

Orale

fondamenti di telecomunicazioni

Flavio Colacicchi

10-17/01/2024

sorgente \rightarrow *canale* \rightarrow *destinazione*

più nel dettaglio da parte della sorgente

$$s(t) \rightarrow \boxed{ADC} \rightarrow s[n] \rightarrow \boxed{\text{quantizzatore}}$$

La quantizzazione converte il segnale campionato in una sequenza di bit, assegnando ad esempio ai campioni che si trovano nello stesso intervallo di quantizzazione il valor medio tra i due anche se i due campioni differiscono, ad esempio se l'intervallo di quantizzazione per il bit 00 è $[0,1]$ sia 0,3 che 0,7 saranno quantizzati in 00. Questo introduce un errore di quantizzazione che è una variabile aleatoria

$$\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right] \ni e_q[n] = x[n] - x_q[n]$$

Prendendo M intervalli di campionamento per quantizzare il segnale in elementi da $K = \log_2 M$ bit il segnale che ha ampiezza $[-V, V]$ possiamo trovare l'ampiezza di un intervallo di quantizzazione

$$\Delta = \frac{2V}{M}$$

e da qui possiamo vedere che la sua densità di probabilità

$$p_E(e) = \frac{1}{\Delta} \text{rect}\left(\frac{e}{\Delta}\right)$$

Ovviamente all'aumentare di M si riduce $e_q[n]$ (ad esempio per la radio e la tv si usa $M = 16$). Si vede facilmente che e_q è uniformemente distribuita, quindi posso calcolarne la potenza

$$P_e = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 de = \frac{e^3}{3\Delta} \Big|_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = \frac{\Delta^2}{12}$$

E da questo ottengo

$$\sigma_E^2 = p_E(e) \qquad \mu_E = 0$$

Riscrivo la potenza in funzione di V e M

$$P_e = \frac{\Delta^2}{12} = \frac{4V^2}{12M^2} = \frac{V^2}{3M^2}$$

Se scrivo la potenza in scala logaritmica ottengo

$$(P_e)_{dB} = 10 \log_{10} \frac{V^3}{3} - 10 \log_{10} M^2 = 10 \log_{10} \frac{V^3}{3} - 10 \log_{10} 2^{2K} \stackrel{10 \log_{10} 2=3}{=} 10 \log_{10} \frac{V^3}{3} - 6K$$

Se considero il segnale x uniformemente distribuito su $[-V, V]$ ottengo la densità di probabilità

$$p_X(x) = \frac{1}{2V} \text{rect}\left(\frac{x}{2V}\right)$$

Da qui ottengo la potenza

$$P_x = \frac{1}{2V} \int_{-V}^V x^2 dx = \frac{1}{2V} \frac{x^3}{3} \Big|_{-V}^V = \frac{V^2}{3}$$

Quindi

$$(P_e)_{dB} = (P_x)_{dB} - 6K$$

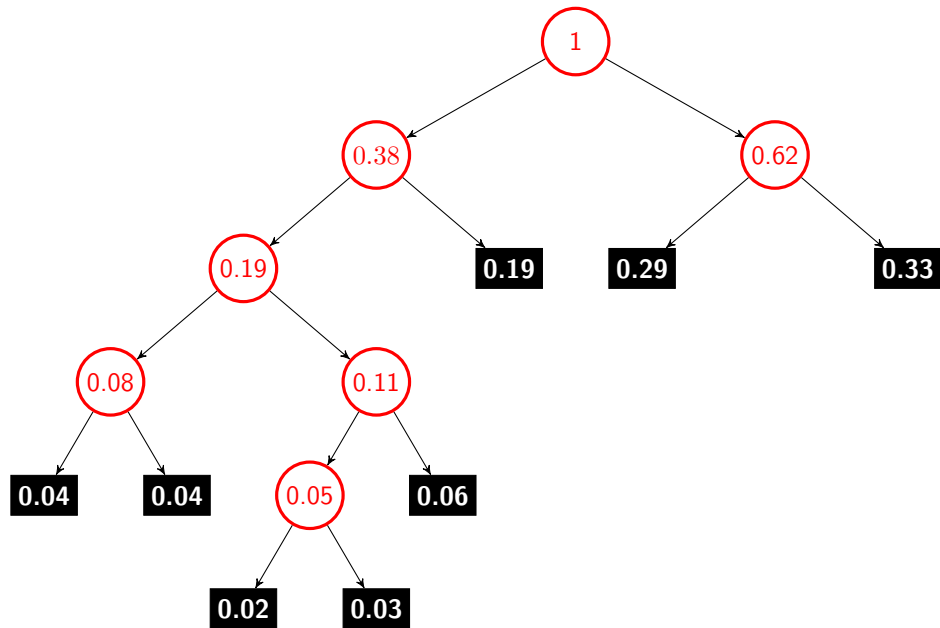
Se prendiamo la frequenza di campionamento $f_c = 8\text{kHz} \Rightarrow T_c = 125\mu\text{s}$ e $K = 8 \Rightarrow 64\text{kb/s}$ abbiamo la pulse code modulation, che è la base della trasmissione dei segnali. Dato che la banda è una risorsa preziosa si fa una codifica di sorgente.

