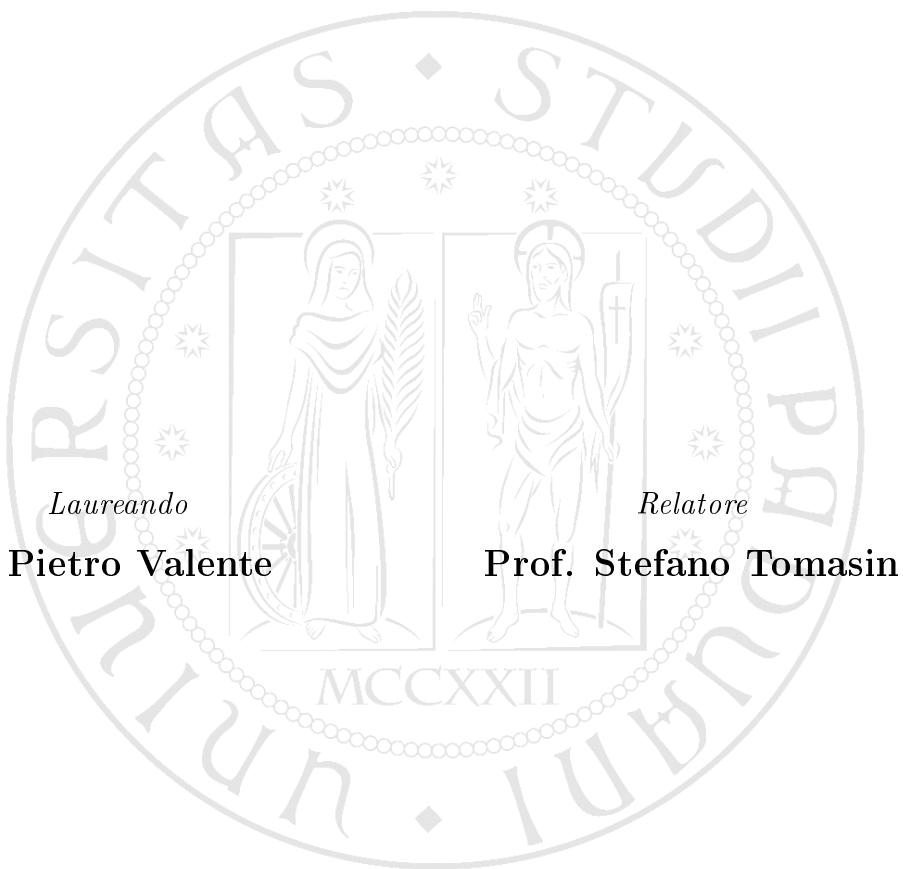


Università di Padova
Corso di Laurea in Ingegneria Informatica

SPOOFING GPS



ANNO ACCADEMICO 2020/2021

A Padova,
un'amica che ha saputo capirmi e darmi risposte,
presentarmi persone importanti e farmi crescere;
resterai sempre nel mio cuore.

Indice

1	Introduzione	1
1.1	Bluetooth	1
1.1.1	Evoluzione	1
1.1.2	Utilizzi e Bluetooth 5.1	3
1.2	GNSS	5
1.2.1	Il primo sistema GNSS: GPS	5
1.2.2	Altri sistemi GNSS	5
1.2.3	Funzionamento sistemi di localizzazione	6
1.2.4	Real-Time Kinematic	7
1.2.5	Networked Transport of RTCM via Internet Protocol . .	8
2	Sperimentazione RSSI	11
2.1	Introduzione allo studio svolto	11
2.2	Misurazioni	12
2.3	Range RSSI - distanza	14
2.4	Attenuazione del modello radio	16
2.5	Anomalia 0.5m	18
2.6	Analisi dati con ostacoli	18
2.7	Conclusioni RSSI	21
3	Sperimentazione RTK	23
3.1	Strumenti usati	23
3.2	Analisi dati	25
3.3	Distanza geodetica	30
3.4	Misurazioni	31
3.5	Conclusioni RTK	32
4	Confronto sperimentazioni	33
4.1	L'algoritmo	33
4.2	Risultati ottenuti	35
4.3	Conclusioni	38

Sommario

L'utilizzo del Global Navigation Satellite Systems (GNSS) durante la vita mondana è progressivamente diventato più importante, se non obbligatorio all'interno di determinati contesti applicativi, con il passare degli anni.

Il GPS è ancora la principale tecnologia che utilizza GNSS e tante sono le applicazioni: navigazione in tempo reale su mappe e carte geografiche, sblocco di porte smart (Auto Unlock di Nuki), sblocco di auto all'avvicinarsi (Tesla model S/X), funzioni di controllo/sicurezza su personale in un'azienda o su merce spedita.

In tutti questi settori la sicurezza è importante e non può essere posta in secondo piano, lo spoofing GPS è proprio un tipo di attacco informatico che ha come obiettivo la falsificazione della localizzazione di un dispositivo.

Questa tesi si pone l'obiettivo di studiare un possibile meccanismo di sicurezza per dispositivi di localizzazione, combinando tecniche di posizionamento ad alta precisione e tecnologia Bluetooth.

Capitolo 1

Introduzione

1.1 Bluetooth

Inventato nel 1995, il Bluetooth è una tecnologia che sfrutta le informazioni incorporate digitalmente sui segnali in radiofrequenza (con frequenza che varia tra 2.402-2.48 GHz) per scambiare dati tra due o più dispositivi. Nato come metodo di comunicazione sicuro e a basso costo, negli anni si è evoluto sia dal punto di vista tecnico che negli utilizzi, fino ad arrivare ad essere un componente comodo, affidabile e quindi immancabile in tutti i dispositivi smart al giorno d'oggi.

1.1.1 Evoluzione

Gli aspetti in cui il Bluetooth è stato migliorato sono tanti, ma possono essere raggruppati in tre categorie: range, velocità dei dati e consumo energetico.¹

Nella prima versione la velocità dei dati arrivava massimo a 1 Mbps, il range aveva una estensione massima di 10 m e il consumo energetico era molto elevato. Lo schema di modulazione utilizzato era il Gaussian Frequency Shift Keying (GFSK), nel quale il trasmettitore modulato alternava due frequenze che rappresentavano i valori logici 1 e 0.²

Nelle versioni successive (Bluetooth 2.0 e 3.0) l'obbiettivo fu quello di ottenere una connessione a corto raggio affidabile e veloce. Questo fu possibile grazie ai nuovi schemi di modulazione che vennero introdotti ($\pi/4$ -DQPSK e 8DPSK), sostituiti successivamente dall'utilizzo del protocollo IEEE 802.11 per il trasferimento di blocchi di dati. Nello schema $\pi/4$ Differential Quadrature Phase-Shift Keying il trasmettitore poteva spedire 4 segnali diversi, differenziati nella fase traslata di $\pi/2$ l'uno dall'altro, a partire da $\pi/4$, da cui il nome (Tab. 1.1 e Fig. 1.1).³

Cambio di fase	Bit Pattern
$-\pi/4$	00
$\pi/4$	01
$-3\pi/4$	10
$3\pi/4$	11

Tabella 1.1: Corrispondenza shift nella frequenza e bit

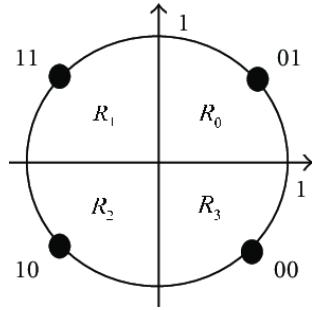


Figura 1.1: Ricezione al ricevitore con regioni di decisione

Lo schema 8 Differential Phase-Shift Keying si comporta in modo simile ma con la possibilità di trasmettere 8 segnali differenti che di conseguenza hanno una traslazione di $\pi/4$ nella fase l'uno dall'altro (Fig. 1.2 e Tab. 1.2).⁴

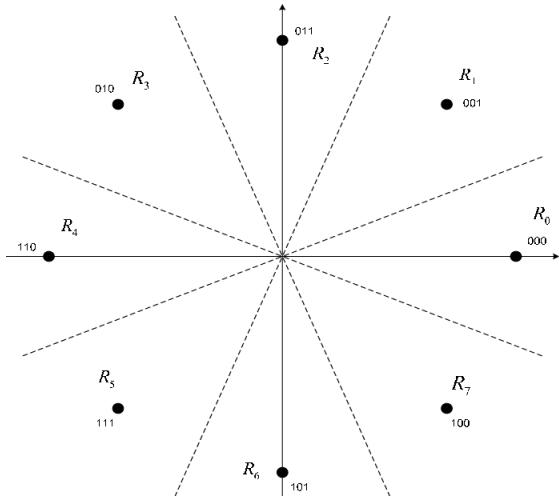


Figura 1.2: Ricezione al ricevitore con regioni di decisione

Cambio di fase	Bit Pattern
0	000
$\pi/4$	001
$\pi/2$	011
$3\pi/4$	010
π	110
$-3\pi/4$	111
$-\pi/2$	101
$-\pi/4$	100

Tabella 1.2: Corrispondenza shift nella frequenza e bit

Con l'introduzione del protocollo IEEE 802.11, la velocità del trasferimento dati non sembrava più essere una limitazione (max 24 Mbps), tuttavia l'elevato consumo energetico rendeva questa tecnologia ancora non matura per l'utilizzo nei dispositivi quotidiani.

Un ulteriore miglioramento avvenne nel 2011 con il Bluetooth 4.0 (chiamato anche BLE, Bluetooth Low Energy): sebbene il suo data throughput (limitato a 1 Mbps) poteva non essere adatto a prodotti che richiedevano un flusso continuo di dati come le cuffie wireless, risultò invece ottimo per altre

applicazioni le cui esigenze erano limitate all'invio di pochi bit periodicamente. Con il focus sul trasferimento dei dati a bassa energia, molte applicazioni Bluetooth alimentate a batteria furono possibili (ad esempio i Beacons).

Il Bluetooth 5.0 (2016) è stato un miglioramento del precedente standard BLE. Infatti, con l'introduzione di 4 diverse velocità di trasmissione (2 Mbps, 1 Mbps, 500 kbps, 125 kbps), tutti i dispositivi poterono sfruttare i miglioramenti legati al consumo energetico ridotto. Interessante sottolineare che, nella fase di ricerca dei dispositivi, venne generalmente utilizzata la trasmissione con bitrate più basso (125 kbps), questo per due principali motivi: modulazioni più basse permettono di ricevere nodi più lontani e assicurare che tutti i dispositivi siano in grado di riconoscere segnali dall'altro dispositivo.⁵

1.1.2 Utilizzi e Bluetooth 5.1

Non esiste limite alla natura di file che possono essere trasmessi: fotografie, documenti, musica e video. Negli ultimi anni, il Bluetooth ha trovato ampia diffusione soprattutto negli smartphone e viene usato per collegare cuffie wireless, ma anche tastiere, mouse e controller senza fili; inoltre, questa tipologia di collegamento viene usata anche per far comunicare dispositivi di casa intelligente (domotica), e Internet of Things (IoT).

La novità principale introdotta nel Bluetooth 5.1 (2019) riguarda il miglioramento della precisione del sistema di localizzazione. Se in passato era possibile determinare solo la distanza da un altro dispositivo Bluetooth, ora, sfruttando la triangolazione dei segnali ottenuti da un certo numero di antenne che inviano/ricevono il segnale con Angle of Arrival (AoA) (Fig. 1.3) e Angle of Departure (AoD) (Fig. 1.4), è possibile determinare la posizione e distanza dell'oggetto che trasmette/riceve i dati.⁶

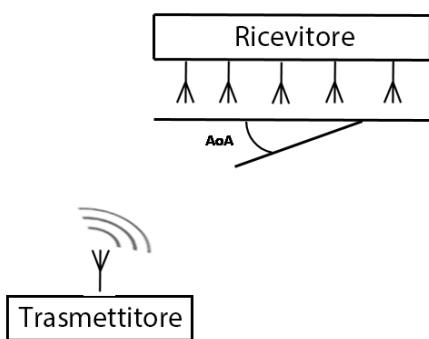


Figura 1.3: Angle of Arrival

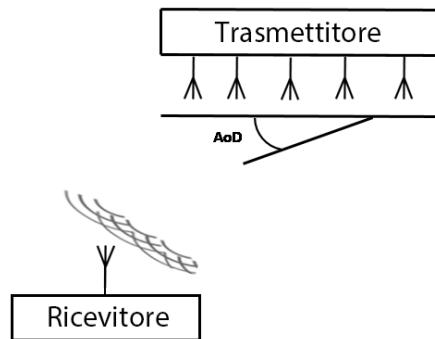


Figura 1.4: Angle of Departure

Questa novità risulta utile per i tag che aiutano a ritrovare gli oggetti smarriti, ma può servire anche per la localizzazione indoor, dove i segnali GNSS faticano ad arrivare.

