



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

COMUNICAZIONE A SINGOLA PORTANTE CON
DISPOSITIVI ADALM-PLUTO

Relatore:
Prof. Stefano Tomasin

Laureando:
Gianluca Caregnato

ANNO ACCADEMICO: 2023/2024

Data di laurea: 25/09/2024

Indice

Introduzione	1
1 Comunicazione a singola portante	5
1.1 Introduzione	5
1.2 Trasmissione Digitale	6
1.2.1 Trasmissione	7
1.2.2 Ricezione	9
2 ADALM-PLUTO e MATLAB Simulink	17
2.1 Introduzione	17
2.2 Caratteristiche principali del PlutoSDR	18
2.3 Architettura del sistema PlutoSDR	18
2.4 MATLAB Simulink	19
2.4.1 Caratteristiche principali di MATLAB Simulink	19
2.5 Applicazioni pratiche	20
2.6 Configurazione e utilizzo	21
2.7 Conclusione	22
3 Trasmissione QAM con ADALM-PLUTO	23
3.1 Introduzione	23
3.2 Generatore di Bit (Bits Generation)	24
3.2.1 Codice di Barker (Barker code)	25
3.2.2 Generazione messaggio	25
3.2.3 Scrambler	26
3.2.4 Composizione frame	27
3.3 Modulatore 4-QAM	27
3.4 Coseno Rialzato Radice Quadrata (Square Root Raised Cosine)	28
3.5 Trasmettitore ADALM-PLUTO	29
3.6 Conclusione	30

4	Ricezione QAM con ADALM-PLUTO	31
4.1	Introduzione	31
4.2	Ricevitore ADALM-PLUTO	32
4.3	Sincronizzazione e decodifica simboli QAM	32
4.3.1	Regolatore automatico del guadagno (Automatic gain controller)	32
4.3.2	Filtro ricevitore a coseno rialzato (Raised cosine receiver filter)	33
4.3.3	Compensatore di frequenza (Coarse frequency compensation)	35
4.3.4	Sincronizzazione simboli (Symbol synchronizer)	36
4.3.5	Sincronizzazione portante (Carrier synchronizer)	37
4.3.6	Sincronizzazione frame (Frame Synchronizer)	39
4.3.7	Decodifica dati (Data decoding)	40
4.3.8	Stampa del messaggio	42
4.3.9	Tasso di errore	43
4.4	Conclusione	43
5	Esperimenti	45
5.1	Introduzione	45
5.2	Analisi a diverse distanze e ostacoli	46
5.2.1	Considerazioni sui dati	48
5.3	Distanza Minima dei Simboli della Costellazione	49
5.3.1	Considerazioni sui dati	51
6	Conclusioni	53
	Bibliografia	56

Sommario

Questa tesi si concentra sull'analisi e implementazione di un sistema di trasmissione e ricezione di segnali digitali modulati in 4-QAM, utilizzando il dispositivo SDR ADALM-PLUTO in ambiente MATLAB Simulink. Verranno esaminati i principali blocchi funzionali del sistema di comunicazione digitale, con una descrizione del processo di modulazione, trasmissione e ricezione del segnale.

In particolare, saranno trattati i vari passaggi che trasformano un flusso di bit in un segnale analogico trasmissibile e viceversa, affrontando le sfide legate alla sincronizzazione, alla demodulazione e alla correzione degli errori. La tesi include anche un'analisi sperimentale che studia l'influenza di fattori esterni, come la distanza e la presenza di ostacoli sul tasso di errore della trasmissione.

Saranno inoltre discussi i risultati degli esperimenti che mettono in evidenza l'effetto dell'incremento della distanza minima tra i simboli sulla qualità complessiva della comunicazione. Infine, verranno presentate le conclusioni basate sui dati ottenuti.

Introduzione

Questa tesi ha l'obiettivo di analizzare il processo di trasmissione e ricezione di segnali digitali modulati in 4-QAM utilizzando la piattaforma SDR ADALM-PLUTO in ambiente MATLAB Simulink. Verranno presentati i principali blocchi funzionali che costituiscono un sistema di comunicazione digitale, con particolare attenzione agli aspetti pratici implementati e simulati in MATLAB.

L'analisi si concentra sulla qualità della trasmissione in scenari reali, includendo la gestione degli errori, la sincronizzazione e la compensazione degli effetti di disturbi e ostacoli presenti lungo il percorso del segnale. Verranno inoltre condotti esperimenti che metteranno in evidenza come vari fattori ambientali, quali la distanza e la presenza di ostacoli, influenzino il tasso di errore. Sarà anche analizzato l'impatto dell'aumento della distanza minima tra i simboli della costellazione sulla qualità del segnale ricevuto.

Nel Capitolo 1 tratteremo le basi della comunicazione digitale a singola portante. Verrà discusso il processo di modulazione digitale e i vari passaggi necessari per trasformare un segnale digitale in un segnale trasmissibile attraverso un canale di comunicazione. Inoltre, si analizzeranno le sfide principali legate alla ricezione, in particolare la ricostruzione del segnale originale e la

gestione degli errori introdotti dal canale.

Nel Capitolo 2 verranno introdotti i dispositivi ADALM-PLUTO e MATLAB Simulink, descrivendone le caratteristiche principali e il loro utilizzo come piattaforma per lo sviluppo e l'implementazione di sistemi di comunicazione digitale. Si discuteranno le potenzialità di queste tecnologie e il loro ruolo nella progettazione e simulazione di sistemi di trasmissione e ricezione wireless.

Il Capitolo 3 si concentrerà sulla trasmissione di segnali modulati QAM tramite ADALM-PLUTO. Verranno descritti in dettaglio la generazione di bit, l'uso del codice di Barker per la sincronizzazione e la modulazione 4-QAM. Inoltre, sarà analizzato l'uso del filtro SRRC per migliorare la qualità del segnale e minimizzare l'interferenza intersimbolica, nonché il processo di trasmissione attraverso il dispositivo ADALM-PLUTO.

Nel Capitolo 4 esploreremo il processo di ricezione del segnale QAM utilizzando ADALM-PLUTO. Verranno analizzati i principali blocchi funzionali della ricezione, come il regolatore automatico del guadagno, il filtro a coseno rialzato e i vari meccanismi di sincronizzazione dei simboli e della portante. Verranno esaminati anche i processi di decodifica e descrambling del messaggio ricevuto.

Nel Capitolo 5 verranno presentati gli esperimenti condotti per valutare l'efficacia del sistema di comunicazione. Saranno esaminati i risultati ottenuti variando la distanza tra i dispositivi e introducendo ostacoli, misurando il tasso di errore risultante. Inoltre, si studierà l'impatto dell'incremento della distanza minima tra i simboli della costellazione sulla qualità della comunicazione.

Nel Capitolo 6 vengono riepilogati i risultati principali ottenuti nella tesi, con particolare attenzione agli esperimenti condotti. Si tratta di un capitolo conclusivo che riassume tutto il percorso svolto in questa tesi per poi terminare con delle considerazioni finali.

Capitolo 1

Comunicazione a singola portante

1.1 Introduzione

In questo capitolo discuteremo delle tecniche di comunicazione a singola portante e della trasmissione digitale. Esploreremo la trasmissione digitale, dove l'informazione viene trasformata in un vettore di bit per essere poi modulata in un segnale trasmissibile attraverso un canale di comunicazione. Per poi concludere con la fase di ricezione, la fase più delicata. Mostreremo come dal segnale analogico trasmesso riusciamo a tornare al vettore di simboli trasmesso e come dai simboli si riesca ad avere nuovamente l'informazione trasmessa con il minor numero di errori possibili.

La discussione sarà strutturata come segue:

- **Trasmissione Digitale:**
 - Processo di modulazione digitale
 - Descrizione del segnale trasmesso e ricevuto

– Problemi introdotti dal disturbo del canale di comunicazione

• **Fasi della Trasmissione:**

– **Trasmissione:**

* Mappatura dei bit (BMAP)

* Modulazione digitale (Digital Modulator)

– **Ricezione:**

* Proiezione del segnale (Signal Projection)

* Rilevatore (Detector)

* Mappatura inversa dei bit (IBMAP)

Attraverso questa struttura, comprenderemo come i dati vengono preparati, modulati, trasmessi e infine ricevuti e demodulati, analizzando le sfide e le tecniche utilizzate per garantire l'integrità dell'informazione nonostante i disturbi introdotti dai canali di comunicazione. Questa panoramica ci fornirà una solida base per comprendere gli esperimenti pratici e le applicazioni con dispositivi ADALM-PLUTO.

1.2 Trasmissione Digitale

Si parla di trasmissione digitale quando l'informazione che viene generata, prima di essere trasmessa, è un vettore di bit $\langle b_l \rangle$. Il compito della modulazione digitale è quello di trasformare tutta l'informazione in un segnale analogico.

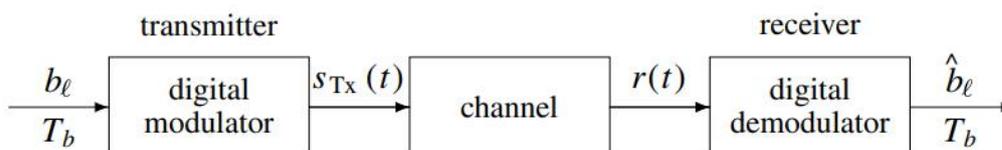


Figura 1.1: Schema trasmissione digitale [1]

Il segnale generato $S_{Tx}(t)$ deve poter essere trasmesso da una trasmittente in un canale di comunicazione, il quale inevitabilmente andrà a introdurre un disturbo nella comunicazione, per poi essere ricevuto dalla ricevente.

Il segnale ricevuto, che chiameremo $r(t)$, a questo punto viene demodolato generando un vettore $\langle \hat{b}_l \rangle$. A causa del disturbo introdotto dal canale potrebbe essere che il vettore ricevuto sia affetto da disturbo e quindi $b_l \neq \hat{b}_l$ per qualche valore di l .

1.2.1 Trasmissione

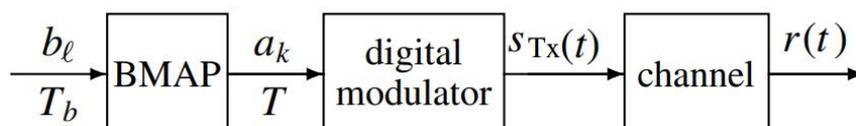


Figura 1.2: Schema trasmettitore [1]

Mappatura dei bit (BMAP)

Nel processo di trasmissione digitale il primo passo è generare una sequenza di bit a partire dall'informazione che desideriamo inviare, come un'email, una foto o una registrazione audio. Questa sequenza di bit deve essere trasformata in una forma che possa essere fisicamente trasmessa attraverso un

canale di comunicazione, come l'aria o un cavo. Questo processo è chiamato modulazione.

