Voice Recognition System Design Aspects for Robotic Car Control

Ayesha Shafiq[1†], Humera Tariq[2†], Fareed Alvi[3†] and Usman Amjad[4†],
aieeshashafique@gmail.com humera@uok.edu.pk faridalvi@email.com usmanamjad87@gmail.com
University of Karachi, Karachi, Pakistan

Abstract
Robotic products are introduced to provide ease and convenience to human beings. Various researchers are working for designing and improving the concepts of robotic cars. In this paper we have presented a voice controlled model for robotic car. The proposed model uses MEL Frequency Cepstrum Coefficient (MFCC) and Hidden Markov Model (HMM) for voice recognition. MFCC is used for feature extraction while HMM is used to perform recognition task. Based on the recognized command the car will perform certain actions. Complete architecture of the proposed system including mobile application, recognition algorithm and hardware design. The model can be extended to various applications and may prove useful for special or handicapped people.

Key words:
Hidden Markov Model, Voice recognition, Robotics.

1. Introduction

Speech recognition is the methodology used to recognize spoken words by a speaker. It is a very useful area having many applications in our daily life. Speech recognition enables a computer to capture the words spoken by a human with the help of microphone [1]. These words are later on recognized by speech recognizer and in the end, system outputs the recognized words [2]. It is the discipline of communicating with the computer using voice, and having it correctly understood by the computer. Generally, speech recognizer is a machine which recognize humans and their vocalized word in some way and can act thereafter [3]. Today speech recognition is automating lot of things in our environment and easing our lives [4]. We can control our homes, gadgets and machines through our voice [5] [6]. Although different techniques have been developed for automatic speech recognition, ranging from knowledge-based systems to neural networks, the main engine behind this progress and currently the dominant technology, has been the data driven statistical approach based on Hidden Markov models [7] [8]. In this paper, we have used Mel Frequency Cepstrum Coefficient (MFCC) for feature extraction and Hidden Markov Model (HMM) for temporal pattern to recognize speech input [9]. Hidden Markov Model (HMM) is often used now a day in Speech recognition [10] [11]. In this paper, we are going to present the design and structure of voice Controlled Robotic car using Hidden Markov Model (HMM) and Mel Frequency Cepstrum Coefficient (MFCC) for feature extraction.

Manuscript received January 5, 2019
Manuscript revised January 20, 2019

2. Requirement Analysis

- The main objective is to build a system that can detect the familiar voice and should not respond to unauthorized users. Using this type of security feature; we can make the system efficient and help users to protect their vehicles. Generally, these kinds of systems are known as Speech Controlled Automation System(SCAS). Our system will be a prototype of the same. We are not aiming to build a robot which can recognize a lot of words. Our basic idea is to develop some sort of menu drive control of our robot. For example, when the recognized or familiar speaker speaks FORWARD, robot should recognize this speech and move forward. When the speaker says BACKWARD it should move backward. When the speaker says TURN LEFT, it should turn left. When the speaker says TURN 30° AROUND, it should turn around 30°. When the speaker says STOP, it should stops doing current task. The robot should have some user feedback, such as if the robot doesn't understand the user commands, it gives the user feedback- "I don't understand". Our Robot should understand the complex statement as well. For example, when speaker says MOVE 10 CM BACKWARD AND TURN RIGHT, the car should move back to 10 cm distance and then should turn to right. Our Robot should also capable of identifying obstacles and has a capability to avoid them to happen. There can be other features as well in the Robotic car like when speaker says PLAY MUSIC, it will ask for the track and music will be played. When the speaker says STOP MUSIC, it will stop the music. Other features can be opening door when the speaker uttered OPEN THE DOOR and closing the door when the speaker uttered CLOSE THE DOOR. Our system should recognize voice commands efficiently and should perform following action in response: (1) Build Robotic Car (2) Controlled using familiar voice commands (3) Recognize Uttered Speech (4) Action in response to authorized voice (5) Should protect vehicles (6) Menu driven (8) Commands to recognize Move, Turn R/L, Stop, Turn 30 around (9) Should response to open/close door, start/ stop music (10) Should provide user feedback such as NR, NU (11) Should understand complex statement, Move x cm F/B/T (12) should identify obstacles (13) Capability to avoid obstacles. It is important to distinguish between need and wishes at this stage for e.g. module which are need of desired system are: Voice recognizer, Speech recognizer and Visual recognizer while wish list includes: Robotic car should understand low voice as well as loud, Robotic car should looks like a real car. Maintenance of sample requirement analysis table is very helpful in future change management as shown in Table 1.