Un esempio è AirTag di Apple (2021),⁷ un accessorio per iPhone che offre una soluzione semplice e pratica per localizzare gli oggetti. Si presenta come un dischetto dotato di Bluetooth 5.1 che viene inserito all'interno di una custodia, diventando un portachiavi da agganciare ad un oggetto che non vogliamo smarrire (per esempio delle chiavi). Tramite l'apposita applicazione *Dov'è* (Fig. 1.5) è poi facilmente ritrovabile conoscendo sempre direzione e distanza.

Questi esempi ci dimostrano come l'utilizzo del Bluetooth si stia espandendo in utilizzi che vanno oltre il semplice trasferimento di file e come questa continua evoluzione lo renda un componente sempre più indispensabile e utile nella vita di tutti i giorni.



Figura 1.5: Applicazione *Dov'è* su iPhone

1.2 GNSS

Per Global Navigation Satellite System (GNSS) si intende una costellazione di satelliti che forniscono un posizionamento geo-spatiale su un qualunque punto della superficie terrestre.

1.2.1 Il primo sistema GNSS: GPS

Nel 1973 il Dipartimento della Difesa americano decise di cercare una tecnologia per la localizzazione globale, da utilizzare per scopi militari. Fu proprio dopo una sessione di brainstorming al Pentagono che nacque il concetto di Global Positioning System (GPS), originariamente chiamato Navstar GPS.

L'idea fu quella di creare un GNSS di 24 satelliti che permettesse in ogni punto della terra di farne rilevare almeno 4. In questo modo sfruttando la triangolazione dei segnali ricevuti si riusciva ad ottenere una posizione con una precisione accettabile. Dal suo inizio nel 1978, la spedizione dei satelliti ha richiesto diversi anni e si è conclusa nel 1994; il sistema venne poi dichiarato operativo l'anno successivo.⁸

Inizialmente questa tecnologia era destinata solo a usi militari, ma quando nel 1983 l'aereo di linea Korean Air Lines Flight 007 venne abbattuto dai sovietici per aver invaso erroneamente una zona di cielo proibita, l'allora presidente degli Stati Uniti, Ronald Reagan, decise che al completamento della tecnologia, questa sarebbe stata resa disponibile anche per usi civili.⁹

1.2.2 Altri sistemi GNSS

Oltre al GPS altre reti di satelliti sono state realizzate nel tempo, come il Global Navigation Satellite System (GLONASS). Questa tecnologia russa, nata in parallelo a quella americana, alla fine degli anni '90 aveva riscontrato un calo di prestazioni, ma è stata poi restaurata negli anni 2000 con grandi investimenti e può vantare oggi di essere il secondo sistema di navigazione in funzione con copertura globale e di precisione comparabile al GPS.¹⁰

Anche la Cina ha avviato l'implementazione di un sistema GNSS noto come sistema satellitare di navigazione BEIDOU (BDS). Il sistema è stato progettato per essere implementato in due fasi: la fase iniziale per una copertura continentale, mentre la seconda fase per una copertura globale.¹¹

La fase iniziale del sistema BeiDou è diventata ufficialmente operativa nel dicembre 2012, fornendo copertura per il continente asiatico grazie a una rete di 14 satelliti.

La seconda fase del sistema BeiDou è stata completata durante l'estate del 2020 e oltre a fornire una copertura globale, migliora anche quella continentale grazie al ben più ampio GNSS (35 satelliti).

Un altro interessante sistema di navigazione satellitare, creato dall'Unione Europea, è Galileo che è entrato in funzione nel 2016 con un GNSS da 18 satelliti. Il progetto finale prevederà l'utilizzo di 30 satelliti (24 operativi e 6 ricambi), disponibili entro la fine del 2021.¹²¹³

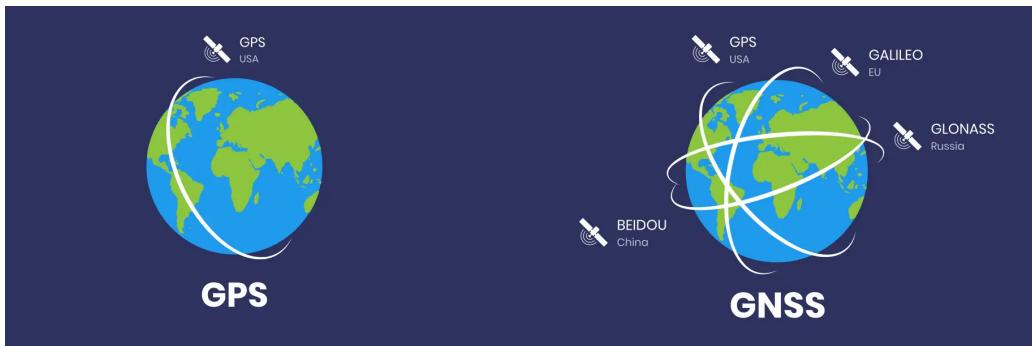


Figura 1.6: Differenza tra GPS e GNSS

1.2.3 Funzionamento sistemi di localizzazione

La tecnica base per la localizzazione dell'utente utilizzando le informazioni dei satelliti è la triangolazione.¹⁴

Questa tecnica sfrutta diverse linee di ricezione provenienti da punti fissi, calcola la distanza da questi e ne determina il punto di convergenza, con l'obiettivo di ridurre il margine di errore.

Nel caso della localizzazione i punti fissi sono quattro satelliti, uno per ogni direzione (x,y,z) e uno per il tempo (t) che viene utilizzato per eliminare l'incertezza sul clock del dispositivo. (Fig. 1.7).

Per calcolare la distanza da un satellite, si utilizza la formula $distanza = velocità * tempo$. I segnali radio inviati dai satelliti viaggiano alla velocità della luce $c = 299.792 \text{ km/s} \approx 300.000 \text{ km/s}$; mentre per misurare il tempo viene calcolato il ritardo di fase tra il segnale spedito e quello ricevuto al ricevitore. Per calcolare il tempo trascorso, il ricevitore utilizza il suo orologio atomico interno, la cui accuratezza può variare in base alla qualità dello strumento; la precisione nel calcolo della distanza è quindi fortemente influenzata da questo parametro.

C'è da sottolineare che il funzionamento di questa tecnica che utilizza GNSS è stata possibile grazie ad alcuni progressi scientifici e ingegneristici

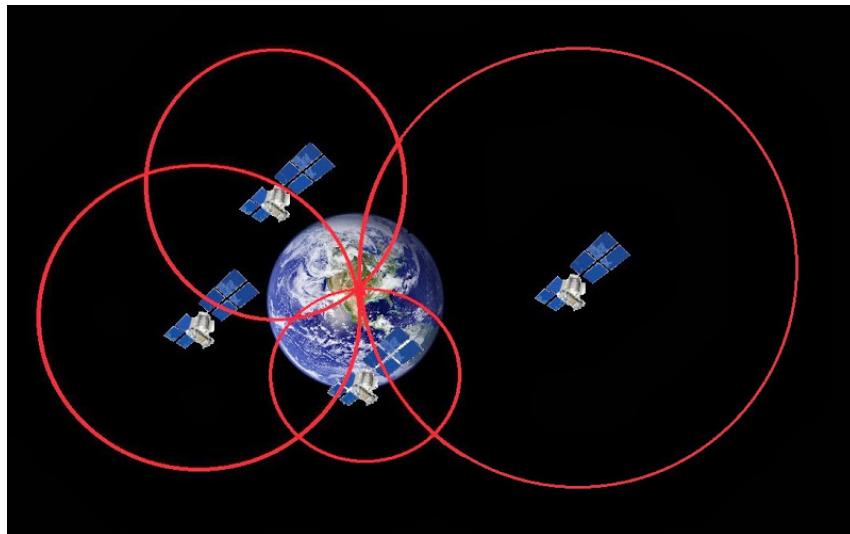


Figura 1.7: Tecnica di triangolazione

avvenuti nel '900: in particolare le teorie della relatività di Einstein e l'invenzione di orologi atomici. Se non venisse considerata infatti la dilatazione temporale (relatività ristretta) e la redshift gravitazionale (relatività generale), si avrebbe un errore nella posizione nell'ordine dei 12km¹⁵ al giorno. Inoltre fu cruciale l'invenzione nel 1949 degli orologi atomici, nel loro funzionamento la base del tempo è determinata dalla frequenza di risonanza di un atomo, questo permette di avere un'estrema precisione.¹⁶

1.2.4 Real-Time Kinematic

Il Real-Time Kinematic (RTK) è una tecnica di positioning che consiste nel migliorare la posizione del ricevitore (Rover) utilizzando la fase d'onda portante e affiancando ai dati del ricevitore quelli provenienti dalla stazione base più vicina, fornendo in questa maniera un'accuratezza di posizionamento centimetrica.¹⁷

Come è stato già descritto, la distanza tra un satellite ed un ricevitore viene misurata dal tempo che un segnale impiega a percorrere il tragitto. Questa misura è soggetta a una serie di errori causati da diversi fattori, tra cui: ritardi introdotti nella ionosfera e troposfera, legati soprattutto alle condizioni meteo; multipath, legato alla riflessione e alla dispersione del segnale; precisione del clock dei satelliti, ecc.

In particolare, il range viene calcolato determinando il numero di cicli che intercorrono tra il satellite e la stazione di Rover, quindi moltiplicando questo numero per lunghezza d'onda del segnale. È necessario un compli-

cato processo, chiamato "ambiguity resolution", per determinare il numero di cicli trascorsi tra la trasmissione del segnale e quando il Rover lo riceve, ma grazie all'alta precisione dei ricevitori GNSS, questo può avvenire quasi istantaneamente.

Il Rover determina la posizione utilizzando algoritmi che incorporano l'ambiguity resolution e la correzione differenziale della fase. La precisione dipende anche dalla sua distanza dalla stazione base (indicata come "linea di base") e dall'accuratezza delle correzioni differenziali.¹⁸

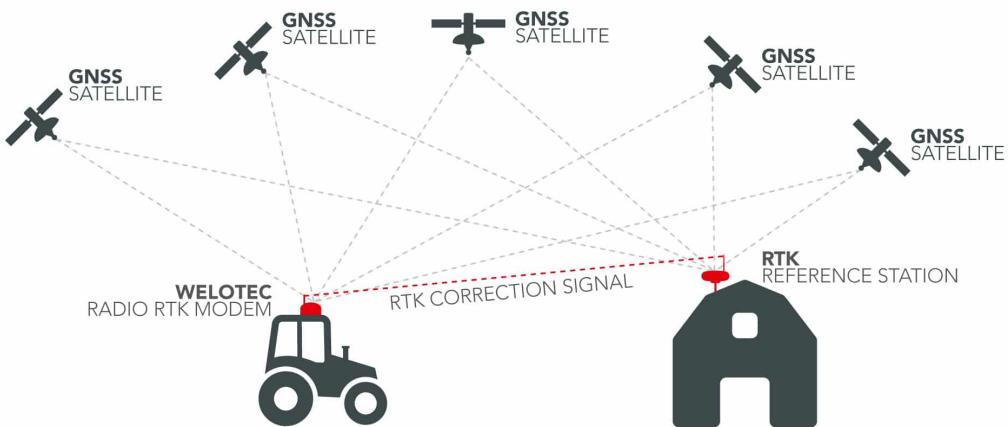


Figura 1.8: Funzionamento Real-Time Kinematic

1.2.5 Networked Transport of RTCM via Internet Protocol

Per eseguire un accurato sondaggio RTK è necessario un rover e una base che trasmette le correzioni in tempo reale, come descritto in precedenza. Si può utilizzare la propria base o una remota utilizzando una tecnologia chiamata NTRIP.¹⁹

Il Networked Transport of RTCM via Internet Protocol (NTRIP) è un sistema che è stato sviluppato dall'agenzia federale tedesca per cartografia e geodesia nel 2004, e permette ad un rover di accettare correzioni da Internet senza necessità di un secondo ricevitore locale che agisca come base.

