EID465 Sound and Space
Final Project Part 3

Yuval Epstain Ofek and Daniel Zuerbig

May 27, 2020

1 Introduction

We completed an acoustic simulation of the Rose Auditorium at The Cooper Union. We began by creating a simple model of the space in SketchUp, and then importing the model in CattAcoustics. Using the 3D model, we assigned acoustic material properties to the various surfaces, and used this to calculated a T_{30}, C_{80}, and EDT for the $125-4000\ Hz$ octave bands. Our results are shown below.
Following this, we obtained room response measurements from tests that had been performed in the Rose Auditorium. The locations of the source and receivers are approximately the same as the ones used to generate the CATTAcoustic responses. Using these measurements, we calculated a T_{30}, C_{80}, and EDT for the $125-4000\ Hz$ octave bands. Our results are shown below.
The model and actual readings were compares and the absorption and scattering properties of the model surfaces were adjusted match the taken data. Finally, suggestions were provided to changes in surfaces in order to improve the Acoustics of Rose Auditorium.

2 Simulation

2.1 Schematic

The model generated with the materials chosen is seen in figures 1 to 6. The meaning of the colors used in the model are described in figure 7.

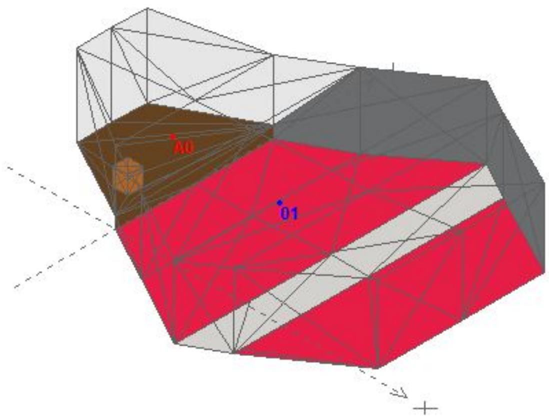

Figure 1: Rose Auditorium Model - Isometric View

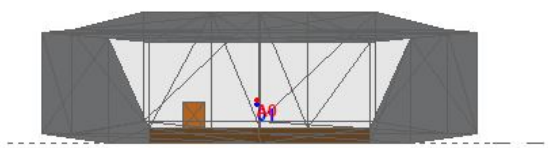

Figure 2: Rose Auditorium Model - +X axis View

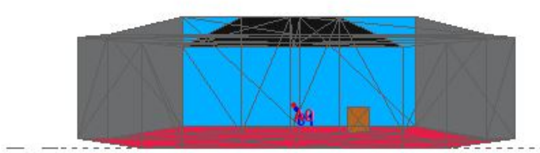

Figure 3: Rose Auditorium Model - -X axis View

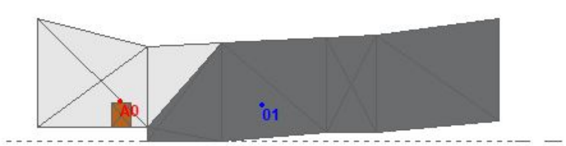

Figure 4: Rose Auditorium Model - Y axis View

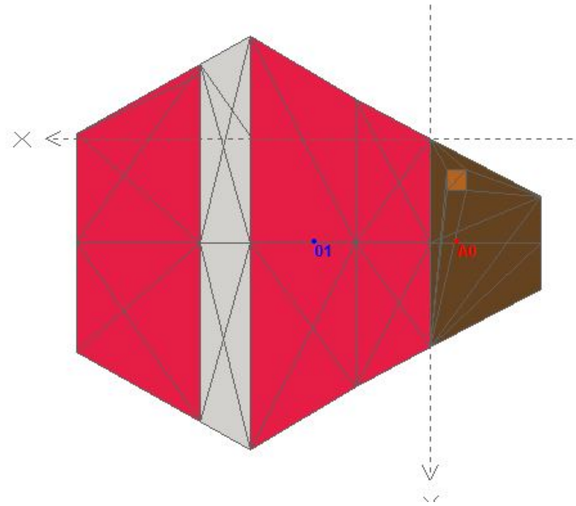

Figure 5: Rose Auditorium Model - +Z axis View

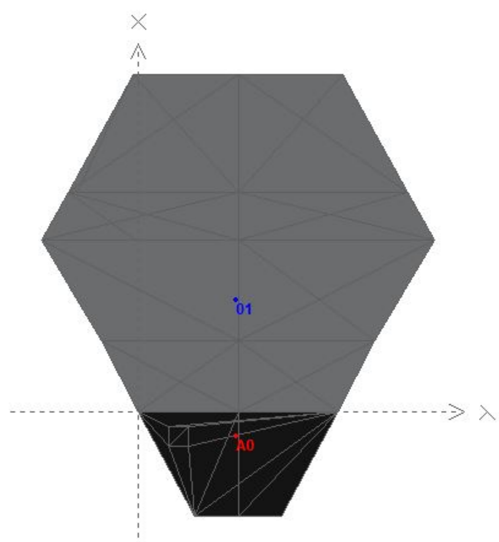

Figure 6: Rose Auditorium Model - -Z axis View

Figure 7: Materials Denoted by Each Color

Color	Material
	Concrete
	Audience Seats
	Metal Mesh
	Plaster
	Stage Wood
	Podium Wood
	Glass
	Stage Ceiling

2.1.1 Receivers

The receiver locations are shown in figures 8 through 11.

