

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

MISURE DI CANALE RADIO PER

AUTENTICAZIONE ALLO STRATO FISICO

Relatore:
Prof. Stefano Tomasin

Laureando:
Zhen Yu Lorenzo Huang

ANNO ACCADEMICO: 2024/2025

Data di laurea: 21 07 2025

Indice

1	Introduzione	1
2	Fondamenti Teorici	3
2.1	Segnali	3
2.2	Modello del canale Wireless	6
2.3	Stima di canale e sequenze pilota	7
2.3.1	Correlazione incrociata	8
2.3.2	Free Space Path Loss (FSPL)	9
2.4	Problema di autenticazione	10
3	Strumenti: ADALM-Pluto e MATLAB	15
3.1	ADALM-Pluto	15
3.2	MATLAB e Support Package	16
3.3	SDRangel	17
4	Sperimentazioni	19
4.1	Trasmissione e ricezione in banda base con PlutoSDR	19
4.2	Analisi del Multipath con Segnale QPSK	20
4.3	Analisi del Canale tramite Correlazione Chirp	22
4.3.1	Obiettivo dell'Esperimento	22
4.3.2	Descrizione del Setup	23
4.3.3	Procedura Sperimentale	23
4.3.4	Elaborazione e Analisi dei Dati	25
5	Conclusioni	29
A	Codice sorgente MATLAB	33
	Bibliografia	46

Sommario

In questa tesi viene analizzato il comportamento del canale radio in condizioni di propagazione in linea di vista (LoS), mediante l'utilizzo della piattaforma Software Defined Radio (SDR) ADALM-Pluto e dell'ambiente MATLAB. Dopo una fase iniziale di verifica delle funzionalità di base di trasmissione e ricezione, viene studiato l'effetto del multipath e dell'attenuazione spaziale attraverso l'impiego di segnali chirp e tecniche di correlazione incrociata. I risultati sperimentali evidenziano una buona corrispondenza con il modello teorico di perdita di spazio libero (Free Space Path Loss, FSPL), consentendo una stima approssimata della risposta impulsiva del canale. Infine, viene proposto un metodo per distinguere trasmettitori differenti in base all'attenuazione delle correlazioni, aprendo prospettive interessanti per applicazioni di autenticazione a basso costo e analisi ambientale del canale.

Capitolo 1

Introduzione

Negli ultimi anni, l'interesse verso i sistemi di comunicazione wireless ha portato allo sviluppo di strumenti sempre più flessibili e accessibili per l'analisi e la prototipazione di scenari radio reali. Tra questi, i dispositivi *Software Defined Radio* (SDR) come l'ADALM-Pluto rappresentano una piattaforma potente ed economica per la sperimentazione in ambito radiofrequenza.

Lo scopo di questa tesi è analizzare il comportamento di un canale radio in presenza di propagazione in linea di vista (LoS), con particolare attenzione ai fenomeni di attenuazione e multipath. Si sfrutta un approccio basato sulla trasmissione ripetuta di un segnale noto (chirp) e sull'analisi della correlazione incrociata tra il segnale ricevuto e quello trasmesso. Questo consente di stimare la risposta impulsiva del canale e verificare la coerenza sperimentale con il modello teorico di attenuazione *Free Space Path Loss* (FSPL).

Inoltre, viene esplorata la possibilità di utilizzare la correlazione come strumento per l'autenticazione del trasmettitore, basandosi sulle caratteristiche spaziali e temporali del segnale ricevuto. Tale approccio può trovare

applicazione in scenari di sicurezza e accesso controllato, dove è importante riconoscere o localizzare l'origine di una trasmissione.

La tesi è articolata come segue: nel Capitolo 2 vengono introdotti i fondamenti teorici relativi alla propagazione radio, alla correlazione incrociata e ai modelli di canale. Il Capitolo 3 descrive gli strumenti hardware e software impiegati, con particolare riferimento al PlutoSDR e all'ambiente MATLAB. Il Capitolo 4 riporta le attività sperimentali, i risultati delle misure e l'analisi dei dati. Infine, nel Capitolo 5 vengono tratte le conclusioni e discussi i possibili sviluppi futuri.

Capitolo 2

Fondamenti Teorici

2.1 Segnali

Principalmente la gestione dei segnali avviene attraverso l'acquisizione e digitalizzazione. Per questo è importante conoscere le caratteristiche dei segnali discreti.

Segnali discreti

I segnali discreti sono delle sequenze di valori, dove ogni valore è associato a un istante di tempo specifico e distinto, generalmente rappresentato da numeri interi, si possono rappresentare come

$$s(nT) \quad n \in \mathbb{Z} \tag{2.1}$$

dove $T = \frac{1}{f_s}$ indica il periodo di campionamento (la quantità temporale), con f_s che rappresenta la frequenza di campionamento, cioè l'inverso del numero

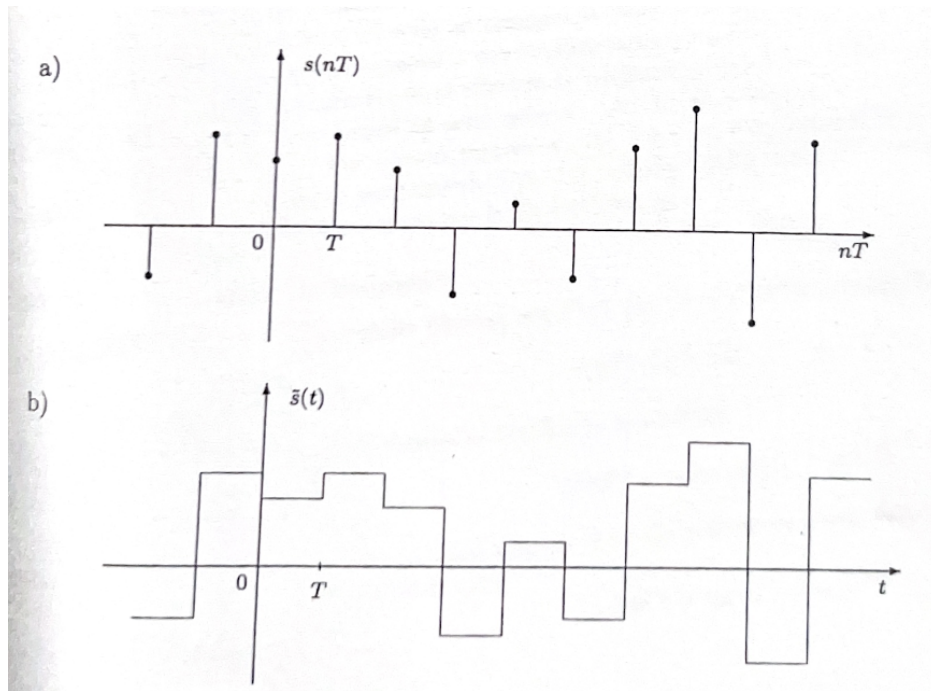


Figura 2.1: a) Esempio di segnale a tempo discreto e b) corrispondente segnale continuo ottenuto mediante "interpolazione a tenuta".

di valori campionati del segnale per unità di tempo.

