

*Aplicación en Scilab para el tratamiento de señales digitales
con la Transformación de Fourier fraccionaria: Filtro de
Wiener fraccionario*

MARCOS AMARIS GONZALEZ

UNIVERSIDAD DEL MAGDALENA
Programa de Ingeniería de Sistemas
Santa Marta D.T.C.H.

Febrero de 2009



Aplicación en Scilab para el tratamiento de señales digitales con la Transformación de Fourier fraccionaria: Filtro de Wiener fraccionario utilizando Scilab

Presentado por:
MARCOS AMARIS GONZALEZ

Proyecto de Investigación realizado con el fin de obtener el
título de Ingeniero de Sistemas

Dirigido por:
Dr. RAFAEL ANGEL TORRES

Universidad del Magdalena
Facultad de Ingeniería
Programa de Ingeniería de Sistemas
Santa Marta D.T.C.H.
Febrero de 2009

Índice general

Índice general	II
Índice de figuras	IV
Índice de cuadros	VI
1. Introducción	1
2. Planteamiento del Problema	3
3. Antecedentes	12
3.1. $\mathfrak{F}_\alpha [f](x)$	12
3.2. Descripción del algoritmo $fracF$	19
3.2.1. Función $fracF$	19
3.2.2. función $corefrmod2(fc,a)$	21
4. Marco Teórico Conceptual	24
5. Justificación	31
5.1. Viabilidad Técnica	32
5.2. Viabilidad Económica	33
5.3. Viabilidad Social	34
6. Objetivos	35
6.1. Objetivo General	35
6.2. Objetivos Específicos	35
7. Formulación & Hipótesis	37
7.1. Transformación de Fourier Discreta	38
7.1.1. Transformación Rápida de Fourier	39
7.2. Transformación de Fourier fraccionaria Discreta	39
7.3. Convolución Fraccionaria	41

8. Metodología	43
8.1. Metodología XP	43
8.1.1. Análisis	44
8.1.2. Diseño	45
8.1.3. Desarrollo	46
8.1.4. Pruebas	46
9. Desarrollo del Trabajo	47
9.1. Función KernelFrFT	49
9.2. Convolución fraccionaria	52
9.3. Filtro de Wiener fraccionario	53
10. Demostración de hipótesis y Resultados	60
11. Limitaciones & inconvenientes	67
12. Conclusiones	73
Bibliografía	74

Índice de figuras

2.1. Fenómeno de aliasing durante un proceso de muestreo y reconstrucción	5
2.2. a) Onda Senoidal b) Wavelet	8
3.1. Ventana Rectangular	14
3.2. Distribución de Wigner asociada a una señal	16
3.3. Operación de Convolución	18
3.4. Operaciones matriciales en Matlab	20
4.1. Jean Baptiste Fourier, (1768-1830)	25
4.2. a) Señal Digital, b) Señal Digitalizada c) Reloj de muestreo d) Señal Binaria Cuantizada	26
4.3. Señal no estacionaria	26
7.1. El tamaño de la Señal es el mismo que el Ancho de Banda de la Transformada	40
7.2. Δx en el teorema de Muestreo fraccionario	41
8.1. Metodología XP	45
9.1. Fases de la $FrFT$ con el algoritmo de la FFT en el kernel a) $a = 0,1$; b) $a = 0,3$; c) $a = 0,5$; d) $a = 0,9$; e) $a = 1,3$; f) $a = 1,9$	51
9.2. $FrFT$ con $a = -0,5$ de una función rectángulo de 1024 muestras . .	55

9.3. Distribución de Wigner asociada a la Transformada de la figura 9.2	56
9.4. Valor absoluto de la Transformada de la figura 9.2 + ruido aditivo de una función Chirp*(Gauss+aleatorio)	56
9.5. Valor real de Transformada de la figura 9.2 + ruido aditivo de una función Chirp*(Gauss+aleatorio)	57
9.6. Distribución de Wigner de la señal en la figura 9.4	57
9.7. Ruido + señal en su dominio directo al implementar una $FrFT$ con $a = 0,5$	58
9.8. Distribución de Wigner de la señal en la figura 9.7	58
10.1. Menú Principal de la aplicación	61
10.2. Función Rectángulo escalada y su Transformada de Fourier	62
10.3. a) Función Rectángulo; b) $FrFT$ con $a=0.1$; c) $FrFT$ con $a=0.3$; d) $FrFT$ con $a=0.5$; e) $FrFT$ con $a=0.7$; f) $FrFT$ con $a=0.9$;	62
10.4. Distribución de Wigner de la señal rectángulo de 1024 muestras	63
10.5. Rotación de Dis. Wigner por medio de la $FrFT$ de una función rectángulo de 1024 muestras	64
10.6. Imagen Tru.jpg de Scilab Image Processing	64
10.7. $FrFT$ con $a = 0,5$ en las filas y columnas	65
10.8. $FrFT$ de Tru.jpg con $a = 1 FT$	65
10.9. $FrFT$ de Tru.jpg con $a = 1,75 FT$	66
10.10 $FrFT$ de Tru.jpg con $a = 0,1$ en filas y $a = 0,9$ en las columnas	66
11.1. Valor de la fase de dos funciones de las $FrFT$ implementadas	69
11.2. Valor de las fases de las funciones de convolución fraccionaria implementadas	71

Índice de cuadros

3.1. Casos Particulares de la $FrFT$, según el valor del operador a	17
7.1. Comparación del número de operaciones entre la DFT y el algoritmo de la FFT de señales Bidimensionales cuadradas	39
11.1. Duración en segundos de transformaciones realizadas a señales unidimensionales y bidimensionales de diferentes tamaños	68

Resumen

“El teorema de Fourier no es solamente uno de los más hermosos resultados del Análisis Moderno, sino que se ha convertido en instrumento indispensable de la física moderna.”¹.

En 1980 surgió una nueva operación para el tratamiento de señales inventada por V. Namías [21], desde ese momento surgieron operaciones como la convolución y correlación fraccionaria, el teorema de muestreo en dominios fraccionarios y nuevos filtros en el dominio fraccionario de fourier [22, 4]; también se encontraron sus propiedades y relaciones con otras funciones, de la cual surgió una de las más importantes, la cual es que la $FrFT$ es una rotación de la distribución de Wigner que se encuentra asociada a una función, esto ha servido como una excelente representación de señales en tiempo-frecuencia y han surgido muchas implementaciones para el tratamiento de señales de muchas categorías entre ellas las no estacionarias [5, 6, 9].

En el capítulo 2 se propone el tratamiento de señales con la $FrFT$ implementada en las operaciones necesarias, con el fin de realizar el filtro de Wiener fraccionario, en el capítulo 3 los antecedentes de esta investigación, en el 4 la teoría de la $FrFT$, en el 5 la justificación, en el 6 los objetivos, en el capítulo 7 se formulan algunas ecuaciones para implementarlas computacionalmente, en 8 la Metodología XP como metodología de ciclo de vida de desarrollo de software, en el 9 y en el 10 el desarrollo de las funciones y la demostración por medio de simulación. En el Capítulo 11 se exponen algunos inconvenientes y limitaciones durante la investigación y al final 12 las conclusiones.

¹Lord Kelvin

Capítulo 1

Introducción

Desde la invención de la Transformación de Fourier, esta siempre ha ocupado los primeros puestos en el tratamiento de señales, no obstante, esta ha tenido que acondicionarse para realizar un estudio de señales que requieren un análisis más especializado, anteriormente no se podía obtener una información real y concreta de señales que variaran su amplitud y frecuencia aperiódicamente, ya que como principal condición de la transformación de Fourier y de otras Transformaciones adaptadas de ella, es que las señales a analizar deben ser periódicas. Las señales no estacionarias son una clase de señales que no cumplen con esta condición.

hace aproximadamente 2 décadas surgió la invención de una nueva Transformación [21], la cual tiene como caso particular la Transformación de Fourier, la Transformación de Fourier inversa y la misma función de entrada, su nombre es Transformación de Fourier fraccionaria (Fractional Fourier Transform, $FrFT$ por sus siglas en inglés). La $FrFT$ puede ser vista como una excelente representación de señales en tiempo-frecuencia [22] y podemos extraer información de la frecuencia y el tiempo de las señales analizadas, pudiendo obtener frecuencias presentes de la señal en tiempos específicos. La $FrFT$ puede verse como una rotación de la Distribución de Wigner asociada a una señal donde la rotación está

dada por el ángulo $\alpha = a\pi/2$) siendo a el a – *esimo* grado de fraccionalización de la Transformación de Fourier fraccionaria.

Además, La $FrFT$ trajo consigo un nuevo enfoque fraccionario para el tratamiento de señales, entre ellos están el Teorema de Muestreo fraccionario o Teorema de Muestreo en el dominio fraccionario de Fourier [38], la convolución fraccionaria, la correlación fraccionaria y seguirá avanzando, trayendo nuevos modelos para el tratamiento de señales.

Capítulo 2

Planteamiento del Problema

La búsqueda de métodos para extraer y recuperar información que poseen señales de cualquier tipo y categoría, con el objeto de conocer atributos relacionados a los procesos físicos que las generan, siempre ha sido tema de interés para ingenieros y matemáticos. La primera división natural de todas las señales son las estacionarias y las no estacionarias [3]. Las señales estacionarias son constantes en sus parámetros estadísticos (media aritmética, variación estándar, entre otras) y las señales no estacionarias son aquellas que sus parámetros de amplitud y/o frecuencia pueden variar a lo largo y ancho de su espacio de estudio [4], ejemplos de estas señales, son los exámenes médicos como electrocardiogramas, electroencefalogramas, señales de presión, análisis de sismología, entre otras.

Una de las técnicas más utilizadas ha sido representar las señales en el dominio del tiempo; esto ha permitido asociar cambios físicos y biofísicos con el de los parámetros amplitud-tiempo de la señal [6]. Posteriormente, el dominio de la frecuencia vino a complementar esta información. El análisis de Fourier informa acerca de la presencia o ausencia de determinadas frecuencias en la señal analizada [7]. La definición de la Transformación de Fourier está dada por la sigu-

iente ecuación.

$$F(y) = \int_{-\infty}^{\infty} f(x)e^{-2ixy} dx \quad (2.1)$$

Para todo x sea este número real, donde x es la variable del dominio directo de la función de entrada e y es la variable a ser representada en la frecuencia.

La Transformación de Fourier (Fourier Transform) es una herramienta de gran utilidad para el Tratamiento de señales estacionarias, sobre todo desde el descubrimiento de la Transformación rápida de Fourier [4], pero para el tratamiento de señales no estacionarias la Transformación de Fourier estándar presenta más inconvenientes que ventajas ya que esta práctica común adolece de una limitada resolución en frecuencia haciéndola algunas veces inútil para el análisis de estas señales [8], ya que esta Transformación supone que el espectro y la frecuencia de las señales son uniformes durante todo el periodo muestreado, y en la práctica podemos observar que muchas señales tanto la frecuencia como la amplitud varían en el tiempo, por tal razón al aplicar la FT en estas señales, ni la representación temporal ni la representación frecuencial arrojarán información sustancial del proceso.

La conversión de una señal análoga a digital se realiza a través de un proceso de muestreo, el teorema de Claude Shannon (padre de la Teoría de la Información), este formula el proceso de muestreo de una señal continua como la frecuencia de muestreo mínima para garantizar condiciones óptimas de fidelidad de la señal tanto en su representación digital como en su reconstrucción análoga, esta frecuencia es comúnmente llamada frecuencia de Nyquist [9].

El Teorema del Muestreo de Shannon, también se conoce como teorema de muestreo de Whittaker-Nyquist-Kotelnikov-Shannon, utilizado básicamente para el tratamiento

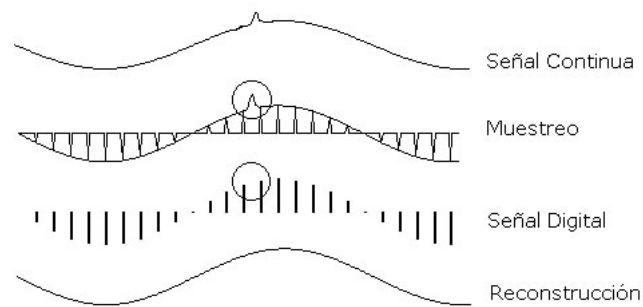


Figura 2.1: Fenómeno de aliasing durante un proceso de muestreo y reconstrucción

de señales, el cual establece una frecuencia mínima de muestreo necesaria para evitar la superposición de los espectros “Aliasing” y hacer una perfecta reconstrucción análoga de una señal $f(x)$.

Según el anterior teorema si al muestrear una señal con función $f(x)$, a una frecuencia menor que $1/\zeta$, siendo ζ el ancho de banda de la Transformada asociada a la función, entonces al intentar reconstruir la señal original podemos obtener frecuencias que esta no contenía, es decir, podemos confundir una frecuencia f_1 con otra f_2 , por ello a f_2 se le llama el alias de f , a este fenómeno se le conoce en el tratamiento de señales como aliasing [10].

En la Figura 2.1, podemos observar el proceso de muestreo de una señal continua con puntos equidistantes entre sí, para su posterior reconstrucción. Pero el espectro resultante es ficticio, ya que contiene frecuencias diferentes, lo anterior debido a que la señal continua al ser reconstruida nuevamente no posee la frecuencia que se encuentra encerrada en el círculo, esto se debe a que la digitalización en el dominio de tiempo es muy grande y posee errores de amplitud o espacios de frecuencia los que no puede llegar por su periodicidad.

Bajo estas necesidades surge la representación tiempo-frecuencia [10], ya que

bajo estas escalas se puede hacer un buen estudio de la frecuencia y la amplitud de la señales a lo largo del tiempo de las mismas, existen varias técnicas para la representación de señales en tiempo-frecuencia, estas técnicas varían en eficiencia y optimización de la información extraída, cabe destacar que estas técnicas son derivadas de la Transformación de Fourier estandar [11].

La representación en tiempo-frecuencia provee de un puente entre estas representaciones simultáneamente, así en cualquier instante de tiempo podemos tener información del espectro frecuencial de la señal [12]. Según esto si se representan las señales en tiempo-frecuencia, entonces podemos obtener información de cómo evoluciona o cambia el espectro de una señal a través del tiempo, de esta forma se puede extraer información, analizar e interpretar una infinidad de procesos físicos y/o biofísicos que se dan en forma de señales no estacionarias; también podremos realizar un tratamiento de señales unidimensionales como electrocardiogramas y representarlos en tiempo-frecuencia, obteniendo información de real interés en cualquier intervalo de tiempo. La *FrFT* ha sido implementada como una función para el análisis de señales en representación tiempo-frecuencia, siendo esta una de las más óptimas para el tratamiento de señales no estacionarias.

Como se dijo anteriormente, existen varias técnicas para el tratamiento de señales representadas en tiempo-frecuencia, la mayoría de estas extensiones de la Transformación de Fourier estándar, las cuales han sido rediseñadas para un tratamiento de señales más especializados, entre las cuales están:

- Transformación de Tiempo Corto de Fourier (STFT por sus siglas en inglés): Este es el método más clásico para esta clase de representaciones, la *STFT* fue propuesta por Gabor en 1946. La idea básica de este algoritmo es la introducción de una especie de ventaneo a lo largo y ancho de la señal [13],

los cuales son trasladados en puntos equidistantes durante estos espacios, esta transformación se calcula con la siguiente ecuación:

$$G(\tau, y) = \int_{-\infty}^{\infty} f(x)g(x - \tau)e^{-2ixy}dx \quad (2.2)$$

La ecuación 2.2 se puede interpretar como los resultado de la transformación de Fourier estándar para cada punto $G(\tau, y)$, donde y es el espectro frecuencial de la señal $f(x)$ para cada punto τ , siendo $f(x)$ la función de entrada y la variable a representar en su dominio directo, $g(x - \tau)$ la función de ventaneo utilizada en la ecuación, ver figura 3.1, la *STFT* no puede mejorar el tiempo y la frecuencia debido a que la función de ventana es estática, además carece de varias propiedades de la transformación de Fourier estándar [14].

- Transformación Wavelet (*WT* por su siglas en inglés): es el Tratamiento de Fourier por medio de pequeñas ondas en un espacio determinado de tiempo llamadas wavelet, ver figura 2.2, la *WT* es otro rediseño de la *FT* la cual introduce una ventana dinámica la cual está dada por la ecuación 2.3:

$$w(t) = \frac{1}{\sqrt{|a|}}w\left(\frac{t - b}{a}\right) \quad (2.3)$$

donde a es la escala y es b la traslación, esta ecuación presenta un análisis tiempo-frecuencia que entrega información redundante [15].

