



Università degli studi di Padova
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Triennale in Ingegneria dell'Informazione

Simulatore di tracciamento di persone
tramite una rete di radar a onde millimetriche

Relatore
Prof . Michele Rossi

Correlatore
Dott. Marco Canil

22 settembre 2022
anno accademico 2021/2022

Laureando
Davide Costa
matr. 1216490

Indice generale

Abstract.....	4
1 Introduzione.....	5
2 Radar a onde millimetriche.....	7
2.1 Fondamenti di funzionamento.....	7
2.2 Proprietà delle onde millimetriche.....	8
2.3 Tipi di segnali.....	9
3 Tracciamento di persone con radar singolo.....	13
3.1 Rilevamento.....	13
3.2 Associazione.....	15
3.3 Tracciamento.....	16
3.4 Problema dell'occlusione.....	16
4 Tracciamento di persone con radar multipli.....	17
4.1 Requisiti.....	17
4.2 Fusione dei dati.....	18
4.3 Scenario.....	19
5 Elezione del cluster head.....	21
5.1 Distributed Mobility-Adaptive Clustering.....	21
5.2 Progettazione del peso per DMAC.....	24
6 Simulatore.....	27
6.1 Struttura e implementazione.....	27
6.2 Utilizzo.....	29
7 Risultati.....	31
7.1 Funzionamento di DMAC.....	31
7.2 Tracciamento di persone.....	36
7.3 Conclusioni.....	38
Bibliografia.....	39

Abstract

Negli ultimi anni, il progresso della tecnologia radar ha permesso di impiegare questo tipo di sensori in nuovi ambiti, diversi da quelli in cui erano soliti essere utilizzati. In particolare, l'uso di frequenze a lunghezza d'onda millimetrica ha reso possibile la rilevazione di obiettivi di piccole dimensioni, aprendo la strada a nuove applicazioni quali, ad esempio, il tracciamento del movimento delle persone. Tuttavia, i radar a onde millimetriche commercialmente disponibili hanno portata limitata (circa 6-8 m) e sono soggetti al problema dell'occlusione, causato da ostacoli che limitano il campo visivo dei sensori. Il monitoraggio di un ambiente esteso o affollato richiede, quindi, di dispiegare molteplici sensori i cui dati possano essere comunicati e fusi tra loro. Per fare ciò, è necessario strutturare e organizzare gerarchicamente la rete di radar. In questo lavoro di tesi si è sviluppato un simulatore di reti di radar per il tracciamento delle persone. Il simulatore è stato usato per implementare e valutare una versione adattata di un algoritmo per l'organizzazione gerarchica della rete, chiamato Distributed Mobility-Adaptive Clustering.

1 Introduzione

I radar sono dispositivi inventati durante la prima metà del ventesimo secolo e usati per una vasta gamma di applicazioni. Storicamente, sono stati impiegati in settori come l'aviazione civile, la meteorologia e l'astronomia, e comunque, in generale, per la rilevazione di oggetti di grandi dimensioni come navi e aerei.

I recenti progressi tecnologici hanno reso possibile la progettazione di sensori radar in grado di operare con frequenze molto maggiori rispetto a quelle impiegate nei radar classici, le cosiddette *onde millimetriche*. Queste onde sono riflesse anche da oggetti notevolmente più piccoli rispetto agli aeromobili, e possono essere impiegate, ad esempio, per il tracciamento dei movimenti delle persone.

Tuttavia, l'utilizzo di un singolo sensore radar a onde millimetriche per il monitoraggio di ambienti è soggetto a limitazioni dovute al suo ristretto raggio d'azione, di circa 6-8 m nei dispositivi commerciali, e al problema dell'occlusione, ossia della presenza di ostacoli tra l'oggetto obiettivo del rilevamento e il radar stesso.

Per questi motivi, spesso è necessario utilizzare più radar contemporaneamente, al fine di fonderne le misure e migliorare le capacità di monitoraggio.

In questo lavoro, dopo aver esposto il funzionamento dei sistemi di tracciamento radar, è presentato un simulatore che permette di testare reti di sensori senza doverle implementare fisicamente. In questo modo, è possibile sviluppare algoritmi per reti di radar in maniera molto veloce ed efficiente, senza ricorrere all'uso di dispositivi fisici.

A tal proposito, è stato adattato e implementato un algoritmo di clusterizzazione chiamato *Distributed Mobility-Adaptive Clustering*, originariamente pensato per reti mobili ad hoc, e qui utilizzato per creare un'organizzazione gerarchica di una rete di radar, utile per il successivo impiego di algoritmi per la fusione dei dati.

2 Radar a onde millimetriche

Il termine *onde millimetriche* indica lo spettro di onde radio comprese tra 30 e 300 GHz, ovvero tra 1 e 10 millimetri di lunghezza d'onda.

A differenza dei radar tradizionali, che si usano per rilevare oggetti di grandi dimensioni come navi e aerei, un sistema radar che utilizza queste lunghezze d'onda è in grado di tracciare il movimento di bersagli con dimensioni ridotte [1].

Le onde millimetriche sono infatti onde molto corte e di conseguenza permettono di impiegare la tecnologia radar in campi che prima risultavano inaccessibili, tra cui la rilevazione di persone.

Un sensore ad onde millimetriche è costituito da elementi di trasmissione e di ricezione a radiofrequenza, componenti analogici, come l'oscillatore di clock e componenti digitali, tra cui convertitori analogico/digitali, microcontroller e processori di segnali digitali.

Negli ultimi anni, gli sviluppi tecnologici relativi alla progettazione dei circuiti e delle antenne hanno reso possibile l'abbassamento del costo di produzione di questo tipo di radar, rendendoli più facilmente reperibili e favorendone la diffusione.

2.1 Fondamenti di funzionamento

Il principio di funzionamento è concettualmente semplice: il radar trasmette un impulso elettromagnetico di breve durata, questo impulso si propaga nello spazio circostante e, quando incontra un ostacolo, come, ad esempio, una persona, vi rimbalza contro e torna indietro come un eco.

Il radar acquisisce l'onda riflessa e, dalla sua analisi, riesce a estrarre informazioni riguardanti l'ambiente monitorato. Ad esempio, misurando il tempo intercorso tra la trasmissione e il ritorno e conoscendo la velocità di riferimento, è possibile calcolare la distanza dell'oggetto colpito dall'impulso elettromagnetico.

La distanza R dell'oggetto può infatti essere calcolata utilizzando la relazione $R = c\tau / 2$, dove τ è il ritardo di propagazione in secondi e c è la velocità della luce in metri al secondo ($c \approx 3 \times 10^8$ m/s).

Il tipo di informazione che si riesce a ottenere dall'analisi delle riflessioni dipende dal tipo di forma d'onda che è inviato. Per esempio un'onda continua a modulazione d'impulso, che consiste in corti impulsi periodici spazati da intervalli di silenzio radar, permette di ricevere il segnale riflesso ed effettuare la stima mentre non viene trasmesso nulla (Figura 2.1.1) .

Al contrario, onde continue prive di modulazione non sarebbero utilizzabili per stimare la distanza del soggetto.

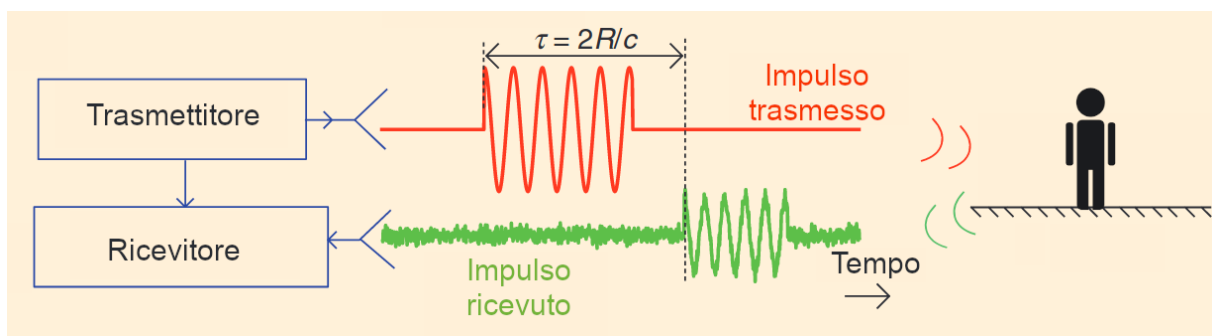


Figura 2.1.1: Andamento temporale di un impulso radar

Un'ulteriore possibilità offerta dai sistemi radar è la stima della velocità, basata sull'effetto Doppler. Supponendo che l'individuo si muova con velocità v , l'esistenza di un moto relativo tra la persona e il radar genera un ritardo nell'onda riflessa pari a $\tau = 2 (R + vt) / c$. La dipendenza dal tempo causa quindi un cambiamento di frequenza nell'onda ricevuta pari a $f_d = 2v / \lambda$, da cui è possibile ricavare v .

2.2 Proprietà delle onde millimetriche

I radar a onde millimetriche presentano diverse proprietà che li rendono interessanti rispetto ad altri sistemi di rilevamento:

- **Robustezza.** Il radar a onde millimetriche è robusto a condizioni atmosferiche come polvere, fumo, nebbia o pioggia; il sensore non è quindi influenzato dalla loro presenza in quanto tali mezzi sono trasparenti alle onde millimetriche.
- **Insensibilità al livello di illuminazione.** Il radar a onde millimetriche può funzionare indifferentemente con luci intense, luci abbaglianti o senza luce.

- **Elevata risoluzione.** Il radar a onde millimetriche offre elevata precisione nella misurazione della distanza, grazie all'impiego di lunghezze d'onda corte e di tecniche di compressione dell'impulso.
- **Miniaturizzazione dell'antenna.** Il radar a onde millimetriche permette di miniaturizzare le antenne in modo maggiore rispetto ad altri radar tradizionali che utilizzano onde centimetriche, raggiungendo anche pochi millimetri quadrati.
- **Ampia larghezza di banda.** L'ampia larghezza di banda della tecnologia radar a onde millimetriche consente di rilevare caratteristiche strutturali dettagliate di un bersaglio utilizzando segnali a modulazione di frequenza.
- **Tutela della privacy.** A differenza di un monitoraggio basato sulle telecamere, il radar a onde millimetriche preserva la privacy dei soggetti in quanto non cattura nessuna loro immagine.

