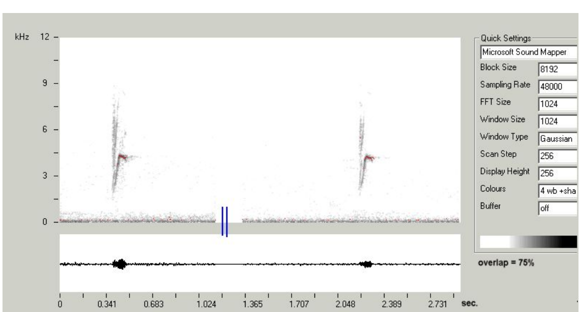
March 21st 2016

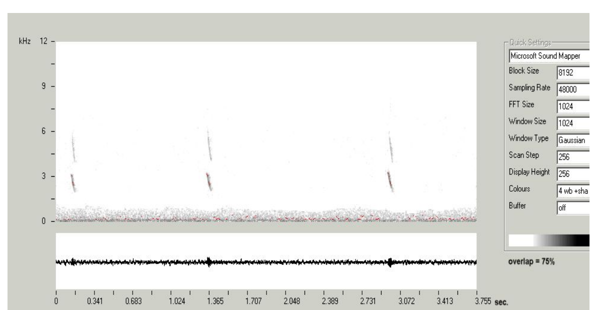
P1: Project Update
CS 489 001: Computational Audio
Sarah Fraser (sm2frase : 20458408)

Reflection on P0 Goals & Direction of Project

As I begun working on this project I decided that it would make sense to do a lot more research into the area of audio fingerprint matching as well as into the area of bird acoustics. I wanted to gain a much better sense of how I could transform a generic music matching algorithm into one that would work for bird acoustics. My findings led me to understand that what I thought would be much more simpler than music matching would actually be much more complex and difficult to implement.

One of the biggest components to basic fingerprinting and matching of music is that the actual song/audio does not fundamentally change. The notes stay in the same progression that they always are in for a song recording and the speed/pitch remain constant (generally). The only variance is the quality and volume of the recording to be matched. This means that when proper steps are taken to exclude noise, ignore quality and volume, songs are basically the same in their fingerprint and thus can be matched. This is not the case for bird acoustics. A bird's song/call can not only vary in pitch and tone, but can also vary in the way the song is produced. A bird may add/subtract a note here and there, making each call fundamentally different even though they come from the same bird species. Take for example two recordings from this web page (http://www.birdsongs.it/songs/fringilla_coelebs/fringilla_coelebs.html) of the Chaffinch bird's song. Both recordings are of the same bird species, yet the call and their respective spectrograms are vastly different. Trying to use the DFT or spectrograms to create a unique fingerprint and match the two is almost impossible for the amount of time I have to complete this project.





Therefore, after realizing this fact I have decided to scale back my project (at the guidance of Prof. Mann) and create almost a "proof of concept" (POC). Instead of creating a program that will take in any bird call recording and try to match it to a database of already fingerprinted bird calls, I will try to match a recording to a database of about 10 fingerprint bird calls in which the input recording is a part of. This way I am able to avoid a huge variance in bird calls of the same species while still exhibiting a fingerprint matching algorithm.

Here are some additional info on the research I did

- Bird songs and calls of the same species actually vary between one another based on a geological location basis. Just like how humans have accents based on where they live, bird calls also have "accents". Thus, "[the] Tufted Titmouse in WI does not sound like a Tufted Titmouse in ME" (http://grow.cals.wisc.edu/environment/smart-birding)
- There are actually a group of professors from the College of Agriculture and Life Sciences working on this exact project. In their post (linked in the first point) they describe the same issues that I have found. In their latest update, they make a call for contributions in order to gain a big enough database of different bird call variations in order to accurately match a bird to its call.
- For the most part, music and songs stay within the same general range of notes. Bird calls on the other hand vary much more, especially from species to species. Therefore, unlike fingerprinting music, fingerprinting bird calls needs to be much more precise and have a broader range. Since I only plan on doing a POC, this will not be as hard to do since I will only have about 10 different calls with max 10 different ranges to work with.

Update of P0 Goals:

0) Choose 10 unique bird call recordings of around the same length to use in POC
1) Implement a basic algorithm that is able to create a fingerprint of a sound with no concern to the time domain or factors and then use it to try and match two of the same bird calls together. Essentially, use the DFT of one bird call and try to match its DFT (within a reasonable error bound) to another DFT of the same bird call recording.
2) Do goal 1 but with more concern to error bounds and not a perfect matching
3) Do goal 2 but with focus on the time domain
4) Finalize the fingerprinting algorithm and create one bird call fingerprint to use for comparison on all other inputs from the 10 chosen bird calls
5) Create a program that takes any bird call input from the 10 chosen bird calls and uses the fingerprinting algo to try and match it to the stored fingerprinted bird call. Essentially the program will be "Is this bird call a (type of bird) call?"
6) ~~Update program so that it will take in any input, not just bird calls~~
7) Create the database with multiple bird calls using different bird call audio inputs from the chosen 10
8) Implement an efficient search algorithm to match a fingerprint to the database
9) Entire project has been completed

Progress of Goals:
Now that I have revised my goals to a more reasonable project, I can work on actually achieving some of them.

Goal 0): Find 10 unique bird call recordings. In order to accurately create a fingerprint for each unique bird call I made some criteria for what I was looking for in a recording:

a) Very little background noise & static. I want the recordings to solely be the bird calls and not a mix of other sounds. Obviously since the recordings cannot be done in a silent room, there will always be some sort of background noise. Because of this I looked for recordings with as little noise as possible
b) Find a mix of unique bird calls that are, for the most part, different but have some recordings that have similarities. This way each fingerprint should be different enough to seem unique but will still show similar sounds. Doing this ensures that I create a fingerprinting algorithm that is precise enough the differentiate similar sounds.
c) Keep the length of recordings under 3 seconds. Since this is only a POC I do not want to have a lot of information in my reference database. I also want to keep the general length of the recordings the same so that they can be more easily compared. If I were doing the entire project with any bird call I would implement some sort of time matching algorithm so that the length of the recording does not matter

In P0 I mentioned I would use the avisoft website to get bird call recordings. I went to their database (http://avisoft.com/sounds.htm) and proceeded to listen to different bird songs/calls. I then used my above criteria to choose 10 different bird calls from 10 different bird species:

Bird	.wav File Name
Blackbird	amsel2.wav
Black Tern	blackter.wav
Blue Tit	blm2.wav
Great Spotted Woodpecker	bsp3.wav
Corncrake	corncrak2.wav
Mountain Chickadee	gambelim.wav
Hedge Sparrow	heckenbr.wav
Northern Mockingbird	mocking.wav
Wood Warbler	waldlaub3.wav
Wryneck	wendehal.wav

* I have included the .wav files in the .zip file handed in for P1

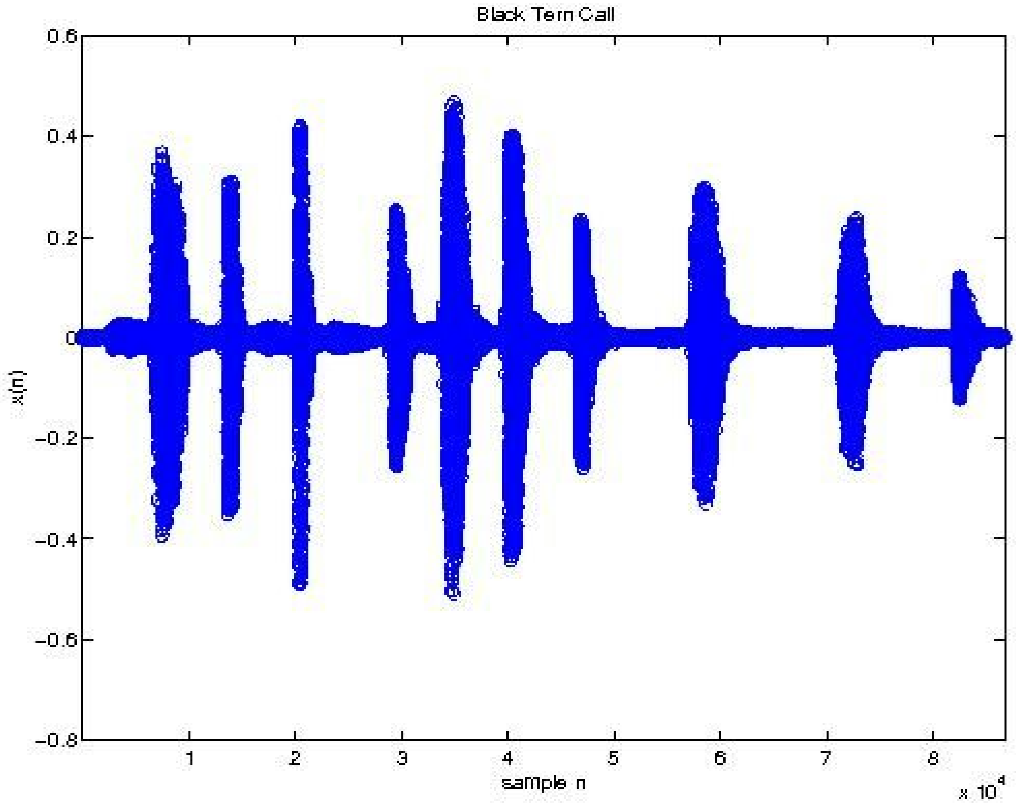
<u>Goal 1)</u> I have decided to follow the general algorithm described in the link I mentioned in P0 (http://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition). However, I will add some tweaks so that it works for my data. However, I figured that before I could implement a basic fingerprint algorithm, I should first analyze the actual sound recordings and do their DFTs. This way I know what I'm working with and the type of DFTs I will be trying to fingerprint/match.

