

# Práctica: Simulación de un sistema de acceso ADSL

Tecnologías de Acceso a Red  
Grado en Ing. de Sistemas de Comunicaciones  
Universidad Carlos III de Madrid  
Curso 2012-2013

Luca Martino (luca@tsc.uc3m.es)  
Francisco Jesús Rodríguez Ruiz (franrruiz@tsc.uc3m.es)  
José Emilio Vila Forcén (jemilio@tsc.uc3m.es)

## 1. Introducción

Hoy en día, gracias al desarrollo de la tecnología de Línea de Abonado Digital (DSL), es posible disfrutar de acceso a Internet de alta velocidad empleando para ello los tradicionales cables de cobre y la infraestructura de la red telefónica, originalmente diseñada para el transporte de señales analógicas de voz.

ADSL (*Asymmetric Digital Subscriber Line*) es una tecnología surgida a mediados de los años noventa que permite transmitir hasta 6 Mbps de bajada y 640 Kbps de subida, y cuyo nombre deriva de esta propiedad asimétrica respecto a la velocidad de transmisión en ambas direcciones.

La coexistencia de servicios tradicionales de telefonía y de transmisión de datos a alta velocidad sobre el mismo medio de transmisión se debe a que ADSL utiliza las frecuencias ubicadas por encima de la banda asignada para la telefonía. La asignación de las frecuencias para ambos servicios se realiza con una técnica de división en frecuencia conocida como FDM (*Frequency Division Multiplexing*), la cual divide el ancho de banda del par trenzado (de 1,104 MHz) en tres bandas diferentes:

- Banda para telefonía tradicional (0 – 4,3125 kHz).
- Banda para transmisión de datos en dirección de subida (25,768 kHz – 138 kHz).
- Banda para transmisión de datos en dirección de bajada (138 kHz – 1,104 MHz).

## 2. Objetivos

Con esta práctica se pretende llevar a cabo una simulación de las funcionalidades de un transceptor (o módem) ADSL simplificado. En particular, la práctica se centrará únicamente

en las capas física y de enlace de los bloques transmisor y receptor para una comunicación en sentido descendente, es decir, en la banda de 138 kHz a 1,104 MHz.

La simulación se llevará a cabo en lenguaje MatLab.

### 3. La capa física

La implementación del sistema ADSL objeto de esta práctica debe tener la capacidad de simular los bloques de la capa física que se muestran en la figura 1. En esta sección se describirá brevemente cada uno de ellos y se proporcionarán algunos detalles prácticos sobre su implementación.

Nótese que el diagrama de bloques presentado en dicha figura corresponde, salvo por ligeras modificaciones, al de una modulación OFDM (*orthogonal frequency-division multiplexing*) estándar, con  $N = 256$  subportadoras, aunque en xDSL se suele emplear el término DMT (*discrete multi-tone modulation*). En este tipo de modulación, el ancho de banda disponible (que en este caso es de 1,104 MHz) se divide en 256 subcanales, de 4,3125 kHz cada uno. De estos 256 subcanales:

- El primer tono o subcanal se usa para servicios de telefonía de voz tradicional (POTS, *plain old telephone service*).
- Los tonos del 2 al 6 (de 4,3125 kHz a 25,875 kHz) se utilizan para evitar la interferencia entre POTS y ADSL.
- Los tonos del 7 al 32 (25,768 kHz a 138 kHz) se utilizan para la dirección de subida.
- Los tonos del 33 al 256 (138 kHz a 1,104 MHz) son los utilizados para la dirección de bajada y son, por tanto, los únicos de los que se podrá hacer uso en esta práctica.

