



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA

DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA DELLE  
TELECOMUNICAZIONI

**Standard DVB-T2:  
architettura e tecniche di  
mappatura-demappatura  
dell'informazione.**

*Laureando:*

Loris Brolo

*Relatore:*

Prof. Stefano Tomasin

Anno accademico 2009/2010



# Indice

<b>Acronimi</b>	<b>5</b>
<b>Introduzione</b>	<b>9</b>
<b>1 Panoramica Sistema DVB-T2</b>	<b>11</b>
1.1 Introduzione allo standard DVB-T2 . . . . .	11
1.2 Il modello architetturale . . . . .	12
1.2.1 Mode Adaption . . . . .	14
1.2.2 Stream Adaption . . . . .	16
1.2.3 Bit Interleaved Coding and Modulation . . . . .	17
1.2.4 Frame Builder . . . . .	19
1.2.5 OFDM Generator . . . . .	20
<b>2 Modulazione OFDM</b>	<b>25</b>
<b>3 Costellazione Ruotata</b>	<b>29</b>
3.1 Principio di funzionamento . . . . .	29
3.2 Criterio hard decision . . . . .	33
3.2.1 Costruzione regioni di decisione . . . . .	37
3.2.2 Calcolo symbol error rate . . . . .	42
3.3 Soft Decision: LLR - Log Likelihood Ratio . . . . .	45
3.4 Demappatura e decodifica iterativa . . . . .	48
<b>Conclusioni</b>	<b>51</b>

4

*INDICE*

**A Simulazione Matlab**

**53**

**Bibliografia**

**83**

# Acronimi

<b>ACE</b>	Active Constellation Extension
<b>AWGN</b>	Additive White Gaussian Noise
<b>BBFRAME</b>	Base Band Frame
<b>BBHEADER</b>	Base Band Header
<b>BCH</b>	Bose-Chaudhuri-Hocquenghem
<b>BICM</b>	Bit Interleaved Coding and Modulation
<b>CBR</b>	Constant Bit Rate
<b>CRC</b>	Cyclic Redundancy Check
<b>DAC</b>	Digital to Analogic Converter
<b>DFT</b>	Discrete Fourier Transform
<b>DVB-S</b>	Digital Video Broadcasting - Satellite
<b>DVB-T</b>	Digital Video Broadcasting - Terrestrial
<b>DVB-T2</b>	Digital Video Broadcasting - 2nd Generation Terrestrial
<b>FEC</b>	Forward Error Correction
<b>FECFRAME</b>	Forward Error Correction Frame
<b>FEF</b>	Future Extension Frame

<b>FFT</b>	Fast Fourier Transform
<b>GCS</b>	Generic Continuous Stream
<b>GFPS</b>	Generic Fixed-length Packetized Stream
<b>GSE</b>	Generic Stream Encapsuled
<b>HDTV</b>	High Definition TeleVision
<b>IFFT</b>	Inverse Fast Fourier Transform
<b>ISI</b>	InterSymbol Interference
<b>LDPC</b>	Low Density Parity Check
<b>LLR</b>	Log Likelihood Ratio
<b>MAP</b>	Maximum A Posteriori
<b>MD</b>	Minimum Distance
<b>MISO</b>	Multiply Input Single Output
<b>ML</b>	Maximum Likelihood
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>PAPR</b>	Peak to Avarage Power Ratio
<b>PLP</b>	Physical Layer Pipes
<b>QAM</b>	Quadrature Amplitude Modulation
<b>QPSK</b>	Quaternary Phase Shift Keying
<b>SFN</b>	Single Frequency Network
<b>SISO</b>	Single Input Single Output
<b>TFS</b>	Time-Frequency Slicing

**TR** Time Remove

**TS** Transport Stream

**UP** User Packet



# Introduzione

In questo elaborato verranno presentate alcune tecniche di mappatura-demappatura dell'informazione applicate al nuovo standard del digitale terrestre di nuova generazione, identificato dalla sigla DVB-T2.

L'elaborato inizierà con un primo capitolo all'interno del quale verrà presentato in generale il nuovo standard DVB-T2: saranno analizzati dapprima i motivi che hanno portato alla sua nascita e, successivamente, verranno esposti i vari vantaggi e svantaggi attraverso un confronto con il già esistente ed affermato standard DVB-T.

All'interno dello stesso capitolo verrà poi presentato un modello strutturale dell'intero sistema digitale terrestre DVB-T2. In questa parte verranno analizzati i vari blocchi che costituiscono la parte del trasmettitore e del ricevitore dello standard, offrendo per ognuno una breve spiegazione riguardo alla loro funzione.

Dopo una breve panoramica sullo standard DVB-T2, si passerà al Capitolo 2 il quale descriverà la tecnica di modulazione OFDM: si tratta di una tecnica di modulazione di tipo multi-portante che utilizza un numero elevato di sottoportanti ortogonali tra di loro, in ognuna delle quali si impiega un metodo di modulazione convenzionale quale il QAM.

Come verrà illustrato successivamente questa nuova tecnica, introdotta nel sistema digitale terrestre di seconda generazione, porta con sé un importante vantaggio: la comunicazione avviene anche in condizioni pessime del canale, ovvero quando il canale presenta forti attenuazioni.

All'interno del Capitolo 3 si discuterà della vera innovazione introdotta

nello standard DVB-T2, ovvero la tecnica della costellazione ruotata. Questo nuovo procedimento consiste nella rotazione di un determinato angolo  $\Phi$  dell'intera segnalazione QAM data in ingresso al sistema e, quindi, anche della rotazione del medesimo angolo della rappresentazione vettoriale della segnalazione (costellazione). Questa operazione comporta un incremento delle prestazioni dell'intero sistema se si esegue un interleaving, ovvero le componenti in fase e quadratura che identificano un punto della costellazione vengono trasmesse su portanti diverse. In questo modo la probabilità che al ricevitore si demoduli un simbolo diverso da quello trasmesso è molto bassa, perché a frequenze diverse il canale introduce amplificazioni/attenuazioni differenti.

All'interno dello stesso capitolo, inoltre, saranno presentate due tecniche per la demappatura dei simboli trasmessi dalla sorgente, ovvero Hard Decision e Soft Decision, quest'ultima meglio indicata come calcolo del valore di LLR. La prima di queste si basa su una regola di decisione ricavata applicando uno dei tre metodi che verranno analizzati all'interno del paragrafo (ML-MAP-MD), mentre la seconda tecnica è una strategia di rilevazione soft decision, che fornisce la misura di quanto il campione del segnale ricevuto sia vicino al simbolo della costellazione. In sintesi, l'LLR è un parametro che assume valore positivo o negativo quando il simbolo da decidere è rispettivamente uno 0 o un 1, ed il modulo di un LLR indica l'affidabilità della decisione.

Verrà inoltre analizzata una terza tecnica la quale, pur basandosi sempre sul calcolo del valore di LLR, in tal caso tale parametro viene calcolato a partire dalla conoscenza a priori della probabilità di stato dei bit trasmessi fino ad un determinato istante. Questa tecnica va sotto il nome di demappatura iterativa.

Le tre tecniche sopra descritte verranno analizzate e saranno poi confrontate tra di esse in modo tale da capire quale sia la migliore nel contesto della situazione che si presenta.

Chiuderà il lavoro un'appendice contenente il codice Matlab che esegue una simulazione del criterio Hard Decision, presentato nel Capitolo 3.

# Capitolo 1

## Panoramica Sistema DVB-T2

### 1.1 Introduzione allo standard DVB-T2

Lo standard DVB-T2 nasce nel 2006 da un gruppo di lavoro costituito ad-hoc e focalizzato nel realizzare un nuovo standard per la televisione digitale terrestre. Esso diventa a tutti gli effetti uno standard nel 2008 a conclusione di un intenso lavoro a cui hanno partecipato circa 70 esperti appartenenti a più di 40 aziende. Tale sistema si basa sulla tecnologia terrestre già esistente DVB-T e considera, inoltre, alcune caratteristiche dello standard DVB-S, tecnologia per la trasmissione satellitare.

La motivazione principale che sta dietro alla nascita di questo standard è il desiderio di offrire ai diversi paesi europei un servizio televisivo HDTV, ma questo nuovo servizio porta con sé una serie di operazioni che possono riservare, a loro volta, aspetti vantaggiosi o svantaggiosi per l'affermarsi della tecnologia, come il cambiamento della codifica di sorgente e l'introduzione di nuovi apparecchi domestici per la ricezione.

Per la realizzazione dello standard DVB-T2 si sono rivelati molto importanti i vincoli legati ai requisiti commerciali, riportati in maniera dettagliata nella Tabella 1.1

Requisiti	Caratteristiche richieste alle specifiche DVB-T2
Tipo di ricezione	Ricezione fissa con possibilità di configurazioni (come DVB-T) per ricezioni portatili e mobili
Vincoli frequenziali	Trasmissione entro i livelli di interferenza e le maschere spettrali definite in 'GE06 Agreement, Geneva 2006' e senza ulteriori interferenze rispetto al DVB-T
Capacità trasmissiva	Massimo incremento della capacità trasmissiva netta rispetto al DVB-T in simili condizioni, (almeno del 30% per ogni canale) con migliori caratteristiche di robustezza.
Trattamento dei flussi	- Trasporto dell' MPEG2 Transport Stream ed anche del GSE (Generic Stream Encapsulation) definito dal DVB; - Trasporto simultaneo di più flussi di trasporto DVB in un singolo canale; - Possibilità di un'efficiente modulazione statistica dei flussi di ingresso; - Conseguenti modifiche alle SI (Service Information) del Transport Stream da recepire nelle specifiche DVB SI.
Robustezza	- Maggiore robustezza, rispetto al DVB-T, nei confronti di interferenze provenienti da altri trasmettitori, incrementando con ciò la possibilità di riuso delle frequenze; - Differenti livelli di protezione da applicare uniformemente a tutti i dati del Transport Stream trasportato dal DVB-T2 in un particolare canale; - Possibile applicazione separata di differenti livelli di protezione a ciascun servizio all'interno del Transport Stream trasportato in un particolare canale. Quando viene trasportato più di un TS, DVB-T2 deve offrire una scelta di differenti livelli di protezione da applicare separatamente per ciascun TS; - Q.o.S che assicuri, attraverso l'intero canale, non più di un grave disturbo (corrupted event)(audio o video) per ciascuna ora di ciascun servizio HDTV e SDTV; - Prestazioni nei confronti di rumore impulsivo sostanzialmente migliori di quelle del DVB-T.
Velocità di adattamento	- Rivelazione automatica, entro 1/2 s, delle variazioni delle opzioni di modulazione. Sebbene il ricevitore potrebbe non essere in grado di adattarsi automaticamente (seamless changeover); - Non più di 0.30 s di ritardo addizionale nello zapping (cambio canale) rispetto al DVB-T.
Ri-uso infrastruttura DVB-T	Ri-uso dei siti e tralci di trasmissione, antenne e cavi delle installazioni domestiche usati per il DVB-T.
Costi	- Riduzione, rispetto al DVB-T, dei costi di trasmettitori, a parità di potenza, sia in termini di investimento sia di costi di gestione; - Economicità nella realizzazione della copertura di aree locali, regionali e nazionali nel contesto della normative sull'allocazione dello spettro radio. Ad esempio ottimizzando i costi delle infrastrutture e l'uso dello spettro con tecniche SFN e/o MFN.
SFN	- Realizzazione di reti SFN su scala più larga rispetto al DVB-T. In una rete SFN la massima distanza tra trasmettitori adiacenti deve essere incrementata di almeno il 30% rispetto a quella offerta da un DVB-T con modalità 8K ed il medesimo livello di mutua interferenza; - Sviluppo di "gap filler" economici e conformi alle normative, allo scopo di una agevole copertura indoor.

Tabella 1.1: Requisiti standard DVB-T2.

## 1.2 Il modello architetturale

Le specifiche di sistema per lo standard DVB-T2 approvate dalla comunità DVB includono le seguenti applicazioni chiave:

- Stessa tecnica di modulazione utilizzata per lo standard DVB-T, ovvero modulazione OFDM con intervallo di guardia: tale modulazione è attualmente utilizzata per la maggior parte dei sistemi di trasmissione e consente di avere un sistema di trasmissione affidabile.
- Utilizzo di FFT di dimensioni maggiori rispetto a DVB-T: questo consente di aumentare le prestazioni delle reti SFN ed incrementare i

possibili valori degli intervalli di guardia.

- Uguali meccanismi di suddivisione dei dati in frame di banda base (BBFRAME) e tecniche di rivelazione e correzione d'errore utilizzate nello standard DVB-S2.
- Utilizzo delle costellazioni  $M$ -QAM, come già avviene nel DVB-T, e aggiunta della costellazione 256-QAM. Inoltre per migliorare le prestazioni del sistema in canali terrestri critici si utilizza una nuova tipologia di costellazione, la *costellazione ruotata*.
- Utilizzo di un unico canale fisico, il quale è suddiviso in tanti canali logici denominati PLP: essi consentono di trasportare servizi differenti associando ad ogni flusso un particolare grado di protezione che varia da servizio a servizio.
- Utilizzo di quattro livelli di interlacciamento dell'informazione (Bit, Cella, Frequenza, Tempo) che permettono di far fronte in modo migliore ai diversi tipi di degradamento del segnale sul canale.
- Struttura di trama flessibile, in cui i dati possono essere dispersi sull'intera trama per ottenere la massima diversità temporale, oppure concentrati in burst per consentire il massimo risparmio energetico nel ricevitore.
- Meccanismi per ridurre il rapporto tra potenza di picco e potenza media del segnale trasmesso (PAPR).
- Meccanismi per permettere in futuro l'introduzione nel sistema di nuove tecniche preservando la compatibilità all'indietro: TFS e FEF.

Diamo ora una descrizione del sistema di trasmissione DVB-T2, che può essere diviso in diverse sezioni, specificate ed analizzate qui di seguito.

L'ingresso di un sistema DVB-T2 consiste di uno o più flussi logici di dati. Un flusso logico è trasportato da un PLP. Nel caso in cui si abbia un singolo

flusso di ingresso lo schema da tenere in considerazione è quello di Figura 1.1a, nel caso di più flussi di ingresso si consideri la Figura 1.1b.

### 1.2.1 Mode Adaption

Il blocco *Mode Adaption* riceve in ingresso i vari flussi provenienti da uno o più PLP e, analizzando separatamente ogni canale, costruisce i frame in banda base (BBFRAME), ovvero unisce in pacchetti il flusso continuo di dati provenienti dal PLP.

Ogni PLP può trasportare diverse tipologie di flusso, ad esempio TS, GSE, GCS o GFPS, ognuna delle quali ha delle caratteristiche specifiche proprie. Come si nota dalla Figura 1.1a e Figura 1.1b, il blocco *Mode Adaption* è composto da:

- *Input Interface*: blocco che ha il compito di mappare i bit in ingresso in un formato logico che possa essere interpretato dal sistema. In tal caso c'è il rischio di una frammentazione dell'informazione, allorché la capacità disponibile del campo dati non sia abbastanza grande da contenere l'intero flusso di bit in arrivo.
- *Input Stream Synchronizer*: blocco che deve garantire degli strumenti idonei per avere un bit rate costante (CBR) e un costante ritardo di trasmissione end-to-end per qualsiasi tipologia di flusso. Inoltre tale blocco è utilizzato per la sincronizzazione tra trasmettitore e ricevitore, in quanto il campo di ISSY porta con sé il valore del rate del contatore di clock riguardante il demodulatore per poi utilizzarlo al ricevitore per rigenerare la corretta tempistica del flusso d'uscita.
- *Compensating delay*: alcune informazioni comuni a tutti i canali logici PLPs anziché essere trasmesse in ogni canale, vengono inviate una sola volta in un canale comune, definito appunto *common PLP*. Il blocco *compensating delay* si preoccupa allora di inserire un ritardo nel flusso di trasmissione di dati, in modo tale che il ricevitore sia in grado

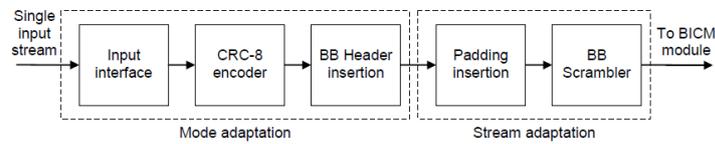
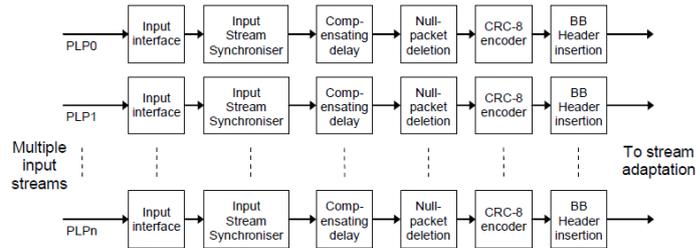
(a) *Input Processing Mode per un singolo flusso d'ingresso.*(b) *Input Processing Mode per flussi d'ingresso multipli.*

Figura 1.1: Stream Adaption.

di applicare la ricostruzione del flusso originale inserendo le eventuali informazioni presenti nel canale comune.

- *Null Packet Deletion*: la regola di trasporto richiede che, all'uscita del multiplexer del trasmettitore e all'ingresso del demultiplexer del ricevitore, il bit rate sia costante e il ritardo di una trasmissione end-to-end sia anch'esso costante. Perché si verifichi questo molte volte parte del flusso trasmesso contiene un elevato numero di pacchetti nulli; in tal caso il blocco *Null Packet Deletion* interviene per memorizzare la posizione di tale pacchetto e successivamente rimuoverlo. Importante è ricordare la posizione del pacchetto eliminato, in quanto una volta che il flusso è arrivato a destinazione esso deve essere reinserito nella medesima posizione, così da riottenere la regola iniziale riguardante il bit-rate.
- *CRC-8 Encoder*: blocco utilizzato per la rivelazione di errori a livello UP; esso può essere utilizzato solo per flussi di tipo TS o GFPS

- *BBHeader Insertion*: blocco che ha il compito di inserire un campo di lunghezza fissa di 10 bytes a monte del campo dati in banda base; tale campo è definito Header e contiene informazioni riguardanti il contenuto del campo dati (payload).

