



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
*FCITEC, Valle de las Palmas*

## ***MANUAL DE PRÁCTICAS DE LABORATORIO DE PDSB***

**Unidad de Aprendizaje:** *PROCESAMIENTO DIGITAL DE  
SEÑALES BIOFISIOLÓGICAS*  
**Clave de la Asignatura:** *11808*  
**Programa Educativo:** *Bioingeniería*

**Elaborado y actualizado por:**

Dra. Norma Alicia Barboza Tello  
Dr. Paúl Medina Castro  
Mtro. Miguel Alejandro Díaz Hernández  
M.I. Irma Uriarte Ramírez  
Dra. Daniela Mercedes Martínez Plata  
Dr. Allen Alexander Castillo Barrón

*Valle de las Palmas, Tijuana, B.C., Noviembre 2018.*

# **¡ANTES DE INICIAR LAS PRÁCTICAS DE LABORATORIO!**

## **REGLAMENTO DEL LABORATORIO DE PDSB**

1. Queda prohibido: Introducir alimentos y bebidas.
2. Queda prohibido: fumar, gritar, correr y sentarse en las mesas de trabajo
3. Se deben respetar las condiciones de seguridad y funcionalidad de laboratorio.
4. Deberá apagar la computadora al terminar la práctica de laboratorio.
5. No mover, sustraer, manipular o hacer uso indebido del equipo sin autorización.
6. Prohibido visitas no autorizadas.
7. Deberá mantener el laboratorio ordenado y limpio.

## ÍNDICE

Práctica No. 1. Introducción a MATLAB.....	4
Práctica No. 2. Teorema de Muestreo y “aliasing”.....	5
Práctica No. 3. Cuantificación de señales discretas.....	6
Práctica No. 4. Convolución y correlación de señales discretas.....	7
Práctica No. 5. Series de Fourier.....	8
Práctica No. 6. Transformada de Fourier.....	9
Práctica No. 7. Espectro de densidad de potencia, teorema de Parseval.....	10
Práctica No. 8. Filtro digital rechaza-banda.....	11
Práctica No. 9. Filtros digitales Pasa-bajas.....	12
Práctica No. 10. Filtros digitales Pasa-altas.....	13
Práctica No. 11. Filtrado digital de señales ECG.....	14
Práctica No. 12. Filtrado de señales por ventaneo.....	15



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 1. Introducción a Matlab.**

**COMPETENCIA**

Representar gráficamente diferentes funciones de señales a través de la utilización del software MATLAB para identificar sus componentes de amplitud y frecuencia con actitud creativa, honesta y responsable.

**INTRODUCCIÓN**

Matlab es un software matemático que es utilizado principalmente por ingenieros como una herramienta para la resolución de problemas a través de la programación utilizando un lenguaje M, el cual es único de la plataforma. El lenguaje M tiene similitud con el lenguaje C o C++, y cuenta con una gran variedad de funciones predeterminadas de uso común tales como las funciones trigonométricas (seno, coseno, tangente, etc.), operaciones simples como sumas, restas, raíz cuadrada hasta funciones más complejas de procesamiento de señales o imágenes, entre otras. Algunas de las herramientas más utilizadas de Matlab son las herramientas para graficar, a través de diferentes funciones como “plot” o “stem” se puede obtener fácilmente la forma de onda de una señal o varias señales en una o varias ventanas.

Recordando que una señal puede definirse como una variación de una variable física con respecto a otra variable (e.g. temperatura con respecto al tiempo) y la información en una señal está contenida en un patrón de variaciones que presenta alguna forma de onda determinada por ejemplo una señal senoidal, tiene una forma de onda específica. Estas señales se pueden clasificar en continuas o discretas, las señales continuas son aquellas donde la variable independiente varía continuamente con el tiempo, por lo que se definen como una sucesión continua de valores de la señal; y las señales discretas, son aquellas en las que la variable independiente solo puede tomar un conjunto discreto de valores, es decir, toma valores determinados cada valor finito de tiempo. En el desarrollo de esta práctica se graficarán diferentes funciones que dependen del tiempo y se realizarán algunas operaciones sobre éstas con el fin de familiarizarse con las funciones de despliegue de señales de Matlab e identificar las propiedades que caracterizan a

cada señal, como son, amplitud, frecuencia y periodo.

