Room Response Estimation using U-Net Machine Learning Architecture

USC ECE Undergraduate Thesis

Yash Gupta
Ming Hsieh Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA
yjgupta@usc.edu

***Abstract*—Audio equipment has always struggled with correcting for adverse low-frequency modes within shoe box shaped rooms. This paper proposes a method to infer a room's magnitude response with a loudspeaker located in the center of a wall using a U-Net convolutional neural network model. The input for this network is a measurement from a microphone located close to the loudspeaker and the output is another measurement from the position of a listener in the center of the room. Data was generated using acoustic simulation software, then was used for supervised training with three models. One model had a low frequency magnitude response (LFMR) as the input and output, another had a room impulse response (RIR) as an input and output, and the third used RIR's but replaced some convolutional layers with fully connected layers. Results from the first and third model are unsatisfactory, but the second model shows promise despite inaccuracies.**

***Index Terms*—Room Acoustics, Machine Learning, U-Net**

I. Introduction

In recent years, consumer audio has moved away from large, bulky speakers and amplifier units towards smaller units relying on digital signal processing to provide better sound experiences. In either case, adverse room acoustics cause a significant degradation to the accuracy of the perceived sound at the position of the listener. Sounds below 250 Hz [1] present large scale changes in amplitude dependent on speaker and listener position. Audio devices currently use a calibration microphone located at the position of the listener to flatten magnitude response with a filter, however this requires the user go through an inconvenient calibration and setup process. There was previously no good way to predict the frequency response of a room using an integrated microphone on the speaker unit. Today, however, machine learning has enabled complex, unknown functions to be approximated and modeled at a low enough computational cost to implement in consumer hardware. For machine learning to be implemented, however, we must validate whether there is a convergent function that can approximate the frequency response of the listener's position in a variety of room sizes. This paper analyzes whether a convolutional neural net (CNN) machine-learning model can approximate the frequency response of a room given only the speaker's impulse response as measured by a microphone placed on a speaker unit in constrained conditions.

II. Literature Review

A. Least-Squares Reflection-Based Room Size Estimation

A similar problem, room size estimation based on Room Impulse Responses (RIRs), has been the subject of multiple research projects. [2] measures timing of first and second order acoustic reflections in a single RIR of a "shoe box" shaped room and infers the dimensions of the room using least-squares regression. A similar methodology is employed in [3], generalizing it to both rectangular and L-shaped rooms. [4] opts to use directional impulse responses and rotates them by set intervals to obtain a polar plot of the room from the first order reflections. [5] similarly uses a uniform array of loudspeakers and one microphone to estimate the geometry of a room. Each of these methods uses statistical inference and least-squares to calculate room geometry, indicating that there is a way to derive the general room geometry from an RIR.

B. Convolutional Neural Net Based Room Size Estimation

In addition to statistical methods, multiple papers have employed Convolutional Neural Net (CNN) architecture to estimate the relationship between RIR and room geometry. Most notably, [6] uses a fairly common CNN architecture of batch normalization layers followed by activation functions. It analyzes the RIR's in the time-domain.

C. U-Net Convolutional Neural Networks

Cited by 523 other papers, [7] shows audio source separation in the time-domain using a U-Net CNN architecture. This architecture features several convolutional encoding layers followed by downsampling layer. Convolutional decoding layers then get concatenated with a skip connection from the corresponding encoding layer and get upsampled. [8], citing [7], uses a U-Net in the frequency domain with spectrograms as inputs to separate instruments within music.

III. Methods

A. Formulation of Question

Our goal is to infer the room response measurement at the listeners location using the measurement from the integrated microphone within the simulated speaker unit. Literature from the subject of room acoustics and associated problems indicates that room size and impulse response both contain enough information to infer one another [3] [2]. Using this, we can infer the room geometry using a single measurement in one location, then, knowing the general location of both the input measurement and listener measurement, we can take both and compare the low-frequency magnitude response (LFMR) (<250 Hz) of each to see how well they match.

