Reporte práctica cero ”Práctica Introducción a Matlab”.

optional subtitle below

Laboratorio de Análisis de Sistemas y Señales

Miranda Serrano Carlos Alberto, Robles Martínez Héctor

Febrero 17, 2022

1 Introducción

En esta práctica, aprenderemos a utilizar el lenguaje de programación Matlab, mediante el análisis de funciones senoidales y cosenoidales, de igual manera, analizaremos las partes pares e impares de las señales, aplicándo los conocimientos adquiridos a señales que representen notas musicales específicas. De igual manera, seremos capaces de graficar las señales, tanto en gráficas separadas, gráficas una sobre otra para poder diferenciarlas y gráficas divididas en pequeños subgráficos para poder comparar. Por último, se manejarán archivos de audio reales, para poder obtener señales del mundo real a las que podremos aplicar lo que aprendamos sobre parte par, parte impar, periodicidad y amplitud de una señal.

2 Desarrollo de actividades

2.1 Periodo y amplitud de una señal senoidal y otra cosenoidal

Considere las siguiente señales:

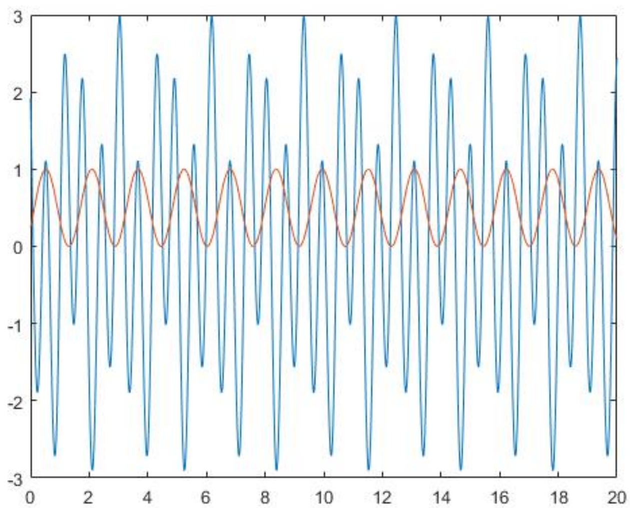
$$x(t) = 2cos(10t + 1) - sin(4t - 1) \tag{1}$$

$$x(t) = [cos(2t - \pi/3)]^2 \tag{2}$$

Determine de ser posible, su periodo y amplitud

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: H ctor Robles
% github: /Hector290601
% creation date: 02 10 2022
% last edit date: 02 17 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format rational;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6.1, considere las siguientes se ales:
%%%%%%%%%%%%%% create vector to store the independient variable
t = 0:0.00001:20;
%%%%%%%%%%%%%% signal 1:
x1 = 2 * cos( 10 * t + 1 ) - sin( 4 * t - 1 );
%%%%%%%%%%%%%% signal 2:
x2 = ( cos( 2 * t - pi / 3 ) ).^2;
%%%%%%%%%%%%%% To signal 1:
%%%%%%%%%%%%%% Periodicity of 2 * cos(10 * t+1):
periodCos = ( 2 * pi ) / abs( 10 );
%%%%%%%%%%%%%% Periodicity of sin(4 * t-1):
periodSin = ( 2 * pi ) / abs( 4 );
periodOne = periodCos / periodSin;
% >> periodOne = 2/5, 2 in Z, 5 in Z :. his period is Pi
%%%%%%%%%%%%%% To signal 2:
%%%%%%%%%%%%%% Periodicity of ( cos( 2 * t - pi / 3 ) ).^2:
periodTwo = ( 2 * pi ) / abs( 2 / 3 );
% >> periodTwo = period of the coseno
%%%%%%%%%%%%%% signal 1 is periodic :. the local max from a period is the
```

