

03063



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

8

Unidad Académica de los Ciclos
Profesionales y de Posgrado

**DISEÑO E IMPLEMENTACIÓN DE
ALGORITMOS PARALELOS PARA
MÉTODOS DE ESTIMACIÓN ESPECTRAL**

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

ERNESTO RUBIO ACOSTA

DIRECTOR: DR. FABIÁN GARCÍA NOCETTI

MÉXICO, D.F.

NOVIEMBRE DE 2000

286717



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Unidad Académica de los Ciclos
Profesionales y de Posgrado

DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS PARALELOS PARA MÉTODOS DE ESTIMACIÓN ESPECTRAL

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

ERNESTO RUBIO ACOSTA

DIRECTOR: DR. FABIÁN GARCÍA NOCETTI

MÉXICO, D.F.

NOVIEMBRE DE 2000



UNAM
IIMAS



**DISEÑO E IMPLEMENTACIÓN DE
ALGORITMOS PARALELOS PARA
MÉTODOS DE ESTIMACIÓN
ESPECTRAL**

TESIS QUE PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS DE LA COMPUTACIÓN PRESENTA

ING. ERNESTO RUBIO ACOSTA

DIRECTOR: DR. FABIÁN GARCÍA NOCETTI
CODIRECTORES: DR. JULIO SOLANO GONZÁLEZ
DR. EDUARDO MORENO HERNÁNDEZ

México, D.F.

Noviembre de 2000

AGRADECIMIENTOS

Se agradece el apoyo brindado por el Departamento de Ingeniería de Sistemas Computacionales y Automatización, DISCA, del Instituto de Investigaciones en Matemáticas aplicada y en Sistemas, IIMAS, de la UNAM.

Se agradece el apoyo brindado por el Dr. Fabián García, director de esta tesis, así como por el Dr. Julio Solano y el Dr. Eduardo Moreno, codirectores de esta tesis.

Se agradece el apoyo brindado por los proyectos CONACYT (No. 27982-A) "Arquitecturas y algoritmos de alto desempeño en imagenología ultrasónica", CONACYT (No. 7350-858) "Red de Desarrollo e Investigación en Informática REDII, 1999", DGAPA-PAPIIT (No. IN106796) "Procesamiento paralelo de señales e imágenes en instrumentación médica" y DGAPA-PAPIIT (No. IN117999) "Arquitecturas y algoritmos de alto desempeño para la generación y procesamiento de imágenes acústicas".

DEDICATORIA

Dedico cariñosamente este trabajo
A mi esposa Laura
A mis padres Ernesto y Beatriz
A mis hermanas Beatriz y Patricia
Y
A mis sobrinos Miguel Ángel y Elizabeth

ÍNDICE

Portada		i
Agradecimientos		iii
Dedicatoria		iv
Índice		v
1	Introducción.....	1
1.1	Introducción	1
1.2	Planteamiento del problema	3
1.3	Hipótesis	4
1.4	Objetivos	5
1.5	Contribuciones	6
1.6	Descripción de contenido	7
1.7	Justificación	9
2	Generalidades.....	11
2.1	Introducción	11
2.2	Señal determinista y señal aleatoria	11
2.3	Señal estacionaria	12
2.4	Frecuencia	12
2.5	Señal no estacionaria	13
2.6	Señal analítica	14
2.7	Frecuencia instantánea	15
2.8	Frecuencia media pseudo instantánea	15
2.9	Expresiones discretizadas	16
3	Distribución Tiempo Frecuencia.....	19
3.1	Introducción	19
3.2	Caso continuo	19
3.3	Caso continua con ventana	20
3.4	Caso discreto con ventana	21
3.5	Análisis de la distribución	21
3.6	Algoritmo eficiente	23
3.7	Tabla de distribuciones	24
4	Distribución de Wigner Ville.....	27
4.1	Introducción	27
4.2	Caso continuo	27
4.3	Caso continuo con ventana	29
4.4	Caso discreto con ventana	29
4.5	Análisis de la distribución	30
4.6	Algoritmo eficiente	31
4.7	Experimentos numéricos en Matlab	34
4.8	Aplicación a una señal Dópler ultrasónica simulada en Matlab	43

4.9	Aplicación a una señal Dópler ultrasónica real en Matlab	48
5	Distribución de Choi Williams.....	49
5.1	Introducción	49
5.2	Caso continuo	51
5.3	Caso continuo con ventana	51
5.4	Caso discreto con ventana	52
5.5	Análisis de la distribución	55
5.6	Algoritmo eficiente	58
5.7	Experimentos numéricos en Matlab	67
5.8	Aplicación a una señal Dópler ultrasónica simulada en Matlab	73
5.9	Aplicación a una señal Dópler ultrasónica real en Matlab	75
6	Distribución de Bessel.....	75
6.1	Introducción	75
6.2	Caso continuo	77
6.3	Caso continuo con ventana	77
6.4	Caso discreto con ventana	78
6.5	Análisis de la distribución	81
6.6	Algoritmo eficiente	84
6.7	Experimentos numéricos en Matlab	93
6.8	Aplicación a una señal Dópler ultrasónica simulada en Matlab	99
6.9	Aplicación a una señal Dópler ultrasónica real en Matlab	101
7	Implantación de algoritmos paralelos para estimación espectral.....	101
7.1	Introducción	101
7.2	Objetivo del algoritmo TFD	101
7.3	Algoritmo secuencial TFD	105
7.4	Arquitectura del algoritmo paralelo TFD	108
7.5	Algoritmo FFT en base 2 de diezmado en frecuencia	110
7.6	Arquitectura del algoritmo paralelo FFT	111
7.7	Arquitectura del algoritmo paralelo para autocorrelación	112
7.8	Medidas de desempeño en procesamiento paralelo	113
7.9	Arquitectura del equipo de cómputo paralelo	115
7.10	Desempeños de los algoritmos paralelos	123
8	Análisis de resultados.....	123
8.1	Introducción	123
8.2	Número de operaciones que se realizan al calcular las distribuciones tiempo frecuencia	123
8.3	Selección de parámetros σ y α para las distribuciones De Choi Williams y de Bessel	124
8.4	Evaluación de las distribuciones tiempo frecuencia con base en el factor de peso de los términos de cruce	125
8.5	Experimentos numéricos en Matlab	129

8.6	Aplicación a una señal Dópler ultrasónica simulada en Matlab	132
8.7	Algoritmos paralelos para estimación espectral	139
9	Conclusiones.....	143
9.1	Introducción	143
9.2	Problemática	143
9.3	Hipótesis	143
9.4	Método para validar la hipótesis	144
9.5	Conclusiones	145
9.6	Trabajos de investigación futuros	150
1	Anexo: Programas secuenciales en Matlab.....	153
A1.1	Introducción	153
A1.2	Distribución de Wigner Ville	153
A1.3	Distribución de Choi Williams	156
A1.4	Distribución de Bessel	159
2	Anexo: Programas paralelos en Occam.....	163
A2.1	Introducción	163
A2.2	Distribución de Wigner Ville	163
A2.3	Distribución de Choi Williams	164
A2.4	Distribución de Bessel	165
A2.5	Programas comunes de procesos	167
A2.6	Programas comunes de configuración	176
A2.7	Diagramas de tiempo	185
3	Anexo: Señal tipo Dópler.....	189
A3.1	Introducción	189
A3.2	Señal Dópler simulada	189
4	Anexo: Otras expresiones para experimentos numéricos.....	193
A4.1	Introducción	193
A4.2	Ancho de banda espectral RMS	193
A4.3	Relación de potencia entre la señal y el ruido SNR	193
A4.4	Error en la estimación de la frecuencia media pseudo instantánea	193
A4.5	Error en la estimación del ancho de banda espectral RMS	194
5	Anexo: Expresiones matemáticas útiles.....	195
A5.1	Introducción	195
A5.2	Distribución de Wigner Ville	195
A5.3	Distribución de Choi Williams	195
A5.4	Distribución de Bessel	196
	Bibliografía.....	197

1 INTRODUCCIÓN

En este capítulo, después de dar una introducción a los métodos de estimación espectral, se describe la problemática que se desea resolver. Luego, se plantea la hipótesis que se desea verificar y los objetivos que se pretenden alcanzar. A continuación, se mencionan las principales contribuciones producto del desarrollo de este trabajo. Finalmente, se describe su contenido y justificación.

1.1 Introducción

En este apartado se da una introducción a los siguientes métodos de estimación espectral: transformada de Fourier, transformada corta de Fourier, distribución de Wigner Ville, distribución de Choi Williams y distribución de Bessel. Estas tres últimas pertenecen a la clase de Cohen de distribuciones tiempo frecuencia.

El cuadrado de la magnitud de la transformada de Fourier es el método clásico que se utiliza para representar la distribución de energía o espectro de una señal estacionaria en el dominio de la frecuencia.

El espectro de una señal $x(t)$ se define como:

$$S(f) = |X(f)|^2 \quad (\text{EQ.1.1})$$

donde $X(f)$ es la transformada de Fourier de la señal $x(t)$, la cual se define como:

$$X(f) = F[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (\text{EQ.1.2})$$

Claramente, esta representación en el dominio de la frecuencia se define sobre un intervalo infinito de tiempo (note los límites de integración con respecto al tiempo). Consecuentemente, para una señal no estacionaria, la variación de su espectro con respecto al tiempo se oculta.

Una representación alternativa consiste en la utilización de la transformada corta de Fourier (Short Time Fourier Transform), la cual se evalúa aplicando una función de ventaneo adecuada a la señal original y calculando la transformada de Fourier convencional del segmento de longitud finita resultante. De esta forma, la transformada corta de Fourier de la señal $x(t)$ se define como:

$$X_w(t, f) = \int_{t-T/2}^{t+T/2} x(\tau)W(\tau-t)e^{-j2\pi f\tau} d\tau \quad (\text{EQ.1.3})$$

donde $W(\tau)$ es una función de ventaneo que satisface la condición de $W(\tau) = 0$ para $|\tau| > T/2$. Utilizando esta técnica, se puede obtener una aproximación del espectro de la señal en el punto medio del intervalo de tiempo de la función de ventaneo a través de la siguiente expresión:

$$S_w(t, f) = |X_w(t, f)|^2 \quad (\text{EQ.1.4})$$

En este desarrollo se asume que la señal es estacionaria durante el corto intervalo de tiempo.

Nótese que las resoluciones temporal y frecuencial de la transformada corta de Fourier son inversamente proporcionales. Ambas resoluciones dependen de la longitud de ventana. Si se incrementa la longitud de ventana entonces se incrementa la resolución

frecuencial pero se disminuye la capacidad de ubicar los diferentes componentes frecuenciales en el tiempo y se debilita el supuesto de la señal estacionaria. Contrariamente, si se disminuye la longitud de ventana entonces se incrementa la capacidad de ubicar los diferentes componentes frecuenciales en el tiempo y se fortalece el supuesto de la señal estacionaria pero se disminuye la resolución frecuencial.

Existen otros métodos de estimación espectral, tales como los métodos autorregresivos, los métodos de promedios móviles y los métodos de varianza mínima. Sin embargo, el desempeño de algunos de estos métodos también depende del uso de una señal estacionaria; por ejemplo, en los métodos autorregresivos se utiliza un intervalo corto de tiempo de una señal no estacionaria, de modo que ésta última se pueda considerar como una estacionaria [52][53][54][55][56][57].

Durante los últimos años se han investigado otros métodos de representación temporal y frecuencial alternativos. Uno de estos métodos es la distribución de Wigner Ville [6], el cual es fundamental para representar la distribución de energía de la señal en el dominio del tiempo y la frecuencia. La distribución de Wigner Ville se define como:

$$WVD(t, \omega) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j\omega\tau} d\tau \quad (\text{EQ.1.5})$$

Las propiedades superiores de la distribución de Wigner Ville con respecto a la transformada corta de Fourier para obtener la distribución de energía o espectro, la hacen muy útil en el procesamiento de señales no estacionarias. El análisis de esta distribución se realiza en el Capítulo 4, Distribución de Wigner Ville.

Sin embargo, esta última distribución adolece de un problema. Si se calcula la distribución de Wigner Ville de la suma de dos señales sinusoidales con amplitud, frecuencia y fase constantes, como la que se enuncia a continuación:

$$x(t) = A_1 \cos(\omega_1 t + \theta_1) + A_2 \cos(\omega_2 t + \theta_2) \quad (\text{EQ.1.6})$$

se obtiene que:

$$WVD(t, \omega) = 2\pi A_1^2 \delta(\omega - \omega_1) + 2\pi A_2^2 \delta(\omega - \omega_2) + 2\pi A_1 A_2 \cos((\omega_1 - \omega_2)t + (\theta_1 + \theta_2)) \delta\left(\omega - \frac{\omega_1 + \omega_2}{2}\right) \quad (\text{EQ.1.7})$$

donde los dos primeros términos representan el verdadero espectro de la señal, mientras que el último término representa una contribución no deseada al espectro debida a la interacción de las dos señales sinusoidales. Este último término oculta el verdadero espectro de la señal. Este problema motiva el planteamiento de otras distribuciones que minimicen esas contribuciones no deseadas. Se observa que todas esas distribuciones tienen una estructura en común: la Clase de Cohen de distribuciones tiempo frecuencia.

La distribución de Wigner Ville se puede considerar como un miembro de la clase de Cohen de distribuciones tiempo frecuencia [1]. Esta clase de distribuciones utiliza una transformación bilineal que depende de dos variables: el tiempo y la frecuencia. La clase de Cohen de distribuciones tiempo frecuencia se define como:

$$TFD(t, \omega, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta(t-\mu)} d\theta \right] x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \left] e^{-j\omega\tau} d\tau \quad (\text{EQ.1.8})$$

donde $\phi(\theta, \tau)$ es una función llamada Kernel de la distribución. Las propiedades de cada distribución quedan definidas por su kernel. El análisis de esta clase se realiza en el capítulo 3, Distribuciones Tiempo Frecuencia. En particular, el kernel de la distribución de Wigner Ville es:

$$\phi(\theta, \tau) = 1 \quad (\text{EQ.1.9})$$

En el caso de las señales que se pueden expandir en una sumatoria de componentes sinusoidales o monocomentes, la estructura bilineal inherente de la clase de Cohen provoca términos de interferencia no deseables debidos a la interacción entre los diferentes monocomentes de la señal. Estos términos de interferencia se llaman términos de cruce. Estos términos de cruce podrían llegar a ser problemáticos ya que ocultan aquellos términos que representan el espectro verdadero de la señal, conocidos como auto-términos. La cantidad y la forma de la interferencia se relaciona directamente con el kernel de la distribución. Un kernel deseable para una distribución es aquel que minimiza los efectos de la interferencia de los términos de cruce.

En este sentido, la distribución de Choi Williams ^[16], miembro de la clase de Cohen de distribuciones tiempo frecuencia, con kernel:

$$\phi(\theta, t) = e^{-\frac{t^2 \theta^2}{\sigma}} \quad (\text{EQ.1.10})$$

la cual se define como:

$$CWD(t, \omega) = \int_{-\infty}^{+\infty} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.1.11})$$

minimiza y distribuye de una forma más conveniente los términos de cruce que la distribución de Wigner Ville. El análisis de esta distribución se realiza en el Capítulo 5, Distribución de Choi Williams. De igual forma lo hace la distribución de Bessel ^[18], también miembro de la clase de Cohen de distribuciones tiempo frecuencia, con kernel:

$$\phi(\theta, \tau) = \frac{2J_1(\alpha\tau\theta)}{\alpha\tau\theta} \quad (\text{EQ.1.12})$$

la cual se define como:

$$BD(t, f) = \int_{-\infty}^{+\infty} \frac{2}{\pi\alpha|\tau|} \int_{-\infty}^{+\infty} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f\tau} d\tau \quad (\text{EQ.1.13})$$

El análisis de esta distribución se realiza en el Capítulo 6, Distribución de Bessel.

El parámetro α del kernel de la distribución de Bessel permite que ésta se adapte con más versatilidad que la distribución de Choi Williams, cuyo parámetro es σ , a diferentes aplicaciones de procesamiento de señales no estacionarias.

1.2 Planteamiento del problema

En términos generales, se requiere calcular la distribución de energía o espectro de una señal no estacionaria.

El contexto físico del problema es el siguiente. Se requiere estimar el grado de obstrucción de una arteria o vena del cuerpo humano. Entre otros factores, la obstrucción se

debe a la acumulación de colesterol en la pared de la arteria o vena. Esta obstrucción causa un flujo sanguíneo turbulento. El grado de turbulencia se puede medir indirectamente a través de un método no invasivo, el cual se describe a continuación. Primero, se hace incidir una onda ultrasónica sobre la arteria o vena. Esta onda ultrasónica se refleja debido al movimiento aleatorio de los diferentes glóbulos que componen el tejido sanguíneo (backscattering of ultrasound). Segundo, se sensa esta onda reflejada, la cual es una señal no estacionaria, ultrasónica de tipo Dópler. Tercero, se calcula el espectro de esta señal. Finalmente, con base en este espectro, se calcula la frecuencia media pseudo instantánea de la señal. La forma de la gráfica de la frecuencia media pseudo instantánea de la señal ultrasónica de tipo Dópler con respecto al tiempo constituye un indicador del grado de obstrucción de una arteria o vena. Determinados modelos de gráficas se asocian con patologías específicas debidas a la obstrucción de arterias o venas en un mayor o menor grado ^{[24][58]}.

Como ya se indicó, la clase de Cohen de distribuciones tiempo frecuencia es una herramienta más poderosa para calcular el espectro que la transformada corta de Fourier. Sin embargo, la complejidad del cálculo de una distribución de la clase de Cohen es mucho mayor que aquella de la transformada corta de Fourier. Por otro lado, el procesamiento de las señales biomédicas, como las señales ultrasónicas tipo Dópler, se realiza en sistemas digitales bajo las restricciones de los procesos en tiempo real.

Entonces, parte del problema consiste en que, para calcular el espectro de una señal no estacionaria a través de una distribución de la clase de Cohen, se debe realizar una gran cantidad de operaciones matemáticas en un muy corto periodo de tiempo. De aquí que sea necesario, primero, diseñar algoritmos numéricos que calculen las distribuciones de la Clase de Cohen en los cuales se minimice el número de operaciones involucradas; y segundo, implantar estos algoritmos numéricos de tal forma que se aproveche las capacidades de cómputo en paralelo de los sistemas digitales. El primer problema se analiza en los capítulos 4, 5 y 6, Distribuciones de Wigner Ville, Choi Williams y Bessel respectivamente. El segundo problema se analiza en el capítulo 7, Implantación de Algoritmos Paralelos para Estimación Espectral.

Otra parte del problema consiste en seleccionar los parámetros adecuados de los kernels de las diferentes distribuciones de la clase de Cohen según las características particulares de las señales ultrasónicas de tipo Dópler que se pretenda analizar, por ejemplo, ancho de banda espectral RMS, frecuencia media pseudo instantánea ^{[6][10][11]} y presencia de ruido. Este problema se analiza en los capítulos 4, 5 y 6, Distribuciones de Wigner Ville, Choi Williams y Bessel respectivamente

1.3 Hipótesis

Primero, la distribución tiempo frecuencia de Bessel estima mejor el espectro de una señal no estacionaria de tipo Dópler ultrasónica que la distribución de Choi Williams, tomando como referencia la precisión en la estimación de su frecuencia media pseudo instantánea y ancho de banda espectral RMS. De igual forma, la distribución de Choi Williams hace una mejor estimación que la distribución de Wigner Ville. Se considera como factores relevantes que afectan la precisión de los cálculos, la relación de potencias de la señal con respecto a la del ruido, así como la longitud de la ventana de muestreo.

Segundo, es posible calcular las distribuciones de Wigner Ville, de Choi Williams y de Bessel en una arquitectura de cómputo paralelo basada en 4 procesadores Transputers T805 de 30 MHz bajo las restricciones de la programación en tiempo real.

La señal ultrasónica tipo Dópler utilizada para realizar los diferentes experimentos se describe en el Anexo A3, Señal Tipo Dópler.

1.4 Objetivos

El objetivo general de este trabajo es desarrollar e implantar algoritmos paralelos para técnicas de estimación espectral, en particular, para aquellas técnicas de estimación espectral basadas en distribuciones tiempo frecuencia, concretamente, de Wigner Ville, de Choi Williams y de Bessel.

Los objetivos específicos de este trabajo son:

- Diseño de algoritmos secuenciales para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, en los cuales se efectúen un menor número de operaciones que las que se efectuarían al calcular estas distribuciones a través de su definición.
- Implantación de los algoritmos secuenciales para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, en una herramienta de cálculo computacional de alto desempeño como Matlab.
- Evaluación práctica de los efectos que resultan de la variación en la longitud de ventana de muestreo de la señal y en la relación entre las potencias de la señal y del ruido (SNR), sobre la medición de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, utilizando las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, sobre un modelo de señal Dópler ultrasónica.
- Evaluación práctica de los efectos que resultan de la variación de los parámetros σ y α de las distribuciones tiempo frecuencia de Choi Williams y de Bessel, respectivamente, sobre la medición de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, utilizando un modelo de señal Dópler ultrasónica.
- Diseño de algoritmos paralelos para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel.
- Implantación de los algoritmos paralelos para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, en lenguaje Occam, para una arquitectura de equipo de cómputo paralelo basada en procesadores Transputer.
- Evaluación práctica del desempeño de los algoritmos paralelos para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel,

implantados en distintas arquitecturas de equipo de cómputo paralelo (topologías) basada en procesadores Transputer.

- Establecimiento de un criterio de comparación entre las diferentes distribuciones tiempo frecuencia, con base en el factor de peso de sus términos de cruce

1.5 Contribuciones

Las contribuciones fundamentales de este trabajo son las siguientes. Primero, la obtención de las expresiones simplificadas para calcular las distribuciones tiempo frecuencia discretas de Wigner Ville, de Choi Williams y de Bessel. Refiérase a las secciones 4.6, 5.6 y 6.6, Algoritmo eficiente, de los capítulos 4, 5 y 6, Distribuciones de Wigner Ville, de Choi Williams y de Bessel, respectivamente.

Segundo, la obtención de los parámetros óptimos σ y α de la distribución de Choi Williams y de Bessel respectivamente, propios para la señal Dópler ultrasónica en estudio. Refiérase a las secciones 5.8.8 y 6.8.8, Variación del parámetro σ y Variación del parámetro α , de los capítulos 5 y 6, Distribuciones de Choi Williams y de Bessel, respectivamente.

Tercero, el establecimiento de un criterio de comparación entre las diferentes distribuciones tiempo frecuencia, con base en el factor de peso de sus términos de cruce. Refiérase a la sección 8.3, Evaluación de las distribuciones tiempo frecuencia con base en el factor de peso de los términos de cruce, del capítulo 8, Análisis de resultados.

Cuarto, la propuesta de una arquitectura general del algoritmo paralelo que calcula las distribuciones tiempo frecuencia, explotando tanto el paralelismo de datos como el paralelismo algorítmico. Refiérase al capítulo 7, Implantación de algoritmos paralelos para estimación espectral.

Finalmente, los artículos que se han derivado de este trabajo son:

- F. García, E. Moreno, E. Rubio, I. Sánchez. "Diseño de un Algoritmo Paralelo para la Estimación Espectral de Señales Dópler Ultrasónicas usando la Distribución Tiempo Frecuencia de Bessel". Memorias SOMI XIV Congreso de Instrumentación, p.p. 518-522. Tonantzintla, Puebla. Octubre 1999.

Resumen: En este trabajo se presenta una expresión simplificada y un algoritmo paralelo para el cálculo de la distribución tiempo frecuencia de Bessel discreta, así como la medición del tiempo de ejecución del algoritmo bajo diferentes configuraciones de hardware. El algoritmo se implanta en una arquitectura de cómputo paralela basada en procesadores Transputer y la señal procesada es de tipo Dópler ultrasónica. Adicionalmente se da una breve descripción de la familia de Cohen de distribuciones tiempo frecuencia y un breve planteamiento de la distribución tiempo frecuencia de Bessel continua

- F. García, J. Solano, E. Rubio and E. Moreno. "Towards the simplified computation of time-frequency distributions for signal analysis". Proceedings of the 6th IFAC Workshop

on Algorithms and Architectures for Real-Time Control, p.p. 161-166. Mallorca, España. Mayor de 2000.

Abstract: Conventional methods for signal analysis utilise the Fourier Transform to estimate the spectral response of a signal. However this current practice suffers from poor frequency resolution when estimating non-stationary signals. This paper presents some alternative methods based on time-frequency distributions such as Wigner Ville, Choi Williams, Bessel and Born Jordan from a Cohen's class point of view. For each case, a continuous and discrete distribution is formulated, a criteria for determining the interaction between the spectral components of the signal is given and the simplified discretised expression for the calculation of the distribution is proposed that can produce a reduction of at least half of the computations realised when using the original time-frequency distribution definition. A general parallel architecture for the parallel computation of the distribution is also proposed.

- F. García, J. Solano, E. Rubio and E. Moreno "Simplified Computation of Time-Frequency Distributions for Doppler Ultrasonic Signal Analysis". II Congreso Iberoamericano de Acústica. Madrid, España. Octubre de 2000 (Aceptado).

1.6 Descripción del contenido

En este capítulo, se enuncia el problema y su contexto que motiva la elaboración de éste trabajo, así como la hipótesis, los objetivos y las contribuciones resultantes de la elaboración de este trabajo. Adicionalmente, se da una introducción a los métodos de estimación espectral y se describe el contenido de cada capítulo.

En el capítulo 2, se enuncian las definiciones de señal analítica, frecuencia, frecuencia instantánea y frecuencia media pseudo instantánea, entre otros conceptos. Además, se plantean expresiones en variable continua y discreta para su cálculo.

En el capítulo 3, se plantea la clase de Cohen distribuciones tiempo frecuencia y se enuncian algunas de las distribuciones que pertenecen a esta clase. Adicionalmente, se propone un procedimiento general para discretizar las distribuciones continuas, para analizar la influencia del kernel de las distribuciones en la disminución de los efectos de interferencia entre los diferentes componentes de una señal y para simplificar la expresión de su definición de modo que se realicen a lo más la mitad de las operaciones.

En el capítulo 4, se plantea la distribución tiempo frecuencia continua de Wigner Ville a partir de la clase de distribuciones tiempo frecuencia de Cohen. Adicionalmente, se plantea su discretización y se propone un algoritmo secuencial simplificado que la calcule. Finalmente, se presenta una evaluación práctica realizada en Matlab del efecto de la longitud de ventana de muestreo y de la relación de potencias de la señal y del ruido en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS sobre una señal Dópler ultrasónica.

En el capítulo 5, se plantea la distribución tiempo frecuencia continua de Choi Williams a partir de la clase de distribuciones tiempo frecuencia de Cohen. Adicionalmente se plantea su discretización y se propone un algoritmo secuencial simplificado que la calcule. Finalmente, se presenta una evaluación práctica realizada en Matlab del efecto de la longitud de ventana de

muestreo y de la relación de potencias de la señal y del ruido en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS sobre una señal Dópler ultrasónica. También, se determina su parámetro σ óptimo para la aplicación mencionada.

En el capítulo 6, se plantea la distribución tiempo frecuencia continua de Bessel a partir de la clase de distribuciones tiempo frecuencia de Cohen. Adicionalmente se plantea su discretización y se propone un algoritmo secuencial simplificado que la calcule. Finalmente, se presenta una evaluación práctica realizada en Matlab del efecto de la longitud de ventana de muestreo y de la relación de potencias de la señal y del ruido en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS sobre una señal Dópler ultrasónica. También, se determina su parámetro α óptimo para la aplicación mencionada.

En el capítulo 7, se propone el diseño y la implantación en Occam de los algoritmos paralelos para el cálculo de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel. Se presenta una evaluación del desempeño de los programas paralelos propuestos, implantados en diferentes arquitecturas de equipo de cómputo paralelo basada en procesadores Transputer, en las cuales se explota el paralelismo de datos y el paralelismo algorítmico.

En el capítulo 8, se discuten los resultados obtenidos en los capítulos anteriores. Estos resultados se analizan para cada distribución y, también, se comparan entre ellos. Además, se presenta un criterio de comparación entre las diferentes distribuciones tiempo frecuencia con base en sus términos de cruce.

En el capítulo 9, después de plantear brevemente una introducción, se enuncia la hipótesis y el método que se utilizó para verificarla. A continuación, se presentan las conclusiones obtenidas de este trabajo. Finalmente, se sugieren trabajos de investigación futuros por realizar.

En el Anexo 1, se listan los códigos de los programas secuenciales en lenguaje propietario de Matlab de cada una de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel.

En el Anexo 2, se listan los códigos de los programas paralelos en lenguaje Occam de cada una de las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, así como sus diagramas de tiempo o de intercambio de mensajes.

En el Anexo 3, se describe la construcción de la señal tipo Dópler ultrasónica que se utiliza en la evaluación de los diferentes algoritmos de estimación espectral.

En el Anexo 4, se plantean expresiones útiles para los experimentos numéricos en Matlab, como son la que calcula el ancho de banda espectral RMS, la relación de potencia entre la señal y el ruido (SNR) y los errores en las diferentes estimaciones.

En el Anexo 5, se detallan secuencias relevantes de los desarrollos matemáticos que se efectúan en los capítulos correspondientes a las distribuciones de Wigner Ville, de Choi Williams y de Bessel.

1.6.1 Nomenclatura

La nomenclatura que se utilizó en este trabajo es la siguiente:

[n]	Referencia bibliográfica número n. Refiérase a la Bibliografía.
(A.c.n)	Referencia matemática número n del capítulo número c. Refiérase al Anexo 5.
(EQ.c.n)	Ecuación número n del capítulo número c.
FIG.c.n	Figura número n del capítulo número c.
TB.c.n	Tabla número n del capítulo número c.

1.7 Justificación

Esta tesis se llevó a cabo como parte de las actividades de investigación y desarrollo en el área de Arquitecturas y Algoritmos de Alto Desempeño para el Procesamiento de Señales e Imágenes, que se realiza en el Departamento de Ingeniería de Sistemas Computacionales y Automatización del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la UNAM, bajo la supervisión del Dr. F. García, Dr. J. Solano y Dr. E. Moreno.

2 GENERALIDADES

2.1 Introducción

En este capítulo se definen los conceptos de señal analítica, frecuencia instantánea y frecuencia media pseudo instantánea y se plantean métodos para su cálculo, tanto para el caso continuo como para el discreto. Adicionalmente se definen otros conceptos asociados como son los de señal determinista, aleatoria, estacionaria, y no estacionaria, así como el de frecuencia. También se presentan ejemplos de los distintos tipos de señales. La bibliografía básica de este capítulo consiste en [6] [7] y [25]; estudios con señales tipo Dópler ultrasónicas se encuentran en [10], [11] y [12].

Las expresiones importantes de este capítulo son:

- Señal analítica.

$$x_a(t) = F^{-1}\{X_a(f)\} \text{ donde } X_a(f) = \begin{cases} 2X(f) & f > 0 \\ X(f) & f = 0 \\ 0 & f < 0 \end{cases} \text{ y } X(f) = F[x(t)]$$

- Frecuencia instantánea

$$f_i(t) = \frac{1}{2\pi} \frac{d}{dt} [\phi(t)] \text{ donde } \phi(t) = \text{fase}[x_a(t)]$$

- Frecuencia media pseudo instantánea.

$$PIMF(t) = \frac{\int_0^{\infty} f \cdot PIPD(t, f) df}{\int_0^{\infty} PIPD(t, f) df} \text{ donde } PIPD(t, f) = \begin{cases} TFD(t, f) & TFD(t, f) \geq 0 \\ 0 & TFD(t, f) < 0 \end{cases} \text{ donde, a su vez,}$$

$TFD(t, f)$ es la distribución tiempo frecuencia de $x(t)$.

2.2 Señal determinista y señal aleatoria

En este apartado se definirá el concepto de señal, señal determinista y señal aleatoria^{[13][15]}. Una señal se define como una cantidad física que varía con el tiempo, el espacio o cualquier otra variable o variables independientes.

El procesamiento de señales requiere que la señal sea descrita matemáticamente. Entonces, cualquier señal que pueda ser definida por una forma matemática explícita o un conjunto de datos explícito, se denomina señal determinista.

Sin embargo, no todas las señales se pueden definir de manera práctica con suficiente precisión a través de una forma matemática explícita. En este caso, se asume que la evolución de la señal con respecto al tiempo, espacio o cualquier otra variable o variables independientes, es impredecible y, consecuentemente, se denomina señal aleatoria. El marco teórico para analizar las señales aleatorias lo constituye la teoría de la probabilidad y los procesos estocásticos^[14]. De este modo, las señales aleatorias se describen mediante técnicas estadísticas.

2.3 Señal estacionaria

En este apartado se definirá señal estacionaria y se presentará algunos ejemplos^{[13][15]}. Una señal estacionaria es aquella cuyas características espectrales son constantes con el transcurrir del tiempo. Las características espectrales se relacionan con la descripción de la señal en el dominio de la frecuencia. Una señal estacionaria es una señal determinista.

Un primer ejemplo de señal estacionaria es una señal sinusoidal:

$$x(t) = A_0 \cos(2\pi f_0 t + \theta_0) \quad (\text{EQ.2.1})$$

donde A_0 es la amplitud, f_0 es la frecuencia y θ_0 es la fase de la señal; todos estos parámetros son constantes con respecto al tiempo.

Un segundo ejemplo de señal estacionaria es una señal compuesta por la suma de un conjunto finito de señales sinusoidales:

$$x(t) = \sum_{i=1}^N A_i \cos(2\pi f_i t + \theta_i) \quad (\text{EQ.2.2})$$

donde A_i es la amplitud, f_i es la frecuencia y θ_i es la fase de la componente i -ésima de la señal; todos estos parámetros son constantes con respecto al tiempo.

Un tercer ejemplo de señal estacionaria es una señal compuesta por la suma de un conjunto infinito de señales sinusoidales:

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{j2\pi f t} df \quad (\text{EQ.2.3})$$

donde $X(f)$ es la Transformada de Fourier de la señal $x(t)$ y constituye el espectro de frecuencias de la señal $x(t)$; $X(f)$ depende solamente de la frecuencia y por lo tanto es constante con respecto al tiempo. La Transformada de Fourier de la señal $x(t)$ se define como:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt \quad (\text{EQ.2.4})$$

2.4 Frecuencia

En este apartado se definirá el concepto de frecuencia^{[6][13][15]}. En mecánica, la frecuencia de un movimiento vibratorio se define como el número de oscilaciones por unidad de tiempo, donde el movimiento vibratorio es cualquier movimiento de vaivén y una oscilación es un movimiento completo de vaivén. Esto es, una oscilación es un movimiento que parte de la posición de equilibrio en dirección hacia uno de los extremos, luego hacia el otro de los extremos y, finalmente, hacia la posición de equilibrio.

Un tipo especial de movimiento vibratorio es el movimiento armónico simple, en el cual la aceleración $s''(t)$ es proporcional del desplazamiento $s(t)$ y siempre dirigida hacia la posición de equilibrio. De esta forma, el movimiento armónico simple queda descrito por la ecuación diferencial:

$$s''(t) = -\omega_0^2 s(t) \quad (\text{EQ.2.5})$$

cuya solución general es:

$$s(t) = c_1 e^{j\omega_0 t} + c_2 e^{-j\omega_0 t} \quad (\text{EQ.2.6})$$

o bien, una solución particular:

$$s(t) = A_0 \cos(\omega_0 t + \theta_0) \quad (\text{EQ.2.7})$$

donde ω_0 es la frecuencia del movimiento armónico simple. Nótese la similitud de esta última ecuación con (EQ.2.1).

Muchos problemas involucran ondas que se propagan en materiales, en los cuales el movimiento de una partícula en un punto fijo puede describirse con un movimiento armónico simple. Para estos casos, la frecuencia de una onda en movimiento se define como el número de ondas que pasan a través de un punto fijo por unidad de tiempo.

2.5 Señal no estacionaria

En este apartado se definirá señal no estacionaria, se presentará algunos ejemplos y se interpretará físicamente el concepto de frecuencia instantánea^{[6][13][15]}. Una señal no estacionaria es aquella cuyas características espectrales varían con el tiempo. Una señal no estacionaria puede ser una señal determinista o una señal aleatoria.

Un ejemplo de señal no estacionaria determinista es la señal Chirp, la cual se puede describir burdamente como una onda sinusoidal cuya frecuencia se incrementa proporcionalmente con el transcurrir del tiempo. En la figura FIG.2.1 se muestra una gráfica de esta señal. La expresión que la describe es:

$$x(t) = A_0 \cos(2\pi(m_0 t + b_0)t + \theta_0) \quad (\text{EQ.2.8})$$

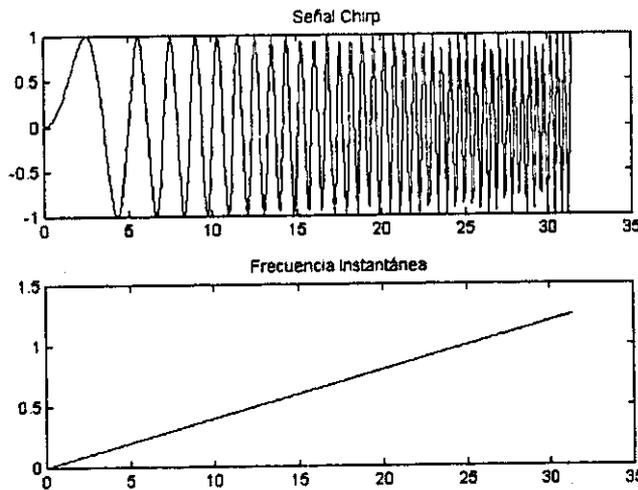


Figura FIG.2.1: Señal Chirp y su frecuencia instantánea.

Para este tipo de señales, la frecuencia instantánea es una característica espectral importante. La frecuencia instantánea es un parámetro que varía con el tiempo, el cual conceptualmente se puede interpretar como la frecuencia de una señal sinusoidal que localmente se ajusta con la señal bajo análisis.

Continuando con el ejemplo de la Señal Chirp (EQ.2.8), su frecuencia instantánea para el instante t_0 es $(m_0 t_0 + b_0)$. En la figura FIG.2.1 se muestra una gráfica de la frecuencia instantánea de esta señal.

Físicamente, la frecuencia instantánea tiene significado solamente para las funciones monocomponentes (aquellas compuesta por una sola señal sinusoidal), donde existe solamente una frecuencia o un estrecho rango de frecuencias que varían con respecto al tiempo. Para señales multicomponentes (aquellas compuestas por varias señales sinusoidales), la noción de frecuencia instantánea única pierde significado.

Además, puesto que la frecuencia se define como el número de ciclos o vibraciones que se presentan por unidad de tiempo en un cuerpo en movimiento periódico, existe una aparente paradoja en la asociación de las palabras frecuencia e instantáneo. Por esta razón, la definición de la frecuencia instantánea causa controversias.

2.6 Señal analítica

En este apartado se definirá señal analítica^{[6][11][13]}, la cual es de utilidad para calcular la frecuencia instantánea.

Sea la señal $x(t)$. Entonces, la señal analítica $x_a(t)$ de la señal $x(t)$ se define como:

$$x_a(t) = x(t) + jH[x(t)] = a(t)e^{j\theta(t)} \quad (\text{EQ.2.9})$$

donde $H[]$ representa el operador transformada de Hilbert. A su vez, la transformada de Hilbert se define como:

$$H[x(t)] = p.v. \int_{-\infty}^{+\infty} \frac{x(t-\tau)}{\pi\tau} dt \quad (\text{EQ.2.10})$$

donde p.v. denota el valor principal de Cauchy de la integral.

Un procedimiento alternativo y práctico para calcular la señal analítica $x_a(t)$ es:

1. Calcular $X(f)$, la transformada de Fourier de la señal $x(t)$.
2. Calcular $X_a(f)$, la transformada de Fourier de la señal analítica $x_a(t)$, a través de la siguiente expresión:

$$X_a(f) = \begin{cases} 2X(f) & f > 0 \\ X(f) & f = 0 \\ 0 & f < 0 \end{cases} \quad (\text{EQ.2.11})$$

3. Calcular $x_a(t)$, la señal analítica, la transformada inversa de Fourier de $X_a(f)$.

Es importante hacer notar que, a lo largo de este trabajo, los cálculos de las diferentes distribuciones tiempo frecuencia se harán con la señal analítica^[2] $x_a(t)$ y no con la señal $x(t)$.

2.7 Frecuencia instantánea

En este apartado se definirá la frecuencia instantánea, la frecuencia media pseudo instantánea y la distribución de potencia pseudo instantánea^{[6][11]}. Sea la señal analítica $x_a(t)$, descrita en (EQ.2.9), en su forma polar:

$$x_a(t) = a(t)e^{j\phi(t)} \quad (\text{EQ.2.12})$$

Entonces, la frecuencia instantánea $f_i(t)$ se define como:

$$f_i(t) = \frac{1}{2\pi} \frac{d}{dt} [\phi(t)] \quad (\text{EQ.2.13})$$

Si se habla de una frecuencia instantánea, intuitivamente ésta debería estar asociada a un valor medio ponderado de un espectro de frecuencias instantáneo. En particular, Ville formuló una distribución tiempo frecuencia comúnmente denominada como distribución tiempo frecuencia de Wigner Ville¹, la cual se define como:

$$WVD(t, f) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \quad (\text{EQ.2.14})$$

y demostró que el valor medio ponderado de $WVD(t, f)$ con respecto a la frecuencia calculaba la frecuencia instantánea, esto es:

$$f_i(t) = \frac{\int_{-\infty}^{+\infty} f \cdot WVD(t, f) df}{\int_{-\infty}^{+\infty} WVD(t, f) df} \quad (\text{EQ.2.15})$$

Generalizando^[10] la ecuación (EQ.2.15) se obtiene:

$$f_i(t) = \frac{\int_0^{+\infty} f \cdot TFD(t, f) df}{\int_0^{+\infty} TFD(t, f) df} \quad (\text{EQ.2.16})$$

donde $TFD(t, f)$ es alguna de las distribuciones tiempo frecuencia de la familia de Cohen (en este trabajo solamente se tratan las distribuciones de Wigner Ville, de Choi Williams y de Bessel), y donde los límites inferiores de integración valen cero porque las señales analíticas no poseen componentes con frecuencias negativas; de otro modo, los límites inferiores de integración valdrían $-\infty$.

2.8 Frecuencia media pseudo instantánea

En este apartado se definirá la frecuencia media pseudo instantánea y la distribución de potencia pseudo instantánea^[11]. Un valor aproximado de la frecuencia instantánea es la frecuencia media pseudo instantánea, la cual se puede calcular a partir de alguna de las distribuciones tiempo frecuencia de la familia de Cohen². Para calcular la frecuencia media pseudo instantánea, primero se debe calcular la distribución de potencia pseudo instantánea.

¹ Ver capítulo de Distribución de Wigner Ville.

² Ver capítulo de Distribuciones Tiempo Frecuencia.

Entonces, la distribución de potencia pseudo instantánea $PIPD(t, f)$ se define como:

$$PIPD(t, f) = \begin{cases} TFD(t, f) & TFD(t, f) \geq 0 \\ 0 & TFD(t, f) < 0 \end{cases} \quad (EQ.2.17)$$

donde $TFD(t, f)$ es alguna de las distribuciones tiempo frecuencia de la familia de Cohen (en este trabajo solamente se tratan las distribuciones de Wigner Ville, de Choi Williams y de Bessel). En los capítulos dedicados a cada una de las distribuciones tiempo frecuencia mencionadas se justifica detalladamente la eliminación de sus valores negativos. Brevemente, lo anterior se debe a que la energía de cada componente de la señal analítica debe ser positiva; en consecuencia, se puede eliminar los términos negativos de la distribución tiempo frecuencia³.

Ahora bien, la frecuencia media pseudo instantánea $PIMF(t)$ se define como:

$$PIMF(t) = \frac{\int_0^{\infty} f \cdot PIPD(t, f) df}{\int_0^{\infty} PIPD(t, f) df} \quad (EQ.2.18)$$

donde los límites inferiores de integración valen cero porque las señales analíticas no poseen componentes con frecuencias negativas; de otro modo, los límites inferiores de integración valdrían $-\infty$.

2.9 Expresiones discretizadas

En este apartado se plantearán las expresiones discretizadas para el cálculo de la frecuencia instantánea, de la frecuencia media pseudo instantánea^{[10][11]}, entre otras expresiones.

Sea la distribución tiempo frecuencia discreta $DTFD(n, k)$, donde n es el tiempo discreto y k es la frecuencia discreta, ambas variables normalizadas⁴.

Entonces la frecuencia instantánea discreta es:

$$f_i(n) = \frac{\sum_{k=0}^{N-1} f_k \cdot DTFD(n, k)}{\sum_{k=0}^{N-1} DTFD(n, k)} \quad (EQ.2.19)$$

donde f_k es la frecuencia discreta correspondiente al punto k (sin normalizar). La distribución pseudo instantánea de potencia discreta $PIPD(n, k)$ es:

$$PIPD(n, k) = \begin{cases} DTFD(n, k) & DTFD(n, k) \geq 0 \\ 0 & DTFD(n, k) < 0 \end{cases} \quad (EQ.2.20)$$

La frecuencia media pseudo instantánea $PIMF(n, k)$ es:

³ Ver apartado de Análisis de la distribución en capítulos de Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel.

⁴ Ver apartado de Caso discreto con ventana en capítulos de Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel.

$$PIMF(n) = \frac{\sum_{k=0}^{N-1} f_k \cdot PIPD(n, k)}{\sum_{k=0}^{N-1} PIPD(n, k)} \quad (\text{EQ.2.21})$$

donde f_k es la frecuencia discreta correspondiente al punto k (sin normalizar).

Para el caso discreto, la transformada de Fourier $X_a(k)$ de la señal analítica $x_a(n)$, se calcula a través de la siguiente expresión⁵:

$$X_a(k) = \begin{cases} 2X(k) & k = 1, \dots, \frac{L}{2} - 1 \\ X(k) & k = 0 \\ 0 & k = \frac{L}{2}, \dots, L-1 \end{cases} \quad (\text{EQ.2.22})$$

donde n es el tiempo discreto y k es la frecuencia discreta, ambas variables normalizadas.

⁵ Para señales discretas de longitud impar.

3 DISTRIBUCIONES TIEMPO FRECUENCIA

3.1 Introducción

En este capítulo se plantea la clase de Cohen de distribuciones de tiempo frecuencia. Además se muestra una tabla con distintas distribuciones tiempo frecuencia. La bibliografía básica de este capítulo consiste en [1] y [2]; estudios comparativos de distintas distribuciones se encuentran en [4], [19] y [20]. Otras definiciones de distribuciones tiempo frecuencia se encuentran en [4] y en [5], la clase RID (*Reduced Interference Distributions*) y la distribución AF-GDTTFD (*Alias-Free Generalized Discrete-Time Time-Frequency Distributions*), respectivamente.

Las expresiones importantes de este capítulo son:

- Distribución tiempo frecuencia.

$$TFD(t, f, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f\tau} d\tau$$

- Distribución tiempo frecuencia con una función ventana.

$$TFD_w(t, f) = \frac{1}{2\pi} \int_{-T_w}^{T_w} \int_{-\infty}^{\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

- Distribución discreta tiempo frecuencia.

$$DTFD(n, k) = \frac{1}{\pi} \sum_{\tau=-N+1}^{N-1} W(\tau) W^*(-\tau) e^{-j2\pi k\tau} \sum_{\mu=-M}^M \psi(-\mu, 2\tau) x(\mu + n + \tau) x^*(\mu + n - \tau)$$

- Transformación de la distribución discreta tiempo frecuencia.

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-j2\pi k\tau} \sum_{\mu=-M}^M \psi(-\mu, 2\tau) x(\mu + \tau) x^*(\mu - \tau) \right] - 2|x(0)|^2$$

3.2 Caso continuo

La clase de Cohen de distribuciones tiempo frecuencia se puede plantear de la siguiente forma ^{[1][2][10]}:

Primero, sea el Kernel de la Distribución Tiempo Frecuencia $\phi(\theta, \tau)$. En la Tabla (TB.3.1) se enuncian los kernels de distintas distribuciones tiempo frecuencia.

Segundo, sea el Kernel del Dominio de la Autocorrelación, el cual se define como la transformada de Fourier del kernel de la distribución tiempo frecuencia:

$$\psi(t, \tau) = \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta t} d\theta \tag{EQ.3.1}$$

Tercero, sea la Función de Autocorrelación Instantánea de la señal $x(t)$, la cual se define como:

$$R_x(t, \tau) = x\left(t + \frac{\tau}{2}\right)x^*\left(t - \frac{\tau}{2}\right) \quad (\text{EQ.3.2})$$

Cuarto, sea la Función de Autocorrelación Generalizada Indexada al Tiempo, la cual se define como:

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) R_x(\mu, \tau) d\mu \quad (\text{EQ.3.3})$$

Si se sustituye (EQ.3.2) en (EQ.3.3) se obtiene:

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.3.4})$$

Quinto, la Clase de Cohen de las Distribuciones Tiempo Frecuencia con kernel $\phi(\theta, \tau)$ se define como la transformada de Fourier de la función de autocorrelación generalizada indexada al tiempo:

$$TFD(t, \omega, \phi) = \int_{-\infty}^{+\infty} R'_x(t, \tau) e^{-j\omega\tau} d\tau \quad (\text{EQ.3.5})$$

Entonces, si se sustituye (EQ.3.4) en (EQ.3.5) se obtiene:

$$TFD(t, \omega, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.3.6})$$

o bien, en función de la frecuencia:

$$TFD(t, f, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f\tau} d\tau \quad (\text{EQ.3.7})$$

Finalmente, si se sustituye (EQ.3.1) en (EQ.3.6) se obtiene:

$$TFD(t, \omega, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta(t-\mu)} d\theta x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.3.8})$$

Entonces, si se reagrupa (EQ.3.8) se obtiene:

$$TFD(t, \omega, \phi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi(\theta, \tau) x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) e^{-j\theta t + j\theta\mu - j\omega\tau} d\theta d\mu d\tau \quad (\text{EQ.3.9})$$

En conclusión, *todas* ^{[1][2]} las representaciones tiempo frecuencia se pueden obtener de (EQ.3.9) en donde el kernel $\phi(\theta, \tau)$ determina la distribución y sus propiedades.

3.3 Caso continuo con ventana

Se desarrollará la clase de Cohen de distribuciones tiempo frecuencia cuando se introduce una función ventana $W(t)$ que multiplica a la función autocorrelación indexada al tiempo generalizada $R'_x(t, \tau)$ ^[9].

Si se introduce una función ventana $W(t)$ que cumpla con la propiedad de $W(t) = 0 \quad |t| > T_w/2$ en (EQ.3.7), entonces se obtiene:

$$TFD_w(t, f) = \frac{1}{2\pi} \int_{-T_w/2}^{T_w/2} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) x\left(\mu + \frac{\tau}{2}\right)x^*\left(\mu - \frac{\tau}{2}\right) d\mu W\left(\frac{\tau}{2}\right)W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \quad (\text{EQ.3.10})$$

3.4 Caso discreto con ventana

Se desarrollará la clase de Cohen de distribuciones tiempo frecuencia discretas a partir de las distribuciones continuas^[9]. Esto es, se discretizará (EQ.3.10).

Primero, si en (EQ.3.10) se realizan los cambios de variables $\tau' = \tau/2$ y $\mu' = \mu - t$, entonces se obtiene:

$$TFD_w(t, f) = \frac{2}{2\pi} \int_{-T_w}^{T_w} \int_{-\infty}^{\infty} \psi(-\mu', 2\tau') x(\mu'+t+\tau') x^*(\mu'+t-\tau') d\mu' W(\tau') W^*(-\tau') e^{-j2\pi f(2\tau')} d\tau' \quad (\text{EQ.3.11})$$

Segundo, si en (EQ.3.11) se discretiza la variable frecuencia $f = k\Delta f$, donde $\Delta f = \frac{1}{L} = \frac{1}{2N-1} \approx \frac{1}{2N}$ ^[8], siendo L el número de muestras por ventana, y se discretiza la variable tiempo $t = n\Delta T$, $\mu' = \mu\Delta T$ y $\tau' = \tau\Delta T$, donde n, μ y τ son índices discretos, entonces se obtiene:

$$DTFD(n\Delta T, k\Delta f) = \frac{2}{2\pi} \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \psi(-\mu\Delta T, 2\tau\Delta T) x(\mu\Delta T + n\Delta T + \tau\Delta T) \cdot x^*(\mu\Delta T + n\Delta T - \tau\Delta T) W(\tau\Delta T) W^*(-\tau\Delta T) e^{-\frac{j4\pi k\tau\Delta T}{2N} \Delta T \Delta T} \quad (\text{EQ.3.12})$$

Tercero, si en (EQ.3.12) se normaliza el incremento en tiempo $\Delta T=1$ y el incremento en frecuencia $\Delta f=1$, entonces se obtiene:

$$DTFD(n, k) = \frac{1}{\pi} \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \psi(-\mu, 2\tau) x(\mu+n+\tau) x^*(\mu+n-\tau) W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \quad (\text{EQ.3.13})$$

Finalmente, la distribución tiempo frecuencia generalizada discreta es:

$$DTFD(n, k) = \frac{1}{\pi} \sum_{\tau=-N+1}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \psi(-\mu, 2\tau) x(\mu+n+\tau) x^*(\mu+n-\tau) \quad (\text{EQ.3.14})$$

en donde si $\tau=0$ entonces $\sum_{\mu=-M}^M \psi(-\mu, 2\tau) x(\mu+n+\tau) x^*(\mu+n-\tau) = x(n)x^*(n)$.

3.5 Análisis de la distribución

Se analizará la influencia del kernel de la clase de Cohen de distribuciones tiempo frecuencia en la disminución de los efectos de interferencia entre los diferentes componentes de una señal^{[10][11][16]}.

Sea la señal multicomponente $x(t)$ la cual se puede representar como la suma de sus componentes unitarios:

$$x(t) = \sum_{n=1}^N A_n e^{j(\omega_n t + \theta_n)} \quad (\text{EQ.3.15})$$

Si se sustituye (EQ.3.15) en (EQ.3.6) se obtiene:

$$TFD(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \psi(t-\mu, \tau) \sum_{n=1}^N A_n e^{j(\omega_n(\mu+\frac{\tau}{2})+\theta_n)} \sum_{m=1}^N A_m e^{-j(\omega_m(\mu-\frac{\tau}{2})+\theta_m)} d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.3.16})$$

Reagrupando términos en (EQ.3.16) se obtiene:

$$TFD(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j(\frac{\omega_n + \omega_m}{2}\tau)} \int_{-\infty}^{+\infty} \psi(t-\mu, \tau) e^{j(\omega_n - \omega_m)\mu} d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.3.17})$$

Si se aplica la propiedad asociativa de la suma en (EQ.3.17), entonces se obtiene:

$$TFD(t, \omega) = TFD_{auto}(t, \omega) + TFD_{cruce}(t, \omega) \quad (EQ.3.18)$$

donde, para el caso en el cual los índices de las sumatorias en (EQ.3.17) son iguales, es decir, $n = m$, se tiene que:

$$TFD_{auto}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) d\mu e^{-j\omega \tau} d\tau \quad (EQ.3.19)$$

y para el caso en el cual los índices de las sumatorias en (EQ.3.17) son diferentes $n \neq m$, se tiene que:

$$TFD_{cruce}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\theta_n - \theta_m) t} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right) \tau} \int_{-\infty}^{+\infty} \psi(t - \mu, \tau) e^{j(\omega_n - \omega_m) \mu} d\mu e^{-j\omega \tau} d\tau \quad (EQ.3.20)$$

Primero se analizará (EQ.3.19), la cual se constituye de los auto-términos de la distribución tiempo frecuencia, esto es, de la distribución de cada componente de la señal. Luego se analizará (EQ.3.20), la cual se constituye de los términos de cruce de la distribución tiempo frecuencia, esto es, de la distribución de la interacción de cada par de componentes diferentes.

Referente a (EQ.3.19) o auto-términos. La integral con respecto a μ es:

$$\int_{-\infty}^{+\infty} \psi(t - \mu, \tau) d\mu = 2\pi \quad (EQ.3.21)$$

Sustituyendo (EQ.3.21) en (EQ.3.19) se obtiene:

$$TFD_{auto}(t, \omega) = \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} e^{-j\omega \tau} d\tau \quad (EQ.3.22)$$

Si se integra (EQ.3.22) se obtiene:

$$TFD_{auto}(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) \quad (EQ.3.23)$$

Referente a (EQ.3.20) o términos de cruce. Para integrar con respecto a μ se aplica la ecuación de Parseval. Esto es:

$$\int_{-\infty}^{+\infty} [\psi(t - \mu, \tau)] [e^{j(\omega_n - \omega_m) \mu}] d\mu = \int_{-\infty}^{+\infty} [2\pi e^{-j\mu} \phi(\mu, \tau)] [\delta(\mu + (\omega_n - \omega_m))] d\mu \quad (EQ.3.24)$$

Integrando (EQ.3.24) se obtiene:

$$\int_{-\infty}^{+\infty} [2\pi e^{-j\mu} \phi(\mu, \tau)] [\delta(\mu + (\omega_n - \omega_m))] d\mu = 2\pi \phi(-(\omega_n - \omega_m), \tau) e^{j(\omega_n - \omega_m) \tau} \quad (EQ.3.25)$$

Si se sustituye (EQ.3.25) en (EQ.3.20) se obtiene:

$$TFD_{cruce}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\theta_n - \theta_m) t} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right) \tau} 2\pi \phi(-(\omega_n - \omega_m), \tau) e^{j(\omega_n - \omega_m) \tau} e^{-j\omega \tau} d\tau \quad (EQ.3.26)$$

Reagrupando términos en (EQ.3.26) se obtiene:

$$TFD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\omega_n - \omega_m) t + \theta_n - \theta_m} \int_{-\infty}^{+\infty} \phi(-(\omega_n - \omega_m), \tau) e^{j\left(\frac{\omega_n + \omega_m}{2}\right) \tau} e^{-j\omega \tau} d\tau \quad (EQ.3.27)$$

Si se integra (EQ.3.27) se obtiene:

$$TFD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\omega_n - \omega_m) t + \theta_n - \theta_m} \psi\left(\omega - \left(\frac{\omega_n + \omega_m}{2}\right), -(\omega_n - \omega_m)\right) \quad (EQ.3.28)$$

Puesto que la distribución es real, entonces:

$$TFD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{m=1}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \psi\left(-(\omega_n - \omega_m), \omega - \left(\frac{\omega_n + \omega_m}{2}\right)\right) \quad (\text{EQ.3.29})$$

Finalmente:

$$TFD(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) + \sum_{n=1}^N \sum_{m=1}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \psi\left(\omega - \left(\frac{\omega_n + \omega_m}{2}\right), -(\omega_n - \omega_m)\right) \quad (\text{EQ.3.30})$$

El factor de peso en (EQ.3.29) es:

$$FP_{TFD}(\omega) = \psi\left(\omega - \left(\frac{\omega_n + \omega_m}{2}\right), -(\omega_n - \omega_m)\right) \quad (\text{EQ.3.31})$$

3.6 Algoritmo eficiente

El algoritmo eficiente calcula (EQ.3.14) valuada en $n = 0$ ^{[9][12][17]}; por consiguiente:

$$DTFD(0, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \quad (\text{EQ.3.32})$$

Si se separa (EQ.3.32) en dos sumatorias, una para los índices positivos y otra para los índices negativos de τ , se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=-N+1}^{-1} W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \quad (\text{EQ.3.33})$$

Si a (EQ.3.33) se suma y se resta el término $2W(0)W^*(0)x(0)x^*(0)$ correspondiente al índice $\tau=0$, se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=-N+1}^0 W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) - 2W(0)W^*(0)x(0)x^*(0) \quad (\text{EQ.3.34})$$

Si en la primera sumatoria de (EQ.3.34) se cambia el índice τ por $-\tau$, se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau) e^{\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, -2\tau)x(\mu-\tau)x^*(\mu+\tau) + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) - 2x(0)x^*(0) \quad (\text{EQ.3.35})$$

Si en el primera sumatoria de (EQ.3.35) se aplica el conjugado complejo del conjugado complejo, se obtiene:

$$\begin{aligned}
 DCWD(0, k) = & \left(2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, -2\tau)x(\mu-\tau)x^*(\mu+\tau) \right)^* \\
 & + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \\
 & - 2x(0)x^*(0)
 \end{aligned} \tag{EQ.3.36}$$

Si en la primera sumatoria de (EQ.3.36) se calcula uno de los conjugados complejos, se obtiene:

$$\begin{aligned}
 DCWD(0, k) = & \left(2 \sum_{\tau=0}^{N-1} W^*(-\tau)W(\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, -2\tau)x^*(\mu-\tau)x(\mu+\tau) \right) \\
 & + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \\
 & - 2x(0)x^*(0)
 \end{aligned} \tag{EQ.3.37}$$

Entonces, puesto que $\psi(-\mu, 2\tau)$ es par, reagrupando términos se tiene:

$$DCWD(0, k) = 2 \operatorname{Re} \left[2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \right] - 2x(0)x^*(0) \tag{EQ.3.38}$$

Finalmente se tiene que:

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \frac{1}{2\pi} \psi(-\mu, 2\tau)x(\mu+\tau)x^*(\mu-\tau) \right] - 2|x(0)|^2 \tag{EQ.3.39}$$

donde es indispensable revisar los límites de la sumatoria respecto a μ para eliminar las multiplicaciones triviales.

3.7 Tabla de distribuciones

A continuación se presenta una tabla con diferentes distribuciones tiempo frecuencia de la familia de Cohen y sus kerneles.