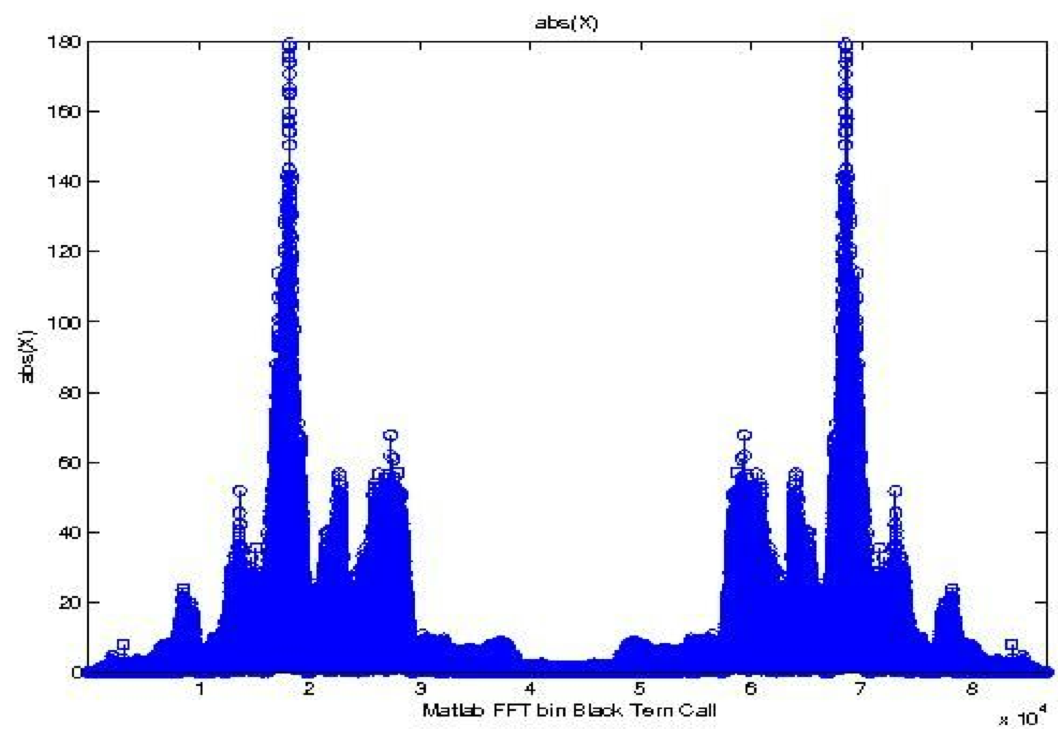
MATLAB Script to generate wave and DFT of recordings:

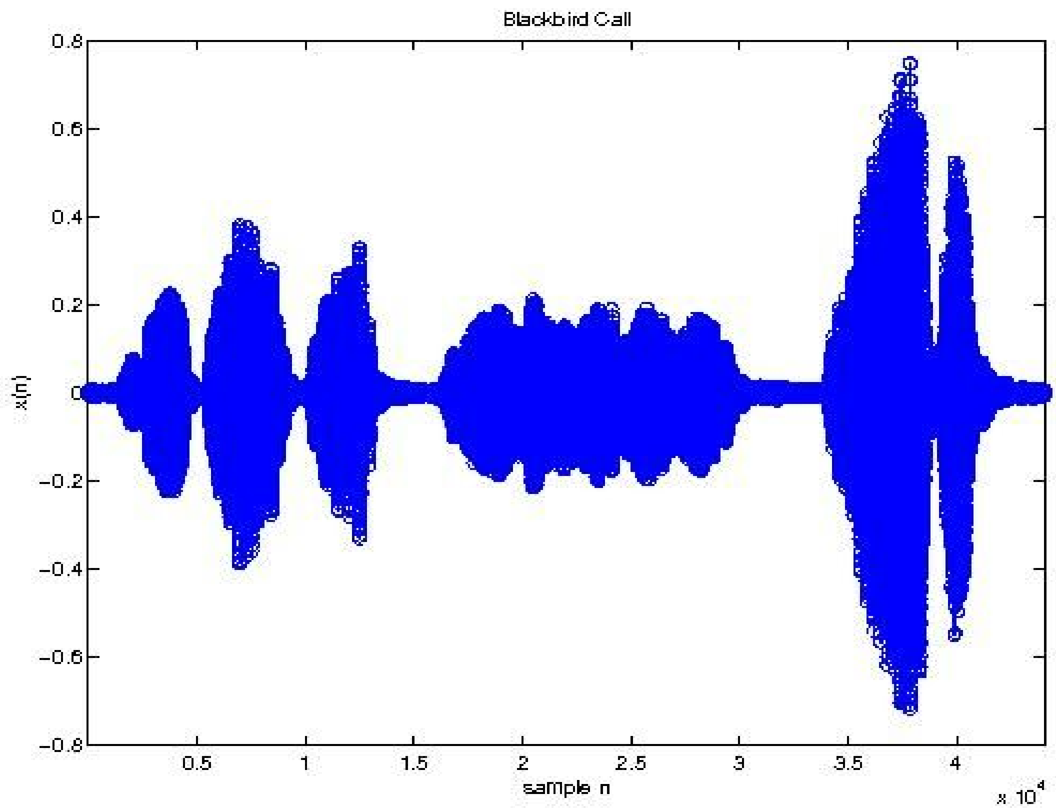
```
info = audioinfo(wav_file);
fs = info.SampleRate
N = info.TotalSamples

audio_signal = audioread(wav_file);

x = audio_signal(1:N);
figure1 = figure;
plot(1:N,x,'-ob');
title(sprintf('%s Call', bird_name));
ylabel('x(n)');
xlabel('sample n');
xlim([1 N]);

saveas(figure1, sprintf('%s Call Wave.jpg', bird_name));

X = fft(x);
figure2 = figure;
stem(1:length(X),abs(X),'-ob');
xlabel(sprintf('Matlab FFT bin %s Call', bird_name));
ylabel('abs(X)');
title(sprintf('abs(X)'));
xlim([1 N]);

saveas(figure2, sprintf('%s Call DFT.jpg', bird_name));
```

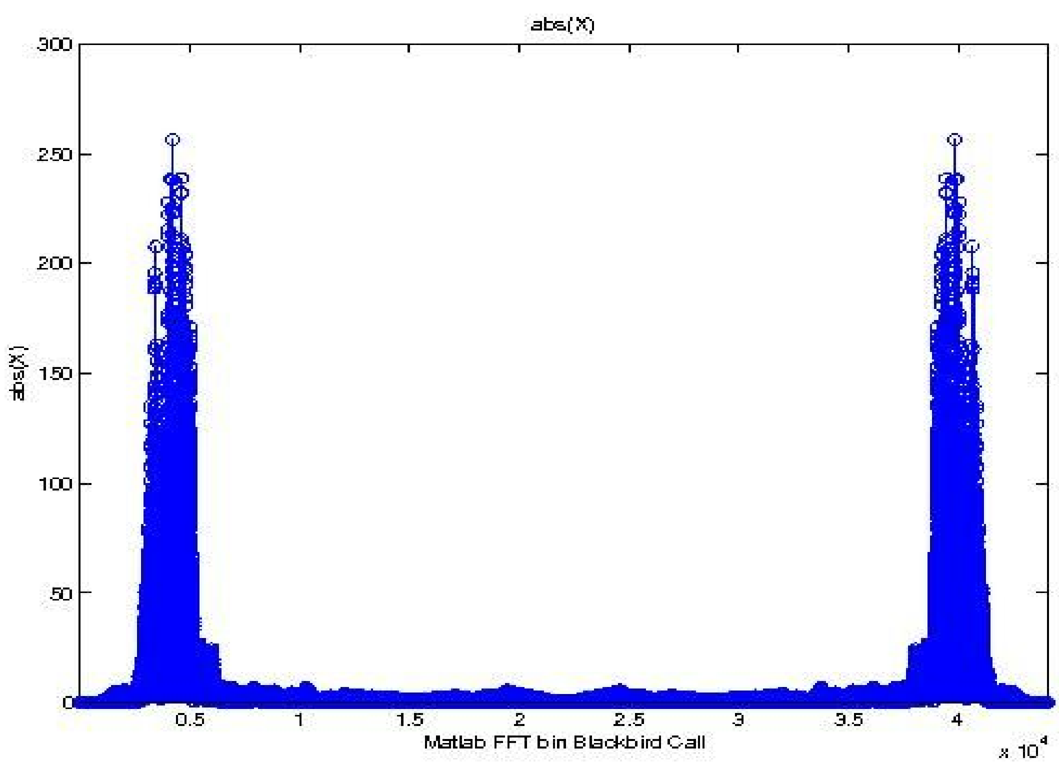
Sample Call in MATLAB:

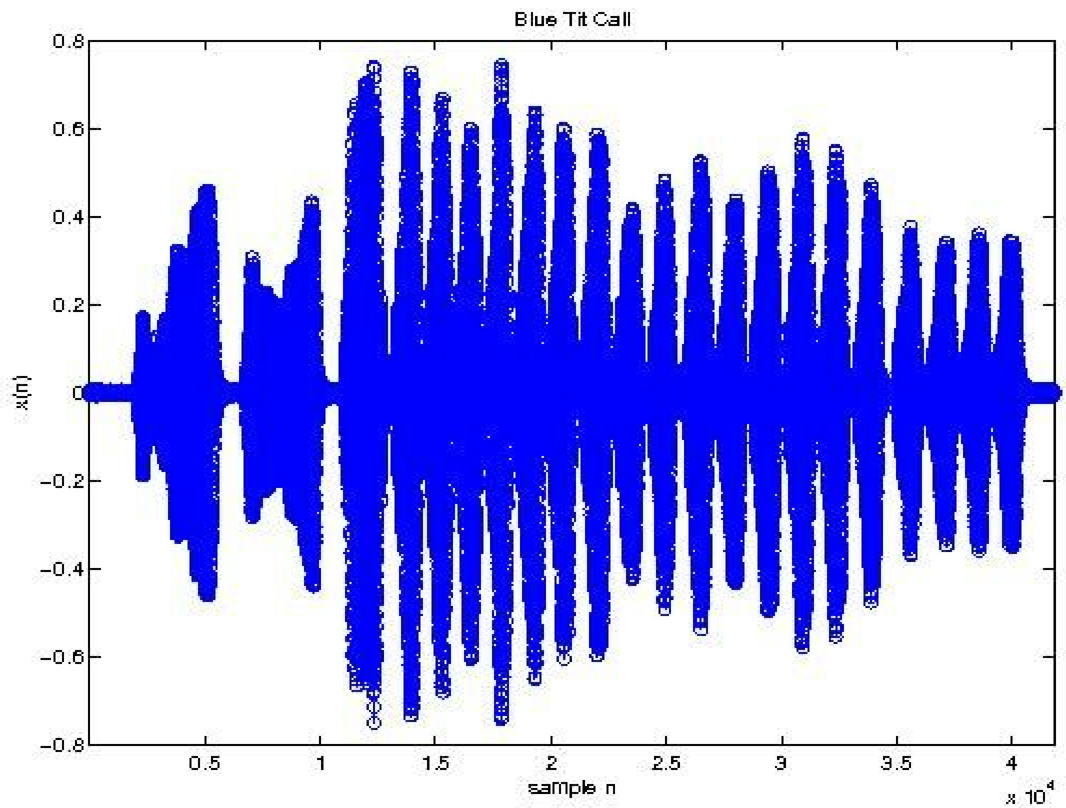
```
clearall;wav_file='wendehal.wav';bird_name='Wryneck';dftsection
```

