**Department of Electrical and Computer Engineering**

**North South University**



**CSE 299.4**

**Junior Design Project**

**Faculty Reviewer Application For NSUers**

**Group 09**

**Md Kamrus Samad ID # 1813059642**

**Sajib Hossain ID # 1821245642**

**Md. Sharukh Fardin ID # 1821463042**

**Faculty Advisor:**

**Zunayeed Bin Zahir**

**Lecturer**

**ECE Department**

**Summer, 2021**

**Acknowledgement**

----------------------------------------------------In Final Report--------------------------------------------------

**Abstract**

In this report, we present a software that can be beneficial for students of North South University regarding getting information and reviews about faculties. The application will be cross-platform. The idea is to make the software available in android and windows platform. In our project, we will use Microsoft’s C# .NET framework. For making it accessible via two platforms and for creating GUI interface we will use Xamarin Forms. Being a part of our university, each semester we can see the high demand in students for knowing the most suitable teachers in their upcoming courses. There are some groups in social media that helps a bit in that case. But the information those groups provide are not always valid and there are no good ways to verify the claims the students put give on teachers. There is another problem that is anyone can put comment on those groups, even people outside of NSU or people who have no idea on the specific faculties. Our software will be NSU domain restricted. Therefore, only the students of NSU will get the access. There will be a well written and critically followed user policy that every student must follow. An admin panel will be there to observe and control activities of the students. Our main aim is to create a faculty reviewing platform for NSU students but beside of this, there will be some other student friendly features such as CGPA calculator, To-Do list and daily remainder. The software will act as a faculty evaluation and information getting tool which will be accessible only by the students of NSU.

**Table of Content**

Chapter 1: [Project Overview 1](#_Toc481763927)

[1.1 Introduction 2](#_Toc481763928)

[1.2 Speech processing and speech recognition basics 2](#_Toc481763929)

[1.2.1 How speech recognition works 3](#_Toc481763930)

[1.2.2 Applications of speech recognition 3](#_Toc481763931)

[1.2.3 Pros and Cons 3](#_Toc481763932)

[1.2.4 Performance 4](#_Toc481763933)

[1.2.5 List of popular speech recognition software 4](#_Toc481763934)

[1.3 Our proposed project 5](#_Toc481763935)

[1.3.1 Description of the idea: 5](#_Toc481763936)

[1.3.2 Difficulty 6](#_Toc481763937)

[1.4 Motivation 6](#_Toc481763938)

[1.5 Summary 7](#_Toc481763939)

Chapter 2: [Related work 8](#_Toc481763941)

[2.1 Introduction 9](#_Toc481763942)

[2.2 Existing work related to Speech Recognition on robot 9](#_Toc481763943)

[2.2.1 Robot-by-voice: Experiments on commanding an industrial robot using the human voice 10](#_Toc481763944)

[2.2.2 Voice Automated Mobile Robot 10](#_Toc481763945)

[2.2.3 Design of a Voice controlled Smart Wheelchair 11](#_Toc481763946)

[2.3 Research and publications on Sound Source Localization 11](#_Toc481763947)

[2.3.1 Robust Sound Source Localization Using a Microphone Array on a Mobile Robot 12](#_Toc481763948)

[2.3.2 Localization Estimation of Sound Source by Microphones Array 13](#_Toc481763949)

[2.3.3 Real-Time Sound Source Localization on an Embedded GPU Using a Spherical Microphone Array 13](#_Toc481763950)

[2.3.4 Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures 14](#_Toc481763951)

[2.4 Summary 14](#_Toc481763952)

Chapter 3: [Theory 15](#_Toc481763954)

[3.1 Introduction 16](#_Toc481763955)

[3.2 Speech Recognition 16](#_Toc481763956)

[3.3 Sound source localization 18](#_Toc481763957)

[3.3.1 Theory 19](#_Toc481763958)

[3.3.2 Algorithm 20](#_Toc481763959)

[3.3.3 Equations and calculation 21](#_Toc481763960)

[3.4 Summary 23](#_Toc481763961)

Chapter 4: [Structure of the system 24](#_Toc481763963)

[4.1 25](#_Toc481763964)

