EL4005 Principios de Comunicaciones

Clase No.28: Introducción a la Codificación para Control de Errores

Patricio Parada

Departamento de Ingeniería Eléctrica
Universidad de Chile

19 de Noviembre de 2010

Contenidos de la Clase (1)

Motivación

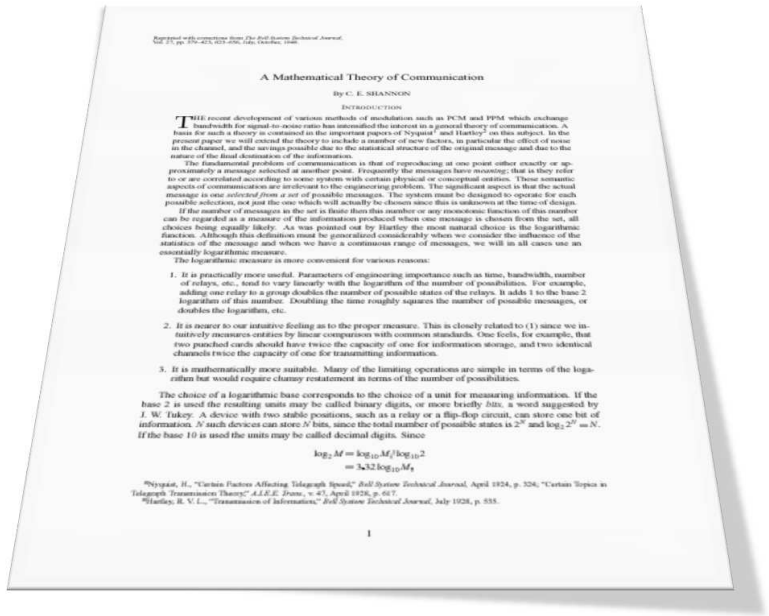
- Hemos visto que en algunas ocaciones, nos resulta imposible cometer errores en nuestro sistema de modulación digital.
- ¿Podemos prevenir de alguna forma la aparición de estos errores?
- Dicho de otra forma, ¿podemos proteger de alguna forma los datos que estamos transmitiendo o almacenando digitalmente?

Introducción a CCE (1)

- Sistemas de comunicaciones modernos requieren altas tasas de transmisión.
- y con bajas tasa de error ($\sim 10^{-12}$).
- Debemos lograr este objetivo con recursos limitados: ancho de banda y potencia.
- Incorporación de redundancia en la transmisión de datos.

Algo de historia (1)

- 1948: Shannon publica “A mathematical theory of communications”.
- Introduce la noción de **capacidad de canal**.
- Se establecen dos “escuelas” de pensamiento: algebraica de y probabilística.



A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.
2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measures entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.
3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits*, a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\log_2 M = \log_{10} M / \log_{10} 2 = 3.32 \log_{10} M$$

[1] Nyquist, H., “Certain Factors Affecting Telegraph Speed,” *Bell System Technical Journal*, April 1924, p. 324; “Certain Topics in Telegraph Transmission Theory,” *A.I.E.E. Trans.*, v. 47, April 1928, p. 617.
[2] Hartley, R. V. L., “Transmission of Information,” *Bell System Technical Journal*, July 1928, p. 535.

1

- 1950’s: Aparecen los **códigos de bloque**, basados en principios de álgebra de campos finitos.

Algo de historia (2)

  - Hamming (single errors)
  - Reed-Muller (multiple errors)
- 1960's:
  - 1960: Bose y Ray-Chaudhuri (1960) y Hocquenghem (1959) descubren códigos de bloque capaces de corregir múltiples errores y proveer altas tasas.
  - Estos códigos se conocen como BCH.
  - 1960: Reed y Solomon (1960) y Arimoto (1961) descubren una clase similar de códigos para canales no binarios.
  - Estos códigos se conocen como Reed-Solomon.

Algo de historia (3)

- La segunda avenida de desarrollo ve el problema de codificación y decodificación con un enfoque probabilístico.
- Se buscaba determinar la probabilidad de error de una familia de códigos, aunque ella no pudiese ser descrita en términos precisos.
- Se establece la idea de **decodificación secuencial**.
- De particular relevancia resultan los **códigos convolucionales**
- 1967: Andrew Viterbi propone un algoritmo para decodificación secuencial óptimo
- 1970's: se funden ambas visiones del problema de codificación.