B. Proposed Models

To do this, I propose the use of a 3 different supervised-learning U-Net CNN architectures to directly relate the LFMR at the speaker position to the LFMR at the listener position. U-Net architectures are suitable for this task due to the use of both convolutional layers and skip connections. In this context, the convolutional layers will find local features such as peaks and troughs in the input, and the skip connections will allow the encoding layers to find global features in the input. Additionally, CNN layers have an order of magnitude less parameters than dense neural network layers, making them much more efficient on low-end hardware.

1) (LFMR model) The first model will input an LFMR from the simulated speaker location and estimate the LFMR at the listeners location.
2) (Time model) The second will input a length 1024 RIR at 1000 Hz and estimate the RIR at the listeners location.
3) (Dense Decoder model) The third is similar to the second model but will replace the decoders in the U-Net with dense layers by flattening the convolutional parameters into a 1-D array and concatenating the skip connections as before.

To accomplish this I will use PyTorch, a Python package enabling GPU accelerated neural network functionality [9]. In addition, the data will be generated using MATLAB using MCRoomSim, a package for MATLAB that enables RIR simulations with configurable room sizes, reflection characteristics, speaker placement, microphone placement, microphone pickup patterns, and speaker directivity patterns [10].

C. Dataset Generation

Sets of 1000 RIR's were generated using MCRoomSim. Each simulated room is rectangular with random dimensions within constraints described below. The sound source (speaker) was simulated to have a cardioid polar pattern output. This is to more closely match real-world speaker units. There were two simulated receivers (microphones) with omnidirectional pickup patterns, one just below the source and one in the center of the room. The specific placement for the source and receivers are described below and are visualized in figure 2.

- Room geometry:

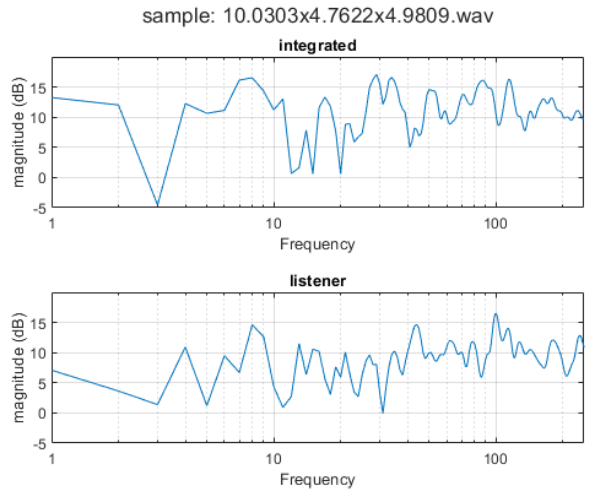

Fig. 1. Example sample simulated with MATLAB and MCRoomSim

  - Horizontal bounds: 3-12 m
  - Height bounds: 2.5-5 m
- Source Placement:
  - In the center of the front wall, 25 centimeters away from the wall.
- Receiver Placement:
  - Integrated: 25 cm below the source
  - Listener: In the center of the room, 1.2 m above the ground.

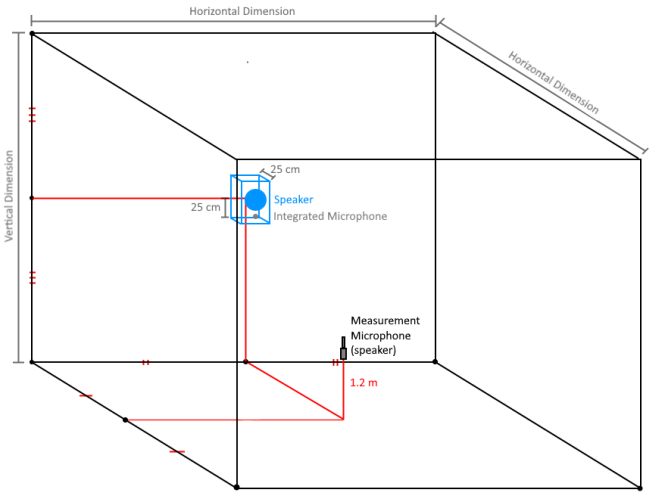

Fig. 2. Room Configuration in MCRoomSim

The simulated RIR's were generated at a 48 kHz sample rate. They were then downsampled to 1000 Hz using a low-pass filter followed by decimation. The first 1024 samples of each of these RIR's were taken — if they were smaller than 1024 samples they are padded to 1024 instead — then were then put through an FFT. The FFT was binned into 1 Hz intervals, from 0 Hz to 250 Hz. The magnitude of this data