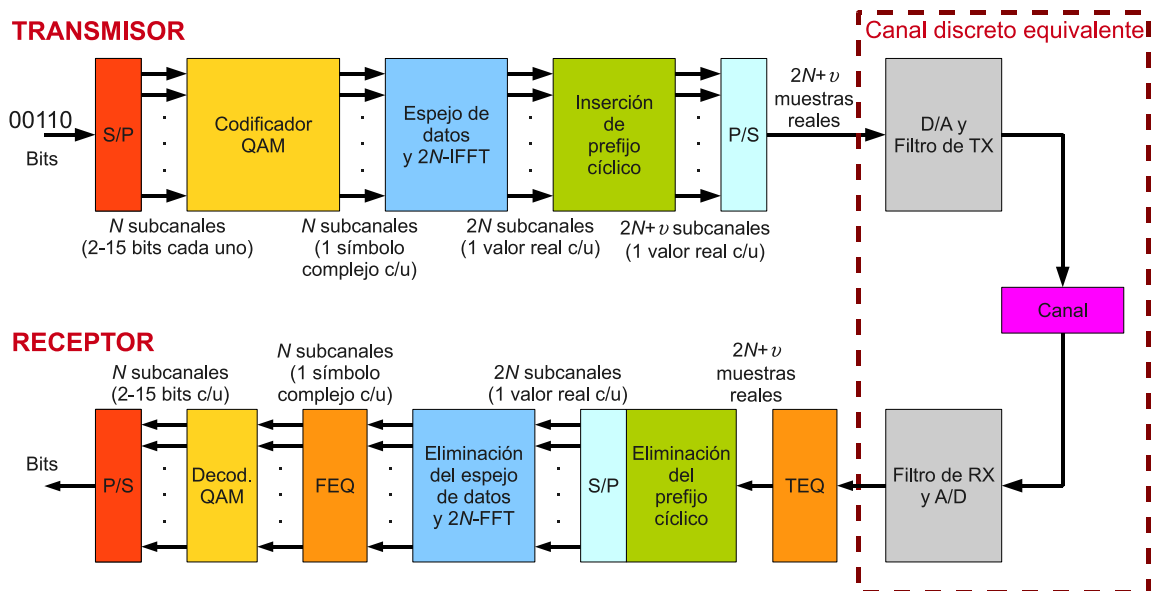


Figura 1: Diagrama de bloques físico de un transmisor/receptor ADSL

### 3.1. Transmisor

#### 3.1.1. Conversor Serie/Paralelo (S/P)

El conversor serie/paralelo recibe un único flujo de datos binarios, y debe encargarse de distribuirlos entre los  $N = 256$  subcanales.

Una de las características más importantes de los sistemas DMT es la posibilidad de realizar el llamado *bit loading*, que consiste en distribuir el número de bits por subcanal en función de las propiedades del canal de transmisión, de manera que no todos los subcanales deban tener el mismo número de bits. Aunque en esta práctica no se persigue implementar ningún algoritmo de *bit loading*, **sí se debe permitir la posibilidad de que cada subcanal tenga un número de bits distinto.**

Se recomienda para ello emplear un vector de longitud  $N$ , llamado  $\mathbf{bn}$ , que indique el número de bits que deben “viajar” por cada subcanal. Cada elemento del vector  $\mathbf{bn}$  puede contener un entero comprendido entre 2 y 15, tal y como indica el estándar ADSL. No obstante, también puede ocurrir que no se haga uso de un subcanal; en tal caso, el elemento correspondiente del vector  $\mathbf{bn}$  se fijaría a cero. A fin de mostrar un ejemplo que clarifique lo aquí expuesto, en la figura 2 se muestra el funcionamiento del conversor S/P cuando  $\mathbf{bn}(1)=2$ ,  $\mathbf{bn}(2)=3$ ,  $\mathbf{bn}(3)=4$ ,  $\dots$ ,  $\mathbf{bn}(256)=2$ .

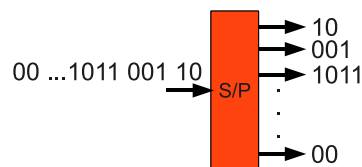


Figura 2: Ejemplo de funcionamiento del conversor S/P

Dado que sólo se van a emplear los subcanales del 33 al 256, debe establecerse  $\mathbf{bn}(1:32)=0$ . En una fase inicial de la práctica, puede establecerse  $\mathbf{bn}(33:256)=2$  (bits por subcanal), aunque a lo largo de la misma se pedirá realizar una modificación de estos valores.

