



UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA DELLE TELECOMUNICAZIONI
TESI DI LAUREA MAGISTRALE

STUDIO DI RICEVITORE GMSK A 300 kbps

RELATORE: Prof. Nevio Benvenuto

LAUREANDO: *Matteo Carlo Carlucci*

Padova, 18 aprile 2011

ANNO ACCADEMICO 2010-2011

Alla mia famiglia

Indice

Introduzione	1
1 Modulazione	3
1.1 Struttura dei pacchetti	3
1.2 Le modulazioni di fase	3
1.3 La modulazione GMSK	5
1.4 Codifica convoluzionale	8
1.5 Interleaver	10
1.6 Precodifica	11
1.7 Modulazione di fase	11
1.8 Spettro	15
1.9 Approssimazione	15
2 Canale	19
2.1 Realizzazioni del canale	20
2.2 Rumore bianco	23
3 Ricevitore	25
3.1 Sincronismo e stima del canale	25
3.1.1 Sincronizzazione	25
3.1.2 Stima del canale	26
3.1.3 Stima della potenza del rumore bianco	29
3.2 Rivelazione dei dati	29
3.2.1 FBA	29
3.2.2 FBA - versione semplificata (Max-Log-MAP)	32
3.2.3 Prestazioni del FBA	33
3.2.4 DFE	33
3.2.5 Calcolo dei coefficienti dei filtri per il DFE	38
3.3 Deinterleaver	39
3.4 Viterbi decoder	39
3.4.1 Valutazione delle prestazioni del codice	40

4	Valutazione delle prestazioni	43
4.1	Ricevitore con FBA	43
4.2	Ricevitore con DFE	45
4.3	Confronto fra le prestazioni del DFE e del FBA	46
5	Canali tempo varianti	47
5.1	Canale statico sul singolo pacchetto	47
5.1.1	Max-Log-MAP	49
5.1.2	DFE	52
5.1.3	Max-Log-MAP (media su 10 pacchetti)	54
5.1.4	DFE (media su 10 pacchetti)	56
5.2	Canale tempo-variante	58
5.2.1	Effetto Doppler	58
5.2.2	Modello a tempo continuo del canale	59
5.2.3	Spettro Doppler	60
5.2.4	Realizzazione dei tappi	61
5.2.5	Max-Log-MAP, stima fissa	66
5.2.6	DFE, stima fissa	68
5.3	Ricevitore Max-Log-MAP con LMS	70
5.3.1	LMS via RAM	70
5.3.2	Accorgimenti adottati	71
5.3.3	Prestazioni del Max-Log-MAP con LMS via RAM	76
5.3.4	Packet error rate	78
A	Prestazioni del metodo di correlazione	79
B	L'algoritmo LMS	83
B.1	Errori di stima con simboli non affetti da errori	86
B.2	Errore di stima con simboli ricavati dai LLR	87
B.3	Prestazioni con correzione di stima via LMS	88
	Bibliografia	91

Elenco delle figure

1.1	Struttura del pacchetto.	4
1.2	Modulatore CPFSK.	4
1.3	Confronto tra segnali FSK e MSK.	6
1.4	Schema globale.	7
1.5	Schema a blocchi del codificatore convoluzionale.	9
1.6	Possibili transizioni: in rosso relative ad un ingresso $\tilde{b}_n = 1$, in nero a $\tilde{b}_n = 0$	9
1.7	Relazione fra indice d'ingresso e d'uscita nell' <i>interleaver</i> usato.	10
1.8	Risposta impulsiva del filtro $g_P(t)$	13
1.9	Diagramma ad occhio del segnale x_n	13
1.10	Andamento della deviazione di fase $\Delta\varphi(t)$ per la sequenza dati mostrata in rosso ($a_k \in \{-1, 1\}$).	14
1.11	Rappresentazione sul piano complesso del segnale s^{bb} e dei valori della costellazione (quadrati neri).	14
1.12	Costellazione di s^{bb} in opportuni istanti multipli di T con modulatore a (a) $T/128$ e (b) $T/1024$	15
1.13	Densità spettrale del segnale modulato.	16
1.14	Risposta impulsiva del filtro equivalente h_{Tx} in (1.18).	17
1.15	Parte immaginaria del segnale s_{bb} ottenuto con lo schema equi- valente di Laurent (in blu) e con la modulazione GMSK (in rosso).	18
2.1	Modulo di due realizzazioni del canale g_{Ch} sui tempi $T_Q = T/8$	22
2.2	Modulo del canale equivalente come convoluzione dell'impulso h_{Tx} con il canale <i>ideal case</i> (a) <i>typical case</i> (b) e <i>worst case</i> (c).	22
2.3	Impulso h_{Tx} (in rosso) e corrispondente g_{eq} (in blu e nero, rispettivamente, parte reale e immaginaria). Canale <i>typical case</i> (a) e <i>worst case</i> (b).	23
2.4	Risposta in frequenza del filtro applicato al rumore bianco.	24
2.5	PSD del rumore bianco filtrato prima del campionamento (ros- so) e dopo (nero).	24

3.1	Modulo della cross-correlazione, $ \mathbf{r}_{rp}(m) $, ed istante di modulo massimo. Canale <i>worst case</i> , $\Gamma = 10$ dB.	26
3.2	Stima del canale equivalente $\hat{g}_{eq,i}$ $i = -L_1, \dots, L_2$ con $L_1 = 1$ e $L_2 = 4$ per canale <i>typical case</i> e $\Gamma = 10$ dB.	27
3.3	Confronto fra andamento teorico e reale di Λ_n per canale <i>ideale</i> (a), <i>typical case</i> (b) e <i>worst case</i> (c). Valori in dB.	28
3.4	Confronto fra $\sigma_{\omega_C}^2$ (in rosso) e la sua stima (in blu) per canale <i>ideale</i> (a), <i>typical case</i> (b) e <i>worst case</i> (c). Valori in dB.	30
3.5	Funzione distribuzione delle percentuali di errore al variare di Γ , canale <i>ideale</i>	34
3.6	Funzione distribuzione delle percentuali di errore al variare di Γ , canale <i>typical case</i>	34
3.7	Funzione distribuzione delle percentuali di errore al variare di Γ , canale <i>worst case</i>	35
3.8	Schema della retroazione in un ricevitore DFE.	36
3.9	Struttura generale del ricevitore DFE.	36
3.10	Modulo della risposta impulsiva equivalente con canale <i>worst case</i> , del filtro f e della loro convoluzione calcolati con $\Gamma = 10$ dB, $M_1 = 6$, $D = 4$ e $M_2 = 6$	39
3.11	Schema a blocchi della struttura utilizzata.	41
3.12	Probabilità di errore in funzione di Γ per lo schema di Fig. 3.11.	42
4.1	Canali noti. Probabilità di errore in funzione di Γ con <i>threshold detector</i> e con algoritmo <i>forward backward</i> (Max-Log-MAP) con e senza codifica (il caso di canale ideale con codifica è tutto al di sotto del valore 10^{-5}).	43
4.2	Canali stimati. Probabilità di errore in funzione di Γ con <i>hard detection</i> dei LLR ottenuti dal FBA normalizzato (senza codifica) e con <i>soft detection</i> tramite algoritmo di Viterbi (decodifica del codice convoluzionale).	44
4.3	Canali stimati. Probabilità di errore in funzione di Γ con <i>hard detection</i> dei LLR ottenuti dal FBA Max-Log-MAP (senza codifica) e con <i>soft detection</i> tramite algoritmo di Viterbi (decodifica del codice convoluzionale).	44
4.4	Canali stimati. Probabilità di errore in funzione di Γ del DFE con e senza codifica convoluzionale.	45
4.5	Canali stimati. Probabilità di errore in funzione di Γ con ricevitore DFE e FBA senza codifica convoluzionale.	46
4.6	Canali stimati. Probabilità di errore in funzione di Γ con ricevitore DFE e FBA con codifica convoluzionale.	46

5.1	Densità di probabilità di Rice per alcuni valori di K_i	48
5.2	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> statico, una realizzazione per pacchetto.	50
5.3	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> statico, una realizzazione per pacchetto.	50
5.4	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> statico, una realizzazione per pacchetto.	51
5.5	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> statico, una realizzazione per pacchetto.	51
5.6	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> statico, una realizzazione per pacchetto.	52
5.7	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> statico, una realizzazione per pacchetto.	52
5.8	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> statico, una realizzazione per pacchetto.	53
5.9	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> statico, una realizzazione per pacchetto.	53
5.10	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>typical case</i> statico, una realizzazione per 10 pacchetti.	54
5.11	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>worst case</i> statico, una realizzazione per 10 pacchetti.	54

5.12	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>typical case</i> statico, una realizzazione per 10 pacchetti.	55
5.13	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>worst case</i> statico, una realizzazione per 10 pacchetti.	55
5.14	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>typical case</i> statico, una realizzazione per 10 pacchetti.	56
5.15	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>worst case</i> statico, una realizzazione per 10 pacchetti.	56
5.16	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>typical case</i> statico, una realizzazione per 10 pacchetti.	57
5.17	Funzione distribuzione delle percentuali di <i>bit</i> sbagliati al variare di Γ , canale <i>worst case</i> statico, una realizzazione per 10 pacchetti.	57
5.18	Illustrazione dell'effetto Doppler.	58
5.19	Modello per generare l' <i>i</i> -esimo tappo del canale.	62
5.20	Modulo di una realizzazione dei tappi nel tempo, canale TY. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.	64
5.21	Modulo di una realizzazione dei tappi nel tempo, canale WC. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.	64
5.22	Fase di una realizzazione dei tappi nel tempo, canale TY. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.	65
5.23	Fase di una realizzazione dei tappi nel tempo, canale WC. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.	65
5.24	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima del canale all'inizio del pacchetto.	66
5.25	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima del canale all'inizio del pacchetto.	66

5.26	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima del canale all’inizio del pacchetto.	67
5.27	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima del canale all’inizio del pacchetto.	67
5.28	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima del canale all’inizio del pacchetto.	68
5.29	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima del canale all’inizio del pacchetto.	68
5.30	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima del canale all’inizio del pacchetto.	69
5.31	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima del canale all’inizio del pacchetto.	69
5.32	Schema del ricevitore con stima del canale tempo-variante tramite memoria temporanea $\Delta u(\mathbf{c})$	71
5.33	Andamento dell’errore e_k senza correzione (rosso) e con correzione di stima (blu) usando 3 <i>bit</i> per indirizzare la memoria. Canale <i>typical case</i> , $\Gamma = 8$ dB.	74
5.34	Andamento dell’errore e_k senza correzione (rosso) e con correzione di stima (blu) usando 3 <i>bit</i> per indirizzare la memoria. Canale <i>typical case</i> , $\Gamma = 14$ dB.	74
5.35	Andamento dell’errore e_k senza correzione (rosso) e con correzione di stima (blu e azzurro) usando 3 <i>bit</i> per indirizzare la memoria. Canale <i>worst case</i> , $\Gamma = 8$ dB.	75
5.36	Andamento dell’errore e_k senza correzione (rosso) e con correzione di stima (blu e azzurro) usando 3 <i>bit</i> per indirizzare la memoria. Canale <i>worst case</i> , $\Gamma = 14$ dB.	75

5.37	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	76
5.38	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	76
5.39	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	77
5.40	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	77
5.41	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	78
5.42	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima di canale via RAM ogni 100 <i>bit</i>	78
A.1	Confronto fra andamento teorico e reale di Λ_n con rumore bianco aggiunto a tempi T per canale <i>ideale</i> (a), <i>typical case</i> (b) e <i>worst case</i> (c). Valori in dB.	80
A.2	Confronto fra andamento teorico e reale di Λ_n con rumore aggiunto a tempi T_Q per canale <i>ideale</i> (a), <i>typical case</i> (b) e <i>worst case</i> (c). Valori in dB.	81
A.3	Confronto fra $\sigma_{\omega_C}^2$ (in rosso) e la sua stima (in blu) per canale <i>ideale</i> (a), <i>typical case</i> (b) e <i>worst case</i> (c). Valori in dB. . . .	82
B.1	Schema del filtro adattativo che implementa l’algoritmo LMS. Il simbolo * indica il complesso coniugato.	84
B.2	Andamento dell’errore e_k senza correzione (rosso) e con correzione di stima usando 3 <i>bit</i> per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale <i>typical case</i> , $\Gamma = 8$ dB.	86

B.3	Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 <i>bit</i> per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale <i>worst case</i> , $\Gamma = 8$ dB.	86
B.4	Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 <i>bit</i> per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale <i>typical case</i> , $\Gamma = 8$ dB.	87
B.5	Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 <i>bit</i> per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale <i>worst case</i> , $\Gamma = 8$ dB.	87
B.6	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 <i>bit</i>	88
B.7	Funzione distribuzione della P_{bit} in <i>assenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 <i>bit</i>	88
B.8	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>typical case</i> tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 <i>bit</i>	89
B.9	Funzione distribuzione della P_{bit} in <i>presenza di codifica</i> per diversi valori di Γ . Il caso "senza errori" è riportato con un quadrato sulla sinistra. Canale <i>worst case</i> tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 <i>bit</i>	89

Elenco delle tabelle

1.1	Valori del filtro (convoluzione di un <i>rect</i> e di un filtro Gaussiano) che opera sulla fase del segnale modulato per un periodo di campionamento $T_Q = \frac{T}{8}$. Filtro Gaussiano per $B_t T = 0.5$	12
1.2	Coefficienti del filtro equivalente h_{Tx} per $T_Q = T/8$, $B_t T = 0.5$	18
2.1	Modello del canale a 5 raggi nel caso TY.	21
2.2	Modello del canale a 5 raggi nel caso WC.	21
2.3	Due realizzazioni del canale g_{Ch} sui tempi $T_Q = T/8$	21
3.1	Decodifica <i>soft</i> . Percentuale di casi in cui la <i>path metric</i>	42
3.2	Decodifica <i>hard</i> . Percentuale di casi in cui la <i>path metric</i>	42
5.1	Valori dei parametri per la creazione dei tappi del canale come variabili aleatorie.	49
5.2	Coefficienti del filtro IIR h_{ds} utilizzato per generare lo spettro Doppler classico [6].	63

Introduzione

La seguente trattazione concerne la realizzazione di un modello *software* di un sistema di modulazione e demodulazione per applicazioni *wireless* basato sulla tecnica GMSK. Lo scopo del lavoro svolto è quello di progettare diverse soluzioni implementative e di valutarne l'efficienza tramite simulazioni di funzionamento.

A tal fine è stata presa come punto di partenza la struttura dati per i pacchetti descritta nel Paragrafo 1.1. La trasmissione, infatti, consiste nell'invio di pacchetti di lunghezza prefissata con un rate trasmissivo "in aria" di 300 *kbps*.

Nel Capitolo 1 verrà trattata la struttura del modulatore che, come descritto in [1] e [2] può essere visto come un modulatore di fase ma anche come un modulatore di ampiezza. Questa peculiarità si rivelerà molto utile in fase di progettazione degli stadi di demodulazione e decodifica. Il Capitolo 2 riguarderà i canali e il rumore che possono interessare questo tipo di trasmissioni e la realizzazione di un loro modello per testare la funzionalità dell'apparato in esame. Il Capitolo 3 tratterà invece tutto il sistema di demodulazione e decodifica. Dal momento che si è deciso di utilizzare una decodifica del codice convoluzionale di tipo *soft*, le metriche verranno fornite da un algoritmo *forward backward* (FBA) o da un *decision feedback equalizer* (DFE). In un primo momento si valuteranno le diverse soluzioni usando delle realizzazioni statiche del canale trasmissivo, in modo da avere una previsione di quali comportamenti aspettarsi nei test successivi; i risultati di queste verifiche saranno presentati nel Capitolo 4.

Infine, nel Capitolo 5 si affronterà il problema della tempo-varianza del canale con particolare attenzione al modello matematico per rappresentarlo e alle modifiche necessaria al sistema ricevente al fine di non risentire di questo comportamento. Le simulazione *software* con cui sono stati ricavati i risultati sono state scritte ed eseguite con linguaggio Matlab e C.

Capitolo 1

Modulazione

1.1 Struttura dei pacchetti

Vediamo in Fig. 1.1 come viene costruito un singolo pacchetto. Si parte da un *payload* di 342 *bit* di informazione non codificati a cui si aggiungeranno un preambolo e un postambolo di 4 *bit* per fare in modo che il codificatore convoluzionale parta e finisca in uno stato noto.

Come ingresso del codificatore troviamo quindi un pacchetto da 350 *bit* di informazione che all'uscita diventano 525 *bit* codificati, essendo il rate di codifica pari a $2/3$. L'*interleaver* agisce su questa sequenza senza variarne la lunghezza. A questo punto viene inserito il preambolo per la stima del canale. Esso è composto da una sequenza PN di 63 *bit* seguita da una ripetizione dei primi 7 per un totale di 70 *bit*. Infine, dovendo garantire all'algoritmo FBA uno stato finale in cui trovarsi per equalizzare il canale si aggiungono altri 6 *bit* alla fine della trasmissione. Per lo stato iniziale dello stesso, invece, si sfrutta la parte finale del preambolo poiché essa è nota e non viene trattata nella fase di decodifica.

Alla fine del procedimento si ha dunque un pacchetto in trasmissione di 600 *bit*.

1.2 Le modulazioni di fase

La modulazione MSK (*Minimum Shift Keying*) è un caso particolare di modulazione di frequenza (FSK) in cui, per contenere l'occupazione di banda del segnale, si impone alla modulazione FSK di soddisfare le seguenti condizioni:

- continuità di fase del segnale modulato in modo tale da limitare l'ampiezza delle componenti spurie ad alta frequenza dovute ai salti di fase;

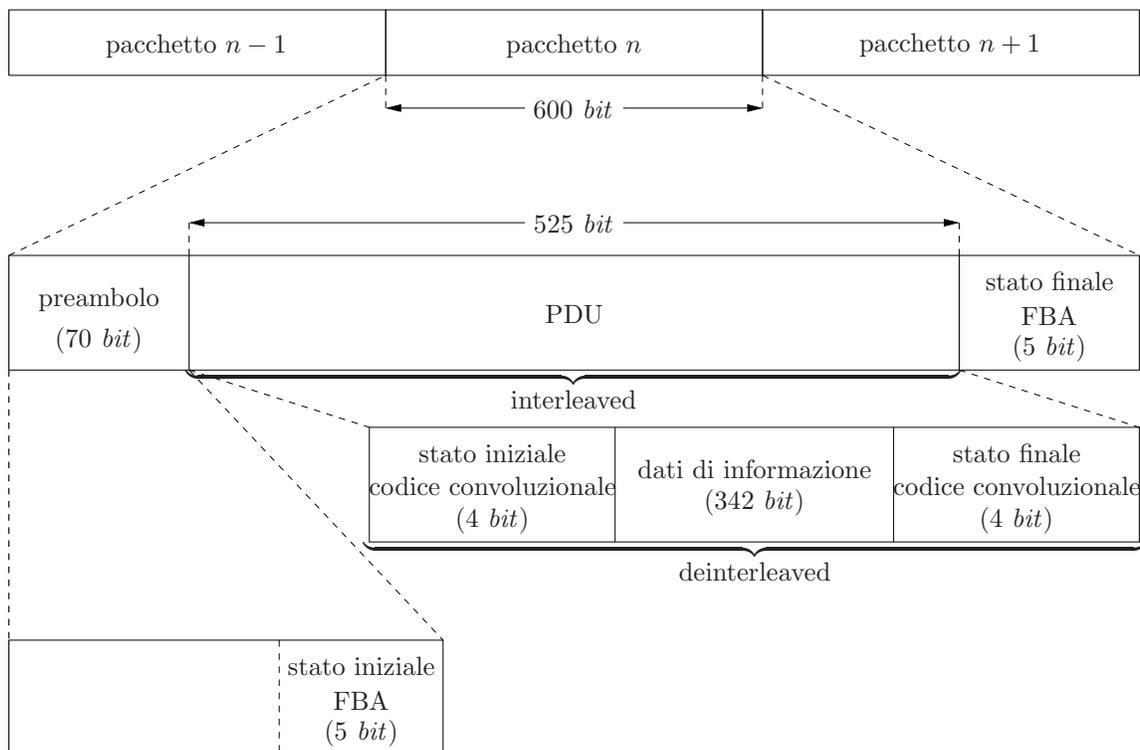


Figura 1.1: Struttura del pacchetto.

- ortogonalità dei segnali associati ai diversi valori del simbolo;
- separazione minima tra le frequenze associate ai diversi valori di simbolo per ridurre l'occupazione di banda.

L'imposizione di queste condizioni fornisce la struttura generale di una modulazione di tipo *Continuous Phase FSK* (CPFSK) il cui modulatore è schematizzato in Figura 1.2.

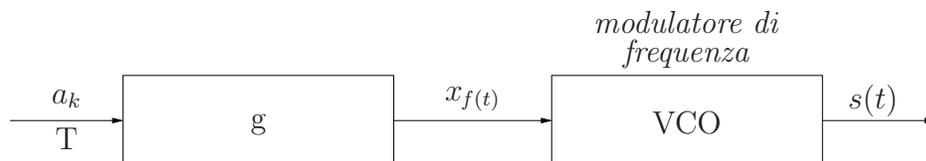


Figura 1.2: Modulatore CPFSK.

La sequenza di dati in ingresso $\{a_k\}$ con a_k appartenente all'alfabeto bipolare bilanciato $\{-1, 1\}$ è filtrata tramite un filtro interpolatore avente

risposta impulsiva $g(t)$ per produrre il segnale PAM x_f dato da:

$$x_f(t) = \sum_{k=-\infty}^{+\infty} a_k g(t - kT) \quad (1.1)$$

dove T è il periodo di simbolo. Nella modulazione MSK si ha:

$$g(t) = \frac{1}{2T} w_T(t) \quad (1.2)$$

dove

$$w_T(t) = \text{rect} \left(\frac{t - T/2}{T} \right) \quad (1.3)$$

è la finestra rettangolare di durata T . Il segnale x_f pilota l'ingresso di un *Voltage Controlled Oscillator* (VCO) la cui uscita $s(t)$ è data da:

$$s(t) = A \cos \left(2\pi f_0 t + 2\pi h \int_{-\infty}^t x_f(\tau) d\tau \right) \quad (1.4)$$

dove f_0 è la frequenza della portante e h è l'indice di modulazione. L'ampiezza dell'involuppo del segnale di uscita è pari a $A = \sqrt{2E_S/T}$ con E_S energia media del sistema. Senza perdita di generalità si assumerà d'ora in poi $A = 1$. Nella (1.4) il termine

$$\phi(t) = 2\pi f_0 t + 2\pi h \int_{-\infty}^t x_f(\tau) d\tau \quad (1.5)$$

rappresenta la fase del segnale modulato che risulta essere continua e composta dalla somma di una parte proporzionale al tempo e di una deviazione di fase $\Delta\phi(t)$.

La modulazione MSK si ottiene da una modulazione binaria CPFSK imponendo l'indice di modulazione $h = 0.5$ e una deviazione di frequenza $d = 1/(4T)$.

In Figura 1.3 si illustra la sostanziale differenza tra il segnale in uscita da un modulatore FSK e il corrispettivo ottenuto mediante modulazione MSK. Si nota con evidenza la continuità della fase che si riflette nella continuità della derivata prima del segnale stesso.

1.3 La modulazione GMSK

La GMSK (*Gaussian Minimum Shift Keying*) è una variazione della modulazione MSK in cui, per ridurre l'occupazione di banda del segnale modulato,

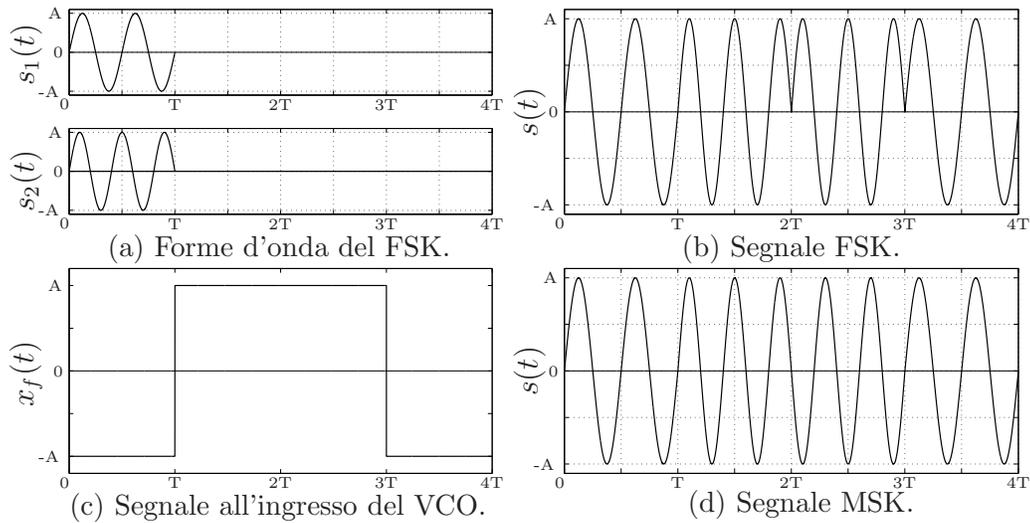


Figura 1.3: Confronto tra segnali FSK e MSK.

il segnale PAM x_f viene filtrato da un filtro gaussiano prima di pilotare il VCO. Questo tipo di modulazione è frequentemente utilizzato nei sistemi *wireless* di seconda generazione, come ad esempio il sistema GSM.

Oltre al pregio di poter reperire con facilità la componentistica preassemblata per poter implementare l'*hardware* necessario, tale modulazione gode di altri punti di forza tra i quali:

- involuppo costante del segnale modulato a beneficio degli stadi di potenza del trasmettitore e dell'incidenza della distorsione introdotta dal mezzo trasmissivo;
- tra le modulazioni di frequenza è quella a minore occupazione spettrale;
- alle alte frequenze lo spettro decade come $1/f^4$.

Tali caratteristiche denotano un'ottima efficienza spettrale della modulazione GMSK. Di contro, l'introduzione di un filtro gaussiano nel premodulatore porta ad un aumento dell'interferenza di intersimbolo (ISI) rispetto al caso del semplice MSK.

Nella seguente trattazione si assumerá sempre:

$$T = 1;$$

$A = 1$ (ampiezza modulatore di fase);

i simboli con $\tilde{}$ assumono valori binari in $\{0, 1\}$, ad esempio $\tilde{b}_n \in \{0, 1\}$, mentre il corrispondente simbolo $b_n \in \{-1, 1\}$.

1.4 Codifica convoluzionale

La codifica permette al ricevitore di correggere/rilevare gli errori sui *bit* introdotti dalla trasmissione. Questa operazione può essere pensata come la trasposizione del segnale su una costellazione di uno spazio con piú dimensioni di quello di partenza. Spostandoci in uno spazio con un numero maggiore di dimensioni, la distanza fra i punti aumenta e dunque si hanno migliori prestazioni nel rivelare/correggere gli errori dovuti alla trasmissione. Il guadagno introdotto da questi codici viene espresso come la quantità di energia di simbolo o rapporto segnale rumore che può essere “persa” a parità di probabilità di errore introducendo la codifica dei dati.

Nel sistema trattato la codifica utilizzata è detta convoluzionale. Un codice convoluzionale genera dei simboli codificati passando i *bit* d’informazione attraverso uno *shift register* a stati finiti come mostrato in Figura 1.5. Questo codificatore ha un *rate* (come rapporto del numero di *bit* di informazione per ogni parola di codice) pari a $1/2$. In particolare abbiamo:

$$\begin{aligned}\tilde{c}_n^{(1)} &= \tilde{b}_n \oplus \tilde{b}_{n-2} \oplus \tilde{b}_{n-3} \oplus \tilde{b}_{n-4} \\ \tilde{c}_n^{(2)} &= \tilde{b}_n \oplus \tilde{b}_{n-1} \oplus \tilde{b}_{n-4}.\end{aligned}\tag{1.6}$$

Il numero M di ritardi viene definito *constraint length* e sarà determinante per la costruzione del decodificatore. Nel nostro caso si ha $M = 4$. Per aumentare il *rate* del codice, nella nostra implementazione eseguiremo un *puncturing* sul flusso dati in uscita, ovvero verranno trasmessi i simboli:

$$\begin{cases} [\tilde{c}_m^{(1)} \tilde{c}_m^{(2)}] & \text{se } m = 2i, \quad i \in \mathbb{N} \\ \tilde{c}_m^{(1)} & \text{se } m = 2i + 1, \quad i \in \mathbb{N} \end{cases}.\tag{1.7}$$

Cosí facendo si otterrá un *rate* finale di $2/3$.

Quando un codificatore convoluzionale genera una parola di codice, essa dipende sia dal *bit* che in quel momento rappresenta l’ingresso del codificatore, sia dallo stato attuale

$$\mathbf{t}_n = [\tilde{b}_{n-1} \tilde{b}_{n-2} \tilde{b}_{n-3} \tilde{b}_{n-4}].\tag{1.8}$$

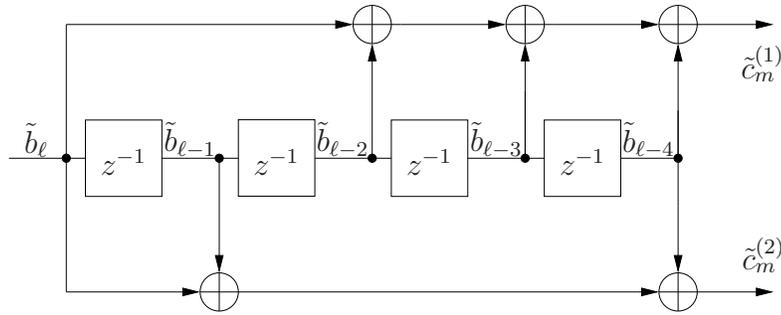


Figura 1.5: Schema a blocchi del codificatore convoluzionale.

costituito dagli M bit che si trovano negli *shift register*. Una rappresentazione comune del codificatore è quella a traliccio (*trellis*). Questa rappresentazione si basa sul fatto che da uno stato si possono raggiungere all'istante successivo 2 diversi stati, essendo 2 la cardinalità dell'alfabeto d'ingresso. Inoltre, dopo M istanti, il traliccio si ripete identico a se stesso. In Figura 1.6 vediamo le possibili transizioni di stato del codice descritto dalla (1.6) relative ad un istante di transizione. Lo stato è rappresentato da 4 bit come dalla (1.8).

Per migliorare le prestazioni della decodifica si imporrà che il codificato-

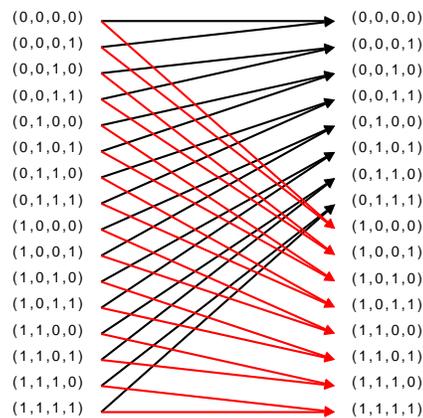


Figura 1.6: Possibili transizioni: in rosso relative ad un ingresso $\tilde{b}_n = 1$, in nero a $\tilde{b}_n = 0$.

re parta e finisca sempre nello stato $(0, 0, 0, 0)$. Per far questo bisognerà introdurre un *overhead* di 8 bit per ogni pacchetto.

1.5 Interleaver

I codici convoluzionali sono efficaci su canali AWGN, ovvero canali che aggiungono rumore bianco. Nei canali radio, in particolare quelli molto soggetti al fenomeno di *fading*, gli errori dovuti alla demodulazione tendono a presentarsi in *burst* (raffiche). In questo caso si ha spesso un degrado delle prestazioni dei codici che non riescono a correggere lunghe sequenze ininterrotte di dati corrotti.

Per ovviare a questo problema la codifica viene usata assieme ad un *interleaver* che mitiga gli effetti del *fading*. Il concetto base di questa tecnica è la dispersione dei *burst* in modo da diminuire la lunghezza massima delle sequenze di errori consecutivi. Le tecniche per riposizionare i dati sono molteplici e spesso vengono rappresentate tramite grafici che associano l'indice di posizione d'ingresso a quello d'uscita; un esempio è quello di Figura 1.7 relativo all'*interleaver* usato nel nostro sistema. L'*interleaver* utilizzato consiste in una matrice in cui i dati vengono scritti sequenzialmente per colonne e successivamente letti per righe e spesso indicato con l'acronimo TB/RL ovvero scrittura dall'alto (Top) al basso (Bottom) e lettura da destra (Right) a sinistra (Left). Esso avrà dimensioni fisse poiché, per ogni pacchetto, la quantità di bit di payload è sempre la stessa e si vuole evitare di introdurre sistematicamente dei *bit* di padding. In particolare, per lavorare su blocchi di 525 *bit*, verrà usata una matrice 21×25 .

