SIGNALS AND SYSTEMS LAB3 REPORT

NEHİR DEMİRLİ

22203611

Objective of the Experiment:

The major goal of this experiment is to determine the output of a Linear Time-Invariant (LTI) system using its impulse response. Because both MATLAB and the sound recording device collect continuous sound signals in discrete form via sampling, the convolution process will be described in discrete terms. The discrete impulse response for LTI systems can be estimated using the equation below.

$$y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k]$$

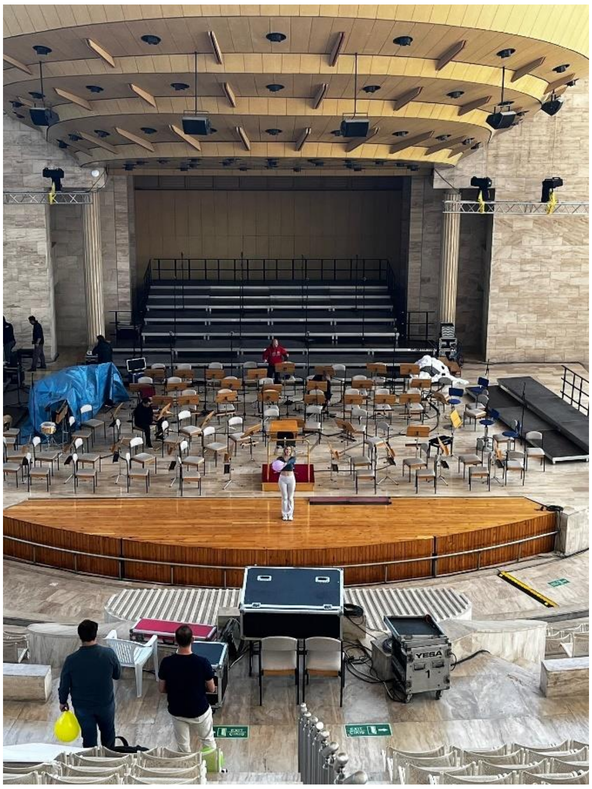
In this experiment, the sound recording of a balloon being blown up with a needle simulates the desired seat's response to an impulse generated on the Odeon stage. The system's input is the anechoic music signal, while the output is the music heard at the Odeon's assigned seat. This can be summarized as follows:

1. The system's input (x[n]) = An anechoic clear music recording(Violin part of Beethoven in this case).

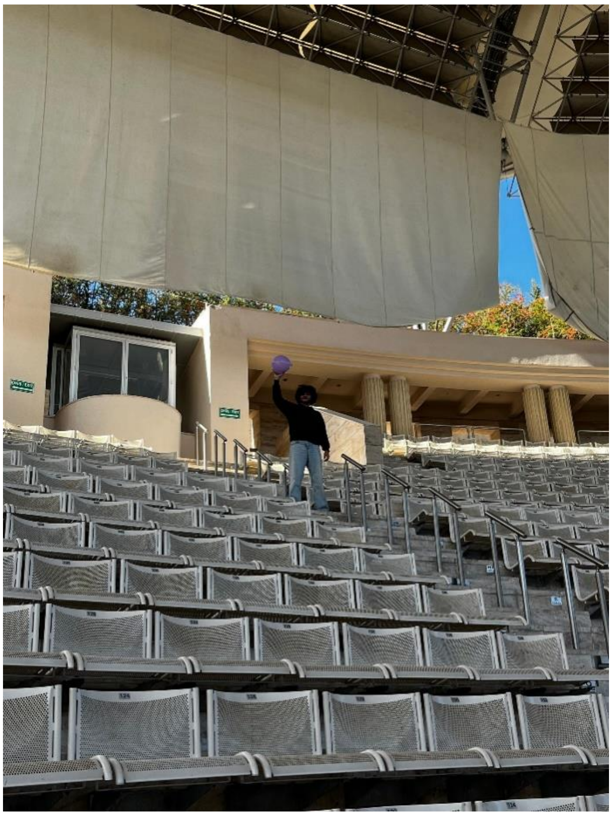
2. Impulse Response of the LTI System (h[n]) = Sound recording of the blown-up balloon

3. The output of the system to the Given Input (y[n]) = Music to be heard from the seat when the music is played on the stage.

After comprehending the premises, I blew up a purple balloon in the center of the Odeon stage (Figure 1) and my best friend Emre recorded the sound from the seat (Figure 2). This sound recording captures the LTI system's response to the impulse signal.

