**Universidad de San Carlos de Guatemala**



**Facultad de Ingeniería**

**Escuela de Mecánica Eléctrica**

**Laboratorio de Electrónica**

**Comunicaciones 1**

**Vacaciones Junio 2014**

**Auxiliar: Edvin Baeza**

**Práctica 3**

**Muestreo Con Matlab**

**Introducción**

Ventajas de la Transmisión Digital

• Inmunidad al ruido.

• Se prestan mejor procesamiento y multiplexado (DSP Procesamiento de Señal

Digital).

• Se pueden transportar a distancias mayores que las analógicas.

• Es mas fácil evaluar las señales digitales y portando es mas fácil comparar la eficiencia de los sistemas digitales.

• Los sistemas digitales se adaptan mas para evaluar el funcionamiento con errores.

Por lo que se pueden detectar y corregir errores. Desventajas de la Transmisión Digital

• La transmisión de señales analógicas codificadas digitalmente requiere un ancho de banda mayor que la señal analógica original. (el ancho de banda es costoso)

• Las señales analógicas se deben convertir en códigos digitales antes de su transmisión y reconvertirse a la forma de onda analógica en el receptor. Por tanto necesita circuitos de Codificación y Decodificación.

• La transmisión digital requiere una sincronización precisa respecto al tiempo entre los relojes del transmisor y del receptor. Por tanto requieren costosos circuitos de recuperación de reloj en todos los receptores. Esto no aplica para todas las modulaciones digitales. Hay algunas que no requieren relojes sincronizados.

• Los sistemas de transmisión digital son incompatibles con los sistemas de transmisión analógica.

• Comentar todas las instrucciones del programa, que hace y para que sirve la instrucción.

function varargout = progm(varargin)

% PROGM M-file for progm.fig

% PROGM, by itself, creates a new PROGM or raises the existing

% singleton\*.

%

% H = PROGM returns the handle to a new PROGM or the handle to

% the existing singleton\*.

%

% PROGM('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in PROGM.M with the given input arguments.

%

% PROGM('Property','Value',...) creates a new PROGM or raises the

% existing singleton\*. Starting from the left, property value pairs

are

% applied to the GUI before progm\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property

application

% stop. All inputs are passed to progm\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help progm

% Last Modified by GUIDE v2.5 21-May-2012 23:28:04

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', [@progm\_OpeningFcn,](mailto:@progm_OpeningFcn) ...

'gui\_OutputFcn', [@progm\_OutputFcn,](mailto:@progm_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

% End initialization code - DO NOT EDIT

% --- Executes just before progm is made visible.

function progm\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to progm (see VARARGIN)

% Choose default command line output for progm handles.output = hObject;

% Update handles structure guidata(hObject, handles);

% UIWAIT makes progm wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line. function varargout = progm\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)%al hacer click

en el boton grabar audio ejecuta lo siguiente

% hObject handle to pushbutton9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grabar Audio------------------------------

%----------------------------------------------------------------

fs = 8000; %frecuencia de muestreo grabacion

ain = wavrecord(6\*fs, fs); %grabando audio sound(ain, fs); %reproduciendo audio wavwrite(ain, fs, 'insound.wav'); %escribiendo audio

global AIN Fx x lx nx tx %variables globales

[AIN, Fx] = wavread('insound.wav'); %leyendo audio

%---------Variables grafica

x = AIN'; %amplitudes del audio in

lx = length(x); %tamaño archivo de audio in

nx = (0:lx-1); %vector tamaño = audio in

tx = nx./Fx; %tiempo para graficar

%---------Variables grafica

global esp fesp %variables globales

esp = abs(fft(x/(lx/2))); %valor absoluto de la transformada rapida de

fourier

fesp=Fx\*linspace(0,1,lx); %crea un vector espaciado lx

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)%boton Grafica

1

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grafica 1---------------------------------

%----------------------------------------------------------------

[y, Fs] = wavread('insound.wav'); %leyendo audio

x=y(:,1)'; % x = a inverso en pasos de 0.1 para que el audio en y se lea

en x

clear y ; %borrar y

step=fix(5\*Fs/25); %redondear matris compleja window=fix(20\*Fs/25);%redondear datos

fftn=2^nextpow2(window); %2 elevado a la transformada rapida de fourier [S,f,t]=specgram(x, fftn, Fs, window, window-step); %devuelve el tiempo transformada de Fourier de la señal de entrada

figure(4);