Il blocco di Mappatura dei Bit (BMAP) (vedi Figura 3.1) svolge un ruolo cruciale in questa trasformazione. Il BMAP converte i singoli bit b_ℓ in simboli a_k di lunghezza $\log_2 M$, dove M rappresenta l'ordine di grandezza della modulazione. Ad esempio, se M è 16, ogni simbolo sarà costituito da 4 bit ($\log_2 16 = 4$).

Questa conversione serve per preparare i dati alla modulazione digitale. I simboli generati dal blocco BMAP vengono quindi trasferiti al modulatore digitale, che li trasforma in un segnale $s_{TX}(t)$ pronto per essere trasmesso attraverso il canale di comunicazione.

Modulazione digitale (digital modulator)

La modulazione può essere considerata come il processo di traduzione del nostro messaggio in un linguaggio che il canale di comunicazione possa comprendere e trasportare.

Durante la modulazione i bit del nostro messaggio vengono convertiti in un segnale analogico $S_{Tx}(t)$. Questo segnale analogico è un'onda continua che può viaggiare agevolmente attraverso il canale di comunicazione, nel nostro caso, trattandosi di una trasmissione radio, il segnale viaggerà attraverso l'aria, generando quindi un'onda che si propaga dall'antenna della nostra trasmittente contenente il messaggio.

Tra le modulazioni digitali più importanti troviamo:

- **PAM (Pulse Amplitude Modulation)**

- **QAM (Quadrature Amplitude Modulation)**
- **PSK (Phase Shift Keying)**
- **FSK (Frequency Shift Keying)**

Nel nostro esperimento andremo ad utilizzare una modulazione QAM, più nello specifico una 4-QAM, una tecnica in cui i dati sono rappresentati attraverso quattro diversi stati di fase e ampiezza del segnale. In una 4-QAM, ogni simbolo codifica due bit di informazione, combinando due livelli di ampiezza e due fasi diverse per creare un segnale che può rappresentare quattro possibili stati distinti, come si può vedere dal grafico della costellazione (vedi Figura 1.3).

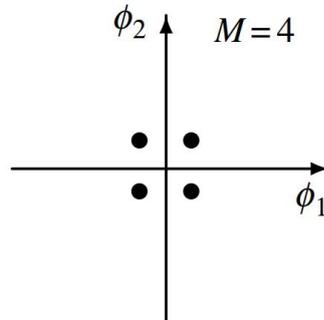


Figura 1.3: Costellazione 4-QAM [1]

1.2.2 Ricezione

Dopo aver generato l'informazione in bit, modulata in un segnale compatibile con un canale di comunicazione ed aver iniziato a trasmettere bisogna riuscire a ricevere le informazioni e demodularle.

Se non considerassimo i rumori introdotti dal canale, per la ricezione sarebbe sufficiente demodulare il segnale in ingresso al ricevitore e decodificare i bit ricevuti, restituendo così esattamente il messaggio inviato a chi sta ricevendo i dati. Purtroppo un canale ideale (che non introduce rumore) non esiste, avremo a che fare quindi con un segnale che potrebbe presentare degli errori in ricezione.

Definiamo il guadagno del canale (channel gain) come C , il rumore introdotto dal canale come $w(t)$, $S_{Tx}(t)$ che abbiamo già introdotto essendo il segnale prodotto dalla trasmittente ed otteniamo $r(t)$, ovvero il segnale ricevuto in ingresso al ricevitore, come:

$$r(t) = C * S_{Tx}(t) + w(t) \quad (1.1)$$

Abbiamo quindi ora un segnale affetto da disturbo che deve essere demodulato.

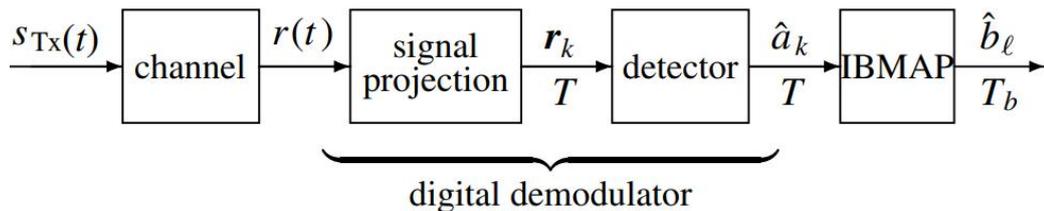


Figura 1.4: Schema del trasmettitore [1]

Proiezione del segnale (signal projection)

La proiezione del segnale (signal projection) è una tecnica utilizzata nei demodulatori, questa serve per elaborare una sequenza di simboli r_k a partire

da un segnale analogico continuo $r(t)$.

In presenza di distorsioni del canale viene utilizzata una nuova base $\tilde{\phi}_i(t)$ al posto di $\phi_i(t)$ per garantire una corretta demodulazione.

In pratica il segnale analogico viene proiettato sulle forme d'onda della base $\{\phi_i(t)\}$ per ottenere i componenti $r_{k,i}$ che costituiscono il vettore di proiezione \mathbf{r}_k .

Questi componenti sono calcolati come il prodotto interno tra il segnale ricevuto $r(t)$ e le forme d'onda traslate $\phi_i(t - kT)$, ossia:

$$r_{k,i} = \langle r(t), \phi_i(t - kT) \rangle. \quad (1.2)$$

I valori $r_{k,i}$ rappresentano la proiezione del segnale analogico sulle forme d'onda della base, permettendo al demodulatore di ricostruire la sequenza di simboli trasmessi.

Nel caso in cui ci fossero delle distorsioni introdotte dal canale, in alcuni casi le forme d'onda base vengono modellate in modo tale da riuscire ad adattarsi alla distorsione presente, questo permette una demodulazione più affidabile in quanto riesce ad operare anche quando ci sono delle interferenze.

Rilevatore (detector)

La seconda e ultima operazione del demodulatore, che esamineremo in dettaglio, è cruciale per il corretto funzionamento del sistema di comunicazione. Questa fase prevede la decisione su quale simbolo sia stato effettivamente trasmesso. Dopo aver ricevuto il segnale modulato e averlo demodulato, il

demodulatore deve analizzare i dati in ingresso per determinare con precisione quale simbolo, tra tutti quelli possibili, è stato inviato dall'emittente. Questa operazione è fondamentale perché da essa dipende l'accuratezza della trasmissione dei dati e, di conseguenza, la qualità complessiva della comunicazione.

Per decidere quale simbolo \hat{a}_k è stato trasmesso osservando il segnale ricevuto ρ_k , il demodulatore può adottare uno dei seguenti criteri: Maximum A Posteriori (MAP), Maximum Likelihood (ML) o Minimum Distance, come descritto dettagliatamente di seguito.

$$\hat{a}_k = \begin{cases} \operatorname{argmax}_{n \in \{1, \dots, M\}} p_w(\rho_k - \tilde{s}_n) p_n, & \text{MAP} \\ \operatorname{argmax}_{n \in \{1, \dots, M\}} p_w(\rho_k - \tilde{s}_n), & \text{ML} \\ \operatorname{argmin}_{n \in \{1, \dots, M\}} \|\rho_k - \tilde{s}_n\|^2, & \text{minimum distance} \end{cases} \quad (1.3)$$

dove:

- ρ_k rappresenta il segnale ricevuto,
 - \tilde{s}_n rappresenta il segnale stimato,
 - p_w rappresenta la funzione di densità di probabilità del rumore,
 - p_n rappresenta la probabilità a priori del segnale trasmesso.
- k rappresenta l'indice temporale del segnale ricevuto,

– n rappresenta l'indice dei simboli della costellazione.

L'insieme dei simboli della costellazione, indicato con $\{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_M\}$, rappresenta tutti i possibili simboli che possono essere trasmessi nel sistema di comunicazione. Ogni simbolo \tilde{s}_n appartiene a questo insieme, dove M è il numero totale dei simboli della costellazione.

MAP (Maximum A Posteriori Criterion)

Il criterio del massimo a posteriori (Maximum A Posteriori MAP) nella demodulazione viene utilizzato per determinare la stima più probabile di un segnale trasmesso dato un segnale ricevuto e alcune informazioni a priori. In altre parole, il criterio MAP cerca di massimizzare la probabilità a posteriori $p_w(\rho_k - \tilde{s}_n)p_n$, dove:

Il processo consiste nel trovare il valore di n che massimizza questa probabilità, cioè:

$$\operatorname{argmax}_{n \in \{1, \dots, M\}} p_w(\rho_k - \tilde{s}_n)p_n \quad (1.4)$$

Ciò significa che tra tutte le possibili stime \tilde{s}_n , si sceglie quella che rende massima la probabilità a posteriori del segnale trasmesso dato il segnale ricevuto. Questo approccio tiene conto sia delle osservazioni effettive (il segnale ricevuto) sia delle informazioni a priori (probabilità a priori del segnale trasmesso), risultando in una stima più robusta rispetto ad altre tecniche che non utilizzano informazioni a priori.

ML (Maximum Likelihood Criterion)

Il criterio della massima verosimiglianza (Maximum Likelihood ML) nella demodulazione è utilizzato per stimare il segnale trasmesso basandosi esclusivamente sul segnale ricevuto, senza considerare informazioni a priori. In altre parole, il criterio ML cerca di massimizzare la funzione di verosimiglianza $p_w(\rho_k - \tilde{s}_n)$.

Il processo consiste nel trovare il valore di n che massimizza questa funzione di verosimiglianza, cioè:

$$\operatorname{argmax}_{n \in \{1, \dots, M\}} p_w(\rho_k - \tilde{s}_n) \quad (1.5)$$

Questo significa che tra tutte le possibili stime \tilde{s}_n , si sceglie quella che rende massima la probabilità del segnale ricevuto data la stima del segnale trasmesso. A differenza del criterio MAP, il criterio ML non utilizza informazioni a priori sulla probabilità del segnale trasmesso, basandosi unicamente sulle osservazioni effettive.

Minimum Distance Criterion

Il criterio della minima distanza nella demodulazione è utilizzato per determinare la stima del segnale trasmesso minimizzando la distanza euclidea tra il segnale ricevuto e le possibili stime del segnale. In altre parole, il criterio a minima distanza cerca di minimizzare la quantità $\|\rho_k - \tilde{s}_n\|^2$.