Algo de historia (4)

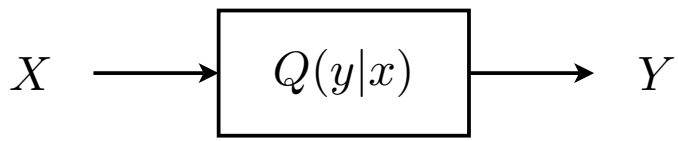
- Massey y Forney proponen la teoría algebraica de códigos convolucionales
- 1966: Forney pone el primer bloque con su teoría de códigos concatenados.
- 1970: Aparecen los códigos de Goppa.
- 1980's: CCE comienza a llegar a la eletrónica de consumo: códificación RS se utiliza en la lectura y grabación de discos compactos.
- Aparecen códigos euclideanos que reemplazan la codificación algebraica en transmisiones telefónicas.

Algo de historia (5)

- 1990's: se hace evidente la convergencia de comunicaciones digitales, procesamiento de señales y codificación.
- 1993: Berrou presenta los códigos turbo.
- 2000's: aplicaciones de CCE es central en aplicaciones de comunicaciones inalámbricas, redes de sensores, etc.
- Los códigos LDPC (low density parity-check) son "redescubiertos" y se vuelven la promesa de la década.
- Se difunden ideas como codificación en red (network coding).

El canal de comunicaciones discreto (1)

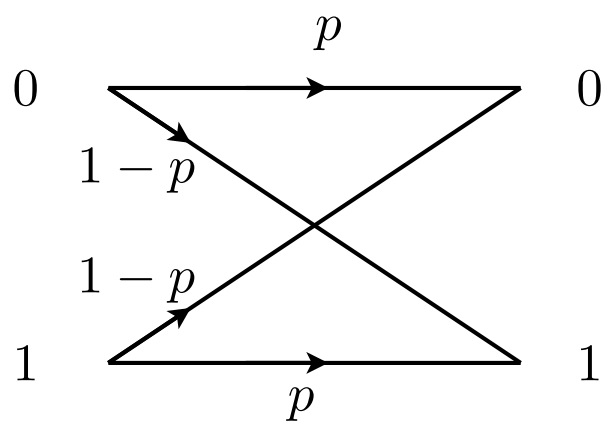
- Un sistema de comunicaciones conecta fuentes de datos con usuarios mediante canales.
- **Definición**:
  Un canal es discreto ssi transmitan símbolos de un alfabeto con un número finito de posibles valores.
- En general, un canal discreto queda completamente definido por la probabilidad de transición entre los símbolos de la entrada y la salida $Q(y|x)$.



El canal de comunicaciones discreto (2)

- Un ejemplo fundamental es el Canal Binario Simétrico (BSC) descrito por las ecuaciones

$$\Pr(0|0) = \Pr(1|1) = p$$
$$\Pr(0|1) = \Pr(1|0) = 1 - p.$$

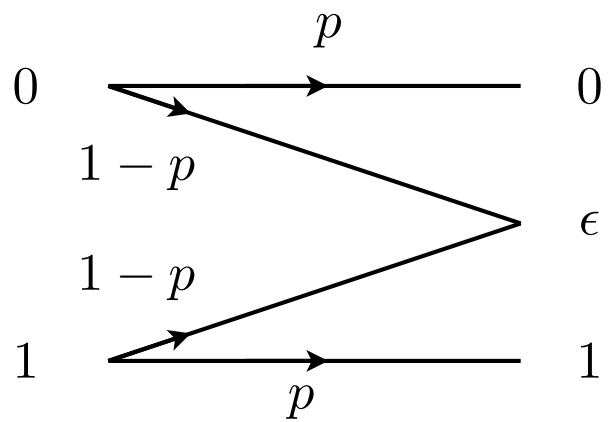


El canal de comunicaciones discreto (3)

- Otro ejemplo habitualmente encontrado es el canal binario con símbolo de borrado (BEC), que permite describir la situación en que un símbolo se pierde en la transmisión.

$$\Pr(0|0) = \Pr(1|1) = p$$

$$\Pr(0|\epsilon) = \Pr(1|\epsilon) = 1 - p.$$



Alfabetos de mayor cardinalidad (1)

- Es habitual encontrar aplicaciones donde se utilizan alfabetos no binarios.
- En códigos cualquiera de estos alfabetos se estudia mediante el conjunto representante

$$\mathcal{A} = \{0, 1, 2, \ldots, q-1\} \tag{1}$$

- Para poder operar sobre este alfabeto, y finalmente codificar, es necesario dotar al conjunto de dos operaciones: adición $\oplus$ y multiplicación $\otimes$.

