Ross Penniman
PAT 407
Final Report

Room Acoustics and Dynamic Range

Introduction

Anyone who spends a considerable amount of time talking to sound engineers or reading about sound recording will inevitably hear the term "room compression" being tossed around. The term refers to the perceived compression of dynamic range whenever a particularly loud instrument or ensemble is placed in a small, reverberant room. With as much as the term gets used, however, it is very difficult to find any formal definition or technical discussion of the phenomenon. The purpose of this study was to find out whether such a phenomenon can be objectively documented and if so, to find the extent to which "room compression" occurs with relation to the method of recording and the acoustic properties of the room.

Background

Knowledge from a variety of disciplines was needed in order to conduct this study. First, one had to understand the propagation, reflection and absorption of sound in a room. This was crucial for understanding how the sound field, which is picked up by the microphone, can change depending on the properties of the room and the locations of sound source and microphone. Second, one had to understand how to measure the level of a signal once it was recorded. Since the sound waves which make up music are both very complex and very dynamic, measuring the amplitude of the sound wave at any moment in time can prove to be very difficult. It was necessary understand the signal processing tools available to measure the changes in an audio signal. Finally, one had to have (at least) a basic understanding of psychoacoustics, of how the objective magnitude of sound level relates to the subjective experience of loudness. The ear-brain system is both complex and non-linear. Understanding this system provided insight as to whether the observed data was significant to the perception of loudness and dynamic range.

<u>Room Acoustics</u>

In studying room acoustics and the properties of reverberation, three important values emerged: the reverberation time, the strength factor and the reverberation radius. The reverberation time (abbreviated RT60) is the time it takes sound to decay 60 dB below its original level once the source has been shut off. The strength factor is the amount of amplification a room provides to the original source. The reverberation radius (also referred to as critical distance) is the distance at which the direct sound from a sound source and the diffuse sound field are equal in intensity. At distances less than the reverberation radius, the direct sound will predominate and at distances greater than the reverberation radius, the diffuse sound field will predominate. There are several ways in which these properties could produce a compression effect. The reverb time and strength factor could provide compression by sustaining and amplifying sounds. Reverberation might spread the total sound energy out over a longer span of time, thus reducing the moment to moment changes in level. The relationship of the microphone placement to the reverb radius could also show compression of the audio signal. The direct sound will most likely contain the dynamics of an instrument exactly as they are produced. In the diffuse sound field, any short, percussive attacks will be smeared in time and will lose their impact. Since the position of the microphone affects the ratio of direct and diffuse sound, it could play into the amount of compression which is observed.

Signal Processing

In the most simple definition, the level (which is closely related to loudness) of a sound wave is determined by its amplitude. Since auditory sound covers such a wide frequency range, however, it often becomes difficult to determine the difference between the envelope of a sound and a low frequency oscillation. In addition, depending of the purpose of the measurement, a shorter or longer integration time will be needed. Short integration times (often referred to as peak metering) are more technically accurate, but can have problems with low frequency oscillation. Long integration times (often referred to as RMS metering) are closer to the way in which sound is perceived (as will be discussed below), but may mask tell tale data for this particular purpose.

The basic algorithm for measuring the level of a signal is simple. The first step is to rectify the signal (same as taking absolute value). The second step is to apply some sort of smoothing filter (which is where many variables can be introduced). The final step is to convert the output to a logarithmic scale (i.e. decibels). The smoothing filters used for this study used differing values for attack and release. This means that one time constant was used for upward changes in level and a different time constant was used for downward changes in level. In addition, the peak meter employed a special shape to the release phase such that it would release very slowly at first and then transition into a very fast release. This allowed the meter to both attack and release quickly, while at the same time oscillating very little in the presence of low frequencies.

Psychoacoustics

Research has shown that there are many variables at play in a person's perception of loudness. These include frequency, absolute SPL, spectral content, and duration. There are two pieces of information which are most relevant, however. The first is that humans are able to hear variations in level as little as 0.5 dB or less depending on the circumstances (Zwicker & Fastl, 1990). The second is that sound durations shorter than 200 ms will be perceived as being softer even when they have the same level. This is most clearly indicated in one of Zwicker's graphs (see Figure 1, note that the sone is a unit of loudness in which 4 sones sounds twice as loud as 2 sones). Further indication that the ear has some sort of integration time for perceiving sounds is that a sound of short duration can be masked by sound following it (see Figure 2).

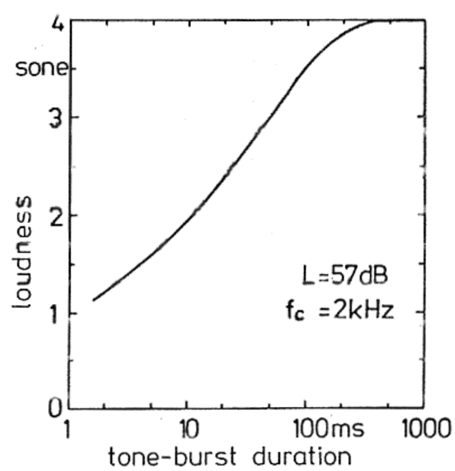


Fig. 1 Loudness of a burst extracted from a 2 kHz tone with an SPL of 57 dB as a function of tone-burst duration. (from Zicker & Fastl)

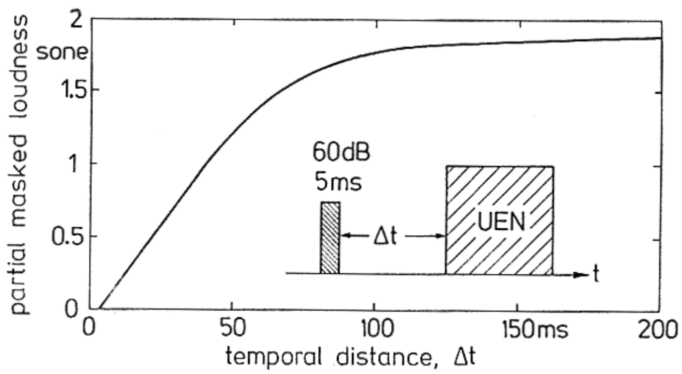


Fig. 2 Temporal partially masked loudness of a 5 ms, 60 dB, 2 kHz tone that is presented before the onset of a uniform-exciting noise as a function of the temporal distance indicated in the inset.
(from Zwicker and Fastl)

These pieces of information indicate that small differences in dynamic range are important, but one must not overvalue short peaks of sound as they may not contribute significantly to the perceived loudness.

Hypothesis

The hypothesis was that the by placing a microphone further from the sound source with respect to the reverberation radius, both upward and downward compression would occur. Upward compression (amplification of soft sounds) would occur because of the reverberation time and strength factor. Downward compression (attenuation of loud sounds) would occur because the direct sound would lose its intensity very quickly.

Method

Data was gathered in three different spaces, a classroom, a recital hall, and a large auditorium. For each space, one microphone was placed near the sound source and set to a cardioid pattern and a second was placed on the opposite side of the room and set to an omni-directional pattern. The objective was for the close mic to pick up as direct a sound field as possible and the distant mic was to pick up as diffuse a sound field as possible. The sound sources were a drum kit in the classroom, a solo trumpet in the recital hall, and a large wind ensemble in the auditorium. These sources were chosen because they have a large dynamic range and can easily overcome ambient noise (such as air handling).

In each case, a short excerpt of music was recorded which demonstrated the dynamic range of the instrument or ensemble. In the case of the trumpet player in the recital hall, the trumpet player was pointed at the back wall of the stage, such that all the sound reaching the distant microphone was reflected sound. In each case, the recording of the close microphone was run through a digital reverberation processor with parameters chosen to closely mimic the sound of the distant microphone.

Summary of the Three Recording Spaces:

Classroom, 2043 E.V. Moore Building
Dimensions: 11.6 m x 5.25 m x 3 m (length, width, height)
Primary Surfaces: Brick, tile, perforated ceiling tile, plastic chairs
Close Mic Position: 0.8 m from center of drum kit
Distant Mic Position: 6.5 m from close mic

Britton Recital Hall
Dimensions: 19.5 m 16.2 m x x 6 m (length, width, height)
Primary Surfaces: Wood, plaster, acoustic tile, upholstered seats
Close Mic Position: 0.9 m from bell of trumpet
Distant Mic Position: 11 m from close mic

Hill Auditorium
Dimensions: 45.7 m x 39.6 m x 16.4 m (length, width, height)
(rough approximate with similar total volume to the true value)
Primary surfaces: Plaster, upholstered seats, carpet, glass, wood
Close Mic Position: 4.6 m from center of ensemble
Distant Mic Position: 25 m from close mic

The impulse response for Hill Auditorium was measured by recording a balloon pop. While not as accurate as other methods, it was easy to perform and was the most practical for the circumstances given. The impulse response of the other two rooms was measured by reproducing a swept sine wave over loudspeakers (sweep generation script in appendix E). The sound was then recorded and the impulse response extracted by means of deconvolution. The built-in deconvolution function in Matlab (deconv.m) unfortunately did not produce the desired result. The problem was remedied by making use of the convolution theorem (division in the frequency domain is the same as deconvolution in the time domain). It was also necessary to ensure that all FFT sizes involved with the process were some power of two (makes process much more efficient). A very steep low pass filter at 8 kHz was also needed. Since the highest exciting tone in the sine sweep was 8 kHz, and there was noise above 8 kHz present in the recording, the deconvolution process introduced a great deal of noise above 8 kHz into the impulse response.

The basic algorithm used in the level meter code was obtained from *DAFX*, a book on audio processing algorithms (Zolzer, 2002). For the creation of an RMS meter, coefficients were chosen such that the meter would drop 20 dB in 300 ms during the release phase and the meter would register –3 dBFS for a signal that was full scale peak to peak (Brixen). The creation of the peak meter was considerably more complex. As mentioned, in order to obtain the desired characteristics (fast release and minimal low frequency oscillation), it was necessary to modify the shape of the release envelope. Rather than simply have a linear release, the algorithm would count the number of samples for which it had been releasing and adjust the shape of the envelope accordingly. The longer the release, the steeper the slope got. A close up of this algorithm in action can be seen in Figure 3. Source code for both metering algorithms are included in appendix E.

It became immediately clear that the data would need to be summarized, compiled, and visualized in order for any meaningful observations to be made. A script was written (numberCrunch1.m) to provide all the necessary computations for a particular sound file. This included the mean level value, standard deviation of level, as well as the crest factor (difference between peak and RMS meters at any instant in time). The script also plotted a histogram of the signal showing how much time it spent at any given level. In addition, the close mic signal was run through a commercially available compressor. Parameters were determined such that the compressed signal would produce similar numbers in analysis to the distant microphone signal. This provided a reference point (albeit imprecise) as to how much compression the analysis indicated.

