Measuring Room Response and Distance with White Noise

Austin Chase Minor

December 1, 2016

1 Introduction

In this paper, we will test the use of white noise for distance measurement and room response. This is a practical application of discrete autocorrelation and the FFT (Fast Fourier Transform). We will start by giving a mathematical description of the concepts related to the problem. Next, we will give the problem statement. From there, we will analyze the results and give our conclusion.

2 Mathematical Description

We know that autocorrelation is a description of how correlated a signal is with it self in time. This is useful for detecting repetitive responses in signals whether by nature of the signal or noise, such as echoes. White noise, by definition, is a constant power signal in the frequency domain. This implies that it is only correlated at the origin. We will be using a band-limited white noise to approximate white noise.

We know that the spectral density is the Fourier Transform of the autocorrelation. Since we will be working with a signal in the discrete domain, we will use the FFT. This is an fast running algorithm to compute the discrete Fourier Transform. From this, we will be able to see what frequencies the room attenuates or accentuates (cuts or boosts) by observing the frequency response since white noise has even power across the spectral density.

The recorded audio is in 16-bit wav format, sampled at 44 kHz. This means that for every second of audio their is 44,000 16-bit samples. Thus if the autocorrelation peaks at 400 tau then that is $\frac{400}{44000} = 9.1$ milliseconds past the origin. The speed of sound in air is 1125.33 ft/sec. Thus in the previous example, the distance from the source of the echo is $.0091 * 1125.33 = 10.24$ ft. Furthermore, since this is a measurement of an echo, the distance should be divided in half. This is because the echo is a reflection off the wall and thus is first picked up by the microphone from the source, travels Δx to the wall and Δx back to the microphone.

3 Problem Description

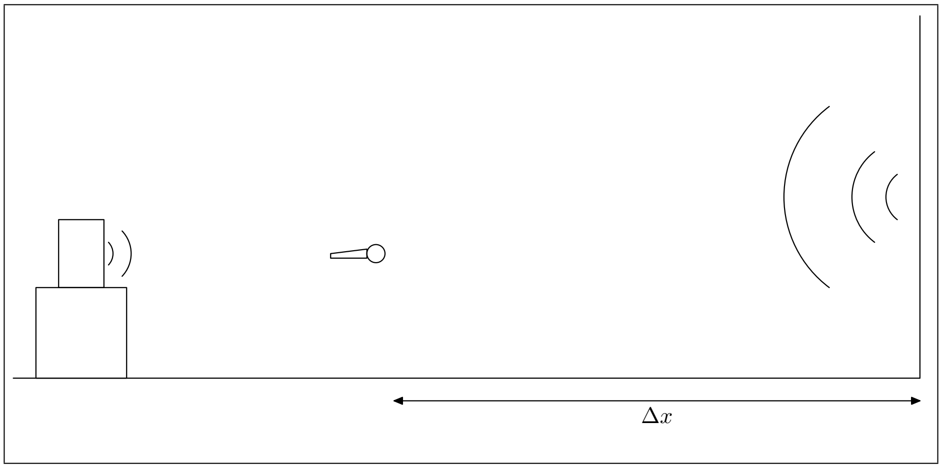

Figure 1: Experiment Setup

The basic setup is as pictured above. We will be using a speaker to broadcast the white noise, a microphone to pick up the transmitted signal, and a wall to measure the distance from. Furthermore, the speaker is a reference monitor which implies that the white noise that it broadcasts will have an approximate flat frequency response. This is the same with the microphone. As in the diagram above, the microphone is facing the back wall and is roughly the same height as the speaker cone in the speaker. The microphone has a hyper-cardioid pickup pattern. This means it will pickup noise in the front and back but not sides. Thus we will be able to pickup the original white noise and the echo off the back wall only. This is of course only an approximation since the microphone merely attenuates signals from other directions. However, the attenuation is large enough to approximate it as described.

For the measurement of distance to the wall, we set the microphone an x-amount of distance from the back wall. For both of the trials this was 8' and 9'9" respectively. Then using a band-limited white noise signal, generated using the audio program Audacity, we recorded audio from the speaker and the reflection off the back wall. Then using Matlab we calculated the autocorrelation. Only the first 1000 entries of the autocorrelation were considered in our code. This is so you can see the echo on the autocorrelation. The code for this is below.