Table 1 Sample Requirement Analysis Table

ID	People	Process	Rules	Tools	
				SW	HW
2	Yes	Yes	Yes	Yes	Yes
3	-	Yes	Yes	Yes	Yes
4	Yes	--	Yes	Yes	Yes
5	Yes	--	Yes	-	-
6	-	-	-	Yes	Yes
7	-	Yes	-	Yes	Yes

3. High Level Design

The high-level design is the complete picture of our project on which we are going to work. As we are going to build voice robotic control car, it's high level design comprise of three main components, Smart phone, Arduino and robotic car. The high level design is showing in Fig. 1 and below we discuss its components one by one:

- **Smart phone**

Smart phone is the component in which we have kept our Voice recognition software. First, we will capture the voice using smart phone and then this voice message is sent to voice recognition software to verify that this voice is of authorized user or not and if it is of authorized user then it will decode the command and would send signal to Arduino to do action and in case if the voice is of other than its owner, it will simply tell that it is not legal.

- **Arduino**

Arduino is a microcontroller and will do the controlling of input and output signals. It is connected to smart phone using USB cable. When the message is decoded using voice recognition software, it will signal 0, 1 to particular motor depending on the message content.

- **Robotic Car**

Robotic car is the toy-based car in which we have the functionalities of moving the car forward/backward, turning car left and right, opening and closing the door, and stopping the car. These operations are done by regular motors use in toy cars. These motors are controlled by Arduino. On getting enable signal for the particular motor, it will do the operation.

4. Architectural Design

As already discussed in high level design section, the three main components are: smart phone, Arduino and robotic car. First component is a smart phone that could be kept inside of a car between car seats or could be bound to the roof of a car. Any authorized user would send its voice through Bluetooth signal to smart phone, as smart phone can detect the voice. The purpose of using the smart phone is to detect the voice or to capture the voice through Bluetooth signal and the other main purpose is to keep the voice recognizer application in smart phone. When the voice signal would be captured through Bluetooth, it would be proceeded to voice recognizer application that is built in android. The voice recognizer would recognize the voice, and would send a message to Arduino through serial communication.

The Arduino message is basically a structure that would consist of 4 parameters. These parameters are message start, message ID, delay and message end respectively. Message start will tell Arduino that it is the start of new message, or new command is ordered. Message ID will tell Arduino that which pin is this, either it is turn left, turn right, move forward, move backward, open door, close door, stop etc. Delay will tell that for how much time, it would give delay after turning this specific motor. Message end will tell Arduino that the message is end. When the Arduino will get the message using com port, then it will read the message or decode the message by tracking its Message ID. After decoding the message structure, Arduino will enable that pin no against some Message ID, and would give some delay with whatever delay value, this message has passed to Arduino. On the basis of Message ID, Arduino will send signal to enable that pin. If the Message structure consist of AE, 1, 3000, AF, then how Arduino will decode it? AE tells as discussed above that it is the start of new message, 1 is the Message ID, 3000 is the delay time of 3000 milliseconds, and AF is the end of that message. Message ID 1 tells Arduino to Turn Right car, and give delay of 3000 milliseconds after turn and it is the way that how voice controlled robotic car will work. The system architecture is given in Figure 2.

5. Low Level Design

5.1 Controlling car through Arduino

Arduino would work with two motor driver chips, Lx293D to control the voice controlled robotic car [12]. Arduino mega 2560 board is used, which has 54 digital input/output pins, from which 20 pins are used in our system. Lx293D driver motors chips are used. Each chip has 2 motors driver. Each driver has 2 input pins and two output pins.