Il processo consiste nel trovare il valore di n che minimizza questa distanza euclidea, cioè:

$$\operatorname{argmin}_{n \in \{1, \dots, M\}} \|\rho_k - \tilde{s}_n\|^2 \quad (1.6)$$

Questo significa che tra tutte le possibili stime \tilde{s}_n , si sceglie quella che rende minima la distanza tra il segnale ricevuto e il segnale stimato. A differenza dei criteri MAP e ML, il criterio a minima distanza non utilizza informazioni probabilistiche, basandosi esclusivamente su una misura di distanza.

Confronto tra i criteri MAP, ML e Minimum Distance: I tre criteri MAP, ML e a minima distanza differiscono nel modo in cui stimano il segnale trasmesso:

- **MAP** (Maximum A Posteriori): Massimizza la probabilità a posteriori del segnale trasmesso, utilizzando sia le osservazioni effettive (segnale ricevuto) che le informazioni a priori (probabilità a priori del segnale trasmesso). Questo criterio è ottimale quando si dispone di informazioni a priori.
- **ML** (Maximum Likelihood): Massimizza la funzione di verosimiglianza del segnale ricevuto data la stima del segnale trasmesso. Non utilizza informazioni a priori, basandosi solo sulle osservazioni effettive. È ottimo quando le informazioni a priori non sono disponibili o non sono affidabili.

- **Minimum Distance:** Minimizza la distanza euclidea tra il segnale ricevuto e le possibili stime del segnale trasmesso. Non utilizza informazioni probabilistiche, basandosi esclusivamente su una misura di distanza. È semplice da implementare ed è ottimo quando il rumore è gaussiano.

Mappatura inversa dei bit (IBMAP)

Il blocco Inverse Bit Mapping (IBMAP) svolge il ruolo inverso rispetto al blocco BMAP. Durante il processo di ricezione, il segnale ricevuto $r(t)$ viene demodulato e convertito in simboli stimati \hat{a}_k . Questi simboli passano poi attraverso il blocco IBBMAP che li riconverte in sequenze di bit \hat{b}_ℓ . Questa riconversione serve a recuperare l'informazione originaria.

Tuttavia, come accennato in precedenza, il canale di comunicazione può introdurre errori nel segnale trasmesso a causa di interferenze, rumore o distorsioni. Di conseguenza, i simboli \hat{a}_k potrebbero non corrispondere esattamente ai simboli a_k trasmessi. Questo errore nei simboli può influire negativamente sul processo di IBBMAP, portando a errori nei bit decodificati \hat{b}_ℓ , compromettendo quindi l'integrità dell'informazione recuperata.

Se il simbolo a_k non è stato correttamente ricevuto a causa degli errori introdotti dal canale, c'è una probabilità che i bit decodificati \hat{b}_ℓ risultino errati. Tuttavia, questo dipende dal tipo e dall'entità dell'errore. Se l'errore introdotto nel simbolo è minore o può essere corretto mediante tecniche di codifica e decodifica, come i codici di correzione degli errori, allora i bit \hat{b}_ℓ potrebbero essere recuperati correttamente.

Capitolo 2

ADALM-PLUTO e MATLAB

Simulink

2.1 Introduzione

In questo capitolo, introdurremo e descriveremo le radio ADALM-PLUTO, dispositivi che, come vedremo nei prossimi capitoli, abbiamo utilizzato per la nostra analisi in MATLAB Simulink.

Il dispositivo ADALM-PLUTO, noto anche come PlutoSDR, è una piattaforma di sviluppo versatile e accessibile in quanto è una radio definita dal software (Software Defined Radio, SDR). Il PlutoSDR, sviluppato da Analog Devices, è un dispositivo di lavoro utilizzato per scopi sperimentali ed educativi, che aiuta gli utenti a imparare e realizzare qualsiasi genere di applicazione per la comunicazione wireless. Inoltre, introdurremo MATLAB Simulink, un ambiente di simulazione e modellazione multidominio e che è stato utilizzato con PlutoSDR durante il nostro esperimento.

2.2 Caratteristiche principali del PlutoSDR

Il PlutoSDR offre diverse caratteristiche che lo rendono una scelta eccellente per lo sviluppo di nuove soluzioni radio.

- **Frequenza operativa:** Il dispositivo copre un'ampia gamma di frequenze, da 325 MHz a 3.8 GHz, permettendo la sperimentazione su numerose bande di frequenza.
- **Larghezza di banda:** Supporta una larghezza di banda di trasmissione fino a 20 MHz.
- **Compatibilità software:** Il PlutoSDR è compatibile con diverse piattaforme software, tra cui MATLAB, Simulink, e la libreria libio di Analog Devices.
- **Flessibilità:** Grazie al suo firmware open-source, è possibile personalizzare il comportamento del dispositivo secondo le necessità specifiche del progetto.

2.3 Architettura del sistema PlutoSDR

L'architettura del PlutoSDR si basa su un design integrato che include [2]:

- **Transceiver RF:** Il cuore del dispositivo è il transceiver AD9363, che gestisce le operazioni di trasmissione e ricezione dei segnali RF.
- **FPGA:** Un chip Xilinx Zynq-Z7010 funge da piattaforma di elaborazione, integrando un processore ARM Cortex-A9 dual-core e una matri-

ce FPGA (Field-Programmable Gate Array: Matrice di Porte Logiche Programmabile).

- **Memoria:** Il dispositivo include 512 MB di RAM DDR3 e 32 MB di memoria flash QSPI (Quad Serial Peripheral Interface: Interfaccia Periferica Seriale Quadrupla) per la memorizzazione del firmware.
- **Interfacce di I/O:** Oltre alla porta USB (Universal Serial Bus), il PlutoSDR dispone di connettori SMA (SubMiniature version A) per l'antenna di trasmissione e ricezione.

2.4 MATLAB Simulink

MATLAB Simulink è un ambiente di simulazione e modellazione multidominio che offre una piattaforma per progettare, simulare e testare sistemi dinamici complessi. È particolarmente utile per l'analisi e la progettazione di sistemi di comunicazione, controlli, elaborazione del segnale e altro ancora.

2.4.1 Caratteristiche principali di MATLAB Simulink

MATLAB Simulink offre diverse caratteristiche che lo rendono un potente strumento per la simulazione e la modellazione multidominio. Una delle principali caratteristiche di Simulink è la sua interfaccia intuitiva che offre un ambiente grafico per la costruzione di modelli mediante diagrammi a blocchi. Questo approccio consente agli utenti di creare modelli complessi in modo visuale, facilitando la comprensione e la manipolazione dei sistemi.

Un'altra caratteristica importante di Simulink è la sua libreria di blocchi incorporata, che offre molte funzionalità matematiche, logiche, di controllo e di comunicazione. I blocchi aiutano gli utenti a costruire dinamicamente gli elementi necessari per modellare i propri sistemi nel minor tempo possibile.

Un altro punto forte di Simulink è il suo collegamento con MATLAB. È possibile utilizzare i codici MATLAB per simulazione post-elaborazione e pre-elaborazione dei dati. Una tale interfaccia compatta tra MATLAB e Simulink aiuta a sfruttare al massimo l'efficienza di questi strumenti.

Infine, Simulink ha una buona compatibilità hardware, in quanto può funzionare bene con molti dispositivi, inclusi i dispositivi SDR come PlutoSDR. Ciò consente agli utenti di interfacciarsi direttamente con l'hardware per condurre esperimenti reali, testare modelli e acquisire dati dal mondo esterno. Le caratteristiche menzionate sopra rendono MATLAB Simulink un ambiente complesso ed estensibile che consente di progettare, simulare e analizzare sistemi in tutte le discipline ingegneristiche.

2.5 Applicazioni pratiche

Il PlutoSDR, in combinazione con MATLAB Simulink, trova applicazione in numerosi ambiti pratici. In particolare, rappresenta uno strumento didattico ideale per i corsi di ingegneria delle telecomunicazioni, dove permette agli studenti di acquisire esperienza pratica con le diverse tecniche di trasmissione e ricezione. Grazie alla sua versatilità e alla facilità d'uso, gli studenti possono esplorare e comprendere meglio i concetti teorici attraverso esperimenti concreti e simulazioni realistiche.

Oltre all'ambito educativo, il PlutoSDR viene utilizzato nella ricerca. In questo contesto, è impiegato per la prototipazione e il test di nuovi algoritmi di comunicazione radio. La possibilità di simulare e verificare rapidamente le idee in un ambiente controllato consente ai ricercatori un ottimo riscontro pratico.

La combinazione di PlutoSDR e MATLAB Simulink offre quindi un potente strumento per affrontare sfide complesse sia in ambito accademico che in quello della ricerca avanzata, promuovendo l'innovazione e l'apprendimento pratico.

2.6 Configurazione e utilizzo

La configurazione del PlutoSDR con MATLAB Simulink è semplice e intuitiva:

1. **Collegamento al PC:** Il primo passo è collegare il dispositivo al computer. I driver necessari per il corretto funzionamento dovrebbero installarsi automaticamente. Se ciò non accade, è possibile scaricarli manualmente dal sito di Analog Devices.
2. **Configurazione iniziale:** Dopo aver scaricato i driver, è necessario procurarsi un programma che permetta di interfacciarsi con ADALM-PLUTO per creare ed eseguire script. In questo caso installeremo MATLAB Simulink.
3. **MATLAB Simulink:** Una volta scaricato MATLAB Simulink, bisogna installare il pacchetto specifico per le comunicazioni radio con

PlutoSDR. Nella sezione *Home -> Add-Ons -> Get Hardware Support Packages*, cerchiamo il pacchetto *Communications System Toolbox Support Package for Analog Devices ADALM-PLUTO Radio*.

Dopo l'installazione, uno strumento permetterà di verificare il corretto funzionamento di ogni radio e se il software installato nella radio necessita aggiornamenti.

2.7 Conclusione

In sintesi, il PlutoSDR è una piattaforma versatile e potente per la sperimentazione e l'educazione nelle comunicazioni wireless, e quando combinato con MATLAB Simulink, diventa un potente strumento per la progettazione, simulazione e analisi di sistemi di comunicazione complessi. Nei prossimi capitoli, vedremo come abbiamo utilizzato questi strumenti per il nostro esperimento.