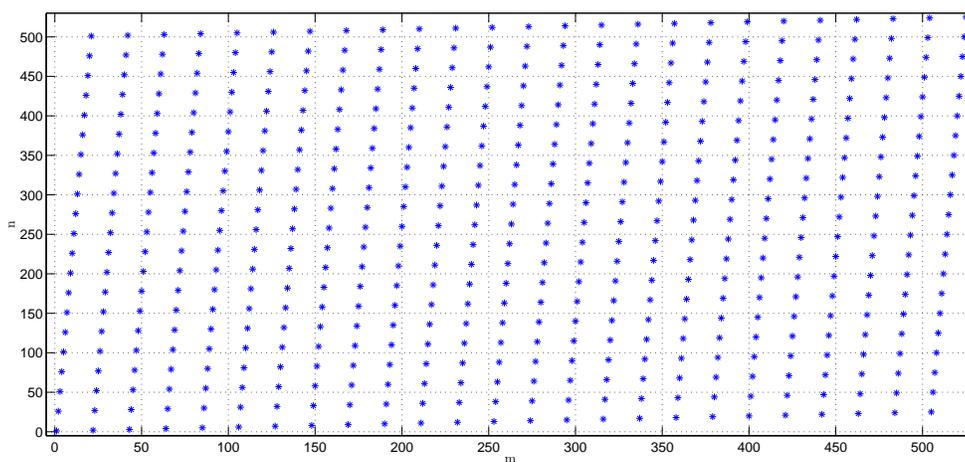


Figura 1.7: Relazione fra indice d'ingresso e d'uscita nell'*interleaver* usato.

1.6 Precodifica

I bit $\{\tilde{c}_k\}$ (codificati e dopo l'*interleaver*) da trasmettere vengono precodificati con il seguente metodo:

$$\tilde{a}_k = \tilde{c}_k \oplus \tilde{c}_{k-1}. \quad (1.9)$$

Come stato iniziale di questa precodifica, all'inizio di ogni preambolo e all'inizio di ogni blocco di *payload*, si assumerá sempre

$$\tilde{c}_{-1} = 0. \quad (1.10)$$

Dai simboli $\tilde{a}_k \in \{0, 1\}$ si passa ai simboli $a_k \in \{-1, 1\}$ secondo la relazione

$$a_k = 1 - 2\tilde{a}_k. \quad (1.11)$$

Gli a_k vengono passati all'ingresso del modulatore che effettua una trasformazione sui dati d'ingresso del tipo $d_k = d_{k-1}a_k$ ovvero $\tilde{d}_k = \tilde{d}_{k-1} \oplus \tilde{a}_k$. Di conseguenza i dati modulati sono scrivibili come:

$$\begin{aligned} \tilde{d}_k &= \tilde{d}_{k-1} \oplus \tilde{a}_k \\ &= \tilde{a}_0 \oplus \tilde{a}_1 \cdots \oplus \tilde{a}_k \\ &= \tilde{c}_{-1} \oplus \tilde{c}_k \end{aligned} \quad (1.12)$$

e considerando la (1.10) si deduce che $\tilde{d}_k = \tilde{c}_k$, ovvero $d_k = c_k$. Possiamo dunque dire che la "precodifica" e la somma dei dati introdotta dalla modulazione si compensano e i simboli \tilde{d}_k sono direttamente legati ai *bit* da trasmettere \tilde{c}_k .

1.7 Modulazione di fase

Come si vede in Figura 1.4, la modulazione di fase viene effettuata tramite il filtraggio dei dati sui tempi T_Q per un filtro dato dalla convoluzione nel tempo (cascata) della finestra rettangolare (1.2) e del filtro con risposta impulsiva gaussiana

$$g_G(t) = \frac{K}{\sqrt{2\pi}} e^{-\frac{K^2 t^2}{2}}, \quad K = \frac{2\pi B_t}{\sqrt{2\pi}} \quad (1.13)$$

essendo nel nostro caso $B_t T = 0.5T$ (banda a 3dB). Per realizzare quest'operazione si prende come risposta impulsiva del filtro g_P la versione campionata a T_Q dell'espressione analitica della convoluzione.

Sia

$$Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-s^2/2} ds \quad (1.14)$$

allora:

$$\begin{aligned}
 g_P(t) = g_I * g_G(t) &= \int_{-\infty}^{\infty} g_I(\tau)g_G(t - \tau)d\tau = \\
 &= \frac{1}{2} \int_0^T g_G(t - \tau)d\tau = \\
 &= \int_0^T \frac{\pi B_t}{\sqrt{2\pi \ln 2}} e^{-\frac{(t-\tau)^2}{2} \left(\frac{2\pi B_t}{\sqrt{\ln 2}}\right)^2} d\tau = \\
 &= \frac{1}{2} \left[Q\left(\frac{2\pi B_t(t - T)}{\sqrt{\ln 2}}\right) - Q\left(\frac{2\pi B_t t}{\sqrt{\ln 2}}\right) \right]. \quad (1.15)
 \end{aligned}$$

Nella Tabella 1.1 vengono riportati i valori dei 31 tappi del filtro di fase $g_P(t)$ realizzato sui tempi $\frac{T}{8}$ con banda normalizzata $B_t T = 0.5$ mentre la sua rappresentazione grafica è visibile in Figura 1.8.

$g_P(nT_Q)$	valore
$g(-11T_Q)$	$5.3005 \cdot 10^{-8}$
$g(-10T_Q)$	$5.9897 \cdot 10^{-7}$
$g(-9T_Q)$	$5.4620 \cdot 10^{-6}$
$g(-8T_Q)$	$4.0253 \cdot 10^{-5}$
$g(-7T_Q)$	0.0002402
$g(-6T_Q)$	0.001163
$g(-5T_Q)$	0.004588
$g(-4T_Q)$	0.0148
$g(-3T_Q)$	0.03926
$g(-2T_Q)$	0.08637
$g(-1T_Q)$	0.1593
$g(0T_Q)$	0.25
$g(1T_Q)$	0.3405
$g(2T_Q)$	0.4125
$g(3T_Q)$	0.4561
$g(4T_Q)$	0.4704

$g_P(nT_Q)$	valore
$g(5T_Q)$	0.4561
$g(6T_Q)$	0.4125
$g(7T_Q)$	0.3405
$g(8T_Q)$	0.25
$g(9T_Q)$	0.1593
$g(10T_Q)$	0.08637
$g(11T_Q)$	0.03926
$g(12T_Q)$	0.0148
$g(13T_Q)$	0.004588
$g(14T_Q)$	0.001163
$g(15T_Q)$	0.0002402
$g(16T_Q)$	$4.0253 \cdot 10^{-5}$
$g(17T_Q)$	$5.4620 \cdot 10^{-6}$
$g(18T_Q)$	$5.9897 \cdot 10^{-7}$
$g(19T_Q)$	$5.3005 \cdot 10^{-8}$

Tabella 1.1: Valori del filtro (convoluzione di un *rect* e di un filtro Gaussiano) che opera sulla fase del segnale modulato per un periodo di campionamento $T_Q = \frac{T}{8}$. Filtro Gaussiano per $B_t T = 0.5$.

Il segnale x_n uscente dal filtro gaussiano, il cui diagramma ad occhio è evidenziato in Figura 1.9, viene accumulato fornendo la deviazione di fase

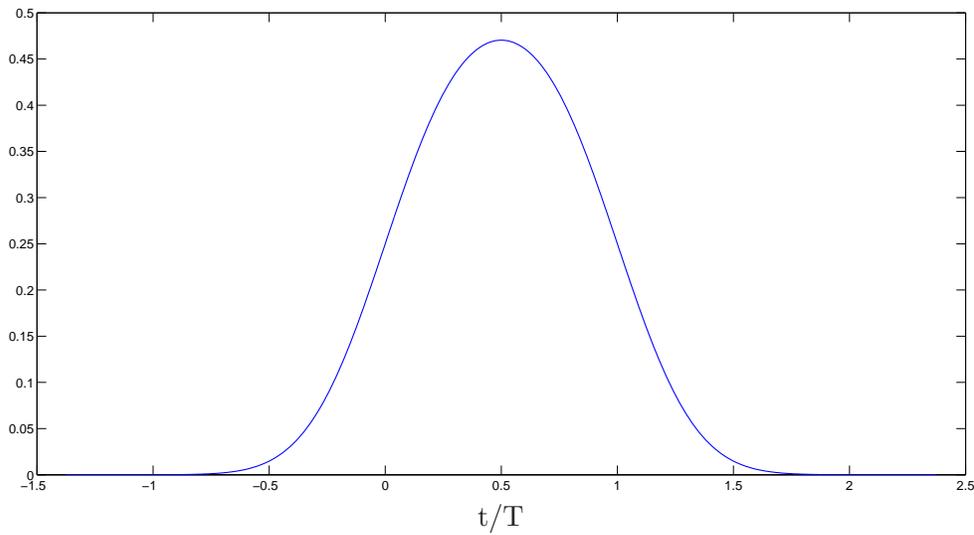


Figura 1.8: Risposta impulsiva del filtro $g_P(t)$.

$\Delta\varphi$ secondo l'equazione:

$$\Delta\varphi_n = \frac{\pi}{8} \sum_{i=-\infty}^n x(iT_Q) \quad (1.16)$$

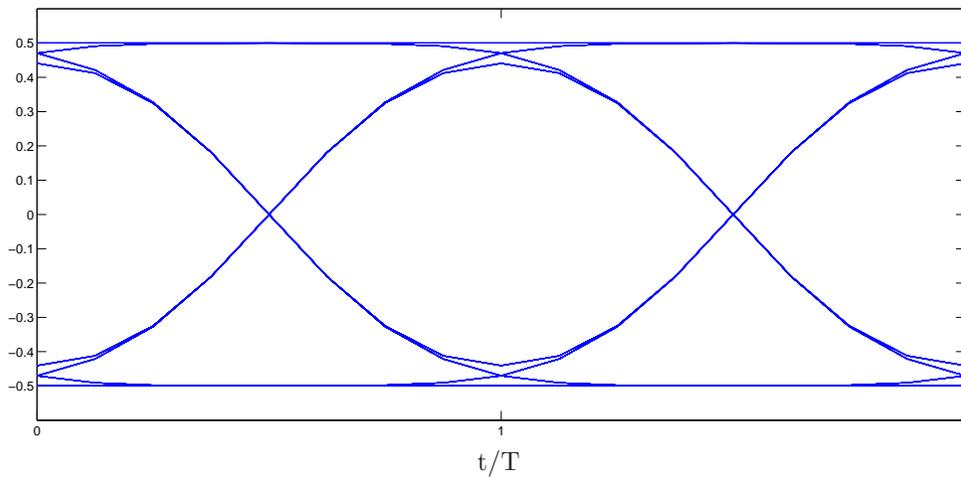


Figura 1.9: Diagramma ad occhio del segnale x_n .

L'ultimo passaggio per ottenere il segnale in banda base (rappresentazione complessa) è:

$$s^{bb}(nT_Q) = Ae^{j\Delta\varphi_n} = e^{j\Delta\varphi_n}, \quad (1.17)$$

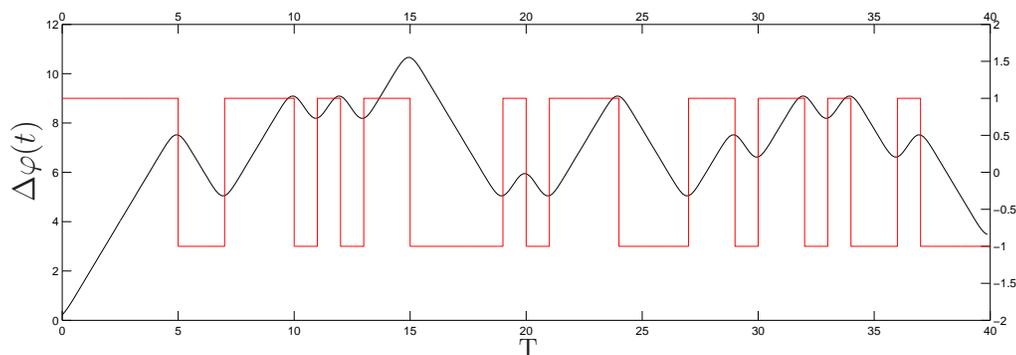


Figura 1.10: Andamento della deviazione di fase $\Delta\varphi(t)$ per la sequenza dati mostrata in rosso ($a_k \in \{-1, 1\}$).

essendo $A = 1$.

Un possibile andamento di $\Delta\varphi(t)$, ovvero l'interpolazione di $\Delta\varphi_n$, viene mostrato in Figura 1.10. Campionando nei punti di massimo di $\Delta\varphi_n$, in Figura 1.11 viene riportato l'insieme dei valori s^{bb} .

A differenza di quanto ci si aspetta da una modulazione GMSK con $B_t T =$

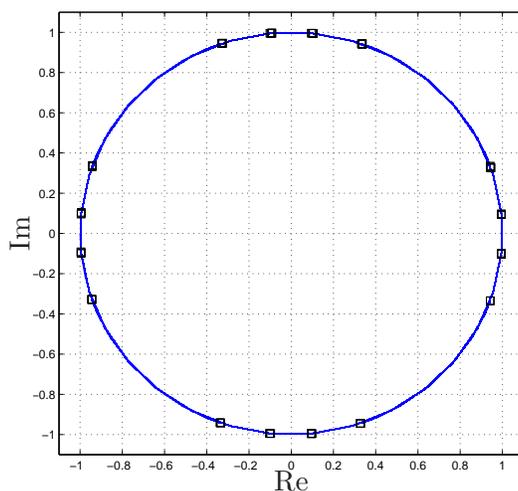


Figura 1.11: Rappresentazione sul piano complesso del segnale s^{bb} e dei valori della costellazione (quadrati neri).

0.5, la costellazione rappresentata è costituita da sedici punti anziché dodici. In particolare si nota che i punti in 1 , j , -1 e $-j$ vengono “sdoppiati”.

Questa irregolarità è dovuta al fatto che il modulatore è stato realizzato a $T/8$ e dunque nel processo di integrazione della deviazione di fase si hanno degli arrotondamenti rilevanti rispetto al caso analogico effettivo. Per vedere la costellazione nella sua versione classica sono state quindi eseguite delle prove con un modulatore realizzato a $T/128$ e $T/1024$. In questi casi, molto più onerosi dal punto di vista computazionale, il comportamento rispecchia molto più fedelmente quello analogico come si evince dalla Figura 1.12.

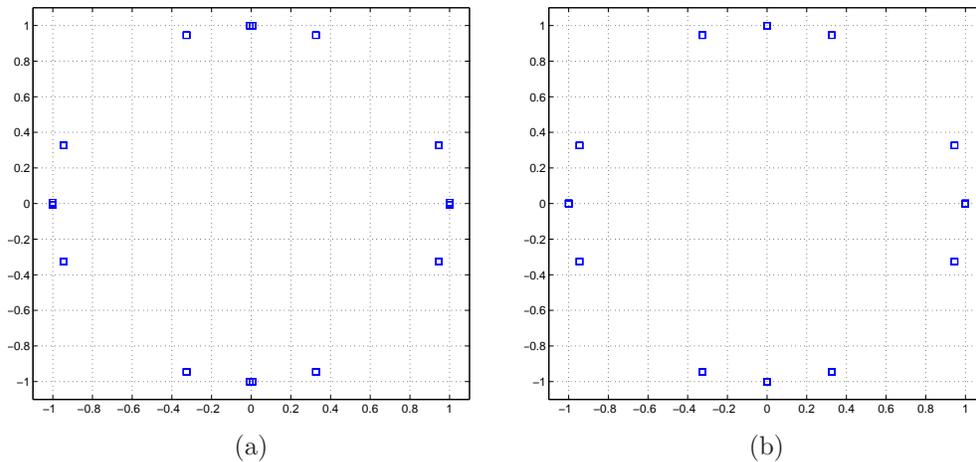


Figura 1.12: Costellazione di s^{bb} in opportuni istanti multipli di T con modulatore a (a) $T/128$ e (b) $T/1024$.

1.8 Spettro

Una volta ottenuto il segnale s^{bb} , che in un modulatore pratico verrebbe convertito in analogico, traslato in banda passante e trasmesso, è stata valutata la *power spectral density* (PSD) attraverso il periodogramma di Welch. Il risultato, normalizzato ad un valore massimo unitario, è mostrato in Figura 1.13.

1.9 Approssimazione

Come dimostrato da Laurent in [2], quando $B_t T \geq 0.3$ la modulazione GM-SK può essere approssimata con una modulazione di ampiezza. Vale infatti l'identità:

$$s^{bb}(nT_Q) = A e^{j\frac{\pi}{8} \sum_{i=-\infty}^n x(iT_Q)} = \sum_{k=-\infty}^{+\infty} (j^{k+1} d_k) h_{T_x}(nT_Q - kT) \quad (1.18)$$

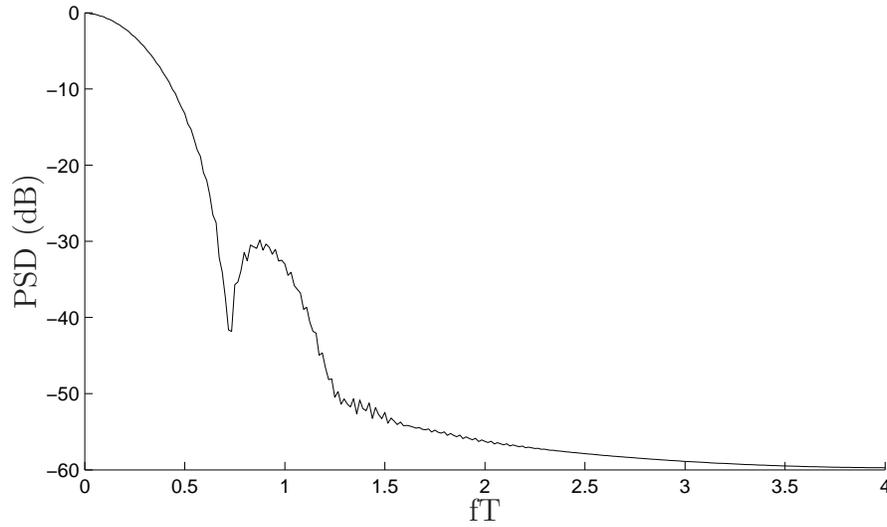


Figura 1.13: Densità spettrale del segnale modulato.

in cui $d_k = d_{k-1}a_k$ (analogamente a quanto visto nella (1.12)) e h_{Tx} è il filtro equivalente ricavato per $B_tT = 0.5$ attraverso le seguenti formule (si veda [3]).

Sia

$$g_0(t) = 1 + (t - 1.5T)Q(\sigma(t - 1.5T)) - (t - 0.5T) \cdot Q(\sigma(t - 0.5T)) - \frac{1}{\sigma\sqrt{2\pi}} \left[e^{-\frac{\sigma^2(t-1.5T)^2}{2}} - e^{-\frac{\sigma^2(t-0.5T)^2}{2}} \right] \quad (1.19)$$

con $\sigma = \pi/\sqrt{\ln 2}$.

Posto

$$g(t) = \frac{g_0(t) - g_0(0)}{g_0(2) - g_0(0)} 1_0(t) \quad (1.20)$$

$$c(t) = \begin{cases} \cos(\frac{\pi}{2}g(t)) & 0 \leq t \leq 2T \\ \cos(\frac{\pi}{2}g(-t)) & -2T \leq t \leq 0 \\ 0 & \text{altrove} \end{cases} \quad (1.21)$$

allora:

$$h_{Tx}(t) = c(t - 2T)c(t - T). \quad (1.22)$$

I coefficienti del filtro ricavati dalla (1.22) sono riportati in Tabella 1.2. L'uso di questa approssimazione semplifica la trattazione della decodifica ma introduce un errore. Per valutare questo errore sono stati realizzati due segnali: $s_{G,n}^{bb}$, ottenuto dalla modulazione GMSK di una sequenza dati, e $s_{L,n}^{bb}$,

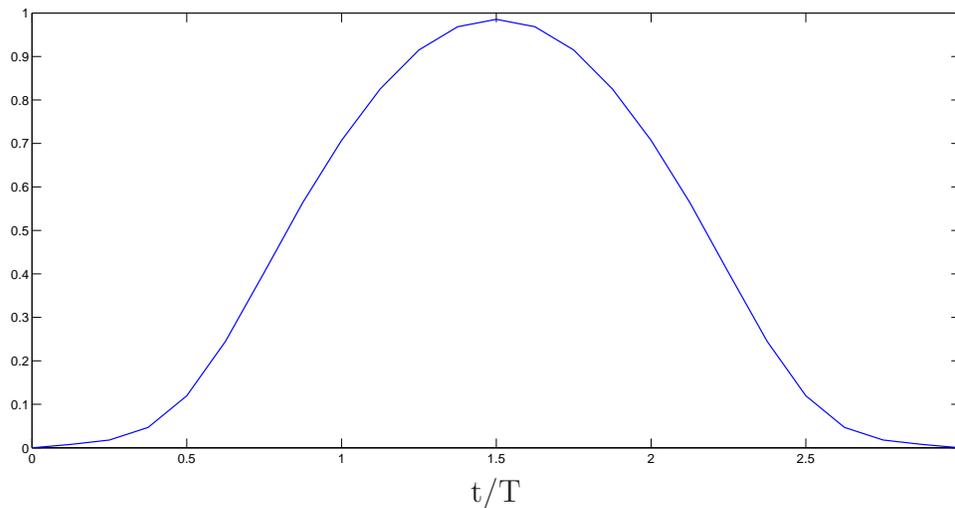


Figura 1.14: Risposta impulsiva del filtro equivalente h_{Tx} in (1.18).

ottenuto dalla modulazione della stessa sequenza dati attraverso lo schema equivalente di modulazione lineare (1.18). Il rapporto fra segnale ed errore di approssimazione risulta

$$\frac{\sum_n |s_{G,n}^{bb}|^2}{\sum_n |s_{G,n}^{bb} - s_{L,n}^{bb}|^2} = 117.31 = (20.69)_{\text{dB}} \quad (1.23)$$

A titolo di esempio, due spezzoni di parti immaginarie di questi due segnali vengono confrontate in Figura 1.15.

$h_{Tx}(nT_Q)$	valore	$h_{Tx}(nT_Q)$	valore
$h(0T_Q)$	0	$h(13T_Q)$	0.96842
$h(1T_Q)$	0.0077722	$h(14T_Q)$	0.91488
$h(2T_Q)$	0.017703	$h(15T_Q)$	0.82551
$h(3T_Q)$	0.046831	$h(16T_Q)$	0.70711
$h(4T_Q)$	0.11957	$h(17T_Q)$	0.5643
$h(5T_Q)$	0.24458	$h(18T_Q)$	0.40327
$h(6T_Q)$	0.40327	$h(19T_Q)$	0.24458
$h(7T_Q)$	0.5643	$h(20T_Q)$	0.11957
$h(8T_Q)$	0.70711	$h(21T_Q)$	0.046831
$h(9T_Q)$	0.82551	$h(22T_Q)$	0.017703
$h(10T_Q)$	0.91488	$h(23T_Q)$	0.0077722
$h(11T_Q)$	0.96842	$h(24T_Q)$	0
$h(12T_Q)$	0.98549		

Tabella 1.2: Coefficienti del filtro equivalente h_{Tx} per $T_Q = T/8$, $B_tT = 0.5$.

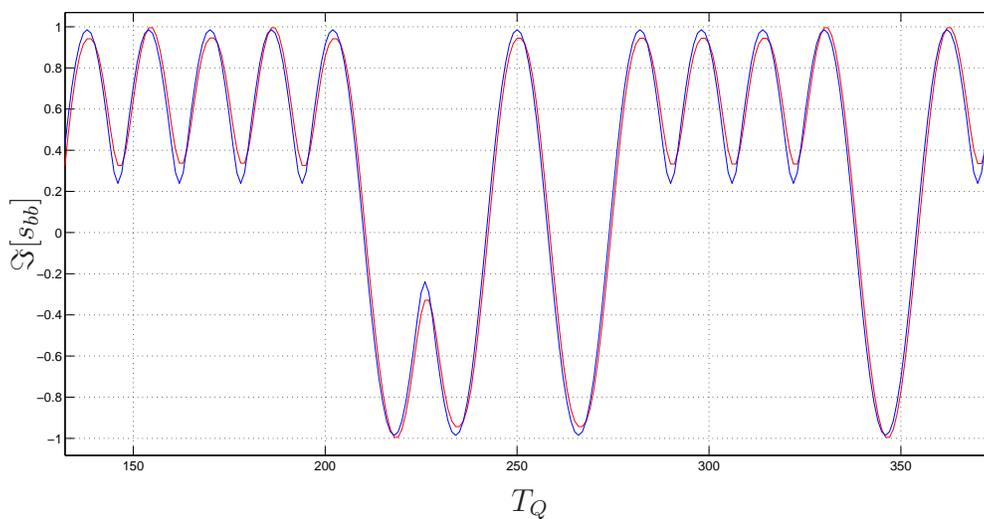


Figura 1.15: Parte immaginaria del segnale s_{bb} ottenuto con lo schema equivalente di Laurent (in blu) e con la modulazione GMSK (in rosso).

Capitolo 2

Canale

Il canale trasmissivo considerato è il canale radio. I canali radio possono riguardare diverse tipologie di trasmissione e avere quindi diversi modelli simulativi. Fra le più comuni ricordiamo le trasmissioni terrestri punto-punto, le trasmissioni terrestri mobili e le trasmissioni terra-satellite. Per il canale radio trattato nelle seguenti simulazioni si prenderanno in considerazione un caso tipico (TY) e un caso pessimo di funzionamento(WC).

Ogni tipo di canale annovera cinque possibili cammini del segnale elettromagnetico dal trasmettitore al ricevitore. Questo fenomeno è detto *multipath* ed è dovuto alla presenza di ostacoli nello spazio interessato dalla trasmissione. Si consideri la propagazione di un segnale a banda stretta in presenza di riflessioni. Qualora un cammino subisca una riflessione per la presenza di una superficie, una parte della sua potenza viene assorbita dalla superficie mentre il resto viene trasmesso in un'altra direzione. Se l' i -esimo cammino ha subito K_i riflessioni prima di giungere al ricevitore e se α_{ij} è il coefficiente di riflessione (in generale complesso) della j -esima riflessione dell' i -esimo cammino, il fattore di riflessione complessivo risulta

$$\alpha_i = \prod_{j=1}^{K_i} \alpha_{ij}. \quad (2.1)$$

Si ha dunque che, in generale, le ampiezze dei segnali corrispondenti a cammini diversi da quello diretto subiscono un'attenuazione dovuta alle riflessioni che si aggiunge all'attenuazione dovuta alla distanza fra trasmettitore e ricevitore.

La fase complessiva di ciascun cammino, inoltre, si può ottenere sommando la fase introdotta dalle varie riflessioni alla fase dovuta alla distanza percorsa; se N_c è il numero di cammini e d_i è la distanza percorsa dall' i -esimo cammino

di questi, la risposta impulsiva del canale radio risulta essere

$$g_{Ch}(\tau) = \Re \left[\frac{A_0}{A_{Tx}} \sum_{i=1}^{N_c} \frac{\alpha_i}{d_i} h^{(a)}(\tau - \tau_i) \right] \quad (2.2)$$

dove A_0 è l'ampiezza del segnale ricevuto a un metro dal trasmettitore, A_{Tx} è l'ampiezza del segnale trasmesso, $\tau_i = d_i/c$ è il ritardo dell' i -esimo cammino e $h^{(a)}(t)$ è la risposta impulsiva del filtro analitico usato per ottenere la rappresentazione in banda base.

L'involuppo complesso di (2.2) attorno alla frequenza della portante f_0 è dato da

$$g_{Ch}^{(bb)}(\tau) = \frac{2A_0}{A_{Tx}} \sum_{i=1}^{N_c} \frac{a_i}{d_i} e^{-j2\pi f_0 \tau_i} \delta(\tau - \tau_i) \quad (2.3)$$

Si tratta quindi di uno *shift* in frequenza del modello in banda base ottenuto grazie al termine di fase addizionale $e^{-j2\pi f_0 \tau_i}$ per l' i -esimo cammino.

Per i segnali a larga banda, con spettro centrato attorno alla frequenza f_0 , il modello del canale (2.3) viene riscritto in funzione del ritardo τ e del tempo t come

$$g_{Ch}^{(bb)}(t, \tau) = \sum_{i=1}^{N_c} g_i(t) \delta(\tau - \tau_i(t)) \quad (2.4)$$

dove g_i rappresenta l'ampiezza complessa dell' i -esimo cammino che arriva al ricevitore con ritardo τ_i . La (2.4) rappresenta il canale come un filtro lineare avente risposta impulsiva tempo-variante in cui la variabilità dei coefficienti è dovuta al reciproco movimento di trasmettitore e ricevitore e/o a variazioni dell'ambiente circostante.

2.1 Realizzazioni del canale

Per le simulazioni si considerano due realizzazioni tempo invarianti del canale: una per un caso TY e una per il caso WC. Le specifiche sulla potenza $M_{g_{Ch,i}}$ dei raggi e sui loro ritardi τ_i sono date nelle Tabelle 2.1 e 2.2 mentre la fase di ogni raggio è scelta come variabile aleatoria uniforme fra $-\pi$ e π .

Considerando un rate trasmissivo di 300 *kbps* si ha un periodo di simbolo $T = 3.3\bar{3}\mu s$ e dunque il periodo di campionamento nel modello interpolato sarà $T_Q = \frac{T}{8} = 0.41\bar{6}\mu s$. Da questi risultati segue che i ritardi possono essere modellati come variabili aleatorie uniformi sugli intervalli $[2T_Q, 10T_Q]$, $[5T_Q, 19T_Q]$, $[10T_Q, 29T_Q]$ e $[15T_Q, 29T_Q]$ con $T_Q = T/8$. Il canale è sempre normalizzato in modo da avere energia unitaria ovvero,

TY	Raggio 0	Raggio 1	Raggio 2	Raggio 3	Raggio 4
τ_i [μs]	0	$\mathcal{U}(1, 4)$	$\mathcal{U}(2, 8)$	$\mathcal{U}(4, 12)$	$\mathcal{U}(6, 12)$
$M_{g_{Ch,i}}$ [dB]	0	-6	-9	-12	-15

Tabella 2.1: Modello del canale a 5 raggi nel caso TY.

WC	Raggio 0	Raggio 1	Raggio 2	Raggio 3	Raggio 4
τ_i [μs]	0	$\mathcal{U}(1, 4)$	$\mathcal{U}(2, 8)$	$\mathcal{U}(4, 12)$	$\mathcal{U}(6, 12)$
$M_{g_{Ch,i}}$ [dB]	0	2	-3	-4	-4

Tabella 2.2: Modello del canale a 5 raggi nel caso WC.

se $g_{Ch,i}$, $i = 0, 1, 2, \dots, N - 1$ è la risposta impulsiva del canale, si ha $\sum_{i=0}^{N-1} |g_{Ch,i}|^2 = 1$. Le ampiezze dei raggi per le due realizzazioni sono riportate in Tabella 2.3 e in Fig. 2.1.

Oltre al canale trasmissivo, nel seguito della trattazione sarà utile considerare la risposta impulsiva del canale equivalente definita sui tempi T_Q come

$$g_{eq,i} = (h_{Tx} * g_{Ch})_i \quad (2.5)$$

riportiamo nelle Figure 2.2 e 2.3 gli andamenti delle risposte impulsive così ottenute.

Canale Typical case		Canale Worst case	
istante	ampiezza	istante	ampiezza
$0T_Q$	$0.9605 \cdot e^{j \cdot 0.8710}$	$0T_Q$	$0.4951 \cdot e^{j \cdot 2.3974}$
$3T_Q$	$0.2413 \cdot e^{-j \cdot 1.0008}$	$3T_Q$	$0.7846 \cdot e^{j \cdot 1.1742}$
$8T_Q$	$0.1209 \cdot e^{j \cdot 2.1990}$	$7T_Q$	$0.2481 \cdot e^{-j \cdot 1.8261}$
$14T_Q$	$0.0606 \cdot e^{-j \cdot 1.0629}$	$14T_Q$	$0.1971 \cdot e^{j \cdot 3.1313}$
$23T_Q$	$0.0304 \cdot e^{j \cdot 0.4766}$	$25T_Q$	$0.1971 \cdot e^{-j \cdot 2.8449}$

Tabella 2.3: Due realizzazioni del canale g_{Ch} sui tempi $T_Q = T/8$.

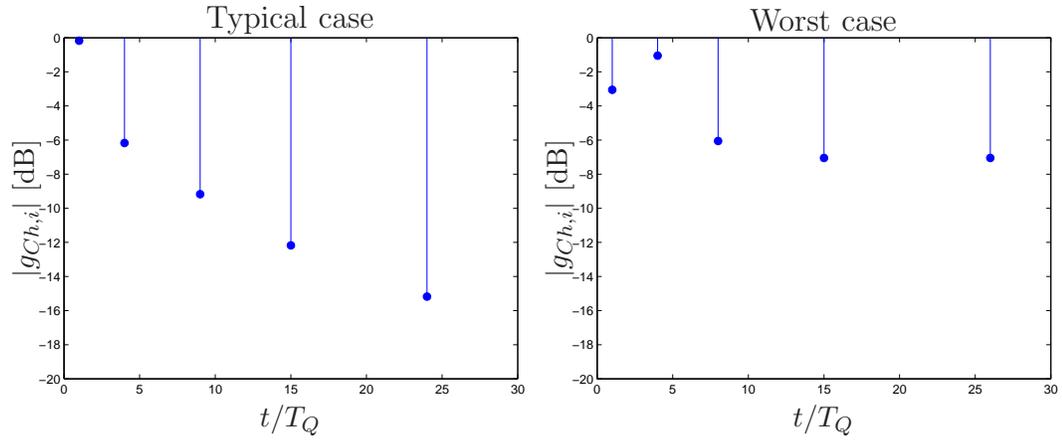


Figura 2.1: Modulo di due realizzazioni del canale g_{Ch} sui tempi $T_Q = T/8$.