Recientemente el área de la óptica de Fourier se ha extendido con nuevas contribuciones relativas a transformaciones no convencionales denominadas Transformaciones fraccionarias. Una de sus principales herramientas es la Transformación de Fourier fraccionaria (*FrFT* por sus siglas en inglés) [16]. Este es un nuevo método de representación de señales en tiempo frecuencia, con el cual se

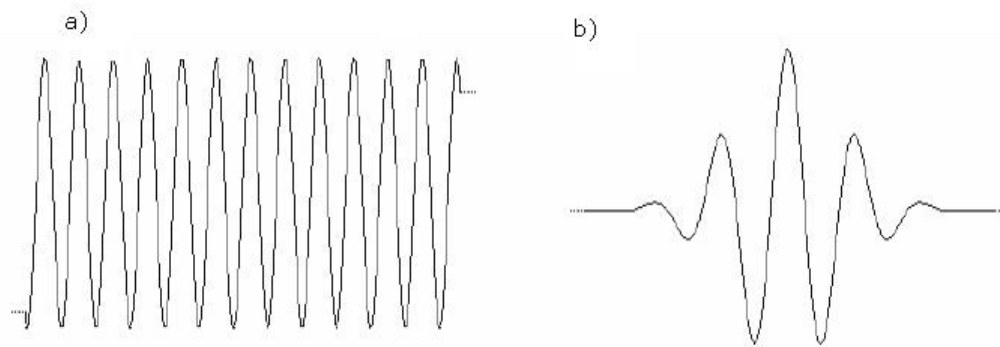


Figura 2.2: a) Onda Senoidal b) Wavelet

han llenado muchos vacíos en el procesamiento de señales no estacionarias que se venían dando con los métodos convencionales antes expuestos, con esta técnica podemos obtener un tratamiento de señales no estacionarias que provee de información verídica y de interés para interpretaciones y toma de decisiones acerca de estas, entre sus utilidades está el cambio y recuperación de la fase en un intervalo de tiempo dado [17].

Se ha propuesto, por ejemplo, la *FrFT* para filtrado espacialmente variante, reconocimiento de caracteres, encriptado, marca de agua, implementación de redes neuronales, solución de ecuaciones diferenciales, entre otras [18], las notaciones de esta función (según se interprete mejor para el caso) se pueden escribir de las siguientes maneras:

$$f_{\alpha} = \mathfrak{F}_{\alpha}[f(x)] = \mathfrak{F}_{\alpha}[f]. \quad (2.4)$$

Donde a la función de entrada $f(x)$ es realizada una Transformación de Fourier fraccional \mathfrak{F}_{α} con un ángulo $\alpha = a\pi/2$, donde a es el orden de fraccionalización de la operación.

La *FrFT* tiene aproximadamente 6 definiciones todas ellas muy parecidas, ya

que sus autores la han adaptado a sus conveniencias para diferentes áreas y clases de investigaciones, sin embargo la que mejor se adapta al procesamiento de señales representadas en tiempo-frecuencia es la propuesta por Victor Namias [21], este autor propone que la *FrFT* de una función $f(x)$ está dada por la siguiente expresión:

$$f_a(y) = C_\alpha e^{i\pi y^2 \cot \alpha} \int f(x) e^{i\pi x^2 \cot \alpha} e^{-i\frac{2\pi}{\sin \alpha} xy} dx. \quad (2.5)$$

a es un número real (en otras definiciones puede tener un valor complejo), que determina el orden fraccionario de la transformación y C_α está dada por la siguiente función:

$$C_\alpha = \frac{e^{i(s(\alpha)\frac{\pi}{4} - \frac{\alpha}{2})}}{\sqrt{|\sin \alpha|}} \quad (2.6)$$

Donde $s(\alpha)$ representa una función sign.

La *FrFT* se ha establecido hoy en día como una de las herramientas más poderosas y de mayor uso para el tratamiento de señales representadas en tiempo-frecuencia, ya que tiene la facilidad de extraer información veraz del espectro [22] en cualquier instante de tiempo [23]. Además de esto ha traído consigo un óptimo tratamiento de diferentes clases de señales (unidimensionales y multidimensionales, continuas y discretas, y periódicas y no periódicas) [24].

La *FrFT* ha sido implementada en filtros de restauración y mejora de señales, reconocimiento de patrones como huellas dactilares y una perfecta optimización de filtros adaptativos como el filtro de Wiener [25], esto último se basa en la posibilidad de la fraccionalización de la transformación en un número a , el cual es utilizado en el filtro de Wiener fraccionario para minimizar el error cuadrático medio (Mean Square Error *MSE* por su siglas en inglés) [26]; el *MMSE* al hacer mínimo el error, y tender a 0, menor es la probabilidad de equivocación al mo-

mento de hacer una comparación entre señales, se conoce como *MMSE* (Minimum Mean Square Error) el error cuadrático medio mínimo.

La restauración de señales se debe a que al ser captada una señal, se presenta cierto ruido el cual necesitamos manejar para una mejor calidad de estas, este hecho lo podemos definir matemáticamente de la siguiente manera:

$$s(x) = e(x) + r(x) \quad (2.7)$$

l donde $s(x)$ es la señal; $e(x)$ es la escena y $r(x)$ es el ruido presente [1, 2].

El filtro de Wiener se utilizan comúnmente en la restauración de señales representadas en tiempo-frecuencia [18]. Un filtro de Wiener se diseña de tal manera que su salida sea lo más parecidamente posible a la imagen de referencia para ello se considera que la imagen y el ruido son procesos aleatorios [19]. El parecido entre la imagen de referencia y la restaurada se puede comparar usando el error cuadrático medio [26].

El filtro de Wiener estándar presenta deficiencias en el dominio fraccionario de Fourier [20, 26], así como también el proceso de muestreo de Shannon que se venía implementando en el tratamiento de señales [11]. Bajo estas deficiencias y/o necesidades surgieron el filtro de Wiener fraccionario [38] y el teorema de muestreo en dominios fraccionarios de Fourier [39]. Juntando estas tres herramientas hace que podamos extraer información de las señales y realizar un óptimo tratamientos de señales, según lo anterior se formula el siguiente interrogante ¿Por qué se hace necesario la implementación computacional de la Transformación de Fourier fraccionaria en el tratamiento de señales y cuales serían los beneficios de esta?

El desarrollo e implementación de este proyecto será realizado sobre Scilab, esta plataforma se puede encontrar sobre Internet, así como también, muchas contribuciones o las llamadas Toolbox, entre sus ventajas frente a otras se puede mencionar que es una plataforma gratis, de código abierto, presenta un lenguaje de programación de alto nivel para cálculo científico, disponible en múltiples sistemas operativos (*Unix, GNU/Linux, Windows, Solaris, Alpha*); y que todos los procesos que se generan para realizar las diferentes operaciones son ejecutados paralelamente, es decir, que a través de esta herramienta se podría llegar a realizar un ambiente totalmente distribuido utilizando las librerías e interfaces de diversos lenguajes de programación tales como *java, C, Fortran* y diferentes sistemas operativos [30], hasta llegar al caso donde un grupo de computadores puedan compartir sus dispositivos de hardware y/o software y realizar todos sus procesos en paralelo.

Y ya que Scilab es una herramienta de software libre el usuario puede personalizarlo o ampliarlo añadiendo sus propias funciones, y como la transformación de Fourier fraccionaria no existe en esta plataforma se propone su desarrollo e implementación en las funciones necesarias para hallar el filtro de Wiener fraccionario para el procesamiento de señales. El desarrollo de la *FrFT* será de gran ayuda para el estudio y análisis de señales de diferentes señales de biofísica, electrofísica, geofísica, acústicas, de voz, fisiológicas, entre otras.

Capítulo 3

Antecedentes

Podemos definir señal como toda función $f(x)$ que contiene información que varía en el tiempo, en el espacio ó en ambos simultáneamente y la Transformada $F(y)$ es la relación de la amplitud que obtenemos de una variable que se esté analizando a lo largo del tiempo ó su dominio directo.

No existe alguna duda que la madre de todas las Transformaciones de señales es la Transformación de Fourier, la cual se denota como $\mathfrak{F}[f](x)$ donde x es el dominio directo de la señal a analizar y f la función de entrada a la Transformación, la ecuación de esta operación está dada en la ecuación 2.1.

3.1. $\mathfrak{F}_\alpha [f](\mathbf{x})$

A pesar de la gran utilidad de la Transformación de Fourier en el tratamiento de señales, esta presenta ciertas limitaciones en la obtención de la información de la señal de entrada, ya que al aplicarla no se puede saber si en las frecuencias que se obtuvieron en $F(y)$ aparece un valor determinado de la función $f(x)$, se puede

saber si la frecuencia existe más no el tiempo cuando se produce.

En la naturaleza hay muchas señales no estacionarias, en las cuales la amplitud y frecuencia varían en el tiempo. Así por ejemplo, en la voz, las frecuencias varían mucho si el tiempo es bastante grande y por tanto un análisis en el dominio de la frecuencia sin tener en cuenta el tiempo es muy ineficiente para este tipo de señales. Por lo tanto la transformación de Fourier estándar no es una herramienta idónea para el análisis de señales no estacionarias. Esta limitante hace que no se pueda obtener suficiente información de la función de entrada, por lo cual surgieron algunas adaptaciones para que esta función pueda ser representada en tiempo-frecuencia.

La Transformación de Fourier de tiempo corto *STFT* por sus siglas en inglés, es la primera en introducir el concepto de ventaneo en la función a analizar, con el fin de hacer el tratamiento en sectores separados periódicamente y así obtener el valor de la Transformación de Fourier en cada uno de estos sectores, así se dividiría el resultado y se podrá saber el espectro de frecuencia en cada uno de estos puntos llamados por otros investigadores átomos de la ventana, Véase figura 3.1, dependiendo de la función de ventana que se seleccione quedará la operación matemática, sin embargo continúan las mismas limitaciones que se mencionan anteriormente con la Transformación de Fourier estándar.

Es decir, si la función a tratar es $f(x)$ y la ventana a utilizarse depende de la función $g(x)$ con un factor de traslación τ , entonces cada punto del espectro de la frecuencia está dado por la ecuación 2.2 al ojo puede verse si se tienen ciertos conocimientos matemáticos que esta operación obtiene también la forma de una convolución por lo tanto se mantienen ciertas propiedades que se mencionan abajo con estas operaciones.

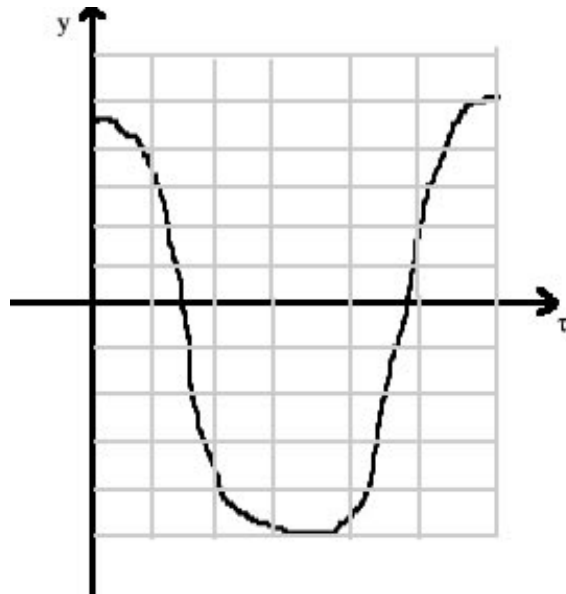


Figura 3.1: Ventana Rectangular

Las técnicas de adaptación de la Transformación de Fourier se volvieron un objetivo para investigadores, principalmente por la necesidad de encontrar una representación de señales de todas las clases, entre ellas las no estacionarias las cuales son aquellas que pueden ser generadas aleatoriamente como las sísmicas, geofísica, biofísica, entre otras.

Una de las más grandes utilidades en la búsqueda de representaciones de señales en tiempo frecuencia, es la descomposición de operaciones en familias de funciones, lo anterior sumado al vasto procesamiento de los computadores modernos ha hecho posible nuevos descubrimientos en el tratamiento de señales digitales.

Aproximadamente en el año de 1985 un Francés de apellido *Meyer* inventó la Transformación Wavelet u ondeletas [49], la cual a su vez lleva su nombre como Wavelet de Meyer, este es un tratamiento realizado con base en pequeñas ondas

que varían su amplitud periódicamente en el tiempo llamadas wavelets o ondeletas, de ahí proviene su nombre, véase figura 2.2.

Otra representación en tiempo frecuencia es la Distribución de Wigner, esta nos permite visualizar el soporte compacto de una señal $f(x)$, cuya Transformada de Fourier es $F(y)$, es decir, esta permite saber el espacio en el que suceden ciertos valores asociados a una frecuencia, esta operación se obtiene con la siguiente ecuación 3.1:

$$\mathcal{W}_f(x, y) = \int_{-\infty}^{\infty} f\left(x + \frac{x'}{2}\right) \overline{f\left(x - \frac{x'}{2}\right)} e^{-2\pi i x' y} dx' \quad (3.1)$$

donde x es la variable a evaluar de la función de entrada y x' es otra variable que está en el dominio de las x . Como puede observarse en la ecuación 3.1 el resultado de la distribución de wigner es una función bidimensional asociada a una función f , por tal razón decimos que es es una función bidimensional. La gráfica de una distribución de Wigner de una función $f(x)$ puede obtenerse como se muestra en la figura 3.2, sin embargo en la mayoría de los casos es visualizada en forma de imagen. En esta figura la sección de cuadrícula azul en los ejes x e y es el soporte de la distribución de Wigner, lo cuales tienen un valor de amplitud cada uno asociado a la función $f(x)$ en cada punto (x, y) de la ecuación 3.1, esta cuadrícula se llama comunmente soporte compacto de la señal.

Ahora si se integra el resultado de la distribución de Wigner, quedarían los siguientes integrales:

$$\int_{-\infty}^{\infty} \mathcal{W}_f(x, y) dy = |f(x)|^2 \quad (3.2)$$

siendo $|f(x)|^2$ la sumatoria de la variable del dominio directo de la función $f(x)$ al cuadrado.

$$\int_{-\infty}^{\infty} \mathcal{W}_f(x, y) dx = |F(y)|^2 \quad (3.3)$$

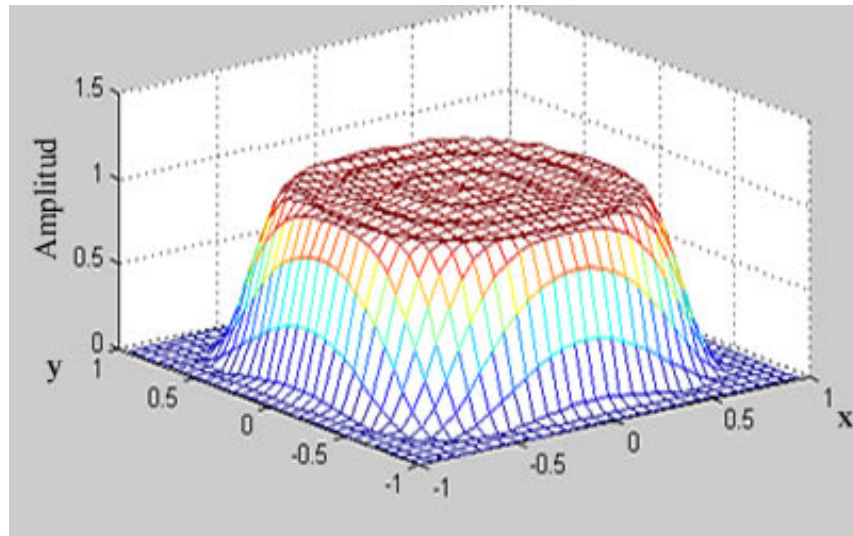


Figura 3.2: Distribución de Wigner asociada a una señal

$|F(y)|^2$ la sumatoria de la variable del dominio de la frecuencia de la Transformada $F(y)$ al cuadrado.

$$\int_{-\infty}^{\infty} \mathcal{W}_f((x \cos \alpha - y \sin \alpha, x \sin \alpha + y \cos \alpha) dy = |f_{\alpha}(x_{\alpha})|^2 \quad (3.4)$$

y $|f_{\alpha}(x_{\alpha})|^2$ es la sumatoria del valor absoluto en el dominio fraccionario de Fourier dado por el ángulo α , siendo esta la transformada de Fourier fraccionaria la cual es el centro de esta investigación, y está dada por la ecuación 2.5.

Esta transformación presenta 4 casos particulares los cuales dependen del grado de fraccionalización de la Transformación, ver tabla 3.1, Los algoritmos que desarrollen la operación de la *FrFT* deben cumplir con estas condiciones.

Existen otras transformaciones que se realizan en el dominio directo como lo es la convolución y la correlación, estas se aplican con el fin de encontrar la magnitud de la relación entre dos señales, la primera está dada por la siguiente integral

a	$\alpha = a \pi / 2$	Operador fraccionario	Operación sobre la señal
0 ó 4	0 ó 2π	$F^0 = F^4 = I$	Operador Identidad
1	$\pi/2$	$F^1 = F$	FT
2	π	$F^2 = FF = \mathfrak{P}$	Operador Paridad
3	$3\pi/2$	$F^3 = FF^2 = F^{-1}$	FT inversa

Cuadro 3.1: Casos Particulares de la $FrFT$, según el valor del operador a

$$[f * g](x) = \int_{-\infty}^{\infty} f(y)g(x - y)dy \quad (3.5)$$

donde $f(x)$ es la señal a convolucionar con la función $g(x)$, y $x = x - y$ con el fin de desplazar la función y así al ser esta móvida se encontrará la relación de la amplitud en todos los puntos de las dos señales, ver figura 3.3. En resumen la figura 3.3 muestra la convolución entre la función $f(x)$ y $g(x)$, donde $g(x)$ se debe reflejar en su dominio directo e irla tarsladando en un valor y , con el fin de hallar la relación de la superposición de las dos funciones.