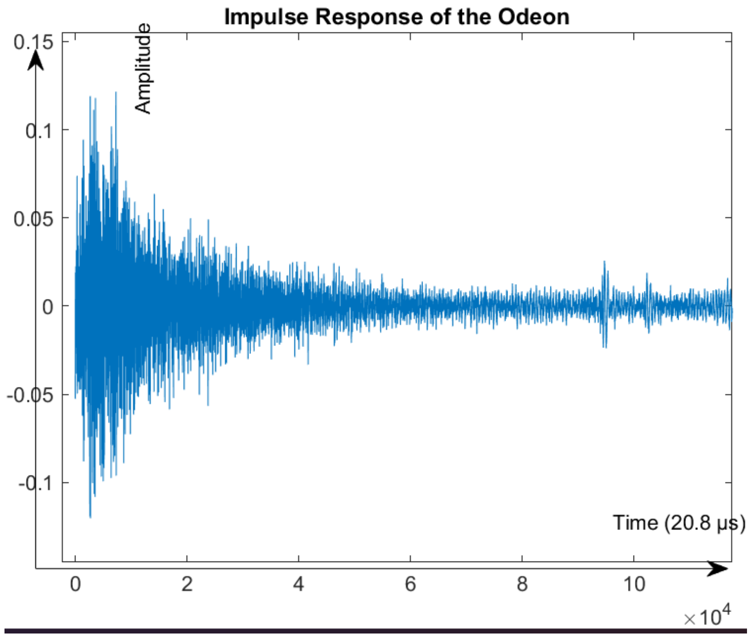


[Figure1: The place where the balloon is blown up with a needle]



[Figure 2: Location of my friend, Emre, at a seat that we have chosen to record the impulse]

After the sound was caught at the Odeon, the entire analysis was carried out using MATLAB. The input sound is an anechoic recording of the second violin part of Beethoven's 7th Symphony [1]. The sound of the balloon burst, which represented the system's impulse response, and the anechoic sound, which served as the input signal, were processed in MATLAB and turned into discrete time arrays with a sampling rate of 48kHz, which translates to one sample every 20.8 microseconds. The properties of each signal can be determined by studying the plots of a piece of the input signal (Fig.3) and the system's impulse response (Fig.4).

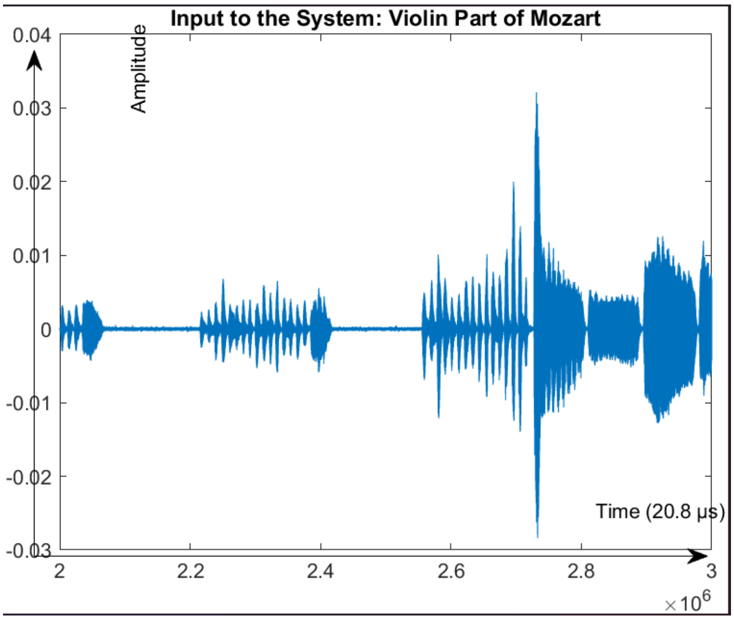


[Figure 3: Sound of ballon at Odeon]

To ensure the system's causality, the sound recording of the balloon was sampled from the point of blowing. The system's impulse response must satisfy the following inequality to be considered a causal system, a requirement for any physical system.

$h[n] < 0 \; if \; n < 0$

There were another approaches as well for casuality,however,;for convenience and to ensure an unshifted result, the signal was manually chopped.



