

Corso di Laurea Triennale
in Ingegneria dell'Informazione

TECNICHE DI SINCRONIZZAZIONE
PER SISTEMI VIDEO DIGITALI
TERRESTRI DI NUOVA GENERAZIONE

Candidato:

Ermanno Lazzarin, 542433

Relatore:

Prof. Stefano Tomasin

A.A. 2009/10

Sommario

1	Introduzione	4
2	Lo standard DVB-T2.....	6
2.1	Cenni storici e requisiti.....	6
2.2	Caratteristiche principali	6
2.3	Il modello architetturale	7
2.4	Scheduling	8
2.5	Tecniche di modulazione.....	9
2.5.1	Costellazioni ruotate	9
2.5.2	Riduzione della PAPR (Peak-to-Average Power Ratio).....	10
2.6	Sincronizzazione e stima del canale	10
2.6.1	Il simbolo P1	11
2.6.2	Il simbolo P2	12
2.6.3	Configurazione dei segnali pilota	12
2.6.4	Altre funzionalità del simbolo P2	13
2.7	Trasmissione con antenne multiple	14
2.8	Considerazioni finali	15
3	La modulazione OFDM	16
3.1	Principali caratteristiche.....	16
3.2	Vantaggi e svantaggi.....	16
3.3	Principio di funzionamento	17
3.4	Considerazioni finali	20
4	Sincronizzazione nello standard DVB-T2	22
4.1	Scopo.....	22
4.2	Determinazione dell'intervallo di guardia.....	22
4.2.1	Guard interval estimation not normalized.....	23
4.2.2	Guard interval estimation normalized	24
4.2.3	Guard interval estimation adaptive threshold	24
4.2.4	Guard interval estimation bubble	25
4.2.5	Guard interval estimation optimum.....	26
4.3	Valutazioni e precisazioni.....	28
4.3.1	Mezzo trasmissivo	28
4.3.2	Contributo del simbolo P1 nell'analisi.....	29

4.3.3	Considerazioni sulla struttura del frame OFDM.....	29
4.4	Simulazioni e risultati	30
4.4.1	Un caso di particolare rilevanza: $N_{\text{fft}}=1\text{K}$	30
4.4.2	Altre possibili configurazioni	35
5	Considerazioni finali e conclusioni	38
6	Bibliografia	40
7	Appendice	42

1 Introduzione

Diversi paesi europei hanno dimostrato da alcuni anni interesse nello sviluppo di uno standard di trasmissione per la televisione digitale terrestre, in grado di assicurare prestazioni più elevate rispetto al sistema corrente. In particolare l'introduzione della TV in alta definizione ha portato alla creazione dello standard DVB-T2 per i digitali terrestri di nuova generazione.

Rispetto all'ormai consolidato standard DVB-T, questa seconda generazione di tecnologie offre considerevoli miglioramenti in termini di efficienza energetica e spettrale. Infatti garantisce a parità di banda un aumento dei canali TV e della loro qualità, il tutto diminuendo la potenza dei siti di trasmissione in relazione con le comuni antenne degli utenti.

In questo elaborato sono presentate alcune tecniche per gestire la sincronizzazione di una trasmissione DVB-T2. In particolare l'analisi si concentra sulle prestazioni dei vari algoritmi in canali sui quali agisce rumore gaussiano.

2 Lo standard DVB-T2

2.1 Cenni storici e requisiti

Nel 2006 nasce, in ambito DVB, un gruppo di lavoro focalizzato sull'obiettivo di definire una nuova piattaforma per la televisione digitale terrestre. Il gruppo è denominato TM-T2 (Technical Module on Next Generation DVB-T), e ha come primo obiettivo la progettazione di uno schema di modulazione avanzato che possa essere adottato da uno standard televisivo digitale terrestre di seconda generazione, da chiamarsi *DVB-T2*.

Nell'aprile 2007, dopo l'approvazione del documento di specifica dei requisiti commerciali, viene lanciata una richiesta di proposte tecniche volte al soddisfacimento di detti requisiti, mentre a luglio iniziano le attività di standardizzazione vera e propria.

Nel giugno 2008, dopo un anno di intenso lavoro, cui hanno partecipato attivamente circa settanta esperti appartenenti a più di quaranta aziende, il DVB vara il "Blue-Book A122" che costituisce il DVB-T2, lo standard di seconda generazione per la televisione digitale terrestre, e lo propone in ETSI per divenire "Norma Europea".

I principali requisiti definiti dal modulo commerciale del DVB per lo standard di seconda generazione di TV digitale terrestre comprendono:

- Utilizzo degli impianti d'antenna domestici esistenti e dell'attuale infrastruttura di trasmettitori;
- Incremento di almeno il 30 % della capacità trasmissiva rispetto al DVB-T, pur con gli stessi vincoli di occupazione spettrale;
- Prestazioni superiori per SFN (Single Frequency Network);
- Meccanismi per fornire caratteristiche di robustezza in funzione del tipo di servizio;
- Flessibilità relativamente a frequenze operative e larghezza di banda;
- Meccanismi per ridurre il rapporto tra potenza di picco e media, ottenendo vantaggi sulle emissioni elettromagnetiche e sul costo dei trasmettitori.

2.2 Caratteristiche principali

L'elenco seguente costituisce un riassunto delle caratteristiche dello standard DVB-T2, alcune delle quali saranno descritte con maggiore dettaglio nei paragrafi successivi:

- Stessa tecnica di modulazione utilizzata dal DVB-T, ovvero OFDM con intervallo di guardia;
- Dimensioni della FFT (Fast Fourier Transform) estese rispetto allo standard corrente, in grado di migliorare le prestazioni in configurazioni di rete a frequenza singola, e di incrementare i possibili valori degli intervalli di guardia;
- Tecnica di suddivisione dei dati in frame;
- Stesse costellazioni del DVB-T (4-, 16-, 64-QAM), con estensione alla 256-QAM, per sfruttare appieno i vantaggi derivanti dall'impiego della nuova generazione di FEC. Viene inoltre introdotta la tecnica delle costellazioni ruotate;

- Metodo per trasportare servizi differenti con differenti gradi di protezione, in un unico canale fisico, ma diversi canali logici, denominati PLP. Ciascun PLP adotta FEC indipendenti, consentendo così di adattare le caratteristiche di robustezza della codifica ai requisiti del particolare servizio convogliato;
- Struttura di trama flessibile, in cui i dati possono essere dispersi sull'intera trama per avere la massima diversità temporale oppure concentrati in bursts per consentire il massimo risparmio energetico nel ricevitore;
- Ampliamento dei possibili segnali per la sincronizzazione e la stima del canale radio;
- Possibilità di trasmissione con antenne multiple, basata sulla tecnica di Alamouti, che consente di migliorare la ricezione di segnali di pari livello provenienti da due trasmettitori;
- Riduzione del rapporto tra potenza di picco e potenza media attraverso tecniche di "tone reservation" e "constellation distortion";
- Segnalazione (a basso livello di potenza) per l'identificazione del trasmettitore;
- Meccanismi per permettere in futuro l'introduzione nel sistema di ulteriori tecniche preservandone la compatibilità;
- Segnali pilota continui per l'eliminazione dell'errore di fase comune;
- Utilizzo di simboli P1 per la sincronizzazione delle trame e la rilevazione rapida del segnale DVB-T2;
- Utilizzo di simboli P2 con dati che descrivono la trama e informazioni di tipo PSI/SI.

2.3 Il modello architetturale

La struttura a blocchi del sistema trasmettente DVB-T2 è riportata in Figura 1. Essa prevede l'utilizzo di più canali logici (PLP) in modo da poter impiegare FEC diverse per ciascuno di questi.

Nella parte inferiore del diagramma si nota la presenza del blocchetto IFFT che realizza l'Inverse Fast Fourier Transform, nucleo della modulazione OFDM presentata nel capitolo 3.

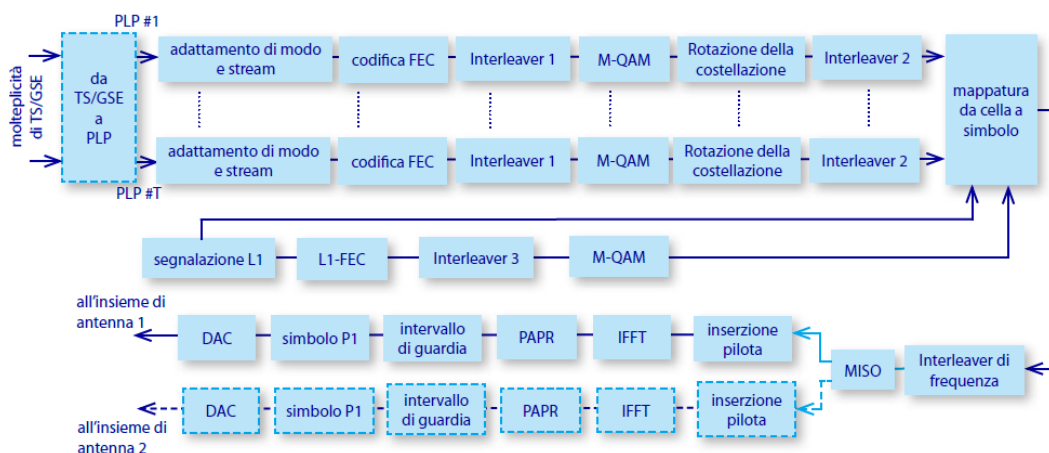


Fig. 1 - Diagramma a blocchi del sistema trasmettente DVB-T2

2.4 Scheduling

La struttura di trama DVB-T2 è riportata in Figura 2.

Nel livello più alto la struttura è composta da super-frame (super-trame) suddivise in trame le quali, a loro volta, sono composte da simboli OFDM.

Il numero di frame per ogni super-frame è definito in modo che, per ogni PLP di dati, vi sia un numero intero di Interleaving Frames in ogni super-trama.

Le trame DVB-T2 iniziano con un simbolo di riferimento denominato P1, differente da un normale simbolo OFDM, seguito da uno o più simboli di riferimento P2, il cui scopo principale è di trasferire la segnalazione. Infine, segue un numero configurabile di simboli OFDM di dati. La durata delle trame può essere dell'ordine di 100÷250 ms.

La capacità di un PLP può variare nel tempo, conseguentemente la posizione in tempo e frequenza delle celle associate a un PLP può cambiare da trama a trama. I dati di un PLP possono non essere mappati su tutte le trame, ma sono possibili salti di più trame.

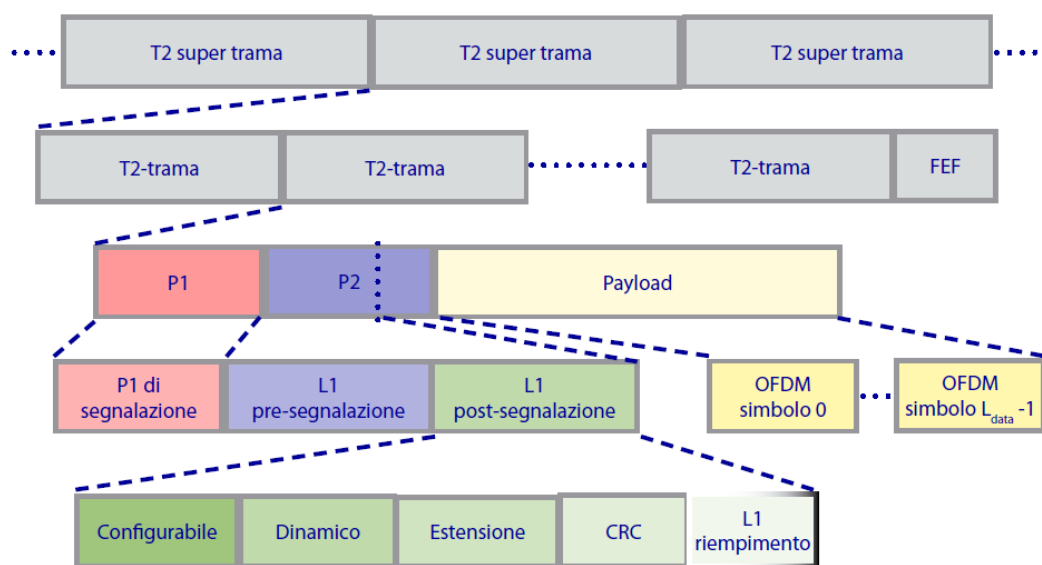


Fig. 2 - Struttura di trame e super-trame del DVB-T2

Dal momento che il DVB-T2 può essere soggetto a disturbi di tipo impulsivo, la segnalazione della posizione delle celle (le cosiddette informazioni L1 dinamiche) ha richiesto accorgimenti particolari (con meccanismi di correzione di errori e trasmissione ripetuta) per consentirne una trasmissione il più possibile robusta. Infatti la segnalazione L1 è trasferita in ogni trama nel simbolo P2.

2.5 Tecniche di modulazione

La tecnica definita dal DVB-T2 riprende quella già utilizzata dallo standard DVB-T, ovvero la modulazione OFDM con intervallo di guardia.

Sono sei le possibili configurazioni del numero di punti su cui viene effettuata la FFT nel modulatore per il DVB-T2: 1K, 2K, 4K, 8K, 16K, 32K. Maggiore è il numero di portanti (a parità di larghezza di banda complessiva), minore è la spaziatura tra di esse e maggiore è la durata del periodo di simbolo: ciò rende il sistema più critico in presenza di interferenze di tipo intercarrier, e di effetto Doppler.

Ulteriori vantaggi nell'operare con FFT di dimensioni elevate sono: maggiore robustezza nei confronti del rumore impulsivo, e inferiori livelli di densità di potenza fuori banda.

