

Lab 04: Fourier Analysis & Synthesis

Introduction

This two week lab focused on the fourier analysis formula. If you are given a periodic signal $x(t)$ with period T , $x(t)$ can be represented by:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$$

where $\omega_0 = \frac{2\pi}{T}$ and

$$a_k = \frac{1}{T} \int_0^T x(t) e^{-jk\omega_0 t} dt \quad \text{for all } k$$

For a general signal, normally an infinite amount of harmonics is needed, but sometimes a finite amount of them give a fairly good approximation. In order to find the a_k coefficient, we use a simple method to approximate the integrals called the Riemann approximation. Suppose the signal $x(t)$ is a nice integral, we can take the Riemann approximation and find the sampling time to be T_s in an integer fraction of period T . In other words, we have an integer number N of samples in a period. The following formula shows this:

$$a_k = \frac{1}{T} \int_0^T x(t) e^{-jk\omega_0 t} dt \simeq \frac{T_s}{T} \sum_{n=0}^{T/T_s} x(T_s n) e^{-jk\omega_0 T_s n}.$$

If you think about it, fixing the sampling time limits the attribute of the approximation. This is because T becomes too large with respect to $k\omega_0$ the frequency of the k th harmonic.

Part 1: Pre-lab

Exercise 1

For this pre-lab, we wrote a function “fanal” that takes in inputs a vector of samples of $x(t)$ worth one period of $x(t)$, the index of the k th coefficient, the period, and the sampling time. This function then outputs the approximate a_k coefficient. Below you will find my script for the “fanal” function:

```
function a_k = fanal(X_T, k, T, T_s)
%X_T is vector of samples of x(t)
%k is index of desired kth coefficient
%T is period
%T_s is sampling time
t=0.0:T_s:T-T_s;
w= 2*pi*(1/T);
complex=exp(-i*k*w*t); %complex exp
size(complex);
size(X_T);
a_k=(T_s/T)* complex * X_T;
end
```

Exercise 2

This next step is to develop a text signal defining $x_T = [\text{ones}(1000,1); \text{zeros}(1000,1)]$ with a period of .01 seconds:

```
yy=[ones(1000,1); zeros(1000,1)];  
fanal(yy,-2,.01,(.01/2000))
```

- a) The sampling time needs to be $T/2000$ otherwise the inner dimensions of matrix multiplication will not be correct, getting an error of:

```
Error using *  
Inner matrix dimensions must agree.  
Error in fanal (line 11)  
a_k=(T_s/T)* complex * X_T;  
Error in fanaltest (line 3)  
fanal(yy,-2,.01,(.01/1000))
```

- b) $a_0=0.5000$; $a_1=0.0005 - 0.3183i$; $a_2=6.1691e-17 - 2.5988e-17i$; $a_3=0.0005 - 0.1061i$

- c) $a_1=0.0005 + 0.3183i$; $a_2=6.1691e-17 + 2.5988e-17i$; $a_3=0.0005 + 0.1061i$

Exercise 3

The synthesis formula gives us the ability to form periodic signals from linear combinations of harmonic complex exponentials. If we assume that the amount of non-zero coefficient is finite, then we have the formula:

$$\tilde{x}(T_s n) = \sum_k a_k e^{jk\omega_0 T_s n}$$

The next part was implement the function “fsynt”. This function inputs a column vector of coefficients, sampling time, and the fundamental period. In the end, it outputs the vector of T/T_s samples corresponding to one period of $x(\sim)$. Please view the “fsynt” code below:

```
%Ex3  
% Lab 4 of EE 224  
% Synthesize a waveform based on the Fourier coefficients  
% It is assumed that the signal is real in the time domain  
  
function x=fsynt(C, T, Ts)  
  
% INPUT: C - the Fourier coefficients [a_0, a_1, a_2, ..., a_m];  
%         T - The fundamental period  
%         Ts - The sampling period  
%  
% OUTPUT: x - a vector containing the samples taken from t in [0, T]  
  
C=C(:); % make sure that C is a column vector  
CC = [conj(C(end:-1:2)); C]; % include the negative frequencies
```

```
N=floor(T/Ts); % will have this many points within one period
n=[0:N]';      % the discrete times

m=length(C)-1; % the max frequency index
k=-m:m;        % indices of the frequencies
F=exp(j*2*pi/T*Ts*n*k); % each row corresponds to a different time

x=F*CC; % The synthesis summation for all n's
end
```

Part 2: Laboratory Assignment

For the actually lab assignment, we were given two signals of a trumpet and a whistle. Both of the signals are sampled at $f_0=44100$ Hz and represented by one period of actual trumpet and whistle tones.