Area e valore medio

L'area di un segnale discreto, intesa nel senso dei segnali continui sarebbe sempre nulla. Per avere un parametro significativo e utile occorre modificare la definizione come segue:

$$area(s) = \sum_{n=-\infty}^{\infty} T s(nT); \quad (2.2)$$

in tal modo ogni singolo valore $s(nT)$ contribuisce con un'area concentrata pari a $T s(nT)$ come illustrato in figura 2.1

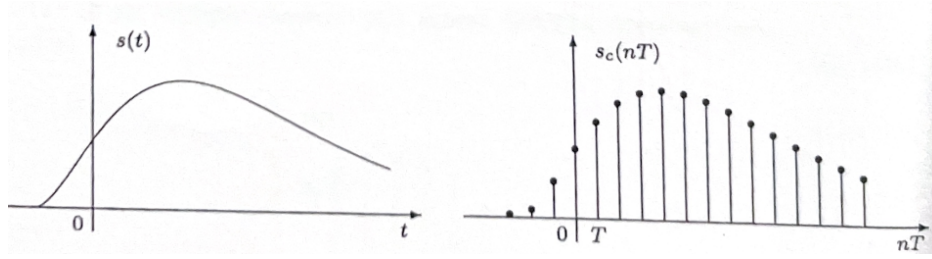


Figura 2.2: Esempio di campionamento di un segnale a tempo continuo.

Si può tramite interpolazione a tenuta ottenere dal segnale $s(nT)$ il segnale $\tilde{s}(t)$, $t \in \mathbb{R}$, dove il valore medio, o componente continua di un segnale discreto è definito:

$$m_s = \lim_{N \rightarrow +\infty} \frac{1}{(2N+1)T} \sum_{n=-N}^{+N} T s(nT) \quad (2.3)$$

Campionamento

I segnali discreti possono essere definiti direttamente, ma più frequentemente essi sono ricavati da segnali continui restringendone i valori dal dominio continuo \mathbb{R} al dominio discreto $Z(T)$. Questa operazione viene detta campionamento:

$$s_c(nT) = s(nT), \quad nT \in (T) \quad (2.4)$$

Come illustrato come in esempio in figura 2.2, dove $s(t)$, $t \in \mathbb{R}$, è il segnale continuo di riferimento ed $s_c(nT)$, $nT \in (T)$ è il segnale discreto ottenuto per campionamento.

2.2 Modello del canale Wireless

Il canale wireless viene principalmente influenzato dalla variazione della potenza del segnale in tempo e in frequenza. Queste variazioni sono chiamate come fenomeni di fading, dove sono suddivisi in:

- Fading di larga scala: viene causato quando il segnale subisce dispersione della potenza causato per passare tra muri e edifici. In questo caso la frequenza non subisce variazioni
- Fading di bassa scala: quando esistono segnali multipli tra il trasmettitore e ricevitore causato da riflessioni del segnale tramite gli edifici, gli oggetti. In questo caso la frequenza dipende dalla distanza tra il trasmettitore e ricevitore.

Il canale wireless può essere modellato come un sistema lineare a tempo-variante

$$y(nT) = \sum_{\ell} h(\ell, nT) \cdot x(nT - \ell) + w(nT) \quad n \in \mathbb{Z} \quad (2.5)$$

dove $y(nT)$ rappresenta la risposta di $h(\ell, nT)$ al tempo l di un impulso trasmesso al tempo $nT - \ell$, $x(nT)$ rappresenta il segnale trasmesso e $w(nT)$ rappresenta il rumore additivo. La risposta impulsiva per causa dell'effetto Multipath, fading di bassa scala è

$$h(\ell) = \sum_i a_i \delta(\ell - \ell_i) \quad (2.6)$$

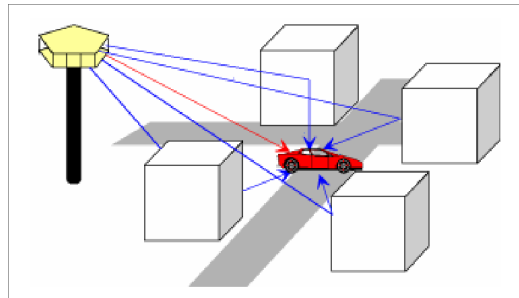


Figura 2.3: Esempio di Multipath[1]

dove ogni componente a_i rappresenta l'attenuazione (e fase) su un percorso i -esimo, con un ritardo relativo ℓ_i dovuto alla diversa lunghezza del percorso. In termini fisici, il primo termine corrisponde tipicamente al *path diretto* tra trasmettitore e ricevitore, mentre gli altri termini rappresentano segnali che arrivano tramite riflessioni su superfici (edifici, muri, oggetti) o rifrazioni. Come illustrato nella figura 2.3 il *path diretto* in rosso arriverà per primo e successivamente tutte le altre riflessioni.

2.3 Stima di canale e sequenze pilota

Per contrastare gli effetti negativi del canale multipath, vengono applicati modelli matematici e tecniche per modellare i segnali. Lo studio della stima di canale consente di stimare l'affidibilità e robustezza del canale, tramite lo studio della risposta impulsiva del canale $h(nT)$ per rilevare nel segnale ricevuto i segnali trasmessi si trasmettono simboli noti \mathbf{p} di lunghezza N con cui tramite la correlazione possiamo identificarli e stimare il canale h . Questi segnali noti possono essere Barker, Pseudo-Noise, chirp etc.[2]

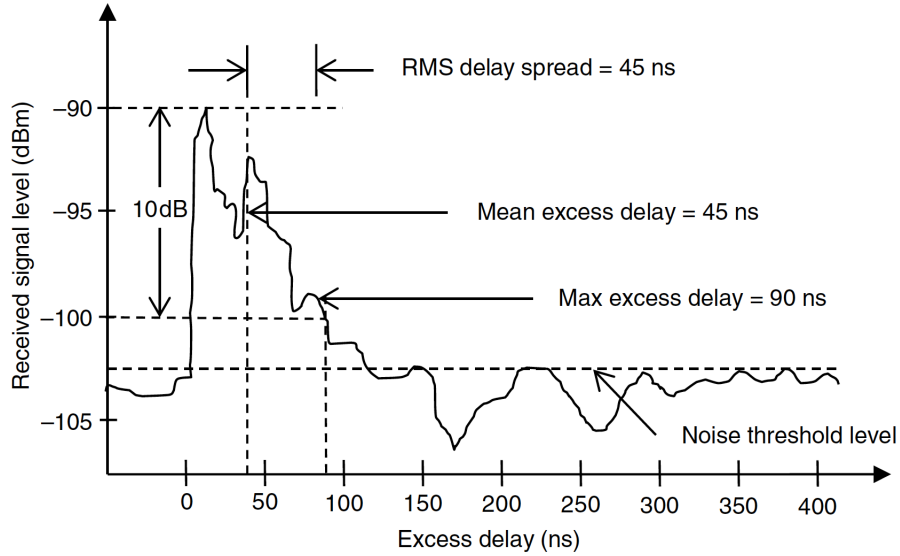


Figura 2.4: Esempio di profilo ritardo-potenza in un canale multipath, con indicazione di ritardo medio, ritardo massimo e delay spread RMS (in ns). I livelli di segnale ricevuto in dBm sono tracciati rispetto al ritardo eccedente (excess delay) rispetto al primo arrivo.[3]