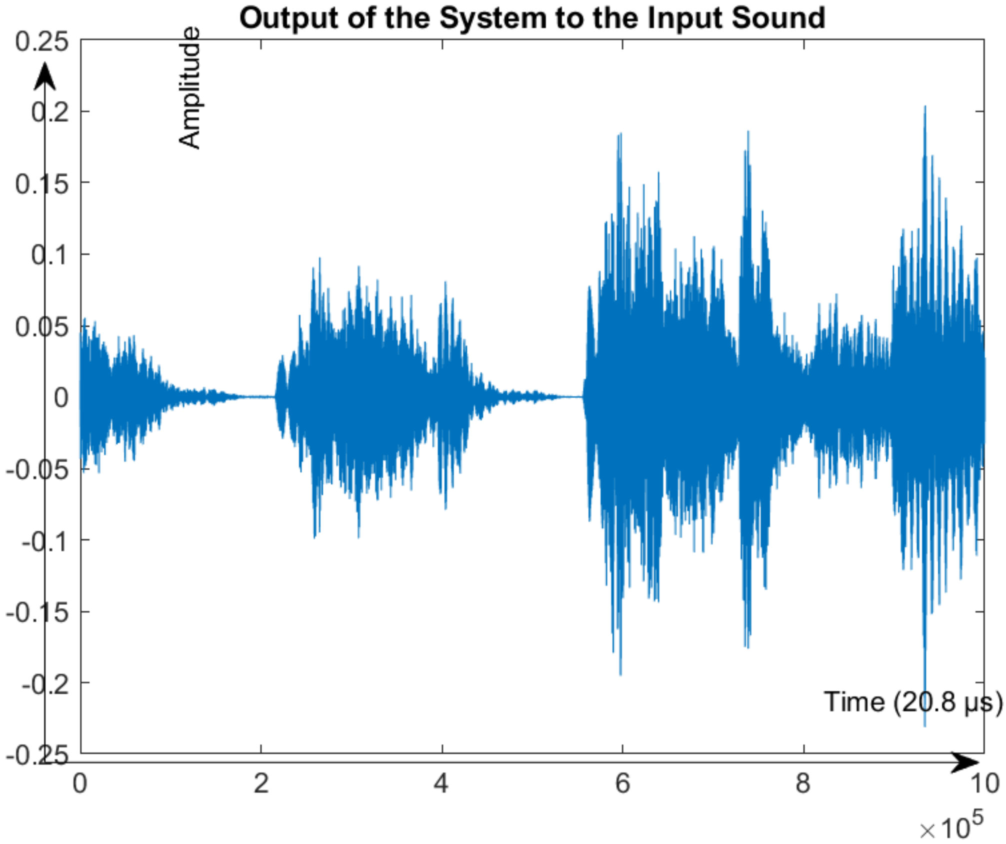
[Figure 4: Plot of the input signal]

After converting these sounds into MATLAB, convolution was done with conv method with discrete time arrays. Figure 5 illustrates the code:

```
[odeonImpulseResponse, sampleRateOdeon] = audioread(fullfile('C:', 'Users', 'yesim', 'labses.m4a'));
[violinSample, sampleRateViolin] = audioread(fullfile('C:', 'Users', 'yesim', 'beethoven_vl2b_6.mp3'));

% Viyolinin ses seviyesini artır
amplifiedViolin = violinSample * 1.8;

% Empuls yanıtını belirli bir aralıkta kırp
trimmedImpulseResponse = odeonImpulseResponse(149000:end);

% Konvolüsyon işlemi ile viyolin sesini filtrele
filteredOutput = conv(amplifiedViolin, trimmedImpulseResponse);

% Verileri kaydet
save('impulseResponse.mat', 'trimmedImpulseResponse');
save('filteredOutput.mat', 'filteredOutput');
save('amplifiedViolin.mat', 'amplifiedViolin');

% Sonuçları ses dosyalarına yaz
audiowrite('output_amplifiedViolin.m4a', filteredOutput, sampleRateOdeon);
audiowrite('input_amplifiedViolin.m4a', amplifiedViolin, sampleRateOdeon);
audiowrite('input_trimmedImpulse.m4a', trimmedImpulseResponse, sampleRateOdeon);

% Kaydedilen verileri yükle
load('filteredOutput.mat', 'filteredOutput');
load('impulseResponse.mat', 'trimmedImpulseResponse');
load('amplifiedViolin.mat', 'amplifiedViolin');

% İmpuls yanıtını çiz
%plot(trimmedImpulseResponse);
%title('Impulse Response of the Odeon');
%xlabel('Time (20.8 µs)');
%ylabel('Amplitude');

plot([2000000:2999999], amplifiedViolin(2000000:2999999));
title('Input to the System: Violin Part of Mozart');
xlabel('Time (20.8 µs)');
ylabel('Amplitude');

%plot([1:1000000], filteredOutput(2000000:2999999));
%title('Output of the System to the Input Sound');
%xlabel('Time (20.8 µs)');
%ylabel('Amplitude');
```

[Figure 5: The code]

Converging two signals yields the LTI system's output, which is then translated to a song file for playback. Figure 6 shows the output which is the concolution of the aforementioned signals.