## 1.2.2 Stream Adaption

Il blocco *Mode Adaption* è seguito dal blocco *Stream Adaption*, che si compone delle seguenti parti:

- *Scheduling*: blocco che stabilisce quali pacchetti dati del segnale T2 vengono trasportati dal canale logico  $N$ -esimo ( $N$ -PLPs).
- *Padding*: blocco che aggiunge informazioni all'interno di un BBFRAME nel caso in cui esso non raggiunga il limite minimo richiesto affinché il pacchetto possa essere trasmesso. Informazioni aggiuntive possono essere inserite quando è necessario rendere pari il numero di pacchetti UP nel BBFRAME.
- *Scrambling*: blocco che esegue una randomizzazione del frame in banda base (BBFRAME) utilizzando una sequenza di scrambling generata dal registro a scorrimento di feed-back.

In conclusione, il flusso di dati in ingresso allo *Stream Adaption* è costituito da pacchetti formati da un header (BBHEADER) seguito da un campo dati (BBFRAME), all'uscita di esso si ha un pacchetto costituito dalla sola parte dati (BBFRAME). La Figura 1.2 evidenzia la struttura del BBFRAME in uscita dal blocco *Stream Adaptor*.

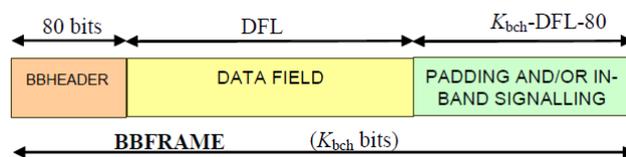


Figura 1.2: Struttura BBFRAME all'uscita del blocco Stream Adaptor.

### 1.2.3 Bit Interleaved Coding and Modulation

Lo schema del blocco BICM è mostrato in Figura 1.3. Tale sistema riceve in ingresso un flusso di BBFRAME ognuno costituito da  $K_{bch}$  bits e dà in uscita un flusso di FECFRAME.

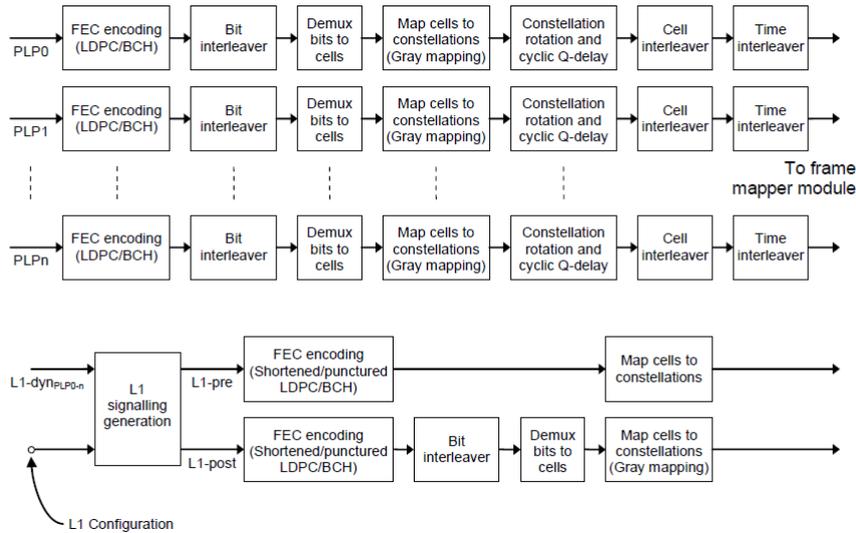


Figura 1.3: Bit Interleaved Code Modulation.

I blocchi che costituiscono il BICM sono i seguenti:

- *FEC encoding (LDPC/BCH)*: blocco che analizza ogni singolo BBFRAME e crea il corrispondente FECFRAME di lunghezza  $N_{ldpc}$  bits. Ad ogni BBFRAME viene aggiunto un numero di bits di parità riguardanti rispettivamente i codici a correzione d'errore BCH e LDPC, mentre in coda al BBFRAME vengono aggiunti rispettivamente i campi BCH-FEC e LDPCFEC.

Ogni FECFRAME sarà formato da una sequenza di 64800 bits per un normale FECFRAME oppure da 16200 bits per un corto FECFRAME.

- *Bit Interleaver*: blocco che, insieme al demultiplexer, mescola i bits della parola codificata tenendo conto che sia nel codice LDPC che nelle modulazioni QAM non tutti i bits sono protetti allo stesso modo.

Questo blocco viene utilizzato solo con costellazioni di tipo  $16\backslash 64\backslash 256$ -QAM.

- *Demux Bits to cells*: blocco che riceve in ingresso un flusso di bits ed esegue una demultiplazione di questi in  $N$  sottoflussi. Il numero  $N$  di sottoflussi dipende dal tipo di modulazione e dalla dimensione del FECFRAME.

Successivamente ogni bit di un sottoflusso entra nel demultiplatore il quale, seguendo una determinata regola definita dallo standard, passa tale bit in un sottoflusso d'uscita. Seguendo il procedimento si ottiene in uscita una sequenza di parole di bit ognuna delle quali viene suddivisa in due parti, formando così le celle che saranno poi mappate in costellazioni dal blocco successivo.

- *Map cell to a constellations*: blocco che riceve in ingresso le celle che si sono formate nel punto precedente (*Demux Bits to cells*) e, a partire da queste, forma i segnali tratti da una delle seguenti costellazioni: QPSK,  $16\backslash 64\backslash 256$ -QAM, ottenendo così in uscita un punto della costellazione non normalizzato.

- *Constellation rotation*: blocco che riceve in ingresso i valori normalizzati delle celle di ogni FECFRAME ed esegue una rotazione di un angolo appropriato della costellazione mappando i punti negli assi I e Q del piano complesso. In questo modo ciascuna coordinata del sistema ruotato è sufficiente per riconoscere il punto della costellazione, e le doppie coordinate (ascissa e ordinata di ciascun punto) risultano quindi ridondanti.

Ruotando la costellazione ed inserendo un ritardo ciclico per l'ordinata Q prima dell'interleaving in tempo e frequenza, ascissa e ordinata di ciascun punto sono trasferite su differenti celle OFDM.

- *Cell interleaver*: blocco che applica una permutazione pseudo-random alle celle all'interno di un blocco FEC. Esso permette di rompere la

regolarità della struttura dell'interleaver di tempo, ed evita possibili interazioni con la struttura del codice LDPC.

- *Time interleaver*: blocco che separa le celle appartenenti ad un blocco di codifica FEC su più simboli OFDM e potenzialmente su più trame. Consente di proteggere il segnale dai disturbi di tipo impulsivo concentrati nel tempo.

### 1.2.4 Frame Builder

La funzione del *Frame Builder* rappresentato in Figura 1.4 è di assemblare le celle prodotte dal blocco *time-interleaver* di ogni PLP e le celle della segnalazione L1 in matrici di celle attive OFDM, corrispondenti a ciascuno dei simboli OFDM che costituiscono la cornice complessiva della struttura. Il *Frame Builder* opera secondo le informazioni dinamiche prodotte dallo scheduler.

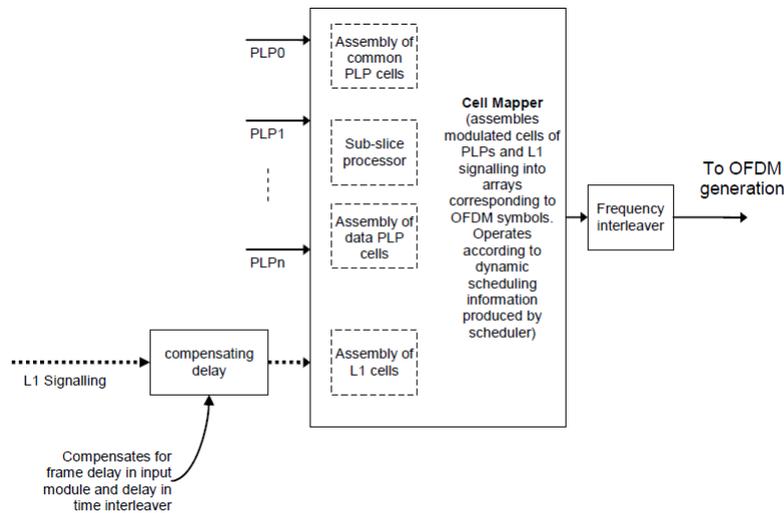


Figura 1.4: Struttura Frame Builder.

### Struttura di un frame

La struttura di trama DVB-T2 consiste al livello più alto di super-frames, i quali hanno una durata massima di 64 s quando non è utilizzato il campo FEF: questi sono poi suddivisi in frame, che a loro volta sono ulteriormente ripartiti in simboli OFDM.

Il campo FEF è un intervallo temporale lasciato libero dal segnale DVB-T2 per eventuali servizi da introdurre successivamente e non ancora definiti.

Ogni frame si compone di tre parti ben individuate: un simbolo di riferimento P1, uno o più simboli P2, un numero configurabile di simboli dati OFDM. La durata di ogni trama è dell'ordine di  $100 \div 250ms$ .

Il simbolo P1 è costituito da un simbolo OFDM su 1K portanti e da una particolare struttura di replicazione nel tempo. Tale campo ha il compito di aiutare il ricevitore ad effettuare una scansione veloce delle frequenze trasmissive.

I simboli P2 trasportano la segnalazione di livello 1 (L1 signalling), che in presenza di PLP multipli può essere quantitativamente rilevante, in quanto ogni canale logico, come già detto in precedenza, ha i propri parametri di trasmissione. Inoltre i simboli P2 contengono le cosiddette portanti pilota, celle dati OFDM di ampiezza e fase note che vengono trasmesse ad intervalli di tempo regolari per permettere al ricevitore di iniziare la procedura di stima del canale trasmissivo.

#### 1.2.5 OFDM Generator

L'*OFDM Generator* è l'ultima parte del sistema DVB-T2 e riceve in ingresso le celle prodotte dal *Frame Builder*, inserisce alcune informazioni rilevanti conosciute come *piloti*, aggiunge poi gli intervalli di guardia e, se necessario, applica una riduzione del rapporto tra il valore di picco e il valore medio di potenza (PAPR), ottenendo così il segnale completo richiesto dallo standard DVB-T2.

La struttura dell'*OFDM Generator* è costituita, come si può osservare dalla

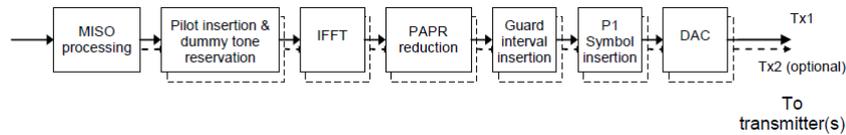


Figura 1.5: Struttura OFDM Generator.

Figura 1.5, dai seguenti blocchi:

- *MISO processing*: blocco che riceve in ingresso le celle dati prodotte dal *Frame Builder* e produce in uscita due insiemi di celle dati simili, ognuna delle quali sarà diretta a due gruppi di trasmettitori. Non sempre tale blocco è utilizzato, in quanto la modalità MISO può essere non supportata; in quest'ultimo caso le celle che arrivano in ingresso a tale blocco vengono mandate direttamente all'uscita senza alcuna elaborazione.
- *Pilot Insertion & dummy tone reservation*: blocco che inserisce nel frame portanti pilota, cioè celle dati OFDM di cui si conosce il contenuto. Le informazioni trasmesse in queste celle sono di tipo Scattered, Continual, Edge o P2. Il valore delle informazioni pilota deriva da una sequenza di riferimento, che è una serie di valori, una per ogni portante trasmessa per un dato simbolo.
- *IFFT*: blocco che esegue la trasformata di Fourier inversa dei simboli OFDM.
- *PAPR reduction*: blocco che impone dei vincoli sul funzionamento degli amplificatori di potenza nei trasmettitori, in quanto con l'aumentare del numero di portanti il segnale OFDM tende ad assomigliare a rumore Gaussiano, con la conseguenza che il rapporto tra potenza di picco e potenza media (PAPR) risulta essere elevato. Due sono le tecniche per diminuire il PAPR:

- ACE, che prevede la distorsione delle costellazioni di modulazione muovendo i punti più esterni della costellazione trasmessa verso locazioni ad ampiezza maggiore.
- TR, che prevede la cancellazione dei picchi di rumore direttamente nel dominio temporale.

La prima tecnica riduce il PAPR senza provocare alcuna perdita in capacità di trasmissione, mentre con la seconda si ha un po' di perdita di capacità trasmissiva, in quanto alcune portanti vengono utilizzate per rilevare i picchi di rumore.

- *Guard Interval Insertion*: blocco che inserisce un intervallo di tempo contenente dei dati, in modo tale da separare i simboli OFDM adiacenti da quello preso in considerazione.

Questa aggiunta viene fatta in quanto una porzione di segnale OFDM è affetta da interferenza intersimbolica, ossia interferenza sul simbolo utile dovuta ai simboli che lo precedono. Per eliminare quindi tale problema si inserisce un cosiddetto intervallo di guardia prima del simbolo OFDM da trasmettere, il quale ha una durata pari a  $T_g$ , corrispondente ad una frazione della durata di simbolo utile, che separa il simbolo OFDM da quelli adiacenti.

- *P1 symbol insertion*: blocco che inserisce il simbolo P1 come preambolo del frame.

Un simbolo P1 è costituito da 1K simboli OFDM con due 1/2 intervalli di guardia aggiunti; esso ha una durata di 224  $\mu\text{s}$  su un sistema da 8 MHz. In questo intervallo di tempo di 224  $\mu\text{s}$  è compresa la durata della parte utile 'A' del simbolo più due sezioni di intervalli di guardia 'C' e 'B', rispettivamente di durata 59  $\mu\text{s}$  (542 campioni) e 53  $\mu\text{s}$  (482 campioni).

- *DAC*: componente elettronico in grado di produrre sul suo terminale di uscita un determinato livello di tensione o di corrente, in funzione

di un valore numerico che viene presentato al suo ingresso. Esegue una conversione da segnale digitale a segnale analogico.



## Capitolo 2

# Modulazione OFDM

L'OFDM è una tecnica di modulazione presente in quasi tutti i sistemi di trasmissione attuali, caratterizzata dalla suddivisione del segnale di informazione ad alta velocità trasmissiva in molti flussi paralleli a bassa velocità, multiplati a divisione di frequenza su portanti ortogonali e quindi non interferenti tra loro.

La modulazione OFDM è utilizzata soprattutto per eliminare il problema del multipath. Tale problema deriva dal fatto che il segnale, durante la propagazione dal trasmettitore al ricevitore, si riflette sugli ostacoli che incontra nel percorso; di conseguenza al ricevitore arrivano più repliche del segnale trasmesso le quali sono attenuate, sfasate e ritardate in modo diverso a seconda del tragitto effettuato.

Lavorando nel dominio del tempo il canale può essere considerato come un filtro avente risposta impulsiva  $h(t)$ . In uscita al canale il segnale originale trasmesso risulta essere distorto: per ovviare a ciò viene aggiunta al termine utile una funzione che tiene conto del rumore, il quale sarà indicato con  $w(t)$ . In generale, lo schema di una modulazione OFDM è basato su un banco di filtri i quali generano segnali a banda stretta attorno alle frequenze delle portanti, per poi sommarli e trasmetterli sul canale.

L'idea base che sta dietro alla modulazione OFDM è quella di considerare un canale tempo-invariante per un tempo sufficientemente lungo, ma non infini-

to, e lavorare nel dominio della frequenza utilizzando quindi le trasformate di Fourier. Sappiamo che per canali lineari tempo-invarianti le sinusoidi sono autofunzioni, ma poiché la trasmissione ha una durata finita, le sinusoidi perdono tale proprietà.

Per recuperare questa caratteristica si aggiunge ad ogni simbolo trasmesso un prefisso ciclico, che consiste nell'inserimento di una parte di informazione presa dalla coda del simbolo e riposta in testa al simbolo stesso.

Questa tecnica, come vedremo successivamente, consente di eliminare l'interferenza di intersimbolo (ISI), problema che si verifica nei sistemi digitali a singola portante proprio perché i segnali trasmessi non sono sinusoidi, che sono di durata infinita.

Consideriamo un modello in banda base a tempo discreto, con relazione ingresso-uscita data da:

$$y(m) = \sum_{l=0}^{L-1} h_l x(m-l) + w(m) \quad (2.1)$$

dove  $h_l$  è la risposta in frequenza del canale e  $L$  indica il numero di valori (finito) in ingresso al sistema.