Un esempio è la codifica naturale dove si associa al simbolo con valore più basso la parola con tutti 0 e a quello più alto quella con tutti 1 ma non è la più efficiente dato che alcune parole di codice potrebbero non essere usate mai, quindi spesso si usa una codifica fatta appositamente per il tipo di segnale in cui si associano ai simboli più comuni parole con meno bit e a quelli più rari quelle con più bit, risparmiando così banda usando simboli con numero di bit variabile che riduce il numero medio di bit usati.

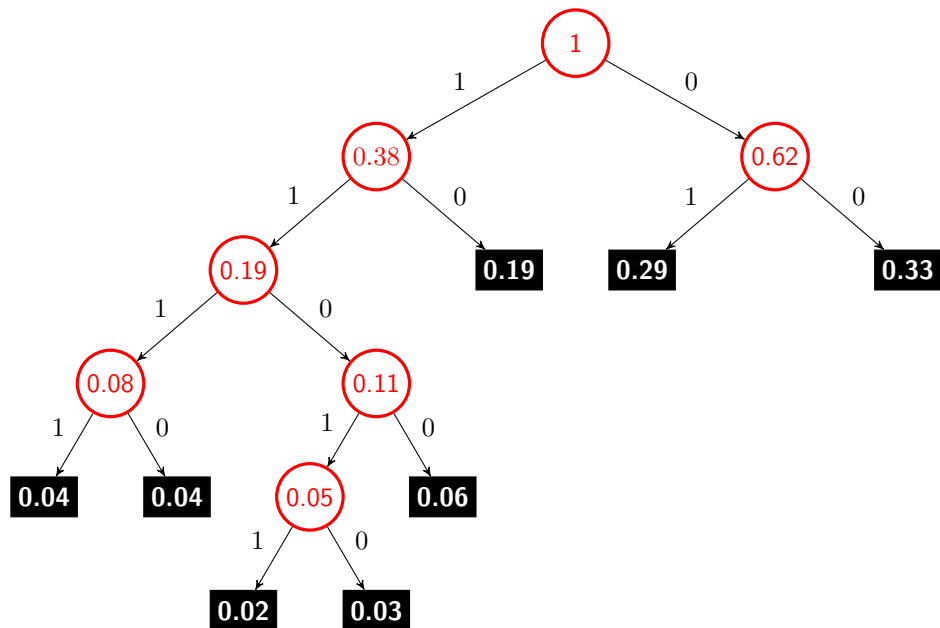
Esempio Prese le probabilità dei simboli

$$[0.02, 0.29, 0.03, 0.04, 0.33, 0.04, 0.06, 0.19]$$

costruiamo un albero (di Huffman) mettendo prima nelle foglie i due valori minori, in questo caso 002 e 003 e mettendo come loro genitore la loro somma, 005 che è molto simile ai valori 004 ripetuto due volte e 006, le metto come foglie mettendo in ordine crescente da destra verso sinistra e ripetendo la procedura precedente, sommando quindi i due 004 e lo 005 con 006, ottenendo rispettivamente 008 e 011 che non sono confrontabili con gli altri bit, quindi li sommo tra di loro ottenendo 019 che è confrontabile con 019, 029 e 033, così ho preso tutti i bit, ora ripeto la procedura fino a ottenere la radice che dovrebbe venire 1.



Con questo albero possiamo distribuire i bit di codifica aggiungendo a ogni figlio sinistro 1 alla codifica e a ogni figlio destro 0. Così facendo ottengo la seguente codifica



0.19	10
0.29	01
0.33	00
0.04	1111
0.04	1110
0.06	1100
0.02	11011
0.03	11010

La ripetizione degli 0.04 è dovuto al fatto che due simboli hanno la stessa probabilità. Il numero medio di bit usati è quindi

$$\#bit = \sum_{i=1}^M p_i K_i = 0.33 \cdot 2 + 0.23 \cdot 2 + \dots = 2.43 \text{ kb/s} < 3 \text{ che sarebbero quelli usati dalla codifica naturale}$$

E questo sistema garantisce la leggibilità del segnale dato che nessuna parola è inizio di un'altra.

