

Procesamiento Digital de Señales - Tarea 4

Fecha de entrega: 25 de Febrero

Transformada de Fourier en Octave

Octave incluye la función `fft` la cual devuelve la *transformada de Fourier discreta* (TDF) $X[k]$, la cual para una señal de entrada $x[n]$ se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad 0 \leq k < N.$$

Notar que la TDF es básicamente una versión muestreada (discretizada) de la transformada continua de Fourier que se vió en clase, donde el muestreo se realiza a intervalos de $2\pi/N$. En Octave, por default se tiene $N=\text{length}(x)$.

Concretamente, podemos calcular en Octave la TDF de una señal x como sigue:

```
X = fft(x);
```

O bien, si se requiere un valor para N distinto al predeterminado, podemos escribir:

```
X = fft(x, N);
```

Recordemos que las frecuencias que evalúa la función `fft(x)` son de la forma $\omega = 2\pi k/N$ para $k = 0, \dots, N-1$, por lo cual $0 \leq \omega < 2\pi$. Comúnmente se grafican las frecuencias de $-\pi$ a π , donde, debido a la periodicidad de la FFT, el intervalo de $-\pi$ a 0 corresponde al intervalo de π a 2π . Por lo tanto, podemos tener la TDF de una señal en el intervalo de $-\pi$ a π simplemente desplazando circularmente la salida de `fft(x)` por $N/2$ muestras. Esto puede lograrse mediante la función `fftshift`, como se observa en el ejemplo siguiente:

```
% Calcula la TDF de x[n]
X = fft(x);
% Grafica la TDF en el intervalo 0 a 2 * pi
plot(linspace(0, 2 * pi, length(X)), abs(X));
% Grafica la TDF en el intervalo -pi a pi
plot(linspace(-pi, pi, length(X)), abs(fftshift(X)));
```

Similarmente, la función `ifft(X)` calcula la transformada inversa de Fourier de una señal X . Esto puede verificarse con el siguiente código:

```
x = rand(1, 100);
% Calcula error cuadrático medio entre x e ifft(fft(x))
mean(abs(ifft(fft(x)) - x).^2)
```

El resultado del código anterior debe ser un número muy cercano a cero (no es cero debido a errores de redondeo).

Ejercicios:

1. Escriba una función `kernel_bp(w0, dw, N)` que devuelva el kernel $h_{bp}[n]$ de un filtro ideal pasa-banda para $-N \leq n \leq N$, donde w_0 (ω_0) es la frecuencia de entonamiento y dw (d_ω) el ancho de banda del filtro. Recuerde que el kernel de este filtro puede obtenerse aplicando la propiedad de corrimiento en frecuencia al kernel de un filtro ideal pasa-bajas:

$$h_{bp}[n] = e^{j\omega_0 n} h_{lp}[n] = e^{j\omega_0 n} \frac{\sin d_\omega n}{\pi n}.$$

Grafique la magnitud de la respuesta en frecuencia de un filtro pasa-banda ideal con $\omega_0 = \pi/2$, $d_\omega = \pi/4$, $N = 64$.

2. Un filtro de Gabor es un filtro pasa-banda con respuesta en frecuencia Gaussiana, el cual puede obtenerse desplazando en frecuencia un kernel Gaussiano $g[n]$, dado por

$$g[n] = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{n^2}{2\sigma_n^2}},$$

donde σ_n es el ancho del filtro en tiempo, y $\sigma_\omega = 1/\sigma_n$ es el ancho de banda.

Escriba una función `kernel_gabor(w0, sigma_w)` que devuelva el kernel $h[n]$ de un filtro de Gabor entonado a una frecuencia w_0 con ancho de banda `sigma_w`, evaluado en $-3\sigma_t \leq n \leq 3\sigma_t$. Suponiendo una frecuencia de muestreo $f_m = 200$ Hz, grafique la magnitud de la respuesta en frecuencia de un filtro de Gabor entonado a 50 Hz con un ancho de banda $\sigma_\omega = 10$ Hz.

3. Implemente una función que devuelva la respuesta al impulso de un filtro de respuesta senoidal (ver página 33 de las notas de clase). Esto puede lograrse calculando la señal $H[k] = H(\exp(j2\pi kn/N))$ para $k = 0, \dots, N-1$, donde $H(e^{j\omega})$ es la respuesta en frecuencia del filtro y N es la longitud del kernel deseado. La función debe tomar como parámetros la frecuencia de entonamiento ω_k , el ancho de banda b , y N .
4. Grafique la respuesta en frecuencia correspondiente a cada uno de los tres filtros anteriores entonados a una frecuencia $\omega_0 = \pi/6$ y con un ancho de banda de $\pi/3$. Discuta las diferencias entre los tres filtros, tanto en la forma de su respuesta en frecuencia, utilizando el mismo tamaño del kernel para todos los filtros.