NTRIP include tre componenti principali: base, caster e il rover del client.²⁰

- La base, chiamata Continuously Operating Reference Stations (CORS), è quella più vicina al client nella rete di basi messe a disposizione dal sistema NTRIP che si sta utilizzando.
- Il caster rappresenta il server del sistema, è responsabile della ricezione del flusso di dati dal ricevitore di base e di reindirizzarli su una porta TCP (Transmission Control Protocol) specificata.
- Il rover del client che è il ricevitore che il client utilizza per ricevere i segnali.

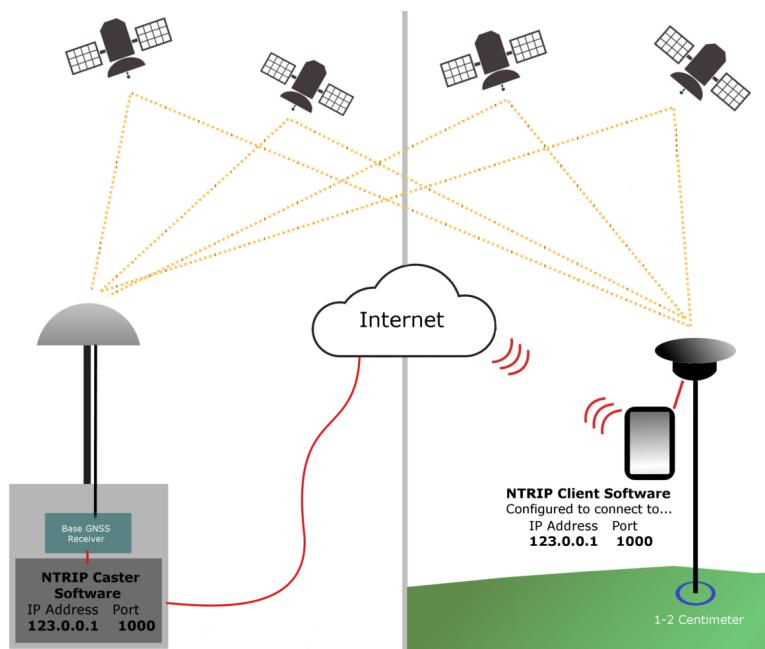


Figura 1.9: Funzionamento NTRIP

Capitolo 2

Sperimentazione RSSI

Il chip Bluetooth presente negli smartphone è in grado di ricevere e trasmettere segnali, ma è anche in grado di stabilire con che potenza un segnale viene ricevuto. Il Received Signal Strength Indicator (RSSI) è la misura della potenza ricevuta e, siccome è sempre molto bassa (nell'ordine dei mW), è misurata in dBm, secondo la seguente formula:

$$P_{dBm} = 10 \cdot \log_{10}(P_W \cdot 10^3)$$

Dove P_W è la potenza espressa in Watt.

2.1 Introduzione allo studio svolto

L'obbiettivo della sperimentazione Bluetooth è stato quello di trovare un modello che fosse in grado di associare ad un certo range di valori RSSI una determinata distanza tra due smartphone. Essendo consapevoli del limitato raggio d'azione di questa tecnologia, le misure sono state concentrate in un sottoinsieme di distanze tra 0.125 e 4 metri, analizzando anche come cambia la potenza con l'introduzione di ostacoli.

L'accuratezza del modello trovato sarà poi confrontata con quella del modello prodotto successivamente tramite la tecnica RTK (Capitolo 3). Questo ci permetterà di capire quale tra i due metodi è migliore per valutare la distanza che separa i due dispositivi e se è possibile combinarli per ottenere una soluzione più efficiente.

Gli strumenti utilizzati sono due smartphone Android, nello specifico uno Xiaomi Mi 8 e un Huawei Mate 20. Su quest'ultimo è stata installata l'applicazione chiamata *Bluetooth Finder* realizzata appositamente ai fini della sperimentazione. Il suo funzionamento prevede due schermate: nella prima (Fig. 2.1) l'utente deve selezionare il dispositivo di cui vuole conoscere l'RSSI

tra una lista di dispositivi, nella seconda (Fig. 2.2) vengono registrati i valori e salvati automaticamente in un file excel in memoria.

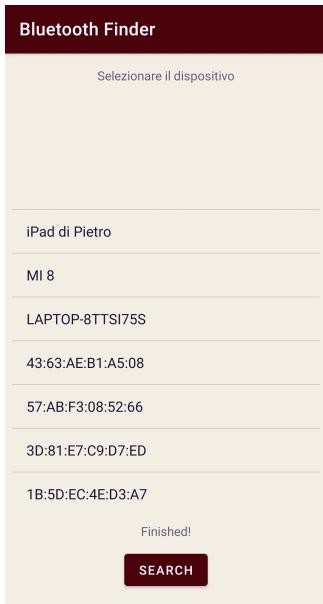


Figura 2.1: Schermata in cui selezionare il dispositivo

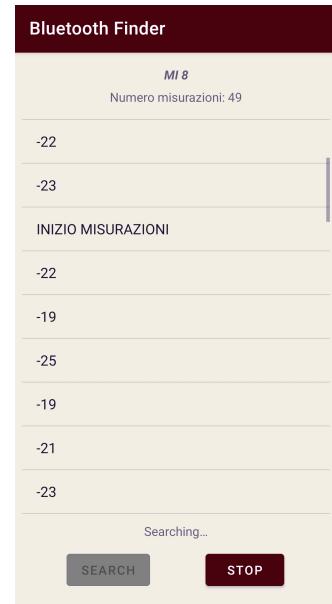


Figura 2.2: Registrazione dei valori RSSI

2.2 Misurazioni

Le misure sono state prese indoor sistemando i due dispositivi a una distanza stabilita e lasciando ogni volta che l'applicazione registrasse 500 valori di RSSI, in alcune situazioni le misure sono state ripetute per ottenere più campioni da elaborare (Sezione 2.5).

Dai dati esposti nelle Fig. 2.3 e Fig. 2.4 si può notare come all'aumentare della distanza i valori dell'RSSI siano distribuiti in un range maggiore. In particolare è interessante sottolineare come per valori inferiori ai 2 metri la funzione di densità di probabilità discreta che meglio rappresenta la distribuzione dei valori assunti dalla variabile aleatoria (anch'essa discreta) sia quella binomiale (Fig. 2.5). Mentre all'aumentare della distanza, e di conseguenza anche del rumore, i valori si appiattiscono facendo assomigliare la funzione ad una discreta uniforme (Fig. 2.6).

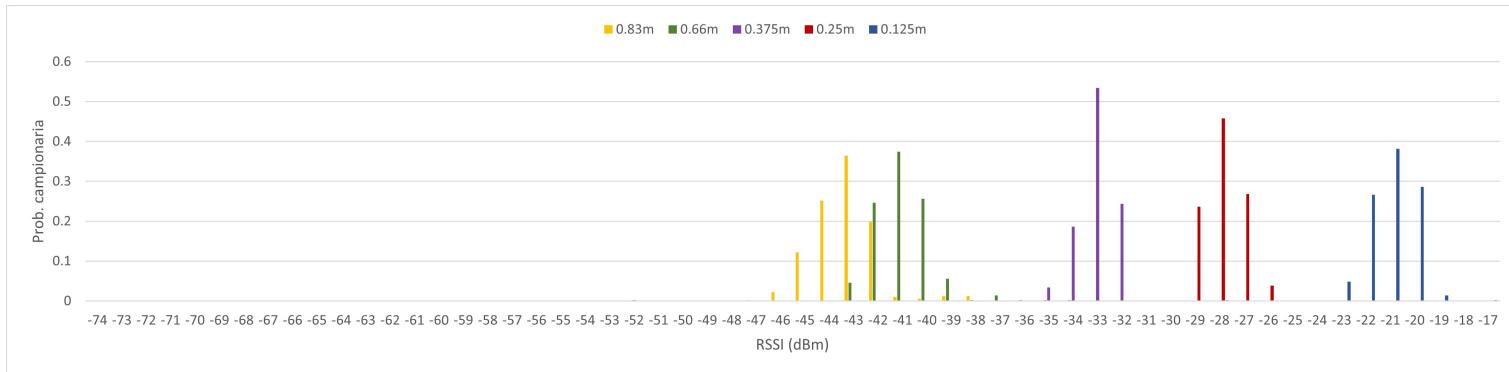


Figura 2.3

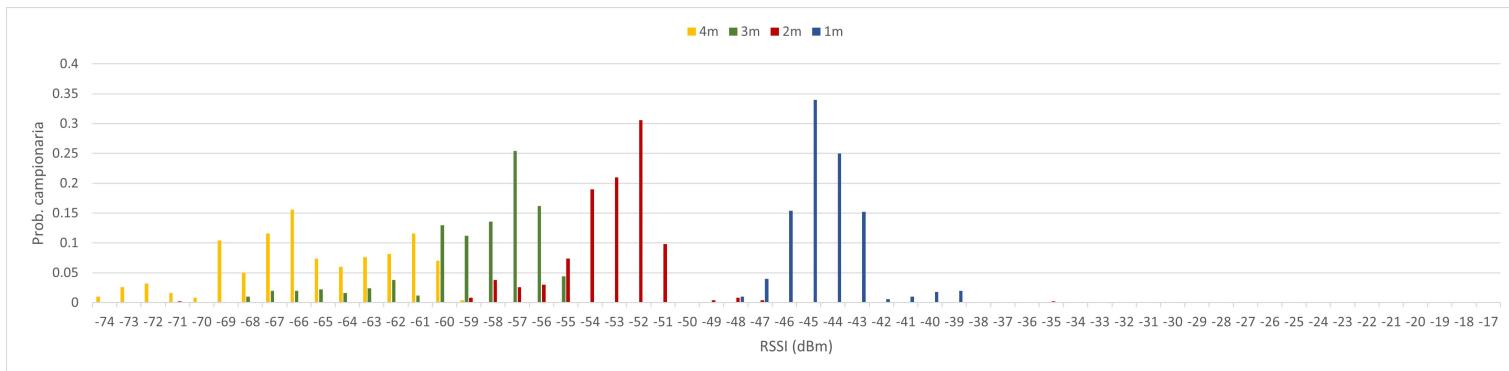


Figura 2.4

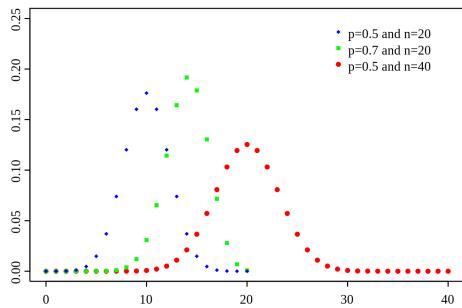


Figura 2.5: Funzione di distribuzione discreta binomiale

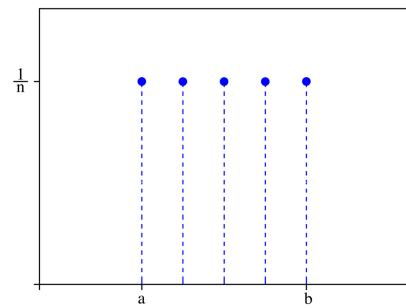


Figura 2.6: Funzione di distribuzione discreta uniforme

2.3 Range RSSI - distanza

Con i dati raccolti è stato possibile associare una relazione tra un range di valori RSSI e la relativa distanza. Per il discorso fatto in precedenza più la distanza aumenta più il range in cui sono distribuiti i valori aumenta. È stato svolto quindi uno studio considerando come criterio di selezione l'intervallo che contiene almeno il 90% dei dati raccolti attorno al valore medio ad ogni distanza, i risultati sono mostrati nella Fig. 2.7 e nella Tab. 2.1.

