

Tratamiento y Transmisión de Señales

Ingenieros Electrónicos

EXAMEN SEGUNDA CONVOCATORIA SEPTIEMBRE 2007

TERCERA PARTE: PRÁCTICAS

Tiempo disponible 1 hora y media.

Instrucciones

Se deben realizar las modificaciones indicadas en la sección **Enunciado** partiendo de las prácticas realizadas durante la asignatura. El resultado del examen se debe entregar en formato electrónico utilizando Microsoft WORD respondiendo claramente a lo que se pide en el apartado **Resultados**. Cuando se pidan explicaciones de algún tipo sobre los resultados obtenidos, éstas se deben dar por escrito en el documento WORD haciendo referencia a la pregunta a la que se responde. El nombre de este documento debe hacer referencia al nombre y apellidos del alumno. Si se piden gráficas se deben copiar en el portapapeles desde Matlab y pegarlas en el documento WORD. Una vez terminado el documento (con las respuestas a las preguntas e incluyendo, en su caso, las gráficas correspondientes) se debe entregar al profesor. Incluir el código Matlab utilizado para la realización del examen.

Enunciado

Vamos a partir del apartado “Transmisión digital en banda base” de la cuarta práctica. Suponer que los bits a transmitir, **bitsTx**, se agrupan de dos en dos para generar una ristra de símbolos a transmitir. La tabla de asignación bits a símbolos es la siguiente:

Bits	Amplitud Símbolo
00	$-3A$
01	$-A$
10	A
11	$3A$

Se emplean pulsos con forma de onda rectangular. Suponer además que la tasa en bits por segundo es de $R_b=48000$, que el número de muestras por símbolo transmitido es de $K=20$, que el número de bits a transmitir es de $M=100000$ y que el valor de amplitud es $A=5.7$. Vamos a suponer un vector **N0** con $N=25$ valores escalados de forma logarítmica entre $N0_{min}=4e-5$ y $N0_{max}=0.01$.

Se pide hacer lo siguiente:

- Determinar la duración de cada bit T_b . Calcular asimismo la duración de cada muestra T_s y la tasa de muestreo f_s de la señal continua que se va a transmitir.
- Determinar el eje de tiempos t .
- Calcular los M bits a transmitir aleatoriamente y almacenarlos en el vector **bitsTx**.
- A partir del vector **bitsTx** determinar las amplitudes de los símbolos a transmitir según la tabla de asignación vista y guardarlo en el vector **simbolosTx** (que tendrá lógicamente $M/2$ elementos).
- Determinar la tasa en símbolos por segundo R y la duración de cada símbolo T .
- Determinar la señal a transmitir **senalTx**.
- Determinar la energía media por bit de la señal transmitida, E_b .

- A partir del vector N_0 determinar el vector de varianzas **varianza**.
- Determinar los tres umbrales necesarios en el decisor tras el filtro adaptado y guardarlos en el vector **umbrales**.
- Hacer lo siguiente para cada elemento del vector de varianzas de ruido, **varianza** (utilizar un bucle **for** con N iteraciones):
 - Generar un vector de ruido blanco Gaussiano con media cero y con varianza dada por una de las posibles varianzas (una para cada repetición del bucle **for**) y almacenarlo en el vector **ruido**. Este vector debe tener el mismo tamaño que **senalTx**.
 - Determinar el vector de señal recibida **senalRuido** sumando al vector **senalTx** el vector **ruido**.
 - Determinar la señal a la entrada del decisor (tras el filtro adaptado y el muestreador) y almacenarla en el vector **salida**.
 - Determinar los símbolos recibidos tras el decisor usando el vector **umbrales** (usar las amplitudes de la tabla para representar los símbolos recibidos) y almacenar el resultado en el vector **simbolosRx**.
 - Usando las asignaciones de la tabla determinar la señal binaria recibida **bitsRx**.
 - A partir de los símbolos recibidos **simbolosRx** generar la señal recibida **senalRx** de la misma forma que como se hizo con **senalTx** a partir de **simbolosTx**.
 - Calcular la probabilidad de error en símbolo contando los símbolos diferentes entre **simbolosTx** y **simbolosRx** y dividiendo el resultado entre el número total símbolos. Almacenar el resultado en el vector **Ps** (un elemento para cada iteración del bucle **for**).
 - Determinar la probabilidad de error en bit contando los bits diferentes entre **bitsTx** y **bitsRx** y dividiendo el resultado entre el número de bits M . Almacenar el resultado en el vector **Pe** (un elemento para cada iteración del bucle **for**).
 - Dibujar en la misma figura las primeras 400 muestras de la señal transmitida **senalTx**, de la señal ruidosa **senalRuido** y de la señal recibida **senalRx** en tres gráficas independientes en la misma figura.
- Después de terminar el bucle **for** los vectores **Ps** y **Pe** deben tener ambos las N probabilidades de error correspondientes a los N niveles de ruido. Determinar ahora el vector **EbN0** como el cociente entre la energía por bit E_b y la densidad espectral de ruido N_0 . Pasar el vector **EbN0** a dB definiendo el vector **EbN0log**. Finalmente, dibujar en una segunda figura la probabilidad de error en símbolo **Ps** y la probabilidad de error en bit **Pe** como función del vector **EbN0log**.

Resultados

Se pide lo siguiente a incluir en el documento WORD:

- ¿Cuál es el valor obtenido para T_b , T_s , f_s , R y T ? Indicar cómo se han calculado.
- ¿Qué valor y cómo se ha obtenido el vector **umbrales**?
- ¿Qué valor y cómo se ha obtenido la energía media por bit transmitido, E_b ?
- ¿Cómo se han determinado las amplitudes de los símbolos transmitidos **simbolosTx** a partir de **bitsTx**?
- Comentar cómo se ha implementado el filtro adaptado y el decisor para calcular el vector **simbolosRx**.
- ¿Cómo se han determinado los **bitsRx** a partir de los **simbolosRx**?
- Comentar cómo se ha calculado la probabilidad de error en símbolo, **Ps**.
- Presentar la figura correspondiente **senalTx**, **senalRuido** y **senalRx** para los casos de mayor y menor nivel de ruido (primera y última iteración del bucle **for**).
- Presentar la figura correspondiente a las probabilidades de error **Ps** y **Pe** como función del vector **EbN0log**. Indicar cuáles son las ventajas e inconvenientes de agrupar los bits de dos en dos frente a no hacerlo (caso práctica cuatro).