```
%%%%%%%%%%%% absolute max
max1 = max(x1);
%%%%%%%%%%%% signal 2 is periodic :. the local max from a period is the
%%%%%%%%%%%% absolute max
max2 = max(x2);
%%%%%%%%%%%% plot the signals
plot(t, x1);
hold on;
plot(t, x2);
```



2.2 Periodicidad de la suma de las notas musicales

Considere ahora que las notas musicales si, la, sol tienen las siguientes frecuencias.

$$f_{si} = 246.942[Hz] \tag{3}$$

$$f_{la} = 220[Hz] \tag{4}$$

$$f_{sol} = 192.998[Hz] \tag{5}$$

Donde el sonido se puede definir a través del siguiente modelo matemático:

$$x = sin(\omega t + \phi) \tag{6}$$

Donde

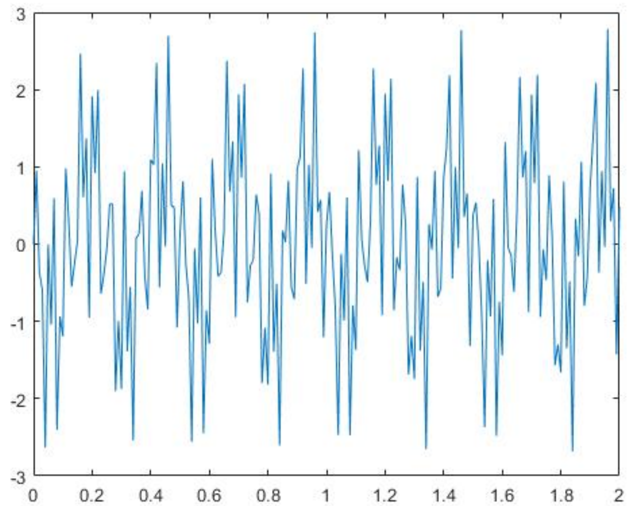
$$\omega$$

es la frecuencia fundamental de cada nota musical. Determine si

$$x_1 = x_{la} + x_{si} + x_{sol} \tag{7}$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: H ctor Robles
% github: /Hector290601
% creation date: 02 10 2022
% last edit date: 02 17 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format rational;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6.2 Considere ahora que las notas
    musicales
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% si, la y sol tienen kas siguientes
    funciones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% fundamentales f_si = 246.942 [Hz],
    f_la = 220
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [Hz] y f_sol = 195.998 [Hz]. Donde
    el sonido
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% se puede definir a trav s del
    siguiente
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% modelo matem tico:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% x = sin( wt + p )
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% donde w es la frecuencia fundamental
    de cada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% nota musical. Determine si:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% x_1 = x_si + x_la + x_sol
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% es peri dica o no.
%%%%%%%%%%%%%% create vector to store the independient variable
t = 0:0.01:2;
%%%%%%%%%%%%%% frequency of si
f_si = 246.042;
%%%%%%%%%%%%%% frequency of sol
f_sol = 195.998;
%%%%%%%%%%%%%% frequency of la
f_la = 220;
%%%%%%%%%%%%%% w of si
w_si = 2 * pi * f_si;
%%%%%%%%%%%%%% w of sol
w_sol = 2 * pi * f_sol;
%%%%%%%%%%%%%% w of la
w_la = 2 * pi * f_la;
%%%%%%%%%%%%%% "sound" of si
x_si = sin( w_si * t);
%%%%%%%%%%%%%% "sound" of sol
x_sol = sin( w_sol * t);
%%%%%%%%%%%%%% "sound" of la
x_la = sin( w_la * t);
%%%%%%%%%%%%%% Sum of all the "sounds"
xs = x_sol + x_si + x_la;
%%%%% Periodicity of periodSumSiSol + periodla:
period_sum_si_sol_la = ( 2 * pi ) ./ abs( xs );
%%%%%%%%%%%%%% plot the sum
figure;
plot(t, xs);
```