S=abs(S(2:fftn\*450/Fs,:));%vaolr absoluto de la transformada de fourier

con un factor de escalamiento a ciertos pasos

%S=abs/max(S(:));

S=max(S, 10^(-60/10));%maximo de la ventana

S=min(S, 10^(0/10));%minimo de la ventana

imagesc(t, f, log10(S));%selecciona una escala correcta para la imagen

set(gca(),'ydir','normal');%posiciona en los ejec correctos la grafica title('n=8, fs=8000');%titulo de la grafica

xlabel('tiempo');%etiqueta del eje x

ylabel('frecuencia');%etiqueta del eje y grid on;%mostrar grilla en la grafica zoom;%habilita el zoom

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grafica 2---------------------------------

%----------------------------------------------------------------

global x tx %variables declaradas para que funcionen dentro de la llamada del boton

%x = AIN'; %amplitudes del audio in

%lx = length(x); %tamaño archivo de audio in

figure(1); %nombre de la grafica aparece como titulo de la ventana plot(tx,x,'r'); %grafica las variables declaradas anteriormente title('señal continua IN','fontsize',14);%tiulos y etiquetas de la grafica

xlabel('tx (seg) \rightarrow');

ylabel('x(tx)');

grid on;

zoom;

% --- Executes on button press in pushbutton4.

function pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grafica 3---------------------------------

%----------------------------------------------------------------

%esp = abs(fft(x/(lx/2))); %valor absoluto de la transformada rapida de

fourier

%fesp=Fx\*linspace(0,1,lx); %crea un vector espaciado lx global esp fesp

figure(2);

plot(fesp,esp);%grafica la amplitud discreta de cada armonica de la

transformada de fourier

title('señal en frecuencia IN','fontsize',14);%lo mismo que en las anteriores

xlabel('Frequency (Hz) \rightarrow');

ylabel('Amplitud');

grid on;

zoom;

% --- Executes on button press in pushbutton5.

function pushbutton5\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grafica 4--------------------------------

%----------------------------------------------------------------

%nx = (0:lx-1); %vector tamaño = audio in

%x = AIN'; %amplitudes del audio in

global x nx figure(3);

stem(nx,x,'.'); %grafica una secuencia discreta

title('señal discreta IN','fontsize',14); xlabel('nx (muestra) \rightarrow'); ylabel('x(nx)');

grid on;

zoom;

% --- Executes on button press in pushbutton6.

function pushbutton6\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton8 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%----------------------------------------------------------------

%----------------------Grafica 5---------------------------------

%---------------------------------------------------------------- global x

fvtool(x);%muestra las curas de bode del sistema

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a

double

% --- Executes during object creation, after setting all properties. function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns

called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),

get(0,'defaultUicontrolBackgroundColor'))

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

end

function edit2\_Callback(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text

% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties. function edit2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

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

end

function edit3\_Callback(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text

% str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties. function edit3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

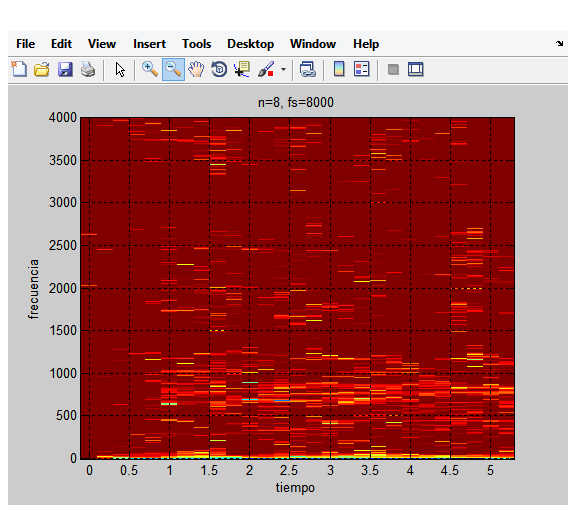
% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

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

end

Grafica 1:

◦

◦ Para qué sirve la gráfica?

