Tema 10. Procesamiento de audio

Procesamiento de Señales, Sonido e Imágenes Digitales

Carlos Quemada Mayoral

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

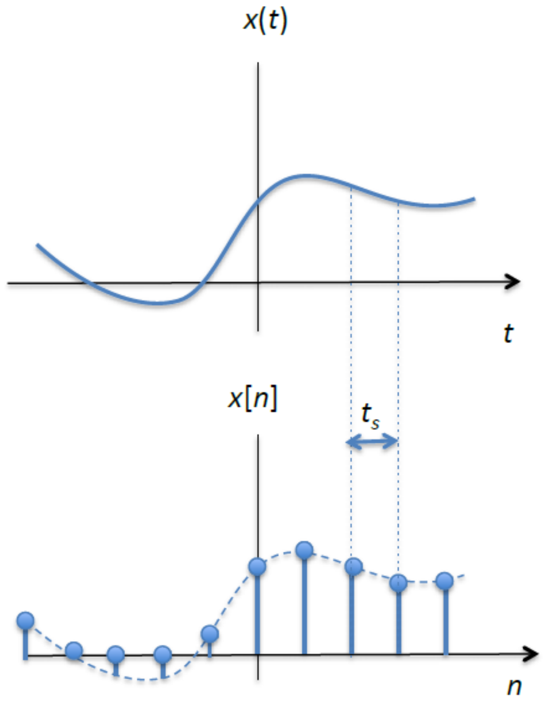
18/02/2021

Índice

10.1. Digitalización de audio

- El micrófono convierte una señal analógica (continua) en muestras discretas mediante los procesos de **muestreo** y **cuantificación** que realiza un **Conversor Analógico Digital** (CAD) incluido en el micrófono.

- **Muestreo**. Consiste en adquirir valores de tensión a intervalos de tiempo, habitualmente equiespaciados.



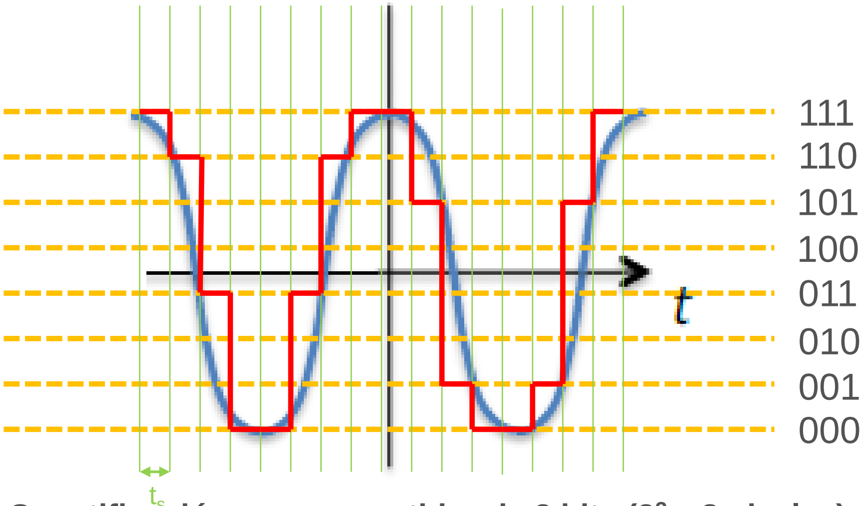
$$t_s = \frac{1}{f_s} = \frac{2\pi}{w_s} \qquad f_s = \frac{1}{t_s} \qquad w_s = \frac{2\pi}{t_s} \qquad w_s = 2\pi f_s$$

ts: periodo de muestreo
fs: frecuencia cíclica de muestreo
ws: frecuencia angular de muestreo

10.1. Digitalización de audio

- El micrófono convierte una señal analógica (continua) en muestras discretas mediante los procesos de **muestreo** y **cuantificación** que realiza un **Conversor Analógico Digital** (CAD) incluido en el micrófono.

- **Cuantificación**. Consiste en asignar un valor numérico discreto a la amplitud de cada muestra. En la práctica la cuantificación se hace con valores de 8 bits, 16 bits, o 32 bits.



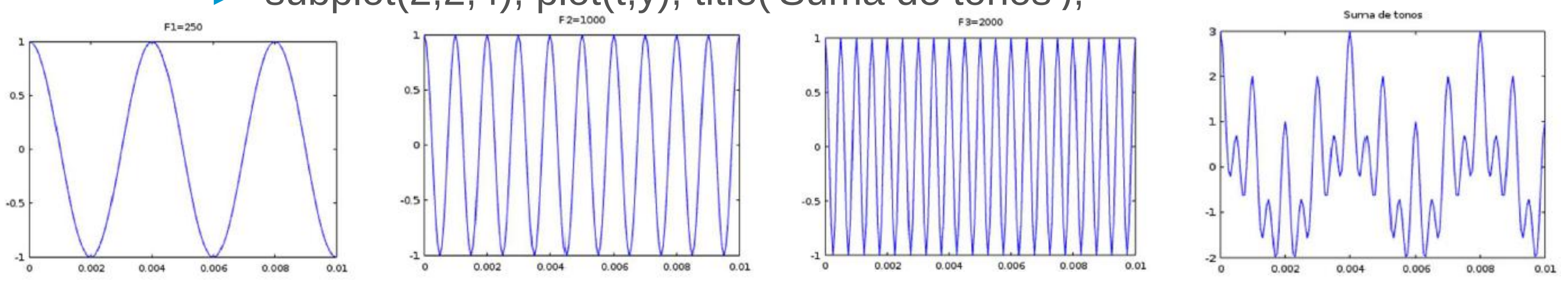
Cuantificación con convertidor de 3 bits (2^3 = 8 niveles)

10.1. Digitalización de audio

- Para reproducir el audio en un altavoz un Conversor Digital Analógico (CDA) realiza el proceso contrario, es decir, recibe la señal digitalizada y genera variaciones de voltaje continuas en el altavoz.