2.3.1 Correlazione incrociata

La *correlazione incrociata* tra due segnali è una misura di similarità in funzione di uno scorrimento temporale (ritardo) di uno dei due segnali rispetto all'altro. Data una serie di campioni ricevuti $r(n)$ e una sequenza nota di riferimento $p(n)$, la funzione di cross-correlazione $C_{rp}(\ell)$ è definita come:

$$C_{rp}(\ell) = \sum_n r(n) p^*(n + \ell) \quad (2.7)$$

dove $*$ denota il complesso coniugato. Calcolando $C_{rp}(\ell)$ si ottiene un segnale di output di cui i *picchi* indicano i valori di ritardo ℓ per cui il segnale ricevuto r "somiglia" maggiormente al segnale noto p . Nei sistemi di comunicazione, l'utilizzo di sequenze pilota con buone proprietà di auto-

correlazione (ad esempio sequenze di Barker o Pseudo-Noise) permette di ottenere un *profilo del canale* stimando i tap multipath come picchi della cross-correlazione.[4] Ideando opportunamente la sequenza trasmessa, l'uscita del correlatore fornisce informazioni su ritardi ed attenuazioni delle varie componenti del canale.[5][6]

2.3.2 Free Space Path Loss (FSPL)

L'attenuazione di spazio libero (Free Space Path Loss, FSPL) rappresenta l'attenuazione della potenza di un segnale elettromagnetico risultante da un percorso in linea di vista attraverso lo spazio libero (di solito l'aria), senza incontrare ostacoli che possano causare riflessioni o diffrazioni. La formula base è

$$\text{FSPL} = \left(\frac{4\pi d}{\lambda} \right)^2 = \left(\frac{4\pi df}{c} \right)^2 \quad (2.8)$$

dove d è la distanza dal ricevitore al trasmettitore in metri, f è la frequenza di trasmissione del segnale, c è la velocità della luce nel vuoto, λ è la lunghezza d'onda del segnale. FSPL può essere espressa anche in decibel tramite la seguente formula:

$$\text{FSPL (dB)} = 20 \log_{10}(d) + 20 \log_{10}(f) + C \quad (2.9)$$

dove la costante C dipende dalle unità utilizzate per d e f . Ad esempio, con d in km e f in GHz, $C \approx 92,45$ dB. In alternativa, con d in metri e f in Hz, si usa spesso $C = -147,55$ dB (o altre costanti equivalenti per MHz, etc.).[7]

Nel caso di studio, supponiamo di operare a frequenza $f \approx 3$ GHz. La Tabella 2.1 mostra i valori teorici di FSPL calcolati a varie distanze entro 2 m, utilizzando la formula sopra (distanza in metri, convertita in km nella formula, con $f = 3$ GHz):

Tabella 2.1: FSPL a 3 GHz per diverse distanze

Distanza (m)	FSPL (dB)
0.10	21.9
0.20	28.1
0.40	34.0
0.60	37.5
0.80	40.0
1.00	41.9
1.20	43.6
1.40	44.9
1.60	46.1
1.80	47.1
2.00	48.0

2.4 Problema di autenticazione

Il problema di autenticazione consiste nel verificare che un messaggio ricevuto sia stato effettivamente trasmesso dal dispositivo che si dichiara come mittente. Per fare questa verifica, possiamo controllare che il canale attraverso cui il messaggio ricevuto coincida con quello del trasmettitore legittimo. A questo fine, prima si stima il canale e lo si confronta con quello legittimo.

Supponendo per ipotesi che l'attenuazione dei picchi di correlazione tra il segnale ricevuto e il segnale trasmesso sia stabile rispetto la distanza e posizione di trasmissione, si può identificare se il segnale trasmesso proviene dal trasmettitore nella stessa posizione, se il trasmettitore è legittimo.

Purtroppo i segnali ricevuti possono oscillare di tanto a causa del clock interno dei dispositivi utilizzati. Per aumentare la probabilità di detection e diminuire P_{FA} , probabilità di falso allarme si introduce il Teorema di Neyman-Pearson.

Teorema di Neyman-Pearson

Il teorema di Neyman-Pearson fornisce il criterio ottimale per il test binario tra due ipotesi semplici per diminuire la probabilità di falso allarme (P_{FA}) nel test del rapporto di verosimiglianza (*Likelihood-ratio test*, LRT). Il teorema afferma che per massimizzare la percentuale di detection(P_D), dato $P_{FA} = \alpha$ si decide \mathcal{H}_1 se

$$L(x) = \frac{p(x|\mathcal{H}_1)}{p(x|\mathcal{H}_0)} > \gamma \quad (2.10)$$

dove γ rappresenta il threshold da

$$P_{FA} = \int_{x:L(x)>\gamma} p(x); \mathcal{H}_0 dx = \alpha \quad (2.11)$$

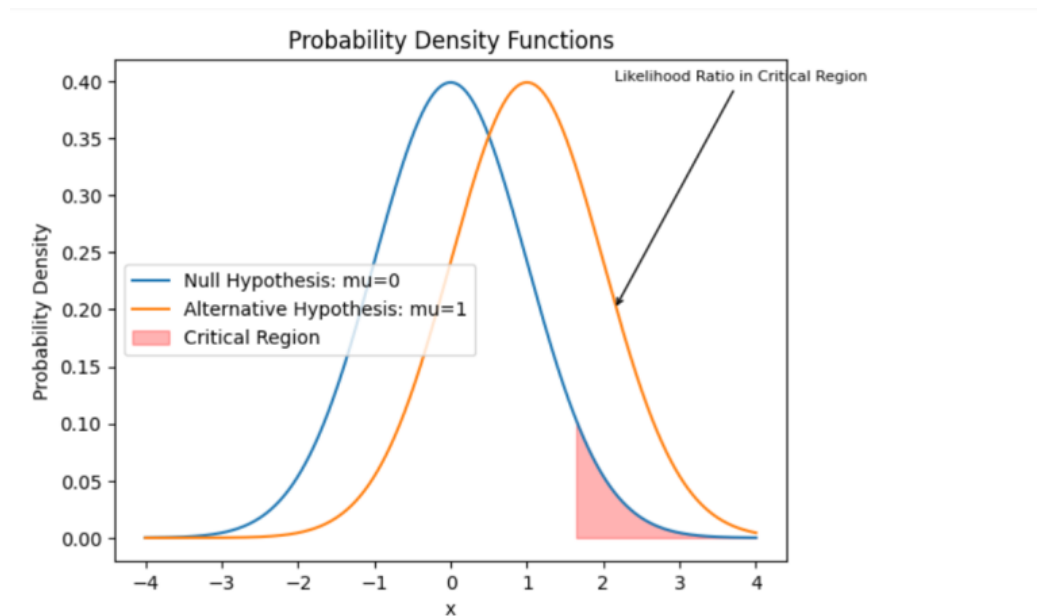
Esso afferma che, per un dato livello di falso allarme massimo tollerabile α , il test di massima somiglianza è il più potente, ovvero quello che massimizza la probabilità di rilevare correttamente l'ipotesi \mathcal{H}_1 .

Il risultato garantisce che, tra tutti i test con $P_{FA} \leq \alpha$, quello basato sul rapporto di verosimiglianza è quello che minimizza il P_{MD} , rendendolo ottimale secondo il criterio di Neyman-Pearson.[8]

Nel prendere decisioni statistiche, si possono commettere due tipi di errore:

- **Falso allarme (false alarm, errore di tipo I):** decidere \mathcal{H}_1 quando è vera \mathcal{H}_0 - ovvero identificare erroneamente un intruso come legittimo;
- **Mancato rilevamento (missed detection, errore di tipo II):** decidere \mathcal{H}_0 quando è vera \mathcal{H}_1 - ovvero non riconoscere un trasmettitore legittimo.

