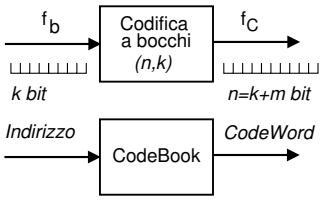
Capitolo 17

Teoria dell'Informazione e Codifica

17.1 Codici di canale

Proseguiamo ora il discorso iniziato al §5.3, relativo al modo di scegliere come aggiungere ridondanza al flusso binario a velocità f_b da trasmettere, in modo realizzare una protezione FEC capace di ridurre la probabilità di errore sul bit P_e in ricezione.

Adottiamo la notazione introdotta per i codici a blocchi, in cui per ogni k bit in ingresso, ne sono prodotti un totale di n, avendone aggiunti m *di protezione* in funzione dei primi k, avendo indicato tale modo di procedere come un codice (n, k), la cui efficienza 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.

La nuova velocità di trasmissione vale pertanto

$$f_b^{'} = \frac{f_b}{R_c}$$

che come abbiamo visto deve essere $f_b^{'} < C$, ossia inferiore al valore di capacità di canale (17.22), pena l'impossibilità di trasmettere senza errori. D'altra parte, aumentando la velocità di segnalazione diminuisce di pari misura il rapporto E_b/N_0, e conseguentemente peggiora anche la probabilità di errore del decisore, in modo che la capacità correttiva del codice deve essere tale da compensare anche quest'altro fattore.

Risulta quindi evidente come ci sia tutto l'interesse a mantenere R_c sufficientemente elevato, pena l'inutilità del processo di codifica di canale. Al tempo stesso però, la capacità di correzione del codice è direttamente legata alla *distanza di Hamming* d_H definita a pag. 87 come il minimo numero di bit diversi tra due parole di codice, sussistendo le relazioni

- per detettare almeno l errori per parola occorre $d_H \geq l + 1$
- per correggere almeno t errori per parola occorre $d_H \geq 2t + 1$

- per correggere almeno t errori e detettare almeno l errori per parole occorre $d_H \geq t + l + 1$

Un codice è tanto più *potente* quanti più errori è in grado di correggere, e dunque deve possedere d_H elevato. In un codice a blocchi (n, k) i k bit del messaggio originale assumono tutte le configurazioni possibili, e quindi contribuiscono alla distanza tra codeword per un solo bit; per ottenere $d_H > 1$ occorre pertanto sfruttare gli $n - k = q$ bit di protezione, portando a scrivere

$$d_H \leq n - k + 1 = q + 1$$

che evidenzia la relazione tra d_H e la quantità di bit aggiunti q. Purtroppo l'uguaglianza sussiste solo per i codici a ripetizione, discussi a pag. 88, che adottando una dimensione di blocco in ingresso $k = 1$ hanno un tasso di codifica $R_c = {}^{k}/_{n} = {}^{1}/_{n}$ molto inefficiente. Mostriamo allora delle soluzioni che consentono di ottenere un adeguato potere di detezione senza per questo aumentare di molto la velocità di trasmissione del flusso codificato.

17.1.1 Codici a blocchi lineari

Le proprietà di questa classe di codici di canale possono essere meglio analizzate interpretando l'insieme delle possibili codeword da un punto di vista algebrico, che ci porta ad adottare una notazione matriciale idonea a descrivere la classe di codici *di Hamming*, mentre per i codici ciclici e di *Reed-Solomon* interverrà una notazione polinomiale.

Distanza d_H per codici lineari Scegliamo di rappresentare una codeword arbitraria X di una codice a blocchi (n, k) mediante un vettore ad elementi binari

$$X = (x_1 \quad x_2 \quad \cdots \quad x_n)$$

che può assumere solo 2^k diversi valori tra i 2^n possibili, mentre le ricezione di una delle rimanenti $2^n - 2^k$ combinazioni segnala la presenza di almeno un errore. Le 2^k codeword costituiscono uno *spazio lineare* se comprendono la codeword nulla, e se la somma di due vettori è anch'essa una parola di codice. La somma tra due codeword è definita in base alla matematica binara modulo due, espressa mediante l'operatore di OR esclusivo $\oplus$ come

$$X + Y = (x_1 \oplus y_1 \quad x_2 \oplus y_2 \quad \cdots \quad x_n \oplus y_n)$$

