



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**ELABORACIÓN DE UN LABORATORIO VIRTUAL PARA MEJORAR EL
APRENDIZAJE DE FILTROS DIGITALES EN DSP EN LA FACULTAD
TÉCNICA PARA EL DESARROLLO DE LA UNIVERSIDAD CATÓLICA
SANTIAGO DE GUAYAQUIL**

Previa la obtención del Título

INGENIERO EN TELECOMUNICACIONES

ELABORADO POR:

Jorge Luis Ojeda Guadamud

Guayaquil, 20 de Febrero del 2014



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr.
Jorge Luis Ojeda Guadamud como requerimiento parcial para la obtención del
título de INGENIERO EN TELECOMUNICACIONES.

Guayaquil, 20 de Febrero del 2014

DIRECTOR

Ing. Efraín Suarez Murillo

REVISADO POR

Ing. Carlos Romero Rosero.

Revisor

Ing. Juan González Bazán.

Revisor



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

JORGE LUIS OJEDA GUADAMUD

DECLARÓ QUE:

El proyecto de tesis denominado “Elaboración de un laboratorio virtual para mejorar el aprendizaje de filtros digitales en DSP en la Facultad Técnica para el Desarrollo de la Universidad Católica Santiago de Guayaquil” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Guayaquil, 20 de Febrero del 2014

EL AUTOR

JORGE LUIS OJEDA GUADAMUD



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, JORGE LUIS OJEDA GUADAMUD

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del proyecto titulado: “Elaboración de un laboratorio virtual para mejorar el aprendizaje de filtros digitales en DSP en la Facultad Técnica para el Desarrollo de la Universidad Católica Santiago de Guayaquil”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y autoría.

Guayaquil, 20 de Febrero del 2014

EL AUTOR

JORGE LUIS OJEDA GUADAMUD

AGRADECIMIENTO

Deseo agradecer profundamente a Dios por la vida y por la oportunidad de alcanzar las metas trazadas en ella. Por ser el pilar fundamental en mi existencia y por permitirme vivir este momento de suma trascendencia para mí y para mi familia.

A la Universidad Católica de Santiago de Guayaquil y a cada uno de los maestros que durante estos años me han impartido los conocimientos y experiencias que con seguridad me ayudarán a emprender mi inmediata carrera profesional.

JORGE LUIS OJEDA GUADAMUD

DEDICATORIA

De todo corazón dedico este trabajo a mi familia y especialmente a mis padres, quienes me han brindado todo su apoyo de manera incondicional especialmente en aquellos momentos de dificultad.

A las nuevas generaciones que oportunamente le darán el uso correspondiente para su beneficio y el de nuestra sociedad.

JORGE LUIS OJEDA GUADAMUD

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	10
ÍNDICE DE TABLAS	12
RESUMEN	13
CAPÍTULO I: DESCRIPCIÓN GENERAL DEL TRABAJO DE TITULACIÓN.....	15
1.1. INTRODUCCIÓN	15
1.2. JUSTIFICACION:.....	15
1.3. ANTECEDENTES.....	16
1.4. DEFINICIÓN DEL PROBLEMA:.....	17
1.5. DELIMITACION DEL PROBLEMA:.....	18
1.6. OBJETIVOS:.....	18
1.6.1. Objetivo General:.....	18
1.6.2. Objetivos Específicos:	18
CAPITULO II: FUNDAMENTACIÓN TEÓRICA DE DSP'S.....	19
2.1. Fundamentación Legal.....	19
2.2. Gráficas de inclusión de las categorías fundamentales.....	19
2.3. Categorías Fundamentales.....	19
2.3.1. Electrónica.....	19
2.3.2. Electrónica Digital	20
2.3.3. DSP	21
2.3.4. Filtros Digitales	26
2.3.4.1. Tipos de filtros	27
2.4. Por su respuesta impulsional:	31
2.4.1. Filtros FIR (respuesta impulsional finita).....	31
2.4.2. Filtros IIR (respuesta impulsional infinita)	33

2.5.	FIR vs IIR	34
2.6.	Diseño de Filtros Digitales.....	35
2.6.1.	Especificaciones de diseño.....	35
2.6.2.	Elección de la aproximación y cálculo de los coeficientes	36
2.6.3.	Diseño de Filtros Digitales FIR con la técnica de ventanas.....	37
2.6.4.	Diseño con ventana de Kaiser.	39
2.6.5.	Diseño de Filtros IIR	40
2.6.6.	Diseño de filtros IIR mediante transformadas bilineales.....	41
2.7.	Pasos para el diseño de filtros utilizando la transformada bilineal	42
2.8.	Laboratorio Virtual.....	43
2.9.	Software.....	46
2.9.1.	MatLab	46
2.9.2.	Simulink.....	47
CAPÍTULO III: FILTROS DIGITALES.....		52
3.1.	Introducción.....	52
3.2.	Filtros Digitales FIR.....	53
3.3.	Diseño De Filtros FIR.....	54
3.3.1.	Método de las ventanas.....	54
3.3.2.	Pasos para el diseño de filtros FIR por el método de la ventana	55
3.3.3.	Funciones en MatLab para Filtros FIR.....	61
3.4.	Filtros IIR.....	61
3.4.1.	Diseño De Filtros IIR Mediante Transformadas Bilineales	62
3.4.2.	Pasos para el diseño de filtros IIR mediante transformadas bilineales:	
	62	
CAPÍTULO IV: DESARROLLO EN MATLAB DE FILTROS DIGITALES		68
4.1.	Diseño de filtros usando MATLAB.....	68

4.2.	Desarrollo de las Prácticas de Filtros FIR.....	70
4.2.1.	Experiencia #1: Filtro FIR Pasa-Bajo.....	75
4.2.2.	Experiencia #2: Filtro FIR Pasa-Alto.....	76
4.2.3.	Experiencia #3: Filtro FIR Pasa-Banda 1.....	77
4.2.4.	Experiencia #4: Filtro FIR Rechaza Banda.....	78
4.2.5.	Conclusiones del diseño de filtros FIR.....	79
4.3.	Desarrollo de las Prácticas de Filtros IIR.....	80
4.3.1.	Experiencia #1: Filtro IIR Pasa Bajo – Butterworth.....	87
4.3.2.	Experiencia #2: Filtro IIR Pasa Alto – Chevyshev 1.....	88
4.3.3.	Experiencia #3: Filtro IIR Pasa Banda – Chevyshev 2.....	89
4.3.4.	Experiencia #4: Filtro IIR Paso Bajo – Elíptica.....	90
4.3.5.	Conclusiones del diseño de filtros IIR.....	92
	CONCLUSIONES:.....	93
	RECOMENDACIONES:.....	94

ÍNDICE DE FIGURAS

Capítulo 2:

Figura 2. 1: Categorías Fundamentales.....	19
Figura 2. 2: Aplicaciones del Procesamiento Digital de Señales.	21
Figura 2. 3: Sistema básico de compresión de pulsos de radar.....	23
Figura 2. 4: Cancelamiento de eco en redes telefónicas de larga distancia.	24
Figura 2. 5: Diagrama de bloques de un sistema de reconocimiento de voz	25
Figura 2. 6: Diagrama esquemático de un Filtro Digital	27
Figura 2. 7: Características de filtro pasa bajo.....	28
Figura 2. 8: Características de un filtro pasa alto.....	29
Figura 2. 9: Características de filtro pasa banda.....	30
Figura 2. 10: Características de filtro rechaza banda.....	31
Figura 2. 11: Diagrama de bloques de la estructura en forma directa de un filtro FIR	32
Figura 2. 12: Diagrama de bloques de la estructura en forma directa tipo 1 de un filtro IIR.....	33
Figura 2. 13: Transformación Bilineal.....	42
Figura 2. 14: Esquema de representación de un laboratorio virtual	43
Figura 2. 14: Librerías de Simulink	51

Capítulo 3:

Figura 3. 1: Condiciones de una ventana de truncamiento.....	55
Figura 3. 2: Especificaciones de la respuesta espectral de un filtro.....	56
Figura 3. 3: Respuesta de magnitud cuadrática.....	63
Figura 3. 4: Transformación bilineal.....	64
Figura 3. 5: Respuesta en frecuencia de un filtro Butterworth	65
Figura 3. 6: Respuesta en frecuencia de un filtro Chevyshev I	65
Figura 3. 7: Respuesta en frecuencia de un filtro Chevyshev II.	66
Figura 3. 8: Respuesta en frecuencia de un filtro Elíptico.	66

Capítulo 4:

Figura 4. 1: Interfaz para el diseño de Filtros FIR	70
Figura 4. 2: Panel de especificaciones para el diseño de Filtros FIR pasa bajo. ..	75
Figura 4. 3: Gráficos de Respuestas Espectrales para Filtros FIR Pasa Bajo.....	76
Figura 4. 4: Gráficos de Respuestas Espectrales para Filtros FIR Pasa Alto.....	77
Figura 4. 5: Gráficos de Respuestas Espectrales para Filtro FIR Pasa Bajo.	78
Figura 4. 6: Gráficos de la Respuesta del filtro de la Práctica 4.....	79
Figura 4. 7: Interfaz para el diseño de Filtros IIR	80
Figura 4. 8: Paneles del Diseño de Filtros IIR.....	87
Figura 4. 9: Gráficos de las respuestas de Filtros IIR Pasa Bajo – Butterworth. ..	88
Figura 4. 10: Gráficos de las respuestas del Filtro IIR Pasa Alto – Chevyshev 1.	89

ÍNDICE DE TABLAS

Capítulo 2.

Tabla 2. 1: Métodos típicos para el diseño de filtros IIR y FIR	36
Tabla 2. 2: Ecuación de diseño para diferentes tipos de Filtros FIR	38
Tabla 2. 3: Tipos de Ventanas para diseño de Filtros FIR	39
Tabla 2. 4: Orden de Filtros FIR según la ventana utilizada	39
Tabla 2. 4: Principales ventajas y desventajas de cada tipo de laboratorio	46

Capítulo 3.

Tabla 3. 1: Funciones más comunes para el diseño de filtros digitales en Matlab	52
Tabla 3. 2: Especificaciones de la respuesta espectral de un filtro	56
Tabla 3. 3: Especificaciones de la respuesta espectral de un filtro real derivadas de las anteriores	56
Tabla 3. 4: Ventanas utilizadas en las operaciones de ventaneo	57
Tabla 3. 5: Espectros de las ventanas utilizadas en las operaciones de ventaneo	58
Tabla 3. 6: Estimación de parámetros de las ventanas comunes versus la ventana Kaiser	59
Tabla 3. 7: Respuesta Impulsiva del filtro ideal deseado $h_d(n)$	60
Tabla 3. 8: Coeficientes del filtro ideal	60
Tabla 3. 9: Especificaciones del espectro para filtros IIR	63
Tabla 3. 10: Especificaciones de la respuesta de magnitud cuadrática.	63
Tabla 3. 11: Relación entre los tipos de filtros y las transformaciones de frecuencia analógica.	67

Capítulo 4.

Tabla 4. 1: Funciones para determinar los coeficientes del filtro IIR	68
Tabla 4. 2: Funciones para determinar el orden mínimo de un filtro IIR	68

RESUMEN

El presente proyecto se trata sobre la elaboración de un laboratorio virtual para mejorar el aprendizaje de filtros digitales en DSP, para la ejecución del proyecto se decidió ocupar un software de fácil acceso y manejo, para este fin se optó por utilizar Matlab, con este se realiza una interfaz gráfica para el diseño de filtros digitales.

En el capítulo I, se detalla el planteamiento del problema; donde, se analiza y se contextualiza mediante un enfoque crítico. Además toda la investigación percibe un objetivo al crear una herramienta para que los estudiantes puedan realizar práctica sobre filtros digitales y de esa manera poner en práctica la teoría aprendida en el aula.

El capítulo II, presenta el Marco teórico; el cual, constituye la fundamentación legal y teórica, los conceptos elementales que fundamentan teóricamente el problema planteado; formulándose la hipótesis a través de la definición de las variables dependiente e independiente.

En el Capítulo III, se presenta el enfoque, la modalidad básica de investigación, los tipos o nivel de investigación, la población y muestra con la cual se trabajó, operacionalización de variables para un análisis posterior

En el Capítulo IV, se realiza las encuestas describiendo el análisis e interpretación de los resultados; y así, asegurar y confirmar la necesidad de dar una solución al problema planteado.

En el Capítulo V, se definen las conclusiones y recomendaciones provenientes del análisis efectuado en el Capítulo anterior, las mismas que dieron las pautas necesarias para dar inicio al desarrollo de la propuesta.

Finalmente en el Capítulo VI, se presenta la propuesta, en este caso un laboratorio virtual para el mejoramiento del aprendizaje de filtros digitales en DSP. Primero se presenta una investigación teórica acerca del diseño de filtros digitales. Esta investigación teórica permite reforzar y entender con claridad los conceptos básicos. Segundo se presentan los enunciados de las prácticas para los temas presentados en el marco teórico. Por último, se muestra el código fuente generado para la solución de los ejercicios de simulación en MATLAB planteados en cada una de las prácticas de laboratorio, así como los gráficos obtenidos en cada una de las prácticas planteadas.

CAPÍTULO I: DESCRIPCIÓN GENERAL DEL TRABAJO DE TITULACIÓN

1.1. INTRODUCCIÓN

El gran avance tecnológico tanto de las Ciencias Exactas e Ingeniería, en el despliegue de circuitos integrados, microprocesadores y computadoras en los últimos 30 años, ha generado gran interés en desarrollar investigaciones acerca del Procesamiento Digital de Señales. En otras palabras, este se ha vuelto tan indispensable, que existen un sinnúmero de aplicaciones en muchas disciplinas.

Por esto el presente trabajo de titulación es de gran interés para la Carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, en la cual se desarrollarán una serie de prácticas para simular filtros digitales en la materia de DSP, ofreciendo una plataforma versátil, la misma que podrán acceder los estudiantes que siguen Ingeniería en Telecomunicaciones para fortalecer la parte práctica ante la falta de laboratorios que dispongan de módulos de entrenamiento de filtros digitales. La idea fundamental es ofrecer un manual de prácticas que permita la simulación a través de la plataforma MATLAB.

1.2. JUSTIFICACION:

El presente trabajo de titulación pretende brindar una herramienta práctica para que los estudiantes de la Carrera de Ingeniería en Telecomunicaciones desarrollen correctamente la parte teórica aprendida en las clases de Procesamiento Digital de Señales (DSP), concretamente al estudio de los filtros digitales. Actualmente el DSP tiene desarrollado un gran campo de acción, es decir, que las prestaciones y las aplicaciones ofrecidas son robustas.

Muchas veces la falta de recursos económicos impiden la adquisición de equipos adecuados para la realización de las prácticas necesarias de las diferentes asignaturas, sin embargo un laboratorio virtual, sería una excelente alternativa para suplir esta

carencia, refiriéndose específicamente al DSP con éste se pueden realizar prácticas no físicamente sino mediante un software, con el cual también se puede realizar procesamiento de señales, teniendo así una herramienta para poder realizar prácticas sin tener que invertir mucho dinero, brindando así una alternativa para poner en práctica las aplicaciones y teoría adquiridos en el módulo de DSP.

1.3. ANTECEDENTES.

A menudo se escucha la frase mundo digital, pero hasta donde se está consciente de esta realidad que se vive. Hasta hace algunas décadas el mundo había subsistido con sistemas que implementaban tecnología análoga para el desarrollo de aplicaciones. Sin embargo con la aparición del microprocesador, la electrónica y el mundo entero toman un nuevo enfoque; explorando las bondades y efectividad de la implementación de procesos digitales, dejando su perspectiva de desarrollo de tecnologías análogas sólo para ciertas aplicaciones.

Con la implementación de esta nueva tecnología nace la necesidad de encontrar nuevas herramientas que realicen con alta efectividad estos procesos; generando un gran interés en el estudio del Procesamiento Digital de Señales o DSP. Este tema de estudio se ha vuelto tan indispensable que ha sido aplicado a muchas disciplinas tanto en ingeniería como economía y desde la astronomía hasta la biología molecular.

A través del DSP, se han desarrollado hoy en día, sofisticados sistemas de comunicación, nació el Internet, se ha podido obtener valiosa información acerca del cosmos a partir de las señales astronómicas, las señales sísmicas pueden ser analizadas para determinar la magnitud de un terremoto o para predecir la estabilidad de un volcán, las imágenes o fotografías por computador pueden ser ahora mejoradas, entre muchas otras cosas más.

En los países europeos, tales como España, Alemania, Italia se desarrollan

proyectos mediante software y hardware de DSP's, cuya utilización es muy amplia, por ejemplo se la utiliza en los campos de: Audio como ecualizadores gráficos, analizadores de espectro, compresión de audio (MiniDisc), discos compactos, efectos acústicos, sistema de teatro casero (THX), reducción de ruido (Dolby); en la Industria Automotriz en frenos antibloqueo, bolsas de aire, supresión de ruido activo, control de mezcla de combustibles; en Computadoras se la utiliza Módems, tarjetas de sonido, tarjetas de video, sintetización y reconocimiento de voz; en Comunicaciones se utiliza para la compresión de voz, eliminación de eco, ecualización adaptativa, multiplexión de canales.

En la carrera de Ingeniería en Telecomunicaciones se ha mostrado un gran interés en el estudio de DSP pues como se lo menciona anteriormente en un mundo totalmente digitalizado es importante conocer los beneficios que ésta herramienta nos ofrece, sin embargo en la institución no se puede apreciar éstos pues no se cuenta con laboratorios adecuados para la realización de prácticas sobre DSP, no obstante la institución aporta con conocimientos para la investigación y futuras aplicaciones de los DSP.

1.4. DEFINICIÓN DEL PROBLEMA:

La falta de recursos económicos es un factor relevante que impide la adquisición de equipos adecuados para realizar prácticas enfocadas a la materia de DSP, lo que logran con esto es un mejor entendimiento de la materias mencionada. En muchas ocasiones la falta de espacios, es decir, de infraestructura, no permite que se desarrollen prácticas de DSP. Por esto surge la necesidad de desarrollar un laboratorio virtual para dar un enfoque claro y práctico de lo aprendido.

Otro factor, es la falta de personas capacitadas en el uso de equipos y mucho menos que manejen correctamente el software, para lo cual es necesaria la capacitación de personal académico y de soporte (asistentes de laboratorio) que conozcan al dedillo el uso adecuado de los equipos, para que estos no sufran daños, así como también un adecuado mantenimiento, en el caso de un

laboratorio virtual se necesita personas con amplios conocimientos tanto de la materia como del software a utilizarse para la simulación de los circuitos que se implementaran para que de esa manera los estudiantes tengan una mejor comprensión de dichas materias.

1.5. DELIMITACION DEL PROBLEMA:

Campo: Electrónica

Área: Procesamiento Digital de Señales

Aspecto: Prácticas de Filtros Digitales en DSP

Delimitación Temporal: el presente trabajo de titulación se lo realizó en 4 meses, a partir de su aprobación por parte de la Comisión Académica de la Carrera de Ingeniería en Telecomunicaciones y por el Consejo Directivo de la Facultad de Educación Técnica para el Desarrollo.

1.6. OBJETIVOS:

1.6.1. Objetivo General:

Analizar las prácticas de laboratorio y su influencia en el aprendizaje de filtros digitales en DSP.

1.6.2. Objetivos Específicos:

- a. Identificar los niveles de aprendizaje y práctica de Filtros Digitales en DSP en los estudiantes de la carrera de Ingeniería en Telecomunicaciones.
- b. Plantear una propuesta de un laboratorio virtual de filtros en DSP.

CAPITULO II: FUNDAMENTACIÓN TEÓRICA DE DSP'S

2.1. Fundamentación Legal.

Esta investigación se basó en las políticas de la Universidad Católica de Santiago de Guayaquil bajo el reglamento y estatutos que decreta de Ley Orgánica de Educación Superior.

2.2. Gráficas de inclusión de las categorías fundamentales.

En la figura 2.1 se aprecia la inclusión tanto de la variable independiente como la dependiente y su relación.

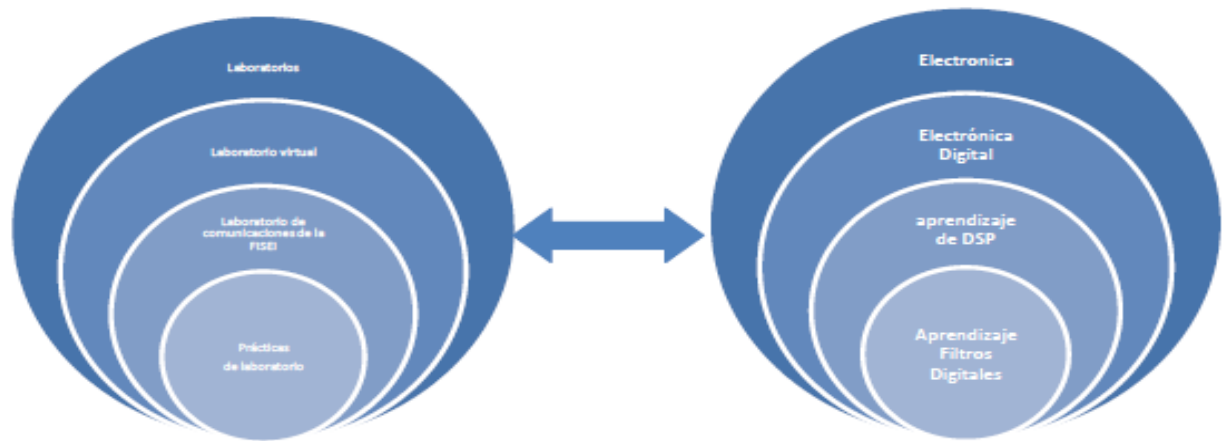


Figura 2. 1: Categorías Fundamentales
Elaborado por el Investigador

2.3. Categorías Fundamentales.

2.3.1. Electrónica

La electrónica es el área de las ciencias físicas o conocido como tecnología de los semiconductores, es decir, que este fenómeno físico se encarga de producir los movimientos de partículas cargadas en el vacío, así como de los gases y los semiconductores (silicio o germanio), que se basan en el diseño y fabricación de dispositivos electrónicos o de ciertos circuitos integrados (compuertas lógicas, microcontroladores, microprocesadores, etc.) o sistemas embebidos basados en ellos.

La electrónica es considerada también una de las ramas de las tecnologías de la información (TIC's), en otras palabras que mediante las TIC's puedan adquirir, producir, almacenar, procesar, comunicar y presentar datos analógicos/digitales o de todo tipo de señales físicas, tales como ondas acústicas, fibra óptica, sistemas eléctricos, etc.). Los circuitos electrónicos ofrecen diferentes funciones para procesar esta información, incluyendo la amplificación de señales débiles hasta un nivel que se pueda utilizar; el generar ondas de radio; la extracción de información, como por ejemplo la recuperación de la señal de sonido de una onda de radio (demodulación); el control, como en el caso de introducir una señal de sonido a ondas de radio (modulación), y operaciones lógicas, como los procesos electrónicos que tienen lugar en las computadoras. Esta ciencia se divide en dos grandes ramas: Analógica y Digital.