2.5.1 Costellazioni ruotate

Nelle costellazioni tradizionali, le coordinate dei differenti punti sono allineate orizzontalmente (per ordinata) o verticalmente (per ascissa). Risultano quindi più punti aventi o la stessa ascissa o la medesima ordinata. La distinzione tra due di questi punti è possibile solo con entrambe le coordinate, non essendoci correlazione tra i due assi. Con la tecnica di rotazione, la costellazione prescelta è ruotata nel piano complesso "I-Q" in modo tale che non esistano più punti con una coordinata uguale.

Ciascuna coordinata del sistema ruotato è quindi sufficiente per riconoscere il punto della costellazione e le doppie coordinate (ascissa ed ordinata di ciascun punto) risultano quindi ridondanti.

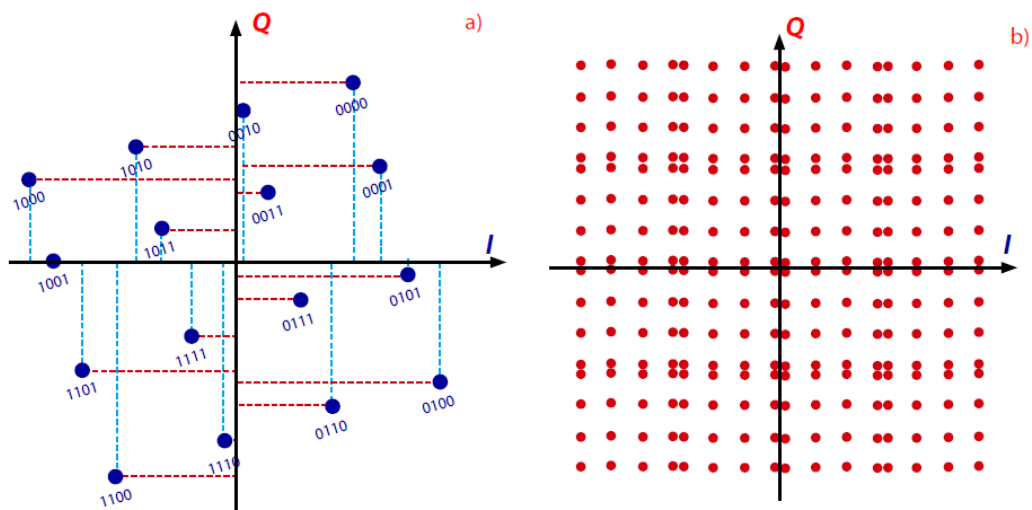


Fig. 3 – a) Costellazione 16-QAM ruotata prima del ritardo ciclico di Q
b) costellazione 16-QAM ruotata dopo il ritardo ciclico di Q

Ad esempio per un sistema 16-QAM i punti della costellazione dopo rotazione assumono per ascissa e ordinata sedici valori distinti (Figura 3a).

Di per sé questa tecnica non dà vantaggi, ma grazie all'introduzione di un ritardo ciclico per l'ordinata Q prima dell'interleaving in tempo e frequenza (Figura 3b), ascissa e ordinata di ciascun punto sono trasferite su differenti celle OFDM. Dopo la riunificazione dei valori di I e Q a valle del de-interleaving al ricevitore, se una delle due coordinate è compromessa per effetto dei disturbi selettivi sul canale radio, l'altra coordinata, se ricevuta correttamente, è in grado, da sola, di consentire una decodifica univoca del simbolo trasmesso.

Tale tecnica non degrada le prestazioni su canale Gaussiano, mentre fornisce, in canali affetti da fading, un guadagno che può raggiungere anche valori molto elevati nel caso di mezzi molto critici.

2.5.2 Riduzione della PAPR (Peak-to-Average Power Ratio)

All'aumentare del numero di portanti, il segnale OFDM tende sempre più ad assomigliare a rumore Gaussiano, con la conseguenza che il rapporto tra potenza di picco e potenza media (PAPR) risulta elevato.

Ciò impone dei vincoli sul funzionamento degli amplificatori di potenza nei trasmettitori, che provocano una riduzione dell'efficienza degli stessi.

Nello standard DVB-T2 si cerca di ridurre questo inconveniente adottando l'utilizzo combinato di due tecniche:

- ACE (Active constellation extension): questa tecnica prevede la modifica delle costellazioni di modulazione muovendo i punti più esterni della costellazione stessa verso ampiezze maggiori;
- Reserved-carrier PAPR Reduction: con questo metodo si riservano alcune portanti non utilizzate per trasportare dati. Tali portanti sono modulate con ampiezza e fase opportuna per evitare picchi di ampiezza sul segnale OFDM.

2.6 Sincronizzazione e stima del canale

Il segnale DVB-T2 è progettato in modo tale da facilitare la sincronizzazione dei ricevitori in tempo e frequenza. Ciascuna trama è suddivisa in preambolo e payload, così come riportato graficamente in Figura 2.

Il preambolo è costituito da un simbolo denominato P1 e da un certo numero di simboli P2, dipendente dalla dimensione della FFT utilizzata.

Segue il payload vero e proprio, (anche se alcuni dati possono essere già trasportati da simboli P2) che consiste di simboli OFDM le cui sottoportanti possono essere modulate da simboli di dati o da simboli pilota predefiniti.

2.6.1 Il simbolo P1

Il simbolo P1 è costituito da un simbolo OFDM su 1K portanti e da una particolare struttura di replicazione nel tempo, come illustrato in Figura 4. La sezione indicata con C è la versione traslata in frequenza dei primi 542 campioni del simbolo OFDM, che costituisce la sezione indicata con A. La sezione B è invece la versione traslata in frequenza degli ultimi 482 campioni della parte A. La traslazione in frequenza risulta pari alla spaziatura delle sottoportanti del simbolo OFDM.

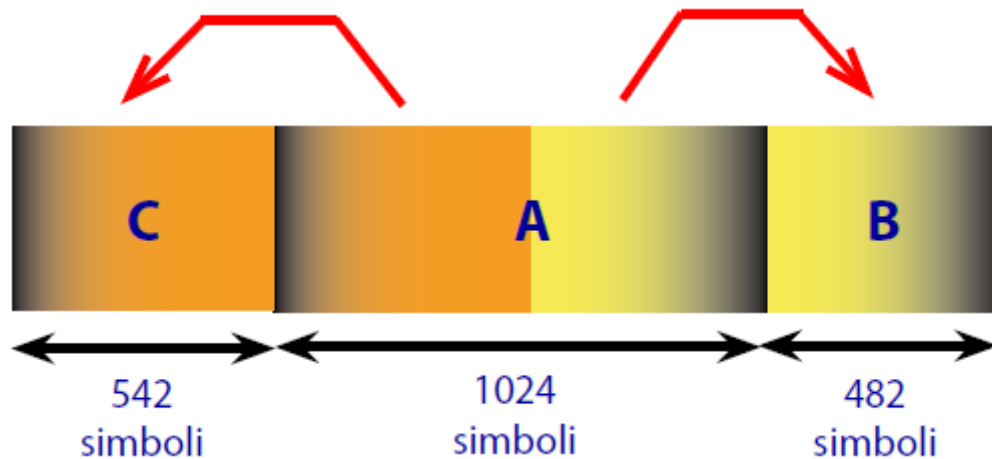


Fig. 4 – Formato e rappresentazione in frequenza del simbolo P1

La struttura prefissata del simbolo P1, con le limitate informazioni di segnalazione trasferite in modalità estremamente robusta, consente al ricevitore di effettuare una scansione veloce delle frequenze trasmissive.

La rivelazione del simbolo P1 può essere utilizzata dal ricevitore come primo riferimento in tempo e frequenza. Il ricevitore può pertanto rilevare la presenza di una trasmissione DVB-T2 ed acquisire alcuni parametri chiave quali la dimensione della FFT o la presenza di Frame FEF. La particolare struttura C-A-B è stata definita per incrementare la capacità di rivelazione del simbolo P1 anche nei canali più critici.

2.6.2 Il simbolo P2

I simboli P2 trasportano la segnalazione di livello 1 della trasmissione (L1), che diventa rilevante in presenza di PLP multipli, in quanto ciascuno di essi può possedere propri parametri di trasmissione.

La segnalazione L1 è organizzata in una parte di pre-segnalazione, contenente informazioni di tipo statico, costanti per tutta la durata del super-frame (ad esempio la dimensione dell'intervallo di guardia), ed una parte post-segnalazione, contenente informazioni di tipo dinamico. I simboli P2 contengono anche le portanti pilota e permettono di iniziare la procedura di stima di canale.

Un esempio di struttura completa della trama dei simboli P2 è riportato in Figura 5.

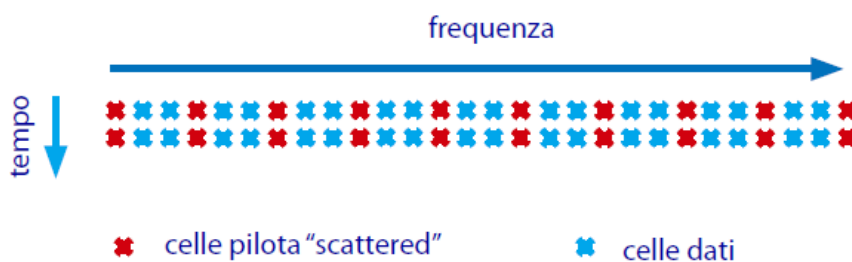


Fig. 5 – Simboli P2 nella modalità 8K e rispettive celle pilota

2.6.3 Configurazione dei segnali pilota

Anche lo standard DVB-T2, come il precedente DVB-T, prevede la dispersione in tempo e frequenza di portanti pilota "scattered" (SP) e continue (CP), cioè di celle dati OFDM di ampiezza e fase prefissate, trasmesse ad intervalli regolari per consentire al ricevitore di stimare le variazioni del canale trasmissivo ed effettuare le opportune compensazioni in tempo e frequenza. Le SP hanno come compito principale quello di permettere una stima affidabile del canale di trasmissione.

A differenza del DVB-T, che prevedeva una sola configurazione di portanti pilota "scattered" per tutti i valori di intervalli di guardia, nel DVB-T2 ne sono definite otto, con distanza tra le portanti circa uguale all'inverso della durata dell'intervallo di guardia.

Le CP invece sono utilizzate per il recupero "fine" di frequenza nel ricevitore e per la rimozione dell'errore di fase comune sul simbolo OFDM.

Solo nel simbolo P2 la struttura delle portanti pilota è fissa e pari a quella del massimo intervallo di guardia possibile, che si assume possa essere acquisito con l'impiego di metodi tradizionali basati su operazioni di correlazione. Il simbolo pilota e la posizione delle sottoportanti nel simbolo P2 sono indipendenti dagli altri parametri di trasmissione, come ad esempio l'estensione di banda ed i metodi PAPR.

Il valore assunto dalle celle pilota SP o CP dipende dall'indice della portante k , come nel DVB-T; ma, a differenza di quest'ultimo, tali valori dipendono anche dall'indice n del simbolo OFDM. Tutti i simboli pilota (CP, SP, P2) in ogni simbolo OFDM sono moltiplicati per 1 o -1, conformemente ad una sequenza pseudo casuale a livello di trama. Ciò consente un approccio più robusto relativamente alla sincronizzazione, in grado di fornire la posizione attuale all'interno della trama anche nel caso in cui il preambolo venga perso a causa, per esempio, di un forte rumore di tipo impulsivo.

Inoltre, gli algoritmi di sincronizzazione possono sfruttare questa sequenza a livello di trama per stimare e monitorare la sincronizzazione a livello di clock, simbolo, frequenza e trama.

2.6.4 Altre funzionalità del simbolo P2

Il simbolo P2 si trova immediatamente dopo la ricezione del pacchetto P1: dunque, dopo aver stimato la grandezza della FFT, al ricevitore è noto anche il numero di simboli P2.

La segnalazione L1, che ci si pone di determinare, è "spalmata" su tutti i simboli P2 presenti, ed è quindi necessario salvare quest'ultimi prima di iniziare il processo di decodifica.

Nel simbolo P2 la struttura delle portanti pilota è fissa e pari a quella del massimo intervallo di guardia possibile. Il simbolo pilota e la posizione delle sottoportanti nel simbolo P2 sono indipendenti dagli altri parametri di trasmissione come ad esempio l'estensione di banda. Il fatto di usare un numero così elevato di portanti pilota ha le sue conseguenze. Si ha necessariamente una perdita di capacità di trasmissione, ma l'effetto negativo può considerarsi poco influente in quanto il numero di simboli P2 è piuttosto limitato. Questa soluzione garantisce però al ricevitore di poter procedere nella scansione del frame utilizzando solamente i pochi parametri ricavati dall'analisi del simbolo P1.

La presenza di queste portanti pilota permette la realizzazione di una prima stima del canale mediante interpolazione temporale. A differenza però delle altre portanti presenti nel corpo del frame, quelle relative al P2 hanno una potenza inferiore, esponendosi così in modo più sensibile al rumore. Tuttavia queste portanti sono molto ravvicinate, garantendo la possibilità di agire in frequenza prima di realizzare l'interpolazione temporale. Si ottiene in questo modo una stima meno intaccata dal rumore.

In Figura 6 è rappresentata l'azione del ricevitore che calcola la stima della risposta del canale (H) mediante rapporto tra simbolo trasmesso e simbolo ricevuto relativo alle portanti pilota.

