

Universidad Nacional Autónoma de Honduras

Tesis presentada previo a obtener el grado de
Master en Física



Algoritmo básico de la transformada rápida de Fourier. Un análisis usando código Python

Presentada por:
Michael Joel Spilsbury Fuentes

Asesor de Tesis:
Armando Euceda, Ph.D.

Departamento de Física
San Pedro Sula, Honduras

mayo, 2016

Autoridades de la Universidad Nacional Autónoma de Honduras

Junta de Dirección Universitaria:

MSc. Aleyda Lizett Romero Escobar

Lic. José Manuel Torres Rubí

Abg. Martha Arguijo Bertrand

MSc. Melba Esperanza Baltodano Molina

Lic. Juan Carlos Ramírez

Dr. Valerio Gutiérrez López

Dr. Ramón Antonio Romero Cantero

Rectoría UNAH:

Lic. Julieta Castellanos

Vicerrectoría Académica UNAH:

Dra. Rutilia Calderón Padilla

Vicerrectoría Asuntos Estudiantiles UNAH:

Abg. Ayax Lempira Irías Coello

Vicerrectoría Relaciones Internacionales UNAH:

MSc. Julio Cesar Raudales Torres

Secretaría General UNAH:

Abg. Emma Virginia Rivera

Decano Facultad de Ciencias UNAH:

Dr. Nabil Kawas

Director Maestría en Física UNAH:

Dr. Armando Euceda

Director Escuela de Física UNAH:

MSc. Alejandro Galo

Dedicada a

Toda persona que influyo en mi desarrollo personal.

Algoritmo básico de la transformada rápida de Fourier. Un análisis usando código Python

Por: Michael Joel Spilsbury Fuentes
Asesor de tesis: Armando Euceda, Ph.D.

Presentada para obtener el grado de Master en Física
Septiembre del 2015

Resumen

El tema central de esta tesis se enfoca en el análisis del algoritmo originalmente propuesto por James Cooley y John Tukey, del cual se origina la Transforma Rápida de Fourier, esperando que contribuya como una herramienta computacional para el estudio de dicho tema en la UNAH. Primero hablaremos de la Transformada de Fourier, pasando por la Transformada Discreta de Fourier y terminar con la Transformada Rápida de Fourier, tomando como base el artículo publicado por James W. Cooley y John W. Tukey, el cual después de su publicación en 1965 sigue teniendo mucha popularidad. Como lenguaje de programación se usará Python, que es un lenguaje de programación multiparadigma, esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Los usuarios de Python se refieren a menudo a la filosofía Python que es bastante análoga a la filosofía de Unix. Al mismo tiempo se integrará un módulo de Fortran para mejorar el desempeño.

Palabras clave: *Transformada Rápida de Fourier, Python, Fortran.*

Declaración

El trabajo en esta tesis se basa en la investigación llevada por mi persona, para la acreditación del grado de Master en Física. Ninguna parte de esta tesis se ha presentado en otro lugar para cualquier otro grado o título y todo es mi propio trabajo, a menos que se haga referencia de lo contrario en el texto.

Derechos Reservados © 2015 por Michael Spilsbury.

“Los derechos de autor de esta tesis descansan sobre el autor. No se debe hacer citas de la misma sin el consentimiento previo y por escrito del autor y la información derivada de ella debe ser reconocida”.

Reconocimientos

A cada una de las personas que ayudo en el desarrollo de este trabajo; aportando información, ideas, mejoras, apoyo moral, innovación y guía.

Índice general

Resumen	IV
Declaración	v
Reconocimientos	VI
Índice de figuras	x
Índice de tablas	xii
1 Introducción	1
2 Revisión de Literatura	5
2.1 Transformada de Fourier	5
2.1.1 Integral de Fourier	5
2.1.2 Propiedades de la transformada de Fourier	8
2.1.3 Convolución	11
2.2 Serie de Fourier	12
2.2.1 Serie de Fourier, caso particular de la integral de Fourier	13
2.2.2 Muestreo de funciones	16
2.2.3 Teorema de muestreo	20
2.3 Transformada discreta de Fourier	24
2.3.1 Estudio gráfico	25
2.3.2 Estudio teórico	28
2.3.3 Transformada inversa discreta de Fourier	33

2.3.4	Propiedades de la transformada discreta de Fourier	34
2.4	Origen de la transformada rápida de Fourier (FFT)	38
2.5	Enfoque de Brigham para la FFT	38
2.5.1	Formulación matricial	39
2.5.2	Desarrollo intuitivo	40
2.5.3	Diagrama de flujo de señal	44
2.5.4	Nodos duales	47
2.5.5	Determinación de W^p	49
2.5.6	Reorganizando la FFT	50
2.5.7	Desarrollo teórico del algoritmo FFT de base-2	52
3	Diseño del programa en Python para cálculos de FFT	56
3.1	Contexto	56
3.2	Programa FFT	57
3.3	Metodología para el uso del programa	62
3.4	Taller de validación del programa	63
4	Análisis de resultados del programa	66
4.1	Datos	66
	Bibliografía	68
	Apéndice	71
A	Función impulso: una distribución	71
A.1	Definición	71
A.2	El concepto de distribución	73
A.3	Propiedades de la función impulso	73
A.4	Prueba de la formula de inversión	76
B	Evaluación gráfica de la integral de convolución	77

B.1	Evaluación gráfica	78
B.2	Convolución que contiene funciones impulso	81
C	Tabla de pares de transformadas	86
D	Tabla de propiedades	91
E	Diagrama de flujo del programa FFT	94
E.1	Inversión de bit	94

Índice de figuras

Figura 1.1	Interpretación de la transformada de Fourier	3
Figura 2.1	Desarrollo gráfico del teorema de convolución para la transformada de Fourier de una función triangular periódica	14
Figura 2.2	Desarrollo gráfico del teorema de frecuencia-convolución para la transformada de Fourier de una función muestreada	17
Figura 2.3	Efecto de aliasing en la transformada de Fourier muestreada a una tasa insuficiente	19
Figura 2.4	Transformada de Fourier de una función muestreada a la tasa de Nyquist	21
Figura 2.5	Deducción gráfica del teorema de muestreo	23
Figura 2.6	Desarrollo gráfico de la transformada discreta de Fourier	26
Figura 2.7	Derivación gráfica del par transformado discreto de Fourier	29
Figura 2.8	Relación de multiplicaciones entre el cálculo directo y la FFT	43
Figura 2.9	Diagrama de flujo de señal para la FFT, $N = 4$	45
Figura 2.10	Ejemplo de nodos duales	46
Figura 2.11	Ejemplo de nodos a omitir	48
Figura 2.12	Ejemplo de la operación de inversión de bit para $N = 16$	51
Figura 3.1	Gráfico de la función $A\text{sen}(2\pi f_0 t)$ y su transformada. $A = 1$ y $f_0 = 250$ Hz	61
Figura 3.2	Gráfico de la función $A\text{cos}(2\pi f_0 t)$ y su transformada. $A = 1$ y $f_0 = 250$ Hz	61
Figura 3.3	Gráfico de $e^{-\alpha t}$ para $t > 0$ y su transformada. $\alpha = 10$ Hz	62

Figura 3.4	Módulo para interpretar el comportamiento de la transformada de Fourier	63
Figura 4.1	Comparación gráfica de los tres métodos para la FFT	67
Figura A.1	Representación de la función impulso en el límite cuando $a \rightarrow 0$. . .	72
Figura B.1	Ilustración de los pasos para la convolución	79
Figura B.2	Ejemplo gráfico de la convolución; $t_1 = T_0/2$	80
Figura B.3	Ilustración de la convolución con funciones impulso	82
Figura B.4	Simulación de convolución, dos pulsos cuadrados	83
Figura B.5	Simulación de convolución, pulso gaussiano y pulso cuadrado	84
Figura B.6	Vista HTML de la simulación de convolución	85
Figura E.1	Diagrama de flujo del programa FFT	96

Índice de tablas

Tabla C.1	Pares de transformadas de Fourier	87
Tabla D.1	Propiedades de la transformada de Fourier de funciones complejas .	91
Tabla D.2	Propiedades de la transformada de Fourier	92
Tabla D.3	Propiedades de la transformada de Fourier continua y discreta . . .	93

Capítulo 1

Introducción

La *transformada rápida de Fourier*, FFT por sus siglas en inglés (Fast Fourier Transform), es un algoritmo para el cálculo de la transformada discreta de Fourier de una forma eficiente en cuanto a la cantidad de operaciones realizadas y tiempo de procesamiento. Tal como lo expone Brigham [5], la FFT es una herramienta fundamental para la solución de problemas en los sectores educacional, industrial y militar. Desde 1965 [9] el uso de la FFT se ha expandido rápidamente y las computadoras personales han impulsado una explosión de aplicaciones adicionales de la FFT. Algunos ejemplos de la aplicación de la FFT son los siguientes:

- Diseño de circuitos.
- Espectroscopia.
- Cristalografía.
- Procesamiento de señales y comunicaciones.
- Imágenes.

La FFT es entonces una forma de calcular eficientemente la transformada discreta de Fourier, está última como un caso discreto de la forma continua, la transformada de Fourier; ya que el procesamiento digital tiene recursos limitados de memoria y capacidad

de cómputo y desde este punto de vista no se puede pensar en procesar la secuencia infinita de los datos continuos propios del tiempo y la frecuencia.

El análisis de Fourier fue originalmente concebido con la representación y análisis de fenómenos periódicos mediante la *serie de Fourier*, y luego fueron extendidas estas ideas a los fenómenos no periódicos vía la *transformada de Fourier*. De hecho, una forma de obtener de la serie de Fourier la transformada de Fourier es considerar a los fenómenos no periódicos (y en general cualquier función) como un caso límite del fenómeno periódico cuando el periodo tiende a infinito. Un conjunto discreto de frecuencias en el caso periódico se vuelve un continuo de frecuencias en el caso no periódico, el *espectro* ha nacido, y con ello viene un importante principio:

“Cada señal tiene un espectro y esta determinada por su espectro. Puede analizarse dicha señal en el dominio del tiempo (o espacial) o en el de la frecuencia.” [22]

Además también es conocido que una forma de onda periódica, tal como una onda acústica asociada con una prolongada nota de un instrumento musical esta compuesta por una frecuencia fundamental y armónicos. Pero como expone Bracewell [3], la serie de Fourier es un caso particular de la transformada de Fourier (simplemente usando impulsos para generar el espectro discreto), por lo tanto partiremos analizando la transformada de Fourier.

Una interpretación sencilla es la que se muestra en la Fig. 1.1. En esencia es descomponer o separar la onda original en una suma de sinusoides de diferentes frecuencias y amplitudes, en la Fig. 1.1 $s(t)$ se descompone en $s_1(t)$ y $s_2(t)$. Si estas sinusoides al sumarse producen la onda original, es decir $s_1(t) + s_2(t) = s(t)$, entonces se ha determinado la transformada de Fourier de la misma. La representación gráfica de la transformada de Fourier es un diagrama que muestra la amplitud y frecuencia de cada una de las sinusoides.

La transformada de Fourier identifica o distingue entre las diferentes frecuencias de las sinusoides (y sus amplitudes) que se combinan para generar una forma de onda arbitraria.

Matemáticamente, esta relación esta dada por

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft} dt \tag{1.0.1}$$

donde $s(t)$ es la forma de onda a ser descompuesta en la suma de sinusoides, $S(f)$ es la transformada de Fourier de $s(t)$, y $j = \sqrt{-1}$.

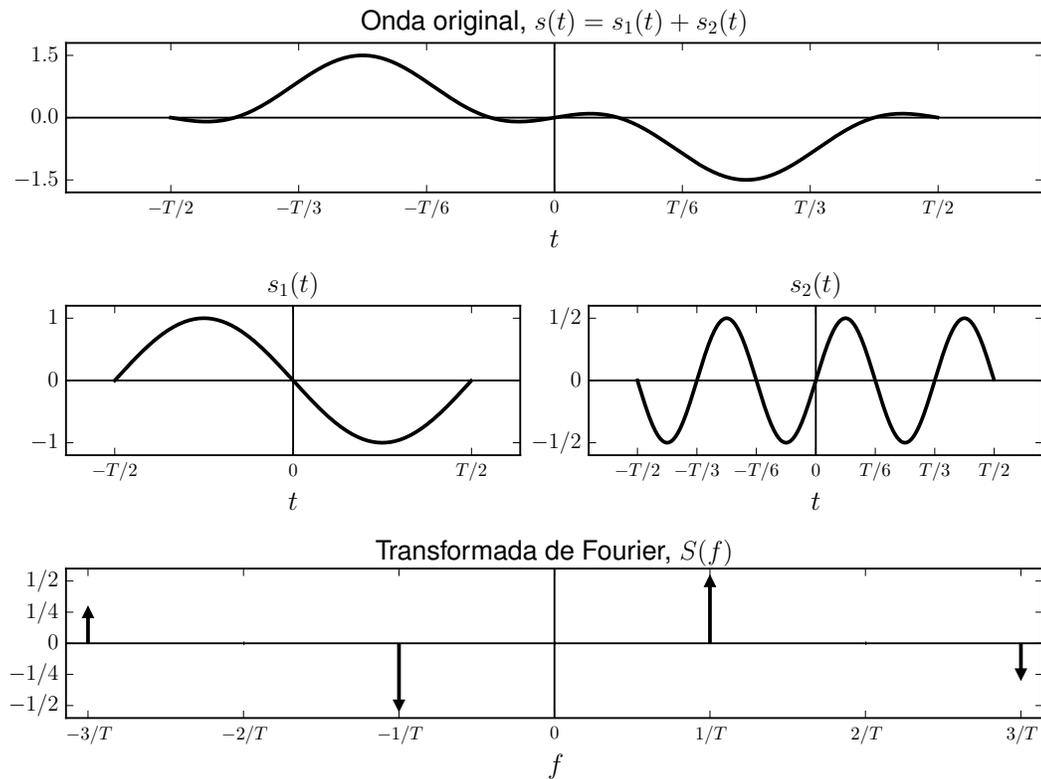


Figura 1.1: Interpretación de la transformada de Fourier¹

También, debido a la gran cantidad de problemas que pueden ser resueltos por medio de la transformada de Fourier, era de esperarse que se extendiera al análisis por medio de computadoras. La integración numérica de la Ec. (1.0.1) esta definida por:

$$S(f_k) = \sum_{i=0}^{N-1} s(t_i)e^{-j2\pi f_k t_i} (t_{i+1} - t_i) \tag{1.0.2}$$

¹Cabe mencionar que la transformada de Fourier es una función compleja y en la Fig. 1.1 la transformada es imaginaria pura como se verá mas adelante en las propiedades.

Para los problemas que no tienen una solución exacta, la transformada discreta de Fourier, Ec. (1.0.2), es de gran ayuda. Revisando detenidamente la Ec. (1.0.2), indica que si hay N datos de la función $s(t_i)$ y si se desea determinar la amplitud de cada una de las N sinusoides, entonces el tiempo de computo es proporcional a N^2 , el número de multiplicaciones. Esto ofrece un problema ya que para un valor elevado de N , el tiempo de computo sería excesivo, a lo cual, como se menciona en un inicio y es el tema principal de esta tesis, la FFT ofrece una reducción del tiempo de computo, siendo este proporcional a $N \log_2 N$.

En el siguiente capítulo se abordará la definición de la transformada de Fourier así como la definición de las propiedades más relevantes para el análisis discreto. Ya que la función impulso es de gran interés, se dedica el Apéndice A para el estudio de esta singularidad de gran importancia.

Capítulo 2

Revisión de Literatura

2.1. Transformada de Fourier

La Transformada de Fourier es una transformación lineal, que provee técnicas para resolver problemas en sistemas lineales. Las transformaciones lineales son usadas como una herramienta física o matemática para alterar el problema en uno que pueda ser resuelto.

La transformada de Fourier juega un papel importante en muchas ramas de la ciencia. Mientras puede ser considerada puramente como un funcional matemático, como es usual con otras transformadas, asume en muchos campos un definido significado físico, como el de las funciones de las que se deriva. Una forma de onda —óptica, eléctrica o acústica— y su espectro son considerados igualmente como entidades físicas observables y medibles: un osciloscopio nos permite ver una forma de onda eléctrica y un espectroscopio o analizador de espectro nos permite ver el espectro óptico o eléctrico. Nuestra apreciación acústica es aún más directa, ya que el oído percibe el espectro audible [3].

2.1.1. Integral de Fourier

La integral de Fourier está definida por la expresión

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{-j2\pi ft} dt \quad (2.1.1)$$

Si la integral existe para todo valor del parámetro f , entonces la Ec. (2.1.1) define la transformada de Fourier $H(f)$, de la función $h(t)$. La transformada de Fourier convierte una función continua en el dominio del tiempo en una función continua en el dominio de la frecuencia, por lo que genera un espectro continuo de frecuencias [6]. La existencia o no de la transformada de Fourier puede pasarse por alto si la función a transformar es exactamente la descripción de una cantidad física. Los eventos físicamente posibles son una condición suficiente para la existencia de una transformada [3].

Al final, la transformada de Fourier es una función compleja, por lo que se podría expresar como:

$$H(f) = R(f) + jI(f) = |H(f)|e^{j\theta(f)} \quad (2.1.2)$$

donde $|H(f)|$ es el *espectro de Fourier* de $f(t)$, $|H(f)|^2$ es el *espectro de energía* y $\theta(f)$ es el *ángulo de fase*.

De la misma forma, se define la *transforma inversa de Fourier* como:

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{j2\pi ft} df \quad (2.1.3)$$

Si las funciones $h(t)$ y $H(f)$ están relacionadas por las Ec. (2.1.1) y (2.1.3), éstas se llaman *par transformado de Fourier* y se indicará esta relación por la notación:

$$h(t) \Leftrightarrow H(f) \quad (2.1.4)$$

En algunos textos se puede encontrar otras representaciones de las Ec. (2.1.1) y (2.1.3), siendo estas de forma general:

$$H(\omega) = a_1 \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt, \quad \omega = 2\pi f \quad (2.1.5)$$

$$h(t) = a_2 \int_{-\infty}^{\infty} H(\omega)e^{j\omega t} d\omega \quad (2.1.6)$$

donde se debe cumplir que $a_1 a_2 = 1/2\pi$.

En sentido estricto, la validez de las Ec. (2.1.1) y (2.1.3), es bajo la consideración que todas las funciones serán de *variación acotada*, significando esto que pueden ser representadas por una curva de longitud finita en cualquier intervalo de longitud finita. De lo expresado por Papoulis [23], se detallan las siguientes condiciones:

Condición 1. Si $h(t)$ es absolutamente integrable, es decir

$$\int_{-\infty}^{\infty} |h(t)| dt < \infty \quad (2.1.7)$$

entonces su integral de Fourier $H(f)$ existe y satisface (2.1.3).

Es importante considerar que (2.1.7) es una condición suficiente pero no necesaria para la existencia de la transformada de Fourier, ya que existen funciones que no cumplen la *Condición 1* pero tienen una transformada que satisface (2.1.3). Un ejemplo de esto último es la función $\text{sen } at/at$, incluida en la *Condición 2*.

Condición 2. Si $h(t) = g(t) \text{sen}(2\pi ft + \alpha)$, donde f y α son constantes arbitrarias, y si, para $|t| > \lambda > 0$, la función $h(t)/t$ cumple con (2.1.7) y $g(t)$ es monótonamente decreciente, entonces $H(f)$ existe y satisface (2.1.3).

Finalmente se hace énfasis, que permitiendo que $h(t)$ y $H(f)$ incluyan funciones de singularidad (p. ej. los impulsos), se desarrollará en el Apéndice A un análisis de la teoría de distribución, ya que éstas tienen aplicaciones importantes pero no satisfacen las *condiciones 1 y 2*. Es de gran utilidad desarrollar la transformada de Fourier de funciones impulso, ya que simplifican en gran medida la derivación de muchos pares de transformadas.

La función impulso $\delta(t)$ esta definida como [Ec. (A.7)]

$$\int_{-\infty}^{\infty} \delta(t - t_0)x(t)dt = x(t_0) \quad (2.1.8)$$

siendo $x(t)$ una función arbitraria continua en t_0 . La aplicación de la definición (2.1.8) da de forma directa la transformada de Fourier de muchas funciones y como ejemplo de su utilidad se puede comprobar la fórmula de la transformada inversa de Fourier. (ver Apéndice A)

En el Apéndice C se muestra una tabla, con ecuaciones y gráficos cualitativos, de los pares de transformadas frecuentemente encontrados.

2.1.2. Propiedades de la transformada de Fourier

A continuación se presenta una pequeña lista de propiedades simples pero básicas para el entendimiento de la transformada de Fourier, estas pueden ser fácilmente deducidas de la definición del par transformado, Ecs. (2.1.1) y (2.1.3). Se asume que todas las funciones en consideración tienen transformada de Fourier.

Linealidad. Si $X(f)$ y $Y(f)$ son las transformadas de Fourier de $x(t)$ y $y(t)$ respectivamente, y sean c_1 y c_2 constantes arbitrarias, entonces

$$c_1x(t) + c_2y(t) \Leftrightarrow c_1X(f) + c_2Y(f) \quad (2.1.9)$$

Es importante este par transformado debido a que refleja que la transformada de Fourier es aplicable a análisis de sistemas lineales. También es válido para sumas finitas, pero la extensión a sumas infinitas no siempre es cierta ya que el intercambio de la integral con la sumatoria es posible solo si ciertas condiciones son cumplidas.

Simetría. Si $h(t)$ y $H(f)$ son el par transformado, entonces

$$H(t) \Leftrightarrow h(-f) \quad (2.1.10)$$

Esta propiedad es una afirmación del carácter simétrico de las Ecs. (2.1.1) y (2.1.3).

Escalamiento en tiempo y frecuencia. Si $h(t)$ y $H(f)$ son un par transformado, y sea k una constante real mayor que cero, entonces los siguientes pares de transformadas definen el *escalamiento*

$$h(kt) \Leftrightarrow \frac{1}{|k|} H\left(\frac{f}{k}\right), \quad \text{en el tiempo} \quad (2.1.11)$$

$$\frac{1}{|k|} h\left(\frac{t}{k}\right) \Leftrightarrow H(kf), \quad \text{en la frecuencia} \quad (2.1.12)$$

Cuando se considera el escalamiento de una función impulso, se debe recordar la Ec. (A.9)

$$\delta(ax) = \frac{1}{|a|} \delta(x)$$

Este comportamiento es muy utilizado en las aplicaciones de espectro y formas de onda (radares y antenas), donde una compresión en la escala del tiempo corresponde a una expansión en la escala de la frecuencia. También, cuando un elemento del par transformado se expande horizontalmente, el otro no solo se contrae horizontalmente, sino que también crece verticalmente de tal forma que se mantiene constante el área debajo de la curva.

Desplazamiento en tiempo y frecuencia. Si $h(t)$ es desplazada en el tiempo por una constante t_0 , entonces su transformada de Fourier es

$$h(t - t_0) \Leftrightarrow H(f)e^{-j2\pi ft_0} \quad (2.1.13)$$

Por otro lado, si $H(f)$ es desplazada en la frecuencia por una constante f_0 , entonces su transformada inversa de Fourier es

$$h(t)e^{j2\pi t f_0} \Leftrightarrow H(f - f_0) \quad (2.1.14)$$

Un ejemplo del desplazamiento en la frecuencia es la *modulación*, que consiste en multiplicar una señal $h(t)$ por un coseno cuya frecuencia está determinada por la frecuencia de desplazamiento f_0 .

Fórmula de inversión alternativa. La Ec. (2.1.3) (transformada inversa de Fourier) se puede escribir alternativamente como

$$h(t) = \left[\int_{-\infty}^{\infty} H^*(f)e^{-j2\pi ft} df \right]^* \quad (2.1.15)$$

donde $H^*(f)$ es el conjugado de $H(f)$. La importancia de este resultado se debe a que ahora tanto la transformada de Fourier como su inversa tienen el término común $e^{-j2\pi ft}$, lo cual es de mucha utilidad en el desarrollo de los programas de FFT.

Propiedades de las funciones pares e impares. Si $h_e(t)$ es una función par, o sea que $h_e(t) = h_e(-t)$, entonces la transformada de Fourier de $h_e(t)$ es una función par y real

$$h_e(t) \Leftrightarrow R_e(f) = \int_{-\infty}^{\infty} h_e(t) \cos(2\pi ft) dt \quad (2.1.16)$$

y de la misma forma, si $H(f)$ es una función real y par, su transformada inversa es par y real.

Por otro lado, si $h_o(t)$ es una función impar, $h_o(t) = -h_o(-t)$, su transformada de Fourier es una función impar e imaginaria

$$h_o(t) \Leftrightarrow -jI_o(f) = -j \int_{-\infty}^{\infty} h_o(t) \operatorname{sen}(2\pi ft) dt \quad (2.1.17)$$

y si $H(f)$ es una función impar e imaginaria, su transformada inversa es impar y real.

Cualquier función puede ser descompuesta o separada en la suma de una función par y otra impar:

$$\begin{aligned} h(t) &= \frac{h(t)}{2} + \frac{h(t)}{2} \\ &= \left[\frac{h(t)}{2} + \frac{h(-t)}{2} \right] + \left[\frac{h(t)}{2} - \frac{h(-t)}{2} \right] \\ &= h_e(t) + h_o(t) \end{aligned} \quad (2.1.18)$$

De las Ecs. (2.1.16) y (2.1.17), la transformada de Fourier de la Ec. (2.1.18) es

$$H(f) = R(f) + jI(f) = H_e(f) + H_o(f) \quad (2.1.19)$$

donde $H_e(f) = R(f)$ y $H_o(f) = jI(f)$. La Ec. (2.1.19) es útil ya que puede incrementar la velocidad de cálculo de la FFT.

Función de tiempo como variable compleja. La transformada de Fourier, Ec. (2.1.1), su inversa, Ec. (2.1.3), y sus propiedades son válidas para el caso que $h(t)$ sea una función compleja. Teniendo

$$h(t) = h_r(t) + jh_i(t) \quad (2.1.20)$$

las siguientes relaciones resultan del análisis de Fourier:

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{-j2\pi ft} dt = R(f) + jI(f) \quad (2.1.21)$$

$$R(f) = \int_{-\infty}^{\infty} [h_r(t) \cos(2\pi ft) + h_i(t) \operatorname{sen}(2\pi ft)] dt \quad (2.1.22)$$

$$I(f) = - \int_{-\infty}^{\infty} [h_r(t) \operatorname{sen}(2\pi ft) - h_i(t) \cos(2\pi ft)] dt \quad (2.1.23)$$

$$h_r(t) = \int_{-\infty}^{\infty} [R(f) \cos(2\pi ft) - I(f) \operatorname{sen}(2\pi ft)] df \quad (2.1.24)$$

$$h_i(t) = \int_{-\infty}^{\infty} [R(f) \operatorname{sen}(2\pi ft) + I(f) \cos(2\pi ft)] df \quad (2.1.25)$$

En el Apéndice D se encuentran las tablas que listan estas propiedades.

2.1.3. Convolución

La convolución es una propiedad de la transformada de Fourier, cuya importancia va más allá de las propiedades anteriormente vistas. A continuación se detalla su análisis.

La convolución de dos funciones es un concepto físico significativo en muchos campos de la ciencia. Sin embargo, como muchas relaciones matemáticas importantes, la integral de convolución no se da a conocer fácilmente, así como también su verdadera importancia.

La integral de convolución está dada por

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau = x(t) * h(t) \quad (2.1.26)$$

Alternativamente la convolución también puede ser expresada como

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau = h(t) * x(t) \quad (2.1.27)$$

La función $y(t)$ es llamada la *convolución* de las funciones $x(t)$ y $h(t)$. Nótese que es muy difícil *visualizar* la operación matemática en las Ecs. (2.1.26) y (2.1.27). En el Apéndice B se detallará el verdadero significado de la convolución con un análisis gráfico.

Si la función $h(t) = \delta(t - T)$ es la función impulso desplazada en el tiempo, la convolución da como resultado, tomando la Ec. (2.1.27)

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} \delta(\tau - T)x(t - \tau)d\tau \\ &= x(t - T) \end{aligned} \quad (2.1.28)$$

recordando (A.7).

