

Tratamiento y Transmisión de Señales

Ingenieros Electrónicos

EXAMEN CONVOCATORIA ORDINARIA JUNIO 2009

TERCERA PARTE: PRÁCTICAS

1 hora y media y 3 puntos.

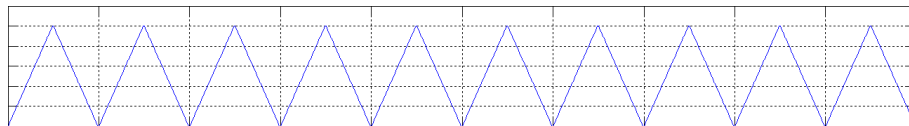
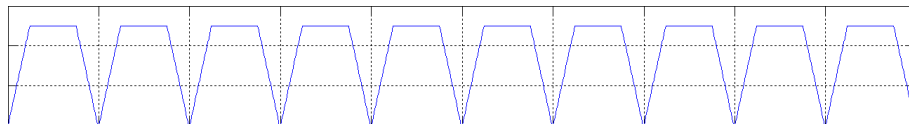
Instrucciones

Se deben realizar los pasos que se indican 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. 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 considerar una frecuencia de muestreo $f_s=50000$ Hz. Hacer un script de Matlab que haga lo siguiente:

- Generar dos ristra de bits `Bits1` y `Bits2` independientes con $N=1000$ bits cada una usando el comando de Matlab `rand`.
- Generar las señales `NRZpolar1` y `NRZpolar2` a partir de las ristra de bits generadas empleando código de línea NRZ polar con nivel $A=4.5$ y empleando $K=500$ muestras para cada bit. Utilizar los comandos de Matlab `repmat` y `reshape`.
- Definir la variable tiempo t (comenzando en $t=0$ seg.) y la variable frecuencia f para las señales que se vayan a definir de la misma longitud que las señales `NRZpolar1` y `NRZpolar2`.
- Definir las señales periódicas `c1` y `c2` que se pueden ver en las siguientes figuras



donde el periodo de dichas señales coincide con la duración de un bit, es decir, cada periodo tiene K muestras. Con respecto a la primera señal las duraciones de los tres tramos rectos es de $K/4$, $K/2$ y $K/4$, respectivamente. Para la segunda señal son $K/2$ y $K/2$, respectivamente. Emplear el comando de Matlab `repmat`. La longitud de las señales `c1` y `c2` debe ser la misma que la longitud de las señales `NRZpolar1` y `NRZpolar2`.

