Proyecto de Simulación Computacional de la Transformada de Fourier Discreta en sus Aplicaciones Físicas (Transmisión de Datos)

Semillero de Comunicación de Datos Alexander Graham Bell
Facultad de Ingeniería de Sistemas
Escuela de Ciencias Naturales e Ingenierías
e-mail: {jlievanom, drojasp, hvecino}@unab.edu.co
Mayo de 2005

Resumen

Con la realización de este proyecto se quiere mostrar al público en general por medio de una simulación, cómo se usa y cómo funciona la transformada de Fourier Discreta al ser aplicada en el campo de la transmisión de datos, e igualmente, se presenta su análisis matemático y comportamiento en el campo de aplicación seleccionado. La Transformada de Fourier tiene grandes aplicaciones en campos como la acústica, la óptica, los estudios de antena, el análisis de sistemas lineales, la teoría de probabilidad, la astronomía, la física cuántica, la sísmica, en los problemas de valores limitantes y comunicaciones y en varios más que el proyecto no estudiará. Éste escrito muestra cómo se realiza el análisis de distintas señales digitales y análogas para realizarse filtrado o estimaciones; además, explica cómo este algoritmo es usado en los módems capaces de representar señales digitales usando señales análogas.

Palabras clave:

Filtrado de señales, comunicación de datos, Transformada de Fourier Discreta.

1. Introducción [1]

Jean-Babtiste Joseph Fourier fue un físico y matemático nacido en Francia que desarrolló la serie de Fourier que fue una innovación a la serie de Taylor ya que aplicó la serie a funciones ortogonales en vez de variables. Su descubrimiento fue desarrollado por otros científicos como Gauss, Danielson Lanczos, Runge y König.

Las Transformadas de Fourier son de importancia fundamental en aplicaciones tan variadas como la óptica, acústica, física cuántica, telecomunicaciones, teoría de sistemas y procesamiento de señales, incluyendo el reconocimiento del habla.

Durante años, el avance en estas áreas de conocimiento ha estado limitado por el hecho de que los algoritmos conocidos para calcular las transformadas de Fourier requieren demasiado tiempo de ejecución. Este hecho cambió con el descubrimiento por parte de Cooley y Tukey en 1965 de un algoritmo rápido llamado transformada de Fourier Rápida.

2. Transformada de Fourier

La transformada de Fourier es una generalización de las series complejas de Fourier cuando el límite se denota como $L \to \infty$. Reemplaza el A_n con el $F(k)\,dk$ continuo dejando que la relación entre n y L tienda a k. Luego cambia la suma a una integral y la ecuación se convierte en

$$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi ikx}\, dk \tag{1}$$

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi ikx}\, dx. \tag{2}$$

Aquí,

$$F(k) = \mathcal{F}_x[f(x)](k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi ikx}\, dx \tag{3}$$

Es llamada la Transformada de Fourier adelantada (-i) y

$$f(x) = \mathcal{F}_k^{-1}[F(k)](x) = \int_{-\infty}^{\infty} F(k) e^{2\pi ikx}\, dk \tag{4}$$

Es llamada la transformada de Fourier inversa (+i). [2]

Algunos autores especialmente físicos prefieren escribir la transformada en términos de frecuencia angular

$\omega \equiv 2\pi\nu$ en vez de la frecuencia oscilatoria v. Sin embargo, esto destruye la simetría, resultando en la transformada par

$$H(\omega) = \mathcal{F}[h(t)] = \int_{-\infty}^{\infty} h(t) e^{-i\omega t}\, dt \tag{5}$$

$$h(t) = \mathcal{F}^{-1}[H(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{i\omega t}\, d\omega. \tag{6}$$

Para reestablecer la simetría de la transformada, la convención

$$g(y) = \mathcal{F}[f(t)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-iyt}\, dt \tag{7}$$

$$f(t) = \mathcal{F}^{-1}[g(y)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(y) e^{iyt}\, dy \tag{8}$$

Es usualmente usada. [3]

En general, el par de la transformada de Fourier puede ser definido usando dos constantes arbitrarias a y b como

$$F(\omega) = \sqrt{\frac{|b|}{(2\pi)^{1-a}}} \int_{-\infty}^{\infty} f(t) e^{ib\omega t}\, dt \tag{10}$$

$$f(t) = \sqrt{\frac{|b|}{(2\pi)^{1+a}}} \int_{-\infty}^{\infty} F(\omega) e^{-ib\omega t}\, d\omega. \tag{11}$$

2.1 Convolución

Un convolución es una integral que expresa la cantidad de sobre posicionamiento de una función *g* mientras es dirigida sobre otra función *f*. Este, entonces dobla una función con otra, por ejemplo, en síntesis de imágenes, el mapa sucio medido es una convolución de un verdadero mapa limpio con un destello sucio. La convolución es algunas veces conocido por su nombre alemán faltung ("plegar"). [4]

2.2 Transformada de Fourier Discreta

La transformada de Fourier continua es definida como:

$$f(\nu) = \mathcal{F}_t[f(t)](\nu) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i \nu t}\, dt. \quad (12)$$

Ahora considere una generalización al caso de una función discreta, $f(t) \to f(t_k)$ [5]

dejando que $f_k \equiv f(t_k)$, donde $t_k \equiv k\Delta$, con $k=0, \ldots, N-1$.

