

MODULACIÓN AM (Propiedades de la TF)

APELLIDOS:	NOMBRE:
APELLIDOS:	NOMBRE:

Fecha de entrega: hasta el 12 de noviembre

Es esta práctica vamos a tratar de entender mejor el concepto de Transformada de Fourier y ver como puede ayudarnos a entender que es lo que está pasando en un sistema mejor que la descripción temporal. También usaremos algunas de sus propiedades, todo ello aplicado a un problema real de enorme importancia: la **modulación/demodulación** de señales.

Introducción

La modulación de señales es un concepto fundamental en comunicaciones, y consiste en usar una señal base (que lleva la información que deseamos transmitir) para modificar las propiedades de una señal portadora (típicamente una frecuencia pura de frecuencia mucho más alta). En el caso más sencillo, la señal base modifica la *amplitud* de la portadora, y tenemos un sistema AM (amplitud modulada). En otros métodos se usa la señal original para modificar la frecuencia (FM) o fase (PM) de la portadora.

Consideremos una señal de voz $v(t)$ cuya transformada de Fourier es $V(\omega)$. Veamos que sucede si cogemos una portadora ideal, una exponencial compleja de frecuencia ω_c y amplitud 1, $e^{-i\omega_c t}$, y hacemos que nuestra señal de voz determine la amplitud de dicha portadora:

$$v(t) e^{+i\omega_c t} \Rightarrow V(\omega - \omega_c)$$

Por propiedades de la TF vemos que la TF de la nueva señal mantiene su forma $V(\omega)$ sólo que ahora se encuentra centrada en ω_c , en vez de en 0. Al mantener la forma está claro que podremos recuperar la señal original simplemente volviendo a multiplicar por $e^{-i\omega_c t}$ (esto es, demodulando).

En el mundo real las cosas son un poco más complicadas. Nosotros debemos trabajar con señales reales, por lo que no podemos multiplicar por exponenciales complejas al modular o demodular. En vez de eso usaremos sinusoides reales, por ejemplo $\cos(\omega_c t)$, para modular nuestras señales.

Simulación de TF con MatLab

Usaremos *Matlab* para visualizar los resultados. Notad que aquí tenemos que hacer trampas de alguna manera. La modulación es un fenómeno esencialmente continuo y *Matlab* sólo trabaja con un discreto. Lo que haremos será utilizar señales con un muestreo muy fino (muestreadas 100000 veces por segundo, técnicamente usando una frecuencia de muestreo, **fs**, de 100 KHz) por lo que a efectos prácticos vosotros desde “fuera” os parecerá ver un continuo.

Cargemos una señal de voz (2.5 segundos) para empezar a trabajar y escuchemosla:

```
[v1 fs]=wavread('voz1'); sound(v1,fs);
```

Echemos un vistazo a su espectro de potencias (el módulo al cuadrado de TF), mediante la función:

```
ver_tf(v1,fs);
```

Notad que al estar trabajando por debajo con un discreto debemos darle la frecuencia de muestreo, lo que le sirve al programa para saber en que instante de tiempo sucedieron las distintas muestras y darnos bien las unidades de frecuencia y tiempo.

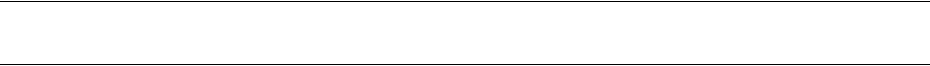
La función **ver_tf** tiene dos argumentos adicionales. El tercer argumento indica el color de la gráfica y el cuarto indica si se pintan todas las frecuencias (**'todo'**, por defecto) o sólo las positivas (**'semi'**). Esto es debido a que TF de una señal real $x(t)$ cumple que $X(-\omega) = X^*(\omega)$ por lo que su módulo será par y es superfluo mostrar ambos ejes.

