

Códigos para detección y corrección de errores en comunicaciones digitales

Raúl Alvarado Escamilla
FIME-UANL
ralvarad@gama.fime.uanl.mx

RESUMEN

El objetivo del presente trabajo es generar programas basados en MATLAB y DIGCOM-T que permitan analizar la operación de distintos tipos de códigos para detección y corrección de errores. En particular se estudia la familia de códigos de bloque conocida como Hamming, así como también los códigos cíclicos Bose-Chaudhuri-Hocquenqhem y Reed Solomon. De igual modo, es visto el código convolucional junto con el diagrama de Trellis usado para representar su operación.

PALABRAS CLAVE

Procesamiento de señales, códigos, detección, error.



ABSTRACT

The purpose of this study is to make several programs based in MATLAB and DIGCOM-T, which make possible to analyze error detection and correction codes operation. In particular it is studied the family of block codes known as Hamming, as well as the cyclic codes Bose-Chaudhuri-Hocquenqhem and Reed Solomon. The convolution code and the Trellis diagram used to represent its operation, are also seen in this study.

KEYWORDS

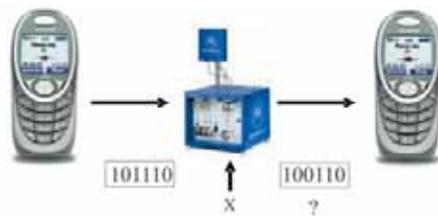
Signal processing, code, detection, error.

INTRODUCCIÓN

Las operaciones básicas de procesamiento de señales en un sistema de comunicación digital son *codificación de la fuente, codificación de canal y modulación digital* en el lado transmisor, como se muestra en el diagrama a bloques de la figura 1, así como los procesos inversos en el lado receptor. Los códigos para detección y corrección de errores corresponden a la codificación de canal.

Uno de los parámetros que miden el desempeño de un sistema de comunicación digital es la tasa de error binaria (BER) para un cierto valor de la relación de energía de bit a densidad espectral de ruido (E_b/N_0). Generalmente, debido al ruido del canal de transmisión, la única manera de proporcionar un valor de BER aceptable es agregando códigos para detección y corrección de los errores.

Actualmente se emplean dos formas de controlar los errores. Una de las técnicas es conocida como *solicitud de confirmación (ARQ de Acknowledgment Request)*, ya que el receptor comprueba los datos recibidos y si hubo error



Sistema de comunicación digital.

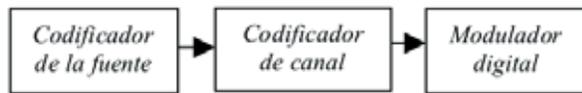


Fig. 1. Procesos de la señal en el lado transmisor de un sistema de comunicaciones.

solicita una retransmisión, de lo contrario retorna una confirmación de recepción correcta. La otra técnica se conoce como corrección de error hacia adelante (FEC, de Forward Error Correction), y es utilizada en los sistemas de comunicación que operan en tiempo real (es decir, el transmisor no almacena los datos que envía como en el caso de voz y vídeo digitales). Por cuestiones de extensión, sólo se abordarán los códigos FEC. En general, el proceso de codificación en los códigos FEC es una operación en la cual a un grupo de bits de datos correspondientes al mensaje que se desea transmitir, se le agrega un grupo de bits conocidos como de paridad para fines de detección y corrección de errores.

El análisis y diseño de los códigos se hará por medio del software DIGCOM-T,¹ el cual es un conjunto de programas que corren sobre la plataforma de MATLAB. Asimismo, la exposición del desarrollo de los experimentos se llevará a cabo describiendo primeramente los conceptos fundamentales, los parámetros principales y la estructura matemática de cada uno de los códigos. Enseguida se efectuará un planteamiento del problema que se desea resolver y se elaborará el programa para la solución del problema con *Matlab* y *Digcom-T*. Posteriormente se realizará el análisis de los resultados del programa donde se enfatizan las características más importantes del código. Finalmente se resumirán las conclusiones generales acerca de los códigos y se especificarán las principales aportaciones del presente estudio.

CÓDIGO HAMMING

El código Hamming es clasificado como un código de bloque porque tiene como entrada un

grupo (bloque) de m bits de datos, a los cuales se le agrega un grupo de k bits de paridad de acuerdo a reglas preestablecidas, dando como salida un grupo de n bits comúnmente llamado palabra-código. Asimismo, se dice que es lineal porque satisface la condición de que cualquier par de palabras-código al ser sumadas en módulo 2 producen otra palabra-código existente dentro del conjunto empleado para codificar los posibles mensajes. Además, se dice que es sistemático porque los bits de paridad se agregan sin alterar la posición de los bits del mensaje.