was then saved to be used for comparison to the neural net. For each data point, there is one LFMR at the speaker location, and one at the listener location, as seen in figure 2.

D. Neural Net Architecture

The PyTorch model utilizes PyTorch's dataset and dataloader processes to organize the 1000 samples into a form that each model understands. They are trained on the data, then the expected and modeled outputs are saved. Here are the details for each model:

1) LFMR model
   a) 3 convolutional encoder/decoder layers + one bottom convolutional layer
   b) The encoder and bottom layers have 15 parameters and the decoder layers have 5 parameters
   c) Trained for 512 epochs
   d) Maxpool downsampler and nearest neighbor upsampler
   e) size 250 LFMR input
2) Time model
   a) 4 decoder/encoder layers + 1 bottom layer
   b) Encoder and bottom layers have 15 parameters, decoder layers have 5
   c) Trained for 3000 epochs
   d) Maxpool downsampler and nearest neighbor upsampler
   e) Size 1024 1000 Hz audio input
3) Dense Decoder Model
   a) 4 encoder layers + 1 bottom layer + 4 fully-connected dense decoder layers
   b) Encoder and Bottom layers have 20 parameters, dense layers have 1536 input channels and 512 output channels
   c) Trained for 2000 epochs
   d) Maxpool downsampler
   e) Size 1024 1000 Hz audio input

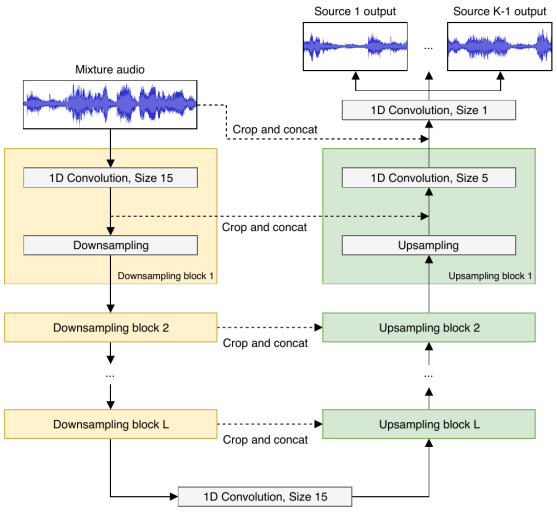

Fig. 3. A U-Net Architecture used for Speech Separation [7]

The models parameter number was adjusted based on the performance of the output. The encoder and decoder layers seemed to function best when the encoder layers were 2-3x larger than the decoder layers. Much of the adjustment was trial and error. The number of epochs were chosen based on when it seemed that the model stopped performing any better with successive training loops.

IV. Results and Discussion

Results were disappointing, but ultimately hopeful for the three models. I began with the LFMR model hoping that the simpler input would give the model less difficulty in estimating the desired output, but it seems that by putting the data into the frequency domain and discarding the phase information, the neural net did not find any solution. As the number of epochs increase, the outputs began outputting a mean of the input, as seen in Fig. 4. This informed me that I might have better luck using the time domain signal instead of a frequency domain one.