2.3.2. Electrónica Digital

La Electrónica Digital es la parte de la electrónica que se encarga del estudio de sistemas electrónicos en los cuales la información está codificada en dos únicos estados, es decir, trabaja con variables discretas. Este hecho implica que un pequeño cambio en alguna de las variables del circuito no producirá un cambio apreciable en el comportamiento del circuito. Es decir, el comportamiento del circuito no depende del valor exacto de la señal.

A dichos estados se les puede llamar "verdadero" o "falso", o más comúnmente 1 y 0. Electrónicamente se les asigna a cada uno un voltaje o rango de voltaje determinado, a los que se les denomina niveles lógicos, típicos en toda señal digital. La razón básica para el estudio de los circuitos digitales radica en el aumento de fiabilidad en el procesamiento y transmisión de la información, ya que una pequeña degradación de la señal no influirá en su valor (o en su influencia como entrada en un circuito digital). La electrónica digital ha alcanzado una gran importancia debido a que es utilizada para realizar autómatas y por ser la piedra angular de los sistemas micro programado como son los ordenadores o computadoras.

2.3.3. DSP

Las señales eléctricas son tensiones o corrientes que contienen información. Además de las señales eléctricas existen otras, de naturaleza magnética, hidráulica, neumática, luminosa, etc. Las señales pueden ser generadas en forma natural o artificial. Algunos ejemplos de señales naturales son la radiación electromagnética de una estrella, la altura de la marea y la velocidad del viento. Algunos ejemplos de señales artificiales son la emisión de un canal de TV, las ondas emitidas y recibidas por radares, teléfonos celulares, sonares, etc. Las señales se representan matemáticamente como funciones de una o más variables independientes. La variable independiente más común es el tiempo, y algunas señales que dependen de él son, por ejemplo, la voz, una onda de radio, un electrocardiograma, etc. Otras señales, tales como las imágenes, son funciones de 2 variables independientes, ya que contienen información de brillo o de colorido en función de las coordenadas X e Y de un plano.

En las últimas décadas se ha producido una migración cada vez mayor desde el procesamiento análogo hacia el procesamiento digital. Al mismo tiempo, han surgido muchas aplicaciones (véase la figura 2.2) y técnicas nuevas, que nunca existieron en el mundo análogo.



Figura 2. 2: Aplicaciones del Procesamiento Digital de Señales.

Elaborado por el Investigador

A continuación se mencionan algunas aplicaciones actuales de DSP's:

- **Radar (*Radio Detection And Ranging*):** Estos son sistemas de lo más sencillos, a través de un transmisor se generan pulsos de energía de radiofrecuencia en tiempos muy pequeños (microsegundos). Estos son emitidos desde una antena altamente direccional. Para el caso de los aviones comerciales que atravesen el haz de esta onda, inmediatamente será reflejada una pequeña energía hasta la antena de recepción, ubicada cerca del sitio de transmisión. Mediante el tiempo transcurrido entre el emisor (pulso) y la recepción (eco), se podrá hallar la separación (distancia) del objeto y del rango. Mientras que la antena direccional nos indicará la dirección del objeto.

La operatividad de un Sistema de Radar se deben a dos parámetros: (1) cantidad de energía contenida en el pulso inicial; y (2) el nivel de ruido del receptor. En general, dicho incremento (energía del pulso) comprime la precisión y la exactitud de la medición del tiempo transcurrido, provocando así inconvenientes entre dos parámetros importantes: la habilidad para detectar objetos lejanos, y determinar su distancia con exactitud.

Mediante los DSP's los radares mejoraron notablemente en tres situaciones, que se describen a continuación:

1. Minimizando el pulso levemente es recibido, mejorando así determinación de separación sin comprimir el rango de operatividad, esto se visualiza en la figura 2.3.
2. Al momento de realizar el filtrado de señales se reduce el nivel de ruido, logrando incrementar el rango, sin que se degrade la determinación de separación.
3. Al generar de manera acelerada los pulsos con diferentes longitudes de onda, se logra optimizar el problema de detección.

Para implementaciones posteriores, este procedimiento es el adecuado, donde la velocidad de muestreo se compara con la radiofrecuencia utilizada en MHz. Para trabajar a altas velocidades en la compresión se requiere tanto del hardware como del software (algoritmos).

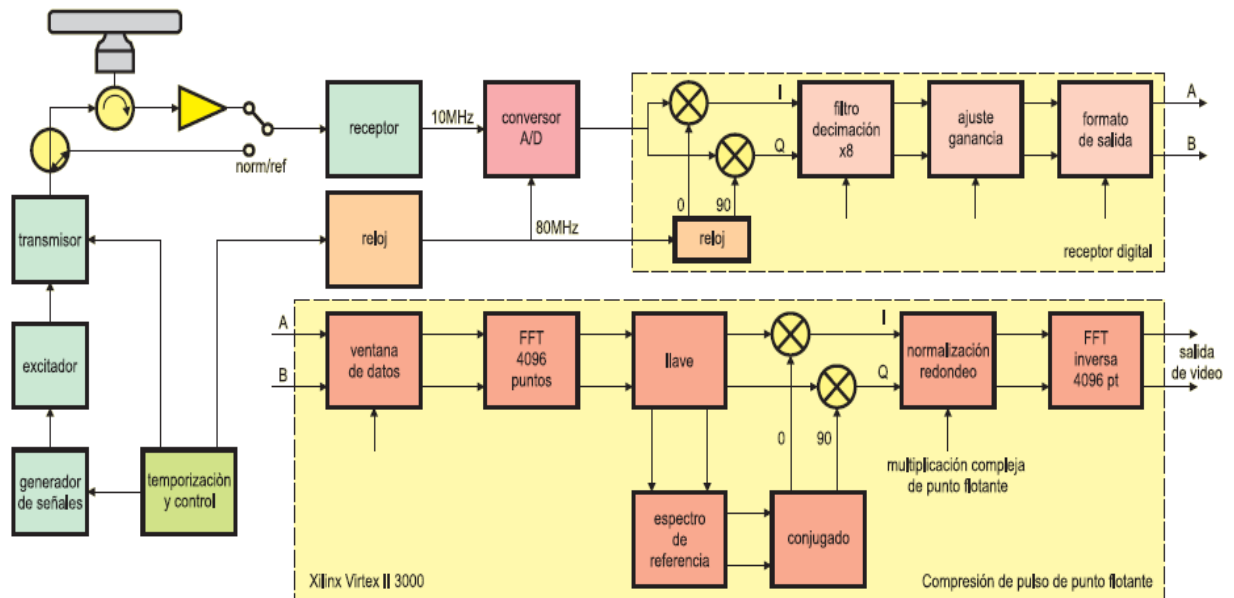


Figura 2. 3: Sistema básico de compresión de pulsos de radar.

Fuente: <http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap01/01-Cap01.pdf>

- **Imágenes:** En realidad las imágenes se consideran señales especiales cuyas características dependerán: (1) En primera instancia, se medirá el parámetro sobre una posición a determinar, aunque todas las señales son medidas en función del tiempo. (2) En segunda instancia, las imágenes disponen de grandes cantidades de información, por ejemplo, si guardamos una señal de TV se requieren de 10 Mbytes. Es decir, que requieren una cantidad mil veces mayor a la disponibilidad de memoria. (3) Finalmente, la calidad de la imagen, es continuamente evaluada (subjetiva, por el ser humano). En conclusión, podemos sostener que el procesamiento de imágenes es un subgrupo distinto al DSP.
- **Telecomunicaciones:** son las encargadas de transmitir/recibir información o datos desde un punto a otro, tales como realizar una conversación telefónicas fija o móvil, transmisión de señales de TV, envío de paquetes de datos

mediante la gran nube (internet), etc. Todo esto se logra a partir de un canal o medio de transmisión necesario entre el emisor y el receptor.

Este medio de transmisión puede ser: cable coaxial, microondas, satélite, fibra óptica, etc. Las empresas públicas y privadas de servicios de Telecomunicaciones cobran por los servicios que prestan a los usuarios o clientes, tales servicios son: telefonía fija, telefonía móvil (celular), internet, DTH, etc. Los DSP's han crecido tecnológicamente para determinadas aplicaciones en la industria de las comunicaciones.

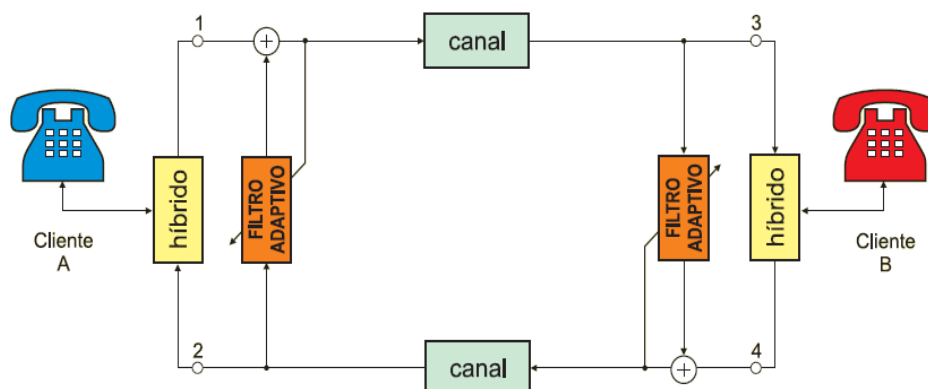


Figura 2. 4: Cancelamiento de eco en redes telefónicas de larga distancia.

Fuente: <http://prof.usb.ve/tperez/investigacion/ondiculas/eco/ECOWavele.htm>

En la figura 2.4 se muestra el diagrama de bloques para el cancelamiento de eco en redes de telefonía convencional a largas distancias, donde la señal de entrada (Cliente A) pasa simultáneamente al híbrido y a un filtro adaptativo, para reducir el eco producido en el canal, restándolo de la señal original.

- **Audio:** el audio se considera como una onda sonora, que tiene un determinado rango de frecuencias (entre 20 Hz y 20 kHz) permisibles para el oído humano. Aunque, esta onda sonora es conocida como señal eléctrica analógica, no significa que el audio sea completamente digital. El proceso de digitalizar el audio, consiste en la secuencia de valores enteros, es decir, que se debe muestrear y cuantificar digitalmente señales eléctricas. Los DSP's disponen diferentes aplicaciones en audio, tales como: ecualizadores, compresión de música en formatos MP3 y MP4, etc.

- **Voz:** La voz es una señal acústica, la cual consiste en un conjunto de sonidos generados por el ser humano, que también es convertida en una señal eléctrica mediante un micrófono. Esta señal acústica (voz) es representada gráficamente en función del tiempo, así como las ondas sonoras, formadas básicamente por las combinaciones de funciones senoidales y cosenoidales, aunque resulten ser complejas. La señal de voz a través de los DSP's realizan diferentes aplicaciones, como por ejemplo el reconocimiento de voz de seres humanos.

En el caso del reconocimiento de voz, los DSP's se enfocan en dos procesos tales como la extracción y comparación de patrones, estos procesos son mostrados en el diagrama de bloques de la figura 2.5.

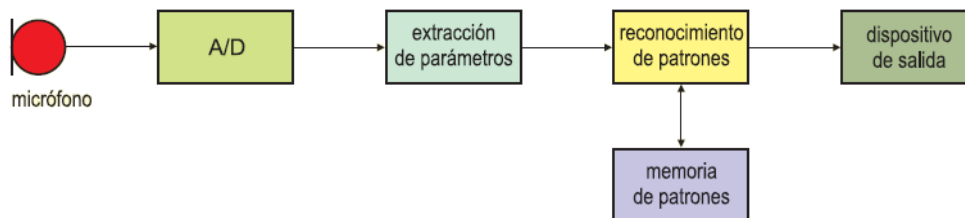


Figura 2. 5: Diagrama de bloques de un sistema de reconocimiento de voz

Fuente: <http://blog.hertasecurity.com/>

Las anteriores aplicaciones de los DSP's no son las únicas, hay otras aplicaciones para el mercado de la Industria Automotriz, para encendidos de vehículos, sistemas de inyección en motores a gasolina y diésel, todo esto hace que se maximice el rendimiento y minimice las emisiones de dióxido de carbono; control del flujo de energía en los vehículos híbridos; para estudios sismológicos, para localizar hipocentros de sismos, búsqueda de minerales y de petróleo analizando los ecos subterráneos generados por pequeñas explosiones; aplicaciones en ciencias médicas, para reducir el ruido y diagnóstico automático de electrocardiogramas y electroencefalogramas; formación de imágenes en tomografía axial computarizada (scanner), resonancia magnética nuclear y ecografía (ultrasonido); Oceanografía en la alerta temprana de maremotos o tsunamis cuando se propagan en el océano abierto, en base a las características de esas ondas que las diferencian de las olas y de las mareas, análisis armónico

y predicción de mareas, medición de la energía de las olas con el objeto de dimensionar muelles y otras estructuras sumergidas; Astronomía en la detección de planetas en estrellas lejanas, en base al movimiento oscilatorio que inducen en las estrellas alrededor de las cuales orbitan.

Estas son sólo algunas aplicaciones de DSP. Su número está en constante aumento, a medida que la tecnología de integración progresa, y se desarrollan nuevos algoritmos.

2.3.4. Filtros Digitales

Los filtros son componentes básicos de todos los sistemas de procesamiento de señal y de telecomunicaciones. Las funciones principales de un filtro son uno o más de los siguientes:

- a. Para confinar una señal en una banda de frecuencia prescrita o canal, por ejemplo, como en el filtro anti-aliasing o un selector de canal de radio/televisión,
- b. Para descomponer una señal en dos o más señales de sub-banda para el procesamiento de señal de sub-banda, por ejemplo, en la codificación de música,
- c. Para modificar el espectro de frecuencia de una señal, por ejemplo, en ecualizadores gráficos de audio, y
- d. Para modelar la relación entrada-salida de un sistema, como un canal de comunicación móvil, producción de la voz, los instrumentos musicales, el eco de la línea telefónica, y la acústica de la sala.

Los filtros digitales se emplean ampliamente en los sistemas de procesamiento y comunicación de señal en aplicaciones tales como ecualización de canal, la reducción de ruido, de radar, procesamiento de audio, el procesamiento de vídeo, procesamiento de señales biomédicas, y el análisis de los datos económicos y financieros. Por ejemplo, en un receptor de radio el filtro paso de banda, se utilizan para extraer las señales de un canal de radio.

En un ecualizador gráfico de audio de la señal de entrada se filtra en una serie de señales de sub-banda y la ganancia para cada sub-banda se puede variar de forma manual con un conjunto de controles para cambiar la sensación de audio percibida. En un sistema Dolby pre-filtrado y filtrado posterior se utilizan para minimizar el efecto del ruido. En audio hi-fi de un filtro de compensación puede ser incluido en el preamplificador para compensar las características de respuesta en frecuencia no ideales de los altavoces. Los filtros también se utilizan para crear efectos audiovisuales de percepción de la música, las películas y en los estudios de emisión.

Para el funcionamiento de los filtros digitales se requiere de una entrada analógica, muestreada de manera periódica, que convierta en varias series cuyas muestras son $x[n]$, $n = 0, 1, \dots$. En la figura 2.6 se muestra el diagrama esquemático de un filtro digital.

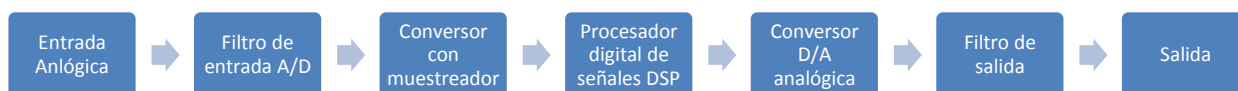


Figura 2. 6: Diagrama esquemático de un Filtro Digital

Elaborado por: El investigador

2.3.4.1. Tipos de filtros

Existen cuatro tipos de filtros y se clasifican de acuerdo a las frecuencias que discriminan, sea dicho que afectan ya sea dejando pasarlas o rechazándolas.

a. Filtros Pasa Bajos

Un filtro que proporcione una salida constante a partir de CD hasta una frecuencia de corte determinada y que más allá de ella no permita pasar a ninguna señal, es un filtro ideal pasa bajas. En otras palabras, se caracteriza por dejar pasar frecuencias bajas en su banda de paso y rechazar frecuencias altas en la banda de rechazo. Dichas bandas quedan establecidas por:

- Banda de paso: desde 0 hasta ω_p
- Banda de rechazo: desde ω_s hasta ∞

Dónde:

ω_p : frecuencia de paso

ω_s : frecuencia de corte.

En la figura 2.7 se ilustra las características de un filtro pasa bajo.

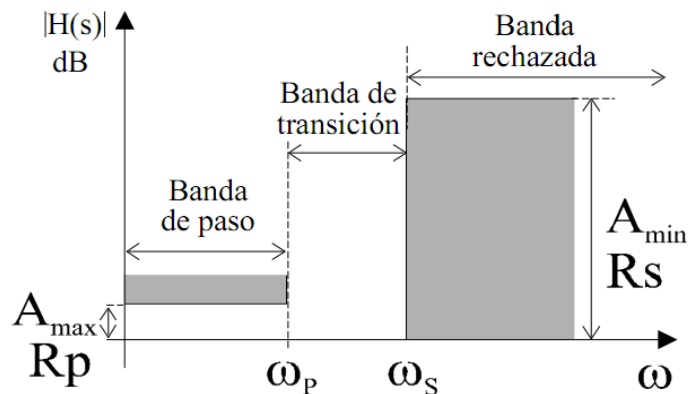


Figura 2. 7: Características de filtro pasa bajo

Fuente: <http://www.dtic.upf.edu/~jlozano/audio/edicion2.html>

b. Filtro Pasa Alto:

Es aquel filtro, que al contrario del anterior, deja pasar frecuencias altas en su banda de paso y rechaza frecuencias bajas en su banda de rechazo. En pocas palabras, únicamente permite el paso de señales arriba de su frecuencia de corte. Las bandas de frecuencia que describen a este tipo de filtro, se describen por:

- Banda de paso: desde ω_p
- Banda de rechazo: desde 0 hasta ω_s

Donde:

ω_p : frecuencia de paso

ω_s : frecuencia de corte

En la figura 2.8 se muestra las características de un filtro pasa alto.

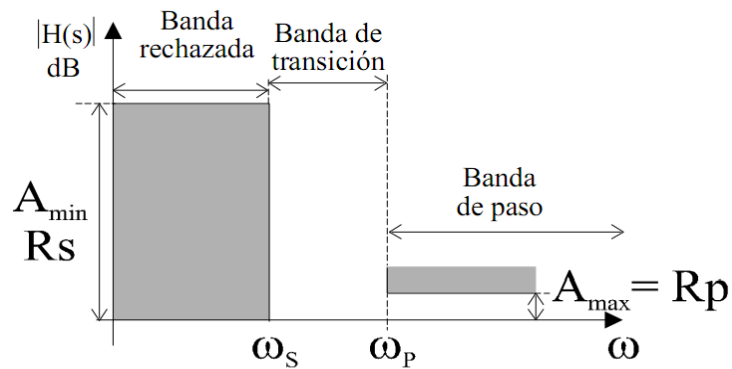


Figura 2. 8: Características de un filtro pasa alto

Fuente: <http://www.dtic.upf.edu/~jlozano/audio/edicion2.html>

c. Filtros Pasa banda

Se denomina de esta manera a aquel en el cual las señales son transmitidas si se encuentran arriba de una primera frecuencia de corte y por debajo de una segunda frecuencia de corte. Esto significa que en este tipo de filtros hay presentes dos bandas de rechazo, una inferior y una superior, las cuales definen la banda de paso y únicamente dejará pasar frecuencias dentro del rango definido. Esto queda definido por:

- Banda de paso: desde ω_{p1} hasta ω_{p2}
- Banda de rechazo inferior: desde 0 hasta ω_{s1}
- Banda de rechazo superior: desde ω_{s2} hasta ∞

Donde:

ω_{p1} y ω_{p2} : frecuencia de paso inferior y superior respectivamente

ω_{s1} y ω_{s2} : frecuencia de corte inferior y superior respectivamente

$$\omega_{s1} < \omega_{p1} < \omega_{p2} < \omega_{s2}$$

En la figura 2.9 se indica las características de un filtro pasa banda.

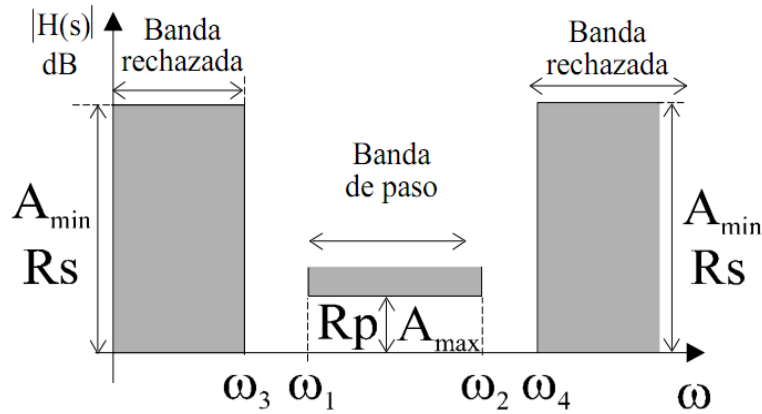


Figura 2. 9: Características de filtro pasa banda

Fuente: <http://www.dtic.upf.edu/~jlozano/audio/edicion2.html>

d. Filtro Rechaza Banda:

Cuando el circuito de filtrado bloquea señales que se encuentran por encima de una frecuencia de corte y por debajo de una segunda frecuencia de corte, se dice que es un filtro rechaza banda. Este tipo de circuito, al contrario del filtro pasa banda, tiene una banda de paso inferior, una banda de rechazo y una banda de paso superior, y funciona bloqueando sólo cierto rango de frecuencias. Lo anterior se indica de la siguiente manera:

- Banda de paso inferior: desde 0 hasta ω_{p1}
- Banda de rechazo: desde ω_{s1} hasta ω_{s2}
- Banda de paso superior: desde ω_{s2} hasta ∞

Donde:

ω_{p1} y ω_{p2} : frecuencia de paso inferior y superior respectivamente

ω_{s1} y ω_{s2} : frecuencia de corte inferior y superior respectivamente

$$\omega_{p1} < \omega_{s1} < \omega_{s2} < \omega_{p2}$$

En la figura 2.10 se muestra las características de este filtro:

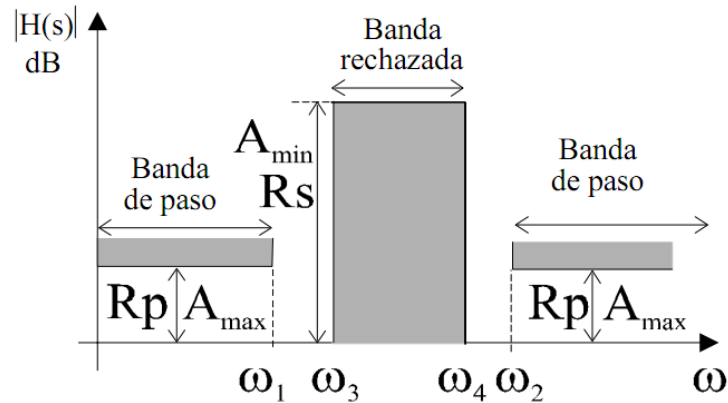


Figura 2. 10: Características de filtro rechaza banda

Fuente: <http://www.dtic.upf.edu/~jlozano/audio/edicion2.html>

2.4. Por su respuesta impulsional:

2.4.1. Filtros FIR (respuesta impulsional finita).

Es un tipo de filtro digital que si su entrada es un impulso la salida será un número limitado de términos no nulos. Para obtener la salida sólo se emplean valores de la entrada actual y anterior. También se llaman filtros digitales no recursivos. Su expresión en el dominio discreto es:

$$y_n = \sum_{k=0}^{N-1} b_k x(n-k)$$

El orden del filtro está dado por N, es decir, el número de coeficientes. También la salida puede ser expresada como la construcción de una señal $x(n)$ con un filtro $h(n)$.

