

Transformada de Fourier

Gabriel Alexander Valdivia Medina, Giulia Alexa Naval Fernandez, Rodrigo Alonso
Torres Sotomayor

Universidad Católica San Pablo, Arequipa.

Abstract

Este trabajo está orientado en presentar la Transformada de Fourier (TF), explicar su funcionamiento, capacidades y versiones, así como proponer un algoritmo en el lenguaje de programación C++ para su uso práctico. Se implementará una aplicación típica del algoritmo a modo de ejemplificación de lo que se puede lograr con este.

Keywords: Transformada de Fourier, solution, degree of a polynomial, linear, quadratic, quadratic formula, roots, cubics, quartics

1. La Serie de Fourier

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_ix^i$$

2. Transformada de Fourier

La transformada de Fourier nos permite, entre muchas cosas, cambiar la forma en la que miramos una señal o potencia, pasando de mirar su comportamiento en base de tiempo a verla en base a su frecuencia.

Se relaciona con la Serie de Fourier, ya puede representar todas las amplitudes de las curvas encontradas en esta serie.

La transformada se halla con el producto interno entre la señal recibida y una exponencial compleja de frecuencia fija:

$$X(f) = \int_{-\infty}^{\infty} X(t) \cdot e^{-2\pi ift} dt \quad (1)$$

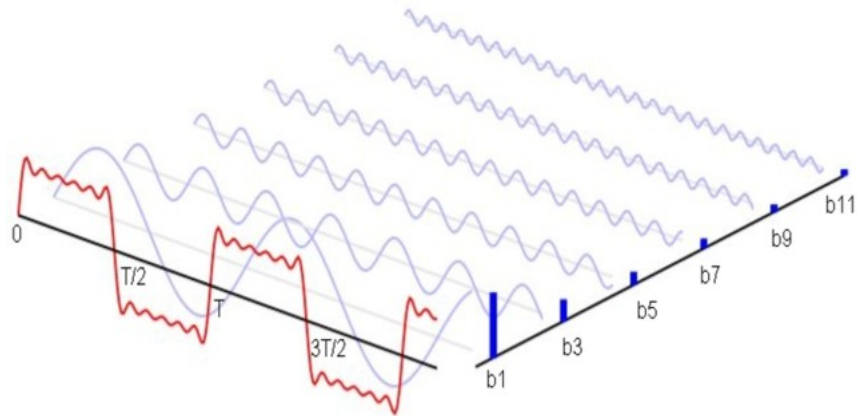


Figura1. En rojo, la señal recibida. En azul intenso, la transformada de Fourier. En morado, las series de Fourier. Fuente: wikimedia commons.

Donde $X(t)$ es la señal en base tiempo definida, $X(f)$ la vista en base frecuencia que se busca. El cambio más común es de tiempo a frecuencia, pero no es el único que la transformada puede manejar.

2.1. Ejemplo 1

Hallaremos la transformada de la siguiente función:

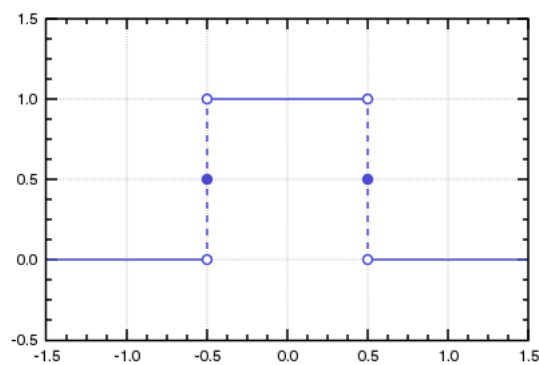


Figura 2. Función rectangular, función cajón, o pulso unitario.