[4.2 Procedure and Functionality 25](#_Toc481763965)

[4.2.1 Procedure 25](#_Toc481763966)

[4.2.2 Functions 26](#_Toc481763967)

[4.3 Workflow and Algorithms 29](#_Toc481763983)

[4.3.1 Outline of the workflow 29](#_Toc481763984)

[4.3.2 Explanation of the algorithm for command execution 30](#_Toc481763986)

[4.4 33](#_Toc481763987)

[4.5 35](#_Toc481763988)

Chapter 5: [Google Speech Recognition API 36](#_Toc481763990)

[5.1 Introduction 37](#_Toc481763991)

[5.2 Google Cloud Speech API 37](#_Toc481763992)

[5.3 Speech API features 37](#_Toc481763993)

[5.4 Google Speech Recognition API vs. Nuance Dragon Speech 38](#_Toc481763994)

[5.5 Google Speech Recognition API vs. Siri 39](#_Toc481763995)

[5.6 Google Speech Recognition API vs. Microsoft Cortana 39](#_Toc481763996)

[5.7 Summary 40](#_Toc481763997)

[Chapter 6:](#_Toc481763998) [The Microcontroller used in this system: Arduino Mega 41](#_Toc481763999)

[6.1 Introduction 42](#_Toc481764000)

[6.2 Arduino Mega 42](#_Toc481764001)

[6.2.1 Hardware 43](#_Toc481764002)

[6.2.2 Application 44](#_Toc481764003)

[6.3 Pin out diagram of Arduino MEGA 45](#_Toc481764004)

[6.4 Summary 45](#_Toc481764005)

[Chapter 7](#_Toc481764006): [Modules used in this system 46](#_Toc481764007)

[7.1 Introduction 47](#_Toc481764008)

[7.2 Ultrasonic Ranging Module HC - SR04 47](#_Toc481764009)

[7.3 Bluetooth Module (hc-05) 48](#_Toc481764010)

[7.4 Force Sensitive Resistor (FSR 400) 50](#_Toc481764011)

[7.5 Motor Driver L293D 51](#_Toc481764012)

[7.5.1 Dual H-bridge logic 52](#_Toc481764013)

[7.5.2 Table of operation 53](#_Toc481764014)

[7.6 Summary 53](#_Toc481764015)

[Chapter 8](#_Toc481764016): [Mechanical Description 54](#_Toc481764017)

[8.1 Introduction 55](#_Toc481764018)

[8.2 Mechanical measurements and explanations of 1st model 55](#_Toc481764019)

[8.3 Overview of 2nd model and 3rd model 59](#_Toc481764020)

[8.4 Summary 61](#_Toc481764021)

[Chapter 9](#_Toc481764022): [Compliance with standards 62](#_Toc481764023)

[9.1 Introduction 63](#_Toc481764024)

[9.2 Compliance with IEEE standards 63](#_Toc481764025)

[9.3 Compliance with US standards 63](#_Toc481764026)

[9.4 Compliance with European standards 63](#_Toc481764027)

[9.5 Summary 64](#_Toc481764028)

[Chapter 10](#_Toc481764029): [Design Impact 65](#_Toc481764030)

[10.1 Introduction 66](#_Toc481764031)

[10.2 Economic impact 66](#_Toc481764032)

[10.3 Environmental impact 66](#_Toc481764033)

[10.4 Social impact 66](#_Toc481764034)

[10.5 Political impact 66](#_Toc481764035)

[10.6 Ethical impact 66](#_Toc481764036)

[10.7 Health and safety impact 67](#_Toc481764037)

[10.8 Manufacturability 67](#_Toc481764038)

[10.9 Sustainability 67](#_Toc481764039)

[10.10 Summary 67](#_Toc481764040)

[Chapter 11](#_Toc481764041): [Results and Discussion 68](#_Toc481764042)

[11.1 Introduction 69](#_Toc481764043)

[11.2 Result and findings of sound source localization 69](#_Toc481764044)

[11.3 Result and findings of speech recognition 71](#_Toc481764045)

[11.3.1 Limitations 74](#_Toc481764046)