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k}$$

La estructura de un filtro FIR se muestra en la figura 2.11.

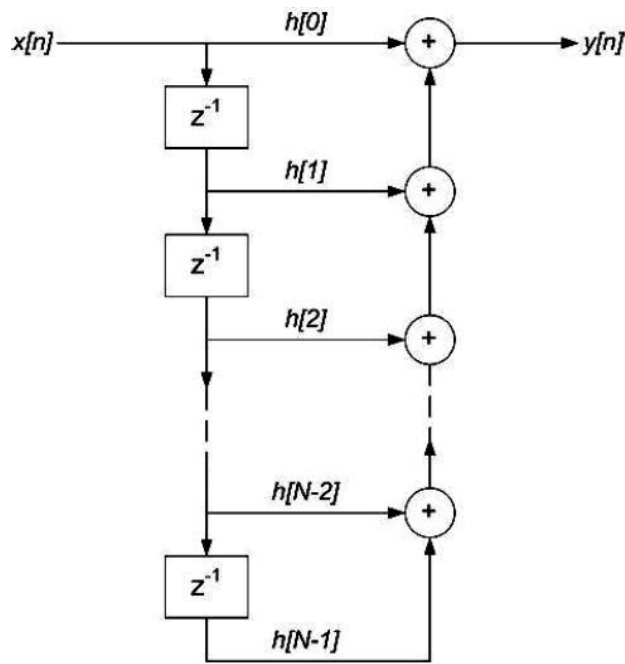


Figura 2. 11: Diagrama de bloques de la estructura en forma directa de un filtro FIR

Fuente: <http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap12/12-Cap12.pdf>

Lo cual puede verse reflejada en la aplicación de la transformada Z:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)}$$

Se puede ver que es la misma entrada retardada cada vez más en el tiempo, multiplicada por diversos coeficientes y finalmente sumada al final. Hay muchas variaciones de esta estructura. Si tenemos una respuesta de frecuencia como objetivo, conseguiremos que la respuesta del filtro se asemeje más a ella cuando más número de coeficientes tenga.

Los filtros FIR son estables puesto que solo tienen polos, es decir, elementos en el numerador en su función de transferencia. También tienen la ventaja que pueden diseñarse para ser de fase lineal, es decir, no introducen desfases en la señal, a diferencia de los IIR o los filtros analógicos.

2.4.2. Filtros IIR (respuesta impulsional infinita)

Es un tipo de filtro digital que si su entrada es un impulso la salida será un número ilimitado de términos no nulos, es decir, que nunca volverá a un estado de reposo. Para obtener la salida se emplean valores de la entrada actual y anterior y, además, valores de salida anteriores que son almacenados en memoria y realimentados a la entrada. También se llaman filtros digitales recursivos. Su expresión en el dominio discreto es:

$$y(n) = \sum_{i=0}^P b_i x(n-i) - \sum_{j=1}^Q a_j y(n-j)$$

El orden del filtro está dado por el máximo entre P y Q. Una de las posibles estructuras para un filtro IIR es la siguiente:

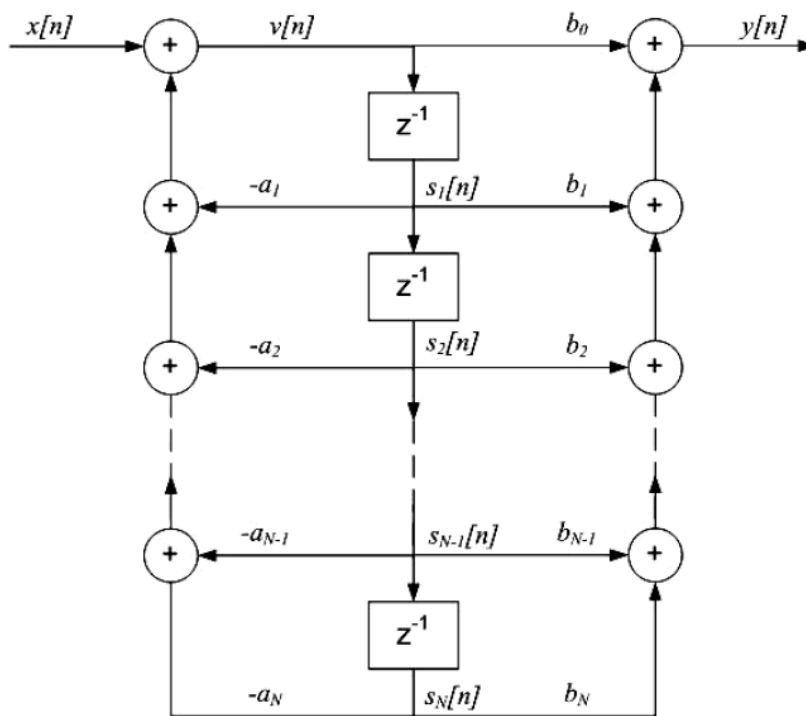


Figura 2. 12: Diagrama de bloques de la estructura en forma directa tipo 1 de un filtro IIR

Fuente: <http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap11/11-Cap11.pdf>

En la figura 2.12 se puede ver cómo la salida $y(k)$ es introducida de nuevo en el sistema. La transformada Z del mismo es:

$$H(z) = \frac{\sum_{i=0}^P b_i z^{-i}}{\sum_{j=0}^Q a_j z^{-j}}$$

2.5. FIR vs IIR

La elección entre una implementación FIR e IIR depende de las ventajas relativas de cada uno de estos dos tipos de filtros.

- Los filtros FIR se pueden diseñar para tener una respuesta de fase estrictamente lineal (distorsión de fase nula), lo que es importante en muchas aplicaciones, como transmisión de datos, audio digital y procesamiento de imágenes. La respuesta de fase de filtros IIR no es lineal, en especial en cercanías de la zona de transición.
- Los filtros FIR implementados de forma no recursiva, son inherentemente estables. En cambio, la estabilidad de los filtros IIR siempre debe comprobarse, ya que son sistemas realimentados.
- Los efectos causados por la implementación con aritmética de punto fijo, tales como los errores de cuantización de los coeficientes y los errores por redondeo en las operaciones aritméticas, son mucho más severos en los filtros IIR que en los FIR.
- Para satisfacer unas especificaciones dadas los filtros FIR necesitan un mayor número de coeficientes que los filtros IIR, sobre todo si las bandas de transición son estrechas. En consecuencia, los requerimientos de memoria, el número de operaciones y los tiempos de procesamiento son mayores para los FIR que para los IIR. Sin embargo, la posibilidad de implementar los FIR mediante la técnica de convolución rápida usando FFT y también el empleo de técnicas “multirate” permite aumentar significativamente la eficiencia de las implementaciones.
- Un filtro analógico convencional puede convertirse en un filtro digital IIR equivalente que satisfaga las especificaciones de diseño de manera sencilla. Sin embargo es más sencillo sintetizar filtros con respuestas en frecuencia arbitrarias utilizando filtros FIR.

- Los filtros FIR ofrecen en general una respuesta de fase más lineal y no entran jamás en oscilación (es decir, no se vuelven inestables), ya que no poseen realimentación. Por otro lado, requieren un gran número de términos en sus ecuaciones y eso les hace más costosos en cuanto a cálculo o carga computacional.
- Un filtro FIR con un corte muy abrupto (es decir, que tenga una banda de transición muy corta) puede requerir hasta centenas de retardos.
- En cuanto a los filtros IIR, son muy eficaces y pueden proporcionar pendientes de corte muy pronunciadas. Por otro lado, al poseer características de realimentación (o feedback), tienen tendencia a entrar en oscilación y en resonancia.

2.6. Diseño de Filtros Digitales

El diseño de un filtro digital involucra los siguientes pasos:

- Especificación de los requerimientos del filtro.
- Elección de una aproximación conveniente y cálculo de los coeficientes.
- Representación del filtro utilizando una estructura adecuada (realización).
- Análisis de los efectos de la longitud finita de palabra en el desempeño.
- Implementación en hardware o software.

Estos cinco pasos no son necesariamente independientes, ni necesitan seguirse en el orden descrito; actualmente, las técnicas de diseño disponibles combinan el segundo y parte del tercero y cuarto. Para lograr un filtro eficiente es necesario iterar entre las distintas etapas, especialmente si, como es habitual, las especificaciones de diseño dejan cierto grado de libertad al diseñador, o si se desean explorar otras alternativas de diseño.

2.6.1. Especificaciones de diseño

Los requerimientos incluyen la especificación de:

- Las características de las señales: tipo de fuente de señal, interfaz de entrada- salida, velocidad de procesamiento, ancho de palabra, la mayor frecuencia de interés.
- Las características del filtro: la respuesta en módulo y/o fase deseados y sus tolerancias, la velocidad de operación, el modo de filtrado (en línea o fuera de línea).
- La forma de implementación: como una rutina de alto nivel en una computadora, o un programa específico para un DSP.

2.6.2. Elección de la aproximación y cálculo de los coeficientes

En esta etapa se elige alguno de los métodos de aproximación:

- Butterworth, Chebyshev, elíptico, etc. si el filtro es IIR
- Equirriple, óptimo, con ventanas, etc. si es FIR

Se calculan los valores de los coeficientes $h[n]$ del filtro FIR, o a_k y b_k del filtro IIR, de modo de satisfacer las especificaciones. El método empleado para determinar los coeficientes es distinto según el filtro sea IIR o FIR.

Tabla 2. 1: Métodos típicos para el diseño de filtros IIR y FIR

IIR	FIR
Invariación al impulso	Ventanas
Transformada bilineal	Bandas de Transición suaves
Ubicación directa de polos y ceros	Muestreo de Frecuencia
Óptimos	Óptimos

Fuente: <http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap09/09-Cap09.pdf>

El método de diseño se elige de acuerdo a la aplicación en particular. En la tabla 2.1 se aprecian los métodos típicos para el diseño de filtros digitales. Si bien influyen varios factores, el más importante es qué tan críticas son las especificaciones. La decisión “difícil” es optar entre FIR o IIR. En aquellos casos

en que las propiedades de los FIR (respuesta de fase estrictamente lineal, estabilidad inherente) son imprescindibles, la mejor elección puede ser el diseño por métodos óptimos, o usando ventanas (generalmente la de Kaiser). Si, en cambio, son deseables las características de los IIR (menor cantidad de coeficientes para especificaciones similares) el método de la transformada bilineal es apropiado para la mayoría de los casos.

2.6.3. Diseño de Filtros Digitales FIR con la técnica de ventanas

La técnica de ventaneo se basa en a la respuesta de un filtro aplicarle la ventana deseada por medio de una multiplicación de sus ecuaciones, la ventana hace que en el filtro real diseñado se tengan menos variaciones de transición o supresión y con esto se logre y filtrado más efectivo.

Lo primero es decidir las especificaciones de respuesta en frecuencia $H_d(\omega)$ y determinar su correspondiente en respuesta al impulso $h_d(n)$. Para diseñar un filtro FIR con características específicas tenemos las siguientes ecuaciones:

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d(n)e^{-j\omega n}$$

La ecuación de diseño para el filtro FIR resulta:

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n} d\omega$$

En general la respuesta $h_d(n)$ es infinita pero esta es truncada mediante el ventaneo, el punto: $n=M-1$ y multiplicando por la ventana rectangular o unitaria se obtiene.

$$h(n) = \begin{cases} h_d(n) & n = 0, 1, \dots, M - 1 \\ 0 & \text{otro caso} \end{cases}$$

Realizando cálculos para diferentes filtros se tiene:

Tabla 2. 2: Ecuación de diseño para diferentes tipos de Filtros FIR

Filtro pasa altas	$hd(n) = \begin{cases} \frac{\sin \pi(n-\frac{M-1}{2}) - \sin \omega_c(n-\frac{M-1}{2})}{\pi(n-\frac{M-1}{2})} \\ 1 - \frac{\omega_c}{\pi} \end{cases}$	Si $n = \frac{M-1}{2}$
Filtro pasa bajas	$hd(n) = \begin{cases} \frac{\sin \pi(n-\frac{M-1}{2}) - \sin \omega_c(n-\frac{M-1}{2})}{\pi(n-\frac{M-1}{2})} \\ \frac{\omega_c}{\pi} \end{cases}$	Si $n \neq \frac{M-1}{2}$ $n = \frac{M-1}{2}$
Filtro Pasa banda	$hd(n) = \begin{cases} \frac{\sin(\omega_c + \Delta\omega_c)(n-\frac{M-1}{2}) - \sin(\omega_c - \Delta\omega_c)(n-\frac{M-1}{2})}{\pi(n-\frac{M-1}{2})} \\ \frac{\omega_c + \Delta\omega_c}{\pi} - \frac{\omega_c - \Delta\omega_c}{\pi} \end{cases}$	
Filtro rechazo de banda	$hd(n) = \begin{cases} \frac{\sin \pi(n-\frac{M-1}{2}) + \sin(\omega_c - \Delta\omega_c)(n-\frac{M-1}{2}) - \sin(\omega_c + \Delta\omega_c)(n-\frac{M-1}{2})}{\pi(n-\frac{M-1}{2})} \\ 1 + \frac{\omega_c - \Delta\omega_c}{\pi} - \frac{\omega_c + \Delta\omega_c}{\pi} \end{cases}$	

Fuente: www.uaq.mx/investigacion/...VII/ITQ%20Martinez%20Barrera.doc

Una vez obtenida la $h(n)$ del filtro deseado se le aplica el tipo de ventana más adecuada a las necesidades realizando la multiplicación correspondiente de acuerdo a las siguientes fórmulas:

Tabla 2. 3: Tipos de Ventanas para diseño de Filtros FIR

Ventanas	w(n) respuesta impulsiva
Bartlett (triangular)	$1 - \frac{2 \left n - \frac{M-1}{2} \right }{M-1}$
Blackman	$0.42 - 0.5 \cos \frac{2\pi n}{M-1} + 0.08 \cos \frac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46 \cos \frac{2\pi n}{M-1}$
Hanning	$\frac{1}{2} \left(1 - \cos \frac{2\pi n}{M-1} \right)$

Fuente: www.uaq.mx/investigacion/...VII/ITQ%20Martinez%20Barrera.doc

El orden N se estima según la tabla.

Tabla 2. 4: Orden de Filtros FIR según la ventana utilizada

Ventana	ws- wp	Orden N	M = N/2
Rectangular	$4\pi/(N+1)$	31	15
Hanning	$8\pi/N$	65	32
Hamming	$8\pi/N$	65	32
Blackman	$12\pi/N$	97	48

Fuente: <http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap12/12-Cap12.pdf>

2.6.4. Diseño con ventana de Kaiser.

La ventana de Kaiser es una ventana ajustable que permite regular la amplitud de los lóbulos laterales a través de un parámetro, y el ancho del lóbulo principal de acuerdo al largo de la ventana.

$$w(n) = \frac{I_0 \left[\beta \sqrt{1 - \left(\frac{n-N/2}{N/2} \right)^2} \right]}{I_0(\beta)}$$

$$0 \leq n \leq N - 1$$

Donde lo es la función de Bessel modificada de primer tipo, definida por:

$$I_0(x) = \sum_{n=0}^{\infty} \left[\frac{1}{k!} \left(\frac{x}{2} \right)^k \right]^2$$

El parámetro β se calcula como:

$$\beta = \begin{cases} 0.1102 (A - 8.7) & A > 50 \\ 0.5842 (A - 21)^{0.4} + 0.07886(A - 21) & 21 \leq A \leq 50 \\ 0 & A < 21 \end{cases}$$

Donde A es la atenuación deseada en la banda de rechazo y el orden N del filtro se estima según:

$$N = \begin{cases} \frac{A - 7.95}{2.285\Delta w} + 1 & A > 21dB \\ \frac{5.794}{\Delta w} + 1 & A \leq 21dB \end{cases}$$

2.6.5. Diseño de Filtros IIR

El método tradicional de diseño de filtros IIR en tiempo discreto se basa en la transformación de un filtro analógico en un filtro digital que cumpla las especificaciones preestablecidas. Esta solución es razonable por varios motivos:

- a. El arte del diseño de filtros IIR analógicos está muy avanzado, y como se pueden obtener resultados útiles, es ventajoso utilizar los procedimientos de diseño que ya se han desarrollado para los filtros en tiempo continuo.
- b. Muchos métodos útiles de diseño de filtros IIR en tiempo continuo dan como resultado fórmulas de diseño simples en forma cerrada. Por tanto, los métodos de filtros IIR digitales que se basan en esas fórmulas estándar de diseño de filtros IIR continuos son fáciles de realizar. El hecho de que los diseños de filtros en tiempo continuo se puedan trasladar a diseños de filtros digitales no quiere decir que tengan la misma respuesta frecuencial. Generalmente sucede que el filtro analógico empleado para la aproximación tiene una respuesta en frecuencia diferente de la respuesta en frecuencia efectiva del filtro digital.

- c. Esta circunstancia indica que al diseñar un filtro digital se parte de un conjunto de especificaciones en tiempo discreto; mientras las características del filtro en tiempo continuo se obtienen de la transformación. Al realizar esta conversión se desea que la respuesta en frecuencia del filtro digital preserve las propiedades esenciales del filtro analógico. Esto implica concretamente que se espera que el eje imaginario del plano s se transforme en la circunferencia unidad del plano z .
- d. Una segunda condición es que un filtro estable analógico se debe de transformar en un filtro estable de tiempo discreto. Esto significa que si el filtro continuo tiene los polos en el semiplano negativos $\text{Re}(s) < 0$, el filtro digital tiene que tener los polos dentro del círculo unidad del plano z . Estas restricciones son básicas para las técnicas de diseño de los filtros digitales IIR.

2.6.6. Diseño de filtros IIR mediante transformadas bilineales

Los filtros recursivos pueden ser diseñados por varios métodos, siendo el más común el basado en las transformaciones bilineales. Este procedimiento requiere del conocimiento de la función de transferencia en el tiempo continuo del filtro a diseñar. Los coeficientes del filtro en el dominio s son transformados a uno equivalente en el dominio z . Los coeficientes de la discretización formarán el filtro IIR. El origen de este proceder viene dado por la cantidad de experiencia acumulada en el diseño de filtros analógicos.

Puede verse que la Transformación Bilineal mapea el eje imaginario del plano s en la circunferencia unitaria del plano z , y el semiplano izquierdo del plano s en el interior de la circunferencia unitaria del plano z . De esta forma, filtros analógicos estables se mapean en filtros digitales estables. El mapeo del plano s en el plano z con la transformación bilineal se representa en la figura 2.13.

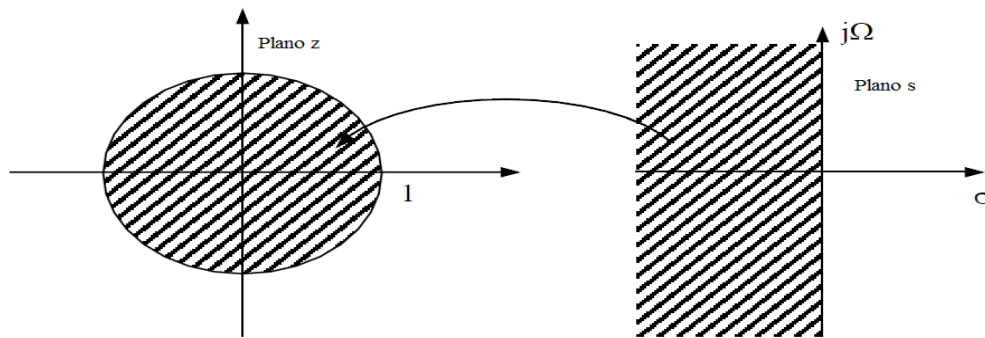


Figura 2. 13: Transformación Bilineal

Fuente: http://www.elai.upm.es:8009/spain/Publicaciones/pub01/intro_procsdig.pdf

El objetivo de la transformación Bilineal es lograr un filtro digital cuya respuesta temporal sea similar a la respuesta temporal del filtro analógico de origen, ante cualquier excitación. A continuación se expresa la función transferencia analógica:

$$H_A(s) = \frac{\sum_{i=0}^N a_i s^{N-i}}{s^N + \sum_{i=0}^N b_i s^{N-i}}$$

Y se obtiene $H_D(z)$ mediante el siguiente reemplazo:

$$s = \frac{2(z-1)}{T(z+1)}$$

Donde T: periodo de muestreo.

Por lo tanto:

$$H_D(z) = H_A(z) \Big|_{s=\frac{2(z-1)}{T(z+1)}}$$

2.7. Pasos para el diseño de filtros utilizando la transformada bilineal

A continuación se especifican los pasos para diseñar filtros mediante la transformada bilineal.

1. Se especifica el filtro digital deseado
 - a. Frecuencia de corte ω_c o bandas de frecuencia (ω_1, ω_2)
 - b. Atenuaciones en las bandas.

2. Se transforman las frecuencias discretas a frecuencias continuas

$$\Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right)$$

3. Se diseña el filtro analógico que cumple con las especificaciones de frecuencia y atenuaciones.

4. Se obtiene el filtro digital $H(z)$ utilizando la transformación.

2.8. Laboratorio Virtual.

Se trata de una serie de programas de simulación creados personal y expresamente, para el trabajo práctico de un laboratorio específico. En este caso se usan los ordenadores para simular el comportamiento de los sistemas a estudiar haciendo uso de modelos matemáticos como se aprecia en la figura 2.14.

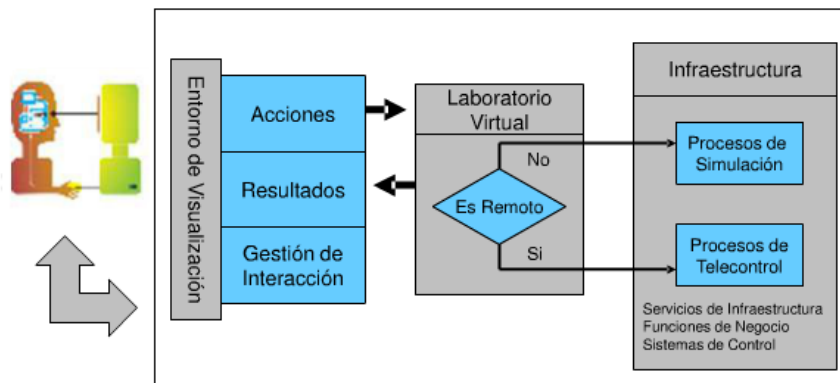


Figura 2. 14: Esquema de representación de un laboratorio virtual

Fuente: http://www.aves.edu.co/cava/cava2009/pdf/Francis_Castellanos.pdf

Aunque en este caso no se interacciona con plantas reales, la experimentación con modelos simulados es comparable siempre que se cumplan las siguientes premisas:

- Se usen modelos matemáticos realistas que representen al alumno los detalles importantes del sistema a analizar.
- Se complementen las gráficas que muestran la evolución temporal de los sistemas con animaciones que permitan a los alumnos visualizar y entender mejor el comportamiento del sistema.

