

# Trasmissione dati in banda base

**D**OPO che un segnale analogico diviene sequenza numerica mediante campionamento (cap. 4), e la sequenza convertita in bit dalla quantizzazione (vedi § 4.3.1.1), non esistono più differenze rispetto ai dati nativamente numerici, come i documenti su di un computer. In questo capitolo si affrontano gli aspetti legati alla trasmissione di *un segnale analogico di banda base* realizzato a partire da *dati numerici* (di qualunque natura), discutendo le alternative per realizzare il segnale in accordo a requisiti sia nel tempo che in frequenza, ed arrivando ad una espressione per la probabilità di errore in presenza di rumore additivo. Sono inoltre introdotte le problematiche della equalizzazione, del controllo di errore e della sincronizzazione.

## 15.1 Trasmissione su canale numerico

Al primo capitolo (§ 1.2.2) abbiamo illustrato come tra sorgente e destinazione di una trasmissione numerica si possa idealizzare la presenza di un *canale numerico*, che racchiude i dispositivi idonei a svolgere diverse operazioni, in modo da permettere la trasmissione di informazioni numeriche mediante un segnale analogico (indicato come *segnale dati*), che viaggia su *canale analogico*. Se quest'ultimo presenta una risposta in frequenza di tipo *passa-banda*<sup>1</sup>, anche il segnale dati dovrà presentare le medesime caratteristiche frequenziali, ed al capitolo 16 sono illustrati i principi di funzionamento dei dispositivi *modem* necessari a generare tali segnali. Nel caso in cui, invece, il canale analogico sia da considerare *passa-basso*, il segnale dati viene detto *di banda base*, ed il *modem* che lo genera è indicato come *codificatore di linea*.

### 15.1.1 Trasmissione numerica di banda base

La figura 15.1 rappresenta lo schema generale di un canale numerico, con evidenziati i principali elementi che lo compongono, la funzione dei quali viene ora brevemente illustrata. L'ingresso al canale è descritto, nella sua forma più generale, come un flusso (o sequenza) di *cifre binarie* con valore logico di 0 ed 1, indicate come bit (*binary digit*), che pervengono ad una velocità (binaria) di  $f_b$  bit/secondo.

<sup>1</sup>Ovvero lascia passare solo frequenze comprese in un intervallo che non comprende l'origine.

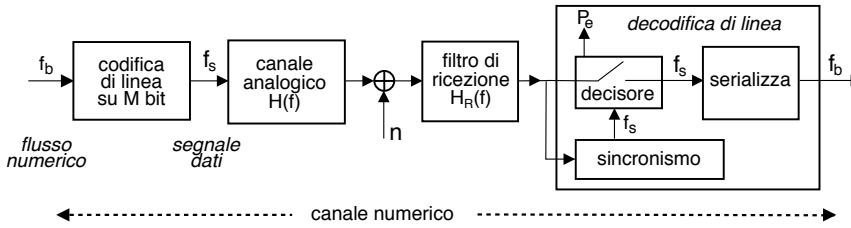
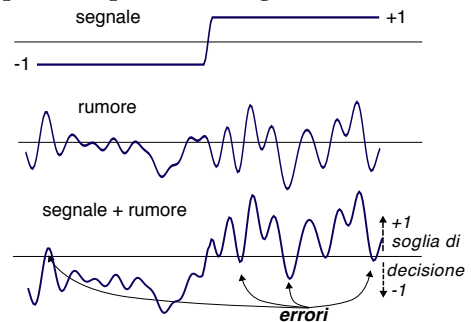


Figura 15.1: Elementi costitutivi di un canale numerico

Il primo elemento del canale numerico prende il nome di *codificatore di linea* (§ 15.1.2) e produce un *segnale dati* analogico che, ogni  $T_s$  secondi, trasporta un *simbolo*, che a sua volta rappresenta *uno o più bit*. Pertanto per la *frequenza di simbolo*<sup>2</sup>  $f_s = 1/T_s$ , si può scrivere  $f_s \leq f_b$ , con il segno di uguale nel caso di simboli *binari*. Al contrario, se ogni simbolo rappresenta  $M > 1$  bit, come vedremo al § 15.1.2.4 si ottiene un *risparmio di banda* per il segnale uscente dal codificatore, in modo da adattare lo stesso alla *banda passante* che caratterizza la risposta in frequenza  $H(f)$  del canale<sup>3</sup>. Per qualunque scelta di  $M$ , ad ogni possibile simbolo è associato ad un diverso *livello* del segnale analogico trasmesso.

L'elemento che svolge le funzioni inverse a quelle finora descritte è il *decodificatore di linea* posto al lato ricevente, che *ricostruisce* la sequenza dei simboli trasmessi a partire dal segnale analogico ricevuto, innanzitutto *campionandolo*<sup>4</sup> (con ritmo  $f_s$ ), e quindi *decidendo* quale simbolo sia stato trasmesso all'istante  $t = nT_s$ , in base al confronto tra il valore campionato, ed alcune *soglie di decisione*.

La presenza di un processo di *rumore additivo*  $n(t)$  all'uscita del canale fa sì che il segnale preso in esame dal decisore possa superare la soglia di decisione, determinando un *errore* (con probabilità  $P_e$ ) a riguardo di quale simbolo sia stato trasmesso. La valutazione di tale probabilità viene svolta al § 15.4.3, ma anticipiamo subito che  $P_e$  è tanto maggiore quanto più è grande la potenza (e dunque l'ampiezza) del rumore in ingresso al decisore, come evidente dalla figura a lato per una trasmissione *a due livelli*. Fortunatamente la potenza di rumore può essere resa



<sup>2</sup>Indichiamo  $T_s$  come *periodo di simbolo*, mentre il suo inverso  $f_s = 1/T_s$  è detto *frequenza di simbolo*, *baud-rate* o *frequenza di segnalazione*, e si misura in simboli/secondo, unità di misura indicata anche come *baud*, in memoria di ÉMILE BAUDOT, vedi [http://it.wikipedia.org/wiki/Codice\\_Baudot](http://it.wikipedia.org/wiki/Codice_Baudot).

<sup>3</sup>Se non fosse preso questo provvedimento, e si trasmettesse un segnale con una occupazione spettrale maggiore della *banda passante* del canale, nel segnale ricevuto verrebbero a mancare alcune componenti frequenziali, e di conseguenza la forma d'onda del segnale risulterebbe modificata, causando così il fenomeno di *interferenza tra simboli* (vedi § 15.1.2.2).

<sup>4</sup>Sembra giusto sottolineare che questo *campionamento* non ha lo scopo discusso al cap. 4, ma si tratta piuttosto qualcosa di più simile al filtro adattato (§ 7.6), che *decide* in base al superamento di una soglia. D'altra parte, mentre la decisione operata dal quantizzatore introduce un errore, quella del ricevitore numerico discrimina tra informazioni già *discrete*.

minima progettando adeguatamente il filtro  $H_R(f)$  di ricezione (vedi § 14.1.1), che può anche realizzare un *filtro adattato* (§ 7.6), oppure ancora un dispositivo di *equalizzazione* (§ 18.4).

Il confronto con la soglia di decisione non avviene *di continuo* come in figura, bensì agli istanti  $t = nT_s$  corrispondenti a quelli in cui sono codificati i simboli. Nello schema di fig. 15.1 è infatti presente anche un imprescindibile dispositivo di *sincronizzazione*, che osservando il segnale ricevuto<sup>5</sup> genera un *segnale di orologio* (CLOCK) che consente al decisore di operare *al passo* con il ritmo  $f_s = 1/T_s$ : al § 15.7 sono descritte alcune tecniche per affrontare questo aspetto.

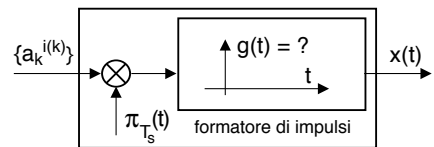
Infine, nel caso in cui ogni simbolo codifichi più di un bit (vedi la trasmissione *multilivello* al § 15.1.2.4), è necessaria la presenza di un *serializzatore* che provvede ad emettere uno dopo l'altro i bit corrispondenti a ciascun simbolo ricevuto.

### 15.1.2 Codifica di linea e segnale dati

Poniamoci nel caso più generale in cui una *sorgente discreta* con alfabeto composto da  $L$  simboli<sup>6</sup> emetta la sequenza  $\{a_k^{i(k)}\}$ , dove il pedice  $k$  individua l'istante temporale  $t = kT_s$  mentre l'indice  $i(k)$  rappresenta l'identità (con  $i$  da 1 a  $L$ ) del simbolo emesso in tale istante. Come anticipato i simboli sono prodotti con frequenza  $f_s = 1/T_s$ , ed il codificatore di linea fa corrispondere alla sequenza  $\{a_k^{i(k)}\}$  un *segnale dati*  $x(t)$  espresso come

$$x(t) = g(t) * \sum_k a_k^{i(k)} \cdot \delta(t - kT_s) = \sum_k a_k^{i(k)} \cdot g(t - kT_s) \quad (15.1)$$

a cui corrisponde lo schema simbolico mostrato a lato, in cui  $\pi_{T_s}(t) = \sum_k \delta(t - kT_s)$  è un treno di impulsi (§ 3.7) con periodo  $T_s = 1/f_s$ , schema del tutto analogo al SAMPLE&HOLD introdotto al § 4.2.4, tranne che ora  $g(t)$  è generico e prende il nome di *impulso dati*.



Generazione del SEGNALE DATI

Dato che i simboli  $a_k^{i(k)}$  sono entità casuali, il segnale  $x(t)$  risulta essere un processo aleatorio, e la valutazione della relativa densità di potenza  $\mathcal{P}_x(f)$  coinvolge il calcolo della autocorrelazione  $\mathcal{R}_x(\tau)$  e l'applicazione del teorema di Wiener (§ 7.2.1). Tale sviluppo è svolto al § 7.7.4, dove l'ipotesi aggiuntiva di simboli  $a_k$  statisticamente indipendenti ed a media nulla porta per  $\mathcal{P}_x(f)$  all'espressione

$$\mathcal{P}_x(f) = \sigma_A^2 \cdot \frac{\mathcal{E}_g(f)}{T_s} \quad (15.2)$$

in cui  $\mathcal{E}_g(f)$  è lo spettro di densità di energia di  $g(t)$  che determina la *sagomatura* della densità di potenza  $\mathcal{P}_x(f)$ , mentre  $\sigma_A^2$  è la varianza dei valori  $a_k$ <sup>7</sup> che ne determina

<sup>5</sup>Oppure mediante una seconda linea di trasmissione.

<sup>6</sup>Se  $L$  risulta essere una potenza di due ovvero  $L = 2^M$ , ogni diverso valore rappresenta un gruppo di  $M = \log_2 L$  cifre binarie (*bit*), e la trasmissione convoglia un messaggio numerico con *frequenza binaria* pari a  $f_b \left[ \frac{\text{bit}}{\text{secondo}} \right] = M \left[ \frac{\text{bit}}{\text{simbolo}} \right] \cdot f_s \left[ \frac{\text{simboli}}{\text{secondo}} \right]$ .

<sup>7</sup>La modalità generale di calcolo per  $\sigma_A^2$  viene descritta alla nota 68 di pag. 224.

il *fattore di scala*.

**Onda PAM** Il segnale dati (15.1) in alcuni contesti è indicato come *onda PAM*, acronimo di *Pulse Amplitude Modulation*, ad indicare la variazione (o modulazione) che i valori  $a_k$  determinano sugli impulsi  $g(t - mT)$  che si susseguono sull'asse temporale  $\sum_m g(t - mT)$ <sup>8</sup>. Un ulteriore punto di vista è illustrato al § 24.9.5.

**15.1.2.1 Segnale dati binario e onda rettangolare**

Partiamo dalla (15.2) per ottenere l'espressione della densità di potenza relativa ad una trasmissione *binaria*, ossia con  $f_s = f_b$  ed  $a_k = \{1, -1\}$ , con valori  $a_k$  *equiprobabili* in modo che la sequenza  $\{a_k\}$  sia a media nulla, ed adottando un impulso rettangolare  $g(t) = \text{rect}_\tau(t - \tau/2)$  con  $\tau \leq T_b$  in modo da mantenere la sua durata inferiore al periodo di simbolo. In tal caso il segnale dati  $x(t)$  assume l'aspetto mostrato in fig. 15.2 e per esso si ottiene<sup>9</sup>  $\sigma_A^2 = 1$  e  $\mathcal{E}_G(f) = \tau^2 \text{sinc}^2(f\tau)$ , dunque la (15.2) diviene

$$\mathcal{P}_x(f) = \sigma_A^2 \cdot \frac{\mathcal{E}_G(f)}{T_b} = \frac{\tau^2}{T_b} \cdot \text{sinc}^2(f\tau)$$

e presenta l'andamento<sup>10</sup> del tipo riportato sulla destra di fig. 15.2 ed espresso in scala in *decibel* (§ 8.1), per i casi  $\tau = T_b$  e  $\tau = T_b/2$ , indicati rispettivamente come *Not-Return-to-Zero* (NRZ) e RZ (vedi § 15.2.1). Osserviamo che il lobo principale di  $\mathcal{P}_x(f)$  è delimitato in  $|f| < \frac{1}{\tau}$ , ovvero  $|f| < f_b$  qualora  $\tau = T_b$ , mentre nel caso RZ il lobo principale *si dilata* fino a  $|f| < 2f_b$  mentre *si abbassa* di 6 dB<sup>11</sup>. L'occupazione di banda complessiva sarà infine approssimata ad un valore pari ad alcuni multipli dell'ampiezza del primo lobo<sup>12</sup>.

**15.1.2.2 Distorsione lineare e interferenza intersimbolica**

Qualora il canale attraversato dal segnale dati (15.1) non sia un *canale perfetto* (§ 8) ovvero presenti una risposta impulsiva  $h(t) \neq a\delta(t - \tau)$ ,  $x(t)$  subisce *distorsione*

<sup>8</sup>Infatti, se i valori  $a_k$  fossero tutti uguali, il segnale  $\sum_m g(t - mT)$  sarebbe semplicemente periodico, come descritto a pag. 81.

<sup>9</sup>Svolgendo i conti si ha

$$\sigma_A^2 = E\{a_k^2\} - (E\{a_k\})^2 = E\{a_k^2\} = \sum p_i (a_k^{(i)})^2 = \frac{1}{2} \cdot 1^2 + \frac{1}{2} \cdot (-1)^2 = 2 \cdot \frac{1}{2} \cdot 1 = 1$$

essendo  $E\{a_k\} = 0$ , mentre per quanto riguarda  $\mathcal{E}_G(f)$  si ottiene

$$\mathcal{E}_G(f) = |G(f)|^2 = |\mathcal{F}\{g(t)\}|^2 = \left| \tau \text{sinc}(f\tau) e^{-j2\pi f\tau/2} \right|^2 = \tau^2 \text{sinc}^2(f\tau)$$

<sup>10</sup>Estendiamo il risultato al caso noto di segnale periodico. Ponendo  $a_k = (-1)^k$  si genera un'onda rettangolare, il cui spettro (mancando la componente aleatoria) è *a righe*, con lo stesso involuppo di tipo  $\text{sinc}^2(fT_b)$ .

<sup>11</sup>Infatti il fattore  $\frac{\tau^2}{T_b}$  passa da  $T_b$  (NRZ) a  $\frac{T_b}{4}$  (RZ), pari ad una riduzione di 6 dB.

<sup>12</sup>

$f_b$	apparato	$T_b$	$10/T_b$
$2.4 \cdot 10^3$	Modem (anni '80)	$4.2 \cdot 10^{-3}$	24 KHz
$28.8 \cdot 10^3$	Modem (anni '90)	$3.5 \cdot 10^{-5}$	288 KHz
$10 \cdot 10^6$	Thin Ethernet (anni '90)	$10^{-7}$	100 MHz
$100 \cdot 10^6$	Fast Ethernet	$10^{-8}$	1 GHz

Nella tabella a fianco è riportata l'occupazione di banda necessaria a contenere 10 lobi di un  $\text{sinc}(fT_b) = 1/T_b \mathcal{F}\{\text{rect}_{T_b}(t)\}$ , ovvero relativa ad una trasmissione binaria a velocità  $f_b = 1/T_b$  per alcuni casi tipici *del pas-*

*sato*: osserviamo che un'onda rettangolare può *andar bene* a basse velocità di trasmissione, infatti già per 10 Msimboli/sec, velocità di una LAN, occorrono 100 MHz di banda.

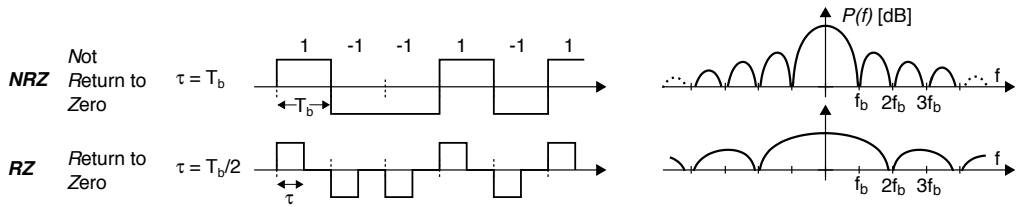
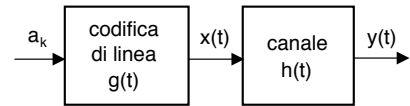


Figura 15.2: Segnale dati binario e relativa densità di potenza espressa in dB/Hz

lineare (§ 8.2), ed in uscita dal canale si presenta<sup>13</sup> il nuovo segnale dati

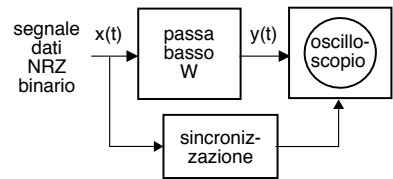
$$y(t) = \sum_k a_k \cdot \tilde{g}(t - kT_s) \quad \text{in cui} \quad \tilde{g}(t) = g(t) * h(t)$$

L'effetto della convoluzione tra  $g(t)$  ed  $h(t)$  è quello di *disperdere* nel tempo la forma d'onda  $g(t)$ , che anche se delimitata entro un periodo di simbolo come per il caso dell'onda rettangolare, arriva ad *invadere* gli intervalli temporali riservati ai simboli adiacenti, dando luogo al fenomeno della *interferenza intersimbolica* indicata anche come ISI ovvero *Inter Symbolic Interference*.



### 15.1.2.3 Diagramma ad occhio

Per valutare sperimentalmente l'effetto della distorsione lineare sul segnale dati  $x(t)$  facciamo riferimento ad uno schema di misura rappresentato a lato, che prevede di limitare la banda di  $x(t)$  ad una frequenza  $W$  e quindi visualizzare il segnale distorto  $y(t)$  ponendolo in ingresso ad un *oscilloscopio*<sup>14</sup>, la cui *base dei tempi* è sincronizzata con quella di  $x(t)$ , in modo da visualizzare sullo schermo la sovrapposizione di forme d'onda corrispondenti a coppie di simboli binari.



L'esito di tale operazione viene raffigurato in fig. 15.3, la cui parte superiore mostra il segnale dati  $x(t)$  originario realizzato in versione NRZ mediante un impulso  $g(t) = \text{rect}_{T_b}(t)$  e valori  $a_k$  binari pari a 11110101000, a fianco della sua versione filtrata a  $W = 2f_b$  (al centro) e  $W = f_b/2$  (a destra), ossia limitato in una banda  $W$  pari rispettivamente al *doppio* ed alla *metà* della *larghezza* del primo lobo di  $\mathcal{P}_x(f)$  (vedi fig. 15.2). Come evidente la limitazione di banda causa una alterazione della forma d'onda, ed il campionamento di  $x(t)$  agli istanti di simbolo  $t = kT_s$  produce valori diversi da quelli originari.

La riga inferiore di fig. 15.3 mostra invece la visualizzazione dei casi descritti da parte dell'oscilloscopio, e la disposizione risultante dei tracciati è detta *diagramma ad occhio* (traduzione di EYE DIAGRAM), termine che deriva dall'aspetto che assume il

<sup>13</sup>Posiamo infatti sviluppare le seguente uguaglianze

$$\begin{aligned} y(t) &= [\sum_k a_k \cdot g(t - kT_s)] * h(t) = [g(t) * \sum_k a_k \cdot \delta(t - kT_s)] * h(t) = \\ &= g(t) * h(t) * \sum_k a_k \cdot \delta(t - kT_s) = \tilde{g}(t) * \sum_k a_k \cdot \delta(t - kT_s) = \\ &= \sum_k a_k \cdot \tilde{g}(t - kT_s) \end{aligned}$$

<sup>14</sup>Vedi <https://it.wikipedia.org/wiki/Oscilloscopio>

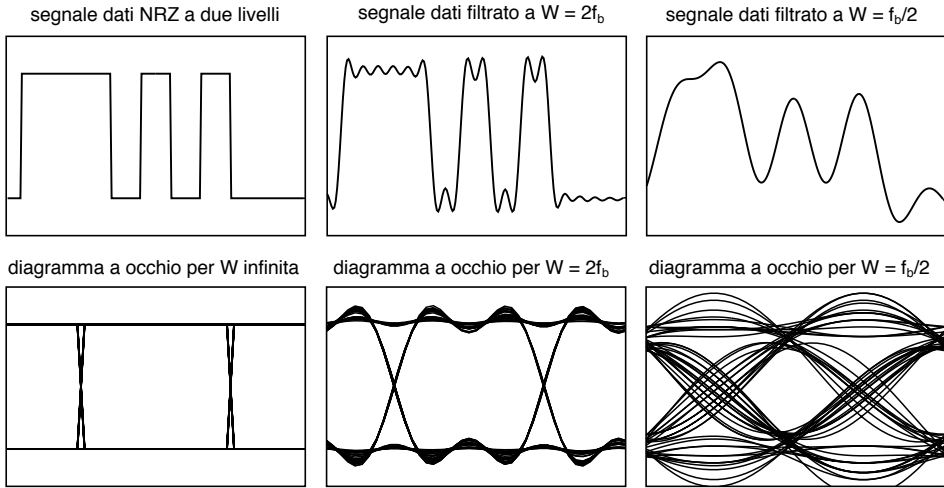


Figura 15.3: Segnale dati a banda infinita e limitata, e relativo diagramma ad occhio

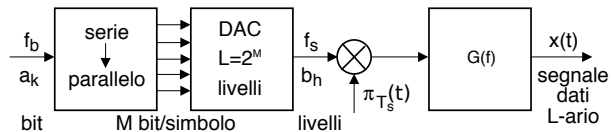
disegno che si forma, analogia che apparirà più evidente in seguito all’adozione di un impulso  $g(t)$  limitato in banda (figura 15.8), ed in presenza di rumore (figura 15.10).

Riservandoci di riprendere l’argomento nel seguito, osserviamo che il problema *non* si presenta se

- la frequenza di simbolo è molto inferiore alla banda del canale, *ovvero*
- la risposta impulsiva  $h(t)$  del canale ha una duratamolto inferiore a  $T_s$ .

### 15.1.2.4 Trasmissione multilivello

Nel caso in cui la banda a disposizione per la trasmissione *sia scarsa*, una soluzione di semplice attuazione è quella di ricorrere ad una trasmissione non più *binaria*, ma che impieghi simboli ad  $L$  valori, detti *livelli*<sup>15</sup>. A tale scopo,  $M$  simboli binari della sequenza originaria  $a_k$  sono raggruppati assieme, ed emessi *in contemporanea* da un dispositivo *serie/parallelo*<sup>16</sup> come una unica parola binaria di  $M$  bit, posta quindi in ingresso ad un convertitore D/A (pag. 97) che produce in uscita uno tra  $L = 2^M$  possibili valori, ampiezze, o livelli, che rappresentano i valori dei simboli per la nuova sequenza  $b_h$ , con  $h = k/M$ . Dato che occorrono  $T_s = MT_b$  secondi per accumulare  $M$  bit, i simboli  $b_h$  della nuova sequenza



<sup>15</sup>Proseguiamo l’esposizione riferendoci direttamente al termine *livelli*, indicando con questo la scelta tra  $L$  possibili valori di *ampiezza* per il segnale trasmesso.

<sup>16</sup>Si tratta di un componente di elettronica digitale noto come *registro a scorrimento* (vedi [https://it.wikipedia.org/wiki/Registro\\_a\\_scorrimento](https://it.wikipedia.org/wiki/Registro_a_scorrimento)), costituito da  $M$  celle di memoria di un bit, ciascuna delle quali (con frequenza  $f_b$ ) *copia* il contenuto della precedente, mentre la prima è caricata (*in serie*) con un nuovo bit; al termine di  $M$  cicli i bit vengono letti tutti assieme, appunto, *in parallelo*.

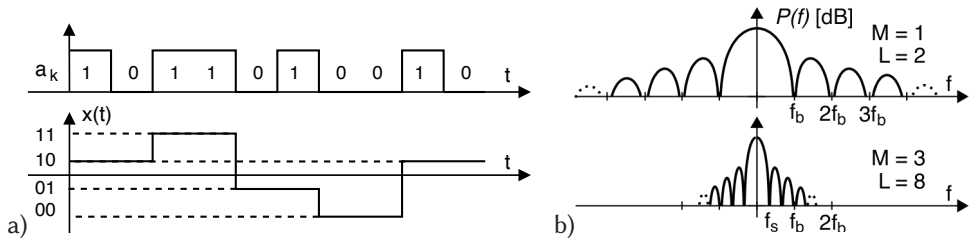


Figura 15.4: a) - codifica a quattro livelli di una sequenza binaria; b) - densità spettrale per un segnale binario ed uno ad otto livelli a parità di  $f_b$

sono prodotti ad una velocità indicata come frequenza di simbolo, pari a

$$f_s = f_b/M = 1/MT_b = 1/T_s$$

ed utilizzati per produrre un segnale dati  $x(t) = \sum_h b_h \cdot g(t - hT_s)$  ad  $L$  livelli e frequenza di simbolo  $f_s$ <sup>17</sup>.

Il risultato finale della codifica multilivello è esemplificato in figura 15.4-a), che mostra un segnale dati binario ad onda rettangolare NRZ assieme al corrispondente segnale dati  $x(t)$  ottenuto per  $L = 4$ . All'aumento del periodo di simbolo  $T_s$  corrisponde quindi l'aumento della durata di  $g(t)$ , ovvero una *contrazione* della  $G(f)$  che compare nella (15.2), determinando quindi la *riduzione* della banda occupata da  $x(t)$ , come mostrato in fig. 15.4-b), relativa all'uso di  $M = 3$  bit/simbolo ( $L = 8$ ). Pertanto l'occupazione di banda del segnale dati può essere ridotta *a piacere*, semplicemente aumentando il numero  $M$  di bit raggruppati in una singola parola.

Semberebbe tutto risolto, se non che al § 15.4 si mostra come, a meno di non aumentare la potenza del segnale dati, in presenza di rumore la codifica multilivello causi un *peggioramento* della probabilità di errore del decisore, in quanto a parità di dinamica complessiva del segnale i valori dei livelli risultano ora *ravvicinati*. Questo fenomeno è rappresentato in figura 15.5, in cui a sinistra si mostra un segnale dati ad 8 livelli, al centro il diagramma ad occhio corrispondente, ed a destra lo stesso diagramma, per un segnale filtrato a meno della sua banda.

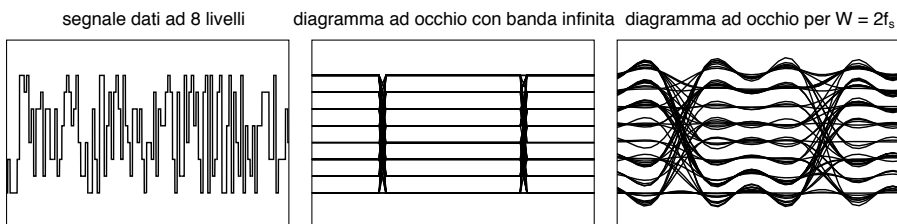


Figura 15.5: Segnale dati ad 8 livelli e diagramma ad occhio per banda infinita o limitata

<sup>17</sup>In ricezione si effettua il procedimento inverso, ripristinando la codifica binaria originaria di  $M$  bit a cui il codificatore ha associato il valore  $L$ -ario ricevuto, e quindi *serializzando* gli  $M$  bit, in modo da ri-ottenere la sequenza binaria di partenza. Vedi anche fig.15.11.