2.3 Tipi di segnali

I sensori considerati per questo lavoro di tesi utilizzano un particolare tipo di segnale elettromagnetico, detto a *onda continua modulata in frequenza (frequency-modulated continuous wave, FMCW)*. I radar FMCW trasmettono segnali chiamati *chirp*, allo scopo di misurare distanza e velocità. Inoltre, grazie all'impiego di antenne multiple in trasmissione e in ricezione è possibile misurare anche l'angolo di arrivo delle riflessioni.

Esistono diversi tipi di segnali chirp:

- Lineare

In un chirp lineare la frequenza istantanea $f(t)$ varia linearmente nel tempo.

$$f(t) = ct + f_0 \qquad c = \frac{f_1 - f_0}{T}$$

Dove f_0 rappresenta la frequenza iniziale, f_1 la frequenza finale, c è il chirp rate. T è invece il tempo necessario per passare da f_0 a f_1 .

Integrando la funzione della frequenza si ottiene la funzione della fase:

$$\begin{aligned}\phi(t) &= \phi_0 + 2\pi \int_0^t f(\tau) d\tau \\ &= \phi_0 + 2\pi \left(\frac{c}{2} t^2 + f_0 t \right)\end{aligned}$$

Dove ϕ_0 rappresenta la fase iniziale. Se si considera un segnale chirp lineare e sinusoidale, la corrispondente funzione nel dominio del tempo è:

$$x(t) = \sin \left[\phi_0 + 2\pi \left(\frac{c}{2} t^2 + f_0 t \right) \right]$$

- Esponenziale

In un chirp esponenziale la frequenza del segnale varia nel tempo con una relazione geometrica.

$$f(t) = f_0 k^t \qquad k = \left(\frac{f_1}{f_0} \right)^{\frac{1}{T}}$$

Dove f_0 è la frequenza iniziale, f_1 la frequenza finale e k il tasso di crescita esponenziale della frequenza stessa. La fase varia nel tempo come:

$$\begin{aligned}\phi(t) &= \phi_0 + 2\pi \int_0^t f(\tau) d\tau \\ &= \phi_0 + 2\pi f_0 \left(\frac{k^t - 1}{\ln(k)} \right)\end{aligned}$$

Dove ϕ_0 rappresenta la fase iniziale. Considerando un segnale chirp esponenziale a forma d'onda sinusoidale, questo è descritto da:

$$x(t) = \sin \left[\phi_0 + 2\pi f_0 \left(\frac{k^t - 1}{\ln(k)} \right) \right]$$

- Iperbolico

In questo caso la frequenza varia in maniera iperbolica allo scorrere del tempo:

$$f(t) = \frac{f_0 f_1 T}{(f_0 - f_1)t + f_1 T}$$

E la funzione della fase risulta essere:

$$\begin{aligned} \phi(t) &= \phi_0 + 2\pi \int_0^t f(\tau) d\tau \\ &= \phi_0 + 2\pi \frac{-f_0 f_1 T}{f_1 - f_0} \ln\left(1 - \frac{f_1 - f_0}{f_1 T} t\right) \end{aligned}$$

Dove ϕ_0 rappresenta la fase iniziale. Un chirp sinusoidale iperbolico varia nel tempo secondo:

$$x(t) = \sin\left[\phi_0 + 2\pi \frac{-f_0 f_1 T}{f_1 - f_0} \ln\left(1 - \frac{f_1 - f_0}{f_1 T} t\right)\right]$$

3 Tracciamento di persone con radar singolo

La presenza di una persona nell'ambiente genera un gran numero di riflessioni nello spettro delle onde millimetriche, che sono rilevate dal radar sotto forma di punti. Questo insieme di punti, chiamato radar point cloud, può essere rappresentato in uno spazio cartesiano tridimensionale dove ogni punto è descritto da un vettore $[x, y, z]$ ottenuto a partire dalla distanza, dall'azimut e dall'angolo di elevazione rilevati dal sensore.

A partire dalle nuvole di punti osservate dal radar è possibile quindi tracciare il movimento dei soggetti all'interno di un ambiente e in questo capitolo saranno introdotti i concetti di base riguardanti il rilevamento basato sull'uso di onde millimetriche.

3.1 Rilevamento

Il rilevamento di persone a partire dalle point cloud richiede l'impiego di algoritmi di clustering che consentano di ricercare strutture e pattern all'interno di un dataset, con lo scopo di separare le riflessioni dei soggetti dal rumore.

È usato un algoritmo di clustering ogni qualvolta si vogliono trovare gruppi nei dati, e possiamo quindi includerlo tra le tecniche esplorative che ci consentono di estrarre informazioni da una point cloud senza avere informazioni a priori su di essa.

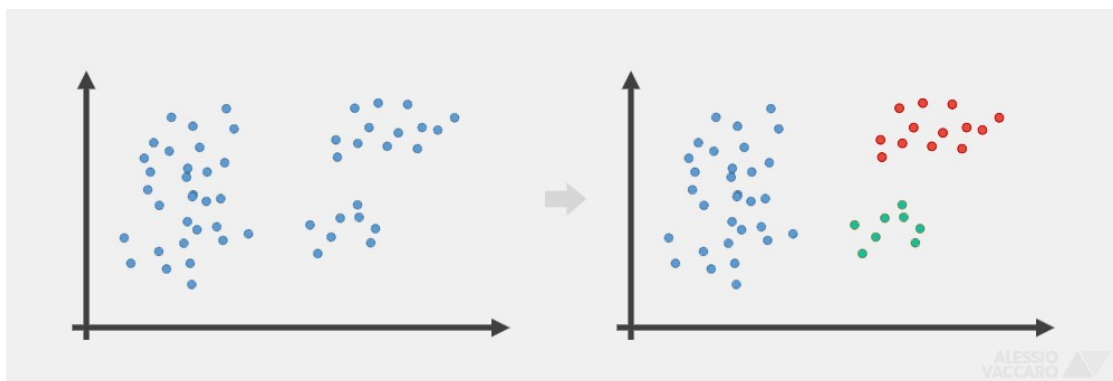


Figura 3.1.1: Individuazione dei cluster, ognuno dei quali è rappresentato da un colore diverso

L'algoritmo *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) rappresenta uno degli approcci più diffusi per il clustering di point cloud, che tiene conto delle differenze di densità tra le osservazioni [3] (Figura 3.1.1).

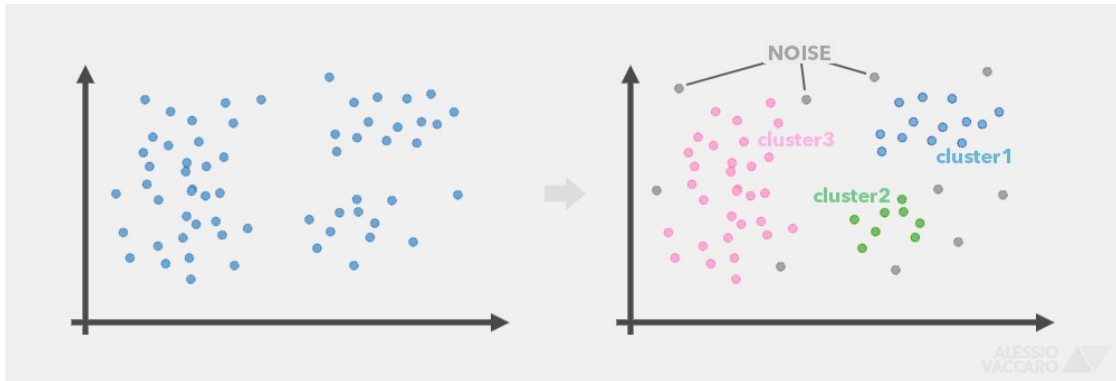


Figura 3.1.2: Identificazione dei cluster e del rumore

L'algoritmo di clusterizzazione DBSCAN permette anche di identificare i campioni che sono associabili al rumore. Tali osservazioni sono chiamate outliers, e sono punti che si discostano in maniera significativa dalle altre osservazioni.

Nell'immagine sopra (Figura 3.1.2) i punti grigi molto distanti dalle zone a maggiore densità vengono esclusi dai cluster e riconosciuti come outliers.

L'algoritmo necessita di due parametri:

- ϵ : distanza entro la quale ricercare punti vicini
- n : numero minimo di punti affinché si formi un cluster

DBSCAN si occupa pertanto di cercare tutti quei cluster i cui punti siano in un numero maggiore o uguale ad n e distanti tra loro meno di ϵ .

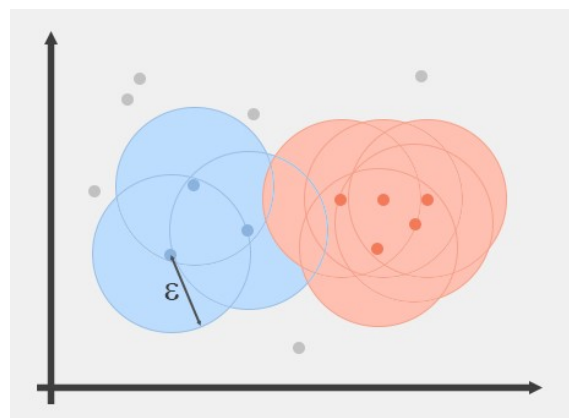


Figura 3.1.3:
Funzionamento di DBSCAN

Si consideri ad esempio *Figura 3.1.3*. In seguito all'invocazione di DBSCAN con $n=3$ vengono identificati due cluster e sei punti di rumore.

Osservando il grafico si può notare che due di questi sei punti, nonostante siano distanti tra loro meno di ε , vengono identificati come outliers perché non sono riusciti a raggiungere la popolazione minima per formare un cluster.

3.2 Associazione

Una componente fondamentale nel processo di tracciamento consiste nell'associare tra loro i cluster identificati da DBSCAN in diversi istanti temporali.