Alfabetos de mayor cardinalidad (2)

- Para algunos conjuntos particulares, estas operaciones satisfacen las siguientes propiedades:
  (1) Cerradura: si $x, y \in \mathcal{A}$ entonces $x \oplus y \in \mathcal{A}$.
  (2) Asociatividad: $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ para todo $x, y, z \in \mathcal{A}$.
  (3) Elemento neutro: existe un único elemento $e \in \mathcal{A}$ tal que $e \oplus x = x \oplus e = x$, para todo $x \in \mathcal{A}$
  (4) Elemento inverso: para cada $x \in \mathcal{A}$ existe un único elemento $-x \in \mathcal{A}$ tal que $x \oplus -x = e$.
- En este caso se dice que la estructura $(\mathcal{A}, \oplus)$ es un **grupo**.

Alfabetos de mayor cardinalidad (3)

- Si además se tiene que la operación es conmutativa, esto es,

$$x \oplus y = y \oplus x$$

para todo $x, y \in \mathcal{A}$ entonces el grupo es abeliano.

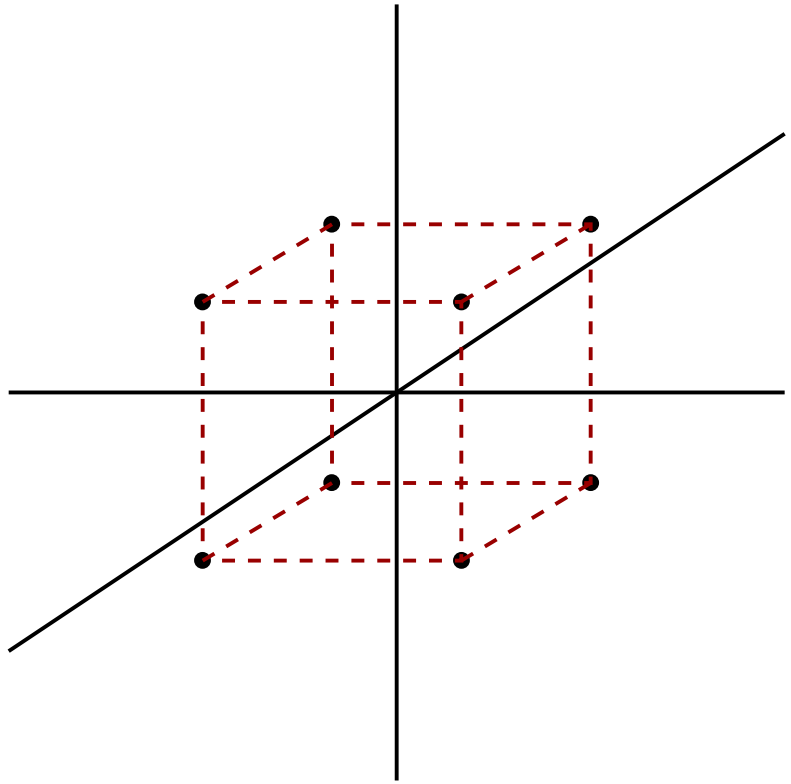
- Cuando $(\mathcal{A}, \oplus)$ y $(\mathcal{A}, \otimes)$ son grupos abelianos, y además $\otimes$ distribuye con respecto a $\oplus$:

$$x \oplus (y \otimes z) = (x \oplus y) \otimes (x \oplus z). \tag{2}$$

decimos que $(\mathcal{A}, \oplus, \otimes)$ es un **campo finito** o **campo de Galois** y lo denotamos por $\mathbb{F}_q$.

Alfabetos de mayor cardinalidad (4)

- Esta estructura nos permite algo bastante poderoso: agrupar elementos en tuplas de largo k, que son elementos de un espacio vectorial
- Este espacio vectorial, denotado como $\mathbb{F}_q^k$ tiene cardinalidad finita igual a q^k.



Problema de codificación (1)

- Vamos a considerar dos tipos de secuencias de símbolos:
  - Secuencia de datos: secuencia de símbolos a la entrada de un codificador.
  - Secuencia codificada: secuencia de símbolos a la salida del codificador.
- El codificador es un dispositivo que mapea secuencias de datos en una secuencia codificada.
- Esta asignación puede realizarse de dos formas:
  - codificando por bloques.
  - codificando en forma secuencial utilizando un árbol de decisión.

Problema de codificación (2)

- **Problema Central**: Dada una secuencia de datos representada en forma binaria, el objetivo de la codificación es agregar redundancia de forma de poder determinar si la secuencia recibida contiene errores y eventualmente corregirlos.
- Un **código binario** de tamaño M y largo de bloque n es un conjunto de M palabras binaria, cada una con n bits.
- Si $M = 2^k$, entonces los denotamos código binario (n, k).