**MATERIAL Y EQUIPO**

- 1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

1. Desarrolle en el editor de MATLAB un código para evaluar las siguientes funciones y obtenga la gráfica de cada una de ellas utilizando la función “plot”.

$$\begin{aligned} a(t) &= 5\text{sen}2t \\ b(t) &= 3\text{cos}5t \\ c(t) &= a(t) \times b(t) \\ e(t) &= a(t) + b(t) \\ f(t) &= a(t) - b(t) \end{aligned}$$

2. Compare los resultados obtenidos en los incisos d y e, y comente sobre ellos.
3. Aumente la amplitud de  $a(t)$  5 veces y obtenga las gráficas para cada uno de los incisos anteriores c al e. Comente las diferencias.
4. Grafique la señal  $x(t)$  utilizando un vector de tiempo que tome valores cada 1ms con un intervalo de 0 a 1.

$$x(t) = \text{sen}(40t) + \text{sen}(80t)$$

5. Ahora genere un vector tiempo que tome valores cada 100ms y grafique de nuevo ambas señales utilizando la función subplot, de tal forma que pueda comparar las gráficas para  $T=1\text{ms}$  y  $T=100\text{ms}$  y comente.
6. Modifique la frecuencia a 2 veces la frecuencia máxima de la señal  $x(t)$  y gráfiquela.
7. Modifique la frecuencia a 10 veces la frecuencia máxima de la señal y gráfiquela.
8. Concluya sobre resultados observados y las funciones utilizadas al realizar la práctica.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

Práctica No.2. Teorema de Muestreo y “aliasing”.

COMPETENCIA

Comprender la importancia del teorema de muestreo a través de la elaboración de un algoritmo en MATLAB que represente gráficamente señales muestreadas a diferentes valores de frecuencia de muestreo para identificar el fenómeno de “aliasing”, con actitud honesta y responsable.

INTRODUCCIÓN

Una señal en tiempo discreto se obtiene si se muestrea cada cierto periodo de tiempo, una señal en tiempo continuo  $x_a(t)$ . La relación entre las señales está dada por:  $x_d(n) = x_a(nT)$ , donde la variable de tiempo se relaciona con la variable de tiempo  $n$ , durante solo en instante de tiempo discreto  $t_n$  dados por:  $t_n = nT = \frac{n}{F_s}$ . Si la frecuencia más alta contenida en una señal analógica  $x_a(t)$  es  $F_{max}$  y la señal se muestrea a una tasa  $F_s > 2F_{max}$ , entonces  $x_d(n)$  se puede recuperar totalmente a partir de sus muestras mediante alguna función de interpolación. A esta condición se le conoce como *teorema de muestreo*, es decir, el teorema de muestreo establece que una señal es recuperable si se muestrea a más de dos veces la frecuencia máxima de esta señal. Si la señal se muestrea a exactamente dos veces la frecuencia máxima se le conoce como *frecuencia de Nyquist*  $f_N$ , tasa de Nyquist o velocidad de Nyquist. Si no se cumple la condición de muestreo, existirán frecuencias que coincidan con otras produciendo el fenómeno de “aliasing”, es objetivo de esta práctica observar este fenómeno y para su desarrollo se utilizará la siguiente función de interpolación:

$$x_d(t) = \sum x_d\left(\frac{n}{F_s}\right) g\left(t - \frac{n}{F_s}\right)$$

donde:

$$g(t) = \frac{\text{sen}(\pi F_s t)}{\pi F_s t}$$

MATERIAL Y EQUIPO

1 Computadora con software MATLAB instalado.

PROCEDIMIENTO

1. Grafique la señal  $x(t)$  utilizando un periodo de tiempo cuasi-continuo a fin de simular una señal analógica.

$$x(t) = \text{sen}(4\pi t) + \text{sen}(8\pi t)$$

2. Simule el muestreo de la señal  $x(t)$  a una frecuencia de 10Hz y gráfiquela. Comente sobre la forma de onda de la gráfica.

3. Implemente la función de interpolación arriba mencionada a través del desarrollo de un algoritmo, para recuperar la señal muestreada a 10Hz y gráfiquela utilizando la función “hold on”. Compare las gráficas y comente.

4. Simule el muestreo de una onda sinusoidal pura de 300Hz. Utilizando una frecuencia de muestreo de 800Hz.

5. Simule el muestreo de las ondas sinusoidales puras cuyas frecuencias se indican a continuación:

- a) 125Hz
- b) 215Hz
- c) 305Hz
- d) 395Hz
- e) 500Hz

6. Observe los cambios en las formas de onda de la señal muestreada y la señal recuperada. Grafique en la misma ventana la señal original, la señal muestreada y la señal recuperada, utilizando una frecuencia de muestreo de 200Hz.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 3. Cuantificación de señales discretas.**

**COMPETENCIA**

Comprender en qué consiste la técnica de cuantificación de señales a través del uso de las técnicas de redondeo y truncamiento para identificar su importancia en el proceso de conversión analógico-digital, con actitud honesta y responsable.

