**DSP Lab**

**Week 1**

**Version 1:**  
Attempted to get accurate fitted curve by splitting the signal into multiple segments of equal length. The length chosen was 1001 units since anything below caused too great of a mathematical error in one of the fitted signals. Resulting signal was one which was similar to original sound signal, but more rectangular. The resulting sounds signal sounds like beeps without recognition of any words.

**All graphs with fitted curves in video below**:



**Original Sound Signal**

Chart, histogram

Description automatically generated

**Fitted Signal:**

Chart

Description automatically generated

**Spectrum of Fitted Signal:**

Chart

Description automatically generated

**Code:**

clear;

clc;

[x, Fs] = audioread('SignalHW2.wav');

% y=[1:length(x)]';

% plot(x,y)

% y1 = [1:2500]';

% y2 = [2500:5000]';

% f1 = fit(y1,x1,'fourier2');

% f2 = fit(y2,x2,'fourier2');

% plot(f1, y1, x1)

% plot(f2, y2, x2)

vector = [];

incrementCount = 1001;

for a = 1.0 : incrementCount : 46954 - (mod(46954,incrementCount))

x1 = x(a:a+(incrementCount-1));

y1 = [a:a+(incrementCount-1)]';

f1 = fit(y1,x1,'fourier2');

for b = 1.0 : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

end

y = vector';

plot(x)

plot(y')

sound(x, Fs)

pause(5);

sound(vector, Fs)

Len\_y=length(y); p=abs(fft(y)); f=(Fs/Len\_y)\*(1:Len\_y); figure; plot(f,p);

title('FFT of signal'); ylabel('|Y(jw)|'); xlabel('f(Hz)'); grid;

**Version 2:**Attempted to get a more clear fitted representation of the signal using Fourier like last version with the difference being that each increment has a unique length in which starts at the beginning at each of the signal bumps of the signal and ends at the ends of these signal bumps. The result ended up being not as good as the attempts of Version 1. It appears that better results are received with smaller increments.

**All graphs with fitted curves:**

Graphical user interface, chart

Description automatically generated

Chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

**Original Sound Signal**

Chart

Description automatically generated

**Fitted Sound Signal:**

Chart

Description automatically generated

**Spectrum of Fitted Sound Signal**

Chart, histogram

Description automatically generated

**Code:**

clear;

clc;

[x, Fs] = audioread('SignalHW2.wav');

vector = [];

% 1st Segment

initialValue = 1;

incrementCount = 2525;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% Second Segment

initialValue = incrementCount;

incrementCount = 5456;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 3rd Segment

initialValue = incrementCount;

incrementCount = 9465;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 4th Segment

initialValue = incrementCount;

incrementCount = 16843;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 5th Segment

initialValue = incrementCount;

incrementCount = 19703;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 6th Segment

initialValue = incrementCount;

incrementCount = 25741;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 7th Segment

initialValue = incrementCount;

incrementCount = 25901;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 8th Segment

initialValue = incrementCount;

incrementCount = 29132;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 9th Segment

initialValue = incrementCount;

incrementCount = 32097;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 10th Segment

initialValue = incrementCount;

incrementCount = 35169;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 11th Segment

initialValue = incrementCount;

incrementCount = 41939;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

% 12th Segment

initialValue = incrementCount;

incrementCount = 46927;

x1 = x(initialValue:incrementCount);

y1 = [initialValue:incrementCount]';

f1 = fit(y1,x1,'fourier2');

for b = initialValue : incrementCount

vector(end+1) = f1(b);

end

plot(f1, y1, x1)

y = vector';

plot(x)

plot(y')

sound(x, Fs)

pause(5);

sound(vector, Fs)

Len\_y=length(y); p=abs(fft(y)); f=(Fs/Len\_y)\*(1:Len\_y); figure; plot(f,p);

title('FFT of signal'); ylabel('|Y(jw)|'); xlabel('f(Hz)'); grid;