Indicando ora come *peso* $w(Z)$ di un vettore Z il numero di suoi elementi non zero, una conseguenza della linearità è la possibilità di valutare la distanza di Hamming tra parole di codice come il *minimo peso* tra tutte le codeword non zero[1], ossia

$$d_H = \min_{X \neq 0} [w(X)]$$

Rappresentazione matriciale per codici sistematici Benché finora implicitamente assunto, l'esistenza di una ripartizione netta degli n bit delle codeword in una prima parte contenente i k bit da proteggere, seguiti dai $q = n - k$ bit di protezione, viene indicata come la condizione che definisce un codice *sistematico*. In tal caso possiamo scrivere le codeword come

$$X = (m_1 \quad m_2 \quad \cdots \quad m_k \quad c_1 \quad c_2 \quad \cdots \quad c_q)$$

[1] Infatti dalla definizione di somma otteniamo che la distanza tra due codeword X e Y è pari al peso della codeword $Z = X + Y$: infatti Zpresenterà componenti $z_j = 1$ solo in corrispondenza di elementi $x_j \neq y_j$. Ma per la linearià anche Z appartiene al codebook, e dunque la ricerca su tutte le coppie si trasforma in una ricerca su tutte le codeword.

ovvero in forma partizionata $X = (M \mid C)$ che permette di rappresentare X a partire dai bit da proteggere M e dalla definizione di una matrice generatrice $k \times n$ con struttura generale $G = [I_k \mid P]$ in cui I_k è una matrice identità $k \times k$ e P è una sotto-matrice di elementi binari $k \times n$, come $X = MG$ ovvero

$$\begin{bmatrix} m_1 & \cdots & m_k & c_1 & \cdots & c_q \end{bmatrix} = \begin{bmatrix} m_1 & \cdots & m_k \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 & p_{11} & \cdots & p_{1q} \\ \vdots & 1 & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & p_{k1} & \cdots & p_{kq} \end{bmatrix}$$

in modo che P produca[2] i q bit di protezione come $C = MP$. Dato che il valore della (generica) j-esima componente di C si calcola come

$$c_j = m_1 p_{1j} \oplus m_2 p_{2j} \oplus \cdots \oplus m_k p_{kj}$$

osserviamo che ciascuna colonna di P individua un sotto-insieme di componenti di M su cui calcolare una somma di parità, e per questo è rappresentata dalla lettera P. Ma non è ancora stato definito nulla che ci possa aiutare a scegliere i coefficienti p_{ij} allo scopo di ottenere i valori d_H e R_c desiderati: il codice di Hamming ci fornisce una possibile soluzione.

Codice di Hamming E' un codice a blocchi lineare $(n.k)$ con $q \geq 3$ bit di controllo e

$$n = 2^q - 1 \qquad k = n - q$$

per cui il tasso di codifica vale

$$R_c = \frac{k}{n} = \frac{n-q}{n} = 1 - \frac{q}{2^q - 1}$$

che quindi aumenta con il crescere di q. Per individuare le codeword, si pongono le righe della sottomatrice P pari a tutte le parole di q bit con due o più uni, in qualsiasi ordine. Ma la cosa più simpatica, è che il codebook risultante esibisce un valore $d_H = 3$, indipendentemente dalla scelta di q.

Esempio: codice di Hamming $(7, 4)$ Consideriamo un codice di Hamming sistematico con $q = 3$, e quindi $n = 2^3 - 1 = 7$ e $k = 7 - 3 = 4$. La matrice generatrice è quindi pari a

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

La seguente tabella mostra le risultanti $2^4 = 16$ codeword, confermando che $d_H = 3$.

