

UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA  
EL7041-1 COMUNICACIONES DIGITALES AVANZADAS

## Proyecto 3

---

# Simulación de un Sistema de Comunicación Digital Empleando Codificación de Canal

**Alumno:** Hernán Arraño S.  
**Profesor:** César Azurdia M.  
**Fecha:** 29 de junio de 2014

# Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>  | <b>2</b>  |
| <b>2. Marco Teórico</b>   | <b>3</b>  |
| 2.1. Codificación e <i>Interleaving</i> para <i>Fading Channels</i> . . . . . | 4         |
| 2.1.1. Codificación de bloque con <i>interleaving</i> . . . . .               | 5         |
| 2.1.2. Codificación convolucional con <i>interleaving</i> . . . . .           | 6         |
| <b>3. Parámetros y Métodos de Simulación</b>                                  | <b>7</b>  |
| 3.1. Pasos código de Hamming . . . . .  | 8         |
| 3.2. Pasos código Convolucional . . . . .                                     | 8         |
| <b>4. Resultados de Simulaciones</b>  | <b>9</b>  |
| 4.1. Código de Hamming(7,4) . . . . .   | 9         |
| 4.1.1. Modulación BPSK . . . . .  | 9         |
| 4.1.2. Modulación 16-QAM . . . . .  | 9         |
| 4.2. Código Convolucional . . . . .   | 10        |
| 4.2.1. Modulación BPSK . . . . .  | 10        |
| 4.2.2. Modulación 16-QAM . . . . .  | 10        |
| 4.2.3. Comparaciones entre código de Hamming(7,4) y Convolucional: . . . . .  | 11        |
| <b>5. Análisis de los Resultados</b>  | <b>12</b> |
| <b>6. Conclusiones</b>  | <b>12</b> |
| <b>7. Anexos: Códigos</b>   | <b>13</b> |
| 7.1. Simulación para BPSK y código Hamming(7,4) . . . . .                     | 13        |
| 7.2. Simulación para 16-QAM y código Hamming(7,4) . . . . .                   | 16        |
| 7.3. Simulación para BPSK y código Convolucional . . . . .                    | 18        |
| 7.4. Simulación para 16-QAM y código Convolucional . . . . .                  | 22        |
| <b>Referencias</b>  | <b>26</b> |

## 1. Introducción

Como ya se ha visto anteriormente, la comunicación sobre un canal ruidoso está sujeta a errores, pero si uno a eso le agrega los efectos generados por un *multipath fading channel* estos aumentan imposibilitando que se genere un enlace de comunicación confiable [1]. Cuando se tienen dificultades para transmitir una palabra o un mensaje, lo que naturalmente se tiende a hacer es retransmitirlo, pero esto suele no ser eficiente y en muchas ocasiones se convierte en algo poco seguro. Por ejemplo; durante la Segunda Guerra Mundial uno de los grandes problemas con los cuales se tuvo que lidiar fue como contactar a espías Estadounidenses mientras se encontraban en territorio alemán. Los espías no podían pedir una retransmisión del mensaje en caso que no les llegara correctamente, ya que siempre existía la posibilidad de que los enemigos interceptaran la señal. Por otro lado, si es que el mensaje era muy corto este podría ser totalmente destruido por el *jamming* producido por los alemanes. Para poder disminuir la cantidad de errores y así lograr las deseadas comunicaciones confiables se notó que era necesario transmitir secuencias que sean lo más diferentes unas de las otras de manera que el ruido del canal no cambie a una secuencia de bits en otra ya existente. Esto significa que es necesario introducir cierta *redundancia* para incrementar así la confiabilidad de la comunicación. Esta introducción de redundancia implica que se estarán enviando bits extra que no contienen información o un mensaje en si, por ende se genera una disminución en la tasa de transmisión [2], pero por otro lado se logra una comunicación confiable y segura. Naturalmente la pregunta que surgió en esa época fue: “¿cuánta redundancia debe ser agregada al mensaje de manera que pueda enviarse seguramente?”

El proceso previamente descrito se conoce con el nombre de codificación de canal y se basa en los principios establecidos por Shannon en 1948 [3], quien fue capaz de formalizar el problema definiendo una medida matemática de la información y un grupo de teoremas bastante complejos que responden a muchas interrogantes. Su respuesta a la pregunta previamente establecida fue: “es necesario agregar “suficiente” redundancia para estar estadísticamente seguros de la efectividad de protección.” La cota inferior de esta cantidad está determinada realmente por la capacidad de canal  $C$ . La prueba empírica de este problema fue rápidamente demostrada por Hamming, Golay y otros [4] y hasta el día de hoy se usan sus principios para generar enlaces seguros en ambientes hostiles como lo puede ser una ciudad urbanizada, comunicaciones con aviones civiles o militares, barcos, satélites, etc.

En el presente Informe se espera probar los efectos generados al introducir codificación previo a la transmisión por un canal Rayleigh. En primera instancia se entrega un Marco Teórico donde se explican las principales técnicas de codificación existentes para canales AWGN como también para *fading channels*. Posteriormente se explicita el problema de simulación, se entregan los resultados, para finalmente concluir con un análisis de estos últimos y generar extrapolaciones hechas sobre el trabajo realizado.

