Introduction

In the third lab of EEE 321, we approximated the impulse response of a specific music auditorium, Odeon in Bilkent, and then simulated possible outputs to anechoic music files. We accomplished this by initially recording an impulse response, being the response of the auditorium to a balloon explosion heard from a high located seat, and then convolving this impulse response in MATLAB to possible music inputs to simulate the heard music at that seat if the inputted music was played.

Methodology and Results

An impulse response of an LTI (linear time invariant) system can be roughly expressed as the output of this system to the minimum length (etc. time) input signal of the impulse signal. Such a response helps one find the output for all potential inputs to this system, as the impulse response would be repeated for every point of an input signal. Thereby, the method of convolution for a continuous LTI system can be expressed as (h(t) is the impulse response and x(t) is any input):

$$h(t) * x(t) = \int_{-\infty}^{\infty} h(t-\propto)x(\propto)d\propto$$

In this lab, we initially found the impulse response by exploding a balloon, possibly the minimum time long signal, and recording the echoed output from the high located seat. Even though there was noise due to external factors such as wind, or errors of recording devices and computer algorithms that convert music files into different types, echoed music outputs were achieved from the selected seat that had a minimal amount of noise. The following are the graphs for input original music and the impulse response.

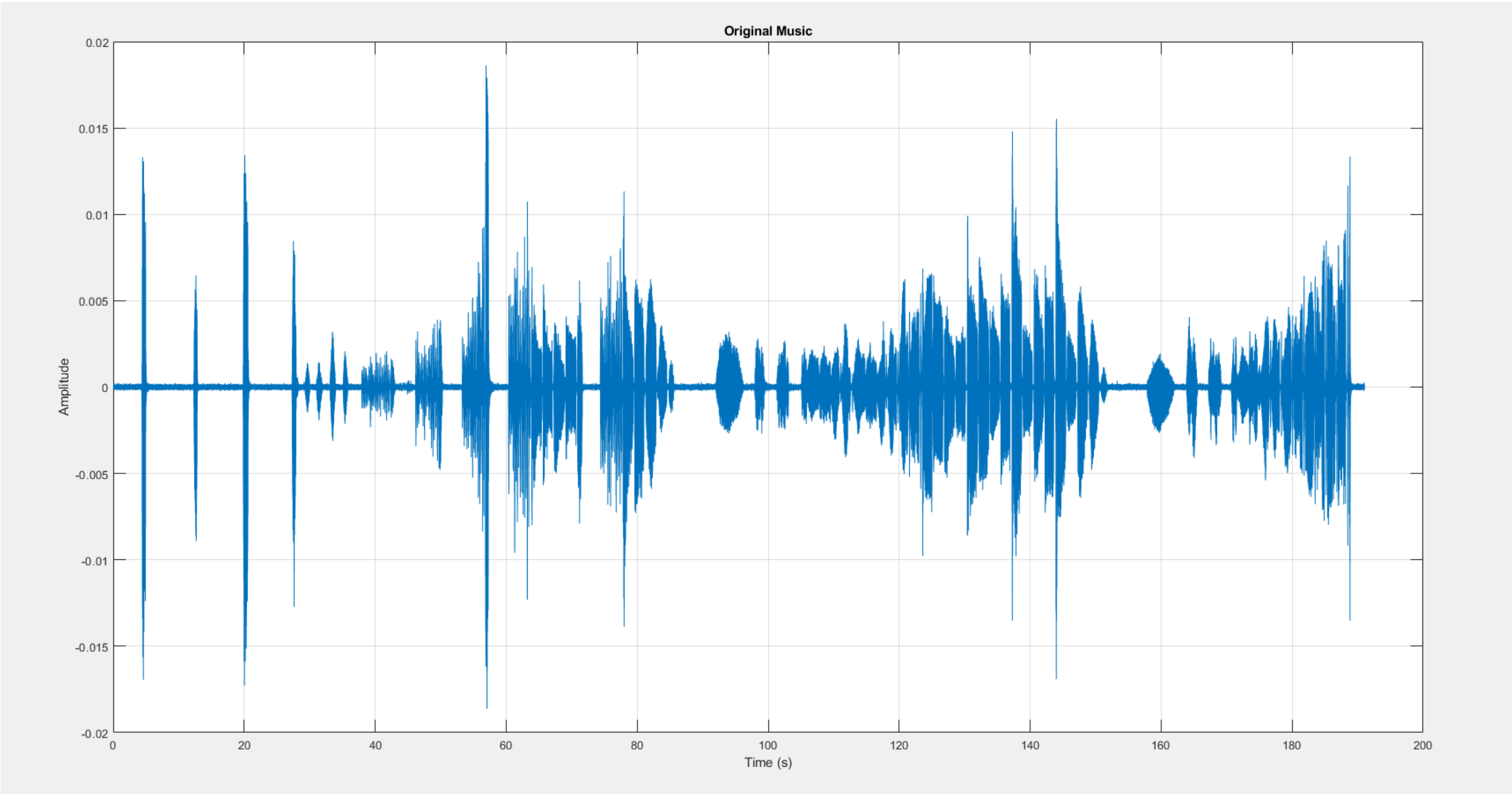


Figure 1: Original Music Waveform

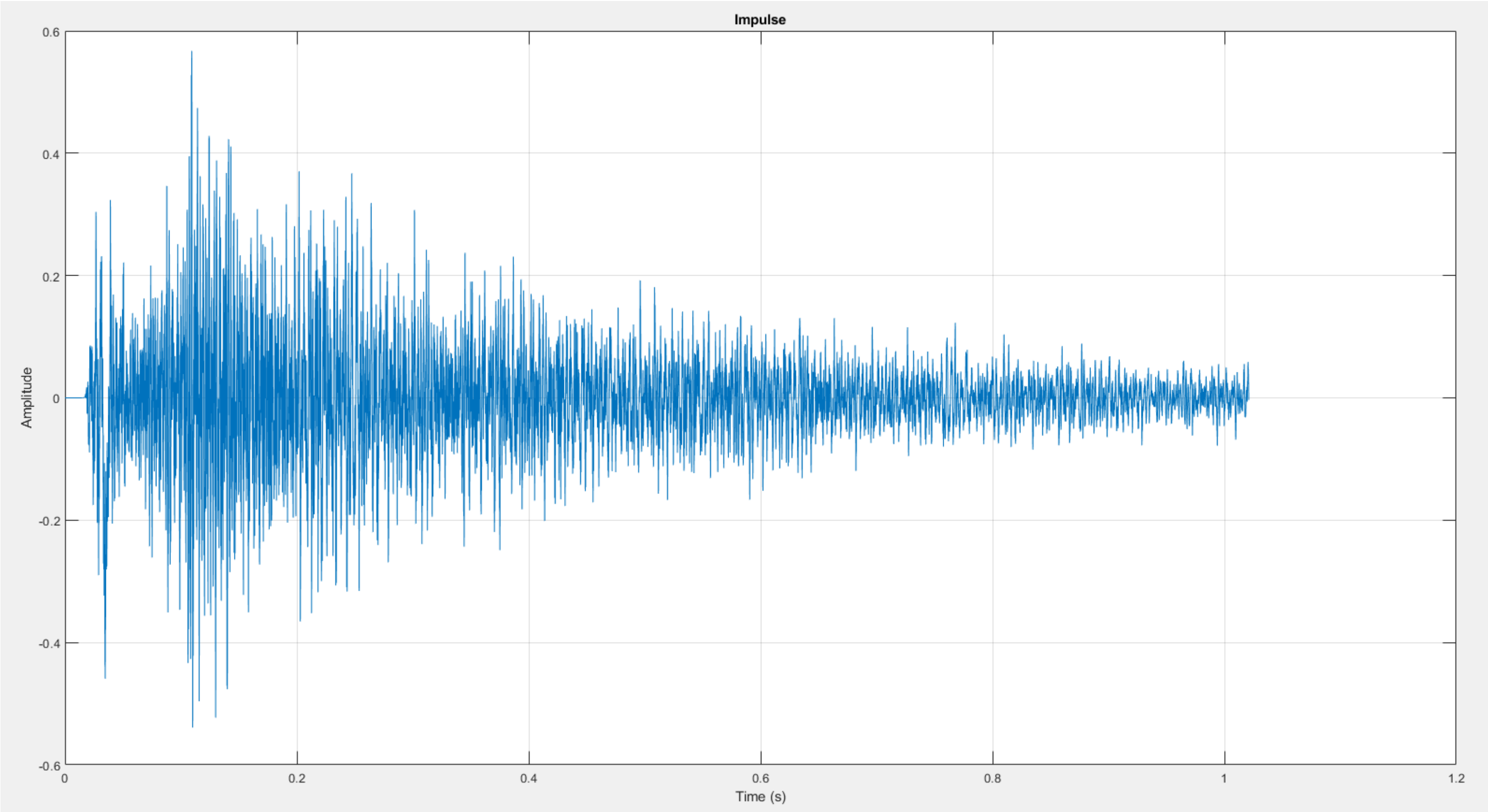


Figure 2: Impulse Response Waveform

The impulse response recorded on the phone was trimmed so that the explosion would begin at t=0, and the format of it was converted to MP3 so that MATLAB's 'audioread' function could use it. The input original music was downloaded from internet and was also converted to readable array form for MATLAB using 'audioread'.

The following are the MATLAB code for the lab session.

```
clc
clear
close all
% 22201832 Emir A. Bayer EEE321 Lab3

mp3file1 = 'lab3/beethoven_mp3/beethoven_v11a_6.mp3';
mp3file2 = 'lab3/impulsebaloon.mp3';

% reading audio data values and sample rates from mp3 files
[data1, sample1] = audioread(mp3file1);
[data2, sample2] = audioread(mp3file2);

% Create a time vector for the x-axis (in seconds)
time1 = (0:length(data1)-1) / sample1;
time2 = (0:length(data2)-1) / sample2;

% !! first sample rate = second = 48000 Hz

% original audio
figure;
plot(time1, data1);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Music');
grid on;

```

Figure 3: MATLAB Code 1

```
27  % impulse from baloon audio
28  figure;
29  plot(time2, data2);
30  xlabel('Time (s)');
31  ylabel('Amplitude');
32  title('Impulse');
33  grid on;
34
35
36  dataC = conv(data1,data2);
37  %sound(dataC, sample1);
38  audiowrite('convolved_output.wav', dataC, sample1);
39  timeC = (0:length(dataC)-1) / sample1;
40
