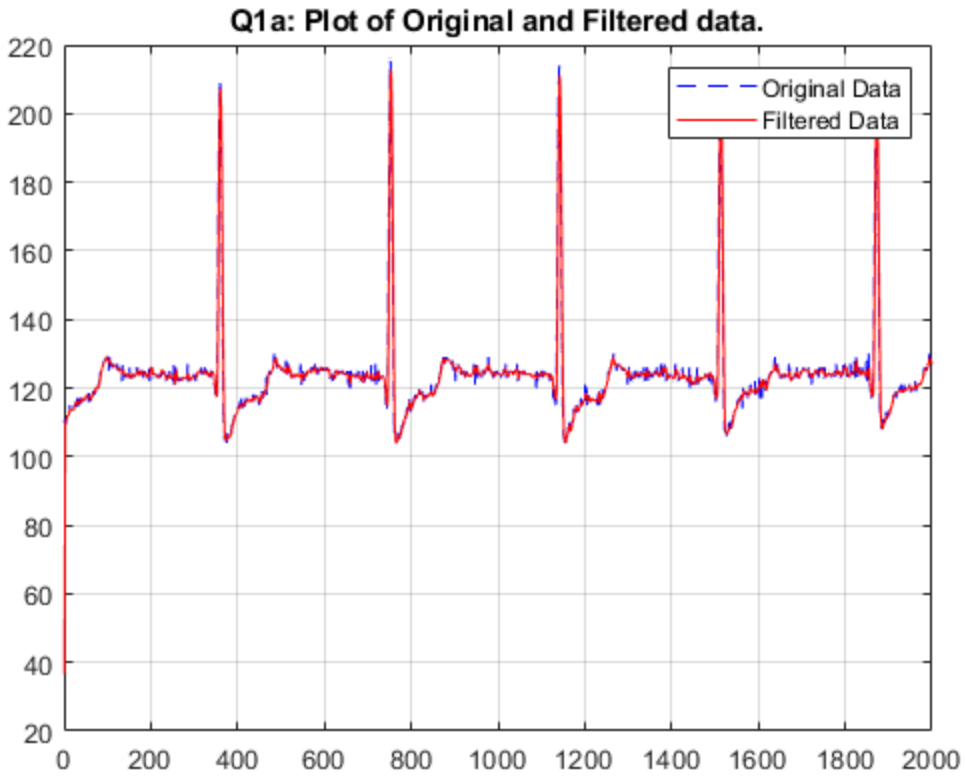
Table of Contents

Dan Otieno. EE 384 -> Spring '24. Classwork 2. Due date: 01/30/24. Sound did not work on my computer on Problem 3, I got the following.. "Error using matlabshared.aysncio.internal.MessageHandler/onError". "PortAudio Error: Unanticipated host error".

Problem 1a.

Re-do problem 3 of classwork 1. Implement y(n) = x(n) + x(n+1) + x(n+2) / 3 using filter command...