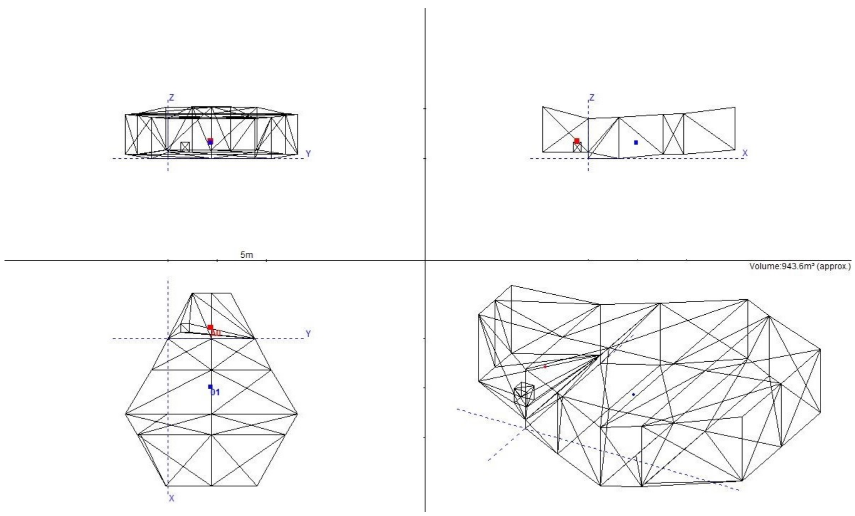

Figure 8: Location of Receiver 1

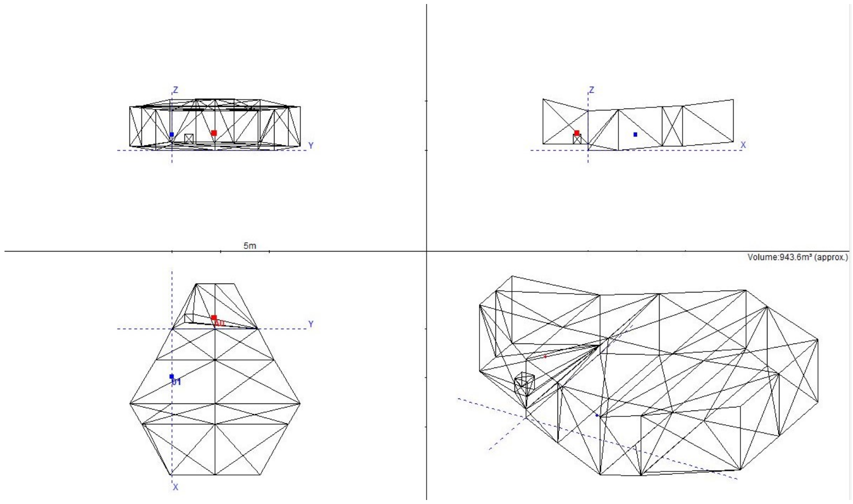

Figure 9: Location of Receiver 2

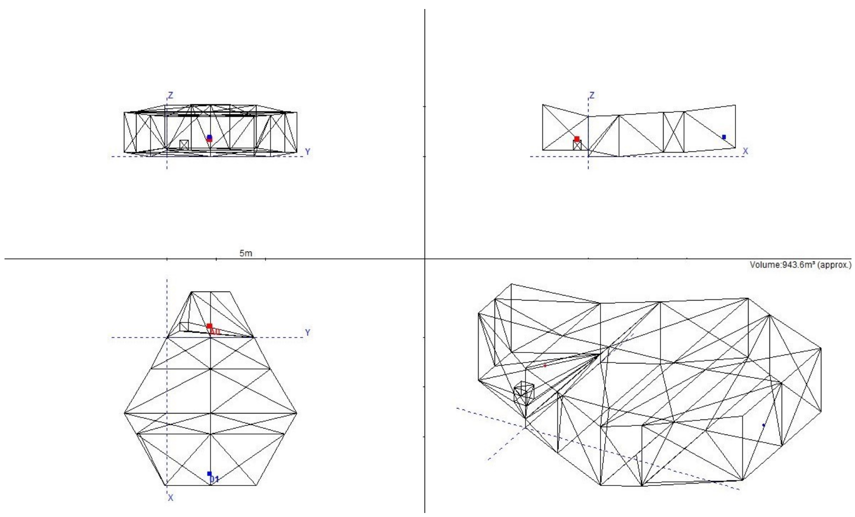

Figure 10: Location of Receiver 3

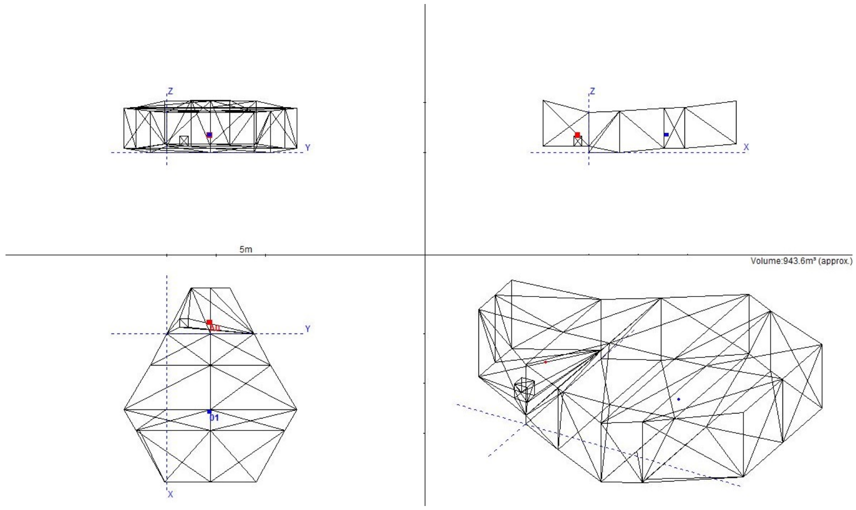

Figure 11: Location of Receiver 4

2.2 Materials

In order to accurately model the Rose Auditorium, we had to assign material properties to different surfaces. For the empty floor, we chose concrete and for the the seating sections we chose to use the empty seats upholstered with fabric covers. These choices were made because the floor is made out of concrete and the seats are plastic with a fabric on top, therefore these coefficients were the closest match. The stage and podium are made out of wood two types of wood, the stage used the coefficients for the ”Wood, stage floor” while the podium used the coefficients for the ”Wood, 1.6 cm thick, on 4 cm wooden planks”. We thought that the stage coefficients will be the closest match for our stage, while the podium is made out of thin wood and is placed on top of the wooden stage, so the coefficients we picked should be a close fit. For the side walls and ceiling of the stage we chose plaster, as we remembered that those were plaster-like hard materials. The metal mesh around the auditorium (walls and ceiling) was modeled as the ”Metal panel ceiling, backed by 20 mm Sillan acoustic tiles, panel width 85 mm, panel spacing 15 mm, cavity 35 cm”. We saw in pictures that the sides and ceiling had a very similar metal cover, and this choice of coefficients made sense as we had a metal paneled ceiling and there are are probably other acoustical materials behind it to enhance the acoustics of the auditorium. The back window to the tech booth we modeled as glass. We did this as the window is often left uncovered and is made out of glass.

The scattering parameters were only assigned to the planes we considered to be mainly irregular: the metal mesh ceiling and walls, the seats, and the ceiling of the stage with all the lighting. The metal mesh has a very wavy pattern to it, so we decided to go with the ”trapezoidal boxes (studio ceiling) “Round Robin III” (after (Bork 2005a))” which resembled waves and also is stated to be used in ceilings. The seats received the ”Theatre audience” scattering coefficients, as they will likely include an audience, while the ceiling of the stage was given the ”Wooden boxes of various sizes, random pattern, average h/a = 0.5 ” scattering parameters because lights are somewhat rectangular and have irregular sizes and orientations.

A list of the materials and assigned acoustic properties is shown below:

Figure 12: Absorption Coefficients

Material	125Hz	250Hz	500Hz	1000Hz	2000Hz	4000Hz
Concrete floor	0.01	0.03	0.05	0.02	0.02	0.02
Seats	0.44	0.60	0.77	0.89	0.82	0.70
Metal panels	0.59	0.80	0.82	0.65	0.27	0.23
Stage Wall	0.02	0.02	0.03	0.03	0.04	0.05
Stage Wood	0.10	0.07	0.06	0.06	0.06	0.06
Podium Wood	0.18	0.12	0.10	0.09	0.08	0.07
Glass	0.10	0.05	0.04	0.03	0.03	0.03
Stage ceiling	0.02	0.02	0.03	0.03	0.04	0.05

Figure 13: Reflection Coefficients

Material	125Hz	250Hz	500Hz	1000Hz	2000Hz	4000Hz
Metal mesh	0.13	0.56	0.95	0.95	0.95	0.95
Seats	0.30	0.50	0.60	0.60	0.70	0.70
Stage Ceiling	0.05	0.05	0.15	0.40	0.70	0.90

2.3 Results

Running simulations in Catt, we determined the T30 decay time, the early decay time (EDT), and the Speech Clarity (C80) as functions of frequency. Figure 14 shows the 30dB decay time as a function of frequency. The plot shows that at 1kHz, the decay time is the highest, at 1.8 seconds. As frequency increases or decreases the decay time goes down. High frequencies have the lowest decay times going down to .4 seconds, while lower frequencies stay at or above 1 second.