**INTRODUCCIÓN**

La técnica de cuantificación se refiere al proceso en el que una señal analógica se aproxima a una señal que puede tomar valores de amplitud finitos. Uno de los factores importantes a considerar en el proceso de cuantificación y que proporciona información importante sobre la calidad de un convertidor analógico-digital es el error de cuantificación. Éste se interpreta como un ruido añadido a la señal después de decodificarla y se representa por  $e(n)$ :

$$e(n) = x_q(n) - x(n)$$

donde  $x_q(n)$ , es la señal cuantificada y  $x(n)$  la señal discreta. El ruido de cuantificación no tiene gran efecto si éste se mantiene en niveles muy por debajo de los niveles de la señal discreta, para esto hay que considerar la relación señal a ruido (SQNR, *Signal to Quantization Noise Ratio*) que podría representarse como un parámetro que nos determina qué tan ruidosa es una señal y se define por:

$$SQNR = 10 \log_{10} \frac{P_x}{P_q}$$

donde:

$$P_q = \frac{1}{N} \sum_{n=0}^{N-1} e^2(n) = \frac{1}{N} [x_q(n) - x(n)]^2$$

es la potencia de la señal cuantificada y  $P_x$  la potencia de la señal discreta dada por:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} [x(n)]^2$$

En esta práctica se desarrollará una técnica para cuantificar algunas señales discretas con diferentes valores de resolución y el impacto de este parámetro en la calidad de la señal.

**MATERIAL Y EQUIPO.**

- 1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

Sea:

$$x(n) = 10 \sin 2\pi f_0 n,$$

1. Para  $f_0 = 50$  y  $F_s = 200$ , escriba un programa para cuantificar la señal  $x(n)$  utilizando truncamiento, con 64, 128 y 256 niveles de cuantificación.
2. Utilice el algoritmo de la práctica anterior para recuperar las señales una vez cuantificadas. En cada caso grafique las señales  $x(n)$ ,  $x_q(n)$ , y  $e(n)$ , y la señal recuperada. Compare gráficas.
3. Repita los pasos 2 y 3 pero utilizando la técnica de redondeo.
4. Comente en tu reporte sobre las diferencias y observaciones encontradas entre los distintos niveles de cuantificación utilizados.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 4. Convolución y correlación de señales discretas.**

**COMPETENCIA**

Comprender las operaciones de convolución y correlación cruzada a través del uso de diferentes funciones de MATLAB para comprender su utilidad en el procesamiento de señales biofisiológicas con actitud propositiva, honesta y responsable.

**INTRODUCCION**

**Convolución de Señales Discretas**

La mayoría de las señales discretas en el tiempo que se encuentran en la práctica pueden representarse como una combinación lineal de un gran número de señales senoidales discretas en el tiempo con diferentes frecuencias angulares. Así, si se conoce la respuesta de un sistema LTI (Linear Time-Invariant System) a una sola señal senoidal, podemos determinar su respuesta a señales más complicadas utilizando la propiedad de superposición del sistema. Una consecuencia de las propiedades de linealidad e invarianza en el tiempo es que la respuesta de un sistema LTI, está perfectamente descrita por su respuesta al impulso, es decir que, conociendo su respuesta al impulso, se puede determinar la salida del sistema ante cualquier entrada arbitraria. La operación para conocer la respuesta al impulso de cualquier sistema LTI, se llama suma de convolución y se representa por:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n - k)$$

Donde  $x(k)$  representa la entrada arbitraria y  $h(n)$  la respuesta del sistema. Para encontrar la convolución de dos señales discretas hay que seguir los siguientes pasos:

- Obtener el reflejo de  $h(n)$ .
- Desplazar hacia la derecha todo el vector de  $h(n)$ .
- Multiplicar el primer valor de  $h(n)$  por el último de  $x(n)$ .
- Sumar los productos.
- Desplazar a la derecha  $x(n)$  y multiplicar los valores que coincidan con  $h(n)$ .
- Sumar los productos.
- Seguir desplazando a la derecha  $x(n)$  hasta que se termine toda la serie discreta.

**Correlación Cruzada**

La correlación es una operación matemática muy parecida a la convolución. La correlación permite identificar la similitud que existe entre dos señales y así extraer información que dependerá de la aplicación para que son detectadas. La operación de correlación se define por:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n - l)$$

y la autocorrelación de  $x(n)$  se define como:

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n - l)$$

Donde  $x(n)$  y  $y(n)$  representan las señales discretas. Para encontrar la correlación de dos señales discretas hay que seguir los siguientes pasos:

- Desplazar hacia la derecha todo el vector de  $h(n)$ .
- Multiplicar el primer valor de  $h(n)$  por el último de  $x(n)$ .
- Sumar los productos.
- Desplazar a la derecha  $x(n)$  y multiplicar los valores que coincidan con  $h(n)$ .
- Sumar los productos.
- Seguir desplazando a la derecha  $x(n)$  hasta que se termine toda la serie discreta.