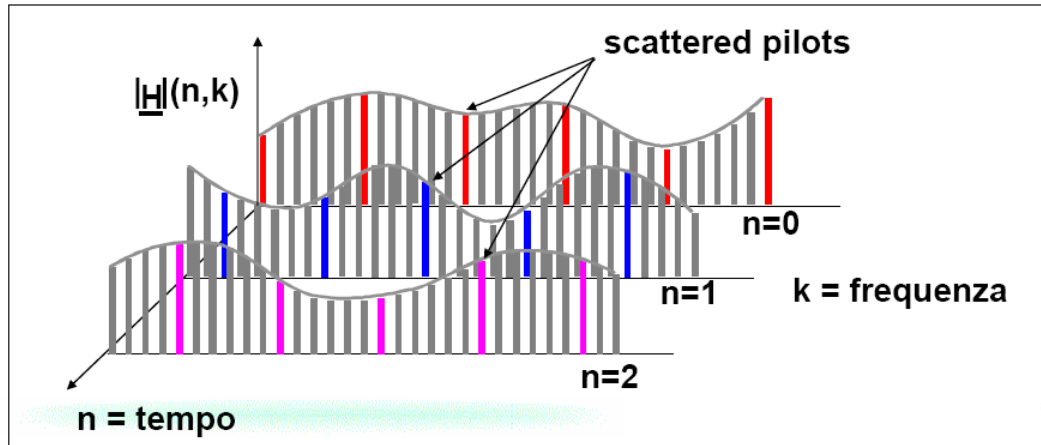


Fig. 6 – Portanti pilota utilizzate per la stima di canale

2.7 Trasmissione con antenne multiple

Lo standard DVB-T2, come già il DVB-T, consente la trasmissione simultanea del segnale sulla stessa frequenza da parte di differenti trasmettitori al fine di realizzare una rete a singola frequenza SFN.

Garantendo stringenti requisiti di sincronizzazione, in una rete SFN i ricevitori “vedono” un canale equivalente ottenuto per sovrapposizione dei canali relativi a più trasmettitori.

Quando si riscontra la presenza di due segnali di livello elevato provenienti da due trasmettitori distinti di una stessa rete, si ha un caso critico per la modalità SFN. Infatti questo comporta una significativa riduzione di margine in quanto il canale risultante può presentare profondi “buchi” (notches) nello spettro di frequenza.

Lo standard DVB-T2 introduce l'uso di una strategia che prevede la trasmissione di due segnali differenti da parte di due trasmettitori adiacenti, secondo la tecnica di Alamouti. Si tratta di un approccio efficiente per sfruttare la presenza di molteplici antenne trasmissive, utile anche nel caso in cui il ricevitore sia equipaggiato con un'antenna singola (MISO). Con ciò si ottiene un incremento di copertura in reti SFN.

I segnali delle due antenne trasferiscono gli stessi dati ma codificati in maniera differente in spazio e frequenza, in modo tale che, in ricezione, sia possibile combinarli in maniera semplice per ottenere la decodifica ottima dei simboli originali.

2.8 Considerazioni finali

Si tratta di un ottimo standard che richiede però uno sviluppo completamente nuovo dell'elettronica per la codifica di sorgente e di canale, pertanto sarà necessaria la sostituzione degli IRD domestici.

In ogni caso l'efficienza spettrale è prossima al limite teorico (Shannon), dunque si tratta di uno standard stabile e duraturo nel tempo.

I siti di trasmissione e gli impianti d'antenna domestici poi non dovrebbero richiedere modifiche.

Un altro vantaggio è la grande flessibilità: sia in termini di varietà dei dati in ingresso, sia per la possibilità di rendere più robusta la trasmissione anche in condizioni critiche.

Lo standard rappresenta il driver per l'introduzione dei servizi HDTV. L'impiego dello standard DVB-T2, congiuntamente a codifiche video di ultima generazione (MPEG-4), consente di ottenere un guadagno di capacità trasmissiva utilizzabile per introdurre servizi HDTV quasi a costo nullo in termini di occupazione di banda.

Tali aspetti sembrano costituire i punti chiave che determineranno nel prossimo futuro il possibile successo dello standard.

3 La modulazione OFDM

3.1 Principali caratteristiche

In questo capitolo sono analizzate le proprietà della modulazione OFDM (Ortogonal Frequency Division Multiplexing). Molte comunicazioni wireless, infatti, hanno iniziato a utilizzare questa tecnica per incrementare le prestazioni.

Le principali caratteristiche di questa modulazione sono:

- Nella trasmissione parallela di dati, la banda di frequenza disponibile è generalmente suddivisa in molti canali;
- La modulazione avviene tra le varie portanti in modo indipendente;
- I vari canali occupano bande relativamente piccole, quindi meno sensibili agli effetti della distorsione dei canali selettivi in frequenza e con rumore impulsivo;
- Si riduce l'effetto dell'interferenza intercanale (ICI), grazie alle relazioni di ortogonalità fra canali.

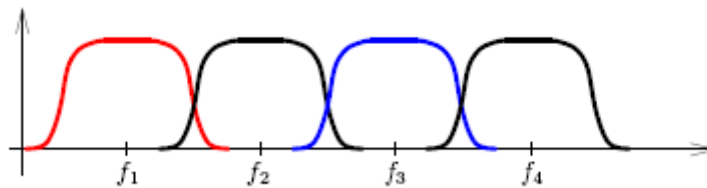


Fig. 7 – Tecnica di trasmissione OFDM

3.2 Vantaggi e svantaggi

Questa tecnica di modulazione garantisce l'introduzione di molteplici benefici:

- Capacità di adattarsi anche in condizioni problematiche per il canale senza complicare il sistema;
- Robustezza rispetto all'interferenza intercanale ICI;
- Robustezza all'ISI e al fading multi spettrale;
- Alta efficienza spettrale;
- Implementazione efficace usando la FFT;
- Bassa sensibilità agli errori di sincronizzazione.

Allo stesso tempo però è caratterizzata anche da alcuni svantaggi:

- Sensibilità all'effetto Doppler;
- Sensibilità ai problemi di sincronizzazione in frequenza;
- Alto Peak-to-Average Power Ratio (PAPR);
- Perdita di efficienza dovuta all'inserimento del prefisso ciclico e dell'intervallo di guardia.

3.3 Principio di funzionamento

In Figura 8 è rappresentata la tipica risposta impulsiva $c(t)$ di un canale. Tipicamente il symbol rate T è molto minore rispetto a T_d e quindi i segnali sono distribuiti su frequenze diverse.

I canali multipath sono caratterizzati da una risposta impulsiva data da:

$$c(t) = \sum_{l=0}^{M-1} \lambda_l g(t - \zeta_l)$$

Dove $g(t)$ è il filtro di trasmissione, i numeri complessi λ_l rappresentano i guadagni delle rispettive porzioni di canale, e ζ_l sono ritardi temporali ($l=0, \dots, M-1$). Questi ritardi, come anche i guadagni, sono spesso determinati con misurazioni pratiche.

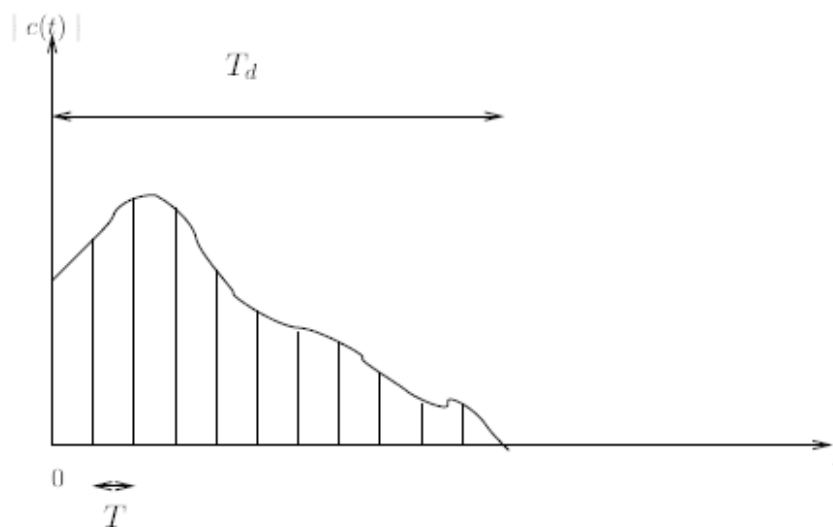


Fig. 8 – Multi-path channel

L'idea principale della modulazione OFDM è quella di trasformare il rapporto di convoluzione esistente tra il canale e il messaggio in ingresso, in una relazione moltiplicativa, allo scopo di semplificare notevolmente le operazioni di estrazione del messaggio stesso al demodulatore. Per fare ciò viene introdotto un prefisso ciclico di ridondanza, il quale serve a rendere la convoluzione tra il segnale trasmesso e la risposta all'impulso del canale equivalente alla convoluzione circolare più adatta alla FFT.

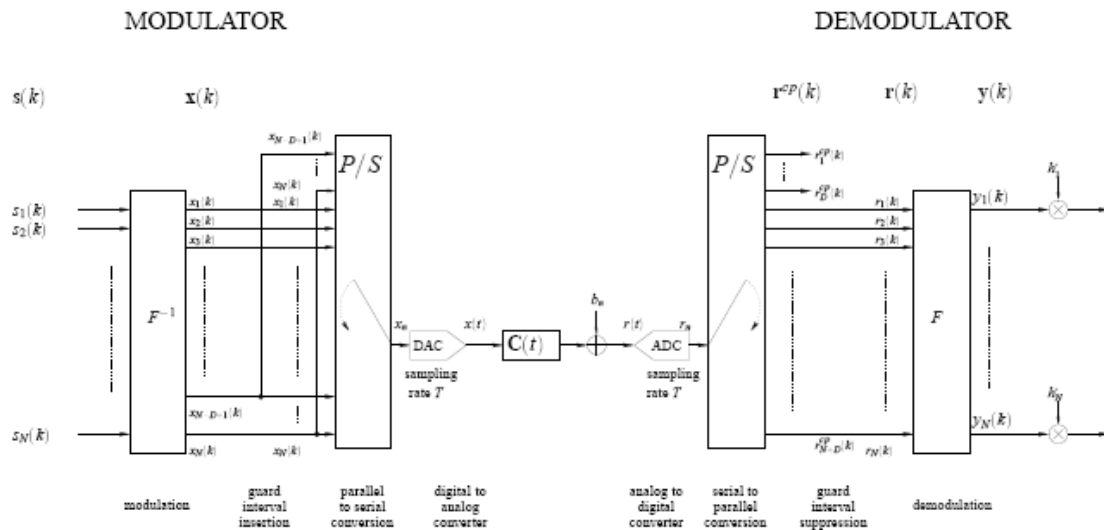


Fig. 9 – OFDM model

In Figura 9 è riportato lo schema della modulazione OFDM, dal quale si possono dedurre le seguenti considerazioni:

- Conversione serie – parallelo: ogni N-esimo simbolo viene applicato ad un modulatore;
- Al ricevitore il segnale viene demultiplato nelle N sottoportanti e gli N segnali vengono demodulati.
- I segnali riportati in banda base vengono poi ricombinati mediante un convertitore parallelo – serie;
- Per tutta la durata di un periodo di simbolo di un convenzionale sistema seriale, ciascuno degli N sottocanali OFDM trasporta un solo simbolo.

Dopo aver svolto la IFFT, viene inserito un intervallo di guardia di D campioni all'inizio di ogni blocchetto:

$$[X_{N-D+1}(k), \dots, X_N(k), X_1(k), \dots, X_i(k), \dots, X_N(k)]$$

Esso consiste nell'estensione ciclica della rappresentazione temporale di un simbolo OFDM.

L'utilizzo del prefisso ciclico diminuisce la velocità di trasmissione e aumenta la banda necessaria: questo comporta lo svantaggio di avere minore efficienza, ma è di fondamentale importanza per la sincronizzazione al ricevitore dei sistemi di trasmissione.

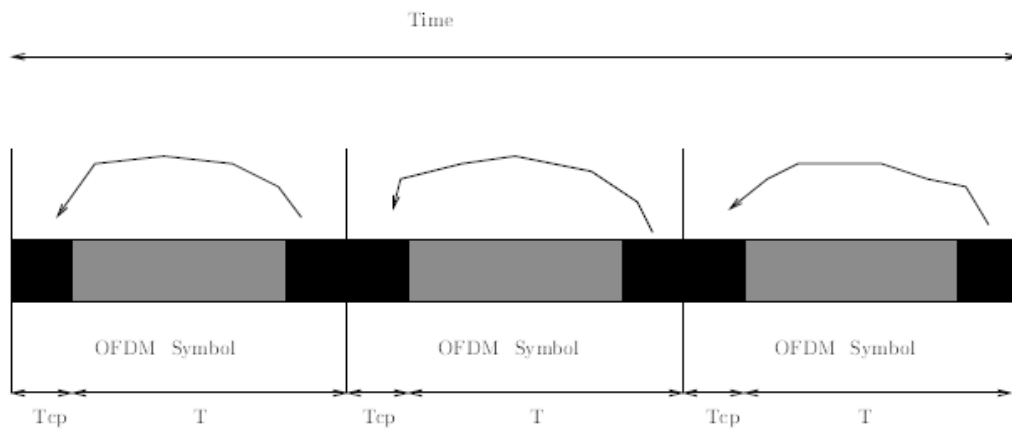


Fig. 10 – Rappresentazione temporale OFDM

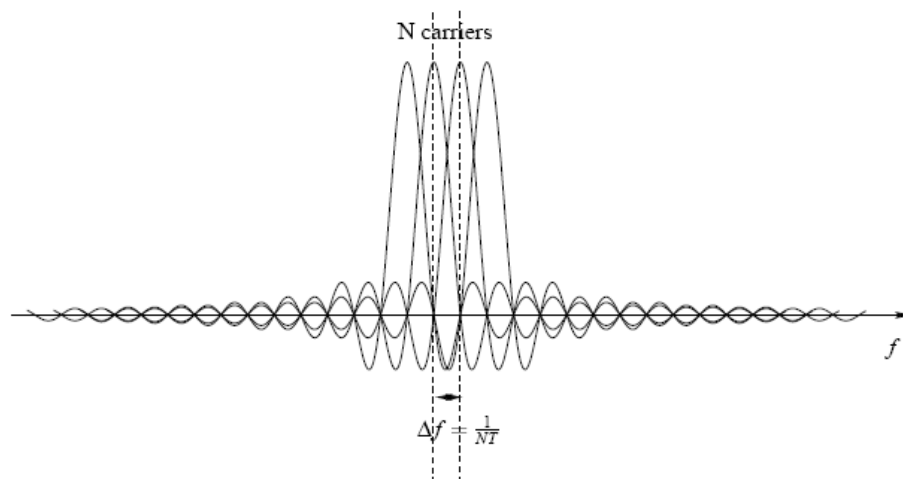


Fig. 11 – Rappresentazione frequenziale OFDM

Al ricevitore vengono svolte le funzioni inverse rispetto a quelle realizzate dal trasmettitore.