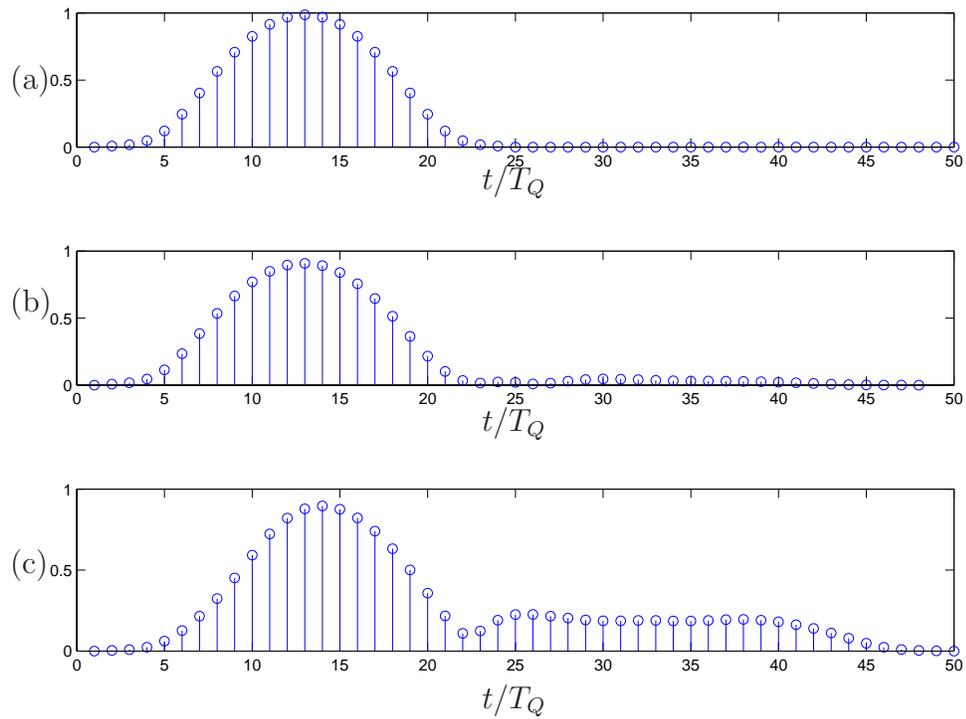


Figura 2.2: Modulo del canale equivalente come convoluzione dell'impulso h_{Tx} con il canale *ideal case* (a) *typical case* (b) e *worst case* (c).

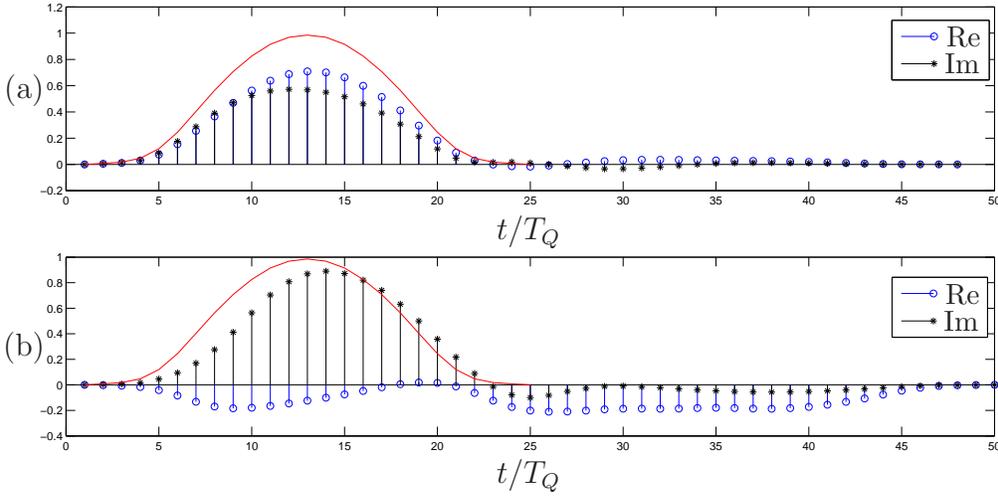


Figura 2.3: Impulso h_{Tx} (in rosso) e corrispondente g_{eq} (in blu e nero, rispettivamente, parte reale e immaginaria). Canale *typical case* (a) e *worst case* (b).

2.2 Rumore bianco

Al segnale in uscita dal canale si aggiunge del rumore AWGN complesso. Per effettuare le varie prove servirà variare il valore di SNR Γ definito come:

$$\Gamma = \frac{M_{s^{bb}}}{N_0(1/T)} = \frac{M_{s^{bb}}}{\sigma_w^2(T_Q/T)} \quad (2.6)$$

dove $M_{s^{bb}}$ è la potenza statistica di s^{bb} mentre N_0 è la PSD del rumore e σ_w^2 la sua varianza.

Essendo $s^{bb}(nT_Q)$ un segnale con modulo costante e unitario, si ha $M_{s^{bb}} = 1$ e dunque:

$$\Gamma = \frac{1}{\sigma_w^2 \cdot (1/8)} \Rightarrow \sigma_w^2 = \frac{8}{\Gamma}. \quad (2.7)$$

Il rumore viene generato come un processo Gaussiano complesso a media nulla e varianza pari a σ_w^2 . In base all'andamento della PSD di s^{bb} di Fig. 1.13, il rumore viene filtrato con un filtro passa basso avente banda di transizione $[0.75/T, 1/T]$ e attenuazione di 40 dB: in Figura 2.4 la sua risposta in frequenza.

La varianza del rumore dopo il filtro passa basso presenta un scalamento per un fattore dato da $\sum_i |h_{LPF,i}|^2$, ovvero dall'energia del filtro. Nel nostro caso

si ha:

$$\frac{\sigma_{\omega_C}^2}{\sigma_w^2} \cong 0.2116. \quad (2.8)$$

Per questo motivo, essendo

$$\sigma_{\omega_C}^2 \cong N_0 \frac{2 \cdot 0.85}{T} \quad (2.9)$$

si ha che le curve di errore tracciate in funzione di Γ sono scostate di circa $10 \cdot \log_{10} 1.7 \cong 2.3$ dB rispetto alle curve teoriche presenti in letteratura dove il rumore viene aggiunto direttamente sui tempi T ed ha quindi una banda di $0.5/T$ anziché $0.85/T$. In Figura 2.5 si possono vedere le densità spettrali di potenza del rumore prima e dopo il campionamento.

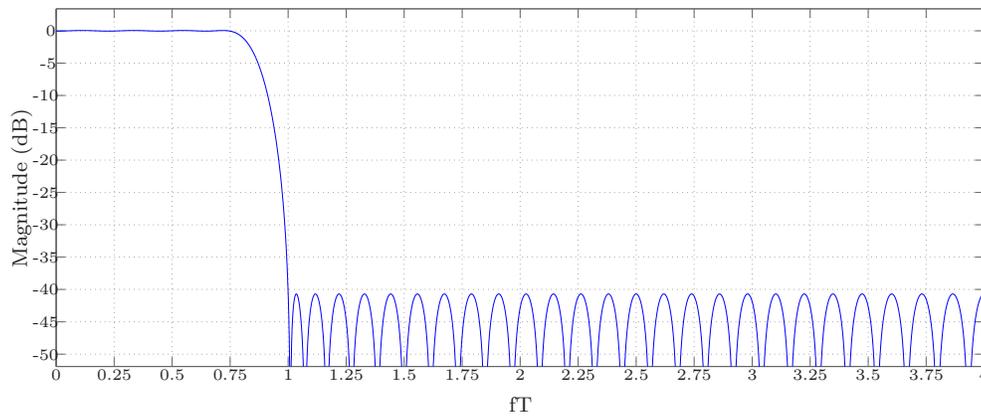


Figura 2.4: Risposta in frequenza del filtro applicato al rumore bianco.

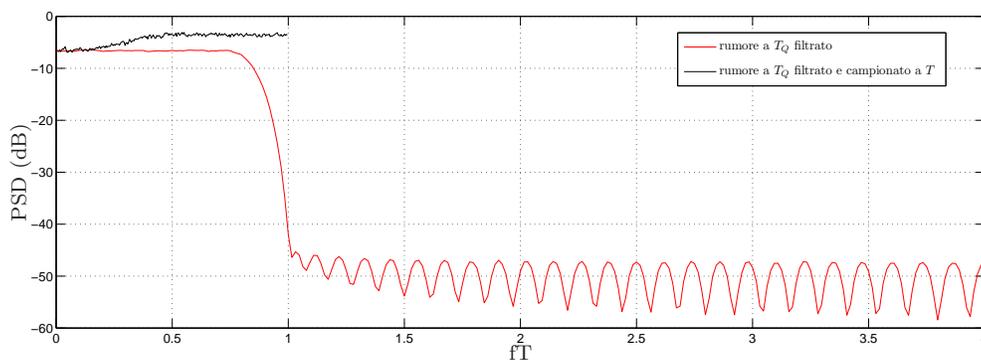


Figura 2.5: PSD del rumore bianco filtrato prima del campionamento (rosso) e dopo (nero).

Capitolo 3

Ricevitore

3.1 Sincronismo e stima del canale

Nella fase di ricezione, per passare dal dominio analogico a quello digitale viene effettuato un campionamento del segnale ricevuto. È dunque necessario trovare la fase ottima di campionamento e, una volta ottenuto il segnale campionato, si procederà a stimare la risposta impulsiva equivalente del sistema.

3.1.1 Sincronizzazione

Per trovare la fase ottima di campionamento si usa il metodo di correlazione applicato ad una sequenza PN di lunghezza $L = 63$ più una sua ripetizione lunga $N = 7$ simboli, pari alla lunghezza massima del canale equivalente, espressa in periodi di simbolo. Complessivamente avremo quindi un preambolo di $L + N = 70$ simboli che vengono trasmessi prima di ogni pacchetto dati.

Sia $r_{C,n}$ il segnale ricevuto campionato a $T_Q = T/8$ e $p_{C,k}$ il preambolo trasformato secondo la regola (1.11) e moltiplicato per $j^{(k+1)}$, ovvero

$$p_{C,k} = p_k j^{(k+1)}. \quad (3.1)$$

Di questi due segnali viene calcolata la cross-correlazione

$$\mathbf{r}_{rp}(m) = \frac{1}{L} \sum_{\ell=0}^{L-1} r_{C,\ell+m} p_{C,\ell}^* \quad 0 < mT_Q < m_{max}T_Q \quad (3.2)$$

dove l'apice * indica il coniugato e $m_{max}T_Q$ è la massima durata che si ritiene possa avere la risposta impulsiva equivalente, nel nostro caso $m_{max} = 63$.

A questo punto sia

$$m_{opt} = \arg \max_m |\mathbf{r}_{rp}(m)|. \quad (3.3)$$

allora la fase ottima di campionamento è data da:

$$t_0 = m_{opt}T_Q. \quad (3.4)$$

Il segnale in ricezione, campionato a T , su cui operare le successive elaborazioni sarà quindi:

$$r_k = r_{C, m_{opt} + 8k} \quad k = 0, 1, 2, \dots \quad (3.5)$$

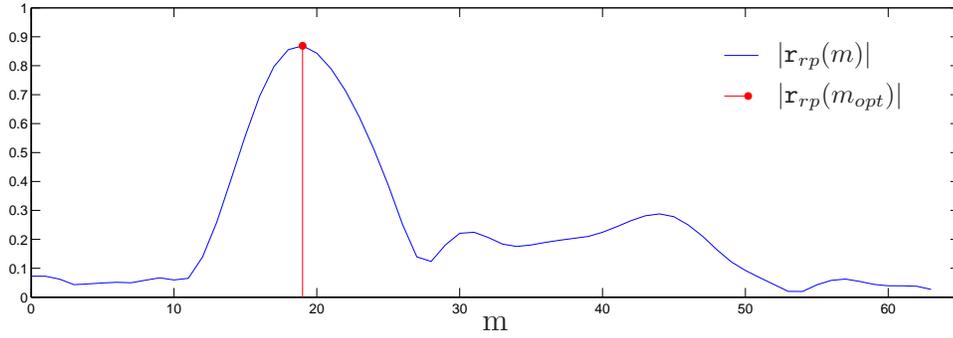


Figura 3.1: Modulo della cross-correlazione, $|\mathbf{r}_{rp}(m)|$, ed istante di modulo massimo. Canale *worst case*, $\Gamma = 10$ dB.

3.1.2 Stima del canale

La stima della risposta impulsiva equivalente viene eseguita sfruttando il preambolo trasmesso all'inizio di ogni pacchetto.

Il metodo utilizzato è detto di correlazione in quanto si basa sulla ricerca della massima correlazione fra il segnale noto inviato e il segnale ricevuto. Questo metodo, applicato al segnale r_k , fornisce i tappi stimati come:

$$\hat{g}_{eq,i} = \frac{1}{L} \sum_{k=0}^{L-1} r_k p_{C,(k-i)_{modL}}^* \quad i = -L_1, \dots, L_2. \quad (3.6)$$

Si noti che, non essendo la lunghezza della PN un multiplo di quattro, la notazione *modL* deve essere applicata alla ripetizione di $p_{C,k}$ già moltiplicato per $j^{(k+1)}$.

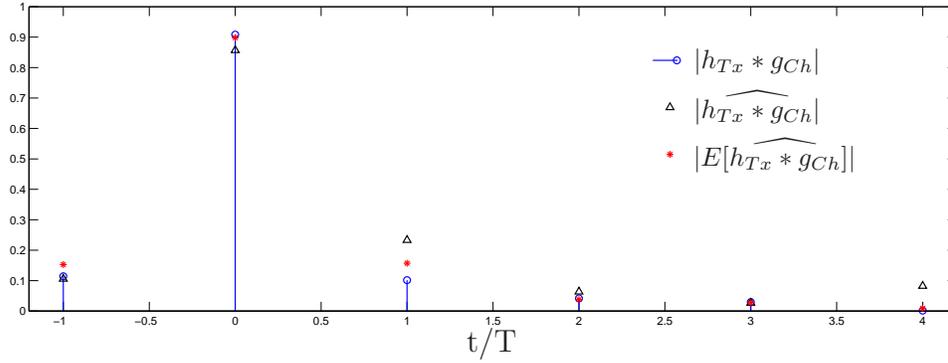


Figura 3.2: Stima del canale equivalente $\hat{g}_{eq,i}$ $i = -L_1, \dots, L_2$ con $L_1 = 1$ e $L_2 = 4$ per canale *typical case* e $\Gamma = 10$ dB.

Le stime ricavate col metodo di correlazione sono anch'esse rumorose, per questo si considera il rapporto normalizzato fra stima e rumore di stima Λ_n . Sia $\Delta \mathbf{g}$ il vettore errore di stima

$$\Delta \mathbf{g} = \hat{\mathbf{g}} - \mathbf{g},$$

omettendo il pedice 'eq' per semplicità e riferendoci alla definizione (2.5). La qualità della stima viene misurata dall'errore di stima rapportato all'energia della risposta impulsiva equivalente,

$$\Lambda_e = \frac{\|\mathbf{g}\|^2}{E[\|\Delta \mathbf{g}\|^2]} \quad (3.7)$$

D'altro canto bisogna considerare il rumore presente nel sistema da stimare misurato come:

$$\Lambda = \frac{M_{s^{bb}} \|\mathbf{g}\|^2}{\sigma_{\omega_C}^2}. \quad (3.8)$$

Dunque, ricordando che $M_{s^{bb}} = 1$ si ha che il rumore di stima normalizzato è dato da:

$$\Lambda_n = \frac{\Lambda_e}{\Lambda} = \frac{\sigma_{\omega_C}^2}{E[\|\Delta \mathbf{g}\|^2]}. \quad (3.9)$$

Il rumore di stima può essere calcolato avendo a disposizione i dati usati per sollecitare il sistema, la sua risposta impulsiva e la varianza del rumore (supposto bianco). In particolare nel nostro caso la sequenza dati consiste nel preambolo $p_{C,k}$ con matrice autocorrelazione Φ definita come

$$\Phi(i, n) = \sum_{k=N-1}^{N-1+L-1} p_{C,k-i}^* p_{C,k-n} \quad i, n = 0, 1, 2, \dots, N \quad (3.10)$$

Indicata con $\text{tr}[\mathbf{\Phi}]$ la traccia di $\mathbf{\Phi}$ e con \mathbf{I} la matrice identità $N \times N$, si ha che Λ_n è dato da:

$$\Lambda_n = \frac{1}{\frac{1}{L^2} \text{tr}[\mathbf{\Phi}] + \left\| \left(\frac{1}{L} \mathbf{\Phi} - \mathbf{I} \right) \mathbf{g} \right\|^2 \frac{1}{\sigma_{\omega_C}^2}}. \quad (3.11)$$

Se si considera la sola sequenza PN in ingresso e la moltiplicazione per $j^{(k+1)}$ come facente parte del canale, questo risultato può essere riscritto come

$$\Lambda_n = \frac{L}{N + \frac{1}{L} \left[\frac{M_{sbb} \|\mathbf{g}\|^2}{\sigma_{\omega_C}^2} + (N-2) \frac{\sum_{i=0}^{N-1} g_{eq,i} j^{-i}}{\sigma_{\omega_C}^2} \right]}. \quad (3.12)$$

Il confronto fra l'andamento teorico e quello pratico viene mostrato in Figura 3.3.

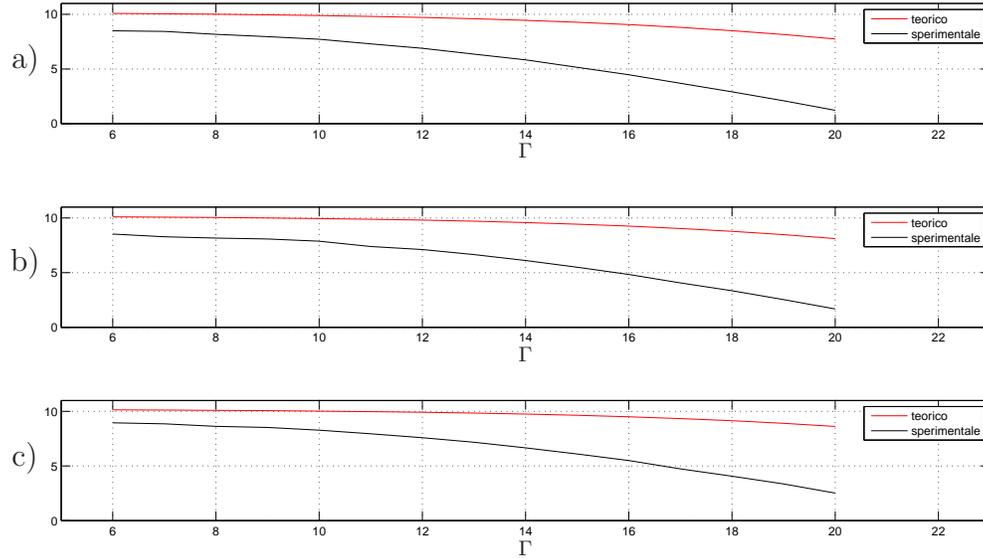


Figura 3.3: Confronto fra andamento teorico e reale di Λ_n per canale *ideale* (a), *typical case* (b) e *worst case* (c). Valori in dB.

Si nota che a bassi valori di SNR Γ l'andamento sperimentale è peggiore di quello teorico e questo è dovuto al fatto che il campionamento modifica la PSD del rumore AWGN con un fenomeno di *aliasing* e dunque la varianza cambia, come visto anche nella (2.9). Ad alti valori di Γ , invece, emerge l'errore della rappresentazione lineare di Laurent visto nella (1.23).

3.1.3 Stima della potenza del rumore bianco

Nel Paragrafo 3.2.1 si vedrà che, per rivelare il segnale ricevuto, oltre alla stima della risposta impulsiva del canale, è necessario avere una stima della potenza del rumore AWGN.

Per far questo consideriamo il segnale ricevuto (3.5) scritto come:

$$r_k = u_k + \omega_{C,k} \quad (3.13)$$

dove u_k è la componente utile,

$$u_k = \sum_{i=-L_1}^{L_2} g_{eq,i} p_{C,k-i}. \quad (3.14)$$

Questo segnale deve essere confrontato con la sua ricostruzione priva di rumore ottenuta dalla stima del canale e dalla *training sequence* nel seguente modo:

$$\hat{u}_k = \sum_{i=-L_1}^{L_2} \hat{g}_{eq,i} p_{C,k-i} \quad k = 0, \dots, L-1. \quad (3.15)$$

A questo punto si può stimare la varianza del rumore come:

$$\hat{\sigma}_{\omega_C}^2 = \frac{1}{L} \sum_{k=0}^{L-1} |r_k - \hat{u}_k|^2. \quad (3.16)$$

Il confronto fra stima e valore effettivo si può vedere in Figura 3.4.

3.2 Rivelazione dei dati

3.2.1 FBA

La rivelazione dei dati affetti dalla dispersione dovuta al modulatore e al canale può essere effettuata tramite l'algoritmo *Forward Backward* (FBA) per la cui struttura dettagliata si è fatto riferimento a [1].

Come dati d'ingresso di questo algoritmo troviamo il segnale ricevuto campionato negli istanti $t_0 + kT$, la stima del canale equivalente precedentemente calcolata e la stima della potenza del rumore bianco.

Lo scopo è quello di calcolare per ogni campione ricevuto due valori di *likelihood* L_k . Essi rappresentano la probabilità *a posteriori* dei simboli trasmessi, ovvero, dato che la sequenza di K simboli osservata in ricezione è $\mathbf{z}^K = \rho^K$, si ha:

$$L_k(\beta) = P[d_k = \beta | \mathbf{z}^K = \rho^K] \quad \beta \in \{-1, 1\} \quad (3.17)$$

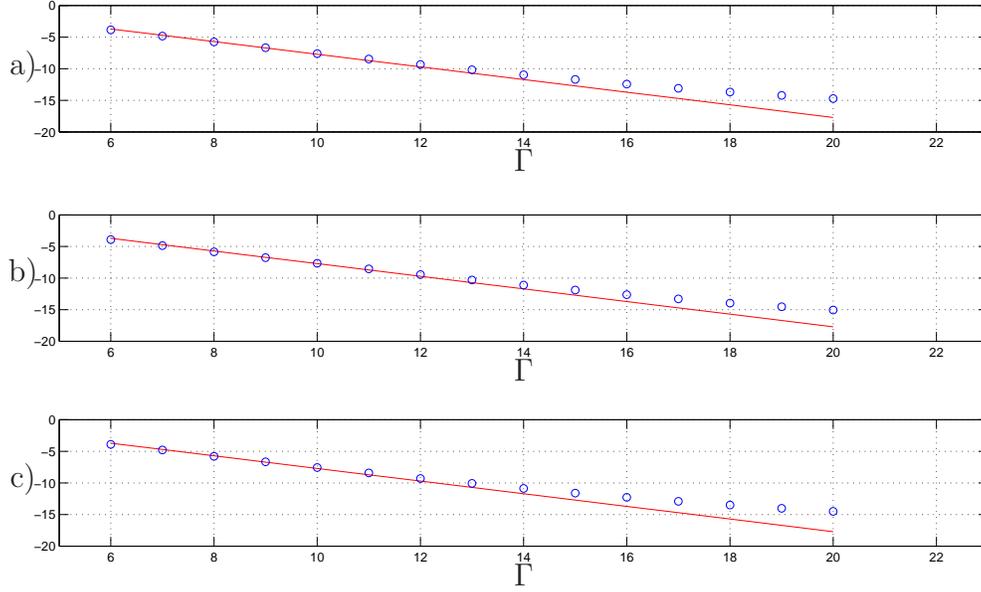


Figura 3.4: Confronto fra $\sigma_{\omega_C}^2$ (in rosso) e la sua stima (in blu) per canale *ideale* (a), *typical case* (b) e *worst case* (c). Valori in dB.

è evidente che al massimo di questi valori corrisponderá il minimo della probabilità di errore.

Per prima cosa definiamo tutti i possibili stati come:

$$\mathbf{s}_k = (d_{k+L_1}, d_{k+L_1-1}, \dots, d_k, \dots, d_{k-L_2+1}). \quad (3.18)$$

Anche in questo caso, poiché l'alfabeto è binario, ogni stato è raggiungibile da due possibili predecessori ed è predecessore di due stati.

Una volta costruito il *trellis*, vengono calcolati per ogni possibile transizione i valori u_k come funzione di \mathbf{s}_k e \mathbf{s}_{k-1} ,

$$u_k = \sum_{i=-L_1}^{L_2} \hat{g}_{eq,i} d_{k-i} \cdot j^{(k+1-i)} = u_k(\mathbf{s}_k, \mathbf{s}_{k-1}) \quad (3.19)$$

Successivamente, per ogni istante k , si calcolano i valori

$$\Theta_k(j|i) = p_{r_k}(\rho_k|j, i) \Pi(j|i) \quad (3.20)$$

dove

$$p_{r_k}(\rho_k|j, i) = \frac{1}{\pi \sigma_{\omega_C}^2} e^{-\frac{1}{\pi \sigma_{\omega_C}^2} |\rho_k - u_k|^2} \quad (3.21)$$

mentre

$$\Pi(j|i) = P[d_{k+L_1} = \beta] \quad (3.22)$$

rappresenta la probabilità *a priori* di un generico simbolo β , ammesso che sia possibile la transizione dallo stato $\mathbf{s}_{k-1} = \boldsymbol{\sigma}_i$ allo stato $\mathbf{s}_k = \boldsymbol{\sigma}_j$. Nel caso d'interesse, ovvero con simboli binari equiprobabili, si avrà $\Pi(j|i) = 1/2$ se è possibile la transizione e $\Pi(j|i) = 0$ altrimenti.

La fase successiva è il calcolo delle *Forward Metric* e delle *Backward Metric* definite ricorsivamente come:

$$F_k(j) = \frac{\sum_{\ell=1}^{N_S} \Theta_k(j|\ell) F_{k-1}(\ell)}{\sum_{n=1}^{N_S} \sum_{\ell=1}^{N_S} \Theta_k(n|\ell) F_{k-1}(\ell)} \quad j = 1, \dots, N_S; \quad k = 1, \dots, K \quad (3.23)$$

mentre

$$B_k(i) = \frac{\sum_{m=1}^{N_S} \Theta_k(m|i) B_{k+1}(m)}{\sum_{n=1}^{N_S} \sum_{\ell=1}^{N_S} \Theta_k(n|\ell) F_{k-1}(\ell)} \quad i = 1, \dots, N_S; \quad k = 1, \dots, K. \quad (3.24)$$

Queste quantità sono nella loro versione normalizzata perché ad ogni iterazione il prodotto di probabilità tende a 0 e il software usato per le simulazioni non ha sufficiente accuratezza.

Lo stato iniziale $\boldsymbol{\sigma}_0$ e quello finale $\boldsymbol{\sigma}_{\text{fin}}$ sono sempre noti poiché, come si è visto nel Paragrafo 1.1, i pacchetti hanno degli *header* e dei *trailer* appositamente costruiti per questo scopo. Possiamo dunque sfruttare la conoscenza di questi stati imponendo le seguenti uguaglianze nell'inizializzazione dell'algoritmo:

$$F_0(i) = \begin{cases} 1 & \boldsymbol{\sigma}_i = \boldsymbol{\sigma}_0 \\ 0 & \text{altrimenti} \end{cases} \quad (3.25)$$

e

$$B_{K+1}(i) = \begin{cases} 1 & \boldsymbol{\sigma}_i = \boldsymbol{\sigma}_{\text{fin}} \\ 0 & \text{altrimenti} \end{cases} \quad (3.26)$$

Usando le (3.23) e (3.24) troviamo le *State metric*

$$V_k(i) = \frac{F_k(i) B_k(i)}{\sum_{n=1}^{N_S} F_k(n) B_k(n)} \quad i = 1, \dots, N_S; \quad k = 1, \dots, K. \quad (3.27)$$

Infine da queste possiamo arrivare alle funzioni *likelihood*:

$$\mathbf{L}_k(\beta) = \sum_{\substack{i=1 \\ [\boldsymbol{\sigma}_i]_{1=\beta}}}^{N_S} V_k(i). \quad (3.28)$$

Poiché i simboli dell'alfabeto di trasmissione sono 1 e -1, come uscita del FBA troveremo i valori detti *log-likelihood ratio* (LLR) ℓ_k calcolati come:

$$\ell_k = \ln(L_k) = \ln\left(\frac{L_k(1)}{L_k(-1)}\right) = \ln(L_k(1)) - \ln(L_k(-1)). \quad (3.29)$$

Essi vengono passati all'ingresso del *deinterleaver*.

3.2.2 FBA - versione semplificata (Max-Log-MAP)

Poiché i simboli sono binari, i.i.d. e trasmessi su un canale con AWGN si può impiegare una versione semplificata dell'algoritmo FBA.

Queste semplificazioni si basano sulla proprietà dei logaritmi di trasformare i prodotti in somme e producono un algoritmo che non richiede la stima della potenza del rumore bianco. Tuttavia, a fronte di un algoritmo più semplice, si hanno delle prestazioni leggermente inferiori.

Le variazioni rispetto a quanto visto nella Sezione 3.2.1 sono (si indica con ρ_k l'osservazione o meglio il valore di r_k):

1. *Metriche di transizione del canale.*

$$\theta_k(j|i) = -|\rho_k - u_k|^2, \quad i, j = 1, \dots, N_S \quad (3.30)$$

essendo u_k definito come in (3.19) e supponendo che sia possibile la transizione dallo stato i allo stato j .

Per $k = K + 1$ si assume:

$$\theta_k(j|i) = 0, \quad i, j = 1, \dots, N_S \quad (3.31)$$

quando possibile la transizione dallo stato i allo stato j .

2. *Forward metric.*

$$\mathcal{F}_k(j) = \max_{\ell \in \{1, \dots, N_S\}} [\mathcal{F}_{k-1}(\ell) + \theta_k(j|\ell)], \quad j = 1, \dots, N_S; \quad k = 1, \dots, K. \quad (3.32)$$

Essendo noto lo stato iniziale σ_0 si impone:

$$\mathcal{F}_0(j) = \begin{cases} 0 & \sigma_j = \sigma_0 \\ -\infty & \text{altrimenti} \end{cases} \quad (3.33)$$

3. *Backward metric.*

$$\mathcal{B}_k(i) = \max_{m \in \{1, \dots, N_S\}} [\mathcal{B}_{k+1}(m) + \theta_{k+1}(m|i)], \quad i = 1, \dots, N_S; \quad k = 1, \dots, K. \quad (3.34)$$

Essendo noto lo stato finale σ_{fin} si impone:

$$\mathcal{B}_{K+1}(i) = \begin{cases} 0 & \sigma_i = \sigma_{\text{fin}} \\ -\infty & \text{altrimenti} \end{cases} \quad (3.35)$$

4. *Metriche di stato.*

$$\mathcal{V}_k(i) = \mathcal{B}_k(i) + \mathcal{F}_k(i) \quad i = 1, \dots, N_S; \quad k = 1, \dots, K. \quad (3.36)$$

5. *Funzioni log-likelihood.*

$$\ell_k(\beta) \cong \max_{\substack{i \in \{1, \dots, N_S\} \\ [\sigma_i]_1 = \beta}} \mathcal{V}_k(i). \quad (3.37)$$

6. *Log-likelihood ratio.*

$$\ell_k = \ell_k(1) - \ell_k(-1). \quad (3.38)$$

3.2.3 Prestazioni del FBA

Nelle Figure 3.5, 3.6 e 3.7 vengono confrontate le prestazioni dell'algoritmo *forward backward* normalizzato del FBA Max-Log-MAP valutate attraverso una *hard detection* sui valori LLR. Si può notare che esse sono molto simili.