Distanza	Range RSSI (dBm)	Affidabilità
0.125m	-20/-22	93.4%
0.25m	-27/-29	96.2%
0.375m	-32/-34	96.4%
0.66m	-39/-42	93.2%
0.83m	-42/-45	94.6
1m	-43/-46	90.6%
2m	-51/-56	90.8%
3m	-55/-63	91.2%
4m	-61/-72	90.2%

Tabella 2.1

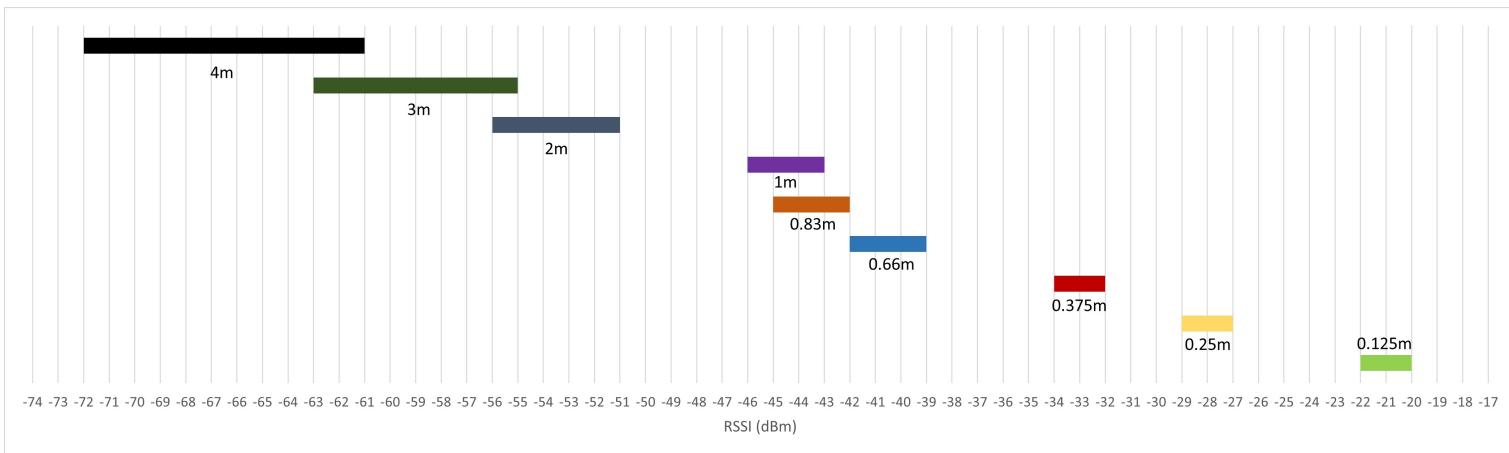


Figura 2.7

2.4 Attenuazione del modello radio

Assumendo che P_{Tx} sia la potenza del segnale trasmesso da una antenna isotropa ideale, che si irradia in modo uniforme in tutte le direzioni nello spazio. A distanza d dalla antenna, la densità di potenza sarà:²¹

$$\Phi = \frac{P_{Tx}}{4\pi d^2}$$

dove $4\pi d^2$ è la superficie di una sfera di raggio d che è uniformemente irradiata dall'antenna. Si osserva che la densità di potenza diminuisce con il quadrato della distanza, che su scala logaritmica (dB) equivale ad una diminuzione di 20 dB per decade.

All'antenna di ricezione, la potenza disponibile in condizione di impedenza abbinata è pari a:

$$P_{Rc} = \Phi \cdot A_{Ant,Rc} \cdot \eta_{Ant,Rc}$$

dove P_{Rc} è la potenza ricevuta, $A_{Ant,Rc}$ è l'area effettiva di ricezione della antenna e $\eta_{Ant,Rc}$ è l'efficienza dell'antenna di ricezione. Il fattore $\eta_{Ant,Rc} < 1$ tiene conto del fatto che l'antenna non cattura tutti i raggi incidenti, questo perché una parte viene riflessa o persa. Per concludere, la potenza del segnale ricevuto è data da:

$$P_{Rc} = \frac{P_{Tx} \cdot A_{Ant,Rc} \cdot \eta_{Ant,Rc}}{4\pi d^2}$$

Trasformando ora la formula in dB si ottiene:

$$\begin{aligned} P_{Rc(dBW)} &= 10 \log_{10} P_{Rc} \\ &= 10 [\log_{10} (P_{Tx} \cdot A_{Ant,Rc} \cdot \eta_{Ant,Rc}) - \log_{10} (4\pi d^2)] \\ &= 10 [\log_{10} (P_{Tx} \cdot A_{Ant,Rc} \cdot \eta_{Ant,Rc}) - \log_{10} (4\pi) - \log_{10} (d^2)] \end{aligned}$$

Assumendo che i parametri P_{Tx} , $A_{Ant,Rc}$, $\eta_{Ant,Rc}$ siano costanti, possiamo raggruppare tutte le costanti sotto K , ottenendo:

$$P_{Rc(dBW)} = K - 20 \log_{10} (d)$$

Come si può notare l'attenuazione della potenza è legata alla distanza con un andamento logaritmico.

È stato possibile verificare questo andamento sperimentalmente come mostrato nella Fig. 2.8. Ogni punto rosso corrisponde alla media di 500 valori RSSI raccolti alla relativa distanza.

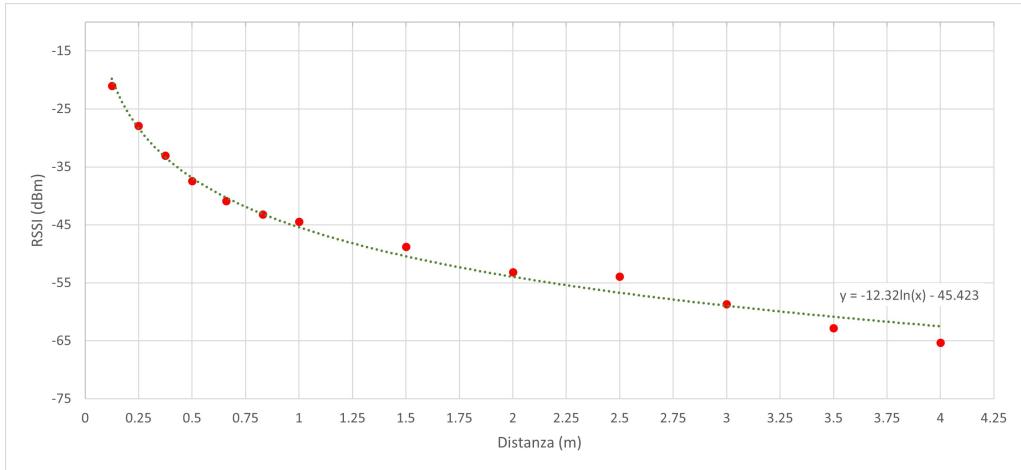


Figura 2.8: RSSI raccolti alle relative distanze

Per ottenere la linea di tendenza che meglio rappresentasse l'andamento dei dati all'aumentare della distanza è stato deciso di utilizzare la tecnica dei minimi quadrati.

L'equazione cercata è di tipo:

$$y = a \ln(x) + b$$

Nelle prossime espressioni verranno usati x, y per i valori della funzione cercata, mentre x_i, y_i per i valori ottenuti dalla sperimentazione. Si può notare che le distanze sono uguali quindi $x = x_i$, mentre in generale $y \neq y_i$. Definiamo:

$$\xi_i = |y_i - y| = |y_i - a \ln(x) - b|$$

Il metodo utilizzato consiste nel minimizzare la somma S :

$$S = \sum_{i=1}^n \xi_i^2 = \sum_{i=1}^n (y_i - a \ln(x) - b)^2$$

dove n è il numero di distanze considerate. Bisogna quindi porre:

$$\frac{dS}{da} = \frac{dS}{db} = 0$$

$$\begin{aligned} \frac{dS}{da} &= \sum_{i=1}^n 2(y_i - a \ln(x_i) - b)(-\ln(x_i)) \\ \frac{dS}{db} &= \sum_{i=1}^n 2(y_i - a \ln(x_i) - b)(-1) \end{aligned}$$

$$\begin{cases} -2 \sum_{i=1}^n (y_i - a \ln(x_i) - b) (\ln(x_i)) = 0 \\ -2 \sum_{i=1}^n (y_i - a \ln(x_i) - b) = 0 \\ \sum_{i=1}^n (\ln(x_i) \cdot y_i) - a \sum_{i=1}^n (\ln(x_i))^2 - b \sum_{i=1}^n (\ln(x_i)) = 0 \\ \sum_{i=1}^n (y_i) - a \sum_{i=1}^n (\ln(x_i)) - b \sum_{i=1}^n 1 = 0 \end{cases}$$

Sostituendo coi dati si ottiene:

$$\begin{cases} a = -12.3183 \\ b = -45.4233 \end{cases}$$

Da cui:

$$y = -12.32 \ln(x) - 45.423$$

2.5 Anomalia 0.5m

A 0.5m si può notare una strana distribuzione dei dati, per questo motivo è stato deciso di studiare più approfonditamente la situazione raccogliendo ulteriori dati (fino a 1600 RSSI circa), ed è stato osservato che il valore atteso dalla funzione trovata nella sezione precedente (Fig. 2.8) di -37dBm non viene praticamente mai registrato, insieme al valore -38dBm (Fig. 2.9). Avendo raccolto così tanti dati possiamo considerare questo comportamento un malfunzionamento del ricevitore e per questo motivo ai fini dello studio è stato deciso di scartare questa distanza. Interessante sottolineare come in mancanza dei due valori centrali (-37dBm e -38dBm) i dati siano comunque equidistribuiti ai bordi e quindi la loro media generi un valore in linea con la linea di tendenza.

2.6 Analisi dati con ostacoli

Durante la sperimentazione sono stati raccolti dati in situazioni diverse per osservare il comportamento RSSI.

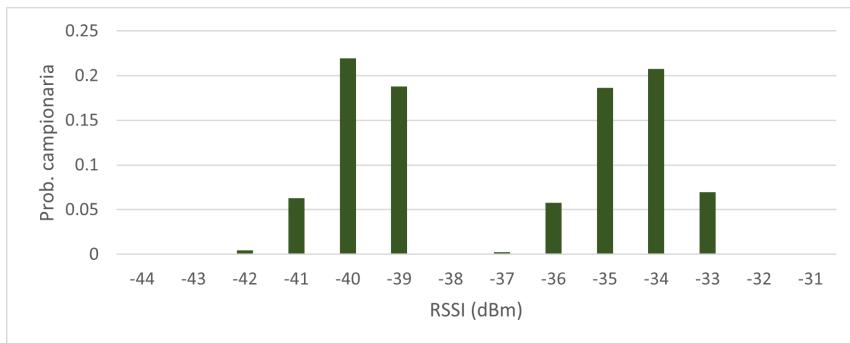


Figura 2.9: RSSI raccolti a 0.5m

Un primo sperimento è stato svolto disponendo i due telefoni girati al contrario ad 1m; il risultato (Fig. 2.10) è stato che la distribuzione dei dati è molto simile alla situazione coi telefoni rivolti uno verso l’altro (Fig. 2.11). Un risultato che ci si poteva aspettare in quanto l’orientamento del dispositivo poco influisce sulla ricezione dei dati.