Nombre	Kernel $\phi(\theta, \tau)$	Distribución TFD(t, ω)
Wigner Ville	1	$\frac{1}{2\pi} \int e^{-j\tau\omega} x^*(t - \frac{1}{2}\tau)x(t + \frac{1}{2}\tau) d\tau$
Margenau Hill	$\cos\left(\frac{1}{2}\theta\tau\right)$	$\text{Re} \left[\frac{1}{-j2\pi} \cdot x(t) e^{-j\omega} X^*(\omega) \right]$
Kirkwood Rihaczek	$e^{\frac{j\theta\tau}{2}}$	$\frac{1}{-j2\pi} \cdot x(t) e^{-j\omega} X^*(\omega)$
Born Jordan	$\frac{\text{sen}\left(\frac{1}{2}\theta\tau\right)}{\frac{1}{2}\theta\tau}$	$\frac{1}{2\pi} \int_{\tau_1}^{\tau_2} e^{-j\tau\omega} \int_{t-\tau/2}^{t+\tau/2} x^*(u - \frac{1}{2}\tau)x(u + \frac{1}{2}\tau) du d\tau$
Born Jordan paramétrica	$\frac{\text{sen}(\alpha\theta\tau)}{\alpha\theta\tau}$	$\frac{1}{4\pi\alpha} \int_{\tau}^{\tau} e^{-j\tau\omega} \int_{t-\alpha\tau}^{t+\alpha\tau} x^*(u - \frac{1}{2}\tau)x(u + \frac{1}{2}\tau) du d\tau$
Page	$e^{\frac{j\theta\tau}{2}}$	$\frac{\partial}{\partial t} \left \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t x(t') e^{-j\omega t'} dt' \right ^2$
Choi Williams	$e^{-\frac{\theta^2 \tau^2}{\sigma}}$	$\frac{1}{4\pi^{\frac{3}{2}}} \int \int_{\tau_1}^{\tau_2} \frac{\sigma}{\tau^2} e^{-\frac{\sigma(u-\tau)^2}{4\tau^2} - j\tau\omega} x^*(u - \frac{1}{2}\tau)x(u + \frac{1}{2}\tau) du d\tau$
Espectrograma	$\int h^*(u - \frac{1}{2}\tau) e^{-j\theta u} \cdot h(u + \frac{1}{2}\tau) du$	$\frac{1}{2\pi} \int e^{-j\omega\tau} x(\tau) h(\tau - t) d\tau$
Zhao Atlas Marks	$g(\tau)\tau \frac{\text{sen}(a\theta\tau)}{a\theta\tau}$	$\frac{1}{4\pi a} \int g(\tau) e^{-j\tau\omega} \int_{t-\tau/a}^{t+\tau/a} x^*(u - \frac{1}{2}\tau)x(u + \frac{1}{2}\tau) du d\tau$

Tabla TB.3.1: Distintas Distribuciones Tiempo Frecuencia de la Clase de Cohen y sus Kerneles".

Nombre	Kernel $\phi(\theta, \tau)$	Distribución $TFD(t, \omega)$
Béssel	$\phi(\theta, \tau) = \frac{2J_1(\alpha\tau\theta)}{\alpha\tau\theta}$	$BD(t, f) = \int_{-\infty}^{+\infty} \frac{2}{\pi\alpha \tau } \int_{-\infty}^{+\infty} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) x\left(\mu + \frac{\tau}{2}\right) x\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j\omega t} d\tau$
Sinc	$rect\left(\frac{\theta\tau}{\alpha}\right)$	$\frac{1}{\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\text{sen}\left(\frac{\alpha(t-\mu)}{2\tau}\right)}{t-\mu} x\left(\mu + \frac{1}{2}\tau\right) x\left(\mu - \frac{1}{2}\tau\right) d\mu e^{-j\omega t} d\tau$

Tabla TB.3.1 (continuación): Distintas Distribuciones Tiempo Frecuencia de la Clase de Cohen y sus Kerneles.

En [1],[2] el Kernel del Dominio de la Autocorrelación se define como:

$$\psi(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta} d\theta$$

mientras que en [10] como:

$$\psi(t, \tau) = \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta} d\theta$$

En este trabajo se adopta la última definición. Los resultados que se obtienen al utilizar la primera definición en lugar de la segunda difieren por un factor de $1/2\pi$.

ii En esta tabla se considera que el Kernel del Dominio de la Autocorrelación se define como:

$$\psi(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta} d\theta$$

excepto para la distribución de Béssel, en donde se define como:

$$\psi(t, \tau) = \int_{-\infty}^{+\infty} \phi(\theta, \tau) e^{-j\theta} d\theta$$

4 DISTRIBUCIÓN DE WIGNER VILLE

4.1 Introducción

En este capítulo se plantea la distribución de Wigner Ville a partir de la clase de distribuciones de tiempo frecuencia de Cohen. Posteriormente se desarrolla esta distribución para el caso discreto. Finalmente se transforma la distribución discreta obtenida en una expresión más adecuada para implantar un algoritmo eficiente que la calcule. Además, se desarrollan experimentos numéricos en Matlab que muestran el comportamiento de la distribución discreta en función de la longitud de ventana de muestreo y en función de la relación señal a ruido de la señal muestreada, así como una aplicación a señales Dópler ultrasónicas. La bibliografía básica de este capítulo consiste en [8], [9] y [27]; estudios con señales tipo Dópler ultrasónicas se encuentran en [10], [11] y [12]; estudios comparativos entre diferentes distribuciones se encuentran en [4] y [20]; un estudio sobre suavizado óptimo de los términos de cruce de la distribución se encuentra en [23]. Es importante mencionar que la distribución de Wigner se puede considerar como una generalización del teorema de autocorrelación de Wiener-Khintchine para funciones de autocorrelación simétricas y dependientes del tiempo^{[3][22]}.

Las expresiones principales de este capítulo son:

- Distribución de Wigner Ville.

$$VWD(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

- Distribución de Wigner Ville con una función ventana.

$$VWD_w(t, f) = \int_{-T_w}^{+T_w} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

- Distribución discreta de Wigner Ville.

$$DVWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} x(n+\tau) x^*(n-\tau) W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}}$$

- Transformación de la distribución discreta de Wigner Ville.

$$DVWD(0, k) = 4 \operatorname{Real} \left[\sum_{\tau=0}^{N-1} x(\tau) x^*(-\tau) W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \right] - 2|x(0)|^2$$

4.2 Caso continuo

Se planteará la distribución tiempo frecuencia de Wigner Ville^{[8][9]} a partir de la clase de distribuciones de tiempo frecuencia de Cohen^{[1][2]}.

Primero, el kernel de la distribución de Wigner Ville es^{[10][11][12]}:

$$\phi(\theta, \tau) = 1$$

(EQ.4.1)

En la figura FIG.4.1 se muestra la gráfica de (EQ.4.1) en el dominio θ, τ .

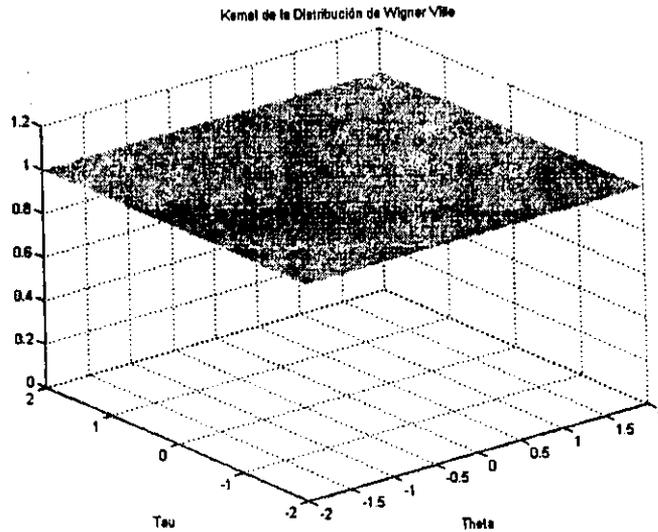


Figura FIG.4.1: Kernel (EQ.4.1) de la distribución tiempo frecuencia de Wigner Ville.

Segundo, el kernel del dominio de la autocorrelación (EQ.3.1) es:

$$\psi(t, \tau) = \int_{-\infty}^{+\infty} e^{-j\theta} d\theta \quad (\text{EQ.4.2})$$

Entonces, si se integra (EQ.4.2) se obtiene (A.4.1):

$$\psi(t, \tau) = 2\pi\delta(t) \quad (\text{EQ.4.3})$$

En la figura FIG.4.2 se muestra la gráfica de (EQ.4.3) en el dominio t, τ . Tercero, la función de autocorrelación instantánea (EQ.3.2) es:

$$R_x(t, \tau) = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \quad (\text{EQ.4.4})$$

Cuarto, la función de autocorrelación indexada al tiempo generalizada (EQ.3.3) es:

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} 2\pi\delta(t - \mu) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.4.5})$$

Entonces, si se integra (EQ.4.5) se obtiene (A.4.2):

$$R'_x(t, \tau) = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \quad (\text{EQ.4.6})$$

Finalmente, la distribución tiempo frecuencia (EQ.3.5) de Wigner Ville es:

$$VWD(t, \omega) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j\omega\tau} d\tau \quad (\text{EQ.4.7})$$

o bien, en función de la frecuencia:

$$VWD(t, f) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \quad (\text{EQ.4.8})$$

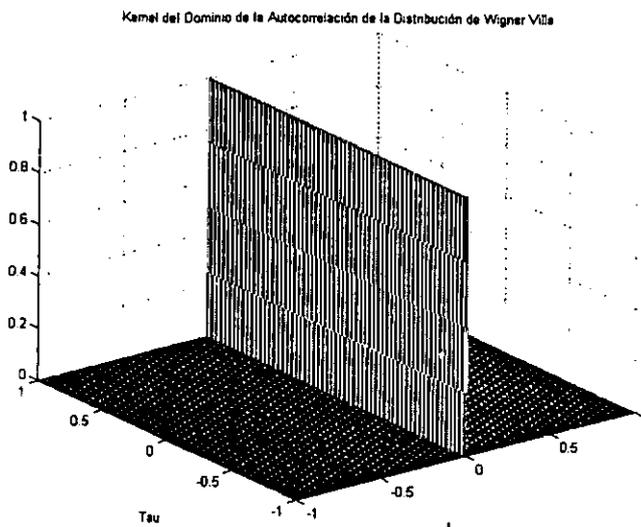


Figura FIG.4.2: Kernel del dominio de la autocorrelación (EQ.4.3) de la distribución tiempo frecuencia de Wigner Ville.

4.3 Caso continuo con ventana

Se desarrollará la distribución tiempo frecuencia de Wigner Ville cuando se introduce una función ventana $W(t)$ que multiplica a la función de autocorrelación indexada al tiempo generalizada $R_x(t, \tau)$ ^{[9][27]}, es decir, a (EQ.4.6).

Si se introduce una función ventana $W(t)$ que cumpla con la propiedad de $W(t) = 0 \quad |t| > T_w/2$ en (EQ.4.8), entonces se obtiene:

$$VWD_w(t, f) = \int_{-T_w/2}^{+T_w/2} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f \tau} d\tau \quad (\text{EQ.4.9})$$

4.4 Caso discreto con ventana

Se desarrollará la distribución tiempo frecuencia de Wigner Ville discreta a partir de la distribución continua^{[9][10][27]}. Esto es, se discretizará (EQ.4.9).

Primero, si en (EQ.4.9) se realizan los cambios de variables $\tau' = \tau/2$ y $\mu' = \mu - t$, entonces se obtiene:

$$VWD_w(t, f) = 2 \int_{-T_w/2}^{T_w/2} x(t + \tau') x^*(t - \tau') W(\tau') W^*(-\tau') e^{-j2\pi f (2\tau')} d\tau' \quad (\text{EQ.4.10})$$

Segundo, si en (EQ.4.10) se discretiza la variable frecuencia $f = k\Delta f$, donde $\Delta f = \frac{1}{L} = \frac{1}{2N-1} \approx \frac{1}{2N}$ ^[8] y k es un índice discreto, siendo L el número de muestras por ventana, y se discretiza la variable tiempo $t = n\Delta T$, y $\tau' = \tau\Delta T$, donde n y τ son índices discretos, entonces se obtiene:

$$DVWD(n\Delta T, k\Delta f) = 2 \sum_{\tau=-N+1}^{N-1} x(n\Delta T + \tau\Delta T)x^*(n\Delta T - \tau\Delta T)W(\tau\Delta T)W^*(-\tau\Delta T)e^{\frac{j4\pi k\tau\Delta T}{2N}} \Delta T \quad (\text{EQ.4.11})$$

Tercero, si en (EQ.4.11) se normaliza el incremento en tiempo $\Delta T = 1$ y el incremento en frecuencia $\Delta f = 1$, entonces se obtiene:

$$DVWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} x(n+\tau)x^*(n-\tau)W(\tau)W^*(-\tau)e^{\frac{j2\pi k\tau}{N}} \quad (\text{EQ.4.12})$$

Finalmente, la distribución tiempo frecuencia de Wigner Ville discreta es:

$$DVWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} x(n+\tau)x^*(n-\tau)W(\tau)W^*(-\tau)e^{\frac{j2\pi k\tau}{N}} \quad (\text{EQ.4.13})$$

4.5 Análisis de la distribución

Se analizará la efectividad del kernel (EQ.4.1) de la distribución de Wigner Ville (EQ.4.7) en la disminución de los efectos de interferencia entre los diferentes componentes de una señal^{[10][11]}.

Sea la señal multicomponente $x(t)$ la cual se puede representar como la suma de sus componentes unitarios:

$$x(t) = \sum_{n=1}^N A_n e^{j(\omega_n t + \theta_n)} \quad (\text{EQ.4.14})$$

Si se sustituye (EQ.4.14) en (EQ.4.7) se obtiene:

$$WVD(t, \omega) = \int_{-\infty}^{+\infty} \sum_{n=1}^N A_n e^{j(\omega_n(t+\frac{\tau}{2}) + \theta_n)} \sum_{m=1}^N A_m e^{-j(\omega_m(t-\frac{\tau}{2}) + \theta_m)} e^{-j\omega\tau} d\tau \quad (\text{EQ.4.15})$$

Reagrupando términos en (EQ.4.15) se obtiene:

$$WVD(t, \omega) = \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j((\omega_n - \omega_m)t + \theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} e^{-j\omega\tau} d\tau \quad (\text{EQ.4.16})$$

Si se aplica la propiedad asociativa de la suma en (EQ.4.16), entonces se obtiene:

$$WVD(t, \omega) = WVD_{\text{auto}}(t, \omega) + WVD_{\text{cruce}}(t, \omega) \quad (\text{EQ.4.17})$$

donde, para el caso en el cual los índices de las sumatorias en (EQ.4.16) son iguales, es decir, $n = m$, se tiene que:

$$WVD_{\text{auto}}(t, \omega) = \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} e^{-j\omega\tau} d\tau \quad (\text{EQ.4.18})$$

y para el caso en el cual los índices de las sumatorias en (EQ.4.16) son diferentes, es decir, $n \neq m$, se tiene que:

$$WVD_{\text{cruce}}(t, \omega) = \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j((\omega_n - \omega_m)t + \theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} e^{-j\omega\tau} d\tau \quad (\text{EQ.4.19})$$

Primero se analizará (EQ.4.18), la cual se constituye de los auto-términos de la distribución tiempo frecuencia, esto es, de la distribución de cada componente. Luego se analizará (EQ.4.19), la cual se constituye de los términos de cruce de la distribución tiempo frecuencia, esto es, de la distribución de la interacción de cada par de componentes diferentes.

Referente a (EQ.4.18) o auto-términos. Si se integra se obtiene^(A.4.3):

$$WVD_{auto}(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) \quad (\text{EQ.4.20})$$

Referente a (EQ.4.19) o términos de cruce. Si se integra se obtiene^(A.4.4):

$$WVD_{cruce}(t, \omega) = 2\pi \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j((\omega_n - \omega_m)t + \theta_n - \theta_m)} \delta\left(\omega - \frac{\omega_n + \omega_m}{2}\right) \quad (\text{EQ.4.21})$$

Puesto que la distribución es real, entonces:

$$WVD_{cruce}(t, \omega) = 2\pi \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \delta\left(\omega - \frac{\omega_n + \omega_m}{2}\right) \quad (\text{EQ.4.22})$$

Finalmente:

$$WVD(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) + 2\pi \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \delta\left(\omega - \frac{\omega_n + \omega_m}{2}\right) \quad (\text{EQ.4.23})$$

El factor de peso en (EQ.4.22), el cual altera la magnitud de los auto-términos, es:

$$FP_{WVD}(\omega) = 2\pi \delta\left(\omega - \frac{\omega_n + \omega_m}{2}\right) \quad (\text{EQ.4.24})$$

En la figura FIG.4.3 se muestra una gráfica de (EQ.4.24) para diversos valores de ω_n y ω_m .

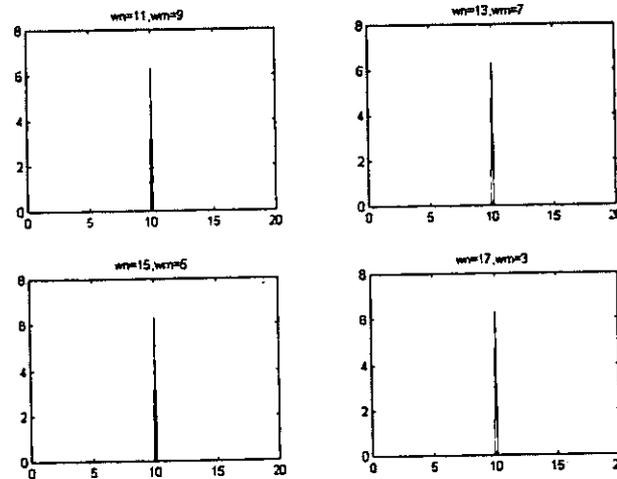


Figura FIG.4.3: Factor de peso (EQ.4.24) de la distribución tiempo frecuencia de Wigner Ville para diversos valores de ω_n y ω_m .

4.6 Algoritmo eficiente

El algoritmo eficiente calcula (EQ.4.13) valuada en $n = 0$ ^{[9][12]}, por consiguiente:

$$DVWD(0,k) = 2 \sum_{\tau=-N+1}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} \quad (\text{EQ.4.25})$$

Si se separa (EQ.4.25) en dos sumatorias, una para los índices positivos y otra para los índices negativos de τ , se obtiene:

$$DVWD(0,k) = 2 \sum_{\tau=-N+1}^{-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} + 2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} \quad (\text{EQ.4.26})$$

Si a (EQ.4.26) se suma y se resta el término $2W(0)W^*(0)x(0)x^*(0)$ correspondiente al índice $\tau=0$, se tiene que:

$$DVWD(0,k) = 2 \sum_{\tau=-N+1}^0 x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} + 2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} - 2W(0)W^*(0)x(0)x^*(0) \quad (\text{EQ.4.27})$$

Si en la primera sumatoria de (EQ.4.27) se cambia el índice τ por $-\tau$, y considerando que $W(0)=1$, se obtiene:

$$DVWD(0,k) = 2 \sum_{\tau=0}^{N-1} x(-\tau)x^*(\tau)W(-\tau)W^*(\tau)e^{\frac{j2k\tau}{N}} + 2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} - 2x(0)x^*(0) \quad (\text{EQ.4.28})$$

Si en la primera sumatoria de (EQ.4.28) se aplica el complejo conjugado del complejo conjugado, se obtiene:

$$DVWD(0,k) = 2 \left(\sum_{\tau=0}^{N-1} x(-\tau)x^*(\tau)W(-\tau)W^*(\tau)e^{\frac{j2k\tau}{N}} \right)^{**} + 2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} - 2x(0)x^*(0) \quad (\text{EQ.4.29})$$

Si en la primera sumatoria de (EQ.4.29) se calcula uno de los complejos conjugados, se obtiene:

$$DVWD(0,k) = 2 \left(\sum_{\tau=0}^{N-1} x^*(-\tau)x(\tau)W^*(-\tau)W(\tau)e^{\frac{j2k\tau}{N}} \right)^* + 2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} - 2x(0)x^*(0) \quad (\text{EQ.4.30})$$

Entonces, reagrupando términos se tiene^(A.4.5):

$$DVWD(0,k) = 2 \text{Real} \left[2 \sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} \right] - 2x(0)x^*(0) \quad (\text{EQ.4.31})$$

Finalmente se tiene que:

$$DVWD(0,k) = 4 \text{Real} \left[\sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} \right] - 2|x(0)|^2 \quad (\text{EQ.4.32})$$

4.6.1 Operaciones realizadas

El número de operaciones que se realizan en el cálculo de la distribución discreta de Wigner Ville según las ecuaciones (EQ.4.13) y (EQ.4.32) para cada k es:

	(EQ. 4.13)	(EQ. 4.32)
Multiplicaciones complejas	6N-3	3N+1
Sumas complejas	2N-2	N
Multiplicaciones por escalares	1	2

N es la longitud de la ventana de muestreo. Se considera a $W(\tau)W^*(-\tau)$ como un solo factor.

4.7 Experimentos Numéricos en Matlab

En esta sección se presentarán diversos experimentos numéricos^[11] realizados en Matlab^{[32][33]} para comparar gráficamente el comportamiento de la frecuencia instantánea con el comportamiento de la frecuencia media pseudo instantánea y para cuantificar éste último, al variar el nivel de la potencia del ruido en la señal (SNR). En el capítulo 2, Generalidades, se presentan las expresiones para calcular la frecuencia instantánea discreta (EQ.2.19) y la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presentan las expresiones para calcular la relación señal a ruido *SNR* (EQ.A4.2) y el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5). Para todos los experimentos, la longitud de ventana de muestreo es de 127 elementos.

4.7.1 Experimento 1: Función seno con frecuencia de 0.32Hz y amplitud constante

En la figura FIG.4.4 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función seno con frecuencia constante de 0.32 Hz y amplitud constante. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.32 Hz.

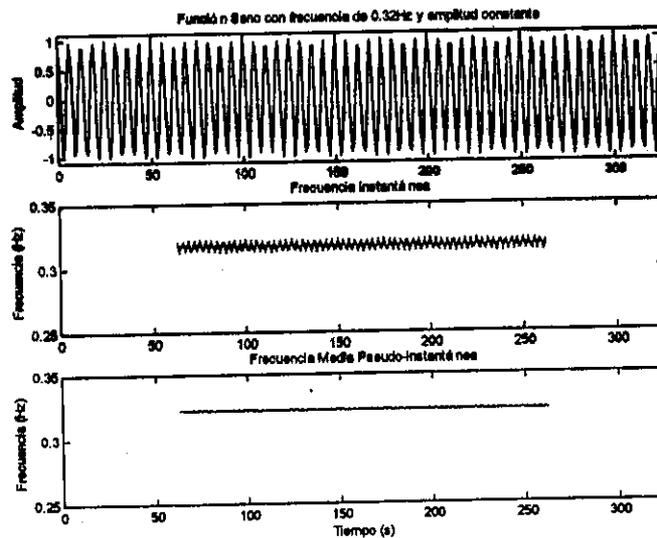


Figura FIG.4.4: a) Función seno con frecuencia de 0.32Hz y amplitud constante; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.4.5 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

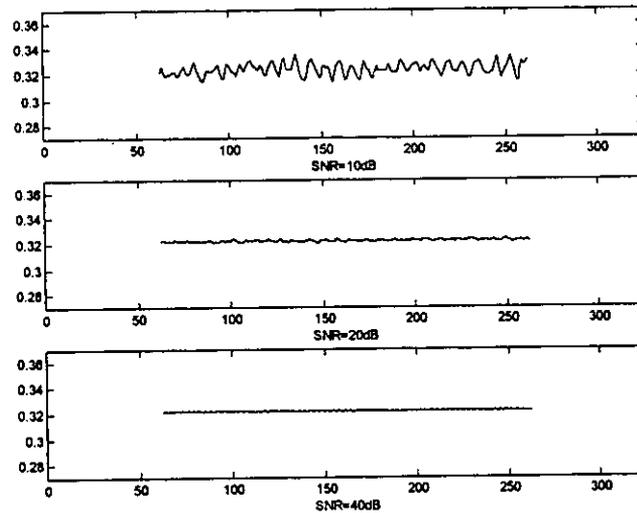


Figura FIG.4.5: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

4.7.2 Experimento 2: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes

En la figura FIG.4.6 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.28 Hz.

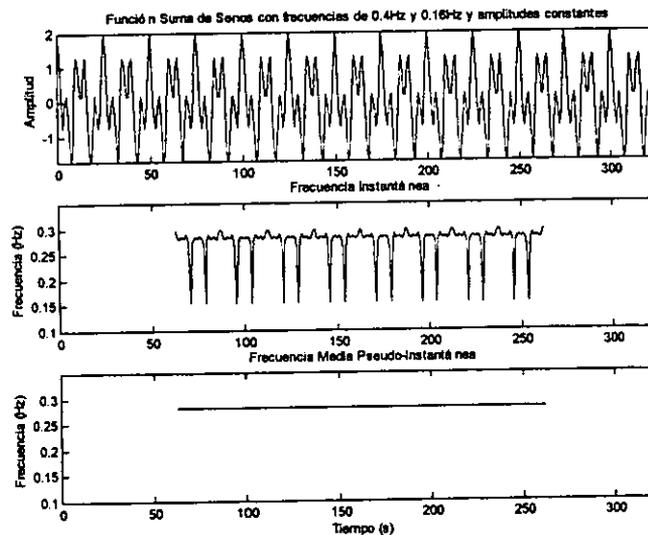


Figura FIG.4.6: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.4.7 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

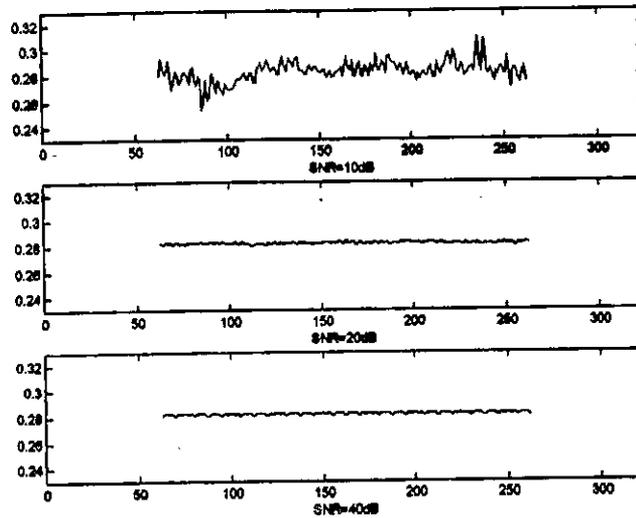


Figura FIG.4.7: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

4.7.3 Experimento 3: Función Suma de Senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes

En la figura FIG.4.8 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.24 Hz.

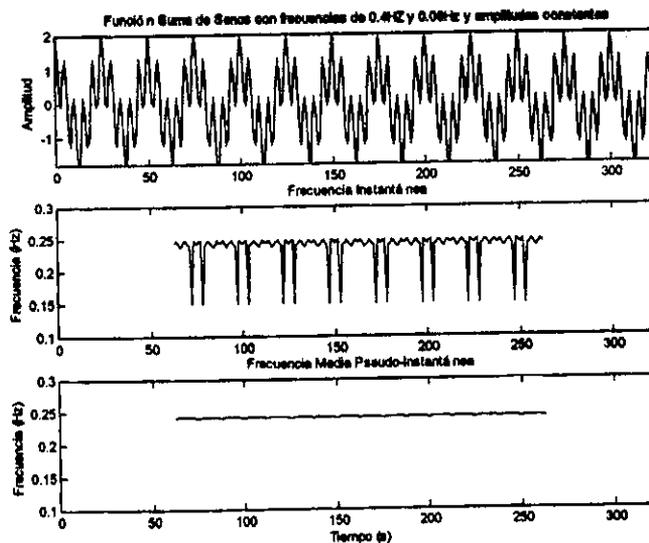


Figura FIG.4.8: a) Función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.4.9 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

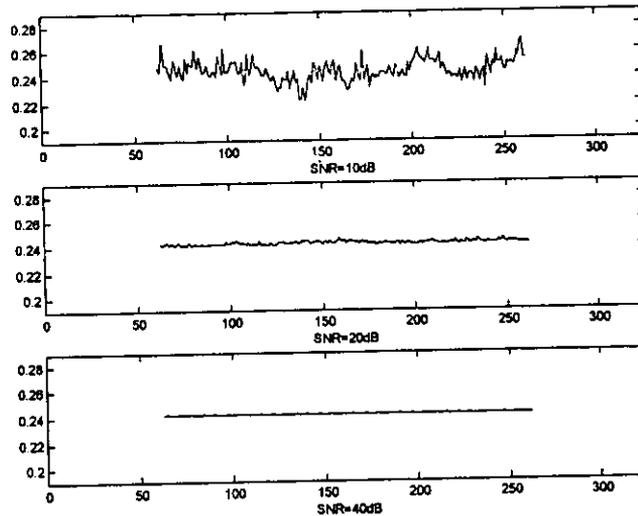


Figura FIG.4.9: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

4.7.4 Experimento 4: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.4.10 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 128 segundos.

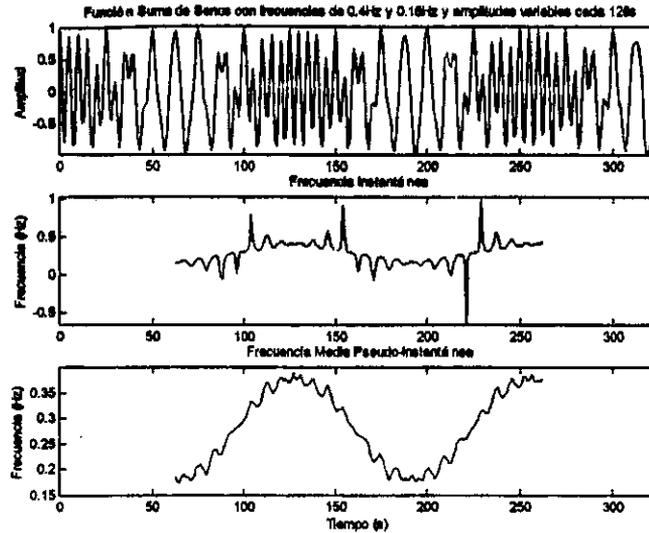


Figura FIG.4.10: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.4.11 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

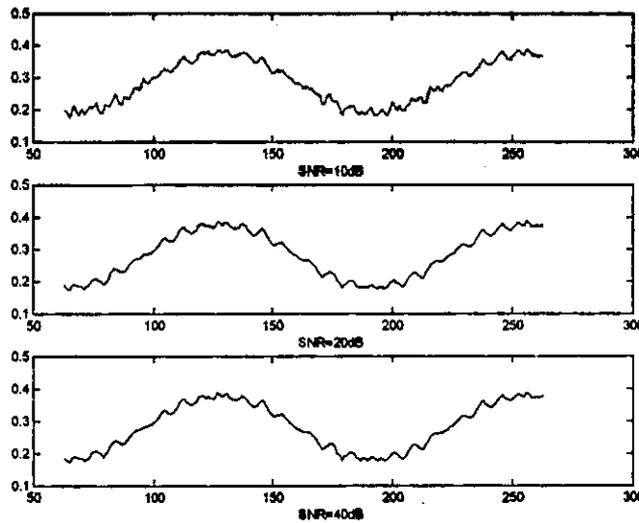


Figura FIG.4.11: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

4.7.5 Experimento 5: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.4.12 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y

0.16Hz y amplitudes variables cada 64 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 64 segundos.

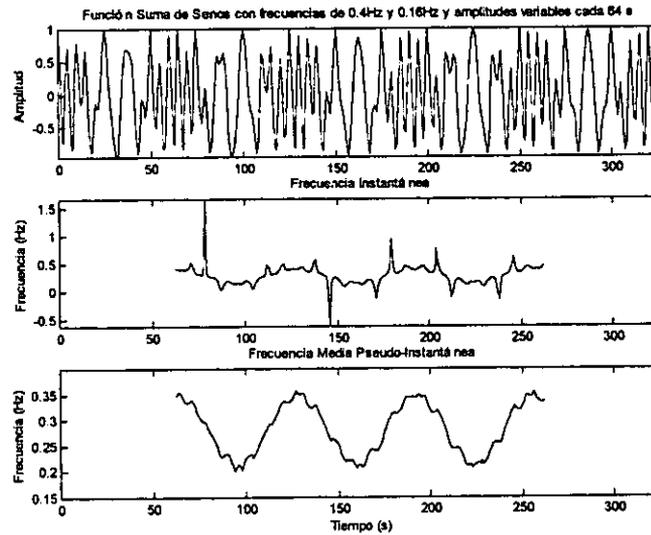


Figura FIG.4.12: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 64 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.4.13 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

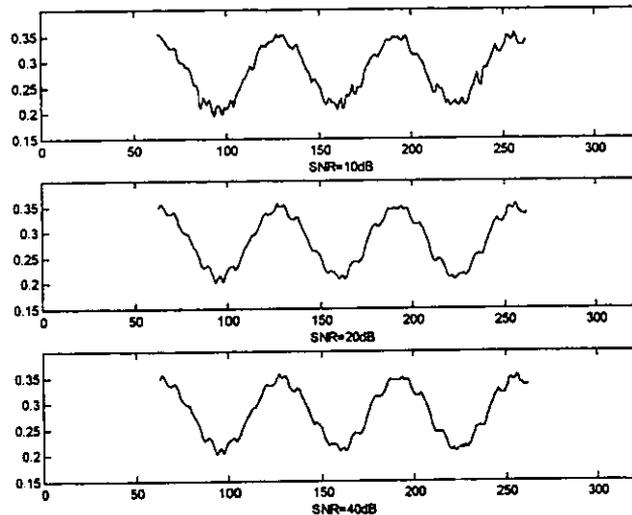


Figura FIG.4.13: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

4.7.6 Errores en la estimación de la frecuencia media pseudo instantánea en los experimentos 1 a 5

En la tabla TB.4.1 se muestra una relación del error promedio en la estimación de la frecuencia media pseudo instantánea con respecto a SNR, obtenido en 10 instancias de cada experimento para cada nivel de SNR.

SNR	Error en Frecuencia Media Pseudo Instantánea			
	Infinito	40dB	20dB	10dB
Experimento 1	0.002066	0.002069	0.002300	0.004001
Experimento 2	0.001849	0.001851	0.002319	0.004686
Experimento 3	0.001663	0.001667	0.002579	0.006351

TablaTB.4.1: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR.

4.7.7 Experimento 6: Función tren de pulsos de senos de diferentes frecuencias

En la figura FIG.4.14.a se muestra la gráfica de una función tren de pulsos de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.4.13). En la figura FIG.4.14.b se muestran diferentes vistas de esta distribución. Nótese que donde no existe señal en el tiempo, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.4.22); además, obsérvese su magnitud.

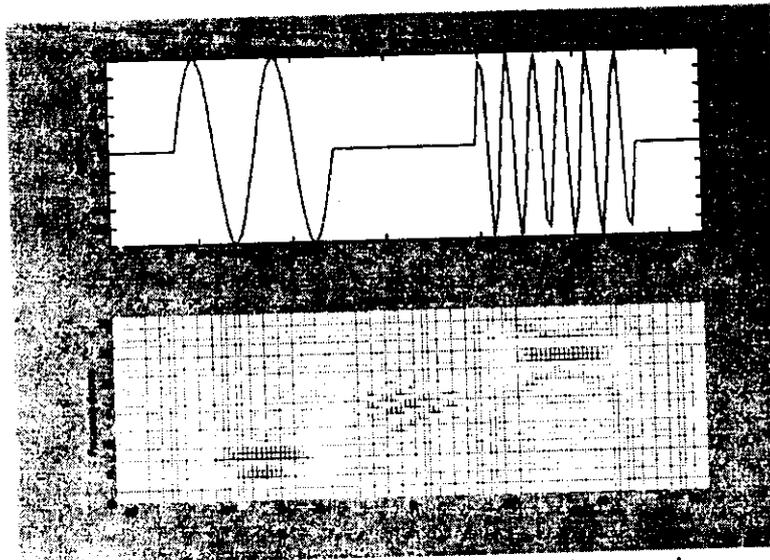


Figura FIG.4.14.a: Función tren de pulsos de senos de diferentes frecuencias y su distribución tiempo frecuencia de Wigner Ville.

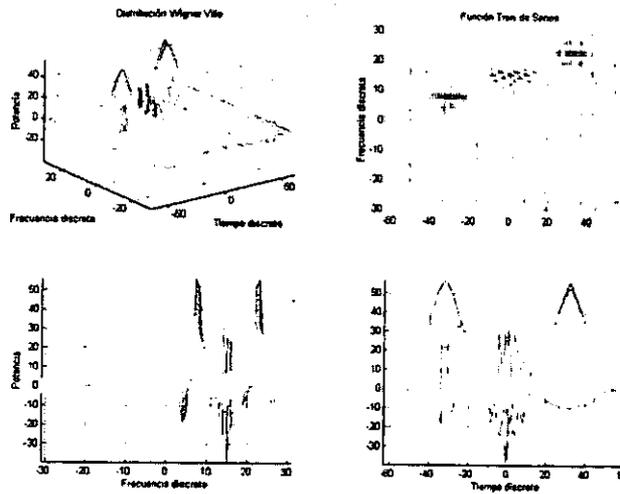


Figura FIG.4.14.b: Diferentes vistas de la distribución tiempo frecuencia de Wigner Ville de una función tren de pulsos de senos de diferentes frecuencias.

Además, en la figura FIG.4.15.a se muestra la gráfica de una función suma de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.4.13). En la figura FIG.4.15.b se muestran diferentes vistas de esta distribución. Nótese que donde no existe señal en la frecuencia, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.4.22); además, obsérvese su magnitud.

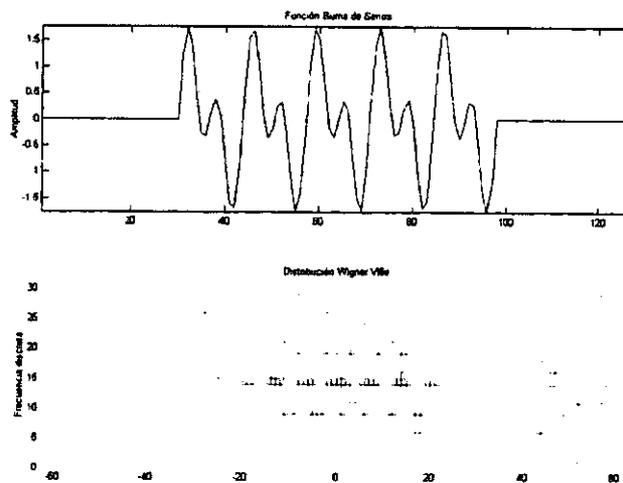


Figura FIG.4.15.a: Función suma de senos de diferentes frecuencias y su distribución tiempo frecuencia de Wigner Ville.

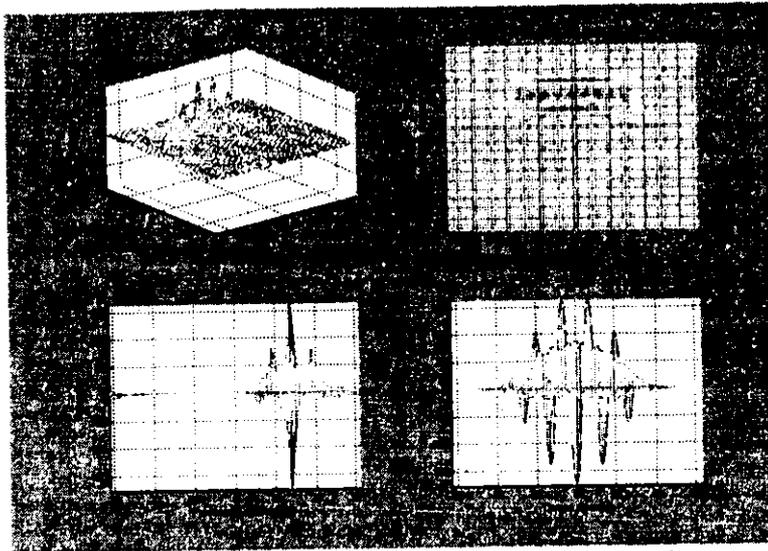


Figura FIG.4.15.b: Diferentes vistas de la distribución tiempo frecuencia de Wigner Ville de una función suma de senos de diferentes frecuencias.

4.8 Aplicación a una señal Dópler ultrasónica simulada en Matlab

En esta sección se presentarán diversos experimentos numéricos^[10] realizados en Matlab^{[32][33]} para mostrar gráfica y cuantitativamente el comportamiento de la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS de la señal, al variar la longitud de la ventana de muestreo (15, 31, 63, 127, 255 y 511 elementos), y al variar la relación de potencias de la señal y el ruido (SNR de 10dB, 20dB, 40dB e infinito). En el capítulo 2, Generalidades, se presenta la expresión para calcular la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presenta la expresión para calcular el ancho de banda espectral RMS (EQ.A4.1). En este mismo anexo se presentan las expresiones para calcular la relación señal a ruido (SNR) (EQ.A4.2), el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5) y el error en la estimación del ancho de banda espectral RMS (EQ.A4.6), (EQ.A4.7) y (EQ.A4.8). La señal tipo Dópler ultrasónica simulada que se considera en este análisis posee una frecuencia media descrita en la figura FIG.A3.2 del Anexo 3 y un ancho de banda espectral RMS constante de 100 Hz.

4.8.1 Longitud de ventana de muestreo de 15 elementos

En la figura FIG.4.16 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 15 elementos, sin presencia de ruido.

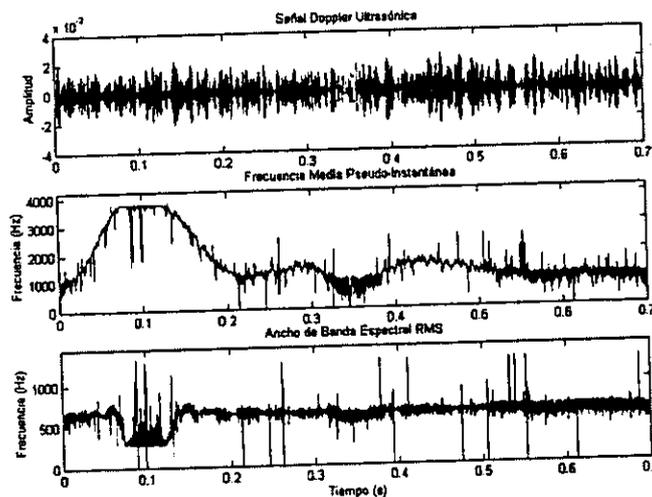


Figura FIG.4.16: Longitud de ventana de muestreo de 15 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.2 Longitud de ventana de muestreo de 31 elementos

En la figura FIG.4.17 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de 31 elementos, sin presencia de ruido.

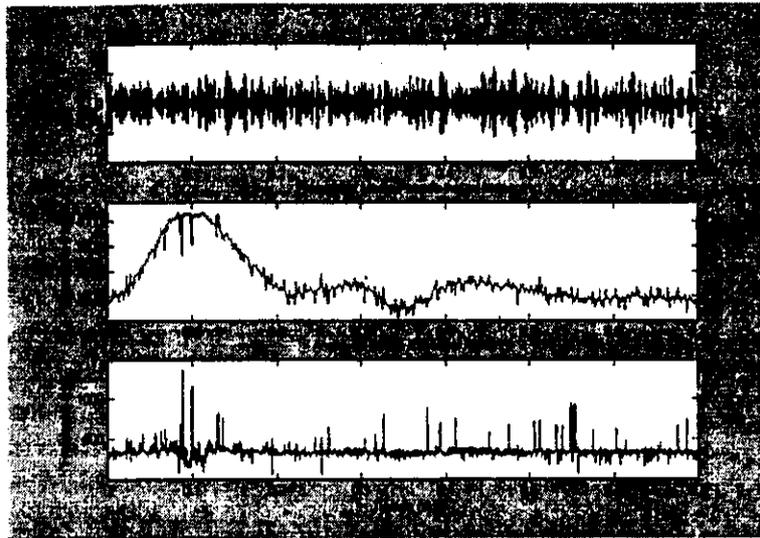


Figura FIG.4.17: Longitud de ventana de muestreo de 31 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.3 Longitud de ventana de muestreo de 63 elementos

En la figura FIG.4.18 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 63 elementos, sin presencia de ruido.

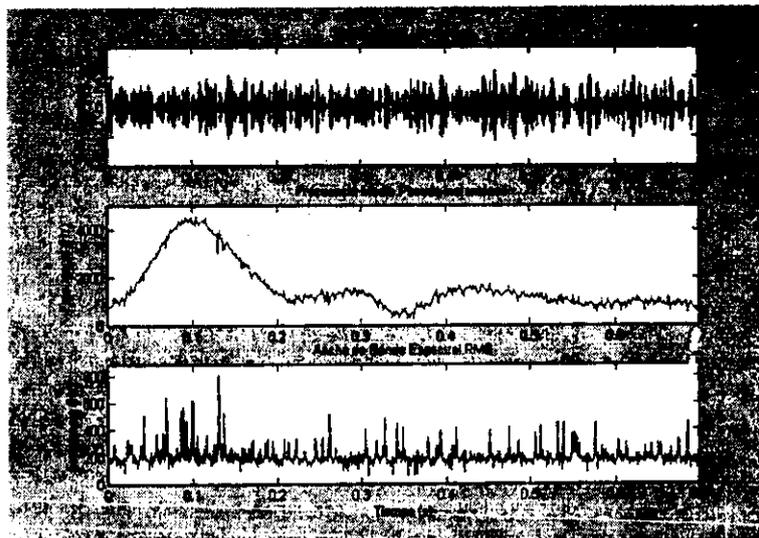


Figura FIG.4.18: Longitud de ventana de muestreo de 63 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.4 Longitud de ventana de muestreo de 127 elementos

En la figura FIG.4.19 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 127 elementos, sin presencia de ruido.

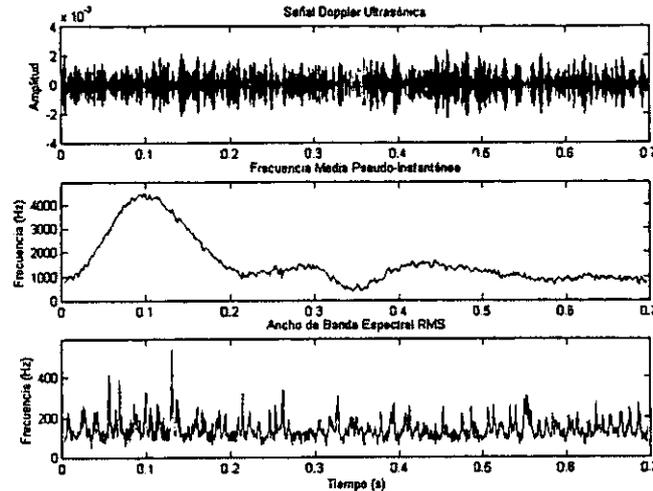


Figura FIG.4.19: Longitud de ventana de muestreo de 127 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.5 Longitud de ventana de muestreo de 255 elementos

En la figura FIG.4.20 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 255 elementos, sin presencia de ruido.

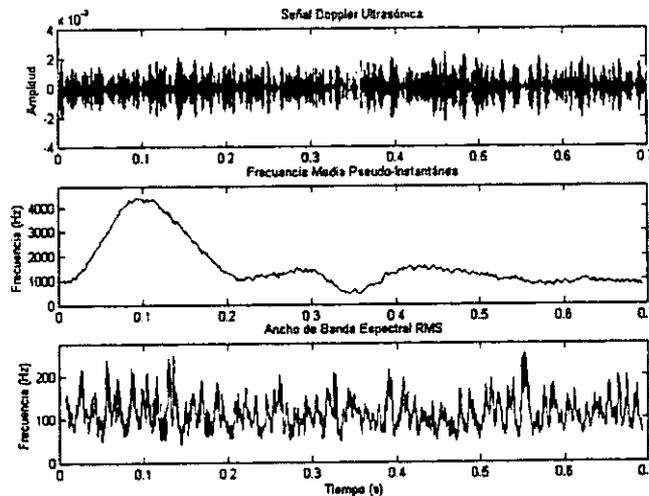


Figura FIG.4.20: Longitud de ventana de muestreo de 255 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.6 Longitud de ventana de muestreo de 511 elementos

En la figura FIG.4.21 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 511 elementos, sin presencia de ruido.

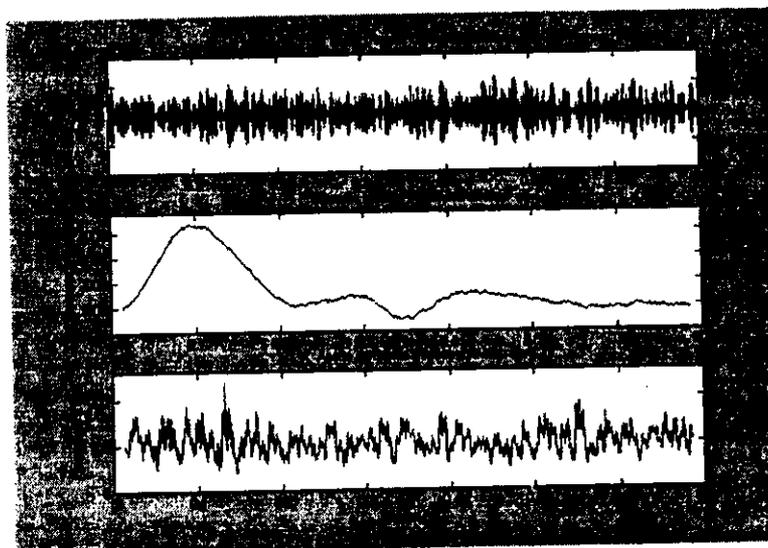


Figura FIG.4.21: Longitud de ventana de muestreo de 511 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

4.8.7 Errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS.

En la figura FIG.4.22 se muestra una gráfica de los errores en la estimación de la frecuencia media pseudo instantánea de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.4.2 se dan los valores respectivos.

Frecuencia Media Pseudo Instantánea	SNR				
	Ventana	10	20	40	infinito
15		152.08	71.77	62.94	62.93
31		211.83	48.04	34.77	34.74
63		325.21	52.00	35.43	35.39
127		544.27	56.07	36.98	37.01
255		905.77	68.02	40.11	40.13
511		1610.77	84.75	45.29	45.30

Tabla TB.4.2: Error en la estimación de la frecuencia media pseudo instantánea con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

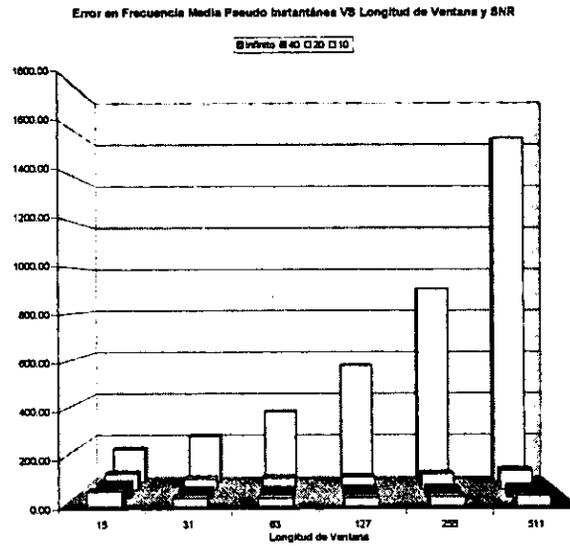


Figura FIG.4.22: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de la ventana de muestreo para una SNR de 10, 20, 40 e infinito.

En la figura FIG.4.23 se muestra una gráfica de los errores en la estimación del ancho de banda espectral RMS de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.4.3 se dan los valores respectivos.

Ancho de Banda Espectral RMS Ventana	SNR			
	10	20	40	infinito
15	624.22	535.97	521.97	521.94
31	536.98	292.52	239.52	239.20
63	629.633	220.876	103.188	101.649
127	798.040	237.718	42.589	39.449
255	1050.37	287.64	22.53	19.06
511	1579.21	382.57	17.84	15.31

Tabla TB.4.3: Error en la estimación del ancho de banda espectral RMS con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

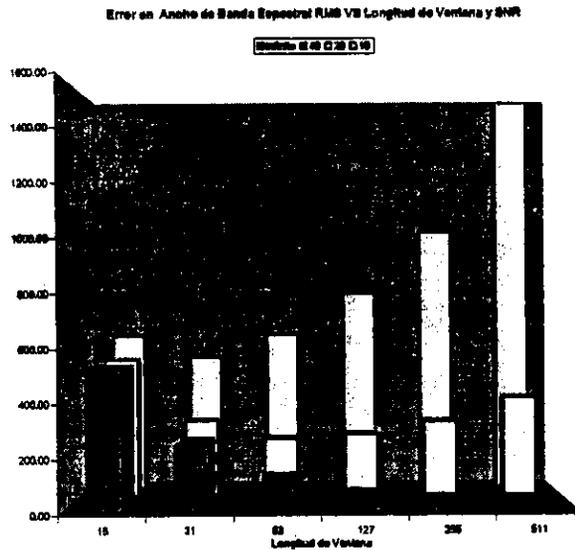


Figura FIG.4.23: Error en la estimación del ancho de banda espectral RMS contra la longitud de la ventana de muestreo para una SNR de 10,20, 40 e infinito.

4.9 Aplicación a una señal Dópler ultrasónica real en Matlab

En la figura FIG.4.24 se muestra la gráfica de una señal Dópler ultrasónica real, su frecuencia pseudo instantánea y su ancho de banda espectral RMS. La frecuencia de muestreo fue de 10kHz. La ventana de muestreo tiene una longitud de 127 elementos. La señal fue medida en el laboratorio de DISCA^[59].

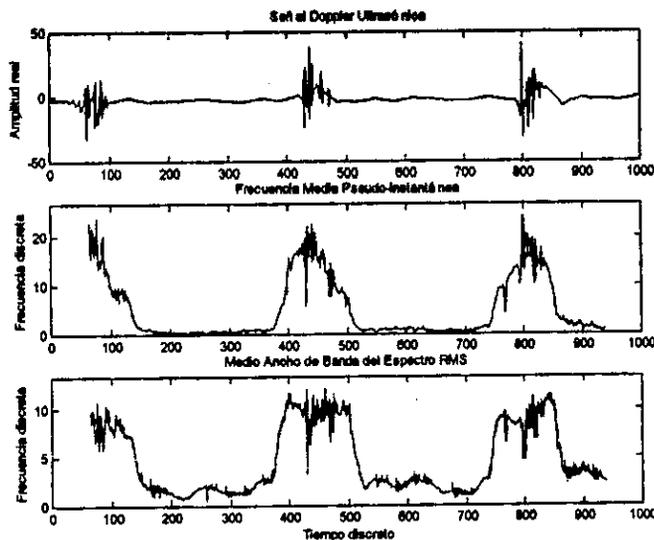


Figura FIG.4.24: Señal Dópler ultrasónica real, su frecuencia media pseudo instantánea y su ancho de banda espectral RMS.

5 DISTRIBUCIÓN DE CHOI WILLIAMS

5.1 Introducción

En este capítulo se plantea la distribución de Choi Williams a partir de la clase de distribuciones tiempo frecuencia de Cohen. Posteriormente se desarrolla esta distribución para el caso discreto. Finalmente se transforma la distribución discreta obtenida en una expresión más adecuada para implantar un algoritmo eficiente que la calcule. Además, se desarrollan experimentos numéricos en Matlab que muestran el comportamiento de la distribución discreta en función de la longitud de ventana de muestreo y en función de la relación señal a ruido de la señal muestreada, así como una aplicación a señales Dópler ultrasónicas. La bibliografía básica de este capítulo consiste en [16]; estudios con señales tipo Dópler ultrasónicas se encuentran en [10]; estudios comparativos entre diferentes distribuciones se encuentran en [4], [19] y [20].

Las expresiones principales de este capítulo son:

- Distribución de Choi Williams.

$$CWD(t, f) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{4\pi\tau^2/\sigma} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f\tau} d\tau$$

- Distribución de Choi Williams con una función ventana.

$$CWD_w(t, f) = \int_{-T_w}^{+T_w} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau$$

- Distribución discreta de Choi Williams.

$$DCWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-N+1}^{N-1} \frac{1}{4\pi\tau^2/\sigma} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+n+\tau) x^*(\mu+n-\tau)$$

- Transformación de la distribución discreta de Choi Williams.

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-N+1+\tau}^{N-1-\tau} \frac{1}{4\pi\tau^2/\sigma} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau) x^*(\mu-\tau) \right] - 2x(0)^2$$

5.2 Caso continuo

Se planteará la distribución tiempo frecuencia de Choi Williams ^[16] a partir de la clase de distribuciones de tiempo frecuencia de Cohen ^{[1][2]}.

Primero, el kernel de la distribución de Choi Williams es ^{[10][19]}:

$$\phi(\theta, \tau) = e^{-\frac{\tau^2 \theta^2}{\sigma}} \tag{EQ.5.1}$$

donde $\sigma > 0$ es un factor de escala. En la figura FIG.5.1 se muestra la gráfica de (EQ.5.1) en el dominio θ, τ para $\sigma=1$.

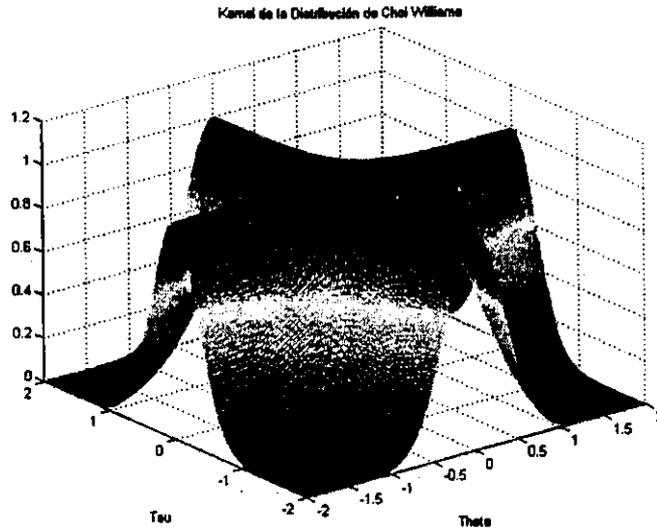


Figura FIG.5.1: Kernel (EQ.5.1) de la distribución tiempo frecuencia de Choi Williams para $\sigma=1$.

Segundo, el kernel del dominio de la autocorrelación (EQ.3.1) es:

$$\psi(t, \tau) = \int_{-\infty}^{+\infty} e^{-\frac{t^2 + \tau^2}{\sigma}} e^{-j\theta} d\theta \quad (\text{EQ.5.2})$$

Entonces, si se integra (EQ.5.2) se obtiene (A.5.1):

$$\psi(t, \tau) = \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{t^2}{4\tau^2/\sigma}} \quad (\text{EQ.5.3})$$

En la figura FIG.5.2 se muestra la gráfica de (EQ.5.3) en el dominio t, τ para $\sigma=1$. Tercero, la función de autocorrelación instantánea (EQ.3.2) es:

$$R_x(t, \tau) = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \quad (\text{EQ.5.4})$$

Cuarto, la función de autocorrelación indexada al tiempo generalizada (EQ.3.3) es:

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.5.5})$$

Entonces, si se reagrupa (EQ.5.5) se obtiene:

$$R'_x(t, \tau) = \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.5.6})$$

Finalmente, la distribución tiempo frecuencia (EQ.3.5) de Choi Williams es:

$$CWD(t, \omega) = \int_{-\infty}^{+\infty} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.5.7})$$

o bien, en función de la frecuencia:

$$CWD(t, f) = \int_{-\infty}^{+\infty} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f\tau} d\tau \quad (\text{EQ.5.8})$$

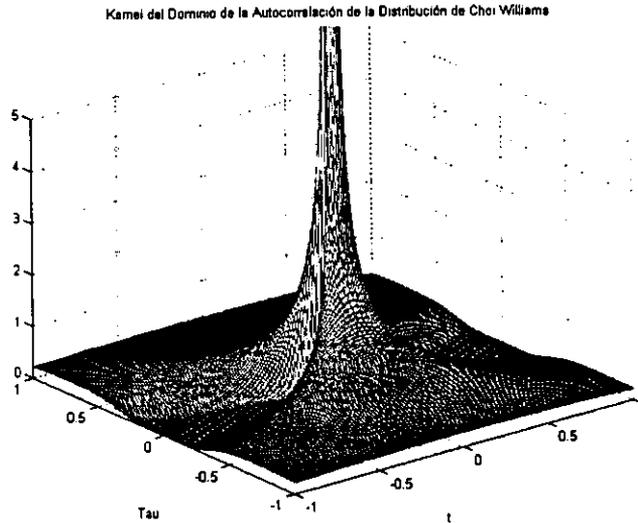


Figura FIG.5.2: Kernel del dominio de la autocorrelación (EQ.5.3) de la distribución tiempo frecuencia de Choi Williams para $\sigma=1$.

5.3 Caso continuo con ventana

Se desarrollará la distribución tiempo frecuencia de Choi Williams cuando se introduce una función ventana $W(t)$ que multiplica a la función de autocorrelación indexada al tiempo generalizada $R_x(t, \tau)$, es decir, a (EQ.5.6).

Si se introduce una función ventana $W(t)$ que cumpla con la propiedad de $W(t) = 0 \quad |t| > T_w/2$ en (EQ.5.8), entonces se obtiene [9]:

$$CWD_w(t, f) = \int_{-T_w/2}^{+T_w/2} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{\frac{(t-\mu)^2}{4\tau^2/\sigma}} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \quad (EQ.5.9)$$

5.4 Caso discreto con ventana

Se desarrollará la distribución tiempo frecuencia de Choi Williams discreta a partir de la distribución continua [9][10][16]. Esto es, se discretizará (EQ.5.9).