**MATERIAL Y EQUIPO**

1 computadora con software MATLAB instalado.

**PROCEDIMIENTO**

- Diseñe un algoritmo en MATLAB que realice la operación de convolución de dos señales discretas, apoyándose de las funciones "for" e "if".
- Utilice la función "conv" de MATLAB para comprobar los resultados de su algoritmo.
- Diseñe un algoritmo en MATLAB que realice la operación de correlación cruzada de dos señales discretas, apoyándose de las funciones "for" e "if".
- Utilice la función "xcorr" de MATLAB para comprobar los resultados de su algoritmo.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 5. Series de Fourier.**

**COMPETENCIA**

Identificar las componentes de la serie de Fourier de distintas funciones a través del diseño de una función de MATLAB que calcule los coeficientes correspondientes, para comprender su utilidad en la representación espectral de señales discretas, de manera honesta y responsable.

**INTRODUCCION**

Mediante las Series de Fourier cualquier señal periódica puede ser representada como la suma trigonométrica de senos y cosenos en el dominio del a frecuencia. Para obtener la serie de Fourier es necesario calcular los tres coeficientes que representaran el comportamiento de la señal periódica. El primer coeficiente  $a_0$  representa una componente de corriente directa en la señal y se calcula a través de la ecuación:

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt$$

Para el cálculo de los demás coeficientes se utilizan las integrales:

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) dt$$

y

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) dt$$

Una vez calculados los coeficientes la señal periódica se representa:

$$x(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 + b_n \sin n\omega_0 t)$$

En esta práctica se desarrollará un algoritmo que permita calcular numéricamente los coeficientes de la serie de Fourier de distintas señales periódicas con el fin de conocer la contribución de cada coeficiente a la representación de ésta, así como identificar el fenómeno de oscilaciones de *Gibbs*.

**MATERIAL Y EQUIPO**

1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

Escriba un código en MATLAB que obtenga los coeficientes de Fourier  $a_0$ ,  $a_n$  y  $b_n$  para señales discretas rectangulares. Utilice ciclos for y las funciones de matlab "sym", "linspace", "input" y "simple".





UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 6. Transformada de Fourier.**

**COMPETENCIA**

Conocer la aplicación de la transformada rápida de Fourier a través de la utilización de la función “fft” de Matlab para conocer el espectro de densidad de potencia de señales periódicas con actitud responsable y honesta.

**INTRODUCCIÓN**

La Transformada de Fourier es una herramienta matemática que permite transformar una señal que se encuentra en el dominio del tiempo al dominio de frecuencia, esta herramienta, por lo tanto, permite visualizar el espectro de frecuencias de cualquier señal. Tiene la característica de ser reversible, lo que permite cambiar entre el dominio del tiempo al dominio de la frecuencia y viceversa según sea requerido en el estudio.

La Transformada de Fourier tiene muchas aplicaciones en la ingeniería, especialmente para la caracterización frecuencial de señales y sistemas lineales, así como para la aplicación de filtros digitales. La Transformada de Fourier se define por:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

donde  $F(\omega)$  es la función o señal en el dominio de la frecuencia (generalmente se escribe en mayúsculas para diferenciarla de la señal en el dominio del tiempo,  $f(t)$  representa la función en dominio del tiempo y  $e^{-j\omega t}$  representa la constante de normalización. Parecería que las Series de Fourier y la Transformada de Fourier son lo mismo pues ambas permiten obtener el espectro de frecuencias de una señal. Sin embargo, a diferencia de las Series de Fourier, la Transformada puede aplicarse tanto a señales periódicas como no periódicas.

**MATERIAL Y EQUIPO**

- 1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

1. Escribir un programa en MATLAB que represente las siguientes señales muestreadas:

$$y = \sin(20\pi)$$

$$y = 2\sin(6\pi)$$

$$y = 3e^{-2\pi}$$

$$y = \frac{1}{2}\sin(10t) - \frac{1}{2}\cos(4t)$$

2. Utilice la función “fft” de Matlab para calcular la Transformada de Fourier de cada una de las señales.
3. Utilice la función “ifft” de Matlab para calcular la Transformada Inversa de Fourier de cada una de las señales.
4. Muestre las gráficas de las señales, de la Transformada de Fourier y de calcular la Transformada Inversa de Fourier, de cada una de las señales.
5. ¿Se lograron recuperar las señales originales a través de la Transformada Inversa de Fourier?
6. ¿Qué aplicación consideran que podría darse a la Transformada Inversa de Fourier?



