Laboratorio de Comunicaciones

ETSII Telecomunicación
Grado en Ingeniería de Tecnologías de Telecomunicación
Universidad de Cantabria
Curso 2018/2019

Práctica 7. Receptor Digital Banda Base

Metodología

El alumno dispone de una sesión (2 horas) de laboratorio para realizar esta práctica, por lo que es imprescindible acudir al laboratorio habiéndola preparado previamente. La realización es individual. Se recomienda crear un fichero .m para cada uno de los apartados de la práctica y hacer uso de los comandos ***disp('texto'), disp('var'), pause, title, xlabel, ylabel, legend, clg ,clc,***... para documentar los resultados presentados en pantalla. Los distintos ficheros se guardarán en la carpeta ***X:/practica7***. El lunes 18 de diciembre se realizará un test de control para evaluar los conocimientos adquiridos por el alumno.

Objetivos

El objetivo de esta práctica es implementar un receptor de comunicaciones digitales banda base comprendiendo el funcionamiento y las características de cada uno de sus componentes. Se prestará especial atención al diseño del receptor óptimo, basado en el filtro adaptado y el decisor MAP, y a las prestaciones que proporciona en términos de BER (*bit error rate*). El alumno aprenderá a evaluar empíricamente la BER mediante repeticiones de Montecarlo y a calcular la relación E_b/N_0.

Implementación del Receptor Digital Banda Base

Al igual que ocurre con el transmisor, los receptores digitales son implementados totalmente (salvo las etapas de radiofrecuencia y amplificación) en el dominio discreto. La señal, $r(t)$, a la entrada del receptor es una versión ruidosa, distorsionada y atenuada de la enviada por el transmisor. La misión del filtro receptor es eliminar la mayor cantidad posible de ruido y proporcionar unos observables lo más distanciados posible. Tras el diezmado se obtienen los observables, una versión distorsionada y ruidosa de la secuencia de símbolos transmitida. El decisor decide qué símbolo fue transmitido en cada instante (por ejemplo, en el caso de una señal multinivel {-3 -1 1 3} el decisor presenta tres umbrales en 2, 0 y -2: si la muestra obtenida es mayor que 2 decide que se ha envidado un 3; si menor que 2 y mayor que 0 un 1, y así sucesivamente). Finalmente, el demapeador (o decodificador de símbolos) recupera los bits originales. La calidad de este sistema se mide en probabilidad de error, esto es, qué porcentaje de símbolos (o bits) transmitidos se recuperan erróneamente.

Se recomienda al alumno tener presente a lo largo de la realización de la práctica el esquema de la figura 1, con el fin de ser capaz de identificar en todo momento los distintos bloques y señales que intervienen y su correspondencia con las funciones y variables del código Matlab que se ejecutará.

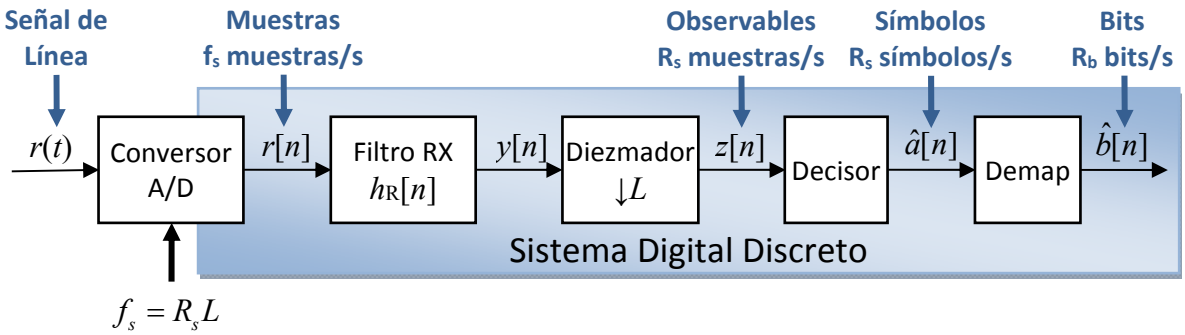

Figura 1. Receptor de comunicaciones digitales banda base.

a) Filtro receptor adaptado.