L'immagine 2.5 rappresenta una tipica applicazione del teorema di Neyman-Pearson nel contesto del test di ipotesi tra due ipotesi semplici. Le due curve mostrano le funzioni di densità di probabilità (PDF) associate all'ipotesi nulla H_0 (in blu, con media $\mu = 0$) e all'ipotesi alternativa H_1 (in arancione, con media $\mu = 1$). L'area evidenziata in rosa rappresenta la regione critica, ovvero l'intervallo dei valori osservati di x per cui si decide di rifiutare H_0 a favore di H_1 . La freccia indica il punto in cui il rapporto di verosimiglianza $L(x) = \frac{p(x|H_1)}{p(x|H_0)}$ supera una certa soglia: secondo il teorema di Neyman-Pearson, questo criterio definisce il test più potente possibile per un dato livello di significatività α . In altre parole, l'immagine mostra come il test basato sul confronto tra le densità consenta di separare le due ipotesi minimizzando la probabilità di errore di tipo II (quando si rifiuta H_0 mentre è vera), dato un vincolo sulla probabilità di errore di tipo I (quando si accetta H_0 mentre è falsa).



[9]

Figura 2.5: Esempio di del teorema di Neyman-Pearson.

Capitolo 3

Strumenti: ADALM-Pluto e MATLAB

3.1 ADALM-Pluto

La piattaforma di *Software-Defined Radio (SDR)* utilizzata è il modulo **ADALM-Pluto**, prodotto da Analog Devices. Pluto è un dispositivo *trasmettitore/ricevitore RF* portatile e a basso costo, basato sul transceiver analogico **ADI AD9363** e su un FPGA Xilinx Zynq-7010 con CPU ARM integrata. Le specifiche principali includono:

- Copertura in frequenza da circa 325 MHz a 3.8 GHz (estensibile fino a 6 GHz in alcune revisioni).
- Banda istantanea fino a 20 MHz.
- Campionamento a 12 bit con rate fino a 61.44 MS/s.
- Alimentazione e interfaccia dati tramite USB 2.0.

Pluto offre connettori SMA per antenna in trasmissione (Tx) e ricezione (Rx) e un supporto software completo, che include librerie (libio) e un *toolbox* per MATLAB. Ciò consente di configurare in modo flessibile i parametri RF (frequenza, guadagno, larghezza di banda) e di trasmettere/ricevere campioni I/Q in tempo reale.[10]

3.2 MATLAB e Support Package

In questa tesi si è fatto uso di *MATLAB* e del **Communications Toolbox Support Package for ADALM-Pluto Radio** di MathWorks. Il toolbox mette a disposizione oggetti `System` e funzioni specifiche per la gestione di Pluto come:

- `comm.SDRTxPluto` per trasmettere campioni I/Q.
- `comm.SDRRxPluto` per ricevere campioni I/Q.

Il supporto package contiene anche esempi preconfigurati di comunicazione digitale, tra cui i due script fondamentali per la trasmissione e ricezione *QPSK*, accessibili con i comandi:

```
openExample('plutoradio/QPSKTransmitterWithADALMPLUTORadioExample')
openExample('plutoradio/QPSKReceiverWithADALMPLUTORadioExample', ...
            'supportingFile','plutoradioqpskreceiver_init.m')
```

Questi esempi implementano un sistema QPSK completo con preambolo (*Barker code*) e funzioni di sincronizzazione (timing, Carrier Frequency Offset (CFO)) al ricevitore.[11]

3.3 SDRangel

SDRangel è un'applicazione software open source progettata per la ricezione e la trasmissione di segnali radio mediante dispositivi a radio definita via software (SDR). Il software fornisce un'interfaccia utente grafica per la sintonizzazione dei segnali radio e una varietà di strumenti di elaborazione del segnale per la demodulazione, la decodifica e la visualizzazione dei dati ricevuti.

SDRangel supporta un'ampia gamma di tipi di segnale, tra cui AM, FM, SSB, CW, DAB, DVB-S2, ADS-B e altri. Il software supporta anche la registrazione e la riproduzione di segnali RF, nonché il controllo remoto tramite una connessione di rete. SDRangel è uno strumento potente e versatile per esplorare il mondo dei segnali radio e può essere utilizzato sia per applicazioni pratiche come il monitoraggio radio, sia per progetti di ricerca accademica che coinvolgono l'elaborazione digitale del segnale. Il software è disponibile per sistemi operativi Linux e Windows, ed è particolarmente apprezzato per la sua flessibilità, le prestazioni offerte e il supporto garantito da una comunità di sviluppo attiva e dinamica.[12]

Capitolo 4

Sperimentazioni

4.1 Trasmissione e ricezione in banda base con PlutoSDR

La prima fase dell'attività sperimentale ha avuto come obiettivo la verifica del corretto funzionamento del dispositivo **ADALM-PLUTO SDR** e la capacità di trasmettere e ricevere un segnale. Inizialmente, è stato trasmesso un segnale composto da due sinusoidi complesse a frequenze differenti (1.1 MHz e 1.5 MHz), generate in ambiente *MATLAB*. Il segnale complesso risultante è stato visualizzato tramite il software **SDRangel**, che consente di osservare in tempo reale lo spettrogramma del segnale ricevuto.

Questa prima analisi ha permesso di verificare visivamente che la trasmissione fosse correttamente configurata e che le componenti spettrali fossero correttamente posizionate. In una fase successiva, si è passati alla trasmissione e ricezione dello stesso segnale direttamente all'interno di *MATLAB*, uti-

lizzando un singolo dispositivo PlutoSDR per entrambe le funzioni (loopback radio).

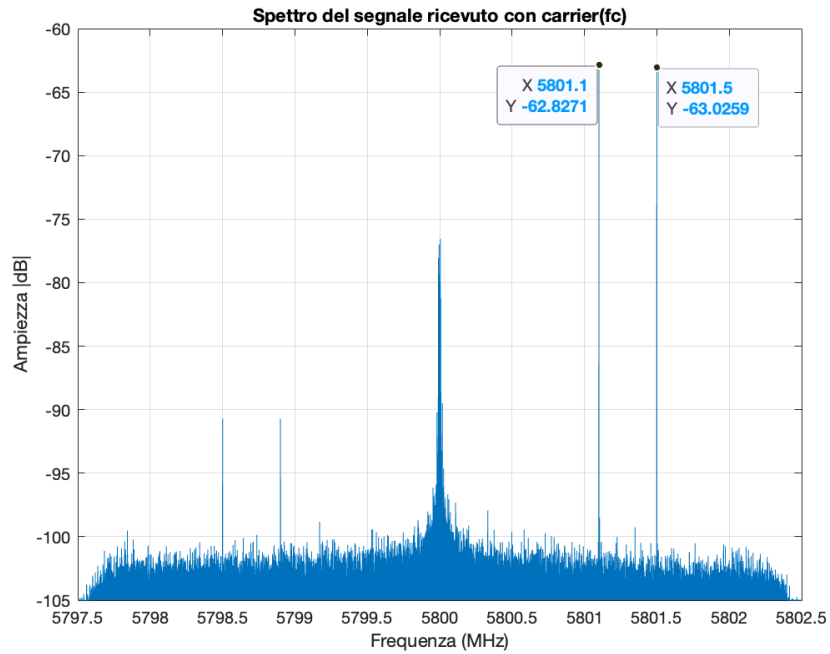
La frequenza centrale della trasmissione è stata impostata a 5.8 GHz, mentre la frequenza di campionamento a 5 Ms^{-1} , valori compatibili con le capacità del dispositivo. Il segnale trasmesso è stato normalizzato (come illustrato nel codice A.1). In ricezione, il segnale acquisito è stato elaborato per visualizzarne lo spettro tramite una trasformata di Fourier (FFT), al fine di confermare la presenza dei due toni trasmessi (come illustrato nel codice A.2).

