POLITECNICO DI TORINO

Facoltà di Ingegneria
Corso di Laurea in Ingegneria delle telecomunicazioni

Tesi di Laurea

Sistema di telecomando e telemetria per aerei stratosferici senza pilota: definizione delle specifiche e validazione tramite simulazione

Relatori:
Prof. Mario PENT
Prof. Marina MONDIN
Ing. Fabio DOVIS

Candidati:
Francesco BAIETTO
Maurizio FODRINI
Maurizio SISTO

Dicembre 2000

Contributi dei candidati

Questa tesi è stata realizzata mediante una stretta collaborazione ed una costante cooperazione tra i candidati. Tutte le problematiche affrontate sono state esaminate singolarmente, suddividendole opportunamente per consentire un'analisi parallela dei relativi aspetti da studiare. I risultati così ottenuti sono poi stati analizzati e confrontati per decidere la strategia più adatta a risolvere tali problematiche, in modo da disporre di una base di partenza con cui impostare il problema successivo. In particolare, il contributo fornito da ciascun tesista per lo studio dei vari argomenti è stato il seguente:

- *Baietto*: analisi del modello di telemetria proposto dal CCSDS e del flusso dati in down-link, realizzazione del modello di canale, teoria della modulazione OFDM, realizzazione e simulazione dello schema DOFDM, teoria e simulazione dei codici a blocco, valutazione delle prestazioni del codice di Reed-Solomon, scansione elettronica delle schiere, antenne ad elica, valutazione del margine disponibile.

- *Fodrini*: analisi dei servizi di telemetria proposti dal CCSDS, trattazione teorica del modello del canale, teoria e simulazione della modulazione GMSK, realizzazione dello schema DOFDM, teoria e simulazione dei codici convoluzionali, valutazione dei ritardi complessivi del sistema, schiera ottenuta mediante sintesi FTM e antenna a dipoli incrociati, valutazione del link-budget per le diverse situazioni di volo.

- *Sisto*: analisi del modello di telecomando proposto dal CCSDS, analisi del flusso dati in up-link, teoria e simulazione delle modulazioni DQPSK e $\pi/4$ DQPSK, teoria e valutazione delle prestazioni fornite dal protocollo ARQ, analisi della correlazione tra i bit, schiere di tipo lineare con distribuzione uniforme, a coseno e a coseno quadrato, studio dei parametri riguardanti il calcolo del link-budget.

Ringraziamenti

Esprimiamo i nostri ringraziamenti al prof. Mario PENT e alla prof.ssa Marina MONDIN per il supporto fornitoci durante lo svolgimento di questa tesi e, in modo particolare, all'ing. Fabio DOVIS per la sua assistenza e costante disponibilità presentata nei nostri confronti.

Siamo grati, inoltre, all'ing Gabriella POVERO per la sua consulenza in merito al simulatore utilizzato, nonché ai docenti dei Dipartimenti di ingegneria Aeronautica e Spaziale, di Energetica e di Elettrica Industriale per le informazioni che gentilmente ci hanno fornito, relativamente al flusso dati.

Desideriamo anche ringraziare il tesista Roberto FANTINI per il suo contributo apportato nello studio del canale di propagazione.

Porgiamo, poi, la più sentita riconoscenza in modo particolare ai nostri rispettivi genitori, Lorenzo e Cesira BAIETTO, Italo e Maria Rosa FODRINI, Ugo e Michela SISTO, per la loro costante presenza ed il loro continuo supporto manifestatoci durante tutti questi anni.

Esprimiamo infine i nostri ringraziamenti a tutti gli amici e a quelle persone che hanno creduto nelle nostre capacità.

Indice

Contributi dei candidati I

Ringraziamenti III

1 Introduzione **1**

2 Schema proposto dal CCSDS per il telecomando e la telemetria **7**

2.1 Introduzione . . . 7
2.2 Telecomando . . . 8
2.3 Telecomando: flusso dati al trasmettitore . . . 11
2.3.1 Application Process Layer . . . 11
2.3.2 System Management Layer . . . 11
2.3.3 Packetization Layer . . . 12
2.3.4 Segmentation Layer . . . 14
2.3.5 Transfer Layer . . . 15
2.3.6 Coding Layer . . . 21
2.3.7 Physical Layer . . . 25
2.4 Telecomando: flusso dati al ricevitore . . . 26
2.4.1 Physical Layer . . . 27
2.4.2 Coding Layer . . . 27
2.4.3 Transfer Layer . . . 28
2.4.4 Segmentation Layer . . . 32
2.4.5 Packetization Layer . . . 32
2.4.6 System Management Layer . . . 33
2.4.7 Application Process Layer . . . 33

2.5 Telemetria 33
2.5.1 Layer D: Space Transfer Layer 35
2.5.2 Layer C: Virtual Channel Access Layer 40
2.5.3 Layer B: Channel Access Layer 45
2.5.4 Layer A: Physical Access Layer 46
2.6 Riepilogo dello standard CCSDS 53
2.6.1 Telecomando secondo il CCSDS 53
2.6.2 Telemetria secondo il CCSDS 56
2.6.3 Applicabilità dello schema CCSDS ad Heliplat 57

3 Analisi del flusso dati in up-link e down-link 61
3.1 Introduzione 61
3.2 Dati aeronautici 63
3.2.1 Parametri relativi agli assi corpo 64
3.2.2 Parametri relativi alla posizione 66
3.2.3 Parametri relativi alla pressione 69
3.2.4 Parametri relativi alla velocità 70
3.2.5 Bit rate relativo al flusso dati aeronautico 71
3.3 Dati energetici 72
3.3.1 Principio di funzionamento delle celle a combustibile . 73
3.3.2 Parametri del sistema energetico 75
3.3.3 Bit rate relativo al flusso dati energetico 78
3.4 Dati elettrici 79
3.4.1 Principio di funzionamento delle celle solari 79
3.4.2 Parametri del sistema elettrico 81
3.4.3 Bit rate relativo al flusso dati elettrico 82
3.5 Flusso dati complessivo in down-link 83
3.6 Flusso dati complessivo in up-link 84

4 Il canale di propagazione 87
4.1 Introduzione 87
4.2 L'effetto dei multipath nel tempo 89
4.2.1 Risposta all'impulso del canale affetto da multipath . 90
4.3 L'effetto dei multipath in frequenza 93

4.3.1 L'effetto Doppler 93
4.3.2 L'effetto Doppler in presenza di multipath 95
4.4 Parametri caratteristici del canale mobile soggetto a multipath 97
4.4.1 Parametri relativi alle caratteristiche del canale 97
4.4.2 Effetti del canale sul segnale trasmesso 99
4.5 Modellizzazione del canale di propagazione 100
4.5.1 Modello generale del canale 101
4.5.2 Modello del canale simulato 104

5 Schemi di modulazione analizzati 107
5.1 Introduzione 107
5.2 La modulazione OFDM 108
5.2.1 Il segnale 108
5.2.2 Ortogonalità delle portanti 110
5.2.3 Spettro del segnale 112
5.2.4 Efficienza spettrale 113
5.2.5 Energia e potenza 114
5.2.6 Intervallo di guardia 115
5.2.7 Modulatore e demodulatore 118
5.2.8 Prestazioni su canale AWGN 119
5.3 Le modulazioni DQPSK e $\pi/4$ DQPSK 121
5.3.1 Il segnale 123
5.3.2 Spettro del segnale 124
5.3.3 Efficienza spettrale 126
5.3.4 Energia e potenza 126
5.3.5 Modulatore e demodulatore DQPSK 128
5.3.6 Modulatore e demodulatore $\pi/4$ DQPSK 130
5.3.7 Prestazioni su canale AWGN 133
5.4 La modulazione GMSK 133
5.4.1 Il segnale 135
5.4.2 Filtro gaussiano e spettro del segnale 137
5.4.3 Efficienza spettrale 140
5.4.4 Modulatore e demodulatore 140
5.4.5 Prestazioni su canale AWGN 144

6 Simulazioni per la scelta della modulazione **147**
6.1 Introduzione 147
6.2 La modulazione OFDM 149
6.3 La modulazione DOFDM 151
6.3.1 Prestazioni ottenute mediante simulazione 152
6.4 La modulazione DQPSK 154
6.4.1 Prestazioni ottenute mediante simulazione 157
6.5 La modulazione $\pi/4$ DQPSK 158
6.5.1 Prestazioni ottenute mediante simulazione 159
6.6 La modulazione GMSK 159
6.6.1 Prestazioni ottenute mediante simulazione 161
6.7 Confronti tra le modulazioni 163
6.7.1 Confronto tra la DOFDM e la DQPSK 163
6.7.2 Confronto tra la DQPSK e la $\pi/4$ DQPSK 164
6.7.3 Confronto tra la DQPSK e la GMSK 166
6.7.4 Considerazioni conclusive 167

7 Tecniche di codifica analizzate **169**
7.1 Introduzione 169
7.2 Generalità sui codici di canale 170
7.2.1 Cenni sui codici a blocco 172
7.2.2 Un sottoinsieme di codici a blocco: i codici ciclici . . . 175
7.2.3 I codici concatenati 177
7.3 Codici esaminati 179
7.3.1 I codici BCH 180
7.3.2 I codici di Reed-Solomon 181
7.3.3 I codici convoluzionali 182
7.4 La tecnica ARQ 188
7.4.1 Stop-and-Wait 189
7.4.2 Go-Back-N 190
7.4.3 Selective-Repeat 191
7.4.4 Prestazioni della tecnica ARQ 192