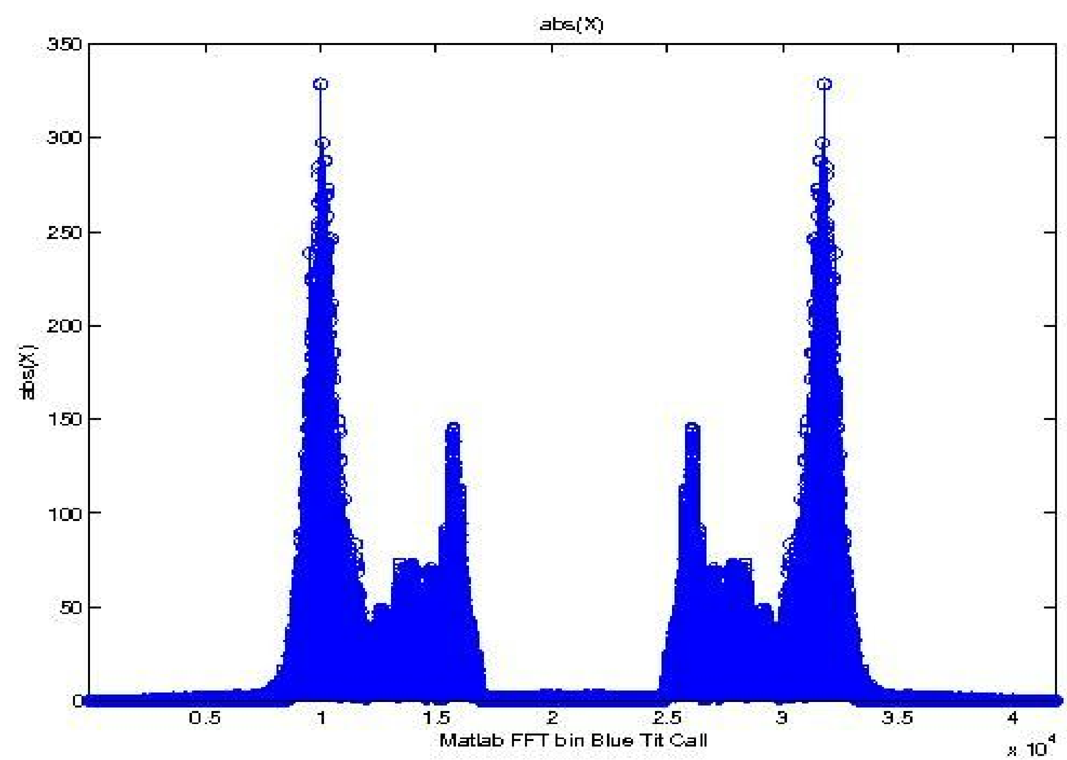

Black Tern Call
x(n)
sample n
x 10^4

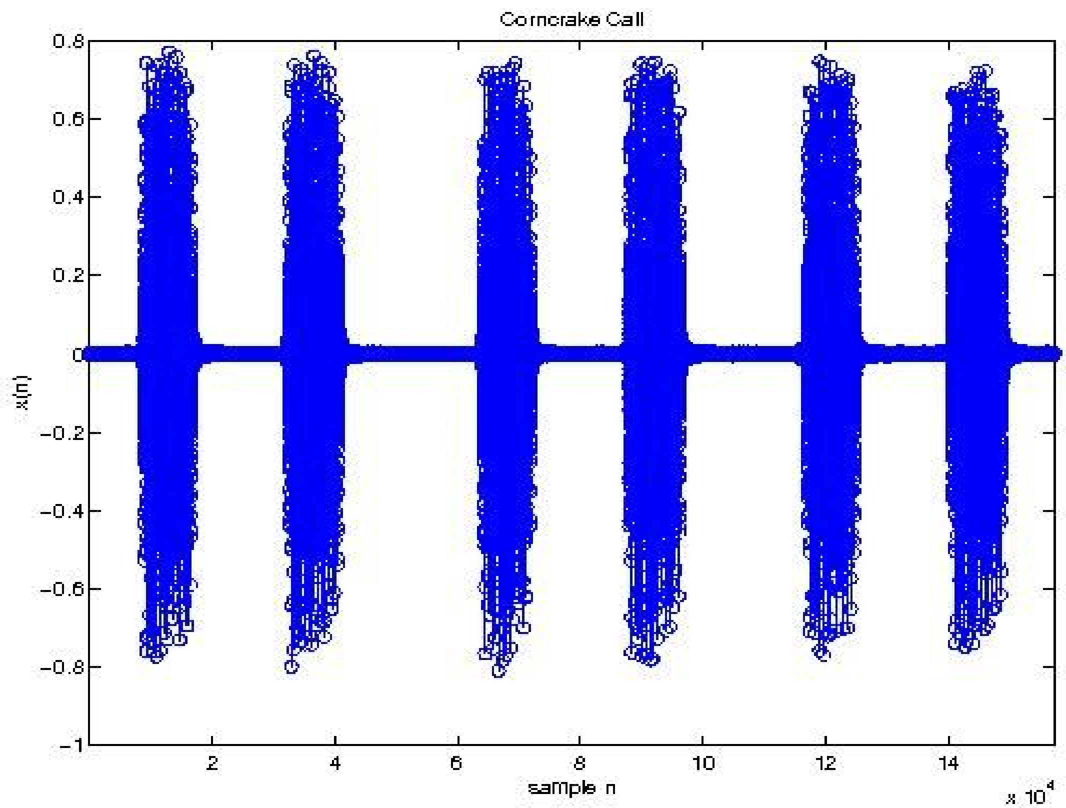

abs(X)
abs(X)
Matlab FFT bin Black Tern Call
x 10^4

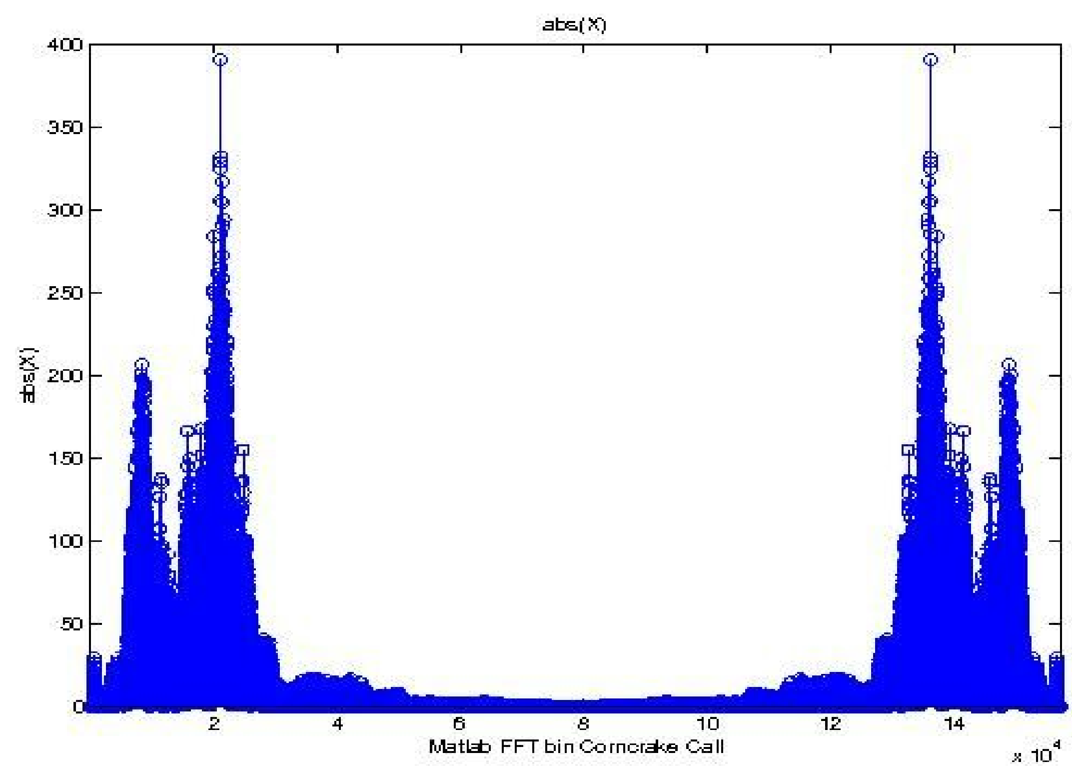

Blackbird Call
x(n)
sample n
x 10^4

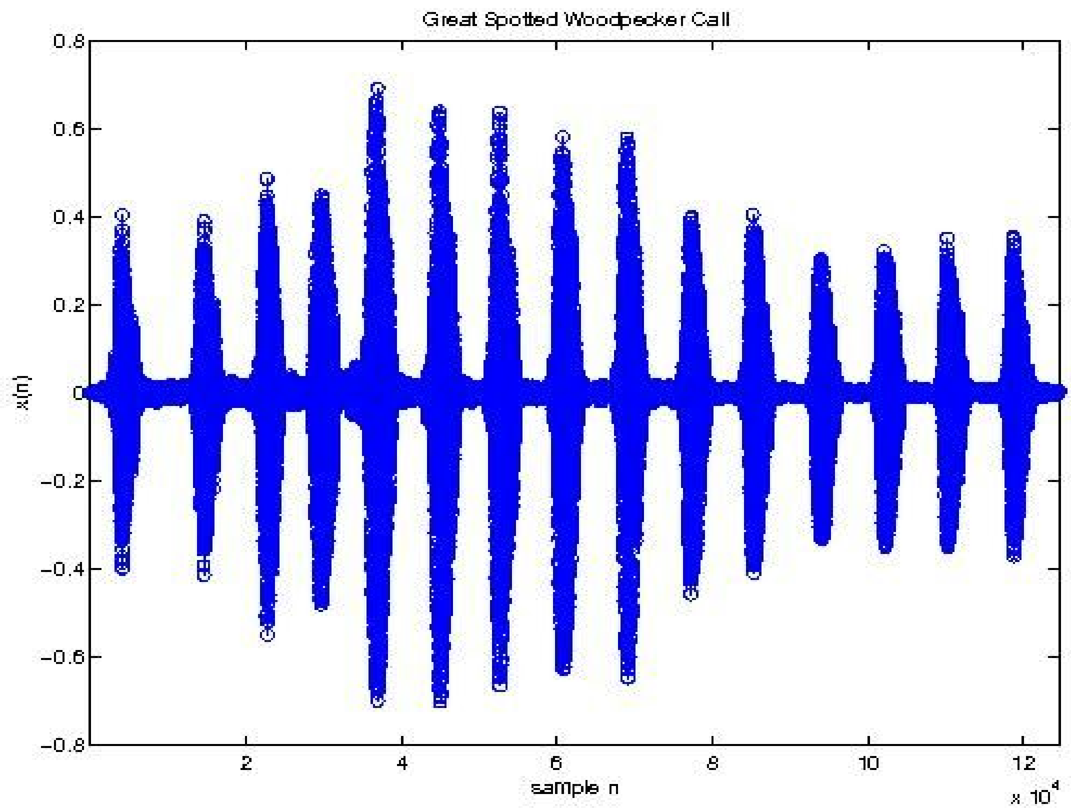

abs(X)
abs(X)
Matlab FFT bin Blackbird Call
x 10^4

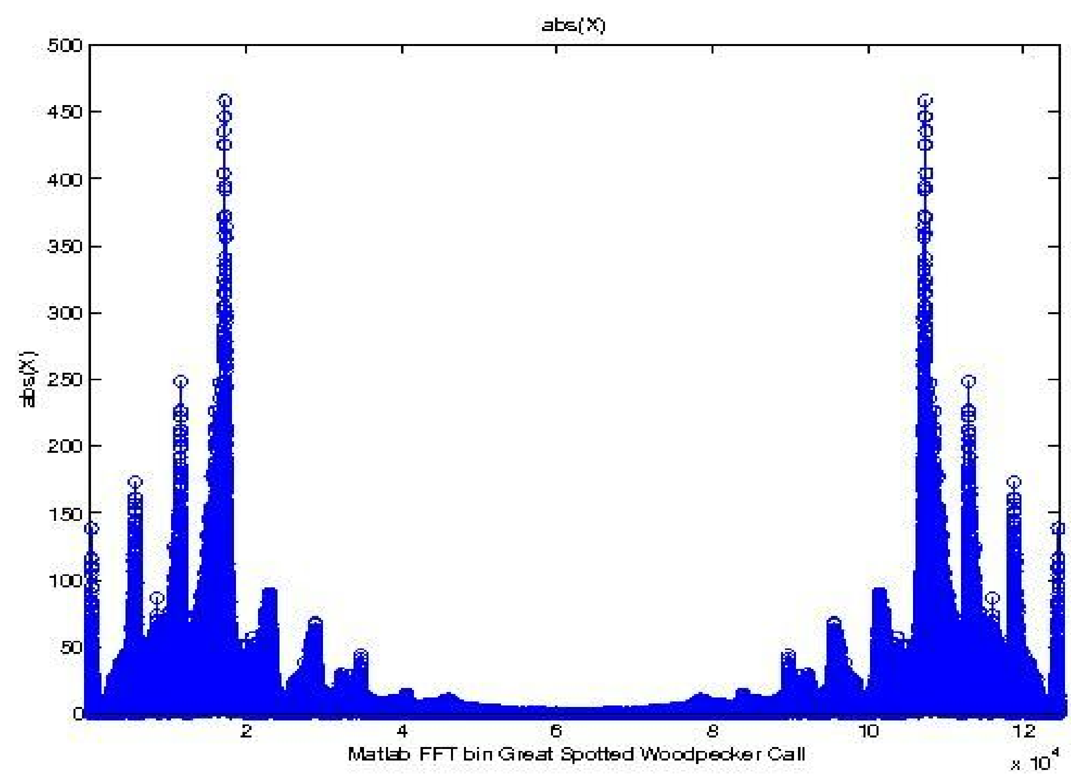

Blue Tit Call
x(n)
