TRATAMIENTO DIGITAL DE LA SEÑAL
Transformada de Fourier Discreta (DFT)

Práctica 2

Fecha: 22 de marzo de 2010

1. Introducción

El objetivo de esta práctica es hacer un programa en `Matlab` que dibuje la transformada de Fourier de una señal discreta. En principio, `Matlab` no puede representar TF de señales discretas porque son funciones de la variable continua w. Sin embargo, puede obtenerse una aproximación bastante buena si se dibuja un número elevado de muestras de la Transformada de Fourier (TF) y luego se representa con la función `plot`. Para obtener las muestras de la TF existen dos procedimientos:

Analíticamente En los casos en que sea posible, se puede calcular analíticamente la TF de la secuencia y, a partir de la expresión obtenida, calcular directamente su valor en las muestras correspondientes a frecuencias equiespaciadas un valor Δ, es decir,

- Paso 1 (cálculo analítico): $X(e^{jw}) = TF[x(n)]$
- Paso 2 (obtención de muestras): $X(k) = X(e^{jw})|_{w=k\Delta}$

Habitualmente se suelen tomar N muestras de la TF en el período $[0, 2\pi)$ lo cual implica una elección de la separación entre las muestras de

$$\Delta = \frac{2\pi}{N} \tag{1}$$

y haciendo variar el índice k desde 0 hasta $N-1$.

Ejemplo: Se desea representar en matlab la TF de

$$x(n) = \begin{cases} 1 & -M \leq n \leq M \\ 0 & resto \end{cases} \tag{2}$$

Calculando analíticamente su transformada de Fourier se obtiene[1]

$$X(e^{jw}) = \frac{sen[w(M+1/2)]}{sen(w/2)} \tag{3}$$

Las muestras de la TF son

$$X(k) = X(e^{jw})|_{w=\frac{2\pi}{N}k} = \frac{sen[\frac{2\pi}{N}k(M+\frac{1}{2})]}{sen(\frac{\pi}{N}k)} \tag{4}$$

donde k es un índice que varía entre 0 y $N-1$.

[1]Observe que en $w=0$ existe una indeterminación que habrá que tener en cuenta como un punto particular de la señal.

Numéricamente Si la secuencia es de duración finita es posible obtener de forma numérica las muestras de la transformada de Fourier, sin necesidad de realizar el cálculo analítico de ésta. Suponiendo que la señal se extiende en n desde 0 hasta $L-1$, las muestras se calculan mediante la ecuación de análisis en el intervalo en que está definida la señal, obteniendo la siguiente expresión

$$X(k) \;=\; X(e^{jw})|_{w=\frac{2\pi}{N}k} = \sum_{n=0}^{L-1} x(n)e^{-jwn}|_{w=\frac{2\pi}{N}k} = \sum_{n=0}^{L-1} x(n)e^{-j\frac{2\pi}{N}kn} \qquad (5)$$

para $k = 0, \cdots, N-1$. Esto es lo que se llama la *Discrete Fourier Transform (DFT)*.

La expresión obtenida en (5) puede implementarse directamente mediante bucles. Sin embargo, cada uno de los elementos de $X(k)$ se puede obtener en `Matlab` de forma muy eficiente mediante un producto escalar entre dos vectores. Por ejemplo, la muestra k-ésima de la TF se puede calcular como el producto escalar de los vectores $\mathbf{x}_n = [x(0), x(1), \cdots, x(L-1)]^T$ y $\mathbf{f}_k = [e^{-j0}, e^{-j\frac{2\pi}{N}k}, e^{-j\frac{2\pi}{N}2k}, \cdots, e^{-j\frac{2\pi}{N}(L-1)k}]^T$, es decir,

$$X(k) = \mathbf{f}_k^T \mathbf{x}_n \qquad (6)$$

donde el superíndice ^{T} denota traspuesto[2].

