

TEORÍA DE LA COMUNICACIÓN

PRÁCTICA 2: MODULACIÓN Y DETECCIÓN EN CANALES GAUSIANOS

CURSO ACADÉMICO 10/11

Objetivos

En esta práctica el alumno aprenderá

- los elementos de un sistema básico de comunicación digital en Matlab,
- los efectos de un canal gaussiano en las comunicaciones y
- la estructura del receptor óptimo y sus dos elementos más relevantes: el filtro adaptado y el decisor.

Normas y plazos de entrega

- La práctica debe realizarse individualmente.
- El alumno deberá responder y entregar a su profesor de prácticas el cuestionario que se encuentra al final de este documento.
- La fecha límite de entrega es una semana después de la sesión de laboratorio. Por cada día de retraso se descontará un 20 % de la nota.

1. Contenido Teórico

1.1. Transmisor

El transmisor de un sistema digital de comunicaciones tiene como función transformar los bits que se desean transmitir en señales en tiempo continuo con el formato adecuado para ser transmitidas con la menor distorsión posible por el canal de comunicaciones.

Habitualmente, el transmisor se divide en dos módulos funcionales: codificador y modulador.

El codificador realiza la asignación de bloques de m bits a un conjunto de $M = 2^m$ vectores de una constelación de dimensión N .

El modulador se encarga de convertir cada uno de los vectores de la constelación en una forma de onda $s_i(t)$. Esta forma de onda se obtiene mediante una combinación lineal de N señales que forman una base ortonormal de dimensión N , donde los coeficientes de la combinación son precisamente las N coordenadas del correspondiente vector de la constelación. Para el caso unidimensional ($N = 1$), la expresión de la señal resultante de la transmisión indefinida de símbolos es

$$s(t) = \sum_{n=0}^{L-1} A[n]\phi_0(t - nT),$$

donde:

- $\phi_0(t)$ es la señal (o pulso) de la base del modulador, normalizada en energía,
- $A[n]$ es el vector de la constelación asociado al símbolo n -ésimo,
- L es el número de símbolos a transmitir,
- y T es el periodo de símbolo.

La forma del pulso y los posibles valores de la constelación dependen del tipo de código utilizado. Aunque existen numerosos códigos M -arios (que agrupan los bits en conjuntos de tamaño $\log_2 M$ para obtener la amplitud del pulso), en esta práctica se van a tratar únicamente esquemas de modulación binarios, en concreto $A[n] \in \{0, 1\}$. El pulso se reducirá a un pulso rectangular y causal de duración T segundos cuya expresión matemática es:

$$\phi_0(t) = \frac{1}{\sqrt{T}} \Pi\left(\frac{t - T/2}{T}\right),$$

donde

$$\Pi(t) = \begin{cases} 1 & |t| < \frac{1}{2} \\ 0 & |t| > \frac{1}{2}. \end{cases}$$

1.2. Transmisión por el canal y recepción de las señales

La señal generada por el modulador se va a transmitir a través de un canal que añadirá ruido aditivo blanco gaussiano (AWGN) a la señal original. En la Figura 1 se muestra un esquema simple de un sistema de comunicaciones digitales (transmisor, canal y receptor). El ruido se suele caracterizar mediante su densidad espectral de potencia (DEP) que, al ser blanco, es constante e igual a $\frac{N_0}{2}$ para cualquier frecuencia.

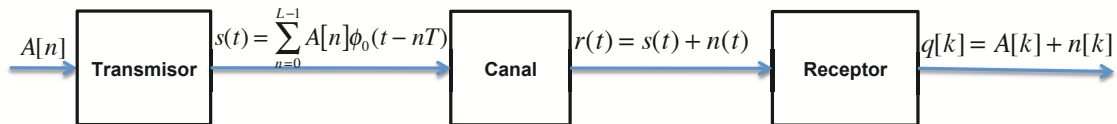


Figura 1: Diagrama de bloques de un sistema de comunicaciones digital

El receptor óptimo siempre va a estar formado por tres elementos,

1. Filtro Adaptado \Rightarrow Filtro adaptado a la forma de onda transmitida, $h(t) = \phi_0(T - t)$, y cuyo objetivo es maximizar la relación señal a ruido a su salida cada T segundos.
2. Muestreador \Rightarrow Se limita a quedarse con una muestra por símbolo a la salida del filtro adaptado.
3. Decisor \Rightarrow Decide el símbolo enviado. En el caso unidimensional ($N = 1$) y binario ($M = 2$), lo hace comparando la observación con un umbral.