[11.4 Summary 74](#_Toc481764047)

[Chapter 12](#_Toc481764048): [Conclusion 75](#_Toc481764049)

[Bibliography 77](#_Toc481764050)

[Appendices 82](#_Toc481764051)

[Appendix A](#_Toc481764052): [Codes Related to Arduino](#_Toc481764053) 83

[Appendix B](#_Toc481764054): [Codes for android application interface](#_Toc481764055) 89

[Appendix C](#_Toc481764056): [Codes for Android programming](#_Toc481764057) 93

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Caption** | **Page No.** |
| **3.1** | Locating sound source using 3 microphones | 21 |
| **4.1** | Workflow diagram of the Android part of the robot | 29 |
| **4.2** | Workflow diagram of the Algorithm of command execution in the microcontroller | 30 |
| **4.3** | Block diagram of robot circuit | 33 |
| **4.4** | Schematic diagram of Block A | 34 |
| **4.5** | Schematic diagram of Block B | 34 |
| **6.1** | Arduino MEGA | 43 |
| **6.2** | Pin out diagram of Arduino MEGA | 45 |
| **7.1** | Sonar sensor module hc-sr04 | 48 |
| **7.2** | Bluetooth module hc-05 | 49 |
| **7.3** | Force sensitive resistor FSR-400 | 51 |
| **7.4** | Pin diagram | 52 |
| **7.5** | H-bridge circuit | 52 |
| **7.6** | Operation of H-bridge in L293D | 53 |
| **8.1** | Side view of the robot | 56 |
| **8.2** | Upper view of the robot | 57 |
| **8.3** | The gripper construction | 58 |
| **8.4** | Model 1 trying to pick up object. | 59 |
| **8.5** | The robotic arm 2nd version upper view and side view | 60 |
| **8.6** | 2nd model and 3rd model of the robot | 60 |
| **11.1** | Sound raw data analysis and Amplitude detection | 69 |
| **11.2** | Data from 3 microphones | 70 |
| **11.3** | Robot lifting “very heavy” category object | 73 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Caption** | **Page No.** |
| **4.1** | Object categorization table A | 32 |
| **4.2** | Object categorization table B | 32 |
| **11.1** | Frequency analysis table | 70 |
| **11.2** | Coordinate analysis table | 71 |
| **11.3** | Time required to lift different object category | 72 |
| **11.4** | Time required to lift different object category | 72 |

# Chapter 1

# Project Overview

### 1.1 Introduction

## 

## 1.3 Our proposed project

### 

### 1.3.1 Description of the idea:

### 

### 1.3.2 Problem Statement

## 

## 1.4 Motivation

## 

## 1.5 Summary

# Chapter 2

# Related work/Literature Review

## 2.1 Introduction

## 

## 2.2 Existing work related to Speech Recognition on robot

### 2.2.1 Robot-by-voice: Experiments on commanding an industrial robot using the human voice

### 

### 2.2.2 Voice Automated Mobile Robot

### 

### 2.2.3 Design of a Voice controlled Smart Wheelchair

## 

## 2.3 Research and publications on Sound Source Localization

### 

### 2.3.1 Robust Sound Source Localization Using a Microphone Array on a Mobile Robot

### 

### 2.3.2 Localization Estimation of Sound Source by Microphones Array

### 

### 2.3.3 Real-Time Sound Source Localization on an Embedded GPU Using a Spherical Microphone Array

### 

### 2.3.4 Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures

## 

## 2.4 Summary

# Chapter 3

# Theory

## 3.1 Introduction

The details of the theory of our system are discussed in this chapter. The theoretical explanation is divided into two sections

1. Speech recognition
2. Sound source localization

## 3.2 Speech Recognition

To convert speech to on-screen text or a computer command, a computer has to go through several complex steps. When you speak, you create vibrations in the air. The **analog-to-digital converter (ADC)** translates this analog wave into digital data that the computer can understand. To do this, it **samples**, or digitizes, the sound by taking precise measurements of the wave at frequent intervals. The system filters the digitized sound to remove unwanted noise, and sometimes to separate it into different bands of **frequency** (frequency is the wavelength of the sound waves, heard by humans as differences in pitch). It also normalizes the sound, or adjusts it to a constant volume level. It may also have to be temporally aligned. People don't always speak at the same speed, so the sound must be adjusted to match the speed of the template sound samples already stored in the system's memory.

