Group 8

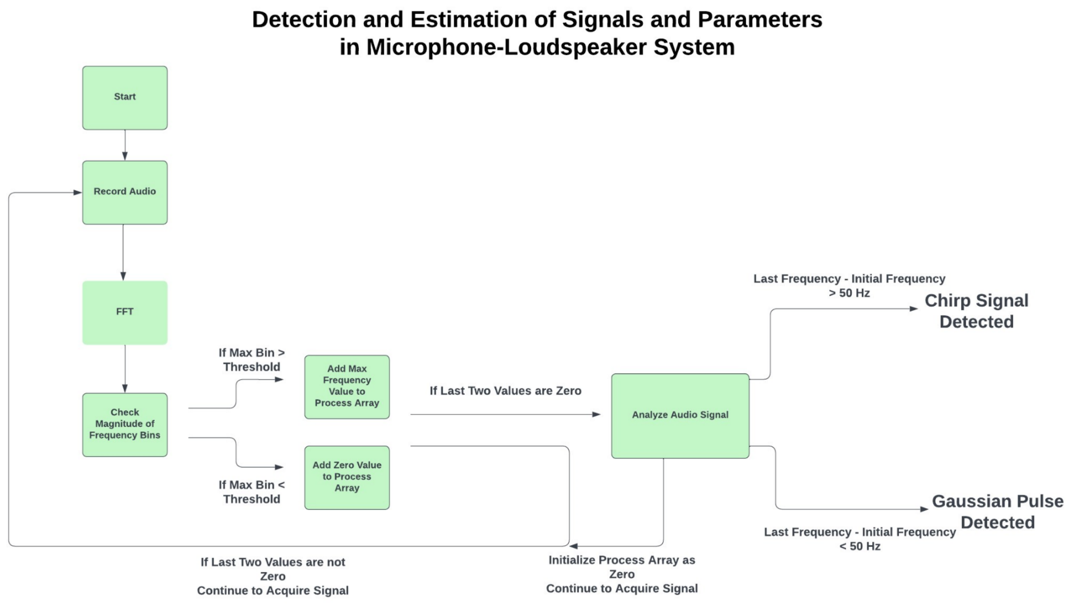
İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288

EE433
Real Time Applications of Signal Processing Term Project:
Detection and Estimation of Signals and Parameters In Microphone-Loudspeaker System

Introduction

This project aims to develop a real-time application of digital signal processing using C/C++, and LabVIEW. The objective is to create a signal that is either a gaussian pulse or a chirp signal with specified parameters and play it through a loudspeaker of a computer and detect the signal by capturing samples from a microphone via another computer while the receiver computer estimates the type and the parameters of the signal transmitted and report the results via a user interface. This report presents an overview of the project requirements and outlines the tasks to be completed.

Modules and Solutions


Detection and Estimation of Signals and Parameters in Microphone-Loudspeaker System
Start
Record Audio
FFT
Check Magnitude of Frequency Bins
If Max Bin > Threshold
Add Max Frequency Value to Process Array
If Max Bin < Threshold
Add Zero Value to Process Array
If Last Two Values are Zero
Analyze Audio Signal
Last Frequency - Initial Frequency > 50 Hz
Chirp Signal Detected
Gaussian Pulse Detected
Last Frequency - Initial Frequency < 50 Hz
If Last Two Values are not Zero Continue to Acquire Signal
Initialize Process Array as Zero Continue to Acquire Signal

İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288

User Interface Design: This module focuses on designing the user interfaces for both the transmitter and receiver computers. It involves creating an intuitive and interactive interface for adjusting signal parameters, selecting signal types, and displaying the detection and estimation results.

Signal Generation and Transmission: This module involves generating the desired pulse type signals, namely the Chirp signal and Gaussian pulse, with the specified parameters such as amplitude, frequency range, rate of frequency increase, pulse duration, and Gaussian parameters. This module is implemented in the LabView interface.

Signal Reception: This module focuses on capturing the transmitted signal using the microphone of the receiver computer. It involves reading the signal samples and preparing them for further analysis and parameter estimation.