È possibile usare un quantizzatore a intervalli non costanti, ad esempio

$$V_1 \rightarrow \Delta_1 = 8 \quad V_2 \rightarrow \Delta_2 = 4 \quad V_3 \rightarrow \Delta_3 = 2 \quad V_4 \rightarrow \Delta_4 = 2$$

quindi

$$p_1 = 0.5 \quad p_2 = 0.25 \quad p_3 = p_4 = 0.125$$

Per calcolare la potenza dell'errore di quantizzazione considero tutti gli intervalli come variabili aleatorie indipendenti e ottengo

$$P_e = \frac{\Delta_1^2}{12} p_1 + \frac{\Delta_2^2}{12} p_2 + \frac{\Delta_3^2}{12} p_3 + \frac{\Delta_4^2}{12} p_4 = \frac{37}{12}$$

In codifica naturale mi servirebbero 2 bit per simbolo ma noi vogliamo ottimizzare quindi creiamo ora l'albero di Huffman e ottengo la codifica

V_1	0
V_2	10
V_3	110
V_4	111

Quindi

$$\#bit = 0.5 \cdot 1 + 0.25 \cdot 2 + 0.125 \cdot 3 + 0.125 \cdot 3 = 1.75 = \frac{7}{4} < 2$$

Ma qual'è il numero minimo di bit che posso usare per trasmettere un segnale senza perdere l'informazione? questo numero è l'entropia della sorgente

$$H = \sum_{i=1}^M p_i \log_2 \frac{1}{p_i} = - \sum_{i=1}^M p_i \log_2 p_i$$

Che in alcuni casi è ottenibile ma non esiste un modo univoco per ottenerlo, dipende dai casi. Si chiama informazione

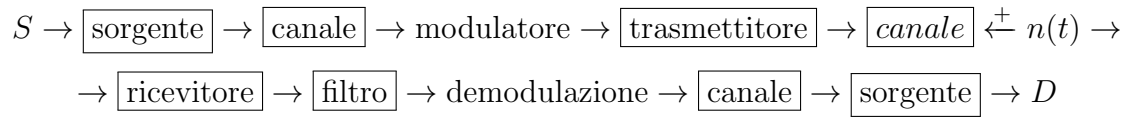
$$I = \log_2 \frac{1}{p_i}$$

La quantità di informazione contenuta in un simbolo. Quindi l'entropia della sorgente è

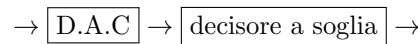
$$H = \sum_{i=1}^M p_i I_i$$

Quindi se un simbolo ha $p_i = 1 \Rightarrow H = 0$ mentre se tutti i simboli sono equiprobabili $p_i = \frac{1}{M} \Rightarrow H = \sum_{i=1}^M \frac{1}{M} \log_2 M = \log_2 M$. Se $M = 2 \Rightarrow H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$

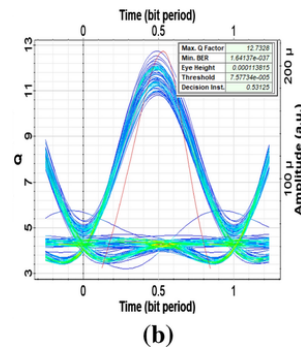
Codifica di linea



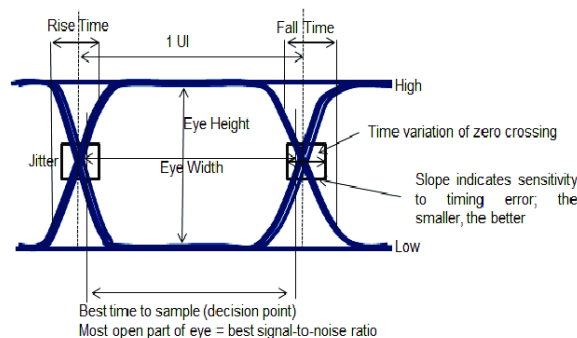
La codifica di linea aggiunge bit per correggere l'errore dovuto al rumore, l'esempio supra riportato è una codifica di linea di modulazione. Tutti i canali attenuano le frequenze alte quindi i canali possono essere visti come dei filtri passa basso con banda pari alla frequenza massima che lasciano passare, inoltre viene introdotto del rumore (che vediamo come un rumore bianco gaussiano). Nella modulazione moltiplica il segnale per una frequenza portante. Dopo che il segnale viene ricevuto dal ricevitore si passa per un filtro, solitamente basato sulla frequenza portante (I segnali wireless sono modellati con delle rect cejtrate in $\pm f_0$ e di ampiezza B che è la banda) per ridurre la potenza del rumore prima della demodulazione che contiene



