

Laboratorio 3: Transformada de Fourier y Filtros

Laboratorio de Señales y Sistemas

Facultad de Ingeniería
Escuela de Ingeniería en Informática y
Telecomunicaciones

1. Introducción

El análisis de Fourier, tanto en su versión de Serie como de Transformada, permiten entender en otra dimensión (frecuencia) el comportamiento de las señales, además de brindar un método alternativo para resolver problemas que en el dominio temporal resultan complejos.

Por otro lado, los filtros son ampliamente utilizados en el tratamiento de señales. Un filtro (o arreglo de filtros) permite separar rangos de frecuencias, o bien atenuar/amplificar amplitudes dentro de cierto rango de frecuencias. En términos muy elementales, los filtros se clasifican, de acuerdo al rango de frecuencias que eliminan o que conservan, en pasabajos, pasaaltos, o pasabanda/eliminabanda.

Unos de los filtros más elementales son los analógicos de una etapa, como un circuito RC, semejante al indicado en la Figura 1.

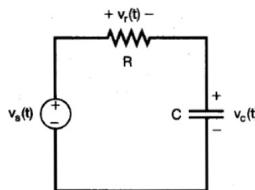


Figura 1: Filtro RC de primer orden.

Debido a que el voltaje de salida, tomado desde el condensador, es función del tiempo, se puede modelar como una ecuación diferencial de primer orden. A su vez, responde a las características de un sistema LTI, y su respuesta en frecuencia muestra una atenuación de la amplitud y un aumento de la fase en la medida en que la frecuencia aumenta, por lo que este circuito en particular se comporta como un filtro pasabajos.

Este mismo proceso se puede realizar de manera digital a través de técnicas de filtrado, consecuencia de aplicar el modelamiento de una función de transferencia $H(z)$.

Además, existen otras categorías, por ejemplo, los filtros multirate (que aprovechan la propiedad de cambiar la velocidad de muestreo internamente, para mejorar la performance del sistema) y los filtros ecualizadores, que actúan simultáneamente sobre distintas bandas, reduciendo la capacidad de procesamiento necesaria.

2. Descripción

Esta experiencia persigue dos objetivos, a saber, introducir al alumno al Análisis de Fourier para observar los resultados de distintos procesos y su reflejo en el espectro de frecuencias. Por otro lado, se busca realizar análisis del comportamiento de filtros de diversa naturaleza. Esta experiencia se realizará utilizando la herramienta Matlab.

Para los análisis de Fourier, se utilizarán las señales empleadas en la primera experiencia de este curso. Por lo tanto, se deberá utilizar como entrada el archivo **acoustic.wav** que está almacenado dentro del archivo comprimido **muestrasguitarra.zip**. Adicionalmente, se utilizarán las funciones senoidales que se especifican en la sección Desarrollo de la presente Guía.

Para realizar los análisis, se utilizarán las funciones de Matlab llamadas **fft()**, que permite obtener una

representación de la Transformada de Fourier para señales discretas, y `spectrogram()`, que entregan una representación tiempo(muestras)/frecuencia/potencia-frecuencia de cada señal.

Para el análisis de filtros, se utilizará la herramienta *filterDesigner* de Matlab, para analizar el comportamiento detallado de los filtros, las limitaciones y las herramientas de diseño disponibles. Luego, se ensaya el diseño de manera sencilla con la herramienta gráfica Simulink, parte de la suite de Matlab. Ambas aplicaciones son gráficas, no requiriendo programación alguna de parte del alumno, sólo comprensión, extracción de resultados y análisis.

Los objetivos de esta parte del Laboratorio son:

- Diseñar filtros de acuerdo a restricciones utilizando *filterDesigner*.
- Estudiar las diferencias entre los tipos de filtros diseñados.
- Definir y utilizar una metodología tanto para realizar las mediciones como para resumir y presentar los resultados, explicando de manera correcta y con lenguaje técnico estos fenómenos.