Identifichiamo ora, mediante il vettore  $\mathbf{d}$ , un insieme di  $N_c$  simboli:

$$\mathbf{d} = [d[0], d[1], \dots, d[N_c - 1]]^T \quad (2.2)$$

A tale blocco di  $N_c$  simboli aggiungiamo il prefisso ciclico di lunghezza  $L-1$ , ottenendo così il seguente blocco d'ingresso:

$$\mathbf{x} = [d[N_c - L + 1], d[N_c - L + 2], \dots, d[N_c - 1], d[0], d[1], d[N_c - 1]]^T \quad (2.3)$$

Applicando  $\mathbf{x}$  in ingresso al canale, otteniamo in uscita il seguente segnale al tempo  $m$ :

$$y(m) = \sum_{l=0}^{L-1} h_l x(m-l) + w(m) \quad m = 1, \dots, N_c + L - 1 \quad (2.4)$$

Il dispositivo che riceve il segnale  $y(m)$  elimina il prefisso ciclico, ovvero considera il segnale a partire dall'istante  $m \in [L, N_c + L - 1]$  e l'uscita

sull'intervallo di interesse di lunghezza  $N_c$  vale:

$$y(m) = \sum_{l=0}^{L-1} h_l d[(m - L - l) \text{ modulo } N_c] + w(m) \quad (2.5)$$

La relazione (2.5) altro non è che la convoluzione tra la risposta impulsiva del canale  $\mathbf{h}$  e i simboli d'ingresso  $\mathbf{d}$ , a cui si aggiunge un determinato rumore  $\mathbf{w}$ , ovvero:

$$\mathbf{y} = \mathbf{h} \otimes \mathbf{d} + \mathbf{w} \quad (2.6)$$

dove:  $y = [y(L), \dots, y(N_c + L - 1)]^T$  rappresenta l'uscita del sistema,  $\mathbf{h} = [h_0, h_1, \dots, h_{L-1}, 0, \dots, 0]^T$  costituisce il canale ed infine  $\mathbf{w} = [w(L), \dots, w(N_c + L - 1)]^T$  indica il vettore del rumore, il quale è costituito da  $N_c$  variabili aleatorie Gaussiane i.i.d. con media nulla e varianza pari ad  $N_0$ .

Richiamando ora la teoria di Fourier, ed in particolar modo la trasformata di Fourier a tempi discreti (DFT), risulta che nel dominio della frequenza l'operazione di convoluzione si trasforma in un prodotto tra le trasformate di Fourier dei singoli elementi, quindi la relazione (2.6) diventa:

$$Y_n = H_n D_n + W_n \quad \text{con } n = 0, \dots, N_c - 1 \quad (2.7)$$

dove

$$D_n = \frac{1}{\sqrt{N_c}} \sum_{m=0}^{N_c-1} d[m] \exp\left(\frac{-j2\pi nm}{N_c}\right) \quad (2.8)$$

rappresenta la DFT dei simboli di ingresso, e

$$H_n = \sum_{l=0}^{N_c-1} h_l \exp\left(\frac{-j2\pi nl}{N_c}\right) \quad (2.9)$$

rappresenta la DFT dell' $i$ -esima portante di  $\mathbf{h}$  del canale.

Lo schema a blocchi per la trasmissione e ricezione per una modulazione OFDM è dato dalla Figura 2.1.

Da tale figura si può notare come il canale venga suddiviso in un insieme di sottocanali posti in parallelo e privi di interferenza intersimbolica, i quali non interferiscono tra loro. Inoltre si può osservare come il vettore finale  $\mathbf{Y}$  sia relazionata con il vettore di simboli di ingresso  $\mathbf{D}$  dalla relazione (2.7).

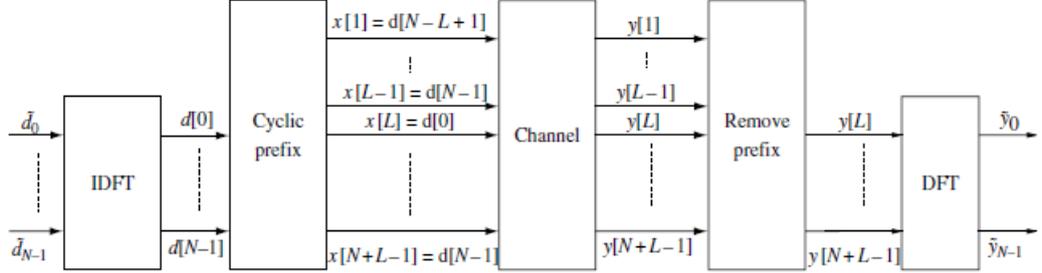


Figura 2.1: Schema per la trasmissione e ricezione modulazione OFDM.

In conclusione, lo schema OFDM anziché trasmettere l'intero flusso in un solo canale a banda larga esegue la suddivisione del canale in sottocanali paralleli a banda molto stretta, eliminando il problema del multipath. Questa tecnica di modulazione provoca però due importanti difetti, dovuti alla presenza del prefisso ciclico utile per eliminare l'interferenza intersimbolica:

- *perdita di tempo*: il prefisso ciclico, come già sottolineato in precedenza, è costituito da una ripetizione di informazione già presente all'interno del simbolo da trasmettere; ciò significa che viene utilizzato del tempo per trasmettere due volte la stessa informazione, anziché trasmettere dati utili. Questa perdita ammonta a circa una frazione  $\frac{L}{(N_c+L)}$  del tempo totale.
- *perdita di potenza*: poiché la trasmissione del prefisso ciclico richiede una determinata potenza, essa viene sottratta alla trasmissione di informazione utile. Questa perdita è pari a una frazione  $\frac{L}{(N_c+L)}$  della potenza media.

Per risolvere questi due problemi si deve ricorrere ad un simbolo costituito da un numero elevato  $N_c$  di campioni; in questo modo la lunghezza del prefisso ciclico diventa irrilevante rispetto alla lunghezza del simbolo, di conseguenza anche il tempo e la potenza utilizzati per trasmettere tale parte di informazione diventano trascurabili.

# Capitolo 3

## Costellazione Ruotata

### 3.1 Principio di funzionamento

Nello standard DVB-T l'ordine massimo della costellazione utilizzata per la modulazione è una  $64$ -QAM. Con il nuovo standard DVB-T2 si è passati all'utilizzo di una costellazione  $256$ -QAM, la quale consente di trasportare 8 bits per simbolo OFDM, anziché i 6 bits trasportati con la modulazione  $64$ -QAM. L'aumento di 2 bits per ogni simbolo consente un incremento dell'efficienza spettrale del 33%, nonché una maggiore capacità di trasporto per un dato rate di codifica.

Lo standard DVB-T2 inoltre introduce, nell'ambito delle possibili opzioni di modulazione, una nuova tecnica basata sulla rotazione della costellazione.

Si ipotizzi di avere a disposizione una costellazione  $M$ -QAM non ruotata, considerando la Figura 3.1a, la quale rappresenta per semplicità una costellazione  $16$ -QAM: si può osservare come i punti della costellazione 1000, 1010, 0010 e 0000 abbiano in comune la stessa componente in quadratura Q. Questo si traduce nel fatto che il ricevitore non è in grado di capire quale informazione è stata trasmessa dal trasmettitore a partire dalla conoscenza di una delle due componenti, se la componente in fase I o quella in quadratura Q.

Lo stesso ragionamento può essere fatto nel caso in cui siano presi in considerazione i simboli 0000, 0001, 0101, 0100, avendo questi la medesima com-

ponente in fase I.

La conoscenza di I infatti non offre alcuna informazione per quanto riguarda la componente Q e viceversa.

In conclusione un punto della costellazione è individuato in modo univoco dalla coppia (I,Q). Inoltre sapendo che, una volta trasmesso il simbolo nel canale, questo arriverà al ricevitore affetto da un rumore, entrambe le componenti (I,Q) soffriranno della stessa distorsione, in quanto sono trasportate con la medesima frequenza.

Si ipotizzi ora di effettuare una rotazione della costellazione di un determinato angolo  $\Phi$ , come rappresentato in Figura 3.1b. Ciò significherà che non esisteranno più punti della costellazione con una coordinata uguale e di conseguenza, essendo eliminato ogni fraintendimento tra la componente in fase I e quella in quadratura Q, il ricevitore sarà in grado di riconoscere il punto della costellazione trasmesso, ovvero un simbolo della costellazione sarà individuato in modo univoco da una sola componente: I o Q.

Questo fatto è molto importante in presenza di rumore, ma non risolve completamente il problema, in quanto se le due componenti vengono trasmesse con la stessa portante, esse sono soggette alla stessa attenuazione.

Per aumentare la robustezza del sistema le due componenti I e Q vengono

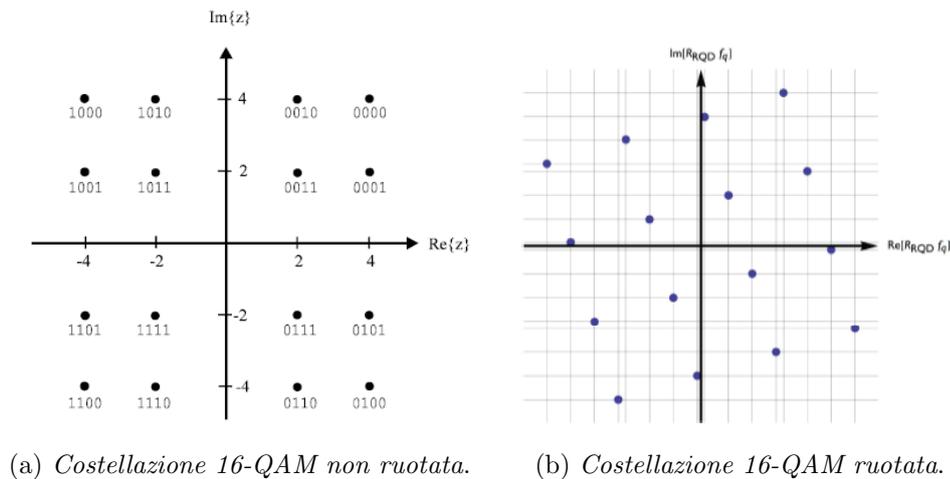


Figura 3.1: Costellazione 16-QAM.

separate dall'interleaving prima della trasmissione, venendo così trasferite su frequenze differenti in istanti differenti, cioè in diverse celle OFDM. In questo modo, se una delle due componenti viene persa o è affetta da rumore, l'altra componente può essere utilizzata per ricostruire l'informazione, e ciò può avvenire perché l'attenuazione e il rumore sulle due coordinate (I,Q) sono indipendenti.

Un'osservazione deve essere fatta riguardo all'angolo di rotazione della costellazione, più precisamente riguardo all'ampiezza di  $\Phi$  sufficiente per ottenere un effettivo aumento delle prestazioni del sistema.

L'angolo di rotazione  $\Phi$  dipende dalla tipologia di modulazione scelta, per una QPSK  $\Phi = 29.0$ , per una 16-QAM  $\Phi = 16.0$ , per una 64-QAM  $\Phi = 8.6$  e infine per 256-QAM  $\Phi = \arctan(1/16)$ .

Un'altra annotazione riguarda la tipologia di canale utilizzato per la trasmissione dei dati: nel caso di canali gaussiani l'utilizzo della costellazione ruotata non degrada le prestazioni, mentre con canali affetti da fading tale tecnica fornisce un guadagno di circa 0.7 dB. Il guadagno che si ottiene viene poi sfruttato nell'utilizzo di codici a correzione d'errore FEC, con rapporti di codifica più elevati e quindi velocità di trasferimento maggiore.

Quanto sopra esposto in modo discorsivo può essere espresso nel seguente modo in termini matematici.

Si ipotizzi di avere in ingresso al trasmettitore una sequenza di bit  $\{b_l\}$  con un periodo di bit  $T_b$ . Tale sequenza è posta in ingresso ad un bit-mapper il quale mappa i bit in una sequenza di simboli  $D_k$ , ogni simbolo è distanziato dal simbolo successivo di un intervallo pari al periodo di simbolo T, tale intervallo di tempo è legato al periodo di bit dalla seguente relazione  $T = T_b \log_2 M$ .

All'istante  $k$  il simbolo QAM è dato dalla seguente relazione:

$$X_k = D_{k,I} + jD_{k,Q} \quad (3.1)$$

dove  $D_{k,I} = \Re\{D_k\}$  e  $D_{k,Q} = \Im\{D_k\}$  costituiscono rispettivamente la componente in fase e quella in quadratura del simbolo.

Successivamente le due componenti vengono modulate rispettivamente dalla portante  $\cos(2\pi f_k t)$  e dalla portante  $-\sin(2\pi f_k t)$ . Sommando tra di esse le due componenti otteniamo il segnale QAM che viene posto nel canale di trasmissione.

$$X_k(t) = D_{k,I} \cos(2\pi f_k t) - D_{k,Q} \sin(2\pi f_k t) \quad (3.2)$$

Nel caso di una costellazione ruotata abbiamo una rotazione di un angolo  $\Phi$  della costellazione; questo significa che il segnale QAM, espresso dalla relazione (3.2), viene moltiplicato di una quantità pari ad  $\exp(j\Phi)$ , ovvero si ottiene il seguente segnale:

$$Z_k(t) = D_{k,I} \exp(j\Phi) \cos(2\pi f_k t) - D_{k,Q} \exp(j\Phi) \sin(2\pi f_k t) \quad (3.3)$$

Come espresso in precedenza mediante la modulazione OFDM, la parte reale del simbolo viene trasmessa mediante la portante  $k$ -esima, mentre la parte immaginaria viene trasmessa mediante la portante  $k+1$ -esima.

Come già rilevato il canale introduce un rumore, quindi il simbolo inviato arriva al ricevitore nel seguente modo:

$$Y_k = H_k Z_k + w_k \quad (3.4)$$

$$Y_{k+1} = H_{k+1} Z_{k+1} + w_{k+1} \quad (3.5)$$

Si osservi che il canale introduce una certa amplificazione/attenuazione ed una ulteriore rotazione ( $\Phi_{H_k}$ ) del simbolo; possiamo quindi esprimere la risposta in frequenza del canale nel seguente modo  $H_k = |H_k| \exp(j\Phi_{H_k})$ .

Il segnale in ingresso al ricevitore si presenta nella seguente forma:

$$Y_k = |H_k| Z_k \exp(j\Phi_{H_k}) + w_k \quad (3.6)$$

$$Y_{k+1} = |H_{k+1}| Z_{k+1} \exp(j\Phi_{H_{k+1}}) + w_{k+1} \quad (3.7)$$

Il ricevitore, una volta ricevuto il segnale, elimina la rotazione introdotta dal canale, per fare ciò moltiplica entrambe le relazioni (3.6) e (3.7) rispettivamente per  $\exp(-j\Phi_{H_k})$  e per  $\exp(-j\Phi_{H_{k+1}})$  ottenendo:

$$P_k = [|H_k|Z_k + w_k] \exp(-j\Phi_{H_k}) \quad (3.8)$$

$$P_{k+1} = [|H_{k+1}|Z_{k+1} + w_{k+1}] \exp(-j\Phi_{H_{k+1}}) \quad (3.9)$$

In conclusione, estraendo dalla relazione (3.8) e dalla (3.9) rispettivamente la parte reale e la parte immaginaria, ed unendo le due componenti, si è in grado di ricostruire al ricevitore il simbolo trasmesso:

$$Q_k = [|H_k|\Re(Z_k) + \Re(w_k)] + j[|H_{k+1}|\Im(Z_{k+1}) + \Im(w_{k+1})] \quad (3.10)$$

Si osservi che idealmente  $Q_k = X_k$ , e di conseguenza anche la costellazione ricevuta è uguale a quella trasmessa, mentre nella pratica tale uguaglianza non è verificata per due semplici ragioni, già accennate in precedenza:

- le celle nelle quali le componenti in fase I e in quadratura Q viaggiano, sono soggette a fenomeni di amplificazione e/o attenuazione;
- il canale introduce un determinato rumore che varia da portante a portante utilizzata.

A causa di questo si dovranno cercare dei metodi che siano in grado di ricostruire il segnale e la costellazione il più possibile uguali a quelli trasmessi, a partire da quello ricevuto  $Q_k$ . I prossimi due paragrafi descrivono due tecniche per raggiungere tale risultato.

## 3.2 Criterio hard decision

In generale il modello per la trasmissione digitale è costituito da tre principali elementi: il *modulatore digitale*, il *mezzo trasmissivo* e il *demodulatore*

*digitale.*

Il primo blocco, il *modulatore digitale*, ha il compito di trasformare la sequenza d'ingresso di bit  $\{b_l\}$  in un segnale continuo nel tempo  $s_{tx}(t)$ . Il modulatore può implementare un insieme di  $M=2$  forme d'onda per generare il segnale da trasmettere sul canale. In questo caso si parla di *modulazione binaria*, o più in generale esso può utilizzare un numero di forme d'onda  $M > 2$ , in tal caso si ha una *modulazione M-aria*.

Il secondo blocco, il *mezzo trasmissivo*, trasporta semplicemente il segnale  $s_{tx}$  dalla sorgente alla destinazione. Si osservi che, come già sottolineato in precedenza, il canale introduce distorsione, interferenza e rumore, di conseguenza il segnale in uscita al canale  $Y(t)$  risulta essere diverso dal segnale trasmesso. La relazione che lega i due segnali è la seguente:

$$Y(t) = (s_{tx} \otimes h_{ch})(t) + w_{rc}(t) \quad (3.11)$$

dove  $h_{ch}$  indica la risposta impulsiva del canale e  $w_{rc}$  è il rumore additivo gaussiano bianco, avente media zero e densità spettrale di potenza  $P_w(f) = N_0/2$ .

Il terzo ed ultimo blocco, il *demodulatore digitale*, esegue un'ulteriore trasformazione, passando dal segnale a tempo continuo ricevuto  $Y(t)$  ad una sequenza di bit  $\hat{b}_l$ . In sostanza, basandosi sul segnale ricevuto  $Y(t)$  il demodulatore digitale cerca di ricostruire al ricevitore la sequenza di bit inviata dalla sorgente. Si osservi che i bit individuati  $\hat{b}_l$  possono essere affetti da errori, quindi potrebbe essere che  $\hat{b}_l \neq b_l$  per alcuni valori di  $l$ , disuguaglianza dovuta alla distorsione introdotta dal canale.