Escribiendo esto, se obtiene la transformada de Fourier discreta $F_n = \mathcal{F}_k[\{f_k\}_{k=0}^{N-1}](n)$ como

$$F_n \equiv \sum_{k=0}^{N-1} f_k e^{-2\pi i n k/N}. \quad (13)$$

Las transformadas de Fourier discretas son extremadamente útiles porque ellas revelan periodicidades en datos de entrada al igual que las fortalezas relativas de cualquier componente periódico. Sin embargo, hay unas pocas sutilezas en la interpretación de la transformada de Fourier discreta. En general, la transformada de Fourier discreta de una secuencia real de números será una secuencia de números complejos de la misma longitud. [6]

En particular si f_k son reales, entonces F_{N-n} y F_n están relacionados por $F_{N-n} = \bar{F}_n$, para n=0,1,2,..., N-1 donde $\bar{z}$ denota el conjugado complejo.

Esto significa que el componente Fo es siempre real para datos reales.
Como resultado de la relacion anterior, una función periódica contendrá picos transformados en varios lugares. Esto pasa porque los periodos de los datos de entrada se dividen en compuestos complejos de frecuencia positivos y negativos. [7]

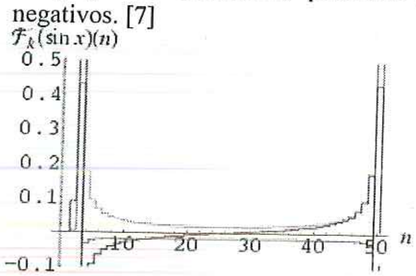

Figura 1. Gráfica F(sinx)(n) vs. n.
Tomada de http://www.mathworld.com

En la figura 1 se puede ver la gráfica cuando se aplica la Transformada de Fourier. En este ejemplo se presenta aliasing.

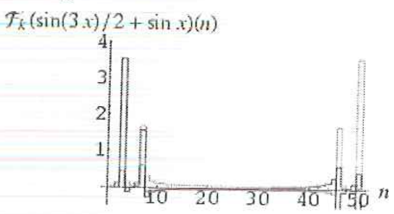

Figura 2. Gráfica F(sin(3x)/2 + sinx)(n) vs. n.
Tomada de http://www.mathworld.com

En la figura 2 se ve una gráfica con la transformada de Fourier aplicada muestral con un N definido.

3. Proyecto

Este proyecto tiene como objeto mostrar las distintas aplicaciones de la Transformada de Fourier Discreta en un

software que sea interactivo para el usuario y que pueda explicar de forma concisa el uso del concepto de la transformada y sus propiedades.

En la realización del proyecto, el recurso de la simulación juega un papel fundamental para el desarrollo del software. Bases de Teoría de Sistemas en cuanto a lo referente a modelos y simulaciones son asistidas en los pilares de la construcción del software y por lo tanto en el desarrollo del proyecto.

La Transformada de Fourier Discreta requiere computacionalmente, que presente un tiempo de ejecución rápido y que su consumo de recursos de la maquina sea mínimo. Es por esto que un algoritmo con estas características es implementado en el software. El algoritmo de la Transformada de Fourier Discreta Rápida (FFT) es conocido por reducir significativamente el retraso en el paso de datos a la transformada. Por lo tanto, la reducción del tiempo de ejecución del programa será lograda con una implementación del algoritmo escrito en el lenguaje asignado.

En cuanto al lenguaje de programación utilizado en el proyecto, se pensó en la interactividad con el usuario y la versatilidad para añadir nuevos componentes al software a medida que se desarrollaba el proyecto.

Para obtener esta versatilidad se utilizó un lenguaje orientado a objetos como Java o C++.

Un desarrollo de entorno gráfico de fácil uso es igual de práctico para ambos lenguajes. Por lo tanto, se usa el lenguaje más robusto por sus características y su habilidad de ofrecer más opciones y herramientas para el programador.

Si es este el caso, Visual C++ .NET sería la herramienta adecuada. Sin embargo, para aplicar el software a un entorno Web, eliminaría la opción de C++, lo cual obligaría a el uso de Java ya sea para crear el proyecto como Applet o como Servlet.