I segnali discreti ricevuti, con intervallo di guardia r^{CP} hanno la seguente forma:

$$r^{CP} = \begin{bmatrix} r^{CP}_1(k) \\ r^{CP}_2(k) \\ \vdots \\ r^{CP}_{N+D}(k) \end{bmatrix}_{(N+D) \times 1} = H_{ISI} \begin{bmatrix} x_{N-D+1}(k) \\ \vdots \\ x_N(k) \\ x_1(k) \\ \vdots \\ x_N(k) \end{bmatrix}_{(N+D) \times 1} + H_{IBI} \begin{bmatrix} x_{N-D+1}(k-1) \\ \vdots \\ x_N(k-1) \\ x_1(k-1) \\ \vdots \\ x_N(k-1) \end{bmatrix}_{(N+D) \times 1}$$

Fig. 12 – Rappresentazione discreta del segnale ricevuto con intervallo di guardia r^{CP}

H_{ISI} rappresenta l'interferenza intersimbolo, generata dal comportamento del canale.

H_{IBI} invece tiene conto dell'interferenza tra blocchi.

Questa modalità di rappresentazione è utile per mettere in evidenza l'azione dei due tipi di interferenza.

3.4 Considerazioni finali

La tecnologia OFDM incontra vastissimi campi applicativi nelle telecomunicazioni moderne.

Alcuni esempi di particolare rilevanza possono essere:

- ADSL
- DVB-T
- WiFi

La tecnica OFDM tuttavia risente anche di due principali limiti:

- suscettibilità ad errori dovuti alla stabilità delle frequenze portanti, determinate sia dagli oscillatori locali che dagli shift Doppler (echi che provengono da oggetti in movimento);
- necessità di un elevato rapporto peak-to-average power ratio (PAPR), in quanto il valore istantaneo della potenza in trasmissione può variare in modo repentino, anche all'interno di uno stesso simbolo, determinando una riduzione dell'efficienza dell'amplificatore di potenza in trasmissione.

4 Sincronizzazione nello standard DVB-T2

4.1 Scopo

Il singolo decoder digitale terrestre deve essere in grado di gestire le informazioni che gli vengono trasmesse e analizzarle nel modo migliore. Per garantire ciò, al ricevitore sono necessarie alcune operazioni di sincronizzazione dei dati passati attraverso il mezzo trasmissivo. In particolare, la determinazione della dimensione della FFT (Fast Fourier Transform) e dell'intervallo di guardia introdotto sono fondamentali al fine di recuperare correttamente le informazioni inviate.

In questo capitolo verranno analizzate le tecniche principali utilizzate a questo scopo, e le problematiche che devono essere affrontare.

4.2 Determinazione dell'intervallo di guardia

Nello standard DVB-T2 i frame iniziano con un particolare simbolo detto P1, il quale differisce per dimensioni e scopi dagli altri simboli OFDM. Questi infatti è particolarmente robusto alle interferenze, grazie alla sua struttura caratteristica.

Il primo riferimento in tempo e frequenza è proprio fornito dall'identificazione al ricevitore del simbolo P1. Questo permette di rilevare la presenza di una trasmissione DVB-T2 e determinare ad esempio la dimensione della FFT.

Ottenuto ora un primo riferimento temporale e frequenziale, procedendo con l'analisi del frame incontriamo una sequenza di simboli OFDM col relativo intervallo di guardia. Come si vede in Figura 13, sono presenti uno o più simboli P2 a seconda della dimensione della FFT. Si tratta comunque di simboli OFDM, che portano però informazioni di tipo statico, come ad esempio la dimensione dell'intervallo di guardia.



Fig. 13 – Rappresentazione di un frame e del relativo intervallo di guardia

Proprio la determinazione di questo parametro ci permette di gestire fisicamente la sincronizzazione.

Le considerazioni iniziali atte a sviluppare un algoritmo che si prefigge di calcolare la dimensione dell'intervallo sono le seguenti:

- Dopo aver identificato il simbolo P1, si determina da quest'ultimo la dimensione della FFT (N_{FFT}). In base a questo risultato saranno possibili solo alcuni valori prefissati dell'intervallo di guardia, come si deduce dalla tabella in Figura 14. Alcune combinazioni FFT size e intervallo di guardia sono infatti non ammesse.
- Avendo dunque una limitata possibilità di scelta, al ricevitore non rimane che calcolare quale sia la soluzione usata.

FFT size	Guard interval fraction (Δ/T_u)						
	1/128	1/32	1/16	19/256	1/8	19/128	1/4
32K	256T	1024T	2048T	2432T	4096T	4864T	NA
16K	128T	512T	1024T	1216T	2048T	2432T	4096T
8K	64T	256T	512T	608T	1024T	1216T	2048T
4K	NA	128T	256T	NA	512T	NA	1024T
2K	NA	64T	128T	NA	256T	NA	512T
1K	NA	NA	64T	NA	128T	NA	256T

Fig. 14 – Durata dell'intervallo di guardia in funzione del periodo elementare T

Nei seguenti sottoparagrafi sono riportati alcuni algoritmi che vanno proprio a stimare il guard interval utilizzato.

4.2.1 Guard interval estimation not normalized

Nota la dimensione della FFT, l'idea è quella di confrontare l'intervallo di guardia successivo al simbolo P1, con i valori presenti a distanza $N_{FFT} + N_{GI}$, dove con N_{GI} si intende la dimensione dell'intervallo di guardia preso in considerazione: il valore che comporta una maggiore corrispondenza tra i possibili candidati sarà l'intervallo di guardia cercato.

In realtà il confronto viene realizzato tramite l'operazione di correlazione, riportata nella seguente formula:

$$\sum_{K=0}^{N_{GI}-1} r_K r_{K+N_{FFT}+N_{GI}}^*$$

Dove "K" rappresenta l'indice che spazia all'interno della grandezza dell'intervallo di guardia, "r" è il frame ricevuto, ed il simbolo "*" fa riferimento all'operazione di complesso coniugato.

Svolgendo il calcolo della sommatoria per tutti i possibili guard interval concorrenti, la correlazione che garantisce un risultato superiore rispetto a tutti gli altri intervalli è quella che rappresenta l'intervallo di guardia ritenuto corretto.

4.2.2 Guard interval estimation normalized

In realtà l'algoritmo descritto nel precedente punto non fornisce risultati molto significativi, infatti una correlazione svolta su un campione di valori più ampio, spesso fornisce un risultato maggiore del calcolo della correlazione svolta sul vero intervallo di guardia generato.

Per ovviare a questo problema, nella funzione `guard_interval_estimation_norma` viene introdotto un fattore di normalizzazione pari alla lunghezza dell'intervallo su cui si va a calcolare la correlazione come illustrato nella seguente formula:

$$\frac{1}{N_{GI}} \sum_{K=0}^{N_{GI}-1} r_K r_{K+N_{FFT}+N_{GI}}^*$$

I parametri sono gli stessi descritti nel punto precedente, ed anche in questo caso la scelta dell'intervallo di guardia stimato ricade in quello che garantisce una correlazione superiore rispetto agli altri.

4.2.3 Guard interval estimation adaptive threshold

L'idea risolutiva di questo algoritmo introduce un sensibile cambiamento rispetto alle due soluzioni presentate precedentemente. Infatti, il suo principio si basa su confronti tra correlazioni stimate e calcolate.

Si inizia calcolando la correlazione sul più piccolo intervallo di guardia ammesso e successivamente si stima il valore verosimile per gli altri intervalli possibili. Questa operazione è puramente matematica, e si tratta semplicemente di una moltiplicazione per il rapporto tra i due intervalli considerati. Si ottiene in questo modo una serie di valori associati ai possibili guard interval ammessi, di cui solo il più piccolo è realmente calcolato, mentre gli altri sono stimati in base al valore di quest'ultimo. Andando a calcolare ora la correlazione anche sul maggiore degli intervalli considerati (`Corr_max`), si può già fornire una stima del prefisso ciclico utilizzato: il guard interval cercato è quello associato al valore più vicino a "`Corr_max`" fra quelli precedentemente stimati.

Per facilitare la comprensione di quest'ultimo passaggio consideriamo l'analisi del caso FFT size = 1K. I possibili intervalli di guardia ammessi sono tre: 1/4, 1/8, 1/16. Si definisce la correlazione per il più piccolo dei tre (ovvero 1/16) "`Corr_min`", e successivamente si stimano i valori per le altre due possibilità ammesse. Nel caso preso in esame si tratta di due moltiplicazioni, che scalano il valore "`Corr_min`" rispettivamente per un fattore due e un fattore quattro, ovvero il rapporto tra gli intervalli considerati. Il calcolo di "`Corr_max`", ovvero della correlazione sul più lungo dei prefissi ciclici possibili (1/4) ci fornisce una stima diretta sul reale guard interval utilizzato: a questo punto andremo infatti ad utilizzare un criterio di minima distanza dal "`Corr_max`" appena calcolato, sui tre valori precedentemente determinati, ovvero "`Corr_min`" e le due stime da esso derivate ed associate agli intervalli 1/8 e 1/4.

4.2.4 Guard interval estimation bubble

Analogamente a quanto visto nel precedente algoritmo, anche nella versione `guard_interval_estimation_bubble` l'idea è quella di utilizzare sia correlazioni calcolate che stimate.

Questa variante si ripropone la stima del GI, realizzando confronti solamente tra intervalli di guardia adiacenti. In altre parole realizza una serie di "spareggi" tra il prefisso ciclico preso in considerazione, e quello ammissibile direttamente successivo (per ordine di grandezza).

Per semplicità limitiamo ora l'analisi al caso FFT size = 1K (per le altre dimensioni la funzione si comporta in modo speculare). Dalla tabella si nota che le possibili soluzioni sulla stima del prefisso sono solamente tre (1/4, 1/8, 1/16). L'idea è quella di calcolare la correlazione sui dati ricevuti considerando il più piccolo intervallo di guardia ammesso:

$$\sum_{K=0}^{N_{GI}-1} r_K r_{K+N_{FFT}+N_{GI}}^*$$

Nel caso $N_{FFT}=1K$ dunque si inizierà valutando la sommatoria sopra riportata con l'indice K che varia tra 0 e $N_{GI}=1024 * 1/16$. Col risultato appena ottenuto si può stimare un ipotetico valore per l'intervallo di guardia successivo (in ordine di grandezza). Nel caso specifico la correlazione valutata per un $GI=1/8$, è il doppio di quella calcolata nel punto precedente: si ottengono dunque due valori numerici, legati tra loro proporzionalmente. Si calcola ora la correlazione del più grande dei due intervalli considerati (1/8) e si va a confrontare con i due valori numerici determinati precedentemente, utilizzando il criterio di minima distanza. I possibili casi sono due:

- La correlazione calcolata per il più grande dei due intervalli è più vicina a quella calcolata del minore;
- La correlazione calcolata per il più grande dei due intervalli è più vicina a quella stimata del maggiore.

Nella prima delle due possibilità si può già definire il guard interval stimato, che in questo caso corrisponde al più piccolo possibile (1/16). Nel secondo caso invece non si può ancora definire con precisione quale sia il prefisso, ed è dunque necessario proseguire con l'analisi iterando le istruzioni viste fin qui per i successivi due valori plausibili del GI (1/4, 1/8). Continuando in questo modo per tutte le possibili dimensioni della FFT, e condizionando tramite il flag `dvbt2_flag`, si coprono tutti i possibili casi previsti dallo standard.

L'approccio dell'algoritmo, che lavora su confronti tra correlazioni calcolate e stimate, procedendo per dimensioni dell'intervallo di guardia via via superiori, garantisce un miglioramento dell'efficienza rispetto ad una semplice stima.

4.2.5 Guard interval estimation optimum

L'algoritmo preso in considerazione si prefigge di determinare una serie di intervalli all'interno dei quali deve ricadere il valore della correlazione calcolato.

La funzione si può suddividere principalmente in due passi:

- Determinazione delle regioni di decisione
- Stima dell'intervallo di guardia utilizzando gli intervalli precedentemente calcolati.

Si procede analizzando preventivamente i possibili intervalli di guardia ammissibili, in base alla dimensione della FFT. Per ognuno di questi, poi, è necessario realizzare una serie di simulazioni per stimare i possibili valori di correlazione associati.

Per ogni GI si svolgono una serie di iterazioni, generando molteplici volte un frame DVB-T2 col prefisso ciclico relativo. Su ognuno di questi frame viene poi calcolata la correlazione relativa all'intervallo di guardia più grande possibile.

Considerando il caso $N_{fft}=1K$, terminati i passaggi appena descritti si ottiene:

- Un vettore di valori di correlazione, in cui l'operazione è stata svolta sull'intervallo lungo $1/4$, nonostante il frame sia stato generato con $GI=1/16$
- Un vettore di valori di correlazione, in cui l'operazione è stata svolta sull'intervallo lungo $1/4$, nonostante il frame sia stato generato con $GI=1/8$
- Un vettore di valori di correlazione, in cui l'operazione è stata svolta sull'intervallo lungo $1/4$, e il frame generato proprio con questo prefisso ciclico.

Ogni vettore dunque contiene una serie di misure, che si distribuiranno attorno ad una determinata media con una certa varianza: si può dunque ipotizzare che si ripartiranno in modo gaussiano. Tramite l'ausilio di MATLAB e in particolare della funzione "hist", è stato possibile rappresentare a video la funzione densità di probabilità realizzata in modo campionario.