Primero, si en (EQ.5.9) se realizan los cambios de variables $\tau' = \tau/2$ y $\mu' = \mu - t$, entonces se obtiene:

$$CWD_w(t, f) = 2 \int_{-T_w/2}^{+T_w/2} \int_{-\infty}^{+\infty} \frac{1}{4\pi(2\tau')^2/\sigma} e^{-\frac{(\mu')^2}{4(2\tau')^2/\sigma}} x(\mu' + t + \tau') x^*(\mu' + t - \tau') d\mu' \cdot W(\tau') W^*(-\tau') e^{-j2\pi f(2\tau')} d\tau' \quad (EQ.5.10)$$

Segundo, si en (EQ.5.10) se discretiza la variable frecuencia $f = k\Delta f$, donde $\Delta f = \frac{1}{L} = \frac{1}{2N-1} \approx \frac{1}{2N}$ [8] y k es un índice discreto, siendo L el número de muestras por ventana, y se discretiza la variable tiempo $t = n\Delta T$, $\mu' = \mu\Delta T$ y $\tau' = \tau\Delta T$, donde n, μ y τ son índices discretos, además de redefinir la constante $\sigma = \sigma' / 4$, entonces se obtiene:

$$DCWD(n\Delta T, k\Delta f) = 2 \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi(\tau\Delta T)^2/\sigma}} e^{-\frac{(-\mu)^2}{4(\tau\Delta T)^2/\sigma}} x(\mu\Delta T + n\Delta T + \tau\Delta T)\Delta T \quad (EQ.5.11)$$

$$\cdot x^*(\mu\Delta T + n\Delta T - \tau\Delta T)W(\tau\Delta T)W^*(-\tau\Delta T)e^{-\frac{j4\mu\tau\Delta T}{2N}\Delta T}$$

Tercero, si en (EQ.5.11) se normaliza el incremento en tiempo $\Delta T=1$ y el incremento en frecuencia $\Delta f=1$, entonces se obtiene:

$$DCWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{(-\mu)^2}{4\tau^2/\sigma}} x(\mu+n+\tau)x^*(\mu+n-\tau)w(\tau)w^*(-\tau)e^{-\frac{j2\mu\tau}{N}} \quad (EQ.5.12)$$

Finalmente, la distribución tiempo frecuencia de Choi Williams discreta es:

$$DCWD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau)W^*(-\tau)e^{-\frac{j2\mu\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+n+\tau)x^*(\mu+n-\tau) \quad (EQ.5.13)$$

en donde si $\tau=0$ entonces $\sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+n+\tau)x^*(\mu+n-\tau) = x(n)x^*(n)$.

5.5 Análisis de la distribución

Se analizará la efectividad del kernel (EQ.5.1) de la distribución de Choi Williams (EQ.5.7) en la disminución de los efectos de interferencia entre los diferentes componentes de una señal^[16].

Sea la señal multicomponente $x(t)$ la cual se puede representar como la suma de sus componentes unitarios:

$$x(t) = \sum_{n=1}^N A_n e^{j(\omega_n t + \theta_n)} \quad (EQ.5.14)$$

Si se sustituye (EQ.5.14) en (EQ.5.7) se obtiene:

$$CWD(t, \omega) = \int_{-\infty}^{+\infty} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} \sum_{n=1}^N A_n e^{j(\omega_n(\mu+\frac{\tau}{2})+\theta_n)} \sum_{m=1}^N A_m e^{-j(\omega_m(\mu-\frac{\tau}{2})+\theta_m)} d\mu e^{-j\omega\tau} d\tau \quad (EQ.5.15)$$

Reagrupando términos en (EQ.5.15) se obtiene:

$$CWD(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j(\frac{\omega_n + \omega_m}{2}\tau)} \int_{-\infty}^{+\infty} \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} e^{j(\omega_n - \omega_m)\mu} d\mu e^{-j\omega\tau} d\tau \quad (EQ.5.16)$$

Si se aplica la propiedad asociativa de la suma en EQ.5.16, entonces se obtiene:

$$CWD(t, \omega) = CWD_{\text{auto}}(t, \omega) + CWD_{\text{cruce}}(t, \omega) \quad (EQ.5.17)$$

donde, para el caso en el cual los índices de las sumatorias en (EQ.5.16) son iguales, es decir, $n=m$, se tiene que:

$$CWD_{\text{auto}}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n\tau} \int_{-\infty}^{+\infty} \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} d\mu e^{-j\omega\tau} d\tau \quad (EQ.5.18)$$

y para el caso en el cual los índices de las sumatorias en (EQ.5.16) son diferentes, es decir, $n \neq m$, se tiene que:

$$CWD_{\text{cruce}}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j(\frac{\omega_n + \omega_m}{2}\tau)} \int_{-\infty}^{+\infty} \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} e^{j(\omega_n - \omega_m)\mu} d\mu e^{-j\omega\tau} d\tau \quad (EQ.5.19)$$

Primero se analizará (EQ.5.18), la cual se constituye de los auto-términos de la distribución tiempo frecuencia, esto es, de la distribución de cada componente. Luego se analizará (EQ.5.19), la cual se constituye de los términos de cruce de la distribución tiempo frecuencia, esto es, de la distribución de la interacción de cada par de componentes diferentes.

Referente a (EQ.5.18) o auto-términos. La integral con respecto a μ es^(A.5.2):

$$\int_{-\infty}^{+\infty} \sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} d\mu = 2\pi \quad (\text{EQ.5.20})$$

Sustituyendo (EQ.5.20) en (EQ.5.18) se obtiene:

$$CWD_{\text{auto}}(t, \omega) = \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} e^{-j\omega \tau} d\tau \quad (\text{EQ.5.21})$$

Si se integra (EQ.5.21) se obtiene^(A.5.3):

$$CWD_{\text{auto}}(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) \quad (\text{EQ.5.22})$$

Referente a (EQ.5.19) o términos de cruce. Para integrar con respecto a μ se aplica la ecuación de Párseval^(A.5.4). Esto es:

$$\int_{-\infty}^{+\infty} \left[\sqrt{\frac{\pi}{\tau^2/\sigma}} e^{-\frac{(t-\mu)^2}{4\tau^2/\sigma}} \right] \left[e^{j(\omega_n - \omega_m)\mu} \right] d\mu = \int_{-\infty}^{+\infty} \left[2\pi e^{-j\mu} e^{-\frac{\tau^2 \mu^2}{\sigma}} \right] \left[\delta(\mu + (\omega_n - \omega_m)) \right] d\mu \quad (\text{EQ.5.23})$$

Integrando (EQ.5.23) se obtiene^(A.5.5):

$$\int_{-\infty}^{+\infty} \left[2\pi e^{-j\mu} e^{-\frac{\tau^2 \mu^2}{\sigma}} \right] \left[\delta(\mu + (\omega_n - \omega_m)) \right] d\mu = 2\pi e^{-\frac{(\omega_n - \omega_m)^2}{\sigma} \tau^2} e^{j(\omega_n - \omega_m)t} \quad (\text{EQ.5.24})$$

Si se sustituye (EQ.5.24) en (EQ.5.19) se obtiene:

$$CWD_{\text{cruce}}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} 2\pi e^{-\frac{(\omega_n - \omega_m)^2}{\sigma} \tau^2} e^{j(\omega_n - \omega_m)t} e^{-j\omega \tau} d\tau \quad (\text{EQ.5.25})$$

Reagrupando términos en (EQ.5.25) se obtiene:

$$CWD_{\text{cruce}}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j((\omega_n - \omega_m)t + \theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{-\frac{(\omega_n - \omega_m)^2}{\sigma} \tau^2} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} e^{-j\omega \tau} d\tau \quad (\text{EQ.5.26})$$

Si se integra (EQ.5.26) se obtiene^(A.5.6):

$$CWD_{\text{cruce}}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j((\omega_n - \omega_m)t + \theta_n - \theta_m)} \sqrt{\frac{\pi\sigma}{(\omega_n - \omega_m)^2}} e^{-\frac{\sigma}{4(\omega_n - \omega_m)^2} \left(\omega - \frac{\omega_n + \omega_m}{2}\right)^2} \quad (\text{EQ.5.27})$$

Puesto que la distribución es real, entonces:

$$CWD_{\text{cruce}}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \sqrt{\frac{\pi\sigma}{(\omega_n - \omega_m)^2}} e^{-\frac{\sigma}{4(\omega_n - \omega_m)^2} \left(\omega - \frac{\omega_n + \omega_m}{2}\right)^2} \quad (\text{EQ.5.28})$$

Finalmente:

$$CWD(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) + \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \sqrt{\frac{\pi\sigma}{(\omega_n - \omega_m)^2}} e^{-\frac{\sigma}{4(\omega_n - \omega_m)^2} \left(\omega - \frac{\omega_n + \omega_m}{2}\right)^2} \quad (\text{EQ.5.29})$$

El factor de peso en (EQ.5.28) es:

$$FP_{CWD}(\omega) = \frac{\pi\sigma}{(\omega_n - \omega_m)^2} e^{-\frac{\sigma}{4(\omega_n - \omega_m)^2} \left(\omega - \frac{\omega_n + \omega_m}{2}\right)^2} \quad (EQ.5.30)$$

En la figura FIG.5.3a se muestran gráficas de (EQ.5.30) para diversos valores de ω_n y ω_m para una σ constante; mientras que en la figura FIG.5.3b, para diversos valores de σ para unos ω_n y ω_m constantes.

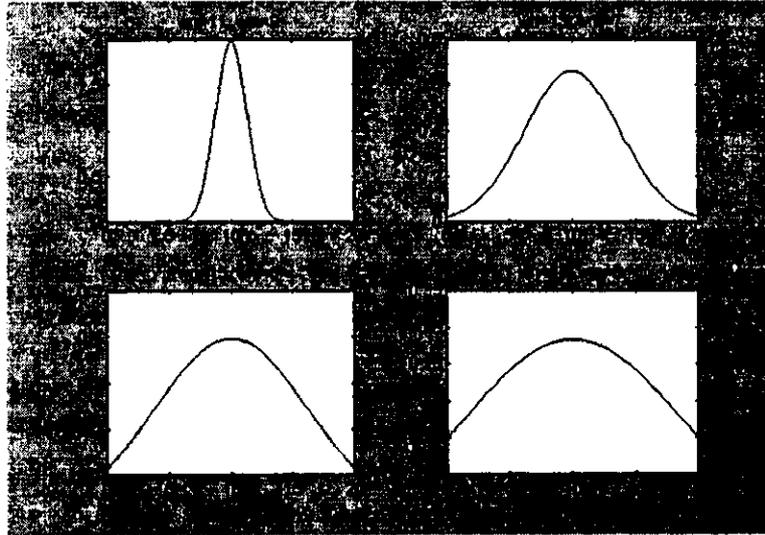


Figura FIG.5.3a: Factor de peso (EQ.5.30) de la distribución tiempo frecuencia de Choi Williams para diversos valores de ω_n y ω_m con $\sigma = 5$.

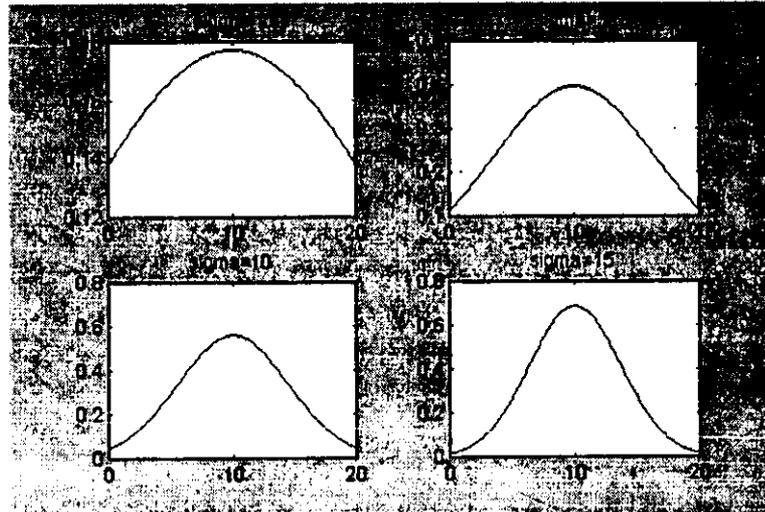


Figura FIG.5.3b: Factor de peso (EQ.5.30) de la distribución tiempo frecuencia de Choi Williams para diversos valores de σ con $\omega_n = 15$ y $\omega_m = 5$.

5.6 Algoritmo eficiente

El algoritmo eficiente calcula (EQ.5.13) valuada en $n = 0$ [9][17]; por consiguiente:

$$DCWD(0, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{(-\mu)^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau) \quad (\text{EQ.5.31})$$

Si se separa (EQ.5.31) en dos sumatorias, una para los índices positivos y otra para los índices negativos de τ , se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=-N+1}^{-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau) \quad (\text{EQ.5.32})$$

$$+ 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau)$$

Si a (EQ.5.32) se suma y se resta el término $2W(0)W^*(0)x(0)x^*(0)$ correspondiente al índice $\tau=0$, se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=-N+1}^0 W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau) \quad (\text{EQ.5.33})$$

$$+ 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau)$$

$$- 2W(0)W^*(0)x(0)x^*(0)$$

Si en la primera sumatoria de (EQ.5.33) se cambia el índice τ por $-\tau$, y considerando que $W(0)=1$, se obtiene:

$$DCWD(0, k) = 2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi(-\tau)^2/\sigma}} e^{-\frac{\mu^2}{4(-\tau)^2/\sigma}} x(\mu-\tau)x^*(\mu+\tau) \quad (\text{EQ.5.34})$$

$$+ 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau)$$

$$- 2x(0)x^*(0)$$

Si en el primera sumatoria de (EQ.5.34) se aplica el conjugado complejo del conjugado complejo, se obtiene:

$$DCWD(0, k) = \left(2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu-\tau)x^*(\mu+\tau) \right)^{**} \quad (\text{EQ.5.35})$$

$$+ 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau)$$

$$- 2x(0)x^*(0)$$

Si en la primera sumatoria de (EQ.5.35) se calcula uno de los conjugados complejos, se obtiene:

$$DCWD(0, k) = \left(2 \sum_{\tau=0}^{N-1} W^*(-\tau)W(\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x^*(\mu-\tau)x(\mu+\tau) \right) \quad (\text{EQ.5.36})$$

$$+ 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau)x^*(\mu-\tau)$$

$$- 2x(0)x^*(0)$$

Entonces, reagrupando términos se tiene^(A.5.7):

$$DCWD(0, k) = 2 \operatorname{Re} \left[2 \sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{\frac{j2\pi k \tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau) x^*(\mu-\tau) \right] - 2x(0)x^*(0) \quad (\text{EQ.5.37})$$

Finalmente se tiene que:

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{\frac{j2\pi k \tau}{N}} \sum_{\mu=-M}^M \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu+\tau) x^*(\mu-\tau) \right] - 2|x(0)|^2 \quad (\text{EQ.5.38})$$

5.6.1 Optimización del índice μ

Se analizará el caso particular en el cual la longitud de ventana es 7. En consecuencia, los elementos del vector X se enumeran desde -3 hasta 3, a saber:

Elemento:	$X(-6)$	$X(-5)$	$X(-4)$	$X(-3)$	$X(-2)$	$X(-1)$	$X(0)$	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$
Valor:	0	0	0	X_3	X_2	X_1	X_0	X_1	X_2	X_3	0	0	0

donde los elementos enumerados anteriores al -3 y posteriores al 3 tienen un valor de cero.

Las casillas sombreadas representan una multiplicación trivial cuyo resultado es cero.

Primero, si $\tau = 0$, se tiene:

μ	-6	-5	-4	-3	-2	-1	0	1	2	3	
$X(\mu+\tau)$	$X(-6)$	$X(-5)$	$X(-4)$	$X(-3)$	$X(-2)$	$X(-1)$	$X(0)$	$X(1)$	$X(2)$	$X(3)$	
$X^*(\mu-\tau)$	$X^*(6)$	$X^*(5)$	$X^*(4)$	$X^*(3)$	$X^*(2)$	$X^*(1)$	$X^*(0)$	$X^*(1)$	$X^*(2)$	$X^*(3)$	

Si $\tau = 1$, se tiene:

μ	-6	-5	-4	-3	-2	-1	0	1	2	
$X(\mu+\tau)$	$X(-7)$	$X(-6)$	$X(-5)$	$X(-4)$	$X(-3)$	$X(-2)$	$X(-1)$	$X(0)$	$X(1)$	
$X^*(\mu-\tau)$	$X^*(-6)$	$X^*(-4)$	$X^*(-3)$	$X^*(-2)$	$X^*(-1)$	$X^*(0)$	$X^*(1)$	$X^*(2)$	$X^*(3)$	

Si $\tau = 2$, se tiene:

μ	-6	-5	-4	-3	-2	-1	0	1	2	
$X(\mu+\tau)$	$X(-8)$	$X(-7)$	$X(-6)$	$X(-5)$	$X(-4)$	$X(-3)$	$X(-2)$	$X(-1)$	$X(0)$	
$X^*(\mu-\tau)$	$X^*(-4)$	$X^*(-3)$	$X^*(-2)$	$X^*(-1)$	$X^*(0)$	$X^*(1)$	$X^*(2)$	$X^*(3)$	$X^*(4)$	

Finalmente, si $\tau = 3$, se tiene:

μ	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
$X(\mu+\tau)$	$X(-9)$	$X(-8)$	$X(-7)$	$X(-6)$	$X(-5)$	$X(-4)$	$X(-3)$	$X(-2)$	$X(-1)$	$X(0)$	$X(1)$	$X(2)$	$X(3)$
$X^*(\mu-\tau)$	$X^*(-3)$	$X^*(2)$	$X^*(1)$	$X^*(0)$	$X^*(1)$	$X^*(2)$	$X^*(3)$	$X^*(4)$	$X^*(5)$	$X^*(6)$	$X^*(7)$	$X^*(8)$	$X^*(9)$

En conclusión, en un primer análisis se observa que el valor máximo y mínimo del índice μ que en general no produce multiplicaciones triviales es -3 y 3 respectivamente. En un análisis más detallado se observa que el valor máximo y mínimo del índice μ para un valor de τ que no produce multiplicaciones triviales es $3-|\tau|$ y $-3+|\tau|$, donde $|\tau|$ significa el valor absoluto de τ .

Generalizando este resultado. Si la longitud de ventana es L , donde $L=2^p-1$, entonces $N=(L+1)/2$. EL valor máximo y mínimo del índice μ en general que no produce multiplicaciones triviales es $N-1$ y $-N+1$ respectivamente. El valor máximo y mínimo del índice μ para un valor de τ que no produce multiplicaciones triviales es $N-1-|\tau|$ y $-N+1+|\tau|$, donde $|\tau|$ significa el valor absoluto de τ . De este modo, (EQ.5.38) se transforma a:

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-N+1+\tau}^{N-1-\tau} \frac{1}{4\pi\tau^2} e^{-\frac{\mu^2}{4\tau^2}\sigma} x(\mu+\tau)x^*(\mu-\tau) \right] - 2x(0)^2 \quad (\text{EQ.5.39})$$

5.6.2 Operaciones realizadas

El número de operaciones que se realizan en el cálculo de la distribución discreta de Choi Williams según las ecuaciones (EQ.5.13), considerando $M = N - 1$, y (EQ.5.39) para cada k es:

	(EQ. 5.13)	(EQ. 5.39)
Multiplicaciones complejas	$8N^2 - 4N$	$2N^2 - 2N + 1$
Sumas complejas	$4N^2 - 6N + 2$	$N^2 - 2N$
Multiplicaciones por escalares	1	2

N es la longitud de la ventana de muestreo. Se considera a $W(\tau)W^*(-\tau)$ como un solo factor;

la misma consideración para $\frac{1}{4\pi\tau^2} e^{-\frac{\mu^2}{4\tau^2}\sigma}$.

5.7 Experimentos Numéricos en Matlab

En esta sección se presentarán diversos experimentos numéricos^[1] realizados en Matlab^{[32][33]} para comparar gráficamente el comportamiento de la frecuencia instantánea con el comportamiento de la frecuencia media pseudo instantánea y para cuantificar éste último, al variar el nivel de la potencia del ruido en la señal (SNR). En el capítulo 2, Generalidades, se presentan las expresiones para calcular la frecuencia instantánea discreta (EQ.2.19) y la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presentan las expresiones para calcular la relación señal a ruido SNR (EQ.A4.2) y el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5). Para todos los experimentos, la longitud de ventana de muestreo es de 127 elementos.

5.7.1 Experimento 1: Función seno con frecuencia de 0.32Hz y amplitud constante

En la figura FIG.5.4 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función seno con frecuencia constante de 0.32 Hz y amplitud constante. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.32 Hz.

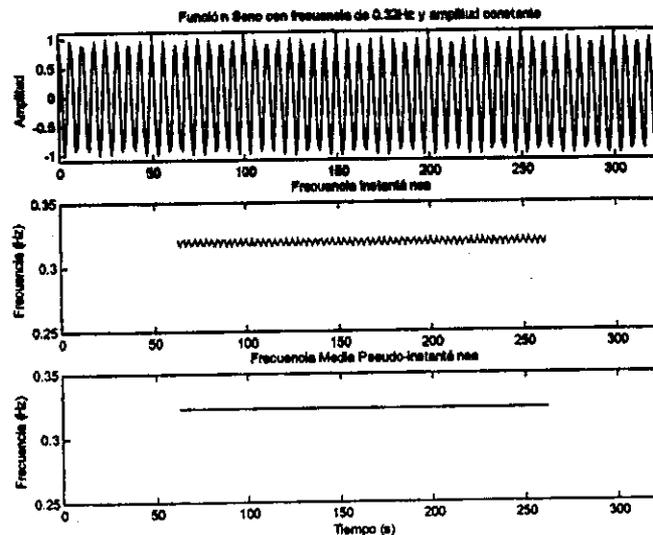


Figura FIG.5.4: a) Función seno con frecuencia de 0.32Hz y amplitud constante; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.5.5 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

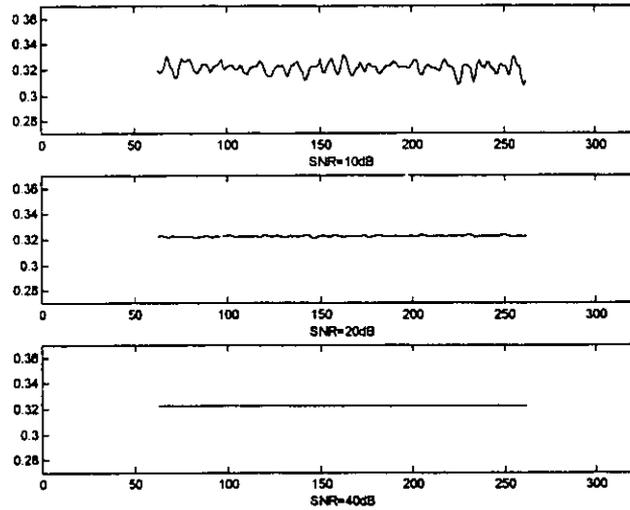


Figura FIG.5.5: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

5.7.2 Experimento 2: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes

En la figura FIG.5.6 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.28 Hz.

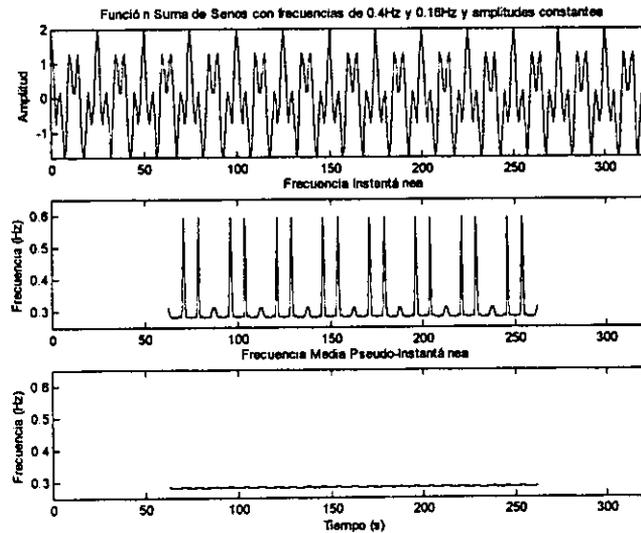


Figura FIG.5.6: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.5.7 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

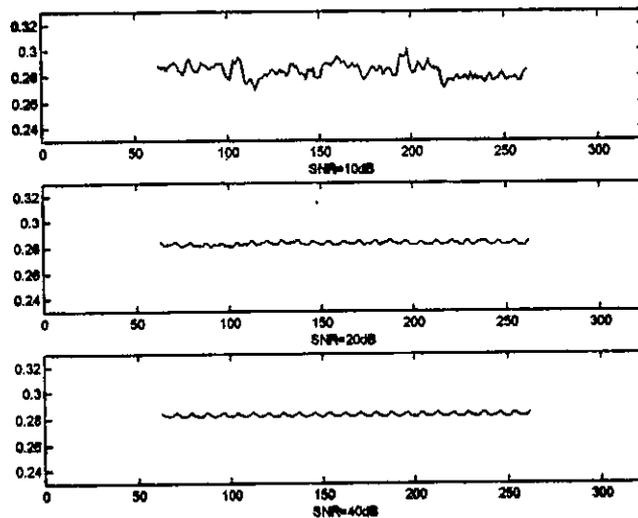


Figura FIG.5.7: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

5.7.3 Experimento 3: Función Suma de Senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes

En la figura FIG.5.8 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.24 Hz.

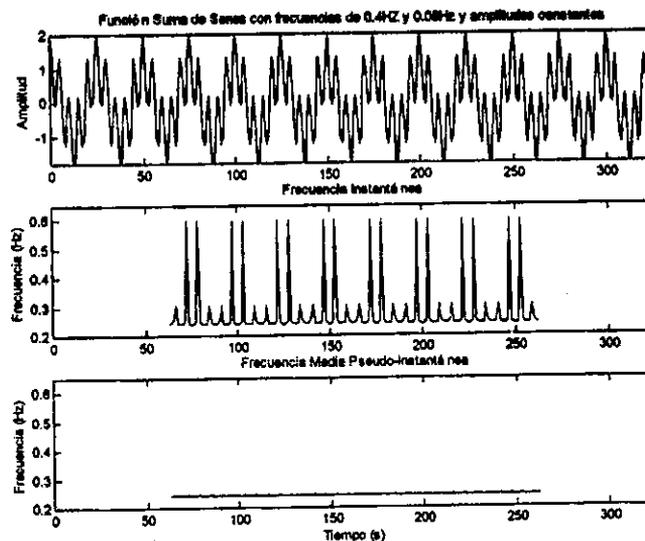


Figura FIG.5.8: a) Función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.5.9 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

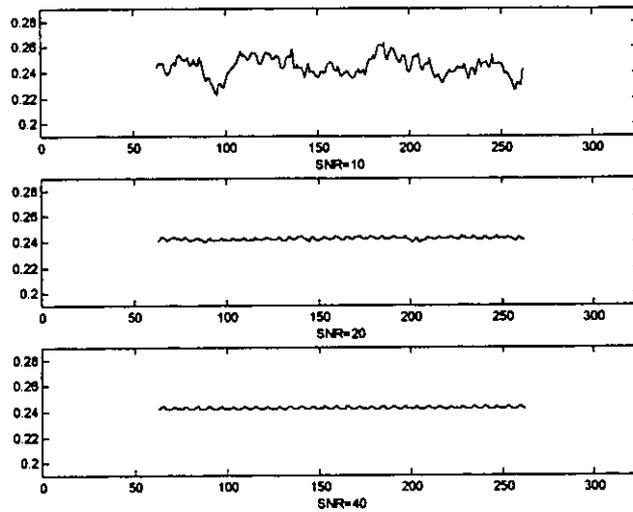


Figura FIG.5.9: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

5.7.4 Experimento 4: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.5.10 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 128 segundos.

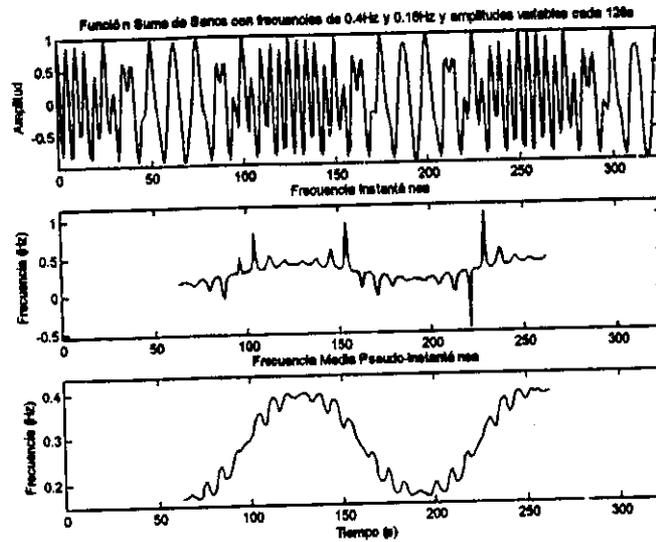


Figura FIG.5.10: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.5.11 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

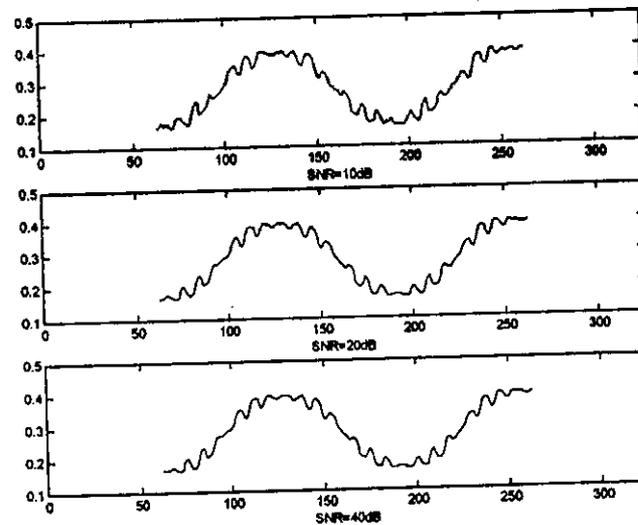


Figura FIG.5.11: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

5.7.5 Experimento 5: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.5.12 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y

0.16Hz y amplitudes variables cada 64 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 64 segundos.

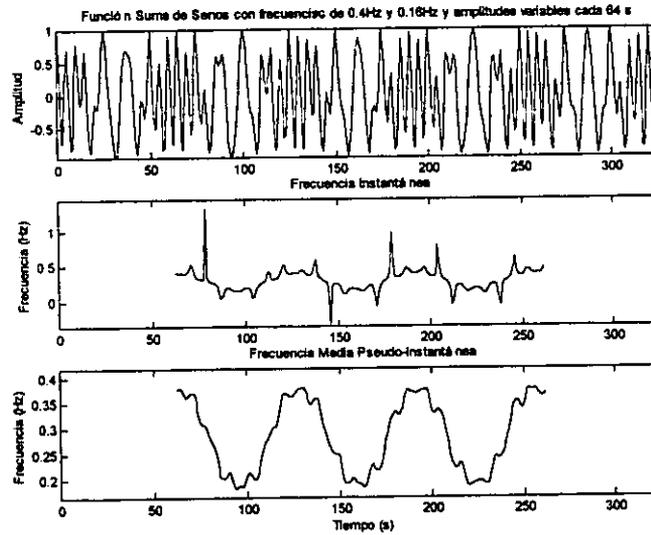


Figura FIG.5.12: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 64 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.5.13 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

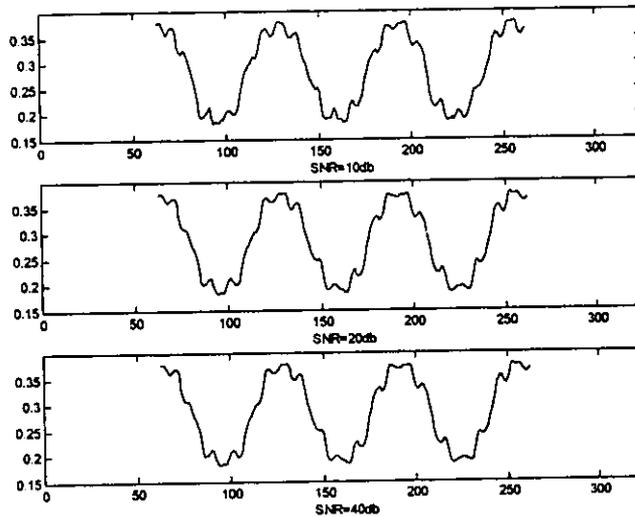


Figura FIG.5.13: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

5.7.6 Errores en la estimación de la frecuencia media pseudo instantánea en los experimentos 1 a 5

En la tabla TB.5.1 se muestra una relación del error promedio en la estimación de la frecuencia media pseudo instantánea con respecto a SNR, obtenido en 10 instancias de cada experimento para cada nivel de SNR.

SNR	Error en Frecuencia Media Pseudo Instantánea			
	Infinito	40dB	20dB	10dB
Experimento 1	0.002066	0.002069	0.002280	0.003170
Experimento 2	0.001849	0.001850	0.002167	0.004905
Experimento 3	0.001663	0.001667	0.002337	0.008053

Tabla TB.5.1: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR.

5.7.7 Experimento 6: Función tren de pulsos de senos de diferentes frecuencias

En la figura FIG.5.14.a se muestra la gráfica de una función tren de pulsos de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.5.13). En la figura FIG.5.14.b se muestran diferentes vistas de esta distribución. Nótese que donde no existe señal en el tiempo, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.5.28); además, obsérvese su magnitud.

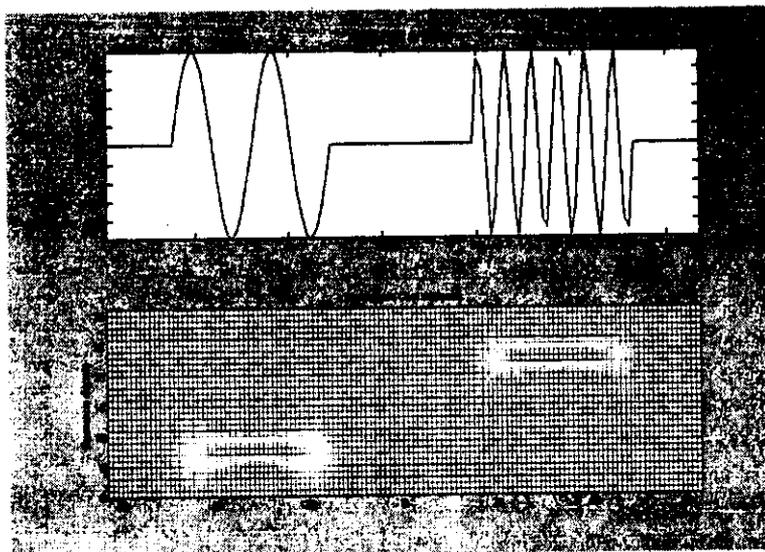


Figura FIG.5.14.a: Función tren de pulsos de senos de diferentes frecuencias y su distribución tiempo frecuencia de Choi Williams.

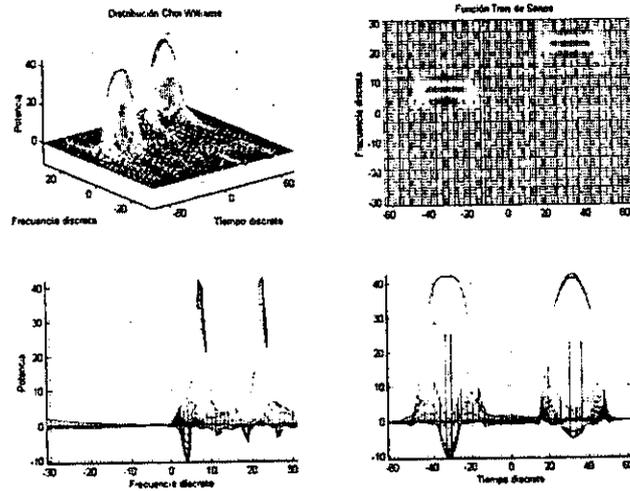


Figura FIG.5.14.b: Diferentes vistas de la distribución tiempo frecuencia de Choi Williams de una función tren de pulsos de senos de diferentes frecuencias.

Además, en la figura FIG.5.15.a se muestra la gráfica de una función suma de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.5.13). En la figura FIG.5.15.b se muestran diferentes vistas de esta distribución. Nótese que donde no existe señal en la frecuencia, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.5.28); además, obsérvese su magnitud.

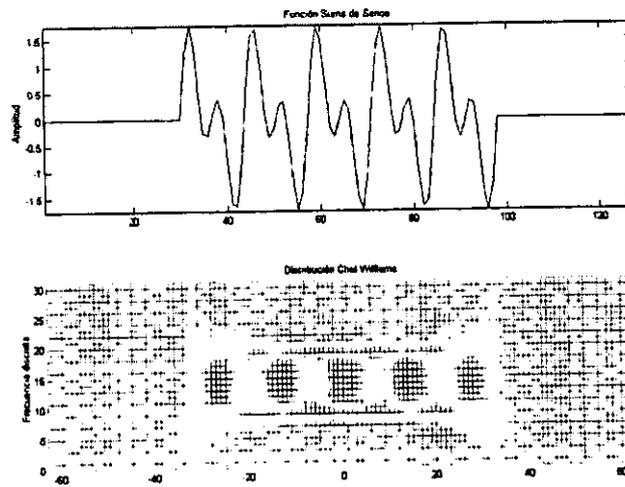


Figura FIG.5.15.a: Función suma de senos de diferentes frecuencias y su distribución tiempo frecuencia de Choi Williams.

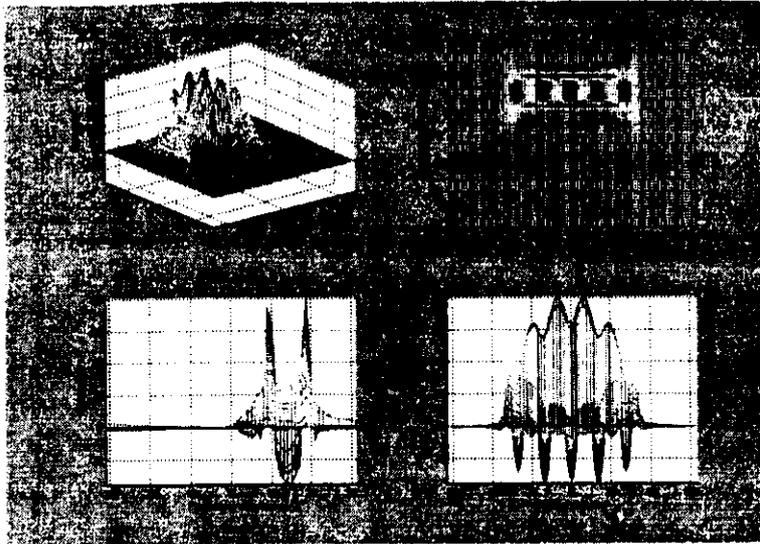


Figura FIG.5.15.b: Diferentes vistas de la distribución tiempo frecuencia de Choi Williams de una función suma de senos de diferentes frecuencias.

5.8 Aplicación a una señal Dópler ultrasónica simulada en Matlab

En esta sección se presentarán diversos experimentos numéricos^[10] realizados en Matlab^{[32][33]} para mostrar gráfica y cuantitativamente el comportamiento de la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS de la señal, al variar la longitud de la ventana de muestreo (15, 31, 63, 127, 255 y 511 elementos), y al variar la relación de potencias de la señal y el ruido (SNR de 10dB, 20dB, 40dB e infinito). En el capítulo 2, Generalidades, se presenta la expresión para calcular la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presenta la expresión para calcular el ancho de banda espectral RMS (EQ.A4.1). En este mismo anexo se presentan las expresiones para calcular la relación señal a ruido (SNR) (EQ.A4.2), el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5) y el error en la estimación del ancho de banda espectral RMS (EQ.A4.6), (EQ.A4.7) y (EQ.A4.8). La señal tipo Dópler ultrasónica simulada que se considera en este análisis posee una frecuencia media descrita en la figura FIG.A3.2 del Anexo 3 y un ancho de banda espectral RMS constante de 100 Hz.

5.8.1 Longitud de ventana de muestreo de 15 elementos

En la figura FIG.5.16 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 15 elementos, sin presencia de ruido.

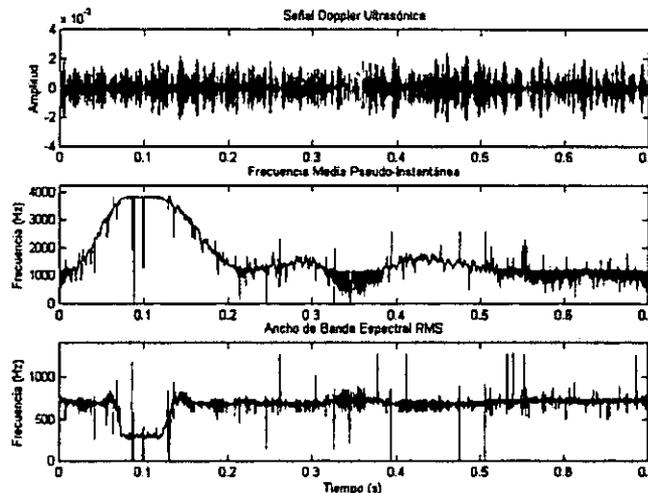


Figura FIG.5.16: Longitud de ventana de muestreo de 15 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.2 Longitud de ventana de muestreo de 31 elementos

En la figura FIG.5.17 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de 31 elementos, sin presencia de ruido.

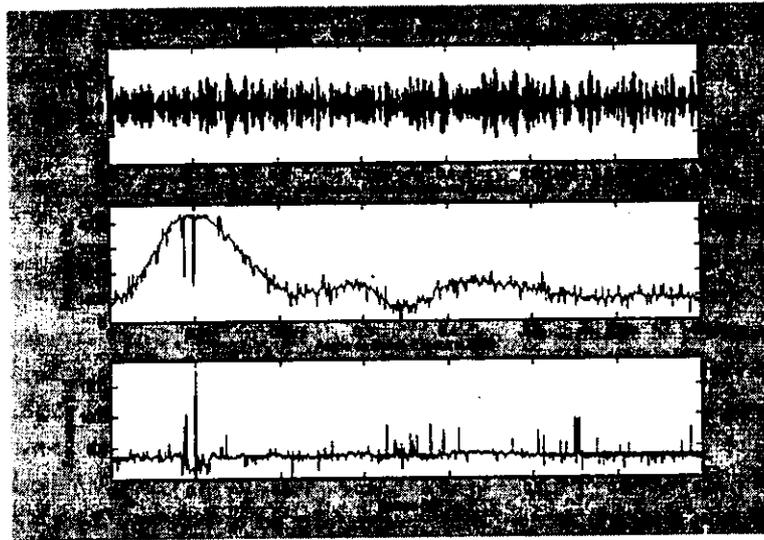


Figura FIG.5.17: Longitud de ventana de muestreo de 31 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.3 Longitud de ventana de muestreo de 63 elementos

En la figura FIG.5.18 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 63 elementos, sin presencia de ruido.

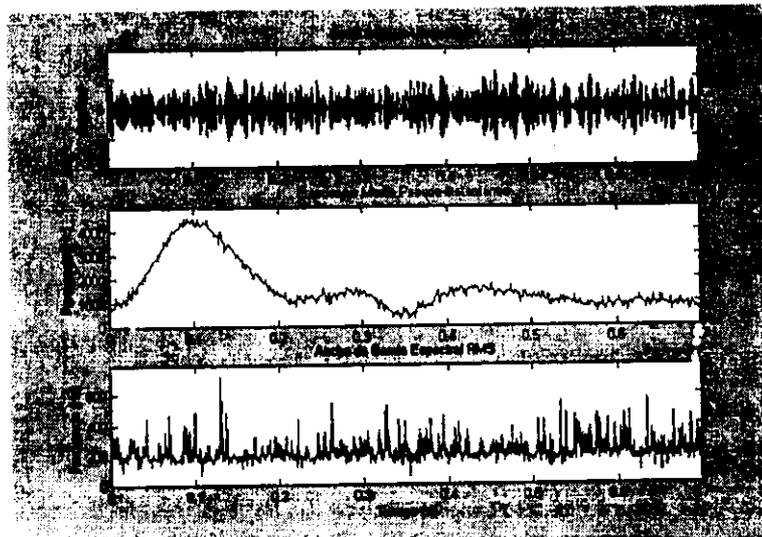


Figura FIG.5.18: Longitud de ventana de muestreo de 63 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.4 Longitud de ventana de muestreo de 127 elementos

En la figura FIG.5.19 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 127 elementos, sin presencia de ruido.

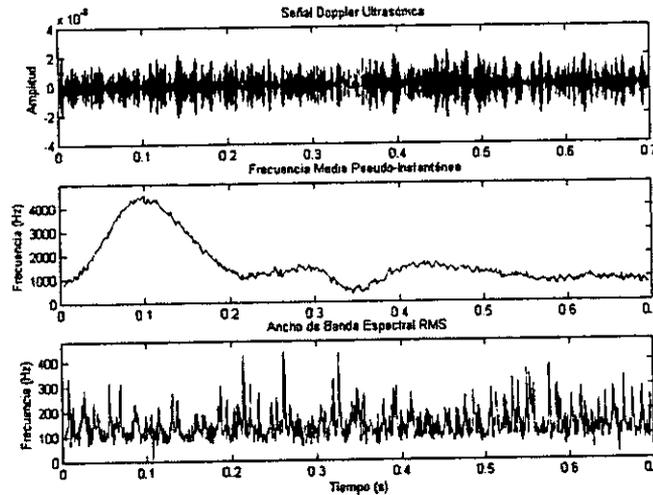


Figura FIG.5.19: Longitud de ventana de muestreo de 127 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.5 Longitud de ventana de muestreo de 255 elementos

En la figura FIG.5.20 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 255 elementos, sin presencia de ruido.

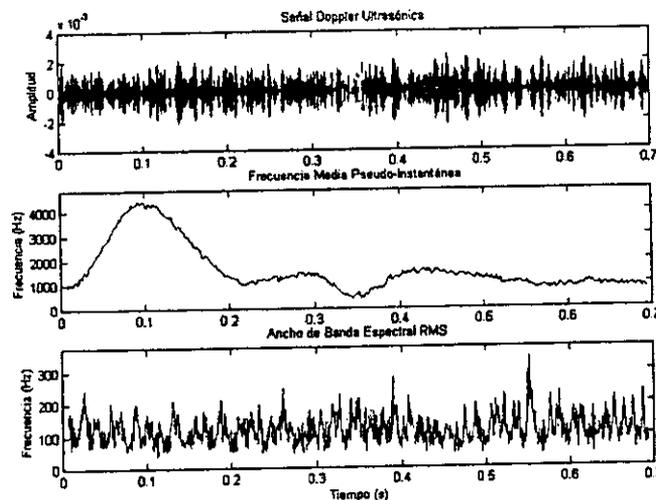


Figura FIG.5.20: Longitud de ventana de muestreo de 255 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.6 Longitud de ventana de muestreo de 511 elementos

En la figura FIG.5.21 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 511 elementos, sin presencia de ruido.

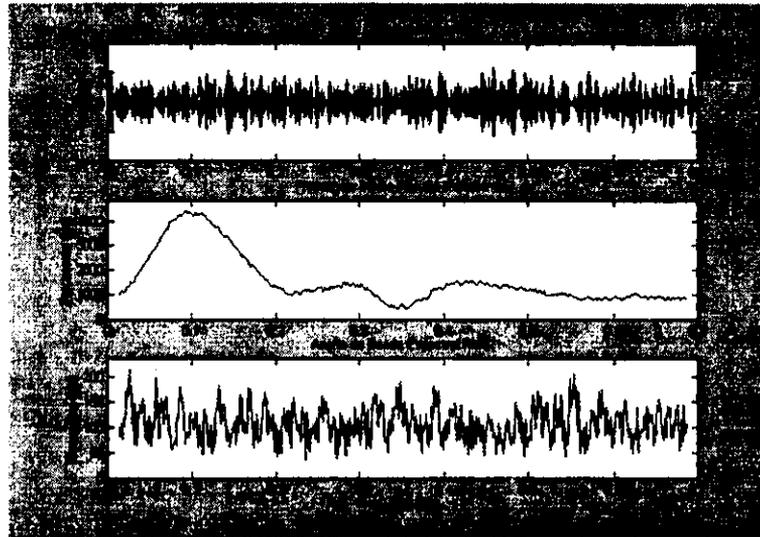


Figura FIG.5.21: Longitud de ventana de muestreo de 511 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

5.8.7 Errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS.

En la figura FIG.5.22 se muestra una gráfica de los errores en la estimación de la frecuencia media pseudo instantánea de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.5.2 se dan los valores respectivos.

Frecuencia Media Pseudo Instantánea Ventana	SNR			
	10	20	40	infinito
15	121.27	80.44	79.21	79.00
31	115.14	44.75	45.79	41.86
63	135.07	42.15	39.66	39.65
127	194.69	44.16	42.20	42.19
255	200.15	55.46	53.56	53.55
511	257.67	80.61	78.14	78.13

Tabla TB.5.2: Error en la estimación de la frecuencia media pseudo instantánea con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

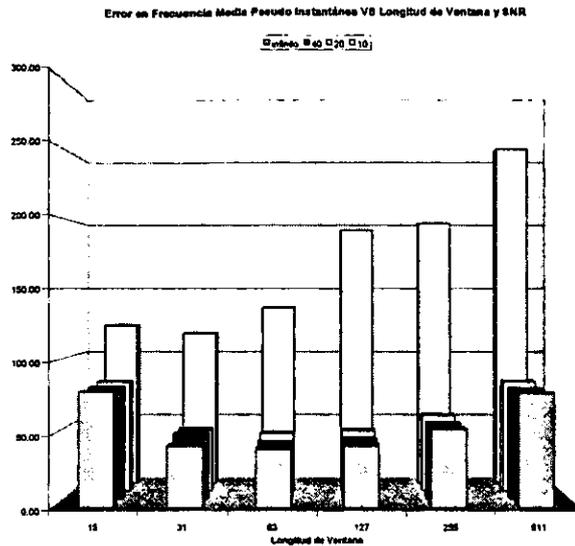


Figura FIG.5.22: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de ventana de muestreo para una SNR de 10, 20, 40 e infinito.

En la figura FIG.5.23 se muestra una gráfica de los errores en la estimación del ancho de banda espectral RMS de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.5.3 se dan los valores respectivos.

Ancho de Banda Espectral RMS Ventana	SNR			
	10	20	40	infinito
15	611.83	567.28	565.71	565.72
31	397.47	266.32	266.37	255.73
63	364.603	138.931	116.808	116.800
127	662.138	84.300	44.274	44.240
255	659.82	77.26	23.94	23.90
511	1018.35	93.62	23.54	23.51

Tabla TB.5.3: Error en la estimación del ancho de banda espectral RMS con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

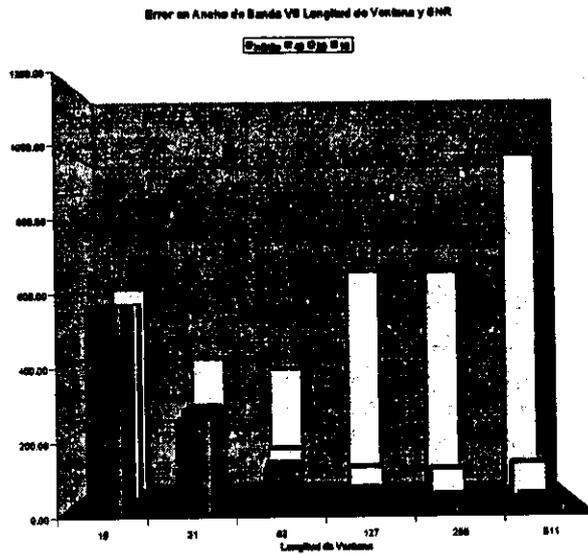


Figura FIG.5.23: Error en la estimación del ancho de banda espectral RMS contra la longitud de la ventana de muestreo para una SNR de 10, 20, 40 e infinito.

5.8.8 Variación del parámetro σ .

En la figura FIG.5.24 se muestra una gráfica de los errores en la estimación del ancho de banda espectral RMS y de la frecuencia media pseudo instantánea de una señal Dópler ultrasónica para una longitud de ventana de muestreo de 127 elementos en función del parámetro σ de la distribución.

Obsérvese que la función de error en la estimación tanto de la frecuencia media pseudo instantánea como del ancho de banda espectral RMS tiene un mínimo. Para el caso de la distribución de Choi Williams, el mínimo se da en $\sigma = 5$.

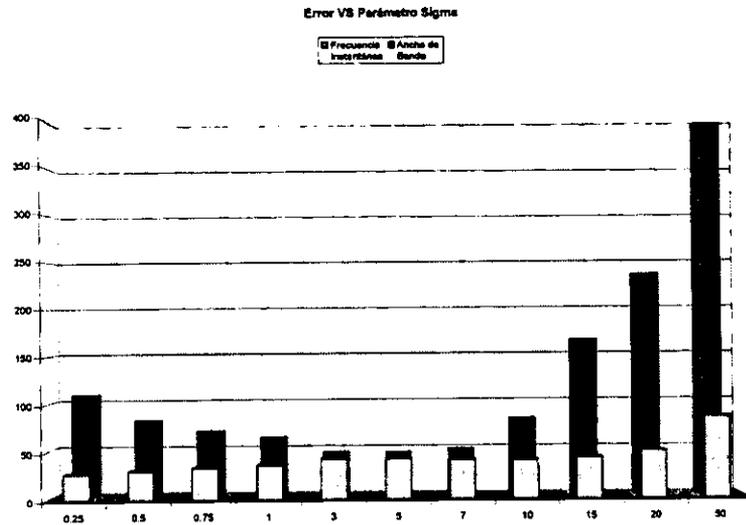


Figura FIG.5.24: Error en la estimación del ancho de banda espectral RMS y de la frecuencia media pseudo instantánea en función del parámetro σ .

5.9 Aplicación a una señal Dópler ultrasónica real en Matlab

En la figura FIG.5.25 se muestra la gráfica de una señal Dópler ultrasónica real, su frecuencia pseudo instantánea y su ancho de banda espectral RMS. La frecuencia de muestreo fue de 10kHz. La ventana de muestreo tiene una longitud de 127 elementos. La señal fue medida en el laboratorio de DISCA^[59].

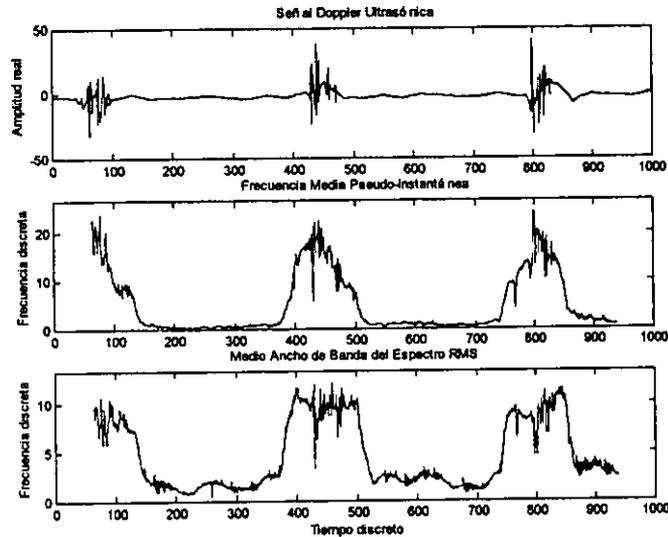


Figura FIG.5.25: Señal Dópler ultrasónica real, su frecuencia media pseudo instantánea y su ancho de banda espectral RMS.

6 DISTRIBUCIÓN DE BÉSEL

6.1 Introducción

En este capítulo se plantea la distribución de Bessel a partir de la clase de distribuciones tiempo frecuencia de Cohen. Posteriormente se desarrolla esta distribución para el caso discreto. Finalmente se transforma la distribución discreta obtenida en una expresión más adecuada para implantar un algoritmo eficiente que la calcule. Además, se desarrollan experimentos numéricos en Matlab que muestran el comportamiento de la distribución discreta en función de la longitud de ventana de muestreo y en función de la relación señal a ruido de la señal muestreada, así como una aplicación a señales Dópler ultrasónicas. La bibliografía básica de este capítulo consiste en [18]; estudios con señales tipo Dópler ultrasónicas se encuentran en [10]; estudios comparativos entre diferentes distribuciones se encuentran en [19].

Las expresiones principales de este capítulo son:

- Distribución de Bessel.

$$BD(t, f) = \int_{-\infty}^{\infty} \frac{2}{\pi \alpha \tau} \int_{-\infty}^{\infty} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu e^{-j2\pi f \tau} d\tau$$

- Distribución de Bessel con una función ventana.

$$BD_w(t, f) = \int_{-\tau_c}^{\tau_c} \frac{2}{\pi \alpha \tau} \int_{-\infty}^{\infty} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu W\left(\frac{\tau}{2}\right) W^*\left(-\frac{\tau}{2}\right) e^{-j2\pi f \tau} d\tau$$

- Distribución discreta de Bessel.

$$DBD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k \tau}{N}} \sum_{\mu=-2\alpha\tau}^{2\alpha\tau} \frac{1}{\pi \alpha \tau} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+n+\tau) x^*(\mu+n-\tau)$$

- Transformación de la distribución discreta de Bessel.

$$DBD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k \tau}{N}} \sum_{\mu=\max\{-2\alpha\tau, -N+1+\tau\}}^{\min\{2\alpha\tau, N-1-\tau\}} \frac{1}{\pi \alpha \tau} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau) x^*(\mu-\tau) \right] - 2|x(0)|^2$$

6.2 Caso continuo

Se planteará la distribución tiempo frecuencia de Bessel ^[18] a partir de la clase de distribuciones de tiempo frecuencia de Cohen ^{[1][2]}.

Primero, el kernel de la distribución de Bessel es ^{[10][19]}:

$$\phi(\theta, \tau) = \frac{2J_1(\alpha\tau\theta)}{\alpha\tau\theta} \tag{EQ.6.1}$$

donde $J_1(t)$ es una función de Bessel de primera clase, de primer orden y $\alpha > 0$ es un factor de escala. En la figura FIG.6.1 se muestra la gráfica de (EQ.6.1) en el dominio θ, τ para $\alpha=1$.

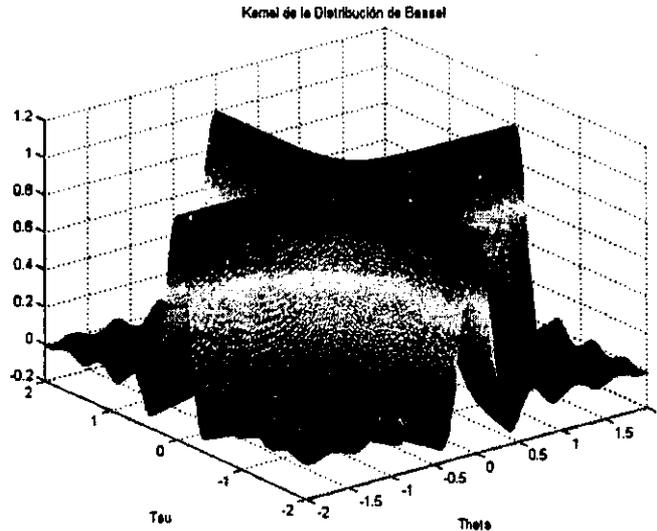


Figura FIG.6.1: Kernel de la distribución tiempo frecuencia de Bessel para $\alpha=1$.

Segundo, el kernel del dominio de la autocorrelación (EQ.3.1) es:

$$\psi(t, \tau) = \int_{-\infty}^{\infty} \frac{2J_1(\alpha\tau\theta)}{\alpha\tau\theta} e^{-j\theta t} d\theta \quad (\text{EQ.6.2})$$

Entonces, si se integra (EQ.6.2) se obtiene (A.6.1):

$$\psi(t, \tau) = \frac{4}{\alpha|\tau|} \sqrt{1 - \left(\frac{t}{\alpha\tau}\right)^2} U_0\left(\frac{t}{\alpha\tau}\right) \quad (\text{EQ.6.3})$$

donde U_0 es el polinomio de Chebyshev de segunda clase:

$$U_0\left(\frac{t}{\alpha\tau}\right) = \begin{cases} 1 & \left(\frac{t}{\alpha\tau}\right)^2 \leq 1 \\ 0 & \text{otro} \end{cases} \quad (\text{EQ.6.3c})$$

En la figura FIG.6.2 se muestra la gráfica de (EQ.6.3) en el dominio t, τ para $\alpha=1$. Tercero, la función de autocorrelación instantánea (EQ.3.2) es:

$$R_x(t, \tau) = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \quad (\text{EQ.6.4})$$

Cuarto, la función de autocorrelación indexada al tiempo generalizada (EQ.3.3) es:

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{4}{\alpha|\tau|} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.6.5})$$

donde:

$$U_0\left(\frac{t-\mu}{\alpha\tau}\right) = \begin{cases} 1 & \left(\frac{t-\mu}{\alpha\tau}\right)^2 \leq 1 \\ 0 & \text{otro} \end{cases} \quad (\text{EQ.6.5c})$$

Entonces, si se reagrupa (EQ.6.5) se obtiene:

$$R'_x(t, \tau) = \frac{2}{\pi\alpha|\tau|} \int_{-\infty}^{\infty} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) d\mu \quad (\text{EQ.6.6})$$

Finalmente, la distribución tiempo frecuencia (EQ.3.5) de Bessel es:

$$BD(t, \omega) = \int_{-\infty}^{\infty} \frac{2}{\pi \alpha \tau} \int_{-\infty}^{\infty} 1 - \left(\frac{t - \mu}{\alpha \tau} \right)^2 U_0 \left(\frac{t - \mu}{\alpha \tau} \right) x \left(\mu + \frac{\tau}{2} \right) x \left(\mu - \frac{\tau}{2} \right) d\mu e^{-j\omega \tau} d\tau \quad (\text{EQ.6.7})$$

o bien, en función de la frecuencia:

$$BD(t, f) = \int_{-\infty}^{\infty} \frac{2}{\pi \alpha \tau} \int_{-\infty}^{\infty} 1 - \left(\frac{t - \mu}{\alpha \tau} \right)^2 U_0 \left(\frac{t - \mu}{\alpha \tau} \right) x \left(\mu + \frac{\tau}{2} \right) x \left(\mu - \frac{\tau}{2} \right) d\mu e^{-j2\pi f \tau} d\tau \quad (\text{EQ.6.8})$$

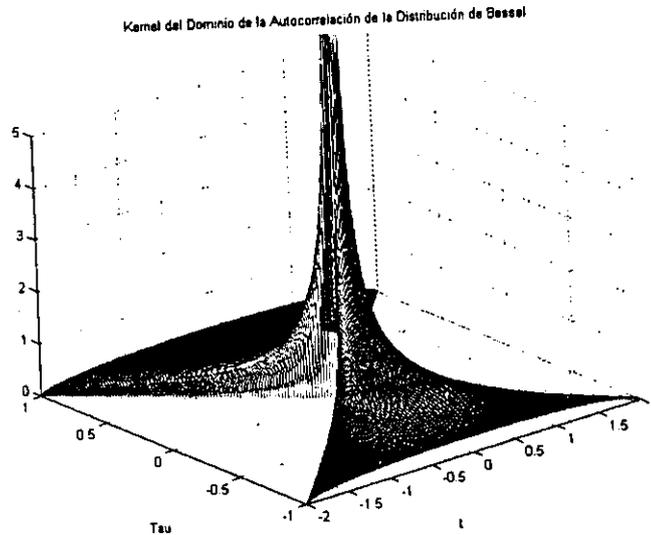


Figura FIG.6.2: Kernel del dominio de la autocorrelación (EQ.6.3) de la distribución tiempo frecuencia de Bessel para $\alpha=1$ (Ventana de Bessel).