The spectral density measurement involved the same setup as above and is included in the code below. The spectral density calculation is carried out as described above.

For measuring the spectral density a utility program was written to compute the partition of a vector into a smaller vector using an average for each partition.

```
%https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie
%=UTF-8#q=speed+of+sound+in+feet+per+second
SPEED_OF_SOUND = 1125.33;

%for matlab and not octave use audioread
audio_8 = wavread('8ft.wav');
audio_9 = wavread('xft_session.wav');
%reduce to one channel
audio_8 = audio_8(:, 1);
audio_9 = audio_9(:, 1);

%correlation and positive lag positions
[cor_8,lags_8] = xcorr(audio_8, 'biased');
[cor_9,lags_9] = xcorr(audio_9, 'biased');
p_lags_8 = find(lags_8 >= 0);
p_lags_9 = find(lags_9 >= 0);

%22kHz of frequency band
%chop upper part of spectrum
%since spectral density is an even function
spec_8 = fft(cor_8, 44000);
spec_8 = spec_8(22001:44000);
spec_9 = fft(cor_9, 44000);
spec_9 = spec_9(22001:44000);

%https://www.mathworks.com/matlabcentral/answers/
%92565-how-do-i-control-axis-tick-labels-limits-and-axes-tick-locations
%Thanks to matlab support team for on this article for
%suggesting how to fix the x axis tick labels
figure
plot(1:1000, cor_8(p_lags_8(1:1000)))
xlabel('tau')
ylabel('autocorrelation')
set(gca, 'XTick', 1:50:1000)

figure
plot(1:1000, cor_9(p_lags_9(1:1000)))
xlabel('tau')
ylabel('autocorrelation')
set(gca, 'XTick', 1:50:1000)

%after checking the Matlab documentation and examples
%not sure if the magnitude is off, however that
%will not affect the interpretation of the room response
%since it is off only by a constant scale factor across all values
%and thus the shape of the graph will be the same

figure
bar(.1:22.000/100:22.000, partition(abs(spec_8)-mean(abs(spec_8)),100))
xlabel('kHz')
ylabel('Spectral Density')

figure
bar(.1:22.000/100:22.000, partition(abs(spec_9)-mean(abs(spec_9)),100))
xlabel('kHz')
ylabel('Spectral Density')
```

```
function y = partition(vector, num_bins)
  y = zeros(1, num_bins);

  if size(vector,1) > size(vector,2)
    sz = size(vector,1);
  else
    sz = size(vector,2);
  end

  p_sz = ceil(sz/num_bins);
  r_sz = p_sz;
  for i = 1:num_bins
    for j = 1:p_sz
      if i*p_sz + j > sz
        r_sz = sz - i*p_sz +j;
        break;
      end
      y(i) = y(i) + vector(i*p_sz + j);
    end
  end

  for i = 1:num_bins
    if i == num_bins
      y(i) = y(i)/r_sz;
    else
      y(i) = y(i)/p_sz;
    end
  end
end
```

