

2 MODULACIÓN OFDM

2.1 FUNDAMENTOS DEL PROCESAMIENTO DE SEÑAL OFDM

2.1.1 INTRODUCCIÓN AL PROCESAMIENTO DE SEÑAL OFDM

La tecnología a la que hacemos referencia mediante OFDM (Orthogonal Frequency Division Multiplexing) puede ser vista como una colección de técnicas de transmisión. Cuando se aplican en un medio “no cableado”, como es el caso de la radiodifusión, son usualmente referenciadas como OFDM. Sin embargo, en medios “cableados”, el término DMT (Discrete Multitone) es más usual.

2.1.1.1 INTRODUCCIÓN HISTÓRICA

La historia de OFDM data de mediados de los años 60, cuando Chang publicó su artículo sobre la síntesis de señales limitadas en banda para transmisiones multicanal. Chang presentó una serie de principios para la transmisión simultánea de mensajes a través de un canal lineal limitado en banda, evitando los perjudiciales efectos derivados de la Interferencia entre Símbolos (ISI) y entre Canales (ICI). Poco después de que Chang publicara sus estudios, Saltzberg realizó un análisis del comportamiento de estos sistemas, concluyendo que:

”La estrategia de diseño de un sistema paralelo eficiente debe centrarse más en reducir la diafonía entre canales adyacentes que en el perfeccionamiento individual de los mismos, dado que la distorsión por diafonía tiende a ser dominante.”

Una mayor contribución a OFDM fue la realizada en 1971 por Weinstein y Ebert, quienes usaron la Transformada Discreta de Fourier (DFT) para desarrollar la modulación y demodulación en banda base. Este trabajo no se orientó hacia el “perfeccionamiento individual” de los canales sino hacia un procesamiento eficiente, eliminando los bancos de osciladores correspondientes a las distintas subportadoras. Para combatir la ISI e ICI usaron tanto un espacio de guarda entre símbolos, como un enventanado por coseno alzado en el dominio temporal. Su sistema no consiguió alcanzar una perfecta ortogonalidad entre subcanales pero es todavía hoy la mayor contribución aportada a OFDM.

Otro importante adelanto fue llevado a cabo por Peled y Ruiz en 1980, quienes introdujeron el uso del prefijo cíclico (CP) o extensión cíclica, resolviendo el problema de la ortogonalidad. En vez de usar un espacio de guarda vacío, lo rellenaron con una extensión cíclica del símbolo OFDM.

2.1.1.2 MODELO DEL SISTEMA

Abordaremos el modelado genérico de los sistemas OFDM de manera discreta en el tiempo, y suponiendo todas las señales en banda base, evitando así una elevada carga computacional de las simulaciones. Ello nos dará una visión más aproximada de la implementación digital que dichos sistemas soportan en la actualidad.

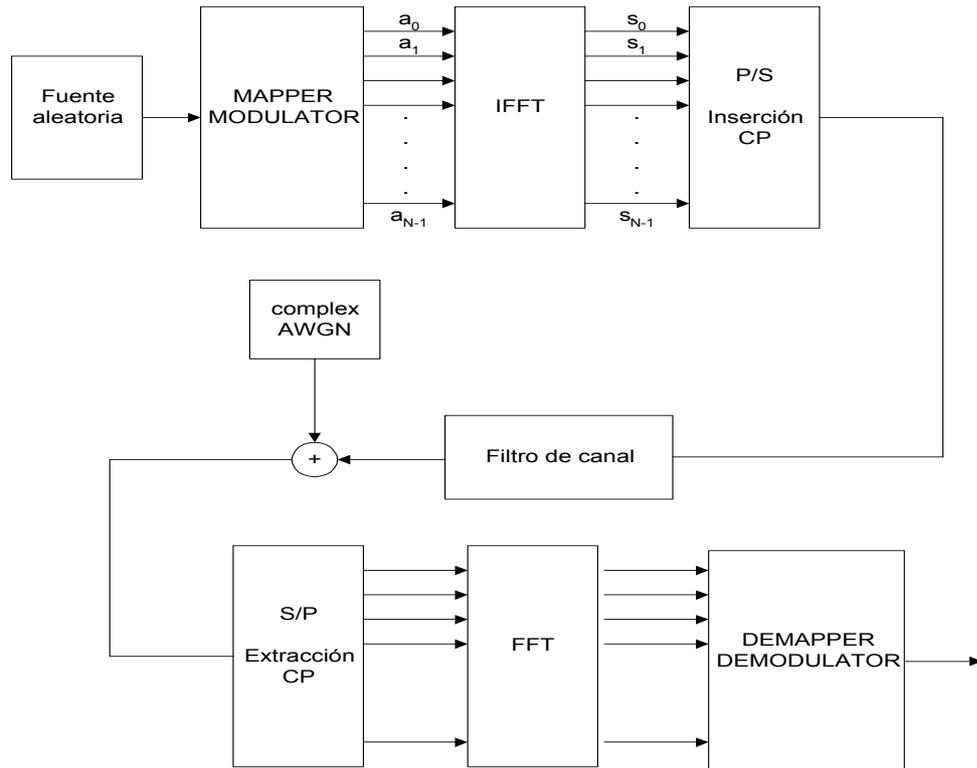


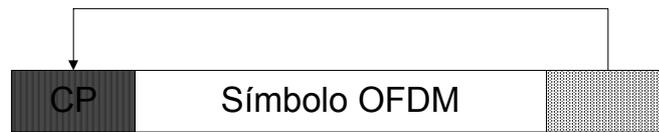
Figura 2.1: Modelo de sistema OFDM

El anterior sistema permite la transmisión en paralelo de un conjunto de N símbolos M-arios, habiéndose formado previamente dichos símbolos a partir del flujo binario generado en una fuente aleatoria. En la práctica, la entrada de nuestro sistema será atacada por una línea digital proveniente de una cadena de codificadores de canal.

Tras el Mapper/Modulator el flujo binario ha sido mapeado (M-PSK o M-QAM) y convertido de serie a paralelo, con lo que se tiene un conjunto de N números complejos a_k provenientes de una cierta constelación de señales.

Los datos a_k son entonces modulados sobre N subportadoras mediante la aplicación de la Transformada Inversa de Fourier (IDFT), generándose los valores complejos s_k . Esta señal ataca a un serializador que copia las últimas L muestras a

modo de preámbulo o prefijo cíclico (CP) y origina el símbolo OFDM, el cual será transmitido a través de un canal discreto en el tiempo.



En el receptor, la extensión cíclica es extraída y la señal resultante se demodula mediante la aplicación de un bloque DFT. La inserción del prefijo cíclico es comúnmente aceptada como la forma de evitar la ISI y preservar la ortogonalidad entre los tonos.

2.1.1.3 MODELO ANALÍTICO

El modelo analítico del sistema de transmisión OFDM viene dado por las ecuaciones que expresan la funcionalidad de los distintos bloques que lo integran. Dada la simetría del mismo nos centraremos en el procesado de la señal en transmisión así como el efecto que sobre la misma realiza el canal.

Las señales complejas a_k provenientes de la constelación de símbolos M-arios, alimentan al bloque IDFT generando s_k , la cual es serializada y transmitida:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} a_k e^{j2\pi kn/N} \quad n = 0, \dots, N-1$$

Supongamos un canal discreto dispersivo de la forma:

$$h(n) = \sum h(i)\delta(n-i)$$

Dicho canal debe tener una longitud inferior a la del CP, siendo ésta una condición esencial a la hora de preservar la ortogonalidad de las subportadoras.

El efecto del canal sobre $s(n)$ nos da la siguiente señal en recepción:

$$r(n) = \sum h(i)s(n-i) \quad (1)$$

$$r(n) = \frac{1}{N} \sum h(i) \left[\sum_{k=0}^{N-1} a_k e^{j2\pi k(n-i)/N} \right] \quad (2)$$

$$r(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum h(i) a_k e^{-j2\pi ki/N} \right] e^{j2\pi kn/N} \quad (3)$$

donde definimos:

$$Z_K = \sum h(i) a_k e^{-j2\pi ki/N} = a_k \sum h(i) e^{-j2\pi ki/N} = a_k H_k \quad (4)$$

como las señales de entrada al bloque DFT. Luego a la salida de dicho bloque observamos que el efecto del canal sobre los símbolos transmitidos ha sido un escalado, lo que facilita enormemente la ecualización.

2.1.1.4 PROPIEDADES DE LOS SISTEMAS OFDM

Ventajas:

- Realizan un uso eficiente del espectro.
- Al dividir el canal en subcanales de banda estrecha el sistema se hace más robusto ante caídas selectivas en frecuencia.
- Eliminan el efecto de la ICI e ISI gracias a la inserción del CP.
- Usando una determinada codificación de canal y entrelazado el sistema puede responder ante pérdidas de datos durante la transmisión.
- La ecualización es más sencilla que en los sistemas de portadora única.
- OFDM es computacionalmente eficiente debido al uso de la FFT.