Para un código lineal de bloque (n, k) se pueden ordenar los bits de paridad a la derecha y los de mensaje a la izquierda (o viceversa) como se muestra en la figura 2.

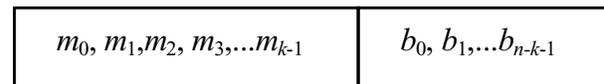


Fig.2. Estructura de la palabra-código.

Los códigos Hamming comúnmente son empleados en aplicaciones donde ocurre un sólo error por bloque, como por ejemplo en el proceso de escritura y lectura de una memoria RAM. Una familia de estos códigos tiene los siguientes parámetros:²

Tamaño del bloque	$n = 2^m - 1$	bits
Bits de paridad	$m ; (m \geq 3)$	bits
Tamaño del mensaje	k	bits
Distancia mínima	$d_{min} \geq 2t + 1$	bits
Errores a corregir	t	bits

Como se observa en los parámetros, el número de errores t que se pueden corregir por bloque de palabra-código está relacionado con la distancia Hamming d . Donde esta distancia es el número de bits distintos entre los vectores que representan las palabras-código. Asimismo, debido a la propiedad de linealidad la distancia Hamming mínima se puede obtener con el peso mínimo de los vectores no ceros. A su vez, el peso Hamming es obtenido con el número



de bits que difieren los vectores de las palabras-código con respecto al vector de puros ceros.

Estructura algebraica

Los n bits de la palabra-código x_i se obtienen con la relación de la ecuación 1.

$$x_i = \begin{cases} m_i & i = 0, 1, \dots, k-1 \\ b_{i-k} & i = k, k+1, \dots, n-1 \end{cases} \quad (1)$$

Los $(n-k)$ bits de paridad son sumas lineales de los k bits del mensaje, como se muestra en la ecuación 2, en la cual se ha sustituido l por $n-k$ por conveniencia en la representación.

$$b_l = p_{l0}m_0 + p_{l1}m_1 + \dots + p_{l, k-1}m_{k-1} \quad (2)$$

$l = 0, 1, \dots, n-k-1; j = 0, 1, \dots, k-1$

donde $p_{lj} = 1$ si b_l depende de m_j y o de otro modo

Los coeficientes de p_{lj} se escogen de manera que las filas de la matriz generadora sean linealmente independientes y las ecuaciones de paridad sean únicas para que el código pueda ser decodificable en forma única en el receptor.

Por simplicidad, es conveniente hacer una representación compacta en forma de vectores y matrices de la estructura. De manera que los vectores de mensaje, paridad y palabra-código quedan como se muestra en la ecuación 3.

$$\begin{aligned} m &= [m_0, m_1, \dots, m_{k-1}] \\ b &= [b_0, b_1, \dots, b_{n-k-1}] \\ x &= [x_0, x_1, \dots, x_{n-1}] \end{aligned} \quad (3)$$

A su vez, las ecuaciones que definen los bits de paridad quedan como lo muestra la ecuación 4.

$$b = mP \quad (4)$$

donde P es una matriz de coeficientes de tamaño k por $n-k$ de la forma siguiente:

$$P = \begin{bmatrix} p_{00} & p_{01} & \dots & p_{0, n-k-1} \\ p_{10} & p_{11} & \dots & p_{1, n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1, 0} & p_{k-1, 1} & \dots & p_{k-1, n-k-1} \end{bmatrix}$$

Generación de la palabra-código en el lado transmisor

La palabra-código en forma compacta puede ser expresada como lo indica la ecuación 5.

$$x = [m;b] \quad (5)$$

Sustituyendo $b = mP$ obtenemos que:

$$x = [m;mP] = m[I_k:P]$$

en la cual I_k es una matriz identidad de tamaño $k \times k$. Ahora bien, definiendo la matriz generadora como se indica en la ecuación 6.

$$G = [I_k:P] \quad (6)$$

donde G debe estar en forma canónica

Entonces el vector de la palabra-código se obtiene multiplicando el vector del mensaje por la matriz generadora como se muestra en la ecuación 7.

$$x = mG \quad (7)$$

Con esto finaliza la operación de codificación de canal en el lado transmisor de un sistema de comunicaciones.

Detección de error en el lado receptor

Si los coeficientes que relacionan los bits de paridad y de mensaje (P) se expresan en forma transpuesta y definimos la matriz H como en la ecuación 8.

$$H = [P^T : I_{n-k}] \quad (8)$$

Enseguida, postmultiplicamos H por la transpuesta de G y obtenemos que:

$$HG^T = [P^T : I_{n-k}] \begin{bmatrix} I_k \\ \dots \\ P^T \end{bmatrix} = P^T + P^T$$

En aritmética módulo 2: $P^T + P^T = 0$. De aquí que $HG^T = 0$ (o bien $GH^T = 0$) donde 0 es una matriz nula de tamaño $n-k$ por k .