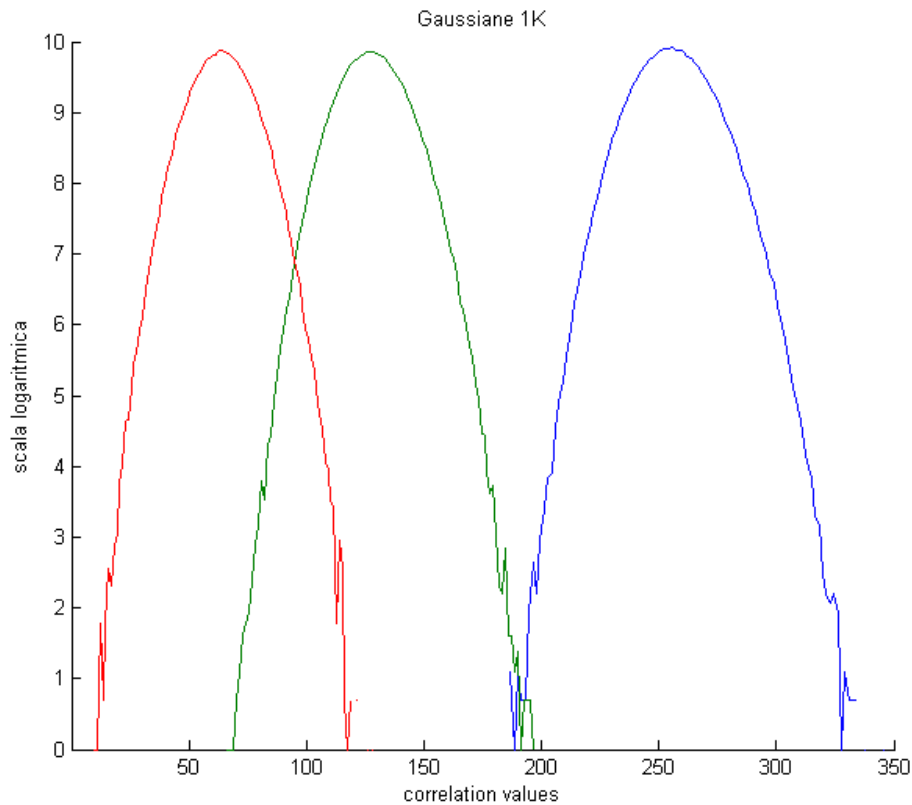


Fig. 15 – Determinazione delle regioni di decisione ottime per il caso $N_{\text{fft}}=1\text{K}$

Come si nota dall'esempio sopra riportato per il caso $N_{\text{fft}}=1\text{K}$, le regioni di decisione sono delimitate dal punto di incontro tra queste gaussiane. Risulta così facilmente realizzabile la suddivisione dell'asse contenente i valori di correlazione per i possibili GI ammessi.

Alla luce dei risultati ottenuti dall'analisi preventiva all'algoritmo, la stima del GI utilizzato in un generico caso risulta ora molto semplice. Calcolando la correlazione sul più grande dei prefissi ciclici ammesso, tramite una serie di confronti, si stabilisce entro quale intervallo ricade il valore appena calcolato: è proprio il prefisso associato a tale intervallo ad essere ritenuto la scelta più verosimile. Nella figura sottostante si può notare come sono state determinate le soglie per il caso $N_{\text{fft}}=8\text{K}$, per il quale sono ammissibili tutte le configurazioni del prefisso ciclico.

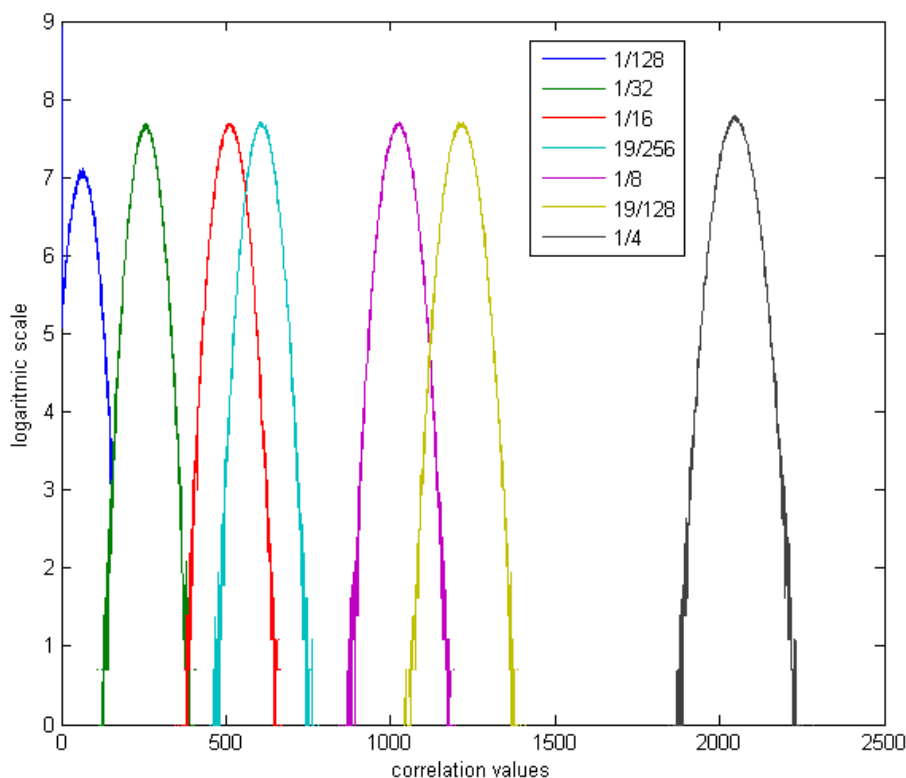


Fig. 16 – Determinazione delle regioni di decisione ottime per il caso $N_{\text{fft}}=8K$

4.3 Valutazioni e precisazioni

Nella realizzazione di questi algoritmi per la gestione della sincronizzazione al ricevitore, sono stati deliberatamente tralasciati alcuni importanti fattori. In questo paragrafo sono riportate alcune precisazioni.

4.3.1 Mezzo trasmissivo

Nel passaggio attraverso il mezzo trasmissivo, i dati sono soggetti all'azione del rumore. Proprio in base alla potenza di questo rumore, rappresentata con il parametro SNR (Signal-to-Noise Ratio), i dati trasmessi subiranno variazioni più o meno sentite. Nelle simulazioni svolte, si è considerato un canale con un rumore gaussiano bianco, che agiva sul frame trasmesso "corrompendo" i dati in base ad un SNR variabile.

Particolare attenzione va dedicata all'algoritmo `guard_interval_estimation_optimum`, in quanto la sua fase iniziale prevede la stima delle regioni di decisione ottime. I valori calcolati e usati successivamente nel codice della funzione sono fortemente dipendenti dal tipo di canale su cui si trasmette: nel caso in cui si volesse simulare la trasmissione in un mezzo trasmissivo differente, sarebbe necessario quindi rivalutare tutte le regioni ottime.

4.3.2 Contributo del simbolo P1 nell'analisi

Nelle considerazioni finora svolte, il contributo dovuto al simbolo P1 non è stato utilizzato pienamente. Si è infatti supposto che l'unico parametro ricavato dal P1 e utile nello sviluppo dell'algoritmo fosse la dimensione della FFT. In realtà, direttamente dal primo simbolo di un frame DVB-T2 è possibile estrapolare alcune informazioni riguardanti l'intervallo di guardia. In particolare dal P1 si può dedurre se il GI appartiene alla configurazione del precedente standard DVB-T (1/4, 1/8, 1/16, 1/32) oppure ricade in una delle nuove possibilità introdotte (1/128, 19/256, 19/128): il vantaggio di conoscere anticipatamente la classe di appartenenza dell'intervallo di guardia permette di poter escludere a priori il calcolo di alcune correlazioni che risulterebbe superfluo.

4.3.3 Considerazioni sulla struttura del frame OFDM

In Figura 13, come nelle valutazioni svolte finora, si è considerata la presenza di un solo simbolo P2 nel frame OFDM. Questo non è vero per tutte le possibili dimensioni della FFT: infatti, il numero dei simboli P2 presenti è prefissato, come evidenziato dalla seguente tabella.

FFT size	N_{P2}
1k	16
2k	8
4k	4
8k	2
16k	1
32k	1

Fig. 17 – Numero di simboli P2 in relazione alla dimensione della FFT

Tuttavia nelle simulazioni svolte questo fattore non è stato implementato, visto che si suppone di raggiungere una corretta sincronizzazione analizzando semplicemente il primo di questi simboli OFDM.

4.4 Simulazioni e risultati

Sono stati presentati finora i vari algoritmi che cercano di stimare l'intervallo di guardia realmente utilizzato: si consideri ora la simulazione completa di una trasmissione DVB-T2.

Il programma principale simula la generazione, la trasmissione e la ricezione di un frame DVB-T2. Si può dunque suddividere in tre sezioni:

- Il trasmettitore genera un frame DVB-T2 utilizzando una data dimensione della FFT e un prefisso ciclico predeterminato. In realtà la trama è sprovvista del contributo del simbolo P1, che si ritiene noto al ricevitore.
- La trasmissione avviene attraverso un canale sul quale agisce rumore gaussiano bianco.
- Il ricevitore ottiene in ingresso un frame parzialmente corrotto dal rumore e su di esso va a stimare l'intervallo di guardia ritenuto corretto tramite l'ausilio delle funzioni prima presentate.

Gli algoritmi presentati precedentemente sono funzioni utilizzate dal ricevitore, e quindi come parametri di ingresso ricevono:

- Il frame comprensivo del rumore.
- Il contributo dato dal simbolo P1 per mezzo di un flag (`dvbt2_flag`) settato in base alla configurazione del guard interval con cui è stato generato il frame.

Come report di uscita ognuno di questi algoritmi restituisce il prefisso ciclico ritenuto la migliore scelta secondo i vari criteri utilizzati.

Per poter valutare l'efficienza e confrontare le prestazioni dei vari algoritmi, è stato utilizzato il programma principale, che simula una trasmissione e la seguente ricezione, facendo variare i parametri in gioco. Si tratta dunque di una serie di simulazioni per determinate configurazioni $N_{\text{fft}}\text{-GI}$ al variare della potenza del rumore nel canale, nelle quali si confronta la probabilità di decisione corretta dei vari algoritmi. La stima di questa probabilità è realizzata verificando quante volte la funzione identifica correttamente il prefisso ciclico realmente utilizzato, per un considerevole numero di ripetizioni.

4.4.1 Un caso di particolare rilevanza: $N_{\text{fft}}=1\text{K}$

Tra le plausibili dimensioni della FFT particolare importanza assume il caso in cui essa sia la più piccola configurazione possibile.

Essendo la lunghezza del prefisso ciclico una frazione della dimensione dei simboli OFDM, la correlazione relativa al caso $N_{\text{fft}}=1\text{K}$ verrà calcolata su un numero inferiore di campioni: ciò comporta una maggiore esposizione alla corruzione da parte del rumore rispetto alle altre possibili dimensioni.

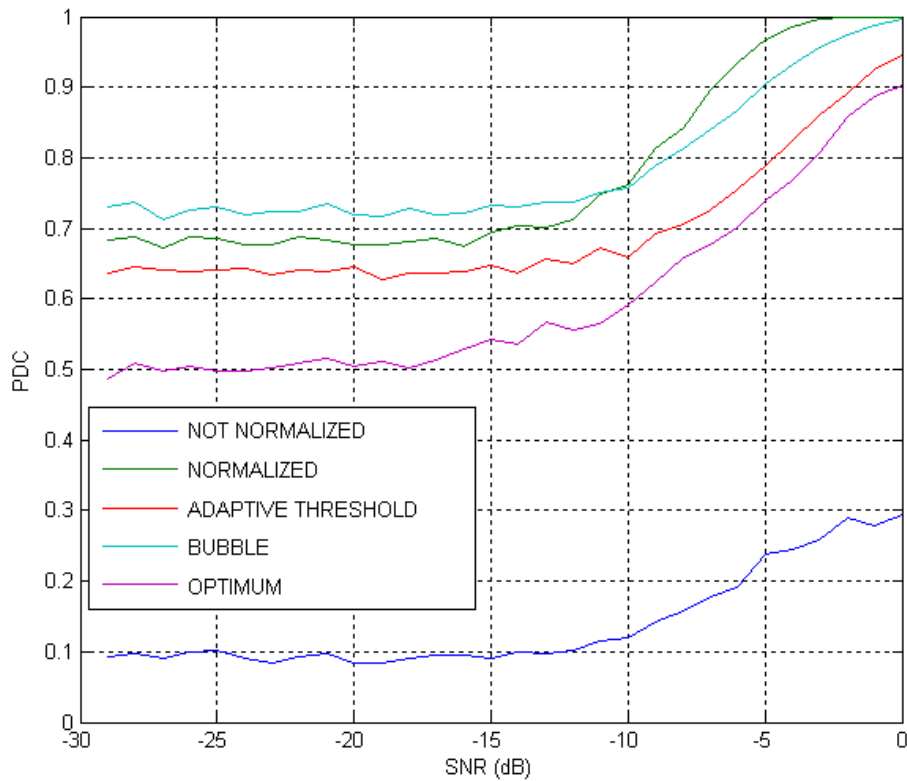


Fig. 18 – Simulazione con $N_{\text{fft}}=1\text{K}$ e $GI=1/16$

In Figura 18 è riportato l'andamento delle varie funzioni relazionando la potenza del rumore con la probabilità di decisione corretta. I frame DVB-T2 sono generati con il più piccolo prefisso ciclico possibile, ovvero $1/16$. L'asse delle ascisse, che rappresenta l'SNR, è espresso in decibel; l'intervallo comprende valori negativi, visto che le prestazioni dei principali algoritmi per intervalli positivi (spesso di interesse pratico) sono vicini all'ottimo.

