Concepto tiempo-frecuencia y filtrado

1 Introducción

La transformada de Fourier es una de las herramientas matemáticas más importantes para el análisis de señales. En esta práctica veremos algunas transformadas y estudiaremos el concepto de filtrado de señales.

2 Transformada de Fourier

La transformada de Fourier de una señal $h(w)$, denotada por $H(w)$, será una función compleja de variable real y, por tanto, representaremos su módulo $\mid H(w) \mid$ y su fase $\prec H(w)$ por separado. Es práctica habitual representar $10\log(|H(w)|^2)$ en lugar de $\mid H(w) \mid$ debido a que muchos sistemas se diseñan para que dejen pasar un determindo rango de frecuencias e introduzcan fuertes atenuaciones en otras. En estos casos los valores del módulo de la respuesta en frecuencia pueden flucturar mucho, por lo que es mejor representarlo sobre una escala logarítmica. A la función $10\log|H(w)|^2 = 20\log|H(w)|$ se le denomina módulo de H(w) expresado en decibelios (dB) o simplemente *ganancia en dB* del filtro.

Ejemplo: utilizando la funcin `dtft` calcule la tansformada de Fourier de la siguiente señal.

```
nh=0:4;
h=ones(1,5)/5;
subplot(211);
stem(nh,h);
title('h(n)');
subplot(212);
[H,W]=dtft(h,500);
HdB=20*log10(abs(H));
plot(W,HdB);
axis([-4 4 -50 0]);
title('Modulo de la respuesta en Frecuencia del Filtro');
```

Observe que la función llamada `dtft` calcula la TF de una secuencia[1]. La respuesta en frecuencia que resulta puede verse en la figura 1.

2.1 Ejercicios

1. Genere y represente las siguiente señales:

$$\begin{aligned} x_0(n) &= cos(\pi n) \\ x_1(n) &= sen(\pi n) \\ x_2(n) &= cos(\frac{\pi}{4}n) \end{aligned}$$

[1]Haga `help dtft` para obtener más información sobre la función `dtft`

2. Calcule y represente las transformadas de Fourier de las señales anteriores.

3. Compare las transformadas de Fourier y razone los resultados.

3 Filtrado

Los filtros son sistemas Lineales e Invariantes en el Tiempo (LTI) y, como tales, su salida es la convolución entre la entrada $x(n)$ y la respuesta al impuso del sistema $h(n)$. La convolución se define por

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{1}$$

MATLAB posee ya la función `conv` que lleva a cabo la convolución de dos vectores de longitud finita [2]. Así pues, dada una respuesta al impulso $h(n)$ de longitud finita y una entrada $x(n)$ también de longitud finita, la salida puede obtenerse utilizando la función `conv`.

Cabe destacar que cuando pasamos a trabajar en el dominio de la frecuencia la operación de convolución se transforma en un simple producto:

`y(n)=x(n)*h(n)` $\longleftrightarrow$ `Y(w)=X(w).H(w)`

donde Y(w), X(w) y H(w) son las transformadas de Fourier de y(n), x(n) y h(n) respectivamente. H(w) se conoce con el nombre de *respuesta en frecuencia* del filtro.

Para ilustrar el significado del filtrado, se ha elaborado el programa `respfrec`. El programa considera un sistema LTI con una respuesta al impulso

$$h(n) = \frac{sen(0.1\pi n)}{\pi n}$$

y determina la salida cuando la entrada es la señal

$$x(n) = x_0(n) + x_1(n) \tag{2}$$

donde

$$\begin{aligned} x_0(n) &= cos(\frac{\pi}{12}n) \\ x_1(n) &= cos(\frac{\pi}{3}n) \end{aligned}$$

Para ello el programa comienza generando $x(n)$ en el rango $0 \leq n \leq 100$. Se dibujan en una primera ventana gráfica las dos componentes sinusoidales que constituyen la entrada. A continuación dibuja la entrada, la respuesta al impulso del filtro y su salida en el dominio temporal. Finalmente, se dibujan las tres señales en el dominio de la frecuencia entre $-\pi$ y π (utilizando la transformada de Fourier en 500 puntos). Puede observarse como, al tratarse de un filtro paso bajo, la componente de alta frecuencia ha sido eliminada y la salida coincide con la componente de baja frecuencia de la entrada.