- a) Computation of the period of the two signals: 44100 is the sampling frequency and trumpet file contained 170 data points (having a 170x1 value). The whistle file contains 85 data points (having a 85x1 value).

```
x=(1/44100)*170;
disp(x)
```

```
ans= .0039 ← period of trumpet signal
```

```
y=(1/44100)*85;
disp(y)
```

```
ans= .0019 ← period of whistle signal
```

- b) For each signal, you can find the first 9 harmonic coefficients using the function “fanal.”

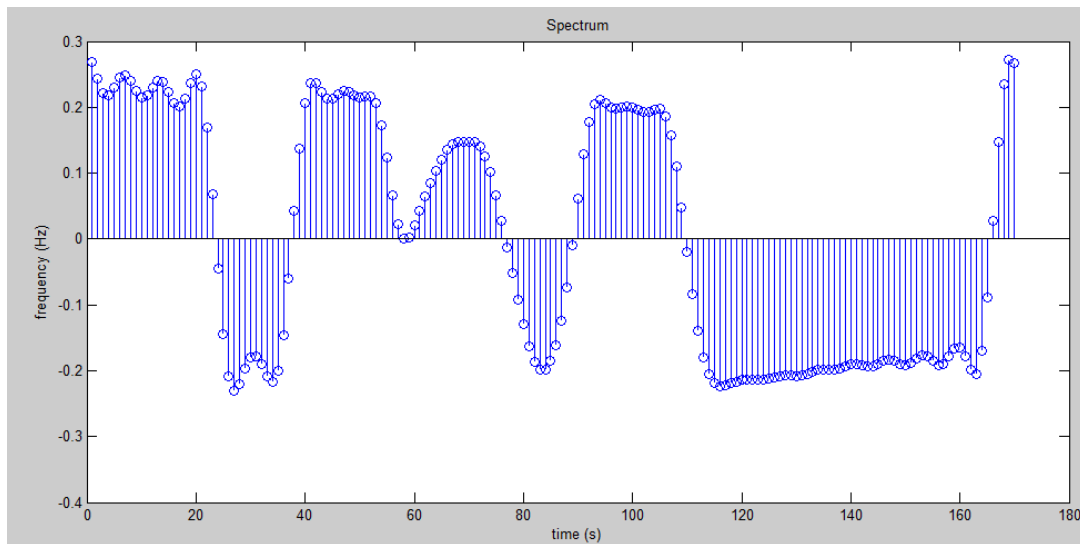
Trumpet spectrum

```
for k= -9: 1 : 9
fanal(trumpet,k,.0039,(.0039/170))
end
```

```
%ans to -9 = 0.0122 - 0.0067i
%ans to -8 =0.0161 + 0.0013i
%ans to -7= 0.0221 - 0.0197i
%ans to -6= -0.0376 + 0.0006i
%ans to -5= 0.0217 + 0.0194i
%ans to -4= 0.0035 + 0.0600i
%ans to -3=0.0466 - 0.0110i
%ans to -2=0.0411 + 0.0258i
%ans to -1=-0.0099 + 0.0644i
%ans to 0=0.0019
%ans to 1=-0.0099 - 0.0644i
%ans to 2=0.0411 - 0.0258i
%ans to 3=0.0466 + 0.0110i
%ans to 4=0.0035 - 0.0600i
```

```
%ans to 5=0.0217 - 0.0194i  
%ans to 6=-0.0376 - 0.0006i  
%ans to 7= 0.0221 + 0.0197i  
%ans to 8=0.0161 - 0.0013i  
%ans to 9=0.0122 + 0.0067i
```

```
stem(trumpet)  
title('Spectrum');  
xlabel('time (s)');  
ylabel('frequency (Hz)');
```

