Naive Room Response Deconvolution

By:
William Howison
Chris Lamontagne
Bryce Luna
David Newell

Naive Room Response Deconvolution

By:
William Howison
Chris Lamontagne
Bryce Luna
David Newell

Online:
< http://cnx.org/content/col10318/1.5/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: December 14, 2005
PDF generated: October 30, 2009
For copyright and attribution information for the modules contained in this collection, see p. 12.

Table of Contents

Chapter 1

Introduction

1.1 Introduction to Naive Room Response Deconvolution[1]

Every room responds differently to an input sound. This fact is due to the reverberations of sound waves off surfaces in the room. The exact response governed by the geometry and structure of that particular room. Even for rooms with the same dimensions, different surfaces will cause the noise to reflect more or less loudly because different materials have different reflection coefficients. A higher reflection coefficient means less energy is absorbed by the wall, and hence more of the sound is reflected off the wall. This can easily cause problems when recording or playing music in an enclosed space. The frequency characteristics of the room are important when sound quality is a concern; audio engineers spend significant amounts of time characterizing the acoustics of a room for the ideal placement of audio sources.

The sound characteristics of the room can be roughly modeled as a linear time-invariant system. Just like any system, the room has an impulse response which is possible to measure by playing an approximate sound impulse. An impulse is played in the room and recorded using a standard microphone. Since the enclosure can be modeled as an LTI system, the frequency response of the room is simply the FFT of this recording, provided there is no other noise interfering with the system.

Given the impulse response of the room, it is possible to predict the output of any signal into the room when given the input. This prediction is possible by simply multiplying the frequency response of the system with the FFT of the output. It is only natural to wonder if this process is reversible: Can we find the input to a room if we record the output? This seemingly complicated process is very easy using deconvolution. Because the model of the room is an LTI system we can take the inverse of the frequency response and multiply by the transformed output to get the frequency domain input. We can then apply the inverse transform to this result to recreate the input signal.

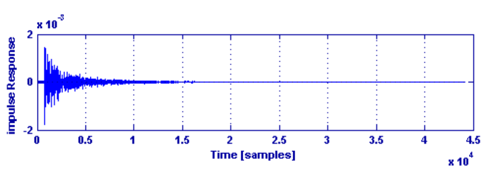
[1]This content is available online at <http://cnx.org/content/m13151/1.2/>.

Chapter 2

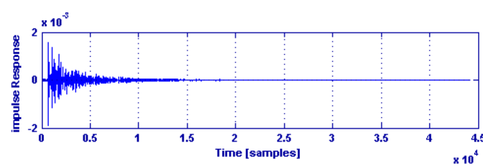
Methods and Data

2.1 Naive Deconvolution Theory[1]

There are many characteristics of a room that determine the impulse response of a room. The physical dimensions of the room and the wall surfaces are crucial in predicting how sound reacts. Using basic properties of geometry, we can predict the paths that sound waves will travel on, even as they bounce off walls. The walls themselves have certain reflection coefficients that describe the power of the reflected signal with respect to the signal in contact with the wall. So it appears that with the dimensions of the room and the reflection coefficients of the walls in the room it is possible to generate an impulse response for that room. Using a simple tape measure we recorded the height, width, and length of Duncan 1075 and a Will Rice dorm room, and used a MATLAB program called Room Impulse Response to find the approximate impulse response of these two rooms. We decided to take two samples in each room, leaving us with four theoretical impulse responses.

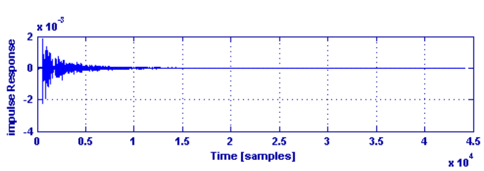


(a) Theoretical Impulse in Duncan - Left

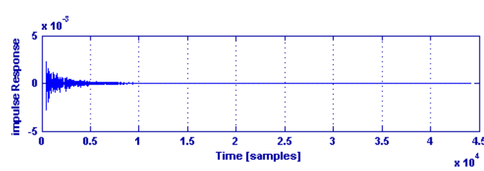


(b) Theoretical Impulse in Duncan - Right

Figure 2.1



(a) Theoretical Impulse in Will Rice - Left



(b) Theoretical Impulse in Will Rice - Right

Figure 2.2

[1]This content is available online at <http://cnx.org/content/m13194/1.3/>.

