

CAPITULO 2. DESCRIPCIÓN DEL PROYECTO.

2.1 OBJETIVOS DEL PROYECTO.

La finalidad del presente proyecto no se centra en el estudio a profundidad de la teoría de los filtros eléctricos, más bien consiste en la puesta en práctica y materialización de los conocimientos relacionados con el diseño en dicha área de estudio. Lo anterior significa que este trabajo no es una investigación documental, sino que tiene como propósito la creación de un instrumento de apoyo que implemente distintos procedimientos de cálculo y diseño para que de ella se sirvan los estudiosos del área, ya sea para realizar operaciones por aproximaciones o para comprobación de resultados teóricos. Pero, obviamente, se debe llevar a cabo una revisión de las bases teóricas para poder entender los temas y procedimientos involucrados.

El objetivo central es realizar un programa que sirva de herramienta a aquellas personas relacionadas con la disciplina, ya sean profesores, estudiantes o incluso profesionistas, al facilitar la obtención de los valores y parámetros de las redes, así como las topologías a implementar.

En resumen, se trata de desarrollar una herramienta eficiente, precisa, veloz y confiable para el cálculo y diseño de circuitos para filtrado de señales, que permita obtener parámetros esenciales como polos, ceros, respuesta en frecuencia, retraso de grupo, función de transferencia, diagramas de polos y ceros, diagrama de implementación.

2.2 JUSTIFICACIÓN.

Como ya se mencionó, el uso de circuitos para tratamiento de señales eléctricas, es decir de filtros eléctricos, es prácticamente inevitable en cualquier dispositivo electrónico, y las aplicaciones son numerosas y muy variadas. Debido a todo esto, es esencial diseñarlos e implementarlos de manera adecuada, para así poder garantizar el óptimo funcionamiento de los sistemas.

La aplicación práctica de la teoría de filtros implica un desarrollo matemático extenso y considerable, por lo que antes de darse a la tarea de estudiar dicha área propiamente, se debe hacer una revisión de ciertas bases matemáticas, lo cual merma el tiempo de estudio del campo de interés. Por ello, es necesaria la creación de herramientas de apoyo que nos desahoguen la pesada tarea de cálculo para poder centrarnos más en el estudio, análisis y comparación de los resultados teóricos y prácticos, así como el estudio de la misma teoría.

El programa desarrollado deberá servir como auxiliar en el estudio de los filtros y, a través del mismo, permitir al estudiante verificar y reafirmar los conceptos vistos en clase, además de tener la facilidad de observar de manera rápida las características de cada tipo de filtro y las consecuencias de cambiar alguno de los parámetros sin necesidad de tener que gastar tiempo en la aplicación de repetitivos algoritmos matemáticos. Por último, servirá como herramienta auxiliar en la realización práctica de diseños, al llevar a cabo simulaciones y desarrollo de topologías.

2.3 CARACTERÍSTICAS GENERALES.

Es importante hacer notar que JAVAFilters es el primer programa en su tipo, ya que no existe hasta el momento ningún paquete computacional con las mismas características. De los programas que hay actualmente realizados en JAVA[®] la mayoría, o todos, están enfocados a filtros digitales, no tienen un buen manejo de la base matemática, no calculan todos los parámetros requeridos ni despliegan información de las topologías. Por ello, el presente proyecto no sólo es innovador sino único.

Durante el desarrollo del proyecto, se buscó que el programa fuera lo más completo y eficiente posible, además de ser de fácil manejo para que la interacción con el usuario resultase lo más sencilla posible. Por estos motivos, se implementaron las siguientes características:

- 1) Las aproximaciones para diseño de Filtros FIR se implementaron usando algoritmos por los tipos de ventana:
 - a) Kaiser
 - b) Bartlett
 - c) Han
 - d) Rectangular
 - e) Hamming
 - f) Blackman

- 2) Para el diseño de Filtros Analógicos y Digitales IIR se implementaron las aproximaciones:
- a) Butterworth
 - b) Chebyshev
 - c) Chebyshev Inverso
 - d) Elíptico
- 3) Para cada tipo de función es posible diseñar filtros de los tipos:
- a) Pasabajas
 - b) Pasa altas
 - c) Pasa banda
 - d) Rechaza banda
- 4) Movilidad de polos y ceros por medio del ratón.
- 5) Correspondencia simultánea entre desplazar los polos y/o ceros y la variación en la respuesta en magnitud o fase.
- 6) La detección del tipo de filtro puede ser de manera automática.
- 7) Validación de algoritmos y valores tales que el programa es capaz de soportar errores como división entre cero, overflow, etc.

- 8) El programa es capaz de autoescalar las gráficas.
- 9) No es posible visualizar varias ventanas simultáneamente, lo cual facilita el manejo y visualización de los resultados.
- 10) Los valores de los elementos de las redes analógicas son calculados en tiempo real, lo cual permite ajustar fácilmente los componentes.
- 11) Despliegue de mensajes en caso de errores o incongruencias.
- 12) El programa no tiene un orden secuencial predefinido, sino que es totalmente dependiente de la interacción con el usuario.
- 13) La captura de los parámetros está habilitada para aceptar valores en distintos formatos.
- 14) En ciertas ventanas se incluyeron barras de desplazamiento para facilitar la visualización de los datos generados.
- 15) Las gráficas son escalables a voluntad, lo que implica que sea más fácil observarlas dependiendo de las necesidades de cada usuario.
- 16) La interfaz está diseñada de manera tal que sea sumamente fácil para cualquier persona el comprender dónde y cómo introducir los datos necesarios, generar los valores correspondientes y cómo modificar parámetros para obtener los resultados deseados.