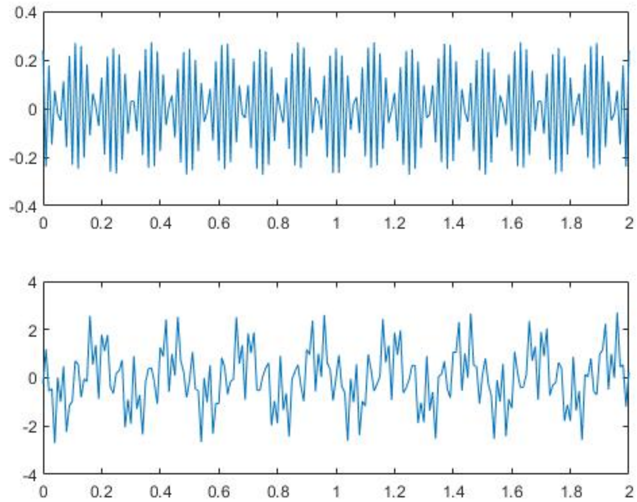


2.3 Parte par e impar

De la suma encontrada en el ejercicio anterior, obtenga la parte par e impar de la señal.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: H ctor Robles
% github: /Hector290601
% creation date: 02 10 2022
% last edit date: 02 17 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format rational;
%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6.3 De la suma encontrada en el
    ejercicio
%%%%%%%%%%%%%%%%%%%%%%%%%%%% anterior, obtenga la parte par e
    impar de la
%%%%%%%%%%%%%%%%%%%%%%%%%%%% funci n
%%%%%%%%%%%%%% even part of x[n] = 1/2{x[n]+x[-n]}
even = 1 / 2 * ( xs +  flip(xs));
%%%%%%%%%%%%%% odd part of x[n] = 1/2{x[n]-x[-n]}
odd = 1 / 2 * ( xs - flip(xs));
%%%%%%%%%%%%%% plot the odd and even part of the sum.
figure()
subplot(2, 1, 1);
plot(t, even);
```

