

PROCESAMIENTO DIGITAL DE SEÑALES

PROFESOR: HUMBERTO LOAIZA C., Ph.D.

LABORATORIO No. 5 (Software)

Transformada de Fourier

Componentes Espectrales - Densidad Espectral de Potencia

Aplicación a señales de Audio – Archivos WAV

I. Objetivos

- *Estudiar la transformada de Fourier inversa y directa en señales de tiempo discreto.*
- *Generar y analizar los espectros de tensión y fase de señales simples y compuestas.*
- *Manipular archivos de audio con formato WAV.*
- *Analizar el comportamiento temporal y frecuencial de señales de audio.*
- *Producir y analizar el espectrograma y el Cepstrum de señales discretas.*

II. Introducción

Depuración de programas

Los programas contenidos en archivos .M pueden ser depurados en Matlab ya sea utilizando las funciones del Editor/Depurador o las funciones de depuración. Ambos métodos pueden utilizarse en forma alterna.

El proceso de depuración involucra las siguientes etapas:

- i. Preparación para la depuración.
- ii. Colocación de los puntos de ruptura (breakpoints)
- iii. Correr el archivo .M con los breakpoints.
- iv. Ejecución paso a paso del archivo .M
- v. Inspección de valores
- vi. Corrección de problemas y finalización

III. Procedimiento

1. Transformada de Fourier

- 1.1. El siguiente programa construye una señal discreta y calcula su transformada de Fourier. La señal es construida teniendo en cuenta la frecuencia y el tiempo total de muestreo, al igual que la frecuencia propia de la señal.

```

clear all ;
%Definición del muestreo
fm=10000;    %frecuencia de muestreo (muestras/segundo)
Tm=1/fm ;    %periodo de muestreo (segundos)

tt=0.5 ;     %tiempo total en segundos (segundos)

t=0:Tm:tt;   %instantes de tiempo en donde evaluar la señal

% Generación de la señal
WW=628;      %Frecuencia angular en el dominio temporal
A=1;         %Amplitud
s=A*sin(WW*t); %señal seno

% Dibujar un periodo de la señal
mpp=floor(fm*2*pi /WW); %calcular muestras en un periodo

h1=figure;
stem(t(1:mpp),s(1:mpp));
title('Senal en el Tiempo');
xlabel('Tiempo (s)');
ylabel('Magnitud');

% Calcular transformada directa de Fourier

fs=fft(s);
[nf,nc]=size(fs);

mag=20*log10(abs(fs)); %magnitud en decibeles

factor=180.0/pi;
fase=factor*atan2(imag(fs),real(fs));

delta=2*pi/nc;
ww=0:delta:2*pi-delta; % Frecuencia en el dominio frecuencial

h2=figure;
stem(ww,mag);
title('Data spectrum');
xlabel('frequency');
ylabel('magnitude [dB]');

h3=figure;
stem(ww,fase);
title('Data spectrum');
xlabel('frequency');
ylabel('Phase [°]');

keyboard; %Detiene ejecución del programa. Para continuar escriba return %en
          la línea de comandos de Matlab.
close all;

```

- 1.2. Ejecute el script del numeral anterior y establezca las modificaciones que pueden realizarse al programa de §1.1 para obtener una mejor visualización de los espectros de fase y de tensión. Implemente, verifique y justifique su sugerencia.

1.3. Corra el programa modificado en §1.2. para diferentes valores de la frecuencia angular, ω , y de la amplitud A de la señal *temporal*. Qué efectos produce las variaciones de ω y de A en el espectro? Justifique sus respuestas.