Teorema de la Convolución 1 *Al aplicar la Transformación de Fourier a una convolución, obtenemos el producto de las dos Transformadas de las funciones relacionadas en la convolución.*

El teorema anterior se puede representar matemáticamente por la siguiente ecuación.

$$\mathfrak{F}[f * g](y) = F(y)G(y) \quad (3.6)$$

Si aplicamos en ambos lados de la ecuación una Transformación de Fourier Inversa, hayaremos la convolución de las dos funciones originales, este proceso es el que se conoce como convólución rápida y se describe matemáticamente con la siguiente ecuación 3.7:

$$\mathfrak{F}^{-1}[\mathfrak{F}[f * g](y)] = \mathfrak{F}^{-1}[F(y)G(y)] \quad (3.7)$$

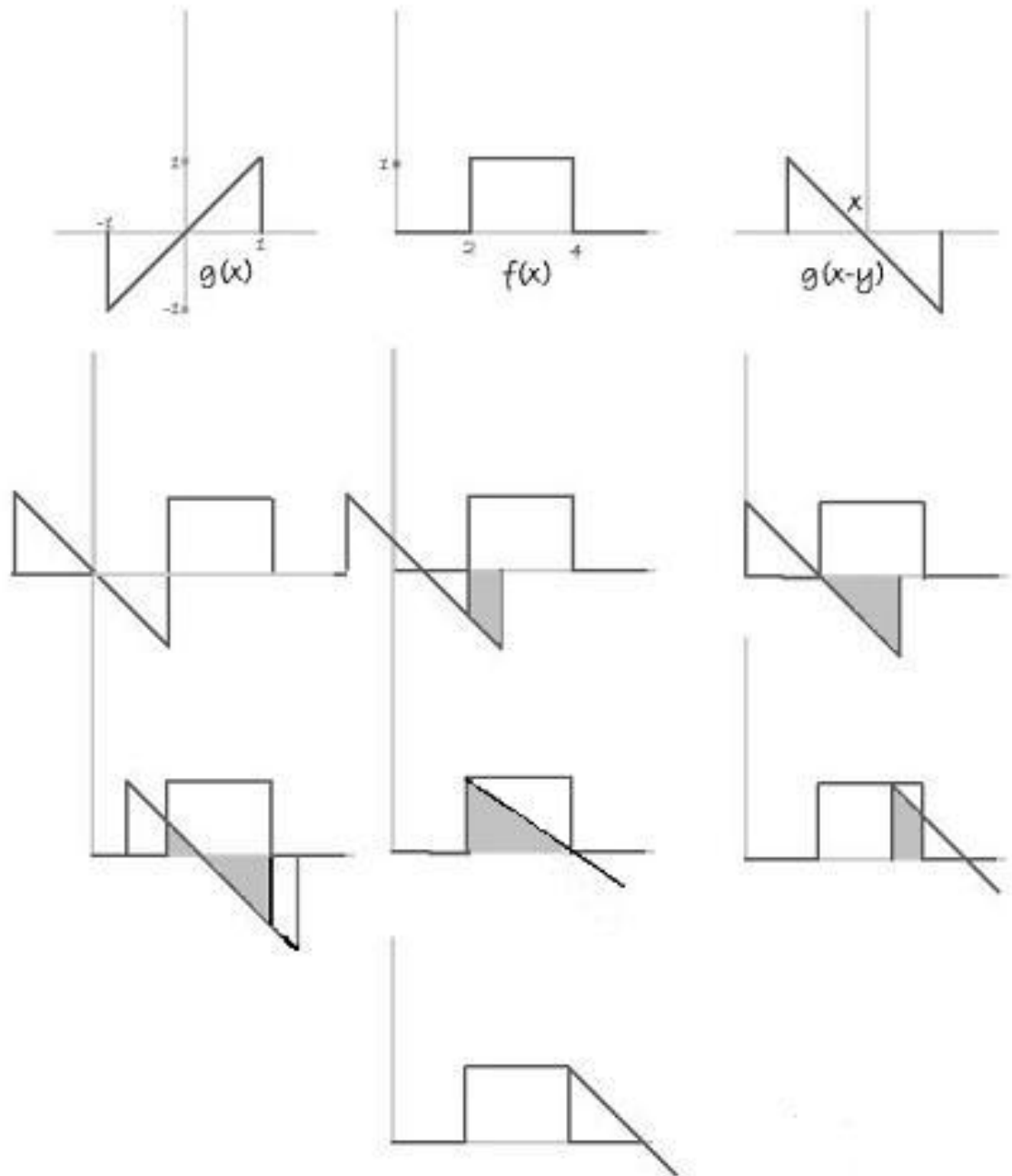


Figura 3.3: Operación de Convolución

3.2. Descripción del algoritmo $fracF$

En esta investigación se analizó el código de la $FrFT$ descrito por el paper [46], este se conoce comúnmente como $fracF$ explicado en la sección 3.2.1 página 19 de este documento, este código se puede hallar en el sitio Web [45].

3.2.1. Función $fracF$

`function[res]=fracF(fc,a)`, los parámetros de entrada de la función son fc y a , donde fc es la señal o muestras de la señal y a el a^{esimo} valor a fraccionar la Transformación, ver figura ??, la salida res , será la Transformada de Fourier fraccionaria de fc .

```
N = length(fc);
```

Se halla el tamaño de la señal, el cual debe ser impar como condición principal de este algoritmo.

```
if fix(N/2) ~= N/2
```

```
error('Length of the input vector should be even');
```

```
end;
```

En las plataformas de calculo numérico es importante evitar los bucles *for* para el uso de operaciones sobre elementos de un vector, en este caso la programación es ineficiente debido a que los bucles *for* son interpretados y esto hace lento el proceso, este bucle se debe utilizar como última opción, no con motivos de calculo, sino en operaciones de control solamente, véase figura 3.4.

En matlab las operaciones matriciales son muy frecuentes por tal razón es recomendable utilizarlas en lugar del uso de bucles. El simbolo (:) es uno de los más utilizados, ya que ahorra el uso de bucles en la creación de vectores, matrices, entre otras facilidades en el uso de estas plataformas de calculo numérico,

este operador es usado de diferentes formas regularmente en la creación de espacios vectoriales.

En la siguiente sentencia, $fc = fc(:)$ se ordena los valores de fc de tal forma

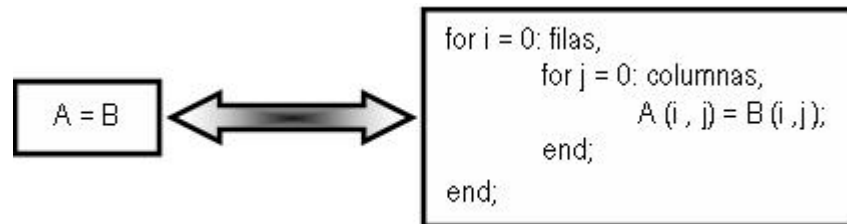


Figura 3.4: Operaciones matriciales en Matlab

que todos formen una sola columna, representando todos sus elementos en una sola columna y se interpola la señal fc , por medio de la función `bizinter(fc)`, la salida de esta operación será la función interpolada a un tamaño $2N$.

```

fc = fc(:);
fc = bizinter(fc);
fc = [zeros(N,1); fc ; zeros(N,1)];

```

La siguiente parte es una sección de condicionales para mantener la potencia a en un intervalo con el fin de aplicar el kernel de este algoritmo. Se puede observar que entre ciertos intervalos de a la variable `flag` toma valores entre 1 y 4 para después ser utilizada como condicional para realizar la función `corefrmod2(fc, a)` explicado en la sección 3.2.2 página 21.

Si a está entre 0 y 0,5 entonces $flag = 1$ y se hace $a = a - 1$

Si a está entre -0,5 y 0 entonces $flag = 2$ y se hace $a = a + 1$

Si a está entre 1,5 y 2 entonces $flag = 3$ y se hace $a = a - 1$

Si a está entre -2 y -1,5 entonces $flag = 4$ y se hace $a = a + 1$

Luego teniendo el valor de $flag$ y la potencia de a podemos preguntar por sus

casos especiales, véase figura 3.1, para realizar la Transformación de Fourier estándar, la inversa, el operador identidad y el de reflexión según los casos que le corresponde a cada uno.

Si $flag = 1$ ó $flag = 3$ entonces realiza el núcleo con $a = 1$, es decir que el ángulo tomaría un valor de $\alpha = \pi/2$.

Si $flag = 2$ ó $flag = 4$ entonces realiza la función `corefrmod2(fc, -1)`, es decir que el ángulo tomaría un valor de $\alpha = -\pi/2$.

Por último si $a = 0$ entonces realiza el operador Identidad de la función de entrada fc .

Si $flag = 2$ ó $flag = -2$ entonces realiza el núcleo con $a = 1$, es decir que el ángulo tomaría un valor de $\alpha = \pi/2$

De lo contrario se realiza la función `corefrmod2(res, a)` en el número fraccionario a

```
res = res(N+1:3*N);
res = bizdec(res);
res(1) = 2*res(1);
```

Como los datos que vienen tienen un tamaño de $4N$ se escogen los datos del centro, luego se decima por medio de la función `bizdec()`, y por último se hace el dato de la posición 1 de la Transformada como característica especial del algoritmo.

3.2.2. función `corefrmod2(fc,a)`

Esta es la función de mayor interés en el algoritmo, ya que es el que desarrolla la operación de Kernel de este algoritmo, este algoritmo se basa en que por medio de algunas operaciones algebraicas y trigonometricas, llevan la ecuación 2.5 a la

forma:

$$F_{\alpha}(y) = C_{\alpha} e^{-i\pi \tan(\alpha/2)y^2} \int_{-\infty}^{\infty} e^{i\pi \csc \alpha (y-x)^2} [e^{-i\pi \tan(\alpha/2)x^2} f(x)] dx \quad . \quad (3.8)$$

Como se puede ver en la ecuación 3.8 hacen que la *FrFT* pueda resolverse como una convolución, y tomando como base esto aplican el teorema de la Convolución e implementan un algoritmo de Convolución rápida para obtener el resultado de esta operación. Para resumir y no ahondar en detalles se mencionan las sentencias que se creen de importancia con el fin de que pueda entenderse este algoritmo, esta parte se explica mejor en el Capítulo 7 en la página 37.

Los parámetros de entrada de la función son *fc* y *a* donde *fc* es el número de muestras de una función y *a* el valor con el que se evaluará el kernel o nucleo de la *FrFT*. `function[res]=corefrmod2(fc,a);`

En la siguiente sentencia se observa la declaración de una función Chirp $e^{-i\pi \tan(\alpha/2)x^2}$, luego se hace esta chirp un vector columna oara luego realizar l multiplicación por las muestras de la señal de entrada.

```
f1 = exp(-i*pi*tan(phi/2)*x.*x); f1 = f1(:);
fc = fc.*f1;
```

Posteriormente a estas operaciones se declara ótra función Chirp $e^{i\pi \cot \alpha y^2}$, luego se hace la variable `hlptcz` la variable `hlptc` recientemente creada para declarar la segunda función chirp y se agregan ceros a la derecha de la función, lo mismo se hace con el resultado de la operación anterior, con el fin de convolucionar y de evitar la superposición entre las dos funciones.

```
hlptc =exp(i*pi*beta*t.*t);
clear t;
hlptc = hlptc(:);
```

```

N2 = length(hlptc);
N3 = 2ceil(log(N2+N-1)/log(2));
hlptcz = [hlptc;zeros(N3-N2,1)];
fcz = [fc;zeros(N3-N,1)];
Hcfft = ifft(fft(fcz).*fft(hlptcz)); clear hlptcz;
clear fcz;
Hc = Hcfft(N:2*N-1);
clear Hcfft;
clear hlptc;

```

Finalmente se declara la variable de la Transformación de Fourier fraccionaria y se multiplica con el resultado de la convolución rápida y con la función chirp declarada de primero. Y así se obtiene la Transformación de Fourier fraccionaria por medio del algoritmo de la convolución rápida mencionado en [46].

```

Aphi = exp(-i*(pi*sign(sin(phi))/4-phi/2))/sqrt(abs(sin(phi)));
xx = [-ceil(N/2):fix(N/2)-1]/deltax1;
f1 = f1(:);
res = (Aphi*f1.*Hc)/deltax1; por último se multiplica por la
Chirp3, vea ecuación 2.5

```

Capítulo 4

Marco Teórico Conceptual

Jean-Baptiste-Joseph Fourier matemático y físico francés, ver figura 4.1 conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier, las series de Fourier tienen la forma:

$$y(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)] \quad (4.1)$$

Donde a_0 , a_n y b_n se denominan coeficientes de Fourier en las series de Fourier de la función $f(x)$ ecuación??, en términos teóricos la investigación de Fourier dice que toda función que se repite periódicamente puede ser expresada como la suma de senos y/o cósenos de diferentes frecuencias, cada uno multiplicado por un coeficiente diferente [3]. Más adelante se definió que aún funciones que no son periódicas, pero con un área finita bajo la curva, es decir absolutamente integrables, pueden ser expresadas como la integral de senos y/o cósenos multiplicada por una función de ponderación o factor de fase. Esta es la transformación de Fourier, y su utilidad es aún más grande que la de las series de Fourier en muchos problemas prácticos [4].

Estas teorías son utilizadas comúnmente para el tratamiento de señales y su im-



Figura 4.1: Jean Baptiste Fourier, (1768-1830)

plementación en esta área se le llama normalmente como óptica de Fourier, luego fue creada la transformación rápida de Fourier y el teorema de muestreo de Shannon [9], lo cual permitió que el tratamiento de señales fuera realizado digitalmente sobre computadoras. Sin embargo esta área ha estado sujeta a cambios debido a las diferentes señales se tienen necesidades de ser estudiadas, y a la complejidad de cada una de ellas, para ello hay que hablar acerca de la principal división natural de las señales, estas son las estacionarias y no estacionarias.

Las señales estacionarias son constantes en sus parámetros estadísticos, es decir, si se observa una señal estacionaria, durante un instante de tiempo y después de una hora se le vuelve a observar, esencialmente se vería igual o muy parecida, en la figura 4.2 se puede ver una señal digital estacionaria.

Las señales estacionarias pueden ser representadas en el dominio del tiempo y pueden ser tratadas por la Transformación de Fourier estándar, pero las señales

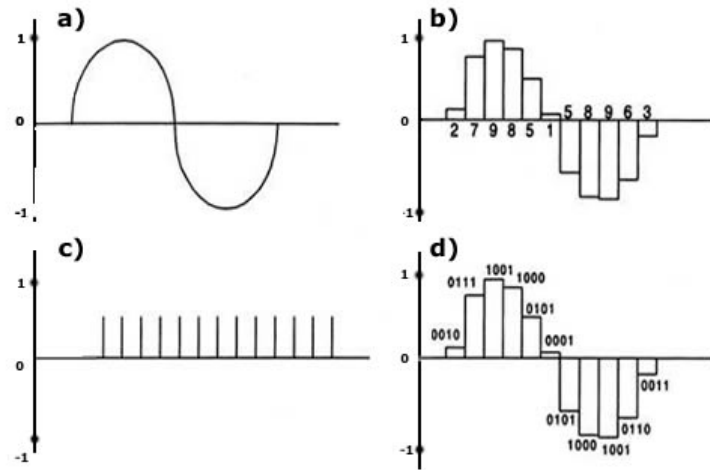


Figura 4.2: *a)* Señal Digital, *b)* Señal Digitalizada *c)* Reloj de muestreo *d)* Señal Binaria Cuantizada

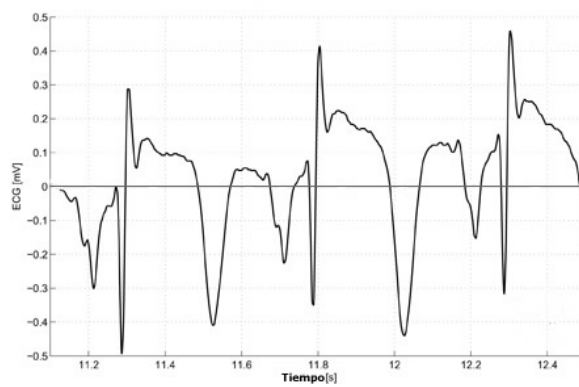


Figura 4.3: Señal no estacionaria

no estacionarias no deben ser analizadas con esta Transformación y deben ser representadas en tiempo-frecuencia, así por ejemplo, como puede observarse en la figura 4.3, un electrocardiograma, la frecuencia y amplitud de este varían mucho en un intervalo de tiempo módico y por lo tanto un análisis periódico en el dominio de la frecuencia sin tener en cuenta el tiempo es muy ineficiente para este tipo de señales; por lo cual surge la Transformación de Fourier fraccionaria para la representación de señales en tiempo-frecuencia [22].