41  % final output music (echoed)
42  figure;
43  plot(timeC, dataC);
44  xlabel('Time (s)');
45  ylabel('Amplitude');
46  title('Echoed Music');
47  grid on;
```

Figure 4: MATLAB Code 2

The 'audioread' function of MATLAB converts music data into an array that consists of the value for each chosen instance of a discrete time signal. The amount of instances reveal the sample rate of the audio file as frequency in Hertz (1/s), and each value incrementing discrete time reveal the amplitude of the music of that moment. Here, the music file has to be in discrete form, as computers cannot process continuous time signals as there would have to be infinite amount of storage. That's why a convolution of discrete read input files is made as in line 36, which forms the final echoed music array. The following is the formula for a discrete LTI system's convolution:

$$h(t) * x(t) = \sum_{k=-\infty}^{\infty} h(t-k)x(k)$$

At each time audioread is used, the functions outputs a sample rate (48000 Hz for both mp3 files in this case), and a data file that consists of the array. After finding the files for both audio inputs, time vectors have to be created in order to properly plot them, and in this case to plot the whole of music their lengths are found in seconds with the formula in lines 14 and 15. Later, the convolution is made as in line 36, and the found array is converted into a listenable .wav file in the MATLAB local directory. Later, the time length (in s) is found for the convolved output music and it is plotted in the same manner.