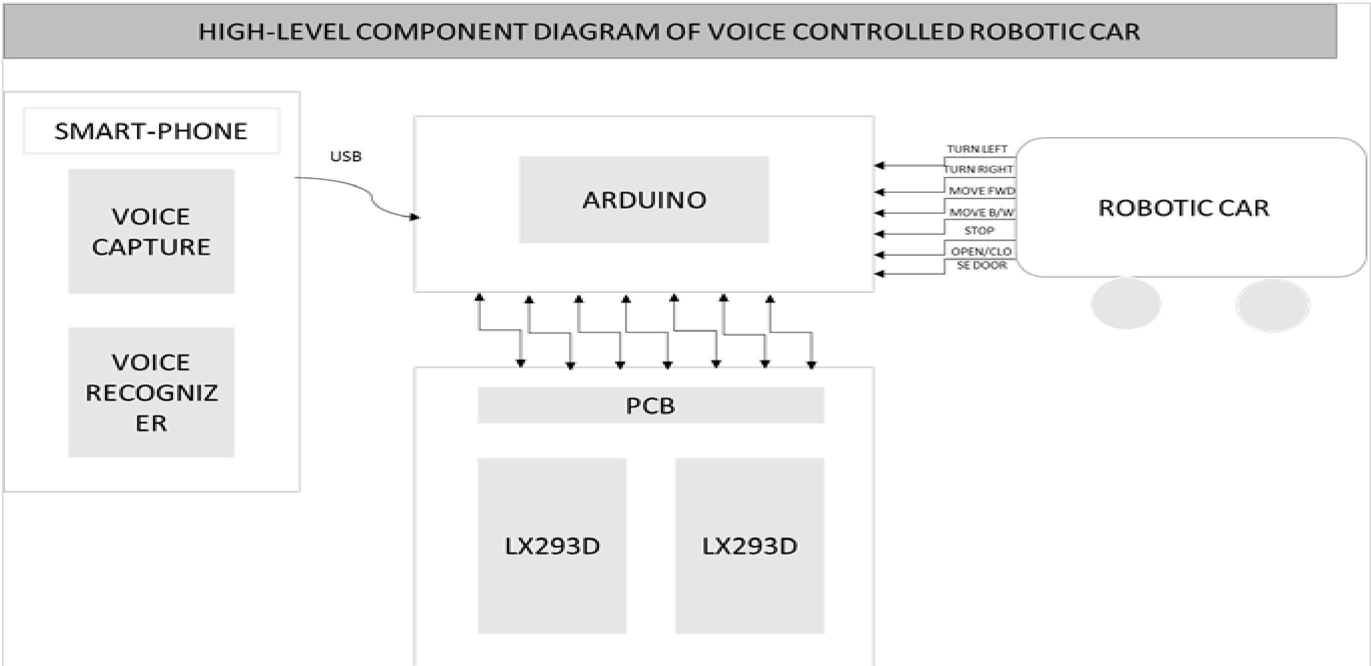


Fig 1: Architecture diagram of proposed system

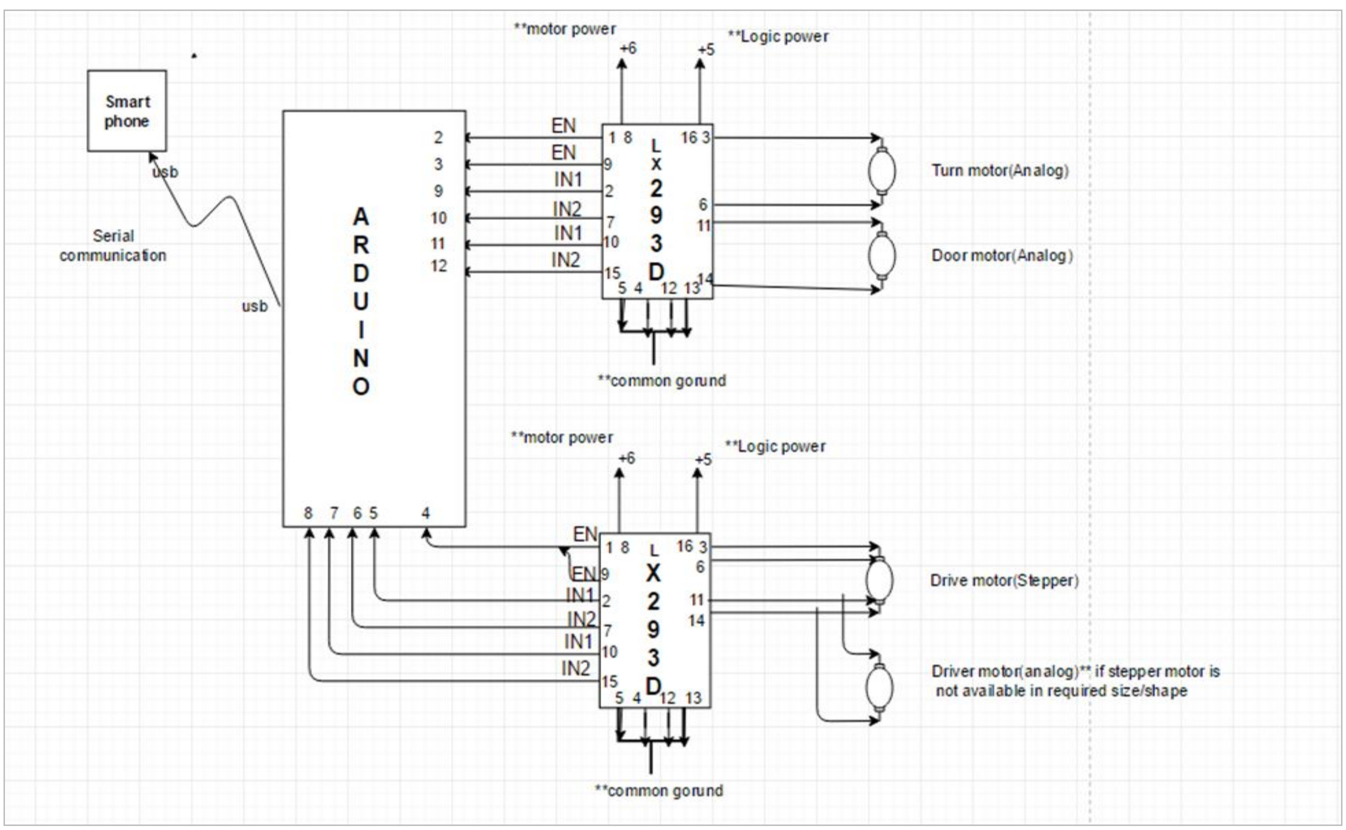


Fig 2: Architecture diagram of proposed system

Manuscript received January 5, 2019
Manuscript revised January 20, 2019

Pin number 8 of each IC will be used to gain motor power from battery pack. Battery pack is the external power supply to run motors. Pin no 16 of each IC will be Each IC has 4 input pins and 2 input pins for each driver motors. Each IC has 4 pins that will have common ground. connected to Arduino's Vcc that will supply 5 volts to IC to start up. These connections of ICs are built on Printed Circuit Board. Basically, we have run 3 analogue motors. Driver motor that will allow the car to move forward or backward, turn motor to allow car to turn left or right and door motor to open or close the door.

The question is how Arduino and the connection of ICS that are built on PCB are connected to each other to control those 3 motors? Here is the detailed description of these connections:

- IC#1 's (DRIVER MOTOR) pin number 1, that is one of the Enable pins of this IC is connected to Arduino's digital pin number 4.
- IC#1 's (TURN MOTOR) pin number 9, that is one of the Enable pins of this IC is connected to Arduino's digital pin number 7.
- IC#1 's (DRIVER MOTOR) pin number 2, that is one of the input pins of this IC is connected to Arduino's digital pin number 3.
- IC#1 's (DRIVER MOTOR) pin number 7, that is one of the input pins of this IC is connected to Arduino's digital pin number 2.
- IC#1 's (TURN MOTOR) pin number 15, that is one of the input pins of this IC is connected to Arduino's digital pin number 5.
- IC#1 's (TURN MOTOR) pin number 10, that is one of the input pins of this IC is connected to Arduino's digital pin number 6.
- IC#2 's (DOOR MOTOR) pin number 1 and 9, that are Enable pins of this IC is connected to Arduino's digital pin number 8.
- IC#2 's (DOOR MOTOR) pin number 10, that is one of the input pins of this IC is connected to Arduino's digital pin number 9.
- IC#2 's (DOOR MOTOR) pin number 15, that is one of the input pins of this IC is connected to Arduino's digital pin number 10.

There are 4 more pins of Arduino that we would use to drive the overall connections.