Para hallar la transformada de una función por partes, se hallan las transformadas de

las partes por separado:

$$X(f) = \int_{-\infty}^{-0.5} rect(t) \cdot e^{-2\pi i f t} dt + \int_{-0.5}^{0.5} rect(t) \cdot e^{-2\pi i f t} dt + \int_{0.5}^{\infty} rect(t) \cdot e^{-2\pi i f t} dt$$

Sin embargo, dado que $rect(t)$ es igual a 0 tanto en la primera como en la tercera parte, nos quedamos sólo con la transformada del medio.

$$X(f) = \int_{-0.5}^{0.5} 1 \cdot e^{-2\pi i f t} dt$$

Desarrollamos la integral:

$$X(f) = \left[\frac{e^{-i2\pi f t}}{-i2\pi f} \right]_{-0.5}^{0.5}$$

$$X(f) = \frac{e^{-i\pi f} - e^{i\pi f}}{-2\pi f i}$$

Por teoría de números complejos, sabemos que las funciones *sen* y *cos* se relacionan con las funciones exponenciales de la forma:

$$\cos(\theta) = \frac{1}{2}(e^{+i\theta} + e^{-i\theta}) \quad (2)$$

$$\sin(\theta) = \frac{1}{2i}(e^{+i\theta} - e^{-i\theta}) \quad (3)$$

Entonces, podemos acomodar nuestra ecuación para que satisfaga alguna:

$$X(f) = \frac{-e^{-i\pi f} + e^{i\pi f}}{2i} \cdot \frac{1}{\pi f}$$

$$X(f) = \frac{\sin(\pi f)}{\pi f}$$

$$X(f) = \text{sinc}(\pi f)$$

3. Transformada Inversa de Fourier

La transformada inversa de Fourier, básicamente, revierte el estado en que se encuentra la representación de una onda luego de aplicarle la Transformada de Fourier normal.

De forma más específica, convierte una serie de tamaño potencia de 2 de números complejos, puntos en el espectro de frecuencia, en una serie del mismo tamaño en un dominio de tiempo.

4. Algoritmo Cooley-Tukey

Pensando en formas de implementar y mejorar la TFD, James W. Cooley y John W. Tukey publicaron en su paper insigne en 1965 una implementación de la transformada utilizando el paradigma "divide y vencerás", denominada como Transformada Rápida de Fourier. En dicho trabajo, se destaca la ventaja que presenta el algoritmo en arreglos de tamaño N potencia de 2 sobre otros con N distintos, al lograr una complejidad de $O(N \log(N))$ para el primer caso.

Este algoritmo consigue reducir el número de cálculos al dividir el arreglo de tamaño N del polinomio en 2 de tamaño $N/2$, uno con el contenido de los índices impares y el otro con el de los índices pares.

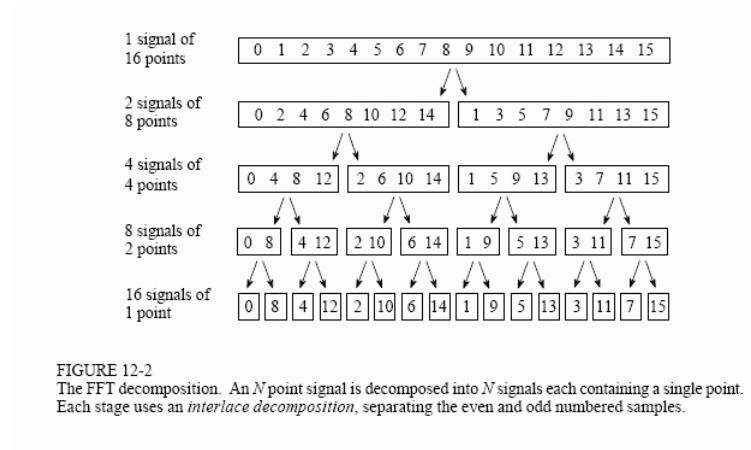


Figura3. Descomposición de arreglo de coeficientes del polinomio. Fuente: The Scientist and Engineer's Guide to Digital Signal Processing.

Esta distribución también se puede lograr a través usando inversión de bits.


Sample numbers in normal order			Sample numbers after bit reversal	
<i>Decimal</i>	<i>Binary</i>		<i>Decimal</i>	<i>Binary</i>
0	0000		0	0000
1	0001		8	1000
2	0010		4	0100
3	0011		12	1100
4	0100		2	0010
5	0101		10	1010
6	0110		6	0100
7	0111		14	1110
8	1000		1	0001
9	1001		9	1001
10	1010		5	0101
11	1011		13	1101
12	1100		3	0011
13	1101		11	1011
14	1110		7	0111
15	1111		15	1111

FIGURE 12-3
The FFT bit reversal sorting. The FFT time domain decomposition can be implemented by sorting the samples according to bit reversed order.