Questa configurazione ha consentito di testare la catena di trasmissione e ricezione, nonché di verificare le impostazioni iniziali del dispositivo in un ambiente controllato come quanto illustrato in Figura 4.1a e dal suo rispettivo segnale ricevuto nel tempo Figura 4.1b.

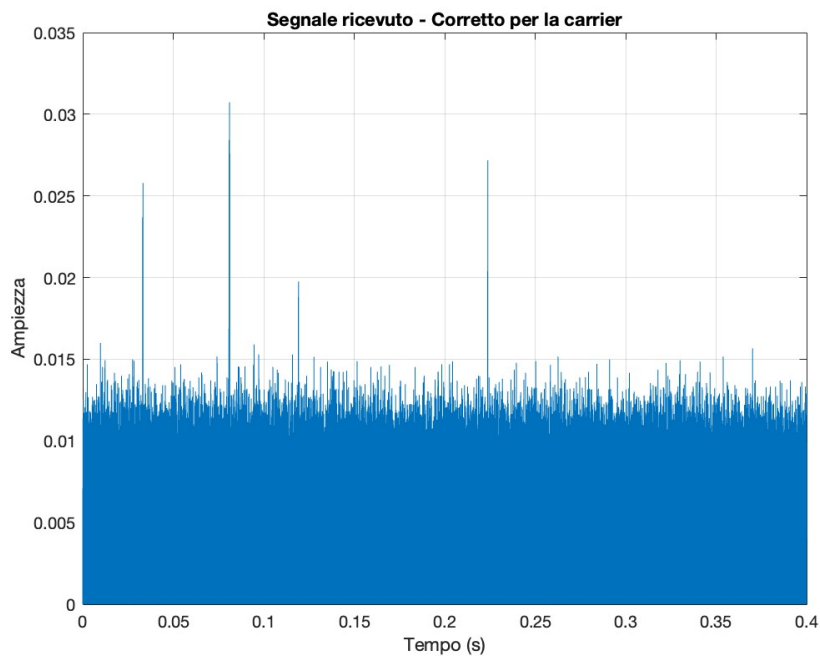
4.2 Analisi del Multipath con Segnale QPSK

Nella seconda fase dell'attività sperimentale è stato impiegato il software sviluppato da MathWorks denominato *QPSK Transmitter Using Software-Defined Radio*, evoluzione del precedente *QPSK Transmitter With ADALM-PLUTO Radio* disponibile prima del 2025. Nella nuova versione, l'architettura è stata resa più flessibile attraverso l'utilizzo dello script `helperFindRadios.m`, che consente la configurazione del tipo di radio SDR impiegata. In parallelo, è stato utilizzato anche il blocco ricevitore corrispondente, *QPSK Receiver Using Software-Defined Radio*.

L'obiettivo principale di questa fase era l'individuazione di eventuali ef-



(a) Spettrogramma del segnale ricevuto con banda centrata a 5GHz



(b) Segnale nel tempo

fetti di propagazione multipath nel segnale ricevuto. A tal fine, è stata visualizzata la classe `FrameSynchronizer.m`, in particolare il comando:

```
[idx, dtMt] = obj.pPreambleDetector(x);
```

Tale comando esegue una correlazione tra il segnale ricevuto e il preambolo, con lo scopo di localizzare l'inizio del frame. Per migliorare la rilevazione dei cammini multipli, è stato tentato un allungamento del preambolo Barker mediante una sequenza pseudo-casuale (PN sequence) di lunghezza fino a 17 simboli.

Tuttavia, i risultati ottenuti non sono stati soddisfacenti. Si è trovato un livello di rumore di fondo particolarmente elevato, tale da mascherare eventuali picchi secondari riconducibili al multipath. Di conseguenza, si è deciso di proseguire alla fase successiva utilizzando un segnale di tipo chirp, più adatto a questo tipo di analisi.

4.3 Analisi del Canale tramite Correlazione Chirp

4.3.1 Obiettivo dell'Esperimento

Lo scopo dell'esperimento è da un lato, individuare la presenza di fenomeni di *multipath* tramite l'analisi dei segnali ricevuti, e dall'altro stimare la risposta del canale radio in scenari in *linea di vista* (*Line of Sight*, LoS). Il metodo impiegato si basa sull'analisi della funzione di *correlazione incrociata* tra un segnale noto trasmesso (chirp) e il segnale ricevuto acquisito da un secondo dispositivo PlutoSDR. Lo scopo è anche mostrare che allontanando

la distanza dal ricevitore si ha che l'attenuazione del segnale decresce come FSPL.

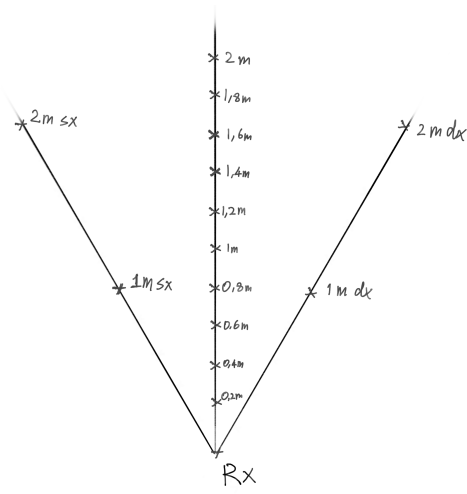
4.3.2 Descrizione del Setup

Per questo esperimento sono stati impiegati due dispositivi *ADALM-Pluto*, uno configurato come trasmettitore e l'altro come ricevitore. Il segnale trasmesso è un *chirp* complesso della durata di 8192 campioni, generato con frequenza lineare crescente da -20 MHz a $+20$ MHz, rispetto al centro banda. Il segnale viene trasmesso in modo ripetuto con un *sampling rate* di 40 MHz e una frequenza centrale di 3 GHz.

Il ricevitore, anch'esso configurato con un *sampling rate* di 40 MHz, acquisisce blocchi di 8 milioni di campioni per ciascuna ricezione.

4.3.3 Procedura Sperimentale

Durante le misure, il ricevitore è stato mantenuto *fisso*, mentre il trasmettitore è stato posizionato a distanze crescenti, da 20 cm fino a 200 cm, in incrementi di 20 cm come illustrato in Figura 4.2. Per ciascuna posizione del ricevitore sono state effettuate 10 acquisizioni indipendenti in condizioni di linea di vista diretta (LoS). Inoltre, sono stati eseguiti test aggiuntivi con spostamenti laterali del trasmettitore alla distanza di 100 cm e 200 cm, per studiare eventuali asimmetrie dovute all'ambiente.