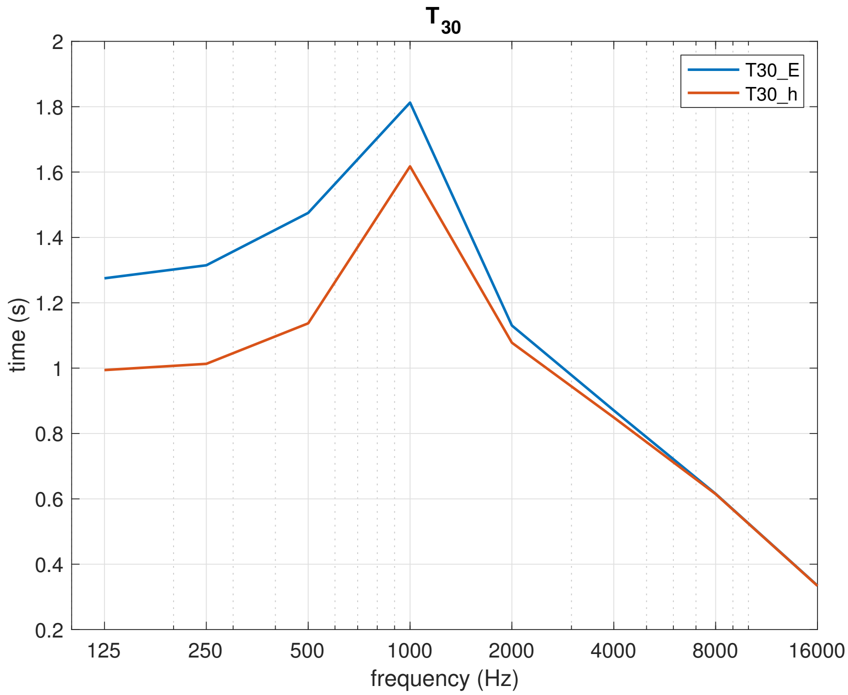

Figure 14: T_{30} Decay Time, Receiver 1

Figure 15 shows the early decay time as a function of frequency. The plot looks very similar to that of the T_{30} decay time, with a peak at 1kHz. However, there is a local minimum at 500Hz of .8 seconds, before increasing again at 125 to 1 second. This could be caused by some frequency specific absorption close to the source.

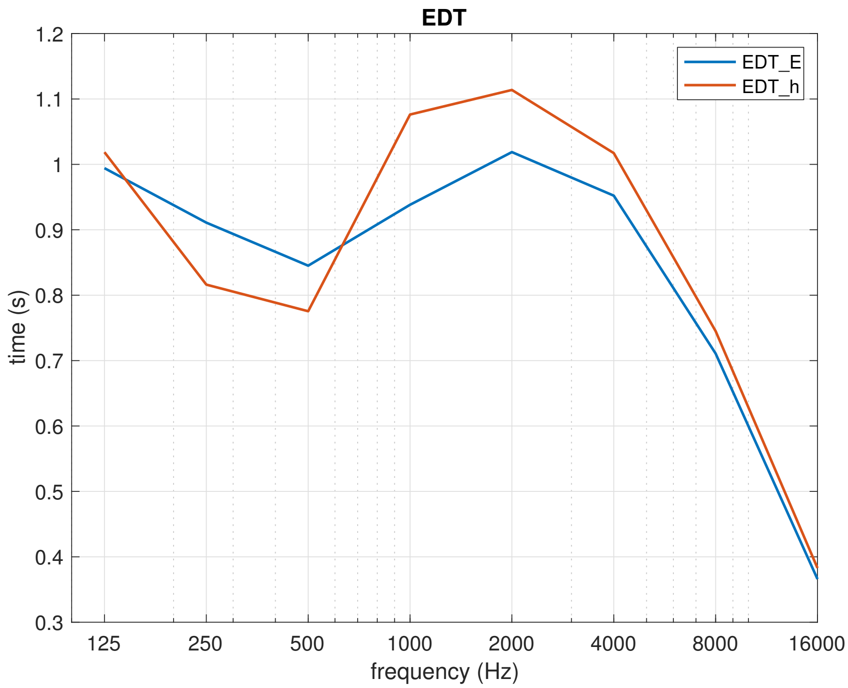

Figure 15: Early Decay Time, Receiver 1

The C80 speech clarity index can be seen in 16. We can see that clarity is lowest in the mid frequency range around 1-4kHz, going up at higher frequencies. This would be congruent with the high decay time found in the other two plots.

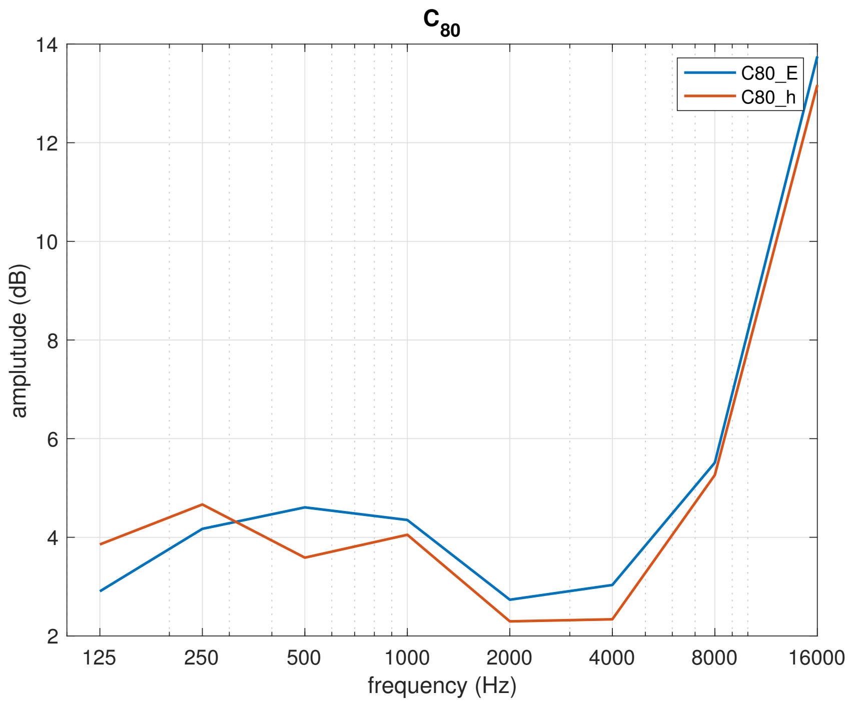

Figure 16: C_{80}, Receiver 1

However, we would argue that voices are very clear and well represented in Rose Auditorium. we wonder if the data is inaccurate due to the choice of material. The walls and ceiling are predominantly covered with the metal mesh which both visibly obscures what is behind it, and also may or may not add it's own acoustic properties. In that sense, the choice of absorption and scattering parameters for the walls and ceiling may be far from a good representation.

Lastly, we decided to look at the difference in the plots between all the receivers. Figures 17, 18, and 19 show the how the metrics change at various receivers. Even though both receiver 1 and 2 have a spike in T_{30} decay time, all receivers have a consistent C_{80} clarity index. This indicates that the room is well suited for listeners at any position.