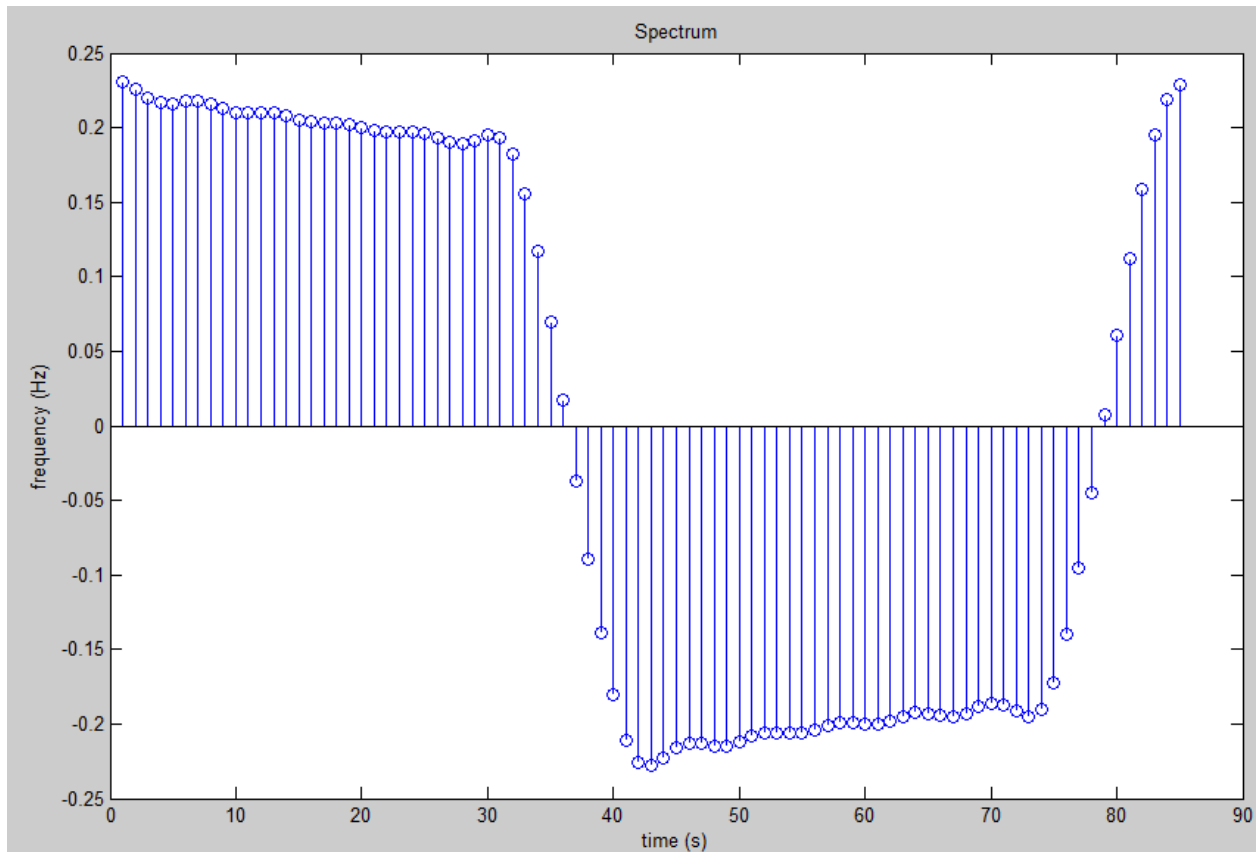


Whistle Spectrum

```
for kk= -9: 1: 9  
    fanal(whistle, kk, .0019, (.0019/85))  
end
```

```
%ans to -9= -0.0017 - 0.0003i  
%ans to -8=1.0874e-04 + 6.5328e-05i  
%ans to -7= -0.0025 - 0.0067i  
%ans to -6=1.8517e-04 + 9.0810e-05i  
%ans to -5= 0.0101 - 0.0138i  
%ans to -4= 1.7095e-04 + 3.3897e-04i  
%ans to -3= 0.0380 + 0.0018i  
%ans to -2=-2.7988e-04 + 7.5134e-04i  
%ans to -1= 0.0697 + 0.1076i  
%ans to 0= 0.0011  
%ans to 1= 0.0697 - 0.1076i  
%ans to 2=-2.7988e-04 - 7.5134e-04i  
%ans to 3= 0.0380 - 0.0018i  
%ans to 4= 1.7095e-04 - 3.3897e-04i  
%ans to 5= 0.0101 + 0.0138i  
%ans to 6= 1.8517e-04 - 9.0810e-05i  
%ans to 7= -0.0025 + 0.0067i  
%ans to 8=1.0874e-04 - 6.5328e-05i  
%ans to 9=-0.0017 + 0.0003i
```

```
stem(whistle)
title('Spectrum');
xlabel('time (s)');
ylabel('frequency (Hz)');
```



c) Please view the above spectrums in part b. You will find the spectrum of the frequency components for both signals. The main differences between the two signals are the various frequencies. The trumpet sound has a very rich spectrum with a more complex sound. On the other hand, the whistle sound doesn't have much of a harmonic component about the ninth nor an even harmonic component.