La creación de laboratorios virtuales tiene algunas ventajas importantes con respecto a los laboratorios reales. Dado que un laboratorio virtual se basa en modelos matemáticos que se ejecutan en ordenadores, su configuración y puesta a punto es mucho más sencilla que la configuración y puesta a punto de los laboratorios reales.

Además, presentan un grado de robustez y seguridad mucho más elevado ya que al no haber dispositivos reales éstos no pueden causar problemas en el entorno. Sin embargo, como inconveniente con respecto a los laboratorios reales cabe señalar que los laboratorios virtuales están limitados por el modelo y para poder ser manejables éstos tienden a simplificarse, con lo que se pierde información con respecto al sistema real. Además, la experimentación con sistemas reales siempre es un valor añadido para los alumnos.

Una forma interesante de utilizar los laboratorios virtuales es junto a los laboratorios reales (presenciales o remotos), de forma que los alumnos realizarían primero las prácticas en laboratorios virtuales, para pasar posteriormente, cuando el instructor lo considerase oportuno, al laboratorio real.

Así se consiguen varios objetivos importantes como son:

- Familiarizarse con el experimento: Evitando que los estudiantes puedan acudir al aula sin haber realizado trabajo previo.
- Optimizar el uso de los recursos: Los estudiantes requieren menos tiempo para realizar las prácticas, haciéndose un mejor uso de los laboratorios reales, tanto locales como remotos.
- Disminución del uso incorrecto del equipamiento: Frecuentemente los dispositivos utilizados en laboratorios reales son delicados, lo que se acentúa si se les hace trabajar fuera de las condiciones de trabajo para las que están diseñados.
- Comparación del comportamiento de modelos matemáticos frente a dispositivos reales: Los modelos matemáticos se obtienen simplificando el

comportamiento de los dispositivos reales, lo que puede producir comportamientos sensiblemente diferentes. Al tener la oportunidad de comparar ambos comportamientos, los alumnos pueden extraer conclusiones acerca de la importancia del modelado realizado.

- Formar en metodologías de trabajo: En su futura vida laboral los estudiantes habitualmente construirán primero modelos matemáticos de los sistemas que simularán bajo diferentes circunstancias como paso previo a construir prototipos, mucho más caros, con los que experimentar.
- Manejo de herramientas informáticas actuales: En la vida profesional, e incluso en la vida diaria, la destreza en el uso de las herramientas informáticas, sean del ámbito que sean, es un elemento diferenciador.

Con ello se consigue aportar al alumno una serie de conocimientos transversales que si bien pueden no ser el objetivo principal del laboratorio que se esté diseñando, le servirán en muchos ámbitos en el futuro.

- Repetitividad de los experimentos: Dado que el comportamiento de los sistemas a estudiar se obtiene mediante el modelado matemático de la realidad, los alumnos pueden repetir de forma totalmente fidedigna las condiciones bajo las que se realizaron los experimentos y reproducirlos ante el docente en caso de necesidad, con la seguridad de que el resultado será el mismo que ellos vieron en su momento.
- Multiplicidad de experimentos simultáneos: Como el único recurso necesario para este tipo de laboratorios es un ordenador, potencialmente todos y cada uno de los alumnos podría estar realizando simultáneamente su experimento sin interferir con sus compañeros, eliminando la necesaria secuencialidad que se da en los laboratorios reales.

Tipo de Laboratorio	Ventajas	Desventajas
Real	<ul style="list-style-type: none"> * Datos reales * Interacción con equipo real * Trabajo en colaboración * Interacción con el supervisor 	<ul style="list-style-type: none"> * Restricciones de tiempo y lugar. * Requiere la programación de horarios estrictos. * Es caro. * Requiere de supervisión.
Virtual	<ul style="list-style-type: none"> * Bueno para el entendimiento de conceptos. * No hay restricciones de tiempo ni lugar. * Es un medio interactivo. * Es de bajo costo. 	<ul style="list-style-type: none"> * Los datos son idealizados. * Falta de colaboración. * No hay interacción con equipo real.
Remoto	<ul style="list-style-type: none"> * Interacción con equipo real. * Datos reales. * No hay restricciones de tiempo ni lugar. * Costo medio. 	<ul style="list-style-type: none"> * Sólo hay presencia virtual en el Laboratorio.

Tabla 2. 5: Principales ventajas y desventajas de cada tipo de laboratorio

2.9. Software.

2.9.1. MatLab

Las herramientas computacionales han sido una gran ayuda para los ingenieros, ya que pueden tener un mejor entendimiento de conceptos aprendidos. Así mismo, se pueden realizar operaciones complejas en muy poco tiempo y tener más eficiencia al momento de hacer un trabajo. Matlab es una herramienta que fue creada por la compañía Mathworks, Inc. Ésta es considerada como un entorno de cálculo técnico, el cual brinda grandes presentaciones para el cálculo numérico y visualización de análisis numéricos, calculo matricial, procesamiento de señales y gráficos. Los usos más comunes de Matlab son:

- Cálculos matemáticos
- Desarrollo de algoritmos
- Modelado y simulación
- Análisis de datos, exploración y visualización
- Gráficas científicas y de ingeniería

El nombre de Matlab proviene de Matriz Laboratory, debido a que permite trabajar con facilidad y eficiencia con matrices, por lo mismo tiene un uso esencial tanto en industria como en universidades para el desarrollo de la investigación. Matlab es considerado un sistema interactivo, su elemento básico de datos es la matriz. Este lenguaje integra programación, visualización y cálculos en un ambiente sencillo. Este método resulta muy útil por la gran cantidad de los problemas como C, BASIC o FORTRAN tardarían mucho más tiempo. Matlab ofrece una gran ventaja sobre otras herramientas, su uso es muy práctico y fácil, no necesita el uso de programación tradicional, los problemas y las soluciones son expresados de la misma manera en que se escriben matemáticamente

Con el transcurso del tiempo Matlab ha mejorado y evolucionado notoriamente gracias a los programadores que han colaborado y contribuido a su desarrollo. Matlab se ha convertido parte importante en las universidades, por ser una herramienta indispensable para diferentes cursos como: Métodos numéricos, álgebra lineal, álgebra aplicada, ecuaciones diferenciales, por decir algunas áreas de ingeniería. Otro uso muy importante es en la industria para el desarrollo de este; mismo que en la ingeniería es muy útil para realizar y resolver los problemas que se van presentando en las diferentes áreas, como pueden ser de procesamiento de señales y de control.

Una de las herramientas principales son los llamados toolboxes, que proporcionan una serie de soluciones. Esto resulta de gran ayuda para los usuarios, ya que son funciones que extienden el entorno de Matlab. Pueden resolver problemas sobre procesamiento de señales, diseño de sistemas de control, simulación de sistemas dinámicos, identificación de sistemas, redes neuronales entre otros.

2.9.2. Simulink

SIMULINK es un paquete de software para modelar, simular y analizar sistemas

dinámicos. Soporta sistemas lineales y no lineales, modelados en tiempo continuo, muestreados o un híbrido de los dos. Los sistemas pueden ser también multifrecuencia, es decir, tienen diferentes partes que se muestrean o actualizan con diferentes velocidades.

Para modelar, Simulink proporciona una interfaz de usuario gráfica (GUI) para construir los modelos como diagramas de bloques, utilizando operaciones con el ratón del tipo pulsar y arrastrar. Con esta interfaz, puede dibujar los modelos de la misma forma que lo haría con lápiz y papel (o como lo representan la mayoría de los libros de texto). Esto es un cambio radical respecto a los paquetes de simulación previos que requieren que formule las ecuaciones diferenciales y las ecuaciones en diferencia en un lenguaje o programa. Simulink incluye una amplia biblioteca de bloques de sumideros, fuentes, componentes lineales y no lineales y conectores. Puede también personalizar y crear sus propios bloques.

Después de definir un modelo, puede simularlo utilizando cualquiera de los métodos de integración que tiene a su disposición o bien desde el menú de SIMULINK o introduciendo órdenes desde la ventana de órdenes de MATLAB. Los menús son apropiados para un trabajo interactivo; mientras que el enfoque de línea de orden es muy útil para ejecutar un lote de simulación. Los resultados de la simulación se pueden transferir al espacio de trabajo de MATLAB para su posterior post-procesamiento y visualización.

Las herramientas de análisis de modelo que incluyen linealización y determinación de estados estacionarios pueden ser accedidas desde la línea de orden de MATLAB, así como las muchas utilidades que MATLAB y sus toolboxes de aplicación poseen. Y como MATLAB y SIMULINK están integrados, pueden simular, analizar y revisar sus modelos en uno u otro entorno en cualquier momento.

Las diferentes bibliotecas que posee, permiten construir funciones y realizar el análisis del modelo de una manera sencilla. Para invocar a Simulink, basta

teclear en la pantalla de comandos:

```
>>simulink
```

A breves rasgos Simulink permite:

- Dibujar elementos y conexiones en una ventana gráfica. Las conexiones indican el recorrido de las señales de un elemento a otra.
- Los elementos se los extrae de la biblioteca del propio Simulink, e inclusive se pueden crear nuevos elementos.
- Los resultados se obtienen como salida de algunos elementos, pudiéndose almacenar, ver gráficamente, etc.
- Los datos o señales de entrada pueden obtenerse de salidas de variables especiales, del disco o de variables utilizadas en MATLAB.
- Simulink permite: la definición del modelo a trabajar y su análisis empleando la simulación. Para la definición del modelo, Simulink dispone de diferentes herramientas. Cada una de ellas dispone de bloques.

Tipos de bloques en SIMULINK

Los bloques de SIMULINK los podemos estructurar en las siguientes categorías:

- Sources: Entradas o fuentes de señales.
 - ✓ Constantes
 - ✓ Senoidales
 - ✓ Cuadradas
 - ✓ Escalón
 - ✓ Aleatorias

- Sinks: Salidas o dispositivos de visualización/almacenamiento de variables del sistema.
 - ✓ Osciloscopio
 - ✓ Fichero
 - ✓ Gráfico

- Discrete/Linear/Continuos: Representan sistemas sencillos mediante su relación entrada/salida.

- ✓ Discrete: sistemas discretos (muestreados). Dominio z.
 - ✓ Linear: Sistemas continuos lineales. Dominio s.
 - ✓ Continuos: Sistemas continuos no lineales. Dominio t.
- Signals&Systems: Se utilizan para conectar elementos o estructurar modelos.
- ✓ Subsistema: permite jerarquizar diseños
 - ✓ Multiplexadores/demultiplexadores: agrupan o desagrupan señales.
 - ✓ Memorias: permiten almacenar valores.
- Functions&Tables/Math: Funciones y elementos matemáticos
- ✓ Polinomios
 - ✓ Ganancias
- Blocksets/toolboxes: Elementos específicos para diferentes aplicaciones.
- ✓ Comunicaciones
 - ✓ Redes neuronales
 - ✓ Control

Todas estas categorías de bloques se muestran en la ventana inicial de SIMULINK como se puede apreciar en la figura 2.15. El procedimiento de trabajo para la creación de un modelo será seleccionar los elementos adecuados de entre los presentes en estas categorías, colocarlos sobre la ventana de diseño y establecer las conexiones entre ellos.

Para analizar un modelo de cualquier tipo, se empieza creando un archivo nuevo tipo "model" (elegir opción desde File...New), y se abren los distintos grupos de funciones a utilizar simplemente arrastrando con el mouse, al área de trabajo, los bloques deseados. Luego se procede a conectarlos uniendo sus entradas y salidas.

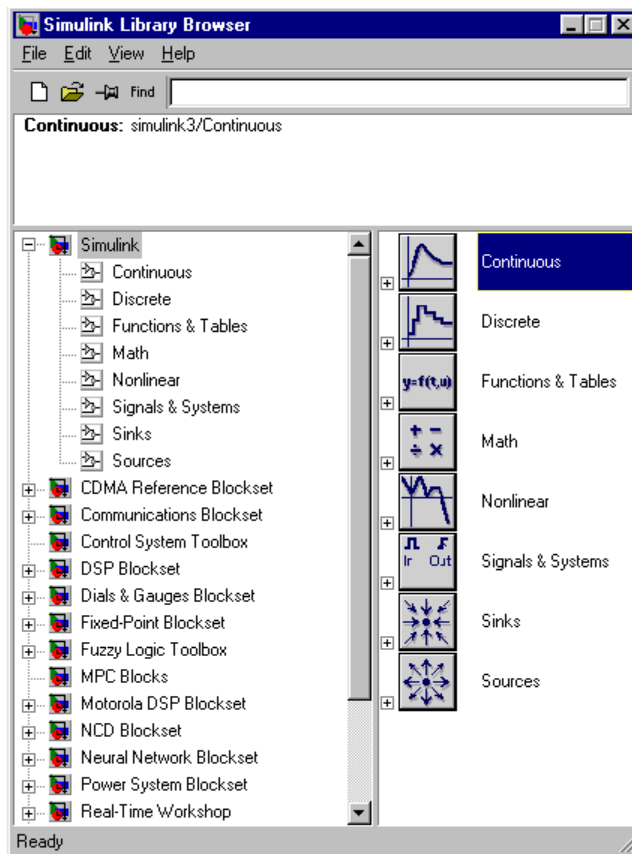


Figura 2. 15: Librerías de Simulink

Fuente: Matlab

Después se configuran los parámetros de cada bloque según el modelo y posteriormente se trabajará en el menú de simulación, con parámetros como el tiempo de inicio, tiempo de finalización; tipo de algoritmo de integración, etc. Finalmente, se inicia (start) la simulación. El progreso de la simulación se puede observar en la pantalla mientras ésta corre y al final, los resultados se pueden guardar en el espacio de trabajo de MATLAB creando archivos .mat de la misma manera para guardar la información y posteriormente analizarla o imprimirla.

CAPÍTULO III: FILTROS DIGITALES

3.1. Introducción.

Los filtros ideales son definidos como

$$H(\omega) = \begin{cases} \text{constante} & \text{para } \omega < \omega_0 & \text{(banda de paso)} \\ 0 & \text{para } \omega > \omega_0 & \text{(banda de rechazo)} \end{cases}$$

No son físicamente realizables ya que no son causales, sin embargo es posible diseñar filtros causales que pueden aproximarse a los ideales con tanta precisión como sea necesaria. Estos filtros causales presentaran rizado tanto en la banda de paso como en la banda de rechazo, por lo general. Por otro lado, la transición de frecuencia entre ambas bandas no se realizará de forma abrupta.

En MATLAB existen varias funciones que ayudan al diseño de un filtro indicando sus características. Para el diseño de filtros del tipo IIR o FIR se utilizan frecuentemente las funciones que se muestran en la tabla 3.1 a continuación:

Tabla 3. 1: Funciones más comunes para el diseño de filtros digitales en Matlab

Filtro Digital	Tipo	Función en MATLAB
IIR	Butterworth	[b,a] = butter(n,Wn,options) [z,p,k] = butter(n,Wn,options) [A,B,C,D] = butter(n,Wn,options)
	Chebyshev Tipo I	[b,a] = cheby1(n,Rp,Wn,options) [z,p,k] = cheby1(n,Rp,Wn,options) [A,B,C,D]=cheby1(n,Rp,Wn,options)
	Chebyshev Tipo II	[b,a] = cheby2(n,Rs,Wn,options) [z,p,k] = cheby2(n,Rs,Wn,options) [A,B,C,D]=cheby2(n,Rs,Wn,options)
	Elíptico	[b,a] = ellip(n,Rp,Rs,Wn,options) [z,p,k] = ellip(n,Rp,Rs,Wn,options) [A,B,C,D]=ellip(n,Rp,Rs,Wn,options)
FIR	Ventana	b = fir1(n,Wn,'ftype',window) b = fir2(n,f,m,npt>window)

Elaborado por: El Investigador

3.2. Filtros Digitales FIR

Los filtros digitales de Respuesta Finita Impulsiva o filtros FIR por sus siglas en inglés Finite Impulse Response, se trata de un tipo de filtros digitales en el que, como su nombre indica, si la entrada es una señal impulso la salida tendrá un número finito de términos no nulos. El modelo matemático de los filtros FIR también se fundamenta la ecuación de diferencias, pero con la particularidad de que todos los coeficientes a_k son iguales a cero. Se tiene entonces que la ecuación que los describe es función del conjunto de coeficientes b_k y de la secuencia de entrada $x(n)$.

$$y(n) = \sum_{k=0}^M b_k x(n - k)$$

Donde $M+1$ corresponde a la longitud del filtro. Este sistema considera sólo las últimas $M+1$ muestras de la señal de entrada y las pondera mediante coeficientes b_k . A este sistema se le denomina FIR, ya que su respuesta al impulso unitario (dada por los coeficientes b_k) es finita. Su diseño requiere la selección de la secuencia que mejor representa la respuesta a impulso de un filtro ideal. Los filtros FIR son siempre estables y son capaces de tener una respuesta lineal en fase.

La salida del filtro es una suma finita de la entrada presente y las M entradas previas al filtro. En los sistemas lineales se tiene que la salida $y(n)$ puede expresarse como la convolución de la señal de entrada con la función de respuesta al impulso del filtro.

$$y(n) = \sum_{k=0}^M h(k) * x(n - k)$$

Comparando esta última ecuación con (la anterior) se tiene que:

$$b_k = h(k)$$

Y por lo tanto los coeficientes del filtro son equivalentes a la respuesta al impulso del filtro. Al aplicar un impulso a la entrada del sistema, se obtiene a la salida una respuesta de longitud limitada.

Cuando se describe un sistema casual con respuesta finita al impulso, se usan normalmente estructuras no-recursivas. En tales casos la función del sistema tiene forma:

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

Donde se tiene que:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} b_n z^{-n}$$

Siendo $x(n)$ la entrada, $y(n)$ la salida $h(n)$ la función de respuesta al impulso y $X(z)$, $Y(z)$ y $H(z)$ sus respectivas transformadas Z.

3.3. Diseño De Filtros FIR

Para diseñar los filtros FIR se tienen que trincar la serie infinita de los coeficientes de Fourier en una serie finita. Si estos coeficientes son truncados, el filtro se verá afectado y se producirá un efecto Gibbs. Se producirán ondulaciones antes y después de cualquier discontinuidad. Estas ondulaciones no desaparecerán, sin importar que tan larga sea la serie, ya que siempre será finita. Por esto es que se emplea el método de las ventanas donde se usa un número finito de secuencias $w(n)$. Primero se debe buscar la transformada inversa de Fourier de la respuesta deseada. Después se aplica la ventana para aplanar los rizos.

3.3.1. Método de las ventanas

Este método permite obtener los coeficientes de un filtro realizable haciendo el producto de una función de truncamiento (con una cantidad finita de valores) con la respuesta impulsiva de un filtro ideal (que requeriría infinitos términos). A continuación se describen los efectos de la ventana en la respuesta en frecuencia del filtro.

- 1) La respuesta en frecuencia de la ventana presenta un gran lóbulo principal de altura proporcional a M y de anchura proporcional a $\frac{1}{M}$.
- 2) El ancho de banda de transición de $H(\omega)$ está dominado por el ancho del lóbulo principal ($\propto \frac{1}{m}$) a mayor M , menor el ancho de banda de transición.
- 3) Los lóbulos laterales $[\omega_p, \omega_c]$ son responsables del rizado tanto de la banda pasante como en la banda eliminada, por el rizado en ambas bandas tiene la misma forma, como se observa en la figura 6.1.

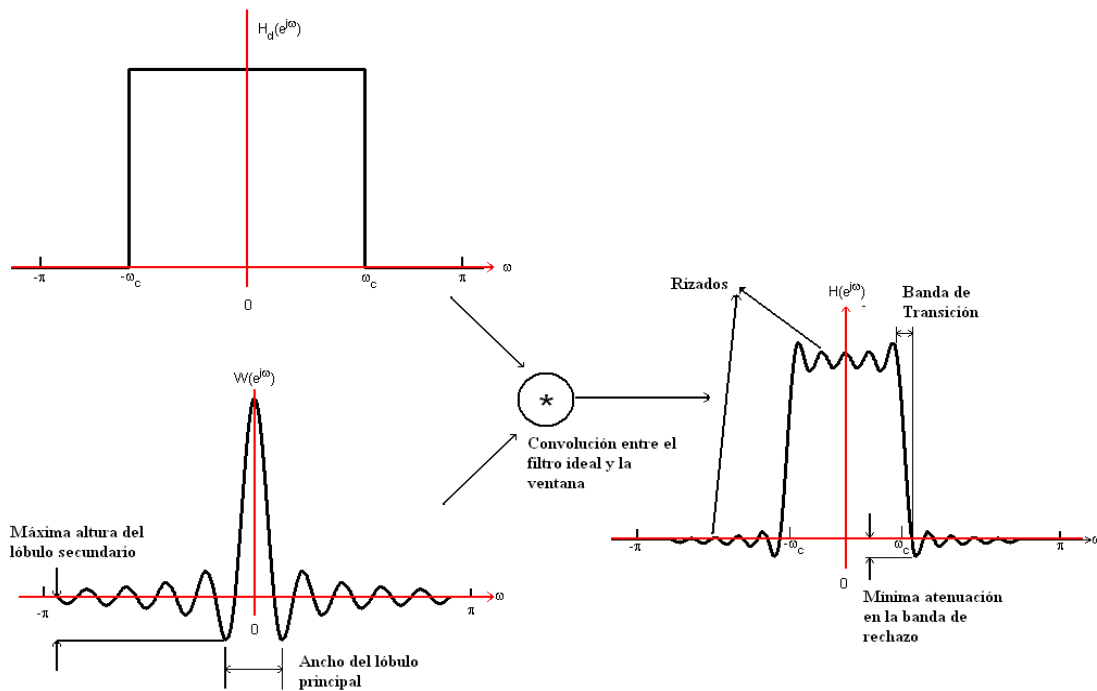


Figura 3. 1: Condiciones de una ventana de truncamiento

Fuente: <http://varoitus.barcelonamedia.org/rafael/Teaching/Courses/DSP/modulo7.pdf>

En el diseño de una ventana de truncamiento con longitud M dada, se busca que:

- 1) El ancho del lóbulo principal $\Delta\omega M$ sea mínimo.
- 2) La atenuación relativa del primer lóbulo lateral $AL1$ sea máxima.

3.3.2. Pasos para el diseño de filtros FIR por el método de la ventana

a. Especificaciones de Filtros FIR.

Los datos iniciales son las frecuencias de corte (o los anchos de banda) y los rizados. Las especificaciones de la respuesta espectral de un filtro se muestran en la tabla 3.2.