Come si può notare dal grafico, le funzioni hanno comportamenti diversi; per valori di SNR prossimi allo zero *normalized* e *bubble* hanno una probabilità di decisione corretta unitaria, mentre *adaptive threshold* e *optimum* tendono comunque ad avere elevata efficienza. Il metodo *not normalized* invece non ottiene buoni risultati.

La motivazione di tale mancanza è da cercarsi nell'idea che sta alla base dell'algoritmo. Infatti, stimando semplicemente la correlazione sui tre intervalli possibili, è molto probabile che il più piccolo ($1/16$) garantisca un valore inferiore. Questo perché correlazioni calcolate su GI maggiori comprendono il contributo fornito da correlazioni calcolate su prefissi minori.

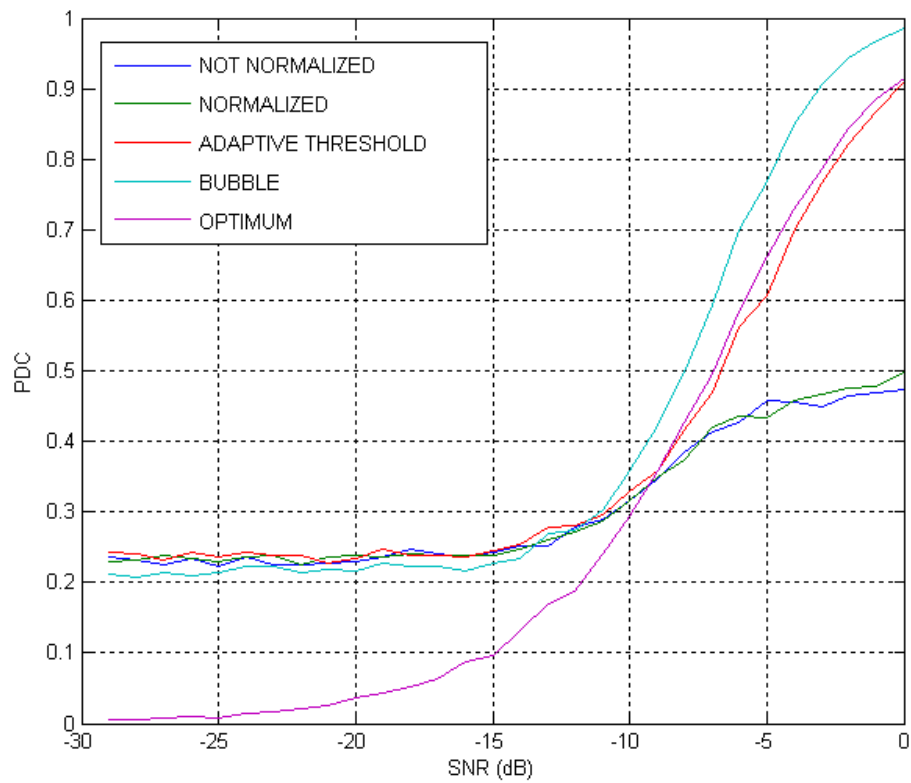


Fig. 19 – Simulazione con $N_{\text{fft}}=1\text{K}$ e $GI=1/8$

Proseguendo l'analisi, in Figura 19 è riportato l'esito della simulazione svolta generando i frame con un prefisso pari a $1/8$ della dimensione della FFT. Per questa configurazione il metodo che sembra funzionare in modo più efficiente è *bubble*. Anche *adaptive threshold* e *optimum* nella zona di interesse tendono rapidamente a ottimi risultati, mentre i metodi *normalized* e *not normalized* appaiono distanti da questo tipo di prestazioni.

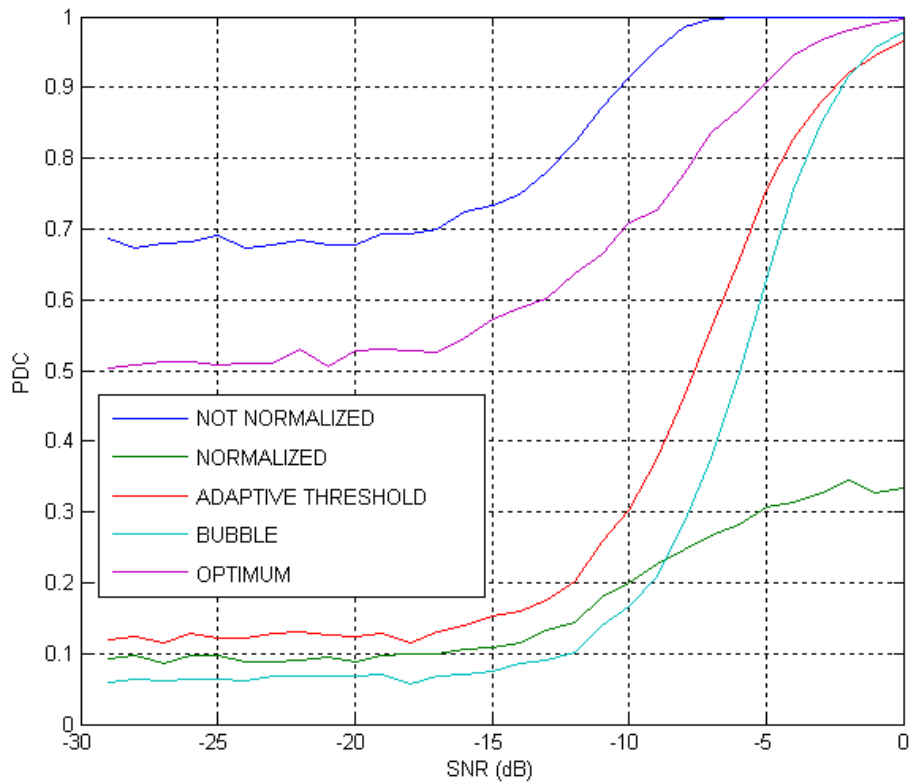


Fig. 20 – Simulazione con $N_{\text{fft}}=1\text{K}$ e $GI=1/4$

Considerando ora l'analisi dell'ultimo guard interval ammissibile, notiamo un miglioramento delle prestazioni da parte di quasi tutti i metodi. In realtà era un risultato piuttosto prevedibile, infatti, utilizzando il prefisso più lungo fra quelli ammissibili aumenta il numero di campioni sul quale viene calcolata la correlazione.

L'algoritmo che fornisce le migliori prestazioni è *not normalized*, che già per valori dell'SNR pari a -7 dB stima sempre l'intervallo di guardia in maniera corretta. In realtà questo metodo molto spesso predilige come scelta l'intervallo maggiore e dunque è spiegato questo ottimo risultato.

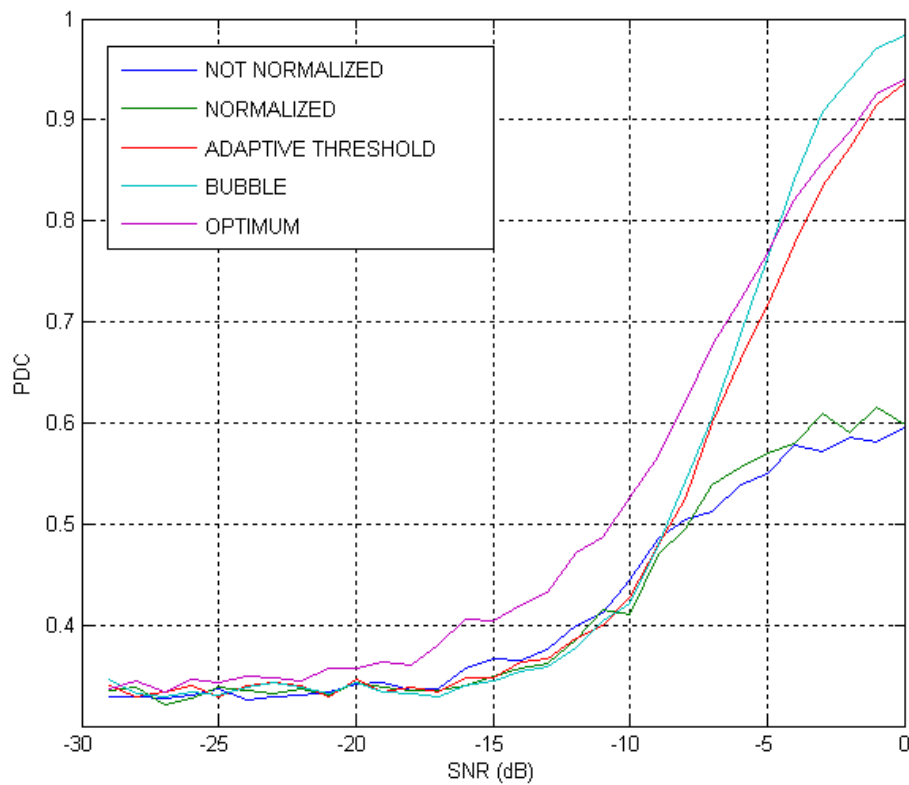


Fig. 21 – Simulazione con $N_{\text{fft}}=1\text{K}$ e GI random

Per avere una visione più generale del caso critico appena analizzato, in Figura 21 è riportato l'andamento delle performance dei vari algoritmi che agiscono su frame il cui prefisso ciclico è generato casualmente tra gli intervalli ammissibili. Si ha dunque una stima più attendibile della reale efficienza dei metodi presentati.

Bubble, con i suoi confronti a “spareggio” tra possibili GI adiacenti, garantisce le migliori prestazioni, soprattutto quando la potenza del rumore risulta minore di -5 dB.

Optimum assicura alte probabilità di decisione corretta anche in canali molto disturbati, anche se, nella zona di maggiore interesse il suo andamento non si distacca molto da quello di *adaptive threshold*.

4.4.2 Altre possibili configurazioni

Le configurazioni finora analizzate non sono le uniche concesse dallo standard DVB-T2, ma sono state presentate con maggiore attenzione essendo le più sensibili all'azione del rumore.

Tuttavia analizzare ogni possibile combinazione risulterebbe dispendioso e poco produttivo.

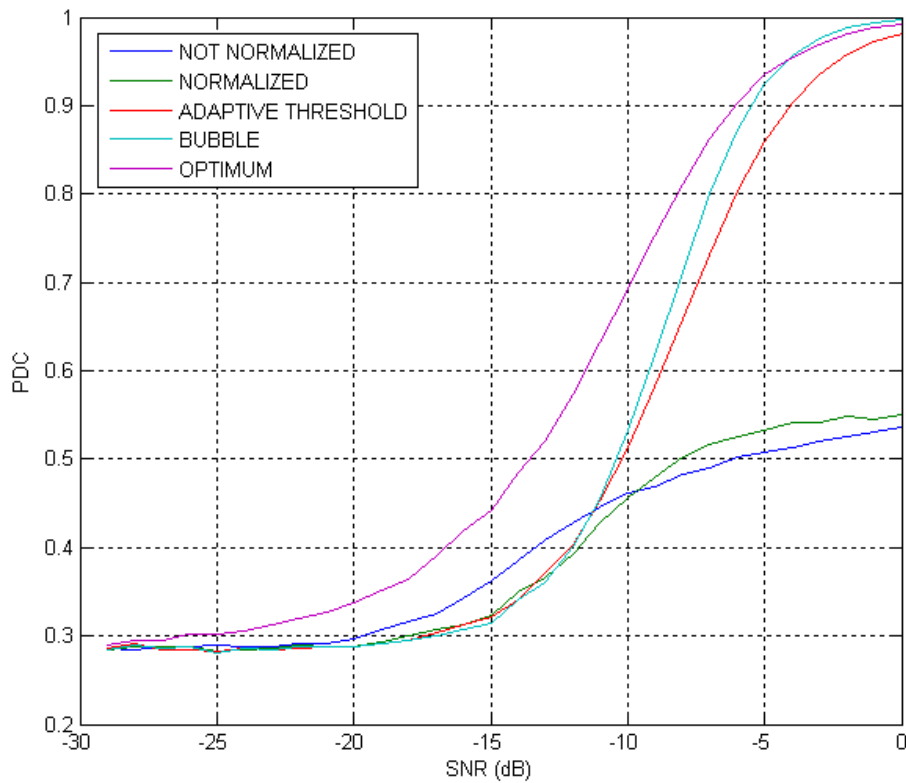


Fig. 22 – Simulazione con $N_{fft}=8K$ e GI random

Consideriamo il caso in cui la dimensione della FFT sia 8K. Generando il frame OFDM con un prefisso ciclico scelto casualmente, e analizzando le prestazioni dei vari algoritmi, otteniamo il grafico di Figura 22. Confrontando l'andamento dei vari metodi, si nota che questo non differisce molto dal caso $N_{fft}=1K$ e GI casuale, se non per un miglioramento delle prestazioni di tutte le funzioni per SNR bassi.