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

**Práctica No. 7. Espectro de Densidad de Potencia.**

**COMPETENCIA**

Conocer la aplicación de la transformada rápida de Fourier a través de la utilización de la función “fft” de Matlab para conocer el espectro de densidad de potencia de señales periódicas.

**INTRODUCCIÓN**

El análisis de Fourier es un conjunto de técnicas matemáticas basadas en descomponer una señal en senoidales. La transformada de Fourier Discreta (DFT, Discrete Fourier Transform) es la herramienta utilizada cuando se trabaja con señales discretas, en la práctica la DFT se calcula en forma eficiente mediante la Transformada de Fourier Rápida (FFT, Fast Fourier Transform) tiene diversas aplicaciones en Procesamiento de señales como: Análisis espectral, Convolución rápida, Síntesis de sonido por modelo espectral, Filtrado de señales, etc.

Las señales pueden clasificarse según sean de energía o de potencia. Se dice que una señal es de energía si su energía es finita, lo que implica que su potencia es cero. Un ejemplo de señal de energía son los pulsos limitados en el tiempo. Se dice que una señal es de potencia si su potencia es finita, lo que implica que su energía es infinita. Un ejemplo de este tipo de señales lo encontramos en las señales periódicas. Recordemos que de acuerdo al *Teorema de Parseval* el espectro de densidad de energía puede determinarse a través de la relación:

$$P_x = \sum_{n=1}^{\infty} C_n^2$$

Para el cálculo de estos espectros se dispone de la función en Matlab “psd”, esta función proporciona la densidad espectral de potencia de una señal en dB.

**MATERIAL Y EQUIPO**

1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

1. Escriba un programa en Matlab que represente las siguientes señales muestreadas:
  - a)  $y = 0.3\sin 2\pi 100t + 0.7\sin 2\pi 50t$
  - b)  $y = 2\sin 6\pi t + 0.5\sin 2\pi 18t$
  - c)  $y = 3e^{-2\pi t}$
2. Utilice la función “fft” de Matlab para obtener el espectro en frecuencias de cada una de las señales.
3. Utilice la función “psd” de Matlab para obtener el espectro de densidad de potencia de las mismas señales.
4. Muestre las gráficas de las señales, de sus transformadas de Fourier y de los espectros de potencia de cada una.
5. ¿Qué puede observar en cada uno de los espectros?



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

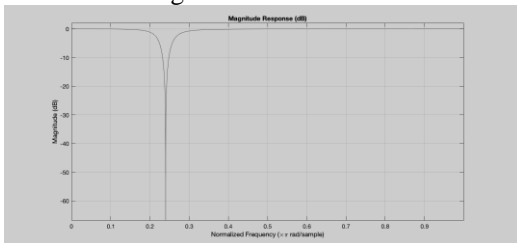
**Práctica No. 8. Filtro Rechaza-banda.**

**COMPETENCIA**

Conocer la importancia de los parámetros de un filtro digital rechaza-banda a través del diseño de un filtro Notch para eliminar componentes de frecuencia específicas con actitud honesta y responsable.

**INTRODUCCIÓN**

Un filtro digital puede definirse como una operación matemática que se aplica sobre una señal con el fin de resaltar o atenuar algunas de sus características. Los filtros digitales al igual que los analógicos se clasifican de acuerdo a la función que realizan y a su respuesta en frecuencia. El filtro digital rechaza-banda, también conocido como filtro de ranura se encarga de eliminar ciertas componentes de frecuencia, se caracteriza por tener ceros perfectos en su respuesta en frecuencia, como se muestra en la figura.



La figura muestra la respuesta en frecuencia de un filtro Notch, el eje “x” está normalizado a una frecuencia de 250Hz, por lo tanto, la frecuencia de cortes de 60Hz.

La función transferencia más general que representa un filtro Notch está dada por:

$$H(z) = b_0(1 - 2\cos\omega_0 z^{-1} + z^{-2})$$

El problema de este tipo de filtros es que suelen tener un ancho de banda muy grande, atenuando frecuencias próximas a la frecuencia de corte del filtro. Sin embargo, existen filtros más sofisticados como el filtro Notch que permite ser más selectivo en la frecuencia a eliminar. En esta práctica se implementará un algoritmo para eliminar frecuencias específicas en varias señales periódicas.