Next the signal is divided into small segments as short as a few hundredths of a second, or even thousandths in the case of **plosive consonant sounds** -- consonant stops produced by obstructing airflow in the vocal tract – like "p" or "t." The program then matches these segments to known **phonemes** in the appropriate language. A phoneme is the smallest element of a language – a representation of the sounds we make and put together to form meaningful expressions. There are roughly 40 phonemes in the English language (different linguists have different opinions on the exact number), while other languages have more or fewer phonemes.

The next step seems simple, but it is actually the most difficult to accomplish and is the focus of most speech recognition research. The program examines phonemes in the context of the other phonemes around them. It runs the contextual phoneme plot through a complex statistical model and compares them to a large library of known words, phrases and sentences. The program then determines what the user was probably saying and either outputs it as text or issues a computer command.

Early speech recognition systems tried to apply a set of grammatical and syntactical rules to speech. If the words spoken fit into a certain set of rules, the program could determine what the words were. However, human language has numerous exceptions to its own rules, even when it's spoken consistently. Accents, dialects and mannerisms can vastly change the way certain words or phrases are spoken. Imagine someone from Boston saying the word "barn." He wouldn't pronounce the "r" at all, and the word comes out rhyming with "John." Or consider the sentence, "I'm going to see the ocean." Most people don't enunciate their words very carefully. The result might come out as "I'm goin' da see tha ocean." They run several of the words together with no noticeable break, such as "I'm goin'" and "the ocean." Rules-based systems were unsuccessful because they couldn't handle these variations. This also explains why earlier systems could not handle continuous speech – you had to speak each word separately, with a brief pause in between them.

Today's speech recognition systems use powerful and complicated **statistical modeling systems**. These systems use probability and mathematical functions to determine the most likely outcome. According to John Garofolo, Speech Group Manager at the Information Technology Laboratory of the National Institute of Standards and Technology, the two models that dominate the field today are the Hidden Markov Model and neural networks. These methods involve complex mathematical functions, but essentially, they take the information known to the system to figure out the information hidden from it.

The Hidden Markov Model is the most common, so we'll take a closer look at that process. In this model, each phoneme is like a link in a chain, and the completed chain is a word. However, the chain branches off in different directions as the program attempts to match the digital sound with the phoneme that's most likely to come next. During this process, the program assigns a probability score to each phoneme, based on its built-in dictionary and user training.

This process is even more complicated for phrases and sentences – the system has to figure out where each word stops and starts. The classic example is the phrase "recognize speech," which sounds a lot like "wreck a nice beach" when you say it very quickly. The program has to analyze the phonemes using the phrase that came before it in order to get it right.

If a program has a vocabulary of 60,000 words (common in today's programs), a sequence of three words could be any of 216 trillion possibilities. Obviously, even the most powerful computer can't search through all of them without some help.

That help comes in the form of program training.

These statistical systems need lots of exemplary training data to reach their optimal performance – sometimes on the order of thousands of hours of human-transcribed speech and hundreds of megabytes of text. These training data are used to create acoustic models of words, word lists, and multi-word probability networks. There is some art into how one selects, compiles and prepares this training data for "digestion" by the system and how the system models are "tuned" to a particular application. These details can make the difference between a well-performing system and a poorly-performing system – even when using the same basic algorithm.

## 3.3 Sound source localization

Localizing the source of sound from audio is challenging due to continuous change of frequency, noise, amplitude and many other factors which effect the nature of sound all the time. There were many ways of localizing the source.

In this project the key idea is to locate sound source in 3 dimensional system using Stokes-Law. We used three microphones placed in a specific distance in a triangular form where centroid of the triangle is considered as origin in 3-dimensional space. We used Stokes law for distance calculation through amplitude measuring from the three microphones. We calculated the frequency of the sound using Digital Signal Processing and auto correlation. This frequency was applied in Stokes law to find the distance of the source from a specific microphone. We used three microphones to get three distances of the same source which provided us three equations to find out the source location in 3 dimensional space. Other than these main focuses, we also had to work on tuning the microphones as the readings were different and amplitudes we received were different too and response to sound was different too. We used Proportional Integrated Derivative (PID) method to synchronize the readings of the microphones.