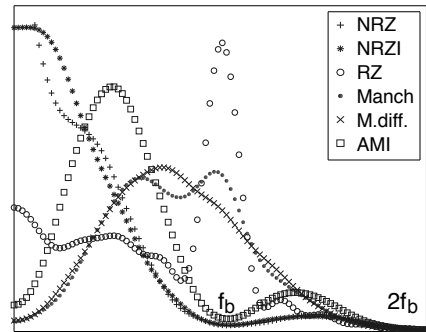
## 15.2 Scelta dell'impulso dati

Fino ad ora si è ragionato sulla base di un segnale dati generato secondo lo schema del § 15.1.2, in cui l'impulso  $g(t)$  è definito come  $g(t) = \text{rect}_\tau(t)$ , mentre nei sistemi di comunicazione questa è solamente una di diverse alternative, i cui criteri di scelta andiamo ora ad illustrare.

### 15.2.1 Codici di linea a banda infinita

Come anticipato al § 15.1.2 il grafico dello spettro di densità di potenza  $\mathcal{P}_X(f) = \sigma_A^2 \frac{\mathcal{E}_g(f)}{T_s}$  di un segnale dati dipende direttamente<sup>18</sup> da quello dello spettro di densità di energia  $\mathcal{E}_g(f)$  della risposta impulsiva  $g(t)$  usata nel formatore di impulsi, e dunque nel caso in cui  $g(t) = \text{rect}_\tau(t)$  si ottiene che  $\mathcal{P}_X(f)$  ha andamento di tipo  $\text{sinc}^2(f\tau)$  (vedi eq. (3.28) a pag. 79), che come noto si estingue come  $1/f^2$ , con il primo zero per  $f = 1/\tau$ . Nel caso in cui si operi a bassa velocità (ossia con  $\tau$  sufficientemente grande), si può considerare il canale come se fosse *a banda infinita* o perfetto, e quindi capace di riprodurre il segnale inalterato.

La figura a lato mostra lo spettro di densità di potenza  $\mathcal{P}_x(f)$  calcolato per  $f$  che va da zero fino al doppio di  $f_b$ , per un segnale dati binario che ricade in una delle categorie illustrate di seguito, dette *codici di linea*. Il risultato è ottenuto generando i valori (0 o 1) per 400 simboli binari  $a_k$  in modo pseudo-casuale<sup>19</sup>, campionando il segnale dati (15.1) con 16 campioni per periodo di bit, e valutando con questi una *stima spettrale*<sup>20</sup>.



Ogni scelta per il codice di linea a cui corrisponde una diversa definizione di impulso  $g(t)$  ha particolari proprietà, e può essere usato per trasmettere informazioni di natura binaria sotto determinate condizioni. Elenchiamo quindi caratteristiche e proprietà di tali scelte, con riferimento agli esempi riportati in figura 15.6.

**Codici unipolari** Sono realizzati come segnali *sbilanciati*,<sup>21</sup> e codificano i due livelli logici 0 ed 1 rispettivamente con un valore nullo, od un valore positivo.

<sup>18</sup>In realtà al § 7.7.4 si mostra come il risultato possa essere un po' diverso nel caso di simboli statisticamente dipendenti e/o non a media nulla.

<sup>19</sup>La non perfetta indipendenza statistica dei simboli prodotti dal generatore di numeri casuali di un computer si può riflettere su di una ridotta generalità del risultato mostrato, che tuttavia rispecchia molto bene i casi reali.

<sup>20</sup>Ottenuta applicando ai dati una finestra triangolare (§ 3.8.4) e quindi valutando il periodogramma (§ 7.3.1).

<sup>21</sup>Viene detto *sbilanciato* un segnale trasmesso mediante un collegamento (ad es. su rame) in cui uno dei due conduttori è connesso *a massa* ad entrambe le estremità, dando luogo ad una maggiore sensibilità a fenomeni di induzione elettromagnetica relativi ad altri segnali in transito nelle vicinanze (vedi *diafonia* a pag. 644), e dunque ad un peggiore SNR rispetto ai segnali (e collegamenti) bilanciati - vedi ad es. [https://it.wikipedia.org/wiki/Linea\\_bilanciata](https://it.wikipedia.org/wiki/Linea_bilanciata).

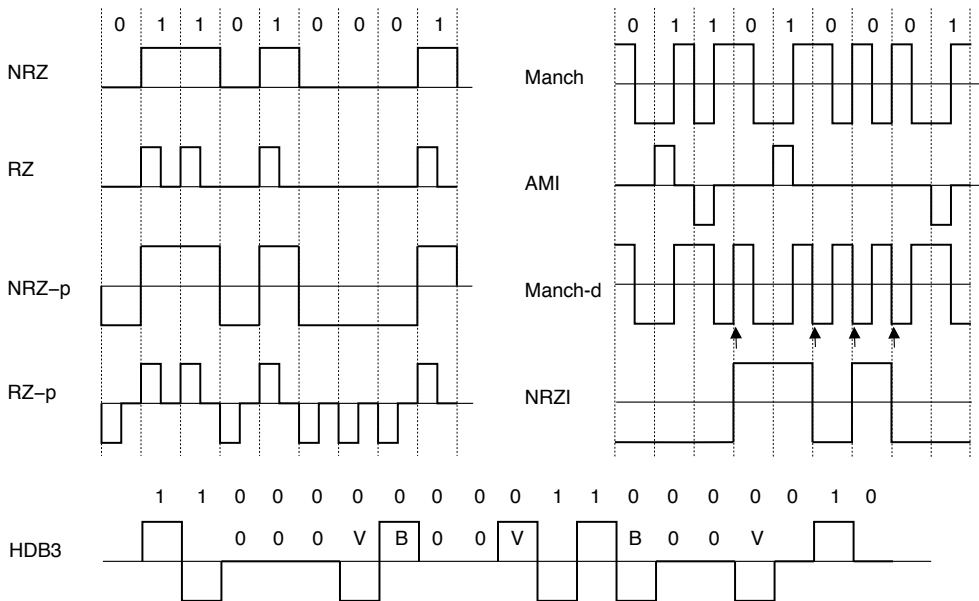


Figura 15.6: Forme d'onda di esempio associate ai codici di linea descritti nel testo

- **NRZ** o *No Return to Zero*: l'acronimo che lo descrive significa che il segnale “non torna a zero” per tutto il periodo di bit, essendo  $g(t) = \text{rect}_{T_b}(t)$ ; pertanto lo spettro  $G(f)$  è di tipo  $\text{sinc}(fT_b)$ , con il primo zero a  $f = 1/T_b$ , e presenta una componente continua. Rimane costante per dati costanti e ciò complica la sincronizzazione (§ 15.7) del clock del ricevitore stante l'assenza in questo caso di transizioni. La mancanza di energia per  $f = 1/T_b$  aggrava inoltre la situazione anche per dati qualsiasi.
- **RZ** o *Return to Zero*: in questo caso l'impulso  $g(t)$  ha durata pari a  $T_b/2$ , il segnale presenta (a parità di ampiezza) minore energia di NRZ, mentre lo spettro presenta una componente pronunciata esattamente a frequenza  $f_b$ , agevolando la sincronizzazione sul bit ma occupando una banda maggiore. Ma anche questo segnale si mantiene costante per lunghe sequenze di zeri.

**Codici bipolari** Usano segnali bilanciati o *antipodali*, e sono ricevuti mediante uno stadio di ingresso differenziale<sup>22</sup>, riducendo la sensibilità al rumore. In funzione del tipo di codice, è possibile garantire l'assenza di una componente continua nel segnale.

- **NRZ polare**, **RZ polare**: realizzano l'impulso con polarità negativa quando associato ad un bit pari a zero, e presentano media nulla solo se i valori 0 ed 1 sono equiprobabili. RZ polare non è mai costante, facilitando il compito della sincronizzazione.

<sup>22</sup>Vedi [https://it.wikipedia.org/wiki/Amplificatore\\_differenziale](https://it.wikipedia.org/wiki/Amplificatore_differenziale)

- **Manchester**<sup>23</sup>: realizza una codifica *di fase*, in quanto usa un impulso RZ a piena dinamica, in salita od in discesa, in corrispondenza dei bit 1 e 0. Per questo motivo il segnale risulta sempre a media nulla. L'occupazione spettrale è intermedia tra il caso NRZ ed RZ, dato che la durata dell'impulso può essere  $T_b$  o  $T_b/2$ . L'uso del codice Manchester è prescritto dallo standard IEEE 802.3 per le LAN a bus con contesa di accesso CDMA/CD (vedi § 23.1.4).
- **AMI** o *Alternate Mark Inversion*: codifica gli 1 con polarità alternate, mediante un impulso  $g(t)$  rettangolare di estensione  $T_b$  o  $T_b/2$ , e gli zeri con assenza di segnale, garantendo assenza di valore medio. La caratteristica di alternare valori positivi e negativi gli fa meritare il nome di codice *pseudo-ternario*, e denota la presenza di memoria. Da un punto di vista spettrale, l'AMI esibisce una occupazione di banda<sup>24</sup> ridotta rispetto a RZ, per via dei periodi silenti corrispondenti agli zeri. Se il periodo silente è prolungato, l'assenza di transizioni può compromettere la sincronizzazione di bit, e per questo motivo sono stati definiti ulteriori codici derivati, come ad esempio l'**HDB3**<sup>25</sup>.

**Codici differenziali** Sono ancora di tipo bipolare, ma la forma d'onda non è più legata al valore di un solo bit, bensì dipende da quello di due bit contigui (vedi anche § 16.4). Ciò permette di risolvere l'ambiguità che si determina qualora si scambino tra loro gli estremi del collegamento<sup>26</sup>.

- **Manchester Differenziale**: usa un impulso RZ a piena dinamica come per il Manchester, la cui polarità risulta però *invertita* rispetto all'impulso precedente se il nuovo bit è uno, mentre è mantenuta uguale nel caso arrivi uno zero (in corrispondenza delle frecce); pertanto, in presenza degli uni non si verifica transizione al confine tra i periodi di bit. Questa soluzione è utilizzata nel contesto dello standard IEEE 802.5 per LAN *Token Ring*. L'occupazione spettrale è simile a quella osservabile per la codifica Manchester.
- **NRZI**: deriva dall'NRZ, e la **I** sta per *Inverted*. Ora il livello del segnale permane nello stesso stato per i bit pari ad uno, e cambia stato per i bit pari a zero. L'assenza di valor medio è legata alla statistica che descrive le sequenze di uni e dunque non può essere garantita, mentre permangono i problemi legati alla sincronizzazione. La ridotta occupazione spettrale lo rende però interessante.

<sup>23</sup>Vedi [https://it.wikipedia.org/wiki/Codifica\\_Manchester](https://it.wikipedia.org/wiki/Codifica_Manchester)

<sup>24</sup>La densità spettrale mostrata in figura è relativa all'uso di una  $g(t)$  di tipo RZ.

<sup>25</sup>La codifica HDB3 è utilizzata per trasmettere il segnale PCM a 2 Mbps (vedi § 24.3.1), e l'acronimo significa *High-Density Bipolar-3-zeroes*. Come per AMI, rappresenta gli uni con polarità alternate, ma rimpiazza le sequenze di quattro zeri consecutivi forzando una *violazione* della regola dell'alternanza sull'ultimo bit dei quattro, in modo che il ricevitore, rilevando la violazione, è in grado di riportare il bit a zero. Dato però che la presenza della violazione creerebbe la comparsa di una componente continua nel segnale, sono inseriti anche dei bit di *bilanciamento*, per rimuovere quest'ultima. Questi si collocano al posto del primo dei quattro zeri, e la loro polarità è scelta in modo che la sequenza delle violazioni abbia una polarità alternata; in definitiva, dopo la prima violazione, si usa sempre anche il bit di bilanciamento.

<sup>26</sup>In tal caso tutti gli zeri diventerebbero uni e viceversa, mentre con la codifica differenziale questo viene evitato.

### 15.2.2 Segnale dati limitato in banda

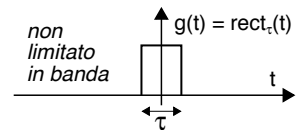
Discutiamo ora come la scelta di un impulso  $g(t)$  non rettangolare permetta di ridurre l'occupazione di banda del relativo segnali dati, indipendentemente dalla codifica multilivello.

#### 15.2.2.1 Requisiti per l'impulso di trasmissione

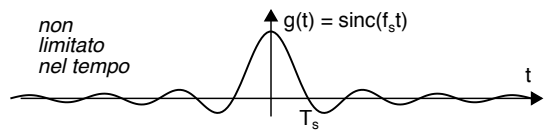
Organizziamo il ragionamento analizzando ordinatamente i requisiti che deve avere l'impulso  $g(t)$  per soddisfare *tre diverse esigenze* parzialmente contrapposte, anticipando che la soluzione sarà necessariamente una forma di compromesso.

**Limitazione di banda** Al § 15.1.2 abbiamo osservato che (sotto opportune condizioni) il segnale dati (15.1) ha densità di potenza pari a  $\mathcal{P}_x(f) = \sigma_A^2 \frac{|G(f)|^2}{T_s}$  (eq. (15.2)), e la sua ricezione inalterata è possibile solo se  $x(t)$  è trasmesso per il tramite di un *canale perfetto* (pag. 8). Se al contrario la banda del segnale eccede quella del canale gli effetti di distorsione lineare non sono trascurabili e gli impulsi  $g(t)$  *si deformano*<sup>27</sup>, causando problemi di *interferenza tra simboli* (ISI).

Ad esempio adottando  $g(t) = \text{rect}_\tau(t)$  con  $\tau \leq T_s$  il relativo spettro di ampiezza  $G(f) = \tau \text{sinc}(f\tau)$  presenta il primo passaggio per zero a frequenza  $\frac{1}{\tau} \geq \frac{1}{T_s} = f_s$ , e la densità di potenza  $\mathcal{P}_x(f)$  può essere considerata nulla solo dopo diversi multipli di tale valore.



**Limitazione nel tempo** Il problema della limitazione di banda potrebbe essere risolto adottando un impulso elementare di tipo  $g(t) = \text{sinc}(f_s t)$  che, essendo  $f_s = 1/T_s$ , ha trasformata  $G(f) = T_s \cdot \text{rect}_{f_s}(f)$  strettamente limitata nella banda

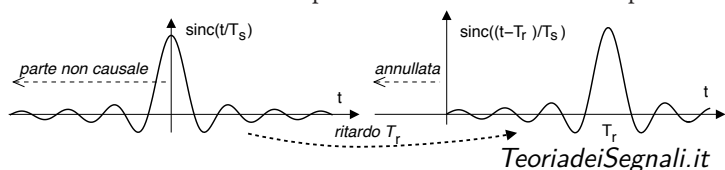


$|f| < \frac{f_s}{2}$ : in tal caso se il canale presenta un comportamento ideale in tale (limitato) intervallo di frequenze il segnale dati non subisce alterazioni. Ma lo svantaggio di adottare una forma d'onda  $g(t)$  limitata in frequenza è che la stessa è illimitata nel tempo, e dunque l'impulso può essere realizzato<sup>28</sup> solo in modo approssimato!

<sup>27</sup>Come mostrato al § 15.1.2.2, il segnale dati filtrato è basato su impulsi  $\tilde{g}(t) = g(t) * h(t)$ , con una durata pari alla somma delle durate di  $g(t)$  e  $h(t)$ . Pertanto, anche se  $g(t)$  è limitato nel tempo, come nei casi descritti al § 15.2.1, l'impulso  $\tilde{g}(t)$  si può estendere a valori di  $t > T_s$ . Considerando ad esempio la trasmissione di soli due simboli  $a_0$  ed  $a_1$ , si otterrebbe  $x(t) = a_0 \tilde{g}(t) + a_1 \tilde{g}(t - T_s)$ , e dunque  $x(T_s) = a_0 \tilde{g}(T_s) + a_1 \tilde{g}(0)$  dipenderà da entrambi i simboli anziché solamente da  $a_1$ , osservando quindi un errore pari a  $a_0 \tilde{g}(T_s)$ , detto appunto *interferenza tra simboli*.

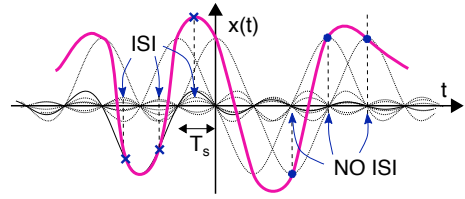
<sup>28</sup>Il requisito di causalità  $h(t) = 0$  per  $t < 0$  (pag. 26) a cui deve sottostare qualsiasi sistema fisico impedisce infatti di realizzare un filtro la cui risposta impulsiva  $g(t)$  abbia una estensione temporale illimitata: per avvicinarsi al risultato desiderato occorre implementare il filtro adottando al posto di  $g(t)$  una sua versione ritardata e limitata

$g'(t) = g(t - T_R)$   
con  $t \geq 0$  e  $g'(t) = 0$   
altrimenti, come in figura.



Notiamo ora che con questa scelta  $g(t)$  *passerebbe* da zero per  $t = nT_s$  e quindi non provocherebbe *interferenza* tra i simboli collocati agli istanti  $nT_s$ , come verificabile notando che in tal caso l'espressione (15.1) risulta del tutto simile alla (4.1) relativa alla ricostruzione *cardinale* di un segnale campionato.

**Limitazione di precisione** Contrariamente al caso del campionamento, ora *non siamo interessati* al valore del segnale negli istanti *intermedi* a quelli a cui sono centrati i simboli, e desideriamo unicamente recuperare i valori originali  $a_k$ . D'altra parte anche ammettendo di poter adottare una  $g(t) = \text{sinc}(f_s t)$  il recupero degli  $a_k$  può avvenire solamente campionando il segnale dati  $x(t)$  *esattamente* agli istanti  $t = nT_s$ , dato che al di fuori di tali istanti il valore del segnale dipende dal valore delle *code* degli impulsi  $g(t)$  centrati sugli altri simboli<sup>29</sup> come mostrato in figura, in cui la linea spessa rossa rappresenta il segnale dati  $x(t)$  risultato della somma dei contributi di termini  $a_k \text{sinc}(f_s(t - kT_s))$  con  $a_k = \pm 1$ . L'orologio (*clock*) del ricevitore deve quindi comandare il campionatore esattamente agli istanti  $kT_s$ , e non in anticipo o in ritardo, perché altrimenti si verifica ISI, tanto maggiore e da parte di tanti più simboli anche lontani, quanto più *lentamente* si attenuano le code della  $g(t)$ . Dato però che nessun oscillatore ha una precisione infinta e che anzi i metodi di sincronizzazione (§ 15.7) presentano piccole variazioni (*jitter*) della loro frequenza, occorre ricercare una soluzione per  $g(t)$  che pur rimanendo limitata in banda presenti oscillazioni di ampiezza ridotta, in modo da tollerare meglio modesti *errori di precisione* nella determinazione degli istanti di campionamento.



**Riepilogando** Vorremmo soddisfare contemporaneamente le esigenze

1. occupare una banda contenuta;
2. ricorrere ad un filtro con  $g(t)$  di durata ridotta e quindi poco complesso;
3. ridurre la sensibilità agli errori di campionamento.

Per i punti 2 e 3 sarebbe sufficiente adottare  $g(t)$  di tipo rettangolare producendo un segnale dati del tipo  $x(t) = \sum_k a_k \cdot \text{rect}_\tau(t - kT_s)$ , ma questo ha lo svantaggio di occupare una banda troppo elevata, che può essere ridotta ad un valore finito (punto 1) pur di accettare una durata per  $g(t)$  maggiore di  $T_s$ , come per il caso del *sinc*, che però ha eccessiva estensione temporale e code troppo ampie. Mostriamo ora come a partire dalla formalizzazione analitica dell'esigenza di non subire ISI sia possibile ottenere una soluzione di compromesso a tutti e tre i problemi.

Se  $T_R \gg T_s$ , l'entità dell'approssimazione è accettabile, ed equivale ad un semplice ritardo pari a  $T_R$ ; d'altro canto, quanto maggiore è la durata della risposta impulsiva, tanto più difficile (ossia costosa) risulta la realizzazione del filtro relativo.

<sup>29</sup>Al contrario, se  $g(t) = \text{rect}_{T_s}(t)$ , il campionamento può avvenire ovunque nell'ambito del periodo di simbolo, ma si torna al caso di elevata occupazione di banda.

### 15.2.2.2 Criterio di Nyquist per l'assenza di ISI

Il fatto che sia un *rect* che un *sinc* permettano di evitare interferenza intersimbolica<sup>30</sup> sembra suggerire che possano essere accomunate dal soddisfare un criterio generale: infatti, sia *rect* che *sinc* sono casi particolari di impulsi che rispettano le *condizioni* espresse nel seguito, e che alla fine ci permettono di *negoziare* il compromesso necessario a soddisfare le tre esigenze espresse sopra.

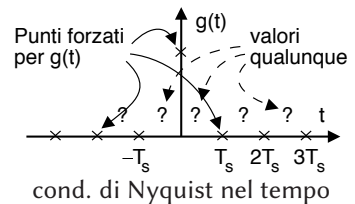
**Condizioni di Nyquist nel tempo** Torniamo a riferirci alla (15.1) per osservare che, affinché  $x(t = nT_s)$  dipenda dal solo valore  $a_n$  e non dagli altri  $a_k$  con  $k \neq n$ , deve risultare

$$g(t) = \begin{cases} 1 & \text{se } t = 0 \\ 0 & \text{se } t = mT_s \text{ con } m \neq 0 \\ \forall & \text{altrove} \end{cases} \quad (15.3)$$

e cioè  $g(t)$  deve passare da zero in tutti gli istanti multipli di  $T_s$ , tranne che per  $t = 0$  dove deve valere 1, mentre per valori di  $t$  intermedi può assumere qualunque valore. In tal caso infatti dalla (15.1) si ottiene:

$$x(nT_s) = \sum_k a_k \cdot g(nT_s - kT_s) = \sum_k a_k \cdot g((n - k)T_s) = a_n$$

dato che  $m = n - k$  è un intero che vale zero solo quando  $k = n$ . Le condizioni (15.3) prendono il nome di *condizioni di Nyquist per l'assenza di interferenza intersimbolo (ISI)* nel dominio del *tempo*. Se una forma d'onda  $g(t)$  soddisfa tali condizioni, allora viene detta *impulso di Nyquist*<sup>(31)</sup>.



**Condizioni di Nyquist in frequenza** Dalle condizioni di Nyquist *nel tempo* (15.3) se ne derivano altre *in frequenza*, mediante i seguenti passaggi. Moltiplicando  $g(t)$  per un treno di impulsi  $\pi_{T_s}(t) = \sum_k \delta(t - kT_s)$  si ottiene

$$g(t) \cdot \pi_{T_s}(t) = \delta(t)$$

dato che  $g(nT_s) = 0$  e  $g(0) = 1$ . Trasformando (vedi eq. (3.31)) si ottiene:

$$1 = G(f) * \frac{1}{T_s} \cdot \Pi_{\frac{1}{T_s}}(f) = G(f) * \frac{1}{T_s} \cdot \sum_k \delta\left(f - k\frac{1}{T_s}\right)$$

Indicando con  $f_s = \frac{1}{T_s}$  la frequenza di simbolo, ed eseguendo la convoluzione tra  $G(f)$  e gli impulsi centrati in  $f = kf_s$ , risulta infine

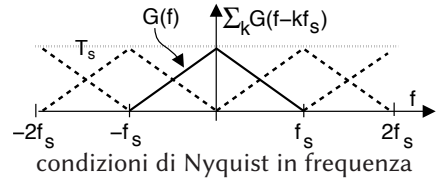
$$\sum_k G(f - kf_s) = T_s \quad (15.4)$$

che rappresenta la condizione *in frequenza* per l'assenza di interferenza intersimbolo. Il risultato ottenuto si interpreta considerando che una qualunque  $G(f)$  va bene purché, se sommata con le sue repliche traslate di multipli di  $f_s$ , dia luogo ad una costante, ovvero se  $G(f)$  manifesta *simmetria dispari* rispetto ad  $f_s/2$ . In tal caso  $G(f)$  può essere descritta come la risposta in frequenza di un *filtro di Nyquist*. Notiamo che

<sup>30</sup>Almeno, in assenza di distorsione lineare!

<sup>31</sup>Ad esempio, l'impulso rettangolare è di Nyquist, in quanto  $\text{rect}_{T_s}(t) = \begin{cases} 1 & \text{se } |t| < \frac{T_s}{2} \\ 0 & \text{se } t = kT_s \end{cases}$ .

seppure  $G(f)$  possa essere qualsiasi, anche non limitata in banda, il nostro interesse è appunto per le  $G(f)$  limitate in banda, come quella triangolare dell'esempio a fianco.



**15.2.2.3 Filtro a coseno rialzato**

Descrive una famiglia parametrica di filtri di Nyquist limitati in banda, detti a *coseno rialzato* in quanto la  $G(f)$  è realizzata mediante 2 semiperiodi di coseno raccordati da una retta, come mostrato a lato. La fig. 15.7-a)<sup>(32)</sup> illustra l'andamento di  $G(f)$  per diverse scelte del parametro  $0 < \gamma < 1$  chiamato *coefficiente di roll-off*<sup>33</sup>, che rappresenta l'indice di *dispersione* del ramo di coseno attorno alla frequenza  $f_s/2$ , detta *frequenza di Nyquist*<sup>34</sup>.

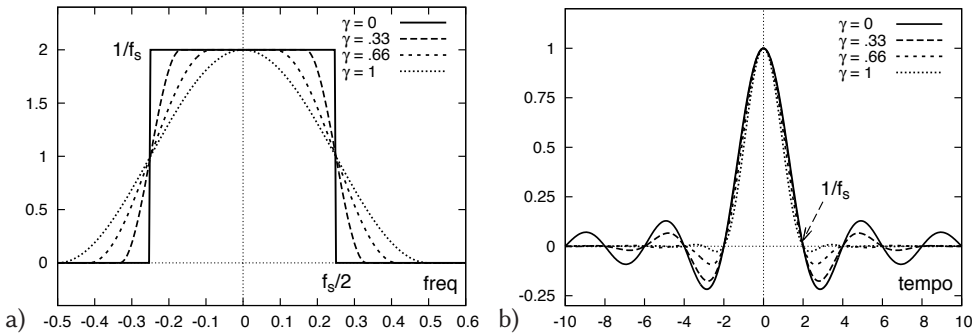
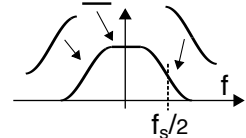
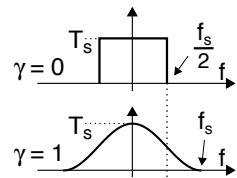


Figura 15.7: a) - filtro a coseno rialzato e b) - impulso di Nyquist per  $f_s = 0.5$ , variando  $\gamma$

La banda occupata a frequenze positive da  $G(f)$  può quindi essere espressa in funzione di  $\gamma$  come

$$B = \frac{f_s}{2} (1 + \gamma) = \frac{f_b}{2 \log_2 L} (1 + \gamma) \tag{15.5}$$

e varia da un *minimo*  $B|_{\gamma=0} = f_s/2$  in corrispondenza di un  $G(f)|_{\gamma=0} = T_s \text{rect}_{f_s}(f)$  rettangolare, ad un *massimo* pari a  $B|_{\gamma=1} = f_s$  a cui corrisponde una



$$G(f)|_{\gamma=1} = \frac{T_s}{2} \left[ 1 + \cos\left(2\pi \frac{1}{2f_s} f\right) \right] \cdot \text{rect}_{2f_s}(f) \quad \text{con } |f| < f_s$$

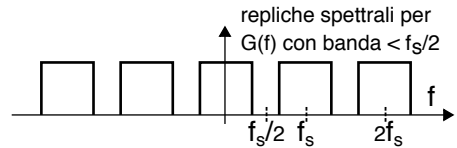
che rappresenta esattamente un periodo di coseno (in  $f$ ) di periodo  $2f_s$ , *rialzato*.

<sup>32</sup>La fig. 16.2 a pag. 494 mostra la stessa funzione su di una scala quadratica e in decibel.

<sup>33</sup>Il termine ROLL-OFF può essere tradotto come "rotola fuori".