Capitolo 3

Trasmissione QAM con ADALM-PLUTO

3.1 Introduzione

In questo capitolo andremo a studiare la trasmissione di simboli QAM con il dispositivo ADALM-PLUTO in Matlab Simulink. Lo schema complessivo del trasmettitore è riportato in Figura 3.1. Verrà introdotta, la generazione dei bit, includendo tecniche di sincronizzazione con l'uso del codice di Barker, che facilita la corretta interpretazione del messaggio trasmesso. Si prosegue con la descrizione del modulo di scrambler, il quale riduce l'interferenza tra i segnali, migliorando la robustezza della comunicazione. Successivamente, viene trattata la modulazione del segnale attraverso il modulatore 4-QAM, che converte i bit generati in simboli. Un altro elemento essenziale descritto è l'applicazione del filtro SRRC (Square Root Raised Cosine), che ottimizza la forma del segnale per la trasmissione, minimizzando l'interferenza

intersimbolica.

Infine, viene esaminato il ruolo del trasmettitore ADALM-PLUTO, che converte il segnale elaborato per la trasmissione in etere.

Questo capitolo fornisce una panoramica di ciascuno di questi processi, supportata da schemi e diagrammi che illustrano l'implementazione pratica del sistema di trasmissione QAM utilizzando dispositivi SDR.

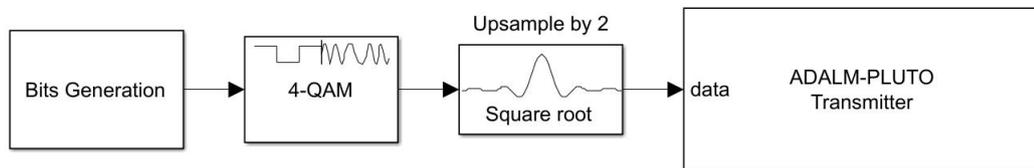


Figura 3.1: Trasmissione Matlab Simulink con PlutoSDR

3.2 Generatore di Bit (Bits Generation)

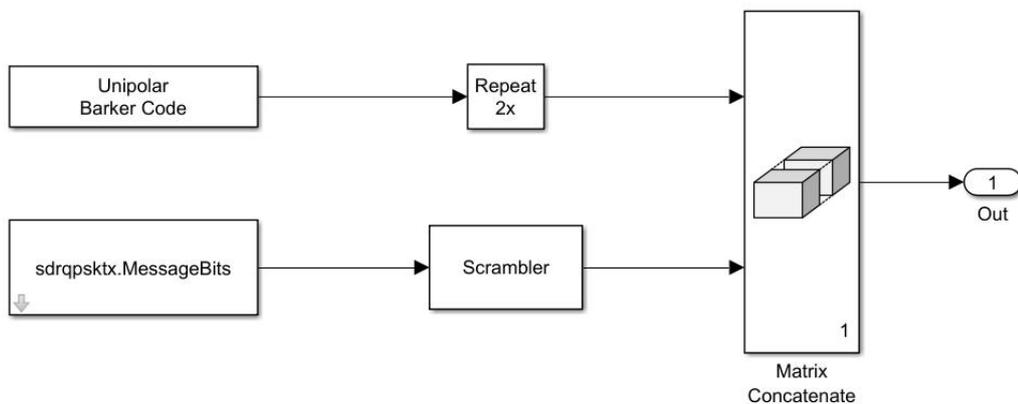


Figura 3.2: Generatore di Bit

Il primo passo per la trasmissione digitale è quello di generare un vettore di bit da trasmettere, andremo quindi a vedere come sarà composto il *frame* e quali tecniche sono state utilizzate per far in modo che si possa sincronizzare il messaggio riducendo al minimo le interferenze. Lo schema per la generazione del frame è riportato in Figura 3.2

3.2.1 Codice di Barker (Barker code)

Il codice di Barker è una sequenza di bit che ha un ruolo cruciale per la sincronizzazione dei messaggi. Il codice è conosciuto sia al trasmettitore che al ricevitore, questo quindi segnala al ricevitore che è iniziata la trasmissione del nuovo frame, permettendo anche, come vedremo più nel dettaglio nella sezione di sincronizzazione in ricezione, di fare una stima del ritardo così da andare a correggerlo.

In questo caso, il codice scelto di Barker, è una sequenza di 13 bit composta da: $[+1+1+1+1+1-1-1+1+1-1+1-1+1]$ che viene ripetuta due volte e si troverà in testa al frame [3].

3.2.2 Generazione messaggio

Per completare il frame è necessario includere il messaggio, in questo caso verrà inviato il messaggio "Hello world" ripetuto 100 volte, con un contatore accanto a ogni messaggio. Inoltre, ogni messaggio deve essere seguito dal carattere di nuova linea.

Il corpo del messaggio finale sarà quindi simile a questo:

```
Hello world 000\n
```

```
Hello world 001\nHello world 002\n...\nHello world 099\n
```

Listing 3.1: Corpo del messaggio

Per codificare il messaggio, avremmo bisogno di 7 bit per carattere. Considerando che ogni riga del messaggio sia composta da:

- "Hello world " (12 caratteri)
- il contatore (3 caratteri)
- il carattere di ritorno a capo \n (1 carattere)

Quindi, il totale dei caratteri per riga è 16, che moltiplicandolo per i 7 bit necessari per ciascuno e per i 100 messaggi che si inviano, otteniamo:

$$16 \text{ caratteri} \times 7 \text{ bit/carattere} \times 100 \text{ messaggi} = 11200 \text{ bit}$$

Pertanto, il corpo del frame sarà composto da 11200 bit

3.2.3 Scrambler

Dopo esser stato generato il vettore contenente il corpo del frame passa per il blocco "Scrambler", un registro a retroazione lineare che riordina i bit del vettore in ingresso.

Lo scopo di questo blocco è quello di evitare che nel messaggio vi siano lunghi tratti di bit identici nel segnale trasmesso. Se il messaggio contiene

sequenze lunghe di bit identici può creare dei problemi al sincronizzatore del ricevitore, rendendo complicato capire quando inizia e finisce ciascun bit.

Per funzionare correttamente devono essere noti sia al trasmettitore che al ricevitore la base di calcolo, il polinomio di scrambling e gli stati iniziali. In questo caso, lavorando con il sistema binario, la base sarà 2. Il polinomio di scrambling è ciò che definisce il registro a scorrimento a retroazione lineare (LFSR), in questo caso: $[1\ 1\ 1\ 0\ 1]$, che corrisponde a $x^4 + x^3 + x^2 + 1$. Questo polinomio determina le posizioni nel registro che devono essere "XORate" per generare il nuovo bit di uscita e aggiornare il registro [4].

3.2.4 Composizione frame

A questo punto è stato generato sia la testa del frame (header), sia il corpo (body), bisogna solo concatenarli e lo si fa attraverso il blocco *Matrix Concatenate*.

Dopo questa operazione si avrà un vettore unico composto da 11226 bit in uscita dal blocco *Bits Generation*.

3.3 Modulatore 4-QAM

La modulazione QAM (Quadrature Amplitude Modulation) come visto nel primo capitolo è una tecnica di modulazione che utilizza due segnali in quadratura per modulare l'ampiezza di un segnale portante. Nel caso della modulazione 4-QAM, il segnale modulato può assumere quattro possibili valori, corrispondenti a quattro diverse combinazioni di ampiezza e fase (vedi Figura 1.3) . Questo permette di trasmettere due bit per ogni simbolo.

I bit in ingresso vengono raggruppati in coppie e mappati su simboli QAM. Le coppie di bit, che possono essere, "00", "01", "10" e "11" saranno mappate rispettivamente ai simboli con coordinate (1,1), (1,-1), (-1,1) e (-1,-1) nel piano complesso.

In uscita quindi dal modulatore 4-QAM si avrà un vettore di lunghezza N che possiamo calcolare sapendo la lunghezza del vettore in ingresso che chiamiamo l e sapendo l'ordine della modulazione utilizzata $M = 4$.

$$N = l / \log_2 M = 5613 \quad (3.1)$$

Il vettore di output sarà quindi un vettore di 5613 simboli QAM, il che significa che conterrà 5613 numeri complessi, ciascuno dei quali rappresenta uno dei 4 possibili simboli. Con la distanza minima impostata a 3, i valori dei simboli possono essere: $(1.5 + 1.5i)$, $(1.5 - 1.5i)$, $(-1.5 + 1.5i)$ e $(-1.5 - 1.5i)$ [5].

3.4 Coseno Rialzato Radice Quadrata (Square Root Raised Cosine)

Il blocco "Square Root Raised Cosine" (SRRC) serve a modellare il segnale prima della trasmissione in una comunicazione digitale. Quando inviamo un segnale, vogliamo che i simboli non si sovrappongano e interferiscano tra loro. Il filtro SRRC aiuta a "modellare" il segnale, rendendolo più adatto alla trasmissione.

Il segnale di output avrà lo stesso tipo di dato del segnale di input, che

nel nostro caso è un vettore di numeri complessi, ma non avrà la stessa dimensione, data dalla dimensione del vettore di ingresso N e dal parametro che si inserisce nel parametro "Campioni di output per simbolo" (Output samples per symbol) K .

La relazione tra i parametri è data dalla formula:

$$out = N \times K = 1226 \text{ campioni} \quad (3.2)$$

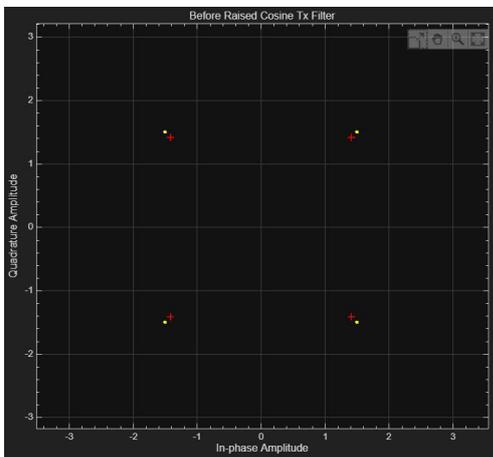


Figura 3.3: Costellazione prima del filtro SRRC

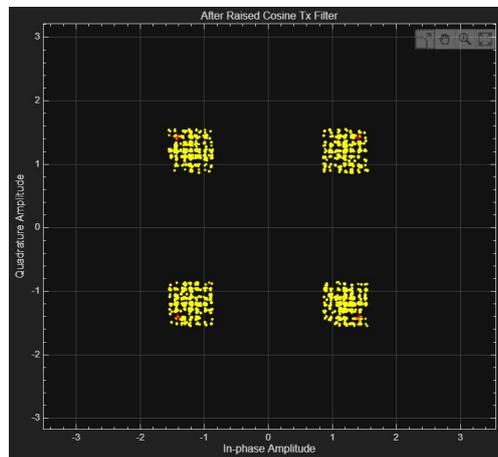


Figura 3.4: Costellazione dopo del filtro SRRC

Nella figura Figura 3.4 possiamo vedere la costellazione del nostro segnale prima e dopo che il passaggio nel filtro [6].