M	C	$w(X)$
0 0 0 0	0 0 0	0
0 0 0 1	0 1 1	3
0 0 1 0	1 1 0	3
0 0 1 1	1 0 1	4
0 1 0 0	1 1 1	4
0 1 0 1	1 0 0	3
0 1 1 0	0 0 1	3
0 1 1 1	0 1 0	4

M	C	$w(X)$
1 0 0 0	1 0 1	3
1 0 0 1	1 1 0	4
1 0 1 0	0 1 1	4
1 0 1 1	0 0 0	3
1 1 0 0	0 1 0	3
1 1 0 1	0 0 1	4
1 1 1 0	1 0 0	4
1 1 1 1	1 1 1	7

[2]Sono valide le normali regole di moltiplicazione tra matrici, tranne per l'accortezza di usare la somma modulo due anziché quella convenzionale.

Sindrome e decodifica di massima verosimiglianza Indichiamo ora con Y la parola di codice ricevuta, che in presenza di errori risulta $Y \neq X$. Il metodo diretto per rivelare ed eventualmente correggere gli errori presenti è di confrontare gli n bit ricevuti con tutte le possibili 2^k codeword, e se nessuna di queste risulta uguale ad Y, scegliere la $\hat{Y}$ più vicina, ossia quella per la quale il peso $w\left(Y + \hat{Y}\right)$ è minimo.

Un metodo che non richiede una ricerca esaustiva si basa invece sul calcolo della cosiddetta *sindrome*, ottenuta mediante moltiplicazione del vettore Y ricevuto con una matrice $n \times q$ di *controllo parità* H definita come $H = \left[\frac{P}{I_q}\right]$ in cui P è la stessa matrice di parità utilizzata nella nella matrice generatrice G, e I_q una matrice identità di dimensioni $q \times q$. La matrice H esibisce la gradevole proprietà che, se moltiplicata per una qualunque codeword valida, fornisce un vettore nullo, ossia

$$XH = (0 \quad 0 \quad \cdots \quad 0)$$

Al contrario, se moltiplicata per un vettore $Y \neq X$ fornisce un vettore *sindrome* $S = YH$ di dimensione q diverso da zero, e quindi il suo calcolo permette la rivelazione (nei limiti consentiti da d_H) dell'occorrenza di errori. Per quanto riguarda la correzione, scriviamo il vettore ricevuto come $Y = X + E$ dove E è un vettore di n bit, le cui componenti sono diverse da zero in corrispondenza dei bit errati di Y. Con questa posizione il calcolo della sindrome risulta

$$S = YH = (X + E)\, H = XH + EH = EH$$

dato che come detto sopra la sindrome delle codeword è nulla. Ora però è evidente un nuovo problema, in quanto tutte le 2^n possibili sequenze di errore E danno luogo a sole 2^q diverse sindromi, segno evidente che la conoscenza della sindrome non consente di risalire direttamente a E. Ma riprendendo i risultati esposti a pag. 83, avviene che la probabilità $P(i, n)$ che si siano verificati i errori su n bit decresce al crescere di i, e pertanto il vettore E che con maggior probabilità ha prodotto ognuna delle 2^q sindromi $S \neq 0$ è quello (tra tutti quelli che producono S) con il minor peso:

$$\hat{E} = \underset{E:EH=S}{\operatorname{argmin}} \{w(E)\}$$

Anziché effettuare questo calcolo ad ogni codeword ricevuta, si può precalcolare, per ogni possibile E, la relativa sindrome, e memorizzare in una tabella quella con peso minore per ciascuna sindrome. La stessa tabella può quindi essere consultata usando come chiave di ricerca la sindrome $S \neq 0$ calcolata per ogni Y ricevuta, ottenendo il vettore di errore più probabile, che se sommato ad Y, permette la correzione degli errori.

A questo punto va spesa una parola di cautela, perché se si sono verificati più errori di quanti d_H permetta correggere, è inutile (anzi dannoso) cercare di eseguire la correzione, perché il numero di errori complessivo potrebbe essere ancora più elevato. Ad esempio considerando il codice di Hamming $(7, 4)$ caratterizzato da $d_H = 1$, esistono solo n vettori E di peso unitario, a cui corrispondono sindromi esattamente pari alle $n = 2^q - 1$ righe di H. Supponiamo ora che E contenga due errori: la sua moltiplicazione per la sindrome produce comunque una delle 2^q possibili sindromi e (se è risultato $S \neq 0$) il tentativo di correzione produrrà un vettore $\hat{X}$ contenente comunque uno o tre errori.