3. Desarrollo

3.1. Análisis de Fourier

Utilice las funciones de Matlab que permitan obtener la Transformada de Fourier y el espectrograma y aplíquelas a las siguientes señales:

1. Señal sinusoidal de amplitud unitaria y frecuencia 200 Hz, 1 segundo de duración.
2. Señal sinusoidal de amplitud 10 y frecuencia 500 Hz, 1 segundo de duración.
3. Suma de las señales indicadas en los puntos previos.
4. Agregar a la suma de las señales, un ruido blanco de distribución normal (función `randn()` de Matlab).
5. Señal generada desde el archivo **acoustic.wav**

Utilizar la función `fft()` para calcular la Transformada de Fourier de cada señal, y la función `spectrogram()` para calcular el espectrograma.

Obtenga los gráficos correspondientes para cada una de las Transformadas de Fourier y los espectrogramas de cada señal solicitada.

Conteste las siguientes preguntas, como parte de sus análisis:

- ¿Son sus resultados coherentes? ¿Por qué?
- ¿Dónde se concentra la potencia de cada señal y por qué?
- Incluya en su informe toda otra observación que le parezca relevante.

3.2. Filtros

1. El primer paso para iniciar el diseño de filtros con *filterDesigner* es invocarlo en el prompt de Matlab:

```
>> filterDesigner
```

2. Luego se despliega la aplicación mostrada en la Figura 2.

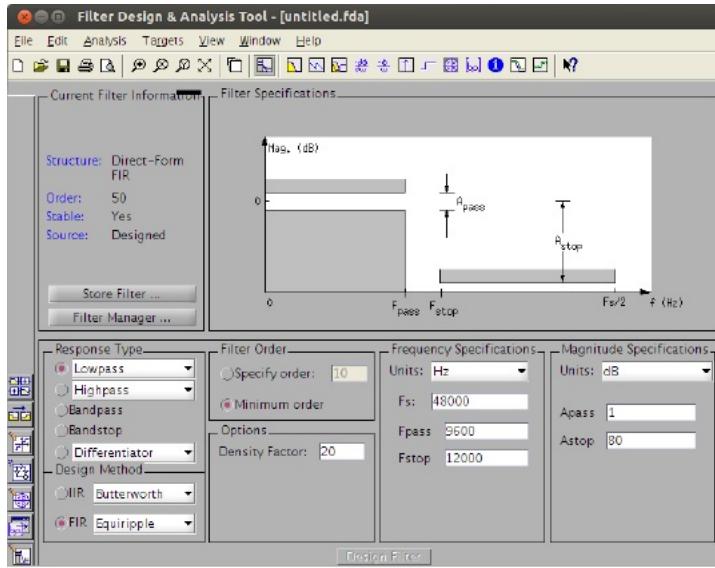


Figura 2: Filter Designer de Matlab.

3. Los filtros digitales se diseñan utilizando una estructura básica de cuociente de polinomios, que en su forma más elemental responde a la solución de una ecuación diferencial. El objetivo es ajustar un polinomio cuyos coeficientes se aproximen de la manera más precisa a las restricciones planteadas a partir de los parámetros de diseño: frecuencia de corte, frecuencia de paso, ripple de la banda de paso, atenuación en la banda de rechazo. Una vez que se obtienen los filtros, a través de un método de diseño, éstos se incluyen en la función de cuociente de polinomios para convertirlo en una etapa de procesamiento. Cada coeficiente implica un producto y una suma: la aplicación de un filtro a una señal de entrada se realiza multiplicando los coeficientes por la muestra más reciente de la señal y por muestras anteriores, según el grado de los polinomios del numerador y del denominador.
4. El resultado de aplicar este cuociente de polinomios a una señal de entrada debe ser intrínsecamente estable; sin embargo, existen expresiones que pueden generar inestabilidades, tales como oscilaciones y amplificaciones para ciertas frecuencias. Para verificar la estabilidad de un filtro, se utiliza el análisis de polos y ceros. Un polo es una raíz del polinomio presente en el denominador, mientras que un cero es una raíz del polinomio presente en el numerador. Este análisis se generaliza a partir de la R.O.C de la Transformada Z.
5. Existen dos tipos fundamentales de filtros: FIR e IIR. Los filtros FIR no se pueden construir de manera analógica, sólo existen en los sistemas de procesamiento digital de señales. Estos filtros se caracterizan por tener sólo elementos en el numerador, por lo tanto son intrínsecamente estables (no hay lugares donde el denominador se haga cero, haciendo el resultado infinito). Los filtros IIR son similares a los que se pueden construir de manera analógica, cuyos métodos de obtención más comunes se denominan Butterworth, Chebishev y Elíptico. Cada uno de ellos tiene sus ventajas respecto a los demás. Estos filtros tienen la principal desventaja de generar coeficientes de un gran rango dinámico, lo que, en primer lugar, fuerza a utilizar cálculo de punto flotante para polinomios de mayor orden, y en segundo lugar, puede generar inestabilidades. Sin embargo, la ventaja de estos filtros es tener la capacidad de obtener una respuesta similar a un FIR, pero con una cantidad de operaciones considerablemente menor. La decisión acerca de utilizar un filtro FIR o uno IIR, y en específico cuál modelo a utilizar dentro de cada categoría, dependerá, entre otros factores, de la aplicación buscada, la cantidad de recursos disponibles, la complejidad de implementación, etc.
6. Para comparar el comportamiento de ambos tipos de filtro, el primer paso es definir las restricciones. Como se indica en la Figura 2, el gráfico describe el tipo de filtro elegido en el plano de frecuencias(Hz)-magnitud(dB). La elección por defecto es un pasabajo. Para este tipo de filtro, existen las restricciones:

- A_{pass} : Es la amplitud tolerada en la respuesta del filtro dentro de la banda de paso. Se entrega en dB. Equivale al ripple pasabanda. Dentro de esa banda, el polinomio puede variar libremente en su respuesta.
- A_{stop} : Es la atenuación mínima exigida al filtro dentro de la banda de rechazo. Por debajo de este límite el polinomio puede variar libremente en su respuesta.
- F_{pass} : Es la frecuencia límite de paso de la señal. A partir de esta frecuencia, el filtro empieza a atenuar la señal hasta la frecuencia F_{stop} .
- F_{stop} : Es la frecuencia límite donde comienza el rechazo de la señal. A partir de esta frecuencia, debe respetarse el parámetro A_{stop} . La banda entre la F_{pass} y la F_{stop} se denomina banda de transición.
- En el caso de un filtro pasabanda o eliminabanda, existen dos F_{pass} y dos F_{stop} .

7. Los valores anteriores se cargan en los frames “Frequency Specifications” y “Magnitude Specifications”, respectivamente, en las unidades seleccionadas en la caja de selección superior. Para realizar una primera comparación, se plantea analizar las diferencias entre un FIR y un IIR, pasabajos.

8. Se colocan los siguientes valores a los parámetros:

Fs=50000 Hz
Fpass=3000 Hz
Fstop=5000 Hz
Apas=2 dB
Astop=80 dB

9. Para diseñar el filtro, se hace click en el botón “Design Filter” en la parte inferior y el resultado se observa en la parte superior. Este es el polinomio que más se ajusta utilizando el método de diseño “Equiripple” para los filtros FIR. En el frame de la izquierda se indica la estructura final (Direct Form). Existen otras estructuras más complejas, que no forman parte del alcance de este laboratorio. También se indican el orden (cantidad de polos en el denominador), si es estable y el estado del resultado (Designed).
10. El mismo trabajo se deberá realizar, ahora con un filtro IIR tipo Butterworth. En la esquina inferior izquierda, se modifica la selección del tipo de filtro dentro del cuadro “Design Method”, sin modificar las condiciones de diseño anteriores. Se vuelve a solicitar el diseño del filtro y se observa el resultado.
11. Responder: ¿Qué diferencias en el orden y respuesta de magnitud se observaron? En especial, notar que aparece un parámetro más, denominado “Sections”, cuando se diseña un filtro IIR. Este valor corresponde a la cantidad de etapas de polinomios de segundo orden que se requieren para satisfacer el diseño IIR. En particular, los IIR se diseñan construyendo una cadena de filtros más simples de segundo orden.
Incluir las capturas de pantalla de cada filtro, para sustentar sus análisis.
12. Los filtros no solamente tienen una respuesta en magnitud, sino también una respuesta en fase. Este último parámetro también cambia en función de la frecuencia, dado que los filtros se realizan con polinomios de coeficientes complejos. En la barra superior se encuentran distintas alternativas para comprender el diseño del filtro. La vista combinada Magnitud-Fase es el cuarto botón de la barra, contado desde la izquierda, según se puede observar en la Figura 3. Este diagrama de frecuencia logarítmica en el eje x, y atenuación en el eje y, se denomina Diagrama de Bode.