El término de Transformación de Fourier fraccionaria aparece por primera vez en 1929 en un artículo de Norbert Wiener [26], posteriormente de forma independiente con Comdom en 1937 [34]. Desde ese momento surgieron muchas aplicaciones de esta operación, en solución de ecuaciones diferenciales, mecánica y óptica cuántica, teoría de difracción óptica y diferentes sistemas ópticos; debido a cierta complejidad en la definición que se daba en estos tiempos esta Transformación no podía utilizarse para el tratamiento de señales, la física tuvo que esperar hasta 1980, año en que Victor Namías [21] propuso una nueva definición. Luego la $FrFT$ se mostró como una herramienta útil, y los investigadores se interesaron por encontrar todas las propiedades, operaciones y teoremas que permitan el empleo de ella en un amplio rango de campos de las ciencias y las ingenierías [20, 21]. La $FrFT$ es una transformación lineal que generaliza la Transformación de Fourier estándar [22], entre las propiedades encontradas están:

- **Propiedad de Linealidad:** Esta propiedad dice que la $FrFT$ de una combinación lineal de dos funciones $f(x)$ y $g(x)$ de entrada se comportan de acuerdo a la definición de sistemas lineales. c_1 y c_2 son constantes.

$$F_{\alpha}[c_1f(x) + c_2g(x)] = c_1F_{\alpha}[f(x)] + c_2F_{\alpha}[g(x)]. \quad (4.2)$$

- **Teorema de Parseval ó Propiedad de la conservación de la energía:** Este

teorema nos dice que la energía de una señal, calculada en el dominio del tiempo es igual a la calculada en el dominio de la frecuencia.

$$\int_{\mathbb{R}} f(x)g^*(x)dx = \int_{\mathbb{R}} f_{\alpha}(y)g^*_{\alpha}(y)dy. \quad (4.3)$$

- **Teorema del Corrimiento o propiedad de traslación en tiempo:** Este Teorema nos dice que si trasladamos una función $f(x)$ en el dominio del tiempo, lo único que se altera es la distribución de fase vs. frecuencia. Siendo ς la variable de corrimiento.

$$F^a[f(x - \varsigma)] = f_a(y - \varsigma \cot \alpha) e^{i\pi \sin \alpha (\varsigma^2 \cos \alpha - 2y\varsigma)}. \quad (4.4)$$

- **Teorema de la Modulación o propiedad de cambio de la frecuencia:** Este teorema es de gran importancia para el tratamiento de la información en cualquiera de sus magnitudes, ya que esta permite hacer un cambio de la señal de entrada en su frecuencia. La propiedad de modulación es muy parecida a la de corrimiento en el tiempo. Siendo δ la variable de Modulación en la siguiente ecuación:

$$F^a[f(x)e^{i2\pi\delta x}] = f_a(y - \delta \sin \alpha) e^{-i\pi \cos \alpha (\delta^2 \sin \alpha - 2\delta y)}. \quad (4.5)$$

- **Teorema del Escalamiento:** La importancia conceptual de este teorema radica en la relación tiempo-frecuencia, la cual es la especialidad de la *FrFT* y este nos dice que si tenemos una función $f(x)$ para expandir en tiempo la función obligatoriamente se debe comprimir la frecuencia de la misma; Intuitivamente esto es lógico ya que comprimir una señal en tiempo equivale a hacer sus cambios más bruscos, por lo tanto tienen que aparecer compo-

nentes de mayor frecuencia.

$$F^a[f(cx)] = \sqrt{\cos \beta / \cos \alpha} e^{\frac{1}{2}(\alpha - \beta)} e^{i\pi x'^2 \cot \alpha (1 - \frac{\cos^2 \beta}{\cos^2 \alpha})} f_\beta(x' \frac{\sin \beta}{c \sin \alpha}), \quad (4.6)$$

donde $\tan \beta = c^2 \tan \alpha$.

De forma casi simultanea en 1993 un grupo de investigadores Almeida, Mendlovic, Lohmann, Ozaktas, “reinventaron” la *FrFT* [22], algunos la interpretaron como una rotación del plano tiempo-frecuencia. Gracias a esta interpretación en la actualidad la Transformación de Fourier fraccionaria es implementada en el tratamiento de señales, y ha suplido muchas falencias de la Transformación de Fourier ordinaria siendo la operación más utilizada para el tratamiento de señales dada sus ventajas sobre el tratamiento de señales no estacionarias [41]. Así la potencia de la fracción realizada en f_a de la operación de la transformación de Fourier ordinaria f consiste en la rotación de la distribución de Wigner que se encuentra asociada a la señal en un ángulo $a\pi/2$ en el tiempo-frecuencia. De ahí la gran relación que tiene esta transformación con la distribución de Wigner [25]. La Distribución de Wigner es una función que extrae la energía de una señal representada en tiempo-frecuencia [43]. La distribución de Wigner de una función $f(x)$ está definida según la ecuación 3.1 [20].

El cálculo matemático de esta operación realizado en los computadores modernos hace de la Transformación de Fourier Fraccionaria una herramienta de gran importancia y utilidad para aplicaciones en el tratamiento de señales en las cuales es manipulada. Varios algoritmos han sido descritos, sin embargo estos algoritmos pueden ser clasificados de dos tipos: Transformación Rápida de Fourier Fraccionaria [44], estos son generados utilizando las funciones de la Transformación Rápida de Fourier y operaciones de convolución, el otro tipo son las Transformación Discreta fraccionarias de Fourier.

El nuevo modelo matemático de la Transformación de Fourier fraccionaria y sus aplicaciones para la recuperación de la fase, caracterización de rayos de luz, filtros, encriptación, marca de agua, entre otros [25, 26], ha llevado a que en la última década la rama de la óptica de Fourier halla avanzado en el tratamiento de una cantidad de señales.

Capítulo 5

Justificacion

La línea de Investigación del Procesamiento Digital de Señales, enmarca tópicos interesantes tales como procesamiento digital de señales estacionarias y no estacionarias, unidimensionales y/o multidimensionales, continuas y discretas, en tiempo real, en audio, en video, en reconocimiento de voz, en visión por computadora, entre otras [3, 4].

En esta investigación se hizo un tratamiento de señales completamente en el dominio fraccionario de fourier, todo esto por medio del desarrollo de funciones en una aplicación realizada sobre la plataforma de calculo numérico Scilab. Este Tratamiento puede ser aplicado en muchas áreas de investigación [23, 24] y últimamente ha tenido mucho auge en el procesamiento de imágenes digitales [25], gracias a la extracción de información de las señales representadas en tiempo-frecuencia en cualquier instante de tiempo [22] y su mayor ventaja se fundamenta en el tratamiento de señales no estacionarias.

El estudio de señales no estacionarias siempre ha representado mayor preocupación en el campo científico, ya que muchos factores en áreas de biofísica, electrofísica, geofísica, entre otras muchas se generan aleatoriamente a lo largo del

tiempo, teniendo de cambiar el Δx , no obstante el poder hacer una rotación en el espacio tiempo-frecuencia, cambiar la fase de las señales en cualquier intervalo de tiempo y utilizar esta información para el análisis, estudio y tratamiento de señales son muchas de las ventajas que tiene la Transformación de Fourier fraccionaria.

Junto con la *FrFT* se implementa un filtro para la restauración de señales llamado filtro de Wiener fraccionario [19, 39], y se busca una efectividad de estas dos operaciones frente a otras utilizadas anteriormente para realizar calculos en el dominio de Fourier estándar, para esto se requiere hacer un estudio detallado de los fundamentos matemáticos de estas dos operaciones, comprender su universalidad y los límites de su aplicabilidad en el área de la óptica y el tratamiento de señales [37].

5.1. Viabilidad Técnica

La elección del software que se usa en una investigación debe ser una decisión fundamental. Para el desarrollo de este proyecto se utilizará la herramienta gratuita Scilab, esta herramienta es un paquete de software de cálculo muy potente en lo que a matrices se refiere ya que fue creado para trabajar con ellas, también posee una extraordinaria versatilidad y capacidad para resolver problemas de matemática aplicada, física, ingeniería, procesamiento de señales y otras muchas aplicaciones. Scilab utiliza una sintaxis parecida a la de Matlab. La utilización de esta herramienta produce algunas ventajas, entre ellas:

- El software hasta ahora es gratis.
- La última versión del software está siempre disponible.
- El acceso al código fuente facilita el aprendizaje de esta herramienta.

- La información y documentación de calidad está siempre disponible.
- Sus procesos los realiza de forma paralela, permitiendo y facilitando la implementación de sistemas distribuidos para tareas que así lo requieran.

El código de la Transformación de Fourier fraccionaria está hecho en la plataforma Matlab, sin embargo, la forma en que es realizada esta función no tiene en cuenta el teorema de muestreo en dominios fraccionarios [38] para la discretización del Kernel lo cual hace que no podamos tener un estudio total y puntual sobre la frecuencia; Scilab tiene funciones que permiten exportar códigos de Matlab [31], sin embargo esta conversión la mayoría de las veces es imperfecta, ya que existen funciones de Matlab como `eye`, `ones`, `size`, `sum`, entre otras [40], que se comportan de diferente forma según la dimensión de sus argumentos y si Scilab no puede deducir el código que está exportando, lo que hace es sustituir la función de llamada correspondiente dentro del fichero `.m` que se está traduciendo, por una función `mltb-<nombre-función>` en Scilab, y no todas las funciones tienen esta función de traducción.

5.2. Viabilidad Económica

Los Proyectos de investigación y desarrollo de software se caracterizan por ser económicos en cuanto a la adquisición de herramientas para llevar a cabo estas investigaciones. Scilab nos provee de muchas facilidades, entre ellas es que es libre y además es gratis; algunas veces se malentienden estos términos y algunas personas los confunden como sinónimos. Cuando hablamos de software libre hablamos de su código fuente, el precio está sometido a las utilidades y necesidades de los usuarios.

5.3. Viabilidad Social

La investigación matemática puede y debe contribuir a todos los aspectos involucrados en el procesamiento de señales: tanto al desarrollo del modelo matemático y su implementación computacional. Para cubrir las necesidades que pueden suplir investigaciones matemáticas se requiere de equipos y de trabajo multidisciplinarios integren a ingenieros y matemáticos.

La interacción con centros de investigación o grupos de investigación de instituciones académicas de educación superior en este caso la Universidad Industrial de Santander (*UIS*) por medio del Grupo de Óptica y Tratamiento de Señales (*GOTS*), la UIS siempre ha estado activa y participe a colaborar en los procesos académicos de la Universidad del Magdalena.

Capítulo 6

Objetivos

6.1. Objetivo General

Desarrollar una aplicación utilizando Scilab para realizar e implementar la Transformación de Fourier fraccionaria y el Filtro de Wiener fraccionario para el tratamiento de señales digitales.

6.2. Objetivos Específicos

- Estudiar y analizar los fundamentos matemáticos de la Transformación de Fourier fraccionaria para el procesamiento de señales digitales.
- Implementar el Teorema de Muestreo fraccionario en la Transformación de Fourier fraccionaria.
- Demostrar la relación existente entre la Transformación de Fourier fraccionaria y la Distribución de Wigner.
- Desarrollar las diferentes funciones para obtener la Transformación de Fourier fraccionaria y el filtro de Wiener fraccionario para el tratamiento de señales

digitales.

- Realizar manual técnico, manual de usuario y demás documentación para el soporte de la aplicación.

Capítulo 7

Formulación & Hipótesis

Según el artículo [46] la ecuación de la transformación de Fourier fraccionaria puede verse sometida a un cambio algebraico y trigonométrico, con el fin de que esta pueda resolverse por medio de una convolución, partiendo desde la Ecuación 2.5, se ingresan todos los términos con exponenciales y al agrupar los terminos similares queda:

$$f_{\alpha}(y) = C_{\alpha} \int_{\infty}^{\infty} f(x) e^{i\pi(y^2 \cot \alpha + x^2 \cot \alpha - 2xy \csc \alpha)} \quad (7.1)$$

Luego se suman la siguientes expresiones, con el fin de completar cuadrados perfectos,

$$x^2 \csc \alpha - x^2 \csc \alpha + y^2 \csc \alpha - y^2 \csc \alpha = 0$$

y la ecuación 7.1 queda de la siguiente manera:

$$f_{\alpha}(y) = C_{\alpha} \int_{\infty}^{\infty} f(x) e^{i\pi(y^2(\cot \alpha - \csc \alpha) + x^2(\cot \alpha - \csc \alpha) + (x-y)^2 \csc \alpha)} \quad (7.2)$$

después de la comprobación de que $\cot \alpha - \csc \alpha = \tan \alpha/2$ se reemplaza esto en la ecuación 7.2 y finalmente se puede observar la ecuación creada por Namías, como un producto de una Función Chirp por una convolución, ver ecuación 3.8

[28].

7.1. Transformación de Fourier Discreta

La Transformación de Fourier Discreta Vista en muchas partes como *DFT* Discrete Fourier Transform por su siglas en inglés es una función ampliamente empleada para el tratamiento de señales y campos a fines, con el fin de analizar las frecuencias que tienen un número de muestras de una señal o función, esta función puede calcularse eficientemente desde la Transformación rápida de Fourier *FFT* explicada en la sección 7.1.1, la ecuación 2.1 se discretiza de la siguiente manera:

$$x = n\Delta x \quad y = k\Delta y$$

$$\mathfrak{F}(k) = \sum_{n=N/2}^{n=N/2-1} f(n)e^{-2i\pi n\Delta x k\Delta y} \quad (7.3)$$

Los valores con Δ se refiere al espacio que debe haber entre cada una de las muestras con el fin de que la señal pueda ser reconstruida perfectamente, por lo cual según el teorema de Shannon si ξ es la duración de la señal el cual debe ser equivalente al tamaño de la señal, y ζ es el ancho de banda de la Transformada, ver figura 7.1, entonces:

$$\Delta x = \frac{\xi}{N} \quad \Delta y = \frac{\zeta}{N} \quad \xi\zeta = N$$

y de esta forma la ecuación 7.3 queda de la siguiente forma

$$\mathfrak{F}(k) = \sum_{N/2}^{N/2-1} f(n)e^{\frac{-2i\pi nk}{N}} \quad (7.4)$$

Tamaño	Operaciones DFT	Operaciones FFT	Mejora
2	16	4	4
4	256	32	8
8	4096	192	21.33
16	$6,554 * 10^4$	1024	64
32	$1,049 * 10^6$	5120	201.8
64	$1,678 * 10^7$	$2,458 * 10^4$	682.67
128	$2,684 * 10^8$	$1,147 * 10^5$	2341
256	$4,295 * 10^9$	$5,243 * 10^5$	8192
512	$6,872 * 10^{10}$	$2,359 * 10^6$	$2,913 * 10^4$
1024	$1,100 * 10^{12}$	$1,049 * 10^7$	$1,049 * 10^5$

Cuadro 7.1: Comparación del número de operaciones entre la DFT y el algoritmo de la FFT de señales Bidimensionales cuadradas

7.1.1. Transformación Rápida de Fourier

EL algoritmo de la FFT lo que busca es hacer una aproximación lo más exacta de la DFT pues como puede verse en la ecuación 7.4 el factor de fase de la función hace que se tenga que hacer N^2 multiplicaciones entre n y k para hallar el resultado. El algoritmo de la FFT hace que se encuentren valores muy cercanos a la transformación de Fourier real, en $N * \log_2(N)$ operaciones, en este algoritmo se encuentra una forma por medio de división modular para que no se repitan valores en el factor de fase de la ecuación 7.4, y así reducir el tiempo de ejecución de esta operación.

7.2. Transformación de Fourier fraccionaria Discreta

De la misma forma como la DFT tiene una forma de ser realizada rápidamente la $FrFT$ también tienen diferentes formas de ser implementada, ya sea la mencionada en la sección 3.2 u otras utilizando algoritmos de la FFT . La forma

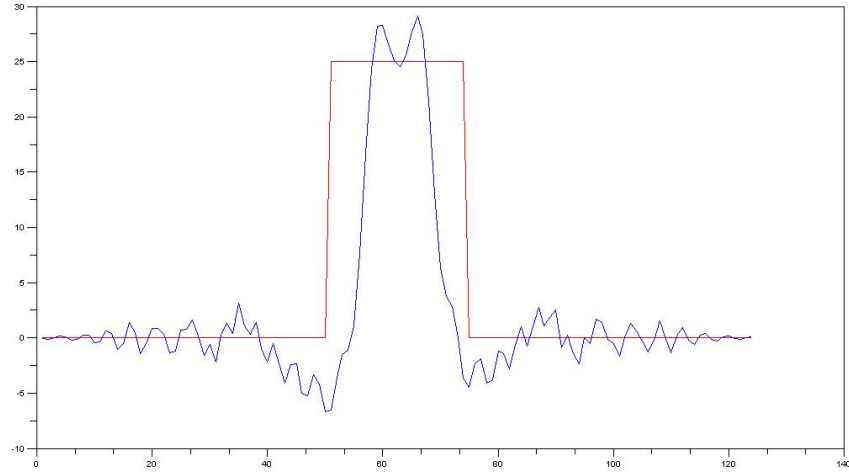


Figura 7.1: El tamaño de la Señal es el mismo que el Ancho de Banda de la Transformada

discreta de la ecuación 2.5 está dada por la siguiente expresión.

$$x = n\Delta x \quad y = k\Delta y$$

$$f_{\alpha}(k) = C_{\alpha} e^{i\pi k^2 \Delta y^2 \cot \alpha} \sum_{n=N/2}^{n=N/2-1} f(n) e^{i\pi n^2 \Delta x^2 \cot \alpha} e^{\frac{-2i\pi n \Delta x k \Delta y}{\sin \alpha}} \quad (7.5)$$

$$\Delta x = \frac{\xi}{N} \quad \Delta y = \frac{\zeta}{N}$$

y de esta forma la ecuación 7.5 queda de la siguiente forma:

$$f_{\alpha}(k) = C_{\alpha} e^{\frac{i\pi k^2 \zeta^2 \cot \alpha}{N^2}} \sum_{n=N/2}^{n=N/2-1} f(n) e^{\frac{i\pi n^2 \xi^2 \cot \alpha}{N^2}} e^{\frac{-2i\pi n k \zeta \xi}{N^2 \sin \alpha}} \quad (7.6)$$

Teorema de Muestreo en dominios fraccionarios 1 Sea $f(x)$ una función tal que $f_{\alpha}[y]$ tiene soporte finito ζ , $f(x)$ puede ser muestreada y reconstruida perfectamente si las muestras se toman a una tasa $\Delta x \leq \sin \alpha / \zeta$ [38].

