

POLITECNICO DI TORINO

III Facoltà di Ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Tesi di Laurea

Ricevitori ottici ad altissimo bit-rate basati
sui algoritmi MLSE/BCJR



Relatori:

prof. Pierluigi Poggiolini

prof. Gabriella Bosco

prof. Vittorio Curri

Candidato:

Antonino Restifo

Luglio 2008

Ringraziamenti

Voglio ringraziare i miei relatori Prof.ssa Gabriella Bosco, Prof. Vittorio Curri e Prof. Pierluigi Poggiolini per avermi dato la possibilità di svolgere la tesi di laurea all'interno del loro gruppo. Specialmente voglio ringraziare la Prof.ssa Gabriella Bosco per tutta la sua disponibilità, pazienza e per i suoi utili suggerimenti lungo il lavoro di tesi.

Voglio anche ringraziare i miei genitori, le mie sorelle, i miei amici e tutti i membri della mia famiglia in Venezuela per il loro sostegno e incoraggiamento lungo questi due anni di studi all'estero, rendendo molto più facile il cammino verso questo traguardo.

Finalmente, voglio ringraziare tutti i miei parenti e amici in Italia per la loro continua disponibilità e collaborazione lungo questi due anni, facendomi sentire veramente a casa.

Indice

Ringraziamenti	iii
Elenco delle figure	ix
Acronimi	xiii
Introduzione	1
1. Formati di modulazione ottica	5
1.1 Modulazioni di ampiezza	6
1.1.1 IMDD	6
1.1.2 Duobinary	8
1.2 Modulazioni di fase	11
1.2.1 DPSK	12
1.2.2 DQPSK	14
2. Effetti della propagazione in fibra ottica	19
2.1 Effetti lineari	19
2.1.1 Attenuazione	20
2.1.2 Dispersione cromatica	21
2.1.3 Polarization Mode Dispersion (PMD)	23
2.1.3.1 State of Polarization (SOP)	24

2.1.3.2 Degree of Polarization (DOP)	25
2.2 Effetti non lineari	28
3. Maximum Likelihood Sequence Estimation (MLSE)	31
3.1 Decodifica a massima verosimiglianza	31
3.2 Stima delle metriche di ramo	34
3.2.1 Metrica Gaussiana	35
3.2.1 Metrica a radice quadrata	36
3.3 Struttura del sistema MLSE	37
3.4 Algoritmo di Viterbi	39
3.5 Caratteristiche del ricevitore MLSE per DQPSK	43
3.5.1 Balanced single-input double-MLSE processor	43
3.5.2 Balanced double-input MLSE processor	45
3.5.3 Unbalanced quad-input MLSE processor	47
4. Prestazioni della decodifica MLSE in presenza della PMD	49
4.1 Caratteristiche del sistema	50
4.2 Modulazione Duobinary	52
4.3 Modulazione IMDD	55
4.4 Modulazione DPSK	58
4.5 Modulazione DQPSK	61
4.6 Confronto fra i diversi formati	64
5. Prestazioni della decodifica MLSE basata sull'algoritmo BCJR in presenza della PMD	69
5.1 Algoritmo BCJR	70
5.2 Caratteristiche del sistema	75
5.3 Modulazione Duobinary	77
5.4 Modulazione IMDD	80
5.5 Modulazione DPSK	82
5.6 Modulazione DQPSK	85

6. Confronto fra le prestazioni degli algoritmi di Viterbi e BCJR con l'utilizzo dei codici FEC opportuni	89
6.1 Caratteristiche dei codici FEC	90
6.2 Codici FEC per il sistema sotto studio	92
6.3 Confronto fra Viterbi e BCJR	93
7. Prestazioni del ricevitore MLSE con un ridotto numero di campioni per bit	97
7.1 Caratteristiche del sistema	98
7.2 Confronto fra 2 campioni/bit e 4 campioni/bit	100
7.3 Confronto fra 1 campione/bit e 4 campioni/bit	102
7.4 Prestazioni del sistema in caso di forte presenza della PMD	106
Conclusioni	109
Referenze	113
Bibliografia	115
A. Codice MatLab dell'algoritmo BCJR per i formati di modulazione Duobinary, IMDD e DPSK	117
B. Codice MatLab dell'algoritmo BCJR per il formato di modulazione DQPSK	121

Elenco delle figure

1.1.	Trasmettitore IMDD con modulazione esterna	7
1.2.	Codifica RZ	7
1.3.	Codifica NRZ	7
1.4.	Struttura tipica del ricevitore basato su modulazione di ampiezza	8
1.5.	Modulatore di Mach-Zehnder	9
1.6.	Schema del sistema Duobinary	10
1.7.	Segnale Duobinary a tre livelli	10
1.8.	Segnale Duobinary a due livelli	11
1.9.	Segnale DPSK	12
1.10.	Trasmettitori DPSK	12
1.11.	Precodificatore DPSK	13
1.12.	Ricevitore DPSK	13
1.13.	Segnale DQPSK	14
1.14.	Trasmettitore DQPSK	15
1.15.	Precodificatore DQPSK	15
1.16.	Ricevitore DQPSK	16
2.1.	Attenuazione per una fibra singolo modo	21
2.2.	I due modi indipendenti E_x e E_y con disposizione ortogonale	24
2.3.	Rappresentazione del SOP s tramite la sfera di Poincaré	26
3.1.	Diagramma a traliccio con 4 stati e $M=2$	33

Elenco delle figure

3.2.	Schema del ricevitore MLSE	37
3.3.	Diagramma ad occhio del segnale ricevuto con rango di quantizzazione [-1,1]	38
3.4.	Approccio ACS dell'algoritmo di Viterbi	40
3.5.	Determinazione del cammino di metrica minore utilizzando l'algoritmo di Viterbi	41
3.6.	Schema del Balanced single-input double-MLSE processor	44
3.7.	Trellis a 16 stati per il Balanced single-input double-MLSE processor	44
3.8.	Schema del Balanced double-input MLSE processor	45
3.9.	Trellis a 16 stati per il Balanced double-input MLSE processor	46
3.10.	Schema del Unbalanced quad-input MLSE processor	47
4.1.	Rappresentazione della funzione $\log(-\log(\text{BER}))$ come una retta	52
4.2.	Schema del sistema per modulazione Duobinary	53
4.3.	DGD vs OSNR @ $\text{BER}=10^{-3}$ per modulazione Duobinary	54
4.4.	Confronto fra filtri ottimi e subottimi per Duobinary	55
4.5.	Schema del sistema per modulazione IMDD	56
4.6.	DGD vs OSNR @ $\text{BER}=10^{-3}$ per modulazione IMDD	56
4.7.	Confronto fra filtri ottimi e subottimi per IMDD	58
4.8.	Schema del sistema per modulazione DPSK	58
4.9.	DGD vs OSNR @ $\text{BER}=10^{-3}$ per modulazione DPSK	59
4.10.	Confronto fra filtri ottimi e subottimi per DPSK	61
4.11.	Schema del sistema per modulazione DQPSK	61
4.12.	DGD vs OSNR @ $\text{BER}=10^{-3}$ per modulazione DQPSK	62
4.13.	Confronto fra filtri ottimi e subottimi per DQPSK	64
4.14.	Confronto fra tutti i formati di modulazione senza MLSE	65
4.15.	Confronto fra tutti i formati di modulazione con MLSE	66
5.1.	Transizioni "in avanti" per gli $\alpha_j(s)$ e "all'indietro" per i $\beta_{j-1}(s')$	72
5.2.	Schema del sistema per modulazione Duobinary	78

Elenco delle figure

5.3.	Confronto fra OptSim e MatLab per modulazione Duobinary	78
5.4.	Confronto fra Viterbi e BCJR per modulazione Duobinary	79
5.5.	Schema del sistema per modulazione IMDD	80
5.6.	Confronto fra OptSim e MatLab per modulazione IMDD	81
5.7.	Confronto fra Viterbi e BCJR per modulazione IMDD	82
5.8.	Schema del sistema per modulazione DPSK	82
5.9.	Confronto fra OptSim e MatLab per modulazione DPSK	83
5.10.	Confronto fra Viterbi e BCJR per modulazione DPSK	84
5.11.	Schema del sistema per modulazione IMDD	85
5.12.	Confronto fra Viterbi e BCJR per modulazione IMDD	86
6.1.	DGD vs OSNR per modulazione Duobinary	93
6.2.	DGD vs OSNR per modulazione IMDD	94
6.3.	DGD vs OSNR per modulazione DPSK	94
6.4.	DGD vs OSNR per modulazione DQPSK	95
7.1.	Schema del sistema per modulazione Duobinary	99
7.2.	DGD vs OSNR per 4 camp/bit e 2 camp/bit variando l'istante di campionamento	101
7.3.	Istante di campionamento vs OSNR per diversi valori di DGD. 2 camp/bit	101
7.4.	DGD vs OSNR per 4 camp/bit e 1 camp/bit variando l'istante di campionamento	103
7.5.	Istante di campionamento vs OSNR per diversi valori di DGD. 1 camp/bit	104
7.6.	DGD vs OSNR per 4 camp/bit e 1 camp/bit ottimizzato	105
7.7.	DGD vs OSNR in caso di forte presenza della PMD	107

Acronimi

ACS:	Add, Compare and Select
ADC:	Analog-to-Digital Converter
AMZ:	Asymmetric Mach-Zehnder interferometer
ASE:	Amplified Spontaneous Emission
BCJR:	Bahl-Cocke-Jelinek-Raviv
BER:	Bit Error Probability
BPD:	Balanced Photo-Detector
DCF:	Dispersion Compensating Fiber
DGD:	Differential Group Delay
DOP:	Degree of Polarization
DPSK:	Differential Phase-Shift Keying
DQPSK:	Differential Quadrature Phase-Shift Keying
EDC:	Electronic Dispersion Compensation
FEC:	Forward Error Correction
FWM:	Four Wave Mixing
GCA:	Gain-Controlled Amplifier
IMDD:	Intensity Modulation Direct Detection
ISI:	Inter-Symbol Interference
Laser:	Light Amplification by Stimulated Emission of Radiation
LDPC:	Low-Density Parity-Check

MLSE:	Maximum Likelihood Sequence Estimation
MZ:	Mach-Zehnder
NRZ:	Non Return to Zero
OSNR:	Optical Signal-to-Noise Ratio
PDF:	Probability Density Function
PMD:	Polarization Mode Dispersion
PRBS:	Pseudo-Random Bit Sequence
PSP:	Principal States of Polarization
RZ:	Return to Zero
SOP:	State of Polarization
SPM:	Self Phase Modulation
WDM:	Wavelength Division Multiplexing
XPM:	Cross Phase Modulation

Introduzione

I sistemi in fibra ottica sono nati negli anni '70 e fin dall'inizio hanno affrontato e superato diversi ostacoli. Il primo di questi ostacoli fu l'elevata attenuazione (maggiore di 15 dB/km nelle prime fibre) che rendeva i collegamenti in fibra ottica non implementabili. E' stato poi dimostrato che l'attenuazione era causata dalle impurità e, una volta rimosse, si è arrivati a valori di attenuazione al di sotto dei 5 dB/km. Verso la fine degli anni '80, una volta risolto il problema dell'attenuazione, sono stati realizzati i primi collegamenti a lunga distanza. Per diminuire ancora le perdite nella fibra, si è passati da lavorare con la luce visibile a lavorare con la luce invisibile, nell'ordine degli infrarossi. In questo modo lo spettro è stato diviso in tre “finestre” particolarmente adatte all'uso nelle telecomunicazioni, con prestazioni e costi crescenti. In termini di lunghezza d'onda: *prima finestra* (850 nm, attenuazione di circa 1 dB/km), *seconda finestra* (1310 nm, attenuazione di circa 0.5 dB/km) e *terza finestra* (1550 nm, attenuazione di circa 0.2 dB/km).

Posteriormente, con l'incremento delle velocità di trasmissione (da 2.5 Gbps a 10 Gbps per esempio) sono emersi altri fenomeni, come la dispersione cromatica e gli effetti non lineari. Questi fenomeni si manifestano come interferenza intersimbolica (*Inter-Symbol Interference*, ISI), la quale deteriora notevolmente le prestazioni del sistema.

Incrementando le velocità di trasmissione dei sistemi ottici fino a 40 Gbps e 100 Gbps, entra in scena un nuovo ostacolo: il fenomeno della *Polarization Mode Dispersion* (PMD), la cui natura distorsiva degli impulsi che formano il segnale penalizza fortemente il budget di trasmissione, rendendo alcuni collegamenti non implementabili.

Per trovare una soluzione a tutti i fenomeni che generano ISI e che si presentano con l'incremento della velocità di trasmissione, sono state sviluppate delle tecniche che lavorano direttamente sul trasmettitore o sul ricevitore senza toccare il canale, quindi senza modificare la fibra ottica attualmente installata ed evitando i costi elevatissimi in termini di tempo e risorse che comporterebbero nuovi scavi per il cambio della fibra.

Una delle più importanti di queste tecniche è quella basata sulla *Electronic Dispersion Compensation* (EDC) nel ricevitore, cioè l'elaborazione del segnale nel dominio elettrico dopo essere stato rivelato dal fotodiodo. Fra le diverse tecnologie EDC quella più promettente è la *Maximum Likelihood Sequence Estimation* (MLSE), la quale lavora sulle sequenze di bit selezionando quella trasmessa più probabile (tramite l'algoritmo di *Viterbi*) partendo dai campioni ricevuti.

Questa tecnologia può essere implementata nei sistemi già installati e lavorare insieme con altre funzioni del ricevitore, come i codici *Forward Error Correction* (FEC). Lo scopo di questa tesi è quello di analizzare le prestazioni di diversi formati di modulazione assieme alla tecnologia MLSE nei sistemi ottici a 40 Gbps e in presenza della PMD.

In letteratura si trovano pochissimi articoli sulla decodifica MLSE sotto la presenza della PMD, per questo si è cercato di partire con le caratteristiche della tecnologia MLSE utilizzata per la compensazione della dispersione cromatica, per poi effettuare delle simulazioni sulla sua implementazione di questa tecnologia in sistemi con 100% di compensazione della dispersione ma in presenza della PMD. Posteriormente, con la finalità di migliorare ulteriormente le prestazioni

dell'MLSE standard, sono stati cercati degli algoritmi che lavorassero sul singolo bit anzich  sull'intera sequenza, a modo di avere in uscita l'affidabilit  di ogni bit per poi essere elaborata tramite codici FEC opportuni. E' stato scelto l'algoritmo *BCJR*, il quale ha le caratteristiche sopra citate, ed   stato scritto il codice che implementa questo algoritmo a modo di confrontare le sue prestazioni con quelle ottenute utilizzando l'algoritmo di Viterbi (MLSE standard).

Questa tesi   divisa in sette capitoli. Nel primo capitolo vengono presentati i formati di modulazione che verranno utilizzati: *Duobinary*, *IMDD*, *DPSK* e *DQPSK*. Nel secondo capitolo vengono descritti i diversi effetti presenti nella propagazione in fibra ottica. Nel terzo capitolo vengono presentate le caratteristiche e funzionamento della tecnologia MLSE. Nel quarto e quinto capitolo vengono illustrati i risultati di uno studio effettuato in un collegamento con 100% di compensazione della dispersione cromatica in presenza della PMD utilizzando la tecnologia MLSE e lavorando con gli algoritmi di Viterbi e BCJR rispettivamente. Nel sesto capitolo vengono confrontate le prestazioni di entrambi gli algoritmi utilizzando i codici FEC opportuni. Infine, nel settimo capitolo vengono analizzate le prestazioni della tecnologia MLSE riducendo il numero di campioni per bit presi in considerazione per l'elaborazione del segnale.

Capitolo 1

Formati di modulazione ottica

Per poter utilizzare la fibra ottica come mezzo per la trasmissione dell'informazione è necessario disporre di un segnale ottico (generalmente la potenza in uscita da un laser). La modulazione consiste nel convertire l'informazione digitale in un segnale in formato ottico che possa essere inviato lungo la fibra.

Esistono diversi formati di modulazione, particolarmente nel dominio ottico sono divisi in due categorie principali: modulazione diretta della sorgente ottica (in questa categoria esistono soltanto modulazioni di ampiezza) e modulazione esterna della sorgente ottica (in questa categoria oltre che modulazioni di ampiezza si possono trovare anche modulazioni di fase o polarizzazione).

La maggior parte dei sistemi di telecomunicazioni in fibra ottica lavorano con modulazione di ampiezza, la cui implementazione è abbastanza semplice ma ha dei limiti riguardanti la capacità dell'informazione. Con lo scopo di migliorare le prestazioni dei sistemi, sono stati adottati formati di modulazione alternativi. La modulazione di fase, ad esempio, consente di mantenere costante la potenza del

segnale, fatto positivo nella trasmissione in fibra ottica, ma ha bisogno di ricevitori coerenti molto complessi.

In questo capitolo verranno presentati quattro formati di modulazione diversi, i quali sono: *Duobinary*, *Intensity Modulation Direct Detection* (IMDD), *Differential Phase-Shift Keying* (DPSK) e *Differential Quadrature Phase-Shift Keying* (DQPSK). Questi formati vengono divisi in due categorie: *Modulazioni di ampiezza* e *Modulazioni di fase*.

1.1 Modulazioni di ampiezza

1.1.1 IMDD

Dagli inizi, i sistemi ottici si sono basati in questo formato trovandolo quello di più facile implementazione. L'informazione viene codificata nella intensità del segnale ottico: se nella finestra temporale corrispondente ad un bit si trova potenza, allora quel bit è un "1". Se non si trova potenza il bit si riconosce come uno "0".

Il modo più facile per implementare questo formato è accendendo e spegnendo la sorgente ottica, il che corrisponde ad una modulazione diretta del laser. Questa è stata la base della prima generazione di sistemi ottici.

Anni dopo, con lo scopo di migliorare le prestazioni dei sistemi è stata introdotta la modulazione esterna. In questo tipo di modulazione il laser viene alimentato a corrente costante dando origine ad una potenza ottica in uscita abbastanza stabile, la quale viene posteriormente sagomata con un modulatore esterno (generalmente di tipo Mach-Zehnder) come viene mostrato nella figura 1.1.

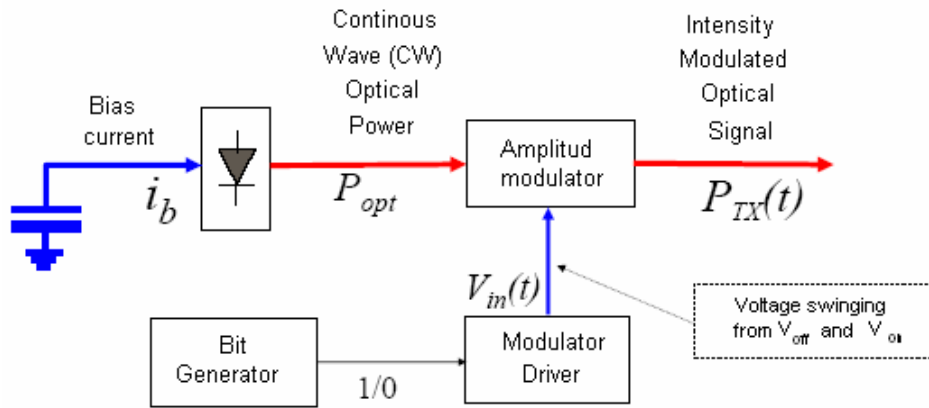


Figura 1.1. Trasmettitore IMDD con modulazione esterna

La modulazione di ampiezza può essere effettuata con due tipi diversi di codifica i quali si differenziano nel tipo di impulso utilizzato per la trasmissione. La codifica *Return to Zero* (RZ) utilizza un impulso che ha una durata minore a quella del bit, cioè ritorna a zero dentro della finestra temporale del bit, mentre la codifica *Non Return to Zero* (NRZ) utilizza impulsi che hanno la stessa durata della finestra temporale del bit. Nelle figure 1.2 e 1.3 si visualizzano le sequenze corrispondenti ad ogni tipo di codifica.

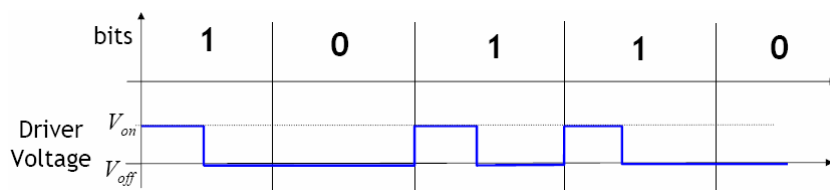


Figura 1.2. Codifica RZ

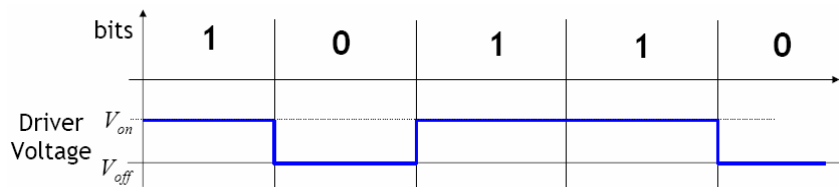


Figura 1.3. Codifica NRZ

La codifica NRZ ha il vantaggio di essere facilmente implementabile e di avere una occupazione spettrale ridotta, favorendo così la presenza di più canali e diventando lo standard nelle comunicazioni ottiche. Dall'altra parte ha lo svantaggio di essere molto sensibile agli effetti non lineari presenti nella fibra ottica. Questi effetti verranno discussi nel capitolo 2.

La codifica RZ presenta le caratteristiche opposte: ha il vantaggio di essere meno sensibile alle non linearità presenti nella fibra ottica, ma dall'altra parte ha lo svantaggio di una implementazione più complessa e di avere una occupazione spettrale maggiore.

Per tutti i formati basati su modulazione di ampiezza la struttura del ricevitore è come quella mostrata nella figura 1.4.

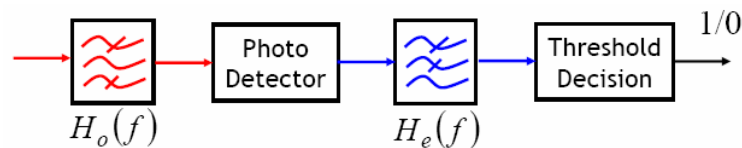


Figura 1.4. Struttura tipica del ricevitore basato su modulazione di ampiezza

1.1.2 Duobinary

Il formato di modulazione Duobinary è uno dei cosiddetti sistemi a “risposta parziale” nei quali la sequenza di bit viene manipolata in diverse maniere prima di entrare nella fibra. In questo modo il segnale in un determinato istante, oltre che dipendere dal bit corrente, dipende anche da uno o più dei bit precedenti. Questo formato di modulazione ha una efficienza spettrale molto buona, infatti si riesce a trasmettere ad un bit-rate pari a R bit/s con una larghezza di banda che sta sotto i R Hz. Il sistema con modulazione Duobinary può anche essere visto come un sistema con modulazione DPSK nel quale una certa quantità di “interferenza” o “correlazione” nei bit viene introdotta al trasmettitore. Il sistema DPSK verrà discusso nella sezione 1.2.1.

In questo modo, il sistema Duobinary avrà impulsi con una certa quantità di *Inter-Symbol Interference* (ISI), anche se questa ISI viene controllata a modo di poter essere tolta al ricevitore per trovare il segnale originale.

Il modulatore più usato è quello di *Mach-Zehnder* (MZ) a *Niobato di Litio* (LiNbO_3). La struttura base di questo modulatore viene mostrata nella figura 1.5. Comprende due guide d'onda, due giunzioni a *Y* e un elettrodo RF/DC. Il segnale ottico che entra al modulatore attraversa la prima giunzione a *Y*, nella quale viene diviso a metà e direzionato nelle due guide d'onda. Quando non si applica nessuna tensione all'elettrodo, i due segnali vengono ricombinati nella seconda giunzione a *Y* ed il segnale in uscita viene riconosciuto come un "1". Quando si applica un determinato valore di tensione all'elettrodo, l'indice di rifrazione risulta modificato a causa di effetti elettro-ottici, avendo come conseguenza un avanzamento nella fase del segnale in uno dei rami ed un ritardo nell'altro. Nel momento in cui i segnali vengono ricombinati nella seconda giunzione a *Y*, questi vengono cancellati ed il segnale in uscita viene riconosciuto come uno "0". La differenza di tensione fra quella che induce un "1" e quella che induce uno "0" viene chiamata *tensione di controllo* (driving voltage), ed è un parametro importante al momento di progettare il modulatore.

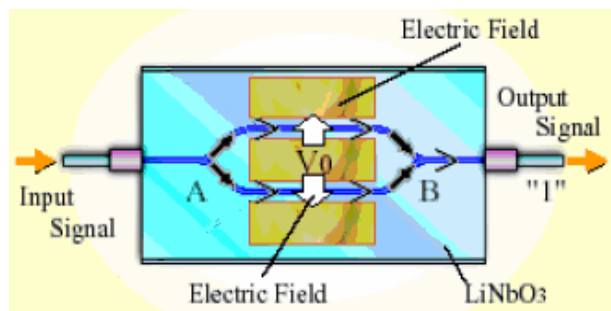


Figura 1.5. Modulatore di Mach-Zehnder

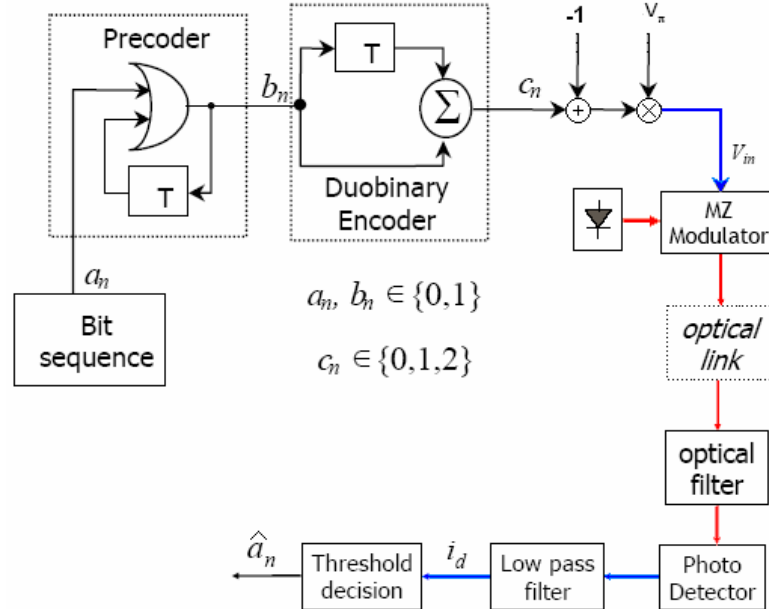


Figura 1.6. Schema del sistema Duobinary

Lo schema del sistema Duobinary viene mostrato nella figura 1.6. La sequenza di bit passa attraverso il *precodificatore differenziale*, il quale consiste nella operazione logica XOR fra il bit corrente ed il bit precedente in uscita dal precodificatore. Posteriormente il segnale entra al *codificatore* dove si effettua la somma fra il bit corrente e quello precedente, generando un segnale a tre livelli “0”, “1” e “2”. Dopo il codificatore il segnale viene traslato aggiungendo un “-1” per avere simmetria rispetto al livello “0”. Il risultato è il segnale mostrato nella figura 1.7. A questo punto il segnale viene moltiplicato per la tensione di controllo (V_π) e passa a controllare il modulatore di Mach-Zehnder.

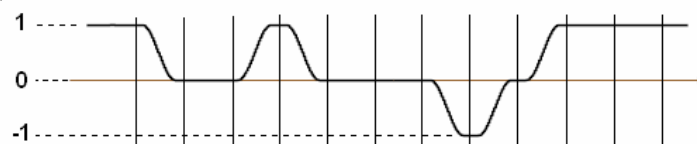


Figura 1.7. Segnale Duobinary a tre livelli

Il ricevitore Duobinary è costituito da un singolo fotodiodo convenzionale il quale rivela la potenza del segnale, così facendo si perde l'informazione relativa alla fase del segnale e in uscita si ha un segnale a due livelli come quello mostrato nella figura 1.8.

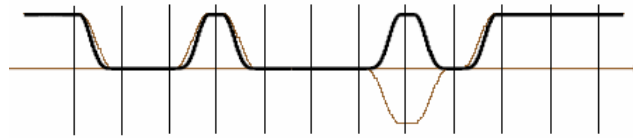


Figura 1.8. Segnale Duobinary a due livelli

1.2 Modulazioni di fase

Nelle comunicazioni ottiche l'uso della modulazione di fase si è reso molto interessante. Ci sono due alternative per rappresentare l'informazione nella fase del segnale: rivelando la fase del segnale stesso (in cui si ha bisogno di un demodulatore coerente che abbia un segnale di riferimento a modo di poter comparare i due valori delle fasi) o rivelando il *cambiamento* nella fase del segnale (questi schemi vengono chiamati *differenziali*).

Nel primo caso si presenta una ambiguità nella fase se per qualche motivo la costellazione presenta una rotazione. Questo problema può essere risolto utilizzando l'informazione per cambiare la fase del segnale anziché impostarla, cioè adottando un formato di modulazione differenziale.

Verranno presentati due formati di modulazione differenziale: il *Differential Phase-Shift Keying* (DPSK) ed il *Differential Quadrature Phase-Shift Keying* (DQPSK).

1.2.1 DPSK

In questo formato di modulazione il livello “1” si trasmette aggiungendo π rad al valore di fase corrente ed il livello “0” si trasmette aggiungendo 0 rad a questa fase. Il segnale modulato viene mostrato nella figura 1.9.

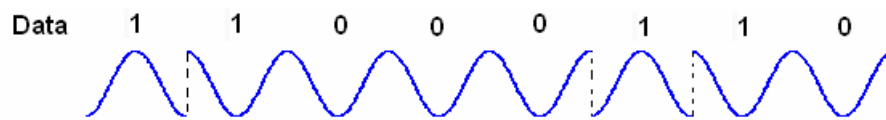


Figura 1.9. Segnale DPSK

Il trasmettitore DPSK può essere realizzato in due maniere diverse: possono essere utilizzati sia un modulatore di fase che un modulatore di ampiezza, come mostrato nella figura 1.10. Utilizzando un modulatore di fase l'informazione viene trasmessa nella fase del segnale, rimanendo inalterata l'ampiezza e controllando il modulatore con una tensione nell'intervallo $0/V_\pi$. Dall'altra parte, utilizzando un modulatore di ampiezza, questo viene controllato con una tensione nell'intervallo $-V_\pi/V_\pi$.

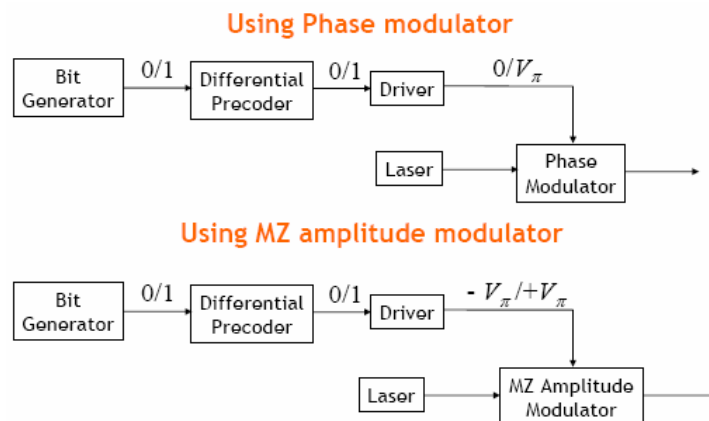


Figura 1.10. Trasmettitori DPSK

Nel precodificatore differenziale viene effettuata l'operazione logica XOR fra il bit corrente e l'uscita precedente del precodificatore. Il precodificatore viene mostrato nella figura 1.11.

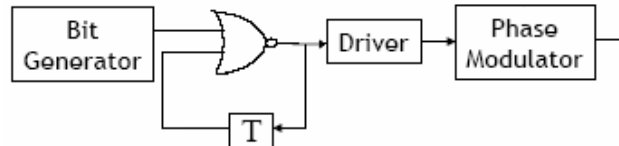


Figura 1.11. Precodificatore DPSK

Il ricevitore DPSK viene mostrato nella figura 1.12. Questo ricevitore consiste in un *Interferometro di Mach-Zehnder Asimmetrico*, il quale aggiunge in uno dei rami un ritardo pari ad un tempo di bit con lo scopo di comparare due bit successivi nello stesso istante, e un *fotorivelatore bilanciato*, il quale deve essere perfettamente bilanciato (avere la stessa responsività in entrambi i rami). Questo ricevitore presenta diversi parametri che lo rendono non ideale, i quali verranno presentati nella sezione 1.2.2. In linea di principio il segnale DPSK può essere rivelato con un singolo fotodiodo in ogni uscita dell'interferometro, ma siccome le uscite hanno segnali diversi i quali producono “occhi” diversi, allora il fotorivelatore bilanciato somma questi “occhi” rafforzandoli.