Tabla 3. 2: Especificaciones de la respuesta espectral de un filtro

Nombre	Expresión
Rizado de banda pasante	δ_1
Rizado de banda suprimida	δ_2
Frecuencia de corte de la banda suprimida	ω_p
Frecuencia de corte de la banda supresora	ω_s

Elaborado por: El Investigador

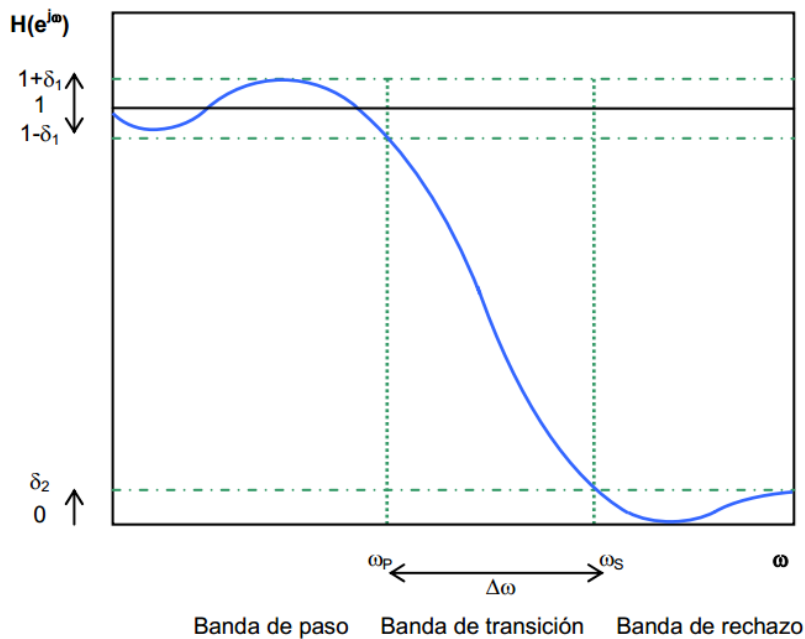


Figura 3. 2: Especificaciones de la respuesta espectral de un filtro

Fuente: <http://dsp1.materia.unsl.edu.ar/Filtros%20FIR%20Resumen.pdf>

De las anteriores especificaciones, se derivan los siguientes parámetros, con los cuales muchas veces se definen los filtros:

Tabla 3. 3: Especificaciones de la respuesta espectral de un filtro real derivadas de las anteriores

Nombre	Expresión
Ancho de la región de transición	$\Delta\omega = \omega_s - \omega_p$
Atenuación en la banda pasante	$A_p = 20 \times \log\left(\frac{1 + \delta_1}{1 - \delta_1}\right)$
Atenuación en la banda suprimida	$A_s = 20 \times \log(\delta_2)$

Elaborado por: El Investigador

b. Conversión de la banda pasante.

Se traducen los valores de frecuencias de corte al de un prototipo pasa bajos. Este paso es requerido en el cálculo manual dado que el inventanado es más fácil de operar con un prototipo pasa bajos, pero no es necesario en el cálculo mediante herramientas de programación.

c. Elección de la ventana.

Para cada ventana, la atenuación mínima en la banda suprimida A_s es un valor fijo, (véase la tabla 3.4), salvo Kaiser (se define el valor), y en función del mismo, se obtiene el resto de parámetros. La ventana a usar se elige considerando si posee atenuación en la banda suprimida A_s , mayor o igual que la de las especificaciones. En la tabla 3.5 se muestran los espectros de las ventanas.

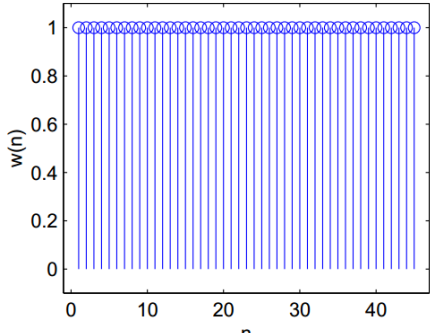
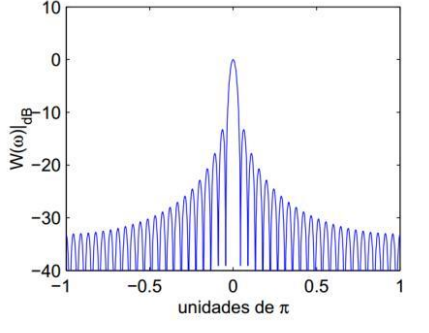
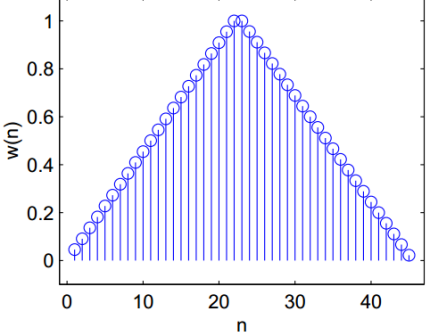
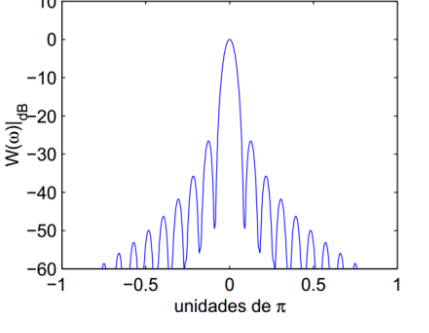
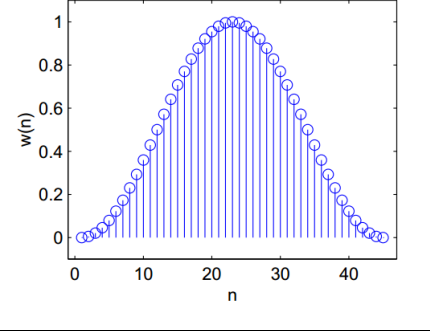
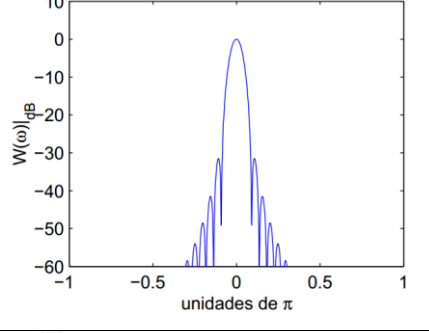
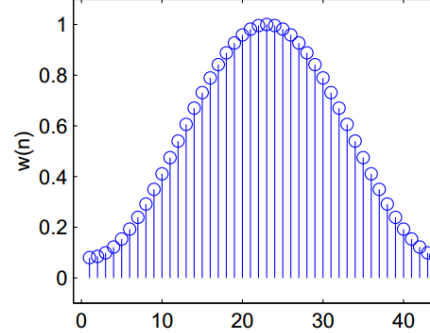
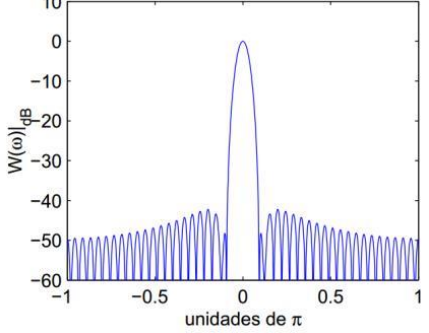
Tabla 3. 4: Ventanas utilizadas en las operaciones de ventaneo

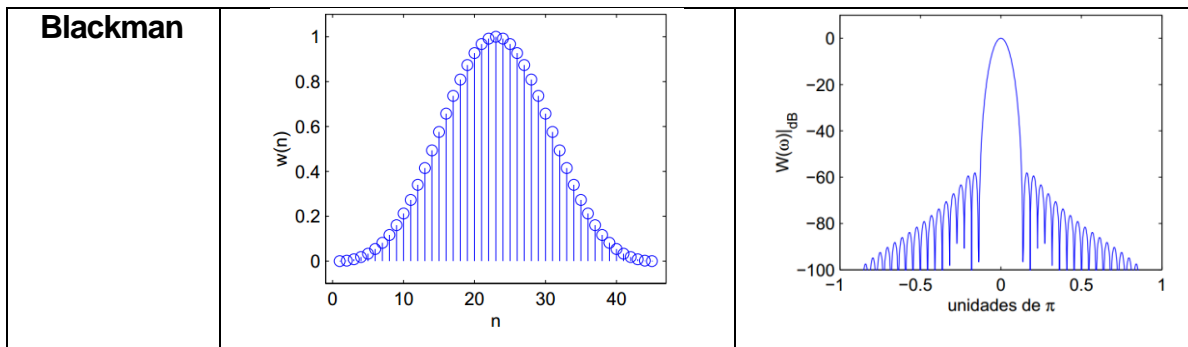
VENTANA	EXPRESIÓN	A_s
Rectangular	$w(n) = 1$ para $0 \leq n \leq N$ $w(n) = 0$ para otro valor	21dB
Barlett o Triangular	$w(n) = \frac{2n}{N}$ para $0 \leq n \leq \frac{N}{2}$ $w(n) = 2 - \frac{2n}{N}$ para $\frac{N}{2} \leq n \leq N$ $w(n) = 0$ para otro valor	25dB
Hanning	$w(n) = 0.5 - 0.5 \cos \frac{2\pi n}{N}$ para $0 \leq n \leq N$ $w(n) = 0$ para otro valor	44dB
Hamming	$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N}$ para $0 \leq n \leq N$ $w(n) = 0$ para otro valor	53dB
Blackman	$w(n) = 0.42 - 0.5 \cos \frac{2\pi n}{N} + 0.08 \cos \frac{4\pi n}{N}$ para $0 \leq n \leq N$ $w(n) = 0$ para otro valor	74dB
Kaiser	$w(n) = \frac{I_0 \left[\beta \sqrt{1 - \left(\frac{2n-N}{N} \right)^2} \right]}{I_0(\beta)}$ para $0 \leq n \leq N$ $w(n) = 0$ para otro valor Siendo I_0 la función de Bessel de primer tipo de orden cero	

Elaborado por: El Investigador

Fuentes: SOLIMAN, Samir Señales y Sistemas Continuos y Discretos

Tabla 3. 5: Espectros de las ventanas utilizadas en las operaciones de ventaneo

VENTANA	Espectro	
Rectangular		
Triangular		
Hanning		
Hamming		



Elaborado y Recopilado por: El Investigador

d. Estimación de Parámetros

En el caso de todas las ventanas, con excepción de la de Kaiser, el único parámetro que se debe estimar es la longitud del filtro L en función del ancho de transición $\Delta\omega$, o viceversa, ya que la atenuación en la banda suprimida A_s es fija. La relación entre N y $\Delta\omega$ se reproduce en la tabla 3.6. A modo comparativo, en la tabla también se dan los valores que tendrían los parámetros β y $\Delta\omega$ para un filtro de Kaiser con las mismas características.

Tabla 3. 6: Estimación de parámetros de las ventanas comunes versus la ventana Kaiser

Ventana	Anchura del lóbulo principal de la ventana	$\Delta\omega$ exacto	β Kaiser equivalente
Rectangular	$\frac{4\pi}{N}$	$\frac{1.8\pi}{N}$	0
Barlett	$\frac{8\pi}{N}$	$\frac{6.1\pi}{N}$	1.33
Hanning	$\frac{8\pi}{N}$	$\frac{6.2\pi}{N}$	3.66
Hamming	$\frac{8\pi}{N}$	$\frac{6.6\pi}{N}$	4.86
Blackman	$\frac{12\pi}{N}$	$\frac{11\pi}{N}$	7.04

Elaborado por: El Investigador

Fuentes: Oppenheim, Alan V, Tratamiento de señales en tiempo discreto.

En el caso de Kaiser, la atenuación A_s no es fija, sino que puede ajustarse a un valor deseado. En función de dicho valor, se determina un parámetro auxiliar llamado β .

$$\beta = \begin{cases} 0 & \text{para } A_s < 21 \\ 0.5842(A_s - 21)^{0.4} + 0.07886(A_s - 21) & \text{para } 21 \leq A_s \leq 50 \\ 0.1102(A_s - 8.7) & \text{para } 50 \leq A_s \end{cases}$$

Con este parámetro, se calcula la longitud del filtro en función del ancho de transición, o viceversa.

$$\Delta\omega = \frac{A_s - 8}{2.285N}$$

e. Enventanado

El primer paso es el cálculo de la respuesta impulsiva del filtro ideal deseado $h_d(n)$.

Tabla 3. 7: Respuesta Impulsiva del filtro ideal deseado $h_d(n)$

Respuesta Impulsiva	Respuesta espectral
$h_d(n) = \frac{\sin\left[\omega_c\left(n - \frac{N}{2}\right)\right]}{\pi\left(n - \frac{N}{2}\right)}$	$H_d\left(e^{j\omega\frac{N}{2}}\right) = \begin{cases} e^{-j\omega\frac{N}{2}} & \text{si } \omega \leq \omega_c \\ 0 & \text{si } 0 < \omega \end{cases}$

Elaborado por: El investigador

Luego se calculan los valores de la función ventana $w[n]$, de acuerdo a la tabla 3.7. Los coeficientes del filtro enventanado se calculan haciendo el producto entre la respuesta impulsiva del filtro ideal y la función ventana. En el dominio de la frecuencia, el enventanado consiste en hacer la convolución de la respuesta espectral del filtro ideal deseado con la transformada de la función ventana, como se muestra en la tabla 3.8.

Tabla 3. 8: Coeficientes del filtro ideal

Respuesta Impulsiva	Respuesta espectral
$h(n) = h_d(n)w(n)$	$H(e^{j\omega}) = H_d e^{j\omega} * W e^{j\omega}$

Elaborado por: El Investigador

3.3.3. Funciones en MatLab para Filtros FIR.

MATLAB tiene varias funciones para el diseño de filtros FIR, una de estas funciones es `fir1` que puede tener los siguientes parámetros:

- ✓ `b = fir1 (n,Wn)`
- ✓ `b = fir1 (n,Wn,'ftype')`
- ✓ `b = fir1 (n,Wn>window)`
- ✓ `b = fir1 (n,Wn,'ftype', window)`

donde `b` es un vector con los coeficientes del filtro, `n` es el orden del filtro, `Wn` es la frecuencia de corte normalizada con respecto a la frecuencia de Nyquist y varía de $0 < Wn < 1$; „*ftype*” especifica el tipo de filtro, esto es „high” pasa altas con frecuencias de corte Wn ; „stop” paso banda $Wn = [w1,w2]$, por defecto se especifica como un filtro tipo paso bajas; *window* especifica el tipo de ventana a utilizar en el diseño. Por defecto utiliza la ventana Hamming.

MATLAB tiene las siguientes funciones para obtener los puntos correspondientes a una ventana específica.

- ✓ `win= hamming (N)`
- ✓ `win= hann (N)`
- ✓ `win= blackman (N)`
- ✓ `win= kaiser (N, β)`

Donde `win` es un vector con los valores de la ventana, `N` es el número de puntos de la ventana.

3.4. Filtros IIR

Son sistemas cuya salida depende además de salidas anteriores y que, estando en reposo, al ser estimulados con una entrada impulsional su salida no vuelve al reposo, de ahí el calificativo de filtros de respuesta impulsional infinita IIR. La ecuación en diferencias general es de la forma:

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_Mx(n-M) - a_1y(n-1) - a_2y(n-2) \\ - \dots - a_Ny(n-N) = \sum_{k=0}^M b_kx(n-k) - \sum_{k=1}^N a_ky(n-k)$$

Donde el orden es igual al máximo de M y N. La función de transferencia en Z del filtro es:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}}$$

De esta caracterización se obtienen polos y ceros, los cuales dependen de la elección de los parámetros del sistema y de las características de la respuesta en frecuencia del sistema. El diseño de filtro IIR está estrechamente relacionado al diseño de filtros analógicos. Usualmente se diseña un filtro analógico, para luego llevar a cabo su transformación en el dominio digital. Existen dos métodos de transformación de invarianza en el impulso y el método de la transformación bilineal.

3.4.1. Diseño De Filtros IIR Mediante Transformadas Bilineales

Los filtros recursivos pueden ser diseñados por varios métodos, siendo el más común el basado en las transformaciones bilineales. Este procedimiento requiere del conocimiento de la función de transferencia en el tiempo continuo del filtro a diseñar. Los coeficientes del filtro en dominio s son transformados a uno equivalente en el dominio z. Los coeficientes de la discretización formarán el filtro IIR.

3.4.2. Pasos para el diseño de filtros IIR mediante transformadas bilineales:

a. Especificaciones:

Los datos iniciales son las frecuencias de corte (o los anchos de banda) y los rizados. Las especificaciones de la respuesta espectral de un filtro se muestran en la tabla 3.9.

Tabla 3. 9: Especificaciones del espectro para filtros IIR

$R_p = -10 \log_{10} \frac{1}{1 + \epsilon^2}$	$\epsilon = \sqrt{10^{\frac{R_p}{10}} - 1}$
$A_s = -10 \log_{10} \frac{1}{A^2}$	$A = 10^{\frac{A_s}{20}}$
$\frac{1 - \delta_1}{1 + \delta_1} = \sqrt{\frac{1}{1 + \epsilon^2}}$	$\epsilon = \frac{2\sqrt{\delta_1}}{1 - \delta_1}$
$\frac{\delta_2}{1 + \delta_1} = \frac{1}{A}$	$A = \frac{1 + \delta_1}{\delta_2}$

Fuente: <http://www.ing.uc.edu.ve/~azozaya/docs/DSP/dfiir.pdf>

En la figura 3.3 se muestra la respuesta de magnitud cuadrática y en la tabla 3.10 se muestran las especificaciones de la respuesta de magnitud cuadrática.

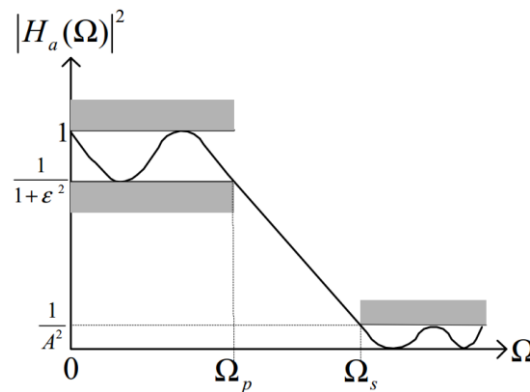


Figura 3. 3: Respuesta de magnitud cuadrática

Fuente: <http://www.ing.uc.edu.ve/~azozaya/docs/DSP/dfiir.pdf>

Tabla 3. 10: Especificaciones de la respuesta de magnitud cuadrática.

$\frac{1}{1 + \epsilon^2} \leq H_a(\Omega) ^2 \leq 1$	$ \Omega \leq \Omega_p$
$0 \leq H_a(\Omega) ^2 \leq \frac{1}{A^2}$	$\Omega_s \leq \Omega $
$ H_a(\Omega) ^2 = \frac{1}{1 + \epsilon^2}$	$\Omega = \Omega_p$
$ H_a(\Omega) ^2 = \frac{1}{A^2}$	$\Omega = \Omega_s$

Fuente: <http://www.ing.uc.edu.ve/~azozaya/docs/DSP/dfiir.pdf>

Donde ϵ es parámetro de rizado, Ω_p es la frecuencia de corte de la banda pasante en rad/s, Ω_s es la frecuencia de corte de la banda eliminada, igualmente en rad/s, y A es un parámetro de atenuación de la banda eliminada.

b. Elección de la Transformación Bilineal

Al elegir el método de transformación bilineal, se deberá superar el efecto “aliasing”, ocasionado por la respuesta en frecuencia analógica que contiene componentes que están en, o más allá de la Frecuencia de Nyquist. La transformada bilineal (véase la figura 3.4), es un método de compresión del infinito, directamente desde el eje de frecuencia analógica a una de longitud finita, la cual se reforma alrededor del círculo unitario una sola vez. Algunas veces se la conoce como deformación en frecuencia (frequency warping). Esta introduce una distorsión en la frecuencia. Se trata de deshacer por medio de pre-deformaciones las frecuencias críticas del filtro análogo (frecuencia de corte y frecuencia central) tal como cuando los filtros analógicos se transforman en filtro digitales, el filtro digital diseñado deberá cumplir con ciertas especificaciones deseadas.

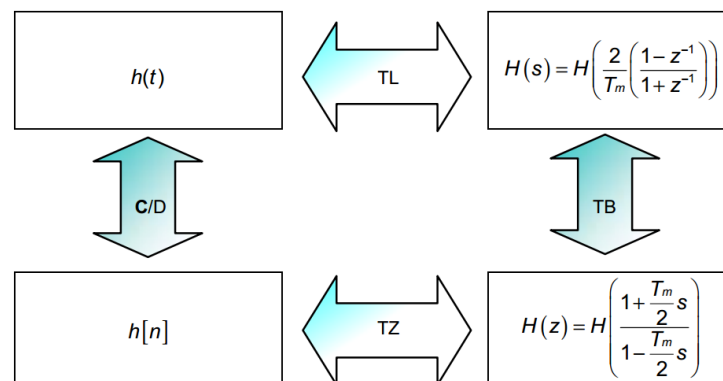


Figura 3. 4: Transformación bilineal

Fuente: <http://dsp1.materia.unsl.edu.ar/Filtros%20IIR%20Resumen.pdf>

c. Elección del filtro analógico

➤ Filtros Butterworth.

Los filtros Butterworth son causales por naturaleza y de varios niveles de orden, los de más bajo orden son lo mejor (lo más corto posibles) en el dominio del tiempo, y los de orden más alto son lo mejor en el dominio de la frecuencia. Los filtros Butterworth o máximamente planos tienen una respuesta en frecuencia de amplitud monótonica la cual es máximamente plana cuando la respuesta en frecuencia es igual a cero, como se observa en la figura 3.5. Los filtros Butterworth tienen un mínimo desplazamiento en fase comparados con otro tipo de filtros convencionales.

$$\overline{B(\omega)}B(\omega) = \frac{1}{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}}$$

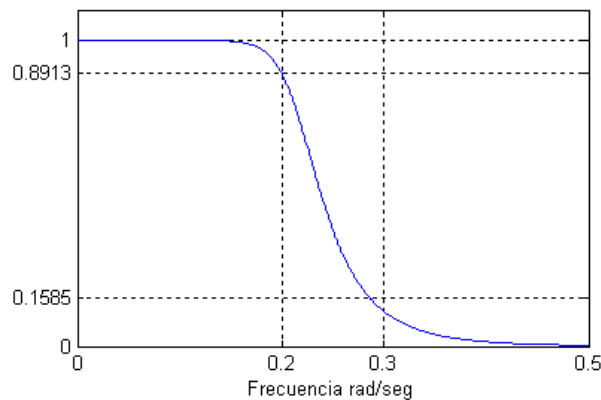


Figura 3. 5: Respuesta en frecuencia de un filtro Butterworth
Elaborado por: El Investigador

➤ Filtros Chebyshev

Los filtros Chebyshev son de dos tipos: los filtros Chebyshev I tienen todos los polos del filtro los cuales están equiripple en la banda de paso y son monotónicos en la banda de rechazo, tal como se ve en la figura 3.6.

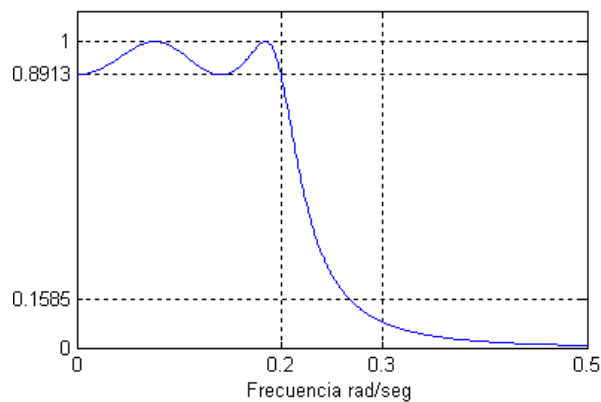


Figura 3. 6: Respuesta en frecuencia de un filtro Chebyshev I
Elaborado por: El Investigador

Los filtros Chebyshev II contienen polos y ceros exhibiendo un comportamiento monotónico en la banda pasante y equiripple en la banda de rechazo, como se observa en la figura 3.7.