3.2.4 DFE

Il *decision feedback equalizer* (DFE) è un equalizzatore non lineare di tipo *data aided*, ovvero basato sulla conoscenza della sequenza dei dati in ingresso, o almeno di una sua parte, detta sequenza di *training* che nel nostro caso consiste nel preambolo di ogni pacchetto. Nota la risposta impulsiva del canale e dato un numero sufficiente di simboli correttamente rivelati, il DFE ricostruisce l'ISI prodotta rispetto al simbolo da rivelare e la cancella dal campione associato. Le prestazioni ottenibili col DFE sono superiori a quelle offerte da un semplice equalizzatore lineare: si ha infatti che nei casi di elevata distorsione introdotta dal canale, l'equalizzatore lineare tende ad amplificare il rumore per portare l'impulso alla forma desiderata, mentre il DFE, essendo *data aided*, non soffre di tale lacuna poiché la cancellazione dell'ISI non modifica il rumore all'uscita dell'equalizzatore. La peculiarità di essere *data aided* tuttavia rende il DFE sensibile ai simboli non correttamente rivelati in quanto l'errata rivelazione di un simbolo produce un errato calcolo dell'ISI sui simboli successivi con la possibilità di incorrere in un effetto propagativo dell'errore potenzialmente catastrofico per le prestazioni del

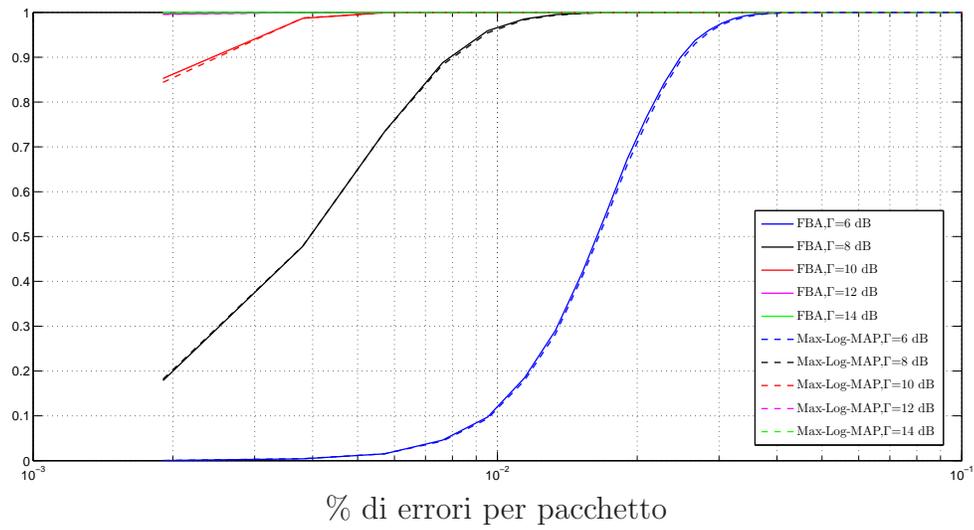


Figura 3.5: Funzione distribuzione delle percentuali di errore al variare di Γ , canale *ideale*.

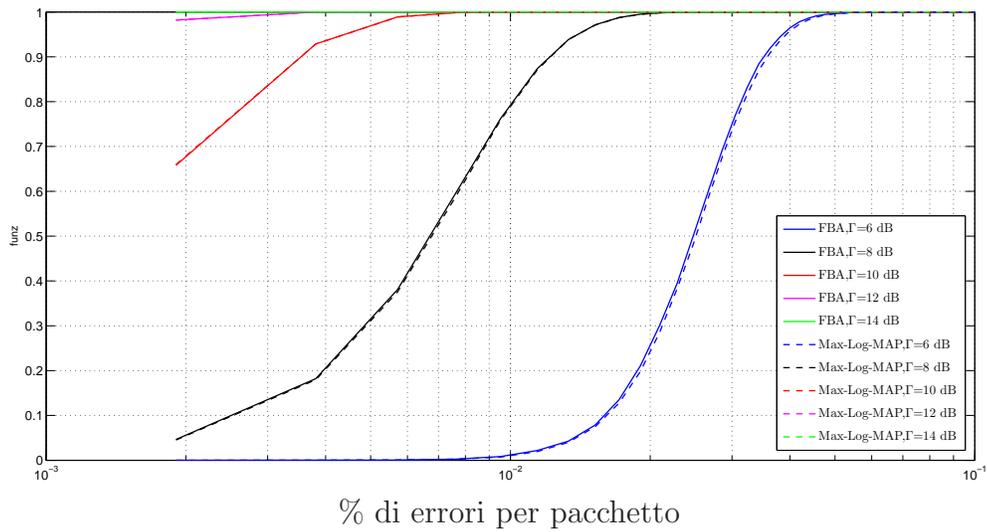


Figura 3.6: Funzione distribuzione delle percentuali di errore al variare di Γ , canale *typical case*.

sistema. Nella pratica però questa circostanza non è mai verificata e l'errata stima di un simbolo ha effetto solo su pochi simboli successivi riducendo le

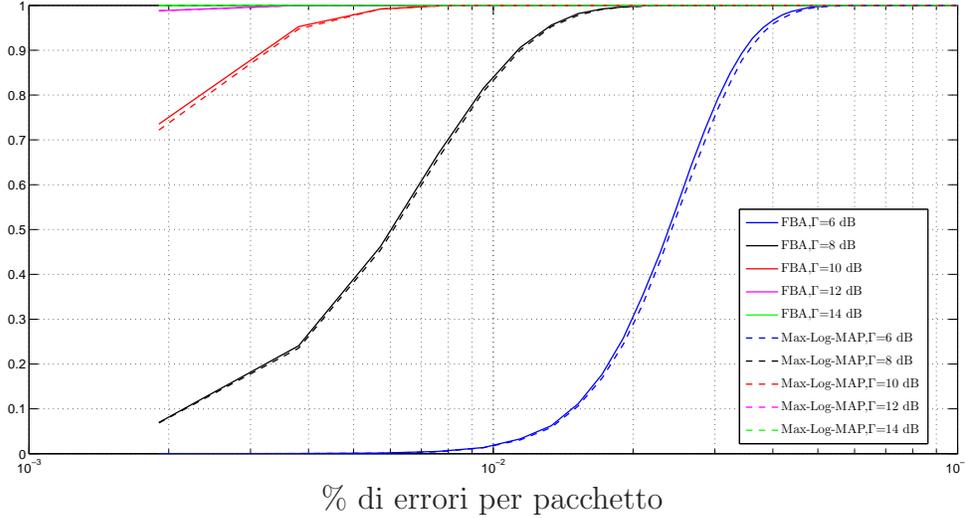


Figura 3.7: Funzione distribuzione delle percentuali di errore al variare di Γ , canale *worst case*.

prestazioni globali, in termini di rapporto segnale rumore Γ di 2-3 dB.

La struttura generale del DFE si compone di due filtri: il primo è denominato *feedforward filter* (FF), mentre il secondo è detto *feedback filter* (FB). Consideriamo il segnale campionato all'uscita del canale

$$r_k = \sum_{i=-\infty}^{+\infty} d_i j^{i+1} g_{eq,k-i} + \omega_{C,k} \quad (3.39)$$

ricordando che $\{g_{eq,k}\}$ è la risposta impulsiva equivalente del sistema a T . Si può dunque scrivere il segnale come la somma di una parte data dalla convoluzione dei dati con la risposta impulsiva del sistema e una parte di rumore bianco.

Avendo la risposta impulsiva un supporto finito $\{-L_1, -L_1 + 1, \dots, L_2 - 1, L_2\}$, la (3.39) può essere riscritta come:

$$r_k = (g_{eq,-L_1} d_{k+L_1} j^{(k+L_1+1)} + \dots + g_{eq,-1} d_{k+1} j^{(k+2)}) + \quad (3.40) \\ + g_{eq,0} d_k j^{(k+1)} + (g_{eq,1} d_{k-1} j^k + \dots + g_{eq,L_2} d_{k-L_2} j^{(k-L_2+1)}) + \omega_{C,k}.$$

Oltre al simbolo attuale d_k che si desidera stimare in base a r_k , nella (3.40) si individuano due termini in parentesi: uno dipendente solamente dai simboli passati e uno dipendente solo dai simboli futuri. Se i simboli passati e la risposta impulsiva fossero perfettamente noti, si potrebbe utilizzare lo schema

di cancellazione dell'ISI limitatamente ai soli postcursori.

Sostituendo i simboli passati con la loro stima, uno schema per cancellare

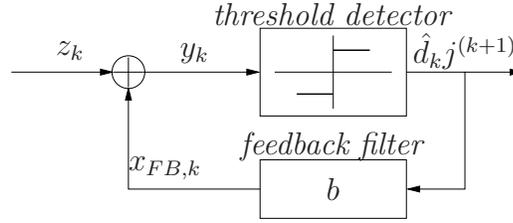


Figura 3.8: Schema della retroazione in un ricevitore DFE.

parzialmente l'ISI è riportato in Figura 3.8 dove, genericamente, il filtro di *feedback* (FB) ha risposta impulsiva $\{b_n\}$, $n = 1, \dots, M_2$ e uscita:

$$x_{FB,k} = b_1 \hat{d}_{k-1,j}^k + \dots + b_{M_2} \hat{d}_{k-L_2,j}^{(k-L_2+1)}. \quad (3.41)$$

Se $M_2 \geq L_2$, $b_n = -g_{eq,n}$ per $n = 1, \dots, L_2$, $b_n = 0$ per $n = L_2 + 1, \dots, M_2$ e $\hat{d}_{k-i} = d_{k-i}$ per $i = 1, \dots, N_2$ allora si ha la cancellazione dell'ISI dovuta ai postcursori.

La struttura generale di un DFE è riportata in Figura 3.9, in cui si possono

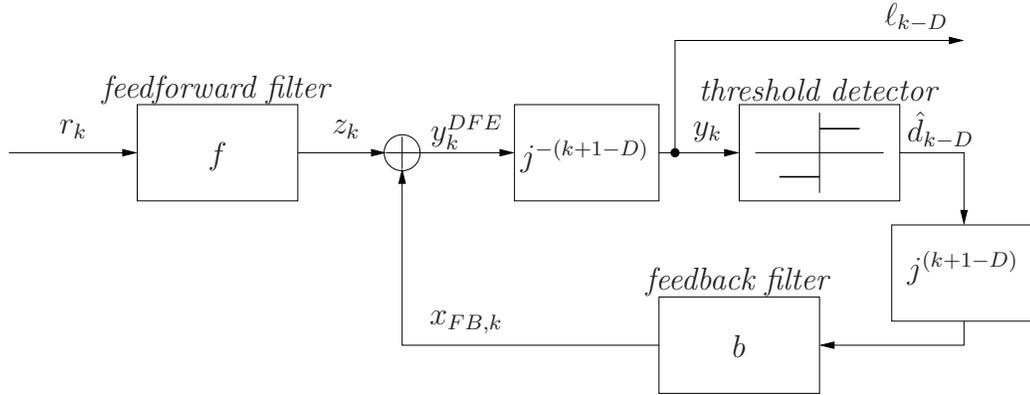


Figura 3.9: Struttura generale del ricevitore DFE.

individuare i due filtri e un ritardo D nella decisione:

1. Filtro di *feedforward* (FF) c , costituito da M_1 coefficienti,

$$z_k = \sum_{i=0}^{M_1-1} f_i r_{k-i} \quad (3.42)$$

2. Filtro di *feedback* (FB) b , costituito da M_2 coefficienti,

$$x_{FB,k} = \sum_{i=1}^{M_2} b_i \hat{d}_{k-i-D} j^{(k-i-D+1)} \quad (3.43)$$

inoltre:

$$y_k^{DFE} = z_k + x_{FB,k} \quad (3.44)$$

e

$$y_k = j^{-(k+1-D)} y_k^{DFE}. \quad (3.45)$$

L'obiettivo del filtro FF è quello di rendere a fase minima la funzione di trasferimento del sistema globale. Ne consegue che la risposta impulsiva globale $\psi_k = g_{eq} * f_k$ esibisce precursori molto piccoli, come si vede in Figura 3.10, dunque quasi tutta l'ISI viene cancellata dal filtro FB.

Riprendendo l'approssimazione lineare di Laurent, il segnale all'ingresso del DFE viene scritto come:

$$\begin{aligned} r_k &= \sum_{i=-\infty}^{+\infty} j^{i+1} d_i g_{eq,k-i} \\ &= (\text{cambio di variabile: } \ell = k - i \Rightarrow i = k - \ell) \\ &= \sum_{\ell=-\infty}^{+\infty} j^{k-\ell+1} d_{k-\ell} g_{eq,\ell} \\ &= \sum_{\ell=-L_1}^{L_2} j^{k-\ell+1} d_{k-\ell} g_{eq,\ell}. \end{aligned} \quad (3.46)$$

Ora, moltiplicando ambo i membri per $j^{-(k+1)}$, ovvero applicando una rotazione inversa rispetto a quella imposta in trasmissione, si ottiene:

$$j^{-(k+1)} r_k = \sum_{\ell=-L_1}^{L_2} j^{-\ell} d_{k-\ell} g_{eq,\ell} \quad (3.47)$$

$$= \underbrace{\sum_{\ell=-L_1}^{-1} j^{-\ell} d_{k-\ell} g_{eq,\ell}}_{\text{precursori}} + \underbrace{d_k g_{eq,0}}_{\text{componente utile}} + \underbrace{\sum_{\ell=1}^{L_2} j^{-\ell} d_{k-\ell} g_{eq,\ell}}_{\text{postcursori}} \quad (3.48)$$

$$(3.49)$$

Sarà dunque necessario imporre una moltiplicazione per $j^{-(k+1)}$ prima della *hard detection* di ciascun simbolo. Questa operazione, eseguita tenendo

conto del ritardo D , porge y_k secondo la (3.45) la quale è in effetti la metrica *soft* per il decodificatore.

Essendo i simboli c_k appartenenti ad un alfabeto binario bilanciato, il *threshold detector* è quello usato per una modulazione 2-PAM con regola di decisione:

$$\begin{aligned} \operatorname{Re}[y_k] > 0 &\Rightarrow \hat{d}_k = 1 \\ \operatorname{Re}[y_k] < 0 &\Rightarrow \hat{d}_k = -1. \end{aligned} \quad (3.50)$$

3.2.5 Calcolo dei coefficienti dei filtri per il DFE

Avendo una stima della risposta impulsiva del sistema $\{\hat{g}_{eq}\}$ e la funzione di autocorrelazione del rumore, si possono trovare i coefficienti ottimi dei filtri FF e FB applicando la teoria di Wiener sui filtri. Essa fornisce le seguenti equazioni (l'apice * denota il trasposto coniugato):

$$[\mathbf{p}]_p = \sigma_d^2 \hat{g}_{eq,D-p}^* \quad p = 0, \dots, M_1 - 1 \quad (3.51)$$

dove D è il ritardo sulla detection e σ_d^2 è la varianza dei simboli trasmessi (unitaria nel nostro caso).

$$[\mathbf{R}]_{p,q} = \sigma_c^2 \left(\sum_{\ell=-N_1}^{N_2} \hat{g}_{eq,\ell} \hat{g}_{eq,\ell-(p-q)}^* - \sum_{\ell=1}^{M_2} \hat{g}_{eq,\ell+D-q} \hat{g}_{eq,\ell+D-p}^* \right) + \mathbf{r}_{\omega_C,p-q} \quad (3.52)$$

con $p, q = 0, \dots, M_1 - 1$ ed assumendo per semplicità $\mathbf{r}_{\omega_C,p-q} = \delta_{pq} \sigma_{\omega_C}^2$ dove δ_{pq} indica la delta di Kronecker.

A questo punto si trova:

$$\mathbf{f}_{opt} = \mathbf{R}^{-1} \mathbf{p} \quad (3.53)$$

e

$$b_i = - \sum_{\ell=0}^{M_1-1} f_{opt,\ell} \hat{g}_{eq,i+D-\ell} \quad i = 1, \dots, M_2. \quad (3.54)$$

Inoltre il funzionale dell'errore secondo il criterio MSE è:

$$\begin{aligned} J_{min} &= \sigma_{\omega_C}^2 - \sum_{\ell=0}^{M_1-1} f_{opt,\ell} [\mathbf{p}]_{\ell}^* \\ &= \sigma_{\omega_C}^2 \left(1 - \sum_{\ell=0}^{M_1-1} f_{opt,\ell} \hat{g}_{eq,D-\ell} \right). \end{aligned} \quad (3.55)$$

Per l'ordine del filtro FB si è seguita la regola $M_2 = L_2 + M_1 - 1 - D$ che corrisponde ad una cancellazione dei postcursori da parte del FB stesso.

Il ritardo D è stato fatto variare da 0 a 20 periodi di simbolo cercando la soluzione che minimizza il funzionale J_{min} .

L'andamento della risposta impulsiva del sistema per $M_1 = 6$, $D = 4$ e $M_2 = 6$ è riportato in Figura 3.10

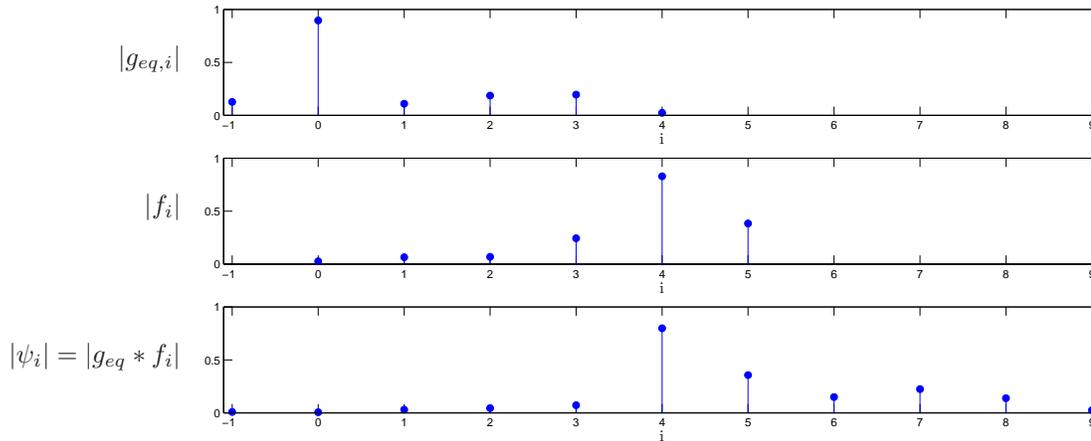


Figura 3.10: Modulo della risposta impulsiva equivalente con canale *worst case*, del filtro f e della loro convoluzione calcolati con $\Gamma = 10$ dB, $M_1 = 6$, $D = 4$ e $M_2 = 6$.

3.3 Deinterleaver

Il *deinterleaver* corrisponde ad una matrice di dimensioni uguali a quelle dell'*interleaver*, ovvero 21×25 , in cui i valori vengono scritti per righe e in seguito letti per colonne (esattamente l'opposto di quanto accadeva nell'*interleaver*). In questo modo le metriche *soft* vengono presentate nell'ordine corretto al decodificatore.

3.4 Viterbi decoder

L'ultimo elemento della catena di ricezione è il decodificatore di Viterbi con ingressi *soft*. Esso riceve infatti i valori ℓ_m che corrispondono al riordino eseguito dal *deinterleaver* sulle metriche *soft* calcolate dal FBA o dal DFE. Sfruttando il *trellis* del codificatore convoluzionale usato in trasmissione, l'algoritmo di Viterbi fornisce la sequenza più probabile di *bit* spediti.

A partire dal *trellis* viene calcolata per ogni stato σ_j , ad ogni istante k , una *path metric*

$$\Gamma(t_k = \sigma_j) = \min_{s_0, \dots, s_k = \sigma_j} \Gamma_k \quad (3.56)$$

essendo ricorsivamente definito

$$\Gamma_k = \Gamma_{k-1} + \|\boldsymbol{\ell}_k - f(\mathbf{t}_k, \mathbf{t}_{k-1})\|^2 \quad (3.57)$$

dove la *path metric* $\|\boldsymbol{\ell}_k - f(\mathbf{t}_k, \mathbf{t}_{k-1})\|^2$ rappresenta la distanza euclidea fra il vettore osservato (LLR) e le possibili uscite del codificatore convoluzionale. Dunque, costruiti i vettori LLR

$$\boldsymbol{\ell}_k = \begin{cases} [\ell_k^{(1)} \ \ell_k^{(2)}] & k = 2i, \ i \in \mathbb{N} \\ \ell_k^{(1)} & k = 2i + 1, \ i \in \mathbb{N} \end{cases} \quad (3.58)$$

viene valutata la loro distanza da:

$$f(\mathbf{t}_k, \mathbf{t}_{k-1}) = \begin{cases} [c_k^{(1)} \ c_k^{(2)}] & k = 2i, \ i \in \mathbb{N} \\ c_k^{(2)} & k = 2i + 1, \ i \in \mathbb{N} \end{cases} \quad (3.59)$$

Arrivati all'ultimo valore osservato, si trova la *path metric* con costo minore. Dalla transizione che l'ha scaturita si risale al piú probabile penultimo *bit* trasmesso. Ripetendo questo procedimento un numero adeguato di volte, seguendo sempre le transizioni indicate dalle *path metric* con costo minimo, si crea a ritroso la sequenza trasmessa piú probabile. Una caratteristica di questo procedimento è che, dopo aver ripercorso all'indietro un certo numero di passi nel *trellis*, il percorso tende a stabilizzarsi, ovvero, si percorrono le stesse transizioni indipendentemente dallo stato finale scelto.

Come nel caso del FBA lo stato iniziale e quello finale sono forzati in trasmissione come descritto nel Paragrafo 1.1.

3.4.1 Valutazione delle prestazioni del codice

Per valutare il guadagno apportato dal codice convoluzionale è stato realizzato uno schema trasmissivo semplificato. Si tratta di una sequenza di *bit* codificata con il codice convoluzionale, riordinata tramite *interleaver*. Ai simboli, appartenenti all'alfabeto $\{-1,1\}$, viene aggiunto del rumore Gaussiano bianco reale. A questo punto, sempre restando sui tempi T , si passa il segnale ricevuto al *deinterleaver* la cui uscita costituisce i valori *soft* per il decodificatore di Viterbi. Per avere un termine di paragone, parallelamente

a questa catena si usa uno schema con un semplice *threshold detector* la cui regola di decisione è:

$$\begin{aligned} r_k > 0 &\Rightarrow \hat{b}_\ell = 1 \\ r_k < 0 &\Rightarrow \hat{b}_\ell = -1. \end{aligned} \quad (3.60)$$

La struttura viene schematizzata in Figura 3.11.

Con questa struttura sono state svolte prove con singoli pacchetti da 3150000

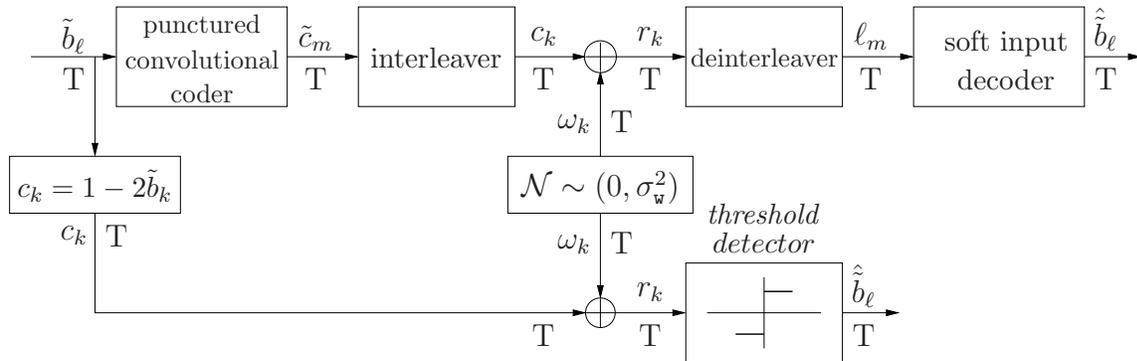


Figura 3.11: Schema a blocchi della struttura utilizzata.

bit di *payload* nei casi di stati iniziale e finale noti e non noti al decodificatore. Lo stesso è stato fatto per sequenze di 15815 pacchetti ciascuno con un *payload* di 342 *bit*. Queste prove sono state fatte anche per una decodifica *hard* del codice.

In Figura 3.12 le prestazioni del codice confrontate con quelle del ricevitore *threshold detector*. Nel caso di decodifica *soft* si può notare un guadagno di circa 6 dB con BER=10⁻⁵. Le differenze fra i casi di stati iniziali e finali noti o non noti e di pacchetti lunghi o corti sono pressoché nulle per la decodifica *hard* mentre sono leggermente più apprezzabili nella decodifica *soft*.

Queste considerazioni sono supportate anche dalle Tabelle 3.1 e 3.2; in esse si mostrano le percentuali dei casi in cui le *path metric* minime alla fine del pacchetto corto sono diverse da quella corretta nel caso di trasmissione senza stato iniziale/finale noto. Queste percentuali sono molto basse e inoltre bisogna considerare il fatto che, essendo relative all'ultima transizione sul pacchetto, rappresentano il caso peggiore di affidabilità del *trellis*; nell'algoritmo di Viterbi, infatti, maggiore è la profondità del *trellis* percorsa a ritroso per prendere la decisione sul *bit*, minore è la probabilità di errore. In questo caso la profondità è nulla trattandosi della decisione sull'ultimo *bit* del pacchetto.

SNR (dB)	6	7	8	9	10	11	12	13	14
% <i>path metric</i> errate	4	1.5	0.5	0.25	0.18	0	0	0	0

Tabella 3.1: Decodifica *soft*. Percentuale di casi in cui la *path metric* minima dello stato finale non è quella corretta.

SNR (dB)	6	7	8	9	10	11	12	13	14
% <i>path metric</i> errate	5	1.75	0.75	0.25	0.23	0.13	0	0	0

Tabella 3.2: Decodifica *hard*. Percentuale di casi in cui la *path metric* minima dello stato finale non è quella corretta.

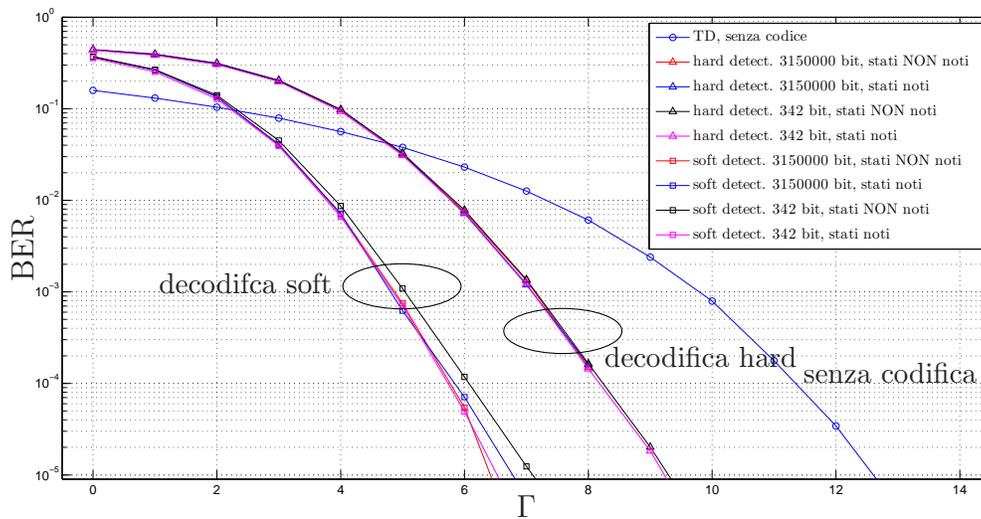


Figura 3.12: Probabilità di errore in funzione di Γ per lo schema di Fig. 3.11.

Capitolo 4

Valutazione delle prestazioni

4.1 Ricevitore con FBA

I seguenti risultati sono stati ottenuti dalla simulazione di trasmissione e ricezione di 16000 pacchetti con canale tempo invariante.

Dalla Figura 4.1 si evince un guadagno apportato dal codice di circa 5.5 dB nei casi di canale ideale e *typical case* e di circa 6 dB per il canale *worst case*. In Figura 4.2 sono riportate le prestazioni del FBA mentre in Figura

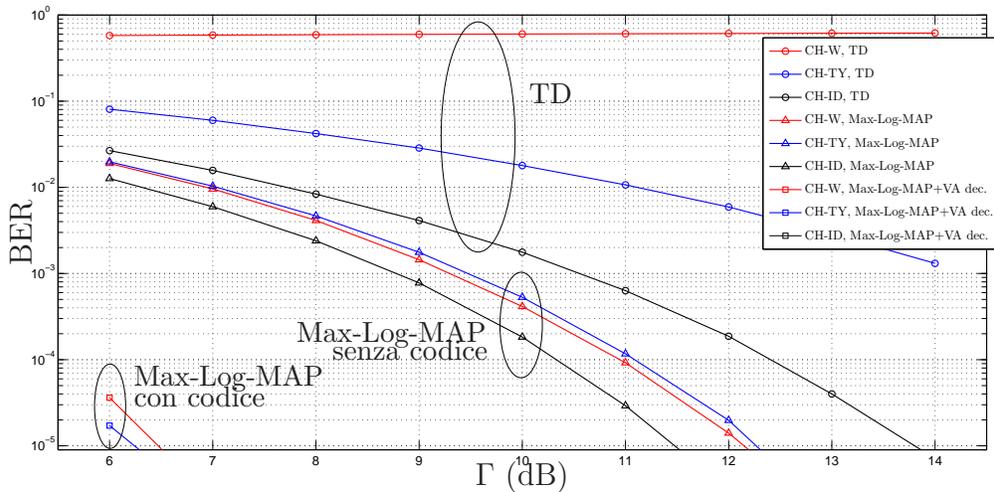


Figura 4.1: Canali noti. Probabilità di errore in funzione di Γ con *threshold detector* e con algoritmo *forward backward* (Max-Log-MAP) con e senza codifica (il caso di canale ideale con codifica è tutto al di sotto del valore 10^{-5}).

4.3 quelle del FBA semplificato (Max-Log-MAP) riprese da Figura 4.1: non si notano grosse differenze nelle prestazioni.

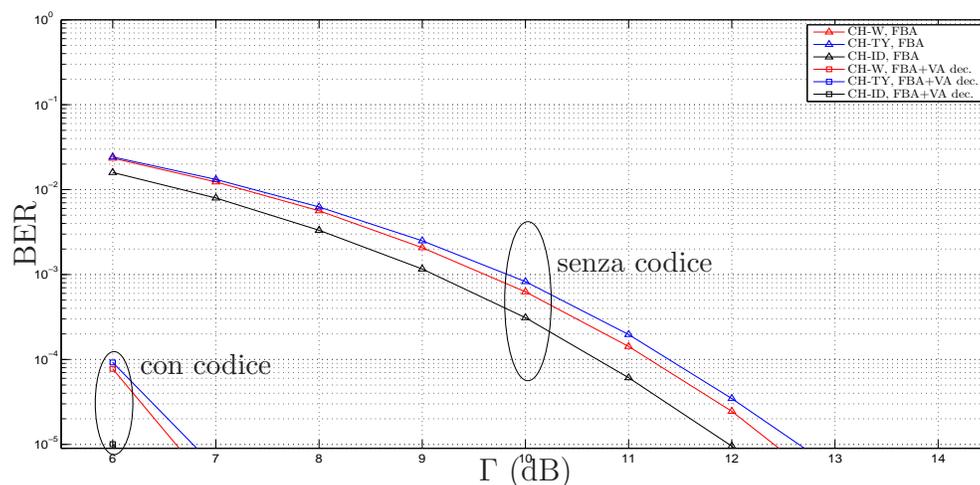


Figura 4.2: Canali stimati. Probabilità di errore in funzione di Γ con *hard detection* dei LLR ottenuti dal FBA normalizzato (senza codifica) e con *soft detection* tramite algoritmo di Viterbi (decodifica del codice convoluzionale).

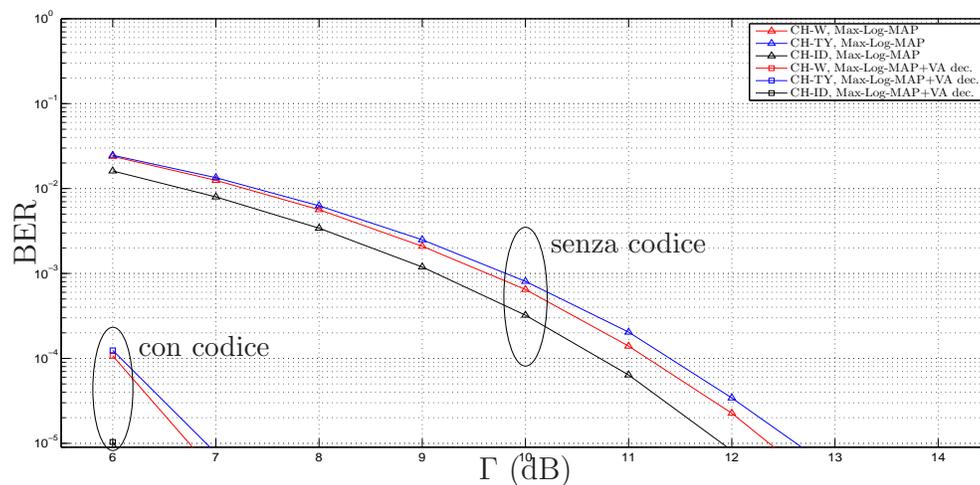


Figura 4.3: Canali stimati. Probabilità di errore in funzione di Γ con *hard detection* dei LLR ottenuti dal FBA Max-Log-MAP (senza codifica) e con *soft detection* tramite algoritmo di Viterbi (decodifica del codice convoluzionale).

4.2 Ricevitore con DFE

Anche in questo caso i risultati sono stati ottenuti da simulazioni su 16000 pacchetti. Ad ogni ricezione, dopo aver eseguito la stima del canale e della potenza del rumore bianco, sono stati eseguiti i calcoli del Paragrafo 3.2.5; come punto di partenza è stato imposto che il filtro *feedforward* abbia lo stesso ordine della risposta impulsiva equivalente del sistema, ovvero 6 tappi. Il secondo vincolo è stata la ricerca del minimo del funzionale visto nella (3.55) per stabilire il ritardo ottimo di decisione

$$D_{opt} = \arg \min_D |J_{min}| \quad (4.1)$$

Da questi presupposti deriva un ordine del filtro *feedback* pari a 6 e un ritardo $D = 4$.

Come si può notare in Figura 4.4 le prestazioni sono molto più uniformi al variare del tipo di canale trasmissivo rispetto a quelle viste per il FBA.