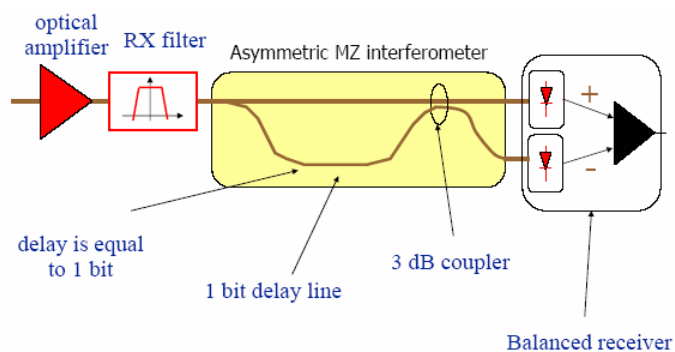


Figura 1.12. Ricevitore DPSK

Si deve notare che la modulazione DPSK riduce l’impatto degli effetti non lineari presenti nella fibra ed evita il problema della ricezione coerente, ma dall’altra parte ha bisogno di un precodificatore nel trasmettitore e di un interferometro asimmetrico ed un fotorivelatore bilanciato nel ricevitore, i quali possono essere molto costosi.

1.2.2 DQPSK

Il formato DQPSK è uno schema di modulazione differenziale a quattro livelli. Negli schemi di modulazione binari convenzionali, il trasmettitore deve emettere tanti impulsi al secondo (baud) quanti sono i bit per secondo che si desiderano trasmettere, ma nel caso dello schema DQPSK, essendo a quattro livelli, ogni impulso contiene l’informazione relativa a due bit. In questo modo per trasmettere ad un bit-rate pari a R bit/s basta che il trasmettitore emetta $R/2$ impulsi al secondo: questo dimezza l’occupazione spettrale e in alcuni casi migliora le prestazioni.

Il trasmettitore DQPSK genera un segnale ottico in cui l’informazione è contenuta nella differenza di fase ottica $\Delta\theta$ fra simboli successivi, $\Delta\theta$ può assumere uno dei 4 valori $[0, \pi/2, \pi, 3\pi/2]$. Ogni fase corrisponde ad un simbolo diverso, composto da 2 bit paralleli (chiamati rispettivamente bit “in fase” e bit “in quadratura”). Il segnale DQPSK viene mostrato nella figura 1.13.

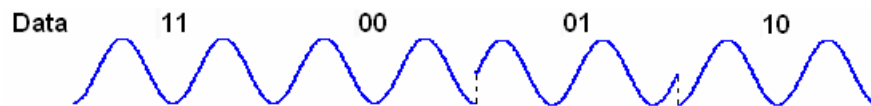


Figura 1.13. Segnale DQPSK

Il trasmettitore DQPSK viene mostrato nella figura 1.14. Dopo aver aggiunto uno sfasamento pari a $\pi/2$ nel ramo inferiore, i segnali sono combinati tramite un

accoppiatore ottico. In DQPSK, data la natura differenziale della codifica, una “precodifica” viene richiesta al trasmettitore per fare sì che i bit trasmessi coincidano con quelli demodulati. La struttura del precodificatore viene mostrata nella figura 1.15 e le sue relazioni ingresso/uscita sono:

$$p_k = \overline{(a_k \oplus b_k)} \cdot (a_k \oplus p_{k-1}) + (a_k \oplus b_k) \cdot (b_k \oplus q_{k-1})$$

$$q_k = \overline{(a_k \oplus b_k)} \cdot (b_k \oplus q_{k-1}) + (a_k \oplus b_k) \cdot (a_k \oplus p_{k-1})$$

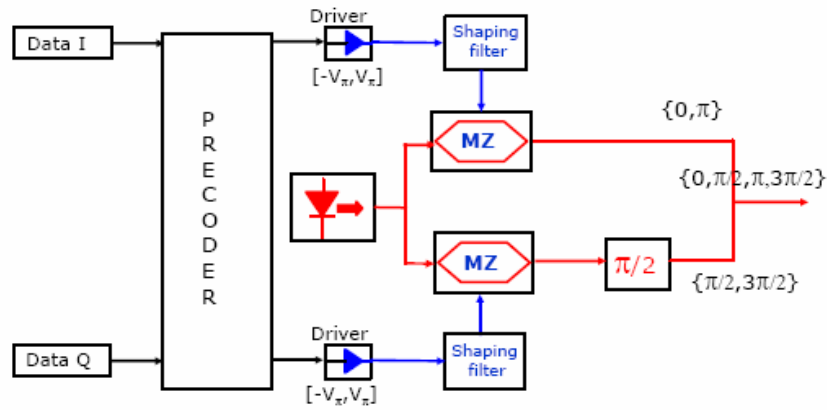


Figura 1.14. Trasmettitore DQPSK

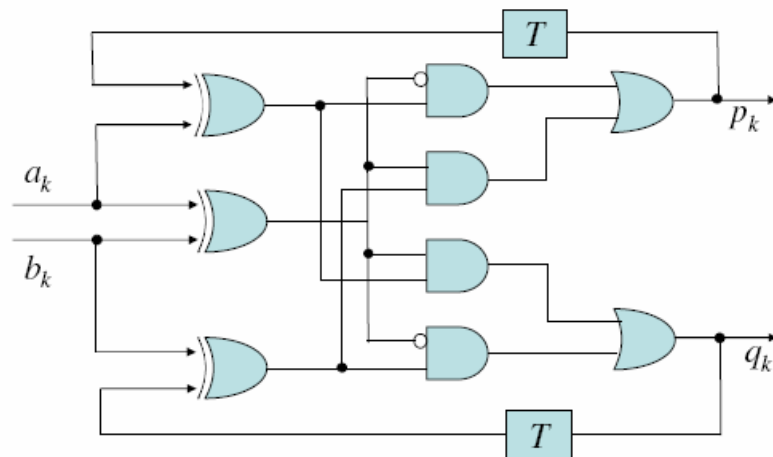


Figura 1.15. Precodificatore DQPSK

Il ricevitore DQPSK demodula il segnale ottico ricevuto generando due segnali elettrici paralleli $y^I(t)$ e $y^Q(t)$, una per ogni quadratura del segnale. Questo ricevitore, mostrato nella figura 1.16, è composto da un filtro ottico e due interferometri asimmetrici di Mach-Zehnder (AMZ) aventi una differenza di ritardo pari alla durata del singolo simbolo. La differenza di fase ottica fra i 2 rami dell'interferometro deve essere di $\pi/4$ e di $-\pi/4$ rispettivamente per l'AMZ superiore e quello inferiore. Dopo ogni AMZ si ha un fotorivelatore bilanciato (BPD, dall'inglese Balanced Photodetector), il quale consiste in 2 fotodiodi (uno per ogni uscita del AMZ) connessi in modo da sottrarre le loro correnti. Infine, per ogni ramo si ha un filtro elettrico di post-rivelazione.

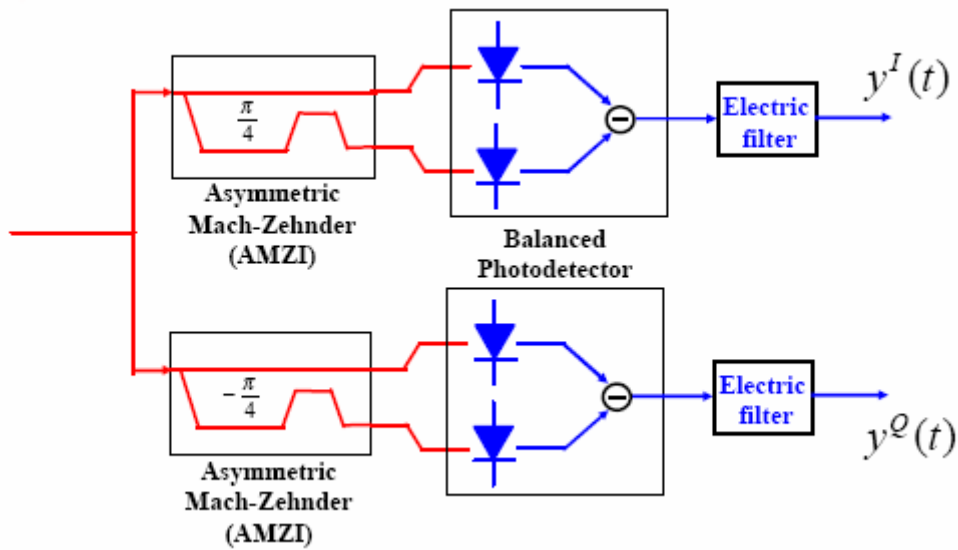


Figura 1.16. Ricevitore DQPSK

Se il campo elettrico all'uscita del filtro ottico di ricezione viene chiamato $E(t)$, i segnali in uscita dal AMZ superiore possono essere scritti come:

$$E_1(t) = \frac{1}{2} [E(t) + \gamma E(t - T_{AMZ}) e^{j\pi/4} e^{j\delta\phi}]$$

$$E_2(t) = \frac{1}{2} [E(t) - \gamma E(t - T_{AMZ}) e^{j\pi/4} e^{j\delta\phi}]$$

dove $\delta\Phi$ è un valore controllato tramite impostazioni termiche dell'AMZ. $\delta\Phi$ può essere trasformato in un parametro di sintonizzazione di frequenza Δf della funzione di trasferimento dell'AMZ tramite la relazione:

$$\Delta f = (\delta\Phi/4\pi) * B_R$$

dove B_R è il bit rate, il quale in DQPSK è pari al doppio del symbol rate S_R . Se $\Delta f = 0$ si ha una perfetta sintonizzazione dell'AMZ. L'errore di ritardo dell'AMZ viene definito come:

$$\delta T_{AMZ} = T_{AMZ} - T$$

dove T_{AMZ} è il ritardo introdotto dall'AMZ fra i suoi 2 rami e $T = 1/S_R$ è il tempo di simbolo. Per una ricezione ottima, si dovrebbe avere $T_{AMZ} = T$, risultando in $\delta T_{AMZ} = 0$. γ può solo assumere valori nell'intervallo $\{0,1\}$ ed è legata all'extinction ratio dell'AMZ ε tramite la relazione:

$$\varepsilon = (1 + \gamma)^2 / (1 - \gamma)^2.$$

Con questa definizione, ε può assumere valori nell'intervallo $\{1,\infty\}$. Fisicamente ε rappresenta il rapporto di potenza più elevato possibile che si potrebbe ottenere fra le 2 uscite dell'AMZ. Il segnale elettrico all'uscita del BPD può essere scritto come:

$$V(t) = \frac{1}{2} [R_1 |E_1(t - \tau_1)|^2 - R_2 |E_2(t - \tau_2)|^2]$$

dove R_1 e R_2 sono la responsivity di ogni ramo del BPD, mentre τ_1 e τ_2 sono i ritardi (ottici o elettrici) accumulati dentro la fase di sottrazione del BPD. Definendo il rapporto di ampiezza del BPD come:

$$K = (R_1 - R_2) / (R_1 + R_2)$$

questo parametro può assumere valori nell'intervallo $\{-1,1\}$ e gli estremi corrispondono al funzionamento di soltanto uno dei due rami del BPD, mentre $K = 0$ corrisponde ad un rapporto perfetto. Si definisce anche:

$$\delta T_{BPD} = \tau_1 - \tau_2$$

che rappresenta lo sbilanciamento del ritardo del segnale al momento della sottrazione delle correnti dei fotodiodi nel BPD. Questo sbilanciamento è tipicamente dovuto ad una differenza nelle lunghezze dei fili che connettono le uscite dell'AMZ con gli ingressi del BPD. Ovviamente, $\delta T_{BPD} = 0$ è il valore ottimo.

Nei capitoli successivi, si assume che tutti i parametri del ricevitore DQPSK siano perfettamente sintonizzati. Cioè $\Delta f = 0$, $\varepsilon = \infty$, $\delta T_{AMZ} = 0$, $K = 0$ e $\delta T_{BPD} = 0$.

Si deve notare che il formato DQPSK ha il vantaggio di avere una occupazione spettrale ridotta, diminuendo la sensibilità alla dispersione e agli effetti non lineari presenti nella fibra ottica, ma come controparte ha lo svantaggio di richiedere un precodificatore molto complesso nel trasmettitore, e anche due interferometri asimmetrici e due fotorivelatori bilanciati molto costosi nel ricevitore.

Capitolo 2

Effetti della propagazione in fibra ottica

In un sistema di telecomunicazioni in fibra ottica, mentre il segnale viaggia lungo la fibra, si manifestano effetti propagativi di natura lineare e non lineare. Gli effetti lineari possono essere modellati con una funzione di trasferimento, la quale è indipendente dal segnale che viene dato in ingresso alla fibra. Dall'altra parte gli effetti non lineari dipendono dal segnale in ingresso e quindi non possono essere modellati con una funzione di trasferimento.

2.1 Effetti lineari

Considerando la propagazione del campo ottico lungo una fibra singolo modo, e trascurando momentaneamente gli effetti non lineari, l'andamento della ampiezza del campo lungo la coordinata di propagazione z viene data dall'espressione:

$$\frac{\partial E}{\partial z} = -\alpha(\omega)E - j\beta(\omega)E$$

dove $\alpha(\omega)$ e $\beta(\omega)$ sono rispettivamente le costanti di attenuazione e propagazione della fibra. Questa equazione ha come soluzione analitica:

$$E(\omega, z) = E(\omega, 0)e^{-\alpha(\omega)z}e^{-j\beta(\omega)z} = E(\omega, 0)H_F(\omega, z)$$

dove $H_F(\omega)$ è la funzione di trasferimento della fibra ottica. Questa funzione di trasferimento dipende, oltre che dalla distanza di propagazione, dai due parametri $\alpha(\omega)$ e $\beta(\omega)$. Il parametro $\alpha(\omega)$ dà origine al fenomeno di *attenuazione* che verrà discusso nella sezione 2.1.1, mentre il parametro $\beta(\omega)$ dà origine al fenomeno di *dispersione cromatica* che verrà discusso nella sezione 2.1.2.

2.1.1 Attenuazione

L'attenuazione introdotta da una tratta di fibra ottica si definisce come il rapporto fra la potenza trasmessa (quella che si ha all'inizio della tratta) e la potenza ricevuta (quella che si ha alla fine della tratta). Quando un segnale viene propagato lungo la fibra ottica, parte della sua energia viene assorbita dal materiale, generando una perdita nella sua potenza.

L'attenuazione può essere causata da diversi fenomeni che dipendono dalle caratteristiche del materiale e si devono alla presenza di impurezze e imperfezioni. L'attenuazione può anche essere causata da possibili piegamenti o forzature applicate alla fibra ottica. Infine, l'attenuazione dipende fortemente dalla lunghezza d'onda e dal tipo di fibra che viene usato.

Nella figura 2.1 viene mostrata l'attenuazione in funzione della lunghezza d'onda, si può notare come esiste un minimo assoluto di 0.2 dB/km attorno alla lunghezza di 1550 nm e un minimo relativo di 0.4 dB/km alla lunghezza di 1300 nm. Queste lunghezze d'onda definiscono le cosiddette *finestre* in cui la fibra viene utilizzata.

In generale l'attenuazione può essere considerata costante alla frequenza utilizzata per la trasmissione e all'interno della banda occupata dal segnale, quindi si conclude che l'attenuazione è un effetto che non produce una distorsione del segnale.

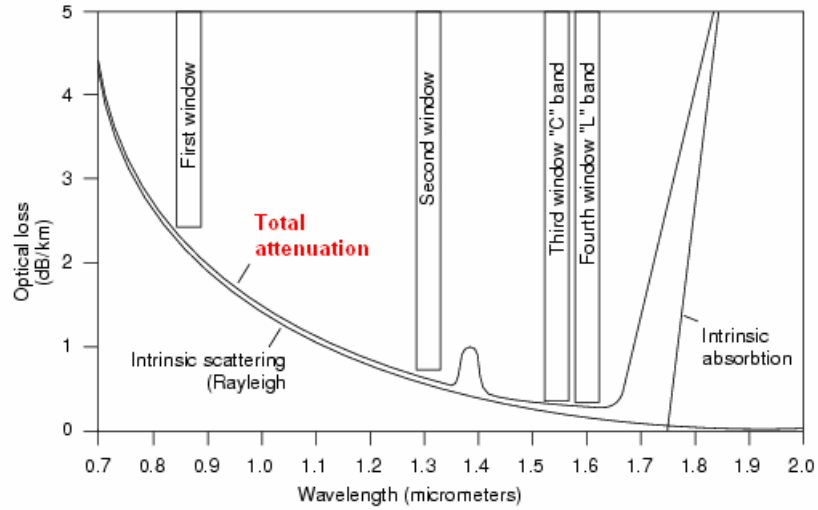


Figura 2.1 Attenuazione per una fibra singolo modo

2.1.2 Dispersione cromatica

La costante di propagazione $\beta(\omega)$ dipende dalla lunghezza d'onda, questo significa che le diverse componenti spettrali del segnale si propagheranno a velocità diverse. Come conseguenza il segnale ricevuto risulta distorto, dato che i diversi impulsi si “allargano” (aumentano la sua durata) ed interferiscono con gli impulsi adiacenti. Questo si traduce in una chiusura dell'occhio dovuta alla generazione di *Inter-Symbol Interference* (ISI), la quale deteriora le prestazioni del sistema. Con l'incremento del bit-rate, il *limite di dispersione* (lunghezza massima alla quale il sistema può lavorare) diminuisce. Dall'altra parte gli effetti della dispersione crescono con il quadrato del bit-rate.

Un approccio per studiare la dispersione cromatica consiste nello sviluppo in serie di Taylor della costante di propagazione attorno alla frequenza centrale, così facendo si ottiene:

$$\beta(\omega) \cong \beta_0 + \beta_1 \cdot (\omega - \omega_0) + \frac{\beta_2}{2} \cdot (\omega - \omega_0)^2 + \frac{\beta_3}{6} \cdot (\omega - \omega_0)^3 + \dots$$

dove

$$\beta_i = \left(\frac{\partial}{\partial \omega} \right)^i \beta(\omega) \Big|_{\omega=\omega_0}$$

Il termine $\beta_0 = \beta(\omega_0)$, ad una certa distanza z , determina una variazione di fase costante, la quale non distorce l'impulso.

Il termine β_1 introduce un ritardo τ_g noto come *ritardo di gruppo*, il quale è semplicemente il ritardo di propagazione degli impulsi:

$$\tau_g(\omega_0) = \beta_1 z = \frac{\partial \beta}{\partial \omega} \Big|_{\omega=\omega_0} z$$

Questo ritardo, proporzionale a z , è indipendente dalla frequenza, il che significa che anche β_1 non contribuisce a distorcere l'impulso.

Il termine β_2 genera un ritardo dipendente dalla frequenza ed è proporzionale a z . Considerando il suo contributo, il ritardo totale è:

$$\tau_g(\omega) = \tau_g(\omega_0) + \beta_2 \cdot (\omega - \omega_0) \cdot z$$

dove $\tau_g(\omega_0)$ rappresenta un ritardo medio di gruppo indipendente dalla frequenza. Dall'altra parte, il secondo termine originato da β_2 rappresenta un ritardo che prende un valore diverso per ogni componente spettrale dell'impulso trasmesso.

La costante β_2 può essere positiva o negativa. Se è positiva si è nel *regime dispersivo normale*, nel quale le componenti spettrali di frequenza più elevata si propagheranno ad una velocità minore rispetto alle componenti spettrali di frequenza più bassa. Se invece la costante β_2 è negativa si è nel *regime dispersivo anormale*, nel quale le componenti spettrali di frequenza più elevata si propagheranno più velocemente rispetto alle componenti spettrali di frequenza più bassa.

Generalmente la dispersione nella fibra ottica viene espressa in termini del parametro D , che è legato a β_2 tramite la relazione:

$$D = \beta_2 \cdot \left(-\frac{2\pi c}{\lambda^2} \right) = -0.785 \cdot \beta_2 \left[\frac{ps}{nm \cdot Km} \right]$$

Per esempio, alla lunghezza d'onda di 1550 nm le fibre standard singolo modo hanno $\beta_2 \cong -20 \text{ ps}^2/\text{km}$ e $D \cong 16 \text{ ps/nm/km}$, quindi operano nel regime dispersivo anormale.

Il ritardo introdotto dal termine β_3 dipende anche dalla frequenza, e quindi genera una distorsione dell'impulso del tipo:

$$\tau_g \beta_3 = \frac{1}{2} \beta_3 \cdot (\omega - \omega_0)^2 \cdot z$$

Siccome β_3 assume valori molto bassi, il suo contributo alla distorsione dell'impulso può essere trascurata.

La dispersione cromatica è un fenomeno che può essere compensato. Esistono tecniche di compensazione della dispersione di natura ottica o elettronica. Ad esempio, nella compensazione della dispersione di natura ottica, dopo ogni tratta del collegamento si introduce una tratta di fibra particolare chiamata *Dispersion Compensating Fiber* (DCF), la quale avendo una dispersione D di segno opposto a quello della fibra precedente, riesce a cancellare il fenomeno della dispersione raggiungendo un valore di dispersione accumulata teoricamente pari a zero. Nella compensazione di natura elettronica, una delle tecniche più efficaci è basata sull'uso della tecnologia MLSE, la quale verrà presentata nel capitolo 3.

2.1.3 Polarization Mode Dispersion (PMD)

In una fibra ottica ideale, il core ha una sezione trasversa perfettamente circolare. In questo caso il modo fondamentale ha due diverse polarizzazioni ortogonali che viaggiano alla stessa velocità. Il segnale che viene trasmesso nella fibra ha una polarizzazione casuale, cioè una sovrapposizione casuale di queste due polarizzazioni. Questo non è un problema nel caso di una fibra ottica ideale, poichè le due polarizzazioni si propagherebbero alla stessa velocità, cioè sarebbero *degeneri*.

In una fibra ottica reale esistono tuttavia imperfezioni che spezzano la simmetria circolare, determinando differenti velocità per le due polarizzazioni. In questo caso le due componenti di polarizzazione del segnale tendono a separarsi lentamente e a generare allargamento e sovrapposizione degli impulsi.

2.1.3.1 State of Polarization (SOP)

La fibra ottica è una guida d'onda con una simmetria circolare trasversa. Come risultato ha due modi degeneri, cioè due modi che hanno lo stesso profilo e caratteristiche di propagazione, ma diversa orientazione spaziale. Questo può essere visualizzato nella figura 2.2. Generalmente il campo che si propaga nella fibra contiene entrambi i modi, dando origine ad un campo globale la cui espressione può essere scritta come:

$$\mathbf{E}(z,t) = \left[A_x \cos \left(\omega t - \frac{2\pi}{\lambda} z + \delta_x \right) \right] \hat{\mathbf{x}} + \left[A_y \cos \left(\omega t - \frac{2\pi}{\lambda} z + \delta_y \right) \right] \hat{\mathbf{y}}$$

dove $E_{x,y}$ e $\delta_{x,y}$ rappresentano l'ampiezza e la fase rispettivamente del campo lungo la coordinata x,y e ω rappresenta la frequenza angolare.

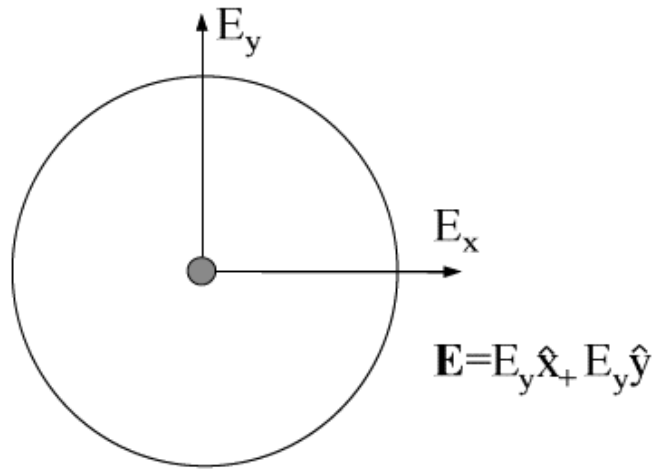


Figura 2.2. I due modi indipendenti E_x e E_y con disposizione ortogonale

Utilizzando la notazione fasoriale e trascurando l'informazione spaziale (si considera che il campo è contenuto in un piano trasverso) si ottiene:

$$\mathbf{E} = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} A_x e^{j\delta_x} \\ A_y e^{j\delta_y} \end{bmatrix}$$

E viene chiamato vettore di Jones del campo e contiene tutta l'informazione riguardante le ampiezze e le fasi di ogni componente, e anche l'informazione relativa allo *Stato di Polarizzazione*. Un modo alternativo per descrivere lo stato di polarizzazione del campo è mediante i quattro *Parametri di Stokes*, definiti come:

$$\begin{aligned} s_0 &= |E_x|^2 + |E_y|^2 \\ s_1 &= |E_x|^2 - |E_y|^2 \\ s_2 &= 2 \operatorname{Re} [E_x E_y^*] \\ s_3 &= 2 \operatorname{Im} [E_x E_y^*] \end{aligned}$$

Lo spazio dei parametri di Stokes è uno spazio tridimensionale dove l'orientamento del vettore $\mathbf{s} = (s_1, s_2, s_3)$, con $s_1^2 + s_2^2 + s_3^2 = s_0^2$, identifica univocamente uno stato di polarizzazione. La lunghezza del vettore è proporzionale a l'intensità del campo. Se a variare è soltanto la polarizzazione, la lunghezza del vettore non cambia, a cambiare sarà soltanto la direzione del vettore su una sfera normalizzata, chiamata *Sfera di Poincaré*, la quale viene mostrata nella figura 2.3.

2.1.3.2 Degree of Polarization (DOP)

Il caso generale della luce polarizzata parzialmente può essere analizzato separando il campo in due componenti: una componente *totalmente polarizzata* e una componente *non polarizzata*. Il *Grado di Polarizzazione* (DOP) si define

come il rapporto fra l'intensità della componente totalmente polarizzata e l'intensità totale del campo, così:

$$DOP \triangleq \frac{\sqrt{s_1^2 + s_2^2 + s_3^2}}{s_0}$$

Per un fascio di luce totalmente polarizzato, i parametri di Stokes soddisfano la relazione $s_1^2 + s_2^2 + s_3^2 = s_0^2$ e quindi $DOP = 1$.

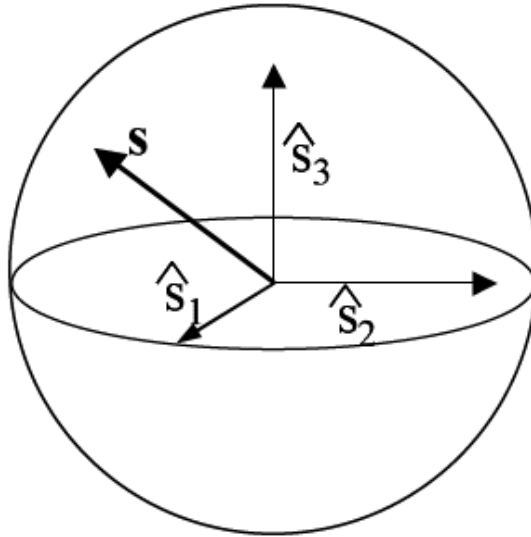


Figura 2.3. Rappresentazione del SOP s tramite la sfera di Poincaré

Il grado di polarizzazione di un segnale ottico può essere misurato utilizzando un polarizzatore variabile. Si orienta il polarizzatore a modo di ottenere il fascio più intenso e quello meno intenso, misurando così i livelli di potenza P_{max} e P_{min} rispettivamente. In questo modo il DOP si ottiene direttamente tramite la relazione:

$$DOP \triangleq \frac{P_{max} - P_{min}}{P_{tot}}$$

dove P_{tot} è la potenza del segnale prima del polarizzatore. Se la luce è completamente polarizzata, allora $P_{max} = P_{tot}$, $P_{min} = 0$ e quindi $DOP = 1$.

La PMD nasce nelle fibre a singolo modo quando si rompe la simmetria circolare, questo può accadere a causa della presenza di un core ellittico o di piegamenti con simmetria non circolare. La perdita di questa simmetria fa sì che i modi non siano più degeneri e come conseguenza le loro velocità di gruppo saranno diverse. Ogni impulso trasmesso si separa in due impulsi polarizzati ortogonalmente. La distorsione nasce come risultato della differenza di ritardo di gruppo esistente fra i due impulsi all'uscita della fibra. Questa distorsione cresce notevolmente con l'incremento della larghezza di banda, per questo motivo i sistemi che lavorano a 40 Gbps patiscono molto di più la PMD in confronto ai sistemi che lavorano a 10 Gbps, anche perché nel primo caso la durata degli impulsi è decisamente minore.

Esistono due stati di polarizzazione ortogonali in uscita (corrispondenti a due stati di polarizzazione ortogonali in ingresso) che sono indipendenti dalla lunghezza d'onda nel primo ordine, così come mostrato in [1]. Questi stati vengono chiamati *Principal States of Polarization* (PSP).

La PMD può essere caratterizzata nello spazio di Stokes dal vettore di PMD $\mathbf{\Omega}(\omega)$, attorno al quale ruota uno stato di polarizzazione $\mathbf{s}$ in uscita con la variazione della frequenza della portante, ossia:

$$\mathbf{s}' = \mathbf{\Omega} \times \mathbf{s} \quad \leftrightarrow \quad \frac{d\mathbf{s}}{d\omega} = \mathbf{\Omega}(\omega) \times \mathbf{s}$$

Il modulo di questo vettore, $\Delta\tau = |\mathbf{\Omega}|$, viene detto *Differential Group Delay* (DGD), il quale è il ritardo temporale esistente fra i due PSP.

La dipendenza del DGD dal vettore $\mathbf{\Omega}(\omega)$ è considerata come una dipendenza dalla PMD del *primo ordine*. In realtà, esiste anche una dipendenza dalla PMD del *secondo ordine*, nella quale la distorsione degli impulsi dipende anche dalla derivata del vettore di PMD $\mathbf{\Omega}'(\omega)$.

Secondo lo studio effettuato in [2], la PMD di secondo ordine influenza soltanto i sistemi a larghezza di banda elevata, come quelli in cui viene utilizzata la *modulazione diretta*. In generale, i sistemi a bassa larghezza di banda, come quelli in cui viene utilizzata la *modulazione esterna*, non patiscono la PMD di secondo ordine.

Per poter quantificare il contributo della PMD in una fibra, siccome le imperfezioni che spezzano la simmetria circolare e danno origine alla stessa sono del tutto casuali, è spesso necessario utilizzare un approccio statistico del fenomeno, così come menzionato in [3]. Questo approccio statistico dà come risultato la espressione per il calcolo del DGD:

$$\Delta\tau = D_{\text{PMD}}\sqrt{L}$$

dove D_{PMD} è il *parametro di PMD* della fibra, misurato in ps/ $\sqrt{\text{km}}$, il quale dà un'idea dell'intensità e della frequenza delle imperfezioni.

2.2 Effetti non lineari

La qualità e la capacità di trasmissione dei sistemi in fibra ottica a lunga distanza è pesantemente condizionata dagli effetti non lineari. Questi effetti sono: *Self Phase Modulation* (SPM), *Cross Phase Modulation* (XPM) e *Four Wave Mixing* (FWM).

Il fenomeno fisico che produce gli effetti non lineari viene chiamato *Effetto Kerr*. Questo si manifesta quando l'indice di rifrazione del materiale dipende dalla potenza del segnale che viene propagato attraverso questo materiale, così:

$$n(z) = n_L + \Delta n = n_L + n_2 \frac{P(z)}{A_e}$$

dove n_L è l'indice di rifrazione convenzionale del materiale, n_2 è l'indice di rifrazione non lineare e A_e è l'area effettiva del modo fondamentale, la quale può

essere approssimata con l'area del core. Si deve notare che gli effetti non lineari dipendono dalla densità di potenza per unità di area e non dalla potenza assoluta.

- *Self Phase Modulation (SPM)*

Consiste in una modulazione della fase del segnale generata dalle variazioni nella potenza del segnale stesso. La SPM di per sé non influenza la larghezza degli impulsi, ma in presenza della dispersione i due effetti interagiscono e possono sia compensarsi e cancellare l'effetto della SPM che rafforzarsi e deteriorare il segnale.

- *Cross Phase Modulation (XPM)*

Consiste in una modulazione della fase del segnale generata dalle variazioni nella potenza dei canali spettralmente adiacenti. Queste variazioni di fase possono diventare variazioni di ampiezza in presenza della dispersione, provocando un deterioramento del segnale.

- *Four Wave Mixing (FWM)*

Consiste in un trasferimento di energia fra i diversi canali il quale genera componenti spettrali del segnale a nuove frequenze. Ogni nuova frequenza ω_i nasce come il contributo di 3 frequenze ω_m , ω_n e ω_k tramite la relazione:

$$\omega_i = \omega_m - \omega_n + \omega_k.$$

Esistono due tipi di FWM: *FWM Degenere*, nel quale due delle tre frequenze sono le stesse ($\omega_m = \omega_k \neq \omega_n$), e *FWM Non degenere*, nel quale tutte e tre le frequenze sono diverse ($\omega_m \neq \omega_n \neq \omega_k$).

Capitolo 3

Maximum Likelihood Sequence Estimation (MLSE)

La Maximum Likelihood Sequence Estimation (MLSE) o *decodifica a massima verosimiglianza* è una tecnica di ricezione ottimizzata la quale minimizza la probabilità di errore su di una determinata sequenza. In questo capitolo verranno discussi il principio della massima verosimiglianza, la stima delle metriche di ramo, la struttura di un sistema MLSE e l'algoritmo di Viterbi.

3.1 Decodifica a massima verosimiglianza

Si consideri un modello rappresentante la funzione di densità di probabilità di una variabile X in funzione di un parametro θ . Per un determinato valore x di X , la funzione $L(\theta|x) = P(X=x|\theta)$ è una *funzione di verosimiglianza* di θ , ossia dà una misura di quanto “verosimile” è un determinato valore di θ , dato che X assume il valore x . Quindi una funzione di verosimiglianza nasce dalla funzione di

probabilità condizionale usando come variabile il secondo parametro e tenendo il primo fisso.