- En unión a técnicas de modulación diferencial desaparece la necesidad de realizar estimación de canal.

Inconvenientes:

- Los sistemas OFDM son altamente sensibles al offset, tanto frecuencial como temporal.
- La señal OFDM presenta un elevado rango dinámico, por lo que requiere amplificadores de potencia en radio frecuencia con una elevada relación “potencia de pico frente a potencia media”.

2.1.2 SINCRONIZACIÓN**2.1.2.1 INTRODUCCIÓN**

El principal argumento en contra de OFDM es sin duda su alta sensibilidad a errores de sincronización, en particular al offset frecuencial. En lo que sigue veremos los tres niveles en los que podemos situar el problema de la sincronía: sincronización de símbolo, de frecuencia de portadora y de frecuencia de muestreo.

2.1.2.2 SINCRONIZACIÓN DE SÍMBOLO

En los sistemas OFDM se presta una gran atención a la sincronización de símbolo por su gran importancia, sin embargo los requerimientos de temporización se relajan gracias a la inclusión del CP.

Un offset de temporización provoca una rotación de las subportadoras dentro de la constelación de señales usada, mayor cuanto más cerca se encuentren de los extremos de la banda de frecuencia. Lo veremos de manera analítica.

Supongamos que la señal en recepción lleva un retraso temporal de $n\varepsilon$ muestras no corregido:

$$r(n - n\varepsilon) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum h(i) a_k e^{-j2\pi ki/N} \right] e^{j2\pi k(n-n\varepsilon)/N} \quad (1)$$

Los puntos de la constelación que toma el demodulador en recepción se ven afectados de dos formas distintas:

$$Z_k = a_k \left[\sum h(i) e^{-j\frac{2\pi ki}{N}} \right] e^{-j\frac{2\pi kn\varepsilon}{N}} \quad (2)$$

En primer lugar a_k toma el efecto del canal mediante el producto con la función de transferencia $H(k)$:

$$H(k) = \sum h(i) e^{-j\frac{2\pi ki}{N}}$$

En segundo lugar a_k experimenta una rotación que viene dada por:

$$e^{-j\frac{2\pi kn\varepsilon}{N}}$$

Podemos comprobar como dicho efecto es más intenso a medida que la frecuencia aumenta, es decir, cuanto más nos aproximemos al extremo de la banda. Veamos gráficamente el efecto sobre una constelación QPSK:

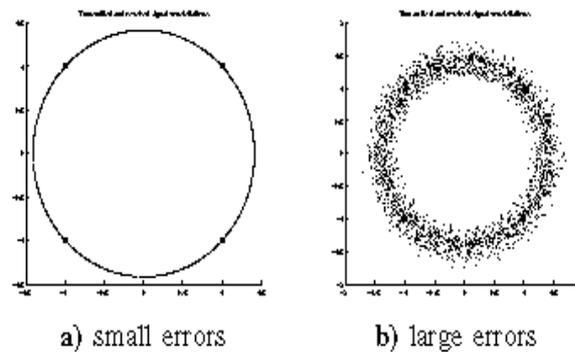


Figura 2.2: Efecto de canal y error de sincronía de símbolo sobre una constelación QPSK.

Si el error de sincronización es suficientemente pequeño como para mantener la respuesta impulsiva del canal dentro del CP, la ortogonalidad se mantiene. En este caso el offset puede ser visto como un desplazamiento de fase introducido por el canal, y las rotaciones de fase estimadas mediante un estimador de canal. Si el desplazamiento temporal es mayor que la duración del CP, la ISI afectará a nuestro sistema.

La literatura recoge dos métodos principales para afrontar este problema: basados en pilotos o en el prefijo cíclico.

Un algoritmo del primer tipo fue sugerido por Warner y Leung, ellos usaron un esquema donde la señal OFDM se transmitía en FM, y el transmisor codificaba un cierto número de subcanales reservados con fases y amplitudes conocidas. El algoritmo está compuesto de tres fases: detección de potencia, sincronización gruesa y sincronización fina.

La primera fase (detección de potencia) detecta si hay presencia de señal OFDM midiendo la potencia recibida y comparándola con un umbral. La segunda fase (sincronización gruesa) se usa para adquirir un alineamiento de sincronización con un

error máximo de 0.5 muestras. Este grado de alineamiento no es aceptable, pero sirve para simplificar el posterior algoritmo de captura, el cual puede suponer que el error de temporización es pequeño. La sincronización gruesa se consigue correlando la señal recibida con una copia retrasada de la misma.

Para detectar el pico de correlación con la suficiente precisión se usa un filtro digital que interpola los datos a una velocidad de cuatro veces la original. En la última fase (sincronización fina), los subcanales reservados se ecualizan con la estimación de canal obtenida a partir de los pilotos.

Dado que la sincronización gruesa garantiza que el error de temporización es menor que 0.5, la respuesta impulsiva del canal se encuentra dentro del prefijo cíclico.

Existen también algoritmos de sincronización basados en el prefijo cíclico. La diferencia entre dos muestras recibidas espaciadas N posiciones se puede representar como $r(k) - r(k + N)$. Cuando una muestra pertenece al prefijo cíclico y la otra al símbolo OFDM del cual se ha copiado, dicha diferencia es pequeña. Si la inventanamos con una ventana rectangular de la misma longitud que el prefijo cíclico obtendremos una serie de mínimos, cada uno de los cuales se encontrará al comienzo de los símbolos OFDM.

Esta idea se formaliza matemáticamente mediante la función de verosimilitud (likelihood function), la cual se maximiza para obtener de manera simultánea los offset temporal y frecuencial. Veámosla para el caso de offset frecuencial nulo y offset temporal θ :

$$\Lambda(\theta) = \sum_{k=\theta}^{\theta+L-1} \frac{2}{SNR+1} \operatorname{Re}\{r(k)r^*(k+N)\} - \frac{SNR}{SNR+1} |r(k) - r(k+N)|^2$$

2.1.2.3 SINCRONIZACIÓN DE FRECUENCIA DE MUESTREO

La señal continua recibida debe ser muestreada en instantes de tiempo dados por el reloj del receptor. Existen dos tipos de métodos que tratan con el error en la frecuencia de muestreo.

En sistemas de muestreo sincronizado un algoritmo de temporización controla un oscilador de cristal controlado por tensión con objeto de alinear los relojes de transmisor y receptor. El otro método es el de muestreo no sincronizado, en el cual la tasa de muestreo permanece fija y se realiza un post-procesado digital.

El efecto de un offset en la frecuencia de reloj es doble: la componente útil de la señal se ve rotada y atenuada, y por otro lado se introduce ICI en el sistema.

A partir de los estudios publicados hasta la fecha se desprende que los sistemas de muestreo no sincronizado son mucho más sensibles al offset de reloj que los de muestreo sincronizado.

2.1.2.4 SINCRONIZACIÓN DE FRECUENCIA DE SUBPORTADORA

2.1.2.4.1 ERRORES FRECUENCIALES

El offset frecuencial se produce como consecuencia de diferencias en los osciladores de transmisor y receptor, desplazamiento Doppler o desfases introducidos por canales no lineales. Hay dos efectos destructivos provocados por el offset en la frecuencia de portadora: uno es la reducción en la amplitud de la señal como consecuencia de que las senoides no se muestrean en su máximo, el otro es la aparición

de ICI en el sistema. Este último es provocado por la pérdida de ortogonalidad entre los subcanales.

Pollet evaluó analíticamente la degradación en la BER provocada por la presencia de offset frecuencial de portadora para un canal AWGN. Denotemos el offset frecuencial relativo, normalizado por el espacio entre subcanales como:

$$\Delta f = \frac{\Delta F}{W/N}$$

donde ΔF es el offset frecuencial, N el número de subportadoras y W el ancho de banda de la señal OFDM. La degradación D que sufre la SNR (en dB) puede ser aproximada por:

$$D(\text{dB}) \approx \frac{10}{3 \ln 10} (\pi \Delta f)^2 \frac{E_s}{N_o} = \frac{10}{3 \ln 10} \left(\pi \frac{N \Delta F}{W} \right)^2 \frac{E_s}{N_o}$$

Vemos como D aumenta con el cuadrado del número de canales cuando ΔF y W son constantes.

2.1.2.4.2 ESTIMADORES DE FRECUENCIA

Al igual que en el caso de la sincronización de símbolo, los distintos esquemas de sincronización de portadora se basan en el uso de pilotos o en el aprovechamiento de las características del prefijo cíclico.