*Nota: en los anexos de este informe se entregan los códigos utilizados en la herramienta computacional MATLAB, para simular la codificación de canal en un “multipath Rayleigh fading channel”.*

## 2. Marco Teórico

La codificación de canal o códigos de canal pueden ser clasificados en dos grandes grupos; *códigos de bloque* y *códigos convolucionales*. En los **códigos de bloque** se tienen  $M = 2^k$  mensajes, donde cada cual representando una secuencia binaria de largo  $k$  llamada *secuencia de información*. Una de estas secuencias es mapeada en una tira binaria de largo  $n$  más conocida como *palabra de código*, donde  $n \geq k$ . La palabra de código es usualmente transmitida sobre un canal de comunicación mediante algún tipo de modulación digital como BPSK, QPSK, QAM, 16QAM, etc. Los bloques de código no tienen memoria. Luego de que se codifica una palabra de código y es enviada, el sistema recibe un nuevo grupo de  $k$  bits de información y los decodifica utilizando el mapeo definido por el esquema de codificación utilizado. La palabra de código resultante contiene solamente de los  $k$  bits de información y es independiente de todas las palabras de código transmitidas anteriormente.

Por otro lado, los **códigos convolucionales** están definidos en base a máquinas de estados finitos. En estos códigos, en cada instante de tiempo  $i$ ,  $k$  bits de información entran en el codificador, causando  $n$  símbolos binarios generados en la salida del codificador y cambiando el estado del codificador desde  $\sigma_{i-1}$  a  $\sigma_i$ . El grupo de posibles estados es finito y se denota usualmente por  $\Sigma$ . Los  $n$  símbolos binarios generados en la salida del codificador y el siguiente estado  $\sigma_i$  depende de los  $k$  bits que entran como también de  $\sigma_{i-1}$ . Se puede representar un código convolucional por un registrador de desplazamientos de largo  $Kk$  como se observa en la Figura 1.

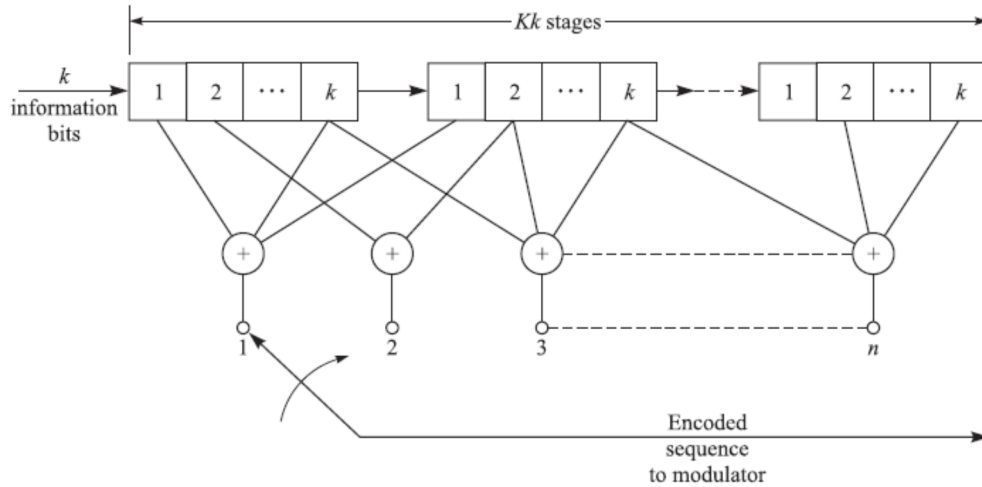


Figura 1: Codificador Convolucional.

En cada instante de tiempo,  $k$  bits entran en el codificador y los contenidos del registrador de desplazamiento son desfasados a la derecha por  $k$  elementos de memoria. Los contenidos de los  $k$  elementos más a la derecha del registrador de desplazamientos dejan el codificador. Después de que los  $k$  bits han entrado en el registrador de desplazamiento, los  $n$  sumadores agregan el contenido de los elementos de la memoria y son conectados (adición módulo-2), así generan el código de secuencia de largo  $n$  el cual es enviado hacia el modulador. El estado de este código convolucional viene dado por el contenido de los primeros  $(K - 1)k$  elementos del registrador de desplazamiento.

La *tasa de código* de un código de bloque o convolucional es denotado por  $R_c$  y viene dado por

$$R_c = \frac{k}{n} \quad (1)$$

La tasa de código representa el número de bits de información que se envían por el canal de comunicación. Generalmente los códigos suelen abreviarse utilizando la notación  $(n, k)$  para indicar que se están enviando  $n$  bits en total, de los cuales solo  $k$  corresponden a información real. La unidad de medida de  $R_c$  es bits de información por transmisión. Dado que generalmente  $n \geq k$ , se tiene que  $R_c \leq 1$ .