10.1. Digitalización de audio

- **Generación de un audio sintético**. Pasos:
- Seleccionamos la duración del audio a generar ⇒ d = 0.01;
- Seleccionamos la frecuencia y periodo de muestreo ⇒ Fs = 16000; Ts = 1/Fs;
- El número total de muestras es d/Ts = 160 muestras
- Generamos los valores de t de muestreo ⇒ t = [0:Ts:d-Ts];
- Generamos tres tonos y su suma ⇒ A = 1; F1 = 250; F2= 1000; F3=2000;
- y1 = A*cos(2*pi*F1*t)'; y2 = A*cos(2*pi*F2*t)'; y3 = A*cos(2*pi*F3*t)';
- y = y1+y2+y3;
- El apóstrofe(') convierte el vector fila en un vector columna
- Dibujamos los tonos y su suma:
    - subplot(2,2,1), plot(t,y1), title('F1=250');
    - subplot(2,2,2), plot(t,y2), title('F1=1000');
    - subplot(2,2,3), plot(t,y3), title('F1=2000');
    - subplot(2,2,4), plot(t,y), title('Suma de tonos');



10.1. Digitalización de audio

- **Reproducción de los tonos**. Pasos:
- Para reproducir un sonido en Octave podemos usar el comando sound(x,Fs,nBits).
- x ≡ Muestras a reproducir. Tradicionalmente Octave ha usado vectores columna pero si se los pasamos en un vector fila también los acepta.
- Fs ≡ La frecuencia de muestreo de la señal de audio.
- nBits ≡ Número de bits del ADC para realizar la cuantificación antes de ser enviados al dispositivo de salida. Pueden ser 8, 16 o 32 bits. Defecto, 8 bits.

- La señal a reproducir debe estar comprendida entre [-1.0,1.0]. Para normalizarla se puede usar el comando y = y ./ max(abs(y));
- Ahora podemos reproducir y con el comando sound(y,Fs);

10.1. Digitalización de audio

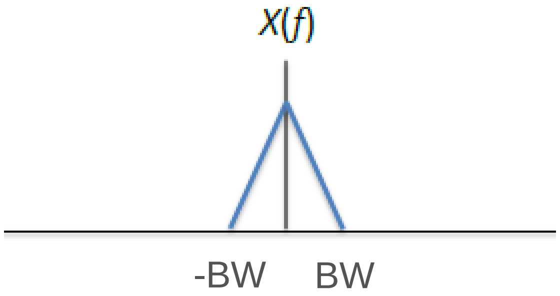
- **Señales mono y estéreo**.
- En Octave, un vector columna representa una señal mono y 2 vectores columna representa una señal estéreo. Por ejemplo, podemos generar una señal de audio que en el lado izquierdo genere un tono de 250Hz y en el lado derecho un tono de 2000Hz con los comandos:
- y_estereo = [y1 y3]; generamos una matriz con dos columnas
- sound(y_estereo,Fs); reproducimos en estéreo y1 e y3
- Una forma sencilla de convertir este sonido estéreo en mono es promediando las muestras del lado izquierdo y derecho $\Rightarrow$ y_mono = sum(y_estereo,2)./2;
- Si no ponemos el .2 hace la suma de los elementos de cada columna y los divide por dos, dando como resultado solo dos números, uno para la suma de la primera columna entre dos y otro para la suma de la segunda columna entre dos (vector 1 x 2).

10.1. Digitalización de audio

- **Grabar y leer ficheros de audio**.
- Para guardar señales de audio podemos usar el comando **wavwrite**: wavwrite(y,Fs,nBits,filename)
- El nombre del fichero **filename** y las muestras a grabar **y** son parámetros obligatorios.
- Los parámetros **opcionales** que podemos no proporcionar son:
  - La frecuencia de muestreo que por defecto vale Fs=8000.
  - El número de bits por muestra, que por defecto vale nBits=16.
- Análogamente, podemos usar el comando **wavread** para leer un fichero de audio:
- [y,Fs,nBits] = wavread('tono1.wav');
- Además de las muestras **y**, devuelve la frecuencia muestreo del fichero Fs y la profundidad de bits nBits de las muestras.
- También se puede indicar un número de muestras N a leer:
- [y,Fs,nBits] = wavread('tono1.wav',N);
- O incluso indicar un rango de muestras a leer:
- [y,Fs,nBits] = wavread('tono1.wav',[N1 N2]);

10.2. Frecuencia de muestreo

- **Criterio de Nyquist**. Establece que la frecuencia de muestreo Fs necesaria para recuperar la señal original continua sin *aliasing* debe ser al menos el doble de la frecuencia más alta de la señal original a digitalizar. Es decir, debe ser el doble del ancho de banda BW de la señal original.



- El oído humano es sensible a señales acústicas entre 20 Hz y 22 KHz. La frecuencia de muestreo que nos permite representar todos los sonidos audibles debería ser un poco mayor a 44 KHz. Por esta razón los formatos de digitalización de audio de alta fidelidad utilizan como frecuencia de muestreo 44.1 KHz.

10.2. Frecuencia de muestreo

- **Aliasing**. Solapamiento frecuencial que se produce cuando muestreamos la señal original a digitalizar a una frecuencia inferior a dos veces el ancho de banda BW de la señal.
- La señal muestreada $x_\delta(t)$ será el producto de la original x(t) por un tren de deltas.
- La transformada de Fourier de la señal muestreada será la convolución de la transformada de x(t) y la transformada del tren de deltas.
- La operación se simplifica porque es fácil demostrar, usando la fórmula de los coeficientes de la FS, que la transformada de un tren de deltas en el dominio del t es otro tren de deltas en el dominio de w.