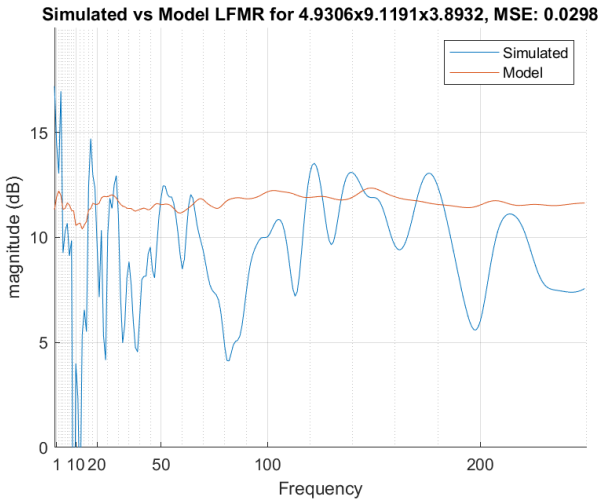

Fig. 4. A typical output from the LFMR model

I then altered the model to accept signals in the time-domain and increased the number of encoder/decoder layers by 1 hoping for better results. Initially, I was seeing a steep roll off in high frequency content from the network, but increasing the number of epochs gave the output of the network better high-frequency output. Eventually, my results began looking closer to what I was looking for. Despite this, the position of the troughs and peaks in the magnitude response did not match well between the expected output and the model output. This model was the most promising but ultimately not ready for implementation. See figures 5 and 6 for typical examples of the output.

Finally, in an attempt to increase the higher-frequency performance of the model, I implemented dense layers instead of decoder layers in the CNN while still utilizing the parameters from the skip connection. This model performed the worst, giving a very similar result for every single input, no matter

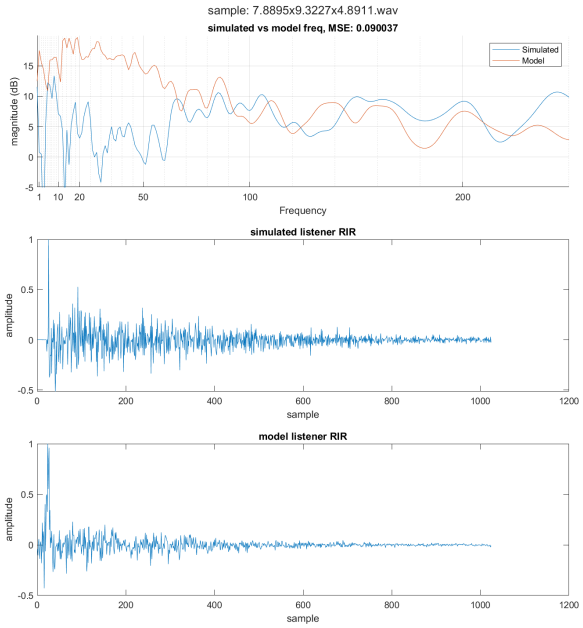

Fig. 5. A high performing example of the time model. We see that, especially for the frequencies above 60 Hz, there is significant overlap between the model and expected output.

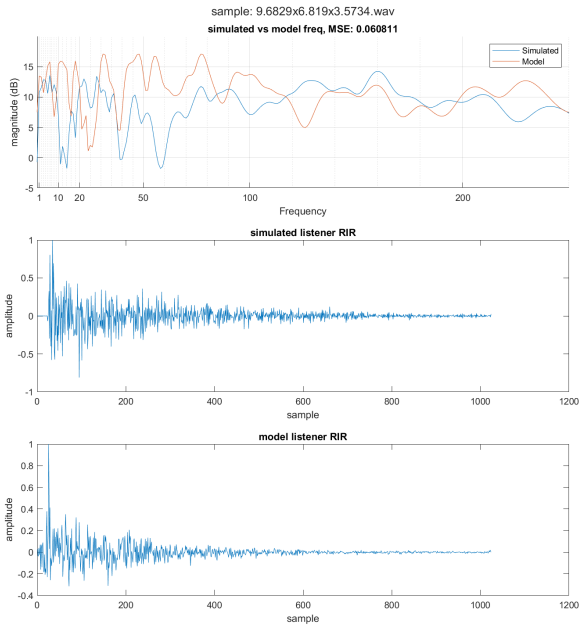

Fig. 6. A badly performing example of the time model. We see that while there are significant peaks and troughs, they don't line up well with the expected outcome well at all.