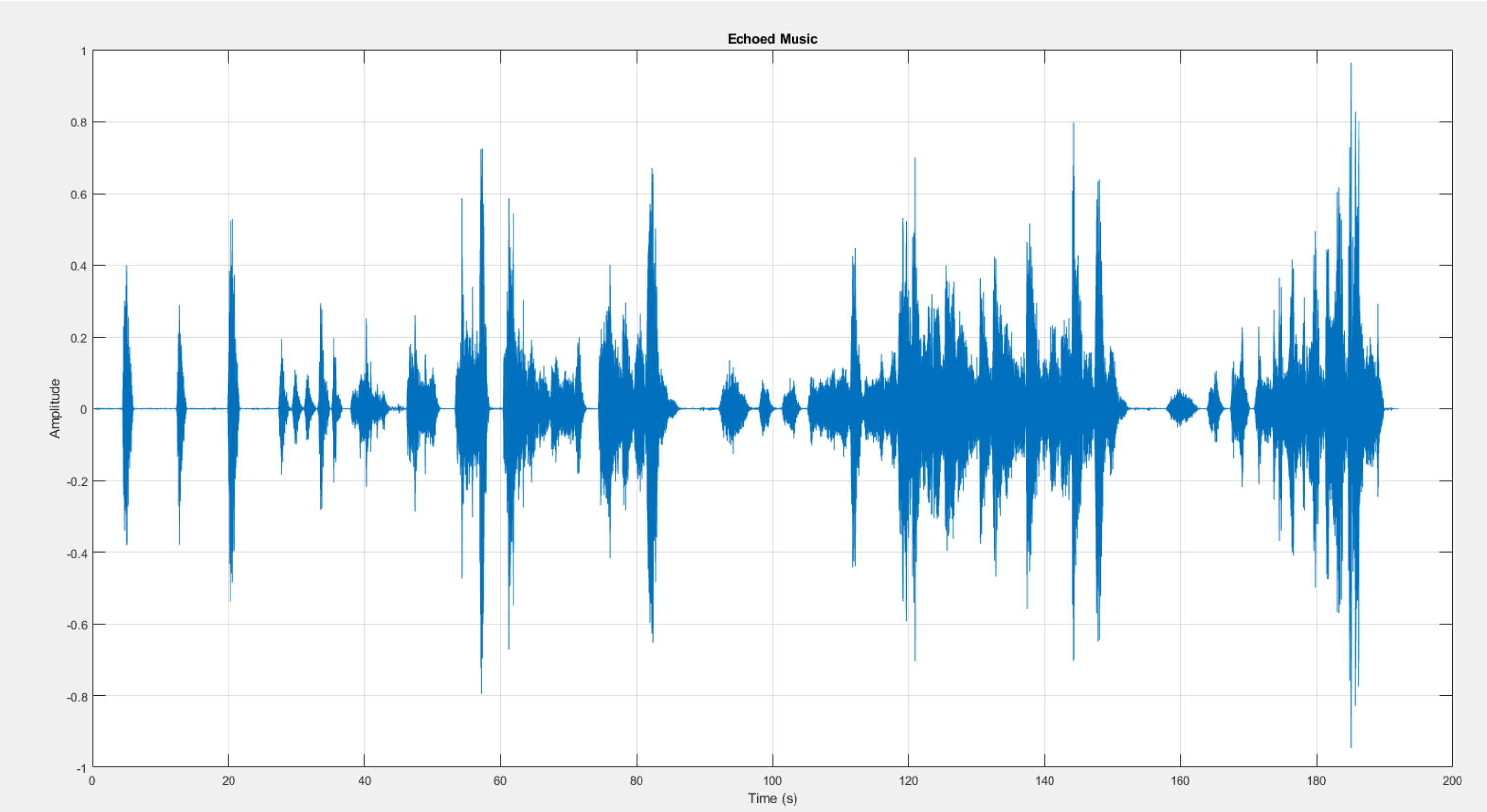


Figure 5: Complete Output Music Waveform