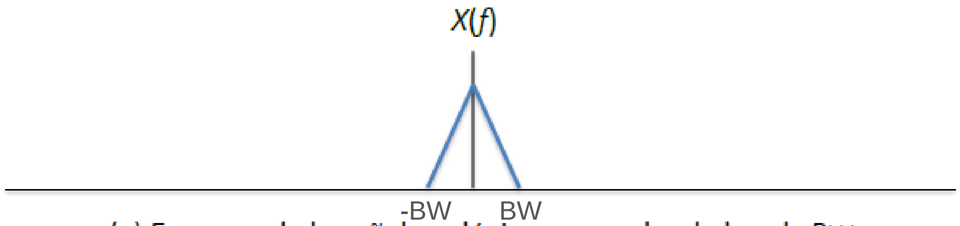
$$\sum_{n=-\infty}^{\infty} \delta(t - nT_s) \Longleftrightarrow F_s \cdot \sum_{m=-\infty}^{\infty} \delta(f - mf_s)$$

$$x_\delta(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \Longleftrightarrow X_\delta(t) = X(f) * F_s \cdot \sum_{m=-\infty}^{\infty} \delta(f - mf_s)$$

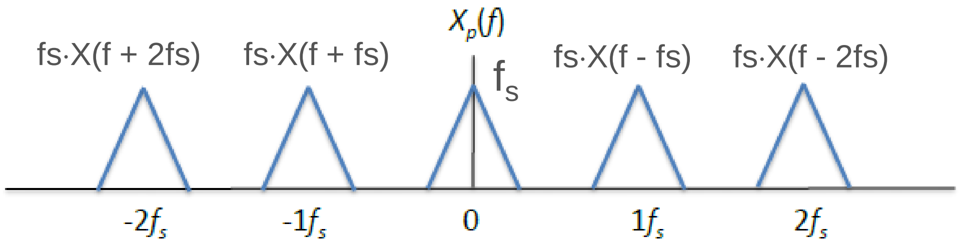
- Observando $X_\delta(f)$, consiste en replicar X(f) en frecuencias múltiplo de F_s.

10.2. Frecuencia de muestreo

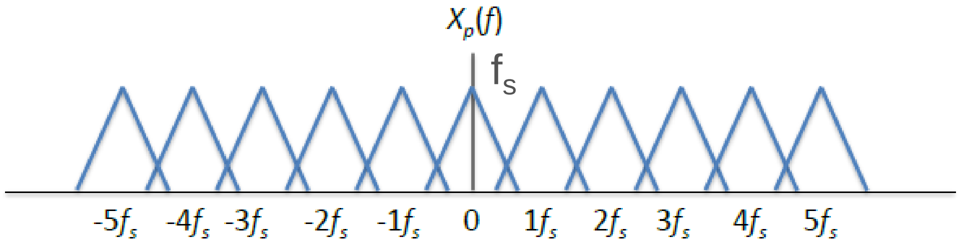
- **Aliasing**. Solapamiento frecuencial que se produce cuando muestreamos la señal original a digitalizar a una frecuencia inferior a dos veces el ancho de banda BW de la señal.



(a) Espectro de la señal analógica con ancho de banda BW



(b) Espectro de la señal muestreada con $f_s > 2B$W



(c) Espectro de la señal muestreada con $f_s < 2B$W

10.2. Frecuencia de muestreo

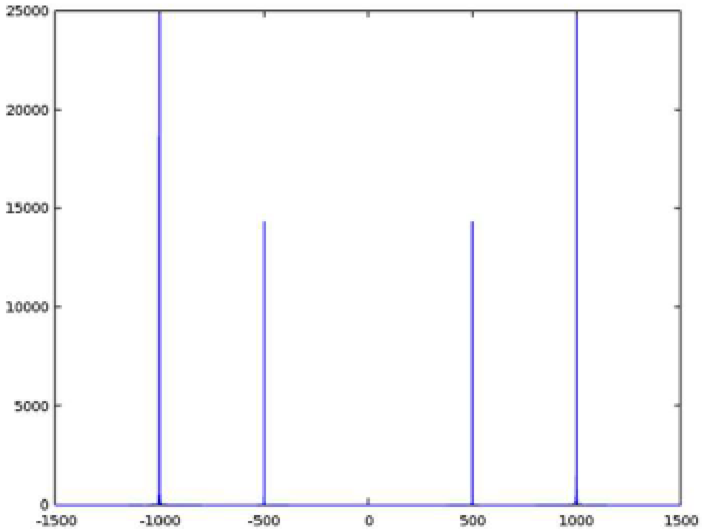
- **Aliasing**. Ejemplo: Sea la señal x(t) = cos(2π500t) + cos(2π1000t) + cos(2π2000t), que sucede cuando la muestreamos 3000 muestras/seg y 5000 muestras/seg.

Fs = 3000; Definimos la frecuencia de muestreo
Ts = 1/Fs; Definimos la separación temporal entre muestras
t = [0: Ts :10-Ts]; Definimos los instantes de muestreo hasta 10 segundos – Ts (Para seleccionar un número entero de periodos. Si lo definimos hasta 10 estamos seleccionando una muestra más correspondiente al periodo siguiente y el resultado cambia)
x = cos(2*pi*500*t) + cos(2*pi*1000*t) + cos(2*pi*2000*t); definimos x(t)
X = fft(x); Calculamos la fft de x(t)
N = length(X);
Tf = Fs/N; Calculamos la separación entre dos componentes contiguas de la fft
f = [-0.5*Fs:Tf:0.5*Fs]; Definimos el eje de frecuencias considerando la separación anterior. Puesto que el muestreo de una señal genera réplicas del espectro original cada Fs Hz, es suficiente con pintar un rango de Fs Hz
f = f(1:end-1); Quitamos una muestra para que tenga las mismas que la fft
plot(f,abs(fftshift(X))); Pintamos la fft centrada en cero

10.2. Frecuencia de muestreo

- **Aliasing**. Ejemplo: Sea la señal x(t) = cos(2π500t) + cos(2π1000t) + cos(2π2000t), que sucede cuando la muestreamos 3000 muestras/seg y 5000 muestras/seg.

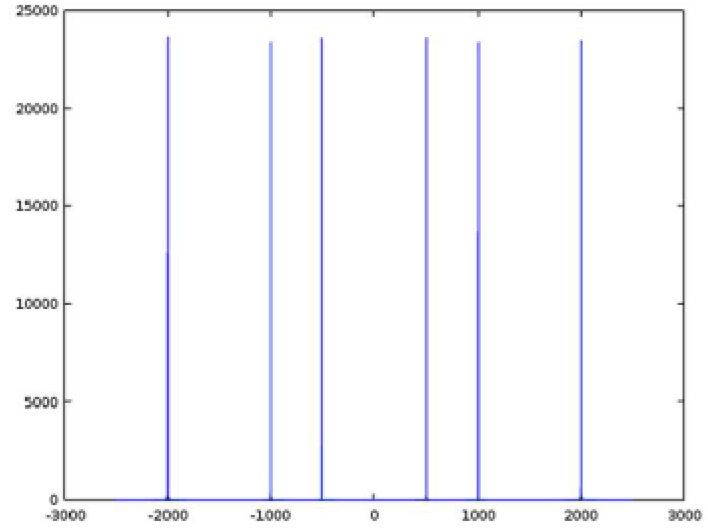
Aliasing



(a) f_{s1}=3000 muestras/seg

Podemos ver el aliasing donde el pulso a 2000Hz se suma al pulso a -1000Hz (y el pulso a -2000Hz se suma al pulso a 1000Hz)

No aliasing



(b) f_{s1}=5000 muestras/seg

Todas las amplitudes llegan hasta 25000. Comprobado en simulación. Al dividir entre el número de muestras, 50000, da amplitud 0.5 que es la amplitud de cada componente del coseno (f0 y –f0).