Lo scopo è fare in modo che i dati siano raggruppati in tracce dotate di identificativo univoco (*ID*), ognuna delle quali rappresenta il movimento di una persona differente allo scorrere del tempo.

In *Figura 3.2.1* le stelle rappresentano tutti i cluster identificati, che in fase di associazione sono aggregati nelle due tracce presenti.

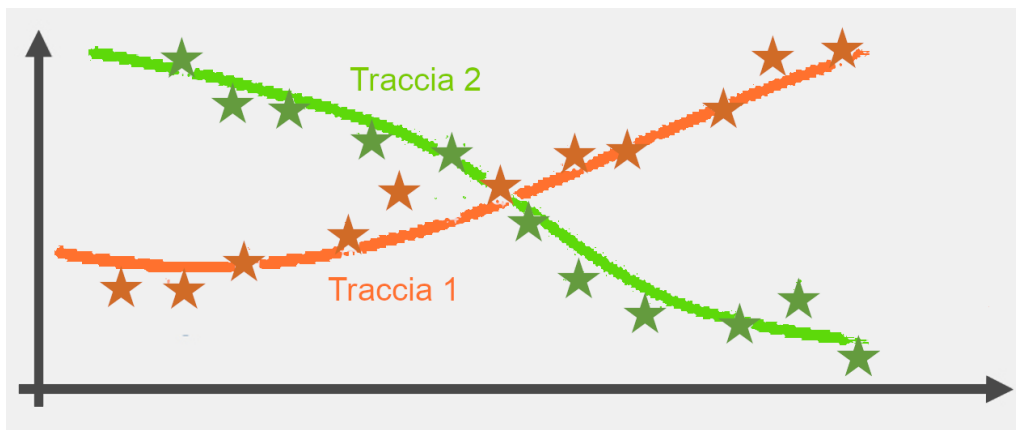


Figura 3.2.1: Associazione dei cluster alle tracce

Alcune tecniche per effettuare l'associazione dei cluster sono la Joint Probabilistic Data Association, il Nearest Neighbor e la Fuzzy Logic [4].

3.3 Tracciamento

In seguito all'associazione, il tracciamento dei movimenti è effettuato tramite un algoritmo di tracking, tra i quali uno dei più utilizzati è il filtro di Kalman [5].

Questo tipo di filtro è detto ricorsivo, in quanto permette di stimare lo stato interno di un sistema dinamico lineare a partire da misurazioni rumorose, usando uno o più dei suoi output come input. È di fondamentale importanza per risolvere importanti problemi della teoria del controllo [5].

Nella maggior parte delle applicazioni, lo stato interno ha più gradi di libertà rispetto ai pochi parametri osservabili che vengono misurati. Tuttavia, combinando una serie di misurazioni, il filtro di Kalman riesce a stimare l'intero stato interno.

Nel caso specifico del tracciamento di persone viene implementato un diverso filtro per ogni singola traccia.

Il filtro di Kalman è anche in grado di predire la posizione del soggetto nel caso in cui l'osservazione in un determinato istante temporale non sia disponibile, basandosi sul modello fisico adottato. Per questo motivo riesce a mantenere la traccia di un soggetto anche se per breve tempo questo non è effettivamente rilevato dal radar.

3.4 Problema dell'occlusione

Un problema frequente che può verificarsi durante il tracciamento di una persona tramite l'utilizzo di un radar è la possibilità che si creino occlusioni che limitino le capacità di rilevamento del radar stesso [2].

Oggetti presenti nella stanza, come mobili o muri, tendono a ridurre l'area che i sensori sono in grado di coprire, limitando le performance del sistema di rilevamento. Inoltre le diverse persone che si muovono nei locali possono occludersi a vicenda, nascondendo parti delle rispettive tracce.

Risulta quindi necessario ricorrere all'uso di radar multipli, allo scopo di unire le informazioni provenienti da vari sensori distribuiti all'interno dell'ambiente.

4 Tracciamento di persone con radar multipli

Come precedentemente citato, il tracciamento di soggetti tramite un singolo radar è fortemente limitato dal problema dell'occlusione [2]. Inoltre, i dispositivi radar a onde millimetriche disponibili in commercio hanno una portata massima che va dai 6 agli 8 metri [1].

Questi fattori rendono necessario l'utilizzo di più radar per tracciare i movimenti degli individui, che vengono posizionati nell'ambiente in modo da ottimizzare la copertura dell'area di interesse, in base alle esigenze specifiche di ogni caso. Avendo a disposizione più sensori che illuminano lo spazio, possono essere sfruttate le regioni comuni per effettuare la fusione dei dati provenienti da diversi dispositivi.

4.1 Requisiti

La fusione dei dati provenienti da dispositivi diversi richiede che siano soddisfatti alcuni prerequisiti [6]. In particolare, è necessario:

- 1) conoscere la posizione e l'orientamento reciproco di ogni radar rispetto agli altri;
- 2) trasformare i dati provenienti dai diversi sensori, ognuno nel suo sistema di riferimento, in un sistema di riferimento comune.

Per quanto riguarda il punto 1), ci sono diverse possibilità. Ad esempio, si potrebbero misurare manualmente i parametri di rototraslazione e usarli per effettuare la trasformazione. Tuttavia, ciò non è pratico, è dispendioso in termini di tempo e risorse e risulta maggiormente pronò a errori umani.

Un'altra possibilità è quella di progettare algoritmi che riescano a stimare automaticamente queste quantità. Uno di questi algoritmi, recentemente sviluppato da ricercatori dell'Università di Trento e di Padova, è mmSCALE [7], e nel seguito si assumerà di usare questo algoritmo per ottenere le posizioni reciproche dei radar.

mmSCALE stima le posizioni e gli orientamenti relativi di una rete di radar ad onde millimetriche, processando le traiettorie delle persone che si muovono all'interno dei vari campi visivi. La precisione di tale stima è stata valutata in un errore mediano della posizione e dell'orientamento pari a 0.18m e 2.86°, rispettivamente.

Uno dei vantaggi derivanti dall'utilizzo di tale metodo è che non sono necessarie traiettorie specifiche o condizioni controllate perché l'algoritmo verifica autonomamente la qualità della calibrazione nel tempo ed è robusto alla presenza di soggetti multipli.

Per quanto riguarda il punto 2), si pone il problema di scegliere il sistema di riferimento da usare come sistema di riferimento comune. Un possibile scelta è quella di usare uno dei radar come radar di riferimento. Tuttavia, non tutti i radar sono equivalenti, in quanto il posizionamento reciproco influisce sulla qualità della trasformazione da attuare (*vedi Sezione 5*). Il compito di scegliere il migliore, tra i radar presenti, per essere usato come radar di riferimento prende il nome di elezione del cluster head e sarà spiegato in maggior dettaglio nei paragrafi successivi.

4.2 Fusione dei dati

Come precedentemente discusso, se sono soddisfatti i requisiti presentati in *Sezione 4.1*, è possibile fondere i dati di più sensori. Ciò permette di ottenere diversi vantaggi:

- Avendo più informazioni a disposizione è possibile migliorare le performance di rilevamento utilizzando i dati provenienti dai diversi punti di vista;
- È possibile limitare il problema dell'occlusione sfruttando i diversi punti in cui vengono posizionati i radar;
- Unendo i dati di più radar, è possibile monitorare in maniera uniforme aree dalla superficie più estesa rispetto a quanto si potrebbe fare con un singolo sensore.

4.3 Scenario

Lo scenario considerato per questo lavoro di tesi consiste in una o più stanze in cui siano dispiegati diversi radar, con o senza sovrapposizione dei rispettivi campi visivi. Ciò che si vuole ottenere è un tracciamento unificato ottenuto dall'unione dei dati di tutti i sensori. Quindi, nel momento in cui una persona entra nell'ambiente, questa deve essere rilevata da tutti i radar che la illuminano e i suoi spostamenti tracciati fondendo le misure acquisite da tutti i sensori contemporaneamente. Si può trovare uno scenario d'esempio in *Figura 4.3.1*.

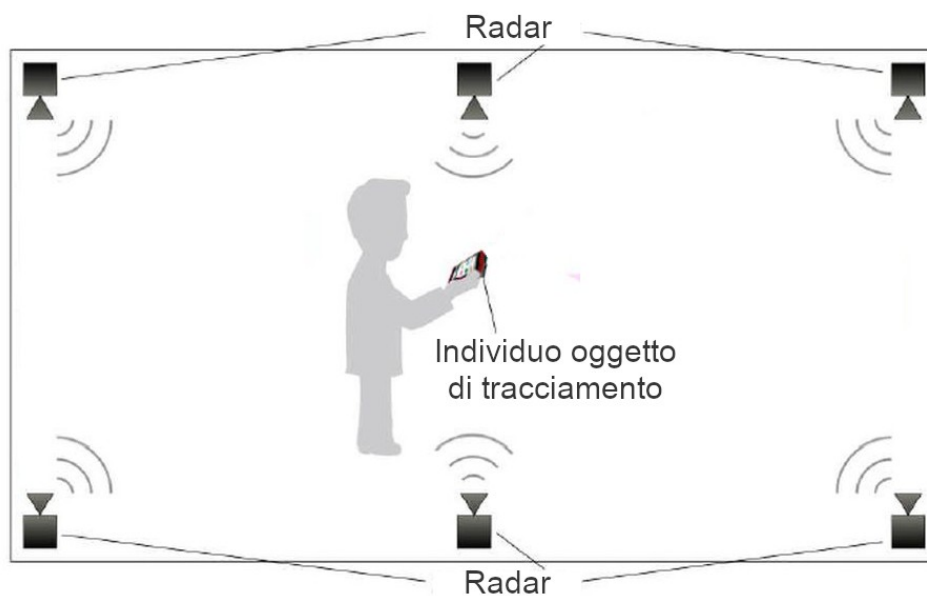


Figura 4.3.1: Esempio di scenario

Si assume, inoltre, che le posizioni reciproche dei radar siano note e che sia nota anche la topologia della rete di radar. La topologia può essere rappresentata da un grafo in cui i nodi siano i singoli radar e gli archi siano presenti tra tutti quei nodi rappresentanti radar i cui campi visivi siano, almeno in parte, sovrapposti.