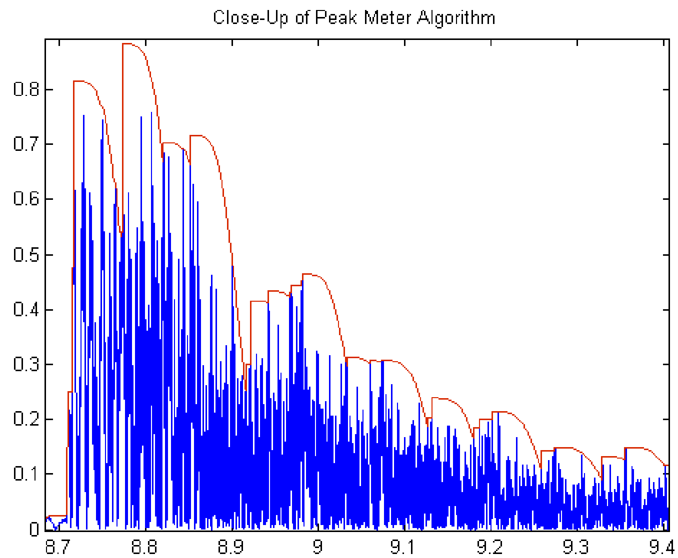


Fig. 3 Demonstration of peak meter algorithm on sample audio signal.

A second script (scatterGen.m) generated scatter plots of one signal versus another. The close mic signal provided the x coordinate and the distant mic signal (or digital reverb signal) provided the y coordinate. The data was also thinned out to make the plotting easier. The hope was to be able to see a transfer function between the close mic and distant mic that had a resemblance to the transfer function of a dynamics processor. Both analysis scripts are included in appendix E.

The reverb time (RT60) was found by running each impulse response through the peak metering algorithm. The slope of the decay of the impulse response was calculated and was then extrapolated to find the RT60 value. The strength factor was found using an equation that Kuttruff has suggested (2000) (See Fig 4). Some approximations needed to be made, however as Kuttruff's equation requires the impulse response of the given sound source in an anechoic space, $g_A(t)$. The value used in this case was the earliest part of the impulse response of Britton recital hall. Since only the relative strength factor is important for this study, this inaccuracy is of minimal significance.

$$G = 10\log_{10}\left\{ \frac{\int_0^\infty [g(t)]^2 \, dt}{\int_0^\infty [g_A(t)]^2 \, dt} \right\}$$

Fig. 4 Equation relating impulse response g(t) to strength factor G. (from Kuttruff)

Reverberation radius was not calculated both because it is difficult to calculate and because its exact value is outside the scope of this study. Since it can be assumed that the close mic was well within the reverberation radius in all cases and the distant mic was well outside the reverberation radius in all cases, the exact value of the reverberation radius does not provide any useful information. In addition, the calculation of the reverberation radius is only approximate at best as it involves a great deal of estimation and is heavily dependent on the directivity of both the source and the microphone (Eargle, 1996)(Dickreiter, 1989). Understanding the concept of reverberation radius, and what it indicates about the nature of the sound field, is sufficient to suit the current needs.

Data

Numerical Analysis of Classroom

	Original Signals		Digital Processing	
	Close Mic	**Distant Mic**	**Reverb**	**Compressor**
Mean Level	-22.29	-17.45	-17.26	-19.13
Std. Dev. of Level	10.33	9.20	8.81	9.18
Std. Dev. Above Mean	9.10	7.68	7.45	7.29
Std. Dev. Below Mean	11.53	10.82	10.21	11.17
Mean Crest Factor	5.23	5.11	5.64	6.12
Std. Dev. of Crest Factor	6.28	5.56	5.11	5.60

(all levels were obtained using peak metering)

The settings for the compressor were: Threshold: -18 dB, Ratio: 1.5, Attack: 0 ms, Release: 200 ms, and Makeup Gain: 4.7 dB.

The first and most noticeable difference is that the average level is higher for the distant mic than it is for the close mic. Since both signals were normalized, a higher average level means smaller dynamic range. It can also be seen that the standard deviation values are all smaller for the distant mic. Again, this indicates less variation in level. The values for standard deviation above and below the mean in the distant mic are very comparable with those of the compressed signal. This suggests that downward compression played more of a role here than did upward compression.

The values for crest factor and standard deviation of crest factor can be a bit difficult to understand. The crest factor itself is probably not very useful as it only changes about 2% and actually increases significantly for both of the digitally processed signals. The standard deviation of crest factor could be a useful figure, though. It is interesting to note that when a histogram of the crest factor is plotted, it forms a near-perfect Gaussian bell curve. The width of this curve is indicated by the standard deviation and it seems to be the width, more than the mean, that changes with dynamic range. Put simply, a larger standard deviation means a larger variation in crest factor, which means a greater change in dynamic range.

Observing the histograms and scatter plots in appendix A, similar trends can be seen. The distant mic histogram and the processed histograms are all shifted towards higher levels when compared with the close mic histogram. While the compressor largely maintained the shape of the histogram, both the distant mic and reverberation spread out the main peak, so that dynamics were more evenly distributed. This would seem to suggest that both compression and expansion are happening simultaneously. The extremes of dynamic range are being compressed in level while the middle values are being expanded. The scatter plots also show a trend that suggests compression. The trend line is highest above reference at the bottom of the range and is the lowest below reference at the top of the range. The size of the cloud, however, indicates that the dynamic effects of reverberation are much more complex than can simply be expressed by a transfer function.

Numerical Analysis of Britton Recital Hall

	Original Signals		Digital Processing	
	Close Mic	**Distant Mic**	**Reverb**	**Compressor**
Mean Level	-23.73	-17.98	-20.44	-13.95
Std. Dev. of Level	17.37	10.31	13.24	10.79
Std. Dev. Above Mean	12.04	7.24	8.79	6.03
Std. Dev. Below Mean	24.07	14.10	19.09	17.93
Mean Crest Factor	9.25	6.55	6.02	10.46
Std. Dev. of Crest Factor	7.75	3.84	4.36	5.53

(all levels were obtained using peak metering)

The settings for the compressor were: Threshold: -36 dB, Ratio: 3.5, Attack: 0 ms, Release: 200 ms, and Makeup Gain: 21.9 dB.

It can be seen here that having the trumpet pointed away from the distant microphone made the differences between the two microphones much greater. This is likely because this prevented almost all of the direct sound from reaching the distant microphone. The sound field at the distant microphone was very diffuse in nature. In order to get a similar value for standard deviation of level, rather extreme settings on the compressor were needed. It is also noticeable that upward compression seems to be taking more of a role in this case. The difference in standard deviation below the mean is about 10 dB between close and distant mics, while the difference in standard deviation above the mean is only 4.8 dB.

Histograms and scatter plots are given in appendix B. The upward compression is also quite clearly visible in the overlay of the close and distant histograms as well as in the scatter plot that compares the close and distant signals.

Numerical Analysis of Hill Auditorium

	Original Signals		Digital Processing	
	Close Mic	**Distant Mic**	**Reverb**	**Compressor**
Mean Level	-21.73	-20.20	-24.10	-18.42
Std. Dev. of Level	10.10	9.05	9.51	8.80
Std. Dev. Above Mean	9.64	8.56	9.21	7.78
Std. Dev. Below Mean	10.57	9.52	9.81	9.86
Mean Crest Factor	7.67	7.07	7.17	7.95
Std. Dev. of Crest Factor	4.84	3.64	3.74	4.42

(all levels were obtained using peak metering)

The settings for the compressor were: Threshold: -18 dB, Ratio: 1.5, Attack: 0 ms, Release: 200 ms, and Makeup Gain: 4.7 dB.

The data from Hill auditorium was the first to be analyzed and was somewhat disappointing at first, as it hardly indicates any compression at all. Even the rather gentle settings of the compressor show more of a difference in the analysis than is found between the close and distant microphones. In looking at the histograms and scatter plots (appendix C), again only a small difference is observed. The overlay of the close and distant histograms is perhaps the best way to see what differences there are. The scatter plots only show a small slope in the trend line towards a compressed dynamic range.

There are likely two reasons for the results being so subtle. The first is that the ensemble was spread out over a large stage. The loudest parts of the ensemble, the brass and percussion, were at the back where they could have been approaching the reverberation radius. The second reason is that Hill has a parabolically shaped ceiling which is very efficient at reflecting sound from the stage out into the audience. As a result, it is very possible that the sound field at the location of the distant microphone was much less diffuse than what one would normally expect for a room of that size.

Room Data

	Classroom	Recital Hall	Auditorium
Strength Factor (dB)	10.07	7.23	11.65
Reverb Time (RT60) (sec)	0.69	1.24	2.03

The strength factors and reverberation times in the table above provide a basis for comparing the acoustic properties of the rooms. The larger strength factor that room 2043 has over Britton can likely be attributed to its smaller size and many reflective surfaces. The very large strength factor of Hill could also provide verification about the efficiency of its ceiling. In addition, sonograms of the impulse responses of the rooms are included in appendix D. It is noticeable that while the impulse response of 2043 is fairly short, it is spectrally rich. The high frequencies and low frequencies decay at a similar rate. The two larger rooms have a tendency to hang onto the low frequencies longer. Note that the sonograms for 2043 and Britton are 1.35 seconds in length while the sonogram for Hill is 2.53 seconds in length.

Conclusions

With the complexity of the data gathered, and the number of variables at play, it is difficult to make a concrete determination as to whether the hypothesis was correct. Reverberation does undoubtedly provide some compression of dynamic range. As can be seen in the scatter plots, the change in dynamics is much more complex than can be expressed by a transfer function. It is also difficult to make any specific connections between the acoustics characteristics of the room and the extent to which upward or downward compression specifically will occur.

In terms of answering the question of “what is room compression?,” it seems to be that it is entirely dependent on the relationship between the microphone position and the reverberation radius. Since the reverberation radius shrinks and grows with the size of the room, this would explain why room compression is observed more often in smaller spaces. As the microphone moves outside of the reverberation radius, the compressing effects of the room’s reverberation come more into play.

Bibliography

Brixen, E. Audio Levels and Readings. (n.d.). Retrieved September 17, 2005, from http://www.broadcastpapers.com/audio/DKAudioLevel&Readings03.htm

Dickreiter, M. (1989). *Tonmeister technology.* New York: Temmer Enterprises, Inc.