#### 3.1.2. Codificador QAM

Los bits que llegan a cada subcanal deben codificarse según un esquema QAM (*Quadrature Amplitude Modulation*), con codificación Gray para disminuir la probabilidad de error de bit. De esta manera, habrá tantos codificadores QAM como subcanales, es decir, 256, siendo la salida de cada uno de ellos un símbolo complejo. Debe prestarse especial atención a que **la energía media de cada constelación sea igual a la unidad.**

A pesar de que ADSL permite constelaciones de hasta 15 bits ( $2^{15} = 32768$  símbolos), para los objetivos de esta práctica bastará con permitir constelaciones de tan sólo 2, 3 y 4 bits, como las que se muestran en la figura 3.

#### 3.1.3. Espejo de datos y $2N$ -IDFT

El siguiente bloque de la cadena de transmisión es el espejo de datos, cuya principal tarea es obtener el complejo conjugado de las  $N$  muestras del banco de codificadores QAM para formar un bloque de  $2N$  muestras. La función que proporciona los valores de salida del espejo de datos, en función de los valores de entrada, viene descrita por:

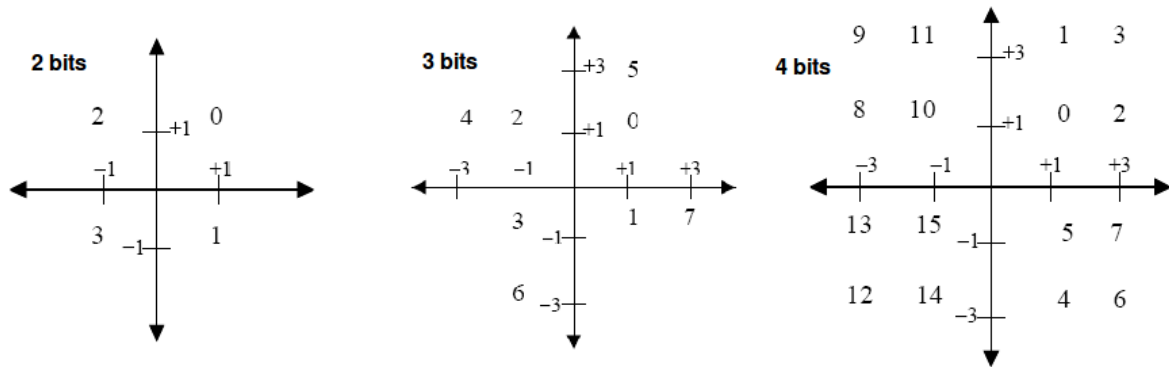


Figura 3: Ejemplo de constelaciones QAM con codificación Gray para 2, 3 y 4 bits

$$X_i^{out} = \begin{cases} \operatorname{Re}(X_0) & i = 0 \\ X_i & i = 1, \dots, N - 1 \\ \operatorname{Im}(X_0) & i = N \\ X_{2N-i}^* & i = N + 1, \dots, 2N - 1 \end{cases} \quad (1)$$

La salida de este bloque, de  $2N$  valores complejos, constituye lo que se conoce como símbolo DMT.

A continuación, se debe aplicar la **transformación IDFT sobre el bloque de  $2N$  valores**, dando lugar a lo que se conoce como un símbolo DMT en el dominio del tiempo. Nótese que, debido a la transformación llevada a cabo por el espejo de datos, los valores resultantes toman valores reales.

### 3.1.4. Inserción del prefijo cíclico

La inserción del prefijo cíclico es una operación sencilla, que aparece representada esquemáticamente en la figura 4, y cuyo objetivo es eliminar la interferencia (ISI) entre símbolos DMT adyacentes. Consiste básicamente en copiar  $\nu$  muestras de cada símbolo DMT al comienzo del mismo, formando bloques de  $2N + \nu$  muestras. Si se elige un valor de  $\nu$  suficientemente elevado, se elimina la ISI que pueda introducir el canal. En particular, no existe ISI si la longitud de la respuesta al impulso del canal es menor de  $\nu + 1$ , pues en tal caso el canal “ve” una secuencia periódica y la convolución circular coincide con la lineal, por lo que la DFT de la salida coincide con el producto de las DFTs del símbolo DMT de entrada y del canal.