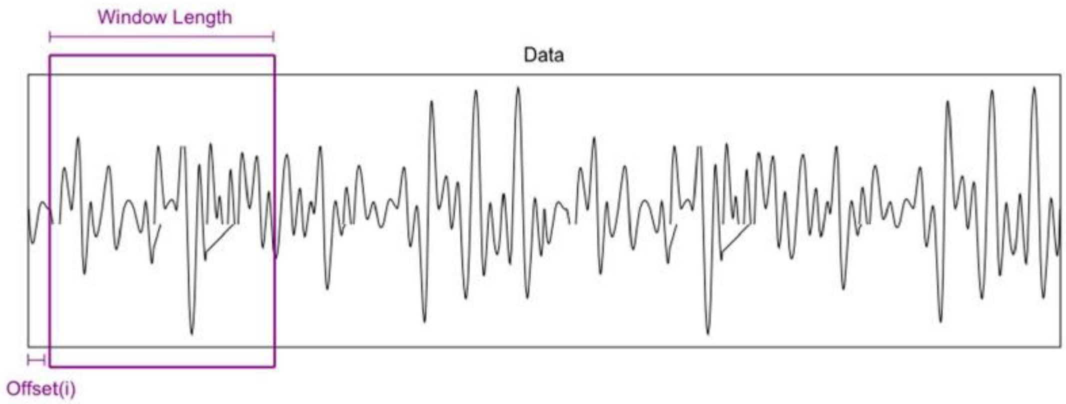
10.3. Enventanado

- La señal de audio es no estacionaria, es decir, que varía con el tiempo.
- Por esta razón, en la mayoría de aplicaciones de audio la señal se divide en ventanas de corta duración (10ms-50ms) y se analiza la señal ventana por ventana.
- Se llama enventanado a la operación de multiplicar una señal x[n] por una ventana w[n] de duración definida WL:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq W_L - 1 \\ 0, & \text{resto} \end{cases}$$

- En cada iteración multiplicamos la señal de audio x[n] por una versión desplazada de la ventana. El resultado es la señal enventanada $x_i[n]$:

$$x_i[n] = x[n]w[n - iW_L]$$

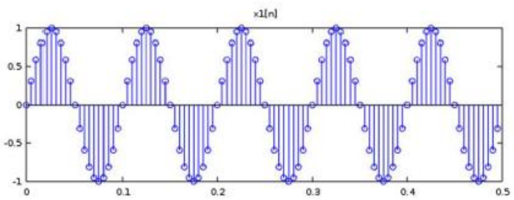


10.3. Enventanado

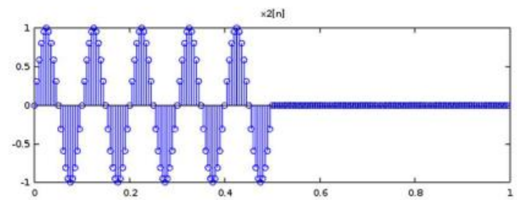
- **Compromiso entre resolución frecuencial y espacial**.
- En el tema 9 se vio que añadiendo ceros a x[n] (aumentando el número de muestras) daba lugar a una DFT con mayor resolución frecuencial.
- Este incremento de la resolución frecuencial es válido siempre que la señal permanezca estacionaria.
- Si las frecuencias que componen la señal cambian con el tiempo, el incremento de N da lugar a un promediado de las frecuencias en diferentes instantes de tiempo.
- Existe por tanto un compromiso entre la resolución frecuencial y la resolución temporal. Experimentalmente se ha comprobado que los mejores análisis de señales de audio se producen con ventanas de entre 10 ms y 50 ms.

10.3. Enventanado

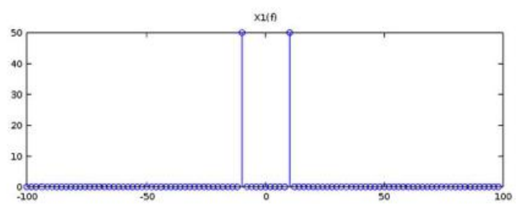
- **Goteo espectral**. La señal se desparrama por frecuencias distintas a la de la señal.
- Dependiendo de la ventana seleccionada, la señal final sobre la que se calcula la FFT puede contener saltos y picos temporales que pueden dar lugar al goteo espectral.
- La primera señal selecciona un número entero de periodos. Si replicamos de forma infinita (que es lo que hace la fft) la señal, obtenemos un seno perfecto. Debido a esto el espectro son deltas puras. Con la segunda, no ocurre eso.



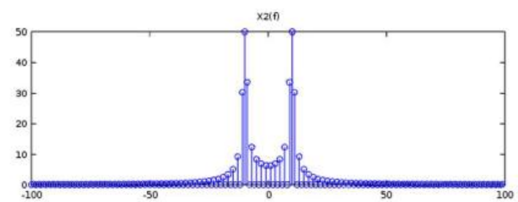
(a) Señal sin padding



(c) Señal con padding



(b) Representación frecuencial de (a)



(d) Representación frecuencial de (c)

El padding ha introducido goteo espectral

10.4. STFT

- La **STFT** (Short Time Fourier Transform) es una representación frecuencial para señales que varían en el tiempo. Consiste en dividir la señal en ventanas y calcular la DFT de cada ventana. La longitud de la ventana juega un papel importante: ventanas más largas dan una mejor resolución espectral a cambio de reducir la calidad de la resolución temporal.

- La forma de representar la STFT es mediante un espectrograma.