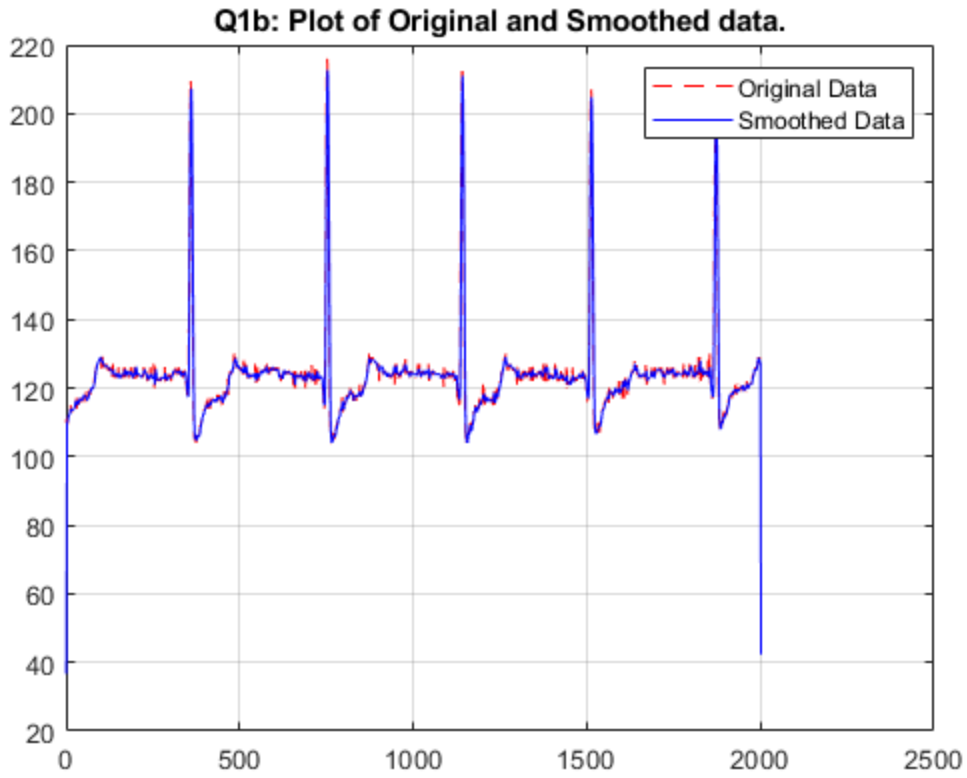
```
%...to filter out the high frequency components of the ECG signal.
clear all; close all; clc;
load SAMPLE_ECG;
x = ECG_Data;
a = 1;
b = [1/3 1/3 1/3];
y = filter(b,a,x);
figure(1)
t = 1:length(x);
plot(t, x, 'b--');
hold on
plot(t, y, 'r-');
grid on
legend('Original Data', 'Filtered Data');
title('Q1a: Plot of Original and Filtered data.');
```



Problem 1b.

Convolute ECG signal with h = [1/3 1/3 1/3]...

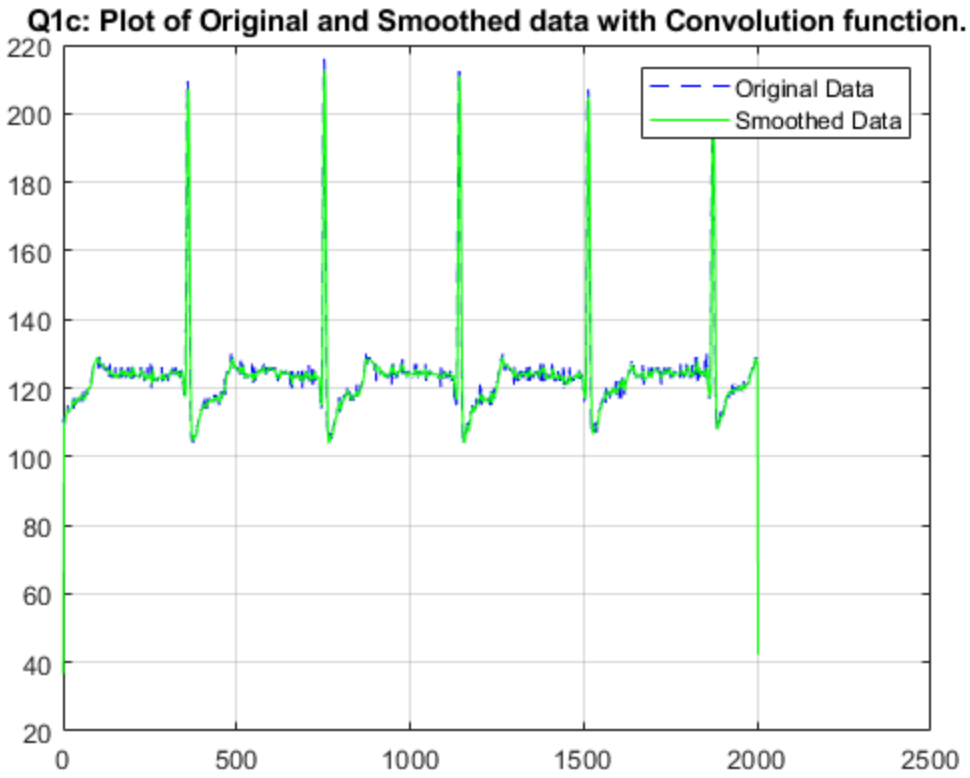
```
%...using conv function, then compare result with that of (a).
h = [1/3 1/3 1/3];
y1 = conv(x,h);
t1 = 1:length(y1);
figure(2)
plot(t, x, 'r--');
hold on
plot(t1, y1, 'b-');
grid on
legend('Original Data', 'Smoothed Data');
title('Q1b: Plot of Original and Smoothed data.');
```