En esta práctica, tomaremos en todo momento un valor de  $\nu = 32$ .

### 3.2. Canal discreto equivalente

En esta práctica, modelaremos directamente la respuesta impulsiva del canal discreto equivalente como un tren de deltas, esto es:

$$h[n] = \sum_{\ell=0}^L \alpha_{\ell} \delta[n - \ell] \quad (2)$$

El canal que emplearemos a lo largo de esta práctica viene dado por:

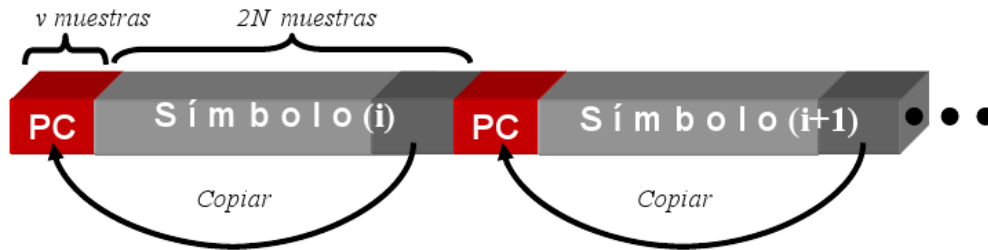


Figura 4: Inserción del prefijo cíclico

$$h[n] = \alpha_0\delta[n] + \alpha_3\delta[n - 3] + \alpha_6\delta[n - 6] + \alpha_{20}\delta[n - 20], \quad (3)$$

donde  $\alpha_0 = 1$ , y el resto de coeficientes se eligen para que estén 2, 8 y 3 dB por debajo de  $\alpha_0$ , respectivamente, esto es, con valores  $\alpha_3 = 0,6310$ ,  $\alpha_6 = -0,1585$  y  $\alpha_{20} = 0,5012$ . Nótese que para el caso de  $\alpha_6$  se ha elegido signo negativo.

El canal, además, **debe introducir ruido aditivo blanco y gaussiano** sobre cada muestra, de media nula y varianza  $\sigma_n^2$  en función de la SNR deseada:

$$\sigma_n^2 = \frac{1}{\text{SNR}} \quad (4)$$

Es importante notar que, dado que el canal podría introducir interferencia intersimbólica (ISI), no se debe transmitir cada símbolo DMT de manera aislada, sino que ha de esperarse a disponer de toda la secuencia completa antes de poder ser enviada.

### 3.3. Receptor

#### 3.3.1. Ecualizador en el dominio del tiempo (TEQ)

El ecualizador en el dominio del tiempo (TEQ, *Time-Domain Equalizer*) es un filtro FIR que se emplea para acortar la respuesta impulsiva equivalente del sistema formado por el canal y el propio TEQ. Esto resulta útil en el caso de que la longitud del prefijo cíclico,  $\nu$ , sea inferior a la longitud de la respuesta impulsiva del canal.

No obstante, en esta práctica **no será necesario implementar el bloque de ecualización en el dominio del tiempo**, por lo que puede ser ignorado.

#### 3.3.2. Eliminación del prefijo cíclico

Este bloque tiene como tarea convertir cada trama física de  $2N + \nu$  muestras en una trama de sólo  $2N$  muestras, eliminando las correspondientes al prefijo cíclico, es decir, las que fueron introducidas por el transmisor siguiendo el esquema de la figura 4.

#### 3.3.3. Eliminación del espejo de datos y $2N$ -DFT

En primer lugar, debe realizarse una **transformación DFT de  $2N$  puntos** sobre las muestras recibidas, construyendo así en el receptor un símbolo DMT en el dominio de la frecuencia.