8 Simulazioni delle tecniche di codifica 197
8.1 Introduzione 197
8.2 Confronto delle prestazioni con DQPSK a differenti bit rate . 198
8.3 Applicazione della codifica nella tratta in up-link 200
8.3.1 Simulazioni a basso bit rate 202
8.3.2 Simulazioni ad alto bit rate 205
8.4 Applicazione della codifica nella tratta in down-link 206
8.4.1 Simulazione del codice convoluzionale 209
8.4.2 Prestazioni del codice di Reed-Solomon 214
8.5 Utilizzo del protocollo ARQ 216
8.5.1 Prestazioni del protocollo ARQ 218
8.5.2 Trattazione teorica dei ritardi 221
8.5.3 Valutazione dei ritardi 226
8.6 Analisi della correlazione tra i bit 232
8.6.1 Analisi dei Gap nella tratta in up-link 234
8.6.2 Analisi dei Gap nella tratta in down-link 239
8.7 Considerazioni Conclusive 241

9 Il sistema di antenne: progetto preliminare 244
9.1 Introduzione 244
9.2 Generalità sulle antenne 244
9.2.1 Parametri caratteristici delle antenne 245
9.3 Studio del sistema di antenne di terra 251
9.3.1 Schiera lineare ad ampiezze unifomi 254
9.3.2 Schiera lineare non uniforme ottenuta per mezzo di sintesi FTM 265
9.3.3 Schiera lineare non uniforme con distribuzione d'ampiezza a coseno 269
9.3.4 Schiera lineare non uniforme con distribuzione d'ampiezza a coseno quadrato 273
9.4 Considerazioni conclusive sulle antenne di terra 276
9.5 Scansione elettronica delle fasi per l'antenna di terra 278
9.6 Studio del sistema di antenne di bordo 281
9.6.1 Soluzione con più antenne a elica 282

9.6.2 Antenna a dipoli incrociati 291
9.7 Considerazioni conclusive sulle antenne di bordo 298

10 Analisi del Link-Budget **301**
10.1 Introduzione 301
10.2 Trattazione teorica sul link-budget 302
10.2.1 Attenuazione di spazio libero 303
10.2.2 Extra-attenuazione media del mezzo trasmissivo 304
10.2.3 Potenza di rumore 306
10.2.4 Perdite da polarizzazione 309
10.2.5 Perdite realizzative 310
10.3 Link-budget relativo alla tratta in up-link 310
10.3.1 Collegamento a 32 kbit/s 311
10.3.2 Collegamento a 100 kbit/s 314
10.4 Link-budget relativo alla tratta in down-link 315
10.5 Considerazioni conclusive 317

11 Conclusioni **319**

Bibliografia **323**

Capitolo 7

Tecniche di codifica analizzate

7.1 Introduzione

I canali di comunicazione sono affetti da notevoli disturbi quali rumore, distorsioni o fading e questo fa sì che il flusso binario d'informazione trasmesso su di esso contenga degli errori. Le prestazioni di un sistema di comunicazione, in termini di probabilità d'errore, possono essere migliorate aumentando la potenza trasmessa, la larghezza di banda o la complessità realizzativa. Le prime due tecniche agiscono su delle risorse preziose e limitate, per cui generalmente ci si orienta sull'ultima tecnica, cioè sulla *codifica di canale*. Essa si basa sull'introduzione di ridondanza nella sequenza d'informazione per migliorarne l'affidabilità; questo risultato è però ottenuto a scapito dell'efficienza di trasmissione. A seconda della modalità seguita dal sistema nell'utilizzo delle capacità del codice, possono essere distinti i codici *rivelatori d'errore* o quelli *correttori d'errore*.

Per poter recuperare l'informazione codificata, in ricezione si utilizza, dopo il demodulatore, un decodificatore che può lavorare secondo due differenti strategie. Nella prima esso osserva la sequenza ricevuta e rivela l'eventuale presenza di errori; a questo punto può limitarsi ad indicare all'utente la qualità della sequenza ricevuta, in modo da scartarla quando l'attendibilità diviene troppo bassa (tecnica *error monitoring*). Il decodificatore può anche realizzare la tecnica ARQ (*Automatic Repeat Request*) nella quale è necessaria una retroazione tra ricevitore e trasmettitore per effettuare la ritrasmissione delle sequenze ricevute in modo errato (la tecnica ARQ verrà analizzata nel § 7.4).

La seconda strategia è detta FEC (*Forward Error Correction*), nella quale il decodificatore tenta di recuperare la corretta sequenza trasmessa ogni qual volta rileva un errore nella sequenza ricevuta. Questa tecnica richiede, ovviamente, degli algoritmi di codifica con una maggior complessità ed è utilizzata nei sistemi che non dispongono di un canale bidirezionale o necessitano di bassissimi ritardi (applicazioni in tempo reale).

7.2 Generalità sui codici di canale

Un semplice sistema codificato è rappresentato in figura 7.1. Il codificatore di

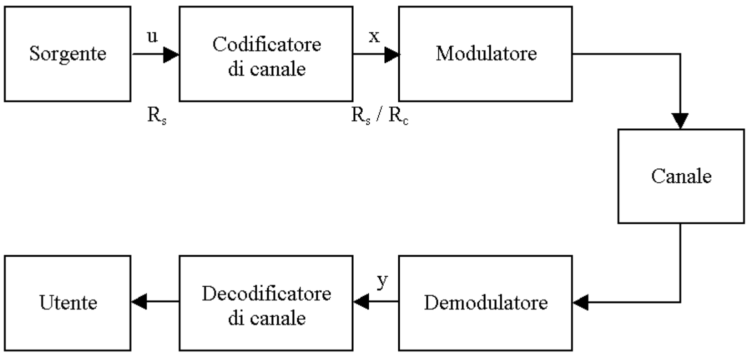

Figura 7.1. Diagramma a blocchi di un generico sistema codificato

canale riceve una sequenza di bit d'informazione dalla sorgente con un rate pari a R_s bit al secondo, ne aggiunge alcuni di ridondanza per proteggere l'informazione e li spedisce sul canale ad una velocità più elevata data da R_s/R_c. Con R_c si indica il rate del codice espresso come il rapporto tra il numero di bit k in ingresso al codificatore di canale e quello in uscita n, cioè $R_c = k/n$. Il codificatore è detto sistematico quando i primi k bit di una parola di codice lunga n bit coincidono con quelli relativi alla parola d'informazione.

La codifica introduce alcune differenze rispetto alla trasmissione non codificata. Innanzitutto, poiché la sorgente trasmette ad un rate R_s e il codificatore aggiunge dei bit di ridondanza, la velocità del trasmettitore sul canale deve essere incrementata di un fattore R_s/R_c per poter mantenere la velocità di trasmissione della sorgente. Questo implica che anche la larghezza di banda deve essere incrementata di un fattore $1/R_c$, con conseguente riduzione dell'efficienza spettrale rispetto alla trasmissione non codificata.

Altra caratteristica della codifica riguarda la riduzione dell'energia per bit da E_b a E_bR_c; questo è dovuto al fatto che, assumendo costante la potenza trasmessa, la durata della forma d'onda utilizzata è minore rispetto al caso non codificato di un fattore R_c. Ne segue che, in assenza di opportune tecniche di decodifica basate sulla capacità di correzione o rilevazione e siccome il ricevitore opera con un rapporto segnale/rumore più sfavorevole, esso consegnerà al decodificatore una sequenza di bit meno affidabili, con conseguente aumento della probabilità d'errore sul bit.

Il processo di decodifica può essere distinto in due diverse classi denominate *hard decision* o *soft decision*; nel primo caso l'uscita del demodulatore è quantizzata su due soli livelli (0 e 1) e la sequenza di bit così ottenuti è mandata al decodificatore che tenterà di recuperare l'informazione grazie alla ridondanza. Nel secondo caso, invece, l'uscita del demodulatore, che rappresenta la statistica sufficiente, non è quantizzata su due soli livelli, ma su $Q > 2$. La soft decision presenta una maggior attendibilità rispetto alla hard decision, in quanto può avvantaggiarsi dell'informazione addizionale contenuta nei campioni non quantizzati.

Il vantaggio di utilizzare una trasmissione codificata rispetto ad una senza codifica è valutato mediante il *guadagno di codifica* che è definito come la differenza, espressa in decibel, nel valore di E_b/N_0 richiesto per raggiungere una data probabilità d'errore tra una trasmissione non codificata e quella codificata. Esso, dunque, dipende dal rapporto segnale/rumore e, in particolare, quando tale quantità raggiunge un valore troppo basso la codifica peggiora le prestazioni, in quanto il numero elevato di errori supera le capacità di rilevazione/correzione del codice e la correlazione introdotta tra i bit trasmessi, causa un aumento del numero di bit errati rispetto al caso non codificato.

Esistono vari tipi di codice che differiscono per la modalità con cui il codificatore assegna le parole di codice rispetto a quelle d'informazione. Si parla di *codice a blocco* quando il codificatore è senza memoria, cioè quando agli stessi k bit d'informazione corrispondono sempre gli stessi n bit di codice; ne risulta che ciascuna parola di codice è ottenuta indipendentemente da quelle precedenti. Quando il codificatore ha memoria, invece, si ottengono i *codici ad albero*; in questo caso gli n bit della parola di codice dipendono non solo dai k bit d'informazione attuali, ma anche da alcune parole di dati precedenti.

I codici possono essere distinti anche per mezzo delle proprietà dell'insieme delle parole di codice. Nei *codici lineari* la somma (realizzata modulo 2) tra due parole di codice qualsiasi fornisce nuovamente una parola che è ancora di codice; quando questa proprietà non è soddisfatta si parla di *codici non lineari*.

Premessa questa breve descrizione generale, nel seguito verranno analizzate più in dettaglio alcune particolari categorie di codici a cui appartengono quelli che sono stati scelti per raggiungere le prestazioni desiderate nel sistema di telecomando e telemetria di Heliplat. Per maggiori dettagli sulle tecniche di codifica si rimanda a [42].