Problem 1c.

Write my own convolution, use it to re-do (b).

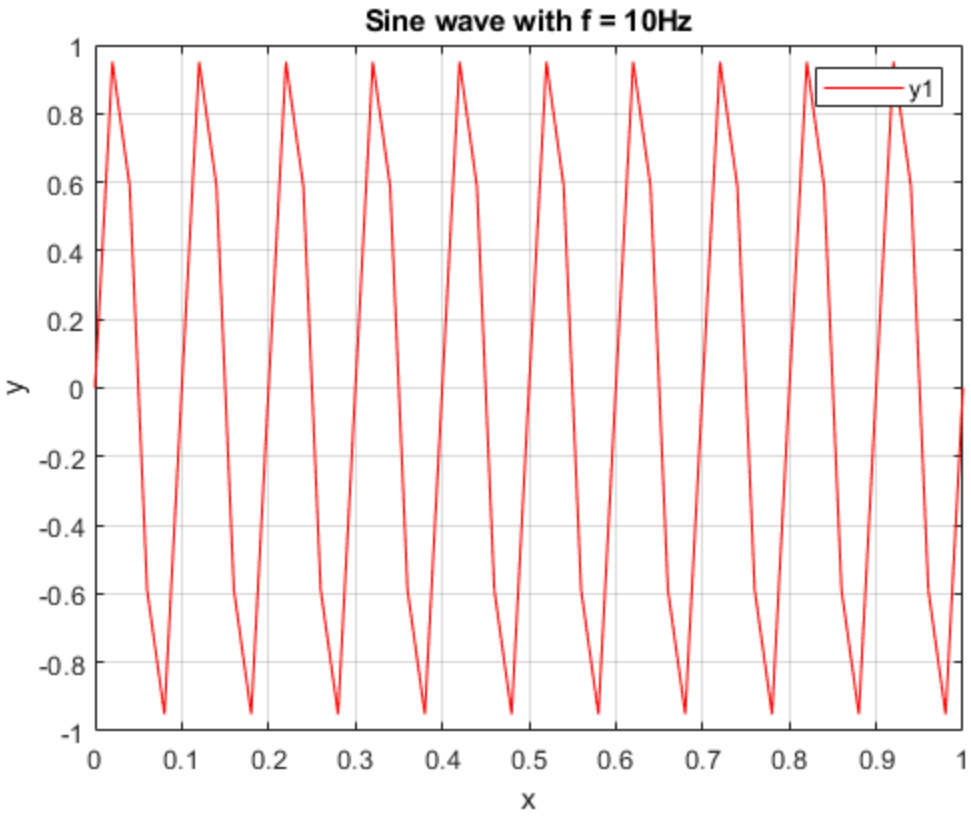
```
[y2] = fnConv(x,h);
t = 1:length(x);
t1 = 1:length(y2);
figure(3)
plot(t, x, 'b--');
hold on
plot(t1, y2, 'g-');
grid on
legend('Original Data', 'Smoothed Data');
title('Q1c: Plot of Original and Smoothed data with Convolution function.');
```



Problem 2a.