A continuación, la eliminación del espejo de datos consiste en suprimir los datos redundantes que fueron introducidos por el transmisor según la expresión (1). Es decir, este bloque

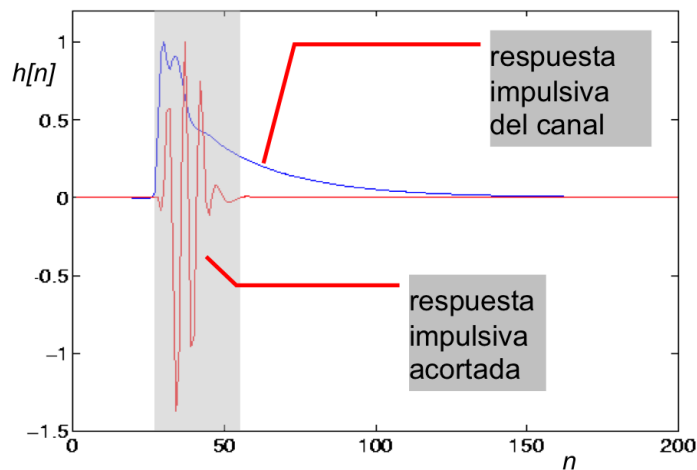


Figura 5: *Ecuador en el dominio del tiempo*

debe **conservar únicamente  $N$  de las muestras** de entrada al mismo, prestando especial atención a la reconstrucción del primer símbolo complejo (es decir, de  $X_0$ ).

### 3.3.4. Ecuador en el dominio de la frecuencia (FEQ)

El ecuador en el dominio de la frecuencia (FEQ, *Frequency-Domain Equalizer*) se incluye con el fin de compensar el efecto del canal sobre los símbolos transmitidos. En particular, el receptor debe implementar estos dos pasos:

1. Realizar una estimación del canal.
2. Deshacer el efecto del mismo:
  - a) Hallar la transformada DFT de  $2N$  puntos de la respuesta impulsiva del canal.
  - b) Una vez calculada, conservar únicamente los primeros  $N$  valores, que denotaremos como  $H_i$  ( $i = 0, \dots, N - 1$ ).
  - c) Dividir los  $N$  símbolos recibidos entre el  $H_i$  correspondiente, esto es determinar para cada  $i$  la relación  $X_i/H_i$  ( $i = 0, \dots, N - 1$ ).

En esta práctica, se obviaré el paso 1 y **se supondrá que el canal es perfectamente conocido por el receptor**. Por tanto, sólo han de implementarse los tres pasos enumerados como a), b) y c).

### 3.3.5. Decodificador QAM

De manera similar al transmisor, deben implementarse un conjunto de  $N$  decodificadores QAM, que produzcan a su salida los bits correspondientes a cada subcanal. La decodificación QAM puede realizarse por el método de máxima verosimilitud, suponiendo que todos los símbolos de cada constelación son equiprobables. En otras palabras, se trata de un **decodificador de mínima distancia**.

Evidentemente, para que el sistema funcione adecuadamente, el vector  $\mathbf{bn}$  (introducido en el apartado 3.1.1) debe ser conocido por el receptor.

### 3.3.6. Conversor Paralelo/Serie (P/S)

Este bloque debe realizar la tarea complementaria a la del conversor S/P del que consta el transmisor. Es decir, debe proporcionar **a su salida un único flujo binario**.

En la figura 6 se muestra un ejemplo de funcionamiento de este conversor, análogo al de la figura 2, y para los mismos valores del vector  $bn$ .

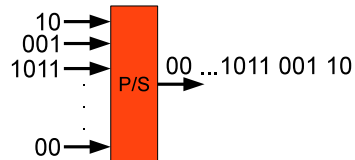


Figura 6: Ejemplo de funcionamiento del conversor P/S

## 4. La capa de enlace

En esta práctica, se considerará la transmisión de **una única trama rápida**, de 224 bytes de longitud. La capa de enlace no se simulará con el mismo nivel de detalle que la capa física, sino que han de implementarse únicamente los bloques descritos más adelante, ignorando multitud de aspectos que sí que habrían de considerarse en un sistema ADSL real.