- Un espectrograma es una imagen que representa la evolución de una señal en los dominios del tiempo y de la frecuencia. Para generarlo, los coeficientes DFT de cada ventana se colocan en una columna de una matriz. Las filas indican las diferentes frecuencias.

- Al poner en cada columna los coeficiente de la DFT de cada ventana, cada columna representa un instante temporal correspondiente a la longitud de la ventana.

- Del mismo modo, cada fila da información frecuencial de la señal.

10.4. STFT

- En Octave se puede calcular el espectrograma de una señal x usando el comando specgram(x,N,Fs), donde **N** indica el tamaño de la ventana en número de muestras y **Fs** la frecuencia de muestreo.

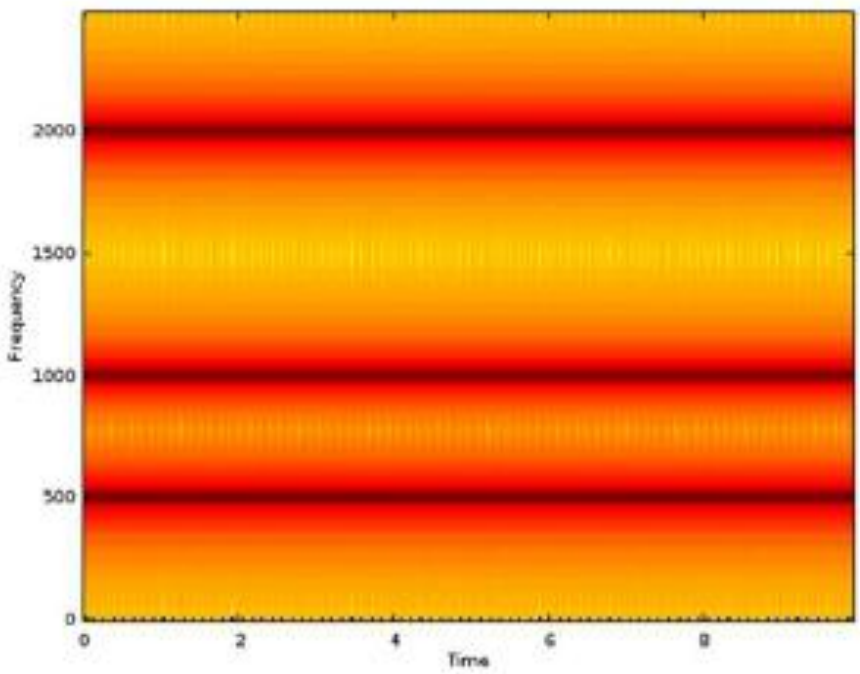
- Ejemplo: Calcular el espectrograma de los tres tonos del ejemplo anterior.

- Ejecutando specgram(x), Octave seleciona por defecto una frecuencia de muestreo de 2 Hz, con lo que es muy recomendable pasar siempre la frecuencia a la que está muestreada la señal. Dado que nuestra señal está muestreada a fs = 5000 Hz, la forma correcta de ejecutarlo es:

- specgram(x,256,5000);

10.4. STFT

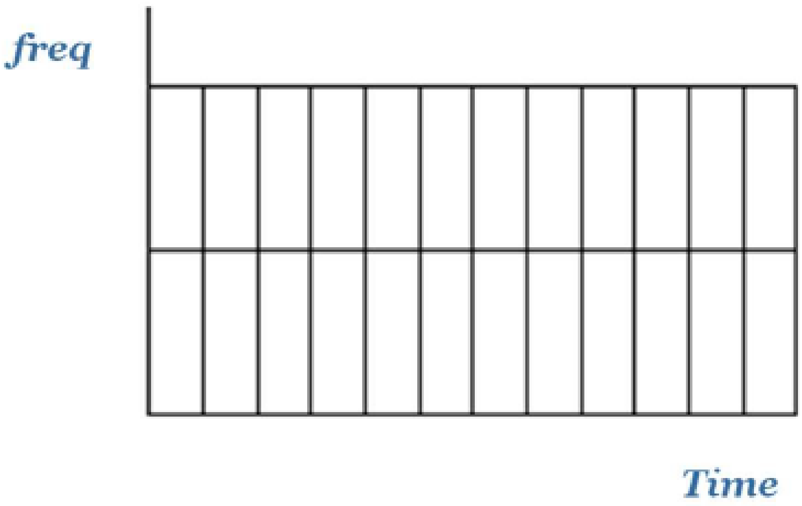
- specgram(x,256,5000);



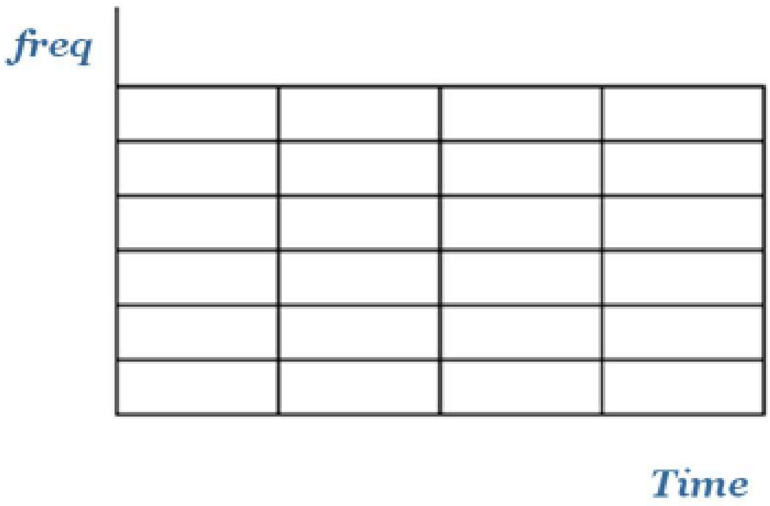
- Eje y: frecuencia, Eje x: tiempo
- Se observa como la energía se concentra en las frecuencias de los tonos y como al cambiar el tiempo, el espectrograma no varía (como debe ser)