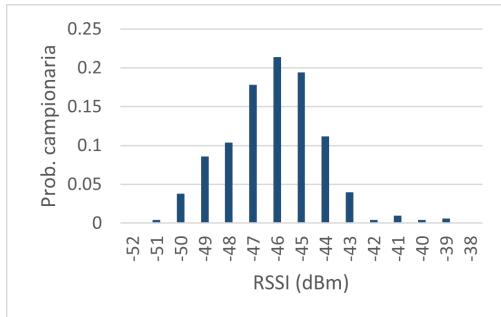


Figura 2.10: Telefoni girati

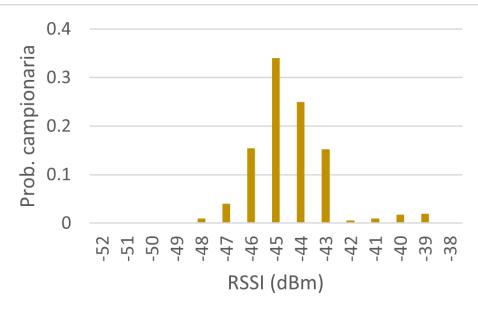


Figura 2.11: Telefoni normali

Successivamente ci si è concentrati sullo studio di ostacoli, in particolare con una persona in mezzo ai due dispositivi che rappresenta l’ostacolo più probabile in una situazione di utilizzo quotidiano.

Come mostrano i dati raccolti (Fig. 2.12, Fig. 2.13 e Fig. 2.14) la potenza del segnale Bluetooth si attenua e i valori di RSSI vengono distribuiti su un intervallo più ampio. Indipendentemente dalla distanza sembrerebbe che il range di valori sia sempre lo stesso, questo probabilmente è dovuto al percorso del segnale, che non essendo più diretto, rimbalza sempre sulla stessa parete per poi arrivare al ricevitore. In questo modo il percorso del segnale aumenta e, siccome la potenza del segnale e la distanza sono legate da una funzione logaritmica, su distanze maggiori i valori di RSSI sono molto simili.

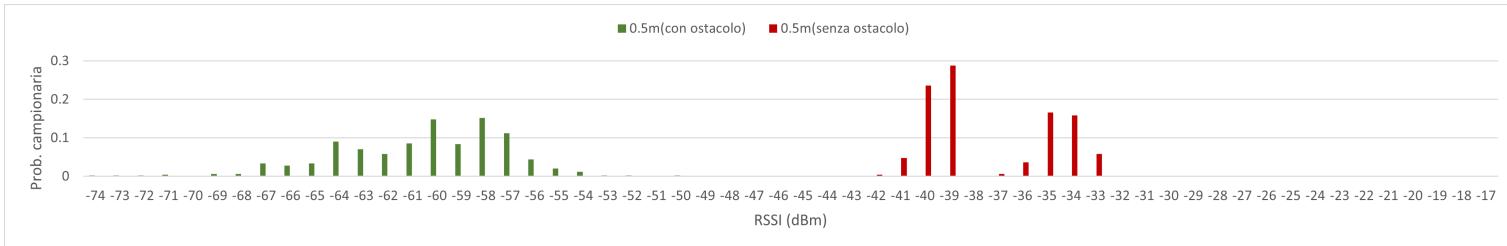


Figura 2.12

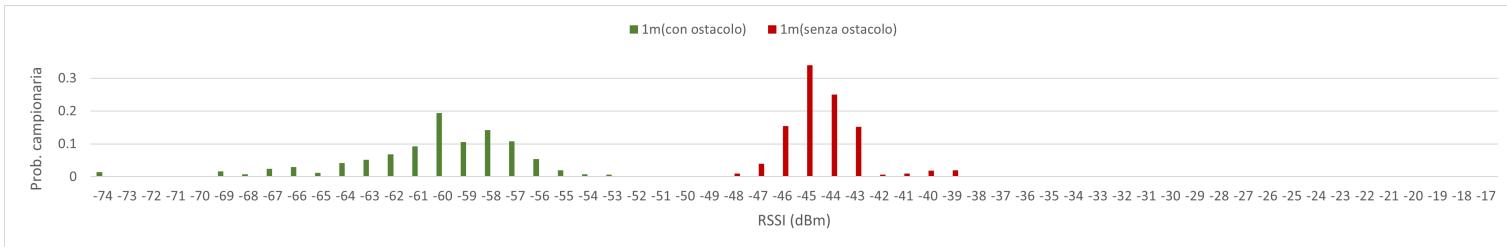


Figura 2.13

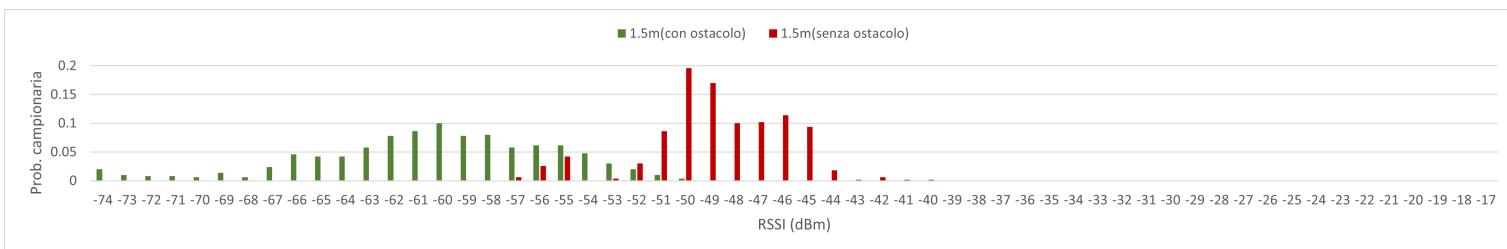


Figura 2.14

2.7 Conclusioni RSSI

Analizzando i dati ottenuti sembrerebbe una buona soluzione poter calcolare la distanza tra due smartphones partendo dalla semplice misura dell’RSSI del segnale che uno smartphone riceve dall’altro. Tuttavia purtroppo non è così. Come peraltro evidenziato anche dallo stesso consorzio che standardizza il Bluetooth²² numerosi fattori impediscono di effettuare una stima di distanza precisa ed in tempo reale utilizzando solamente l’RSSI del segnale, tra l’altro unico valore misurabile su smartphone Android e iOS.

Oltre al ristretto raggio d’azione, un altro grande limite è legato alla impossibilità di trovare un modello universale da poter utilizzare per tutti i dispositivi. Infatti due dispositivi di marca e modello diversi, posizionati a una distanza predefinita che rimane costante durante la misurazione, non ricevono reciprocamente un segnale con lo stesso valore di RSSI.

Una possibile soluzione, anche se piuttosto laboriosa, prevede di testare i due dispositivi alle relative distanze prima di utilizzarli, in modo da poter individuare i vari range di RSSI-distanza. Questo è molto importante perché, considerato legame logaritmico che lega l’RSSI e la distanza, piccole variazioni della misura dell’RSSI possono provocare errori rilevanti nella stima della distanza.

Capitolo 3

Sperimentazione RTK

Le misure con tecnica RTK sono state svolte in maniera simile a quanto fatto con le misure RSSI: i due dispositivi sono stati posti a una determinata distanza e al posto della potenza Bluetooth, sono stati registrati dati GNSS. Già da qualche anno Android fornisce accesso all'interno del suo framework alle misure GNSS grezze su diversi dispositivi che lo supportano: questo aspetto è stato fondamentale per svolgere la sperimentazione.

3.1 Strumenti usati

Il primo passo è stato utilizzare *Geo++ RINEX Logger* (Fig. 3.1), un'applicazione già presente sul Play Store che cerca i satelliti visibili e, dato il via alle registrazioni dei dati, li salva all'interno del dispositivo in due cartelle, una contenente le misure migliori e l'altra contenente tutte le misure.

L'applicazione immagazzina i dati sotto forma di Receiver Independent Exchange Format (RINEX), un formato per registrare dati grezzi del sistema di navigazione satellitare (Fig. 3.2). Questo ci consentirà di post-elaborare i dati ricevuti per produrre un risultato più accurato, con altri dati sconosciuti al ricevitore originale.

Per l'elaborazione dei file RINEX prodotti dall'applicazione è stata utilizzata la libreria open source RTKLIB disponibile per Windows e Linux, sviluppato da Tomoji Takasu dell'università di Tokyo (<http://www.rtklib.com/>). RTKLIB è un pacchetto di programmi open source per posizionamento standard e di precisione che viene utilizzato per il post-processing, in particolare per unire informazioni da più file RINEX ed ottenere una posizione finale precisa.

3.1. STRUMENTI USATI CAPITOLO 3. Sperimentazione RTK

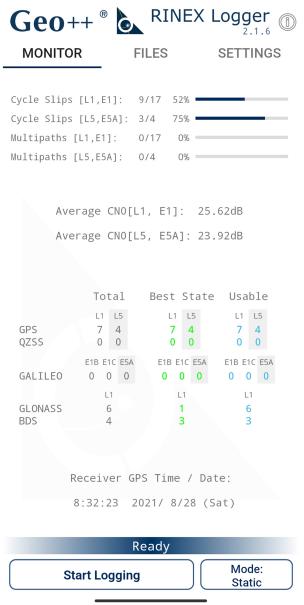


Figura 3.1: Schermata *Geo++ RINEX Logger*

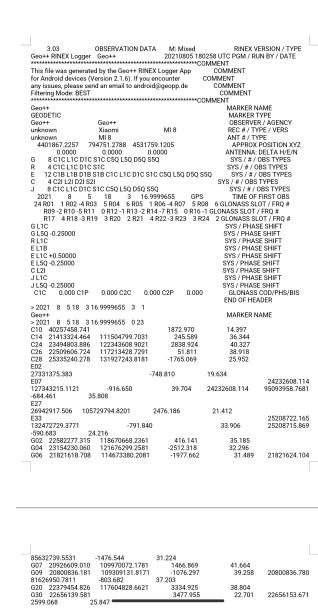


Figura 3.2: Esempio di un file RINEX

Le applicazioni racchiuse all'interno della stessa libreria sono svariate ma quelle che sono state adoperate ai fini delle sperimentazioni sono RTKPOST (Fig. 3.3) e RTKPLOT.

RTKPOST necessita in input il file RINEX del dispositivo Rover di cui si vuole calcolare la posizione, quindi nel nostro caso quello generato o da Xiaomi Mi 8 o da Huawei Mate 20, come secondo input il file RINEX della Base Station e infine come terzo file di ingresso quello navigazionale. Gli ultimi due file dipendono da diversi fattori (condizioni meteo, disposizione dei satelliti, ecc.) e cambiano di giornata in giornata, sono stati reperiti dal Servizio di Posizionamento Interregionale GNSS (SPIN3 GNSS) (<http://www.spingnss.it/spiderweb/frmIndex.aspx>).

RTKPOST restituisce in output un file posizionale con estensione .pos che contiene un ID seguito da una o più posizioni, in longitudine e latitudine, che gli sono associate.

Le misure sono state prese in un parco a Brescia, lontano da ostacoli, a circa 100m dalla Base Station BREU, situata nel Dipartimento di Ingegneria dell'Informazione dell'università di Brescia.

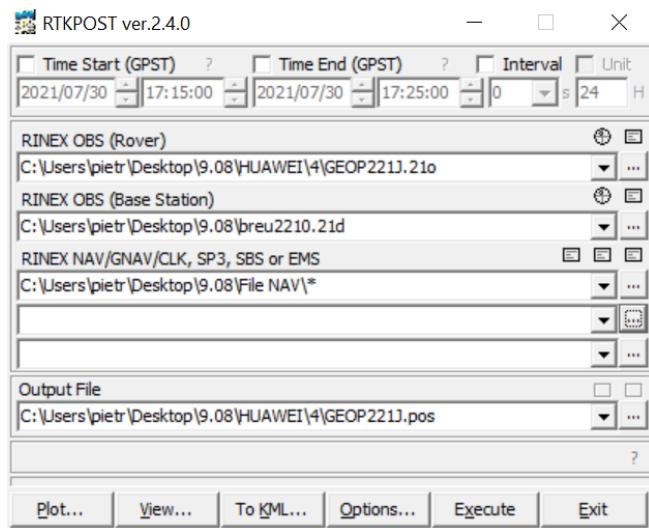


Figura 3.3: RTKPOST

3.2 Analisi dati

Dopo aver raccolto i dati e averli elaborati, c'è stata la fase di analisi. Utilizzando un altro programma appartenente alla libreria RTKLIB chiamato RTKPLT è stato possibile visualizzare graficamente i punti geografici generati dall'esecuzione di RTKPOST.

