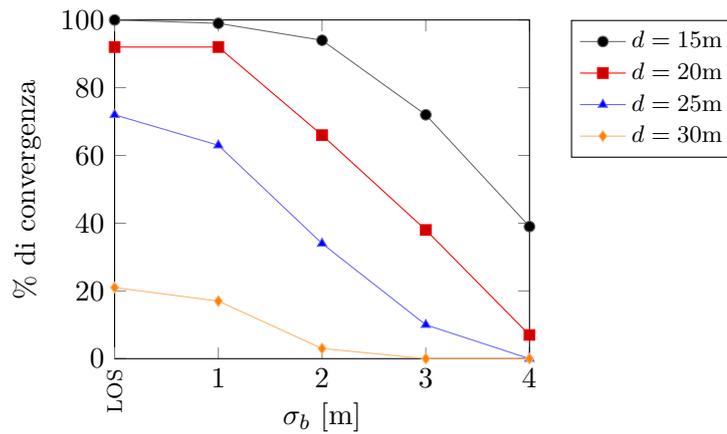
Il plot di Figura 4.16c, invece, sono rappresentati i risultati di simulazione



(a) Nodi e distanza iniziali variabili, $\sigma_b = 2m$.



(b) Distanza iniziale $d = 20m$, nodi e bias variabili.



(c) Numero di nodi $n = 10$, distanza iniziale e bias variabile.

Figura 4.16: Percentuali di convergenza: comparazione LOS e NLOS.

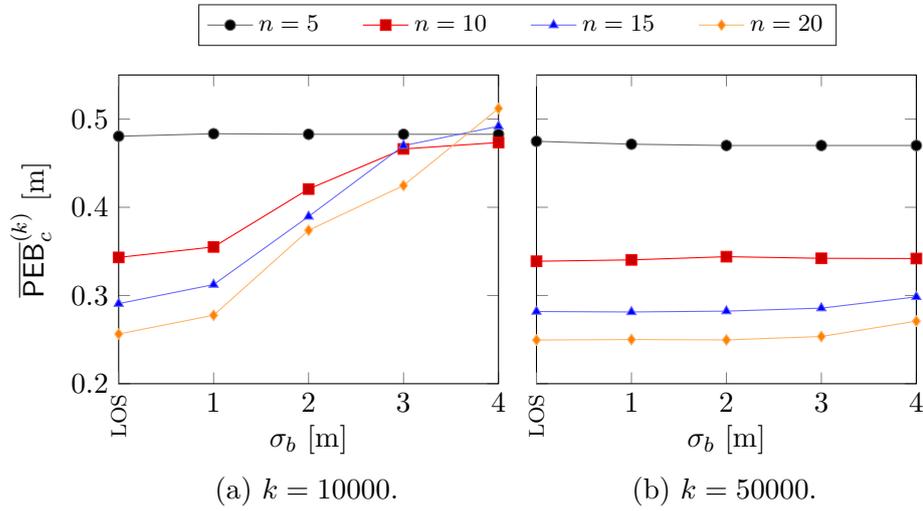


Figura 4.17: PEB al variare del bias con distanza iniziale $d = 20\text{m}$ per diversi n dopo $k = 10000$ e $k = 50000$ iterazioni.

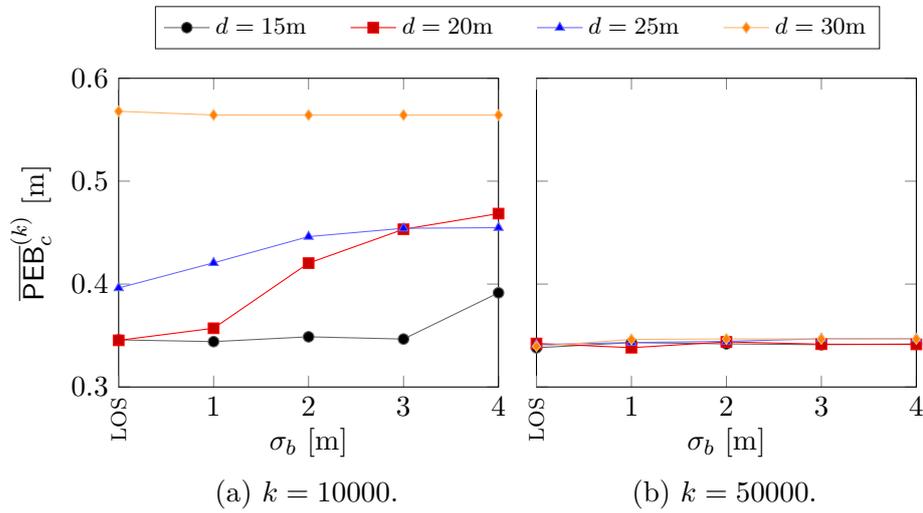


Figura 4.18: PEB al variare del bias per diverse d con $n = 10$ dopo $k = 10000$ e $k = 50000$ iterazioni.

fissando il numero di nodi $n = 10$ e facendo variare d e σ_b . Si evince come la percentuale di convergenza peggiori molto rapidamente quando entrambe le quantità analizzate aumentano.

Le rappresentazioni appena descritte, tengono conto solo della convergenza dell'algoritmo ma non danno informazioni sull'accuratezza con cui può essere stimata la posizione del target. Per lo stesso scenario, sono stati riportati alcuni valori di PEB ottenuti in media al variare dei parametri n , d , e σ_b . A causa delle variabili aleatorie, il modo migliore per fornire un'indicazione sulla prestazione è quello di utilizzare la media aritmetica del Position Error Bound utilizzando la formula:

$$\overline{\text{PEB}}_c^{(k)} = \frac{1}{N_{\text{MC}}} \sum_{m=1}^{N_{\text{MC}}} \text{PEB}_m^{(k)} \quad (4.14)$$

dove k è l'indice rappresentante il time step, m è l'indice corrispondente alla m -esima iterazione Monte Carlo e N_{MC} è il numero di simulazioni Monte Carlo per configurazione. In Figura 4.17-4.18 sono rappresentati i valori di Position Error Bound medio $\overline{\text{PEB}}_c^{(k)}$ per $k = 10000$ e $k = 50000$, così da cogliere quanto le condizioni NLOS incidano sulla velocità di convergenza alla formazione ottimale dell'algoritmo. In Figura 4.17a sono descritti i risultati del PEB dopo $k = 10000$ iterazioni nel caso in cui vengano fatti partire a distanza iniziale $d = 20\text{m}$ e fatti variare il numero di nodi e σ_b , mentre in Figura 4.17b è possibile vedere i risultati a regime. Come ci si aspettava, Figura 4.17b mostra che all'aumentare del numero di nodi si ottengono prestazioni migliori a livello di Position Error Bound e restano costanti, da cui si deduce che la formazione dei droni sia quella ottimale. Inoltre, i valori del PEB tendono al valore ottenuto per il caso LOS, dove il rallentamento causato dalla comunicazione NLOS non è presente. Da Figura 4.17a è interessante notare che l'aumentare del valore di σ_b porta ad un rallentamento della velocità dei droni e, di conseguenza, ad un rallentamento della convergenza del sistema alla formazione ottimale. Dalle curve si evince che l'effetto del rallentamento viene accentuato all'aumentare dei nodi da come si evince dalle pendenze delle curve.

Un risultato simile si ottiene in Figura 4.17 fissando n e facendo variare la distanza iniziale d e la varianza di dispersione aggiuntiva σ_b dovuta alle condizioni NLOS. Il rallentamento dovuto al link NLOS è chiaramente visibile in Figura 4.18a, dove all'aumentare di σ_b le curve tendono ad alzarsi. È importante evidenziare che l'andamento del PEB all'aumentare della distanza d le curve si fanno via via sempre più alte: questo era ipotizzabile, in quanto per un numero di iterazioni fisse a velocità costante, tutti i nodi percorrono lo stesso spazio quindi quelli più distanti inizialmente avranno una performance peggiore perché saranno meno vicini all'emettitore. In Figura 4.18b sono riportati i valori a regime che, come ci si aspettava, sono gli stessi per tutti gli scenari perché l'unica differenza riguarda il tempo con cui i nodi si avvicinano all'emettitore.

Simulazioni con algoritmo distribuito

NEL seguente capitolo vengono riportate delle simulazioni per le quali è stato applicato l'algoritmo distribuito di navigazione degli UAV. Analogamente a quanto fatto per la versione centralizzata, l'algoritmo iterativo distribuito è stato progettato per essere fermato una volta raggiunto un numero massimo di iterazioni K . Per semplicità, vengono riportati unicamente i risultati ottenuti con target non in movimento e si suppone che gli UAV siano dotati di un buffer nel quale viene salvata l'ultima posizione conosciuta degli altri nodi. Si sceglie di utilizzare un modello Booleano a disco con raggio d_{hop} , trascurando lo shadowing. In questo scenario, l'equazione da minimizzare è data dalla (3.124), riportata per comodità:

$$\begin{aligned}
 \mathbf{p}_i^* = \arg \min_{\mathbf{p}_i \in \mathbb{R}^2} \quad & \text{PEB}_i \left(\hat{\mathbf{p}}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) \\
 \text{tale che} \quad & \begin{aligned}
 1) \quad & d_{ij} \geq d_{\min}, \quad j = 1, \dots, n, \quad j \neq i \\
 2) \quad & d_{it} \geq d_{\min_t} \\
 3) \quad & \mathcal{T}_i \cap \mathcal{O} = \emptyset \\
 4) \quad & \left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|_2 = v
 \end{aligned}
 \end{aligned} \tag{5.1}$$

dove il PEB calcolato dall' i -esimo UAV è dato dalla (2.70):

$$\text{PEB}_i \left(\hat{\mathbf{p}}_{t_i}^{(k)}; \mathbf{q}_i^{(k)} \right) = \sqrt{\frac{A_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)}}{S_{x,i}^{(k)} S_{y,i}^{(k)} - \left(S_{xy,i}^{(k)} \right)^2}} \tag{5.2}$$

con:

$$S_{x,i}^{(k)} = A_i^{(k)} \left(c_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(c_i^{(k-h_{ij}+1)} \right)^2 \quad (5.3)$$

$$S_{y,i}^{(k)} = A_i^{(k)} \left(s_i^{(k)} \right)^2 + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} \left(s_i^{(k-h_{ij}+1)} \right)^2 \quad (5.4)$$

$$S_{xy,i}^{(k)} = A_i^{(k)} c_i^{(k)} s_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^n A_j^{(k-h_{ij}+1)} c_i^{(k-h_{ij}+1)} s_i^{(k-h_{ij}+1)}. \quad (5.5)$$

Nel seguito del capitolo verranno analizzati risultati ottenuti da simulazioni variando parametri legati unicamente all'algorithmo distribuito in quanto tutti gli altri aspetti sono già stati discussi in Capitolo 4.

Il capitolo è strutturato come segue: i) in Sezione 5.1 vengono descritti gli aspetti che differenziano il comportamento degli UAV nel caso distribuito rispetto al centralizzato, riportando una singola realizzazione di simulazione; ii) in Sezione 5.2 viene effettuata un'analisi statistica con simulazioni Monte Carlo per studiare i risultati medi dell'algorithmo distribuito. In particolare, nelle sottosezioni vengono analizzati gli effetti di alcuni parametri fondamentali dell'algorithmo distribuito come il communication range, il numero di hops massimo e il numero di nodi.