Signal Detection: This module is responsible for detecting the transmitted signal from the captured samples. It analyzes the received signal and determines whether a detection has occurred or not by looking at the magnitude of the frequency components. If a signal is fully detected, the type of signal (Chirp or Gaussian pulse) is reported. The algorithm returns Gaussian pulse detection, if the difference between initial and final frequencies is smaller than a previously determined threshold value. On the other hand, the algorithm returns chirp detection, if the difference between initial and final frequencies is greater than a previously determined threshold value.

Parameter Estimation: This module estimates the parameters of the detected signal. The estimated parameters are then displayed in the user interface of the receiver computer.
Parameters for Gaussian pulse:

- f0 = f1 : Frequency of the gaussian pulse
- Duration: The time-duration of the pulse in terms of milliseconds

Parameters for Chirp signal:

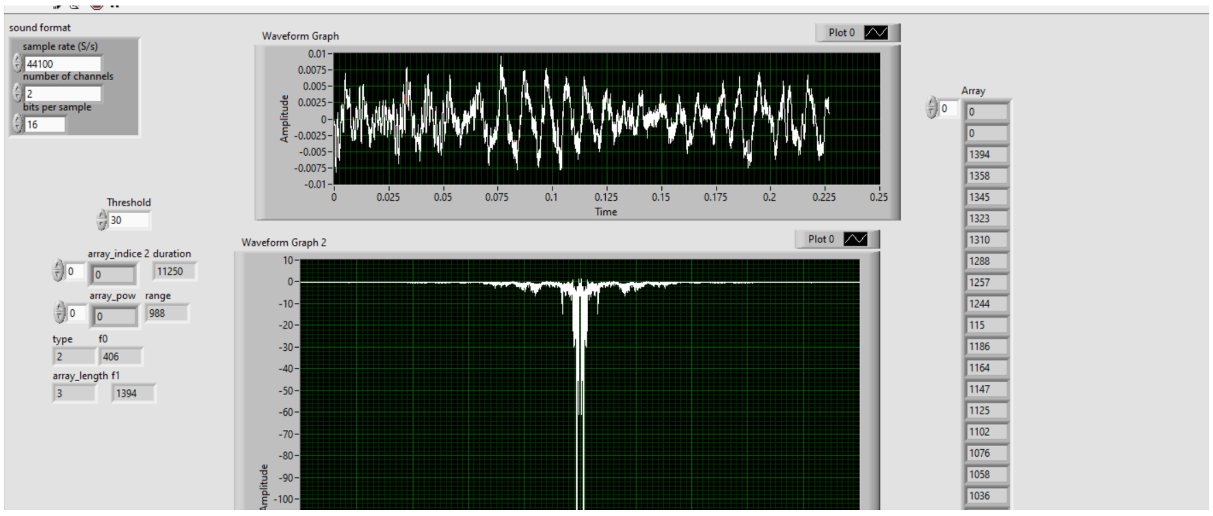
- f0 : Initial Frequency
- f1: Last Frequency
- Duration: The time-duration of the pulse in terms of milliseconds

Integration and Real-Time Execution: This module involves integrating the different components, including C shared library, LabVIEW, microphone, and loudspeaker, to create a real-time system. It ensures that the system functions seamlessly, with accurate signal detection and parameter estimation. With the implementation of the analyze algorithm in C, low computational complexity is achieved while having more control on the algorithm.

