DSP Audio Research Memo

Test Date: Friday October 24, 2025
Attendees: Ben Polzin, Matt Stowell, Dr. Christian Hearn
Location: NB202 / Staircase

---

1 Overview

The main purpose of this test was to demonstrate the current state of the project and test our software and hardware in action. We wanted to determine how well the current software could capture an impulse response *in practice*, and how well the custom hardware would work in a test setup.

We began by testing the main Python code on a Laptop in the graduate research room using the DUO audio card, and then proceeded to repeat the same test on the Raspberry Pi and custom hardware. Once we verified that everything was working sufficiently well for a real test, we moved the setup to the staircase near the ECE office and conducted several tests.

After finishing the tests, we discussed the possibility of writing a research paper and presenting at a conference. We also discussed attending the St. Joseph Christmas choir performance to get a reference for what the end goal of the project should sound like.

2 Test Setup

To conduct the test, we began by setting up the Raspberry Pi hardware and the Speaker on the 3rd floor. We then daisy-chained two microphone cables together and dropped it down the staircase to the microphone. We then captured 5 impulse responses: 1 on the first floor, 2 on the second floor, and 2 on the third floor. To test the directionality of the microphone, we reversed the microphone direction for the second impulse response on the second and third floors.

3 Findings

We found that the general test setup does work, but there are several anomalies that degrade the quality of the captured impulse response.

First, the speaker makes a loud popping noise at the start of playback. This may or may not be an issue, but it is possible that it is creating unwanted artifacts in the impulse response. The cause of this might be something with the Raspberry Pi audio card, or it could be how Python is handling playback. If the popping noise cannot be eliminated, it might be sufficient to delay the sine sweep and the recording for 500ms to avoid recording it.

The second anomaly is the strange repeating noises in the impulse response. It sounds like the sine sweep is being played very quietly in reverse underneath the impulse response. It repeats two times. The cause of this is unknown at this time, and it requires further exploration. We did notice however that the sine sweep artifacts are more prevalent the farther away the microphone is from the speaker. This means that it might be a noise issue or a correlation issue that is more prevalent when signal to noise ratio is lower.

One final issue is the low sampling rate of the Raspberry Pi audio card. The max sample rate appears to be 44.1kHz, which is low enough that noticeable distortion can be heard when the sine sweep reaches high frequencies (15kHz). It would be much better if we could record and playback at a sample rate of 192kHz, however this might require a different audio card.

4 Action Items

- Investigate popping noise. Consider delaying playback to compensate.
- Investigate repeating sound / sine sweep artifacts
- End recording based on noise floor. Add windowing to impulse response.
- Add a level meter program for speaker volume calibration

Figure 1: Top floor test setup. The Raspberry Pi and Speaker were setup on the 3rd floor, with the speaker facing outward. The program was controlled using the Linux terminal interface.

Figure 2: Omnidirectional Microphone on the first floor, facing forward. The microphone was connected to two cables daisy-chained together.

Figure 3: Microphone setup on the second floor, again facing forward. The microphone was also tested facing the glass. We found that the direction of the microphone did not make a significant difference in the sound quality.

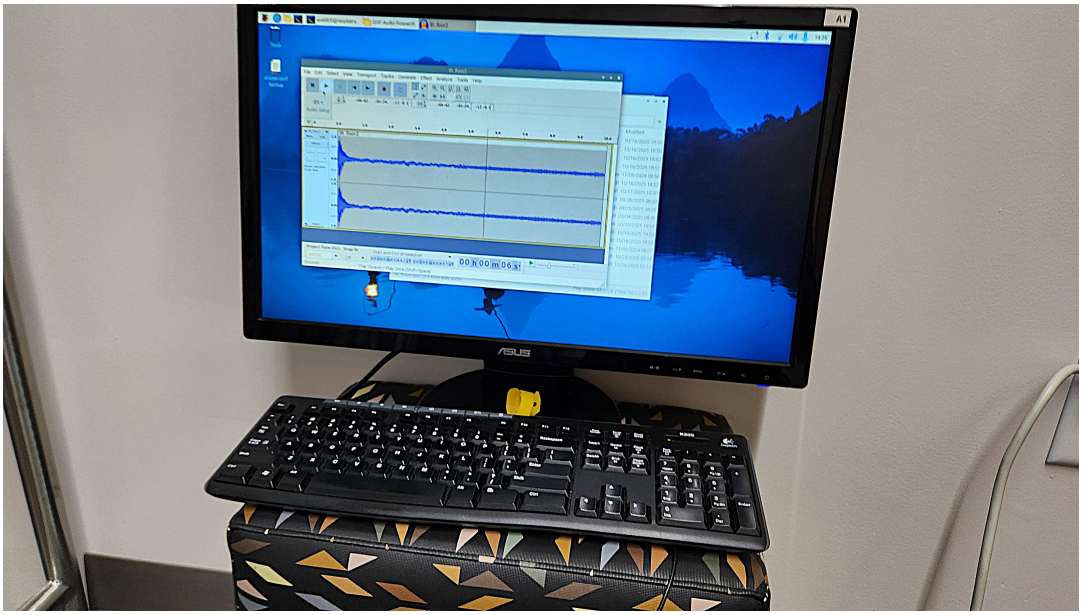

Figure 4: Analyzing the impulse response in audacity. The figure shows the large amount of noise that is present on the tail of the impulse response, which is mostly low frequency aliasing.