Otro aspecto a resaltar de la gráfica anterior es que el eje de ordenadas se representa en una escala logarítmica en decibelios (dB), relacionados con la amplitud de la forma siguiente:

$$dB = 10 \log_{10} |X(\omega)|^2 = 20 \log_{10} |X(\omega)|.$$

Un valor de 1 en la amplitud de $|X(\omega)|$ se convierte en 0 dB. Valores menores de 1 se traducen en dB negativos (dB de atenuación). Por ejemplo, una caída de 20 dB, supone una atenuación de la amplitud en un factor 10, 40 dB suponen una atenuación en un factor 100 (casi total para muchas situaciones).

¿Para que frecuencia la atenuación del espectro alcanza los 30 dB? ¿y los 40 dB? Podeis hacerlo a ojo sobre la gráfica o usar el comando **ginput(n)** que os permite pinchar en n puntos de la gráfica y os devuelve las coordenadas sobre los ejes.



Al rango de frecuencias donde se concentra la energía de una señal se le denomina ancho de banda o banda base. Cargar otra señal de voz, **voz2.wav**, y correr **ver_tf** sobre ella, haciendo un **hold on** previo para visualizar ambas a la vez.

```
[v2,fs]=wavread('voz2'); hold on; ver_tf(v2,fs,'r');
```

¿Difiere mucho de la anterior? ¿Que valor le daríais al ancho de banda de una señal de voz, suponiendo que los contenidos por debajo de -40dB son despreciables?



Para comprobar que las frecuencias superiores contribuyen muy poco a la señal de voz vamos a hacer pasar nuestra señal por un sistema *pasobajo* que elimina aquellas frecuencias por encima de una cierta frecuencia de corte, que en nuestro caso es de 4 KHz.

Aplicamos dicho filtro a nuestra señal y observamos su TF, comparandola con la original:

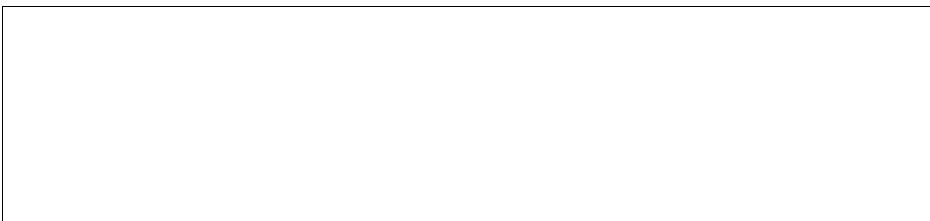
```
ver_tf(v1,fs); hold on;
pb1=paso_bajo(v1);
ver_tf(pb1,fs,'g');hold off
```

Ahora vemos que no hay prácticamente nada de energía por encima de unos 4 KHz (la frecuencia de corte del filtro). La función `show2(x,y,fs)` abre una ventana y muestra dos ejes donde podemos comparar dos señales `x` e `y` (muestreadas con frecuencia `fs`). Usarla para ver las diferencias entre la señal original y su versión pasobajo en el dominio temporal:

```
show2(v1,pb1,fs);.
```

Con audio la única comparación importante es como suena. Usando `sound(pb1,fs)`, podemos comprobar que es perfectamente inteligible y no se ha perdido prácticamente nada.

La función `paso_bajo` anterior admite un parámetro opcional que nos permite imponer la frecuencia de corte (en KHz) usada en el filtrado pasobajo. Hacer `paso_bajo(v1,F)` usando como frecuencia de corte `F=3, 2, 1, ... KHz`. ¿Cuándo empezáis a notar una distorsión significativa? Comparar la señal original y la filtrada para `F=1 KHz` usando `show2`. Comentar los resultados.



Veamos que pasa para otro tipo de señales. `musica.wav` contiene una señal de audio, pero con música en vez de voz, mientras que en `bits.wav` tenemos una simulación de una comunicación digital a 4000 bits/sec. Consiste en 10000 bits aleatorios ($2.5 \text{ sec} \times 4000 \text{ bits/sec}$) cada uno de ellos representado por un voltaje positivo (1) o negativo (0) con una duración de $1/4000 \text{ sec} = 0.25 \text{ msec}$. Plotear sus espectros superpuestos con los de una señal de voz:

```
[v1,fs]=wavread('voz1'); ver_tf(v1,fs,'b','semi'); hold on;
[music,fs]=wavread('musica'); ver_tf(music,fs,'r','semi');
[bits,fs]=wavread('bits'); ver_tf(bits,fs,'g','semi'); hold off;
```