(a)

(b)



(c)

Figura 4.2: Foto per le rilevazioni

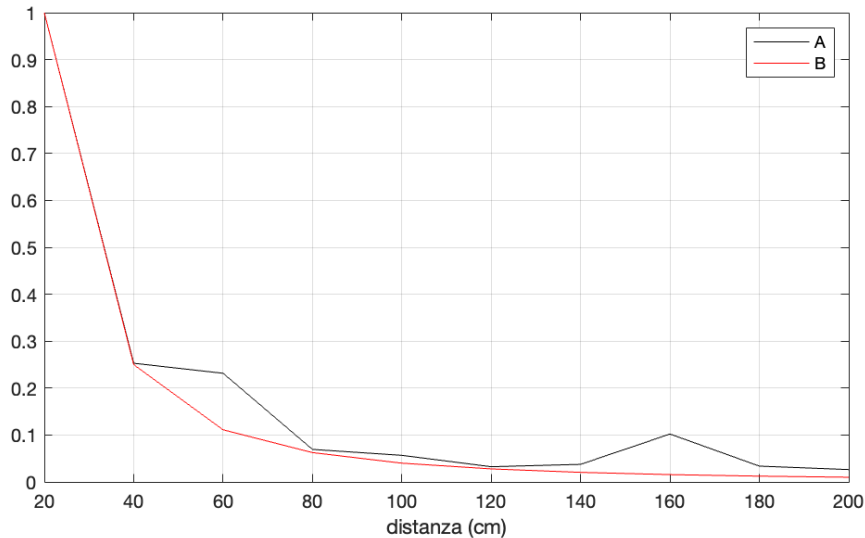


Figura 4.3: Attenuazioni del segnale normalizzato rispetto alla distanza. Dove A rappresenta la media dei valori di l'attenuazione dei picchi di correlazione; B rappresenta l'inverso della FSPL normalizzato.

4.3.4 Elaborazione e Analisi dei Dati

L'analisi è stata condotta mediante la *correlazione incrociata* tra il segnale ricevuto e il chirp trasmesso. Data la periodicità della trasmissione, la correlazione presenta una sequenza di picchi equispaziati nel tempo. I massimi della correlazione sono stati estratti blocco per blocco (8192 campioni per blocco) e analizzati in termini di ampiezza.

È stata osservata una variazione nell'ampiezza dei picchi a seconda della distanza del trasmettitore, coerente con il modello di attenuazione *Free Space Path Loss* (FSPL). Tuttavia, è stato anche riscontrato che il clock interno dei dispositivi Pluto può introdurre variazioni temporali e deriva di fase, influenzando la coerenza tra le acquisizioni e causando fluttuazioni nei massimi di correlazione.

Per ogni posizione sono stati calcolati(come illustrato nel Codice A.4):

- il valore **massimo** dei picchi di correlazione;

```
max_tot=max(val_max);
```

- il valore **medio** e **minimo** tra i picchi rilevati;

```
media_tot = mean(val_max); % media dei picchi
```

```
min_tot = min(val_max);
```

- una **media su 10 acquisizioni** per ciascuna distanza;

```
somma = somma + media_tot;
```

```
if mod(j, 10) == 0
```

```
    media = [media, somma / 10];
```

```
    somma = 0;
```

```
end
```

- la **normalizzazione** dei valori misurati per confronto con il modello teorico come dalla Figura 4.3.

```
xmedia = [20 40 60 80 100 120 140 160 180 200]; % distanze in cm
```

```
lambda = physconst('LightSpeed') / fc; % lunghezza d'onda
```

```
% Calcolo della FSPL per tutte le distanze
```

```
L_lin_t = [];
```

```
L_FSPL = [];
```

```

for i = 1:10
    d_m = xmedia(i) / 100; % converte cm in m
    L = (4 * pi * d_m / lambda)^2;
    L_FSPL = [L_FSPL, 10 * log10(L)];
    L_lin_t = [L_lin_t, (lambda / (4 * pi * d_m))^2];
end

% Normalizzazione per confronto con misure sperimentali
L_lin_t_norm = L_lin_t / max(L_lin_t);
ymedia_norm = ymedia / max(ymedia);

```

dove l'inverso della FSPL normalizzato indica il guadagno di trasferimento del canale in spazio libero

$$\frac{1}{\text{FSPL}} = \left(\frac{\lambda}{4\pi d} \right)^2 \quad (4.1)$$

Infine, le ampiezze medie ottenute sono state confrontate con la risposta teorica del modello FSPL, mostrando un buon accordo nella tendenza, ma con scostamenti legati alla riflessione ambientale e al rumore.

Nella tabella 4.1 si trovano le attenuazioni medie dei picchi rispetto a diverse posizioni del trasmettitore.

Tabella 4.1: Medie e varianze dei picchi di correlazione

Distanza	Media	Varianza
20cm	309,16	38,68
40cm	78,19	9,014
60cm	71,60	10,50
80cm	21,43	3,16
100cm	17,52	4,07
100cm sinistra	21,02	7,98
100cm destra	20,34	3,09
120cm	10,01	2,75
140cm	11,59	2,61
160cm	31,56	7,61
180cm	10,42	2,87
200cm	8,13	3,88
200cm sinistra	12,38	2,28
200cm destra	5,68	1,62

Capitolo 5

Conclusioni

Nel corso di questa tesi è stato analizzato il comportamento del canale radio in scenari di propagazione in linea di vista (LoS), con particolare attenzione agli effetti di attenuazione e multipath. L'utilizzo combinato del dispositivo *ADALM-Pluto SDR* e dell'ambiente *MATLAB* ha permesso di condurre esperimenti in grado di caratterizzare, con buona precisione, la risposta impulsiva del canale mediante tecniche di correlazione incrociata.

L'impiego di un segnale *chirp* ripetuto ha mostrato di essere particolarmente efficace per stimare i ritardi di propagazione. Le ampiezze dei picchi di correlazione osservati alle varie distanze sono risultate coerenti, in media, con il modello teorico di attenuazione *Free Space Path Loss* (FSPL), confermando la validità dell'approccio adottato.

Un ulteriore obiettivo ha riguardato la possibilità di riconoscere se un segnale ricevuto provenisse dallo stesso trasmettitore, sfruttando le variazioni dell'attenuazione e della risposta del canale. Sebbene il sistema presenti alcune limitazioni - tra cui la risoluzione in tempo legata alla banda disponibile

e l'instabilità dei clock interni dei dispositivi - l'esperimento ha dimostrato la fattibilità di una forma base di autenticazione spaziale del trasmettitore, all'aumentare della distanza si trova difficile riconoscere il trasmettitore a causa degli effetti multipath.

Purtroppo, a causa delle limitazioni hardware e della presenza di rumore, non è stato possibile osservare in modo chiaro gli echi del segnale associabili a percorsi riflessi. Tuttavia, il metodo proposto si è rivelato promettente per futuri studi in ambienti più controllati o con dispositivi a maggiore risoluzione temporale.

In conclusione, l'approccio sperimentale sviluppato offre una base solida per l'analisi del canale radio e per l'implementazione di tecniche semplici di autenticazione, con potenziali applicazioni nei sistemi wireless sicuri, nella localizzazione e nello studio dell'ambiente propagativo.

Ringraziamenti

Desidero esprimere la mia sincera gratitudine a tutte le persone che hanno contribuito alla realizzazione di questa tesi.

In primo luogo, ringrazio il mio relatore, Prof. Stefano Tomasin, per la guida esperta, i preziosi consigli e il supporto costante lungo tutto il percorso di ricerca. Un ringraziamento speciale va anche al nostro Tutor di laboratorio Mattia Piana e ai colleghi del laboratorio, che mi hanno fornito assistenza con l'attrezzatura SDR e l'ambiente MATLAB.

Ringrazio la mia famiglia e gli amici, per la pazienza, l'incoraggiamento e la comprensione nei momenti di maggiore impegno e stress. La loro presenza è stata fonte di motivazione e conforto.

Infine, un grazie a tutti coloro che, anche indirettamente, hanno contribuito alla realizzazione di questo lavoro di tesi.

Grazie di cuore.