Questa scelta deriva dal fatto che si vogliono fondere le misure dei soli radar che hanno dati in comune relativi agli stessi soggetti e questo è possibile solo se parte dei campi visivi si sovrappone. Il ruolo del grafo rappresentante la topologia della rete sarà meglio chiarito in seguito, nella *Sezione 5*.

5 Elezione del cluster head

La fusione dei dati provenienti dai diversi radar richiede che le posizioni dei soggetti rilevate da ogni sensore vengano trasformate in un sistema di riferimento comune. Avendo a disposizione algoritmi in grado di stimare la posizione reciproca dei radar in modo automatico (*vedi Sezione 4.1*) si assume di scegliere come sistema di riferimento comune il sistema di riferimento di uno dei radar, che sarà detto cluster head.

La scelta di tale cluster head non può essere casuale, in quanto le caratteristiche fisiche dell'ambiente favoriscono i sensori con specifici posizionamenti. Ad esempio un radar posto all'estremo di un cluster risulterebbe inadeguato come head, dato che per trasformare i campioni da esso generati sarebbero necessarie maggiori trasformazioni intermedie rispetto ad un sensore interno.

In questo modo si assisterebbe ad una degradazione della qualità dei dati, causata dall'imprecisa conoscenza delle posizioni reciproche dei radar fornite da mmSCALE.

Il problema della scelta del cluster head tra un insieme di sensori prende il nome di cluster head election. Si tratta di un problema ampiamente studiato in passato per le reti mobili, per cui è possibile adattare alcuni algoritmi preesistenti al caso d'uso esposto in questa tesi.

5.1 Distributed Mobility-Adaptive Clustering

L'algoritmo *Distributed Mobility-Adaptive Clustering* (DMAC) [8], proposto per reti di dispositivi mobili ad hoc, è ideato per partizionare i nodi della rete in cluster e fornire un'organizzazione gerarchica alla rete stessa. L'algoritmo è distribuito in quanto richiede, come uniche informazioni, quelle relative ai vicini di ogni nodo, e non necessita, invece, di conoscenze riguardanti tutto il resto della rete, che comporterebbe problemi di scalabilità. È inoltre un algoritmo dinamico che permette di ristrutturare e adattare la gerarchia dei nodi alla creazione e rimozione di archi tra i nodi della rete.

DMAC è eseguito in ogni nodo della rete e ognuno di questi nodi deve essere associato un identificativo (ID) e un peso. Ad eccezione della routine iniziale, l'algoritmo funziona per scambio di messaggi: una specifica procedura è istanziata nel nodo a seconda dei messaggi ricevuti dagli altri nodi.

Sono utilizzati due tipi di messaggi: $CH(v)$, impiegato dal nodo v per informare i suoi vicini del fatto che è stato eletto clusterhead, e $JOIN(v, u)$, con cui un nodo v comunica ai suoi vicini che entrerà a far parte del cluster il cui clusterhead è il nodo u .

Sono inoltre introdotte le variabili $Cluster(v)$ e $Clusterhead$ per indicare l'insieme di nodi nel cluster il cui clusterhead è v e il clusterhead di un nodo, rispettivamente. Infine, la variabile booleana $Ch(v)$ è impostata a un valore vero se v ha inviato un messaggio CH.

Quella che segue è la descrizione delle sei procedure che sono eseguite da ogni nodo v .

- *Init.* Nella fase di inizializzazione del clustering, o quando il nodo è stato aggiunto alla rete, il nodo stesso esegue la procedura necessaria a determinare il proprio ruolo. Se tra i suoi vicini è presente almeno un clusterhead con peso maggiore rispetto al suo, allora v entrerà a far parte del suo cluster. In caso contrario si proclamerà clusterhead.
- *Link_failure.* Ogni nodo è portato a conoscenza della rottura di un link da un servizio di livello più basso. In questo caso v verifica se il suo ruolo sia quello di clusterhead e se il nodo u caduto appartenesse al suo cluster. In questo caso v rimuove u da $Cluster(v)$. Se v è un nodo ordinario e u era il suo clusterhead, allora sarà necessario determinare nuovamente il ruolo di v . A tal proposito, v verifica se esista almeno un clusterhead z il cui peso sia maggiore di quello di v . Nel caso sia trovato, v entra nel cluster di z , altrimenti v diventa clusterhead esso stesso.
- *New_link.* Lo stesso servizio che informa i nodi della rottura di un link entra in funzione nel caso siano creati nuovi link. Detto u il nuovo nodo in questione, v verifica se u sia un clusterhead. Se l'esito è positivo e il peso di u è maggiore

del peso dell'attuale clusterhead di v , allora, a prescindere dal suo ruolo, v entra nel cluster di u .

- *Ricezione di JOIN(u, z)*. In caso di ricezione del messaggio di JOIN, il comportamento di v varia in base al fatto che sia o meno clusterhead. In caso affermativo, v verifica se u sta entrando nel suo cluster (u è aggiunto a $Cluster(v)$) o se u facesse parte del suo cluster e si stia apprestando ad inserirsi in un altro (u è rimosso da $Cluster(v)$). Se v non è clusterhead, deve verificare che u lo sia. Solo in questo caso, v deve decidere il suo ruolo: si unirà al clusterhead con peso maggiore tra i suoi vicini o, in alternativa, sarà lui stesso clusterhead.
- *Ricezione di CH(u)*. Quando un vicino u diventa clusterhead, nel momento in cui è ricevuto il messaggio CH corrispondente, il nodo v verifica se debba entrare nel suo cluster in base al suo peso.

La struttura del DMAC garantisce che le tre seguenti proprietà, utili per un algoritmo di clustering di reti ad hoc, siano soddisfatte:

1. Ogni nodo ordinario ha almeno un clusterhead come vicino.
(proprietà di dominanza)
2. Ogni nodo ordinario si affilia con il clusterhead nel suo vicinato avente peso maggiore.
3. Due clusterhead non possono essere vicini.
(proprietà di indipendenza)

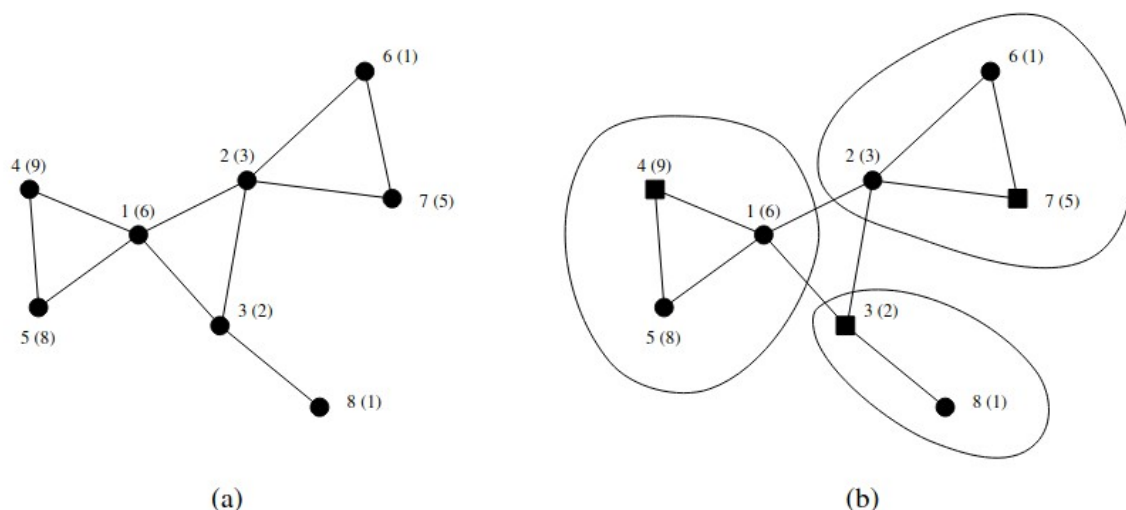


Figura 5.1.1: Una rete clusterizzata correttamente con DMAC

In *Figura 5.1.1* si può trovare un esempio di rete prima e dopo l'operazione di clustering tramite DMAC. I cerchi indicano i nodi ordinari, mentre i quadrati rappresentano i cluster head. Il numero fuori dalle parentesi è l'ID di ogni nodo, mentre tra parentesi è riportato il rispettivo peso.

5.2 Progettazione del peso per DMAC

L'esito dell'algoritmo DMAC dipende dal valore assegnato al peso di ogni nodo. Risulta quindi necessario progettare quest'ultimo accuratamente in modo da ottenere una divisione in cluster che rispetti le successive esigenze di fusione dei dati. Partendo dall'osservazione che il nodo nella posizione migliore per essere cluster head sia quello localizzato più "centralmente" rispetto agli altri nodi suoi vicini, una scelta ragionevole è quella di usare come peso la somma delle aree di campo visivo che ogni radar ha in comune con gli altri radar. L'idea alla base di questa scelta è che il radar con il più alto valore di campo visivo sovrapposto con altri radar sia quello con il quale il maggior numero di radar possano trasformare le proprie misure con una singola trasformazione, evitando il più possibile l'uso di trasformazioni intermedie che possono degradare la qualità dei dati [7].

Nell'immagine a fianco (*Figura 5.2.1*) è rappresentata una configurazione in cui si può notare come il Radar 1 abbia un peso maggiore rispetto agli altri vicini.

Il calcolo è quindi eseguito a partire dalle geometrie dei campi visivi, ottenuti in base alla portata dei radar, al loro angolo di apertura e orientamento. È sufficiente trovare le intersezioni dei relativi settori e calcolarne l'area.