En los algoritmos basados en pilotos, algunas subportadoras se usan para la transmisión de símbolos conocidos. Este conocimiento permite la estimación de las rotaciones de fase, las cuales tienen una correspondencia directa con el offset frecuencial.

Las técnicas basadas en el CP utilizan la redundancia introducida por éste de dos maneras: creando una función que alcanza su máximo en el instante de offset nulo y hallando dicho máximo, o bien usando estimación de máxima verosimilitud.

Es interesante resaltar la relación entre la sincronización temporal y frecuencial. Si deseamos facilitar ésta última, lo podemos hacer rebajando el número de subportadoras, lo cual hace aumentar el espaciado entre ellas, sin embargo esto dificultaría la sincronización temporal debido a que la disminución en N reduciría la longitud de los símbolos. Podemos ver por tanto que ambos aspectos se encuentran estrechamente relacionados. Matemáticamente dicha relación se expresa como:

$$\Delta f_c \frac{N}{W} = 1$$

donde Δf_c es el espaciado frecuencial entre subportadoras, N el número de subportadoras y W el ancho de banda de la señal OFDM.

2.1.3 ECUALIZACIÓN EN SISTEMAS OFDM

2.1.3.1 ESTIMACIÓN DE CANAL

Las técnicas de modulación pueden clasificarse en coherentes o diferenciales. Cuando se usa modulación diferencial no se necesita la estimación de canal, ya que la información está codificada en la diferencia entre dos símbolos consecutivos. Esta es una técnica común en sistemas inalámbricos, en los que si no se necesita implementar el estimador de canal se reduce la complejidad del receptor. La modulación diferencial se usa por ejemplo en el estándar europeo para DAB (Digital Audio Broadcasting).

Los inconvenientes de la modulación diferencial son una pérdida de 3-4 dB en la SNR, y la imposibilidad de aumentar la eficiencia espectral con constelaciones multiamplitud. Una interesante alternativa a la modulación coherente es la modulación

DAPSK (Diferential Amplitude PSK), donde la eficiencia espectral es mayor que en DPSK, y es alcanzada con una codificación diferencial también de la amplitud. Esto requiere una distribución de amplitud no uniforme.

Por otro lado, la modulación coherente permite usar constelaciones de señal arbitrarias y es una elección obvia en los sistemas cableados, donde el canal apenas cambia con el tiempo. En sistemas inalámbricos la eficiencia de la modulación coherente la hace interesante cuando la tasa binaria a transmitir es alta, como ocurre en los estándares europeos DVB (Digital Video Broadcasting) e HIPERLAN/2.

Dichos esquemas de modulación coherente presentan el problema de la necesidad de ecualización, lo cual hace necesario el desarrollo de algoritmos de estimación de canal.

Existe una amplia gama de algoritmos de estimación de canal, si bien éstos derivan por lo general de lo que veremos a continuación.

2.1.3.1.1 ESTIMADORES DE CANAL BASADOS EN LA DFT

Consideremos el siguiente sistema:

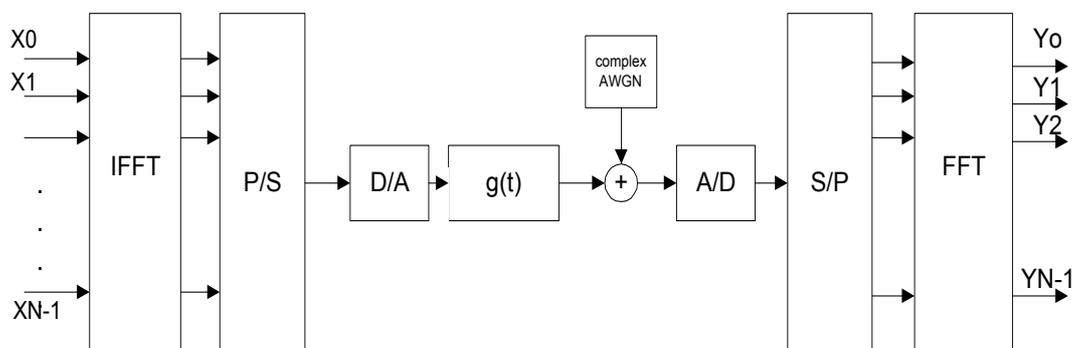


Figura 2.3: Modelo OFDM continuo.

Podemos modelar la respuesta impulsiva del canal según la expresión:

$$g(t) = \sum_m \alpha_m \delta(t - \tau_m T_s) \quad (1)$$

donde las amplitudes α_m son valores complejos y la duración de $g(t)$ es inferior a la del espacio de guarda.

El sistema se modela en base a una DFT de N puntos como:

$$Y = DFT_N \left(IDFT_N(X) \otimes \frac{g}{\sqrt{N}} + \hat{n} \right) \quad (2)$$

donde \otimes denota la convolución cíclica, n un vector de ruido complejo gaussiano y g se determina por el equivalente cíclico de funciones sampling.

$$G_k = \frac{1}{\sqrt{N}} \sum_m \alpha_m e^{-j \frac{\pi}{N} (k + (N-1)\tau_m)} \frac{\sin(\pi \tau_m)}{\sin\left(\frac{\pi}{N}(\tau_m - k)\right)} \quad (3)$$

La validez del anterior modelo depende de cómo de bien el espacio de guarda elimine la ISI, y puede ser reescrito en base a un conjunto de N canales gaussianos independientes:

$$Y_k = H_k X_k + n_k \quad k = 0 \dots N-1 \quad (4)$$

donde H_k es la atenuación compleja del canal dada por $H = DFT_N(g)$ y

$n = DFT_N(\hat{n})$ un vector de ruido gaussiano complejo. Si lo expresamos de manera matricial aparece de una forma más compacta:

$$Y = XFg + n$$

Basándonos en el anterior desarrollo matemático nos planteamos el obtener una expresión que nos de una estimación de la respuesta impulsiva del canal lo más exacta posible. Para ello establecemos dos criterios estadísticos distintos, los cuales darán lugar

a las estimaciones conocidas como MMSE (Minimum Mean Square Error) y LSE (Least Square Error). A continuación se muestran los resultados obtenidos para cada criterio:

- **MMSE:**

$$\hat{\mathbf{g}}_{MMSE} = \mathbf{R}_{gy} \mathbf{R}_{yy}^{-1} \mathbf{y}$$

donde $\mathbf{R}_{gy} = E(\mathbf{g}\mathbf{y}^H) = \mathbf{R}_{gg} \mathbf{F}^H \mathbf{X}^H$

$$\mathbf{R}_{yy} = E(\mathbf{y}\mathbf{y}^H) = \mathbf{X}\mathbf{F}\mathbf{R}_{gg}\mathbf{F}^H\mathbf{X}^H + \sigma_n^2 \mathbf{I}_N$$

son matrices de covarianza cruzada.

- **LSE:**

$$\hat{\mathbf{g}}_{LS} = \mathbf{X}^{-1} \mathbf{y}$$

2.1.3.2 ECUALIZACIÓN

Partamos del modelo discreto de sistema OFDM representado en la figura 2.1.1, donde en adelante llamaremos X_m a cada uno de los N símbolos que serán modulados mediante la IDFT. Las muestras de la señal OFDM resultante se pueden expresar como:

$$x_n = \frac{1}{N} \sum_{m=0}^{N-1} X_m e^{j2\pi nm/N} \quad 0 \leq n \leq N$$

donde x_n representa la n -ésima muestra de salida del bloque IDFT.

Consideremos el canal discreto constituido por L muestras, la señal recibida se puede expresar como:

$$y_n = \sum_{l=0}^{L-1} h_{n,l} x_{n-l} + w_n = h_{n,0} x_n + h_{n,1} x_{n-1} + \dots + h_{n,L-1} x_{n-L+1} + w_n \quad 0 \leq n \leq N$$

donde $h_{n,l}$ representa la l -ésima muestra compleja de la respuesta impulsiva del canal y w_n es el ruido AWGN en el instante temporal n .