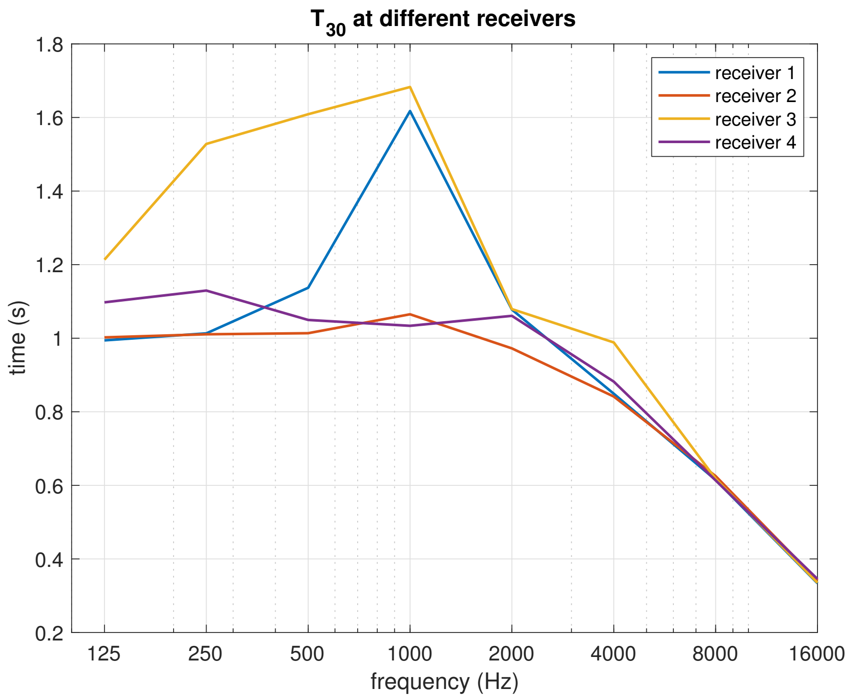

Figure 17: T_{30}, All Receivers

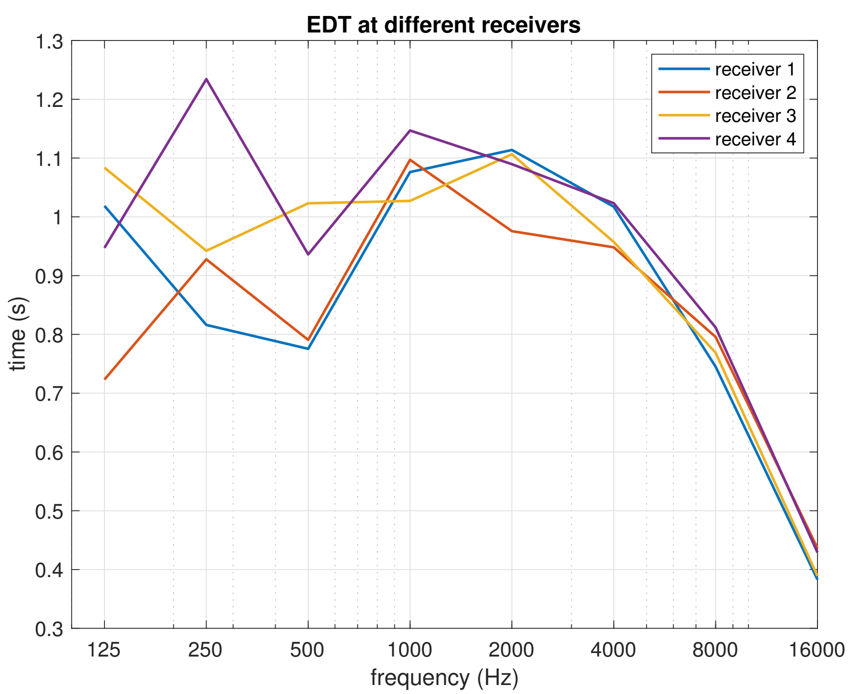

Figure 18: EDT, All Receivers

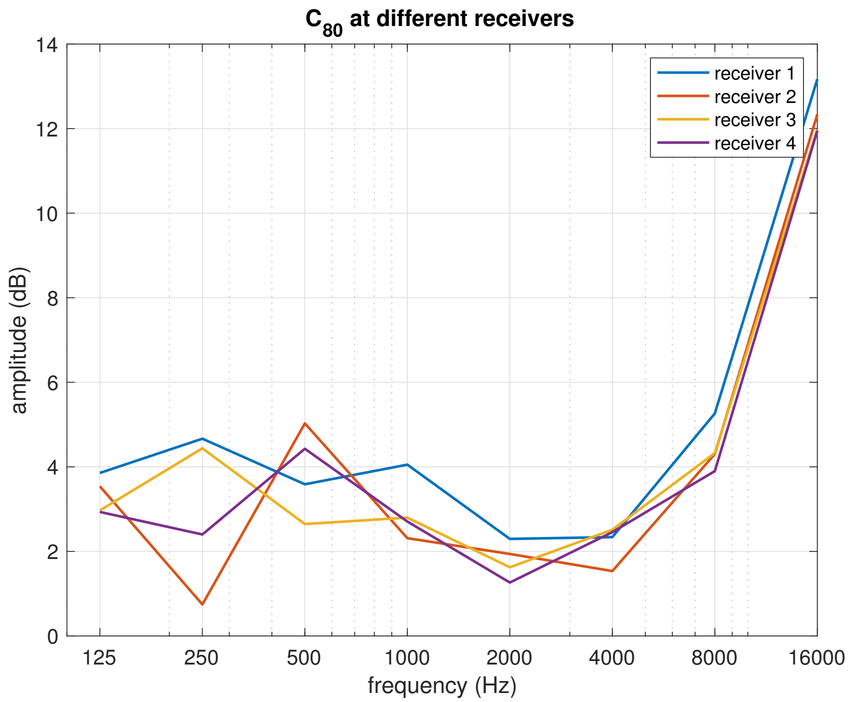

Figure 19: C_{80}, All Receivers

3 Acoustic Measurements

3.1 Receiver Locations

The receivers for this data are placed as described in the table below (figure 20) as well as in figure 21.

Figure 20: Measurement Locations for Actual Data

Number	Dist. to S [m]	Ref. 1	Dist. to Ref. 1 [m]	Ref. 2	Dist. to Ref. 2 [m]
S	0	Back Wall	3.447	Table	2.998
R1	6.02	Stage Left	5.847	Stage Right	6.203
R2	7.667	Stage Right	1.778	Table	6.048
R3	14.896	Stage Right	4.192	Back	2.56
R4	9.015	Stage Left	7.283	Stage Right	6.126

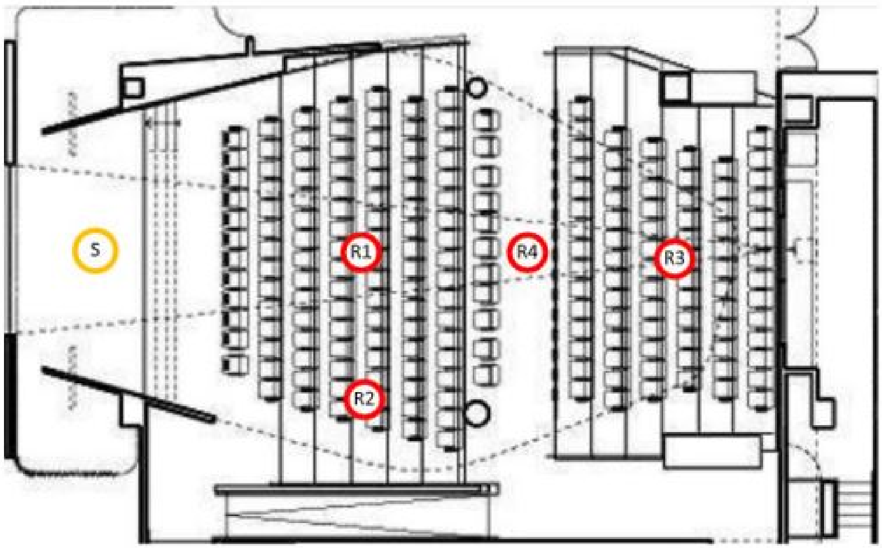

Figure 21: Source and Receiver Locations

3.2 Measurement Analysis

Figures 22, 23, and 24 show the T_{30}, EDT, and C_{80} calculated from the actual data.