6.3 Caso continuo con ventana

Se desarrollará la distribución tiempo frecuencia de Bessel cuando se introduce una función ventana $W(t)$ que multiplica a la función autocorrelación indexada al tiempo generalizada $R_x(t, \tau)$, es decir, a (EQ.6.6).

Si se introduce una función ventana $W(t)$ que cumpla con la propiedad de $W(t) = 0 \quad |t| > T_w/2$ en (EQ.6.8), entonces se obtiene^[9]:

$$BD_w(t, f) = \int_{-T_w}^{T_w} \frac{2}{\pi \alpha \tau} \int_{-\infty}^{\infty} 1 - \left(\frac{t - \mu}{\alpha \tau} \right)^2 U_0 \left(\frac{t - \mu}{\alpha \tau} \right) x \left(\mu + \frac{\tau}{2} \right) x \left(\mu - \frac{\tau}{2} \right) d\mu W \left(\frac{\tau}{2} \right) W^* \left(-\frac{\tau}{2} \right) e^{-j2\pi f \tau} d\tau \quad (\text{EQ.6.9})$$

6.4 Caso discreto con ventana

Se desarrollará la distribución tiempo frecuencia de Bessel discreta a partir de la distribución continua^{[9][10][18]}. Esto es, se discretizará (EQ.6.9).

Primero, si en (EQ.6.9) se realizan los cambios de variables $\tau' = \tau/2$ y $\mu' = \mu - t$, entonces se obtiene:

$$BD_w(t, f) = 2 \int_{-T_w/2}^{T_w/2} \int_{-\infty}^{\infty} \frac{2}{\pi \alpha |2\tau|} \sqrt{1 - \left(\frac{-\mu'}{\alpha(2\tau')}\right)^2} U_0\left(\frac{-\mu'}{\alpha(2\tau')}\right) x(\mu' + t + \tau') x^*(\mu' + t - \tau') d\mu' \quad (\text{EQ.6.10})$$

$$\bullet W(\tau') W^*(-\tau') e^{-j2\pi f(2\tau') d\tau'}$$

Segundo, si en (EQ.6.10) se discretiza la variable frecuencia $f = k\Delta f$, donde $\Delta f = \frac{1}{L} = \frac{1}{2N-1} \approx \frac{1}{2N}$ [8] y k es un índice discreto, siendo L el número de muestras por ventana, y se discretiza la variable tiempo $t = n\Delta T$, $\mu' = \mu\Delta T$ y $\tau' = \tau\Delta T$, donde n, μ y τ son índices discretos, entonces se obtiene:

$$DBD(n\Delta T, k\Delta f) = 2 \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \frac{1}{\pi \alpha |\tau\Delta T|} \sqrt{1 - \left(\frac{-\mu\Delta T}{2\alpha\tau\Delta T}\right)^2} U_0\left(\frac{-\mu\Delta T}{2\alpha\tau\Delta T}\right) x(\mu\Delta T + n\Delta T + \tau\Delta T) \Delta T \quad (\text{EQ.6.11})$$

$$\bullet x^*(\mu\Delta T + n\Delta T - \tau\Delta T) W(\tau\Delta T) W^*(-\tau\Delta T) e^{-\frac{j4\pi k n \tau \Delta T}{2N} \Delta T}$$

Tercero, si en (EQ.6.11) se normaliza el incremento en tiempo $\Delta T = 1$ y el incremento en frecuencia $\Delta f = 1$, entonces se obtiene:

$$DBD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} \sum_{\mu=-M}^M \frac{1}{\pi \alpha |\tau|} \sqrt{1 - \left(\frac{-\mu}{2\alpha\tau}\right)^2} U_0\left(\frac{-\mu}{2\alpha\tau}\right) x(\mu + n + \tau) x^*(\mu + n - \tau) \quad (\text{EQ.6.12})$$

$$\bullet W(\tau) W^*(-\tau) e^{-\frac{j2\pi k n \tau}{N}}$$

Finalmente, la distribución tiempo frecuencia de Bessel discreta es:

$$DBD(n, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2\pi k n \tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi \alpha |\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu + n + \tau) x^*(\mu + n - \tau) \quad (\text{EQ.6.13})$$

en donde si $\tau = 0$ entonces $\sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi \alpha |\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu + n + \tau) x^*(\mu + n - \tau) = x(n) x^*(n)$.

6.5 Análisis de la distribución

Se analizará la efectividad del kernel (EQ.6.1) de la distribución de Bessel (EQ.6.) en la disminución de los efectos de interferencia entre los diferentes componentes de una señal [10][11].

Sea la señal multicomponente $x(t)$ la cual se puede representar como la suma de sus componentes unitarios:

$$x(t) = \sum_{n=1}^N A_n e^{j(\omega_n t + \theta_n)} \quad (\text{EQ.6.14})$$

Si se sustituye (EQ.6.14) en (EQ.6.7) se obtiene:

$$BD(t, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{2}{\pi \alpha |\tau|} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) \sum_{n=1}^N A_n e^{j\left(\omega_n \left(\mu + \frac{\tau}{2}\right) + \theta_n\right)} \sum_{m=1}^N A_m e^{-j\left(\omega_m \left(\mu - \frac{\tau}{2}\right) + \theta_m\right)} d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.6.15})$$

Reagrupando términos en (EQ.6.15) se obtiene:

$$BD(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{m=1}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} \int_{-\infty}^{\infty} \frac{4}{\alpha |\tau|} \sqrt{1 - \left(\frac{t-\mu}{\alpha\tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha\tau}\right) e^{j(\omega_n - \omega_m)\mu} d\mu e^{-j\omega\tau} d\tau \quad (\text{EQ.6.16})$$

Si se aplica la propiedad asociativa de la suma en (EQ.6.16), entonces se obtiene:

$$BD(t, \omega) = BD_{\text{auto}}(t, \omega) + BD_{\text{cruce}}(t, \omega) \quad (\text{EQ.6.17})$$

donde, para el caso en el cual los índices de las sumatorias en (EQ.6.16) son iguales, es decir, $n = m$, se tiene que:

$$BD_{auto}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} \int_{-\infty}^{+\infty} \frac{4}{\alpha \tau} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) d\mu e^{-j\omega \tau} d\tau \quad (EQ.6.18)$$

y para el caso en el cual los índices de las sumatorias en (EQ.6.16) son diferentes $n \neq m$, se tiene que:

$$BD_{cruce}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} \int_{-\infty}^{+\infty} \frac{4}{\alpha \tau} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) e^{j(\omega_n - \omega_m)\mu} d\mu e^{-j\omega \tau} d\tau \quad (EQ.6.19)$$

Primero se analizará (EQ.6.18), la cual se constituye de los auto-términos de la distribución tiempo frecuencia, esto es, de la distribución de cada componente de la señal. Luego se analizará (EQ.6.19), la cual se constituye de los términos de cruce de la distribución tiempo frecuencia, esto es, de la distribución de la interacción de cada par de componentes diferentes.

Referente a (EQ.6.18) o auto-términos. La integral con respecto a μ es^(A.6.2):

$$\int_{-\infty}^{+\infty} \frac{4}{\alpha \tau} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) d\mu = 2\pi \quad (EQ.6.20)$$

Sustituyendo (EQ.6.20) en (EQ.6.18) se obtiene:

$$BD_{auto}(t, \omega) = \sum_{n=1}^N A_n^2 \int_{-\infty}^{+\infty} e^{j\omega_n \tau} e^{-j\omega \tau} d\tau \quad (EQ.6.21)$$

Si se integra (EQ.6.21) se obtiene^(A.6.3):

$$BD_{auto}(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) \quad (EQ.6.22)$$

Referente a (EQ.6.19) o términos de cruce. Para integrar con respecto a μ se aplica la ecuación de Parseval^(A.6.4). Esto es:

$$\int_{-\infty}^{+\infty} \frac{4}{\alpha \tau} \sqrt{1 - \left(\frac{t-\mu}{\alpha \tau}\right)^2} U_0\left(\frac{t-\mu}{\alpha \tau}\right) \left[e^{j(\omega_n - \omega_m)\mu} \right] d\mu = \int_{-\infty}^{+\infty} 2\pi e^{-j\mu} \frac{2J_1(\alpha \tau \mu)}{\alpha \tau \mu} \left[\delta(\mu + (\omega_n - \omega_m)) \right] d\mu \quad (EQ.6.23)$$

Integrando (EQ.6.23) se obtiene^(A.6.5):

$$\int_{-\infty}^{+\infty} 2\pi e^{-j\mu} \frac{2J_1(\alpha \tau \mu)}{\alpha \tau \mu} \left[\delta(\mu + (\omega_n - \omega_m)) \right] d\mu = 2\pi \frac{2J_1(\alpha \tau (- (\omega_n - \omega_m)))}{\alpha \tau (- (\omega_n - \omega_m))} e^{j(\omega_n - \omega_m)\tau} \quad (EQ.6.24)$$

Si se sustituye (EQ.6.24) en (EQ.6.19) se obtiene:

$$BD_{cruce}(t, \omega) = \frac{1}{2\pi} \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} 2\pi \frac{2J_1(\alpha \tau (- (\omega_n - \omega_m)))}{\alpha \tau (- (\omega_n - \omega_m))} e^{j(\omega_n - \omega_m)\tau} e^{-j\omega \tau} d\tau \quad (EQ.6.25)$$

Reagrupando términos en (EQ.6.25) se obtiene:

$$BD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\omega_n - \omega_m)\tau + j(\theta_n - \theta_m)} \int_{-\infty}^{+\infty} \frac{2J_1(\alpha \tau (- (\omega_n - \omega_m)))}{\alpha \tau (- (\omega_n - \omega_m))} e^{j\left(\frac{\omega_n + \omega_m}{2}\right)\tau} e^{-j\omega \tau} d\tau \quad (EQ.6.26)$$

Si se integra (EQ.6.26) se obtiene^(A.6.6):

$$BD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{\substack{m=1 \\ m \neq n}}^N A_n A_m e^{j(\omega_n - \omega_m)\tau + j(\theta_n - \theta_m)} \frac{4}{\alpha - (\omega_n - \omega_m)} \sqrt{1 - \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{-\alpha(\omega_n - \omega_m)}\right)^2} U_0\left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{-\alpha(\omega_n - \omega_m)}\right) \quad (EQ.6.27)$$

Puesto que la distribución es real, entonces:

$$BD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{m=1}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \frac{4}{\alpha|\omega_n - \omega_m|} \sqrt{1 - \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right)^2} U_0 \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right) \quad (\text{EQ.6.28})$$

Finalmente:

$$BD(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) + \sum_{n=1}^N \sum_{m=1}^N A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \frac{4}{\alpha|\omega_n - \omega_m|} \sqrt{1 - \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right)^2} U_0 \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right) \quad (\text{EQ.6.29})$$

El factor de peso en (EQ.6.28) es:

$$FP_{BD}(\omega) = \frac{4}{\alpha|\omega_n - \omega_m|} \sqrt{1 - \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right)^2} U_0 \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)}\right) \quad (\text{EQ.6.30})$$

En la figura FIG.6.3a se muestran gráficas de (EQ.6.30) para diversos valores de ω_n y ω_m para una α constante; mientras que en la figura FIG.6.3b, para diversos valores de α para unos ω_n y ω_m constantes.

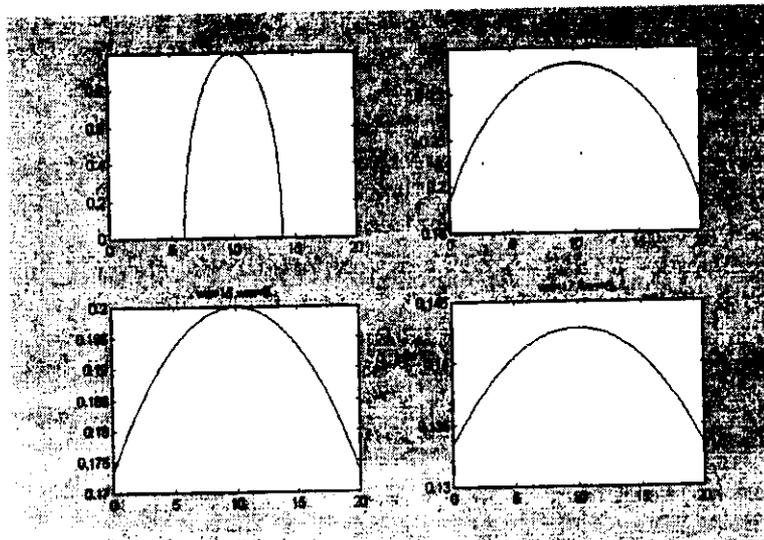


Figura FIG.6.3a: Factor de peso (EQ.6.30) de la distribución tiempo frecuencia de Bessel para diversos valores de ω_n y ω_m con $\alpha=2$.

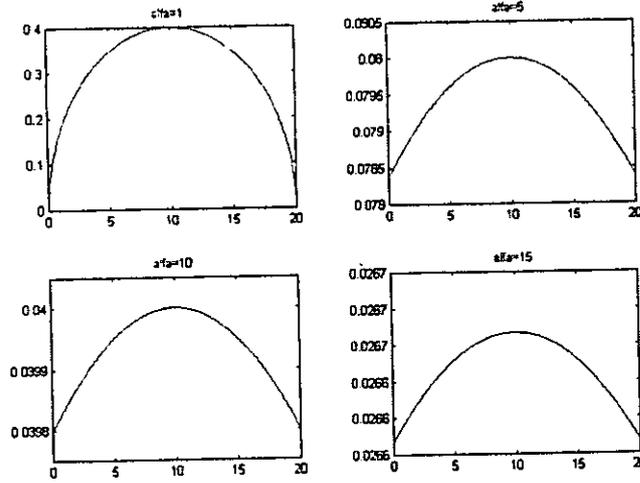


Figura FIG.6.3b: Factor de peso (EQ.6.30) de la distribución tiempo frecuencia de Bessel para diversos valores de α con $\omega_n = 15$ y $\omega_m = 5$.

6.6 Algoritmo eficiente

El algoritmo eficiente calcula (EQ.6.13) valuada en $n = 0$ [2]; por consiguiente:

$$DBD(0, k) = 2 \sum_{\tau=-N+1}^{N-1} W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha\tau}^{2\alpha\tau} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \quad (EQ.6.31)$$

Si se separa (EQ.6.31) en dos sumatorias, una para los índices positivos y otra para los índices negativos de τ , se obtiene:

$$DBD(0, k) = 2 \sum_{\tau=-N+1}^{-1} W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\ + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha\tau}^{2\alpha\tau} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \quad (EQ.6.32)$$

Si a (EQ.6.32) se suma y se resta el término $2W(0)W^*(0)x(0)x^*(0)$ correspondiente al índice $\tau=0$, se obtiene:

$$DBD(0, k) = 2 \sum_{\tau=-N+1}^0 W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\ + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha\tau}^{2\alpha\tau} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\ - 2W(0)W^*(0)x(0)x^*(0) \quad (EQ.6.33)$$

Si en la primera sumatoria de (EQ.6.33) se cambia el índice τ por $-\tau$, se obtiene:

$$DBD(0, k) = 2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha-\tau}^{2\alpha-\tau} \frac{1}{\pi\alpha-\tau} \sqrt{1 - \left(\frac{\mu}{2\alpha(-\tau)}\right)^2} x(\mu-\tau)x^*(\mu+\tau) \\ + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau)e^{-\frac{j2\pi k\tau}{N}} \sum_{\mu=-2\alpha\tau}^{2\alpha\tau} \frac{1}{\pi\alpha\tau} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\ - 2x(0)x^*(0) \quad (EQ.6.34)$$

Si en el primera sumatoria de (EQ.6.34) se aplica el conjugado complejo del conjugado complejo, se obtiene:

$$\begin{aligned}
 DBD(0, k) = & \left(2 \sum_{\tau=0}^{N-1} W(-\tau)W^*(\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu-\tau)x^*(\mu+\tau) \right)^* \\
 & + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\
 & - 2x(0)x^*(0)
 \end{aligned} \tag{EQ.6.35}$$

Si en la primera sumatoria de (EQ.6.35) se calcula uno de los conjugados complejos, se obtiene:

$$\begin{aligned}
 DBD(0, k) = & \left(2 \sum_{\tau=0}^{N-1} W^*(-\tau)W(\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x^*(\mu-\tau)x(\mu+\tau) \right)^* \\
 & + 2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \\
 & - 2x(0)x^*(0)
 \end{aligned} \tag{EQ.6.36}$$

Entonces, reagrupando términos se tiene ^(A.6.7):

$$\begin{aligned}
 DBD(0, k) = & 2 \operatorname{Re} \left[2 \sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \right] \\
 & - 2x(0)x^*(0)
 \end{aligned} \tag{EQ.6.37}$$

Finalmente se tiene que:

$$DBD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=-2\alpha|\tau|}^{2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \right] - 2|x(0)|^2 \tag{EQ.6.38}$$

6.6.1 Optimización del índice μ

Con base en el resultado de la optimización del índice μ en la distribución de Choi Williams, se concluye que el valor máximo y mínimo del índice μ para un valor de τ que no produce multiplicaciones triviales es $\min\{2\alpha|\tau|, N-1-|\tau|\}$ y $\max\{-2\alpha|\tau|, -N+1+|\tau|\}$ respectivamente, donde $|\tau|$ significa el valor absoluto de τ . De este modo, (EQ.6.38) se transforma a:

$$DBD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau)W^*(-\tau) e^{\frac{j2k\tau}{N}} \sum_{\mu=\max\{-2\alpha|\tau|, -N+1+|\tau|\}}^{\min\{2\alpha|\tau|, N-1-|\tau|\}} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu+\tau)x^*(\mu-\tau) \right] - 2|x(0)|^2 \tag{EQ.6.39}$$

6.6.2 Operaciones realizadas

El número de operaciones que se realizan en el cálculo de la distribución discreta de Bessel según las ecuaciones (EQ.6.13) y (EQ.6.39) para cada k : es:

	(EQ. 6.13)	(EQ. 6.39)
Multiplicaciones complejas	$8\alpha N^2 - 8\alpha N$	Menor que $4\alpha N^2 - 4\alpha N + 1$
Sumas complejas	$4\alpha N^2 - 4\alpha N - 2N$	Menor que $2\alpha N^2 - 2\alpha N - N$
Multiplicaciones por escalares	1	2

N es la longitud de la ventana de muestreo. Se considera a $W(\tau)W^*(-\tau)$ como un solo factor;

la misma consideración para $\frac{1}{\pi\alpha\tau} \left[1 - \left(\frac{\mu}{2\alpha\tau} \right)^2 \right]$.

6.7 Experimentos Numéricos en Matlab

En esta sección se presentarán diversos experimentos numéricos^[11] realizados en Matlab^{[32][33]} para comparar gráficamente el comportamiento de la frecuencia instantánea con el comportamiento de la frecuencia media pseudo instantánea y para cuantificar éste último, al variar el nivel de la potencia del ruido en la señal (SNR). En el capítulo 2, Generalidades, se presentan las expresiones para calcular la frecuencia instantánea discreta (EQ.2.19) y la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presentan las expresiones para calcular la relación señal a ruido SNR (EQ.A4.2) y el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5). Para todos los experimentos, la longitud de ventana de muestreo es de 127 elementos.

6.7.1 Experimento 1: Función seno con frecuencia de 0.32Hz y amplitud constante

En la figura FIG.6.4 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función seno con frecuencia constante de 0.32 Hz y amplitud constante. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.32 Hz.

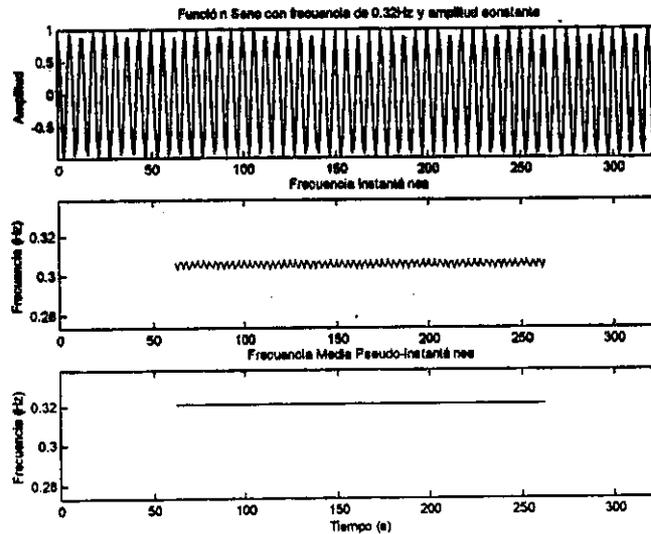


Figura FIG.6.4: a) Función seno con frecuencia de 0.32Hz y amplitud constante; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.6.5 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

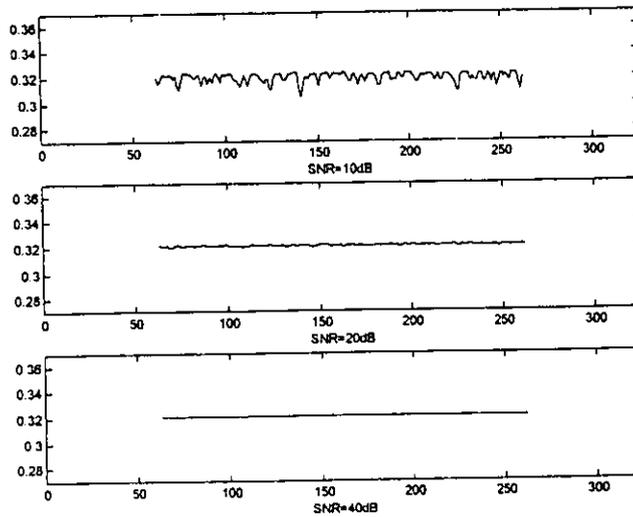


Figura FIG.6.5: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

6.7.2 Experimento 2: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes

En la figura FIG.6.6 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.28 Hz.

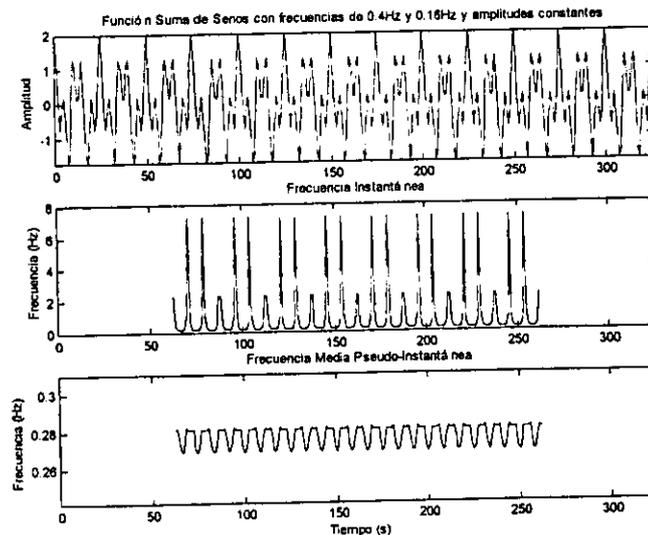


Figura FIG.6.6: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.6.7 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

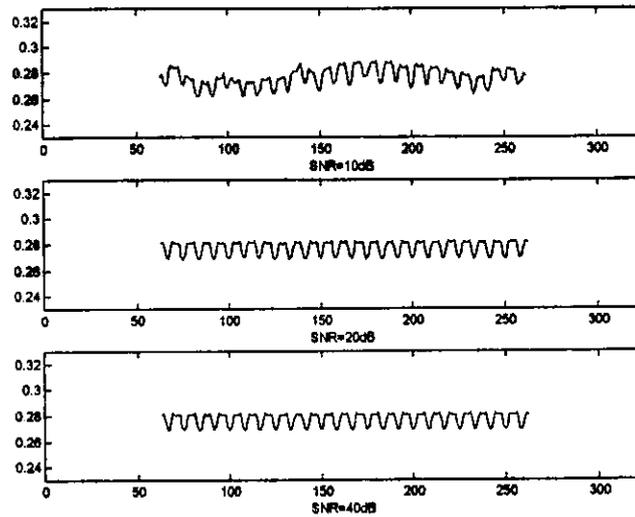


Figura FIG.6.7: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

6.7.3 Experimento 3: Función Suma de Senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes

En la figura FIG.6.8 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, 0.24 Hz.

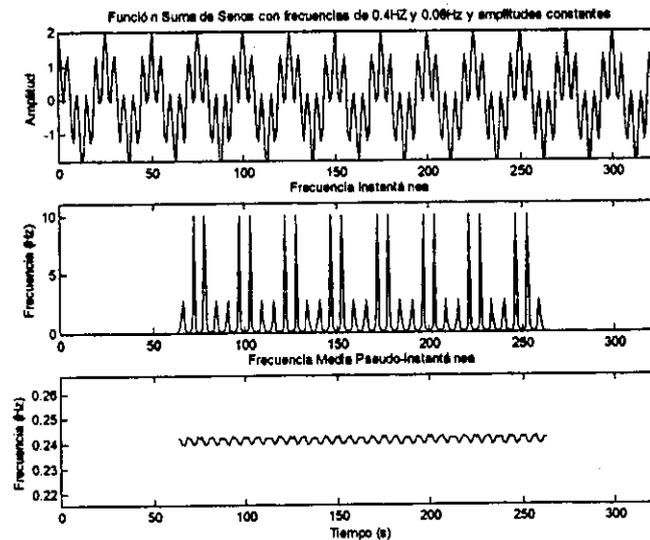


Figura FIG.6.8: a) Función suma de senos con frecuencias de 0.40Hz y 0.08Hz y amplitudes constantes; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.6.9 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

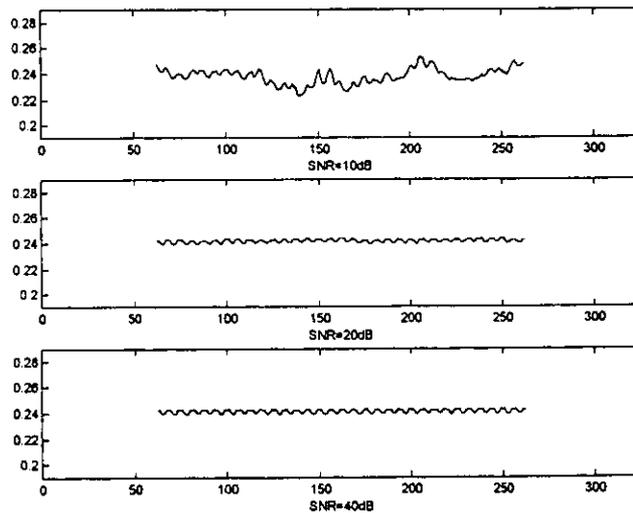


Figura FIG.6.9: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

6.7.4 Experimento 4: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.6.10 se muestran las gráficas de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 128 segundos.

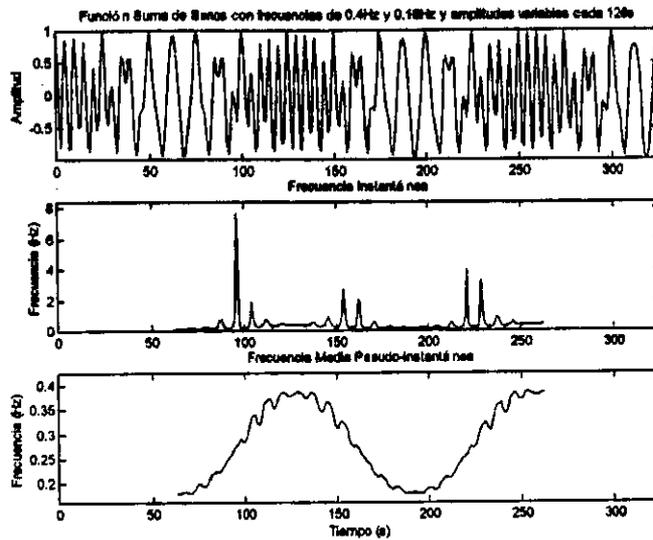


Figura FIG.6.10: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 128 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.6.11 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

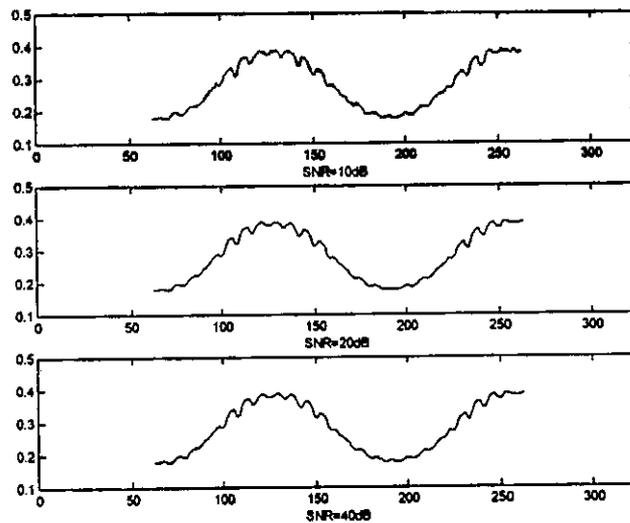


Figura FIG.6.11: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

6.7.5 Experimento 5: Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables con periodo de 128s

En la figura FIG.6.12 se muestra la gráfica de la frecuencia instantánea y de la frecuencia media pseudo instantánea de la función suma de senos con frecuencias de 0.40Hz y

0.16Hz y amplitudes variables cada 64 segundos. Se observa que la frecuencia media pseudo instantánea da una mejor aproximación de la frecuencia real de la función, variación lineal desde 0.16 Hz hasta 0.40 Hz y nuevamente hasta 0.16 Hz cada 64 segundos.

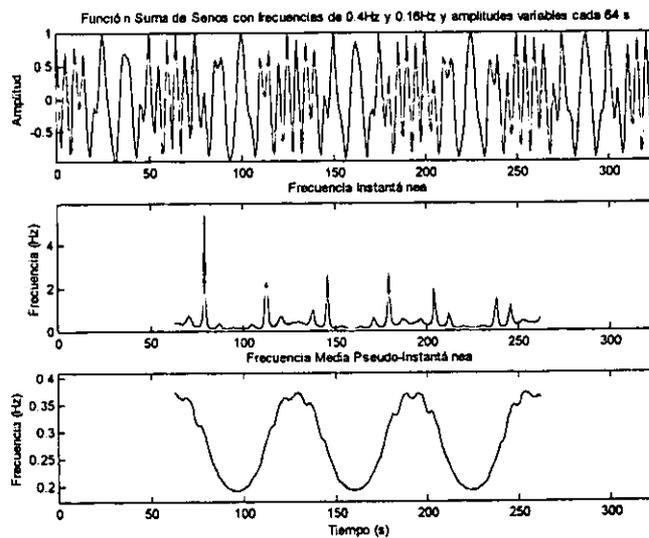


Figura FIG.6.12: a) Función suma de senos con frecuencias de 0.40Hz y 0.16Hz y amplitudes variables cada 64 segundos; b) Frecuencia Instantánea de la función; c) Frecuencia Media Pseudo Instantánea de la función.

En la figura FIG.6.13 se muestran las gráficas de la frecuencia media pseudo instantánea para distintos niveles de SNR.

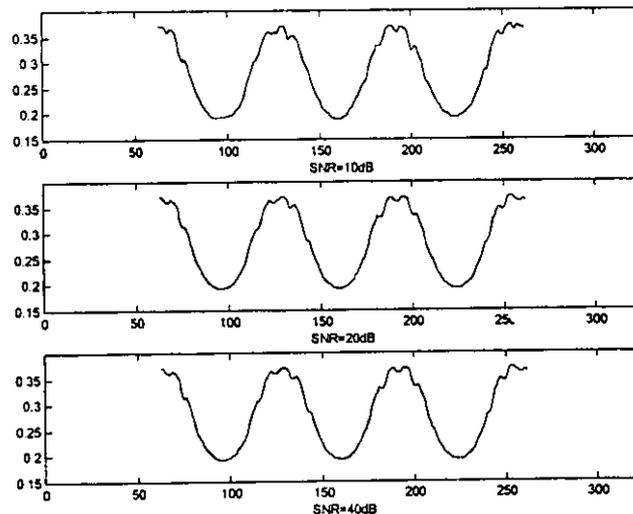


Figura FIG.6.13: Frecuencia Media Pseudo Instantánea para distintos valores de SNR. a) SNR=10dB; b) SNR=20dB; c) SNR=40dB.

6.7.6 Errores en la estimación de la frecuencia media pseudoinstantánea en los experimentos 1 a 5

En la tabla TB.6.1 se muestra una relación del error promedio en la estimación de la frecuencia media pseudo instantánea con respecto a SNR, obtenido en 10 instancias de cada experimento para cada nivel de SNR.

SNR	Error en Frecuencia Media Pseudo Instantánea			
	Infinito	40dB	20dB	10dB
Experimento 1	0.001101	0.001101	0.001085	0.000833
Experimento 2	0.003336	0.003336	0.003330	0.004364
Experimento 3	0.001399	0.001400	0.001400	0.002672

Tabla TB.6.1: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR.

6.7.7 Experimento 6: Función tren de pulsos de senos de diferentes frecuencias

En la figura FIG.6.14.a se muestra la gráfica de una función tren de pulsos de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.6.13). En la figura FIG.6.14.b se muestran diferentes vistas de esta distribución. Nótese que donde no existe señal en el tiempo, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.6.2*i*); además, obsérvese su magnitud.

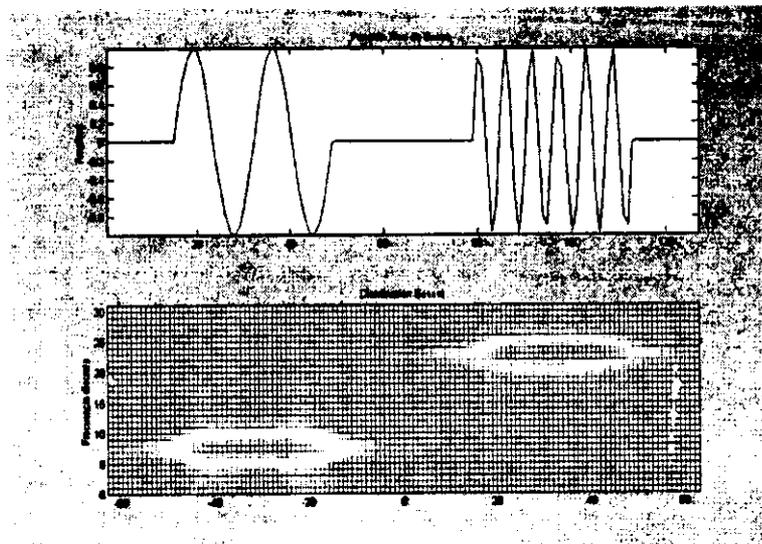


Figura FIG.6.14.a: Función tren de pulsos de senos de diferentes frecuencias y su distribución tiempo frecuencia de Bessel.

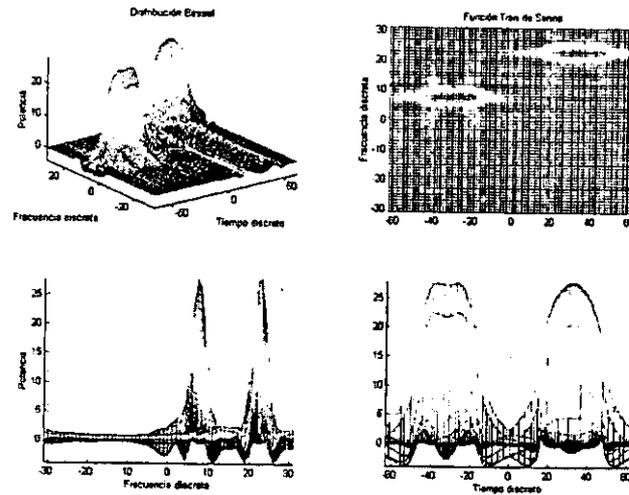


Figura FIG.6.14.b: Diferentes vistas de la distribución tiempo frecuencia de Bessel de una función tren de pulsos de senos de diferentes frecuencias.

Además, en la figura FIG.6.15.a se muestra la gráfica de una función tren de pulsos de senos de diferentes frecuencias, así como su distribución tiempo frecuencia calculada con la ecuación (EQ.6.13). En la figura FIG.6.15.b se muestran diferentes vis'as de esta distribución. Nótese que donde no existe señal en la frecuencia, sí existe una distribución frecuencial. Ésta distribución frecuencial se asocia con la potencia de los términos de cruce (EQ.6.28); además, obsérvese su magnitud.

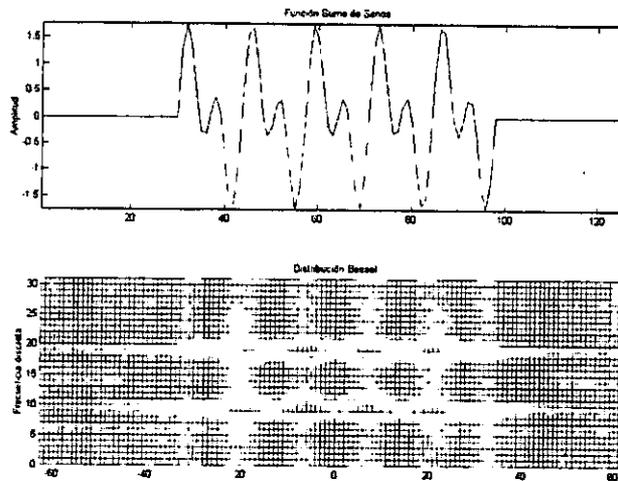


Figura FIG.6.14.a: Función suma de senos de diferentes frecuencias y su distribución tiempo frecuencia de Bessel.

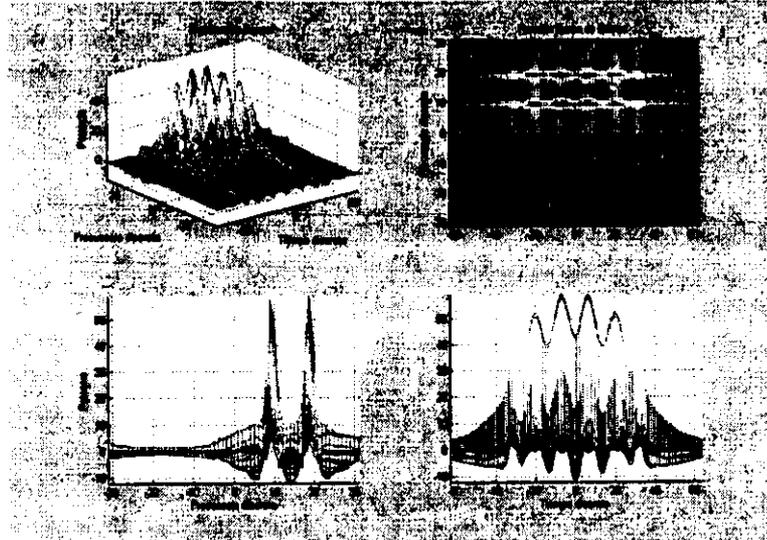


Figura FIG.6.15.b: Diferentes vistas de la distribución tiempo frecuencia de Bessel de una función suma de senos de diferentes frecuencias.

6.8 Aplicación a una señal Dópler ultrasónica simulada en Matlab

En esta sección se presentarán diversos experimentos numéricos^[10] realizados en Matlab^{[32][33]} para mostrar gráfica y cuantitativamente el comportamiento de la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS de la señal, al variar la longitud de la ventana de muestreo (15, 31, 63, 127, 255 y 511 elementos), y al variar la relación de potencias de la señal y el ruido (SNR de 10dB, 20dB, 40dB e infinito). En el capítulo 2, Generalidades, se presenta la expresión para calcular la frecuencia media pseudo instantánea discreta (EQ.2.21). En el anexo 4, Otras expresiones para experimentos numéricos, se presenta la expresión para calcular el ancho de banda espectral RMS (EQ.A4.1). En este mismo anexo se presentan las expresiones para calcular la relación señal a ruido (SNR) (EQ.A4.2), el error en la estimación de la frecuencia media pseudo instantánea (EQ.A4.3), (EQ.A4.4) y (EQ.A4.5) y el error en la estimación del ancho de banda espectral RMS (EQ.A4.6), (EQ.A4.7) y (EQ.A4.8). La señal tipo Dópler ultrasónica simulada que se considera en este análisis posee una frecuencia media descrita en la figura FIG.A3.2 del Anexo 3 y un ancho de banda espectral RMS constante de 100 Hz.

6.8.1 Longitud de ventana de muestreo de 15 elementos

En la figura FIG.6.16 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 15 elementos, sin presencia de ruido.

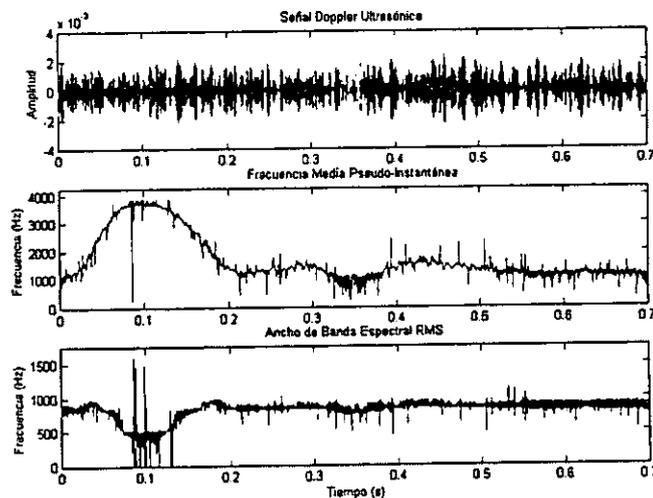


Figura FIG.6.16: Longitud de ventana de muestreo de 15 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.2 Longitud de ventana de muestreo de 31 elementos

En la figura FIG.6.17 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de 31 elementos, sin presencia de ruido.

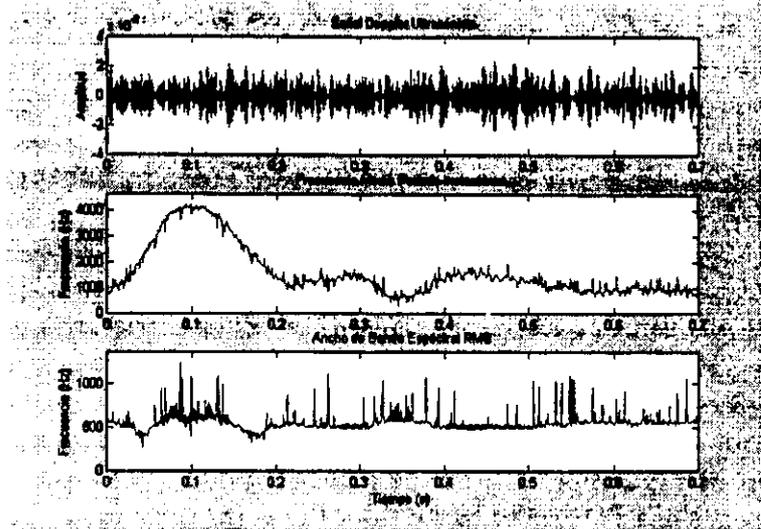


Figura FIG.6.17: Longitud de ventana de muestreo de 31 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.3 Longitud de ventana de muestreo de 63 elementos

En la figura FIG.6.18 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 63 elementos, sin presencia de ruido.

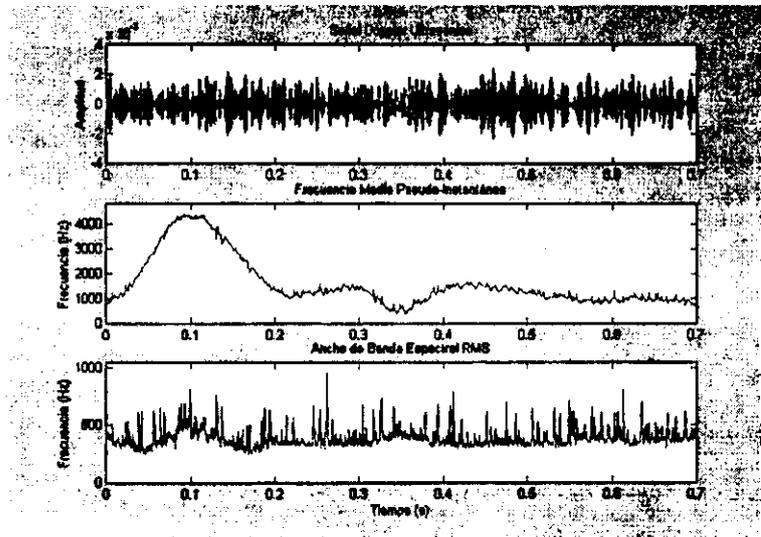


Figura FIG.6.18: Longitud de ventana de muestreo de 63 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.4 Longitud de ventana de muestreo de 127 elementos

En la figura FIG.6.19 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 127 elementos, sin presencia de ruido.

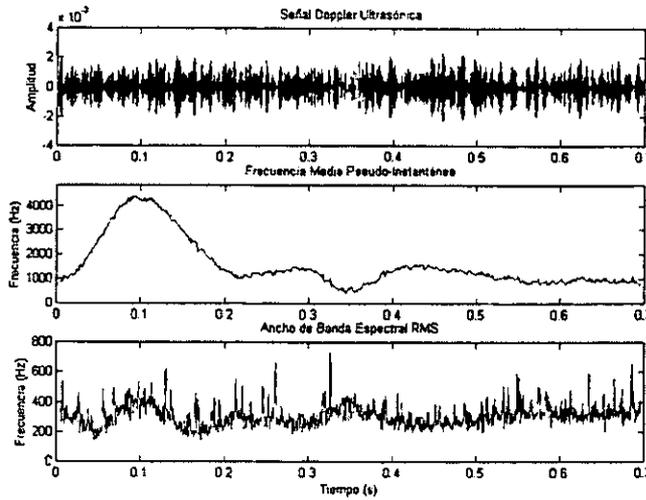


Figura FIG.6.19: Longitud de ventana de muestreo de 127 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.5 Longitud de ventana de muestreo de 255 elementos

En la figura FIG.6.20 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 255 elementos, sin presencia de ruido.

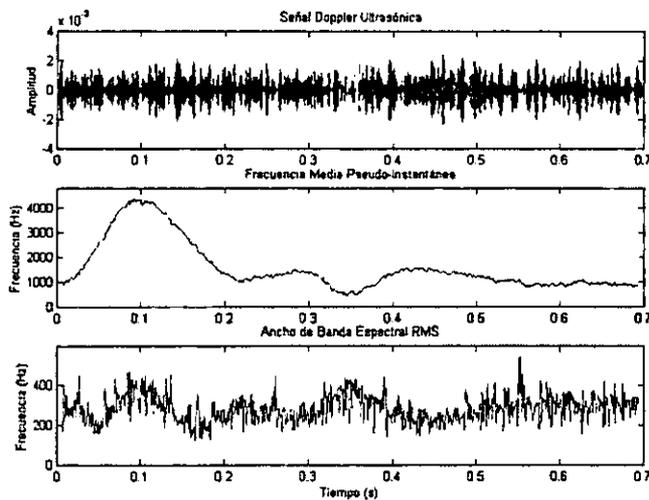


Figura FIG.6.20: Longitud de ventana de muestreo de 255 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.6 Longitud de ventana de muestreo de 511 elementos

En la figura FIG.6.21 se muestran las gráficas de la señal Dópler ultrasónica así como de su frecuencia media pseudo instantánea y de su ancho de banda espectral RMS para una longitud de ventana de muestreo de 511 elementos, sin presencia de ruido.

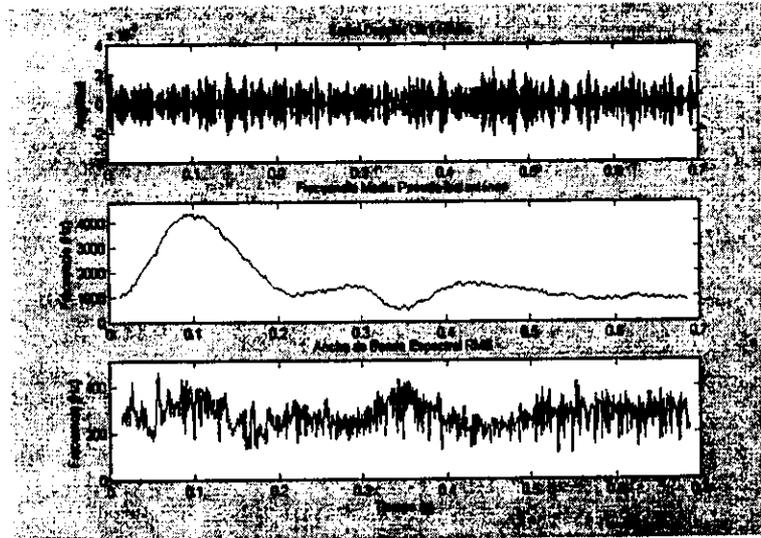


Figura FIG.6.21: Longitud de ventana de muestreo de 511 elementos. a) Señal Dópler Ultrasónica; b) Frecuencia Media Pseudo Instantánea; c) Ancho de Banda Espectral RMS.

6.8.7 Errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS.

En la figura FIG.6.22 se muestra una gráfica de los errores en la estimación de la frecuencia media pseudo instantánea de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.6.2 se dan los valores respectivos.

Frecuencia Media Pseudo Instantánea	SNR				
	Ventana	10	20	40	infinito
15		105.38	86.14	85.69	85.69
31		77.49	49.69	49.45	49.45
63		86.94	47.61	47.10	47.10
127		82.65	36.76	36.14	36.14
255		114.03	47.32	46.37	46.38
511		183.25	80.98	79.56	79.56

Tabla TB.6.2: Error en la estimación de la frecuencia media pseudo instantánea con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

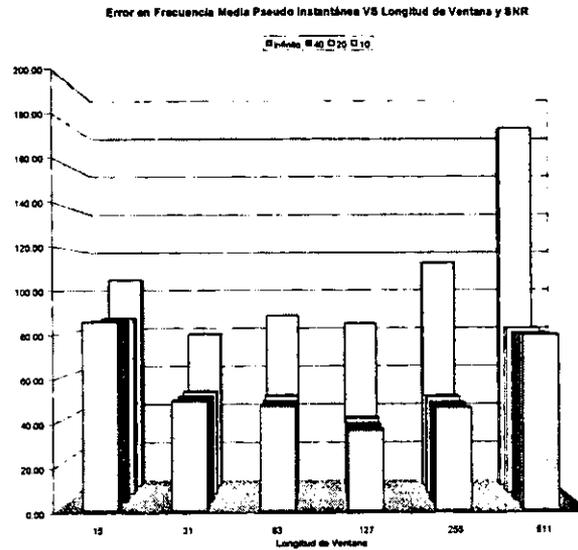


Figura FIG.6.22: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de la ventana de muestreo para una SNR de 10, 20, 40 e infinito.

En la figura FIG.6.23 se muestra una gráfica de los errores en la estimación del ancho de banda espectral RMS de una señal Dópler ultrasónica contra la longitud de ventana de muestreo y la relación de potencias de la señal y el ruido (SNR). En la tabla TB.6.3 se dan los valores respectivos.

Ancho de Banda Espectral RMS Ventana	SNR			
	10	20	40	infinito
15	741.83	710.59	709.83	709.83
31	487.50	456.20	456.10	456.10
63	364.81	285.73	285.01	285.01
127	380.07	216.92	215.47	215.47
255	528.07	193.54	188.94	188.94
511	848.45	208.34	195.22	195.22

Tabla TB.6.3: Error en la estimación del ancho de banda espectral RMS con respecto a la longitud de ventana de muestreo y a la relación de potencias de la señal y el ruido (SNR)

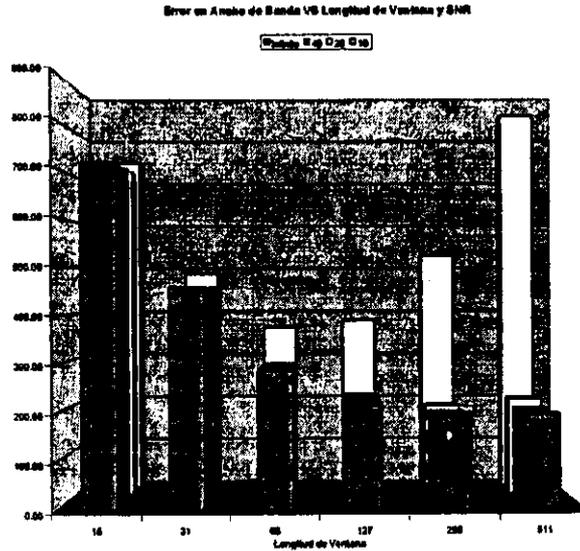


Figura FIG.6.23: Error en la estimación del ancho de banda espectral RMS contra la longitud de la ventana de muestreo para una SNR de 10,20, 40 e infinito.

6.8.8 Variación del parámetro α .

En la figura FIG.6.24 se muestra una gráfica de los errores en la estimación del ancho de banda espectral RMS y de la frecuencia media pseudo instantánea de una señal Dópler ultrasónica para una longitud de ventana de muestreo de 127 elementos en función del parámetro α de la distribución.

Obsérvese que la función de error en la estimación tanto de la frecuencia media pseudo instantánea como del ancho de banda espectral RMS tiene un mínimo. Para el caso de la distribución de Bessel, el mínimo se da en $\alpha = 2$.

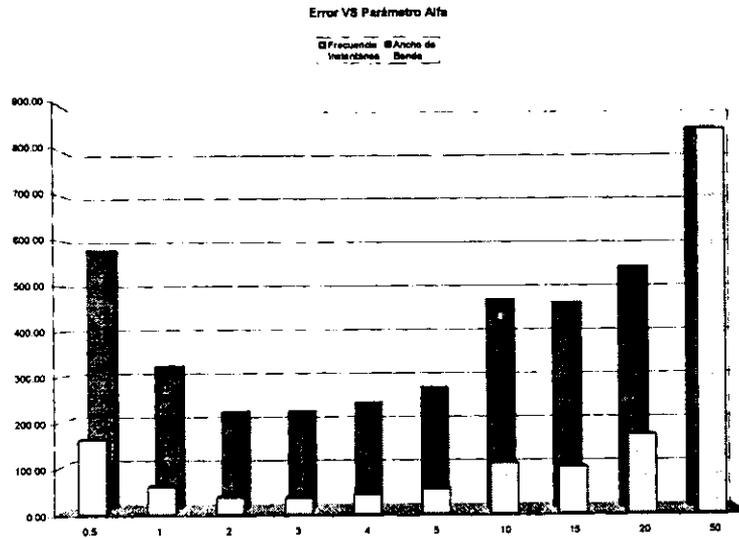


Figura FIG.6.24: Error en la estimación del ancho de banda espectral RMS y de la frecuencia media pseudo instantánea en función del parámetro α .

6.9 Aplicación a una señal Dópler ultrasónica real en Matlab

En la figura FIG.6.25 se muestra la gráfica de una señal Dópler ultrasónica real, su frecuencia pseudo instantánea y su ancho de banda espectral RMS. La frecuencia de muestreo fue de 10kHz. La ventana de muestreo tiene una longitud de 127 elementos. La señal fue medida en el laboratorio de DISCA^[59].

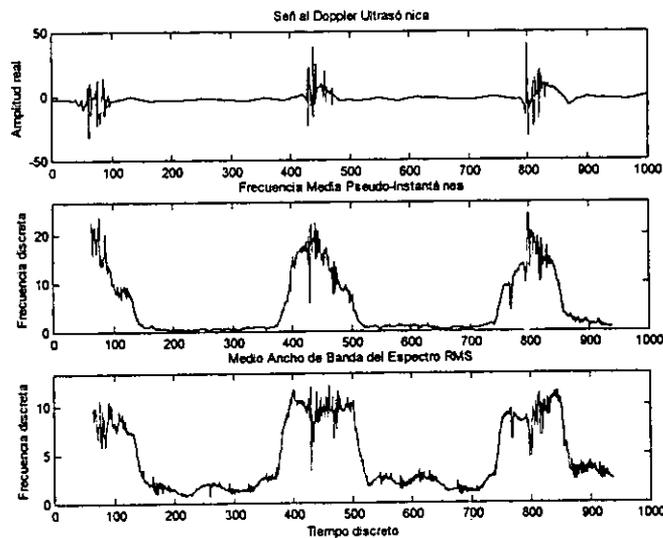


Figura FIG.6.25: Señal Dópler ultrasónica real, su frecuencia media pseudo instantánea y su ancho de banda espectral RMS.

7 IMPLANTACIÓN DE ALGORITMOS PARALELOS PARA ESTIMACIÓN ESPECTRAL

7.1 Introducción

En este capítulo se diseñarán algoritmos paralelos para estimación espectral que calculen las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel. Primero se plantearán los algoritmos secuenciales y luego se planteará su paralelización. La implantación será en lenguaje de programación Occam^{[34][35][36][37]} en una arquitectura de cómputo paralelo de alto desempeño basada en procesadores Transputers T805^{[39][40]}. La bibliografía disponible sobre algoritmos paralelos para calcular la transformada de Fourier en una arquitectura de cómputo paralelo basada en Transputers es [30], [41], [42], [43] y [44].

7.2 Objetivo del algoritmo paralelo TFD

El objetivo de los algoritmos paralelos para estimación espectral es calcular las distribuciones tiempo frecuencia (TFD), ya sea de Wigner Ville, de Choi Williams o de Bessel, de un vector real de cierta longitud, ya sea de 15, 31, 63, 127 ó 255, los cuales se implantarán en lenguaje de programación Occam en una arquitectura de cómputo paralelo de alto desempeño basada en procesadores Transputers T805.

7.3 Algoritmo secuencial TFD

Antes de proceder a la descripción del algoritmo paralelo que calcula las distribuciones tiempo frecuencia, se procederá a describir el algoritmo secuencial correspondiente.

El algoritmo secuencial para calcular la distribución tiempo frecuencia (TFD) de un vector real es el siguiente:

Datos:

Señal real.

Algoritmo secuencial:

1. Calcular la señal analítica de la señal real.
2. Calcular la función de autocorrelación generalizada indexada al tiempo de la señal analítica.
3. Calcular la transformada de Fourier de la función de autocorrelación generalizada indexada al tiempo, lo cual constituye la distribución tiempo frecuencia de la señal real.

Respuesta:

Distribución tiempo frecuencia del vector real.

En la figura FIG.7.1 se muestra el diagrama de flujo del algoritmo secuencial para calcular la distribución tiempo frecuencia.

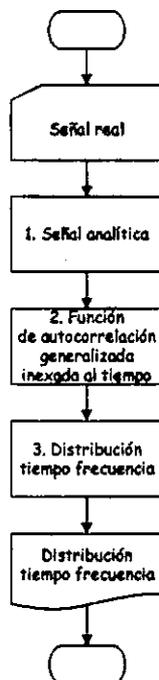


Figura FIG.7.1: Diagrama de flujo del algoritmo secuencial que calcula la distribución tiempo frecuencia.

La descripción de cada paso del algoritmo secuencial TFD se hará a continuación.

7.3.1 Cálculo de la señal analítica del vector real

En el inciso Señal analítica del capítulo 2, Generalidades, se plantea el cálculo de la señal analítica. Ver ecuaciones (EQ.2.11) y (EQ.2.22). Para el cálculo de la transformada directa e inversa de Fourier refiérase a la sección 7.5, algoritmo FFT en base 2 de diezmo en frecuencia, de este capítulo. El algoritmo que calcula la señal analítica del vector real para el caso discreto es:

Datos:

Señal real $x(n)$, de longitud L y con índice $n = 0, \dots, L - 1$.

$x(0)$	$x(1)$...	$x(L-2)$	$x(L-1)$
--------	--------	-----	----------	----------

Algoritmo:

1. Calcular $X(k)$, la transformada de Fourier de la señal real $x(n)$, a través de la siguiente expresión:

$$X(k) = \sum_{n=0}^{L-1} x(n) W_L^{kn} \quad (\text{EQ.7.1})$$

donde $W_L^{kn} = e^{-j2\pi kn/L}$ y con índice $k = 0, \dots, L - 1$.

2. Calcular $X_a(k)$, la transformada de Fourier de la señal analítica $x_a(n)$, a través de la siguiente expresión:

$$X_a(k) = \begin{cases} 2X(k) & k = 1, \dots, \frac{L}{2} - 1 \\ X(k) & k = 0 \\ 0 & k = \frac{L}{2}, \dots, L-1 \end{cases} \quad (\text{EQ.7.2})$$

3. Calcular la señal analítica $x_a(n)$, la transformada inversa de Fourier de $X_a(k)$, a través de la siguiente expresión:

$$x_a(n) = \frac{1}{L} \sum_{k=0}^{L-1} X_a(k) W_L^{-kn} \quad (\text{EQ.7.3})$$

donde $W_L^{-kn} = e^{\frac{j2\pi kn}{L}}$ y con índice $n = 0, \dots, L-1$.