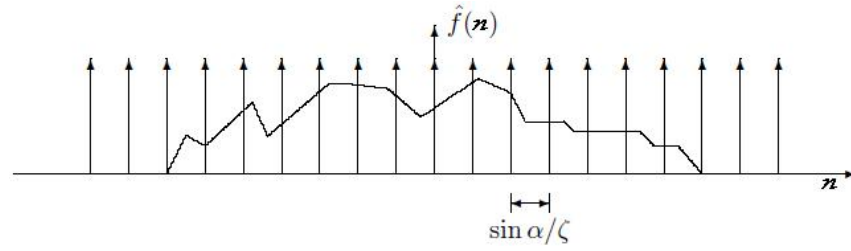


Figura 7.2: Δx en el teorema de Muestreo fraccionario

El teorema de muestreo estandar de Shannon-Whittaker se encuentra inmerso como caso particular de este, cuando $\alpha = \pi/2$; tomando en cuenta el Teorema de Muestreo en dominios fraccionarios se infiere lo siguiente:

$$\frac{\xi}{N} = \frac{\sin \alpha}{\zeta} \quad \Rightarrow \quad \xi \zeta = N \sin \alpha \quad \xi = \zeta = \sqrt{N \sin \alpha}$$

Según lo anterior se puede definir una Transformación de Fourier fraccionaria discreta que cumple todas las propiedades y características de la ecuación 2.5, la cual, puede solucionarse por diferentes métodos utilizando el algoritmo de la *FFT* u otras operaciones en su defecto [43], lo cual es explicado en el capítulo 10; al incluir este teorema del muestreo fraccionario la *DFrFT* queda de la siguiente forma:

$$f_{\alpha}(k) = C_{\alpha} e^{\frac{i\pi k^2 \cos \alpha}{N}} \sum_{n=N/2}^{n=N/2-1} f(n) e^{\frac{i\pi n^2 \cos \alpha}{N}} e^{\frac{-2i\pi nk}{N}} \quad (7.7)$$

El kernel discreto en la ecuación 7.7 contiene el kernel discreto de la transformación de Fourier estándar mostrado en la ecuación 7.4, permitiendo hacer los cálculos con base en la función *FFT* explicado en la sección 7.1.1.

7.3. Convolución Fraccionaria

Gracias a la convolución en el dominio directo de las señales se pueden realizar infinidad de operaciones como filtros, detección de patrones, entre otras.

Pero ahora, con el advenimiento del tratamiento de señales en el dominio fraccionario llegaron nuevas operaciones como la convolución y la correlación fraccionaria [25, 50], la convolución fraccionaria de dos funciones $f(x)$ y $g(x)$ se denota de la siguiente manera:

$$[f \overset{*}{\alpha} g] = \mathfrak{F}_{-\alpha}[\mathfrak{F}_{\alpha}[f]\mathfrak{F}_{\alpha}[g]] \quad (7.8)$$

Teniendo en cuenta la tabla 3.1 se nota que cuando el ángulo de la *FrFT* es $\pi/2$ se realiza una Transformación de Fourier estándar por lo cual la convolución fraccionaria será una convolución estándar también, y si $\alpha = 0$ entonces se cumplirá el caso del operador identidad para ambas funciones y se realizará un simple producto entre estas dos funciones [51, 52]. La Convolución fraccionaria se soluciona en forma de operadores utilizando *FrFT* como se muestra adelante.

$$[f \overset{*}{\alpha} g](x) = \mathfrak{F}_{-\alpha}[\mathfrak{F}_{\alpha}[f](x_{\alpha})\mathfrak{F}_{\alpha}[g](x_{\alpha})e^{i\pi x^2 \cot \alpha}](x). \quad (7.9)$$

Y la forma integral se muestra de la siguiente forma, ver capítulo 10:

$$[f \overset{*}{\alpha} g](x) = \int f(y)g(x-y)e^{2i\pi y(x-y) \cot \alpha} dy. \quad (7.10)$$

Capítulo 8

Metodología

Al considerar la implementación computacional de la Transformación de Fourier fraccionaria y el filtro de Wiener fraccionario sobre la plataforma Scilab, se buscan las ventajas del tratamiento de señales en el dominio fraccionario de Fourier frente al análisis estándar y otras que se venían realizando en el tratamiento de señales representadas en tiempo frecuencia [47]. Esta práctica requiere de mucho sacrificio de tiempo en investigación en fases como análisis y requerimientos de la información para llevar a cabo esta investigación.

8.1. Metodología *XP*

Tras buscar metodologías de desarrollo de software que se adaptaran a esta investigación, se pudo constatar que todas presentan fases y conceptos en común, por tal razón, para seleccionar la metodología en esta investigación se tomó en cuenta algunos aspectos de tiempo y personal presente durante el ciclo de vida de la aplicación, la metodología que se ajusta a esta investigación es la *Xtreme Programming* (*XP* por sus siglas en inglés). *XP* es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer, está basada en una

serie de valores y de prácticas de buenas maneras de programación que persigue el objetivo de aumentar la productividad a la hora de desarrollar aplicaciones de software, esta consta de 4 fases: análisis, diseño, desarrollo y pruebas.

La interacción de estas cuatro fases las podemos describir como un rompecabezas figura 8.1, en la medida en que se avanza por cada una de las fases siempre hay comunicación entre ellas, la fase de pruebas se encuentra inmersa durante todo el proceso de desarrollo, de esta forma siempre hay una retroalimentación continua a través de las fases que se enmarcan en la metodología *XP*. Utilizar esta técnica para el desarrollo de esta aplicación no solo es posible sino que también puede ser provechosa para nuestra experiencia en el desarrollo de aplicaciones en el campo del software libre. La documentación se hace al final de aplicación; a continuación se relatan las actividades que se realizarán durante la investigación según cada una de las fases de la metodología seleccionada.

8.1.1. Análisis

Esta fase es la más importante de todas ya que en ella se realiza el estudio de los fundamentos matemáticos de la transformación de Fourier fraccionaria, sus propiedades, y el filtro de Wiener fraccionario sobre señales digitales; de la misma forma se trazan a lapiz, los modelos de las diferentes funciones a utilizar para llevar a cabo las operaciones que permitan culminar los objetivos propuestos. También se toman en cuenta los requerimientos funcionales de la aplicación y se modelan los procesos a través de las diferentes fases aquí expuestas.

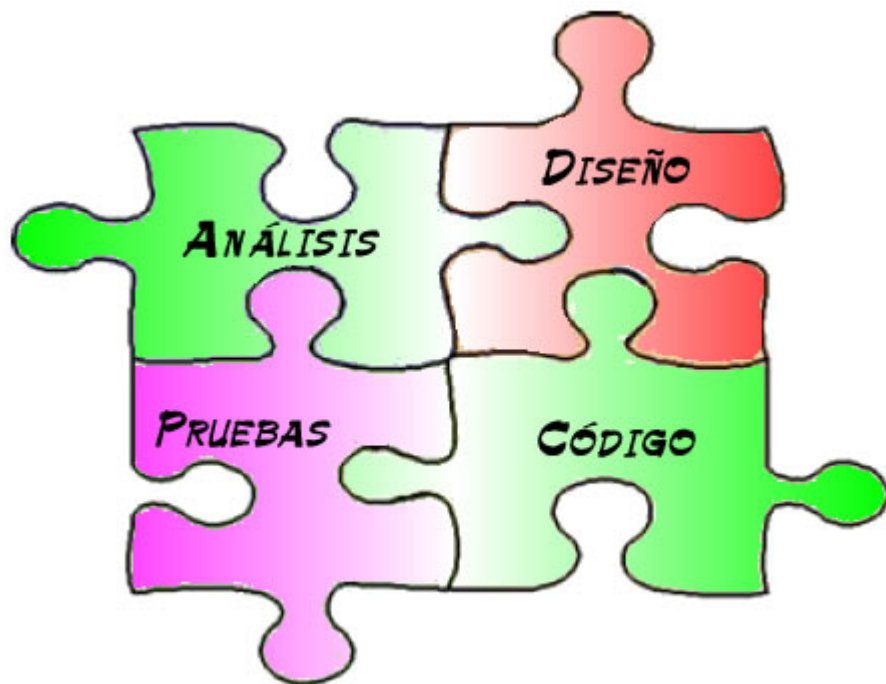


Figura 8.1: Metodología XP

8.1.2. Diseño

En esta fase se establece el diseño de la interfaz que interactuará con el usuario y el esquema de funciones que se necesitarán para llevar a cabo la Transformación de Fourier fraccionaria, la operación de la distribución de Wigner, la operación de convolución y el filtro de Wiener fraccionario, quizás también otras transformaciones representadas en tiempo-frecuencia para obtener conclusiones experimentales en la investigación. En la aplicación se implementa un diseño simple, para enfocarse en los resultados científicos que se obtienen, y también porque una simplicidad del diseño, hará que sea más fácil añadir nuevas funcionalidades en un futuro.

8.1.3. Desarrollo

Teniendo en cuenta que en esta fase se ha entendido el fondo y la teoría matemática que abarca la $FrFT$ y el filtro de Wiener fraccionario, como también la discretización de cada una de las formulas a utilizar en estos algoritmos; se comienza a desarrollar el código de estas operaciones y también se establecen los mecanismos y pautas, para que sea revisado la fase de diseño de la investigación.

En esta fase se hace la integración de las funciones y los recursos necesarios para que el sistema funcione y se obtenga un óptimo procesamiento de señales digitales representadas en tiempo-frecuencia en el dominio fraccionario de Fourier. Si seguimos al pie de la letra las normas y técnicas de la presente metodología de desarrollo, al final de esta fase se obtendrá un código simple y funcional.

8.1.4. Pruebas

Scilab es una plataforma rápida que permite ejecutar rápidamente funciones en el interprete de código, por lo cual desde el primer momento en que vamos programando vamos realizando pruebas en la aplicación, y de esta forma nos aseguramos que el código escrito hasta entonces en algún momento es correcto y no estropea lo anterior.

Por tal razón esta fase no está precisamente al final sino durante todo el proceso del ciclo de vida del software, y al final de la fase de desarrollo estaría la aplicación debiera estar lista. Sin embargo, se hacen las suficientes pruebas para que la aplicación sea estable y se puedan cumplir con los objetivos propuestos en la presente investigación.

Capítulo 9

Desarrollo del Trabajo

En este capítulo se muestran los pasos que se realizaron para desarrollar las funciones presentadas en esta aplicación, y de la misma forma algunos que no se pudieron implementar ya que no arrojaban los resultados esperados.

Como se muestra en la ecuación 7.7 al implementar el Teorema de Muestreo fraccionario el cual dice que la $\Delta x = \sin \alpha / \zeta$, siendo ζ el ancho de banda de la transformada de Fourier de la señal a muestrear. Teniendo en cuenta la formulación de la transformación de Fourier estándar descrita en la ecuación 2.1, la transformación de Fourier fraccionaria puede ser implementada utilizando el algoritmo de la Transformación de Fourier de la siguiente forma:

$$f_{\alpha}(k) = C_{\alpha} e^{\frac{i\pi k^2 \cos \alpha}{N}} \mathfrak{F} \left[f(n) e^{\frac{i\pi n^2 \cos \alpha}{N}} \right]. \quad (9.1)$$

Al implementar el kernel descrito en la ecuación 9.1, podemos obtener la *FrFT* para un intervalo fraccionario $0.5 < |a| < 1.5$, para hallar los valores que no están en este intervalo se deben aplicar transformaciones de Fourier estándar e inversas, y luego se le adiciona o se le resta en complemento al valor fraccionario a ; esto se explica con el siguiente código, el cual es implementado en la función *FrFT* desarrollada en esta investigación.

Con el fin de reducir el intervalo fraccionario de la *FrFT*, lo primero que se hace es hallar el módulo del número a , el cual es un parámetro de entrada de la función. $a = \text{modulo}(a, 4)$; . Al hacer esto se pregunta por el valor absoluto de a , y si este es igual a 2, se aplica la propiedad de reflexión el cual es un caso particular de la *FrFT*, de lo contrario se reduce el intervalo de fraccionalización a .

```
if (abs(a) == 2)
    Fy = fc($:-1:1, :)
else
    a = modulo(a, 2); end;
```

Luego de tener el valor fraccionario de la *FrFT* entre $0 \leq |a| < 2$, lo primero que hay que hacer es descartar los intervalos que no son posibles de realizar directamente con el kernel de la función. Otro caso particular es la propiedad identidad la cual se da cuando $a = 0$, el cual se nota en las dos primeras sentencias que se muestran en el siguiente código.

En las sentencias anteriores se observa una reducción en el intervalo del operador fraccionario de la *FrFT* y se implementa el código de la Transformación rápido de Fourier, explicado en la sección 7.1.1, y se le suma algebraicamente el excedente para que realice la operación en el operador que se requiere, por ejemplo:

```
if (a == 0)
    Fy = fc;
else
    if (a > -0.5) & (a < 0)
        Fy = kernelFrFT(fftshift(fft(fftshift(fc))), -1 - abs(a));
```



```

end;

if (a>0) & (a<0.5)
    y = kernelFrFT(fftshift(fft(fftshift(fc))),a-1);
end;
if (a>-2) & (a<-1.5)
    Fy = kernelFrFT(fftshift(fft(fftshift(fc),1)),1-abs(a));
end;
if (a>1.5) & (a<2)
    Fy = kernelFrFT(fftshift(fft(fftshift(fc),1)),a-3);
end;
if (abs(a)>=0.5) & (abs(a)<=1.5)
    Fy = kernelFrFT(fc,a);
end;
end;

```

Si se supone $a = 0,1$, el cual encaja perfectamente en el segundo condicional, entonces se aplica una FFT a la función de entrada fc , y se le resta 1 al operador fraccionario, con esto el Kernel de la $FrFT$ realizará una Transformación en el orden $-,9$ por la propiedad de aditividad que mantiene la $FrFT$, en forma de operadores será entendido mejor.

$$a = 0,1;$$

$$0,1 - 1 = -0,9;$$

$$\mathfrak{F}_{1-0,9} = \mathfrak{F}_{0,1}$$

9.1. Función KernelFrFT

Como ya se había dicho anteriormente el kernel de la $FrFT$ puede ser implementado utilizando el algoritmo de la Transformación de Fourier en su kernel,

y se logró para esta investigación, sin embargo, este algoritmo da una aproximación muy buena de la *FrFT* pero por problemas en su fase ver figura 9.1, y no cumple varias propiedades de esta Transformación de Fourier fraccionaria como es la aditividad, no obstante, en futuras investigaciones se podrá solucionar este inconveniente y se logrará utilizar la *FFT* en la *FrFT*, haciendo este algoritmo más rápido y eficiente

En las siguientes sentencias de código se muestra el kernel que no funcionó para la función de la Transformación de Fourier fraccionaria, claramente se muestra la multiplicación de una función chirp por la función de entrada, para luego aplicar a este resultado una Transformación rápida de Fourier y luego multiplicar por otra función Chirp y la constante mostrada en la ecuación 2.6

```
x = [-ceil(N/2):fix(N/2)-1];
chirp1 = exp((%i*%pi*cotg(alpha)*(x.*x))/N);
chirp1 = chirp1(:);
fch = fc.*chirp1;
Hcfft = fftshift(fft(fftshift(fch),-1));
Aphi=exp(-%i*(%pi*sign(sin(alpha))/4-alpha/2))/sqrt(abs(sin(alpha)));
y = (Aphi*chirp1.*Hcfft);
```

Por tal razón con el fin de solucionar la expresión de la ecuación 9.1 se hace uso de la discretización total de la señal de la siguiente manera, explicada en la sección 7.2:

```
x = [-ceil(N/2):fix(N/2)-1];
x=x(:);
chirp1 = exp((%i*%pi*cotg(alpha)*(x.*x))/N);
```

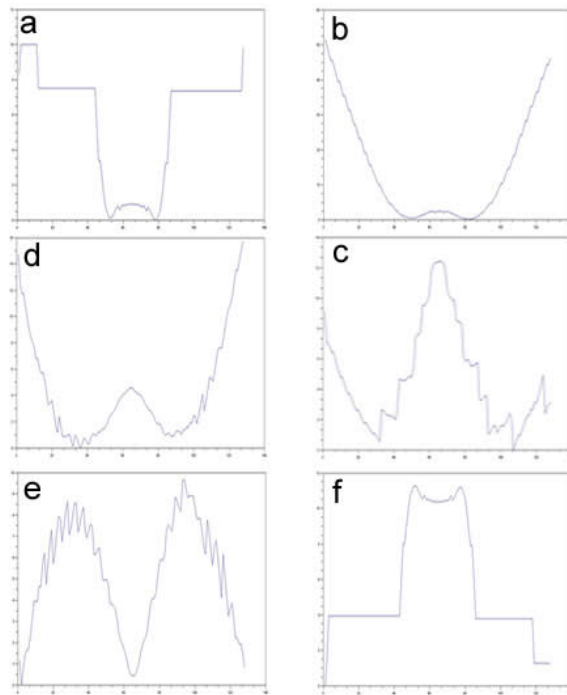


Figura 9.1: Fases de la *FrFT* con el algoritmo de la *FFT* en el kernel a) $a = 0,1$; b) $a = 0,3$; c) $a = 0,5$; d) $a = 0,9$; e) $a = 1,3$; f) $a = 1,9$.

```
chirp1 = chirp1(:); fc = fc.*chirp1;
W = exp(-2*i*pi*(x*x')/(sin(alpha)*N));
Hcfft=W*fc;
Aphi = exp(-i*(pi*sign(sin(alpha))/4-alpha/2)/sqrt(abs(sin(alpha))));
y = (Aphi*chirp1.*Hcfft);
```

En las sentencias de código arriba mostradas, se observa la declaración de una función Chirp de la siguiente forma $e^{\frac{i\pi \cot \alpha x^2}{N}}$, luego esta es multiplicada por la función de entrada y posteriormente este resultado es multiplicado por una matriz de $N \times N$, realizando una operación matricial, donde se obtiene una sumatoria del producto de cada una de las filas de la matriz W por el vector resultante. Después se multiplica por la constante y otra función Chirp similar a la descrita anteriormente.