Conociendo la convolución, se puede encontrar la transformada de Fourier de esta, que comúnmente es conocida como *teorema de convolución en el tiempo* o *teorema de convolución en la frecuencia*, y respectivamente se enuncian

$$h(t) * x(t) \Leftrightarrow H(f)X(f) \quad (2.1.29)$$

$$h(t)x(t) \Leftrightarrow H(f) * X(f) \quad (2.1.30)$$

la prueba de estas últimas ecuaciones también se puede ver en el Apéndice B.

2.2. Serie de Fourier

Sabemos de la teoría de *ajuste de curvas* que se puede aproximar la ecuación de una curva mediante la combinación lineal de los monomios $1, x, x^2, \dots, x^m$; y además, como lo sugiere la teoría de Fourier, mediante una combinación lineal de funciones trigonométricas con términos $1, \cos x, \cos 2x, \dots, \cos mx, \sin x, \sin 2x, \dots, \sin nx$ [6].

Debido a la gran variedad de sistemas que oscilan o vibran, es de esperarse que debido a la naturaleza de las funciones trigonométricas sean tomadas en cuenta en la descripción y análisis de dichos problemas. Ahora si analizamos una onda de sonido producida por un instrumento musical, sabemos que está asociada con una frecuencia fundamental y sus armónicos, y además la energía se concentra en estas frecuencias (múltiplos enteros de la frecuencia fundamental). Todo esto se incluye en el análisis expuesto en la transformada de Fourier. Pero si hablamos estrictamente de periodicidad, algo físicamente imposible, claramente nos conduce a un espectro discreto, el cuál puede lograrse por medio de un tren de impulsos periódicos y con esto ligamos el muestreo asociado al espectro armónico.

La serie de Fourier será definida como un caso particular extremo de la transformada de Fourier.

Una función periódica $f(t)$ con período T_0

$$f(t + T_0) = f(t) \quad (2.2.1)$$

representada como una serie de Fourier esta dada por la expresión

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(2\pi n f_0 t) + b_n \sen(2\pi n f_0 t)] \quad (2.2.2)$$

donde f_0 es la frecuencia fundamental igual a $1/T_0$. La magnitud de los coeficientes esta dada por las integrales

$$a_n = \frac{2}{T_0} \int_{-T_0/2}^{T_0/2} f(t) \cos(2\pi n f_0 t) dt, \quad n = 0, 1, 2, 3, \dots \quad (2.2.3)$$

$$b_n = \frac{2}{T_0} \int_{-T_0/2}^{T_0/2} f(t) \sen(2\pi n f_0 t) dt, \quad n = 1, 2, 3, \dots \quad (2.2.4)$$

Utilizando la identidad Euler y realizando una serie de manipulaciones en los resultados, se puede escribir la serie de Fourier, Ec. (2.2.2), en forma exponencial como

$$f(t) = \sum_{n=-\infty}^{\infty} \alpha_n e^{j2\pi n f_0 t} \quad (2.2.5)$$

y los coeficientes α_n , en general, son complejos

$$\alpha_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} f(t) e^{-j2\pi n f_0 t} dt, \quad n = 0, \pm 1, \pm 2, \dots \quad (2.2.6)$$

La expresión de la serie de Fourier en forma exponencial, Eq. (2.2.5), y sus coeficientes complejos, Eq. (2.2.6), son los enfoques preferidos en el análisis.

2.2.1. Serie de Fourier, caso particular de la integral de Fourier

Considere la función periódica triangular ilustrada en la Fig. 2.1(e). De las Ecs. (2.2.5) y (2.2.6) se encuentra que la representación en serie de Fourier es:

$$y(t) = \frac{1}{T_0} + \frac{8}{\pi^2 T_0} \left[\cos(2\pi f_0 t) + \frac{1}{3^2} \cos(6\pi f_0 t) + \frac{1}{5^2} \cos(10\pi f_0 t) + \dots \right] \quad (2.2.7)$$

donde $f_0 = 1/T_0$. Ahora se mostrará que la misma relación se puede obtener de la integral de Fourier.

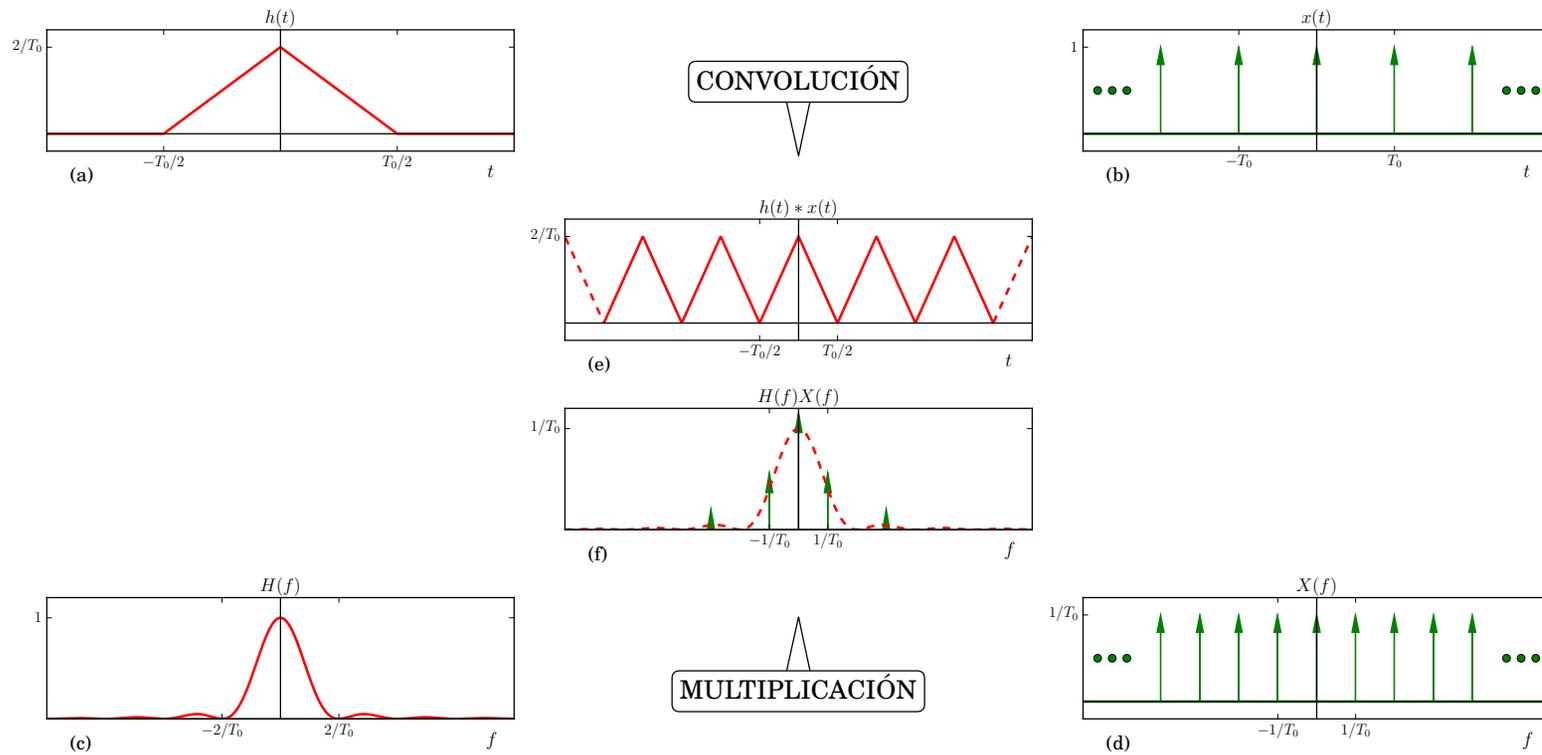


Figura 2.1: Desarrollo gráfico del teorema de convolución para la trasformada de Fourier de una función triangular periódica

Para llevar a cabo esta tarea, se utilizará el teorema de convolución, Ec. (2.1.29). Nótese que la función periódica triangular, con período T_0 , es simplemente la convolución de un simple pulso triangular mostrado en la Fig. 2.1(a) y la infinita secuencia de impulsos equidistantes mostrados en la Fig. 2.1(b) (revisar la Sección B.2). Entonces la función periódica $y(t)$ puede expresarse como

$$y(t) = h(t) * x(t) \quad (2.2.8)$$

La transformada de Fourier de $h(t)$ y $x(t)$ son ilustradas en la Fig. 2.1(c) y la Fig. 2.1(d) respectivamente (ver Apéndice C). Del teorema de convolución, la transformada de Fourier es el producto de estas dos funciones de frecuencia:

$$\begin{aligned} Y(f) &= H(f)X(f) \\ &= H(f)\frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_0}\right) \\ &= \frac{1}{T_0} \sum_{n=-\infty}^{\infty} H\left(\frac{n}{T_0}\right) \delta\left(f - \frac{n}{T_0}\right) \end{aligned} \quad (2.2.9)$$

donde se debe recordar la Ec. (A.11). Una prueba de que la transformada de Fourier de un tren de impulsos equidistantes es otra secuencia de impulsos equidistantes se puede ver en el libro de Papoulis [23].

La transformada de Fourier de una función periódica es entonces una secuencia de impulsos equidistantes con amplitud $H(n/T_0)$. Recordemos que la serie de Fourier de una función periódica es una suma infinita de sinusoides con amplitudes dadas por α_n , Ec. (2.2.6). Pero debido a que los límites de integración de Ec. (2.2.6) son $-T_0/2$ y $T_0/2$, y además que

$$h(t) = f(t), \quad -\frac{T_0}{2} < t < \frac{T_0}{2} \quad (2.2.10)$$

la función $f(t)$ puede ser reemplazada por $h(t)$ y la Ec. (2.2.6) puede ser escrita como

$$\alpha_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} h(t) e^{-j2\pi n f_0 t} dt = \frac{1}{T_0} H(n f_0) = \frac{1}{T_0} H\left(\frac{n}{T_0}\right) \quad (2.2.11)$$

Por lo tanto, los coeficientes obtenidos por medio de la integral de Fourier y por medio de la convencional serie de Fourier son los mismos para una función periódica. También,

comparando las Figs. 2.1(c) y 2.1(f) revelan que excepto por el factor $1/T_0$, los coeficientes α_n de la expansión en serie de Fourier de $y(t)$ son iguales a los valores de la transformada de Fourier $H(t)$ evaluada en n/T_0 .

En conclusión, se recalca que la clave para el desarrollo precedente es la incorporación de la teoría de distribución en la integral de Fourier. Esto será la base para un entendimiento minucioso de la transformada discreta de Fourier y posteriormente la transformada rápida de Fourier.

2.2.2. Muestreo de funciones

Ya se desarrollo la teoría que considera la transformada de Fourier de funciones de tiempo continuas e impulsos. Ahora se extenderá esta teoría para incluir funciones muestreadas. Si la función $h(t)$ es continua en $t = T$, entonces una muestra de $h(t)$ en el tiempo T es expresada como

$$\hat{h}(t) = h(t)\delta(t - T) = h(T)\delta(t - T) \quad (2.2.12)$$

donde el producto debe ser interpretado con la teoría de distribuciones, Ec. (A.11). Si $h(t)$ es continua en $t = nT$, para $n = 0, \pm 1, \pm 2, \dots$, entonces

$$\hat{h}(t) = \sum_{n=-\infty}^{\infty} h(nT)\delta(t - nT) \quad (2.2.13)$$

donde $\hat{h}(t)$ es la función muestreada de $h(t)$ con intervalo de muestreo T . El muestreo de $h(t)$ es entonces una secuencia infinita de impulsos equidistantes, cada uno con una amplitud dada por el valor de $h(t)$ correspondiente al tiempo de ocurrencia del impulso. La Fig. 2.2 ilustra gráficamente el concepto de muestreo. Debido a que la Ec. (2.2.13) es el producto de la función continua $h(t)$ y la secuencia de impulsos, se puede utilizar el teorema de frecuencia-convolución, Ec. (2.1.30), para encontrar la transformada de Fourier de la función muestreada. Como se ilustra en la Fig. 2.2, la función muestreada [Fig. 2.2(e)] es igual al producto de la función continua $h(t)$ mostrada en la Fig. 2.2(a) y la secuencia de impulsos $\Delta(t)$ como se muestra en la Fig. 2.2(b), donde esta última es llamada función de muestreo e implica una secuencia infinita de impulsos separados por T .

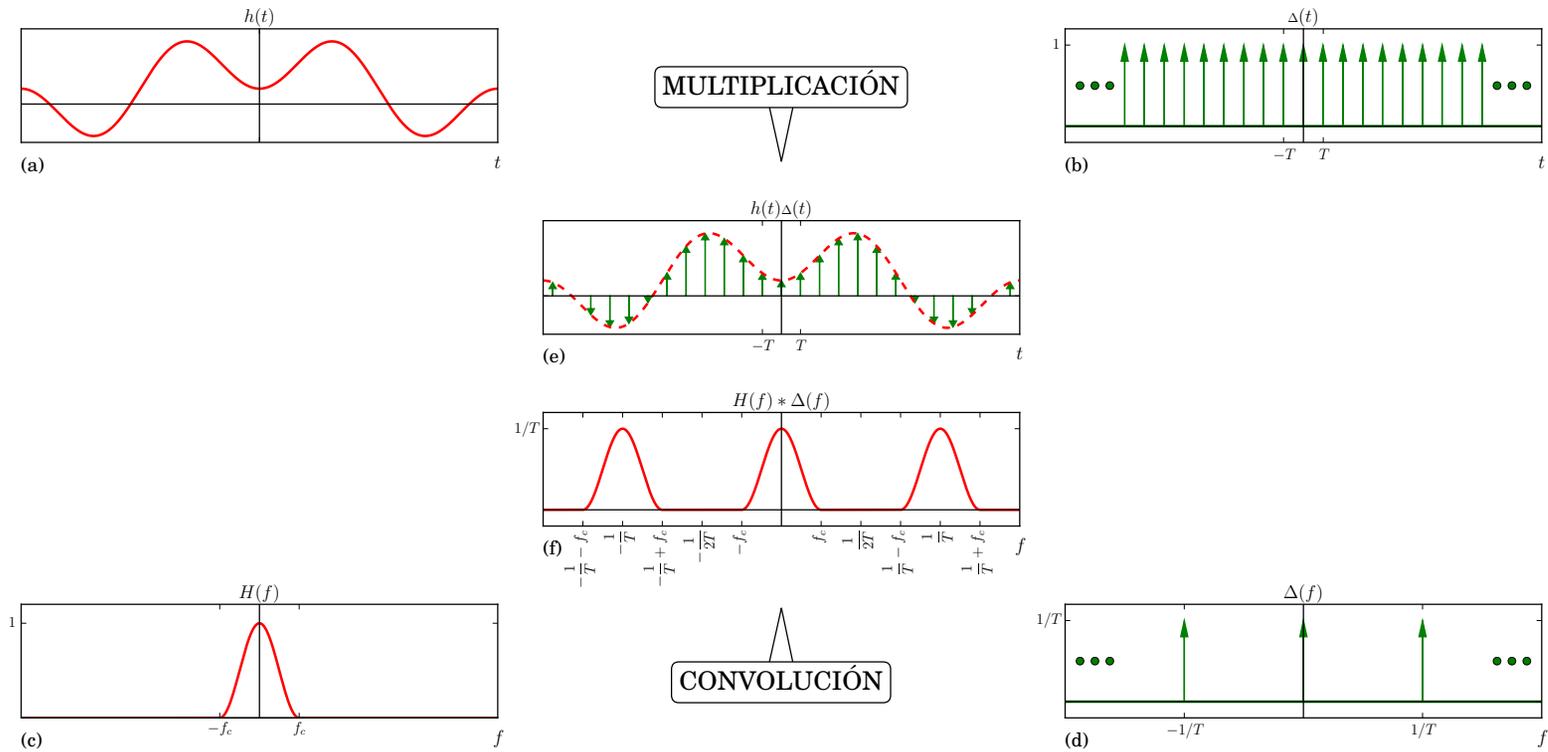


Figura 2.2: Desarrollo gráfico del teorema de frecuencia-convolución para la trasformada de Fourier de una función muestreada

Las transformadas de Fourier de $h(t)$ y $\Delta(t)$ se muestran en las Figs. 2.2(c) y 2.2(d) respectivamente. Del teorema de frecuencia-convolución, la transformada de Fourier deseada es la convolución de las funciones de frecuencia de las Figs. 2.2(c) y 2.2(d). La transformada de Fourier de una función muestreada es entonces una función periódica, donde un período contiene a la transformada de Fourier de función continua $h(t)$ [ver Figs. 2.2(c) y 2.2(f)]. Esta última declaración es válida si el intervalo de muestreo T es suficientemente pequeño.

Si T se escoge demasiado grande, el resultado se muestra en la Fig. 2.3. Nótese que a medida que el intervalo de muestreo se incrementa [Fig. 2.2(b) y Fig. 2.3(b)], los impulsos equidistantes de $\Delta(f)$ se acercan más [Fig. 2.2(d) y Fig. 2.3(d)]. Debido a la disminución del espacio de los impulsos en la frecuencia, su convolución con la función de frecuencia $H(f)$ [Fig. 2.3(c)] resulta en el solapamiento de la función mostrado en la Fig. 2.3(f). Esta distorsión de la deseada transformada de Fourier de una función muestreada es conocida como *aliasing* (*solapamiento*). Como se describe, el aliasing ocurre porque la función de tiempo no fue muestreada a una tasa suficientemente alta. Entonces ¿cómo se puede asegurar que la transformada de Fourier de una función muestreada no presente aliasing? Examinando las Figs. 2.3(c) y 2.3(d), muestran el hecho que el traslape en la convolución se presenta hasta que la separación de los impulsos de $\Delta(f)$ se incrementa hasta $1/T = 2f_c$, donde f_c es la componente de frecuencia más alta de la transformada de Fourier de la función continua $h(t)$. Por lo tanto, si el intervalo de muestreo T se escoge igual a la mitad del recíproco de la componente de frecuencia más alta, el aliasing no ocurre. Este es un concepto extremadamente importante en muchos campos de aplicación científica; la razón es porque solo necesitamos retener muestras de la función continua para determinar una réplica de la transformada de Fourier continua. Además, si una función es muestreada de tal forma que no ocurra aliasing, estas muestras pueden ser utilizadas para reconstruir de forma idéntica la función continua. Esto no es más que el teorema de muestreo que se verá a continuación.

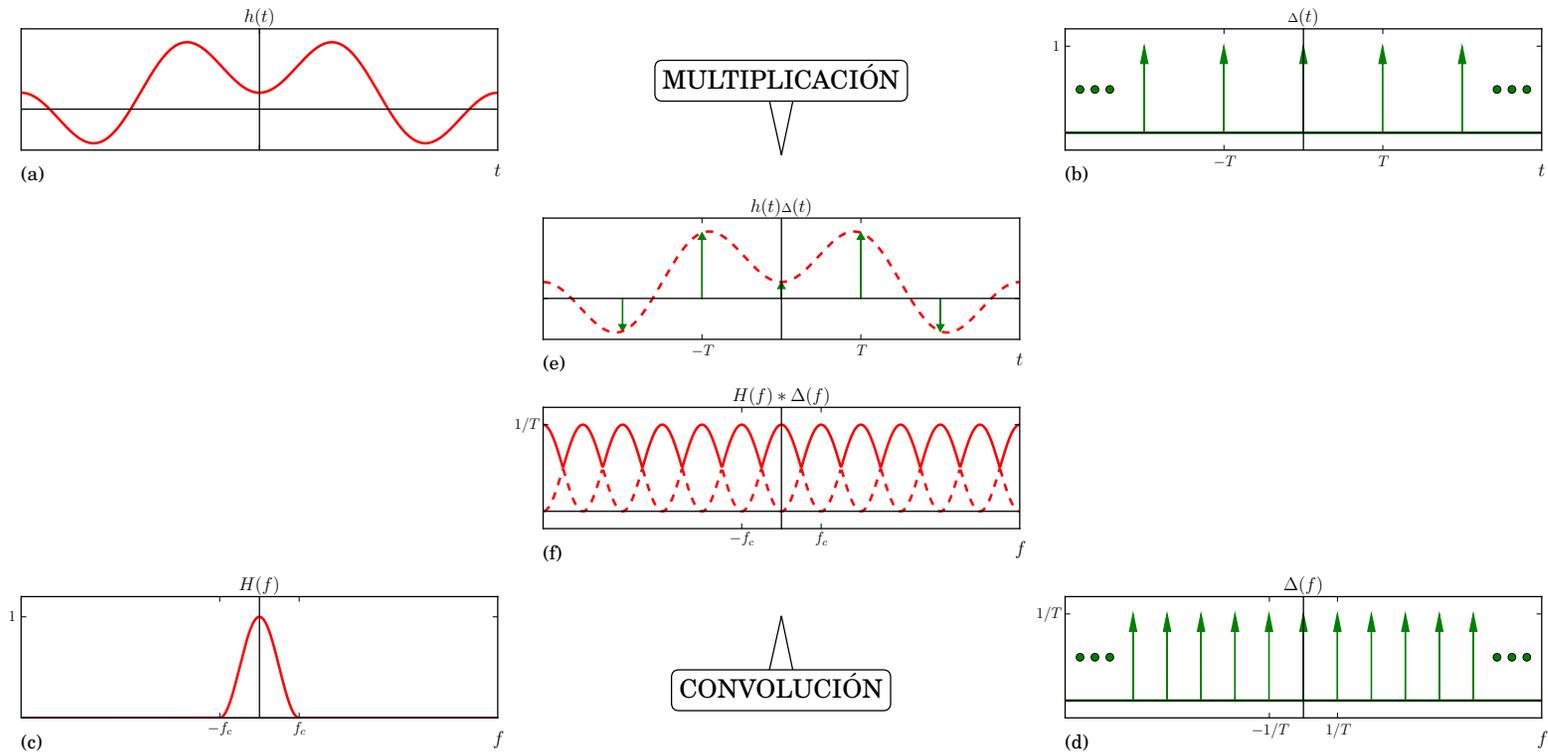


Figura 2.3: Efecto de aliasing en la transformada de Fourier muestreada a una tasa insuficiente

2.2.3. Teorema de muestreo

El teorema de muestreo establece que si la transformada de Fourier de una función $h(t)$ es cero para frecuencias mayores a cierta frecuencia f_c , entonces la función continua $h(t)$ se puede determinar de forma única a partir del conocimiento de las muestras

$$\hat{h}(t) = \sum_{n=-\infty}^{\infty} h(nT)\delta(t - nT) \quad (2.2.14)$$

donde $T = 1/(2f_c)$.

En particular, $h(t)$ esta representada por

$$h(t) = T \sum_{n=-\infty}^{\infty} h(nT) \frac{\text{sen}[2\pi f_c(t - nT)]}{\pi(t - nT)} \quad (2.2.15)$$

Restricciones del teorema son ilustradas gráficamente en la Fig. 2.4. Primero, es necesario que la transformada de Fourier de $h(t)$ sea cero para frecuencias mayores a f_c . Como se muestra en la Fig. 2.4(c), la función de frecuencia es de *banda limitada* a la frecuencia f_c ; el termino banda limitada es una forma corta de expresar que la transformada de Fourier es cero para $|f| > f_c$. El ancho de banda es el ancho de la banda de frecuencia positiva donde la amplitud no es cero, para la Fig. 2.4(c) el ancho de banda es f_c . La segunda restricción es que la separación de las muestras es elegida como $T = 1/(2f_c)$, y como consecuencia, el tren de impulsos de la Fig. 2.4(d) requiere estar separada por $1/T = 2f_c$. Esta separación asegura que cuando se aplique la convolución entre $\Delta(f)$ y $H(f)$, no exista aliasing. También se observa que las funciones $H(f)$ y $H(f) * \Delta(f)$, mostradas en las Figs. 2.4(c) y 2.4(f) respectivamente, son iguales en el intervalo $|f| < f_c$, con el factor de escala T . Si $T > 1/(2f_c)$, existirá aliasing; si $T < 1/(2f_c)$, el teorema aún aplica. El requerimiento que $T = 1/(2f_c)$ es simplemente la máxima separación entre muestras para la cual el teorema aplica. La frecuencia $1/T = 2f_c$ es conocida como la *tasa de muestreo de Nyquist*. Si estas dos condiciones son verdaderas, el teorema establece que la función $h(t)$ [Fig. 2.4(a)] puede ser reconstruida conociendo los impulsos mostrados en la Fig. 2.4(e).

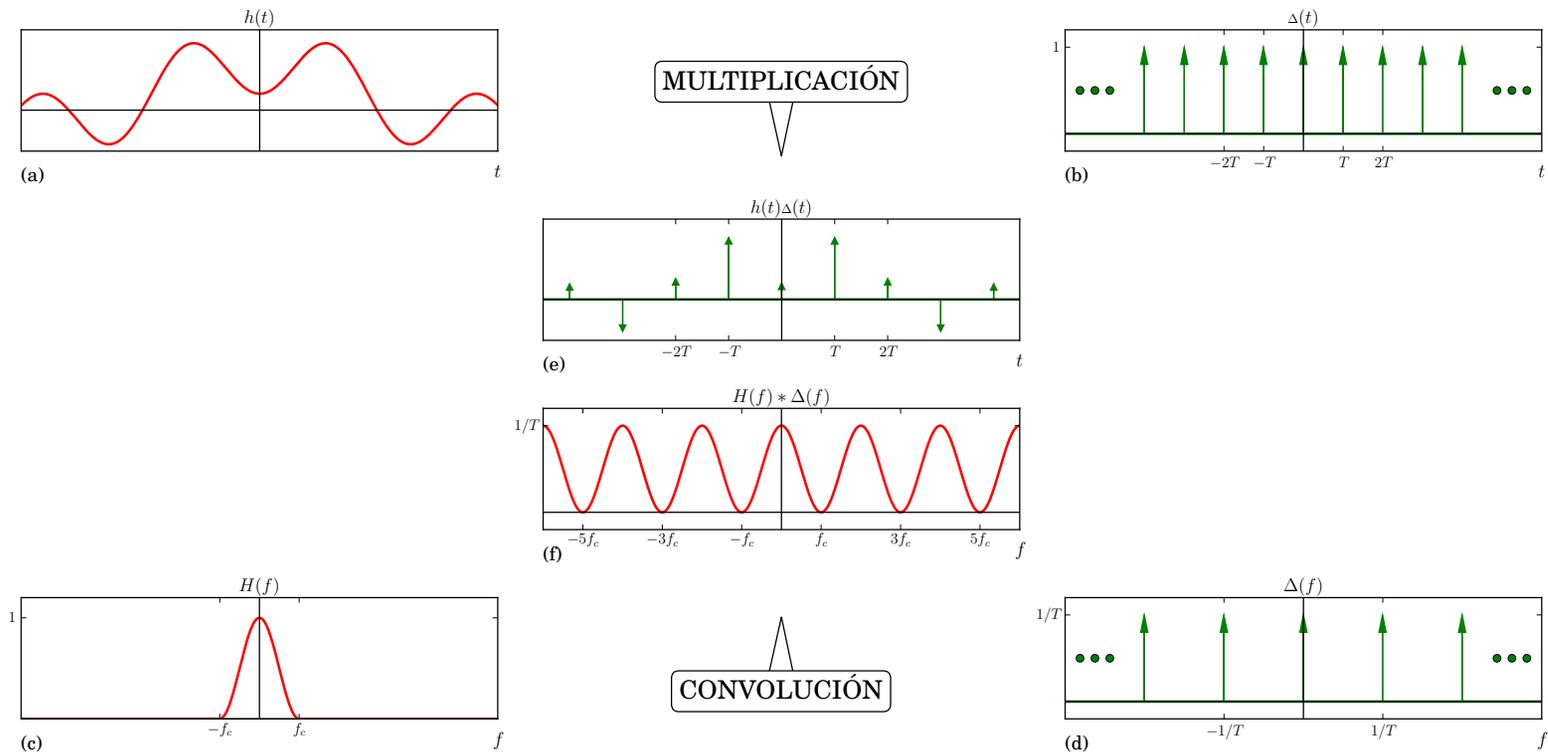


Figura 2.4: Transformada de Fourier de una función muestreada a la tasa de Nyquist

Para elaborar la prueba del teorema de muestreo, recordemos de la discusión de las restricciones del teorema que la transformada de Fourier de la función muestreada es idéntica, con el factor de escala T , a la transformada de Fourier de la función no muestreada, en el rango de frecuencias $-f_c \leq f \leq f_c$. De la Fig. 2.4(f), la transformada de Fourier de la función de tiempo muestreada está dada por $H(f) * \Delta(f)$. Por lo tanto, como se ilustra en las Figs. 2.5(a), 2.5(b) y 2.5(e), la multiplicación de una función de frecuencia rectangular de amplitud T con la transformada de Fourier de una función muestreada es la transformada de Fourier $H(f)$:

$$H(f) = [H(f) * \Delta(f)] Q(f) \quad (2.2.16)$$

La transformada inversa de Fourier de $H(f)$ es la función original $h(t)$, mostrada en la Fig. 2.5(f). Pero del teorema de convolución, $h(t)$ resulta de la convolución de las transformadas inversas de $H(f) * \Delta(f)$ y de la función de frecuencia rectangular, dicho de otra forma, $h(t)$ resulta de la convolución de $h(t)\Delta(t)$ y $q(t)$, mostrada en la Figs. 2.5(c) y 2.5(d) respectivamente:

$$\begin{aligned} h(t) &= [h(t)\Delta(t)] * q(t) \\ &= \sum_{n=-\infty}^{\infty} [h(nT)\delta(t - nT)] * q(t) \\ &= \sum_{n=-\infty}^{\infty} h(nT)q(t - nT) \\ &= T \sum_{n=-\infty}^{\infty} h(nT) \frac{\text{sen}[2\pi f_c(t - nT)]}{\pi(t - nT)} \end{aligned} \quad (2.2.17)$$

El par transformado de la función $q(t)$ se puede observar en el Apéndice C. La Ec. 2.2.17 es la expresión que se necesita para reconstruir $h(t)$ conociendo solo muestras de la misma.