Por simplicidad se ha obviado en las ecuaciones al prefijo cíclico correspondiente a las N muestras del símbolo OFDM. La señal demodulada en el dominio frecuencial se obtiene mediante la DFT de y_n como:

$$Y_m = \sum_{k=0}^{N-1} \sum_{l=0}^{L-1} X_k H_l^{m-k} e^{-j2\pi k l / N} + W_m \quad 0 \leq m \leq N-1$$

donde W_m denota la DFT de w_n y H_l^{m-k} representa la DFT de la respuesta impulsiva del canal como:

$$H_l^{m-k} = \sum_{n=0}^{N-1} h_{n,l} e^{-j\pi n(m-k) / N}$$

Si expresamos el anterior desarrollo de manera matricial obtendremos una expresión muy compacta del tipo:

$$Y = HX + W$$

donde cada uno de los elementos de la matriz H puede ser reescrito como:

$$a_{m,k} = H_0^{m-k} + H_1^{m-k} e^{-j2\pi k / N} + \dots + H_{L-1}^{m-k} e^{-j2\pi k(L-1) / N}$$

$$0 \leq m, k \leq N-1$$

Dada la anterior expresión matricial el problema de la equalización se centra en estimar la matriz H y calcular su inversa, lo cual requiere algoritmos encaminados hacia la reducción de la carga computacional que las operaciones matriciales requieren.

2.2 MODELADO EN MATLAB DE UN SISTEMA BÁSICO DE COMUNICACIONES OFDM

2.2.1 HIPÓTESIS DE PARTIDA

A lo largo de esta sección se describirá el modelo base sobre el cual se han realizado las distintas pruebas y simulaciones.

Puesto que se ha intentado modelar un sistema lo más genérico posible, obviaremos en esta sección los problemas de ecualización y sincronización, centrándonos en mostrar el procesamiento OFDM en sus distintas etapas, así como los efectos negativos que causa el canal de transmisión sobre la señal.

El presente estudio se ha enfocado desde un modelo básico. Este hecho limita la exactitud de los resultados obtenidos dado que no considera los procesos inherentes a la conversión A/D y D/A, los cuales implican una serie de operaciones de procesado de señal tales como filtrado y muestreo. En contrapartida, el modelado discreto permite una observación más directa y sencilla del sistema de comunicaciones.

Es importante ser consciente de las limitaciones introducidas por la herramienta de simulación utilizada, el paquete MATLAB 5.2. Dicho software realiza las operaciones en punto flotante, introduciendo redondeo tras un cierto número de cifras. Los sistemas digitales operan generalmente con aritmética en punto fijo, dada su mayor sencillez y menor coste hardware, por tanto un grado de abstracción menor en las simulaciones requeriría introducir nuevas líneas de código que simulasen el efecto del punto fijo.

Las características básicas del sistema OFDM presentado son:

- La señal de información constituye un proceso aleatorio uniformemente distribuido.
- Nfft subportadoras $W_k = \frac{2\pi}{N_{fft}}k \quad k = 0, \dots, N_{fft} - 1$.
- Modulación coherente 16-QAM.
- Canal dispersivo con adición de ruido complejo AWGN.
- Demodulación sectorial.

A continuación se desglosan los distintos módulos que integran el modelo.

2.2.2 TRANSMISOR

El diagrama de bloques del transmisor modelado es el siguiente:

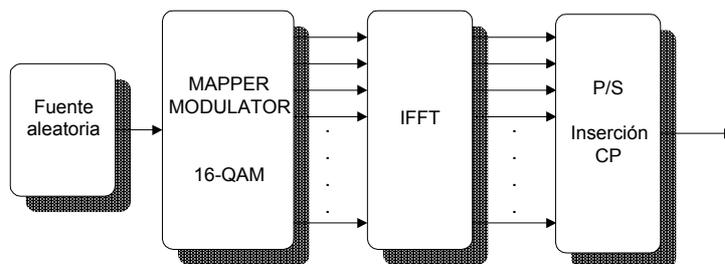


Figura 2.4: Transmisor OFDM.

El anterior esquema se ha implementado en MATLAB mediante tres módulos, de los cuales el principal toma parámetros de entrada desde la línea de comandos e invoca al módulo secundario, valiéndose a su vez éste último del tercero.

Se ha intentado dotar a los códigos de modularidad con el objeto de facilitar cambios en la implementación.

El primer código que se muestra es el del módulo principal, el cual llama a los módulos secundarios.

```
function [trama,sinf]=tbb(Nt,Ncp)
```

- Parámetros de entrada:

Nt: nº de super-símbolos OFDM transmitidos

Ncp: nº de muestras del CP

- Parámetros de salida:

trama: señal que ataca al canal.

sinf: fuente de información aleatoria que ataca
al modulador OFDM.

El módulo *tbb* simula el procesador digital OFDM en transmisión, realizando las siguientes operaciones:

- Crea la señal de información aleatoria
- Modula en 16-QAM
- Halla la IFFT
- Añade el CP

Por otro lado, la función *tbb2* toma la señal de información y la longitud del prefijo cíclico, realizando las siguientes operaciones:

function so=tbb2(si,Ncp)

- Parámetros de entrada:

si: señal de información procedente de la fuente aleatoria

Ncp: nº de muestras del CP

- Parámetros de salida:

so: señal modulada en OFDM + CP.

El anterior módulo tiene como misión estructurar las muestras de información en bloques de tamaño N_{fft} para el posterior procesado de *tbb1*, cuyo resultado lo toma *tbb2* para formar la trama de salida del transmisor.

A continuación veremos como *tbb1* realiza las operaciones básicas para OFDM. Toma las N_{fft} muestras que se transmitirán de forma paralela, las modula en 16-QAM y calcula la IFFT (N_{fft} puntos) de las mismas. Finalmente, a cada símbolo se le inserta un prefijo cíclico de N_{cp} muestras.

function [spc,sem]=tbb1(se,Ncp)

- Parámetros de entrada:

se: bloque de señal de información procedente de la fuente.

Ncp: nº de muestras del CP.

- Parámetros de salida:

spc: señal modulada en OFDM + CP.

sem: señal modulada 16-QAM.

Es necesario reseñar el hecho de que al usarse una modulación M-aria para $M=16$, el modulador digital tomaría palabras de tamaño 4 bits. Por simplicidad el código MATLAB que lo implementa y que a continuación se expone interpreta dichas palabras como valores enteros comprendidos entre 0 y $2^4 - 1$.

El código aplicado es el Gray puesto que presenta unas características muy robustas de cara a rotaciones en la constelación provocadas por el canal y/o errores de sincronización temporal o frecuencial.

El módulo que implementa el bloque 16-QAM identifica la muestra de entrada y da a su salida el punto de la constelación correspondiente.

function [r,i]=QAM(sie)

- Parámetros de entrada:

sie: muestra de información.

- Parámetros de salida:

r: parte real de la muestra modulada.

i: parte imaginaria de la muestra modulada.

2.2.3 CANAL

El canal de transmisión es un elemento crucial dentro del modelado de un sistema de comunicaciones. El procesamiento de la señal en transmisión y recepción va

encaminado a acomodar la señal a las características del mismo y a reducir los efectos perjudiciales sobre la información que transporta.

De cara a aportar generalidad al modelo, consideraremos un canal discreto compuesto de tres etapas:

- Retraso temporal puro $\delta(n - \theta)$.
- Respuesta impulsiva $h(n)$ correspondiente a un canal dispersivo, con una longitud inferior a la del prefijo cíclico.

$$h(n) = K \exp(-\alpha n) \quad n = 0, \dots, L-1 \quad L \leq N_{cp}$$

- Adición de ruido complejo AWGN.

El esquema de bloques de dicho canal es:

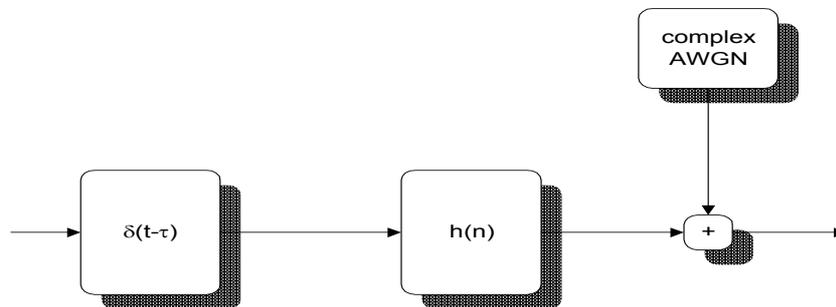


Figura 2.5: Canal de transmisión.

El módulo *canal* toma una serie de parámetros de entrada que permiten definir las características de la respuesta impulsiva del canal, así como el retraso que introduce. Una vez que el código toma la amplitud, duración y coeficiente de caída de $h(n)$ se realiza una normalización del mismo de forma que su energía sea unitaria:

$$E_h = \sum_{n=0}^{L-1} |\hat{h}(n)|^2 = 1$$

Por otra parte, el módulo evalúa la potencia de la señal filtrada, añadiendo ruido complejo AWGN para una cierta SNR que se da como parámetro de entrada.

function r=canal(s,ret,SNR,n,K,alfa)

- Parámetros de entrada:

s: señal procedente del transmisor digital

ret: nº de muestras de retraso que introduce el canal

SNR: relación señal a ruido en dB tras los procesos de filtrado

n: longitud de la respuesta impulsiva del canal dispersivo

K: amplitud de la respuesta impulsiva del canal dispersivo

alfa: coeficiente de caída de la respuesta impulsiva del canal dispersivo

- Parámetros de salida:

r: señal de salida del canal

El módulo canal realiza los siguientes procesos:

- Introduce un retraso puro.
- Filtra con $h(n)=K*\exp(-\text{alfa}*n)$.
- Introduce ruido AWGN complejo.