Figura4. Ordenamiento por inversión de bits. Fuente: The Scientist and Engineer's Guide to Digital Signal Processing.

Una vez dividido el arreglo en partes más pequeñas, el algoritmo busca encontrar su valor en el espectro de frecuencia, donde obviamente, la frecuencia de las divisiones más pequeñas, es decir de 1 solo punto, será igual a sí mismo. Luego de esto, tendrá que regresar y juntar los resultados en el orden inverso al que se hicieron las divisiones. Para sumar los puntos de cada una, también se tienen que diluir la señal completando con 0s las posiciones pares o impares, dependiendo del sub-arreglo, lo que equivale a una duplicación de la data de la señal en el dominio de la frecuencia. Esta discordancia en la separación con 0s, o de forma más precisa, este corrimiento del segundo sub-arreglo corresponde a la multiplicación del mismo por un senoide.

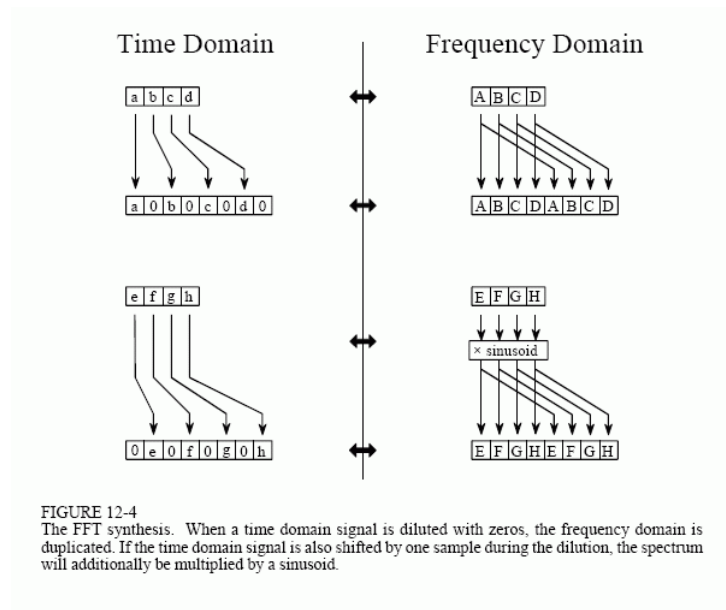


Figura5. Separación de puntos con 0s y multiplicación por sinusoides. Fuente: The Scientist and Engineer's Guide to Digital Signal Processing.

Esta forma de sumar los puntos del espectro mientras se los reordena, tiene el nombre de **mariposa** y es la operación base de toda la TFR.

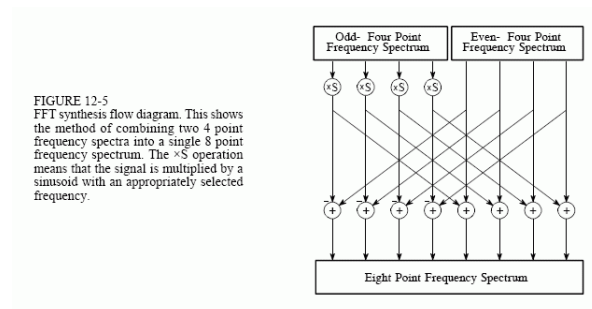


Figura6. Flujo cruzado de operaciones o *mariposas*. Fuente: The Scientist and Engineer's Guide to Digital Signal Processing.

5. Algoritmo ...

6. Comparación de algoritmos

7. Aplicación Práctica: manipulación de audios usando la transformada de Fourier

8. Aplicación Práctica: procesamiento de imágenes usando la transformada de Fourier

Apéndice

A field is a set on which the binary operations $+$, $-$, \times and \div are defined. Besides integral fields, there are other fields where polynomials will behave differently and a polynomial with finite terms and of a finite degree can also have infinite solutions.

Bibliography

<http://www.ijsea.com/archive/volume2/issue7/IJSEA02071002.pdf>
http://people.scs.carleton.ca/~maheshwa/courses/5703COMP/16Fall/FFT_Report.pdf