Nótese cuidadosamente que es posible reconstruir una función muestreada perfectamente solo si la función es de banda limitada. En la práctica, esta condición raramente se presenta. La solución es tomar muestras a una tasa tal que el aliasing sea insignificante; puede ser necesario filtrar la señal antes de cuantizarla para asegurar que existe, en la medida de lo posible, una función de banda limitada.

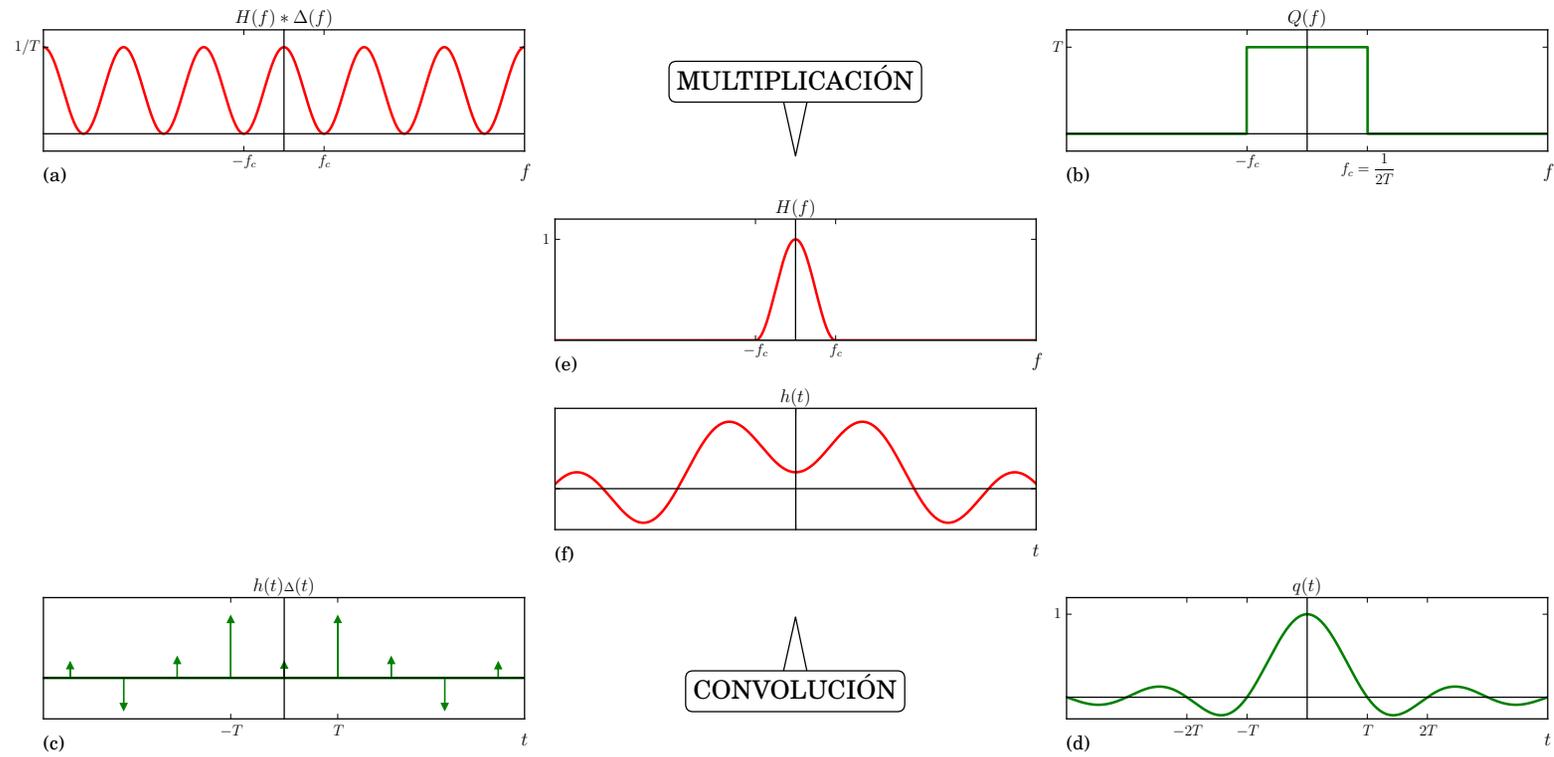


Figura 2.5: Deducción gráfica del teorema de muestreo

Las funciones de banda limitada consideradas en esta sección son referidas como *señales de banda base*. Esta nomenclatura se refiere a señales cuyo espectro de frecuencias *generalmente* ocupan el rango de frecuencias $0 \leq f < f_c$. Una *señal pasa banda* es una cuyo espectro de frecuencias ocupa el rango de frecuencias $f_{baja} < f < f_{alta}$ y $f_{baja} \gg 0$. El teorema del muestreo expuesto aquí puede ser aplicado a señales de banda base y pasa banda, pero existen teoremas de muestreo más eficientes para las señales pasa banda.

Teorema del muestreo en la frecuencia. Similar al muestreo en el dominio del tiempo, es un teorema de muestreo en el dominio de la frecuencia. Si la función $h(t)$ es de tiempo limitado, siendo,

$$h(t) = 0, \quad |t| > T_c \quad (2.2.18)$$

entonces su transformada de Fourier $H(f)$ puede determinarse unívocamente de muestras equidistantes de $H(f)$. En particular, $H(f)$ se representa por

$$H(f) = \frac{1}{2T_c} \sum_{n=-\infty}^{\infty} H\left(\frac{n}{2T_c}\right) \frac{\text{sen}\left[2\pi T_c\left(f - \frac{n}{2T_c}\right)\right]}{\pi\left(f - \frac{n}{2T_c}\right)} \quad (2.2.19)$$

La prueba es similiar a la del teorema del muestreo en el dominio del tiempo.

2.3. Transformada discreta de Fourier

El presente y futuro demandan que el procesamiento de señales continuas sea realizado por métodos discretos. Computadoras y sistemas de procesamiento digital pueden trabajar solamente con sumas finitas. Para convertir lo continuo a lo discreto y finito se requiere que una señal sea de tiempo limitado y banda limitada, algo que sabemos no puede ser cierto, y tomar un número finito de muestras, algo que sabemos no es suficiente. Pero funciona. Al menos esas aproximaciones funcionan en la medida en que una gran parte de la economía del mundo depende de ellas, y eso no es una mala medida del éxito [22].

Claramente, se puede desarrollar la teoría de la transformada discreta de Fourier independientemente de la transformada continua de Fourier, o simplemente transformada de Fourier; pero se seguirá el enfoque de Brigham [5], en el cual se deriva la transforma

discreta como un caso especial de la continua. Primero se desarrollará una derivación gráfica basada en la transformada continua de Fourier. Estos argumentos gráficos son luego justificados por un desarrollo teórico. Ambas aproximaciones enfatizan las modificaciones a la teoría de la transformada continua de Fourier que son necesarias para definir un par transformado orientado al análisis por computadoras.

2.3.1. Estudio gráfico

Considérese la función $h(t)$ y su transformada de Fourier $H(f)$, Fig. 2.6(a). Es deseable modificar el par transformado de tal manera que sea amigable para el análisis digital por computadora. Este par modificado, llamado *transformada discreta de Fourier*, se debe aproximar lo más cerca posible a la transformada continua de Fourier.

Para determinar la transformada de Fourier de $h(t)$ por medio de técnicas de análisis digital, es necesario muestrear $h(t)$, como se describió en la sección anterior. El muestreo es realizado por la multiplicación de $h(t)$ por la función de muestreo mostrada en la Fig. 2.6(b), donde el intervalo de muestreo es T . La función muestreada $\hat{h}(t)$ y su transformada de Fourier son ilustradas en la Fig. 2.6(c). Este par transformado de Fourier representa la primera modificación al par original, necesario en la definición del par transformado discreto. Nótese que en este punto el par transformado modificado difiere del par transformado original solo por el efecto de aliasing que resulta del muestreo. Como se expuso en la Subsección 2.2.3, si la frecuencia a la que se toman muestras de la función $h(t)$ es al menos el doble de la componente de frecuencia más alta de $h(t)$, no existirá pérdida de información como consecuencia del muestreo. Si la función no es de banda limitada, p. ej., $H(f) \neq 0$ para alguna $|f| > f_c$, entonces el muestreo producirá aliasing como se muestra en la Fig. 2.6(c). Para reducir este error solo se tiene una opción, tomar muestras rápidamente, o sea, tomar T muy pequeña.

El par transformado de Fourier de la Fig. 2.6(c) no es adecuado para el análisis por computadora porque un infinito número de muestras son consideradas; es necesario truncar la función muestreada $\hat{h}(t)$ y así solo un finito número de puntos, N , son considerados.

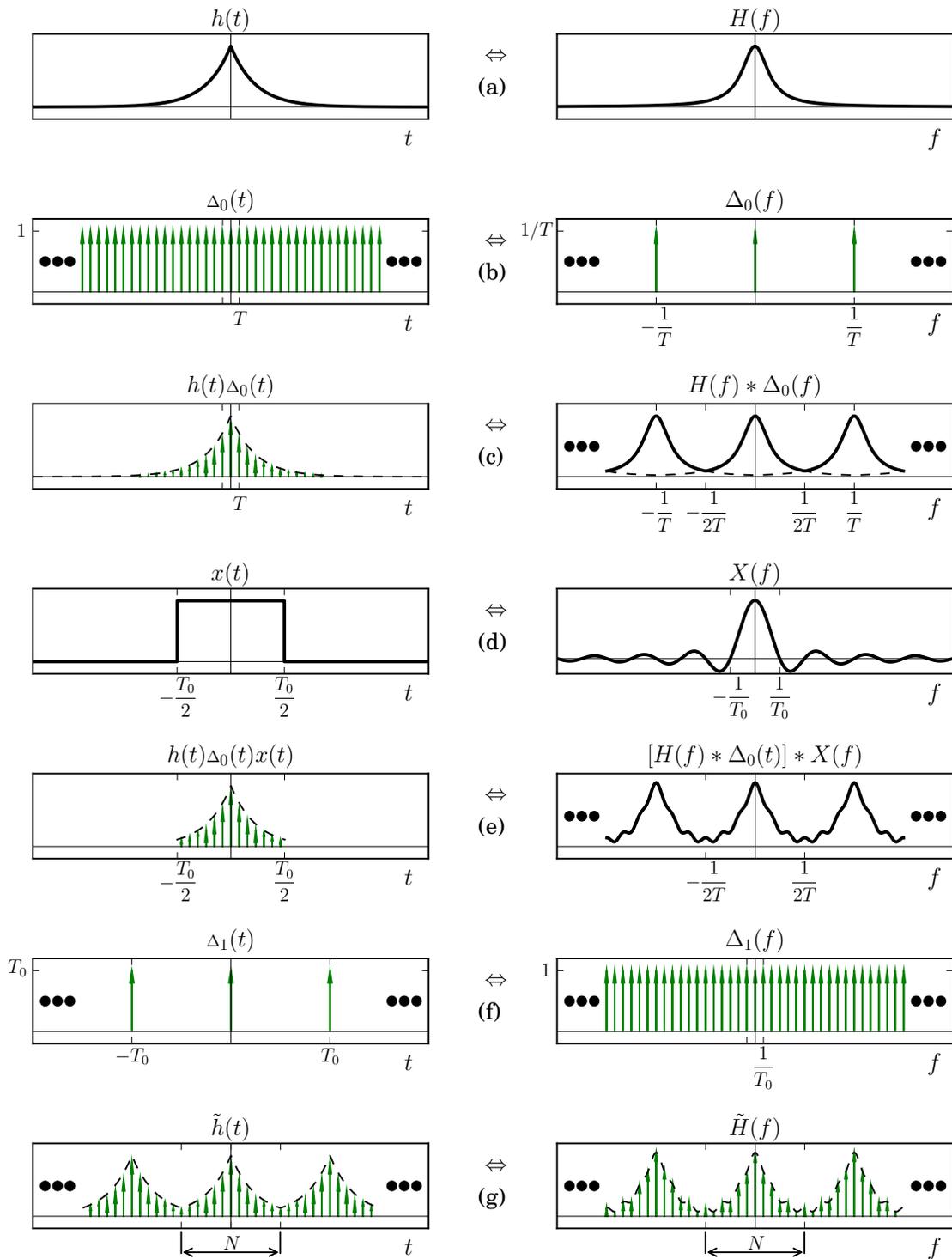


Figura 2.6: Desarrollo gráfico de la transformada discreta de Fourier

La función rectangular, o de truncamiento, y su transformada de Fourier son mostradas en la Fig. 2.6(d). El producto entre la infinita secuencia de impulsos representando $h(t)$ y la función de truncamiento arrojando una función de longitud finita de tiempo se muestra en la Fig. 2.6(e). La acción de truncar introduce la segunda modificación al par transformado de Fourier original; este efecto es convolucionar en la frecuencia la transformada que sufrió aliasing de la Fig. 2.6(c) con la transformada de Fourier de la función de truncamiento mostrada en la Fig. 2.6(d). Como se muestra en la Fig. 2.6(e), la transformada en la frecuencia ahora tiene ondulaciones, este efecto se ha acentuado en la ilustración para enfatizarlo. Para reducir este efecto, recordemos la relación inversa que existe entre el *ancho* de una función del tiempo y su transformada de Fourier [Ecs. (2.1.11) y (2.1.12)]. A medida que la función de truncamiento (rectangular) incrementa su longitud, entonces la función $\text{sen}(f)/f$ se aproxima a un impulso; entre más cercana sea la aproximación de la función $\text{sen}(f)/f$ a un impulso, menos ondulaciones o error es introducido por la convolución como resultado del truncamiento. Por lo tanto, es deseable escoger la longitud de la función de truncamiento lo más ancho posible.

El par transformado de la Fig. 2.6(e) aún no es aceptable para un par transformado discreto de Fourier porque la transformada en la frecuencia es una función continua. Para el análisis por computadora, solo muestras de la función de frecuencia son aceptables; es necesario modificar la transformada en la frecuencia con una función de muestreo de frecuencia, la cual es mostrada en la Fig. 2.6(f) y cuya frecuencia es $1/T_0$.

El par transformado discreto de Fourier de la Fig. 2.6(g) es aceptable para propósitos de análisis digital por computadora debido a que tanto el dominio del tiempo como el de la frecuencia son representados por valores discretos. Como se ilustra en la Fig. 2.6(g), la función de tiempo original $h(t)$ es aproximada por N muestras; la transformada de Fourier original $H(f)$ también es aproximada por N muestras. Estas N muestras definen el par transformado discreto de Fourier y lo aproximan al par transformado de Fourier original. Tenga en cuenta que el muestreo en el dominio del tiempo resulta en una función de frecuencia periódica; el muestreo en el dominio de la frecuencia resulta en una función de tiempo periódica. Por lo tanto, la transformada discreta de Fourier requiere que las

funciones originales de los dominios de tiempo y frecuencia sean modificadas tal que se conviertan en funciones periódicas. N muestras de tiempo y N valores de frecuencia representan un período en los dominios de tiempo y frecuencia respectivamente. Debido a que los N valores de tiempo y frecuencia están relacionados por la transformada continua de Fourier, entonces una relación discreta puede ser derivada.

2.3.2. Estudio teórico

El desarrollo gráfico anterior ilustra el punto que si un par transformado continuo de Fourier es modificado adecuadamente, entonces el par modificado es aceptable para el análisis digital por computadora. Por lo tanto, para desarrollar el par transformado discreto de Fourier, solo es necesario derivar las relaciones matemáticas que resultan de las modificaciones requeridas: muestreo en el dominio del tiempo, truncamiento, muestreo en el dominio de la frecuencia.

Considere el par transformado de Fourier mostrado en la Fig. 2.7(a). Para discretizar este par transformado, primero es necesario muestrear la función $h(t)$; la función muestreada puede ser escrita como $h(t)\Delta_0(t)$, donde $\Delta_0(t)$ es la función de muestreo en el dominio del tiempo con intervalo de muestreo T , ilustrada en la Fig. 2.7(b). De la Ec. (2.2.13), la función de muestreo puede ser escrita como

$$\begin{aligned} h(t)\Delta_0(t) &= h(t) \sum_{k=-\infty}^{\infty} \delta(t - kT) \\ &= \sum_{k=-\infty}^{\infty} h(kT)\delta(t - kT) \end{aligned} \quad (2.3.1)$$

El resultado de esta multiplicación es mostrado en la Fig. 2.7(c). Nótese el efecto de aliasing que resulta de la elección de T . Luego, la función muestreada es truncada por la multiplicación con la función rectangular $x(t)$, como se muestra en la Fig. 2.7(d):

$$x(t) = \begin{cases} 1, & \text{para } -\frac{T}{2} < t < T_0 - \frac{T}{2} \\ 0, & \text{de lo contrario} \end{cases} \quad (2.3.2)$$

donde T_0 es el ancho de la función de truncamiento.

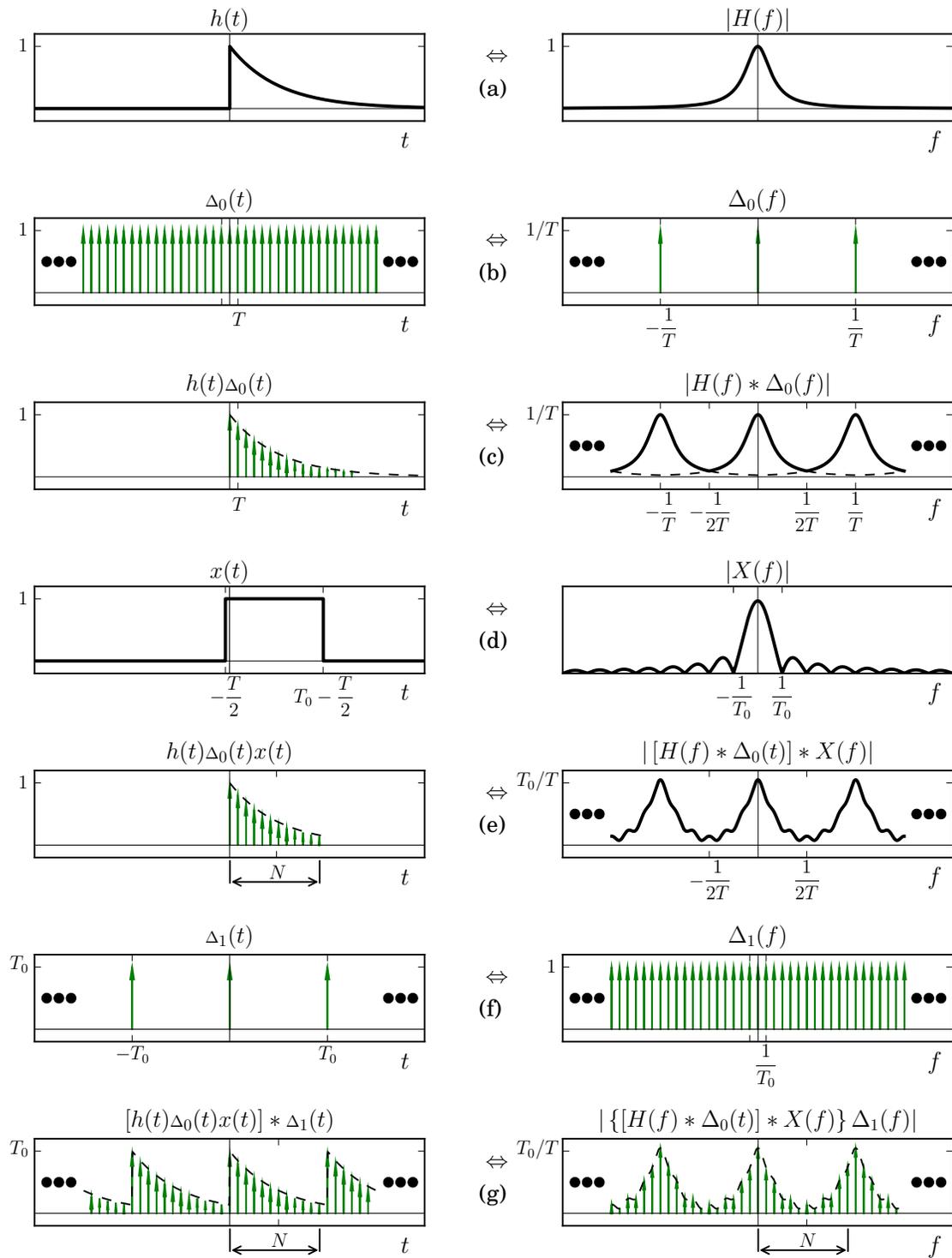


Figura 2.7: Derivación gráfica del par transformado discreto de Fourier

Una pregunta obvia en este punto es: ¿por qué la función rectangular $x(t)$ no está centrada en cero o en $T_0/2$? No se centra en cero para evitar problemas de notación y la razón por la que no se centra en $T_0/2$ se explicará en breve.

El truncamiento arroja

$$\begin{aligned} h(t)_{\Delta_0}(t)x(t) &= \left[\sum_{k=-\infty}^{\infty} h(kT)\delta(t - kT) \right] x(t) \\ &= \sum_{k=0}^{N-1} h(kT)\delta(t - kT) \end{aligned} \quad (2.3.3)$$

donde se ha asumido que la función tiene N impulsos equidistantes dentro del intervalo de truncamiento, esto es, $N = T_0/T$. La función muestreada y truncada junto con su transformada de Fourier se muestran en la Fig. 2.7(e). Como en el ejemplo anterior, el truncamiento en el dominio del tiempo implica *ondulaciones* en el dominio de la frecuencia.

El paso final para modificar el par transformado de Fourier original al par transformado discreto de Fourier es muestrear la transformada de Fourier de la Ec. (2.3.3), que en el dominio del tiempo equivale a realizar una convolución entre la función muestreada y truncada de la Ec. (2.3.3) con la función $\Delta_1(t)$ mostrada en la Fig. 2.7(f). En el Apéndice C se encuentra que

$$\Delta_1(t) = T_0 \sum_{r=-\infty}^{\infty} \delta(t - rT_0) \quad (2.3.4)$$

La relación deseada es $[h(t)_{\Delta_0}(t)x(t)] * \Delta_1(t)$; por lo tanto,

$$\begin{aligned} [h(t)_{\Delta_0}(t)x(t)] * \Delta_1(t) &= \left[\sum_{k=0}^{N-1} h(kT)\delta(t - kT) \right] * \left[T_0 \sum_{r=-\infty}^{\infty} \delta(t - rT_0) \right] \\ &= \dots + T_0 \sum_{k=0}^{N-1} h(kT)\delta(t + T_0 - kT) \\ &\quad + T_0 \sum_{k=0}^{N-1} h(kT)\delta(t - kT) \\ &\quad + T_0 \sum_{k=0}^{N-1} h(kT)\delta(t - T_0 - kT) + \dots \end{aligned} \quad (2.3.5)$$

Nótese que la Ec. (2.3.5) es periódica con período T_0 ; otra forma de escribirla es

$$\tilde{h}(t) = T_0 \sum_{r=-\infty}^{\infty} \left[\sum_{k=0}^{N-1} h(kT) \delta(t - kT - rT_0) \right] \quad (2.3.6)$$

Se escoge la notación $\tilde{h}(t)$ que indica que es una aproximación de la función $h(t)$.

Ahora se explicará el porqué se escogió $x(t)$ como se expresa en la Ec. (2.3.2). Nótese que el resultado de la convolución en la Ec. (2.3.6) es una función periódica con período T_0 consistente en N muestras. Si la función rectangular hubiera sido escogida de tal forma que dos muestras coincidieran con cada extremo de la función, la convolución de dicha función rectangular con los impulsos espaciados a intervalos de T_0 resultaría en aliasing en el dominio del tiempo. Dicho de otra forma, la n -ésima muestra de un período coincidiría y se sumaría a la primer muestra del siguiente período. Para asegurar entonces que el aliasing no ocurra en el dominio del tiempo, es necesario elegir el intervalo de truncamiento como se muestra en la Fig. 2.7(d). La función de truncamiento también puede elegirse como se ilustra en la Fig. 2.6(d), pero nótese que los extremos de la función de truncamiento se ubican en el punto medio de dos muestras adyacentes para evitar el aliasing en el dominio del tiempo.