Asumamos que la palabra de código de largo  $n$  es transmitida utilizando una constelación  $N$ -dimensional de tamaño  $M$ , donde  $M$  se asume que es una potencia de 2 y  $L = \frac{n}{\log_2 M}$  es un entero que representa el número de símbolos  $M$ -arios transmitidos por palabra de código. Si es que la duración del símbolo es  $T_s$ , entonces el tiempo de transición para  $k$  bits es  $T = LT_s$  y la tasa de transmisión viene dada por

$$R = \frac{k}{LT_s} = \frac{k}{n} \times \frac{\log_2 M}{T_s} = R_c \frac{\log_2 M}{T_s} \quad \text{bits/segundo.} \quad (2)$$

La dimension del espacio de la señal codificada y modulada es  $LN$ , y utilizando el teorema de dimensionalidad<sup>1</sup> se concluye que el mínimo ancho de banda para la transmisión es

$$W = \frac{N}{2T_s} = \frac{RN}{2R_c \log_2 M} \quad \text{bits/segundo} \quad (3)$$

de la cual se desprende que la tasa espectral de bits es

$$r = \frac{R}{W} = \frac{2 \log_2 M}{N} R_c. \quad (4)$$

Esta última ecuación idica que si uno compara un sistema codificado con uno no codificado que usa el mismo esquema de modulación, la tasa de bit cambia por un factor  $R_c$  y el ancho de banda es cambiado por un factor  $1/R_c$ , es decir, existe una disminución en la tasa pero un incremento en el ancho de banda.

## 2.1. Codificación e *Interleaving* para *Fading Channels*

Los códigos de bloque y convolucionales están diseñados para mostrar buen rendimiento en canales AWGN. Para *fading channels*, los errores asociados con el demodulador tienden a ocurrir en “ráfagas”, correspondientes a los tiempos cuando el canal presenta un desvanecimiento profundo. La mayor parte de los códigos diseñados para canales AWGN pueden mostrar un peor rendimiento en *fading channels*, incluso peores que el caso en que el sistema no ocupa codificación de canal.

Para mejorar el rendimiento de la codificación en *fading channels*, la codificación típicamente se combina con *interleaving* para mitigar el efecto producido por las ráfagas de errores. La premisa básica de la codificación más *interleaving* es separar las ráfagas de errores debido a los profundos desvanecimientos en muchas palabras de código, de manera que cada palabra de código reciba solamente a lo más una pequeña cantidad de errores de símbolos simultáneos. Esta separación de los errores es lograda por un *interleaver* y la corrección de errores por su parte se hace mediante el código utilizado. El tamaño del *interleaver* debe ser lo suficientemente grande como para que el desvanecimiento sea independiente a lo largo de la palabra de código recibida. Los *fading channels* lentos requieren de *interleavers* de mayor tamaño, los cuales pueden introducir grandes retardos en las señales recibidas [5].

<sup>1</sup>El teorema de dimensionalidad puede encontrarse en la página 227 de [6].

La codificación más *interleaving* es una forma de *diversity*, la cual suele ser función de la distancia mínima de Hamming del código. Por lo tanto, el diseño de la codificación más *interleaving* en *fading channels* debe enfocarse en maximizar la distancia de Hamming del código, en vez de la distancia Eucladiana utilizada para incrementar el desempeño en canales AWGN<sup>2</sup>.

### 2.1.1. Codificación de bloque con *interleaving*

Un bloque *interleaver* es un arreglo de  $d$  filas y  $n$  columnas como se observa en la Figura 2. Las palabras de código que ingresan en el *interleaver* son ordenadas por filas (cada fila contiene una palabra de código o conjunto de símbolos de palabras de código). Luego, se envían las respectivas columnas al modulador y posteriormente se transmite la información, de manera que durante la transmisión los símbolos de una misma palabra de código estén separados lo más posible; para este caso la distancia de separación es equivalente a  $d - 1$ . A continuación, la señal atraviesa el canal y es demodulada, para pasar posteriormente por el *de-interleaver* en el cual se ordenan los símbolos recibidos por columna y finalmente la salida de este bloque lee la información por filas, recuperando así el mensaje original. Todo este proceso anteriormente descrito puede observarse en la Figura 2.