Clearly these will not be incredibly accurate, as they assume a completely rectangular, and empty, room. Neither of these rooms were completely rectangular, and they were also not empty. However, this gives us a good approximation of the impulse response. The signals decay significantly as time increases, which is expected. When we record the actual response using an approximate impulse, this data will help determine if we have an accurate measurement.

Once we have the impulse response of each room, we must find an appropriate signal to deconvolve. We chose a piano tune, as a piece of music should have a more simple frequency response than speech. After recording the impulse response and the input, we theoretically have enough data to reconstruct the signal using deconvolution. The recorded output is the convolution of the input with the system.

$$y(t) = x(t) * h(t)$$

The recorded output has a frequency spectrum defined by

$$Y(\text{jw}) = X(\text{jw})\,H(\text{jw})$$

Using simple algebra, we can solve for the input frequency coefficients:

$$X(\text{jw}) = Y(\text{jw})\,/H(\text{jw})$$

We have H(jw), the impulse response, and Y(jw), the FFT of the recorded signal. Thus we can find X(jw), the frequency spectrum of the input signal, and by taking the inverse FFT we are left with the input signal x(t).

2.2 Recording the Impulse Response of a Room[2]

After obtaining the theoretical data we moved on to the measurements of the impulse response in both rooms and audio test trials. The equipment that was used for the measurements was as follows: a laptop computer, a pre amplifier, amplifier, speaker, and omnidirectional microphones. The microphones and source were placed in accordance with the locations we specified in the simulation software, given in rough estimate by the diagram.

[2] This content is available online at <http://cnx.org/content/m13192/1.3/>.