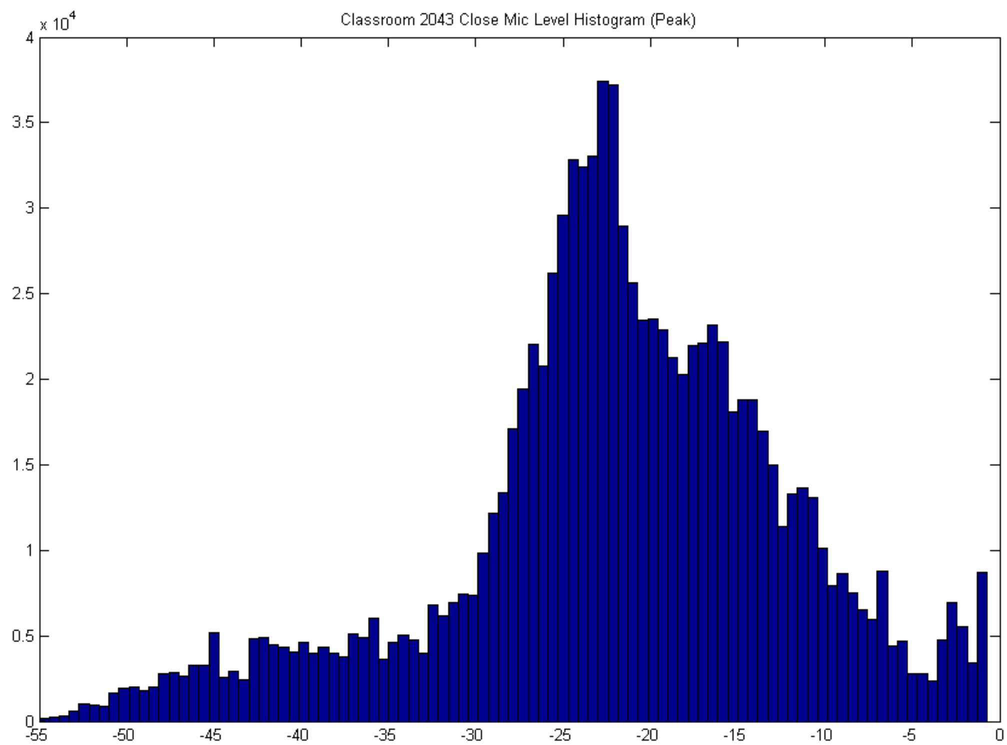
Eargle, J. (1996). *Handbook of Recording Engineering (third ed.).* New York: Chapman & Hall.

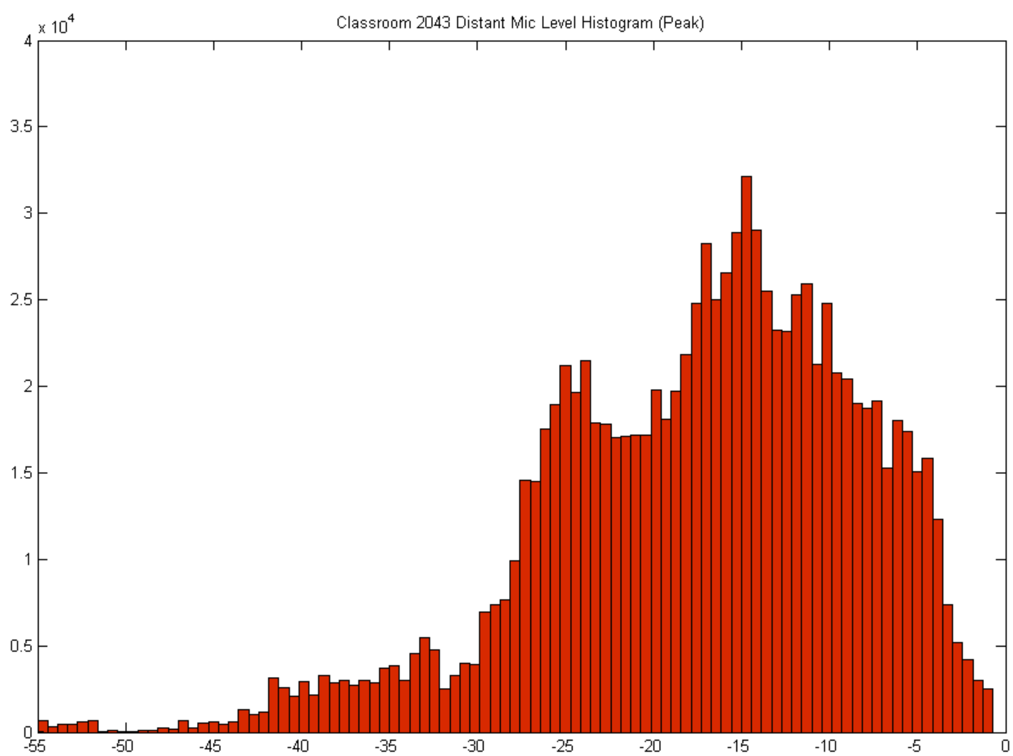
Kuttruff, H. (2000). *Room acoustics (fourth ed.).* New York: Spon.

Zolzer, U. (2002). *DAFX: Digital Audio Effects. New York: Wiley.*

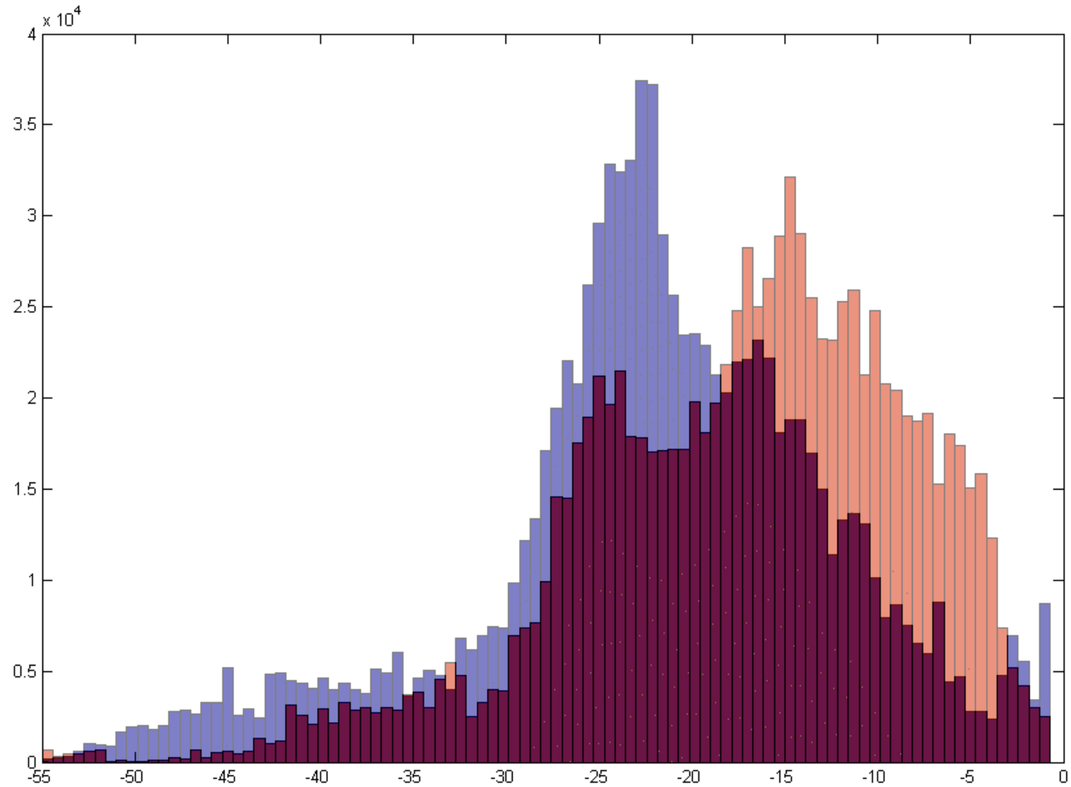
Zwicker, E. & Fastl, H. (1990). *Psychoacoustics*. Berlin: Springer-Verlag.