3.5 Trasmettitore ADALM-PLUTO

L'ultimo elemento della sequenza di trasmissione dell'intero sistema di comunicazione proposto è il blocco "ADALM-PLUTO Radio Transmitter", il cui scopo è trasmettere i dati elaborati all'hardware ADALM-PLUTO.

Il blocco acquisisce il segnale modulato e lo elabora in modo che possa essere trasmesso per etere. Nella finestra di configurazione del blocco, è possibile specificare alcuni parametri importanti. Tra questi la frequenza centrale, in questo caso 91.5 MHz, ovvero la frequenza a cui si trova il segnale nel dominio delle frequenze. Il guadagno del segnale in dB, che è il fattore di moltiplicazione per l'ampiezza del segnale di uscita. Impostato a 0 in questo caso, il che permette al dispositivo di regolare automaticamente il guadagno. Ed il "Tasso di campionamento in banda base", impostato a 400 kHz, specifica la frequenza alla quale vengono elaborati i campioni del segnale in banda base e vengono trasmessi [7].

3.6 Conclusione

In questo capitolo, abbiamo esplorato il processo di trasmissione di segnali QAM utilizzando un dispositivo SDR, nello specifico l'ADALM-PLUTO, in ambiente Matlab Simulink. Partendo dalla generazione dei bit e passando per la sincronizzazione tramite il codice di Barker, abbiamo visto come ogni componente del sistema contribuisce alla robustezza e all'efficienza della comunicazione. La modulazione 4-QAM è stata trattata nel dettaglio, evidenziando come i bit siano mappati su simboli QAM. L'inclusione del filtro Square Root Raised Cosine ha permesso di migliorare la qualità del segnale, riducendo l'interferenza intersimbolica e ottimizzando la forma d'onda trasmessa. Infine, il trasmettitore ADALM-PLUTO ha completato il processo, convertendo il segnale digitale modulato in un segnale radio analogico pronto per essere trasmesso.

Capitolo 4

Ricezione QAM con ADALM-PLUTO

4.1 Introduzione

In questo capitolo verrà descritta la procedura di ricezione di un segnale modulato QAM utilizzando ADALM-PLUTO. L'obiettivo è fornire una panoramica completa dei blocchi funzionali coinvolti nel processo di ricezione, dalla cattura del segnale fino alla decodifica e visualizzazione dei dati trasmessi.

Nelle sezioni successive analizzeremo in dettaglio ciascuna fase della ricezione, fornendo esempi visivi delle trasformazioni del segnale e del comportamento del sistema in presenza di rumore e distorsioni tipiche di un canale di comunicazione. Particolare attenzione sarà dedicata alla sincronizzazione e alla correzione di eventuali disallineamenti, che sono fondamentali per garantire l'integrità dei dati ricevuti.

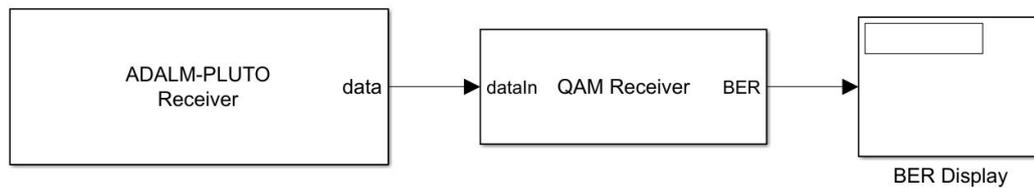


Figura 4.1: Ricezione Matlab Simulink con PlutoSDR

4.2 Ricevitore ADALM-PLUTO

In ricezione, al contrario della trasmissione, troviamo come primo blocco il ricevitore ADALM-PLUTO. Il suo compito è quello di ricevere il segnale analogico trasmesso dal programma di trasmissione.

Per il corretto funzionamento del blocco è necessario configurare, come in trasmissione, la frequenza centrale (91.5 MHz) e il "Tasso di campionamento in banda base" (400 kHz). Per la ricezione inoltre, bisogna impostare il tipo di dati che usciranno e il numero di campioni per frame in output dal blocco, in questo caso sarà un vettore di *double* di 11226 campioni [8].

4.3 Sincronizzazione e decodifica simboli QAM

4.3.1 Regolatore automatico del guadagno (Automatic gain controller)

Il compito del regolatore di guadagno è quello di modificare automaticamente il livello di amplificazione di un segnale in modo da mantenere la sua potenza entro un intervallo desiderato.

Impostato un valore di riferimento della potenza, in questo caso 2, il blocco va a calcolare in tempo reale la potenza del segnale in ingresso. Se il

valore ottenuto è maggiore di quello desiderato il guadagno sarà diminuito, al contrario se il valore è minore verrà aumentato il guadagno.

Un ulteriore parametro da impostare è quello del guadagno massimo, ovvero il valore massimo di guadagno che il blocco può applicare al segnale, in questo caso impostato a 60 [9].

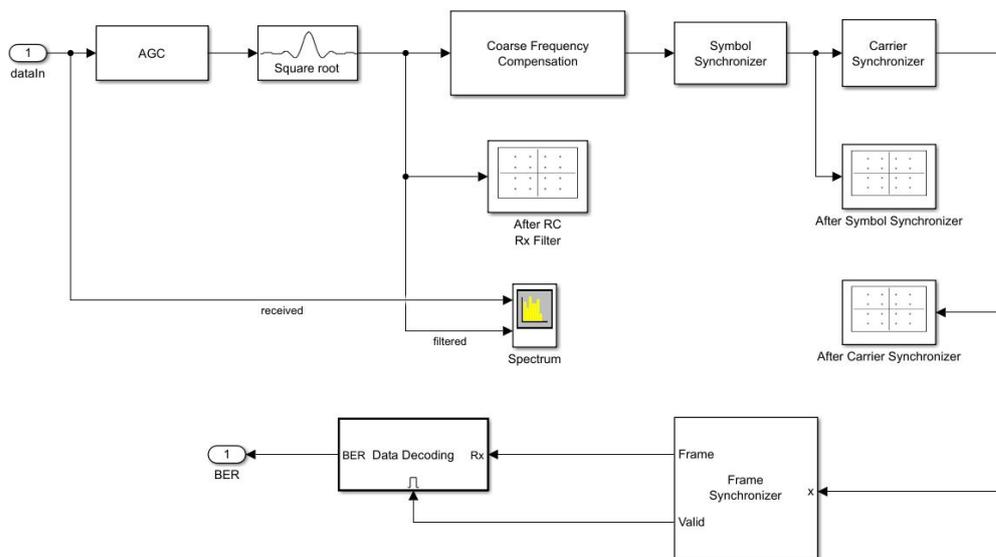


Figura 4.2: Sincronizzazione e decodifica simboli QAM

4.3.2 Filtro ricevitore a coseno rialzato (Raised cosine receiver filter)

Dopo aver regolato il guadagno del segnale passiamo il nostro vettore nel "Filtro ricevitore a coseno rialzato", blocco che lavora in tandem con il filtro di trasmissione a coseno rialzato radice quadrata (SRRC) descritto nella sezione 3.4.

Il filtro SRRC modella il segnale in trasmissione per prevenire la sovrapposizione tra simboli, il filtro ricevitore a radice di coseno rialzato svolge una funzione simile al momento della ricezione, assicurando che i simboli siano recuperati con precisione senza interferenze intersimboliche (ISI).

L'interferenza intersimbolica (ISI) è un fenomeno che si verifica nelle comunicazioni digitali quando i simboli trasmessi in successione si sovrappongono temporalmente, causando interferenze tra di loro. Questo accade perché ogni simbolo non viene confinato perfettamente all'interno del proprio intervallo temporale ed "eccede" parzialmente nell'intervallo dei simboli adiacenti.

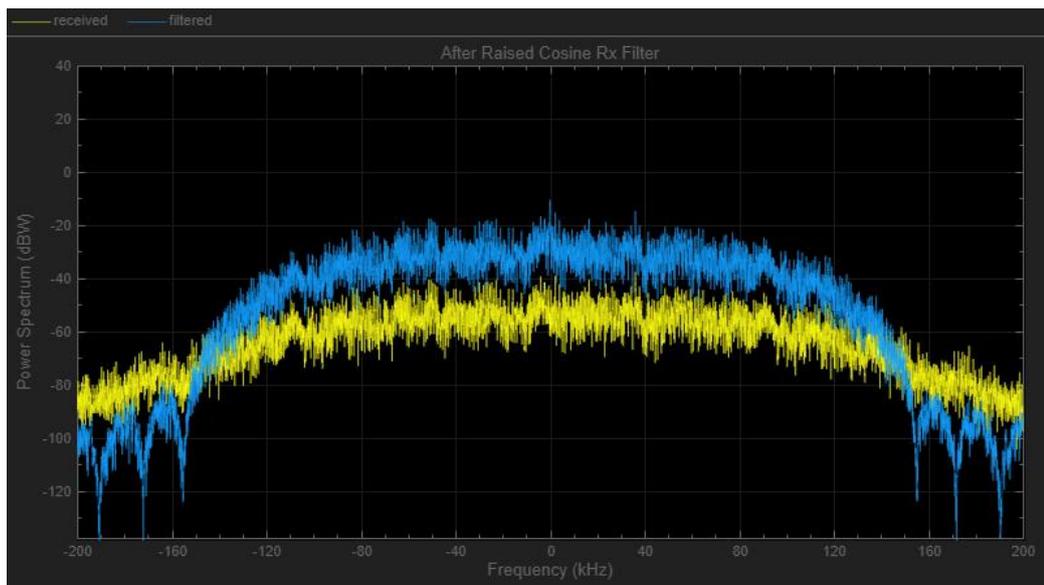


Figura 4.3: Segnale ricevuto e filtrato

Nella Figura 4.3 possiamo osservare in giallo lo spettro di frequenza del segnale ricevuto dalla radio ed in azzurro la variazione dello spettro dopo il blocco del regolatore automatico del guadagno della sezione 4.3.1 e dal filtro ricevitore a coseno rialzato della sezione 4.3.2

Osservando l'immagine dello spettro di frequenza nella Figura 4.3, possiamo notare una chiara riduzione del rumore e una maggiore definizione del segnale dopo l'applicazione del filtro ricevitore a coseno rialzato e del regolatore automatico del guadagno.