En la Figura 2 podemos ver un esquema del filtro adaptado más el muestreador.

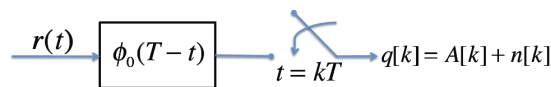


Figura 2: Filtro adaptado a $\phi_0(t)$ y muestreador.

2. Ejemplos de transmisión y recepción con Matlab

2.1. Características de la transmisión

Para generar una señal modulada por un pulso cuadrado, se proporciona la función:

```
x = wave_gen(bs, Rb)
```

Cuyos parámetros son:

- x es el código de línea generado

- `bs` es la secuencia binaria a transmitir obtenida previamente
- `Rb` es el régimen binario

Utilizando el código en Matlab proporcionado en el fichero `Ejercicio21.m` genere el código para una secuencia b de 10 bits equiprobables obtenidos de forma aleatoria y $R_b = 1000$ bits/s. Para representar el resultado obtenido puede utilizar el comando `waveplot(x)`.

2.2. Densidad Espectral de Potencia

La densidad espectral de potencia es uno de los parámetros clave a la hora de seleccionar un determinado código de línea para una cierta aplicación. Genere una secuencia de 1000 bits aleatorios e independientes, genere la señal modulada usando la función `x = wave_gen(bs, Rb)` y por último utilice la función `psd(x)`, que genera una gráfica con la densidad espectral de potencia de `x`.

2.3. Efectos del Canal: ruido gaussiano

El principal efecto negativo del canal es el ruido. En los ejercicios de este apartado se va a utilizar una función que simula el paso de la señal codificada por un canal con ruido gaussiano. El canal se encuentra definido por su ganancia en continua y su potencia de ruido (a la salida). Para su simulación se dispone de la función `y = channel(x, ganancia, pot_ruido)`, que devuelve la salida del canal, `y`, a partir de la entrada, `x`, y la potencia de ruido.

Genere la señal de salida de un codificador de pulso cuadrado con la función `x = wave_gen(bs, Rb)`, con una secuencia, `bs`, de 100 bits aleatorios e independientes y para un régimen binario de $R_b = 1000$ bits/s. Luego obtenga la salida de dicha secuencia al pasar por un canal lineal, ganancia unidad y sin ruido, usando la función `y = channel(x, 1, 0)`. Compare la entrada y la salida del canal (representándolas en la misma gráfica mediante el comando `subplot(rows , columns , num)`). Use la ayuda de Matlab para aprender a usar este comando.

A continuación, vaya aumentando la potencia de ruido de 0.02W en 0.02W empezando desde 0 W. Observe el efecto del incremento del ruido y anote el valor para el que no es posible distinguir la señal transmitida.

3. Receptor óptimo

3.1. Filtro Adaptado

Realice los ejercicios y responda a las preguntas que se formulan en el cuestionario del final.

Ejercicio 3.1

Modifique el código del fichero `Ejercicio31.m`, para generar una secuencia binaria, `bs`, de 10 bits equiprobables obtenidos de forma aleatoria y genere la señal modulada, `x`, con un pulso cuadrado.

Ejercicio 3.2

Genere un programa que utilizando la sentencia `x = wave_gen(bs, Rb)`, genere un vector `x` que contenga 70 bits aleatorios de un código de línea con forma de onda cuadrada y con régimen de símbolos $R_b = 1000$ bit/s. Después calcule el filtro adaptado llamando a la función `h = matched('pulso_cuadrado')`. Por último calcule la salida al filtro adaptado mediante la función `y = conv_a(x,h)`. Mediante el uso de los comandos `subplot` y `waveplot` visualice el código generado, `x`, y la salida del filtro adaptado, `y`.