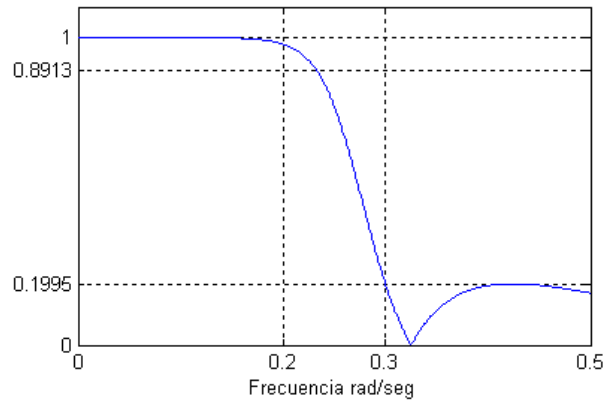


Figura 3. 7: Respuesta en frecuencia de un filtro Chebyshev II.

Elaborado por: El Investigador

La respuesta en frecuencia de este filtro está dada por:

$$|H(\Omega)|^2 = \left(1 + \varepsilon^2 T_N^2 \left(\frac{\Omega}{\Omega_p} \right) \right)^{-1}$$

Donde T_N es un parámetro relacionado al rizo presente en la banda pasante.

$$T_N = \frac{\cos(N \cos^{-1}(x)) \quad |x| \leq 1}{\cosh(N \cosh^{-1}(x)) \quad |x| \geq 1}$$

➤ Filtros Elípticos.

Los filtros elípticos son caracterizados por ser igualmente rizados en ambos lados de la banda de paso y de rechazo, como se aprecia en la figura 3.8, éste nos provee una realización con el orden más bajo para ciertas condiciones particulares establecidas.

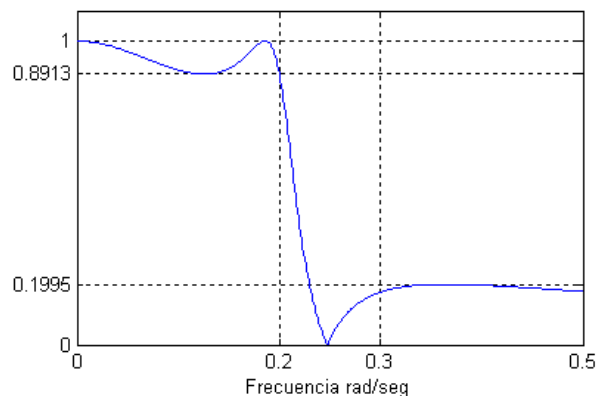


Figura 3. 8: Respuesta en frecuencia de un filtro Elíptico.

Elaborado por: El Investigador

d. Transformaciones de frecuencia.

Esta es una de las mayores técnicas empleadas en el diseño de filtros. Se puede diseñar un filtro pasa-bajos analógico o digital y luego este se transforma en un filtro digital pasa-altos o pasa-banda.

e. Transformaciones de frecuencia analógica.

Las transformaciones en frecuencia que pueden ser usadas para obtener un filtro pasa-altas, pasa-bajos, pasa-banda o rechaza banda se pueden observar en la tabla 3.11.

Aquí $\Omega_0^2 = \Omega_1 * \Omega_2$ la cual es definida como la frecuencia de corte para un filtro pasa-bajos o pasa-altos y la frecuencia central para los filtros en la banda de paso y de rechazo.

$$|H(j\Omega)| = \frac{\Omega_0}{\Omega_2 - \Omega_1}$$

Donde Ω_2 y Ω_1 son la más alta y la más baja frecuencia de corte respectivamente. $\Omega_2 - \Omega_1$ dan como resultado el ancho de banda.

Tabla 3. 11: Relación entre los tipos de filtros y las transformaciones de frecuencia analógica.

Tipo de filtros	Transformación
Pasa-bajos	$S = \frac{s}{\Omega_c}$
Pasa-altas	$S = \frac{\Omega_c}{s}$
Pasa-banda	$S = \frac{Q(s^2 + \Omega_0^2)}{\Omega_0 s}$

Fuente: SOLIMAN, Samir Señales y Sistemas Continuos y Discretos

CAPÍTULO IV: DESARROLLO EN MATLAB DE FILTROS DIGITALES

4.1. Diseño de filtros usando MATLAB

El Toolbox de MATLAB dedicado al procesamiento de señales, incluye algunas funciones útiles para diseñar ambas clases de filtros digitales IIR así como también para el diseño tradicional de filtros analógicos. Los filtros considerados fueron los filtros Butterworth, Chebyshev I & II, y los filtros Elípticos, como se puede observar en la tabla 4.1.

Tabla 4. 1: Funciones para determinar los coeficientes del filtro IIR

Butterworth	[b,a] = butter (N,Wc,options) [z,p,k] = butter (N,Wc,options)	B y A son los coeficientes del numerador y del denominador respectivamente, N es el orden del filtro(calculado previamente) y Wn es la frecuencia de corte.
Chebyshev Tipo I	[b,a] = cheby1 (N,Wc,options) [z,p,k] =cheby1(N,Wc,options)	Se deben especificar el orden del filtroN, el rizado de pasabanda permitido R y la frecuencia de corte normalizada con respecto a la frecuencia de Nyquist.
Chebyshev Tipo II	[b,a] = cheby2 (N,Wc,options) [z,p,k] =cheby2 (N,Wc,options)	Se deben especificar el orden del filtro N, el rizado de parabanda permitido R y la frecuencia de corte normalizada con respecto a la frecuencia de Nyquist.
Elliptic	[b,a] = ellip (N,Wc,options) [z,p,k] = ellip (N,Wc,options)	Rp y Rs son los rizados de pasabanda y parabanda.

Fuente: MATLAB Elaborado por: El Investigador

En la tabla 4.2 se observa las funciones que calculan el orden N mínimo de un filtro determinado dado un conjunto de especificaciones.

Tabla 4. 2: Funciones para determinar el orden mínimo de un filtro IIR

Butterworth	[N,Wn] = buttord (Wp, Ws, Rp, Rs)
Chebyshev Tipo I	[N,Wn] = cheb1ord (Wp, Ws, Rp, Rs)
Chebyshev Tipo II	[N,Wn] = cheb2ord (Wp, Ws, Rp, Rs)
Elliptic	[N,Wn] = ellipord (Wp, Ws, Rp, Rs)

Fuente: MATLAB

Elaborado por: El Investigador

Para el presente capítulo se detalla la parte práctica de filtros digitales que deberá ser utilizado en la asignatura de procesamiento digital de señales. Cada práctica consta de dos partes:

a. Simulación utilizando MATLAB

La parte de simulación es necesaria para repasar los conocimientos adquiridos durante el módulo de Procesamiento Digital de Señales, específicamente lo referente a Filtros Digitales. Asimismo, el estudiante será capaz de aplicar estos conocimientos en la solución de los ejercicios propuestos. Después el estudiante deberá analizar cada uno de los resultados obtenidos y realizar sus propias conclusiones. El M-file de este programa es el siguiente:

```
function varargout = Presentacion(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename,...
'gui_Singleton', gui_Singleton,...
'gui_OpeningFcn', @Presentacion_OpeningFcn,...
'gui_OutputFcn', @Presentacion_OutputFcn,...
'gui_LayoutFcn', [] ,...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Presentacion_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = Presentacion_OutputFcn(hObject, eventdata, handles
varargout{1} = handles.output;
function FIR_Callback(hObject, eventdata, handles) FIR
function IIR_Callback(hObject, eventdata, handles) IIR
```

4.2. Desarrollo de las Prácticas de Filtros FIR

Para realizar las prácticas, se elaboró la pantalla que se muestra en la figura 4.1 a través de GUI de MatLab, este programa nos permitirá diseñar los filtros tipo FIR de acuerdo a lo explicado en el capítulo 3.

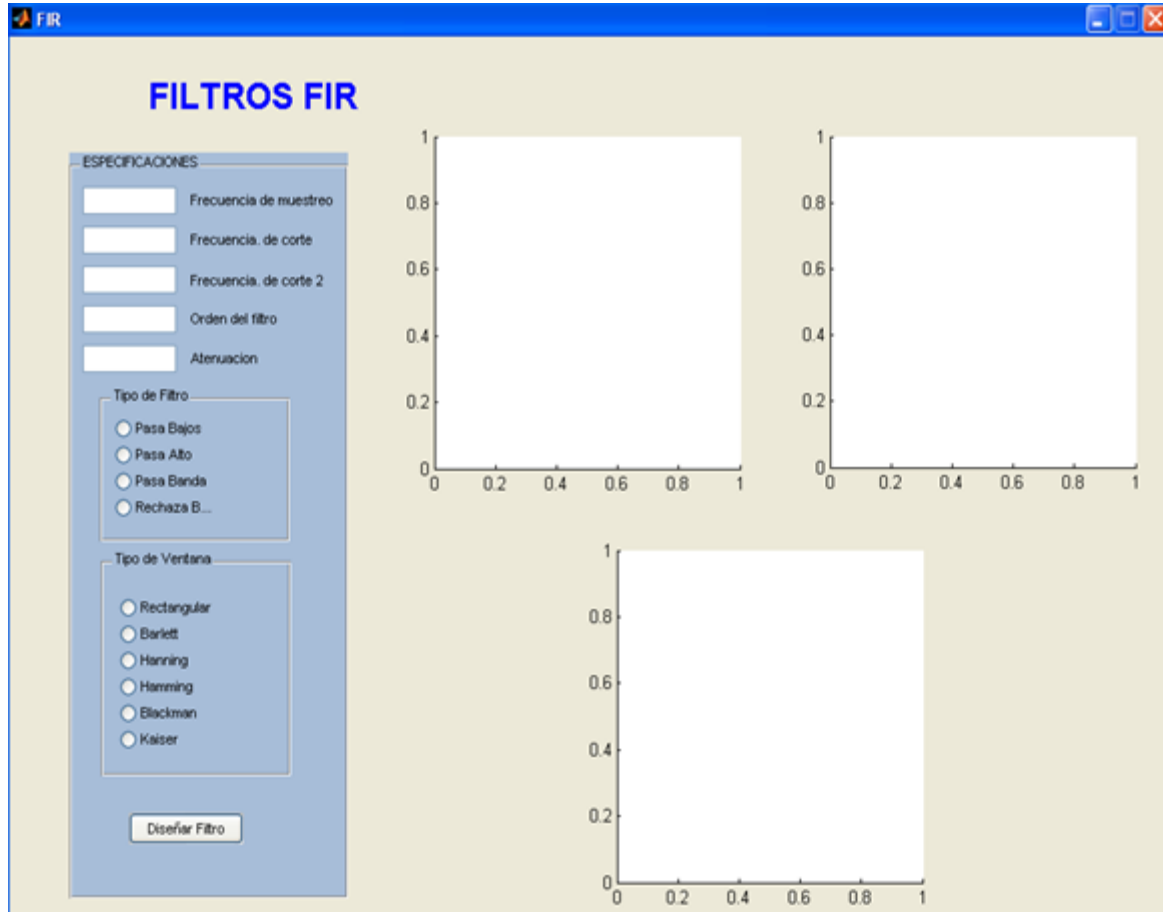


Figura 4. 1: Interfaz para el diseño de Filtros FIR

Elaborado por: El Investigador

La interfaz que se muestra en la figura 4.1 dispone de tres paneles, el primero corresponde a las “Especificaciones” en la que se indican las variables tales como frecuencia: de muestreo, de corte, de corte 2, el orden del filtro y la atenuación respectiva; en el segundo es para seleccionar el tipo de filtro tales como: pasa bajo, pasa alto, pasa banda y rechazo de banda; y en el tercer panel nos indica el tipo de ventana que puede ser del tipo: Rectangular, Barlett, Hanning, Hamming, Blackman y Kaiser.

En otras palabras, los alumnos que evalúen estos tipos de filtros FIR, bastará con

que se introduzcan los parámetros del filtro deseado, seleccionar correctamente el tipo y ventana de Filtros FIR. Finalmente, y para todos los Filtros FIR simplemente debemos seleccionar Diseñar Filtro, para así obtener las gráficas correspondientes a la respuesta espectral, impulsiva y fase del filtro, a continuación se muestra el código fuente o programación en MatLab que permite el correcto funcionamiento de los Filtros FIR:

```
function varargout = FIR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @FIR_OpeningFcn, ...
'gui_OutputFcn', @FIR_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function FIR_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = FIR_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function Fmuestreo_Callback(hObject, eventdata, handles)
function Fmuestreo_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```

function Fcorte_Callback(hObject, eventdata, handles)
function Fcorte_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function Orden_Callback(hObject, eventdata, handles)
function Orden_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function atenuacion_Callback(hObject, eventdata, handles)
function atenuacion_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function grafico_Callback(hObject, eventdata, handles)

%Se limpia el workspace de MATLAB y las variables en uso. clc;
%*****

Fm = str2double(get(handles.Fmuestreo,'String'));%f de muestreo
Fc = str2double(get(handles.Fcorte,'String'));% f de corte
N = str2double(get(handles.Orden,'String'));% orden
A = str2double(get(handles.atenuacion,'String'));% atenuacion
Fc2 = str2double(get(handles.Fcorte2,'String'));
puntos=512; %cantidad de puntos para el grafico

%Cálculo de parámetros
wc=2*pi*Fc/Fm; %pulsacion digital de corte
    vc=2*Fc/Fm;%frecuencia relativa a Fm/2 usada por Matlab de corte fcb = 2*pi*Fc/Fm;
fcb1 = 2*pi*Fc2/Fm;

```



```

Fcb = 2*Fc/Fm; Fcb1=2*Fc2/Fm;
Vc = [Fcb Fcb1];
win = handles.win;
tipo = handles.clase;

%Coeficientes de los filtros enventanados if tipo == 1
b=fir1((N-1),vc,win);
elseif tipo == 2
b=fir1((N-1),vc,'high',win);
elseif tipo == 3
b=fir1((N-1),Vc,'bandpass',win);
else
b=fir1((N-1),Vc,'stop',win);
end

%Respuesta espectral de los filtros enventanados
[H,w]=freqz(b,1,puntos);
%Respuesta espectral del filtro ideal
Hideal=rectpuls(w,2*wc);
%Graficos de la respuesta espectral
axes(handles.axes1) plot(w,abs(H),'b');title('Respuesta Espectral') xlabel('\omega,rad/muestra')
ylabel('H(e^j\omega)')
%Graficos de la respuesta impulsiva
axes(handles.axes2)
stem(b,'b'); title('Respuesta Impulsiva')
xlabel('n')
ylabel('h[n]')

%Graficos de la fase
axes (handles.axes4) plot(w/pi,unwrap(angle(H))); title('Respuesta de fase del filtro'); xlabel('\omega en
unidades de \pi'); ylabel('Fase (rads)');
grid on;

function kaiser_ButtonDownFcn(hObject, eventdata, handles)
function PB_Callback(hObject, eventdata, handles)
% ----- function Ventana_SelectionChangeFcn(hObject,
eventdata, handles)
A =str2double(get(handles.atenuacion,'String'));

```

```

%parametro beta de Kaiser if(A<=21)
beta=0;
elseif (A>=50)
    beta=0.1102*(A-8.7);
else
    beta=0.5842*(A-21)^0.4+0.07886*(A-21);
end
N=str2double(get(handles.Orden,'String'));

if hObject == handles.rectangular
    ventana = window(@boxcar,N);
elseif hObject == handles.hann_ventana = window(@hann,N);
elseif hObject == handles.hamming_ventana = window(@hamming,N);
elseif hObject == handles.bartlett_ventana = window(@bartlett,N);
elseif hObject == handles.blackman_ventana = window(@blackman,N);
else
    ventana = window(@kaiser,N,beta);
end

handles.ven = ventana;
guidata(hObject,handles)
function Tipo_Callback(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% -----
function tipofiltro_Selection ChangeFcn(hObject, eventdata, handles)
if hObject == handles.PB
    fil = 1;
elseif hObject == handles.PA
    fil = 2;
elseif hObject == handles.Pbanda fil = 3;
else
    fil = 4;
end

```

```

handles.clase = fil;
guidata (hObject,handles)
function Fcorte2_Callback(hObject, eventdata, handles)
function Fcorte2_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function axes4_CreateFcn(hObject, eventdata, handles)

```

4.2.1. Experiencia #1: Filtro FIR Pasa-Bajo.

Analizar la variación del rizado en la respuesta espectral para el caso de un filtro FIR pasa-bajos con ventana de Hanning, en la figura 4.2 se muestran las especificaciones tanto para las frecuencias de muestreo, corte, el orden del filtro y atenuación.

ESPECIFICACIONES

30000 Frecuencia de muestreo

4500 Frecuencia de corte

Frecuencia de corte 2

30 Orden del filtro

60 Atenuación

Tipo de Filtro

- Pasa Bajos
- Pasa Alto
- Pasa Banda
- Rechaza B...

Tipo de Ventana

- Rectangular
- Barlett
- Hanning
- Hamming
- Blackman
- Kaiser

Diseñar Filtro

Figura 4. 2: Panel de especificaciones para el diseño de Filtros FIR pasa bajo.

Elaborado por: El Investigador

Una vez ingresados los valores correspondientes a las especificaciones mostrada por la figura 4.2, oprimiremos “Diseñar Filtro”, para lo cual se visualizarán la respuesta espectral, impulsiva y de fase del filtro FIR, tal como se muestra en la figura 4.3.

De la figura 4.3 podemos describir que la primera gráfica corresponde a un filtro pasa bajos, cuya frecuencia de corte es $0,94 \text{ rad/muestra}$; la segunda gráfica podemos ver que la respuesta impulsional tiene un ancho de transición muy pequeño mientras que en el lóbulo lateral la atenuación es alta. Es decir, que estos valores satisfacen al diseño del filtro FIR.

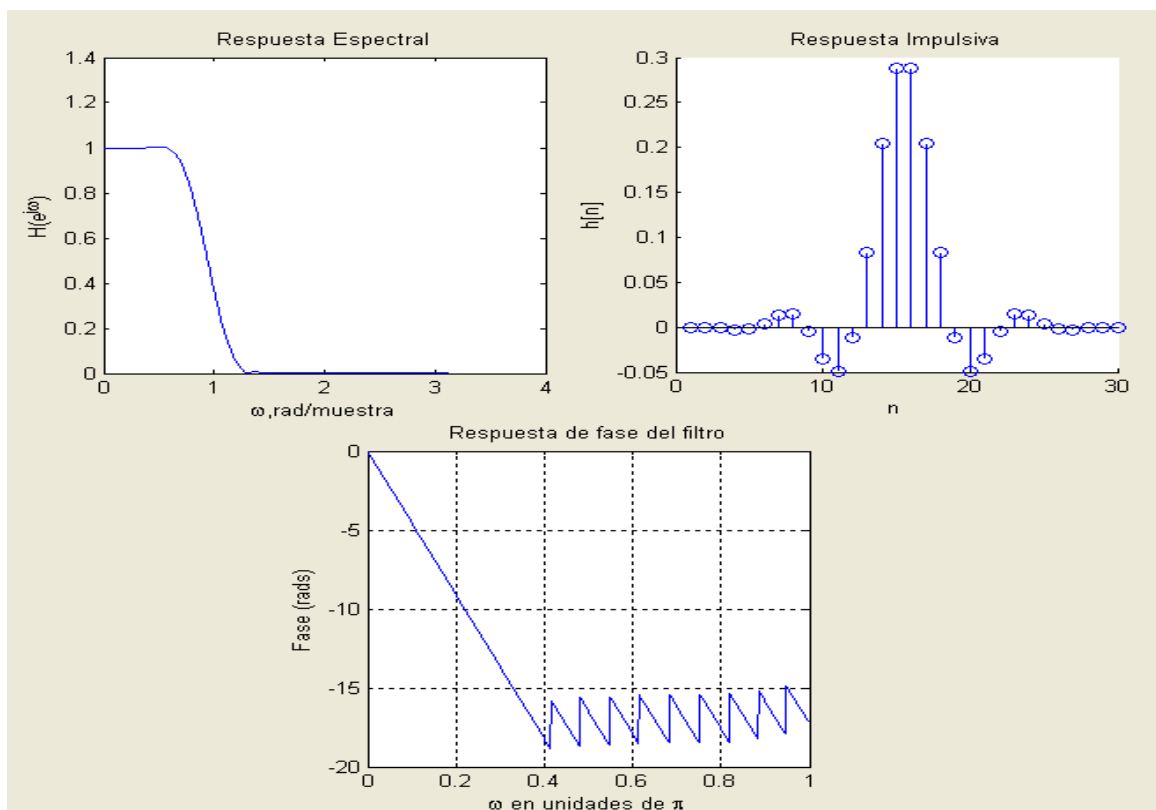


Figura 4. 3: Gráficos de Respuestas Espectrales para Filtros FIR Pasa Bajo.

Elaborado por: El Investigador

4.2.2. Experiencia #2: Filtro FIR Pasa-Alto.

Para esta experiencia práctica analizaremos un Filtro FIR Pasa Alto, específicamente la varianza de rizado de respuesta espectral, en la figura 4.4 se muestran los valores ingresados en las especificaciones y el tipo de ventana elegido es Kaiser. En la misma figura 4.4 se muestran cada una de las gráficas

correspondientes a las respuestas espectral, impulsiva y de fase del filtro FIR Pasa Alto con ventana Kaiser, como podemos ver la primera gráfica corresponde a un filtro pasa alto cuya frecuencia de corte es $2,01 \text{ rad/muestra}$; la segunda gráfica se aprecia la respuesta impulsional con ancho de transición pequeño y atenuación grande del lóbulo lateral. Para concluir, las gráficas cumplen los requerimientos de los filtro FIR pasa altos con ventana Kaiser.