Si può dire che la funzione di verosimiglianza ha un comportamento opposto rispetto alla funzione di probabilità: se la prima serve a determinare parametri sconosciuti sulla base di eventi noti, la seconda serve a determinare eventi sconosciuti sulla base di parametri noti. Il principio della massima verosimiglianza consiste nella scelta del parametro corrispondente al valore massimo nella funzione di verosimiglianza.

Tutto questo può essere esteso ad un sistema di trasmissione. Si consideri la ricezione di un vettore di segnale $r = [r_0, r_1, \dots, r_{N-1}]$. Se S_t rappresenta una di tutte le possibili sequenze trasmesse $[s_0, s_1, \dots, s_{N-1}]$, allora la regola di decisione ottima indica che la sequenza trasmessa S_T corrisponderà alla sequenza S_t che massimizzi la funzione di probabilità:

$$p(S_t / r) = \frac{f(r | S_t) p(S_t)}{f(r)}$$

Questa funzione si reduce alla regola di massima verosimiglianza:

$$s(r) = \arg \max_{S_t} f(r | S_t)$$

dove le funzioni $f(r / S_t)$ sono funzioni di verosimiglianza.

La sequenza che più probabilmente è stata trasmessa è la associata alla minima distanza Euclidea fra il segnale ricevuto r e la possibile sequenza trasmessa S_t :

$$D(r, S_t) = \sum_{k=0}^{N-1} (r_k - s_k)^2$$

Questa funzione viene chiamata *metrica*, è può essere riscritta come:

$$\Lambda(S_t) = \sum_{k=0}^{N-1} \lambda(x_k, x_{k+1})$$

dove x_k è la rappresentazione del vettore $[s_{k-L+1}, s_{k-L+2}, \dots, s_{k-1}]$, contenente $(L-1)$ simboli consecutivi di informazione.

La metrica ottima si ottiene come la somma delle metriche parziali $\lambda(x_k, x_{k+1})$. Il termine k -esimo, $\lambda(x_k, x_{k+1})$, dipende dai vettori corrispondenti a simboli consecutivi $[s_{k-L+1}, s_{k-L+2}, \dots, s_{k-1}]$ e $[s_{k-L+2}, s_{k-L+3}, \dots, s_k]$ rispettivamente.

Queste considerazioni permettono l'utilizzo di una formula ricorsiva per il calcolo di $\Lambda(S_t)$, quindi si definisce la relazione:

$$\Lambda(s_k) = \Lambda(s_{k-1}) + \lambda(x_k, x_{k+1})$$

dove $s_i = [s_0, s_1, \dots, s_i]$, $(s_{N-1} = S_t)$ e $\Lambda(s_0) = 0$. Dopo N iterazioni si ottiene:

$$\Lambda(S_t) = \Lambda(s_{N-1})$$

Una rappresentazione geometrica della ricerca della metrica ottima può essere descritta tramite un *diagramma a traliccio*, o trellis, con N_s stati come quello della figura 3.1. Il trellis è un diagramma che rappresenta tutte le possibili traiettorie che può assumere una catena di Markov con N_s stati dall'istante 0 all'istante N . Se δ è la memoria del canale, il numero di stati necessari per tenere conto di tutta l'ISI introdotta dal canale è pari a $N_s = 2^{\delta+1}$. Ogni stato viene "etichettato" con i δ simboli precedenti, mentre il simbolo corrente etichetta il ramo in uscita dallo stato.

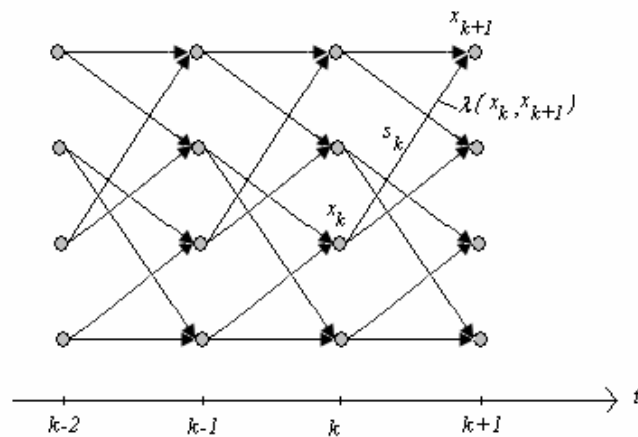


Figura 3.1. Diagramma a traliccio con 4 stati e $M=2$

Nel k -esimo istante ogni stato rappresenta uno dei possibili N_s valori che x_k può assumere, ed è collegato tramite M rami ai possibili stati seguenti x_{k+1} (dove M è il numero di simboli M-ari, in trasmissione binaria $M=2$). Il ramo che collega gli stati x_k e x_{k+1} viene etichettato con il simbolo corrente s_k e la *metrica di ramo* $\lambda(x_k, x_{k+1})$. In questo modo ogni sequenza S_t ha una corrispondenza biunivoca con una sequenza di stati nel trellis. Cercare la sequenza ottima è equivalente a cercare il cammino che abbia la minore metrica accumulata lungo il trellis. Questa ricerca richiederebbe un'analisi esaustiva e la comparazione fra tutte le possibili sequenze trasmesse. Per una sequenza di N simboli, si dovrebbero effettuare M^N comparazioni per poter scegliere la sequenza ottima, quindi il numero di comparazioni cresce esponenzialmente rispetto a N . Procedere in questo modo ha una complessità eccessiva, soprattutto nei sistemi in fibra ottica, i quali lavorano con sequenze molto lunghe.

Questa ricerca può essere effettuata ricorsivamente con una complessità decisamente minore utilizzando l'*Algoritmo di Viterbi*, nel quale il numero di comparazioni necessarie per poter scegliere la sequenza ottima cresce linearmente rispetto a N . L'algoritmo di Viterbi verrà discusso nella sezione 3.4.

3.2 Stima delle metriche di ramo

Come è stato detto precedentemente, il sistema MLSE viene implementato utilizzando l'algoritmo di Viterbi. Ad ogni iterazione, si deve calcolare la metrica per tutti i rami del trellis nel seguente modo:

$$M^{(n)} = -\sum_{k=1}^K \ln \{f_{y_{n,k}}(y)\}$$

dove n va da 1 al numero totale di rami esistenti nel trellis ($M \cdot N_s$), k va da 1 al numero totale K di campioni per simbolo, $y_{n,k}$ è una variabile aleatoria associata al segnale rumoroso, essendo il segnale trasmesso quello associato al k -esimo

campione dell' n -esimo ramo, $f_{y_{n,k}}(y)$ è la funzione di densità di probabilità (pdf) di $y_{n,k}$ e y è il campione rumoroso corrente preso dal segnale elettrico $y(t)$ ricevuto dal fotorivelatore.

L'espressione per il calcolo della metrica di ramo ha bisogno delle pdf dei campioni del segnale. Purtroppo, queste pdf non possono essere calcolate analiticamente nei sistemi ottici che lavorano con filtri elettrici di post-rivelazione, quindi sono state implementate tecniche semi-analitiche basate sulla espansione di Karhunen-Loève [4] in modo da poter calcolare correttamente le metriche di ramo.

Con la finalità di ridurre la complessità computazionale associata al calcolo delle pdf con la espansione di Karhunen-Loève, nella pratica vengono utilizzati altri modelli per approssimare le statistiche del segnale. Alcuni di questi modelli si basano sull'ipotesi che i campioni ricevuti abbiano una distribuzione *Gaussiana*. Di seguito vengono descritte due metriche basate su questa assunzione.

3.2.1 Metrica Gaussiana

La complessità nel calcolo della metrica può essere ridotta considerando che i campioni ricevuti del segnale non siano correlati e abbiano una distribuzione Gaussiana. L'espressione per il calcolo della metrica è la seguente:

$$M_{GAUSS}^{(n)} = -\sum_k \left\{ \frac{(y_k - \mu_{n,k})^2}{\sigma_{n,k}^2} + \ln(\sigma_{n,k}^2) \right\}$$

dove y_k è il k -esimo campione rumoroso ricevuto, e $\mu_{n,k}$ e $\sigma_{n,k}^2$ sono rispettivamente la media e la varianza del k -esimo campione dell' n -esimo ramo del trellis.

La metrica Gaussiana tiene conto della non stazionarietà della varianza del rumore. Di conseguenza, oltre al valor medio dei campioni, anche la varianza

deve essere stimata per ogni ramo del trellis tramite la matrice di varianza $\sigma_{n,k}^2$. Questo rende la metrica Gaussiana molto intensa in termini di complessità computazionale.

3.2.2 Metrica a radice quadrata

La metrica a radice quadrata (square root) è una versione semplificata nella quale si considera che la radice quadrata del segnale ricevuto abbia una distribuzione Gaussiana e che la sua varianza sia stazionaria.

Con queste considerazioni il calcolo della metrica a radice quadrata si fa mediante l'espressione:

$$M_{SQRT}^{(n)} = -\sum_k \left(\sqrt{y_k} - \mu'_{n,k} \right)^2$$

dove y_k è il k -esimo campione rumoroso ricevuto, e $\mu'_{n,k}$ è la media del k -esimo campione corrispondente all' n -esimo ramo del trellis.

Questa metrica ha come vantaggio il fatto che durante la fase di stima del canale deve essere calcolata soltanto la media del segnale, e che la sua espressione può essere scritta in forma chiusa e semplice.

Studi sulle prestazioni della metrica a radice quadrata e sulla stazionarietà della sua varianza sono stati effettuati in [5], dove si conclude che questa metrica è quella che fra tutte quelle studiate si avvicina di più alle prestazioni ottenute con la metrica esatta. Si presume anche che la metrica a radice quadrata sia una forte candidata per le prossime generazioni di sistemi MLSE, i quali avranno un numero di stati molto più elevato da quelli esistenti al momento, che lavorano con trellis a quattro stati.

3.3 Struttura del sistema MLSE

Lo schema generale di un ricevitore MLSE viene mostrato nella figura 3.2. Il ricevitore è composto da 4 blocchi principali:

- Convertitore Analogico-Digitale (ADC), il quale trasforma il segnale elettrico in un segnale digitale (campioni quantizzati).
- Recupero del clock, il quale effettua la sincronizzazione dei simboli nel dominio elettrico.
- Blocco MLSE, il quale effettua la decodifica a massima verosimiglianza dei campioni ricevuti.
- Blocco di stima del canale, il quale calcola i parametri del canale che sono usati per controllare i blocchi ADC e MLSE.

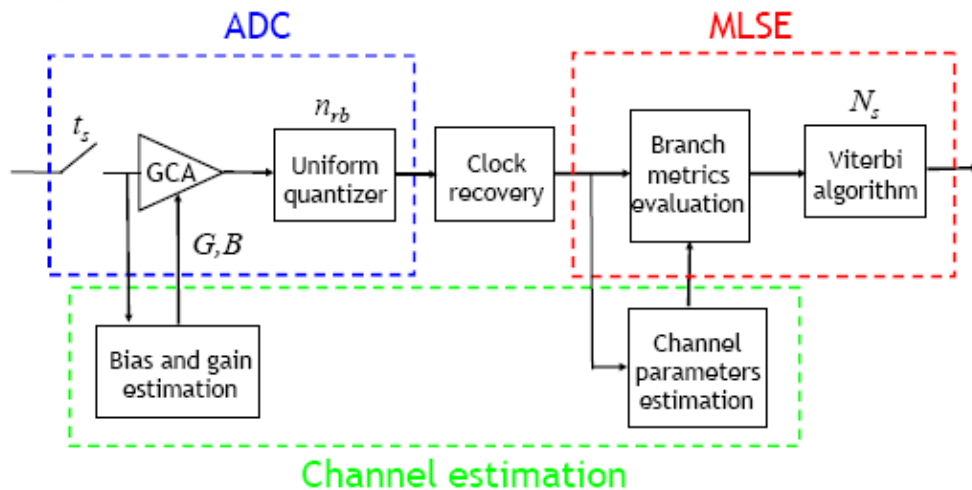


Figura 3.2. Schema del ricevitore MLSE

Prima di entrare al blocco MLSE, il segnale elettrico analogico viene campionato e quantizzato dal blocco ADC. Si assume che il blocco ADC è composto da un campionario seguito da un *amplificatore a guadagno controllato* (GCA) e da un quantizzatore uniforme con “range” fisso a $[-1,1]$.

Se il “range” del quantizzatore si adatta perfettamente alla variazione del segnale $[\min\{y_i^{out}\}, \max\{y_i^{out}\}]$, si dice che il quantizzatore non ha *overflow*. Generalmente, le migliori prestazioni si raggiungono quando si effettua un “taglio” o *clipping* dei campioni in ingresso, questo accade quando il “range” del quantizzatore è minore dell’intervallo di variazione del segnale, così come mostrato nella figura 3.3.

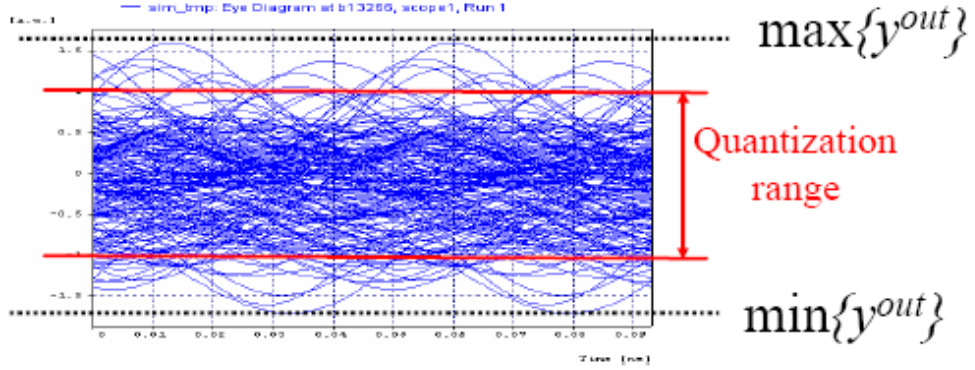


Figura 3.3. Diagramma ad occhio del segnale ricevuto con rango di quantizzazione $[-1,1]$

Si definisce il *fattore di clipping* K_f come il rapporto fra la variazione del segnale e il “range” di quantizzazione, così:

$$K_f = \frac{\max\{y^{out}\} - \min\{y^{out}\}}{2}$$

In questo modo, il fattore di clipping K_f è il numero per il quale si devono dividere i valori dei campioni quantizzati per poter adattarli al “range” di quantizzazione $[-1,1]$.

In seguito, i campioni in uscita dal quantizzatore, i quali hanno un’intervallo di variazione fisso, passano al blocco MLSE, nel quale si fa la stima delle metriche di ramo (studiata nella sezione 3.2) e si applica l’algoritmo di Viterbi (studiato nella sezione 3.4).

3.4 Algoritmo di Viterbi

L'algoritmo di Viterbi è stato presentato in [6] e la sua esecuzione consiste nel trovare il cammino che minimizzi la metrica fra tutti quelli che attraversano il trellis da sinistra verso destra (dall'istante $k = 0$ all'istante $k = N$). La metrica associata ad un cammino è la somma delle metriche che etichettano i rami che formano il cammino.

Se x_k denota lo stato all'istante k , che può assumere i valori $\{X_i\}_{i=1}^{N_S}$, e $\lambda(x_k, x_{k+1})$ denota la metrica associata al ramo che parte dallo stato x_k e arriva allo stato x_{k+1} , allora l'algoritmo prova a minimizzare la funzione:

$$\Lambda(x_0, x_1, \dots, x_{N-1}) = \sum_{k=0}^{N-1} \lambda(x_k, x_{k+1})$$

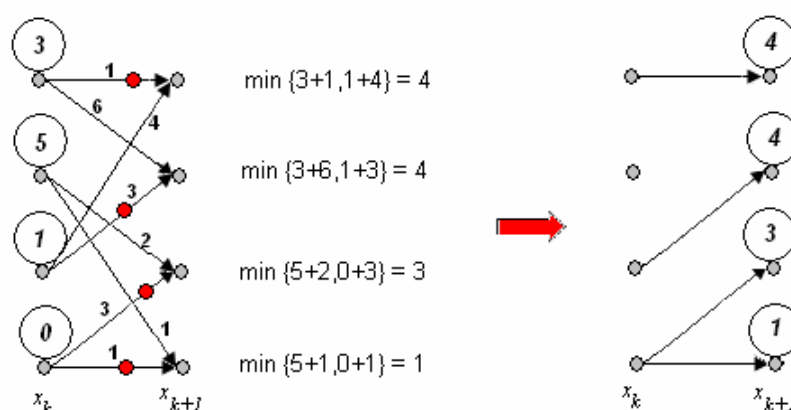
fra tutte le possibili sequenze di stati $(x_0, \dots, x_{N-1})$ che siano compatibili con la struttura del trellis.

La soluzione può essere trovata calcolando tutti i valori possibili che prende la funzione Λ e scegliendo il minore. Purtroppo questa tecnica sarebbe troppo difficile da implementare dato che richiederebbe un numero troppo elevato di calcoli e di capacità di elaborare informazione, dal momento che questo numero cresce esponenzialmente con la lunghezza della sequenza N .

L'algoritmo di Viterbi riesce a minimizzare la funzione Λ con una complessità computazionale che cresce linearmente con N . Questo viene fatto utilizzando un approccio a tre passi: *Sommare* (Add), *Comparare* (Compare) e *Selezionare* (Select) (ACS).

Si consideri la figura 3.4 nella quale vengono mostrati gli stati del trellis per gli istanti k e $k+1$. I rami che collegano i diversi stati sono etichettati con le corrispondenti metriche di ramo, mentre gli stati sono etichettati con la *metrica accumulata*, la quale verrà definita in seguito. I tre passi dell'approccio ACS sono:

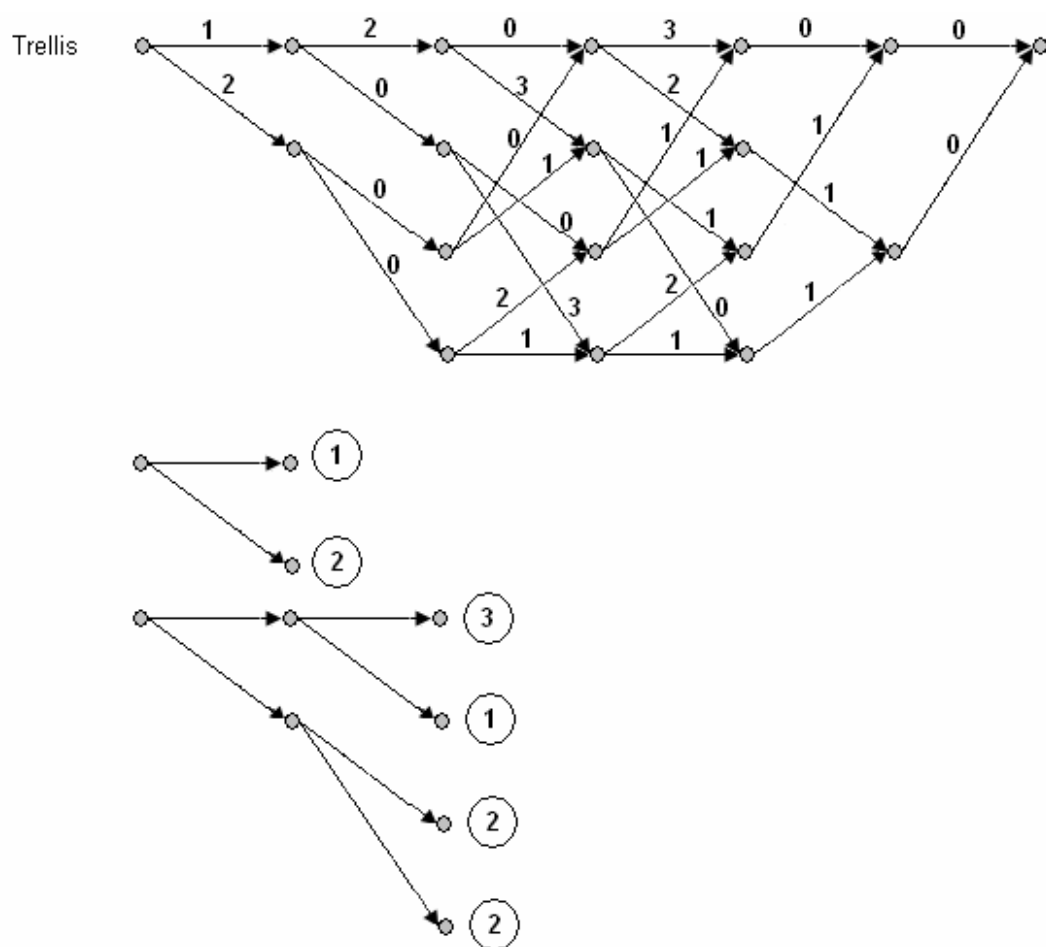
- Per ogni stato x_{k+1} vengono esaminati i rami che partono da i diversi stati x_k e arrivano a lui. Per ognuno di questi rami si *somma* la metrica del ramo stesso a quella dello stato x_k .
- Si *comparano* i risultati di queste somme.
- Si *seleziona* il ramo associato al valore più basso e gli altri rami vengono scartati (se il valore più basso viene raggiunto da più di un ramo, può essere scelto qualsiasi di quei rami a caso). Questo valore minimo viene associato all'istante x_{k+1} e forma la sua *metrica accumulata* (questo valore viene elaborato solo nel passo successivo dell'ACS, dopodichè viene scartato).



L'algoritmo di Viterbi consiste in ripetere l'approccio ACS dal primo fino all'ultimo stato del trellis. Dopo aver effettuato l'ACS per ogni stato si memorizza il valore della metrica accumulata ed il cammino che porta ad esso. Quindi, in ogni istante k , per ogni x_k sopravviverà soltanto un valore di metrica accumulata ed un cammino che attraversa il trellis dal suo stato iniziale fino a x_k . Questo cammino è quello che ha la metrica minore per quello stato. Una volta applicato l'ACS N volte fino alla fine del trellis, si ottiene un valore di metrica accumulata

ed un cammino di N rami. Questi corrispondono al *valore di metrica minore* ed al *cammino di metrica minore* rispettivamente.

Nella figura 3.5 viene mostrato un esempio della determinazione del cammino di metrica minore utilizzando l'algoritmo di Viterbi nel caso di un trellis a quattro stati con $N=6$.



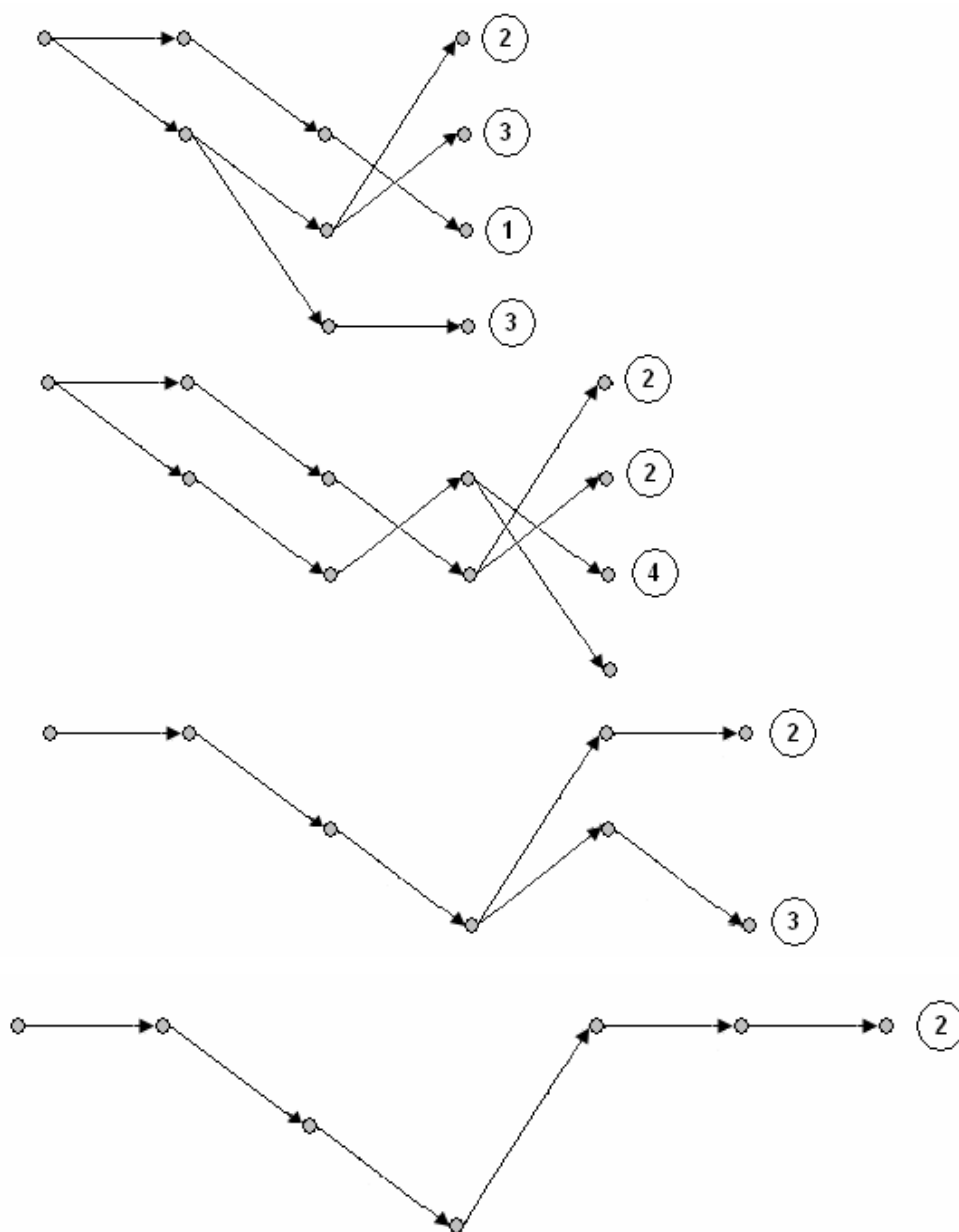


Figura 3.5. Determinazione del cammino di metrica minore utilizzando l'algoritmo di Viterbi

3.5 Caratteristiche del ricevitore MLSE per DQPSK

Il formato di modulazione DQPSK, formato da un segnale a quattro livelli, utilizza un ricevitore che demodula il segnale ottico ricevuto generando due segnali elettrici paralleli $y^I(t)$ e $y^Q(t)$, uno per ogni quadratura del segnale. Questi due segnali possono essere visti come due sequenze di bit: una sequenza per i bit *in fase* e l'altra per i bit *in quadratura*.

Essendo il DQPSK l'unico dei formati di modulazione studiati dove si ha più di una sequenza di bit contemporaneamente, occorre analizzare le diverse configurazioni che la decodifica MLSE di questo segnale può assumere.

Esistono tre configurazioni diverse per il ricevitore MLSE del DQPSK: *Balanced single-input double-MLSE processor* (in cui si ha il ricevitore bilanciato e due blocchi MLSE, uno per ogni quadratura del segnale), *Balanced double-input MLSE processor* (in cui si ha il ricevitore bilanciato e viene utilizzato un singolo blocco MLSE che lavora con simboli composti da due bit paralleli) e *Unbalanced quad-input MLSE processor* (in cui i segnali elettrici dei rami di ogni AMZ non vengono combinati e vengono parallelamente dati in ingresso ad un singolo blocco MLSE che lavora con simboli composti da due bit paralleli).

3.5.1 Balanced single-input double-MLSE processor

Lo schema del ricevitore viene mostrato nella figura 3.6. Un ricevitore bilanciato standard DQPSK genera i segnali elettrici in fase e in quadratura $y^I(t)$ e $y^Q(t)$, i quali entrano in una coppia di convertitori ADC. I campioni quantizzati y_k^I e y_k^Q vengono dati in ingresso a due blocchi MLSE indipendenti.

I blocchi MLSE sono basati sull'algoritmo di Viterbi. Un esempio di trellis a 16 stati viene mostrato nella figura 3.7.

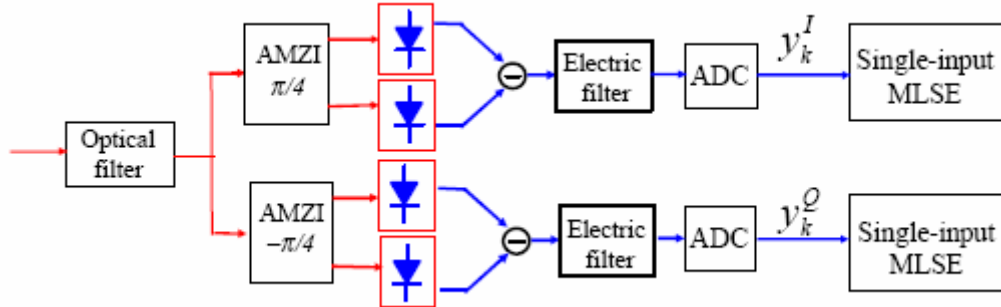


Figura 3.6. Schema del Balanced single-input double-MLSE processor

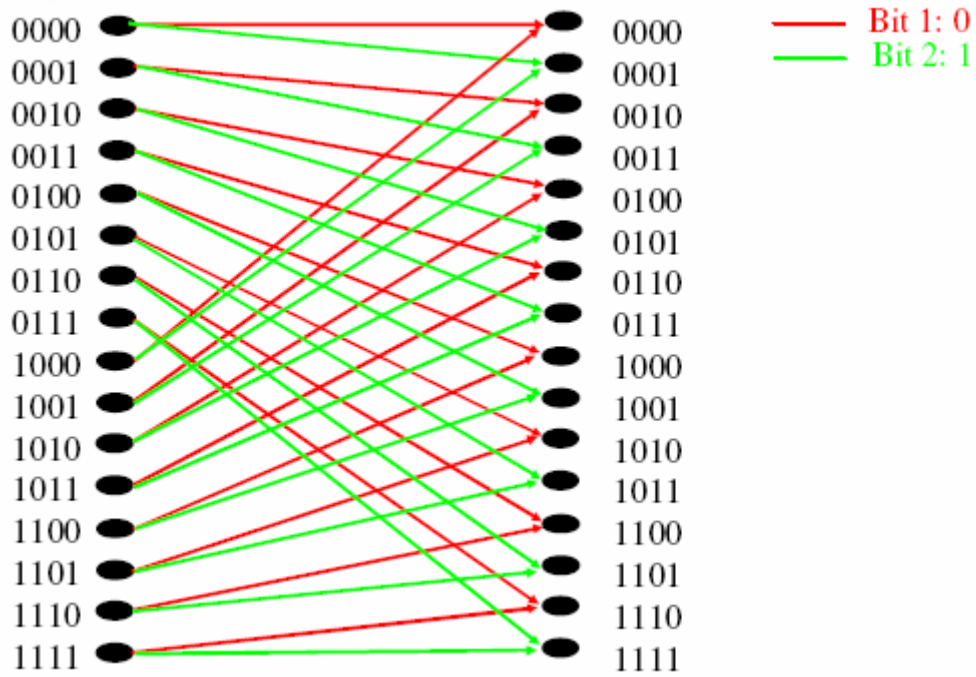


Figura 3.7. Trellis a 16 stati per il Balanced single-input double-MLSE processor

Il trellis della figura 3.7 corrisponde ad una memoria del canale pari a 4, cioè ogni stato del trellis viene etichettato dai 4 bit precedenti. Ogni stato ha 2 rami in ingresso e altrettanti in uscita, ognuno corrispondente al fatto che il bit sia uno “0” o un “1”.

Le metriche di ramo vengono calcolate così:

$$M_{DQPSK}^n = - \sum_{k=1}^K \log \left\{ f^{(n)}(y_k^{I,Q}) \right\}$$

dove n è il numero del ramo, K è il numero di campioni per bit e $f^{(n)}(y_k^{I,Q})$ è la probabilità di ricevere il campione y_k^I (o y_k^Q) per l' n -esimo ramo del trellis.

Assumendo che i campioni del segnale in fase e in quadratura abbiano una distribuzione Gaussiana, la loro PDF viene data da:

$$f^{(n)}(y_k^{I,Q}) = \frac{1}{\sqrt{2\pi}(\sigma_{n,k}^{I,Q})^2} \cdot \exp \left\{ -\frac{(y_k - \mu_{n,k}^{I,Q})^2}{(\sigma_{n,k}^{I,Q})^2} \right\}$$

dove i diversi μ e σ sono rispettivamente la media e la deviazione standard del k -esimo campione del segnale corrispondente all' n -esima transizione del trellis.

3.5.2 Balanced double-input MLSE processor

Lo schema del ricevitore viene mostrato nella figura 3.8.