A continuación, postmultiplicamos $x = mG$ por H^T y se obtiene $xH^T = 0$, donde H es la matriz de comprobación de paridad y $xH^T = 0$ es la ecuación de comprobación de paridad empleada por el receptor para la detección de errores.

Por otra parte, para simplificar el algoritmo que realiza la detección de errores, se recomienda hacer el cálculo de un valor conocido como síndrome como lo muestra la ecuación 9.

$$s = yH^T \quad (9)$$

donde y es la palabra-código recibida posiblemente con error.

Finalmente, con el síndrome y el patrón de errores más comunes se identifica la posición del error. Por ejemplo, el patrón de errores, matriz e y su síndrome, matriz s para un código (7,4) son mostrados en la tabla I.

Tabla I. Patrón de errores más comunes y el síndrome para un código (7,4).

Patrón de errores (e)							Síndrome s
Posición de los bits							
1	2	3	4	5	6	7	
0	0	0	0	0	0	0	0 0 0
1	0	0	0	0	0	0	1 1 1
0	1	0	0	0	0	0	1 1 0
0	0	1	0	0	0	0	1 0 1
0	0	0	1	0	0	0	0 1 1
0	0	0	0	1	0	0	1 0 0
0	0	0	0	0	1	0	0 1 0
0	0	0	0	0	0	1	0 0 1

Al síndrome $s = 000$, le corresponde el valor de $e = 0000000$, lo cual significa para el lado receptor que no hubo error en la transmisión, mientras que 111, indica que hay un error en la posición uno de la palabra-código y así sucesivamente el resto de las combinaciones del síndrome indican un error en la posición donde e vale uno.

Experimento 1: Considere que en el lado transmisor se emplea un código Hamming (7,4), y que existe un error de un bit en el canal de transmisión. Obtener la siguiente información:

1. La Matriz generadora.
2. La Matriz de comprobación de paridad.
3. Las palabras-código.
4. El número de errores que puede corregir.
5. Corrección de un error en la recepción.

El programa elaborado para resolver el problema fue nombrado *EjCodigoHamming*.³ En él se emplea la función $[G,H] = \text{HammingCode}(n)$ del software DIGCOM-T. Los resultados arrojados por el programa son mostrados en las tablas II y III.

Análisis de resultados del experimento uno

La organización de las palabras-código se muestra en la tabla III.

Las características principales del código

Tabla II. Resultados del programa para el experimento uno.

»Matriz generadora						
1	0	0	0	1	1	1
0	1	0	0	1	1	0
0	0	1	0	1	0	1
0	0	0	1	0	1	1
Matriz de comprobación de paridad						
1	1	1	0	1	0	0
1	1	0	1	0	1	0
1	0	1	1	0	0	1
Dato de la palabra-código con error (entre 1 y 2^k)						
= 3						
Dato de la posición del bit de error (entre 1 y n)						
= 4						
Distancia Hamming mínima de x						
= 3						
Número de errores que se pueden corregir						
t=1						
Palabra-código recibida con un error						
0 1 0 1 1 1 0						
Síndrome para detección de error s =						
0 1 1						
Palabra-código con corrección del error corr =						
0 1 0 0 1 1 0						

Tabla III. Palabras-código del código Hamming (7,4).

Número de la palabra	Palabra-Código							
	Posición	1	2	3	4	5	6	7
1		0	0	0	0	0	0	0
2		1	0	0	0	1	1	1
3		0	1	0	0	1	1	0
4		1	1	0	0	0	0	1
5		0	0	1	0	1	0	1
6		1	0	1	0	0	1	0
7		0	1	1	0	0	1	1
8		1	1	1	0	1	0	0
9		0	0	0	1	0	1	1
10		1	0	0	1	1	0	0
11		0	1	0	1	1	0	1
12		1	1	0	1	0	1	0
13		0	0	1	1	1	1	0
14		1	0	1	1	0	0	1
15		0	1	1	1	0	0	0
16		1	1	1	1	1	1	1

Hamming que podemos observar en las tablas II y III de este experimento son las siguientes:

- El código obtenido es sistemático puesto que los bits del mensaje m_0, m_1, m_2 y m_3 están en las posiciones 1 a 4 de las palabras código obtenidas,

sin ser alteradas por los bits de paridad.