5.1 Singola realizzazione con algorithmo distribuito

In questa prima simulazione si riporta una singola realizzazione dell'algorithmo distribuito per analizzarne il comportamento con $n = 4$ UAV, ipotizzando che siano caratterizzati da un communication range d_{hop} costante e uguale per tutti. A causa delle componenti casuali introdotte nel rumore di stima, infatti, per ottenere dei risultati statisticamente validi è necessario effettuare delle simulazioni di Monte Carlo, ripetendo molte volte la stessa simulazione per estrapolarne il comportamento medio. Tuttavia, questa prima simulazione ha lo scopo di mostrare alcuni aspetti relativi alla differenza fra quello che si verifica utilizzando l'algorithmo centralizzato.

Si simula un ambiente caratterizzato dalla presenza di ostacoli rettangolari, con posizioni e forma non nota a priori, che possono generare link di tipo NLOS quando sono posti in corrispondenza dell'intercetta tra nodo e target. Per eseguire questa simulazione sono stati usati i seguenti parametri:

```
pt0=[0,0]
n=4
x0=linspace(-10,10,n)
y0=150*ones(1,n)
```

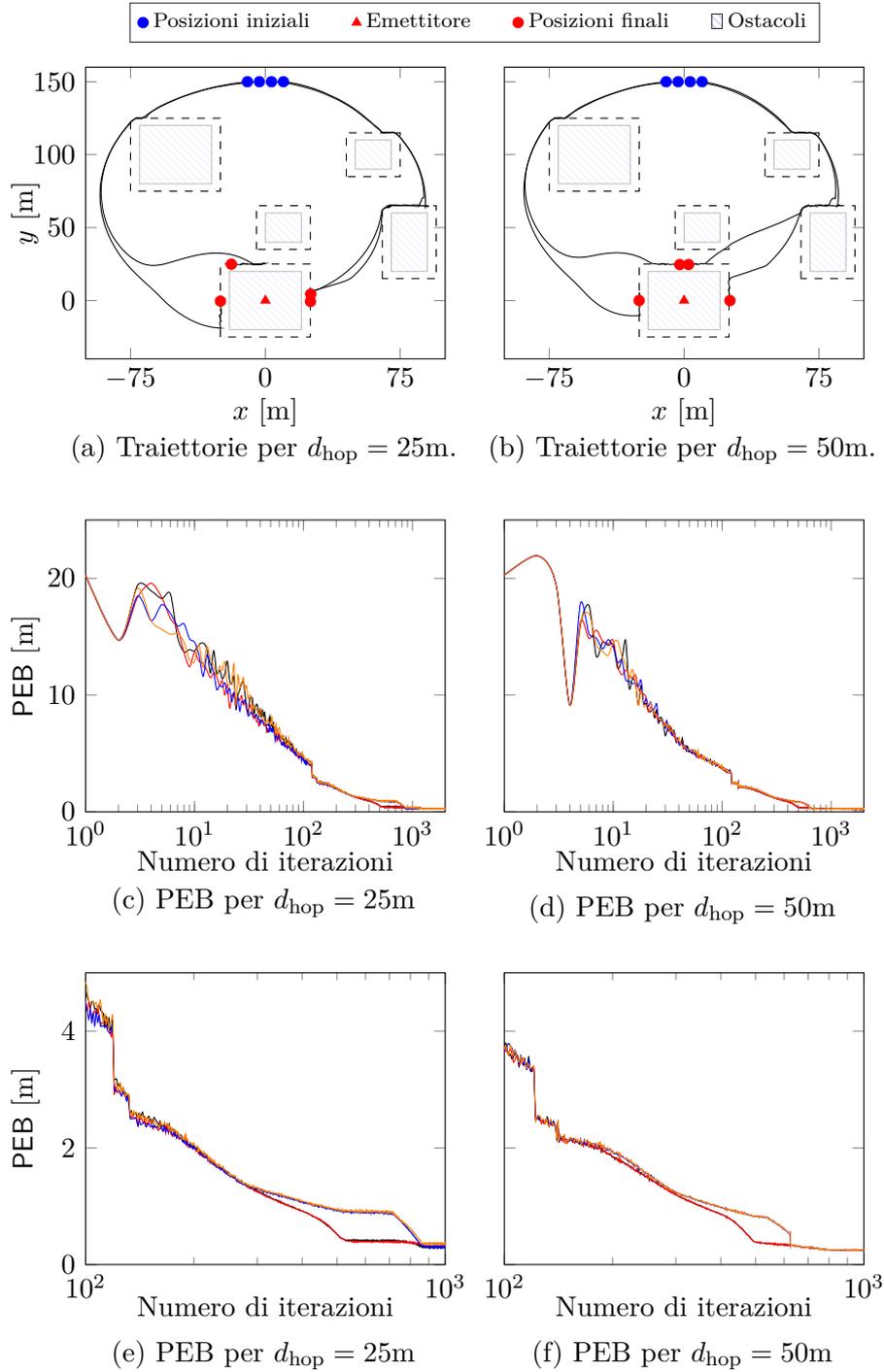


Figura 5.1: Traiettorie e PEB di ciascun UAV per due connection ranges.

```

p0=[x0' y0']
dhop=[25,50]
hop_max=3
sigma0=0.01
alpha=2
v=0.5
K=2000
Nmc=1
d_obs=5
d_agents=5
d_emitter=15
cr=[pt0;-50,100;60,100;80,40;10,50]
lunghe=[40,40,20,20,20]
alt=[40,40,20,40,20]
bias=2

```

dove vengono evidenziate le nuove variabili d_{hop} e hop_max necessarie per creare l'ambiente di simulazione per l'algoritmo distribuito.

Analizzando Figura 5.1a e Figura 5.1b, è possibile vedere le differenze che il variare di d_{hop} porta alle traiettorie descritte dagli UAV. Come si era già intuito dalle rappresentazioni grafiche delle traiettorie risultanti dalle simulazioni nel Capitolo 4, per motivi geometrici, i nodi tendono a muoversi in clusters. Per questa ragione, è stato possibile verificare che nelle simulazioni ottenute applicando l'algoritmo distribuito il raggio incide notevolmente sulle performance. Questi ragionamenti per il caso con $n = 4$ portano gli UAV a muoversi a coppie e ognuna di esse avrà i due nodi con un valore molto simile di PEB, ma il valore può differire notevolmente dall'altra coppia. Inoltre, a differenza del caso centralizzato dove la stima della posizione del target $\hat{\mathbf{p}}_t$ effettuata dall'unità centrale è unica, per l'algoritmo distribuito questo non vale. Infatti, ogni UAV è costretto ad effettuare una sua stima della posizione del target $\hat{\mathbf{p}}_{t_i}$ che è in generale diversa per tutti. Questi ragionamenti spiegano l'andamento del PEB delle Figura 5.1c-5.1f, dove compaiono 4 linee corrispondenti al numero di nodi n . In prima approssimazione, è possibile analizzare il transitorio di Figura 5.1c e Figura 5.1d, dove si nota come all'aumentare del communication range la differenza di PEB vista da ciascun UAV tende a decrescere. Questo è intuibile dal fatto che all'aumentare di d_{hop} , la performance dell'algoritmo distribuito tende alla sua versione centralizzata, in quanto tutti i nodi riescono a comunicare la propria posizione tra loro, con l'unica differenza che ogni UAV effettua una stima della posizione del target.

Un aspetto molto interessante può essere dedotto da Figura 5.1e e Figura 5.1f, dove viene rappresentato l'andamento del PEB nelle ultime iterazioni precedenti alla condizione di regime. Queste rappresentazioni, oltre a confermare che gli UAV si sono mossi in coppia, permettono di osservare

che la differenza di prestazione tra coppie di UAV può essere abbastanza elevata. In particolare, quando il communication range è basso la differenza di performance è ampia mentre tende ad abbassarsi aumentando il raggio del disco d_{hop} . Un altro effetto positivo portato dall'aumento di connection range è dato dalla maggiore velocità con cui i nodi arrivano a convergenza. Infatti, un valore di d_{hop} sufficientemente elevato significa che gli UAV possono calcolare la correzione di posizione utilizzando posizioni aggiornate, producendo un controllo migliore rispetto a quello ottenuto con posizioni out-of-date. Per questa ragione il controllo di posizione sarà più efficace, ottenendo un beneficio a livello di performance.

5.2 Analisi statistica dell'algoritmo distribuito

A differenza del caso centralizzato, per lo studio di convergenza dell'algoritmo distribuito si rende necessaria la definizione di una nuova metrica. Questo è dovuto al fatto che il precedente criterio di convergenza era riferito all'impossibilità dell'unità di elaborazione centrale del calcolo del PEB, uguale per tutti i nodi. Nell'algoritmo distribuito, invece, ogni UAV lavora in modo proprio per cui occorre un'analisi più approfondita. Infatti, oltre al problema dell'impossibilità di calcolare il PEB, occorre anche controllare se tutti gli UAV sono giunti ad una performance di localizzazione simile.

Per derivare la metrica, si è deciso di utilizzare la definizione statistica di *Cumulative Density Function* (CDF). Per una variabile aleatoria X continua a valori reali, tale funzione $F : \mathbb{R} \rightarrow [0, 1]$ può essere espressa analiticamente con:

$$F(x) = p(X \leq x) = \int_{-\infty}^x f(t) dt \quad (5.6)$$

dove $F(x)$ è la CDF e $p(X \leq x)$ è la probabilità che la variabile aleatoria X sia minore di una soglia x .

In tal senso, si definisce una *success rate* $\text{SR}^{(k)}$ relativa al k -esimo time step:

$$\text{SR}^{(k)} = \frac{1}{nN_{\text{MC}}} \sum_{m=1}^{N_{\text{MC}}} \sum_{i=1}^n \mathbf{1} \left(\text{PEB}_{i_m} \left(\hat{\mathbf{p}}_{t_{i_m}}^{(k)}; \mathbf{q}_{i_m}^{(k)} \right) \leq \text{PEB}_{\text{goal}} \right) \quad (5.7)$$

dove N_{MC} sono le iterazioni di Monte Carlo, $\hat{\mathbf{p}}_{t_{i_m}}^{(k)}$ è la stima della posizione del target dell' i -esimo UAV alla m -esima iterazione di Monte Carlo, PEB_{goal} è la soglia per il quale si considera un nodo giunto a convergenza e $\mathbf{1}(x)$ è la funzione indicatore definita da:

$$\mathbf{1}(x) = \begin{cases} 1, & \text{se la condizione } x \text{ è vera} \\ 0, & \text{se la condizione } x \text{ è falsa.} \end{cases} \quad (5.8)$$

La metrica $\text{SR}^{(k)}$, ad ogni time-slot k , tramite $\sum_{i=1}^n \mathbf{1}(x)$ calcola la somma di UAV giunti ad un PEB tale da essere considerato convergente, ed effettua

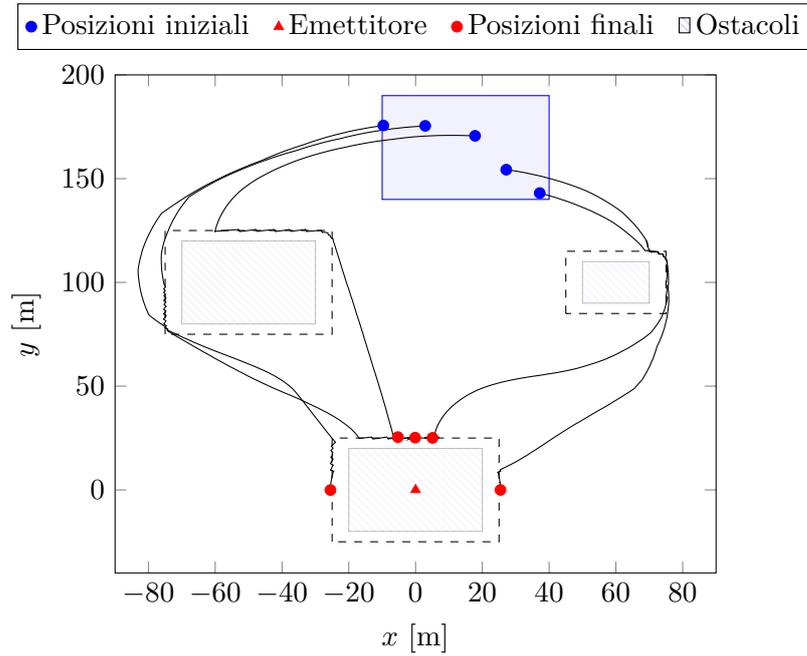


Figura 5.2: Scenario dove viene evidenziata la regione dedicata agli UAV per la partenza.

la media aritmetica rispetto alla m -esima iterazione di Monte Carlo. Fissata una soglia PEB_{goal} , tramite la metrica $SR^{(k)}$ è possibile analizzare il tasso di convergenza e la velocità di convergenza al variare di parametri di simulazione. Di fatto, la $SR^{(k)}$ rappresenta la funzione di distribuzione cumulativa empirica.