In particolare, la traccia in giallo rappresenta il segnale ricevuto dalla radio, caratterizzato da una maggiore larghezza spettrale e una distribuzione meno uniforme della potenza su tutte le frequenze. La traccia in azzurro, invece, mostra come il segnale venga "pulito" e meglio delimitato in termini di frequenza dopo l'elaborazione [10].

4.3.3 Compensatore di frequenza (Coarse frequency compensation)

Quando un segnale viene trasmesso per varie ragioni sarà soggetto a variazioni di frequenza. Queste variazioni possono essere date sia dall'hardware, con una instabilità degli oscillatori, oppure anche da effetti introdotti dal canale.

Il blocco di compensazione di frequenza "grossolana" (Coarse Frequency Compensation) va a ridurre, come dice il nome "grossolanamente", i disallineamenti portando il segnale il più vicino possibile alla frequenza desiderata.

Il lavoro del compensatore di frequenza è essenziale per il garantire che i processi di sincronizzazione successivi descritti nella sezione 4.3.4 e 4.3.5 eseguano una sincronizzazione più fine e precisa [11].

4.3.4 Sincronizzazione simboli (Symbol synchronizer)

Il compito del symbol synchronizer è quello di determinare qual è il momento esatto per campionare il segnale ricevuto. Una errata sincronizzazione può compromettere l'integrità dei dati andando a perdere dell'informazione.

Il sincronizzatore di simboli ci assicura quindi che i simboli siano campionati correttamente, preparandoli per una corretta decodifica.

Il lavoro svolto dal symbol synchronizer si può riassumere in 4 punti:

1. **Errore di temporizzazione:** In questa fase il symbol synchronizer estrae l'errore di temporizzazione (timing error) dal segnale che viene ricevuto.

Per estrarlo viene applicato un algoritmo che va a calcolare la differenza tra la posizione del campione attuale e la posizione ideale del campione, basandosi sull'andamento del segnale.

2. **Filtraggio e correzione errore:** Una volta calcolato l'errore verrà filtrato il segnale in modo tale da andare a limare il segnale ricevuto. Questo passaggio permette di avere un segnale ricevuto più stabile migliorando l'integrità dei dati ricevuti.

Il blocco Symbol Synchronizer utilizza un loop di controllo ad anello chiuso (feedback loop) per correggere continuamente l'istante di campionamento in base all'errore calcolato.

3. **Interpolazione del segnale:** In questo passaggio viene utilizzato un algoritmo di interpolazione che andrà a determinare il valore esatto del segnale nell'istante ideale di campionamento.

4. **Allineamento dei Simboli:** Con l'istante di campionamento corretto determinato e il segnale interpolato, il blocco allinea i simboli ricevuti per garantire che vengano campionati nel loro punto ottimale. Questo garantisce che i simboli digitali vengano recuperati con il minimo errore possibile.

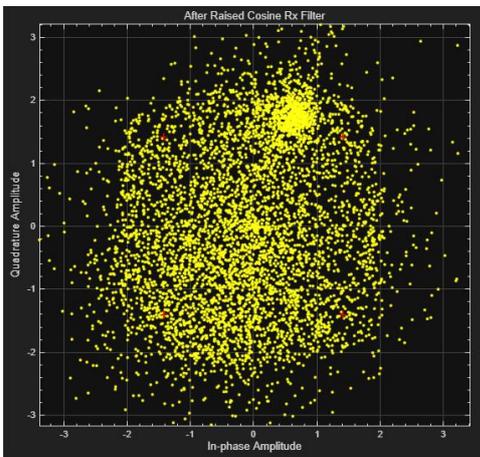


Figura 4.4: Costellazione prima del symbol synchronyzer

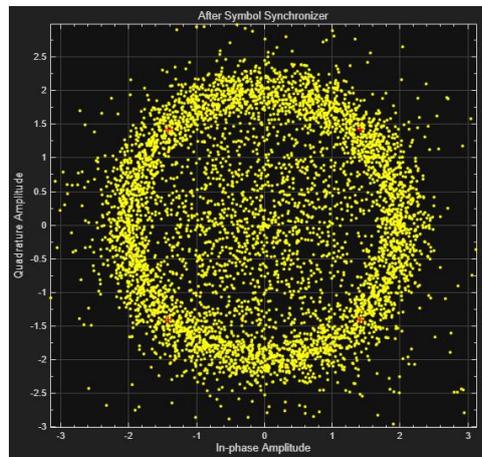


Figura 4.5: Costellazione dopo del symbol synchronyzer

Osservando le costellazioni mostrate nelle Figura 4.4 e Figura 4.5, è evidente l'effetto significativo del processo di sincronizzazione dei simboli [12].

4.3.5 Sincronizzazione portante (Carrier synchronizer)

Il Carrier Synchronizer serve per correggere i piccoli errori di fase e frequenza residui che rimangono dopo i precedenti processi di compensazione e sincronizzazione. Questo garantisce che il segnale demodulato rappresenti fedelmente i dati trasmessi, minimizzando gli errori di decodifica [13].

- **Correzione dell'Errore di Fase:** Una volta rilevato l'errore di fase, il Carrier Synchronizer applica una correzione al segnale ricevuto. Questa correzione viene eseguita moltiplicando il segnale complesso ricevuto per un fattore di correzione che compensa l'errore di fase. Questo fattore è generalmente un'esponenziale complessa del tipo $e^{-j\theta}$, dove θ è l'angolo di fase dell'errore rilevato.
- **Correzione dell'Errore di Fase:** Se c'è un errore di frequenza, il Carrier Synchronizer agisce come un Phase-Locked Loop (PLL) per seguire la deriva della frequenza nel tempo. Il PLL cerca di allineare la frequenza dell'oscillatore locale del ricevitore con quella della portante originale, applicando una correzione continua all'angolo di fase del segnale.

Osservando le costellazioni mostrate nelle Figura 4.6 e Figura 4.7, si nota l'importanza della sincronizzazione della portante nel processo di demodulazione.

Nella prima immagine, prima dell'applicazione del Carrier Synchronizer, la costellazione mostra un chiaro effetto di rotazione e dispersione dei punti. Questo fenomeno è indicativo di errori residui di fase e frequenza che, se non corretti, possono causare significativi errori di decodifica, compromettendo la qualità della trasmissione.

Nella seconda immagine, dopo l'applicazione del Carrier Synchronizer, la costellazione risulta ben definita e i punti sono concentrati nelle loro posizioni corrette (vedi Figura 1.3).

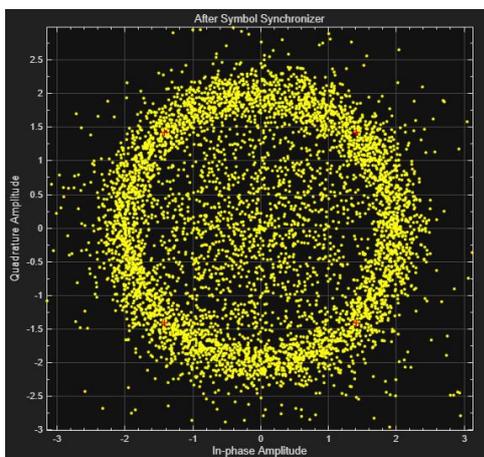


Figura 4.6: Costellazione prima del Carrier Synchronizer

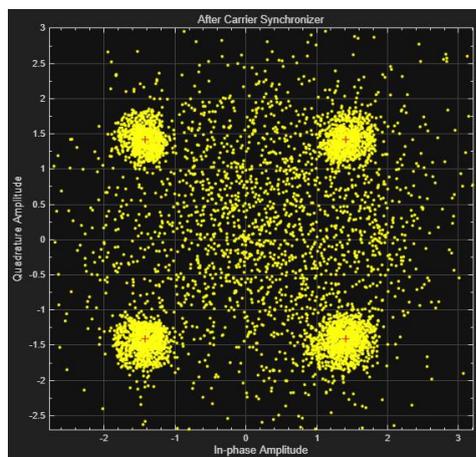


Figura 4.7: Costellazione dopo del Carrier Synchronizer

4.3.6 Sincronizzazione frame (Frame Synchronizer)

Il sincronizzatore del frame, comunemente chiamato frame synchronizer, rileva l'inizio di un frame identificando una sequenza di simboli nota come preambolo. Nel nostro caso la sequenza di simboli (il preambolo) corrisponde al codice di Barker ripetuto 2 volte.

Una volta rilevato il preambolo, il frame synchronizer estrae un frame di una lunghezza specificata dal segnale di ingresso e restituisce anche un segnale booleano che indica se il frame estratto è valido o meno. La validità del frame è determinata confrontando la correlazione incrociata tra il preambolo e l'ingresso con una soglia specificata; se il valore di correlazione supera la soglia, il frame è considerato valido. Il frame synchronizer è configurabile tramite proprietà come il preambolo, la lunghezza del frame di uscita e la soglia di rilevamento, e può gestire segnali sia reali che complessi.

Il segnale booleano, sopra citato, ha lo scopo di attivare il blocco *Enable* (vedi Figura 4.8), iniziando così la procedura di demodulazione.

4.3.7 Decodifica dati (Data decoding)

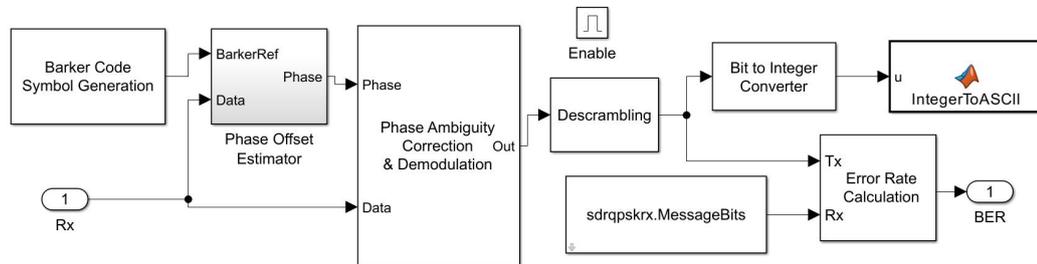


Figura 4.8: Decodifica simboli QAM

Generazione Barker code

La prima fase della decodifica è quella della correzione di fase, per fare questo è necessario confrontare i simboli in arrivo con quelli trasmessi. In ricezione non è noto il messaggio trasmesso quindi non è possibile utilizzare quella parte per la correzione di fase, è però noto il codice di Barker.