- El código obtenido es lineal, lo cual se puede comprobar tomando como ejemplo la palabra-código 3 (0100110) y la 6 (1010010), las cuales al sumarlas se obtiene la 8 (1110100).
- Como la distancia Hamming mínima es 3, entonces un código Hamming solo puede corregir 1 error en la transmisión ($t=1$).
- La detección del error se obtiene con el síndrome 011, cuyo patrón de error es 0001000 (ver tabla I), indicando que el error está en la posición 4.
- La corrección del error se obtuvo sumando en módulo 2 el valor del patrón de error (0001000) con la palabra-código recibida con error (0101110), es decir la palabra corregida es 0100110.

CÓDIGOS CÍCLICOS

Son una subclase de los códigos de bloque lineales,⁴ los cuales tienen esquemas de decodificación eficientes, es decir con algoritmos relativamente simples. Se dice que un código es cíclico cuando cualquier desplazamiento en lazo cerrado de una palabra-código da como resultado otra palabra-código existente dentro del conjunto empleado para codificar los posibles mensajes.

Existen una gran variedad de códigos cíclicos. Por ejemplo, el Código de Redundancia Cíclica empleado en comunicaciones de datos y el código Golay que es un código binario como el Hamming. Además, están los códigos Q-arios como el Bose-Chaudhuri-Hocquenghem y el Reed Solomon. Dada la versatilidad de parámetros de estos dos últimos son los que se seleccionaron para ser analizados en las secciones siguientes.



Telefonía para comunicación de voz IP. Código de redundancia cíclica en su interior.

Polinomios de palabra-código

La representación matemática de la operación de los códigos cíclicos está basada en el uso de polinomios.

Los elementos de una palabra-código de tamaño n pueden ser los coeficientes de un polinomio de grado $n-1$. Por ejemplo, la palabra-código con elementos x_0, x_1, \dots, x_{n-1} puede ser representada en forma de polinomio como:

$$X(D) = x_0 + x_1D + \dots + x_{n-1}D^{n-1}$$

donde D es una variable Real arbitraria

Polinomio generador

Un código cíclico (n, k) es especificado por un conjunto de polinomios de palabra-código de grado $n-1$ o menos, el cual contiene un polinomio de grado mínimo $n-k$ como un factor. Este factor especial, denotado por $g(D)$ es seleccionado como el Polinomio Generador del código.

Procedimiento para codificación de un código cíclico

Multiplicar el polinomio del mensaje $m(D)$ por D^{n-k}

$$D^{n-k} m(D) = m_0D^{n-k} + m_1D^{n-k+1} + \dots + m_{k-1}D^{n-1}$$

Dividir $D^{n-k} m(D)$ por el polinomio generador $g(D)$, obteniendo el residuo $b(D)$

$$\frac{D^{n-k} m(D)}{g(D)} = a(D) + \frac{b(D)}{g(D)}$$

Agregar $b(D)$ a $D^{n-k} m(D)$ para obtener el polinomio de la palabra-código $x(D)$.

$$x(D) = b(D) + D^{n-k} m(D)$$

Cálculo del síndrome para detección de error

Considerando que la palabra-código recibida con error sea $Y(D) = y_0 + y_1D + \dots + y_{n-1}D^{n-1}$, el polinomio del síndrome se obtiene con el residuo de la división del polinomio de la palabra-código entre el polinomio generador $g(D)$.

$$\frac{Y(D)}{g(D)} = q(D) + \frac{s(D)}{g(D)}$$

donde q es el cociente y s es el síndrome

De manera similar al código Hamming, con el síndrome y el patrón de errores se hace la detección

del error o los errores y su posición dentro de la palabra-código.

CÓDIGO BOSE - CHAUDHURI -HOCQUENQHEM (BCH)

Es un código cíclico con una gran variedad de parámetros. Para $m \geq 3$ y $t < (2m-1)/2$ existen códigos *BCH* con los siguientes parámetros:

Para tamaños de unos pocos cientos de bits o menos, los códigos *BCH* son de los mejores para un mismo tamaño de bloque e índice de código (relación entre el bloque de entrada y el bloque de salida). Algunos códigos comunes⁵ expresados en la forma (n, k, t) de *BCH* son: (7, 4, 1), (15, 11, 1), (15, 7, 2), (15, 5, 3), (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5) y (31,6, 7).

Experimento 2

Considere que en el lado transmisor se emplea un

Tamaño del bloque	$n = 2^m - 1$	bits
Bits de paridad	$n - k$	bits
Tamaño del mensaje	$K \geq n - mt$	bits
Distancia mínima	$d_{min} \geq 2t + 1$	bits
Errores a corregir	t	bits

código *BCH* (15, 7, 2). Obtener lo siguiente:

1. El polinomio generador.
2. La palabra-código generada para el mensaje 1100110.
3. La palabra-código recibida con dos errores.
4. Estimación del mensaje enviado.
5. Estimación de la palabra-código enviada.