Un altro aspetto da considerare è il numero di curve generate dal simulatore. Ogni UAV fornirà una diversa prestazione, quindi per n nodi si avranno n andamenti del PEB. Per questo motivo, per valutare la prestazione complessiva del sistema di nodi, occorre utilizzare:

$$\overline{PEB}_d^{(k)} = \frac{1}{nN_{\text{MC}}} \sum_{m=1}^{N_{\text{MC}}} \sum_{i=1}^n PEB_{i_m}^{(k)} \quad (5.9)$$

dove viene effettuata una media aritmetica rispetto il numero di iterazioni N_{MC} e al numero di nodi n per ogni time step k . Questa operazione permette di ottenere la performance media del sistema di UAV utilizzati in simulazione.

Nelle seguenti simulazioni, verranno analizzate le prestazioni medie dell'algoritmo al variare del communication range, del numero di nodi e del numero di hop massimi. Applicando l'algoritmo ad uno scenario caratterizzato da ostacoli con conseguente possibilità di comunicazione NLOS sono stati fissati i parametri legati alla varianza dell'errore di stima, distanze di

sicurezza, velocità, numero di iterazioni di Monte Carlo, ostacoli e numero di time step massimo, riassunti in:

```
pt0=[0,0]
sigma0=0.01
alpha=2
v=1
K=2000
Nmc=100
d_obs=5
d_agents=0.5
d_emitter=10
cr=[pt0;-50,100;60,100]
lungh=[40,40,20]
alt=[40,40,20]
```

sono stati posizionati i nodi in posizioni casuali all'interno di una zona stabilita. Infatti, per ogni m -esima iterazione di Monte Carlo, gli UAV vengono posizionati utilizzando:

```
x0=-10+50*rand(1,n)
y0= 140+50*rand(1,n)
p0=[x0',y0']
```

dove la funzione `rand` di MATLAB 2017a crea numeri casuali distribuiti uniformemente nell'intervallo $[0, 1]$. Così facendo si avranno coordinate limite date da:

$$x_{\min} = -10, \quad x_{\max} = 40, \quad y_{\min} = 140, \quad y_{\max} = 190. \quad (5.10)$$

Un esempio di una realizzazione dell'algoritmo distribuito applicato a posizioni di partenza random è raffigurata in Figura 5.2, dove è evidenziata la regione rettangolare in cui si trovano gli UAV per $k = 0$. Grazie alla configurazione degli ostacoli, inizialmente i nodi saranno caratterizzati da comunicazione LOS per limitare il problema del loop degenerativo della stima iniziale e soffermarsi unicamente su aspetti legati alle performance. Da Figura 5.2 si nota, inoltre, che le simulazioni sono state effettuate in condizioni di target statico.

Questa configurazione è quella utilizzata per tutte le simulazioni che seguiranno, dove verranno dati i parametri appena descritti per fissati. Per descrivere le differenze fra una simulazione e l'altra, verranno forniti al lettore unicamente i parametri in esame per la data simulazione.

5.2.1 Dipendenza dal communication range

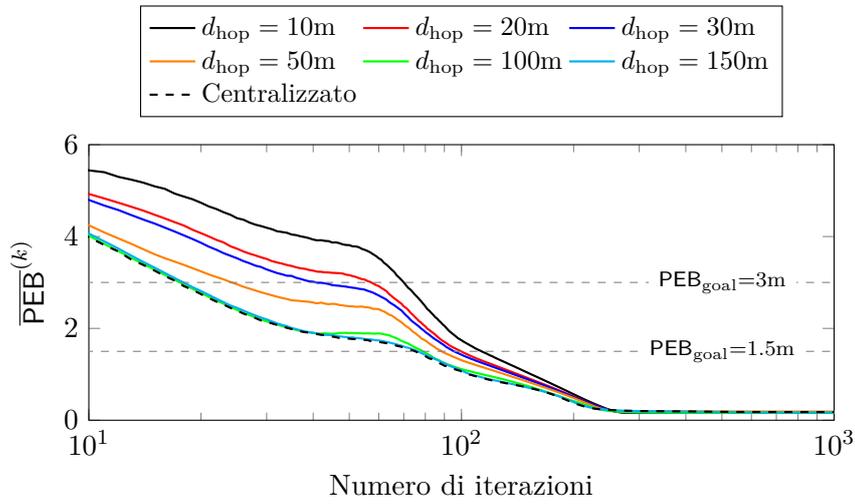
I primi risultati analizzati sono relativi ad una simulazione atta a studiare gli effetti che il raggio di comunicazione finito ha sulla performance di localizzazione. Gli andamenti delle curve rappresentate in Figura 5.3 sono state ottenute fissando il numero di nodi $n = 10$ e il numero massimo di hops $h_{\max} = 1$, permettendo unicamente lo scambio diretto di informazioni basandosi sulle matrici \mathcal{A}_{i_1} generate dagli UAV utilizzando la (3.119). I parametri di simulazione sono riassunti da:

```
dhop=[10,20,30,50,100,150]
n=10
hmax=1
PEBgoal=[1.5,3]
```

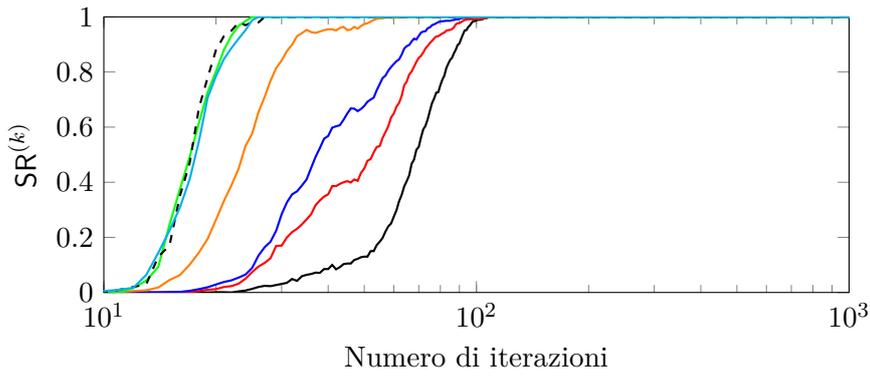
ai quali vengono aggiunti i parametri per effettuare un'altra simulazione con l'algoritmo centralizzato (dove si assume $d_{\text{hop}} = \infty$ e $\hat{\mathbf{p}}_{t_i} = \hat{\mathbf{p}}_t$) per comparare le prestazioni dei due algoritmi.

In Figura 5.3a vengono riportate le curve relative al PEB medio calcolato dallo stormo di UAV mediato sulle iterazioni di Monte Carlo $\overline{\text{PEB}}_d^{(k)}$ in funzione del time step k . Come ci si aspettava, all'aumentare del communication range d_{hop} si ha un miglioramento di prestazione complessivo, partendo già dai primi istanti temporali. A causa delle dimensioni dello scenario, si nota che il guadagno portato dall'aumento di d_{hop} tende a diminuire oltre ad un certo valore. Va notato che, come ci si aspettava, l'algoritmo centralizzato ha prestazioni migliori rispetto a quelle dell'algoritmo distribuito. La differenza è ampia quando il raggio del disco Booleano è piccolo e i nodi sono costretti ad utilizzare per molto tempo posizioni non aggiornate calcolando aggiornamenti di posizione non ottimi. Questa peggior performance viene ridotta all'aumentare di d_{hop} , dove le prestazioni tendono a raggiungere quelle ottenute trascurando i vincoli di latenza come nell'algoritmo centralizzato. Anche se il transitorio vede un algoritmo notevolmente migliore, a regime le performance tendono ad uniformarsi allo stesso valore, dove i due algoritmi tendono ad avere lo stesso PEB.

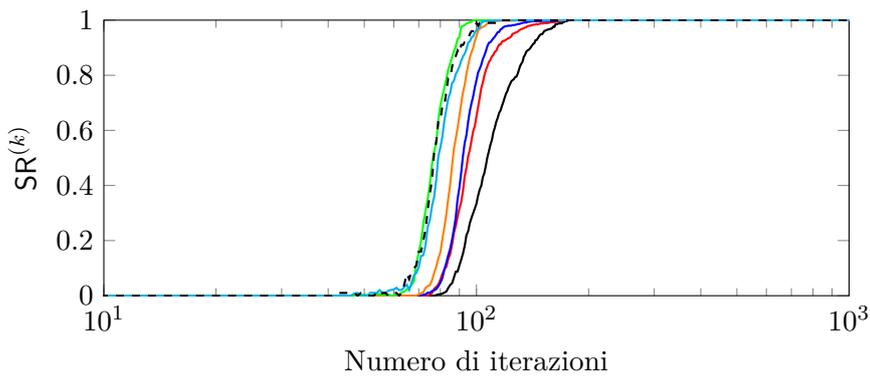
In Figura 5.3a vengono anche espresse le soglie utilizzate per ottenere le curve di success rate $\text{SR}^{(k)}$. In particolare si è scelto di studiare la velocità con cui l'algoritmo converge per $\text{PEB}_{\text{goal}} = 1.5\text{m}, 3\text{m}$. Da questo plot risulta ancor più evidente il rallentamento nel raggiungimento della soglia di convergenza quando il raggio di copertura è piccolo. Chiaramente gli UAV convergono molto prima al valore $\text{PEB}_{\text{goal}} = 3\text{m}$ rispetto a $\text{PEB}_{\text{goal}} = 1.5\text{m}$, spiegando il posizionamento dei fasci di curve. Un aspetto interessante deriva dalla pendenza delle curve. Si può vedere come per il fascio di curve ottenuto per $\text{PEB}_{\text{goal}} = 3\text{m}$ la pendenza sia di un valore più basso rispetto al secondo fascio. Questo porta a concludere che il parametro d_{hop} crei delle difficoltà



(a) $\overline{PEB}_d^{(k)}$ in funzione di d_{hop} .



(b) Success rate in funzione di d_{hop} con $PEB_{goal} = 3m$.



(c) Success rate in funzione di d_{hop} con $PEB_{goal} = 1.5m$.

Figura 5.3: Performance e convergenza dell' algoritmo distribuito al variare di d_{hop} .