sample n
x 10^4

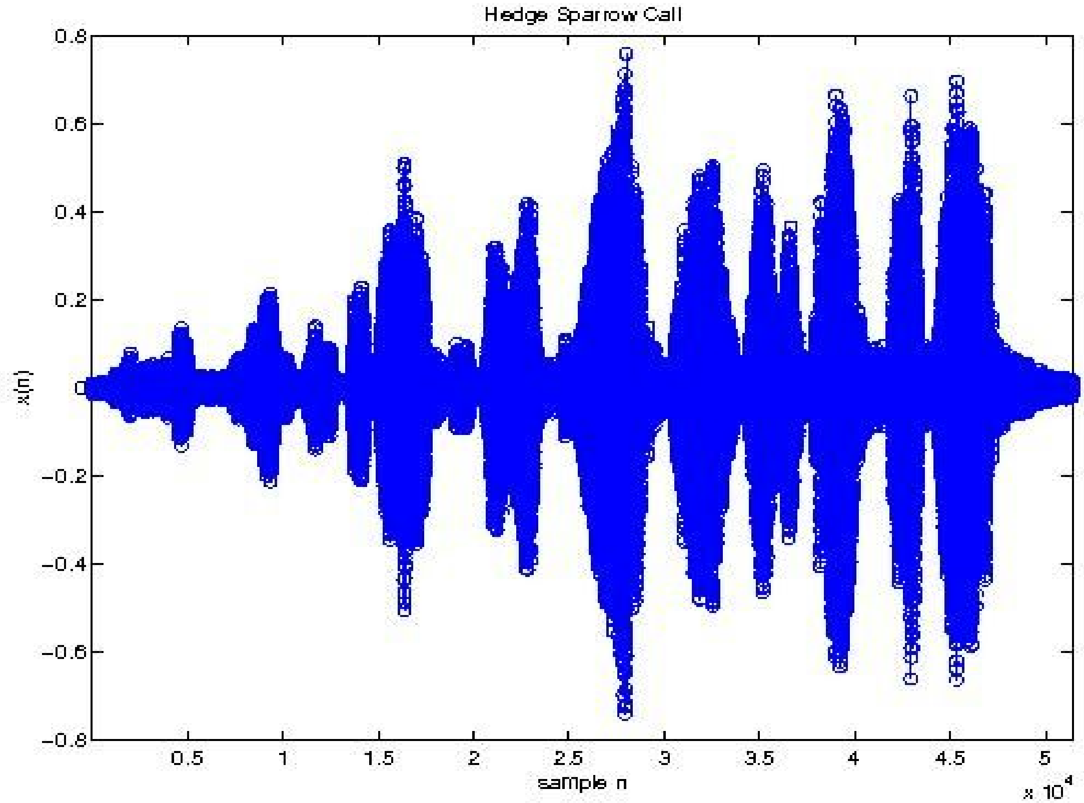

abs(X)
abs(X)
Matlab FFT bin Blue Tit Call
x 10^4

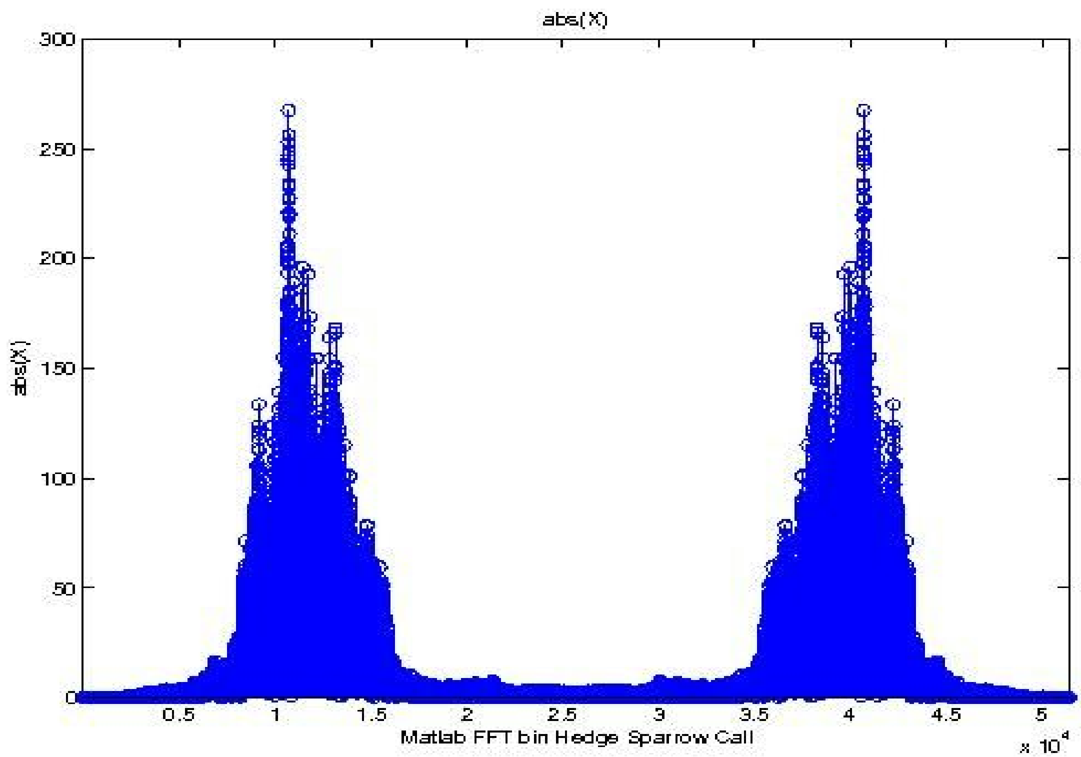

Corncrake Call
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
x(n)
2
4
6
8
10
12
14
sample n
x 10^4

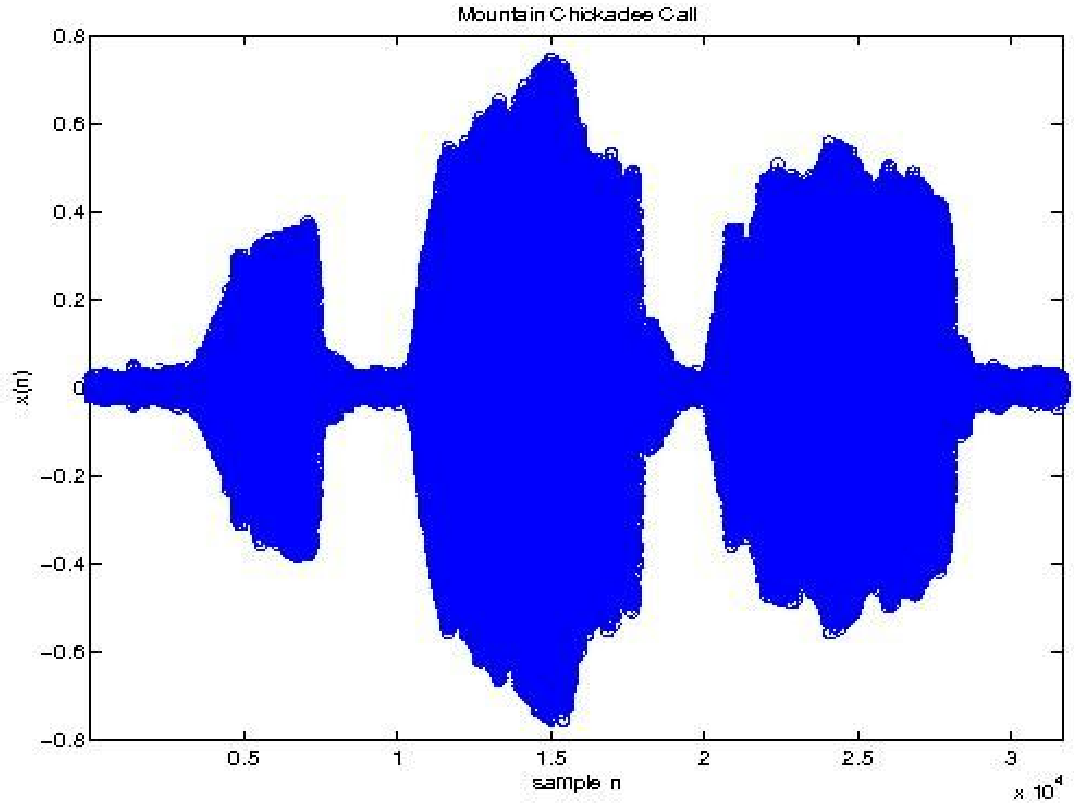

abs(X)
400
350
300
250
200
150
100
50
0
abs(X)
2
4
6
8
10
12
14
Matlab FFT bin Corncrake Call
x 10^4