El programa elaborado para resolver el problema fue nombrado *EjCodigoBCH*.³ En el se emplean las funciones *BinaryBCHgen*(n, d_star) y *SystematicEncode*(q, g, m) del software DIGCOM-T. Los resultados arrojados por el programa se muestran en la tabla IV:

Análisis de resultados del experimento dos

La organización de la palabra código es mostrada en la siguiente tabla:

Palabra-código															
Posición	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	1	1	0	1	1	1	1	1	0	0	1	1	0

Tabla IV. Resultados del programa para el experimento dos.

»El polinomio generador es=	1	0	0	0	1	0	1	1	1
Datos del mensaje =	[1	1	0	0	1	1	0]		
La palabra-código es =	0	0	1	1	0	1	1	1	1
Dato de la posición de los t bits de error (entre 1 y n) =	[3	8]							
Mensaje con error	0	0	0	1	0	1	1	0	1
Estimación del mensaje enviado									
Estimación de palabra-código	0	0	1	1	0	1	1	1	1

Las características principales del código *BCH* que podemos observar en la tabla IV de este experimento son las siguientes:

- El polinomio generador obtenido con 100010111 es $1 + D^4 + D^6 + D^7 + D^8$.
- El código obtenido es sistemático puesto que los bits del mensaje $m_0, m_1, m_2, m_3, m_4, m_5$ y m_6 están en las posiciones 9 a 15 de la palabra código obtenida, sin ser alterada por los bits de paridad.
- Con el *BCH* es posible detectar y corregir más de un error (dos en este ejemplo), en lugar de uno sólo como en los códigos Hamming.

CÓDIGO REED-SOLOMON

Un código Reed Solomon (*RS*) es una subclase de los códigos *BCH* no binarios. Se distingue de los códigos binarios porque éste opera con símbolos de b bits cada uno en lugar de bits individuales. Es útil cuando los errores ocurren en ráfagas, como en los sistemas de grabación de discos compactos digitales.

Un código *RS* que corrige t errores tiene los siguientes parámetros:

Un polinomio generador no binario se define

Tamaño del bloque	$n = 2^b - 1$	símbolos
Tamaño de paridad	$n - k = 2t$	símbolos
Tamaño del mensaje	k	símbolos
Distancia mínima	$d_{min} \geq 2t + 1$	símbolos
Errores a corregir	t	símbolos



Grabadora de CD's y DVD's
Codigo Reed Solomon en su interior.

en función de los campos de Galois⁶ y son denotados por $GF(q)$. Cuando q es primo se construye un campo finito $GF(q)$ cuyos elementos son $\{0,1,\dots,q-1\}$. Las operaciones de suma y multiplicación de los elementos de $GF(q)$ están basadas en módulo q . En general, el campo finito $GF(q)$ sólo puede ser construido si q es primo o una potencia de un primo. Si $q=p^b$, donde p es primo y b es un entero positivo, es posible extender el campo $GF(p)$ al campo $GF(p^b)$. Éste es conocido como campo extendido de $GF(p)$ y sus operaciones de suma y multiplicación están basadas en módulo p .

En particular, en un código RS $q=2^b$, donde q es el número de símbolos y b es el número de bits por símbolo. Por ejemplo, si $q=2^3$, el polinomio para un mensaje compuesto por los símbolos $\{2,5,0,7\}$ será $m(D) = 2 + 5D + 7D^3$.

Experimento 3

Considere que en el lado transmisor se emplea un código Reed Solomon (7, 5), es decir $n=7$ y $k=5$. Obtener lo siguiente:

1. El Polinomio generador.
2. La palabra-código generada para un mensaje con los símbolos del 1 al 5.
3. La palabra-código recibida con un sólo error.
4. Estimación del mensaje enviado.
5. Estimación de la palabra-código enviada.

El programa elaborado para resolver el problema fue nombrado EjCodigoReedSolomon³ En él se emplean las funciones $g = RSgenerator(q, r, 1)$ y $[m_hat, c_hat, n_err] = RSdecode(q, g, y, 1, n)$ del software DIGCOM-T. Los resultados arrojados por el programa son mostrados en la tabla V:

Tabla V. Resultados del programa para el experimento tres.