- 17) El diseño gráfico de la interfaz incluye colores y elementos gráficos que además de facilitar el uso del programa, la hace más amigable.
- 18) La interacción con el programa puede ser directa, es decir, como aplicación o mediante una página Web en la cual está incrustado el applet programado. Cabe mencionar, que esto no significa que se requiera una conexión a Internet para su uso, ya que los archivos necesarios para su ejecución se encuentran en el disco compacto anexo a este documento.

2.4 REQUERIMIENTOS.

En cuanto a la plataforma o arquitectura computacional no existe un requerimiento específico por parte del programa, debido a que fue diseñado de manera tal que sea ejecutable en cualquier tipo de plataforma, ya sea SPARC, PC o MacIntosh. Sólo es necesario que se cumplan los requerimientos de software y hardware que pide Java[®] 2 Runtime Environment 1.4.2, y lógicamente tenerlo instalado . Dichos requerimientos dependen de la plataforma, y se muestran en la siguiente tabla:

Solaris SPARC Requerimientos de Sistema 32-bit: SDK J2RE 64-bit: SDK J2RE	Solaris 9 (32/64 bit)	Gnome2-metacity, CDE-dtwm
	Solaris 8 (32/64 bit)	CDE-dtwm/OpenWin-olwm
	Solaris 7 (32 bit)	CDE-dtwm/OpenWin-olwm
Solaris x86 Requerimientos de Sistema (same as SPARC 32-bit)	Solaris 9 (32 bit)	Gnome2-metacity, CDE-dtwm
	Solaris 8 (32 bit)	CDE-dtwm/OpenWin-olwm
	Solaris 7 (32 bit)	CDE-dtwm/OpenWin-olwm
IA32 Windows Requerimientos de Sistema 32-bit: SDK J2RE Procesador Intel Pentium 166 MHz mínimo 32 MB de memoria RAM mínimo 48 MB de memoria RAM recomendado 130 MB libres en disco duro	Windows XP Professional (SP1)	Windows Active & Classic Desktop
	Windows XP Home	Windows Active & Classic Desktop
	Windows 2000 Professional (SP3)	Windows Active & Classic Desktop
	Windows 98 (1st & 2nd Editions)	Windows Active & Classic Desktop
	Windows NT 4.0 (SP6a)	Windows Active & Classic Desktop
	Windows ME	Windows Active & Classic Desktop
	Windows Server 2003, Web Edition	Windows Active & Classic Desktop
	Windows Server 2003, Standard Edition	Windows Active & Classic Desktop
	Windows Server 2003, Enterprise Edition	Windows Active & Classic Desktop
	Windows Server 2003, Datacenter Edition	Windows Active & Classic Desktop
Itanium 2 Windows Requerimientos de Sistema 64-bit: SDK	Windows Server 2003, Enterprise Edition	Windows Active & Classic Desktop
	Windows Server 2003, Datacenter Edition	Windows Active & Classic Desktop
IA32 Linux Requerimientos de Sistema 32-bit: SDK J2RE 32 MB de memoria RAM mínimo 48 MB de memoria RAM recomendado Modo de color de 16bits 75 MB libres en disco duro	Red Hat 7.3	Gnome-sawfish 1.0
	Red Hat 8.0	Gnome2-metacity 2.4.2
	Red Hat Enterprise Linux WS 2.1	Gnome-sawfish 1.0
	Red Hat Enterprise Linux ES 2.1	Gnome-sawfish 1.0
	Red Hat Enterprise Linux AS 2.1	Gnome-sawfish 1.0
	SuSE 8.0	Gnome-sawfish 2.0
	TurboLinux 7.0	Gnome-sawfish 1.0
	SLEC 8	Gnome2-metacity 2.4.34
Itanium 2 Linux Requerimientos de Sistema 64-bit: SDK	Red Hat Advanced Server 2.1	Gnome-sawfish 1.0
	Red Hat Advanced Workstation 2.1	Gnome-sawfish 1.0
Navegadores Soportados (Se requiere tener alguno de ellos para poder ejecutar el applet y contar con la versión correspondiente del plug-in de Java®)		
Netscape 4.7.x, 6.2.2, 7	Mozilla 1.2.1, 1.3	Internet Explorer 5.5 (SP2+), 6.x
Nota: IA32 = Intel Architecture 32-bit		

2.5 NOTA AL USUARIO.

Como se mencionó anteriormente, el propósito del presente proyecto es el de simplificar la tarea de diseño de filtros tanto analógicos como digitales, pero cabe hacer hincapié en algunos aspectos importantes. En primera instancia algunas aproximaciones y topologías no han sido cubiertas, en segundo término, el programa no incluye capacidad de ayuda dentro del mismo. De este modo, dichas restricciones deberán ser tomadas en cuenta para futuras revisiones y mejoras, tales como programar los algoritmos faltantes y la posibilidad de ventanas de ayuda o un tutorial. Para estos fines, el código fuente del programa está incluido dentro del mismo disco y se encuentra documentado para facilitar el hacer modificaciones al mismo.

Por último, se considera importante recalcar que el presente trabajo está basado en los proyectos mencionados con anterioridad como: Diseño de Filtros Auxiliado por Computadora (**DOSFilters**) de Eduardo Jiménez López, **WinFilters** (Diseño de Filtros Analógicos y Digitales en Ambiente Windows® PC) realizado por Ryszard Jaroszyński, y **MFilters** (Diseño de Filtros Analógicos y Digitales en MATLAB®) de José René Alcántara M.