Create and plot a sine signal y1(t) = sin (2*pi*f1*t). Sampling rate, fs = 50Hz. f1 = 10Hz.

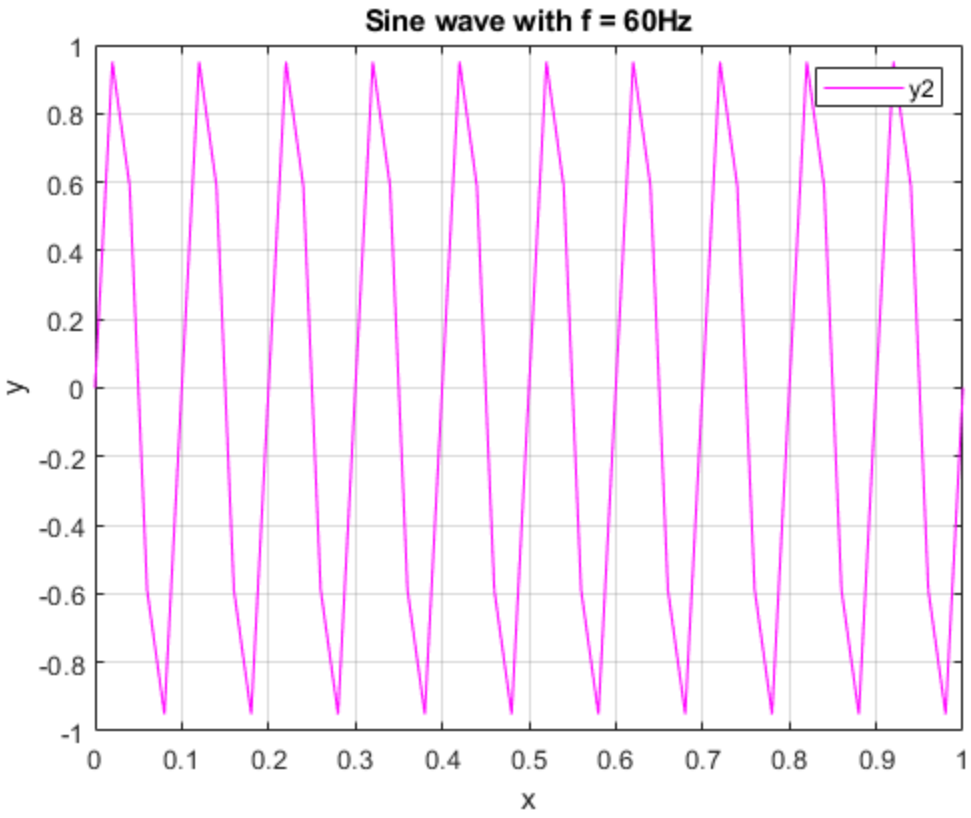
```
fs = 50;
ts = 1/fs;
f1 = 10;
t = 0:ts:1;
y1 = sin(2*pi*f1*t);
figure(4)
plot(t, y1, 'r-');
grid on
title('Sine wave with f = 10Hz');
xlabel('x');
ylabel('y');
legend ('y1');
```



Problem 2b.

Plot signal with f2 = 60Hz.