Recording Setup

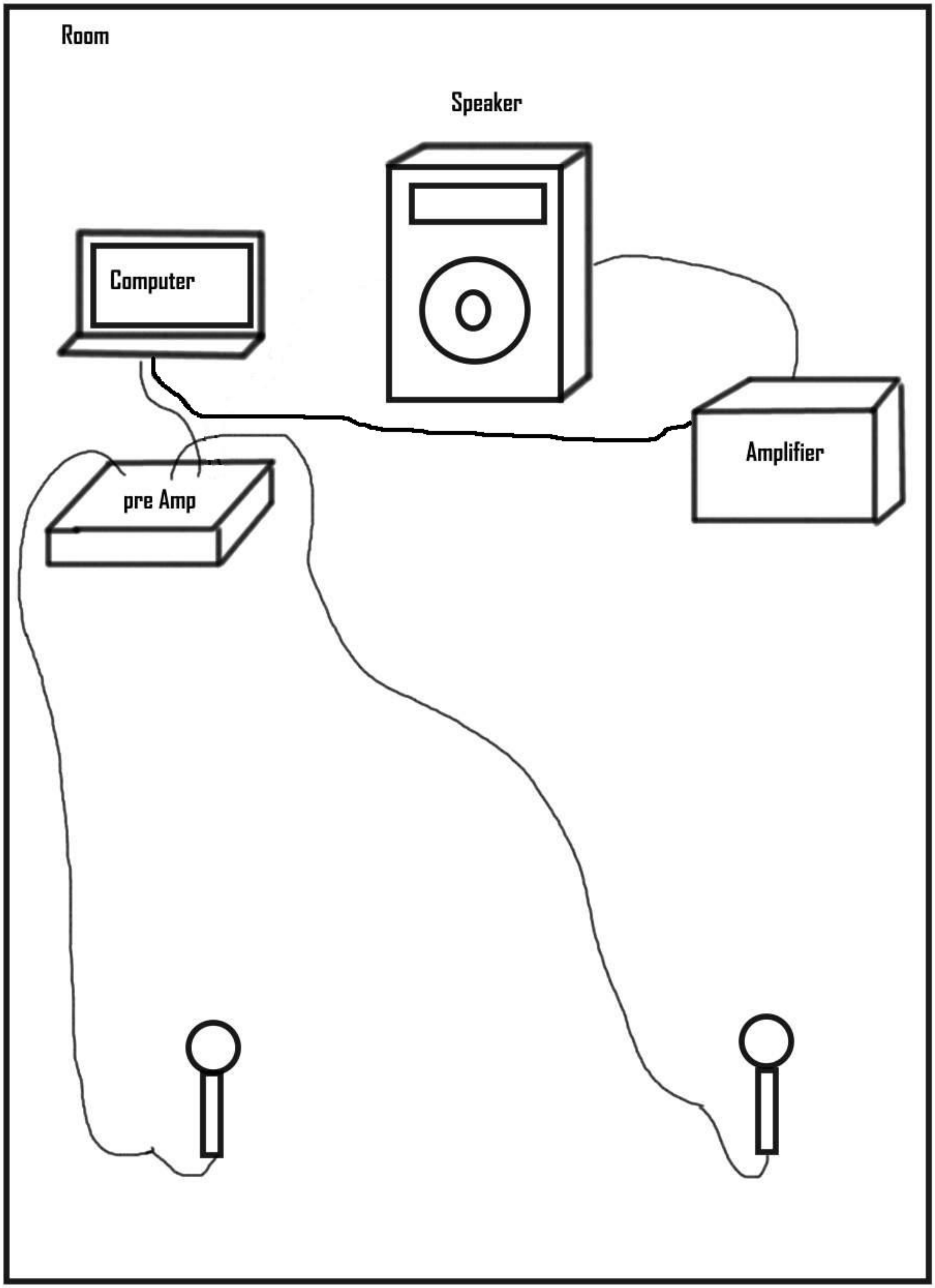


Figure 2.3

The use of the laptop was necessary to not only record the test impulse response but as the source of the impulse and test audio. To avoid the difficulty of making an impulse sound physically, using a "clapper," we generated an impulse digitally on the laptop with MATLAB. We used a piano tune as the input hoping to become slightly more cultured while working on our project. In order to properly record a significant number of reflections, very loud impulses and inputs were played. This most likely resulted in clipping, but was necessary to determine accurate responses.

This is an unsupported media type. To view, please see http://cnx.org/content/m13192/latest/Impulse.wav

This is an unsupported media type. To view, please see http://cnx.org/content/m13192/latest/Input.wav

The impulse and input audio are used to perform the deconvolution experiment. The impulse and the input signal were played in each of the rooms and the room responses to both of these were recorded in .wav format. We recorded both responses in two rooms, Duncan 1075 and a Will Rice College dorm room. We chose Duncan 1075 because it was the ELEC 301 classroom this year, and a generic dorm room should help all of the audiophile students get the best sound quality possible. We recorded two samples in each room, in case we found a null zone in one of the locations. The results are displayed on the next page.

2.3 The Effectiveness of Naive Room Response Deconvolution[3]

After playing both the impulse responses and our input signal, and recording the output for two points in each room, we were anxious to deconvolve our recorded input and get a perfect replica of the input signal. All of our signals were recorded in .wav format, which is lossless, so we didn't lose any important data due to audio compression. The results of our deconvolution in .wav format are below.

This is an unsupported media type. To view, please see http://cnx.org/content/m13191/latest/Input.wav

This is an unsupported media type. To view, please see http://cnx.org/content/m13191/latest/DeconvolvedDuncanInputResponseLeft.wav

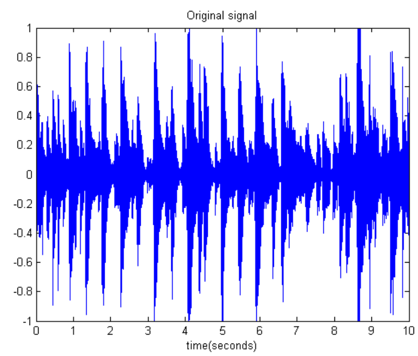
This is an unsupported media type. To view, please see http://cnx.org/content/m13191/latest/DeconvolvedDuncanInputResponseRight.wav

This is an unsupported media type. To view, please see http://cnx.org/content/m13191/latest/DeconvolvedWillRiceInputResponseLeft.wav

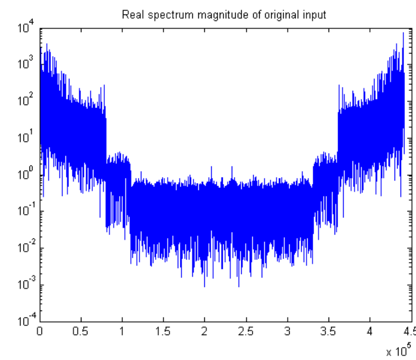
This is an unsupported media type. To view, please see http://cnx.org/content/m13191/latest/DeconvolvedWillRiceInputResponseRight.wav

[3]This content is available online at <http://cnx.org/content/m13191/1.3/>.

We were not able to receive a perfect input after deconvolution due to a number of reasons, which are discussed in our conclusion. However, we can clearly tell that the signal exists in all of these recordings. This would imply that our theory is correct to a degree, while our implementation must take into account some things we ignored. The following are a selection of MATLAB plots describing the deconvolved signals.