Para identificar sonidos fonéticos y procesado del habla.

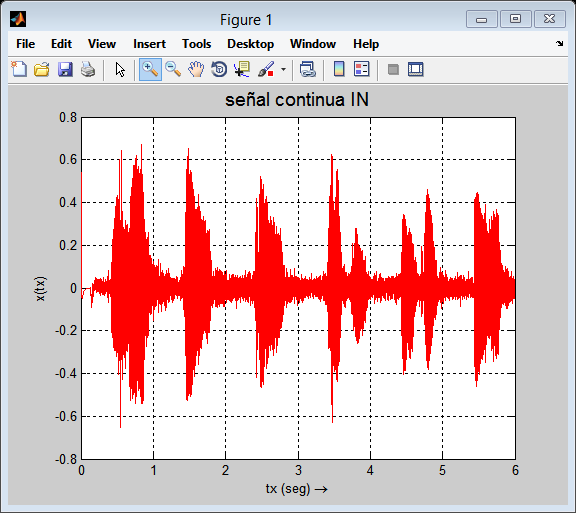
Como se interpreta?

Las líneas brillantes son los fundamentales de cada tono y las otras líneas brillantes cercanas son los sobretonos armónicos; en conjunto, forman el [espectro de frecuencias.](http://es.wikipedia.org/wiki/Espectro_de_frecuencias)

◦ Que información le proporciona?.

Energía del contenido frecuencial de la señal según va variando ésta a lo largo del tiempo.

Grafica 2



◦ Para que sirve la gráfica?

Para observar las variaciones de amplitud con respecto al tiempo.

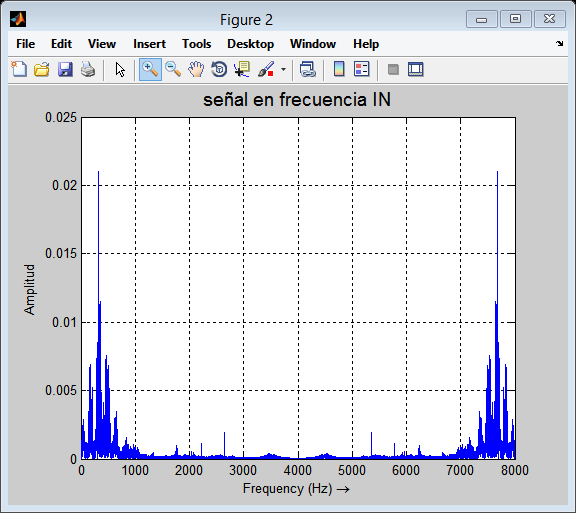
◦ Como se interpreta?

Basados en el movimiento armónico simple las lecturas que nos muestra son amplitudes de la señal variando en el tiempo.

◦ Que información le proporciona?

Con cuanta intensidad varia la señal en cierto tiempo, podría decirse que en el intervalo en donde las saturación es máxima (en donde hay mas líneas azules y mas altas), se ha recibido una señal coherente de sonido interpretable tomando el resto como ruido.

Grafica 3



◦ Para que sirve la gráfica?

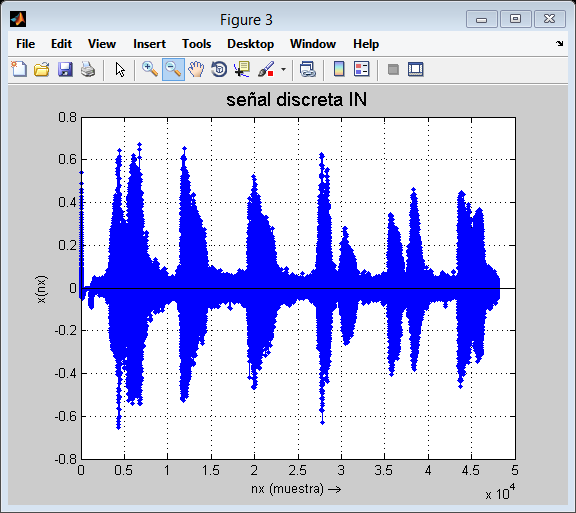
Para observar cómo cambia la amplitud en función de la frecuencia.