Para encontrar la transformada de Fourier de la Ec. (2.3.6), recordemos de la discusión de la serie de Fourier, Subsección 2.2.1, que la transformada de Fourier de una función periódica $h(t)$ es una secuencia de impulsos equidistantes:

$$\tilde{H}(nf_0) = \sum_{n=-\infty}^{\infty} \alpha_n \delta(f - nf_0), \quad f_0 = \frac{1}{T_0} \quad (2.3.7)$$

donde

$$\alpha_n = \frac{1}{T_0} \int_{-T/2}^{T_0-T/2} \tilde{h}(t) e^{-j2\pi nt/T_0} dt, \quad n = 0, \pm 1, \pm 2, \dots \quad (2.3.8)$$

Sustituyendo la Ec. (2.3.6) en la Ec. (2.3.8) resulta

$$\alpha_n = \frac{1}{T_0} \int_{-T/2}^{T_0-T/2} T_0 \sum_{r=-\infty}^{\infty} \left[\sum_{k=0}^{N-1} h(kT) \delta(t - kT - rT_0) \right] e^{-j2\pi nt/T_0} dt$$

Esta integral se realiza solo sobre un período; por lo tanto,

$$\begin{aligned}
\alpha_n &= \int_{-T/2}^{T_0-T/2} \sum_{k=0}^{N-1} h(kT) \delta(t - kT) e^{-j2\pi nt/T_0} dt \\
&= \sum_{k=0}^{N-1} h(kT) \int_{-T/2}^{T_0-T/2} e^{-j2\pi nt/T_0} \delta(t - kT) dt \\
&= \sum_{k=0}^{N-1} h(kT) e^{-j2\pi knT/T_0}
\end{aligned} \tag{2.3.9}$$

Ya que $T_0 = NT$, la Ec. (2.3.9) se puede escribir como

$$\alpha_n = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kn/N}, \quad n = 0, \pm 1, \pm 2, \dots \tag{2.3.10}$$

y la transformada de Fourier de la Ec. (2.3.6) es

$$\tilde{H}\left(\frac{n}{NT}\right) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kn/N} \tag{2.3.11}$$

De la Ec. (2.3.11), a simple vista no se observa que es periódica. Sin embargo, solamente hay N valores distintos computables de la Ec. (2.3.11). Para demostrar la periodicidad, tomemos $n = r$, donde r es un entero arbitrario; evaluando en la Ec. (2.3.11)

$$\tilde{H}\left(\frac{r}{NT}\right) = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kr/N} \tag{2.3.12}$$

Ahora tomemos $n = r + N$ y obsérvese que

$$\begin{aligned}
e^{-j2\pi k(r+N)/N} &= e^{-j2\pi kr/N} e^{-j2\pi k} \\
&= e^{-j2\pi kr/N}
\end{aligned} \tag{2.3.13}$$

debido a que $e^{-j2\pi k} = \cos(j2\pi k) - j \operatorname{sen}(j2\pi k) = 1$, para k entero. Entonces, para $n = r + N$,

$$\begin{aligned}
\tilde{H}\left(\frac{r+N}{NT}\right) &= \sum_{k=0}^{N-1} h(kT) e^{-j2\pi k(r+N)/N} \\
&= \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kr/N} \\
&= \tilde{H}\left(\frac{r}{NT}\right)
\end{aligned} \tag{2.3.14}$$

Por lo tanto, solo existen N valores distintos para los cuales la Ec. (2.3.11) puede ser evaluada; $\tilde{H}(n/NT)$ es periódica con período de N muestras. La transformada de Fourier de la Ec. (2.3.11) puede ser expresada equivalentemente como

$$\tilde{H}\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} h(kT)e^{-j2\pi nk/N}, \quad n = 0, 1, \dots, N-1 \quad (2.3.15)$$

La Ec. (2.3.15) es la transformada discreta de Fourier buscada; esta expresión relaciona N muestras de tiempo y N muestras de frecuencia por medio de la transformada continua de Fourier. Entonces la transformada discreta de Fourier es un caso especial de la transformada continua de Fourier. La notación $\tilde{H}(n/NT)$ indica que la transformada discreta de Fourier es una aproximación de la transformada continua de Fourier. Generalmente, la Ec. (2.3.15) se expresa como

$$G\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} g(kT)e^{-j2\pi nk/N}, \quad n = 0, 1, \dots, N-1 \quad (2.3.16)$$

porque la transformada de Fourier de la función periódica muestreada $g(kT)$ es idéntica a $G(n/NT)$.

2.3.3. Transformada inversa discreta de Fourier

La transformada inversa discreta de Fourier esta dada por

$$g(kT) = \frac{1}{N} \sum_{n=0}^{N-1} G\left(\frac{n}{NT}\right) e^{j2\pi nk/N}, \quad k = 0, 1, \dots, N-1 \quad (2.3.17)$$

Para probar que la Ec. (2.3.17) y la relación de transformación, Ec. (2.3.16), forman un par transformado discreto de Fourier, se debe sustituir la Ec. (2.3.17) en Ec. (2.3.16)

$$\begin{aligned} G\left(\frac{n}{NT}\right) &= \sum_{k=0}^{N-1} \left[\frac{1}{N} \sum_{r=0}^{N-1} G\left(\frac{r}{NT}\right) e^{j2\pi rk/N} \right] e^{-j2\pi nk/N} \\ &= \frac{1}{N} \sum_{r=0}^{N-1} G\left(\frac{r}{NT}\right) \left[\sum_{k=0}^{N-1} e^{j2\pi rk/N} e^{-j2\pi nk/N} \right] \\ &= G\left(\frac{n}{NT}\right) \end{aligned} \quad (2.3.18)$$

La identidad de la Ec. (2.3.18) se deriva de la relación de ortogonalidad:

$$\sum_{k=0}^{N-1} e^{j2\pi rk/N} e^{-j2\pi nk/N} = \begin{cases} N, & \text{para } r = n \\ 0, & \text{de lo contrario} \end{cases} \quad (2.3.19)$$

donde una prueba de esta relación se puede encontrar en el texto de Briggs [4].

La fórmula de inversión discreta, Ec. (2.3.17), exhibe periodicidad al igual que la transformada discreta; el período es definido por N muestras de $g(kT)$. Esta propiedad resulta de la naturaleza periódica de $e^{j2\pi nk/N}$. Por lo tanto, $g(kT)$ es definida por el conjunto completo de enteros $k = 0, \pm 1, \pm 2, \dots$, y por la restricción de la identidad

$$g(kT) = g[(rN + k)T], \quad r = 0, \pm 1, \pm 2, \dots \quad (2.3.20)$$

En resumen, el par transformado discreto de Fourier está dado por

$$g(kT) = \frac{1}{N} \sum_{n=0}^{N-1} G\left(\frac{n}{NT}\right) e^{j2\pi nk/N} \Leftrightarrow G\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} g(kT) e^{-j2\pi nk/N} \quad (2.3.21)$$

Es importante recordar que el par transformado de la Ec. (2.3.21) requiere que las funciones, tanto del dominio del tiempo como de la frecuencia, sean periódicas:

$$G\left(\frac{n}{NT}\right) = G\left(\frac{rN + n}{NT}\right), \quad r = 0, \pm 1, \pm 2, \dots \quad (2.3.22)$$

$$g(kT) = g[(rN + k)T], \quad r = 0, \pm 1, \pm 2, \dots \quad (2.3.23)$$

2.3.4. Propiedades de la transformada discreta de Fourier

Las propiedades establecidas para la transformada de Fourier en la Subsección 2.1.2 se pueden extender a la transformada discreta de Fourier porque se mostró que ésta es un caso especial de la transformada continua. Aunque por lo general se utilizan los equivalentes continuos en la resolución de problemas, son las propiedades de la transformada discreta que forman la base teórica para la aplicación de la FFT. Se reemplaza kT por k y n/NT por n por conveniencia en la notación.

Linealidad. Si $x(k)$ y $y(k)$ tienen transformadas discreta de Fourier $X(n)$ y $Y(n)$, respectivamente, y sean c_1 y c_2 constantes arbitrarias, entonces

$$c_1x(k) + c_2y(k) \Leftrightarrow c_1X(n) + c_2Y(n) \quad (2.3.24)$$

Simetría. Si $h(k)$ y $H(n)$ son un par transformado discreto de Fourier, entonces

$$\frac{1}{N}H(k) \Leftrightarrow h(-n) \quad (2.3.25)$$

Desplazamiento en tiempo y frecuencia. Si $h(k)$ es desplazada en el tiempo por una constante i , entonces

$$h(k - i) \Leftrightarrow H(n)e^{-j2\pi ni/N} \quad (2.3.26)$$

Por otro lado, si $H(n)$ es desplazada en la frecuencia por una constante i , entonces su transformada inversa discreta de Fourier es

$$h(k)e^{j2\pi ik/N} \Leftrightarrow H(n - i) \quad (2.3.27)$$

Fórmula de inversión alternativa. La fórmula de inversión discreta, Ec. (2.3.17), también puede ser escrita de la siguiente forma:

$$h(k) = \frac{1}{N} \sum_{n=0}^{N-1} H(n)e^{j2\pi nk/N} \quad (2.3.28)$$

$$= \frac{1}{N} \left[\sum_{n=0}^{N-1} H^*(n)e^{-j2\pi nk/N} \right]^* \quad (2.3.29)$$

La importancia de la fórmula de inversión alternativa es que la transformada discreta, Ec. (2.3.16), puede ser utilizada para calcular tanto la transformada discreta de Fourier como su inversa. Por lo tanto, solo se necesita desarrollar un solo programa para la FFT.

Funciones pares. Si $h_e(k)$ es una función par, entonces $h_e(k) = h_e(-k)$ y la transformada discreta de Fourier de $h_e(k)$ es una función par y real:

$$h_e(k) \Leftrightarrow R_e(n) = \sum_{k=0}^{N-1} h_e(k) \cos\left(\frac{2\pi nk}{N}\right) \quad (2.3.30)$$

Funciones impares. Si $h_o(k)$ es una función impar, entonces $h_o(k) = -h_o(-k)$ y la transformada discreta de Fourier de $h_o(k)$ es una función impar e imaginaria:

$$h_o(k) \Leftrightarrow jI_o(n) = -j \sum_{k=0}^{N-1} h_o(k) \operatorname{sen} \left(\frac{2\pi nk}{N} \right) \quad (2.3.31)$$

Descomposición de funciones. Para descomponer una función arbitraria $h(k)$ en una función par y otra impar, simplemente se suma y resta la función común $h(-k)/2$.

$$\begin{aligned} h(k) &= \frac{h(k)}{2} + \frac{h(k)}{2} \\ &= \left[\frac{h(k)}{2} + \frac{h(-k)}{2} \right] + \left[\frac{h(k)}{2} - \frac{h(-k)}{2} \right] \\ &= h_e(k) + h_o(k) \end{aligned} \quad (2.3.32)$$

Los términos entre paréntesis satisfacen la definición de una función par e impar, respectivamente. Debido a que $h(k)$ es periódica con período N , entonces

$$h(-k) = h(N - k) \quad (2.3.33)$$

y

$$h_e(k) = \frac{h(k)}{2} + \frac{h(N - k)}{2}, \quad h_o(k) = \frac{h(k)}{2} - \frac{h(N - k)}{2} \quad (2.3.34)$$

Para funciones periódicas discretas, la Ec. (2.3.34) es la relación deseada para la descomposición. De las Ecs. (2.3.30) y (2.3.31), la transformada discreta de Fourier de la Ec. (2.3.32) es

$$H(n) = R(n) + jI(n) = H_e(n) + H_o(n) \quad (2.3.35)$$

donde

$$H_e(n) = R(n) \quad \text{y} \quad H_o(n) = jI(n) \quad (2.3.36)$$

Funciones de tiempo como variable compleja. Si $h(k) = h_r(k) + jh_i(k)$, donde $h_r(k)$ y $h_i(k)$ son, respectivamente, la parte real e imaginaria, entonces la transformada

discreta de Fourier será

$$\begin{aligned}
 H(n) &= \sum_{k=0}^{N-1} [h_r(k) + jh_i(k)] e^{-j2\pi nk/N} \\
 &= R(n) + jI(n) \\
 &= \sum_{k=0}^{N-1} \left[h_r(k) \cos\left(\frac{2\pi nk}{N}\right) + h_i(k) \operatorname{sen}\left(\frac{2\pi nk}{N}\right) \right] \\
 &\quad - j \sum_{k=0}^{N-1} \left[h_r(k) \operatorname{sen}\left(\frac{2\pi nk}{N}\right) - h_i(k) \cos\left(\frac{2\pi nk}{N}\right) \right]
 \end{aligned} \tag{2.3.37}$$

Teorema de convolución en el tiempo. La convolución discreta en el tiempo esta definida por la sumatoria:

$$y(k) = \sum_{i=0}^{N-1} x(i)h(k-i) \tag{2.3.38}$$

donde $x(k)$, $h(k)$, y $y(k)$ son funciones periódicas con período N .

Análogo a la teoría de la transformada continua de Fourier, una de las más importantes propiedades de la transformada discreta de Fourier se expone por medio de la transformada discreta de Fourier de la Ec. (2.3.38). La transformación discreta de Fourier produce el teorema de convolución discreto que se expresa como

$$\sum_{i=0}^{N-1} x(i)h(k-i) \Leftrightarrow X(n)H(n) \tag{2.3.39}$$

Teorema de convolución en la frecuencia. Considere la convolución en la frecuencia:

$$Y(n) = \sum_{i=0}^{N-1} X(i)H(n-i) \tag{2.3.40}$$

Aplicando la transformada inversa discreta de Fourier se obtiene el par transformado

$$x(k)h(k) \Leftrightarrow \frac{1}{N} \sum_{i=0}^{N-1} X(i)H(n-i) \tag{2.3.41}$$

En el libro de Brigham [5] puede encontrar en detalle la prueba de estas propiedades y en el Apéndice D se puede encontrar un resumen con estas propiedades.

2.4. Origen de la transformada rápida de Fourier (FFT)

La transformada rápida de Fourier (FFT por sus siglas en inglés—*fast Fourier transform*) tiene una amplia e interesante historia, pero debido a que el tiempo no lo permite no se darán detalles en este documento, pero si esta interesado puede consultar [7], [11], [12] y [19]. En términos simples, la FFT es un método eficiente para calcular la transformada discreta de Fourier. En palabras de G. D. Bergland:

“El algoritmo de la transformada rápida de Fourier puede reducir el tiempo necesario para encontrar la transformada discreta de Fourier de varios minutos a menos de un segundo, y puede reducir los costos de varios dolares a varios centavos.” [2]

Después de la publicación del artículo de Cooley and Tukey en 1965 [9], su amplia aceptación y difusión se considera hasta la fecha (véase [2], [8], [13], [14], [15], [16], [24], [25]). En dicho artículo se detalla la esencia del algoritmo y su desarrollo.

El algoritmo de Cooley-Tukey hace a la FFT extremadamente útil al reducir el número de cálculos a partir de algo del orden de n^2 a $n \log n$, lo que obviamente ofrece una enorme reducción en el tiempo de cálculo. Es tan útil, de hecho, que la FFT hizo que la revista *Computing in Science & Engineering* la ubicara en la lista de los 10 principales algoritmos en un artículo que indica que el algoritmo es, “quizás, el algoritmo más ubicuo en uso hoy en día.” El método usado en el algoritmo de Cooley-Tukey se puede aplicar a otras transformaciones ortogonales tales como la Hadamard, Hartley, y Haar [15].

2.5. Enfoque de Brigham para la FFT

La interpretación de los resultados de la transformada rápida de Fourier no requiere una enseñanza bien fundada en el propio algoritmo, sino más bien un conocimiento profundo de la transformada discreta de Fourier. Esto es consecuencia del hecho de que la FFT es simplemente un algoritmo (es decir, un método particular para realizar una serie de cálculos) que puede calcular la transformada discreta de Fourier mucho más rápidamente

que otros algoritmos disponibles. Por esta razón, la discusión de la FFT se enfoca sólo al aspecto computacional del algoritmo.

Un ejemplo de factorización matricial sencillo se utiliza para justificar intuitivamente el algoritmo FFT. Las matrices factorizadas se representan alternativamente mediante gráficos de flujo de señales. A partir de estos gráficos, se construye la lógica de un programa de FFT para computadora. A continuación, se presentan los desarrollos teóricos de diversas formas del algoritmo FFT.

2.5.1. Formulación matricial

Considere la transformada discreta de Fourier, Ec. (2.3.16)

$$X(n) = \sum_{k=0}^{N-1} x_0(k) e^{-j2\pi nk/N}, \quad n = 0, 1, \dots, N-1 \quad (2.5.1)$$

donde se ha reemplazado kT por k y n/NT por n por conveniencia en la notación. Nótese que la Ec. (2.5.1) describe el cálculo de N ecuaciones. Por ejemplo, si $N = 4$ y tomamos

$$W = e^{-j2\pi/N} \quad (2.5.2)$$

entonces la Ec. (2.5.1) puede ser escrita como

$$\begin{aligned} X(0) &= x_0(0)W^0 + x_0(1)W^0 + x_0(2)W^0 + x_0(3)W^0 \\ X(1) &= x_0(0)W^0 + x_0(1)W^1 + x_0(2)W^2 + x_0(3)W^3 \\ X(2) &= x_0(0)W^0 + x_0(1)W^2 + x_0(2)W^4 + x_0(3)W^6 \\ X(3) &= x_0(0)W^0 + x_0(1)W^3 + x_0(2)W^6 + x_0(3)W^9 \end{aligned} \quad (2.5.3)$$

Las Ecs. (2.5.3) pueden ser representadas de forma matricial como

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2.5.4)$$

o de forma más compacta

$$\mathbf{X}(n) = \mathbf{W}^{nk} \mathbf{x}_0(k) \quad (2.5.5)$$

Denotando el tipo de letra, en negrita y cursiva, una matriz.

Examinando la Ec. (2.5.4) se observa que debido a que \mathbf{W} y posiblemente $\mathbf{x}_0(k)$ son complejos, son necesarias N^2 multiplicaciones complejas y $(N)(N-1)$ adiciones complejas para desarrollar el cálculo matricial requerido. La FFT debe su éxito al hecho de que el algoritmo reduce el número de multiplicaciones y adiciones requeridas en el cálculo de la Ec. (2.5.4). Ahora vamos a discutir, a nivel intuitivo, cómo se lleva a cabo esta reducción.

2.5.2. Desarrollo intuitivo

Para ilustrar el algoritmo FFT, es conveniente escoger el número de muestras de $x_0(k)$ de acuerdo a la relación $N = 2^\gamma$, donde γ es un entero. En otros desarrollos se puede remover esta restricción (véase [1] y [5]). Recordemos que la Ec. (2.5.4) resulta de tomar $N = 4 = 2^\gamma = 2^2$; por lo tanto, podemos aplicar la FFT para el cálculo de la Ec. (2.5.4).

El primer paso en el desarrollo del algoritmo FFT para este ejemplo es reescribir la Ec. (2.5.4) como

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^0 & W^2 \\ 1 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2.5.6)$$

La matriz de la Ec. (2.5.6) fue derivada de la Ec. (2.5.4) usando la relación $W^{nk} = W^{nk \bmod(N)}$. Recuerde que $[nk \bmod(N)]$ es el resto o residuo de la división de nk por N ; entonces si $N = 4$, $n = 2$ y $k = 3$, entonces

$$W^6 = W^2 \quad (2.5.7)$$

porque

$$\begin{aligned} W^{nk} &= W^6 = \exp \left[\left(\frac{-j2\pi}{4} \right) 6 \right] = \exp(-j3\pi) \\ &= \exp(-j\pi) = \exp \left[\left(\frac{-j2\pi}{4} \right) 2 \right] = W^2 = W^{nk \bmod(N)} \end{aligned} \quad (2.5.8)$$

El segundo paso en el desarrollo es factorizar la matriz cuadrada de la Ec. (2.5.6) como sigue:

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2.5.9)$$

El método de factorización está basado en la teoría del algoritmo FFT que se desarrollará más adelante. Por el momento, es suficiente mostrar que el producto de las matrices cuadradas de la Ec. (2.5.9) resulta en la matriz cuadrada de la Ec. (2.5.6) con la excepción de que las filas 1 y 2 han sido intercambiadas (las filas están numeradas como 0, 1, 2 y 3). Nótese que este intercambio ha sido tomado en cuenta en la Ec. (2.5.9) reescribiendo el vector columna $\mathbf{X}(\mathbf{n})$; denotando al vector con intercambio de filas como

$$\overline{\mathbf{X}(\mathbf{n})} = \begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \quad (2.5.10)$$

Esta factorización es la clave del algoritmo FFT.

Aceptando el hecho que la Ec. (2.5.9) es correcta, a pesar que los resultados están *desorganizados*, se debe examinar el número de multiplicaciones requeridas para calcular la ecuación. Primero, tomemos

$$\begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2.5.11)$$

Entonces, el vector columna $\mathbf{x}_1(\mathbf{k})$ es igual al producto de las dos matrices a la derecha en la Ec. (2.5.9). El elemento $x_1(0)$ es calculado con una multiplicación compleja y una suma compleja (W^0 no es reducido a la unidad para desarrollar el resultado general).

$$x_1(0) = x_0(0) + W^0 x_0(2) \quad (2.5.12)$$

El elemento $x_1(1)$ también es determinado por una multiplicación y una adición compleja. Solo una adición compleja es requerida para calcular $x_1(2)$. Esto debido a que $W^0 = -W^2$; por lo tanto,

$$\begin{aligned} x_1(2) &= x_0(0) + W^2 x_0(2) \\ &= x_0(0) - W^0 x_0(2) \end{aligned} \quad (2.5.13)$$

donde la multiplicación compleja $W^0 x_0(2)$ ya ha sido determinada al calcular $x_1(0)$ [Ec. (2.5.12)]. Por la misma razón, $x_1(3)$ es calculada por una adición compleja sin multiplicación. El vector intermedio $\mathbf{x}_1(\mathbf{k})$ es determinado por cuatro adiciones complejas y dos multiplicaciones complejas.

Siguiendo el cálculo de la Ec. (2.5.9), del resultado anterior obtenemos

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} x_2(0) \\ x_2(1) \\ x_2(2) \\ x_2(3) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad (2.5.14)$$

El término $x_2(0)$ es determinado por una multiplicación y una adición compleja:

$$x_2(0) = x_1(0) + W^0 x_1(1) \quad (2.5.15)$$

El elemento $x_2(1)$ es calculado por una adición debido a que $W^0 = -W^2$. Similarmente, $x_2(2)$ se determina por una multiplicación y una adición compleja, y $x_2(3)$ por solo una adición.

El cálculo de $\overline{\mathbf{X}(\mathbf{n})}$ por medio de la Ec. (2.5.9) requiere un total de cuatro multiplicaciones complejas y ocho adiciones complejas. El cálculo de $\mathbf{X}(\mathbf{n})$ por medio de la Ec. (2.5.4) requiere 16 multiplicaciones complejas y 12 adiciones complejas. Nótese que el proceso de factorización matricial introduce ceros en las matrices factorizadas y, como resultado, reduce el número requerido de multiplicaciones. Para este ejemplo, el proceso de factorización matricial redujo el número de multiplicaciones por un factor de dos. Debido a que el tiempo de cálculo se rige en gran medida por el número de multiplicaciones requeridas, observamos la razón de la eficiencia del algoritmo FFT.

Para $N = 2^\gamma$, el algoritmo FFT es entonces un simple procedimiento de factorizar una matriz de $N \times N$ en γ matrices (cada una de $N \times N$), tal que cada una de las matrices factorizadas tiene la propiedad especial de minimizar el número de multiplicaciones y adiciones complejas. Si generalizamos el resultado del ejemplo previo, se observa que la FFT requiere $N\gamma/2 = 4$ multiplicaciones *complejas* y $N\gamma = 8$ adiciones *complejas*, mientras que el método directo [Ec. (2.5.4)] requiere N^2 multiplicaciones *complejas* y $N(N - 1)$ adiciones *complejas*. Si asumimos que el tiempo de computo es proporcional al número de multiplicaciones, entonces la relación aproximada del tiempo de computo directo al de la FFT esta dada por

$$\frac{N^2}{N\gamma/2} = \frac{2N}{\gamma} \quad (2.5.16)$$

donde si $N = 1024 = 2^{10}$ se tiene una reducción de computo de más de 200 a 1. La Fig. 2.8 ilustra la relación entre el número de multiplicaciones requeridas usando el algoritmo FFT comparado con el número de multiplicaciones usando el método directo.

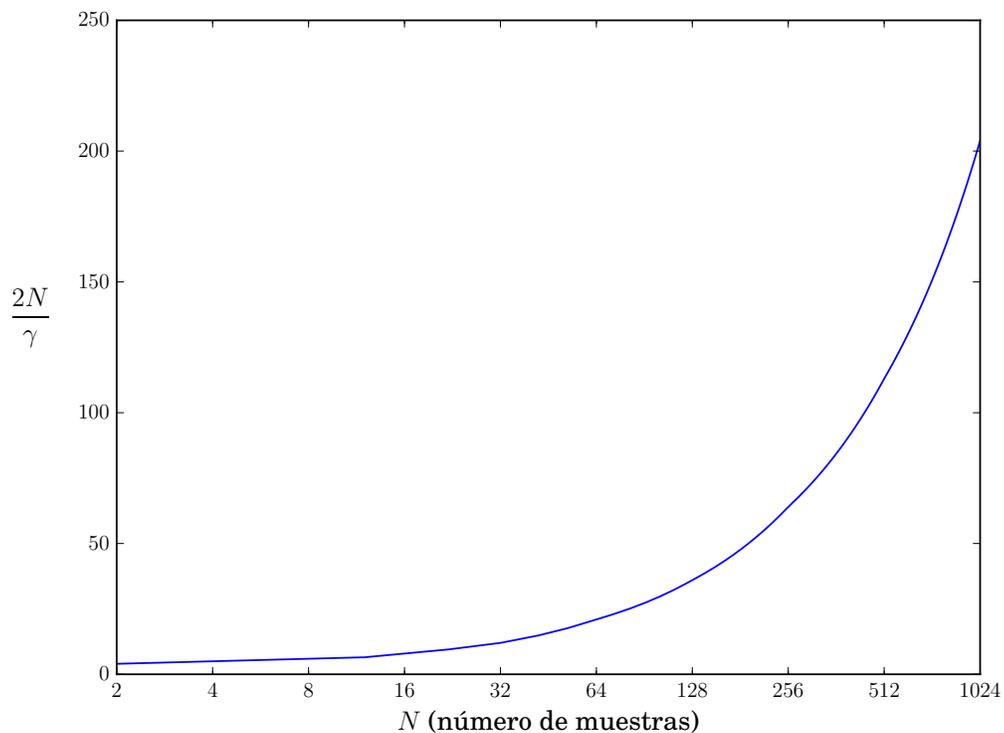


Figura 2.8: Relación de multiplicaciones entre el cálculo directo y la FFT

El proceso de factorización matricial introduce una discrepancia. Recordemos que el computo de la Ec. (2.5.9) arroja $\overline{\mathbf{X}(\mathbf{n})}$ en lugar de $\mathbf{X}(\mathbf{n})$; es decir,

$$\overline{\mathbf{X}(\mathbf{n})} = \begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \text{ en lugar de } \mathbf{X}(\mathbf{n}) = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} \quad (2.5.17)$$

Este reordenamiento es inherente en el proceso de factorización matricial y es un problema menor ya que es fácil generalizar una técnica para *reorganizar* $\overline{\mathbf{X}(\mathbf{n})}$ y obtener $\mathbf{X}(\mathbf{n})$.

Reescribiendo $\overline{\mathbf{X}(\mathbf{n})}$ con el reemplazo del argumento \mathbf{n} por su equivalente binario:

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \text{ se cambia por } \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \quad (2.5.18)$$

Observar que si a los argumentos binarios de la Ec. (2.5.18) se les gira o hace la inversión de bit (p. ej., 01 se cambia por 10, 10 por 01, etc.), entonces

$$\overline{\mathbf{X}(\mathbf{n})} = \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \text{ cambia a } \begin{bmatrix} X(00) \\ X(01) \\ X(10) \\ X(11) \end{bmatrix} = \mathbf{X}(\mathbf{n}) \quad (2.5.19)$$

Es fácil desarrollar un resultado general para reorganizar la FFT.

Para N mayor que 4, es complicado describir el proceso de factorización matricial análogo a la Ec. (2.5.9). Por esta razón, interpretamos (2.5.9) de una forma gráfica. Usando esta formulación gráfica, podemos describir suficientes generalidades para desarrollar un gráfico de flujo para un programa de computadora.

2.5.3. Diagrama de flujo de señal

Ahora representaremos la Ec. (2.5.9) en un diagrama de flujo de señal ilustrado en la Fig. 2.9. Como se muestra, se representa el vector de datos o arreglo $\mathbf{x}_0(\mathbf{k})$ por una

columna vertical de nodos a la izquierda del diagrama. El segundo arreglo vertical de nodos es el vector $\mathbf{x}_1(\mathbf{k})$ calculado en la Ec. (2.5.11), y el siguiente arreglo vertical corresponde al vector $\mathbf{x}_2(\mathbf{k}) = \overline{\mathbf{X}(\mathbf{n})}$, Ec. (2.5.14). En general, habrán γ arreglos computacionales donde $N = 2^\gamma$.