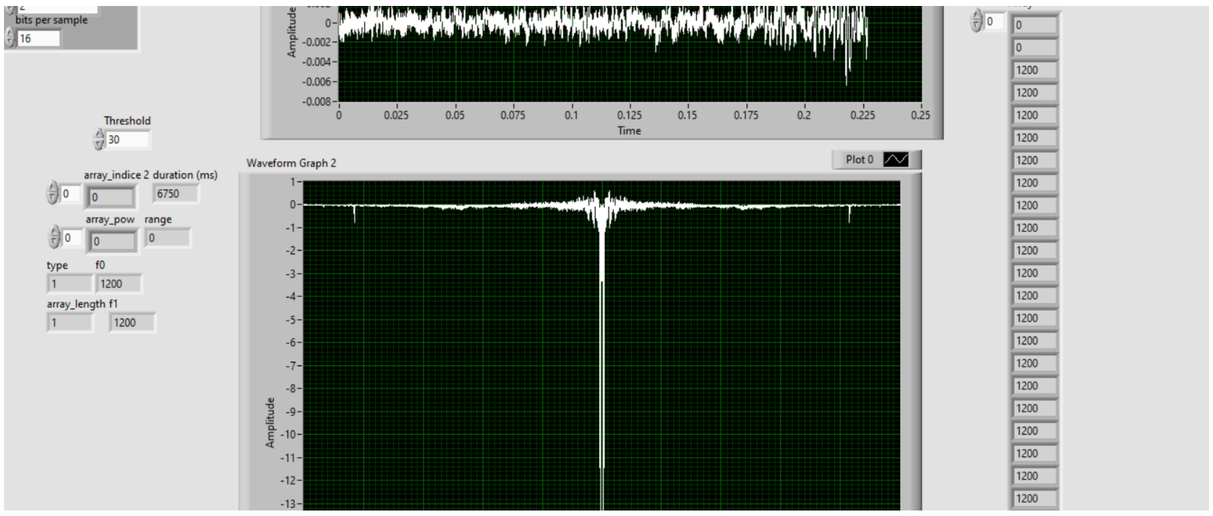
Experimental Results

İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288

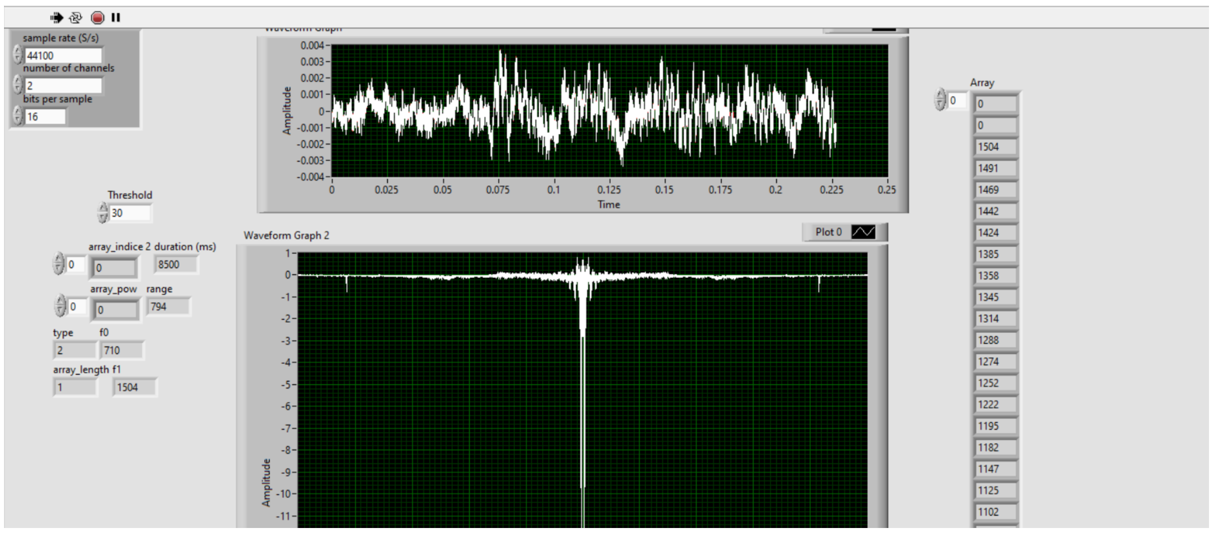
<u>Result for 400 1400 Hz Chirp for 10 secs</u>


```
sound format
sample rate (S/s)   44100
number of channels  2
bits per sample     16

Threshold  30

array_indice 2  0     duration  11250
array_pow       0     range     988
type            2     f0        406
array_length    3     f1        1394

Array
0
0
1394
1358
1345
1323
1310
1288
1257
1244
115
1186
1164
1147
1125
1102
1076
1058
1036
```