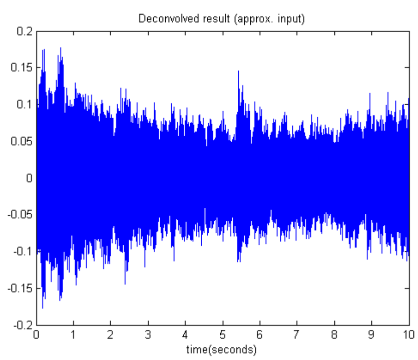


(a) Input - Time Spectrum

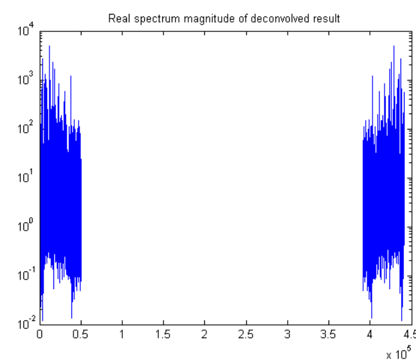


(b) Input - Frequency Spectrum

Figure 2.4

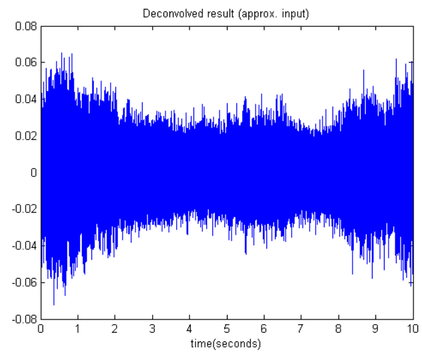


(a) Deconvolved Result in Duncan - Time

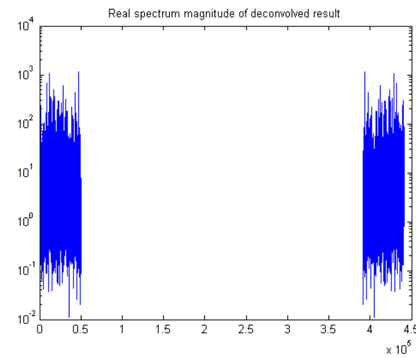


(b) Deconvolved Result in Duncan - Frequency

Figure 2.5



(a) Deconvolved Result in Will Rice- Time



(b) Deconvolved Result in Will Rice - Frequency

Figure 2.6

We will deal with the negatives before discussing the positives. The large gap in the frequency spectrums are due to a filter in the frequency domain. The recorded signal has very low signal strength in the middle frequencies. In an attempt to remove noise from the deconvolved signal we removed these frequencies. This is the cause of our lose in signal strength in the time domain. The deconvolved results here are still overwhelmed by enough noise to blanket out the input. However, the input exists in the signal, which is apparent when listening to the .wav files. Where the deconvolved input frequency spectrum is not removed it resembles the frequency response of the input signal. The audio data shows that while our results are far from ideal, we were able to deconvolve the recorded impulse and return the input signal.

Chapter 3

Conclusion

3.1 Problems and Future Considerations in Naive Room Response Deconvolution[1]

We began data comparison by observing the differences between the theoretical impulse responses and actual impulse responses in each room. None of the four actual responses were similar to the theoretical responses. The differences could be due the non ideality of the rooms, as the rooms were neither perfectly rectangular nor empty, while Room Demo Response assumed both of these conditions. Clipping was also neglected in the theoretical model. Clipping equalizes all signals above a certain threshold determined by the soudn card in the laptop; this non-linear effect removed information from the signal in such a way that the lost information was unrecoverable by our Fourier analysis. Commercial applications of room impulse response measurements, such as measuring the response from each seat in an orchestra hall, require a more robust theoretical model that accounts for objects in the room as well as the exact shape of a room. Such elaborate theoretical data was not necessary; we were able to access the rooms in question and find the impulse response through direct measurement.