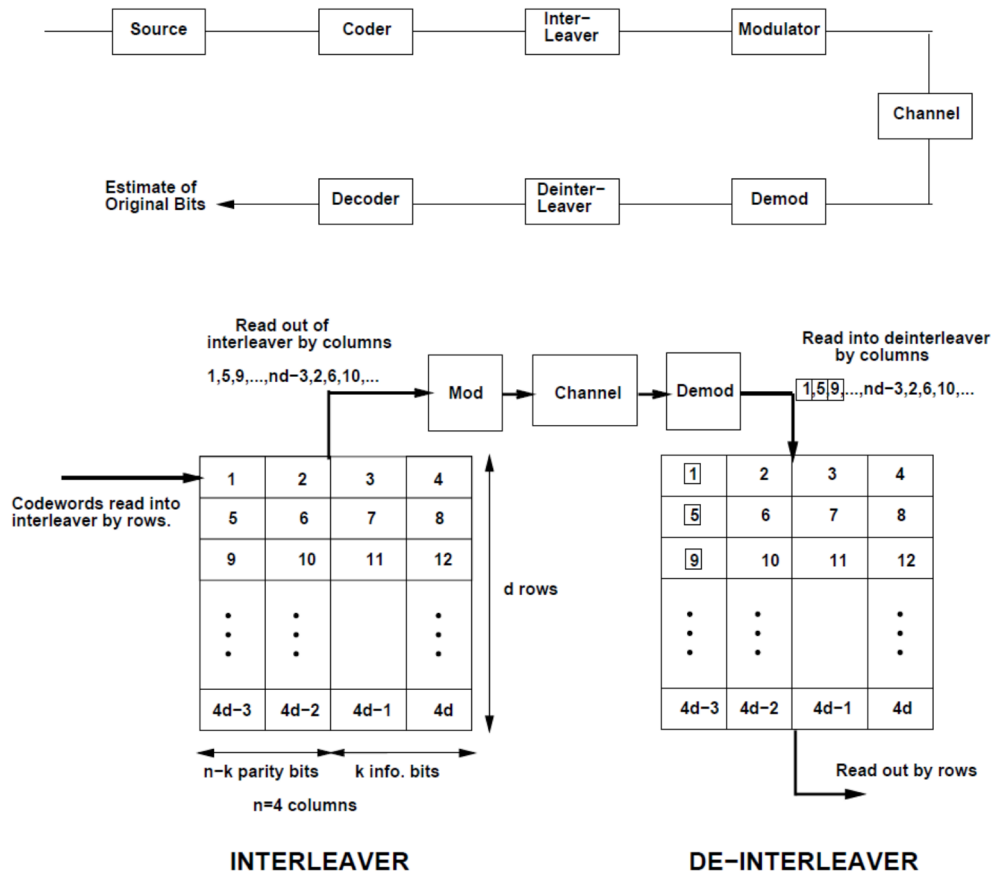


Figura 2: La operación *Interleaver/De-interleaver*.

La Figura 2 ilustra la habilidad que tiene codificar y utilizar *interleaving* para corregir una ráfaga de errores. Suponga ahora que nuestro esquema de transmisión utiliza un código de bloque binario

<sup>2</sup>La distancia Eucladiana tiene relación con aumentar la distancia entre símbolos al modular digitalmente en presencia de un canal AWGN.

$(n, k)$  capaz de corregir  $t = 2$  errores. Si es que la palabra de código es transmitida por un canal que genera una ráfaga de errores de tres símbolos, entonces tres de cuatro de los símbolos de las palabras de código serán recibidos con error. Dado que el código puede corregir hasta  $t = 2$  errores, entonces la palabra de código será decodificada erróneamente. Pero, si es que la palabra de código ingresa a un *interleaver* previo a la modulación, entonces la ráfaga de tres errores de la palabra de código será esparcido por sobre tres otras palabras de código. Ahora, dado que un solo símbolo erróneo puede ser fácilmente corregido por el decodificador, la información original puede ser decodificada correctamente [7].

### 2.1.2. Codificación convolucional con *interleaving*

Un ejemplo de como podría lucir un diagrama de bloque de un *interleaver* convolucional se puede observar en la Figura 3. La salida del codificador es multiplexada en *buffers* de tamaño creciente. La entrada al canal es similarmente multiplexada. La operación opuesta se hace en el decodificador, ‘por lo tanto, un *interleaver* convolucional retrasa la transmisión por el canal, dado que el tamaño de la salida del codificador aumenta a lo largo del tiempo, pero este retraso es revertido en el receptor. Este *interleaver* toma salidas secuenciales del codificador y las separa por  $N - 1$  otros símbolos, por lo tanto de esta forma rompe las ráfagas de error que introduce el canal. La memoria total asociada a un *interleaver* convolucional es  $0,5N(N - 1)$  y el retardo es  $N(N - 1)T_s$ .

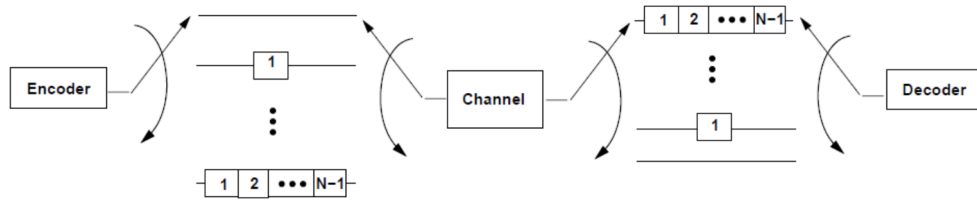
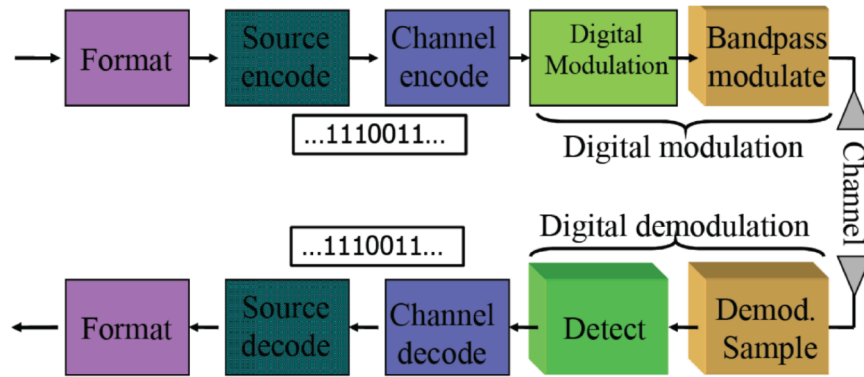


Figura 3: Codificación convolucional más *interleaving*.