- VCC: - 5 volts provided by Arduino
- GND: - common ground for both Arduino and ICS.
- GND: - common ground for both Arduino and ICS.
- Vin: - provided by battery pack to the Arduino, around 8-9volts.

5.2 Message processing in Arduino

There is some sequence of steps followed by Arduino to process the message.

• Wait for the message on serial port. As we have already discussed above that every message start from "AE", so when it will see "AE" on serial port, it will understand that it is the start of the new message and would ready to receive the message.
• It will open the message body
• Read command
• Switch case

1. Case Turn Right
- Make pin no 7 (enable pin) of Arduino's=1
- Make pin no 5 of Arduino's =1
- Make pin no 6 of Arduino's =0

2. Case Turn Left
- Make pin no 7 (enable pin) of Arduino's=1
- Make pin no 5 of Arduino's =0
- Make pin no 6 of Arduino's =1

3. Case Move Forward
- Make pin no 4 (enable pin) of Arduino's
- Make pin no 3 of Arduino's =1
- Make pin no 2 of Arduino's =0

4. Case Move Backward
- Make pin no 4 (enable pin) of Arduino's=1
- Make pin no 3 of Arduino's =0
- Make pin no 2 of Arduino's =1

5. Case Stop
- Make enable pin 1 that is connected to Arduino's pin number =0, regardless of input pins

6. Case Open the door
- Make pin no 8 (enable pin) of Arduino's=1
- Make pin no 9 of Arduino's =0
- Make pin no 10 of Arduino's =1

7. Case Close the door
- Make pin no 8 (enable pin) of Arduino's=1
- Make pin no 9 of Arduino's =0
- Make pin no 10 of Arduino's =0

5.3 Serial Message Structure

The structure of message consists of four parameters. These parameters are message start, message ID, delay and message end respectively. Message start will tell Arduino that it is the start of new message, or new command is ordered. Message ID will tell Arduino that which pin is this, either it is turn left, turn right, move forward, move backward, open door, close door, stop etc. Delay will tell that for how much time, it would give delay after turning this specific motor. Message end will tell Arduino that the message is finished. When the Arduino will get the message using com port, it will read the message or decode the message by tracking its Message ID. After decoding the message structure, Arduino will enable that pin number against some Message ID, and will give some delay with the provided delay value. On the basis of Message ID, Arduino will send signal to enable that pin. If the Message structure consist of AE, 1, 3000, AF, then AE is the start of new message, 1 is the Message ID, 3000 is the delay time of 3000 milliseconds, and AF is the end of that message. Message ID 1 tells Arduino to Turn Right car, and give delay of 3000 milliseconds after turn and it is the way that how voice controlled robotic car will work. The structure of message is elucidated in Figure 3. Message response from Arduino is explained in this block diagram given in Figure 4.