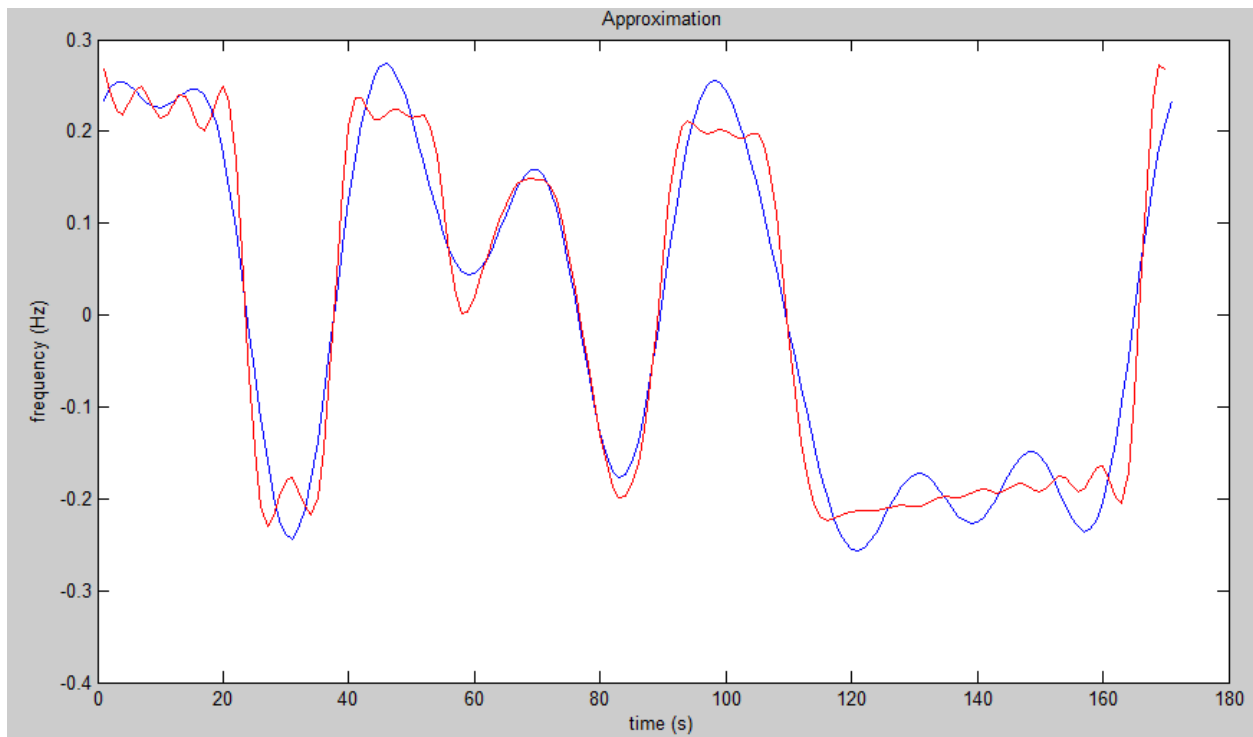
d) This is an approximation for each signal by using the periods in part a), the coefficients found, and the sampling time $1/f_0$ in conjunction with the function "fsynt":

Trumpet Approximation

```
for kk= 0: 1 : 9
A(kk+1)= fanal(trumpet, kk, .0039, (.0039/170));
end
```

```
b=fsynt(A, .0039, .0039/170);
```