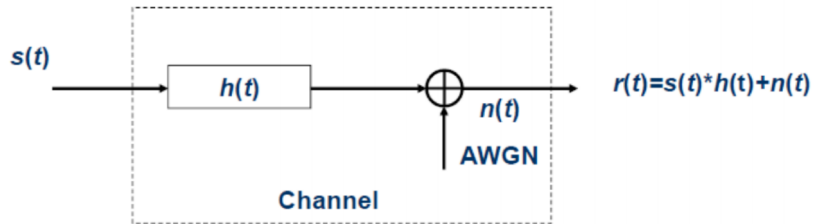
### 3. Parámetros y Métodos de Simulación

Las simulaciones que presenta este Informe se centran en la implementación de los sistemas descritos en la Figura 4 (a) y (b), donde  $h(t)$  es considerado como un canal Rayleigh y las modulaciones digitales empleadas son **BPSK** y **16-QAM**, cumpliendo con los siguientes parámetros de diseño:

1. Simulación de Monte Carlo.
2. 100,000 bits generados de forma aleatoria.
3. Energía de cada símbolo normalizada  $E_s = 1$ .
4. Se evalúa el sistema para diferentes valores de  $SNR_{dB}$  (entre 0 y 30 dB).
5. Modulación digital en banda base.
6. Ruido aditivo AWGN con  $\mu = 0$  y  $\sigma^2 = N_0/2$ .
7. Código Hamming(7,4) empleando soft decoding.
8. Código Convolutivo con las siguientes características: *Convolutional code of Rate 1/2, constraint length  $K = 3$ , Generator polynomial  $[7, 5]_{octal}$ , and soft decoding.*



(a) Sistema de comunicaciones digital estándar.



(b) Modelo del canal.

Figura 4: Sistema de comunicaciones digital y caracterización de un canal de transmisión.



### 3.1. Pasos código de Hamming

1. Generar un arreglo de bits aleatorios.
2. Agrupar los bits del paso anterior en grupos de 4, agregando tres bits de paridad para finalmente convertirlo en un grupo de 7 bits utilizando el código Hamming(7,4).
3. Agregar el efecto de un canal Rayleigh, más ruido AWGN.
4. Detectar y decodificar utilizando *soft decision*.
5. Contar el número de errores.
6. Repetir lo mismo para múltiples  $E_b/N_0$ .

### 3.2. Pasos código Convolutacional

1. Generar un arreglo de bits aleatorios.
2. Codificarlos convolutacionalmente utilizando una tara de  $1/2$ , y un polinomio generador  $[7, 5]_{octal}$ .
3. Agregar el efecto de un canal Rayleigh, más ruido AWGN.
4. Detectar y decodificar utilizando *soft decision*.
5. Contar el número de errores.
6. Repetir lo mismo para múltiples  $E_b/N_0$ .

*Nota: la simulación del “Rayleigh multipath fading channel” ya está hecha en [1], por lo que se utiliza en el presente trabajo. El script para generar el Código de Hamming(7,4) tiene su base en ideas obtenidas de [9]. Por otro lado, para realizar el Código Convolutacional se utiliza [10] y [11].*

## 4. Resultados de Simulaciones

### 4.1. Código de Hamming(7,4)

#### 4.1.1. Modulación BPSK

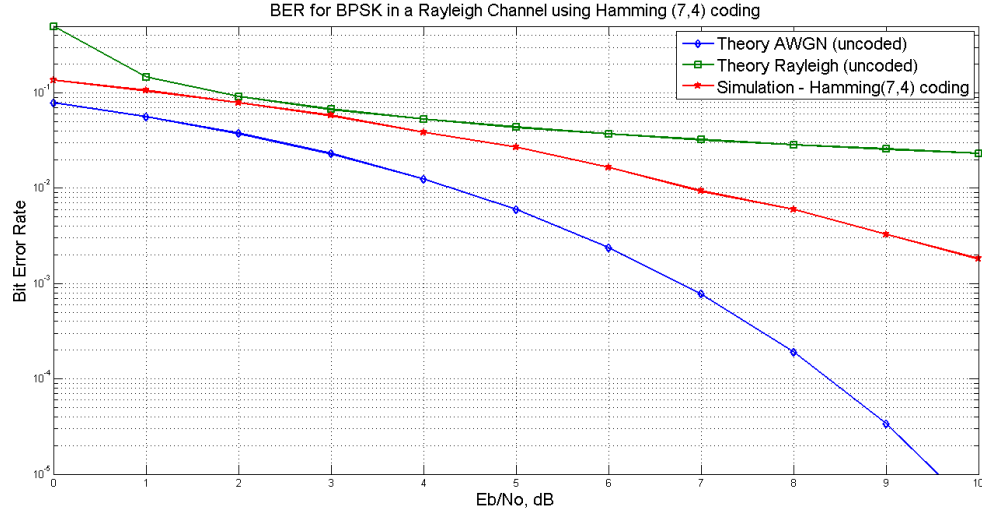


Figura 5: Curva del BER utilizando modulación BPSK en un canal Rayleigh con código de Hamming(7,4).