Ejercicio 3.3

Añada al programa escrito para el Ejercicio 3.2. una llamada a la función `y = channel(entrada, ganancia, pot_ruido)`, ya vista anteriormente, y simule la salida de un canal con ganancia unidad y 0.5 W de potencia de ruido (la variable de entrada correspondería a la señal `x` generada con 10

bits de un código con un pulso cuadrado). Visualice la señal de salida del canal y responda a las preguntas del cuestionario. Utilice la sentencia `y = conv_a(y,h)` para calcular la salida del filtro adaptado cuando la entrada es la señal degradada por el canal. Visualice la señal a la salida del filtro adaptado y responda a las preguntas del cuestionario.

3.2. Receptor óptimo: Decisor

La segunda parte del receptor óptimo consiste en un decisor. Básicamente el decisor funciona quedándose con una única muestra por símbolo a la salida del filtro adaptado en aquel punto en que la SNR es máxima, y aplicando posteriormente un umbral a dichas muestras. En el caso binario y para $A[n] \in \{0, 1\}$, si la muestra supera el umbral se decide que el bit transmitido ha sido un 1, y en caso contrario se decide que ha sido un 0. En el caso M -ario existirán múltiples umbrales ($M-1$) que permitirán determinar cual de los M símbolos posibles es el que se ha enviado con mayor probabilidad (y el que se decide en consecuencia). En este apartado se explora la construcción de un detector óptimo para el caso binario.

Ejercicio 3.4

La función `ym = muestras(y_match, instant_mues)` devuelve un vector con una muestra por símbolo de la señal `y_match` (que es la salida del filtro adaptado), simulando la salida del muestreador. El instante en el que se toma la primera muestra (sincronismo de símbolo) está definido por el parámetro `instant_mues`. Añada una llamada a la función `muestras` para visualizar la salida del muestreador para un código con pulso cuadrado. Visualice la salida del muestreador y compárelo con la representación de los 10 símbolos transmitidos inicialmente.

Referencias

- *Comunicaciones Digitales*. A. Artés, F. Pérez González, J. Cid Sueiro, R. López Valcarce, C. Mosquera Nartallo y F. Pérez Cruz. Ed. Pearson Educación. 2007.
- *Communication Systems Engineering*. J.G. Proakis y M. Salehi. Prentice-Hall. 1994.

Apellidos:	Calificación	
Nombre:		T
NIA: Profesor de Prácticas		
Firma		

Sección 2: Ejemplos de transmisión y recepción con Matlab

Apartado 2.2. (Ejercicio opcional)

Obtenga una ecuación general para la densidad espectral de potencia (DEP) de la señal modulada generada en el Ejercicio 2.2, $S_s(j\omega)$, en función de la autocorrelación de la secuencia de símbolos, $R_A[k]$ (que se considera estacionaria) y la transformada de Fourier del pulso modulador, $\Phi_0(j\omega)$.

Apartado 2.3.

A la vista de la densidad espectral de potencia (DEP) obtenida, relacione las características de la señal modulada, $s(t)$, con el tipo de canal por el que se puede transmitir.

Sección 3: Receptor Óptimo

Apartado 3.1.

Escriba el código necesario para generar una secuencia binaria de 10 bits equiprobables obtenidos de forma aleatoria y genere la señal modulada con un pulso cuadrado en Matlab.

```
%-----bs-----  
bs = -----;  
Rb = 1000;  
  
%----- inicializacion de variables -----  
SAMPLING_CONSTANT = 10; % by default for this program  
fs = Rb * SAMPLING_CONSTANT;  
Ts = 1/fs;  
Tb = 1/Rb;  
  
L.bs = length(bs);  
  
%----- Modulacion -----  
pulso_cuadrado = -----;  
Lp = length(pulso_cuadrado);  
x_1 = zeros(1, Lp*length(bs) );  
for i=1:L.bs  
    x_1( 1, i*Lp-( Lp-1): i*Lp ) = -----;  
end
```

Apartado 3.2.

A la vista del resultado del Apartado 3.2, ¿cuánto vale la amplitud máxima de la señal a la salida del filtro adaptado?

Si después del filtro adaptado se toma únicamente una muestra por símbolo, ¿cuál es el instante óptimo de muestreo?

Apartado 3.3.

Visualice la señal de salida del canal, ¿puede distinguir qué símbolos se transmitieron? Razone su respuesta.

Visualice la señal de salida del filtro adaptado, ¿puede deducir ahora qué símbolos se han transmitido? Razone su respuesta.