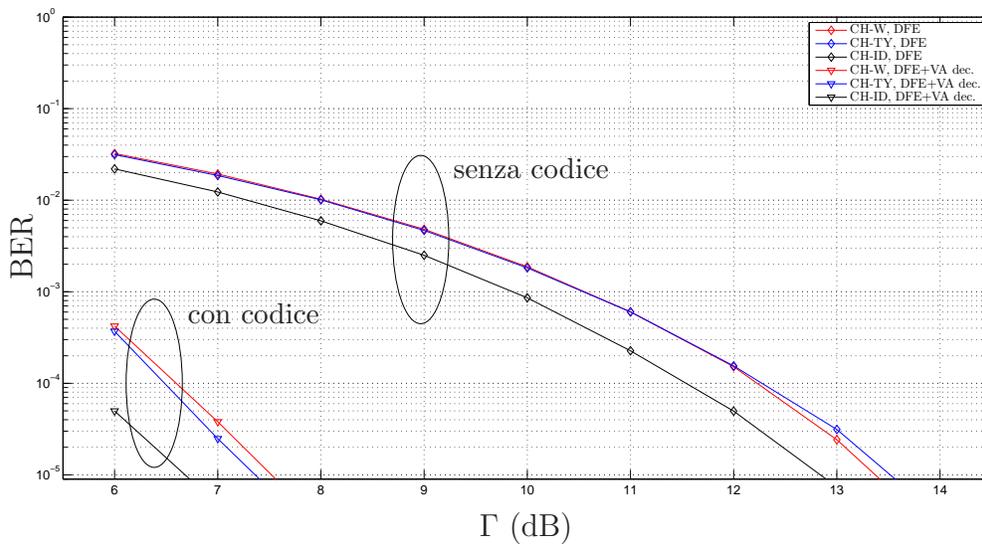


Figura 4.4: Canali stimati. Probabilità di errore in funzione di Γ del DFE con e senza codifica convoluzionale.

4.3 Confronto fra le prestazione del DFE e del FBA

Come riportato nelle Figure 4.5 e 4.6 il confronto fra presenza e assenza del codice evidenzia l'importanza della codifica e la superiorità prestazionale dell'algoritmo Max-Log-MAP rispetto al DFE.

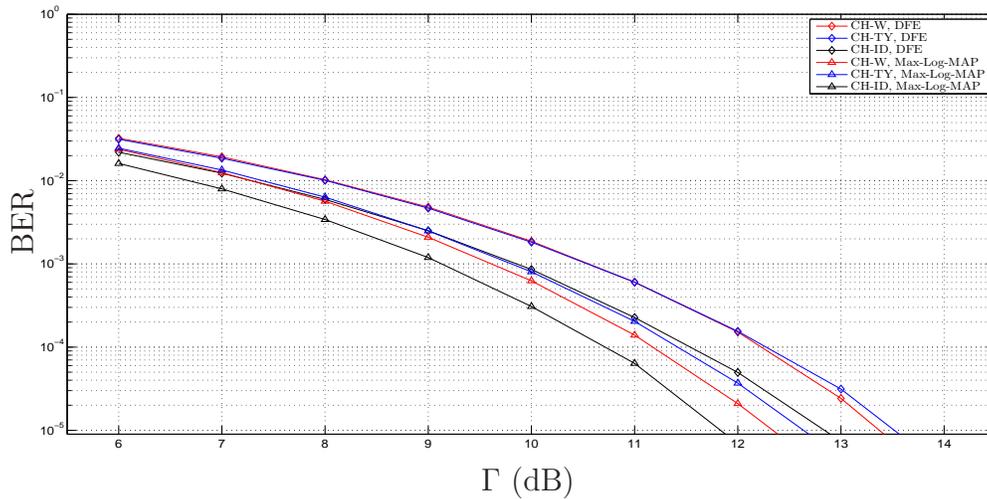


Figura 4.5: Canali stimati. Probabilità di errore in funzione di Γ con ricevitore DFE e FBA senza codifica convoluzionale.

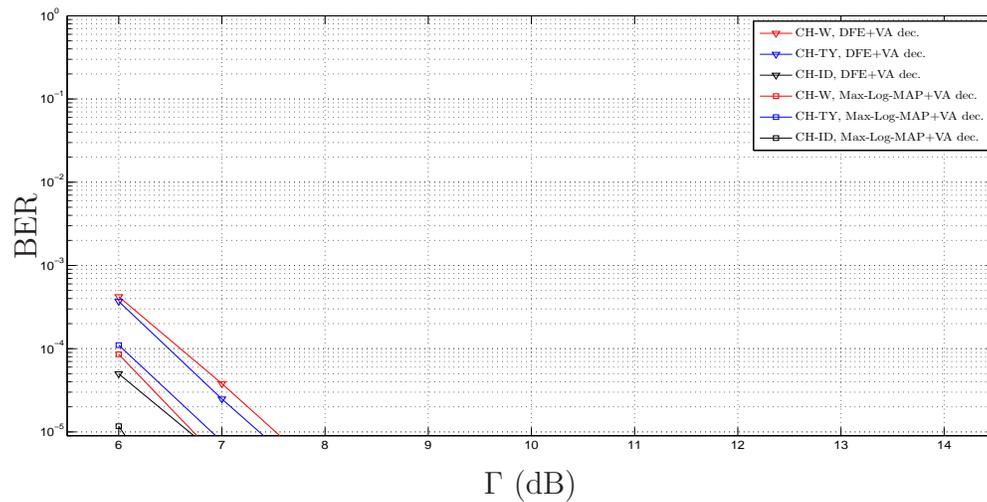


Figura 4.6: Canali stimati. Probabilità di errore in funzione di Γ con ricevitore DFE e FBA con codifica convoluzionale.

Capitolo 5

Canali tempo varianti

I risultati visti finora sono basati su realizzazioni statiche di due tipi di canale. Tuttavia, come visto nel Capitolo 2, i modelli del canale prevedono anche una tempo varianza. Passeremo quindi ora alla trattazione di una risposta impulsiva equivalente che cambia nel tempo. In un primo momento affronteremo il caso di un canale che cambia da pacchetto a pacchetto ma che rimane invariato durante una singola trasmissione; in seguito si vedrà anche il caso in cui la variazione avvenga entro la di trasmissione di un pacchetto.

5.1 Canale statico sul singolo pacchetto

Per questa sezione sono state considerate 5000 realizzazioni dei canali TY e WC visti nel Capitolo 2.

I tappi $g_{Ch,i}$ della risposta impulsiva sono modellati come variabili aleatorie date da:

$$g_{Ch,i} = c_i e^{j\varphi_i} + s_i \quad i = 0, \dots, 4 \quad (5.1)$$

con c_i costante reale, φ_i fase dell' i -esimo coefficiente uniformemente distribuita in $[-\pi, \pi]$ e s_i variabile aleatoria complessa di tipo gaussiano a media nulla e varianza $\sigma_{s_i}^2$.

Analizzando la (5.1) si nota che la statistica di $|g_{Ch,i}|$ è del tipo Rice [5] di parametro K_i con densità di probabilità data da:

$$f_{|g_{Ch,i}|}(a) = 2(1 + K_i) a e^{-K_i - (1+K_i)a^2} I_0[2a\sqrt{K_i(1 + k_i)}] 1_0(a) \quad (5.2)$$

dove I_0 è la funzione di Bessel:

$$I_0(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{x \cos \alpha} d\alpha. \quad (5.3)$$

Il parametro K_i viene definito *fattore di Rice* ed esprime il rapporto tra il quadrato della componente diretta e la potenza statistica della componente riflessa e/o diffusa, ovvero:

$$K_i = \frac{c_i^2}{\sigma_{s_i}^2}. \quad (5.4)$$

In Figura 5.1 riportiamo l'andamento di (5.2) per alcuni valori di K_i .

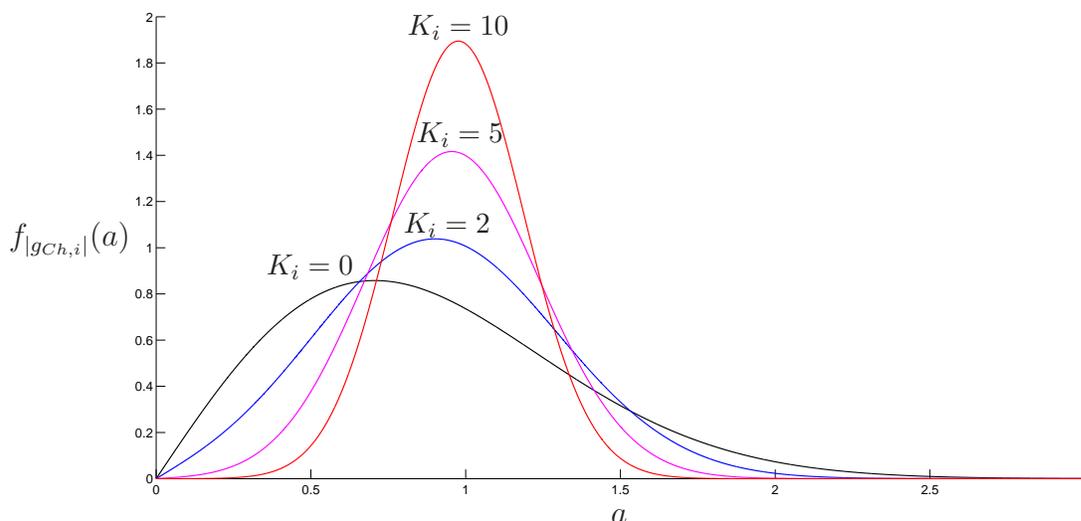


Figura 5.1: Densità di probabilità di Rice per alcuni valori di K_i .

Il parametro K_i viene posto pari a 9 dB per ogni tappo del canale TY e pari a 12 dB per il canale WC.

Dalla relazione (5.4) e da

$$M_{g_{Ch,i}} = c_i^2 + \sigma_{s_i}^2 \quad (5.5)$$

si ricava:

$$\begin{aligned} \sigma_{s_i}^2 &= \frac{M_{g_{Ch,i}}}{1 + K_i} \\ c_i &= \sqrt{K_i \sigma_{s_i}^2} \end{aligned} \quad (5.6)$$

e dunque, tenendo presente le specifiche date dalle Tabelle 2.1 e 2.2 e la normalizzazione sulla *potenza statistica del canale*, cioè che la somma delle potenze statistiche dei vari raggi, o tappi del canale, sia unitaria, si trovano i valori dei parametri 5.6 riportati in Tabella 5.1

	TY					WC				
	$g_{Ch,0}$	$g_{Ch,1}$	$g_{Ch,2}$	$g_{Ch,3}$	$g_{Ch,4}$	$g_{Ch,0}$	$g_{Ch,1}$	$g_{Ch,2}$	$g_{Ch,3}$	$g_{Ch,4}$
$\sigma_{s_i}^2$	0.0760	0.0191	0.0096	0.0048	0.0024	0.0153	0.0242	0.0077	0.0061	0.0061
c_i	0.7768	0.3893	0.2756	0.1951	0.1381	0.4922	0.6197	0.3485	0.3106	0.3106

Tabella 5.1: Valori dei parametri per la creazione dei tappi del canale come variabili aleatorie.

5.1.1 Max-Log-MAP

Per un Max-Log-MAP detector seguono nelle Figure 5.2 e 5.3, rispettivamente per il canale *typical case* e *worst case*, le distribuzioni delle probabilità d'errore per pacchetto in assenza di codice per diversi valori di Γ . Ricordiamo che sono state effettuate 5000 realizzazioni del canale.

Le corrispondenti prestazioni in presenza di codice sono invece riportate nelle Figure 5.4 e 5.5.

In generale si hanno delle prestazioni peggiori rispetto a quanto visto finora e questo può essere attribuito al fatto che nelle realizzazioni dei canali possono verificarsi casi particolarmente sfavorevoli e probabilmente le due realizzazioni dei capitoli precedenti erano un caso leggermente più performante della media. Comunque in presenza di codifica, è sufficiente un SNR medio di 10 dB affinché più del 90% dei canali abbia una trasmissione senza errori e questo vale per entrambi i modelli di canale.

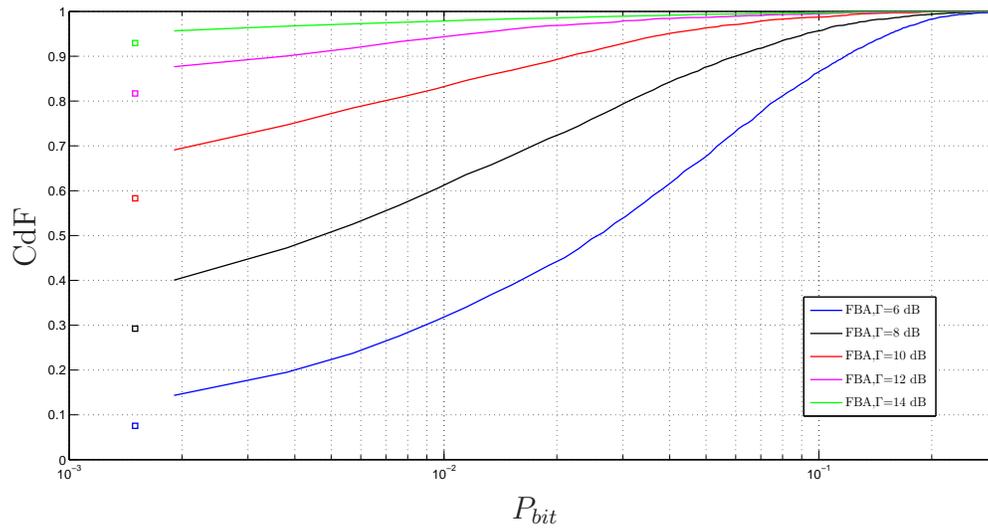


Figura 5.2: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* statico, una realizzazione per pacchetto.

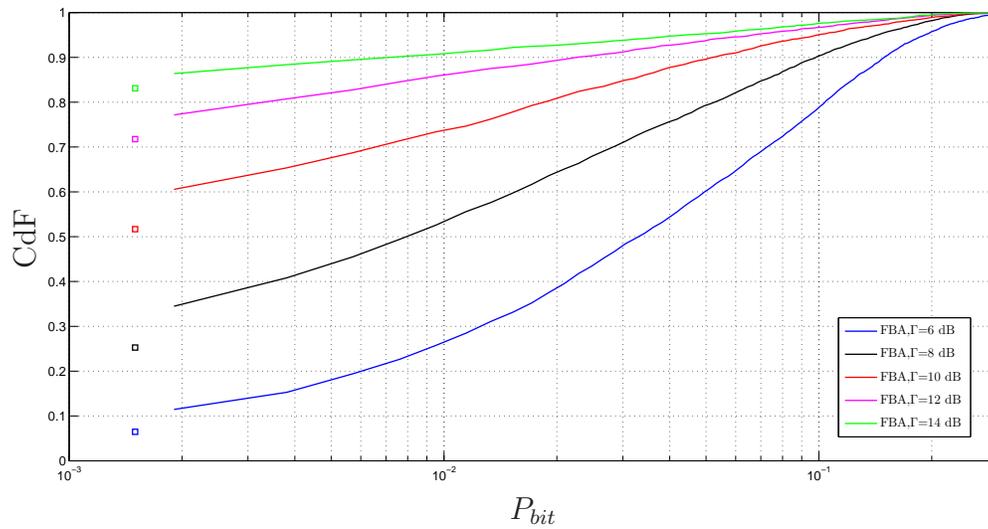


Figura 5.3: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* statico, una realizzazione per pacchetto.

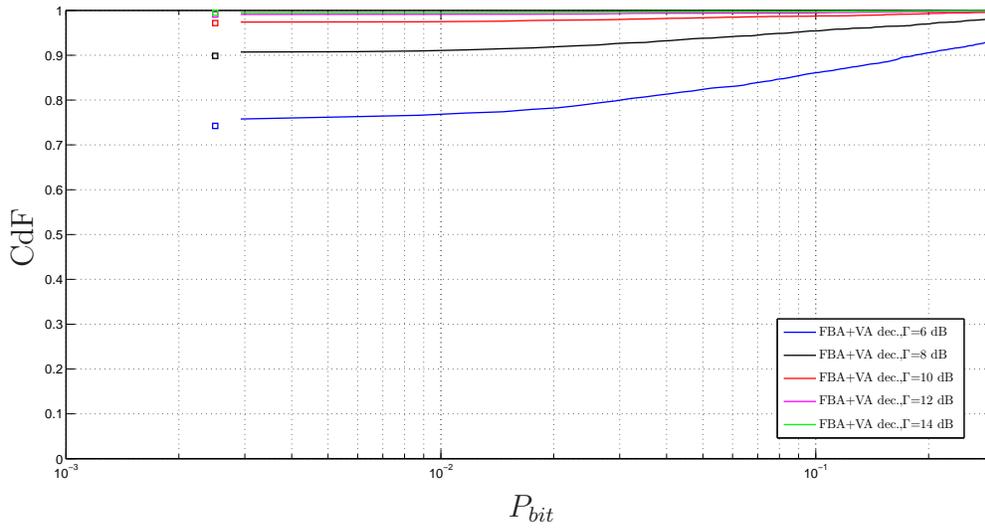


Figura 5.4: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* statico, una realizzazione per pacchetto.

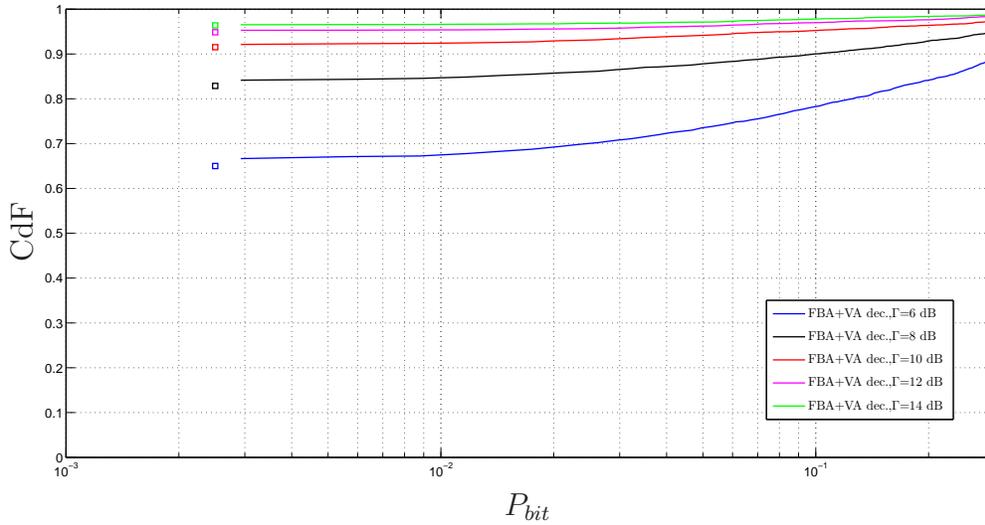


Figura 5.5: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* statico, una realizzazione per pacchetto.

5.1.2 DFE

Curve simili alle precedenti sono riportate per un DFE con e senza codice nelle Figure 5.6 - 5.9. I risultati sono simili a quelli ottenuti per un detector Max-Log-MAP anche se leggermente inferiori.

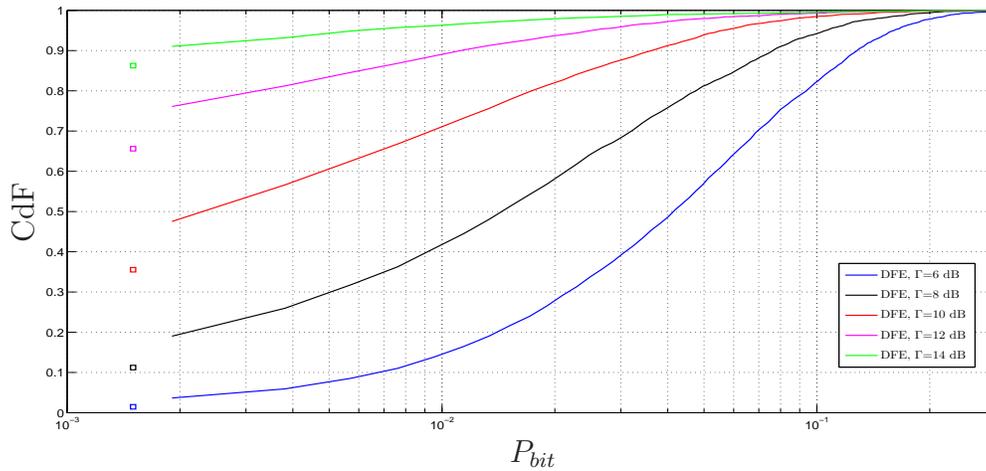


Figura 5.6: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* statico, una realizzazione per pacchetto.

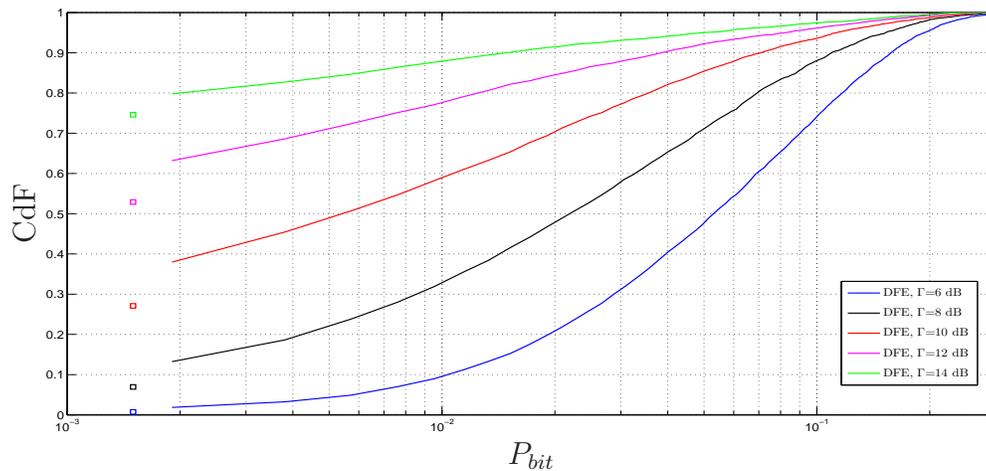


Figura 5.7: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* statico, una realizzazione per pacchetto.

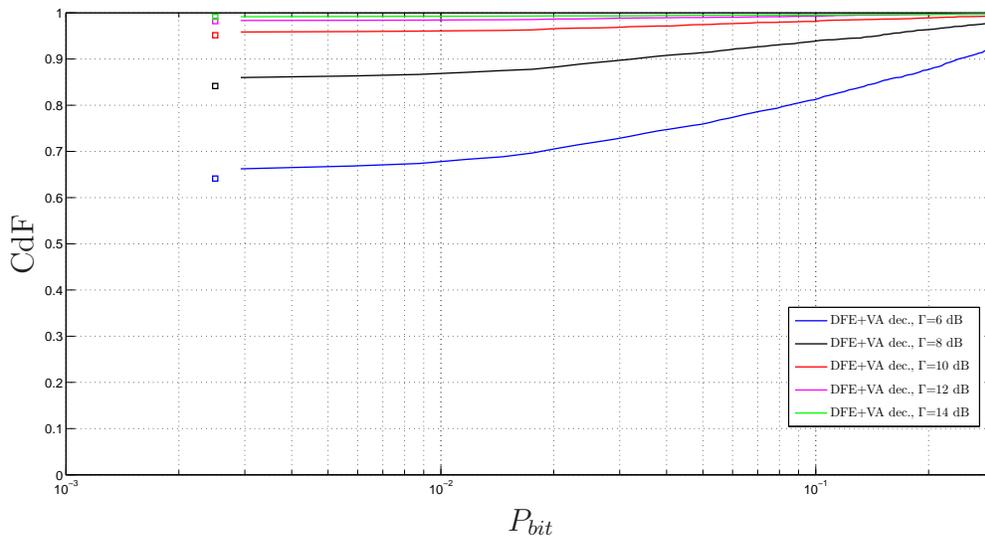


Figura 5.8: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* statico, una realizzazione per pacchetto.

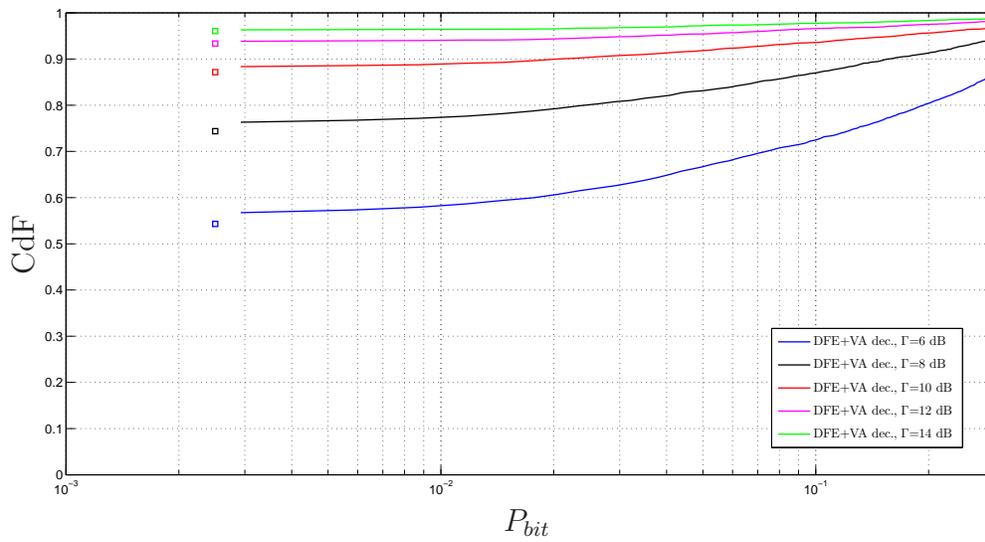


Figura 5.9: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* statico, una realizzazione per pacchetto.

5.1.3 Max-Log-MAP (media su 10 pacchetti)

Vediamo ora i risultati di simulazioni simili a quelle del Paragrafo 5.1.1 in cui sono trasmessi 10 pacchetti per ogni realizzazione del canale e si è poi mediato il risultato. Naturalmente i risultati sono leggermente peggiorati.

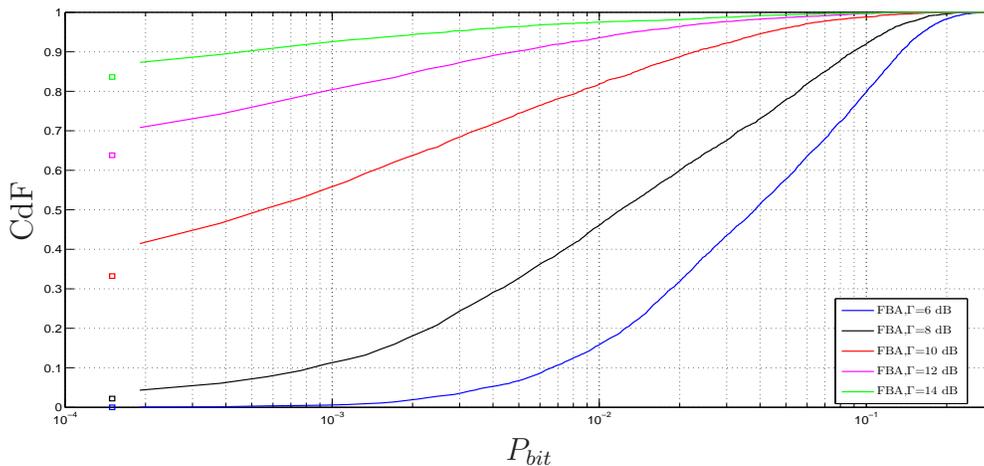


Figura 5.10: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *typical case* statico, una realizzazione per 10 pacchetti.

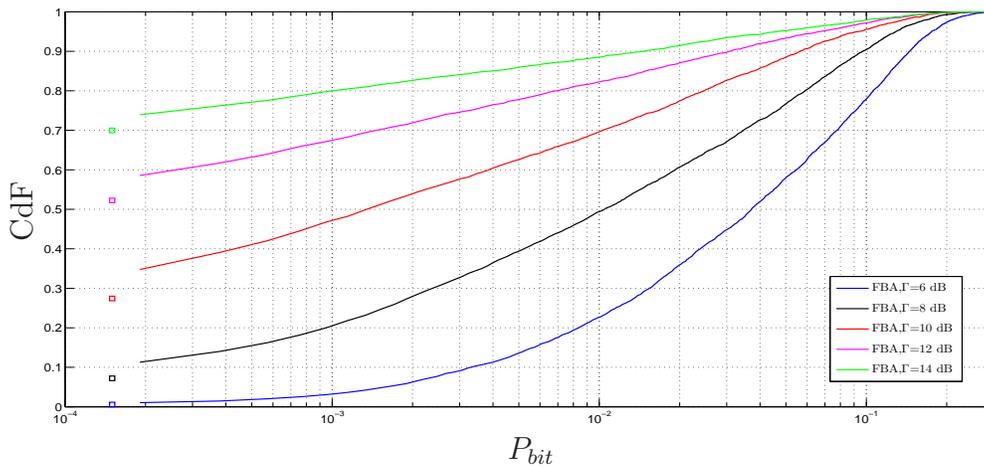


Figura 5.11: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *worst case* statico, una realizzazione per 10 pacchetti.

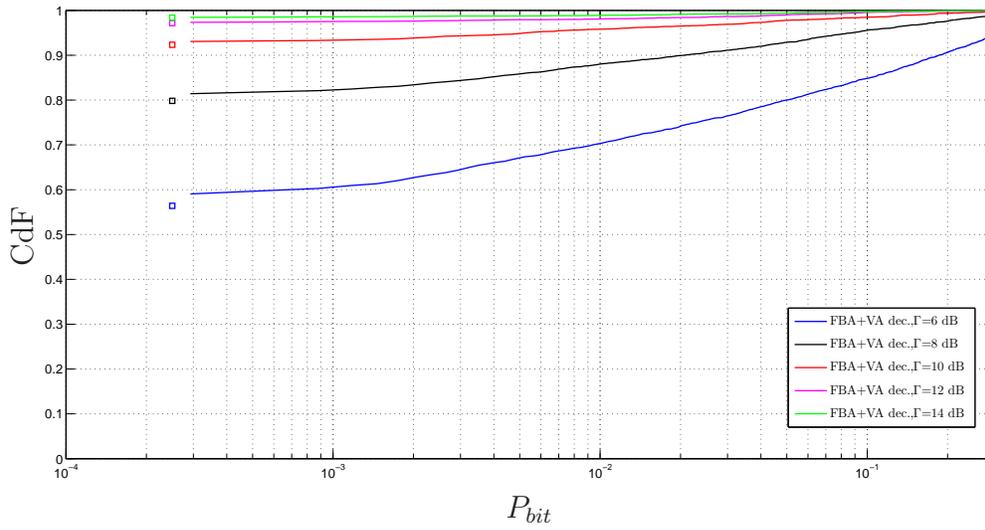


Figura 5.12: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *typical case* statico, una realizzazione per 10 pacchetti.

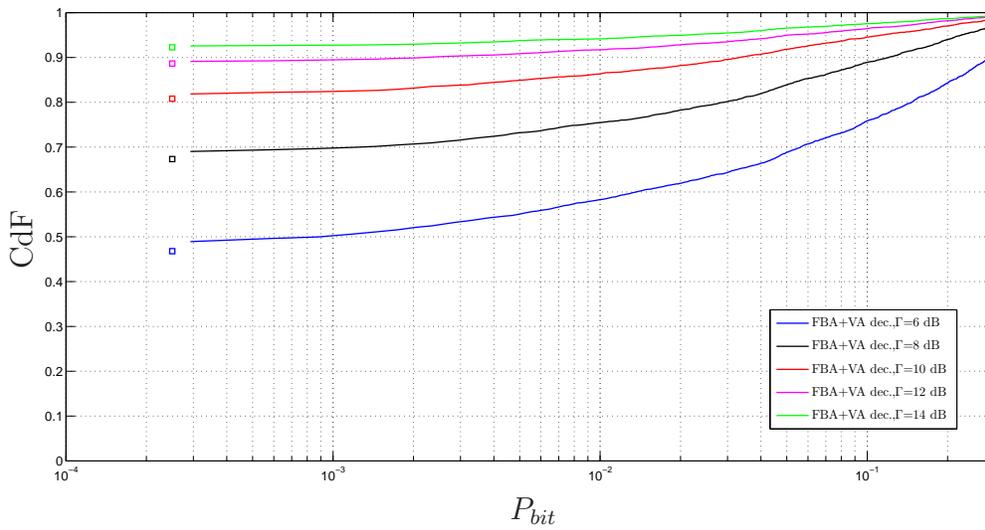


Figura 5.13: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *worst case* statico, una realizzazione per 10 pacchetti.

5.1.4 DFE (media su 10 pacchetti)

Vediamo ora i risultati di simulazioni simili a quelle del Paragrafo 5.1.2 in cui sono trasmessi 10 pacchetti per ogni realizzazione del canale e si è poi mediato il risultato.