7.2.1 Cenni sui codici a blocco

Nel § 7.2 si è già accennato che i codici a blocco utilizzano un codificatore senza memoria; in particolare esso può essere realizzato, si veda figura 7.2, sommando un sottoinsieme delle k celle di un registro a scorrimento contenente la parola d'informazione, e inserendo ciascun risultato in un altro registro a scorrimento di n posizioni che contiene la parola codificata. La regola per la codifica dell'informazione può essere rappresentata in modo conciso per mezzo della *matrice di generazione* G del tipo $k \times n$, nella quale la posizione (i, j) è pari ad '1' se l'i-esima cella del registro d'ingresso è connessa con il j-esimo sommatore. Se si indica con u la parola d'informazione e con x quella di codice, la regola citata può essere espressa matematicamente come:

$$\bar{x} = \bar{u}G \tag{7.1}$$

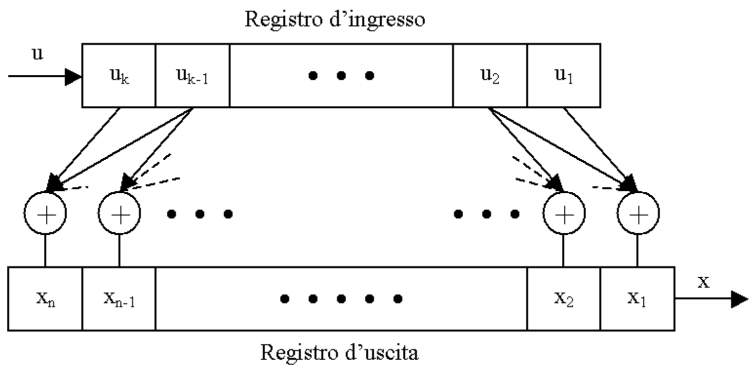

Figura 7.2. Schema di un generico codificatore per un codice a blocco

Se il codificatore è sistematico, allora le prime k colonne di G formano una matrice identità $k \times k$ e, quindi, G può essere rappresentata nella forma:

$$G = [I_k|P] \tag{7.2}$$

dove I_k indica la matrice identità $k \times k$ e P la matrice di controllo di parità $k \times (n-k)$.

I codici a blocco sono costituiti da tutte le possibili somme delle righe della matrice di generazione e dalla n-upla di tutti zeri, che è sempre una parola di codice; inoltre essi sono codici lineari. Le k righe di G sono linearmente indipendenti e, quindi, qualunque insieme di k parole di codice può essere usato come matrice di generazione. Risulta pertanto che, mediante opportune operazioni sulle righe e permutazioni sulle colonne, la matrice può essere trasformata nella forma sistematica; questi cambiamenti portano a codici detti *equivalenti* e lasciano inalterata la probabilità d'errore sulla parola, mentre quella sul bit può differire.

Un importante parametro delle parole di codice è il loro *peso di Hamming*, cioè il numero di '1' che esse contengono e, considerando tutti i pesi delle parole di codice, si ottiene la *distribuzione dei pesi* che, per quanto detto in precedenza, è la stessa per tutti i codici equivalenti. Altra quantità fondamentale è la *distanza di Hamming*, definita come il numero di posizioni in cui due parole di codice differiscono tra loro; la più piccola tra esse è indicata come *distanza minima* d_{min} e caratterizza le prestazioni del codice. La distanza minima di un codice a blocco lineare coincide anche con il più piccolo peso di Hamming delle sue parole di codice. Si può dimostrare, si veda [29] (pp. 413-414), che un codice a blocco con distanza minima d_{min} può rilevare fino a $(d_{min} - 1)$ errori, mentre può correggere fino a t errori, dove:

$$t = \lfloor (d_{min} - 1)/2 \rfloor \tag{7.3}$$

La rivelazione e la correzione degli errori avviene mediante il calcolo della *sindrome* s, cioè un vettore che contiene il risultato dei controlli di parità realizzati sulla parola di codice ricevuta. Essa può essere espressa come:

$$s = yH' \tag{7.4}$$

dove con y si è indicata la sequenza ricevuta e l'apice su H indica l'operazione di trasposizione; H, invece, è una matrice $(n-k)\times n$ detta *matrice di controllo di parità*, in quanto le sue righe rappresentano i simboli di controllo di parità calcolati dal decodificatore, cioè:

$$H = [P'|I_{n-k}] \tag{7.5}$$

e, ricordando l'equazione (7.2), si ha che $GH' = 0$. Si noti che il canale può introdurre 2^n vettori d'errore differenti e, di questi, 2^k non sono rilevabili in quanto corrispondono all'insieme delle parole di codice. Per la correzione degli errori si utilizza lo *standard array*, una tabella di 2^k colonne e 2^{n-k} righe. Queste ultime sono dette *cosets*: la prima riga contiene le parole di codice, mentre la prima parola di ciascuna riga è detta *cosets leader*. L'algoritmo di decodifica calcola la sindrome per la sequenza ricevuta, trova il vettore correttore d'errore (coset leader) corrispondente e quindi stima la parola di codice sommando quest'ultimo alla parola ricevuta[1]. Per quanto riguarda le prestazioni si può dimostrare, si veda [29] (pp. 446-455), che nel caso di hard decision valgono le seguenti relazioni, per la probabilità d'errore sulla parola:

$$P_w(e) \leq (M-1)\left[\sqrt{4p(1-p)}\right]^{d_{min}} \tag{7.6}$$

e per quella sul bit:

$$P_b(e) \cong \frac{2t+1}{n} P_w(e) \tag{7.7}$$

[1] Le procedure complete di rivelazione e correzione degli errori sono trattate in [29] (pp. 411-417).

dove $M = 2^k$ è il numero di parole di codice (assunte equiprobabili), p la probabilità di transizione del canale, d_{min} la distanza minima del codice e t la sua capacità di correzione. Nel caso di soft decision, invece, si ottiene:

$$P_w(e) \leq \frac{M-1}{2} erfc\left(\sqrt{R_c d_{min} \frac{E_b}{N_0}}\right) \tag{7.8}$$

dove con R_c si è indicato il rate del codice, mentre per la probabilità d'errore sul bit resta valida l'equazione (7.7).

7.2.2 Un sottoinsieme di codici a blocco: i codici ciclici

I codici ciclici fanno parte dei codici a blocco e, come tali, presentano tutte le proprietà analizzate nel paragrafo precedente. In più, però, hanno la peculiarità di presentare una ben definita struttura matematica che consente delle facili operazioni di codifica e dei semplici algoritmi di decodifica; per questa ragione essi hanno un notevole interesse pratico.

Un codice (n, k) si definisce ciclico se una qualunque rotazione ciclica di una parola di codice conduce ancora ad un'altra parola di codice. In questa trattazione è utile rappresentare una sequenza binaria di n bit mediante un polinomio di grado non maggiore di $(n-1)$ avente dei coefficienti binari (cioè di valore '0' o '1'); in particolare se la parola di codice è $x = [x_{n-1}, x_{n-2}, \ldots, x_1, x_0]$, la sua rappresentazione polinomiale vale:

$$x(Z) = x_{n-1}Z^{n-1} + x_{n-2}Z^{n-2} + \ldots + x_1 Z + x_0 \tag{7.9}$$

Da questa rappresentazione segue che le parole di codice sono esprimibili con dei *polinomi di codice* legati tra loro da shift ciclici; questi ultimi vengono indicati con $x^{(i)}(Z)$ e sono ottenuti[2] prendendo il resto della divisione di $Z^i x(Z)$ per $(Z^n + 1)$. Come conseguenza, anche la matrice di generazione G analizata nel § 7.2.1 può essere riscritta in forma algebrica e, in particolare, ciascuna riga può essere espressa come multiplo di un polinomio $g(Z)$ che è l'unico ad avere grado $(n-k)$: esso è detto *polinomio generatore*. Qualsiasi codice ciclico è completamente definito dal suo polinomio generatore, in

[2] Questa e le altre dimostrazioni relative alle affermazioni che vengono esposte in questo paragrafo possono essere analizzate in [29], pp. 421-435.

quanto gli altri $2^k - 1$ polinomi di codice sono multipli di $g(Z)$; inoltre si può dimostrare che qualsiasi polinomio di grado $(n-1)$ o minore che è divisibile per $g(Z)$ deve essere un polinomio di codice. È possibile notare che $g(Z)$ è un divisore di $(Z^n + 1)$ e, quindi, importanti risultati algebrici sono disponibili guardando le proprietà di tale monomio.

Indicando con $u(Z)$ il polinomio relativo alla sequenza d'informazione, il polinomio codificato può essere ottenuto secondo il seguente algoritmo:

$$x(Z) = u(Z)g(Z) \tag{7.10}$$

Esso però non rappresenta un codificatore sistematico; per renderlo tale è necessario ripetere il polinomio della sequenza d'informazione aggiungendo poi quello di controllo di parità che può essere ottenuto prendendo il resto $r(Z)$ della divisione di $Z^{n-k}u(Z)$ per il polinomio generatore $g(Z)$, cioè:

$$x(Z) = Z^{n-k}u(Z) + r(Z) \tag{7.11}$$

Quanto esposto rende necessario realizzare l'operazione di divisione per ottenere il resto $r(Z)$ rappresentante la sezione di controllo di parità. Tale compito può essere realizzato mediante un registro a scorrimento di $(n-k)$ stadi opportunamente retroazionato mediante connessioni che corrispondono ai coefficienti del divisore. Ad ogni impulso di clock, il dividendo viene inserito nel registro e alla sua uscita si ottiene direttamente il quoziente. Secondo questa strategia esistono diverse realizzazioni del codificatore; una di esse è rappresentata in figura 7.3 per il caso specifico di un classico codice ciclico: il codice di Hamming. Con l'interruttore in (A) viene trasferita la parte sistematica, mentre quando è in (B) si spedisce la parte relativa al controllo di parità. Nel caso in cui $k < (n-k)$, è più conveniente realizzare un codificatore con k stadi basato sul *polinomio di controllo di parità* definito come:

$$h(Z) \triangleq \frac{Z^n + 1}{g(Z)} \tag{7.12}$$

Raramente però ci si trova in questa condizione in quanto essa comporta l'introduzione di una ridondanza più che doppia rispetto ai bit utili.