- | | |
|--------------------|--------------|
| a- $\omega = 160$ | e- $A = 0.5$ |
| b- $\omega = 450$ | f- $A = 10$ |
| c- $\omega = 1500$ | g- $A = 100$ |
| d- $\omega = 2400$ | h- $A = -10$ |

1.4. De los resultados obtenidos en el numeral anterior, es posible encontrar alguna relación entre la frecuencia (dominio del tiempo) de la señal y la frecuencia donde se producen los picos en el espectro? La frecuencia de muestreo f_m es importante en esta relación? Pruebe para los casos a, b, c y d. La relación se mantiene constante para cada caso? Justifique sus respuestas.

1.5. Realice modificaciones al programa de §1.1. para obtener una función en Matlab que permita obtener el espectro de tensión y fase de una señal, al igual que algunos periodos (dos o tres) de la señal de entrada y la reconstruida (transformada inversa). Las gráficas deben presentarse por parejas en dos ventanas. La función debe tener el siguiente encabezado:

function [esp, rsenal] = espectro (fsenal, tt, fm)

donde, **fsenal** es la función simbólica de la señal, es decir, una cadena de caracteres; **tt** es el tiempo total de muestreo [s]; **fm** es la frecuencia de muestreo [muestras/s]; **esp** es la transformada de Fourier de la señal, y **rsenal**, es la secuencia que representa la señal reconstruida.

La función *espectro* debe incluir instrucciones pertinentes para interpretar correctamente la cadena *fsenal*. Incluya el código y explique el algoritmo utilizado.

1.6. Pruebe la función **espectro** para las siguientes señales:

- $10 - 5 \sin(251 t) \times 25.4 \cos(623 t)$
- $5 \sin(65 t) + 25.4 \cos(621 t) - 12.5 \sin(65 t + \pi/4)$

1.7. Compare las señales antes y después de reconstruidas. Qué se observa?

1.8. Repita §1.6 y §1.7. pero modifique la frecuencia de muestreo respecto a la señal original como se indica:

- $f_m = 150$
- $f_m = 100$
- $f_m = 50$

1.9. Qué relación encuentra entre las magnitudes máximas de las señales en el tiempo y en sus respectivos espectros?. Justifique su respuesta.

2. Manipulación de señales de audio

- 2.1. El siguiente programa permite crear una señal de audio y dibujar su diagrama de frecuencia. Además permite grabar la señal en un archivo .WAV, el cual podrá ser escuchado posteriormente con la función *sound* de Matlab (y con cualquier otro programa reproductor, como el mplayer.exe de Windows).

```
function [s_norm]=analyze1(t_total, f_muestreo )

%Crear t_total segundos de una señal de sonido muestreada a f_muestreo

t=0:1/f_muestreo: t_total;

% Generar señal de sonido
f1=1000 ;
f2=2000 ;
f3=3000 ;
s= 2* sin(2*pi*f1*t)+ 3 *sin(2*pi*f2*t)/2+ 5* sin(2*pi*f3*t)/2;
s_norm=s/max(abs(s)); % normalizar los valores entre -1 y 1.

% crear archivo .WAV y almacenarlo en disco
wavwrite(s_norm, f_muestreo,'son1.wav');

%Diagrama de frecuencia

mag_ss=abs(fft(s_norm)); %magnitud del espectro de s
nm=length(mag_ss); %numero de muestras
delta=f_muestreo/nm;
eje_frec= 0:delta:f_muestreo-delta;

plot(eje_frec,mag_ss);
title('Componentes Frecuenciales en el Dominio Temporal');
xlabel('Frecuencia en Herz');
ylabel('Magnitud de los componentes frecuenciales');
```

- 2.2. Estudie el código del programa anterior y explique la función de las principales instrucciones.
- 2.3. Estudie a fondo la porción de código para obtener el diagrama de frecuencia. Explique cómo se logra graficar el eje horizontal con una escala de frecuencias en el dominio temporal, y por qué se utiliza la transformada inversa de Fourier? Encuentre la relación de las magnitudes de los picos con la amplitud de la señal en el tiempo?
- 2.4. Corra la función para $t_total=1.5$ segundos y $f_muestreo=10000$ Hz. Para ello, escriba desde la línea de comandos:

```
>> [son1] = analyze1(1.5, 10000);
```

Analice e interprete los resultados obtenidos.