»El Polinomio generador es =	3	6	1				
Datos del mensaje (i.e. los símbolos del 1 al 5)							
							= [1 2 3 4 5]
Palabra-código =	6	4	1	2	3	4	5
Posición y valor de los t errores (entre 1 y n y de 0 a 2 ^b)							$[p_1, p_2, \dots, v_1, v_2] = [4 6]$
Mensaje con error	6	4	1	6	3	4	5
Estimación del Mensaje enviado							
							1 2 3 4 5
Estimación de la palabra-código enviada							
							6 4 1 2 3 4 5

Análisis de resultados del experimento tres

La organización de los símbolos y bits por símbolo es mostrada en la siguiente tabla:

Valor del Símbolo	Valor en binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Las características principales del código RS que podemos observar con este experimento son las siguientes:

- El polinomio generador obtenido con 3 6 1 es: $3 + 6D + D^2$
- La organización de la palabra-código es mostrada en la siguiente tabla:

Palabra-código							
Posición	1	2	3	4	5	6	7
	6	4	1	2	3	4	5

- El código obtenido es sistemático puesto que los símbolos del mensaje $sm_0, sm_1, sm_2, sm_3,$ y sm_4 están de la posición 3 a la 7 de la palabra-código obtenida, sin ser alterada por los símbolos de paridad.
- En este ejemplo es posible detectar y corregir sólo un símbolo de error $[t=(n-k)/2 = (7-5)/2 = 1]$.

CÓDIGOS CONVOLUCIONALES

Un código convolucional⁷ es visto como una máquina de estados finitos que consta de un registro de corrimiento de M etapas con conexiones preestablecidas a n sumadores módulo 2 y con un multiplexor que serializa la salida de los sumadores. En la figura 3 se muestra un codificador con un registro de $M=2$ y $n=2$.

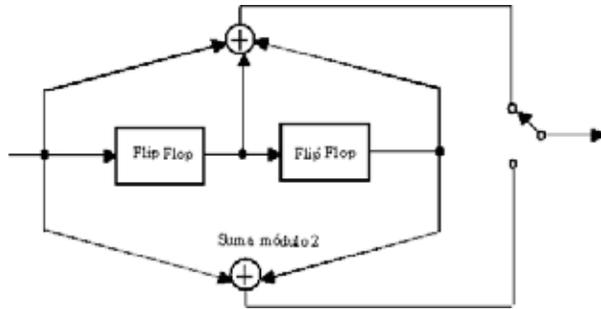


Fig. 3. Codificador convencional de 2 etapas.

Una secuencia de entrada, correspondiente a un mensaje de L bits produce una secuencia de salida codificada de tamaño $n(L + M)$ bits. De manera que la relación entre la entrada y la salida, conocida como índice de código se obtiene con $r=L / n(L+M)$.

Los códigos convolucionales son ampliamente usados en enlaces de radio de línea de vista y en comunicaciones móviles, tal como telefonía celular.



Teléfono celular, código convolucional en su interior.

Respuesta al impulso y polinomio generador

Cada trayectoria que conecta la salida a la entrada puede ser caracterizada en términos de su respuesta al impulso, definida como la respuesta de la trayectoria a un símbolo 1 aplicado a su entrada, con cada flip flop del codificador inicializado en el estado cero. Cada trayectoria también se puede caracterizar en términos de un polinomio generador cuyos coeficientes son los elementos de la respuesta al impulso. La respuesta de un código convolucional

puede ser obtenida en el dominio del tiempo o en el dominio de la frecuencia de acuerdo a la teoría de sistemas discretos lineales e invariantes en el tiempo.⁸ En el tiempo, la salida se obtiene con la convolución de la entrada con la respuesta al impulso, mientras que en la frecuencia se obtiene con el producto de las transformadas de Fourier de la entrada y de la respuesta al impulso (las cuales corresponden a la representación en polinomios).

La operación de un codificador convolucional se puede observar gráficamente mediante un diagrama de árbol, un diagrama de estados, o bien en un diagrama de Trellis como el de la figura 4.

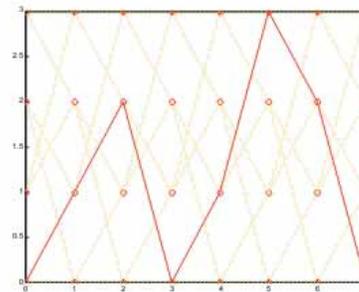


Fig. 4. Trayectoria para el mensaje 10011.

Experimento 4

Para el codificador convolucional de la figura 3 obtener lo siguiente:

1. La respuesta al impulso y el polinomio generador.
2. La gráfica de la trayectoria seguida en el diagrama de Trellis para el mensaje 10011.
3. Cálculo de la palabra-código para el mensaje 10011.

El programa elaborado para resolver el problema fue nombrado EjCodigoConvolucional.³ En el se emplean las funciones $G = \text{BestBCC}(1, 2, 2)$ y $\text{TrellisPath_plot}(m, \text{Edges})$ del software DIGCOM-

Tabla VI. Resultados del programa para el experimento cuatro.