Dove il decisore a soglia è un sistema di confronto che confronta il segnale ricevuto con una soglia per vedere se il valore ricevuto è uno 0 o un 1. Una cosa fondamentale nella trasmissione del segnale è il **diagramma ad occhio** che stabilisce una zona in cui il segnale non deve entrare altrimenti diventa non leggibile da decisore a soglia. Le codifiche possono essere del tipo "return to 0", dove il segnale torna a 0 prima di trasmettere il bit successivo, in questo caso il diagramma ad occhio è così



Mentre se si usa una modulazione che non torna a 0 prima di trasmettere il diagramma ad occhio simile a questa



Per velocizzare le trasmissioni si tende a non usare una trasmissione binaria ma una trasmissione M-aria, ovvero con M simboli, ad esempio per il wifi si usano 256 simboli

Un esempio: sia il nostro alfabeto

$$c_m \in \{c_1, c_2, \dots, c_M\}$$

e il tempo di trasmissione di un simbolo

$$T_s$$

e quindi il baudrate è

$$R_s = \frac{1}{T_s}$$

quindi il bitrate di trasmissione, che è il tempo per ogni simbolo moltiplicato per il bit di ogni simbolo, si calcola

$$R_b = R_s \log_2 M$$

Ora prendiamo

$$M = 4 \quad c_m = \{-1, -0.5, 0.5, 1\}$$

E vogliamo trasmettere la sequenza

$$s[n] = [1, 0.5, -0.5, 1]$$

Ora lo moduliamo e otteniamo un'onda da trasmettere

$$x(t) = \sum_{n=1}^{\text{length}(s[n])} s[n]g(t - nT_s)$$

Dove $g(t)$ è l'impulso base che deve essere

$$g(t) = \begin{cases} 1 & t = 0 \\ 0 & t = kT_s, k \in \mathbb{Z} \setminus \{0\} \end{cases}$$

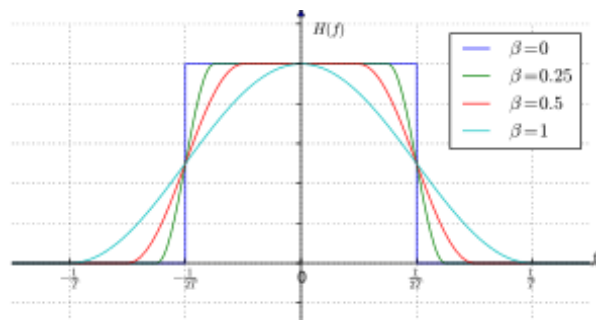
E si cerca di rimanere nel caso in cui la sua trasformata resta limitata in banda per evitare di usarne troppa in quanto è una risorsa preziosa. Quindi un segnale tipicamente usato è

$$g(t) = \text{sinc}\left(\frac{t}{T_s}\right)$$

Dato che la sua trasformata

$$G(f) = \text{rect}(T_s f)$$

È quindi limitata in banda. Un altro impulso molto usato è quello di Nyquist a coseno rialzato (β nel grafico è il rolloff)



Facciamo un altro esempio di trasmissione con un segnale binario

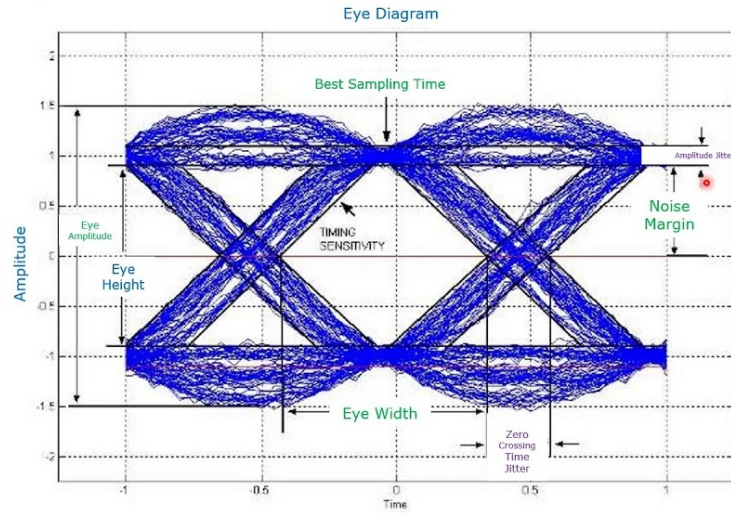
$$s[n] \in \{c_1, c_2\}$$

$$x(t) = \sum_{n=-\infty}^{+\infty} s[n]g(t - nT_s)$$

Ora dobbiamo aggiungere il rumore e il passaggio per il canale

$$y(t) = x(t) * h(t) + n(t) = x(t) + n(t)$$

quindi il nostro segnale non verrà fuori pulito ma a causa del rumore diventa “frastagliato”