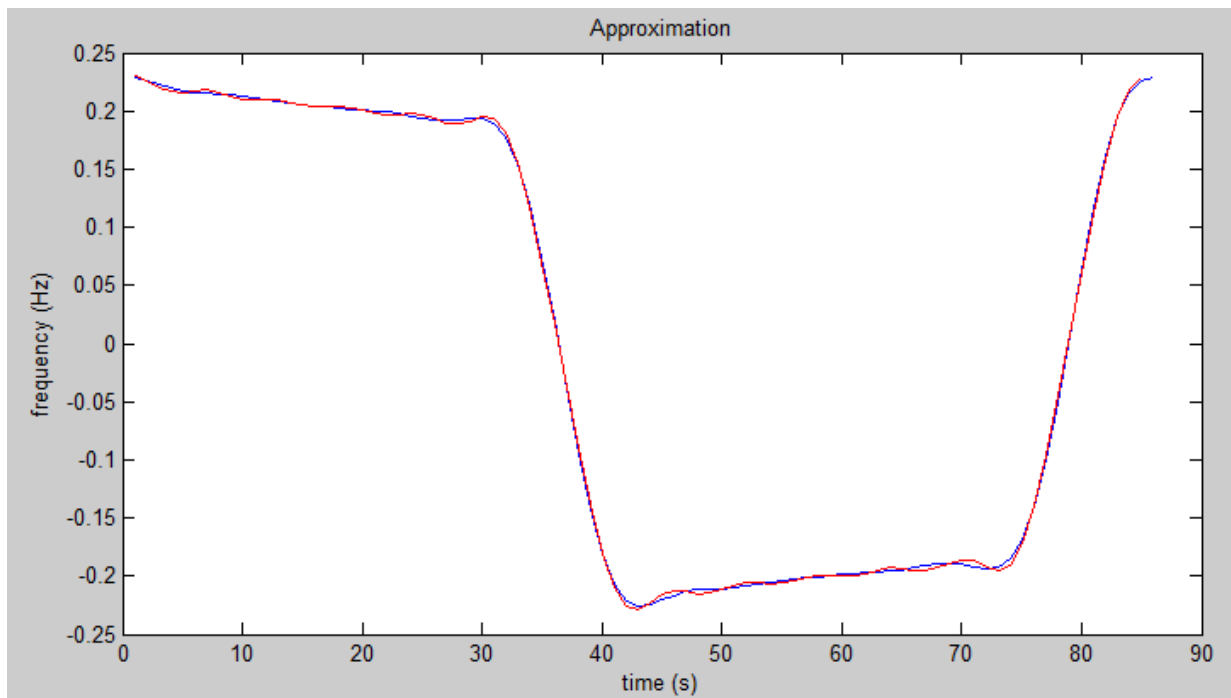
```
plot(b)
xx=real(b)
plot(xx, '-b')
hold on
plot(trumpet, '-r')
xlabel('time (s)');
ylabel('frequency (Hz)');
title('Approximation');
```



Whistle approximation

```
for qq= 0: 1 : 9
A(qq+1)= fanal(whistle,qq,1.9, (1.9/85));
end
```

```
bb=fsynt(A,1.9,1.9/85);
xx=real(bb)
plot(xx, '-b')
hold on
plot(whistle, '-r')
xlabel('time (s)');
ylabel('frequency (Hz)');
title('Approximation');
```



- e) Please see part d. You will find the new coefficient plot on the same plot of the given signal and its synthesized approximation. The quality of the approximation is actually very close with minor faults as you can see. The approximation of the whistle is almost exactly in line with the original signal.
- f) The code below generates a new signal for each signal by repeating them for 1000 periods. Also, the function “sound” was used along with the sampling frequency to hear the sound generated. I used the code as follows to repeat and enlarge the signal to 1000 times. When this happens, I can hear an obvious difference between the enlarged signal and the original one. The sound of both whistle and trumpet seems to be longer and more like a trumpet and whistle sound after the signal was repeated for 1000 periods.

Whistle:

```
for qq= 0: 1 : 1000 %make 1000 copies in the for loop  
BB(qq+1)= fanal(whistle,qq,1.9, (1.9/85));  
end
```

```
bb=fsynt(BB,1.9,1.9/85);  
xx=real(bb)  
plot(xx, '-b')  
hold on  
plot(whistle, '-r')  
xlabel('time (s)');  
ylabel('frequency (Hz)');  
title('Approximation');
```

```
sound(xx,44100);  
sound(whistle,44100);
```

Trumpet:

```
for kk= 0: 1 : 1000  
A(kk+1)= fanal(trumpet, kk, .0039, (.0039/170));  
end
```

```
b=fsynt(A, .0039, .0039/170);  
plot(b)  
xx=real(b)  
plot(xx, '-b')  
hold on  
plot(trumpet, '-r')  
xlabel('time (s)');  
ylabel('frequency (Hz)');  
title('Approximation');
```

```
sound(xx,44100);  
sound(trumpet,44100);
```