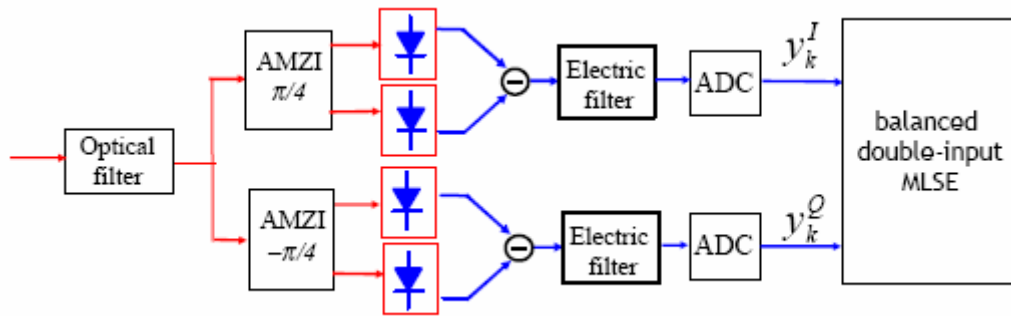


Figura 3.8. Schema del Balanced double-input MLSE processor

Un ricevitore bilanciato standard DQPSK genera i segnali elettrici in fase e in quadratura $y^I(t)$ e $y^Q(t)$, i quali entrano in una coppia di convertitori ADC. I

campioni quantizzati y_k^I e y_k^Q vengono dati in ingresso ad un singolo blocco MLSE il quale lavora con simboli composti da 2 bit paralleli (uno per ogni quadratura del segnale).

Anche in questo caso il blocco MLSE è basato sull'algoritmo di Viterbi. Un esempio di trellis a 16 stati viene mostrato nella figura 3.9.

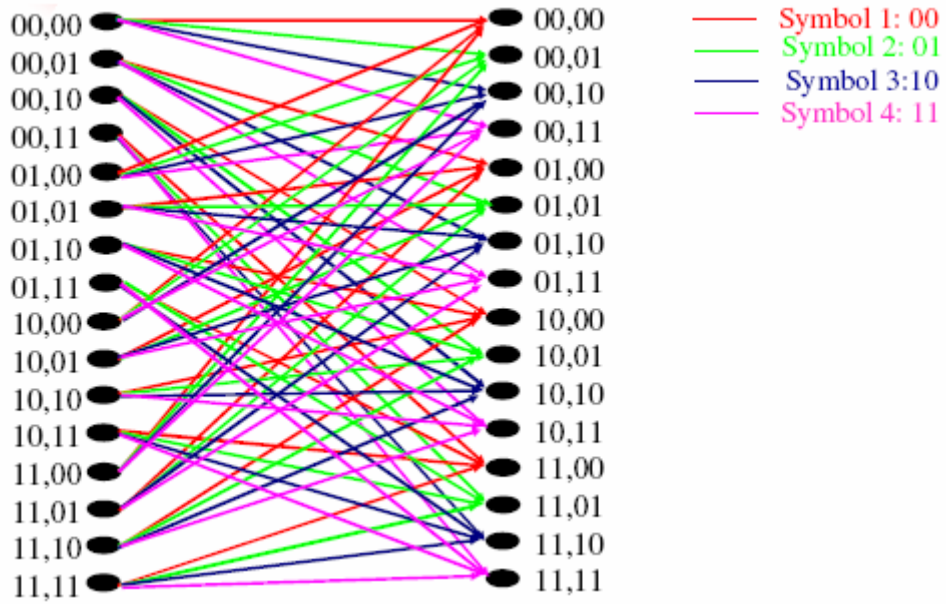


Figura 3.9. Trellis a 16 stati per il Balanced double-input MLSE processor

Il trellis della figura 3.9 corrisponde ad una memoria pari a 2, cioè ogni stato del trellis viene etichettato dai 2 simboli precedenti. Ogni stato ha 4 rami in ingresso e altrettanti in uscita, corrispondenti ai simboli “00”, “01”, “10” e “11”.

Le metriche di ramo vengono calcolate così:

$$M_{DQPSK}^n = - \sum_{k=1}^K \log \left\{ f^{(n)}(y_k^I, y_k^Q) \right\}$$

dove n è il numero del ramo, K è il numero di campioni per bit e $f^{(n)}(y_k^I, y_k^Q)$ è la probabilità di ricevere i campioni y_k^I e y_k^Q per l' n -esimo ramo del trellis.

Assumendo che i campioni del segnale in fase e in quadratura abbiano una distribuzione Gaussiana, la loro PDF viene data dalla densità di probabilità congiunta. Assumendo anche che i campioni siano indipendenti, questa funzione non è che il prodotto delle due singole densità di probabilità, cioè:

$$f^{(n)}(y_k^I, y_k^Q) = \frac{1}{\sqrt{2\pi(\sigma_{n,k}^I)^2}} \cdot \exp \left\{ -\frac{(y_k - \mu_{n,k}^I)^2}{(\sigma_{n,k}^I)^2} \right\} \cdot \frac{1}{\sqrt{2\pi(\sigma_{n,k}^Q)^2}} \cdot \exp \left\{ -\frac{(y_k - \mu_{n,k}^Q)^2}{(\sigma_{n,k}^Q)^2} \right\}$$

dove i diversi μ e σ sono rispettivamente la media e la deviazione standard del k -esimo campione del segnale corrispondente all' n -esima transizione del trellis.

3.5.3 Unbalanced quad-input MLSE processor

Lo schema del ricevitore viene mostrato nella figura 3.10.

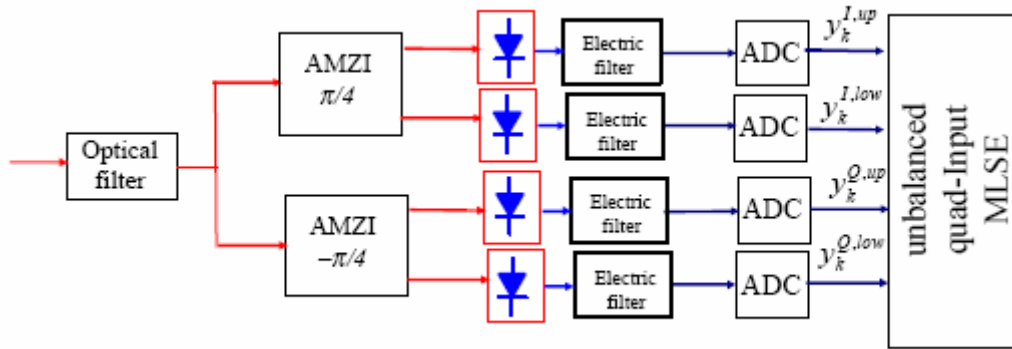


Figura 3.10. Schema del Unbalanced quad-input MLSE processor

I 4 segnali elettrici in uscita dei rami superiori e inferiori dei 2 AMZ non vengono combinati ma entrano parallelamente in 4 convertitori ADC. I 4 campioni quantizzati $y_k^{I,up}$, $y_k^{I,low}$, $y_k^{Q,up}$ e $y_k^{Q,low}$ vengono dati in ingresso ad un singolo blocco MLSE il quale lavora con simboli composti da 2 bit paralleli (uno per ogni quadratura del segnale).

La struttura del trellis è la stessa di quella del Balanced double-input MLSE (figura 3.9). Siccome i valori in ingresso al blocco MLSE non sono più bipolari (sono delle ampiezze detettate dai singoli fotodiodi e poi quantizzate, quindi sono valori positivi), la metrica a radice quadrata può essere usata.

Assumendo che tanto i campioni in fase come quelli in quadratura di entrambi i rami sono indipendenti, le metriche di ramo vengono calcolate così:

$$M_{DQPSK}^n = \sum_{k=1}^K \left(\sqrt{y_k^{I,low}} - \mu_{n,k}^{I,low} \right)^2 + \left(\sqrt{y_k^{I,up}} - \mu_{n,k}^{I,up} \right)^2 + \left(\sqrt{y_k^{Q,low}} - \mu_{n,k}^{Q,low} \right)^2 + \left(\sqrt{y_k^{Q,up}} - \mu_{n,k}^{Q,up} \right)^2$$

dove n è il numero del ramo, K è il numero di campioni per bit e i diversi μ sono le medie del k -esimo campione del segnale corrispondente all' n -esimo ramo del trellis.

I risultati mostrati nei prossimi capitoli sono stati ottenuti utilizzando il *Balanced double-input MLSE processor*, scelto in quanto rappresenta il miglior compromesso tra prestazioni e complessità.

Capitolo 4

Prestazioni della decodifica MLSE in presenza della PMD

La PMD è un fenomeno che penalizza fortemente i sistemi ottici che lavorano a bit-rate di 40 Gbps e superiori, richiedendo valori di *Optical Signal-to-Noise Ratio* (OSNR) molto superiori di quelli necessari nel caso di assenza della PMD. Con la finalità di ridurre questa penalità vengono utilizzati i ricevitori MLSE, i quali realizzano una elaborazione del segnale digitale ricevuto minimizzando direttamente la probabilità di errore sulla sequenza e di conseguenza il *Bit Error Rate* (BER) del sistema.

In questo capitolo verranno studiate le prestazioni della decodifica MLSE in un sistema ottico con bit-rate pari a 43 Gbps. Questo bit-rate corrisponde allo standard di 40 Gbps più l'overload necessario per lavorare con i codici *Forward Error Correction* (FEC). Questi codici e le loro caratteristiche verranno presentate nel capitolo 6.

Il sistema sotto studio prevede il collegamento diretto fra trasmettitore e ricevitore (back-to-back) e l'inserimento artificiale della PMD nella forma di *Differential Group Delay* (DGD). In questo modo si simula un sistema di

trasmissione in cui la dispersione è compensata otticamente al 100%. Verranno prima analizzate le prestazioni di un sistema standard con l'utilizzo del ricevitore MLSE e dopo i miglioramenti che si possono ottenere ottimizzando la larghezza di banda dei filtri elettrici in trasmissione e in ricezione. Queste prestazioni verranno analizzate utilizzando quattro formati di modulazione diversi: Duobinary, IMDD, DPSK e DQPSK. Alla fine del capitolo viene presentato un confronto fra le prestazioni ottenute dai diversi formati di modulazione.

Le simulazioni del sistema ottico e del ricevitore MLSE sono state effettuate utilizzando il software OptSim.

4.1 Caratteristiche del sistema

Le caratteristiche del sistema sotto studio e delle simulazioni effettuate sono le seguenti:

- Frequenza di lavoro: 194 THz (1545 nm).
- Bit-rate (R_b): 43 Gbps. Symbol-rate (R_s): 21.5 Gbaud (per DQPSK).
- Fotodiodo ideale (per Duobinary e IMDD). Ricevitore DPSK singolo (per DPSK) e doppio (per DQPSK).
- Rotatore di polarizzazione a 90° . Con questo la potenza si divide esattamente alla metà in ognuno dei PSP (caso peggiore).
- Densità spettrale di rumore ottico accomodata per avere l'OSNR richiesto su una banda pari a 0.1 nm.
- Filtro ottico di tipo supergaussiano con $B_{-3dB}=215$ GHz (per Duobinary, IMDD e DPSK) e $B_{-3dB}=50$ GHz (per DQPSK).
- Filtro elettrico di trasmissione di Bessel a 5 poli con $B_{-3dB}=10$ GHz (per Duobinary), $B_{-3dB}=32.25$ GHz (per IMDD e DPSK) e $B_{-3dB}=9$ GHz (per DQPSK).

- Filtro elettrico di ricezione di Bessel a 5 poli con $B_{-3dB}=32.25$ GHz (per Duobinary, IMDD e DPSK) e $B_{-3dB}=18$ GHz (per DQPSK).
- Sequenza pseudo-casuale o *Pseudo-Random Bit Sequence* (PRBS) di ordine 18. Numero di bits presi in considerazione per l'analisi MLSE pari a 2^{18} (262144 bits). Numero di campioni per bit pari a 24 (per Duobinary, IMDD e DPSK) e 20 (per DQPSK).
- Nel caso di Duobinary, DPSK e DQPSK la PRBS viene opportunamente *precodificata*.
- Numero di campioni per bit utilizzati dal MLSE pari a 4 (per Duobinary, IMDD e DPSK) e 2 (per DQPSK). Non viene realizzata l'ottimizzazione dell'istante di campionamento.
- Numero di stati utilizzati dall'algoritmo di Viterbi pari a 64, corrispondenti ad una memoria del canale pari a 6 bits (per Duobinary, IMDD e DPSK) e 3 simboli (per DQPSK).
- Le prestazioni del sistema vengono misurate attraverso l'OSNR richiesto per avere $BER=10^{-3}$, il quale assicura il funzionamento con l'utilizzo dei FEC.

L'OSNR viene definito come il rapporto fra la potenza del segnale e quella del rumore, su una banda B pari a 0.1 nm (12.5 GHz), così:

$$OSNR = \frac{\bar{P}_{RX}}{2N_0B}$$

Nelle simulazioni, è stato ottenuto il segnale elettrico e filtrato all'uscita del ricevitore per diversi valori della coppia (OSNR , DGD). Questo segnale viene elaborato sia dal ricevitore standard con soglia di decisione opportuna (per l'analisi senza MLSE) che dal ricevitore MLSE, ottenendo in ogni caso un valore di BER pari al rapporto fra il numero di errori ed il numero di bit trasmessi (questo rapporto viene effettuato confrontando la sequenza trasmessa con quella ottenuta dopo la decisione). Questi valori di BER vengono interpolati linearmente

(si calcola la funzione $\log(-\log(\text{BER}))$) essendo la più simile ad una retta) per trovare per ogni valore di DGD il valore di OSNR necessario per avere $\text{BER}=10^{-3}$. Un'esempio di questo tipo di curva viene mostrato nella figura 4.1.

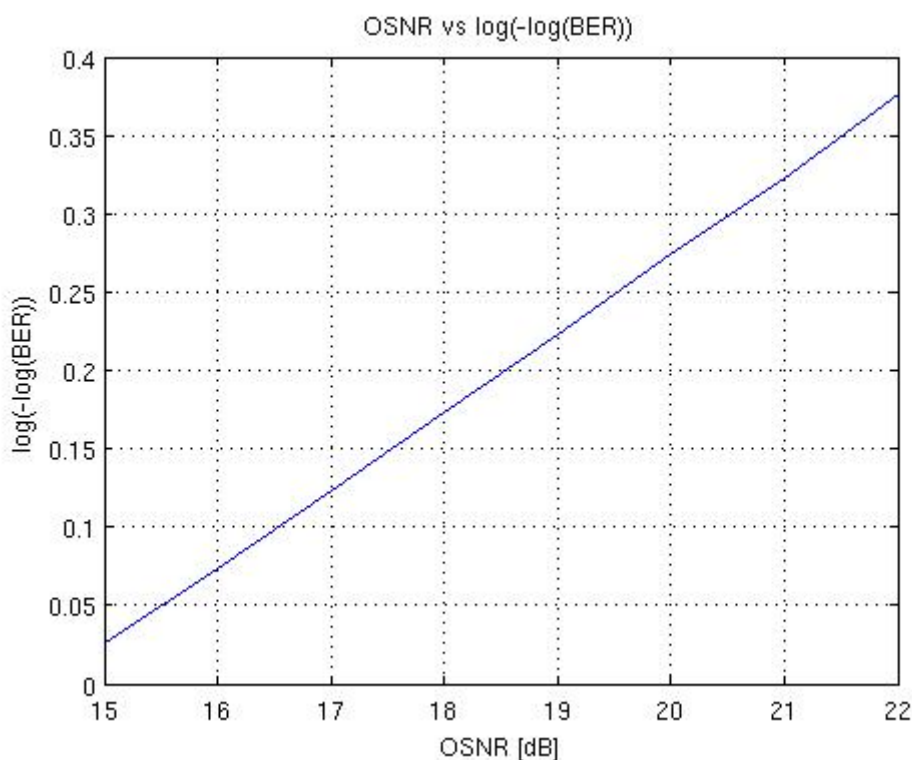


Figura 4.1. Rappresentazione della funzione $\log(-\log(\text{BER}))$ come una retta

4.2 Modulazione Duobinary

Lo schema del sistema simulato in OptSim viene mostrato nella figura 4.2. Il driver mappa il livello alto come “5.0” e il livello basso come “-5.0”. Questo segnale elettrico viene filtrato per “allargare” gli impulsi prima di entrare nel modulatore di ampiezza ideale del tipo “ $\sin^2$ ”.

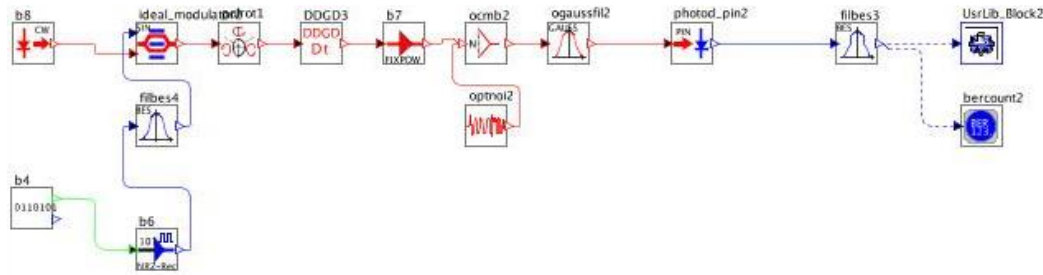


Figura 4.2 Schema del sistema per modulazione Duobinary

Come primo approccio è stato calcolato l'andamento dell'OSNR richiesto al ricevitore per garantire $BER=10^{-3}$ al variare del Differential Group Delay (DGD) con e senza l'utilizzo del blocco MLSE per migliorare le prestazioni. Il risultato viene mostrato nella figura 4.3.

Il DGD si è stato variato da 0 a 30 ps. All'interno di questo "range" c'è il valore corrispondente a $T_b = 1/R_b = 23.25$ ps. Si può notare come il massimo di penalità sull'OSNR rispetto al caso $DGD=0$ si raggiunge quando il DGD prende valori vicini al tempo di bit, e che questa penalità è pari a circa 5 dB.

Senza l'utilizzo del blocco MLSE questa penalità si raggiunge già per $DGD=15$ ps e andando oltre il segnale ricevuto viene deteriorato notevolmente.

Per cercare di migliorare le prestazioni, sono state ottimizzate le larghezze di banda dei filtri elettrici in trasmissione e in ricezione, lasciando inalterato il filtro ottico di ricezione. Per fare questo ci siamo posizionati nel caso $DGD=23.25$ ps e abbiamo fissato l'OSNR pari a quello richiesto per avere $BER=10^{-3}$ (22.0935 dB).

La larghezza di banda del filtro di trasmissione è stata variata da $0.2 \cdot R_b$ a $0.3 \cdot R_b$ con passi da $0.01 \cdot R_b$ (il valore non ottimizzato era pari a $0.2325 \cdot R_b$), mentre quella del filtro di ricezione è stata variata da $0.5 \cdot R_b$ a $2 \cdot R_b$ con passi da $0.25 \cdot R_b$ (il valore non ottimizzato era pari a $0.75 \cdot R_b$). E' stata cercata la coppia (B_{TX}, B_{RX}) corrispondente al BER minore.

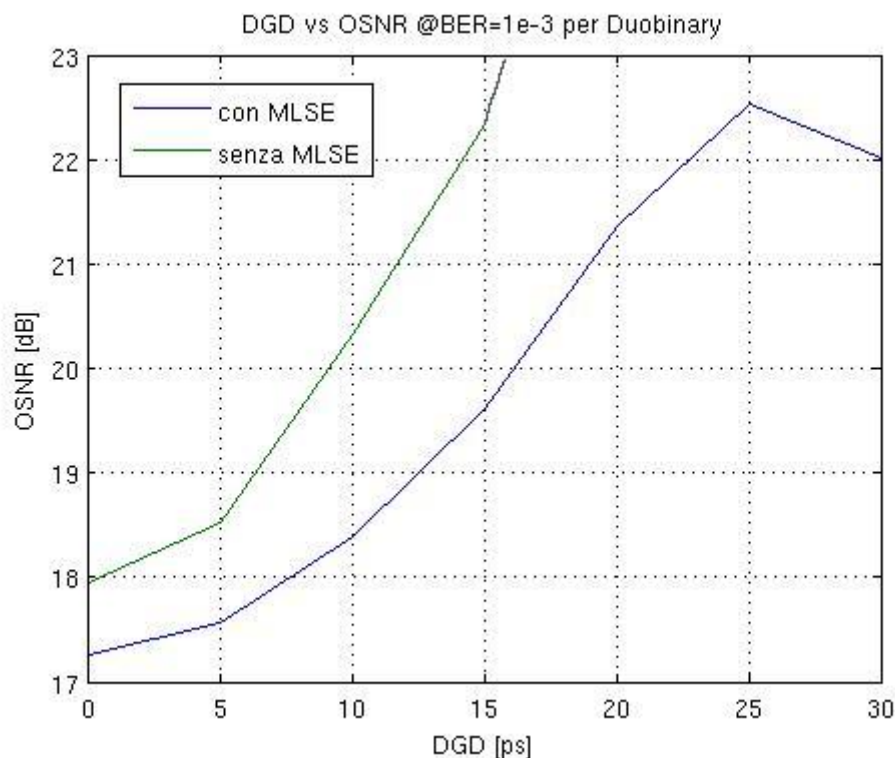


Figura 4.3. DGD vs OSNR @ $BER=10^{-3}$ per modulazione Duobinary

Il BER più basso ottenuto è pari a $0.80115e-4$ ed è quello corrispondente a $B_{TX}=0.2 \cdot R_b$ e $B_{RX}=2 \cdot R_b$.

Fissando le larghezze di banda nei valori ottimi, è stato ricalcolato l'andamento dell'OSNR richiesto per avere $BER=10^{-3}$ al variare del DGD. Questo andamento è stato confrontato con quello ottenuto con i filtri sub-ottimi. Il risultato viene mostrato nella figura 4.4, dove si può concludere che ottimizzando la larghezza di banda dei filtri elettrici in trasmissione e in ricezione si ha un guadagno pari a circa 1 dB lungo tutti i valori di DGD.

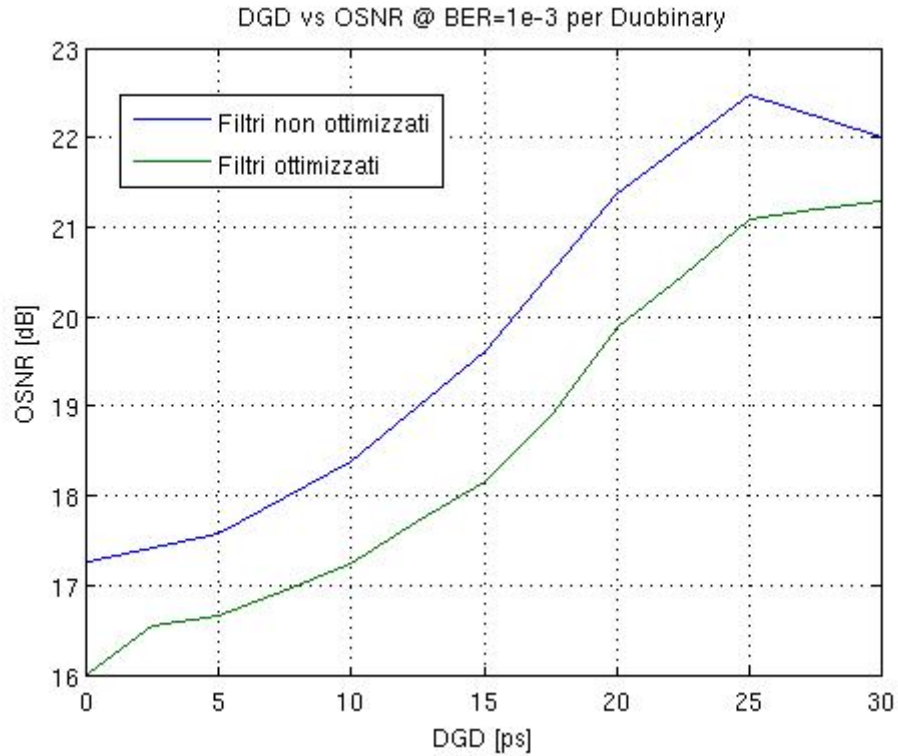


Figura 4.4. Confronto fra filtri ottimi e subottimi per Duobinary

4.3 Modulazione IMDD

Lo schema del sistema simulato in OptSim viene mostrato nella figura 4.5. Il driver mappa il livello alto come “5.0” e il livello basso come “0.0”. Questo segnale elettrico viene filtrato per poi entrare nel modulatore di ampiezza ideale del tipo “ $\sin^2$ ”.

Come primo approccio è stato calcolato l’andamento dell’OSNR richiesto al ricevitore per garantire $\text{BER}=10^{-3}$ al variare del Differential Group Delay (DGD) con e senza l’utilizzo del blocco MLSE per migliorare le prestazioni. Il risultato viene mostrato nella figura 4.6.

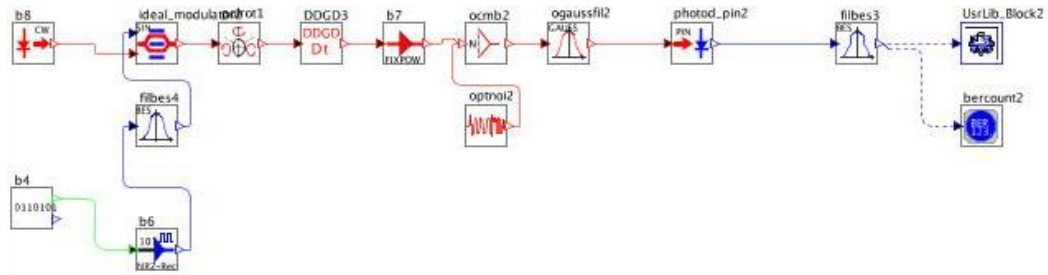


Figura 4.5. Schema del sistema per modulazione IMDD

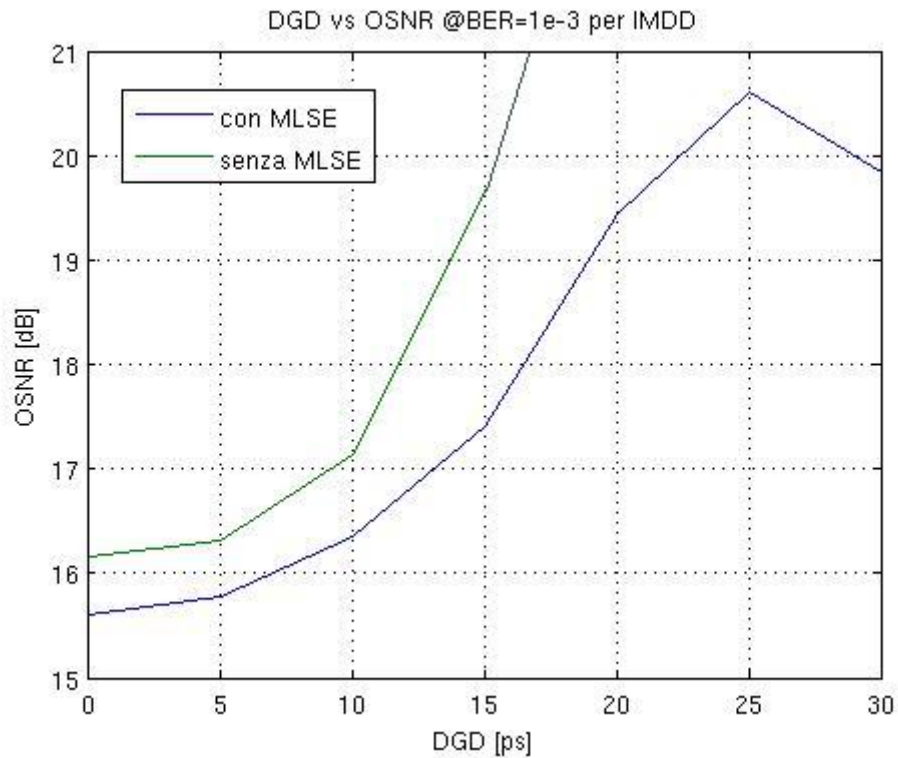


Figura 4.6. DGD vs OSNR @ $BER=10^{-3}$ per modulazione IMDD

Le considerazioni da fare sono le stesse del caso Duobinary. Il massimo di penalità si raggiunge per $DGD=25$ ps e questa penalità rispetto al caso con $DGD=0$ è di circa 5dB. Anche in questo caso, senza l'utilizzo del blocco MLSE il valore di DGD massimo tollerabile è pari a $DGD=15$ ps. Come ulteriore considerazione, l'andamento della curva è simile a quello ottenuto per

modulazione Duobinary ma “spostato” sotto di circa 1dB. Questo vuole dire che per modulazione IMDD l’OSNR richiesto per avere $BER=10^{-3}$ è sempre minore rispetto a quello richiesto per modulazione Duobinary.

Per cercare di migliorare le prestazioni, sono state ottimizzate le larghezze di banda dei filtri elettrici in trasmissione e in ricezione, lasciando inalterato il filtro ottico di ricezione. Per fare questo ci siamo posizionati nel caso $DGD=23.25$ ps e abbiamo fissato l’OSNR pari a quello richiesto per avere $BER=10^{-3}$ (20.2065 dB).

Le larghezze di banda dei filtri di trasmissione e ricezione sono state variate da $0.5 \cdot R_b$ a $2 \cdot R_b$ con passi da $0.25 \cdot R_b$ (il valore non ottimizzato era pari a $0.75 \cdot R_b$ per entrambi). E’ stata cercata la coppia (B_{TX} , B_{RX}) corrispondente al BER minore.

Il BER più basso ottenuto è pari a $0.82404e-4$ ed è quello corrispondente a **$B_{TX}=2 \cdot R_b$ e $B_{RX}=1.75 \cdot R_b$** .

Fissando le larghezze di banda nei valori ottimi, è stato ricalcolato l’andamento dell’OSNR richiesto per avere $BER=10^{-3}$ al variare del DGD. Questo andamento è stato confrontato con quello ottenuto con i filtri sub-ottimi. Il risultato viene mostrato nella figura 4.7, dove si può concludere che ottimizzando la larghezza di banda dei filtri elettrici in trasmissione e in ricezione si ha un guadagno pari a circa 0.5-0.6 dB lungo tutti i valori di DGD.

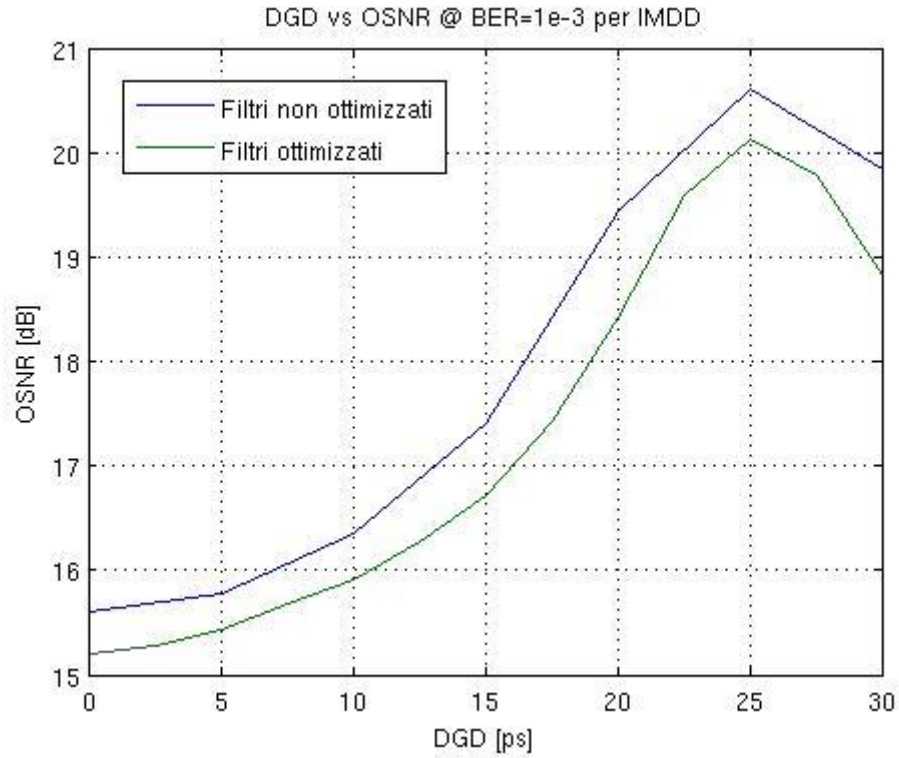


Figura 4.7. Confronto fra filtri ottimi e subottimi per IMDD

4.4 Modulazione DPSK

Lo schema del sistema simulato in OptSim viene mostrato nella figura 4.8.

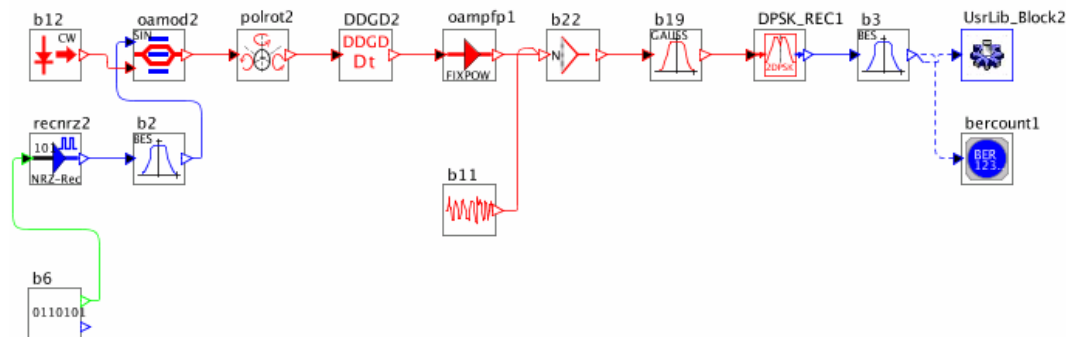


Figura 4.8. Schema del sistema per modulazione DPSK

Il driver mappa il livello alto come “5.0” e il livello basso come “-5.0”. Questo segnale elettrico viene filtrato per dopo entrare nel modulatore di ampiezza ideale del tipo “sin²”.