The goal of signal deconvolution was to remove the room response on a recorded signal. However, this process also amplified the noise. Some of the noise was already prevalent in the recorded signal, as our microphones were sensitive enough to hear a group of people conversing outside with the door closed. Since the noise was already in the signal and was not entirely random it could not be easily filtered out. The deconvolution did reproduce the original signal; however the quality was significantly worse than the recorded signal. If we could record the impulse and input responses without noise, our method of deconvolution would be able to reproduce the original high quality recorded signal. Unfortunately this is not possible under normal conditions. Another attempt could also be made using a more complex deconvolution scheme, such as Wiener deconvolution. Ideally we could find a method that is either resistant to noise or removess noise entirely; this would immediately lead to better results. Perfect deconvolution would be useful in creating a clean input signal given a recorded signal regardless of recording environment. Naive deconvolution only works well with noiseless signals. Future applications of our theory would have to use more complex deconvolution methods.

3.2 Authors' Contact Information[2]

In alphabetical order the creators of this project were:

Howison, William. Rice University, Lovett College. Junior ECE student. **willhow at rice.edu**
Lamontagne, Chris. Rice University, Will Rice College. Junior ECE student. **clamonta at rice.edu**
Luna, Bryce. Rice University, Lovett College. Senior ECE/Physics student. **brycel at rice.edu**

[1]This content is available online at <http://cnx.org/content/m13190/1.2/>.
[2]This content is available online at <http://cnx.org/content/m13198/1.1/>.

Newell, David. Rice University, Lovett College. Junior ECE student. **beta at rice.edu**

3.3 Room Response Deconvolution M-Files[3]

deconvolve.m[4] - This is the m-file that performs the deconvolution of the impulse response and recorded output.

impulse.m[5] - This m-file generates an approximate impulse that can be played.

sigcorrect.m[6] - This m-file synchronizes the actual input and recorded input signals and makes them the same length.

[3]This content is available online at <http://cnx.org/content/m13199/1.1/>.
[4]http://www.owlnet.rice.edu/~willhow/deconvolve.m
[5]http://www.owlnet.rice.edu/~willhow/impulse.m
[6]http://www.owlnet.rice.edu/~willhow/sigcorrect.m

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Collection: *Naive Room Response Deconvolution*
Edited by: William Howison, Chris Lamontagne, Bryce Luna, David Newell
URL: http://cnx.org/content/col10318/1.5/
License: http://creativecommons.org/licenses/by/2.0/

Module: "Introduction to Naive Acoustic Deconvolution"
Used here as: "Introduction to Naive Room Response Deconvolution"
By: Chris Lamontagne, Bryce Luna
URL: http://cnx.org/content/m13151/1.2/
Page: 1
Copyright: Chris Lamontagne, Bryce Luna
License: http://creativecommons.org/licenses/by/2.0/

Module: "Naive Deconvolution Theory"
By: Chris Lamontagne, Bryce Luna
URL: http://cnx.org/content/m13194/1.3/
Pages: 3-4
Copyright: Chris Lamontagne, Bryce Luna
License: http://creativecommons.org/licenses/by/2.0/

Module: "Recording the Impulse Response of a Room"
By: Chris Lamontagne, Bryce Luna
URL: http://cnx.org/content/m13192/1.3/
Pages: 4-6
Copyright: Chris Lamontagne, Bryce Luna
License: http://creativecommons.org/licenses/by/2.0/

Module: "The Effectiveness of Naive Audio Deconvolution in a Room"
Used here as: "The Effectiveness of Naive Room Response Deconvolution"
By: Chris Lamontagne
URL: http://cnx.org/content/m13191/1.3/
Pages: 6-8
Copyright: Chris Lamontagne
License: http://creativecommons.org/licenses/by/2.0/

Module: "Problems and Future Considerations in Naive Room Response Deconvolution"
By: David Newell, Chris Lamontagne
URL: http://cnx.org/content/m13190/1.2/
Page: 9
Copyright: Chris Lamontagne
License: http://creativecommons.org/licenses/by/2.0/
Based on: Conclusion
By: David Newell
URL: http://cnx.org/content/m13150/1.3/

Module: "Authors' Contact Information"
By: William Howison
URL: http://cnx.org/content/m13198/1.1/
Pages: 9-10
Copyright: William Howison
License: http://creativecommons.org/licenses/by/2.0/

Module: "Room Response Deconvolution M-Files"
By: Chris Lamontagne
URL: http://cnx.org/content/m13199/1.1/
Page: 10

Naive Room Response Deconvolution

ELEC 301 project by William Howison, Chris Lamontagne, Bryce Luna, and David Newell. Given the output of a system and the system characteristics we can determine the input. We will determine the system characteristics of two rooms by playing an (approximate) impulse and recording the impulse response, and then we will play music into the same rooms and record the output. Using MATLAB we will deconvolve the output with the system response to determine a rough approximation of the input.