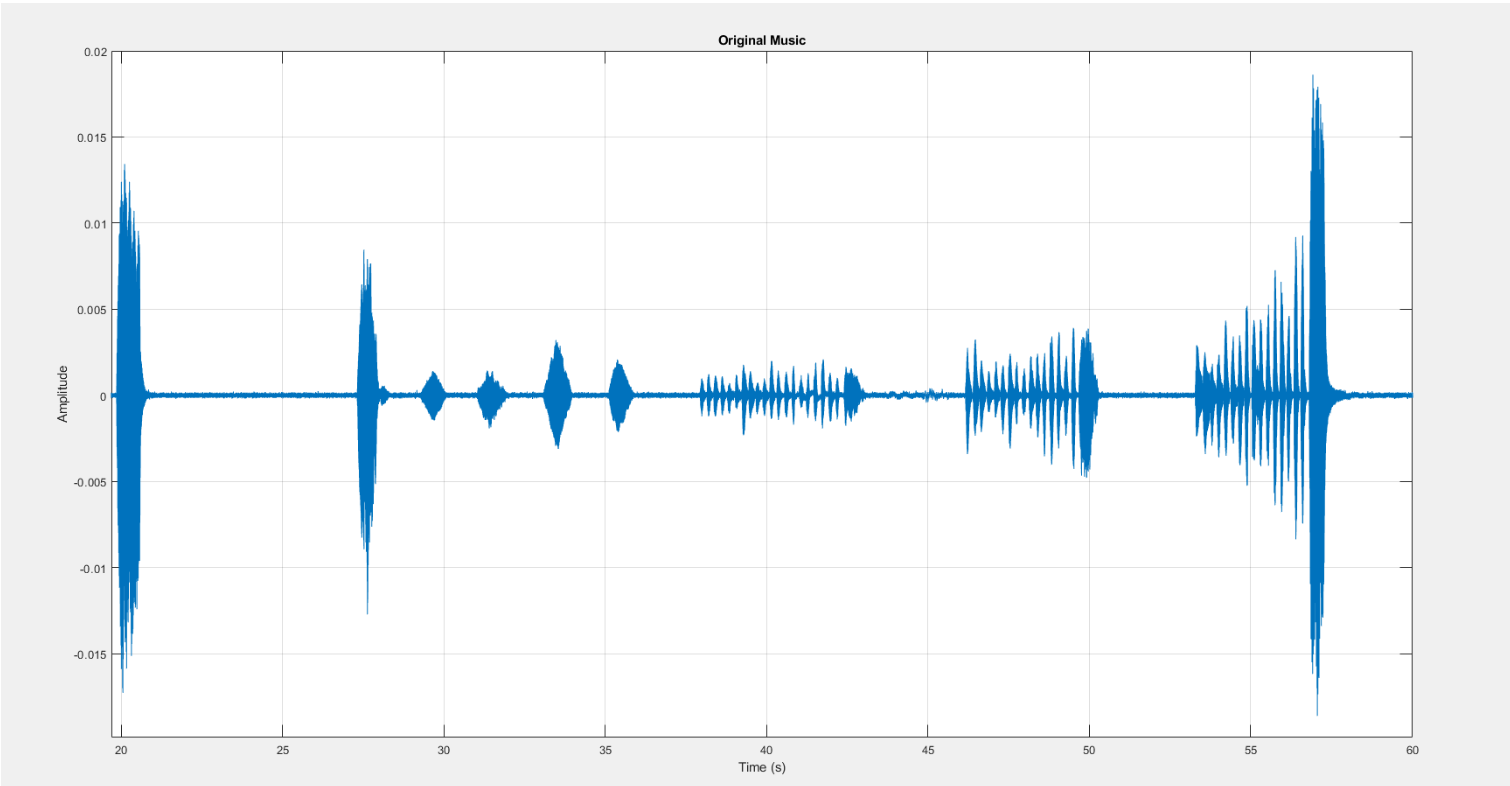


Figure 6: Input Music for 20s - 60s