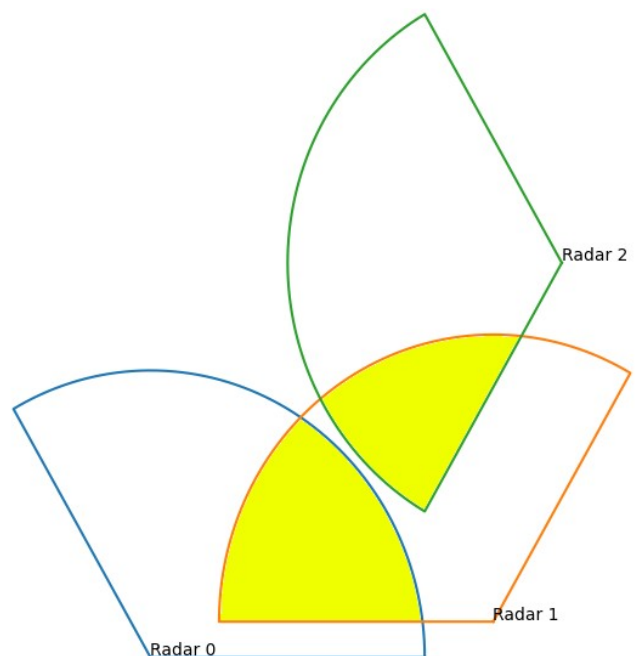


Figura 5.2.1: In giallo il campo visivo in comune

Come evidenziato in *Sezione 3.4*, la presenza di eventuali oggetti nell'ambiente potrebbe ridurre l'area entro la quale il radar è in grado di osservare la presenza di una persona.

Di conseguenza, si configura la possibilità che le sezioni di campi visivi in comune tra i vari radar siano in realtà differenti. Per questo motivo, in futuro, potrebbe essere necessario progettare dei nuovi modi per calcolare le parti di campi visivi in comune. Tuttavia, non essendo questo lo scopo dello studio, per ora ci si limiterà ad usare il sistema di calcolo poc'anzi presentato.

6 Simulatore

Con lo scopo di testare in maniera veloce ed efficace l'implementazione dell'algoritmo di clusterizzazione DMAC con una rete di radar senza aver bisogno di impiegare dispositivi fisici, abbiamo progettato un simulatore utilizzando Python.

6.1 Struttura e implementazione

Il simulatore consente di creare dei radar virtuali e posizionarli nello spazio 2D, ognuno determinato da posizione, orientamento, portata e angolo di apertura. È possibile creare delle traiettorie che simulino il movimento di persone e i singoli radar registrano il movimento delle persone, ognuno nel proprio sistema di riferimento, ogni qual volta la traiettoria passi entro il proprio campo visivo. Inoltre, i radar virtuali sono dotati di capacità di comunicazione tra loro, rendendo possibile lo scambio di messaggi necessario per l'implementazione di DMAC.

L'applicativo in questione è formato da due script, un main e una libreria contenente due classi. La prima, chiamata Sensor, è costruita per fare in modo che ognuna delle sue istanze rappresenti un radar posizionato nell'ambiente. Presenta al suo interno parametri fisici, ossia posizione, rotazione rispetto all'asse verticale, angolo di apertura e range del campo visivo, oltre ai parametri relativi a DMAC discussi nella *Sezione 5.1*, necessari all'algoritmo per funzionare correttamente. In *Figura 6.1.1* sono mostrati i parametri usati per istanziare alcuni oggetti Sensor di esempio.

```
sensors_data = [ {"t": [-5, 1] , "theta": -30, "opening": 120, "range": 8},  
                 {"t": [5, 2] , "theta": 30 , "opening": 120, "range": 8},  
                 {"t": [7, 12] , "theta": 90 , "opening": 120, "range": 8},  
                 {"t": [-3, 18], "theta": 180, "opening": 120, "range": 8} ]
```

Figura 6.1.1: Esempio di parametri fisici relativi a quattro sensori

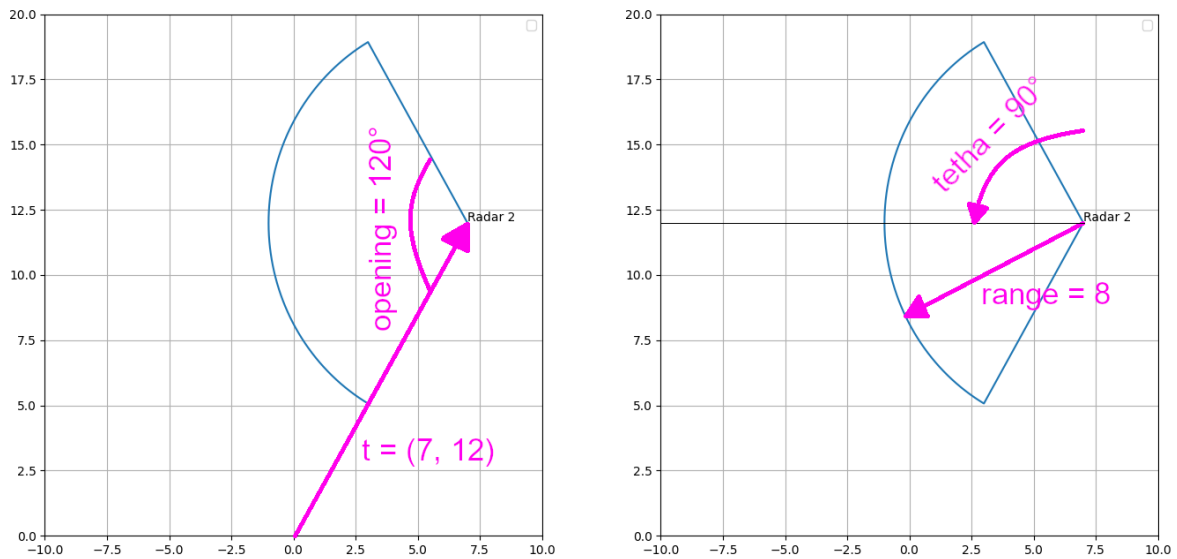


Figura 6.1.2: Rappresentazione grafica del terzo sensore nella figura precedente

La seconda, che prende il nome di Central, funge da servizio di basso livello che permette di mettere in comunicazione tra loro i vari sensori, scambiando messaggi di informazione e gestendo le tracce dei movimenti delle persone.

Per dispiegare un radar è quindi necessario generare un'istanza della classe Sensor e avvisare il Central del nuovo arrivo, il quale provvede ad inserire il radar in una lista di sensori in gestione e avvia il processo di clusterizzazione. Una volta che i sensori hanno terminato di scambiarsi messaggi in base al funzionamento di DMAC, è raggiunta la stabilità della rete e ogni dispositivo è a conoscenza del proprio ruolo - ordinario o clusterhead - e del cluster di riferimento (Figura 6.1.3).

```

laura@laura-PC:~/Scaricati/Sviluppo$ python3 -W ignore main.py
Sensor 0 is in the cluster of sensor 1
Sensor 1 is CH with cluster [1, 0, 2]
Sensor 2 is in the cluster of sensor 1
Sensor 3 is CH with cluster [3]

```

Figura 6.1.3: Il simulatore informa del ruolo che hanno i vari nodi

Inoltre, è possibile fornire al simulatore liste di punti nello spazio cartesiano, rappresentanti le tracce di movimento delle persone, e fare in modo che i radar le rilevino. Le posizioni sono fornite ai sensori a passi discreti. Ogni radar registra, man mano, i punti che si trovano all'interno del proprio campo visivo e li salva in delle tracce interne, associando ad ognuna un ID corrispondente. Ogni radar salva le tracce nel proprio sistema di riferimento.

6.2 Utilizzo

In questo paragrafo sono descritti i passi da compiere per l'utilizzo del simulatore.

Innanzitutto è necessario definire i radar, in termini dei relativi parametri. Questi vanno salvati all'interno di appositi dizionari. Successivamente, i dizionari vanno inseriti in una lista, chiamata, *sensors_data*, contenente tanti elementi quanti sono i sensori che si intende dispiegare, in modo da generare per ognuno un'istanza della classe *Sensor* con i relativi parametri.

Poi, l'algoritmo provvede ad aggiungere le istanze nella lista di sensori della classe *Central*, che si occupa automaticamente delle fasi successive di configurazione. In particolare avvia il processo di clusterizzazione basato su DMAC e permette lo scambio di messaggi tra i vari radar.

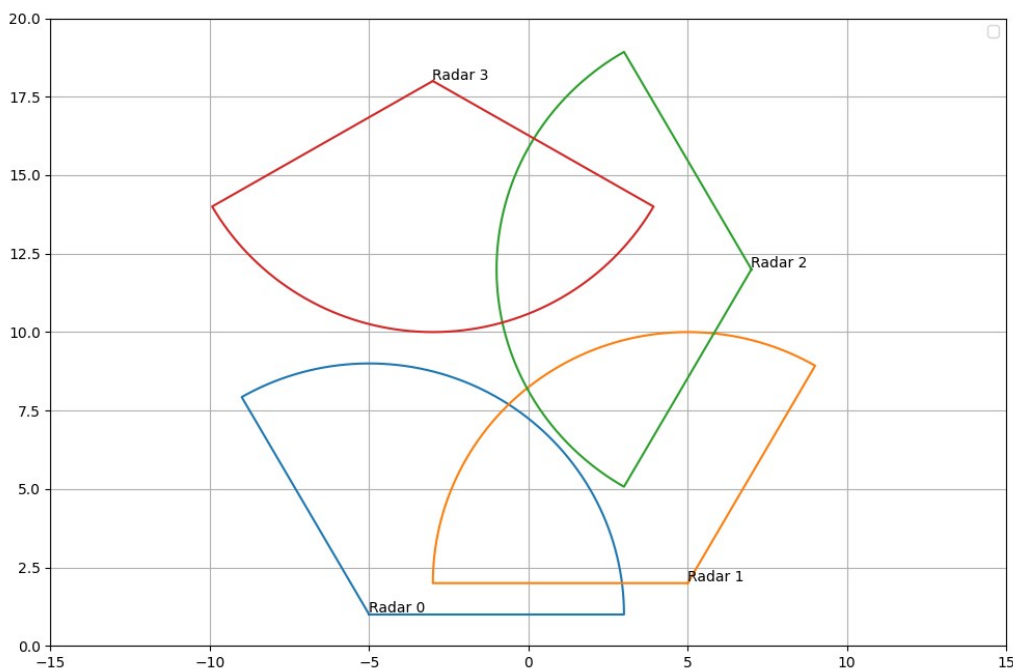


Figura 6.2.1: Rappresentazione di un dispiegamento di radar

L'ultimo passo manuale da eseguire è la generazione delle tracce di movimento nella forma di liste di punti nello spazio cartesiano che verranno poi passate ai radar.

```

track_0 = {"id": 0, "pos": [[7, 1],
                             [7, 2],
                             [6, 3],
                             [5, 4],
                             [4, 5],
                             [3, 6],
                             [3, 7],
                             [2, 8]], "start_time": 0}

```

Figura 6.2.2: Esempio di traccia

Per creare le tracce di test in maniera più agevole, è stato utilizzato un programma in linguaggio matlab, già a disposizione, che permette di costruire delle traiettorie fluide tramite interpolazione *spline*, a partire da una serie di punti selezionati con il mouse da un interfaccia grafica (Figura 6.2.3).