<sup>34</sup>Molto intimamente legata alla *velocità* di Nyquist definita al § 4.1 come la *minima* frequenza di campionamento  $f_c = 2W$  per un segnale analogico di banda  $W$ , mentre la *frequenza* di Nyquist si riferisce invece a *metà della massima* frequenza di segnalazione  $f_s/2 = B$  per un segnale dati che transita su di un canale limitato in banda  $B$ , per motivi presto chiari. Come evidente due aspetti dello stesso fenomeno, ma in contesti differenti, vedi [https://en.wikipedia.org/wiki/Nyquist\\_rate](https://en.wikipedia.org/wiki/Nyquist_rate).

**Filtro a banda minima** Il caso di  $\gamma = 0$  individua una  $G(f) = T_s \text{rect}_{f_s}(f)$  e come già discusso a pag. 445 corrisponde ad un impulso  $g(t) = \text{sinc}(f_s t)$ . Tale scelta viene detta a *banda minima* poiché non è possibile occupare una banda inferiore, dato che in tal caso non sarebbero verificate le condizioni di Nyquist in frequenza, in quanto nella (15.4) resterebbero dei "buchi".



Abbiamo già osservato alla nota (28) a pagina 445 come la realizzazione di  $G(f)$  a *banda minima* sia complicata, dato che la corrispondente  $g(t) = \text{sinc}(f_s t)$  va a zero con  $t \rightarrow \infty$  come  $\frac{1}{\pi f_s t}$ , sviluppando *code* che si estendono su di un elevato numero di simboli adiacenti: oltre a complicare la realizzazione del filtro, ciò comporta la possibilità di introdurre notevole ISI in presenza di errori negli istanti di campionamento. Ma la situazione migliora decisamente usando  $\gamma > 0$ , con  $\gamma$  via via più grande, come ora illustriamo.

**Roll off  $\gamma$  diverso da zero** In questo caso per  $g(t)$  si può ottenere l'espressione generale<sup>35</sup>

$$g(t) = \text{sinc}(f_s t) \cdot \frac{\cos \gamma \pi t f_s}{1 - (2\gamma t f_s)^2} \quad (15.6)$$

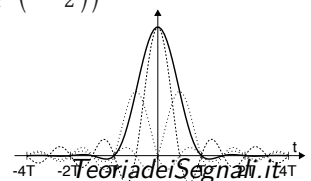
a cui corrisponde una forma d'onda *simile* a  $\text{sinc}(f_s t)$ , ma che va a zero molto più rapidamente, come verificabile osservando la parte destra di Fig. 15.7. Pertanto se  $\gamma \rightarrow 1$  le oscillazioni di  $g(t)$  sono molto più smorzate, ed anche in presenza di errori negli istanti di campionamento  $t = kT_s$  ogni impulso estende la sua *influenza* ad un numero di simboli limitrofi *molto ridotto* rispetto al caso  $\gamma = 0$ .

Per verificare visivamente quanto affermato, aiutiamoci con la fig. 15.8 che in alto a sinistra mostra l'andamento di un segnale dati realizzato adottando la  $g(t)$  fornita dalla (15.6), calcolata per  $\gamma = 0.5$ , e per simboli  $a_k$  a due valori, pari a 0 e 1. Notiamo che al di fuori degli istanti di simbolo  $t = kT_s$  il segnale può assumere valori arbitrari, anche oltre la dinamica degli  $a_k$ . La rappresentazione fornita dal diagramma ad occhio per questo segnale dati, mostrato in alto a destra in fig 15.8, permette di valutare meglio la precisione di temporizzazione che è necessaria per evitare ISI, e che è pari a metà della *apertura orizzontale* dell'occhio.

<sup>35</sup>Non ho trovato questi passaggi già svolti in nessun posto, qualche lettore può aiutare? Tutto quel che sono riuscito a calcolare è relativo al caso  $\gamma = 1$ , per cui  $g(t) = \mathcal{F}^{-1} \left\{ \frac{T}{2} [1 + \cos(\pi T f)] \right\} \cdot \text{rect}_{2/T}(f)$  che fornisce

$$\begin{aligned} g(t) &= \frac{T}{2} \left[ \delta(t) + \frac{1}{2} \delta\left(t - \frac{T}{2}\right) + \frac{1}{2} \delta\left(t + \frac{T}{2}\right) \right] * \frac{2}{T} \text{sinc}\left(\frac{2}{T}t\right) = \\ &= \text{sinc}\left(\frac{2}{T}t\right) + \frac{1}{2} \text{sinc}\left(\frac{2}{T}\left(t - \frac{T}{2}\right)\right) + \frac{1}{2} \text{sinc}\left(\frac{2}{T}\left(t + \frac{T}{2}\right)\right) \end{aligned}$$

che una volta graficato, conferma l'andamento di fig. 15.7. Osserviamo che per  $t \rightarrow 1/2\gamma f_s$  il denominatore di (15.6) si annulla, ma lo stesso avviene anche per il numeratore, che in tal caso tende a  $\cos \frac{\pi}{2}$ , dando luogo alla forma  $\frac{0}{0}$ , ed il cui limite sembra tendere a poco meno di uno.



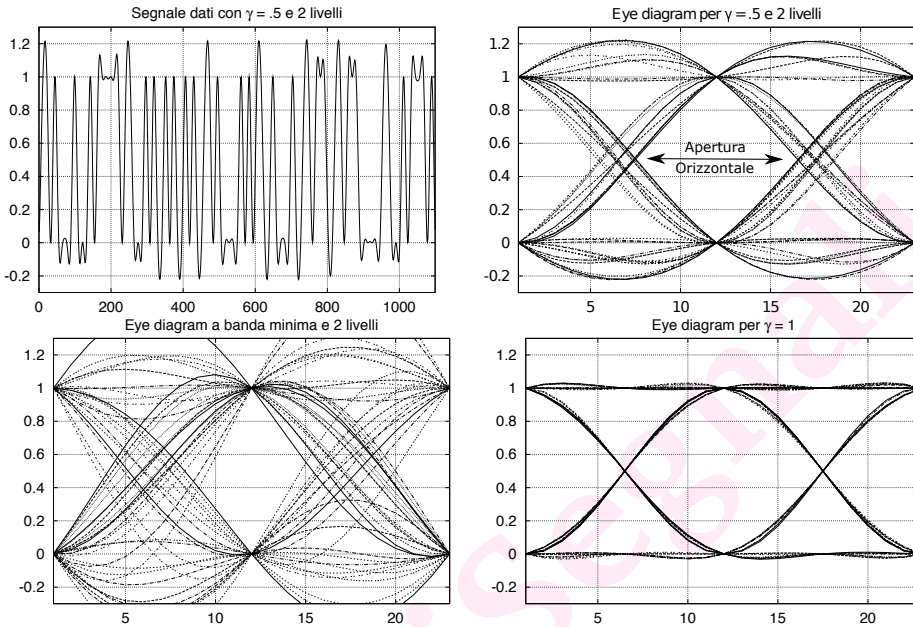


Figura 15.8: Segnale dati e *diagramma ad occhio* per diversi valori di *roll-off*

Gli ultimi due diagrammi nella parte inferiore di fig. 15.8 permettono il confronto tra le scelte relative alla banda minima e *massima*: notiamo che nel primo caso (a destra)  $g(t)$  è un *sinc*, e l'apertura orizzontale dell'occhio si è ristretta, mentre nel secondo (a sinistra, con  $\gamma = 1$ ) l'occhio è alla sua apertura massima, ed i tracciati sono quasi del tutto identici tra loro, indipendentemente dai simboli precedenti e successivi.

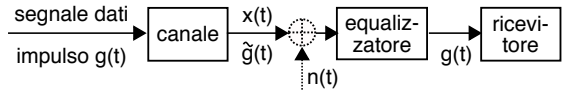
### 15.3 Equalizzazione

Torniamo ad occuparci del problema legato all'attraversamento da parte del segnale dati di un canale che presenta una  $H(f)$  *non ideale* (vedi pag. 231), subendo quindi *distorsione lineare* (§ 8.2) e causando così per il sistema di trasmissione la comparsa di ISI, dato che (vedi § 15.1.2.2) il segnale ricevuto  $x(t)$  in uscita dal canale risulta ora realizzato mediante un impulso *distorto*  $\tilde{g}(t) = g(t) * h(t)$  anziché  $g(t)$ . Fortunatamente la distorsione lineare è (almeno in linea di principio) completamente reversibile, e nel caso in cui  $H(f)$  sia nota può essere *compensata* facendo transitare  $x(t)$  attraverso un *filtro di equalizzazione*  $H_{eq}(f)$  tale che in cascata ad  $H(f)$  ripristini le condizioni di canale perfetto, cioè tale che

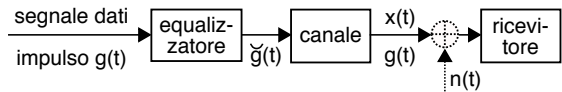
$$\begin{cases} H(f) H_{eq}(f) = ae^{-j2\pi f\tau} \\ h(t) * h_{eq}(t) = a\delta(t - \tau) \end{cases} \quad \text{e dunque} \quad H_{eq}(f) = \frac{ae^{-j2\pi f\tau}}{H(f)}$$

Come noto scrivere  $H(f) H_{eq}(f)$  oppure  $H_{eq}(f) H(f)$  è la stessa cosa, e dunque il filtro di equalizzazione può essere posto sia in trasmissione che in ricezione, con le seguenti conseguenze.

**Al ricevitore** Se il lato ricevente conosce  $H(f)$  può calcolare la  $H_{eq}(f)$  teorica e sintetizzare un filtro (vedi § 5.2) che la approssimi<sup>36</sup>. Se invece il ricevitore non la conosce,  $H(f)$  può essere stimata (producendo  $\hat{H}(f)$ ) a partire dal segnale ricevuto, facendo precedere la trasmissione vera e propria da una fase di *apprendimento*, durante la quale sono trasmessi dati che anche il ricevitore conosce, in modo da poter utilizzare il segnale di errore per stimare la distorsione lineare introdotta dal canale, ovvero la  $H(f)$ .



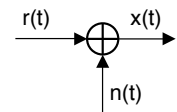
**Al trasmettitore** Il problema principale della metodica precedente è che anche il rumore presente in ingresso al canale passa attraverso il filtro di equalizzazione, trasformandosi da bianco a colorato, e questo peggiora le prestazioni. Pertanto se il trasmettitore conosce a priori la  $H(f)$  del canale, oppure questa è stimata al lato ricevente ed esiste un canale di ritorno, può essere preferibile attuare l'equalizzazione *in partenza*, utilizzando al posto dell'impulso  $g(t)$  originario un impulso definito come  $\check{g}(t) = g(t) * h_{eq}(t)$ , e tale quindi da ripresentarsi come  $g(t)$  in uscita dal canale, essendo  $h_{eq}(t) * h(t) = a\delta(t - \tau)$ .



La trattazione delle tecniche di equalizzazione prosegue al § 15.5.1 dove sono chiariti alcuni aspetti qui solamente accennati, ed al § 18.4 in cui si discutono soluzioni di tipo numerico.

## 15.4 Probabilità di errore per trasmissioni di banda base

Fin qui abbiamo trascurato di prendere in considerazione gli effetti del rumore additivo, a cui si è accennato al § 15.1.1, e che provoca la ricezione di un segnale  $x(t) = r(t) + n(t)$ . Al segnale utile  $r(t)$  risulta dunque sovrapposto un diverso segnale  $n(t)$  indicato come disturbo o rumore (*noise*<sup>37</sup>), membro di un processo ergodico (vedi § 6.3), con densità di probabilità del primo ordine *gaussiana* (vedi § 6.2.4) a media nulla, e spettro di densità di potenza *bianco*, ossia costante in frequenza.



Nel caso in cui siano presenti più cause di disturbo, anche localizzate in punti diversi del collegamento, si fa in modo (vedi § 8.4) di ricondurle tutte ad un'unica fonte di rumore (equivalente) in ingresso al decisore. Come appare dalla figura a pag. 436, l'effetto del rumore è quello di causare degli *errori* nelle decisioni sui livelli, e quindi sui simboli e sui bit ricevuti.

Sviluppiamo dunque una analisi per valutare la *probabilità* di questi errori, in funzione delle grandezze che vi contribuiscono, in modo a poter successivamente affrontare problematiche di progetto, vedi § 19.1.

<sup>36</sup>Le esigenze di mantenere basso l'ordine del filtro tentando al contempo di rispondere ai requisiti sulla fase oltre che sul modulo impediscono di ottenere una sintesi perfetta di  $H_{eq}(f)$ .

<sup>37</sup>Vedi il § 8.4.2.1 per una descrizione della sua natura fisica.

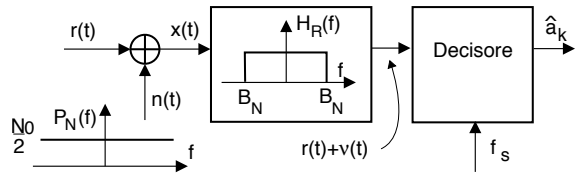
### 15.4.1 Banda di ricezione e dinamica del rumore

Come anticipato il disturbo  $n(t)$  è la realizzazione di un processo ergodico gaussiano a valor medio nullo e con spettro di densità di potenza *bianco* o costante<sup>38</sup>

$$\mathcal{P}_N(f) = N_0/2$$

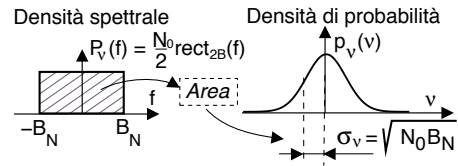
spesso indicato come *Additive White Gaussian Noise* o AWGN<sup>39</sup>.

Allo scopo di limitare la potenza del rumore alla minima possibile il ricevitore vero e proprio è preceduto da un filtro passa-basso ideale<sup>40</sup> con risposta in frequenza  $H_R(f)$  limitata in una banda  $\pm B_N$  (detta *banda di rumore*, vedi § 14.1.2), in modo da lasciar passare *per intero* le componenti frequenziali del segnale  $r(t)$  e limitare la banda del rumore  $v(t)$  in uscita da  $H_R(f)$  al minimo. Il rumore *filtrato*  $v(t)$  è anch'esso un processo gaussiano ergodico (vedi nota 36 a pag. 208) a media nulla, la cui potenza vale<sup>41</sup>



$$\mathcal{P}_v = \int_{-\infty}^{\infty} \mathcal{P}_N(f) |H_R(f)|^2 df = \int_{-B_N}^{B_N} \frac{N_0}{2} df = N_0 B_N$$

In virtù della ergodicità di  $v(t)$  il valore di  $\mathcal{P}_v$  eguaglia quello del momento di secondo ordine  $m_v^{(2)} = E\{(v)^2\}$  di una v.a.  $v$  ottenuta campionando una sua qualsiasi realizzazione; dato inoltre che  $n(t)$  e dunque  $v$  sono a media nulla, si ha<sup>42</sup>  $m_v^{(2)} = \sigma_v^2$  e quindi  $\mathcal{P}_v$  individua anche *la dinamica* dei valori della v.a. di rumore sovrapposta ai valori di segnale, come esemplificato in figura.



### 15.4.2 Dinamica del segnale e decisione a massima verosimiglianza

Proseguiamo l'analisi descrivendo il segnale ricevuto nella forma

$$r(t) = \sum_k a[k] \cdot g(t - kT_s) \tag{15.7}$$

con  $g(t)$  che è un impulso di Nyquist (15.3); si assume inoltre una perfetta sincronizzazione temporale (§ 15.7) in modo da poter considerare l'ISI assente. Gli elementi della sequenza  $\{a[k]\}$  sono v.a. discrete, i.i.d. con d.d.p. uniforme, che assumono uno tra  $L$  possibili valori  $a_i$  equispaziati in un intervallo con dinamica  $\Delta = a_L - a_1$ , in modo da poter scrivere  $a_i = a_1 + \frac{\Delta}{L-1} \cdot (i - 1)$  con  $i = 1, 2, \dots, L$ .

<sup>38</sup>Al § 8.4.2.1 si illustra come in realtà  $\mathcal{P}_N(f)$  non è costante per qualsiasi valore di  $f$  fino ad infinito, ma occupa una banda grandissima ma limitata: altrimenti, avrebbe una potenza infinita.

<sup>39</sup>I due aggettivi *White* e *Gaussian* non sono per nulla *inscindibili*, nel senso che un processo può essere gaussiano ma non bianco, o bianco ma non gaussiano!

<sup>40</sup>Al § 15.5 si descrive un diverso modo di progettare  $H_R(f)$ , in modo da minimizzare la probabilità di errore anziché la potenza di rumore, e che di fatto realizza un *filtro adattato*, descritto al § 7.6. Come vedremo al § 15.5, nel caso di un impulso *a banda minima* i due approcci portano al medesimo risultato.

<sup>41</sup>Per i dettagli relativi al filtraggio di processi, ci si può riferire al § 7.4.1.

<sup>42</sup>vedi eq. (6.9) a pag. 152

Agli istanti  $t = kT_s = k/f_s$  multipli del periodo di simbolo  $T_s$  il decisore acquisisce il valore di segnale più rumore  $x(kT_s) = (r(t) + \nu(t))|_{t=kT_s}$  ed anziché ritrovare il valore  $a[k] = a_{i(k)}$  del simbolo trasmesso, osserva la realizzazione di una v.a. gaussiana  $\check{x} = x(kT_s)$  con valor medio  $a_{i(k)}$  e varianza  $\sigma_v^2 = N_0 B_N$ , essendo  $\nu(t)$  a media nulla. Per stabilire quale valore  $a_i$  sia stato (più probabilmente) associato al simbolo  $k$ -esimo il ricevitore effettua quindi una decisione di massima verosimiglianza (o ML, vedi § 6.6.2.1) confrontando tra loro le densità di probabilità condizionate alle diverse ipotesi che sia stato trasmesso uno tra i simboli  $a_i$ :

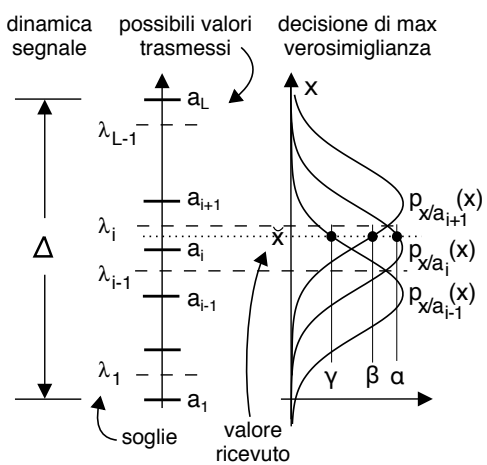
$$P_{X/a_i}(\check{x}) = \frac{1}{\sqrt{2\pi}\sigma_v} e^{-\frac{(\check{x}-a_i)^2}{2\sigma_v^2}} \quad (15.8)$$

e scegliendo per  $\hat{a}_i$  tale che  $P_{X/\hat{a}_i}(\check{x})$  è la più grande, ossia

$$\hat{a}_i = \arg \max_{a_i} \{P_{X/a_i}(\check{x})\}$$

Il criterio di massima verosimiglianza equivale pertanto (vedi figura) a definire  $L - 1$  soglie di decisione  $\lambda_i, i = 1, 2, \dots, L - 1$  poste a metà tra i valori  $a_i$  ed  $a_{i+1}$ <sup>43</sup>, e decidere per il valore  $a_i$  se il segnale ricevuto  $\check{x}$  cade all'interno dell'intervallo compreso tra  $\lambda_{i-1}$  e  $\lambda_i$ <sup>(44)</sup>, dato che (con riferimento alla notazione in figura) ciò corrisponde ad imporre

$$\alpha = P_{X/a_i}(\check{x}) > \beta = P_{X/a_{i+1}}(\check{x}) > \gamma = P_{X/a_{i-1}}(\check{x})$$

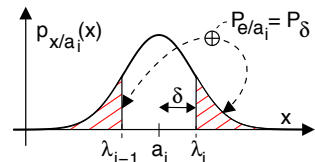


### 15.4.3 Probabilità dell'errore gaussiano

A seguito dell'applicazione del criterio di massima verosimiglianza il decisore commette errore quando, a fronte della trasmissione di un simbolo  $a_i$ , il campione di rumore filtrato  $\nu(kT_s)$  assume un valore abbastanza elevato da far oltrepassare a  $\check{x}$  una soglia di decisione, ovvero qualora  $|\nu(kT_s)| > \delta$  in cui  $\delta$  è la metà dell'intervallo tra due soglie e cioè  $\delta = |\lambda_i - a_i| = \frac{\Delta}{2(L-1)}$ . La probabilità di questo errore si dice *condizionata* alla trasmissione di  $a_i$  e vale

$$P_{e/a_i} = 2 \int_{\lambda_i}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_v} e^{-\frac{(x-a_i)^2}{2\sigma_v^2}} dx = P_\delta \quad (15.9)$$

che chiameremo  $P_\delta$ , e che rappresenta (vedi figura) la somma delle aree tratteggiate.



<sup>43</sup>La proprietà di *equidistanza* delle soglie dal valore di simboli deriva dalla *simmetria pari* della d.d.p. gaussiana rispetto al suo valor medio: in generale, le soglie sono poste in modo da rendere eguali le probabilità di *falso allarme* e di *perdita*, vedi § 6.6.1.

<sup>44</sup>Chiaramente, tutti i valori  $x$  minori di  $\lambda_1$  provocano la decisione a favore di  $a_1$ , e quelli maggiori di  $\lambda_{L-1}$  indicano la probabile trasmissione di  $a_L$ .

Lo stesso valore  $P_\delta$  è valido per tutti gli indici  $i$  compresi tra 2 ed  $L - 1$ , mentre per  $a_1$  ed  $a_L$  la probabilità di errore è dimezzata, in quanto in tali casi esiste solamente una delle due soglie il cui superamento determina una decisione errata, e dunque scriviamo  $P_{e/a_1} = P_{e/a_L} = \frac{1}{2}P_\delta$ . Applicando ora alla (15.9) il cambiamento di variabile descritto al § 6.2.4, si ottiene  $P_\delta = \operatorname{erfc} \left\{ \frac{\lambda_i - a_i}{\sqrt{2}\sigma_v} \right\}$ , ed esprimendo l'intervallo  $\lambda_i - a_i$  in funzione della dinamica di segnale  $\Delta$  troviamo

$$P_\delta = \operatorname{erfc} \left\{ \frac{\Delta}{2\sqrt{2}\sigma_v (L - 1)} \right\} \quad (15.10)$$

Per arrivare all'espressione della probabilità di errore *incondizionata*, ovvero indipendente dall'identità del simbolo trasmesso, occorre eseguire una operazione di *valore atteso* (§ 6.2.2) rispetto a tutti gli indici  $i$ , con  $i = 1, 2, \dots, L$ , cioè pesare le diverse probabilità di errore condizionate  $P_{e/a_i}$  con le rispettive probabilità  $P_i = \operatorname{Pr} \{a_i\}$  degli eventi condizionanti  $a_i$ . Avendo assunto l'ipotesi di valori  $a_i$  *equiprobabili* risulta  $P_i = \frac{1}{L}$  e quindi

$$P_e = E_{a_i} \{P_{e/a_i}\} = \sum_{i=1}^L P_i P_{e/a_i} = \frac{1}{L} \left[ (L - 2) P_\delta + 2 \frac{1}{2} P_\delta \right] = \left( 1 - \frac{1}{L} \right) P_\delta \quad (15.11)$$

in cui si è tenuto conto del diverso valore della probabilità condizionata per i livelli intermedi e per i due agli estremi.

#### 15.4.4 Parametri di sistema e di trasmissione

Il risultato ottenuto, benché già idoneo a valutare la  $P_e$  con i dati con cui è stata impostata l'analisi, deve attraversare qualche ulteriore passaggio per poter esprimere  $P_e$  in funzione dei parametri di sistema<sup>45</sup> potenza di segnale  $\mathcal{P}_R$ , densità di potenza di rumore  $\mathcal{P}_v(f) = N_0/2$  e velocità binaria  $f_b$ , nonché dei parametri di trasmissione<sup>46</sup>  $L$  e  $\gamma$ , in modo da poter affrontare gli aspetti di *bilancio di collegamento* (cap. 19). Approfondiamo a tale scopo alcune relazioni per giungere alla definizione di un nuovo parametro riassuntivo.

**Legame tra potenza del segnale  $\mathcal{P}_R$  e dinamica  $\Delta$**  Al § 15.8.1 si ottiene che, sotto le ipotesi (che manterremo valide anche nel seguito) in cui

- si adotti un impulso di Nyquist a coseno rialzato con roll-off  $\gamma$ ;
- i valori dei simboli  $a[k]$  siano statisticamente indipendenti, equiprobabili, a media nulla e distribuiti uniformemente su  $L$  livelli con dinamica  $a_L - a_1 = \Delta$ ;

la relazione tra  $\mathcal{P}_R$  e  $\Delta$  risulta<sup>47</sup>

<sup>45</sup>Sono detti *di sistema* in quanto indipendenti dalla natura della trasmissione, infatti  $\mathcal{P}_R$  dipende da amplificatori e mezzi trasmissivi,  $\mathcal{P}_v(f)$  dall'entità dei disturbi additivi presenti in uscita dal canale, mentre  $f_b$  è imposta dal contratto di servizio con il produttore di contenuti, o sorgente informativa.

<sup>46</sup>Questi sono invece parametri *negoziati* allo scopo di ottemperare ai vincoli relativi alla banda occupata ed alla precisione del temporizzatore.

<sup>47</sup>Anche se il risultato sarà dimostrato al § 15.8.1, merita comunque un commento: osserviamo che  $\mathcal{P}_R$  diminuisce all'aumentare di  $\gamma$  (si *stringe* infatti l'impulso nel tempo); inoltre  $\mathcal{P}_R$  diminuisce al crescere di  $L$ , in quanto nel caso di più di 2 livelli, la forma d'onda assume valori molto vari all'interno della dinamica di segnale, mentre con  $L = 2$  ha valori molto più *estremi*.

$$\mathcal{P}_R = \frac{\Delta^2 L + 1}{12 L - 1} \left(1 - \frac{\gamma}{4}\right) \quad (15.12)$$

Essendo il termine  $\frac{L+1}{L-1} \left(1 - \frac{\gamma}{4}\right)$  decrescente per  $L \geq 2$  e  $\gamma \geq 0$ , la potenza ricevuta assume il valore massimo  $\mathcal{P}_R = \frac{\Delta^2}{4}$  nel caso di trasmissione binaria a banda minima, ossia per  $L = 2$  e  $\gamma = 0$ . Per essere utilizzata nella (15.10), la (15.12) deve prima essere invertita in modo da esprimere  $\Delta$  in funzione di  $\mathcal{P}_R$

$$\Delta = \sqrt{12 \frac{L-1}{L+1} \frac{\mathcal{P}_R}{(1-\gamma/4)}} \quad (15.13)$$

Facciamo ora entrare in gioco anche la conoscenza di  $f_b$ , introducendo un nuova grandezza:

**Energia per bit o  $E_b$**  Dato che la potenza rappresenta l'energia sviluppata per unità di tempo, e che in un secondo entrano  $f_b$  bit, possiamo pensare  $\mathcal{P}_R$  suddivisa tra i bit presenti, in modo da definire una quantità detta *energia per bit*

$$E_b = \mathcal{P}_R T_b = \frac{\mathcal{P}_R}{f_b} \quad (15.14)$$

che riassume in sé i parametri di sistema *potenza di segnale e velocità binaria*, mentre non dipende dai *parametri di trasmissione*  $L$  e  $\gamma$ , e consente di sostituire  $\mathcal{P}_R = E_b f_b$  nella (15.13).

**Dipendenza di  $\mathcal{P}_v$  da  $L$  e  $\gamma$**  Ora nella (15.10) l'unico termine rimasto incognito sembra essere  $\sigma_v$ , pari a  $\mathcal{P}_v$  per via del valor medio nullo del rumore. D'altra parte la potenza di rumore  $\mathcal{P}_v = N_0 B_N$  dipende anche da  $L$  e  $\gamma$  attraverso la (15.5) ovvero  $B_N = \frac{f_b(1+\gamma)}{2 \log_2 L}$ , ma vorremmo mantenere separati i contributi dei parametri *di sistema* da quelli *di trasmissione*. Allora, anziché tentare di esprimere la (15.10) in funzione di  $SNR = \frac{\mathcal{P}_R}{\mathcal{P}_v}$ , introduciamo un diverso *rapporto di qualità*.