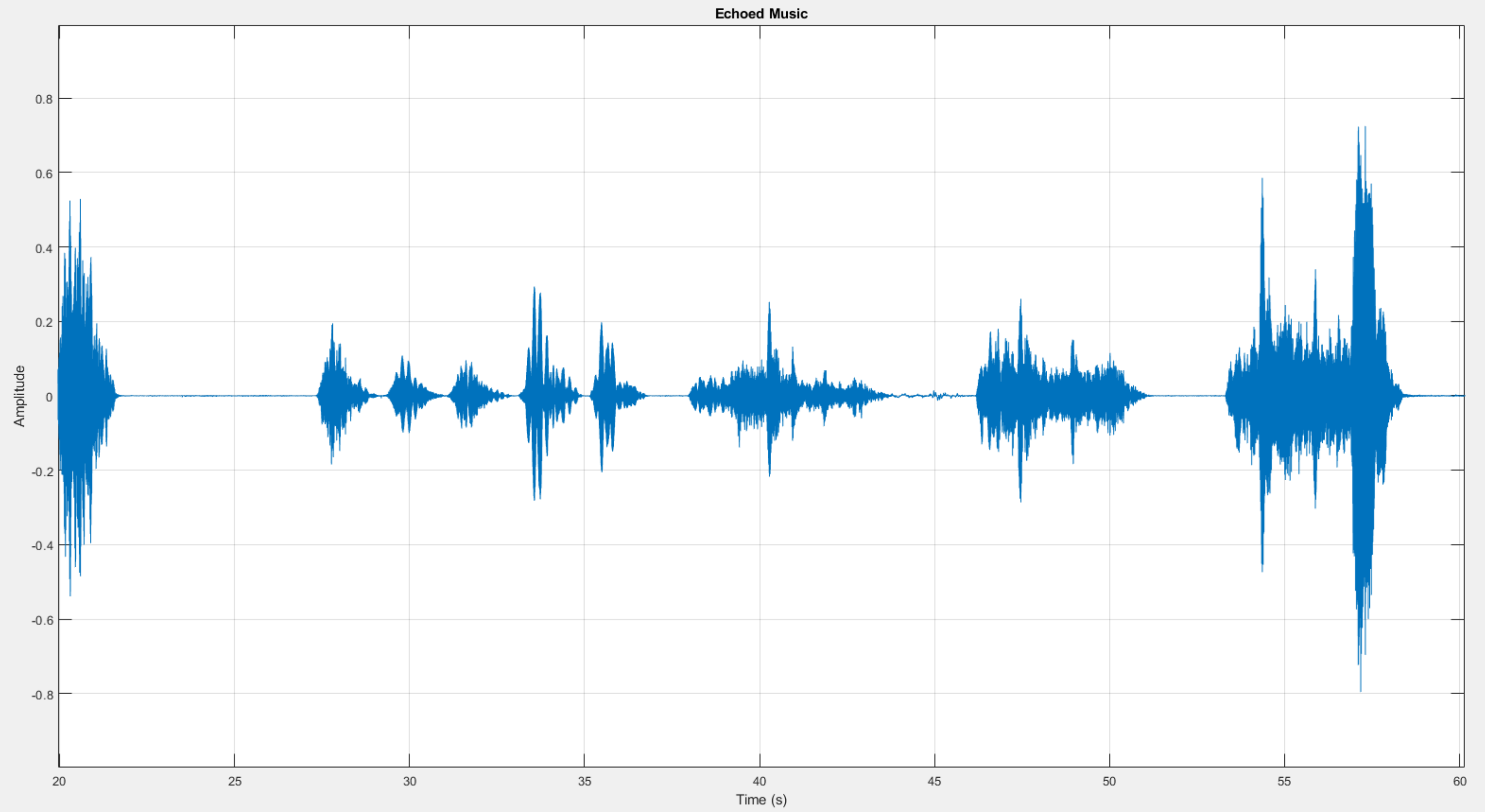
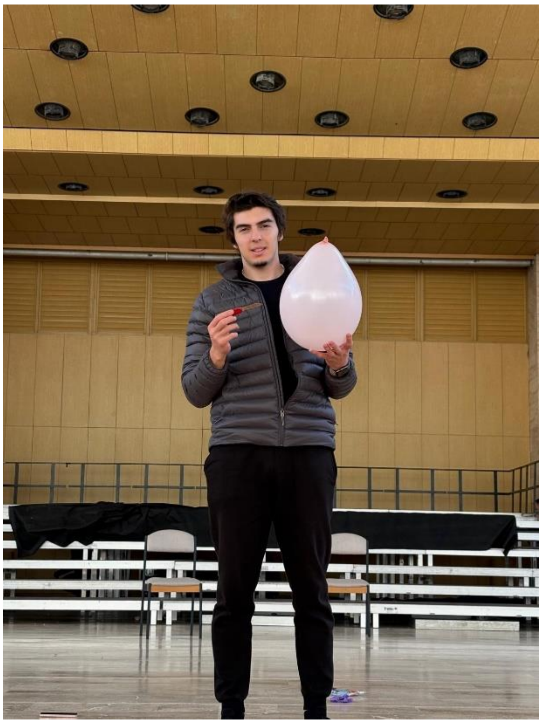