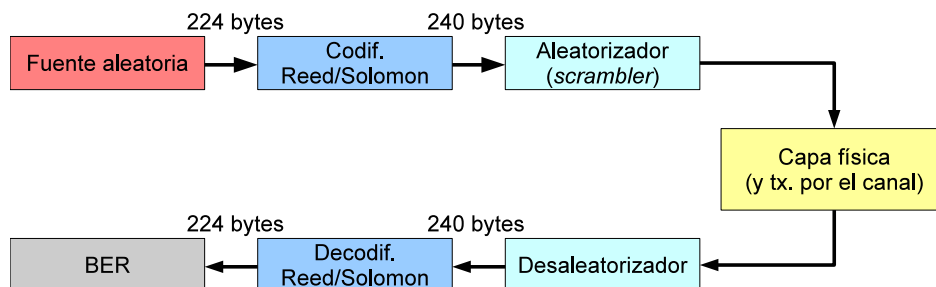


Figura 7: Diagrama de bloques del nivel de enlace

### 4.1. Transmisor

#### 4.1.1. Fuente aleatoria

Simúlese la construcción de la trama rápida como una fuente aleatoria de bits equiprobables, ignorando por completo los campos que constituyen realmente una trama rápida. La salida de este bloque debe ser una secuencia de 224 bytes, es decir, 1792 bits.

#### 4.1.2. Codificador Reed/Solomon

Para la codificación de canal, se utilizará un código Reed/Solomon (240,224). Es decir, la salida del codificador ha de ser una secuencia de 240 bytes de longitud. Recuérdese que el codificador de canal de ADSL opera **a nivel de byte** (con símbolos de 8 bits).

Para la implementación de este bloque se recomienda hacer uso de las funciones de MatLab `rsenc` y `gf`.

### 4.1.3. Aleatorizador

La aleatorización en ADSL opera en modo bit. Deben aleatorizarse todos los datos de entrada para garantizar una salida incorrelada y evitar largas cadenas de ceros o unos. Para ello, ha de aplicarse la siguiente expresión:

$$d_a[n] = d_i[n] \oplus d_a[n - 18] \oplus d_a[n - 23], \quad (5)$$

donde  $d_i[n]$  son los bits de entrada al aleatorizador y  $d_a[n]$  los bits de salida.

## 4.2. Envío de los datos a la capa física

Nótese que para la transmisión de los 240 bytes que constituyen la trama rápida, en general ha de utilizarse más de una trama física. Es decir, el conversor serie/paralelo de la sección 3.1.1 debe tomar tantos bits como indique la suma de los elementos del vector  $\mathbf{bn}$ , y distribuirlos entre todos los subcanales disponibles. Dado que la trama rápida (en el caso aquí analizado) contiene más bits de los que se pueden distribuir entre todos los subcanales, es necesario repetir este procedimiento y construir varias tramas físicas hasta poder transmitir por completo la trama rápida del nivel de enlace. Para el caso de la última trama física, puede ocurrir que haya más bits disponibles de los que realmente quedan por transmitir. De ser así, **añádanse ceros a la trama rápida** hasta cubrir por completo las necesidades de la capa física.

Los ceros así añadidos deberán ser eliminados más tarde por la capa de enlace del sistema receptor.

## 4.3. Receptor

### 4.3.1. Desaleatorizador

El desaleatorizador debe implementar la función complementaria a la dada por la expresión (5) para obtener un flujo de 240 bytes que en el caso ideal debe coincidir con la trama rápida transmitida.

Para ello, debe aplicarse la siguiente expresión:

$$d_o[n] = d_a[n] \oplus d_a[n - 18] \oplus d_a[n - 23], \quad (6)$$

siendo  $d_o[n]$  los bits de salida del desaleatorizador y  $d_a[n]$  los bits de entrada al mismo.

### 4.3.2. Decodificador Reed/Solomon

Este bloque debe eliminar la redundancia insertada por el codificador de canal, arrojando a su salida un flujo de 224 bytes. Para ello, pueden emplearse las funciones de MatLab `rsdec` y `gf`.

### 4.3.3. Cálculo de la BER

Este módulo se encarga únicamente de comparar los bits transmitidos con los recibidos y estimar así la probabilidad de error de bit (BER) resultante.