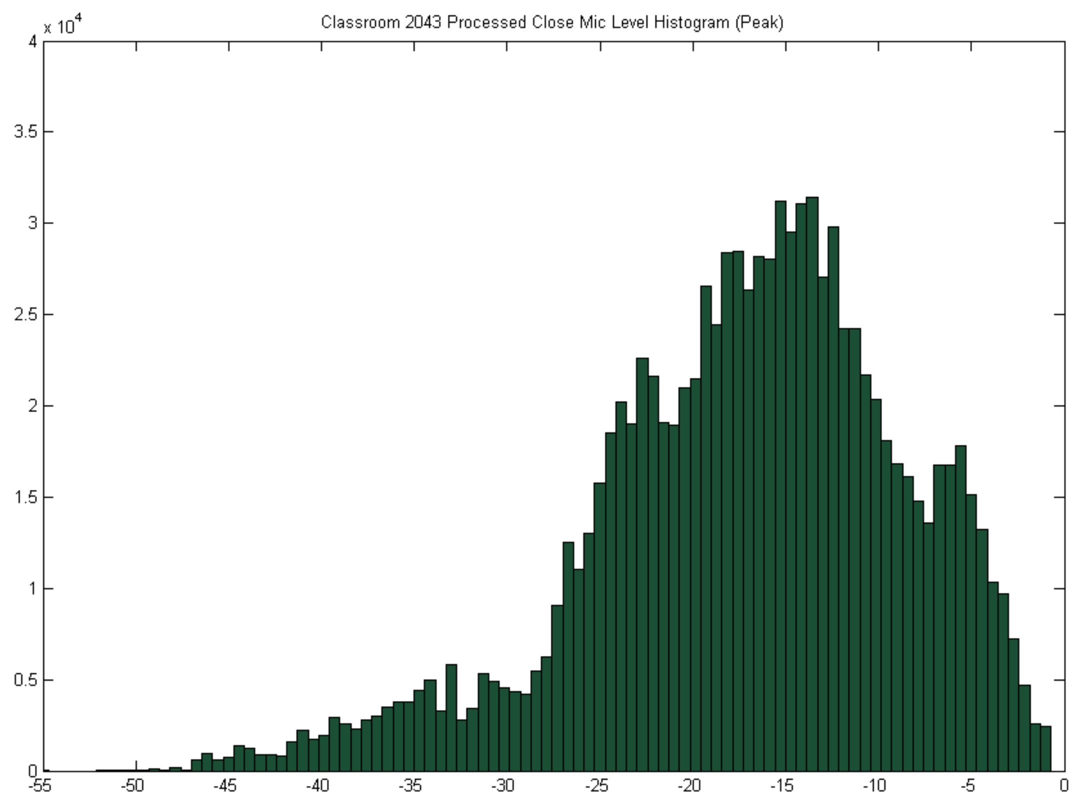
Appendix A: Room 2043 Analysis Graphs

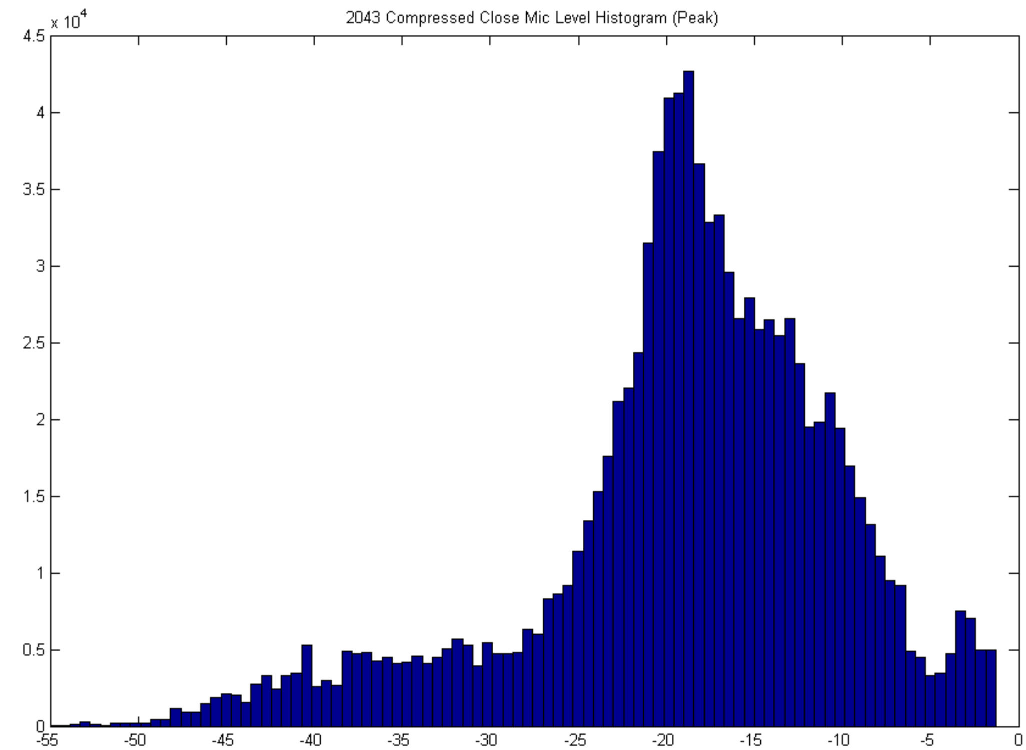




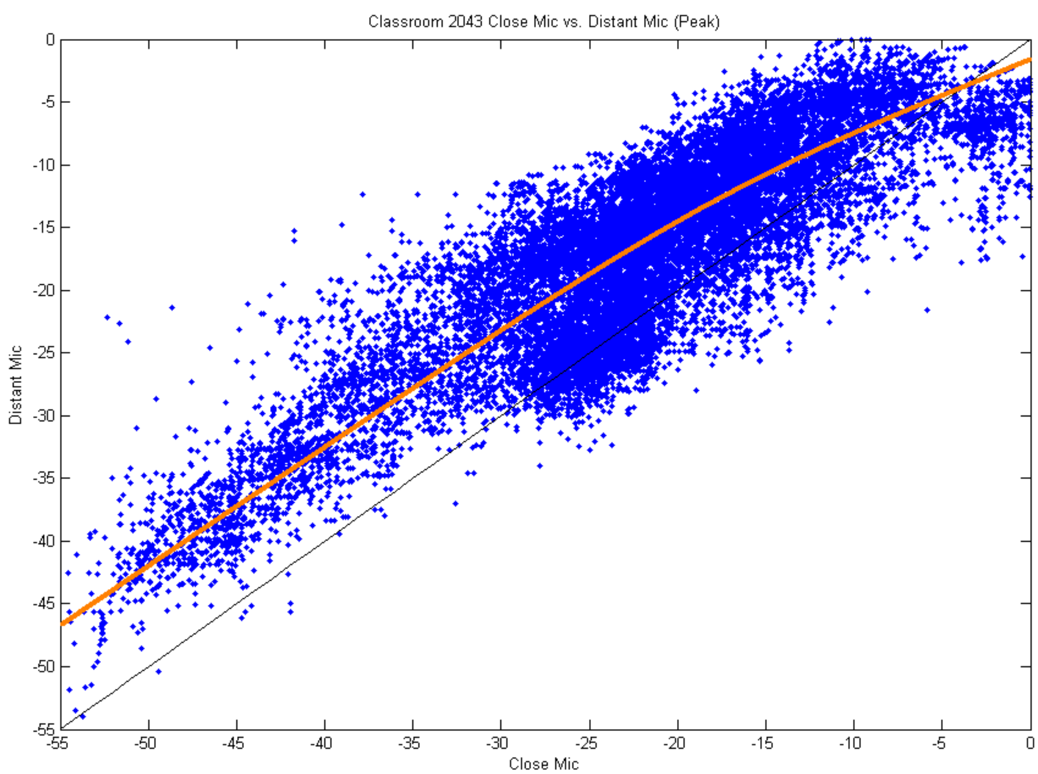
Close Mic (Blue) and Distant Mic (Red) Histograms Overlaid



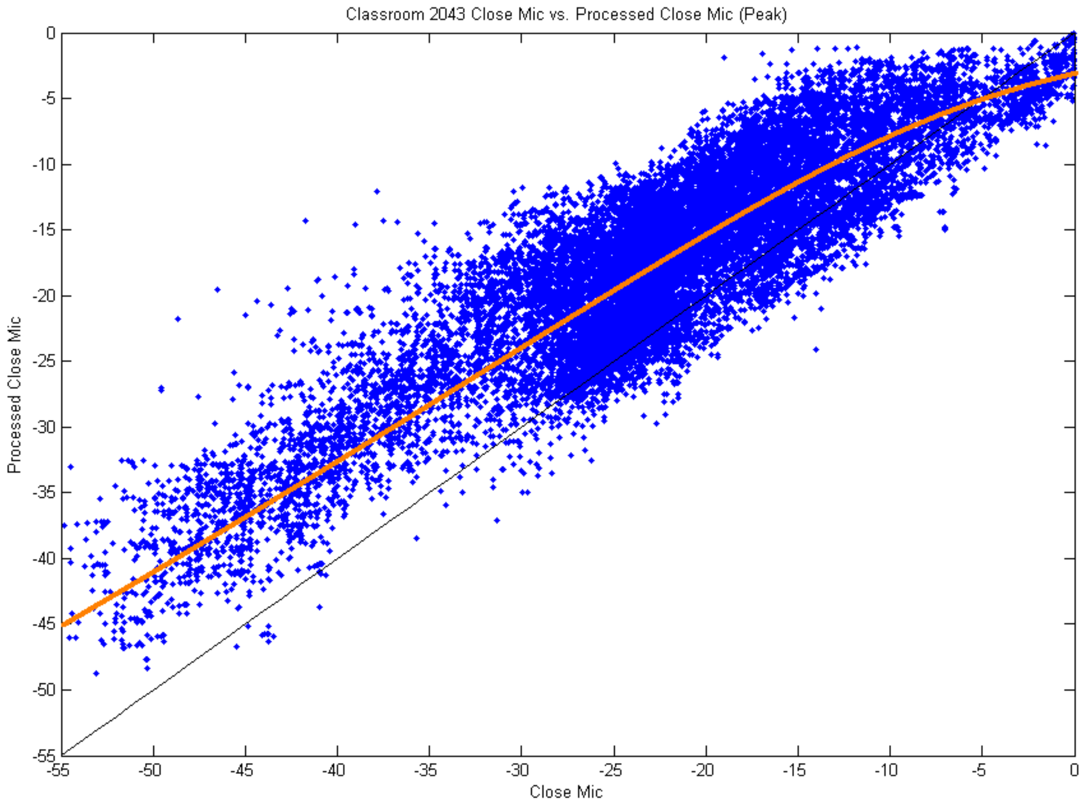




(note different scale)

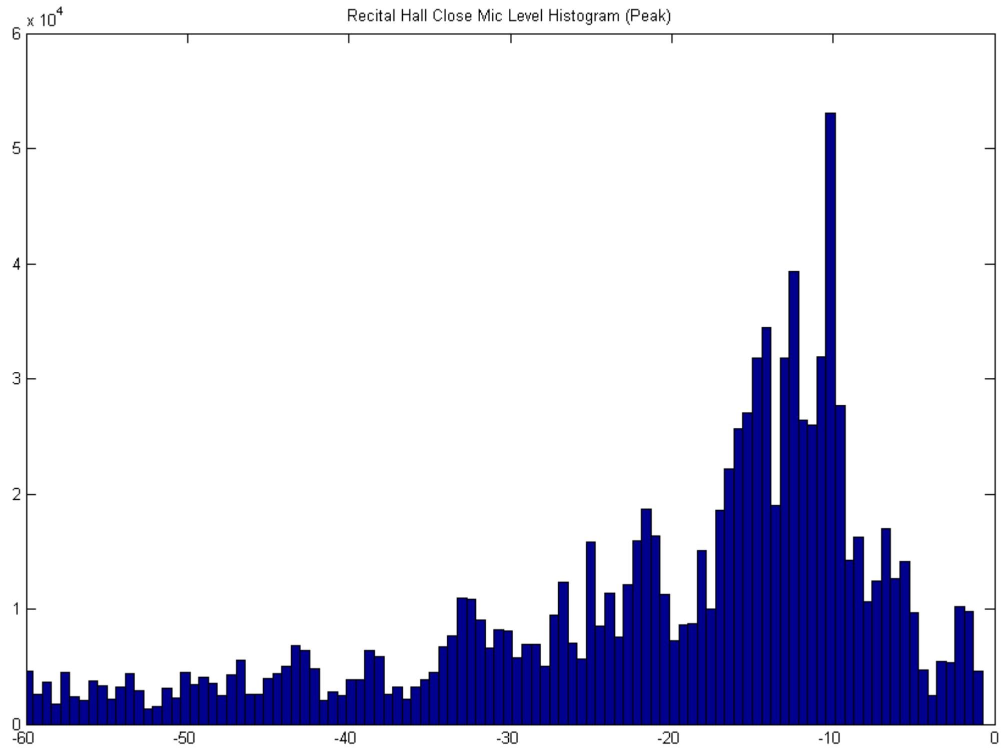


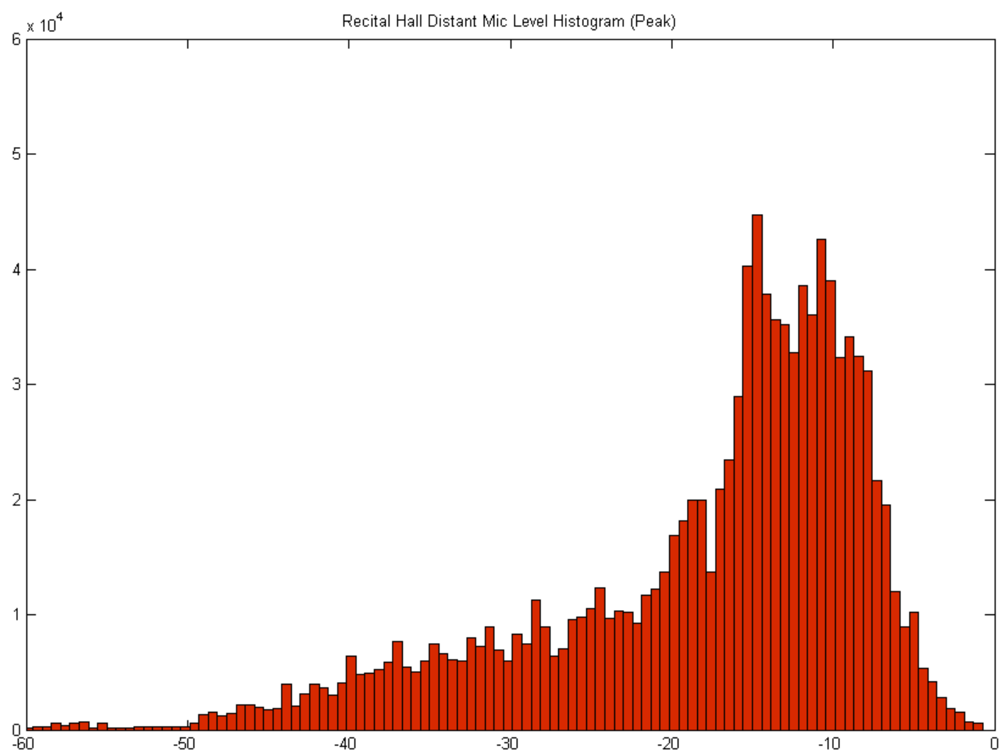
(estimated trend line drawn in orange)