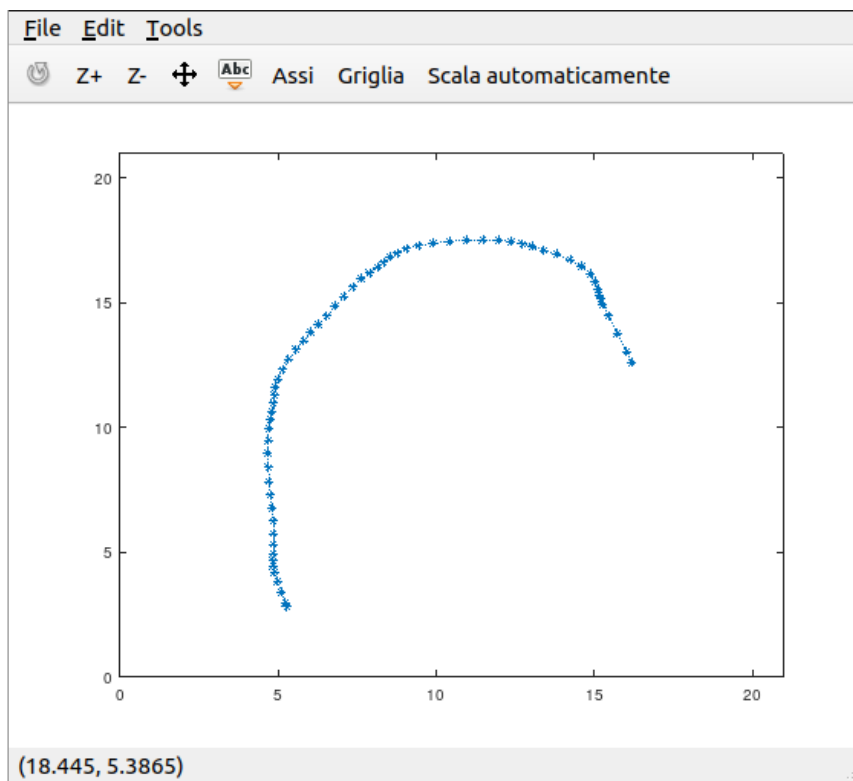


Figura 6.2.3: Generazione di tracce su GNU Octave, applicazione software per l'analisi numerica

Dopodiché, l'algoritmo è pronto per essere eseguito con in Python.

Grazie all'utilizzo di due librerie, chiamate Shapely e Matplotlib, il software è in grado di renderizzare una rappresentazione dell'ambiente contenente i campi visivi dei sensori e le tracce di movimento delle persone (Figura 6.2.1).

In seguito, sarà sfruttata questa possibilità per illustrare diverse configurazioni di radar, al fine di verificare la corretta implementazione dell'algoritmo DMAC.

7 Risultati

In questo capitolo è analizzato il funzionamento dell'algoritmo di clusterizzazione DMAC in tre diversi scenari, con i radar dispiegati ogni volta in maniera differente. L'obiettivo è verificare che la rete si configuri in modo ottimale in ognuna di queste situazioni, tenendo conto dei parametri di posizionamento che caratterizzano i sensori.

7.1 Funzionamento di DMAC

Prima configurazione

Si consideri uno scenario in cui tutti i quattro radar presenti abbiano regioni del campo visivo che si sovrappongono a vicenda.

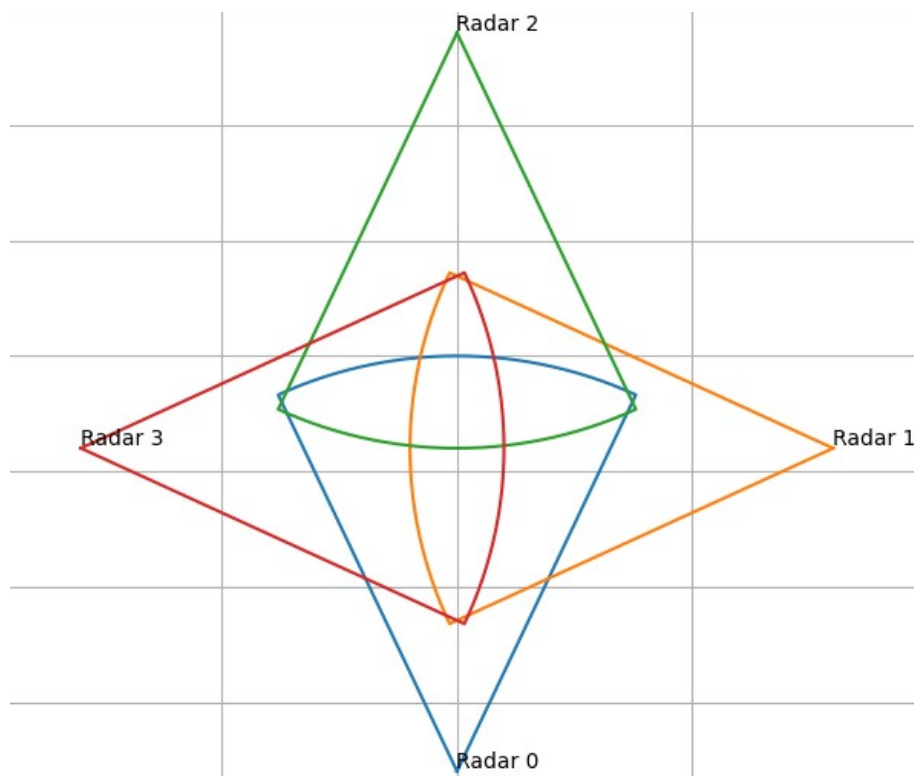


Figura 7.1.1: Prima configurazione di radar

Osservando la Figura 7.1.1 è ragionevole ipotizzare che i sensori formino un unico cluster, e che uno di essi assuma quindi il ruolo di cluster head.

Per raggiungere tale decisione il simulatore calcola, per ogni radar, la somma delle porzioni dei campi visivi comuni che costituiranno il peso di ogni nodo (vedi Sezione 5.2).

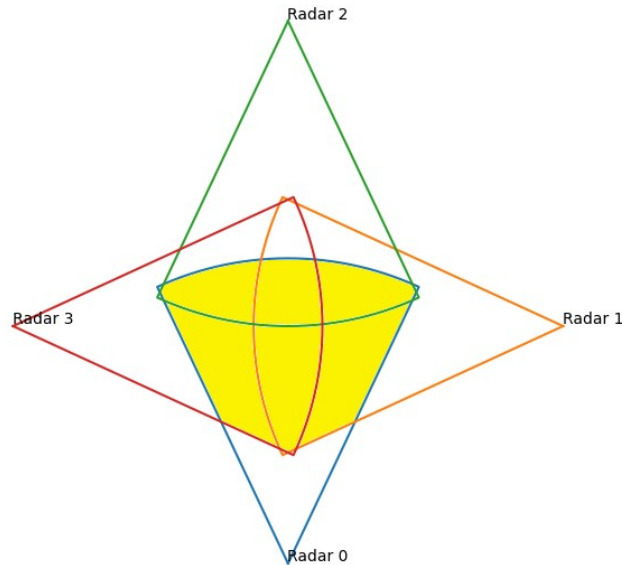


Figura 7.1.2: In giallo il peso di Radar 0

È possibile osservare come Radar 0 abbia la porzione di area sovrapposta più grande rispetto a tutti gli altri sensori, e, di conseguenza, è scelto come cluster head (Figura 7.1.3). In Figura 7.1.4 è mostrata la topologia della prima configurazione.

DISPOSITIVO	PESO	CH	CLUSTER	CLUSTER HEAD
Sensor 0	49.08	✓	[0, 1, 2, 3]	
Sensor 1	42.14			Sensor 0
Sensor 2	35.19			Sensor 0
Sensor 3	42.14			Sensor 0

Figura 7.1.3: Tabella contenente i pesi e i cluster di tutti i sensori

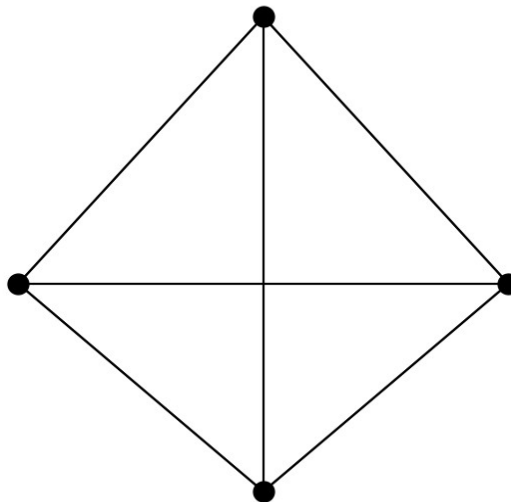


Figura 7.1.4: Topologia della prima rete

Seconda configurazione

Il secondo scenario testato è composto da sei radar, come mostrato in figura.