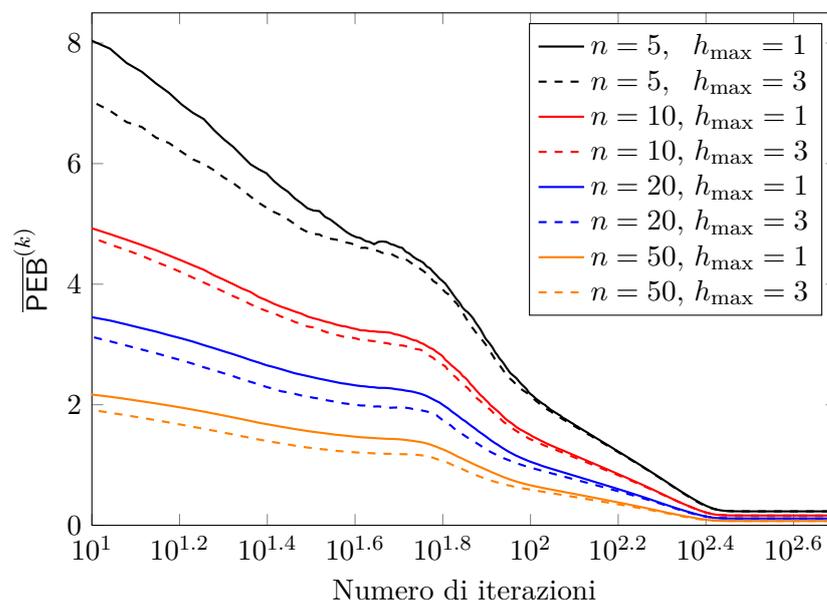


Figura 5.4: Andamento del PEB medio al variare del numero dei nodi n e del numero massimo di hops h_{max} con $d_{hop} = 20m$

soprattutto negli istanti iniziali, causando un rallentamento nel transitorio per poi incidere in minor misura nei pressi del regime.

5.2.2 Dipendenza dal numero massimo di hops

È interessante indagare anche come il numero di hop massimi incida sulla performance dell'algoritmo di localizzazione per vedere se può avere senso effettuare uno studio approfondito sul protocollo di routing. Infatti, si può ipotizzare che l'utilizzo di un protocollo di routing complesso porti alla necessità di UAV con capacità di calcolo superiore e/o batterie più performanti, portando ad un aumento del costo dello strumento, nonché un considerevole traffico di rete. Questa simulazione è stata ideata per fornire un'idea sul trade-off che può venirsi a creare nella scelta del drone che, in ogni caso, dipende fortemente dall'applicazione.

In simulazione sono state analizzate le prestazioni per uno scenario in cui siano permesse unicamente le comunicazioni tra nodi all'interno del communication range (single-hop) e uno scenario in cui siano abilitati collegamenti multi-hop attraverso una funzione di relay, nell'ipotesi che gli UAV siano in grado di svolgere. Fissando il communication range, sono state effettuate delle simulazioni al variare del parametro h_{max} e il numero di nodi, utilizzando:

```
dhop=20
n=[5,10,20,50]
```

`hmax=[1,3]`.

Le curve relative al $\overline{\text{PEB}}_d^{(k)}$ sono riportate in Figura 5.4 dove vengono rappresentate con lo stesso colore gli andamenti legati allo stesso numero di UAV, con la linea continua il caso con $h_{\max} = 1$ mentre con la linea tratteggiata il caso con $h_{\max} = 3$. Questo plot permette di vedere la differenza di performance dell'algoritmo al variare del numero massimo di hops. Abilitando i collegamenti multi-hop, si nota un miglioramento nel transitorio delle curve, fino a convergere a regime ad un valore pressoché identico. Analizzando i valori di $\overline{\text{PEB}}_d^{(k)}$ per un basso numero di time steps k , si può vedere come il principale miglioramento nell'utilizzo di un numero massimo di hops più elevato si ottenga per un numero limitato di nodi. Questo è dovuto al fatto che per un basso numero di nodi, la possibilità che questi si trovino ad una distanza tale da non instaurare il collegamento aumenta a causa della minor densità di UAV. Quest'ultimo risultato suggerisce l'impiego di protocolli di routing più avanzati nel caso in cui il numero di UAV impiegati non sia molto elevato, mentre potrebbe semplificare l'unità di calcolo dei droni quando questi sono in numero maggiore per limitare i costi.

5.2.3 Dipendenza dal numero di UAV

Quest'ultima simulazione permette di analizzare la performance dell'algoritmo distribuito al variare unicamente del parametro n , fissando communication range e numero di hops massimi. Più precisamente sono stati utilizzati i seguenti parametri:

```
dhop=20
n=[3,5,10,15,20,50,100,200]
hmax=1
PEBgoal=[0.5,1.5]
```

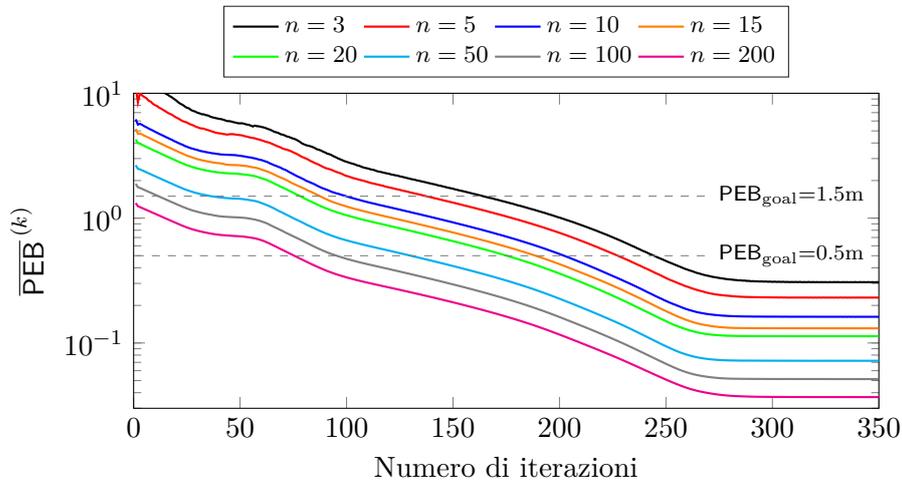
per il calcolo delle curve di $\overline{\text{PEB}}_d^{(k)}$ e $\text{SR}^{(k)}$, riportate in Figura 5.5.

Analizzando le curve di PEB medio ottenute e raffigurate in Figura 5.5a è facile intuire come all'aumentare si ottenga un generale aumento della performance, sia a regime che nel transitorio. Questo comportamento è dovuto all'aumento di densità di nodi che permette di avere molte posizioni aggiornate al generico UAV anche durante il transitorio, quando i nodi si dividono in cluster per il raggiungimento della posizione migliore. È interessante notare come all'aumentare del numero di nodi avvenga una sorta di traslazione verso il basso delle curve. Tuttavia, la differenza di prestazione tende a decrescere quando il numero di UAV è molto elevato. Focalizzando l'attenzione sulle curve relative a $n = 3$ e $n = 5$ si può notare che il miglioramento apportato aumentando di due nodi è circa lo stesso ottenuto nel passaggio da $n = 100$ a $n = 200$, nonostante la differenza molto più elevata in termini di numero di UAV.

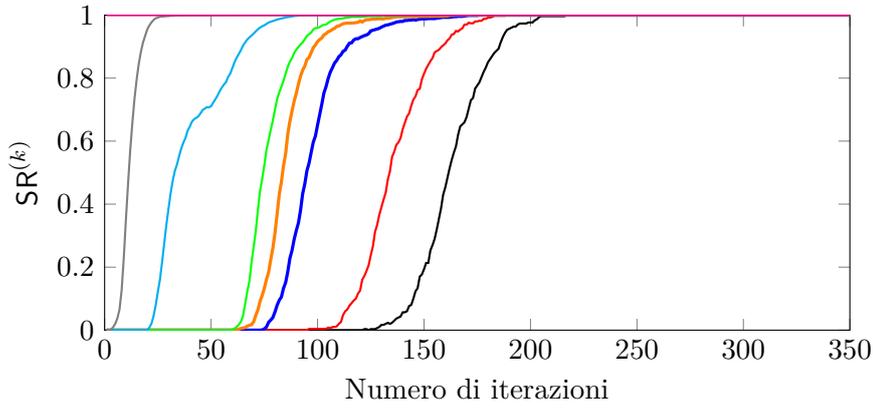
Il miglioramento di performance all'aumentare degli UAV si può notare anche nelle Figura 5.5b e Figura 5.5c relative ad un $\text{PEB}_{\text{goal}} = 1.5\text{m}, 0.5\text{m}$. Partendo da una soglia più alta, si nota come le curve siano tutte giunte al massimo valore di $\text{SR}^{(k)}$ per un basso numero di time steps k , che tendono a diminuire ulteriormente all'aumentare dei nodi. Ad esempio, nella configurazione $n = 200$ il PEB medio risulta essere minore della soglia già all'iterazione iniziale.

Abbassando la soglia a $\text{PEB}_{\text{goal}} = 0.5\text{m}$, i risultati di success rate tendono a traslare quasi rigidamente verso un numero di iterazioni maggiore, mantenendo il comportamento descritto in precedenza.

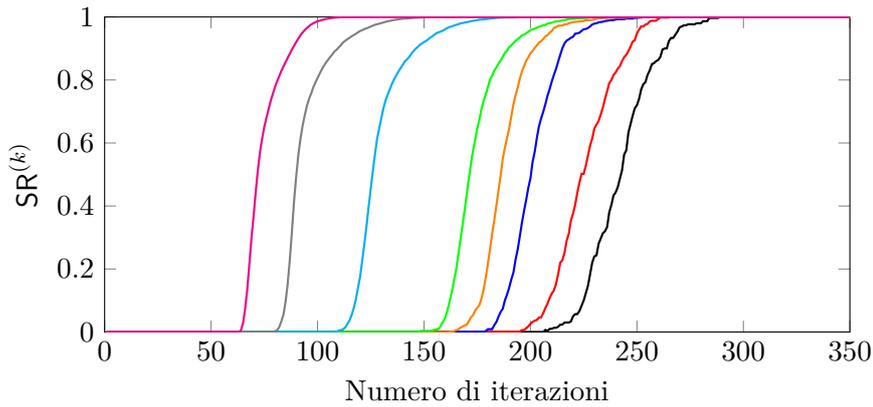
I risultati appena descritti in questo capitolo, sono stati selezionati e inseriti nell'articolo "Collaborative Target-Localization and Information-based Control in Networks of UAVs", sottoposto all'attenzione della commissione di una conferenza internazionale. È possibile leggere l'articolo integrale e ulteriori dettagli sulla conferenza in Appendice A.



(a) $\overline{PEB}_d^{(k)}$ in funzione del numero di nodi n .



(b) Success rate in funzione di n con $PEB_{\text{goal}} = 1.5\text{m}$.



(c) Success rate in funzione di n con $PEB_{\text{goal}} = 0.5\text{m}$.

Figura 5.5: Performance e convergenza dell'algoritmo distribuito al variare di n .