10.4. STFT

- Forma que adoptan los coeficientes dependiendo de N



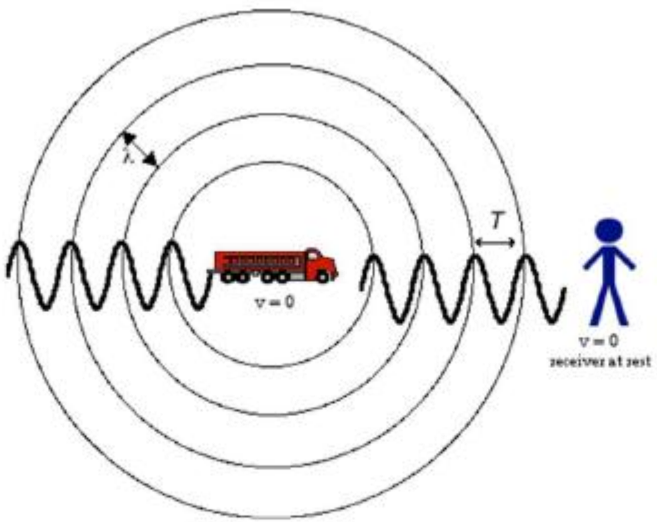
N más pequeño da lugar a más resolución temporal y menos frecuencial

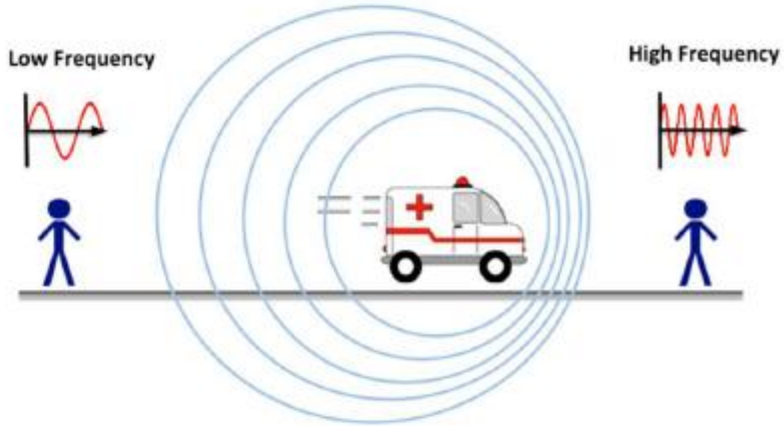


N más grande da lugar a más resolución frecuencial y menos temporal

10.5. Efecto Doppler

- Cuando está en movimiento el emisor o receptor de una onda (p.e., ondas acústicas o electromagnéticas) se produce una diferencia entre la frecuencia a la que el emisor produce las ondas $\mathbf{f_e}$ y la frecuencia a la que el receptor las recibe $\mathbf{f_r}$. Este fenómeno recibe el nombre de efecto **Doppler**, en honor a su descubridor, el físico Australiano Christian Doppler (1842).





- En la figura izq. el emisor está en reposo. En la figura se muestran los frentes de onda: ondas concéntricas centradas en el emisor y separadas por un periodo T. La figura dcha. muestra que cuando el emisor se acerca al receptor la frecuencia percibida en el receptor $\mathbf{f_r}$ aumenta y cuando se aleja disminuye.

10.5. Efecto Doppler

- **Longitud de onda**. Llamamos longitud de onda λ_e a la distancia recorrida por la onda en un periodo de vibración

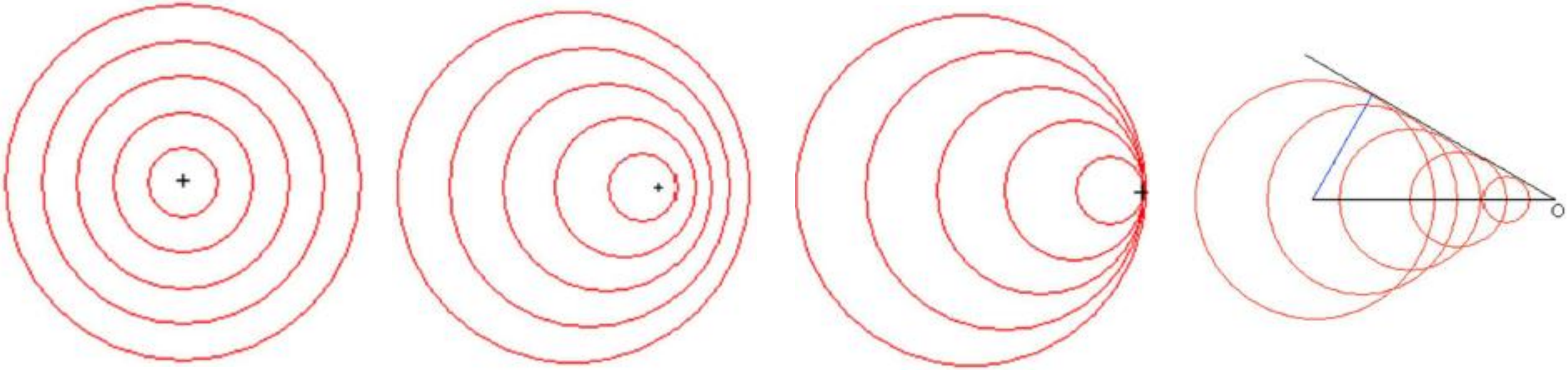
$$\lambda_e = c \cdot T_e = \frac{c}{f_e}$$

$c \equiv$ Velocidad de propagación de la señal
$T_e \equiv$ Periodo de la onda
$f_e \equiv$ frecuencia de la onda

- Donde c es la velocidad de propagación de la señal,

10.5. Efecto Doppler

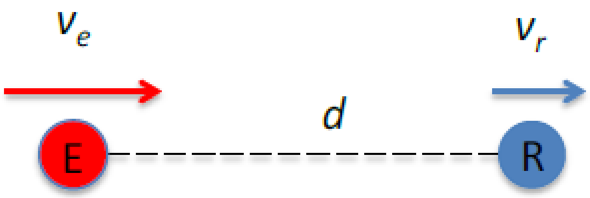
- **Frentes de onda para el movimiento del emisor (Rx a la derecha)**.