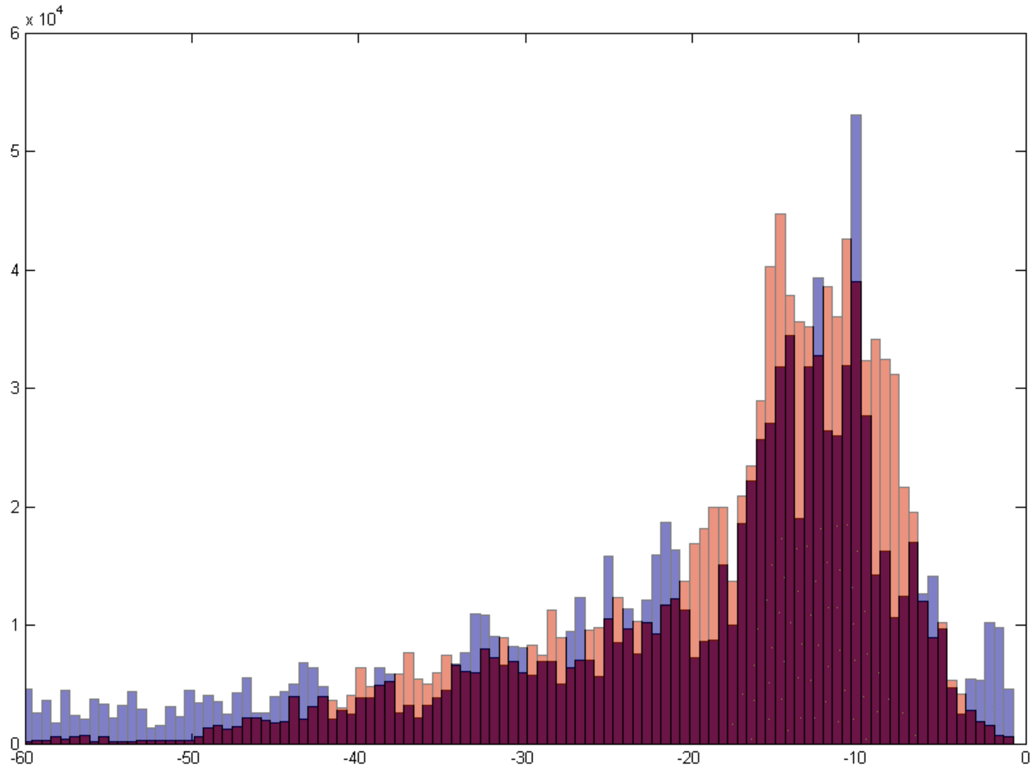
(estimated trend line drawn in orange)

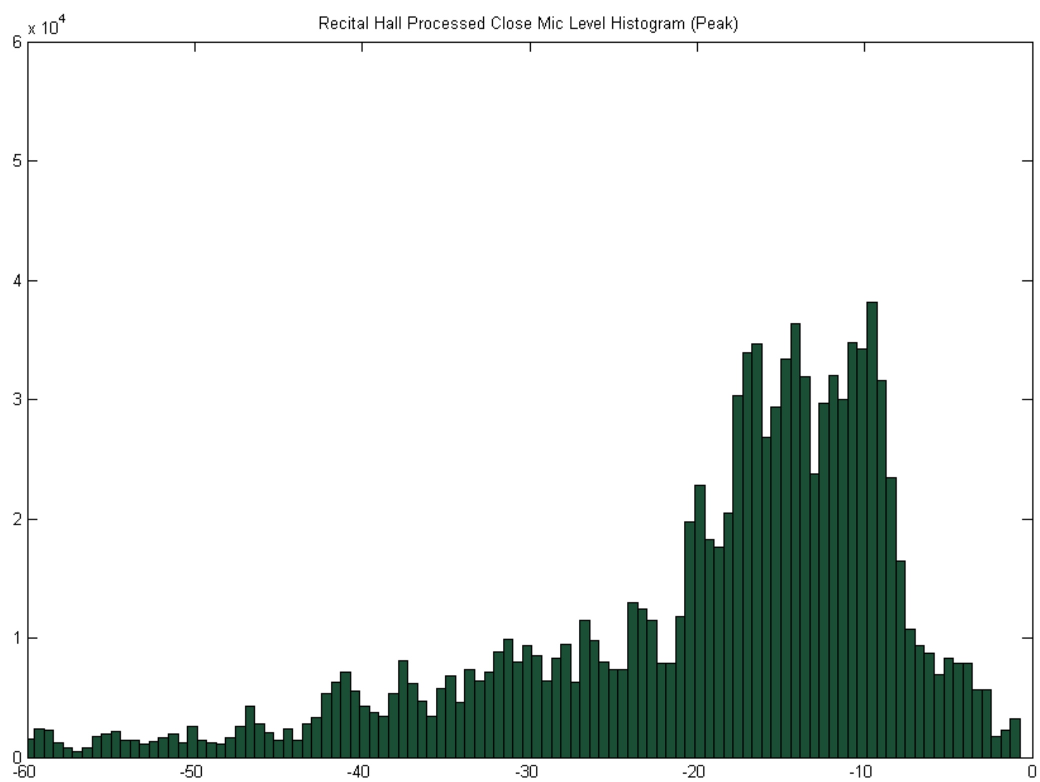
Appendix B: Britton Recital Hall Analysis Graphs

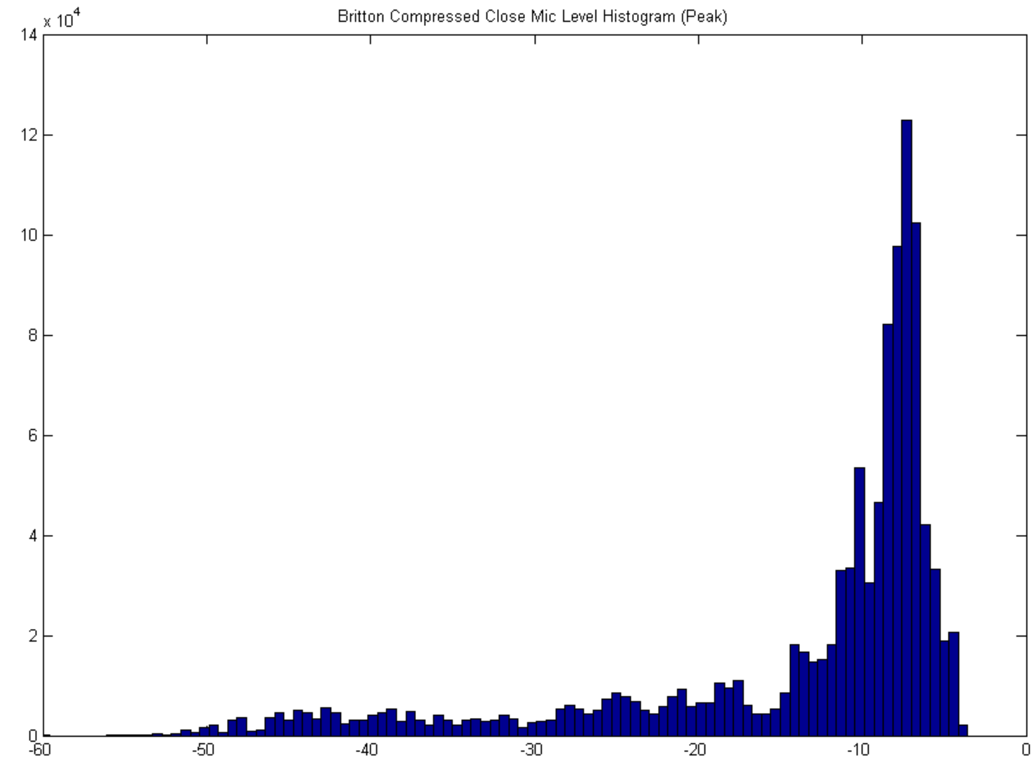




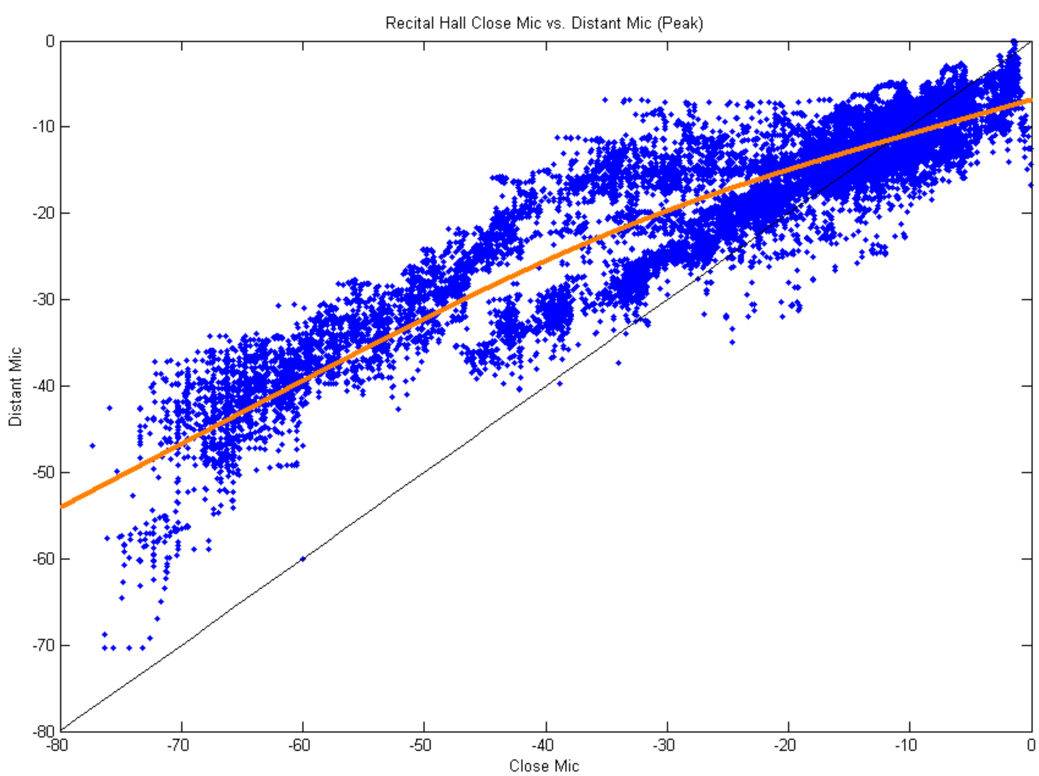
Close Mic (Blue) and Distant Mic (Red) Histograms Overlaid