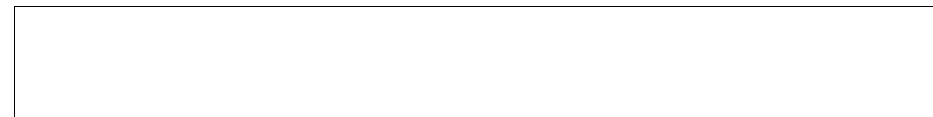
Dibujar los espectros resultantes en cada caso. Con el criterio de una caída de 35dB ¿qué ancho de banda asignaríamos a cada señal (voz, música, bits)? Indicarlos sobre la gráfica.



Haced pasar la señal `bits` por el filtro paso bajo y visualizar las diferencias usando `show2`. ¿Se ha degradado apreciablemente la señal? ¿Son reconocibles los bits?



Haced pasar la señal `music` por el filtro paso bajo (defecto, 4 KHz). ¿Se nota alguna diferencia con el original? ¿Es aceptable la calidad?



Simulación de Modulación con Matlab

AM Normal: Estudiemos ahora el efecto de la modulación AM usando un $\cos(\omega_c t)$ como portadora. En su versión más simple simplemente generamos una señal correspondiente a un coseno de la frecuencia escogida y la multiplicamos por la original, usando la función `modular(x,fc,fs)`, donde `x` es la señal a modular, `fs` es la frecuencia de muestreo y `fc` es la frecuencia portadora (en KHz). El código siguiente (contenido en la función `modular()`, lleva a cabo los pasos anteriores:

```
function y=modular(x,fc,fs) % Código de la función modular (no teclear)
L=length(x); t=[0:L-1]'/fs; % Generamos vector de tiempos
fc=1000*fc; % De KHz a Hz
wc=2*pi*fc; % Pasamos de Hz(ciclos/sec) a rad/sec
carrier=cos(wc*t); % Portadora: coseno de frecuencia fc Hz
y=x.*carrier; % Multiplicamos entrada y portadora.
```

Como ejemplo modulemos la señal de voz contenida en `voz1.wav` con 15 KHz:

```
[x fs]=wavread('voz1'); y=modular(x,15,fs); show2(x,y,fs);
```

Intentemos escuchar la señal modulada: `sound(y,f);`.

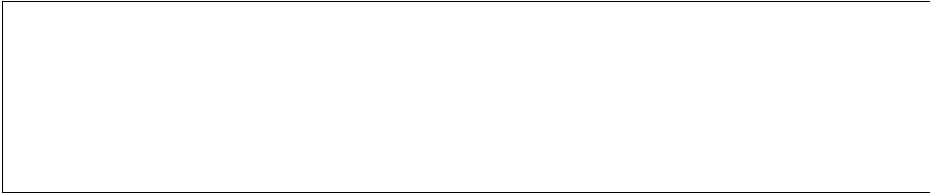
¿Cuál es el espectro esperado de la señal modulada? Considerando que

$$\left\{ \begin{array}{lcl} x(t) & \Rightarrow & X(\omega) \\ \cos(\omega_c t) & \Rightarrow & \pi(\delta(\omega + \omega_c) + \delta(\omega - \omega_c)) \end{array} \right\}$$

y recordando que multiplicación en un dominio se traduce en convolución en el otro tendremos que:

$$x(t) \cos(\omega_c t) \Rightarrow \frac{1}{2\pi} (X(\omega) * \pi(\delta(\omega + \omega_c) + \delta(\omega - \omega_c))) = \frac{1}{2} (X(\omega + \omega_c) + X(\omega - \omega_c))$$

Es decir, cuando usamos una modulación con un coseno obtenemos no una copia de $X(\omega)$ sino dos, una centrada en ω_c y la otra en $-\omega_c$. Usar `ver_tf(y,fs)`; y dibujad el espectro resultante:



Antes para demodular bastaba con volver a multiplicar por la portadora ideal $e^{i\omega_c t}$. En nuestro caso, si nos limitamos a la parte real, volvemos a tener que usar el $\cos(\omega_c t)$ ¿Qué pasará si ahora intentamos *demodular* la señal volviendo a multiplicar por la misma portadora? Recordando que al multiplicar por un coseno cada “bulto” del espectro se desdobla en dos, pensad como debería ser dicho espectro y bosquejarlo:



Comprobemoslo con *Matlab* : `xx=demodular(y,15,fs); ver_tf(xx,fs);`

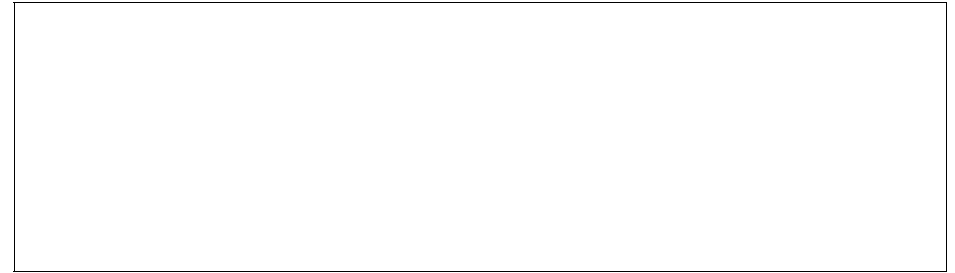
¿Se recupera el espectro original? ¿Qué falta/sobra para recuperar la señal original?



Escuchar la señal resultante. ¿Se entiende? ¿Tiene algún ruido adicional? ¿Qué puede estar pasando?



¿Cuál de las funciones que hemos visto se podría usar para quedarnos únicamente con la copia del espectro centrada en el origen?. Aplicadla y echad un vistazo al espectro resultante. Dibujad un diagrama que explique esquemáticamente en el dominio de frecuencias todo el proceso de modular y demodular una señal en AM.



Modulación AM con portadora:

La forma vista anteriormente es muy sencilla pero tiene algunos problemas prácticos que no hemos sido capaz de ver en nuestra simulación. Para demodular la señal el receptor debe multiplicar la señal recibida por un coseno de exactamente la misma frecuencia que el emisor. No sólo eso, sino que debe estar en fase (empezar el ciclo exactamente al mismo tiempo). Desajustes en la frecuencia o fase del coseno en la etapa de recepción degradan la señal recibida.

En la práctica, para conseguir dicha sincronización el receptor debe tener una circuitería adicional, además de un oscilador que genere la onda sinusoidal, etc. Todo ello encarecerá el producto (aunque en la actualidad la inmensa mayoría de los receptores son de este último tipo).

Hay otra variante de modulación AM (la que se usaba en los inicios de la radio) que permite una recepción extraordinariamente simple (las antiguas radios de galena o de cristal). Consiste en lo siguiente:

$$x_m(t) = (1 + x(t)) \cos(\omega_c t)$$

Lo importante aquí es que $(1+x(t))$ no llegue a hacerse nunca negativa (si ese fuese el caso meteríamos una constante $k < 1$ multiplicando a $x(t)$ para evitarlo). Notad que $x_m(t) = x(t) \cos(\omega_c t) + \cos(\omega_c t)$. El primer término es igual que la modulación anterior, luego la única diferencia es que adicionalmente mandamos el $\cos(\omega_c t)$. Por eso este tipo de modulación se denomina full AM o AM con transmisión de la portadora. Comparemos ambos tipos de modulación en el dominio de frecuencias y temporal:

```
[v fs]=wavread('voz1');  
vm1=modular(v,15,fs);   vm2=modular(v,15,fs,'amtc');  
figure(1); ver_tf(vm1,fs); hold on; ver_tf(vm2,fs,'r'); hold off;  
figure(2); show2(vm1,vm2);
```

¿Que diferencia podemos apreciar en el dominio de frecuencias? Escribir la expresión teórica de la TF de la nueva señal modulada. ¿Explica la diferencia encontrada?