Considerando che il rumore ha una densità di probabilità gaussiana possiamo vedere $y(t)$ come una variabile aleatoria

$$Y = y(0) = x + n = \begin{cases} c_1 + n \\ c_2 + n \end{cases}$$

$$P_N(n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{n^2}{2\sigma_n^2}} \quad \sigma_n^2 = BN_0$$

$$P_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(y-c_i)^2}{2\sigma_n^2}}$$

$$p_r\{y < \bar{c}|c_2\} = \int_{-\infty}^{\bar{c}} R_Y(y)dy = \int_{-\infty}^{\bar{c}} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(y-c_2)^2}{2\sigma_n^2}} dy \stackrel{t=\frac{c_2-y}{\sqrt{2}\sigma_n}}{=} \int_{+\infty}^{\frac{c_2-\bar{c}}{\sqrt{2}\sigma_n}} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-t^2} (-dt\sqrt{2}\sigma_n) =$$

$$= \int_{\frac{c_2-\bar{c}}{\sqrt{2}\sigma_n}}^{+\infty} \frac{1}{\sqrt{\pi}} e^{-t^2} = \frac{1}{2} \operatorname{erfc}\left(\frac{c_2-\bar{c}}{\sqrt{2}\sigma_n}\right) = Q\left(\frac{c_2-\bar{c}}{\sigma_n}\right)$$

$$p_r\{y > \bar{c}|c_1\} = \int_{\bar{c}}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(y-c_1)^2}{2\sigma_n^2}} dy = \frac{1}{2} \operatorname{erfc}\left(\frac{\bar{c}-c_1}{\sqrt{2}\sigma_n}\right) = Q\left(\frac{\bar{c}-c_1}{\sigma_n}\right)$$

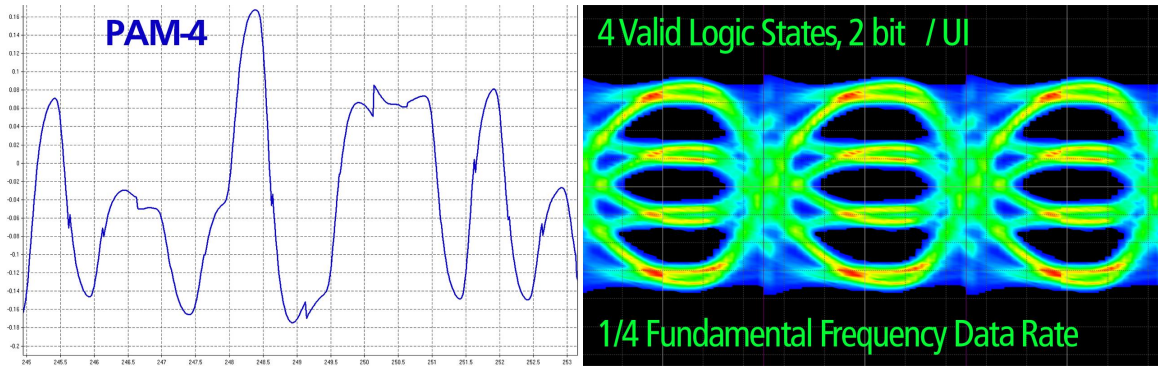
Se voglio calcolare il bit error rate (BER) devo fare la sommatoria delle probabilità di ciascun simbolo per la rispettiva Q , quindi in questo caso

$$BER = \frac{1}{2}Q\left(\frac{\bar{c} - c_1}{\sigma_n}\right) + \frac{1}{2}Q\left(\frac{c_2 - \bar{c}}{\sigma_n}\right)$$

che in questo caso è

$$BER = Q\left(\frac{c_2 - c_1}{\sigma_n}\right) = Q\left(\frac{d}{\sigma_n}\right) = Q\left(\frac{d}{\sqrt{N_0 B}}\right)$$

Un modo per trasmettere molteplici simboli è la codifica PAM (pulse amplitude modulation) in cui a valori diversi del segnale trasmesso corrisponde un diverso simbolo, ecco un'immagine del grafico della trasmissione e del rispettivo diagramma ad occhio generato



Bisogna ricordarsi che quando si trasmette un segnale attraverso un canale possiamo modellarlo così

$$x(t) \rightarrow \boxed{H(f) + n(t)} \rightarrow y(t)$$

Dove $n(t)$ è un AWGN (additive white gaussian noise).