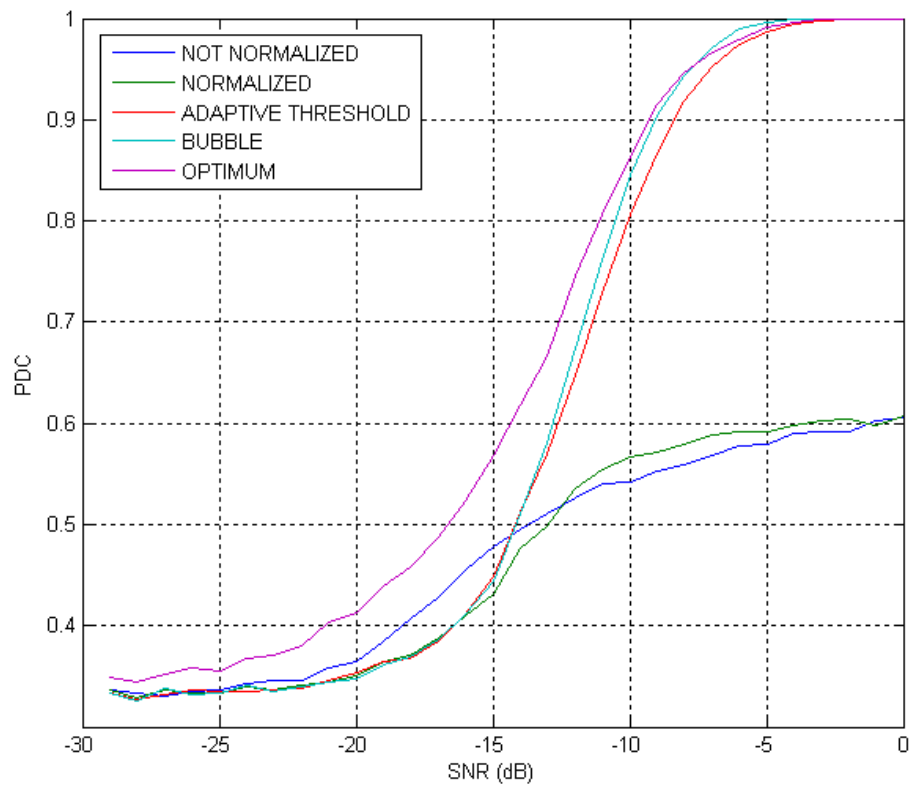


Fig. 23 – Simulazione con $N_{fft}=32K$ e GI random

La dimensione della Fast Fourier Transform, che meno risente dell'azione del rumore, è sicuramente 32K, ovvero la maggiore tra tutte quelle ammesse dallo standard. Come si nota dal relativo grafico, già per SNR attorno ai -3 dB i principali algoritmi identificano sempre il prefisso ciclico utilizzato. Questo notevole miglioramento in termini di prestazioni è garantito dall'elevato campione di valori su cui viene calcolata la correlazione.

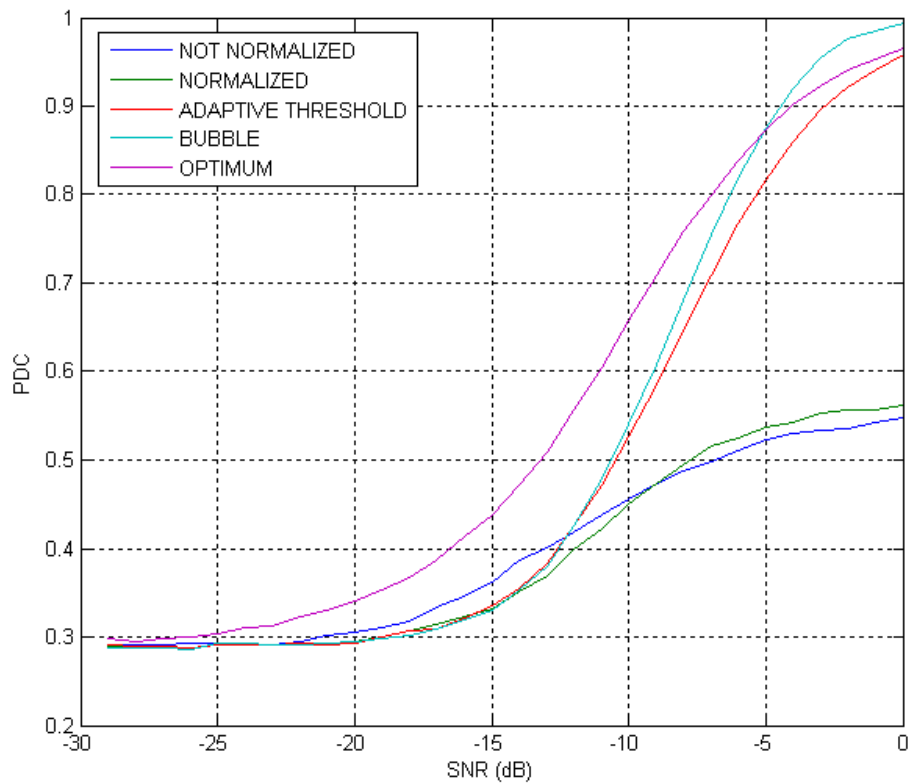


Fig. 24 – Simulazione con N_{fft} e GI random

Per avere un'analisi completa, senza riferirsi a particolari dimensioni dei simboli OFDM o guard interval generati, in Figura 24 è riportato il comportamento degli algoritmi nella più generica delle situazioni. Scelta casualmente la dimensione della FFT, i simboli OFDM costituenti il frame possiedono un prefisso ciclico generato anch'esso in maniera casuale tra tutti i possibili candidati.

Osservando il grafico e tenendo presente delle altre simulazioni fin qui analizzate, si può dire che la funzione *bubble* fornisce le migliori prestazioni.

Optimum tuttavia per canali molto disturbati garantisce performance superiori agli altri metodi.

In realtà nella regione di maggior interesse anche *adaptive threshold* non si discosta molto dall'andamento dei due appena citati.

5 Considerazioni finali e conclusioni

Realizzando la simulazione di una trasmissione DVB-T2, sono stati trascurati molteplici fattori che avrebbero reso più complicata l'analisi, distogliendo l'attenzione dai concetti fondamentali.

In particolare nell'analisi non sono stati considerati fenomeni quali:

- Interferenze intersimbolo.
- Canali sui quali agiscono rumori diversi.
- Interferenze intercampiono.
- Altri fattori caratterizzanti una trasmissione dati ma di rilevanza inferiore.

Anche nella generazione e ricezione di un frame sono state introdotte alcune semplificazioni, al fine di non appesantire troppo le simulazioni.

Inoltre il contributo del simbolo P1, con il quale inizia una trama DVB-T2, si suppone già noto al ricevitore. Questo permette di approfondire una serie di valutazioni:

- Le funzioni assumono come parametri noti la dimensione della FFT e il flag_dvbt2, e dunque questi termini non sono esposti a possibili errori.
- Alla luce di questa semplificazione risulta inutile generare e trasmettere fisicamente il P1, creando virtualmente un frame DVB-T2 che inizia direttamente con il simbolo P2.
- Il numero di simboli OFDM appartenenti al frame generato non serve sia elevato. Infatti, nella simulazione non ci si propone di scambiare dati, ma semplicemente di svolgere le operazioni di sincronizzazione.
- Non esistono ulteriori passi di sincronizzazione dopo quella realizzata dagli algoritmi sul primo simbolo OFDM ricevuto.

Spostando l'attenzione sulle prestazioni delle varie funzioni realizzate, si può notare come l'esito di una buona sincronizzazione dipenda fortemente dal tipo di configurazione utilizzata. Se un trasmettitore ad esempio genera i frame DVB-T2 utilizzando il prefisso ciclico maggiore possibile, l'algoritmo che meglio si appropria a questa soluzione è sicuramente *not normalized*. Viceversa, se prima di attraversare il mezzo trasmissivo i simboli OFDM sono generati con un intervallo di guardia piccolo, il metodo *normalized* garantisce prestazioni superiori. Non avendo alcuna conoscenza sulla dimensione del guard interval utilizzato invece, le funzioni *optimum*, *adaptive threshold* e *bubble* permettono un'efficienza di gran lunga superiore rispetto alle altre due soluzioni.

Osservando più nel dettaglio i risultati ottenuti, si nota come *bubble* sia preferibile nella maggior parte dei casi. Il suo funzionamento nella zona di interesse per lo studio di una trasmissione, lo rende principale candidato alla soluzione ottimale. Per SNR inferiori ai -5 dB, e quindi canali più disturbati, *optimum* garantisce performance superiori, e dunque una probabilità di sincronizzazione migliore. Ponendo l'attenzione sull'andamento degli algoritmi

nell'area di interesse più pratico, si nota come la differenza tra i tre migliori candidati non sia molto marcata. Anche adaptive threshold dunque non è una soluzione da scartare.

L'analisi fin qui svolta si è incentrata principalmente sulle prestazioni in termini di probabilità di decisione corretta, ma questa non è l'unico fattore di cui bisogna tener conto. Le complessità logiche dei vari algoritmi, e la loro realizzazione a livello circuitale infatti sono fattori fondamentali nella scelta della migliore tecnologia: sarà dunque indispensabile cercare un buon compromesso tra prestazioni e semplicità di implementazione.

6 Bibliografia

- [1] Digital Video Broadcasting, *Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)*., 2009.
- [2] Digital Video Broadcasting, *Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)*., 2008.
- [3] Pramod Viswanath David Tse, *Fundamentals of Wireless Communication.*: Cambridge, 2005.
- [4] N.Laurenti, N.Benvenuto, T.Erseghe, and R:Corvaja, *Communication System.*: Wiley, 2007.
- [5] Merouane Debbah. Short introduction to OFDM. [Online].
<http://www.supelec.fr/d2ri/flexibleradio/cours/ofdmtutorial.pdf>
- [6] L.Vangelista et al., "Key technologies for next-generation terrestrial digital television standard DVB-T2 ," *Communications Magazine, IEEE, Vol. 47*, 2009.
- [7] Li Zou, "Automatic Detection of the Guard Interval Length in OFDM System," *Journal of Communications, Vol 1* , 2006.
- [8] T.Yucek and H. Arslan, "OFDM Signal Identification and Transmission Parameter Estimation for Cognitive Radio Applications," *Global Telecommunications Conference*, 2007.
- [9] Centro Ricerche e Innovazione Tecnologica - RAI, "DVB-T2 la nuova piattaforma di diffusione della TV digitale terrestre," *Elettronica e Telecomunicazioni N°3*, 2008.
- [10] Charbel El Hajjar, "Synchronization Algorithms for OFDM Systems," *IEEE802.11a, DVB-T*, 2008.
- [11] [Online]. <http://www.dvb.org/>

7 Appendice

```

%Guard_interval_estimation_not_norma

%Determining the Guard Interval using Correlation

function GI_size = guard_interval_estimation_not_norma(r,dvbt2_flag)
global Receive_param

%Auxiliart variables

Ngi1=1024*Receive_param.FFT_size*1/4;
Ngi2=1024*Receive_param.FFT_size*1/8;
Ngi3=1024*Receive_param.FFT_size*1/16;
Ngi4=1024*Receive_param.FFT_size*1/32;
Ngi5=1024*Receive_param.FFT_size*1/128;
Ngi6=1024*Receive_param.FFT_size*19/256;
Ngi7=1024*Receive_param.FFT_size*19/128;

%Auxiliart vector to store the correlation
cor=[0 0 0 0 0 0 0];

%Possible correlation using the contribute of the flag dvbt2

if Receive_param.FFT_size==1

cor(1)=(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');
cor(2)=(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)');
cor(3)=(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)');
elseif (Receive_param.FFT_size==2) || (Receive_param.FFT_size==4)

cor(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');
cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)');
cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)');
cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)');

elseif (Receive_param.FFT_size==8) || (Receive_param.FFT_size==16)
    if dvbt2_flag==0

cor(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');
cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)');
cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)');

```

```

cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')
);
    else
cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')
);

cor(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi6)')
);

cor(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)')
);
    end

elseif Receive_param.FFT_size==32
    if dvbt2_flag==0

cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)')
);

cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)')
);

cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')
);
    else
cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')
);

cor(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi6)')
);

cor(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)')
);
    end
end

%The highest value of the vector gives the GI_size

k=max(cor);

if (k==cor(1))
    GI_size=1/4;
elseif (k==cor(2))
    GI_size=1/8;
elseif (k==cor(3))
    GI_size=1/16;
elseif (k==cor(4))
    GI_size=1/32;
elseif (k==cor(5))
    GI_size=1/128;
elseif (k==cor(6))
    GI_size=19/256;
elseif (k==cor(7))
    GI_size=19/128;
end

```

%Guard_interval_estimation_norma**%Determining the Guard Interval using Correlation**

```
function GI_size = guard_interval_estimation_norma(r,dvbt2_flag)
global Receive_param
```

%Auxiliar variables

```
Ngil=1024*Receive_param.FFT_size*1/4;
Ngi2=1024*Receive_param.FFT_size*1/8;
Ngi3=1024*Receive_param.FFT_size*1/16;
Ngi4=1024*Receive_param.FFT_size*1/32;
Ngi5=1024*Receive_param.FFT_size*1/128;
Ngi6=1024*Receive_param.FFT_size*19/256;
Ngi7=1024*Receive_param.FFT_size*19/128;
```

%Auxiliar vector to store the correlation

```
cor=[0 0 0 0 0 0 0];
```

%Possible correlation using the contribute of the flag dvbt2

```
if Receive_param.FFT_size==1
```

```
cor(1)=(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)')/Ngi1;
```

```
cor(2)=(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)')/Ngi2;
```

```
cor(3)=(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)')/Ngi3;
```

```
elseif (Receive_param.FFT_size==2) || (Receive_param.FFT_size==4)
```

```
cor(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)')/Ngi1;
```

```
cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)')/Ngi2;
```

```
cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)')/Ngi3;
```

```
cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')/Ngi4;
```

```
elseif (Receive_param.FFT_size==8) || (Receive_param.FFT_size==16)
```

```
    if dvbt2_flag==0
```

```
cor(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)')/Ngi1;
```

```
cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)')/Ngi2;
```

```
cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)')/Ngi3;
```

```
cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')/Ngi4;
```

```

else

cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')/Ngi5;
cor(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi6)')/Ngi6;
cor(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)')/Ngi7;

end

elseif Receive_param.FFT_size==32

if dvbt2_flag==0

cor(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)')/Ngi2;
cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)')/Ngi3;
cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')/Ngi4;
else

cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')/Ngi5;
cor(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi6)')/Ngi6;
cor(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)')/Ngi7;

end

end

%The highest value of the vector gives the GI_size

k=max(cor);

if (k==cor(1))
GI_size=1/4;
elseif (k==cor(2))
GI_size=1/8;
elseif (k==cor(3))
GI_size=1/16;
elseif (k==cor(4))
GI_size=1/32;
end