Veamos el proceso de demodulación de una señal modulada AM con portadora.

1. Los voltajes negativos se ponen a cero usando un diodo. Usando *Matlab* :

```
vm=modular(v,15,fs,'amtc'); r1=vm; r1(r1<0)=0; show2(vm,r1,fs);
```

Vemos que siguiendo con el dedo el perfil de la señal resultante tenemos nuestra señal original.

2. Para eliminar las caídas a cero de nuestra señal, podemos aplicar el filtro pasabajo de antes (de nuevo fácilmente implementable con un condensador y una resistencia):

```
r2=paso_bajo(r1); show2(r1,r2,fs);
```

3. La señal casi se ha recuperado, lo único es que la señal original tenía media cero y esta no. Pero eso es fácil de arreglar simplemente haciendo: $r2=r2-\text{mean}(r2)$; . Electrónicamente, un condensador nos arregla el problema, pues deja “pasar” los cambios de voltaje mientras que bloquea una DC (la media).

La comparación visual o auditiva nos hará ver como realmente hemos recuperado (sin sincronizaciones ni la necesidad de un oscilador) la forma de la señal original: `show2(v1,r2,fs); sound(r2,fs)`; *Ventajas de AM con portadora:* Al mandar la portadora simplificamos enormemente el proceso de recepción, pudiéndose usar receptores muy simples (y baratos). La Onda Media usa Full AM.

Inconvenientes de AM con portadora: Muy ineficiente. La mayor parte de la energía se gasta en transmitir la portadora que no lleva ninguna información (pensad en el caso de que $x(t)$ sea nula: AM normal no estaría mandando nada, $x_m(t) = 0$. AM-TC estaría mandando $\cos(\omega_c t)$). Por eso las emisoras Onda Media suelen tener un alcance local. Si tenemos restricciones en la potencia a emplear la transmisión AM normal (sin portadora) es mucho más eficiente.

Utilidad de la modulación

Ya sabemos como modular y demodular una señal en su versión AM más sencilla. ¿Para que nos puede valer modular una señal? Dos aspectos fundamentales:

- Una **transmisión más eficiente**. Eligiendo el valor de ω_c adecuadamente podemos hacer que la señal enviada entre en una ventana de frecuencias donde la transmisión sea más efectiva o simplemente donde sea posible.

Veamos un ejemplo con *Matlab* . La función `y=canal(x)` simula un medio o canal que sólo transmite aquellas frecuencias comprendidas entre 10 y 40 KHz. La función `ver_canal` nos pinta (en verde) la función de transferencia de nuestro canal (recordad la escala logarítmica, 0 dB indican una perfecta transmisión, dB negativos indican una atenuación). Si cogemos la señal original y la hacemos pasar por el canal, visualizando la TF resultante:

```
[v fs]=wavread('voz1'); out=canal(v); sound(out,fs);  
ver_canal; hold on; ver_tf(v,fs); hold off;
```

Vemos que no tenemos nada a la salida, ya nuestro canal no deja pasar las frecuencias (0-5 KHz) presentes en la señal de voz. La modulación nos ayuda a superar esa barrera. ¿Que rango de frecuencia podemos dar a la portadora y esperar que sea capaz de “atravesar” nuestra sistema transportando nuestra información sin distorsiones? ¿Por qué?

- El otro aspecto fundamental es la **posibilidad de que varias señales compartan un mismo medio de comunicación**. Si p.e. cargamos tres señales de voz y las sumamos directamente, al escuchar el resultado tenemos obviamente una situación de solapamiento:

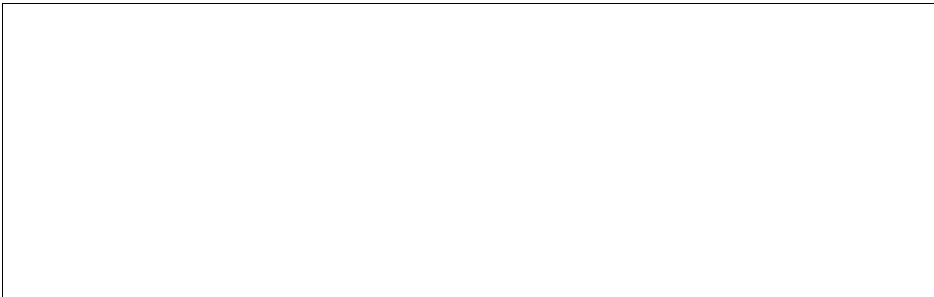
```
[v1 f]=wavread('voz1'); [v2 f]=wavread('voz1'); [v3 f]=wavread('voz1');  
sound(v1+v2+v3,f);
```

Considerandolo en el dominio temporal parece que no hay nada que hacer, ya que las tres señales comparte su dominio temporal y al sumarlas solaparán. Si por el contrario consideramos el dominio de frecuencias hoy hemos aprendido que:

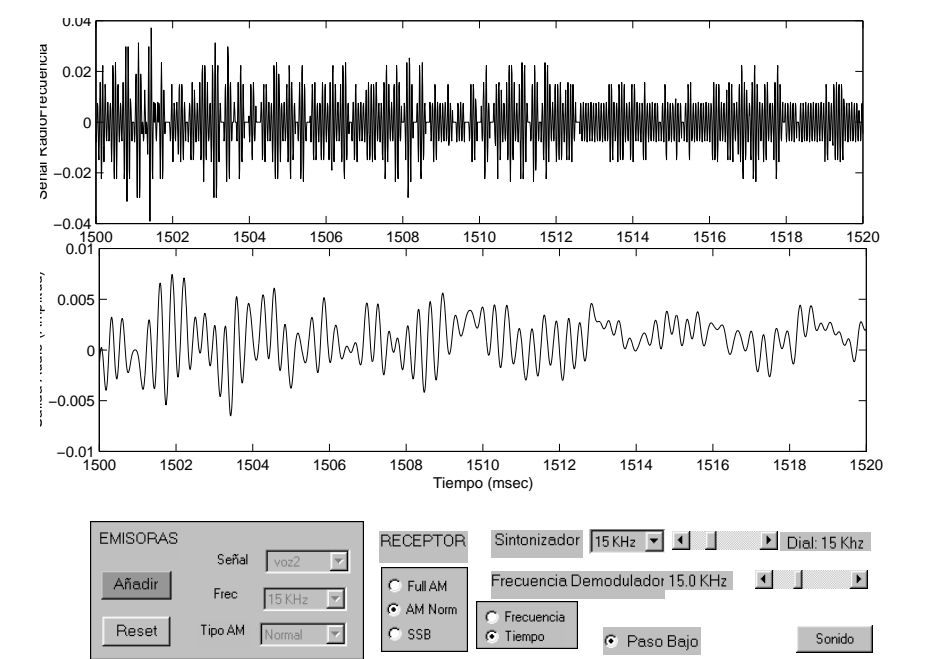
- En la práctica el rango de frecuencias o banda base de una señal de voz es finito.
- Usando la modulación podemos mover su TF a la frecuencia deseada.

Como veíamos en el ejemplo anterior existe un rango de frecuencias para el cual nuestro sistema es “transparente”. Dicho rango es el ancho de banda del sistema o medio de comunicación. Si dicho ancho de banda es suficientemente grande podríamos meter varias señales simultáneamente sin solaparse (modulandolas con diferentes portadoras). En nuestro caso: ¿cual es

el ancho de banda de nuestro sistema? ¿Cuántas de nuestras señales de voz podríamos meter simultaneamente? ¿En qué frecuencias? Haced un bosquejo indicando como se situarían los espectros de las diferentes señales.