Utilizando una frecuencia de muestreo ***fs***=8000 muestras/s, genere una señal, ***x***, NRZ polar de 1000 bits/s (recuerde la relación entre ***fs***, R_s y ***L***) correspondiente a ***Nb***=10 bits. Dibuje la señal transmitida:

```
%% Transmisor
fs=8000;                        % Frecuencia de muestreo
L=8;                            % 8 muestras/símbolo
Nb=10;                          % Número de bits a transmitir
b=[1 0 1 randn(1,Nb-3)>0];      % Señal binaria aleatoria de Nb bits. Los tres primeros son 1 0 1
a=2*b-1;                        % Mapeador Polar: '0' ==> -1, '1' ==> +1
Ns=length(a);                   % Número de símbolos transmitidos
aL=[ ];aL(1:L:Ns*L)=a;          % introducimos L-1 ceros entre cada símbolo
ht=ones(1,L)/sqrt(L);           % Filtro transmisor NRZ normalizado (energía unitaria)
x=conv(aL,ht);                  % Convolucionamos con el filtro transmisor
t=(0:length(x)-1)/fs;           % Vector con los instantes de muestreo (para los plots)
figure(1);subplot(311);plot(t,x);grid;title('señal transmitida');xlabel('tiempo (s)')
```

En las siguientes instrucciones simularemos el efecto del canal y del ruido. Asumiendo que el canal es ideal (no distorsiona la señal de línea), simplemente atenuamos la señal y añadimos ruido blanco gaussiano (AWGN) de potencia[1] ***Pn***$=25$ μW.

```
%% Canal
Aten=10000;      % Atenuación del Canal 10000=40dB. Relación entre Potencia TX y Potencia RX
Pn=25e-6;        % Potencia de AWGN discreto
r=x/sqrt(Aten)+sqrt(Pn)*randn(size(x));
subplot(312);plot(t,r);grid;title('señal recibida');xlabel('tiempo (s)')
```

Filtre la señal recibida, ***r***, con el filtro adaptado. Si ha denominado ***ht*** a la respuesta al impulso del filtro transmisor (pulso transmisor), entonces ***ht(end:-1:1)*** es la respuesta al impulso del filtro adaptado. Observe detenidamente la señal resultante, ***y***:

```
%% Receptor
hr=ht(end:-1:1);          % Filtro receptor adaptado al transmisor (invertido temporalmente)
y=conv(r,hr);             % Convolución con el filtro adaptado
t=(0:length(y)-1)/fs;
subplot(313);plot(t,y);grid;title('señal tras filtro adaptado');xlabel('tiempo (s)');
```

Si lo prefiere, utilice la función ***stem(y)*** en lugar de ***plot(t,y)*** para observar las distintas señales y apreciar más claramente que se trata de señales discretas[2].

Puede probar a hacer la potencia de ruido cero (***Pn=0***) para observar mejor las señales. A partir de la observación de la señal tras el filtro receptor, ***y***, intente determinar el instante (muestra número n_0, es decir, instante t_0 segundos) en el que decidir respecto al primer símbolo. Los símbolos siguientes se decidirán a partir del observable ***y(n0+k*L)*** con K=0, 1, 2,…

En esta simulación, el valor de ***n0*** (muestra empleada para decidir el primer símbolo) es igual al instante en el que se emite el primer símbolo (muestra número 1) más el retardo combinado del filtro transmisor[3] y del filtro receptor (y del canal, si éste introdujera retardo).

---

[1] Al contrario de lo que ocurre con el AWGN en tiempo continuo, cuya potencia es infinita, el AWGN en tiempo discreto tiene potencia finita.

[2] En Matlab, la primera muestra de una señal o el primer elemento de un vector/matriz tiene índice 1 y así se representa gráficamente si se emplea ***stem(y)***. Si desea que en la gráfica la primera muestra se asocie al instante $n=0$ puede hacer ***stem(0:length(y)-1,y)***.