Resultados:

Señal analítica $x_a(n)$, de longitud L , con índice $n = 0, \dots, L-1$.

$x_a(0)$	$x_a(1)$...	$x_a(L-2)$	$x_a(L-1)$
----------	----------	-----	------------	------------

7.3.2 Cálculo de la función de autocorrelación generalizada indexada al tiempo de la señal analítica

En el inciso Algoritmo eficiente de los capítulos 4, 5 y 6, Distribuciones de Wigner Ville, de Choi Williams y de Bessel respectivamente, se plantea el cálculo de la función de autocorrelación generalizada indexada al tiempo. Ver las ecuaciones (EQ.4.32), (EQ.5.39) y (EQ.6.39). El algoritmo que calcula la función de autocorrelación generalizada indexada al tiempo de la señal analítica para el caso discreto es:

Datos:

Señal analítica $x_a(n)$, de longitud $L = 2N - 1$, con índice $n = -N + 1, \dots, 0, \dots, N - 1$.

$x_a(-N+1)$...	$x_a(0)$...	$x_a(N-1)$
-------------	-----	----------	-----	------------

Nótese la reenumeración del índice n respecto al resultado de la sección 7.3.1.

Algoritmo:

1. Calcular $R_{x_w}^*(0, \tau)$, la función de autocorrelación generalizada indexada al tiempo valuada en $n=0$, de longitud N y con índice $\tau = 0, \dots, N-1$, multiplicada por una ventana $W(\tau)$, de longitud $2N-1$ y con índice $\tau = -N+1, \dots, 0, \dots, N-1$, a través de la expresión correspondiente a la distribución tiempo frecuencia en cuestión.

- 1.a. Específicamente, para la distribución tiempo frecuencia de Wigner:

$$R_{x_w}^*(0, \tau) = W(\tau) W^*(-\tau) x_a(\tau) x_a^*(-\tau) \quad (\text{EQ.7.4})$$

1.b. Para la distribución tiempo frecuencia de Choi Williams:

$$R'_{xw}(0, \tau) = W(\tau)W^*(-\tau) \sum_{\mu=-N+1+|\tau|}^{N-1-|\tau|} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x_a(\mu+\tau)x_a^*(\mu-\tau) \quad (\text{EQ.7.5})$$

1.c. Para la distribución tiempo frecuencia de Bessel:

$$R'_{xw}(0, \tau) = W(\tau)W^*(-\tau) \sum_{\mu=\max\{-2\alpha|\tau|, -N+1+|\tau|\}}^{\min\{2\alpha|\tau|, N-1-|\tau|\}} \frac{1}{\pi\alpha|\tau|} \sqrt{1-\left(\frac{\mu}{2\alpha\tau}\right)^2} x_a(\mu+\tau)x_a^*(\mu-\tau) \quad (\text{EQ.7.6})$$

Resultados:

Función $R'_{xw}(0, \tau)$, de longitud N y con índice $\tau=0, \dots, N-1$, la cual constituye la función de autocorrelación generalizada indexada al tiempo $R'_x(n, \tau)$, valuada en $n=0$, multiplicada por una ventana $W(\tau)$.

$R'_{xw}(0,0)$	$R'_{xw}(0,1)$...	$R'_{xw}(0,N-2)$	$R'_{xw}(0,N-1)$
----------------	----------------	-----	------------------	------------------

7.3.3.1 Ventana $W(t)$

La ventana que se utiliza en este trabajo es una ventana de Hanning, cuya definición es:

$$W(\tau) = 0.5 - 0.5 \cos\left(\frac{2\pi(\tau+N)}{L+1}\right) \quad (\text{EQ.7.6.c})$$

donde $L=2N-1$ es la longitud de la ventana y con índice $\tau=-N+1, \dots, 0, \dots, N-1$. Nótese que $W(0)=1$ y que la función es real, es decir, $W(\tau)=W^*(\tau)$.

$W(-N+1)$...	$W(0)$...	$W(N-1)$
-----------	-----	--------	-----	----------

7.3.3 Cálculo de la distribución tiempo frecuencia del vector real

En el inciso Distribuciones tiempo frecuencia del capítulo 3, Distribuciones Tiempo Frecuencia, se plantea el cálculo de la distribución tiempo frecuencia dada la función de autocorrelación generalizada indexada al tiempo. Ver la ecuación (EQ.3.5). Adicionalmente, en el inciso Algoritmo eficiente de los capítulos 4, 5 y 6, Distribuciones de Wigner Ville, de Choi Williams y de Bessel respectivamente, se plantea el cálculo de la distribución tiempo frecuencia. Ver las ecuaciones (EQ.4.32), (EQ.5.39) y (EQ.6.39). El algoritmo que calcula la distribución tiempo frecuencia dada la función de autocorrelación generalizada indexada al tiempo de la señal analítica para el caso discreto es:

Datos:

Función $R'_{xw}(0, \tau)$, de longitud N y con índice $\tau=0, \dots, N-1$, la cual constituye la función de autocorrelación generalizada indexada al tiempo $R'_x(n, \tau)$, valuada en $n=0$, multiplicada por una ventana $W(\tau)$.

$R'_{xw}(0,0)$	$R'_{xw}(0,1)$...	$R'_{xw}(0,N-2)$	$R'_{xw}(0,N-1)$
----------------	----------------	-----	------------------	------------------

Algoritmo:

1. Calcular $TFD(0, k)$, la trasformada de Fourier de la función $R_{xw}^i(0, \tau)$, a través de la siguiente expresión:

$$TFD(0, k) = 4 \operatorname{Real} \left[\sum_{\tau=0}^{N-1} R_{xw}^i(0, \tau) W_N^{k\tau} \right] - 2 x_a(0)^2 \quad (\text{EQ.7.7})$$

donde $W_N^{k\tau} = e^{-j2\pi k\tau/N}$ y con índice $k=0, \dots, N-1$. Esta última expresión constituye la distribución tiempo frecuencia de la señal real valuada en $n=0$.

Resultados:

Distribución tiempo frecuencia $TFD(n, k)$, valuada en $n=0$ y con índice $k=0, \dots, N-1$.

$TFD(0,0)$...	$TFD(0, \frac{N-1}{2})$	$TFD(0, \frac{N}{2})$ =0	...	$TFD(0, N-1)$ =0
------------	-----	-------------------------	-----------------------------	-----	---------------------

Con respecto a este último resultado, los valores asociados con el índice $k=0, \dots, \frac{N}{2}-1$ representan la parte relevante de la distribución tiempo frecuencia. Nótese que los valores asociados con el índice $k=\frac{N}{2}, \dots, N-1$ valen cero porque la distribución tiempo frecuencia se obtiene de una señal analítica cuyas componentes espectrales negativas son nulas.

7.4 Arquitectura del algoritmo paralelo TFD

Una vez descrito el algoritmo secuencial que calcula las distribuciones tiempo frecuencia, se procederá a describir el algoritmo paralelo correspondiente.

Recapitulando, el algoritmo secuencial consiste de tres pasos. En el primer paso se calcula la señal analítica de la señal real; en el segundo paso se calcula la función de autocorrelación generalizada indexada al tiempo de la señal analítica; y en la tercera etapa se calcula la trasformada de Fourier de la función de autocorrelación generalizada indexada al tiempo, lo cual finalmente constituye la distribución tiempo frecuencia de la señal real. Nótese que los resultados de los pasos primero y segundo son los datos del segundo y tercero respectivamente. A su vez, el primer paso, el cálculo de la señal analítica de la señal real, se puede dividir en dos subpasos. En el primer subpaso se calcula la trasformada de Fourier de la señal real y en el segundo subpaso se calcula tanto el espectro de la señal analítica como su transformada inversa de Fourier. Nuevamente, nótese que el resultado del primer subpaso es el dato del segundo subpaso.

Puesto que en el algoritmo secuencial, el resultado del paso anterior es el dato del paso siguiente, entonces éste se puede distribuir en cuatro unidades de procesamiento. La distribución consiste en lo siguiente. En la primera unidad de procesamiento se procesa el primer subpaso del primer paso del algoritmo secuencial; en la segunda unidad, el segundo

subpaso del primer paso; en la tercera unidad, el segundo paso; y en la cuarta unidad, el tercer paso. Entonces, en un mismo instante de tiempo t , la primera unidad de procesamiento calcula la transformada de Fourier de la señal real de la ventana asociada a ese instante t ; la segunda unidad de procesamiento, la transformada inversa de Fourier de la señal analítica de la ventana de $t-1$; la tercera unidad de procesamiento, la función de autocorrelación generalizada indexada al tiempo de la ventana de $t-2$; y la cuarta unidad de procesamiento, la distribución tiempo frecuencia de la ventana de $t-3$. De igual forma, en el siguiente instante de tiempo $t+1$, en la primera unidad de procesamiento habrá datos de la ventana $t+1$; en la segunda unidad de procesamiento, de la ventana t ; en la tercera unidad de procesamiento, de la ventana $t-1$; y en la cuarta unidad, de la ventana $t-2$. Y así sucesivamente. En la figura FIG.7.2 se muestra la distribución de los cuatro pasos del algoritmo secuencial en sus respectivas unidades de procesamiento a lo largo del tiempo.

Consecuentemente, la arquitectura general del algoritmo paralelo para calcular la distribución tiempo frecuencia de una señal real consiste en una arquitectura de tipo entubamiento (*pipeline*) de 4 etapas^{[31][38]}. En las dos primeras etapas se calcula la señal analítica de la señal real. En la tercera etapa se calcula la función de autocorrelación generalizada indexada al tiempo de la señal analítica. En la cuarta etapa se calcula la distribución tiempo frecuencia de la señal real. En la figura FIG.7.3 se muestra la arquitectura general del algoritmo paralelo para la estimación de la distribución tiempo frecuencia.

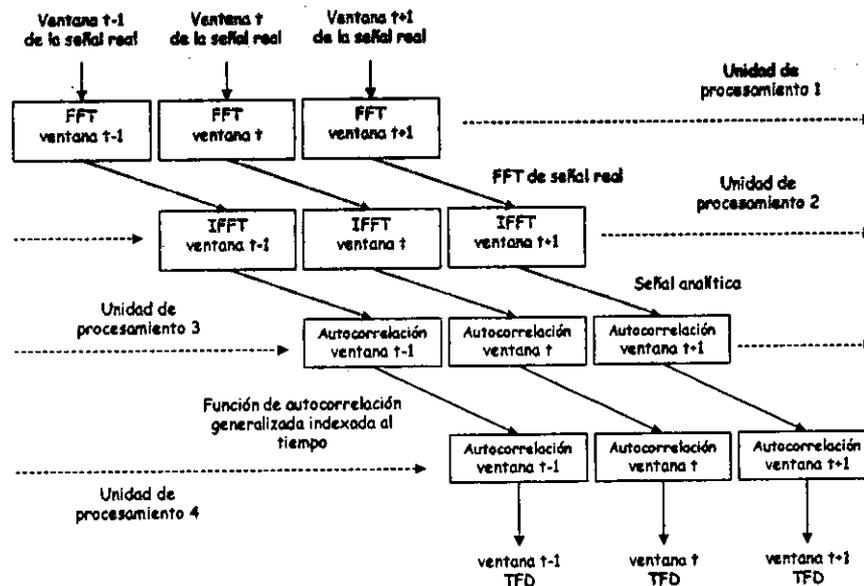


Figura FIG.7.2: Distribución de los pasos del algoritmo secuencial TFD en las unidades de procesamiento a lo largo del tiempo.

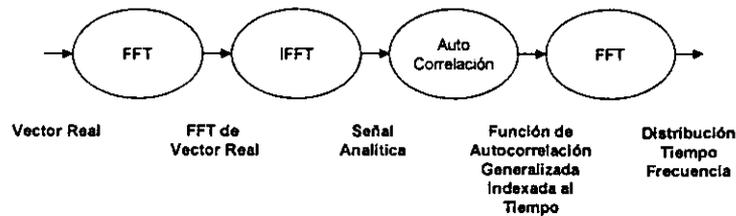


Figura FIG.7.3: Arquitectura general tipo entubamiento (*pipeline*) del algoritmo paralelo para la estimación de la distribución tiempo frecuencia.

El cálculo de la señal analítica del vector real se realiza en las primeras dos etapas del entubamiento (*pipeline*), utilizando el algoritmo de la Transformada Rápida de Fourier (*FFT*) en base 2 de diezmado en frecuencia.

El cálculo de la función de autocorrelación generalizada indexada al tiempo se realiza en la tercera etapa del entubamiento (*pipeline*), utilizando las consideraciones de los algoritmos eficientes propuestos en este trabajo. Ver las ecuaciones (EQ.7.4), (EQ.7.5) y (EQ.7.6)

El cálculo de la distribución tiempo frecuencia se realiza en la cuarta etapa del entubamiento (*pipeline*), utilizando nuevamente el algoritmo de la Transformada Rápida de Fourier (*FFT*) en base 2 de diezmado en frecuencia.

Para cada etapa, a su vez, se puede diseñar un algoritmo paralelo. Para las etapas en donde se calcula una transformada directa o inversa de Fourier, se puede diseñar un algoritmo paralelo *FFT* con una arquitectura de tipo arborescente (*tree*); mientras que para la etapa en donde se calcula una autocorrelación, se puede diseñar un algoritmo paralelo para autocorrelación con una arquitectura de tipo estrella (*star*).

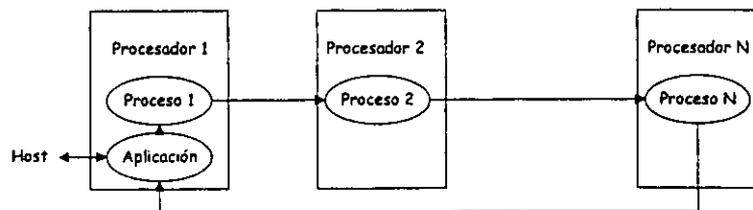


Figura FIG.7.3.c: Topología de unidades de procesamiento en paralelo de tipo anillo (ring) para n procesadores. El proceso de aplicación se encarga de las comunicaciones con el exterior y de la administración de los procesos del entubamiento (*pipeline*).

Ahora bien, la arquitectura del algoritmo paralelo de tipo entubamiento (*pipeline*), se puede implantar en una arquitectura de cómputo paralelo, también llamada topología, de tipo anillo (*ring*). En la figura FIG.7.3.c se muestra la topología de tipo anillo.

7.5 Algoritmo FFT en base 2 de diezmo en frecuencia

Existen diversos algoritmos para calcular la transformada de Fourier; entre ellos está el algoritmo de la Transformada Rápida de Fourier (*FFT, Fast Fourier Transform*), el cual posee variedades, como la de diezmo en tiempo y la de diezmo en frecuencia, ambas para distintas bases. A continuación se plantea el algoritmo FFT en base 2 de diezmo en frecuencia^[15].

7.5.1 Algoritmo secuencial para FFT

Sea la transformada de Fourier $X(k)$ de la señal $x(n)$ de longitud $N=2^p$:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (\text{EQ.7.8})$$

donde $W_N^{kn} = e^{-\frac{j2\pi kn}{N}}$.

Si se reagrupan los términos de (EQ.7.8), entonces se obtiene:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + \sum_{n=N/2}^{N-1} x(n) W_N^{kn} \quad (\text{EQ.7.9})$$

Si en la segunda sumatoria de (EQ.7.9) se cambia el límite inferior $n=N/2$ por $n=0$, entonces se obtiene:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right) W_N^{kn} \quad (\text{EQ.7.10})$$

Si se sustituye $W_N^{Nk/2} = (-1)^k$ en (EQ.7.10) y se reagrupan sus términos, entonces se obtiene:

$$X(k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{kn} \quad (\text{EQ.7.11})$$

Si se sustituye $W_N^2 = W_{N/2}$ en (EQ.7.11) y se dividen los componentes frecuenciales (*diezmado en frecuencia*) pares e impares, entonces se obtiene, para las muestras pares:

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{kn} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (\text{EQ.7.12})$$

y para las muestras impares:

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left[\left\{ x(n) - x\left(n + \frac{N}{2}\right) \right\} W_N^n \right] W_{N/2}^{kn} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (\text{EQ.7.13})$$

Finalmente, si se definen las secuencias de $N/2$ puntos $g_1(n)$ y $g_2(n)$ como:

$$g_1(n) = x(n) + x\left(n + \frac{N}{2}\right) \quad n = 0, 1, 2, \dots, \frac{N}{2} - 1 \quad (\text{EQ.7.14})$$

$$g_2(n) = \left\{ x(n) - x\left(n + \frac{N}{2}\right) \right\} W_N^n \quad n = 0, 1, 2, \dots, \frac{N}{2} - 1 \quad (\text{EQ.7.15})$$

entonces se obtiene que:

$$X(2k) = \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn} \quad (\text{EQ.7.16})$$

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn} \quad (\text{EQ.7.17})$$

Este procedimiento de cálculo puede repetirse diezmando las transformadas de Fourier de $N/2$ puntos $X(2k)$ y $X(2k+1)$. El proceso completo conlleva $\log_2 N$ etapas de diezmando, donde cada etapa implica $N/2$ unidades estructurales de cálculo o mariposas básicas. En la figura FIG.7.4 se muestra la mariposa básica del algoritmo para la FFT en base 2 de diezmando en frecuencia.

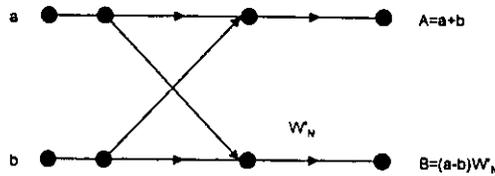


Figura FIG.7.4: Mariposa básica del algoritmo para la FFT en base 2 de diezmando en frecuencia.

En la figura FIG.7.5 se muestra como un ejemplo el algoritmo de la transformada rápida de Fourier en base 2 de diezmando en frecuencia para una señal $x(n)$ con $N = 8$ puntos.

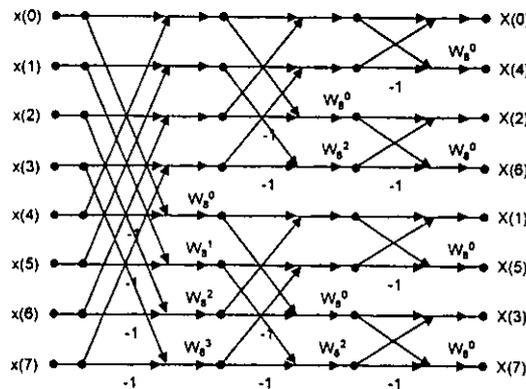


Figura FIG.7.5: Algoritmo para la FFT en base 2 de diezmando en frecuencia para una señal $x(n)$ con $N = 8$.

7.5.2 Algoritmo secuencial para FFT inversa

El algoritmo para calcular la transformada inversa de Fourier es similar al algoritmo para calcular la transformada (directa) de Fourier^[15]. En particular, se puede utilizar el algoritmo FFT en base 2 de diezmando en frecuencia intercambiando los factores W^{kn} por los

factores W^{-kn} y multiplicando solamente el resultado final por $1/N$, es decir, el resultado de la última etapa. Lo anterior obedece a que la transformada inversa de Fourier se define como:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad (\text{EQ.7.18})$$

donde $W_N^{-kn} = e^{\frac{j2\pi kn}{N}}$.

7.6 Arquitectura del algoritmo paralelo para FFT

En el caso de arquitecturas de cómputo paralelas, se prefiere el algoritmo FFT en base 2 por su regularidad estructural y por su complejidad aritmética^[15]. La arquitectura del algoritmo paralelo para calcular la FFT en base 2 de diezclado en frecuencia consiste en una de tipo arborescente (*tree*)^{[31][38]}. En cada proceso (representado por un nodo del árbol), se calcula el algoritmo de su mariposa básica. En la figura FIG.7.6 se muestra la arquitectura del algoritmo paralelo para la FFT.

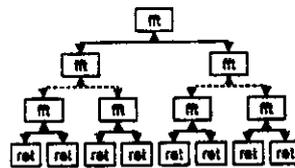


Figura FIG.7.6: Arquitectura del algoritmo paralelo para la FFT en base 2 de diezclado en frecuencia.

Ahora bien, la arquitectura del algoritmo paralelo de tipo arborescente (*tree*), se puede implantar en una arquitectura de cómputo paralelo, también llamada topología, de tipo arborescente (*tree*). En la figura FIG.7.6.c se muestra la topología de tipo arborescente (*tree*).

Otra posibilidad es implantar el algoritmo paralelo en una topología de tipo hipercúbica (*hipercube*).

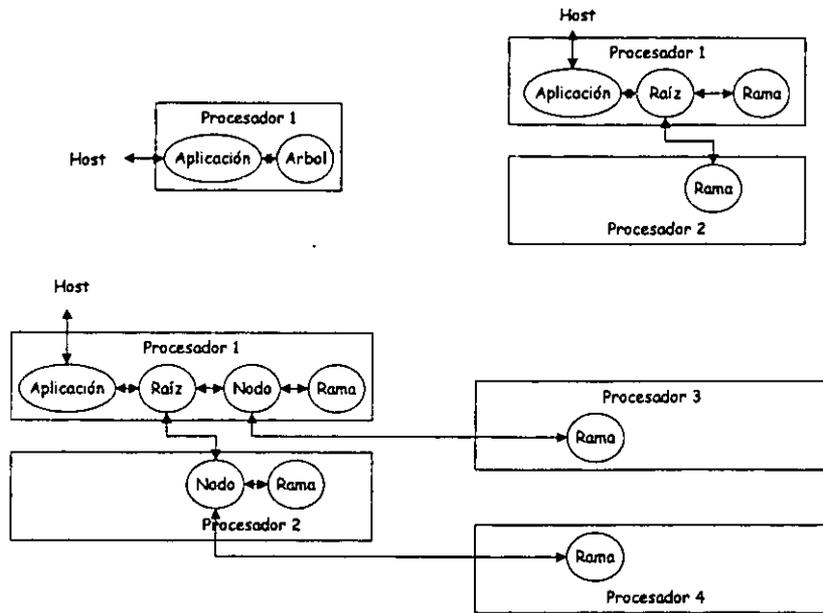


Figura FIG.7.6.c: Topología de unidades de procesamiento en paralelo de tipo arborescente (*tree*) para 1,2 y 4 procesadores. El proceso de aplicación se encarga de las comunicaciones con el exterior y de la administración de los procesos del árbol (*tree*).

7.7 Arquitectura del algoritmo paralelo para autocorrelación

La arquitectura del algoritmo paralelo para calcular la función de autocorrelación generalizada indexada al tiempo consiste en una arquitectura de tipo estrella (*star*)^{[31][38]}. En cada proceso (representado por un nodo de la estrella), se calcula un segmento del algoritmo de la autocorrelación. En la figura FIG.7.7 se muestra la arquitectura del algoritmo paralelo para la autocorrelación.

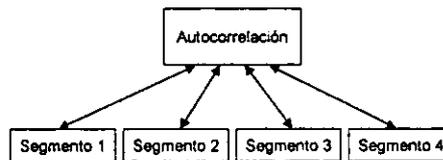


Figura FIG.7.7: Arquitectura del algoritmo paralelo para la función de autocorrelación generalizada indexada al tiempo.

Ahora bien, la arquitectura del algoritmo paralelo de tipo estrella (*star*), se puede implantar en una arquitectura de cómputo paralelo, también llamada topología, de tipo estrella (*star*). En la figura FIG.7.7.c se muestra la topología de tipo estrella (*star*).

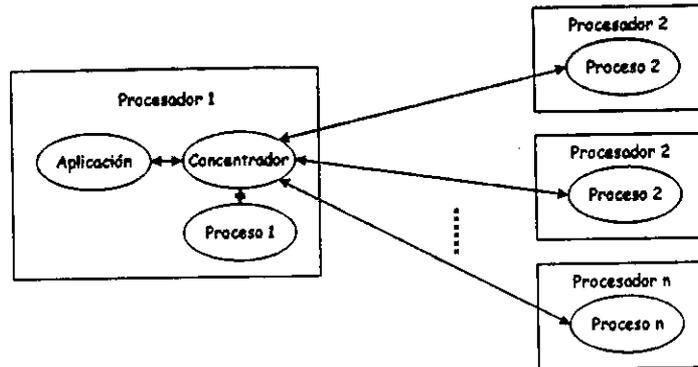


Figura FIG.7.7.c: Topología de unidades de procesamiento en paralelo de tipo estrella (*star*) para n procesadores. El proceso de aplicación se encarga de las comunicaciones con el exterior y de la administración de los procesos de la estrella (*star*).

Otra posibilidad es diseñar el algoritmo paralelo en una arquitectura de tipo granja (*farmer*). Esta última se puede implantar en una topología de tipo lineal (*linear*).

7.8 Medidas de desempeño en procesamiento paralelo

Existen muchas maneras de medir el desempeño de un algoritmo paralelo que se ejecuta en múltiples unidades de procesamiento de manera paralela. Unas de las más comunes son^{[28][29]} la eficiencia (*efficiency*), la aceleración (*speed up*) y la fracción serial (serial fraction).

7.8.1 Eficiencia

La eficiencia (*efficiency*) se define como el tiempo de ejecución del algoritmo en un solo procesador dividido entre p veces el tiempo de ejecución del algoritmo en p procesadores. Una eficiencia que tienda a la unidad es deseable. En el caso ideal, si un algoritmo tardara en ejecutarse t segundos en 1 procesador, entonces ese mismo algoritmo debería tardar en ejecutarse t/p segundos en p procesares. La expresión que calcula la eficiencia es:

$$e = \frac{T_1}{pT_p} \quad (\text{EQ.7.19})$$

7.8.2 Aceleración

La aceleración (*speed up*) se define como el tiempo de ejecución del algoritmo en un solo procesador dividido entre el tiempo de ejecución del algoritmo en p procesadores. Una aceleración lineal es deseable. En el caso ideal, si se duplicara el número de procesadores,

entonces el tiempo de ejecución del algoritmo debería disminuir a la mitad. La expresión que calcula la aceleración es:

$$s = \frac{T_1}{T_p} = ep \quad (\text{EQ.7.20})$$

7.8.3 Fracción serial

La fracción serial se define como el tiempo que toma el segmento del algoritmo que debe ejecutarse serialmente en p procesadores dividido entre el tiempo de ejecución del algoritmo en un solo procesador. Una fracción serial que tienda a cero es deseable. En el caso ideal, si un algoritmo no tuviera segmentos que se debieran calcular serialmente, entonces el algoritmo sería totalmente paralelizable. La expresión que calcula la fracción serial en función de la aceleración y del número de procesadores es:

$$f = \frac{1 - \frac{1}{s}}{1 - \frac{1}{p}} \quad (\text{EQ.7.21})$$

7.9 Arquitectura del equipo de cómputo paralelo

La arquitectura de cómputo paralelo fue fabricada por Inmos y consiste en una tarjeta madre modelo IMS B008 con cuatro procesadores Transputer modelo T805 de 30 Mhz, cada uno con 4Mb de memoria RAM, y con cuatro canales de comunicación seriales de 10Mbit/s a través de un switch de alto desempeño modelo IMS C004. La tarjeta IMS B008 se conecta a una computadora personal a través de una ranura tipo ISA. Al hablar de *procesadores* se hará referencia a los procesadores Transputer; mientras que al hablar de *host* se hará referencia a la computadora personal.

El algoritmo paralelo para calcular la distribución tiempo frecuencia se implanta en diferentes arquitecturas de cómputo paralelo, es decir, en diferentes esquemas de interconexión entre los procesadores para uno, dos y cuatro procesadores.

7.9.1 Arquitectura para dos procesadores

La distribución de procesos para esta arquitectura es la siguiente:

Procesador 1:

Etapa 1: Cálculo de la transformada de Fourier de la señal real.

Etapa 4: Cálculo de la transformada de Fourier de la función de autocorrelación generalizada indexada al tiempo.

Procesador 2:

Etapa 2: Cálculo de la transformada inversa de Fourier de la señal analítica.

Etapa 3: Cálculo de la función de autocorrelación generalizada indexada al tiempo de la señal analítica.

Experimentalmente se determina que la distribución de los diferentes procesos en los 2 procesadores descrita anteriormente es la óptima. En la figura FIG.7.8 se muestra la arquitectura del equipo de cómputo paralelo para dos procesadores.

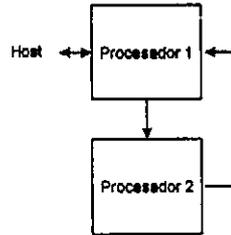


Figura FIG.7.8: Arquitectura del equipo de cómputo paralelo para 2 procesadores.

7.9.2 Arquitectura para cuatro procesadores (tipo I)

La distribución de procesos para esta arquitectura es la siguiente:

Procesador 1:

Etapa 1: Cálculo de la transformada de Fourier de la señal real.

Procesador 2:

Etapa 2: Cálculo de la transformada inversa de Fourier de la señal analítica.

Procesador 3:

Etapa 3: Cálculo de la función de autocorrelación generalizada indexada al tiempo de la señal analítica.

Procesador 4:

Etapa 4: Cálculo de la transformada de Fourier de la función de autocorrelación generalizada indexada al tiempo.

En la figura FIG.7.9 se muestra la arquitectura de tipo I del equipo de cómputo paralelo para cuatro procesadores.

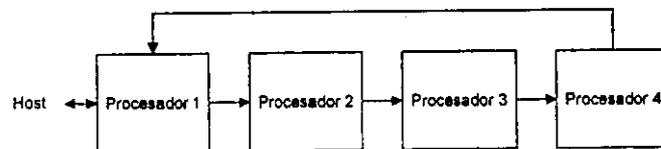


Figura FIG.7.9: Arquitectura tipo I del equipo de cómputo paralelo para 4 procesadores

7.9.3 Arquitectura para cuatro procesadores (tipo II)

La distribución de procesos para esta arquitectura es la siguiente:

Procesador 1:

Etapa 1: Cálculo de la transformada de Fourier de la señal real.

Etapa 4: Cálculo de la transformada de Fourier de la función de autocorrelación generalizada indexada al tiempo.

Procesador 2:

Etapa 2: Cálculo de la transformada inversa de Fourier de la señal analítica.

Procesador 3 y 4:

Etapa 3: Cálculo de la función de autocorrelación generalizada indexada al tiempo de la señal analítica.

Debido a que la complejidad algorítmica del cálculo de la función de autocorrelación generalizada indexada al tiempo es de mayor orden que la complejidad algorítmica del cálculo de la transformada directa o inversa de Fourier, se dirigen los esfuerzos de paralelización en el primero. En la figura FIG.7.10 se muestra la arquitectura de tipo II del equipo de cómputo paralelo para cuatro procesadores.

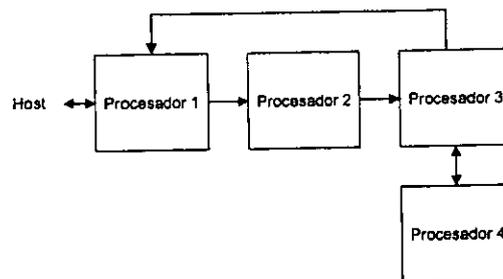


Figura FIG.7.10: Arquitectura tipo II del equipo de cómputo paralelo para 4 procesadores

7.10 Desempeños de los algoritmos paralelos

Se medirá el desempeño para los algoritmos paralelos que calculan la distribución tiempo frecuencia de Wigner Ville (EQ.4.32), de Choi Williams (EQ.5.39) y de Bessel (EQ.6.39) implantados en las arquitecturas de cómputo paralelo para uno, dos y cuatro procesadores en sus dos variedades, (Ver figuras FIG.7.6, FIG.7.7 y FIG.7.8). Las mediciones del desempeño serán las de eficiencia (EQ.7.20), aceleración (EQ.7.21) y fracción serial (EQ.7.22), así como el tiempo de ejecución.

7.10.1 Desempeño del algoritmo para la distribución tiempo frecuencia de Wigner Ville

En las tablas TB.7.1, TB.7.2, TB.7.3 y TB.7.4 se muestran los desempeños de tiempo de ejecución, aceleración, eficiencia y fracción serial, respectivamente, del algoritmo paralelo para estimar la distribución tiempo frecuencia de Wigner Ville. En la tabla TB.7.1 se sombrea

Capítulo 7. Implantación de Algoritmos Paralelos para Estimación Espectral

el tiempo de ejecución más rápido entre las configuraciones de cuatro procesadores, con el objeto de resaltar cuál de ellas es más eficiente respecto a la longitud de ventana de muestreo.

Tiempo de ejecución [ms]	Número de procesadores				
	1	1	2	4	4
Longitud de Ventana	Sin conf.		con conf. #1		con conf. #2
15	8.54	8.46	5.52	4.09	5.55
31	19.09	18.73	11.65	8.69	11.69
63	42.00	41.16	25.08	18.66	25.20
127	91.47	89.82	54.22	40.20	54.67
255			117.54	86.51	118.33

Tabla TB.7.1: Tiempo de ejecución del algoritmo para la distribución de Wigner Ville.

Aceleración	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		1.53	2.07	1.52
31		1.61	2.15	1.60
63		1.64	2.21	1.63
127		1.66	2.23	1.64
255				

Tabla TB.7.2: Aceleración del algoritmo para la distribución de Wigner Ville.

Eficiencia	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		77%	52%	38%
31		80%	54%	40%
63		82%	55%	41%
127		83%	56%	41%

Tabla TB.7.3: Eficiencia del algoritmo para la distribución de Wigner Ville.

Fracción serial	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		0.31	0.31	0.54
31		0.24	0.29	0.50
63		0.22	0.27	0.48
127		0.21	0.26	0.48
255				

Tabla TB.7.4: Fracción serial del algoritmo para la distribución de Wigner Ville.

7.10.2 Desempeño del algoritmo para la distribución tiempo frecuencia de Choi Williams

En las tablas TB.7.5, TB.7.6, TB.7.7 y TB.7.8 se muestran los desempeños de tiempo de ejecución, aceleración, eficiencia y fracción serial, respectivamente, del algoritmo paralelo para estimar la distribución tiempo frecuencia de Choi Williams. En la tabla TB.7.5 se sombrea el tiempo de ejecución más rápido entre las configuraciones de cuatro procesadores, con el objeto de resaltar cuál de ellas es más eficiente respecto a la longitud de ventana de muestreo.

Tiempo de ejecución [ms]	Número de procesadores				
	1	1	2	4	4
Longitud de Ventana	sin conf.		Con conf. #1		con conf. #2
15	9.65	9.59	5.69	4.13	5.56
31	23.71	23.36	13.79	8.79	11.76
63	61.64	60.79	38.73	23.67	25.40
127	168.98	167.32	118.08	84.92	65.93
255			420.86	341.57	255.61

Tabla TB.7.5: Tiempo de ejecución del algoritmo para la distribución de Choi Williams.

Aceleración	Número de procesadores			
	1	2	4	4
Longitud de Ventana		Con conf. #1		con conf. #2
15		1.69	2.32	1.72
31		1.69	2.66	1.99
63		1.57	2.57	2.39
127		1.42	1.97	2.54
255				

Tabla TB.7.6: Aceleración del algoritmo para la distribución de Choi Williams.

Eficiencia	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		84%	58%	43%
31		85%	66%	50%
63		78%	64%	60%
127		71%	49%	63%
255				

Tabla TB.7.7: Eficiencia del algoritmo para la distribución de Choi Williams.

Fracción serial	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		0.19	0.24	0.44
31		0.18	0.17	0.34
63		0.27	0.19	0.22
127		0.41	0.34	0.19
255				

Tabla TB.7.8: Fracción serial del algoritmo para la distribución de Choi Williams.

7.10.3 Desempeño del algoritmo para la distribución tiempo frecuencia de Bessel

En las tablas TB.7.9, TB.7.10, TB.7.11 y TB.7.12 se muestran los desempeños de tiempo de ejecución, aceleración, eficiencia y fracción serial, respectivamente, del algoritmo paralelo para estimar la distribución tiempo frecuencia de Bessel. En la tabla TB.7.12 se sombrea el tiempo de ejecución más rápido entre las configuraciones de cuatro procesadores, con el objeto de resaltar cuál de ellas es más eficiente respecto a la longitud de ventana de muestreo.

Tiempo de ejecución [ms]	Número de procesadores				
	1	1	2	4	4
Longitud de Ventana	sin conf.		con conf. #1		con conf. #2
15	9.38	9.28	5.55	4.11	5.56
31	22.84	22.39	12.82	8.85	11.74
63	57.27	56.24	34.17	19.13	25.33
127	154.78	152.73	103.49	70.33	55.26
255			337.36	265.11	186.84

Tabla TB.7.9: Tiempo de ejecución del algoritmo para la distribución de Bessel.

Aceleración	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		1.67	2.26	1.67
31		1.75	2.53	1.91
63		1.65	2.94	2.22
127		1.48	2.17	2.76
255				

Tabla TB.7.10: Aceleración del algoritmo para la distribución de Bessel.

Eficiencia	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		84%	56%	42%
31		87%	63%	48%
63		82%	73%	56%
127		74%	54%	69%
255				

Tabla TB.7.11: Eficiencia del algoritmo para la distribución de Bessel.

Fracción serial	Número de procesadores			
	1	2	4	4
Longitud de Ventana		con conf. #1		con conf. #2
15		0.20	0.26	0.47
31		0.15	0.19	0.37
63		0.22	0.12	0.27
127		0.36	0.28	0.15
255				

Tabla TB.7.12: Fracción serial del algoritmo para la distribución de Bessel.

7.10.4 Comparación de desempeños

En las figuras FIG.7.11, FIG.7.12, FIG.7.13, FIG.7.14, FIG.7.15 y FIG.7.16 se muestran gráficas que comparan los tiempos de ejecución en milisegundos contra la longitud de ventana de muestreo, de las distribuciones de Wigner Ville, Choi Williams y Bessel, para las arquitecturas de cómputo paralelo de un procesador, dos procesadores y cuatro procesadores tipo I y II.

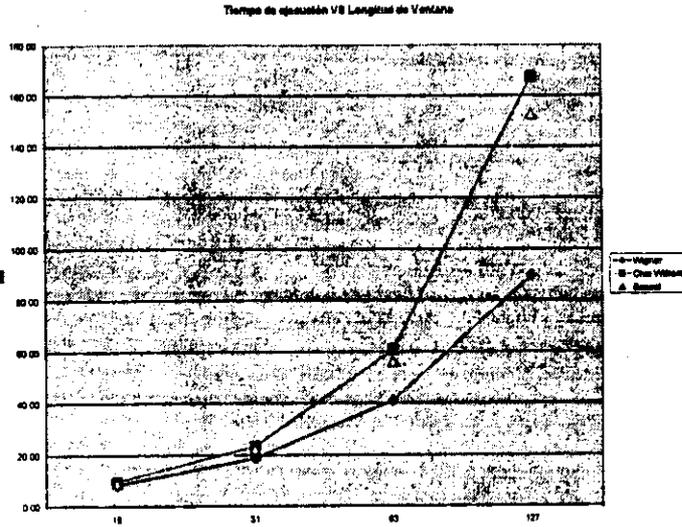


Figura FIG.7.11: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para las distribuciones tiempo frecuencia de Wigner Ville, Choi Williams y Bessel, para una arquitectura de cómputo paralelo de un procesador.

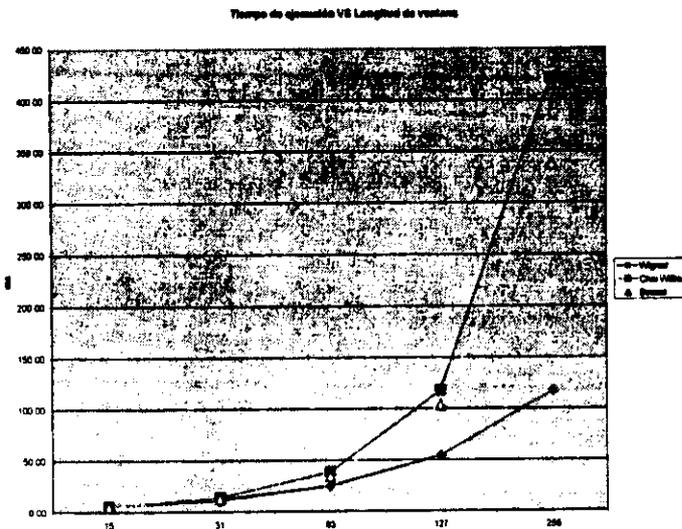


Figura FIG.7.12: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para las distribuciones tiempo frecuencia de Wigner Ville, Choi Williams y Bessel, para una arquitectura de cómputo paralelo de dos procesadores.

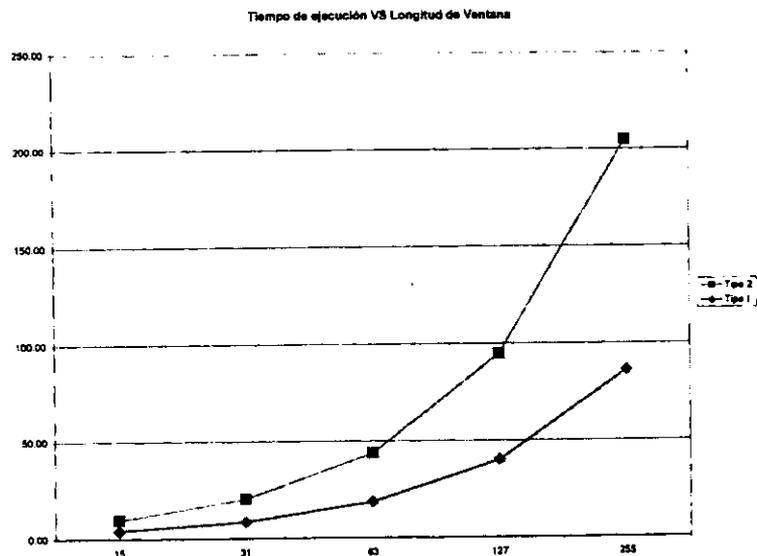


Figura FIG.7.13: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para la distribución tiempo frecuencia de Wigner Ville, para una arquitectura de cómputo paralelo de cuatro procesadores tipo I y II.

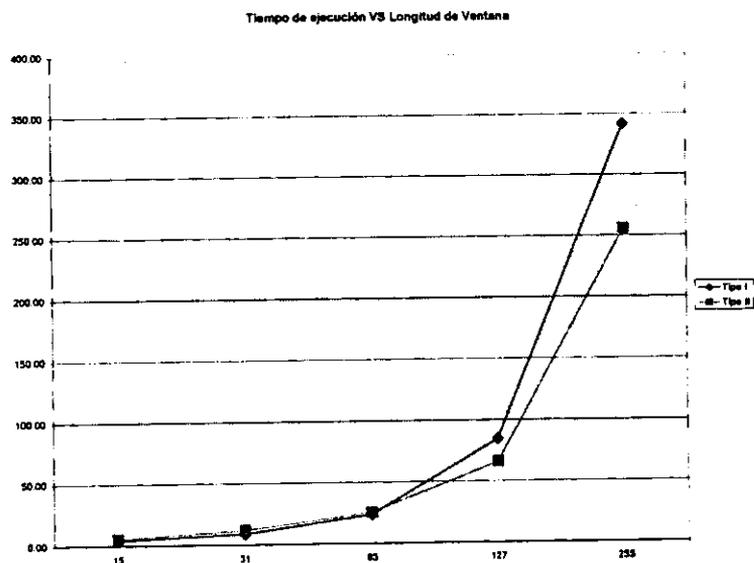


Figura FIG.7.14: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para la distribución tiempo frecuencia de Choi Williams, para una arquitectura de cómputo paralelo de cuatro procesadores tipo I y II.

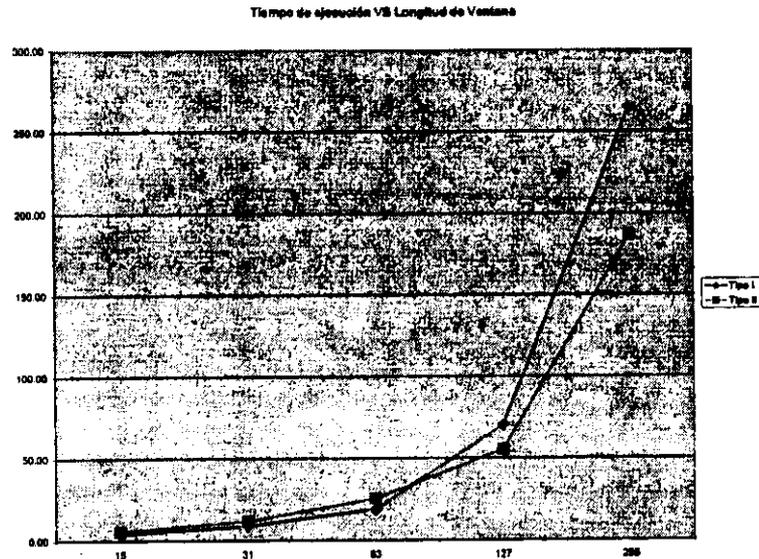


Figura FIG.7.15: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para la distribución tiempo frecuencia de Bessel, para una arquitectura de cómputo paralelo de cuatro procesadores tipo I y II.

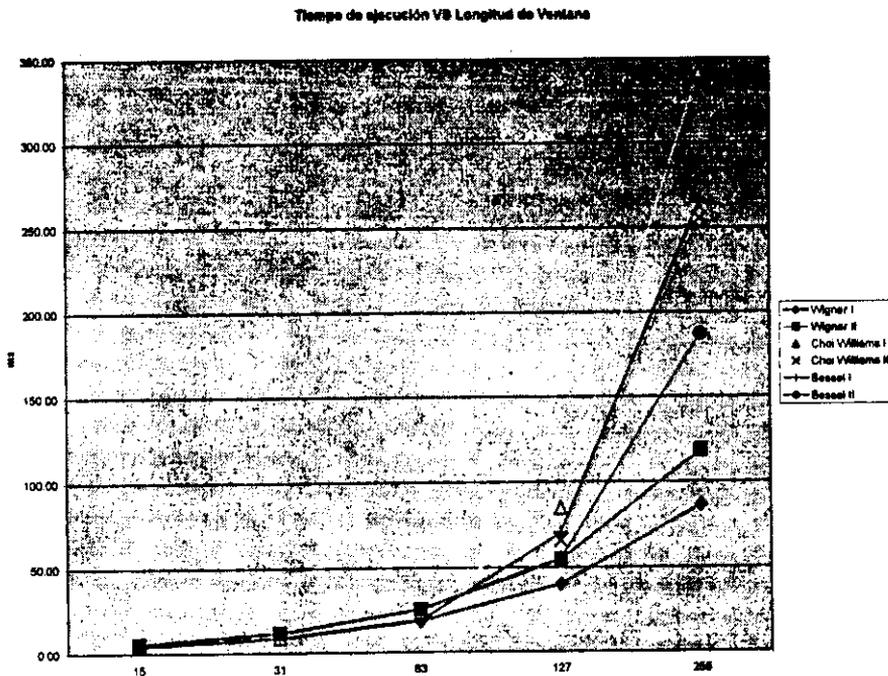


Figura FIG.7.16: Gráfica de tiempos de ejecución contra longitud de ventana de muestreo, para las distribuciones tiempo frecuencia de Wigner Ville, Choi Williams y Bessel, para una arquitectura de cómputo paralelo de cuatro procesadores tipo I y II.

8 ANÁLISIS DE RESULTADOS

8.1 Introducción

En este capítulo se presenta el análisis de resultados correspondientes a cada distribución tiempo frecuencia y a la comparación entre ellas. Primero se analiza el número de operaciones que se realiza al calcular las distribuciones tiempo frecuencia. Luego, se analiza la selección de los parámetros σ y α para las distribuciones de Choi Williams y de Bessel. A continuación, se proporciona un criterio de evaluación de las distribuciones tiempo frecuencia con base en el factor de peso de los términos de cruce. También, se presenta el análisis de resultados tanto de los experimentos numéricos (influencia de SNR) como de la aplicación a una señal tipo Dópler ultrasónica (influencia de longitud de ventana y de SNR). Finalmente, se discuten los tiempos de ejecución de los algoritmos paralelos de estimación espectral.

8.2 Número de operaciones que se realizan al calcular las distribuciones tiempo frecuencia.

Se analizan los resultados obtenidos en las secciones 4.6, 5.6 y 6.6, Algoritmo eficiente, de los capítulos 4, 5 y 6, Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel, respectivamente. En las secciones 4.6.1, 5.6.1 y 6.6.1 se muestran tablas del número de operaciones que se realizan en el cálculo de las distribuciones discretas de Wigner Ville, Choi Williams y Bessel para cada valor de k (componente frecuencial), respecto a la longitud de ventana de muestreo N , según su definición:

	Wigner Ville (EQ.4.13)	Choi Williams (EQ.5.13)	Bessel (EQ.6.13)
Multiplicaciones complejas	$6N-3$	$8N^2-4N$	$8\alpha N^2-8\alpha N$
Sumas complejas	$2N-2$	$4N^2-6N+2$	$4\alpha N^2-4\alpha N-2N$
Multiplicaciones por escalares	1	1	1

Tabla TB.8.1: Número de operaciones que se realiza en el cálculo de las distribuciones tiempo frecuencia según su definición para cada componente frecuencial. N es la longitud de ventana de muestreo.

y según su expresión simplificada:

	Wigner Ville (EQ.4.32)	Choi Williams (EQ.5.39)	Bessel (EQ.6.39)
Multiplicaciones complejas	$3N+1$	$2N^2-2N+1$	Menos $4\alpha N^2-4\alpha N+1$
Sumas complejas	N	N^2-2N	Menos $2\alpha N^2-2\alpha N-N$
Multiplicaciones por escalares	2	2	2

Tabla TB.8.2: Número de operaciones que se realiza en el cálculo de las distribuciones tiempo frecuencia según las expresiones simplificadas para cada componente frecuencial. N es la longitud de ventana de muestreo.

Se observa que el cálculo de la distribución de Wigner Ville para cada k (componente frecuencial), es de orden N , mientras que tanto el de Choi Williams como el de Bessel para cada k (componente frecuencial) es de orden N^2 .

Si bien las expresiones propuestas simplificadas algebraicamente conservan el mismo orden que las expresiones basadas en su definición, para calcular las primeras solamente se debe realizar, en todos los casos, a lo más, la mitad de las operaciones que se requieren para calcular las segundas.

8.3 Selección de parámetros σ y α para las distribuciones de Choi Williams y de Bessel

Se analizan los resultados obtenidos en las secciones 5.8 y 6.8, Aplicación a una señal Dópler ultrasónica simulada en Matlab, de los capítulos 5 y 6, Distribución de Choi Williams y Distribución de Bessel, respectivamente. En las secciones 5.8.8 y 6.8.8 se muestran gráficas del error en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS en función del parámetro σ , sigma, para la distribución de Choi Williams y del parámetro α , alfa, para la distribución de Bessel.

Para ambas distribuciones, se observa que la función de error en la estimación tanto de la frecuencia media pseudo instantánea como del ancho de banda espectral RMS, tienen un único mínimo. Para el caso de la distribución de Choi Williams, el mínimo se da en $\sigma = 5$; mientras que para la de Bessel, en $\alpha = 2$. Refiérase a las figuras FIG 8.1.a y FIG 8.1.b.

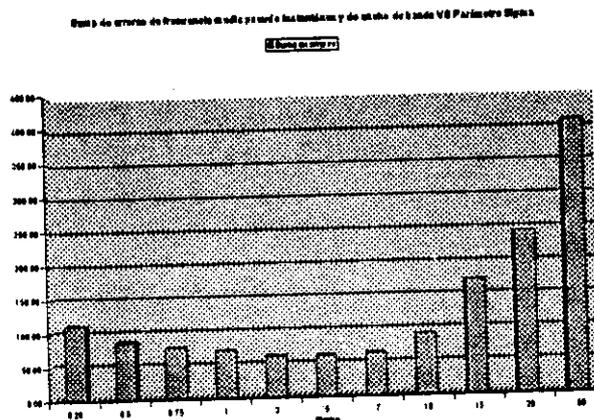


Figura FIG 8.1.a: Suma de errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda contra el parámetro sigma de la distribución de Choi Williams

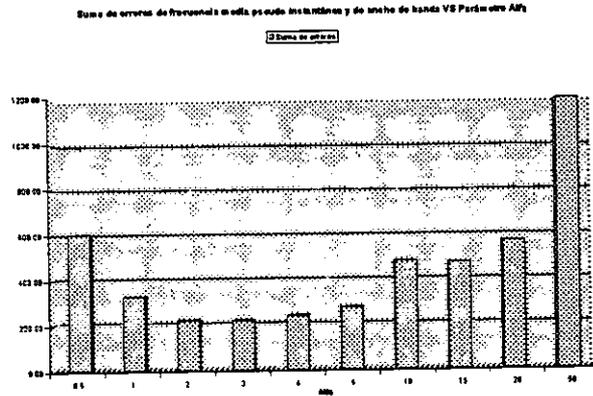


Figura FIG 8.1. b: Suma de errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda contra el parámetro sigma de la distribución de Bessel.

El método para obtener los parámetros óptimos es experimental. Primero se genera una función que relaciona, por extensión, el error en la estimación tanto de la frecuencia media pseudo instantánea como del ancho de banda espectral RMS contra el valor del parámetro utilizado. Finalmente se determina el mínimo de la función generada por inspección. Este mismo método se puede utilizar para determinar el valor de los parámetros óptimos de otras distribuciones tiempo frecuencia. Es importante mencionar que los resultados obtenidos dependen de la señal que se estudia: una señal tipo Dópler ultrasónica.

8.4 Evaluación de las distribuciones tiempo frecuencia con base en el factor de peso de los términos de cruce

Con el objeto de establecer un criterio de comparación entre las diferentes distribuciones tiempo frecuencia, es necesario desarrollar un método que determine el grado en el cual los diversos componentes de una señal interactúan entre sí. Para tal efecto, considérese la señal $x(t)$ la cual se compone de un número finito de señales sinusoidales con amplitud A_n , frecuencia ω_n y fase θ_n constantes:

$$x(t) = \sum_{n=1}^N A_n e^{j(\omega_n t + \theta_n)} \quad (\text{EQ.8.1})$$

Al efectuar el cálculo de la distribución tiempo frecuencia según (EQ.4.7), (EQ.5.7) y (EQ.6.7), las n componentes de la señal descrita en (EQ.8.1) interactúan entre ellas a través de la función de autocorrelación instantánea descrita en (EQ.4.4), (EQ.5.4) y (EQ.6.4). Las interacciones de los componentes consigo mismos generan los llamados autotérminos de la distribución, los cuales siempre son positivos y constituye el contenido espectral de la señal (distribución de energía) propiamente dicho. Por otra parte, las interacciones entre componentes diferentes generan los llamados términos de cruce de la distribución, los cuales pueden ser positivos o negativos y se añaden al contenido espectral de la señal (distribución de

energía) ocultándolo. De aquí surge la necesidad de minimizar los términos de cruce de una distribución.

Ahora bien, si se reagrupan los autotérminos y los términos de cruce en una distribución tiempo frecuencia, esta última se puede expresar como:

$$TFD(t, \omega) = TFD_{auto}(t, \omega) + TFD_{cruce}(t, \omega) \quad (EQ.8.2)$$

donde los autotérminos son:

$$TFD_{auto}(t, \omega) = 2\pi \sum_{n=1}^N A_n^2 \delta(\omega - \omega_n) \quad (EQ.8.3)$$

y los términos de cruce son:

$$TFD_{cruce}(t, \omega) = \sum_{n=1}^N \sum_{m=1}^N F_{TFD}(\omega) A_n A_m \cos((\omega_n - \omega_m)t + \theta_n - \theta_m) \quad (EQ.8.4)$$

El término $F_{TFD}(\omega)$ de (EQ.8.4) recibe el nombre de factor de peso de los términos de cruce y es una medida cuantitativa para evaluar las diferentes distribuciones tiempo frecuencia.

Un factor de peso de los términos de cruce ideal sería aquel que los eliminara. Situaciones deseables, fuera de la ideal, serían que el factor de peso concentrara los términos de cruce debidos a dos componentes frecuenciales diferentes alrededor de tales frecuencias y no alrededor de otras frecuencias, o bien, que los diseminara adecuadamente en un amplio rango de frecuencias. La primera situación provoca que una distribución estime con mayor precisión los parámetros espectrales de una señal con un ancho de banda pequeño en ausencia de ruido; mientras que la segunda situación provoca que una distribución estime con mayor precisión los parámetros espectrales de una señal en presencia de ruido.

Los factores de peso de los términos de cruce se enuncian a continuación. Primero, según (EQ.4.24), el factor de peso para la distribución de Wigner Ville es:

$$F_{WVD}(\omega) = 2\pi \delta\left(\omega - \frac{\omega_n + \omega_m}{2}\right) \quad (EQ.8.5)$$

Segundo, según (EQ.5.30), el factor de peso para la distribución de Choi Williams es:

$$F_{CWD}(\omega) = \frac{\pi\sigma}{\sqrt{(\omega_n - \omega_m)^2}} e^{-\frac{\sigma}{4(\omega_n - \omega_m)^2} \left(\omega - \frac{\omega_n + \omega_m}{2}\right)^2} \quad (EQ.8.6)$$

Finalmente, según (EQ.6.30), el factor de peso para la distribución de Bessel es:

$$F_{BD}(\omega) = \frac{4}{\alpha|\omega_n - \omega_m|} U_0 \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)} \right) \left| 1 - \left(\frac{\omega - \frac{\omega_n + \omega_m}{2}}{\alpha(\omega_n - \omega_m)} \right)^2 \right| \quad (EQ.8.7)$$

En las figuras FIG.8.2.a, FIG.8.2.b y FIG.8.2.c se muestran vistas globales de los factores de peso de los términos de cruce de las distribuciones de Wigner Ville, Choi Williams y Bessel respectivamente. Nótese que los factores de peso son funciones de ω y contienen como parámetros las frecuencias de las componentes sinusoidales que interactúan entre sí, es decir, ω_n y ω_m . Tales frecuencias aparecen sumándose o restándose de manera particular en (EQ.8.5), (EQ.8.6) y (EQ.8.7). Con el objeto de obtener gráficas comparables, normalícese su suma, esto es:

$$\omega_n + \omega_m = 1 \tag{EQ.8.8}$$

donde $0 < \omega_n < 1$ y $\omega_m = 1 - \omega_n$. De esta forma, las gráficas relacionan solamente el factor de peso contra ω y ω_n . Por ejemplo, si ω_n y ω_m son iguales, al normalizar su suma resulta que $\omega_n = \omega_m = 0.5$; si ω_n es mayor que ω_m , al normalizar su suma resulta que $\omega_n > 0.5$ y si es menor entonces $\omega_n < 0.5$.

De esta forma, el eje correspondiente a ω_m (el mismo que el de ω_n) representa el par de componentes frecuenciales que interactúan entre sí; mientras que, el eje correspondiente a ω representa el rango de frecuencias en donde se disemina la energía generada por el par de componentes frecuenciales que interactúan entre sí.

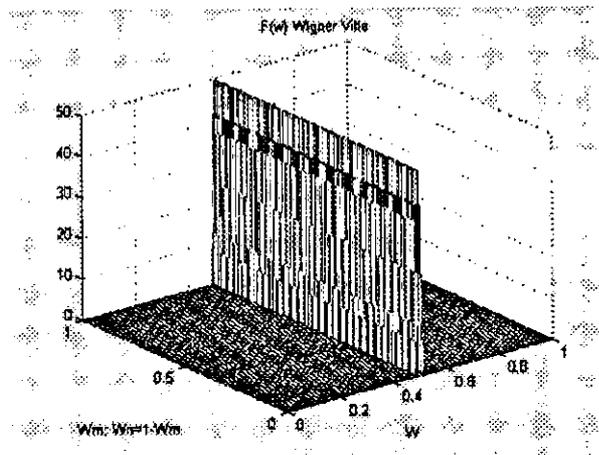


Figura FIG.8.2.a: Factor de peso de los términos de cruce para la distribución de Wigner Ville.

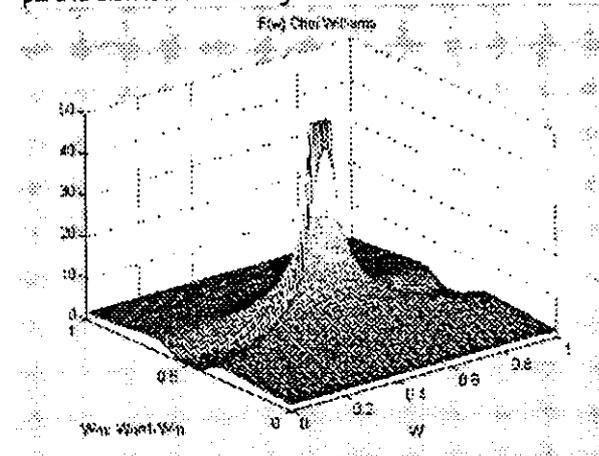


Figura FIG.8.2.b: Factor de peso de los términos de cruce para la distribución de Choi Williams con $\sigma = 5$.

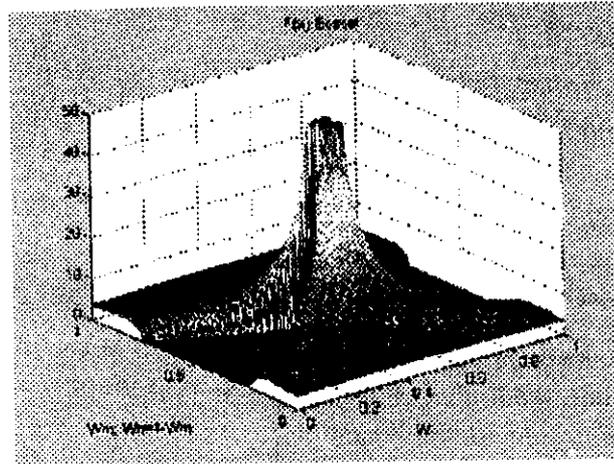


Figura FIG.8.2.c: Factor de peso de los términos de cruce para la distribución de Bessel con $\alpha = 2$.