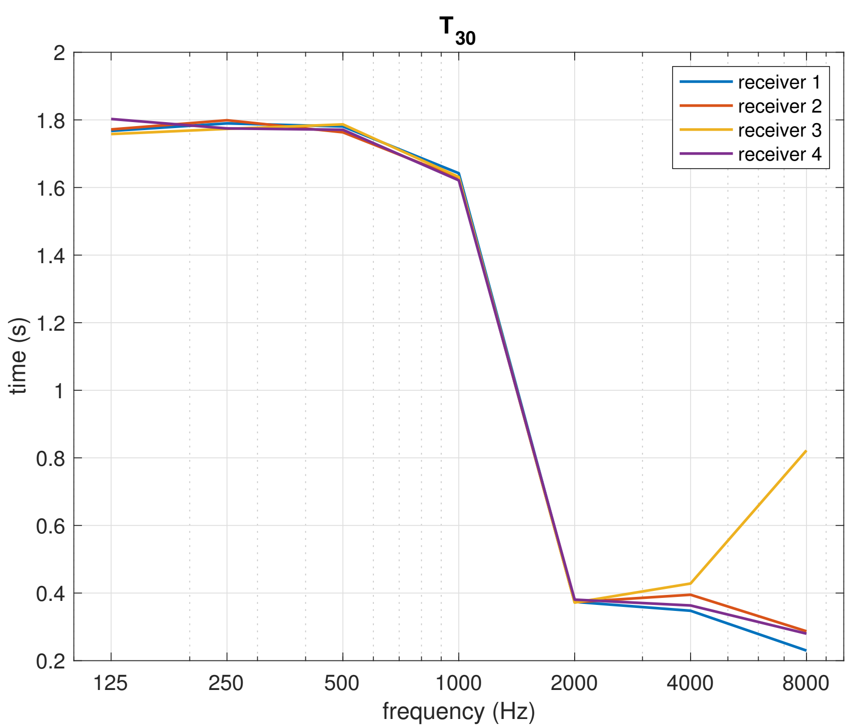

Figure 22: Actual T_{30}, All Receivers

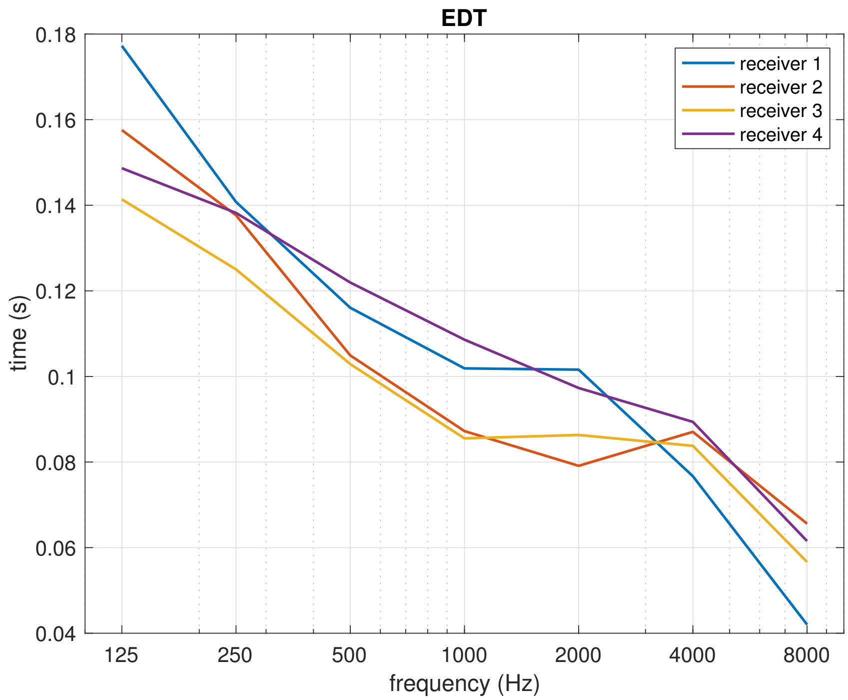

Figure 23: Actual EDT, All Receivers

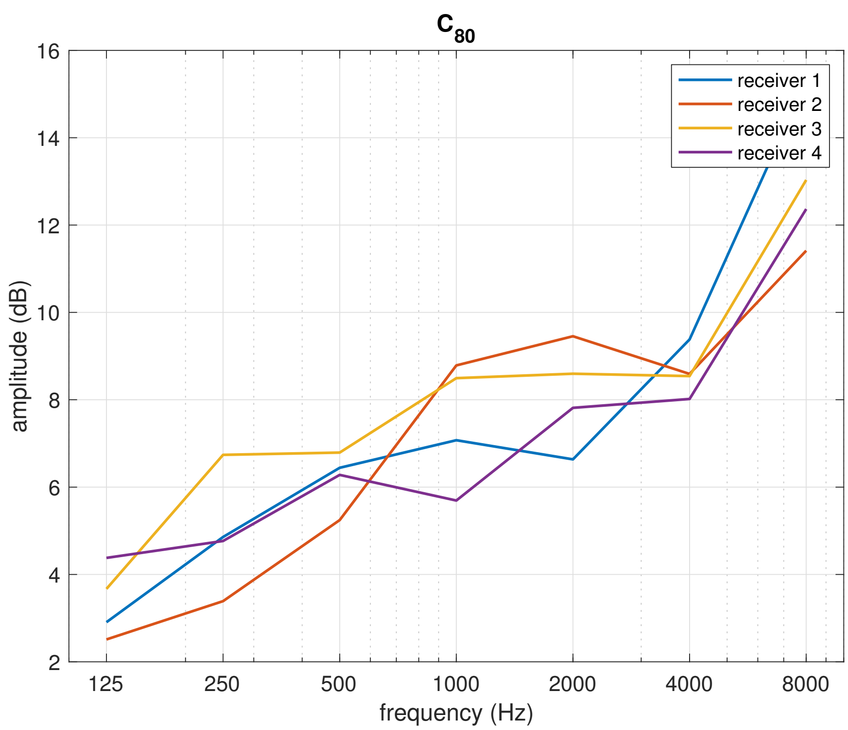

Figure 24: Actual C_{80}, All Receivers

For the C_{80} data, we time windowed the impulse response, and then filtered the time windowed data through octave band filters to produce the results. However, for T_{30} and EDT we couldn't do this, as we couldn't know where to time window prior to octave band filtering. Therefore, we octave band filtered the impulse response, and then calculated T_{30} and EDT for the individual frequency bands. This created some error in the calculated response times because the filters introduced some time domain spreading to the filtered impulse response. Lastly, there was some noise in the data due to a clipping audio amplifier, although we tried synchronous averaging to remove this noise, we couldn't remove all of it.