Problema de codificación (3)

- El cuociente

$$R = \frac{n}{k} \tag{3}$$

  recibe el nombre de tasa del código.

- Diremos que se ha cometido un error si la secuencia recibida difiere en una o más posiciones con respecto de la secuencia transmitida.

Ejemplo (1)

$$\mathcal{C} = \left\{ \begin{array}{l} (1\ 0\ 1\ 0\ 1) \\ (1\ 0\ 0\ 1\ 0) \\ (0\ 1\ 1\ 1\ 0) \\ (1\ 1\ 1\ 1\ 1) \end{array} \right\}, \quad M = 4,\ n = 5.$$

- Este código nos permite representar 4 palabras binarias de datos de largo 2 bits.

Ejemplo (2)

- Por ejemplo

$$0\ 0 \longleftrightarrow 1\ 0\ 1\ 0\ 1$$

$$0\ 1 \longleftrightarrow 1\ 0\ 0\ 1\ 0$$

$$1\ 1 \longleftrightarrow 0\ 1\ 1\ 1\ 0$$

$$1\ 0 \longleftrightarrow 1\ 1\ 1\ 1\ 1.$$

Construcción de buenos códigos (1)

- El objetivo de la teoría de códigos es encontrar "buenos" códigos.
- ¿Qué se entiende por buenos?
- Que corrijan un alto número de errores manteniendo una tasa cercana a 1.
- ¿Cómo podemos desarrollar esta búsqueda?
- Digamos que tenemos un código binario (n, k) y queremos buscar todos los códigos que difieran en al menos d posiciones respecto de las otras palabras.

Construcción de buenos códigos (2)

- Una opción para buscar estos códigos es dejar que un computador haga una búsqueda exhaustiva de todas las combinaciones posibles.
- ¿De qué orden sería esta búsqueda?
- Existen 2^k palabras en cada código en cada código binario (n, k).
- Por lo tanto, el código tiene $n \times 2^k$ símbolos binarios.
- P: ¿De cuántas formas uno puede elegir esto $n \times 2^k$ símbolos?
- R: $2^{n \times 2^k}$.
- Por lo tanto, el espacio de búsqueda es de tamaño $2^{n \times 2^k}$ códigos binarios (n, k).

Construcción de buenos códigos (3)

- Si $n = 40$ y $k = 20$, entonces existen $2^{10,000,000}$ códigos binarios $(40, 20)$.
- Los métodos de fuerza bruta son impracticables con estos números.

Códigos de Bloque

[Definición]

Un código de bloque de tamaño M sobre un alfabeto con q símbolos, es un conjunto con secuencias q-arias de largo n llamados **palabras de código**.

Distancia de Hamming (1)

[Definición]

La **distancia de Hamming** $d(\mathbf{x}, \mathbf{y})$ entre dos secuencias q-arias de largo n, $\mathbf{x}$ e $\mathbf{y}$, es el número de lugares en que ellas difieren, esto es,

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \mathbf{1} x_i \neq y_i(i). \tag{4}$$

- La distancia de Hamming satisface las condiciones de una métrica:
  - $d(\mathbf{x}, \mathbf{y}) \geq 0$, con $d(\mathbf{x}, \mathbf{y}) = 0$ ssi $\mathbf{x} = \mathbf{y}$.
  - $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

Distancia de Hamming (2)

- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

[Definición]

La **distancia mínima** de un código $\mathcal{C} = \{\mathbf{c}_l |\ l = 0, 1, \ldots, M-1\}$ es la distancia de Hamming mínima entre todos las combinaciones de palabras de código:

$$d_{\text{min}} = \min_{\mathbf{c}_i, \mathbf{c}_j \in \mathcal{C},\ i \neq j} d(\mathbf{c}_i, \mathbf{c}_j). \tag{5}$$

Distancia de Hamming (3)

- Los códigos de bloque son usualmente juzgados por 3 parámetros: n, k and d_{min}.
- Frecuentemente se los denota por $(n, k, d_{\text{mín}})$.

Cota fundamental para el número de errores (1)

- Sea $\mathcal{C}(n, k, d_{\mathsf{min}})$ un código de bloque.
- Supongamos que se transmite $\mathbf{c}_l \in \mathcal{C}$, para algún l.
- Sea $\mathbf{r}$ la palabra recibida y supongamos que $d(\mathbf{c}_l, \mathbf{r}) = 1$.
- Supongamos además que la distancia con todas las otras palabras de código con r

$$d(\mathbf{c}_j, \mathbf{r}) \geq d(\mathbf{c}_l, \mathbf{r}), \ \forall \ j \neq l,$$

- Entonces, podríamos corregir un error pues $\mathbf{c}_l$ es la palabra más cercana a $\mathbf{r}$.