**Definizione di  $E_b/N_0$  e suo contributo all'SNR** Esprimendo le potenze  $\mathcal{P}_v$  e  $\mathcal{P}_R$  in funzione di  $T_b = 1/f_b$ , e considerando sempre un segnale dati a coseno rialzato, le eq. (15.5) e (15.14) permettono di scrivere

$$\mathcal{P}_v = N_0 B_N = \frac{N_0 (1 + \gamma)}{T_b 2 \log_2 L} \quad \text{e} \quad \mathcal{P}_R = \frac{E_b}{T_b} \quad (15.15)$$

in modo da ottenere

$$SNR = \frac{\mathcal{P}_R}{\mathcal{P}_v} = \frac{E_b T_b 2 \log_2 L}{T_b N_0 (1 + \gamma)} = \frac{E_b 2 \log_2 L}{N_0 (1 + \gamma)} \quad (15.16)$$

Quindi, mentre  $SNR$  dipende anche da  $L$  e da  $\gamma$ , il rapporto  $\frac{E_b}{N_0}$  coinvolge solo i parametri di sistema  $\mathcal{P}_R$ ,  $f_b$  ed  $N_0$ : sarà questa la variabile *indipendente* rispetto alla quale valutare la  $P_e$ .

### 15.4.5 Probabilità di errore per simbolo

Non resta ora che inserire la (15.13) nella espressione di  $P_\delta$  (eq. 15.10), ricordare che  $\sigma_v^2 = \mathcal{P}_v$ , e tenere conto della (15.16), in modo da ottenere la probabilità di decidere per un simbolo  $a_j$  diverso da quello trasmesso<sup>48</sup>:

$$P_e^{simb} = \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{3 \log_2 L}{(L^2 - 1)(1 + \gamma)(1 - \frac{\gamma}{4})}} \right\} \quad (15.17)$$

la cui dipendenza da  $\frac{E_b}{N_0}$  (espresso in dB, vedi § 8.1) è graficata alla Fig 15.9 per tre condizioni operative.

In particolare notiamo che per  $L = 2$  e  $\gamma = 0$  la (15.17) diviene

$$P_e = \frac{1}{2} \operatorname{erfc} \left\{ \sqrt{\frac{E_b}{N_0}} \right\} \quad (15.18)$$

mentre, a parità di  $E_b/N_0$ , scelte progettuali diverse da  $L = 2$  e  $\gamma = 0$  determinano immancabilmente un peggioramento della  $P_e$ : tali scelte possono essere comunque adottate per soddisfare esigenze di risparmio di banda (aumentando  $L$ )<sup>49</sup>, e per ridurre i termini di interferenza intersimbolica (aumentando  $\gamma$ ).

Due domande riassuntive:

- perché  $P_e$  peggiora se aumento i livelli? *Risposta* (<sup>50</sup>).
- perché  $P_e$  peggiora se aumento  $\gamma$ ? *Risposta* (<sup>51</sup>).

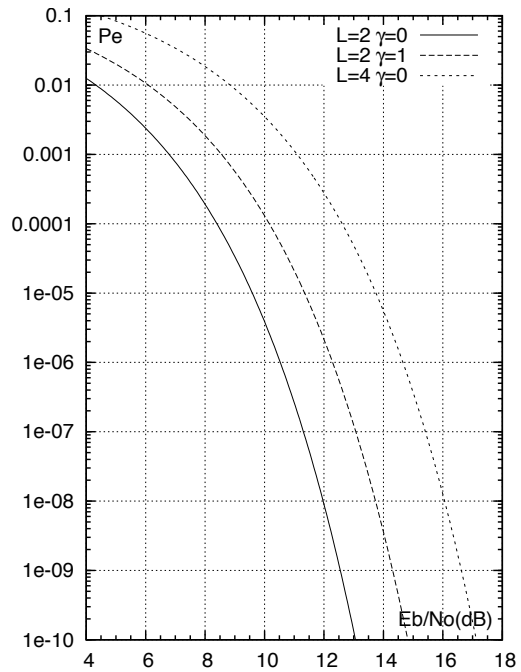


Figura 15.9: Andamento di  $P_e$  vs.  $E_b/N_0$

<sup>48</sup>Per completezza sviluppiamo i passaggi, piuttosto banali anche se non ovvi:

$$\begin{aligned} P_e &= \left(1 - \frac{1}{L}\right) P_\delta = \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \frac{\Delta}{2\sqrt{2}\sigma_v(L-1)} \right\} = \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{12 \frac{L-1}{L+1} \frac{\mathcal{P}_R}{(1-\gamma/4)} \frac{1}{2\sqrt{2}\mathcal{P}_v(L-1)}} \right\} = \\ &= \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ 2\sqrt{3 \frac{L-1}{L+1} \frac{1}{(1-\gamma/4)} \frac{1}{2\sqrt{2}} \sqrt{\frac{\mathcal{P}_R}{\mathcal{P}_v} \frac{1}{L-1}}} \right\} = \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{\frac{3}{2} \frac{L-1}{L+1} \frac{1}{(L-1)^2} \frac{1}{(1-\gamma/4)} \text{SNR}} \right\} = \\ &= \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{\frac{3}{2} \frac{1}{L^2-1} \frac{1}{(1-\gamma/4)} \frac{E_b}{N_0} \frac{2 \log_2 L}{1+\gamma}} \right\} = \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{3 \log_2 L}{(L^2-1)(1+\gamma)(1-\frac{\gamma}{4})}} \right\} \end{aligned}$$

<sup>49</sup>Aumentando  $L$  l'argomento di (15.17) diminuisce in quanto  $(L^2 - 1)$  cresce più velocemente di  $\log_2 L$ .

<sup>50</sup>Perché a parità di  $\mathcal{P}_R$  gli intervalli di decisione sono più ravvicinati, le "code" della gaussiana sottendono un'area maggiore, e questo peggioramento prevale sul miglioramento legato alla diminuzione di  $\sigma_v$  conseguente alla riduzione della banda di rumore.

<sup>51</sup>Perché occorre aumentare la banda del filtro di ricezione e dunque far entrare più rumore. D'altra parte questo peggioramento è compensato dalla riduzione dell'ISI.

### 15.4.6 Relazione con il filtro adattato

Qualche lettore può chiedersi come mai si sia utilizzato come filtro di ricezione un semplice passa basso, anziché operare come descritto al § 7.6. Tale opzione viene esplorata al § 15.5, ma possiamo notare fin da subito l'equivalenza tra i risultati (15.18) e (7.27). Infatti l'energia di un singolo impulso  $\mathcal{E}_g$  equivale all'energia per bit  $E_b$ , ed un segnale dati a media nulla e simboli binari corrisponde ad una segnalazione antipodale. Quanto all'adozione di un impulso di Nyquist a coseno rialzato con  $\gamma = 0$ , ovvero a *banda minima* (§ 15.2.2.3), ciò corrisponde ad aver posto  $G(f) = T_s \text{rect}_{T_s}(f)$ , ovvero proprio il passa basso ideale qui adottato in ricezione, che si rivela essere anche *adattato* nel caso appunto di trasmissione binaria a banda minima. Viceversa, il passa basso ideale non è più adattato qualora si scelga  $g(t)$  con  $\gamma > 0$ , e questo è il motivo della dipendenza della (15.17) dal parametro  $\gamma$ .

### 15.4.7 Compromesso banda - potenza

Osservando le fig. 15.9 e 15.12 notiamo che al crescere di  $L$ , e dunque occupando una banda minore, si può ottenere la stessa  $P_e$  solo a patto di aumentare  $E_b/N_0$ , ovvero (a parità di  $f_b$ ) aumentando la potenza trasmessa: questo è un aspetto di un risultato più generale della *teoria dell'informazione*. Si può infatti dimostrare (vedi pag. 560) che è *possibile trasmettere senza errori* (ricorrendo a tecniche di codifica di canale ottimali) purché la velocità di trasmissione  $f_b$  non ecceda la *capacità di canale*, definita come

$$C = B \log_2 \left( 1 + \frac{\mathcal{P}_R}{N_0 B} \right) \quad (15.19)$$

in cui  $B$  è la banda del canale,  $\mathcal{P}_R$  la potenza ricevuta, e  $N_0 B$  la potenza del rumore. Un secondo canale con minor *banda passante*  $B$  dispone di una minore capacità, in quanto anche se in tal caso l'argomento di  $\log_2(\cdot)$  aumenta, il logaritmo cresce più lentamente di quanto non decresca  $B$  che compare a fattore nella (15.19); pertanto per mantenere la stessa capacità è necessario trasmettere con una maggiore potenza di segnale  $\mathcal{P}_R$ . Per questo motivo qualora sussistano limitazioni di potenza ma non di banda, come ad esempio nelle *comunicazioni satellitari*, conviene occupare la maggior banda possibile, mantenendo  $L = 2$ , in modo da risparmiare potenza. L'argomento viene approfondito a pag. 561.

Coerentemente con queste osservazioni, un ulteriore aumento di banda occupata si può ottenere con l'aggiunta di bit di ridondanza, come avviene applicando le tecniche di *codifica di canale* introdotte al § 15.6 ed approfondite cap. 17, dato che a questo corrisponde un *aumento* della velocità di trasmissione complessiva. Mostreremo in tale sede come ciò consenta di *ridurre* la probabilità di errore, e dunque migliorare la *fedeltà* del flusso binario, anche a parità di potenza ricevuta.

### 15.4.8 Diagramma ad occhio in presenza di rumore

Si tratta dello stesso tipo di grafico già descritto a pag. 450, e che ora ci aiuta a valutare in modo visivo la qualità di una trasmissione numerica. In fig. 15.10 sono riportati i grafici per un segnale dati a 4 livelli, in presenza di due diversi valori per la potenza

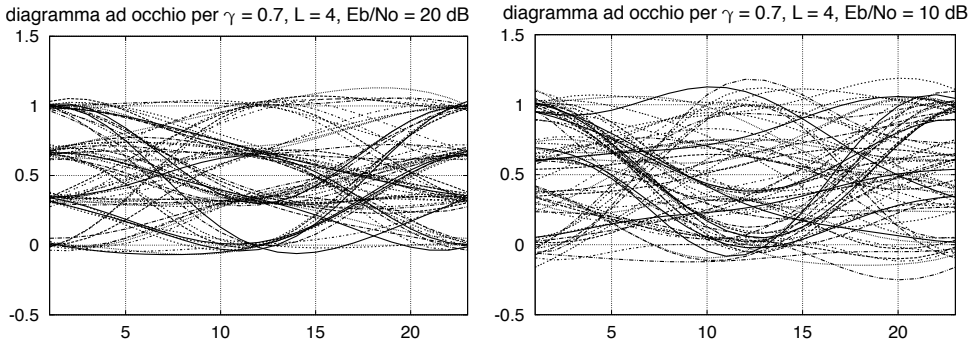
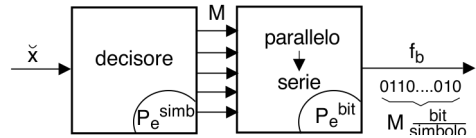


Figura 15.10: Diagramma ad occhio con  $E_b/N_0$  pari a 20 e 10 dB,  $\gamma = .7$ ,  $L = 4$

di rumore: notiamo che al peggiorare del rapporto  $\frac{E_b}{N_0}$  da 20 a 10 dB la zona priva di traiettorie (*l'occhio*) riduce la sua estensione verticale (*tende a chiudersi*). Con un tale approccio la qualità di un segnale numerico può essere valutata in modo approssimato, qualora si disponga di un oscilloscopio, esaminando il *grado di apertura dell'occhio*.

### 15.4.9 Valutazione della probabilità di errore per bit

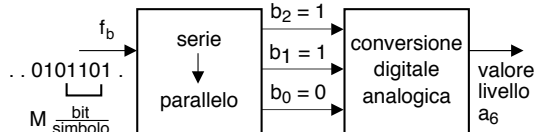
La probabilità di errore  $P_e^{symb}$  (15.17) si riferisce all'evento di decidere per la ricezione del simbolo  $a_i$  quando invece ne è stato trasmesso un altro, mentre ora intendiamo valutare la probabilità che sia errato *un qualunque bit* presente nel flusso a velocità  $f_b$ , ricostruito dopo la *serializzazione* (vedi figura a lato) della codifica binaria associata al simbolo  $a_i$  emesso dal decisore.



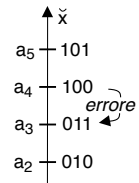
Precisiamo subito che quando la decisione per  $a_i$  è errata significa che in realtà è stato trasmesso  $a_{i-1}$  o  $a_{i+1}$  e non un altro simbolo qualsiasi, dato che la probabilità che il rumore provochi il salto di *due o più* livelli è molto inferiore a quella di *un salto singolo*. Questa circostanza ha permesso di ideare il procedimento (che ora illustriamo) di associare ad ogni simbolo (o livello) una particolare *codifica binaria*, capace di garantire la presenza di *un solo* bit errato per ogni simbolo errato.

#### 15.4.9.1 Codice di Gray

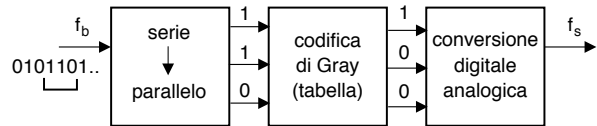
Per illustrare il problema di cui questo codice è soluzione, riprendiamo in esame la fig. 15.4-a) dove si mostra un segnale dati multilivello con  $L = 4$  per il quale i valori  $a_i$  sono associati a coppie di bit  $b_1b_0$  che sono la semplice *codifica binaria* dell'indice  $i$  corrispondente, ovvero  $i = b_12^1 + b_02^0$ : ciò significa che i valori  $a_i$  possono essere prodotti da un semplice convertitore D/A (pag. 97) alimentato da una parola di  $M$  bit  $b_{M-1} \dots b_1b_0$  per una trasmissione ad  $L = 2^M$  livelli, come mostrato a lato per  $M = 3$ .



Prendiamo quindi in esame la situazione (per  $M = 3$ ) mostrata a lato, e consideriamo ad esempio di trasmettere il livello  $a_4$  a cui è associata la codifica 100, e che il decisore a causa del rumore commetta l'errore di ritenere di aver ricevuto il livello  $a_3$ , associato alla sequenza 011: in tal caso avremmo sbagliato tutti e tre i bit!



Per evitare di osservare un numero di bit errati che dipende dal simbolo trasmesso e dal segno del rumore, la conversione D/A viene fatta precedere da una  *riscrittura* della parola di  $M$  bit attuata consultando una tabella dove è memorizzato il  *codice di Gray*.



Possiamo immaginare l'operazione come quella di un accesso a una  *memoria associativa*<sup>52</sup>, in cui la parola originaria costituisce la  *chiave* con cui individuare la parola  *codificata* da utilizzare al suo posto, come rappresentato nella tabella a lato per  $M = 3$ . Notiamo che la colonna di sinistra ha la proprietà di codificare righe adiacenti mediante configurazioni binarie che differiscono tra loro  *in una sola posizione*, ovvero per un solo bit. Per analizzare la conseguenza di ciò, osserviamo che ora al posto della parola 110 di ingresso (quarta riga) si usa il codice 100 a cui il DAC fa corrispondere il livello  $a_4$ , lo stesso dell'esempio precedente.

Ingresso	Codifica
100	111
101	110
111	101
110	100
010	011
011	010
001	001
000	000

**Decodifica** In ricezione si attua la trasformazione inversa che utilizza la tabella  *al contrario*, individuando nella seconda colonna la riga in cui compare la  *codifica binaria* associata al livello ricevuto, e sostituendo ad essa la parola nella prima colonna. In assenza di errori si riottiene la parola binaria originale; se invece si verifica un errore, ovvero ad es. come prima al posto di  $a_4$  si decide per  $a_3$  (011), il  *decodificatore* di Gray al lato ricevente individua tale  *chiave* alla 5<sup>a</sup> riga della seconda colonna, a cui fa corrispondere la sequenza 010 che trova alla prima colonna, e che infatti differisce dall'originale (110) per un solo bit (il primo).

In presenza di un errore sul simbolo il procedimento illustrato produce  *sempre* un solo bit errato. Ciò comporta che con  $M$  bit a simbolo la probabilità di osservare un bit errato si riduce di  $M$  volte rispetto a quella di errore sul simbolo, ossia risulta  $P_e^{bit} = P_e^{simb} / M$ , dato che

$$P_e^{bit} = \frac{\text{n.bit errati}}{\text{n.bit totali}} = \frac{\text{n.simboli errati}}{M \cdot \text{n.simboli}} = P_e^{simb} \frac{1}{M} \quad (15.20)$$

**Esempio** Con  $L = 256$  livelli ovvero  $M = 8$  bit/simbolo la  $P_e$  sul bit si riduce di  $\log_2 L = 8$  volte.

<sup>52</sup>Mentre in un  *array* gli elementi sono individuati in base alla loro posizione od  *indice*, una memoria associativa  *non è ordinata* e restituisce l'elemento  *associato* alla chiave, come ad esempio colore[banana]=giallo.

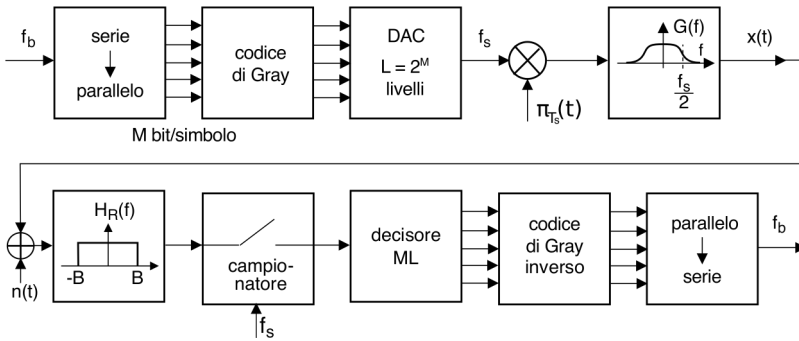


Figura 15.11: Co-decodifica di linea per un segnale dati multilivello a coseno rialzato e codifica di Gray

**Riassumendo** La figura 15.11 mostra l'intera sequenza di operazioni necessarie a generare un segnale dati multilivello, con codifica di Gray ed impulso a coseno rialzato, e quindi riceverlo recuperando la sequenza trasmessa. Ricordiamo che mentre al flusso binario di ingresso compete una velocità di  $f_b$  bit/secondo, la sequenza multilivello possiede invece un ritmo di  $f_s = \frac{f_b}{M} = \frac{f_b}{\log_2 L}$  simboli/secondo, ed il segnale dati risultante  $x(t)$  occupa una banda a frequenze positive  $B = \frac{f_s(1+\gamma)}{2} = \frac{f_b(1+\gamma)}{2\log_2 L}$ , vedi eq. (15.5).

#### 15.4.9.2 Probabilità di errore per bit

Alla luce dell'evidente vantaggio di ottenere un solo bit errato per ogni simbolo errato il codice di Gray discusso al § precedente viene adottato in modo *sistematico*, e la (15.20) può essere letta "l'evento di errore *sul bit* si verifica quando il simbolo a cui appartiene è errato, e il bit è quello errato, ovvero  $Pr\{\text{bit errato}\} = Pr\{\text{simbolo errato}\} \cdot Pr\{\text{bit errato/simbolo errato}\} = P_e^{simb} \cdot \frac{1}{\log_2 L}$ ". L'espressione (15.17) della  $P_e$  per bit nel caso si adottino una codifica di Gray diviene quindi

$$P_e^{bit} = \frac{1}{\log_2 L} \left(1 - \frac{1}{L}\right) \operatorname{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{3 \log_2 L}{(L^2 - 1)(1 + \gamma)(1 - \frac{\gamma}{4})}} \right\} \quad (15.21)$$

Le curve in fig. 15.12 mostrano il valore di  $P_e^{bit}$  così determinato, per  $\gamma = 0$ , in funzione di  $\frac{E_b}{N_0}$  espresso in dB, per diversi valori di  $L$ . Valori di  $\gamma \neq 0$  equivalgono ad un peggioramento<sup>53</sup> per  $\frac{E_b}{N_0}|_{dB}$  pari a  $10 \log_{10} (1 + \gamma)(1 - \frac{\gamma}{4})$ , o detto in altri termini, conseguono la stessa  $P_e^{bit}$  del caso  $\gamma = 0$ , a patto di incrementare  $\frac{E_b}{N_0}|_{dB}$  della stessa quantità<sup>54</sup>.

**Dimensionamento di una trasmissione numerica** Una tipica metodologia operativa di progetto può basarsi sull'imporre un determinato valore di  $P_e^{bit}$ , una volta

<sup>53</sup>Infatti con  $\gamma > 0$  l'argomento di  $\operatorname{erfc}\{\cdot\}$  si riduce. Ma non di molto: per  $\gamma = 1$  il peggioramento risulta di 1.76 dB.

<sup>54</sup>Una volta scelto un valore per  $L$  è individuata la curva da usare in fig. 15.12, ed una volta imposta una  $P_e$  sulle ordinate l' $\frac{E_b}{N_0}|_{dB}$  necessario a conseguire tale  $P_e$  con  $\gamma = 0$  si ottiene sulle ascisse seguendo la curva. Aumentando  $\gamma$  l'argomento di  $\operatorname{erfc}\{\cdot\}$  nella 15.21 si riduce, e ciò equivale a spostarsi verso sinistra sull'asse delle ascisse della stessa quantità di dB, a cui corrisponde (seguendo la curva) un aumento della  $P_e$ . Per ristabilire la  $P_e$  desiderata non resta quindi altro da fare che aumentare  $\frac{E_b}{N_0}|_{dB}$  dello stesso numero di dB.

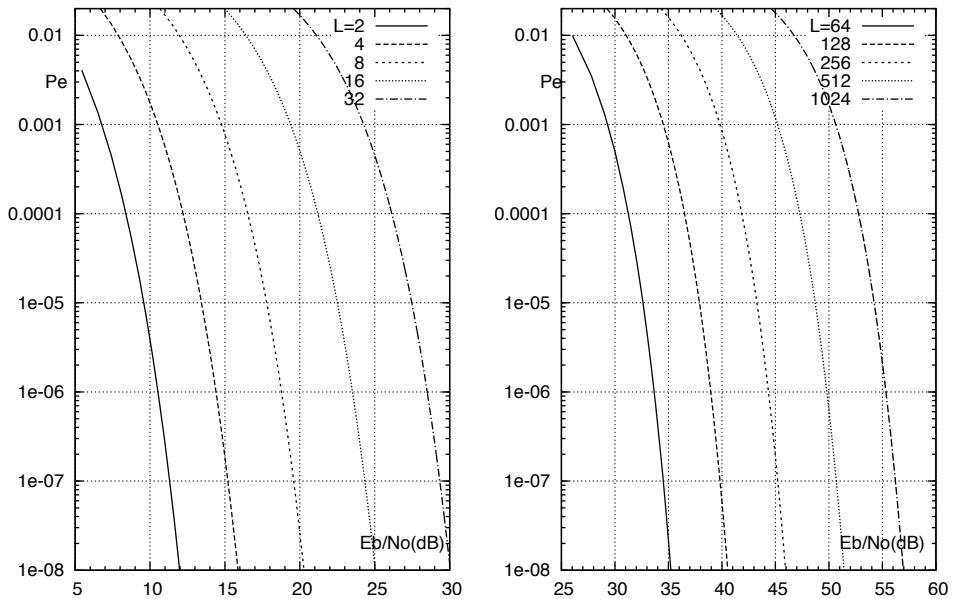


Figura 15.12: Probabilità di errore sul bit per trasmissione multilivello a banda minima e con codifica di Gray

nota la banda disponibile  $B$  e la velocità  $f_b$  richiesta. In tal caso

- in base a  $B$  e  $f_b$  si può determinare il valore di  $L$  mediante la (15.5), nell'ipotesi di adottare  $\gamma = 0$ ;
- in base alle curve di fig. 15.12 ed al valore di  $L$  individuato, si determinano i valori di  $E_b/N_0$  (in dB) necessari per ottenere la  $P_e^{bit}$ ;
- noto il livello di rumore  $N_0$ , si determina  $E_b$ ;
- note le esigenze di precisione nella temporizzazione, si impone un valore del roll-off  $\gamma$ , e conseguentemente si aumenta il valore di  $E_b$ ;
- si determina la minima potenza che è necessario ricevere, come  $W_{R_{min}} = E_b \cdot f_b$ .

**Esempio** Un canale analogico con banda a frequenze positive  $B = 500$  KHz è utilizzato per realizzare la trasmissione numerica di un flusso binario a velocità  $f_b = 10$  Mbps adottando una codifica di linea multilivello con codice di Gray ed impulso a banda minima. Al punto di ricezione è presente un rumore gaussiano bianco a media nulla e densità di potenza  $\mathcal{P}_n(f) = 10^{-12} \frac{W}{Hz}$ , la cui potenza è limitata dal ricevitore mediante un filtro passa basso con la medesima banda del canale. Desiderando una  $P_e \leq 10^{-5}$ , determinare la potenza di segnale che è necessario ricevere.

**Svolgimento** Per prima cosa determiniamo il numero di livelli: sapendo che  $B = \frac{f_b}{2 \log_2 L}$  si ottiene  $\log_2 L = \frac{f_b}{2B} = 0.5 \cdot 10^7 \frac{1}{5 \cdot 10^5} = 10$ , e dunque  $L = 2^{10} = 1024$  livelli. Dalle curve  $P_e(E_b/N_0)$  otteniamo quindi che per avere  $P_e = 10^{-5}$  con 1024 livelli occorre un  $E_b/N_0|_{dB} \geq 54$  dB, ossia  $E_b/N_0 \geq 10^{5.4}$ . Osservando infine che  $N_0 = 2\mathcal{P}_n(f)$  si ottiene la potenza del segnale come  $\mathcal{P}_x = E_b \cdot f_b = \frac{E_b}{N_0} \cdot N_0 \cdot f_b = 10^{5.4} \cdot 2 \cdot 10^{-12} \cdot 10 \cdot 10^6 = 2 \cdot 10^{0.4} = 5.2$  Watt.

A pag. 486 viene proposto un diverso esercizio che comprende anche alcuni concetti introdotti alla sezione 15.6.

### 15.5 Ricevitore ottimo

In questa sezione rimettiamo in discussione i risultati ottenuti ai §§ 15.2.2.3 e 15.4.5. Infatti come è stato illustrato al § 7.6 in relazione al *filtro adattato* in presenza di rumore bianco, il valore di  $SNR$  al punto di decisione è *massimo* se si usa un filtro di ricezione  $h_R(t)$  *adattato* alla forma dell'impulso trasmesso  $g(t) = h_T(t)$ , ovvero (a meno di traslazioni temporali) per il quale risulti  $H_R(f) = G^*(f)$ . Al contrario, nello schema adottato per la figura a pag. 452 il filtro di ricezione ha l'unico scopo di limitare la banda del rumore, ed è sempre un passa-basso ideale, indipendentemente dalla scelta fatta per  $g(t)$ . In tal caso, se si adotta una  $G(f)$  di Nyquist *non* a banda minima, i campioni di rumore (sovrapposti a quelli di segnale) danno luogo a v.a.  $x(kT_s)$  gaussiane ma *non più indipendenti*<sup>55</sup>, e quindi la  $P_e$  che si ottiene *non* è la minima possibile<sup>56</sup>.

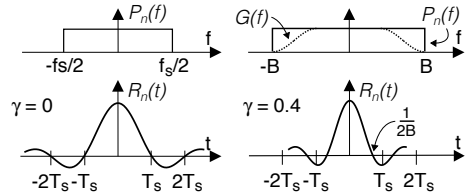
Per rendere incorrelati i campioni di rumore e ridurre la  $P_e$  al minimo, realizzando al contempo le condizioni di Nyquist in ricezione, tentiamo di verificare anche le condizioni  $H_R(f) = H_T^*(f)$  di filtro adattato, decomponendo il filtro a coseno rialzato  $G(f)$  in parti uguali tra trasmettitore e ricevitore e dando quindi luogo allo schema di figura 15.13, in cui

$$H_T(f) = H_R(f) = \sqrt{G(f)}$$

In tal modo al decisore giunge esattamente lo stesso segnale di prima<sup>57</sup>, mentre la

<sup>55</sup>Infatti il segnale  $n(t)$  uscente da  $H_R(f) = \text{rect}_{2B}(f)$  ha autocorrelazione  $\mathcal{R}_N(\tau) = \mathcal{F}^{-1}\{|H_R(f)|^2\} = 2B \text{sinc}(2B\tau)$  (vedi § 7.2.4), che passa da zero per  $\tau = \frac{1}{2B}$ .