Capitolo 6

Conclusioni

IN questa tesi sono stati analizzati aspetti legati all'implementazione di algoritmi per la navigazione di UAV per la localizzazione di un target, in condizioni statiche e dinamiche. Dopo aver derivato le equazioni dei limiti fondamentali degli stimatori di posizione per diversi modelli di osservazione per un approccio centralizzato e distribuito considerando i vincoli di latenza, sono stati descritti gli strumenti matematici per la creazione di un algoritmo di ottimizzazione libera e vincolata, sia per un algoritmo centralizzato che per un algoritmo distribuito. Seguendo tali algoritmi, è possibile calcolare iterativamente le posizioni ottimali degli UAV in modo da massimizzare l'accuratezza nella stima della posizione del target. Attraverso questi algoritmi, sono stati ottenuti importanti risultati per la comprensione del comportamento degli UAV e sono state descritte le performance in funzione di diversi parametri, come ad esempio il raggio di comunicazione. In particolare, per l'algoritmo centralizzato si è descritto il comportamento degli UAV in condizioni di varianza di errore di stima della distanza costante dove il problema è unicamente geometrico per analizzare le posizioni che i nodi tendono ad occupare, ricavando nella totalità delle simulazioni una formazione semicircolare a causa di simmetrie presenti nella formulazione del Position Error Bound. Si sono analizzati anche i risultati ottenuti con algoritmo vincolato per il modello a varianza di errore di stima dipendente dalla distanza, dove i UAV tendono ad avvicinarsi al target per migliorare la qualità della misura. In questo scenario, sono state lanciate molte simulazioni al variare dello scenario, a partire dal numero di nodi, dalla presenza o assenza di ostacoli, con target statico o in movimento. Si è effettuato un confronto fra il calcolo del PEB con la reale posizione e utilizzando uno stimatore Maximum Likelihood in condizioni asintotiche per la stima delle coordinate del target, analizzando le differenze al variare dell'intensità del rumore di stima della distanza. È stato analizzato l'effetto che le condizioni di canale NLOS portano alla performance di posizionamento e si sono effettuati degli studi

di convergenza dell'algoritmo. Infine, sono state discusse le performance dell'algoritmo distribuito considerando vincoli di latenza e sono state lanciate delle simulazioni per l'analisi del comportamento al variare dei parametri legati unicamente all'algoritmo distribuito. In particolare, si è analizzata la performance al variare del communication range, numero di nodi e numero massimo di hops consentiti per rispettare i vincoli di latenza. Dai risultati è stato possibile derivare delle importanti linee guida sul dimensionamento della rete di UAV al variare delle condizioni operative. A partire da questo lavoro potrebbe essere interessante studiare le performance utilizzando degli stimatori pratici, abbandonando l'ipotesi di condizione di stimatore efficiente. Inoltre, un aspetto importante riguarda l'effettiva implementazione pratica su droni dell'algoritmo di navigazione e localizzazione per valutare le reali performance e per fornire uno strumento molto importante per garantire un servizio di localizzazione flessibile e dinamico. Infine, un interessante sviluppo è legato alla localizzazione relativa, dove gli UAV non sono a conoscenza della loro posizione: questa applicazione potrebbe rivelarsi interessante quando anche i droni si trovano in condizioni indoor. Un primo esempio di questo sviluppo può essere quello di fornire un supporto ad un vigile del fuoco all'interno di un edificio in fiamme dove sono gli UAV ad effettuare un'analisi dell'ambiente, così da recuperare informazioni e comunicare il percorso più sicuro da seguire all'operatore per salvare chi si trova in pericolo più rapidamente e con meno rischi.

Collaborative Target-Localization and Information-based Control in Networks of UAVs

NELLE pagine seguenti viene riportato l'articolo che è stato scritto parallelamente a questa tesi utilizzando lo stesso simulatore per il calcolo dei risultati, in collaborazione con la Dott. Anna Guerra (Università di Bologna), il Prof. Davide Dardari (Università di Bologna) e il Prof. Petar Djuric (Stony Brook University, New York). L'articolo si focalizza sullo studio del comportamento dell'algoritmo centralizzato in presenza di vincoli di latenza nella comunicazione delle informazioni fra UAV.

L'articolo è stato scritto per la diciannovesima conferenza per l'elaborazione dei segnali per comunicazione wireless *Signal Processing Advances in Wireless Communications* (SPAWC). La commissione di SPAWC 2018 [<http://spawc2018.org/>] è composta da personalità scientifiche da tutto il mondo e si occuperà della decisione se accettare o meno il lavoro. Il programma tecnico della conferenza si compone soprattutto delle emergenti tematiche di:

- machine learning e analisi dei dati;
- sicurezza e privacy del physical-layer;
- elaborazione dei segnali e biological communications;
- tecnologia 5G.

Collaborative Target-Localization and Information-based Control in Networks of UAVs

Anna Guerra, *Member, IEEE*, Nicola Sparnacci, Davide Dardari, *Senior Member, IEEE*
 Petar M. Djurić, *Fellow, IEEE*

Abstract—In this paper, we study the capacity of UAV networks for high-accuracy collaborative localization of target users. We address the problem of designing a distributed control scheme for navigation and formation based on an information-seeking criterion maximizing the localization accuracy. Each UAV is assumed to be able to communicate and collaborate with other UAVs that are within a neighboring region, allowing for a feasible distributed solution which takes into account a trade-off between localization accuracy and speed of convergence to a suitable localization of a target. Such an investigation also considers communication latency constraints as well as safety requirements such as inter-UAV and obstacle collision avoidance.

Index Terms—UAV networks, Information-Seeking Control, Target Localization, Cramér-Rao Lower Bound.

I. INTRODUCTION

In the future, unmanned aerial vehicles (UAVs) (e.g., drones) are expected to become an integral part of our daily life as they become smaller, cheaper, smarter and more versatile [1], [2]. Most of the actual and foreseen applications of UAVs require that they are networked and are able to execute specific tasks. This is typically accomplished through cooperating UAVs that navigate in a collision-free and energy efficient manner according to centralized or decentralized controls [3], [4]. UAV networks are well addressed for outdoor applications in surveillance, entertainment, precision agriculture, emergency or after-disaster events and also as base stations for flying ad-hoc networks (FANETs) [5], [6]. A potential large interest scenario related to emergency and rescue situations is depicted in Fig. 1 where the UAVs serve as a localization network able to guide users (e.g., firemen) inside a building. In such mixed indoor/outdoor scenarios, the UAV positions are considered known, e.g., retrieved from Global Positioning System (GPS), and the main goal is to track users that cannot access neither GPS nor ad-hoc localization infrastructure. The presence of walls requires dedicated wireless technologies capable of estimating the distance (ranging) between each UAV and the user (target) in harsh non line-of-sight (NLOS) conditions. An example is given by the Near-

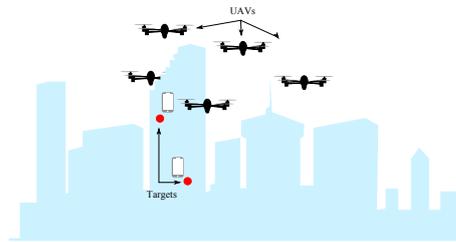


Fig. 1. Mixed indoor/outdoor target-localization using a UAV network.

Field Electromagnetic Ranging (NFER) technology working at extremely low frequencies [7].

In this context, the possibility of using UAV networks is increasingly attractive because of their flexibility and reconfigurability. Moreover, their capability to dispose in a well-structured formation and to maintain it during navigation is an essential feature. The formation goal can be that of maximizing an information measure (or minimizing an uncertainty measure) by seeking more informative measurements for better estimating the state of the system (information-seeking), which is the position of the target in our case [8]. In this sense, many research contributions focus on optimal sensor placement [9]–[11] and optimal control [12], [13], but they neglect the latency of the network as a figure of merit, which becomes critical especially when the control is decentralized. In fact, UAVs need to take decisions in few milliseconds. This means that in a decentralized setup, there is no time to diffuse updated information using multi-hop communications. Conversely, if multi-hops are used, this information becomes out-of-date thus compromising the performance of the navigation control necessary to achieve the desired localization accuracy.

In contrast to the common approach of a centralized control, the goal of this paper is to develop a decentralized control for UAVs able to assess a trade-off between the localization accuracy and the convergence speed. To this end, the network of UAVs acts as a distributed wireless sensor network (WSN) in which each UAV exchanges the collected information with its closest neighbors in order to take into account communication latency constraints. Then, this information is fused so that a single UAV becomes able to rapidly decide where to go next in order to minimize the localization error.

Anna Guerra, Nicola Sparnacci and Davide Dardari are with the Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi" - DEI, University of Bologna, 47521 Cesena, Italy. (e-mail: anna.guerra3@unibo.it, nicola.sparnacci@studio.unibo.it, davide.dardari@unibo.it).

Petar M. Djurić is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA. (e-mail: petar.djuric@stonybrook.edu).

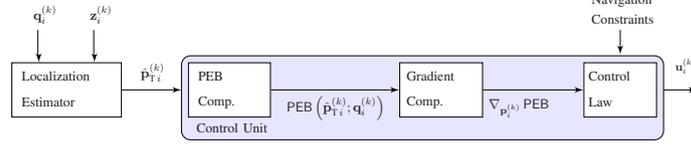


Fig. 2. Decentralized block-diagram for joint localization and navigation at the i th UAV.

II. UAVS NETWORK MODEL

We consider a network of N UAVs where each UAV acts as a (mobile) reference node, i.e., its location $\mathbf{p}_i^{(k)} = [x_i^{(k)}, y_i^{(k)}]^T$, $i = 1, 2, \dots, N$, is assumed known for all time slots $k = 1, \dots, K$, for instance, because it is derived from GPS. The purpose of the UAVs network is to infer the position of a target which is GPS-denied, namely $\mathbf{p}_T = [x_T, y_T]^T$.

To this end, in each time slot the UAVs communicate and exchange information about their positions and ranging measurements with their respective one-hop neighbors within a radius d_{hop} . We indicate with $\mathcal{R}_i^{(k)} = \{\mathbf{p}_j^{(k)} : \|\mathbf{p}_i^{(k)} - \mathbf{p}_j^{(k)}\|_2 \leq d_{\text{hop}}, j \neq i\}$ the set of neighbor positions of the i th UAV at time slot k . For all the UAVs that are located at a greater distance, measurements and UAV coordinates are collected through multi-hop communication with a corresponding delay equal to $h_{ij}^{(k)}$ time slots, where $h_{ij}^{(k)}$ is the number of hops between the i th and j th UAV at time slot k .¹ As a consequence, such measurements and position information are partially out-of-date.

Starting from the collected information, the aim is to estimate the best control signals for UAVs to minimize the target localization error. More specifically, given the transition model of the UAVs,

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \mathbf{u}_i^{(k)} + \mathbf{w}_i^{(k)}, \quad (1)$$

with $\mathbf{w}_i^{(k)}$ being the transition perturbation, the goal is to estimate the control signal $\mathbf{u}_i^{(k)} = [\Delta x_i^{(k)}, \Delta y_i^{(k)}]^T$ in order to minimize the target localization error, i.e., $\|\hat{\mathbf{p}}_{T_i}^{(k)} - \mathbf{p}_T\|_2$, where $\hat{\mathbf{p}}_{T_i}^{(k)} = [\hat{x}_{T_i}^{(k)}, \hat{y}_{T_i}^{(k)}]^T$ is the estimated target position by the i th UAV at time slot k . This is done in a decentralized manner by each UAV.

In the following section, we specify the adopted observation model and, based on it, we derive the Position Error Bound (PEB) of the target position that is used in the sequel as a cost function of the control optimization problem.