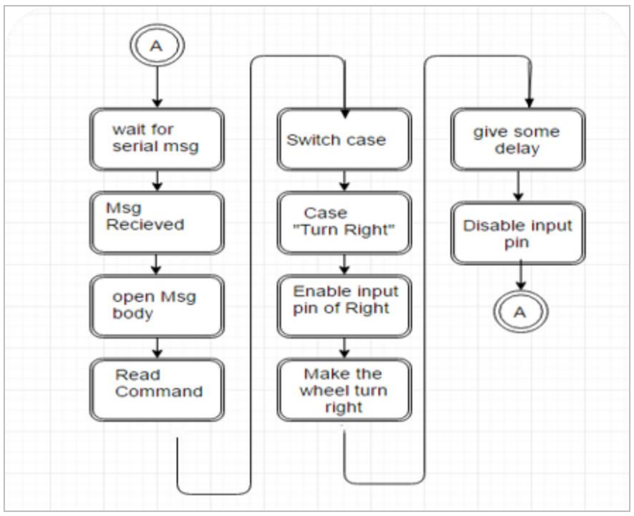


Fig. 4 Response algorithm of Structured Message

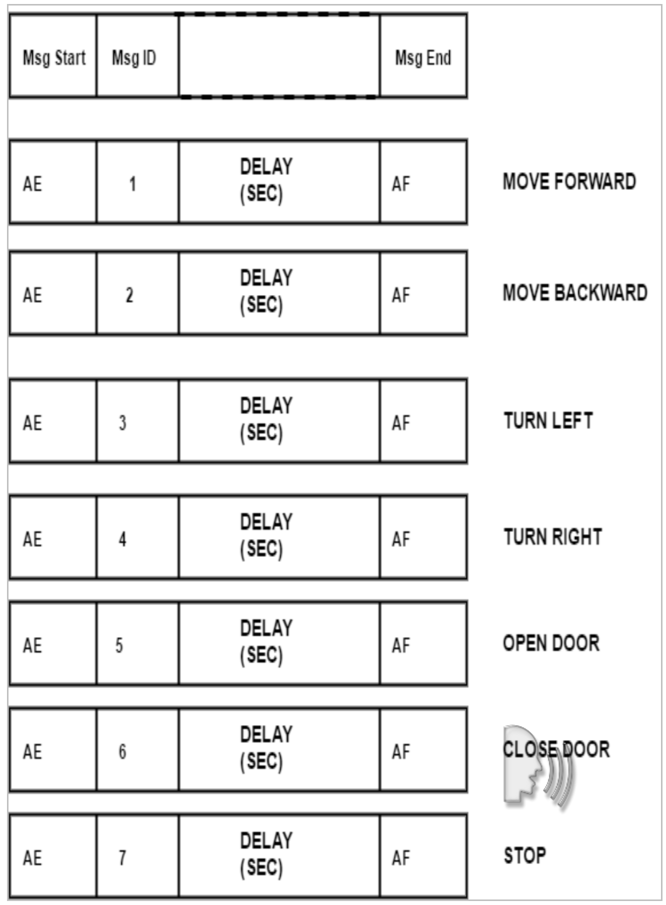


Fig 3: Serial message structure

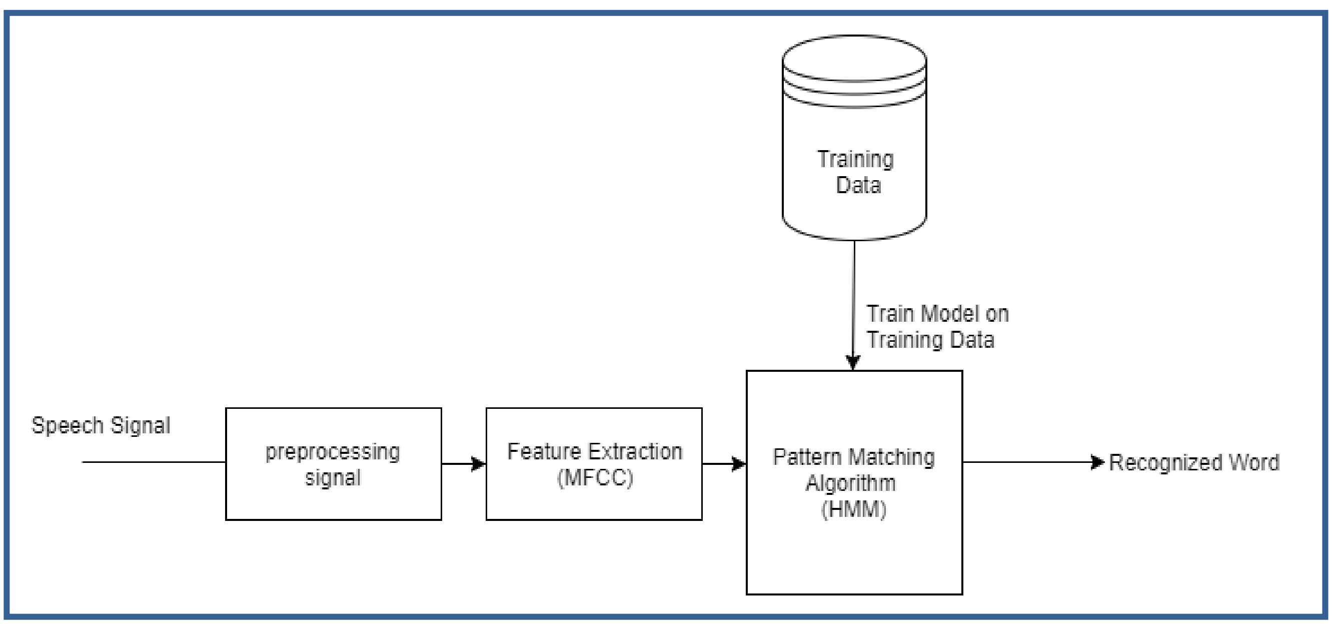


Fig 5: Architecture diagram of proposed system

6. Speech Recognition

The overall architecture of a speech recognition or speaker verification system is shown in Figure 5 which begins from speech uttering to speech recognition. When user utters speech signal, it will first go for preprocessing. Preprocessing is the mandatory step to do in speech recognition because speech signal involves noise and we can't send it to the machine learning model directly, we have to process it first. After preprocessing, this speech signal is sent to Feature extraction algorithm that is MFCC here [13]. It will extract the important features from voice and cut down other irrelevant information. Then these feature vectors will go for training in pattern matching algorithm that is Hidden Markov Model (HMM) [14] [15]. This model will predict the pattern of the feature algorithm and will result in recognized word at the end [16]. The block diagram is shown in Figure 6.

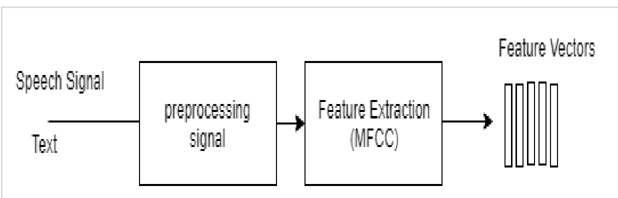


Fig 6. Preprocessing and Feature extraction

MEL FREQUENCY CEPSTRUM COEFFICIENT
MFCC converts a speech signal into a sequence of acoustic feature vector to identify the components of linguistic content and discarding all the other stuff which carries information like background noise, emotion etc. The purpose of this module is to convert the speech waveform to some type of parametric representation. MFCC is used to extract the unique features of speech samples. It represents the short-term power spectrum of human speech [17]. The MFCC technique makes use of two types of filters, namely, linearly spaced filters and logarithmically spaced filters. To capture the phonetically important characteristics of speech, signal is expressed in the Mel frequency scale. The Mel scale is mainly based on the study of observing the pitch or frequency perceived by the human. The scale is divided into the units mel. The Mel scale is normally a linear mapping below 1000 Hz and logarithmically spaced above 1000 Hz. MFCC consists of six computational steps. Each step has its own function and mathematical approaches as discussed briefly in the following:

STEP 1: PRE–EMPHASIS
This step processes the passing of signal through a filter which emphasizes higher frequency in the band of frequencies the magnitude of some higher frequencies with respect to magnitude of other lower frequencies in order to improve the overall SNR. It increases & with this process will increase the energy of signal at higher frequency.

STEP 2: FRAMING
The process of segmenting the sampled speech samples into a small frame. The speech signal is divided into frames of N samples. Adjacent frames are being separated by M (M<N). Typical values used are M = 100 and N= 256(which is equivalent to ~ 30 m sec windowing).

STEP 3: HAMMING WINDOWING
Each individual frame is windowed so as to minimize the signal discontinuities at the beginning and end of each frame. Hamming window is used as window and it integrates all the closest frequency lines. The Hamming window equation is given as:

If the window is defined as
W (n), $0 \leq n \leq N-1$ where
N = number of samples in each frame
Y[n] = Output signal
X (n) = input signal

W (n) = Hamming window, then the result of windowing signal is shown below:
Y (n) = X (n) * W (n)
W (n) =0.54–0.46 Cos (2∏ n / N-1); 0 < n < N-1

STEP 4: FAST FOURIER TRANSFORM
To convert each frame of N samples from time domain into frequency domain FFT is applied.

STEP 5: MEL FILTER BANK PROCESSING
The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. The bank of filters according to Mel scale as shown in Fig 6 is then performed.

STEP 6: DISCRETE COSINE TRANSFORM
This is the process to convert the log Mel spectrum into time domain using Discrete Cosine Transform (DCT). The result of the conversion is called Mel Frequency Cepstrum Coefficients. The set of coefficients is called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vector.