Come primo approccio è stato calcolato l’andamento dell’OSNR richiesto al ricevitore per garantire $BER=10^{-3}$ al variare del Differential Group Delay (DGD) con e senza l’utilizzo del blocco MLSE per migliorare le prestazioni. Il risultato viene mostrato nella figura 4.9.

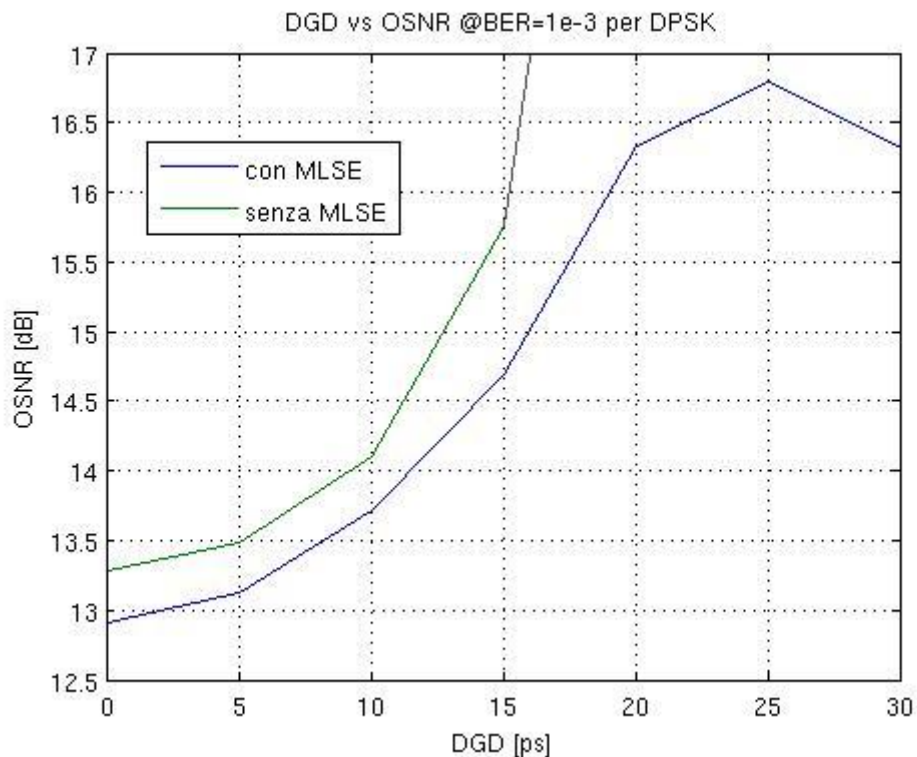


Figura 4.9. DGD vs OSNR @ $BER=10^{-3}$ per modulazione DPSK

Anche in questo caso l’andamento è molto simile a quello visto nei formati di modulazione precedenti, cioè per $DGD=25$ ps si ha la penalità massima ed è leggermente inferiore a 4 dB rispetto al caso in cui $DGD=0$. Anche per il DPSK, senza l’utilizzo del blocco MLSE il valore di DGD massimo tollerabile è pari a $DGD=15$ ps.

Per il formato di modulazione DPSK c'è un importante dettaglio da notare, e questo sta nel fatto che i valori di OSNR richiesti per avere $BER=10^{-3}$ per DPSK sono notevolmente inferiori rispetto a Duobinary (-4 dB circa) e IMDD (-3 dB circa).

Per cercare di migliorare le prestazioni, sono state ottimizzate le larghezze di banda dei filtri elettrici in trasmissione e in ricezione, lasciando inalterato il filtro ottico di ricezione. Per fare questo ci siamo posizionati nel caso $DGD=23.25$ ps e abbiamo fissato l'OSNR pari a quello richiesto per avere $BER=10^{-3}$ (16.6335 dB).

Le larghezze di banda dei filtri di trasmissione e ricezione sono state variate da $0.5 \cdot R_b$ a $2 \cdot R_b$ con passi da $0.25 \cdot R_b$ (il valore non ottimizzato era pari a $0.75 \cdot R_b$ per entrambi). E' stata cercata la coppia (B_{TX} , B_{RX}) corrispondente al BER minore.

Il BER più basso ottenuto è pari a $0.85838 \cdot 10^{-3}$ ed è quello corrispondente a **$B_{TX}=1.25 \cdot R_b$ e $B_{RX}=2 \cdot R_b$** .

Fissando le larghezze di banda nei valori ottimi, è stato ricalcolato l'andamento dell'OSNR richiesto per avere $BER=10^{-3}$ al variare del DGD. Questo andamento è stato confrontato con quello ottenuto con i filtri sub-ottimi. Il risultato viene mostrato nella figura 4.10, dove si può notare come una volta ottimizzate le larghezze di banda dei filtri elettrici in trasmissione e in ricezione si ha un guadagno minore rispetto alle modulazioni Duobinary e IMDD. Questo guadagno è pari a circa 0.2 dB lungo tutti i valori di DGD e cresce leggermente dopo $DGD=22.5$ ps.

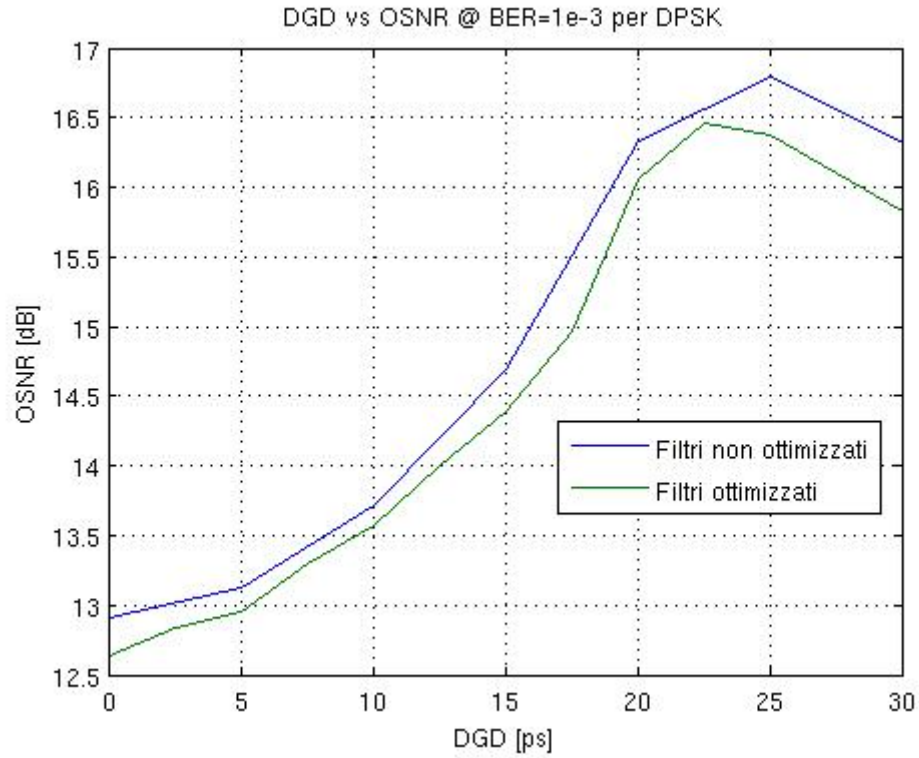


Figura 4.10. Confronto fra filtri ottimi e subottimi per DPSK

4.5 Modulazione DQPSK

Lo schema del sistema simulato in OptSim viene mostrato nella figura 4.11.

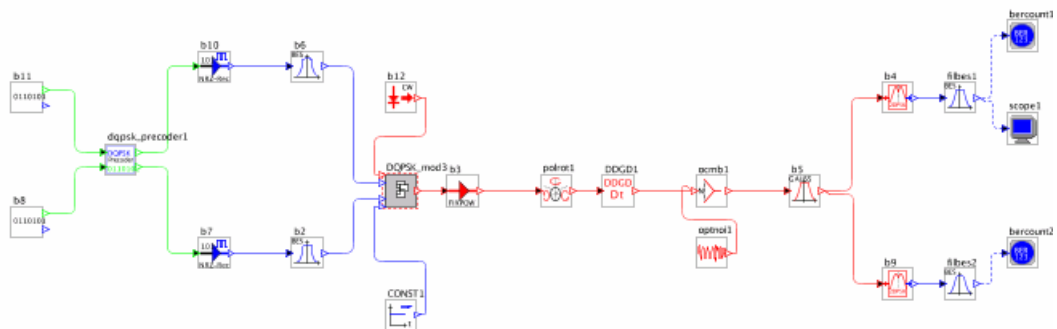


Figura 4.11. Schema del sistema per modulazione DQPSK

Il driver mappa i livelli alti come “5.0” e i livelli bassi come “-5.0”. Questi segnali elettrici vengono filtrati per dopo entrare nel modulatore DQPSK, il quale ha dentro di se una coppia di modulatori di tipo “ $\sin^2$ ” e aggiunge uno shift nella fase ottica del segnale a modo di creare la costellazione DQPSK.

Come primo approccio si è cercato di vedere quale è l’andamento dell’OSNR richiesto al ricevitore per garantire $\text{BER}=10^{-3}$ al variare del Differential Group Delay (DGD) con e senza l’utilizzo del blocco MLSE per migliorare le prestazioni. Il risultato viene mostrato nella figura 4.12.

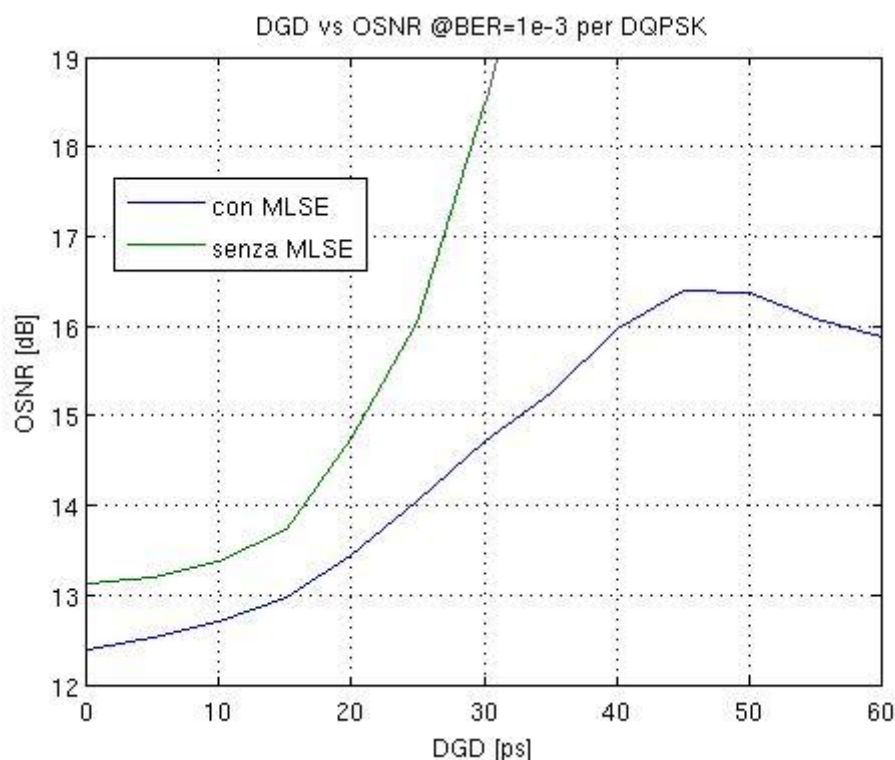


Figura 4.12. DGD vs OSNR @ $\text{BER}=10^{-3}$ per modulazione DQPSK

Il DGD si è stato variato da 0 a 60 ps. All’interno di questo “range” c’è il valore corrispondente a $T_s = 1/R_s = 46.5$ ps. Il massimo di penalità sull’OSNR rispetto al caso di $\text{DGD}=0$ si raggiunge quando il DGD prende valori vicini al tempo di simbolo. Si può anche notare come questa penalità è pari a 4dB circa. Senza

l'utilizzo del blocco MLSE questa penalità si raggiunge già per $DGD=30ps$ e andando oltre il segnale ricevuto viene deteriorato notevolmente.

Per cercare di migliorare le prestazioni, sono state ottimizzate le larghezze di banda dei filtri elettrici in trasmissione e in ricezione, lasciando inalterato il filtro ottico di ricezione. Per fare questo ci siamo posizionati nel caso $DGD=46.5 ps$ e abbiamo fissato l'OSNR pari a quello richiesto per avere $BER=10^{-3}$ (16.4679dB).

La larghezza di banda dei filtri di trasmissione è stata variata da $0.5 \cdot R_s$ a $2 \cdot R_s$ con passi da $0.25 \cdot R_s$ (il valore non ottimizzato era pari a $0.4186 \cdot R_s$), e la larghezza di banda dei filtri di ricezione è stata variata da $0.5 \cdot R_s$ a $2 \cdot R_s$ con passi da $0.25 \cdot R_s$ (il valore non ottimizzato era pari a $0.8372 \cdot R_s$). E' stata cercata la coppia (B_{TX} , B_{RX}) corrispondente al BER minore.

Il BER più basso ottenuto è pari a $7.57275e-4$ ed è quello corrispondente a **$B_{TX}=0.5 \cdot R_s$ e $B_{RX}=1.25 \cdot R_s$** .

Fissando le larghezze di banda nei valori ottimi, è stato ricalcolato l'andamento dell'OSNR richiesto per avere $BER=10^{-3}$ al variare del DGD. Questo andamento è stato confrontato con quello ottenuto con i filtri sub-ottimi. Il risultato viene mostrato nella figura 4.13, dove si può notare come il guadagno ottenuto in termini di OSNR richiesto dopo l'ottimizzazione dei filtri elettrici sta sempre attorno ai 0.2 dB. Con questo si può concludere che la penalità dovuta al fatto di non avere i filtri elettrici di trasmissione e ricezione con la larghezza ottima è minima, e queste larghezze di banda non influiscono molto nelle prestazioni del sistema.

Va anche notato che nel caso di $DGD=0$ il sistema con i filtri ottimizzati va leggermente peggio di quello con i filtri non ottimizzati. Questo può essere attribuito al fatto che l'ottimizzazione è stata effettuata per $DGD=T_s$ (massima penalità).

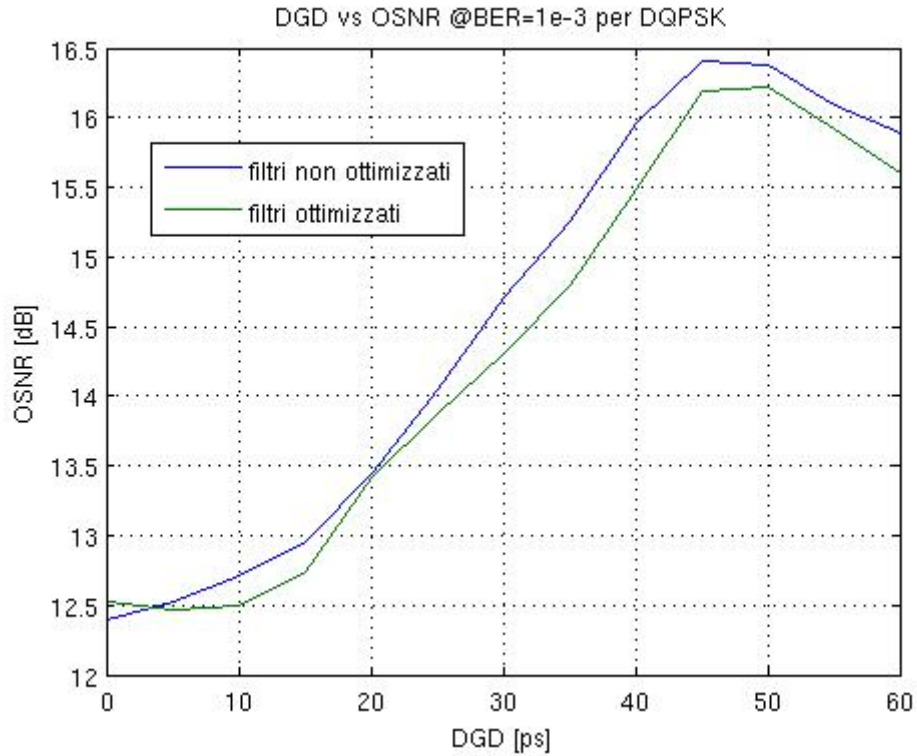


Figura 4.13. Confronto fra filtri ottimi e subottimi per DQPSK

4.6 Confronto fra i diversi formati

Le prestazioni del sistema variano dipendendo dal formato di modulazione. Per questo è conveniente provare con diversi formati (4 in questo caso), e confrontandoli si riesce a determinare quello che ha le prestazioni migliori. Si deve considerare anche che, siccome ogni formato di modulazione ha costi di implementazione diversi, bisogna scegliere il formato che abbia il miglior rapporto prestazioni-costi di implementazione.

Come primo confronto, si analizzano le prestazioni di ogni formato di modulazione senza l'utilizzo del MLSE. Il risultato viene mostrato nella figura 4.14.

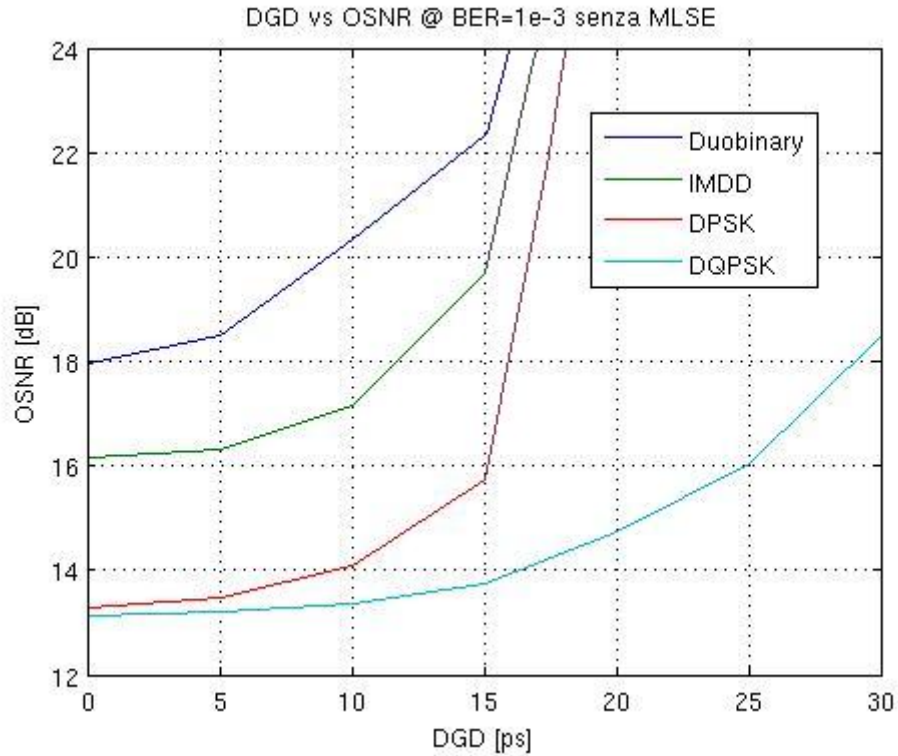


Figura 4.14. Confronto fra tutti i formati di modulazione senza MLSE

Si può notare come ogni formato studiato migliora le prestazioni di quello precedente. Così, il formato IMDD migliora di circa 2 dB le prestazioni di Duobinary, e a sua volta il formato DPSK migliora di quasi 3 dB le prestazioni di IMDD.

Risulta invece evidente come le prestazioni del DQPSK siano nettamente superiori rispetto agli altri formati. Questo si giustifica nel fatto che la penalità sull'OSNR richiesto per avere $BER=10^{-3}$ al variare del DGD ha il valore massimo quando quest'ultimo raggiunge valori vicini al tempo di simbolo. Per Duobinary, IMDD e DPSK il tempo di simbolo è 23.25 ps (ed è anche il tempo di bit); dall'altra parte per DQPSK il tempo di simbolo è di 46.5 ps il che significa che si ha una maggiore robustezza alla PMD. Questa robustezza è dovuta al fatto che il symbol-rate è stato dimezzato rispetto ai formati precedenti.

Mentre per i formati a 2 livelli (Duobinary, IMDD e DPSK) poteva essere tollerato un DGD fino a 15 ps senza subire penalità sull'OSNR superiori ai 5dB rispetto al caso in cui $DGD=0$, per DQPSK (4 livelli) può essere tollerato un DGD fino a 30 ps mantenendosi sotto i 5dB di penalità, e per $DGD=15ps$ questa penalità è inferiore a 1dB.

Va notato anche che per $DGD=0$ le prestazioni del DPSK sono praticamente uguali a quelle del DQPSK. Questo vuole dire che considerando i costi superiori di implementazione del DQPSK (trasmettitori e ricevitori di una complessità molto più elevata), questo formato conviene solo nel caso di avere forte presenza della PMD.

Adesso vengono confrontate le prestazioni di ogni formato di modulazione con l'utilizzo del MLSE e una volta effettuata l'ottimizzazione dei filtri elettrici in trasmissione e in ricezione. Il risultato viene mostrato nella figura 4.15.

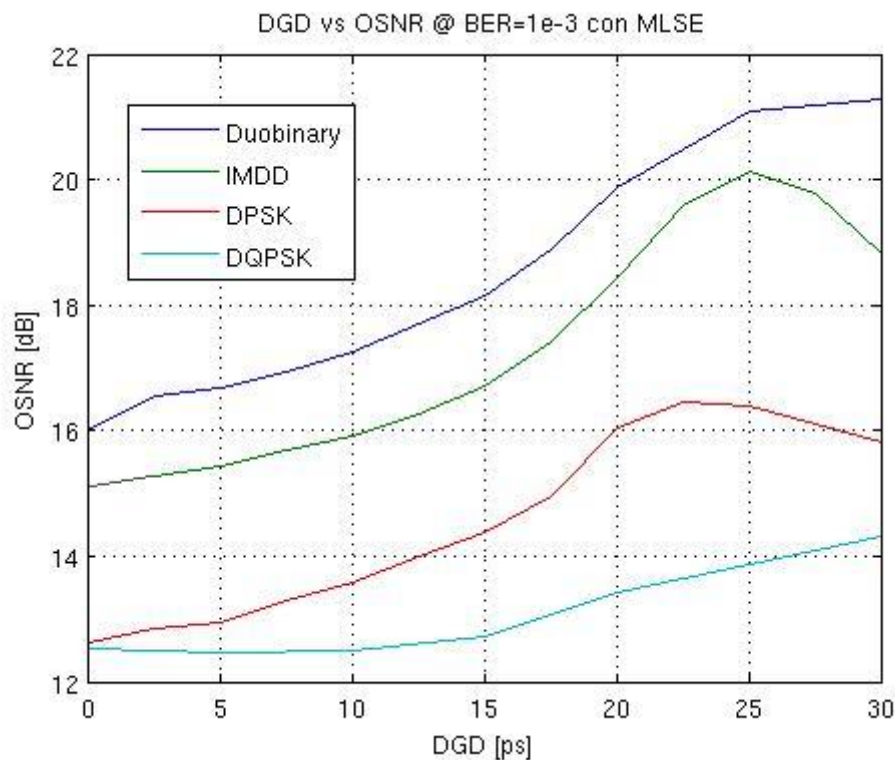


Figura 4.15. Confronto fra tutti i formati di modulazione con MLSE

Si può notare come anche con l'utilizzo del MLSE le prestazioni del DQPSK sono superiori a quelle degli altri formati, ma questa superiorità viene completamente giustificata in termini di robustezza alla PMD. Il DQPSK ha una maggiore robustezza alla PMD perché il symbol-rate è la metà di quello degli altri formati.

Nelle curve per i formati a 2 livelli (Duobinary, IMDD e DPSK), si raggiunge il massimo di penalità sull'OSNR quando la DGD prende valori vicini al tempo di simbolo (23.25 ps). Per DQPSK succede lo stesso, solo che in questo caso il tempo di simbolo è di 46.5 ps (fuori dall'area del grafico).

Anche in questo caso, per $DGD=0$ i formati DPSK e DQPSK hanno le stesse prestazioni. In realtà, la curva del DQPSK può essere vista come la stessa del DPSK ma "allargata" fino a 60 ps. Questo si può notare nel fatto che l'OSNR massimo richiesto è appena sopra i 16dB in entrambi i casi (guardando anche la figura 4.12), e anche nel fatto che il valore di OSNR richiesto in DPSK per $DGD=15$ ps è molto simile a quello richiesto in DQPSK per $DGD=30$ ps.

Infine, si può concludere che il formato DQPSK ha il vantaggio di avere una maggiore robustezza alla PMD e un symbol-rate dimezzato rispetto agli altri formati di modulazione (questo implica anche una occupazione spettrale ridotta), ma lo svantaggio che ha è quello di avere una complessità più elevata e quindi costi di implementazione superiori.

Capitolo 5

Prestazioni della decodifica MLSE basata sull'algoritmo BCJR in presenza della PMD

Nel capitolo 4 sono state analizzate le prestazioni della decodifica MLSE standard basata sull'algoritmo di Viterbi, utilizzando quattro diversi formati di modulazione. In questo capitolo queste prestazioni vengono confrontate con quelle ottenute utilizzando l'algoritmo BCJR per prendere la decisione sui bit ricevuti.

Anche in questo capitolo il sistema sotto studio prevede il collegamento diretto fra trasmettitore e ricevitore (back-to-back) e l'inserimento artificiale della PMD nella forma di DGD. In questo modo si simula un sistema di trasmissione in cui la dispersione è compensata otticamente al 100%. Nella prima parte viene presentata la struttura ed il funzionamento dell'algoritmo BCJR, e nel seguito verranno analizzate le prestazioni del sistema utilizzando quattro formati di modulazione diversi: Duobinary, IMDD, DPSK e DQPSK. Per ogni formato si parte dai valori di larghezza di banda dei filtri elettrici in trasmissione e in

ricezione ottimizzati, ottenuti nel capitolo 4. Verranno prima confrontate le prestazioni della decodifica MLSE standard (implementata nel software OptSim) con quella implementata usando il software MatLab, e posteriormente vengono confrontate la decodifica MLSE standard e la decodifica MLSE basata sull'algoritmo BCJR, anche quest'ultima implementata usando il software MatLab.

5.1 Algoritmo BCJR

L'algoritmo BCJR è stato presentato in [7] da L.R. Bahl, J. Cocke, F. Jelinek e J. Raviv nel 1974. Ha come scopo minimizzare la probabilità di errore sul bit (a differenza di Viterbi, il quale minimizza la probabilità di errore sulla sequenza).

Si consideri un collegamento in fibra ottica in cui si trasmette una sequenza di bit che viene chiamata u_I^N (sequenza di N bit), questi bit attraversano il canale e arrivano al ricevitore rappresentati dalla sequenza y_I^N .

Si assume che il canale attraverso il quale hanno viaggiato i bit ha una memoria v , quindi si ha dipendenza da v bit precedenti. Si costruisce il trellis con 2^v stati per ogni istante, dall'istante zero (0) fino a N (quando sono stati ricevuti tutti i bit).

Come risultato, si vuole avere un vettore $L(u_I^N)$ contenente i valori di affidabilità di ognuno dei bit decodificati.

Si definiscono i parametri $\alpha_j(s)$, $\beta_j(s)$ e $\gamma_j(s',s)$:

- $\alpha_j(s) = \text{Log } p(s_j=s, y_I^j)$;
- $\beta_j(s) = \text{Log } p(y_{j+1}^N / s_j=s)$;
- $\gamma_j(s',s) = \text{Log } p(s_j=s, y_j / s_{j-1}=s')$.

dove s_j denota lo stato in cui ci si trova all'istante j .

Per ogni transizione, ci saranno tanti $\gamma_j(s',s)$ quanti rami che collegano gli stati. Siccome da ogni stato escono 2 rami, ci saranno quindi 2^{v+1} valori di $\gamma_j(s',s)$ in ogni transizione.

Dall'altra parte, per ogni istante j ci saranno tanti $\alpha_j(s)$ e $\beta_j(s)$ come stati (ogni stato avrà la sua coppia di valori). Quindi ci saranno 2^v valori di $\alpha_j(s)$ e $\beta_j(s)$ per ogni istante.

Alla fine ci saranno 3 matrici *gamma*, *alfa* e *beta* contenenti tutti i rispettivi valori lungo il trellis. Le dimensioni di queste matrici sono:

- gamma [$2^{v+1} \times N$];
- alfa [$2^v \times (N+1)$];
- beta [$2^v \times (N+1)$].

Supponendo di conoscere tutti i $\gamma_j(s',s)$, gli $\alpha_j(s)$ e $\beta_j(s)$ vengono calcolati ricorsivamente:

- $\alpha_j(s) = \max_{s'} [\alpha_{j-1}(s') + \gamma_j(s',s)]$;
- $\beta_{j-1}(s') = \max_s [\beta_j(s) + \gamma_j(s',s)]$.

con $\max^*(x,y) = \max(x,y) + \text{Log}(1 + e^{|x-y|})$.

Per il calcolo degli $\alpha_j(s)$, vengono presi in considerazione tutti gli stati s' che portano allo stato s , fatte le diverse operazioni e scelto il valore massimo col criterio $\max^*$. Allo stesso modo per il calcolo dei $\beta_{j-1}(s')$ vengono presi in considerazione tutti gli stati s a cui si può arrivare dallo stato s' , fatte le diverse operazioni e scelto il valore massimo col criterio $\max^*$. Questi passi vengono mostrati nella figura 5.1.

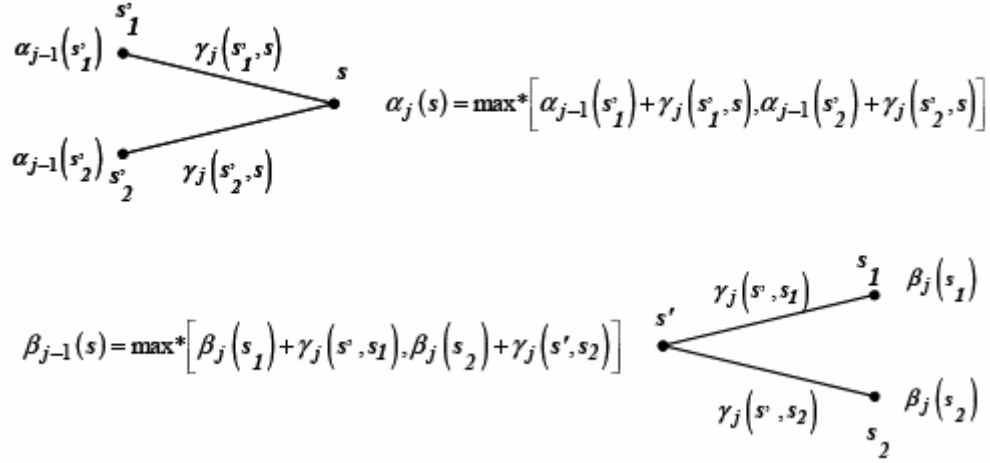


Figura 5.1. Transizioni “in avanti” per gli $\alpha_j(s)$ e “all’indietro” per i $\beta_{j-1}(s')$

Per come è fatto il trellis, ad ogni stato arrivano sempre 2 rami e ne escono altrettanti. Quindi per il calcolo degli $\alpha_j(s)$ e $\beta_j(s)$ si confrontano sempre 2 valori e si prende il $\max^*$.

Per poter fare il calcolo ricorsivo bisogna inizializzare i valori di α e β ($\alpha_0(s)$ e $\beta_N(s)$). Esistono 2 opzioni:

- 1) Se si è sicuri che lo stato iniziale è quello di tutti zeri (s_0) e che alla fine si arrivi anche allo stato finale di tutti zeri (sempre s_0), allora in quegli istanti ci si trova in quello stato con probabilità 1. Quindi:

- $\alpha_0(s) = \text{Log}(1) = 0$ per $s = s_0$
 $\text{Log}(0) = -\infty$ per $s \neq s_0$
- $\beta_N(s) = \text{Log}(1) = 0$ per $s = s_0$
 $\text{Log}(0) = -\infty$ per $s \neq s_0$

- 2) Supporre che tutti gli stati iniziali e finali siano equiprobabili con probabilità $1/2^v$ (1/no. di stati). Quindi:

- $\alpha_0(s) = \text{Log} (1/2^v) = -v \cdot \text{Log}(2)$ per tutti gli s
- $\beta_N(s) = \text{Log} (1/2^v) = -v \cdot \text{Log}(2)$ per tutti gli s

Ad ogni passo (per ogni bit y_j ricevuto) si possono calcolare i $\gamma_j(s',s)$ e gli $\alpha_j(s)$, ma bisogna aspettare fino alla fine (una volta ricevuto y_N) per poter calcolare i $\beta_{j-1}(s')$ all'indietro (questo perché $\beta_j(s)$ è stato inizializzato alla fine del trellis).