2.2.4 RECEPTOR

El módulo receptor realiza la demodulación OFDM de la señal proveniente del canal, permitiéndonos conocer la información de fuente transmitida.

Su diagrama de bloques es:

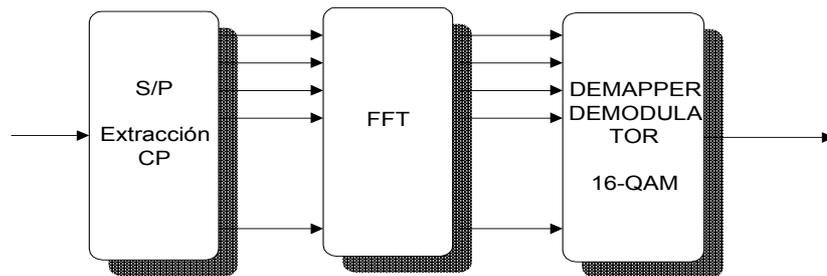


Figura 2.6: Receptor OFDM.

El código MATLAB que implementa la anterior estructura presenta ciertos aspectos importantes a destacar.

El parámetro de entrada *origen* indica al receptor el comienzo de la señal OFDM útil. Esta entrada se supone que coincide con la salida del bloque sincronizador, el cual se considera externo al procesador en recepción a efectos de modelado.

El módulo receptor evalúa la calidad de la transmisión mediante el cálculo de la BER, por ello se le da el mensaje original *men*. Dicha información de fuente es traducida a binario y luego comparada bit a bit con la salida del demodulador.

Veamos el módulo principal del receptor:

```
function [sd,ber,scons]=rbb(si,origen,men,Ncp,Nt)
```

- Parámetros de entrada:

si: señal proveniente del canal.

origen: comienzo temporal de la señal de información.

men: mensaje originado en la fuente aleatoria.

Ncp: nº de muestras del CP.

Nt: nº de super-símbolos OFDM transmitidos.

- Parámetros de salida:

sd: señal de información aleatoria proveniente
del demodulador OFDM.

ber: a la salida del receptor.

scons: puntos de la constelación recibidos.

El módulo rbb simula el procesador digital OFDM en recepción, realizando las siguientes operaciones:

- Paraleliza extrayendo el CP
- Halla la FFT
- Demodula 16-QAM

El módulo anterior se apoya en otro secundario *rbb1* que realiza propiamente las funciones de procesamiento de señal:

```
function ssm=rbb1(spc)
```

- Parámetros de entrada:

spc: super-símbolo OFDM + CP.

- Parámetros de salida:

ssm: señal que ataca al demodulador 16-QAM.

Este módulo toma un vector de $N_{fft}+N_{cp}$ símbolos, extrae el CP y lo transforma FFT (N_{fft} puntos).

El funcionamiento del demodulador 16-QAM se ha basado en evaluar la región de la constelación en la que se encuentra el valor complejo recibido. Esta forma de operar es válida en tanto que la rotación provocada por errores de sincronización no sea crítica, lo cual desgraciadamente sucede en la práctica para desfases muy pequeños.

```
function ss=dQAM(sd)
```

- Parámetros de entrada:

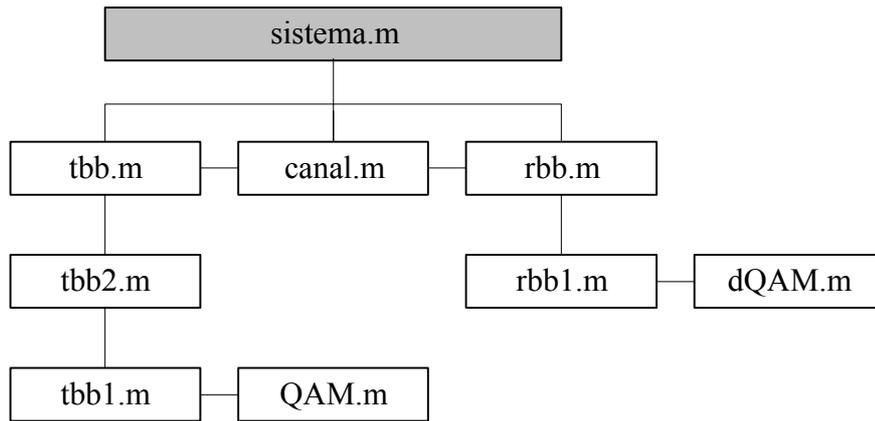
sd: muestra de información modulada.

- Parámetros de salida:

ss: muestra de información demodulada.

2.2.5 JERARQUÍA DE MÓDULOS

A continuación vemos gráficamente la relación entre todas los módulos implementados en Matlab descritos anteriormente:



Si queremos realizar la simulación del sistema para unos parámetros dados, tendríamos que arrancar Matlab y cambiarnos a un directorio donde estuviesen todas las funciones vistas. Ejecutando la función `sistema.m` (con entradas adecuadas), obtenemos todas las variables útiles del sistema, incluida la BER. Si después ejecutamos la función `dibcons.m` (con entrada `scons`) que se mostrará más adelante, obtendremos la constelación de puntos recibidos. Si además ejecutamos la función `dep.m`, que también veremos, tendremos en pantalla la densidad espectral de potencia de la señal OFDM en banda base que ataca al canal.

2.2.6 SIMULACIÓN MATLAB DE SISTEMA BÁSICO OFDM PARA HIPERLAN/2

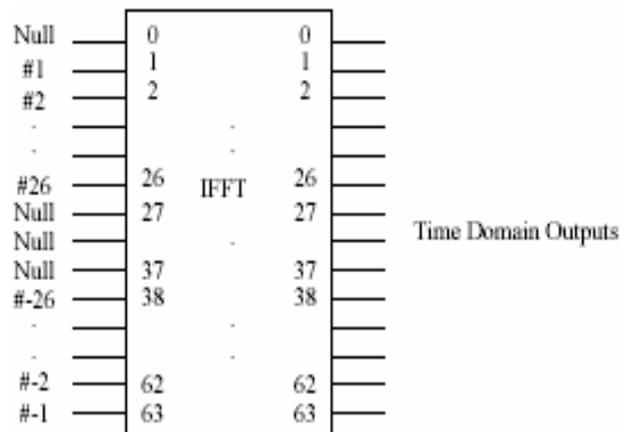
2.2.6.1 ESPECIFICACIONES PARA SUBSISTEMA OFDM EN HIPERLAN/2

Los siguientes valores numéricos son los que se han usado para las simulaciones, y se pueden consultar en el estándar ETSI TS 101 475 V1.3.1:

PARÁMETRO	VALOR
Tasa de muestreo: $f_s=1/T$	20 MHz
Duración de la parte útil del símbolo: T_u	$64 \times T = 3.2 \mu s$
Duración del prefijo cíclico: T_{cp}	$16 \times T = 0.8 \mu s$
Intervalo de símbolo: T_{sim}	$80 \times T = 4 \mu s (T_u + T_{cp})$
Número de subportadoras para datos: N_{sd}	48
Número de subportadoras para pilotos: N_{sp}	4
Número total de subportadoras usadas: N_{st}	$52 (N_{sd} + N_{sp})$
Número de puntos de algoritmos FFT/IFFT: N_{fft}	64
Espaciado entre subportadoras: Δf	$0.3125 \text{ MHz } (1/T_u)$
Espaciado entre subportadoras extremas	$16.25 \text{ MHz } (N_{st} \Delta f)$

Hay que decir que para las simulaciones hemos tratado tanto los datos como los pilotos de la misma forma, es decir, como un único flujo de símbolos 16-arios.

Por otro lado y tal como se puede observar en la tabla, los algoritmos FFT/IFFT usados son de 64 puntos, lo cual implica que tendríamos 64 subportadoras útiles, pero sólo se usan 52 de esas subportadoras (48 para datos y 4 para pilotos). En el siguiente esquema se muestra que subportadoras se transmiten al canal y cuales simplemente se modulan con una amplitud nula:



Para las simulaciones también se ha supuesto un canal ideal que simplemente añade un ruido complejo AWGN a la señal OFDM transmitida. Por tanto se han usado los siguientes parámetros de entrada para ejecutar la función sistema.m: $ret=0$, $n=1$,

$K=1$, $\alpha=0$. Dichos parámetros corresponden a un canal que no introduce retraso y no es dispersivo. Por tanto se han obviado los procesos de sincronización y ecualización.