La tecnica *Hard Decision* esegue l'ultimo passo appena descritto, ovvero ricerca una regola per l'individuazione dei bit trasmessi a partire da un segnale ricevuto. La teoria per stabilire il criterio di decisione si basa su tre importanti elementi:

- *Suddivisione dello spazio  $R^I$* : lo spazio  $R^I$  viene suddiviso in  $M$  regioni disgiunte  $R_n$ , ad ognuna delle quali è associato un possibile valore di

uscita  $b_0$ . In definitiva si deve avere che:

$$\bigcup_{n=1}^M R_n = R^I \quad e \quad R_m \cap R_n = \emptyset, \quad m \neq n \quad (3.12)$$

- *Regola di decisione*: deve essere adottata una regola tale per cui ad un dato segnale d'ingresso deve essere associato un determinato bit d'uscita; la regola da considerare è così esprimibile:

se  $Y \in R_m$  allora scegli l'ipotesi  $m$ -esima e  $\hat{b}_0 = m$ .

$$(3.13)$$

- *Ottimizzazione regioni di decisione*: la modalità con cui vengono scelte le regioni di decisione è molto importante, in quanto influenza le prestazioni del sistema. Le regioni vengono scelte in modo tale che la probabilità di corretta decisione  $P[C]$  sia massimizzata. La probabilità di corretta decisione è così esprimibile:

$$P[C] = P[\hat{b}_0 = b_0] \quad (3.14)$$

Dalla definizione (3.14) le regioni di decisione sono così ricavabili:

$$\{R_{n,opt} = \operatorname{argmax}_{\{R_n\}} P[C]\} \quad (3.15)$$

Un ulteriore modo per ricavare le regioni di decisione è quello di considerare la probabilità d'errore  $P[E]$  anziché la probabilità di corretta decisione. La probabilità d'errore è così definita:

$$P[E] = P[\hat{b}_0 \neq b_0] \quad (3.16)$$

Tale probabilità è esprimibile anche nel seguente modo:

$$P[E] = 1 - P[C] \quad (3.17)$$

Considerando la definizione (3.16), le regioni di decisione vengono ricavate con la seguente regola:

$$\{R_{n,opt} = \operatorname{argmin}_{\{R_n\}} P[E]\} \quad (3.18)$$

Spesso è difficile conoscere la probabilità di corretta decisione, quindi per semplificare il calcolo si applicano la formula di Bayes e il teorema della probabilità totale, eseguendo un condizionamento sul simbolo trasmesso, per cui si ottiene la seguente relazione:

$$P[C] = \sum_{n=1}^M P[\hat{b}_0 = n | b_0 = n] P[b_0 = n] = \sum_{n=1}^M P[Y \in R_n | b_0 = n] P[b_0 = n] \quad (3.19)$$

Ovviamente, in modo duale, si ottiene la relazione per la probabilità d'errore condizionando sul simbolo trasmesso:

$$P[E] = \sum_{n=1}^M P[\hat{b}_0 \neq n | b_0 = n] P[b_0 = n] = \sum_{n=1}^M P[Y \notin R_n | b_0 = n] P[b_0 = n] \quad (3.20)$$

Nella pratica per la costruzione delle regioni di decisione sono disponibili tre importanti criteri, i quali si basano tutti sulla definizione di probabilità di corretta decisione e probabilità d'errore, definite rispettivamente nelle relazioni (3.19) e (3.20):

- ML: consiste nello scegliere il vettore appartenente alla regione  $R_n$  della costellazione che ha la maggior probabilità di coincidere col segnale emesso. Tale criterio è utilizzabile nel momento in cui i simboli trasmessi sono equiprobabili, ovvero quando  $P[b_0] = 1/M$ , con  $M$  che indica il numero di punti della costellazione.
- MAP: metodo uguale al criterio ML ma viene tenuto in considerazione anche il fatto che non necessariamente tutti i segnali appartenenti alla costellazione hanno la stessa probabilità di essere emessi.
- MD: consiste nello scegliere il simbolo che ha distanza più vicina al punto emesso.

Di seguito verrà analizzato in maggior dettaglio il criterio a minima distanza, utilizzando anche un riferimento esemplificativo.

La regola di decisione nel caso si utilizzi tale metodo è la seguente:

$$\hat{b}_0 = \operatorname{argmin}_{n \in \{1, \dots, M\}} d(\rho, q_n) \quad (3.21)$$

dove  $d(\rho, q_n)$  è la distanza tra il punto della costellazione e il punto ricevuto,  $\rho$  una realizzazione del segnale ricevuto e  $Q_n$  il segnale ricevuto.

Dalla definizione di cui sopra, le regioni di decisione ottime sono così determinate:

$$R_m = \{\rho | m = \operatorname{argmin}_{n \in \{1, \dots, M\}} d(\rho, q_n)\} \quad (3.22)$$

### 3.2.1 Costruzione regioni di decisione

Si ipotizzi di avere un sistema digitale che applica una modulazione QAM, per semplicità si assuma che  $M = 16$ , ma si può anche assumere  $M = 4 \setminus 64 \setminus 256$ . La costellazione al trasmettitore (Figura 3.2) è costituita da  $M = 16$  punti, ognuno identificato da componente in fase  $D_{k,I}$  e componente in quadratura  $D_{k,Q}$ , ottenendo così il punto:

$$X_k = D_{k,I} + j * D_{k,Q} \quad \text{con } k = 1, \dots, M \quad (3.23)$$

Applicando ora il criterio della minima distanza vengono create le regioni di decisione (Figura 3.3); per fare ciò si considera un punto della costellazione identificato dalla relazione (3.23) e, per ogni punto ad esso adiacente, si calcola il punto a distanza minima, successivamente si traccia un segmento ortogonale al segmento congiungente i due punti adiacenti e passante per il punto a distanza intermedia. Ripetendo tale procedimento per ogni elemento ciò che si ottiene per una  $M$ -QAM è un insieme di celle quadrate le quali individuano appunto le regioni di decisione per tale costellazione.

La distanza tra due punti della costellazione è ottenibile dalla seguente relazione:

$$d(m, n) = \|(D_{m,I} + j * D_{m,Q}) - (D_{n,I} + j * D_{n,Q})\| \quad (3.24)$$

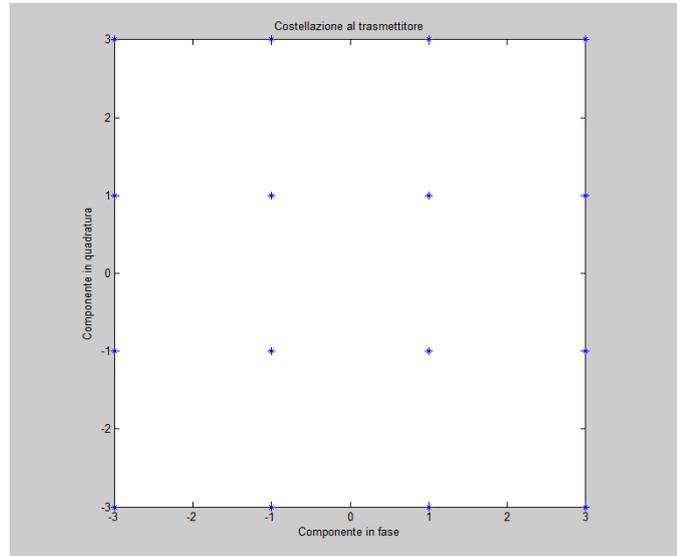


Figura 3.2: Costellazione 16-QAM al trasmettitore.

mentre la distanza minima è data da:

$$d_{min}(m, n) = \left\{ \frac{\| (D_{m,I} + j * D_{m,Q}) - (D_{n,I} + j * D_{n,Q}) \|}{2} \right\} \quad (3.25)$$

dove  $m \neq n$  e  $m, n = 1, \dots, M$ , inoltre si è indicato con  $\|x\|$  il modulo di un valore complesso.

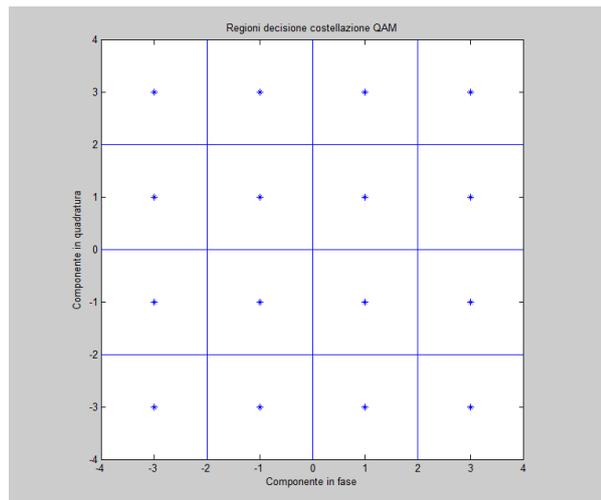


Figura 3.3: Regioni di decisione Costellazione 16-QAM.

Come già espresso nel capitolo precedente, lo standard DVB-T2 a differenza dello standard DVB-T prima di trasmettere il segnale nel mezzo trasmissivo esegue una rotazione della costellazione, in modo tale che ogni punto sia individuato dalla conoscenza di una sola componente, o quella in fase o quella in quadratura.

Utilizzando le specifiche dello standard DVB-T2 per una costellazione 16-QAM, affinché le prestazioni del sistema siano incrementate, la rotazione deve essere pari ad un angolo  $\Phi = 16$  gradi. Ogni punto  $X_k$  della (3.23) deve essere quindi ruotato di un angolo  $\Phi$ ; la relazione che consente di effettuare ciò è la seguente:

$$Z_k = (X_{k,I} + j * X_{k,Q})(\cos(\Phi) + j * \sin(\Phi)) \quad (3.26)$$

con  $k = 1, 2, \dots, M$ . Ciò che si ottiene è la seguente costellazione:

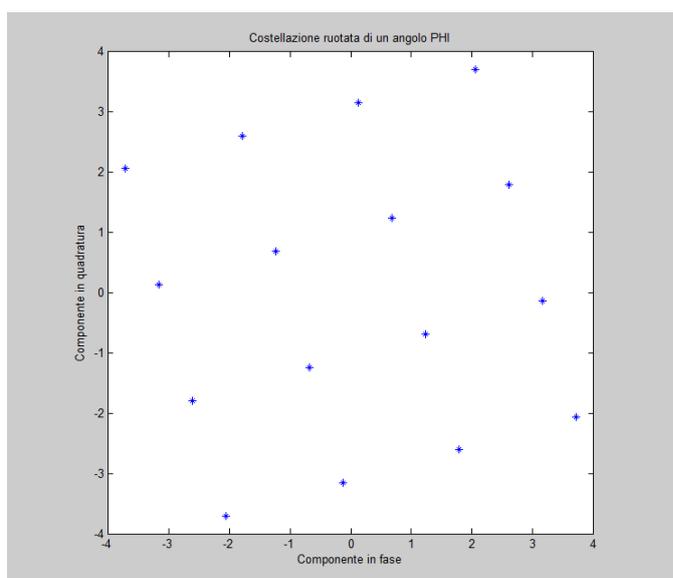


Figura 3.4: Costellazione 16-QAM ruotata di un angolo  $\Phi=16$ .

Applicando nuovamente il criterio della regione di decisione minima, ovvero eseguendo il procedimento sopra indicato per il caso in cui la costellazione non sia ruotata, otteniamo le regioni di decisione di Figura 3.5.

Si osservi che, a questo punto anche le regioni di decisione risultano essere

ruotate dello stesso angolo  $\Phi$  di cui è stata ruotata la costellazione rispetto al caso non ruotato. Basandosi su tale osservazione, si è applicato il seguente metodo per tracciare i segmenti che individuano le celle: si sono presi due punti qualsiasi appartenenti all'asse delle ordinate, quindi un punto di coordinata  $A = 0 + j*s$  e  $B = 0 - j*s$  con  $s \in (+\infty, -\infty)$ . Tali punti sono stati successivamente ruotati dello stesso angolo  $\Phi$  con cui è stata ruotata l'intera costellazione, quindi:

$$A_{rot} = (0 + j * s)(\cos(\Phi) + j * \sin(\Phi)) \quad (3.27)$$

e

$$B_{rot} = (0 - j * s)(\cos(\Phi) + j * \sin(\Phi)) \quad (3.28)$$

Tracciando poi una retta passante per i due punti  $A_{rot}$ ,  $B_{rot}$ , si ottiene la retta a distanza minima tra i punti della costellazione. Applicando tale procedimento per un numero di volte pari a  $\sqrt{M}-1$ , con  $M$  ordine della costellazione, e facendo variare la parte reale dei punti  $A$  e  $B$  di una quantità pari alla distanza tra due punti della costellazione, distanza calcolata mediante la relazione (3.24), otteniamo le rette parallele alla retta tracciata in precedenza.

Intersecando tali rette con le rette trovate applicando il medesimo procedimento di cui sopra, ma a partire da due punti sull'asse delle ascisse,  $C = s + j*0$  e  $B = s - j*0$ , e facendo variare ora la parte immaginaria dello stesso valore pari alla distanza tra due punti della costellazione, si ottengono le regioni di decisione per la costellazione ruotata.

Come già menzionato all'inizio del paragrafo, il canale introduce una distorsione del segnale, di conseguenza i punti della costellazione individuati al ricevitore subiscono un'amplificazione o attenuazione.

In questo caso si è supposto che il canale sia assimilabile ad un filtro avente risposta impulsiva  $h_k(t) = A_0\delta(t-t_0)$ , ovvero risposta in frequenza  $H_k(f) = A_0 \exp(-j2\pi ft_0)$ , con  $k = 0, \dots, M$  che indica la portante e  $A_0$  indica il fattore di amplificazione ( $|A_0| > 1$ ) o attenuazione ( $|A_0| < 1$ ).

Nella Figura 3.6, dove è visualizzata la costellazione costruita al ricevitore a partire dal segnale ricevuto  $Q_k$ , si è assunto che il canale introduca

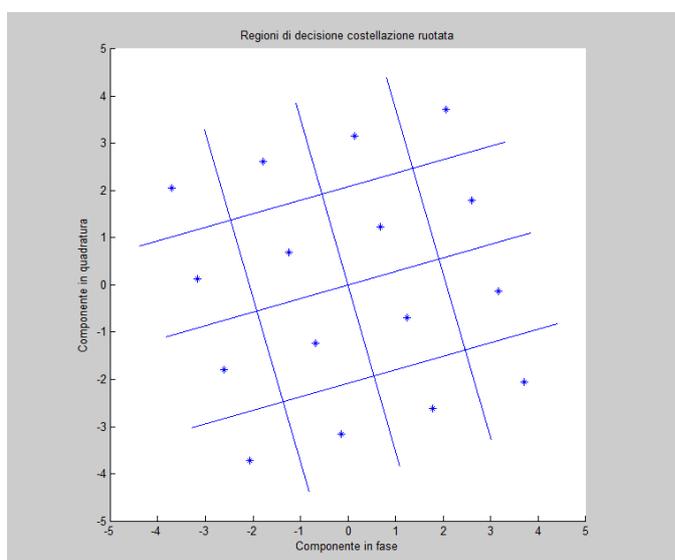


Figura 3.5: Regioni di decisione costellazione ruotata di un angolo  $\Phi=16$ .

un'attenuazione di  $A_0 = 1/2$  sulla componente in fase ed un'amplificazione di  $A_0 = 3/2$  sulla componente in quadratura. Ricordiamo che la componente in fase e quella in quadratura dello stesso simbolo non subiscono la stessa distorsione dal canale, perché esse vengono inviate utilizzando portanti diverse.

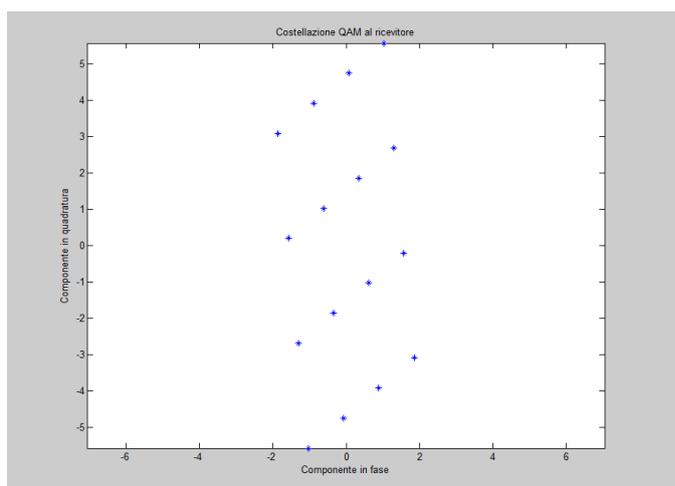


Figura 3.6: Costellazione al ricevitore.

Applicando ora alla costellazione di Figura 3.6 il criterio della minima distanza MD, ciò che si ottiene sono le regioni di decisione riguardanti la segnalazione ricevuta in ingresso al ricevitore.

Come si può osservare dalla Figura 3.7a, a differenza della costellazione al trasmettitore, al ricevitore le regioni di decisione non subiscono la sola rotazione pari all'angolo di cui è stata ruotata la segnalazione QAM, perché il canale ha introdotto delle distorsioni.

Si osservi anche il risultato ottenuto considerando un'applicazione non corretta del criterio MD, Figura 3.7b, ciò che si ottiene sono delle regioni che non considerano la distanza minima tra tutti i punti. Le conseguenze dovute ad una tale suddivisione si ripercuotono nel calcolo del SER come sarà visto nel sottoparagrafo successivo 3.2.2.