◦ Como se interpreta?

En frecuencias medias la amplitud de la señal de sonido es mínima

◦ Que información le proporciona?. El ancho de banda de señal

Grafica 4



◦ Para que sirve la gráfica?

Para observar la amplitud de las armónicas.

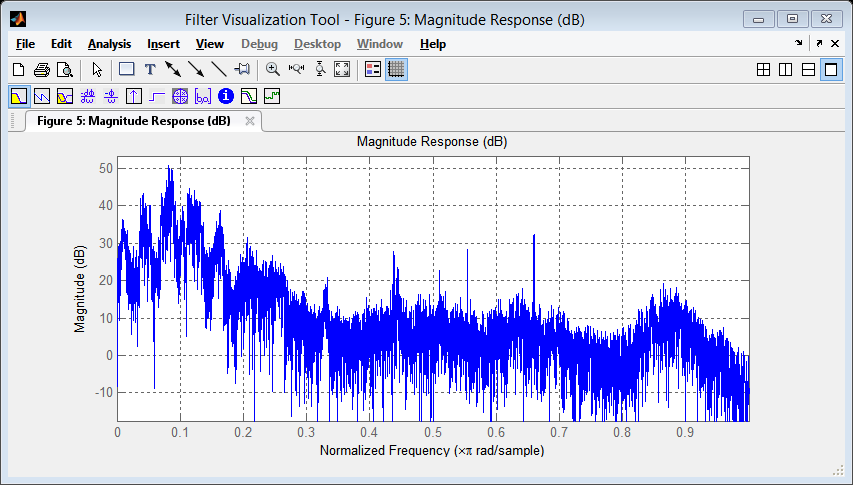
◦ Como se interpreta?

Al obtener la serie de Fourier de cada una de las muestras de audio recibidas se obtiene una amplitud para su frecuencia fundamental

◦ Que información le proporciona?.

La amplitud de las muestras a n-esima frecuencia.

Grafica 5



◦ Para que sirve la gráfica?

Para observar la relación entrada salida.

◦ Como se interpreta?

Se interpreta viendo cual es la caída en decibeles en respuesta a una frecuencia normalizada.

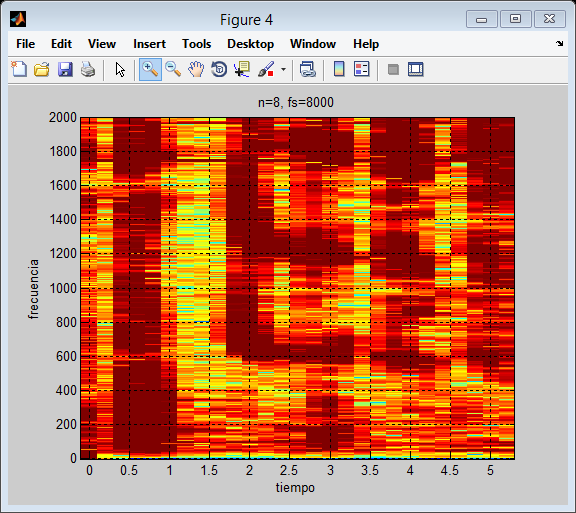
◦ Que información le proporciona?

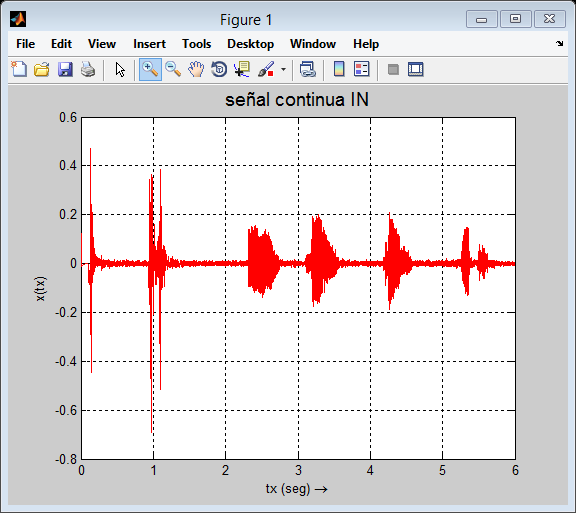
El ancho de banda del sistema en cuestión.