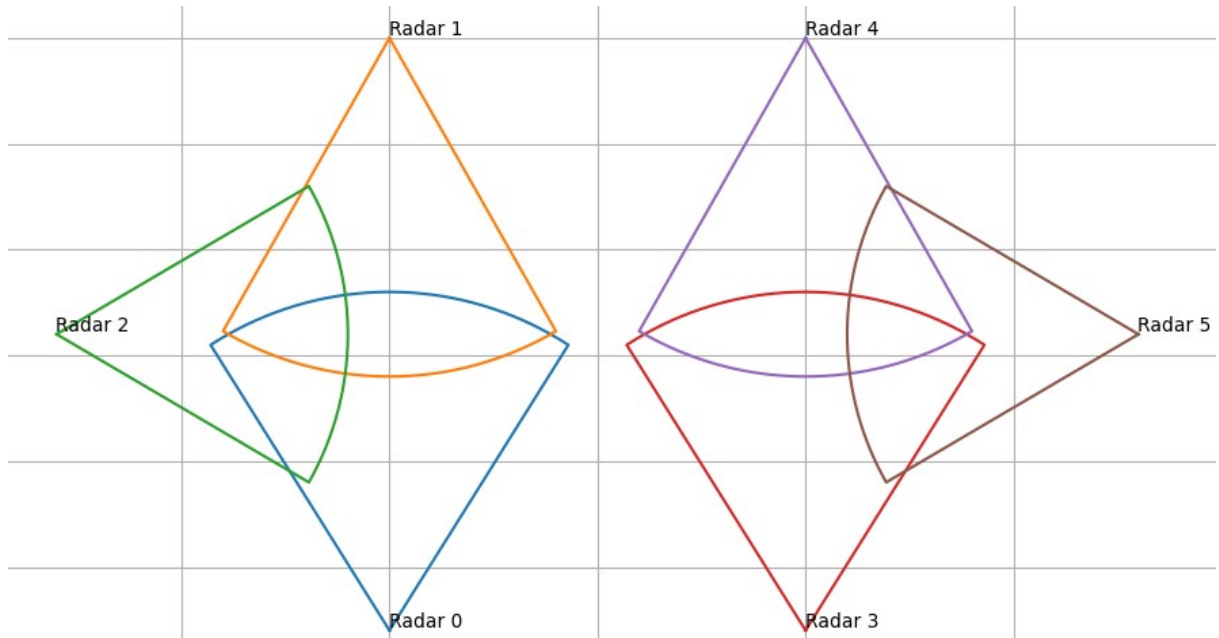


Figura 7.1.5: Secondo dispiegamento di radar

In questo caso andranno a formarsi due cluster speculari, ognuno dei quali con il rispettivo cluster head. Come prima è effettuato il calcolo dei pesi ed è eseguito l'algoritmo DMAC, i cui risultati sono riportati in *Figura 7.1.6*.

DISPOSITIVO	PESO	CH	CLUSTER	CLUSTER HEAD
Sensor 0	19.11	✓	[0, 1, 2]	
Sensor 1	18.12			Sensor 0
Sensor 2	16.3			Sensor 0
Sensor 3	19.11	✓	[3, 4, 5]	
Sensor 4	18.12			Sensor 3
Sensor 5	16.3			Sensor 3

Figura 7.1.6: Tabella contenente i pesi e i cluster di tutti i sensori

La topologia della rete risulta composta da due gruppi di nodi, accomunati dalla condivisione di una parte dei campi visivi. Infine, in ogni cluster è presente il sensore che possiede il peso maggiore, ed è quindi eletto cluster head.

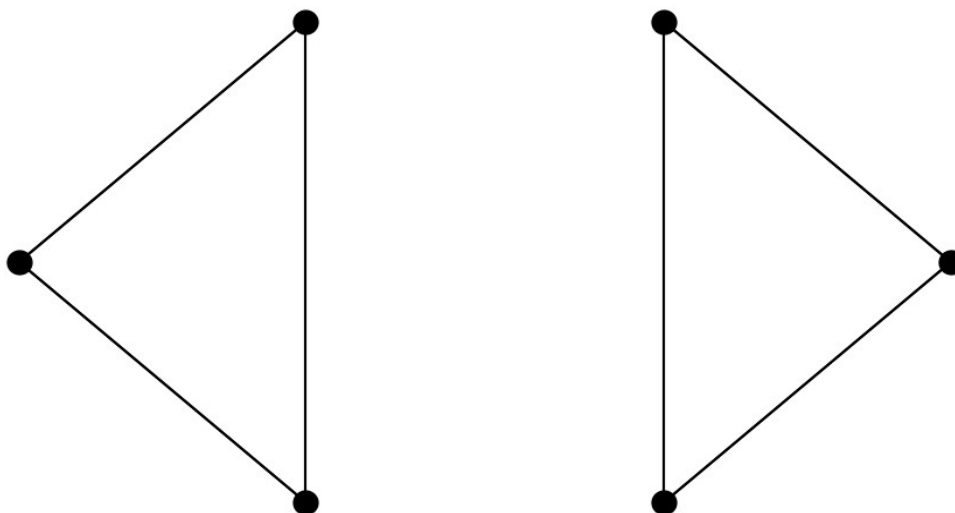


Figura 7.1.7: Topologia della seconda rete

Terza configurazione

Nell'ultimo scenario sono presenti sette radar che compongono tre diversi cluster, raggruppati a due e a tre.

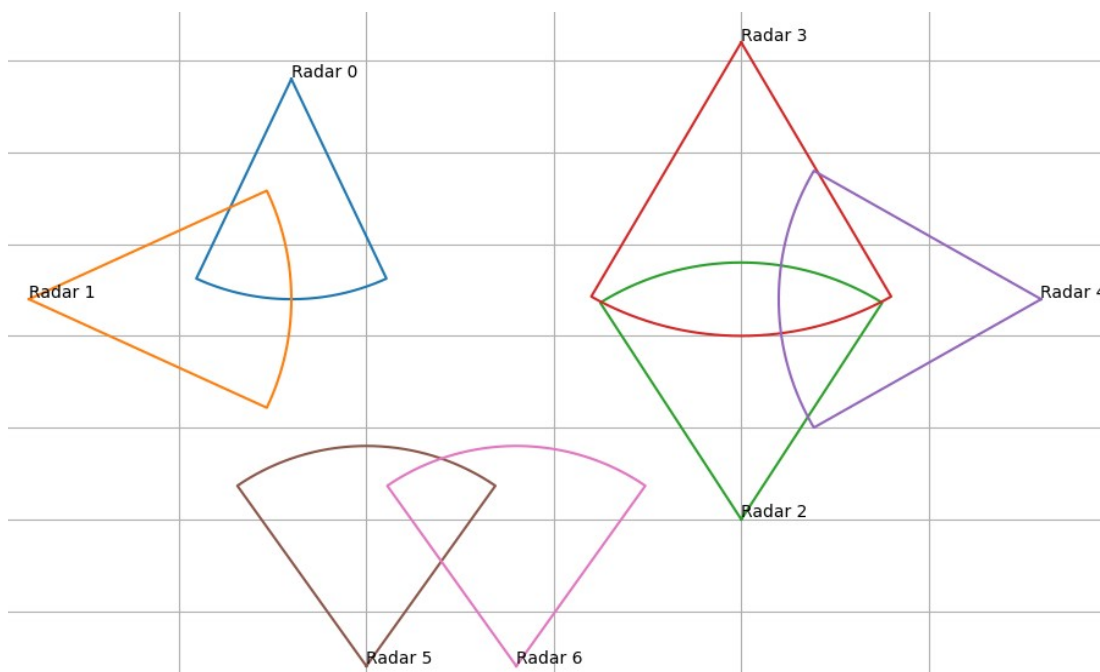


Figura 7.1.8: Terzo dispiegamento di radar

La particolarità di questa configurazione risiede nella presenza di cluster formati da due soli nodi. In tali casi, considerando che il peso è l'area di campo visivo comune, entrambi i sensori saranno equivalenti per DMAC (Figura 7.1.9), e la scelta del cluster head sarà basata sui minimi errori di calcolo commessi durante la determinazione del peso.

DISPOSITIVO	PESO	CH	CLUSTER	CLUSTER HEAD
Sensor 0	4.89	✓	[0, 1]	
Sensor 1	4.89			Sensor 0
Sensor 2	16.49			Sensor 3
Sensor 3	17.75	✓	[3, 2, 4]	
Sensor 4	14.05			Sensor 3
Sensor 5	4.16			Sensor 6
Sensor 6	4.16	✓	[6, 5]	

Figura 7.1.9: Tabella contenente i pesi e i cluster di tutti i sensori

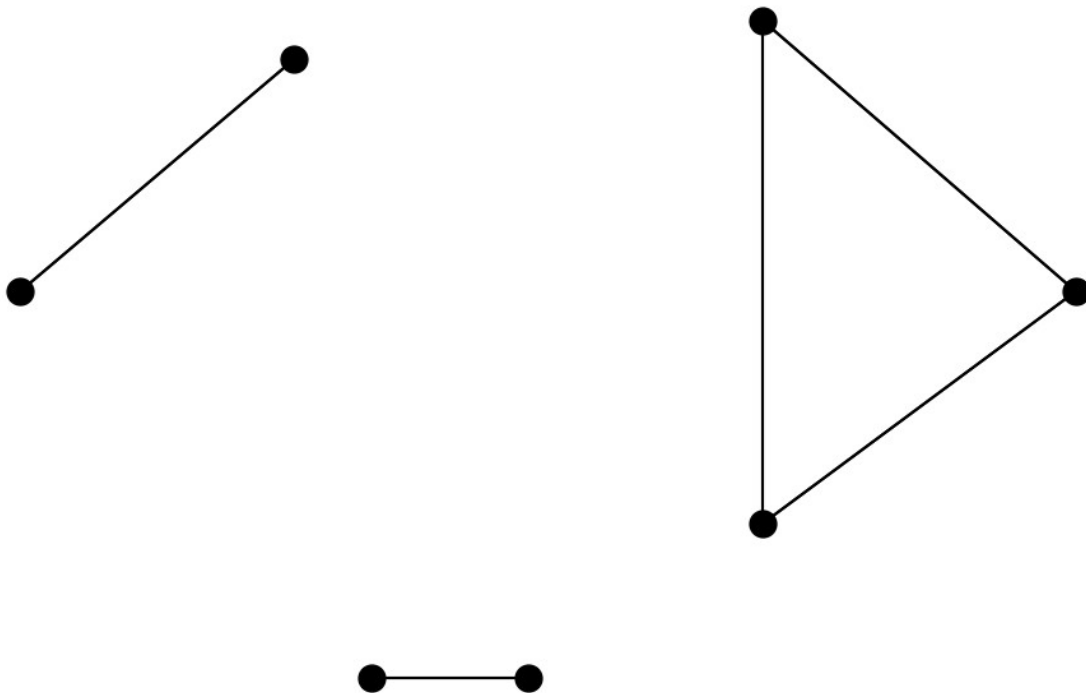


Figura 7.1.10: Topologia della terza rete

7.2 Tracciamento di persone

Nel momento in cui i soggetti passano attraverso il campo visivo dei radar, questi tengono traccia delle posizioni rilevate nel loro sistema di riferimento. Aggiungendo la traccia di una persona ad ognuna delle precedenti configurazioni, è illustrato di seguito il risultato ottenuto.