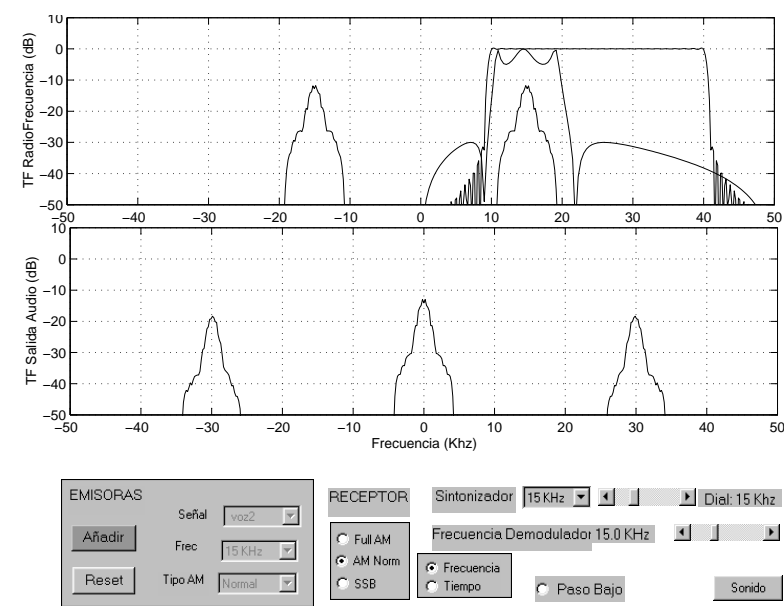
Simulador de emisor/receptor de AM: esta parte es opcional. En los apartados siguientes se plantean diversas cuestiones que debeis entregar junto con los comentarios que creais pertinentes en una hoja/s aparte, adjuntándolas a este guión.



Descripción de la aplicación

Trás entender los conceptos básicos de la modulacióni de señales y su aplicación a las comunicaciones vamos a ponerlos en práctica, jugueteando con una aplicación que simula un receptor de radio y varias emisoras. Teclear **simulador** para obtener la carátula mostrada en la página anterior.

Es una sencilla simulación de un receptor de radio que lleva incorporado un “osciloscopio” para poder visualizar tanto la señal de “radio” frecuencia que le llega (arriba) como la señal que envía a los altavoces (abajo). Ademas, como en las simulaciones el dinero no es un gran problema le hemos añadido un analizador de Fourier, que nos permite elegir entre ver las señales en el dominio temporal o en el dominio de frecuencias, sin más que darle al botón adecuado (ver figura siguiente).



Abajo a la izquierda se encuentran los mandos de la emisora. En princpio no estamos emitiendo nada. Pulsando **Añadir** se nos permite elegir la señal a emitir, la frecuencia de modulación y el tipo de modulación. Una vez elegidos debemos pulsar **Done** para que la señal resultante llegue al receptor. Pulsando **Reset** borramos todo y volvemos a empezar de cero.

A la derecha se muestran los mandos del receptor. Los más importantes son los botones que determinan el tipo de receptor o el dominio (tiempo/frecuencia) en el que se muestran los datos. Aparte de los aspectos de la interfaz gráfica las funciones usadas por debajo son las mismas que hemos visto antes: **modular**, **demodular**, **paso_bajo**, **ver_tf**, etc. La única novedad es que ahora habrá varias emisoras presentes, por lo que previo a la demodulación tendremos que elegir una. Esto lo lleva a

cabo el sintonizador (equivalente al dial de vuestra radio) que consiste en un filtro pasobanda de la frecuencia elegida. Dicha etapa no es muy crítica, por lo que podemos usar un filtro bastante ancho.

“Experimentando” con el simulador:

Esta son las cuestiones opcionales que podéis entregar, junto con los comentarios oportunos, adjuntándolas al guión de la práctica.