Listing 2: Function to Parition Array using Average Scheme

4 Analysis

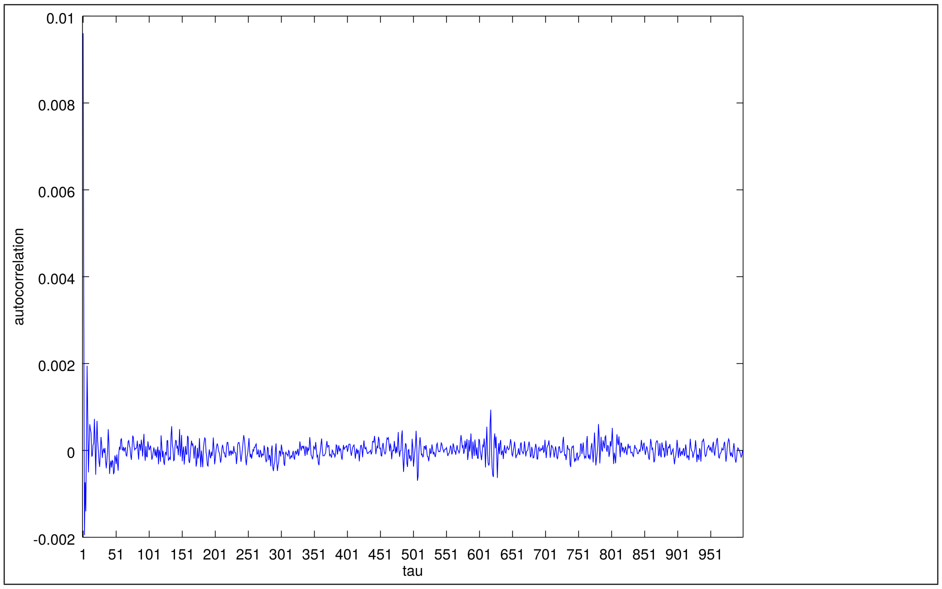

Figure 2: Correlation of 8' signal

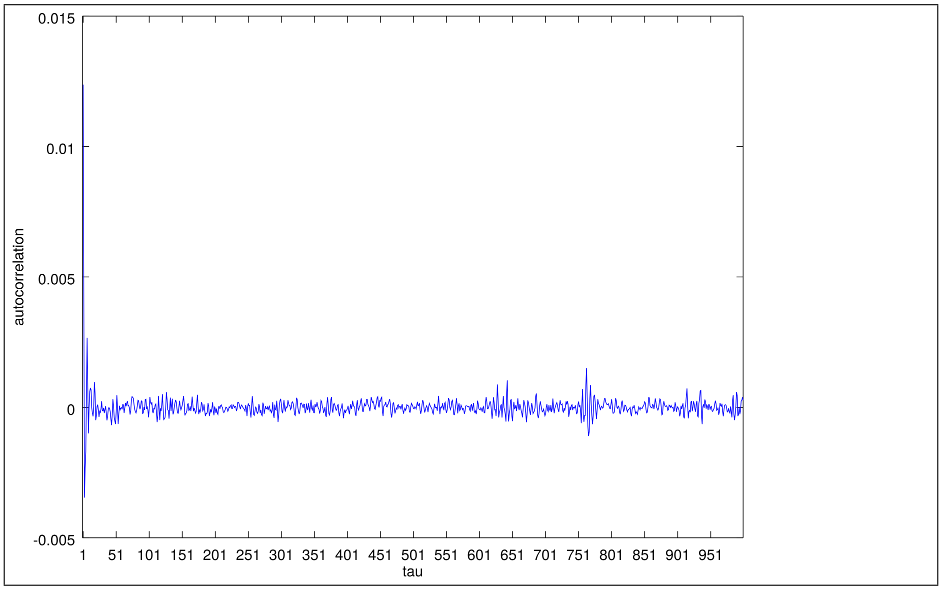

Figure 3: Correlation of 9'9" signal

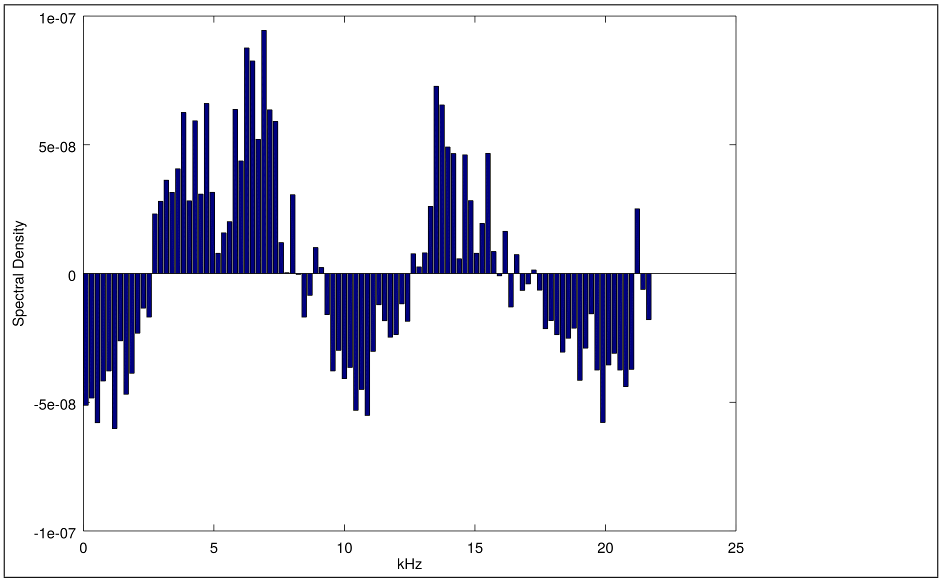

Figure 4: Mean subtracted Spectogram of 8' signal

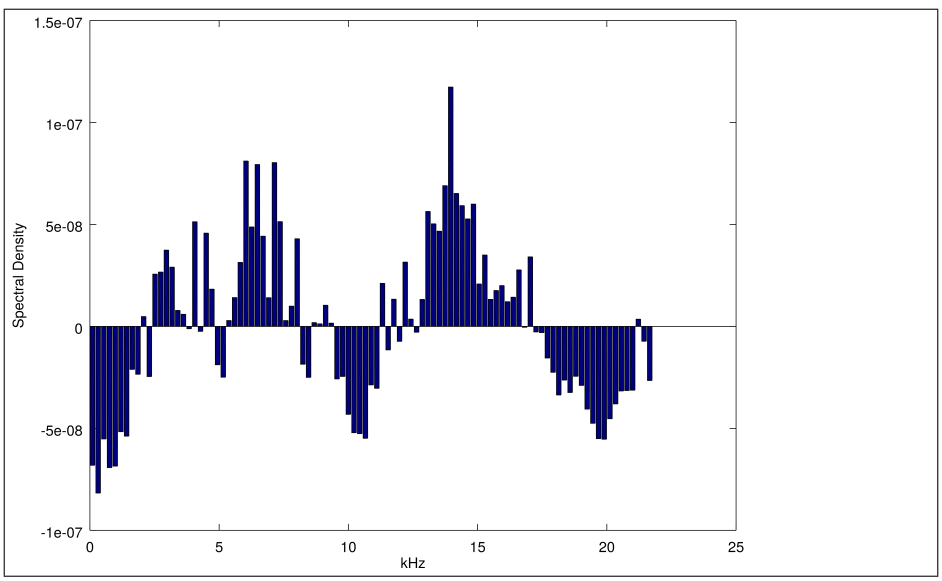

Figure 5: Mean subtracted Spectogram of 9'9" signal

Pictured above is the result of running the Matlab code. As noticeable their is an echo in the autocorrelation of both signals at 620,762 for the 8', 9'9" signal respectively. These numbers were approximated from their respective graphs. Computing the distance as described earlier yields the following information:

Signal	Approximately Max Correlation	Theoretical Distance
8'	620 (tau)	7.93'
9'9"	762 (tau)	9.74' = 9'8.9"

Thus compared to the original distances, the calculated distance differed only by less than 1% for both signals, using the midpoint method. Thus the method by which we measured the distance to the wall is an accurate method for the setup we had. It is noticeable that their is some blips in both signals; however, they are not the size of the main blip. Also, we assumed the distance of the signal to consider. This is not unreasonable since it may be assumed that you can guess two wide bounds to search between. For example, a submarine may only want to know the objects within 100 meters of itself.

For the spectrograms the mean was subtracted from both and the signal partitioned. This was to normalize and smooth the signal. The mean was subtracted to determine if the frequencies were attenuated or accentuated. Examining the graphs above shows that in both cases the bass and mid-bass (0 - 2.5 kHz) was attenuated. The mids (2.5 - 9 kHz) was accentuated with a neutral zone The only neutral zones are around the transition points except in the mid zone and the low high-mids. This shows that the room was not neutral in terms of frequency response. around 5 kHz. The same procedure can be done for the high-mids and highs. It may be noted, as noted in the code, that the exact amplitude of the spectrogram may not be accurate; however, for the measurement of the 'flatness' of a room this obviously is not as important.

5 Conclusion

Thus from the experiment we have determined that white noise is a potential signal to use for the use of distance measurement. This signal lead to an accurate distance measurement to within a 1% of error. Secondly, we showed the overall room response was tainted. The only neutral tones was in small areas of the frequency band. It was interesting that the frequency response followed a cosine wave pattern. Overall, the use of band-limited white noise was a success in two diverse, but related, areas of signal analysis.