A. Observation Model

We denote with $\mathbf{z}_i^{(k)}$ the vector containing the measurements available at the i th UAV at time slot k ,

$$\mathbf{z}_i^{(k)} = [\dots, z_j^{(k-h+1)}, \dots]^T = \mathbf{h}_i^{(k)}(\mathbf{p}_T) + \mathbf{v}_i^{(k)}, \quad (2)$$

¹In order to simplify the notation, in the following we drop the UAV and time slot indexes, i.e., we use h instead of $h_{ij}^{(k)}$. Moreover, we set $h_{ii}^{(k)} = 1$.

where the generic element $z_j^{(k-h+1)}$ is the measurement of the j th UAV delayed by the number of hops between the i th and j th UAVs. With h_{max} we denote the maximum number of allowed hops, which impacts the amount of measurements available at each UAV. The function $\mathbf{h}_i^{(k)}(\cdot)$ relates the target position and the observations, and $\mathbf{v}_i^{(k)}$ is a vector of zero-mean Gaussian measurement noise with covariance matrix $\mathbf{R}_i^{(k)} = \text{diag}(\dots, (\sigma_j^{(k-h+1)})^2, \dots)$.

We assume that the measurements represent range estimates and, therefore, the generic element in (2) relative to the i th UAV at the k th time slot becomes

$$z_i^{(k)} = d_i^{(k)} + v_i^{(k)}, \quad (3)$$

where $d_i^{(k)} = \|\mathbf{p}_i^{(k)} - \mathbf{p}_T\|_2$ is the actual distance between the i th UAV and the target, and $v_i^{(k)} \sim \mathcal{N}(v_i^{(k)}; 0, (\sigma_i^{(k)})^2)$. We model the ranging error as distance-dependent, i.e., $(\sigma_i^{(k)})^2 = \sigma_0^2 (d_i^{(k)})^\alpha + \eta_i^{(k)} \sigma_b^2$, with σ_0^2 being the variance at a reference distance d_0 , and α the path-loss exponent [14]. Moreover, we consider an additional dispersion σ_b^2 , which is absent when the link is in line-of-sight (LOS) ($\eta_i^{(k)} = 0$) and present when in NLOS ($\eta_i^{(k)} = 1$). Note that the ranging error is an indicator of the quality of the corresponding measurement.

Given (3), the (distributed) log-likelihood function at the i th UAV in the k th time slot is given by

$$\Lambda(\mathbf{z}_i^{(k)} | \mathbf{p}_T) = \ln f(z_i^{(k)} | \mathbf{p}_T) + \sum_{j=1, j \neq i}^N \ln f(z_j^{(k-h+1)} | \mathbf{p}_T), \quad (4)$$

where $f(\mathbf{z}_i^{(k)} | \mathbf{p}_T) = \mathcal{N}(\mathbf{z}_i^{(k)}; \mathbf{h}_i^{(k)}(\mathbf{p}_T), \mathbf{R}_i^{(k)})$.

B. Position Error Bound

The PEB is a figure of merit related to the Cramér-Rao lower bound (CRLB) on the position and provides a measure of the theoretical localization accuracy. It is defined by [14]

$$\text{PEB}(\mathbf{p}_T; \mathbf{q}_i^{(k)}) = \sqrt{\text{tr}(\mathbf{J}^{-1}(\mathbf{p}_T; \mathbf{q}_i^{(k)}))}, \quad (5)$$

where $\text{tr}(\cdot)$ is the trace operator, $\mathbf{J}(\cdot)$ is the Fisher Information Matrix (FIM), and $\mathbf{q}_i^{(k)} = [\dots, \mathbf{p}_j^{(k-h+1)}, \dots]^T$ is a vector whose elements are the locations of all the UAVs as known by the i th UAVs at time slot k . In particular, the locations of

the other UAVs are delayed by the number of hops between them and the i th UAV. The PEB will be the cost function in the information-seeking control that follows. According to Fig. 2, each UAV computes the PEB considering the positions of the neighboring nodes and the quality of ranging measurements. Further mathematical details are given below.

The FIM is a measure of the information about the parameter of interest, e.g., the position, brought by the measurements. It is defined as [15]

$$\mathbf{J}(\mathbf{p}_T; \mathbf{q}_i^{(k)}) = \mathbb{E}_{\mathbf{z}_i^{(k)}} \left\{ \left[\nabla_{\mathbf{p}_T} \Lambda(\mathbf{z}_i^{(k)} | \mathbf{p}_T) \right] \left[\nabla_{\mathbf{p}_T} \Lambda(\mathbf{z}_i^{(k)} | \mathbf{p}_T) \right]^T \right\}, \quad (6)$$

where $\mathbb{E}_{\mathbf{z}_i^{(k)}} \{\cdot\}$ is the expectation over the measurements, $\nabla_{\mathbf{p}_T}(\cdot)$ is the gradient with respect to the target position, and $\Lambda(\mathbf{z}_i^{(k)} | \mathbf{p}_T)$ is the log-likelihood function defined as in (4).

Following [14], we obtain that the PEB can be written as

$$\text{PEB}(\mathbf{p}_T; \mathbf{q}_i^{(k)}) = \sqrt{\frac{A_i^{(k)} + \sum_{j \neq i} A_j^{(k-h+1)}}{S_{x,i}^{(k)} S_{y,i}^{(k)} - (S_{xy,i}^{(k)})^2}}, \quad (7)$$

where

$$\begin{aligned} S_{x,i}^{(k)} &= A_i^{(k)} (c_i^{(k)})^2 + \sum_{j \neq i} A_j^{(k-h+1)} (c_j^{(k-h+1)})^2, \\ S_{y,i}^{(k)} &= A_i^{(k)} (s_i^{(k)})^2 + \sum_{j \neq i} A_j^{(k-h+1)} (s_j^{(k-h+1)})^2, \\ S_{xy,i}^{(k)} &= A_i^{(k)} c_i^{(k)} s_i^{(k)} + \sum_{j \neq i} A_j^{(k-h+1)} c_j^{(k-h+1)} s_j^{(k-h+1)}, \end{aligned} \quad (8)$$

with $A_i^{(k)} = \frac{\sigma^2}{2(d_i^{(k)})^2} + \frac{1}{(\sigma_i^{(k)})^2}$ while $s_i^{(k)}$ and $c_i^{(k)}$ are given by

$$c_i^{(k)} = \frac{(x_i^{(k)} - x_T)}{d_i^{(k)}}, \quad s_i^{(k)} = \frac{(y_i^{(k)} - y_T)}{d_i^{(k)}}. \quad (9)$$

Note that the PEB depends on the UAVs and target positions through (9) and on the ranging error through $A_i^{(k)}$.

III. UAVS INFORMATION-SEEKING CONTROL

The goal of the formation control is to estimate the control signals at each UAV to minimize the target localization error. Ideally, given the target position \mathbf{p}_T , one can find the optimal geometric configuration of the UAVs that minimize the PEB. This formation is obtained by solving the following minimization problem:

$$\mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbb{R}^{2N}} \text{PEB}(\mathbf{p}_T; \mathbf{p}) \quad (10)$$

with $\mathbf{p} = [\mathbf{p}_1, \dots, \mathbf{p}_i, \dots, \mathbf{p}_N]^T$ being the vector containing the actual UAVs positions. The problem in (10) represents an unconstrained minimization problem and it can be solved in an iterative way using the *steepest descent gradient* method for each UAV according to [16]

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} - \gamma \nabla_{\mathbf{p}_i^{(k)}} \text{PEB}(\hat{\mathbf{p}}_T^{(k)}; \mathbf{q}_i^{(k)}), \quad (11)$$

where γ represents the spatial step, and $\nabla_{\mathbf{p}_i^{(k)}}(\cdot)$ is the gradient operator with respect to the UAVs positions. Note that, unlike in (10) where the target position is assumed exactly known, here the gradient is applied to the PEB of the target position estimate $\hat{\mathbf{p}}_T^{(k)}$ computed by the localization algorithm running at the i th UAV at time slot k .

Further, we introduce some constraints regarding the UAV speed and the inter-UAV, the UAV-target and obstacle collision avoidance. More specifically, we have

$$\begin{aligned} 1) & d_{ij}^{(k)} \geq d_a \quad i, j = 1, \dots, N, i \neq j, \\ 2) & d_i^{(k)} \geq d_b \quad \forall i, \\ 3) & \mathcal{T}_i \cap \mathcal{O} = \emptyset \quad \forall i, \\ 4) & \|\mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)}\|_2 = v \quad \forall i, \end{aligned} \quad (12)$$

where $d_{ij}^{(k)} = \|\mathbf{p}_i^{(k)} - \mathbf{p}_j^{(k)}\|_2$ is the inter-UAV distance, d_a is the safety distance each UAV should keep from the others to not collide, d_b is the safety distance with respect to the target, \mathcal{T}_i is the set of spatial points composed of allowed positions of the i th UAV, \mathcal{O} is the set of obstacles present in the environment, and v is the UAV speed. Obviously, it would be also important to consider a constraint on the minimum number of neighbors to ensure the connectivity of the network which would be partially incompatible with the first constraint whose effect is that of spreading the UAVs in space. Nevertheless, in this paper, we assume that the connectivity is always guaranteed and that, if by chance a UAV gets isolated, it is possible to rely on the latest known information. The *projection gradient* method can be used to solve the constrained minimization problem according to [16]. Thus, we obtain

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} - \gamma \mathbf{P} \nabla_{\mathbf{p}_i^{(k)}} \text{PEB}(\hat{\mathbf{p}}_T^{(k)}; \mathbf{q}_i^{(k)}) - \mathbf{N} (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{g}_a, \quad (13)$$

where $\mathbf{P} = \mathbf{I} - \mathbf{N} (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T$ is the projection matrix with \mathbf{I} being the identity matrix and $\mathbf{N} = \nabla_{\mathbf{p}_i^{(k)}}(\mathbf{g}_a)$ being the gradient of the V active constraints.² More specifically, the vector of active constraints is given by $\mathbf{g}_a = [\mathbf{g}_1 \ \mathbf{g}_2 \ \mathbf{g}_3]$, where

$$\mathbf{g}_1 = \mathbf{d}_{a,1} - d_a, \quad \mathbf{d}_{a,1} = \left\{ d_{ij}^{(k)} : d_{ij}^{(k)} < d_a \right\}, \quad (14)$$

$$\mathbf{g}_2 = \mathbf{d}_{a,2} - d_b, \quad \mathbf{d}_{a,2} = \left\{ d_i^{(k)} : d_i^{(k)} < d_b \right\}, \quad (15)$$

and where \mathbf{g}_3 contains the active obstacle avoidance constraints. The fourth constraint in (12) is used to set the UAV speed once the control signal has been estimated.

IV. RESULTS

The case study proposed here considers a 2D indoor/outdoor scenario with obstacles (e.g., buildings), represented in gray in Fig. 3, and with the target, indicated as a green triangle in the plot, located inside one of them. The blue and red dots indicate the initial and final UAV positions, respectively.

For testing the control design strategy, the UAV speed was set to $v = 1$ m/step and $\gamma = 1$ m for each time slot.

²The notation has been simplified as follows: $\mathbf{P}_i^{(k)} = \mathbf{P}$ and $\mathbf{N}_i^{(k)} = \mathbf{N}$.

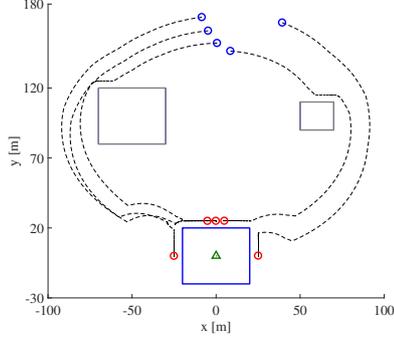


Fig. 3. Example of a scenario with $N = 5$ UAVs.