**MATERIAL Y EQUIPO**

- 1 Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

1. Genere y grafique en MATLAB una señal de la forma:

$$x(t) = 3\sin 45t + 4\sin 50t + 3\sin 55t + 4\sin 60t$$

2. Obtenga el espectro de frecuencias de la señal y observe las componentes de frecuencia de la señal.
3. Diseñe un filtro Notch utilizando la función “iirnotch” con frecuencia de central de 50Hz y el ancho de banda que considere adecuado, (puede graficar la respuesta del filtro utilizando la función “fvtool”).
4. Aplique el filtro notch a la señal generada en el punto 1, utilizando la función “filter”.
5. Grafique la señal filtrada.
6. Repita los pasos 3 al 4 para un ancho de banda 20 veces mayor al que seleccionó.
7. Repita los pasos 3 al 4 para un ancho de banda 20 veces menor al que seleccionó.
8. Concluya sobre los resultados obtenidos.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

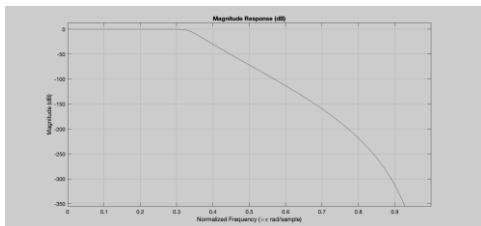
**Práctica No. 9. Filtros Digitales Pasa-Bajas.**

**COMPETENCIA**

Conocer la respuesta en amplitud y fase de distintas funciones de filtros pasa bajas a través de la utilización de distintas funciones de Matlab para conocer su aplicación en el filtrado de señales biofisiológicas.

**INTRODUCCIÓN**

Un filtro digital, es un algoritmo que opera sobre señales digitales, y tiene como objetivo eliminar o resaltar características propias de la señal. Puede diseñarse como una ecuación, un programa e incluso como un circuito integrado. En comparación con los filtros analógicos, el desempeño y los resultados obtenidos con un filtro digital son muy superiores por lo que, en varias aplicaciones, se opta por digitalizar la señal solo para filtrarla digitalmente. Existen distintos tipos de filtros de acuerdo con la corrección que se desea en la señal, y uno de ellos son los filtros pasa-bajas, estos filtros atenúan todas aquellas frecuencias que estén por encima de una determinada frecuencia de corte, es decir, permite únicamente el paso de las frecuencias que estén por debajo de la frecuencia de corte. Existen diferentes modelos de funciones de transferencia para los filtros pasa-bajas, la que se elija dependerá del tipo de señal que se desea procesar, de entre los modelos de filtros más comunes se encuentran los filtros Butterworth, Chebyshev, RC y Bessel entre otros, cada uno con sus ventajas y desventajas. En la siguiente figura se muestra un ejemplo de la respuesta en frecuencia de un filtro butterworth, la principal característica de este tipo de filtro es que tiene una banda de transición muy ancha, lo que provoca que se atenúen frecuencias no deseadas.



Respuesta en frecuencia de un filtro pasa-bajas Butterworth, puede observarse una banda de transición muy ancha.

En esta práctica se trabajará en el diseño de filtros pasa-bajas, utilizando diferentes opciones que ofrece el programa de MATLAB.

**MATERIAL Y EQUIPO**

1 computadora con el software MATLAB instalado.

**PROCEDIMIENTO**

1. Diseñe un algoritmo para generar una señal senoidal (a partir de la suma de senos) que tenga las siguientes componentes de frecuencias: 5, 10, 20, 25, 35, 40, 55, 60, 65, 80, 85, 95 y 100 Hz; procure que cada componente tenga diferente amplitud, eligiendo los valores que considere adecuados.
2. Obtenga el espectro de frecuencias de la señal utilizando la función “fft”.
3. Obtenga el orden de un filtro Butterworth pasa bajas utilizando la función “buttord” que tenga una frecuencia de corte de 70Hz, un ancho de banda de transición de 30Hz y atenuación en ambas bandas no mayor a 5dBs.
4. A partir del orden del filtro diseñe un filtro pasa bajas Butterworth utilizando la función “butter”.
5. Utilizando la función “fvtool” grafique la respuesta en frecuencia del filtro diseñado.
6. Aplique el filtro digital a la señal del punto 1 utilizando la función “filter”.
7. Obtenga el espectro de frecuencias de la señal filtrada y compárelo con el espectro obtenido en el punto 2. Comente las diferencias.
8. Repita los puntos 3 al 7 para filtros pasa-bajas Chebyshev 1, Chebyshev 2 y Bessel.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS

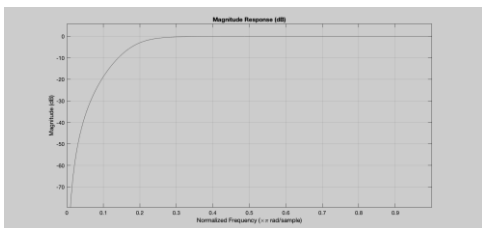
Práctica No. 10. Filtros Digitales Pasa-Altas.