Quindi

$$y(t) = \sum_m c_m g(t - mT_s) + n(t) \Rightarrow y(0) = Y = c_m + n$$

$$p_N(n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{n^2}{2\sigma_n^2}}$$

$$\sigma_n^2 = \int_{-\infty}^{+\infty} \frac{N_0}{2} df = \int_{-B}^B \frac{N_0}{2} df = BN_0$$

$$\Rightarrow p_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(y-c_1)^2}{2\sigma_n^2}}$$

Se le due trasmissioni si vanno a sovrapporre è possibile che si crei un errore dovuto alla trasmissione di un simbolo sbagliato dato che con il pezzettino della trasmissione precedente viene superata la soglia per un altro simbolo, quindi si aggiunge la funzione di errore complementare

$$\frac{1}{2} \operatorname{erfc}\left(\frac{\bar{c} - c_1}{\sqrt{2}\sigma_n}\right)$$

Dove

$$\operatorname{erfc}(x) = \int_x^{+\infty} e^{-t^2} dt \frac{2}{\sqrt{\pi}}$$

Da cui si può calcolare il BER sommando la funzione di errore complementare per ogni simbolo moltiplicata per la probabilità di trasmissione del simbolo stesso. Ovviamente è minimo quando i due simboli sono uguali.

per pigrizia è stata creata la funzione

$$Q(x) \doteq \frac{1}{2} \operatorname{erfc} \left(\frac{x}{\sqrt{2}} \right)$$

Il ber è considerato *error free* quando

$$BER = 10^{-9} \Rightarrow Q(6) \xrightarrow{BER=Q\left(\frac{d}{2\sqrt{N_0 B}}\right)} d \gg 0$$

Ovviamente una volta ricevuta la trasmissione questa viene filtrata per rimuovere almeno il rumore fuori banda

$$x(t) \rightarrow \boxed{\text{canale}} \rightarrow \boxed{H_0(f)} \rightarrow y(t)$$

Ma anziché usare una *rect* si può usare un filtro adattato per migliorare le prestazioni

$$y(t) = (x(t) + n(t)) * h_0(t) = \left[\sum_m c_m g(t - mT_s) + n(t) \right] * h_0(t)$$

Per vedere la statistica evito di portarmi dietro la sommatoria e analizzo il caso di trasmissione di un singolo campione

$$c_m g(t) * h_0(t) + n(t * h_0(t))$$

Ora controllo il segnale utile

$$c_m g(t) * h_0(t) = c_m \int_{-\infty}^{+\infty} G(f) H_0(f) e^{i2\pi f t} df$$

E lo calcolo in $t = 0$

$$c_m \int_{-\infty}^{+\infty} G(f) H_0(f) df$$

A questo punto posso calcolare il numeratore del *Signal to Noise Ratio* calcolando la potenza del segnale utile

$$SNR = \frac{c_m^2 \left| \int_{-\infty}^{+\infty} G(f) H_0(f) df \right|^2}{\dots}$$

Ora guardiamo la densità spettrale del rumore una volta uscito dal filtro

$$\frac{N_0}{2} |H_0(f)|^2$$

Ora calcoliamone la potenza

$$\sigma_w^2 = \int_{-\infty}^{+\infty} \frac{N_0}{2} |H_0(f)|^2 df$$

E abbiamo anche il denominatore dell'SNR

$$SNR = \frac{c_m^2 \left| \int_{-\infty}^{+\infty} G(f) H_0(f) df \right|^2}{\int_{-\infty}^{+\infty} \frac{N_0}{2} |H_0(f)|^2 df}$$

A questo punto ricordiamo la disuguaglianza di Schwartz che dice che il quadrato del modulo dell'integrale è minore o uguale del prodotto degli integrali dei moduli quadri

$$SNR = \frac{c_m^2 \left| \int_{-\infty}^{+\infty} G(f) H_0(f) df \right|^2}{\int_{-\infty}^{+\infty} \frac{N_0}{2} |H_0(f)|^2 df} \leq \frac{c_m^2 \int_{-\infty}^{+\infty} |G(f)|^2 df \int_{-\infty}^{+\infty} |H_0(f)|^2 df}{\frac{N_0}{2} \int_{-\infty}^{+\infty} |H_0(f)|^2 df}$$

Dato che vogliamo che valga l'uguaglianza per avere il massimo SNR usiamo la condizione della disuguaglianza di Schwartz che ci dice che l'uguaglianza vale quando una funzione è la complessa coniugata dell'altra

$$\begin{aligned} H_0(f) = G^*(f) &\Rightarrow h_0(t) = g^*(-t) \stackrel{g(t) \in \mathbb{R} \forall t \in \mathbb{R}}{=} g(-t) \\ \Rightarrow SNR &= \frac{c_m^2 2 \int_{-\infty}^{+\infty} |G(f)|^2 df}{N_0} \end{aligned}$$

Inoltre so che

$$E_G = \int_{-\infty}^{+\infty} |G(f)|^2 df = \int_{-\infty}^{+\infty} |g(t)|^2 dt \Rightarrow SNR = \frac{2c_m^2 E_G}{N_0}$$

E questo è il massimo valore possibile dell'SNR e si ottiene con un filtro adattato.