Finalmente, si se quiere obtener un vector $\mathbf{X}_k$ que contenga todas las muestras $X(k)$, $\mathbf{X}_k = [X(0), X(1), \cdots, X(N-1)]^T$, se puede hacer multiplicando el vector $\mathbf{x}_n$ por una matriz $N \times L$ que llamaremos $\mathbf{F}$, cuyas filas son los vectores $\mathbf{f}_l^T$, donde

$$\mathbf{F} = \begin{pmatrix} \mathbf{f}_0^T \\ \vdots \\ \mathbf{f}_{N-1}^T \end{pmatrix} \qquad (7)$$

Es decir

$$\mathbf{X}_k = \mathbf{F}\mathbf{x}_n \qquad (8)$$

2. Objetivo

La presente práctica tiene como objetivo el uso del paquete matemático `Matlab` para la representación de la transformada de Fourier de las siguientes secuencias discretas:

$$x_1(n) \;=\; \begin{cases} 1 & 0 \le n \le 2M \\ 0 & resto \end{cases} \qquad (9)$$

$$x_2(n) \;=\; \begin{cases} n+1 & 0 \le n \le 2M \\ 4M-n+1 & 2M+1 \le n \le 4M \\ 0 & resto \end{cases} \qquad (10)$$

Observe que las secuencias (9) y (10) tienen longitud $L = 2M+1$ y $L = 4M+1$, respectivamente.

Recomendaciones: Para el cálculo analítico de las TF's trata de expresar (9) como una transformación de la secuencia ejemplo (2), y la secuencia (10) como resultado de convolucionar dos secuencias como la especificada en (9).

El cálculo de la TF se realizará de cuatro formas distintas[3]:

[2]Recuerde que en MATLAB x.'y x'representan el transpuesto y el transpuesto conjugado de x

[3]Para evitar errores de redondeo en los métodos numéricos, haga un redondeo a las milésimas para la representación de las DFT (puede basarse en el uso de la instrucción de matlab `round()`).

1. Mediante el muestreo, con N muestras espaciadas en múltiplos de $\frac{2\pi}{N}$, de la TF calculada analíticamente.

2. Por medio de bucles utilizando directamente la fórmula de la DFT de (5).

3. Por medio del producto escalar (8) en donde se usará la matriz $\mathbf{F}$, definida en (7). En este método **no se permite el uso de bucles**[4].

4. Usando la *transformada rápida de Fourier* (FFT, Fast Fourier Transform). Esta transformada es una implementación más eficiente[5] de (5). La instrucción de matlab `fft()` tiene implementada ésta [1].

2.1. Programa en matlab

El programa ha de representar la TF (módulo y fase) de una de las secuencias (9) ó (10), obtenida por los cuatro métodos indicados antes. Para ello, el programa preguntará al usuario cual será la secuencia a considerar y los valores de M y N.

El programa mostrará en una figura la secuencia a analizar[6]. En otra figura se representarán las gráficas del módulo y de la fase, correspondientes a un período $[0, 2\pi)$, de las TF obtenidas por los cuatro métodos especificados[7]. Además, el programa ha de mostrar, en la ventana de comandos, la matriz $\mathbf{F}$ utilizada para el tercer método.

2.1.1. Eficiencia de los Algoritmos

Para comparar la efectividad entre los tres últimos métodos, se realizará un *benchmark* respecto al tiempo empleado en su cómputo. Para realizar estos tenga en cuenta lo siguiente:

Tiempo de cómputo . Los comandos `clock` y `etime` de MATLAB permiten contar la fecha/hora actual y la diferencia en segundos entre dos instantes, respectivamente [1]. Usando un código similar al que a continuación se detalla, se puede comprobar el tiempo empleado para ejecutar el algoritmo.

```
tinicio=clock;
...
(CODIGO CALCULO DFT)
...
etime(clock,tinicio);
```

Aunque en este tiempo influyen otras tareas ejecutándose en el sistema donde se está utilizando matlab, nos servirá como referencia.