Para obtener una buena estimación de la BER, ha de repetirse el experimento completo un número de veces suficientemente elevado y promediar los resultados obtenidos en cada una de las repeticiones. En particular, realícese un **número de simulaciones comprendido entre 50 y 100**.



## 5. Desarrollo de la práctica

La práctica debe realizarse en MatLab en grupos de 2 ó 3 alumnos. El reparto de tareas entre los miembros de cada grupo puede hacerse como se desee; no obstante, todos los alumnos deben comprender el funcionamiento de cada uno de los bloques.

Cada grupo debe entregar una breve memoria en la que consten las gráficas y respuestas a las preguntas que se detallan en los siguientes apartados. No es necesario que la memoria contenga ninguna información extra: basta con incluir exactamente lo que se pide en los siguientes apartados. La fecha límite para la entrega de dicha memoria es el **viernes 2 de diciembre**.

Además de la memoria, deberán adjuntarse todos los ficheros de código MatLab realizados. No es necesario que contengan comentarios explicativos ni instrucciones de uso, pero sí se exigirá que exista un fichero principal, de nombre `principal.m`, que permita lanzar la simulación.

### 5.1. Diseño del sistema completo

Realice una simulación del sistema completo y dibuje la **gráfica de la BER obtenida como función de la SNR**, tomando para la misma valores comprendidos entre 15 y 30 dB. Nótese que, dado que se está manteniendo fija la energía transmitida, la SNR únicamente afecta en la varianza del ruido introducido por el canal, según la expresión dada por la ecuación (4). Incluya dicha gráfica en la memoria.

### 5.2. Comparativa con/sin aleatorizador

Elimine los bloques correspondientes al aleatorizador y al desaleatorizador, manteniendo intacto el resto del sistema. Repita la simulación para los mismos valores de SNR que en el apartado anterior. Obtenga la **gráfica de la BER** como función de la SNR e inclúyala en la memoria.

**Responda** brevemente a las siguientes preguntas:

1. ¿Mejoran o empeoran las prestaciones? ¿Por qué?
2. En caso de empeorar las prestaciones, ¿por qué cree que se incluye este bloque en los sistemas ADSL reales?

### 5.3. Comparativa con/sin FEQ

Vuelva a incluir el aleatorizador y el desaleatorizador en el sistema. Elimine ahora el equalizador en el dominio de la frecuencia (FEQ). Obtenga la **gráfica de la BER** como función de la SNR e inclúyala en la memoria.

### 5.4. Comparativa con una tasa diferente

Vuelva a incluir el FEQ. Establezca ahora los valores del vector `bn` como se detalla a continuación:

`bn(1:32) = 0`

`bn(33:100) = 4`

`bn(101:end) = 2`

Represente la **curva de la BER** en función de la SNR e inclúyala en la memoria. **Responda** brevemente a las siguientes cuestiones:

1. ¿Por qué empeora la probabilidad de error?
2. ¿Cree que es posible, para una tasa binaria fija, obtener una menor probabilidad de error simplemente redistribuyendo los bits de manera diferente entre las subportadoras (es decir, modificando el orden de los elementos de `bn`)?

### 5.5. Comparativa con un canal diferente

Vuelva a utilizar la configuración inicial para el vector `bn`; es decir:

`bn(1:32)=0`

`bn(33:256)=2`

Modifique el canal para que se ajuste a la siguiente respuesta impulsiva:

$$h[n] = \alpha_0\delta[n] + \alpha_3\delta[n - 20] + \alpha_6\delta[n - 40] + \alpha_{20}\delta[n - 50], \quad (7)$$

donde los coeficientes  $\alpha$  son los mismos que los empleados anteriormente (véase la sección 3.2) y se han modificado únicamente los valores de los retrasos.

Obtenga nuevamente la **gráfica de la BER** en función de la SNR y adjúntela en la memoria. **Responda** a esta pregunta:

1. ¿Observa alguna diferencia respecto a la gráfica inicial? ¿A qué se debe este efecto?