Con el objeto de ilustrar la comparación entre dos factores de peso, en la figura FIG.8.3 se muestra una gráfica de la diferencia entre el factor de peso de los términos de cruce de la distribución de Choi Williams menos el de la distribución de Bessel.

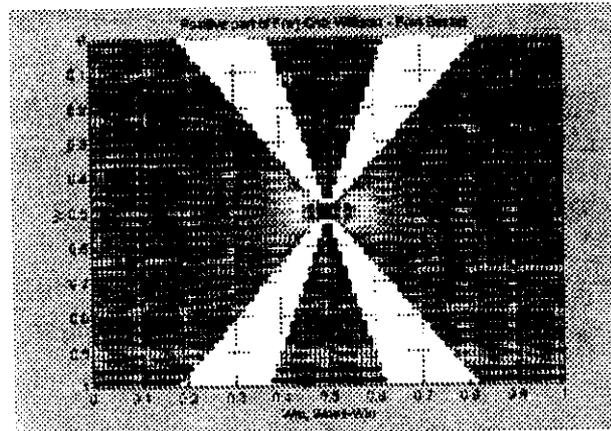


Figura FIG.8.3: Parte positiva (en oscuro) de la diferencia entre los factores de peso de la distribución de Choi Williams menos el de la distribución de Bessel.

Los puntos oscuros en la gráfica de la figura FIG.8.3 corresponden a los puntos donde el factor de peso de los términos de cruce de la distribución de Choi Williams es mayor que el de la distribución de Bessel. Se observa que el factor de peso de la distribución de Choi Williams con $\sigma = 5$ es generalmente mayor que el factor de peso de la distribución de Bessel con $\alpha = 2$. Este resultado indica que el factor de peso de la distribución de Bessel con $\alpha = 2$ dispersa mejor los términos de cruce que el de la distribución de Choi Williams con $\sigma = 5$. Esto podría explicar la razón de por qué la distribución de Bessel es menos sensible a la presencia de ruido que la distribución de Choi Williams ^{[10][16]}.

En las figuras FIG.8.4.a y FIG.8.4.b se muestran gráficas de las partes positiva y negativa de la diferencia entre los factores de peso de la distribución de Choi Williams menos el de la distribución de Bessel.

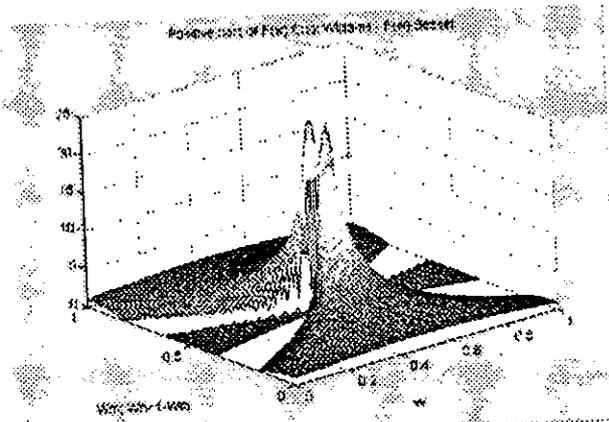


Figure FIG.8.4.a: Parte positiva de la diferencia entre los factores de peso de la distribución de Choi Williams menos el de la distribución de Bessel. Puntos en donde la distribución de Choi Williams es mayor.

Los puntos oscuros en la gráfica de la figura FIG.8.4.a corresponden a los puntos donde el factor de peso de los términos de cruce de la distribución de Choi Williams es mayor que el de la distribución de Bessel.

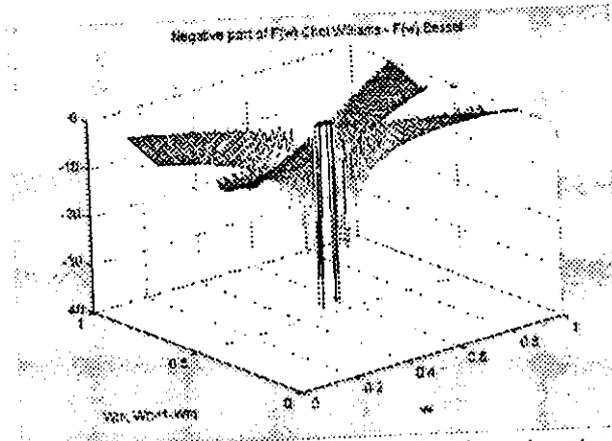


Figure FIG.8.4.b: Parte negativa de la diferencia entre los factores de peso de la distribución de Choi Williams menos el de la distribución de Bessel. Puntos en donde la distribución de Bessel es mayor.

Los puntos oscuros en la gráfica de la figura FIG.8.4.b corresponden a los puntos donde el factor de peso de los términos de cruce de la distribución de Bessel es mayor que el de la distribución de Choi Williams.

8.5 Experimentos numéricos en Matlab

El objetivo de estos experimentos fue caracterizar el comportamiento de la estimación de la frecuencia media pseudo instantánea con respecto a la variación del nivel de potencia del ruido presente en la señal (SNR).

Se analizarán los resultados obtenidos en las secciones 4.7, 5.7 y 6.7, Experimentos Numéricos en Matlab, de los capítulos 4, 5 y 6, Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel, respectivamente. En las secciones 4.7.6, 5.7.6 y 6.7.6 se muestran tablas que concentran los errores en la estimación de la frecuencia media pseudo instantánea con respecto a SNR en los diferentes experimentos.

En el experimento numérico 1, función seno con frecuencia de 0.32 Hz y amplitud constante, la estimación de la frecuencia media pseudo instantánea con base en la distribución de Bessel es la mejor para cualquier SNR. Véase la figura FIG.8.5.

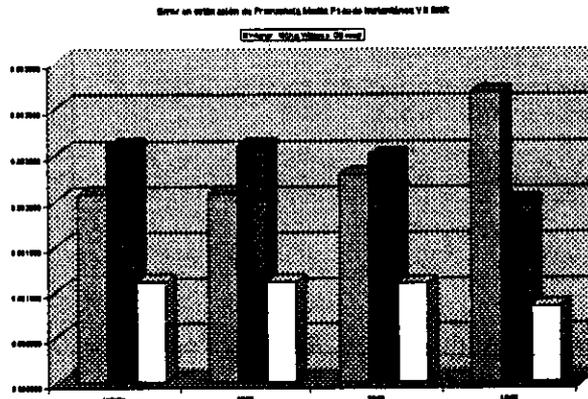


Figura FIG.8.5: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR en el experimento numérico 1, función seno con frecuencia de 0.32 Hz y amplitud constante.

Obsérvese en la figura FIG.8.5, que la función de error en la estimación de la frecuencia media pseudo instantánea es creciente con respecto a SNR para el caso de la distribución de Wigner Ville, mientras que es decreciente para la distribución de Choi Williams y de Bessel.

En el experimento numérico 2, función suma de senos con frecuencias de 0.40 Hz y 0.16 Hz y amplitudes constantes, la estimación de la frecuencia media pseudo instantánea con base en la distribución de Wigner Ville es la mejor para una SNR infinita, de 40dB y de 20dB; mientras que en la de Choi Williams para una SNR de 10dB. Véase la figura FIG.8.6.

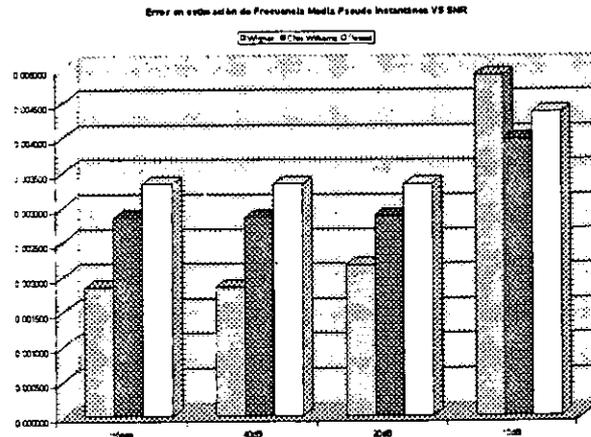


Figura FIG.8.6: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR en el experimento numérico 2, función suma de senos con frecuencias de 0.40 Hz y 0.16 Hz y amplitudes constantes.

Obsérvese en la figura FIG.8.6, que la función de error en la estimación de la frecuencia media pseudo instantánea es creciente con respecto a SNR. Sin embargo, la tasa de crecimiento del error es mayor para la distribución de Wigner Ville que para las distribuciones de Choi Williams y de Bessel.

En el experimento numérico 3, función suma de senos con frecuencias de 0.40 Hz y 0.08 Hz y amplitudes constantes, la estimación de la frecuencia media pseudo instantánea con base en la distribución de Bessel es la mejor para cualquier SNR. Véase la figura FIG.8.7.

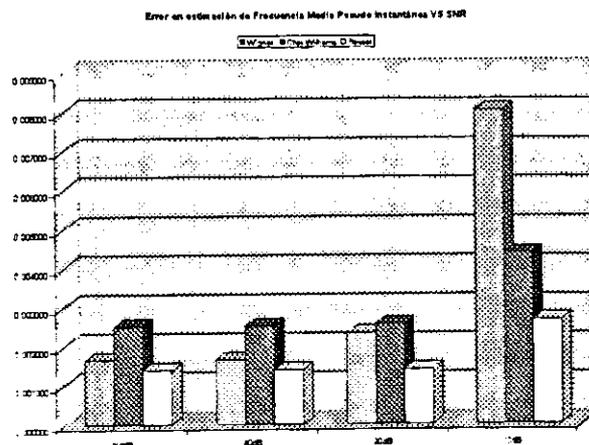


Figura FIG.8.7: Error en la estimación de la frecuencia media pseudo instantánea con respecto a SNR en el experimento numérico 3, función suma de senos con frecuencias de 0.40 Hz y 0.08 Hz y amplitudes constantes.

Obsérvese en la figura FIG.8.7, que la función de error en la estimación de la frecuencia media pseudo instantánea es creciente con respecto a SNR. Sin embargo, la tasa de

crecimiento del error es mayor para la distribución de Wigner Ville que para las distribuciones de Choi Williams y de Bessel.

En estos tres experimentos se muestra que las distribuciones de Choi Williams y de Bessel estiman con mayor precisión los parámetros espectrales de una señal en presencia de ruido que la distribución de Wigner Ville. Específicamente, en los experimentos 1 y 3 se muestra que la distribución de Bessel estima con mayor precisión los parámetros espectrales de una señal en presencia de ruido que la distribución de Choi Williams: la función de error de la primera crece a una tasa menor que la de la segunda a medida que el ruido se incrementa.

8.6 Aplicación a una señal Dópler ultrasónica simulada en Matlab

El objetivo de estos experimentos fue caracterizar el comportamiento de la estimación de la frecuencia media pseudo instantánea y de la estimación del ancho de banda espectral RMS con respecto a la variación de la longitud de la ventana de muestreo y de la relación de potencias de la señal y el ruido (SNR).

Se analizarán los resultados obtenidos en las secciones 4.8, 5.8 y 6.8, Aplicación a una señal Dópler ultrasónica simulada en Matlab, de los capítulos 4, 5 y 6, Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel, respectivamente. En las secciones 4.8.7, 5.8.7 y 6.8.7 se muestran tablas que concentran los errores en la estimación de la frecuencia media pseudo instantánea y en la estimación de ancho de banda espectral RMS con respecto a la longitud de la ventana de muestreo y a SNR en los diferentes experimentos.

Primero se analizará la estimación de la frecuencia media pseudo instantánea y luego la del ancho de banda espectral RMS. Posteriormente se agruparán los resultados obtenidos por transformada.

8.6.1 Análisis de la estimación de la frecuencia media pseudo instantánea

Nótese que, en general, las funciones de error en la estimación de la frecuencia media pseudo instantánea tienen un punto mínimo.

En el caso de los experimentos sin presencia de ruido se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 31 elementos; la de Choi Williams, con una de 63; y la de Bessel, con una de 127. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Wigner Ville, excepto para una longitud de ventana de 127 elementos que fue hecha por la de Bessel. Ver figura FIG.8.8.a.

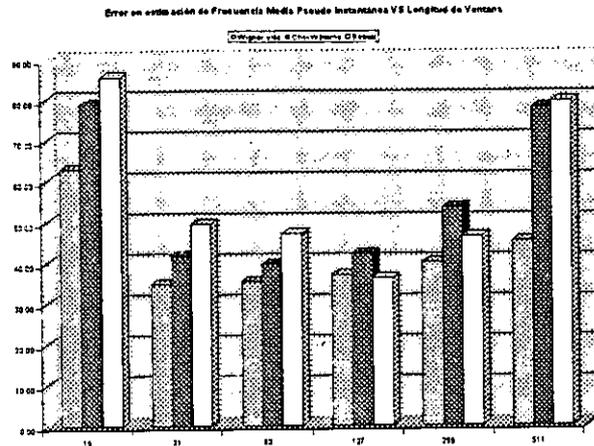


Figura FIG.8.8.a: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de ventana y sin presencia de ruido.

En el caso de los experimentos con una SNR de 40dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 31 elementos; la de Choi Williams, con una de 63; y la de Bessel, con una de 127. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Wigner Ville, excepto para una longitud de ventana de 127 elementos que fue hecha por la de Bessel. Ver figura FIG.8.8.b.

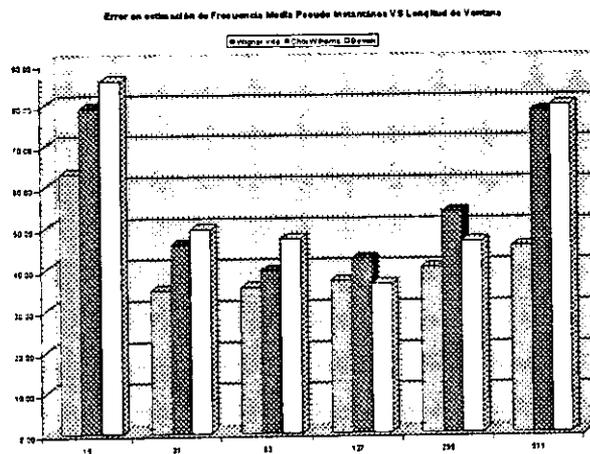


Figura FIG.8.8.b: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de ventana y SNR=40dB.

En el caso de los experimentos con una SNR de 20dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 31 elementos; la de Choi Williams, con una de 63; y la de Bessel, con una de 127. Por otro lado, la mejor estimación para una ventana de 15 elementos fue hecha por la distribución de Wigner Ville; para una de 31, 63 y 511, por la de Choi Williams; y para una de 127 y 255, por la de Bessel. Ver figura FIG.8.8.c.

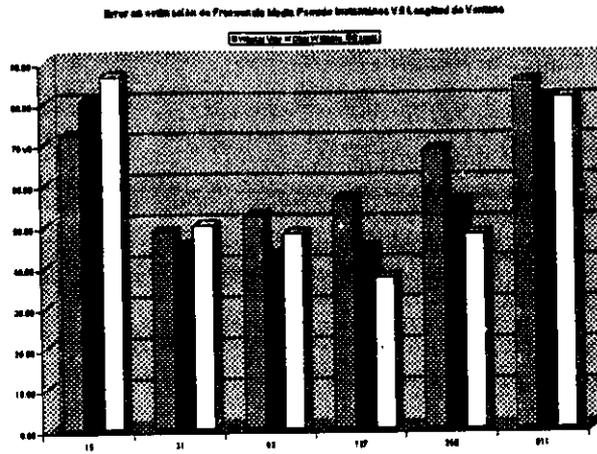


Figura FIG.8.8.c: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de ventana y SNR=20dB.

En el caso de los experimentos con una SNR de 10dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 15 elementos; la de Choi Williams, con una de 31; y la de Bessel, con una de 31. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Bessel. Ver figura FIG.8.8.d.

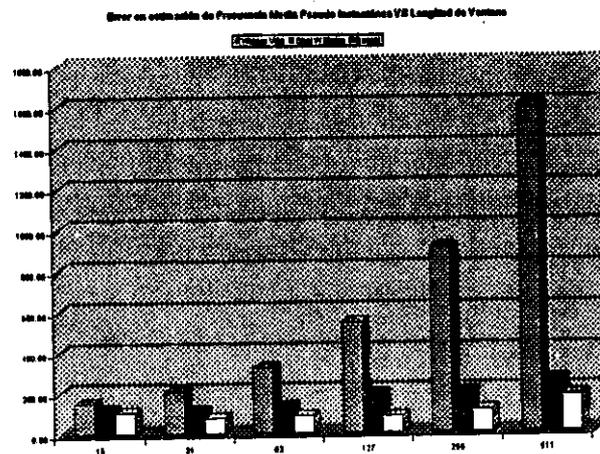


Figura FIG.8.8.d: Error en la estimación de la frecuencia media pseudo instantánea contra la longitud de ventana y SNR=10dB.

En ausencia de ruido y con una SNR de 40 dB, en general, la distribución de Wigner Ville estima con mayor precisión la frecuencia media pseudo instantánea de una señal Dópler ultrasónica. Con una SNR de 20 dB, en general, la distribución de Choi Williams la estima con mayor precisión. Con una SNR de 10 dB, en general, la distribución de Bessel la estima con mayor precisión. Obsérvese que para una ventana de longitud 127, la distribución de Bessel hace la mejor estimación de la frecuencia media pseudo instantánea, para cualquier nivel de SNR. En la tabla TB.8.3 se muestra con detalle los resultados anteriores.

SNR	Longitud de ventana de muestreo					
	15	31	63	127	255	511
Infinito	Wigner Ville	Wigner Ville	Wigner Ville	Béssel	Wigner Ville	Wigner Ville
40 dB	Wigner Ville	Wigner Ville	Wigner Ville	Béssel	Wigner Ville	Wigner Ville
20 dB	Wigner Ville	Choi Williams	Choi Williams	Béssel	Béssel	Choi Williams
10 dB	Béssel	Béssel	Béssel	Béssel	Béssel	Béssel

Tabla TB.8.3: Distribución tiempo frecuencia que mejor estima la frecuencia media pseudo instantánea en función de la longitud de ventana de muestreo y SNR.

8.6.2 Análisis de la estimación del ancho de banda espectral RMS

Nótese que, en general, las funciones de error en la del ancho de banda espectral RMS tienen un punto mínimo.

En el caso de los experimentos sin presencia de ruido se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 511 elementos; la de Choi Williams, con una de 511; y la de Béssel, con una de 255. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Wigner Ville. Ver figura FIG.8.9.a.

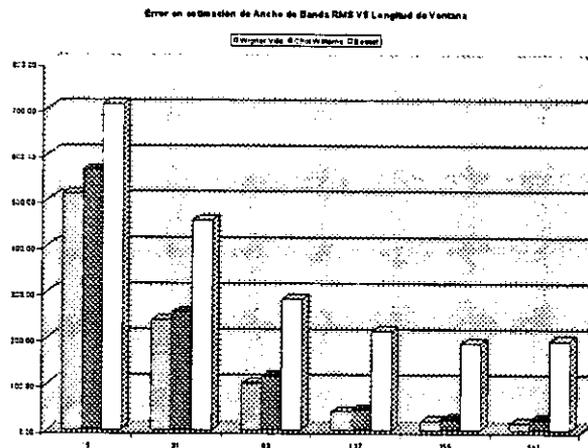


Figura FIG.8.9.a: Error en la estimación del ancho de banda RMS contra la longitud de ventana de muestreo y sin presencia de ruido.

En el caso de los experimentos con una SNR de 40 dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 511 elementos; la de Choi Williams, con una de 511; y la de Béssel, con una de 255. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Wigner Ville. Ver figura FIG.8.9.b.

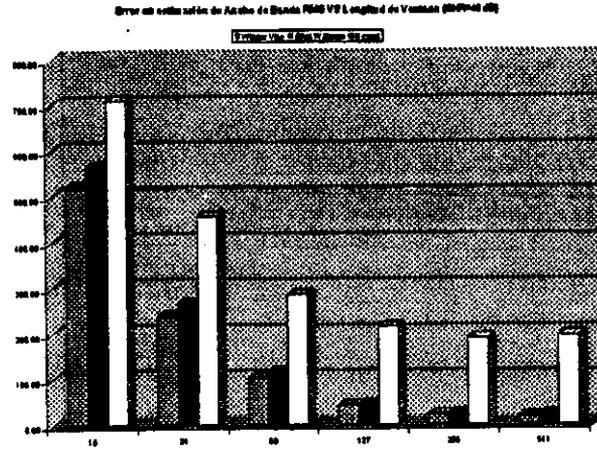


Figura FIG.8.9.b: Error en la estimación del ancho de banda RMS contra la longitud de ventana y SNR=40dB.

En el caso de los experimentos con una SNR de 20 dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 63 elementos; la de Choi Williams, con una de 255; y la de Bessel, con una de 255. Por otro lado, en general la mejor estimación para cualquier longitud de ventana fue hecha por la distribución de Choi Williams, excepto para una longitud de ventana de 15 elementos que fue hecha por la de Wigner Ville. Ver figura FIG.8.9.c.

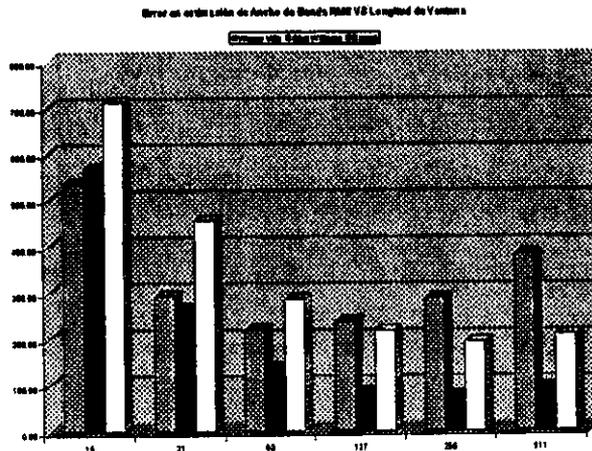


Figura FIG.8.9.c: Error en la estimación del ancho de banda RMS contra la longitud de ventana y SNR=20dB.

En el caso de los experimentos con una SNR de 10 dB se tiene que la distribución de Wigner Ville tuvo su mínimo error con una ventana de muestreo de 31 elementos; la de Choi Williams, con una de 63; y la de Bessel, con una de 63. Por otro lado, la mejor estimación para una ventana de 15, 31 y 63 elementos fue hecha por la distribución de Choi Williams; y para una de 127, 255 y 511, por la de Bessel. Ver figura FIG.8.9.d.

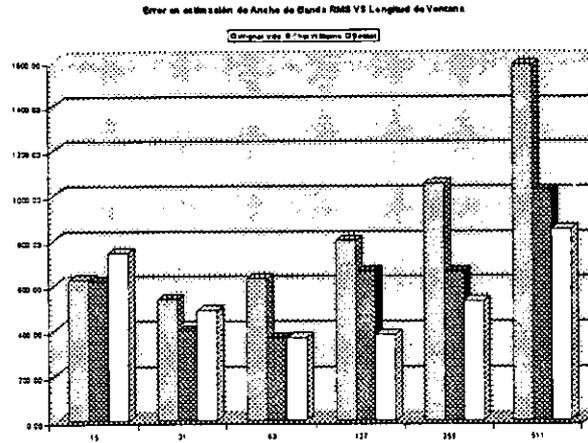


Figura FIG.8.9.d: Error en la estimación del ancho de banda RMS contra la longitud de ventana y SNR=10dB.

En ausencia de ruido y con una SNR de 40 dB, en general, la distribución de Wigner Ville estima con mayor precisión el ancho de banda RMS de una señal Dópler ultrasónica. Con una SNR de 20 dB, en general, la distribución de Choi Williams lo estima con mayor precisión. Con una SNR de 10 dB, en general, la distribución de Bessel lo estima con mayor precisión. En la tabla TB.8.4 se muestra con detalle los resultados anteriores.

SNR	Longitud de ventana de muestreo					
	15	31	63	127	255	511
Infinito	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville
40 dB	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville	Wigner Ville
20 dB	Wigner Ville	Choi Williams				
10 dB	Choi Williams	Choi Williams	Choi Williams	Bessel	Bessel	Bessel

Tabla TB.8.4: Distribución tiempo frecuencia que mejor estima en ancho de banda espectral RMS en función de la longitud de ventana de muestreo y SNR.

8.6.3 Resultados de la distribución de Wigner Ville

Respecto a la estimación de la frecuencia media pseudo instantánea, la distribución de Wigner Ville obtiene su mejor aproximación con ventanas de longitud 31, para cualquier nivel de SNR; excepto para 10 dB, con una ventana de longitud 15. Respecto a la estimación del ancho de banda espectral RMS, mientras mayor sea la presencia de ruido menor deberá ser la longitud de ventana, para obtener una mejor aproximación. En la tabla TB.8.5 se muestran con detalle los resultados obtenidos.

SNR	10	20	40	infinito
Frecuencia media pseudo instantánea	15	31	31	31
Ancho de banda espectral RMS	31	63	511	511

Tabla TB.8.5: Longitud de ventana en la cual se minimiza el error en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS para la distribución de Wigner Ville.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtiene su mejor compromiso para ventanas de longitud 127 y 255.

8.6.4 Resultados de la distribución de Choi Williams

Respecto a la estimación de la frecuencia media pseudo instantánea, en general la distribución de Choi Williams obtiene su mejor aproximación con ventanas de longitud 63, para cualquier nivel de SNR; excepto para 10dB, con una ventana de longitud 31. Respecto a la estimación del ancho de banda espectral RMS, mientras mayor sea la presencia de ruido menor deberá ser la longitud de ventana para obtener una mejor aproximación. En la tabla TB.8.6 se muestran con detalle los resultados obtenidos.

SNR	10	20	40	infinito
Frecuencia media pseudo instantánea	31	63	63	63
Ancho de banda espectral RMS	255	255	511	511

Tabla TB.8.6: Longitud de ventana en la cual se minimiza el error en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS para la distribución de Choi Williams.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtiene su mejor compromiso para ventanas de longitud 127 y 255.

8.6.5 Resultados de la distribución de Bessel

Respecto a la estimación de la frecuencia media pseudo instantánea, en general la distribución de Bessel obtiene su mejor aproximación con ventanas de longitud 127, para cualquier nivel de SNR; excepto para 10dB, con una ventana de longitud 31. Respecto a la estimación del ancho de banda espectral RMS, en general la distribución de Bessel obtiene su mejor aproximación con ventanas de longitud 255, para cualquier nivel de SNR; excepto para 10dB, con una ventana de longitud 63. En la tabla TB.8.7 se muestran con detalle los resultados obtenidos.

SNR	10	20	40	Infinito
Frecuencia media pseudo instantánea	31	127	127	127
Ancho de banda espectral RMS	63	255	255	255

Tabla TB.8.7: Longitud de ventana en la cual se minimiza el error en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS para la distribución de Bessel.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtiene su mejor compromiso para ventanas de longitud 127 y 255.

8.7 Algoritmos paralelos para estimación espectral

Se analizarán los resultados obtenidos en la sección 7.10, Desempeños de los algoritmos paralelos, del capítulo 7, Implantación de algoritmos paralelos para estimación espectral. En las secciones 7.10.1, 7.10.2 y 7.10.3 se muestran tablas que concentran los resultados de las figuras de desempeño (tiempo de ejecución, aceleración, eficiencia y fracción serial), para las distribuciones de Wigner Ville, de Choi Williams y de Bessel, respectivamente. En la sección 7.10.4 se muestran gráficas que comparan los tiempos de ejecución.

Existen dos factores preponderantes en el desempeño de los sistemas de cómputo paralelo: el tiempo de procesamiento y el tiempo de comunicación. Los algoritmos paralelos para estimación espectral implantados en este trabajo, en general, se forman de dos tipos de algoritmos: el algoritmo que calcula la FFT y el algoritmo que calcula la autocorrelación. El algoritmo que calcula la autocorrelación tiene una complejidad algorítmica de mayor orden que el algoritmo que calcula la FFT, es decir, para un mismo número de datos el primer algoritmo realiza más operaciones que el segundo; consecuentemente, el primero requiere de más tiempo de procesamiento. Por esto, los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al algoritmo que calcula la autocorrelación.

El tiempo de comunicación entre los diferentes procesos es un factor relevante en los algoritmos paralelos de estimación espectral, ya que los datos que procesan son vectores de cierta longitud. Para vectores de longitud pequeña, el tiempo de comunicación pesa más que el tiempo de ejecución; esto se debe a que la velocidad de los canales de transmisión es mucho menor que la velocidad de los procesadores. Sin embargo, para vectores de longitud grande, el tiempo de procesamiento pesa más que el tiempo de transmisión; esto se debe a que la lentitud en la transmisión se enmascara por la gran cantidad de operaciones (aunque rápidas) que se deben realizar. Por esto, los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al procesamiento de vectores de longitud grande.

Ahora bien, primero se analizarán los resultados para una arquitectura de cómputo paralelo con 1 procesador, luego para una arquitectura con 2 procesadores y, finalmente, para dos clases diferentes de arquitecturas con 4 procesadores.

8.7.1 Arquitectura para un procesador

En la tabla TB.8.8 se muestran con detalle los resultados obtenidos para una arquitectura de cómputo paralelo con 1 procesador.

Longitud de Ventana	Tiempo de ejecución [ms], 1 procesador		
	Wigner Ville	Choi Williams	Bessel
15	8.46	9.59	9.28
31	18.73	23.36	22.39
63	41.16	60.79	56.24
127	89.82	167.32	152.73

Tabla TB.8.8 Tiempo de ejecución de las diferentes distribuciones tiempo frecuencia en una arquitectura de cómputo paralelo con 1 procesador contra longitud de ventana.

Obsérvese que la distribución de Wigner Ville es la que requiere del menor tiempo para su cálculo, seguida por la de Bessel y, finalmente, por la de Choi Williams.

8.7.2 Arquitectura para dos procesadores

Respecto al algoritmo paralelo que calcula la distribución de Wigner Ville, para el caso de la arquitectura de 2 procesadores, tanto la aceleración como la eficiencia aumentan al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial disminuye: situación deseable.

Respecto al algoritmo paralelo que calcula la distribución de Choi Williams, para el caso de la arquitectura de 2 procesadores, tanto la aceleración como la eficiencia disminuyen al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial aumenta: situación no deseable.

Respecto al algoritmo paralelo que calcula la distribución de Bessel, para el caso de la arquitectura de 2 procesadores, tanto la aceleración como la eficiencia disminuyen al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial aumenta: situación no deseable. En la tabla TB.8.9 se muestran con detalle los resultados obtenidos.

Longitud de Ventana	Tiempo de ejecución [ms], 2 procesadores		
	Wigner Ville	Choi Williams	Bessel
15	5.52	5.69	5.55
31	11.65	13.79	12.82
63	25.08	38.73	34.17
127	54.22	118.08	103.49
255	117.54	420.86	337.36

Tabla TB.8.9 Tiempo de ejecución de las diferentes distribuciones tiempo frecuencia en una arquitectura de cómputo paralelo con 2 procesadores contra longitud de ventana.

Con relación a la arquitectura de cómputo paralelo de dos procesadores, el algoritmo que calcula la autocorrelación se efectúa en un solo procesador. Para el caso de la distribución de Wigner Ville, sí existe un balanceo de carga adecuado entre los dos procesadores; mientras que para el caso de las distribuciones de Choi Williams y de Bessel, no existe un balanceo de cargas adecuado entre los dos procesadores. Esto se debe a que la complejidad del algoritmo que calcula las distribuciones de Choi Williams y de Bessel es mayor que la complejidad del algoritmo que calcula la distribución de Wigner. Se confirma que los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al algoritmo que calcula la autocorrelación.

8.7.3 Arquitectura para cuatro procesadores

Respecto al algoritmo paralelo que calcula la distribución de Wigner Ville, para el caso de la arquitectura de 4 procesadores de tipo I y II (ver figuras FIG.8.10.a y FIG.8.10.b), tanto la aceleración como la eficiencia aumentan al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial disminuye: situación deseable. Sin embargo, la

arquitectura de tipo I presenta una mejor aceleración, eficiencia y fracción serial para todas las longitudes de ventana de muestreo que la arquitectura de tipo II.



Figura FIG.8.10.a: Arquitectura tipo I, paralelismo de datos (sección 7.9.2)

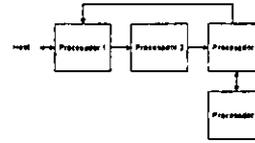


Figura FIG.8.10.b: Arquitectura tipo II, paralelismo de datos y algorítmico (sección 7.9.3)

Respecto al algoritmo paralelo que calcula la distribución de Choi Williams, para el caso de la arquitectura de 4 procesadores de tipo I, presenta un máximo tanto en la aceleración como en la eficiencia y un mínimo en la fracción serial para ventana de muestreo de longitud 31. En lo que respecta a la arquitectura de tipo II, tanto la aceleración como la eficiencia aumentan al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial disminuye: situación deseable. Comparando ambas arquitecturas, la arquitectura de tipo I presenta una mejor aceleración, eficiencia y fracción serial para las ventanas de muestreo de longitud 15, 31 y 63, mientras que la arquitectura de tipo II para las ventanas de muestreo de longitud 127 y 255.

Respecto al algoritmo paralelo que calcula la distribución de Bessel, para el caso de la arquitectura de 4 procesadores de tipo I, presenta un máximo tanto en la aceleración como en la eficiencia y un mínimo en la fracción serial para ventana de muestreo de longitud 61. En lo que respecta a la arquitectura de tipo II, tanto la aceleración como la eficiencia aumentan al incrementarse la longitud de la ventana de muestreo, mientras que la fracción serial disminuye: situación deseable. Comparando ambas arquitecturas, la arquitectura de tipo I presenta una mejor aceleración, eficiencia y fracción serial para las ventanas de muestreo de longitud 15, 31 y 63, mientras que la arquitectura de tipo II para las ventanas de muestreo de longitud 127 y 255. En la tabla TB.8.10 se muestran con detalle los resultados obtenidos.

Longitud de Ventana	Tiempo de ejecución [ms], 4 procesadores		
	Wigner Ville	Choi Williams	Bessel
15	4.09 (Tipo I)	4.13 (Tipo I)	4.11 (Tipo I)
31	8.69 (Tipo I)	8.79 (Tipo I)	8.85 (Tipo I)
63	18.66 (Tipo I)	23.67 (Tipo I)	19.13 (Tipo I)
127	40.20 (Tipo I)	65.93 (Tipo II)	55.26 (Tipo II)
255	86.51 (Tipo I)	255.61 (Tipo II)	186.84 (Tipo II)

Tabla TB.8.10 Tiempo de ejecución de las diferentes distribuciones tiempo frecuencia en una arquitectura de cómputo paralelo con 4 procesadores contra longitud de ventana.

Con relación a la arquitectura de cómputo paralelo de cuatro procesadores tipo I, el algoritmo que calcula la autocorrelación se efectúa en un solo procesador. Para el caso de la distribución de Wigner Ville, sí existe un balanceo de carga adecuado entre los cuatro procesadores; mientras que para los casos de las distribuciones de Choi Williams y de Bessel,

no existe un balanceo de cargas adecuado entre los cuatro procesadores. Nuevamente se confirma que los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al algoritmo que calcula la autocorrelación.

Con relación a la arquitectura de cómputo paralelo de cuatro procesadores tipo II, el algoritmo que calcula la autocorrelación se efectúa en dos procesadores. Para los casos de la distribución de Wigner Ville, de Choi Williams y de Bessel, sí existe un balanceo de carga adecuado entre los cuatro procesadores. Una vez más, se confirma que los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al algoritmo que calcula la autocorrelación.

Sin embargo, puesto que existe un compromiso en el comportamiento del tiempo de comunicación y del tiempo de procesamiento, la arquitectura de cómputo paralelo de cuatro procesadores tipo I es mejor para calcular la distribución de Wigner Ville para ventanas de muestreo de cualquier longitud y también para calcular las distribuciones de Choi Williams y de Bessel para ventanas de muestreo de longitudes pequeñas. Por su parte, la arquitectura de cómputo paralelo de cuatro procesadores tipo II es mejor para calcular las distribuciones de Choi Williams y de Bessel para ventanas de muestreo de longitudes grandes.

9 CONCLUSIONES

9.1 Introducción

En este capítulo se presentan las conclusiones correspondientes a cada distribución tiempo frecuencia y a la comparación entre ellas. Primero se describe la problemática; luego se plantea la hipótesis y, antes de las conclusiones, se describe el método de verificación. Finalmente, se sugieren trabajos de investigación futuros que continúen esta labor.

9.2 Problemática

En términos generales, el problema consiste en calcular el espectro (distribución de energía) de una señal no estacionaria Dópler ultrasónica.

Un método clásico de estimación espectral es la Transformada de Fourier. Sin embargo, su uso se limita a señales estacionarias, proporcionando una pobre resolución frecuencial cuando se aplica a señales no estacionarias.

Otra clase de estimadores espectrales que se han desarrollado es la clase de Cohen de distribuciones tiempo frecuencia. La aplicación de estas distribuciones no se limita a señales estacionarias, sino también se aplica a señales no estacionarias. A pesar de esta importante ventaja, el número de operaciones involucradas en su cálculo se incrementa substancialmente comparado con los métodos tradicionales. Entonces, es deseable simplificar la formulación de las distribuciones de tal modo que el número de operaciones involucrado en su cálculo se reduzca sin sacrificar la resolución espectral. También es deseable implementar los algoritmos que calculen estas distribuciones, los cuales aprovechen las capacidades de cómputo en paralelo de los sistemas digitales.

Por otro lado, puesto que existen diversas distribuciones tiempo frecuencia, sería útil desarrollar herramientas de análisis que proporcionen un criterio para seleccionar la distribución tiempo frecuencia adecuada, así como sus parámetros óptimos, de acuerdo a las características de la señal en estudio.

9.3 Hipótesis

Primero, la distribución tiempo frecuencia de Bessel estima mejor el espectro de una señal no estacionaria de tipo Dópler ultrasónica que la distribución de Choi Williams, tomando como referencia la estimación de su frecuencia media pseudo instantánea y ancho de banda espectral RMS. De igual forma, la distribución de Choi Williams hace una mejor estimación que la distribución de Wigner Ville. Se consideran factores relevantes que afectan la precisión de los cálculos la relación de potencias de la señal con respecto a la del ruido, así como la longitud de la ventana de muestreo.

Segundo, es posible calcular las distribuciones de Wigner Ville, de Choi Williams y de Bessel en una arquitectura de cómputo paralelo basada en 4 procesadores Transputers T805 de 30 MHz bajo las restricciones de la programación en tiempo real.

9.4 Método para validar la hipótesis

Las distribuciones tiempo frecuencia en estudio son las de Wigner Ville, Choi Williams y Bessel. Refiérase al capítulo 4, 5 y 6 respectivamente. Todas pertenecen a la clase de Cohen de distribuciones tiempo frecuencia. Refiérase al capítulo 3.

Primero, se generó una señal no estacionaria tipo Dópler ultrasónico típica con características conocidas, misma que se utilizó como dato controlado en los experimentos propuestos. Refiérase al apéndice A3, Señal tipo Dópler.

Luego, para cada una de las distribuciones en estudio, se diseñaron y realizaron experimentos, utilizando algoritmos secuenciales, para cuantificar el error en la estimación, tanto de la frecuencia media pseudo instantánea como del ancho de banda espectral RMS, de la señal en estudio, al variar la relación de potencias de la señal y del ruido (SNR), así como al variar la longitud de la ventana de muestreo. Ver las secciones 4.8, 5.8 y 6.8, Aplicación a una señal Dópler ultrasónica simulada en Matlab, así como las secciones 4.7, 5.7 y 6.7, Experimentos numéricos en Matlab.

Previo al experimento anterior y solamente para el caso de las distribuciones de Choi Williams y Bessel, se diseñaron y realizaron experimentos para determinar sus factores de escala óptimos (σ y α , respectivamente), los cuales minimizan el error en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, en ausencia de ruido. Ver las secciones 5.8.8 y 6.8.8, Variación del parámetro σ y Variación del parámetro α .

Brevemente se describe el procedimiento para obtener la frecuencia media instantánea y el ancho de banda espectral RMS. Primero, dada una señal real discreta de longitud finita, se calcula su señal analítica y se obtiene su distribución tiempo frecuencia. Luego, a partir de ésta última se calcula su distribución de potencia pseudo instantánea. Finalmente, con base en la anterior, se calcula la frecuencia media pseudo instantánea y el ancho de banda espectral RMS. Refiérase a las secciones 2.6, Señal analítica, 2.8, Frecuencia media pseudo instantánea, y A4.2, Ancho de banda espectral RMS. Además, refiérase a las secciones 4.4, 5.4 y 6.5, Caso discreto con ventana. En el anexo A4, Otras expresiones útiles para experimentos numéricos, se plantean las expresiones que cuantifican el error en las estimaciones mencionadas y la relación de potencias de la señal y del ruido (SNR).

Por otra parte, se plantearon expresiones simplificadas para el cálculo de las distribuciones tiempo frecuencia en estudio. Refiérase a las secciones 4.6, 5.6 y 6.6, Algoritmo eficiente. Utilizando las expresiones simplificadas, se diseñaron y realizaron experimentos, utilizando algoritmos paralelos, para cuantificar su desempeño, implantados en diferentes arquitecturas de cómputo paralelo, en las cuales se emplea paralelismo de datos y paralelismo algorítmico. Refiérase a la sección 7.8, Medidas de desempeño en procesamiento paralelo, a las secciones 7.3, 7.4, 7.5 y 7.6 para la arquitectura del algoritmo paralelo, y a la sección 7.9 para

las arquitecturas de cómputo paralelo. En la sección 7.10 se presentaron los desempeños de los algoritmos paralelos.

Nótese que para validar la hipótesis fue necesario alcanzar los objetivos específicos de diseñar, implantar y evaluar algoritmos secuenciales y paralelos para las distribuciones tiempo frecuencia de Wigner Ville, Choi Williams y Bessel. Además, se alcanzó el objetivo de evaluar de manera práctica los efectos de la variación de la longitud de ventana de muestreo y de la relación de potencias señal a ruido sobre la medición de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS.

9.5 Conclusiones

Una aportación de este trabajo fue el desarrollo y análisis detallado de las diferentes distribuciones tiempo frecuencia. En la literatura consultada sistemáticamente se obviaban pasos intermedios en los desarrollos y análisis que, en mi opinión, eran relevantes para su buen entendimiento. Una de mis intenciones al desarrollar este trabajo era elaborarlo a manera de tutorial.

Las conclusiones importantes se refieren a los parámetros óptimos de las distribuciones, así como a sus expresiones simplificadas que las calculan y a su implantación en arquitecturas de cómputo paralelas. También se refieren al criterio de comparación entre ellas. Éstas se enuncian a continuación.

9.5.1 De las distribuciones tiempo frecuencia

En el capítulo 3, Distribuciones tiempo frecuencia, se presentó una análisis general del planteamiento de las distribuciones tiempo frecuencia de la familia de Cohen. En particular, primero se propuso un método general para discretizar la distribución continua y, en consecuencia, obtener la distribución discreta (EQ.3.14). Luego, se propuso un método general para analizar la influencia del Kernel que define la distribución en la disminución de los efectos de interferencia entre los diferentes componentes de una señal (términos de cruce), expresado como un factor de peso (EQ.3.31). Finalmente, se propuso un método general para obtener una expresión con la cual se puede diseñar un algoritmo eficiente para el cálculo de la distribución discreta (EQ.3.39).

Además, en el capítulo 8, Análisis de resultados, en la sección 8.4, Evaluación de las distribuciones tiempo frecuencia con base en el factor de peso de los términos de cruce, se ofreció un criterio teórico de comparación entre las diferentes distribuciones tiempo frecuencia: se prefieren aquellas distribuciones tiempo frecuencia con un menor factor de peso de los términos de cruce.

En este sentido, un factor de peso ideal sería aquel que eliminara los términos de cruce. Sin embargo, situaciones deseables y posibles, son que el factor de peso concentre los términos de cruce debidos a dos componentes frecuenciales diferentes alrededor de tales frecuencias y no alrededor de otras frecuencias, o bien, que los disemine adecuadamente en un amplio rango de frecuencias. La primera situación provoca que una distribución tiempo

frecuencia estime con mayor precisión los parámetros espectrales de una señal con un ancho de banda pequeño en ausencia de ruido; mientras que la segunda situación provoca que una distribución tiempo frecuencia estime con mayor precisión los parámetros espectrales de una señal en presencia de ruido.

Con éste criterio se concluye que la distribución de Bessel con $\alpha = 2$ distribuye mejor los términos de cruce que la distribución de Choi Williams con $\sigma = 5$. Esto indica que la distribución de Bessel estima con mayor precisión el espectro de una señal en presencia de ruido que la distribución de Choi Williams.

También se concluye que la distribución de Wigner Ville es la que estima los parámetros espectrales con mayor precisión en ausencia de ruido; pero con la menor precisión con presencia de ruido.

9.5.2 De la distribución de Wigner Ville

En el capítulo 4, Distribución de Wigner Ville, se presentó una expresión con la cual se puede diseñar un algoritmo eficiente para el cálculo de la distribución de Wigner Ville discreta (EQ.4.32).

$$DVWD(0,k) = 4 \text{Real} \left[\sum_{\tau=0}^{N-1} x(\tau)x^*(-\tau)W(\tau)W^*(-\tau)e^{\frac{j2k\tau}{N}} \right] - 2|x(0)|^2 \quad (\text{EQ.9.1})$$

En esta expresión se realizan la mitad de las operaciones de las que se realizarían si se calculara esta misma distribución a través de su definición (EQ.4.13). Se efectúan, para el cálculo de cada k (componente frecuencial), $3N + 1$ multiplicaciones complejas y N sumas complejas. En consecuencia, su complejidad algorítmica, para el cálculo de cada k , es de orden N .

Respecto a la estimación de la frecuencia media pseudo instantánea, en general, la distribución de Wigner Ville obtuvo su mejor aproximación con ventanas de longitud 31 para cualquier nivel de SNR. Respecto a la estimación del ancho de banda espectral RMS, se obtuvo su mejor aproximación con ventanas de longitud 511 para cualquier nivel de SNR.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtuvo su mejor compromiso para ventanas de longitud 127.

9.5.3 Distribución de Choi Williams

En el capítulo 5, Distribución de Choi Williams, se presentó una expresión con la cual se puede diseñar un algoritmo eficiente para el cálculo de la distribución de Choi Williams (EQ.5.39).

$$DCWD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=-N+1+|\tau|}^{N-1-|\tau|} \sqrt{\frac{1}{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu + \tau) x^*(\mu - \tau) \right] - 2|x(0)|^2 \quad (\text{EQ.9.2})$$

En esta expresión se realizan la cuarta parte de las operaciones de las que se realizarían si se calculara esta misma distribución a través de su definición (EQ.5.13). Se efectúan, para el cálculo de cada k (componente frecuencial), $2N^2 - 2N + 1$ multiplicaciones complejas y $N^2 - 2N$ sumas complejas. En consecuencia, su complejidad algorítmica, para el cálculo de cada k , es de orden N^2 .

Respecto a la estimación de la frecuencia media pseudo instantánea, en general, la distribución de Choi Williams obtuvo su mejor aproximación con ventanas de longitud 63 para cualquier nivel de SNR. Respecto a la estimación del ancho de banda espectral RMS, se obtuvo su mejor aproximación con ventanas de longitud 255 para cualquier nivel de SNR.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtuvo su mejor compromiso para ventanas de longitud 255.

Ahora bien; respecto al parámetro σ de la distribución de Choi Williams, la mejor estimación conjunta de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, se obtuvo con $\sigma = 5$. Este parámetro óptimo se determinó de manera experimental y dependía de la señal que se analiza.

9.5.4 Distribución de Bessel

En el capítulo 6, Distribución de Bessel, se presentó una expresión con la cual se puede diseñar un algoritmo eficiente para el cálculo de la distribución de Bessel (EQ.6.39).

$$DBD(0, k) = 4 \operatorname{Re} \left[\sum_{\tau=0}^{N-1} W(\tau) W^*(-\tau) e^{-\frac{j2k\tau}{N}} \sum_{\mu=\max\{-2\alpha|\tau|, -N+1+|\tau|\}}^{\min\{2\alpha|\tau|, N-1-|\tau|\}} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(\mu + \tau) x^*(\mu - \tau) \right] - 2|x(0)|^2 \quad (\text{EQ.9.3})$$

En esta expresión se realizan a lo más la mitad de las operaciones de las que se realizarían si se calculara esta misma distribución a través de su definición (EQ.6.13). Se efectúan, para el cálculo de cada k , menos que $4\alpha N^2 - 4\alpha N + 1$ multiplicaciones complejas y menos que $2\alpha N^2 - 2\alpha N - N$ sumas complejas. En consecuencia, su complejidad algorítmica, para el cálculo de cada k (componente frecuencial), es de orden N^2 .

Respecto a la estimación de la frecuencia media pseudo instantánea, en general, la distribución de Bessel obtuvo su mejor aproximación con ventanas de longitud 127 para cualquier nivel de SNR. Respecto a la estimación del ancho de banda espectral RMS, se obtuvo su mejor aproximación con ventanas de longitud 255 para cualquier nivel de SNR.

Referente a una estimación conjunta del ancho de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, ésta obtuvo su mejor compromiso para ventanas de longitud 255. Refiérase a la sección 6.8.7, Errores en la estimación de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS.

Ahora bien; respecto al parámetro α de la distribución de Bessel, la mejor estimación conjunta de la frecuencia media pseudo instantánea y del ancho de banda espectral RMS, se obtuvo con $\alpha = 2$. Este parámetro óptimo se determinó de manera experimental y dependía de la señal que se analiza.

9.5.5 Comparación entre las distribuciones de Wigner Ville, de Choi Williams y de Bessel

Primero, el cálculo de la distribución de Wigner Ville requirió del menor número de operaciones; mientras que el cálculo de la distribución de Choi Williams requirió del mayor número de operaciones. En un lugar intermedio quedó la distribución de Bessel. Nótese que la cantidad de operaciones es directamente proporcional a los tiempos de ejecución de los algoritmos que calculan cada distribución.

Segundo, la tasa de crecimiento del error en la estimación de los parámetros espectrales fue menor en la distribución de Bessel; mientras que fue mayor en la distribución de Wigner Ville. En un lugar intermedio quedó la distribución de Choi Williams. Una tasa de crecimiento del error menor se asocia con una estimación más precisa en presencia de ruido.

Tercero, en ausencia de ruido y con una SNR de 40 dB, en general, la distribución de Wigner Ville estimó con mayor precisión tanto la frecuencia media pseudo instantánea como el ancho de banda espectral RMS de una señal Dópler ultrasónica. Con una SNR de 20 dB, en general, la distribución de Choi Williams estimó con mayor precisión tanto la frecuencia media pseudo instantánea como el ancho de banda espectral RMS de una señal Dópler ultrasónica. Con una SNR de 10 dB, en general, la distribución de Bessel estimó con mayor precisión tanto la frecuencia media pseudo instantánea como el ancho de banda espectral RMS de una señal Dópler ultrasónica. Obsérvese que para una ventana de longitud 127, la distribución de Bessel hizo la mejor estimación de la frecuencia media pseudo instantánea, para cualquier nivel de SNR.

Resumiendo, en ausencia de ruido, se prefiere la Distribución de Wigner Ville, ya que estima los parámetros espectrales con la mayor precisión requiriendo del menor número de operaciones. Sin embargo, en presencia de ruido, se prefiere la Distribución de Bessel, ya que estima los parámetros espectrales con la mayor precisión aunque requiriendo un mayor número de operaciones pero sin alcanzar la cantidad que requiere la Distribución de Choi Williams.

Específicamente, si solamente se requiere calcular la frecuencia media pseudo instantánea con la mayor precisión en presencia de ruido, se prefiere la distribución de Bessel utilizando una longitud de ventana de 127 elementos. Ahora bien, si solamente se requiere calcular el ancho de banda espectral RMS, se prefiere la distribución de Choi Williams utilizando una longitud de ventana de 255 elementos.

9.5.6 Algoritmos paralelos para estimación espectral

En el capítulo 7, Implantación de Algoritmos Paralelos para estimación espectral, se presentó el diseño y la implantación de algoritmos paralelos que calculan la distribución de Wigner Ville, la distribución de Choi Williams y la distribución de Bessel. En el anexo 1, programas secuenciales en Matlab, se presentó el código en lenguaje propietario de Matlab de los programas secuenciales que calculan dichas distribuciones. Además, en el anexo 2, programas paralelos en Occam, se presentó el código en lenguaje Occam de los programas paralelos que calculan esas distribuciones.

Primero, para una misma longitud de ventana de muestreo, el cálculo de la distribución de Wigner Ville requirió del menor tiempo; mientras que el cálculo de la distribución de Choi Williams requirió del mayor. En un lugar intermedio quedó la distribución de Bessel.

Segundo, en general, en una arquitectura de cómputo paralelo de un procesador, los tiempos de ejecución para el cálculo de cualquier distribución fueron similares para ventanas de longitud de 15 y 31. La diferencia se acentuó gradualmente para ventanas de longitud de 63, 127 y 255 elementos. Lo mismo ocurrió en una arquitectura de cómputo paralelo de dos procesadores. Sin embargo, para una arquitectura de cuatro procesadores, los tiempos de ejecución para el cálculo de cualquier distribución fueron similares para ventanas de longitud de 15, 31 y 63 elementos. La diferencia se acentuó gradualmente para ventanas de longitud de 127 y 255 elementos.

Ahora bien, en una arquitectura de cómputo paralelo de un procesador, los tiempos de ejecución para el cálculo de cualquier distribución fueron inferiores a 10 ms para ventanas de longitud 15, e inferiores a 20 ms para ventanas de longitud 31. Lo mismo ocurrió en una arquitectura de cómputo paralelo de dos procesadores. Sin embargo, para una arquitectura de cuatro procesadores, los tiempos de ejecución para el cálculo de cualquier distribución fueron inferiores a 10 ms para ventanas de longitud 15 y 31, e inferiores a 20 ms para ventanas de longitud 63. Si se utilizara una arquitectura de equipo de cómputo paralelo con una capacidad de hacer cálculos numéricos 25 veces mayor que la de los procesadores Transputers, se lograrían tiempos de ejecución para el cálculo de cualquier distribución inferiores a 10 ms para ventanas de longitud 255.

Tercero, se propuso una arquitectura general de tipo entubamiento (*pipeline*) del algoritmo paralelo para la estimación espectral de la distribución tiempo frecuencia, compuesta de tres etapas: en la primera se calcula la señal analítica; en la segunda, la función de Autocorrelación Generalizada Indexada al Tiempo; y en la tercera, la Distribución Tiempo Frecuencia. Refiérase a la sección 7.4, Arquitectura del algoritmo paralelo TFD.

Los algoritmos paralelos para estimación espectral implantados en este trabajo, en general, se forman de dos tipos de algoritmos: el algoritmo que calcula la FFT (primera y tercera etapas), y el algoritmo que calcula la autocorrelación (segunda etapa). El algoritmo que calcula la autocorrelación tiene una complejidad algorítmica de mayor orden que el algoritmo que calcula la FFT. Por esto, los esfuerzos para distribuir el procesamiento entre múltiples procesadores se orientaron al algoritmo que calcula la autocorrelación. Para el algoritmo paralelo que calcula la FFT se propuso una arquitectura de tipo arborescente (*tree*). Refiérase

a la sección 7.6, Arquitectura del algoritmo paralelo para FFT. Para el algoritmo paralelo que calcula la autocorrelación se propuso una arquitectura de tipo estrella (*star*). Refiérase a la sección 7.7, Arquitectura del algoritmo paralelo para autocorrelación.

Cuarto, además de considerar que los esfuerzos para distribuir el procesamiento entre múltiples procesadores deben estar orientados al algoritmo que calcula la autocorrelación, también se consideró que estos esfuerzos deben estar orientados al procesamiento de ventanas de muestreo de longitud grande.

Puesto que existe un compromiso entre el comportamiento del tiempo de comunicación y del tiempo de procesamiento, la arquitectura de cómputo que no utilizó un algoritmo paralelo en la autocorrelación calculó más rápido la distribución de Wigner Ville para ventanas de muestreo de cualquier longitud y también las distribuciones de Choi Williams y de Bessel para ventanas de muestreo de longitudes pequeñas. Por su parte, la arquitectura de cómputo que utilizó un algoritmo paralelo en la autocorrelación calculó más rápido las distribuciones de Choi Williams y de Bessel para ventanas de muestreo de longitudes grandes.

Se concluye que la arquitectura del equipo de cómputo paralelo es determinante en las figuras de desempeño (tiempo de ejecución, eficiencia, aceleración y fracción serial), para los algoritmos paralelos de estimación espectral, para rangos específicos de longitudes de vectores.

9.5.7 Señal tipo Dópler ultrasónica real.

En las secciones 4.9, 5.9 y 6.9, Aplicación a una señal Dópler ultrasónica real en Matlab, de los capítulos 4, 5 y 6, Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel, respectivamente, se presentó un experimento con una señal Dópler ultrasónica real medida en el laboratorio de DISCA. Para sendas distribuciones, se obtuvieron gráficas de la frecuencia media pseudo instantánea y de del ancho de banda espectral RMS, las cuales corresponden a formas esperadas. Desafortunadamente, solo se poseía una muestra de la señal Dópler ultrasónica, por lo que no fue posible hacer un análisis estadístico; solamente se realizó un análisis cualitativo.

9.5.8 Señal tipo Dópler ultrasónica simulada.

En el anexo 3 se propuso la construcción de una señal tipo Dópler ultrasónica modelo típica para verificar la validez de las expresiones de las distribuciones tiempo frecuencia.

9.6 Trabajos de investigación futuros

Se sugiere la realización de los siguientes trabajos de investigación, los cuales constituirían una continuación de este trabajo:

- Diseño e implantación de los algoritmos paralelos que calculen la distribución de Wigner Ville, de Choi Williams y de Bessel, en lenguaje de programación C en una arquitectura de cómputo paralelo basada en procesadores digitales de señales SHARC.

- Estudio teórico práctico de las distribuciones tiempo frecuencia de Margenau Hill, Kirkwood Rihaczek, Born Jordan, Page, Zhao Atlas Marks y Sinc^{[1][2]}. Véase sección 3.7, Tabla de distribuciones. El método de estudio puede ser el que se siguió en este trabajo e incluiría una comparación de propiedades y desempeños con los resultados ya obtenidos.
- Estudio teórico práctico de las distribuciones tiempo frecuencia tipo RID^[4] (Reduced Interference Distributions) y tipo AFGDTTFD^[5] (Alias-Free Generalized Discrete-Time Time-Frequency Distributions). El método de estudio puede ser el que se siguió en este trabajo e incluiría una comparación de propiedades y desempeños con los resultados ya obtenidos.
- Investigación sobre métodos de estimación espectral basados en Wavelets^[50]. Una señal se puede descomponer para resaltar dos aspectos relevantes: la localización en el tiempo de fenómenos transitorios y la presencia en el tiempo de frecuencias específicas. La técnica de descomposición puede consistir en una expansión con una base ortonormal waveletes; es decir, descomponer la señal en átomos tiempo-frecuencia. En particular se sugiere el análisis tipo *wavelet packet*.
- Investigación sobre métodos de análisis de señales basados en Fractales. ¿Servirá este tipo de análisis para efectuar estimación espectral en una señal?^[51].

A1 PROGRAMAS SECUENCIALES EN MATLAB

A1.1 Introducción

En este anexo se listan los códigos en lenguaje propietario de Matlab de los programas que calculan las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, así como la frecuencia media pseudo instantánea y el ancho de banda espectral RMS basado en ellas.