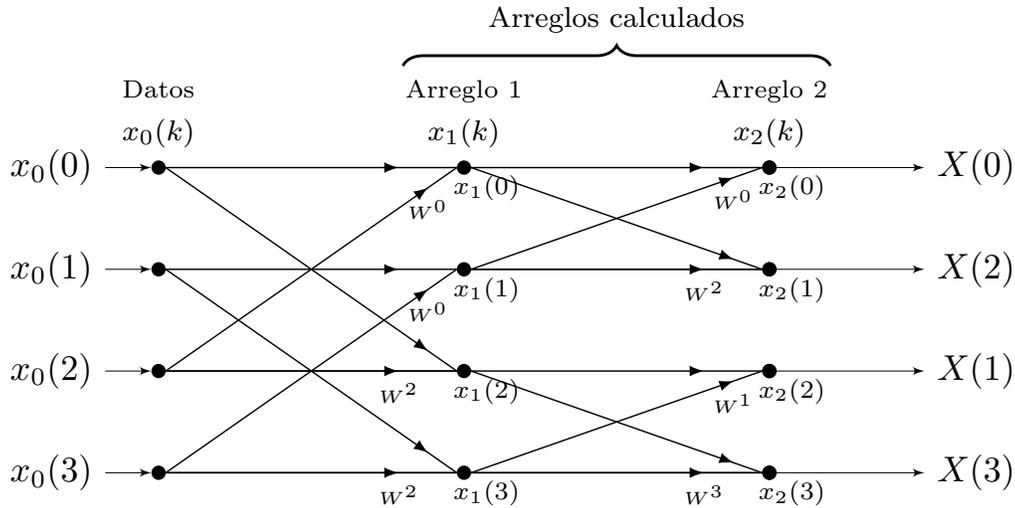


Figura 2.9: Diagrama de flujo de señal para la FFT, $N = 4$

El diagrama de flujo de señal es interpretado como sigue. A cada nodo entran dos líneas representando trayectorias de *transmisión* de los nodos previos. Una trayectoria transmite o lleva una cantidad desde un nodo en un arreglo, multiplica la cantidad por W^p , e ingresa el resultado en el nodo del siguiente arreglo. El factor W^p se encuentra cerca de la punta de la flecha de la trayectoria de transmisión; ausencia de este factor implica que $W^p = 1$. Los resultados que ingresan a un nodo de dos vías de transmisión se combinan de forma aditiva.

Para ilustrar la interpretación del diagrama de flujo de señal, considere el nodo $x_1(2)$ en la Fig. 2.9. De acuerdo a las reglas para la interpretación del diagrama,

$$x_1(2) = x_0(0) + W^2 x_0(2) \tag{2.5.20}$$

que es simplemente la Ec. (2.5.13). Cada nodo del diagrama es expresado de forma similar.

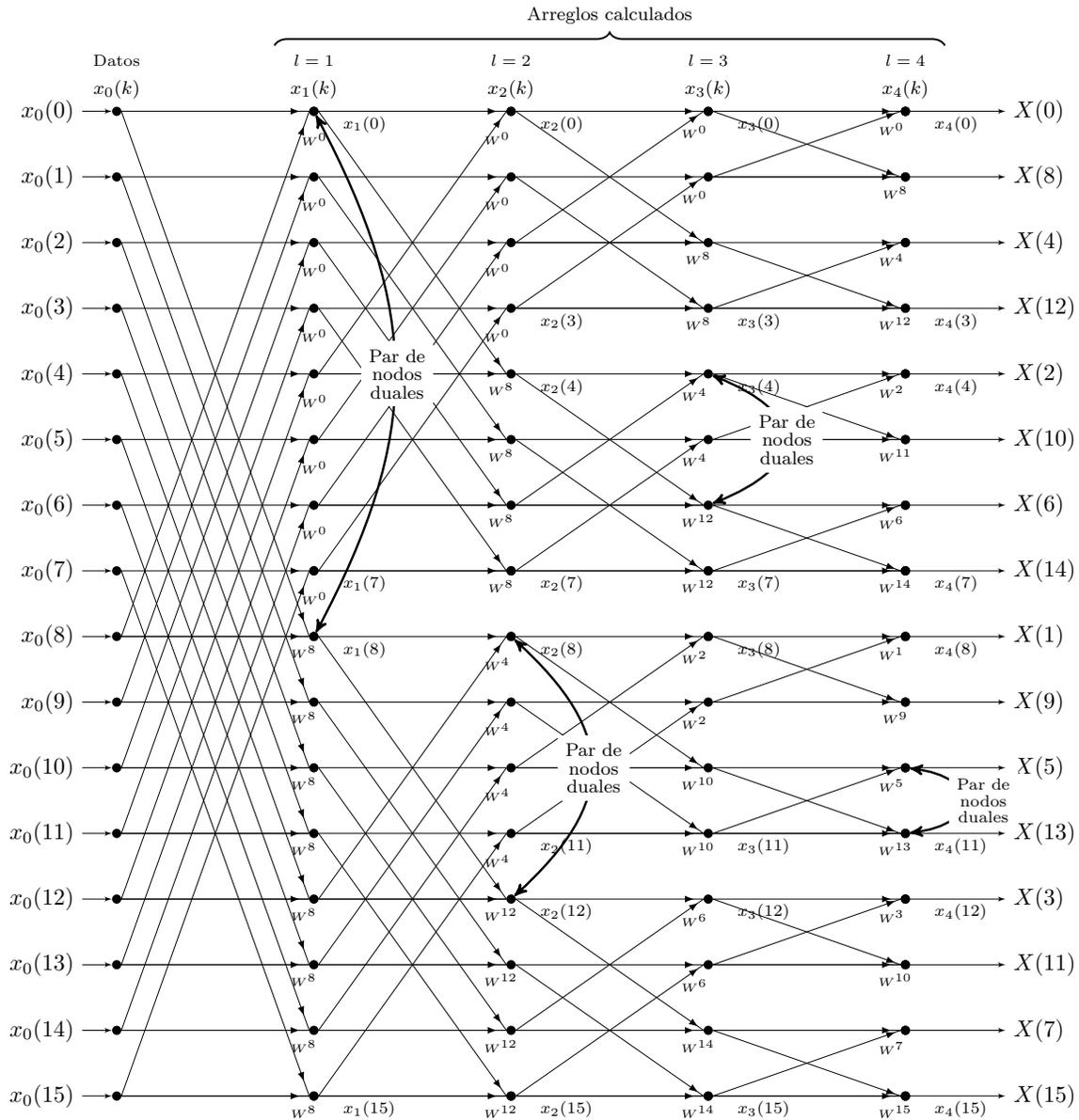


Figura 2.10: Ejemplo de nodos duales

El diagrama de flujo de señal es entonces un método conciso para la representación de los cálculos requeridos en la matriz factorizada del algoritmo FFT de la Ec. (2.5.9). Cada columna calculada en el diagrama corresponde a una matriz factorizada; γ arreglos verticales de N puntos cada uno ($N = 2^\gamma$) son requeridos. El uso de esta representación gráfica nos permite fácilmente describir el proceso de factorización matricial para valores grandes de N .

Se muestra en la Fig. 2.10 el diagrama de flujo de señal para $N = 16$. Con un diagrama de flujo de este tamaño, es posible desarrollar propiedades generales relativas al proceso de factorización matricial y así proporcionar un marco para el desarrollo de un diagrama de flujo para computadora de la FFT.

2.5.4. Nodos duales

Por inspección de la Fig. 2.10 se observa que en cada arreglo siempre podemos encontrar dos nodos cuyas trayectorias de transmisión de entrada provienen del mismo par de nodos del arreglo anterior. Por ejemplo, los nodos $x_1(0)$ y $x_1(8)$ se calculan a partir de los nodos $x_0(0)$ y $x_0(8)$. Nótese que los nodos $x_0(0)$ y $x_0(8)$ no entran en el cálculo de cualquier otro nodo. Definimos a dos nodos con estas características como un *par de nodos duales*.

Debido a que el cálculo de un par de nodos duales es independiente de otros nodos, es posible llevar a cabo el cálculo *en la misma dirección de memoria*. Para ilustrar esto, obsérvese que a partir de la Fig. 2.10 podemos calcular de forma simultánea $x_1(0)$ y $x_1(8)$ en términos de $x_0(0)$ y $x_0(8)$ y devolver los resultados a las direcciones de memoria previamente ocupadas por $x_0(0)$ y $x_0(8)$. Los requerimientos de memoria son entonces limitados al arreglo de datos $\mathbf{x}_0(\mathbf{k})$ solamente. A medida que cada arreglo es calculado, los resultados son devueltos a este arreglo.

Separación de nodos duales. Analicemos la separación (medida verticalmente en términos del índice k) entre un par de nodos duales. A lo siguiente refiérase a la Fig. 2.10. En el arreglo $l = 1$, un par de nodos duales, por ejemplo $x_1(0)$ y $x_1(8)$, están separados por $k = 8 = N/2^l = N/2^1$. En el arreglo $l = 2$, un par de nodos duales, entre ellos $x_2(8)$ y $x_2(12)$, están separados por $k = 4 = N/2^l = N/2^2$. De forma similar, un par de nodos duales, siendo estos $x_3(4)$ y $x_3(6)$, en el arreglo $l = 3$ están separados por $k = 2 = N/2^l = N/2^3$; y en el arreglo $l = 4$, un par de nodos duales, tomando $x_4(10)$ y $x_4(11)$, están separados por $k = 1 = N/2^l = N/2^4$.

Generalizando, se observa que la separación entre nodos duales en el arreglo l está dado por $N/2^l$. Así, si se considera un nodo particular $x_l(k)$, entonces su nodo dual es

$x_l(k + N/2^l)$. Esta propiedad nos permite identificar fácilmente un par de nodos duales.

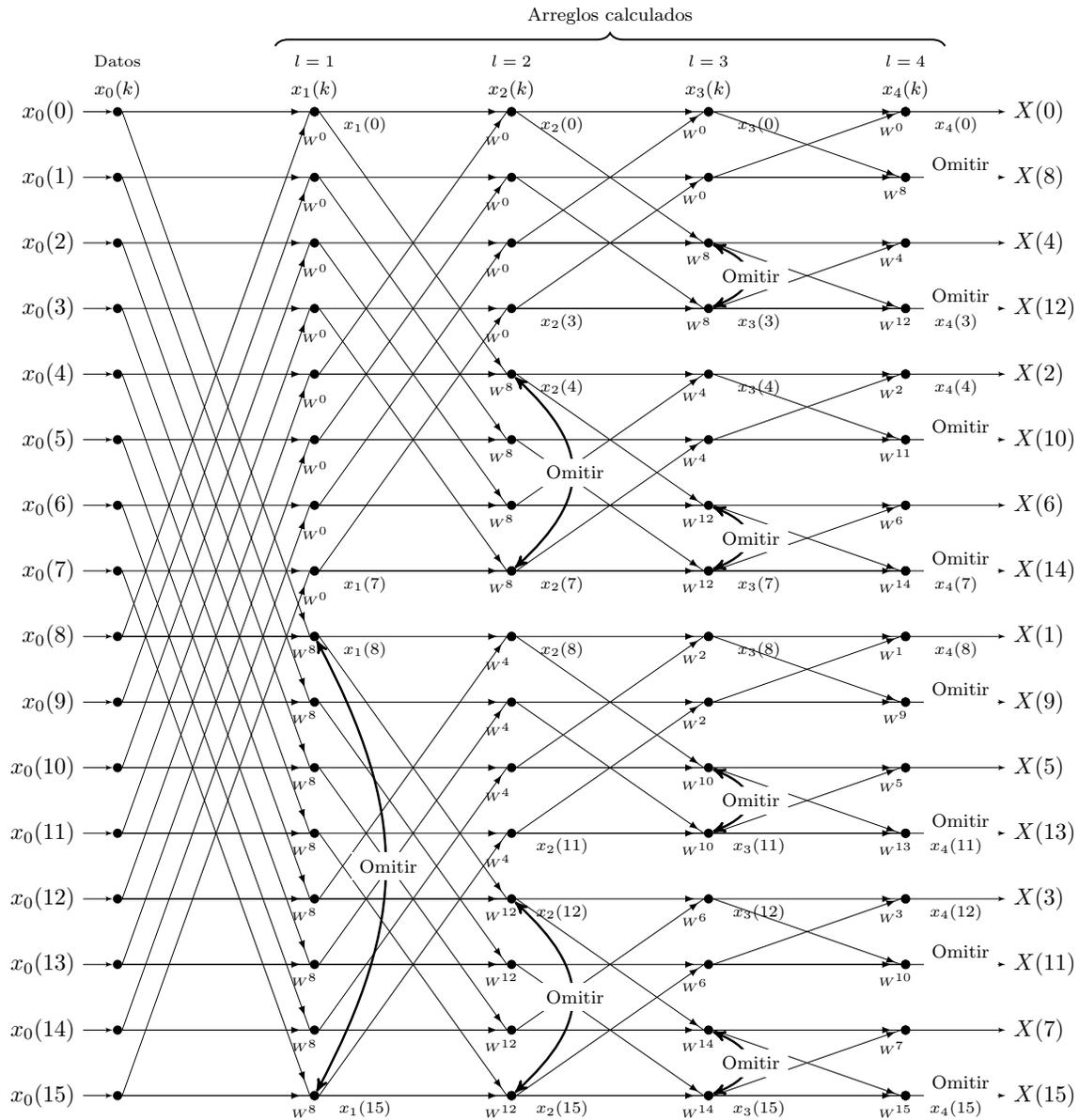


Figura 2.11: Ejemplo de nodos a omitir

Cálculo de nodos duales. El cálculo del par de nodos duales requiere solo una multiplicación compleja. Para clarificar este punto, considere el nodo $x_2(8)$ y su dual $x_2(12)$, como se ilustra en la Fig. 2.10. Las trayectorias de transmisión derivadas del nodo $x_1(12)$ son multiplicadas por W^4 y W^{12} previo a ingresar en los nodos $x_2(8)$ y $x_2(12)$, respectivamente. Es importante notar que $W^4 = -W^{12}$ y que solo una multiplicación es

requerida porque el mismo dato $x_1(12)$ se debe multiplicar por estos términos. En general, si el factor multiplicador en un nodo es W^p , entonces el factor multiplicador en el nodo dual es $W^{p+N/2}$. Debido a que $W^p = -W^{p+N/2}$, solo una multiplicación es requerida en el cálculo de un par de nodos duales. El cálculo de cualquier par de nodos duales esta dado por el par de ecuaciones:

$$\begin{aligned} x_l(k) &= x_{l-1}(k) + W^p x_{l-1}(k + N/2^l) \\ x_l(k + N/2^l) &= x_{l-1}(k) - W^p x_{l-1}(k + N/2^l) \end{aligned} \quad (2.5.21)$$

En el cálculo de un arreglo, normalmente se comienza con el nodo $k = 0$ y secuencialmente se continua calculando en el arreglo, utilizando el par de ecuaciones de la Ec. (2.5.21). Como se apuntó previamente, el dual de cualquier nodo en el l -ésimo arreglo está siempre a $N/2^l$ nodos en el arreglo. Debido a que el espaciado es $N/2^l$, entonces se concluye que se deben *omitir* después de cada $N/2^l$ nodos, los siguientes $N/2^l$ nodos. Para ilustrar este punto, considere el arreglo $l = 2$ en la Fig. 2.11. Si se comienza con el nodo $k = 0$, entonces de acuerdo a lo expuesto anteriormente, el nodo dual está localizado en $k = N/2^2 = 4$, que puede ser verificado por inspección en la Fig. 2.11. Continuando en el arreglo, se observa que el nodo dual está siempre a 4 nodos en el arreglo hasta que se alcanza el nodo 4. En este punto, tenemos un conjunto de nodos calculados indirectamente, es decir, tenemos los duales de los nodos $k = 0, 1, 2$ y 3 . Es necesario *omitir* los nodos $k = 4, 5, 6$ y 7 . Los nodos $8, 9, 10$ y 11 siguen la convención original del nodo dual localizado a 4 nodos en el arreglo. En general, si se trabaja de arriba hacia abajo en el l -ésimo arreglo, entonces con la Ec. (2.5.21) se calcularán los primero $N/2^l$ nodos, se omitirán los siguientes $N/2^l$ y así sucesivamente hasta que se alcance el nodo con índice mayor a $N - 1$.

2.5.5. Determinación de W^p

Basados en lo argumentado anteriormente, se han definido las propiedades de cada arreglo con excepción del valor p en la Ec. (2.5.21). El valor de p esta determinado por (a) la representación del índice k en forma binaria con γ bits, (b) escalar o deslizar este número binario $\gamma - l$ bits a la derecha, rellenando las posiciones de bits liberados a la

izquierda con ceros, y al final (*c*) invertir el orden de los bits. El número con bits invertidos es el término p .

Para ilustrar este procedimiento, refiérase a la Fig. 2.11 y considere el nodo $x_3(8)$. Debido a que $\gamma = 4$, $k = 8$ y $l = 3$, entonces k en binario es 1000. Se escala este número $\gamma - l = 4 - 3 = 1$ posición a la derecha y rellenamos con ceros; el resultado es 0100. Entonces se invierte el orden de los bits para obtener 0010 o el entero 2. El valor de p es entonces 2.

En el Apéndice E se detalla un procedimiento para encontrar el número binario y realizar la inversión de bit.

2.5.6. Reorganizando la FFT

El paso final en el cálculo de la FFT es *reorganizar* los resultados de acuerdo a la Ec. (2.5.19). Recordando que el procedimiento para reorganizar el vector $\overline{\mathbf{X}(\mathbf{n})}$ es representar el índice n en binario y luego invertir o girar dicho número binario. En la Fig. 2.12 se muestra el resultado de la operación de inversión de bit: los términos $x_4(k)$ y $x_4(i)$ simplemente han sido intercambiados, donde i es el entero obtenido por la inversión de bit del entero k .

Obsérvese que una situación similar al concepto de nodo dual existe cuando se reorganiza el arreglo de salida. Si se opera en el arreglo, intercambiando $x(k)$ con el apropiado $x(i)$, eventualmente se encontrará con un nodo que ha sido previamente intercambiado. Por ejemplo, en la Fig. 2.12, el nodo $k = 0$ permanece en su ubicación, los nodos $k = 1, 2$ y 3 son intercambiados con los nodos $8, 4$ y 12 respectivamente. El siguiente nodo a ser considerado es el nodo 4 , pero este nodo fue previamente intercambiado con el nodo 2 . Para eliminar la posibilidad de considerar un nodo que ha sido previamente intercambiado, simplemente se verifica si i (el entero obtenido de la inversión de bit de k) es menor que k . Si es así, implica que el nodo ha sido intercambiado por una operación previa. Con esta verificación, se puede fácilmente asegurar el proceso de reorganizado.

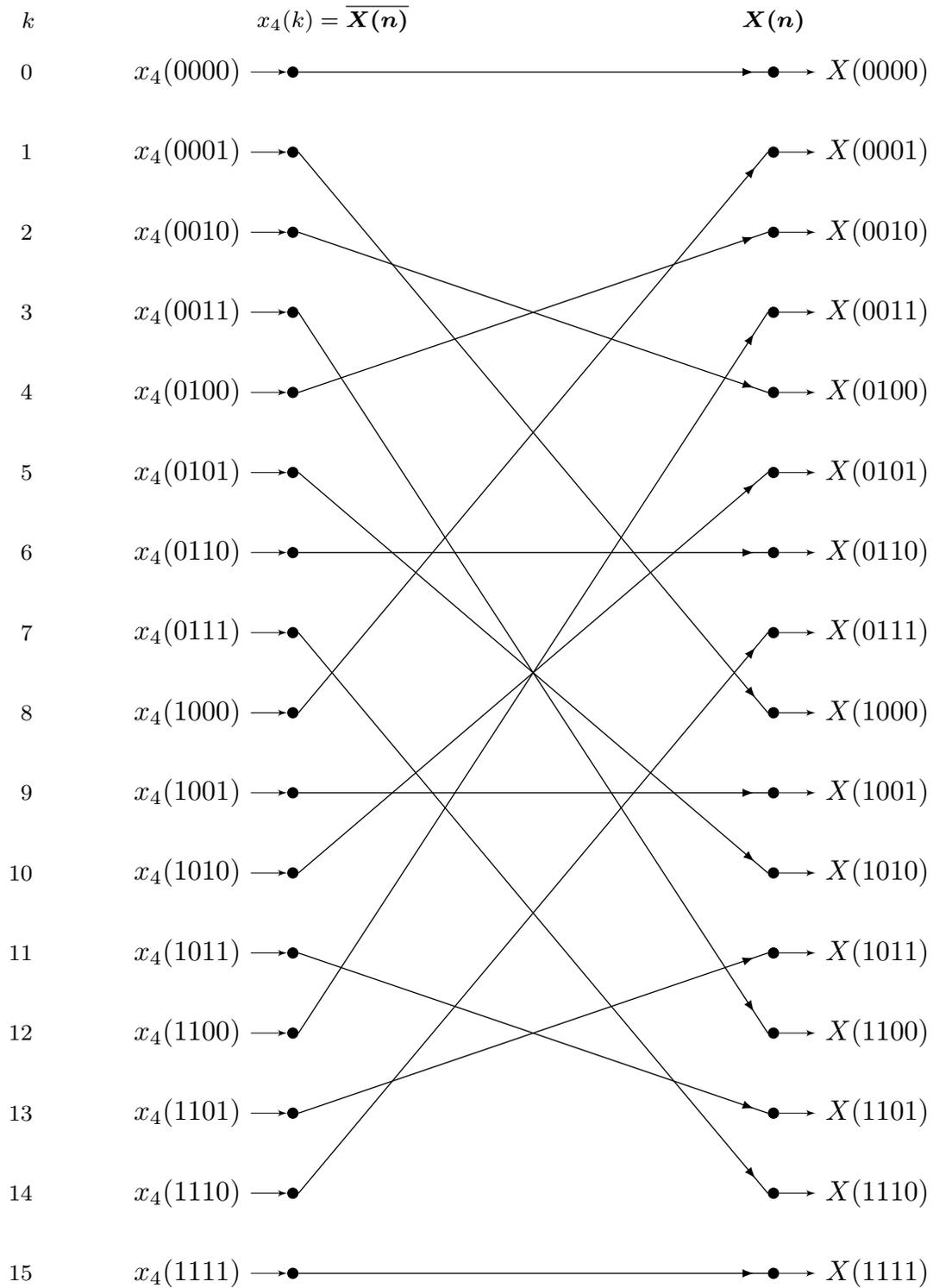


Figura 2.12: Ejemplo de la operación de inversión de bit para $N = 16$

2.5.7. Desarrollo teórico del algoritmo FFT de base-2

En la Subsección 2.5.2, se usó un argumento matricial para desarrollar un entendimiento del porqué la FFT es un algoritmo eficiente. Se contruyó entonces un diagrama de flujo de señal que describe el algoritmo para cualquier $N = 2^\gamma$. Ahora relacionaremos cada uno de estos argumentos en una base teórica. Se desarrollará una prueba teórica del algoritmo para el caso de $N = 4$. Para el caso $N = 2^\gamma$, donde γ es un valor entero, se puede consultar la formulación original de Cooley-Tukey [9] y el libro de Brigham [5].

Definición de la notación. Considérese la transformada de Fourier discreta presentada en la Ec. (2.5.1)

$$X(n) = \sum_{k=0}^{N-1} x_0(k)W^{nk}, \quad n = 0, 1, \dots, N-1 \quad (2.5.22)$$

donde se define $W = e^{-j2\pi/N}$. Es conveniente representar los enteros n y k como números binarios; por lo tanto, si se toma $N = 4$, entonces $\gamma = 2$ y se pueden representar k y n como números binarios de dos bits,

$$\begin{aligned} k = 0, 1, 2, 3 & \quad \text{o} \quad k = (k_1, k_0) = 00, 01, 10, 11 \\ n = 0, 1, 2, 3 & \quad \text{o} \quad n = (n_1, n_0) = 00, 01, 10, 11 \end{aligned}$$

Un método compacto de escribir k y n es

$$k = 2k_1 + k_0 \quad n = 2n_1 + n_0 \quad (2.5.23)$$

donde k_0, k_1, n_0 y n_1 pueden tomar solamente los valores de 0 y 1. La Ec. (2.5.23) es simplemente el método para escribir un número binario en su equivalente en base-10.

Usando la representación de la Ec. (2.5.23), se puede reescribir la Ec. (2.5.22) para el caso de $N = 4$ como

$$X(n_1, n_0) = \sum_{k_0=0}^1 \sum_{k_1=0}^1 x_0(k_1, k_0)W^{(2n_1+n_0)(2k_1+k_0)} \quad (2.5.24)$$

Observe que la única sumatoria en la Ec. (2.5.22) debe ser reemplazada por γ sumatorias con el fin de evaluar todos los bits de la representación binaria de k .

Factorización de W^p . Ahora considere el termino W^p . Debido a que $W^{a+b} = W^a W^b$, entonces

$$\begin{aligned} W^{(2n_1+n_0)(2k_1+k_0)} &= W^{(2n_1+n_0)2k_1} W^{(2n_1+n_0)k_0} \\ &= (W^{4n_1k_1}) W^{2n_0k_1} W^{(2n_1+n_0)k_0} \\ &= W^{2n_0k_1} W^{(2n_1+n_0)k_0} \end{aligned} \quad (2.5.25)$$

Nótese que el termino entre paréntesis es igual a la unidad debido a que

$$W^{4n_1k_1} = (W^4)^{n_1k_1} = (e^{-j2\pi \cdot 4/4})^{n_1k_1} = (1)^{n_1k_1} = 1 \quad (2.5.26)$$

Por lo tanto, la Ec. (2.5.24) puede ser escrita en la forma:

$$X(n_1, n_0) = \sum_{k_0=0}^1 \left[\sum_{k_1=0}^1 x_0(k_1, k_0) W^{2n_0k_1} \right] W^{(2n_1+n_0)k_0} \quad (2.5.27)$$

Esta ecuación representa el fundamento del algoritmo FFT. Para demostrar este punto, considere individualmente cada una de las sumatorias de la Ec. (2.5.27). Primero, reescriba la sumatoria entre corchetes como

$$x_1(n_0, k_0) = \sum_{k_1=0}^1 x_0(k_1, k_0) W^{2n_0k_1} \quad (2.5.28)$$

Enumerando las ecuaciones representadas por la Ec. (2.5.28), se obtiene

$$\begin{aligned} x_1(0, 0) &= x_0(0, 0) + W^0 x_0(1, 0) \\ x_1(0, 1) &= x_0(0, 1) + W^0 x_0(1, 1) \\ x_1(1, 0) &= x_0(0, 0) + W^2 x_0(1, 0) \\ x_1(1, 1) &= x_0(0, 1) + W^2 x_0(1, 1) \end{aligned} \quad (2.5.29)$$

Si reescribimos la Ec. (2.5.29) en notación matricial, obtenemos

$$\begin{bmatrix} x_1(0, 0) \\ x_1(0, 1) \\ x_1(1, 0) \\ x_1(1, 1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{bmatrix} x_0(0, 0) \\ x_0(0, 1) \\ x_0(1, 0) \\ x_0(1, 1) \end{bmatrix} \quad (2.5.30)$$

Nótese que la Ec. (2.5.30) es exactamente la matriz factorizada de la Ec. (2.5.11), desarrollada en la Subsección 2.5.2, con el índice k escrito en notación binaria. Por lo tanto, la sumatoria interna de la Ec. (2.5.27) especifica la primera de las matrices factorizadas para el ejemplo desarrollado en la Subsección 2.5.2 o, equivalentemente, el arreglo $l = 1$ de el diagrama de flujo de señal ilustrado en la Fig. 2.9.

De forma similar, si escribimos la sumatoria externa de la Ec. (2.5.27) como

$$x_2(n_0, n_1) = \sum_{k_0=0}^1 x_1(n_0, k_0) W^{(2n_1+n_0)k_0} \quad (2.5.31)$$

y representamos el resultado en forma matricial, se obtiene

$$\begin{bmatrix} x_2(0,0) \\ x_2(0,1) \\ x_2(1,0) \\ x_2(1,1) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} \quad (2.5.32)$$

la cual es la Ec. (2.5.14). Así, la sumatoria externa de la Ec. (2.5.27) determina la segunda de las matrices factorizadas del ejemplo de la Subsección 2.5.2.

De las Ecs. (2.5.27) y (2.5.31) tenemos

$$X(n_1, n_0) = x_2(n_0, n_1) \quad (2.5.33)$$

Es decir, los resultados finales $x_2(n_0, n_1)$ tal como se obtuvieron de la sumatoria externa están en orden inverso de bit respecto a los valores deseados $X(n_1, n_0)$. Esto es simplemente la desorganización que resulta del algoritmo FFT.