[3] Quizás pueda ser útil observar la respuesta al impulso (ejecutando ***stem(0:length(ht)-1,ht)***) y recordar el retardo de un filtro FIR con ese tipo de simetría que se estudió en Tratamiento de Señales.

Compruebe cuando hay ruido que, a la vista de la señal recibida ***r***, es muy difícil determinar en un instante concreto si el pulso que se está recibiendo es positivo ('1') o negativo ('0'). Sin embargo, si nos fijamos en el valor a la salida del filtro adaptado (señal ***y***) en las muestras n_0+kL (es decir, en los instantes t_0+kT), ¿cometeríamos más o menos errores que observando ***r***?, ¿resulta, por tanto, de utilidad el filtro adaptado? Ejecute varias veces el código para apreciar la variabilidad del ruido y de los bits generados.

Repita el ejercicio utilizando un filtro transmisor de tipo raíz cuadrada de coseno alzado con roll-off=0.5 (use ***ht=rcosine(1,L,'sqrt',0.5,6)***) y el filtro receptor adaptado correspondiente. Cuidado, ahora ***n0*** ha cambiado, pues los filtros transmisores y receptores son otros.

b) Decisor óptimo

Como se observa en la Figura 1, antes del decisor es necesario obtener los observables diezmando la señal tras el filtro receptor. Para ello, si sabemos que el instante óptimo para decidir sobre el primer símbolo de la NRZ polar se corresponde con la muestra ***n0***, entonces tendremos que tomar las muestras ***n0***, ***n0+L***, ***n0+2*L*** ,..., es decir,

```
nn0=(0:Ns-1)*L+n0;
z=y(nn0);            % observables
figure(2);stem(y,'b');grid;title('señal tras filtro adaptado');xlabel('n')
hold on;stem(nn0,z,'r');hold off;legend('y[n]','z[n]')
```

La señal ***z*** constituye lo que conocemos como observables, pues es de lo único de que dispone el decisor para decidir qué símbolos fueron transmitidos. En nuestro ejemplo, el decisor óptimo (MAP) para una señal polar equiprobable contaminada por ruido gaussiano es

$$z \underset{H_2}{\overset{H_1}{\gtrless}} 0 .$$

En Matlab, ejecutando la instrucción ***br=(z>0)*** implementamos en un solo paso el decisor y el demapeador polar. Compruebe que los bits recibidos, ***br***, coinciden con alta probabilidad con los transmitidos, ***b***. En ausencia de ruido (***Pn=0***) la coincidencia será siempre del 100%.

Pruebe con diversos valores de potencia de ruido, ***Pn***. Repita el ejercicio empleando filtros en raíz cuadrada de coseno alzado (cuidado, ahora el instante ***n0*** vuelve a ser el calculado para el sistema basado en filtros raíz cuadrada de coseno alzado).

Simulaciones Montecarlo

c) Cálculo empírico de la probabilidad de error de bit

La medida de calidad más importante de un sistema de comunicaciones digitales es la probabilidad de error de bit (BER: *bit error rate*). Para estimar la BER se recurre a realizar repeticiones de Montecarlo, es decir, se lleva a cabo la transmisión de una determinada (y normalmente elevada) cantidad de bits, N_b, y se cuenta el número de errores cometidos en el receptor, n_e, resultando

$$\mathrm{BER} \approx \frac{n_e}{N_b} .$$

En general, si se estudian sucesos con probabilidades del orden de p hay que realizar, al menos, $10/p$ repeticiones. Las probabilidades de error de muchos sistemas de comunicaciones comerciales oscilan entre 10^{-5} y 10^{-12}, por lo que habría que enviar entre un millón y 10 billones de bits para poder obtener resultados significativos. En estas prácticas, para que los programas de Matlab se realicen en un tiempo razonable, nos conformaremos con enviar 10000 bits y observar las probabilidades de error que se alcanzan con potencias de ruido más elevadas de lo habitual.