[Figure 6: Plot of the output of the system]

With these findings, the experiment is over. There are three unique audio files: the input song, the impulse response (represented by the sound of the balloon pop), and the output signal produced by the convolution process.

COMMENTS ON THE RESULTS OF THE EXPERIMENT:

1) QUALITY OF IMPULSE:

The method by which the balloon is burst has a significant impact on the efficiency of the impulse signal. To ensure that the impulse is precisely focused at t = 0, the physical activity of the burst must be both rapid and concentrated at this particular point. Furthermore, because musical instruments make sound by vibrating air particles, impulse reactions cannot be based on other mechanical causes. Thus, inducing an impulse by dropping an object to the ground is ineffectual for a variety of reasons. First, upon impact, the object is likely to rebound, resulting in several collisions before coming to rest. This sequence produces many impulse-like noises of decreasing intensity, lowering the quality of the primary impulse. Secondly, this form of hit causes vibrations in the solid ground before reaching the air, as opposed to most musical instruments, which induce air particles to vibrate without first making contact with a solid surface.

All in all, the act of bursting a balloon meets the physical requirements of an ideal impulse in such an experiment. First, the impulse must be brief and concentrated at $t = 0$, in accordance with the ideal impulse signal. Second, only the immediate consequences of this single impulse are desired, with no lingering signals from the source. Finally, the initial vibration medium should correspond to that of musical instruments, which is air. The balloon's burst sound is caused by the flexibility of its material and the high air pressure inside, which meets the requirement for an air-based medium.

2) NATURE ABOUT THE IMPULSE RESPONSE

In this context, the system's impulse response is captured using sound signals emitted from the stage when a balloon is ruptured with a needle. Assuming that this balloon burst represents an appropriate impulse representation, the echoes produced by the Odeon's acoustics at a certain seating position are interpreted as the system's response to an impulse on stage. The impulse reaction includes all details provided by the surrounding materials, such as ceiling openings, floor and seat materials and forms, stage design, building architectural arrangement, air quality, temperature, weather, and other environmental aspects. Some components, such as the Odeon's structural architecture, have a greater impact than others, such as an individual seat in the back.

However, filtering the anechoic sound using an impulse response captured when the Odeon was empty would result in significant inaccuracies. This is because, throughout a concert, audience members take different seats, altering the acoustic response of the Odeon's structure.

3) VALIDITY OF THE LTI SYSTEM ASSUMPTION OF THE ACUSTIC ENVIRONMENT

A system's time invariance implies that it should consistently produce the same outcomes regardless of changes in time or temperature. However, the Odeon does not entirely fit this condition because factors like as heat, air density, audience size, and external noise levels might change, affecting the system's responsiveness[1]. Thus, the experiment's conclusion reflects the system's reaction to an input at a certain time. However, because these elements have a lower influence on the impulse response than the building's structural design, they can be excluded from this portion of the experiment. The basic assumptions are that no things enter or exit the Odeon, and that the temperature, air pressure, and meteorological conditions(a bit cloudy and rainy in our case) remain constant throughout.

Linearity is also a significant consideration. Fortunately, as demostrated the air medium generally behaves linearly for sound waves with safe amplitudes, allowing the convolution method to be used. Higher amplitude signals, such as shock waves, can disturb linearity by changing the sound speed, making signal combination difficult. For lesser amplitude signals, however, the system's response maintains a dependable linear approximation.

Assuming that these small elements have no impact, we can regard the Odeon as an LTI system, with the output determined by convolving its impulse response with any given input signal.

4) DISTORTIONS AND THEIR REASONS

The physical environment is a major source of distortion in this configuration; producing a totally isolated impulse is difficult due to the myriad minor sounds that surround the Odeon[1]. Even slight background sounds can interfere with the purity of the impulse signal, affecting the overall output. Another important issue is the recording device itself. We tried to clean the micrphone output of Emre's phone, believing that little dust or dirt would damage the microphone apertures, but this does not eliminate all potential sources of distortion and hence, materials within the microphone and surrounding circuitry can cause slight inaccuracies.

Furthermore, the phone's signal sampling mechanism, which records audio for storage, naturally loses some information, resulting to additional distortion. When we use MATLAB to convolve the impulse response with the input signal, there is a possibility of some distortion due to MATLAB's numeric restrictions, which only allow for a set byte size for each integer.