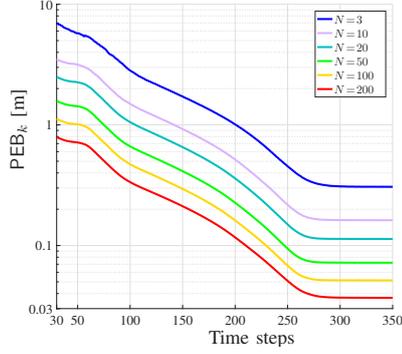


Fig. 4. Averaged PEB as a function of N with $d_{\text{hop}} = 20$ m and $h_{\text{max}} = 1$.

The standard deviation of the ranging error at the reference distance of 1 m was $\sigma_0 = 0.01$ m whereas the path loss exponent was $\alpha = 2$. The additional standard deviation due to NLOS channel conditions was set to $\sigma_b = 2$ m. Further, we considered the following safety distances: the inter-UAV was $d_a = 0.5$ m, the distance from the target was $d_b = 10$ m, and the distance from the obstacles was 5 m. We ran $N_{\text{MC}} = 100$ Monte Carlo iterations, where for each iteration we generated a different target position. In a real set up, the target position estimate, as shown in Fig. 2, is provided by a localization estimator. In order to get results that are independent from any specific localization estimator, and hence that can be used as a benchmark, we drew the position estimate $\hat{\mathbf{p}}_{T_i}^{(k)}$ from a zero mean 2D Gaussian distribution with a covariance matrix given by the inverse of the FIM.³

In Fig. 3, we present a Monte Carlo realization that illus-

³The impact of specific localization algorithms can be evaluated through an ad-hoc study.

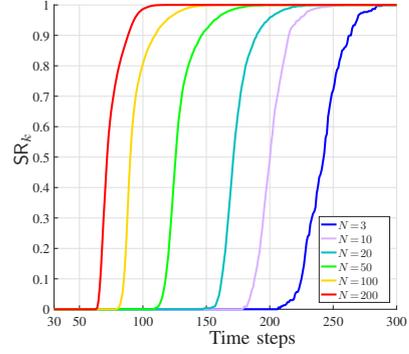


Fig. 5. SR_k vs. N with $d_{\text{hop}} = 20$ m, $h_{\text{max}} = 1$ and $\text{PEB}^* = 1.5$ m.

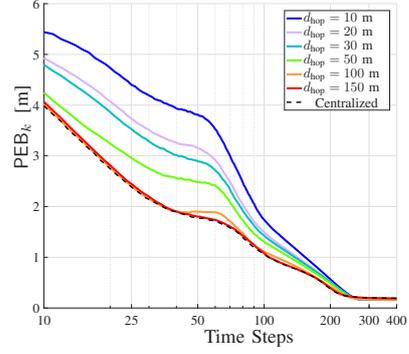


Fig. 6. Averaged PEB vs. d_{hop} with $N = 10$ and $h_{\text{max}} = 1$.

trates qualitatively how the UAVs are able to avoid collisions between them and with obstacles.

For getting more insight about the proposed approach, we evaluated its performance by computing the average PEB and the success rate SR_k defined as

$$\text{SR}^{(k)} = \frac{1}{N N_{\text{MC}}} \sum_{m=1}^{N_{\text{MC}}} \sum_{i=1}^N \mathbf{1} \left(\text{PEB} \left(\hat{\mathbf{p}}_{T_{im}}^{(k)}, \mathbf{p}_m^{(k)} \right) \leq \text{PEB}^* \right), \quad (16)$$

where $\mathbf{1}(x)$ is the indicator function; $\mathbf{1}(x) = 1$ if x is true and 0 otherwise, and PEB^* is the PEB required by the application. For each time slot k , this metric indicates the number of times that the PEB attains the mission-desired PEB. The index m indicates that the variables correspond to the m th Monte Carlo iteration. The results are obtained as a function of the number of UAVs, the maximum communication range d_{hop} and h_{max} .

Figs. 4-5 show the results obtained for $h_{\text{max}} = 1$ and $d_{\text{hop}} = 20$ m as a function of N . In Fig. 4, we plotted the PEB, averaged over Monte Carlo realizations, for different

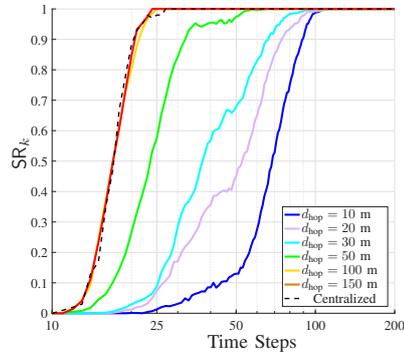


Fig. 7. Success rate vs. d_{hop} with $N = 10$, $h_{\text{max}} = 1$ and $\text{PEB}^* = 3$ m.

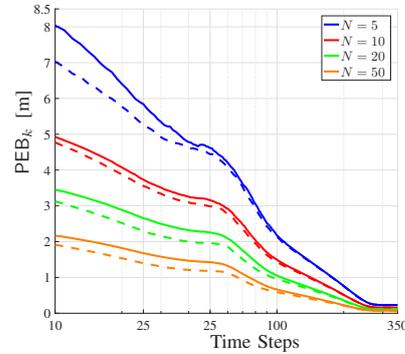


Fig. 8. Averaged PEB vs. h_{max} with $d_{\text{hop}} = 20$. Dashed lines refer to the case $h_{\text{max}} = 3$ while continuous lines to $h_{\text{max}} = 1$.

numbers of UAVs as a function of time steps. Notably, an increased number of UAVs translates in a better localization accuracy. Moreover, from Fig. 5 it emerges that the use of a bigger swarm is beneficial also for the convergence speed as the number of discrete time steps needed to attain the specific localization accuracy drastically decreases.

Figs. 6-7 show the results obtained for $N = 10$ and $h_{\text{max}} = 1$ as a function of the maximum communication range d_{hop} in terms of averaged PEB and success rate, respectively. In particular, in Fig. 6 it can be noticed that in general an increased d_{hop} allows for the collection of more up-to-date measurements and hence improved performance, especially when the UAVs are far from each other, reaching the performance one would obtain with a centralized control scheme. By contrast, at the end of their trajectories, the UAVs were more grouped and the impact of d_{hop} was less evident. This suggests that the communication range can be reduced without loss of performance. The effect of larger communication ranges on the speed of convergence is evident from Fig. 7, especially when the desired accuracy is not so stringent (e.g., $\text{PEB}^* = 3$ m). Finally, Fig. 8 displays the averaged PEB as a function of h_{max} . As can be seen, passing from $h_{\text{max}} = 1$ to $h_{\text{max}} = 3$, i.e., collecting more not updated measurements from UAVs that are further away, does not significantly improve the performance and, hence, one might argue that the use of multi-hop communication can be kept limited.

V. CONCLUSIONS

In this paper, we investigated a distributed control law for a network of UAVs realizing a flexible and dynamic localization and tracking infrastructure. The observation model was designed to include realistic propagation effects related to mixed indoor/outdoor scenarios with the possibility of NLOS conditions. Our results showed a trade-off between the positioning accuracy and the convergence speed as a function of the number of UAVs in the network, the maximum communication range, and the impact of multi-hop communication.

REFERENCES

- [1] J. Shamma, *Cooperative Control of Distributed Multi-agent Systems*. John Wiley & Sons, 2008.
- [2] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.
- [3] M. Dorigo *et al.*, "Swarmoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 60–71, 2013.
- [4] G. Beni, "From swarm intelligence to swarm robotics," in *Int. Workshop Swarm Robot.* Springer, 2004, pp. 1–9.
- [5] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," in *Proc. 2017 IEEE Int. Conf. Commun. (ICC)*. IEEE, 2017, pp. 1–6.
- [6] —, "Local map-assisted positioning for flying wireless relays," *arXiv preprint arXiv:1801.03595*, 2018.
- [7] D. Dardari and N. Decarli, "Method and apparatus for estimating a distance and a location through near-field multi-frequency radio transmissions," European Patent EP2 815 249B1, Feb. 12, 2012.
- [8] F. Meyer *et al.*, "Distributed estimation with information-seeking control in agent networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2439–2456, 2015.
- [9] A. N. Bishop *et al.*, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.
- [10] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [11] D. Moreno-Salinas, A. M. Pascoal, and J. Aranda, "Optimal sensor placement for underwater positioning with uncertainty in the target location," in *Proc. 2011 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2011, pp. 2308–2314.
- [12] B. M. Miller and E. Y. Rubinovich, *Impulsive Control in Continuous and Discrete-Continuous Systems*. Springer Science & Business Media, 2012.
- [13] A. J. Sinclair, R. J. Prazenica, and D. E. Jeffcoat, "Optimal and feedback path planning for cooperative attack," *J. Guidance, Control, and Dynamics*, vol. 31, no. 6, p. 1708, 2008.
- [14] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position error bound for UWB localization in dense cluttered environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, 2008.
- [15] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, 2004.
- [16] D. G. Luenberger, Y. Ye *et al.*, *Linear and Nonlinear Programming*. Springer, 1984, vol. 2.

Bibliografia

- [1] R. Olfati-Saber. «Flocking for multi-agent dynamic systems: algorithms and theory». In: *IEEE Transactions on Automatic Control* 51.3 (mar. 2006), pp. 401–420. ISSN: 0018-9286. DOI: 10.1109/TAC.2005.864190.
- [2] D Kaplan Elliott e J Hegarty Christopher. «Understanding GPS: principles and applications». In: *Edition Kaplan* (1996).
- [3] Guochang Xu e Yan Xu. *GPS: theory, algorithms and applications*. Springer, 2016.
- [4] Bradford W Parkinson, James J Spilker Jr et al. *Global Positioning System: Theory and Applications, vol. 1. Washington, DC: American Institute of Aeronautics and Astronautics*. 1996.
- [5] C. C. Counselman. «Multipath-rejecting GPS antennas». In: *Proceedings of the IEEE* 87.1 (gen. 1999), pp. 86–91. ISSN: 0018-9219. DOI: 10.1109/5.736343.
- [6] S. Gezici et al. «Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks». In: *IEEE Signal Processing Magazine* 22.4 (lug. 2005), pp. 70–84. ISSN: 1053-5888. DOI: 10.1109/MSP.2005.1458289.
- [7] M. Z. Win et al. «History and Applications of UWB [Scanning the Issue]». In: *Proceedings of the IEEE* 97.2 (feb. 2009), pp. 198–204. ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.2008762.
- [8] D. Dardari et al. «Ranging With Ultrawide Bandwidth Signals in Multipath Environments». In: *Proceedings of the IEEE* 97.2 (feb. 2009), pp. 404–426. ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.2008846.
- [9] D. Cassioli, M. Z. Win e A. F. Molisch. «The ultra-wide bandwidth indoor channel: from statistical model to simulations». In: *IEEE Journal on Selected Areas in Communications* 20.6 (ago. 2002), pp. 1247–1257. ISSN: 0733-8716. DOI: 10.1109/JSAC.2002.801228.