A1.2 Distribución de Wigner Ville

A continuación se lista el código en lenguaje propietario de Matlab del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Wigner Ville.

```
%% .....  
%%  
%% IIMAS, UNAM  
%% Ernesto Rubio Acosta  
%% Distribucion Tiempo Frecuencia Wigner Ville  
%% ** Se analiza un vector de 13440 elementos  
%% ** correspondiente a una señal Doppler ultrasónica  
%% ** con una ventana de longitud N,  $[-(N-1)/2:(N-1)/2]$  que se traslada  
%% ** para N = 31, 63, 127, 255, 511,  
%% ** valuada en 0  
%% ** implementada con FFT  
%% 19 de mayo de 1999  
%% .....  
%%  
%% Señal real en todo su dominio  
%% Señal Doppler Ultrasónica  
%%  
load dopplers; % Asigna el vector s1  
titulo = 'Señal Doppler Ultrasónica';  
%%  
%% Longitud de ventana  
%%  
opcion = input('Opcion:Longitud de ventana {1:15, 2:31, 3:63, 4:127, 5:255, 6:511} > ');  
if opcion == 1  
    n = 15;  
elseif opcion == 2  
    n = 31;  
elseif opcion == 3  
    n = 63;  
elseif opcion == 4  
    n = 127;  
elseif opcion == 5  
    n = 255;  
elseif opcion == 6  
    n = 511;  
else  
    n = 127;  
end  
%%  
%% Ruido  
%%  
opcion = input('¿Se adiciona Ruido Gaussiano? No=0, Si de otro modo >');  
if opcion == 0  
    % Espectro de la Señal de Ruido  
    ruidoR = randn(1,13440);  
    ruidoI = randn(1,13440);  
    ruidoC = ruidoR + 1j*ruidoI;  
    modRuidoC = abs( ruidoC );  
    ruidoN = ruidoC ./ modRuidoC;  
    % Señal de Ruido  
    ruido = real( ifft( ruidoN ) );  
    % Potencia de la Señal Real  
    potenciasenal = sum(s1.^2)/13440; % 0.5;  
    % Potencia de la Señal de Ruido  
    potenciaaruido = sum(ruido.^2)/13440; % 0.5;  
    % Relacion Señal a Ruido  
    SNR = input( 'Relacion Señal a Ruido en decibelios > ' );  
    potenciaobjetivoruido = potenciasenal/10^(SNR/10);  
    factor = potenciaobjetivoruido / potenciaaruido;  
    s1 = s1 + factor*ruido;  
end
```

```

%%
%% Inicializacion de arreglos
%%
bmv = zeros(13440-(n-1),1); % Ancho de Banda Espectral RMS
fww = zeros(13440-(n-1),1); % Frecuencia Media Pseudo-Instantánea
pwv = zeros(13440-(n-1),(n+1)/4); % Distribución de Potencia Pseudo-Instantánea
%%
%% Ventana de Hanning
%% Semánticamente los índices del vector d van de -(n-1)/2 a (n-1)/2
%% En Matlab éstos índices van de 1 a n respectivamente
%% Semánticamente los índices del vector ventana van de 0 a (n-1)/2
%% En Matlab éstos índices van de 1 a (n+1)/2 respectivamente
%% Algoritmo:
%% 1: Se genera una ventana de Hanning de longitud n
%% 2: Se calcula ventana(k) = d(k)d*(-k), k=0:(n-1)/2
%%
d = hanning(n)';
ventana = d((n+1)/2:n) .* conj( d((n+1)/2:-1:1) );
%%
%% Para cada ventana t
%% En un vector de 13440 elementos existen 13440-(n-1) ventanas de longitud n
%%
tic; %% Cronómetro
for t=1:13440-(n-1)
    if mod(t,100) == 0
        fprintf('.');
        if mod(t,1000) == 0
            fprintf('\n');
        end
    end
    %% Señal real
    %% Semánticamente los índices del vector xr van de -(n-1)/2 a (n-1)/2
    %% En Matlab éstos índices van de 1 a n respectivamente
    %% Algoritmo:
    %% 1: Se toma un segmento de n elementos de la señal real
    %%
    xr = s1(t:t+(n-1));
    %%
    %% Transformada de Fourier de la señal real
    %% Semánticamente los índices del vector fftxr van de 0 a (n-1)/2 y de -(n-1)/2 a -1
    %% En Matlab la frecuencia 0 está en el índice 1
    %% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
    %% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
    %% Algoritmo:
    %% 1: Se calcula la transformada de Fourier de xr
    %%
    fftxr = fft(xr);
    %%
    %% Transformada de Fourier de la señal analítica
    %% Semánticamente los índices del vector fftx van de 0 a (n-1)/2 y de -(n-1)/2 a -1
    %% En Matlab la frecuencia 0 está en el índice 1 del vector
    %% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
    %% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
    %% Algoritmo
    %% 1: Frecuencias negativas se igualan a cero
    %% 2: Frecuencia cero no se modifica
    %% 3: Frecuencias positivas se duplican
    %%
    fftx = zeros(1,n);
    fftx(1) = fftxr(1);
    fftx(2:(n+1)/2) = 2 * fftxr(2:(n+1)/2);
    %%
    %% Señal analítica
    %% Semánticamente los índices del vector x van de -(n-1)/2 a (n-1)/2
    %% En Matlab éstos índices van de 1 a n respectivamente
    %% Algoritmo
    %% 1: Calcular la transformada inversa de Fourier de fftx
    %%
    x = ifft( fftx );
    %%
    %% Función de Distribución Pseudo-Wigner
    %% Semánticamente los índices del vector kernel van de 0 a (n-1)/2
    %% En Matlab éstos índices van de 1 a (n+1)/2 respectivamente
    %% Semánticamente los índices del vector wvd van de -(n+1)/4 a ((n+1)/4-1)
    %% En Matlab éstos índices van de ((n+1)/4+1) a (n+1)/2 y de 1 a (n+1)/4 respectivamente
    %% Algoritmo
    %% 1: Se calcula kernel(T) = ventana(T) x(T) x*(-T), T=0:(n-1)/2
    %% 2: Se calcula wvd(k) = 4*real(fft(kernel))-2*(0)^2, k=-((n+1)/4):((n+1)/4-1)
    %%
    kernel = ventana .* x((n+1)/2:n) .* conj( x((n+1)/2:-1:1) );
    wvd = 4 * real ( fft( kernel ) ) - 2 * x(0+(n+1)/2) * conj( x(0+(n+1)/2) );
    %%
    %% Distribución de Potencia Pseudo-Instantánea
    %% Semánticamente los índices del vector pwv van de 0 a ((n+1)/4-1)
    %% En Matlab éstos índices van de 1 a (n+1)/4 respectivamente
    %% Semánticamente los índices del vector wvd van de -(n+1)/4 a ((n+1)/4-1)
    %% En Matlab éstos índices van de ((n+1)/4+1) a (n+1)/2 y de 1 a (n+1)/4 respectivamente
    %% Algoritmo
    %% 1: Potencias negativas se eliminan
    %% 2: Potencias cero y positivas no se modifican
    %%
    for k = 0:((n+1)/4-1)
        if wvd(k+1) > 0
            pwv(t,k+1) = wvd(k+1);
        end
    end
end

```

```

else
    pwv(t,k+1) = 0;
end
end
%%
%% Frecuencia Media Pseudo-Instantánea
%% Semánticamente los índices del vector pwv van de 0 a ((n+1)/4-1)
%% En Matlab estos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula fvw = sumatoria( k pwv(k) ) / sumatoria( pwv(k) ),
%%                               k = 0, ((n+1)/4-1)
%%
sumpwv = sum( pwv(t,: ) );
if sumpwv == 0
    fvw(t) = 0;
else
    fvw(t) = sum( (0:((n+1)/4-1)) .* pwv(t,: ) /sumpwv;
end
%%
%% Ancho de Banda Espectral RMS
%% Semánticamente los índices del vector pwv van de 0 a ((n+1)/4-1)
%% En Matlab estos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula bwv = ( sumatoria( ( fvw - k )^2 pwv(k) ) /
%%                       sumatoria( pwv(k) ) )^0.5, k = 0, ((n+1)/4-1)
%%
if sumpwv == 0
    bwv(t) = 0;
else
    bwv(t) = ( sum( (fvw(t)-(0:((n+1)/4-1)).^2 .* pwv(t,: ) /sumpwv )^0.5;
end
end
tiempo = toc; %% Cronómetro
fprintf('\n')
fprintf('Tiempo de ejecución total: %f\n', tiempo)
fprintf('Tiempo de ejecución por transformada: %f\n', tiempo/(13440-(n-1)))
%%
%% Estimación de Error RMS en el Ancho de Banda Espectral RMS
%%
biasbwv = mean( bwv - 100/(19200/n) );
desviacionbwv = std( bwv - 100/(19200/n) )^2;
errorbwv = ( biasbwv^2 + desviacionbwv^2 )^0.5;
fprintf('Error RMS en el Ancho de Banda Espectral RMS = %12.8f\n', (19200/n)*errorbwv )
%%
%% Estimación de Error en la Frecuencia Media Pseudo-Instantánea
%%
load f; % Frecuencia Instantánea
biasf = mean( fvw - f((n+1)/2:(13440-(n-1)/2)')/(19200/n) );
desviacionf = std( fvw - f((n+1)/2:(13440-(n-1)/2)')/(19200/n) )^2;
errorf = ( biasf^2 + desviacionf^2 )^0.5;
fprintf('Error en la Frecuencia Media Pseudo-Instantánea = %12.8f\n', (19200/n)*errorf )
%%
%% *****
%% Graficación
%% *****
%%
%% Función real
%%
figure(1)
subplot(3,1,1)
plot( (0:0.7*(13440-1):0.7*(13440-1)), 0:19200/n:19200/n, 'r')
xlabel('Amplitud')
title('título');
%%
%% Frecuencia Media Pseudo-Instantánea
%%
subplot(3,1,2)
plot( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)), (19200/n)*f )
ylabel('Frecuencia (Hz)')
title('Frecuencia Media Pseudo-Instantánea')
axis( [ 0 0.7 0 (19200/n)*max(f)*1.1 ] )
%%
%% Ancho de Banda Espectral RMS
%%
subplot(3,1,3)
plot( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)), (19200/n)*bwv )
xlabel('Tiempo (s)')
ylabel('Frecuencia (Hz)')
title('Ancho de Banda Espectral RMS')
axis( [ 0 0.7 0 (19200/n)*max(bwv)*1.1 ] )
%%
%% Distribución de Potencia Pseudo-Instantánea
%%
figure2 = 0;
if figure2 == 0
    figure(2)
    mesh( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)), 0:19200/n:19200/n, 'r', fvw )
    ylabel('Frecuencia (Hz)')
    xlabel('Tiempo (s)')
    zlabel('Potencia')
    title('Distribución de Potencia Pseudo-Instantánea')
    axis( [ 0 0.7 0 (19200/n)*max(fvw)*1.1 ] )
    view(1,0)
end

```

A1.3 Distribución de Choi Williams

A continuación se lista el código en lenguaje propietario de Matlab del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Choi Williams.

```

%% .....
%%
%% IIMAS, UNAM
%% Ernesto Rubio Acosta
%% Distribución Tiempo Frecuencia Choi Williams
%% ** Se analiza un vector de 13440 elementos
%% ** correspondiente a una señal Doppler ultrasónica
%% ** con una ventana de logitud 127, (-63:63) que se traslada
%% ** valuada en 0
%% ** índices optimizados
%% ** implementada con FFT
%% 14 de mayo de 1999
%%
%% .....
%%
%% Señal real en todo su dominio
%% Señal Doppler Ultrasónica
%%
load dopplers; % Asigna el vector s1
titulo = 'Señal Doppler Ultrasónica';
%%
%% Longitud de ventana
%%
opcion = input('Opción:Longitud de ventana (1:15, 2:31, 3:63, 4:127, 5:255, 6:511) > ');
if opcion == 1
    n = 15;
elseif opcion == 2
    n = 31;
elseif opcion == 3
    n = 63;
elseif opcion == 4
    n = 127;
elseif opcion == 5
    n = 255;
elseif opcion == 6
    n = 511;
else
    n = 127;
end
%%
%% Ruido
%%
opcion = input('¿Se adiciona Ruido Gaussiano? No=0, Si de otro modo >');
if opcion == 0
    % Espectro de la Señal de Ruido
    ruidoR = randn(1,13440);
    ruidoI = randn(1,13440);
    ruidoC = ruidoR + 1j*ruidoI;
    modRuidoC = abs( ruidoC );
    ruidoN = ruidoC ./ modRuidoC;
    % Señal de Ruido
    ruido = real( ifft( ruidoN ) );
    % Potencia de la Señal Real
    potenciasenal = (sum(s1.^2)/13440)^0.5;
    % Potencia de la Señal de Ruido
    potenciaruido = (sum(ruido.^2)/13440)^0.5;
    % Relación Señal a Ruido
    SNR = input( 'Relación Señal a Ruido en decibeles >' );
    potenciaobjetivoruido = potenciasenal/(10^(SNR/10));
    factor = potenciaobjetivoruido / potenciaruido ;
    s1 = s1 + factor*ruido;
end
%%
%%
%% Inicializacion de arreglos
%%
bcw = zeros(13440-(n-1),1); % Ancho de Banda Espectral RMS
fcw = zeros(13440-(n-1),1); % Frecuencia Media Pseudo-Instantánea
pcw = zeros(13440-(n-1), (n+1)/4); % Distribución de Potencia Pseudo-Instantánea
%%
%% Parámetro Sigma
%%
sigma = 5;
%%
%% Ventana de Choi Williams ( Valores Reales )
%% Semánticamente los índices de la matriz vcw van de 0 a (n-1)/2 y de -(n-1)/2 a (n-1)/2
%% En Matlab éstos índices van de 1 a (n+1)/2 y de 1 a n respectivamente
%% Algoritmo:
%% 1: Se calcula vcw(T,M) = ... , T=1:(n-1)/2, M=-(n-1)/2:(n-1)/2, i<>0
%% 2: vcw(0,M) = 0, M=-(n-1)/2:(n-1)/2, M<>0
%% 3: vcw(0,0) = 1

```

```

%%
vcw = zeros((n+1)/2,n);
T = 0;
vcw(T+1,(n+1)/2) = 1;
for T = 1 : (n-1)/2;
    M = -(n-1)/2 : (n-1)/2;
    vcw(T+1,M+(n+1)/2) = (1/((4*pi*T^2)/sigma).^0.5)*exp(-M.^2/((4*T^2)/sigma));
end
%%
%% Ventana de Hanning
%% Semánticamente los índices del vector d van de -(n-1)/2 a (n-1)/2
%% En Matlab estos índices van de 1 a n respectivamente
%% Semánticamente los índices del vector ventana van de 0 a (n-1)/2
%% En Matlab estos índices van de 1 a (n+1)/2 respectivamente
%% Algoritmo:
%% 1: Se genera una ventana de Hanning de longitud n
%% 2: Se calcula ventana(T) = d(T)d*(-T), T=0:(n-1)/2
%%
d = hanning(n)';
ventana = d((n+1)/2:n) .* conj( d((n+1)/2:-1:1) );
%%
%% Para cada ventana t
%% En un vector de 13440 elementos existen 13440-(n-1) ventanas de longitud n
%%
tic; %% Cronometro
for t = 1 : 13440-(n-1)
    if mod(t,100) == 0
        fprintf(' ');
        if mod(t,1000) == 0
            fprintf('\n');
        end
    end
    %%
    %% Señal real
    %% Semánticamente los índices del vector xr van de -(n-1)/2 a (n-1)/2
    %% En Matlab estos índices van de 1 a n respectivamente
    %% Algoritmo:
    %% 1: Se toma un segmento de n elementos de la señal real
    %%
    xr = s((t):(t+n-1));
    %%
    %% Transformada de Fourier de la señal real
    %% Semánticamente los índices del vector fftxr van de 0 a (n-1)/2 y de -(n-1)/2 a -1
    %% En Matlab la frecuencia 0 está en el índice 1
    %% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
    %% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
    %% Algoritmo:
    %% 1: Se calcula la transformada de Fourier de xr
    %%
    fftxr = fft(xr);
    %%
    %% Transformada de Fourier de la señal analítica
    %% Semánticamente los índices del vector fftx van de 0 a (n-1)/2 y de -(n-1)/2 a -1
    %% En Matlab la frecuencia 0 está en el índice 1 del vector
    %% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
    %% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
    %% Algoritmo
    %% 1: Frecuencias negativas se igualan a cero
    %% 2: Frecuencia cero no se modifica
    %% 3: Frecuencias positivas se duplican
    %%
    fftx = zeros(1,n);
    fftx(1) = fftxr(1);
    fftx(2:n-1) = 2 * fftxr(2:(n+1)/2);
    %%
    %% Señal analítica
    %% Semánticamente los índices del vector x van de -(n-1)/2 a (n-1)/2
    %% En Matlab estos índices van de 1 a n respectivamente
    %% Algoritmo
    %% 1: Calcular la transformada inversa de Fourier de fftx
    %%
    x = ifft( fftx );
    %%
    %% Función de Distribución Pseudo-Choi-Williams
    %% Semánticamente los índices del vector kernel van de 0 a (n-1)/2
    %% En Matlab estos índices van de 1 a (n+1)/2 respectivamente
    %% Semánticamente los índices del vector cwi van de -(n+1)/4 a (n+1)/4-1
    %% En Matlab estos índices van de (n+1)/4+1 a (n+1)/2 Y de 1 a (n+1)/4 respectivamente
    %% Algoritmo:
    %% 1: Se calcula kernel(T) = ventana(T) * sumatoria( ... ), T=0:(n-1)/2
    %% 2: Se calcula cwi(k) = 4*real(fft(kernel))-2.*x(0)^2, k=(n+1)/4:(n+1)/4-1
    %%
    for T = 0 : (n-1)/2
        M = -(n-1)/2-abs(T) : ((n-1)/2-abs(T));
        sumatoria = sum( vcw(T+1,M+(n+1)/2) .* x(M+T+(n+1)/2) .* conj( x(M-T+(n+1)/2) ) );
        kernel(T+1) = ventana(T+1) * sumatoria;
    end
    cwi = 4 * real( ifft( kernel ) ) - 2 * x(0+(n+1)/2) .* conj( x(0+(n+1)/2) );
    %%
    %% Distribución de Potencia Pseudo-Instantánea
    %% Semánticamente los índices del vector pow van de 0 a (n+1)/4-1
    %% En Matlab estos índices van desde 1 hasta (n+1)/4 respectivamente
    %% Semánticamente los índices del vector cwi van de -(n+1)/4 a (n+1)/4-1
    %% En Matlab estos índices van de (n+1)/4+1 a (n+1)/2 y de 1 a (n+1)/4 respectivamente
    %% Algoritmo

```

```

%% 1: Potencias negativas se igualan a cero
%% 2: Potencias cero y positivas no se modifican
%%
for k = 0:(n+1)/4-1
    if cwd(k+1) > 0
        pcw(t,k+1) = cwd(k+1);
    else
        pcw(t,k+1) = 0;
    end
end
%%
%% Frecuencia Media Pseudo-Instantánea
%% Semánticamente los índices del vector pcw van de 0 a (n+1)/4-1
%% En Matlab éstos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula  $fcw = \text{sumatoria}(k \cdot pcw(k)) / \text{sumatoria}(pcw(k))$ ,
%%  $k = 0, (n+1)/4-1$ 
%%
sumpcw = sum( pcw(t,:) );
if sumpcw == 0
    fcw(t) = 0;
else
    fcw(t) = sum( (0:(n+1)/4-1) .* pcw(t,:) )/sumpcw;
end
%%
%% Ancho de Banda Espectral RMS
%% Semánticamente los índices del vector pcw van de 0 a (n+1)/4-1
%% En Matlab éstos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula  $bcw = (\text{sumatoria}((fcw - k)^2 \cdot pcw(k)) / \text{sumatoria}(pcw(k)))^{0.5}$ ,  $k = 0, (n+1)/4-1$ 
%%
if sumpcw == 0
    bcw(t) = 0;
else
    bcw(t) = ( sum( (fcw(t)-(0:(n+1)/4-1)).^2 .* pcw(t,:) )/sumpcw )^0.5;
end
end
tiempo = toc; %% Cronómetro
fprintf('\n');
fprintf('Tiempo de ejecución total: %f\n', tiempo);
fprintf('Tiempo de ejecución por transformada: %f\n', tiempo/13314);
%%
%% Estimación de Error RMS en el Ancho de Banda Espectral RMS
%%
biasbcw = mean( bcw - 100/(19200/n) );
desviacionbcw = std( bcw - 100/(19200/n) ) ^2;
errorbcw = ( biasbcw^2 + desviacionbcw^2 )^0.5;
fprintf('Error RMS en el Ancho de Banda Espectral RMS = %12.8f\n', (19200/n)*errorbcw )
%%
%% Estimación de Error en la Frecuencia Media Pseudo-Instantánea
%%
load f; % Frecuencia Instantánea
biasfcw = mean( fcw - f((n+1)/2:13440-(n-1)/2)/(19200/n) );
desviacionfcw = std( fcw - f((n+1)/2:13440-(n-1)/2)/(19200/n) ) ^2;
errorfcw = ( biasfcw^2 + desviacionfcw^2 )^0.5;
fprintf('Error en la Frecuencia Media Pseudo-Instantánea = %12.8f\n', (19200/n)*errorfcw )
%% *****
%% Graficación
%% *****
%%
%% Función real
%%
figure(1)
subplot(3,1,1)
plot( 0:0.7/(13440-1):0.7 , s1 )
ylabel('Amplitud')
title(titulo)
%%
%% Frecuencia Media Pseudo-Instantánea
%%
subplot(3,1,2)
plot( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)), (19200/n)*fcw )
ylabel('Frecuencia (Hz)')
title('Frecuencia Media Pseudo-Instantánea')
axis( [ 0 0.7 0 (19200/n)*max(fcw)*1.1 ] )
%%
%% Ancho de Banda Espectral RMS
%%
subplot(3,1,3)
plot( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)), (19200/n)*bcw )
xlabel('Tiempo (s)')
ylabel('Frecuencia (Hz)')
title('Ancho de Banda Espectral RMS')
axis( [ 0 0.7 0 (19200/n)*max(bcw)*1.1 ] )
%%
%% Distribución de Potencia Pseudo-Instantánea
%%
figura2 = 0;
if figura2 == 0
    figure(2)
    mesh( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)) , 0:19200/n:(19200/n)*((n+1)/4-1)
        , pcw' )
    ylabel('Frecuencia (Hz)')
    xlabel('Tiempo (s)')

```

```

xlabel('Potencia')
title('Distribución de Potencia Pseudo-Instantánea')
axis([ 0 0.7 0 (19200/n)*((n+1)/4-1) 0 max(max(pcw))*1.1 ] )
view(0,90)
end

```

A1.4 Distribución de Bessel

A continuación se lista el código en lenguaje propietario de Matlab del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Bessel.

```

%% .....
%%
%% IIMAS, UNAM
%% Ernesto Rubio Acosta
%% Distribución Tiempo Frecuencia Bessel
%% ** Se analiza un vector de 13440 elementos
%% ** correspondiente a una señal Doppler ultrasónica
%% ** con una ventana de logitud N,  $(-(N-1)/2:(N-1)/2)$  que se traslada
%% ** para N = 31, 63, 127, 255, 511,
%% ** valuada en 0.
%% ** 7 de junio de 1999
%%
%% .....
%%
%% Señal real en todo su dominio
%% Señal Doppler Ultrasónica
%%
load dopplers; % Asigna el vector s1
titulo = 'Señal Doppler Ultrasónica';
%%
%% Longitud de ventana
%%
opcion = input('Opción:Longitud de ventana {1:15, 2:31, 3:63, 4:127, 5:255, 6:511} > ');
if opcion == 1
    n = 15;
elseif opcion == 2
    n = 31;
elseif opcion == 3
    n = 63;
elseif opcion == 4
    n = 127;
elseif opcion == 5
    n = 255;
elseif opcion == 6
    n = 511;
else
    n = 127;
end
%%
%% Ruido
%%
opcion = input('¿Se adiciona Ruido Gaussiano? No=0, Si de otro modo >');
if opcion == 0
    % Espectro de la Señal de Ruido
    ruidoR = randn(1,13440);
    ruidoI = randn(1,13440);
    ruidoC = ruidoR + 1j*ruidoI;
    modRuidoC = abs( ruidoC );
    ruidoN = ruidoC ./ modRuidoC;
    % Señal de Ruido
    ruido = real( ifft( ruidoN ) );
    % Potencia de la Señal Real
    potenciasenal = (sum(s1.^2)/13440)^0.5;
    % Potencia de la Señal de Ruido
    potenciaruido = (sum(ruido.^2)/13440)^0.5;
    % Relacion Señal a Ruido
    SNR = input('Relacion Señal a Ruido en dB-cibels > ');
    potencioactivoruido = potenciasenal/(10^SNR/10);
    factor = potencioactivoruido / potenciaruido;
    s1 = s1 + factor*ruido;
end
%%
%% Inicialización de arreglos
%%
bbl = zeros(13440-(n-1),1); % Ancho de Banda Espectral RMS
fbl = zeros(13440-(n-1),1); % Frecuencia Media Pseudo-Instantánea
pbl = zeros(13440-(n-1),(n+1)/4); % Distribución de Potencia Pseudo-Instantánea
%%
%% Parametro Alfa
%%
alfa = 2;
%%
%% Ventana de Bessel ( Valores Reales )
%% Simultáneamente los índices de la matriz view van de 0 a (n-1)/2 y de -(n-1)/2 a (n-1)/2
%% En Matlab estos índices van de 1 a (n+1)/2 y de 1 a n respectivamente

```

```

%% Algoritmo
%% 1: Se calcula vbl(T,M) = ... , T=:(n-1)/2, M=-(n-1)/2:(n-1)/2, i<>0
%% 2: vbl(0,M) = 0, M=-(n-1)/2:(n-1)/2, M<>0
%% 3: vbl(0,0) = 1
%%
vbl = zeros((n+1)/2,n);
T = 0;
vbl(T+1,(n+1)/2) = 1;
for T = 1 : (n-1)/2;
    limM = min( (n-1)/2 , 2*alfa*abs(T) );
    M = -limM : limM;
    vbl(T+1,M+(n+1)/2) = (1/(pi*alfa*abs(T)))*(1-(M./(2*alfa*T)).^2).^0.5;
end
%%
%% Ventana de Hanning
%% Semánticamente los índices del vector d van de -(n-1)/2 a (n-1)/2
%% En Matlab éstos índices van de 1 a n respectivamente
%% Semánticamente los índices del vector ventana van de 0 a (n-1)/2
%% En Matlab éstos índices van de 1 a (n+1)/2 respectivamente
%% Algoritmo:
%% 1: Se genera una ventana de Hanning de longitud n
%% 2: Se calcula ventana(T) = d(T)d*(-T), T=0:(n-1)/2
%%
d = hanning(n)';
ventana = d((n+1)/2:n) .* conj( d((n+1)/2:-1:1) );
%%
%% Para cada ventana t
%% En un vector de 13440 elementos existen 13440-(n-1) ventanas de longitud 127
%%
tic; %% Cronómetro
for t = 1 : 13440-(n-1)
    if mod(t,100) == 0
        fprintf('.');
    if mod(t,1000) == 0
        fprintf('\n');
    end
end
end
%%
%% Señal real
%% Semánticamente los índices del vector xr van de -(n-1)/2 a (n-1)/2
%% En Matlab éstos índices van de 1 a n respectivamente
%% Algoritmo:
%% 1: Se toma un segmento de n elementos de la señal real
%%
xr = s1(t:t+(n-1));
%%
%% Transformada de Fourier de la señal real
%% Semánticamente los índices del vector fftxr van de 0 a (n-1)/2 y de -(n-1)/2 a -1
%% En Matlab la frecuencia 0 está en el índice 1
%% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
%% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
%% Algoritmo:
%% 1: Se calcula la transformada de Fourier de xr
%%
fftxr = fft(xr);
%%
%% Transformada de Fourier de la señal analítica
%% Semánticamente los índices del vector fftx van de 0 a (n-1)/2 y de -(n-1)/2 a -1
%% En Matlab la frecuencia 0 está en el índice 1 del vector
%% En Matlab las frecuencias positivas están en los índices 2 al (n+1)/2
%% En Matlab las frecuencias negativas están en los índices ((n+1)/2+1) al n
%% Algoritmo
%% 1: Frecuencias negativas se igualan a cero
%% 2: Frecuencia cero no se modifica
%% 3: Frecuencias positivas se duplican
%%
fftx = zeros(1,n);
fftx(1) = fftxr(1);
fftx(2:(n+1)/2) = 2 * fftxr(2:(n+1)/2);
%%
%% Señal analítica
%% Semánticamente los índices del vector x van de -(n-1)/2 a (n-1)/2
%% En Matlab éstos índices van de 1 a n respectivamente
%% Algoritmo
%% 1: Calcular la transformada inversa de Fourier de fftx
%%
x = ifft( fftx );
%%
%% Función de Distribución Pseudo-Bessel
%% Semánticamente los índices del vector kernel van de 0 a (n-1)/2
%% En Matlab éstos índices van de 1 a (n+1)/2 respectivamente
%% Semánticamente los índices del vector bld van de -(n+1)/4 a ((n+1)/4-1)
%% En Matlab éstos índices van de ((n+1)/4+1) a (n+1)/2 Y de 1 a (n+1)/4 respectivamente
%% Algoritmo:
%% 1: Se calcula kernel(T) = ventana(T) sumatoria( ... ), T=0:(n-1)/2
%% 2: Se calcula bld(x) = 4*real(fft(kernel))-2|x(0)|^2, k=-(n+1)/4:((n+1)/4-1)
%%
for T = 0 : (n-1)/2
    limM = min( (n-1)/2-abs(T) , 2*alfa*abs(T) );
    M = -limM : limM;
    sumatoria = sum( vbl(T+1,M+(n+1)/2) .* x(M+T+(n+1)/2) .* conj( x(M-T+(n+1)/2) ) );
    kernel(T+1) = ventana(T+1) * sumatoria;
end
bld = 4 * real( fft( kernel ) ) - 2 * x(0+(n+1)/2) * conj( x(0+(n+1)/2) );
%%

```

```

%% Distribución de Potencia Pseudo-Instantánea
%% Semánticamente los índices del vector pbl van de 0 a ((n+1)/4-1)
%% En Matlab estos índices van desde 1 hasta (n+1)/4 respectivamente
%% Semánticamente los índices del vector bld van de -(n+1)/4 a ((n+1)/4-1)
%% En Matlab estos índices van de ((n+1)/4+1) a ((n+1)/2 y de 1 a ((n+1)/4 respectivamente
%% Algoritmo
%% 1: Potencias negativas se igualan a cero
%% 2: Potencias cero y positivas no se modifican
%%
for k = 0:(n+1)/4-1)
    if bld(k+1) > 0
        pbl(t,k+1) = bld(k+1);
    else
        pbl(t,k+1) = 0;
    end
end
end
%%
%% Frecuencia Media Pseudo-Instantánea
%% Semánticamente los índices del vector pbl van de 0 a ((n+1)/4-1)
%% En Matlab estos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula fbl = sumatoria( k pbl(k) ) / sumatoria( pbl(k) ),
%%                                     k = 0,((n+1)/4-1)
%%
sumpbl = sum( pbl(t,:) );
if sumpbl == 0
    fbl(t) = 0;
else
    fbl(t) = sum( (0:((n+1)/4-1)) .* pbl(t,:) )/sumpbl;
end
end
%%
%% Ancho de Banda Espectral RMS
%% Semánticamente los índices del vector bbl van de 0 a ((n+1)/4-1)
%% En Matlab estos índices van de 1 a (n+1)/4 respectivamente
%% Algoritmo
%% 1: Se calcula bbl = ( sumatoria( ( fbl - k )^2 pbl(k) ) /
%%                                     sumatoria( pbl(k) ) )^0.5, k = 0,((n+1)/4-1)
%%
if sumpbl == 0
    bbl(t) = 0;
else
    bbl(t) = ( sum( (fbl(t)-(0:((n+1)/4-1)))^2 .* pbl(t,:) )/sumpbl )^0.5;
end
end
tiempo = toc; %% Cronometro
fprintf('\n');
fprintf('Tiempo de ejecución total: %f\n', tiempo);
fprintf('Tiempo de ejecución por transformada: %f\n', tiempo/13114);
%%
%% Estimación de Error RMS en el Ancho de Banda Espectral RMS
%%
biasbbl = mean( bbl - 100/(19200/n) );
desviacionbbl = std( bbl - 100/(19200/n) ) ^2;
errorbbl = ( biasbbl^2 + desviacionbbl^2 )^0.5;
fprintf('Error RMS en el Ancho de Banda Espectral RMS = %12.8f\n', (19200/n)*errorbbl )
%%
%% Estimación de Error en la Frecuencia Media Pseudo-Instantánea
%%
load f; % Frecuencia Instantánea
biasfbl = mean( fbl - f((n+1)/2:13440-(n-1)/2)/(19200/n) );
desviacionfbl = std( fbl - f((n+1)/2:13440-(n-1)/2)/(19200/n) ) ^2;
errorfbl = ( biasfbl^2 + desviacionfbl^2 )^0.5;
fprintf('Error en la Frecuencia Media Pseudo-Instantánea = %12.8f\n', (19200/n)*errorfbl )
%%
%% Graficación
%%
%% Funcion real
%%
figure(1)
subplot(3,1,1)
plot( 0:0.7/(13440-1):0.7 , s1 )
ylabel('Amplitud')
title(titulo)
%%
%% Frecuencia Media Pseudo-Instantánea
%%
subplot(3,1,2)
plot( (n-1)/2*0.7/(13440-1):0.7/(13440-1):13440-(n+1)/2*0.7/(13440-1), (19200/n)*fbl )
ylabel('Frecuencia .Hz.')
title('Frecuencia Media Pseudo-Instantánea')
axis( [ 0 0.7 0 (19200/n)*max(fbl)*1.1 ] )
%%
%% Ancho de Banda Espectral RMS
%%
subplot(3,1,3)
plot( (n-1)/2*0.7/(13440-1):0.7/(13440-1):(13440-(n+1)/2)*0.7/(13440-1), (19200/n)*bbl )
xlabel('Tiempo (s)')
ylabel('Frecuencia .Hz.')
title('Ancho de Banda Espectral RMS')
axis( [ 0 0.7 0 (19200/n)*max(bbl)*1.1 ] )
%%
%% Distribución de Potencia Pseudo-Instantánea
%%
figura2 = 0;

```

```

if figura2 == 0
figure(2)
mesh( ((n-1)/2)*(0.7/(13440-1)):0.7/(13440-1):(13440-(n+1)/2)*(0.7/(13440-1)) , 0:19200/n:(19200/n)*((n+1)/4-1)
      , pbl' )
ylabel('Frecuencia (Hz)')
xlabel('Tiempo (s)')
zlabel('Potencia')
title('Distribución de Potencia Pseudo-Instantánea')
axis( [ 0 0.7 0 (19200/n)*((n+1)/4-1) 0 max(max((pbl)))*1.1 ] )
view(0,90)
end

```

A2 PROGRAMAS PARALELOS EN OCCAM

A2.1 Introducción

En este anexo se listan los códigos en Occam de los programas que calculan las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel, así como los códigos de los programas de configuración para uno, dos y cuatro procesadores. Además, se muestran los diagramas de tiempos o de intercambio de mensajes.

A2.2 Distribución de Wigner Ville

A continuación se lista el código en Occam del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Wigner Ville (archivo *corr.occ*).

```
-- *****
-- * MCC, IMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
-- *****
-- *** Distribucion Tiempo Frecuencia ***
-- *** Wigner Ville ***
-- *****
#include "mathvals.inc"
#include "declarar.inc"
-- *****
-- * Proceso corr
-- *****
PROC corr(CHAN OF DATOS dedist, adist)
  #USE "dblmath.lib"
  -- *****
  -- * x : transformada de Hilbert de señal real
  -- * Nx : longitud de x
  -- * kernel : kernel de TFD de señal real
  -- * ventana : ventana de Hanning
  -- *****
  [N-1][2]REAL64 x :
  [N/8][2]REAL64 kernel :
  [N/8]REAL64 ventana :
  INT Nx :
  INT segmento, d1, d2 :
  BOOL trabajar :
  -- *****
  -- * Inicio de proceso corr
  -- *****
  SEQ
  trabajar := TRUE
  -- *****
  -- * Inicializacion
  -- *****
  dedist ? CASE longitud; Nx; segmento
  -- *****
  -- * Ventana de Hanning al cuadrado
  -- *****
  d1 := (segmento*(N/8))+(N/2)
  SEQ i = 0 FOR N/8
  REAL64 tmp :
  SEQ
  tmp := 0.5*(REAL64)*(1.0-(REAL64)-DCOS((2.0*(REAL64)*PI)*(REAL64 ROUND(i+d1)))/(REAL64 ROUND N)))
  ventana[i] := tmp * tmp
  -- *****
  -- * Haz mientras haya datos
  -- *****
  WHILE trabajar
  -- *****
  -- * Recepcion de datos de proceso autocorr
  -- *****
  dedist ? CASE
  -- *****
  -- * Vector
  -- *****
  vector; NX::x
  SEQ
  -- *****
  -- * Correlacion
```

```

-- *****
d1 := ((segmento*(N/8))+(N/2))-1
d2 := (N-2)-d1
SEQ i = 0 FOR N/8
  SEQ
    kernel[i][0] := ventana[i]*((x[d1+i][0]*x[d2-i][0])+(x[d1+i][1]*x[d2-i][1]))
    kernel[i][1] := ventana[i]*((x[d1+i][1]*x[d2-i][0])-(x[d1+i][0]*x[d2-i][1]))
-- *****
-- * Envio de resultados a proceso autocorr
-- *****
adist ! vector; N/8::kernel
-- *****
-- * Terminar
-- *****
terminar
SEQ
  trabajar := FALSE
  adist ! terminar

```

A2.3 Distribución de Choi Williams

A continuación se lista el código en Occam del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Choi Williams (programa *corr.occ*).

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
-- *****
-- *** Distribucion Tiempo Frecuencia ***
-- *** Choi Williams ***
-- *****
#include "mathvals.inc"
#include "declarar.inc"
-- *****
-- * Proceso corr
-- *****
PROC corr(CHAN OF DATOS dedist, adist)
  #USE "dblmath.lib"
  -- *****
  -- * x : transformada de Hilbert de senal real
  -- * Nx : longitud de x
  -- * kernel : kernel de TFD de senal real
  -- * ventana : ventana de Hanning
  -- * choi : ventana de Choi Williams
  -- *****
  [N-1][2]REAL64 x :
  [N/8][2]REAL64 kernel :
  [N/8]REAL64 ventana :
  [N/8][N/2]REAL64 choi :
  INT Nx, segmento, d, absM:
  [2]REAL64 sumatoria :
  BOOL trabajar :
  -- *****
  -- * Inicio de proceso CORR
  -- *****
  SEQ
    trabajar := TRUE
    -- *****
    -- * Inicializacion
    -- *****
    dedist ? CASE longitud; Nx; segmento
    -- *****
    -- * Ventana de Hanning al cuadrado
    -- *****
    d := (segmento*(N/8))+(N/2)
    SEQ i = 0 FOR N/8
      REAL64 tmp :
      SEQ
        tmp := 0.5*(REAL64)*(1.0*(REAL64)-DCOS(((2.0*(REAL64)*PI)*(REAL64 ROUND(i+d)))/(REAL64 ROUND N)))
        ventana[i] := tmp * tmp
      -- *****
      -- * Ventana de Choi Williams
      -- *****
    d := (segmento*(N/8))
    SEQ T = d FOR (N/8)
      SEQ M = 0 FOR N/2
        REAL64 tmp.power, tmp.exp :
        IF
          T <> 0
          SEQ
            tmp.power := sigma/(((4.0*(REAL64)*PI)*(REAL64 ROUND T))*(REAL64 ROUND T))
            tmp.exp := -(((REAL64 ROUND M)*(REAL64 ROUND M))*sigma)/((4.0*(REAL64)*PI)*(REAL64 ROUND T))*(REAL64

```

```

                                ROUND T)))
    choi[T-d][M] := DPCWER( tmp.power, 0.5(REAL64) ) * DEXP( tmp.exp )
T = J
IF
  M <> 0
    choi[T][M] := 0.0(REAL64)
  M = 0
    choi[T][M] := 1.0(REAL64)
-- *****
-- * Haz mientras haya datos
-- *****
WHILE trabajar
-- *****
-- * Recepcion de datos de proceso autocorr
-- *****
dedist ? CASE
-- *****
-- * Vector
-- *****
vector; Nx::x
SEQ
-- *****
-- * Correlacion
-- *****
SEQ T = d FOR N/R
SEQ
  sumatoria[0] := 0.0(REAL64)
  sumatoria[1] := 0.0(REAL64)
  SEQ M = -(N/2)-1-T FOR (N-1,-12*T)
  SEQ
    IF
      M < 0
        absM := -M
      M >= 0
        absM := M
      sumatoria[0] := sumatoria[0] + (choi[T-d][absM] * ((x[(M+T)+(N/2)-1][0] * x[(M-T)+(N/2)-1][1]) + x[(M+T)+(N/2)-1][1] * x[(M-T)+(N/2)-1][0])))
      sumatoria[1] := sumatoria[1] + (choi[T-d][absM] * ((x[(M+T)+(N/2)-1][1] * x[(M-T)+(N/2)-1][0]) - (x[(M+T)+(N/2)-1][0] * x[(M-T)+(N/2)-1][1])))
      kernel[T-d][0] := ventana[T-d] * sumatoria[0]
      kernel[T-d][1] := ventana[T-d] * sumatoria[1]
-- *****
-- * Envio de resultados a proceso autocorr
-- *****
adist ! vector; N/R::kernel
-- *****
-- * Terminar
-- *****
terminar
SEQ
  trabajar := FALSE
  adist ! terminar
:

```

A2.4 Distribución de Bessel

A continuación se lista el código en Occam del programa que calcula la función de autocorrelación instantánea indexada al tiempo para la distribución tiempo frecuencia de Bessel (programa *corr.occ*).

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
-- *****
-- *** Distribucion Tiempo Frecuencia ***
-- *** Bessel ***
-- *****
#INCLUDE "mathval.inc"
#INCLUDE "declarar.inc"
-- *****
-- * Proceso corr
-- *****
PROC corr CHAN OF DATOS dedist, adist:
  #USE "tblmath.lib"
  -- *****
  -- * x : transformada de Hilbert de señal real
  -- * Nx : longitud de x
  -- * kernel : kernel de TED de señal real
  -- * ventana : ventana de Hanning
  -- * Bessel : ventana de Bessel
  -- *****
  [N-1]:REAL64 x :
  [N/R]:REAL64 kernel :
  [N/R]:REAL64 ventana :

```

```

(N/8)[N/2]REAL64 bessel :
INT Nx, segmento, d:
BOOL trabajar :
-- *****
-- * Inicio de proceso corr
-- *****
SEQ
trabajar := TRUE
-- *****
-- * Inicializacion
-- *****
dedist ? CASE longitud; Nx; segmento
-- *****
-- * Ventana de Hanning al cuadrado
-- *****
d := (segmento*(N/8))+(N/2)
SEQ i = 0 FOR N/8
  REAL64 tmp :
  SEQ
  tmp := 0.5(REAL64)*(1.0(REAL64)-DCOS(((2.0(REAL64)*PI)*(REAL64 ROUND(i+d)))/(REAL64 ROUND N)))
  ventana[i] := tmp * tmp
-- *****
-- * Ventana de Bessel
-- *****
d := (segmento*(N/8))
SEQ T = d FOR (N/8)
  SEQ M = 0 FOR N/2
  REAL64 tmp.power :
  IF
  T <> 0
  SEQ
  tmp.power := 1.0(REAL64)-(((REAL64 ROUND M)*(REAL64 ROUND M))/((((4.0(REAL64)*alfa)*alfa)*(REAL64
  ROUND T))*(REAL64 ROUND T))))
  IF
  tmp.power > 0.0(REAL64)
  bessel[T-d][M] := (1.0(REAL64)/((PI*alfa)*(REAL64 ROUND T)))*DPOWER( tmp.power, 0.5(REAL64) )
  tmp.power <= 0.0(REAL64)
  bessel[T-d][M] := 0.0(REAL64)
  T = 0
  IF
  M <> 0
  bessel[T][M] := 0.0(REAL64)
  M = 0
  bessel[T][M] := 1.0(REAL64)
-- *****
-- * Haz mientras haya datos
-- *****
WHILE trabajar
-- *****
-- * Recepcion de datos de proceso autocorr
-- *****
dedist ? CASE
-- *****
-- * Vector
-- *****
vector; Nx::x
SEQ
-- *****
-- * Correlacion
-- *****
SEQ T = d FOR N/8
  [2]REAL64 sumatoria :
  INT limM, absM :
  SEQ
  sumatoria[0] := 0.0(REAL64)
  sumatoria[1] := 0.0(REAL64)
  IF
  (((N/2)-1)-T) < ((2*(INT ROUND alfa))*T)
  limM := ((N/2)-1)-T
  (((N/2)-1)-T) >= ((2*(INT ROUND alfa))*T)
  limM := (2*(INT ROUND alfa))*T
  SEQ M = -limM FOR (2*limM)+1
  SEQ
  IF
  M < 0
  absM := -M
  M >= 0
  absM := M
  sumatoria[0] := sumatoria[0]+(bessel[T-d][absM]*((x[(M+T)+(N/2)-1])[0]*x[(M-T)+(N/2)-
  1])[0])+(x[(M+T)+(N/2)-1])[1]*x[(M-T)+(N/2)-1])[1]))
  sumatoria[1] := sumatoria[1]+(bessel[T-d][absM]*((x[(M+T)+(N/2)-1])[1]*x[(M-T)+(N/2)-
  1])[0])-(x[(M+T)+(N/2)-1])[0]*x[(M-T)+(N/2)-1])[1]))
  kernel[T-d][0] := ventana[T-d]*sumatoria[0]
  kernel[T-d][1] := ventana[T-d]*sumatoria[1]
-- *****
-- * Envio de resultados a proceso autocorr
-- *****
adist ! vector; N/8::kernel
-- *****
-- * Terminar
-- *****
terminar
SEQ
trabajar := FALSE
adist ! terminar

```

A2.5 Programas comunes de procesos

A continuación se listan los códigos en Occam de los programas de procesos que son comunes en la implantación de las distintas distribuciones tiempo frecuencia (archivos *app.occ*, *hilbert1.occ*, *hilbert2.occ*, *autocorr.occ*, *tfd.occ*, *fft2.occ*, *fftb.occ*, *dist.occ*, *declarar.inc*).

A2.5.1 Archivo *declarar.inc*

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****

-- *****
-- * Constantes
-- *****

-- *** Longitud del vector que representa la señal
VAL INT N IS 128 :
-- *** Numero de ventanas procesadas
VAL INT numero.ventanas IS 50 :
-- *** Pi
VAL REAL64 PI IS 3.14159265359(REAL64) :
-- *** Banderas para el proceso fft2
VAL INT forward IS (-1) :
VAL INT backward IS 1 :
-- *** Constante para el cronometro
VAL REAL64 ticks.per.second IS 15625.0(REAL64) :
-- *** Constante Distribucion Choi Williams
VAL REAL64 sigma IS 5.0(REAL64) :
-- *** Constante Distribucion Bessel
VAL REAL64 alta IS 2.0(REAL64) :

-- *****
-- Protocolos
-- *****

PROTOCOL DATOS
CASE
  longitud: INT: INT
  vector: INT: {[2]REAL64
  vector.real: INT: {},REAL64
  terminar
:

```

A2.5.2 Archivo *app.occ*

```

-- .. *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- .. *****
#include "hostio.inc"
#include "declarar.inc"
-- *****
-- * Proceso app
-- *****
PROC app(CHAN OF SP fs, ts, CHAN OF DATOS shilbert1, detfd)
#USE "hostio.lib"
#USE "convert.lib"
-- *****
-- * x : vector real (señal)
-- * N : longitud de x (constante)
-- * tidx : distribucion tiempo frecuencia de x
-- * Ntidx : longitud de tidx
-- *****
[N-1]REAL64 x :
[N/2]REAL64 tidx :
INT Ntidx :
-- *****
-- * Cronometro
-- *****
TIMER clock :

```

```

INT ini.time, end.time, time :
REAL64 time.real :
-- *****
-- * Archivo
-- *****
[14]BYTE registro.archivo : -- registro de longitud fija de 14 caracteres
INT32 f.id :
BYTE f.estatus, lectura.estatus :
BOOL conversion.estatus :
INT bytes.leidos :
-- *****
-- * Inicio de proceso app
-- *****
SEQ
-- *****
-- * Inicializacion
-- * Lectura de archivo "dopplers.dat"
-- *****
so.open( fs, ts, "dopplers.dat", spt.text, spm.input, f.id, f.estatus )
SEQ t = 0 FOR N-1
  SEQ
    so.gets( fs, ts, f.id, bytes.leidos, registro.archivo, lectura.estatus )
    STRINGTOREAL64( conversion.estatus, x[t][0], registro.archivo )
    x[t][1] := 0.0(REAL64)
  so.close( fs, ts, f.id, f.estatus )
  -- *****
  -- * Impresion de datos
  -- *****
  so.write.string.nl(fs, ts, "Senal real x" )
  SEQ t = 0 FOR N-1
    SEQ
      so.write.string(fs, ts, "x(" )
      so.write.int(fs, ts, t, 4)
      so.write.string(fs, ts, ") = " )
      so.write.real64(fs, ts, x[t][0], 4, 15)
      so.write.string(fs, ts, "+" )
      so.write.real64(fs, ts, x[t][1], 4, 15)
      so.write.string.nl(fs, ts, ")i" )
    -- *****
    -- * Calculo de Distribucion Tiempo Frecuencia
    -- *****
  clock ? ini.time
  PAR
    -- *****
    -- * Envio de datos a proceso hilbert1
    -- *****
    SEQ envio = 0 FOR numero.ventanas
      SEQ
        shilbert1 ! vector: N-1::x
        -- *****
        -- * Recepcion de resultados de proceso tfd
        -- *****
        SEQ recepcion = 0 FOR numero.ventanas
          SEQ
            detfd ? CASE vector.real: Ntfdx::tfdx
            clock ? end.time
            -- *****
            -- * Impresion de resultados
            -- *****
            so.write.string.nl(fs, ts, "TFD de senal real x" )
            SEQ t = 0 FOR Ntfdx
              SEQ
                so.write.string(fs, ts, "tfdx(" )
                so.write.int(fs, ts, t, 4)
                so.write.string(fs, ts, ") = " )
                so.write.real64(fs, ts, tfdx[t], 4, 15)
                so.write.string.nl(fs, ts, " " )
              -- *****
            -- * Desempeno
            -- *****
            time := end.time MINUS ini.time
            time.real := (REAL64 ROUND time)/(ticks.per.second*(REAL64 ROUND numero.ventanas))
            so.write.string(fs, ts, "tiempo de proceso por ventana : ")
            so.write.real64(fs, ts, time.real, 0, 0)
            so.write.string.nl(fs, ts, " segundos")
            -- *****
            -- * Finalizacion
            -- *****
            shilbert1 ! terminar
            detfd ? CASE terminar
            so.exit(fs, ts, sps.success)

```

A2.5.3 Archivo hilbert1.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia

```

```

-- *****
#include "declarar.inc"
-- *****
-- * Proceso hilbert1
-- *****
PROC hilbert1(CHAN OF DATOS ahibert2, deapp, afft, defft)
-- *****
-- * x : vector real x (senal)
-- * Nx : longitud de x
-- * fftx : transformada de Fourier de x
-- * NfftX : longitud de fftx
-- *****
[N] [2] REAL64 x, fftx :
INT Nx, NfftX :
BOOL trabajar :
-- * inicio de proceso hilbert1
-- *****
SEQ
  trabajar := TRUE
  -- *****
  -- * Inicializacion de procesos fft2
  -- * -- Longitud N
  -- * -- Transformada (Directa) de Fourier
  -- *****
  afft ! longitud: N; forward
  -- *****
  -- * Inicializacion de x
  -- * Padding en elemento N-esimo
  -- *****
  x[N-1][0] := 0.0(REAL64)
  x[N-1][1] := 0.0(REAL64)
  -- *****
  -- * Haz mientras haya datos
  -- *****
  WHILE trabajar
  -- *****
  -- * Recepcion de datos de proceso app
  -- *****
  deapp ? CASE
  -- *****
  -- * Vector
  -- *****
  vector: Nx::[x FROM 0 FOR N-1]
  SEQ
    -- *****
    -- * Envio de datos a procesos fft2
    -- *****
    afft ! vector: Nx:x
    -- *****
    -- * Recepcion de resultados de procesos fft2
    -- *****
    defft ? CASE vector: NfftX::fftx
    -- *****
    -- * Envio de datos a proceso hilbert2
    -- * Se requeriran solamente los elementos asociados con
    -- * las frecuencias centrales y las positivas
    -- *****
    ahibert2 ! vector: (N/2)+1::[fftx FROM 0 FOR (N/2)+1 ]
    -- *****
  -- * Terminar
  -- *****
  terminar
  terminar
  SEQ
    trabajar := FALSE
    afft . terminar
    defft ? CASE terminar
    ahibert2 ! terminar

```

A2.5.4 Archivo hilbert2.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "declarar.inc"
-- *****
-- * Proceso hilbert2
-- *****
PROC hilbert2(CHAN OF DATOS autocorr, dehilbert1, afft, defft)
-- *****
-- * fftx : transformada de Fourier de senal real
-- * NfftX : longitud de x
-- * hx : transformada de Hilbert de senal real
-- * Nhx : longitud de hx
-- *****
[N] [2] REAL64 fftx, hx :

```

```

INT NfftX, Nhx :
BOOL trabajar :
-- *****
-- * Inicio de proceso hilbert2
-- *****
SEQ
  trabajar := TRUE
  -- *****
  -- * Inicializacion de procesos fft2
  -- * -- Longitud N
  -- * -- Transformada Inversa de Fourier
  -- *****
  afft ! longitud; N; backward
  -- *****
  -- * Inicializacion de fftx
  -- * Frecuencias negativas igualadas a cero
  -- *****
  SEQ i = (N/2)+1 FOR (N/2)-1
    SEQ
      fftx[i][0] := 0.0(REAL64)
      fftx[i][1] := 0.0(REAL64)
      -- *****
      -- * Haz mientras haya datos
      -- *****
  WHILE trabajar
    -- *****
    -- * Recepcion de datos de proceso hilbert1
    -- *****
    dehilbert1 ? CASE
      -- *****
      -- * Vector
      -- *****
      vector; NfftX:([ fftx FROM 0 FOR (N/2)+1 ])
      SEQ
        -- *****
        -- * Frecuencias centrales sin cambio
        -- * Frecuencias positivas multiplicadas por 2
        -- *****
        SEQ i = 1 FOR (N/2)-1
          SEQ
            fftx[i][0] := fftx[i][0]*2.0(REAL64)
            fftx[i][1] := fftx[i][1]*2.0(REAL64)
            -- *****
            -- * Envio de datos a procesos fft2
            -- *****
            afft ! vector; N::fftx
            -- *****
            -- * Recepcion de resultados de procesos fft2
            -- *****
            defft ? CASE vector; Nhx:hx
            -- *****
            -- * Envio de datos a proceso autocorr
            -- *****
            autocorr ! vector; N-1:([hx FROM 0 FOR N-1])
            -- *****
            -- * Terminar
            -- *****
            terminar
            SEQ
              trabajar := FALSE
              afft ! terminar
              defft ? CASE terminar
              autocorr ! terminar

```

A2.5.5 Archivo autocorr.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "declarar.inc"
-- *****
-- * Proceso autocorr
-- *****
PROC autocorr(CHAN OF DATOS atfd, dehilbert2, adist1, dedist1, adist2, dedist2)
-- *****
-- * hx : transformada de Hilbert de senal real
-- * Nhx : longitud de x
-- * kernel : kernel de TFD de senal real
-- * Nk : longitud de kernel
-- * kernel2 : kernel de TFD de senal real (segunda mitad)
-- * Nk2 : longitud de kernel2
-- *****
[N-1][2]REAL64 hx :
[N/2][2]REAL64 kernel :
[N/4][2]REAL64 kernel2 :
INT Nhx, Nk, Nk2 :

```

```

BOOL trabajar :
-- *****
-- * Inicio de proceso autocorr
-- *****
SEQ
  trabajar := TRUE
  -- *****
  -- * Inicializacion de procesos corr a traves de procesos dist
  -- *****
  PAR
    adist1 : longitud; N; 0
    adist2 : longitud; N; 1
    -- *****
    -- * Haz mientras haya datos
    -- *****
  WHILE trabajar
    -- *****
    -- * Recepcion de datos de proceso hilbert2
    -- *****
    dehilbert2 ? CASE
      -- *****
      -- * Vector
      -- *****
      vector; Nhx::hx
      SEQ
        -- *****
        -- * Envio de datos a procesos corr a traves de procesos dist
        -- * Envio de dato hx(0) a proceso tfd
        -- *****
        PAR
          adist1 : vector; N-1::hx
          adist2 : vector; N-1::hx
          atfd : vector; 1::[hx FROM (N/2)-1 FOR 1]
          -- *****
          -- * Recepcion de resultados de procesos corr a traves de procesos dist
          -- *****
          PAR
            dedist1 ? CASE vector; Nk::[kernel FROM 0 FOR N/4]
            dedist2 ? CASE vector; Nk2::kernel2
            -- *****
            -- * Union de resultados de procesos corr
            -- *****
            SEQ i = 0 FOR N/4
              INT d :
                SEQ
                  d := i+(N/4)
                  kernel[d][0] := kernel2[i][0]
                  kernel[d][1] := kernel2[i][1]
                  -- *****
                  -- * Envio de datos a proceso tfd
                  -- *****
                  atfd ! vector; N/2::kernel
            -- *****
            -- * Terminar
            -- *****
          terminar
        SEQ
          trabajar := FALSE
        PAR
          SEQ
            adist1 ! terminar
            dedist1 ? CASE terminar
          SEQ
            adist2 ! terminar
            dedist2 ? CASE terminar
          atfd ! terminar

```

A2.5.6 Archivo tfd.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "declarar.inc"
-- *****
-- * Proceso *fd
-- *****
PROC tfd(CHAN OF DATOS sapp, deautocorr, afft, defft)
-- *****
-- * k : kernel de TFD de senal real
-- * Nk : longitud de k
-- * fftk : transformada de Fourier de k
-- * Nfftk : longitud de fftk
-- * tfdk : TFD de senal real
-- * x0 : elemento 0 de senal real
-- *****
(N/2)::[2,REAL64 k, fftk :

```

```

[1][2]REAL64 x0 :
[N/2]REAL64 tfdx :
REAL64 mx0 :
INT Nx0, Nk, NfftK :
BOOL trabajar :
-- *****
-- * Inicio de proceso tfd
-- *****
SEQ
  trabajar := TRUE
  -- *****
  -- * Inicializacion de procesos fft2
  -- * -- Longitud N/2
  -- * -- Transformada (Directa) de Fourier
  -- *****
  afft ! longitud; N/2; forward
  -- *****
  -- * Haz mientras haya datos
  -- *****
  WHILE trabajar
  -- *****
  -- * Recepcion de datos de proceso autocorr
  -- *****
  deautocorr ? CASE
  -- *****
  -- * Vector
  -- *****
  vector; Nx0::x0
  SEQ
    deautocorr ? CASE vector; Nk::k
    -- *****
    -- * Envio de datos a procesos fft2
    -- *****
    afft ! vector; N/2::k
    -- *****
    -- * Recepcion de resultados de procesos fft2
    -- *****
    defft ? CASE vector; NfftK::fftK
    -- *****
    -- * Procesamiento de la transformada tiempo frecuencia
    -- *****
    mx0 := 2.0(REAL64)*((x0[0][0]*x0[0][0])+(x0[0][1]*x0[0][1]))
    SEQ i = 0 FOR N/2
      SEQ
        tfdx[i] := (4.0(REAL64)*fftK[i][0]) - mx0
        -- *****
        -- * Envio de datos a proceso app
        -- *****
        aapp ! vector.real; N/2::tfdx
        -- *****
  -- * Terminar
  -- *****
  terminar
  SEQ
    trabajar := FALSE
    afft ! terminar
    defft ? CASE terminar
    aapp ! terminar

```

A2.5.7 Archivo fft2.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "mathvals.inc"
#include "declarar.inc"
-- *****
-- * Proceso fft2
-- *****
PROC fft2(CHAN OF DATOS aHijo1, aHijo2, dePadre, deHijo1, deHijo2, aPadre)
  #USE "dblmath.lib"
  -- *****
  -- Factores para el algoritmo FFT base 2 diezmado en frecuencia
  -- *****
  PROC factores([1][2]REAL64 w, INT Nx, tipo)
    SEQ i = 0 FOR Nx/2
      REAL64 arg :
      SEQ
        arg := (REAL64 ROUND(tipo))*((2.0(REAL64)*PI)*(REAL64 ROUND(i)))/(REAL64 ROUND(Nx))
        w[i][0] := DCOS( arg )
        w[i][1] := DSIN( arg )
  :
  -- *****
  -- Mariposa para el algoritmo FFT base 2 diezmado en frecuencia
  -- *****
  PROC mariposa([1][2]REAL64 x, fftK1, fftK2, w, INT Nx)

```

```

INT Nx2 :
BCOL trabajar :
SEQ
  trabajar := TRUE
  -- * Haz mientras haya datos
  -- *
  WHILE trabajar
  -- * Recepcion de datos de etapa anterior
  -- *
  dePadre ? CASE
  -- * Vector
  -- *
  vector: Nx::x
  SEQ
    Nx2 := Nx/2
    PAR
      -- * Secuencia Par
      -- *
      SEQ
        -- * Calculo de datos para etapa siguiente
        -- *
        SEQ i = 0 FOR Nx2
          SEQ
            -- Suma parte real
            fftx1[i][0] := x[i][0] + x[i+Nx2][0]
            -- Suma parte imaginaria
            fftx1[i][1] := x[i][1] + x[i+Nx2][1]
            -- * Envio de datos a etapa siguiente
            -- *
            aHijo1 ! vector: Nx2::fftx1
            -- * Recepcion de resultados de etapa siguiente
            -- *
            INT Nfftx1 :
            SEQ
              deHijo1 ? CASE vector: Nfftx1::fftx1
              -- * Secuencia Impar
              -- *
              SEQ
                -- * Calculo de datos para etapa siguiente
                -- *
                -- Primera etapa
                SEQ i = 0 FOR Nx2
                  INT d :
                  SEQ
                    d := i + Nx2
                    -- Suma parte real
                    fftx2[i][0] := x[i][0] - x[d][0]
                    -- Suma parte imaginaria
                    fftx2[i][1] := x[i][1] - x[d][1]
                -- Segunda etapa
                SEQ i = 0 FOR Nx2
                  REAL64 fft.real :
                  SEQ
                    -- Multiplicacion parte real
                    fft.real := (fftx2[i][0]*w[i][0]) - (fftx2[i][1]*w[i][1])
                    -- Multiplicacion parte imaginaria
                    fftx2[i][1] := (fftx2[i][0]*w[i][1]) + (fftx2[i][1]*w[i][0])
                    fftx2[i][0] := fft.real
                    -- * Envio de datos a etapa siguiente
                    -- *
                    aHijo2 ! vector: Nx2::fftx2
                    -- * Recepcion de resultados de etapa siguiente
                    -- *
                    INT Nfftx2 :
                    SEQ
                      deHijo2 ? CASE vector: Nfftx2::fftx2
                      -- * Envio de resultados de etapa anterior
                      -- *
                      SEQ i = 0 FOR Nx2
                        INT d :
                        SEQ
                          d := i + 2
                          x[d][0] := fftx1[i][0]
                          x[d][1] := fftx1[i][1]
                          x[d+1][0] := fftx2[i][0]
                          x[d+1][1] := fftx2[i][1]
                    aPadre ! vector: Nx::x
                    -- * Terminar
                    -- *
                    terminar
                    SEQ
                      trabajar := FALSE