Appendice A

Codice sorgente MATLAB

```
% Impostazione della frequenza di campionamento
SamplingRate = 5e6; % Frequenza di campionamento in Hz (da 1 MHz fino a 61.44 MHz supportati)
fc = 5.8e9; % Frequenza centrale della trasmissione (carrier) in Hz

% ID del dispositivo PlutoSDR
% Inserisci qui il numero seriale ottenuto con findPlutoRadio, es: "sn:xxxxxxxx"
% Mantieni "usb:0" se usi un solo Pluto
idTX = 'usb:0'; % Identificatore del Pluto per la trasmissione

%% Generazione del segnale da trasmettere

fsin1 = 1.1e6; % Frequenza del primo tono sinusoidale (1.1 MHz)
fsin2 = 1.5e6; % Frequenza del secondo tono sinusoidale (1.5 MHz)
t = (1:1:1e7) ./ SamplingRate; % Asse temporale per 10 milioni di campioni

% Genera due sinusoidi complesse
sig1 = exp(1i * 2 * pi * fsin1 * t).'; % Trasposta per ottenere colonna
sig2 = exp(1i * 2 * pi * fsin2 * t).';
```

```

% Somma le due sinusoidi e amplifica di un fattore 2
sig = (sig1 + sig2) * 2;

% Normalizzazione dell'ampiezza del segnale per evitare clipping
txNorm = 0.5 * (sig) / max(abs(sig)); % Scalo in modo che la potenza massima s

%% Impostazione dell'oggetto trasmettitore

txPluto = sdrtx('Pluto', ...
    'RadioID', idTX, ... % ID del Pluto
    'CenterFrequency', fc, ... % Frequenza di portante (fc)
    'Gain', -15, ... % Guadagno TX (range valido: da -89 a
    'BasebandSampleRate', SamplingRate); % Frequenza di campionamento

%% Avvio trasmissione

transmitRepeat(txPluto, txNorm); % Trasmette il segnale in loop continuo

% release(txPluto); % Rilascia la risorsa (opzionale, da us

```

Listing A.1: Script base trasmettitore

```

% Impostazioni di base
SamplingRate = 5e6; % Frequenza di campionamento del ricevitore (5 MHz)
fc = 5.8e9; % Frequenza centrale di ricezione (5.8 GHz)

% Creazione dell'oggetto ricevitore PlutoSDR
rxPluto = sdrxx('Pluto', 'RadioID', "usb:0", ... % Identificativo del disposit
    'CenterFrequency', fc, ... % Imposta la frequenza centra
    'GainSource', 'Manual', ... % Usa un guadagno manuale

```

```

    'Gain', 40, ...           % Guadagno in dB
    'OutputDataType', 'single', ... % Tipo di dato di output
    'BasebandSampleRate', SamplingRate); % Frequenza di campionamento in band

% Acquisizione del segnale
tic;                         % Avvia il timer
[rxWave, rx_meta] = capture(rxPluto, 2e6); % Acquisisce 2 milioni di campioni
toc;                         % Stampa il tempo impiegato per la c

% I campioni acquisiti sono ora disponibili in rxWave

%% Visualizzazione del segnale ricevuto nel dominio temporale

N = length(rxWave);          % Numero totale di campioni ricevuti
t = (0:N-1) / SamplingRate;  % Asse temporale associato ai campioni

% Parte reale del segnale ricevuto
figure;
plot(t, real(rxWave));
xlabel('Tempo (s)');
ylabel('Ampiezza');
title('Segnale ricevuto - Parte reale');
grid on;

% Parte immaginaria del segnale ricevuto
figure;
plot(t, imag(rxWave));
xlabel('Tempo (s)');
ylabel('Ampiezza');
title('Segnale ricevuto - Parte immaginaria');

```

```

grid on;

%% Calcolo e visualizzazione dello spettro del segnale

X2 = fftshift(fft(rxWave, N) / N);           % Trasformata di Fourier centrata (no
X2_db = 20 * log10(abs(X2));                 % Magnitudo dello spettro in dB
db_min = -105;                               % Soglia minima per la visualizzazione
X2_dbm = max(X2_db, db_min);                 % Evita valori estremamente negativi

f2 = (-N/2:N/2-1) * SamplingRate / N;       % Asse delle frequenze centrato, in H

% Plot dello spettro centrato sulla frequenza portante
figure;
plot(f2/1e6 + fc/1e6, X2_dbm);              % Frequenza in MHz, centrata sulla ca
xlabel('Frequenza (MHz)');
ylabel('Ampiezza |dB|');
title('Spettro del segnale ricevuto centrato su fc');
grid on;

```

Listing A.2: Script base ricevitore

```

clear all; close all; clc;

%% Parametri di sistema
SamplingRate = 40e6;           % Frequenza di campionamento (40 MHz)
fc = 3e9;                       % Frequenza centrale (3 GHz)

% Specifica degli ID delle radio PlutoSDR
% Puoi usare 'findPlutoRadio' per ottenere i seriali esatti
% Esempio: idTX = 'sn:104473ae73930001f5ff0c009c682d8063';
idTX = 'usb:1';                 % ID trasmettitore (oppure 'usb:0')

```

```

idRX = 'usb:0';           % ID ricevitore

%% Generazione segnale chirp
Fs = SamplingRate;       % Alias per comodità
nschirp = 8192;         % Numero di campioni del chirp
t_end = (nschirp - 1) / Fs; % Durata del segnale
t = 0:(1/Fs):t_end;     % Vettore dei tempi

f1 = -20e6;              % Frequenza iniziale del chirp (-20 MHz)
f2 = 20e6;               % Frequenza finale del chirp (+20 MHz)

% Generazione chirp complesso
txsig = chirp(t, f1, t_end, f2, 'complex').';

%% Trasmissione del segnale

% Normalizzazione in ampiezza (massimo = 1)
txNorm = txsig / max(abs(txsig));

% Creazione oggetto trasmettitore Pluto
txPluto = sdrtx('Pluto', ...
    'RadiolD', idTX, ...
    'CenterFrequency', fc, ...
    'Gain', -3, ...           % Guadagno TX (da -89 a 0 dB)
    'BasebandSampleRate', Fs);

% Trasmissione continua del chirp normalizzato
transmitRepeat(txPluto, txNorm);
% release(txPluto); % Da usare se si vuole liberare manualmente la radio

```

```
%% Ricezione del segnale

pause(1); % Piccola pausa per sincronizzazione

% Creazione oggetto ricevitore Pluto
rxPluto = sdrx('Pluto', ...
    'RadiolD', idRX, ...
    'CenterFrequency', fc, ...
    'GainSource', 'Manual', ...
    'Gain', 30, ... % Guadagno RX in dB (da -3 a 71 dB)
    'OutputDataType', 'single', ...
    'BasebandSampleRate', Fs);

nSamples = 8e6; % Numero di campioni da acquisire (max 20e6)

% Acquisizione del segnale ricevuto
tic
[rxWave, rx_meta] = capture(rxPluto, nSamples);
toc

%% Salvataggio dati ricevuti

% Esempi di salvataggi per diverse distanze:
% save("misurazione20.mat"); % 20 cm
% save("misurazione30.mat"); % 30 cm
% ...
% save("misurazione130.mat"); % 200 cm, sinistra/destra

save("misurazione150.mat");
```

Listing A.3: *tx_rx_save.m*

```

clear all; close all; clc;

%% Analisi di tutte le misurazioni salvate

% Codice per plottare ogni singolo file (disattivato)
% for i=1:84
%     d = "2025-06-09";
%     filename = [d + i + ".mat"];
%     load(filename);
%     figure;
%     plot(abs(val_corr));
% end

somma = 0;           % variabile per media temporanea su 10 file
media = [];         % vettore delle medie (una ogni 10 misure)
v_max = [];        % valori massimi per ogni blocco
v_med = [];        % valori medi dei picchi
v_min = [];        % valori minimi dei picchi
tot_peaks=[];      % Array che conterrà tutti i max da tutte le misure

% Etichette delle varie condizioni di misura
indice = ["", "Line of sight 20cm", "Line of sight 40cm", ...
          "Line of sight 60cm", "Line of sight 80cm", ...
          "Line of sight 100cm", "Line of sight 120cm", ...
          "Line of sight 140cm", "Line of sight 160cm", ...
          "Line of sight 180cm", "Line of sight 200cm", ...