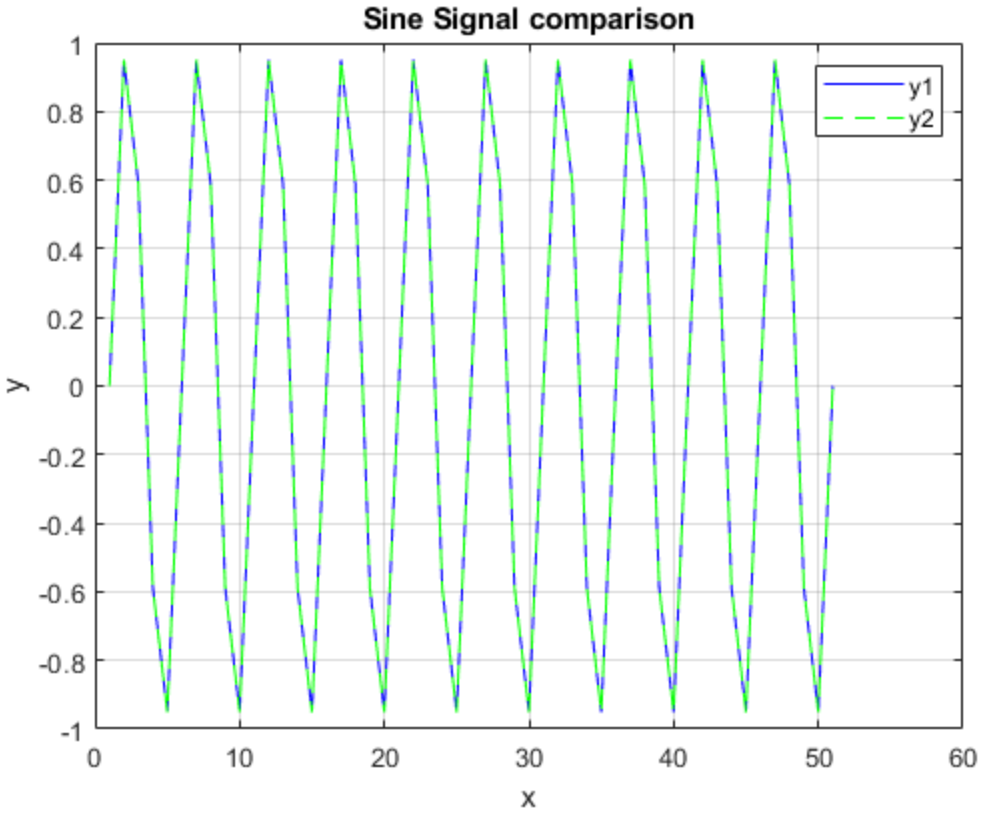
```
f2 = 60;
y2 = sin(2*pi*f2*t);
figure(5)
plot(t, y2, 'm-');
grid on
title('Sine wave with f = 60Hz');
xlabel('x');
ylabel('y');
legend ('y2');
```



Problem 2c.

Plot both y1 and y2, compare.

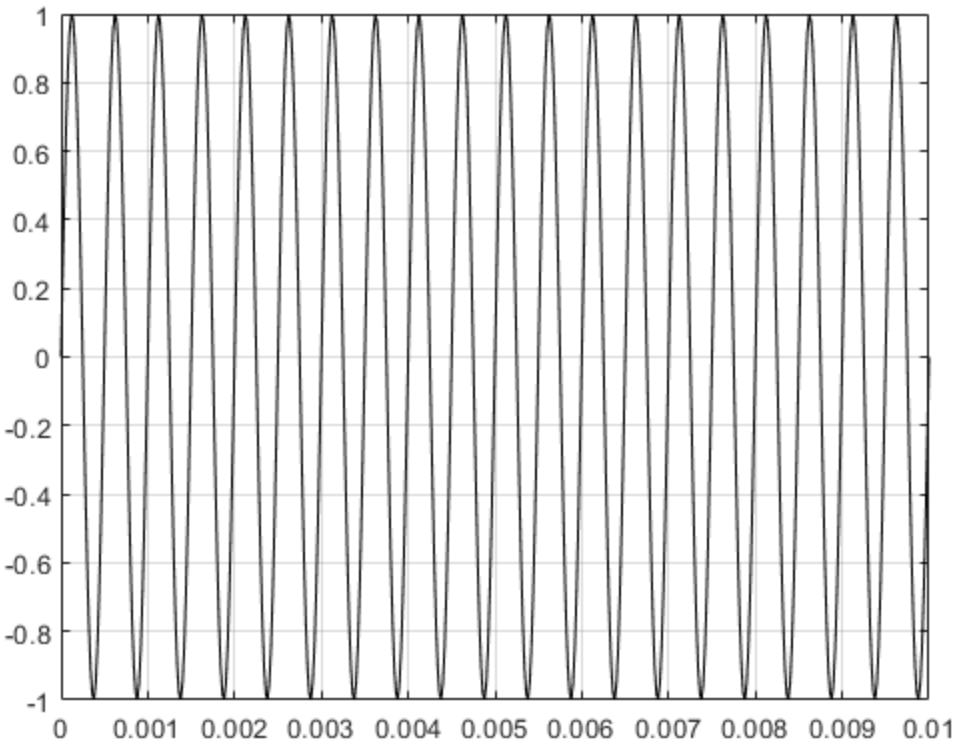
```
figure(6)
plot(y1, 'b-');
hold on;
plot(y2, '--g');
grid on
title('Sine Signal comparison');
xlabel('x');
ylabel('y');
legend('y1', 'y2')
```