Una volta calcolati tutti i valori, si procede a calcolare l'affidabilità sul bit u_j trasmesso $L(u_j)$ così:

$$L(u_j) = \max_{(s',s) : u_j=0}^* [\alpha_{j-1}(s') + \gamma_j(s',s) + \beta_j(s)] - \max_{(s',s) : u_j=1}^* [\alpha_{j-1}(s') + \gamma_j(s',s) + \beta_j(s)]$$

In ogni istante, alla metà degli stati ci si arriva con uno 0 e all'altra metà ci si arriva con un 1. Ad ognuno di questi stati arrivano 2 stati precedenti, quindi per ogni operazione di $\max^*$ nel calcolo di $L(u_j)$ si devono confrontare tanti valori come il numero di stati, cioè 2^v .

In ogni $\max^*$ appaiono gli $\alpha_{j-1}(s')$ di tutti gli stati (da tutti gli stati si esce sempre con uno 0 o con un 1), mentre appaiono la metà dei $\gamma_j(s',s)$ (metà delle transizioni generate da zeri e metà da uni) e appaiono metà dei $\beta_j(s)$ ma 2 volte ciascuno (ci si arriva alla metà degli stati con uno 0 e all'altra metà con un 1, ma ad ogni stato ci si arriva da 2 stati precedenti diversi).

La decisione "hard" sul bit si prende così:

$$u_j = \begin{cases} 1 & \text{se } L(u_j) < 0 \\ 0 & \text{se } L(u_j) \geq 0 \end{cases}$$

Il funzionamento dell'algoritmo BCJR presentato, è valido per formati di modulazione binaria (a 2 livelli) come Duobinary, IMDD e DPSK. Per il formato

di modulazione DQPSK l'algoritmo BCJR segue un approccio leggermente diverso.

Il segnale DQPSK è un segnale a 4 livelli (2 bit/simbolo). La differenza principale fra il trellis di un segnale a 4 livelli rispetto a quello a 2 livelli è che se v è la memoria del canale, ci saranno 4^v stati (e non 2^v come per i segnali a 2 livelli). Inoltre, da ogni stato usciranno 4 rami (e non 2 come per i segnali a 2 livelli).

Per ogni transizione, ci saranno tanti $\gamma_j(s',s)$ come rami che comunicano gli stati. Siccome da ogni stato escono 4 rami, ci saranno quindi 4^{v+1} valori di $\gamma_j(s',s)$ in ogni transizione.

Dall'altra parte, per ogni istante j ci saranno tanti $\alpha_j(s)$ e $\beta_j(s)$ quanti sono gli stati (ogni stato avrà la sua coppia di valori). Quindi ci saranno 4^v valori di $\alpha_j(s)$ e $\beta_j(s)$ per ogni istante.

Alla fine ci saranno 3 matrici *gamma*, *alfa* e *beta* contenenti tutti i rispettivi valori lungo il trellis. Le dimensioni di queste matrici sono:

- *gamma* [$4^{v+1} \times N$];
- *alfa* [$4^v \times (N+1)$];
- *beta* [$4^v \times (N+1)$].

In questo modo, se si decidono di inizializzare i valori di α e β ($\alpha_0(s)$ e $\beta_N(s)$) supponendo che tutti gli stati iniziali e finali siano equiprobabili, allora la probabilità da prendere in considerazione è $1/4^v$ (1/no. di stati).

In ogni istante, si deve calcolare l'affidabilità di ognuno dei 4 livelli per poi scegliere l'affidabilità del livello "vincente" $L(u_j)$.

Per ogni livello del segnale ci si arriva alla quarta parte degli stati, ma ad ognuno di questi stati arrivano rami provenienti da 4 stati precedenti diversi. Quindi per ogni operazione di $\max^*$ nel calcolo di $L(u_j)$ si devono confrontare tanti valori come il numero di stati, cioè 4^v .

In ogni $\max^*$ appaiono gli $\alpha_{j-1}(s')$ di tutti gli stati (da tutti gli stati si esce sempre con qualsiasi livello), mentre appaiono la quarta parte dei $\gamma_j(s',s)$ (ogni quarta parte per ognuno dei 4 livelli) e appaiono la quarta parte dei $\beta_j(s)$ ma 4 volte ciascuno (ci si arriva alla quarta parte degli stati con ogni livello, ma ad ogni stato ci si arriva da 4 stati precedenti diversi).

La decisione “hard” sul livello si dovrebbe prendere scegliendo in ogni istante il valore di affidabilità più elevato dei 4.

L'algoritmo BCJR, sia per la versione a 2 livelli che per quella a 4 livelli, è stato implementato in MatLab come una “estensione” del MLSE standard. Questo dovuto al fatto che le fasi di sincronizzazione del segnale, elaborazione del trellis e calcolo delle metriche di ramo sono le stesse in ogni caso. Quindi il codice dell'algoritmo BCJR si separa da quello standard dopo aver calcolato la matrice *gamma*, cioè quella contenente le metriche di ramo. I codici da questo punto in avanti vengono presentati negli appendici A e B.

Per ridurre la complessità dell'algoritmo BCJR, è stata applicata la funzione *max* tradizionale al posto di $\max^*$, dato che si è verificato che il suo utilizzo non rappresenta una degradazione delle prestazioni, mentre i tempi di esecuzione vengono ridotti notevolmente, così come presentato in [8].

5.2 Caratteristiche del sistema

Le caratteristiche del sistema sotto studio e delle simulazioni effettuate sono le seguenti:

- Frequenza di lavoro: 194 THz (1545 nm).
- Bit-rate (R_b): 43 Gbps. Symbol-rate (R_s): 21.5 Gbaud (per DQPSK).
- Fotodiodo ideale (per Duobinary e IMDD). Ricevitore DPSK singolo (per DPSK) e doppio (per DQPSK).

- Rotatore di polarizzazione a 90° . Con questo la potenza si divide esattamente alla metà in ognuno dei PSP (caso peggiore).
- Densità spettrale di rumore ottico accomodata per avere l'OSNR richiesto su una banda pari a 0.1 nm.
- Filtro ottico di tipo supergaussiano con $B_{-3dB}=215$ GHz (per Duobinary, IMDD e DPSK) e $B_{-3dB}=50$ GHz (per DQPSK).
- Filtro elettrico di trasmissione *ottimizzato* di Bessel a 5 poli con $B_{-3dB}=8.6$ GHz ($0.2 \cdot R_b$, per Duobinary), $B_{-3dB}=86$ GHz ($2 \cdot R_b$, per IMDD), $B_{-3dB}=53.75$ GHz ($1.25 \cdot R_b$, per DPSK) e $B_{-3dB}=10.75$ GHz ($0.5 \cdot R_s$, per DQPSK).
- Filtro elettrico di ricezione *ottimizzato* di Bessel a 5 poli con $B_{-3dB}=86$ GHz ($2 \cdot R_b$, per Duobinary), $B_{-3dB}=73.25$ GHz ($1.75 \cdot R_b$, per IMDD), $B_{-3dB}=86$ GHz ($2 \cdot R_b$, per DPSK) e $B_{-3dB}=26.875$ GHz ($1.25 \cdot R_s$, per DQPSK).
- Sequenza pseudo-casuale o *Pseudo-Random Bit Sequence* (PRBS) di ordine 18. Numero di bits presi in considerazione per l'analisi MLSE pari a 2^{18} (262144 bits). Numero di campioni per bit pari a 24 (per Duobinary, IMDD e DPSK) e 20 (per DQPSK).
- Nel caso di Duobinary, DPSK e DQPSK la PRBS viene opportunamente *precodificata*.
- Numero di campioni per bit utilizzati dal MLSE pari a 4 (per Duobinary, IMDD e DPSK) e 2 (per DQPSK). Non viene realizzata l'ottimizzazione dell'istante di campionamento.
- Numero di stati utilizzati dall'algoritmo di Viterbi pari a 64. Corrispondenti ad una memoria del canale pari a 6 bits (per Duobinary, IMDD e DPSK) e 3 simboli (per DQPSK).

- Le prestazioni del sistema vengono misurate attraverso l'OSNR richiesto per avere $\text{BER}=10^{-3}$, il quale assicura il funzionamento con l'utilizzo dei FEC.

Nelle simulazioni, è stato ottenuto il segnale elettrico e filtrato all'uscita del ricevitore per diversi valori della coppia (OSNR, DGD). Questo segnale viene memorizzato per poi essere caricato ed utilizzato dai programmi MatLab che simulano il ricevitore MLSE standard e quello basato sull'algoritmo BCJR, ottenendo in ogni caso un valore di BER pari al rapporto fra il numero di errori ed il numero di bit trasmessi (questo rapporto viene valutato confrontando la sequenza trasmessa con quella ottenuta dopo la decisione). Questi valori di BER vengono interpolati linearmente (si calcola la funzione $\log(-\log(\text{BER}))$ essendo la più simile ad una retta) per trovare per ogni valore di DGD il valore di OSNR necessario per avere $\text{BER}=10^{-3}$. I parametri riguardanti l'MLSE sono gli stessi di quelli utilizzati nel capitolo 4, a modo di essere uguali a quelli che sono stati dati in ingresso al ricevitore MLSE del software OptSim.

5.3 Modulazione Duobinary

Lo schema del sistema simulato in OptSim viene mostrato nella figura 5.2. Il driver mappa il livello alto come “5.0” e il livello basso come “-5.0”. Questo segnale elettrico viene filtrato per “allargare” gli impulsi prima di entrare nel modulatore di ampiezza ideale del tipo “ $\sin^2$ ”.

Come primo approccio si confrontano le curve DGD vs OSNR @ $\text{BER}=10^{-3}$ ottenute con il ricevitore MLSE standard del software OptSim e con il ricevitore MLSE standard del software MatLab. Questo viene fatto con lo scopo di determinare se esistono differenze fra le due “versioni” dello stesso ricevitore. I risultati vengono mostrati nella figura 5.3.

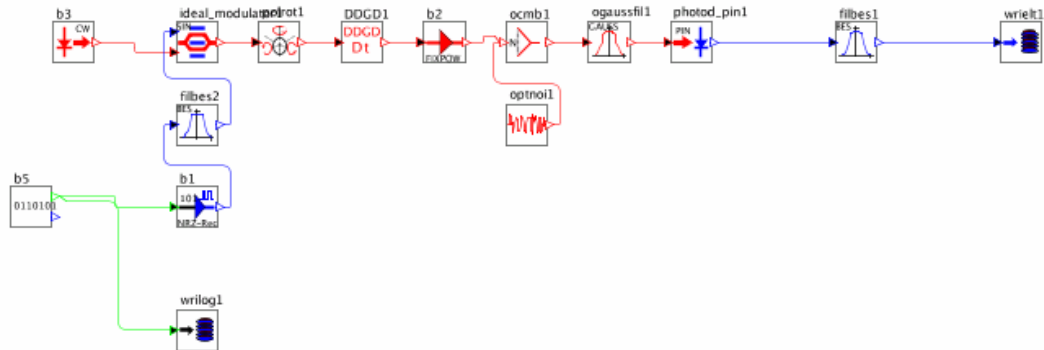


Figura 5.2. Schema del sistema per modulazione Duobinary

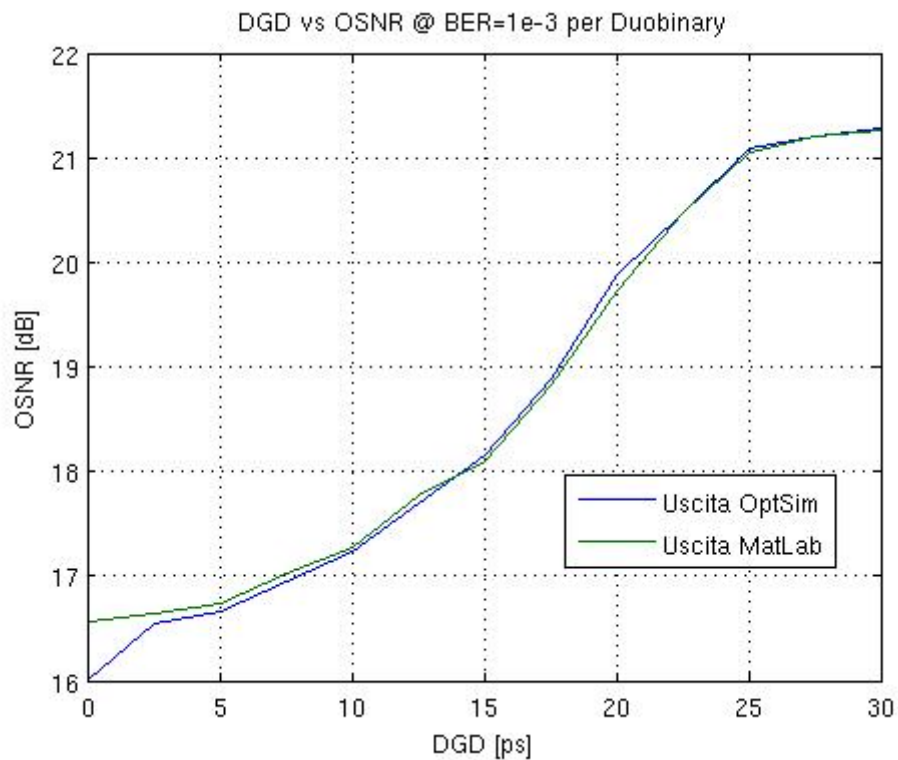


Figura 5.3. Confronto fra OptSim e MatLab per modulazione Duobinary

Si può notare come l'andamento delle due curve sia praticamente uguale, quindi si verifica che elaborando il segnale con qualsiasi dei due ricevitori MLSE (OptSim o MatLab) il risultato è lo stesso. Questo risultato era da aspettarsi dal fatto che il ricevitore MLSE di OptSim è stato elaborato a partire del codice scritto in MatLab

e ottimizzato in modo da ridurre i tempi di esecuzione e di avere il risultato direttamente dentro il simulatore, senza la necessità di memorizzare il segnale ricevuto ed elaborarlo posteriormente.

Adesso vengono confrontate le prestazioni del programma MLSE in MatLab esistente (che applica l'algoritmo di Viterbi) con quello che applica l'algoritmo BCJR. Una volta elaborato il segnale con il programma BCJR è stata ottenuta la curva DGD vs OSNR @ BER=10⁻³ e confrontata con quella ottenuta prima utilizzando il programma MLSE standard. Il risultato viene mostrato nella figura 5.4.

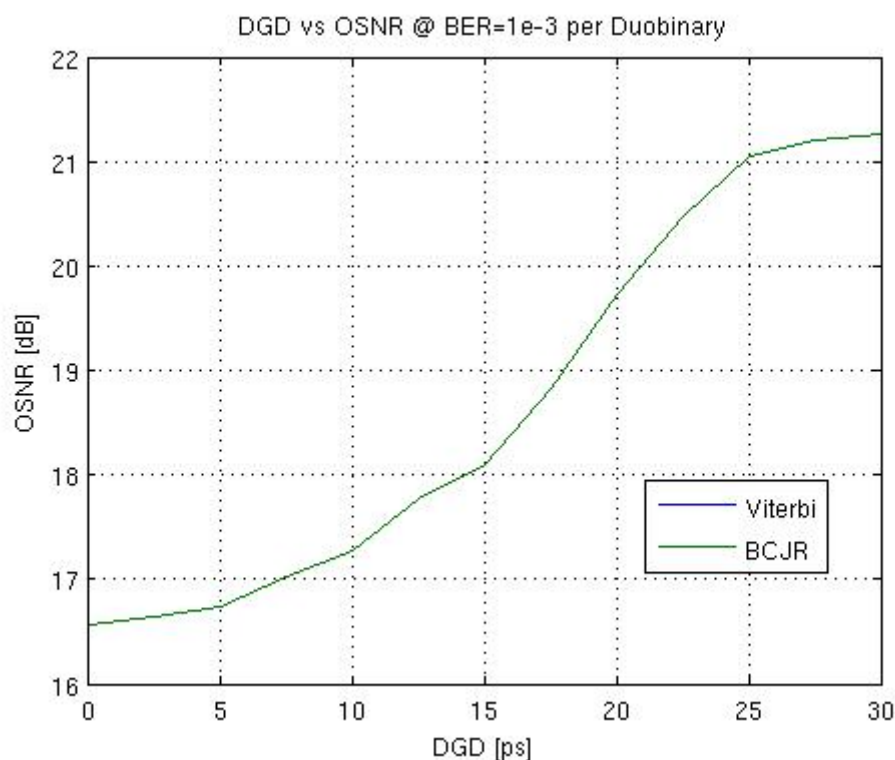


Figura 5.4. Confronto fra Viterbi e BCJR per modulazione Duobinary

Si può notare come le due curve coincidono perfettamente, questo significa che entrambi gli algoritmi (applicati sotto queste condizioni) danno lo stesso risultato, cioè prendono quasi le stesse decisioni su tutti i bit (il BCJR fa 1 o 2 errori in

meno, ma non si riesce a vedere nella grafica). Quindi, si conclude che l'elaborazione del segnale con l'algoritmo BCJR non riporta nessun vantaggio diretto rispetto ai risultati ottenuti precedentemente lavorando con l'algoritmo di Viterbi.

5.4 Modulazione IMDD

Lo schema del sistema simulato in OptSim viene mostrato nella figura 5.5. Il driver mappa il livello alto come “5.0” e il livello basso come “0.0”. Questo segnale elettrico viene filtrato per poi entrare nel modulatore di ampiezza ideale del tipo “sin²”.

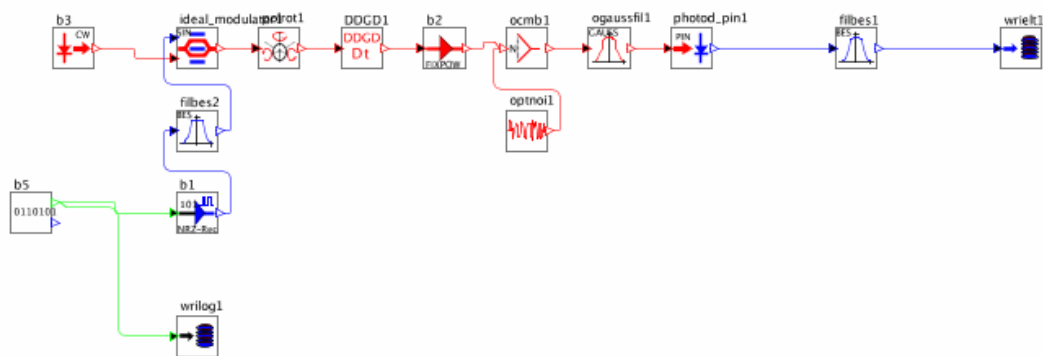


Figura 5.5. Schema del sistema per modulazione IMDD

Come primo approccio si confrontano le curve DGD vs OSNR @ BER= 10^{-3} ottenute con il ricevitore MLSE standard del software OptSim e con il ricevitore MLSE standard del software MatLab. Questo viene fatto con lo scopo di determinare se esistono differenze fra le due “versioni” dello stesso ricevitore. I risultati vengono mostrati nella figura 5.6. Si verifica che l'andamento delle due curve è abbastanza simile, e si interpreta che elaborando il segnale con qualsiasi dei due ricevitori MLSE (OptSim o MatLab) il risultato è lo stesso.

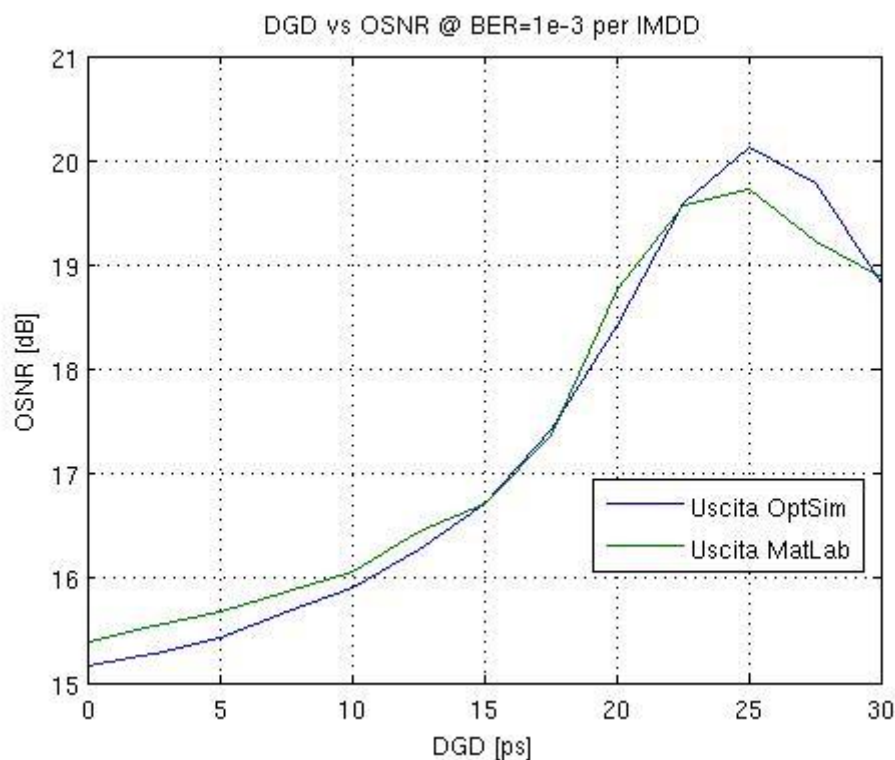


Figura 5.6. Confronto fra OptSim e MatLab per modulazione IMDD

Adesso vengono confrontate le prestazioni del programma MLSE in MatLab esistente (che applica l'algoritmo di Viterbi) con quello che applica l'algoritmo BCJR. Una volta elaborato il segnale con il programma BCJR è stata ottenuta la curva DGD vs OSNR @ BER=10⁻³ e confrontata con quella ottenuta prima utilizzando il programma MLSE standard. Il risultato viene mostrato nella figura 5.7. Anche in questo caso (così come per modulazione Duobinary) si può notare come le due curve coincidono perfettamente, questo significa che entrambi gli algoritmi (applicati sotto queste condizioni) danno lo stesso risultato, cioè prendono le stesse decisioni su tutti i bit. Quindi, si conclude che l'elaborazione del segnale con l'algoritmo BCJR non riporta nessun vantaggio diretto rispetto ai risultati ottenuti precedentemente lavorando con l'algoritmo di Viterbi.

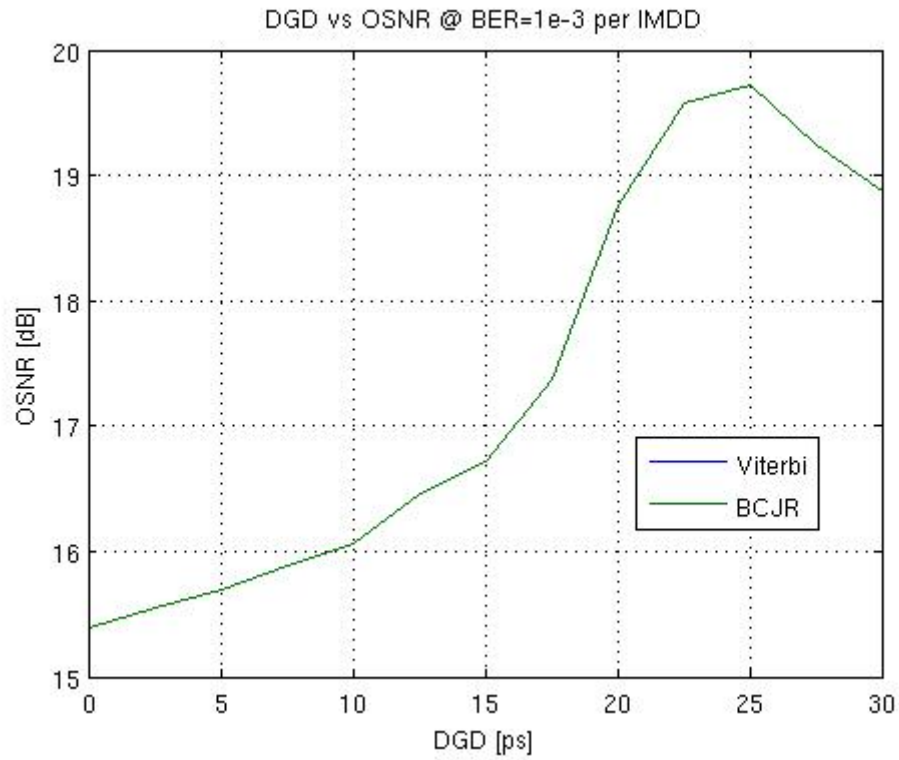


Figura 5.7. Confronto fra Viterbi e BCJR per modulazione IMDD

5.5 Modulazione DPSK

Lo schema del sistema simulato in OptSim viene mostrato nella figura 5.8.

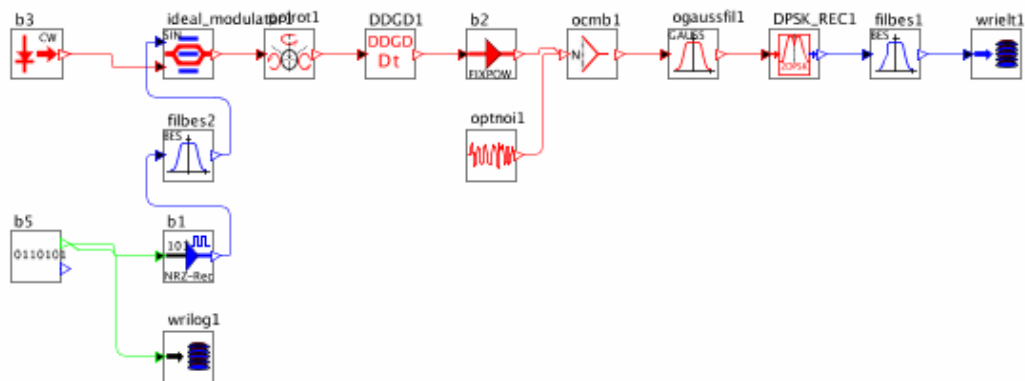


Figura 5.8. Schema del sistema per modulazione DPSK

Il driver mappa il livello alto come “5.0” e il livello basso come “-5.0”. Questo segnale elettrico viene filtrato per poi entrare nel modulatore di ampiezza ideale del tipo “sin²”.

Come primo approccio si confrontano le curve DGD vs OSNR @ BER=10⁻³ ottenute con il ricevitore MLSE standard del software OptSim e con il ricevitore MLSE standard del software MatLab. Questo viene fatto con lo scopo di determinare se esistono differenze fra le due “versioni” dello stesso ricevitore. I risultati vengono mostrati nella figura 5.9.

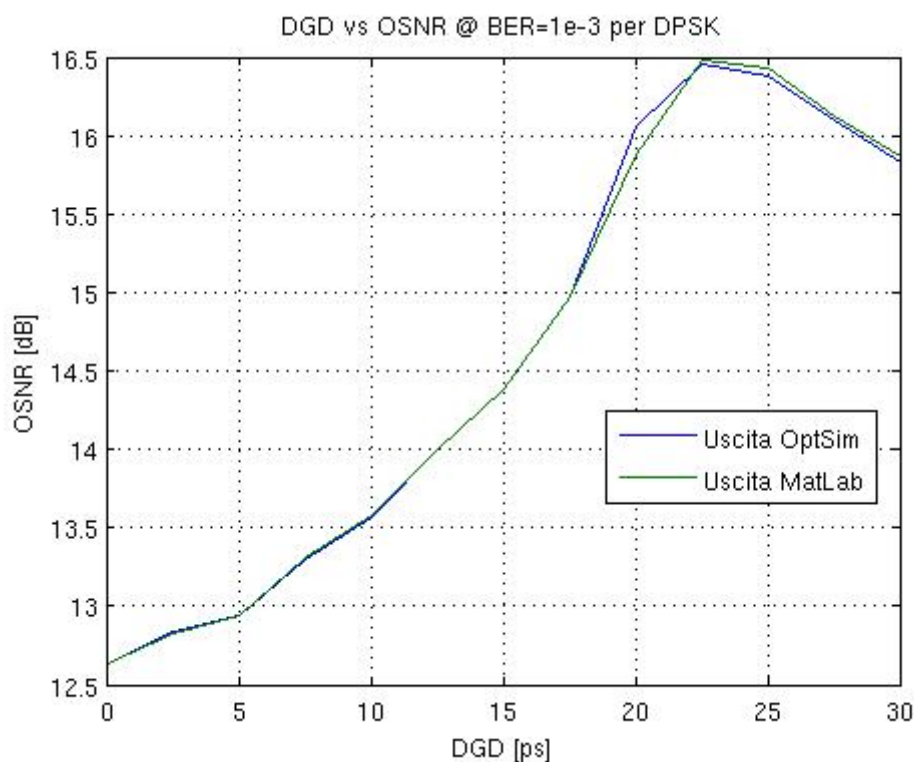


Figura 5.9. Confronto fra OptSim e MatLab per modulazione DPSK

Si verifica che le due curve coincidono quasi perfettamente, quindi si conclude che entrambi i ricevitori MLSE (sia quello OptSim che quello MatLab) lavorano in modo uguale e danno gli stessi risultati.

Adesso vengono confrontate le prestazioni del programma MLSE in MatLab esistente (che applica l'algoritmo di Viterbi) con quello che applica l'algoritmo BCJR. Una volta elaborato il segnale con il programma BCJR è stata ottenuta la curva DGD vs OSNR @ BER=10⁻³ e confrontata con quella ottenuta prima utilizzando il programma MLSE standard. Il risultato viene mostrato nella figura 5.10.

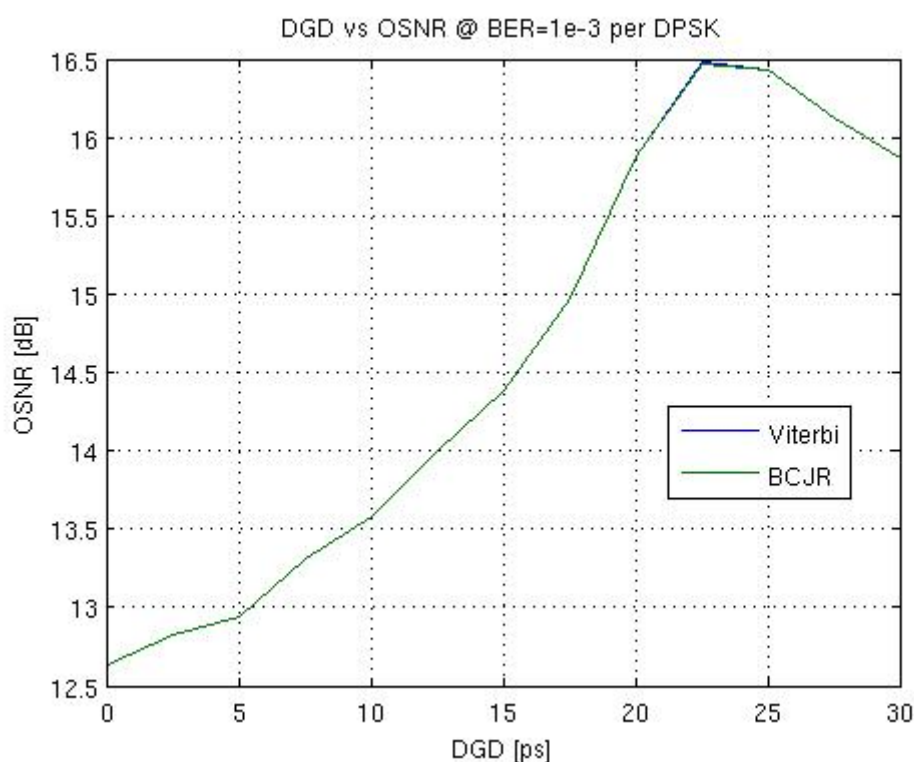


Figura 5.10. Confronto fra Viterbi e BCJR per modulazione DPSK

Ancora una volta (dopo i risultati ottenuti per Duobinary e IMDD) si può notare come le 2 curve coincidono perfettamente, il che significa che sia l'algoritmo di Viterbi che l'algoritmo BCJR (lavorando sotto queste condizioni di sistema) funzionano in maniera uguale, cioè decidono per gli stessi bit. Quindi si conclude che l'implementazione dell'algoritmo BCJR non riporta nessun vantaggio rispetto all'algoritmo di Viterbi.

5.6 Modulazione DQPSK

Lo schema del sistema simulato in OptSim viene mostrato nella figura 5.11.

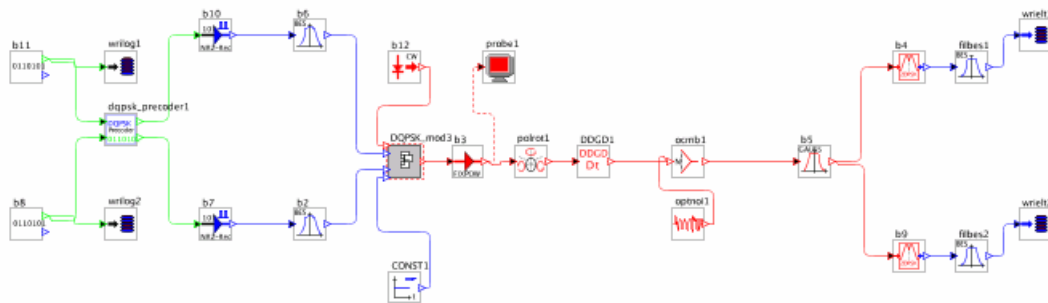


Figura 5.11. Schema del sistema per modulazione DQPSK

Il driver mappa i livelli alti come “5.0” e i livelli bassi come “-5.0”. Questi segnali elettrici vengono filtrati per dopo entrare nel modulatore DQPSK, il quale ha dentro di sé una coppia di modulatori di tipo “ $\sin^2$ ” e aggiunge uno shift nella fase ottica del segnale a modo di creare la costellazione DQPSK.