3.1 Ejercicios

1. Ejecute el programa `respfrec` consultando aquellas instrucciones para las que desconozca su funcionamiento.

[2]Haga `help conv` para obtener más información sobre la función `conv`

2. Genere y dibuje la siguiente señal de entrada, considerando los mismos rangos que antes para la generación de las señales,

$$x(n) = x_0(n) + x_1(n) + x_2(n) + x_3(n) \tag{3}$$

donde

$$\begin{aligned} x_0(n) &= cos(\pi n) \\ x_1(n) &= cos(\frac{\pi}{2}n) \\ x_2(n) &= cos(\frac{\pi}{4}n) \\ x_3(n) &= cos(\frac{\pi}{8}n) \end{aligned}$$

Modifique el programa para que funcione con esta entrada y las siguientes respuestas al impulso:

(a) Filtro paso bajo

$$h_{bajo}(n) = \frac{sen(Wn)}{\pi n}$$

Observe que la salida coincide con $x_3(n)$ para $W = \frac{3\pi}{16}$.
Haga lo mismo con filtros de este tipo usando estos anchos de banda:$\frac{3\pi}{8}$, $\frac{3\pi}{4} y\ \pi$.
Observe con quien se corresponden las salidas de estos filtros.

(b) Filtro paso banda

$$h_{banda}(n) = \frac{sen(Wn)}{\pi n}\ cos(W_c n)$$

Observe que la salida coincide con $x_2(n)$ para $W = \frac{\pi}{16}$ y $W_c = \frac{\pi}{4}$.
Calcule el filtro paso banda para quedarse con la señal $x_1(n)$ y observe su salida.

(c) Filtro paso alto

$$h_{alto}(n) = \delta(n) - \frac{sen(Wn)}{\pi n}$$

Observe que la salida coincide con $x_0(n)$ cuando $W = \frac{3\pi}{4}$. Eliga la frecuencia W para que la salida sez $x_0 + x_1$.

4 Función dtft.m

```
function [H,W]=dtft(h,N)

%DTFT calculate DTFT at N equally spaced frequencies
% usage: [H,W]=dtft(h,N);
% h: finite-length input vector, whose length is L
% N: number of frequencies for evaluation over [-pi,pi]
% ==> constraint: N>=L
%
% H: DTFT values (complex)
% W: vector of freqs where DTFT is computed
%
N=fix(N);
```

```
L=length(h);
h=h(:); %<-- for vectors ONLY %
if (N<L)
error('DTFT: # data samples cannot exceed # freq samples')
end
W=(2*pi/N) * [0:(N-1)]';
mid=ceil(N/2)+1;
W(mid:N)=W(mid:N)-2*pi; %<-- move [pi,2pi] to [-pi,0]
W=fftshift(W);
H=fftshift(fft(h,N)); %<-- move negative freq components
```

5 Función respfrec.m

```
%*******************************************************************
%
% Programa: respfrec.m
%
%
% Fecha: Noviembre 2004
%
%*********************************************************************
clear all;
close all;

% Generacion senal de entrada
Lx=100;
n=0:Lx;
x0=cos(pi/12*n);
x1=cos(pi/3*n);
x=x0+x1;

% Dibujo componentes de la senal
figure(1);
subplot(311);
plot(n,x0);
title('Componente x0(n)');
grid
subplot(312);
plot(n,x1);
title('Componente x1(n)');
grid
subplot(313);
plot(n,x);
title('x(n)=x0(n)+x1(n)');
grid
disp('Pulse una tecla...')
pause

%       Calculo de la transformada de Fourier
[X,Wx]=dtft(x,500);
```

```
%XdB=20*log10(abs(X));

%=========================================================
% Obtencion de respuesta al impulso del filtro
Lh=50;
nh=(-Lh:Lh)+eps;
h=sin(0.1*pi*nh)./(pi*nh);

%       Calculo de la transformada de Fourier
[H,Wh]=dtft(h,500);
%HdB=20*log10(abs(H));


%=====================================================
% Calculo de la salida y dibujo de la salida y su T.F.

y=conv(x,h);

%       Calculo de la transformada de Fourier
[Y,Wy]=dtft(y,500);
%YdB=20*log10(abs(Y));

%==================================================
% Representacion de las senhales en tiempo

figure(2);
subplot(311);
plot(n,x);
title('Entrada al filtro: x(n)=x0(n)+x1(n)');
grid

subplot(312)
plot(nh,h)
grid;
title('Respuesta al impulso del filtro, h(n)')

subplot(313)
plot(-Lh:Lx+Lh,y)
axis([0 Lx floor(min(y)) ceil(max(y))])
grid;
title('Salida del filtro')

disp('Pulse una tecla...')
pause

%=======================================================
% Representacion de las senhales en frecuencia


figure(3)
subplot(311);
```

```
plot(Wx,abs(X));
title('Entrada, X(W)');
grid

subplot(312)
plot(Wh,abs(H))
grid;
title('Filtro, H(W)')

subplot(313)
plot(Wy,abs(Y));
title('Salida, Y(W)');
grid
```