#### 4.1.2. Modulación 16-QAM

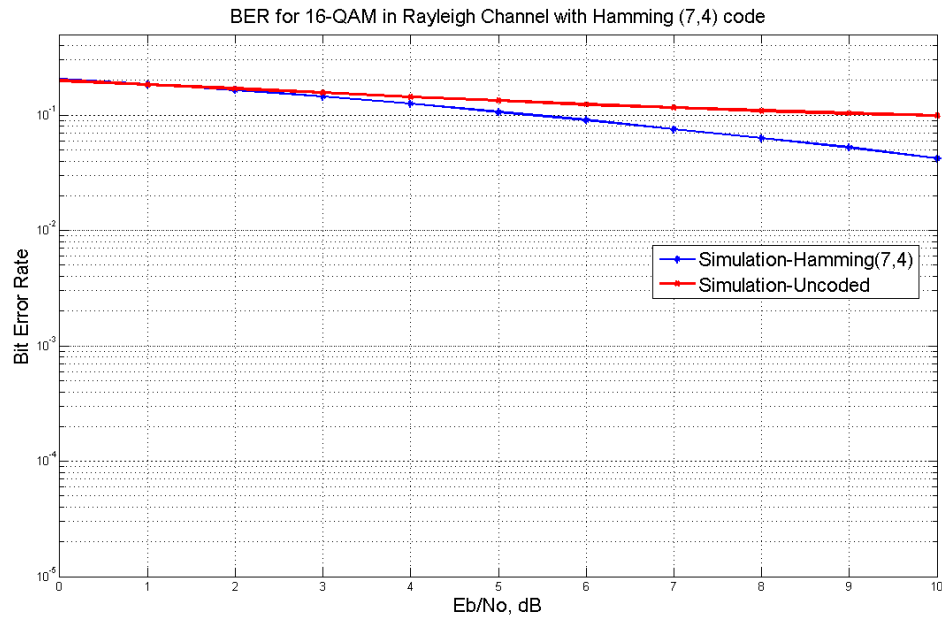


Figura 6: Curva del BER utilizando modulación 16-QAM en un canal Rayleigh con código de Hamming(7,4).

## 4.2. Código Convolutacional

### 4.2.1. Modulación BPSK

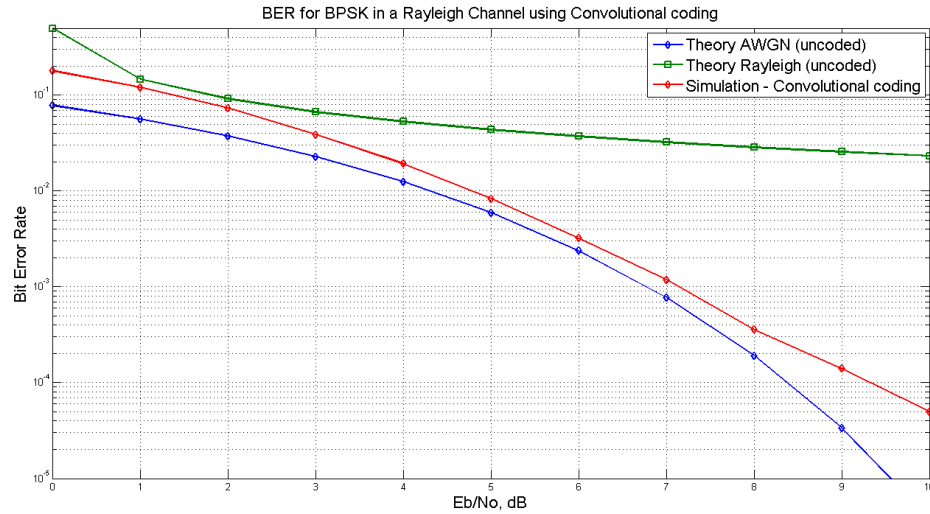


Figura 7: Curva del BER utilizando modulación BPSK en un canal Rayleigh con código Convolutacional.