» Respuesta al impulso o polinomio generador					
		1	0	1	
		1	1	1	
La trayectoria del mensaje 10011 (se muestra en la Fig. 4).					
Para el mensaje		1	0	0	1
La palabra-código es:	3	2	3	3	1

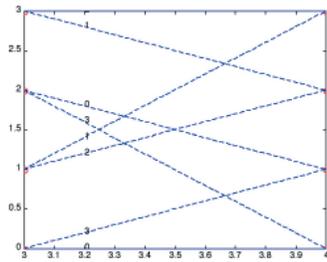


Fig. 5. Código generado por cada rama de mensaje.

T. Los resultados son mostrados en la tabla VI y las figuras 4 y 5.

Análisis de resultados del experimento cuatro

La organización de la palabra-código es mostrada en la siguiente tabla:

Palabra-código en base 4 (Nº de estados)	Palabra-código (en binario)
Posición 1 2 3 4 5 6 7	1 2 3 4 5 6 7 8 9 10 11 12 13 14
3 2 3 3 1 1 3	1 1 1 0 1 1 1 1 0 1 0 1 1 1

Las características principales del código convolucional que podemos observar con este experimento son las siguientes:

- La respuesta al impulso de la trayectoria superior entre la entrada y la salida es $h_1(n) = \{1 \ 1 \ 1\}$; mientras que la de la trayectoria inferior es $h_2(n) = \{1 \ 0 \ 1\}$. A su vez, los respectivos polinomios generadores son $1 + D + D^2$ y $1 + D^2$. Observé que el orden de aparición en la tabla de resultados está invertido con respecto a la ubicación de las trayectorias de la figura 3.
- El código obtenido es no sistemático puesto que los bits del mensaje m_0, m_1, m_2, m_3 , y m_4 (1 0 0 1 1) están mezclados con los bits de paridad en los 14 bits del código binario mostrado en la tabla anterior.
- El diagrama de Trellis de la figura 4 muestra la trayectoria seguida por el mensaje 1 0 0 1 1 (más dos ceros agregados como cola para que el último bit de entrada llegue hasta la salida). Se inicia en el estado cero y en cada bifurcación se sigue una rama de acuerdo al bit de entrada. Si es uno se sigue la rama superior y si es cero, la inferior.
- El código generado por el mensaje se puede obtener siguiendo cada rama de la figura 4 y

consultando el código de la misma en la figura 5, la cual es una sección (de 3 a 4 en este ejemplo) del diagrama de Trellis.

PROGRAMA CON MATLAB Y DIGCOM-T

El programa que se muestra a continuación es el que se ha empleado para resolver el experimento 4 planteado anteriormente. Obsérvese que las instrucciones se acompañan (en forma de comentarios) con una amplia explicación de cómo proporcionar los datos del código, así como de la función que realiza cada una de ellas.

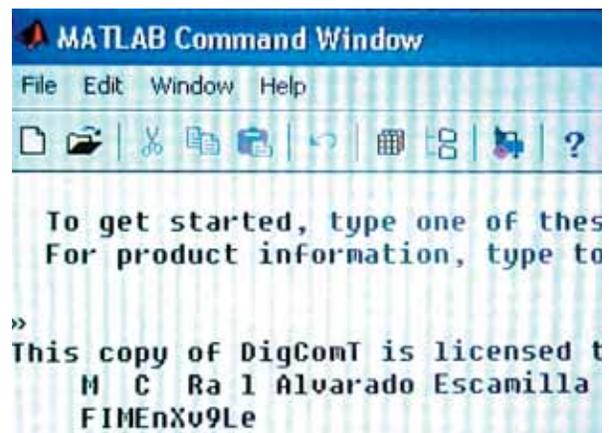
Con este tipo de programas se facilita el análisis y diseño de los códigos para detección y corrección de errores, ya que es fácil reprogramar con otros parámetros para efectos de comparación y evaluación de su desempeño.

Es conveniente recordar que el software DIGCOM-T es de un propietario independiente de MATLAB. Por lo mismo, se debe disponer de éste para que el programa pueda correr correctamente.