Se si utilizza una  $G(f)$  a coseno rialzato con  $\gamma > 0$  occorre estendere la banda di ricezione a  $B = \frac{f_s}{2}(1 + \gamma)$ , a cui corrispondono campioni di rumore incorrelati se prelevati a distanza multipla di  $\tau = \frac{1}{2B} = \frac{1}{f_s(1+\gamma)}$ , mentre invece il segnale è campionato con frequenza pari a quella di simbolo  $f_s$ , e dunque con campioni a distanza  $\tau = T_s = \frac{1}{f_s}$ . Pertanto i campioni di rumore



sono correlati, con autocorrelazione pari a  $\mathcal{R}_N(T_s) = 2B \text{sinc}(1 + \gamma)$ .

<sup>56</sup>Al § 6.5.1 si dimostra come delle v.a. gaussiane incorrelate siano anche statisticamente *indipendenti*, mentre nel nostro caso i campioni di rumore sono correlati, e statisticamente dipendenti. In accordo alla trattazione della *regressione* (§ 7.7.1) e della *predizione lineare* (§ 10.1.2.2), osserviamo che la *dipendenza statistica* tra campioni di rumore implica la possibilità di ridurre l'incertezza relativa ai nuovi valori a partire dalla conoscenza dei valori passati. Il *vero* valore di un campione di *rumore* può essere calcolato sottraendo al valore  $s_{k-1}$  del segnale ricevuto all'istante di simbolo  $k - 1$ , il valore del simbolo *deciso* senza commettere errore; da questo risultato è possibile *predire* il successivo campione di rumore come  $\hat{n}_k = n_{k-1} \frac{\mathcal{R}_N(T_s)}{\mathcal{R}_N(0)}$ , che viene quindi sottratto al successivo valore  $s_k$  osservato. In tal modo, anche se la regressione non è *esatta*, l'ampiezza (e la varianza) del rumore residuo sono comunque ridotte, ed altrettanto la probabilità di errore del decisore.

<sup>57</sup>Infatti quando  $G(f)$  è tutta al trasmettitore il segnale generato (e ricevuto) ha espressione (15.7) (vedi anche la (15.1)); indicando ora  $g^\vee(t) = \mathcal{F}^{-1}\{\sqrt{G(f)}\}$ , ed eseguendo un calcolo del tutto analogo a quello svolto in § 15.1.2.2, si ottiene che il segnale ricevuto nel caso di scomposizione di  $G(f)$  ha espressione

$$r(t) = h_T(t) * h_R(t) * \sum_k a_k \cdot \delta(t - kT_s) = \sum_k a [k] \cdot g(t - kT_s)$$

in quanto  $h_T(t) * h_R(t) = g^\vee(t) * g^\vee(t) = g(t)$  per la proprietà di prodotto in frequenza.

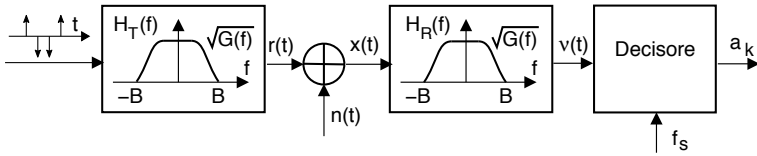


Figura 15.13: Ricevitore ottimo con impulso a radice di coseno rialzato

densità di potenza del rumore a valle di  $H_R(f)$  non è più costante, ma ora vale

$$\mathcal{P}_v(f) = \frac{N_0}{2} |H_R(f)|^2 = \frac{N_0}{2} G(f) \quad (15.22)$$

Pertanto i campioni di rumore presi a distanza  $T_s$  sono incorrelati (e quindi statisticamente indipendenti perché gaussiani, vedi § 6.5.1) in quanto  $\mathcal{R}_v(\tau) = \mathcal{F}^{-1}\{\mathcal{P}_v(f)\}$  è ora un impulso di Nyquist, che passa da zero per  $\tau = kT_s$ . Notiamo che, essendo  $G(f)$  reale pari, la fattorizzazione di  $G(f)$  realizza effettivamente la condizione  $H_R(f) = H_T^*(f)$  che definisce un filtro adattato.

**Prestazioni** Per ottenere risultati comparabili con quelli ottenuti per  $H_R(f) = \text{rect}_{f_s(1+\gamma)}(f)$  consideriamo un filtro a coseno rialzato  $G(f)$  per il quale  $\max |G(f)| = 1$  (anziché  $T_s$  come al § 15.2.2.3), e notiamo che mentre la banda passante di  $H_R(f)$  (e dunque del rumore) si è mantenuta pari a  $B = \frac{f_s}{2}(1+\gamma)$ , la potenza del rumore ora vale<sup>58</sup>

$$\mathcal{P}_v = \int_{-\infty}^{\infty} \mathcal{P}_v(f) df = \frac{N_0}{2} \int_{-\infty}^{\infty} G(f) df = \frac{N_0}{2} f_s = \frac{N_0}{2T_b \log_2 L} \quad (15.23)$$

riducendosi di un fattore  $(1+\gamma)$  se confrontata con (15.15), e causando un aumento equivalente per l'SNR; lo stesso fattore  $(1+\gamma)$  è quindi rimosso anche nella (15.21), portando a

$$P_e^{bit} = \frac{1}{\log_2 L} \left(1 - \frac{1}{L}\right) \text{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{3 \log_2 L}{(L^2 - 1) \left(1 - \frac{\gamma}{4}\right)}} \right\} \quad (15.24)$$

il valore della probabilità di errore sul bit ottenuta adottando il ricevitore ottimo ed il codice di Gray. Dato che al massimo  $1+\gamma = 2$ , questo corrisponde ad un *miglioramento massimo* di 3 dB nel valore di  $E_b/N_0$ , permettendo di usare ancora le curve di fig. 15.12. D'altra parte, il fatto che per  $\gamma = 0$  la (15.24) coincida con la (15.21) non è un risultato inatteso: infatti, se  $\gamma = 0$  si attua una trasmissione *a banda minima*, e dunque un  $H_R(f)$  rettangolare passabasso realizza esattamente un filtro adattato!

**Conseguenze** Notiamo che l'adozione di un filtro di trasmissione  $H_T(f) = \sqrt{G(f)}$  comporta che ora nel segnale trasmesso è presente ISI, che può essere rimossa solo mediante filtraggio dello stesso attraverso il filtro adattato  $H_R(f) = \sqrt{G(f)}$ . La figura 15.14 mostra poi l'andamento di  $g^\vee(t) = \mathcal{F}^{-1}\{\sqrt{G(f)}\}$  a confronto con una  $g(t)$  a coseno rialzato, per valori di roll-off pari a 0.5 ed 1, ottenuta mediante IFFT della corrispondente risposta in frequenza di modulo unitario nell'origine. Si può notare

<sup>58</sup>Il risultato si può ottenere visivamente, a partire dalla  $G(f)$  a coseno rialzato mostrata in fig. 15.7 a pag. 448 ma con altezza 1, e in base alle sue proprietà di simmetria attorno a  $\pm f_s/2$ : il risultato dell'integrale  $\int_{-\infty}^{\infty} G(f) df$  è quindi pari all'area di un rettangolo di altezza 1 e base  $f_s = 1/T_s$ .

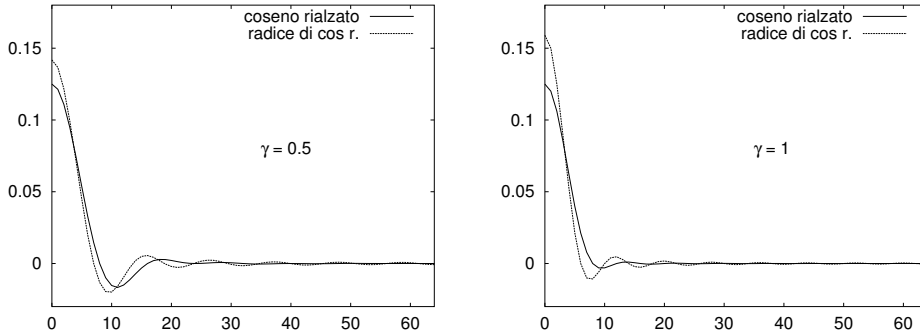


Figura 15.14: Confronto della risposta impulsiva del filtro ottimo e subottimo

un aumento sia della durata che dell'ampiezza delle oscillazioni: questa circostanza determina una maggiore complessità realizzativa del filtro di trasmissione, che deve avere una risposta impulsiva più lunga<sup>59</sup>.

### 15.5.1 Equalizzazione del ricevitore ottimo

Ulteriori considerazioni possono essere svolte qualora il canale trasmissivo presenti una risposta in frequenza  $H(f)$  non ideale, rendendo necessaria l'adozione di un filtro di equalizzazione che in accordo con quanto discusso al § 15.3 può essere semplicemente inglobato in quello di trasmissione, sintetizzando lo stesso come  $H_T(f) = \sqrt{G(f)}/H(f)$ . In tal caso la risposta in frequenza complessiva<sup>60</sup>  $H_T(f)H(f)H_R(f) = G(f)$  torna ad essere quella di un filtro di Nyquist, il segnale al punto di decisione è esente da ISI, e la densità di potenza del rumore sovrapposto è ancora espressa dalla (15.22), in modo che i campioni della componente di rumore negli istanti di simbolo sono tuttora incorrelati, e la probabilità di errore *dovrebbe essere* la minima possibile.

Tuttavia la presenza del canale distorcente e dell'equalizzatore determinano, a parità di altre condizioni (potenza di segnale, velocità binaria e livello di rumore), un peggioramento della  $P_e$  causato dalla riduzione del rapporto  $E_b/N_0$  da utilizzare nella (15.24) di una quantità  $\alpha_{dB}$ , che al § 15.8.2 è valutata in

$$\alpha_{dB} = 10 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|^2} df \quad dB \quad (15.25)$$

**Equalizzazione distribuita** Una soluzione alternativa al problema dell'equalizzazione è quella di suddividerne il compito in parti uguali sia al lato di trasmissione che a quello di ricezione, realizzando  $H_T(f) = H_R(f) = \sqrt{G(f)}/H(f)$ : anche così la risposta in frequenza complessiva  $H_T(f)H(f)H_R(f)$  è pari a quella di un filtro a coseno rialzato<sup>61</sup>, ma stavolta il rumore al punto di decisione ha densità spettrale

<sup>59</sup>Per una analisi degli effetti della limitazione temporale dell'impulso  $g^\vee(t)$ , vedere il contributo disponibile presso <https://engineering.purdue.edu/~ee538/SquareRootRaisedCosine.pdf>.

<sup>60</sup>A meno di un contributo di fase lineare  $e^{j2\pi f\tau}$  necessario a garantire la causalità dell'insieme.

<sup>61</sup>Anche in questo caso, a meno di un termine di fase lineare, che viene ommesso per non appesantire la notazione.

$\mathcal{P}_y(f) = \frac{N_0}{2} |H_R(f)|^2 = \frac{N_0}{2} \frac{G(f)}{|H(f)|}$  e dunque i campioni di rumore agli istanti di decisione sono correlati, vedi nota 55. Cionostante, al § 15.8.2.1 si mostra che la scelta di suddividere l'equalizzazione è migliore: in questo caso il peggioramento di  $E_b/N_0$  ammonta a

$$\beta_{dB} = 20 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|} df$$

e risulta essere  $\beta_{dB} \leq \alpha_{dB}$ , con l'uguaglianza solo se  $|H(f)| = \text{costante}$ , nel qual caso  $\alpha_{dB} = \beta_{dB} = 0$  dB. Per una discussione del risultato, si rimanda al § 15.8.2.1.

**Esercizio** Una trasmissione numerica binaria antipodale con potenza  $\mathcal{P}_x = 1$  Volt<sup>2</sup> e velocità  $f_b = 10$  Mbps adotta un impulso a coseno rialzato con roll-off  $\gamma = 1$ , mentre la densità di potenza del rumore bianco in ingresso al ricevitore è pari a  $\mathcal{P}_n(f) = 0.5 \cdot 10^{-8}$ . Determinare la probabilità di errore qualora

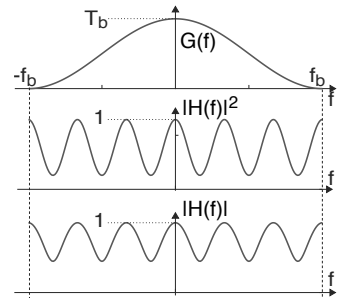
1. il ricevitore non sia ottimizzato ed il canale sia perfetto;
2. il filtro a coseno rialzato sia ripartito tra trasmettitore e ricevitore;
3. sia presente un canale con  $|H(f)|^2 = 0.4 \cdot \cos(2\pi 3 \frac{f}{f_b}) + 0.6$  e l'equalizzazione sia tutta al lato trasmettente;
4. come 3., ma con l'equalizzazione ripartita tra i due estremi del collegamento.

1. - Per il valore di  $\frac{E_b}{N_0}$  otteniamo  $\frac{E_b}{N_0} = \frac{\mathcal{P}_x}{f_b} \frac{1}{2P_n(f)} = \frac{1}{10^7} \frac{1}{10^{-8}} = 10$  pari a 10 dB. Per valutare la  $P_e$  utilizziamo la seconda curva di fig. 15.9 che raffigura la (15.21) per  $\gamma = 1$  ed  $L = 2$  (il nostro caso), ovvero  $P_e^{bit} = \frac{1}{2} \text{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{1}{1.5}} \right\}$ . Dunque per  $\frac{E_b}{N_0} |_{dB} = 10$  dB troviamo una  $P_e \approx 10^{-4}$ ;

2. - siamo nelle condizioni di ricevitore ottimo, e l'espressione della  $P_e$  è data dalla (15.24) ovvero (per  $\gamma = 1$  ed  $L = 2$ )  $P_e^{bit} = \frac{1}{2} \text{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{1}{(1-\frac{\gamma}{4})}} \right\} = \frac{1}{2} \text{erfc} \left\{ \sqrt{\frac{E_b}{N_0} \frac{1}{0.75}} \right\}$ ; essendo il denominatore sotto radice dimezzato, ciò corrisponde ad un equivalente incremento sotto radice di un fattore 2 ovvero di 3 dB, ed in corrispondenza del nuovo  $\frac{E_b}{N_0} |_{dB} = 10 + 3 = 13$  dB troviamo una  $P_e \approx 10^{-7}$ ;

3. - in presenza di un canale distorto equalizzato al trasmettitore, il ricevitore ottimo subisce un peggioramento di prestazioni equivalente ad un decremento di  $\frac{E_b}{N_0} |_{dB}$  della quantità  $\alpha_{dB} = 20 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|^2} df$  che, valutata per via numerica con un programmino Octave, risulta pari a circa 3.5 dB; pertanto al nuovo  $\frac{E_b}{N_0} |_{dB} = 13 - 3.5 = 9.5$  dB corrisponde una  $P_e \approx 3 \cdot 10^{-4}$ ;

4. - qualora l'equalizzazione sia ripartita tra i due estremi il decremento equivalente di  $\frac{E_b}{N_0} |_{dB}$  è pari alla quantità  $\beta_{dB} = 20 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|} df$  che valutata per via numerica risulta pari a circa 3.15 dB, e quindi al nuovo  $\frac{E_b}{N_0} |_{dB} = 13 - 3.15 = 9.85$  dB corrisponde una  $P_e \approx 1.5 \cdot 10^{-4}$ ;



**Equalizzazione al ricevitore** Spesso non è possibile equalizzare il canale al trasmettitore od in modo ripartito<sup>62</sup> e l'equalizzazione deve essere svolta tutta al lato ricevente

<sup>62</sup>Può essere che  $H(f)$  non sia nota a priori e che dunque debba essere stimata al ricevitore (§ 18.4), ma non sia disponibile un canale di ritorno per comunicarla al trasmettitore; oppure si tratti di una

imponendo  $H_T(f) = \sqrt{G(f)}$  e  $H_R(f) = \sqrt{G(f)}/H(f)$ ; si può mostrare che il peggioramento in dB del rapporto  $E_b/N_0$  rispetto al caso di assenza di distorsione è dato anche ora dalla (15.25), mentre il rumore che perviene al decisore ha densità spettrale

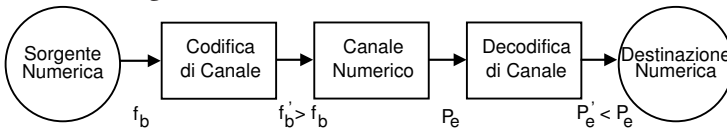
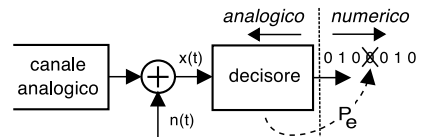
$$\mathcal{P}_v(f) = \frac{N_0}{2} |H_R(f)|^2 = \frac{N_0}{2} \frac{G(f)}{|H(f)|^2}$$

L'argomento viene ripreso al § 18.4, dove si approfondiscono le tecniche di filtraggio *adattivo* che sintetizzano  $H_R(f)$  in modo da minimizzare alcune diverse funzioni obbiettivo.

Ora che abbiamo esaurito la discussione su natura e entità degli errori di trasmissione, occupiamoci di trattare come questi possano essere *mitigati*, ovvero come poterci *convivere*.

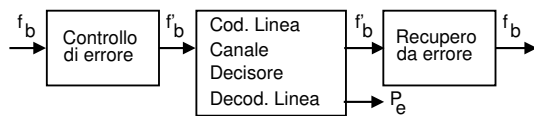
### 15.6 Gestione degli errori di trasmissione

La figura a lato, già proposta a pag. 6, ricorda ancora una volta come le conseguenze prodotte dagli errori del decisore siano alcuni bit *diversi* da quelli trasmessi. Al § 15.4.9.2 siamo giunti al calcolo probabilità di errore  $P_e^{bit}$  dovuta a rumore additivo, mentre al § 15.2.2.3 è stato illustrato come anche una non perfetta temporizzazione, od una alterazione dell'impulso  $g(t)$ , possano causare errori di decisione dovuti all'ISI. Sempre al primo capitolo è anche mostrata la figura riproposta sotto e che illustra come la  $P_e$ , qualunque ne sia la causa, possa essere *ridotta* adottando tecniche di *codifica di canale* e *controllo di errore*, discusse nel seguito.



#### 15.6.1 Controllo di errore

Con questo termine si individuano le strategie atte a *proteggere* le informazioni da trasmettere, aumentando il numero effettivo dei bit inviati (che passano da  $f_b$  a  $f'_b > f_b$  bit/secondo, vedi figura), in modo che i bit aggiunti siano *dependenti* dagli altri, permettendo la gestione degli eventuali errori di trasmissione. Vedremo che la quantità di bit aggiunti, indicata anche come *ridondanza* (§ 15.6.2.1), può essere appena sufficiente a permettere di *accorgersi* della presenza di errori di trasmissione, o (se più elevata), può mettere il lato ricevente anche in grado di *correggere* fino ad una certa percentuale dei bit errati. Sussistono dunque due diverse modalità di gestione degli errori:



trasmissione broadcast (§ 11.1.1.1), e la  $H(f)$  è differente per ognuno dei ricevitori, oppure ancora  $H(f)$  varia nel tempo, e la sua equalizzazione deve essere modificata di continuo.

**Forward error correction o FEC** Qualora il sistema di trasmissione sia da considerare *unidirezionale*<sup>63</sup>, e dunque sprovvisto di un canale di ritorno idoneo a chiedere la *ritrasmissione* dei dati errati, l'unica soluzione consiste nell'aggiunta di una ridondanza sufficiente a correggere direttamente la maggior parte degli eventuali errori. La soluzione prende il nome di correzione di errore *in avanti* o FEC, e si è sviluppata nel contesto della *codifica di canale* (vedi § 15.6.2), appannaggio del mondo delle telecomunicazioni.

**Automatic repeat request o ARQ** Se al contrario è presente un canale di comunicazione *a ritroso*, e non sussistono rigidi vincoli temporali sul massimo ritardo tra trasmissione e ricezione corretta, allora ci si può accontentare di un minor grado di ridondanza, sufficiente ad *accorgersi* degli errori, ovvero a *rivelarli*, ma non a correggerli. Infatti è ora possibile invocare la *ritrasmissione* del dato errato, dando luogo ad una *strategia di richiesta* di ripetizione<sup>64</sup> o ARQ (vedi § 22.6): tale approccio si è sviluppato nel contesto delle reti di computer e della trasmissione dati, ed ha dato origine ai *protocolli a finestra* (§ 23.1.2.3).

**Suddivisione in parole** Le unità informative su cui operano FEC e ARQ in generale non sono singoli bit, ma gruppi di bit denominati *parole* o *word*<sup>65</sup>, e per questo siamo interessati a valutare come la  $P_e^{bit}$  calcolata al § 15.4.9.2 influenzi il *numero* di errori in una parola, dato che da ciò dipende la possibilità di rilevarli e/o correggerli.

### 15.6.1.1 Errori su parole

Occupiamoci quindi di determinare la probabilità  $P(i, n)$  di trovare  $0 \leq i \leq n$  bit errati in una parola di  $n$  bit, qualora ciascun bit possa essere errato con probabilità  $p = P_e^{bit}$  e nell'ipotesi che gli eventi di errore siano statisticamente indipendenti (§ 6.1.5), ovvero che il verificarsi o meno di un errore su di un bit non condizioni gli altri.

Per i casi  $i = 0$  (tutti giusti) ed  $i = n$  (tutti sbagliati) il risultato è immediato, in quanto risulta  $P(0, n) = (1 - p)^n$  e  $P(n, n) = p^n$ . Per  $0 < i < n$  ci troviamo in un

<sup>63</sup>Oltre al caso banale in cui la comunicazione sia effettivamente *half-duplex* (pag. 6), il canale deve essere considerato unidirezionale anche qualora la trasmissione a distanza riguardi informazioni generate in *tempo reale* e *consumate* immediatamente in ricezione, come nel caso televisivo o telefonico, in cui l'attesa di una ritrasmissione introdurrebbe, oltre ad una temporanea interruzione, anche un ritardo aggiuntivo a tutto ciò che viene dopo, impossibile da sostenere in una applicazione interattiva.

Un altro caso di applicazione della tecnica FEC riguarda ad es. il caso di informazioni memorizzate in forma numerica su *data storage*, come ad esempio *CD/DVD*, *chip di memoria*, *hard disk*... in cui pur se possibile ri-leggere le informazioni, ciò non cambierebbe nulla, in quanto l'errore è attribuibile al supporto rovinato, e non al rumore. Per questo i dispositivi di memoria aggiungono una ridondanza ai propri dati, usata per rimediare al possibile deterioramento della loro conservazione, o per segnalare la cella di memoria come inaffidabile.

<sup>64</sup>L'aggettivo *automatic* si riferisce al fatto che spesso la gestione della ritrasmissione avviene a carico di uno strato protocollare (§ 22.5.2.3) di livello *inferiore* a quello che effettivamente consuma il messaggio, che in definitiva neanche si avvede della presenza del meccanismo di ritrasmissione.

<sup>65</sup>In generale questo raggruppamento è indipendente da quello in simboli operato dal codificatore di linea multilivello, così come non riflette altre suddivisioni come i bit di un campione quantizzato (§ 4.3) o gli intervalli temporali di una multiplexazione (§ 24.2) mediante trame (§ 24.3.1) o pacchetti § 22.5.1.

classico caso di *prove ripetute*<sup>66</sup>, in quanto si tratta di osservare l'evento di errore (con probabilità  $p$ ) su  $n$  ripetizioni. La probabilità che ci siano esattamente (ad es. i primi)  $i$  bit errati ha valore  $p^i (1 - p)^{n-i}$ , ma dato che i bit errati possono essere comunque distribuiti su  $n$ , e che vi sono  $\binom{n}{i} = \frac{n(n-1)\cdots(n-i+1)}{i!}$  modi di scegliere  $i$  oggetti su  $n$ , il risultato cercato è espresso dalla d.d.p. di Bernoulli (vedi § 22.1)

$$P(i, n) = \frac{n(n-1)\cdots(n-i+1)}{i!} p^i (1-p)^{n-i} \tag{15.26}$$

che, se  $np < 0.1$ , può essere approssimata come<sup>67</sup>

$$P(i, n) \approx \frac{n(n-1)\cdots(n-i+1)}{i!} p^i \tag{15.27}$$

La (15.27) applicata al caso di un singolo errore ( $i = 1$ ) su  $n$  fornisce  $P(1, n) \approx np$ <sup>68</sup>, ovvero la probabilità di *un solo* bit errato su  $n$  è circa pari ad  $n$  volte la  $P_e^{bit}$  nel flusso binario. D'altro canto, per la probabilità di *due* bit errati su  $n$  la (15.27) fornisce  $P(2, n) \approx \frac{1}{2}n(n-1)p^2$  e, sempre se  $np < 0.1$ , osserviamo che  $P(2, n) \ll np \approx P(1, n)$ , ovvero inferiore a quella di un solo bit errato. Più in generale, risulta che

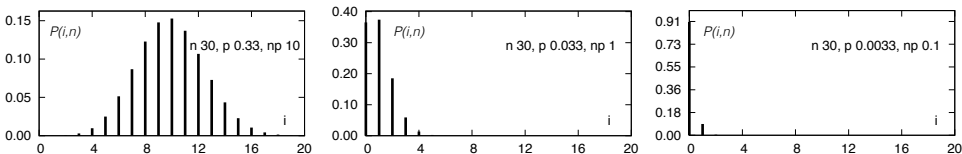
$$P(i+1, n) \ll P(i, n) \tag{15.28}$$

e quindi si può considerare la probabilità di ricevere  $i$  o *più* bit errati su  $n$ , praticamente uguale a quella di osservare solo  $i$  errori. All'aumentare di  $p$  e/o di  $n$ , l'approssimazione perde validità, e la probabilità  $P(i, n)$  può invece *aumentare* con  $i$ , e in tal caso il sistema di trasmissione è praticamente inusabile. E' questo il motivo per cui non è opportuno che la lunghezza  $n$  delle parole<sup>69</sup> ecceda il limite  $np < 0.1$  imposto dalla probabilità di errore sul bit  $p = P_e^{bit}$  offerta dal collegamento.

L'esposizione prosegue introducendo per prime definizioni e soluzioni che rea-

<sup>66</sup>Vedi ad es. [https://it.wikiversity.org/wiki/Prove\\_ripetute](https://it.wikiversity.org/wiki/Prove_ripetute)

<sup>67</sup>Dal confronto tra (15.26) e (15.27) osserviamo che l'approssimazione consiste nel considerare  $(1-p)^{n-i} \approx 1$ . Per verificare che ciò sia lecito qualora  $np \ll 1$ , cerchiamo il valore di  $p$  tale che  $(1-p)^n > 0.9$ , da cui discende che anche  $(1-p)^{n-i} > 0.9$  per qualunque  $i$ . Dato che si può scrivere (vedi nota 2 a pag. 764)  $(1-p)^n = \sum_{k=0}^n \binom{n}{k} p^k > 1 - np$ , otteniamo la condizione  $(1-p)^n > 1 - np > 0.9$ , da cui si ottiene  $np < 1 - 0.9 = 0.1$ : ad esempio, se  $n = 1000$  occorre che sia almeno  $p = 10^{-4}$ . Altrimenti l'approssimazione non è valida, e la (15.28) deve essere verificata; qui sotto mostriamo la d.d.p. di Bernoulli (15.26) per diversi valori di  $np$ , evidenziando come qualora  $np < 0.1$  essa risulti monotona decrescente con  $i$ .