Figura 3: Barra de vistas

13. Responder: ¿Cómo se comporta la fase en la banda de paso y de rechazo en ambos tipos de filtros?
 Capturar las figuras necesarias para sustentar sus análisis.
14. El análisis temporal al impulso y a una función escalón muestra las condiciones de mayor exigencia al filtro, en especial el tiempo que tarda en estabilizarse y el máximo valor de la señal en porcentaje respecto del valor del impulso (sobrepasso). Se puede observar este valor en el gráfico de respuesta a escalón, donde en la transición en el filtro IIR, la señal de salida sobrepasa la señal de entrada, cuya amplitud es 1. Por otro lado, se puede observar el retardo de la salida con respecto de la entrada, dado que ambas señales (impulso y escalón) son aplicadas en el instante 0 a la entrada del filtro. Los gráficos de respuesta a impulso y respuesta a escalón pueden observarse en los gráficos asociados a los botones séptimo y octavo, respectivamente, contados desde la izquierda, en la barra de vistas.
15. Finalmente, para exportar el filtro a la herramienta de simulación Simulink, se utiliza el cuarto botón de la barra de diseño, comenzando desde arriba. El botón “Realize Model” genera el módulo Simulink. Allí se puede probar el diseño utilizando un generador de barrido “Chirp” (genera una forma de onda de frecuencia creciente configurable), y un “Time Scope”, ambos extraídos de la librería “DSP System Toolbox” de Simulink. Esta configuración se puede observar en la Figura 4. Probar la respuesta de ambos filtros (FIR e IIR), analizar la respuesta y realizar las capturas de gráficos necesarias para sustentar sus análisis. La configuración del generador Chirp se indica en la Figura 5. El generador Chirp permitirá producir un barrido en un rango de frecuencias, en un lapso de tiempo determinado. Esto se puede hacer una vez, o bien repetirse periódicamente.

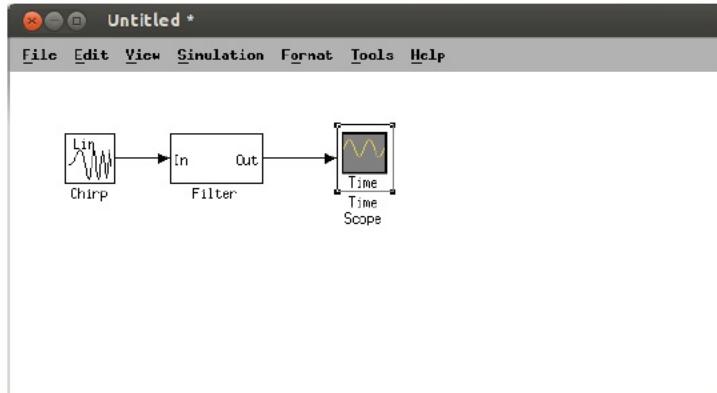


Figura 4: Filtro en Simulink

16. Realizar un análisis comparativo considerando:

- Respuesta en amplitud.
- Respuesta en fase.
- Respuesta al impulso.
- Respuesta a la función escalón.
- Cantidad de operaciones de suma y producto necesarias.
- Efecto de la cuantización entre double precision y fixed point en 16 bits.
- Respuesta al Chirp en Simulink.

Para lo solicitado, se debe construir un filtro pasabanda FIR, IIR Elíptico con las siguientes características:

- Frecuencia de corte 1: 2000Hz.
- Frecuencia de paso 1: 3000Hz.
- Frecuencia de paso 2: 5000Hz.

- Frecuencia de corte 2: 5500Hz.
- Mínima atenuación 1: 80dB.
- Mínima atenuación 2: 90dB.
- Máxima variación de amplitud en la banda de paso: 1dB.

Realizar las capturas de las figuras que requiera, para sustentar los análisis comparativos.