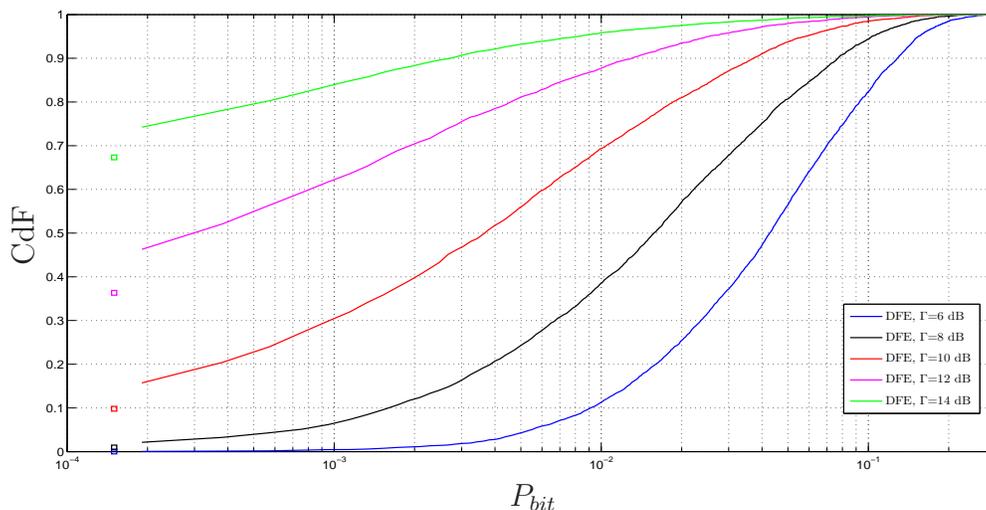


Figura 5.14: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *typical case* statico, una realizzazione per 10 pacchetti.

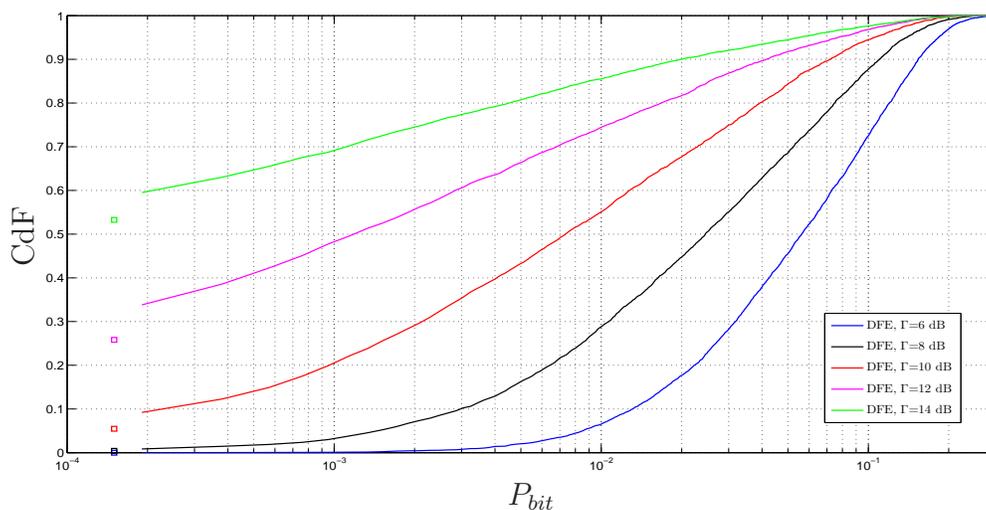


Figura 5.15: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *worst case* statico, una realizzazione per 10 pacchetti.

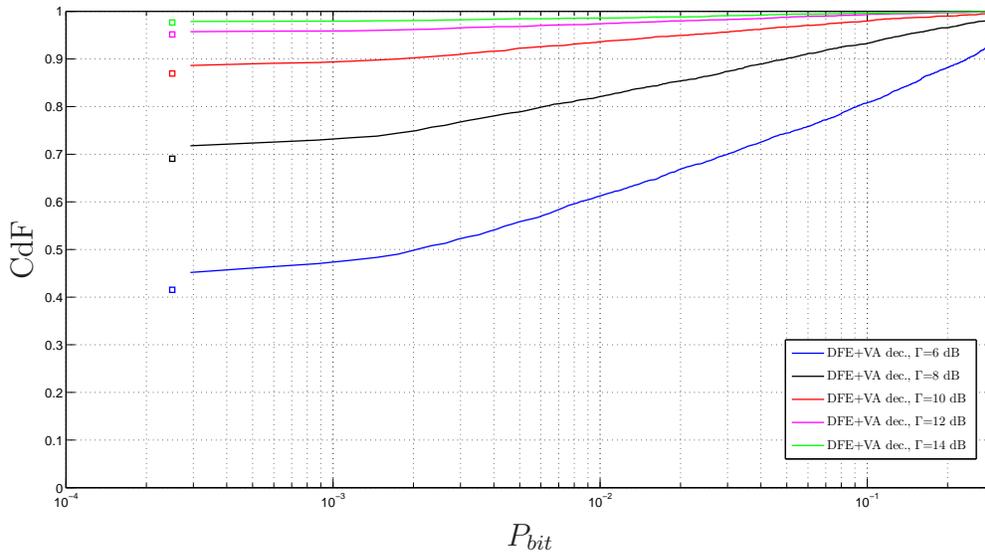


Figura 5.16: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *typical case* statico, una realizzazione per 10 pacchetti.

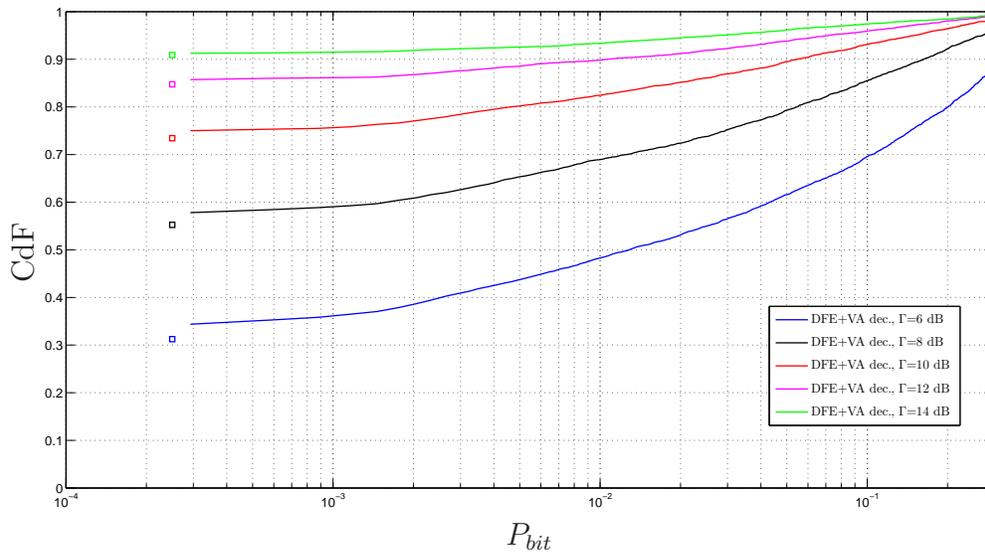


Figura 5.17: Funzione distribuzione delle percentuali di *bit* sbagliati al variare di Γ , canale *worst case* statico, una realizzazione per 10 pacchetti.

5.2 Canale tempo-variante

Passiamo ora alla trattazione dell'ultimo e piú complesso caso di tempo varianza del canale. In questa situazione i tassi variano come descritto in precedenza ma viene meno l'assunzione che il canale sia statico durante la trasmissione di un pacchetto. Inoltre si terrá in considerazione anche l'eventuale movimento relativo fra sorgente e ricevitore tramite lo spettro Doppler. A causa di questa tempo varianza dovremo introdurre una stima del canale non piú effettuata *una tantum* su un pacchetto ma che riesca a tenere sotto controllo le variazioni della risposta impulsiva equivalente anche durante la decodifica di un singolo pacchetto.

5.2.1 Effetto Doppler

Consideriamo il caso di un trasmettitore Tx e un ricevitore Rx visualizzato nella Figura 5.18 con segnale a banda stretta (monocromatico). Il ricevitore si muove dal punto P al punto Q con velocità v_p . La variazione di distanza fra trasmettitore e ricevitore è $\Delta\ell = v_p\Delta t \cos\theta$, dove v_p è la velocità del ricevitore rispetto al trasmettitore, Δt è il tempo impiegato dal ricevitore per andare da P a Q e θ è l'angolo di incidenza del segnale rispetto alla direzione del moto (si assume che θ sia uguale in P e Q). La variazione di fase del segnale ricevuto dovuta ai differenti cammini è data da:

$$\Delta\varphi = \frac{2\pi\Delta\ell}{\lambda} = \frac{2\pi v_p\Delta t}{\lambda} \cos\theta \quad (5.7)$$

e dunque l'effetto Doppler si manifesta come un apparente *shift* di frequenza del segnale ricevuto pari a

$$f_s = \frac{1}{2\pi} \frac{\Delta\varphi}{\Delta t} = \frac{v_p}{\lambda} \cos\theta \quad (5.8)$$

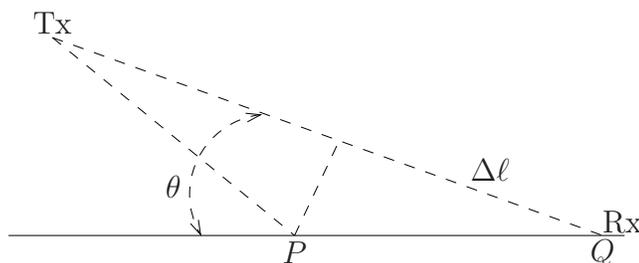


Figura 5.18: Illustrazione dell'effetto Doppler.

Dalla (5.8) si ricava per $\theta = 0$ la seguente relazione fra velocità del ricevitore e frequenza f_s :

$$f_s = 9.259 \cdot 10^{-4} v_p |_{km/h} f_0 |_{MHz} \text{ (Hz)} \quad (5.9)$$

dove $f_0 |_{MHz}$ è la frequenza della portante. Ad esempio, se $v_p = 100$ km/h e $f_0 = 900$ MHz si ha $f_s = 83$ Hz.

Consideriamo ora un ambiente *indoor*, ovvero la situazione che potrebbe crearsi all'interno di un edificio con muri di vario spessore e vari materiali fra trasmettitore e ricevitore. In questo caso il segnale ricevuto è composto da molti raggi che percorrono cammini ognuno di diversa lunghezza. Come visto nella (5.2.1), lo *shift* di frequenza f_s dipende dall'angolo θ . Dunque, a causa dei diversi cammini, il segnale ricevuto non è piú monocromatico ma presenta uno spettro attorno alla frequenza originale che viene detto *spettro Doppler*. Questo fenomeno si manifesta anche qualora trasmettitore e ricevitore siano fermi ma si muovano degli oggetti nello spazio che li separa.

Una caratteristica importante dello spettro Doppler è il *Doppler spread* che misura la dispersione in frequenza ottenuta trasmettendo un segnale sinusoidale; piú il canale varia nel tempo, piú grande sarà il *Doppler spread*. Questo si ripercuote sulle prestazioni che dovrà avere un eventuale algoritmo di equalizzazione adattativa al ricevitore: il suo tempo di convergenza dovrà essere molto minore dell'inverso del *Doppler spread* per seguire le variazioni del canale.

5.2.2 Modello a tempo continuo del canale

Assumendo che la propagazione del segnale avvenga attraverso un considerevole numero di percorsi e che dunque sia soggetta ad un grande numero di fenomeni aleatori, la rappresentazione (in banda base) della risposta impulsiva del canale puó essere rappresentata con buona approssimazione da un processo Gaussiano complesso tempo-variante $g(t, \tau)$. In particolare, $g(t, \tau)$ rappresenta l'uscita del canale all'istante t dovuta ad un impulso ideale applicato all'ingresso nell'istante $(t - \tau)$.

Valutiamo dunque l'autocorrelazione della risposta impulsiva valutata in due diversi istanti e due diversi ritardi

$$\mathbf{r}_g(t, t - \Delta t; \tau, \tau - \Delta \tau) = E[g(t, \tau)g^*(t - \Delta t, \tau - \Delta \tau)]. \quad (5.10)$$

In accordo con un modello di riflessione scorrelato e stazionario in senso lato (*wide-sense stationary uncorrelated scattering*, WSSUS), i valori di g per raggi che arrivano con ritardi diversi sono scorrelati e g è stazionario rispetto a t . Dunque si ha:

$$\mathbf{r}_g(t, t - \Delta t; \tau, \tau - \Delta \tau) = \mathbf{r}_g(\Delta t; \tau) \delta(\Delta \tau) \quad (5.11)$$

cioè l'autocorrelazione è non nulla solo per le risposte impulsive riferite allo stesso ritardo ed essendo g stazionario rispetto a t , essa dipende solo dalla differenza di tempi Δt fra le risposte impulsive valutate.

5.2.3 Spettro Doppler

Analizziamo ora il modello WSSUS del canale con riferimento alla tempovarianza e introduciamo la funzione correlazione della risposta in frequenza valutata negli istanti t e $t - \Delta t$ e, rispettivamente, alle frequenze f e $f - \Delta f$,

$$\mathbf{r}_G(t, t - \Delta t; f, f - \Delta f) = E[G(t, f)G^*(t - \Delta t, f - \Delta f)] \quad (5.12)$$

Sostituendo nella (5.12) la relazione

$$G(t, f) = \int_{-\infty}^{+\infty} g(t, \tau) e^{-j2\pi f\tau} d\tau \quad (5.13)$$

troviamo che \mathbf{r}_G dipende solo da Δt e Δf ; inoltre vale:

$$\mathbf{r}_G(\Delta t; \Delta f) = \int_{-\infty}^{+\infty} \mathbf{r}_g(\Delta t; \tau) e^{-j2\pi(\Delta f)\tau} d\tau \quad (5.14)$$

cioè che $\mathbf{r}_G(\Delta t; \Delta f)$ è la trasformata di Fourier di $\mathbf{r}_g(\Delta t; \tau)$. La trasformata di Fourier di \mathbf{r}_G è data da

$$\mathcal{P}_G(\lambda, \Delta f) = \int_{-\infty}^{+\infty} \mathbf{r}_G(\Delta t; \Delta f) e^{-j2\pi\lambda(\Delta t)} d(\Delta t) \quad (5.15)$$

La variazione temporale della risposta in frequenza è misurata da $\mathcal{P}_G(\lambda, 0)$.

Ora introduciamo lo spettro Doppler $\mathcal{D}(\lambda)$ che rappresenta la potenza dell'effetto Doppler per diversi valori della frequenza λ . $\mathcal{D}(\lambda)$ è la trasformata di Fourier dell'autocorrelazione della risposta impulsiva in corrispondenza dello stesso ritardo τ , valutata in due istanti diversi, ovvero ¹

$$\mathcal{D}(\lambda) = \int_{-\infty}^{+\infty} \frac{\mathbf{r}_g(\Delta t; \tau)}{\mathbf{r}_g(0; \tau)} e^{-j2\pi\lambda\Delta t} d(\Delta t) \quad (5.16)$$

Il termine $\mathbf{r}_g(0; \tau)$ nella (5.16) rappresenta un fattore di normalizzazione in modo che

$$\int_{-\infty}^{+\infty} \mathcal{D}(\lambda) d\lambda = 1 \quad (5.17)$$

¹In generale si potrebbe avere un spettro Doppler diverso per ogni percorso del canale.

Notiamo che la (5.16) implica che $\mathbf{r}_g(\Delta t; \tau)$ è una funzione separabile,

$$\mathbf{r}_g(\Delta t; \tau) = d(\Delta t) \cdot \mathbf{r}_g(0; \tau) = d(\Delta t)\mathbf{M}(\tau) \quad (5.18)$$

dove $d(\Delta t) = \mathcal{F}^{-1}[\mathcal{D}(\lambda)]$, con

$$d(0) = 1 \quad (5.19)$$

e $\mathbf{M}(\tau)$ è il *power delay profile* normalizzato definito come

$$\mathbf{M}(\tau) = E[|g(t, \tau)|^2] \quad \text{con} \quad \int_{-\infty}^{+\infty} \mathbf{M}(\tau) d\tau = 1 \quad (5.20)$$

Con questi presupposti vale:

$$\mathcal{D}(\lambda) = \mathcal{P}_G(\lambda; 0). \quad (5.21)$$

Quindi $\mathcal{D}(\lambda)$ può essere ottenuta come trasformata di Fourier di $\mathbf{r}_G(\Delta t; 0)$ che si può determinare trasmettendo un segnale sinusoidale ($\Delta f = 0$) e misurando l'autocorrelazione delle ampiezze del segnale ricevuto.

La massima frequenza f_d dello spettro Doppler viene detta *Doppler spread* o *frequenza Doppler*. L'inverso di questa misura è detto *tempo di coerenza* e dá una misura dell'intervallo di tempo nel quale il canale può essere assunto tempo-invariante o statico. Si dice in genere che il canale è *fast fading* se $f_d T > 10^{-2}$ e *slow fading* se $f_d T < 10^{-3}$.

Fra i vari tipi di spettro Doppler presenti in letteratura, prenderemo in considerazione lo spettro detto *classico* o *modello di Jake*. Denotata con f_d la frequenza Doppler, esso è definito come:

$$\mathcal{D}(f) = \begin{cases} \frac{1}{\pi f_d} \frac{1}{\sqrt{1-(f/f_d)^2}} & |f| \leq f_d \\ 0 & \text{altrimenti} \end{cases} \quad (5.22)$$

5.2.4 Realizzazione dei tappi

Dalle rappresentazioni appena viste a tempo continuo del canale, vogliamo ora passare ad un modello sfruttabile nella nostra simulazione a tempi T_Q . Come già visto in precedenza i ritardi saranno tutti multipli di questo quanto temporale e il canale sarà modellato come un filtro discreto a risposta impulsiva tempo variante. In particolare ogni tappo di questo filtro corrisponderà a $g_{Ch,i}$ e il processo per generarne l'andamento nel tempo è schematizzato in Figura 5.19. L'equazione (5.1) diventa:

$$g_{Ch,i}(kT_Q) = c_i e^{j(\varphi_i + 2\pi f_d k T_Q)} + s_i(kT_Q) \quad i = 0, \dots, 4 \quad (5.23)$$

dove kT_Q rappresenta l'istante di osservazione, c_i è una costante reale, mentre s_i varierà nel tempo nel modo che andremo ora ad esaminare. Inoltre $j2\pi f_d kT_Q$ è il *frequency shift* della componente diretta.

Per simulare ogni tappo si parte da un processo gaussiano bianco a media nulla e varianza unitaria, $w_i(lT_P)$, che viene filtrato attraverso un filtro a banda stretto h_{ds} sui tempi T_P per ottenere lo spettro Doppler desiderato ed infine interpolato per diventare il segnale $g'_i(kT_Q)$. Questo viene quindi moltiplicato per la deviazione standard σ_{s_i} per avere la giusta potenza statistica. Infine si aggiunge la componente $c_i e^{j(\varphi_i + 2\pi f_d kT_Q)}$ per ottenere $g_{Ch,i}(kT_Q)$.

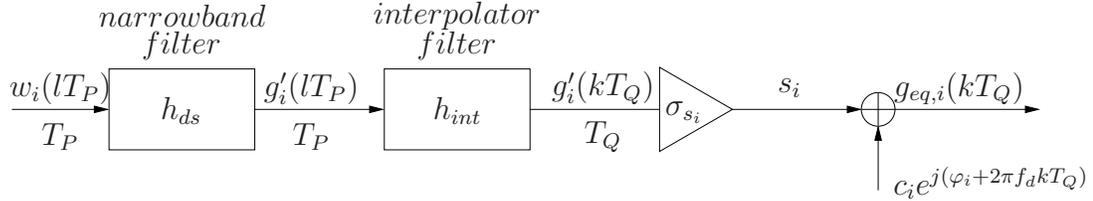


Figura 5.19: Modello per generare l' i -esimo tappo del canale.

Nella nostra realizzazione si considera una portante a 450 MHz e una velocità di 100 km/h; dunque la (5.8) porge:

$$f_d \cong 41 \text{ Hz} \quad (5.24)$$

da cui segue:

$$f_d T_P = 0.1 \Rightarrow T_P = 2439 \mu s \quad (5.25)$$

e dunque il fattore di interpolazione sarà

$$P = \frac{T_P}{T_Q} = \frac{2439}{0.416} \cong 5862 \quad (5.26)$$

Filtro h_{ds}

Per ottenere lo spettro Doppler classico visto in (5.22) con $f_d T_P = 0.1$ è stato usato un filtro IIR i cui parametri sono riportati in Tabella 5.2.

Filtro h_{int}

Per passare dalla realizzazione sui tempi T_P a alla realizzazione sui tempi T_Q viene usato un interpolatore quadratico con rapporto P .

L'equazione che fornisce i campioni fra $g'_i((l-1)T_P)$ e $g'_i(lT_P)$ è:

$$\begin{aligned} g'_i(\xi T_Q) &= \frac{n'}{2P} \left(\frac{n'}{P} - 1 \right) g'_i((l-1)T_P) + \left(1 - \frac{n'}{P} \right) \left(1 + \frac{n'}{P} \right) g'_i(lT_P) + \\ &+ \frac{n'}{2P} \left(\frac{n'}{P} + 1 \right) g'_i((l+1)T_P) \end{aligned} \quad (5.27)$$

$H_{ds}(z) = B(z)/A(z) \quad f_d T_P = 0.1$			
$\{\mathbf{a}_n\}, n = 0, \dots, 11 :$			
1.0000 e + 0	-4.4153 e + 0	8.6283 e + 0	-9.4592 e + 0
6.1051 e + 0	-1.3542 e + 0	-3.3622 e + 0	7.2390 e + 0
-7.9361 e + 0	5.1221 e + 0	-1.8401 e + 0	2.8706 e - 1
$\{\mathbf{b}_n\}, n = 0, \dots, 21 :$			
1.3651 e - 4	8.1905 e - 4	2.0476 e - 3	2.7302 e - 3
2.0476 e - 3	9.0939 e - 4	6.7852 e - 4	1.3550 e - 3
1.8067 e - 3	1.3550 e - 3	5.3726 e - 4	6.1818 e - 5
-7.1294 e - 5	-9.5058 e - 5	-7.1294 e - 5	-2.5505 e - 5
1.3321 e - 5	4.5186 e - 5	6.0248 e - 5	4.5186 e - 5
1.8074 e - 5	3.0124 e - 6		

Tabella 5.2: Coefficienti del filtro IIR h_{ds} utilizzato per generare lo spettro Doppler classico [6].

con $n' = 0, 1, \dots, P - 1$ e $\xi = (l - 1)P + n'$.

Riportiamo di seguito nelle Figure 5.20 e 5.21 il modulo dei vari tappi del canale, rispettivamente TY e WC. Le corrispondenti fasi sono riportate nelle Figure 5.22 e 5.23.

Si può notare come per ogni tappo la componente diretta sia prevalente su quella diffusa per cui la fase di ogni tappo varia abbastanza linearmente nel tempo.

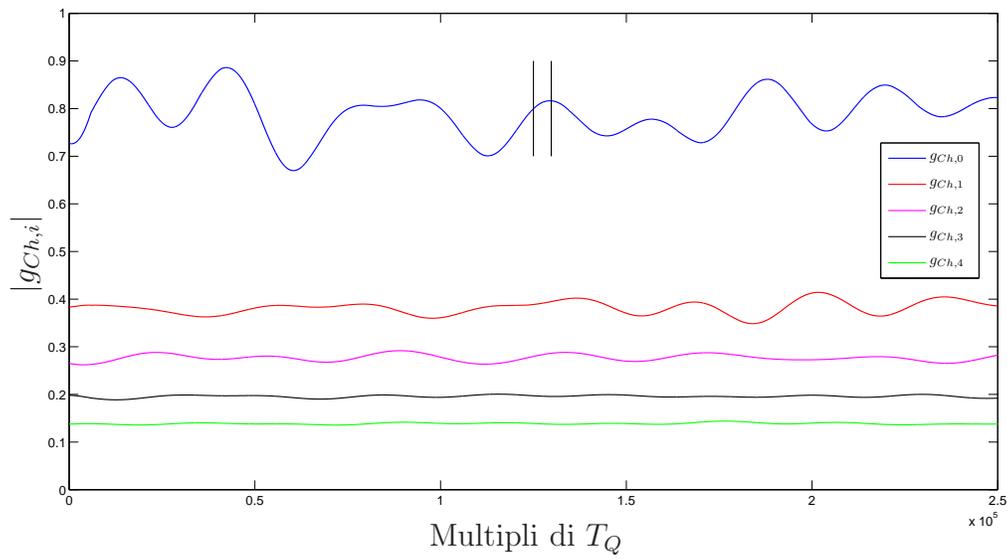


Figura 5.20: Modulo di una realizzazione dei tappi nel tempo, canale TY. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.

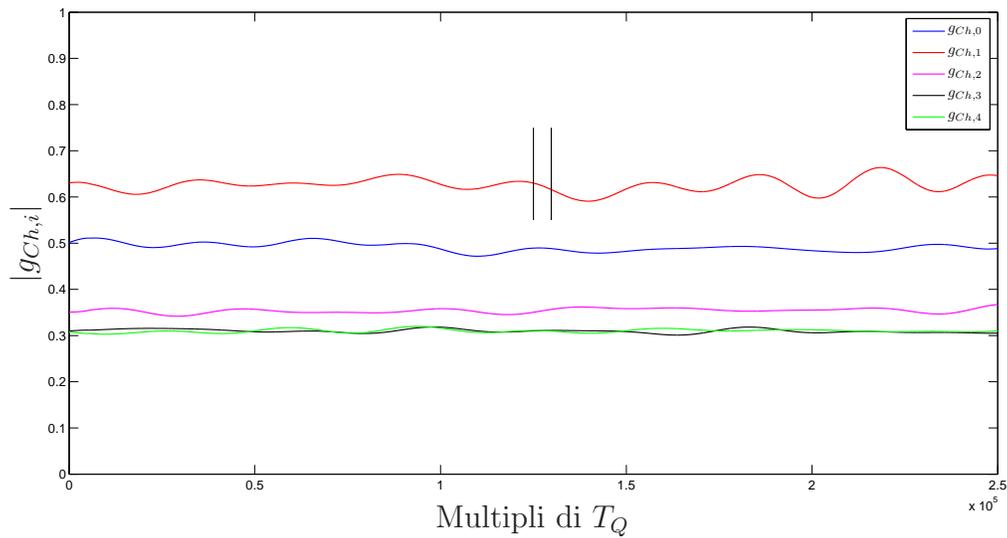


Figura 5.21: Modulo di una realizzazione dei tappi nel tempo, canale WC. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.

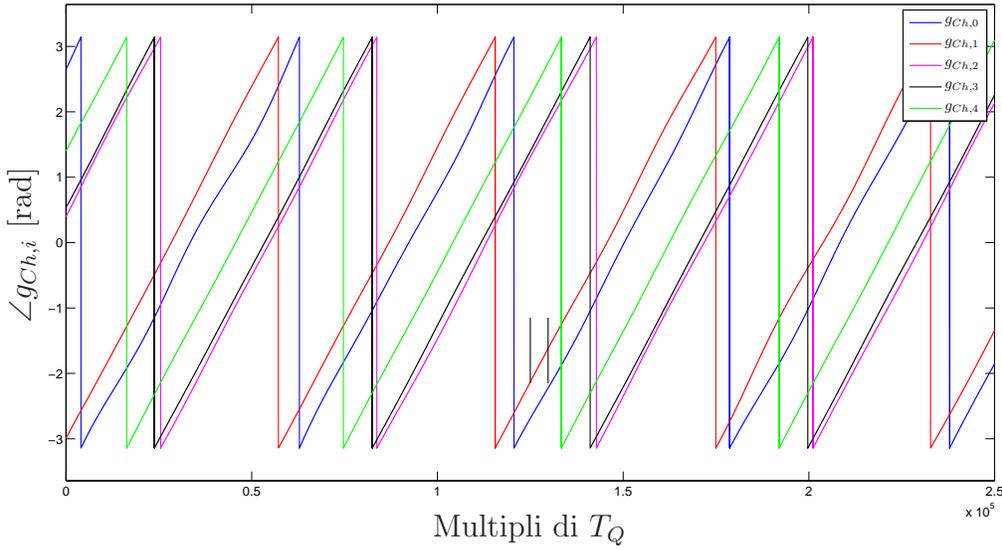


Figura 5.22: Fase di una realizzazione dei tappi nel tempo, canale TY. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.

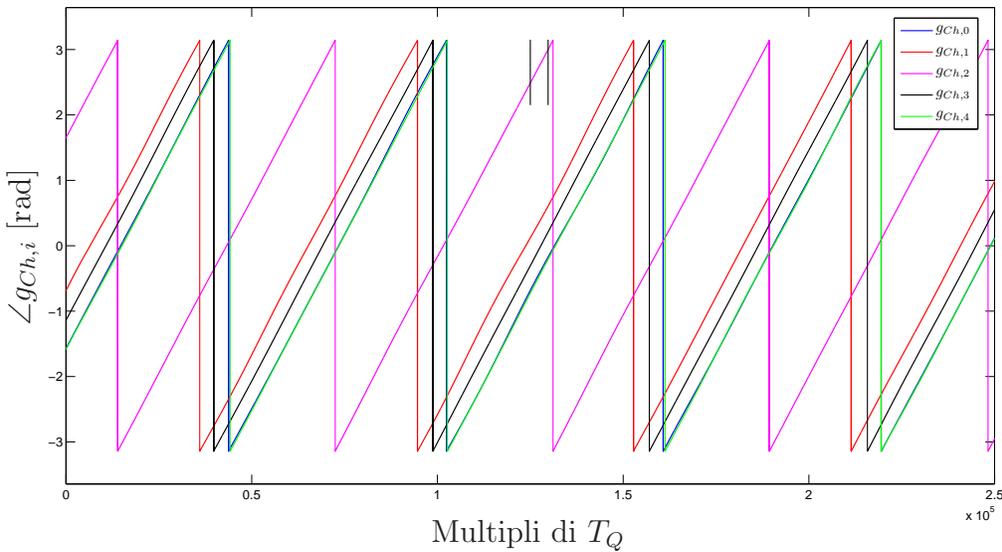


Figura 5.23: Fase di una realizzazione dei tappi nel tempo, canale WC. Le linee nere verticali indicano la durata della trasmissione di un pacchetto.

5.2.5 Max-Log-MAP, stima fissa

Nelle Figure 5.24 e 5.25 sono riportate le prestazioni dell'algoritmo Max-Log-MAP nel caso di canale Doppler e stima di canale effettuata *una tantum* all'inizio del pacchetto dati. Si nota che in tutti i casi le prestazioni peggiorano sensibilmente rispetto al caso di canali statici (v. Figure 5.2-5.5).

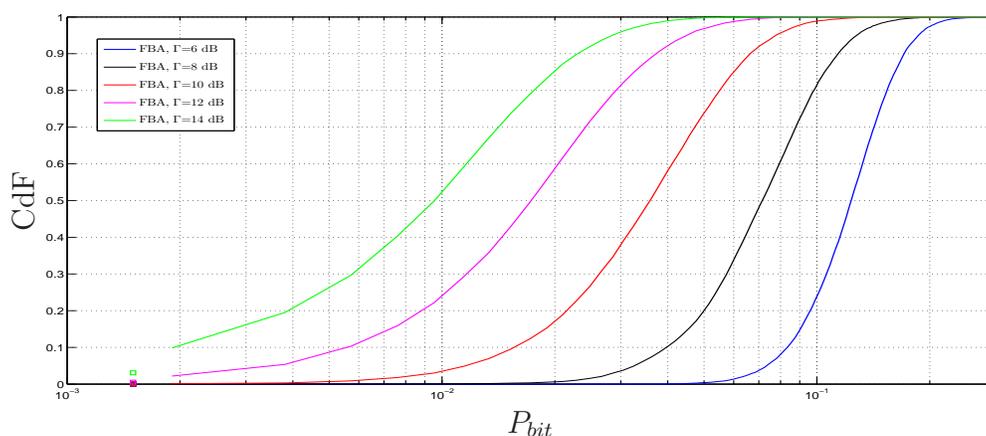


Figura 5.24: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima del canale all'inizio del pacchetto.

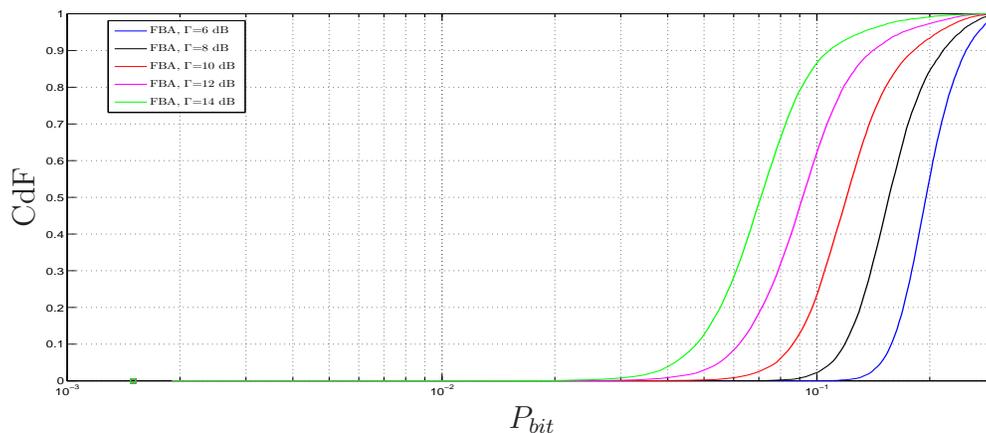


Figura 5.25: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima del canale all'inizio del pacchetto.