```

```

PAR
  SEQ
    aHijo1 ! terminar
    deHijo1 ? CASE terminar
  SEQ
    aHijo2 ! terminar
    deHijo2 ? CASE terminar
  aPadre ! terminar
:
-- *****
-- * x : vector complejo
-- * Nx : longitud de x
-- * fftx1 : transformada de Fourier de x (primera mitad)
-- * fftx2 : transformada de Fourier de x (segunda mitad)
-- * w : factores de giro para transformada de Fourier
-- * tipo : transformada de Fourier Directa o Inversa
-- *****
INT Nx, tipo :
-- *****
-- * Inicio de proceso fft2
-- *****
SEQ
-- *****
-- * Inicializacion de procesos fft2
-- *****
dePadre ? CASE longitud: Nx; tipo
PAR
  aHijo1 : longitud: Nx/2; tipo
  aHijo2 : longitud: Nx/2; tipo
CASE Nx
256
  [256][2]REAL64 x :
  [128][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
128
  [128][2]REAL64 x :
  [64][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
64
  [64][2]REAL64 x :
  [32][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
32
  [32][2]REAL64 x :
  [16][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
16
  [16][2]REAL64 x :
  [8][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
8
  [8][2]REAL64 x :
  [4][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
4
  [4][2]REAL64 x :
  [2][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
2
  [2][2]REAL64 x :
  [1][2]REAL64 fftx1, fftx2, w :
  SEQ
    factores(w, Nx, tipo)
    mariposa(x, fftx1, fftx2, w, Nx)
:

```

A2.5.8 Archivo fftb.occ

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "declarar.inc"
-- *****

```

```

-- * Proceso fftb
-- .....
PROC fftb(CHAN OF DATOS deNodo, aNodo)
  [1][2]REAL64 buffer :
  INT Nb, tipo :
  BOOL trabajar :
  -- .....
  -- * Inicio de proceso fftb
  -- .....
  SEQ
  trabajar := TRUE
  deNodo ? CASE longitud; Nb; tipo
  -- .....
  -- * Haz mientras haya datos
  -- .....
  WHILE trabajar
  -- * .....
  -- * Recepcion de datos de etapa anterior
  -- .....
  deNodo ? CASE
  -- * .....
  -- * Vector
  -- .....
  vector; Nb::buffer
  SEQ
  IF
  -- * .....
  -- Transformada inversa de Fourier
  -- .....
  tipo = backward
  SEQ
  buffer[0][0] := buffer[0][0]/(REAL64 ROUND(N))
  buffer[0][1] := buffer[0][1]/(REAL64 ROUND(N))
  -- * .....
  -- Transformada (directa) de Fourier
  -- .....
  tipo = forward
  SKIP
  TRUE
  SKIP
  -- * .....
  -- * Envio de resultados a etapa anterior
  -- .....
  aNodo ! vector; Nb::buffer
  -- * .....
  -- * Terminar
  -- .....
  terminar
  SEQ
  trabajar := FALSE
  aNodo ! terminar
  :

```

A2.5.9 Archivo dist.occ

```

-- .....
-- * MCT, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- .....
#include "declarar.inc"
-- * Proceso dist
-- .....
PROC dist(CHAN OF DATOS aCorr1, aCorr2, deAuto, deCorr1, deCorr2, aAuto)
  -- .....
  -- * hx : transformada de Hilbert de senal real
  -- * Nhx : longitud de hx
  -- * k1 : kernel de TFD de senal real
  -- * Nk1 : longitud de k1
  -- * k2 : kernel de TFD de senal real (segunda mitad)
  -- * Nk2 : longitud de k2
  -- .....
  [N-1][2]REAL64 hx :
  [N/4][2]REAL64 k1 :
  [N/4][2]REAL64 k2 :
  INT Nhx, Nk1, Nk2 :
  INT segmento :
  BOOL trabajar :
  -- .....
  -- * Inicio de proceso dist
  -- .....
  SEQ
  trabajar := TRUE
  -- * .....
  -- * Inicializacion de procesos corr
  -- .....
  deAuto ? CASE longitud; Nhx; segmento
  PAR

```

```

aCorr1 ! longitud; N; segmento*2
aCorr2 ! longitud; N; (segmento*2)+1
-- *****
-- * Haz mientras haya datos
-- *****
WHILE trabajar
-- *****
-- * Recepcion de datos de proceso autocorr
-- *****
deAuto ? CASE
-- *****
-- * Vector
-- *****
vector; Mhk::hx
SEQ
  PAR
  -- *****
  -- * Envio de datos y recepcion de resultados de proceso corr1
  -- *****
  SEQ
    aCorr1 ! vector; N-1::hx
    deCorr1 ? CASE vector; Nk1::[k1 FROM 0 FOR N/8]
    -- *****
    -- * Envio de datos y recepcion de resultados de proceso corr2
    -- *****
    SEQ
      aCorr2 ! vector; N-1::hx
      deCorr2 ? CASE vector; Nk2::k2
      -- *****
      -- * Union de resultados de procesos corr
      -- *****
      SEQ i = 0 FOR N/8
        INT d :
          SEQ
            d := i+(N/8)
            k1[d][0] := k2[1][0]
            k1[d][1] := k2[1][1]
            -- *****
            -- * Envio de resultados a proceso autocorr
            -- *****
            aAuto ! vector; N/4::k1
          -- *****
          -- * Terminar
          -- *****
          terminar
        SEQ
          trabajar := FALSE
      PAR
      SEQ
        aCorr1 ! terminar
        deCorr1 ? CASE terminar
      SEQ
        aCorr2 ! terminar
        deCorr2 ? CASE terminar
      aAuto ! terminar

```

A2.6 Programas comunes de configuración

A continuación se listan los códigos de los programas de configuración que son comunes en la implantación de las distintas distribuciones tiempo frecuencia (*archivos main.pgm, main1.pgm, main2.pgm, main4.pgm, main4b.pgm*).

A2.6.1 Un procesador sin configuración: archivo main.pgm

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- *****
#include "hostio.inc"
#include "declarar.inc"
-- *****
-- * Proceso Main
-- *****
PROC Main(CHAN OF SP fs, ts, [INT memory])
  #USE "hostio.lib"
  #USE "fuentes.tco"
  -- *****
  -- * #USE "app.tco"
  -- * #USE "hilbert1.tco"
  -- * #USE "hilbert2.tco"

```

```

-- * #USE "autocorr.tco"
-- * #USE "dist.tco"
-- * #USE "corr.tco"
-- * #USE "tfd.tco"
-- * #USE "fft2.tco"
-- * #USE "ftb.tco"
-- *****
[(2*N)-1]CHAN OF DATOS afft1, defft1 :
[(2*N)-1]CHAN OF DATOS afft2, defft2 :
[N-1]CHAN OF DATOS afft3, defft3 :
[2]CHAN OF DATOS adist, dedist :
[4]CHAN OF DATOS acorr, decorr :
CHAN OF DATOS app.hilbert1, hilbert1.hilbert2, hilbert2.autocorr,
autocorr.tfd, tfd.app :
-- *****
-- * Inicio de proceso Main
-- *****
SEQ
PAR
-- *****
-- Proceso Aplicacion
-- *****
app( fs, ts, app.hilbert1, tfd.app )
-- *****
-- * Proceso Hilbert
-- *****
hilbert1( hilbert1.hilbert2, app.hilbert1, afft1[0], defft1[0] )
hilbert2( hilbert2.autocorr, hilbert1.hilbert2, afft2[0], defft2[0] )
-- *****
-- * Proceso Autocorrelacion
-- *****
autocorr( autocorr.tfd, hilbert2.autocorr, adist[0], dedist[0],
adist[1], dedist[1] )
-- *****
-- * Proceso Distribucion Tiempo Frecuencia
-- *****
tfd( tfd.app, autocorr.tfd, afft3[0], defft3[0] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert1
-- *****
PAR nodo = 0 FOR N-1
fft2( afft1[(nodo*2)+1], afft1[(nodo*2)+2], afft1[nodo],
defft1[(nodo*2)+1], defft1[(nodo*2)+2], defft1[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert1
-- *****
PAR nodo = N-1 FOR N
ftb( afft1[nodo], defft1[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert2
-- *****
PAR nodo = 0 FOR N-1
fft2( afft2[(nodo*2)+1], afft2[(nodo*2)+2], afft2[nodo],
defft2[(nodo*2)+1], defft2[(nodo*2)+2], defft2[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert2
-- *****
PAR nodo = N-1 FOR N
ftb( afft2[nodo], defft2[nodo] )
-- *****
-- Procesos de distribucion para proceso autocorr
-- *****
PAR nodo = 0 FOR 2
dist( acorr[nodo*2], acorr[(nodo*2)+1], adist[nodo],
decorr[nodo*2], decorr[(nodo*2)+1], dedist[nodo] )
-- *****
-- Procesos de correlacion para proceso autocorr
-- *****
PAR nodo = 0 FOR 4
corr( acorr[nodo], decorr[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso tfd
-- *****
PAR nodo = 0 FOR (N/2)-1
fft2( afft3[(nodo*2)+1], afft3[(nodo*2)+2], afft3[nodo],
defft3[(nodo*2)+1], defft3[(nodo*2)+2], defft3[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso tfd
-- *****
PAR nodo = (N/2)-1 FOR (N/2)
ftb( afft3[nodo], defft3[nodo] )

```

A2.6.2 Un procesador con configuración: archivo main1.pgm

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia

```

```

-- * Para 1 procesador con configuracion
-- *****
VAL K IS 1024 :
VAL M IS K*K :
VAL [511]INT p IS [ 0, 0, 2, 0, 1, 2, 3, 0, 0, 1, 1, 2, 2, 3, 3,
0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 ] :

-- *****
-- * Descripcion de hardware
-- *****
[1]NODE T:
ARC Hostlink :
NETWORK
DO
  SET T[0] (type, memsize := "T805", 4*M)
  CONNECT T[0][link][0] TO HOST WITH Hostlink
:

-- *****
-- * Descripcion de mapeo
-- *****
NODE app.p :
[4]NODE hilbert1.p, hilbert2.p, autocorr.p, tfd.p :
MAPPING
DO
  MAP app.p ONTO T[0]
  DO i=0 FOR 4
    MAP hilbert1.p[i] ONTO T[0]
  DO i=0 FOR 4
    MAP hilbert2.p[i] ONTO T[0]
  DO i=0 FOR 4
    MAP autocorr.p[i] ONTO T[0]
  DO i=0 FOR 4
    MAP tfd.p[i] ONTO T[0]
:

-- *****
-- * Descripcion del software
-- *****
#include "hostio.inc"
#include "declarar.inc"
#USE "app.lku"
#USE "hilbert1.lku"
#USE "hilbert2.lku"
#USE "autocorr.lku"
#USE "dist.lku"
#USE "corr.lku"
#USE "tfd.lku"
#USE "fft2.lku"
#USE "ftb.lku"
CONFIG
CHAN OF SP fs, ts :
PLACE fs, ts ON Hostlink :
[(2*N)-1]CHAN OF DATOS afft1, defft1 :
[(2*N)-1]CHAN OF DATOS afft2, defft2 :
[N-1]CHAN OF DATOS afft3, defft3 :
[2]CHAN OF DATOS adist, dedist :
[4]CHAN OF DATOS acorr, decorr :
CHAN OF DATOS app.hilbert1, hilbert1.hilbert2, hilbert2.autocorr,
autocorr.tfd, tfd.app :
PAR
  PAR
    -- *****
    -- Proceso Aplicacion
    -- *****

```

```

PROCESSOR app.p
  app( fs, ts, app.hilbert1, tfd.app )
  .. *****
  -- * Proceso Hilbert
  .. *****
PROCESSOR hilbert1.p[0]
  hilbert1( hilbert1.hilbert2, app.hilbert1, afft1[0], defft1[0] )
PROCESSOR hilbert2.p[0]
  hilbert2( hilbert2.autocorr, hilbert1.hilbert2, afft2[0], defft2[0] )
  .. *****
  -- * Proceso Autocorrelacion
  .. *****
PROCESSOR autocorr.p[0]
  autocorr( autocorr.tfd, hilbert2.autocorr, adist[0], dedist[0],
            adist[1], dedist[1] )
  .. *****
  -- * Proceso Distribucion Tiempo Frecuencia
  .. *****
PROCESSOR tfd.p[0]
  tfd( tfd.app, autocorr.tfd, afft3[0], defft3[0] )
  .. *****
  -- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert1
  .. *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert1.p[p[nodo]]
    fft2( afft1[(nodo*2)+1], afft1[(nodo*2)+2], afft1[nodo],
          defft1[(nodo*2)+1], defft1[(nodo*2)+2], defft1[nodo] )
  .. *****
  -- Procesos de retorno para fft2 para proceso hilbert1
  .. *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert1.p[p[nodo]]
    fftb( afft1[nodo], defft1[nodo] )
  .. *****
  -- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert2
  .. *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert2.p[p[nodo]]
    tft2( afft2[(nodo*2)+1], afft2[(nodo*2)+2], afft2[nodo],
          defft2[(nodo*2)+1], defft2[(nodo*2)+2], defft2[nodo] )
  .. *****
  -- Procesos de retorno para fft2 para proceso hilbert2
  .. *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert2.p[p[nodo]]
    fftb( afft2[nodo], defft2[nodo] )
  .. *****
  -- Procesos de distribucion para proceso autocorr
  .. *****
PAR nodo = 0 FOR 2
  PROCESSOR autocorr.p[nodo*2]
    dist( acorr[nodo*2], acorr[(nodo*2)+1], adist[nodo],
          decorr[nodo*2], decorr[(nodo*2)+1], dedist[nodo] )
  .. *****
  -- Procesos de correlacion para proceso autocorr
  .. *****
PAR nodo = 0 FOR 4
  PROCESSOR autocorr.p[nodo]
    corr( acorr[nodo], decorr[nodo] )
  .. *****
  -- Procesos FFT base 2 diezmado en frecuencia para proceso tfd
  .. *****
PAR nodo = 0 FOR (N/2)-1
  PROCESSOR tfd.p[p[nodo]]
    fft2( afft3[(nodo*2)+1], afft3[(nodo*2)+2], afft3[nodo],
          defft3[(nodo*2)+1], defft3[(nodo*2)+2], defft3[nodo] )
  .. *****
  -- Procesos de retorno para fft2 para proceso tfd
  .. *****
PAR nodo = (N/2)-1 FOR (N/2)
  PROCESSOR tfd.p[p[nodo]]
    fftb( afft3[nodo], defft3[nodo] )

```

A2.6.3 Dos procesadores con configuración: archivo main2.pgm

```

.. *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1993
-- * Distribucion Tiempo Frecuencia
-- * Para 2 procesadores con configuracion
.. *****
VAL K IS 1024 :
VAL M IS K*K :
VAL {511}INT p IS [ 0, 0, 2, 0, 1, 2, 3, 0, 0, 1, 1, 2, 2, 3, 3,
                   0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3,
                   0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
                   2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
                   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

```



```

PROCESSOR hilbert2.p[0]
  hilbert2( hilbert2.autocorr, hilbert1.hilbert2, afft2[0], defft2[0] )
-- *****
-- * Proceso Autocorrelacion
-- *****
PROCESSOR autocorr.p[0]
  autocorr( autocorr.tfd, hilbert2.autocorr, adist[0], dedist[0],
            adist[1], dedist[1] )
-- *****
-- * Proceso Distribucion Tiempo Frecuencia
-- *****
PROCESSOR tfd.p[0]
  tfd( tfd.app, autocorr.tfd, afft3[0], defft3[0] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert1
-- *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert1.p[p[nodo]]
    fft2( afft1[(nodo*2)+1], afft1[(nodo*2)+2], afft1[nodo],
          defft1[(nodo*2)+1], defft1[(nodo*2)+2], defft1[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert1
-- *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert1.p[p[nodo]]
    fftb( afft1[nodo], defft1[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert2
-- *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert2.p[p[nodo]]
    fft2( afft2[(nodo*2)+1], afft2[(nodo*2)+2], afft2[nodo],
          defft2[(nodo*2)+1], defft2[(nodo*2)+2], defft2[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert2
-- *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert2.p[p[nodo]]
    fftb( afft2[nodo], defft2[nodo] )
-- *****
-- Procesos de distribucion para proceso autocorr
-- *****
PAR nodo = 0 FOR 2
  PROCESSOR autocorr.p[nodo*2]
    dist( acorr[nodo*2], acorr[(nodo*2)+1], adist[nodo],
          decorr[nodo*2], decorr[(nodo*2)+1], dedist[nodo] )
-- *****
-- Procesos de correlacion para proceso autocorr
-- *****
PAR nodo = 0 FOR 4
  PROCESSOR autocorr.p[nodo]
    corr( acorr[nodo], decorr[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso tfd
-- *****
PAR nodo = 0 FOR (N/2)-1
  PROCESSOR tfd.p[p[nodo]]
    fft2( afft3[(nodo*2)+1], afft3[(nodo*2)+2], afft3[nodo],
          defft3[(nodo*2)+1], defft3[(nodo*2)+2], defft3[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso tfd
-- *****
PAR nodo = (N/2)-1 FOR (N/2)
  PROCESSOR tfd.p[p[nodo]]
    fftb( afft3[nodo], defft3[nodo] )

```

A2.6.4 Cuatro procesadores con configuración tipo I: archivo main4.pgm

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- * Para 4 procesadores con configuracion
-- *****
VAL K IS 1024 :
VAL M IS K*K :
VAL [511]INT p IS {
  0, 0, 2, 0, 1, 2, 3, 0, 0, 1, 1, 2, 2, 3, 3,
  0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3,
  0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
  2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

```

```

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 ] :

-- *****
-- * Descripcion de hardware
-- *****
[4]NODE T:
ARC Hostlink :
NETWORK
DO
  SET T[0] (type, memsize := "T805", 4*M )
  SET T[1] (type, memsize := "T805", 4*M )
  SET T[2] (type, memsize := "T805", 4*M )
  SET T[3] (type, memsize := "T805", 4*M )
  CONNECT T[0][link][0] TO HOST WITH Hostlink
  CONNECT T[0][link][2] TO T[1][link][1]
  CONNECT T[1][link][2] TO T[2][link][1]
  CONNECT T[2][link][2] TO T[3][link][1]
  CONNECT T[3][link][0] TO T[0][link][3]
:

-- *****
-- * Descripcion de mapeo
-- *****
NODE app.p :
[4]NODE hilbert1.p, hilbert2.p, autocorr.p, tfd.p :
MAPPING
DO
  MAP app.p ONTO T[0]
  DO i=0 FOR 4
    MAP hilbert1.p[i] ONTO T[0]
  DO i=0 FOR 4
    MAP hilbert2.p[i] ONTO T[1]
  DO i=0 FOR 4
    MAP autocorr.p[i] ONTO T[2]
  DO i=0 FOR 4
    MAP tfd.p[i] ONTO T[3]
:

-- *****
-- * Descripcion del software
-- *****
#include "hostio.inc"
#include "declarar.inc"
#USE "app.lku"
#USE "hilbert1.lku"
#USE "hilbert2.lku"
#USE "autocorr.lku"
#USE "dist.lku"
#USE "corr.lku"
#USE "tfd.lku"
#USE "fft2.lku"
#USE "fftb.lku"
CONFIG
CHAN OF SP fs, ts :
PLACE fs, ts ON Hostlink :
[(2*N)-1]CHAN OF DATOS afft1, defft1 :
[(2*N)-1]CHAN OF DATOS afft2, defft2 :
[N-1]CHAN OF DATOS afft3, defft3 :
[2]CHAN OF DATOS adist, dedist :
[4]CHAN OF DATOS acorr, decorr :
CHAN OF DATOS app.hilbert1, hilbert1.hilbert2, hilbert2.autocorr,
autocorr.tfd, tfd.app :
PAR
PAR
-- *****
-- Proceso Aplicacion
-- *****
PROCESSOR app.p
app( fs, ts, app.hilbert1, tfd.app )
-- *****
-- * Proceso Hilbert
-- *****
PROCESSOR hilbert1.p[0]
hilbert1( hilbert1.hilbert2, app.hilbert1, afft1[0], defft1[0] )
PROCESSOR hilbert2.p[0]
hilbert2( hilbert2.autocorr, hilbert1.hilbert2, afft2[0], defft2[0] )

```

```

-- *****
-- * Proceso Autocorrelacion
-- *****
PROCESSOR autocorr.p[0]
autocorr( autocorr.tfd, hilbert2.autocorr, adist[0], dedist[0],
          adist[1], dodist[1] )
-- *****
-- * Proceso Distribucion Tiempo Frecuencia
-- *****
PROCESSOR tfd.p[0]
tfd( tfd.app, autocorr.tfd, afft3[0], defft3[0] )
-- *****
-- Procesos FFT base 2 diazmado en frecuencia para proceso hilbert1
-- *****
PAR nodo = 0 FOR N-1
PROCESSOR hilbert1.p[p[nodo]]
fft2( afft1[(nodo*2)+1], afft1[(nodo*2)+2], afft1[nodo],
      defft1[(nodo*2)+1], defft1[(nodo*2)+2], defft1[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert1
-- *****
PAR nodo = N-1 FOR N
PROCESSOR hilbert1.p[p[nodo]]
fftb( afft1[nodo], defft1[nodo] )
-- *****
-- Procesos FFT base 2 diazmado en frecuencia para proceso hilbert2
-- *****
PAR nodo = 0 FOR N-1
PROCESSOR hilbert2.p[p[nodo]]
fft2( afft2[(nodo*2)+1], afft2[(nodo*2)+2], afft2[nodo],
      defft2[(nodo*2)+1], defft2[(nodo*2)+2], defft2[nodo] )
-- *****
-- Procesos de retorno para ffr2 para proceso hilbert2
-- *****
PAR nodo = N-1 FOR N
PROCESSOR hilbert2.p[p[nodo]]
fftb( afft2[nodo], defft2[nodo] )
-- *****
-- Procesos de distribucion para proceso autocorr
-- *****
PAR nodo = 0 FOR 2
PROCESSOR autocorr.p[nodo*2]
dist( acorr[nodo*2], acorr[(nodo*2)+1], adist[nodo],
      decorr[nodo*2], decorr[(nodo*2)+1], dedist[nodo] )
-- *****
-- Procesos de correlacion para proceso autocorr
-- *****
PAR nodo = 0 FOR 4
PROCESSOR autocorr.p[nodo]
corr( acorr[nodo], decorr[nodo] )
-- *****
-- Procesos FFT base 2 diazmado en frecuencia para proceso tfd
-- *****
PAR nodo = 0 FOR (N/2)-1
PROCESSOR tfd.p[p[nodo]]
fft2( afft3[(nodo*2)+1], afft3[(nodo*2)+2], afft3[nodo],
      defft3[(nodo*2)+1], defft3[(nodo*2)+2], defft3[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso tfd
-- *****
PAR nodo = (N/2)-1 FOR (N/2)
PROCESSOR tfd.p[p[nodo]]
fftb( afft3[nodo], defft3[nodo] )

```

A2.6.5 Cuatro procesadores con configuración tipo II: archivo main4.pgm

```

-- *****
-- * MCC, IIMAS, UNAM
-- * Ernesto Rubio Acosta
-- * 17 de septiembre de 1999
-- * Distribucion Tiempo Frecuencia
-- * Para 4 procesadores con configuracion
-- *****
VAL K IS 1024 :
VAL M IS K*K :
VAL [511]INT p IS [ 0, 0, 2, 0, 1, 2, 3, 0, 0, 1, 1, 2, 2, 3, 3,
0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

```

```

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3 1 :

```

```

-- *****
-- * Descripcion de hardware
-- *****

```

```

[4]NODE T:
ARC Hostlink :
NETWORK
DO
SET T[0] (type, memsize := "T805", 4*M )
SET T[1] (type, memsize := "T805", 4*M )
SET T[2] (type, memsize := "T805", 4*M )
SET T[3] (type, memsize := "T805", 4*M )
CONNECT T[0][link][0] TO HOST WITH Hostlink
CONNECT T[0][link][1] TO T[1][link][1]
CONNECT T[1][link][2] TO T[2][link][1]
CONNECT T[2][link][2] TO T[3][link][1]
CONNECT T[2][link][0] TO T[0][link][3]
:

```

```

-- *****
-- * Descripcion de mapeo
-- *****

```

```

NODE app.p :
[4]NODE hilbert1.p, hilbert2.p, autocorr.p, tfd.p :
MAPPING
DO
MAP app.p ONTO T[0]
DO i=0 FOR 4
MAP hilbert1.p[i] ONTO T[0]
DO i=0 FOR 4
MAP hilbert2.p[i] ONTO T[1]
DO i=0 FOR 2
MAP autocorr.p[i] ONTO T[2]
DO i=2 FOR 2
MAP autocorr.p[i] ONTO T[3]
DO i=0 FOR 4
MAP tfd.p[i] ONTO T[0]
:

```

```

-- *****
-- * Descripcion del software
-- *****

```

```

#include "hostio.inc"
#include "declarar.inc"
#USE "app.lku"
#USE "hilbert1.lku"
#USE "hilbert2.lku"
#USE "autocorr.lku"
#USE "dist.lku"
#USE "corr.lku"
#USE "tfd.lku"
#USE "fft2.lku"
#USE "fttb.lku"
CONFIG
CHAN OF SP fs, ts :
PLACE fs, ts ON Hostlink :
[(2*N)-1]CHAN OF DATOS afft1, defft1 :
[(2*N)-1]CHAN OF DATOS afft2, defft2 :
[N-1]CHAN OF DATOS afft3, defft3 :
[2]CHAN OF DATOS adist, dedist :
[4]CHAN OF DATOS acorr, decorr :
CHAN OF DATOS app.hilbert1, hilbert1.hilbert2, hilbert2.autocorr,
autocorr.tfd, tfd.app :
PAR
PAR
-- *****
-- Proceso Aplicacion
-- *****
PROCESSOR app.p
app( fs, ts, app.hilbert1, tfd.app )
-- *****
-- * Proceso Hilbert
-- *****
PROCESSOR hilbert1.p[0]
hilbert1( hilbert1.hilbert2, app.hilbert1, afft1[0], defft1[0] )
PROCESSOR hilbert2.p[0]
hilbert2( hilbert2.autocorr, hilbert1.hilbert2, afft2[0], defft2[0] )

```

```

-- *****
-- * Proceso Autocorrelacion
-- *****
PROCESSOR autocorr.p{0}
  autocorr( autocorr.tfd, hilbert2.autocorr, adist[0], dedist[0],
            adist[1], dedist[1] )
-- *****
-- * Proceso Distribucion Tiempo Frecuencia
-- *****
PROCESSOR tfd.p{0}
  tfd( tfd.app, autocorr.tfd, afft3[0], defft3[0] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert1
-- *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert1.p{p[nodo]}
    fft2( afft1[(nodo*2)+1], afft1[(nodo*2)+2], afft1[nodo],
          defft1[(nodo*2)+1], defft1[(nodo*2)+2], defft1[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert1
-- *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert1.p{p[nodo]}
    fftb( afft1[nodo], defft1[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso hilbert2
-- *****
PAR nodo = 0 FOR N-1
  PROCESSOR hilbert2.p{p[nodo]}
    fft2( afft2[(nodo*2)+1], afft2[(nodo*2)+2], afft2[nodo],
          defft2[(nodo*2)+1], defft2[(nodo*2)+2], defft2[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso hilbert2
-- *****
PAR nodo = N-1 FOR N
  PROCESSOR hilbert2.p{p[nodo]}
    fftb( afft2[nodo], defft2[nodo] )
-- *****
-- Procesos de distribucion para proceso autocorr
-- *****
PAR nodo = 0 FOR 2
  PROCESSOR autocorr.p{[nodo*2]}
    dist( acorr[nodo*2], acorr[(nodo*2)+1], adist[nodo],
          decorr[nodo*2], decorr[(nodo*2)+1], dedist[nodo] )
-- *****
-- Procesos de correlacion para proceso autocorr
-- *****
PAR nodo = 0 FOR 4
  PROCESSOR autocorr.p{[nodo]}
    corr( acorr[nodo], decorr[nodo] )
-- *****
-- Procesos FFT base 2 diezmado en frecuencia para proceso tfd
-- *****
PAR nodo = 0 FOR (N/2)-1
  PROCESSOR tfd.p{p[nodo]}
    fft2( afft3[(nodo*2)+1], afft3[(nodo*2)+2], afft3[nodo],
          defft3[(nodo*2)+1], defft3[(nodo*2)+2], defft3[nodo] )
-- *****
-- Procesos de retorno para fft2 para proceso tfd
-- *****
PAR nodo = (N/2)-1 FOR (N/2)
  PROCESSOR tfd.p{p[nodo]}
    fftb( afft3[nodo], defft3[nodo] )

```

A2.7 Diagramas de tiempo

En las figuras FIG.A2.1 a FIGA2.5 se muestran los diagramas de tiempo o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia.

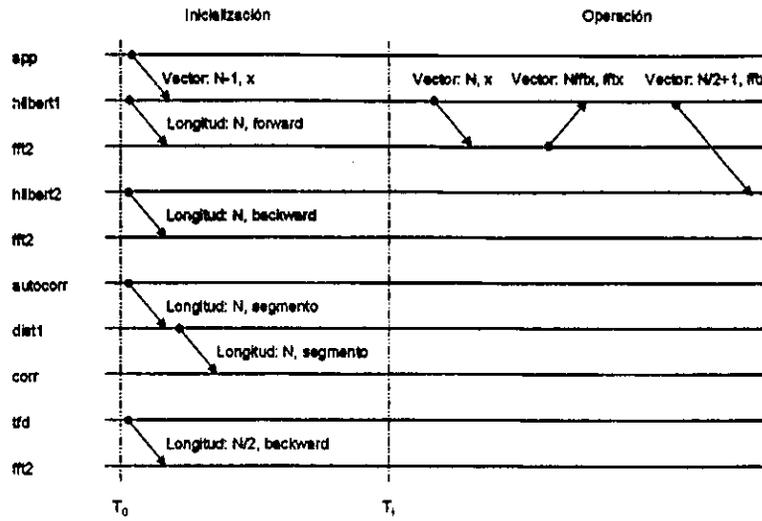


Figura FIG.A2.1: Diagrama de tiempos o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia.

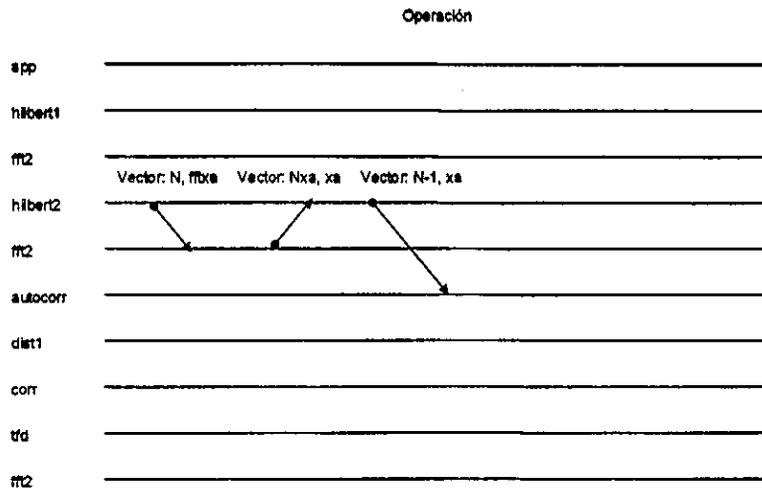


Figura FIG.A2.2: Diagrama de tiempos o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia (continuación).

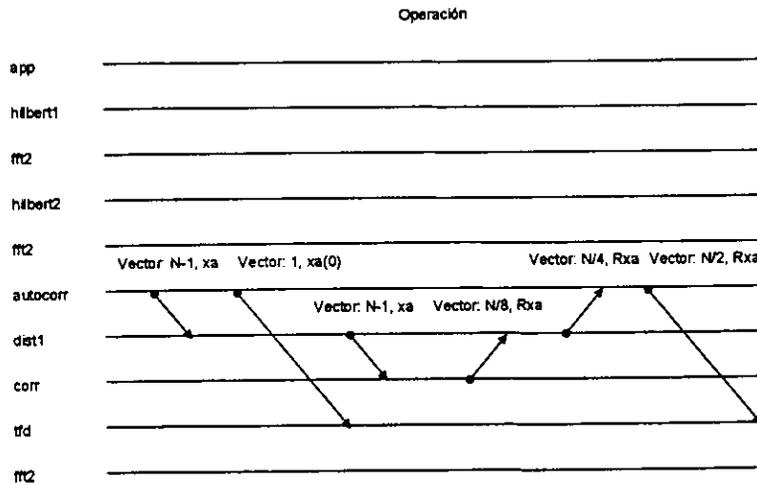


Figura FIG.A2.3: Diagrama de tiempos o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia (continuación).

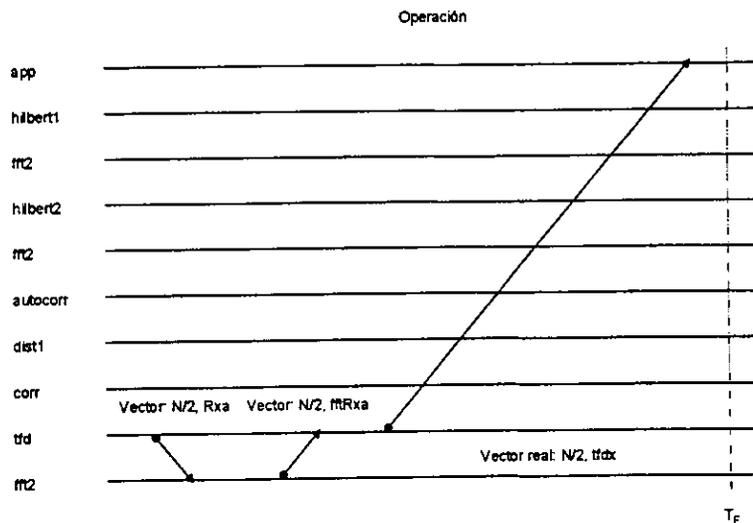


Figura FIG.A2.4: Diagrama de tiempos o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia (continuación).

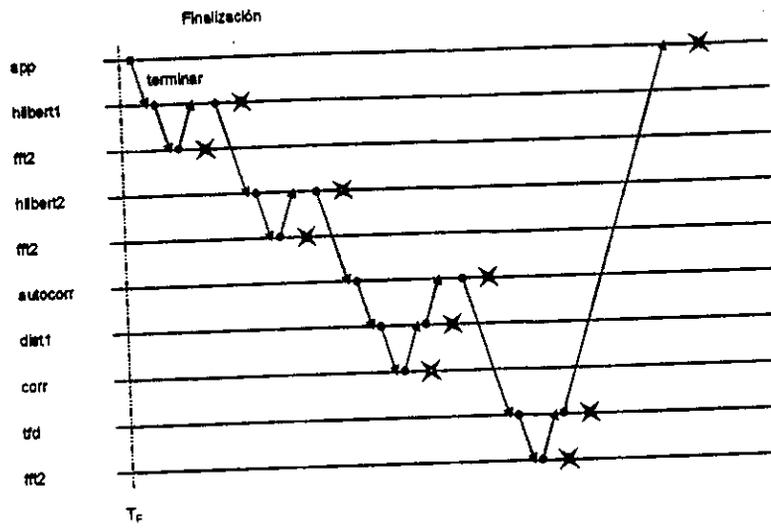


Figura FIG.A2.5: Diagrama de tiempos o de intercambio de mensajes de los procesos involucrados en el cálculo de las distintas distribuciones tiempo frecuencia (continuación).

A3 SEÑAL TIPO DÓPPLER

A3.1 Introducción

En este anexo se presenta la construcción de la señal tipo Dópler que se utiliza en la evaluación de los algoritmos secuenciales que calculan la frecuencia media pseudo instantánea con base en las distribuciones tiempo frecuencia de Wigner Ville, de Choi Williams y de Bessel. La bibliografía básica de este anexo consiste en [10] y [24].

A3.2 Señal Dópler Simulada

El desempeño de las diferentes distribuciones tiempo frecuencia se evaluará con una señal Dópler simulada con características conocidas^{[10][24]}.

La señal Dópler no estacionaria se puede escribir como:

$$x_{Doppler}(t) = A(t)e^{j\phi_a(t)}e^{j\phi_d(t)} \quad (\text{EQ.A3.1})$$

donde $A(t)e^{j\phi_a(t)}$ es una función aleatoria de banda base y $e^{j\phi_d(t)}$ es una función de corrimiento en frecuencia con una frecuencia instantánea igual a la frecuencia media $f_i(t)$ del espectro de la señal $x_{Doppler}(t)$.

La función aleatoria de banda base de la señal $x_{Doppler}(t)$ se puede escribir como:

$$A(t)e^{j\phi_a(t)} = F^{-1}[G(f)N(f)] \quad (\text{EQ.A3.2})$$

donde $F^{-1}[\]$ es el operador transformada inversa de Fourier, $N(f)$ es una variable aleatoria compleja con una densidad de probabilidad normal y módulo unitario, y $G(f)$ es una función de filtrado real con valor medio en cero y con la forma de la envolvente de la amplitud del espectro de la señal $x_{Doppler}(t)$.

En este trabajo $G(f)$ es una envolvente Guassiana dada por:

$$G(f) = \sqrt{e^{-0.5\left(\frac{f}{b}\right)^2}} \quad (\text{EQ.A3.3})$$

de modo que el ancho de banda RMS del espectro de potencia $G^2(f)$ es b . En particular, en este trabajo, se elige un ancho de banda RMS constante de 100Hz. En la figura FIG.A3.1 se muestra la gráfica de $G(f)$ para $b = 100$.

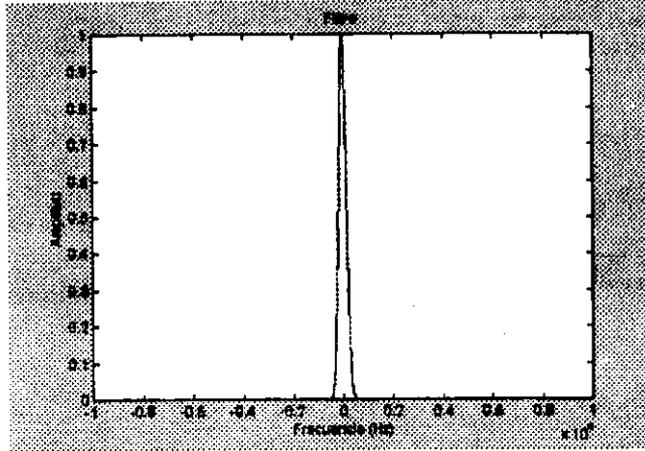


FIG.A3.1: Gráfica de la función $G(f)$ para $b=100$, envoltura de la amplitud del espectro de la señal Dópler.

La fase determinística $\phi_d(t)$ de la función de corrimiento en frecuencia de la señal $x_{Doppler}(t)$ se puede escribir como:

$$\phi_d(t) = 2\pi \int_0^t f_i(t) dt \tag{EQ.A3.4}$$

donde $f_i(t)$ es la frecuencia media del espectro de la señal $x_{Doppler}(t)$. En la figura FIG.A3.2 se muestra la gráfica de $f_i(t)$ y en la figura FIG.A3.3 la de $\phi_d(t)$.

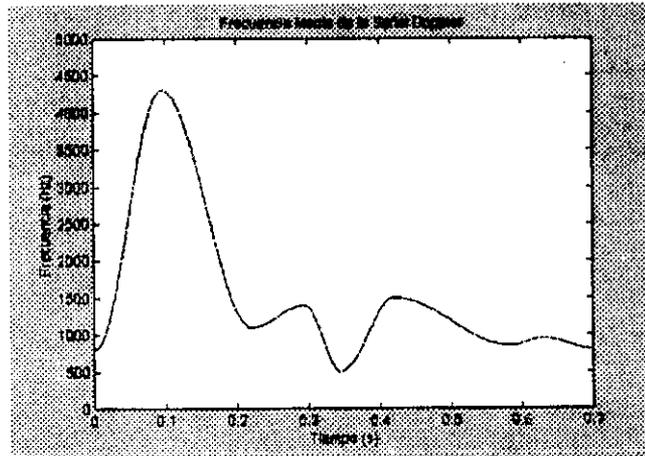


FIG.A3.2: Gráfica de la función $f_i(t)$, frecuencia media del espectro de la señal Dópler.

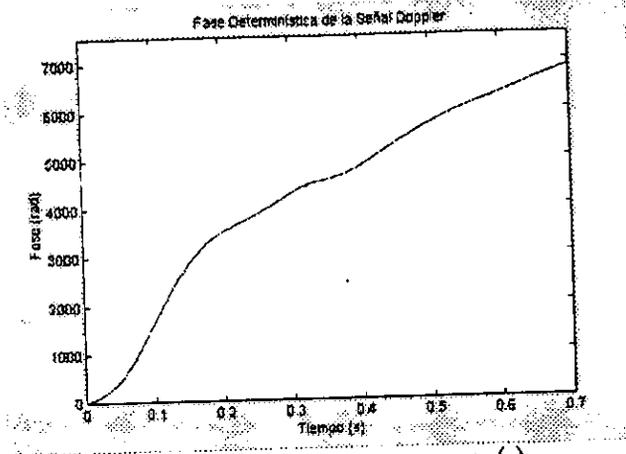


FIG.A3.3: Gráfica de la función $\phi_d(t)$, fase determinística de la señal Dópler.

Finalmente, en este trabajo, la señal Dópler simulada que se obtiene se muestrea a 19.2 kHz.

A4 OTRAS EXPRESIONES PARA EXPERIMENTOS NUMÉRICOS

A4.1 Introducción

En este anexo se plantean expresiones útiles para los experimentos numéricos en Matlab, como son la que calcula el ancho de banda espectral RMS, la relación de potencia entre la señal y el ruido (SNR) y los errores en las diferentes estimaciones.

A4.2 Ancho de banda espectral RMS

El ancho de banda espectral RMS $\delta(n)$ se define como^[10]:

$$\delta(n) = \sqrt{\frac{\sum_{k=0}^{N-1} (PIMF(n) - f_k)^2 \cdot PIPD(n, k)}{\sum_{k=0}^{N-1} PIPD(n, k)}} \quad (\text{EQ.A4.1})$$

donde $PIMF(n)$ es la frecuencia media pseudo instantánea discreta (sin normalizar), f_k es la frecuencia discreta correspondiente al punto k (sin normalizar) y $PIPD(n, k)$ es la distribución pseudo instantánea de potencia discreta.

A4.3 Relación de potencia entre la señal y el ruido SNR

La relación de potencia entre la señal y el ruido SNR se define como^[11]:

$$SNR = \frac{\frac{1}{L} \sum_{i=0}^{L-1} x^2(i)}{\frac{1}{L} \sum_{i=0}^{L-1} noise^2(i)} \quad (\text{EQ.A4.2})$$

donde $x(n)$ es la señal y $noise(n)$ es una señal de ruido distribuido normalmente.

A4.4 Error en la estimación de la frecuencia media pseudo instantánea

El error en la estimación de la frecuencia media pseudo instantánea^[10] se calcula con la siguiente expresión:

$$error_{rms} = \sqrt{bias^2 + varianza} \quad (\text{EQ.A4.3})$$

donde $bias$ y $varianza$ constituyen el promedio y la varianza estadística respectivamente del error en la estimación de la frecuencia media pseudo instantánea con respecto a un conjunto de muestras. El $bias$ se calcula con la siguiente expresión:

$$bias = \frac{1}{m} \sum_{n=0}^{m-1} (PIMF(n) - f_{i/real}(n)) \quad (\text{EQ.A4.4})$$

donde m es el número de muestras, $PIMF(n)$ es la frecuencia media pseudo instantánea calculada con (EQ.2.21) y $f_{i/real}(n)$ es la frecuencia instantánea verdadera (ambas sin normalizar). Mientras que la $varianza$ se calcula con la siguiente expresión:

$$\text{varianza} = \frac{1}{m} \sum_{n=0}^{m-1} (\text{PIMF}(n) - f_{\text{real}}(n))^2 \quad (\text{EQ.A4.5})$$

A4.5 Error en la estimación del ancho de banda espectral RMS

El error en la estimación del ancho de banda espectral RMS^[10] se calcula con la siguiente expresión:

$$\text{error}_{\text{rms}} = \sqrt{\text{bias}^2 + \text{varianza}} \quad (\text{EQ.A4.6})$$

donde *bias* y *varianza* constituyen el promedio y la varianza estadística respectivamente del error en la estimación del ancho de banda espectral RMS con respecto a un conjunto de muestras. El *bias* se calcula con la siguiente expresión:

$$\text{bias} = \frac{1}{m} \sum_{n=0}^{m-1} (\hat{b}(n) - b_{\text{real}}(n)) \quad (\text{EQ.A4.7})$$

donde *m* es el número de muestras, $\hat{b}(n)$ es el ancho de banda espectral RMS calculado con (EQ.A4.1) y $b_{\text{real}}(n)$ es el ancho de banda espectral RMS. Mientras que la *varianza* se calcula con la siguiente expresión:

$$\text{varianza} = \frac{1}{m} \sum_{n=0}^{m-1} (\hat{b}(n) - b_{\text{real}}(n))^2 \quad (\text{EQ.A4.8})$$

A5 EXPRESIONES MATEMÁTICAS ÚTILES

A5.1 Introducción

En este anexo se detallan secuencias relevantes, es decir, expresiones matemáticas útiles, de los desarrollos matemáticos que se efectúan en los capítulos 4, 5 y 6, Distribución de Wigner Ville, Distribución de Choi Williams y Distribución de Bessel, respectivamente.

A5.2 Distribución de Wigner Ville

$$(A.4.1) \text{ Transformada de Fourier: } 1 \xrightarrow{F} 2\pi\delta(\omega) \quad [45]$$

$$(A.4.2) \text{ Simetría par: } \delta(t) = \delta(-t)$$

$$\text{Integral con función delta: } \int_{-\infty}^{\infty} \delta(t-t_0)\phi(t)dt = \phi(t_0) \quad [45]$$

$$(A.4.3) \text{ Transformada de Fourier: } e^{j\omega_0 t} \xrightarrow{F} 2\pi\delta(\omega - \omega_0) \quad [45]$$

$$(A.4.4) \text{ Transformada de Fourier: } e^{j\omega_0 t} \xrightarrow{F} 2\pi\delta(\omega - \omega_0) \quad [45]$$

$$(A.4.5) \text{ Si } Z \text{ es un número complejo, entonces } Z + Z^* = 2\text{Re}[Z]$$

A5.3 Distribución Choi Williams

$$(A.5.1) \text{ Transformada de Fourier: } e^{-at^2} \xrightarrow{F} \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}} \quad [45]$$

$$(A.5.2) \text{ Cambio de variable: } y = \frac{t-\mu}{2\tau/\sqrt{\sigma}}; \text{ límites de integración } \infty \text{ a } -\infty.$$

$$\text{Integral de función Gaussiana: } \int_{-\infty}^{\infty} e^{-y^2} dy = \sqrt{\pi} \quad [45]$$

$$(A.5.3) \text{ Transformada de Fourier: } e^{j\omega_0 t} \xrightarrow{F} 2\pi\delta(\omega - \omega_0) \quad [45]$$

$$(A.5.4) \text{ Ecuación de Parseval: si } F[f(t)] = F(\omega) \text{ y } F[g(t)] = G(\omega), \text{ entonces}$$

$$\int_{-\infty}^{\infty} f(x)G(x)dx = \int_{-\infty}^{\infty} F(x)g(x)dx \quad [45]$$

Para la función $f(x)$:

$$\text{Si } F[f(t)] = F(\omega), \text{ entonces } F(-t+t_0) \xrightarrow{F} 2\pi e^{-j\omega t_0} f(\omega) \quad [45]$$

$$\text{Transformada de Fourier: } e^{-at^2} \xrightarrow{F} \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}} \quad [45]$$

Para la función $g(x)$:

$$\text{Transformada de Fourier: } \delta(t-t_0) \xrightarrow{F} e^{-j\omega t_0} \quad [45]$$

$$(A.5.5) \text{ Integral con función delta: } \int_{-\infty}^{\infty} \delta(t-t_0)\phi(t)dt = \phi(t_0) \quad [45]$$

$$(A.5.6) \text{ Si } F[f(t)] = F(\omega), \text{ entonces } f(t)e^{j\omega_0 t} \xrightarrow{F} F(\omega - \omega_0) \quad [45]$$

Transformada de Fourier: $e^{-a^2 t^2} \xrightarrow{F} \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}}$ [45]

(A.5.7) Si Z es un número complejo, entonces $Z + Z^* = 2\text{Re}[Z]$

A5.4 Distribución de Bessel

(A.6.1) Transformada de Fourier: $t^{-1} J_n(t) \xrightarrow{F} \begin{cases} \frac{2i}{n} (-i)^n \sqrt{1-\omega^2} U_{n-1}(\omega) & \omega^2 < 1 \text{ [46]} \\ 0 & \omega^2 > 1 \end{cases}$ donde $U_{n-1}(\omega)$

es el polinomio de Chebyshev de segunda clase: $U_n(z) = \frac{\text{sen}((n+1)\arccos(z))}{\sqrt{1-z^2}}$ [47]

Si $F[f(t)] = F(\omega)$, entonces $f(at) \xrightarrow{F} \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$ [45]

(A.6.2) Cambio de variable: $y = \frac{t-\mu}{\alpha\tau}$; límites de integración de 1 a -1

Cambio de variable: $\text{sen } \theta = y$; límites de integración de $\pi/2$ a $-\pi/2$

(A.6.3) Transformada de Fourier: $e^{j\omega_0 t} \xrightarrow{F} 2\pi\delta(\omega - \omega_0)$ [45]

(A.6.4) Ecuación de Parseval: si $F[f(t)] = F(\omega)$ y $F[g(t)] = G(\omega)$, entonces

$$\int_{-\infty}^{\infty} f(x)G(x)dx = \int_{-\infty}^{\infty} F(x)g(x)dx \text{ [45]}$$

Para la función $f(x)$:

Si $F[f(t)] = F(\omega)$, entonces $F(-t + t_0) \xrightarrow{F} 2\pi e^{-j\omega t_0} f(\omega)$ [45]

Transformada de Fourier: $t^{-1} J_1(t) \xrightarrow{F} \begin{cases} 2\sqrt{1-\omega^2} U_0(\omega) & \omega^2 < 1 \text{ donde } U_0(\omega) = 1 \text{ [46][47]} \\ 0 & \text{otro} \end{cases}$

Para la función $g(x)$:

Transformada de Fourier: $\delta(t - t_0) \xrightarrow{F} e^{-j\omega t_0}$ [45]

(A.6.5) Integral con función delta: $\int_{-\infty}^{\infty} \delta(t - t_0) \phi(t) dt = \phi(t_0)$ [45]

(A.6.6) Si $F[f(t)] = F(\omega)$, entonces $f(t)e^{j\omega_0 t} \xrightarrow{F} F(\omega - \omega_0)$ [45]

Transformada de Fourier: $t^{-1} J_n(t) \xrightarrow{F} \begin{cases} \frac{2i}{n} (-i)^n \sqrt{1-\omega^2} U_{n-1}(\omega) & \omega^2 < 1 \text{ [46]} \\ 0 & \omega^2 > 1 \end{cases}$ donde $U_{n-1}(\omega)$

es el polinomio de Chebyshev de segunda clase: $U_n(z) = \frac{\text{sen}((n+1)\arccos(z))}{\sqrt{1-z^2}}$ [47]

(A.6.7) Si Z es un número complejo, entonces $Z + Z^* = 2\text{Re}[Z]$

BIBLIOGRAFÍA

- [1] L. Cohen. "Time-Frequency Distributions-A Review". Proceedings of the IEEE. Vol. 77. No. 7. P.P. 941-981. July 1989.
- [2] L. Cohen. "Time-Frequency Analysis". New Jersey. Prentice-Hall PTR. 1995.
- [3] J. Wood, A. Buda, D. Barry. "Time - Frequency Transforms: A New Approach to first Heart Sound Frequency Dynamics". IEEE Transactions on Biomedical Engineering. Vol. 39. No. 7. P.P. 730-740. July 1992.
- [4] J. Jeong. W. Williams. "Kernel Design for Reduced Interference Distributions". IEEE Transactions on Signal Processing. Vol. 40. No. 2. P.P. 402-412. February 1992.
- [5] J. Jeong. W. Williams. "Alias-Free Generalized Discrete-Time Time-Frequency Distributions". IEEE Transactions on Signal Processing. Vol. 40. No. 11. P.P. 2757-2765. November 1992.
- [6] B. Boashash. "Estimating and Interpreting the Instantaneous Frequency of a Signal- Part 1: Fundamentals". Proceedings of the IEEE. Vol. 80. No. 4. P.P. 520-538. April 1992.
- [7] B. Boashash. "Estimating and Interpreting the Instantaneous Frequency of a Signal- Part 2: Algorithms and Applications". Proceedings of the IEEE. Vol. 80. No. 4. P.P. 540-568. April 1992.
- [8] W. Martin, P. Flandrin. "Wigner-Ville Spectral Analysis of Nonstationary Processes". IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP-33. No. 6. P.P. 1461-1470. December 1985.
- [9] B. Boashash, P. Black. "An Efficient Real-Time Implementation of the Wigner-Ville Distribution". IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP-35. No. 11. P.P. 1611-1618. November 1987.
- [10] J. Cardoso, G. Ruano, P. Fish. "Nonstationary Broadening Reduction in Pulsed Doppler Spectrum Measurements Using Time-Frequency Estimators". IEEE Transactions on Biomedical Engineering. Vol. 43. No. 12. P.P. 1176-1186. December 1996.
- [11] L. Fan, D. Evans. "Extracting Instantaneous Mean Frequency Information from Doppler Signals Using the Wigner Distribution Function". Ultrasound in Med. & Biol. Vol. 20. No. 5. P.P. 429-443. 1994.
- [12] L. Fan, D. Evans. "A Real-Time and Fine Resolution Analyzer Used to Estimate the Instantaneous Energy Distribution of Doppler Signals". Ultrasound in Med. & Biol. Vol. 20. No. 5. P.P. 445-454. 1994.
- [13] A. Papoulis. "Signal Analysis". Singapore. International Student Edition. McGraw Hill. 1985.
- [14] A. Papoulis. "Probability, Random Variables, and Stochastic Processes. Singapore. International Student Edition. McGraw Hill, 1984.
- [15] J. Proakis, D. Manolakis. "Tratamiento digital de señales. Principios, algoritmos y aplicaciones". España. Prentice Hall. 1998.
- [16] H. Choi, W. Williams. "Improved Time-Frequency Representation of Multicomponent Signals Using Exponential Kernels". IEEE Transactions on Acoustics, Speech and Signal Processing. Vol. 37. No. 6. P.P. 862-871. June 1989.
- [17] J. Cardoso, P. Fish, G. Ruano. "Choi-Williams Time-Frequency Estimator: A Parallel Implementation". Preprint.

- [18] Z. Guo, L. Durand, H. Lee. "The Time-Frequency Distributions of Nonstationary Signals Based on a Bessel Kernel". *IEEE Transactions on Signal Processing*. Vol. 42. No. 7. P.P. 1700-1707. July 1994.
- [19] Z. Guo, L. Durand, H. Lee. "Comparison of Time-Frequency Distribution Techniques for Analysis of Simulated Doppler Ultrasound Signals of the Femoral Artery". *IEEE Transactions on Biomedical Engineering*. Vol. 41. No. 4. P.P.332-342. April 1994.
- [20] P. Loughlin, J. Pitton, L. Atlas. "Bilinear Time-Frequency Representations: New Insights and Properties". *IEEE Transactions on Signal Processing*. Vol. 41. No. 2. P.P. 750-766. February 1993.
- [21] G. Ruano, P. Fish. "Cost/Benefit Criterion for Selection of Pulsed Doppler Ultrasound Spectral Mean Frequency and Bandwidth Estimators". *IEEE Transactions on Biomedical Engineering*. Vol. 40. No. 12. P.P. 1338-1341. December 1993.
- [22] D. Lampard. "Generalization of the Wiener-Khintchine Theorem to Nonstationary Processes". *Journal of Applied Physics*. Vol. 25. No. 6. P.P. 802-803. June. 1954.
- [23] J. Andrieux, M. Feix, G. Mourgues, P. Bertrand, B. Izrar, T. Nguyen. "Optimum Smoothing of the Wigner-Ville Distribution". *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Vol. ASSP-35. No.6, P.P. 764-768. June 1987.
- [24] P. Fish. "Physics and Instrumentation of Diagnostic Medical Ultrasound". Reino Unido. John Wiley & Sons. 1990.
- [25] B. Boashash. "Use of the Cross Wigner-Ville Distribution for Estimation of Instantaneous Frequency". *IEEE Transactions on Signal Processing*. Vol. 41. No. 3. P.P. 1439-1445. March 1993.
- [26] L. Cohen. "On a Fundamental Property of the Wigner Distribution". *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Vol. ASSP-35. No. 4. P.P. 559-561. April 1987.
- [27] A. Zeira, E. Zeira, S. Holland. "Pseudo-Wigner Distribution for Analysis of Pulsed Doppler Ultrasound". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. Vol. 41. No. 3. P.P. 346-352. May 1994.
- [28] A. Karp, H. Flatt. "Measuring Parallel Processor Performance". *Communications of the ACM*. Vol. 33. No. 5. P.P. 539-543. May 1990.
- [29] P. Flatt, K. Kennedy. "Performance of parallel processors". *Parallel Computing*. North Holland. No. 12. P.P. 1-20. 1989.
- [30] R. Creutzburg, T. Minkwitz, M. Roggenbach, T. Umland. "Parallel FFT-like Transform Algorithms on Transputers" en I. Pitas (editor) "Parallel Algorithms for Digital Image Processing, Computer Vision and Neuronal Networks". John Wiley & Sons Ltd. P.P. 53-89. 1993.
- [31] V. Kumar, A. Grama, A. Gupta, G. Karypis. "Introduction to Parallel Computing: Design and Analysis of Parallel Algorithms". California. Benjamin/Cummings Publishing Company Inc. 1994.
- [32] C. Thompson, L. Shure, J. Little. "Signal Processing Toolbox, For Use with Matlab". The Math Works Inc. 1996.
- [33] "Matlab, User's Guide". The Math Works Inc. 1996.
- [34] "Occam 2 Toolset User Guide" (72 TDS 366 01). Inmos. 1993.
- [35] "Occam 2 Toolset Reference Manual" (). Inmos. 1993.
- [36] "Occam 2 Toolset Language and Libraries Reference Manual" (72 TDS 368 01). Inmos. 1993.

- [37] G. Jones, M. Goldsmith. "Programming in Occam 2". Prentice Hall International Series in Computer Science. Reino Unido. 1988.
- [38] J. Galletly. "Occam 2". Pitman Publishing. 1990.
- [39] "S708 User Guide" (72 OEK 227 01). Inmos. 1990.
- [40] "IMS B008 User Guide and Reference Manual" (72 TRN 223 00). Inmos. 1990.
- [41] H. Roebbers, P. Welch, K. Wijbrans. "A generalized FFT algorithm on Transputers". Preprint.
- [42] D. Evans, S. Mai. "A parallel algorithm for the Fast Fourier Transform". Parallel Algorithms & Architectures. Elsevier Science Publishers B.V. (North-Holland). P.P. 47-60. 1986.
- [43] U. Thoeni. "Programming Real-Time Multicomputers for Signal Processing". Reino Unido. Prentice-Hall International Series in Acoustics, Speech and Signal Processing. 1994.
- [44] I. Bouras, G. Glentis, N. Kalouptsidis. "Efficient Computation of the Fourier Transform on Star Graph" en S. Tzafestas, P. Borne, L. Grandinetti (Editores) "Parallel and Distributed Computing in Engineering Systems" Proceedings of the IMACS/IFAC 1991. Holanda. North-Holland. P.P.233-238. 1992.
- [45] H. Hsu. "Fourier Analysis". Simon-Schuster. New York, 1970.
- [46] M. Abramowitz, I. Stegun. "Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables". National Bureau of Standards Applied Mathematics Series 55. USA. 1970.
- [47] A. Prudnikov, Y. Brychkov, O. Marichev. "Integrals and Series. Volume 2: Special Functions". Gordon and Breach Science Publishers. London. 1988.
- [48] S. Kay. "Modern Spectral Estimation. Theory & Application". New Jersey. Prentice Hall. 1988.
- [49] J. Proakis, C. Rader, F. Ling, C. Nikias. "Advanced Digital Signal Processing". New York. Macmillan Publishing Company. 1992.
- [50] N. Hess-Nielsen, M. Wickerhauser. "Wavelets and Time-Frequency Analysis". Proceedings of the IEEE. Vol. 84. No. 4. P.P. 523-540. April. 1996.
- [51] G. Wornell. "Signal Processing with Fractals: A Wavelet Based Approach". Prentice Hall. 1995.
- [52] S. Kay, S. Lawrence. "Spectrum Analysis - A Modern Perspective". Proceedings of the IEEE. Vol. 69. No. 11. P.P. 1380-1419. November 1981.
- [53] E. Robinson. "A Historical Perspective of Spectrum Estimation". Proceedings of the IEEE. Vol. 70. No. 9. P.P. 885-907. September 1982.
- [54] B. Sigel. "A Brief History of Doppler Ultrasound in the Diagnosis of Peripheral Vascular Disease". Ultrasound in Med. & Biol. Vol. 24. No. 2. P.P. 169-176. 1998.
- [55] P. Vaitkus, R. Cobbold. "A Comparative Study and Assessment of Doppler Ultrasound Spectral Estimation Techniques. Part I: Estimation Methods". Ultrasound in Med. & Biol. Vol. 14. No. 8. P.P. 661-672. 1988.
- [56] P. Vaitkus, R. Cobbold. "A Comparative Study and Assessment of Doppler Ultrasound Spectral Estimation Techniques. Part II: Methods and Results". Ultrasound in Med. & Biol. Vol. 14. No. 8. P.P. 673-688. 1988.
- [57] J. David, S. Jones. "Modern Spectral Analysis Techniques for Blood Flow Velocity and Spectral Measurements with Pulsed Doppler Ultrasound". IEEE Transactions on Biomedical Engineering. Vol. 38. No. 6. P.P. 589-596. June 1991.

- [58] P. Fish. "Physics and Instrumentation of Diagnostic Medical Ultrasound". Reino Unido. John Wiley & Sons. 2000.
- [59] I. Sánchez, Dir. F. García. Tesis de Licenciatura. Ingeniería en Electrónica. DISCA, IIMAS, UNAM.