Si combinamos las Ecs. (2.5.28), (2.5.31) y (2.5.33),

$$\begin{aligned} x_1(n_0, k_0) &= \sum_{k_1=0}^1 x_0(k_1, k_0) W^{2n_0k_1} \\ x_2(n_0, n_1) &= \sum_{k_0=0}^1 x_1(n_0, k_0) W^{(2n_1+n_0)k_0} \\ X(n_1, n_0) &= x_2(n_0, n_1) \end{aligned} \quad (2.5.34)$$

donde el conjunto de la Ec. (2.5.34) representa la formulación original de Cooley-Tukey [9] del algoritmo FFT para $N = 4$. A este tipo de ecuaciones se les conoce como *recursivas*, ya que la segunda se calcula a partir de la primera.

Capítulo 3

Diseño del programa en Python para cálculos de FFT

3.1. Contexto

Como se apuntó inicialmente, el lenguaje base para la programación de la FFT en este trabajo es Python, que es un lenguaje interpretado y debido a esto, cuando se tiene un gran número de muestras la velocidad de procesamiento se ve reducida; por lo que para dar velocidad y poder de procesamiento a los cálculos se utilizará un módulo de Fortran (lenguaje compilado) que será utilizado por Python para realizar los cálculos. La versión de Python utilizada es la 2.7 por ser la estable de las últimas.

Python tiene desarrolladas muchas extensiones que le dan mayor versatilidad, una de ellas es *Numpy* que es el paquete fundamental para el cálculo científico, que contiene entre otras cosas:

- un poderoso arreglo de objetos N-dimensionales
- funciones (transmitidas) sofisticadas
- herramientas para la integración de código C/C++ y Fortran
- útiles funciones de álgebra lineal, transformada de Fourier y números aleatorios

Como se indica, Numpy contiene dentro de sus ventajas varias funciones para transformadas de Fourier, que nos servirán para comparar los tiempos de cálculo y los datos obtenidos.

3.2. Programa FFT

A continuación se presenta el programa que se desarrolló utilizando únicamente Python, en el cual se utiliza como prueba la función $A\text{sen}(2\pi f_0 t)$, tomando $A = 1$, $f_0 = 250$ Hz y $N = 2^{10} = 1024$ muestras:

```
1 ##Transformada rápida de Fourier
2 from __future__ import print_function, division
3 from pylab import *
4
5 ##Variables de entrada
6 ##xr = parte real de los datos del muestreo
7 ##xi = parte imaginaria de los datos del muestreo
8 ##W = exp(-1j * 2 * np.pi / N)
9 ##T1 = valor temporal consistente en (Wp)* x(k + N2)
10 xr = []
11 xi = []
12 nu = 10
13 n = np.power(2, nu)
14
15 ##Datos de pruebas
16 k_muestras = np.arange(n)
17 ##Frecuencia y periodo función seno
18 f_0 = 250 ##Frecuencia
19 T_0 = 1 / f_0 ##Periodo
20 nT = 20 ##Veces el periodo a graficar
21 T = nT * T_0
```

```

22 delta_t = 1 / n                                ##factor entre TFC y TFD
23 delta_t1 = T / n                               ##Delta_t del muestreo
24 delta_f = 1 / T                               ##Delta_f del muestreo
25 t = np.array(delta_t1 * k_muestras)
26 xr = np.sin(2 * np.pi * f_0 * t)
27 xi = [0] * n
28 xz = np.array(xr) + 1j * np.array(xi)
29
30 ##Inicializar variables auxiliares
31 l = 1
32 n2 = np.int_(n / 2)
33 nu1 = nu - 1
34 k = 0
35
36 ##Implementación Python-Inicio
37 xza = np.array(xz)
38 while l <= nu:
39     while k <= n - 1:
40         q = 1
41         while q <= n2:
42             M = np.int_(k / np.power(2, nu1))
43             P = np.int_('{:0{width}b}'.format(M, width = nu)
44                 [::-1], 2)
45             W = np.exp(- 1j * 2 * np.pi / n)
46             T1 = xza[k + n2] * W**P
47             xza[k + n2] = xza[k] - T1
48             xza[k] = xza[k] + T1
49             k += 1
50             q +=1
51         k += n2

```

```
52     l += 1
53     n2 = np.int_(n2 / 2)
54     nu1 -= 1
55     k = 0
56 while k < n - 1:
57     i = np.int_('{:0{width}b}'.format(k, width = nu)
58               [::-1], 2)
59     if i > k:
60         T3 = xza[k]
61         xza[k] = xza[i]
62         xza[i] = T3
63     k += 1
64 xza *= delta_t
65 ##Implementación Python-Final
66
67 ##Ajuste de datos de frecuencia
68 f = []
69 for s in range(len(k_muestras)):
70     if s <= (n / 2 + 1):
71         f.append(s * delta_f)
72     else:
73         f.append((s - n) * delta_f)
74
75 ##Gráficos
76 grafico = plt.figure()
77 subplot1 = plt.subplot(2,1,1)
78 plt.plot(t, xr)
79 plt.title('Grafico_funcion_del_tiempo')
80 plt.xlabel('tiempo_(s)')
81 plt.ylabel('Amplitud')
```

```

82 subplot1.set_ylim([min(min(xr), min(xi)) * 1.2,
83                    max(max(xr), max(xi)) * 1.2])
84 subplot2 = plt.subplot(2,1,2)
85 plt.plot(f,xza.real,f,xza.imag)
86 plt.title('Grafico implementacion Python')
87 plt.xlabel('Frecuencia (Hz)')
88 plt.ylabel('Amplitud')
89 subplot2.set_xlim(-4000,4000)
90 subplot2.set_xticks([-4000,-2000,-250,0,250,2000,4000])
91 subplot2.set_xticklabels([-4000,-2000,-250,0,250,2000,4000],rotation=90)
92 subplot2.set_ylim([min(min(xza.real), min(xza.imag)) * 1.2,
93                    max(max(xza.real), max(xza.imag)) * 1.2])
94 plt.tight_layout(pad=0.8, w_pad=0.5, h_pad=1.0)
95 plt.show()

```

En la Fig. 3.1 se muestran los datos obtenidos para el caso mencionado $[A \text{sen}(2\pi f_0 t)]$, y al comparar con el par transformado (Apéndice C)

$$A \text{sen}(2\pi f_0 t) \Leftrightarrow -j\frac{A}{2}\delta(f - f_0) + j\frac{A}{2}\delta(f + f_0) \quad (3.2.1)$$

se observa coincidencia con los resultados arrojados por el programa.

Ademas en la Fig. 3.2 se observa el caso para $A \text{cos}(2\pi f_0 t)$, tomando $A = 1$, $f_0 = 250$ Hz y $N = 2^{10} = 1024$ muestras y cuyo par transformado de Fourier es

$$A \text{cos}(2\pi f_0 t) \Leftrightarrow \frac{A}{2}\delta(f - f_0) + \frac{A}{2}\delta(f + f_0) \quad (3.2.2)$$

Nótese el color de línea diferente en el gráfico, en contraste con la Fig. 3.1.

Precisamente, ya que la transformada de Fourier en general es una función compleja [Eq. (2.1.2)], en el programa la parte real $[R(f)]$ aparecerá con línea color azul y la parte imaginaria $[I(f)]$ en color verde. Para ejemplo tómese la función:

$$h(t) = \begin{cases} \beta e^{-\alpha t} & t > 0 \\ 0 & t < 0 \end{cases} \quad (3.2.3)$$

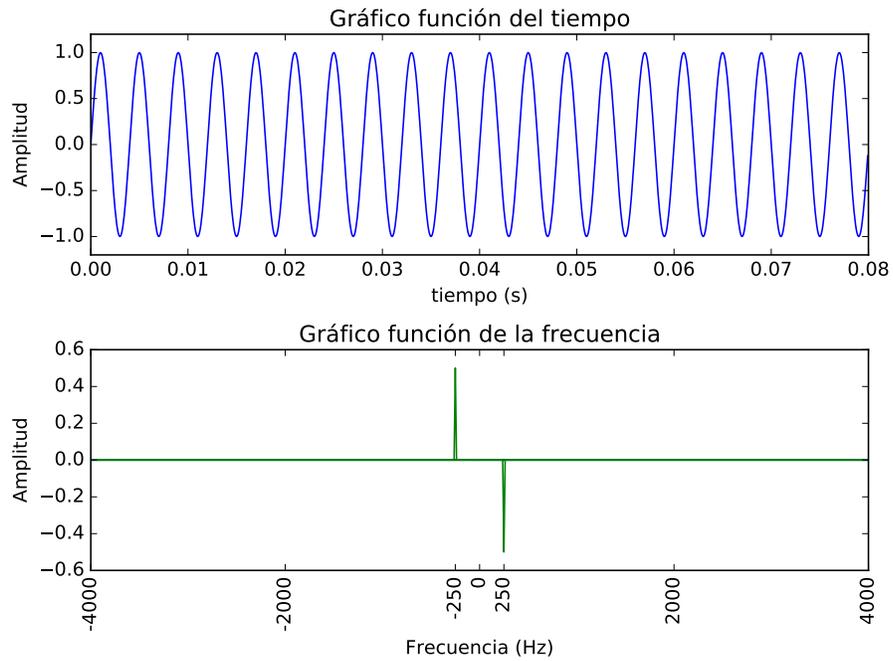


Figura 3.1: Gráfico de la función $A \sin(2\pi f_0 t)$ y su transformada. $A = 1$ y $f_0 = 250$ Hz

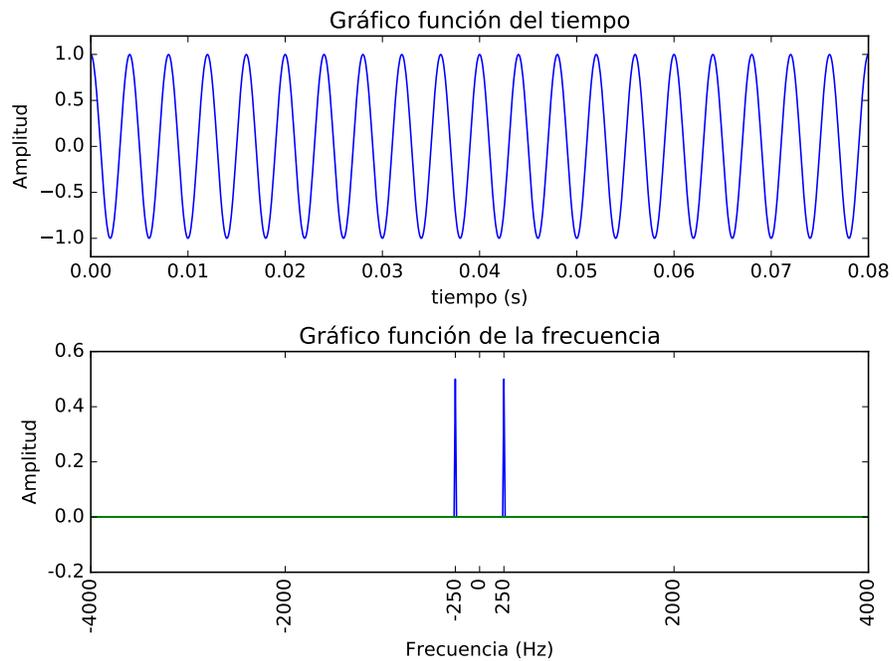


Figura 3.2: Gráfico de la función $A \cos(2\pi f_0 t)$ y su transformada. $A = 1$ y $f_0 = 250$ Hz

cuya transformada de Fourier es:

$$\begin{aligned} H(f) &= R(f) + jI(f) \\ &= \frac{\beta\alpha}{\alpha^2 + (2\pi f)^2} - \frac{j2\pi f\beta}{\alpha^2 + (2\pi f)^2} \end{aligned} \quad (3.2.4)$$

En la Fig. 3.3 se muestra el gráfico correspondiente

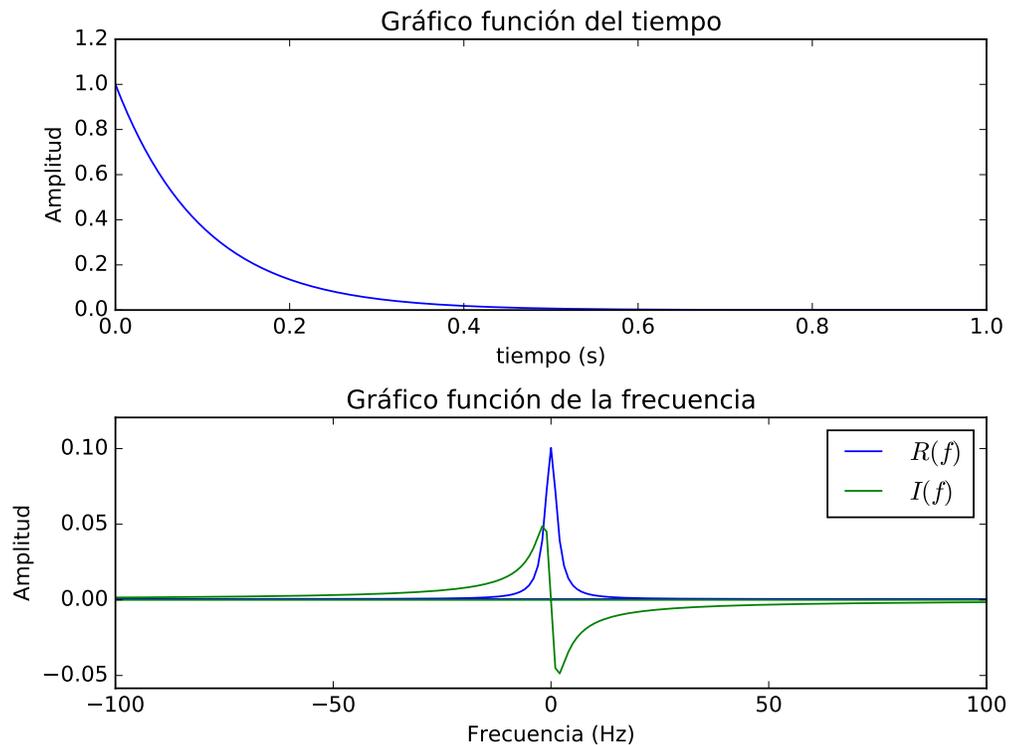


Figura 3.3: Gráfico de $e^{-\alpha t}$ para $t > 0$ y su transformada. $\alpha = 10$ Hz

3.3. Metodología para el uso del programa

Para los ejemplos anteriores se utilizó el mismo programa, lo que cambia en algunas situaciones es considerar que la presentación del gráfico se debe ajustar en función de los datos obtenidos de la FFT. Como se apuntó en la Sección 2.3, el concepto más importante es mantener en mente que la transformada discreta de Fourier implica periodicidad tanto en el dominio del tiempo como en el de la frecuencia. Si se recuerda que las N muestras

de la función en el dominio del tiempo representan un periodo de la función periódica, entonces la aplicación del programa FFT debe resultar en pocos resultados sorprendidos.

Los bloques que se deben considerar para los datos y gráficos, son los que corresponden a las líneas de la 17 a la 28 (datos de entrada para el ejemplo de la función *seno*) y de la línea 75 en adelante.

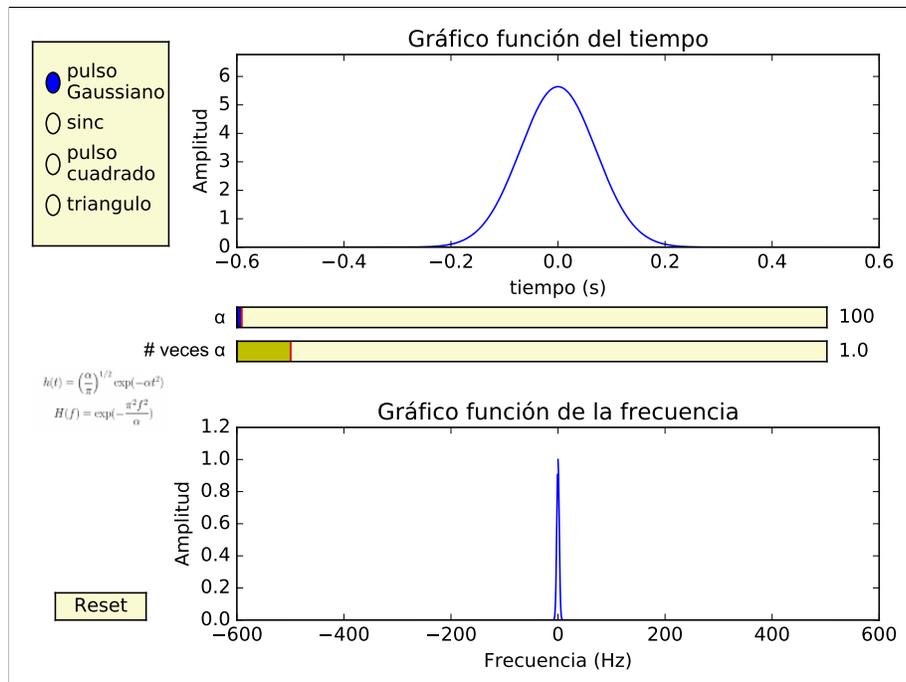


Figura 3.4: Módulo para interpretar el comportamiento de la transformada de Fourier

Además, se desarrolló un módulo con el cual se pretende ayudar en la interpretación del comportamiento de la transformada de Fourier. En la Fig. 3.4 se muestra una imagen del mismo, en el cual al momento de modificar ciertos parámetros el gráfico mostrará el resultado y con esto se confirmará o corregirá lo esperado en cuanto a los cambios.

3.4. Taller de validación del programa

Para observar el comportamiento del programa se realizó un pequeño taller con alumnos, instructores y catedráticos del Departamento de Física de la UNAH-VS. Durante la sesión se explicó las generalidades de la transformada continua y discreta de Fourier, así

como el algoritmo FFT. Después de esto, con la ayuda de la extensión de Python llamada PyInstaller, se generó un archivo ejecutable de Windows con el cual no se necesita tener instalado Python para poder correr el programa que genera la aplicación para la ayudar en la interpretación de la transformada de Fourier (Fig. 3.4). Para realizar esto último basta con instalar PyInstaller y con línea de comando enviar “`pyinstaller pyfftapp.spec`”, donde el archivo `pyfftapp.spec` le indica a PyInstaller de módulos y archivos únicos para el código. El archivo de especificaciones utilizado es el siguiente:

```
1 # -*- mode: python -*-
2 a = Analysis(['C:\\Users\\michael.spilsbury\\Python27\\tesis\\
      python_fortran\\modif\\modif9.py'],
3             pathex=['C:\\Users\\michael.spilsbury\\dist'],
4             hiddenimports=[],
5             hookspath=None,
6             runtime_hooks=None)
7 a.datas += [('im12.png', 'C:\\Users\\michael.spilsbury\\Python27\\
      tesis\\python_fortran\\modif\\im12.png', 'DATA'),\
8             ('im22.png', 'C:\\Users\\michael.spilsbury\\Python27\\
      tesis\\python_fortran\\modif\\im22.png', 'DATA'),\
9             ('im32.png', 'C:\\Users\\michael.spilsbury\\Python27\\
      tesis\\python_fortran\\modif\\im32.png', 'DATA'),\
10            ('im42.png', 'C:\\Users\\michael.spilsbury\\Python27\\
      tesis\\python_fortran\\modif\\im42.png', 'DATA'),]
11 pyz = PYZ(a.pure)
12 exe = EXE(pyz,
13           a.scripts,
14           exclude_binaries=True,
15           name='pyfftapp.exe',
16           debug=False,
17           strip=None,
18           upx=True,
```

```
19         console=True )
20 coll = COLLECT(exe ,
21               a.binaries ,
22               a.zipfiles ,
23               a.datas ,
24               strip=None ,
25               upx=True ,
26               name='pyfftapp ')
```

Información acerca de PyInstaller se puede encontrar en la página <http://www.pyinstaller.org/>

Se puede concluir que el modulo es amigable y arroja los resultados esperados. Al mismo tiempo se corrigieron errores de sintaxis en variables. Por lo demás el programa fue aceptado por los participantes del taller.

Capítulo 4

Análisis de resultados del programa

4.1. Datos

Revisando el tiempo de ejecución del programa utilizando código Python únicamente, le tomaba alrededor de 200 veces el tiempo con Numpy. Esta diferencia se debe a que Numpy realiza el cálculo basado en Fortran, por lo que se elaboró un modulo de Fortran que es llamado por Python y de esta forma reducir el tiempo de ejecución de las rutinas de cálculo. Con esto se obtuvo una mejora notable, casi igual al tiempo de la extensión Numpy. Por ejemplo, para la función pulso cuadrado, que forma el par transformado,

$$h(t) = \begin{cases} A & |t| < T_0 \\ A/2 & |t| = T_0 \\ 0 & |t| > T_0 \end{cases} \Leftrightarrow H(f) = 2AT_0 \frac{\text{sen}(2\pi T_0 f)}{2\pi T_0 f} \quad (4.1.1)$$

con un número de muestras igual a 1024, se realizó el calculo por los tres métodos obteniéndose los siguientes tiempos de ejecución:

Tiempo implementación Python:	0.064370020345 s
Tiempo implementación Python_Fortran:	0.000663668916201 s
Tiempo implementación FFT_Numpy:	0.000339165116232 s

En cuando a los errores obtenidos respecto de la función continua, arrojó (el error

relativo se calculó de la suma de las diferencias entre el número de muestras multiplicado por 100):

Diferencia mayor en los datos = 0.000607593968998

Error relativo = 0.0178151019639

A continuación se muestran las figuras obtenidas los tres métodos

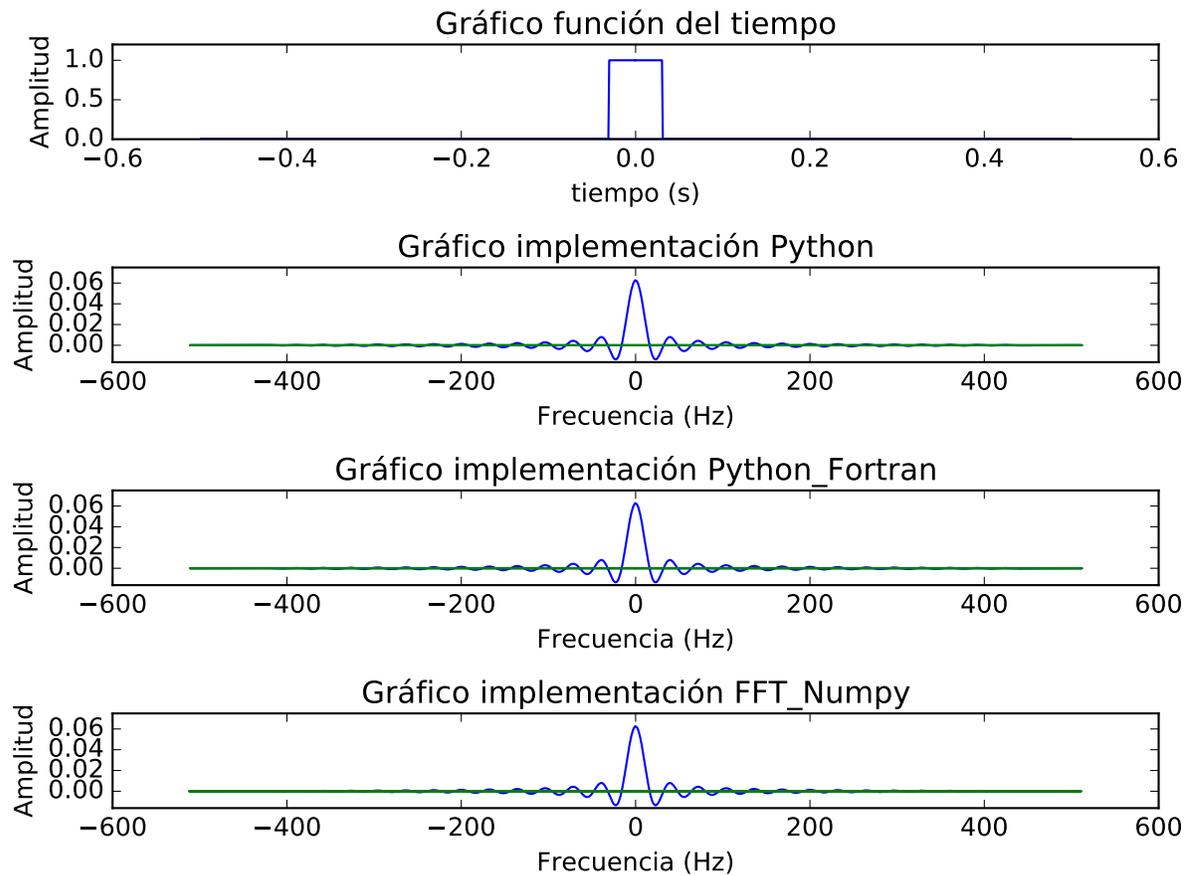


Figura 4.1: Comparación gráfica de los tres métodos para la FFT

Bibliografía

- [1] G. D. Bergland. A fast Fourier transform algorithm using base 8 iterations. *Mathematics of Computation*, 22(102):275, 1968.
- [2] G. D. Bergland. A guided tour of the fast Fourier transform. *IEEE Spectrum*, 6(7):41–52, 1969.
- [3] Ronald N. Bracewell. *The Fourier transform and its applications*. McGraw-Hill electrical and electronic engineering series. McGraw-Hill, New York, 2d ed edition, ©1978.
- [4] William L. Briggs and Emden van Henson, editors. *The DFT: An owner's manual for the discrete Fourier transform*. Society for Industrial and Applied Mathematics, Philadelphia, 1995.
- [5] E. Oran Brigham, editor. *The fast Fourier transform and its applications*. Prentice Hall, Englewood Cliffs, N.J., ©1988.
- [6] Steven C. Chapra, Raymond P. Canale, Javier Enríquez Brito, and Roa Hano, Ma. del Carmen, editors. *Métodos numéricos para ingenieros*. McGraw-Hill Interamericana, México, 5a ed edition, 2007.
- [7] J. W. Cooley, P.A.W. Lewis, and P. D. Welch. Historical notes on the fast Fourier transform. *Proceedings of the IEEE*, 55(10):1675–1677, 1967.
- [8] James W. Cooley, Peter A. W. Lewis, and Peter D. Welch. The fast Fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34, 1969.

-
- [9] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297, 1965.
- [10] Departamento de Electrónica. Fast Fourier transform (FFT). Universidad Distrital Francisco José de Caldas.
- [11] Departamento de Ingeniería Eléctrica y Computadoras. Métodos rápidos para el cálculo de la TDF. Universidad Nacional del Sur.
- [12] Dima Batenkov. Fast Fourier transform: key papers in computer science, seminar 2005. *Weizmann Institute of Science*, 2005.
- [13] D. Donnelly. The fast Fourier transform for experimentalists, part V: filters. *Computing in Science & Engineering*, 8(1):92–95, 2006.
- [14] D. Donnelly. The fast Fourier transform for experimentalists, part VI: chirp of a bat. *Computing in Science & Engineering*, 8(2):72–78, 2006.
- [15] D. Donnelly and B. Rust. The fast Fourier transform for experimentalists, part I: concepts. *Computing in Science and Engineering*, 7(2):80–88, 2005.
- [16] D. Donnelly and B. Rust. The fast Fourier transform for experimentalists, part II: convolutions. *Computing in Science and Engineering*, 7(4):92–95, 2005.
- [17] I. J. Good. The interaction algorithm and practical Fourier analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):361–372, 1958.
- [18] Eugene Hecht, editor. *Optics*. Addison-Wesley, Reading, Mass., 4th ed edition, 2002.
- [19] Michael T. Heideman, Don H. Johnson, and C. Sidney Burrus. Gauss and the history of the fast Fourier transform. *Archive for History of Exact Sciences*, 34(3):265–277, 1985.
- [20] Hans Petter Langtangen. *A primer on scientific programming with Python*, volume 6 of *Texts in computational science and engineering*. Springer, Berlin and New York, 3rd ed edition, ©2012.