Anche la decodifica è analizzabile algebricamente. Per quanto concerne la rivelazione degli errori è sufficiente controllare il resto della divisione del polinomio $y(Z)$ ricevuto per il polinomio generatore $G(Z)$; esso è detto *polinomio*

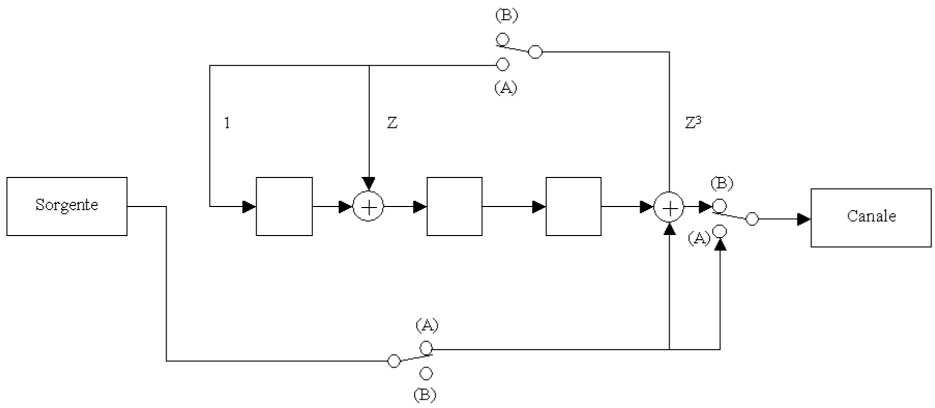

Figura 7.3. Tipica realizzazione del codificatore per il codice di Hamming (7,4) con polinomio generatore $g(Z) = Z^3 + Z + 1$

sindrome, è anch'esso calcolato mediante un registro a scorrimento opportunamente retroazionato e nel caso in cui sia nullo non vi sono errori rivelati. Per eseguire la correzione degli errori il decodificatore deve essere in grado di trovare un polinomio correttore d'errore $e(Z)$ a partire dalla sindrome; in questo modo la parola di codice trasmessa $x(Z)$ è ottenuta sommando $e(Z)$ alla sequenza ricevuta $y(Z)$. Le tecniche per il calcolo di $e(Z)$ sono, in generale, abbastanza complesse; tuttavia nel caso della correzione di errori singoli vi è una semplice tecnica applicabile, basata sul fatto che se gli errori di $e(Z)$ sono confinati nelle $(n-k)$ posizioni di controllo di parità di $y(Z)$, allora la sindrome $s(Z)$ coincide con $e(Z)$. In questo caso risulta quindi sufficiente sommare la sindrome agli $(n-k)$ bit di parità ricevuti. Se gli errori sono consecutivi, ma non sono confinati nei bit di parità, è sufficiente effettuare una rotazione ciclica per riportarli in tale posizione.

7.2.3 I codici concatenati

I codici concatenati sono una classe di codici studiata per riuscire ad ottenere una probabilità d'errore esponenzialmente decrescente, ma con una complessità di decodifica crescente solo algebricamente. Per questo motivo essi sono utilizzati in quelle applicazioni che richiedono un alto guadagno di codifica, ma necessitano di una complessità limitata.

L'idea di base per realizzazione di un codice concatenato, si veda [29] (pp. 481-483), è quella di utilizzare più codici connessi in serie tra loro; il caso più semplice e più comune è quello di utilizzare due livelli di codifica, come mostrato in figura 7.4. Con questa disposizione si ha che la parola di

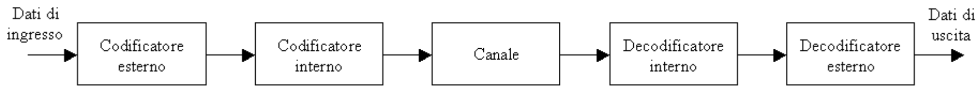

Figura 7.4. Diagramma a blocchi di un generico codice concatenato a due livelli

codice all'uscita del primo codificatore rappresenta la parola d'informazione all'ingresso del secondo codificatore, la cui uscita rappresenta la parola di codice che verrà effettivamente trasmessa. Nel caso considerato di soli due codici, il primo applicato è detto *outer code*, mentre il successivo *inner code*.

I due codici sono legati tra loro da precise relazioni riguardanti le parole di codice e i rate di codifica. In particolare se i codici esterno e interno sono a blocco, di valori rispettivamente (n_0, k_0) e (n_i, k_i), allora i parametri n_0 e k_i sono strettamente legati, in quanto devono essere multipli o sottomultipli interi tra loro. Tipicamente si ha che n_0 è un multiplo di k_i:

$$n_0 = mk_i \tag{7.13}$$

con m intero, cioè una parola di codice del codice esterno contiene più di una parola d'informazione di quello interno. Per quanto concerne il rate, invece, si ha che quello del codice concatenato è minore rispetto a quelli dei due codici utilizzati e, in particolare, esso è pari al loro prodotto. Infatti, indicando con R_c^0 il rate dell'outer code, con R_c^i quello dell'inner code e con R_c quello totale del concatenato, si può scrivere:

$$R_c = \frac{k_0}{mn_i} = \frac{k_0}{n_0}\frac{n_0}{mn_i} = R_c^0 R_c^i \tag{7.14}$$

Tale risultato, dimostrato per i codici a blocco, è anche valido nel caso si utilizzino dei codici convoluzionali. Il grosso vantaggio dei codici concatenati risiede nella decodifica, in quanto si lavora separatamente sui due codici ottenendo così una complessità minore di quella che si avrebbe utilizzando un unico codice (mn_i, k_0); il ricevitore, infatti, prima decodifica il codice interno e poi quello esterno. I modi di effettuare la decodifica sono differenti, a seconda che il modulatore fornisca all'inner decoder una decisione hard o soft e che quest'ultimo, a sua volta, fornisca all'outer decoder una stima hard o soft dei suoi simboli d'ingresso. Tuttavia è stato dimostrato, si veda [43] (pp. 35-55), che il modo migliore per decodificare in cascata è quello che richiede operazioni soft sia al demodulatore che al decodificatore interno: questa scelta comporta l'utilizzo dell'algoritmo di Viterbi.

Nei codici concatenati viene solitamente inserito un *interleaver* tra il codificatore interno e quello esterno; esso è un dispositivo, si veda quanto esposto nel § 2.5.4, che modifica l'ordine della sequenza di simboli in un modo deterministico. In ricezione vi sarà un *deinterleaver* che applica le operazioni inverse per riottenere la sequenza nel suo ordine originale. Questa operazione consente di far fronte alla presenza di errori a *burst* in quanto aumenta il grado di scorrelazione dei bit, riuscendo così a migliorare le prestazioni complessive del sistema.

7.3 Codici esaminati

Nel sistema di comunicazione relativo alla piattaforma stratosferica Heliplat vanno considerati due canali; si indica con il termine up-link la tratta relativa al collegamento dalla stazione di terra verso il velivolo, cioè la trasmissione relativa all'informazione necessaria per il telecomando, mentre con down-link la tratta complementare, necessaria per la telemetria.

Per la scelta della strategia da utilizzare al fine di garantire una maggior affidabilità nel flusso dati, si è deciso di adottare lo schema presentato dal CCSDS (si veda [7] e [2]). Esso prevede, in up-link, di codificare l'informazione trasmessa mediante un codice BCH (*Bose-Chaudhury-Hocquenghem*) e di utilizzare un protocollo di tipo ARQ, che verrà analizzato nel § 7.4, per migliorare ulteriormente le prestazioni. Quest'ultimo, però, non lavora sui bit, bensì ad un livello superiore dove l'informazione è raggruppata in trame. La trattazione del CCSDS relativa a queste scelte è stata riportata nel § 2.3.6 per quanto concerne il BCH, mentre nel § 2.3.5 e nel § 2.4.3 per quanto riguarda il protocollo ARQ.

Nella tratta in down-link, invece, lo schema CCSDS prevede l'utilizzo di un codice concatenato realizzato mediante un convoluzionale come codice esterno ed un Reed-Solomon come interno, secondo quanto esposto nel § 2.5.4.

Viene dunque affrontata, nel seguito, la trattazione teorica dei codici proposti.

7.3.1 I codici BCH

I codici BCH (si veda [39], pp. 436-438) appartengono alla classe dei codici ciclici, i quali a loro volta sono una sottoclasse dei codici a blocco, come analizzato nel § 7.2.2. Tra i vari tipi di codici ciclici sono tra i più utili per correggere errori casuali fondamentalmente perchè gli algoritmi di decodifica necessari possono essere realizzati con complessità contenuta. Esistono diversi tipi di codici BCH, differenziati a seconda del valore assunto da due numeri interi m e t; in particolare, per una coppia generica si ha un codifica BCH con una parola di codice lunga n data da:

$$n = 2^m - 1 \tag{7.15}$$

mentre i bit di controllo di parità devono soddisfare alla seguente relazione:

$$n - k \leq mt \tag{7.16}$$

Si ottiene, inoltre, una distanza minima fornita da:

$$d_{min} \geq 2t + 1 \tag{7.17}$$

da cui risulta, si ricordi l'equazione (7.3), che questo codice è in grado di correggere tutte le combinazioni fino a t errori. Essendo di tipo ciclico, questa classe di codici viene costruita sulla base di opportuni polinomi generatori e, in particolare, essi sono costituiti dai fattori del monomio $(Z^{2^m-1} + 1)$; la procedura di fattorizzazione, però, non è semplice. Esistono, comunque, delle liste di polinomi generatori in funzione dei parametri m e t che vengono generalmente rappresentate in forma ottale.