1. Añadir voz1 a 15 KHz, AM normal. Ver como el aspecto en frecuencias nos ayuda a ver que hay que hacer para recuperarla. Sintonizar a 15 Khz, escuchar, aplicar paso-bajo, escuchar. Comentar resultados.
2. Añadir una segunda voz a 20 KHz, AM normal. Intentar sintonizar tanto a 15 como a 20 KHz ¿Qué está sucediendo?.
3. Hacer un reset. Meter dos voces, pero ahora en 15 y 25 KHz. Contemplar la TF recibida. ¿Tendremos problemas esta vez? Demodularlas y escucharlas. Con lo visto en la práctica, ¿que separación mínima impondrías entre emisoras que emitan voz?
4. En Europa la separación entre emisoras en Onda Media es de 9 KHz y en Estados Unidos de 10 KHz. Recordando la extensión del espectro de una señal de ”música”, ¿se puede emitir música con buena calidad en OM?
5. ¿Qué pasaría si una emisora violase las normas de radiocomunicaciones y emitiese un pitido (frecuencia pura) de p.e. 8 KHz además de su programación habitual. ¿Molestaría a sus oyentes? ¿Y a los de las cadenas vecinas?
Hacer la prueba emitiendo voz1 en 15 Khz, voz2 en 35 Khz y la señal voz3++, que consiste en una señal de voz con un pitido de 8000 Hz en 25 Khz. ¿Qué pasa al sintonizar 25 Khz? ¿Y en 15 y 35 Khz?
6. Vamos a estudiar la sensibilidad del receptor ante desajustes del sintonizador y del demodulador con AM normal. Hacer un Reset, emitir voz1 en 25 Khz. Demodular a 25 KHz y escuchad el resultado.
Variar la frecuencia de demodulación en 0.5 o 1 KHz. ¿Se sigue escuchando bien la emisora? ¿Es muy crítico el acertar exactamente con la frecuencia del demodulador?
Afortunadamente hay métodos para que el receptor detecte la frecuencia exacta que le está llegando y se adapte a ella, por lo que en la práctica esto no supone un problema.
7. Hacer un Reset y seleccionar un receptor de tipo AM Full (uno que precisa portadora). Notad que ahora han desaparecido los controles de demodulación, ya que un receptor de estos simple-

mente aplica un diodo y un pasobajo. Hacer un Reset y emitir voz1 en 15 KHz AM Full y voz2 en 25 KHz AM normal. Sintonizar a 15 KHz, aplicar el diodo y el pasobajo y escuchar. Como aquí no usamos demodulación a través del $\cos()$, nuestro receptor es muy robusto aunque se desajustase un poco. Sintonizar ahora a 25 KHz, que corresponde a una emisora emitida con AM sin portadora. ¿Se oye correctamente?

8. Cambiar a receptor de tipo AM Normal (con demodulación, heterodino) y escuchad ambas emisoras. Un receptor AM Normal puede “lidiar” con emisoras FullAM, pero lo contrario no es posible.

SSB (Single Side Band)

Podríamos pensar que hemos llegado al limite del aprovechamiento de nuestro sistema. Un canal con 30 Khz de ancho de banda nos permite 3 emisoras con 10 Khz cada una.

Vamos a considerar un tipo adicional de modulación, SSB (Single Side Band):

1. Hacer un reset y emitir voz1 con 15 Khz en modo SSB. Observar el espectro resultante. ¿Qué hemos hecho? ¿Podremos recuperar la señal original? ¿En que propiedad de la TF nos basamos? ¿Cuántas emisoras cabrán ahora en nuestros 30 Khz?

2. Añadir voz2 a 20 Khz. Intentad sintonizar a 15 Khz. Observad el espectro y escuchad la señal resultante. ¿Qué sucede ahora? ¿Por qué?

Para recibir SSB hay que hacer un trabajo extra. Para que no se mezclen espectros adyacentes, precisaremos un paso-banda con un ancho de 5 Khz, medio canal de antes, luego es un filtro más crítico y más caro de implementar.

3. Seleccionar el receptor SSB. El filtro pasobanda del sintonizador ahora es más estrecho. Con nuestro nuevo receptor ya no hay solapamiento y la emisora se escucha OK. Un receptor SSB puede manejar también emisiones AM normal o FullAM. Comprobadlo.

AM SSB ahorra tanto ancho de banda como energía. Se usa en comunicaciones de largo alcance con emisoras de baja potencia, p.e. radioaficionados. Más cerca de nosotros la red telefónica usa SSB AM para meter muchas conversaciones por el mismo cable. A veces se distingue entre LSB (Lower Side Band) y USB (Upper Side Band) según qué mitad del espectro se retenga.

Referencias:

Basic Communication Theory, John Pearson, Prentice Hall International, 1992