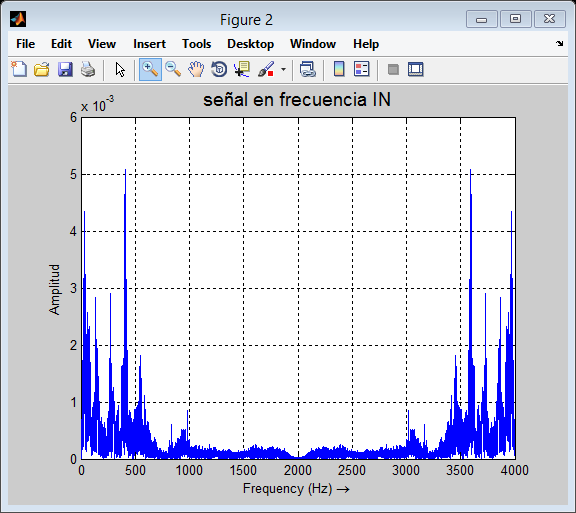
Analizar la misma señal de audio que ustedes elijan (mínimo 15 segundos), y muestrear la señal a:

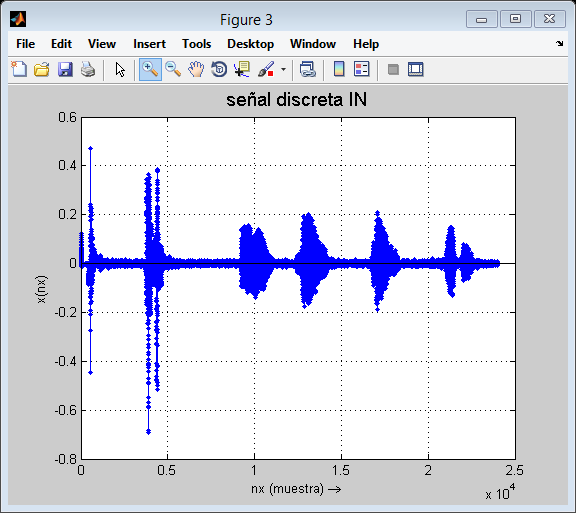
◦ Fs<<2Fmax

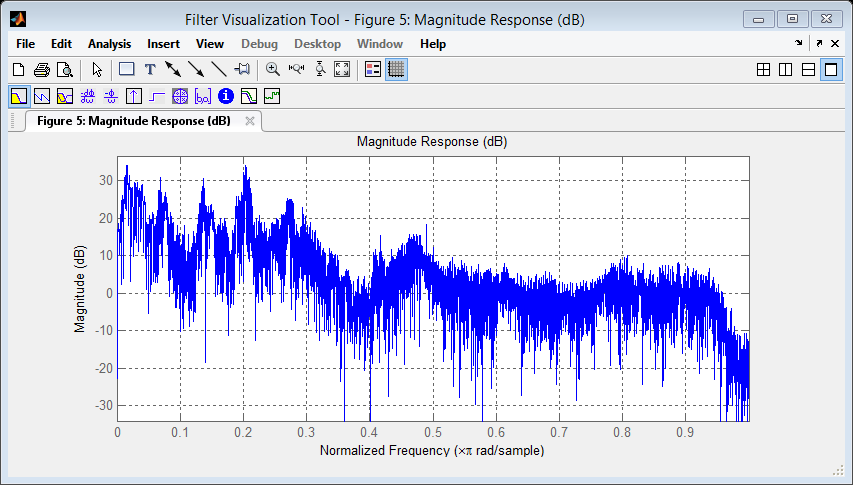
Fs=4000:





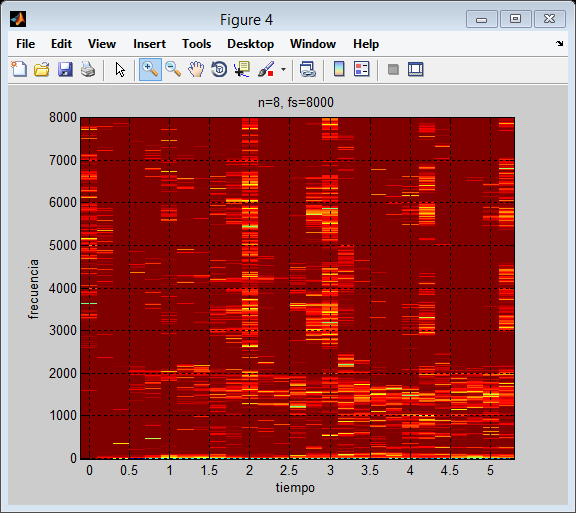


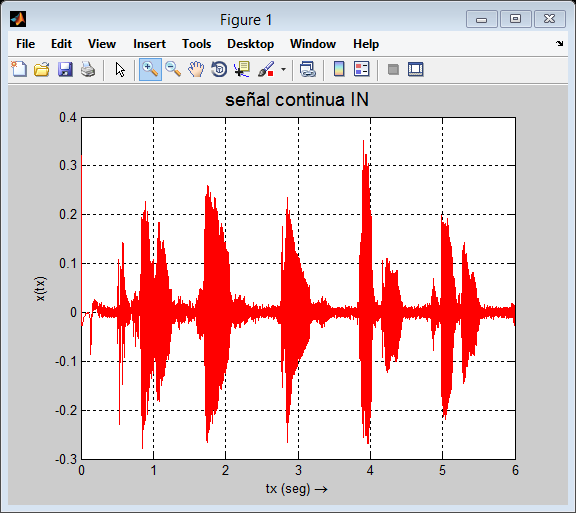


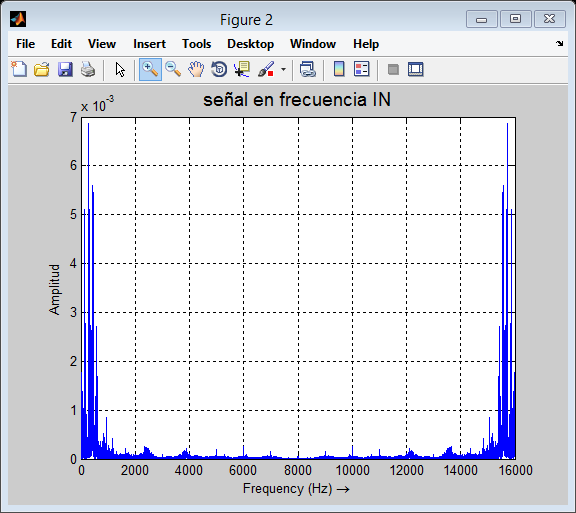


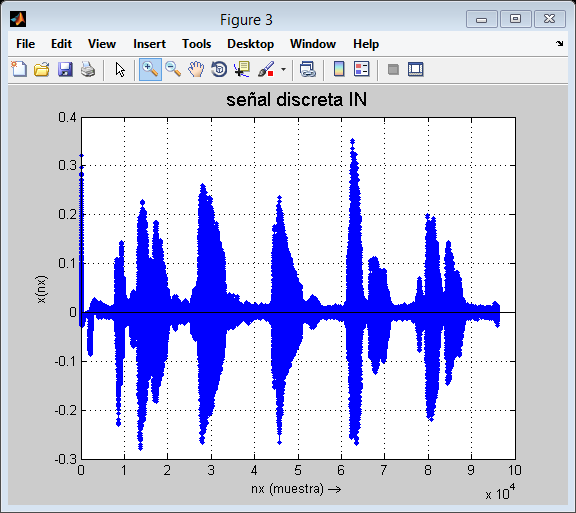
◦ Fs=2Fmax

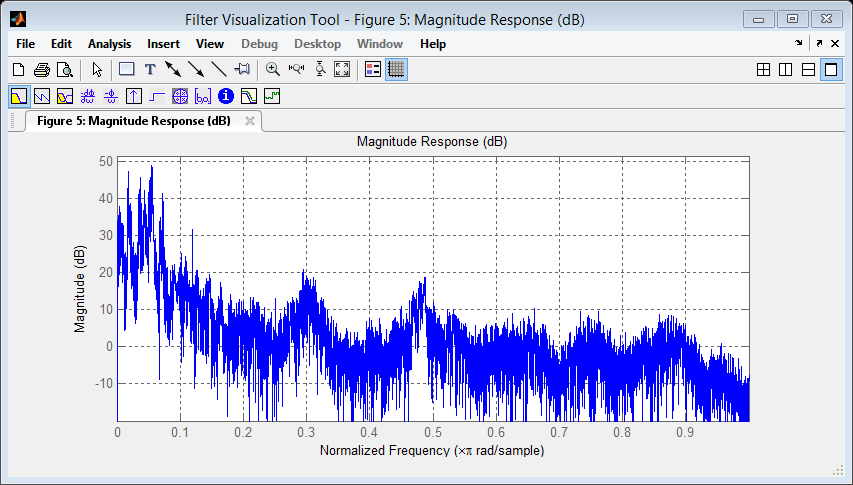
FS = 16000:





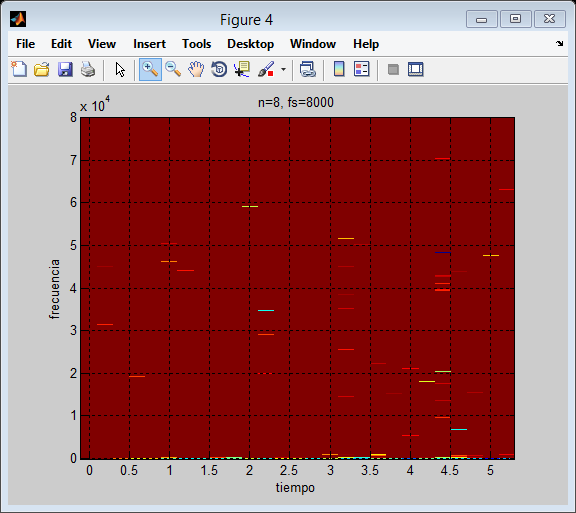


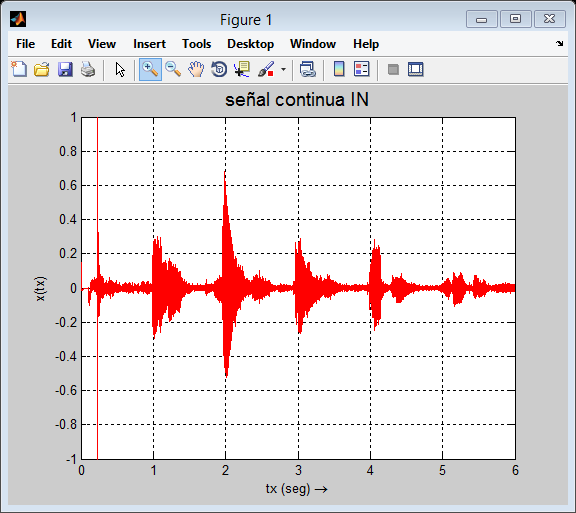


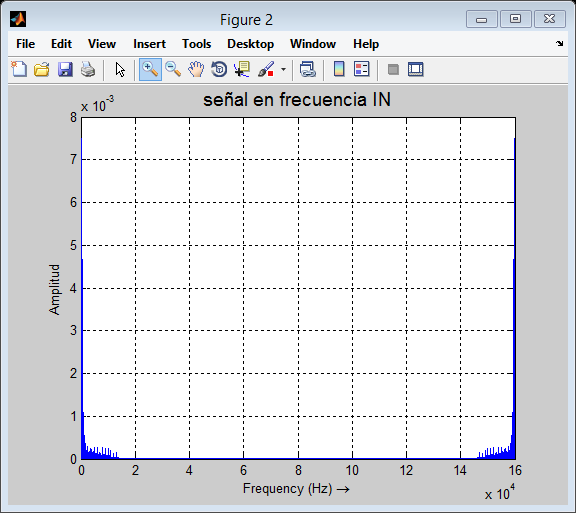


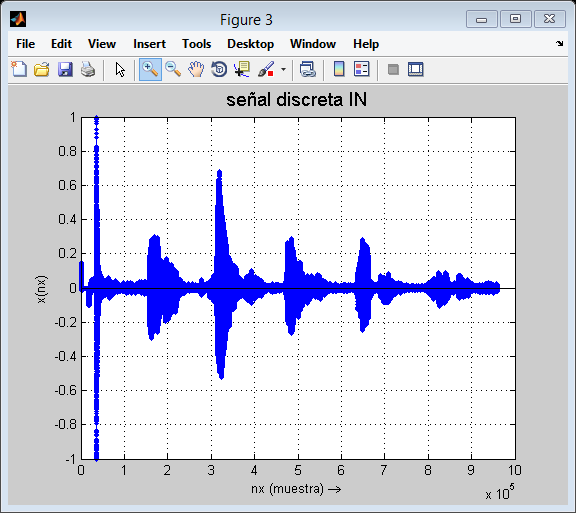
◦ Fs>>2Fmax

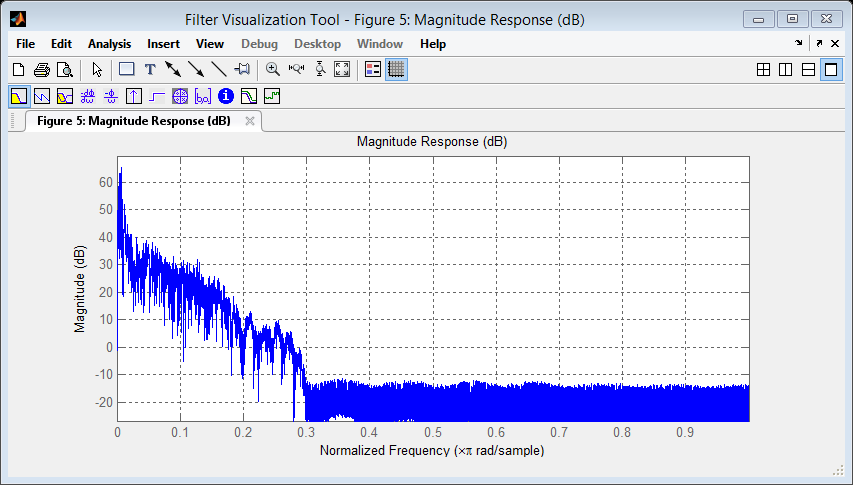
Fs = 160000











Que concluye de las 3 señales muestreadas, al interpretar las gráficas?

la reconstrucción de una señal es **idéntica** sin importar la tasa de muestreo, solo se contrae por algún factor de escala pero en general la calidad del audio obtenido solo es deficiente cuando Fs<<Fmax.

**Hoja de calificación**



**Universidad de San Carlos de Guatemala**

**Facultad de Ingeniería**

**Escuela de Mecánica Eléctrica Laboratorio de Electrónica Comunicaciones 1**

**Vacaciones Junio 2014**

**Auxiliar: Edvin Baeza**

**Practica No. 3**

**Grupo: C0070**

**Muestreo Con Matlab**

|  |  |  |
| --- | --- | --- |
| **Nombres Completos** | **Carnet** | **Asistencia puesta por el Auxiliar** |
| Sosimo Adan Canel Garcia | 200915083 |  |
| Angel Estuardo Lorenzo Marroquin | 201113947 |  |
| Diego Armando Taguite Flores | 201122930 |  |
| **Para uso exclusivo del Auxiliar** | | |
| Hora a la que entrego la practica. | Sello del Laboratorio | |
| : (AM/PM) |

**Instrucciones**

• Comentar todas las instrucciones del programa, que hace y para que sirve la instrucción.

• Para todas las gráficas debe investigar:

◦ Para que sirve la gráfica?

◦ Como se interpreta?

◦ Que información le proporciona?.

• Analizar la misma señal de audio que ustedes elijan (mínimo 15 segundos), y muestrear la señal a:

◦ Fs<<2Fmax

◦ Fs=2Fmax

◦ Fs>>2Fmax

• Que concluye de las 3 señales muestreadas, al interpretar las gráficas?