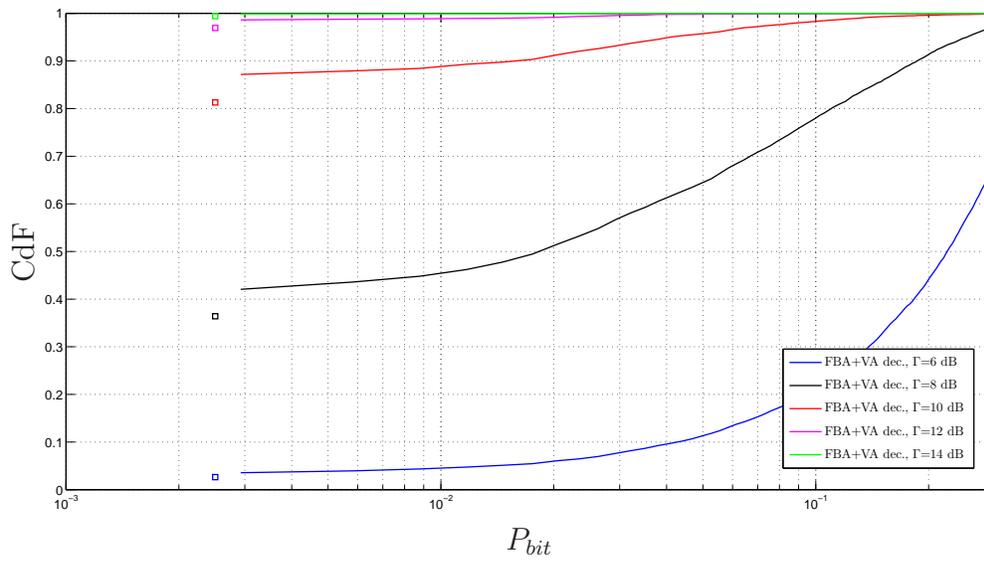


Figura 5.26: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima del canale all’inizio del pacchetto.

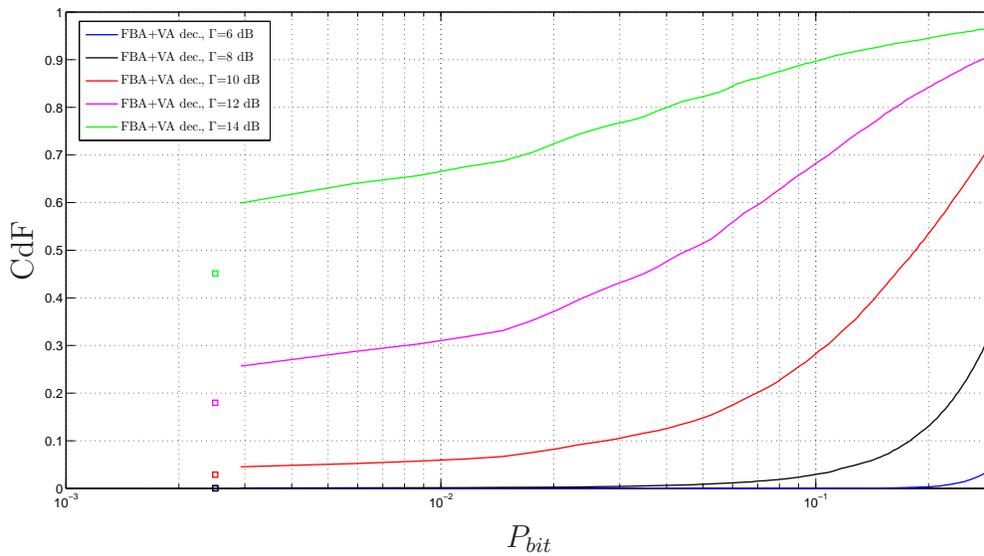


Figura 5.27: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima del canale all’inizio del pacchetto.

5.2.6 DFE, stima fissa

Nelle Figure 5.28-5.31 sono riportate le prestazioni dell'algoritmo DFE nel caso di canale Doppler e stima di canale effettuata *una tantum*. Anche qui si nota che in tutti i casi le prestazioni peggiorano sensibilmente.

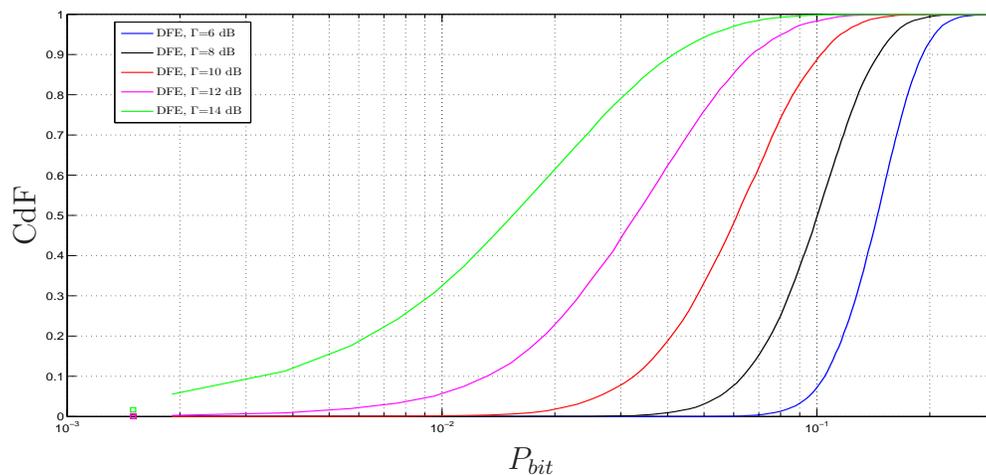


Figura 5.28: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima del canale all’inizio del pacchetto.

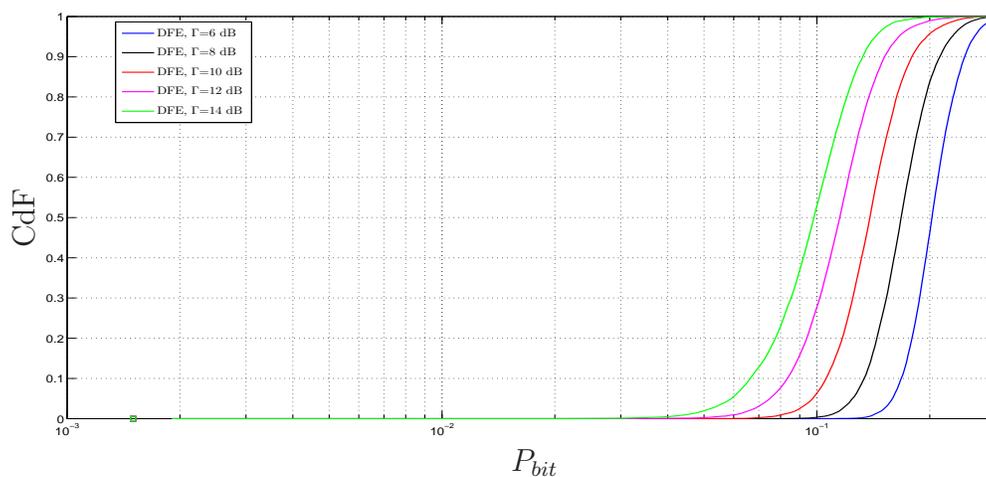


Figura 5.29: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima del canale all’inizio del pacchetto.

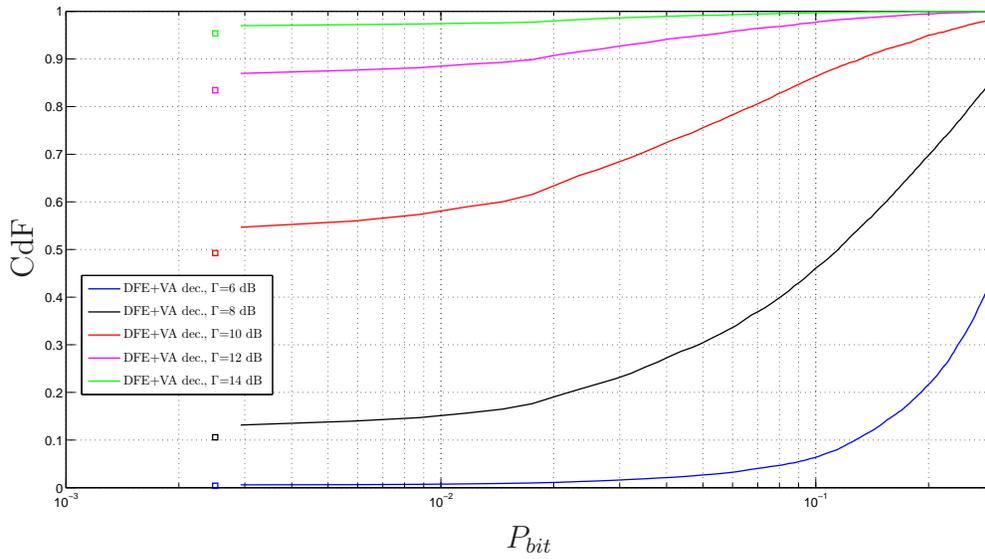


Figura 5.30: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima del canale all’inizio del pacchetto.

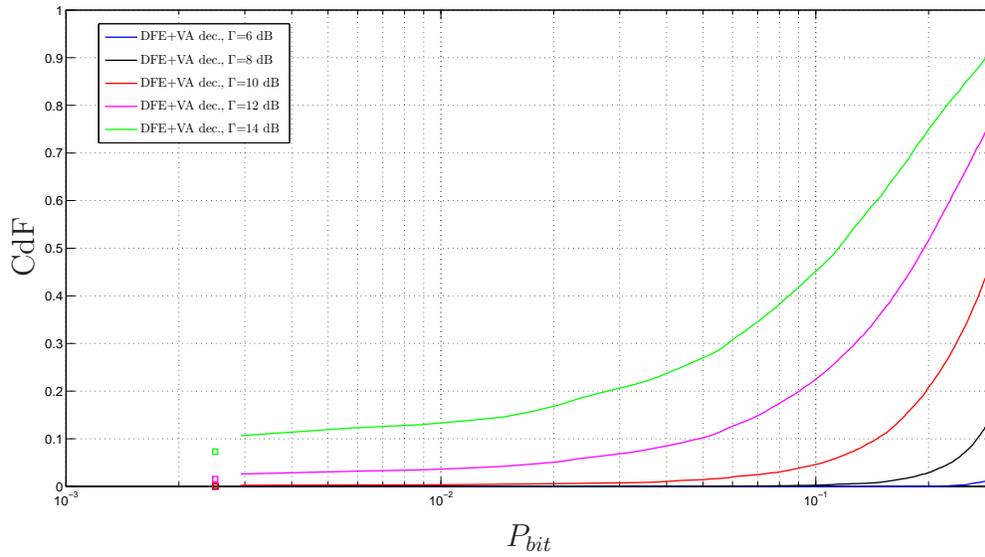


Figura 5.31: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima del canale all’inizio del pacchetto.

5.3 Ricevitore Max-Log-MAP con LMS

Nella sezione precedente si è visto come le prestazioni degradino quando la risposta impulsiva del canale varia significativamente nel tempo di trasmissione di un pacchetto. Per questo motivo non è piú sufficiente stimare la risposta impulsiva equivalente all'inizio della trasmissione del pacchetto ma questa stima deve essere in qualche modo aggiornata anche durante la trasmissione del *payload*.

5.3.1 LMS via RAM

La soluzione adottata è una variante dell'algoritmo *least mean-square* (LMS), la cui descrizione si trova in Appendice B. Questo algoritmo permette di stimare la risposta impulsiva del sistema e in particolare i suoi cambiamenti nel tempo. Grazie ad un sistema di retroazione basato sul segnale che sollecita il sistema, infatti, la stima può essere corretta durante il funzionamento; sarà quindi necessario avere a disposizione la sequenza dati inviata. Al ricevitore, ovviamente, è nota solo la sequenza PN iniziale e quindi, durante la trasmissione del *payload*, ci si baserà sulle rivelazioni del Max-Log-MAP per inseguire la risposta impulsiva.

In un LMS classico si ha un filtro che tenta di riprodurre il segnale uscente dal sistema da stimare e che corregge i suoi tapper basandosi sulla differenza fra le due uscite. Nel nostro caso, vengono stimate direttamente le uscite del sistema da stimare cioè i valori u_k per il Max-Log-MAP come visto in (3.19). Considerando che il canale ha risposta impulsiva finita e che i simboli di ingresso appartengono ad un alfabeto finito, si trova che c'è un numero finito di ingressi che possono sollecitare il sistema; in particolare, essendo l'alfabeto binario, avremo $2^{(L_1+L_2+1)}$ possibili ingressi; nel peggiore dei casi studiati si può arrivare ad avere $M = L_1 + L_2 + 1 = 7$.

Questo ci permette di mappare i valori u_k in una memoria i cui indirizzi sono dati dalla rappresentazione binaria del vettore d'ingresso

$$\tilde{\mathbf{c}}_k = [\tilde{c}_{k+L_1}, \dots, \tilde{c}_{k-L_2}] \quad (5.28)$$

in cui i primi $L_1 + L_2$ *bit* rappresentano lo stato j di arrivo e gli ultimi $L_1 + L_2$ rappresentano lo stato i di partenza della transizione. Per ogni istante k si valuta l'errore come

$$e_k = r_k - u(\hat{\mathbf{c}}_k) \quad (5.29)$$

e lo si accumula in una memoria temporanea $\Delta u(\hat{\mathbf{c}}_k)$ relativa all'indirizzo $\hat{\mathbf{c}}_k$. Oltre all'accumulo dell'errore si istanzia anche un contatore $\mu(\hat{\mathbf{c}}_k)$ che indica

il numero di accessi all'indirizzo. In questo modo, alla fine dell'aggiornamento di Δu , si potrà mediare il valore accumulato col numero di visite. Inoltre si effettuerà l'aggiornamento della metrica

$$u(\mathbf{c}) \leftarrow u(\mathbf{c}) + \Delta u(\mathbf{c}). \quad (5.30)$$

Per avere a disposizione del LMS la sequenza dati spedita, si esegue una rivelazione Max-Log-MAP su sottoslot (ad esempio lunghi 120 *bit*) e si effettua una *hard detection* dei primi 100 *bit* andando ad utilizzare la miglior sequenza di *survivor*. Gli ultimi 20 *bit* vengono trascurati essendo il transitorio terminale dell'algoritmo Max-Log-MAP e non conoscendo lo stato finale. Il sottoslot successivo partirà dall'istante 101 fino al 220. La rivelazione fornirà i *bit* da 101 a 200. Allora per ogni sottoslot rivelato viene creata una variazione dell'uscita $\Delta u(\hat{\mathbf{c}}_k)$ che verrà sommata, usando $u(\hat{\mathbf{c}}_k)$ per fornire l'uscita attuale. In Figura 5.32 si può vedere lo schema di questa soluzione.

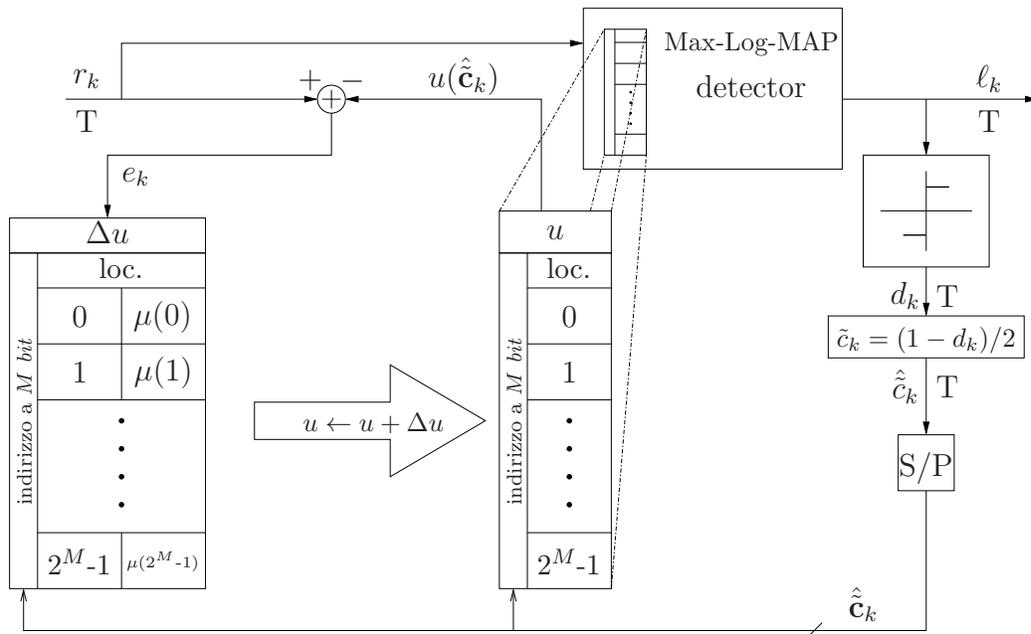


Figura 5.32: Schema del ricevitore con stima del canale tempo-variante tramite memoria temporanea $\Delta u(\mathbf{c})$.

5.3.2 Accorgimenti adottati

L'accuratezza della correzione di stima è evidentemente proporzionale al numero di visite che vengono fatte su ogni indirizzo. Dunque più si aspetta prima di aggiornare le metriche u , più si avrà una correzione dettagliata essendo la media di più valori. Bisogna considerare d'altra parte che, lasciando

passare troppo tempo, la mutazione del canale diventa deleteria per le prestazioni del Max-Log-MAP. La valutazione di questo *trade off* ha portato a fare le seguenti scelte:

- La matrice \mathbf{u}_k , che prima veniva calcolata tenendo conto della rotazione dovuta al fattore $j^{(k+1-i)}$ nella (3.19), viene ora calcolata una sola volta tramite il seguente cambiamento:

$$\begin{aligned} \rho_k &\rightarrow \dot{\rho}_k = \rho_k j^{-(k+1)} \\ u_k &\rightarrow \dot{u}_k = u_k j^{-(k+1)} = \sum_{i=-L_1}^{L_2} (\hat{g}_{eq,i} j^{-i}) d_{k-i} \end{aligned} \quad (5.31)$$

cosí facendo la rotazione viene spostata sul segnale ricevuto anziché sulla matrice. Questo quadruplica il numero di visite possibili ad un indirizzo poiché non esisteranno piú quattro diverse versioni della matrice in base alla rotazione.

- Il precedente accorgimento non basta ad avere un numero sufficiente di visite per indirizzo nel tempo in cui si vuole correggere la stima. Per questo si ignorano i *bit* meno significativi degli indirizzi che corrispondono ai tappi meno influenti della risposta impulsiva equivalente. In questo modo con un solo campione si aggiornano piú indirizzi che riguardano risposte impulsive molto affini. Cosí facendo la correzione apportata è piú efficace.
- Il periodo di aggiornamento della memoria u viene posto pari a $100T$. Questo valore è stato scelto considerando i risultati mostrati nelle Figure 5.33 - 5.36. Si vede infatti che aggiornando la memoria solamente due volte per pacchetto (ovvero ad intervalli di $200T$) le prestazioni peggiorano rispetto al caso a $100T$. I pacchetti verranno quindi divisi in sottosequenze da 100 *bit*. Tuttavia la rivelazione attraverso il Max-Log-Map verrà fatta su sottogruppi da 120 *bit* per dare modo alla metrica *backward* di stabilizzarsi. In questa configurazione infatti si può ancora sfruttare la conoscenza dello stato iniziale in trasmissione ma fermandosi prima della fine del pacchetto non si può fare riferimento allo stato finale noto. Usando una sovrapposizione di 20 *bit* e facendo partire la metrica *backward* da stati finali equiprobabili, si possono ritenere stabilizzate le metriche entro i primi 100 campioni e soltanto di queste si considerano i LLR. Solo nell'ultima sottosequenza si parte da uno stato finale noto, trattandosi della fine del pacchetto. Le metriche *forward* invece partono dallo stato imposto in trasmissione

e all'inizio di ogni sottosequenza ripartono dalla corrispondente metrica della sottosequenza precedente.

Nelle Figure 5.33 - 5.36 è riportato l'andamento dell'errore e_k con e senza correzione di stima durante il pacchetto. Si notano, in presenza di correzione di stima, degli abbattimenti dell'errore in concomitanza con gli istanti in cui viene aggiornata la metrica u .

Facciamo notare che nei primi 63 istanti l'errore $|e_k|$ è al di sotto del suo normale andamento perché questa prima parte è relativa alla sequenza di *training* su cui viene eseguita la stima.

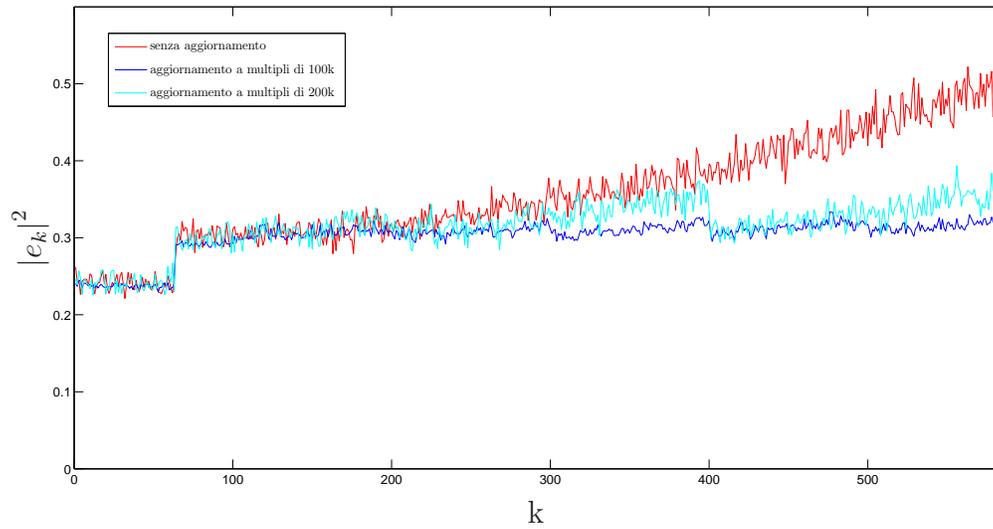


Figura 5.33: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima (blu) usando 3 *bit* per indirizzare la memoria. Canale *typical case*, $\Gamma = 8$ dB.

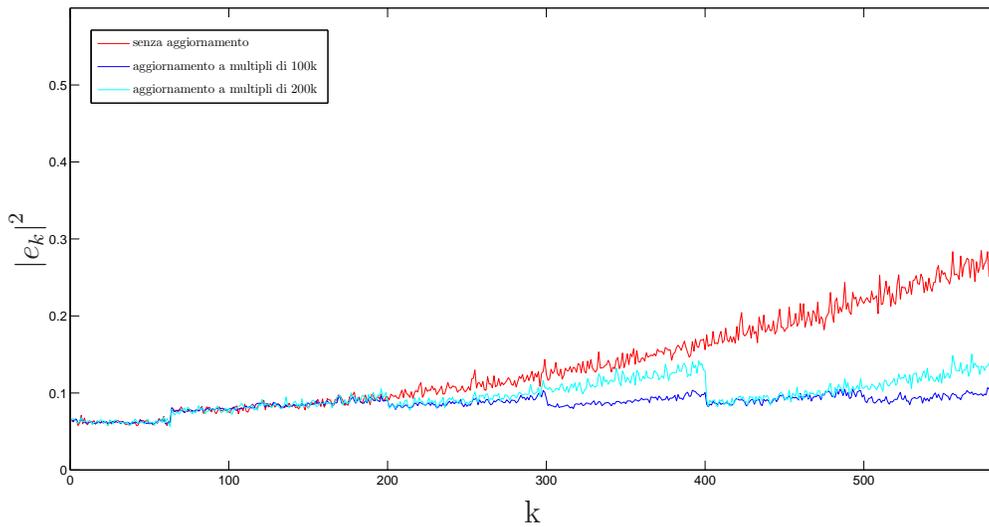


Figura 5.34: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima (blu) usando 3 *bit* per indirizzare la memoria. Canale *typical case*, $\Gamma = 14$ dB.

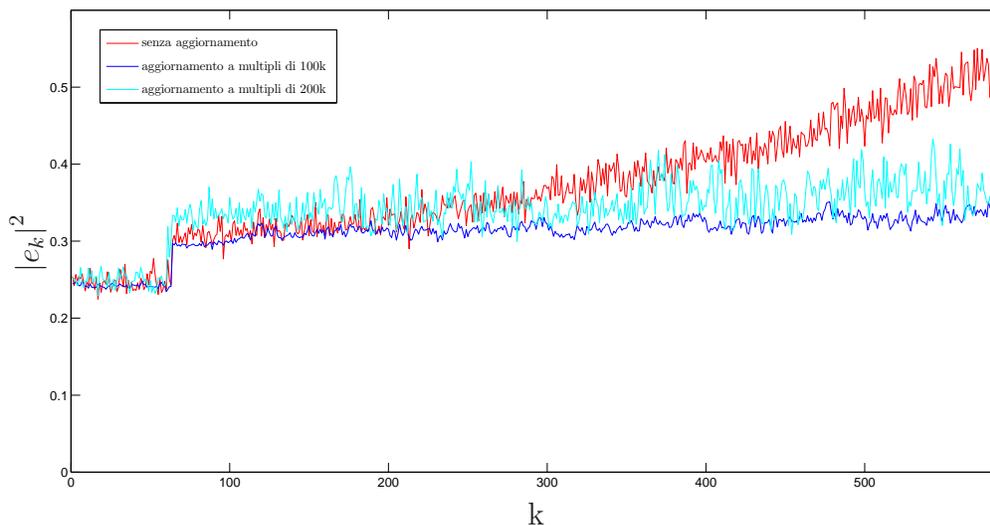


Figura 5.35: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima (blu e azzurro) usando 3 *bit* per indirizzare la memoria. Canale *worst case*, $\Gamma = 8$ dB.

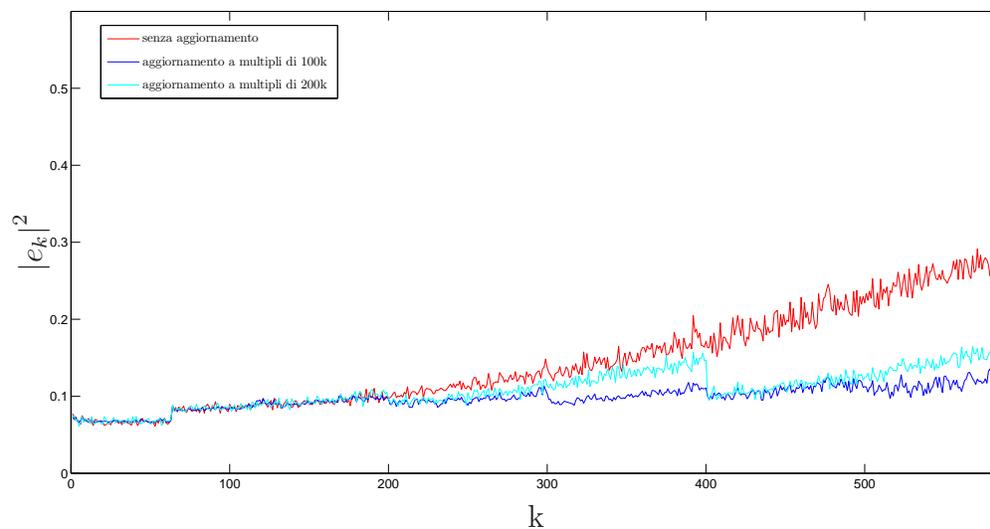


Figura 5.36: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima (blu e azzurro) usando 3 *bit* per indirizzare la memoria. Canale *worst case*, $\Gamma = 14$ dB.

5.3.3 Prestazioni del Max-Log-MAP con LMS via RAM

Vediamo dunque le prestazioni del Max-Log-MAP con l'algoritmo LMS via RAM a 3 *bit*. In ognuno dei quattro casi si ha un miglioramento rispetto a quanto visto nelle Figure 5.24 - 5.27. Di fatto le prestazioni si avvicinano di molto a quelle delle Figure 5.2-5.5 per un canale tempo-invariante all'interno di ogni slot.

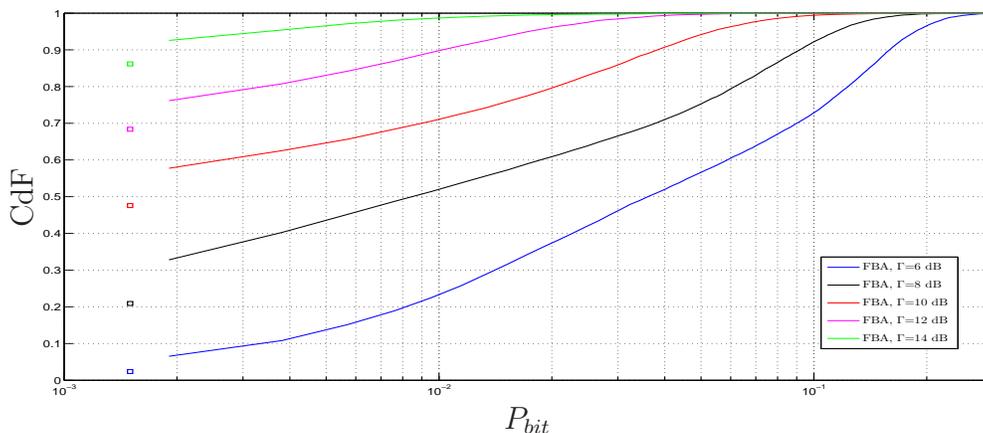


Figura 5.37: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

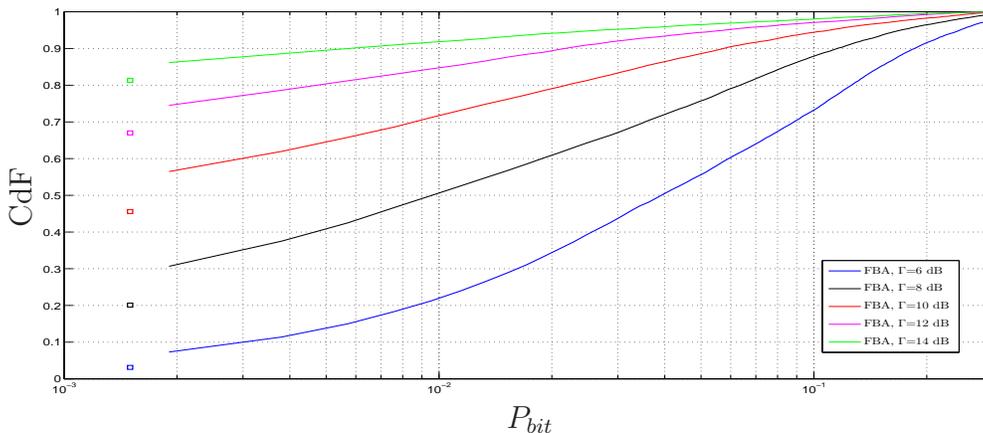


Figura 5.38: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

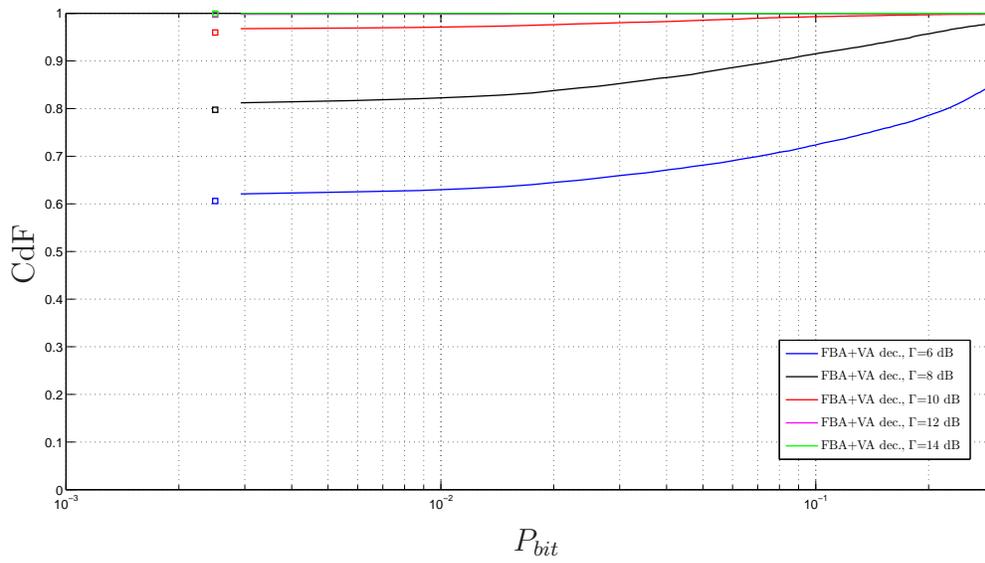


Figura 5.39: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

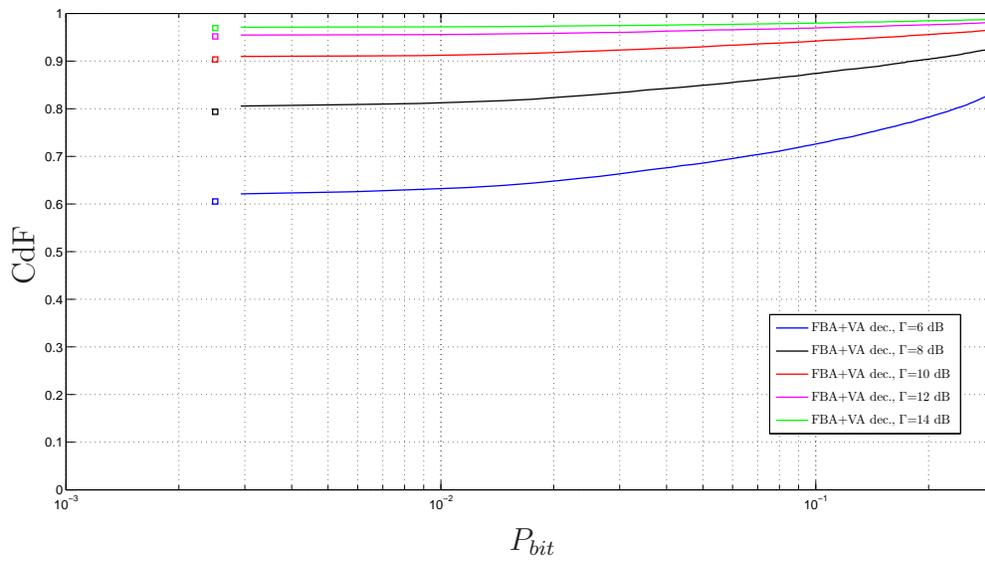


Figura 5.40: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

5.3.4 Packet error rate

Al fine di valutare la percentuale di pacchetti *error free* con target di 10^{-3} sono state effettuate delle simulazioni di trasmissione su 50000 pacchetti. Nelle Figure 5.41 e 5.42 vediamo i risultati.