In conclusione il metodo basato sull'hard decision potrebbe essere complicato da mettere in atto per ottenere la demappatura dei bit trasmessi al ricevitore, perché la regola di decodifica ottenuta non è semplice da ricavare e da applicare, basti guardare alla struttura delle regioni di decisione di Figura 3.7. Una tecnica più accorta viene presentata nel paragrafo 3.3, secondo un metodo che si presenta sotto il nome di Log Likelihood Ratio.

### 3.2.2 Calcolo symbol error rate

Ci occupiamo ora del calcolo del symbol error rate, ovvero della probabilità che un simbolo trasmesso venga decodificato in maniera errata dal ricevitore.

Si è visto nel sottoparagrafo precedente 3.2.1 com'è avvenuta la costruzione delle regioni di decisione a partire dalla costellazione in uscita dal mezzo trasmissivo, ottenendo i due grafici di Figura 3.7a e Figura 3.7b. A partire da queste due suddivisioni dello spazio calcoliamo ora il symbol error rate rispettivamente nei due casi.

Nella simulazione presentata in appendice A, per il calcolo del SER si ipotizza che il sistema riceva in ingresso un flusso binario random di  $n=10000$  bits, successivamente mediante la funzione *bi2de()*, messa a disposizione da

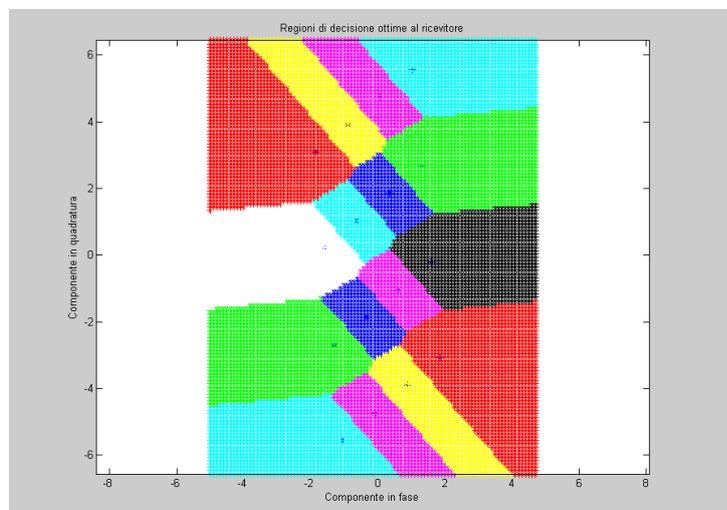
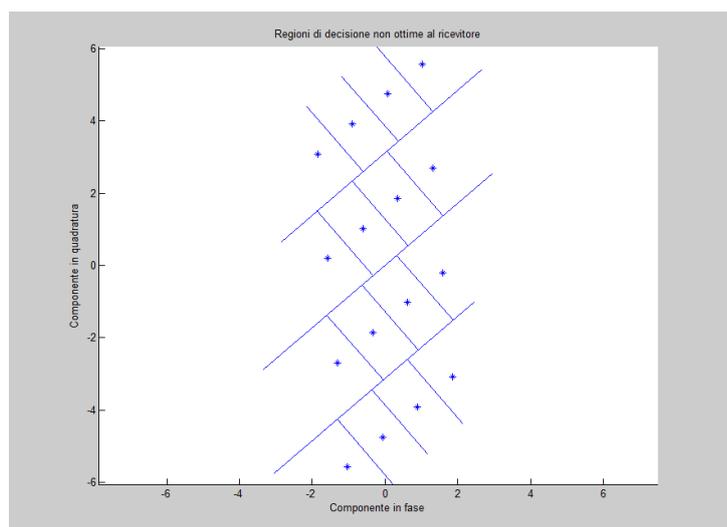
(a) *Regioni di decisione ottime costruite al ricevitore.*(b) *Regioni di decisione non ottime costruite al ricevitore.*

Figura 3.7: Regioni di decisione costruite al ricevitore.

Matlab, tale sequenza di bits viene decodificata in simboli. Si osservi che ogni simbolo è costituito da  $k = \log_2(M)$  bits, dove  $M$  corrisponde alla dimensione della costellazione.

I simboli vengono poi modulati mediante una modulazione  $M$ -QAM e, come richiesto dallo standard DVB-T2, vengono ruotati di un determinato angolo

$\Phi$  definito dall'utente in base alle proprie necessità, all'inizio del programma. A seguito di questa sequenza d'istruzioni ciò che si ottiene è l'informazione che deve essere trasmessa nel canale.

Come già espresso in precedenza il canale introduce un rumore, utilizzando la funzione messa a disposizione da Matlab *awgn()* ed impostando un canale avente un determinato SNR, i simboli trasmessi vengono affetti da un rumore gaussiano bianco. In questo modo, i punti che costituiscono l'informazione in ingresso al ricevitore risulteranno diversi dai punti trasmessi dalla sorgente.

Il ricevitore dovrà quindi decodificare tali punti in base alla regione di decisione che essi appartengono, regioni di decisione costruita a partire dai punti della costellazione in uscita dal canale, i quali hanno subito un'amplificazione/attenuazione sulla componente in fase I e sulla componente in quadratura Q. Ipotizzando ora che lo spazio in cui è contenuta la costellazione ricevuta sia suddiviso in maniera ottimale utilizzando il criterio MD, Figura 3.7a ed eseguendo un confronto tra i simboli trasmessi e quelli ricevuti, si esegue il conteggio dei simboli decodificati in modo errato.

Allo stesso modo anziché considerare la suddivisione dello spazio in regioni ottime, si considera la suddivisione in regioni non ottime, Figura 3.7b, e si esegue nuovamente il conteggio dei simboli decodificati in maniera errata.

In conclusione, il symbol error rate in entrambi i casi sarà dato dal rapporto tra il numero di simboli decodificati in maniera errata  $SE$ , e il numero totale di simboli trasmessi  $TS$ .

$$SER = \frac{SE}{TS} = \frac{SE}{\frac{n}{k}} \quad (3.29)$$

con  $n$  = numero di bits in ingresso al sistema e  $k$  = numero di bits per simbolo.

Mantenendo gli stessi simboli in ingresso al sistema, ma facendo variare il rapporto SNR del mezzo trasmissivo tra un valore compreso nell'insieme  $[0 \div 25]$ db, si ottiene per il SER l'andamento di Figura 3.8.

Si osservi dal grafico che a parità di SNR, la probabilità che il simbolo ricevuto sia decodificato in maniera errata è leggermente maggiore nel caso si

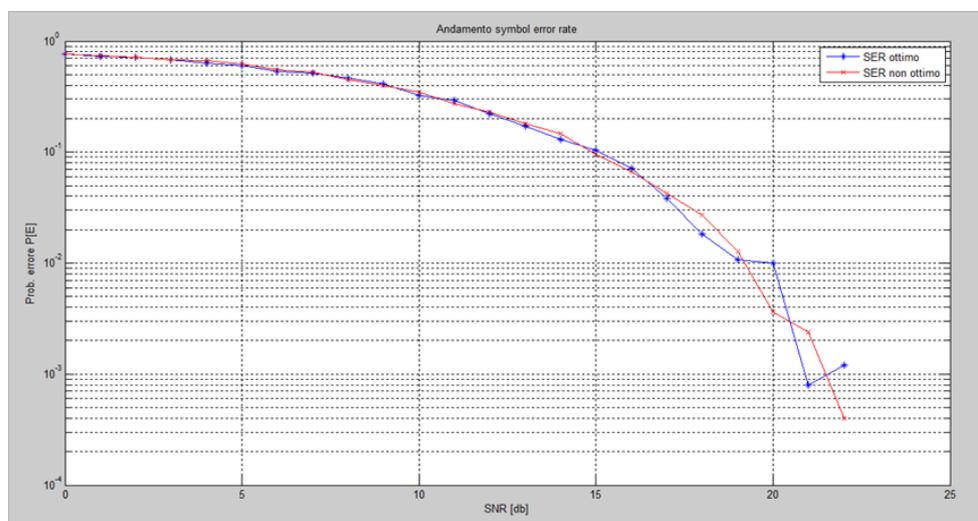


Figura 3.8: Andamento symbol error rate.

utilizzi una suddivisione dello spazio in regioni di decisione non ottime rispetto alla suddivisione in regioni di decisione ottime.

Lo scostamento tra i due andamenti in questo caso è molto piccolo in quanto le regioni di decisione hanno pressappoco la medesima struttura, nel caso cioè non si verificasse allora il SER sarebbe molto più elevato nel caso di suddivisione dello spazio in regioni non ottime e, di conseguenza, i due andamenti avrebbero scostamenti molto maggiori.

### 3.3 Soft Decision: LLR - Log Likelihood Ratio

Ipotizzando l'utilizzo di una modulazione  $M$ -QAM, gli  $M$  punti della costellazione sono rappresentati in un diagramma complesso (I,Q). Ogni punto della costellazione è rappresentato da  $k = \log_2 M$  bits binari e rappresenta il corrispondente simbolo trasmesso. La presenza del rumore nel canale fa sì che il segnale ricevuto possa essere distorto, cosicché può accadere che qualche bit al ricevitore venga interpretato come uno 0 anziché un 1 o viceversa. Il LLR è un criterio di decisione, conosciuto come soft decision, il

quale esegue una de-mappatura bit a bit, formando la costellazione a partire da quella ricevuta  $Q_k$ .

Considerando lo schema del sistema DVB-T2 presentato in precedenza, si ricava che il segnale ricevuto dal destinatario si presenta nella seguente forma:

$$Q_k = [|H_k|\Re(Z_k) + \Re(w_k)] + j[|H_{k+1}|\Im(Z_{k+1}) + \Im(w_{k+1})] \quad \text{con } k = 0, \dots, N_c - 1 \quad (3.30)$$

dove:

- $Q_k$  è il  $k$ -esimo simbolo ricevuto, avendo preso la parte reale ed immaginaria dal simbolo arrivato al ricevitore rispettivamente mediante la portante  $k$ -esima e la portante  $k+1$ -esima;
- $|H_k|, |H_{k+1}|$  corrispondono al modulo della risposta in frequenza del canale associato al  $k$ -esimo e  $k+1$ -esimo simbolo;
- $Z_k, Z_{k+1}$  corrispondono al  $k$ -esimo e  $k+1$ -esimo simbolo inviato dal trasmettitore, tal simbolo è quello effettivamente posto nel canale, quindi modulato e ruotato di un angolo  $\Phi$ ;
- $w_k, w_{k+1}$  corrispondono al rumore introdotto dal canale alla portante  $k$ -esima e  $k+1$ -esima. Il rumore è sintetizzato come una variabile aleatoria gaussiana a media nulla e varianza  $\sigma_w^2$ :  $w_k \sim \mathcal{G}(0, \sigma_w^2)$ .

Il log likelihood ratio del bit  $i$ -esimo è definito nel seguente modo:

$$LLR(b_i) = \ln \left( \frac{P(b_i = 1|Q_k)}{P(b_i = 0|Q_k)} \right) \quad i = 0, \dots, m - 1. \quad (3.31)$$

con  $Q_k$  dato dalla relazione (3.30). Dalla relazione (3.31) il risultato ottenuto può essere un valore negativo e in questo caso significa che il bit inviato  $b_i$  è stato trasmesso molto probabilmente come uno 0, quindi il bit ricevuto è posto pari a  $\hat{b}_i=0$ ; in caso contrario (LLR positivo) si assume il bit ricevuto pari a  $\hat{b}_i=1$ .

Indichiamo con  $C_i^j$  l'insieme dei punti della costellazione per i quali l' $i$ -esimo bit,  $b_i$ , assume il valore  $j$  (0 o 1). Tale insieme è così esprimibile:

$$C_i^1 = \{|H_k|\Re\{\exp(j\Phi)X_k\} + j|H_{k+1}|\Im\{\exp(j\Phi)X_k\} : [X_k, X_{k+1}] \in QAM \vee b_i = 1\} \quad (3.32)$$

la stessa definizione è adottata per definire  $C_i^0$ :

$$C_i^0 = \{|H_k|\Re\{\exp(j\Phi)X_k\} + j|H_{k+1}|\Im\{\exp(j\Phi)X_k\} : [X_k, X_{k+1}] \in QAM \vee b_i = 0\} \quad (3.33)$$

La definizione (3.31), applicando la relazione (3.32) e (3.33), diventa:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} P(Z_k = \alpha | Q_k)}{\sum_{\beta \in C_i^0} P(Z_k = \beta | Q_k)} \right) \quad i = 0, \dots, m-1. \quad (3.34)$$

Applicando la formula di Bayes si ottiene:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} P(Q_k | Z_k = \alpha) P[Z_k = \alpha]}{\sum_{\beta \in C_i^0} P(Q_k | Z_k = \beta) P[Z_k = \beta]} \right) \quad i = 0, \dots, m-1. \quad (3.35)$$

Assumendo che i simboli siano equiprobabili, dalla relazione (3.35) otteniamo:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} P(Q_k | Z_k = \alpha)}{\sum_{\beta \in C_i^0} P(Q_k | Z_k = \beta)} \right) \quad i = 0, \dots, m-1. \quad (3.36)$$

Dalla definizione data di  $Q_k$  la relazione di cui sopra può essere così riscritta:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} P(W_k = Q_k - \alpha_k)}{\sum_{\beta \in C_i^0} P(W_k = Q_k - \beta_k)} \right) \quad i = 0, \dots, m-1. \quad (3.37)$$

Recuperando l'ipotesi iniziale, in cui si era stabilito che il rumore è modellizzato come una variabile aleatoria gaussiana, e conoscendo la densità di probabilità di una tale variabile aleatoria, otteniamo il valore LLR richiesto per il bit  $i$ -esimo del punto della costellazione preso in considerazione:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} \exp\left(-\frac{(Q_k - \alpha)^2}{2\sigma_w^2}\right)}{\sum_{\beta \in C_i^0} \exp\left(-\frac{(Q_k - \beta)^2}{2\sigma_w^2}\right)} \right) \quad i = 0, \dots, m-1. \quad (3.38)$$

Si è arrivati a tale relazione finale per il calcolo del valore di LLR ipotizzando che i simboli trasmessi  $Z_k$  siano equiprobabili, ovvero che  $P[Z_k = 0] = P[Z_k = 1]$ . Nella pratica tale assunzione non è verificata, quindi la relazione (3.38) deve essere leggermente modificata.

Se consideriamo la relazione (3.35) sotto l'ipotesi che i simboli non siano equiprobabili, abbiamo che  $P[Z_k = \alpha]$  e  $P[Z_k = \beta]$ , presenti rispettivamente al numeratore e denominatore, non si semplificano. Detto ciò otteniamo che:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} P(W_k = Q_k - \alpha_k) q}{\sum_{\beta \in C_i^0} P(W_k = Q_k - \beta_k) (1 - q)} \right) \quad i = 0, \dots, m - 1. \quad (3.39)$$

dove si è posto  $q = P[Z_k = 1]$ .

Mantenendo sempre l'ipotesi che il rumore sia sintetizzabile come una variabile aleatoria Gaussiana, otteniamo la relazione finale per il calcolo del valore di LLR nel caso di simboli non equiprobabili:

$$LLR(b_i) = \ln \left( \frac{\sum_{\alpha \in C_i^1} \exp \left( -\frac{(Q_k - \alpha)^2}{2\sigma_w^2} \right) q}{\sum_{\beta \in C_i^0} \exp \left( -\frac{(Q_k - \beta)^2}{2\sigma_w^2} \right) (1 - q)} \right) \quad i = 0, \dots, m - 1. \quad (3.40)$$

### 3.4 Demappatura e decodifica iterativa

La demappatura e decodifica iterativa è una tecnica che consente di effettuare il calcolo del valore di LLR per il bit  $i$ -esimo  $\{b_i\}$  a partire dalla conoscenza a priori della probabilità di stato degli altri  $(m-1)$  bits. Tali valori delle probabilità di stato sono noti dalle precedenti iterazioni effettuate per il calcolo del valore di LLR.

Si parta dalla definizione già vista nel sottoparagrafo precedente di LLR, ovvero:

$$LLR(b_i) = \ln \left( \frac{P(b_i = 1|Q_k)}{P(b_i = 0|Q_k)} \right) \quad i = 0, \dots, m - 1. \quad (3.41)$$

Calcolando da essa la relazione inversa è possibile ricavare la probabilità che il bit  $i$ -esimo sia pari a 0 o viceversa pari ad 1:

$$P(b_i = 1) = \frac{\exp(LLR(b_i))}{1 + \exp(LLR(b_i))} \quad (3.42)$$

$$P(b_i = 0) = \frac{1}{1 + \exp(LLR(b_i))} \quad (3.43)$$

Inoltre la probabilità condizionata di ricevere il valore  $Q_k$ , sapendo che è stato trasmesso il bit  $i$ -esimo  $\{b_i\}$  come un 1(0) è data dalla seguente espressione:

$$P(Q_k|b_i = 1) = \frac{1}{2^m \pi \sigma^2} \sum_{\alpha \in C_i^1} \left( \exp\left(-\frac{Q_k - \alpha}{2\sigma^2}\right) P_{priori}(\alpha|b_i = 1) \right) \quad (3.44)$$

e viceversa:

$$P(Q_k|b_i = 0) = \frac{1}{2^m \pi \sigma^2} \sum_{\beta \in C_i^0} \left( \exp\left(-\frac{Q_k - \beta}{2\sigma^2}\right) P_{priori}(\beta|b_i = 0) \right) \quad (3.45)$$

Le relazioni (3.44) e (3.45) sommano i contributi di tutti i  $2^{m-1}$  possibili punti  $\alpha, \beta$  appartenenti rispettivamente all'insieme  $C_i^1$  e  $C_i^0$ . Tali due insiemi sono già stati definiti nel sottoparagrafo precedente dalle relazioni (3.32) e (3.33). Ogni punto ha una determinata probabilità di essere trasmesso, e tale probabilità può essere espressa come funzione delle probabilità che gli  $(m-1)$  bits oltre al bit  $\{b_i\}$  assumano il valore 0 o 1.