- 2.5. Escuchar una señal de audio puede realizarse al utilizar las funciones *sound* y *wavplay* de Matlab. Desde la línea de comandos de Matlab, ejecute la siguiente instrucción:

```
>> sound(son1, 10000);
```

y posteriormente,

```
>> wavplay(son1, 10000);
```

Consulte el manual de ayuda y establezca cuales son las principales características de las funciones *sound* y *wavplay* citando ventajas y desventajas.

- 2.6. Corra el programa de §2.1. para cuatro diferentes conjuntos de frecuencias {f1,f2, f3} y repita §2.5. Analice los efectos que se producen al utilizar frecuencias {f1,f2, f3} bajas y altas. Si es del caso, amplíe la duración del sonido para escucharlo mejor.

Pueden diferenciarse auditivamente los diferentes tonos de la señal (un tono lo produce una sola señal senoidal)? Explique.

- 2.7. Para los diferentes casos implementados en §2.6. analice los diagramas de frecuencia obtenidos y determine su significado práctico (para qué es útil la información de las abscisas y ordenadas de la gráfica?)

3.0. Sonido 1

- 3.1. Con base en el programa §2.1. realice una función en Matlab que permita visualizar adecuadamente la señal de audio en el dominio del tiempo (visualizar por partes si la señal tiene muchas muestras), el diagrama de frecuencia (como la obtenida con la función *analyze1*) y su respectivo espectro de tensión y fase (función *fft*). También debe permitirse almacenar la señal de sonido en un archivo .wav con nombre seleccionado por el usuario. Denomine a esta función **sonido1**. Consigne el código y explique el algoritmo del programa.

- 3.2. Corra el programa *sonido1* con la misma señal y con los mismos valores de frecuencia usados en §2.7. Analice los resultados y establezca una relación entre los picos presentes en el espectro de tensión y el diagrama de frecuencias. Consigne algunos resultados numéricos.

Además, encuentre una expresión que relacione las frecuencias en ambos dominios. La frecuencia de muestreo interviene en esta expresión?

- 3.3. El siguiente programa permite cargar una señal de audio a partir de un archivo .wav, visualizar el diagrama de frecuencia y escuchar el sonido grabado.

```
function [s]=analyze2(archivo_son)

%archivo_son = cadena de caracteres que contiene el nombre del archivo .wav

%Obtener datos del archivo de sonido
[s,fs,bits] = wavread(archivo_son)

%Diagrama de frecuencia
```

```

mag_ss=abs(fft(s));           %magnitud del espectro de s
nm=length(mag_ss) ;          %numero de muestras
delta=fs/nm;
eje_frec= 0:delta:fs-delta;

h1=figure;
plot(eje_frec,mag_ss);
title('Componentes Frecuenciales en el Dominio Temporal');
xlabel('Frecuencia en Herz');
ylabel('Magnitud de los componentes frecuenciales');

%Reproducir sonido
wavplay(s,fs);

```

3.4. Estudie el código del programa anterior y explique la función que realizan las principales instrucciones.

3.5. Ejecute el programa anterior con la siguiente instrucción desde la línea de comandos,

```
>> analyze2 ( 'son1.wav' );
```

Compare los resultados obtenidos con los arrojados en el numeral §2.4 y §2.5.

3.6. Copie en su carpeta de trabajo varios archivos con extensión .wav y utilice la función analyze2 sobre dos archivos .wav. Analice los resultados observados.