- **1**. $V_e = 0$. Los frentes de onda observados por el receptor son concéntricos y el receptor percibe la misma frecuencia f_e
- **2**. $V_e > 0$ y $V_e < c$. La longitud de onda percibida por el receptor es menor y la frecuencia mayor.
- **3.** $V_e = c$. La longitud de onda percibida por el receptor es cero y la frecuencia infinito.
- **4**. $V_e > c$. La longitud de onda percibida por el receptor comienza a ser negativa (aumenta en negativo conforme V_e aumenta) y la frecuencia percibida comienza a ser negativa (disminuye en negativo conforme V_e aumenta) (tiende a cero conforme V_e tiende a infinito). **La onda cónica se llama de choque**.

10.5. Efecto Doppler

- **Fórmula**. Supongamos que el emisor E y el receptor R se desplazan en la misma dirección y están separados por una distancia d



- La frecuencia de la señal recibida en el receptor $\mathbf{f_r}$ se relaciona con la frecuencia de la señal emitida por el emisor $\mathbf{f_e}$ del siguiente modo:

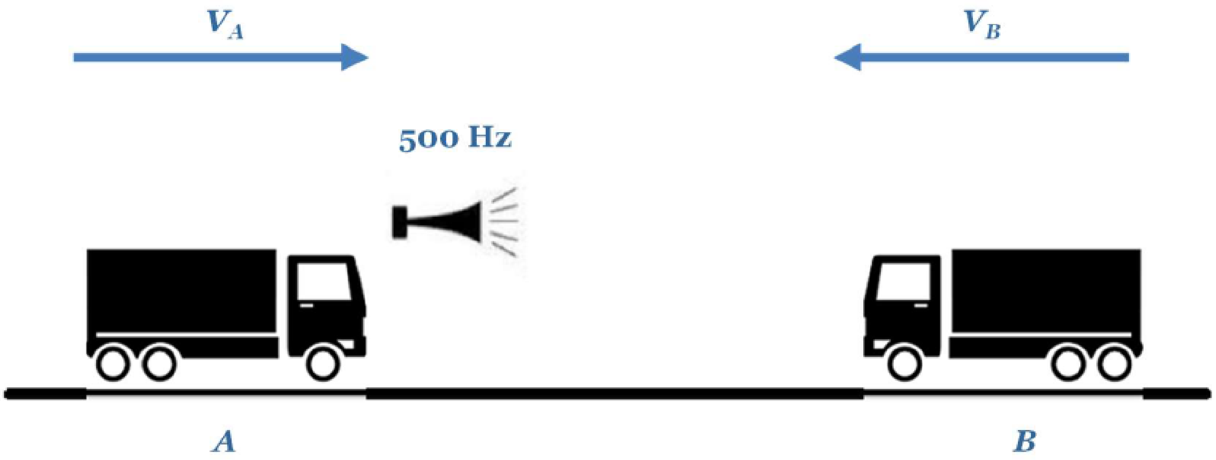
$$f_r = \frac{c - v_r}{c - v_e} f_e$$

- Si alguna de las velocidades cambia de sentido habría que cambiar el signo de $\mathbf{v_r}$ o $\mathbf{v_e}$.

- Si el emisor y receptor no se mueven en la línea que une ambos, habría que calcular las componentes de velocidad en dicha dirección multiplicando por el seno o coseno del ángulo correspondiente.

10.5. Efecto Doppler

- **Ejemplo**. Los camiones A, B se desplazan en direcciones opuestas por una carretera, tal como muestra la figura. Sus respectivas velocidades son VA=120 km/h, VB=95 km/h. El camión A toca la bocina que emite un tono con una frecuencia de 500Hz. Asumiendo una propagación del sonido de 1224 km/h, ¿con qué frecuencia la recibirá el camión B?



- Dado que las direcciones son opuestas las velocidades de emisión y recepción serán v_e=+120 km/h y v_r=-95 km/h. Aplicando la fórmula de Doppler:

$$f_e = \frac{c - v_r}{c - v_e} f_e = \frac{1224 + 95}{1224 - 120} 500 \text{ Hz} = 597 \text{ Hz}$$

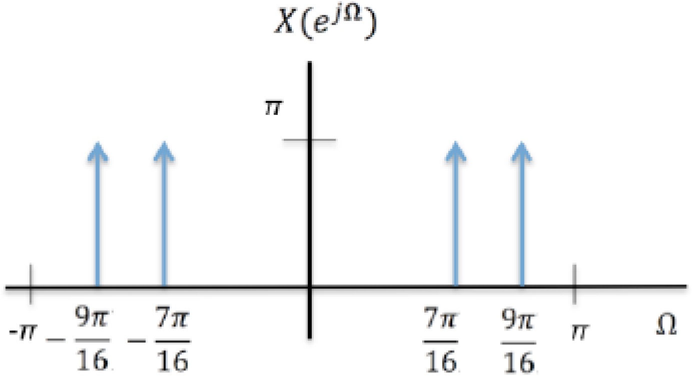
Ejercicio 1

- Calcular la DTFT y DFT de la señal $x_w[n]$ que resulta de truncar la señal sinusoidal x[n] con la ventana w[n].

$$x_w[n] = x[n]w[n] \qquad x[n] = \cos\left(\frac{7\pi}{16}n\right) + \cos\left(\frac{9\pi}{16}n\right) \qquad w[n] = \begin{cases} 1, & |n| \leq W_L/2 \\ 0, & \text{resto} \end{cases}$$

- Calculamos la DTFT de x[n] $\Rightarrow$ $X(e^{j\Omega}) = DTFT\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$
- Para calcular la DTFT de señales periódicas ilimitadas en tiempo, se usa el par transformado $\Rightarrow 2\pi\delta(\Omega - \Omega_0) \Leftrightarrow e^{j\Omega_0 n}$

$$X(e^{j\Omega}) = \pi\delta\left(\Omega + \frac{9\pi}{16}\right) + \pi\delta\left(\Omega + \frac{7\pi}{16}\right) + \pi\delta\left(\Omega - \frac{9\pi}{16}\right) + \pi\delta\left(\Omega - \frac{7\pi}{16}\right) \quad -\pi < \Omega < \pi$$