2.2.6.2 IMPLEMENTACIÓN MATLAB DEL SISTEMA OFDM PARA HIPERLAN/2

A continuación vamos a ver el código Matlab de todas las funciones usadas en las simulaciones:

TRANSMISOR:

```
function [trama,sinf]=tbb(Nt,Ncp)

% [trama,sinf]=tbb(Nt,Ncp)
%
% Parámetros de entrada:
%     Nt: n° de super-símbolos OFDM transmitidos
%     Ncp: n° de muestras del CP
%
% Parámetros de salida:
%     trama: señal que ataca al canal; formada por
%           super-símbolos OFDM, siendo un supersímbolo
%           igual a CP+símbolo.
%     sinf: fuente de información aleatoria que
%           ataca al modulador OFDM. Se usa como salida
%           porque se utiliza en otra función para hallar
%           la BER.
%
% El módulo tbb simula el procesador digital OFDM en
% transmisión, realizando las siguientes operaciones:
%
%     - Crea la señal de información aleatoria
%     - Modula en 16-QAM
%     - Halla la IFFT-64
%     - Añade el CP
%
%
s64=zeros(1,64*Nt);
N=52; % n° de muestras de información (datos o pilotos)
%     del super-símbolo OFDM
% creación de la señal de información
sinf=round(rand(1,N*Nt)*15);
% modulación OFDM de la señal de información
for k=0:Nt-1
    s64(k*64+1)=0;
    s64(k*64+2:k*64+27)=sinf(k*52+27:k*52+52);
```

```

s64(k*64+28:k*64+38)=0;
s64(k*64+39:k*64+64)=sinf(k*52+1:k*52+26);
end
trama=tbb2(s64,Ncp);

```

```
function [spc,sem]=tbb1(se,Ncp)
```

```

%       [spc,sem]=tbb1(se,Ncp)
%
%   Parámetros de entrada:
%       se: bloque de señal de información
%           procedente de la fuente
%       Ncp: n° de muestras del CP
%
%   Parámetros de salida:
%       spc: señal modulada en OFDM + CP
%       sem: señal modulada 16-QAM
%
%   El módulo tbb1 toma un vector de
%   Nfft símbolos, los modula en
%   16QAM y antitransforma IFFT-64
%   añadiendo el CP

```

```

Nfft=64;
sem=zeros(1,Nfft);
spc=zeros(1,Nfft+Ncp);
i=sqrt(-1);
sem(1)=0+0*i;
for k=2:27
    [pr pi]=QAM(se(k));
    sem(k)=pr+i*pi;
end
sem(28:38)=0+i*0;
for k=39:64
    [pr pi]=QAM(se(k));
    sem(k)=pr+pi*i;
end
y=ifft(sem,Nfft);
spc=[y(Nfft-Ncp+1:Nfft) y];

```

```
function so=tbb2(si,Ncp)
```

```

%       so=tbb2(si,Ncp)
%
%   Parámetros de entrada:
%       si: señal de información procedente de
%           la fuente aleatoria
%       Ncp: n° de muestras del CP
%
%   Parámetros de salida:
%       so: señal modulada en OFDM + CP
%
%   El módulo tbb2 estructura la señal procedente

```

```

% de la fuente en bloques de tamaño Nfft,
% invocando después a tbb1. Posteriormente forma
% la señal de información de salida del
% procesador digital.

Nfft=64;
l=floor(length(si)/Nfft);
for k=0:l-1
    s=si(k*Nfft+1:(k+1)*Nfft);
    [spc,sem]=tbb1(s,Ncp);
    so(k*(Nfft+Ncp)+1:(k+1)*(Nfft+Ncp))=spc;
end

```

```

function [r,i]=QAM(sie)

%           [r,i]=QAM(sie)
%
% Parámetros de entrada:
%   sie: muestra de información
%
% Parámetros de salida:
%   r: parte real de la muestra modulada
%   i: parte imaginaria de la muestra modulada
%
% El módulo QAM realiza la modulación 16-QAM de
% la muestra dada, suponiendo esta un valor
% entero entre 0 y 15. Dicha modulación se basa
% en el código Gray.
if sie==0
    r=-3;
    i=-3;
    return
end
if sie==1
    r=-3;
    i=-1;
    return
end
if sie==2
    r=-3;
    i=3;
    return
end
if sie==3
    r=-3;
    i=1;
    return
end
if sie==4
    r=-1;
    i=-3;
    return
end
if sie==5
    r=-1;

```

```
        i=-1;
        return
    end
    if sie==6
        r=-1;
        i=3;
        return
    end
    if sie==7
        r=-1;
        i=1;
        return
    end
    if sie==8
        r=3;
        i=-3;
        return
    end
    if sie==9
        r=3;
        i=-1;
        return
    end
    if sie==10
        r=3;
        i=3;
        return
    end
    if sie==11
        r=3;
        i=1;
        return
    end
    if sie==12
        r=1;
        i=-3;
        return
    end
    if sie==13
        r=1;
        i=-1;
        return
    end
    if sie==14
        r=1;
        i=3;
        return
    end
    if sie==15
        r=1;
        i=1;
    end
```

```

function r=canal(s,ret,SNR,n,K,alfa)

%           r=canal(s,ret,SNR,n,K,alfa)
%
% Parámetros de entrada:
%   s: señal procedente del tx digital
%   ret: nº de muestras de retraso que
%        introduce el canal
%   SNR: relación señal a ruido en dB
%        tras los procesos de filtrado
%   n: longitud de la respuesta impulsiva del
%      canal dispersivo
%   K: amplitud de la respuesta impulsiva del
%      canal dispersivo
%   alfa: coef. de caída de la respuesta
%        impulsiva del canal dispersivo
% Parámetros de salida:
%   r: señal de salida del canal
%
% El módulo canal realiza los siguientes
% procesos:
%
%   - Introduce un retraso puro
%   - Filtra con  $h(n)=K*\exp(-alfa*n)$ 
%   - Introduce ruido AWGN complejo
%
% Formación del filtro dispersivo de canal
l=0:n-1;
h=K.*exp(-alfa*l);
% Normalización del filtro dispersivo de canal
Eh=sum(abs(h).^2);
h=h*(1/sqrt(Eh));
% Formación del retraso puro
delta=zeros(1,ret+1);
delta(ret+1)=1;
% Proceso de filtrado
y=conv(delta,s);
y=filter(h,1,y);
% Adición de ruido
randn('state',sum(100*clock))
nr=randn(1,length(y));
ni=randn(1,length(y));
i=sqrt(-1);
n=nr+i*ni;
n=n*(1/std(n));
ps=sum(abs(y(ret+1:length(y))).^2)...
    /length(y(ret+1:length(y)));
R=ps/(10^(SNR/10));
n=n*sqrt(R);
r=y+n;

```

```

function [sd,ber,scons]=rbb(si,origen,men,Ncp,Nt)

% [sd,ber,scons]=rbb(si,origen,men,Ncp,Nt)
%
% Parámetros de entrada:
%   si: señal proveniente del canal
%   origen: comienzo temporal de la señal de
%   información
%   men: mensaje originado en la fuente
%   aleatoria
%   Ncp: n° de muestras del CP
%   Nt: n° de super-símbolos OFDM
%   transmitidos
%
% Parámetros de salida:
%   sd: señal de información aleatoria
%   proveniente del demodulador OFDM.
%   ber: a la salida del receptor
%   scons: puntos de constelación recibidos
% El módulo rbb simula el procesador digital
% OFDM en recepción, realizando las siguientes
% operaciones:
%
%   - Paraleliza extrayendo el CP
%   - Halla la FFT-64
%   - Demodula 16-QAM
%

Nfft=64;
st=si(origen:length(si));
sfft=zeros(1,Nfft*Nt);
% Forma bloques de tamaño Nfft+Ncp y los entrega
% a rbb1
for k=0:Nt-1
    s=st(k*(Nfft+Ncp)+1:(k+1)*(Nfft+Ncp));
% rbb1 extrae el CP y halla la FFT
    sm=rbb1(s);
    sfft(k*Nfft+1:(k+1)*Nfft)=sm;
end
% Demodulación de la señal
s52=zeros(1,Nt*52);
for k=0:Nt-1
    s52(k*52+1:k*52+26)=sfft(k*64+39:k*64+64);
    s52(k*52+27:k*52+52)=sfft(k*64+2:k*64+27);
    scons=s52;
end
sd=dQAM(s52);
% Conversión a binario y evaluación de la ber
fo=dec2bin(sd,4);
fi=dec2bin(men,4);
[a,b]=find(fo~=fi);
ber=length(a)/(length(men)*4);