Great Spotted Woodpecker Call
x(n)
sample n
x 10^4

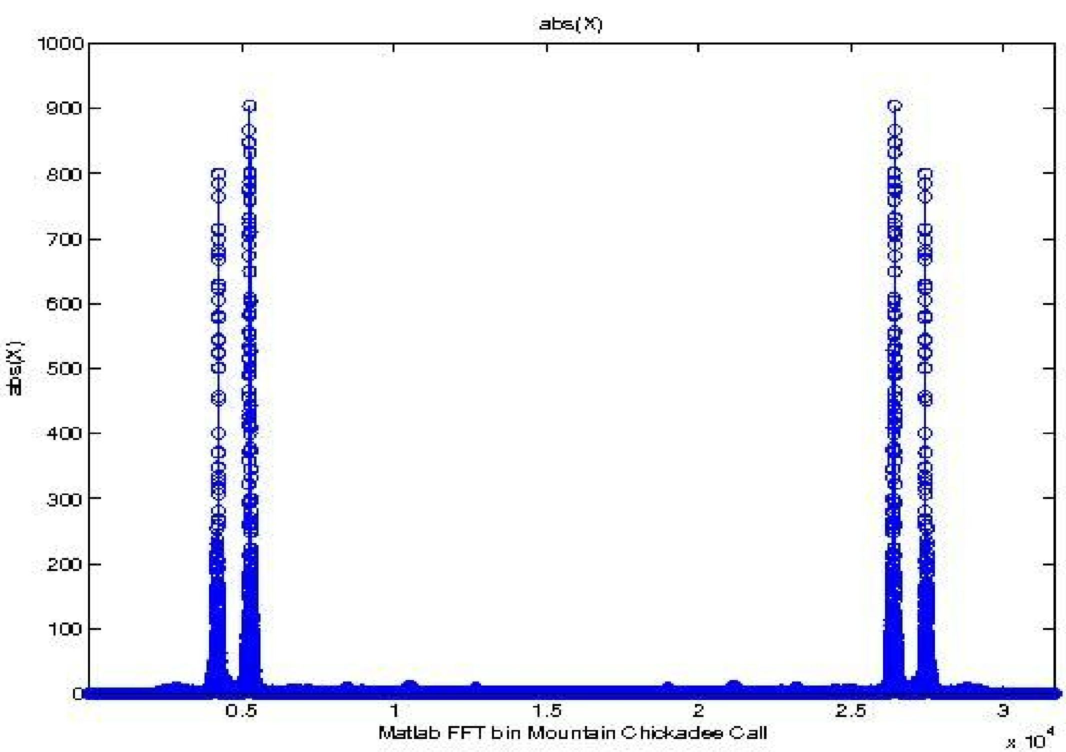

abs(X)
abs(X)
Matlab FFT bin Great Spotted Woodpecker Call
x 10^4


Hedge Sparrow Call
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
x(n)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
sample n
x 10^4

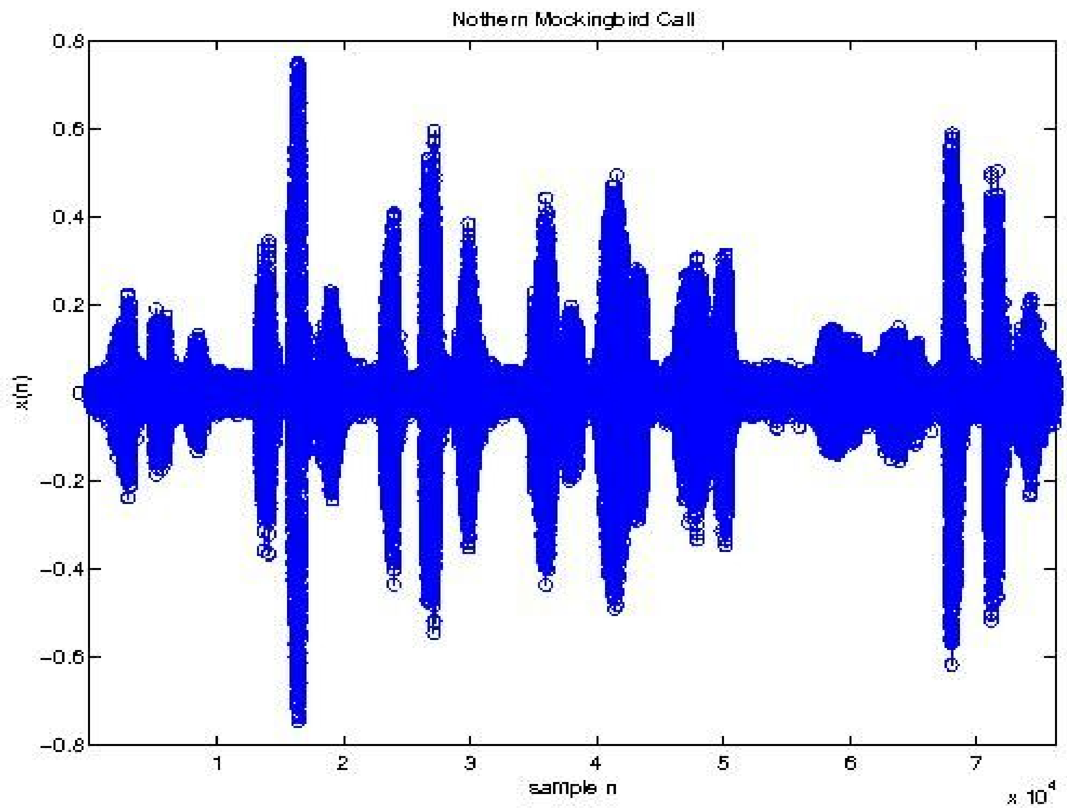

abs(X)
300
250
200
150
100
50
0
abs(X)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Matlab FFT bin Hedge Sparrow Call
x 10^4