Per il formato di modulazione DQPSK, non esiste un ricevitore MLSE nel software OptSim, questo significa che le prestazioni ottenute per questo formato nel capitolo 4 sono state ottenute utilizzando il ricevitore MLSE del software MatLab.

A differenza degli altri formati, per il DQPSK non si fa il confronto fra i ricevitori OptSim e MatLab. Quindi vengono direttamente confrontate le prestazioni del programma MLSE in MatLab esistente (che applica l'algoritmo di Viterbi) con quello che applica l'algoritmo BCJR. Una volta elaborato il segnale con il programma BCJR è stata ottenuta la curva DGD vs OSNR @ BER= 10^{-3} e confrontata con quella ottenuta prima utilizzando il programma MLSE standard. Il risultato viene mostrato nella figura 5.12.

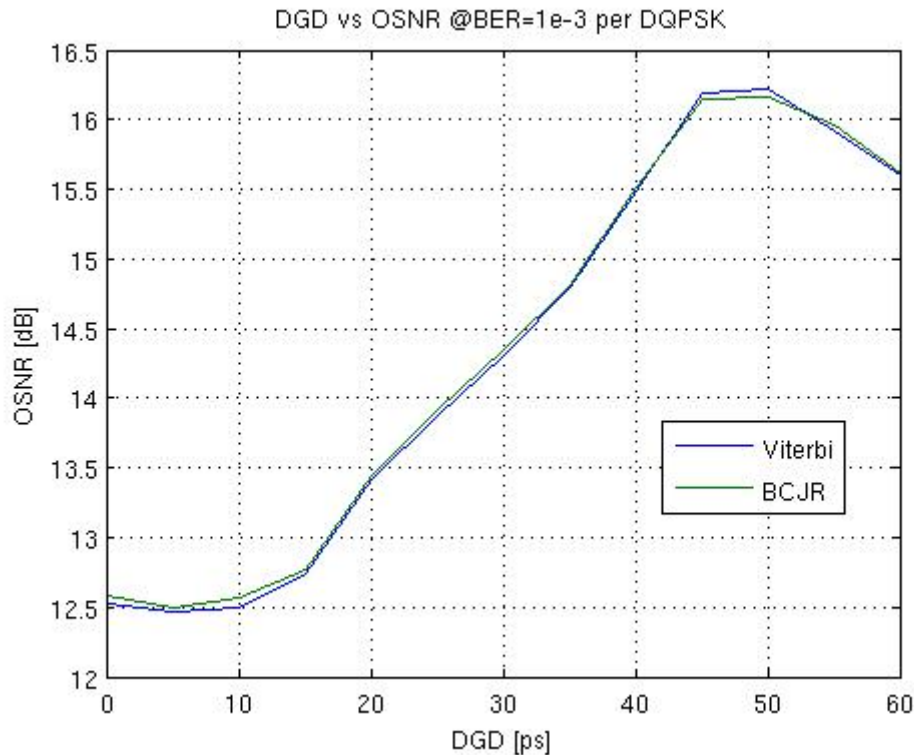


Figura 5.12. Confronto fra Viterbi e BCJR per modulazione DQPSK

Si può notare come l'andamento delle due curve è quasi uguale. La curva di Viterbi va leggermente meglio da DGD=0 fino a DGD=30 ps circa, mentre da lì fino a DGD=60 ps è quella di BCJR che va leggermente meglio. Quindi, si conclude che l'elaborazione del segnale con l'algoritmo BCJR non riporta nessun vantaggio diretto rispetto ai risultati ottenuti precedentemente lavorando con l'algoritmo di Viterbi.

Rimane comunque da studiare il fatto che l'uscita *soft* dell'algoritmo BCJR possa rivelarsi utile modificando altri componenti del sistema. Per esempio, tutte queste prestazioni sono state calcolate per un $\text{BER}=10^{-3}$, assumendo che dopo il ricevitore MLSE ci sia un decodificatore FEC il quale porta il BER da 10^{-3} a 10^{-12} (livello di BER richiesto in quasi tutti i sistemi in fibra ottica). Un codice a correzione di errore con prestazioni di questo tipo lavora tipicamente con l'uscita *hard* del ricevitore MLSE precedente, e siccome l'uscita *hard* di entrambi gli

algoritmi (Viterbi e BCJR) è la stessa, allora l'utilizzo del BCJR non riporta nessun vantaggio nelle prestazioni globali.

Il codice FEC che viene utilizzato in questo sistema lavora all'7.5% di ridondanza (il bit-rate si porta da 40 Gbps a 43 Gbps). Esistono però i codici FEC di terza generazione [9], basati su i codici *Low-Density Parity-Check* (LDPC), i quali lavorano anche con l'uscita *soft* e che per raggiungere in uscita un BER pari a 10^{-12} non richiedono di avere in ingresso un BER pari a 10^{-3} ma gli basterebbe avere un BER un pò superiore ($5 \cdot 10^{-3}$ per esempio). Avere bisogno di un BER più alto significa accettare un OSNR più basso e quindi si può predire che con l'utilizzo del algoritmo BCJR e l'aggiunta di un codice LDPC a ridondanza opportuna (7.5%) si migliorerebbero le prestazioni del sistema.

Capitolo 6

Confronto fra le prestazioni degli algoritmi di Viterbi e BCJR con l'utilizzo dei codici FEC opportuni

Precedentemente sono state studiate le prestazioni dell'MLSE per diversi formati di modulazione (Duobinary, IMDD, DPSK e DQPSK). Nei sistemi di telecomunicazioni in fibra ottica, il collegamento deve operare con una probabilità di errore pari a 10^{-12} come valore standard. Per raggiungere questo valore, vengono utilizzati i codici a correzione di errori o *Forward Error Correction* (FEC). Questi codici raggiungono il $\text{BER}=10^{-12}$ a patto che la sequenza che ricevono in ingresso abbia un BER non superiore ad una certa soglia. Il valore del BER di riferimento varia dipendendo del tipo di FEC utilizzato.

Negli studi precedenti sono state analizzate le prestazioni dell'MLSE considerando l'utilizzo degli algoritmi di Viterbi e BCJR. In entrambi i casi, il BER di riferimento è stato fissato pari a $\text{BER}=10^{-3}$, assumendo che questa sequenza fosse elaborata posteriormente con un FEC opportuno a modo di portare il BER fino a 10^{-12} . Le curve di DGD vs OSNR ottenute sono state praticamente le

stesse per entrambi gli algoritmi, indipendentemente dal formato di modulazione studiato.

In questo capitolo verranno prima presentate le caratteristiche e la struttura dei codici FEC, per dopo scegliere i codici che più si adattano alle caratteristiche del sistema sotto studio e degli algoritmi di decodifica a disposizione (Viterbi e BCJR). Infine, verranno presentati i risultati ottenuti dal confronto fra questi due algoritmi per quattro diversi formati di modulazione (Duobinary, IMDD, DPSK e DQPSK).

6.1 Caratteristiche dei codici FEC

I codici FEC sono stati progettati per i sistemi *wireless* per sopperire a deterioramenti del segnale, generalmente associati ad ostacoli, pioggia, etc. In seguito sono stati introdotti nei sistemi ottici a *Wavelength Division Multiplexing* (WDM) per sopperire al rumore introdotto dagli amplificatori ottici detto *Amplified Spontaneous Emission* (ASE). Questa *prima generazione* di codici FEC aveva un guadagno in termini di OSNR richiesto al ricevitore di circa 6 dB.

Succesivamente, con l'incremento delle velocità di trasmissione fino a 10 Gbps, sono entrati in scena altri fenomeni di natura distorsiva come gli effetti non lineari, la dispersione cromatica e la PMD. Per sopperire a questi fenomeni è nata la *seconda generazione* di codici FEC, con un guadagno associato di circa 8 dB [10].

Con la crescita continua delle dimensioni dei sistemi e delle velocità di trasmissione fino a 40 Gbps, i costi di implementazione hanno costretto l'imposizione della condizione di *trasparenza ottica*, il che consiste nell'eliminazione delle conversioni optoelettroniche. Così, per permettere l'implementazione di questi sistemi, è nata la *terza generazione* di codici FEC, i quali raggiungono guadagni superiori ai 10 dB.

Il primo parametro che caratterizza un codice FEC è la coppia (n,k) , dove n rappresenta la lunghezza del messaggio codificato e k rappresenta la lunghezza dell'informazione. La codifica FEC consiste in aggiungere dei bit di parità ai bit di informazione, generando delle *parole di codice*. La codifica assicura che tutte le parole di codice differiscono di un minimo numero di bit, questo numero corrisponde alla *distanza minima* del codice (d_{min}).

Nel trasmettitore, la velocità di trasmissione viene incrementata da un fattore $R=1/r$, dove $r=k/n$ è il “rate” del codice, o *code-rate*.

Nel ricevitore vengono effettuate due operazioni: la rivelazione degli errori (verificando la relazione fra i bit di informazione e i bit di parità) e la correzione degli stessi. Generalmente, un valore di r minore rappresenta un potenziale maggiore per la correzione degli errori, ma anche incrementa i costi di implementazione dovuto al maggiore incremento del bit-rate.

Esistono diverse distinzioni fra i codici FEC, ad esempio:

- *Outband e Inband*: i codici outband sono quelli tradizionali in cui il bit-rate deve essere incrementato in maniera coerente con il code-rate, mentre i codici inband sono quelli in cui il segnale contiene dei bit previamente destinati alla parità e vengono utilizzati dal codice senza l'incremento del bit-rate.
- *Decisioni Hard e Soft*: quando il ricevitore decide direttamente il valore del bit (se “0” o “1”) questa decisione viene chiamata hard. Se invece il ricevitore distingue valori intermedi fra “0” e “1” allora la decisione viene chiamata soft. I ricevitori a decisione soft hanno prestazioni superiori a quelli a decisione hard, ma anche costi e difficoltà di implementazione superiori dovuti alle alte velocità di trasmissione.
- *Blocco e Convolutionali*: i codici blocco sono quelle che hanno una quantità finita di parole di codice di lunghezza fissa, mentre i codici convoluzionali lavorano su una sequenza di bit teoricamente infinita, aggiungendo della ridondanza che dipende dalla *memoria* del codice.

6.2 Codici FEC per il sistema sotto studio

I FEC si basano sulla ridondanza dell'informazione trasmessa. Ogni FEC ha una percentuale di ridondanza di riferimento, la quale rappresenta la percentuale in cui deve essere incrementato il bit-rate per garantire il buon funzionamento del FEC. In questo caso si studiano collegamenti con un bit-rate pari a $R_b=43$ Gbps. Questo bit-rate in realtà corrisponde a 40 Gbps (rate standard nei sistemi in fibra ottica) con una ridondanza del 7.5% compresa. Per questo, si devono usare FEC che abbiano una ridondanza simile al 7.5% per cui è stato studiato il sistema.

L'algoritmo di Viterbi ha una uscita di tipo *hard*, la quale consiste nei bit che formano la sequenza più simile a quella trasmessa. Dall'altra parte, l'algoritmo BCJR ha una uscita di tipo *soft*, la quale aggiunge anche l'affidabilità corrispondente ad ogni bit della sequenza scelta. Per funzionare in maniera adeguata, il FEC che si usa per Viterbi deve lavorare con l'uscita *hard*, e dallo stesso modo il FEC utilizzato per BCJR deve lavorare con l'uscita *soft*. Naturalmente, i FEC che lavorano con uscita *soft* hanno una complessità maggiore, ma in controparte hanno un BER di riferimento maggiore a quello dei FEC che lavorano con uscita *hard*.

Sono stati cercati dei FEC che siano compatibili con le caratteristiche del sistema e dell'uscita di ogni algoritmo. Quelli scelti sono:

- Viterbi: FEC di seconda generazione standard. BCH(3860,3824)+BCH(2040,1930). Overhead 6.7%. BER di riferimento 2×10^{-3} , decodifica *hard* iterativa.
- BCJR: FEC di terza generazione (in fase di standardizzazione). LDPC(7225,6720). Overhead 7.5%. BER di riferimento 5×10^{-3} , decodifica *soft* iterativa.

6.3 Confronto fra Viterbi e BCJR

Utilizzando i codici FEC scelti, e supponendo che il BER in uscita da entrambi sia lo stesso, si devono confrontare le curve DGD vs OSNR di Viterbi (@BER= 2×10^{-3}) e di BCJR (@BER= 5×10^{-3}) per ogni formato di modulazione. In questo modo può essere notato il vantaggio che ha l'utilizzo del BCJR nei confronti di Viterbi.

I risultati vengono mostrati nelle figure 6.1, 6.2, 6.3 e 6.4, corrispondenti ai formati di modulazione Duobinary, IMDD, DPSK e DQPSK rispettivamente.

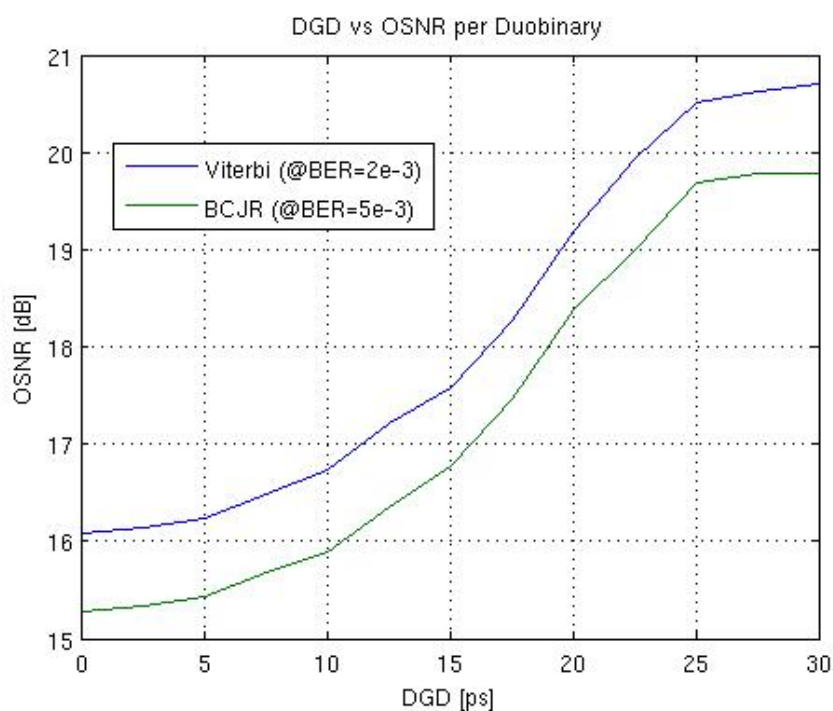


Figura 6.1. DGD vs OSNR per modulazione Duobinary

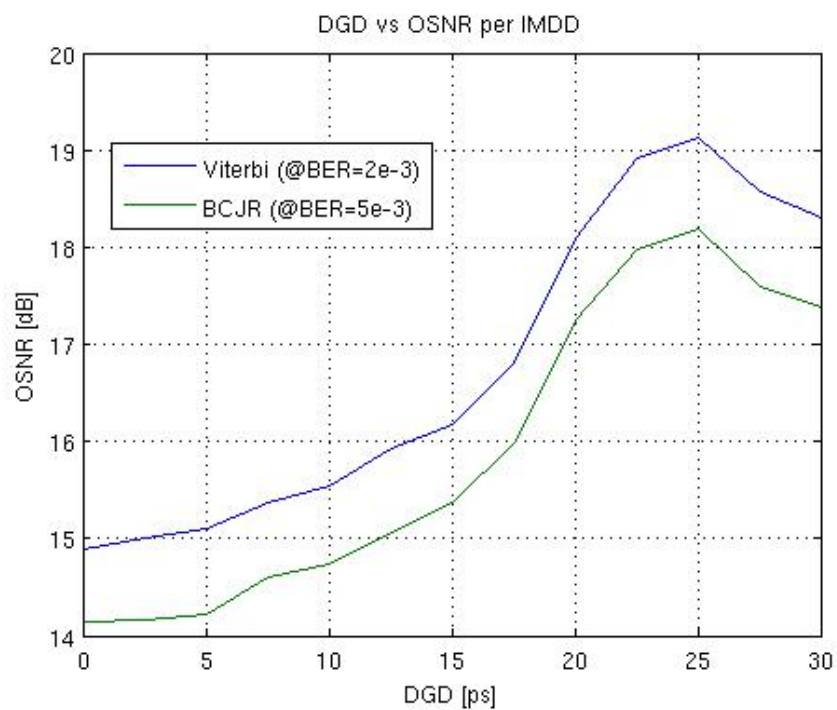


Figura 6.2. DGD vs OSNR per modulazione IMDD

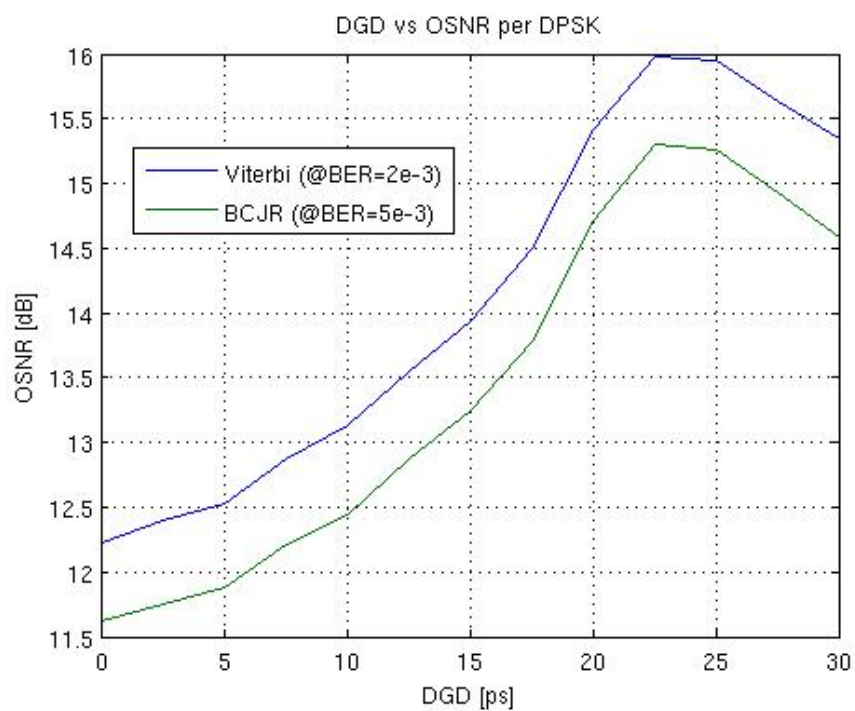


Figura 6.3. DGD vs OSNR per modulazione DPSK

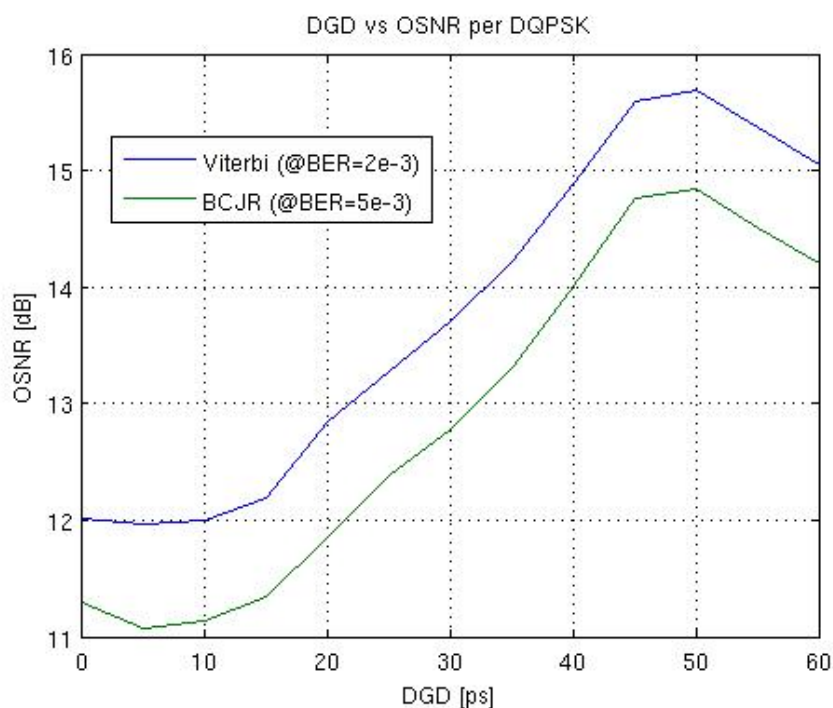


Figura 6.4. DGD vs OSNR per modulazione DQPSK

Guardando le curve ottenute, per tutti i formati di modulazione (e utilizzando i FEC) le prestazioni del BCJR sono migliori di quelle di Viterbi. Questo si spiega nel fatto che il FEC del BCJR ha un BER di riferimento maggiore (5×10^{-3}) rispetto al FEC di Viterbi (2×10^{-3}), e siccome entrambi gli algoritmi danno le stesse prestazioni @BER uguali (risultati ottenuti precedentemente @BER= 10^{-3}) adesso il BCJR migliora perché ha un BER di riferimento maggiore.

Per tutti i formati di modulazione, il guadagno del BCJR nei confronti di Viterbi in termini di OSNR è sempre di 1dB circa. Rimane da verificare il fatto che questo guadagno di 1dB sia sufficiente per rendere fattibile l'utilizzo del BCJR+FEC di terza generazione a decodifica *soft* iterativa, tenendo conto del costo e complessità maggiore che questa soluzione ha rispetto a quella di Viterbi+FEC di seconda generazione a decodifica *hard* iterativa.

Capitolo 7

Prestazioni del ricevitore MLSE con un ridotto numero di campioni per bit

Precedentemente sono state studiate le prestazioni della decodifica MLSE per diversi formati di modulazione (Duobinary, IMDD, DPSK e DQPSK). In tutti i casi, è stato considerato che il blocco MLSE prende 4 campioni per bit per calcolare le metriche di ramo ed eseguire l'algoritmo di Viterbi.

Siccome il sistema sotto studio ha un bit-rate pari a $R_b=43$ Gbps, l'implementazione del ricevitore MLSE che lavori con 4 campioni per bit avrebbe bisogno di un campionatore che prendesse 172 Gcampioni/s. Tuttora, non è possibile l'implementazione di un sistema del genere, per questo motivo si rende necessaria la riduzione del numero di campioni per bit presi dal blocco MLSE.

Per visualizzare le prestazioni dell'MLSE che lavori con un numero minore di campioni per bit, è stato scelto soltanto uno dei quattro formati di modulazione studiati precedentemente. Il formato scelto è il Duobinary, per essere considerato quello più critico (richiede i valori di OSNR maggiori).

7.1 Caratteristiche del sistema

Le caratteristiche del sistema sotto studio e delle simulazioni effettuate sono le seguenti:

- Frequenza di lavoro: 194 THz (1545 nm).
- Bit-rate (R_b): 43 Gbps.
- Fotodiodo ideale.
- Rotatore di polarizzazione a 90° . Con questo la potenza si divide esattamente alla metà in ognuno dei PSP (caso peggiore).
- Densità spettrale di rumore ottico accomodata per avere l'OSNR richiesto su una banda pari a 0.1 nm.
- Filtro ottico di tipo supergaussiano con $B_{3dB}=50$ GHz.
- Filtro elettrico di trasmissione di Bessel a 5 poli con $B_{3dB}=8.6$ GHz.
- Filtro elettrico di ricezione di Bessel a 5 poli con $B_{3dB}=86$ GHz.
- Sequenza pseudo-casuale o *Pseudo-Random Bit Sequence* (PRBS) di ordine 18. Numero di bits presi in considerazione per l'analisi MLSE pari a 2^{18} (262144 bits). Numero di campioni per bit pari a 12.
- La PRBS viene opportunamente *precodificata*.
- Numero di stati utilizzati dall'algoritmo di Viterbi pari a 64. Corrispondenti ad una memoria del canale pari a 6 bits.
- Le prestazioni del sistema vengono misurate attraverso l'OSNR richiesto per avere $BER=10^{-3}$, il quale assicura il funzionamento con l'utilizzo dei FEC.

Nelle simulazioni, è stato ottenuto il segnale elettrico e filtrato all'uscita del ricevitore per diversi valori della coppia (OSNR, DGD). Questo segnale viene memorizzato per poi essere elaborato dal ricevitore MLSE implementato nel software MatLab, ottenendo un valore di BER pari al rapporto fra il numero di

errori ed il numero di bit trasmessi (questo rapporto viene valutato confrontando la sequenza trasmessa con quella ottenuta dopo la decisione). Questi valori di BER vengono interpolati linearmente (si calcola la funzione $\log(-\log(\text{BER}))$ essendo la più simile ad una retta) per trovare per ogni valore di DGD il valore di OSNR necessario per avere $\text{BER}=10^{-3}$.

Lo schema del sistema viene mostrato nella figura 7.1.

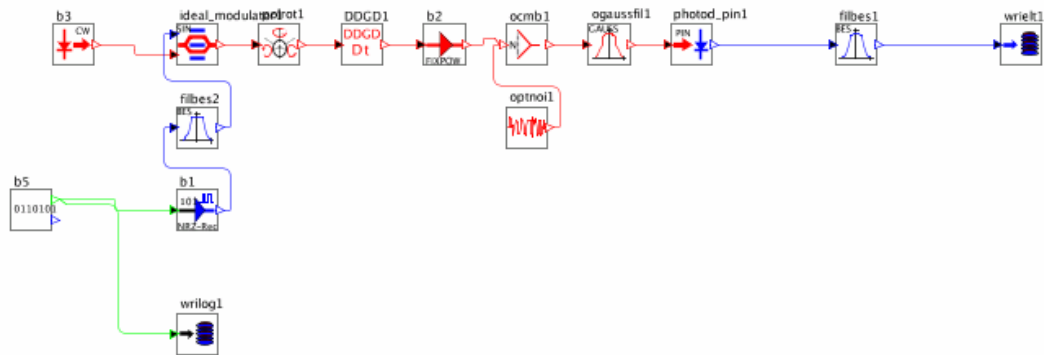


Figura 7.1. Schema del sistema per modulazione Duobinary

Il primo passo consiste nell'ottimizzazione della banda dei filtri elettrici in trasmissione e in ricezione. Simulando l'andamento del OSNR richiesto per avere $\text{BER}=10^{-3}$ al variare del DGD, è stato ottenuto un OSNR pari a 19.488dB @ $\text{DGD}=23.25\text{ps}$ (1 tempo di bit).

Fissandoci in questo punto ($\text{OSNR}=19.488\text{dB}$, $\text{DGD}=23.25\text{ps}$) sono state variate le larghezze di banda dei filtri elettrici in trasmissione e ricezione:

- Per il filtro in trasmissione: da $0.1 \cdot R_b$ a $0.3 \cdot R_b$ con passi di $0.02 \cdot R_b$.
- Per il filtro in ricezione: da $0.5 \cdot R_b$ a $2 \cdot R_b$ con passi di $0.25 \cdot R_b$.

Il BER più basso ottenuto è stato pari a $1.1255 \cdot 10^{-3}$, corrispondente alla coppia di bande $B_{TX}=0.2 \cdot R_b=8.6\text{GHz}$ e $B_{RX}=2 \cdot R_b=86\text{GHz}$. Questo significa che le larghezze di banda ottime ottenute coincidono con quelle impostate inizialmente.

7.2 Confronto fra 2 campioni/bit e 4 campioni/bit

Scendendo a 2 campioni per bit, è necessario ottimizzare nuovamente le larghezze di banda dei filtri elettrici in trasmissione e in ricezione. Le bande sono state variate nello stesso modo che nel caso di 4 campioni per bit, con la differenza che l'analisi è stata ripetuta più volte variando l'istante di campionamento sui 6 valori possibili: dall'istante 1 (prendendo il primo ed il settimo campione dei 12 che compongono il bit) all'istante 6 (prendendo il sesto ed il dodicesimo campione).

Per ognuno di questi istanti di campionamento è stata identificata la coppia di larghezze di banda (B_{TX} , B_{RX}) corrispondente al BER minore. Il BER più basso in assoluto ottenuto è stato pari a $1.0949e-3$ nell'istante 2, corrispondente alla coppia di bande $B_{TX}=0.2 \cdot R_b=8.6GHz$ e $B_{RX}=2 \cdot R_b=86GHz$. Questo significa che le larghezze di banda ottimizzate coincidono con quelle ottenute prendendo 4 campioni per bit.

In seguito, sono state calcolate le curve DGD vs OSNR @ BER= 10^{-3} per il caso di 2 campioni per bit variando l'istante di campionamento e confrontate con la curva corrispondente a 4 campioni per bit. Il risultato viene mostrato nella figura 7.2. Si può notare come sia la curva corrispondente a 4 campioni per bit che tutte le curve corrispondenti a 2 campioni per bit coincidono quasi perfettamente. Questo significa che nelle condizioni in cui lavora il sistema, scendendo da 4 a 2 campioni per bit non si ha alcuna penalità sull'OSNR richiesto per avere BER= 10^{-3} , e che le prestazioni dipendono molto poco dall'istante di campionamento.

Un modo migliore per visualizzare la dipendenza dall'istante di campionamento consiste nel guardare l'andamento dell'OSNR al variare dell'istante di campionamento per diversi valori di DGD. Il risultato viene mostrato nella figura 7.3.

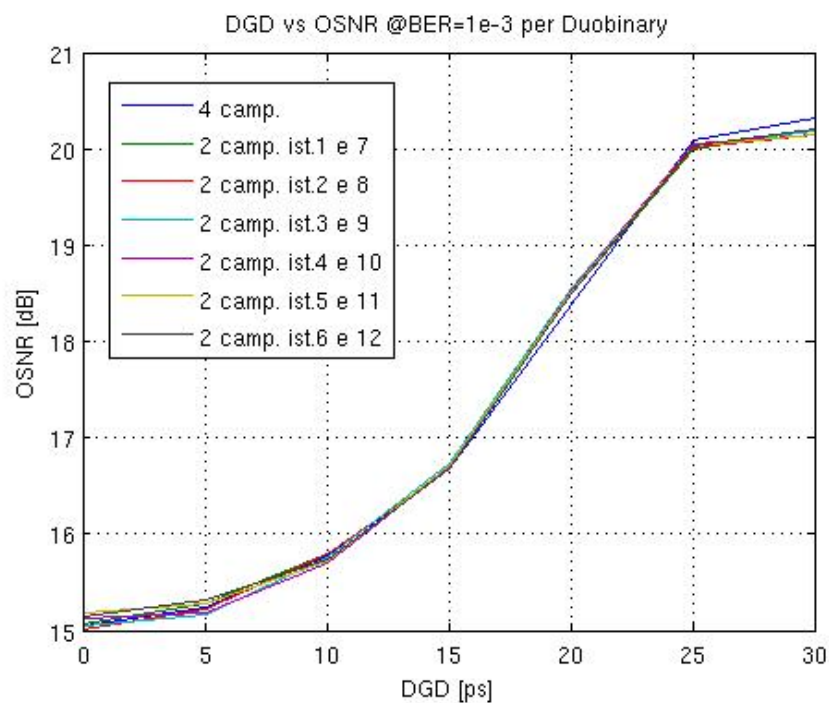


Figura 7.2. DGD vd OSNR per 4 camp/bit e 2 camp/bit variando l'istante di campionamento

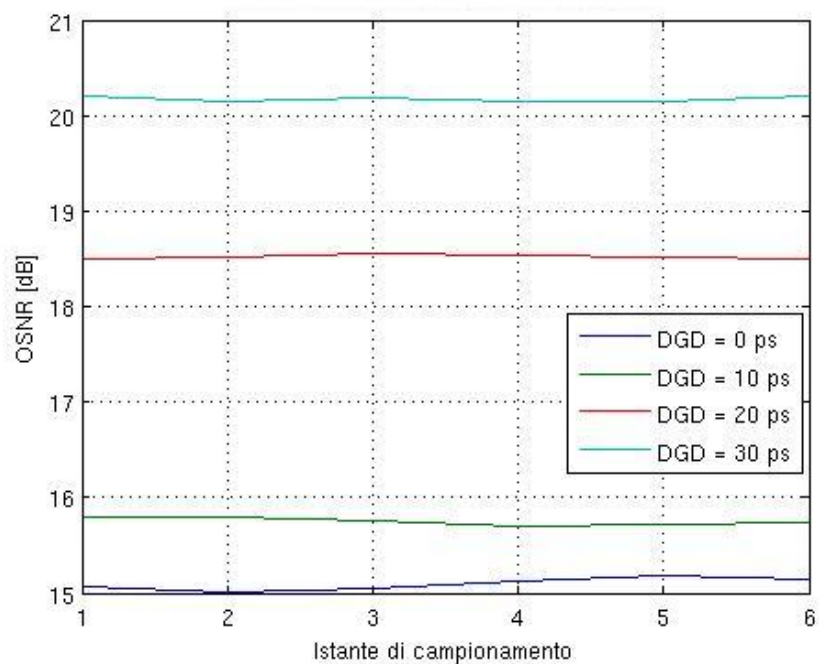


Figura 7.3. Istante di campionamento vs OSNR per diversi valori di DGD. 2 camp/bit

Si può notare come le curve di OSNR per diversi valori di DGD al variare dell'istante di campionamento sono praticamente delle rette orizzontali. Questo significa che non si ha dipendenza dall'istante di campionamento quando il blocco MLSE lavora prendendo 2 campioni per bit.