-
- [21] Newman, M. E. J. *Computational physics*. CreateSpace Independent Publishing Platform, University of Michigan, revised and expanded edition, 2013.
- [22] Brad Osgood. EE 261, The Fourier transform and its applications. Stanford University, Fall, 2007.
- [23] Athanasios Papoulis, editor. *The Fourier integral and its applications*. McGraw-Hill, New York, 1962.
- [24] B. Rust and D. Donnelly. The fast Fourier transform for experimentalists, part III: classical spectral analysis. *Computing in Science and Engineering*, 7(5):74–78, 2005.
- [25] B. Rust and D. Donnelly. The fast Fourier transform for experimentalists, part IV: autoregressive spectral analysis. *Computing in Science and Engineering*, 7(6):85–90, 2005.

Apéndice A

Función impulso: una distribución

La función impulso $\delta(t)$ es una herramienta muy importante en matemática aplicada. Su uso simplifica en gran medida la derivación de resultados que de otra forma implicarían argumentos complicados. En realidad, la definición usual (A.1) carece de significado; sin embargo, el concepto es aplicado correctamente al resolver muchos problemas. La razón de esto es debido a que ciertas propiedades de $\delta(t)$ son significativas e independientes de la definición de $\delta(t)$. La definición formal de $\delta(t)$, nos hace reconocer el hecho que no es una función ordinaria, que deben especificarse sus propiedades por medio de la teoría de distribuciones.

A continuación, se desarrollará una pequeña descripción de la teoría de distribución, tomando como referencia lo expuesto por Papoulis [23] en el apéndice I y Brigham [5] en el apéndice A.

A.1. Definición

Usualmente la función impulso es definida como

$$\delta(t - t_0) = 0, \quad \text{para } t \neq t_0 \tag{A.1}$$

$$\int_{-\infty}^{\infty} \delta(t - t_0) dt = 1 \tag{A.2}$$

lo que nos dice que la función impulso, no tiene magnitud definida en el tiempo de ocurrencia y cero en cualquier otro, además de que el área debajo de la curva debe ser la unidad.

Siguiendo la definición anterior, la función impulso se puede crear al tener una función cuya área debajo de la curva sea la unidad y progresivamente se aumente su magnitud y disminuya su duración manteniéndose su área igual a la unidad, esto nos dará una función muy grande en magnitud y muy pequeña en duración, dando así una definición alternativa

$$\delta(t) = \lim_{a \rightarrow 0} f(t, a) \tag{A.3}$$

La Fig. A.1 muestra varias funciones que pueden representar la función impulso aplicando la definición de la Ec. (A.3) y que satisfacen las Ec. (A.1) y (A.2)

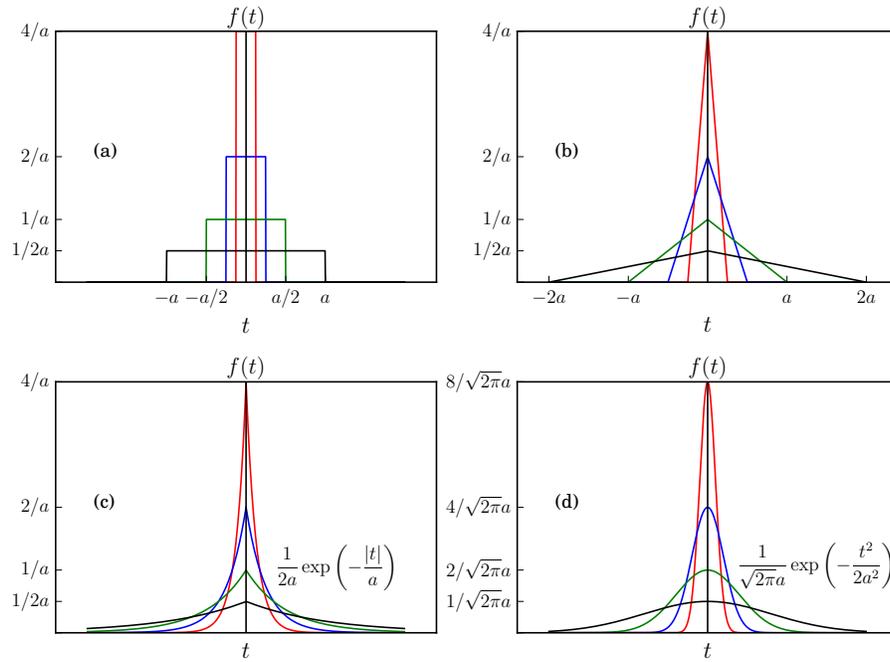


Figura A.1: Representación de la función impulso en el límite cuando $a \rightarrow 0$

A.2. El concepto de distribución

Una distribución, o función generalizada $g(t)$, es el proceso de asignar a una función arbitraria $\phi(t)$ una respuesta o un número

$$N[\phi(t)] \tag{A.4}$$

La función $\phi(t)$ es llamada *función de prueba* y la respuesta o número podrá ser el valor de $\phi(t)$ o su derivada (de cualquier orden) para algún $t = t_0$, el área debajo de $\phi(t)$ para algún intervalo, o cualquier otra cantidad dependiente de $\phi(t)$. El número asignado a la función de prueba $\phi(t)$ por la distribución $g(t)$ esta dado por la integral

$$\int_{-\infty}^{\infty} g(t)\phi(t)dt = N[\phi(t)] \tag{A.5}$$

Para funciones ordinarias, una integral tiene significado independiente y la Ec. (A.5) no cumple con esto, ya que nos indica que la integral y la función $g(t)$ están *definidas* por el número $N[\phi(t)]$ asignado a la función $\phi(t)$, y por lo tanto no tienen significado independiente.

Generalmente todas las funciones que tienen derivadas de cualquier orden y tienden a cero mas rápidamente que cualquier potencia de t , cuando t tiende a infinito, son del tipo de funciones de prueba utilizadas en la Ec. A.5 (ver capítulo 4 de [22]). Para ciertas distribuciones, como por ejemplo la función impulso, es suficiente asumir que $\phi(t)$ es continua en el origen.

Ahora vamos a aplicar esta definición, Ec. (A.5), a la función impulso y analizar sus propiedad.

A.3. Propiedades de la función impulso

La función impulso, $\delta(t)$, es una distribución que asigna a la función de prueba $\phi(t)$ el valor $\phi(0)$, tal que:

$$\int_{-\infty}^{\infty} \delta(t)\phi(t)dt = \phi(0) \tag{A.6}$$

Nuevamente, recordando que la relación de la Ec. (A.6) no tiene sentido como integral, la integral y la función $\delta(t)$ quedan definidas por el valor $\phi(0)$ asignado a la función $\phi(t)$.

Ahora se mostrarán las importantes propiedades de la función $\delta(t)$.

Propiedad de desplazamiento. La función $\delta(t - t_0)$ esta definida por

$$\int_{-\infty}^{\infty} \delta(t - t_0) \phi(t) dt = \phi(t_0) \quad (\text{A.7})$$

Propiedad de escala. La función $\delta(at)$ esta definida por

$$\int_{-\infty}^{\infty} \delta(at) \phi(t) dt = \frac{1}{|a|} \int_{-\infty}^{\infty} \delta(t) \phi(t/a) dt \quad (\text{A.8})$$

de lo cual se puede concluir que

$$\delta(at) = \frac{1}{|a|} \delta(t) \quad (\text{A.9})$$

Producto de $\delta(t)$ por una función ordinaria. El producto de la función $\delta(t)$ por una función ordinaria $h(t)$ esta definida por

$$\int_{-\infty}^{\infty} [\delta(t)h(t)] \phi(t) dt = \int_{-\infty}^{\infty} \delta(t) [h(t)\phi(t)] dt \quad (\text{A.10})$$

Si $h(t)$ es continua en $t = t_0$, entonces

$$h(t)\delta(t - t_0) = h(t_0)\delta(t - t_0) \quad (\text{A.11})$$

En general el producto de dos distribuciones no esta definido.

Propiedad de convolución. La convolución de dos impulsos se define como

$$\int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \delta_1(\tau) \delta_2(t - \tau) d\tau \right] \phi(t) dt = \int_{-\infty}^{\infty} \delta_1(\tau) \left[\int_{-\infty}^{\infty} \delta_2(t - \tau) \phi(t) dt \right] d\tau \quad (\text{A.12})$$

Entonces,

$$\begin{aligned} \delta(t - t_1) * \delta(t - t_2) &= \int_{-\infty}^{\infty} \delta(\tau - t_1) \delta(t - \tau - t_2) d\tau \\ &= \delta[t - (t_1 + t_2)] \end{aligned} \quad (\text{A.13})$$

Función impulso como un límite generalizado. Considere la secuencia de distribuciones $g_n(t)$. Si existe una distribución $g(t)$ tal que para cada función de prueba $\phi(t)$, se tiene

$$\lim_{n \rightarrow \infty} \int_{-\infty}^{\infty} g_n(t) \phi(t) dt = \int_{-\infty}^{\infty} g(t) \phi(t) dt \quad (\text{A.14})$$

entonces se dice que $g(t)$ es el límite de $g_n(t)$.

$$g(t) = \lim_{n \rightarrow \infty} g_n(t) \quad (\text{A.15})$$

También se puede definir una distribución como un límite generalizado de una secuencia $f_n(t)$ de funciones ordinarias. Suponga que $f_n(t)$ es tal que el límite

$$\lim_{n \rightarrow \infty} \int_{-\infty}^{\infty} f_n(t) \phi(t) dt \quad (\text{A.16})$$

existe para toda función de prueba. Este límite es entonces un valor que depende de $\phi(t)$ y define, por lo tanto, a una distribución $g(t)$, siendo esta

$$g(t) = \lim_{n \rightarrow \infty} f_n(t) \quad (\text{A.17})$$

e interpretándose en el sentido de la Ec. (A.14). Si la Ec. (A.17) existe como un límite ordinario, entonces define a la función equivalente, asumiendo la validez del intercambio del orden del límite y la integral en la Ec. (A.14).

La función impulso puede entonces definirse como un límite generalizado de una secuencia de funciones ordinarias que satisfacen

$$\lim_{n \rightarrow \infty} \int_{-\infty}^{\infty} f_n(t) \phi(t) dt = \phi(0) \quad (\text{A.18})$$

Si la Ec. (A.18) se cumple, entonces

$$\delta(t) = \lim_{n \rightarrow \infty} f_n(t) \quad (\text{A.19})$$

Cada una de las funciones ilustradas en la Fig. A.1 satisfacen la Ec. (A.18) y definen a la función impulso en el sentido de la Ec. (A.19).

Otra importante forma de definir la función impulso es

$$\delta(t) = \lim_{\omega \rightarrow \infty} \frac{\text{sen } \omega t}{\pi t} \quad (\text{A.20})$$

Usando la Ec. (A.20), se puede mostrar, ver [23], que

$$\int_{-\infty}^{\infty} \cos(2\pi ft) df = \int_{-\infty}^{\infty} e^{j2\pi ft} df = \delta(t) \quad (\text{A.21})$$

que es de considerable importancia al evaluar particulares transformadas de Fourier.

A.4. Prueba de la formula de inversión

Con lo expuesto anteriormente se puede derivar una sencill prueba formal para la formula de inversión

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df \quad (\text{A.22})$$

Si se sustituye $H(f)$ por

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \quad (\text{A.23})$$

en la Ec. (A.22); resulta que

$$\int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df = \int_{-\infty}^{\infty} e^{j2\pi ft} df \int_{-\infty}^{\infty} h(x) e^{-j2\pi fx} dx \quad (\text{A.24})$$

Usando la Ec. (A.21) e intercambiando los integrandos en la Ec. (A.24), se obtiene

$$\begin{aligned} \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df &= \int_{-\infty}^{\infty} h(x) dx \int_{-\infty}^{\infty} e^{j2\pi f(t-x)} df \\ &= \int_{-\infty}^{\infty} h(x) \delta(t-x) dx \end{aligned} \quad (\text{A.25})$$

De la definición de la función impulso en la Ec. (A.7), la Ec. (A.25) es igual a $h(t)$. Esta ecuación es válida sólo si $h(t)$ es continua.¹ Sin embargo, si se asume que $h(t)$ está definida como el valor medio en el punto de discontinuidad, es decir

$$h(t) = \frac{h(t^+) + h(t^-)}{2} \quad (\text{A.26})$$

entonces la formula de la inversión aún es valida.

¹Esto debido a que la definición de la función impulso se basa en la continuidad de la función de prueba.

Apéndice B

Evaluación gráfica de la integral de convolución

Vamos a comenzar demostrando el teorema de convolución en el tiempo. Si $h(t)$ y $x(t)$ tienen transformadas de Fourier $H(f)$ y $X(f)$, entonces

$$\begin{aligned} h(t) * x(t) &= \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau & (B.1) \\ \int_{-\infty}^{\infty} [h(t) * x(t)] e^{-j2\pi ft} dt &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau \right] e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} x(\tau) \left[\int_{-\infty}^{\infty} h(t - \tau)e^{-j2\pi ft} dt \right] d\tau \\ &= \int_{-\infty}^{\infty} x(\tau) \left[\int_{-\infty}^{\infty} h(u)e^{-j2\pi f(u+\tau)} du \right] d\tau, \quad u = t - \tau \\ &= \int_{-\infty}^{\infty} x(\tau) \left[e^{-j2\pi f\tau} \int_{-\infty}^{\infty} h(u)e^{-j2\pi fu} du \right] d\tau \\ &= \int_{-\infty}^{\infty} x(\tau)e^{-j2\pi f\tau} H(f) d\tau \\ &= H(f)X(f) \end{aligned}$$

$$h(t) * x(t) \Leftrightarrow H(f)X(f) \quad (B.2)$$

La prueba de la inversión se desarrolla de forma similar y de igual forma el teorema de convolución en la frecuencia

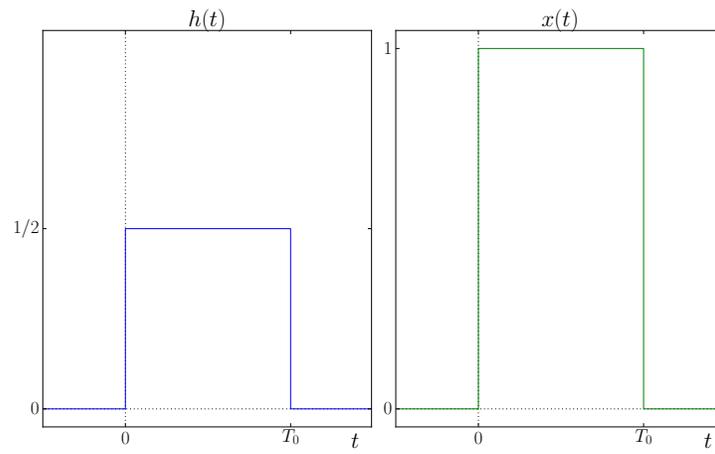
$$h(t)x(t) \Leftrightarrow H(f) * X(f) \quad (B.3)$$

B.1. Evaluación gráfica

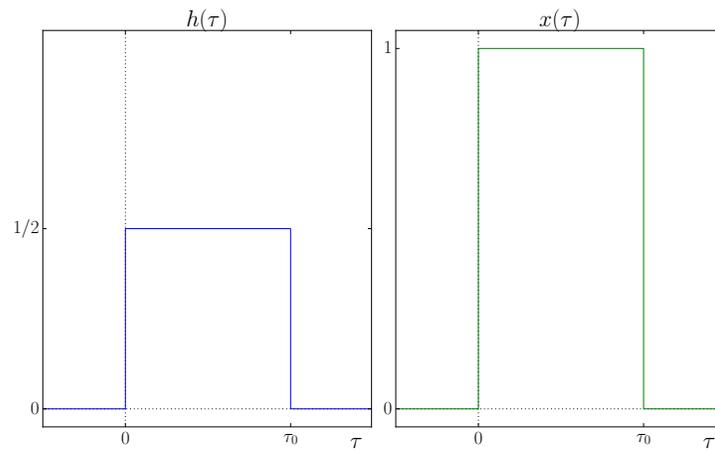
Ahora analicemos la evaluación gráfica de la integral de convolución. Sea $h(t)$ y $x(t)$ dos funciones del tiempo representadas gráficamente en la Fig. B.1(a). Para evaluar la Ec. (B.1) se requieren las funciones $x(\tau)$ y $h(t - \tau)$; $x(\tau)$ y $h(\tau)$ se obtienen sustituyendo la variable t por τ en $x(t)$ y $h(t)$, véase la Fig. B.1(b). Ahora $h(-\tau)$ es la imagen de $h(\tau)$ alrededor del eje de las ordenadas y $h(t - \tau)$ es la función $h(-\tau)$ desplazada por un valor t , mostrando estas últimas dos en la Fig. B.1(c). Para calcular la integral de la Ec. (B.1), es necesario multiplicar e integrar las funciones $x(\tau)$ [Fig. B.1(b)] y $h(t - \tau)$ [Fig. B.1(c)] por cada valor de t desde $-\infty$ hasta ∞ , esto es ilustrado en la Fig. B.2(c). Se observa que cuando $t \leq 0$, este producto es cero. Para valores entre $0 < t < 4t_1$, el producto y la integral de $x(\tau)$ con $h(t - \tau)$ están representados por el área sombreada debajo de la curva, y estos valores de área son los puntos de la función triángulo que es la representación gráfica de la convolución de $x(t)$ y $h(t)$, considerando que t puede tomar valores continuos. Para terminar, cuando $t \geq 4t_1$ este producto nuevamente es cero.

El procedimiento descrito es una técnica gráfica conveniente para evaluar la integral de convolución. Resumiendo, los pasos serían:

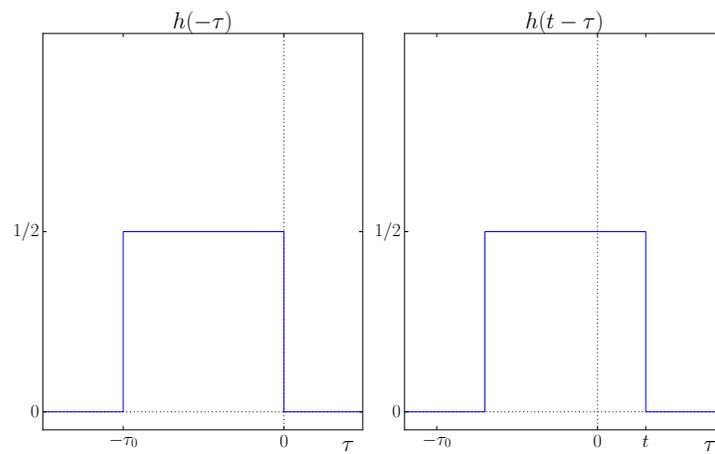
1. *Giro.* Tome la imagen de $h(\tau)$ alrededor del eje de las ordenadas.
2. *Desplazamiento.* Desplace $h(-\tau)$ por un valor t .
3. *Multiplicación.* Multiplique la función desplazada $h(t - \tau)$ por $x(\tau)$.
4. *Integración.* El área debajo de la curva resultante del producto $h(t - \tau)$ y $x(\tau)$ es el valor de la convolución en el tiempo t .



(a)



(b)



(c)

Figura B.1: Ilustración de los pasos para la convolución

B.2. Convolución que contiene funciones impulso

La integral de convolución más simple de evaluar es una que incluye funciones impulso. Para ilustrar este punto, tomemos $h(t)$ como la función impulso y $x(t)$ la función rectangular mostradas en la Figs. B.3(a) y B.3(b) respectivamente. Para estas funciones la Ec. (B.1) nos da

$$y(t) = \int_{-\infty}^{\infty} [\delta(\tau - T) + \delta(\tau + T)] x(t - \tau) d\tau \quad (\text{B.4})$$

y recordando la Ec. (A.7)

$$\int_{-\infty}^{\infty} \delta(\tau - T) x(\tau) d\tau = x(T)$$

entonces la Ec. (B.4) se puede escribir como

$$y(t) = x(t - T) + x(t + T) \quad (\text{B.5})$$

La función $y(t)$ es ilustrada en la Fig. B.3(c). Note que la convolución de la función $x(t)$ con una función impulso es evaluada por el simple desplazamiento de $x(t)$ a la posición de la función impulso.

Se recomienda dar un vistazo a la página web de la Universidad Johns Hopkins de señales, sistemas y control

<http://pages.jh.edu/~signals/>

Al mismo tiempo, se desarrollo en formato HTML, una pequeña simulación de la convolución vista gráficamente, esto tomando como base el código *html_movie.py* de la página <http://physicalmodelingwithpython.blogspot.com/>. En las Figs. B.4, B.5 y B.6 se muestran ejemplos de dicha simulación.