COMPETENCIA

Conocer la respuesta en amplitud y fase de distintas funciones de filtros pasa bajas a través de la utilización de distintas funciones de Matlab para conocer su aplicación en el filtrado de señales biofisiológicas.

INTRODUCCIÓN

Un filtro digital, es un algoritmo que opera sobre señales digitales, y tiene como objetivo eliminar o resaltar características propias de la señal. Puede diseñarse como una ecuación, un programa e incluso como un circuito integrado. En comparación con los filtros analógicos, el desempeño y los resultados obtenidos con un filtro digital son muy superiores por lo que, en varias aplicaciones, se opta por digitalizar la señal solo para filtrarla digitalmente. Existen distintos tipos de filtros de acuerdo con la corrección que se desea en la señal, y uno de ellos son los filtros pasa-altas, estos filtros atenúan todas aquellas frecuencias que estén por encima de una determinada frecuencia de corte, es decir, permite únicamente el paso de las frecuencias que estén por encima de la frecuencia de corte. Existen diferentes modelos de funciones de transferencia para los filtros pasa-altas, la que se elija dependerá del tipo de señal que se desea procesar, de entre los modelos de filtros más comunes se encuentran los filtros Butterworth, Chebyshev, RC y Bessel entre otros, cada uno con sus ventajas y desventajas. En la siguiente figura se muestra un ejemplo de la respuesta en frecuencia de un filtro butterworth pasa altas.



Respuesta en frecuencia de un filtro pasa-altas Butterworth.

En esta práctica se trabajará en el diseño de filtros pasa-altas, utilizando diferentes opciones que ofrece el programa de cómputo MATLAB.

MATERIAL Y EQUIPO

1 computadora con el software MATLAB instalado.

PROCEDIMIENTO

- Diseñe un algoritmo para generar una señal senoidal (a partir de la suma de senos) que tenga las siguientes componentes de frecuencias: 5, 10, 20, 25, 35, 40, 55, 60, 65, 80, 85, 95 y 100 Hz; procure que cada componente tenga diferente amplitud, eligiendo los valores que considere adecuados.
- Obtenga el espectro de frecuencias de la señal utilizando la función “fft”.
- Obtenga el orden de un filtro Butterworth pasa altas utilizando la función “buttord” que tenga una frecuencia de corte de 20Hz, un ancho de banda de transición de 40Hz y atenuación en ambas bandas no mayor a 5dBs.
- A partir del orden del filtro diseñe un filtro pasa altas Butterworth utilizando la función “butter”.
- Utilizando la función “fvtool” grafique la respuesta en frecuencia del filtro diseñado.
- Aplice el filtro digital a la señal del punto 1 utilizando la función “filter”.
- Obtenga el espectro de frecuencias de la señal filtrada y compárelo con el espectro obtenido en el punto 2. Comente las diferencias.
- Repita los puntos 3 al 7 para filtros pasa-altas Chebyshev 1, Chebyshev 2 y Bessel.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA  
ECITEC, VALLE DE LAS PALMAS**

**Práctica No. 11. Filtrado de una señal ECG.**

**COMPETENCIA**

Identificar las características de una señal ECG a través del diseño e implementación de filtros digitales para conocer su importancia en el diagnóstico médico con actitud honesta y responsable.

**INTRODUCCIÓN**

Las señales biofisiológicas requieren de un tratamiento digital especial debido a que son de baja frecuencia y de amplitudes del orden de los mV. En particular una señal electrocardiográfica (ECG) la cual se encarga de registrar la actividad eléctrica del corazón, permitiendo a través de su estudio la detección de enfermedades del corazón, podría presentar frecuencias útiles desde 1Hz hasta los 250Hz. El filtrado de señales biofisiológicas consiste de 3 etapas básicas, primero debe diseñarse un filtro rechaza-banda para eliminar la componente de CA, la cual está siempre presente en la adquisición de este tipo de señales, a menos que se elimine analógicamente directamente con el circuito de detección. La segunda etapa consiste en diseñar un filtro pasa-bajas cuya frecuencia de corte sea la máxima esperada y la tercera etapa consiste en diseñar un filtro pasa-altas de frecuencia de corte muy baja,  $\leq 1\text{Hz}$  para eliminar cualquier ruido que sea producido por el movimiento torácico. El diseño de los filtros y la determinación de las frecuencias de corte de los mismos, dependerá por supuesto de la posición en la que se hayan colocado los electrodos y de la derivación que se esté adquiriendo. En esta práctica de trabajará con señales ECG obtenidas de la derivación DII de diferentes individuos.