Nei punti geografici raccolti mantenendo il telefono nella stessa posizione si può notare quasi sempre un raggruppamento di punti in una zona (Fig. 3.4). Poiché si può osservare come le aree di clustering coincidono molto spesso con i punti finali della misurazione, è stato deciso di mantenere solo gli ultimi 50 punti di ogni misurazione (Fig. 3.5), in questo modo sono stati ottenuti dei risultati più precisi e con un margine di errore accettabile.

È stato creato uno script Java che fosse in grado di leggere le coordinate nel file e farne una media, in modo da ottenere il punto geografico che meglio rappresentasse il file posizionale.

Successivamente lo script è stato ampliato in modo che l'operazione del calcolo della coordinata media venisse effettuato su due file posizionali (XIAO-MI.pos e HUAWEI.pos) e venisse calcolata la distanza tra i due punti geografici, seguendo quanto descritto nella Sez. 3.3.

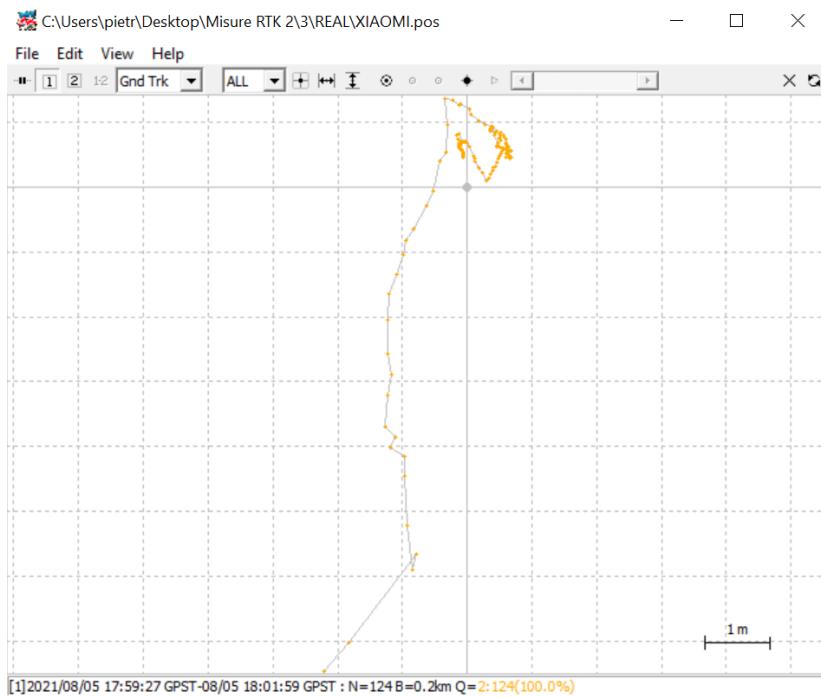


Figura 3.4: File posizionale completo di Xiaomi Mi 8

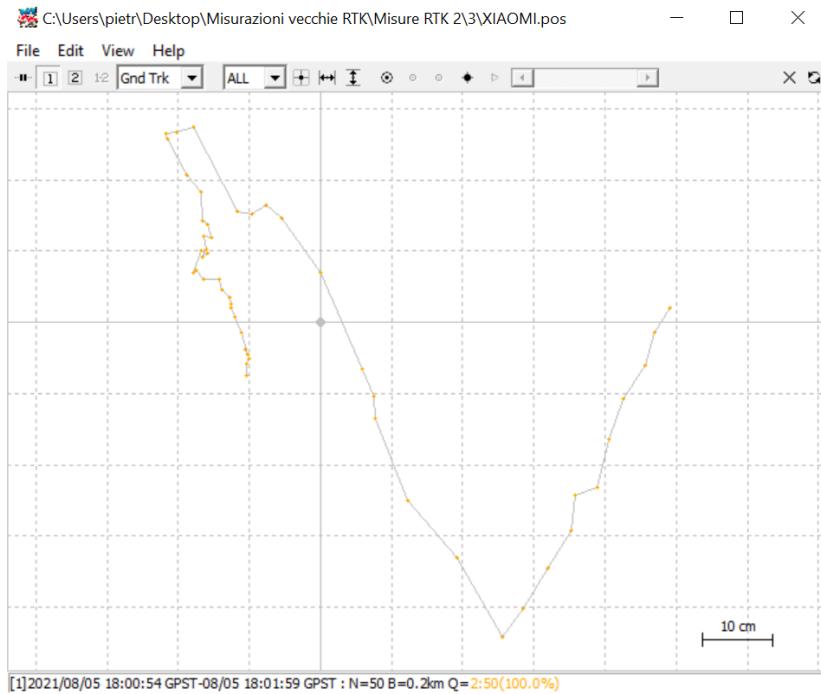


Figura 3.5: Ultimi 50 punti del file posizionale di Xiaomi Mi 8

```

1 import java.io.*;
2
3 class Distanza{
4
5     /*MAIN*/
6     public static void main(String[] args) throws
7         FileNotFoundException, IOException{
8         double[] XIAOMI = coordinata_media("XIAOMI.pos");
9         double[] HUAWEI = coordinata_media("HUAWEI.pos");
10        System.out.println("La distanza e': " + disgeod(
11            XIAOMI[0], XIAOMI[1], HUAWEI[0], HUAWEI[1]) +
12            "m");
13    }
14
15    /*METODO CHE LEGGE LE COORDINATE DEL FILE POSIZIONALE E
16       NE TROVA LA MEDIA IN LATITUDINE E LONGITUDINE*/
17    public static double[] coordinata_media(String nomeFile)
18        throws FileNotFoundException, IOException{
19
20        /*operazioni per leggere il file*/
21        BufferedReader reader = new BufferedReader(new
22            FileReader(nomeFile));
23        String line = reader.readLine();
24
25        /*operazione per contare il numero di righe*/
26        int numLines = countLines(nomeFile);
27
28        /*for per scartare le prime 25 righe del file che
29           contengono altre informazioni*/
30        for(int i = 1; (line!=null) && (i<25) ; i++){
31            line = reader.readLine();
32        }
33
34        int num = 25; /*numero di riga*/
35        double finalLat = 0;
36        double finalLon = 0;
37
38        while(line!=null) { /*legge tutte le righe*/
39            num++;
40            if(num>numLines-49){ /*controlla se la riga e'
41                nelle ultime 50*/
42                double lat =
43                    Double.parseDouble(line.substring(26,38));
44                /*parte della riga in cui e' scritta la
45                   latitudine*/
46
47                finalLat += lat;
48                finalLon += Double.parseDouble(line.substring(48,50));
49
50            }
51        }
52
53        finalLat = finalLat / (double) num;
54        finalLon = finalLon / (double) num;
55
56        double[] result = {finalLat, finalLon};
57
58        return result;
59    }
60
61    /*metodo per contare il numero di righe in un file*/
62    public static int countLines(String nomeFile){
63        int numLines = 0;
64
65        try {
66            BufferedReader reader = new BufferedReader(new
67                FileReader(nomeFile));
68            String line;
69            while((line = reader.readLine()) != null) {
70                numLines++;
71            }
72        } catch (IOException e) {
73            e.printStackTrace();
74        }
75
76        return numLines;
77    }
78
79    /*metodo per calcolare la distanza tra due coordinate
80       utilizzando la formula di Haversine*/
81    public static double disgeod(double lat1, double lon1,
82        double lat2, double lon2) {
83
84        final double R = 6371000; //Earth radius in meters
85
86        final double phi1 = Math.toRadians(lat1);
87        final double phi2 = Math.toRadians(lat2);
88
89        final double dLat = Math.toRadians(lat2 - lat1);
90        final double dLon = Math.toRadians(lon2 - lon1);
91
92        final double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
93            Math.cos(phi1) * Math.cos(phi2) *
94            Math.sin(dLon/2) * Math.sin(dLon/2);
95
96        final double c = 2 * Math.atan2(Math.sqrt(a),
97            Math.sqrt(1-a));
98
99        final double distance = R * c;
100
101        return distance;
102    }
103}

```

```

35         double lon =
36             Double.parseDouble(line.substring(41,53));
37             /*parte della riga in cui e' scritta la
38             longitudine*/
39             finalLat = finalLat + lat; /*somma tutte le
40             latitudini*/
41             finalLon = finalLon + lon; /*somma tutte le
42             longitudini*/
43         }
44         line = reader.readLine();
45     }
46
47     double[] result = new double[2]; /*crea l'array dove
48     salvare il risultato*/
49
50     result[0] = finalLat = finalLat/50; /*divide
51     latitudine per 50*/
52     result[1] = finalLon = finalLon/50; /*divide
53     longitudine per 50*/
54
55     return result;
56 }
57
58 /*METODO CHE CALCOLA LA DISTANZA GEODETICA*/
59 public static double disgeod (double latA, double lonA,
60     double latB, double lonB){
61
62     /*definisce le costanti e le variabili */
63     final double R = 6373.044737;
64     final double pigreco = 3.1415927;
65     double lat_alfa, lat_beta;
66     double lon_alfa, lon_beta;
67     double fi;
68     double p, d;
69
70     /*converte i gradi in radianti */
71     lat_alfa = pigreco * latA / 180;
72     lat_beta = pigreco * latB / 180;
73     lon_alfa = pigreco * lonA / 180;
74     lon_beta = pigreco * lonB / 180;
75
76     fi = Math.abs(lon_alfa - lon_beta); /*calcola l'angolo
77     compreso fi */
78     p = Math.acos(Math.sin(lat_beta) *
79                 Math.sin(lat_alfa) + Math.cos(lat_beta) *
80                 Math.cos(lat_alfa));
81
82     d = R * p;
83
84     return d;
85 }
```

```

        Math.cos(lat_alfa) * Math.cos(fi)); /*calcola
        il terzo lato del triangolo sferico */
69     d = p * R; /*calcola la distanza sulla superficie
        terrestre R = ~6371 km */
70
71     return(d*1000); /*trasforma il risultato in metri*/
72 }
73
74 /*METODO CHE CALCOLA IL NUMERO DI RIGHE IN UN FILE*/
75 public static int countLines(String filename) throws
    IOException {
76     InputStream is = new BufferedInputStream(new
        FileInputStream(filename));
77     try {
78         byte[] c = new byte[1024];
79         int count = 0;
80         int readChars = 0;
81         boolean empty = true;
82         while ((readChars = is.read(c)) != -1) {
83             empty = false;
84             for (int i = 0; i < readChars; ++i) {
85                 if (c[i] == '\n') {
86                     ++count;
87                 }
88             }
89         }
90         return (count == 0 && !empty) ? 1 : count;
91     } finally {
92         is.close();
93     }
94 }
95 }
```

Listing 3.1: Script Java

3.3 Distanza geodetica

Per calcolare la distanza tra due coordinate geografiche è stata considerata la terra perfettamente sferica, anche se in realtà non lo sarebbe. Così facendo la precisione non è massima ma sufficiente allo scopo della ricerca, il problema ora si riduce quindi a come calcolare la distanza tra due punti su una superficie sferica.²³

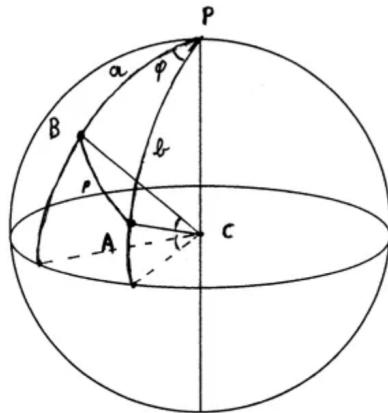


Figura 3.6: Sfera esemplificativa

Osservando la Fig. 3.6 si può osservare che, in base alla trigonometria sferica (teorema di Eulero), tra i lati a , b e p del triangolo sferico ABP vale la relazione:

$$\cos p = \cos a \cdot \cos b + \sin a \cdot \sin b \cdot \cos \varphi$$

Dette $\text{lat}(A)$, $\text{lon}(A)$, $\text{lat}(B)$, $\text{lon}(B)$ la latitudine e la longitudine dei punti A e B si può affermare che:

$$\begin{aligned} a &= 90^\circ - \text{lat}(B) \\ b &= 90^\circ - \text{lat}(A) \\ \varphi &= \text{lon}(A) - \text{lon}(B) \end{aligned}$$

Sostituendo si ottiene:

$$\begin{aligned} \cos p &= \cos(90^\circ - \text{lat}(B)) \cdot \cos(90^\circ - \text{lat}(A)) + \sin(90^\circ - \text{lat}(B)) \cdot \\ &\quad \cdot \sin(90^\circ - \text{lat}(A)) \cdot \cos(\text{lon}(A) - \text{lon}(B)) \\ &= \sin(\text{lat}(B)) \cdot \sin(\text{lat}(A)) + \cos(\text{lat}(B)) \cdot \cos(\text{lat}(A)) \cdot \\ &\quad \cdot \cos(\text{lon}(A) - \text{lon}(B)) \end{aligned}$$

Infine, la distanza d in metri si ottiene:

$$d = p \cdot R \cdot 10^3$$

dove $R \sim 6371\text{km}$ è il raggio della terra.