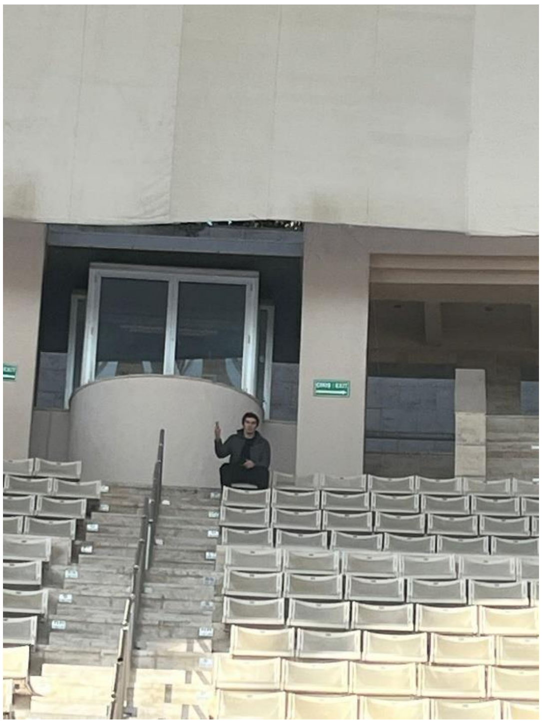
Figure 7: Echoed Output Music for 20s - 60s

Further Comments On The Results

The impulse is not infinitesimally minimal in length of time, as physically the balloon explosion the due needle takes a countable amount of time. That's why the impulse response is actually the response for a minimal length physical input. There are possible distortions in the input due to physical conditions of the balloon. Due to the acoustic nature of Odeon, the input noise was echoed, and the impulse response was in the form of a decaying exponential beginning at t=0. For the system to be LTI, it had to be time invariant and linear, and it can be said that the results would be approximately same for the conduction of the experiment at different times, however the external effects of wind or possible noise sources could be drastically different. Any combination of inputs for a single output would reveal the almost same output for the combination of the outputs for these two inputs, but changes of natural external factors could again change the behaviour of the impulse, impulse response and acoustic characteristics of the auditorium, as sound is impacted by the state of air it passes through. For instance, the combination of different outputs of separate instruments give the output of the combination of individual outputs. Microphone of the mobile phone I used for recording, computer sampling and file conversion was also highly likely to cause further distortions in output music, as such systems convert the input continuous time sound signals into discrete ones, and make calculations over selected discrete points in the input music. There was unpreventable noise due to computer recording characteristics and natural wind. These changed the input singal inputted into MATLAB, as they were almost constantly existent and changed the ideal form of infinitesimally short timed impulse input. The acoustic design of Odeon created echoes on the input music, echoing every discrete sound signal into exponential decrease. Such reverberation may add a kind of artistic effect to the anechoic sound signals, possibly making it more pleasant to listen to in real life, as decaying character of sound may have a calming effect. Such reverberation of sound coming from all angles would not exist in a completely open environment. However, due to distortions in the simulated environment for the impulse response, the simulated output can be argued to be less pleasant due to constant existence of the natural factors that existed at the moment of recording such as wind.

Appendices





Figures 8-11: Me and My Partner Conducting the Lab in ODEON