```

%Guard_interval_estimation_adaptive_threshold**%Determining the Guard Interval using Correlation**

```

function GI_size = guard_interval_estimation_adaptive_threshold(r,dvbt2_flag)
global Receive_param

%Auxiliart variables

Ngi1=1024*Receive_param.FFT_size*1/4;
Ngi2=1024*Receive_param.FFT_size*1/8;
Ngi3=1024*Receive_param.FFT_size*1/16;
Ngi4=1024*Receive_param.FFT_size*1/32;
Ngi5=1024*Receive_param.FFT_size*1/128;
Ngi6=1024*Receive_param.FFT_size*19/256;
Ngi7=1024*Receive_param.FFT_size*19/128;

%Auxiliart set to a symbolic value
cor=[1e5 1e5 1e5 1e5 1e5 1e5];

%Possible correlation using the contribute of the flag dvbt2

if Receive_param.FFT_size==1

k=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');

cor(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi3)');
cor(1)=abs(4*cor(3)-k);
cor(2)=abs(2*cor(3)-k);
cor(3)=abs(cor(3)-k);

elseif (Receive_param.FFT_size==2) || (Receive_param.FFT_size==4)

k=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');

cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)');
cor(1)=abs(8*cor(4)-k);
cor(2)=abs(4*cor(4)-k);
cor(3)=abs(2*cor(4)-k);
cor(4)=abs(cor(4)-k);

elseif (Receive_param.FFT_size==8) || (Receive_param.FFT_size==16)

    if dvbt2_flag==0

k=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1)');

cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)');
cor(1)=abs(8*cor(4)-k);
cor(2)=abs(4*cor(4)-k);
cor(3)=abs(2*cor(4)-k);
cor(4)=abs(cor(4)-k);

    else

```

```

k=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)');

cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')
);
    cor(6)=abs(19*cor(5)/2-k);
    cor(7)=abs(19*cor(5)-k);
    cor(5)=abs(cor(5)-k);

    end

elseif Receive_param.FFT_size==32

    if dvbt2_flag==0

k=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi2)');

cor(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi4)')
);
    cor(2)=abs(4*cor(4)-k);
    cor(3)=abs(2*cor(4)-k);
    cor(4)=abs(cor(4)-k);

    else

k=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)');

cor(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi5)')
);
    cor(6)=abs(19*cor(5)/2-k);
    cor(7)=abs(19*cor(5)-k);
    cor(5)=abs(cor(5)-k);

    end
end

%The lowest value of the vector gives the GI_size

k=min(cor());

if (k==cor(1))
    GI_size=1/4;
elseif (k==cor(2))
    GI_size=1/8;
elseif (k==cor(3))
    GI_size=1/16;
elseif (k==cor(4))
    GI_size=1/32;
elseif (k==cor(5))
    GI_size=1/128;
elseif (k==cor(6))
    GI_size=19/256;
elseif (k==cor(7))
    GI_size=19/128;
end

```

%Guard_interval_estimation_bubble**%Determining the Guard Interval using Correlation**

```
function GI_size = guard_interval_estimation_bubble(r,dvbt2_flag)
global Receive_param
```

%Auxiliart variables

```
Ngil=1024*Receive_param.FFT_size*1/4;
Ngi2=1024*Receive_param.FFT_size*1/8;
Ngi3=1024*Receive_param.FFT_size*1/16;
Ngi4=1024*Receive_param.FFT_size*1/32;
Ngi5=1024*Receive_param.FFT_size*1/128;
Ngi6=1024*Receive_param.FFT_size*19/256;
Ngi7=1024*Receive_param.FFT_size*19/128;
```

%Auxiliart vector to store the correlation estimate

```
cor_estimate=NaN*ones(1,7);
```

%Auxiliart vector to store the correlation calculate

```
cor_calc=NaN*ones(1,7);
```

%Possible correlation using the contribute of the flag dvbt2

```
if Receive_param.FFT_size==1
```

```
cor_calc(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi1)');

```

```
cor_calc(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi2)');

```

```
cor_calc(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi3)');

```

```
cor_estimate(2)=2*cor_calc(3);
```

```
cor_estimate(1)=4*cor_calc(3);
```

```
if ( abs(cor_calc(2)-cor_calc(3)) < abs(cor_estimate(2)-cor_calc(2)) )
```

```
GI_size=1/16;
```

```
elseif ( abs(cor_calc(1)-cor_calc(2)) < abs(cor_estimate(1)-cor_calc(1)) )
```

```
GI_size=1/8;
```

```
else
```

```
GI_size=1/4;
```

```
end
```

```
elseif (Receive_param.FFT_size==2) || (Receive_param.FFT_size==4)
```

```
cor_calc(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi1)');

```

```
cor_calc(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi2)');

```

```
cor_calc(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi3)');

```



```

cor_calc(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi4)');
    cor_estimate(3)=2*cor_calc(4);
    cor_estimate(2)=4*cor_calc(4);
    cor_estimate(1)=8*cor_calc(4);

    if ( abs(cor_calc(3)-cor_calc(4)) < abs(cor_estimate(3)-cor_calc(3)) )
        GI_size=1/32;
    elseif ( abs(cor_calc(2)-cor_calc(3)) < abs(cor_estimate(2)-cor_calc(2)) )
        GI_size=1/16;
    elseif ( abs(cor_calc(1)-cor_calc(2)) < abs(cor_estimate(1)-cor_calc(1)) )
        GI_size=1/8;
    else
        GI_size=1/4;
    End

elseif (Receive_param.FFT_size==8) || (Receive_param.FFT_size==16)

    if dvbt2_flag==0

cor_calc(1)=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi1)');
cor_calc(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi2)');
cor_calc(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi3)');
cor_calc(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi4)');

        cor_estimate(3)=2*cor_calc(4);
        cor_estimate(2)=4*cor_calc(4);
        cor_estimate(1)=8*cor_calc(4);

        if ( abs(cor_calc(3)-cor_calc(4)) < abs(cor_estimate(3)-cor_calc(3)) )
            GI_size=1/32;
        elseif ( abs(cor_calc(2)-cor_calc(3)) < abs(cor_estimate(2)-cor_calc(2)) )
            GI_size=1/16;
        elseif ( abs(cor_calc(1)-cor_calc(2)) < abs(cor_estimate(1)-cor_calc(1)) )
            GI_size=1/8;
        else
            GI_size=1/4;
        end
    else

cor_calc(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi5)');
cor_calc(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi6)');
cor_calc(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
gi7)');

        cor_estimate(6)=19*cor_calc(5)/2;
        cor_estimate(7)=19*cor_calc(5);

        if ( abs(cor_calc(6)-cor_calc(5)) < abs(cor_estimate(6)-cor_calc(6)) )
            GI_size=1/128;
        elseif ( abs(cor_calc(7)-cor_calc(6)) < abs(cor_estimate(7)-cor_calc(7)) )
            GI_size=19/256;
        else
            GI_size=19/128;
        end
    end

elseif Receive_param.FFT_size==32

```

```

    if dvbt2_flag==0

    cor_calc(2)=sum(r(1:Ngi2)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi2)');
    cor_calc(3)=sum(r(1:Ngi3)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi3)');
    cor_calc(4)=sum(r(1:Ngi4)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi4)');

        cor_estimate(3)=2*cor_calc(4);
        cor_estimate(2)=4*cor_calc(4);

        if (abs(cor_calc(3)-cor_calc(4))<abs(cor_estimate(3)-cor_calc(3)))
            GI_size=1/32;
        elseif (abs(cor_calc(2)-cor_calc(3))<abs(cor_estimate(2)-cor_calc(2)))
            GI_size=1/16;
        else
            GI_size=1/8;
        end
    else

    cor_calc(5)=sum(r(1:Ngi5)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi5)');
    cor_calc(6)=sum(r(1:Ngi6)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi6)');
    cor_calc(7)=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+N
    gi7)');

        cor_estimate(6)=19*cor_calc(5)/2;
        cor_estimate(7)=19*cor_calc(5);

        if (abs(cor_calc(6)-cor_calc(5))<abs(cor_estimate(6)-cor_calc(6)))
            GI_size=1/128;
        elseif (abs(cor_calc(7)-cor_calc(6))<abs(cor_estimate(7)-cor_calc(7)))
            GI_size=19/256;
        else
            GI_size=19/128;
        end
    end
end
end

```

```

%Test Program used to calculate the optimum decision regions for  $N_{fft}=1K$ 
clear

N_iter=10000;

%Vectors that store correlation values
vettcorr4=[];
vettcorr8=[];
vettcorr16=[];

global Receive_param

%Cicle where generate GI is ¼

for i=1:N_iter
    clear r;
    Receive_param.FFT_size=1;
    dvbt2_flag=0;
    GI=1/4;

    Ngi=GI*1024*Receive_param.FFT_size;

```

```

r=(randn(1,1024*Receive_param.FFT_size+(1024*Receive_param.FFT_size/2))+j*randn(1,1024*
Receive_param.FFT_size+(1024*Receive_param.FFT_size/2)))/sqrt(2);

    %Cyclic prefix

    r(1:Ngi)=r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi);
    Ngil=1024*Receive_param.FFT_size*1/4;

vettcorr4(i)=sum(r(1:Ngil)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+
Ngil)');

end

%Cicle where generate GI is 1/8

for i=1:N_iter
    clear r;
    Receive_param.FFT_size=1;
    dvbt2_flag=0;
    GI=1/8;

    Ngi=GI*1024*Receive_param.FFT_size;

r=(randn(1,1024*Receive_param.FFT_size+(1024*Receive_param.FFT_size/2))+j*randn(1,1024*
Receive_param.FFT_size+(1024*Receive_param.FFT_size/2)))/sqrt(2);

    %Cyclic prefix

    r(1:Ngi)=r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi);
    Ngil=1024*Receive_param.FFT_size*1/4;

vettcorr8(i)=sum(r(1:Ngil)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+
Ngil)');

end

%Cicle where generate GI is 1/16

for i=1:N_iter
    clear r;
    Receive_param.FFT_size=1;
    dvbt2_flag=0;
    GI=1/16;

    Ngi=GI*1024*Receive_param.FFT_size;

r=(randn(1,1024*Receive_param.FFT_size+(1024*Receive_param.FFT_size/2))+j*randn(1,1024*
Receive_param.FFT_size+(1024*Receive_param.FFT_size/2)))/sqrt(2);

    %Cyclic prefix

    r(1:Ngi)=r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi);
    Ngil=1024*Receive_param.FFT_size*1/4;

vettcorr16(i)=sum(r(1:Ngil)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size
+Ngil)');

end

[x16,n16]=hist(vettcorr16, 100);
[x8,n8]=hist(vettcorr8, 100);
[x4,n4]=hist(vettcorr4, 100);

```

```
x4=log(x4);
x8=log(x8);
x16=log(x16);
figure;
plot(n16,x16,n8,x8,n4,x4);
xlabel('correlation values');
ylabel('logarithmic scale');
legend('1/16','1/8','1/4');
```

%Guard_interval_estimation_optimum

%Determining the Guard Interval using Correlation

```
function GI_size = guard_interval_estimation_optimum(r,dvbt2_flag)
global Receive_param
```

%Auxiliart variables

```
Ngil=1024*Receive_param.FFT_size*1/4;
Ngi7=1024*Receive_param.FFT_size*19/128;
```

%Auxiliart variable used to store the correlation

```
cor_calc=sum(r(1:Ngi1)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi1
)');
```

%GI detection using the interval calculated before, and using the contribute of the flag dvbt2

```
if Receive_param.FFT_size==1

    if (cor_calc<=95)
        GI_size=1/16;
    elseif (cor_calc>95) & (cor_calc<=190)
        GI_size=1/8;
    else GI_size=1/4;
    end

elseif (Receive_param.FFT_size==2)

    if (cor_calc<=95)
        GI_size=1/32;
    elseif (cor_calc>95) & (cor_calc<=190)
        GI_size=1/16;
    elseif (cor_calc>190) & (cor_calc<380)
        GI_size=1/8;
    else GI_size=1/4;
    end

elseif (Receive_param.FFT_size==4)

    if (cor_calc<=190)
        GI_size=1/32;
    elseif (cor_calc>190) & (cor_calc<=380)
        GI_size=1/16;
    elseif (cor_calc>380) & (cor_calc<760)
        GI_size=1/8;
    else
        GI_size=1/4;
    end

end
```

```

elseif (Receive_param.FFT_size==8)

    if dvbt2_flag==0
        if (cor_calc<=380)
            GI_size=1/32;
        elseif (cor_calc>380)& (cor_calc<=765)
            GI_size=1/16;
        elseif (cor_calc>765)& (cor_calc<1530)
            GI_size=1/8;
        else GI_size=1/4;
        end
    else
        if (cor_calc(1)<=335)
            GI_size=1/128;
        elseif (cor_calc>335)& (cor_calc<=900)
            GI_size=19/256;
        else GI_size=19/128;
        end
    end

elseif (Receive_param.FFT_size==16)

    if dvbt2_flag==0
        if (cor_calc<=770)
            GI_size=1/32;
        elseif (cor_calc>770)& (cor_calc<=1530)
            GI_size=1/16;
        elseif (cor_calc>1530)& (cor_calc<3070)
            GI_size=1/8;
        else GI_size=1/4;
        end
    else
        if (cor_calc<=665)
            GI_size=1/128;
        elseif (cor_calc>665)& (cor_calc<=1818)
            GI_size=19/256;
        else GI_size=19/128;
        end
    end

cor_calc=sum(r(1:Ngi7)*r(1024*Receive_param.FFT_size+1:1024*Receive_param.FFT_size+Ngi7)');

elseif Receive_param.FFT_size==32

    if dvbt2_flag==0
        if (cor_calc<=1530)
            GI_size=1/32;
        elseif (cor_calc>1530)& (cor_calc<=3060)
            GI_size=1/16;
        else GI_size=1/8;
        end
    else
        if (cor_calc<=1325)
            GI_size=1/128;
        elseif (cor_calc>1325)& (cor_calc<=3635)
            GI_size=19/256;
        else GI_size=19/128;
        end
    end

end
end

```