3.4 Misurazioni

Sono state prese 10 misurazioni per ogni distanza, in cui ogni file posizionale ha più di 100 posizioni geografiche salvate (che corrispondono a circa 3 minuti di registrazione dati) di cui però ne vengono considerate solo le ultime 50 come già accennato. Come mostra la Fig. 3.7 ad ogni distanza si può notare una distribuzione che si concentra in un intorno di quella reale, con solo qualche distanza errata.

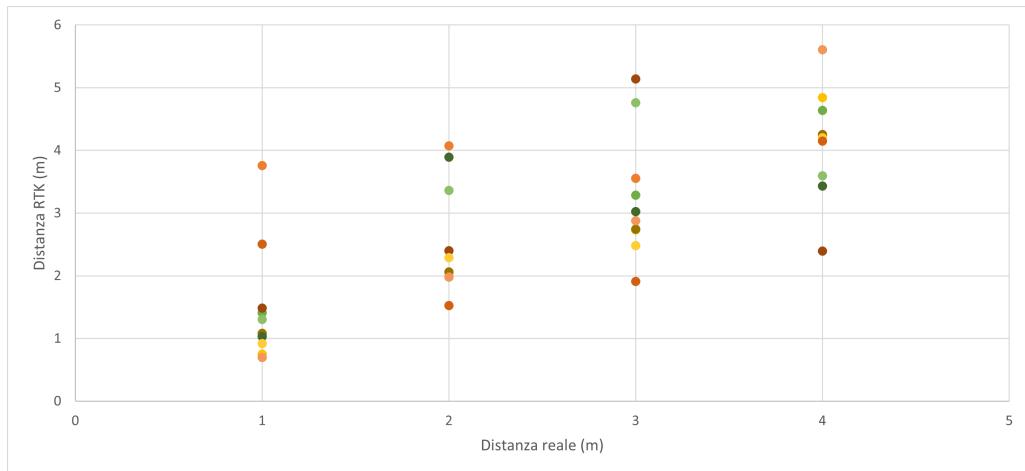


Figura 3.7: Misure RTK raccolte

Come si può notare dalle Fig. 3.8 che rappresenta le medie dei punti alle varie distanze, l'andamento rimane abbastanza lineare, che è quello che ci si aspetta perché le due distanze sono legate linearmente (a meno di un offset che in questo caso risulta di 0.7067).

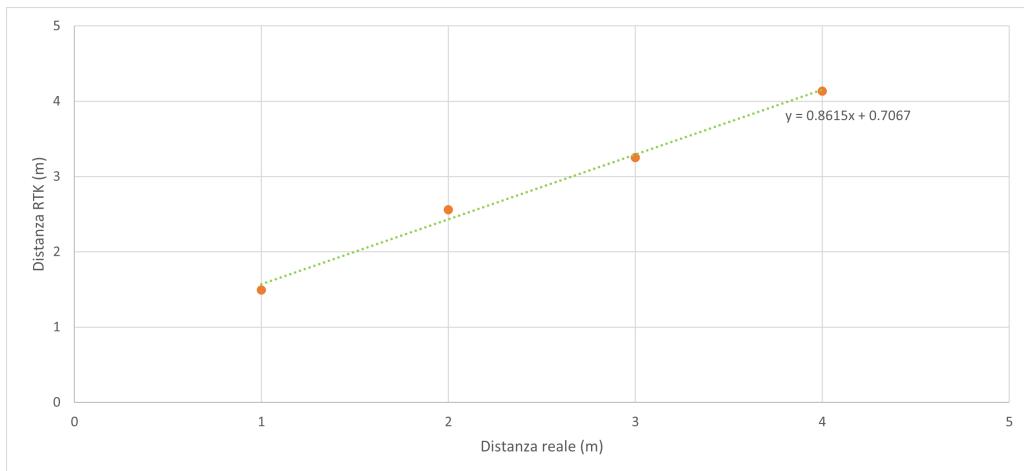


Figura 3.8: Media delle misure RTK raccolte

3.5 Conclusioni RTK

Inizialmente i risultati aspettati con la tecnica RTK erano di una maggiore precisione, infatti diversi articoli parlano di precisione a livello centimetrico¹⁷ che non siamo riusciti ad ottenere. Probabilmente però, risultati con una così alta accuratezza necessitano di ricevitori GNSS qualitativamente superiori a quelli inseriti negli smartphone.

Nonostante ciò è stato comunque possibile individuare un raggruppamento di dati nei file posizionali, che ha permesso di ottenere un'accuratezza media inferiore al metro, che risulta comunque accettabile e quindi sufficiente per essere utilizzata nel confronto con RSSI.

È importante sottolineare che i dati sono stati raccolti in condizioni ottimali, in un parco senza ostacoli, il che ha permesso una maggiore precisione. Inoltre nonostante ciò, alcune misure sono state riprese perché non sufficientemente precise (probabilmente per condizioni meteo non ideali). Questi esempi servono a dimostrare quanto l'RTK sia uno strumento che ha ancora parecchi limiti con i ricevitori GNSS degli smartphone attuali e di conseguenza non troppo affidabile; questo non esclude che in un futuro, con il miglioramento dell'hardware, questa tecnica possa diventare più efficace.

Capitolo 4

Confronto sperimentazioni

La parte conclusiva di questa tesi ha come obiettivo l'unione delle due sperimentazioni effettuate per trovare un algoritmo in grado di stabilire se il telefono sotto esame sia sottoposto ad un attacco di spoofing GPS oppure no.

La grande differenza che si può notare nei dati raccolti durante le due sperimentazioni è che mentre per l'RTK l'errore sulla misura è indipendente dalla distanza, per il Bluetooth non è altrettanto: all'aumentare della distanza le misure risultano più incerte. È risultato quindi necessario creare un modello che dipendesse dalla lontananza e proprio al variare di questa capisca quanta affidabilità dare a una tecnica rispetto che all'altra.

Per questa sezione ci si è voluti concentrare su quattro distanze: 1m, 2m, 3m e 4m.

4.1 L'algoritmo

La soluzione migliore trovata consiste nel calcolare:

$$x_m = a \cdot x_{BLT} + b \cdot x_{RTK}$$

dove x_{BLT} corrisponde alla distanza misurata dal Bluetooth e x_{RTK} a quella misurata dall'RTK. a e b sono dei coefficienti che variano in base alla distanza stessa.

Trovata x_m , che risulta essere quindi la miglior stima della distanza, sia $var[x_{RTK}]_m$ la varianza media delle misure ottenute in fase di sperimentazione con l'RTK; si procede confrontando la probabilità che x_{RTK} appartenga alla gaussiana con media x_m e varianza $var[x_{RTK}]_m$.

La base teorica su cui si fonda l'algoritmo è legata all'osservazione che dati ottenuti con RTK sono l'insieme di errori indipendenti (quali la misura

del metro, errori legati alle condizioni atmosferiche del segnale, riflessi di altri segnali, ecc...) e utilizzando la legge dei grandi numeri la media aritmetica di un campione di n variabili aleatorie, indipendenti e identicamente distribuite, per n crescente, tende o converge al valore atteso teorico μ . Inoltre il Teorema del limite centrale (TLC) afferma che in questa situazione la distribuzione dei dati obbedisce ad una legge gaussiana, nonostante le distribuzioni delle singole variabili possano essere del tutto generiche.²⁴

Per trovare i parametri a e b cercati, l'obbiettivo è quello di minimizzare lo scarto quadratico medio dei valori ottenuti, quindi di minimizzare la funzione:

$$E[(a \cdot x_{BLT} + b \cdot x_{RTK} - x)^2]$$

dove x è la distanza effettiva. Sviluppando i calcoli:

$$\begin{aligned} E[a^2 \cdot x_{BLT}^2 + b^2 \cdot x_{RTK}^2 + x^2 + 2ab \cdot x_{BLT} \cdot x_{RTK} - 2a \cdot x_{BLT} \cdot x - \\ - 2b \cdot x_{RTK} \cdot x] \end{aligned}$$

$$\begin{aligned} a^2 E[x_{BLT}^2] + b^2 E[x_{RTK}^2] + E[x^2] + 2ab E[x_{BLT} \cdot x_{RTK}] - 2a E[x_{BLT} \cdot x] - \\ - 2b E[x_{RTK} \cdot x] \end{aligned}$$

dove x è fissata alla distanza stabilita, assumiamo che, ad ogni distanza $E[x_{BLT}] = x$ e che $E[x_{RTK}] = x$, sostituendo otteniamo:

$$a^2 E[x_{BLT}^2] + b^2 E[x_{RTK}^2] + x^2 + 2abx^2 - 2ax^2 - 2bx^2$$

poichè in generale:

$$\begin{aligned} E[x^2] &= E[(x - E[x] + E[x])^2] \\ &= E[(x - E[x])^2 + E[x]^2 + 2E[x](x - E[x])] \\ &= E[(x - E[x])^2] + E[x]^2 + 2E[x]E[x - E[x]] \\ &= var[x] + E[x]^2 \end{aligned}$$

Sostituendo nel nostro caso otteniamo:

$$f(a, b) = a^2(var[x_{BLT}] + x^2) + b^2(var[x_{RTK}] + x^2) + x^2 + 2abx^2 - 2ax^2 - 2bx^2$$

Per minimizzare questa quantità, bisogna porre:

$$\frac{df(a, b)}{da} = \frac{df(a, b)}{db} = 0$$

$$\begin{cases} \frac{df(a, b)}{da} = 2a(var[x_{BLT}] + x^2) + 2bx^2 - 2x^2 = 0 \\ \frac{df(a, b)}{db} = 2b(var[x_{RTK}] + x^2) + 2ax^2 - 2x^2 = 0 \end{cases}$$

$$\begin{cases} a = \frac{x^2 - bx^2}{var[x_{BLT}] + x^2} \\ b = \frac{x^2 - ax^2}{var[x_{RTK}] + x^2} \end{cases}$$

$$\begin{cases} a = \frac{x^2 \cdot var[x_{RTK}]}{var[x_{BLT}] \cdot var[x_{RTK}] + x^2(var[x_{BLT}] + var[x_{RTK}])} \\ b = \frac{x^2 \cdot var[x_{BLT}]}{var[x_{BLT}] \cdot var[x_{RTK}] + x^2(var[x_{BLT}] + var[x_{RTK}])} \end{cases}$$

Ora abbiamo i parametri a e b espressi solo in funzione delle varianze delle variabili aleatorie e della distanza.

4.2 Risultati ottenuti

Le Fig. 4.1 e Fig. 4.2 mostrano un analisi complessiva dei dati raccolti sia con RTK che con RSSI, mettendo in evidenza la varianza campionaria. Interessante notare come fino a distanze inferiori a 3m il Bluetooth risulta più affidabile rispetto all'RTK, infatti mentre la varianza di quest'ultimo rimane abbastanza stabile con una media di 0.8610, quella del Bluetooth cresce in maniera quasi esponenziale.