Problem 3a.

Create and plot 2kHz signal y1(t) = sin(2*pi*f1*t). Let horizontal axis vary from 0 to 0.01s. Use command 'sound(signal, sampling rate) to listen to tone.

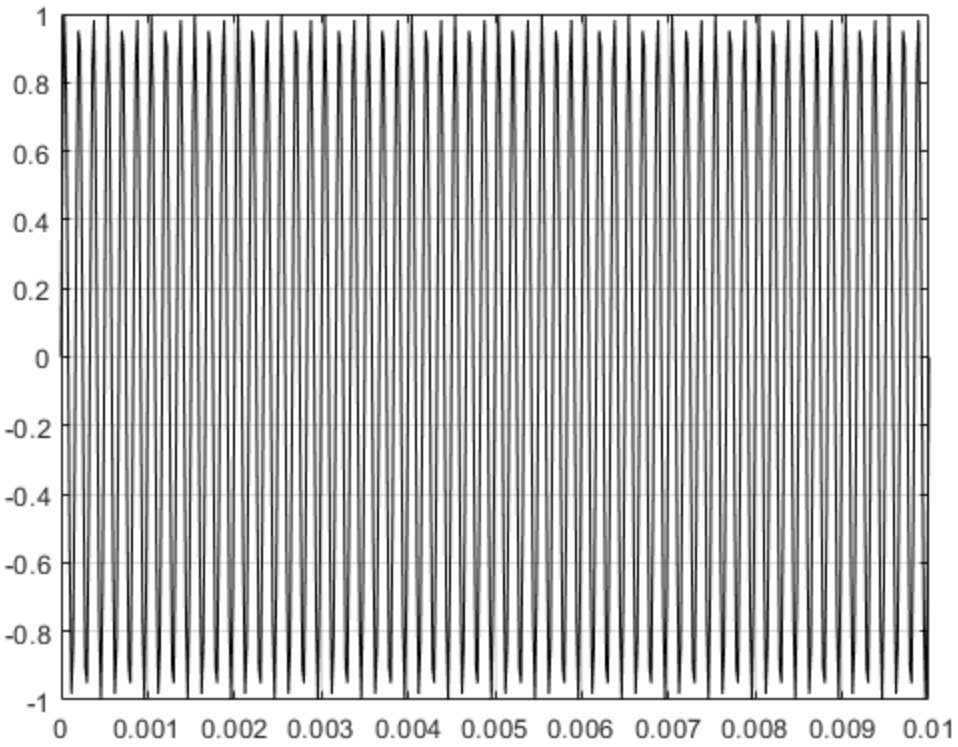
```
fs = 50e3; %sampling rate
f1 = 2e3; %signal frequency.
t = 0:1/fs:0.01;
y1 = sin(2*pi*f1*t);
sound(y1,fs);
figure(7)
plot(t, y1, 'k-');
grid on
```



Problem 3b.

