FFT: Transformada Rápida de Fourier

Schmidt, Ana Lucía

Estudiante de Ingeniería en Sistemas de Computación
Universidad Nacional del Sur, Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina
l.s.88904@gmail.com

Marzo 2013

Resumen: la Transformada Rápida de Fourier es un algoritmo que permite calcular eficientemente la Transforma de Fourier Discreta y su inversa. La Transformada Rápida de Fourier es de suma importancia en el análisis, diseño y realización de algoritmos y sistemas de procesamiento de señales dado que brinda mayor eficiencia tanto en tiempo como en recursos.

Palabras clave: Transformada Rápida de Fourier, FFT, Transformada de Fourier Discreta, Procesamiento de Señales.

I. INTRODUCCIÓN

Toda señal periódica puede ser representada por la suma de series de Fourier. Con un análisis adecuado es posible obtener una representación de Fourier para señales de duración finita. Esta representación es la que se conoce como la Transformada de Fourier Discreta (TFD). La TFD se puede representar como:

$$X[k] = \sum_{n=0}^{N-1} x[n].W_N^{kn} \qquad k = 0, 1, \dots, N-1 \tag{1}$$

donde $W_N = e^{-j\frac{2\pi}{N}}$

Se puede observar a simple vista que su resolución directa implica *N* multiplicaciones complejas y *N-1* adiciones complejas por cada *k*. Por lo tanto, el cálculo directo de una TFD es de orden $O(N^2)$.

Para valores pequeños de *N* la resolución en sí no consume mucho tiempo ni recursos. Sin embargo, para valores de *N* lo suficientemente grandes el cálculo directo se torna poco eficiente, no sólo por el gran tiempo que consume sino también por el acaparamiento de los recursos necesarios. Por ejemplo, para $N = 2^{30}$ las operaciones a realizar serían 2^{60}; asumiendo que cada operación toma aproximadamente 1ns el cálculo directo de la TFD tardaría unos 13343 días.

Se puede ver, entonces, que el orden del cálculo directo impone un límite en aquellas aplicaciones que hacen uso de la TFD, especialmente las de tiempo real, dado que para valores mayores a cierto *N* el cálculo podrá resultar demasiado lento y los recursos disponibles podrán ser insuficientes.

Es así que aparece la Transformada Rápida de Fourier (en inglés Fast Fourier Transform, FFT), un algoritmo para el cálculo eficiente de la TFD. Su importancia radica en el hecho que elimina una gran parte de los cálculos repetitivos a los que se ve sometida la TFD, por lo que se logra un cálculo más rápido a menor costo.

El algoritmo de la FFT fue originalmente inventado por Carl Friedrich Gauss en 1805. Diferentes versiones del algoritmo fueron descubiertas a lo largo de los años, pero la FFT no se hizo popular sino hasta 1965, con la publicación de James Cooley y John Tukey, quienes reinventaron el algoritmo al describir como ejecutarlo de forma eficiente en una computadora.

II. DESARROLLO DE LA FFT

La idea básica detrás de la FFT consiste en la división del tiempo, es decir, en la descomposición iterativa en Transformadas de Fourier Discretas más simples. La FFT hace uso de dos propiedades de la Transformada de Fourier Discreta. La FFT presentada asume que *N* es potencia de 2, sin embargo, existen métodos para adaptar otros valores de *N* a las condiciones necesarias de este algoritmo.

Las propiedades que se aprovechan son las siguientes:

Simetría Conjugada Compleja: $W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$ (2)

Periodicidad en n,k: $W_N^{kn} = W_N^{k(N+n)} = W_N^{(k+N)n}$ (3)

La FFT divide la Transformada de Fourier Discreta a calcular en dos TFD menores según la paridad de los términos:

$$X[k] = \sum_{n=0}^{N-1} x[n].W_N^{kn} = \sum_{r=0}^{N/2-1} x[2r].W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1].W_N^{(2r+1)k} \quad r = 0, 1, \dots, \frac{N}{2} - 1$$

$$X[k] = \sum_{r=0}^{N/2-1} x[2r].(W_N^2)^{rk} + W_N^K \sum_{r=0}^{N/2-1} x[2r+1].(W_N^2)^{rk}$$