	1m		2m		3m		4m	
Media	0.93	Media	1.91	Media	3.02	Media	5.24	
Errore standard	0.00	Errore standard	0.02	Errore standard	0.04	Errore standard	0.07	
Mediana	0.97	Mediana	1.85	Mediana	2.78	Mediana	5.31	
Moda	0.97	Moda	1.71	Moda	2.56	Moda	5.31	
Deviazione standard	0.11	Deviazione standard	0.42	Deviazione standard	0.84	Deviazione standard	1.57	
Varianza campionaria	0.01	Varianza campionaria	0.17	Varianza campionaria	0.71	Varianza campionaria	2.47	
Curtosi	1.45	Curtosi	90.92	Curtosi	3.58	Curtosi	0.67	
Asimmetria	-0.51	Asimmetria	6.66	Asimmetria	1.95	Asimmetria	0.95	
Intervallo	0.64	Intervallo	7.54	Intervallo	4.07	Intervallo	7.16	
Minimo	0.59	Minimo	0.43	Minimo	2.18	Minimo	3.01	
Massimo	1.23	Massimo	7.97	Massimo	6.25	Massimo	10.17	
Somma	465.14	Somma	955.01	Somma	1509.77	Somma	2619.85	
Conteggio	500	Conteggio	500	Conteggio	500	Conteggio	500	
Livello di confidenza(95.0%)	0.01	Livello di confidenza(95.0%)	0.04	Livello di confidenza(95.0%)	0.07	Livello di confidenza(95.0%)	0.14	

Figura 4.1: Dati relativi alle misurazioni RSSI

36

	1m		2m		3m		4m	
Media	1.49	Media	2.56	Media	3.25	Media	4.13	
Errore standard	0.30	Errore standard	0.28	Errore standard	0.32	Errore standard	0.27	
Mediana	1.19	Mediana	2.18	Mediana	2.95	Mediana	4.22	
Moda	#N/D	Moda	#N/D	Moda	#N/D	Moda	#N/D	
Deviazione standard	0.95	Deviazione standard	0.89	Deviazione standard	1.00	Deviazione standard	0.87	
Varianza campionaria	0.90	Varianza campionaria	0.78	Varianza campionaria	1.01	Varianza campionaria	0.75	
Curtosi	3.28	Curtosi	-0.68	Curtosi	0.34	Curtosi	1.26	
Asimmetria	1.85	Asimmetria	0.91	Asimmetria	0.96	Asimmetria	-0.44	
Intervallo	3.06	Intervallo	2.55	Intervallo	3.23	Intervallo	3.21	
Minimo	0.70	Minimo	1.53	Minimo	1.91	Minimo	2.39	
Massimo	3.76	Massimo	4.07	Massimo	5.14	Massimo	5.61	
Somma	14.94	Somma	25.60	Somma	32.52	Somma	41.35	
Conteggio	10	Conteggio	10	Conteggio	10	Conteggio	10	
Livello di confidenza(95.0%)	0.68	Livello di confidenza(95.0%)	0.63	Livello di confidenza(95.0%)	0.72	Livello di confidenza(95.0%)	0.62	

Figura 4.2: Dati relativi alle misurazioni RTK

Possiamo ora calcolare a e b per ogni distanza, come mostrato nella Tab. 4.1.

Distanza	a (BLT)	b (RTK)	Distanza	x_m
1m	0.97559	0.01284	1m	0.93m
2m	0.79167	0.17419	2m	1.96m
3m	0.56148	0.39441	3m	2.98m
4m	0.22577	0.73943	4m	4.24m

Tabella 4.1

Tabella 4.2

Come esempio possiamo utilizzare come dati i valori medi ottenuti dalle misurazioni con RTK e RSSI, trovando i relativi x_m come mostra la Tab. 4.2. Focalizzandoci ora ad esaminare i casi di 1m e 3m, ricordando anche che $\text{var}[x_{RTK}]_m = 0.8610$ e ipotizzando di ottenere come valore RSSI proprio il valor medio a quelle distanze, stimiamo la probabilità che ognuna delle x_{RTK} prese a questa distanza appartenga alla gaussiana $N(x_m, \sqrt{0.8610})$.

n	x_{RTK}	x_m	Prob.	n	x_m	x_{RTK}	Prob.
1	3.76m	0.96m	0.44%	6	1.03m	0.92m	42.7%
2	0.75m	0.92m	42.3%	7	0.69m	0.92m	41.8%
3	1.41m	0.93m	37.5%	8	0.92m	0.92m	42.9%
4	1.48m	0.93m	35.9%	9	1.30m	0.92m	39.6%
5	1.08m	0.92m	42.3%	10	2.51m	0.94m	10.4%

Tabella 4.3: Risultati ottenuti con le misurazioni a 1m

n	x_{RTK}	x_m	Prob.	n	x_m	x_{RTK}	Prob.
1	3.555m	3.097m	38.1%	6	3.025m	2.888m	42.5%
2	2.732m	2.773m	42.9%	7	2.880m	2.831m	42.9%
3	3.288m	2.992m	40.9%	8	2.481m	2.674m	42.1%
4	5.140m	3.722m	13.3%	9	4.763m	3.574m	18.9%
5	2.744m	2.777m	42.9%	10	1.908m	2.448m	36.3%

Tabella 4.4: Risultati ottenuti con le misurazioni a 3m

Risultati simili si possono ottenere anche per le altre misure, in cui circa 2/3 misure hanno una probabilità nettamente inferiore alle altre.

4.3 Conclusioni

Come si può notare se venisse registrata una sola distanza RTK i risultati potrebbero essere fuorvianti e potrebbe essere segnalato un falso allarme di spoofing.

Per evitare questo comportamento la soluzione migliore è quella di raccogliere un insieme di misure e di utilizzarne la media per calcolare la probabilità. Infatti negli esempi precedenti (Tab. 4.3 e Tab. 4.4) considerando le medie si ottengono rispettivamente le probabilità del 35.6% e 41.2%. Si può quindi stabilire una soglia del 30%, al di sotto della quale viene considerato il dispositivo sotto attacco GPS.

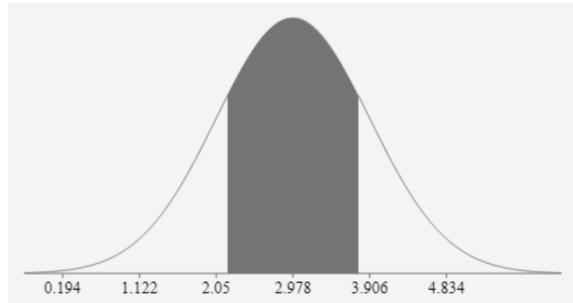


Figura 4.3: Esempio a 3m della soglia 30%

Un altro aspetto importante da tenere in considerazione riguarda il parametro b nel calcolo di x_m : all'aumentare della distanza, l'affidabilità legata alla potenza RSSI va diminuendo (parametro a) e di conseguenza aumenta quella legata all'RTK (parametro b). Questo comportamento sposta il valor medio della gaussiana verso il valore di distanza misurato dall'RTK, di conseguenza la probabilità che la distanza RTK rientri nella gaussiana si alza considerevolmente (come nella misura 10 della Tab. 4.4). Per evitare questo fenomeno l'unica soluzione è mantenere i due telefoni ad una distanza di massimo 1/2 metri, in modo che la misurazione fornita dall'RSSI sia prevalente rispetto a quella dell'RTK.

In conclusione quindi, mantenendo i telefoni vicini, l'algoritmo trovato può stabilire se il telefono in considerazione è sotto attacco di spoofing GPS solo se l'attacco è superiore a circa 80cm dalla posizione reale.

Bibliografia

¹ Rune Bearson, "*What is the difference between Bluetooth versions?*", Ear-Rockers, 2019.

<https://earrockers.com/difference-between-bluetooth-versions/>

² Ian Poole, "*Bluetooth radio interface, modulation, channels*", Electronics-Notes.

<https://www.electronics-notes.com/articles/connectivity/bluetooth/radio-interface-modulation-channels.php>

³ Keysight team, " *$\pi/4$ DQPSK (Digital Demod)*", Keysight.

https://rfmw.em.keysight.com//wireless/helpfiles/89600b/webhelp/subsystems/digdemod/content/dlg_digdemod_fmt_pi4dpsk.htm

⁴ Keysight team, "*D8PSK (Digital Demod)*", Keysight.

https://rfmw.em.keysight.com//wireless/helpfiles/89600b/webhelp/Subsystems/digdemod/Content/dlg_digdemod_fmt_d8psk.htm

⁵ Andrea Zanella, "*Network performance indexes*", Reti di calcolatori 1, 2021.

⁶ Fastweb team, "*Caratteristiche Bluetooth 5.1*", FastwebDigitalMagazine, 2019.

<https://www.fastweb.it/internet/caratteristiche-bluetooth-5-1/>

⁷ Apple team, "*Apple presenta AirTag*", COMUNICATO STAMPA, 2021.

<https://www.apple.com/it/newsroom/2021/04/apple-introduces-airtag/>

⁸ Thuy Mai, "*Global Positioning System History*", NASA, 2012.

https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html

⁹ Giulio Peruzzi, "*Lezione 4 - Dalla scoperta dell'elettrone alle teorie della relatività di Einstein*", Corso di Storia della Tecnologia dell'Informazione 20/21, 2020.

- ¹⁰ Richard B. Langley, "*Innovation: GLONASS — past, present and future*", GPS WORLD, 2017.
<https://www.gpsworld.com/innovation-glonass-past-present-and-future/>
- ¹¹ BeiDou team, "*System*", BeiDou Navigation Satellite System.
<http://en.beidou.gov.cn/SYSTEMS/System/>
- ¹² Marco Bruno, "*Il GPS – Global Positioning System*", AstronautiNEWS, 2018.
<https://www.astronautinews.it/2018/06/il-gps-global-positioning-system/>
- ¹³ Gian Bartolomeo Siletto, Piera Belotti, Monica Segré, Marzio Pipino, Mattia De Agostino, "*L'introduzione della costellazione Galileo nelle reti GNSS: quali vantaggi?*", Asita, 2019.
<http://atti.asita.it/ASITA2019/Pdf/096.pdf>
- ¹⁴ Salvatore Mele, Paolo Strolin, "*Il Sistema di Posizionamento Globale (GPS)*", 2009.
http://www1.na.infn.it/fisicainbarca/GPS_INFN.pdf
- ¹⁵ Giulio Peruzzi, "*Lezione 4 - Dalla scoperta dell'elettrone alle teorie della relatività di Einstein*", Corso di Storia della Tecnologia dell'Informazione 20/21, pp. 60-61, 2020.
- ¹⁶ Gary Taubes, "*The Global Positioning System: The Role of Atomic Clocks*", Beyond Discovery, pp. 1-3 1997.
<http://www.nasonline.org/publications/beyond-discovery/the-global-positioning-system.pdf>
- ¹⁷ Hexagon AB team, "*Real-Time Kinematic (RTK)*", Hexagon AB.
<https://novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk>
- ¹⁸ Mobiix team, "*GPS ad alta precisione, RTK e RT81A*", Mobiix srl.
<https://www.mobiix.it/gps-alta-precisione-rtk-rt81a/>
- ¹⁹ Ksenia Volodina, "*What Is NTRIP and How to Use It for RTK with Reach*", Emlid.
<https://emlid.com/what-is-ntrip-and-how-to-use-it-for-rtk-with-reach/>
- ²⁰ Tyler, "*WHAT IS NTRIP?*", Adopting Mobile GIS, 2017.
https://www.agsgis.com/What-is-NTRIP_b_42.html
- ²¹ Nevio Benvenuto, Michele Zorzi, "*Principles of Communications Networks and Systems*", WILEY, pp. 260-262, 2011.

²² Vincent Gao, "*Proximity and RSSI*", Bluetooth team, 2015.
<https://www.bluetooth.com/blog/proximity-and-rssi/>

²³ Mario Spada, "*Calcolo della distanza geodetica tra due punti della superficie terrestre*", Mario Spada website, 2007.

²⁴ Web Tutor team, "*Legge dei Grandi Numeri e Teorema del Limite Centrale (TLC)*", WebTutor website.
<https://webtutordimematica.it/materie/statistica-e-probabilita/teoria-dei-campioni/legge-dei-grandini-numeri-e-teorema-del-limite-centrale-tlc>