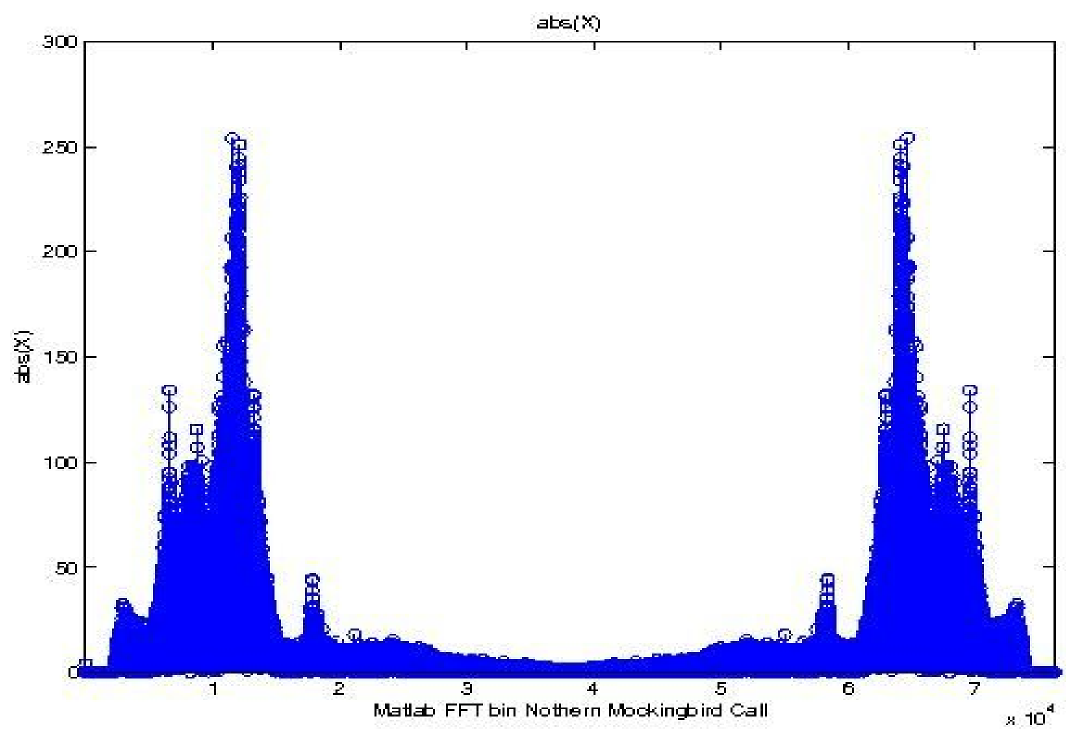


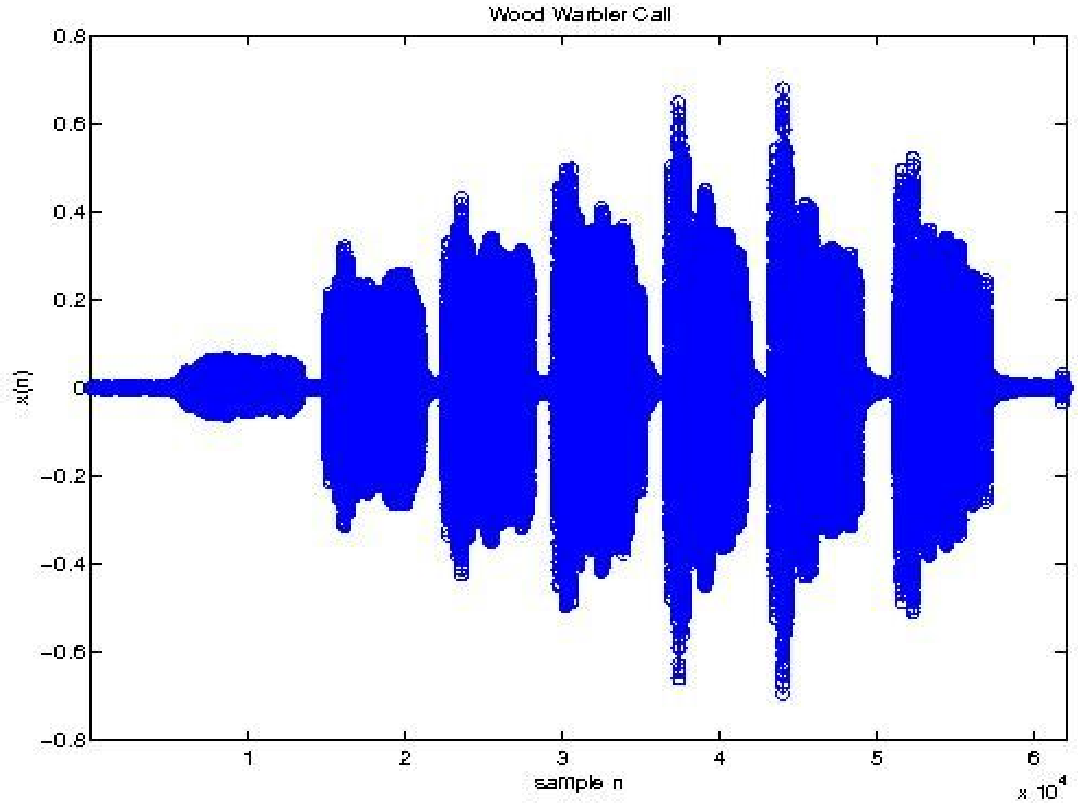
Mountain Chickadee Call
x(n)
sample n
x 10^4

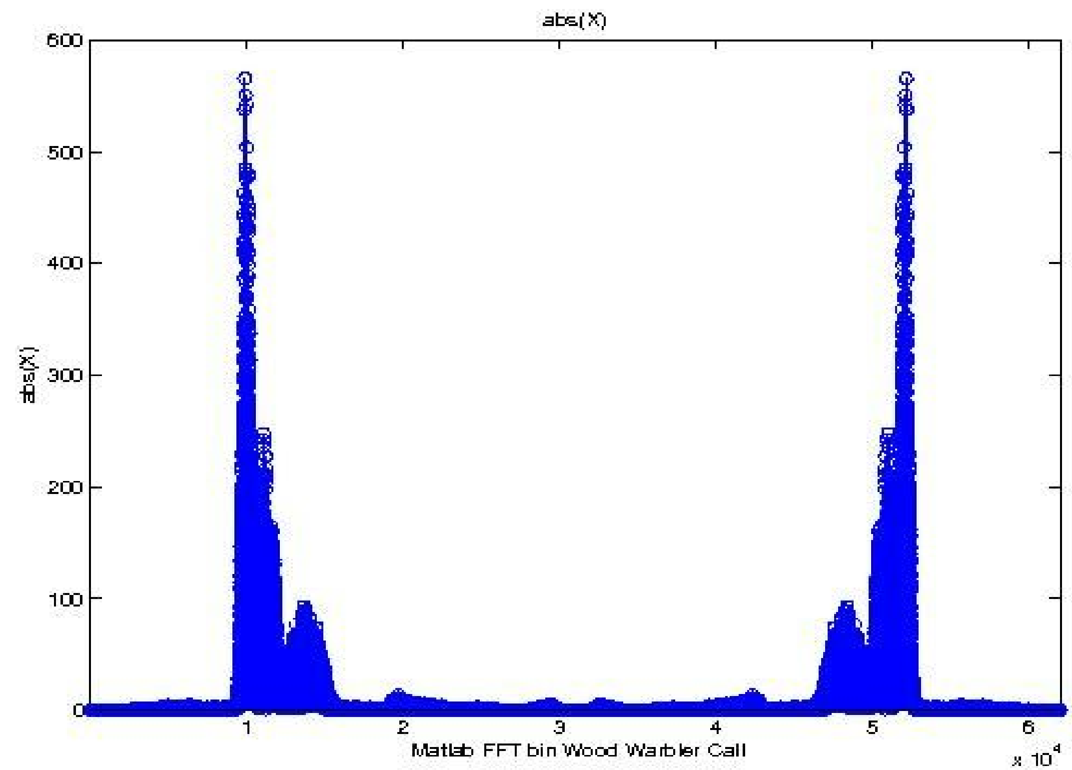


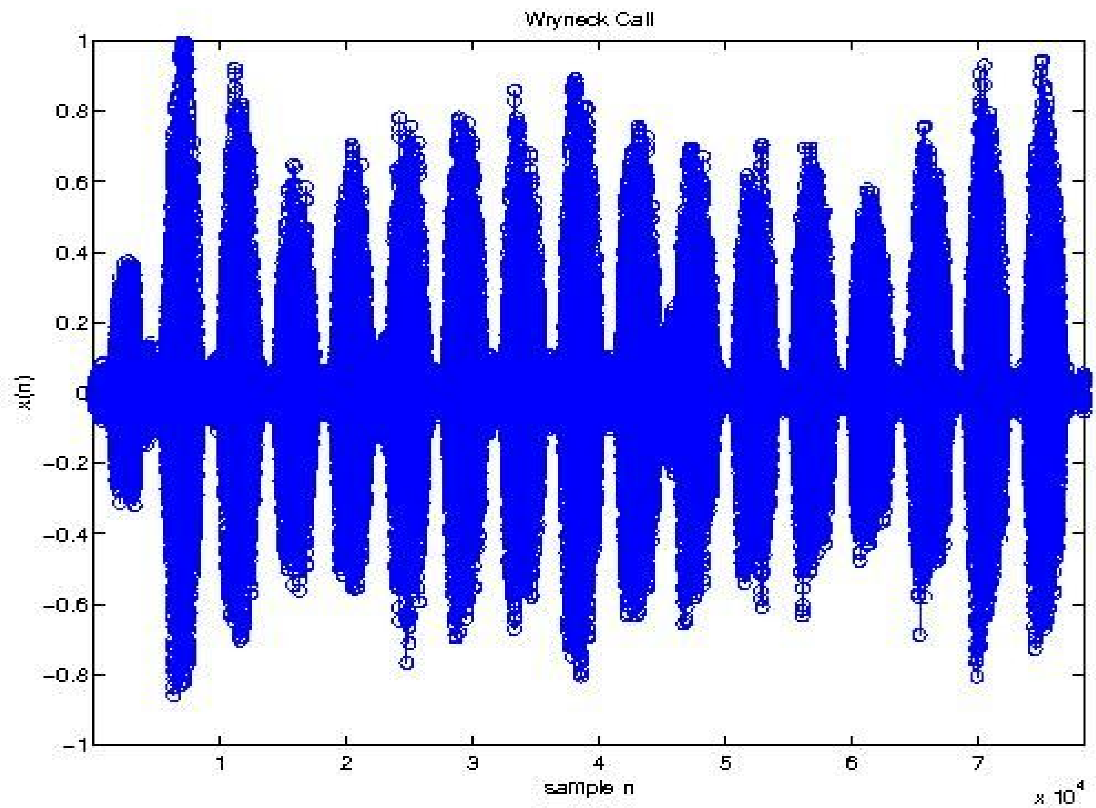
abs(X)
abs(X)
Matlab FFT bin Mountain Chickadee Call
x 10^4