Fino a questo punto si può dire che visti i risultati ottenuti l'utilizzo di 4 campioni per bit nel ricevitore MLSE è uno spreco, dato che prendendo soltanto 2 campioni per bit si hanno le stesse prestazioni e queste non dipendono dall'istante di campionamento, cioè i 2 campioni possono essere presi in qualsiasi istante lungo la durata del bit e questo non influisce sul risultato.

7.3 Confronto fra 1 campione/bit e 4 campioni/bit

Adesso vengono analizzati i risultati ottenuti quando il ricevitore MLSE prende soltanto 1 campione per bit. Anche in questo caso è necessario ottimizzare nuovamente le larghezze di banda dei filtri elettrici in trasmissione e in ricezione. Le bande sono state variate nello stesso modo che nel caso di 2 campioni per bit, ma prendendo soltanto 5 dei 12 istanti di campionamento possibili. Gli istanti presi in considerazione sono stati 1, 3, 6, 9 e 12.

Per ognuno di questi istanti di campionamento è stata identificata la coppia di larghezze di banda (B_{TX} , B_{RX}) corrispondente al BER minore. Il BER più basso in assoluto ottenuto è stato pari a $1.1140e-3$ nell'istante 3, corrispondente alla coppia di bande $B_{TX}=0.2*R_b=8.6GHz$ e $B_{RX}=0.5*R_b=21.5GHz$.

Rispetto ai casi di 4 e 2 campioni per bit, il filtro elettrico in ricezione viene fortemente ristretto, passando da $2*R_b$ a $0.5*R_b$. Il motivo per cui il filtro in ricezione deve essere ristretto in quel modo è che prendendo soltanto 1 campione per bit il segnale viene disturbato dal rumore in una maniera più evidente, e quindi si rende necessario stringere il filtro a modo da tagliare il più possibile il rumore (con la penalità di tagliare anche un pezzo di segnale).

Succesivamente, sono state calcolate le curve DGD vs OSNR @ BER= 10^{-3} per il caso di 1 campione per bit variando l'istante di campionamento e confrontate con la curva corrispondente a 4 campioni per bit. Il risultato viene mostrato nella figura 7.4.

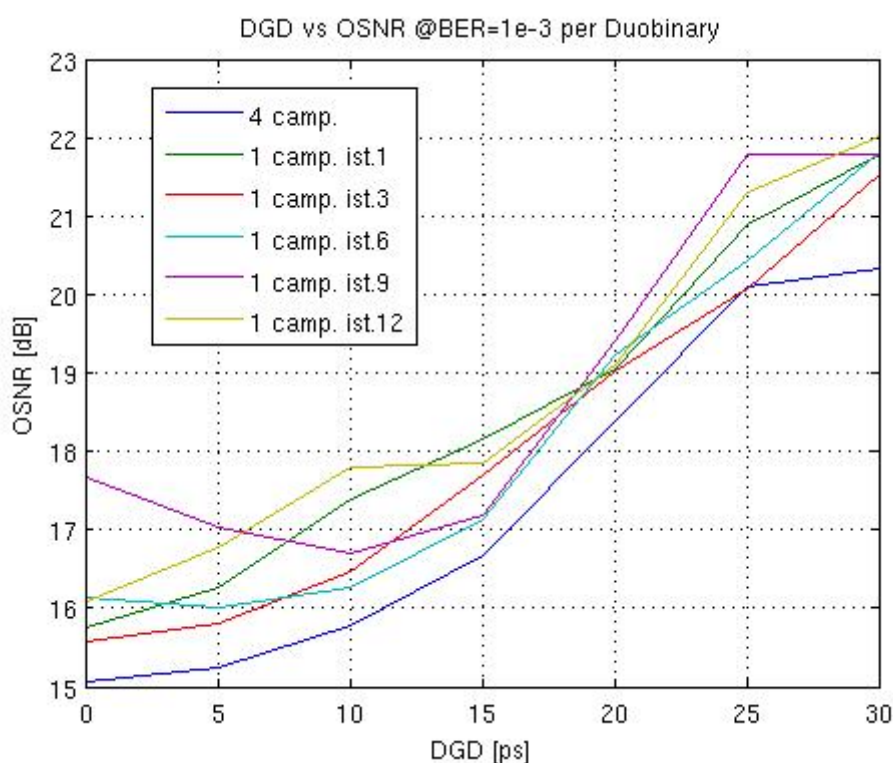


Figura 7.4. DGD vs OSNR per 4 camp/bit e 1 camp/bit variando l'istante di campionamento

Si può notare come in questo caso esiste una differenza notevole fra la curva corrispondente a 4 campioni per bit e quelle corrispondenti a 1 campione per bit. Si verifica anche che nel caso di 1 campione per bit si ottengono curve diverse per ogni istante di campionamento. Quindi si conclude che le prestazioni dipendono dall'istante di campionamento.

Fra tutte le curve corrispondenti a 1 campione per bit, quella che richiede l'OSNR minore al variare del DGD è la corrispondente all'istante di campionamento 3, ad eccezione dei casi DGD=10 ps e DGD=15 ps in cui la

curva che richiede l'OSNR minore è la corrispondente all'istante di campionamento 6.

Un modo migliore per visualizzare la dipendenza dall'istante di campionamento è guardare l'andamento dell'OSNR al variare dell'istante di campionamento per diversi valori di DGD. Il risultato viene mostrato nella figura 7.5.

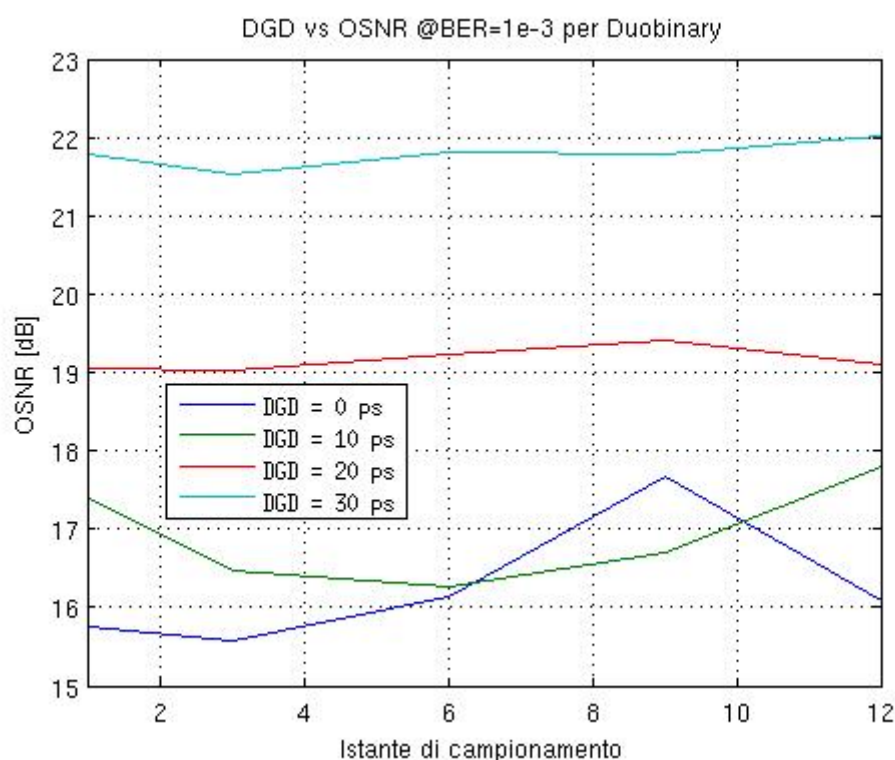


Figura 7.5. Istante di campionamento vs OSNR per diversi valori di DGD. 1 camp/bit

Si può notare come rispetto al caso di 2 campioni per bit nel quale le curve di OSNR erano simili a delle rette orizzontali, nel caso di 1 campione per bit esiste una forte dipendenza dall'istante di campionamento. Si verifica anche che la dipendenza è più forte per valori di DGD “bassi” (per DGD=0 ps c'è una variazione di 2dB fra l'istante 3 e l'istante 9), e che questa dipendenza diminuisce per valori di DGD “alti” (per DGD=30 ps la variazione massima è di circa 0.5dB).

Prendendo l'OSNR minore fra tutti gli istanti di campionamento per ogni valore di DGD, si può generare una curva corrispondente a 1 campione per bit con istante di campionamento ottimizzato. Il confronto fra questa curva e quella corrispondente a 4 campioni per bit viene mostrato nella figura 7.6.

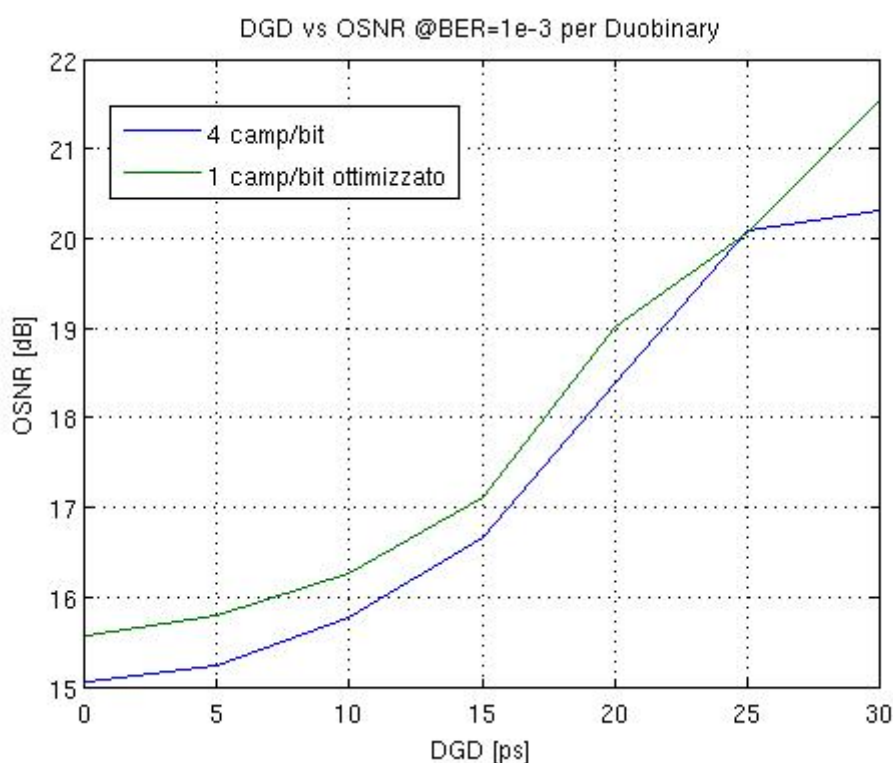


Figura 7.6. DGD vs OSNR per 4 camp/bit e 1 camp/bit ottimizzato

Si può notare la penalità esistente fra l'utilizzo di 4 campioni per bit nel ricevitore MLSE e l'utilizzo di soltanto 1 campione per bit dopo aver ottimizzato l'istante di campionamento. Questa penalità è di circa 0.5dB lungo tutti i valori di DGD, salvo eccezioni in cui la penalità cresce fino a 1dB (DGD=30 ps) o diventa inesistente (DGD=25 ps).

Con questi risultati si può concludere che ottimizzando l'istante di campionamento è possibile l'implementazione del sistema con un ricevitore MLSE che lavori prendendo solo 1 campione per bit (cioè 43 Gcampioni/s) con

una penalità di circa 0.5dB rispetto ai ricevitori MLSE che lavorano a 2 e 4 campioni per bit.

7.4 Prestazioni del sistema in caso di forte presenza della PMD

Fino a questo punto, le analisi sono state effettuate considerando valori di DGD fino a 30 ps (con eccezione del formato di modulazione DQPSK in cui la DGD arrivava fino a 60 ps). Siccome la DGD consiste nello sfasamento temporale dei 2 stati di polarizzazione del segnale, la penalità massima sull'OSNR richiesto è stata ottenuta per valori di DGD vicini al tempo di bit. Con l'aumento del DGD questa penalità scendeva leggermente e si ipotizzava che raggiungesse altri massimi per valori di DGD vicini ai multipli del tempo di bit.

Adesso viene calcolato l'andamento dell'OSNR richiesto al ricevitore per avere $BER=10^{-3}$ facendo arrivare la DGD fino a 100ps, a modo di vedere se quella ipotesi viene confermata. E' stato utilizzato il formato di modulazione Duobinary prendendo 4 campioni per bit nel ricevitore MLSE e con le larghezze di banda dei filtri elettrici in trasmissione e in ricezione ottimizzate. I risultati possono essere visualizzati nella figura 7.7.

Si osserva come una volta raggiunto il massimo di penalità, questa scende leggermente per poi tornare a crescere. Si può notare anche come la distanza fra i massimi di penalità è più o meno la stessa e vicina al tempo di bit, dando l'idea di "periodicità" nell'andamento dell'OSNR.

Questo comportamento della curva si mantiene fino a $DGD=85$ ps. Per i valori di DGD più elevati si ha una crescita repentina e notevole della penalità, assumendo valori molto superiori ai 5dB dentro i quali si mantiene per valori di DGD più bassi. Questa crescita della penalità può essere giustificata nel fatto che per quei valori di DGD la memoria del canale diventa maggiore della memoria del

treillis e quindi il ricevitore MLSE a 64 stati non riesce più a garantire le stesse prestazioni.

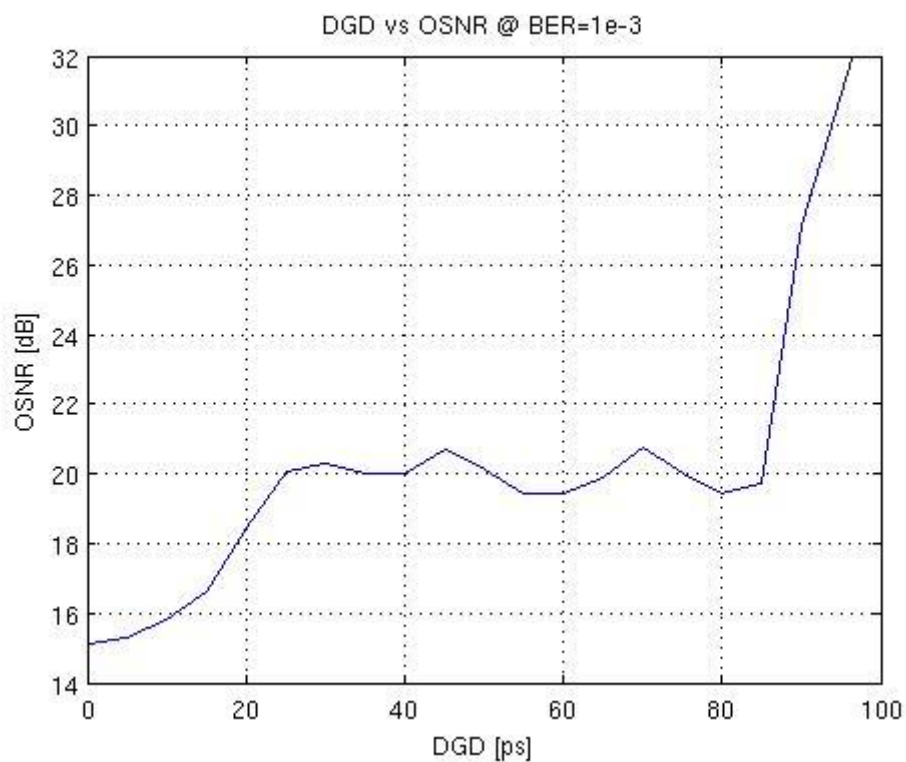


Figura 7.7. DGD vs OSNR in caso di forte presenza della PMD

Rimangono da studiare i casi in cui il ricevitore MLSE lavora con più stati e la penalità esistente elaborando il segnale con 2 e 1 campione per bit.

Conclusioni

Dagli inizi, i sistemi ottici hanno superato diversi ostacoli per trovare possibilità di implementazione prima e lo slancio definitivo come principale tecnologia per le telecomunicazioni dopo. L'avanzamento della tecnologia sta nel continuo incremento delle velocità di trasmissione dei dati; una volta arrivati alla velocità di 40 Gbps e oltre, la PMD inizia a penalizzare notevolmente le prestazioni dei sistemi rendendoli inutilizzabili, cosa che non accade quando le velocità di trasmissione stanno al di sotto dei 10 Gbps.

Un'alternativa per sopperire alla forte penalità introdotta dalla PMD senza introdurre compensatori che lavorano con diversi polarizzatori retroalimentati, è l'utilizzo dell'MLSE nel ricevitore come sistema di compensazione elettronica.

E' stato verificato che nel sistema sotto studio, senza ottimizzazione del filtro ottico in ricezione, l'utilizzo dell'MLSE consente di tollerare valori di DGD fino a 30 ps (superiori al tempo di bit) con una penalità inferiore ai 5dB rispetto al caso di assenza della PMD. Questa penalità viene raggiunta già per valori di DGD attorno ai 15 ps quando non viene implementato l'MLSE, quindi si può dire che l'utilizzo dell'MLSE consente di raddoppiare la tolleranza dei sistemi alla PMD. Sono stati studiati quattro formati di modulazione: Duobinary, IMDD, DPSK e DQPSK. In questo ordine (e per il sistema sotto studio), ogni formato di modulazione migliora le prestazioni di quello precedente. Questo miglioramento si visualizza in assenza di PMD, mentre l'andamento al variare della DGD è

molto simile per tutti i formati, mantenendo le differenze iniziali ad eccezione del formato DQPSK, il quale ha una tolleranza alla DGD fino a 60 ps mantenendosi sotto i 5 dB di penalità. Questo è dovuto al fatto che il formato DQPSK (essendo a quattro livelli) ha un symbol-rate dimezzato rispetto agli altri formati di modulazione, e quindi la sua robustezza di fronte alla PMD viene raddoppiata. Con questo si può dire che la penalità dovuta alla presenza della PMD in un sistema è direttamente proporzionale al symbol-rate di esso.

Il sistema MLSE lavora con l'algoritmo di Viterbi, il quale permette di minimizzare la probabilità di errore sulla sequenza, scegliendo quella più probabile fra tutte le possibili sequenze. Sono stati effettuati degli studi sulla prestazione del sistema MLSE lavorando con l'algoritmo BCJR, il quale permette di minimizzare la probabilità di errore sul bit avendo una uscita di tipo soft, a differenza di Viterbi che ha una uscita di tipo hard. I risultati indicano che a parità di BER di riferimento gli algoritmi hanno le stesse prestazioni, quindi la scelta bit a bit di BCJR da come risultato la stessa sequenza scelta da Viterbi.

Per poter capire il possibile miglioramento che ha l'utilizzo dell'algoritmo BCJR nella elaborazione del segnale elettrico, bisogna trovare i BER di riferimento tipici dei codici FEC utilizzabili in ognuno dei casi. Quando si utilizza l'algoritmo di Viterbi si può lavorare con codici FEC di seconda generazione a decodifica hard iterativa (quello scelto ha un BER di riferimento pari a 2×10^{-3}). Dall'altra parte, utilizzando l'algoritmo BCJR si può lavorare con codici FEC di terza generazione a decodifica soft iterativa, dove quello scelto ha un BER di riferimento pari a 5×10^{-3} . Confrontando l'andamento delle curve DGD vs OSNR ai rispettivi BER di riferimento è stato trovato un guadagno pari a circa 1 dB del BCJR nei confronti di Viterbi, il quale si deve alla differenza nei valori di BER di riferimento. Nelle implementazioni pratiche si deve determinare se questo guadagno riesce a compensare la maggiore complessità che ha l'utilizzo dell'algoritmo BCJR, anche perché i codici FEC di terza generazione sono ancora in fase di standardizzazione.

Come ultima considerazione, sono state studiate le prestazioni del sistema MLSE passando da 4 campioni per bit a 2 e 1 campione per bit, a modo di avere nelle implementazioni pratiche dei campionatori a velocità minore. I risultati indicano che nelle condizioni del sistema studiato (Duobinary) è completamente innecessario l'utilizzo di 4 campioni per bit, dato che lavorando con 2 campioni per bit si ottengono le stesse prestazioni a prescindere dall'istante di campionamento. Dall'altra parte, lavorando con soltanto 1 campione per bit si ha una forte dipendenza dall'istante di campionamento. E' stato determinato che una volta ottimizzato l'istante di campionamento la penalità rispetto al caso di 4 campioni per bit è inferiore a 1 dB. Rimane da verificare nelle implementazioni pratiche se questo leggero peggioramento può essere tollerato a patto di avere un campionatore con velocità pari al bit-rate (1 campione per bit) e non un campionatore più complesso con almeno il doppio della velocità (2 campioni per bit).

Sono state studiate anche le prestazioni del sistema in caso di forte presenza della PMD, avendo come risultato un andamento periodico con periodo pari al tempo di bit, e una tolleranza limitata dalla memoria del canale, cioè dal numero di stati presi in considerazione nel ricevitore MLSE. Possono essere futuri temi di studio le prestazioni del sistema variando sia il numero di stati del ricevitore MLSE che il numero di campioni per bit presi in considerazione.

Referenze

- [1] Poole, Wagner, *IEEE Electronic Letters*, Sep. 1996.
- [2] H. Bülow, “System Outage Probability Due to First- and Second- Order PMD”, *IEEE Photonics Technology Letters*, Vol. 10, No. 5, May 1998.
- [3] G.J. Foschini, C.D. Poole, “Statistical Theory of polarization dispersion in Single mode fibers”, *IEEE Photonics Technology Letters*, Vol. 9, No. 11, pp. 1439-1456, Nov. 1991.
- [4] N. Alic, G. C. Papen, R. E. Saperstein, L. B. Milstein, Y. Fainman, “Signal statistics and maximum likelihood sequence estimation in intensity modulated fiber optic links containing a single optical preamplifier”, *Opt. Exp.*, Vol. 13, issue 5, pp. 4568-4579, Jun. 2005.
- [5] G. Bosco, P. Poggiolini, M. Visintin, “Performance Analysis of MLSE Receivers Based on the Square-Root Metric”, *ECOC 2007*, Berlin, Germany, 16-20 Sep. 2007.

- [6] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Inform. Theory*, Vol. IT-13, pp. 260-269, Apr. 1967.
- [7] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. Inform. Theory*, pp. 284-287, Mar. 1974.
- [8] I.B. Djordjevic, B. Vasic, "Nonlinear BCJR equalizer for suppression of intrachannel nonlinearities in 40 Gb/s optical communications systems", *Optics Express*, Vol. 14, No. 11, pp. 4625-4635, May. 2006.
- [9] O. Koufopavlou, I. Tomkos, A. Tychopoulos, "FEC IN OPTICAL COMMUNICATIONS" *IEEE Circuits & Devices Magazine*, pp. 79-86, Nov./Dec. 2006.
- [10] K. Seki, K. Mikami, A. Katayama, S. Suzuki, N. Shinohara, M. Nakabayashi, "Single-chip FEC codec using a concatenated BCH code for 10 Gbps LH optical transmission systems", *Proc. IEEE Custom Integr. Circuits Conf. 2003*, pp. 279-282.

Bibliografia

- L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate”, *IEEE Trans. Inform. Theory*, pp. 284-287, Mar. 1974.
- A. Barbieri, G. Colavolpe, D. Fertonani, “Reduced-Complexity BCJR Algorithm for Turbo Equalization”, *IEEE Transactions on Communications*, Vol. 55, No. 12, pp. 2279-2287, Dec. 2007.
- G. Bosco, “Performance analysis of optical communication systems” (Tesi di Dottorato), Torino: Politecnico di Torino, XIV Ciclo.
- G. Bosco, I. Cano, V. Curri, P. Poggiolini, “Optimization of Branch Metric Exponent and Quantization Range in MLSE Receivers for Duobinary Systems”.
- G. Bosco, P. Poggiolini, M. Visintin, “Performance Analysis of MLSE Receivers Based on the Square-Root Metric”.
- H. Bülow, “System Outage Probability Due to First- and Second- Order PMD”, *IEEE Photonics Technology Letters*, Vol. 10, No. 5, May 1998.
- H. Bülow, F. Buchali, G. Thielecke, “Viterbi equalizer for mitigation of distortions from chromatic dispersion and PMD at 10 Gb/s”, *Optical Society of America*, 2004.

- I.B. Djordjevic, B. Vasic, "Nonlinear BCJR equalizer for suppression of intrachannel nonlinearities in 40 Gb/s optical communications systems", *Optics Express*, Vol. 14, No. 11, pp. 4625-4635, May. 2006.
- C. Dorschky et al., "Measurement of the Dispersion Tolerance of Optical Duobinary with an MLSE-Receiver at 10.7 Gb/s", *Optical Society of America*, 2005.
- G.D. Forney JR, "The Viterbi Algorithm", *Proceedings of the IEEE*, Vol. 61, No. 3, pp. 268-278, Mar. 1973.
- R. Gaudino, P. Poggiolini, "Reti in fibra ottica", Torino, Politeko, 2005.
- L. Giugno, M. Luise, "Introduzione ai Low-Density Parity-Check codes (LDPC)", May 2003.
- A.H. Gnauck, S. Chandrasekhar, P.J. Winzer, "Dispersion-Tolerant 10-Gb/s Duobinary System Employing Heterodyne Detection and MLSE", *IEEE Photonics Tech. Letters*, Vol. 18, No. 5, pp. 697-699, Mar. 2006.
- J. Hagenauer, P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *IEEE Trans. Inform. Theory*, pp. 1680-1686, 1989.
- O. Koufopavlou, I. Tomkos, A. Tychopoulos, "FEC IN OPTICAL COMMUNICATIONS" *IEEE Circuits & Devices Magazine*, pp. 79-86, Nov./Dec. 2006.
- T. Kupfer, S. Langenbach, J. Whiteaway, "PMD Compensation using Electronic Equalization particular Maximum Likelihood Sequence Estimation", *Optical Society of America*, 2006.
- E. Roff , "Use of MLSE receivers in optical communications" (Tesi), Torino: Politecnico di Torino, 2007.
- P.J. Winzer, H. Kogelnik, K. Ramanan, "Precise Outage Specifications for First-Order PMD", *IEEE Photonics Technology Letters*, Vol. 16, No. 2, pp. 449-451, Feb. 2004.

Apendice A

Codice MatLab dell'algoritmo BCJR per i formati di modulazione Duobinary, IMDD e DPSK

Il codice MatLab dell'algoritmo BCJR è esattamente uguale a quello dell'algoritmo di Viterbi esistente per quanto riguarda le fasi di sincronizzazione del segnale e di stima delle metriche di ramo. Ad ogni iterazione, per compilare le matrici *gamma*, *alfa* e *beta* il codice utilizzato è il seguente:

```
p=2^nu;           %nu è la memoria del canale, così p è
                   %il numero di stati

gamma=zeros(2*p,N); %N è la lunghezza della sequenza
alfa=zeros(p,N+1);
beta=zeros(p,N+1);

alfa(:,1)=a0;      %a0 è il vettore con gli  $\alpha_0(s)$ 
beta(:,N+1)=bn;    %bn è il vettore con gli  $\beta_N(s)$ 
```

```

for j=1:N
    %{si prende il vettore con le metriche di ramo "mu"}
    gamma(:,j)=mu;
    k=1;
    for s=1:2:p-1
        alfa(s,j+1)=max*((alfa(k,j)+gamma(s,j)),(alfa(p/2+k,j)
        +gamma(p+s,j)));
        alfa(s+1,j+1)=max*((alfa(k,j)+gamma(s+1,j)),(alfa(p/2+
        k,j)+gamma(p+s+1,j)));
        k=k+1;
    end
end

for j=N:-1:1          %si scorre N dalla fine verso l'inizio
    k=1;
    m=1;
    for s=1:p
        beta(s,j)=max*((beta(m,j+1)+gamma(k,j)),(beta(m+1,j+1)
        +gamma(k+1,j)));
        k=k+2;
        m=m+2;
        if (m>p)
            m=1;
        end
    end
end

L=zeros(1,N);          %L è il vettore con gli L(uj)
U=zeros(1,N);          %U è il vettore con gli uj presi dalla
                        %decisione "hard" sugli L(uj)

for j=1:N
    Lzero=zeros(1,p);    %Lzero contiene i valori fra cui
                        %si fa max* per i 0
    Luno=zeros(1,p);     %Luno contiene i valori fra cui
                        %si fa max* per gli 1

```

```
k=1;
m=1;
for s=1:p
    Lzero(s)=alfa(s,j)+gamma(k,j)+beta(m,j+1);
    Luno(s)=alfa(s,j)+gamma(k+1,j)+beta(m+1,j+1);
    k=k+2;
    m=m+2;
    if (m>p)
        m=1;
    end
end

Lzero2=max(Lzero);
Luno2=max(Luno);

L(j)=Lzero2-Luno2;

if (L(j)<0)
    u(j)=1;
end
end
```


Apendice B

Codice MatLab dell'algoritmo BCJR per il formato di modulazione DQPSK

Anche per il formato DQPSK, il codice MatLab dell'algoritmo BCJR è esattamente uguale a quello dell'algoritmo di Viterbi esistente per quanto riguarda le fasi di sincronizzazione del segnale e di stima delle metriche di ramo. Differisce la parte restante dato che il segnale DQPSK è a quattro livelli. Ad ogni iterazione, per compilare le matrici *gamma*, *alfa* e *beta* il codice utilizzato è il seguente:

```
p=4^nu;                %nu è la memoria del canale, così p è
                        %il numero di stati

gamma=zeros(4*p,N);    %N è la lunghezza della sequenza
alfa=zeros(p,N+1);
beta=zeros(p,N+1);

alfa(:,1)=a0;          %a0 è il vettore con gli  $\alpha_0(s)$ 
beta(:,N+1)=bn;        %bn è il vettore con gli  $\beta_N(s)$ 
```

```
for j=1:N
    %{si prende il vettore con le metriche di ramo "mu"}
    gamma(:,j)=mu;
    k=1;
    for s=1:4:p-3
        alfa(s,j+1)=max*((alfa(k,j)+gamma(s,j)), (alfa(p/4+k,j)
        +gamma(p+s,j)),
        (alfa(p/2+k,j)+gamma(2*p+s,j)), (alfa(3*p/4+k,j)+gamma(
        3*p+s,j))));

        alfa(s+1,j+1)=max*((alfa(k,j)+gamma(s+1,j)), (alfa(p/4+
        k,j)+gamma(p+s+1,j)),
        (alfa(p/2+k,j)+gamma(2*p+s+1,j)), (alfa(3*p/4+k,j)+gamm
        a(3*p+s+1,j))));

        alfa(s+2,j+1)=max*((alfa(k,j)+gamma(s+2,j)), (alfa(p/4+
        k,j)+gamma(p+s+2,j)),
        (alfa(p/2+k,j)+gamma(2*p+s+2,j)), (alfa(3*p/4+k,j)+gamm
        a(3*p+s+2,j))));

        alfa(s+3,j+1)=max*((alfa(k,j)+gamma(s+3,j)), (alfa(p/4+
        k,j)+gamma(p+s+3,j)),
        (alfa(p/2+k,j)+gamma(2*p+s+3,j)), (alfa(3*p/4+k,j)+gamm
        a(3*p+s+3,j))));

        k=k+1;
    end
end

for j=N:-1:1          %si scorre N dalla fine verso l'inizio
    k=1;
    m=1;
    for s=1:p
```

```
beta(s,j)=max*((beta(m,j+1)+gamma(k,j)),(beta(m+1,j+1)
+gamma(k+1,j)),
(beta(m+2,j+1)+gamma(k+2,j)),(beta(m+3,j+1)+gamma(k+3,
j)));
k=k+4;
m=m+4;
if (m>p)
    m=1;
end
end
end

L=zeros(1,N);          %L è il vettore con gli L(uj)
U=zeros(1,N);          %U è il vettore con gli uj presi dalla
                        %decisione "hard" sugli L(uj)

for j=1:N
    Lzero=zeros(1,p);   %Lzero contiene i valori fra cui
                        %si fa max* per i 0
    Luno=zeros(1,p);    %Luno contiene i valori fra cui
                        %si fa max* per gli 1
    Ldue=zeros(1,p);    %Ldue contiene i valori fra cui
                        %si fa max* per i 2
    Ltref=zeros(1,p);   %Ltref contiene i valori fra cui
                        %si fa max* per i 3

    k=1;
    m=1;
    for s=1:p
        Lzero(s)=alfa(s,j)+gamma(k,j)+beta(m,j+1);
        Luno(s)=alfa(s,j)+gamma(k+1,j)+beta(m+1,j+1);
        Ldue(s)=alfa(s,j)+gamma(k+2,j)+beta(m+2,j+1);
        Ltref(s)=alfa(s,j)+gamma(k+3,j)+beta(m+3,j+1);
        k=k+4;
        m=m+4;
    end
end
```

```
        if (m>p)
            m=1;
        end
    end

    Lzero2=max(Lzero);
    Luno2=max(Luno);
    Ldue2=max(Ldue);
    Ltre2=max(Ltre);

    decision=[Lzero2 Luno2 Ldue2 Ltre2];
    Lmax=Lzero2;
    level=0;
    for t=2:length(decision)
        if (decision(t)>Lmax)
            Lmax=decision(t);
            level=t-1;
        end
    end

    L(j)=Lmax;
    U(j)=level;
end
```