L'ampia classe di codici BCH che può essere ottenuta combinando i suddetti parametri è molto utile non solo per la grande flessibilità nella scelta delle caratteristiche del codice (come la lunghezza e il rate), ma anche perchè i BCH con lunghezza dei blocchi di qualche centinaia di bit o meno sono, tra i codici di pari lunghezza e rate conosciuti, quelli migliori. Per quanto riguarda le prestazioni, esse possono essere ricavate dalle espressioni generali indicate nelle equazioni (7.6), (7.7) e (7.8).

7.3.2 I codici di Reed-Solomon

Anche i codici di Reed-Solomon, così come i BCH analizzati in precedenza, appartengono alla classe dei codici ciclici (si veda [40], pp. 2-8) e, in particolare, possono essere considerati come una sottoclasse dei codici BCH. Questi ultimi lavorano nel caso binario, mentre i Reed-Solomon ne sono la generalizzazione nel caso non binario, cioè i simboli appartengono ad un set di cardinalità $q = 2^l$. Malgrado questa differenza, i simboli possono essere rappresentati come delle l-uple binarie e, quindi, il codice può essere considerato come un particolare caso di codice binario ([41], cap. 7).

Nel codice di Reed-Solomon i simboli sono costituiti da l bit, mentre la lunghezza del blocco è pari a $(2^l - 1)$ simboli e quindi in bit si ha:

$$n = l(2^l - 1) \tag{7.18}$$

Gli $(n - k)$ bit di controllo di parità sono costituiti da $2t$ simboli, cioè in bit:

$$(n - k) = 2lt \tag{7.19}$$

dove, al solito, con t è indicata la capacità di correzione del codice. I codici di Reed-Solomon sono in grado di correggere tutte le combinazioni che hanno fino a t simboli errati. Inoltre, se vengono interpretati come codici binari, essi si adattano bene nella correzione di errori che si presentano a burst; in questo caso, infatti, avere un simbolo errato significa avere un numero compreso tra 1 e l di bit adiacenti sbagliati all'interno della parola di codice.

Per quanto riguarda le prestazioni è possibile ottenere, si veda [42], il seguente limite superiore per la probabilità d'errore sul bit dopo una decodifica completa:

$$p_b(e) < \frac{2^{l-1}}{2^l - 1} \sum_{j=t+1}^{n} \frac{j+t}{n} C_j^n p_{sc}^j (1 - p_{sc})^{n-j} \tag{7.20}$$

dove n indica la lunghezza del codice di Reed-Solomon, l il numero di bit per simbolo del codice, mentre C_i^n fornisce il simbolo binomiale dato da:

$$C_i^n = \frac{n!}{i!(n-i)!} \tag{7.21}$$

Con p_{sc}, invece, si indica la probabilità che un simbolo in ingresso al Reed-Solomon sia errato, e vale:

$$p_{sc} = 1 - (1-p)^l \tag{7.22}$$

dove p è la probabilità d'errore sul canale. Quella sul simbolo risulta più complicata da valutare, ma, nel caso di decodifica a coset leader unico[3], si può dimostrare [41] che essa vale:

$$p_s = \frac{1}{n} \sum_{h=t+1}^{n} B_h P(h) \tag{7.23}$$

$P(h)$ rappresenta la probabilità che si verifichi una sequenza d'errore con peso di Hamming pari ad h, cioè:

$$P(h) = \frac{1}{(q-1)^h} p_{sc}^h (1-p)^{n-h} \tag{7.24}$$

qualora il codice di Reed-Solomon sia realizzato su un alfabeto di q simboli. I termini B_h, invece, sono dei coefficienti valutati come il numero di errori residui dopo una decodifica completa; la loro espressione è molto complessa, ma si può ottenere un andamento asintotico considerando solo il primo temine B_{t+1}.

Le applicazioni dei codici di Reed-Solomon sono molteplici, ma essi vengono utilizzati soprattutto negli schemi di codifica concatenati analizzati nel § 7.2.3.

7.3.3 I codici convoluzionali

I codici convoluzionali (per una trattazione completa si veda [43]) si differenziano da quelli a blocco fondamentalmente perchè hanno un codificatore

[3]Questa regola di decisione consiste nel correggere tutti i blocchi d'errore con peso fino a t, altrimenti nel prendere semplicemente i bit d'informazione così come sono stati ricevuti; per i codici di Reed-Solomon (e per altri) questa regola fornisce una probabilità d'errore sul simbolo inferiore rispetto a quella ottenuta decodificando a massima verosimiglianza le parole.

con memoria il quale, cioè, genera n_0 bit in uscita dipendenti non solo dai k_0 bit d'informazione attuali, ma anche da alcune trame precedenti; si noti che, a differenza dei codici a blocco, nei codici convoluzionali k_0 e n_0 sono usualmente numeri piccoli. Inoltre tali codici non hanno tutta la struttura algebrica dei codici a blocco e la maggior parte, infatti, sono stati trovati mediante ricerca esaustiva con tecniche al calcolatore; anche gli algoritmi di decodifica non possono fondarsi sulle proprietà algebriche, ma derivano dalla natura della macchina a stati sequenziale del codificatore convoluzionale.

La struttura generale di un codificatore convoluzionale è rappresentata in figura 7.5. I bit d'informazione sono inseriti k_0 alla volta nello shift register

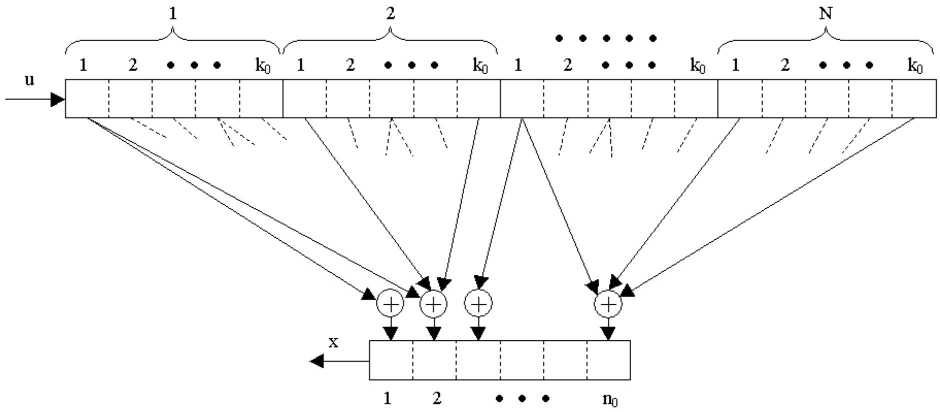

Figura 7.5. Diagramma a blocchi di un generico codificatore convoluzionale in forma seriale per un codice (n_0,k_0) con constraint length N

d'ingresso che contiene Nk_0 posizioni, mentre gli n_0 sommatori modulo 2 riempono il registro d'uscita con altrettanti bit, i quali vengono mandati in uscita. Quando un nuovo blocco di k_0 bit si presenta all'ingresso, l'ultimo viene traslato a destra e va perso. Il rate è dunque dato da $R_c = k_0/n_0$, ma gli n_0 bit generati dal codificatore dipendono dai corrispondenti k_0 bit d'ingresso e dai precedenti bit che costituiscono la memoria del codificatore ν, data quindi da:

$$\nu = k_0(N-1) \tag{7.25}$$

Un simile codice è indicato come un codice convoluzionale (n_0, k_0, N) e il parametro N, cioè il numero di blocchi di dati contenuto nel registro d'ingresso, è detto *constraint length* del codice.

Come analizzato per i codici a blocco (§ 7.2.1), anche nei codici convoluzionali è possibile descivere le relazioni tra il vettore d'ingresso u e quello d'uscita x, i quali, però, sono ora delle sequenze semi-infinite. Questo può ancora essere ottenuto mediante la matrice di generazione, le cui entrate sono a loro volta delle matrici; in particolare ne vengono utilizzate N e ciascuna di esse contiene k_0 righe e n_0 colonne. La generica sottomatrice G_i descrive le connessioni dell'i-esimo blocco di k_0 bit del registro d'ingresso con le n_0 celle del registro di uscita. La matrice di generazione così ottenuta si estende indefinitamente verso il basso e a destra ed è indicata con G_∞; segue che:

$$\bar{x} = \bar{u} G_\infty \tag{7.26}$$

Il codificatore è detto sistematico se degli n_0 bit generati per ogni blocco d'ingresso, i primi k_0 sono uguali a quelli d'informazione.