Per andare quindi a confrontare i simboli ricevuti dobbiamo generare nuovamente, come è stato fatto in trasmissione, i simboli QAM del Barker code. Come si può osservare dalla Figura 4.9 l'operazione per la generazione dei simboli QAM consiste nel prendere il vettore in bit del Barker code, ripeterlo due volte e poi affidarlo al modulatore 4-QAM.

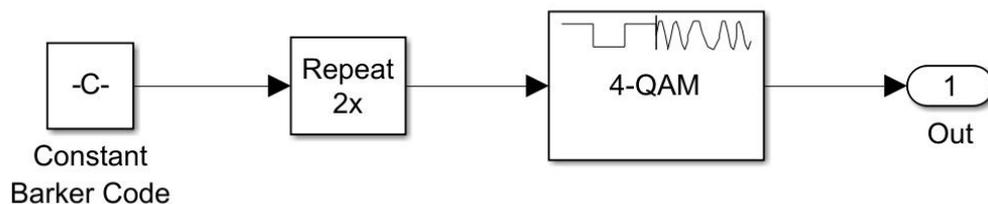


Figura 4.9: Decodifica simboli QAM

Correzione fase e Demodulazione

Generati i simboli da comparare e avendo i simboli ricevuti il blocco "Phase Offset Estimator" (vedi Figura 4.8) stimerà il ritardo di fase del segnale. Il blocco "phase ambiguity correction & demodulation" (vedi Figura 4.8) poi correggerà l'errore di fase e di andrà a demodulare il segnale.

Al suo interno troviamo: (vedi Figura 4.10)

- **Complex phase shift:** Il blocco applica una rotazione di fase a questo segnale complesso. La rotazione di fase è ottenuta moltiplicando il segnale complesso in ingresso per un'esponenziale complessa del tipo $e^{j\theta}$, dove θ è l'angolo di fase (in radianti) che abbiamo ottenuto confrontando il codice di Barker generato a quello ricevuto precedentemente. Questa operazione trasla la fase del segnale di un angolo θ .
- **Demodulatore QAM:** Il compito del demodulatore è quello di tradurre i simboli QAM che sono stati ricevuti in bit.

Al contrario della trasmissione la costellazione dei simboli in ricezione non è concentrata nelle sole 4 coordinate $(1.5 + 1.5i)$, $(1.5 - 1.5i)$, $(-1.5 + 1.5i)$ e $(-1.5 - 1.5i)$. Come si può osservare dalla Figura 4.7 il demodulatore dovrà prendere delle decisioni per andare a demodulare nel migliore dei modi tutti i simboli.

In questo caso verrà utilizzato il criterio a minima distanza descritto nella sezione 1.2.2 e ad ogni simbolo ricevuto verranno quindi assegnati i bit di decodifica corrispondenti al simbolo trasmesso più vicino.

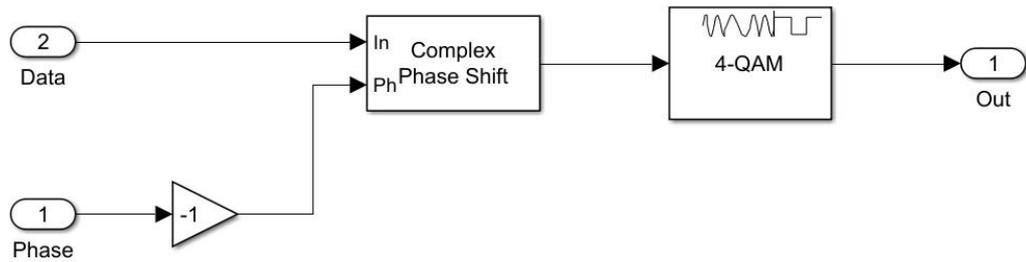


Figura 4.10: Correzione fase e decodifica simboli QAM

Descrambling

Il "Descrambler" è un dispositivo utilizzato nel ricevitore per ripristinare la sequenza originale di bit di un messaggio che è stato dallo scrambler nel trasmettitore della sezione 3.2.3.

Questo funziona in modo inverso rispetto allo scrambler: il descrambler infatti applica la stessa sequenza di operazioni, definita dal polinomio di scrambling e dagli stessi stati iniziali, per annullare l'effetto del riordino dei bit.

Questo processo permette di recuperare il messaggio originale, garantendo che il contenuto ricevuto corrisponda a quello trasmesso. Per funzionare correttamente, il descrambler deve conoscere esattamente il polinomio e le condizioni iniziali usate dallo scrambler nel trasmettitore [14].

4.3.8 Stampa del messaggio

I blocchi "Bit to Integer Converter" e "IntegerToASCII" (vedi Figura 4.8), sono responsabili della stampa del messaggio ricevuto nella console.

Il blocco "Bit to Integer Converter" si occupa di convertire i bit in interi, poiché per poter visualizzare correttamente un carattere sulla console è

necessario utilizzare la corrispondente codifica ASCII.

Il blocco "IntegerToASCII", invece, è una funzione di MATLAB che converte gli interi in caratteri ASCII e stampa l'intero frame direttamente sulla console (vedi Listing 4.1).

```
function IntegerToASCII(u)
    fprintf('%s', char(u));
end
```

Listing 4.1: Funzione IntegerToASCII in MATLAB

4.3.9 Tasso di errore

Come ultima operazione, andiamo a controllare alla fine della ricezione quanti bit di informazione sono stati persi dalla comunicazione, calcolando il tasso di errore.

A fare questo sarà il blocco "Error Rate Calculation" (vedi Figura 4.8) il quale in input avrà sia il messaggio ricevuto, sotto forma di bit, sia il messaggio trasmesso. Chiaramente questo passaggio è possibile solo conoscendo a priori il messaggio trasmesso, passaggio extra che non va ad intaccare quindi il processo della ricezione bensì a dare un feedback sulla qualità della trasmissione.

4.4 Conclusione

In questo capitolo, abbiamo esplorato il processo di ricezione di un segnale modulato QAM utilizzando il ricevitore ADALM-PLUTO integrato con

MATLAB e Simulink. Abbiamo iniziato descrivendo il ruolo del ricevitore ADALM-PLUTO nella cattura del segnale analogico trasmesso e la sua conversione in un formato digitale.

Successivamente ci siamo concentrati sui blocchi funzionali che garantiscono la corretta sincronizzazione del segnale ricevuto. Il regolatore automatico del guadagno ha mostrato come regolare il livello di amplificazione per mantenere la potenza del segnale entro un intervallo desiderato, mentre il filtro a coseno rialzato ha permesso di eliminare le interferenze intersimboliche, migliorando la qualità del segnale ricevuto.

Abbiamo esaminato in dettaglio il processo di sincronizzazione dei simboli, evidenziando l'importanza di questi passaggi mostrando come andava a variare la costellazione nel nostro segnale, garantendo così una corretta demodulazione. Il Carrier Synchronizer ha svolto un ruolo cruciale nella correzione degli errori di fase e frequenza residui, mentre il Frame Synchronizer ha garantito l'allineamento preciso dei frame di dati.

Infine, abbiamo descritto le fasi finali della decodifica dei dati, inclusa la correzione di fase, la demodulazione e il descrambling, che hanno permesso di recuperare il messaggio originale trasmesso. Il capitolo si è concluso con un'analisi del tasso di errore.

In sintesi, questo capitolo ha fornito una panoramica completa e dettagliata del processo di ricezione di un segnale QAM, evidenziando l'importanza di ciascun blocco funzionale nel garantire l'integrità e la qualità dei dati ricevuti.

Capitolo 5

Esperimenti

5.1 Introduzione

In questo capitolo analizzeremo i risultati ottenuti utilizzando il sistema di trasmissione descritto nei Capitoli 3 e 4. Grazie al blocco "BER Display" (vedi Figura 4.2) e al sistema illustrato nella Sezione 4.3.9, esamineremo come la variazione della distanza tra il dispositivo di trasmissione e quello di ricezione influenzi il tasso di errore. Inoltre esploreremo come l'incremento della distanza minima tra i simboli possa contribuire a migliorare significativamente la qualità della comunicazione.

I dati presentati nelle prossime sezioni sono il risultato di una media di cinque esperimenti condotti per ciascuna configurazione. Per garantire l'affidabilità dei risultati, tutti i test sono stati eseguiti nello stesso luogo e nella stessa giornata, riducendo così al minimo le possibili variazioni dovute a condizioni esterne.

5.2 Analisi a diverse distanze e ostacoli

In questi esperimenti, abbiamo posizionato i dispositivi ADALM-PLUTO a diverse distanze e introdotto vari ostacoli tra di essi, con l'obiettivo di analizzare come tali variazioni influenzino il tasso di errore della comunicazione, consentendo una valutazione accurata della qualità del segnale in diverse condizioni ambientali.

Per ciascun esperimento, sono stati trasmessi $1,363 \times 10^6$ bit. Il tasso di errore è stato calcolato dividendo il numero di bit errati ricevuti per il numero totale di bit trasmessi. La variazione in percentuale rispetto all'esperimento di riferimento a 1m è stata determinata confrontando il numero di bit errati in ogni esperimento successivo con quello del primo e calcolando la percentuale di incremento o diminuzione degli errori.

I risultati ottenuti, insieme alle relative variazioni, sono riportati di seguito:

- **Distanza di 1m:** Il tasso di errore registrato è stato dello 0,18%, che funge da riferimento per confrontare gli altri scenari.
 - **Variazione:** Questo esperimento verrà utilizzato come punto di partenza.

- **Distanza di 2m:** Anche a 2m di distanza, il tasso di errore è rimasto, approssimando il risultato alla 4 cifra decimale, allo 0,18%.
 - **Variazione:** Senza approssimazioni otteniamo una variazione del +1,10% rispetto all'esperimento a 1 metro. Questo lieve aumento indica che l'incremento di distanza ha un effetto trascurabile.

- **Distanza di 5m + porta:** Con una distanza di 5m e una porta come ostacolo, il tasso di errore è sceso allo 0,12%.
 - **Variazione:** -36,40%. Sorprendentemente, l'errore diminuisce significativamente, probabilmente grazie a minori riflessioni.

- **Distanza di 8m + 2 porte:** A 8m e con due porte, il tasso di errore è leggermente aumentato allo 0,13%.
 - **Variazione:** -29,67%. Nonostante la maggiore distanza e il doppio ostacolo, l'errore resta inferiore rispetto al caso a 1m, suggerendo che il sistema mantenga una buona performance anche in queste condizioni.