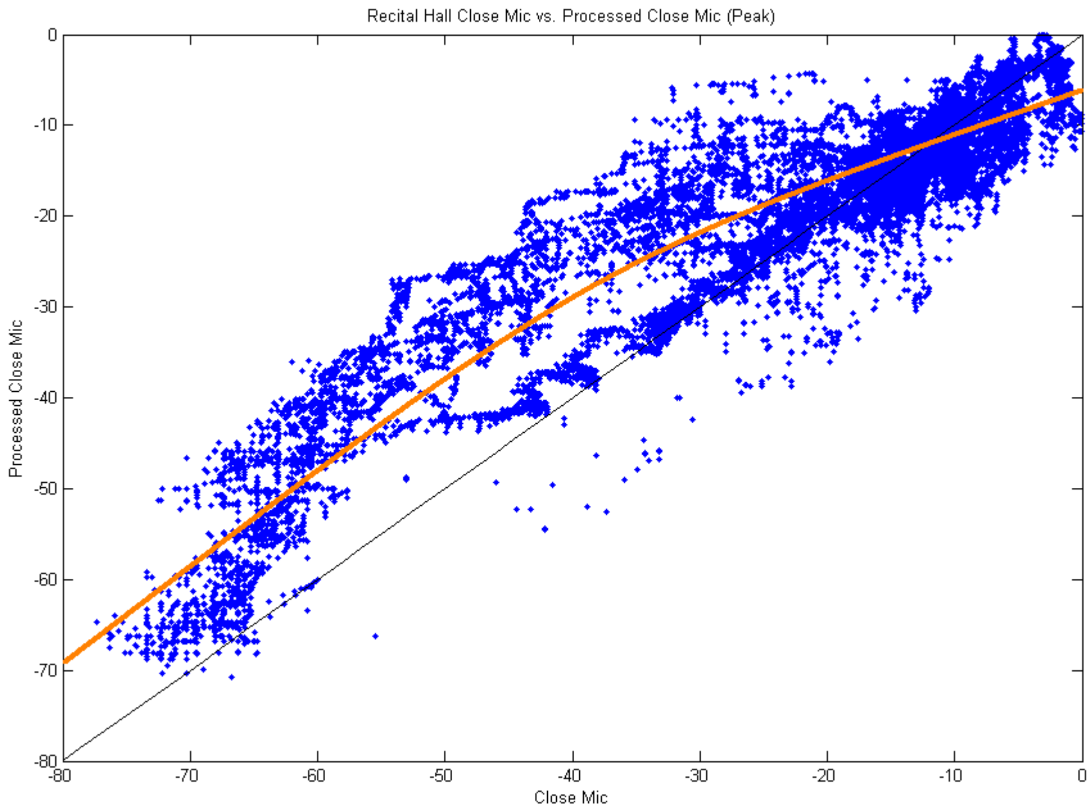




(note different scale)

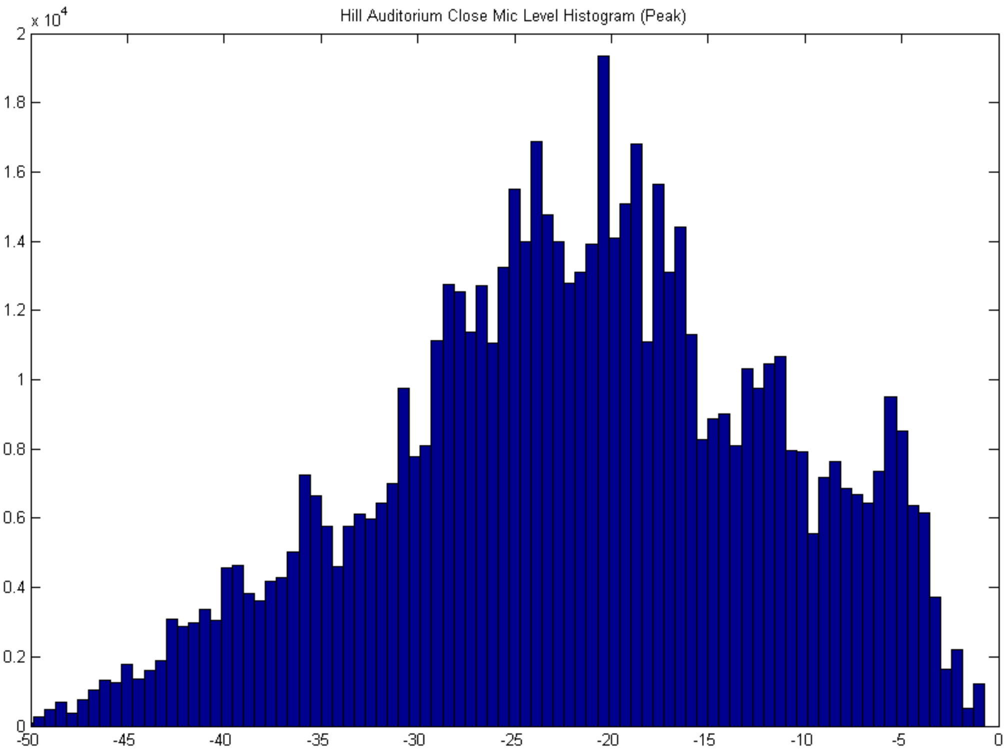


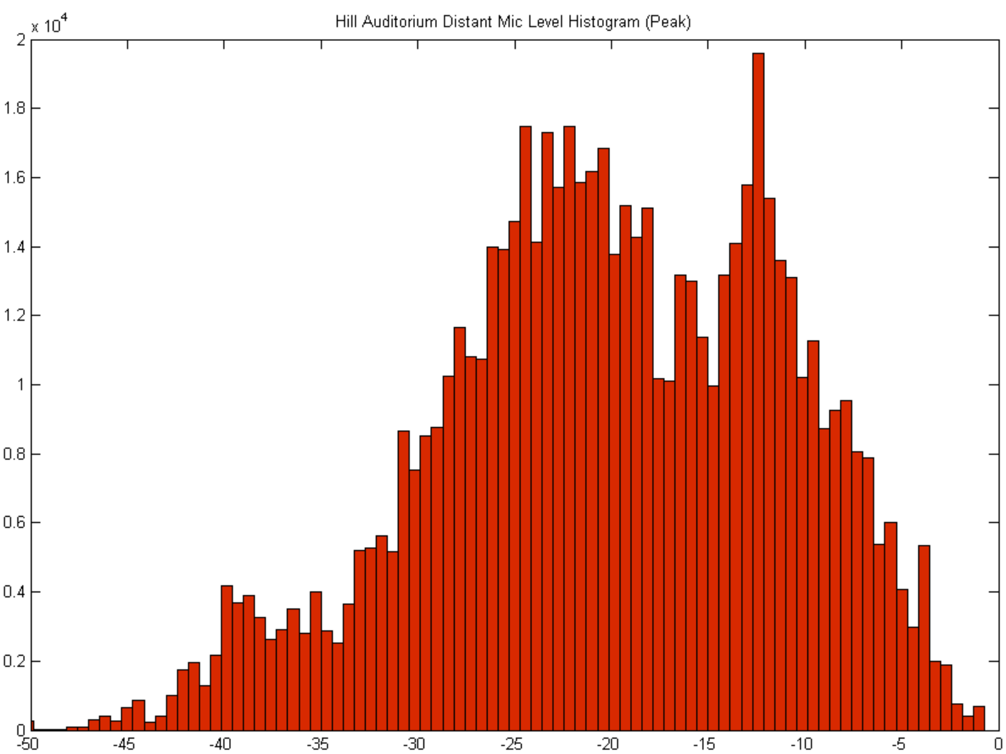
(estimated trend line drawn in orange)



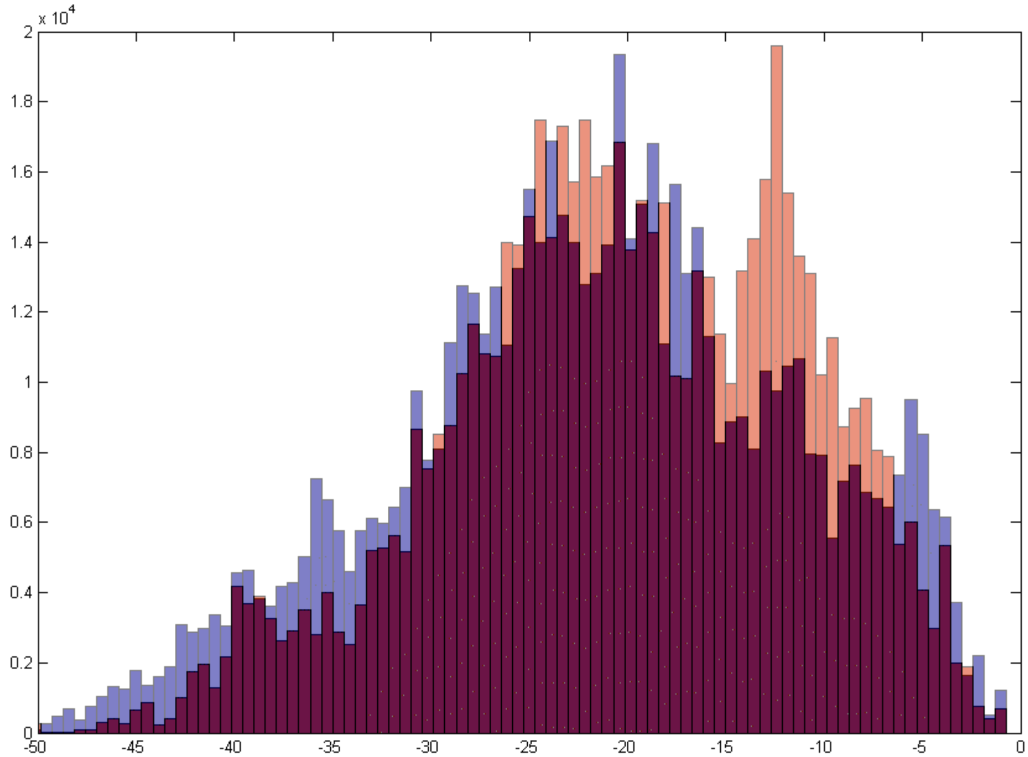
(estimated trend line drawn in orange)