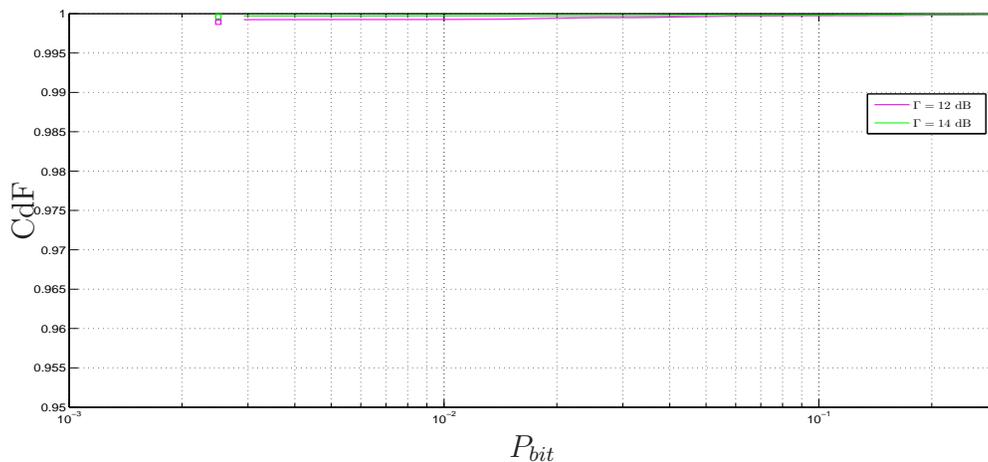


Figura 5.41: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

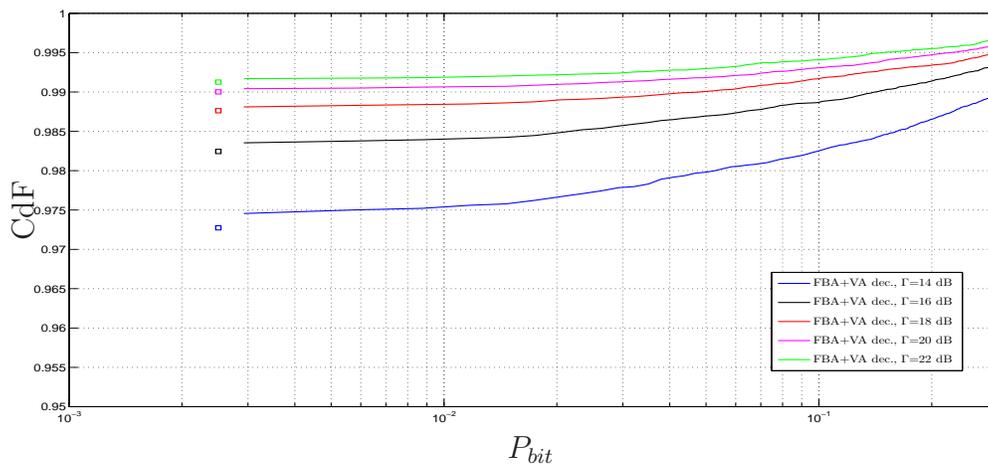


Figura 5.42: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima di canale via RAM ogni 100 *bit*.

Appendice A

Prestazioni del metodo di correlazione

La stima di canale eseguita col metodo di correlazione vista nel Paragrafo 3.1.2 risente dell'errore di approssimazione del modello di Laurent. Per avere un riscontro teorico è stata fatta una prova di stima su un sistema che impiega effettivamente lo schema equivalente.

Come visto per il caso di modulazione GMSK si valuta il rapporto Λ_n ottenuto dalle stime e lo si confronta con quello teorico ottenuto dalla (3.11). Il risultato viene mostrato in Figura A.1 per un modello trasmissivo in cui il rumore bianco viene aggiunto sui tempi T . In Figura A.2, invece, si mostra il risultato ottenuto da un modello in cui il rumore bianco viene creato sui tempi T_Q e filtrato col filtro di Figura 2.4. Si nota che nel primo caso, essendo il rumore bianco, la previsione teorica viene rispettata. Nel secondo caso, invece, il rumore filtrato non ha più una PSD perfettamente bianca a causa di un fenomeno di aliasing dovuto al campionamento e dunque a bassi valori di Γ i risultati si scostano dalle previsioni teoriche.

Sempre con un sistema basato sullo schema equivalente di Laurent è stata eseguita una stima della potenza del rumore bianco. Il confronto fra valore effettivo e stimato viene riportato in Figura A.3 per tre tipi di canale.

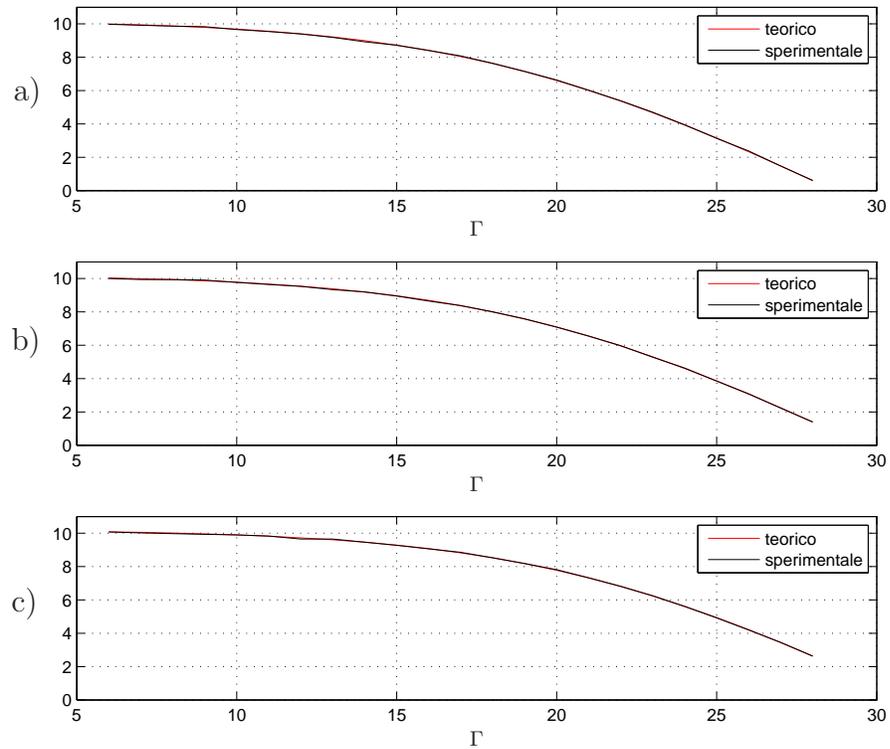


Figura A.1: Confronto fra andamento teorico e reale di Λ_n con rumore bianco aggiunto a tempi T per canale *ideale* (a), *typical case* (b) e *worst case* (c). Valori in dB.

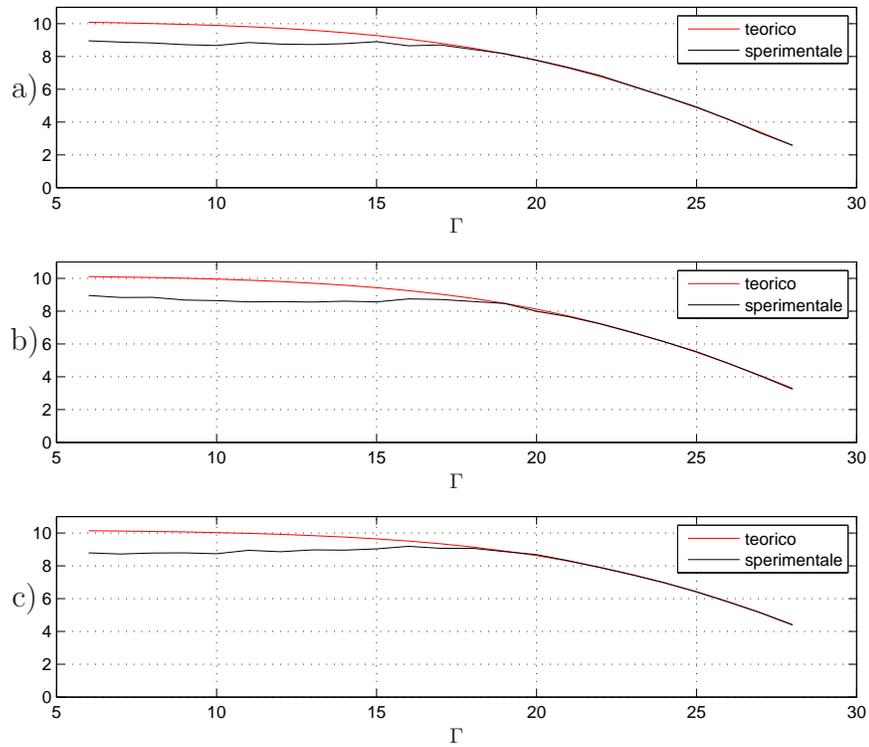


Figura A.2: Confronto fra andamento teorico e reale di Λ_n con rumore aggiunto a tempi T_Q per canale *ideale* (a), *typical case* (b) e *worst case* (c). Valori in dB.

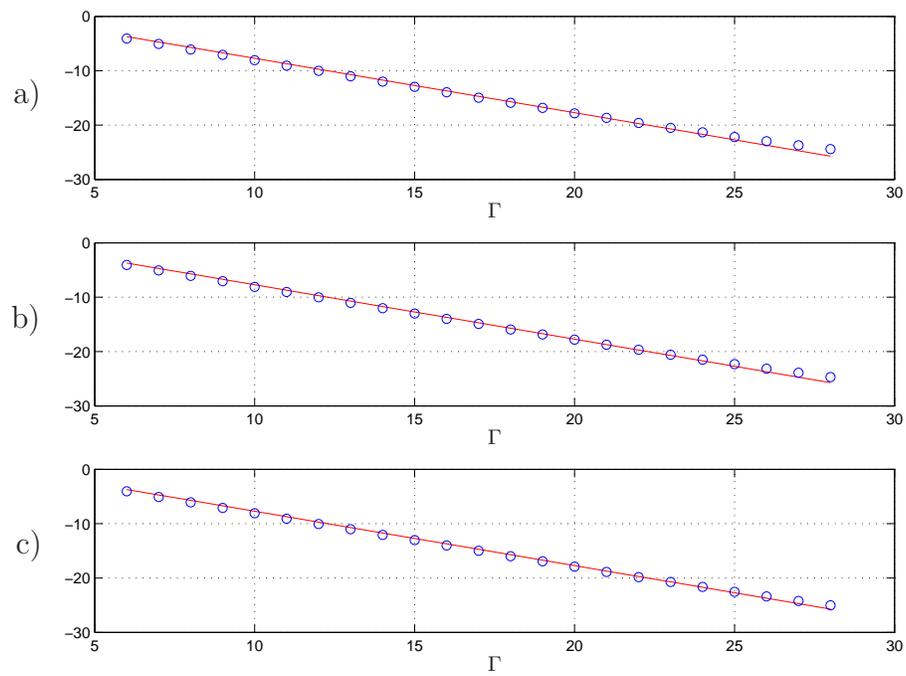


Figura A.3: Confronto fra $\sigma_{\omega_C}^2$ (in rosso) e la sua stima (in blu) per canale ideale (a), *typical case* (b) e *worst case* (c). Valori in dB.

Appendice B

L'algoritmo LMS

L'algoritmo *least mean-square* (LMS) è un algoritmo a bassa complessità computazionale che fornisce una approssimazione della soluzione ottima del filtro di Wiener-Hopf senza la conoscenza delle matrici di autocorrelazione del segnale ricevuto e del segnale che viene usato per sollecitare tale sistema. L'LMS si basa su un filtro tempo variante i cui coefficienti q_i sono regolati dalla retroazione illustrata in Figura B.1. Nel nostro caso la sequenza che sollecita il sistema è quella rivelata dal FBA, il segnale ricevuto è dato da r_k e il filtro \mathbf{q} ha ordine pari a $M = L_1 + L_2 + 1 = 7$.

Le equazioni che descrivono il sistema sono:

1. *Uscita del filtro*

$$y_k = \mathbf{r}_k^T \mathbf{q}_k \quad (\text{B.1})$$

essendo \mathbf{r}_k il vettore degli ultimi M valori ricevuti $[r_k, r_{k-1}, \dots, r_{k-M+1}]^T$.

2. *Errore di stima*

$$e_k = d_k - y_k. \quad (\text{B.2})$$

3. *Aggiornamento dei coefficienti*

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mu e_k \mathbf{r}_k^* \quad (\text{B.3})$$

dove l'apice * indica il complesso coniugato.

4. *Stato iniziale*

Come stato iniziale si parte dalla stima della risposta impulsiva equivalente fatta sul preambolo col metodo a correlazione.

$$\mathbf{q}_0 = \hat{\mathbf{g}}_{eq}. \quad (\text{B.4})$$

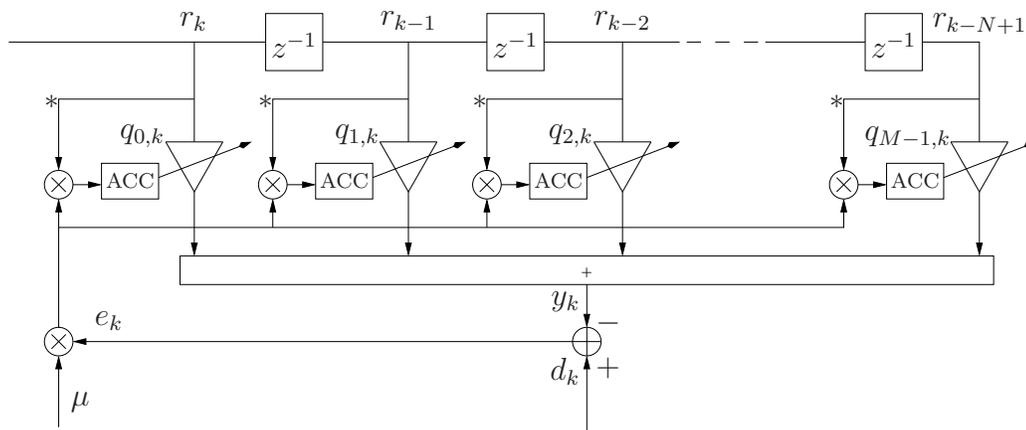


Figura B.1: Schema del filtro adattativo che implementa l'algoritmo LMS. Il simbolo * indica il complesso coniugato.

Per ogni iterazione vengono eseguite $2N + 1$ moltiplicazioni complesse (N per il filtraggio e $N + 1$ per l'aggiornamento) e $2N$ somme complesse. Dunque la complessità di questo algoritmo è $O(N)$.

Sulla scelta di μ

Il parametro che svolge un ruolo fondamentale nel determinare le prestazioni dell'algoritmo è μ . Questo valore infatti determina il peso con cui l'errore di stima influisce sull'aggiornamento dei coefficienti che, grazie all'accumulazione, rappresentano una media su un numero crescente di valori. Non esiste un valore universalmente valido da assegnare a tale parametro e per questo è necessario eseguire alcune prove numeriche per individuare la soluzione ottima. In generale vale

$$0 < \mu < \frac{2}{\text{potenza statistica del vettore d'ingresso}}. \quad (\text{B.5})$$

Al diminuire di μ si hanno stime più accurate ma tempi di convergenza più elevati. Al contrario, a valori maggiori di questo parametro corrispondono tempi di convergenza più limitati ma errori di stima più accentuati.

Nel caso trattato vale la (B.4) e dunque non è necessario avere tempi di convergenza molto ridotti. Per questo i valori di μ presi in considerazione sono tra 0.1 e 0.005. Con valori maggiori si ha solo un'amplificazione dell'effetto del rumore bianco sulla stima. Come per l'LMS via RAM è stato valutato l'errore e_k in varie situazioni. Queste simulazioni sono divise in due principali casi: nelle Figure B.2 e B.3 viene riportato il caso teorico in cui vengono

forniti i dati effettivamente trasmessi d_k mentre nelle Figure B.4 e B.5 viene riportato il caso in cui i dati sono ricavati dai LLR calcolati dal Max-Log-MAP. Come prevedibile, nel primo caso le migliori prestazioni sono del LMS classico rispetto a quelle del LMS via RAM. Nel secondo caso invece, che rappresenta la situazione in cui effettivamente opereranno gli algoritmi, l'LMS via RAM sembra risentire meno degli errori di detection commessi dal Max-Log-MAP. Questo è dovuto al fatto che l'algoritmo LMS aggiorna ad ogni iterazione tutti i suoi tappi e dunque un errore di rivelazione ha conseguenze più ampie rispetto a quelle provocate nel LMS via RAM in cui l'errore in un dato istante colpisce solo parte della memoria e dunque della stima.

B.1 Errori di stima con simboli non affetti da errori

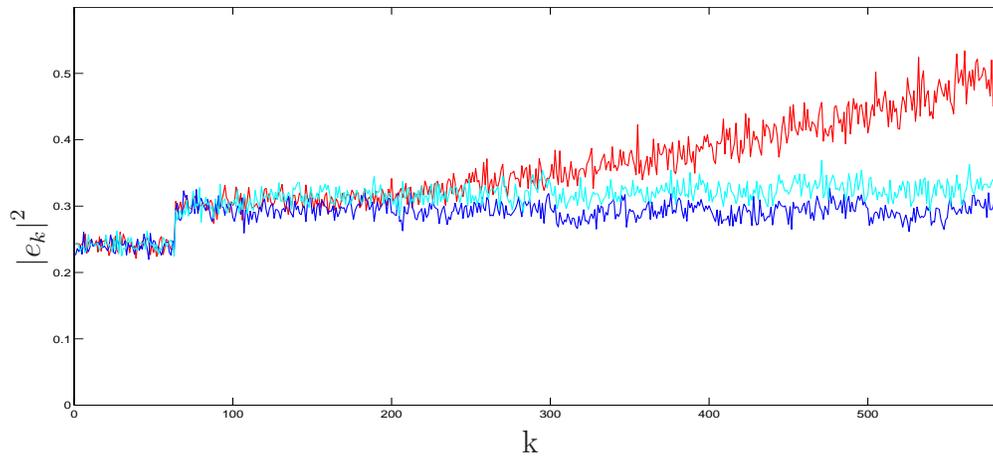


Figura B.2: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 *bit* per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale *typical case*, $\Gamma = 8$ dB.

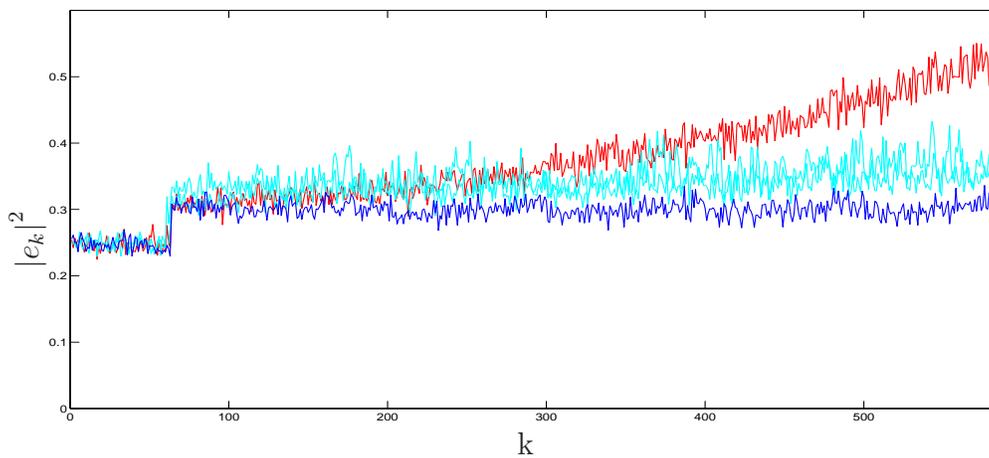


Figura B.3: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 *bit* per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale *worst case*, $\Gamma = 8$ dB.

B.2 Errore di stima con simboli ricavati dai LLR

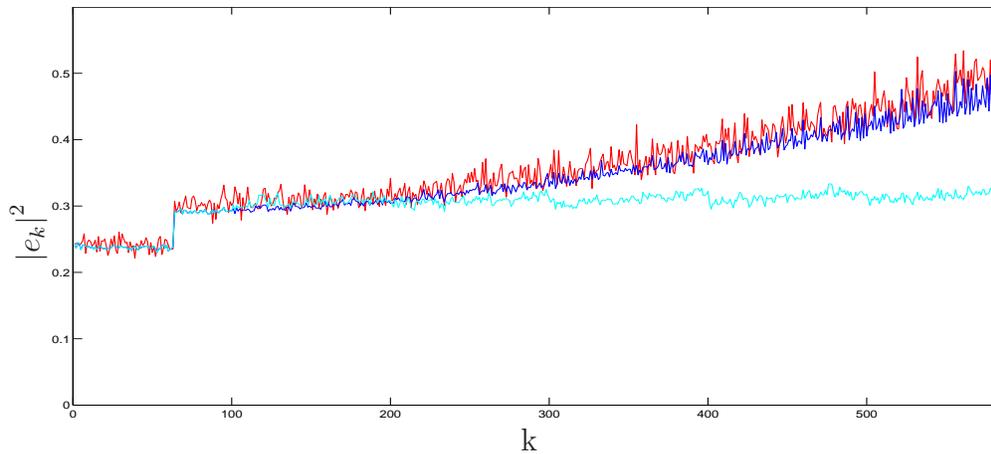


Figura B.4: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 *bit* per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale *typical case*, $\Gamma = 8$ dB.

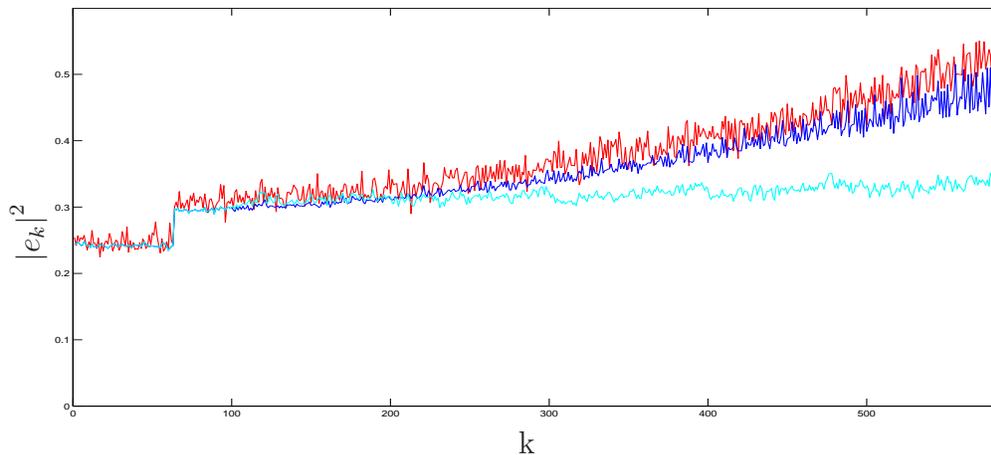


Figura B.5: Andamento dell'errore e_k senza correzione (rosso) e con correzione di stima usando 3 *bit* per indirizzare la memoria nel LMS via RAM (azzurro) e $\mu = 0.01$ nel LMS classico (blu). Canale *worst case*, $\Gamma = 8$ dB.

B.3 Prestazioni con correzione di stima via LMS

Vediamo le prestazioni del Max-Log-MAP con l'algoritmo LMS. In ognuno dei quattro casi si ha un miglioramento rispetto a quanto visto nelle Figure 5.24 - 5.27. Tuttavia il miglioramento apportato è inferiore a quello visto nel caso del LMS via RAM.

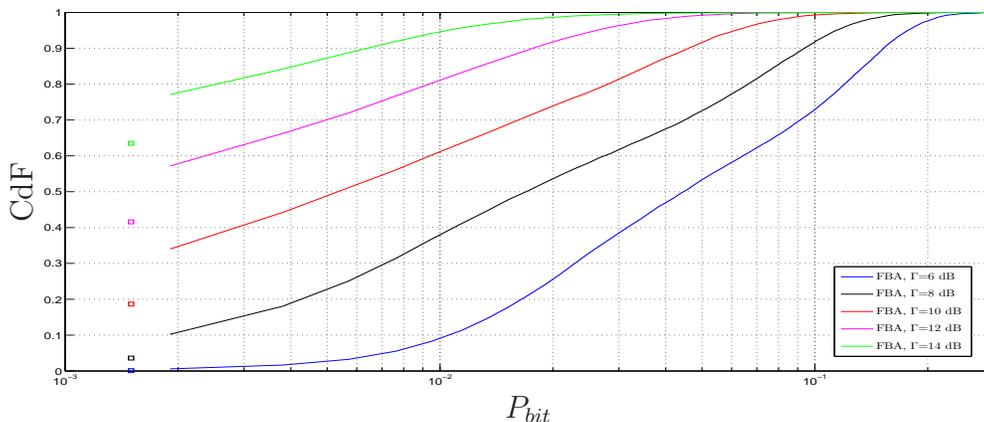


Figura B.6: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 *bit*.

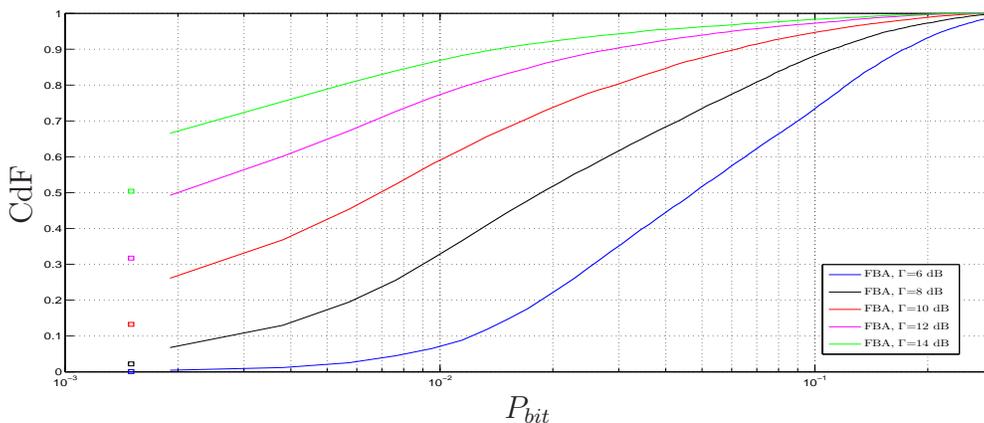


Figura B.7: Funzione distribuzione della P_{bit} in assenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 *bit*.

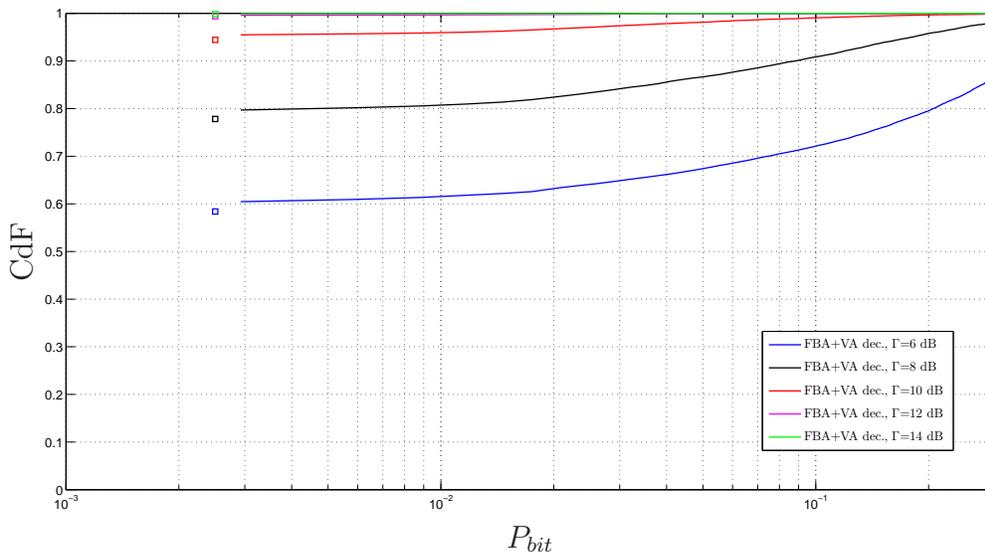


Figura B.8: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *typical case* tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 *bit*.

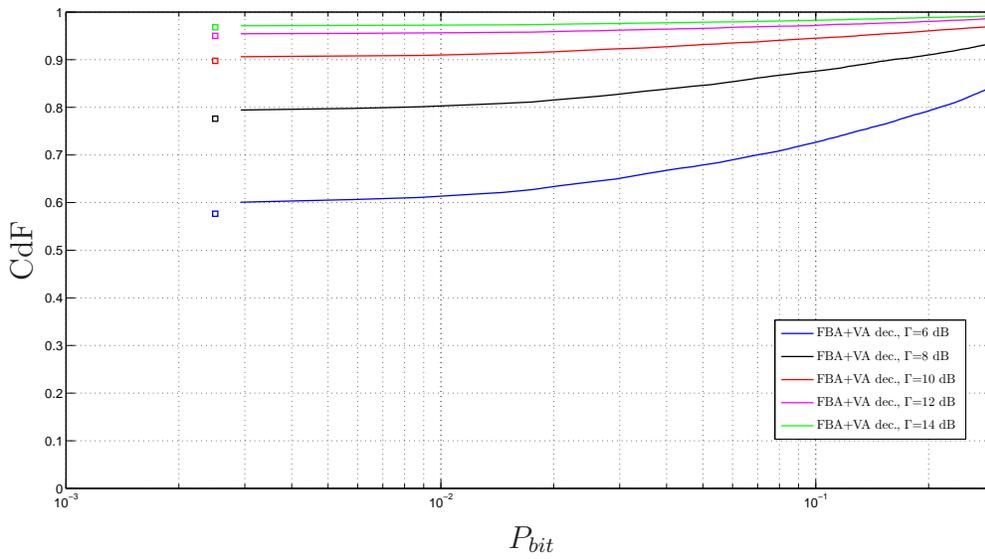


Figura B.9: Funzione distribuzione della P_{bit} in presenza di codifica per diversi valori di Γ . Il caso “senza errori” è riportato con un quadrato sulla sinistra. Canale *worst case* tempo variante. Stima di canale via LMS ($\mu = 0.01$) ogni 100 *bit*.

Bibliografia

- [1] N. Benvenuto e G. Cherubini, “*Algorithms for Communications Systems and their Applications*”. Chichester: Wiley, 2002.
- [2] P. Laurent, “Exact and approximate construction of digital phase modulations by superposition of amplitude modulate pulses (AMP)”, *IEEE Trans. on Communications*, vol. 34, pagg. 150-160, Feb. 1986.
- [3] G.L. Lui, “Threshold detection performance of GMSK signal with BT=0.5”, *IEEE Military Communications Conference, 1998. MILCOM 98*. vol. 2.
- [4] Andrea Goldsmith, “*Wireless Communications*”. Cambridge University Press, 2005.
- [5] G. Cariolaro, G. Pierobon, *Processi aleatori*. Padova: libreria Progetto, 2008.
- [6] A. Anastasopoulos e K. M. Chugg, “An efficient method for simulation of frequency selective isotropic Rayleigh fading”, *IEEE Vehicular Technology Conference*, vol. 3, 1997.