- **Distanza di 5m e un muro:** Quando il segnale è stato trasmesso attraverso un muro a una distanza di 5m, il tasso di errore è salito allo 0,15%.
 - **Variazione:** -17,58%. L'aumento dell'errore rispetto agli ostacoli più leggeri indica l'influenza negativa del muro sulla trasmissione, sebbene l'errore resti comunque contenuto. Se consideriamo il miglior risultato ottenuto come riferimento, ovvero il terzo esperimento (5m + porta), il muro ha portato ad un incremento dell'errore del 30%

- **Distanza di 8m e 2 muri, di cui uno portante:** A 8m, con due muri di cui uno portante, il tasso di errore è aumentato allo 0,45%.

- **Variazione:** +148,35%. Qui si osserva un aumento significativo rispetto all’esperimento da 1 metro, evidenziando come la complessità degli ostacoli influisca pesantemente sulla qualità del segnale. Nel risultato finale l’errore rimane comunque irrilevante.
- **Distanza di 12m e 3 muri, di cui 2 portanti:** Infine, con una distanza di 12m e la presenza di tre muri, di cui due portanti, il tasso di errore è esploso al 15,50%.
 - **Variazione:** +8415,38%. Questo drammatico incremento dimostra che la combinazione di lunga distanza e ostacoli fisici significativi compromette gravemente la comunicazione, rendendola inaffidabile.

Esp. n°	1m	2m	5m + porta	8m + 2 por- te	5m + muro	8m + 2 mu- ri	12m + 3 muri
1	0,0016	0,0012	0,0017	0,0006	0,0026	0,0015	0,2607
2	0,0022	0,0028	0,000088	0,0028	0,0019	0,0031	0,1531
3	0,0019	0,003	0,0014	0,0012	0,0022	0,0038	0,1163
4	0,0014	0,002	0,0019	0,0004	0,0002	0,0042	0,1152
5	0,002	0,0002	0,0007	0,0014	0,0006	0,01	0,1296
Err. medio	0,18%	0,18%	0,12%	0,13%	0,15%	0,45%	15,50%

Tabella 5.1: Risultati degli esperimenti con diverse distanze e ostacoli

5.2.1 Considerazioni sui dati

Dall’analisi emerge che il sistema di trasmissione mantiene una buona performance con tassi di errore relativamente bassi fino a quando le distanze e gli

ostacoli rimangono moderati. Tuttavia, l'introduzione di ostacoli più densi, specialmente cemento armato, combinata con distanze più lunghe, porta a un incremento esponenziale del tasso di errore. Questo dimostra chiaramente che la qualità della comunicazione è fortemente influenzata non solo dalla distanza ma anche dalla natura e dal numero degli ostacoli incontrati lungo il percorso del segnale.

5.3 Distanza Minima dei Simboli della Costellazione

In questo capitolo analizziamo l'effetto dell'aumento della distanza minima tra i simboli della costellazione sul tasso di errore. Aumentare la distanza minima dei simboli significa andare a variare le coordinate di quest'ultimi nella costellazione e questo viene fatto dal blocco Modulatore 4-QAM studiato nella sezione 3.3 andando ad impostare come parametro la distanza minima che si vuole.

Conseguentemente, essendo la potenza di trasmissione proporzionale al quadrato della distanza dall'origine dei punti della costellazione, avremmo che il blocco Trasmettitore ADALM-PLUTO (vedi Sezione 3.5) dovrà aumentare la potenza per riuscire a trasmettere correttamente i simboli. Raddoppiando la distanza minima dei simboli nel caso della costellazione 4-QAM avremmo che la potenza necessaria per trasmetterli quadruplica.

Gli esperimenti sono stati fatti mantenendo invariata la configurazione degli ostacoli e la distanza tra i dispositivi, più precisamente sono stati con-

dotti mantenendo una distanza di 12m tra i dispositivi, con tre muri, di cui due portanti, posti tra il trasmettitore e il ricevitore equivalentemente all'ultimo esperimento svolto nella sezione 5.2, in cui la distanza minima dei simboli era pari a 2.

I risultati dei nuovi esperimenti sono riportati di seguito:

- **Distanza minima 4:** In questo esperimento, la distanza minima tra i simboli della costellazione è stata raddoppiata a 4. Il tasso di errore risultante è stato del 12,18%. Questo rappresenta una riduzione significativa rispetto al 15,50% ottenuto con la distanza minima originale di 2, con una diminuzione del tasso di errore del 21%. Questo suggerisce che un aumento moderato della distanza tra i simboli può contribuire a migliorare la qualità della comunicazione, sebbene l'errore rimanga ancora elevato a causa delle condizioni ambientali avverse.
- **Distanza minima 8:** Successivamente, la distanza minima è stata ulteriormente aumentata a 8. In questo caso, il tasso di errore è sceso drasticamente all'1,62%, con una riduzione del 90% rispetto all'esperimento con distanza minima 2. Questo risultato evidenzia come l'aumento significativo della distanza minima tra i simboli possa ridurre in modo considerevole il numero di errori, migliorando notevolmente la robustezza della comunicazione anche in presenza di ostacoli critici come muri portanti.

5.3.1 Considerazioni sui dati

Dai risultati emerge chiaramente che l'aumento della distanza minima tra i simboli della costellazione ha un impatto positivo sulla riduzione del tasso di errore, specialmente in condizioni di trasmissione difficili. L'incremento da 2 a 8 della distanza minima ha portato a un miglioramento sostanziale, riducendo il tasso di errore di un ordine di grandezza. Questi risultati indicano che, in scenari con ostacoli significativi, aumentare la distanza minima tra i simboli può essere una strategia efficace per migliorare la qualità complessiva della comunicazione.

Esperimento	Distanza 2	Distanza 4	Distanza 8
1	0,2607	0,0759	0,03163
2	0,1531	0,1659	0,01218
3	0,1163	0,1284	0,01507
4	0,1152	0,1377	0,01162
5	0,1296	0,101	0,01054
Err. medio	15,50%	12,18%	1,62%

Tabella 5.2: Risultati degli esperimenti con diverse distanze minime tra i simboli

Capitolo 6

Conclusioni

In questa tesi, abbiamo esplorato il processo di trasmissione e ricezione di segnali digitali modulati in 4-QAM utilizzando la piattaforma SDR ADALM-PLUTO integrata con MATLAB Simulink. Partendo dalla trasmissione digitale, si è dimostrato come il sistema proposto riesca a trasformare un flusso di bit in segnali analogici, passando attraverso fasi fondamentali come la modulazione e tecniche atte al fine di migliorare la qualità della comunicazione. Analogamente, la fase di ricezione ha mostrato il processo inverso, evidenziando l'importanza della demodulazione e della correzione degli errori di fase e frequenza.

Una parte cruciale di questa analisi è stata dedicata agli esperimenti pratici, che hanno permesso di verificare le prestazioni del sistema in condizioni reali. Gli esperimenti descritti nel Capitolo 5 hanno evidenziato come la qualità della trasmissione sia fortemente influenzata da variabili ambientali, come la distanza tra trasmettitore e ricevitore e la presenza di ostacoli fisici. I risultati mostrano che, a distanze moderate (fino a 8m), il sistema riesce a

mantenere un tasso di errore molto basso, anche in presenza di ostacoli come porte e muri. Tuttavia, con l'aumento della distanza e la presenza di barriere più spesse, come muri portanti, il tasso di errore subisce un incremento significativo, fino a diventare insostenibile in scenari più complessi.

Un altro aspetto chiave emerso dagli esperimenti riguarda l'impatto dell'aumento della distanza minima tra i simboli della costellazione QAM. Come dimostrato, incrementare tale distanza riduce drasticamente il tasso di errore, anche in condizioni di trasmissione difficili. In particolare, l'incremento della distanza minima da 2 a 8 ha portato a una riduzione del tasso di errore dal 15,50% all'1,62%, confermando l'efficacia di questa strategia nel migliorare la robustezza del sistema.

In conclusione, i risultati ottenuti dimostrano che l'uso di SDR come ADALM-PLUTO, combinato con adeguate tecniche di modulazione e sincronizzazione, può offrire prestazioni robuste in ambienti variabili. L'analisi condotta in questa tesi rappresenta un punto di partenza per future ottimizzazioni, che potrebbero includere ulteriori miglioramenti nelle tecniche di correzione degli errori e nella gestione del rumore, al fine di rendere la comunicazione ancora più affidabile in scenari complessi.

Bibliografia

- [1] N. Benvenuto and M. Zorzi, “5: Digital modulation systems,” in *Principles of Communications Networks and Systems*, pp. 259–355, Wiley, 2011.
- [2] I. Analog Devices, “Adi adalm-pluto active learning module,” 2024. Disponibile online: <https://www.mouser.it/new/analog-devices/adi-adalm-pluto/>.
- [3] MathWorks, “Barker code generator.” <https://www.mathworks.com/help/comm/ref/barkercodegenerator.html>, 2024.
- [4] MathWorks, “Scrambler.” <https://www.mathworks.com/help/comm/ref/scrambler.html>, 2024.
- [5] MathWorks, “Quadrature amplitude modulation (qam).” <https://it.mathworks.com/help/comm/ref/qammod.html>, 2024.
- [6] MathWorks, “Square root raised cosine filter.” <https://it.mathworks.com/help/comm/ug/raised-cosine-filtering.html>, 2024.
- [7] MathWorks, “Adalm-pluto radio transmitter.” <https://it.mathworks.com/help/comm/plutoradio/ref/plutotransmitter.html>, 2024.

- [8] MathWorks, “Adalm-pluto radio receiver.” <https://it.mathworks.com/help/comm/plutoradio/ref/plutoreceiver.html>, 2024.
- [9] MathWorks, “Automatic gain controller.” <https://www.mathworks.com/help/comm/ref/agc.html>, 2024.
- [10] MathWorks, “Raised cosine receiver filter.” <https://it.mathworks.com/help/comm/ug/raised-cosine-filtering.html>, 2024.
- [11] MathWorks, “Coarse frequency compensation.” <https://it.mathworks.com/help/comm/ref/coarsefrequencycompensator.html>, 2024.
- [12] MathWorks, “Symbol synchronizer.” <https://www.mathworks.com/help/comm/ref/symbolsynchronizer.html>, 2024.
- [13] MathWorks, “Carrier synchronizer.” <https://www.mathworks.com/help/comm/ref/carriersynchronizer.html>, 2024.
- [14] MathWorks, “Descrambler.” <https://www.mathworks.com/help/comm/ref/descrambler.html>, 2024.