**MATERIAL Y EQUIPO**

1 computadora con el software MATLAB instalado.

**PROCEDIMIENTO.**

1. Elija tres señales ECG de las proporcionadas por el profesor. Estas señales fueron muestreadas a una frecuencia de 500Hz durante 12 seg.
2. Elimine la componente de corriente directa.
3. Genere el vector tiempo correspondiente y grafique la señal en función del tiempo.
4. Utilice la función “fft” de Matlab para obtener el espectro en frecuencias de cada una de las señales.
5. Grafique cada uno de los espectros de frecuencia de las señales e identifique la componente de 60Hz correspondiente a AC.
6. Diseñe un filtro notch para eliminar la componente de 60Hz en cada una de las señales.
7. Obtenga la FFT de la señal ya filtrada y gráfiquela para verificar que se haya eliminado la componente de AC.
8. Diseñe un filtro pasabajas de cada tipo (butterworth, cheby1 y cheby2) para cada señal con frecuencia de corte de 100Hz y filtre las señales. Recuerde graficar la respuesta de los filtros diseñados utilizando la función fvtool.
9. Diseñe un filtro pasaaltas de cada tipo (butterworth, cheby1 y cheby2) para cada señal con frecuencia de corte de 0.7Hz y filtre cada una de las señales. Recuerde graficar la respuesta de los filtros diseñados utilizando la función fvtool.
10. Grafique las señales filtradas y sin filtrar, así como sus respectivos espectros de frecuencias para identificar los cambios en ellas.
11. ¿Logró eliminar totalmente las componentes de frecuencia no deseadas? ¿Con cuáles de filtros obtuvo mejores resultados? Comente sus resultados.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**ESCUELA DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA**  
**ECITEC, VALLE DE LAS PALMAS**

**Práctica No. 12. Filtrado por “ventaneo”**

**COMPETENCIA**

Conocer la respuesta en amplitud y fase de distintas funciones de ventana fijas a través de la utilización de distintas funciones de MATLAB para conocer su aplicación en el filtrado de señales biofisiológicas con actitud honesta y responsable.

**INTRODUCCIÓN**

Como se ha observado a través de la transformada rápida de Fourier se pueden obtener las componentes de frecuencia de una señal determinada, si la señal tiene una sola componente de frecuencia, ésta se observa como una línea espectral muy bien definida, sin embargo en la vida real generalmente las señales de entrada se componen de una mezcla de frecuencias y cuando se obtiene su espectro en frecuencias estas líneas no serán tan definidas si no que presentarán cierta dispersión lo que dificultará la interpretación del espectro. Este efecto de dispersión se puede reducir si se multiplica la señal por una función de ventana en el dominio del tiempo o bien se “convolucionan” ambas señales (señal de entrada y la función de ventana) en el dominio de la frecuencia. Existen varios tipos de ventanas según su forma que en general representa una forma de “campana” con diferentes grosores las más comunes son la ventana rectangular, ventana de Hamming, Ventana de Hanning, Ventana Blackman-Harris entre muchas otras que se eligen de acuerdo con el efecto que se desea obtener sobre la señal bajo estudio. En el desarrollo de esta práctica se utilizarán diferentes funciones de ventanas ya predeterminadas en el software MATLAB para conocer la forma de onda, la respuesta en frecuencia y el efecto de cada una de las ventanas sobre una señal en específico.

**MATERIAL Y EQUIPO**

1. Computadora con software MATLAB instalado.

**PROCEDIMIENTO**

1. Escriba un programa en Matlab represente la forma temporal de una ventana rectangular de 9 puntos, utilizando la función “ones”.
2. Obtenga la transformada de Fourier de la ventana generada anteriormente, para esto debe normalizar el eje “x” a 512 puntos.
3. Grafique la forma temporal y la forma espectral de la ventana.
4. Repita los pasos 1 al 3 para una ventana triangular utilizando la función “triang”, para una ventana “Hamming” utilizando la función “hanning”, para una ventana Hanning utilizando la función “hanning” y para una ventana Blackman utilizando la función “blackman”.
5. Cargue al “workspace” de matlab una de las señales ECG proporcionadas por el docente.
6. Aplique cada una de las ventanas generadas a la señal ECG y grafique las señales resultantes.
7. Concluya sobre los resultados obtenidos.