In this paper we derived a way to locate the sound source its direction and distance in 3 dimensional space using only 3 microphones and a microcontroller for the real time calculation. We used Stokes Law for distance sensing. Our research provides 85% accuracy of sound source localization and has 8 meter distance accuracy from the source.

### 3.3.1 Theory

Stokes law of sound attenuation [5] tells us that in a Newtonian fluid [6] due to fluid viscosity amplitude of a plane wave decreases exponentially respect to the distance travelled. As air is a Newtonian fluid so the Stokes Law is applicable in the case of Air. Stokes Law provides us the following equation.

Where n = dynamic viscosity coefficient, ω= sound frequency, p is fluid density and V is the speed of sound. Apart from the frequency all of the other components in this equation is constant for a certain Newtonian fluid. Again for an isotropic and homogenous fluid as air if we apply Stokes Law we will get the following equation.

Where Aₒ is the original amplitude of the sound source and A(d) is the amplitude of the sound at a distance d and α is the result of the Stokes law equation. If we can find the frequency of the sound source we will be able to achieve α and if we get three equation from three microphones places in a specific orientation in this case in an equilateral triangle formation we can easily get the position of the sound source in 3 Dimensional space.

### 3.3.2 Algorithm

As we will be working with the raw data received from the microphone we have to process the data to get the frequency. For that we used Auto Correlation [7] and Fast Fourier Transformation (FFT) [8] to get a smooth sinusoidal signal. We calculated the frequency using Pitch Determination [7] from auto correlation and calculating the distance between pitch and then we applied the Proportional Integral Derivative (PID) technique for high accuracy for frequency. Getting the average frequency from the three microphones we estimated the original frequency which was applied in Stokes Law and the rest is as the theory stated. We achieved three equation in order to get the position in Three Dimensional Coordinate system.

1. Construct an equilateral triangle where 3 microphones will be placed at the three vertex of the triangle.
2. Measure the equal distance between the vertices.
3. Microcontroller will receive the raw data of 3 microphones.
4. Microcontroller will process the raw data using Fast Fourier Transformation [8].
5. The result of the Fast Fourier Transformation will be auto correlated [7] to get a clean signal.
6. Frequency will be detected using time interval between the two pitches.
7. α will be generated from the equation below.
8. As amplitude at the microphone is Known we will be able to get three equations from the given formula
9. From the three equations we will be able to generate the position of the source in 3 Dimensional coordinate system.

### 3.3.3 Equations and calculation

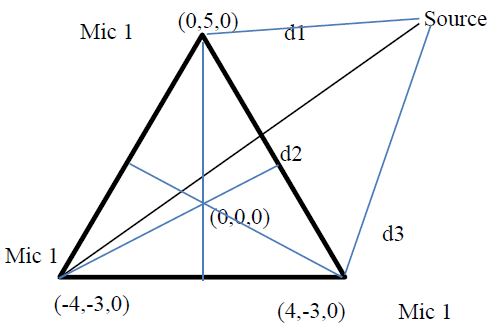
In order to find out the three equation let us consider an equilateral triangle

Fig 3.1 Locating sound source using 3 microphones

Let d1, d2, d3 be the distance of microphones from the source. We have the following equation of three microphones.

Where we receive α from FFT [8] and Auto Correlation [7] and Aₒ is original amplitude and , , are amplitudes of the microphone received respectively.

As Aₒ is the same for three equations then we can write the following

From equation (5) and (6) we can get

Similarly we can get

From the three equations we can derive the three distances from which we can derive the three equations for the coordinate system to find the position of a source in 3 Dimensional system.

In order to increase the efficiency we also established an error calculation system in which it calculated the errors and found out the possible accuracy. We did that calculation using Proportional Integral Derivative (PID) technique. In this technique we calculated the error estimation and did a feed back to the circuit and it again calculated the error in again did the feedback. Through continuous feedback we calculated the possible positioning until we get the lowest possible error. In our experiment we were able to reduce the error margin significantly having an accuracy of 8 meters.