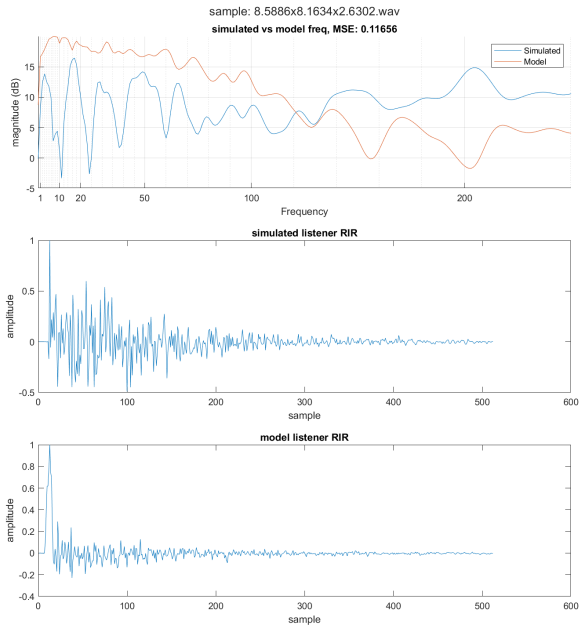

Fig. 7. An example of the dense decoder model. Every model output had a very similar shape to this model output.

the size of room. This made it clear that the U-Net is not compatible with dense layers in the way that I implemented them.

V. Conclusion and Future Work

While the observed results weren't what I was hoping for, this paper shows that there is promise in using machine learning to extrapolate the RIR for a different location in the room than the actual microphone. The first model, a U-Net with an LFMR as an input, converged to a mean, rendering the model useless. The second model, utilizing an RIR based U-Net CNN model, was the most effective at estimating a realistic frequency response. Perhaps the model would have performed better at a higher samplerate or if it were given a longer impulse response. The third model, a U-Net with a fully-connected decoder layer that had an RIR as an input, converged to a single output RIR, rendering it useless as well. Either way, it's clear that U-Nets have a significant time dependence, as prior implementations either use a spectrogram or time-domain signals as inputs [7], [8]. In the future, I will attempt to use a spectrogram as an input to the network to see if it is the phase information or the time-dependence that made the LFMR model fail. Another potential direction for a model such as this is to have one model detect room dimensions, as has been shown to work by [6], and use that as an input along with the simulated impulse response to have a model estimate the RIR at the listener position.

Assuming the potential success of a model such as this, future steps would be to generalize the input constraints by letting the integrated microphone and listener locations be

variable. This information could be passed to the CNN along with the input. In this state, the model could be implemented in a real-life smart speaker and tested in real environments.

ACKNOWLEDGMENTS

I want to thank my advisor, Dr. Chris Kyriakakis for the great direction and flexibility that he gave me throughout this project. He has influenced me immensely when it comes to the direction that I want my career to go and has taught me so much of what I know today.

I'd also like to thank Dr. Brandon Franzke for the precious hours he spent helping me program my PyTorch models. This project would have been much more time-consuming without his expertise in the field.

REFERENCES

[1] B. Fazenda, "Perception of room modes in critical listening spaces," Ph.D. dissertation, University of Salford, 2004.

[2] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.

[3] D. Markovic´, F. Antonacci, A. Sarti, and S. Tubaro, "Estimation of room dimensions from a single impulse response," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[4] B. Gunel, "Room shape and size estimation using directional impulse response measurements," 09 2002.

[5] Y. E. Baba, A. Walther, and E. A. P. Habets, "3d room geometry inference based on room impulse response stacks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 857–872, 2018.

[6] W. Yu and W. B. Kleijn, "Room acoustical parameter estimation from room impulse responses using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 436–447, 2021.

[7] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," 2018.

[8] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," 2021.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[10] A. Wabnitz, N. Epain, C. Jin, and A. Van Schaik, "Room acoustics simulation for multichannel microphone arrays," in *Proceedings of the International Symposium on Room Acoustics*. Citeseer, 2010, pp. 1–6.