Implemente el transmisor (envíe 10000 bits) y receptor NRZ-Polar de los ejercicios anteriores (si elimina/comenta los plots del código, el programa se ejecutará más rápido). A partir de los bits recibidos, ***br***, obtenga la probabilidad de error de bit. Una forma muy sencilla de hacerlo es ejecutar ***mean(br~=b)***. Asegúrese de entender esta instrucción de Matlab.

d) E_b/N_0 y probabilidad de error teórica

Con el fin de determinar o comparar las prestaciones de los sistemas de comunicaciones es necesario evaluar la "calidad" que tiene la señal a la entrada del receptor. La relación señal a ruido ($\mathrm{SNR}=P_s/P_n$) puede no ser un parámetro demasiado significativo en comunicaciones digitales, por lo que se recurre a E_b/N_0, es decir, la relación a la entrada del receptor entre la energía media de que se dispone por cada bit que se recibe y la densidad espectral de potencia de AWGN que contamina la señal.

Para la señal recibida del ejercicio a) (NRZ-polar de Rb=1000 bps, amplitud $1/\sqrt{L}$ voltios y atenuación de 40dB) calcule teóricamente (sobre el papel) la potencia recibida, P_{RX}. ¿Cuál es la energía media de bit, E_b, disponible a la entrada del receptor?. ¿Qué relación existe entre ambas magnitudes?.

Puede calcular y comprobar con Matlab dichos valores. Recuerde que la potencia de la señal transmitida se puede calcular como **Ptx=mean(**x**.^2)**, por lo que sólo tendrá que incorporar el efecto de la atenuación para comprobar que el resultado teórico es correcto.

Respecto al ruido, para una frecuencia de muestreo dada, f_s Hz, la relación entre la densidad espectral de potencia, $N_0/2$ Vatios/Hz, y la potencia de ruido blanco discreto, P_n Vatios, viene dada por la expresión

$$P_n = N_0 \frac{f_S}{2},$$

puesto que el ancho de banda de nuestra simulación (de nuestro sistema digital discreto) es igual a la mitad de la frecuencia de muestreo (recuerde el teorema de muestreo).

A partir del ejercicio c), calcule con Matlab el valor de N_0 y de E_b y compruebe que la probabilidad de error empírica obtenida se corresponde con el valor teórico. Utilice la función ***qfunc*** de Matlab

Ejercicios adicionales

1. El diagrama de ojo sirve también para apreciar el valor de los observables y la influencia del ruido en los mismos. Represente en Matlab el diagrama de ojo de la señal tras el filtro receptor, ***y***, para diversos valores de la potencia de ruido, incluido el caso sin ruido. Note que en el instante de decisión (allí donde el ojo está más abierto) lo que se representa son los observables, ***z***. ¿Qué valor tienen los observables sin ruido?. Relacione los distintos diagramas de ojo con el histograma de los observables (***hist(z,40)***), que no es más que una estima de su función densidad de probabilidad.

2. Para este ejercicio se precisan dos alumnos. El primero de ellos implementará el transmisor NRZ-Polar de 1000 bits/s (genere únicamente ***Nb=20*** bits) y enviará la señal de línea a la tarjeta de sonido repetida un número suficientemente elevado de veces (por ejemplo ***envía(x,1000)***). El segundo alumno deberá adquirir la señal (***y=recibe(1,2*número_de_muestras_de_x)***) y procesarla con el receptor óptimo comprobando que se reciben los bits sin error. No olvide conectar el cable TX del alumno 1 con el RX del alumno 2 y determinar en cada ocasión el ***n0*** adecuado (sincronismo de símbolo y de trama). Es posible que las tarjetas de sonido introduzcan un offset de continua residual que, en su caso, deberá ser eliminado en el receptor.

3. Si el transmisor fuera del tipo 4PAM-NRZ (Práctica 5.e), implemente el receptor óptimo prestando especial cuidado en el decisor (ahora es multinivel) e incluyendo el correspondiente demapeador.