La generica struttura del codificatore rappresentata in figura 7.5 può anche essere realizzata in una struttura parallela, la quale permette di poter utilizzare più blocchi d'ingresso contemporaneamente; questa soluzione è presentata in figura 7.6. Nel caso in cui i vari N_i siano differenti, la constraint

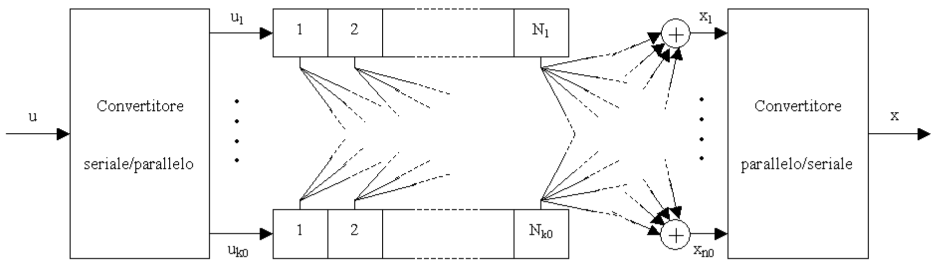

Figura 7.6. Diagramma a blocchi di un generico codificatore convoluzionale in forma parallela per un codice (n_0, k_0, N)

length è data da:

$$N \triangleq \max_i N_i \tag{7.27}$$

e, ricordando l'equazione (7.25), ora possiamo scrivere:

$$\nu = k_0 \sum_{i=1}^{k_0} (N_i - 1) \tag{7.28}$$

Si può ancora notare che, siccome normalmente il rate è pari ad 1/2 o 1/3, vi sono pochi (2 o 3) sommatori e risulta più facile descrivere le connessioni mediante la *matrice della funzione di trasferimento* G, in cui le singole posizioni $g_{i,j}$ sono dei vettori riga binari di N elementi che descrivono la connessione tra l'i-esimo ingresso ($i = 1, \ldots, k_0$) e la j-esima uscita ($j = 1, \ldots, n_0$).

Una rappresentazione dei codici convoluzionali, alternativa a quella algebrica, è fornita dai *diagrammi di stato*, basata sul fatto che il codificatore è un sistema a memoria finita e, quindi, la sua sequenza d'uscita dipende da quella d'ingresso e dallo stato del dispositivo. All'istante di tempo discreto l, lo stato del codificatore vale:

$$\sigma_l \triangleq (u_{l-1}, \ldots, u_{l-\nu}) \tag{7.29}$$

e il numero di stati è pari a:

$$N_\sigma \triangleq 2^\nu \tag{7.30}$$

Nel diagramma a stati vi sono 2^{k_0} archi entranti e altrettanti uscenti da ogni stato; ne deriva che all'aumentare della memoria esso si espande e diventa difficile ricordare i cammini passati, in quanto si passa più volte per lo stesso arco. A tale scopo è stata realizzata una rappresentazione, detta *diagramma a traliccio*, che tiene conto esplicitamente della varibile temporale. Ogni sezione verticale del traliccio rappresenta la condizione degli stati all'istante l, che ora viene chiamato *profondità* del traliccio. Ogni ramo all'interno del traliccio rappresenta una sequenza codificata e, per ogni profondità l, vi sono 2^l possibili cammini rappresentanti tutte le possibili sequenze codificate di lunghezza l.

Come per i codici a blocco, anche nei codici convoluzionali le capacità di rivelazione e correzione degli errori dipendono dalle proprietà di distanza delle sequenze codificate. A questo proposito si utilizza la *distanza libera* d_f, che rappresenta la minima distanza di Hamming tra sequenze di lunghezza infinita. Matematicamente è definita come:

$$d_f \triangleq \lim_{l \to \infty} d_c(l) \tag{7.31}$$

dove $d_c(l)$ è detta *distanza colonna* di ordine l definita come il minimo peso di Hamming tra tutte le coppie di sequenze presenti alla profondità l. La distanza d_f può essere calcolata sul traliccio valutando il minimo peso di Hamming tra le sequenze che, dopo la divergenza dalla sequenza di tutti zero[4], si ricongiunge ad essa. Il calcolo pratico della distanza libera, in realtà, viene effettuato matematicamente calcolando la *funzione enumeratrice dei pesi*:

$$T(D) \triangleq \sum_{d=d_f}^{\infty} A_d D^d \tag{7.32}$$

dove A_d è il numero di cammini di peso d che divergono da un stato e vi ritornano successivamente. Tale funzione caratterizza lo spettro delle distanze, in quanto è una serie che fornisce tutte le informazioni circa i pesi dei cammini che partono da uno stato e tornano nello stesso: il peso minore sarà la d_f. È possibile caratterizzare ancora più in dettaglio le proprietà del codice mediante la *funzione enumeratrice dei pesi d'ingresso-uscita*:

$$T_3(W,D,L) \triangleq \sum_{w=1}^{\infty} \sum_{d=d_f}^{\infty} \sum_{l=1}^{\infty} C_{w,d,l} W^w D^d L^l \tag{7.33}$$

dove $C_{w,d,l}$ è il numero di cammini, divergenti da uno stato e ricongiungenti in esso, con peso d e lunghezza l generati da una sequenza d'informazione di peso w (maggiori dettagli sono descritti in [29], pp. 461-465).

Risulta interessante valutare la complessità di un codice convoluzionale, la quale può essere espressa come il numero di archi da visitare nel diagramma a stati per bit decodificato. Ricordando l'equazione (7.30) e che ogni stato ha 2^{k_0} archi entranti e uscenti, la complessità assume l'espressione:

$$\mathcal{D} = \frac{2^{k_0+\nu}}{k_0} \tag{7.34}$$

da cui si nota come essa aumenti rapidemente con k_0; per questo motivo i codici utilizzati sono quelli con rate $1/n_0$, mentre per ottenere codici con rate k_0/n_0 si utilizza la tecnica della *punturazione*, basata sull'eliminazione di opportuni bit di parità, in modo da massimizzare la distanza libera, partendo da un codice convoluzionale con rate $1/n_0$. I codici così ottenuti presentano, di solito, uno spettro leggermente peggiore rispetto a quello dei codici

[4]Si considera di trasmettere la sequenza di tutti zero grazie alla proprietà di errore uniforme dei codici lineari, la quale afferma che la probabilità d'errore condizionata ad una parola di codice trasmessa è indipendente da quale essa sia (si veda [29], p. 447).

non punturati dello stesso rate, ma guadagnano in complessità. Il caso più comune è quello di partire con un codice convoluzionale di rate 1/2 e, per ottenerne uno di rate k_0/n_0, si considerano k_0 sezioni del traliccio iniziale; la complessità ottenuta vale:

$$\mathcal{D}_{punc} = \frac{k_0 2^{1+\nu}}{k_0} \tag{7.35}$$

cioè risulta diminuita rispetto al caso non punturato, come si evidenzia rapportando le rispettive complessità:

$$\frac{\mathcal{D}}{\mathcal{D}_{punc}} = \frac{2^{k_0}}{2k_0} \tag{7.36}$$

Nel caso dei codici a blocco, la decodifica a massima verosimiglianza è ottenuta scegliendo la parola di codice con distanza minima rispetto alla parola ricevuta; lo stesso principio può essere applicato anche per i codici convoluzionali. In questo caso, siccome ogni sequenza può essere rappresentata con un cammino all'interno del traliccio, si dovrà scegliere il percorso relativo alla sequenza che è più "simile" a quella ricevuta, cioè che ne minimizza la distanza di Hamming. Questo risultato è ottenibile applicando l'*algoritmo di Viterbi*, descritto e analizzato in [43] (pp. 97-104). La complessità di tale algoritmo, si veda l'equazione (7.34), cresce esponenzialmente con k_0 e ν e, quindi, le applicazioni pratiche si limitano a valori di $(k_0 + \nu)$ pari a qualche decina. Inoltre, la decodifica ottima richiederebbe di decodificare la sequenza solo alla fine del processo di ricezione, introducendo così dei lunghi ritardi e richiedendo delle elevate capacità di memorizzazione; si può far fronte a questo problema utilizzando una soluzione subottima che forza la decisione dopo un intervallo di tempo opportunamente prestabilito (*Viterbi troncato*). Esistono anche altre tecniche, precedenti all'algoritmo di Viterbi, per decodificare i codici convoluzionali e vengono applicate nei casi in cui si utilizzino delle constraint length elevate; una visione complessiva delle tecniche qui accennate è riportata in [43] (pp. 112-146).

Per quanto concerne le prestazioni dei codici convoluzionali, è possibile ottenere dei limiti superiori che forniscono strette approssimazioni per rapporti segnale/rumore medio-alti, mentre divergono quando tali valori sono bassi. In particolare essi dipendono dalle funzioni enumeratrici dei pesi analizzate in precedenza e si può dimostrare che, si veda [29] (pp. 476-481), utilizzando la decodifica di Viterbi e quella hard, valgono le seguenti relazioni per la probabilità d'evento d'errore[5]:

$$P(e) \leq T(D)\Big|_{D=\sqrt{4p(1-p)}} \tag{7.37}$$

e sul bit:

$$P_b(e) \leq \frac{1}{k_0}\frac{\partial T_2(W,D)}{\partial W}\Bigg|_{D=\sqrt{4p(1-p)},W=1} \tag{7.38}$$

dove la funzione $T(D)$ è data dall'equazione (7.32), mentre $T_2(W,D)$ può essere ottenuta ponendo $L = 1$ nell'equazione (7.33). Nel caso della decodifica soft, invece, la probabilità d'evento d'errore è data da:

$$P(e) \leq \frac{1}{2}\sum_{d=d_f}^{\infty} A_d erfc\left(\sqrt{\frac{dR_cE_b}{N_0}}\right) < \frac{1}{2}T(D)\Big|_{D=e^{-R_c(E_b/N_0)}} \tag{7.39}$$

dove si è applicata la relazione:

$$\frac{1}{2}erfc(\sqrt{x}) < \frac{1}{2}e^{-x} \tag{7.40}$$

Sempre facendo uso di tale equazione, per la probabilità d'errore sul bit si ottiene:

$$P_b(e) < \frac{1}{2k_0}\frac{\partial T_2(W,D)}{\partial W}\Bigg|_{D=e^{-R_c(E_b/N_0)},W=1} \tag{7.41}$$

7.4 La tecnica ARQ

La tecnica ARQ è il principale approccio alternativo alla correzione d'errore realizzata con la tecnica FEC e si basa sulla sola rivelazione degli errori con la successiva richiesta al trasmettitore della ritrasmissione del blocco errato; con questa tecnica, i pacchetti scambiati a livello superiore richiedono un'intestazione aggiuntiva per consentire la richiesta di tale ritrasmissione. L'ARQ è usato quasi universalmente per il controllo degli errori nei protocolli DLC (*Data Link Control Protocol*), cioè relativi al flusso di dati nelle comunicazioni punto-punto sullo strato di connessione dei dati (*Data Link*

[5]L'evento errore è definito come un segmento di cammino incorretto che si ricongiunge in uno stato dopo il distaccamento da esso.