<sup>68</sup>In linea di principio, dato che la probabilità che solo *il primo* bit su  $n$  sia sbagliato è pari a  $p(1-p)^{n-1}$  e che lo stesso risultato si ottiene anche per gli altri  $n-1$  casi possibili, la probabilità  $P(1, n)$  di *un solo* (generico) bit sbagliato su  $n$  è pari a  $P(1, n) = np(1-p)^{n-1}$ , che si approssima come  $np$  qualora si consideri  $(1-p)^{n-1} \approx 1$  in virtù della condizione  $np \ll 1$ .

<sup>69</sup>Così come non è opportuno aumentare di troppo la dimensione di un *pacchetto dati*, anche se in tal modo si riduce l'*overhead*, vedi § 22.5.1.

lizzano la *correzione* degli errori, il cui approfondimento è svolto al § 17.4, mentre al § 15.6.3 sono illustrate tre tecniche comunemente utilizzate per la *detezione* di errore; la descrizione dei protocolli ARQ è infine rimandata al § 22.6.

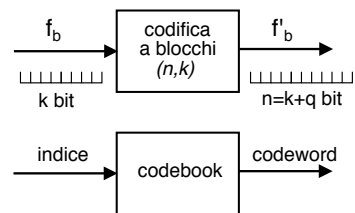
## 15.6.2 Correzione di errore e codifica di canale

Affrontiamo i criteri con cui scegliere i bit di ridondanza allo scopo di realizzare un sistema di tipo FEC basato sulle tecniche di *codifica di canale*, peraltro idonee ad essere utilizzate anche per il caso dei sistemi ARQ. Descriviamo quindi due semplici metodi di correzione di errore, ossia il *codice a ripetizione* e *l'interleaving*.

### 15.6.2.1 Codice a blocco

La *codifica a blocco* opera raggruppando  $k$  bit consecutivi del messaggio originale, distinti dai precedenti e dai successivi blocchi di  $k$ . Per ogni  $k$  bit da trasmettere il codificatore produce  $n = k + q$  bit in uscita che sono trasmessi al posto dei  $k$  di ingresso, da cui dipendono in modo univoco.

**Codeword e codebook** L'operazione di *mappatura* svolta dal codificatore può essere pensata o realizzata come l'accesso ad una tabella (o memoria) denominata CODEBOOK (o *cifrario*), dove i  $k$  bit da codificare rappresentano un *indice* che individua  $2^k$  differenti righe, in cui si trovano scritte le *parole di codice* (CODEWORD), costituite ognuna da  $n = k + q$  bit. Un codice siffatto è detto *codice*  $(n, k)$ .



**Ridondanza** Esprime la proporzione percentuale tra il numero dei  $q$  bit di protezione aggiunti rispetto ai  $k$  (di informazione) in ingresso al codificatore, ovvero

$$\rho = \frac{q}{k} \cdot 100 \%$$

ed è una misura del grado di protezione offerto dal processo di codifica.

**Esempio** In una trasmissione con ridondanza del 50% per ogni coppia di bit di informazione ne viene inserito uno di protezione.

La presenza di ridondanza fa sì che il numero  $2^k$  di codeword esistenti sia *inferiore* a quello delle  $2^n$  possibili configurazioni di  $n$  bit, in modo che se un bit di una codeword viene modificato a causa di un errore del decisore, la nuova parola di  $n$  bit individua una configurazione che nel codebook *non esiste*, permettendo così di rivelare e/o correggere l'errore. Ma cosa succede se avviene più di un errore per codeword? Per rispondere, introduciamo un nuovo concetto:

**Distanza di Hamming** E' indicata come  $d_H(c_i, c_j)$  e misura la *dissimilarità* tra due codeword  $c_i, c_j$ , espressa come il numero di posizioni di bit in cui esse sono diverse. Viene calcolata eseguendo *l'OR esclusivo* delle rispettive rappresentazioni binarie, e contando il numero di *uni* del risultato.

**Esempio** Con  $c_i = 011010$  e  $c_j = 010110$  si ottiene il risultato mostrato a lato, che evidenzia come le codeword differiscano in due posizioni di bit, e dunque  $d_H(c_i, c_j) = 2$ .

$$\begin{array}{r} 011010 \oplus \\ 010110 = \\ \hline 001100 \end{array}$$

**Distanza del codice** Individua la *minima* distanza di Hamming  $d_m$  tra tutte le possibili coppie di codeword di uno stesso codebook

$$d_m = \min_{i \neq j} d_H(c_i, c_j)$$

e permette di valutare la capacità di rivelazione e correzione del codebook, in quanto rappresenta il minimo numero di errori necessario a trasformare una codeword in un'altra, almeno nel caso peggiore di due parole a distanza  $d_m$ .

**Rivelazione e correzione di errore** Un codice con distanza  $d_m$  può

- *rivelare* al massimo  $d_m - 1$  errori, in quanto se ne avvengono di più *si finisce* in un'altra codeword, e
- *correggere* fino a  $\frac{d_m-1}{2}$  errori, oltre i quali si finisce *più vicini* ad una altra codeword.

Mentre la prima azione è possibile ogniqualvolta si osservi una codeword  $c_\alpha$  non presente nel codebook, per intraprendere la seconda occorre operare una decisione relativa a quale sia la codeword  $\hat{c}$  *realmente* trasmessa in base ad un criterio di *minima distanza*, ovvero scegliendo la codeword più vicina (nel senso della distanza di Hamming) a quella ricevuta:

$$\hat{c} = \arg \min_{c_i} d_H(c_i, c_\alpha)$$

**Esempio** Nel caso in cui  $d_m = 3$ , la presenza di *un solo* bit errato su  $n$  fa sì che la codeword ricevuta differisca da quella trasmessa per *un* bit, mentre mantiene almeno *due bit* di differenza rispetto a tutte le altre possibili, permettendo così al ricevitore di correggere l'errore. Se sono invece presenti *due* errori, la parola ricevuta diviene più vicina ad una codeword *diversa* da quella trasmessa, ed in tal caso la procedura di correzione sceglierebbe una codeword errata.

E' quindi un po' come se attorno ad ogni codeword fosse costruita *una sfera* contenente tutte le parole di  $n$  bit che *non sono codeword* e che sono *distanti* dalla codeword al massimo  $\frac{d_m-1}{2}$  bit: la ricezione di ciascuna di esse comporta la decisione per la codeword al centro della sfera.

**Massima distanza minima** Per un generico codice a blocco  $(n, k)$  la distanza *del codice*  $d_m$  rispetta la disegualianza

$$d_m \leq q + 1$$

ovvero  $d_m \leq n - k + 1$  dato che  $n = k + q$ . Infatti i  $2^k$  bit da proteggere sono qualunque, e dunque il contributo di questi  $k$  bit alla distanza tra codeword è uno; tale distanza può aumentare al massimo di una ulteriore unità per ognuno dei  $q$  bit aggiunti.

**Esempio** Aggiungendo  $q = 3$  bit di ridondanza si ha  $d_m \leq q + 1 = 4$ , e se per una particolare scelta del codebook (ad es. di Hamming, § 17.4.1.1) si ottiene  $d_m = 3$ , saremo in grado di correggere un errore e rivelarne 2.

**Probabilità di errore residua per codeword** Da quanto illustrato fin qui occorre stabilire *a priori* se utilizzare il codice a fini di correzione oppure di detezione. Indicando con  $e_M$  il massimo numero di errori che è possibile correggere o rivelare, può essere definita una probabilità *residua* di errore *su parola*  $P_e^r$  per descrivere il caso in cui, anche dopo l'esecuzione delle procedure di controllo di errore<sup>70</sup>, siano ancora presenti errori, perché in numero superiore ad  $e_M$  e dunque eccedenti la capacità correttiva del codice. Risulta quindi che  $P_e^r = P(i > e_M, n)$  che, nelle condizioni di validità della (15.27), è pari a  $P(e_M + 1, n)$ . In linea generale, la *valutazione* dell'errore residuo *sul bit* dipende dal tipo di controllo di errore attuato.

**Efficienza** L'efficienza del codice è misurata dal *tasso di codifica* (CODE RATE)

$$R_c = \frac{k}{n} < 1$$

che rappresenta la frazione di bit informativi sul totale di quelli trasmessi, e che consente di scrivere la velocità di uscita dal codificatore come

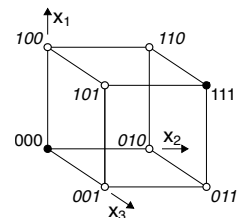
$$f_b' = \frac{f_b}{R_c} > f_b \quad (15.29)$$

**Esempio** In una trasmissione con un tasso di codifica pari a 0.5, il numero di bit uscenti (per unità di tempo) dal codificatore di canale è il doppio del numero dei bit entranti.

L'argomento dei codici a blocco è molto vasto<sup>71</sup>, e fornisce molteplici soluzioni, la cui trattazione esauriente eccede il livello di approfondimento del presente capitolo (vedi però il § 17.4.1); gli stessi tre casi di controllo di errore trattati al § 15.6.3 (parità, somma di controllo e CRC) possono essere inquadrati nel contesto dei *codici a blocco*. Qui ci limitiamo ad un esempio di codice a correzione molto elementare, il codice *a ripetizione*, mentre al § 17.4.1.1 è illustrata una tecnica nota come codice *di Hamming*, in grado di conseguire una efficienza di gran lunga migliore.

### 15.6.2.2 Codice a ripetizione n:1

Realizza un codice a blocco molto semplice e con proprietà correttive, per il quale  $k = 1$  e le uniche due codeword sono di  $n$  bit tutti uguali al bit in ingresso; ponendo ad esempio  $n = 3$  si ottiene il codice *a ripetizione* 3 : 1, con codeword 000 ed 111, di cui in figura è riportata la disposizione (rappresentate come pallini) in uno spazio vettoriale descritto dai tre bit di codice  $x_1 x_2 x_3$ , ed in cui i cerchi vuoti indicano gli 8-2=6 vettori binari che *non sono* codeword, scritti in *corsivo*. Se gli errori sono sufficientemente distanti nel tempo la correzione può basarsi su di una "votazione a maggioranza" (*majority voting*): sempre nel caso di  $n = 3$  si



<sup>70</sup>Che nel caso di rivelazione richiede la ritrasmissione della parola errata.

<sup>71</sup>Senza pretesa di esaustività, possiamo annoverare l'esistenza dei codici *di Hamming*, *di Hadamard*, *BCH*, *Reed-Solomon*, *Reed-Muller*, *di Golay*, *di Gallager*, *turbo*, *a cancellazione*, *a fontana*, *punturati*...

ottiene  $d_m = 3^{72}$ , ed il codice è in grado di correggere un errore e rivelarne due<sup>73</sup>.

Notiamo come questo codice sia particolarmente poco efficiente, dato che per esso si ottiene un tasso di codifica  $R_c = \frac{k}{n} = \frac{1}{n}$ ; d'altra parte, il codice a ripetizione è uno dei pochi per cui  $d_m = q + 1$ , e non meno.

**Esercizio** Calcolare la probabilità di errore residua  $P^r$  per un codice a ripetizione 3:1, in presenza di una  $P_e^{bit} = p$ . **Risp.** Come discusso il codice può correggere un errore singolo, mentre in presenza di un errore doppio la decisione a maggioranza modifica anche il terzo bit, ed un errore triplo passa inosservato. La (15.27) approssima la probabilità che due o più bit su tre siano errati come<sup>74</sup>  $P(2, 3) \simeq \frac{1}{2}3 \cdot 2p^2 = 3p^2$  che è la  $P^r$  cercata per codeword. Dato che per il codice a ripetizione ad ogni codeword corrisponde un solo bit del messaggio originario, lo stesso valore di  $P^r = 3p^2$  è anche la  $P_e^{bit}$  residua. Ad esempio, in corrispondenza di una  $p = 10^{-4}$  iniziale, dopo decodifica si ottiene una  $P_e^{bit} = 3 \cdot 10^{-8}$ .

**Compromesso banda - potenza** Torniamo su questo argomento (§ 15.4.7) in quanto l'adozione di un codice di canale aumenta di fatto la  $f_b$  eq. (15.29) e dunque l'occupazione di banda eq. (15.5); non solo, ma a parità di potenza  $\mathcal{P}_x$  ricevuta l'aumento di  $f_b$  comporta la riduzione di  $E_b = P_x/f_b$  e dunque di  $E_b/N_0$  della stessa frazione, e quindi un peggioramento della  $P_e$  di base su cui opera il decodificatore di linea. Evidentemente il miglioramento apportato dalla codifica FEC sopperisce anche al peggioramento dovuto all'aumento di banda, che la teoria dell'informazione prevede possa essere ripagato da un minor bisogno di potenza per ottenere le stesse prestazioni.

**Esercizio** Proseguendo l'esercizio precedente, osserviamo che il codice a ripetizione determina una  $f_b$  tripla, e dunque un  $E_b/N_0$  ridotto di un terzo, ovvero di circa -4.7 dB inferiore rispetto al caso non codificato. Dalle curve di fig. 15.12 riscontriamo che il miglioramento sulla  $P_e$  (da  $10^4$  a  $3 \cdot 10^{-8}$ ) dovuto al codice corrisponde più o meno allo stesso numero di dB. Per codici più efficienti (con minore aumento di banda come ad esempio il codice di Hamming, vedi § 17.4.1.1) il bilanciamento tra i due effetti conferma il risultato anticipato dal compromesso banda-potenza, qui applicato alla codifica di canale.

### 15.6.2.3 Interleaving

Le capacità di correzione fino ad ora discusse sono valide purché gli eventi di errore siano *staticamente indipendenti*. In realtà gli errori possono presentarsi in maniera *non* indipendente, e concentrati in un breve intervallo di tempo: questa circostanza è indicata come *caso degli errori a pacchetto*<sup>75</sup>. In tal caso, si usa ricorrere alla tecnica

<sup>72</sup>Con riferimento alla figura, 3 è il numero di vertici da attraversare, ovvero di errori da subire, per passare da una codeword all'altra.

<sup>73</sup>Poniamo di dover trasmettere 0110. La sequenza diventa 000 111 111 000 e quindi, a causa di errori, ricevo 000 101 110 100. Votando a maggioranza, ricostruisco la sequenza corretta 0 1 1 0.

<sup>74</sup>Volendo essere *esatti* la probabilità di 2 bit errati su 3 è data dalla d.d.p. binomiale (§ 22.1) ed è pari a  $\binom{3}{2}p^2(1-p) = 3p^2(1-p)$ , a cui va sommata la probabilità di 3 bit errati, pari a  $p^3$ . Pertanto  $P^r = 3p^2(1-p) + p^3 = 3p^2 - 3p^3 + p^3 = 3p^2 - 2p^3 \simeq 3p^2$ , approssimazione legittima se  $np = 3p \ll 1$ .

<sup>75</sup>In inglese, errori a *burst* (*scoppio*). Dovuti a rumori e disturbi di tipo *impulsivo*, ad esempio a causa di scintille come per motori elettrici o candele di accensione, o fenomeni di fast fading (§ 20.4.4).

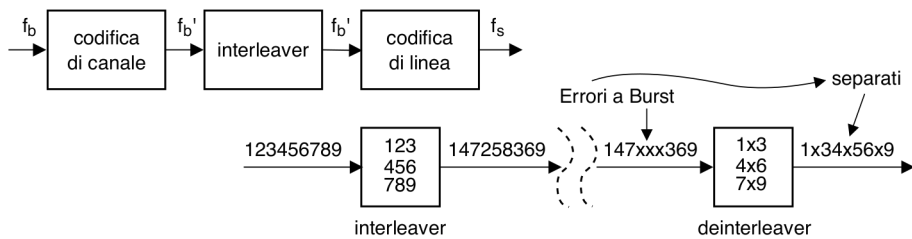


Figura 15.15: Schema di principio della trasmissione a dati intercalati

nota come *scrambling*<sup>76</sup> o *interleaving*<sup>77</sup> attuabile a patto di accettare un ritardo di trasmissione. Si tratta infatti di modificare l'ordine dei dati inviati, in modo che gli errori che si manifestano su bit *vicini* si riflettano in errori su bit... *lontani*, e quindi appartenenti a codeword differenti, come illustrato in fig. 15.15 in cui i bit sono scritti per righe e letti per colonne. Una analisi più approfondita viene svolta a pag. 573. Ovviamente, occorre prevedere un processo inverso (*descrambling* o *deinterleaving*) all'altro capo del collegamento. E' appena il caso di notare che lo scrambler (similmente al codice di Gray) *non altera* il *numero* dei bit trasmessi, e dunque *non è* una forma di codifica FEC.

La trattazione della codifica di canale di tipo FEC prosegue in modo più approfondito al § 17.4, dove sono illustrate le tecniche più avanzate.

### 15.6.3 Detezione di errore

In questo caso ci si limita a mettere il ricevitore in grado di *accorgersi* della presenza di errori, senza poterli correggere. Le parole errate possono essere scartate, oppure può esserne richiesta la ritrasmissione.

#### 15.6.3.1 Controllo di parità

Viene comunemente usato nell'ambito della trasmissione asincrona (§ 15.7.1) e sincrona orientata al carattere (pag. 481) per rivelare errori sul bit, e consiste nell'aggiungere alla parola da trasmettere un unico ulteriore bit, in modo che in totale ci sia un numero *pari* di uni<sup>78</sup>, applicando così una regola di *parità pari* (EVEN). Il caso opposto, ossia l'aggiunta di un bit in modo da rendere *dispari* il numero di uni, prende nome di parità ODD.

In entrambi i casi<sup>79</sup> dopo aver raggruppato i bit pervenuti il ricevitore esegue un *controllo* detto appunto *di parità*, semplicemente contando<sup>80</sup> il numero di uni, ed

<sup>76</sup>Letteralmente: arrampicamento, ma anche "arruffamento", vedi *scrambled eggs*, le uova strapazzate dell'*english breakfast*.

<sup>77</sup>LEAVE = *foglia, sfogliare, rastrellare*, ed il termine potrebbe essere tradotto come *intercalamento*.

<sup>78</sup>Ad esempio, alla sequenza 001001 verrà aggiunto uno 0, mentre a 010101 si aggiungerà ancora un 1, perché altrimenti gli uni complessivi sarebbero stati 3, che è dispari.

<sup>79</sup>Il ricevitore deve comunque essere al corrente del fatto se la parità sia ODD o EVEN !

<sup>80</sup>La *conta* può essere realizzata in forma algoritmica o circuitale, eseguendo la somma modulo due di tutti i bit che compongono la parola (ovvero complementando il risultato, nel caso di parità *dispari*). La somma modulo due è equivalente all'operazione di OR esclusivo, viene a volte indicata con il simbolo  $\oplus$ , e corrisponde alla regola  $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$ .

accorgendosi così se nella parola si sia verificato un errore (uno zero divenuto uno o viceversa). In tal caso, il ricevitore invierà all'altro estremo del collegamento una richiesta di ritrasmissione del gruppo di bit errato. Se invece si è verificato un errore che coinvolge *due* bit della parola, questo passerebbe inosservato, in quanto la parità prescritta verrebbe mantenuta. Infatti, la *distanza di Hamming* (vedi pag. 469) relativa ad un codebook ottenuto aggiungendo ad ogni possibile parola di  $k$  bit il corrispondente di parità, è pari a due<sup>81</sup>.

**Esempio:** indicando la probabilità di errore sul bit con  $p$  (es  $10^{-3}$ ) ed applicando la (15.27) si ottiene che la probabilità di  $i = 2$  errori su  $n = 10$  bit vale  $P_e^{word} = \frac{1}{2}n(n-1)p^2 = 4.5 \cdot 10^{-5}$ , che rappresenta il *tasso residuo* di errore *su parola* legato all'uso di un bit di parità: essendo due i bit errati su 10, la  $P_e^{bit} = P\{err/word_e\} \cdot P_e^{word}$  risulta pari a  $\frac{2}{10} \cdot 4.5 \cdot 10^{-5} = 0.9 \cdot 10^{-5}$ , un bel risultato rispetto al  $10^{-3}$  di partenza.

Il concetto di parità può essere esteso calcolando  $q$  bit di parità, ognuno a partire da un *diverso sottoinsieme* dei  $k$  bit di ingresso, con sottoinsiemi eventualmente sovrapposti. Un codice del genere prende il nome di codice di *Hamming*, descritto al § 17.4.1.1.

### 15.6.3.2 Somma di controllo o checksum

Quando il messaggio è composto da  $M$  diverse parole di  $N$  bit, la probabilità che almeno una di queste sia errata aumenta in modo circa proporzionale ad  $M$ , in base ad un ragionamento del tutto analogo a quello della nota 68.

Per aumentare le capacità di rivelazione del controllo di parità applicato sulle singole parole (indicato ora come parità *di riga*, o *trasversale*), si aggiunge al gruppo di  $M$  parole una ulteriore parola (detta *somma di controllo* o *checksum*), i cui bit si ottengono applicando il controllo di parità a tutti i bit "omologhi" delle  $M$  parole incolonnate, generando così una parità *di colonna* (o *longitudinale*), come esemplificato in figura.

000010	0	}	parità di riga ( <i>ODD</i> )		
010100	1				
100011	0				
010000	0				
010110	1				
100000	0				
110011	1				
000011	1				
100001	1			←	parità di colonna ( <i>EVEN</i> )
100001	1				

A volte, si preferisce calcolare la somma di controllo mediante una operazione di somma *modulo uno*<sup>82</sup>, direttamente realizzabile in software in modo veloce. In tal caso, il ricevitore calcola una nuova somma di controllo longitudinale, includendo anche la somma di controllo originaria: in assenza di errori, il risultato deve fornire zero.

### 15.6.3.3 Codice polinomiale e CRC

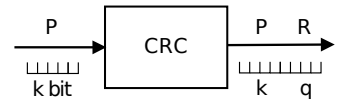
L'utilizzo di una somma di controllo può produrre risultati scadenti nel caso di una distribuzione temporale dei bit errati particolarmente sfavorevole, mentre la tecnica nota come *Cyclic Redundancy Check* (o CRC)<sup>83</sup> garantisce prestazioni *più uniformi*.

<sup>81</sup>Considerando parole di 3 bit, le codeword (di 4 bit, in cui l'ultimo è una parità pari) risultano: (0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111). E' facile constatare che ognuna di esse differisce da tutte le altre per almeno due bit.

<sup>82</sup>La somma modulo uno è l'equivalente binario dell'operazione di somma (decimale) tradizionale, comprese quindi le operazioni di riporto verso le cifre più elevate. Il riporto finale viene poi nuovamente sommato al risultato della somma.

<sup>83</sup>Tale denominazione indica un'azione di controllo (*check*) realizzata mediante l'aggiunta di una *ridondanza* ottenuta applicando un codice *ciclico* - vedi § 17.4.1.2.

Questo metodo consiste nell'aggiungere ad una parola  $P$  di  $k$  bit che si desidera trasmettere, un gruppo  $R$  di  $q < k$  ulteriori bit di protezione, calcolati a partire dai primi  $k$ , in modo da permettere la detezione di eventuali errori; sotto questo aspetto, il CRC rientra nella categoria dei codici a blocco (§ 15.6.2.1).



L'aggettivo *polinomiale* trae origine dalla associazione tra un numero binario  $B$  di  $n + 1$  bit, indicati con  $b_i, i = 0, 1, \dots, n$ , ed un polinomio<sup>84</sup>  $B(x)$  a coefficienti binari nella variabile  $x$ , di grado  $n$ , con espressione

$$B(x) = b_n x^n + b_{n-1} x^{n-1} + \dots + b_1 x^1 + b_0$$

Un *codice polinomiale* è definito a partire da un *polinomio generatore*  $G(x)$  di grado  $q$ , i cui coefficienti binari identificano una parola  $G = g_q g_{q-1} \dots g_1 g_0$  di  $q + 1$  bit.

Indicando ora con  $P$  la sequenza dei  $k$  bit  $p_i$  da proteggere, *aggiungiamo* a destra di questi un gruppo di  $q$  bit pari a zero, ottenendo una nuova parola  $P \cdot 2^q$  lunga  $k + q$  bit, che quindi dividiamo per  $G$  (mediante aritmetica modulo due<sup>85</sup>), ottenendo un quoziente  $Q$ , ed un resto  $R$  con al massimo  $q$  bit. Pertanto, possiamo scrivere

$$\frac{P \cdot 2^q}{G} = Q \oplus \frac{R}{G}$$

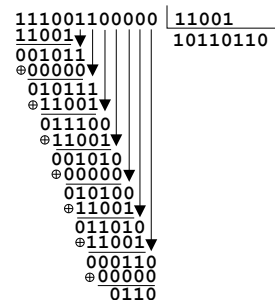
Le sequenze  $Q$  ed  $R$  costituiscono rispettivamente i coefficienti dei polinomi quoziente  $Q(x)$  e resto  $R(x)$ , ottenibili dalla divisione di  $P(x) \cdot 2^q$  per  $G(x)$ . I  $q$  bit  $R$  del resto sono quindi utilizzati come *parola di protezione*, in modo da esprimere la sequenza  $T$  da trasmettere come  $T = P \cdot 2^q \oplus R$  di  $k + q$  bit, ovvero con i  $k$  bit più significativi pari a  $P$  ed i  $q$  bit in coda pari ad  $R$ . Il ricevitore effettua anch'esso una divisione, stavolta tra  $T$  e  $G$ , che in assenza di errori produce un *resto nullo*

$$\frac{T}{G} = \frac{P \cdot 2^q \oplus R}{G} = \frac{P \cdot 2^q}{G} \oplus \frac{R}{G} = Q \oplus \frac{R}{G} \oplus \frac{R}{G} = Q$$

<sup>84</sup>L'insieme di tutti i polinomi di grado minore od uguale ad  $n$  costituisce un particolare spazio algebrico, per il quale è possibile dimostrare una serie di proprietà, la cui verifica trascende dallo scopo di questo testo, e che consentono di stabilire le capacità del codice di rivelare gli errori.

<sup>85</sup>Per fissare le idee, consideriamo  $k = 8$  bit a da proteggere, pari a  $P = 11100110$ ,  $q = 4$  bit di CRC, ed un generatore  $G = 11001$ . La sequenza  $P \cdot 2^q$  risulta pari a  $11100110\ 0000$ , e la divisione modulo 2 tra  $P$  e  $G$  fornisce un quoziente  $Q = 10110110$  (che viene ignorato) ed un resto  $R$  pari a  $0110$ . Pertanto, viene trasmessa la sequenza  $T = P \cdot 2^q \oplus R = 11100110\ 0110$  con  $k + q = 12$  bit.

La divisione modulo 2 si realizza come mostrato nella figura a lato: considerando i bit più significativi di  $P \cdot 2^q$  e  $G$ , l'uno nell'uno ci sta una volta, e scriviamo uno come primo bit di  $Q$ . Riportiamo ora  $G$  sotto  $P \cdot 2^q$ , ed anziché sottrarre i bit, ne calcoliamo l'OR-ESCLUSIVO  $\oplus$  bit-a-bit, ottenendo  $00101$ , a cui aggiungiamo un uno *abbassando* il successivo bit (1) di  $P \cdot 2^q$ . Stavolta l'uno nello zero ci sta zero volte, e dunque aggiungiamo uno zero a  $Q$ , riportiamo cinque zeri (come la lunghezza di  $G$ ) allineati sotto al resto parziale, eseguiamo l'EXOR, ed abbassiamo un'altra cifra (1) di  $P$ . Il confronto ora è tra il quinto bit da destra del resto parziale (1) ed il bit più significativo (il quinto, 1) di  $G$ , ottendo la terza cifra di  $Q$  (1). Ripetiamo il procedimento, e quando tutti i bit del divisore sono stati usati, l'ultima operazione  $\oplus$  fornisce il resto  $R$  cercato.



in quanto sommando in aritmetica modulo due un numero per se stesso, si ottiene un risultato nullo. Pertanto, se  $T/G = Q$  con resto nullo, la parola  $P$  è riottenuta semplicemente shiftando  $T$  a destra di  $q$  posizioni.