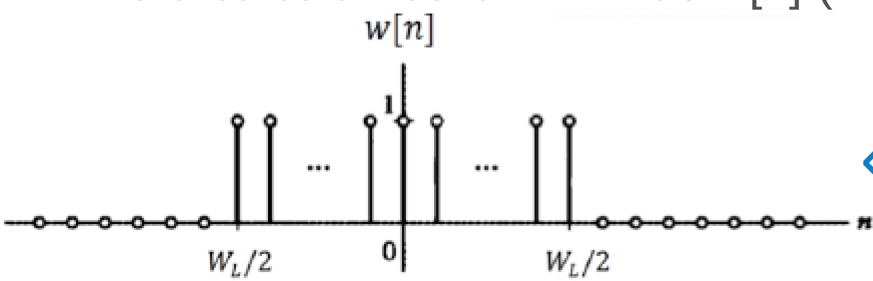


Ejercicio 1

- Calcular la DTFT y DFT de la señal $x_w[n]$ que resulta de truncar la señal sinusoidal x[n] con la ventana w[n].

$$x_w[n] = x[n]w[n] \qquad x[n] = \cos\left(\frac{7\pi}{16}n\right) + \cos\left(\frac{9\pi}{16}n\right) \qquad w[n] = \begin{cases} 1, & |n| \leq W_L/2 \\ 0, & \text{resto} \end{cases}$$

- Ahora calculamos la DTFT de w[n] (vista en Tema 8. Punto 8.6):



$$\Longleftrightarrow \qquad W(e^{j\Omega}) = \frac{\sin[\Omega(W_L+1)/2]}{\sin[\Omega/2]}$$

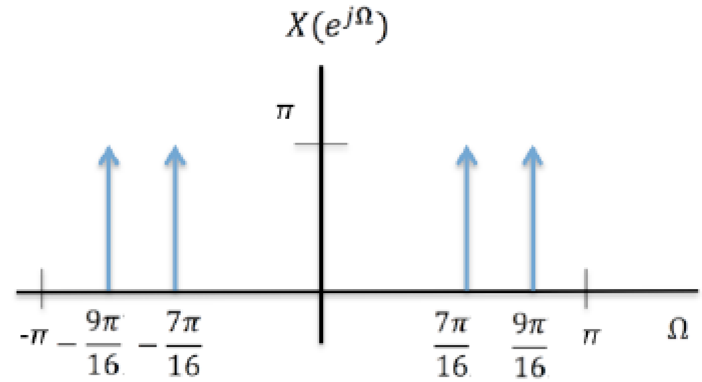
- Ahora aplicamos la propiedad del producto de señales de la DTFT:

$$x_w[n] = x[n]w[n] \xleftrightarrow{\text{DTFT}} X_w(e^{j\Omega}) = \frac{1}{2\pi} X(e^{j\Omega}) \circledast W(e^{j\Omega})$$

Ejercicio 1

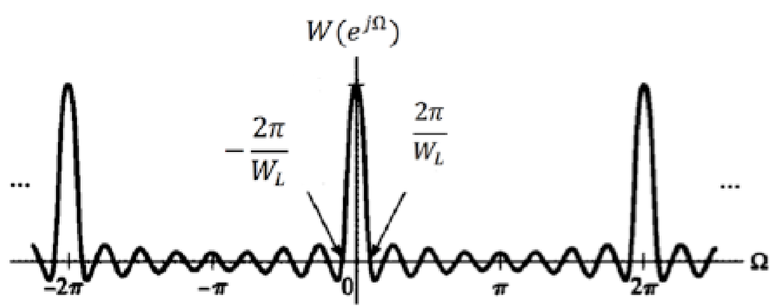
$$x_w[n] = x[n]w[n] \overset{\text{DTFT}}{\longleftrightarrow} X_w(e^{j\Omega}) = \frac{1}{2\pi} X(e^{j\Omega}) \circledast W(e^{j\Omega})$$

$$W(e^{j\Omega}) = \frac{\sin[\Omega(W_L+1)/2]}{\sin[\Omega/2]}$$



- Hay que hacer la convolución de W con cuatro deltas. Esto consistirá en evaluar W en las posiciones de las deltas.

Error: sustituir WL por WL + 1



$$X_w(e^{j\Omega}) = \frac{\sin\left(\left(\Omega + \frac{9\pi}{16}\right) W_L/2\right)}{2\sin\left(\left(\Omega + \frac{9\pi}{16}\right)/2\right)} + \frac{\sin\left(\left(\Omega + \frac{7\pi}{16}\right) W_L/2\right)}{2\sin\left(\left(\Omega + \frac{7\pi}{16}\right)/2\right)} + \frac{\sin\left(\left(\Omega - \frac{9\pi}{16}\right) W_L/2\right)}{2\sin\left(\left(\Omega - \frac{9\pi}{16}\right)/2\right)} + \frac{\sin\left(\left(\Omega - \frac{7\pi}{16}\right) W_L/2\right)}{2\sin\left(\left(\Omega - \frac{7\pi}{16}\right)/2\right)}$$

Ejercicio 1

Relación de la DTFT con la DFT

$$X[m] = X(e^{j\Omega})\Big|_{\Omega=\frac{2\pi m}{N}} = X\left(e^{j\frac{2\pi m}{N}}\right) \quad \text{con} \quad m = 0,1,\ldots,N-1$$

Coeficientes de la DFT

Error: sustituir WL por WL + 1

$$X_w[m] = \frac{\sin\left(\left(\frac{2\pi m}{N}+\frac{9\pi}{16}\right)W_L/2\right)}{2\sin\left(\left(\frac{2\pi m}{N}+\frac{9\pi}{16}\right)/2\right)} + \frac{\sin\left(\left(\frac{2\pi m}{N}+\frac{7\pi}{16}\right)W_L/2\right)}{2\sin\left(\left(\frac{2\pi m}{N}+\frac{7\pi}{16}\right)/2\right)}$$

$$+\frac{\sin\left(\left(\frac{2\pi m}{N}-\frac{9\pi}{16}\right)W_L/2\right)}{2\sin\left(\left(\frac{2\pi m}{N}-\frac{9\pi}{16}\right)/2\right)} + \frac{\sin\left(\left(\frac{2\pi m}{N}-\frac{7\pi}{16}\right)W_L/2\right)}{2\sin\left(\left(\frac{2\pi m}{N}-\frac{7\pi}{16}\right)/2\right)}$$

Sustituimos Ω por $2\pi m/N$

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

unir

www.unir.net