## 3.4 Summary

In this chapter, the theoretical part of our project has been described. We have tried to explain how speech recognition works in detail, and the theory of how we can locate the source of a sound in 3D-coordinate system using Inverse square law of sound propagation has also been explained. All necessary equations and calculations have been shown.

# Chapter 4

# Structure of the system

## 4.1 Introduction

## 

## 4.2 Procedure and Functionality

### 

### 4.2.1 Procedure

### 4.2.2 Functions

## 

## 4.3 Workflow and Algorithms

### 

### 4.3.1 Outline of the workflow

### 4.3.2 Explanation of the algorithm for command execution

## 4.4 Equipment and Schematic Diagrams

## 4.5 Summary

# Chapter 5

# Google Speech Recognition API

## 5.1 Introduction

## 

## 5.2 Google Cloud Speech API

## 

## 5.3 Speech API features

## 

## 5.4 Google Speech Recognition API vs. Nuance Dragon Speech

## 5.5 Google Speech Recognition API vs. Siri

## 

## 5.6 Google Speech Recognition API vs. Microsoft Cortana

## 5.7 Summary

# Chapter 6

# The Microcontroller used in this system: Arduino Mega

## 6.1 Introduction

## 

## 6.2 Arduino Mega

### 

### 6.2.1 Hardware

## 6.3 Pin out diagram of Arduino MEGA

## 6.4 Summary

# Chapter 7

# Modules used in this system

## 7.1 Introduction

## 7.2 Ultrasonic Ranging Module HC - SR04

## 7.3 Bluetooth Module (hc-05)

## 

## 7.4 Force Sensitive Resistor (FSR 400)

## 

## 7.5 Motor Driver L293D

### 

### 7.5.1 Dual H-bridge logic

### 

### 7.5.2 Table of operation

## 7.6 Summary

# Chapter 8

# Mechanical Description

## 8.1 Introduction

## 

## 8.2 Mechanical measurements and explanations of 1st model

## 8.3 Overview of 2nd model and 3rd model

## 

## 8.4 Summary

# Chapter 9

# Compliance with standards

## 9.1 Introduction

## 

## 9.2 Compliance with IEEE standards

## 

## 9.3 Compliance with US standards

## 

## 9.4 Compliance with European standards

## 

## 9.5 Summary

# Chapter 10

# Design Impact

## 10.1 Introduction

## 

## 10.2 Economic impact

## 

## 10.3 Environmental impact

## 

## 10.4 Social impact

## 

## 10.5 Political impact

## 

## 10.6 Ethical impact

## 

## 10.7 Health and safety impact

## 

## 10.8 Manufacturability

## 

## 10.9 Sustainability

## 

## 10.10 Summary

# Chapter 11

# Results and Discussion

## 11.1 Introduction

## 

## 11.2 Result and findings of sound source localization

## 

## 11.3 Result and findings of speech recognition

### 11.3.1 Limitations

## 

## 11.4 Summary

# Chapter 12

# Conclusion

In Final Report

# Bibliography

1. Norberto Pires, J. (2005). Robot-by-voice: Experiments on commanding an industrial robot using the human voice. *Industrial Robot: An International Journal*, *32*(6), 505-511.
2. Jain, R., & Saxena, S. K. (2011). Voice Automated Mobile Robot. *International Journal of Computer Applications*, *16*(2), 32-35.
3. Abed, A. A. Design of Voice Controlled Smart Wheelchair.
4. Singh, A., Singh, D., & Dwivedy, S. K. Design And Fabrication Of A Gripper For Grasping Irregular Objects. *Department of Mechanical Engineering, Indian Institute of Technology, Guwahati*.
5. Deaconescu, T., & Deaconescu, A. (2011). Pneumatic Muscle Actuated Gripper. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 2).

# Appendices

# Appendix A

**Codes Related to Arduino**

In Final Report

# Appendix B

**Codes for android application interface**

In Final Report

# Appendix C

**Codes for Android programming**

In the Final Report