%CÁLCULO DE LA PALABRA-CÓDIGO DE UN CODIGO CONVOLUCIONAL Y SU REPRESENTACIÓN POR MEDIO DEL DIAGRAMA DE TRELIS

```
% Obtener el Polinomio Generador (o la respuesta al impulso) para un código convolucional binario.
% Óptimo con G = BestBCC(k, n, nu).
% k=1 (un solo registro de corrimiento entre entrada - salida).
% n= 2 (dos sumadores en módulo 2).
% nu = 2 (log base 2 del número de estados (4 estados en este caso)).
```

```
G = BestBCC(1, 2, 2);
Gb=OctaltoVector(G);
disp('Respuesta al Impulso o Polinomio
Generador');disp(Gb)
```



```

% Calcular las transiciones (edges) de la máquina de
estados finita con Edges=makeBCC
Edges = makeBCC(G);
disp ('tabla de transiciones de la forma [estado actual
entrada salida estado siguiente]'); disp (Edges);
% El diagrama de estados se puede obtener a partir
de una parte del diagrama de Trellis con disp ('Sección
del diagrama de Trellis para obtener el diagrama de
estados (Figura)');
Trellis_plot(Edges, [3 4], 3)
% Con el diagrama de estados, empezando por el Cero
y siguiendo los estados de acuerdo al mensaje
% Se obtiene el código de salida
% Para visualizar la trayectoria dentro del diagrama de
Trellis se usa TrellisPath_plot(message, Edges)
% donde al mensaje de este ejemplo se le agregan
dos bits (a la derecha) de cola para que el último
bit del mensaje termine su recorrido en el Registro de
Corrimiento
m=[1 0 0 1 1 0 0];
disp('Mensaje');disp(m);
figure
disp('Trayectoria del mensaje 10011 en el diagrama de
Trellis (Figura)');
TrellisPath_plot(m, Edges)
%Codificación empleando programas ya hechos por
DigComT
%las funciones requeridas son [Table_out, Table_ns] =
EdgesToEncoderTable(Edges) y
%c = EncodeFromTable(m, Table_out, Table_ns)
[Table_out, Table_ns] = EdgesToEncoderTable(Edges);
c = EncodeFromTable(m, Table_out, Table_ns);
disp('Palabra-código');disp(c)

```

CONCLUSIONES Y APORTACIONES

En el presente trabajo se elaboraron cuatro programas basados en MATLAB y DIGCOM-T para analizar la codificación de canal de un sistema de comunicaciones digitales.

Con el primero, llamado EjCodigoHamming, se generaron las palabras-código en el lado transmisor empleando una matriz generadora, asimismo se simuló un error en el canal de transmisión y luego en el lado receptor se utilizaron el síndrome y el patrón de errores para detectar y corregir un sólo error por bloque. Con el segundo, denominado EjCodigoBCH se generó una palabra-código empleando un polinomio generador, se simularon dos errores en el canal de transmisión y luego se utilizaron el síndrome y el patrón de errores para detectar y corregir los dos errores.

De manera similar, con el tercer programa llamado EjCodigoReedSolomon, se analizó la detección y corrección de errores en una comunicación con símbolos correspondientes a varios bits cada uno. A su vez, con el programa EjCodigoConvolutivo se obtuvo la palabra-código realizando la convolución entre el mensaje y las respuestas al impulso de un codificador de dos etapas de memoria y se analizó la operación del codificador por medio del diagrama de Trellis.

Por lo tanto, se ha comprobado que el análisis con MATLAB y DIGCOM-T es un excelente complemento didáctico para la enseñanza de la teoría de las comunicaciones digitales, ya que al utilizar este tipo de programas en la solución de problemas se logra consolidar la comprensión de los conceptos básicos y las características principales de los códigos.

Asimismo, se da a conocer el uso de una herramienta computacional poderosa para analizar y diseñar códigos que detectan y corrigen errores, misma que podría ser usada para desarrollar nuevas técnicas de codificación de canal que mejorarían la confiabilidad de los sistemas de comunicación digitales existentes o futuros.

BIBLIOGRAFÍA

1. Chris Heegard, "DigComT: The Digital Communication Toolbox Manual", included in DigComT software, Native Intelligence, 2000.
2. Simon Haykin, "Digital Communications", pag. 378, John Wiley & Sons, 1988.
3. Raúl Alvarado, "Software para comunicaciones digitales", disponible a solicitud expresa en ralvarad@gama.fime.uanl.mx
4. Leon W. Couch, "Sistemas de comunicación digitales", pag. 653, 4th edition, Prentice Hall, 2001.
5. Symon Haykin, "Communication Systems", pag. 653, 4th edition, John Wiley & Sons, 2001.
6. Marvin K. Simon, Sami M. Hinedi and William C. Lindsey, "Digital Communication Techniques; Signal Design and Detection", pag 747, Prentice Hall, 1995.
7. John G. Proakis, "Digital Communication", pag. 470, 3th edition, Mc Graw Hill, 1995.
8. John G. Proakis, "Tratamiento Digital de Señales", 3th edición, Prentice-hall, 1998.