4 Model Calibration and Recommendations

4.1 Comparison and Model Calibration

The model and actual data are compared. To simplify the process, we compared the T_{30}, C_{80}, and EDT of receiver 1 only. The results show that for T_{30}, the model predicts a lower value than in actuality for lower frequencies and for higher frequencies it predicts more reverb than it should. The C_{80} plot shows that the model predicts higher values than are actually the case throughout all frequency bands. The EDT plot shows again that the model predicts higher values than are actually observed.
Based on these observations from before, we saw that we needed to increase the T_{30} for lower frequencies, decrease it for higher frequencies, and have the C_{80} increase more as frequency increases. For the EDT, we just need to decrease the overall magnitude throughout all frequencies. To decrease T_{30} for the lower frequencies we lowered the absorption value at these frequencies, and to increase T_{30} at higher frequencies we decreased the absorption coefficients. Ass for C_{80}, we lowered some of the scattering coefficients at higher frequencies.
To do this, we looked at our surface properties and decided to only change ones we were not sure about, namely the metal mesh and the seating. The rest of the surfaces we think that the properties chosen were very close to what is actually in rose, so they were left unchanged. The T_{30}, C_{80}, and EDT of the model before editions, after the editions, and the actual data are superimposed in figures 25, 26, 27.

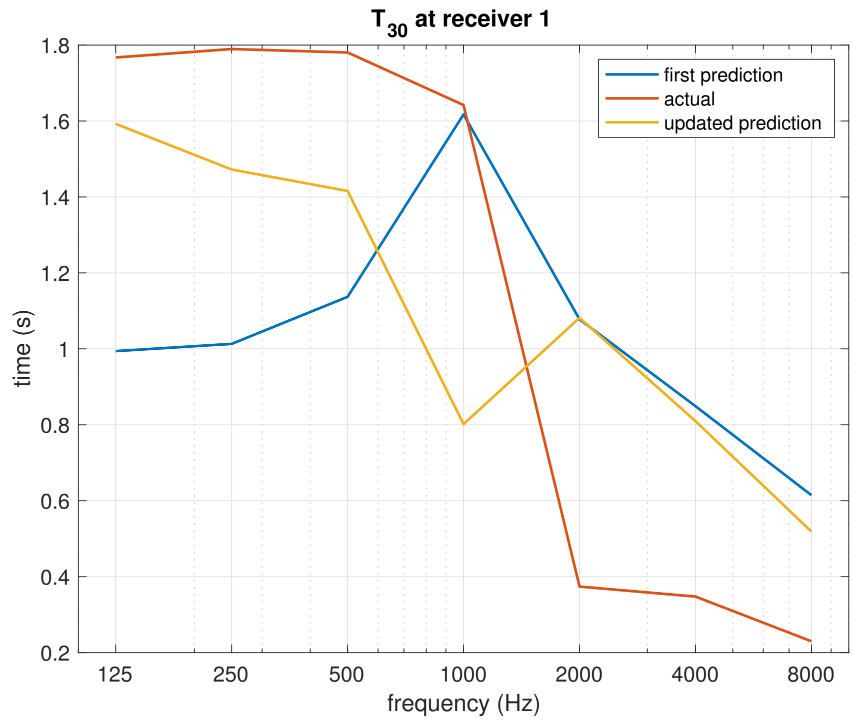

Figure 25: Model and Actual T_{30} for Receiver 1

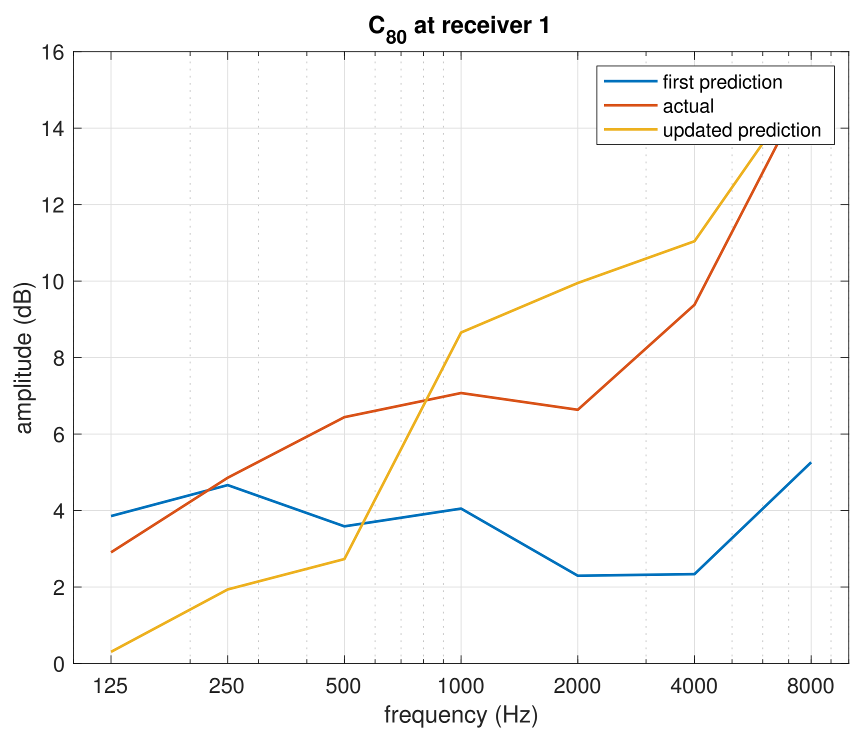

Figure 26: Model and Actual C_{80} for Receiver 1

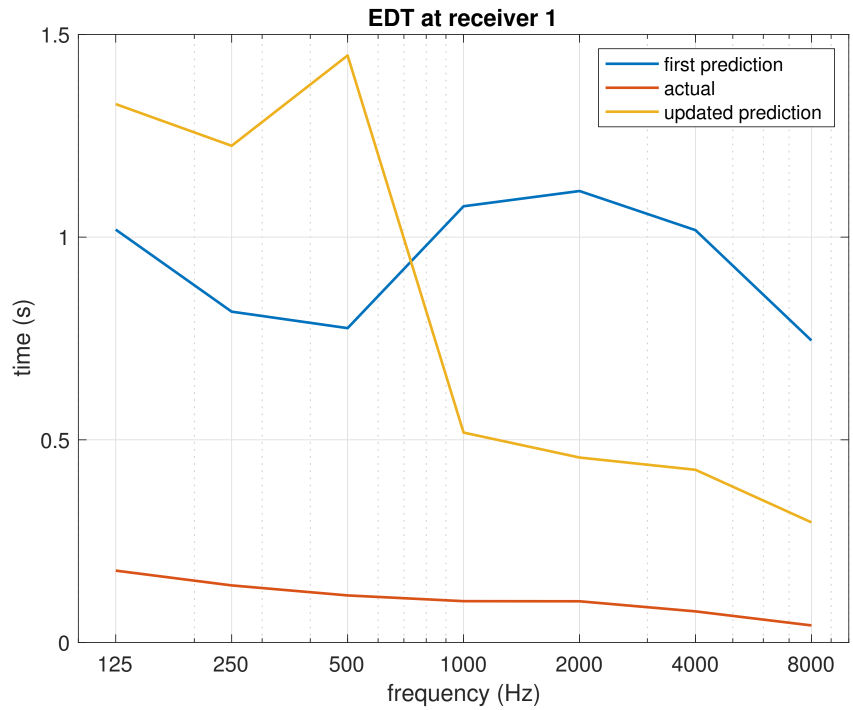

Figure 27: Model and Actual EDT for Receiver 1

One can see that the changes to the model brought it closer to the taken measurements. There is still room to go in this calibration, but since the model itself has restrictions we decided that the changes made are more than sufficient.