### 4.2.2. Modulación 16-QAM

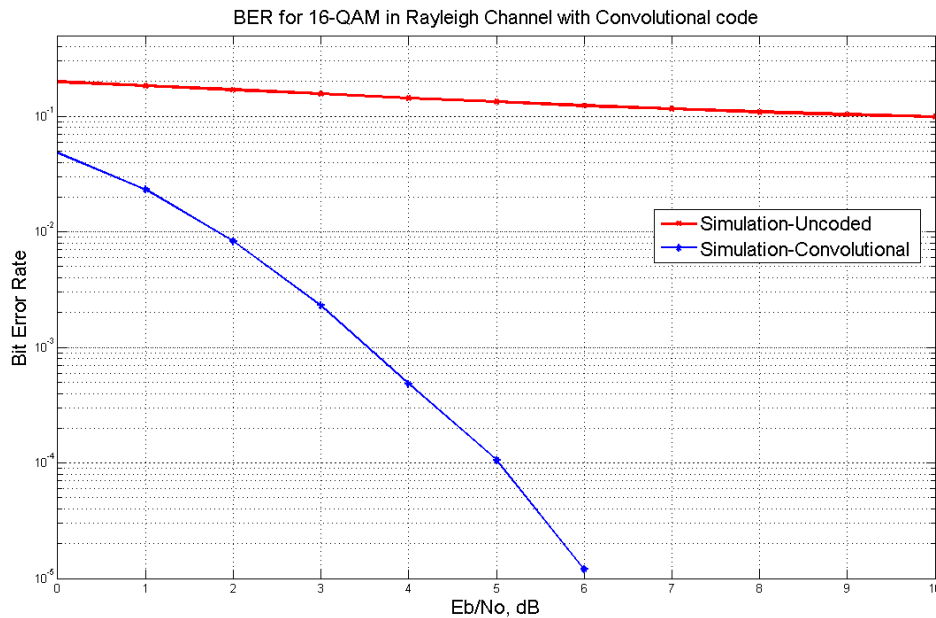


Figura 8: Curva del BER utilizando modulación 16-QAM en un canal Rayleigh con código Convolutacional.

### 4.2.3. Comparaciones entre código de Hamming(7,4) y Convolutacional:

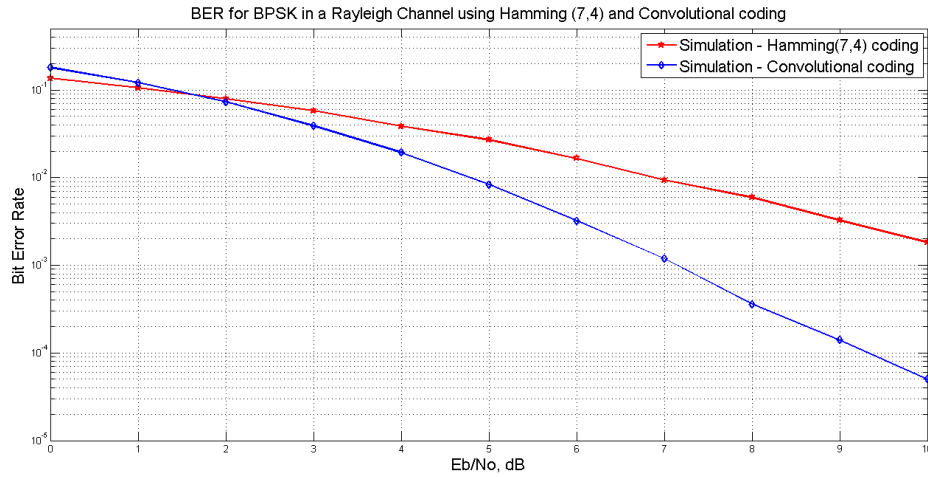


Figura 9: Comparación de la curva del BER utilizando modulación BPSK en un canal Rayleigh con código Hamming(7,4) y Convolutacional.

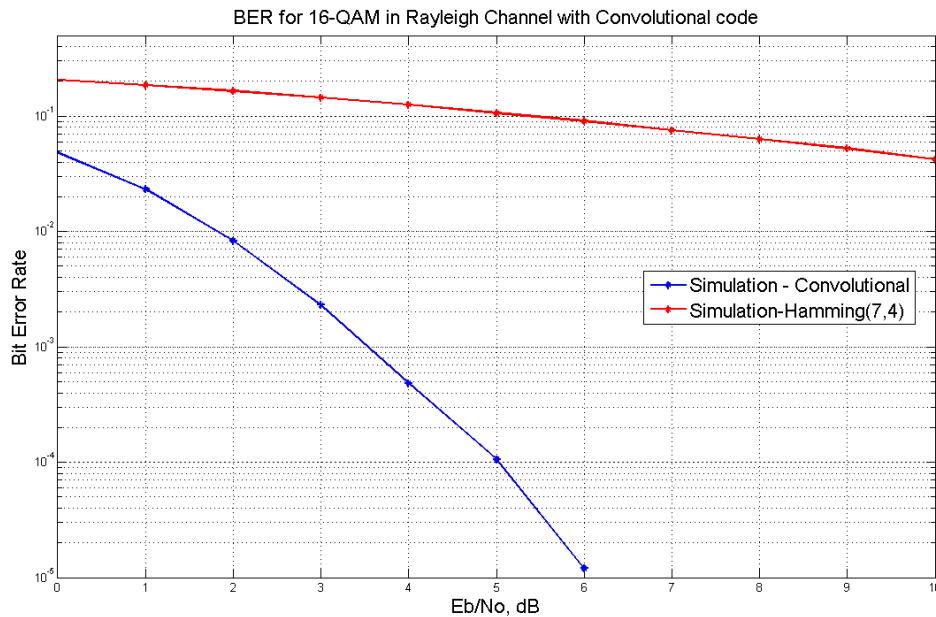


Figura 10: Comparación de la curva del BER utilizando modulación 16-QAM en un canal Rayleigh con código Hamming(7,4) y Convolutacional.