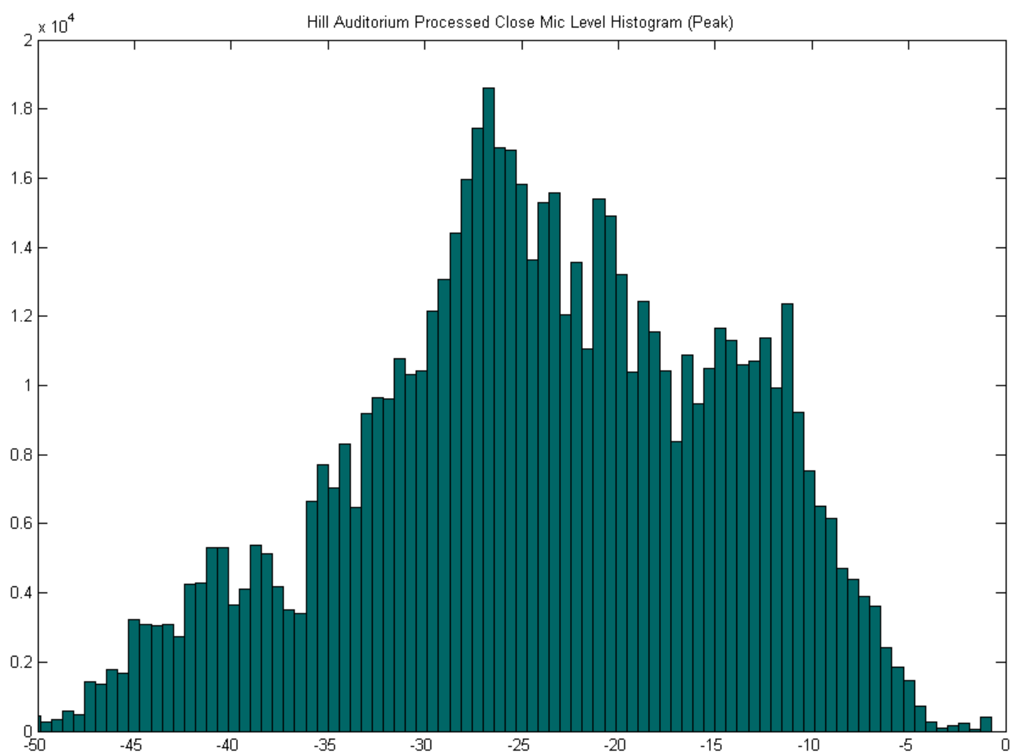
Appendix C: Hill Auditorium Analysis Graphs

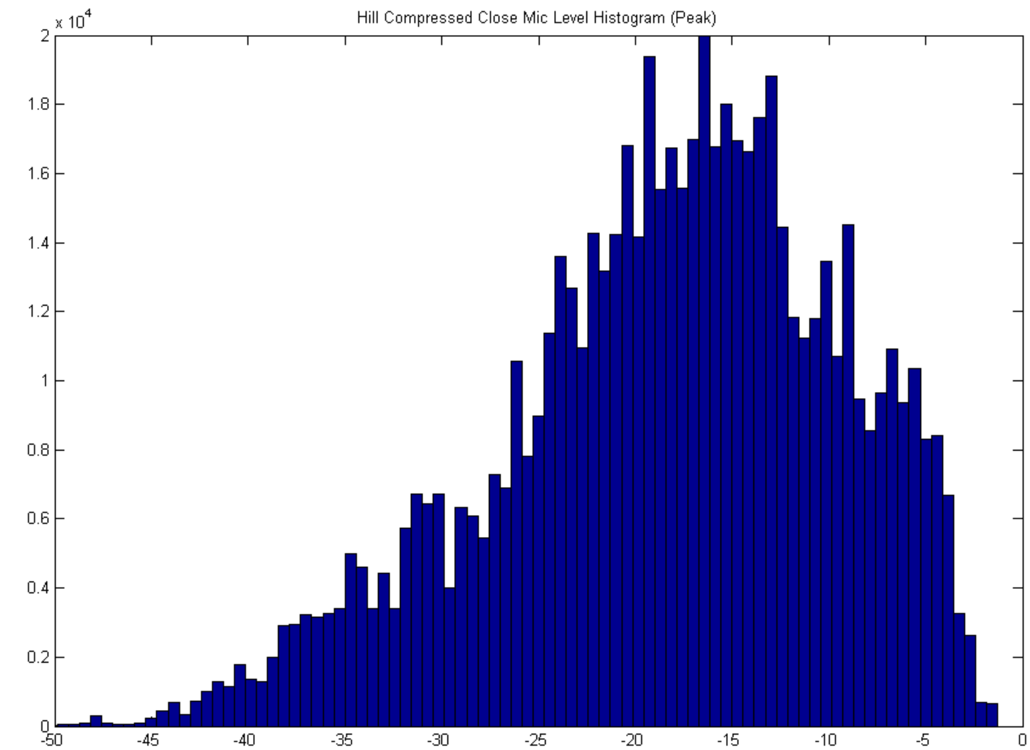


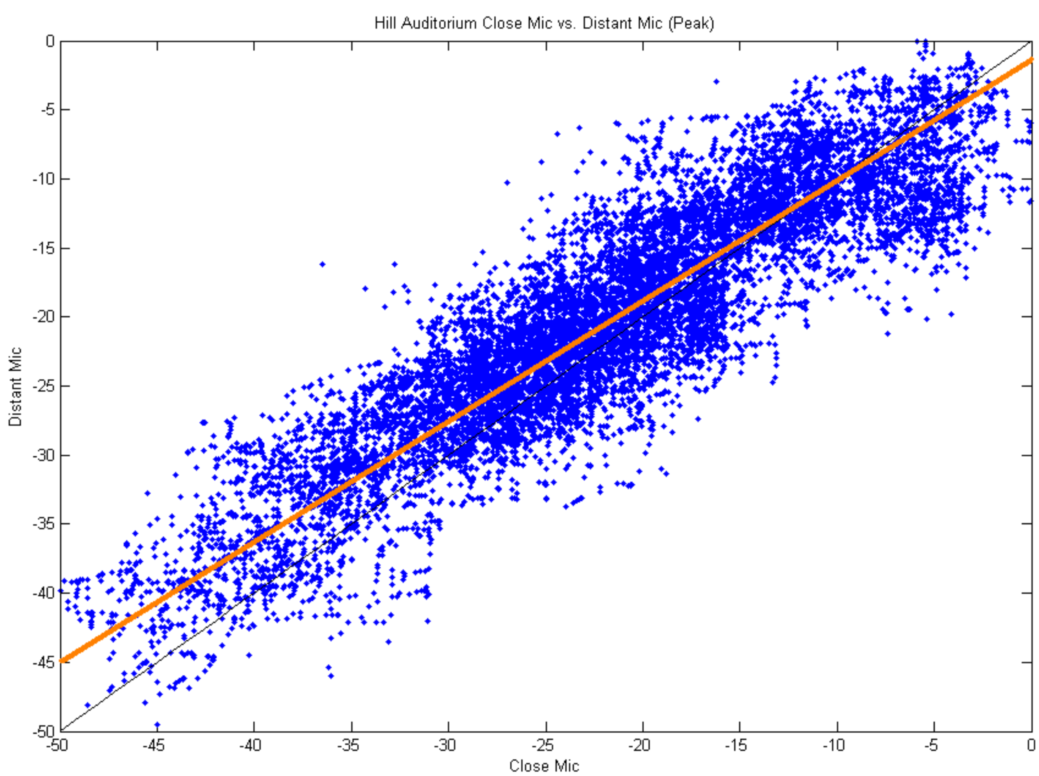


Close Mic (Blue) and Distant Mic (Red) Histograms Overlaid

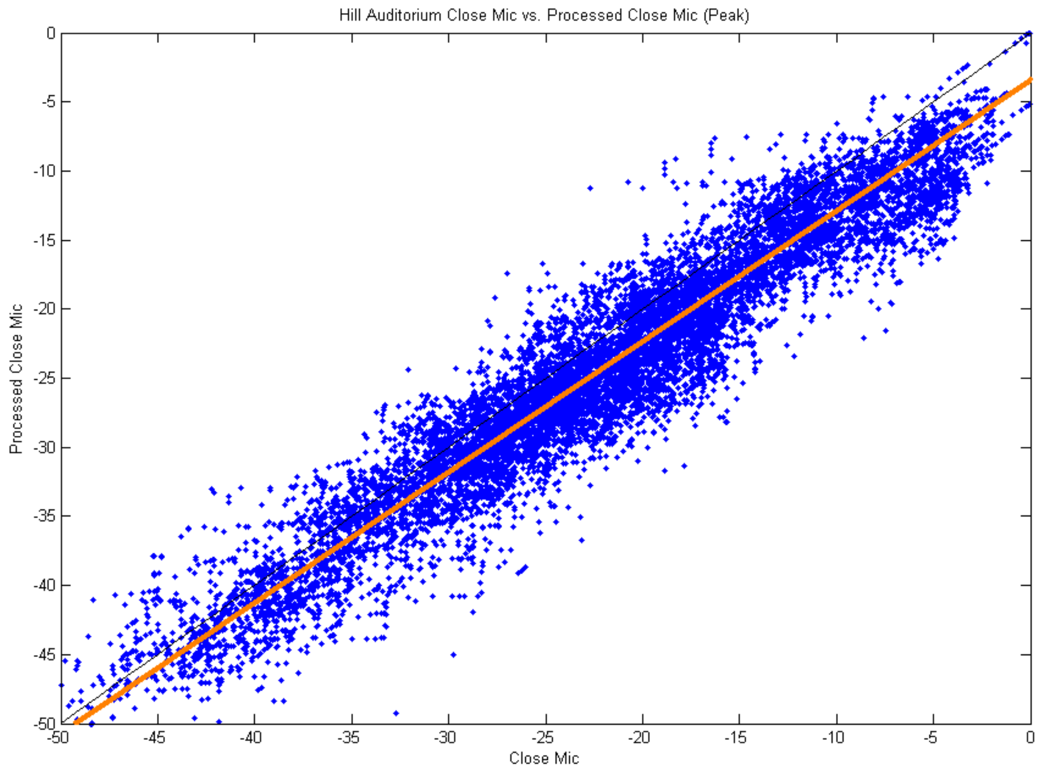








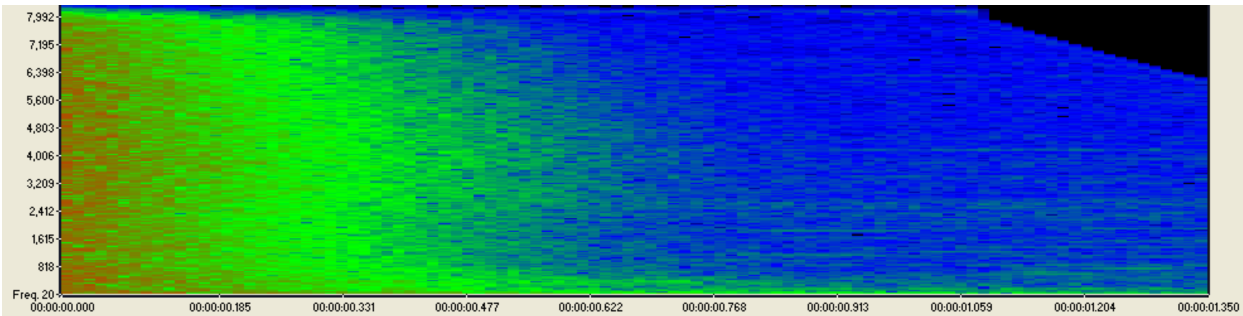
(estimated trend line drawn in orange)