Para explicar el hecho anterior véase la transformación de Fourier discreta con el teorema de muestreo estandar:

$$\mathfrak{F}(k) = \sum_{N/2}^{N/2-1} f(n) \left[e^{\frac{-2i\pi}{N}} \right]^{nk}. \quad (9.2)$$

Según la ecuación anterior al introducir el teorema de muestreo en dominios fraccionarios esta quedará de la siguiente forma:

$$W = \sum_{N/2}^{N/2-1} \left[e^{\frac{-2i\pi}{N}} \right]^{\frac{nk}{\sin \alpha}}. \quad (9.3)$$

9.2. Convolución fraccionaria

Como se menciona anteriormente en la sección 7.3, la convolución fraccionaria de la ecuación 7.8 presenta casos particulares según el grado de fraccionalización, de la misma forma el integral mostrado en la ecuación 9.4, la cual realiza esta operación puede ser solucionado de diferentes formas [52] entre ellas: Al evaluar el factor de fase del integral resulta la expresión ??:

$$[f \overset{*}{\alpha} g](x) = \int f(y)g(x-y)e^{\pi i y^2 \cot \alpha} e^{-2\pi i x y \cot \alpha} dy \quad (9.4)$$

otra forma es la siguiente la mostrada en la ecuación 9.5

$$[f \overset{*}{\alpha} g](x) = e^{\pi i x^2 \cot \alpha} (f'_\alpha * g'_\alpha) \quad (9.5)$$

siendo

$$f'_\alpha = f(x)e^{\pi i x^2 \cot \alpha} \quad y \quad g'_\alpha = g(x)e^{\pi i x^2 \cot \alpha}$$

La forma más fácil y rápida de implementar la convolución fraccionaria se expone en la siguiente expresión:

$$h(k) = \sum_{n=N/2}^{n=N/2-1} f(n)g(k-n)e^{\frac{-2\pi i k(n-k) \cot \alpha}{N}} \quad (9.6)$$

otra forma se realiza tomando la forma de la ecuación 9.4, donde se observa que la forma del factor que multiplica ambas funciones se asemeja la transformación de Fourier estándar, con la diferencia que el factor de fase tendría la función $\cot \alpha$; esta forma quedaría de la siguiente manera:

$$h(k) = \sum_{n=N/2}^{n=N/2-1} [f(n) \times Chirp] \times W * [g(n)W] Chirp \quad (9.7)$$

siendo

$$W = e^{-2i\pi n k \cot \alpha} \quad y \quad Chirp = e^{-\pi i n^2 \cot \alpha}$$

después de realizar estos productos se aplica la función de Convolución estándar.

En esta investigación se lograron implementar estos algoritmos, pero no fue posible hacer que dieran los resultados esperados debido a incoherencias que se obtenían en su simulación. Por lo cual se toma la decisión de implementar la convolución fraccionaria aplicando la *FrFT* por medio de operadores descrita en la ecuación 7.9 del capítulo 7 la cual sumple el teorema de la convolución planteado en el Teorema 1 del capítulo 3.

9.3. Filtro de Wiener fraccionario

El proceso de filtrar una señal $f(x)$ casi siempre se ha entendido en términos frecuenciales. Es decir, los filtros realizados a señales están fuertemente relaciona-

dos con la transformación de Fourier. Todo el proceso de filtrado consiste en si un filtro de función $g(x)$ es una versión paso bajo, paso banda o paso alto de la original [3].

Lo anterior ha restringido demasiado el proceso realizado por un filtro $g(x)$, el cual venía determinado por la viabilidad tecnológica de producir la salida $h(x)$, siendo esta otra versión de la señal de entrada. La puesta en marcha del tratamiento digital de señales no es más que el uso de la tecnología capaz de realizar cualquier operación matemática sobre una señal $f(x)$, más específicamente sobre su versión muestreada $f((n))$, esto abrió enormemente las perspectivas de los tipos de filtrado que podían ser realizados sobre una función, llegando incluso hasta predecir [39, 50].

El objetivo de esta sección es plantear el diseño de una respuesta impulsional (filtro) $g(x)$, de modo que la salida $h(x)$ sea lo más parecidamente posible a una señal, denominada referencia, $r(x)$ [4].

Sea $f(x)$ una señal que presenta un ruido aditivo aleatorio, tal que $f(x) = e(x) + r(x)$, siendo e la escena y r el ruido, entonces, se propone un filtro el cual a través de la operación de convolución fraccionaria explicada en la sección 9.2, elimina el ruido sumergido en la naturaleza de la señal.

Este proceso se describe así, en la figura 9.2 se muestra la transformada de una señal rectángulo de 1024 muestras con un operador fraccionario $a = -.5$ lo que representaría una rotación de la distribución de Wigner en un ángulo $\alpha = -\frac{\pi}{4}$, ver figura 9.3, luego en el dominio fraccionario de Fourier se le agrega un ruido aditivo de una función chirp multiplicado por una función gaussiana afectada por un valor randómico obteniendo un ruido no estacionario de frecuencias y

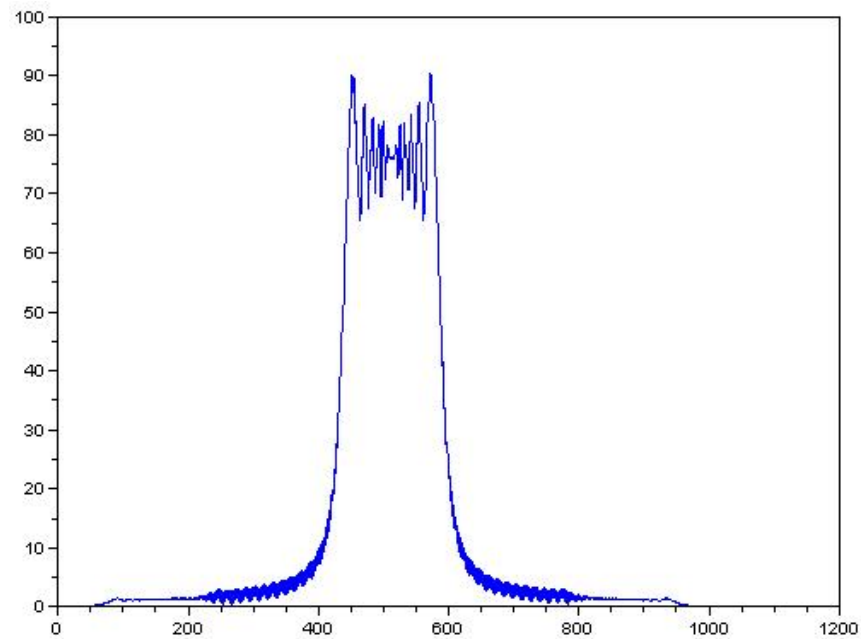


Figura 9.2: $FrFT$ con $a = -0,5$ de una función rectángulo de 1024 muestras

amplitudes linealmente variantes en el tiempo, el valor absoluto de este resultado se muestra en la figura 9.4 y en la figura 9.5 su valor real donde se muestra perfectamente la diferencia entre las dos señales, en la figura 9.6 se visualiza la distribución de wigner asociada a la señal de la figura 9.4 notándose que en el dominio fraccionario de la señal el ruido se encuentra en su dominio directo, es decir al regresar la señal rectángulo a su dominio directo el ruido se encontrará en el dominio fraccionario de la señal, con el fin de regresar la señal a su dominio directo se niega el operador de fraccionalización a , en este caso, al aplicar una $FrFT$ con $a = 0,5$ a la transformada de la figura 9.2, se obtiene nuevamente la señal en su dominio directo afectada por el ruido aditivo tal como se muestra en la figura 9.7, luego se grafica la distribución de wigner y se observa el ruido en un dominio fraccionario de la señal original, ver figura 9.8.

Para el caso en que $X = [Y_{\alpha}^* D] + B$, siendo Y la señal de referencia, B el ruido aditivo en la señal y D la distorsión, entonces se propone la ecuación 9.8 como

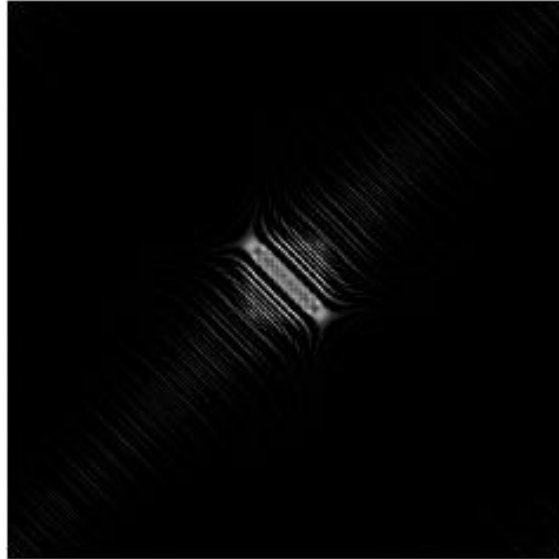


Figura 9.3: Distribución de Wigner asociada a la Transformada de la figura 9.2

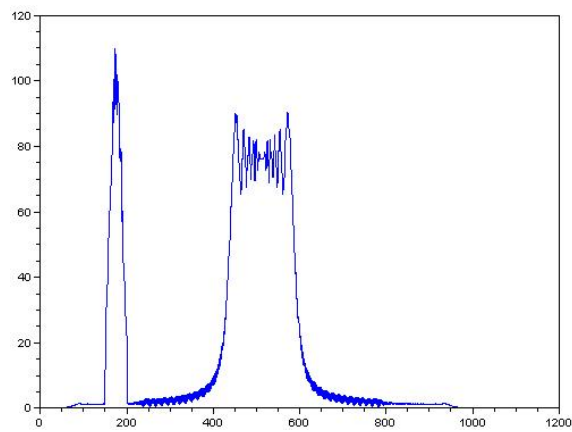


Figura 9.4: Valor absoluto de la Transformada de la figura 9.2 + ruido aditivo de una función Chirp*(Gauss+aleatorio)

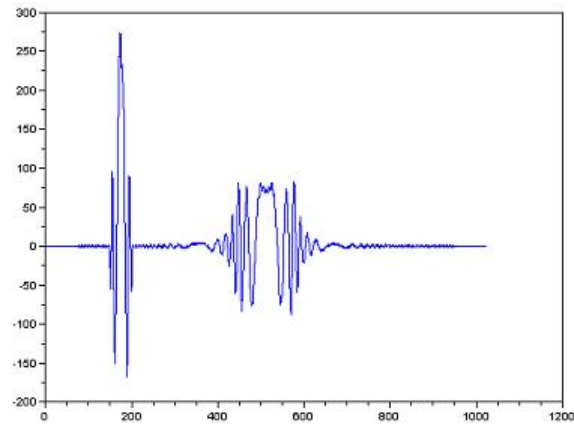


Figura 9.5: Valor real de Transformada de la figura 9.2 + ruido aditivo de una función Chirp*(Gauss+aleatorio)

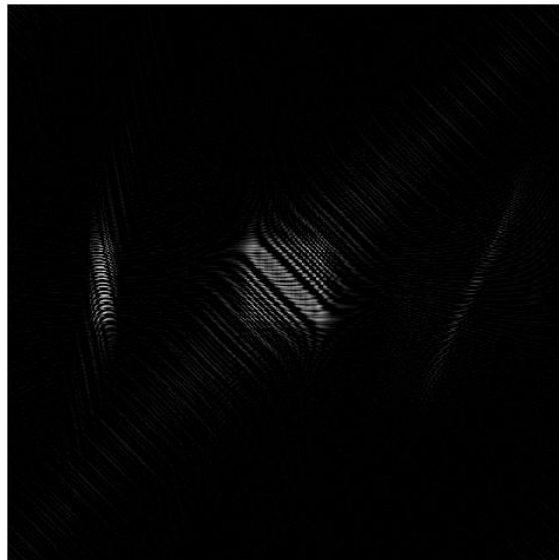


Figura 9.6: Distribución de Wigner de la señal en la figura 9.4

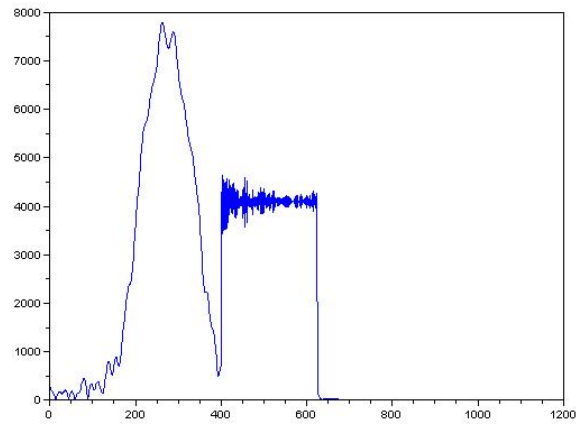


Figura 9.7: Ruido + señal en su dominio directo al implementar una $FrFT$ con $a = 0,5$



Figura 9.8: Distribución de Wigner de la señal en la figura 9.7

filtro adaptativo para la eliminación de ruido aleatorio inmerso en una señal con la siguiente expresión:

$$H^\alpha(v_\alpha) = \frac{1}{D^\alpha(v_\alpha)} \frac{S_D^\alpha(v_\alpha)}{S_D^\alpha(v_\alpha) + \frac{S_B^\alpha(v_\alpha)}{S_Y^\alpha(v_\alpha)}} \quad (9.8)$$

donde $D^\alpha = \mathfrak{F}_\alpha[D]$, este filtro debe ser implementando usando la convolución fraccionaria de la siguiente forma:

$$\hat{F}_y(x) = [X_y \overset{*}{\alpha} h](x) \quad (9.9)$$

Así, este tratamiento consigue un filtrado de menor error cuadrático medio, con relación al filtrado de Wiener estándar, debido a las rotaciones de la señal en la representación de la señal en la distribución de Wigner la cual permite adaptar el parámetro a en el proceso de filtrado.

Dado que el ángulo α permanece como un parámetro adaptativo, este mismo permite encontrar la solución del mínimo error de forma recursiva, lo que constituye una mejora significativa al tratamiento de Wiener estándar.

Capítulo 10

Demostración de hipótesis y Resultados

Durante todo el desarrollo de la presente investigación se tuvo en cuenta la Metodología *XP* y el cronograma de actividades presentado en el anteproyecto. Primero se realizó un estudio exhaustivo de la *FrFT* y las demás funciones que se relacionan con esta operación, como la Distribución de Wigner, la operación de convolución, la convolución fraccionaria, entre otras [22, 50, 52].

Posterior al análisis matemático se procedió a realizar cada una de las funciones expuestas en el capítulo 9 y modelar la interfaz gráfica para la interacción con el usuario, ver figura 10.1, en esta aplicación el usuario tiene la posibilidad de cargar diferentes funciones y de aplicar el algoritmo de la transformación de Fourier fraccionaria (*FrFT*), la Distribución de Wigner, graficar cada una de estas según sea el caso entre la función o la transformada, entre otras funcionalidades.