Cota fundamental para el número de errores (2)

- Consideremos ahora que

$$d(\mathbf{c}_l, \mathbf{r}) = t \tag{6}$$

y que

$$d(\mathbf{c}_j, \mathbf{r}) > t, \ \forall \ j \neq l.$$

- Entonces podemos corregir hasta t erores.
- Notemos que

$$d_{\text{min}} = \min_{\mathbf{c}_i, \mathbf{c}_j \in \mathcal{C}, \ i \neq j} d(\mathbf{c}_i, \mathbf{c}_j).$$

Cota fundamental para el número de errores (3)

- Como el número de combinaciones de índices i y j es finito, existen dos palabras de código $\mathbf{c}_l$ y $\mathbf{c}_k$ tal que

$$d_{\mathsf{min}} = d(\mathbf{c}_l, \mathbf{c}_k). \tag{7}$$

- En el peor de los casos, $\mathbf{r}$ está en la línea que une $\mathbf{c}_l$ con $\mathbf{c}_k$.

Cota fundamental para el número de errores (4)

- Luego

$$d(\mathbf{c}_l, \mathbf{c}_k) = d(\mathbf{c}_l, \mathbf{r}) + d(\mathbf{r}, \mathbf{c}_k) \tag{8a}$$

$$> t + t \tag{8b}$$

$$\geq 2t + 1 \tag{8c}$$

$$\Rightarrow d_{\mathsf{min}} \geq 2t + 1. \tag{8d}$$

Interpretación geométrica (1)

- Consideremos el espacio de n-tuplas de símbolos sobre el alfabeto $\mathcal{A}_q = \{0, 1, \ldots, q-1\}$.
- Definiremos la **esfera de Hamming** centrada en $\mathbf{v}$ de radio t como:

$$\text{Esf}_t(\mathbf{v}) = \{\mathbf{w} \in \mathbb{F}_q^n |\ d(\mathbf{v}, \mathbf{w}) \leq t\}. \tag{9}$$

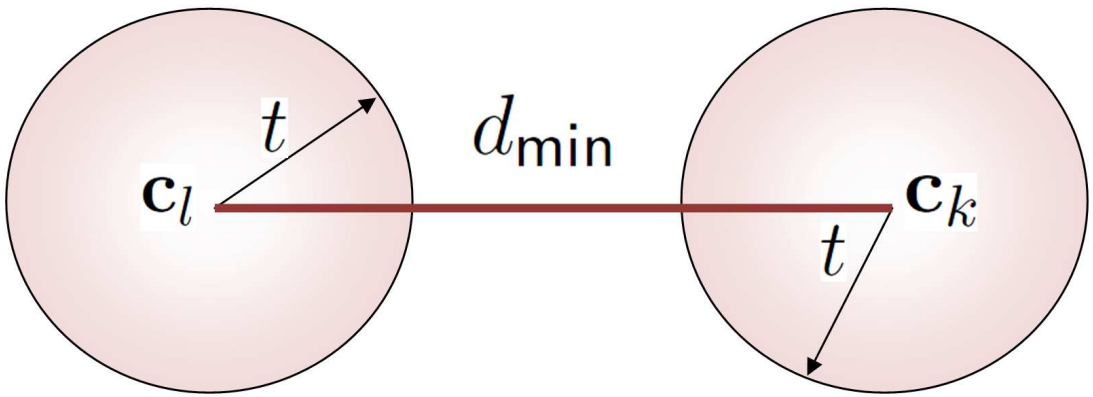
- Consideremos ahora un código en $\mathbb{F}_q^n$ con $M = 2^k$ palabras de código.
- La condición para que las esferas no se intersecten es

$$d(\mathbf{c}_l, \mathbf{c}_k) \geq 2t + 1,\ \forall \mathbf{c}_l, \mathbf{c}_k. \tag{10}$$

Interpretación geométrica (2)

- En particular, esta condición se cumple para el mínimo:

$$d_{\mathsf{min}} \geq 2t + 1. \tag{11}$$



Resumen

Hemos revisado:

- Introducción a la codificación para el control de errores.
- Nociones sobre canales discretos.
- Principios básicos de codificación de bloques: tasa y distancia mínima.

Lecturas

- Proakis & Salehi, *Communication System Engineering*, capítulo 9, secciones 9.4 y 9.5.
- R. Blahut, *Algebraic Codes for Data Transmission*, capítulos 1 y 3.
- W. Peterson & E. Weldon Jr, *Error-Correcting Codes*, capítulo 1.