```

```

function ssm=rbb1(spc)

%       ssm=rbb1(spc)
% Parámetros de entrada:
%   spc: super-símbolo OFDM + CP
% Parámetros de salida:
%   ssm: señal que ataca al demodulador 16-QAM
%
% Módulo que toma un vector de Nfft+Ncp símbolos,
% extrae el CP y
% lo transforma FFT-64

Nfft=64;
Ncp=16;
y=spc(Ncp+1:Nfft+Ncp);
ssm=fft(y,Nfft);

```

```

function ss=dQAM(sd)

%       ss=dQAM(sd)
%
% Parámetros de entrada:
%   sd: vector de muestras de información
%       modulada
%
% Parámetros de salida:
%   ss: vector de muestras de información
%       demodulada
%
% El módulo dQAM realiza la demodulación 16-QAM
% de la muestra dada, suponiendo esta un punto
% de su constelación. Dicha demodulación se basa
% en dividir la constelación en una serie de
% regiones, evaluando en cual de ellas se
% encuentra la señal recibida

ss=zeros(1,length(sd));
for k=1:length(sd)
    r=real(sd(k));
    i=imag(sd(k));
    if r>=0
        if i>=0
            flag=0;
        else
            flag=1;
        end
    else
        if i>=0
            flag=2;
        else
            flag=3;
        end
    end
    switch flag
    case 0,
        if r<=2

```

```
        if i<=2
            s=15;
        else
            s=14;
        end
    else
        if i<=2
            s=11;
        else
            s=10;
        end
    end
case 1,
    if r<=2
        if i>=-2
            s=13;
        else
            s=12;
        end
    else
        if i>=-2
            s=9;
        else
            s=8;
        end
    end
case 2,
    if r<=-2
        if i<=2
            s=3;
        else
            s=2;
        end
    else
        if i<=2
            s=7;
        else
            s=6;
        end
    end
case 3,
    if r<=-2
        if i>=-2
            s=1;
        else
            s=0;
        end
    else
        if i>=-2
            s=5;
        else
            s=4;
        end
    end
end
ss(k)=s;
end
```

```
function ber=sistema (Nt,Ncp,ret,SNR,n,K,alfa)
%ber=sistema (Nt,Ncp,ret,SNR,n,K,alfa)
[so,men]=tbb (Nt,Ncp);
soc=canal (so,ret,SNR,n,K,alfa);
origen=ret+1;
[sd,ber,scons]=rbb (soc,origen,men,Ncp,Nt);
keyboard
```

```
function nada=dibcons (scons)
plot (real (scons), imag (scons), '.')
```

FUNCIÓN dep.m:

```
[espectro,f]=psd (real (so),1024,20e6);
espneg=flipud (espectro);
espneg=espneg (1:512);
fneg=-flipud (f);
fneg=fneg (1:512);
esptotal=[espneg;espectro];
ftotal=[fneg;f];
plot (ftotal,9+10*log10 (esptotal));
```

FUNCIÓN dep2.m:

```
%sobremuestreo por factor de 4
for k=0:1:length(so)-1
    soint(k*4+1)=so(k+1);
    soint(k*4+2:1:(k+1)*4)=0;
end
%filtrado LP con filtro FIR (fase lineal),
%BW=10 Mhz y Fsam=4*20e6 Hz
[hlp,n]=genfilfir(40,2,pi/4,pi/4);
sofil=conv(soint,hlp);
[espectro,f]=psd(real(sofil),1024,4*20e6);

espneg=flipud(espectro);
espneg=espneg(1:512);
fneg=-flipud(f);
fneg=fneg(1:512);
ftotal=[fneg;f];
%a continuación incluimos el efecto de un
%mantenedor de orden cero
esptotal=[espneg;espectro].*20e6.*...
    abs((sin(pi.*ftotal./20e6)./(pi.*ftotal)));

fhiper=[-40,-30,-20,-11,-9,9,11,20,30,40];
fnorm=ftotal./1e6;
specmask=[-40,-40,-28,-20,0,0,-20,-28,-40,-40];
plot(fhiper,specmask,fnorm,(15+10*log10(esptotal)));
```

FUNCIÓN genfilfir.m (usada en dep2.m):

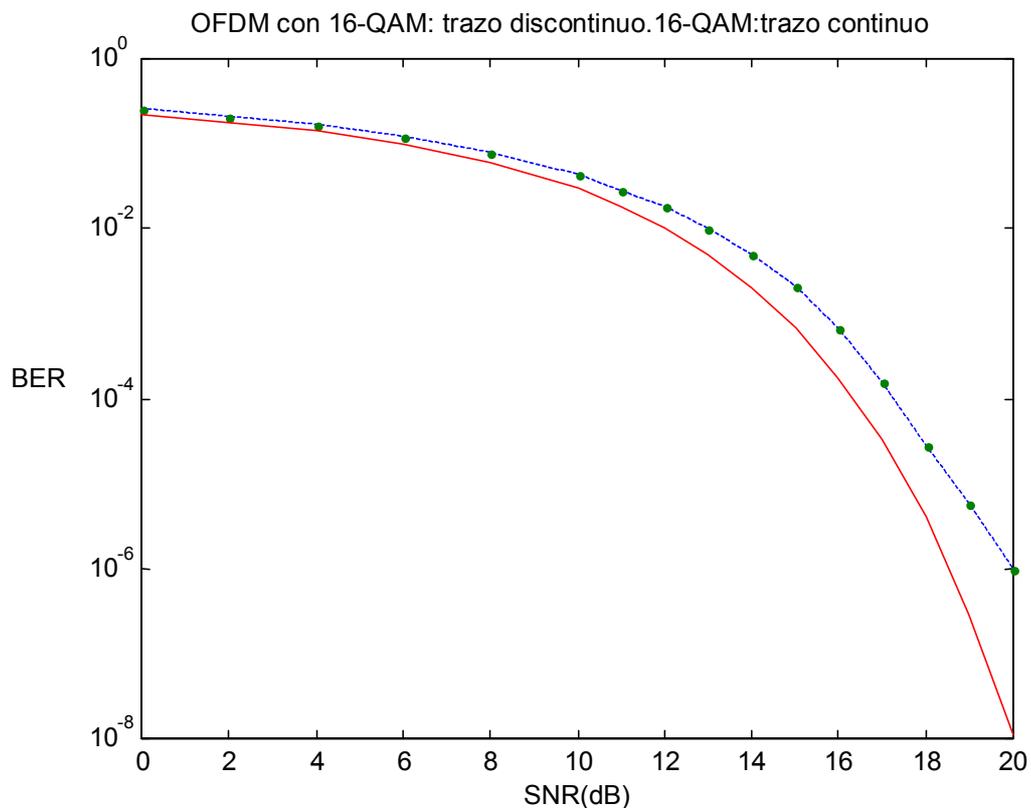
```
function [h,n]=genfilfir(N,vent,wp,ws)
%function [h,n]=genfilfir(N,vent,wp,ws)

%genera la respuesta impulsiva de un filtro FIR
%de longitud N paso de baja, con frecuencia de
%corte w0=(wp+ws)/2, ganancia=1 y fase lineal,
%usando una ventana: vent=0:rectangular,
%vent=1:hamming,vent=2:hanning,vent=3:triangular.
%Se considera una banda de transición que va
%entre wp y ws.

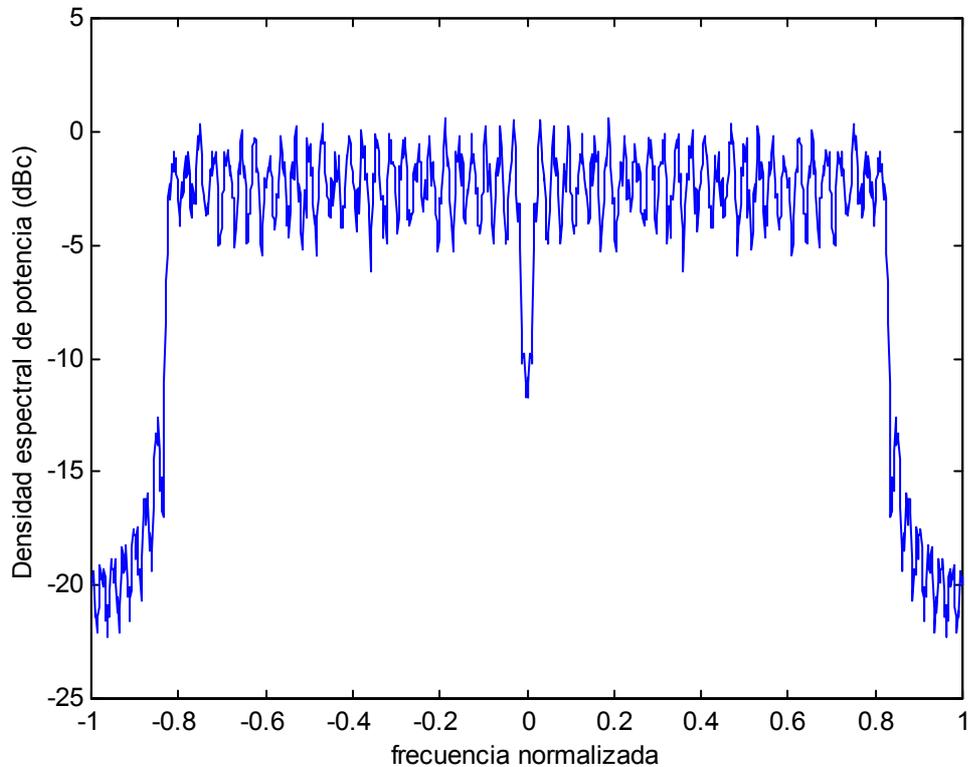
if vent==0
    ventana=boxcar(N);
end
if vent==1
    ventana=hamming(N);
end
if vent==2
    ventana=hanning(N);
end
if vent==3
    ventana=triang(N);
end
d=ws-wp;
n=(0:N-1);
M=(N-1)/2;
w0=(wp+ws)/2;
if wp==ws
    haux=(sin(w0*(n-M)))/(pi*(n-M));
else
    haux=(sin(w0*(n-M)).*sin(d*(n-M)/2))/(pi*(n-M).*...
        (d*(n-M)/2));
end
Npar=((N-1)/2)-floor(N/2);
if Npar==0
    haux(M+1)=w0/pi;
end
h=haux.*(ventana');
```