4.2 Recommendations to Improve Rose

We chose to decrease the reverb time for the Rose Auditorium. To do this we changed the absorption parameters of the seats and the metal, as they are a majority of the surfaces in our model. The new absorption parameters are seen in figure 28. The resulting T_{30}, C_{80}, and EDT are seen in figures 29, 30, and 31 respectively. The EDT and T_{30} at low frequencies was lowered, to improved speech clarity by decreasing the excess amount of bass present in the room. Additionally, this also improved the clarity of lower frequencies, as seen in figure 30. We presume that this would help in the use of this room for musical presentations where music was being played. (Although we notice that currently there are 0 subwoofers in Rose Auditorium, and as such, bass reproduction is abysmal anyways.)

Figure 28: Modified Absorption Coefficients

Material	125Hz	250Hz	500Hz	1000Hz	2000Hz	4000Hz
Seats	0.70	0.80	0.75	0.80	0.90	0.85
Metal panels	0.70	0.80	0.75	0.70	0.90	0.85

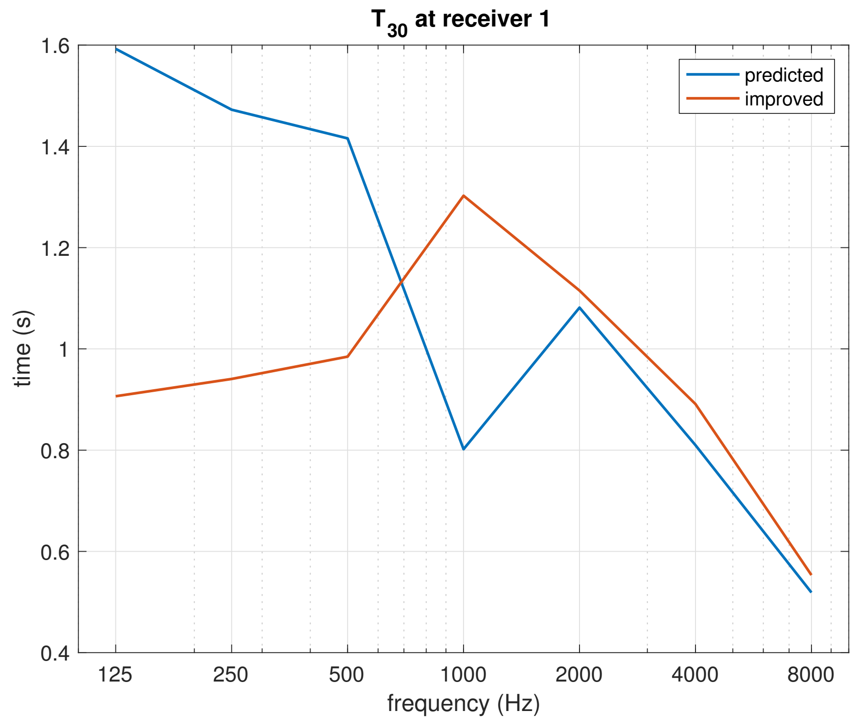

Figure 29: Updated and improved T_{30} for Receiver 1

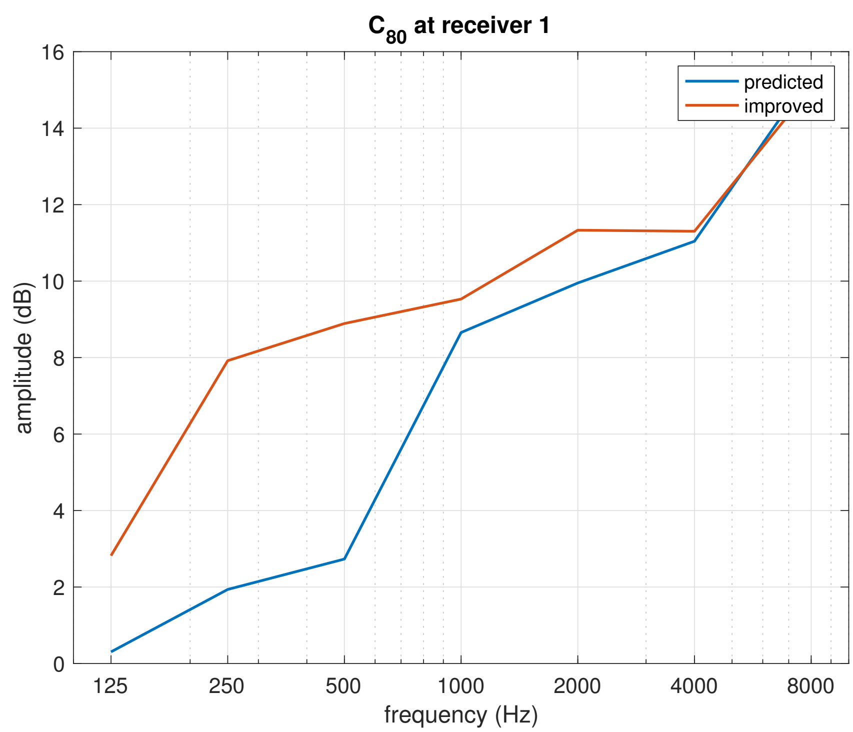

Figure 30: Updated and improved C_{80} for Receiver 1

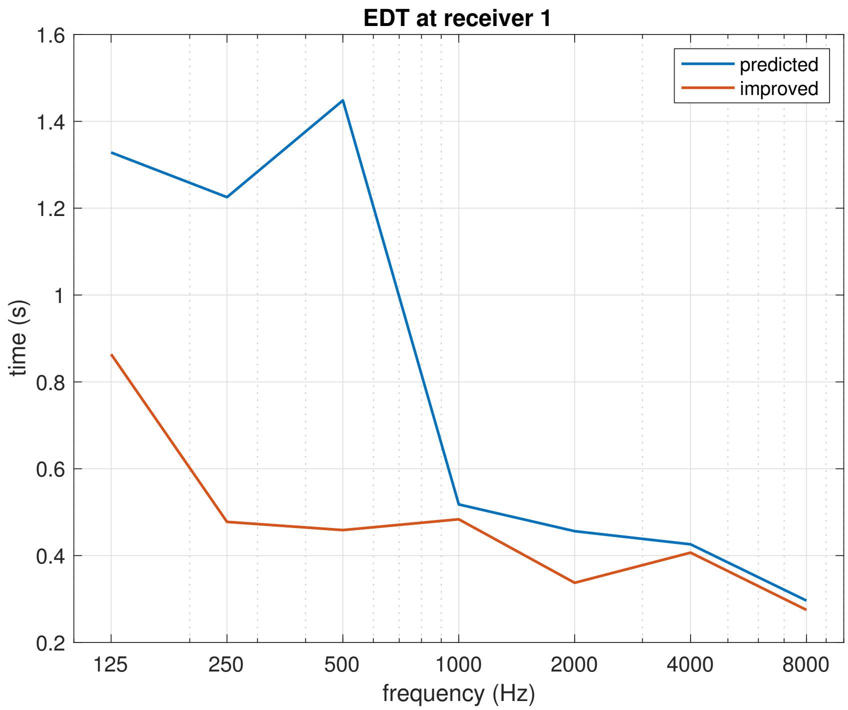

Figure 31: Updated and improved EDT for Receiver 1

5 Appendix I- MATLAB Code

We used the following MATLAB code to analyize the real world data.

```
close all; clc; clear all;

[r1, fs] = audioread('R1.wav');
r2 = audioread('R2.wav');
r3 = audioread('R3.wav');
r4 = audioread('R4.wav');

[b,a,cents] = create_octave_filt(fs,3);
m = length(cents);

L = length(r1);

T = 4;
Tp = T/2;
dt = 1/fs;

Np = fs*Tp;
tp = (0:Np-1)*dt; tp=tp.';


% create your LOGARITHMIC sine sweep and averaging
f1 = 20; f2 = 20000;
phi = f1*((f2/f1).^(tp/Tp))*(Tp/log(f2/f1));
LogChirp = sin(2*pi*phi);



r1ave = mean(reshape(r1,Np,L/Np),2);
r2ave = mean(reshape(r2,Np,L/Np),2);
r3ave = mean(reshape(r3,Np,L/Np),2);
r4ave = mean(reshape(r4,Np,L/Np),2);


Chirp = fft_new(LogChirp,fs);
R1 = fft_new(r1ave,fs);
R2 = fft_new(r2ave,fs);
R3 = fft_new(r3ave,fs);
R4 = fft_new(r4ave,fs);

% frequency responses
```

```
H1 = R1 ./ Chirp;
H2 = R2 ./ Chirp;
H3 = R3 ./ Chirp;
H4 = R4 ./ Chirp;

% impulse responses
h1 = ifft_new(H1,fs);
h2 = ifft_new(H2,fs);
h3 = ifft_new(H3,fs);
h4 = ifft_new(H4,fs);




%% C80
ms80 = .08 * fs; % 80 ms in samples

[~,peak1] = findpeaks( (abs(h1)>5).*abs(h1) );
peak1 = peak1(1); % first peak above threshold
[~,peak2] = findpeaks( (abs(h2)>5).*abs(h2) );
peak2 = peak2(1);
[~,peak3] = findpeaks( (abs(h3)>5).*abs(h3) );
peak3 = peak3(1);
[~,peak4] = findpeaks( (abs(h4)>5).*abs(h4) );
peak4 = peak4(1);


r1_b80 = h1(peak1:peak1+ms80); % peak to 80 ms
r1_a80 = h1(peak1+ms80:end); % 80ms after peak to end
r2_b80 = h2(peak2:peak2+ms80);
r2_a80 = h2(peak2+ms80:end);