Layer). Esistono numerose varianti sul protocollo ARQ, basate soprattutto sulla scelta del riconoscimento positivo (ACK, *acknowledgment*) e negativo (NAK, *negative ACK*), oppure solo positivo. Un ACK indica la corretta ricezione di una trama o, meglio ancora, che i bit di parità ricalcolati al ricevitore sono validi; un NAK, invece, può essere restituito se questo calcolo non fornisce un esito positivo. Utilizzare entrambi i tipi di conferme permette una maggiore efficienza, però l'approccio con i soli ACK permette di eliminare i messaggi superflui. La scelta di utilizzare solo i NAK non è praticabile, perchè il trasmettirore non sarebbe in grado di distinguere tra la ricezione di una trama corretta e la sua completa perdita, in quanto entrambi i casi sarebbero caratterizzati dall'assenza di risposta da parte del ricevitore. Gli schemi convenzionali ARQ sono principalmente di tipo *Stop-and-Wait* e *continuo*; quest'ultima categoria, a sua volta, si suddivide in *Go-Back-N* e *Selective-Repeat*. Nel seguito verrà fornita una breve descrizione delle strategie qui accennate (si veda [44], pp. 279-286).

7.4.1 Stop-and-Wait

Lo schema noto come Stop-and-Wait è la più semplice forma per realizzare un protocollo di tipo ARQ; in esso il trasmettitore va nello stato di *idle*, cioè resta inattivo, dopo aver spedito un pacchetto e vi rimane fino a quando non riceve una conferma mediante ACK dal ricevitore. In questo lasso di tempo il pacchetto appena trasmesso resta ancora memorizzato in un *buffer* in attesa della risposta del ricevitore. Nel caso si utilizzino sia gli ACK che i NAK, il ricevitore spedisce una conferma positiva se non ha rivelato errori nel pacchetto ricevuto, altrementi invia un NAK. Se riceve un ACK, il trasmettitore scarta il pacchetto memorizzato in precedenza, mentre se gli ritorna un NAK allora esso rispedisce il pacchetto in questione; questo processo viene ripetuto fino a quando il pacchetto non viene ricevuto correttamente e, quindi, fino a che il trasmettitore non riceve un ACK dal ricevitore. Il protocollo può funzionare anche con le sole conferme positive più un orologio indicato come *timeout*; in un sistema che adotta tale tecnica la perdita di una trama di dati originali o di un ACK comporta l'attesa da parte del trasmettitore il quale, trascorso il tempo di timeout, ritrasmette il pacchetto. Inoltre il trasmettitore non è in grado di distinguere queste situazioni da quella in cui la mancanza di risposta indica che la trama è stata ricevuta in modo errato. Se viene perso l'ACK, il ricevitore può vedersi consegnare una copia della trama già ricevuta correttamente; per poter distinguere tra il messaggio originale e la ritrasmissione vi è un numero di sequenza, di solito un bit, contenuto nell'intestazione del pacchetto. La numerazione delle trame, comunque, è necessaria anche nel caso della tecnica con entrambe le conferme positiva e negativa, a causa della possibilità di perdita degli ACK o dei NAK.

La strategia di tipo Stop-and-Wait è ovviamente inefficiente, specialmente quando i tempi di ritardo sono significamente elevati, come può succedere nella trasmissione via satellite; tuttavia presenta il grosso vantaggio di possedere una complessità minima, in quanto non è richiesta memoria al ricevitore.

7.4.2 Go-Back-N

Come già accennato in precedenza, lo schema Go-Back-N è, a differenza dello Stop-and-Wait, di tipo continuo, cioè i pacchetti sono trasmessi ininterrottamente senza aspettare alcuna conferma dal ricevitore. Essi sono numerati sequenzialmente e memorizzati al trasmettitore per averli disponibili in un possibile uso futuro. Se il ricevitore rivela un errore in un particolare pacchetto, allora esso provvede a spedire al trasmettitore un NAK insieme con il numero di sequenza del pacchetto errato; quando il NAK è ricevuto dal trasmettitore, esso potrebbe essere in procinto di trasmettere un nuovo paccheto con numero di sequenza maggiore rispetto a quello relativo al pacchetto errato. Questa situazione fa sì che il trasmettitore "torni indietro" per ritrasmettere il pacchetto errato e, successivamente, tutti quelli dopo di esso, anche se già trasmessi in precedenza. Risulta quindi chiaro come diversi pacchetti possano essere spediti più di una volta anche se essi erano stati ricevuti correttamente; questo è dovuto al fatto che il ricevitore attende la ritrasmissione del pacchetto rivelato come errato, scartando, nel mentre, tutti quelli successivi.

Fino ad ora si è considerata la trasmissione unidirezionale, ma in generale essa potrebbe essere bidirezionale. Questo fatto può essere sfruttato per evitare di mandare gli ACK con delle apposite trame di controllo. Si può usare una tecnica nota con il nome di *piggybacking*, la quale consiste nell'inviare il riscontro includendolo in un opportuno campo di controllo di una trama trasmessa nella direzione opposta. In questo caso ogni trama ha un campo contenente in numero di sequenza, ed un altro contenente il numero di sequenza dell'ultima trama ricevuta correttamente. Altra importante considerazione da tener presente è il controllo di flusso, che consiste nell'impedire che i pacchetti vengano trasmessi ad un ritmo medio più alto rispetto a quello che il ricevitore può, o vuole, accettare; in questo modo si garantisce che il ricevitore abbia sempre a disposizione della memoria in cui conservare i pacchetti ricevuti. La tecnica tipicamente usata per realizzare questo tipo di controllo è nota con il nome di *Sliding Window* (Finestra Scorrevole), la quale consiste, essenzialmente, nel porre un limite massimo al numero di pacchetti che possono essere spediti senza ricevere conferma. Questo schema ha quindi un'azione auto-regolante, infatti nel caso in cui il ricevitore invii un ACK con un ritmo compatibile a quello con cui riceve i pacchetti, il trasmettitore lavora effettivamente in modo continuo; se però il ricevitore dovesse trovarsi in ritardo, allora gli basterebbe smettere di inviare ACK per bloccare il trasmettitore.

7.4.3 Selective-Repeat

Anche questa strategia è di tipo continuo, ma risulta più efficiente rispetto allo schema Go-Back-N. Quest'ultimo, infatti, usa il canale continuamente, ma un pacchetto può essere spedito più di una volta anche nel caso in cui esso sia stato ricevuto correttamente: tale situazione è chiaramente inefficiente. La tecnica Selective-Repeat permette di rimuovere questa inefficienza ritrasmettendo in modo selettivo solo quei particolari pacchetti che sono stati ricevuti non correttamente. Per consentire tale approccio, il ricevitore deve disporre di buffer in grado di memorizzare i pacchetti ricevuti correttamente, ma con numero di sequenza maggiore rispetto a quelli errati per i quali è stata richiesta la ritrasmissione; la memorizzazione deve avvenire fino a quando il pacchetto che era errato viene ricevuto correttamente, così da poter consegnare in modo ordinato l'informazione all'utente. Per poter essere ideale, però, questo protocollo richiederebbe una quantità di memoria infinita al ricevitore, ma in pratica viene utilizzato un numero finito di buffer. Nonostante la strategia qui analizzata sia più efficiente nell'uso della capacità di trasferimento dei dati per mezzo di poche ritrasmissioni, lo schema più diffusamente realizzato è il Go-Back-N. Questa scelta è giustificata dal fatto che esso semplifica la realizzazione del collegamento, in quanto non richiede che il ricevitore riordini i pacchetti; inoltre il Selective-Repeat diventa complesso quando avvengono degli errori multipli.

7.4.4 Prestazioni della tecnica ARQ

Scopo di questa sezione è quello di arrivare ad un'espressione che permetta di calcolare qual è la *probabilità d'errore residua* sul bit d'informazione, cioè la probabilità che il bit d'informazione errato venga invece accettato come corretto dal blocco ARQ anche dopo tutte le ritrasmissioni. A tale scopo è necessario valutare la *probabilità di ritrasmissione*, che può essere espressa come:

$$P(R) = P_r(e)P_d(e) \tag{7.42}$$

dove $P_r(e)$ è la probabilità che venga ricevuta una trama con almeno un errore, mentre $P_d(e)$ fornisce la probabilità che tale evento venga rilevato, con conseguente richiesta di ritrasmisione al trasmettitore. In particolare, la ricezione di un pacchetto di n bit contenente almeno un errore è data da:

$$P_r(e) = 1 - (1 - p)^n \tag{7.43}$$

dove con p si è indicata la probabilità d'errore nel BSC (*Binary Simmetric Channel*) equivalente del canale reale; la quantità $(1 - p)^n$, quindi, rappresenta la probabilità di ricevere n bit giusti. La probabilità di rivelare errata la parola ricevuta è invece data da:

$$P_d(e) = 1 - \frac{2^k}{2^n} = 1 - 2^{-(n-k)} \tag{7.44}$$

Il termine 2^k rappresenta tutte le possibili parole appartenenti al codice, mentre 2^n indica tutte le possibili parole di n bit che è possibile ricevere; il rapporto tra queste due quantità fornisce la probabilità che la parola ricevuta appartenga al codice e, quindi, il complementare permette di rilevare l'errore in ricezione. Sostituendo le equazioni (7.43) e (7.44) nella (7.42), si ottiene quindi:

$$P(R) = [1 - (1-p)^n][1 - 2^{-(n-k)}] \tag{7.45}$$

La probabilità d'errore residua può essere espressa come:

$$P_{ARQ} = \frac{P_{n,ARQ}(e)}{k} \tag{7.46}$$

cioè dividendo per k la probabilità di accettare come corretti, anche se errati, n bit, benchè siano state effettuate tutte le ritrasmissioni; quest'ultima è stata indicata con $P_{n,ARQ}(e)$, la quale quindi rappresenta la probabilità che la trama di n bit in uscita dal blocco ARQ sia errata. La probabilità $P_{n,ARQ}(e)$ è data da:

$$P_{n,ARQ}(e) = 1 - P_{n,ARQ}(c) \tag{7.47}$$

dove con $P_{n,ARQ}(c)$ si è indicata la probabilità di fornire in uscita dal blocco ARQ n bit giusti dopo l'ARQ, la quale coincide con quella di ottenerne n giusti in ricezione dopo le ritrasmissioni. Quest'ultima è fornita dalla sommatoria delle probabilità di ricevere n bit corretti immediatamente, o dopo una ritrasmissione, o dopo due e così via, cioè:

$$\begin{aligned} P_{n,ARQ}(c) &= (1-p)^n[1 + P(R) + P(R)^2 + \ldots + P(R)^i + \ldots] \\ &= (1-p)^n \sum_{i=0}^{\infty} P(R)^i \\ &= (1-p)^n \frac{1}{1-P(R)} \end{aligned} \tag{7.48}$$

Sostituendo l'equazione (7.48) nella (7.47) si ottiene:

$$P_{n,ARQ}(e) = 1 - \frac{(1-p)^n}{1-P(R)} \tag{7.49}$$

Inserendo quest'ultima equazione nella (7.46) si ottiene l'espressione definitiva cercata:

$$P_{ARQ} = \frac{1 - \frac{(1-p)^n}{1-P(R)}}{k} \tag{7.50}$$

Bibliografia

[1] *Telecommand: Summary of Concept and Rationale*, Report CCSDS 200.0-G-6, Green Book, Issue 6, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telecommand

[2] *Telecommand, Part 1: Channel Service*, Recommendation CCSDS 201.0-B-3, Blue Book, Issue 3, Washington D.C., June, 2000, http://www.ccsds.org/all_books.html#telecommand

[3] *Telecommand, Part 2: Data Routing Service*, Recommendation CCSDS 202.0-B-2, Blue Book, Issue 2, Washington D.C., November, 1998, http://www.ccsds.org/all_books.html#telecommand

[4] *Telecommand, Part 2.1: Command Operation Procedures*, Recommendation CCSDS 202.1-B-1, Blue Book, Issue 1, Washington D.C., October, 1991, http://www.ccsds.org/all_books.html#telecommand

[5] *Telecommand, Part 3: Data Management Service*, Recommendation CCSDS 203.0-B-1, Blue Book, Issue 1, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telecommand

[6] *Telemetry: Summary of Concept and Rationale*, Report CCSDS 100.0-G-1, Green Book, Issue 1, Washington D.C., December, 1987, http://www.ccsds.org/all_books.html#telemetry

[7] *Telemetry Channel Coding*, Recommendation CCSDS 101.0-B-3, Blue Book, Issue 3, Washington D.C., May, 1992,

http://www.ccsds.org/all_books.html#telemetry

[8] *Packet Telemetry*, Recommendation CCSDS 102.0-B-2, Blue Book, Issue 2, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telemetry

[9] *Packet Telemetry Services*, Recommendation CCSDS 103.0-B-1, Blue Book, Issue 1, Washington D.C., May, 1996, http://www.ccsds.org/all_books.html#telemetry

[10] A. Lausetti, F. Filippi, *Elementi di meccanica del volo*, Levrotto & Bella, Torino, Italy, 1956

[11] R. Foschini, *Nozioni di aerodinamica e meccanica del volo*, Esculapio, Bologna, Italy, 1980

[12] A. Leich, *GPS Satellite Surveying*, John Wiley & Sons, New York, USA, 1990

[13] F. Crosilla, L. Mussio, *Il sistema di posizionamento globale satellitare GPS*, CISM, Udine, Italy, 1991

[14] T. Logsdon, *Understanding the Navstar: GPS, GIS, and IVHS*, 2nd ed., Van Nostrand Reinhold, New York, USA, 1995

[15] S. Focardi, I. Massa, U. Uguzzoni *Fisica generale: meccanica e termodinamica*, Ambrosiana, Milano, Italy, 1999

[16] R. Noyes, *Fuel cells for public utility and industrial power*, Noyes Data Corporation, Park Ridge, New Jersey, USA, 1977

[17] W. Mitchell jr., *Fuel Cells*, Accademic Press, New York, USA, 1963

[18] C. Hu, R. M. White *Solar Cells from basic to advanced systems*, McGraw Hill, New York, USA, 1983

[19] M. Sittig, *Solar cells for photovoltaic generation of elettricity: materials, devices and application*, Noyes Data Corporation, Park Ridge, New Jersey, USA, 1979

[20] T. S. Rappaport, *Wireless communications: principles and practice*, Prentice Hall International Editions, New Jersey, USA, 1996

[21] W. C. Jakes jr., *Microwave mobile communications*, John Wiley & Sons, North Andover, Massachusetts, USA, 1974

[22] L. J. Cimini jr., *Analysis and simulation of digital mobile channel using orthogonal frequency division multiplexing*, IEEE Transactions on Communications, Vol. 33, No. 7, July, 1985

[23] M. Alard, R. Lassalle, *Principles of modulation and channel coding for digital broadcasting for mobile receivers*, EBU Review, Tech. No. 224, August, 1987

[24] L. Lo Presti, F. Neri, *L'analisi dei segnali*, 2nd ed., C.L.U.T., Torino, Italy, 1992

[25] S. Benedetto, E. Biglieri, *Trasmission with wireless applications*, Plenum, 1998

[26] A. Baciotti, F. Ricci, *Analisi matematica volume I*, Liguori editore, Napoli, Italia, Ottobre, 1994

[27] J. G. Proakis, M. Salehi, *Communication systems engineering*, Prentice Hall International Editions, New Jersey, USA, 1994

[28] J. Saltz, S. B. Weinstein, *Fourier transform communication system*, presented at the Ass. Comput. Machinary Conference Computers and Communication, Pine Mountains, October, 1969

[29] S. Benedetto, E. Biglieri, V. Castellani, *Digital transmission theory*, Prentice Hall International Editions, Englewood Cliffs, New Jersey, USA, 1987

[30] F. Fagnola, G. Pistone, *Primo semestre di probabilità* , C.L.U.T., Torino, Italy, 1996

[31] P. A. Laurent, *Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulse (AMP)*, IEEE Transactions on Communications, Vol. COM-34, No. 2, February, 1986

[32] K. Kuchi, V. K. Prabhu, *Power spectral density of GMSK modulation using matrix methods*, IEEE Military Communications Conference Proceedings, Vol. 1, 1999

[33] K. Kuchi, V. K. Prabhu, *Spectral occupancy and error rate considerations of coherent GMSK*, IEEE Wireless Communications and Networking Conference, Vol. 3, 1999

[34] A. Linz, A. Hendrickson, *Efficient implementation of an I-Q GMSK modulator*, IEEE Transactions on Circuit and Systems II: Analog and Digital Signal Processing, Vol. 43, Issue 1, January, 1999

[35] K. Murota, K. Hirade, *GMSK modulation for digital mobile radio telephony*, IEEE Transactions on Communications, Vol. COM-29, No. 7, July, 1981

[36] *TOPSIM IV: user manual*, Politecnico di Torino, Torino, Italy, 1993

[37] L. Lo Presti, M. Mondin, *Design of optimal FIR raised cosine filter*,

Electronics Letters, Vol. 25, No. 7, March, 1989

[38] G. David Forney jr., *Concatenated codes*, M.I.T. Press, Cambridge, Massachusetts, USA, 1966

[39] W. W. Peterson, *Error correcting codes*, M.I.T. Press and John Wiley & Sons, New York, USA, 1961

[40] S. B. Wicker, V. K. Bhargava, *Reed-Solomon codes and their application*, IEEE Press, New York, USA, 1994

[41] R. E. Blahut, *Theory and practice of error control codes*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1983

[42] G. C. Clark jr., J. B. Cain, *Error-correction coding for digital communications*, Plenum, New York, USA, 1981

[43] A. Dholakia, *Introduction to convolutional codes with applications*, Kluwer Accademic Publishers, Norwell, Massachusetts, USA, 1994

[44] J. D. Spragins, J. L. Hammond, K. Pawlikowski *Telecomunications protocols and design*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1991

[45] K. Sigmon, *Matlab: primer*, 4th ed., CRC Press, Boca Raton, Florida, USA, 1994

[46] C. A. Balanis, *Antenna theory: analysis and design*, 2nd ed., John Wiley & Sons, New York, USA, 1997

[47] J. D. Kraus, *Antennas*, McGraw Hill, New York, USA, 1950

[48] W. L. Stutzman, G. A. Thiele *Antenna theory and design*, John Wiley & Sons, New York, USA, 1981

[49] G. Vecchi, *Appunti sull'irradiazione e le antenne*, C.L.U.T., Torino, Italy, 1998

[50] G. Monti Guarnieri, *Nozioni sui ponti radio*, IV ed., Tamburini editore, Milano, Italia, 1971

[51] L. J. Ippolito jr., *Radiowave propagation in satellite communication*, Van Nostrand Reinhold Company, New York, USA, 1986