## 5. Análisis de los Resultados

En base a lo que se exhibe en todas las gráficas, es posible argumentar que el efecto de la codificación de canal, independiente el método utilizado (código de Hamming(7,4) o Convolutacional), genera un gran impacto en la respuesta del sistema, ya que la curva del BER mejora considerablemente en relación al caso en donde no se codifica previo a la transmisión. Por otro lado, se puede notar que existen casos, como el observado en la Figura 7, donde la codificación puede hacer que la curva del BER de un sistema de comunicación en un canal Rayleigh sea casi idéntica a la de un simple canal AWGN. Lo anterior demuestra que una “buena” codificación de canal puede minimizar de manera crítica los efectos producidos por un medio que distorsiona la señal.

Al mismo tiempo, viendo las Figuras 9 y 10, se puede testificar que en términos de confiabilidad el código convolutacional es mejor que el código de Hamming(7,4), ya que el primero muestra un respuesta del BER notoriamente superior al segundo. Pero, como dice el refrán *“no todo en esta vida es gratis”*, debido a que el código Convolutacional es claramente más complejo que el de Hamming(7,4), hecho que hace que al momento de generar la simulación computacional la entrega de resultados sea lenta, haciendo que sea un código poco eficiente.

## 6. Conclusiones

La codificación de canal tiene un alto impacto en un sistema de comunicación, ya que permite que la cantidad de errores de transmisión disminuya considerablemente, porque los detecta y en muchos casos los corrige. Este hecho, hace que sea posible recibir correctamente un mensaje, independiente que el medio físico por el cual se propaga la onda sea hostil, por ende aumenta la confiabilidad total del sistema. Lo anterior a favorecido a que el número de aplicaciones que utiliza codificación de canal se extienda desde los usos más cotidianos como lo son las redes de telefonía celular, hasta comunicaciones extremadamente complejas como lo son las satelitales o militares, pudiendo generar redes de gran alcance que lidian con muchos de los problemas que implica establecer enlaces a distancia o inalámbricos. Al mismo tiempo, su uso variado ha significado que se creen diversos códigos que permitan cumplir con los estándares de calidad de cada aplicación, factor que evidentemente favorece a la comunicaciones actuales, como a las futuras.

Por otro lado, los resultados obtenidos demuestran que el código a utilizar es un factor de suma importancia al momento de diseñar un sistema de comunicación, ya que dependiendo de la aplicación que se le esté dando, podría causar problemas serios al momento de generar un enlace. Lo anterior se explica, ya que existen ocasiones en las cuales no es posible *“darse el lujo”* de corregir tantos errores mediante códigos extremadamente confiables, porque estos son generalmente poco eficientes y demoran en entregarle el mensaje al receptor, afectando considerablemente a la tasa de transmisión de datos.

Como ejemplo, supóngase que se está diseñando una red de telefonía celular digital; evidentemente esta debe utilizar algún método de codificación de canal para brindarle al usuario la calidad necesaria que le permita establecer llamadas en un canal que distorsiona considerablemente el mensaje. En este caso es importante llegar a un equilibrio, ya que si un usuario realiza una llamada y el retardo existente entre lo que dice, y lo que escucha el destinatario es grande, sería casi imposible generar una comunicación fluida. Por lo tanto, la elección del código a utilizar, debe asegurar en cada instante la calidad mínima (entregar correctamente el mensaje), junto con un retardo máximo, entre la emisión y entrega del mensaje, que no pueda ser captado por los agentes del sistema.

## 7. Anexos: Códigos

### 7.1. Simulación para BPSK y código Hamming(7,4)

```

1  % Script for computing BER with Hamming (7,4) code
2  % with soft and hard decision decoding
3
4  clear all;
5  close all;
6  clc;
7  N = 10^5 ; % number of bits
8
9  min_Eb_N0=0;
10 max_Eb_N0=30;
11 Eb_N0_dB = [min_Eb_N0:1:max_Eb_N0]; % multiple Eb/N0 values
12 Ec_N0_dB = Eb_N0_dB - 10*log10(7/4);
13
14 h = [ 1 0 1 ; 1 1 1; 1 1 0; 0 1 1];
15 ht = [h ; eye(3)];
16 g = [eye(4) h];
17 synRef = [ 5 7 6 3 ];
18 bitIdx = [ 7 7 4 7 1 3 2].';
19
20 c_vec = zeros(2^4,7);
21 for (kk = 0:2^4-1)
22     m_vec = base2dec(dec2bin(kk,4).',2).';
23     c_vec(kk+1,:) = mod(m_vec*g,2);
24 end
25
26 for yy = 1:length(Eb_N0_dB)
27
28     % Transmitter
29     ip = rand(1,N)>0.5; % generating 0,1 with equal probability
30
31     % Hamming coding (7,4)
32     ipM = reshape(ip,4,N/4).';
33     ipC = mod(ipM*g,2);
34     cip = reshape(ipC.',1,N/4*7);
35
36     % Modulation
37     s = 2*cip-1; % BPSK modulation 0 -> -1; 1 -> 0
38
39     [iout, qout]=rayleigh(real(s),imag(s),length(cip),0.5,1,5,1);
40
41     % Channel - AWGN
42     n = 1/sqrt(2)*[randn(size(cip)) + j*randn(size(cip))]; % white
43     % gaussian noise, 0dB variance
44
45     % Noise addition

```