- [10] M. Z. Win e R. A. Scholtz. «Impulse radio: how it works». In: *IEEE Communications Letters* 2.2 (feb. 1998), pp. 36–38. ISSN: 1089-7798. DOI: 10.1109/4234.660796.
- [11] R. S. Kshetrimayum. «An introduction to UWB communication systems». In: *IEEE Potentials* 28.2 (mar. 2009), pp. 9–13. ISSN: 0278-6648. DOI: 10.1109/MPOT.2009.931847.
- [12] S. Gezici e H. V. Poor. «Position Estimation via Ultra-Wide-Band Signals». In: *Proceedings of the IEEE* 97.2 (feb. 2009), pp. 386–403. ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.2008840.
- [13] S. Han et al. «Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G». In: *IEEE Communications Magazine* 53.1 (gen. 2015), pp. 186–194. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7010533.
- [14] T. S. Rappaport et al. «Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!» In: *IEEE Access* 1 (2013), pp. 335–349. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2013.2260813.
- [15] F. Lemic et al. «Localization as a feature of mmWave communication». In: *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. Set. 2016, pp. 1033–1038. DOI: 10.1109/IWCMC.2016.7577201.
- [16] H. Wymeersch et al. «5G mmWave Positioning for Vehicular Networks». In: *IEEE Wireless Communications* 24.6 (dic. 2017), pp. 80–86. ISSN: 1536-1284. DOI: 10.1109/MWC.2017.1600374.
- [17] Heinrich Hertz e William Thomson Baron Kelvin. *Electric waves*. Macmillan London, 1893.
- [18] C. W. Kim, F. P. S. Chin e H. K. Garg. «Selection of Frequency for Near Field Electromagnetic Ranging (NFER) Based on its Cramer-Rao Bound». In: *IEEE Signal Processing Letters* 14.12 (dic. 2007), pp. 1000–1003. ISSN: 1070-9908. DOI: 10.1109/LSP.2007.903274.
- [19] H. G. Schantz. «Near field phase behavior». In: *2005 IEEE Antennas and Propagation Society International Symposium*. Vol. 3B. Lug. 2005, 134–137 vol. 3B. DOI: 10.1109/APS.2005.1552452.
- [20] Glenn S Smith. *An introduction to classical electromagnetic radiation*. Cambridge University Press, 1997.
- [21] D. Dardari e N. Decarli. *Method and apparatus for estimating a distance and a location through near-field multi-frequency radio transmissions*. European Patent Grant EP2815249B1. Feb. 2012.
- [22] Davide Dardari, Emanuela Falletti e Marco Luise. *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2012.

- [23] Merrill I Skolnik. «Introduction to radar». In: *Radar Handbook 2* (1962).
- [24] Yihong Qi. «Wireless geolocation in a non-line-of-sight environment». Tesi di dott. Princeton University Princeton, 2003.
- [25] A. Mallat, J. Louveaux e L. Vandendorpe. «UWB Based Positioning in Multipath Channels: CRBs for AOA and for Hybrid TOA-AOA Based Methods». In: *2007 IEEE International Conference on Communications*. Giu. 2007, pp. 5775–5780. DOI: 10.1109/ICC.2007.957.
- [26] D. Dardari, P. Closas e P. M. Djurić. «Indoor Tracking: Theory, Methods, and Technologies». In: *IEEE Transactions on Vehicular Technology* 64.4 (apr. 2015), pp. 1263–1278. ISSN: 0018-9545. DOI: 10.1109/TVT.2015.2403868.
- [27] «On Optimal Sensor Placement and Motion Coordination for Target Tracking». In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Apr. 2005, pp. 4544–4549. DOI: 10.1109/ROBOT.2005.1570820.
- [28] J. Liu et al. «Distributed state representation for tracking problems in sensor networks». In: *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*. Apr. 2004, pp. 234–242. DOI: 10.1109/IPSIN.2004.1307343.
- [29] Jeremy Packer e Josh Reeves. «Romancing the drone: Military desire and anthropophobia from SAGE to Swarm». In: *Canadian Journal of Communication* 38.3 (2013).
- [30] John F Keane e Stephen S Carr. «A brief history of early unmanned aircraft». In: *Johns Hopkins APL Technical Digest* 32.3 (2013), pp. 558–571.
- [31] Y. Zeng, R. Zhang e T. J. Lim. «Wireless communications with unmanned aerial vehicles: opportunities and challenges». In: *IEEE Communications Magazine* 54.5 (mag. 2016), pp. 36–42. ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.7470933.
- [32] ENAC. *Mezzi Aerei a Pilotaggio Remoto*. URL: www.enac.gov.it (visitato il 12/02/2018).
- [33] M. Mirahsan, R. Schoenen e H. Yanikomeroğlu. «HetHetNets: Heterogeneous Traffic Distribution in Heterogeneous Wireless Cellular Networks». In: *IEEE Journal on Selected Areas in Communications* 33.10 (ott. 2015), pp. 2252–2265. ISSN: 0733-8716. DOI: 10.1109/JSAC.2015.2435391.

- [34] M. Alzenad et al. «3-D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage». In: *IEEE Wireless Communications Letters* 6.4 (ago. 2017), pp. 434–437. ISSN: 2162-2337. DOI: 10.1109/LWC.2017.2700840.
- [35] J. Chen e D. Gesbert. «Optimal positioning of flying relays for wireless networks: A LOS map approach». In: *2017 IEEE International Conference on Communications (ICC)*. Mag. 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996921.
- [36] Dariusz Ucinski. *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.
- [37] N. Levanon. «Lowest GDOP in 2-D scenarios». In: *IEE Proceedings - Radar, Sonar and Navigation* 147.3 (giu. 2000), pp. 149–155. ISSN: 1350-2395. DOI: 10.1049/ip-rsn:20000322.
- [38] Sulema Aranda, Sonia Martinez e Francesco Bullo. *On optimal sensor placement and motion coordination for target tracking*. Rapp. tecn. California University of Santa Barbara Dept. of Mechanical e Environmental Engineering, 2004.
- [39] Sonia Martinez e Francesco Bullo. «Optimal sensor placement and motion coordination for target tracking». In: *Automatica* 42.4 (2006), pp. 661–668.
- [40] A. N. Bishop et al. «Optimality Analysis of Sensor-Target Geometries in Passive Localization: Part 1 - Bearing-Only Localization». In: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. Dic. 2007, pp. 7–12. DOI: 10.1109/ISSNIP.2007.4496811.
- [41] A. N. Bishop et al. «Optimality Analysis of Sensor-Target Geometries in Passive Localization: Part 2 - Time-of-Arrival Based Localization». In: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. Dic. 2007, pp. 13–18. DOI: 10.1109/ISSNIP.2007.4496812.
- [42] A. N. Bishop e P. Jensfelt. «An optimality analysis of sensor-target geometries for signal strength based localization». In: *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. Dic. 2009, pp. 127–132. DOI: 10.1109/ISSNIP.2009.5416784.
- [43] D. Moreno-Salinas, A. M. Pascoal e J. Aranda. «Optimal sensor placement for underwater positioning with uncertainty in the target location». In: *2011 IEEE International Conference on Robotics and Automation*. Mag. 2011, pp. 2308–2314. DOI: 10.1109/ICRA.2011.5980152.

- [44] S. Zhang et al. «Location-Aware Formation Control in Swarm Navigation». In: *2015 IEEE Globecom Workshops (GC Wkshps)*. Dic. 2015, pp. 1–6. DOI: 10.1109/GLOCOMW.2015.7414165.
- [45] R. Olfati-Saber. «Flocking for multi-agent dynamic systems: algorithms and theory». In: *IEEE Transactions on Automatic Control* 51.3 (mar. 2006), pp. 401–420. ISSN: 0018-9286. DOI: 10.1109/TAC.2005.864190.
- [46] D. B. Jourdan, D. Dardari e M. Z. Win. «Position error bound for UWB localization in dense cluttered environments». In: *IEEE Transactions on Aerospace and Electronic Systems* 44.2 (apr. 2008), pp. 613–628. ISSN: 0018-9251. DOI: 10.1109/TAES.2008.4560210.
- [47] N. Decarli, F. Guidi e D. Dardari. «A Novel Joint RFID and Radar Sensor Network for Passive Localization: Design and Performance Bounds». In: *IEEE Journal of Selected Topics in Signal Processing* 8.1 (feb. 2014), pp. 80–95. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2013.2287174.
- [48] D. B. Jourdan et al. «Monte Carlo localization in dense multipath environments using UWB ranging». In: *2005 IEEE International Conference on Ultra-Wideband*. Set. 2005, pp. 314–319. DOI: 10.1109/ICU.2005.1570005.
- [49] N. Decarli et al. «LOS/NLOS detection for UWB signals: A comparative study using experimental data». In: *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*. Mag. 2010, pp. 169–173. DOI: 10.1109/ISWPC.2010.5483704.
- [50] Germund Dahlquist e Åke Björck. «Numerical methods in scientific computing, volume I». In: *Society for Industrial and Applied Mathematics* 8 (2008).
- [51] Germund Dahlquist e Åke Björck. «Numerical methods in scientific computing, volume II». In: *Society for Industrial and Applied Mathematics* 8 (2008).
- [52] Jo Bo Rosen. «The gradient projection method for nonlinear programming. Part I. Linear constraints». In: *Journal of the society for industrial and applied mathematics* 8.1 (1960), pp. 181–217.
- [53] JB Rosen. «The gradient projection method for nonlinear programming. Part II. Nonlinear constraints». In: *Journal of the Society for Industrial and Applied Mathematics* 9.4 (1961), pp. 514–532.
- [54] S. S. Ge e Y. J. Cui. «New potential functions for mobile robot path planning». In: *IEEE Transactions on Robotics and Automation* 16.5 (ott. 2000), pp. 615–620. ISSN: 1042-296X. DOI: 10.1109/70.880813.
- [55] Y. K. Hwang e N. Ahuja. «A potential field approach to path planning». In: *IEEE Transactions on Robotics and Automation* 8.1 (feb. 1992), pp. 23–32. ISSN: 1042-296X. DOI: 10.1109/70.127236.

- [56] J. Sun, J. Tang e S. Lao. «Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm». In: *IEEE Access* 5 (2017), pp. 18382–18390. DOI: 10.1109/ACCESS.2017.2746752.
- [57] O. Khatib. «Real-time obstacle avoidance for manipulators and mobile robots». In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. Mar. 1985, pp. 500–505. DOI: 10.1109/ROBOT.1985.1087247.
- [58] Jen-Hui Chuang. «Potential-based modeling of three-dimensional workspace for obstacle avoidance». In: *IEEE Transactions on Robotics and Automation* 14.5 (ott. 1998), pp. 778–785. ISSN: 1042-296X. DOI: 10.1109/70.720353.
- [59] W. Kowalczyk e K. Kozłowski. «Artificial potential based control for a large scale formation of mobile robots». In: *Proceedings of the Fourth International Workshop on Robot Motion and Control (IEEE Cat. No.04EX891)*. Giu. 2004, pp. 285–291. DOI: 10.1109/ROMOCO.2004.240742.
- [60] Y. K. Hwang e N. Ahuja. «A potential field approach to path planning». In: *IEEE Transactions on Robotics and Automation* 8.1 (feb. 1992), pp. 23–32. ISSN: 1042-296X. DOI: 10.1109/70.127236.
- [61] Ashraf Hossain e Rashmita Mishra. «Sensing and Link Model for Wireless Sensor Network: Coverage and Connectivity Analysis». In: *arXiv preprint arXiv:1406.1275* (2014).
- [62] Bang Wang. «Sensor coverage model». In: *Coverage Control in Sensor Networks*. Springer, 2010, pp. 19–34.
- [63] Norman Biggs, E Keith Lloyd e Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1976.
- [64] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.
- [65] Fan RK Chung. *Spectral graph theory*. 92. American Mathematical Soc., 1997.
- [66] Bojan Mohar et al. «The Laplacian spectrum of graphs». In: *Graph theory, combinatorics, and applications* 2.871-898 (1991), p. 12.
- [67] Anne Marsden. «Eigenvalues of the Laplacian and Their Relationship to the Connectedness of a Graph». In: *University of Chicago, REU* (2013).