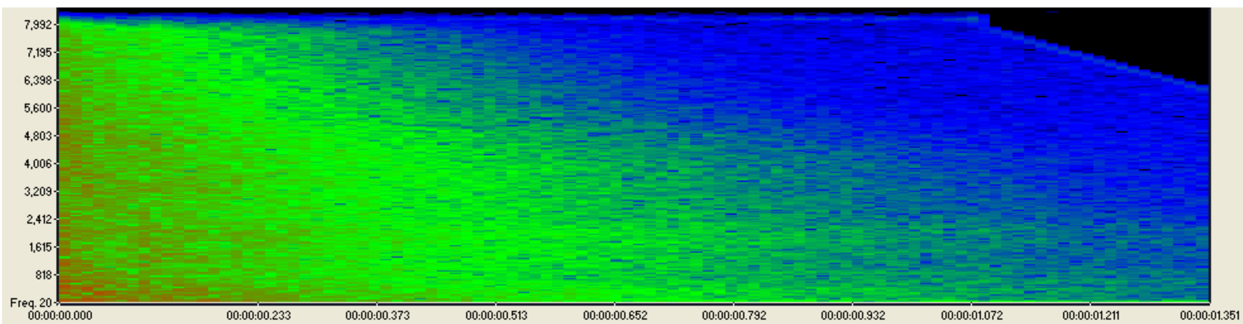
(estimated trend line drawn in orange)

Appendix D: Impulse Response Sonograms

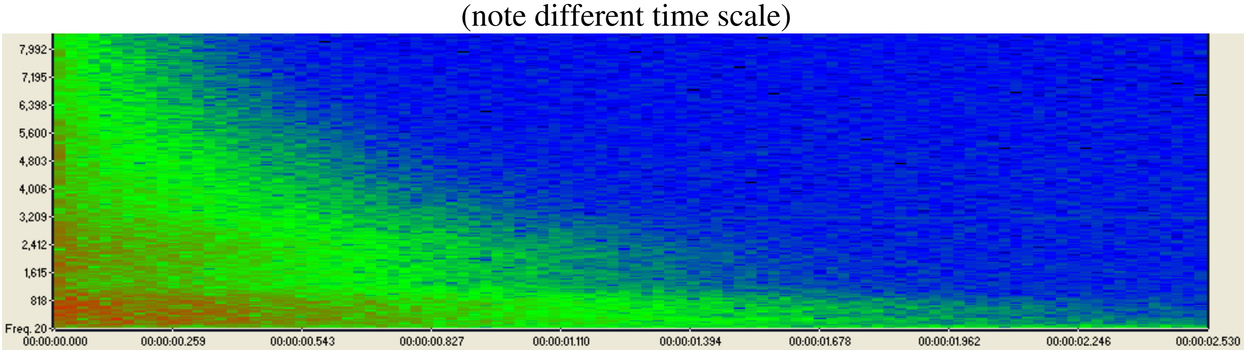
Classroom 2043



Britton Recital Hall



Hill Auditorium
(note different time scale)



Appendix E: Matlab Scripts

sineSwp.m
Script for generating a swept sine wave

```
%Sine sweep
%Sweeps a sine wave from 63 Hz to 8000 kHz

% Sample rate
FS=44100;

% Sample period
TS=1/FS;

% Duration in seconds
Dur=5;

% Time vector
t=0:TS:Dur;

% Frequency vector
freq=logspace(log10(63), log10(8000), length(t));
% Frequency * sample period
freqT=freq.*TS;

% Integrate freqT
freqI=zeros(size(t));
freqI(1)=freqT(1);
for i=2:length(t);
    freqI(i)=freqT(i)+freqI(i-1);
end

% Remove initial offset
freqI=freqI-freqI(1);

% Compute sweep
sweep=sin(2*pi*freqI);

% Write wav file
wavwrite(sweep, FS, 16, 'sine_sweep.wav');
```

variRMS.m
Script for RMS metering of a signal

```
function output=variRMS(input, atkTime, relTime);
% output=variRMS(input, atkTime, relTime);
% input= input signal (single vector only)
% atkTime= attack time in ms
% relTime= release time in ms
%
% Function finds the RMS value of the input audio signal
% based on the given attack and release times.
%
% Values used for the RMS meter were:
% atkTime=87; relTime=150;
```

```
% sample rate
FS=44100;

% Rectify input
input=abs(input);

atkTime=atkTime/1000;
relTime=relTime/1000;

% Attack/release envelope values
ga=exp(-1/(FS*atkTime));
gr=exp(-1/(FS*relTime));

output=zeros(size(input));
output(1)=0.001; % Set first output sample to -60 dB

for n=2:length(input);
    if(output(n-1) < input(n)); % Attack phase
        output(n)=(1-ga)*input(n)+ga*output(n-1);
    else % Release phase
        output(n)=(1-gr)*input(n)+gr*output(n-1);
    end
end

% Convert Output to dB
output=20*log10(output);
```

superPeak.m
Script for peak metering of a signal

```
function output=superPeak(input, relTime, divFactor, powFactor);
% output=superPeak(input, relTime, divFactor, powFactor);
% input= input signal (single vector only)
% Suggested values:
% relTime = 100;
% divFactor = 188;
% powFactor = 3;
%

% sample rate
FS=44100;

% Rectify input
input=abs(input);

relTime=relTime/1000;

relSamps=0; % number of samples releasing for

output=zeros(size(input));
output(1)=0.001; % Set first output sample to -60 dB

for n=2:length(input);
    if(output(n-1) < input(n)); % Attack phase
        output(n)=input(n);
```

```
        relSamps=0; % Reset relSamps
    else % Release phase
        relSamps=relSamps+1; % Increment relSams while releasing
        % Compute variable release time
        VrelTime=relTime*((divFactor/relSamps).^powFactor);

        % Release envelope value
        gr=exp(-1/(FS*VrelTime));
        output(n)=(1-gr)*input(n)+gr*output(n-1);
    end
end

% Convert Output to dB
output=20*log10(output);
```

numberCrunch1.m
Script for analyzing audio files

```
GraphTitle='Britton Compressed Close Mic Level Histogram (Peak)';
DataPath='Britton Data/Compress1.wav';

% Input Data
[sig1,FS,NBITS]=wavread(DataPath);
sig1=abs(sig1(:,1));
dataLen=length(sig1);

disp('Results for: ');
disp(DataPath);

% Meter Signal Both by Peak and RMS
peakSig1=superPeak(sig1, 100, 188, 3);
rmsSig1=variRMS(sig1, 87, 150);

% Crest Factor
crestSig1=peakSig1-rmsSig1;

% Find Mean Level
peakMean=mean(peakSig1);
disp('Peak Mean');
disp(peakMean);
crestMean=mean(crestSig1);
disp('Crest Mean');
disp(crestMean);

% Find Standard Deviation of Level
peakStd=std(peakSig1);
disp('Peak Std Dev');
disp(peakStd);
crestStd=std(crestSig1);
disp('Crest Std Dev');
disp(crestStd);

% Find Upper Standard Deviation (Above Average)
peakHighStd=highStd(peakSig1, peakMean, dataLen);
disp('Peak High Std Dev')
disp(peakHighStd);
```

```
% Find Lower Standard Deviation (Below Average)
peakLowStd=lowStd(peakSig1, peakMean, dataLen);
disp('Peak Low Std Dev')
disp(peakLowStd);

% Histogram Level Vector
levelVec=linspace(-69, -1, 120);

% Plot Peak Level Histogram
hist(peakSig1, levelVec);
title(GraphTitle);
```

highStd.m
Script for finding standard deviation above the mean

```
function output=highStd(sig, mean, dataLen);
% output=highStd(sig, mean, dataLen);
%
% highStd finds the standard deviation while only counting values which are
% above the mean (average).

dataSet=zeros(1, dataLen);

dataCount=0;

% Sort values which are above average into dataSet
for i=1:dataLen;
    if(sig(i) > mean)
        dataCount=dataCount+1;
        dataSet(dataCount)=sig(i);
    end
end

% Compact size of dataSet
dataSet=dataSet(1:dataCount);

% Find standard deviation of dataSet
dataSet=(dataSet-mean).^2;
output=sum(dataSet)/dataCount;
output=sqrt(output);
```

lowStd.m
Script for finding standard deviation below the mean

```
function output=lowStd(sig, mean, dataLen);
% output=lowStd(sig, mean, dataLen);
%
% lowStd finds the standard deviation while only counting values which are
% below the mean (average).

dataSet=zeros(1, dataLen);

dataCount=0;
```

```
% Sort values which are below average into dataSet
for i=1:dataLen;
    if(sig(i) < mean)
        dataCount=dataCount+1;
        dataSet(dataCount)=sig(i);
    end
end

% Compact size of dataSet
dataSet=dataSet(1:dataCount);

% Find standard deviation of dataSet
dataSet=(mean-dataSet).^2;
output=sum(dataSet)/dataCount;
output=sqrt(output);
```

scatterGen.m
Script for generating scatter plots

```
% Thinning factor
thinFac=50;

% Lower limit of graph display (in dB)
lowLim=-80;

refLine=lowLim:0;

% Input Data
[sig1,FS,NBITS]=wavread('Hill Data/Close1.wav');
[sig2,FS,NBITS]=wavread('Hill Data/Distant1.wav');
[sig3,FS,NBITS]=wavread('Hill Data/Compress1.wav');
sig1=abs(sig1(:,1));
sig2=abs(sig2(:,1));
sig3=abs(sig3(:,1));

dataLen=length(sig1);

% Truncate sigs 2 and 3 to length of sig1
sig2=sig2(1:dataLen);
sig3=sig3(1:dataLen);

% Meter Signals Both by Peak and RMS
peakSig1=superPeak(sig1, 100, 188, 3);
peakSig2=superPeak(sig2, 100, 188, 3);
peakSig3=superPeak(sig3, 100, 188, 3);

i=1;
j=1;
TdataLen=ceil(dataLen/thinFac);

TpeakSig1=zeros(1, TdataLen);
TpeakSig2=zeros(1, TdataLen);
TpeakSig3=zeros(1, TdataLen);

while(i <= dataLen)
    TpeakSig1(j)=peakSig1(i);
```

```
    TpeakSig2(j)=peakSig2(i);
    TpeakSig3(j)=peakSig3(i);
    i=i+thinFac;
    j=j+1;
end

% Peak Sig1 vs. Sig2
plot(TpeakSig1, TpeakSig2, '.', refLine, refLine, 'k-');
title('Hill Close Mic vs. Distant Mic (Peak)');
xlabel('Close Mic');
ylabel('Distant Mic');
axis([lowLim 0 lowLim 0]);
figure;

% Peak Sig1 vs. Sig3
plot(TpeakSig1, TpeakSig3, '.', refLine, refLine, 'k-');
title('Hill Close Mic vs. Compressed Close Mic (Peak)');
xlabel('Close Mic');
ylabel('Compressed Close Mic');
axis([lowLim 0 lowLim 0]);
% figure;
```