La probabilità che il punto ricevuto sia  $\alpha$ , sapendo che il bit è stato trasmesso come un 1, è data dal prodotto delle probabilità che gli  $(m-1)$  bits assumano il valore  $\alpha$  ovvero:

$$P_{priori}(\alpha|b_i = 1) = \prod_{k \neq i} P(b_k = \alpha_k) \quad (3.46)$$

e

$$P_{priori}(\beta|b_i = 0) = \prod_{k \neq i} P(b_k = \beta_k) \quad (3.47)$$

In conclusione, sostituendo le relazioni (3.42) e (3.43) rispettivamente con (3.46) e (3.47), e a sua volta riconsiderando la definizione di LLR, otteniamo

il seguente risultato finale:

$$LLR(b_i) = \ln \left\{ \frac{\sum_{\alpha \in C_i^1} \left( \exp\left(-\frac{(Q_k - \alpha)^2}{2\sigma^2}\right) \prod_{k \neq i} \frac{\exp(LLR(b_k))}{1 + \exp(LLR(b_k))} \right)}{\sum_{\beta \in C_i^0} \left( \exp\left(-\frac{(Q_k - \alpha)^2}{2\sigma^2}\right) \prod_{k \neq i} \frac{1}{1 + \exp(LLR(b_k))} \right)} \right\} \quad (3.48)$$

Tale espressione finale esprime il valore di LLR per il bit  $i$ -esimo all'uscita del demapper prendendo in considerazione un processo iterativo.

# Conclusioni

Lo standard DVB-T2 è la nuova piattaforma di diffusione terrestre candidata a sostituire l'attuale standard DVB-T. Molti paesi europei hanno già iniziato la conversione dallo standard DVB-T a quello di seconda generazione, DVB-T2, spinti anche dal fatto che quest'ultimo porta con sé molti vantaggi. Com'è stato analizzato nel capitolo 1, tra i fattori che hanno incentivato al passaggio a questa nuova piattaforma, c'è senza dubbio la diffusione della televisione ad alta definizione.

Un altro fattore molto importante è il seguente: a parità di banda con il nuovo standard DVB-T2 si hanno a disposizione più canali TV\servizi aggiungendo anche una qualità ben più elevata rispetto a quella offerta dal DVB-T. Infatti, tra gli obiettivi primari da raggiungere all'inizio della progettazione della nuova piattaforma, i progettisti avevano imposto un aumento dell'efficienza spettrale del 30%, obiettivo ampiamente raggiunto, arrivando in alcuni contesti anche ad un aumento del 50%.

I vantaggi sopra esposti ed elencati in modo molto più dettagliato nella tabella 1.1 presente nel capitolo 1, sono stati raggiunti a fronte di un compromesso: un aumento della complessità del sistema rispetto alle vecchie piattaforme DVB.

Nel capitolo 2 e nel capitolo 3 sono state analizzate due tecniche che hanno dato la svolta per la nascita dello standard DVB-T2, ovvero la modulazione *OFDM* e la tecnica della *costellazione ruotata*. Mediante l'utilizzo della modulazione *OFDM* si è stati in grado di costruire un sistema nel quale sorgente e destinatario siano in grado di comunicare anche in condizioni piut-

tosto critiche del canale.

La seconda tecnica, ovvero la rotazione di un determinato angolo dei punti di una costellazione  $M$ -QAM, accompagnata dall'introduzione di un ritardo ciclico sull'ordinata  $Q$  prima dell'interleaving in tempo e frequenza, consente di trasmettere la componente in fase e in quadratura su portanti  $OFDM$  diverse. Questo comporta che la probabilità che entrambe le componenti siano compromesse dal canale è molto bassa, quindi come conseguenza a questo fatto il ricevitore è in grado di decodificare in modo univoco il simbolo trasmesso a partire dalla conoscenza di una sola coordinata.

Il vantaggio introdotto è un incremento del guadagno nel caso di canali critici, e questo si traduce in una maggiore capacità trasmissiva del canale.

In conclusione lo standard DVB-T2 si prospetta essere un'ottima piattaforma la quale richiede però uno sviluppo completamente nuovo di tutta l'elettronica per la codifica di sorgente e di canale. Questo potrebbe causare alcuni problemi per l'affermarsi della nuova tecnologia poiché sarà necessaria la sostituzione degli IRD domestici, ma una soluzione è data dal fatto che questa nuova tecnologia rimanga in vigore per molto tempo prima di essere sostituita da qualche altro nuovo standard.

Un ulteriore fattore che comporterà l'affermarsi del nuovo standard è la sua flessibilità, ovvero è in grado di lavorare con un'elevata varietà di dati in ingresso rendendo così più robusta la trasmissione di servizi\programmi di particolare interesse o destinati ad un'utenza in condizioni critiche di ricezione. Tutti questi fattori sopra elencati potranno far in modo che entro qualche anno questa nuova tecnologia chiamata DVB-T2 vada a sostituire il già presente standard DVB-T.

# Appendice A

## Simulazione Matlab

In quest'appendice è ripreso l'argomento trattato nel sottoparagrafo 3.2, all'interno del quale si è cercato di costruire le regioni di decisione di una costellazione QAM che sia di ordine  $M = 4 \setminus 16 \setminus 64 \setminus 256$ .

Tale programma, scritto in codice Matlab, consente la simulazione delle operazioni svolte dal trasmettitore e dal ricevitore che implementano il digitale terrestre di seconda generazione, affinché siano inviate le informazioni tra i due estremi (sorgente e destinatario).

Il programma è costituito da una funzione *main.m* che costituisce il cuore del progetto, da questa sono richiamate altre funzioni le quali eseguono operazioni specifiche che saranno di seguito esaminate.

La funzione *main.m* consente di specificare una serie di parametri riguardanti la costellazione, quali possono essere la dimensione della costellazione  $M$ , l'angolo di rotazione della costellazione  $\phi$ ; successivamente consente di inizializzare parametri riguardanti il mezzo trasmissivo come il rapporto segnale/rumore  $SNR$ , il fattore di amplificazione/attenuazione della componente in fase e quadratura, rispettivamente *fatt\_stretching\_x* e *fatt\_stretching\_y*.

Un'ultima parte riguarda l'inizializzazione delle variabili utili per il calcolo del symbol error rate, ovvero il numero di bit che sono posti in ingresso al sistema  $n$ , la frequenza di campionamento del segnale d'ingresso e d'uscita

identificate rispettivamente dalle variabili  $Fd$  e  $Fs$  e, in conclusione, la frequenza (*frequency*) con cui lavora il sistema.

Tutti questi parametri sopra specificati devono essere impostati dall'utente in base al sistema che vuole simulare.

A seguito di questa prima parte di inizializzazione delle variabili viene costruita la costellazione, ovvero vengono inseriti all'interno di una matrice tutti i punti, espressi in forma complessa, che costituiscono la costellazione. Successivamente viene richiamata la funzione *regioni\_decisione\_non\_ruotata.m* che, come output, crea un grafico dove vengono visualizzati i punti della costellazione  $M$ -QAM e le relative regioni di decisione.

All'interno del sottoparagrafo 3.2 è stato introdotto il concetto di costellazione ruotata analizzando i vantaggi e svantaggi che tale tecnica porta con sé. Il codice che segue la chiamata alla funzione *regioni\_decisione\_non\_ruotata.m*, a partire dalla matrice contenente i punti della costellazione iniziale, esegue una rotazione della costellazione di un angolo  $\Phi$ . Anche tali punti vengono poi inseriti in una matrice.

A seguito della rotazione viene richiamata la funzione *regioni\_decisione\_ruotata.m* che visualizza in un grafico la costellazione ruotata e le relative regioni di decisione. Come si potrà osservare quest'ultimo grafico è la versione ruotata di un angolo  $\Phi$  dell'output ottenuto richiamando la funzione *regioni\_decisione\_non\_ruotata.m*.

La costellazione ruotata viene immessa poi nel mezzo trasmissivo, il quale introduce rumore ed una certa amplificazione\attenuazione sui punti. Le righe di codice che seguono alla rotazione simulano gli effetti del canale, ovvero ogni punto della costellazione subisce un'amplificazione\attenuazione sulla componente in fase I e un'amplificazione\attenuazione sulla componente in quadratura Q.

La costellazione in entrata al ricevitore risulta quindi essere diversa da quella trasmessa e, per decodificare i simboli ricevuti, il ricevitore a partire dai punti ricevuti dovrà creare le regioni di decisione e ad ognuna di queste associare un simbolo d'uscita.

La costruzione delle regioni di decisione al ricevitore è eseguita richiamando la funzione *regioni\_decisione\_stretching\_ottime.m*. Tale procedura, applicando il criterio MD, visualizza in un grafico la suddivisione dello spazio in cui è contenuta la costellazione in modo ottimale, successivamente viene anche richiamata una funzione, *regioni\_decisione\_stretching.m*, che esegue una suddivisione dello spazio in modo non ottimale.

La seconda parte della simulazione consiste nel calcolo del symbol error rate; tale parametro è valutato considerando le due situazioni presentate sopra, ovvero che la costellazione ricevuta sia suddivisa in regioni di decisione ottime, e il caso opposto, nel caso in cui lo spazio sia suddiviso in regioni di decisione non ottime.

Per il calcolo del SER viene richiamata la funzione *Creazione\_flusso\_tx.m* che crea un flusso binario random di lunghezza  $n$  bits, questi vengono poi convertiti in simboli e successivamente modulati utilizzando appunto una modulazione  $M$ -QAM. Come espresso in precedenza, i simboli prima di essere immessi nel mezzo trasmissivo devono essere ruotati, quindi essi subiscono una rotazione di un angolo  $\Phi$ . La funzione restituisce un vettore il quale contiene i simboli generati dal flusso binario e i simboli modulati espressi in forma complessa.

Dopo aver creato il flusso da trasmettere, vengono richiamate le due funzioni *Calcolo\_SER\_non\_ottimo.m* e *Calcolo\_SER\_ottimo.m*, che rispettivamente mandano come output un vettore colonna contenente il symbol error rate al variare del rapporto SNR introdotto dal mezzo trasmissivo, nel caso in cui le regioni di decisione siano non ottime e ottime. Tali due funzioni al loro interno richiamano un'ulteriore funzione: *Creazione\_flusso\_rx.m*, essa simula la presenza del mezzo trasmissivo, ovvero aggiunge un rumore gaussiano bianco e un'amplificazione/attenuazione ai simboli che vengono trasmessi dal trasmettitore e quindi i simboli che sono stati creati dalla funzione *Creazione\_flusso\_tx.m*.

Per concludere, le ultime righe di codice della funzione *main.m* visualizzano l'andamento, in un unico grafico, del symbol error rate nei due casi sopra

considerati, in modo tale da eseguire un confronto.

Di seguito vengono presentate tutte le funzioni sopra citate.

### **main.m**

```

1 function Main
2 %PARAMETRI RIGUARDANTI LA COSTELLAZIONE M-QAM INIZIALI
3 %MODIFICABILI IN BASE ALLA RICHIESTA DELL'UTENTE.
4 M = 16; %-->dimensione della costellazione;
5 phi = 16; % -->angolo di rotazione della costellazione;
6
7 %PARAMETRI RIGUARDANTI LA PARTE RELATIVA AL CALCOLO
8 %DEL SYMBOL ERROR RATE.
9 k = log2(M); %-->numero di bit per simbolo;
10 n = 10000; %-->numero di bit da considerare in
11 %ingresso al sistema;
12 Fd = 1; %-->frequenza di campionamento del
13 %messaggio d'ingresso;
14 Fs = 1; %-->frequenza di campionamento del
15 %messaggio d'uscita;
16 frequency = 100; %-->frequenza del sistema;
17 nsamp = 1; %-->rate di sovracampionamento;
18
19 %PARAMETRI RIGUARDANTI IL CANALE TRASMISSIVO
20 start = 22;
21 SNR = start; %-->rapporto segnale/rumore
22 %espresso in decibel;
23 fatt_stretching_x = 0.5; %-->fattore di amplificazione
24 %componente in fase;
25 fatt_stretching_y = 1.5; %-->fattore di amplificazione
26 %componente in quadratura;
27
28 %COSTRUZIONE E RAPPRESENTAZIONE DELLA COSTELLAZIONE
29 %INIZIALE. TALE COSTELLAZIONE CORRISPONDE ALLA
30 %RAPPRESENTAZIONE VETTORIALE DELLA SEGNALAZIONE
31 %IN INGRESSO AL TRASMETTITORE.
32 row = (sqrt(M));

```

```

33 column = (sqrt(M));
34 x = -row+1;
35 y = -column+1;
36 for (a=1 : row)
37     for (b=1 : column)
38         Cost(a,b) = x + j*y;
39         x = x+2;
40     end
41     y = y+2;
42     x = -row+1;
43 end
44 figure(1);
45 plot(real(Cost),imag(Cost), '*r');
46 title('Costellazione al trasmettitore');
47 xlabel('Componente in fase');
48 ylabel('Componente in quadratura');
49 axis('square');
50 grid on
51
52 %RICHIAMA LA FUNZIONE regioni_decisione_non_ruotata()
53 %PER COSTRUIRE LE REGIONI DI DECISIONE DI UNA
54 %COSTELLAZIONE M-QAM.
55 regioni_decisione_non_ruotata(Cost);
56
57 %ROTAZIONE DELLA COSTELLAZIONE INIZIALE M-QAM DI UN
58 %ANGOLO phi, E RAPPRESENTAZIONE GRAFICA.
59 Cost_rot = Cost*(cos((phi*2*pi)/360)+j*sin((phi*2*pi)/360));
60 figure(3)
61 plot(real(Cost_rot),imag(Cost_rot), '*R');
62 title('Costellazione ruotata di un angolo PHI');
63 xlabel('Componente in fase');
64 ylabel('Componente in quadratura');
65 axis('square');
66 grid on
67
68 %RICHIAMA LA FUNZIONE regioni_decisione_ruotata() CHE
69 %CREA E VISUALIZZA LE REGIONI DI DECISIONE DI UNA
70 %COSTELLAZIONE RUOTATA DI UN ANGOLO phi.

```

```

71 regioni_decisione_ruotata(M, Cost_rot , phi);
72
73 %COSTELLAZIONE M-QAM IN INGRESSO AL RICEVITORE. OGNI PUNTO
74 %DELLA COSTELLAZIONE VIENE AMPLIFICATO/ATTENUATO DI UN
75 %DETERMINATO VALORE.
76 Cost_stretching_real = real(Cost_rot)*fatt_stretching_x;
77 Cost_stretching_imag = imag(Cost_rot)*fatt_stretching_y;
78 Cost_stretching = Cost_stretching_real+j*Cost_stretching_imag;
79 figure(5);
80 plot((Cost_stretching_real), Cost_stretching_imag, '*r');
81 title('Costellazione QAM al ricevitore');
82 xlabel('Componente in fase');
83 ylabel('Componente in quadratura');
84 axis('equal');
85 grid
86
87 %RICHIAMA LA FUNZIONE regioni_decisione_stretching()
88 %CHE CREA E VISUALIZZA LE REGIONI DI DECISIONE NON
89 %OTTIME A PARTIRE DAI PUNTI DELLA COSTELLAZIONE
90 %USCENTI DAL CANALE DI TRASMISSIONE.
91 regioni_decisione_stretching(M, Cost_stretching);
92
93 %RICHIAMA LA FUNZIONE regioni_decisione_stretching_ottime()
94 %CHE CREA E VISUALIZZA LE REGIONI DI DECISIONE OTTIME A
95 %PARTIRE DAI PUNTI DELLA COSTELLAZIONE USCENTI DAL
96 %CANALE DI TRASMISSIONE.
97 regioni_decisione_stretching_ottime(Cost_stretching);
98
99 %PARTE DI PROGRAMMA RIGUARDANTE IL CALCOLO DEL SYMBOL
100 %ERROR RATE.
101
102 %ridefinizione della matrice contenente i punti della
103 %costellazione iniziale ruotata e comprensiva di stretching.
104 r = 4;
105 c = 1;
106 a = 1;
107 b = 1;
108 while(r>0)

```

```

109 while(c<= length(Cost_stretching))
110     vet_prova(a,b) = Cost_stretching(r,c);
111     b = b+1;
112     c = c+1;
113 end
114 b = 1;
115 c = 1;
116 a = a+1;
117 r = r-1;
118 end
119 Cost_stretching = vet_prova;
120
121 %sistemazione matrice in modo tale che le regioni di
122 %decisione siano uguali a quelle utilizzate dal
123 %comando 'dmodece()' .
124 c = Cost_stretching;
125 Cost_ottima = [c(2,3) c(2,2) c(3,3) c(3,2);...
126               c(2,4) c(2,1) c(3,4) c(3,1);...
127               c(1,3) c(1,2) c(4,3) c(4,2);...
128               c(1,4) c(1,1) c(4,4) c(4,1)];
129 %Inserimento della matrice 'Cost_ottima' all'interno
130 %di un vettore colonna [Mx1].
131 vet_prova = zeros(M,1);
132 q = 1;
133 for(r=1 : length(Cost_ottima))
134     for(c=1 : length(Cost_ottima))
135         vet_prova(q,1) = Cost_ottima(r,c);
136         q = q+1;
137     end
138 end
139 Cost_ottima = vet_prova;
140
141 %RICHIAMA LA FUNZIONE Creazione_flusso LA QUALE CREA UNA
142 %SEQUENZA DI SIMBOLI DA TRASMETTERE E MANDA IN USCITA
143 %I SIMBOLI RICEVUTI DAL MITTENTE.
144 vet = Creazione_flusso_tx(M,k,n,Fd,Fs,phi);
145 %Inserimento all'interno di due vettori colonna [kx1],
146 %symbol_tx e symbol_tx_complex, rispettivamente dei