Nel caso invece in cui si siano verificati errori, indichiamo con  $E$  la sequenza binaria di errore, di lunghezza  $k + q$  bit, ognuno dei quali è pari ad uno se in quella posizione si verifica errore, o zero in caso contrario, in modo da rappresentare il segnale ricevuto  $R$  come  $R = T \oplus E$ . Se  $E \neq 0$  la divisione operata al ricevitore ora fornisce

$$\frac{R}{G} = \frac{T \oplus E}{G} = \frac{T}{G} \oplus \frac{E}{G} = Q \oplus \frac{E}{G} \quad (15.30)$$

e quindi si verifica la presenza di un resto diverso da zero<sup>86</sup>, che indica appunto la presenza di errori, tranne nei casi in cui  $E$  risulti perfettamente divisibile per  $G$ , evento con bassa probabilità se  $G$  è scelto opportunamente. Nel caso in cui  $q = 1$  si ricade nel caso del controllo di parità (§ 15.6.3.1), a cui corrisponde  $G(x) = x + 1$ .

Per applicare il metodo, sia il trasmettitore che il ricevitore devono utilizzare lo stesso generatore  $G(x)$ , per il quale esistono diverse scelte standardizzate<sup>87</sup>. Si può dimostrare che scegliendo  $G(x)$  in modo opportuno, il metodo discusso permette di rivelare

- tutti gli errori singoli;
- se  $G(x)$  contiene il termine noto +1, tutti gli errori a burst che si estendono per  $q$  o meno bit;
- se  $x + 1$  è un fattore di  $G(x)$ , tutti gli errori in numero dispari;
- se  $G(x)$  è un polinomio primitivo<sup>88</sup>, tutti gli errori doppi;
- se  $G(x)$  è un polinomio primitivo di grado  $q - 1$  moltiplicato per il fattore  $x + 1$ , tutti gli errori doppi entro un intervallo di  $2^{q-1} - 1$  bit, e tutti gli errori in numero dispari.

**Calcolo del CRC** L'aspetto che ha reso popolare questo metodo è la maniera in cui è possibile calcolare i  $q$  bit di controllo, che vengono essi stessi indicati come CRC. Infatti la divisione binaria illustrata alla nota 85 è realizzabile a livello circuitale in modo relativamente semplice<sup>89</sup>.

Si tratta di utilizzare un registro a scorrimento *controeazionato* (vedi figura seguente), in cui i  $k$  bit da proteggere sono immessi ad uno ad uno da destra verso sinistra,

<sup>86</sup>Dalla (15.30) sembrerebbe che il resto sia  $E$ , ma dato che  $E(x)$  può avere grado  $> q$ , esso è divisibile per  $G(x)$ , e dunque il resto *non è*  $E$  - altrimenti, *sarebbe possibile correggerlo!*

<sup>87</sup>Ecco quattro scelte utilizzate nei sistemi di trasmissione:

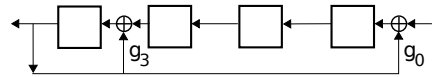
CRC-12	$G(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	$G(x) = x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$G(x) = x^{16} + x^{12} + x^5 + 1$
CRC-32	$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Come discusso, un polinomio di ordine  $q$  genera un CRC di  $q$  bit; pertanto il CRC-12, che è usato per caratteri a 6 bit, genera 12 bit di CRC, mentre CRC-16 e CRC-CCITT, utilizzati in America ed in Europa rispettivamente per caratteri ad 8 bit, producono 16 bit di CRC. In alcuni standard di trasmissione sincrona punto-punto, è previsto l'uso di CRC-32.

<sup>88</sup>[https://it.wikipedia.org/wiki/Polinomio\\_primitivo](https://it.wikipedia.org/wiki/Polinomio_primitivo)

<sup>89</sup>Vedi ad es. [http://en.wikipedia.org/wiki/Computation\\_of\\_cyclic\\_redundancy\\_checks](http://en.wikipedia.org/wiki/Computation_of_cyclic_redundancy_checks)

seguiti da  $q$  zeri consecutivi. Per ogni valore immesso, quelli già presenti *scorrono* a sinistra nel registro, ed il bit che *trabocca*, oltre a rappresentare una cifra del quoziente, alimenta gli OR esclusivi presenti nel registro; questi ultimi sono posti in corrispondenza dei coefficienti di  $G(x)$  pari ad uno, tranne che per quello corrispondente ad  $x^q$ .



Al termine dell'inserimento dei  $k + q$  bit di  $P \cdot 2^q$ , lo stato del registro a scorrimento costituisce proprio il resto  $R$ , da usare come CRC. L'esempio in figura rappresenta il caso mostrato alla nota 85, in cui di  $G(x) = x^4 + x^3 + 1$ : con un po' di pazienza, è possibile verificare che il circuito effettivamente svolge i calcoli prescritti dalla procedura di divisione, e che si ottiene lo stesso risultato.

## 15.7 Sincronizzazione dati

Come mostrato al § 15.1.1 il segnale dati deve essere campionato al ricevitore con cadenza pari alla frequenza di simbolo  $f_s$  il più possibile in prossimità del centro dell'intervallo di simbolo, in modo da contrastare gli effetti della limitazione di banda (vedi fig. 15.3) e dell'ISI (§ 15.2.2.2); per questo motivo occorre che il temporizzatore mostrato in fig. 15.1 determini gli istanti di campionamento più idonei, effettuando la *sincronizzazione di simbolo*<sup>90</sup>. Le diverse scelte per l'onda elementare  $g(t)$  discusse al § 15.2.1, determinano differenti gradi di "difficoltà" nel conseguimento di tale obiettivo.

Un diverso aspetto della sincronizzazione riguarda il problema di ricostruire la struttura *sintattica*<sup>91</sup> del segnale binario, in primo luogo per permettere il corretto svolgimento delle operazioni di decodifica FEC e di controllo di errore. Inoltre, la sequenza di bit al ricevitore è spesso il risultato della serializzazione (al lato del trasmettitore) di informazioni *a carattere* (o *parola*, o *WORD*), come nel caso di un *file* di testo oppure dei campioni di un segnale<sup>92</sup>, oltre alle quali possono essere state introdotte ulteriori strutture sintattiche legate ai protocolli di moltiplicazione sia a circuito (§ 24.3.4 e § 24.4) che a pacchetto (§ 22.5.1). Lo *strato di collegamento*, il secondo della pila ISO-OSI, si occupa appunto di segmentare il flusso di bit ricevuto in accordo a tali strutture, coinvolgendo essenzialmente il sotto-strato MAC (§ 23.1.4), che deve quindi essere in grado di conseguire la sincronizzazione di *simbolo*, di *parola* e di *trama*.

Nel seguito analizziamo le esigenze e le soluzioni di sincronizzazione che emergono

<sup>90</sup>In alternativa al recupero del sincronismo da parte del ricevitore, l'informazione di temporizzazione può essere trasmessa su di una diversa linea, come avviene nel caso di dispositivi ospitati su di una stessa *motherboard*.

<sup>91</sup>Una sintassi definisce un linguaggio, prescrivendo le regole con cui possono essere costruite sequenze di simboli noti (l'alfabeto), e l'analisi delle sequenze eseguita nei termini degli elementi definiti dalla sintassi, ne permette una interpretazione semantica. Il parallelismo linguistico porta spontaneamente ad indicare i simboli trasmessi come *alfabeto*, gruppi di simboli come *parole*, e gruppi di parole come *frasi*, od in alternativa, *trame* (FRAME, ovvero *telaio*).

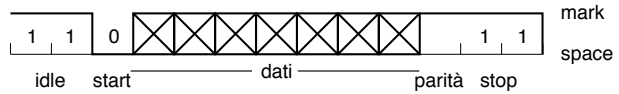
<sup>92</sup>In appendice 15.8.4 è riportata la codifica in termini di sequenze binarie dei caratteri stampabili, definita dallo standard ASCII; al § 24.3.1 si mostra la struttura della *trama PCM*, che trasporta i campioni di più sorgenti analogiche campionate.

nell'ambito di due diverse tecniche di trasmissione, indicate come *asincrona* e *sincrona*, che si differenziano per il fatto che mentre nella prima le parole sono separate tra loro, nella seconda fluiscono senza interruzione.

### 15.7.1 Trasmissione asincrona

Viene adottata, ad esempio, nel caso di un terminale *stupido*<sup>93</sup> collegato ad un computer *centrale*: la trasmissione in questo caso avviene in modo discontinuo, ossia quando l'operatore *digita* sui tasti del terminale, e per questo la modalità di trasmissione è indicata come *asincrona*. In tal caso, la linea di comunicazione permane abitualmente in uno stato di libero (IDLE), contraddistinta da uno stato di tensione positiva, indicato anche come stato *mark*<sup>94</sup>.

Quando è pronto un carattere da trasmettere, il segnale viene portato nello stato *zero* (detto SPACE) per la durata di 1 simbolo, che prende il nome di *bit di start*: la transizione *in discesa* viene rilevata dal ricevitore, che si predispone a contare un numero fisso di simboli (7 in figura, indicati con una croce a significare la loro variabilità) basandosi su di un suo orologio indipendente. Segue poi un bit di *parità* (se prevista, vedi § 15.6.3.1) ed uno o due di *stop* (realizzati come MARK), presenti per assicurare una durata minima dello stato di IDLE, prima della trasmissione del carattere successivo.



Il vantaggio di una simile modalità operativa è che il ricevitore non ha bisogno di generare con estrema esattezza la temporizzazione del segnale entrante; si basa infatti su di un proprio orologio locale, di precisione non elevata<sup>95</sup>, che viene *risvegliato* in corrispondenza del bit di start. Tale semplicità operativa causa una inefficienza, in quanto oltre ai dati ed al bit di parità, si introduce anche lo start e lo stop, utili solo ai fini della sincronizzazione ma privi di contenuto informativo.

In fig. 15.16 è mostrato uno schema di funzionamento per trasmettitore e ricevitore, in cui le parole *entrano* in modo parallelo nel trasmettitore, che le *incapsula* tra un bit di start e due di stop, disponendo la parola dal *less* al *most significant bit*. Il risultato viene quindi prelevato in modalità seriale e da questo generato il segnale dati a velocità  $f_s = T_{xC}$ , in cui  $T_{xC} = T_{xClock}$  è la frequenza (CLOCK) di *trasmissione* ( $T_x$ ).

**Sincronizzazione di bit e di parola** Il ricevitore dispone di un orologio interno operante ad un ritmo  $R_{xC}$  ( $R_{xClock}$ ) multiplo di quello di trasmissione ovvero  $R_{xC} = N \cdot T_{xC}$ , che *decrementa* un contatore, il quale viene inizializzato con il valore  $N/2$

<sup>93</sup>Un DUMB TERMINAL non ha capacità di calcolo, e provvede solo alla visualizzazione di informazioni testuali. Fino agli anni '70, è stato l'unico meccanismo di interazione (comunque migliore delle schede perforate !!!) con un computer. Per lungo tempo ogni computer disponeva di interfacce seriali RS-232 che possono funzionare sia in modalità asincrona che sincrona, oggi soppiantate dalle prese USB.

<sup>94</sup>In tal caso la linea “.. IS MARKING TIME” (sta *marcando* il tempo).

<sup>95</sup>Ovviamente, occorre stabilire un accordo a priori a riguardo la velocità, ossia della frequenza, sia pure approssimata, della trasmissione.

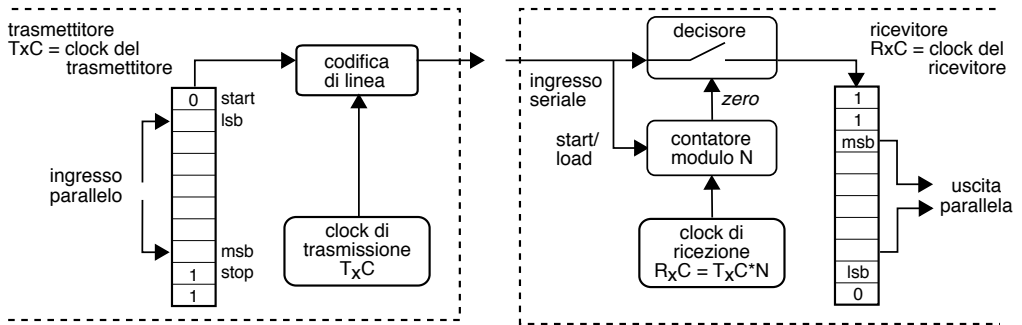
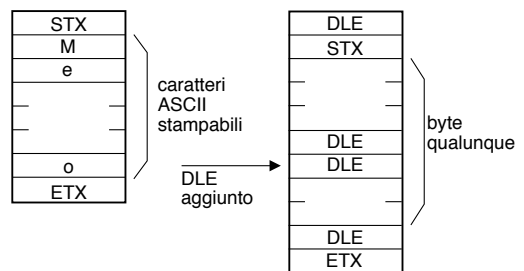


Figura 15.16: Trasmissione asincrona

in corrispondenza del fronte di *discesa* del bit di start.<sup>96</sup> Il contatore quindi si azzerava dopo  $NR_{xc}/2$  periodi di  $R_{xc}$ , ovvero a metà del bit di start, determinando l'istante in cui campionare il segnale da parte del decisore, l'esito della cui decisione viene accodato in un registro di ricezione.

Da quel punto in poi il contatore viene inizializzato con il valore  $N$ , in modo che le successive *letture* del segnale di ingresso cadano sempre a metà del periodo di bit, fino alla ricezione di una intera parola. Al suo termine (segnalato dall'arrivo in prima posizione del bit di start a zero), il registro di ricezione è letto in modo parallelo, recuperando una intera parola.

**Sincronizzazione di trama** I caratteri trasmessi possono far parte di messaggi più estesi, come ad esempio i paragrafi di un *file* di testo. Per questo motivo, può essere necessario inserire dei caratteri speciali tra quelli trasmessi, con lo scopo di delimitare i gruppi di caratteri che appartengono ad uno stesso messaggio. Se le diverse parole da trasmettere non sono tutte quelle possibili in base alla lunghezza di parola adottata<sup>97</sup>, la delimitazione può essere attuata mediante l'uso di caratteri speciali (di controllo) che non compaiono nel messaggio, come ad esempio i caratteri STX (*Start of Text*) e ETX (*End of Text*) dell'insieme ASCII (pag. 490), come mostrato nella figura precedente.



Se invece le parole trasmissibili sono qualsiasi, come nel caso della trasmissione di campioni di segnale, allora il carattere speciale ETX potrebbe essere *simulato* dai dati trasmessi<sup>98</sup>, causando un *troncamento* prematuro del messaggio. In tal caso, sia

<sup>96</sup>Per accorgersi di questa e delle altre transizioni, il ricevitore può ad esempio sfruttare un circuito che approssimi l'operazione di derivata, di cui constatare il superamento di una soglia.

<sup>97</sup>Una parola di  $M$  bit descrive uno spazio di  $2^M$  diversi elementi. Se le parole trasmissibili non sono tutte le  $2^M$  possibili, alcune di queste (che non compariranno mai all'interno del messaggio) possono essere usate per la sua delimitazione.

<sup>98</sup>Cioè, i dati trasmessi, che ora riempiono tutto lo spazio delle configurazioni possibili, contengono al loro interno la configurazione che è propria del carattere ETX.

STX che ETX vengono fatti precedere da un terzo carattere speciale, il DLE (*Data Link Escape*). Il trasmettitore, dopo aver inserito la coppia DLE-STX iniziale, ispeziona ogni carattere da inviare, e se questo *simula* un DLE, *inserisce* un secondo DLE, attuando una procedura detta CHARACTER (o BYTE) STUFFING. Il ricevitore a sua volta, per ogni DLE ricevuto, controlla se la parola successiva è un ETX, nel qual caso considera terminata la trasmissione; altrimenti, controlla se è un secondo DLE, che è stato inserito dal trasmettitore, e lo rimuove. Altri casi non sono possibili, e se si verificano, rivelano la presenza di un errore di trasmissione.

## 15.7.2 Trasmissione sincrona

La trasmissione dei bit di start e di stop necessaria per effettuare una trasmissione asincrona è fonte di inefficienza, e per questo a velocità più elevate si preferisce non inframmezzare i dati da trasmettere con delimitatori aggiuntivi. Ciò comporta l'esigenza di adottare in ricezione soluzioni apposite per individuare gli istanti di decisione corretti, e quindi conseguire il sincronismo di simbolo. Il successivo sincronismo di parola si basa in generale sull'uso di parole di lunghezza costante, mentre quello di trama prevede due possibili soluzioni, l'una orientata al carattere, e l'altra al bit.

### 15.7.2.1 Sincronizzazione di simbolo

La figura 15.17-a mostra uno schema idoneo ad estrarre la temporizzazione RxC dal segnale ricevuto, basata sull'uso di un circuito DPLL (*Digital Phase Locked Loop*), il cui funzionamento richiede la presenza di *transizioni* nel segnale ricevuto. Analogamente allo schema già analizzato nel caso di trasmissione asincrona, un orologio locale opera ad una frequenza  $N$  volte più elevata di quella nominale, e il DPLL (fig. 15.17-b) ne divide l'orologio per  $N$ , fornendo il segnale RxC necessario al decisore per individuare gli istanti posti al centro di un intervallo di simbolo. La divisione per  $N$  è realizzata all'interno del DPLL mediante un contatore *all'indietro*, che al suo azzeramento produce il segnale RxC di sincronismo, e la ricarica del contatore con la costante  $N$ . Nel caso in cui si verifichi uno *slittamento di fase* tra il segnale ricevuto e l'orologio locale, questo può essere rilevato osservando che la transizione (quando presente) nel segnale non ricorre nella posizione presunta, ossia a metà del conteggio, ma in anticipo od in ritardo (fig. 15.17-c). In tal caso, il contatore che realizza la divisione viene inizializzato con un numero rispettivamente minore o maggiore di  $N$ , in modo che il successivo impulso di sincronismo RxC risulti spostato verso il centro del periodo di simbolo<sup>99</sup>.

Nel caso di una differenza di velocità tra l'orologio di ricezione ed il ritmo di segnalazione le correzioni avvengono di rado, e sono di entità ridotta. Al contrario

<sup>99</sup>In termini generali, questo circuito è assimilabile ad un circuito di controllo, in quanto il suo principio di funzionamento si basa sul tentativo di *azzerare* una grandezza di errore. Infatti, la sincronizzazione dell'orologio del decisore di ricezione con il periodo di simbolo del segnale ricevuto avviene effettuando un confronto tra la *velocità* dell'orologio locale ed un *ritmo* presente nel segnale in arrivo: questo segnale di errore alimenta quindi un circuito di controreazione, che mantiene il clock locale *al passo* con quello dei dati in arrivo. Un diverso caso particolare di questo stesso principio è analizzato al § 16.11.1, ed anche ai § 12.2.2.2 e 12.3.2.1 a proposito del PLL.

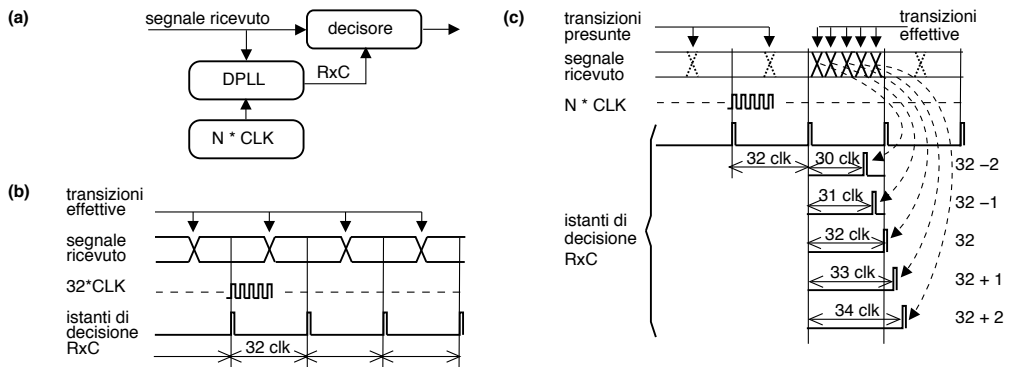


Figura 15.17: Funzionamento del DPLL: (a) schema circuitale; (b) condizione di sincronismo; (c) correzione di fase

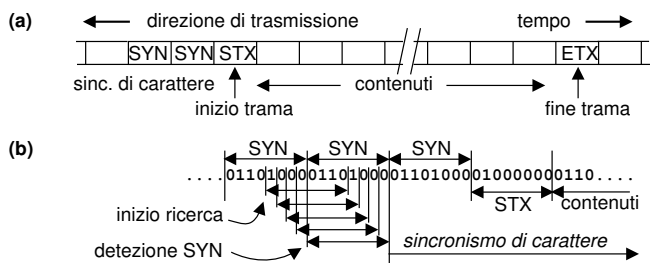
all’inizio di una trasmissione la differenza di fase può essere qualsiasi: per questo motivo prima dei dati veri e propri viene trasmessa una sequenza di dati fittizi o *trailer*, allo scopo di permettere l’acquisizione del sincronismo di simbolo. La durata del trailer dipende dalla velocità di convergenza della procedura, in cui sono imposte correzioni di maggiori entità per errori di fase più elevati.

### 15.7.2.2 Sincronizzazione di parola e di trama

Distinguiamo due casi.

**Trasmissione orientata al carattere** La trasmissione orientata al carattere è usata principalmente nel caso di contenuti testuali, come per i file ASCII. In assenza dei bit di start e di stop, la sincronizzazione di carattere è ottenuta per mezzo della trasmissione, prima dei dati veri e propri, di una sequenza di caratteri SYN (*Synchronous Idle*), che permette sia di conseguire (o mantenere) il sincronismo di bit, che di consentire l’individuazione dei confini di carattere, e quindi il sincronismo di carattere.

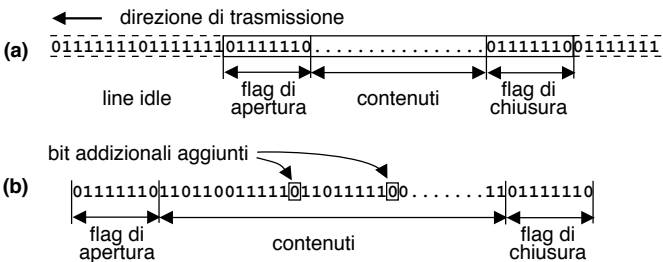
La figura a lato mostra (a) che la sincronizzazione di trama è ottenuta come per il caso asincrono, racchiudendo il blocco da trasmettere entro una coppia di caratteri STX-ETX. Una volta che il ricevitore ha conseguito il sincronismo di bit, passa in una modalità di ricerca, verificando (b) se l’allineamento di 8 bit consecutivi corrisponde al carattere SYN, ed in caso negativo, ripete il tentativo bit a bit.



Dopo aver individuato il SYN, il ricevitore consegue l’allineamento sul carattere, ed inizia ad attendere il carattere STX, che indica l’inizio della trama, la quale è terminata da un ETX. Nel caso in cui la trasmissione contenga caratteri qualunque, e dunque l’ETX possa essere simulato dai dati, si ricorre alla stessa soluzione del caso asincrono,

e cioè sia l'STX che l'ETX vengono fatti precedere da un DLE, ed all'interno dei dati si esegue il *byte stuffing*, sostituendo gli eventuali DLE simulati con una coppia di DLE.

**Trasmissione orientata al bit** Questa tecnica viene preferita sia nel caso in cui i dati da trasmettere non siano organizzati in caratteri, sia per ridurre l'inefficienza legata all'uso di caratteri di controllo aggiuntivi, nonché per evitare la dipendenza da quei particolari caratteri, così da trasportare l'informazione in modo più universale e *trasparente*. Nella trasmissione orientata al bit, la sincronizzazione di bit e di trama non impiega i caratteri SYN e STX, bensì degli *idle bytes* 01111111 nei periodi di inattività, e dei *flag byte* 01111110 per indicare sia l'inizio che la fine di una trama.



La figura a lato mostra in (a) un esempio di trama, ed in (b) la soluzione del *bit stuffing*, necessaria ad evitare che il *flag byte* possa essere simulato dal contenuto della trasmissione. Ora i dati trasmessi non devono essere necessariamente in numero multiplo della lunghezza di carattere, ed ogni qualvolta sono presenti 5 bit pari ad uno consecutivi, il trasmettitore inserisce forzatamente un bit pari a zero. Viceversa, quando il ricevitore osserva un bit pari a 0 preceduto da 5 bit pari ad uno consecutivi, lo rimuove, conseguendo così la *trasparenza dai dati*, e permettendo il corretto rilevamento del flag byte di fine trama. Ovviamente, la procedura di bit stuffing/destuffing è applicata solamente al *contenuto* della trama.

La procedura ora descritta può fallire qualora applicata ad un flusso binario affetto da errori, ed in tal caso occorre attendere la successiva opportunità di sincronizzazione. Tale problematica può essere risolta evitando la necessità del *flag byte* finale grazie alla presenza di un campo *lunghezza* nella intestazione di pacchetto, come avviene ad es. per la trama *ethernet*, vedi § 23.1.4.2.

La procedura ora descritta può fallire qualora applicata ad un flusso binario affetto da errori, ed in tal caso occorre attendere la successiva opportunità di sincronizzazione. Tale problematica può essere risolta evitando la necessità del *flag byte* finale grazie alla presenza di un campo *lunghezza* nella intestazione di pacchetto, come avviene ad es. per la trama *ethernet*, vedi § 23.1.4.2.

## 15.8 Appendici

### 15.8.1 Potenza di un segnale dati

Al § 15.4.4 si è affermato che ad un segnale dati

$$s(t) = \sum_n a[n] g(t - nT)$$

(in cui  $g(t)$  è la risposta impulsiva di un filtro a coseno rialzato con roll-off  $\gamma$ , e  $a[n]$  è una sequenza di v.a. discrete, statisticamente indipendenti, a media nulla ed uniformemente distribuite su L valori in una dinamica  $-\frac{\Delta}{2} \leq a_i \leq \frac{\Delta}{2}$ ) corrisponde una potenza

$$\mathcal{P}_s = \frac{\Delta^2 L + 1}{12 L - 1} \left(1 - \frac{\gamma}{4}\right) \tag{15.31}$$

Svolgiamo qui i passi necessari per arrivare al risultato (15.31). Al § 7.7.4 si è mostrato che per lo stesso segnale risulta  $\mathcal{P}_s(f) = \sigma_A^2 \frac{|G(f)|^2}{T}$ , e dunque

$$\mathcal{P}_s = \int \mathcal{P}_s(f) df = \int \sigma_A^2 \frac{|G(f)|^2}{T} df$$

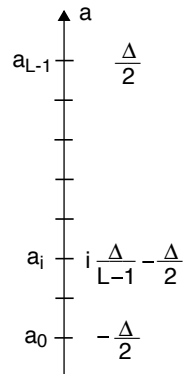
Si può mostrare<sup>100</sup> che

$$\int |G(f)|^2 df = T \left(1 - \frac{\gamma}{4}\right) \quad (15.32)$$

e quindi  $\mathcal{P}_s = \sigma_A^2 \left(1 - \frac{\gamma}{4}\right)$ ; resta pertanto da calcolare

$$\sigma_A^2 = E_A \{(a - m_A)^2\} = E_A \{a^2\}$$

in virtù della media nulla. I valori che la v.a.  $a$  può assumere corrispondono a quelli dei diversi livelli del segnale dati, esprimibili come  $a_i = i \cdot \frac{\Delta}{L-1} - \frac{\Delta}{2}$  (vedi figura), in modo che per  $i = 0, 1, \dots, L-1$  corrispondano ad  $L$  valori uniformemente spaziatati entro l'intervallo  $(-\frac{\Delta}{2}, \frac{\Delta}{2})$ ; inoltre, ogni possibile valore  $a_i$  ricorre con probabilità  $p(a_i) = p_i = 1/L$ . Possiamo dunque sviluppare i conti:



$$\begin{aligned} \sigma_A^2 &= E_A \{a_i^2\} = \sum_{i=0}^{L-1} p_i \cdot a_i^2 = \\ &= \frac{1}{L} \sum_{i=0}^{L-1} \left(i \frac{\Delta}{L-1} - \frac{\Delta}{2}\right)^2 = \frac{\Delta^2}{L} \sum_{i=0}^{L-1} \left(\frac{i^2}{(L-1)^2} + \frac{1}{4} - \frac{i}{L-1}\right) = \\ &= \frac{\Delta^2}{L} \left(\frac{1}{(L-1)^2} \sum_{i=0}^{L-1} (i)^2 + \frac{L}{4} - \frac{1}{L-1} \sum_{i=0}^{L-1} i\right) = {}^{(101)} = \\ &= \frac{\Delta^2}{L} \left(\frac{L}{4} - \frac{1}{L-1} \frac{L(L-1)}{2} + \frac{1}{(L-1)^2} \frac{(L-1)L(2(L-1)+1)}{6}\right) = \Delta^2 \left(\frac{1}{4} - \frac{1}{2} + \frac{2L-2+1}{6(L-1)}\right) = \\ &= \Delta^2 \frac{6L-6-12L+12+8L-8+4}{24(L-1)} = \Delta^2 \frac{2L+2}{24(L-1)} = \frac{\Delta^2}{12} \frac{L+1}{L-1}. \end{aligned}$$

### 15.8.2 Prestazioni del ricevitore ottimo equalizzato

Mostriamo quanto affermato al § 15.5.1, ovvero che in presenza di un canale non perfetto con risposta in frequenza  $H(f) \neq ae^{-j2\pi f\tau}$  le prestazioni del ricevitore ottimo *equalizzato al trasmettitore*, in cui cioè

$$H_T(f) = \sqrt{G(f)/H(f)} \quad \text{e} \quad H_R(f) = \sqrt{G(f)}$$

(vedi fig. 15.18-a), subiscono una degradazione del rapporto  $E_b/N_0$  (a parità di parametri di sistema, § 15.4.4) valutabile come una perdita di  $10 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|^2} df$  dB.