Le tracce raffigurate sono state generate utilizzando il programma che permette di costruire delle traiettorie fluide tramite interpolazione *spline* (vedi Sezione 6.2).

Prima configurazione

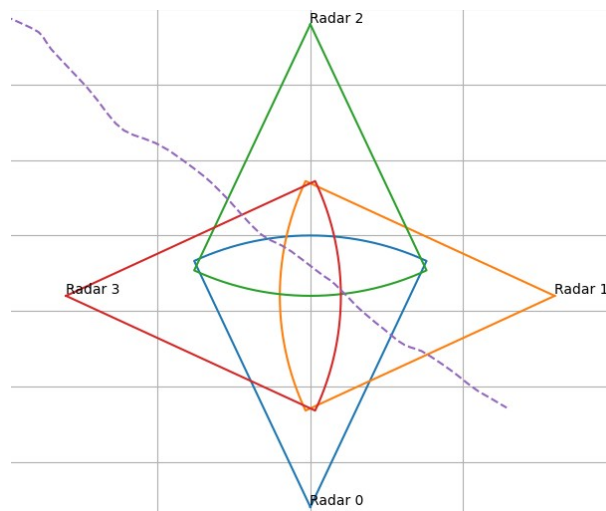


Figura 7.2.1: Topologia della prima rete

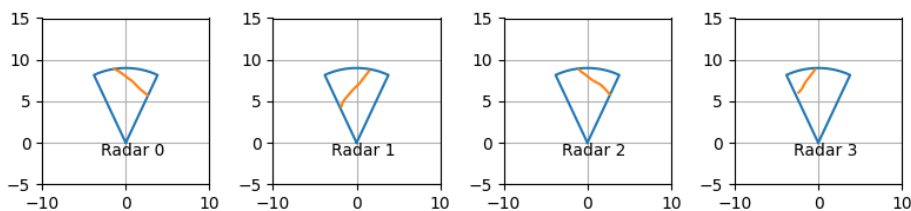


Figura 7.2.2: Tracce dei singoli radar

Seconda configurazione

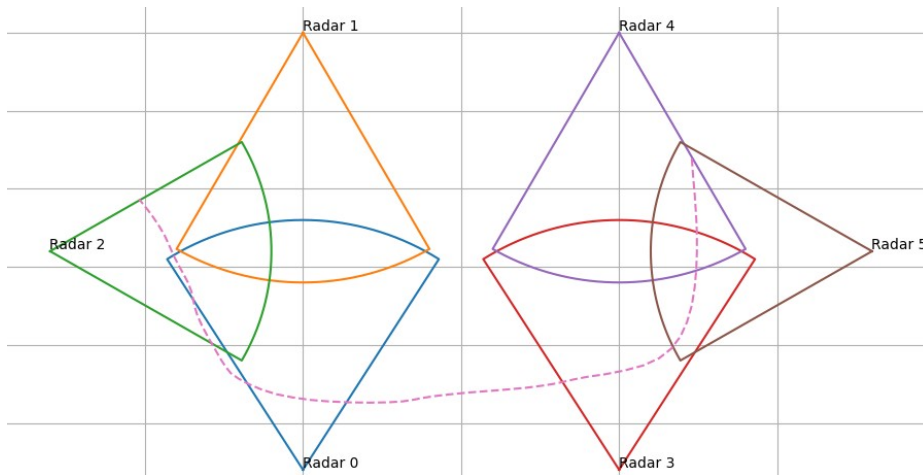


Figura 7.2.3: Topologia della seconda rete

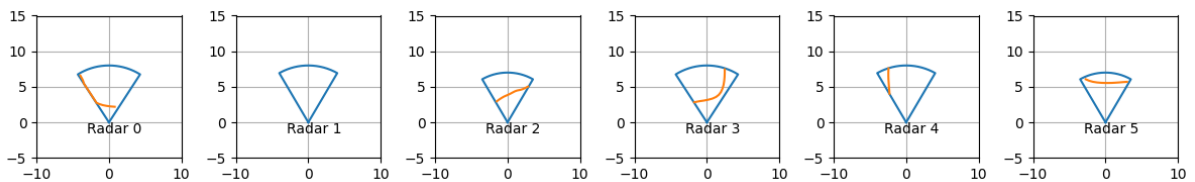


Figura 7.2.4: Tracce dei singoli radar

Terza configurazione

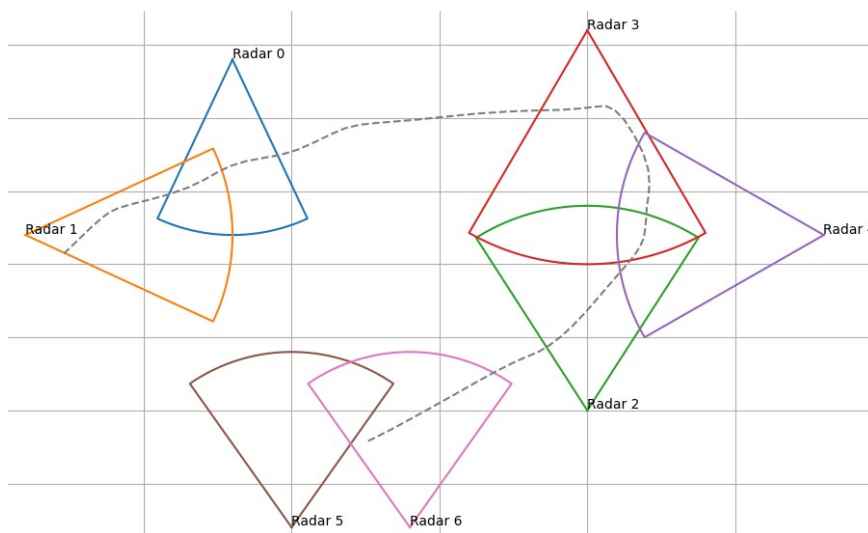


Figura 7.2.5: Topologia della terza rete

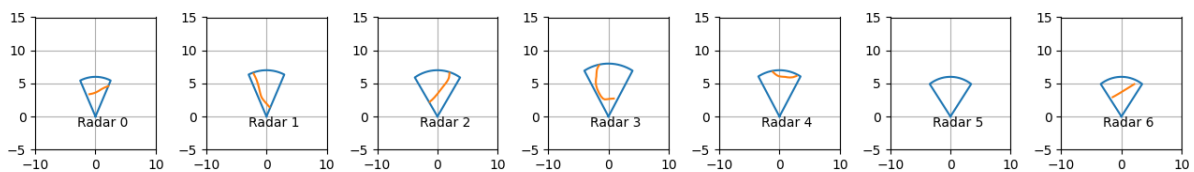


Figura 7.2.6: Tracce dei singoli radar

7.3 Conclusioni

In questo lavoro di tesi è stato sviluppato un simulatore di reti di radar per il tracciamento del movimento delle persone nello spazio. Il programma simula anche la comunicazione tra i dispositivi, rendendo possibile l'implementazione e la sperimentazione di algoritmi distribuiti come DMAC.

Gli esperimenti effettuati permettono di concludere che DMAC è un algoritmo adeguato per la creazione di una gerarchia nelle reti di radar, utile alla gestione della rete stessa, in quanto permette di clusterizzare i sensori ed eleggere il cluster head in maniera distribuita ed efficiente.

Possibili evoluzioni future comprendono sistemi più avanzati per il calcolo del peso dei nodi, basati non più su un calcolo geometrico dell'intersezione dei campi visivi ma sul movimento dei soggetti. In questa maniera, si andrebbero a calcolare le parti dei campi visivi effettivamente sovrapposte, risolvendo il problema dell'eventuale occlusione dei campi visivi stessi dovuta a oggetti statici presenti nell'ambiente, che non può essere predetta a priori a meno di una conoscenza specifica dell'ambiente in cui sarà dispiegata la rete.

Bibliografia

- [1] P. Zhao et al., "mID: Tracking and Identifying People with Millimeter Wave Radar," in Proc. DCOSS, May 2019.
- [2] J. Pegoraro and M. Rossi, "Real-time people tracking and identification from sparse mm-wave radar point-clouds," IEEE Access, May 2021.
- [3] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in 2nd International Conference on Knowledge Discovery and Data Mining, Aug 1996.
- [4] A. Macaveiu, A. Campeanu, and I. Nafornita, "Kalman-based tracker for multiple radar targets," in Proc. Int. Conf. Communication, Bucharest, Romania, 2014.
- [5] R. E. Kalman, "A new approach to linear filtering and prediction problems," ASME Transactions, Journal of Basic Engineering, 1960.
- [6] S. Iwata et al., "Multiradar data fusion for respiratory measurement of multiple people," arXiv preprint arXiv:2107.11525, 2021.
- [7] A. Shastri et al., "mmSCALE: Self-Calibration of mmWave Radar Networks from Human Movement Trajectories," IEEE Radar Conference 2022 (RadarConf22).
- [8] S. Basagni, "Distributed Clustering for Ad Hoc Networks," IEEE Xplore, March 2022.