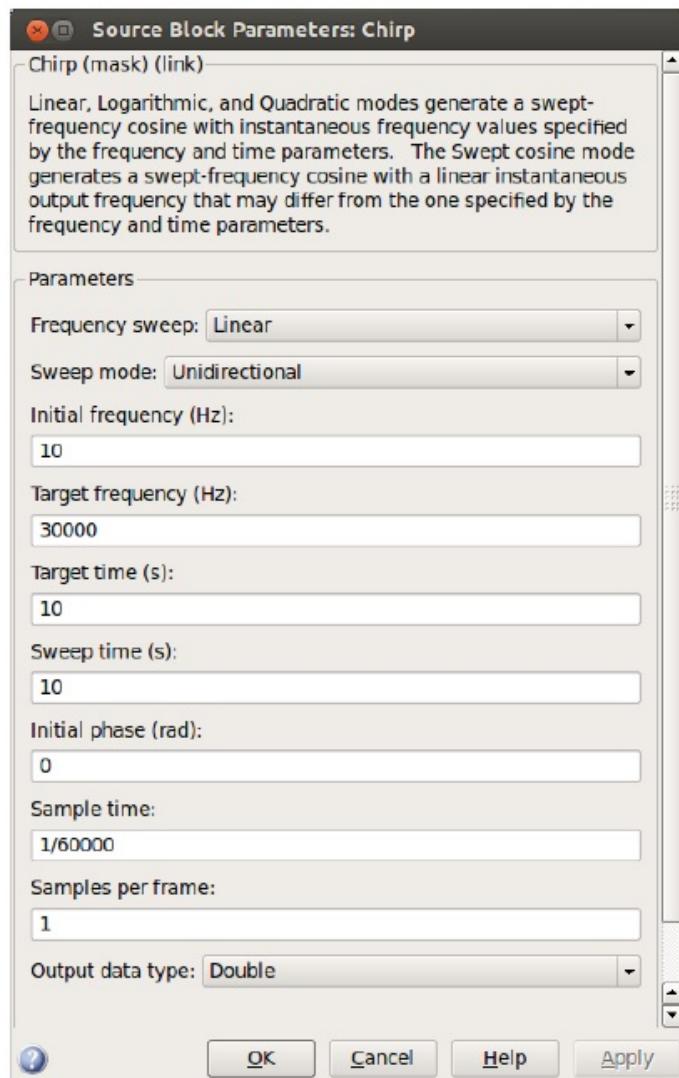


Figura 5: Configuración del generador Chirp

4. Informe Final

Redacte un Informe Final, **que no sobrepase las 3.000 palabras (aproximadamente 6 páginas)**, que contenga:

1. Carátula.
2. Resumen.
3. Introducción que describa cada una de los problemas planteados en esta guía.
4. Descripción de los algoritmos desarrollados/tareas ejecutadas.

5. Resultados (numéricos y gráficos). Adjuntar archivos con el código Matlab, y otros archivos multimedia según corresponda.
6. Análisis de resultados, incluyendo explicaciones sobre los efectos del procesamiento sobre las señales de entrada, posibles impactos y aplicaciones.
7. Dificultades encontradas.
8. Conclusiones.
9. Bibliografía.

5. Pauta de Corrección

Para la corrección del Informe Final de esta Experiencia se tomará en cuenta:

- Condición necesaria, pero no suficiente de aprobación: Los códigos y proyectos generados deben operar de manera correcta.
- Ortografía: No se tolerarán más que 2 errores ortográficos como máximo (descuento de 1 décima por error ortográfico adicional).
- Presentación: Todas las secciones mencionadas deben existir en el informe.
- Extensión y formato: Debe respetarse la extensión máxima. Se descontará puntoje por informes que excedan la restricción de cantidad de palabras. No debe usarse Wikipedia como cita bibliográfica.
- Calidad del análisis de los resultados: Debe contener de manera clara y precisa la explicación de por qué se observan los comportamientos, de acuerdo a los resultados.
- Presentación de gráficos: Las figuras y tablas deben tener ejes en las unidades y etiquetas correspondientes. Adicionalmente, todo gráfico y tabla debe poseer la numeración y etiquetas correspondientes, y deben ser generadas con gráficos vectoriales. No se deben generar en formato tipo pixel.
- Claridad y especificidad en la explicación de la metodología en la introducción y su coherencia con los resultados mostrados.
- Las conclusiones deben resumir los puntos importantes del informe, así como sus hallazgos, evidencias y observaciones más relevantes, a la luz del marco teórico que sustenta la Experiencia. Expresiones del tipo "se cumplieron los objetivos de la Experiencia", sin mayor evidencia, no son un aporte a las Conclusiones, y no recibirán calificación.