A questo punto calcoliamo il BER nel caso di un filtro adattato

$$BER = Q\left(\frac{\sqrt{SNR_2} - \sqrt{SNR_1}}{2}\right) = Q\left(\frac{c_2 - c_1}{2} \sqrt{\frac{2E_G}{N_0}}\right) = Q\left((c_2 - c_1) \sqrt{\frac{E_G}{2N_0}}\right)$$

Ora vediamo i casi particolari

- UNIPOLARE ($c_1 = 0, c_2 \neq 0$)

$$BER = Q\left(c_2 \sqrt{\frac{E_g}{2N_0}}\right)$$

- BIPOLARE ($c_1 = -c_2$)

$$BER = Q\left(2c_2 \sqrt{\frac{E_g}{2N_0}}\right)$$

Ma in questo caso sto considerando l'energia del simbolo pensando che siano uguali nella bipolare e nella unipolare, sarebbe meglio usare l'energia del bit

- UNIPOLARE

$$E_b = \frac{c_2^2 E_g + 0}{2} \Rightarrow BER = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

- BIPOLARE

$$E_b = \frac{c_1^2 E_g + c_2^2 E_g}{2} \Rightarrow BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

E questi dati sono quelli che di solito si usano per generare un grafico che mostra le prestazioni di un sistema mettendo sull'asse x $\frac{E_b}{N_0}$ e su y BER

Integrare appunti lezione del 14:00 del 16/01/24

Supponiamo di aver codificato un segnale $x(t)$ e che lo stiamo trasmettendo tramite un canale in banda base

$$x(t) = \sum_m c_m g(t - mT_s) \rightarrow \boxed{H(f) = \begin{cases} \cdots & -B \leq f \leq B \\ 0 & \text{altrove} \end{cases} + n(t)} \rightarrow \\ \rightarrow \boxed{\text{filtro adattato } G^*(f)} \rightarrow \boxed{DAC} \rightarrow \boxed{\text{decisore a soglia}}$$

Da qui abbiamo

$$BER = Q\left((c_2 - c_1)\sqrt{\frac{E_g}{2N_0}}\right) \xrightarrow{\text{casobipolare}} Q = \left(\sqrt{\frac{E_b 2}{N_0}}\right)$$

Se vogliamo un segnale error free dobbiamo avere

$$BER = 10^{-9} \Rightarrow \sqrt{\frac{E_b 2}{N_0}} = 6 \Rightarrow \frac{E_b}{N_0} = 18 = 12.5dB$$

A questo punto vogliamo sapere quanto è il bitrate massimo, per questo ci serve il secondo teorema di Shannon che dice

Prendiamo un canale con banda B corrotto da un rumore bianco gaussiano, allora la capacità di canale (misurata in $\frac{b}{s}$) per trasmettere un segnale con errore piccolo a piacere è

$$C = B \log_2 \left(1 + \frac{P_R}{P_N}\right)$$

dove P_R è la potenza ricevuta e P_N la potenza del rumore, che noi abbiamo chiamato σ_n^2 . Allora lo riscrivo così

$$C = B \log_2 \left(1 + \frac{P_R}{\sigma_n^2}\right) = B \log_2(1 + SNR)$$

Ma avevamo detto che nel segnale in banda base il symbol rate doveva essere $R_s \leq 2B$ e quindi il bit rate $R_b \leq 2B \log_2(M)$ quindi confrontandolo con il teorema sopracitato abbiamo

$$2B \log_2 M = B \log_2(1 + SNR) \Rightarrow \log_2 M^2 = \log_2(1 + SNR) \Rightarrow M = \sqrt{1 + SNR}$$

Vediamo i vari casi in rapporto ai valori dell'SNR

- caso peggiore

$$SNR = 0dB \Rightarrow \frac{P_R}{P_N} = 1 \Rightarrow C = B$$

- Caso decente

$$SNR = 20dB \Rightarrow \frac{P_R}{P_N} = 100 \Rightarrow C = B \log_2(101) \sim 6.65B$$

-

$$B = 10kHz, C = 50 \frac{kb}{s} \Rightarrow \frac{C}{B} = 5 \Rightarrow 1 + SNR = 2^5$$