4.0. Sonido2

4.1. Realice un programa en Matlab (denomínelo **sonido2**) que permita cargar y crear archivos .wav, para realizar manipulaciones sobre ellos. La operación a realizar así como los parámetros deben activarse con botones y menús. El programa debe permitir :

- a. Visualizar las señales de audio antes y después de la manipulación en el dominio del tiempo.
- b. Generar mensajes de error por ventanas.
- c. Los nombres de los archivos de audio deben indicarse y seleccionarse a través de ventanas de diálogo, así como los parámetros requeridos.
- d. Las manipulaciones incluyen **modificación de amplitud** (modificación de volumen sin saturación), **mezcla** (los datos se colocan intercalados), **suma** (suma punto a punto), **inversión** (escuchar de atrás hacia adelante), **cambio de mono a estéreo y viceversa** (escuchar sonidos diferentes por cada parlante de señales de audio y voz), **fan in** (atenuación progresiva) y **fan out** (amplificación progresiva).
- e. **El programa debe incluir una etapa de verificación e igualación de las frecuencias de muestreo, del número de bits por muestra, del número de canales y de la longitud de los archivos de audio antes de procesarlos.**
- f. La función debe permitir visualizar el **espectro de tensión** (con eje de frecuencia en hertz) y el **espectrograma** de las señales.

- g. Reproducir audiblemente los archivos audio de entrada y salida.
- h. Capturar voz mediante el micrófono y generar archivo .wav de almacenamiento.
- i. Grabar los archivos procesados en formato .wav mediante ventanas de dialogo.

Consigne el código del programa y explique el algoritmo utilizado.

- 4.2. Con ayuda de un micrófono genere archivos que contengan voces de hombre y de mujer. Utilice el programa **sonido2** para encontrar sus espectros y diagramas frecuenciales. Pruebe el funcionamiento de su programa con archivos de voz que contengan:
 - a. Las 5 vocales pronunciadas por un hombre (espaciadas en el tiempo)
 - b. Las 5 vocales pronunciadas por una mujer (espaciadas en el tiempo)
 - c. Una palabra pronunciada por un hombre
 - d. Una palabra pronunciada por una mujer
 - e. Las frases: "anita lava la tina" y "dábale arroz a la zorra el abad"
- 4.3. Con los resultados obtenidos es posible diferenciar directamente cuando se trata de una voz masculina o de una femenina? Justifique su respuesta.
- 4.4. Con base en los programas desarrollados en esta sección, implemente un programa que permita calcular el CEPSTRUM de una señal de audio (No se debe utilizarse las funciones `cceps` ni `rceps` de Matlab). El programa debe visualizar las señales en el dominio del tiempo, al igual que sus diagramas frecuenciales y el cepstrum. Denomine a la función ***cepstrum***(). Investigue como se calcula y qué utilidades presenta en procesamiento de señales el cepstrum de una señal de audio y voz. Sirve para identificar al hablante o para identificar el habla (es decir, lo que se dice) ?
- 4.5. Pruebe la función `cepstrum`() con los archivos de audio utilizados en §4.2. Con los resultados obtenidos es posible diferenciar directamente cuando se trata de una voz masculina o de una femenina? Justifique su respuesta.
- 4.6. Según los resultados y conclusiones de los numerales §4.3 y §4.5 cuál de los métodos es más apropiado para diferenciar personas según las características del habla? Justifique su respuesta.
- 4.7. Valide ante el grupo y el profesor el buen funcionamiento de los programas **sonido2**() y **cepstrum**().

5.0. Informe

- 5.1. Presente un informe escrito claro en donde se consigne los programas, las figuras, las respuestas, las justificaciones y los resultados obtenidos. También incluya el análisis e interpretación de los resultados. Igualmente consigne conclusiones, observaciones, y la literatura consultada. Utilice en el informe la *misma numeración* de la guía de laboratorio.
- 5.2. Incluya información de ayuda para cada una de las funciones implementadas. Estas ayudas deben poder consultarse con el comando **help** de Matlab.

Nota: *La omisión de alguno de los items en el informe representa una disminución de la nota.*

Para facilitar la sustentación ante el profesor, realice scripts donde se definan los datos y se invoquen las funciones desarrolladas.