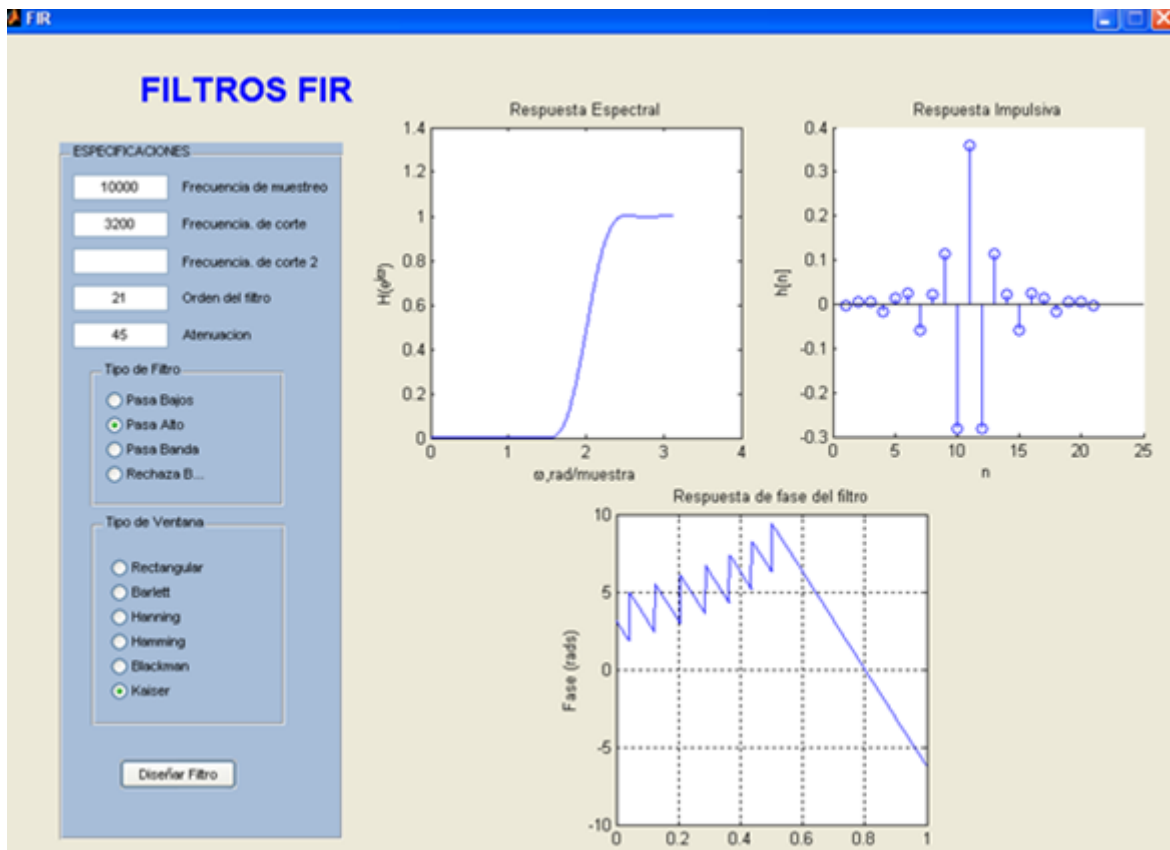


Figura 4. 4: Gráficos de Respuestas Espectrales para Filtros FIR Pasa Alto.

Elaborado por: El Investigador

4.2.3. Experiencia #3: Filtro FIR Pasa-Banda 1.

La presente experiencia consiste en el análisis de la varianza de rizado de la respuesta espectral de los filtros FIR pasa banda con el tipo de ventana Barlett. En la figura 4.5 se muestran las especificaciones a utilizar para el filtro FIR en mención. Una vez introducidos los valores para la interfaz Filtro FIR, damos click en Diseñar Filtro, en la cual se muestran las gráficas de las respuestas: espectral, impulsiva y de fase.

De la figura 4.5 se puede concluir lo siguiente: la primera gráfica es la respuesta

espectral, en la cual las frecuencias de corte se encuentran en $1,0 \text{ rad/muestra}$ y $2,4 \text{ rad/muestra}$; así mismo se observa que la respuesta espectral está muy distorsionada; mientras que para la respuesta impulsional el ancho de transición es muy elevado, lo cual nos indica que la ventana Barlett no es óptima en el diseño de este filtro FIR pasa banda.

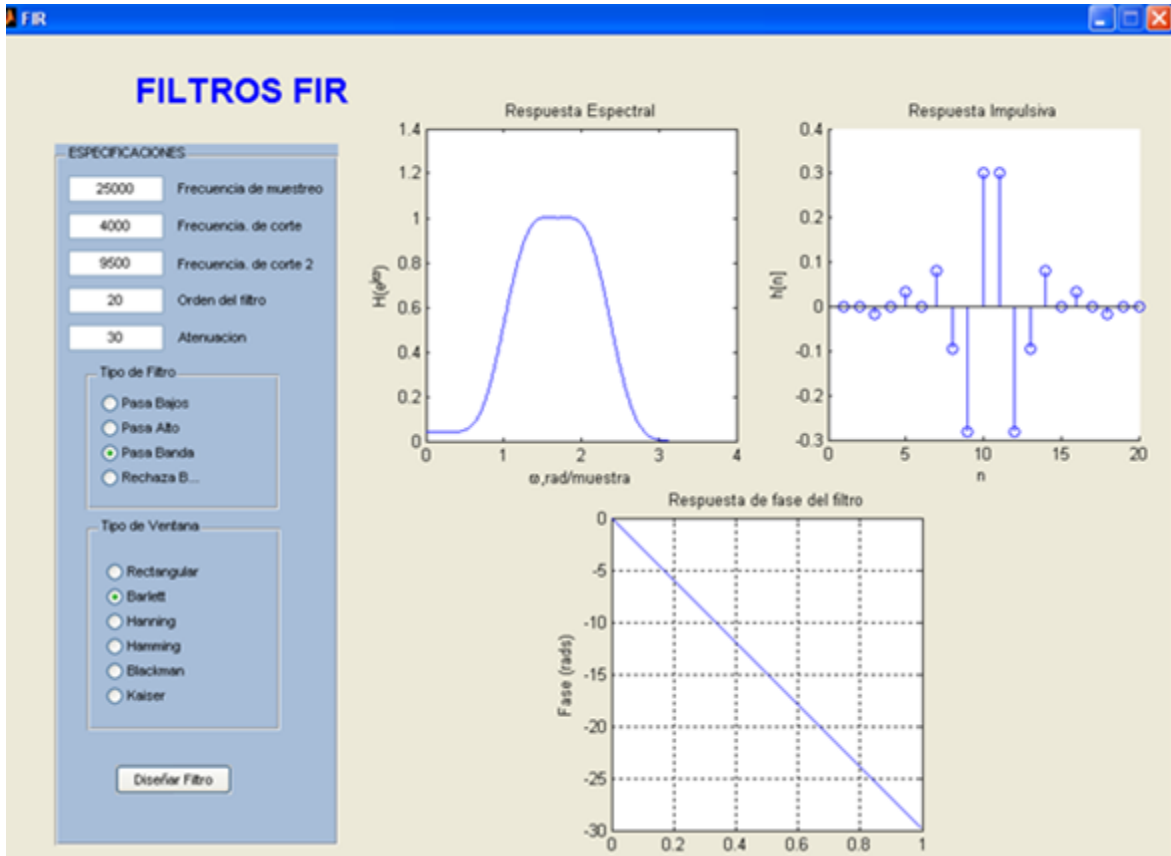


Figura 4. 5: Gráficos de Respuestas Espectrales para Filtro FIR Pasa Bajo.

Elaborado por: El Investigador

4.2.4. Experiencia #4: Filtro FIR Rechaza Banda.

Como en las anteriores experiencias, seguimos analizando la varianza de rizado de respuesta espectral para Filtros FIR Rechaza Banda, cuyas especificaciones se muestran en la figura 4.6. Posterior al ingreso de las especificaciones, damos click en Diseñar Filtro en la cual se obtienen las gráficas correspondientes a las respuestas espectrales (véase la figura 4.6).

En la primera gráfica, se pueden apreciar las frecuencias de corte de $0,75 \text{ rad/muestra}$ y $2,20 \text{ rad/muestra}$ que cumple las características de un filtro FIR rechaza banda con ventana rectangular, pero tiene una señal muy distorsionada

lo cual no cumpliría los requerimientos básicos del tipo de ventana rectangular.

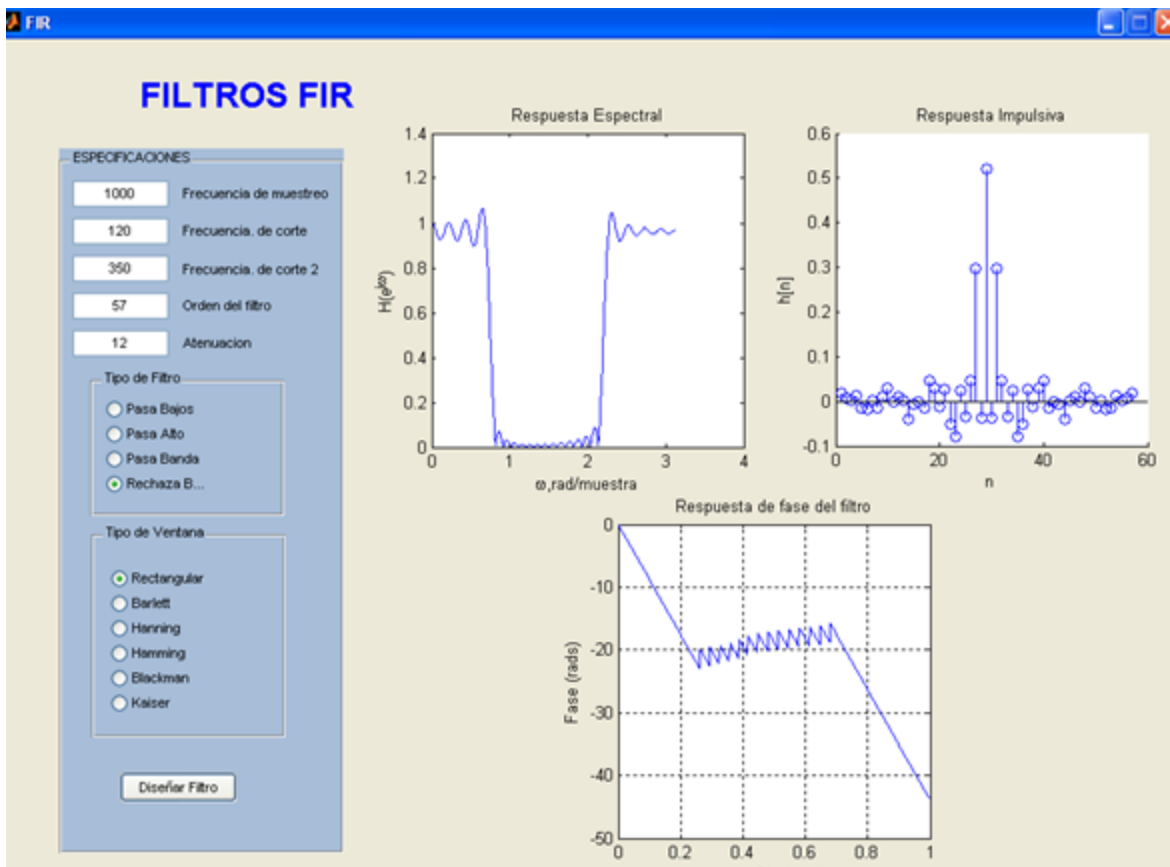


Figura 4. 6: Gráficos de la Respuesta del filtro de la Práctica 4

Elaborado por: El Investigador

4.2.5. Conclusiones del diseño de filtros FIR

A continuación describiremos las conclusiones que se obtienen de los filtros FIR estudiados en la sección 4.2:

- a) Los filtros FIR poseen respuestas de fase lineal, siendo éstos muy utilizados en aplicaciones para transmisión de datos, para desarrollo de equipos de biomedicina, para diseño de amplificadores de audio y para analizar imágenes.
- b) Los Filtros FIR son estables, debido a que se implementan a través de ecuaciones no recursivas.
- c) Los filtros FIR demandan de un orden mayor, y que disponga de memoria mayor así como la disminución del tiempo de procesamiento.
- d) Los Filtros FIR cuya ventana es Kaiser, son las más requeridas en el mercado, debido a dos parámetros, que son necesarios para obtener el

filtro, cuyos parámetros (β y N) son fáciles de obtener a través de sencillas fórmulas.

- e) Los filtros FIR tienen un mejor comportamiento desde punto de vista del retardo de grupo y de la distorsión de fase.

4.3. Desarrollo de las Prácticas de Filtros IIR

Similar al desarrollo de Filtros FIR vistos en la sección 4.2, veremos una interfaz gráfica de Filtros IIR muy idéntica a la figura 4.1, dicha interface se observa en la figura 4.7.

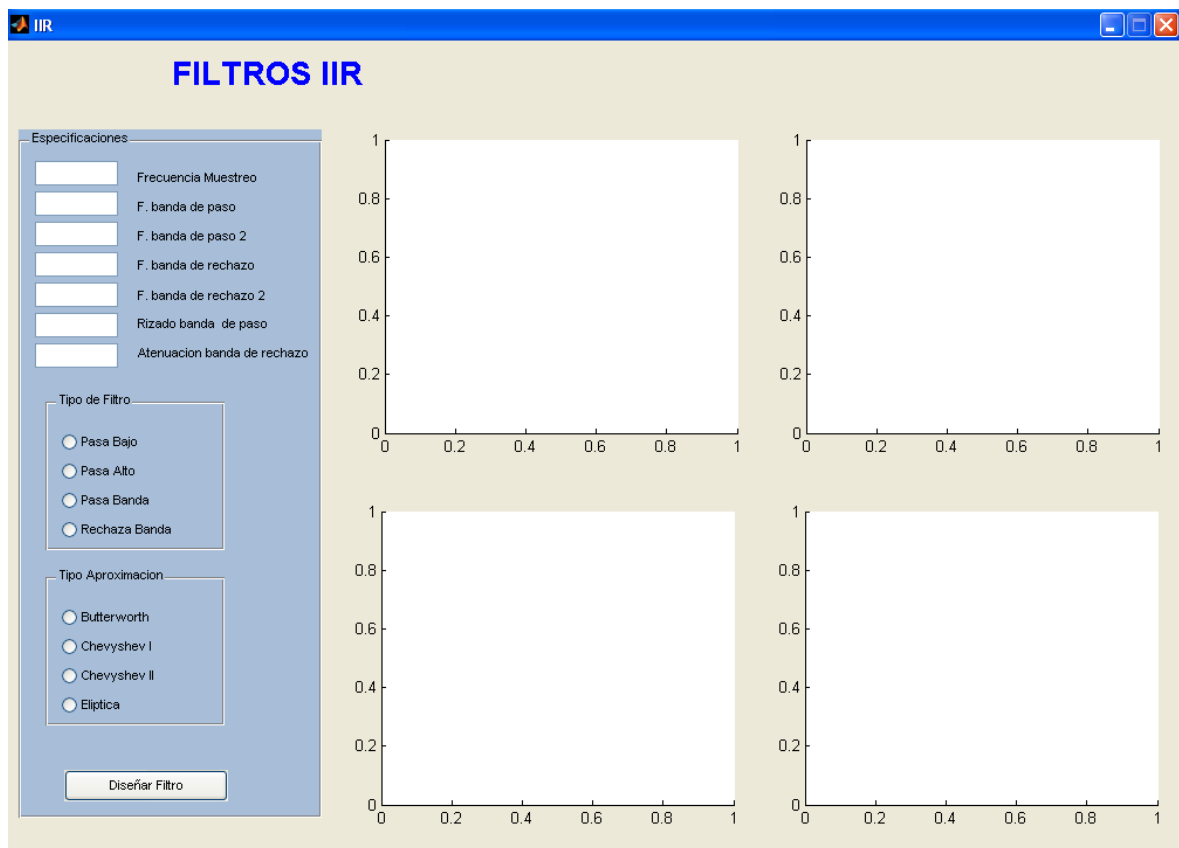


Figura 4. 7: Interfaz para el diseño de Filtros IIR

Elaborado por: El Investigador

En la interfaz mostrada por la figura 4.7 podemos visualizar tres paneles: en el primer panel se deberán ingresar las especificaciones del Filtro IIR, tales como frecuencias: de muestreo, de paso, de rechazo, rizado banda de paso y atenuación de banda de rechazo; en el segundo panel tenemos los tipos de

Filtros IIR, tales como pasa bajo, pasa alto, pasa banda; en el tercer panel se presentan los tipos de aproximación de los filtros IIR a utilizarse, tales como los filtros: Butterworth, Chebyshev tipo 1, Chebyshev tipo 2 y elípticos.

Adicional, se muestra un icono o botón que se lo denomina “Diseñar Filtro”, en el mismo se mostrarán las gráficas de respuesta espectral, impulsional y el diagrama de polos y ceros. A continuación se muestra el código fuente de la interfaz gráfica para el diseño de filtros IIR.

```
function varargout = IIR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @IIR_OpeningFcn, ...
'gui_OutputFcn', @IIR_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function IIR_OpeningFcn(hObject, eventdata, handles, varargin)
```

```

handles.output = hObject;
guidata(hObject, handles);
function varargout = IIR_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function Fmuestreo_Callback(hObject, eventdata, handles)
function Fmuestreo_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol or'));
end

```

```

function Fpaso_Callback(hObject, eventdata, handles)
function Fpaso_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol or'));
end

```

```

function Rizado_Callback(hObject, eventdata, handles)
function Rizado_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol or'));
end

```

```

function Frechazo_Callback(hObject, eventdata, handles)
function Frechazo_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol or'));
end

```

```

function atenuacion_Callback(hObject, eventdata, handles)
function atenuacion_CreateFcn(hObject, eventdata, handles)

```

```

if ispc
    set(hObject,'BackgroundColor','white');
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function IIR_Callback(hObject, eventdata, handles)
%Se limpia el workspace de MATLAB y las variables en uso. clc;
%*****

```

```

Fs=str2double(get(handles.Fmuestreo,'String'));
Fpb=str2double(get(handles.Fpaso,'String'));
Fsb=str2double(get(handles.Frechazo,'String'));
Fpb1=str2double(get(handles.fpaso2,'String'));
Fsb1=str2double(get(handles.Frechazo2,'String'));
Rpb=str2double(get(handles.Rizado,'String'));
Rsb=str2double(get(handles.atenuacion,'String'));
fs=Fs/2; %Frec. de muestreo fpb=Fpb/fs;
%Frec. de borde pasa banda fsb=Fsb/fs;
% Frec. de borde rechaza banda fsb1=Fsb1/fs;
fpb1=Fpb1/fs;
wpb = [fpb fpb1]; wsb = [fsb fsb1]; aprox = handles.type;
TipFil = handles.clase;

```

```

switch TipFil case 1

```

```

if aprox == 1

```

```

    [n,fn]=buttord(fpb, fsb, Rpb, Rsb); [b,a] =butter(n,fn,'low');

```

```

elseif aprox == 2

```

```

    [n,fn]=cheb1ord(fpb, fsb, Rpb, Rsb); [b,a] =cheby1(n,Rpb,fn,'low');

```

```

elseif aprox ==3

```

```

    [n,fn]=cheb2ord(fpb, fsb, Rpb, Rsb); [b,a] =cheby2(n,Rpb,fn,'low');

```

```

else

```

```

    [n,fn]=ellipord(fpb, fsb, Rpb, Rsb); [b,a] =ellip(n,Rpb,Rsb,fn,'low');

```

```

end

```

```

case 2
if aprox == 1
    [n,fn]=buttord(fpb, fsb, Rpb, Rsb); [b,a] =butter(n,fn,'high');
elseif aprox == 2
    [n,fn]=cheb1ord(fpb, fsb, Rpb, Rsb); [b,a] =cheby1(n,Rpb,fn,'high');
elseif aprox ==3
    [n,fn]=cheb2ord(fpb, fsb, Rpb, Rsb); [b,a] =cheby2(n,Rpb,fn,'high');
else
    [n,fn]=ellipord(fpb, fsb, Rpb, Rsb); [b,a] =ellip(n,Rpb,Rsb,fn,'high');
end

case 3
if aprox == 1
    [n,fn]=buttord(wpb, wsb, Rpb, Rsb); [b,a] =butter(n,fn,'bandpass');
elseif aprox == 2
    [n,fn]=cheb1ord(wpb, wsb, Rpb, Rsb); [b,a] =cheby1(n,Rpb,fn,'bandpass');
elseif aprox ==3
    [n,fn]=cheb2ord(wpb, wsb, Rpb, Rsb); [b,a] =cheby2(n,Rpb,fn,'bandpass');
else
    [n,fn]=ellipord(wpb, wsb, Rpb, Rsb); [b,a] =ellip(n,Rpb,Rsb,fn,'bandpass');
end

otherwise
if aprox == 1
    [n,fn]=buttord(wpb, wsb, Rpb, Rsb); [b,a] =butter(n,fn,'stop');
elseif aprox == 2
    [n,fn]=cheb1ord(wpb, wsb, Rpb, Rsb); [b,a] =cheby1(n,Rpb,fn,'stop');
elseif aprox ==3
    [n,fn]=cheb2ord(wpb, wsb, Rpb, Rsb);
    [b,a]=cheby2(n,Rpb,fn,'stop');
else

```

```

[n,fn]=cheb2ord(wpb, wsb, Rpb, Rsb);
    [b,a]=cheby2(n,Rpb,fn,'stop');
else
    [n,fn]=ellipord(wpb, wsb, Rpb, Rsb);
    [b,a]=ellip(n,Rpb,Rsb,fn,'stop');
end
end

[H,w]=freqz(b,a,512,1);
%Trazado de la respuesta en Magnitud
axes(handles.axes1) plot(w,20*log10(abs(H)), 'k');
grid on;
title(['Respuesta en magnitud, orden=', num2str(n)]);
xlabel('frecuencia');
ylabel('H(f) db') axis([0 0.5 -Rsb-10 0]) axes(handles.axes2) plot(w,angle(H),'r'); grid on;
title(['Respuesta en magnitud, orden=', num2str(n)]);
xlabel('frecuencia')
ylabel('ángulo de H rad')

%Respuesta al impulso axes(handles.axes3) [y,t]= impz(b,a,60);
stem(t,y,'m'); xlabel('n') ylabel('h[n]')
title(['Respuesta al impulso, orden=', num2str(n)]);
%Ploteo de los polos y ceros z= roots(b); %Zeros
p = roots(a); % Polos axes(handles.axes4) zplane(z,p)
title(['Polos y Ceros'])

% -----
function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
if hObject == handles.button
    tipo = 1;
elseif hObject == handles.chey1 tipo = 2;
elseif hObject == handles.chey2
    tipo = 3;
else
    tipo = 4;

```

```
end
```

```
handles.type = tipo;
```

```
guidata(hObject,handles)
```

```
% -----
```

```
function uipanel3_SelectionChangeFcn(hObject, eventdata, handles)
```

```
if hObject == handles.PB
```

```
    fil = 1;
```

```
elseif hObject == handles.PA
```

```
    fil = 2;
```

```
elseif hObject == handles.PBanda fil = 3;
```

```
else
```

```
    fil = 4;
```

```
end
```

```
handles.clase = fil;
```

```
guidata(hObject,handles)
```

```
function fpaso2_Callback(hObject, eventdata, handles)
```

```
function fpaso2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc
```

```
    set(hObject,'BackgroundColor','white');
```

```
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end
```

```
function Frechazo2_Callback(hObject, eventdata, handles)
```

```
function Frechazo2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc
```

```
    set(hObject,'BackgroundColor','white');
```

```
else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end
```

4.3.1. Experiencia #1: Filtro IIR Pasa Bajo – Butterworth.

Para evaluar la presente experiencia práctica acerca de los Filtros IIR Pasa Bajo, empleamos una muestra de 100 Hz de la señal de datos, una atenuación de rizo de banda de paso igual a 1 dB, una frecuencia de banda de paso igual a 12 Hz y con una atenuación de banda de rechazo igual a 30 dB cuya frecuencia debe ser a 15 Hz.

Para la presente simulación, procederemos de forma similar a los Filtros FIR tratados en la sección 4.2, ingresamos los parámetros de las especificaciones que requieren los Filtros IIR para su correcto funcionamiento, posteriormente seleccionamos el tipo de filtro y la aproximación necesarias para este tipo de filtro. En la figura 4.8 se muestra los paneles de especificaciones, tipo de filtro y aproximación.