Result for 1200 Hz gaussian for 7secs


```
bits per sample  16

Threshold  30

array_indice 2  0     duration (ms)  6750
array_pow       0     range          0
type            1     f0             1200
array_length    1     f1             1200

0
0
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
```

Results for 700 1560 Hz for 8 secs


```
sample rate (S/s)   44100
number of channels  2
bits per sample     16

Threshold  30

array_indice 2  0     duration (ms)  8500
array_pow       0     range          794
type            2     f0             710
array_length    1     f1             1504

Array
0
0
1504
1491
1469
1442
1424
1385
1358
1345
1314
1288
1274
1252
1222
1195
1182
1147
1125
1102
```

İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288

Conclusion

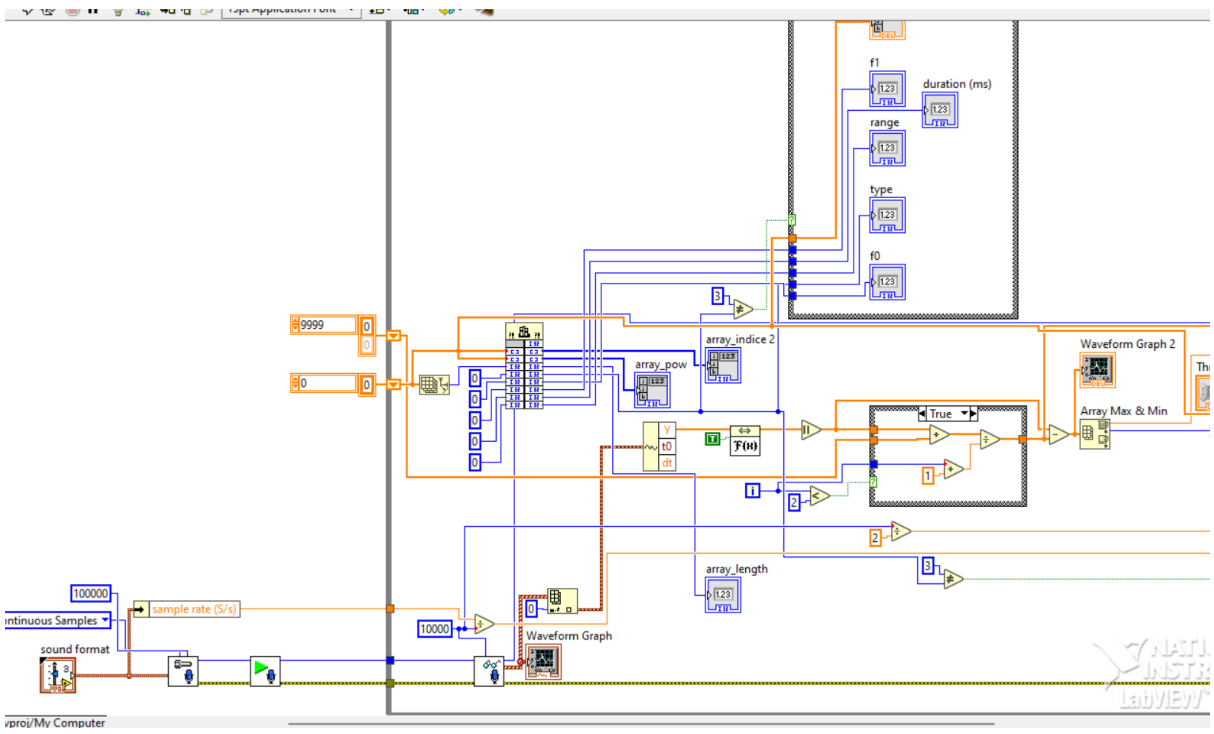
This project presented a comprehensive opportunity to explore the detection and estimation of signals and parameters in a microphone-loudspeaker system. Through the implementation of C and LabVIEW, we were able to develop a functional system that allowed for adjusting signal parameters, transmitting signals, detecting them using the receiver computer, and estimating the parameters accurately. The project emphasized the importance of the utilization of shared libraries to enhance the system's functionality and real-time performance. Using C code implementation in LabView where the computational complexity increases, the real time performance of the system is increased. During the time worked on this project; valuable experience in signal processing, programming, and the integration of different software tools are gained. Overall, this project provided a solid foundation for understanding and applying digital signal processing techniques in real-world applications.

İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288

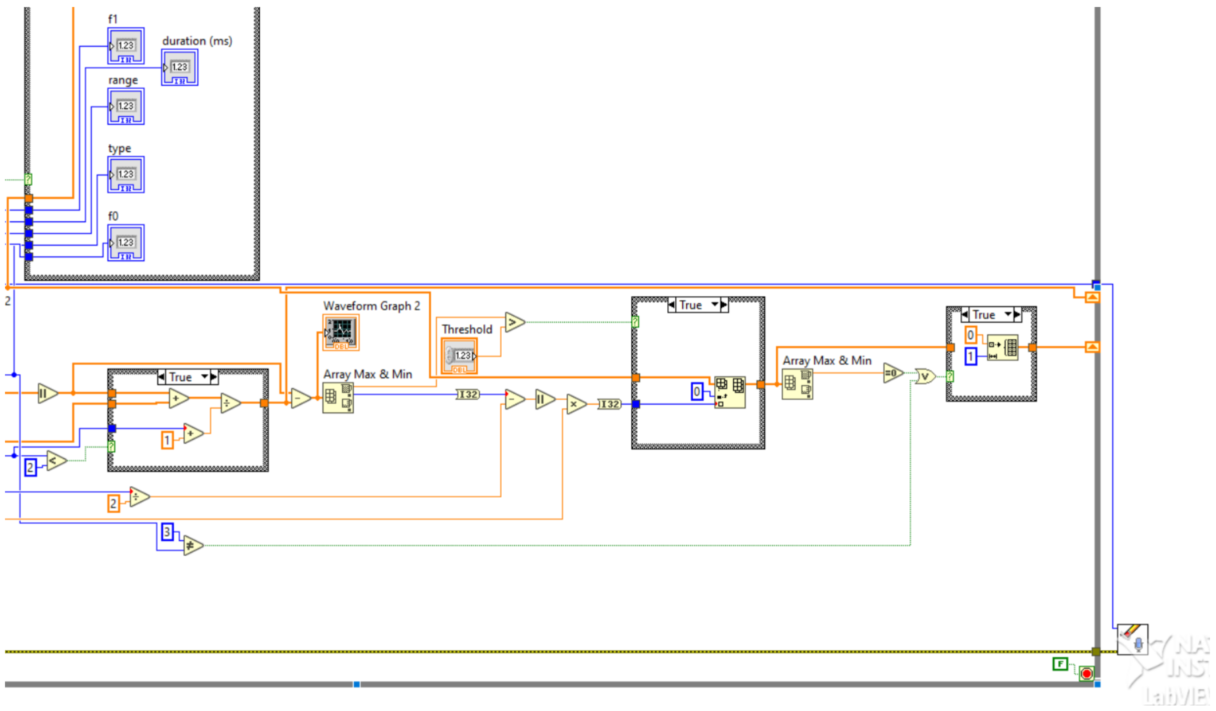
APPENDIX

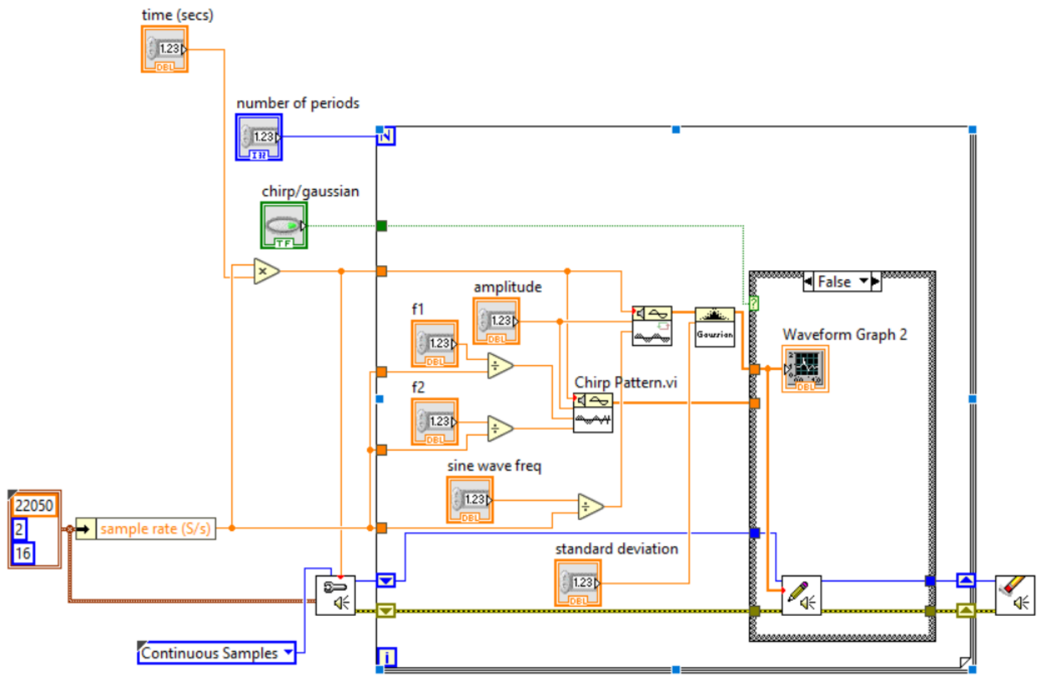
```
#include <cvidef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Exported function declarationa
int__declspec(dllexport){}

func(int *array_indice, int *array_pow, int array_length, int* type, int* f0, int*f1, int*
duration, int *range){
   int f = 4;
         *type = 3;
         *f0 = 3;
         *f1 = 3;

/*
   if (*pointer_to_string != NULL) {
      free(*pointer_to_string);
   }
*/
   //*pointer_to_string = (char *)calloc(100 + 1, sizeof(char));
   if (array_length == 1) {
      //snprintf(*pointer_to_string, 101, "no signal yet");
                  return 0;
   }

   if ((array_indice[0] == 0) && (array_indice[1] == 0)) {
      int start = array_indice[array_length - 2]; // first val = 0?
      int end = array_indice[2];
                  *range = end- start;
      if (*range < 50) {
                  *type =  1;
                  *f0 = ((start + end) / 2);
                  *f1 = ((start + end) / 2);
                  *duration = (array_length - 3) * 1000 / f;
//snprintf(*pointer_to_string, 101, "The signal is gaussian pulse with frequency=%d ,
duration=%.2f", frequency, duration);
      }
                  else{
                  *type = 2;
```

```
                    *f0 = start;
                    *f1 = end;
                    *duration = (array_length - 3) *1000 / f;
//snprintf(*pointer_to_string, 101, "The signal is chirp with start frequency=%d , end
frequency=%d, duration=%.2f", start, end, duration);
            }
            return 0;
        }
                return 0;
}
```



İhsan Atakan Adalı – 2165702
Mert Arslanoğlu – 2374437
Eren Akçanal – 2374288


f1
duration (ms)
range
type
f0
Waveform Graph 2
Threshold
True
Array Max & Min
I32
0
1
2
3
F


time (secs)
number of periods
chirp/gaussian
amplitude
f1
f2
Chirp Pattern.vi
Gaussian
False
Waveform Graph 2
sine wave freq
standard deviation
22050
2
16
sample rate (S/s)
Continuous Samples
N
i