El objetivo de cargar las mismas señales con diferentes tamaños consiste en corroborar lo dicho en la ecuación 4.6 acerca del teorema de escalamiento, el cual dice que si una función es escalada entonces su transformación de Fourier frac-

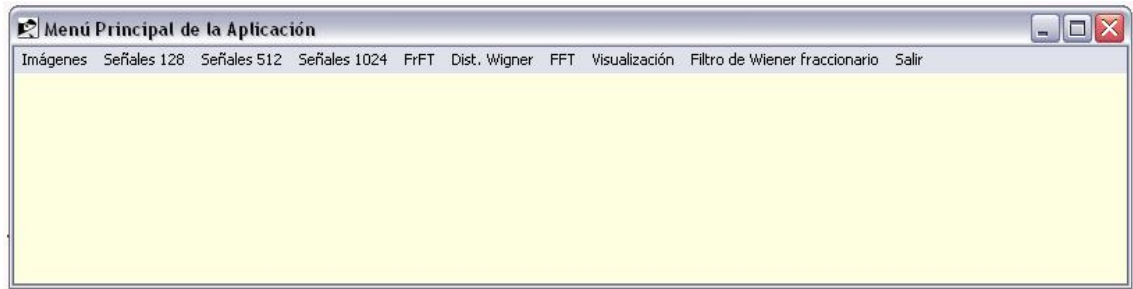


Figura 10.1: Menú Principal de la aplicación

cionaria será escalada también de manera inversa que la función y la $FrFT$ no será la misma ya que el ángulo con el cual esta es realizada estaría dado por la potencia fraccionaria $\beta = \arctan(c^2 \tan \alpha)$, y el ángulo estaría dado por $\beta = b\pi/2$. Algo similar ocurre con la Transformación de Fourier estándar con la diferencia que la segunda premisa es falsa y la FT de una función escalada será la misma Transformada escalada inversamente a la función, es decir, al aplicar un factor de escala m a una función, su FT se escala inversamente, ver figura 10.2. matemáticamente se describe esto de la siguiente manera:

$$f(x/m) \xrightarrow{\mathfrak{F}} |M|F(my).$$

Explicado esto en el lenguaje coloquial sería, al aumentar la escala de una función $f(x)$ entonces el espectro de frecuencia de su transformada $F(y)$ se reduce, es decir, el escalamiento es inverso o indirectamente proporcional.

En la figura 10.3 se puede observar que el operador entre más tiende a 0, se obtiene un espectro frecuencial muy parecido a la función de entrada, y de esta forma podemos tener la información real de la frecuencia de la señal durante todo su espacio o dominio directo de la misma.

Una de las propiedades y utilidades más importantes de la $FrFT$, es su representación como una rotación de la Distribución de Wigner de una función, en la

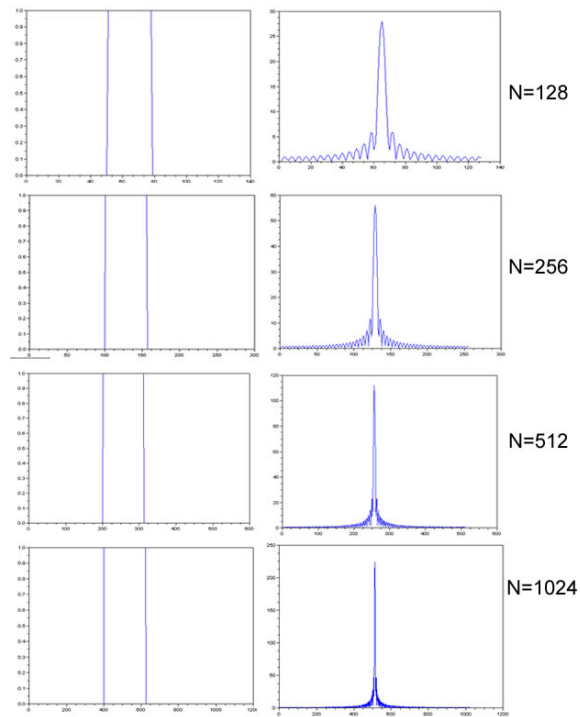


Figura 10.2: Función Rectángulo escalada y su Transformada de Fourier

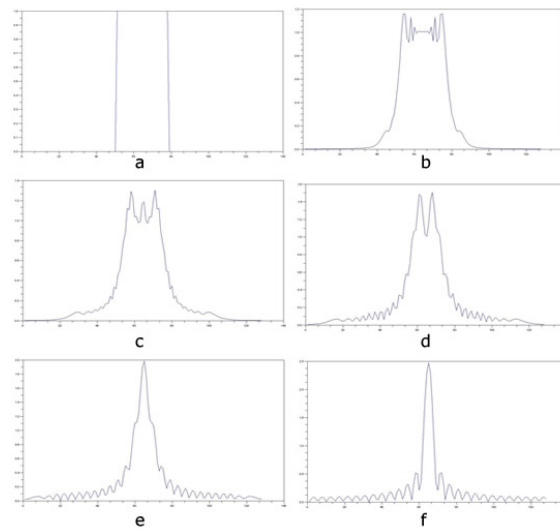


Figura 10.3: a) Función Rectángulo; b) $FrFT$ con $a=0.1$; c) $FrFT$ con $a=0.3$; d) $FrFT$ con $a=0.5$; e) $FrFT$ con $a=0.7$; f) $FrFT$ con $a=0.9$;

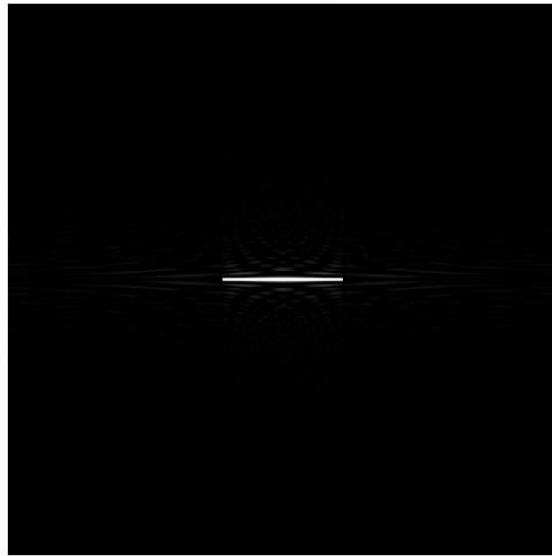


Figura 10.4: Distribución de Wigner de la señal rectángulo de 1024 muestras

figura 10.4 se puede observar la distribución de Wigner de una señal rectángulo de 1024 muestras, y en la figura 10.5 se muestran las rotaciones de esta función en la distribución de wigner aplicando la Transformación de Fourier fraccionaria, empezando desde 0.1 y terminando en 2 de derecha a izquierda y de arriba a abajo, siendo el ángulo de rotación $\alpha = a\pi/2$ y a como se ha venido diciendo durante todo este trabajo el operador fraccionario de la $FrFT$.

De la misma forma se implementa la $FrFT$ para el tratamiento imágenes realizando Transformaciones en las filas y columnas.

En la figura 10.9 se halló la transformada de Fourier estándar valiéndose de la propiedad de aditividad de la $FrFT$ y se le aplica una potencia fraccionaria de $a = 0,5$ a la señal de la figura 10.7, siendo esta última la transformada de 10.6.

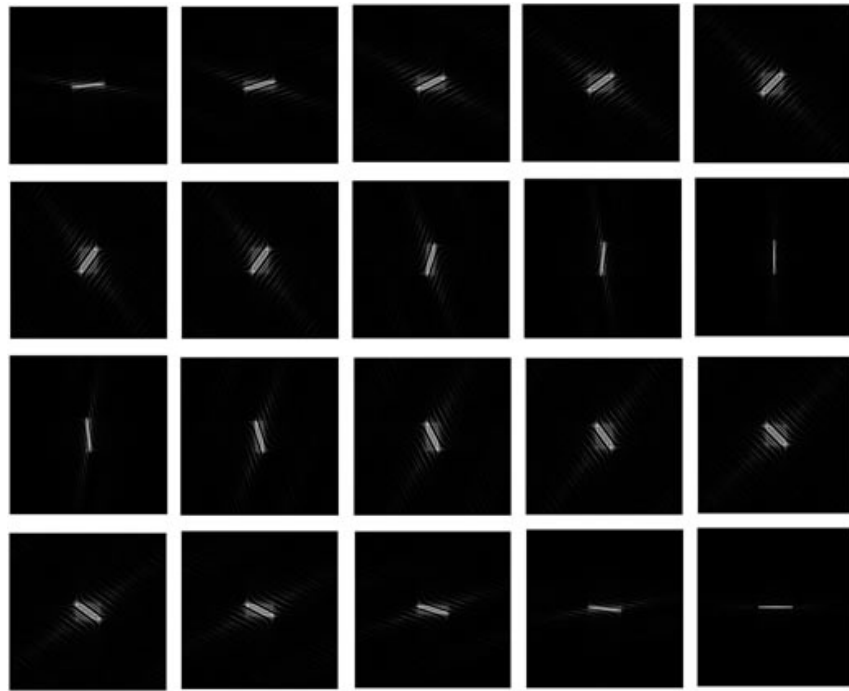


Figura 10.5: Rotación de Dis. Wigner por medio de la $FrFT$ de una función rectángulo de 1024 muestras

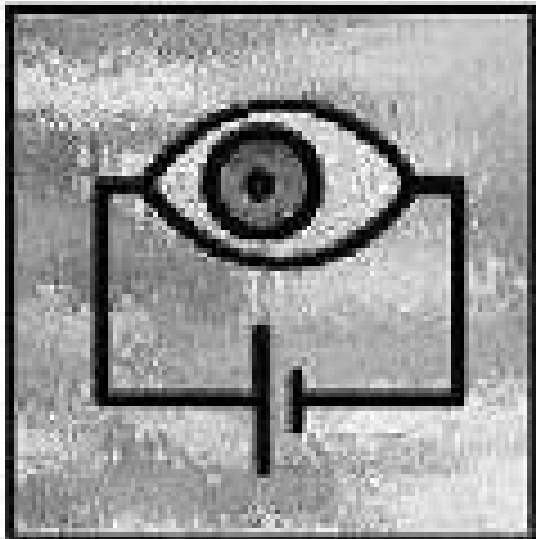


Figura 10.6: Imagen Tru.jpg de Scilab Image Processing

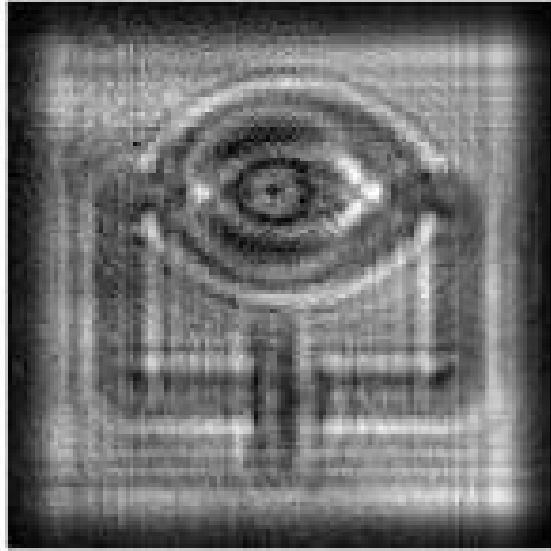


Figura 10.7: $FrFT$ con $a = 0,5$ en las filas y columnas

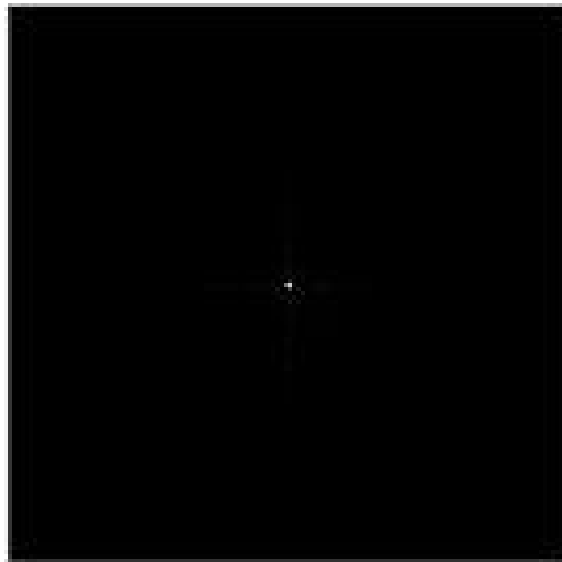


Figura 10.8: $FrFT$ de Tru.jpg con $a = 1 FT$

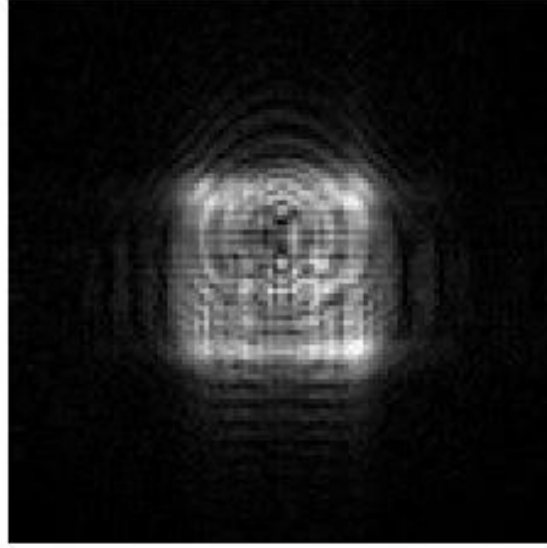


Figura 10.9: $FrFT$ de Tru.jpg con $a = 1,75 FT$



Figura 10.10: $FrFT$ de Tru.jpg con $a = 0,1$ en filas y $a = 0,9$ en las columnas

Capítulo 11

Limitaciones & inconvenientes

Con el fin de cumplir la atapa de análisis matemático de las ecuaciones que tenían relación con la Transformación de Fourier fraccionaria, se notaron deficiencias las cuales debieron ser sometidas a un riguroso estudio para desarrollar las diferentes funciones de la aplicación realizada en esta investigación.

La incorporación de un nuevo editor de texto por medio de macros y/o funciones \LaTeX [1], sugerido y requerido por el Director de la investigación, exhortando así el uso e implementación de aplicaciones de software libre, este fue creado por el matemático Leslie Lamport [2], por lo cual además de los programas utilizados para la aplicación en Scilab, se utilizó Mik \TeX y \TeX nicCenter, con estos se pueden generar documentos de diferentes clases de archivos tales como .Pdf, .Dvi, .Ps y .Html por medio de un “lenguaje de programación ”. - *I think \LaTeX is f^u*.

A pesar de implementarse diferentes soluciones para la Transformación de Fourier fraccionaria discreta, en los cuales aparece la expresión de la Transformación de Fourier la cual no fue posible implementar, debido a algunas incoherencias que se obtenían la fase de este algoritmo, ver figura 11.1. En algunas partes donde se menciona este algoritmo se le llama $FrFT0$.

Por tal razón para la solución del algoritmo se hace una operación matricial, entre una matriz $N \times N$ multiplicado por la función f_c de tamaño N , la expresión de una operación matricial se muestra como en la ecuación 11.1, haciendo una sumatoria por cada uno de los elementos de las filas de la matriz por los elementos del vector columna. En el algoritmo propuesto esta operación puede llegar a ser tediosa para el tratamiento de medianas y grandes imágenes, donde puede llegar a durar más de una decena minutos para hallar el resultado de una imagen de altas magnitudes, ver tabla 11.1

$$\begin{pmatrix} w_{11} & w_{12} & \dots & a_{1n} \\ w_{21} & w_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{k1} & w_{k2} & \dots & a_{kn} \end{pmatrix} \begin{pmatrix} f_{c1} \\ f_{c2} \\ \vdots \\ f_{cn} \end{pmatrix} \quad (11.1)$$

No. de muestras	dimensión	fracF	FrFT0	FrFT
128	1	0	0	0.094
256	1	0.16	0.023	0.312
512	1	0.16	0.017	1.052
1024	1	0.18	0.016	4.657
128	2	1.265	1.219	20.937
256	2	2.813	2.828	177.984
512	2	6.672	8.141	1259.312
1024	2	17.515	20.843	6167.481

Cuadro 11.1: Duración en segundos de transformaciones realizadas a señales unidimensionales y bidimensionales de diferentes tamaños

Algo similar ocurrió con la operación en integral de la convolución fraccionaria, mostrada en la ecuación 9.4, la cual fue resuelta en diferentes formas y se discretizaron todas estas funciones para la digitación de estos algoritmos en Scilab,

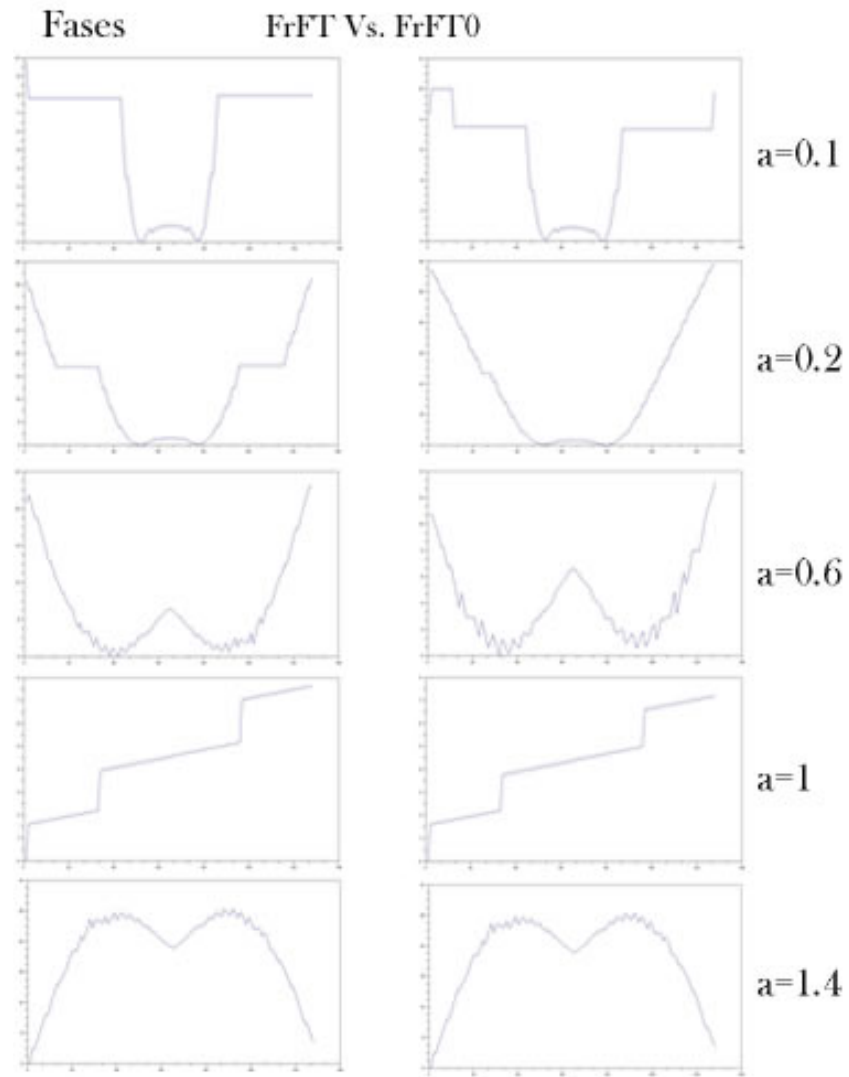


Figura 11.1: Valor de la fase de dos funciones de las *FrFT* implementadas

pero los resultados que esta arrojaba no eran los esperados para la operación de convolución fraccionaria.