Per semplificare l'analisi consideriamo una segnalazione binaria antipodale in cui il segnale trasmesso ha espressione

$$x(t) = \sum_k a_k \cdot h_T(t - kT_b) \quad \text{con} \quad a_k = \{+d, -d\} \quad \text{equiprobabili}$$

<sup>100</sup>La dimostrazione sarà sperabilmente sviluppata in una prossima edizione... è una delle poche a mancare in questo libro! ..al momento, la fonte che trovo più in accordo con questa tesi, è ancora una volta [https://en.wikipedia.org/wiki/Raised-cosine\\_filter](https://en.wikipedia.org/wiki/Raised-cosine_filter)

<sup>101</sup>Facciamo uso delle relazioni  $\sum_{n=1}^N n = \frac{N(N+1)}{2}$  e  $\sum_{n=1}^N n^2 = \frac{N(N+1)(2N+1)}{6}$ , che sono ovviamente ancora valide anche qualora la somma parta da  $n=0$ .

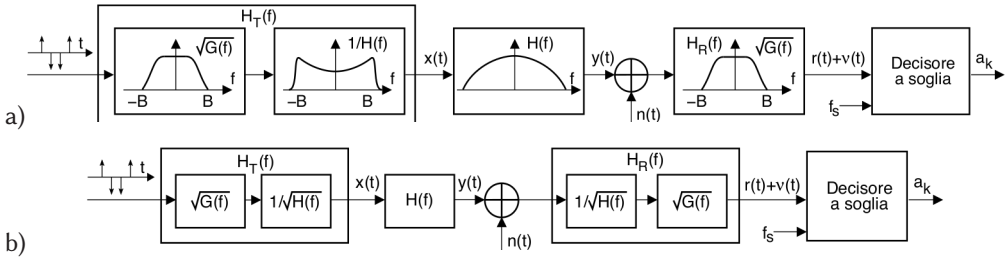


Figura 15.18: Ricevitore ottimo equalizzato: a) - al trasmettitore; b) - ripartito

ed  $h_T(t) = \mathcal{F}^{-1}\{\sqrt{G(f)}/H(f)\}$  è l'impulso a radice di coseno rialzato equalizzato, mentre il segnale in ingresso al decisore è

$$r(t) + v(t) = \sum_k a_k \cdot g(t - kT_b) + v(t)$$

in cui  $g(t)$  è l'impulso a coseno rialzato e  $v(t)$  è un rumore gaussiano bianco a media nulla, filtrato attraverso  $H_R(t)$ , dunque con potenza

$$\mathcal{P}_v = \int \mathcal{P}_v(f) df = \frac{N_0}{2} \int |H_R(f)|^2 df = \frac{N_0}{2} \int G(f) df = \frac{N_0}{2} = \sigma_v^2$$

in quanto  $\int G(f) df = 1$ , vedi nota 58 ma considerando qui  $G(f)|_{f=0} = 1/f_s = T_b$ .

In tali condizioni agli istanti di campionamento il decisore osserva una v.a. gaussiana con media  $\pm d$  (a seconda dell' $a_k$  per quel simbolo) e varianza  $\sigma_v^2$ , e dunque commette errore quando  $v > d$  o  $v < -d$  (a seconda di  $a_k$ , vedi fig. al § 7.6.1) ovvero con probabilità (vedi eq. (6.11) e § 7.6.1)  $P_e = \frac{1}{2} \operatorname{erfc}\left\{\frac{d}{\sqrt{2\sigma_v}}\right\}$ , tanto minore quanto maggiore è il rapporto  $R = d^2/2\sigma_v^2$ : esprimiamo dunque quest'ultimo in funzione della potenza trasmessa  $\mathcal{P}_x = \int \mathcal{P}_x(f) df$ . Sapendo che  $\mathcal{P}_x(f) = \sigma_a^2 \cdot \frac{\mathcal{E}_{h_T}(f)}{T_b}$  (eq. (15.2)) dove  $\sigma_a^2 = d^2$  e  $\mathcal{E}_{h_T}(f) = G(f)/|H(f)|^2$  troviamo che la potenza trasmessa vale

$$\mathcal{P}_x = \int \mathcal{P}_x(f) df = \frac{d^2}{T_b} \int_{-B}^B \frac{G(f)}{|H(f)|^2} df$$

da cui otteniamo  $d^2 = \frac{\mathcal{P}_x T_b}{\int_{-B}^B \frac{G(f)}{|H(f)|^2} df}$ , potendo così scrivere

$$R = \frac{d^2}{2\sigma_v^2} = \frac{\mathcal{P}_x T_b}{2 \int_{-B}^B \frac{G(f)}{|H(f)|^2} df \cdot N_0/2} = \frac{E_b}{N_0} \frac{1}{\int_{-B}^B \frac{G(f)}{|H(f)|^2} df}$$

In assenza di distorsione lineare si ha<sup>103</sup>  $|H(f)| = 1$  e dunque il denominatore vale  $1^{104}$ , riottenendo il risultato noto (7.27) per un filtro adattato con segnalazione antipodale. Se invece  $|H(f)|$  non è costante (ovvero  $|H(f)| \leq 1$ ) il denominatore è più grande; pertanto la presenza del canale  $H(f)$  determina la riduzione del rapporto  $E_b/N_0$  di un fattore  $\alpha = \int_{-B}^B \frac{G(f)}{|H(f)|^2} df$ , ovvero della quantità  $\alpha_{dB}$  (15.25) qualora il rapporto sia espresso in dB.

<sup>102</sup>  $\sigma_a^2 = E\{(a_k)^2\} = \frac{1}{2}d^2 + \frac{1}{2}(-d)^2 = d^2$

<sup>103</sup> A meno di un valore costante, ininfluenza ai fini della valutazione che stiamo svolgendo.

<sup>104</sup> Sempre in quanto  $\int G(f) df = 1$

### 15.8.2.1 Equalizzazione distribuita

Qualora si realizzi invece

$$H_T(f) = H_R(f) = \sqrt{G(f)/H(f)}$$

in modo da ripartire l'equalizzazione in parti uguali ad entrambi i lati del collegamento come mostrato in fig. 15.18-b), la densità di energia dell'impulso usato in trasmissione vale  $\mathcal{E}_{h_T}(f) = G(f)/|H(f)|$  e dunque il segnale trasmesso  $x(t)$  ha una potenza

$$\mathcal{P}_x = \int \sigma_a^2 \cdot \frac{\mathcal{E}_{h_T}(f)}{T_b} df = \frac{d^2}{T_b} \int_{-B}^B \frac{G(f)}{|H(f)|} df$$

fornendo così  $d^2 = \frac{\mathcal{P}_x T_b}{\int_{-B}^B \frac{G(f)}{|H(f)|} df}$ , mentre ora la potenza del rumore filtrato attraverso

$H_R(f)$  risulta

$$\mathcal{P}_v = \int \mathcal{P}_v(f) df = \frac{N_0}{2} \int |H_R(f)|^2 df = \frac{N_0}{2} \int_{-B}^B \frac{G(f)}{|H(f)|} df = \sigma_v^2$$

da cui

$$R = \frac{d^2}{2\sigma_v^2} = \frac{\mathcal{P}_x T_b}{2 \int_{-B}^B \frac{G(f)}{|H(f)|} df \cdot \frac{N_0}{2} \int \frac{G(f)}{|H(f)|} df} = \frac{E_b}{N_0} \frac{1}{\left| \int_{-B}^B \frac{G(f)}{|H(f)|} df \right|^2}$$

che qualora  $|H(f)| \leq 1$ , determina la riduzione del rapporto  $\frac{E_b}{N_0}$  per un fattore  $\beta$  pari a  $\left| \int_{-B}^B \frac{G(f)}{|H(f)|} df \right|^2$  ovvero un suo decremento in dB di  $\beta_{dB} = 20 \log_{10} \int_{-B}^B \frac{G(f)}{|H(f)|} df$ .

Mostriamo infine che  $\beta \leq \alpha$ : ricordando infatti la disuguaglianza di Schwartz (pag. 55)

$$\left| \int_{-\infty}^{\infty} X(f) Y(f) df \right|^2 \leq \int_{-\infty}^{\infty} |X(f)|^2 df \cdot \int_{-\infty}^{\infty} |Y(f)|^2 df$$

ed identificando  $X(f)$  con  $\sqrt{G(f)}$  e  $Y(f)$  con  $\sqrt{G(f)}/|H(f)|$  troviamo che

$$\left| \int_{-\infty}^{\infty} \sqrt{G(f)} \frac{\sqrt{G(f)}}{|H(f)|} df \right|^2 = \left| \int_{-\infty}^{\infty} \frac{G(f)}{|H(f)|} df \right|^2 \leq \int_{-\infty}^{\infty} G(f) df \cdot \int_{-\infty}^{\infty} \frac{G(f)}{|H(f)|^2} df = \int_{-\infty}^{\infty} \frac{G(f)}{|H(f)|^2} df$$

di nuovo in quanto  $\int_{-\infty}^{\infty} G(f) df = 1$ , e quindi

$$\beta = \left| \int_{-\infty}^{\infty} \frac{G(f)}{|H(f)|} df \right|^2 \leq \int_{-\infty}^{\infty} \frac{G(f)}{|H(f)|^2} df = \alpha$$

con l'uguaglianza valida qualora  $\sqrt{G(f)} = k \frac{\sqrt{G(f)}}{|H(f)|}$ , ovvero quando  $|H(f)| = 1/k =$  costante.

### 15.8.2.2 Discussione

A prima vista le migliori prestazioni dell'equalizzazione distribuita rispetto a quella localizzata al trasmettitore appaiono contraddittorie, vista la correlazione dei campioni del rumore e la mancata realizzazione delle condizioni di filtro adattato nel primo caso, in quanto l'impulso che giunge al ricevitore ha densità spettrale

$$H_T(f) H(f) = \sqrt{G(f) H(f)} \neq H_R(f) = \sqrt{G(f)/H(f)}$$

In realtà quest'ultima affermazione non è vera, perché nulla vieta di considerare (vedi fig. 15.18-b)  $H_R(f) = \sqrt{G(f)}$  ed inglobare il termine  $1/\sqrt{H(f)}$  assieme al canale ed all'altra mezza equalizzazione, per riottenere un filtro adattato. A mio avviso le migliori prestazioni rispetto all'equalizzazione al trasmettitore sembrano essere dovute ad una minore "distorsione totale" subita dal segnale in transito, in quanto anziché invertire *per intero*  $H(f)$  per poi subirne la distorsione sempre per intero, nel caso distribuito la massima distorsione in ogni singolo passo è limitata a  $\sqrt{H(f)}$  od al suo inverso. Mentre per quanto riguarda la correlazione dei campioni di rumore al decisore, questo significa solamente che è possibile fare *ancora di meglio*, vedi la nota 56 a pag. 462, così come il § 18.4.4.

### 15.8.3 Esercizio

Un sistema di trasmissione basato sul campionamento e sulla trasmissione numerica è rappresentato in figura 15.19. Il canale riportato all'estremità destra è considerato ideale entro una banda  $\pm B = \pm 31.5$  KHz, purché la potenza al suo ingresso non superi il valore  $\mathcal{P}_y^{Max} = 1$  Volt<sup>2</sup>; in tal caso la potenza in uscita risulta  $\mathcal{P}_{y'} = 0.01 \cdot \mathcal{P}_y$ . Al segnale ricevuto è sovrapposto un rumore additivo gaussiano bianco stazionario ergodico a media nulla, con spettro di densità di potenza  $\mathcal{P}_N(f) = \frac{N_0}{2} = 4.61 \cdot 10^{-14}$  Volt<sup>2</sup>/Hz, e limitato nella banda  $\pm B$ .

- 1) Se  $G(f)$  è a coseno rialzato con  $\gamma = .5$ , determinare la massima frequenza di simbolo  $f_s = \frac{1}{T_s}$ .
- 2) Desiderando una  $P_e = P_e^c$  per la sequenza  $\{c'\}$  pari a  $P_e = 10^{-4}$ , determinare il massimo numero di livelli/simbolo  $L$ .
- 3) Indicare la frequenza binaria  $f_b$  per la sequenza  $\{b'\}$ .
- 4) Valutare  $P_e^b$  per la sequenza  $\{b'\}$  e mostrare che il numero di errori per unità di tempo in  $\{b'\}$  è lo stesso che in  $\{c'\}$ .
- 5) Mostrare che, adottando una codifica di canale a ripetizione 3 : 1, la probabilità di errore  $P_e^a$  per la sequenza  $\{a'\}$  risulta pari a circa  $P_e^a \approx 3 (P_e^b)^2$ .
- 6) Indicare la frequenza binaria  $f_a$  per le sequenze  $\{a\}$  ed  $\{a'\}$ .

Supponiamo ora che  $P_e^a = 0$ , e si desideri un  $SNR_Q = \mathcal{P}_x / \mathcal{P}_{z-x} = 10000$ . Nel caso in cui  $x(t)$  sia un processo con densità di probabilità  $p(x)$  uniforme, ed indicando con  $W$  la banda di  $x(t)$ ;

- 7) Determinare il minimo numero di bit/campione  $M$ .
- 8) Determinare la massima banda  $W$ .
- 9) Se la banda è ridotta a  $W' = \frac{1}{2}W$ , determinare il nuovo valore di  $SNR_Q$  ottenibile.

### Svolgimento

- 1) La banda  $B$  occupata dal segnale  $y$  vale  $B = \frac{f_s}{2} (1 + \gamma)$ , e quindi deve risultare  $f_s = \frac{2B}{1+\gamma} = \frac{2 \cdot 31.5 \cdot 10^3}{1.5} = 42 \cdot 10^3 = 42.000$  baud (*baud = simboli/secondo*).

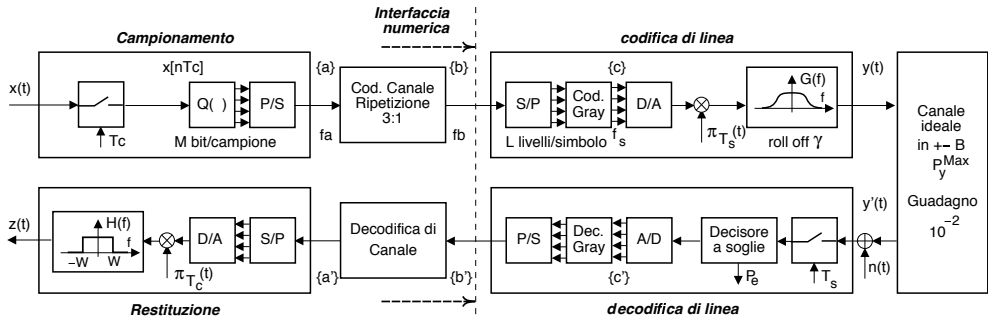


Figura 15.19: Sistema di trasmissione a cui si riferisce l'esercizio

- 2) Osserviamo che in questo caso la (15.17) non può essere applicata direttamente, in quanto non essendo ancora nota la  $f_b$ , non è possibile calcolare il valore di  $E_b = \frac{\mathcal{P}_{y'}}{f_b}$ . Notiamo però che essendo  $f_b = f_s \cdot \log_2 L$ , indicando con  $y$  l'argomento dell' $erfc \{ \cdot \}$ , questo può essere riscritto come

$$y = \sqrt{\frac{E_b}{N_0} \frac{3 \log_2 L}{(L^2 - 1)(1 + \gamma)(1 - \frac{\gamma}{4})}} = \sqrt{\frac{\mathcal{P}_{y'}}{f_s \cdot \log_2 L} \frac{1}{N_0} \frac{3 \cdot \log_2 L}{(L^2 - 1) 1.31}}$$

$$= \sqrt{\frac{\mathcal{P}_{y'}}{f_s \cdot N_0} \cdot \frac{2.29}{L^2 - 1}}$$

avendo tenuto conto che se  $\gamma = 0.5$ , allora  $(1 + \gamma)(1 - \frac{\gamma}{4}) \approx 1.31$ . Inoltre, se  $L \gg 1$  (come verificheremo), la (15.17) può essere approssimata come  $P_e \approx erfc \{y\}$ , e dunque per  $P_e = 10^{-4}$  la figura di pag. 154 ci permette di individuare il valore di  $y \approx 2.7$ , e pertanto

$$\frac{\mathcal{P}_{y'}}{f_s \cdot N_0} \cdot \frac{2.29}{L^2 - 1} = y^2 = (2.7)^2 = 7.29$$

e, conoscendo i valori di  $f_s$ ,  $\mathcal{P}_{y'}$  e  $N_0$ , scriviamo

$$L^2 = 1 + \frac{\mathcal{P}_{y'}}{f_s \cdot N_0} \cdot \frac{2.29}{7.29} = 1 + \frac{10^{-2}}{42 \cdot 10^{-3} \cdot 4.61 \cdot 10^{-4}} \cdot 0.31$$

$$= 1 + 5.16 \cdot 10^6 \cdot 0.31 \approx 1.6 \cdot 10^6$$

e quindi  $L = \sqrt{1.6 \cdot 10^6} = 1265$  che, essendo un valore massimo, limitiamo a  $L = 1024$  livelli

- 3) Dato che ad ogni simbolo di  $\{c\}$  ad  $L$  livelli, con frequenza di emissione pari a  $f_s$ , corrisponde ad un gruppo di  $N_b = \log_2 L = 10$  bit della sequenza  $\{b\}$ , la frequenza  $f_b$  è di 10 volte  $f_s$ , e quindi  $f_b = 10 \cdot f_s = 10 \cdot 42 \cdot 10^3 = 420$  Kbps.
- 4) Grazie all'adozione del codice di Gray, in caso di errore tra livelli contigui per i simboli di  $\{c'\}$ , nella sequenza  $\{b'\}$  solo uno (tra  $N_b$ ) dei bit associati ad un simbolo è errato; il bit errato è uno qualsiasi del gruppo di  $N_b$ , e pertanto la probabilità che un bit specifico sia errato (quando è errato il simbolo di  $\{c'\}$ ) è  $\frac{1}{N_b}$ . Pertanto  $P_e^b = P_e^{b/c} P_e^c = \frac{1}{N_b} P_e^c$ , in cui  $P_e^{b/c}$  è la probabilità condizionata che un generico bit di  $\{b'\}$  sia sbagliato quando è sbagliato il simbolo di  $\{c'\}$  da cui ha origine.

- Il numero di bit (della sequenza  $\{b'\}$ ) errati per unità di tempo è dato da  $P_e^b \cdot f_b$ ; sostituendo:  $P_e^b \cdot f_b = \frac{P_e^c}{N_b} \cdot f_b = P_e^c \cdot \frac{f_b}{N_b} = P_e^c \cdot f_s$ , ovvero è numericamente pari ai simboli errati (nella sequenza  $\{c'\}$ ) per unità di tempo;
- risulta dunque infine:
  - $P_e^b = \frac{P_e^c}{N_b} = \frac{10^{-4}}{10} = 10^{-5}$ ;
  - $P_e^b \cdot f_b = P_e^c \cdot f_s = 10^{-5} \cdot 420 \cdot 10^3 = 10^{-4} \cdot 42 \cdot 10^3 = 4.2 \frac{\text{errori}}{\text{secondo}}$

- 5) Ogni bit di  $\{a'\}$  è sbagliato solo se sono sbagliati 2 o più bit in un gruppo di 3; come mostrato al § 15.6.1.1, la probabilità di 2 bit errati su 3 è calcolabile dalla distribuzione di Bernoulli, e vale  $\binom{3}{2} p_e^2 (1 - p_e) = 3 p_e^2 (1 - p_e)$ , a cui va sommata la probabilità di 3 bit errati, pari a  $p_e^3$ . Pertanto  $p_e^a = 3 p_e^2 (1 - p_e) + p_e^3 = 3 p_e^2 - 3 p_e^3 + p_e^3 \approx 3 p_e^2$  in cui ovviamente  $p_e = P_e^b$ , e l'approssimazione è legittima in quanto se  $p_e = 10^{-5}$  allora  $p_e^2 = 10^{-10}$  e  $p_e^3 = 10^{-15}$ , trascurabili rispetto a  $p_e$ . Lo stesso risultato si ottiene osservando che 2 bit errati su 3 hanno probabilità  $p_e^2 (1 - p_e)$ , e questi possono essere scelti in tre modi diversi ( $1^o$  e  $2^o$ ,  $1^o$  e  $3^o$ ,  $2^o$  e  $3^o$ ). In definitiva, risulta  $P_e^a \approx 3 (P_e^b)^2 = 3 \cdot 10^{-10}$ .
- 6) Dato che ad ogni 3 bit di  $\{b'\}$  corrisponde un solo bit di  $\{a'\}$ , si ottiene  $f_a = \frac{f_b}{3} = \frac{420 \cdot 10^3}{3} = 140$  Kbps, a cui corrisponde  $P_e^a \cdot f_a = 3 \cdot 10^{-10} \cdot 140 \cdot 10^3 = 4.2 \cdot 10^{-5} \frac{\text{errori}}{\text{secondo}}$ .
- 7) Sappiamo che per un processo uniforme l'SNR di quantizzazione risulta approssimativamente  $SNR_q = (L - 1)^2$ , in cui  $L$  è il numero di livelli del quantizzatore, a cui corrisponde l'utilizzo di  $M = \log_2 L$  bit/campione. Risulta pertanto  $L = 1 + \sqrt{SNR_q} = 1 + \sqrt{10^4} = 101$  livelli. Per ottenere un numero intero di bit/campione ed un  $SNR_q$  migliore od uguale a quello desiderato, determiniamo l'intero superiore:  $M = \lceil \log_2 L \rceil = 7$  bit/campione (equivalente a 128 livelli).
- 8) Come sappiamo, la frequenza di campionamento  $f_c = \frac{1}{T_c}$  non può essere inferiore a  $2W$ ; inoltre, la frequenza binaria  $f_a$  risulta pari al prodotto dei bit/campione per i campioni a secondo:  $f_a = f_c \cdot M$ ; pertanto  $f_c = \frac{f_a}{M} = \frac{140 \cdot 10^3}{7} = 20$  KHz e dunque la  $W$  massima risulta  $W_{Max} = \frac{f_c}{2} = 10$  KHz.
- 9) Nel caso in cui  $W' = \frac{1}{2}W$ , allora si può dimezzare anche la frequenza di campionamento  $f'_c = \frac{f_c}{2} = 10$  KHz, e pertanto utilizzare un  $M' = 2M$  per ottenere la stessa  $f_a$ . Pertanto il nuovo  $SNR_q$  risulta  $SNR'_q = (L' - 1)^2 = (2^{2M} - 1)^2 = (2^{14} - 1)^2 \approx 2.68 \cdot 10^8$ , ovvero  $SNR'_q$  (dB) = 84.3 dB.

### 15.8.4 Codifica di carattere

Il codice ASCII (*American Standard Code for Information Interchange*) è un codice a 7 bit, e molti codici ad 8 bit (come l'ISO 8859-1) si riducono ad ASCII nella loro metà bassa (con il bit più significativo a zero); i primi 32 codici corrispondono a caratteri *non stampabili*, detti codici di controllo, ottenibili su di una tastiera mediante la pressione del tasto CONTROL, e che hanno un significato speciale, come il *carriage return* (CR), il *line feed* (LF), *start of text* (STX), *backspace* (BS), *data link escape* (DLE). La tavola 15.1 mostra i 128 caratteri ASCII. La controparte internazionale dell'ASCII è nota come ISO 646; lo standard è stato pubblicato dallo *United States of America Standards Institute* (USASI) nel 1968.

#### 15.8.4.1 Codifica UNICODE

Dal 2004 ISO/IEC non si occupa più della manutenzione delle codifiche di carattere ad 8 bit, supportando invece attivamente il consorzio UNICODE nella definizione dello *Universal Character Set*, che contiene centinaia di migliaia di caratteri di praticamente tutte le lingue del mondo, ognuno identificato in modo non ambiguo da un nome, e da un numero chiamato *Code Point*. Mentre per enumerare tutti i caratteri previsti occorre una parola di ben 21 bit, sono state definite codifiche a lunghezza variabile, la più diffusa delle quali prende il nome di UTF-8, in base alla quale

- i primi 127 CodePoints, che corrispondono all'alfabeto ASCII, sono rappresentati da un singolo byte; pertanto un file ASCII è anche un file UTF-8 corretto;
- i valori numerici associati ai caratteri dell'insieme ISO 8859-1 corrispondono ai CodePoints degli stessi caratteri;
- i primi 1792 CodePoints, mediante i quali sono rappresentati i caratteri usati dalla totalità delle lingue occidentali, sono rappresentati (esclusi gli ASCII) mediante due byte;
- i 65536 CodePoints del *Piano di Base* entro cui ricade la quasi totalità delle assegnazioni fatte finora, sono rappresentati (esclusi i casi precedenti) mediante tre byte;
- i restanti CodePoints sono rappresentati mediante quattro byte.

<i>dec</i>	<i>hex</i>	<i>char</i>	<i>dec</i>	<i>hex</i>	<i>char</i>	<i>dec</i>	<i>hex</i>	<i>char</i>	<i>dec</i>	<i>hex</i>	<i>char</i>
0	00	NUL	32	20		64	40	@	96	60	'
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	CD2	50	32	2	82	52	R	114	72	r
19	13	CD3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Tabella 15.1: Tabella di codici e caratteri ASCII

L'opera

# ***Trasmissione dei Segnali e Sistemi di Telecomunicazione***

è il risultato di un progetto ventennale di cultura libera, aggiornato di continuo ed evolutosi fino alla forma attuale. La sua disponibilità pubblica è regolata dalle norme di licenza CREATIVE COMMONS

*Attribuzione - Non commerciale -  
Condividi allo stesso modo*



<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

e tutte le risorse relative al testo sono accessibili presso

<https://teoriadeisignali.it/libro/>

Puoi contribuire al suo successo promuovendone la diffusione e supportarne lo sviluppo attraverso una donazione, in buona parte devoluta ai progetti *open source*<sup>1</sup> che ne hanno resa possibile realizzazione e divulgazione. Ai donatori viene accordato un accesso *vitalizio* al formato PDF *navigabile* di tutte le edizioni presenti *e future*.

---

1

- . Lyx - <http://www.lyx.org/>
- . L<sup>A</sup>T<sub>E</sub>X - <https://www.latex-project.org/>
- . TeX Users Group - <https://tug.org/>
- . Inkscape - <http://www.inkscape.org/>
- . Gnuplot - <http://www.gnuplot.info/>
- . Octave - <http://www.gnu.org/software/octave/>
- . Geany - <https://www.geany.org/>
- . Linux - <https://www.linux.it/>
- . Free Software Foundation - <https://shop.fsf.org/>
- . GNOME Foundation - <https://www.gnome.org/>
- . Mozilla Foundation - <https://www.mozilla.org/it/>
- . Wikipedia - <https://it.wikipedia.org>
- . Internet Archive - <https://archive.org/about/>
- . Creative Commons - <https://creativecommons.it/chapterIT/>
- . WordPress - <https://it.wordpress.org/>
- . Phplist - <https://www.phplist.org/>