2.2.6.3 RESULTADOS DE LAS SIMULACIONES

La gráfica siguiente muestra la BER frente a la SNR que se obtuvo para los siguientes valores de los parámetros del sistema (corresponden a canal AWGN): $N_t=5000$ ($5000 \cdot 52 \cdot 4 = 1.040.000$ bits transmitidos), $N_{cp}=16$, $ret=0$, $n=1$, $k=1$, $\alpha=0$. Para SNR a partir de 22 dB se obtuvo BER nula para $N_t=5000$, por lo que se hicieron simulaciones con $N_t=50000$ (para poder estimar probabilidades de error binario de hasta $1E-6$). Pero para ese valor de N_t las simulaciones no se finalizaban debido a la falta de espacio en disco para albergar todas las variables intermedias. Por otro lado hemos añadido en la misma gráfica la BER correspondiente a un sistema 16-QAM simple.



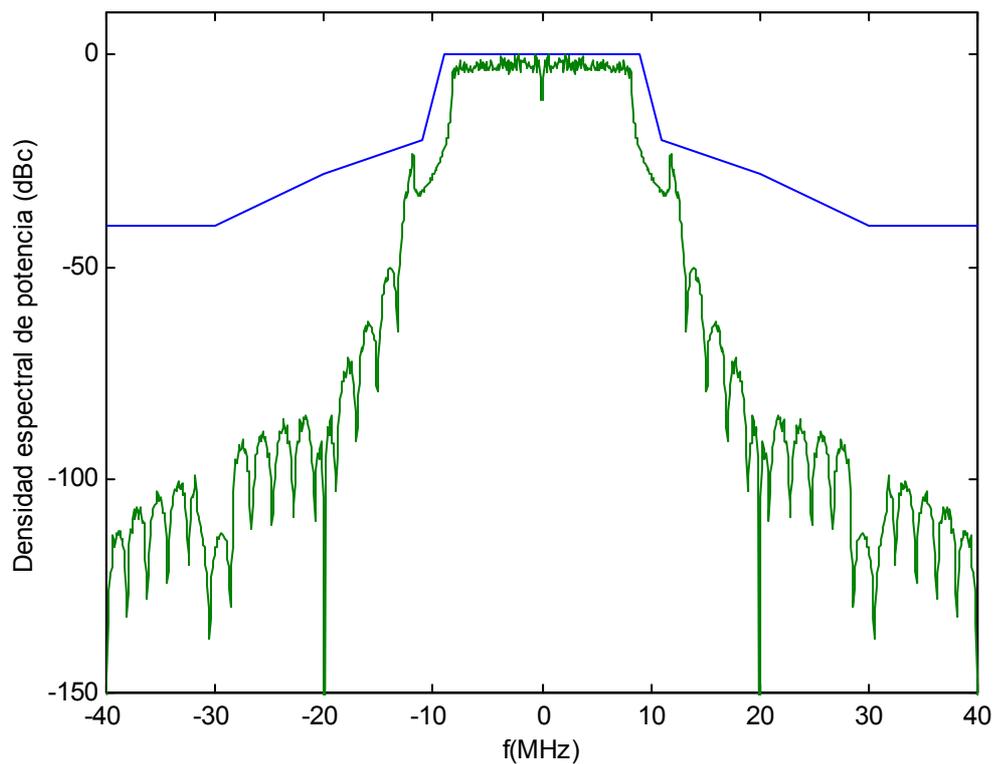
Por otro lado, obtuvimos (con la función **dep.m**) que la densidad espectral de potencia de la señal OFDM en banda base tenía la siguiente forma:



Debido a que estamos trabajando con una señal muestreada a 20MHz, el anterior espectro volvería a aparecer repetido centrado en los valores de frecuencia normalizada $\pm 2, \pm 4, \text{etc.}$ Para quedarnos con el espectro centrado en la frecuencia normalizada nula habría que usar un filtro digital paso de baja con $BW=10\text{MHz}$ situado a la salida del convertidor P/S del transmisor (ver figura en página 24), y además habría que considerar el efecto producido por un conversor digital analógico (hemos supuesto un mantenedor de orden cero). Todo lo anterior se modela en la función **dep2.m** y sus resultados se verán en la página siguiente.

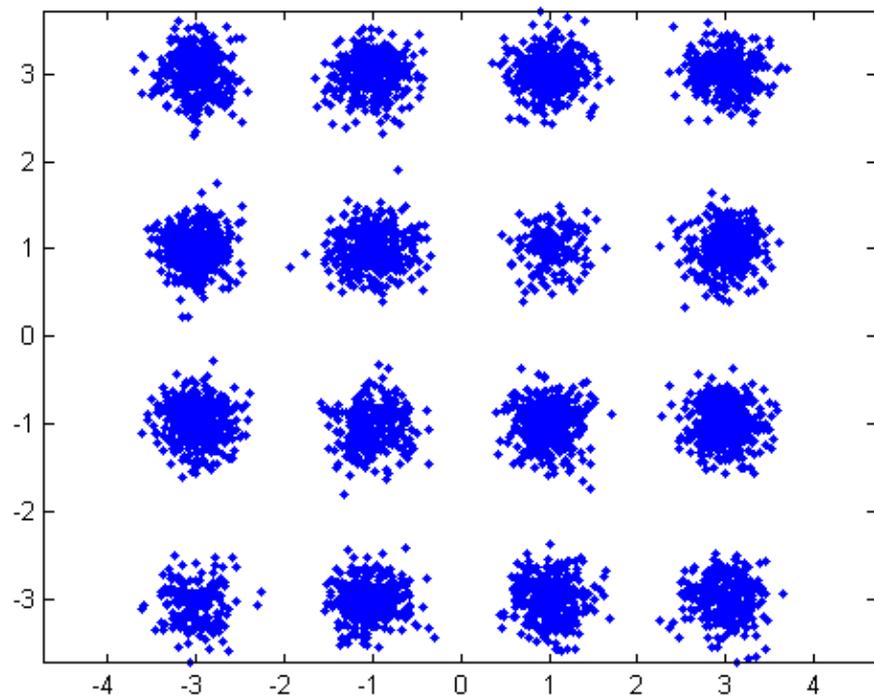
También vemos en la gráfica anterior que la frecuencia de continua no se transmite, debido a que el primer coeficiente en el cálculo de la IFFT se hizo nulo (ver esquema de página 49).

A continuación volvemos a representar la densidad espectral de potencia, pero con la frecuencia sin normalizar y junto a los niveles máximos permitidos por el estándar HIPERLAN/2 (tramos rectilíneos), pero ahora usando la función **dep2.m**:



Para mostrar el efecto sobre la dispersión de los puntos de la constelación recibidos al aumentar la potencia de ruido vemos las siguientes gráficas, donde SNR vale 19 dB en la primera y 15 en la segunda:

Constelación
para
SNR=19 dB



Constelación
para
SNR=15 dB