Haga este *benchmarck* para cualquiera de las secuencias anteriores y para los siguientes conjuntos de valores: $M = 10, N = 50$ y $M = 10, N = 64$. ¿Cuál de los métodos es el más eficiente? De los dos conjuntos de valores, compruebe para cuál se calcula más rápido la DFT en cada método, e interprete los resultados.

[4]Tenga en cuenta que si $\mathbf{A}$ es una matriz, la instrucción de matlab $exp(\mathbf{A})$ obtendrá una matriz cuyos elementos serán $e^{\mathbf{A}(i,j)}$ donde $\mathbf{A}(i,j)$ es el elemento de la fila i y columna j de $\mathbf{A}$.

[5]Particularmente eficiente cuando N es una potencia de 2.

[6]Para la representación de secuencias discretas se recomienda el uso del comando `stem`.

[7]Para representar varias gráficas en una misma figura se recomienda el uso de la instrucción de matlab `subplot`
.

3. Análisis frecuencial de señales

En este apartado se analizará la TF de señales sinusoidales como las generadas en la práctica anterior. Como las señales sinusoidales son periódicas y, por lo tanto, de longitud infinita, tendremos que recortar su duración. Para ello consideraremos la secuencia definida en la práctica anterior (ecuación (9)) de la siguiente forma:

$$x(n) \quad = \quad x_1(n)sen(2\pi\frac{f_0}{f_s}n) \tag{11}$$

donde $x_1(n)$ es el pulso rectangular definido en (9), f_0 es la frecuencia de oscilación de la señal sinusoidal continua y f_s es la frecuencia de muestreo. Se pide:

1. ¿Qué forma tiene $x(n)$? ¿Qué longitud tiene esta secuencia? Recordando que la TF del pulso rectangular, $x_1(n)$, calculado en el apartado anterior, tiene forma de sinc y que la de una señal sinusoidal son dos deltas centradas en la frecuencia f_0: ¿qué forma tendrá la TF de $x(n)$? **Recomendación:** recuerde la propiedad de modulación de la TF.

2. Mediante el algoritmo FFT (con $N = 1024$ muestras), ponga en una misma gráfica el módulo de la TF de $x(n)$ (con $M = 50$) para f_0 variando de 100 a 475Hz en saltos de 125Hz y con $f_s = 8000$Hz. Teniendo en cuenta que 2π equivale a f_s, muestre el eje horizontal en herzios. Interprete los resultados y compruebe que la TF de $x(n)$ es la que esperaba. ¿Dónde se centran los espectros de las señales?

3. Haga lo mismo para f_0 variando de 7525 a 7900Hz en saltos de 125Hz y con $f_s = 8000$Hz. ¿Donde se centran los espectros de las señales? ¿Qué ocurre al incrementar la frecuencia? ¿Por qué?

4. Repita el punto 2, considerando el intervalo de frecuencias $[-\pi, \pi)$ (es decir, $[-f_s/2, f_s/2)$)

4. Fecha de entrega

La fecha de entrega de la práctica será el día **15 de abril**. Se deberá(n) depositar el(los) fichero(s) `*.m` de el(los) programa(s) de matlab en un nuevo directorio del servidor subversion (SVN), llamado "Practica 2", siguiendo las instrucciones de la WIKI de la Facultad [2].

En la medida de lo posible CADA APARTADO DEBE REALIZARSE EN SUBDIRECTORIOS DIFERENTES CON EL NOMBRE DE ESTE. NO SE ACEPTAN FICHEROS COMPRIMIDOS.

Los programas ha de estar debidamente comentados, incluyendo en sus primeras líneas de comentarios el nombre del alumno. Las respuestas a las preguntas planteadas a lo largo de la práctica deben de escribirse en un fichero de texto, `respuestas.txt`. **Cada alumno** ha de conservar una copia de todos los programas para su evaluación individual el día del examen de prácticas.

Referencias

[1] Manual *online* de `matlab`. Instrucción `help`.

[2] https://wiki.fic.udc.es/cecafi