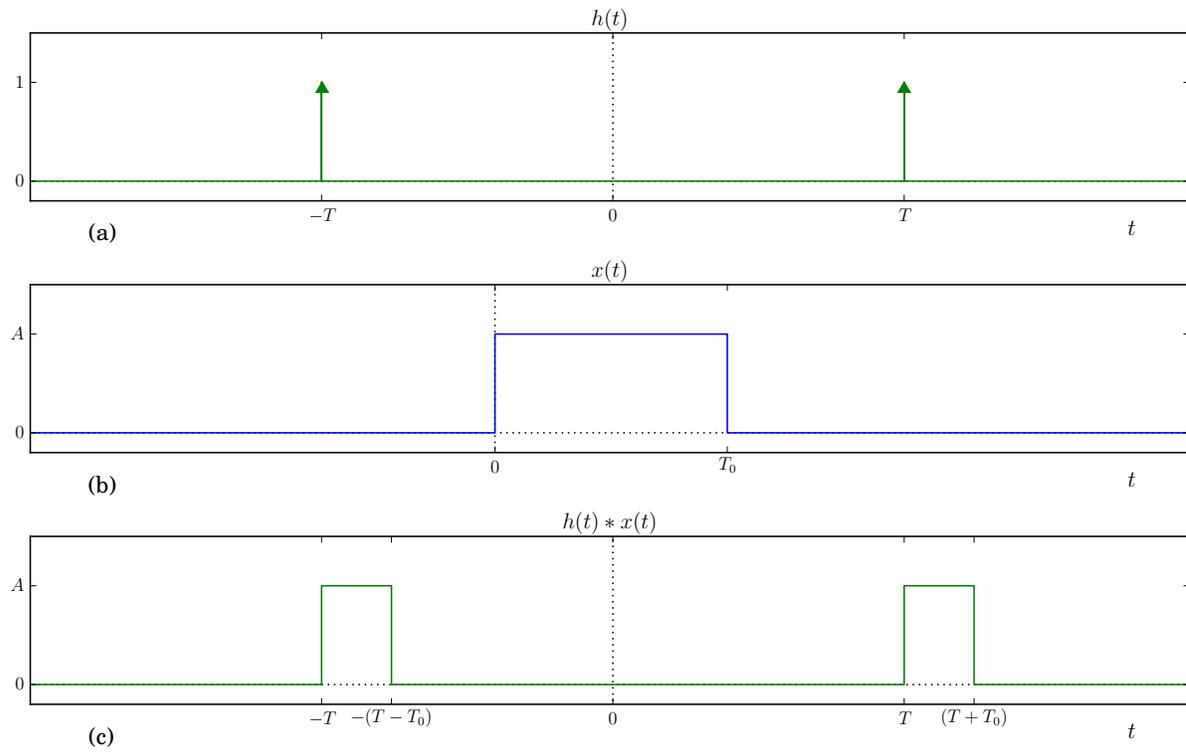


Figura B.3: Ilustración de la convolución con funciones impulso

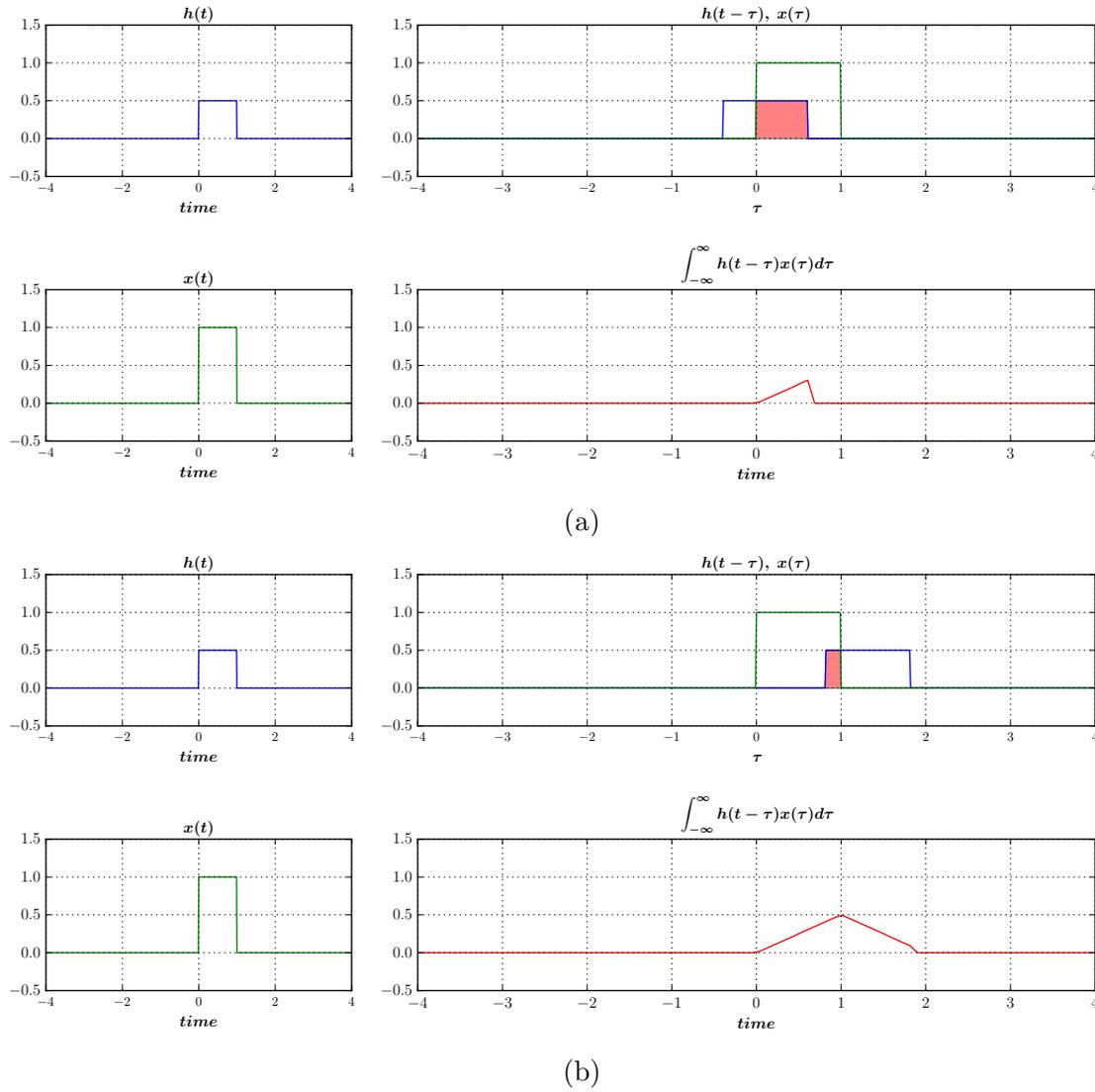


Figura B.4: Simulación de convolución, dos pulsos cuadrados

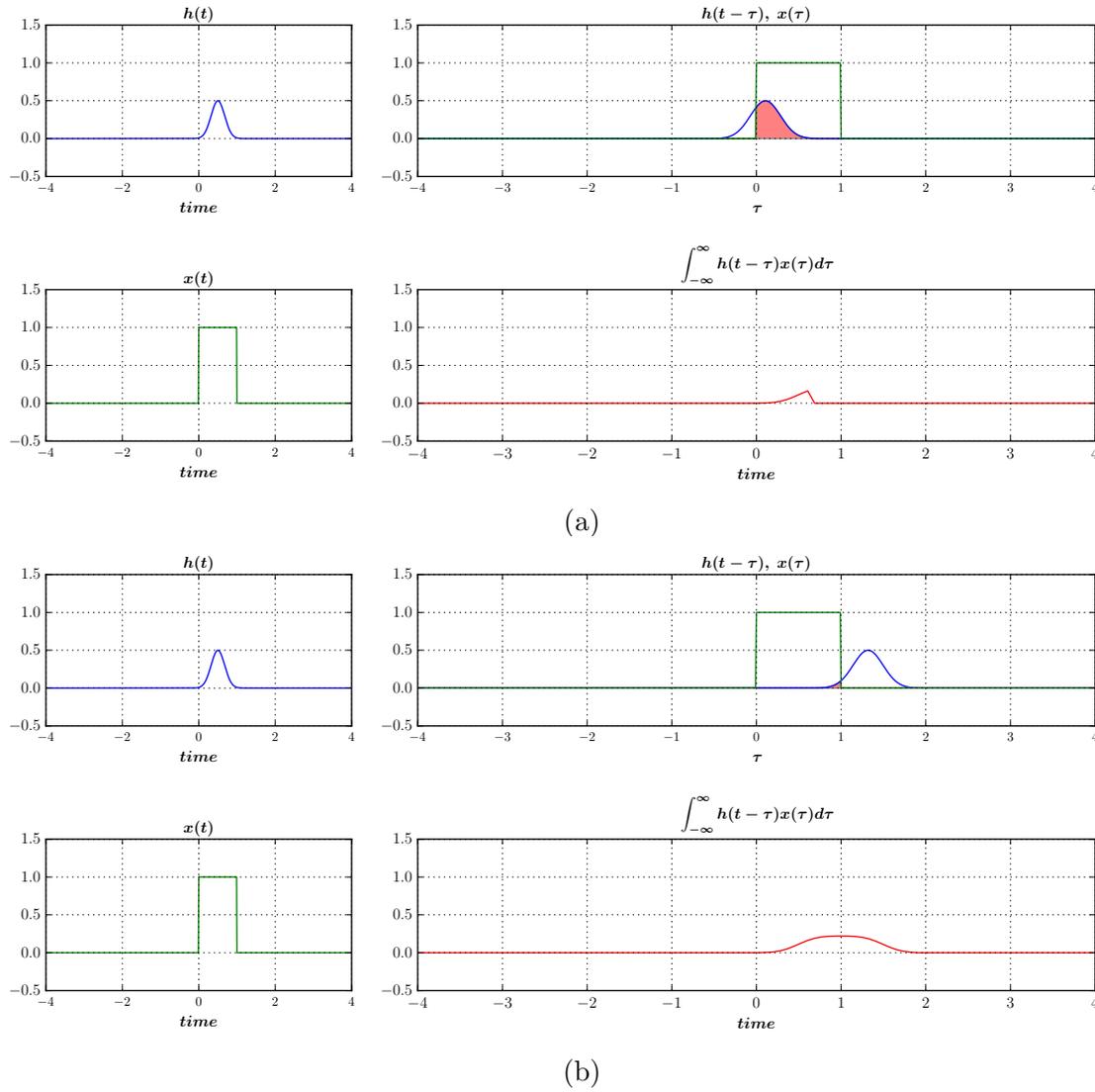


Figura B.5: Simulación de convolución, pulso gaussiano y pulso cuadrado

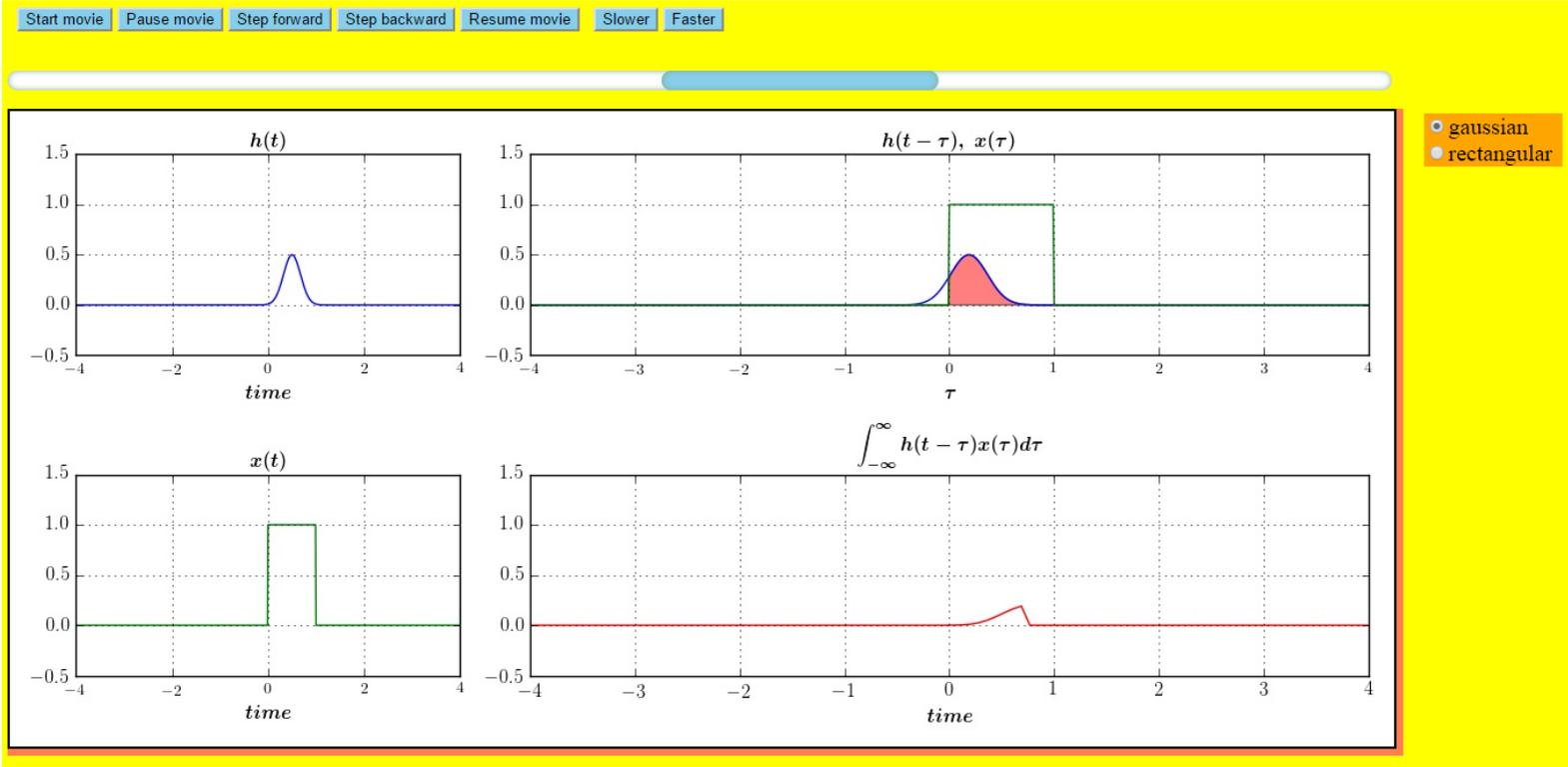


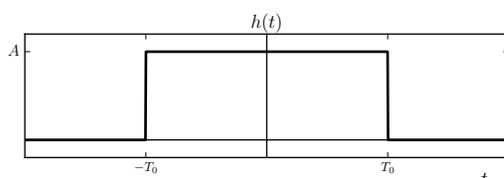
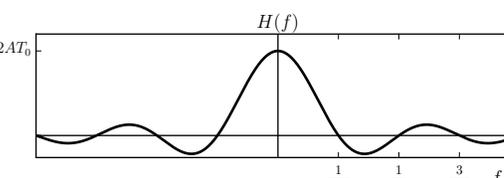
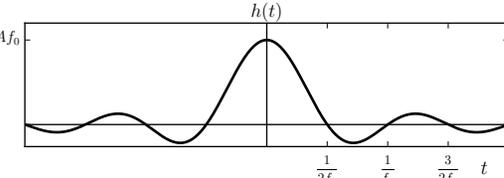
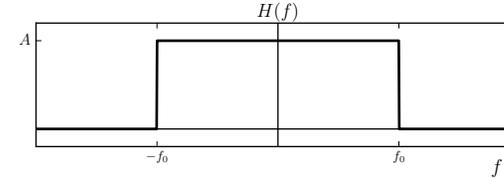
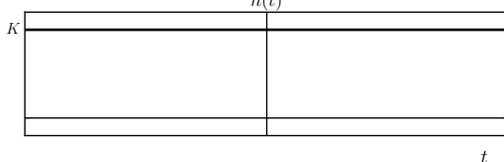
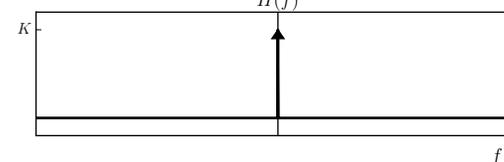
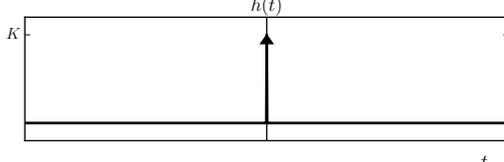
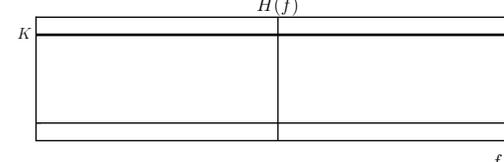
Figura B.6: Vista HTML de la simulación de convolución

Apéndice C

Tabla de pares de transformadas

A continuación se presentan los pares de transformadas de las funciones mas importantes que se analizan en el presente documento.

Tabla C.1: Pares de transformadas de Fourier

Dominio del tiempo	Par transformado	Dominio de la frecuencia
	$h(t) = \begin{cases} A & t < T_0 \\ \frac{A}{2} & t = T_0 \\ 0 & t > T_0 \end{cases} \Leftrightarrow H(f) = 2AT_0 \frac{\text{sen}(2\pi T_0 f)}{2\pi T_0 f}$	
	$h(t) = 2Af_0 \frac{\text{sen}(2\pi f_0 t)}{2\pi f_0 t} \Leftrightarrow H(f) = \begin{cases} A & f < f_0 \\ \frac{A}{2} & f = f_0 \\ 0 & f > f_0 \end{cases}$	
	$h(t) = K \Leftrightarrow H(f) = K\delta(f)$	
	$h(t) = K\delta(t) \Leftrightarrow H(f) = K$	

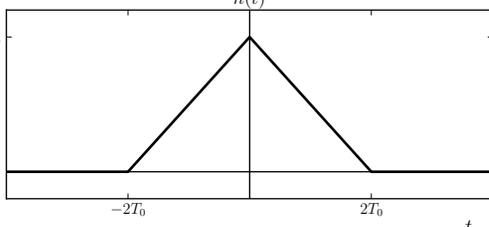
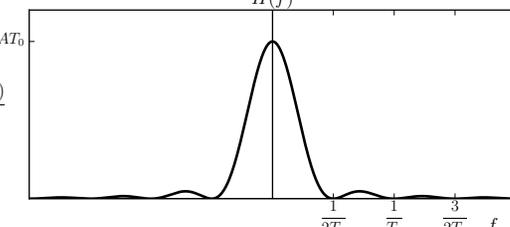
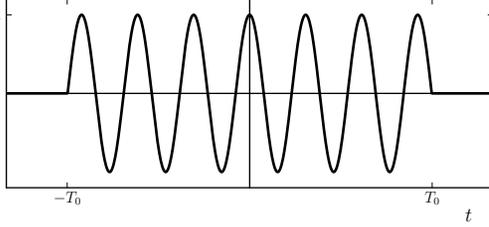
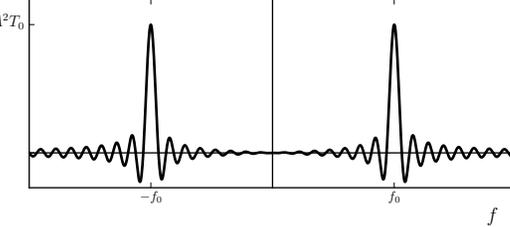
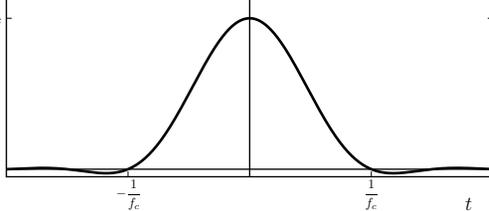
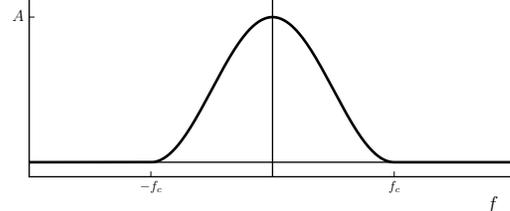
continua ...

Pares de transformadas de Fourier

Dominio del tiempo	Par transformado	Dominio de la frecuencia
	$h(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \Leftrightarrow H(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right)$	
	$h(t) = A \cos(2\pi f_0 t) \Leftrightarrow H(f) = \frac{A}{2} \delta(f - f_0) + \frac{A}{2} \delta(f + f_0)$	
	$h(t) = A \text{sen}(2\pi f_0 t) \Leftrightarrow H(f) = -j \frac{A}{2} \delta(f - f_0) + j \frac{A}{2} \delta(f + f_0)$	

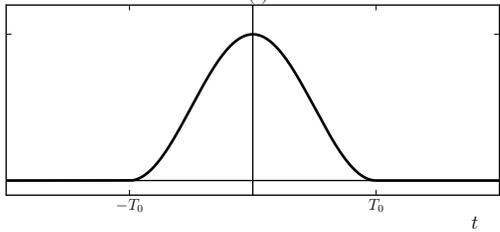
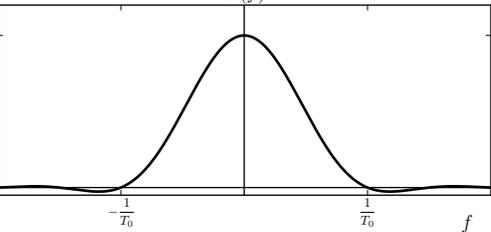
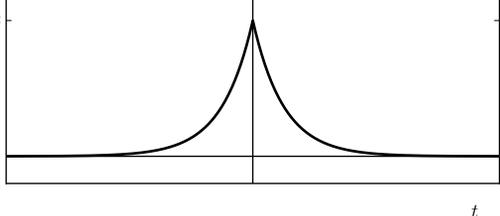
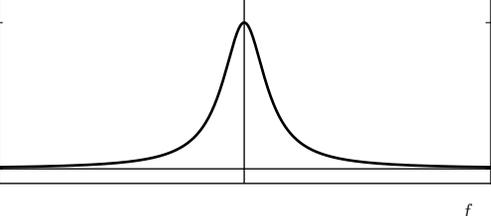
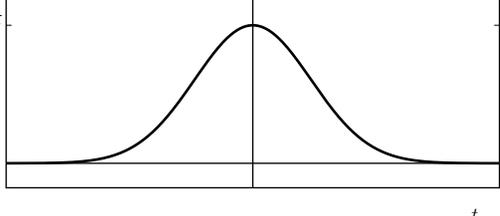
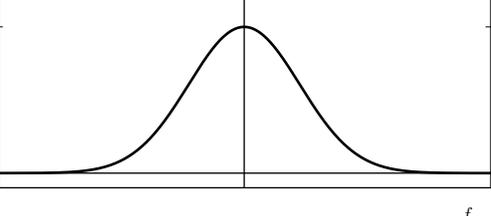
continua ...

Pares de transformadas de Fourier

Dominio del tiempo	Par transformado	Dominio de la frecuencia
	$h(t) = \begin{cases} -\frac{A}{2T_0}t + A & t < 2T_0 \\ 0 & t > 2T_0 \end{cases} \Leftrightarrow H(f) = 2AT_0 \frac{\text{sen}^2(2\pi T_0 f)}{(2\pi T_0 f)^2}$	
	$h(t) = \begin{cases} A \cos(2\pi f_0 t) & t < T_0 \\ 0 & t > T_0 \end{cases} \Leftrightarrow H(f) = A^2 T_0 [Q(f + f_0) + Q(f - f_0)]$ $Q(f) = \frac{\text{sen}(2\pi T_0 f)}{(2\pi T_0 f)}$	
	$h(t) = \frac{Af_c}{2} q(t) \Leftrightarrow H(f) = \begin{cases} \frac{A}{2} + \frac{A}{2} \cos\left(\frac{\pi f}{f_c}\right) & f \leq f_c \\ 0 & f > f_c \end{cases}$ $q(t) = \frac{\text{sen}(2\pi f_c t)}{\pi f_c t}$	

continua ...

Pares de transformadas de Fourier

Dominio del tiempo	Par transformado	Dominio de la frecuencia
	$h(t) = \begin{cases} \frac{1}{2} + \frac{1}{2} \cos\left(\frac{\pi t}{T_0}\right) & t \leq T_0 \\ 0 & t > T_0 \end{cases} \Leftrightarrow H(f) = T_0 Q(f) + \frac{T_0}{2} Q\left(f + \frac{1}{2T_0}\right) + \frac{T_0}{2} Q\left(f - \frac{1}{2T_0}\right)$ $Q(f) = \frac{\text{sen}(2\pi T_0 f)}{2\pi T_0 f}$	
	$h(t) = \frac{\alpha}{2} \exp(-\alpha t) \Leftrightarrow H(f) = \frac{\alpha^2}{\alpha^2 + 4\pi^2 f^2}$	
	$h(t) = \sqrt{\frac{\alpha}{\pi}} \exp(-\alpha t^2) \Leftrightarrow H(f) = \exp\left(\frac{-\pi^2 f^2}{\alpha}\right)$	

Apéndice D

Tabla de propiedades

Tabla D.1: Propiedades de la transformada de Fourier de funciones complejas

Dominio del tiempo, $h(t)$	Dominio de la frecuencia, $H(f)$
Real	Parte real, par; parte imaginaria, impar
Imaginaria	Parte real, impar; parte imaginaria, par
Parte real, par; parte imaginaria, impar	Real
Parte real, impar; parte imaginaria, par	Imaginaria
Real y par	Real y par
Real e impar	Imaginaria e impar
Imaginaria y par	Imaginaria y par
Imaginaria e impar	Real e impar
Compleja y par	Compleja y par
Compleja e impar	Compleja e impar

Tabla D.2: Propiedades de la transformada de Fourier

Dominio del tiempo	Dominio de la frecuencia
Linealidad $x(t) + y(t)$	Linealidad $X(f) + Y(f)$
Simetría $H(t)$	Simetría $h(-f)$
Escalamiento en el tiempo $h(kt)$	Cambio de escala inverso $\frac{1}{ k }H\left(\frac{f}{k}\right)$
Cambio de escala inverso $\frac{1}{ k }h\left(\frac{t}{k}\right)$	Escalamiento en la frecuencia $H(kf)$
Desplazamiento en el tiempo $h(t - t_0)$	Desplazamiento de fase $H(f)e^{-j2\pi ft_0}$
Modulación $h(t)e^{j2\pi t f_0}$	Desplazamiento en la frecuencia $H(f - f_0)$
Función par $h_e(t)$	Función real $H_e(f) = R_e(f)$
Función impar $h_o(t)$	Función imaginaria $H_o(f) = jI_o(f)$
Función real $h(t) = h_r(t)$	Parte real, par; parte imaginaria, impar $H(f) = R_e(f) + jI_o(f)$
Función imaginaria $h(t) = jh_i(t)$	Parte real, impar; parte imaginaria, par $H(f) = R_o(f) + jI_e(f)$

Tabla D.3: Propiedades de la transformada de Fourier continua y discreta

Transformada de Fourier	Propiedad	Transformada discreta de Fourier
$x(t) + y(t) \Leftrightarrow X(f) + Y(f)$	Linealidad	$x(k) + y(k) \Leftrightarrow X(n) + Y(n)$
$H(t) \Leftrightarrow h(-f)$	Simetría	$\frac{1}{N}H(k) \Leftrightarrow h(-n)$
$h(t - t_0) \Leftrightarrow H(f)e^{-j2\pi ft_0}$	Desplazamiento en el tiempo	$h(k - i) \Leftrightarrow H(n)e^{-j2\pi ni/N}$
$h(t)e^{j2\pi ft_0} \Leftrightarrow H(f - f_0)$	Desplazamiento en la frecuencia	$h(k)e^{j2\pi ki/N} \Leftrightarrow H(n - i)$
$\left[\int_{-\infty}^{\infty} H^*(f)e^{-j2\pi ft} df \right]^*$	Formula de inversión alterna	$\left[\frac{1}{N} \sum_{n=0}^{N-1} H^*(n)e^{-j2\pi kn/N} \right]^*$
$h_e(t) \Leftrightarrow R_e(f)$	Funciones par	$h_e(k) \Leftrightarrow R_e(n)$
$h_o(t) \Leftrightarrow jI_o(f)$	Funciones impar	$h_o(k) \Leftrightarrow jI_o(n)$
$h(t) = h_e(t) + h_o(t)$ $= \left[\frac{h(t)}{2} + \frac{h(-t)}{2} \right]$ $+ \left[\frac{h(t)}{2} - \frac{h(-t)}{2} \right]$	Descomposición	$h(k) = h_e(k) + h_o(k)$ $= \left[\frac{h(k)}{2} + \frac{h(N - k)}{2} \right]$ $+ \left[\frac{h(k)}{2} - \frac{h(N - k)}{2} \right]$
$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau = x(t) * h(t)$	Convolución	$y(k) = \sum_{i=0}^{N-1} x(i)h(k - i) = x(k) * h(k)$
$y(t) * h(t) \Leftrightarrow Y(f)H(f)$	Teorema de convolución en el tiempo	$y(k) * h(k) \Leftrightarrow Y(n)H(n)$
$y(t)h(t) \Leftrightarrow Y(f) * H(f)$	Teorema de convolución en la frecuencia	$y(k)h(k) \Leftrightarrow \frac{1}{N}Y(n) * H(n)$
$\int_{-\infty}^{\infty} h^2(t)dt = \int_{-\infty}^{\infty} H(f) ^2 df$	Teorema de Parseval	$\sum_{k=0}^{N-1} h^2(k) = \frac{1}{N} \sum_{n=0}^{N-1} H(n) ^2$

Apéndice E

Diagrama de flujo del programa FFT

A continuación se presenta el diagrama de flujo, usando las propiedades discutidas del flujo de la FFT. Primero se calcula el arreglo $l = 1$ empezando en el nodo $k = 0$ hasta el último de los nodos en el arreglo. En cada nodo k , se hacen los cálculos con el par de ecuaciones de la Ec. (2.5.21), donde p es determinado por el proceso descrito en la sección siguiente. Se continúa de esta forma en el arreglo utilizando el par de ecuaciones anteriores hasta que se alcanza una región de nodos que deben ser omitidos. Nos saltamos los nodos apropiados y continuamos hasta que hayamos calculado todo el arreglo. A continuación, se procede a calcular los arreglos restantes utilizando los mismos procedimientos. Finalmente, se reorganiza el arreglo final para obtener los resultados deseados. La Fig. E.1 ilustra el diagrama de flujo para la programación del algoritmo FFT.

E.1. Inversión de bit

Consideremos el siguiente procedimiento para implementar la operación de inversión de bit. Sabemos que un número binario, es decir $b_3b_2b_1b_0$, puede ser escrito en base 10 como $b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$. El número con bits invertidos que se trata de encontrar está dado por $b_0 \times 2^3 + b_1 \times 2^2 + b_2 \times 2^1 + b_3 \times 2^0$. Si se describe una técnica para determinar los bits (del número binario) b_3, b_2, b_1 y b_0 , entonces se habrá definido la operación de inversión de bit.

Ahora, asuma que M es un número decimal igual a $b_3b_2b_1b_0$. Divida M por 2, tome la parte entera, y este resultado truncado multiplíquelo por 2. Entonces calcule $(b_3b_2b_1b_0) - 2(b_3b_2b_1)$. Si el bit b_0 es 0, entonces esta diferencia es cero porque la división por 2, truncamiento, y subsecuente multiplicación por 2 no modifica M . Sin embargo, si el bit b_0 es 1, el truncamiento cambia el valor de M y la diferencia anterior no es cero. Mediante esta técnica se puede determinar si el bit b_0 es 0 o 1.

Se puede determinar de forma similar el bit b_1 . La expresión para la diferencia sería $(b_3b_2b_1) - 2(b_3b_2)$. Si la diferencia es cero, entonces b_1 es cero. Los bits b_2 y b_3 son determinados de forma similar. Este procedimiento es la base para el desarrollo de la subrutina de la inversión de bit en la Fig. E.1, la cual encontrará a continuación.

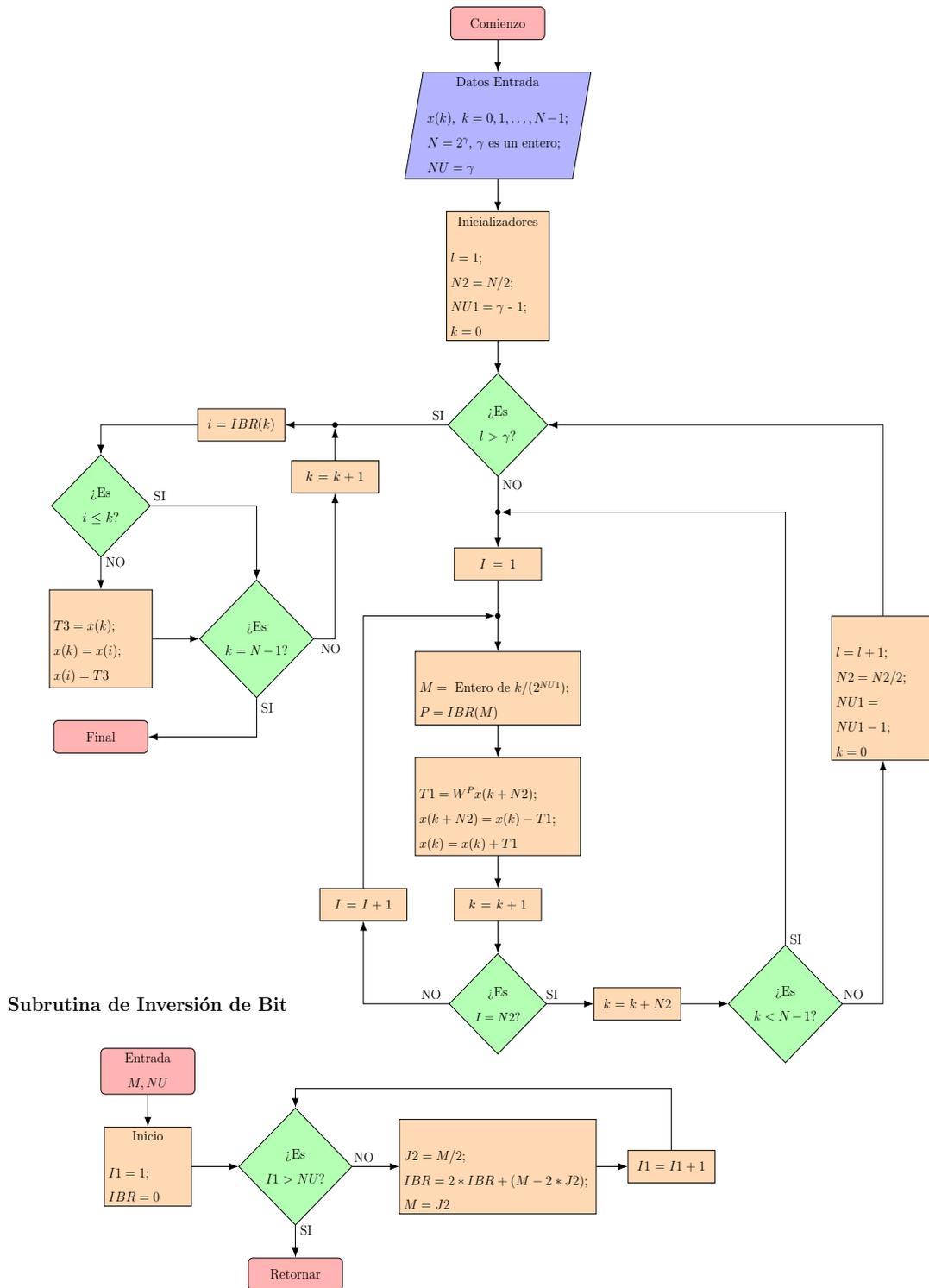


Figura E.1: Diagrama de flujo del programa FFT, tomado de Brigham [5]