Finally and lastly , ambient noise during the recording process might enter the signal stream at various points, ranging from wind impacts to faint background hums, all of which have an affect on impulse response. In LTI system terms, this interference is represented as an extra noise function N(n) that overlays the ideal impulse response h(n). Consequently, while minor, these distortions diminish the clarity of the impulse signal, with the recording device's microphone being the principal source of these difficulties.

5) NOISE DURING RECORDING AND ITS EFFECTS

In physical locations, particularly those with limited isolation, such as our experimental setup, a variety of ambient noises, such as wind, weather variations, and distant traffic, can cause unanticipated echoes and disturbance. In the Odeon, ambient sounds merge into the main audio stream, interfering with the impulse response recording by adding layers of extraneous noise. The interference can be stated as follows:

$$h[n] = h'[n] + N[n]$$

where $h'[n]$ h ′ [n] represents the actual impulse response and $N[n]$ denotes the noise supplied by various sources[2].

These background noises degrade the clarity and authenticity of the recorded signal. Acoustically, I believe, structures like as the Odeon are intended by architects to augment and deepen the harmonics of sounds on stage, naturally adding echoes that increase the music's depth and create a "surround sound" effect. This concept is comparable to home theater systems, which project sound from various angles to provide a more immersive audio experience. Echoed, harmonically enriched sounds are frequently more pleasing than flat, unechoed versions because they provide texture and depth to the listening experience. As listeners, we are ready to accept some distortion since the harmonics and echoes add to a more complete, immersive audio experience.

REFERENCES:

[1] siteadmin, “Bilkent Odeon,” *Bilkent Symphony Orchestra*, 2024. https://bso.bilkent.edu.tr/en/index.php/the-orchestra/bilkent-odeon/ (accessed Oct. 27, 2024)

[2] “Anechoic recordings of symphonic music..,” *diyAudio*, Apr. 15, 2021. https://www.diyaudio.com/community/threads/anechoic-recordings-of-symphonic-music.371001/ (accessed Oct. 27, 2024).

APPENDIX:

The code:

```
[odeonImpulseResponse, sampleRateOdeon] = audioread(fullfile('C:', 'Users', 'yesim',
'labses.m4a'));
[violinSample, sampleRateViolin] = audioread(fullfile('C:', 'Users', 'yesim',
'beethoven_vl2b_6.mp3'));


amplifiedViolin = violinSample * 1.5;


trimmedImpulseResponse = odeonImpulseResponse(149000:end);


filteredOutput = conv(amplifiedViolin, trimmedImpulseResponse);


save('impulseResponse.mat', 'trimmedImpulseResponse');
save('filteredOutput.mat', 'filteredOutput');
save('amplifiedViolin.mat', 'amplifiedViolin');


audiowrite('output_amplifiedViolin.m4a', filteredOutput, sampleRateOdeon);
audiowrite('input_amplifiedViolin.m4a', amplifiedViolin, sampleRateOdeon);
audiowrite('input_trimmedImpulse.m4a', trimmedImpulseResponse, sampleRateOdeon);


load('filteredOutput.mat', 'filteredOutput');
load('impulseResponse.mat', 'trimmedImpulseResponse');
load('amplifiedViolin.mat', 'amplifiedViolin');
```

```
%figure;
%plot(trimmedImpulseResponse);
%title('Impulse Response of the Odeon');
%xlabel('');
%ylabel('');


%annotation('arrow', [0.1, 0.9], [0.1, 0.1]);
%annotation('arrow', [0.1, 0.1], [0.1, 0.9]);
%text(1.02, 0.1, 'Time (20.8 µs)', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top');
%text(0.1, 1.02, 'Amplitude', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top', 'Rotation', 90);


%figure;
%plot(2000000:2999999, amplifiedViolin(2000000:2999999));
%title('Input to the System: Violin Part of Mozart');
%xlabel('');
%ylabel('');



%annotation('arrow', [0.1, 0.9], [0.1, 0.1]);
%annotation('arrow', [0.1, 0.1], [0.1, 0.9]);
%text(1.02, 0.1, 'Time (20.8 µs)', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top');
%text(0.1, 1.02, 'Amplitude', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top', 'Rotation', 90);


figure;
plot([1:1000000], filteredOutput(2000000:2999999));
title('Output of the System to the Input Sound');
xlabel('');
ylabel('');


annotation('arrow', [0.1, 0.9], [0.1, 0.1]);
annotation('arrow', [0.1, 0.1], [0.1, 0.9]);
text(1.02, 0.1, 'Time (20.8 µs)', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top');
text(0.1, 1.02, 'Amplitude', 'Units', 'normalized', 'HorizontalAlignment', 'right',
'VerticalAlignment', 'top', 'Rotation', 90);
```