```

```

        "Line of sight 200cm sx", "Line of sight 200cm dx", ...
        "Line of sight 100cm pos sx", "Line of sight 100cm pos dx"];

% Loop su tutti i file da 11 a 150
for j = 11:150
    j1 = j - 1;

    % Caricamento del file ricevuto
    load("misurazione" + j + ".mat");

    % Correlazione tra segnale ricevuto e trasmesso
    [corr_out, lags] = xcorr(rxWave, txsig);
    val_corr = abs(corr_out(nSamples:end));

    % Estrazione dei massimi di correlazione per ogni blocco di chirp
    indici_max = trovaMax(val_corr, nschirp);
    n_blocchi = floor(length(val_corr) / nschirp);
    val_max = zeros(1, n_blocchi);

    for i = 1:n_blocchi
        val_max(i) = val_corr(indici_max(i));
    end

    % Statistiche di correlazione
    max_tot = max(val_max); % massimo assoluto
    media_tot = mean(val_max); % media dei picchi
    min_tot = min(val_max); % minimo tra i picchi

    somma = somma + media_tot; % somma parziale per medie a gruppi di 10

```

```

% Salvataggio dei valori statistici
v_min = [v_min, min_tot];
v_med = [v_med, media_tot];
v_max = [v_max, max_tot];
tot_peaks=[tot_peaks val_max];

% Asse temporale per il plot
t_totchirp = (0:nSamples-1) / Fs;

% Etichette da mostrare nel grafico
message1 = "valore max dei picchi = " + max_tot;
message2 = "valore medio dei picchi = " + media_tot;
message3 = "valore minimo dei picchi = " + min_tot;

% Plot della correlazione con linee di riferimento
figure;
plot(t_totchirp, val_corr);
yline(max_tot, 'r—', message1);
yline(media_tot, 'g—', message2);
yline(min_tot, 'm—', message3);
xlabel('Tempo(s)');
ylabel('Ampiezza della correlazione');
title(indice(floor(j1 / 10) + 1));
ylim([0 max_tot * 1.1])

% Salvataggio grafico come immagine
saveas(gcf, indice(floor(j1 / 10) + 1) + "-" + j + ".png");
close(gcf);

% Calcolo media ogni 10 file

```

```

    if mod(j, 10) == 0
        media = [media, somma / 10];
        somma = 0;
    end
end

%% Analisi della curva media in funzione della distanza

ymedia = media(1:10); % valori medi (uno ogni 10 misure)
xmedia = [20 40 60 80 100 120 140 160 180 200]; % distanze in cm

figure;
plot(xmedia, ymedia);
xlabel('distanza (cm)');
ylabel('Valore della Correlazione media');

%% Calcolo della FSPL teorica

lambda = physconst('LightSpeed') / fc; % lunghezza d'onda
% R = 2; % distanza in metri
% L_lin = (4 * pi * R / lambda)^2; % FSPL lineare
% L_dB = 10 * log10(L_lin); % FSPL in dB

% Calcolo della FSPL per tutte le distanze
L_lin_t = [];
L_FSPL = [];

for i = 1:10

```

```

    d_m = xmedia(i) / 100; % converte cm in m
    L = (4 * pi * d_m / lambda)^2;
    L_FSPL = [L_FSPL, 10 * log10(L)];
    L_lin_t = [L_lin_t, (lambda / (4 * pi * d_m))^2];
end

% Plot della FSPL in forma lineare
figure;
plot(xmedia, L_lin_t);

% Normalizzazione per confronto con misure sperimentali
L_lin_t_norm = L_lin_t / max(L_lin_t);
ymedia_norm = ymedia / max(ymedia);

figure;
plot(xmedia, ymedia_norm, "black");
hold on;
plot(xmedia, L_lin_t_norm, "red");
xlabel('distanza (cm)');
legend('Correlazione media', 'FSPL teorico normalizzato');
grid on;
title("Confronto tra misura e modello FSPL");

%% Visualizzazione separata dei gruppi di 9760 picchi

% Ogni blocco rappresenta 10 misurazioni (976 chirp per misura  $\tilde{A}$  10 da cui
% si riesce a visualizzare tutte la variazione delle attenuazioni in ogni
% posizione)
for i = 1:14
    n1 = 9760 * (i - 1) + 1;

```

```

n2 = 9760 * i;
max_tot = max(tot_peaks(n1:n2));    % massimo assoluto
media_tot = mean(tot_peaks(n1:n2)); % media dei picchi
min_tot = min(tot_peaks(n1:n2));    % minimo tra i picchi
message1 = "valore max dei picchi = " + max_tot;
message2 = "valore medio dei picchi = " + media_tot;
message3 = "valore minimo dei picchi = " + min_tot;
figure;
plot(tot_peaks(n1:n2));
grid on;
xticks(0:976:9760);
xlabel('Indice_campione_(chirp)');
ylabel('Ampiezza_picco');
yline(max_tot, 'r—', message1);
yline(media_tot, 'g—', message2);
yline(min_tot, 'm—', message3);
title("Blocchi di picchi di correlazioni: Misure " + indice(i+1));
saveas(gcf, "Blocchi-"+indice(i+1) + ".png");
close(gcf);
end

```

Listing A.4: visualizza.m

Bibliografia

- [1] G. E. Z. Sylvie Picol, Gheorghe Zaharia, “Hardware simulator for mimo radio channels,” *IEEE*, 2005.
- [2] Wikipedia, “Channel state information,” *Wikipedia*, 2024.
- [3] N. Explained, “Multipath channel basics,” *NR Explained*, 2025.
- [4] Q. Chaudhari, “Channel estimation in wireless communication,” *WirelessPI.com blog*, 2022 aprile.
- [5] D. P. A. W. T. Collins, R. Getz, “Signals and systems,” in *Software-Defined Radio for Engineers*, pp. 19–85, Analog Devices, 2018.
- [6] P. V. D. Tse, “Fundamentals of wireless communication,” ambridge University Press, 2005.
- [7] Wikipedia contributors, “Free-space path loss — Wikipedia, The Free Encyclopedia.” https://en.wikipedia.org/wiki/Free-space_path_loss, 2024. [Online; accessed 15-July-2025].
- [8] S. m. Kay, “Neyman-pearson theorem,” in *fundamentals of statistical signal processing detection theory*, vol. 2, pp. 61–67, 1998.

- [9] A. Kumar, “Neyman-pearson lemma: Hypothesis test, examples,” *Analytics Yogi*, 2023.
- [10] I. P. Magazine, “Low-cost sdr for education: Adalm-pluto,” *Proc. IEEE International Symposium on Information Theory*, 2023 settembre/ottobre.
- [11] MathWorks, “Adalm-pluto radio support from communications toolbox,” MathWorks.
- [12] “Sdrangel - open-source tx & rx software defined radio.” <https://www.sdrangel.org/>, 2025. Accessed: 2025-07-12.