- GPS

$$B = 1MHz, SNR = -30dB = 10^{-3} \Rightarrow C = B \log_2(1 + 10^{-3}) = 1.4 \frac{kb}{s}$$

Ricordiamo che

$$P_R = \frac{E_b}{T_b} \Rightarrow P_R = E_b C$$

E che

$$R_N = N_0 B$$

Quindi

$$C = B \log_2 \left(1 + \frac{E_b C}{N_0 B} \right)$$

$$\frac{C}{B} = \log_2 \left(1 + \frac{E_b}{N_0} \cdot \frac{C}{B} \right)$$

$$2^{\frac{C}{B}} = 1 + \frac{E_b}{N_0} \cdot \frac{C}{B}$$

$$\frac{E_b}{N_0} = \frac{2^{\frac{C}{B}} - 1}{\frac{C}{B}}$$

Così possiamo vedere il rapporto $\frac{C}{B}$ in funzione di $\frac{E_b}{N_0}$, che viene graficato con quest'ultimo espresso in scala logaritmica sulle x e $\frac{C}{B}$ sulle y. Da qui c'è un limite molto importante che dobbiamo vedere, che è quello per $C = 0$ e per vederlo riscriviamo la formula così

$$\frac{N_0}{E_b} = \frac{N_0 B}{E_b C} \log_2 \left(1 + \frac{E_b}{N_0} \cdot \frac{C}{B} \right) \xrightarrow{\lambda = \frac{E_b}{N_0} \cdot \frac{C}{B}} \frac{1}{\lambda} \log_2(1 + \lambda) \Rightarrow \frac{N_0}{E_b} = \frac{1}{\lambda} \frac{\ln(1 + \lambda)}{\ln 2}$$

$$\lim_{\lambda \rightarrow 0} \frac{1}{\lambda} \frac{\ln(1 + \lambda)}{\ln 2} \xrightarrow{\lim_{\lambda \rightarrow 0} \frac{\ln(1 + \lambda)}{\lambda} = 1} \frac{1}{\ln 2}$$

E questo è il valore che assume $\frac{N_0}{E_b}$ quando $\frac{C}{B} \rightarrow 0$

Ma c'è anche un'altra curva da studiare, quella del rapporto tra C (sulle y) e B (sulle x) assumendo costante P_R

$$C = B \log_2 \left(1 + \frac{P_R}{N_0 B} \right)$$

Ed è una curva logairtmica con asintoto in $1.44 \frac{P_R}{N_0}$ che troviamo così

$$C = \frac{B N_0}{P_R} \frac{P_R}{N_0 \log_2 \left(1 + \frac{P_R}{N_0 B} \right)} \xrightarrow{\lambda = \frac{P_R}{N_0 B}} \frac{1}{\lambda} \frac{P_R}{N_0} \log_2(1 + \lambda) = \frac{P_R}{N_0} \frac{\ln(1 + \lambda)}{\ln 2 \cdot \lambda} \xrightarrow{\lim_{\lambda \rightarrow 0}} \frac{P_R}{N_0} = 1.44 \frac{P_R}{N_0} = 1.44 B \cdot SNR$$

Se $\frac{P_R}{N_0} \gg 1$

$$C = B \log_2 \left(\frac{P_R}{N_0 B} \right) = B \log_{10} \frac{P_R}{N_0 B} \cdot \frac{1}{\log_{10} 2} = B \cdot SNR|_{dB} = 0.33$$

Codici

Per ridurre la possibilità di errori Shannon ci dice di trasmettere più bit del necessario per avere dei bit di ridondanza. Mettiamo che la nostra informazione sia composta da R bit, allora trasmetteremo $L = R + K$ bit. Quindi se vogliamo trasmettere la nostra informazione con un bitrate di R_b dovremo trasmettere con un bitrate effettivo di

$$R_c = \frac{R_b L}{K}$$

Facciamo un esempio. Vogliamo trasmettere con un alfabeto sorgente fatto di 3 bit, ma per evitare errori ne trasmetto 4, ma quindi tra le 16 parole

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

Ne scelgo solo 8 da trasmettere dato che voglio trasmettere solo 8 simboli, allora sceglierò tutte parole che distino almeno 2 bit l'una dall'altra, ad Esempio

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

⇓

0000	0011	0101	0110	1001	1010	1100	1111
------	------	------	------	------	------	------	------

così ogni parola ha almeno 2 bit diversi da un'altra (**distanza di humming minima**). Per identificare un errore la distanza di humming deve essere

$$d_{\min} \leq R + 1$$

Mentre per poterlo anche correggere deve essere

$$d_{\min} \leq 2R + 1$$