- Modificar la forma de las señales NRZ_{polar1} y NRZ_{polar2} multiplicándolas por las señales periódicas $c1$ y $c2$ para obtener las señales $m1$ y $m2$.
- Definir los coeficientes $h1$ de un filtro paso bajo con ancho de banda $BW1=500$ Hz y orden 128 usando el comando de Matlab `fir1`.
- Determinar las versiones filtradas, $m1f$ y $m2f$, de las señales $m1$ y $m2$ usando el filtro $h1$ y el comando de Matlab `filter`.
- Usando una portadora con amplitud $Ac1=1$ y frecuencia $fc1=1000$ Hz, generar una señal DSB usando como señal moduladora $m2f$. La señal DSB resultante sumarla a la señal $m1f$ para obtener la señal $s1$.
- Considerando ahora la señal $s1$ como una señal moduladora con ancho de banda $BW2=1500$, se modula en frecuencia utilizando una segunda portadora con amplitud $Ac2=5$ y frecuencia $fc2=12500$ Hz para dar lugar a la señal $s2$. Se sabe que el modulador FM empleado tiene una sensibilidad en frecuencia $kf=300$ V/Hz.
- Supongamos ahora que la señal FM $s2$ se transmite por un canal ruidoso (ruido Gaussiano, aditivo, blanco, con media cero y densidad espectral $N0/2$) cuyo nivel de ruido va cambiando. Generar un vector de $M=10$ valores de densidad espectral de ruido $N0$ espaciados logarítmicamente entre los valores $N0_{min}=1e-4$ W/Hz y $N0_{max}=1e-2$ W/Hz, usando el comando de Matlab `logspace`.
- Hacer lo siguiente para cada nivel de ruido (dentro de un bucle `for`):
 - Generar un vector de ruido blanco w correspondiente al nivel de ruido $N0$ correspondiente.
 - Determinar la señal FM ruidosa $s2w$.
 - Definir los coeficientes del filtro hIF de frecuencia intermedia para la señal FM. Considerar un filtro de orden 128 y que además para el ancho de banda de transmisión de la señal FM se puede considerar el valor $BT=13000$ Hz. Determinar entonces la señal $s2n$ a la salida del filtro así definido.
 - El demodulador está formado por un limitador, seguido de un discriminador de frecuencias y el filtro de postdetección. Determinar la señal a la salida del limitador $s2n1$. El limitador se puede implementar mediante un saturador seguido de un filtro paso banda. Utilizar la función de Matlab `sign` como saturador y el filtro paso banda hIF ya definido.
 - Para la implementación del discriminador en frecuencia definir los filtros $H1$ y $H2$ en el dominio de la frecuencia correspondientes a los circuitos pendiente complementarios utilizando el parámetro $a=6e-5$. Igualmente, será necesario definir los coeficientes de un filtro $h2$ paso bajo con frecuencia de corte $BW2$ y orden 128. Al final se dispondrá de la señal y a la salida del receptor FM.
 - Utilizando el filtro paso bajo $h1$ ya definido extraer la primera señal digital $m1r$ de la señal y .
 - Definir ahora el filtro paso banda $h3$ de orden 128 necesario para extraer la señal DSB $s1r$ de la señal y .
 - Demodular la señal DSB $s1r$ para extraer la segunda señal digital $m2r$. Utilizar para ello un detector coherente con portadora local con amplitud $Ac3=2$ y frecuencia $fc1$ y como filtro paso bajo el ya definido por los coeficientes $h1$.
 - Si para cada Bit se utiliza la muestra número $k=K/2$ y se la compara con el umbral 0, determinar las dos ristas de bits recibidos: $Bits1Rx$ y $Bits2Rx$. Utilizar el comando de Matlab `reshape`.
 - Comparando los bits transmitidos $Bits1$ y $Bits2$ y los recibidos $Bits1Rx$ y $Bits2Rx$, determinar las probabilidades de error $Pe1$ y $Pe2$, respectivamente.

Al final del bucle `for` se deberá disponer de los vectores $Pe1$ y $Pe2$ de M componentes cada uno con las probabilidades de error correspondientes a cada nivel $N0$ de ruido del canal.

Resultados

Se pide lo siguiente a incluir en el documento WORD:

- Dibujar en una misma figura usando el comando `subplot` los 10 primeros bits de las señales `m1` y `m2`, escalando adecuadamente los ejes para su correcta visualización.
- Dibujar en una misma figura usando el comando `subplot` el módulo del espectro de las señales `m1` y `m2`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/20, fs/20]$.
- Dibujar en una misma figura usando el comando `subplot` los 10 primeros bits de las señales `m1f` y `m2f`, escalando adecuadamente los ejes para su correcta visualización.
- Dibujar en una misma figura usando el comando `subplot` el módulo del espectro de las señales `m1f` y `m2f`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/20, fs/20]$.
- Dibujar en una figura el módulo del espectro de la señal `s1`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/20, fs/20]$.
- Dibujar en una misma figura utilizando dos colores la respuesta en frecuencia de los circuitos pendiente `H1` y `H2`. Dibujar el intervalo frecuencial $[-fs/2, fs/2]$.
- Para el nivel de ruido `N0` menor dibujar en una misma figura usando el comando `subplot` el módulo del espectro de las señales `s2w`, `s2n` y `s2nl`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/2, fs/2]$.
- Para el nivel de ruido `N0` menor dibujar el módulo del espectro de la señal `y`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/20, fs/20]$.
- Para el nivel de ruido `N0` menor dibujar en una misma figura usando el comando `subplot` los 10 primeros bits de las señales `m1r` y `m2r`, escalando adecuadamente los ejes para su correcta visualización.
- Para el nivel de ruido `N0` menor dibujar en una misma figura usando el comando `subplot` el módulo del espectro de las señales `m1r` y `m2r`, escalando adecuadamente los ejes para su correcta visualización. Dibujar el intervalo frecuencial $[-fs/20, fs/20]$.
- Finalmente dibujar en una misma figura utilizando dos colores distintos las probabilidades de error `Pe1` y `Pe2`, como función del nivel de ruido `N0` del canal. Dibujar ambos ejes en escala logarítmica usando el comando de Matlab `loglog`.