r3_b80 = h3(peak3:peak3+ms80);
r3_a80 = h3(peak3+ms80:end);
r4_b80 = h4(peak4:peak4+ms80);
r4_a80 = h4(peak4+ms80:end);

 % octave band filtering time windowed results
r1_b80_oct = apply_filtbank(b,a,r1_b80);
r1_a80_oct = apply_filtbank(b,a,r1_a80);
r2_b80_oct = apply_filtbank(b,a,r2_b80);
r2_a80_oct = apply_filtbank(b,a,r2_a80);
r3_b80_oct = apply_filtbank(b,a,r3_b80);
r3_a80_oct = apply_filtbank(b,a,r3_a80);
r4_b80_oct = apply_filtbank(b,a,r4_b80);
```

```
r4_a80_oct = apply_filtbank(b,a,r4_a80);

% calculating c_80
c801 = 10*log10( sum(r1_b80_oct.^2) ./ sum(r1_a80_oct.^2) );
c802 = 10*log10( sum(r2_b80_oct.^2) ./ sum(r2_a80_oct.^2) );
c803 = 10*log10( sum(r3_b80_oct.^2) ./ sum(r3_a80_oct.^2) );
c804 = 10*log10( sum(r4_b80_oct.^2) ./ sum(r4_a80_oct.^2) );



%% T30

% schroder curves for each reciever

h1_oct = apply_filtbank(b,a,h1);
h2_oct = apply_filtbank(b,a,h2);
h3_oct = apply_filtbank(b,a,h3);
h4_oct = apply_filtbank(b,a,h4);


sr1 = 10*log10(flipud( cumtrapz(tp, flipud(h1_oct.^2) ) ) );
sr2 = 10*log10(flipud( cumtrapz(tp, flipud(h2_oct.^2) ) ) );
sr3 = 10*log10(flipud( cumtrapz(tp, flipud(h3_oct.^2) ) ) );
sr4 = 10*log10(flipud( cumtrapz(tp, flipud(h4_oct.^2) ) ) );

sr1m = max(sr1);
sr2m = max(sr2);
sr3m = max(sr3);
sr4m = max(sr4);

% finds first point that starts to tip down
% they look all mostly the same (per octave band) so I just
    pick one
n = 3;
plat1 = find( abs(sr1(:,n) - sr1m(n)) < .01);
plat1 = plat1(end);
plat2 = find( abs(sr2(:,n) - sr2m(n)) < .01);
plat2 = plat2(end);
plat3 = find( abs(sr3(:,n) - sr3m(n)) < .01);
plat3 = plat3(end);
plat4 = find( abs(sr4(:,n) - sr4m(n)) < .01);
plat4 = plat4(end);

t301 = zeros(1,m);
```

```
t302 = zeros(1,m);
t303 = zeros(1,m);
t304 = zeros(1,m);


for i = 1:m

    [~,ind11] = min(abs(sr1(:,i)-max(sr1(:,i))+5));
    [~,ind12] = min(abs(sr1(:,i)-max(sr1(:,i))+35));

    [~,ind21] = min(abs(sr2(:,i)-max(sr2(:,i))+5));
    [~,ind22] = min(abs(sr2(:,i)-max(sr2(:,i))+35));

    [~,ind31] = min(abs(sr3(:,i)-max(sr3(:,i))+5));
    [~,ind32] = min(abs(sr3(:,i)-max(sr3(:,i))+35));

    [~,ind41] = min(abs(sr4(:,i)-max(sr4(:,i))+5));
    [~,ind42] = min(abs(sr4(:,i)-max(sr4(:,i))+35));

    t301(i) = (ind12 - ind11) * dt;
    t302(i) = (ind22 - ind21) * dt;
    t303(i) = (ind32 - ind31) * dt;
    t304(i) = (ind42 - ind41) * dt;

end

% EDT

edt1 = zeros(1,m);
edt2 = zeros(1,m);
edt3 = zeros(1,m);
edt4 = zeros(1,m);
for i = 1:m

    [~,ind1] = min(abs(sr1(:,i)-max(sr1(:,i))+10));

    [~,ind2] = min(abs(sr2(:,i)-max(sr2(:,i))+10));

    [~,ind3] = min(abs(sr3(:,i)-max(sr3(:,i))+10));

    [~,ind4] = min(abs(sr4(:,i)-max(sr4(:,i))+10));

    edt1(i) = (ind1 - plat1) * dt;
    edt2(i) = (ind2 - plat2) * dt;
```

```
    edt3(i) = (ind3 - plat3) * dt;
    edt4(i) = (ind4 - plat4) * dt;

end
```