Sabiendo que $W_N^2 = e^{-j\frac{2\pi}{N}2} = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$ se puede re-expresar la TFD de *N* muestras en la suma de dos TFD de *N/2* muestras.

$$X[k] = \sum_{r=0}^{N/2-1} x[2r].(W_{N/2})^{rk} + W_N^K \sum_{r=0}^{N/2-1} x[2r+1].(W_{N/2})^{rk}$$

Dado que se tratan de dos TFD, esto significa que podemos aplicar el mismo método de división en pares e impares para así obtener dos pares de TFD de *N/4* muestras. El método es así aplicado hasta que se obtienen TFD de 1 muestra, cuyo cálculo resulta sencillo. Una vez obtenidos los valores de las TFD simples, es cuestión de adicionar los resultados.

Se puede observar que si se tenía inicialmente una TFD de *N* muestras, se podrán llevar a cabo $p = log_2 N$ divisiones. Si calculamos el costo de las operaciones que hay que llevar a cabo con este método, se tiene que el algoritmo es de $O(N.log_2 N)$.

III. CONCLUSIONES

La Transformada Rápida de Fourier es un algoritmo para el cálculo de la Transformada Discreta de Fourier basado en la división del tiempo, eliminando así gran parte de los cálculos repetitivos que hay que llevar a cabo si se desea resolver la TFD de forma directa. Si hacemos una comparación del costo de los dos métodos, el cálculo directo de la TFD y la FFT, podemos observar el factor de mejora que brinda la FFT.

Consideremos el ejemplo presentado inicialmente, para $\mathrm{N} = 2^{30}$ teníamos que el tiempo de cálculo total era de 13343 días. Ahora, aplicando la FFT resulta que hay que realizar sólo 30.2^{30} operaciones; asumiendo nuevamente que cada operación tarda 1ns tenemos que el tiempo de cálculo total es de aproximadamente 32 segundos.

N	***N° de operaciones usando cálculo directo (N^2)***	***N° de operaciones usando FFT ($N.log_2N$)***	***Factor de Mejora***
4	8	4	2,0
8	64	12	5,3
16	256	32	8,0
32	1.024	80	12,8
64	4.096	192	21,3
28	16.384	448	36,6
256	65.536	1.024	64,0
512	262.144	2.304	113,8
1.024	1.048.576	5.120	204,8
2^{30}	2^{60}	30×2^{30}	35.791.394,1

Figura 1: Tabla Comparativa que muestra la cantidad de operaciones a realizar con Calculo Directo y con FFT para diversos valores de *N*.

Se puede observar en la Figura 1 que para valores pequeños de *N* no hay gran diferencia entre el costo de ambos método. Sin embargo, el factor de mejora incrementa considerablemente cuando *N* se vuelve grande, esto implica que la FFT puede efectuarse en tiempos razonables a costos razonables en situaciones en las que el cálculo directo se vuelve inviable.

Podemos concluir, que la Transformada Rápida de Fourier beneficia considerablemente a las aplicaciones de procesamiento de señales no sólo de forma genérica al brindar una forma más eficiente que elimina cálculos redundantes, sino también porque permite la resolución de Transformadas de Fourier Discretas para números grandes de muestras en situaciones en las que el método directo no es aplicable. Las FFT es así un algoritmo esencial para aplicaciones sensibles al tiempo y/o de recursos limitados.

REFERENCIAS

[1] "La Transformada Rápida de Fourier (FFT)", [internet], disponible en http://www.ehu.es/Procesadodesenales/tema7/ty3.html, [acceso el 13 de marzo del 2013]

[2] Wikipedia, *La enciclopedia libre*, [internet], disponible en http://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm, [acceso el 13 de marzo del 2013]

[3] Wikipedia, *La enciclopedia libre*, [internet], disponible en http://en.wikipedia.org/wiki/Fast_Fourier_transform#Algorithms, [acceso el 13 de marzo del 2013]

[4] MathWorld, "Fast Fourier Transform", [internet], disponible en http://mathworld.wolfram.com/FastFourierTransform.html, [acceso el 13 de marzo del 2013]

[5] Steven W. Smith, "The Scientist and Engineer Guide to Digital Signal Processing", [internet], disponible en http://www.dspguide.com/ch12.htm, [acceso el 13 de marzo del 2013]