3.1. El Software

Debido a que el objetivo del proyecto es mostrar la aplicación de la Transformada de Fourier Discreta en aplicaciones físicas, el software mostrará distintos eventos simulados donde se apliquen las Transformadas. Un menú que dispone de opciones para ver los distintos eventos se encuentra presente.

Se considera que el entorno gráfico del software es sencillo y que su manejo de igual manera es simple mostrando solamente lo esencial para entender los distintos fenómenos.

El software, por su naturaleza de proyecto, está abierto para mejoramiento y actualización. El uso de clases y módulos fue necesario y es notable la exigencia de un lenguaje orientado a objetos. A medida que se vayan entendiendo nuevos fenómenos que utilicen la transformada de Fourier y que se vea la forma de instaurar una simulación computacional se podrá agregar nuevas actualizaciones al software.

3.2. El algoritmo

El algoritmo base usado en el software es el FFT (Fast Fourier Transform). En el modelo actual del proyecto, la estructura del algoritmo es el siguiente:

Teniendo un arreglo bidimensional A[L][2] , donde A[][0] guarda la parte real y A[][1] guarda la parte imaginaria se tiene:

```
j = 1
Desde i = 1 mientras i < n hacer
  Si i < j
    t_r = A[i-1][0]
    t_i = A[i-1][0]
    A[i-1][0] = A[j-1][0]
    A[i-1][1] = A[j-1][1]
    A[j-1][0] = t_r
    A[j-1][1] = t_i
  k = longitud de A /2
  mientras k < j
    j = j – k
    k = k/2
  j = j + k
  i = i + 1
desde l = 1 mientras l <= ln(n)
  le = ln(l)
  u_r = 1.0
  u_i = 0.0
  w_r = cos(2pi/le)
  w_i = -sen(2pi/le)
  desde j = 1 mientras j <= le/2
    desde i = j mientras i <= longitud A
      ip = i + le/2
      t_r = A[ip-1][0]*u_r-u_i-A[ip-1][1]
      t_i = A[ip-1][1]*u_r-u_i-A[ip-1][0]
      A[ip-1][0] = A[i-1][0] – t_r
      A[ip-1][1] = A[i-1][1] – t_i
      A[i-1][0] = A[i-1][0] + t_r
      A[i-1][1] = A[i-1][1] = t_i
      i = i + 1
    t_r = u_r * w_r – w_i * u_i
    u_i = w_r * u_i + w_i * u_r
    u_r = t_r
    j = j+1
  l = l + 1
```

Este algoritmo devuelve el arreglo A con la transformada de Fourier aplicada. El tamaño L del arreglo debe ser exponencial de 2.

3. Conclusiones

La transformada de Fourier se divide en cuatro secciones, las cuales son la transformada de fourier continua en tiempo y en frecuencia y la transformada de fourier discreta descrita en tiempo y en frecuencia.

Las transformadas continuas son usadas para investigaciones cuantitativas como en química, óptica, acústica y física.

Las transformadas discretas son usadas en Ciencias computacionales aplicadas hacia la comunicación de datos en señales binarias y análogas y su análisis.

4. Autores

Juan Gabriel Liévano Martinez-Villalba, estudiante de la Facultad de Ingeniería de Sistemas de la Universidad Autónoma de Bucaramanga, bachiller bilingüe Colegio La Quinta del Puente de Floridablanca. Biología Molecular, Síntesis de Proteínas, Nanotecnología, Redes Neuronales.

David Enrique Rojas Peralta. Estudiante de la Facultad de Ingeniería de Sistemas de la Universidad Autónoma de Bucaramanga. Bachiller distinguido Colegio San Pedro Claver

de Bucaramanga. Administración Empresarial, Políticas dirigidas a la tecnología.

Hugo Vecino Pico. Docente Auxiliar de la Facultad de Ingeniería de Sistemas, Ingeniero de Sistemas UNAB 2000.

5. Referencias

[1]. BRASSARD, G. y BRATLEY, P. 2000. “Fundamentos de Algoritmia” Prentice Hall.

[2] KRANTZ, S. G. "The Fourier Transform." §15.2 in *Handbook of Complex Variables.* Boston, MA: Birkhäuser, pp. 202-212, 1999.

[3] MATHEWS, J. y WALKER, R. L. *Mathematical Methods of Physics, 2nd ed.* Reading, MA: W. A. Benjamin/Addison-Wesley, 1970.

[4] MORRISON, N. *Introduction to Fourier Analysis.* New York: Wiley, 1994.

[5] SNEDDON, I. N. *Fourier Transforms.* New York: Dover, 1995.

[6] TOLSTOV, G. P. *Fourier Series.* New York: Dover, 1976.

[7]PROAKIS, J.G., y DIMITRIS, G.M. “Tratamiento digital de Señales” Prentice Hall. 2001