The image shows a software dialog box titled "Especificaciones" (Specifications) for designing an IIR filter. It contains several input fields and radio button options. The input fields are: "Frecuencia Muestreo" (100), "F. banda de paso" (12), "F. banda de paso 2" (empty), "F. banda de rechazo" (15), "F. banda de rechazo 2" (empty), "Rizado banda de paso" (1), and "Atenuacion banda de rechazo" (30). Below these are two sections: "Tipo de Filtro" (Filter Type) with radio buttons for "Pasa Bajo" (selected), "Pasa Alto", "Pasa Banda", and "Rechaza Banda"; and "Tipo Aproximacion" (Approximation Type) with radio buttons for "Butterworth" (selected), "Chevyshev I", "Chevyshev II", and "Eliptica". A "Diseñar Filtro" (Design Filter) button is at the bottom.

Figura 4. 8: Paneles del Diseño de Filtros IIR

Elaborado por: El Investigador

Posterior al ingreso de los valores mostrados en la figura 4.8, dar click en Diseñar Filtro, dando como resultado cuatro gráficas: respuesta impulsional, espectral y diagrama de polos y ceros ilustradas en la figura 4.9. La primera gráfica corresponde a la respuesta espectral e impulsional del Filtro IIR Butterworth en el cual el número de orden es el 17, cuya frecuencia de corte es $0,12 \text{ rad/s}$. Es importante recalcar, que esta aproximación es muy lineal versus otras aproximaciones. Mientras que para el diagrama de polos y ceros podemos ver que los polos se encuentran dentro de la circunferencia, esto significa que el filtro es de fase mínima; pero podemos apreciar que el 50% de los ceros se encuentran fuera de la circunferencia, por lo tanto el filtro es inestable.

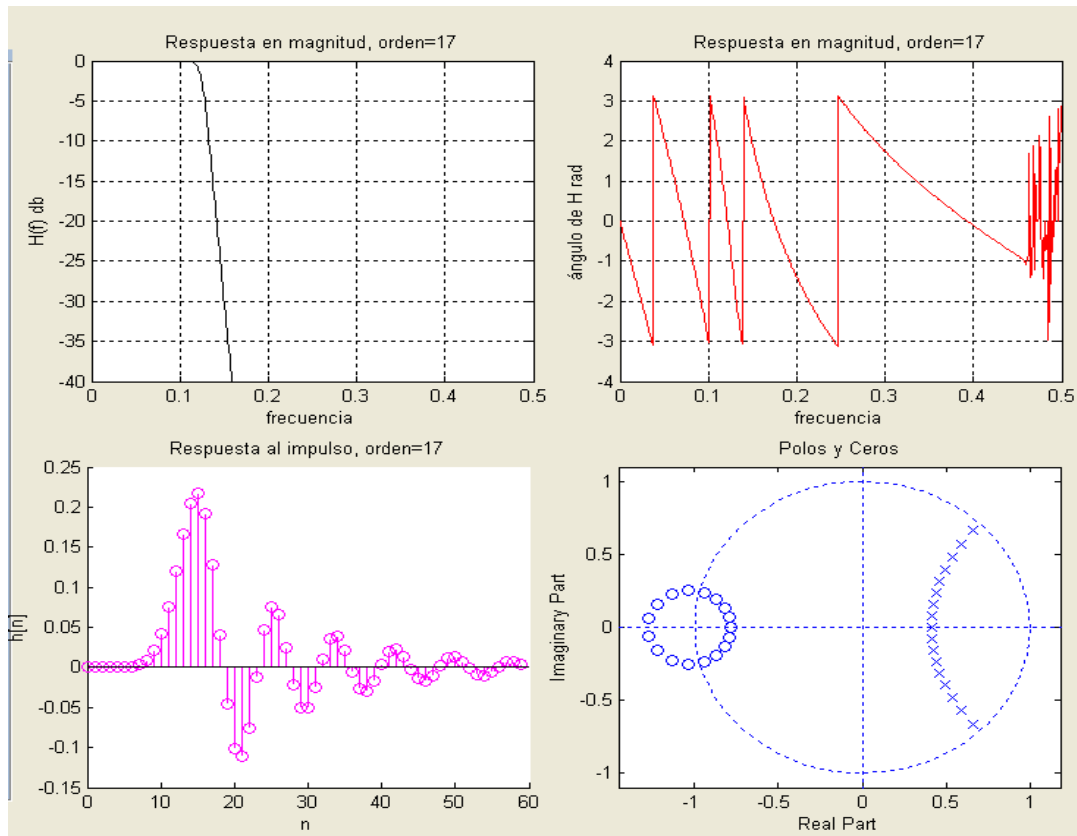


Figura 4. 9: Gráficos de las respuestas de Filtros IIR Pasa Bajo – Butterworth.

Elaborado por: El Investigador

4.3.2. Experiencia #2: Filtro IIR Pasa Alto – Chebyshev 1.

Ahora evaluamos al Filtro IIR Pasa Alto – Chebyshev 1, empleando una muestra de 11,50 kHz de la señal de datos, una atenuación de rizo de banda de paso igual

a 2 dB, una frecuencia de banda de paso igual a 1,20 kHz y con una atenuación de banda de rechazo igual a 50 dB cuya frecuencia debe ser igual a 4,30 kHz, tal como se ilustra en la figura 4.10. Podemos observar que tanto la respuesta espectral como la impulsional del filtro pasa alto corresponde al orden 3, con frecuencia de corte de $0,10 \text{ rad/muestras}$. Comparando al filtro de la sección 4.3.1., el orden del filtro es menor. Ahora, los diagramas de polos y ceros apreciamos tanto los polos como los ceros se encuentran dentro de la circunferencia, por lo tanto el filtro es mínima fase y estable a la vez.

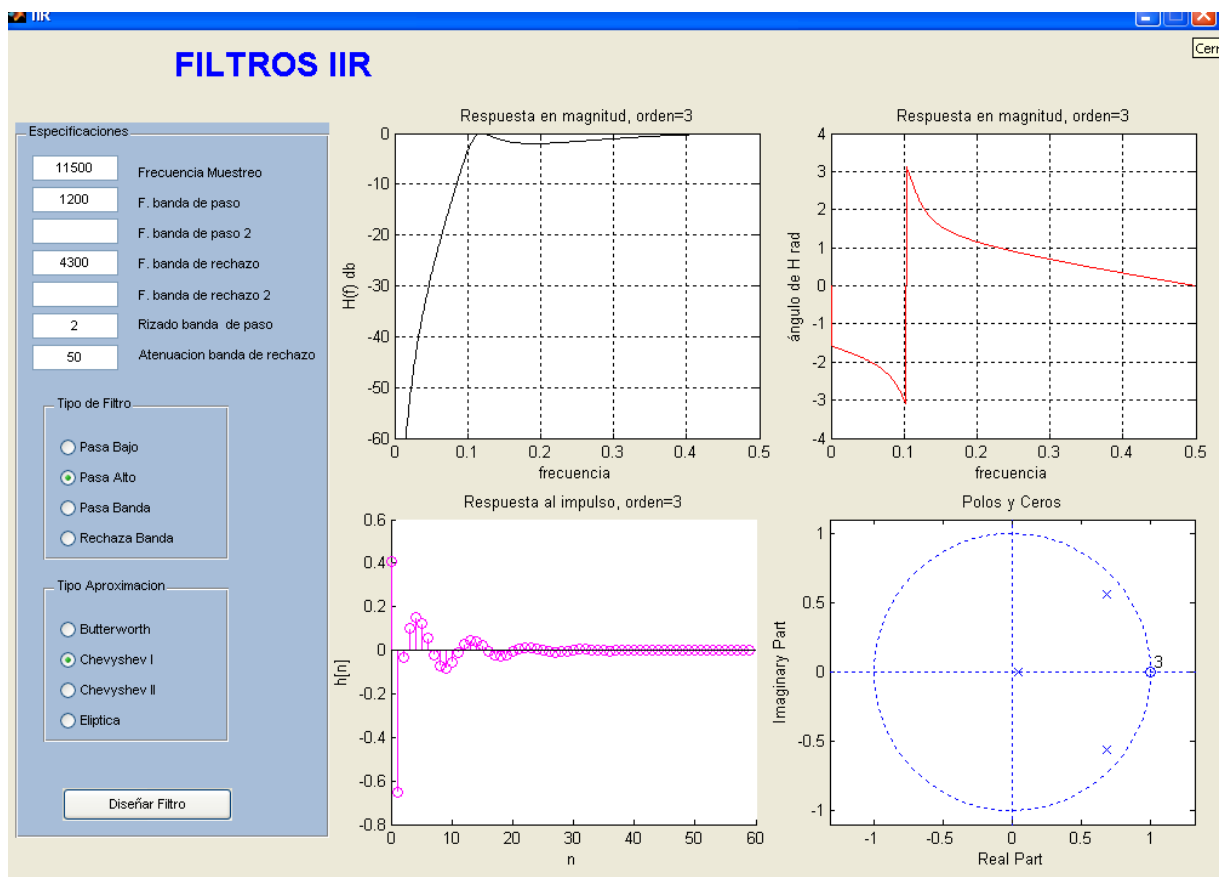


Figura 4. 10: Gráficos de las respuestas del Filtro IIR Pasa Alto – Chebyshev 1.

Elaborado por: El Investigador

4.3.3. Experiencia #3: Filtro IIR Pasa Banda – Chebyshev 2.

Para una señal de datos muestreada a 500 Hz, diseñar un filtro pasa banda con las siguientes especificaciones:

- Frecuencia de paso : 85 Hz y 160 Hz

- Frecuencia de rechazo: 60Hz y 190Hz
- Rizo en la banda de paso: 0.5dB
- Atenuación en la banda de rechazo: 15

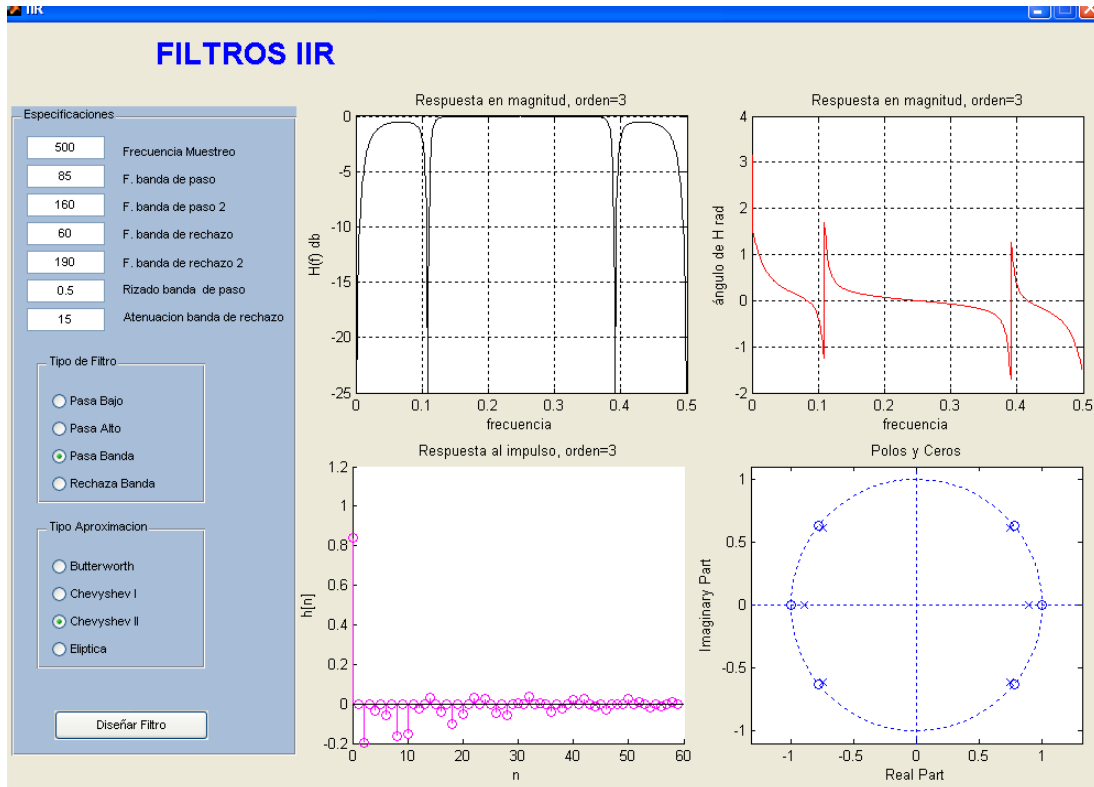


Figura 6.22 Gráficos de las respuestas del Filtro IIR pasa banda con Aproximación de Chevyshev II

Elaborado por: El Investigador

En la figura 6.22 se observa la respuesta espectral e impulsional del filtro pasa banda aplicando la aproximación Chevyshev II. Se aprecia que el orden del filtro es 3. En el diagrama de polos y ceros se aprecia que los polos están de forma simétrica, es decir están equiespaciados y se encuentran dentro de la circunferencia por consiguiente el filtro tiene una fase mínima y estable.

4.3.4. Experiencia #4: Filtro IIR Paso Bajo – Elíptica.

Diseñar un filtro digital IIR pasa-bajos muestreada, y debe ser filtrada según los siguientes requerimientos:

- Frecuencia de corte de la banda pasante: 1000 Hz
- Frecuencia de corte de la banda suprimida: 1600 Hz

- Frecuencia de muestreo y reconstrucción: 8 kHz
- Atenuación en la banda pasante: 3 dB
- Atenuación en la banda suprimida: 23 dB

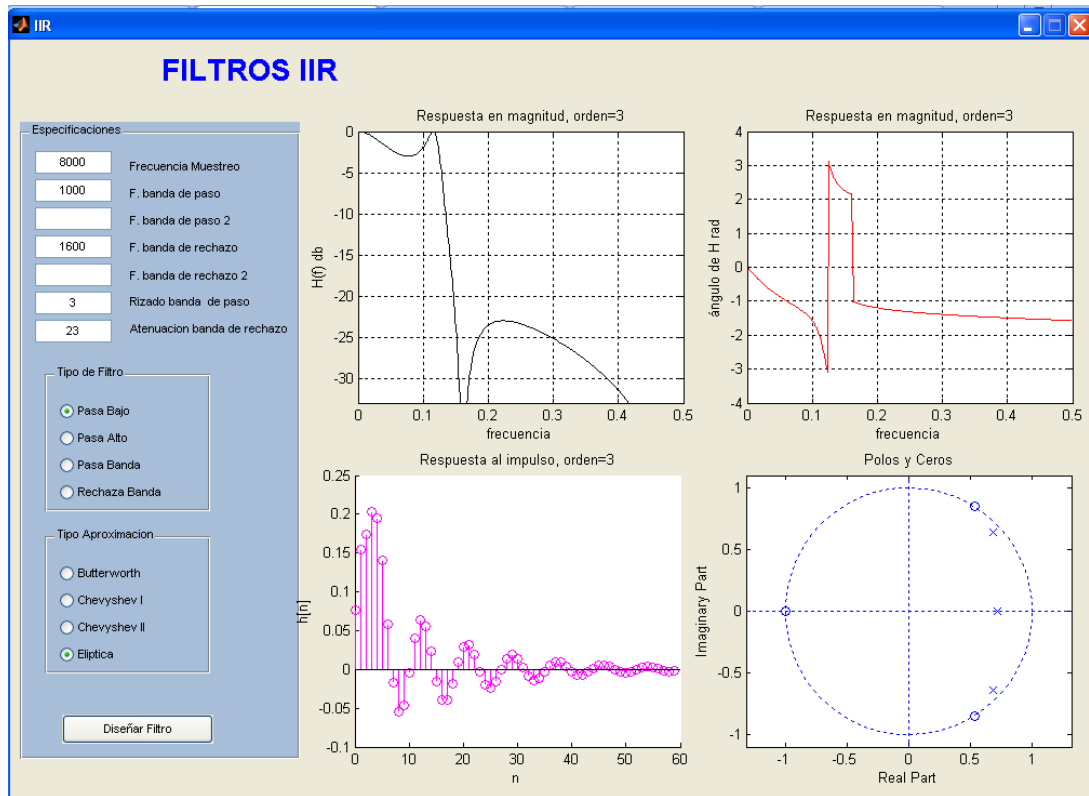


Figura 6.23 Gráficos de las respuestas del Filtro IIR pasa bajo con Aproximación Elíptica

Elaborado por: El Investigador

En la figura 6.23 se observa la respuesta espectral e impulsional del filtro pasa bajo aplicando la aproximación Elíptica. Se aprecia que el orden del filtro es 3 y su frecuencia de corte es de 0.13 rad/muestra.

La aproximación Elíptica requiere de un orden menor para su diseño, por consiguiente tiene una menor complejidad computacional. Sin embargo como se aprecia en los gráficos la fase es lo menos lineal, es una de las desventajas en la aproximación Elíptica. Mediante el diagrama de polos y ceros se puede dar cuenta que es un filtros de fase mínima y estable.

4.3.5. Conclusiones del diseño de filtros IIR

- Este tipo de filtros presenta polos y ceros que determina la estabilidad y la causalidad del sistema.
- IIR pueden cumplir las mismas exigencias que los filtros FIR; pero con menos orden de filtro. Esto es importante a la hora de implementar el filtro, pues presenta una menor carga computacional.
- Este tipo de filtros puede ser inestables, aún cuando se diseñen para ser estables.
- Es importante escoger la aproximación adecuada para que se cumpla con las especificaciones dadas.
- La implementación que mayor carga computacional requeriría es la Butterword, pues el orden del filtros es mayor que dicha aproximación, mientras que el de menor orden es la elíptica.
- La aproximación elíptica es, entre todas, la que requiere menor orden, por lo que es comúnmente utilizada cuando el principal interés se centra en minimizar el orden del filtro.
- La aproximación de Butterworth es la que presenta una fase más próxima al ideal para un orden dado, pero el orden que necesita para cumplir las especificaciones suele ser notablemente mayor al que requieren las demás.

CONCLUSIONES:

- La creación de laboratorios virtuales tiene algunas ventajas importantes con respecto a los laboratorios reales. Dado que un laboratorio virtual, en este caso específico de filtros digitales en DSP, se basa en modelos matemáticos que se ejecutan en ordenadores, su configuración y puesta a punto es mucho más sencilla que la configuración y puesta a punto de los laboratorios reales. Además, presentan un grado de robustez y seguridad mucho más elevado ya que al no haber dispositivos reales éstos no pueden causar problemas en el entorno.
- En la vida profesional, e incluso en la vida diaria, la destreza en el uso de las herramientas informáticas, sean del ámbito que sean, es un elemento diferenciador. Con ello se consigue aportar al alumno una serie de conocimientos transversales que si bien pueden no ser el objetivo principal del laboratorio que se esté diseñando, le servirán en muchos ámbitos en el futuro, por ello se propuso realizar la propuesta con un software, Matlab, para que los alumnos se familiaricen con dicho software.
- Actualmente los DSP tienen un amplio campo de acción, es decir, las prestaciones y las aplicaciones que ofrecen son enormes, por lo cual se ve la necesidad de estar en vanguardia, por ende el desarrollo una serie de prácticas de simulación de filtros digitales aporta a los alumnos a poner en práctica todos los conocimientos adquiridos en el curso de Procesamiento Digital de Señales que imparte dicha Facultad.
- El programa está diseñado enfocado a los objetivos planteados dentro de la propuesta facilitando el aprendizaje de filtros digitales. Se facilitó el proceso práctico en la utilización del programa creando una GUI (Graphic User Interface- Interfaz Gráfica de Usuario) a través de la cual el usuario puede llevar a cabo diferentes funciones sin necesidad de entrar a la programación.

RECOMENDACIONES:

- Una forma interesante de utilizar los laboratorios virtuales es junto a los laboratorios reales, de forma que los alumnos realizarían primero las prácticas en laboratorios virtuales, para pasar posteriormente, cuando el instructor lo considerase oportuno, al laboratorio real.
- Proponer más prácticas de Filtros digitales, para la mejor comprensión acerca del diseño de los mismos, así como también los alumnos saquen sus propias conclusiones de acuerdo a los gráficos obtenidos en cada una de las prácticas.
- Realizar el diseño de filtros digitales, tanto FIR como IIR, utilizando otros métodos y algoritmos matemáticos.

3.3.3 BIBLIOGRAFIA:

- IRARRÁZVAL Pablo, Análisis de Señales, Editorial Mc Graw Hill, Primera Edición.
- KARRIN, Steven: Signal and Systems with Matlab Computing and Simulink Modeling, Orchard Publications, Cuarta Edición.
- OPPENHEIN Alan, WILLSKY Alan, Señales y Sistemas, Editorial Pearson Prentice Hall, Segunda Edición.
- PROAKIS John G., Manolakis Dimitris G., Tratamiento digital de señales, Prentice Hall, In., Cuarta Edición, España, 1998
- PROAKIS John G., Vinay Ingle., Digital Signal Processing using Matlab, Global Engineering, Cuarta Edición, 1999
- SOLIMAN, Samir. Señales y Sistemas Continuos y discretos, Editorial Prentice Hall, Segunda Edición, 1999
- TOCCI Ronald, Sistemas Digitales Principios y Aplicaciones, Editorial Pearson Prentice Hall, Décima Edición.

INTRODUCCIÓN AL FILTRADO DIGITAL:

- <http://www.dtic.upf.edu/~egomez/teaching/sintesi/SPS1/Tema7-FiltrosDigitales.pdf>
- <http://www.frsn.utn.edu.ar/tecnicas3/manuales/Apunte%20Filtros%20Digitales.pdf>
- <http://www.elai.upm.es:8009/spain/Publicaciones/pub99/intropds.pdf>

FILTROS FIR:

- <http://varoitus.barcelonamedia.org/rafael/Teaching/Courses/DSP/modulo7.pdf>
- www.ingelec.uns.edu.ar/pds2803/Materiales/Cap07/07-Cap07.pdf
- <http://www.ing.uc.edu.ve/~azozaya/docs/DSP/dfir.pdf>
- http://www.dsp.efn.unc.edu.ar/documentos/Filtro_FIR.pdf
- www.uaq.mx/investigacion/...VII/ITQ%20Martinez%20Barrera.doc
- <http://dsp1.materia.unsl.edu.ar/Filtros%20FIR%20Resumen.pdf>

FILTROS IIR:

- http://www.elai.upm.es:8009/spain/Publicaciones/pub01/intro_procsdig.pdf
- http://www.dsp.efn.unc.edu.ar/documentos/Filtro_IIR.pdf
- <http://www.ing.uc.edu.ve/~azozaya/docs/DSP/dfiir.pdf>
- <http://varoitus.barcelonamedia.org/rafael/Teaching/Courses/DSP/modulo8.pdf>
- <http://dsp1.materia.unsl.edu.ar/Filtros%20IIR%20Resumen.pdf>

MATLAB

- <http://www.escet.urjc.es/~matemati/TCTS/matlab-filtros.pdf>

- <http://www.slideshare.net/lonely113/procesamiento-digital-de-seales-con-matlab>
- http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo3.pdf
- http://www.dspace.espol.edu.ec/bitstream/123456789/10740/11/MATLAB_GUIDE.pdf