```
subplot(2, 1, 2);
plot(t, odd);
```



2.4 Archivos de audio

Obtenga un archivo de audio y realice las siguientes actividades

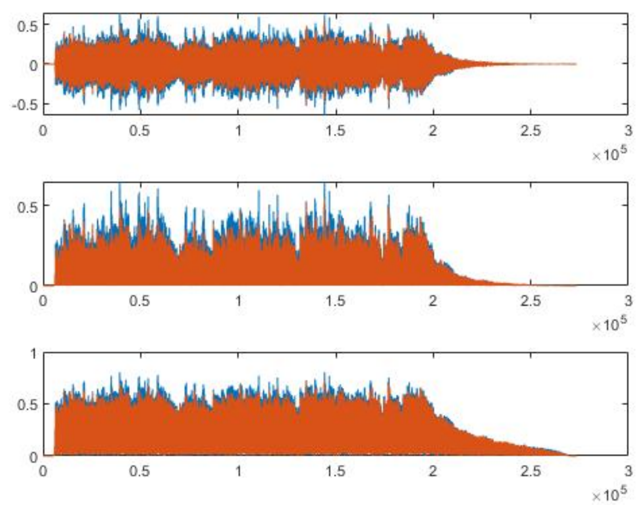
1. Obtenga la parte par e impar de la señal
2. Obtenga la raíz cuadrada y el valor absoluto de la señal de audio ¿cómo se modifica el sonido con cada una de estas operaciones?
3. Sume dos señales de aurio ¿cómo se modificó la gráfica?
4. Obtenga las componentes par e impar de la suma anterior

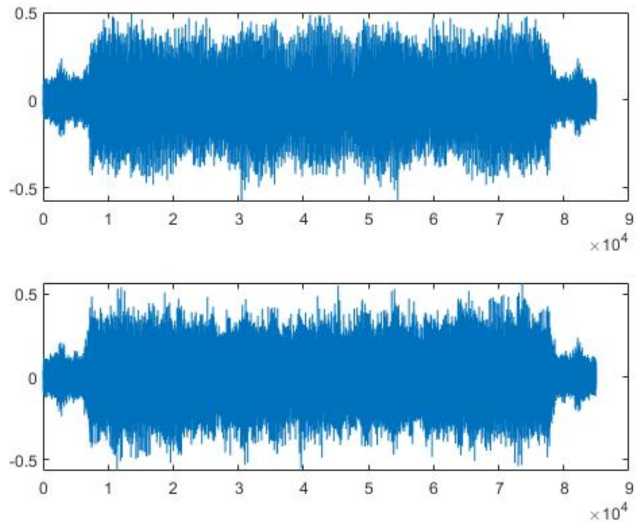
Aquí tuvimos problemas para reproducir el audio, pero en clase se escuchó más agudo cuando se obtuvo la raíz cuadrada y muy ruidoso en el valor absoluto.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: H ctor Robles
% github: /Hector290601
% creation date: 02 10 2022
% last edit date: 02 17 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format rational;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6.4 obtenga un archivo de audio y realice lo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% siguiente
%%%%%%%%%%%%% read the audio called "sound.ogg"
[ y1, Fs1 ] = audioread('sound.ogg');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Obtenga la parte par e impar de la se al
%%%%%%%%%%%%% even part of x[n] = 1/2{x[n]+x[-n]}
even_sound = 1 / 2 * ( y1 +  flip(y1));
%%%%%%%%%%%%% odd part of x[n] = 1/2{x[n]-x[-n]}
odd_sound = 1 / 2 * ( y1 - flip(y1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Obtenga la ra z cuadrada y el valor absoluto
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% de la se al de audio  cmo  se modifica el
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sonido con cada una de estas
    operaciones?
%%%%%%%%%%%%%% Absolute value of sound
y1_abs = abs(y1);
%%%%%%%%%%%%%% Square root of sound
y1_sqrt = sqrt(y1_abs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Sume dos se ales de audio   cmo
    se modific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% la gr fica?
%%%%%%%%%%%%%% read the audio called "sound_to_sum.ogg"
[ y2, Fs2 ] = audioread('sound_to_sum.ogg');
%%%%%%%%%%%%%% Absolute value of sound
y2_abs = abs(y2);
%%%%%%%%%%%%%% Square root of sound
y2_sqrt = sqrt(y2_abs);
%%%%%%%%%%%%%% Get the size of the first sound
size1 = size(y1);
%%%%%%%%%%%%%% Get the size of the second sound
size2 = size(y2);
%%%%%%%%%%%%%% crop the size to the lowest size
%%%%%%%%%%%%%% select lower size
if size1 < size2
    %%%%%%%%%%%%%% size1 < size2
    %%%%%%%%%%%%%% crop y1 to y2 size
    yn1 = y1(1:size1);
    %%%%%%%%%%%%%% crop y2 to y2 size
    yn2 = y2(1:size1);
else
    %%%%%%%%%%%%%% size1 >= size2
    %%%%%%%%%%%%%% crop y1 to y1 size
    yn1 = y1(1:size2);
    %%%%%%%%%%%%%% crop y1 to y2 size
    yn2 = y2(1:size2);
end
%%%%%%%%%%%%%% sum the two sounds
ys = yn1 + yn2;
```

```
%%%%%%%%%%%%%% even part of x[n] = 1/2{x[n]+x[−n]}
even_sounds = 1 / 2 * ( ys +  flip(ys));
%%%%%%%%%%%%%% odd part of x[n] = 1/2{x[n]−x[−n]}
odd_sounds = 1 / 2 * ( ys − flip(ys));
%%%%%%%%%%%%%% plot the original sound, the abs part and the sqrt
figure;
subplot(3, 1, 1);
plot(y1);
subplot(3, 1, 2);
plot(y1_abs);
subplot(3, 1, 3);
plot(y1_sqrt);
figure;
subplot(2, 1, 1);
plot(even_sounds);
subplot(2, 1, 2);
plot(odd_sounds);
```





3 Análisis de resultados

Los resultados de la práctica, concuerdan con lo aprendido en la clase teórica, por lo que podemos confiar en la teoría aplicada a casos reales. Es interesante cómo se modifica la señal de audio, y por consecuencia cómo se escucah el audio, al obtener su raíz cuadrada, así como su valor absoluto, es muy diferente al cambio de stereo a mono, donde también combinamos ambos canales de audio en uno solo.

4 Conclusiones

Podemos recatar muchas cosas aprendidas de la práctica, entre ellas, el uso básico del lenguaje Matlab, así como la observación de los elementos teóricos de la clase con los resultados experimentales obtenidos en esta clase. También, podemos ver que no podemos asegurar que una señal es par o impar a simple vista, puesto que se tienen que hacer los desarrollos matemáticos correspondientes para tener la certeza.

Puede encontrar el código, archivos de audio y demás cosas referentes a la práctica en: https://github.com/Hector290601/signals_and_systems/tree/main/practices/first