Create and plot 6kHz signal y2(t) = sin(2*pi*f2*t). Use command 'sound(signal, sampling rate) to listen to tone. How does sound differ for y1 and y2?

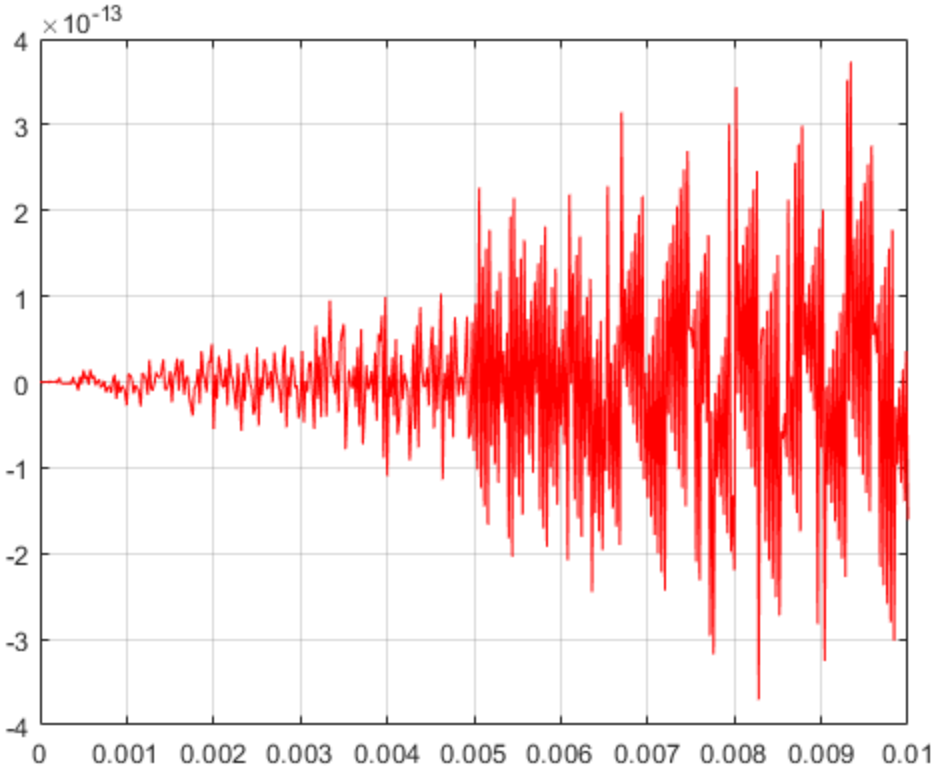
```
f2 = 6e3;
t = 0:1/fs:0.01;
y2=sin(2*pi*f2*t);
sound(y2,fs);
figure(8)
plot(t, y2, 'k-');
grid on
```



Problem 3c.

Create and plot 25kHz signal y3(t) = sin(2*pi*f3*t). Use command 'sound(signal, sampling rate) to listen to tone. Can we still hear the sound?

```
f3 = 25e3; %Signal frequency.
t = 0:1/fs:0.01;
y3 = sin(2*pi*f3*t);
sound(y3,fs);
figure(9)
plot(t, y3, 'r-');
grid on
% undersampled, so no signal.
```



Functions.

Convolution for Q1c. Credit to Sayan Samanta for function pseudocode.

```
function [y] = fnConv(x,h)
    N = length(h);
    M = length(x);
    y = zeros(1, M+N-1);
    for n = 1:M + N-1
        y(n) = 0;
        for k = 1:M
            if(n-k+1)>0 && (n-k+1)<=N
                y(n) = y(n) + x(k)*h(n-k+1);
            end
        end
    end
end
```

Published with MATLAB® R2023a