Se observa en la figura 11.2 que ambos algoritmos que realizan la operación de convolución fraccionaria obtienen el mismo resultado en valores absolutos, pero al graficar sus fases estas no corresponden y son muy diferentes, lo cual no permite seguir adelante en la realización de la función del filtro propuesto en la sección 9.3.

Por tal motivo se implementa la ecuación 9.5 la cual resuelve la convolución fraccionaria por medio del teorema 1 utilizando la *FrFT*. Al hallar la transformación de Fourier fraccionaria discreta y utilizarla en el teorema de la convolución no hay dudas que obtendremos la convolución fraccionaria en un ángulo $\alpha = a\pi/2$. La convolución fraccionaria es la operación por medio de la cual debe ser implementado el filtro de Wiener fraccionario y sin esta, este no puede obtenerse, ello llevó a la demora de la realización del Filtro pero al final se cumplieron todos los objetivos propuestos en la presente investigación.

El tamaño de memoria demandado por Scilab es dinámico realizado mediante las funciones `pwd`, `m=pwd()`, y `m=getcwd()`; por esto el número de muestras de la señal de entrada aceptado por los parámetros iniciales en scilab en la *FrFT* es de 280 muestras y la función de entrada siempre debe ser una función par, por la discretización de la señal en el intervalo $(-N/2, N/2 - 1)$ 9.1, al realizar la petición de memoria por medio de la sentencia `stacksize(7e7)`; se aumenta el número de muestras que pueden ser analizadas por la *FrFT* a un tamaño de 1044 muestras, sin embargo; la aplicación de esta función a imágenes medianas y grandes puede ser algo tedioso, ver tabla 11.1 ya que podría llegar a demorar mucho tiempo para hallar la transformada de la imagen, por tal razón se acon-

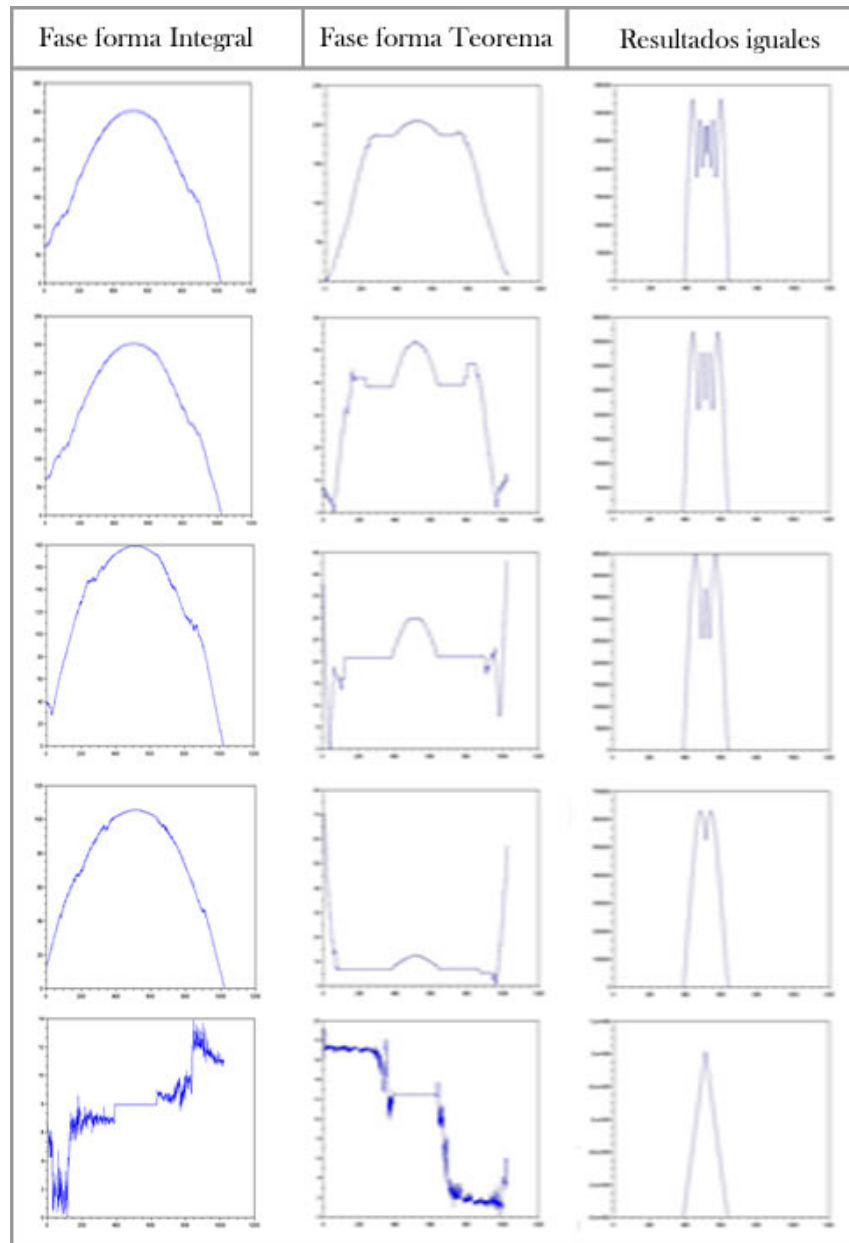


Figura 11.2: Valor de las fases de las funciones de convolución fraccionaria implementadas

seja trabajar con imágenes pequeñas de tamaño igual o menor a 124 muestras para ambas dimensiones para que su proceso no sea tedioso.

Capítulo 12

Conclusiones

El estudio de señales no estacionarias siempre ha representado mayor preocupación en el campo científico, ya que muchos factores en áreas de biofísica, electrofísica, geofísica, entre muchas otras se generan aleatoriamente a lo largo del tiempo, no obstante, poder hacer una rotación en el espacio tiempo-frecuencia, y utilizar esta información para el análisis, estudio y tratamiento de señales son muchas de las ventajas que tiene la Transformación de Fourier fraccionaria.

La dependencia del filtro con el orden fraccionario a lo convierte en un filtro adaptativo y por consiguiente lo hace ideal para la restauración de señales [46]. El ángulo α se adapta a la estadística de la señal, dado que el orden a permanece como un parámetro adaptativo lo que constituye una mejora significativa al filtro de Wiener estándar el cual se encuentra en este filtro cuando $\alpha = \pi/2$ [27].

Bibliografía

- [1] Oetiker T. Partl H. Hyna Irene. Schlegl E. The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$. Or $\text{\LaTeX} 2_{\epsilon}$ in 87 minutes. Version 3.2, Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA. September, 1998.
- [2] L. Lamport. **\LaTeX A Document Preparation System** Addison-Wesley, California 1986.
- [3] García, J. Procesamiento Digital de Señales. 2003. 493 pág.
- [4] Mitra, S. Digital Signal Processing: A computer-Based Approach. Editorial Mc. Graw-Hill. Tercera Edición. Año 2006.
- [5] Perov M. Yu. ; Yakimov A. V. The Influence of Fast Fourier Transform on the signal-spectrum Estimation. Radiophysics and Quantum Electronics. 2002, Vol. 45, No.3, Pág. 239-245.
- [6] Azar Y. Fiat A. Karlin A. McSherry F. Saia J. Spectral Analysis of Data. Symposium on Theory Of Computing STOC'0. Hersonissos, Grecia. Julio 2001. Pág.619-626.
- [7] Hlawatsch F. Gerald M. Time-Frecuency Signal Processing: A Statical Perspective*. Invited paper in Proc. CSSP-98, Mierlo (NL). Nov. 1998. Pág. 207-219.

- [8] Lara A. Sobre la Transformación Tiempo-Frecuencia y la aplicación del proceso de Convolución a la dinámica de sistemas físicos. Revista de acústica ISSN 0210-3680. Vol. 38. No. 1-2, Julio-Agost de 2006. Pág. 7-13.
- [9] Willett R. Sampling Theory and Spline Interpolation [Sitio Web]. Visitado el 24 de 2003 Disponible en: <http://cnx.org/content/m11126/2.3/>.
- [10] Brueller N., Peterfreund N., Porat M. Non-stationary signals: optimal sampling and instantaneous bandwidth estimation. Proc. de la IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, Pittsburgh, USA, 6-9 Oct 1998. Pág.113-115.
- [11] Bilinskis I., Mikelsons A. Randomized Signal Processing //Prentice-Hall, 1992. Citado por: Greitans M. Advanced Proccesing of Nonuniformly Sampled Non-Stationary Signals. Elektronika Ir Elektrotechnika. 2005. No. 3 Pág 42-45
- [12] Cekic Y. Akan A. Chaparro L. A Discrete Fractional Gabor Expansion for Time- Frequency Signal Analysis. Istanbul University engineering Faculty, Journal of Electrical 'amdpersan' Electronics. 2002. Vol. 2. No. 2. Pág. 483-487.
- [13] Greitans M. Advanced Proccesing of Nonuniformly Sampled Non-Stationary Signals. Elektronika Ir Elektrotechnika. 2005. No. 3 Pág 42-45.
- [14] Bouvrie J. Ezzat T. An Incremental Algorithm for Signal Reconstruction from Short-Time Fourier Transform Magnitude", pittsburg, USA. In INTERSPEECH-2006, paper 1691-Thu2BuP.9.
- [15] García R. Comprensión de Imágenes fijas en Matlab a través de DCT y Wavelet. [Tesis de Maestría]. Cholula, Puebla, México. Departamento de

Ingeniería Electrónica, escuela de Ingeniería. Universidad de las Américas Puebla. 2003. 101 Páginas.

- [16] Torres O. Torres Y. Pellat-finet P. Optica Fraccional de Fourier.
- [17] Alieva T., Bastiaans M. Calvo Ma. L. Fractional Transform in optical Information Processing. J. Applied. 2005. Vol 10: 1498-1519.
- [18] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Department of Computer Science, University of North Carolina at Chape Hill, Tech. Rep. NC 27599-3175, Apr. 2004.
- [19] Chin H. Iterative Wiener filters for image restoration. Sig. Proc IEEE. Vol. 39. Agosto de 1991 Pág. 1892 - 1899.
- [20] Ozaktas H. Barshan Billur. Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transform. Sociedad de Óptica de América. J. Opt. Soc. Am. A Vol. 11, No. 2/February 1994. Pág. 547-559
- [21] Namias V. The Fractional Order Fourier Transform and its Applications to Quantum Mechanics, J. Inst. Maths. Applica. 1980 Vol. 25: 241-265.
- [22] Almeida, L.B. The fractional Fourier transform and time-frequency representation. IEEE Trans. Sig. Proc. 1994. vol 42:3084-3091.
- [23] Alieva T., Bastiaans M., Calvo Ma. L. Fractional Transform in optical Information Processing. J. Applied. 2005. Vol 10: 1498-1519.
- [24] Ozaktas, H. M., Zalevsky Z., et al. The Fractional Fourier Transform Applications in Optics and Signal Processing, 2001. John Wiley 'andpersan' Sons.
- [25] Saxena, R.; Kulbir S. Fractional Fourier transform: A novel tool for signal processing. J. Indian Inst. Sci., ene-feb 2005, Vol 85: 11-26.

- [26] Wiener. N. Hermitian polynomials and Fourier analysis. J. Math. Phys. (MIT), 8, 70-73, (1929).
- [27] Tsai-Sheng K. Wiener Filter as an Optimal MMSE Interpolator. Taipei, Taiwan PWASET Vol. 24 Octubre de 2007 ISSN 1307-6884. 216-218.
- [28] Bultheel, A; Martínez H; Computation of the Fractional Fourier Transform; Dept. of Computer Science, Celestijnenlaan 200A, B-3001 Leuven.
- [29] Bailey, D; The Fractional Fourier Transform and applications; SIAM Review 1995. Vol. 33: Pág. 389-404
- [30] Scilab Group (INRIA Meta2 Project/ENPC Cergrene). Introduction to Scilab: User's Guide, Rocquencourt(Francia): Sitio Web <http://www-rocq.inria.fr/scilab>; 2007. 125 Pág.
- [31] Scilab Group. (INRIA Meta2 Project/ENPC Cergrene). Signal Processing With SciLab. Rocquencourt(Francia): Sitio Web: <http://www.scilab.org/product/index-product.php?page=old-documentation>; 2007. 205 Pág.
- [32] Cruel J. A Scilab Image Processing User Manual. Versión 0.3 (rev. 1). 2004. Armentieres-Francia. 30 Pág.
- [33] Cornejo J. Torres Y. Cálculo de la Transformada de Fourier fraccionaria sobre un DSP, Revista Colombiana de Física, Vol. 36. No. 1, 2004.
- [34] E. U. Condom. Immersion of the Fourier transform in a continuous group of functional transformations, Proc. Nat. Acad. Sc. USA, 23, 158-164, (1937).
- [35] Mustard D. Lie group imbeddings of the Fourier transform. School of Mathematics Preprint. A987. AM87/13, The university of New South Wales, Kensington, Australia.

- [36] Duffieux P.M. Lè Intègrale de Fourier et ses Applications a la Op-tique.Masson et Cie. 1970. euxième edición.
- [37] Pellat-Finet P. Lecciones de óptica de Fourier. División Editorial y de Publi-caciones Universidad Industrial de Santander (UIS). Bucaramanga. (2004).
- [38] Torres, P. Pellat-Finet y Y. M. Torres Moreno. Sampling theorem in fractional fourier domains. En «5th Iberoamerican Meeting on Optics and 8th Latin American Meeting on Optics, Lasers, and Their Applications», tomo 5622 de « Presented at the Society of Photo-Optical Instramentation Engineers (SPIE) Conference». Pág1188-1192. A. Marcano O.and J. L. Paz (oct 2004).
- [39] Z. Zalevsky and D. Mendlovic, "Fractional Wiener filter,". 1996. Appl. Opt. Vol. 35, Publicación 20. Pág. 3930-3936.
- [40] Scilab Group (INRIA Meta2 Project/ENPC Cergrene). Scilab Refer-ences Manual: Online Documentation. Rocquencourt(Francia): Sitio Web <http://www-rocq.inria.fr/scilab>; 2007. 649 pág.
- [41] Arikan O. Kutay A. Özaktas H. Akdemir. The Discrete Fractional fourier Transformation. 1996. TFTS. Pág. 205-207.
- [42] Djurovi I. Stankovic C. Pitas I. Digital watermarking in the fractional Fourier transformation domain, 2001. Journal of Network and Computer Applica-tions Vol. 24, Pág. 167-173.
- [43] A. Bultheel H. Martínez. A shattered survey of the Fractional Fourier Transform. Department of Computer Science, K.U.Leuven. Bélgica. Reporte TW337, Abril de 2002.
- [44] Yang et al. Improved fast fractional-Fourier-transform algorithm. Vol. 21, No. 9/Septiembre 2004/ J. Opt. Soc. Am. A Pág. 1677-1681

- [45] M.A. Kutay, fracF: Fast Computation of the Fractional Fourier Transform, 1996; <http://www.ee.bilkent.edu.tr/haldun/fracF.m>.
- [46] Ozaktas H. Kutay M. Bozdagi G. Digital computation of the fractional Fourier transform. IEEE Trans. Signal Process. 44 (1996) 2141-2150.
- [47] De la Escalrera, A. Visión por Computadora: Fundamentos y Métodos. Pearson Education, S.A. Madrid (España). 2001. 304 pág.
- [48] Bultheel, A. Marínez H. An introduction to the Fractional fourier Transform and friends. Preprint, February 25. Belgian.
- [49] Bultheel, Meyer Y., 1990: Ondelettes et operateurs. Paris, Hermann
- [50] Mustard, D. Fractional Convoluction. J. Austral. Math. Soc. Ser. B 40(1998). Pag. 257-265.
- [51] Torres, R. Tratamiento de señales por Transformación de Fourier fraccionaria. aplicaciones a la holografía sintética y al filtrado óptico. Agosto de 2008. Tesis Doctoral. Bucaramanga-Colombia. 134 pág.
- [52] Almeida, L. Product and convolution therems for the fractional fourier transform. IEEE Signal Process. Lett. 4, 15-17. 1997.