```

```

147 %simboli che vengono posti nel canale in forma numerica
148 %ecomplessa.
149 v = 1;
150 while (v <= (n/k))
151     symbol_tx(v,1) = vet(v,1);
152     symbol_tx_complex(v,1) = vet(v,2);
153     v = v+1;
154 end
155 symbol_tx;
156 symbol_tx_complex;
157
158 %RICHIAMA LA FUNZIONE Calcolo_SER_non_ottimo CHE
159 %CALCOLA IL SYMBOL ERROR RATE CONSIDERANDO LA
160 %SUDDIVISIONE DELLO SPAZIO, IN CUI E' CONTENUTA LA
161 %COSTELLAZIONE, IN REGIONI DI DECISIONE NON OTTIME
162 SER_non_ottimo = Calcolo_SER_non_ottimo(M,symbol_tx,...
163 Cost_stretching,start,fatt_stretching_x,...
164 fatt_stretching_y, Cost_ottima,symbol_tx_complex);
165
166 %RICHIAMA LA FUNZIONE Calcolo_SER_ottimo CHE
167 %CALCOLA IL SYMBOL ERROR RATE CONSIDERANDO LA
168 %SUDDIVISIONE DELLO SPAZIO, IN CUI E' CONTENUTA LA
169 %COSTELLAZIONE, IN REGIONI DI DECISIONE NON OTTIME
170 SER_ottimo = Calcolo_SER_ottimo(Cost_stretching,...
171     Cost_ottima,symbol_tx_complex,symbol_tx,start...
172     ,SNR,fatt_stretching_x,fatt_stretching_y);
173
174 %RAPPRESENTAZIONE GRAFICA DELL'ANDAMENTO DEL SYMBOL
175 %ERROR RATE AL VARIARE DEL RAPPORTO SNR DEL CANALE
176 %IN ENTRAMBE LE CONDIZIONI : SUDDIVISIONE DELLO SPAZIO
177 %IN REGIONI OTTIME E NON OTTIME.
178 figure (8);
179 t = length(SER_ottimo);
180 s = 1;
181 while (t > 0)
182     W(s,1) = SER_non_ottimo(t,1);
183     R(s,1) = SER_ottimo(t,1);
184     t = t-1;

```

```

185 SNR(s) = s-1;
186 s = s+1;
187 end
188 semilogy(SNR,W,'b*-');
189 hold on;
190 semilogy(SNR,R,'rx-');
191 hold on;
192 title('Andamento symbol error rate');
193 xlabel('SNR [db]');
194 ylabel('Prob. errore P[E]');
195 legend('SER ottimo','SER non ottimo');
196 end %FINE FUNCTION

```

### regioni\_decisione\_non\_ruotata.m

```

1 %FUNZIONE CHE COSTRUISCE LE REGIONI DI DECISIONE
2 %DI UNA COSTELLAZIONE M-QAM NON RUOTATA.
3 %VARIABILI D'INGRESSO:
4 % Cost = matrice contenente i punti della costellazione;
5 %
6 %
7 function regioni_decisione_non_ruotata(Cost)
8 row = length(Cost);
9 column = length(Cost);
10 x = -row+2;
11 y = -column+2;
12 figure(2);
13 plot(real(Cost),imag(Cost), '*r');
14 axis('square');
15     for(s=1 : column-1)
16         hold on;
17         vetx_or = [0+x 0+x];
18         vety_or = [column -column];
19         vetx_ve = [-row row];
20         vety_ve = [0+y 0+y];
21         title('Regioni decisione costellazione QAM');
22         xlabel('Componente in fase');
23         ylabel('Componente in quadratura');

```

```

24         plot(vetx_or, vety_or, '-b');
25         hold on;
26         plot(vetx_ve, vety_ve, '-b');
27         x = x+2;
28         y = y+2;
29     end
30 end %FINE FUNCTION

```

### regioni\_decisione\_ruotata.m

```

1 %FUNZIONE CHE CREA LE REGIONI DI DECISIONE NEL CASO
2 %IN CUI LA COSTELLAZIONE M-QAM SIA RUOTATA DI UN
3 %ANGOLO phi.
4 %VARIABILI D'INGRESSO:
5 %M = dimensione della costellazione;
6 %Cost_rot = matrice contenente i punti della costellazione
7 %           M-QAM ruotati di un angolo 'phi';
8 %phi = angolo di rotazione della costellazione;
9 function regioni_decisione_ruotata(M, Cost_rot, phi)
10 row = length(Cost_rot);
11 column = length(Cost_rot);
12 figure(4);
13 hold on;
14 plot(real(Cost_rot), imag(Cost_rot), '*r');
15 title('Regioni di decisione costellazione ruotata');
16 xlabel('Componente in fase');
17 ylabel('Componente in quadratura');
18 axis('square');
19 x = -row+2;
20 for (s=1 : column-1)
21     if (M == 256)
22         a = x+j*16;
23         b = x-j*16;
24     end
25     if (M == 64)
26         a = x+j*8;
27         b = x-j*8;
28     end

```

```

29     if(M == 16)
30         a = x+j*4;
31         b = x-j*4;
32     end
33     a_rot = a*(cos(phi*2*pi/360)+j*sin(phi*2*pi/360));
34     b_rot = b*(cos(phi*2*pi/360)+j*sin(phi*2*pi/360));
35     vetx_rot = [real(a_rot) real(b_rot)];
36     vety_rot = [imag(a_rot) imag(b_rot)];
37     plot(vetx_rot , vety_rot , '-b');
38     hold on;
39     x = x+2;
40 end
41 y = -column+2;
42 for(s=1 : column-1)
43     if(M == 256)
44         a = 16+j*y;
45         b = -16+j*y;
46     end
47     if(M == 64)
48         a = 8+j*y;
49         b = -8+j*y;
50     end
51
52     if(M == 16)
53         a = 4+j*y;
54         b = -4+j*y;
55     end
56     a_rot = a*(cos(phi*2*pi/360)+j*sin(phi*2*pi/360));
57     b_rot = b*(cos(phi*2*pi/360)+j*sin(phi*2*pi/360));
58     vetx_rot = [real(a_rot) real(b_rot)];
59     vety_rot = [imag(a_rot) imag(b_rot)];
60     plot(vetx_rot , vety_rot , '-b');
61     hold on;
62     axis( 'square' );
63     y = y+2;
64 end
65 end %FINE FUNCTION.

```

**regioni\_decisione\_stretching.m**

```

1 %FUNZIONE CHE CREA LE REGIONI DI DECISIONE NON OTTIME
2 %DELLA COSTELLAZIONE M-QAM RICEVUTA AL RICEVITORE,
3 %QUINDI SOGGETTA A DISTURBI E AMPLIFICAZIONI/ATTENUAZIONI
4 %INTRODOTTE DAL CANALE.
5 %VARIABILI D'INGRESSO:
6 %Cost_stretching = matrice contenente i punti della
7 %costellazione in ingresso al ricevitore;
8 %M = dimensione della costellazione
9 %
10 %
11 function regioni_decisione_stretching(M, Cost_stretching)
12 figure(6)
13 hold on;
14 plot(real(Cost_stretching),imag(Cost_stretching),'*r');
15 title('Regioni di decisione non ottime al ricevitore');
16 xlabel('Componente in fase');
17 ylabel('Componente in quadratura');
18 axis('equal');
19 a = 1;
20 b = 1;
21 s = 1;
22 for(r=1 : length(Cost_stretching))
23   for(c=1 : length(Cost_stretching))
24     if(c == 1)
25       point_cost(s,1) = Cost_stretching(r,c);
26       s = s+1;
27     else
28       %Calcolo lunghezza dei cateti e ipotenusa
29       %del triangolo rettangolo.
30       c_min = imag(Cost_stretching(a,b)) - ...
31               imag(Cost_stretching(r,c));
32       c_max = real(Cost_stretching(a,b)) - ...
33               real(Cost_stretching(r,c));
34       ipotenusa = sqrt(c_min^2+c_max^2);
35       %Calcolo angolo di rotazione.
36       beta = acos(abs(c_max)/ipotenusa);

```

```

37  end
38  end
39  a = a+1;
40  end
41  %Inserimento della nuova costellazione all'interno
42  %della matrice 'matr_new'.
43  for (r=1 : length(Cost_stretching))
44  for (c=1 : length(Cost_stretching))
45  matr_new(r,c) = Cost_stretching(r,c)*(cos(-beta)...
46  +j*sin(-beta));
47  end
48  end
49  %Calcolo la distanza tra due punti appartenenti
50  %a righe diverse
51  d = abs(imag(matr_new(1,1))-imag(matr_new(2,1)));
52  %Calcolo del punto medio tra due punti adiacenti e
53  %appartenenti alla medesima riga.
54  for (r=1 : length(matr_new))
55  for (c=1 : length(matr_new)-1)
56  p_medi(r,c) = punti_medi(matr_new(r,c),...
57  matr_new(r,c+1));
58  end
59  end
60  %Calcolo punti estremi del segmento che definisce
61  %il confine della regione di decisione.
62  row = length(p_medi);
63  column = row-1;
64  for (r=1: row)
65  for (c=1 : column)
66  p_prev(r,c) = p_medi(r,c)+j*(d/2);
67  p_next(r,c) = p_medi(r,c)-j*(d/2);
68  end
69  end
70  for (r=1: row)
71  for (c=1 : column)
72  p_prev_rot(r,c) = p_prev(r,c)*(cos(beta)+j*sin(beta));
73  p_next_rot(r,c) = p_next(r,c)*(cos(beta)+j*sin(beta));
74  end

```

```

75 end
76 hold on;
77 for (r=1: row)
78   for (c=1 : column)
79     vetx = [real(p_prev_rot(r,c)) real(p_next_rot(r,c))];
80     vety = [imag(p_prev_rot(r,c)) imag(p_next_rot(r,c))];
81     hold on;
82     plot(vetx,vety,'-b');
83   end
84 end
85 r = 1;
86 while (r < length(matr_new))
87   p1 = matr_new(r,1);
88   if (real(p1)<0)
89     p1 = p1-4;
90   else
91     p1 = p1+1;
92   end
93   p2 = matr_new(r+1,1);
94   if (real(p2)<0)
95     p2 = p2-1;
96   else
97     p2 = p2+1;
98   end
99   p3 = matr_new(r,length(matr_new));
100  if (real(p3)<0)
101    p3 = p3-1;
102  else
103    p3 = p3+1;
104  end
105  p4 = matr_new(r+1,length(matr_new));
106  if (real(p4)<0)
107    p4 = p4-1;
108  else
109    p4 = p4+3;
110  end
111  p12 = punti_medi(p1,p2)*(cos(beta)+j*sin(beta));
112  p34 = punti_medi(p3,p4)*(cos(beta)+j*sin(beta));

```

```

113 hold on
114 vetx = [real(p12) real(p34)];
115 vety = [imag(p12) imag(p34)];
116 plot(vetx, vety, '-b');
117 hold on;
118 r = r+1;
119 end
120 end %FINE FUNCTION

```

### regioni\_decisione\_stretching\_ottime.m

```

1 %FUNZIONE CHE CREA LE REGIONI DI DECISIONE OTTIME DELLA
2 %COSTELLAZIONE M-QAM RICEVUTA AL RICEVITORE, QUINDI
3 %SOGGETTA A DISTURBI E AMPLIFICAZIONI/ATTENUAZIONI
4 %INTRODOTTE DAL CANALE.
5 %VARIABILI D'INGRESSO:
6 %Cost_stretching = matrice contenente i punti della
7 %costellazione in ingresso al ricevitore;
8 function regioni_decisione_stretching_ottime(Cost_stretching)
9 figure(7);
10 plot((real(Cost_stretching)), imag(Cost_stretching), '*r');
11 title('Regioni di decisione ottime al ricevitore');
12 xlabel('Componente in fase');
13 ylabel('Componente in quadratura');
14 axis('equal');
15 hold on;
16 i = 1;
17 for(m=1 : length(Cost_stretching))
18 for(n=1 : length(Cost_stretching))
19 x_cost(i) = real(Cost_stretching(m,n));
20 y_cost(i) = imag(Cost_stretching(m,n));
21 i = i+1;
22 end
23 end
24 fase = -abs(real(Cost_stretching(1,1)))-1;
25 quadratura = -abs(imag(Cost_stretching(1,1)))-1;
26 h = abs(quadratura)*2/0.1;
27 w = abs(fase)*2/0.1;

```

```

28 for (k=1 : h)
29     fase = -5;
30     for (q=1 : (3/4)*h)
31         punto = fase+j*quadratura;
32         for (m=1 : length(x_cost))
33             d(m) = sqrt((x_cost(m) - real(punto))^2 ...
34                 + (y_cost(m) - imag(punto))^2);
35         end
36         minimo = min(d);
37         length_vet = length(x_cost);
38         pos = search(minimo, length_vet, d);
39         fase = fase + 0.1;
40
41     switch(pos)
42         case 1
43             plot(real(punto), imag(punto), '*c');
44             hold on;
45         case 2
46             plot(real(punto), imag(punto), '*m');
47             hold on;
48         case 3
49             plot(real(punto), imag(punto), '*y');
50             hold on;
51         case 4
52             plot(real(punto), imag(punto), '*r');
53             hold on;
54         case 5
55             plot(real(punto), imag(punto), '*g');
56             hold on;
57         case 6
58             plot(real(punto), imag(punto), '*b');
59             hold on;
60         case 7
61             plot(real(punto), imag(punto), '*m');
62             hold on;
63         case 8
64             plot(real(punto), imag(punto), '*k');
65             hold on;

```

```

66     case 9
67         plot ( real (punto) , imag (punto) , '*w' );
68         hold on;
69     case 10
70         plot ( real (punto) , imag (punto) , '*c' );
71         hold on;
72     case 11
73         plot ( real (punto) , imag (punto) , '*b' );
74         hold on;
75     case 12
76         plot ( real (punto) , imag (punto) , '*g' );
77         hold on;
78     case 13
79         plot ( real (punto) , imag (punto) , '*r' );
80         hold on;
81     case 14
82         plot ( real (punto) , imag (punto) , '*y' );
83         hold on;
84     case 15
85         plot ( real (punto) , imag (punto) , '*m' );
86         hold on;
87     case 16
88         plot ( real (punto) , imag (punto) , '*c' );
89         hold on;
90     otherwise
91         error ('Impossibile');
92 end
93 end
94 quadratura = quadratura + 0.1;
95 end
96 end %FINE FUNCTION

```

### Creazione flusso\_tx.m

```

1 %FUNZIONE CHE ESEGUE UNA SIMULAZIONE DI TRASMISSIONE
2 %DI BIT IN UN SISTEMA DVB-T2
3 %VARIABILI D'INGRESSO:
4 %M = dimensione della costellazione

```

```

5 %k = numero di bit per simbolo
6 %n = numero di bit da considerare in ingresso al sistema
7 %Fd = frequenza di campionamento del messaggio d'ingresso
8 %Fs = frequenza di campionamento del messaggio d'uscita
9 %VARIABILI D'USCITA:
10 %vet = vettore contenente i simboli trasmessi e i simboli
11 %ricevuti dal mittente, quest'ultimi in forma complessa
12 function [vet] = Creazione_flusso_tx(M,k,n,Fd,Fs,phi)
13 %creazione del flusso binario in ingresso al sistema.
14 %Il flusso è costituito da 'n' bit scelti in modo
15 %casuale tra 0 e 1.
16 binary_data_stream = randint(n,1);
17 %conversione del flusso binario in simboli.
18 %Ogni simbolo è costituito da 'k' bit.
19 symbol_tx = bi2de(reshape(binary_data_stream,k,...
20     length(binary_data_stream)/k).', 'left-msb');
21 symbol_tx;
22 %segnale posto in ingresso al canale trasmissivo
23 %utilizzando una modulazione M-QAM.
24 ytx = dmodce(symbol_tx,Fd,Fs, 'qask',M);
25 %rotazione dei simboli modulati prima che vengano
26 %posti nel canale.
27 ytx_rot = ytx*(cos((phi*2*pi)/360)+j*(sin((phi*2*pi)/360)));
28 vet = [symbol_tx ytx_rot];
29 end %FINE FUNCTION

```

### Creazione\_flusso\_rx.m

```

1 %FUNZIONE CHE CREA I SIMBOLI IN USCITA DAL CANALE. AD ESSI
2 %VIENE AGGIUNTO UN RUMORE GAUSSIANO BIANCO ED
3 %UN'AMPLIFICAZIONE/ATTENUAZIONE SULLE COMPONENTI IN FASE E
4 %QUADRATURA.
5 %VARIABILI D'INGRESSO:
6 %%symbol_tx_complex = simboli trasmessi nel canale in
7 %forma complessa;
8 %SNR = rapporto segnale rumore introdotto dal canale
9 %trasmissivo espresso in db.
10 %fatt_stretching_x = fattore di amplificazione

```

```

11             %componente in fase;
12 %fatt_stretching_y = fattore di amplificazione
13             %componente in quadratura;
14 %VARIABILI D'USCITA:
15 %vet = vettore colonna contenente i simboli in ingresso
16 %al ricevitore espressi in forma complessa.
17 function [vet] = Creazione_flusso_rx(symbol_tx_complex ,...
18         start ,SNR, fatt_stretching_x ,fatt_stretching_y)
19 %Il canale introduce un rumore gaussiano bianco distorcendo
20 %cosi il segnale trasmesso.
21 z = 1;
22 yrx_channel = awgn(symbol_tx_complex ,SNR, 'measured');
23 yrx_channel_stretching = real(yrx_channel)*fatt_stretching_x...
24         +j*(imag(yrx_channel)*fatt_stretching_y);
25 vet = [yrx_channel_stretching];
26 end %FINE FUNCTION