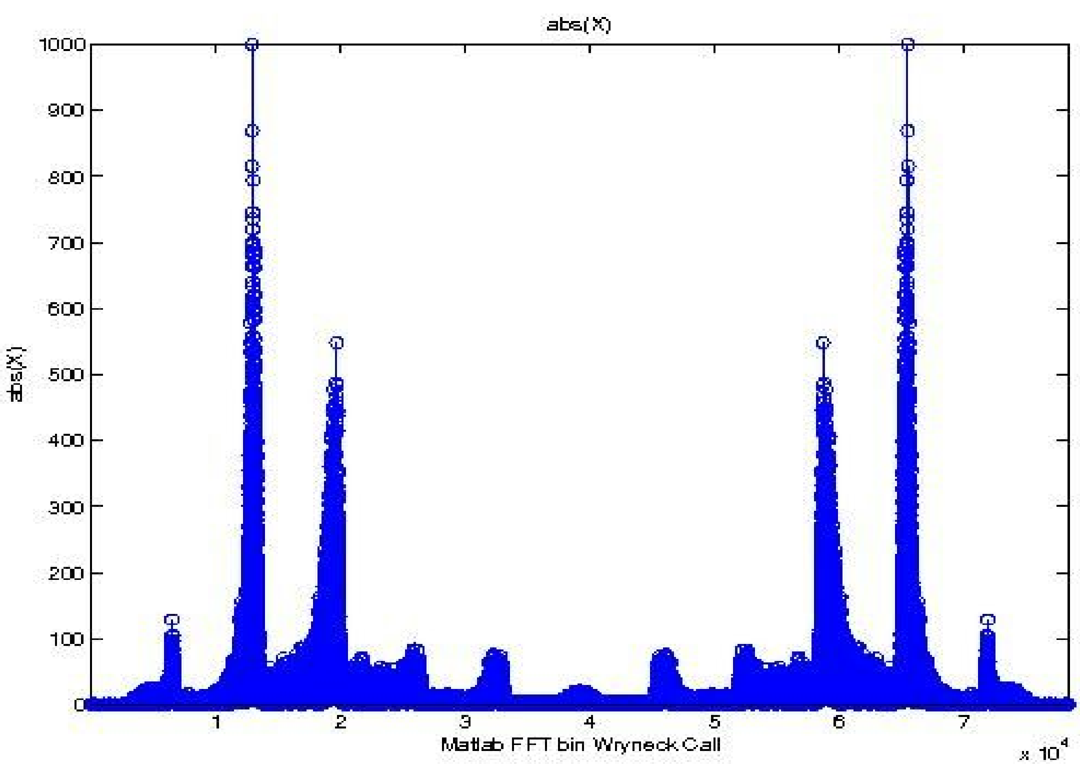

Nothern Mockingbird Call
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
x(n)
1
2
3
4
5
6
7
sample n
x 10^4


abs(X)
300
250
200
150
100
50
0
abs(X)
1
2
3
4
5
6
7
Matlab FFT bin Nothern Mockingbird Call
x 10^4

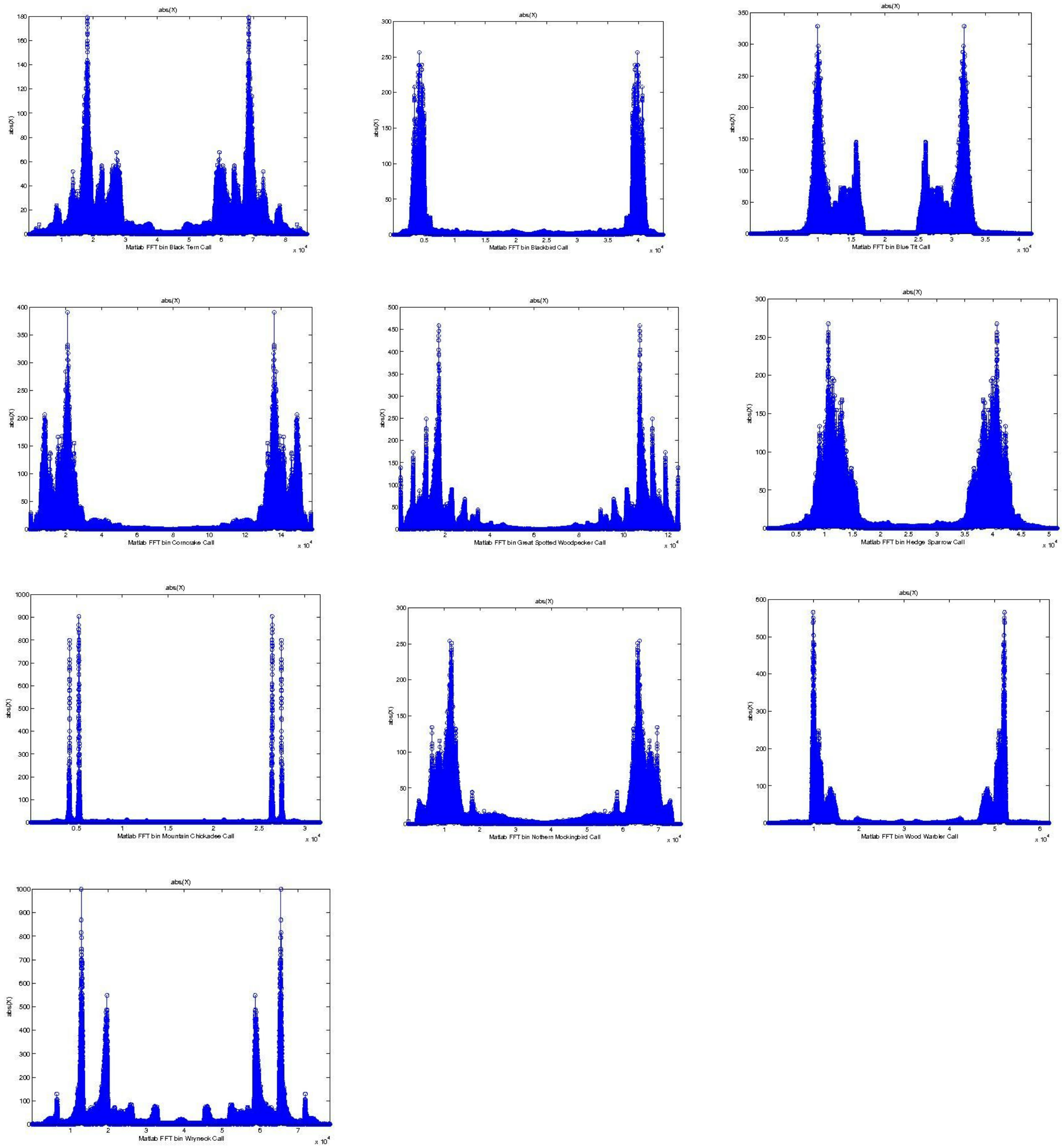

Wood Warbler Call
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
x(n)
1
2
3
4
5
6
sample n
x 10^4


abs(X)
600
500
400
300
200
100
0
abs(X)
1
2
3
4
5
6
Matlab FFT bin Wood Warbler Call
x 10^4


Wryneck Call
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
x(n)
1
2
3
4
5
6
7
sample n
x 10^4


abs(X)
1000
900
800
700
600
500
400
300
200
100
0
abs(X)
1
2
3
4
5
6
7
Matlab FFT bin Wryneck Call
x 10^4

Comparisons of DFTs:



Observations:

- Some DFTs look fairly similar, others look vastly different
- All are unique in at least some way such that one would not be mistaken for the other
- It would be easy to simply compare DFTs to do the fingerprint matching. HOWEVER, as this is a POC, I will try to make a more comprehensive & detailed fingerprint that takes into account more than just the overall DFT

- Very cluttered DFTs which is expected since I have just done one DFT for the entire recording clip. Hopefully, if I do the DFT of sections of the wave as time progresses I should have a more comprehensive DFT that tells me more information

Algorithm for Fingerprinting

a) Read in input audio and convert to data points/samples
b) Split total samples into “chunks” of some chosen size
c) Perform DFT on each chunk of data
d) Determine intervals for strong frequencies
e) Analyze each chunk’s DFT and record the highest frequency in each interval
f) Create a hash value for each chunk and create entry in database:
    i) Entry will look like: <hash value> <bird name>
    ii) Note there will multiple hashes for the same bird recording

Algorithm for matching

a) Do the same algorithm for fingerprinting up to step e)
b) Create hash for each chunk of the input recording
c) For each chunk, go through the database and try to match the chunk hash to other chunk hashes
    i) They chunk may match to more than one chunk. This is ok.
    ii) For each chunk match, keep a tally of what bird corresponds to each match
d) After all chunks have been computed and matched, find the bird with the most chunk matches. This will be the most likely bird that matches the input bird.

“After Project Completion” Ideas:

As I have been working on this project I have come up with ideas on some additional things I could do one the project is complete. They are:

- Experiment with sample chunk size: larger chunk size == less precise fingerprint VS smaller chunk size == extra precision not needed
- Once done project, try inputting some other bird calls that are not in the database and possibly see which bird they match closest too. This would be a good “reflection” activity to see how close my algorithm can two different recordings of the same bird species.