```

### Calcolo\_SER\_non\_ottimo.m

```

1 %FUNZIONE CHE CALCOLA IL SYMBOL ERROR RATE CONSIDERANDO
2 %DELLE REGIONI CHE SUDDIVIDONO LO SPAZIO CONTENENTE LA
3 %COSTELLAZIONE RICEVUTA, IN MODO NON OTTIMALE.
4 %VARIABILI D'INGRESSO:
5 %M = dimensione della costellazione;
6 %symbol_tx = simboli trasmessi nel canale;
7 %Cost_stretching = matrice contenente i punti...
8 %della costellazione iniziale ruotati e strecciati;
9 %fatt_stretching_x = fattore di amplificazione
10             %componente in fase;
11 %fatt_stretching_y = fattore di amplificazione
12             %componente in quadratura;
13 %Cost_ottima = matrice contenente i punti della
14 %costellazione inseriti in modo ottimale;
15 %symbol_tx_complex = simboli trasmessi nel canale in
16 %forma complessa;
17 %VARIABILI D'USCITA:
18 %SER_non_ottimo = vettore colonna contenente il symbol
19 %error rate al variare del rapporto SNR introdotto

```

```

20 %dal canale;
21 function SER_non_ottimo = Calcolo_SER_non_ottimo(M,...
22 symbol_tx , Cost_stretching , start , fatt_stretching_x ,...
23 fatt_stretching_y , Cost_ottima , symbol_tx_complex)
24 %Rotazione in senso orario della costellazione ruotata
25 %e strecciata in modo tale che i punti della costellazione
26 %siano paralleli all'asse x.
27 Cost_stretching;
28 a = 1;
29 b = 1;
30 s = 1;
31 for (r=1 : length(Cost_stretching))
32 for (c=1 : length(Cost_stretching))
33 if (c == 1)
34     point_cost(s,1) = Cost_stretching(r,c);
35     s = s+1;
36 else
37 %Calcolo lunghezza dei cateti e ipotenusa
38 %del triangolo rettangolo.
39 c_min = imag(Cost_stretching(a,b)) -...
40         imag(Cost_stretching(r,c));
41 c_max = real(Cost_stretching(a,b)) -...
42         real(Cost_stretching(r,c));
43 ipotenusa = sqrt(c_min^2+c_max^2);
44 %Calcolo angolo di rotazione.
45 beta = acos(abs(c_max)/ipotenusa);
46 %Punto da ruotare.
47 point = c_max + j*c_min;
48 point_rot = point*(cos(-beta)+j*sin(-beta));
49 %Punto con riferimento asse cartesiano(0,0).
50 point_cost(s,1) = -real(point_rot)+...
51                 real(Cost_stretching(a,b))...
52                 +j*(imag(point_rot)+...
53                 imag(Cost_stretching(a,b)));
54 s = s+1;
55 end
56 end
57 a = a+1;

```

```

58 end
59 s=1;
60 %Inserimento della nuova costellazione all'interno
61 %della matrice 'matr_new'.
62 for (r=1 : length(Cost_stretching))
63   for (c=1 : length(Cost_stretching))
64     matr_new(r,c) = point_cost(s,1);
65     s = s+1;
66   end
67 end
68
69 %CALCOLO DEI PUNTI MEDI DELLA NUOVA COSTELLAZIONE.
70 %Calcolo punto medio tra due punti adiacenti ed
71 %appartenenti alla stessa riga della matrice 'matr_new'
72 q=1;
73 for (r=1 : length(matr_new))
74   for (c=1 : length(matr_new)-1)
75     p_medi(q,1) = punti_medi(matr_new(r,c) ...
76                             ,matr_new(r,c+1));
77     q = q+1;
78   end
79 end
80 p_medio_row = p_medi;
81 %Calcolo punto medio tra due punti adiacenti ma
82 %appartenenti a due righe diverse della matrice Cost.
83 q = 1;
84 r=1;
85 while (q < length(matr_new))
86   p_medio1(q,1) = punti_medi(matr_new(r,1) ...
87                             ,matr_new(r+1,1));
88   q = q+1;
89   r = r+1;
90 end
91 p_medio_column = p_medio1;
92
93 %RICHIAMA LA FUNZIONE CHE CREA IL FLUSSO DI SIMBOLI
94 %RICEVUTI DAL MITTENTE.
95 t = 1;

```

```

96 while (start >=0)
97 SNR = start;
98 yrx = zeros(length(symbol_tx),1);
99 vet = Creazione_flusso_rx(symbol_tx_complex, start, ...
100     SNR, fatt_stretching_x, fatt_stretching_y);
101 %vettore contenente i simboli ricevuti espressi in
102 %forma complessa.
103 yrx = vet;
104 vet = zeros(length(symbol_tx),1);
105 %Per ogni punto espresso in forma complessa
106 %dal ricevitore stabiliamo a quale regione
107 %appartiene. Ricordiamo che il punto è stato
108 %ruotato al trasmettitore, il canale ha introdotto
109 %un rumore gaussiano ed eseguito uno stretching.
110 %punto_rx = yrx(1,1);
111 k = 1;
112 while(k <= length(yrx))
113 punto_rx = yrx(k,1);
114 %Calcolo lunghezza dei cateti e ipotenusa
115 %del triangolo rettangolo
116 for(r=1 : length(Cost_stretching))
117 c_min = imag(Cost_stretching(r,1)) - ...
118     imag(punto_rx);
119 c_max = real(Cost_stretching(r,1)) - ...
120     real(punto_rx);
121 ipotenusa = sqrt(c_min^2+c_max^2);
122 %Punto da ruotare.
123 point = -c_max - j*c_min;
124 point_rot_prova(r,1) = point*(cos(-beta)+j*sin(-beta));
125 %Vettore contenente solo la parte immaginaria
126 %dei punti del vettore point_rot.
127 point_rot_imag(r) = abs(imag(point_rot_prova(r,1)));
128 end
129 minimo = min(point_rot_imag);
130 %pos_x indica a quale riga il valore complesso
131 %appartiene all'interno della matrice Cost_stretching.
132 pos_x = search(minimo, length(point_rot_imag), ...
133     point_rot_imag);

```

```

134 point_rot = point_rot_prova(pos_x,1);
135 %Punto con riferimento asse cartesiano (0,0).
136 %Corrisponde al punto ricevuto ma solo strecciato
137 %e non ruotato.
138 point_cost_rif = real(point_rot)+...
139                 real(Cost_stretching(pos_x,1))...
140                 +j*(imag(point_rot)+...
141                 imag(Cost_stretching(pos_x,1)));
142 %Ricerca della colonna a cui appartiene
143 %il punto di 'yrx' di posizione (pos_x,1) che si sta
144 %analizzando.
145 point_real = real(point_cost_rif);
146 v = 1;
147 s = ((pos_x-1)*(sqrt(M)-1))+1;
148 % il vettore 'A' è un vettore colonna che contiene
149 %la parte reale dei soli punti di riga 'pos_y' della
150 %matrice 'p_medio_row'.
151 A = zeros(sqrt(M)-1,1);
152 while(v <= (length(Cost_stretching)-1))
153     A(v,1) = real(p_medio_row(s,1));
154     s = s+1;
155     v = v+1;
156 end
157 A((length(A)+1)) = point_real;
158 A = sort(A);
159 %pos_y contiene la colonna a cui appartiene
160 %il punto 'point_cost_rif'.
161 pos_y = search(point_real, length(A),A);
162 %Consideriamo il punto di coordinate (pos_y, pos_x)
163 %all'interno della matrice 'Cost_stretching' contenente
164 %i punti della costellazione ruotata e strecciata.
165 punto_Cost = Cost_stretching(pos_x, pos_y);
166 %Consideriamo la sola parte reale del punto appena
167 %trovato ed eseguiamo la ricerca della posizione
168 %di tale valore all'interno del vettore 'Cost_ottima'
169 prova = real(punto_Cost);
170 vet_real_ottima = real(Cost_ottima);
171 pos_cost_ottima = search(prova, length...
```

```

172             (vet_real_ottima),vet_real_ottima);
173 %La posizione del punto all'interno del vettore
174 %'Cost_ottima' coincide con la regione di decisione
175 %a cui il punto ricevuto 'yrx' appartiene.
176 %Tale valore viene inserito nel vettore 'symbol_rx'.
177 symbol_rx(k,1) = pos_cost_ottima-1;
178 k = k+1;
179 end
180 symbol_rx;
181 SER_non_ottimo(t,1) = symbol_error_rate...
182             (symbol_tx ,symbol_rx);
183 t = t+1;
184 start = start-1;
185 end
186 SER_non_ottimo;
187 end %FINE FUNCTION.

```

### Calcolo\_SER\_ottimo.m

```

1 %FUNZIONE CHE CALCOLA IL SYMBOL ERROR RATE IPOTIZZANDO
2 %CHE LO SPAZIO CARTESIANO IN CUI E' CONTENUTA LA
3 %COSTELLAZIONE SIA SUDDIVISO IN REGIONI DI DECISIONE OTTIME.
4 %VARIABILI D'INGRESSO:
5 %Cost_stretching = matrice contenente i punti...
6 %della costellazione iniziale ruotati e strecciati;
7 %Cost_ottima = matrice contenente i punti della
8 %costellazione inseriti in modo ottimale;
9 %symbol_tx_complex = simboli trasmessi nel canale in
10 %forma complessa;
11 %symbol_tx = simboli trasmessi nel canale;
12 %SNR = rapporto segnale rumore introdotto dal canale
13 %trasmissivo espresso in db;
14 %fatt_stretching_x = fattore di amplificazione
15 %componente in fase;
16 %fatt_stretching_y = fattore di amplificazione
17 %componente in quadratura;
18 %VARIABILI D'USCITA:
19 %SER_ottimo = vettore colonna contenente il symbol

```

```

20 %error rate al variare del rapporto SNR introdotto
21 %dal canale;
22 function SER_ottimo = Calcolo_SER_ottimo(Cost_stretching ,...
23 Cost_ottima ,symbol_tx_complex ,symbol_tx ,start ,SNR,...
24 fatt_stretching_x ,fatt_stretching_y)
25 t = 1;
26 while (start >=0)
27 SNR = start;
28 yrx = zeros (length(symbol_tx) ,1);
29 %richiama la funzione 'Creazione_flusso_rx' che manda
30 %in uscita i simboli ricevuti dal mittente.
31 vet = Creazione_flusso_rx(symbol_tx_complex ,start ,...
32 SNR,fatt_stretching_x ,fatt_stretching_y);
33 %vettore contenente i simboli ricevuti espressi
34 %in forma complessa.
35 yrx = vet;
36 vet = zeros (length(symbol_tx) ,1);
37 w = 1;
38 x_cost = real(Cost_ottima);
39 y_cost = imag(Cost_ottima);
40 while (w <= length(yrx))
41 punto = yrx(w,1);
42 d = zeros (16,1);
43 for (q=1 : length(x_cost))
44 d(q) = sqrt ((x_cost(q,1) - real(punto))^2 ...
45 + (y_cost(q,1) - imag(punto))^2);
46 end
47 dist_min = min(d);
48 length_vet = length(x_cost);
49 pos_min = search(dist_min ,length_vet ,d);
50 symbol_rx(w,1) = pos_min-1;
51 w = w+1;
52 end
53
54 symbol_rx;
55 SER_ottimo(t,1) = symbol_error_rate...
56 (symbol_tx ,symbol_rx);
57 t = t+1;

```

```

58 start = start - 1;
59 end
60
61 SER_ottimo;
62 end %FINE FUNCTION

```

### symbol\_error\_rate.m

```

1 %FUNZIONE CHE CALCOLA IL SYMBOL ERROR RATE, OVVERO
2 %ESEGUE UN CONFRONTO TRA IL VETTORE Symbol_rx e
3 %Symbol_tx, RISPETTIVAMENTE VETTORI CONTENENTI I
4 %SIMBOLI RICEVUTI E QUELLI TRASMESSI, CALCOLANDO
5 %IL NUMERO DI SIMBOLI DECODIFICATI IN MODO ERRATO.
6 %IL RAPPORTO TRA IL NUMERO DI SIMBOLI ERRATI E IL
7 %NUMERO TOTALE DI SIMBOLI INVIATI COINCIDE CON IL
8 %SYMBOL ERROR RATE.
9 %VARIABILI D'INGRESSO:
10 %-symbol_tx = vettore colonna contenente i simboli trasmessi
11 %-symbol_rx = vettore colonna contenente i simboli ricevuti
12 %VARIABILI D'USCITA:
13 %-ser = symbol error rate
14 function SER = symbol_error_rate(symbol_tx, symbol_rx)
15 v = 1;
16 s = 0;
17 while(v <= length(symbol_tx))
18     if(symbol_tx(v,1) ~= symbol_rx(v,1))
19         s = s+1;
20     end
21     v = v+1;
22 end
23 SER = s/length(symbol_tx);
24 end %FINE FUNCTION

```

### search.m

```

1 %FUNZIONE CHE ESEGUE LA RICERCA DELLA POSIZIONE
2 %DEL VALORE minimo PASSATO COME PARAMETRO
3 %ALL'INTERNO DELL'ARRAY.
4 %VARIABILI D'INGRESSO:

```

```

5 %vet_dist = contiene i punti della costellazione;
6 %minimo = valore che si vuole ricercare nell'array;
7 %VARIABILI D'USCITA:
8 %pos = posizione in cui è presente il valore minimo
9 %      passato come parametro all'interno dell'array;
10 function pos = search(minimo,length_vet,vet_dist)
11     for (m=1 : length_vet)
12         if (vet_dist(m) == minimo)
13             pos = m;
14             m = length_vet+2;
15         end
16     end
17 end

```

### punti\_medi.m

```

1 %FUNZIONE CHE CALCOLA LE COORDINATE DEL PUNTO MEDIO TRA
2 %DUE PUNTI DATI IN INGRESSO ALLA FUNZIONE.
3 %VARIABILI D'INGRESSO:
4 %p1,p2 = coordinate dei due punti di cui si vuole
5 %      calcolare il punto medio;
6 %VARIABILI D'USCITA:
7 %p_medio = coordinate del punto medio.
8 function p_medio = punti_medi(p1, p2)
9     p1_real = real(p1);
10    p2_real = real(p2);
11    p1_imag = imag(p1);
12    p2_imag = imag(p2);
13    d_real = p1_real -((p1_real - p2_real)/2);
14    d_imag = p1_imag -((p1_imag - p2_imag)/2);
15    p_medio = d_real+ j*d_imag;
16 end %FINE FUNCTION

```



# Elenco delle figure

1.1	Stream Adaption. . . . .	15
1.2	Struttura BBFRAME all'uscita del blocco Stream Adaptor. . .	16
1.3	Bit Interleaved Code Modulation. . . . .	17
1.4	Struttura Frame Builder. . . . .	19
1.5	Struttura OFDM Generator. . . . .	21
2.1	Schema per la trasmissione e ricezione modulazione OFDM. . .	28
3.1	Costellazione 16-QAM. . . . .	30
3.2	Costellazione 16-QAM al trasmettitore. . . . .	38
3.3	Regioni di decisione Costellazione 16-QAM. . . . .	38
3.4	Costellazione 16-QAM ruotata di un angolo $\Phi=16$ . . . . .	39
3.5	Regioni di decisione costellazione ruotata di un angolo $\Phi=16$ . .	41
3.6	Costellazione al ricevitore. . . . .	41
3.7	Regioni di decisione costruite al ricevitore. . . . .	43
3.8	Andamento symbol error rate. . . . .	45



# Bibliografia

- [1] David Tse, Pramod Viswanath, “*Fundamentals of Wireless Communication*”, Cambridge University Press, 2005
- [2] N. Benvenuto, R. Corvaja, T. Erseghe, N. Laurenti, “*Communication System (Fundamentals and Design Methods)*”, University of Padova, John Wiley & Sons, Ltd 2007
- [3] Lorenzo Vangelista, Nevio Benvenuto and Stefano Tomasin, (University of Padova), Chris Nokes and Jonathan Stott, (BBC Research), Alessio Filippi and Marnix Vlot, (Philips), Vittoria Mignone and Alberto Morello, (RAI CRIT), “*Key Technologies for Next-Generation Terrestrial Digital Television Standard DVB-T2*”, Dicembre 2008
- [4] DVB-T2 Standard [<http://www.dvb.org/technology/standards>], “*Digital Video Broadcasting (DVB);Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)*”, Settembre 2009.
- [5] DVB-T2 Standard [<http://www.dvb.org/technology/standards>], “*Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2) (draft TR 102 831 V1.1.1)*”, Dicembre 2009.
- [6] DVB-T2 Standard [[http://www.dvb.org/technology/fact\\_sheets](http://www.dvb.org/technology/fact_sheets)], “*DVB-T2 - 2nd Generation Terrestrial Broadcasting*”, Marzo 2010.

- [7] DVB-T2 Standard [[http://www.dvb.org/news\\_events/dvbscene\\_magazine](http://www.dvb.org/news_events/dvbscene_magazine)], “*DVB-Scene 27 : T2 Extra Terrestrial*”, Agosto 2008.
- [8] Ing. Vittoria Mignone, RAI CRIT [<http://www.crit.rai.it/eletel/2005-3>], “*Che cosa è, come funziona: La modulazione COFDM*”, Dicembre 2005.
- [9] D. Perez-Calderon, C. Oria, J. Garcia, P. Lopez, V. Baena, I. Lacadena, Electronic Engineering Department, University of Seville Escuela Superior de Ingenieros, Avda. “*Rotated Constellation for DVB-T2*”.