7. Machine Learning Model for Speech Recognition:

Hidden Markov Model (HMM) is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [18], [19]. HMM creates stochastic models from known utterances and compares the probability that the unknown utterance was generated by each model [17]. HMM can be characterized by following:

- Transition probability matrix
- Emission probability matrix
- Initial probability matrix

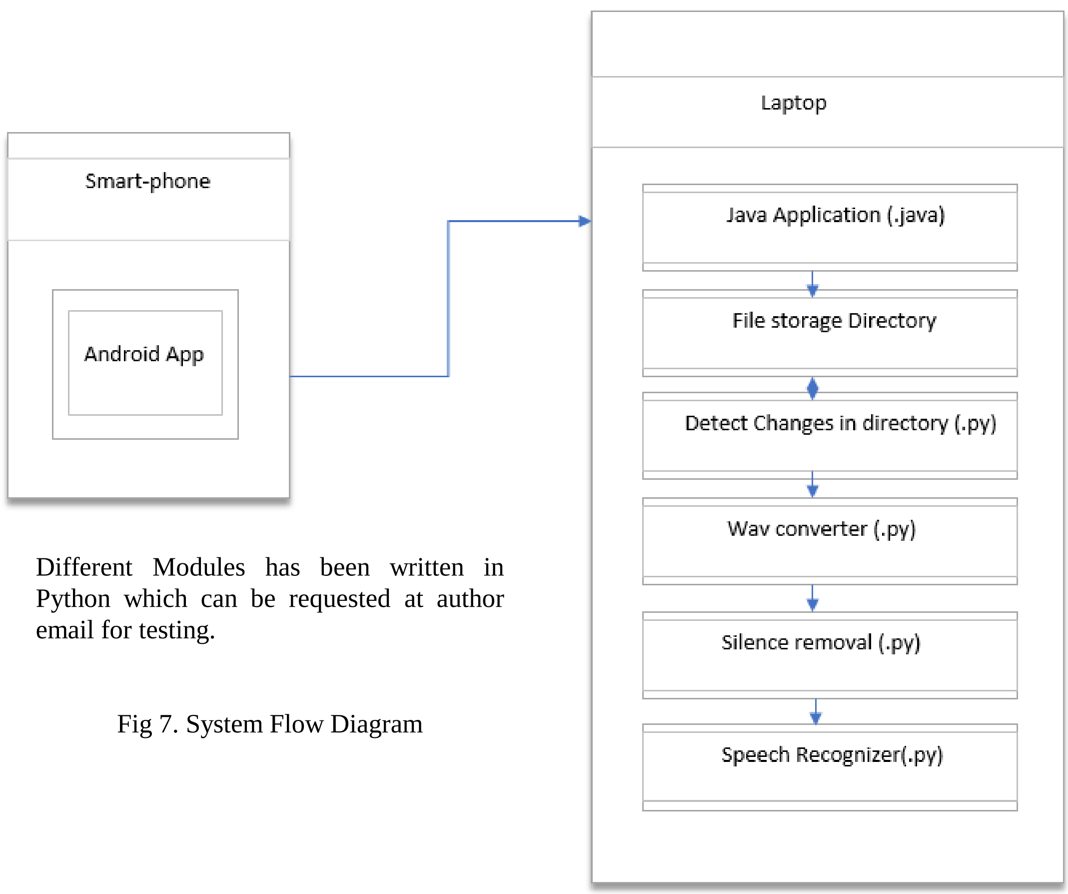
- $\lambda = (A, B, \pi)$, is shorthand notation for an HMM.

Other notation is used in Hidden Markov Models are:

- π = initial state distribution (πi)
- A = state transition probabilities (aij)
- B = observation probability matrix (bj(k))
- N = number of states in the model {1,2...N} or the state at time t $\rightarrow$st
- M = number of output observation symbols per state
- O = {O0, O1, . . ., OM−1} = Set of observation symbols
- T = length of the observation sequence
- Q = {q0, q1, . . ., qN − 1} = Set of Hidden states

6. Conclusion and future work:

In this paper we have presented a voice recognition model using MEL frequency cepstrum coefficient and Hidden Markov Model for controlling a robotic car. This model uses MFCC for voice feature extraction and then those features are sent to HMM for voice recognition. This model can be extended further in future to work in a noisy environment by introducing some noise reduction techniques. Moreover, various sensors can be added for object detection and distance measurements with other objects so as to improve the robotic car for autonomy [20]. Same voice recognition model can be applied to various other application areas like voice controlled wheel chairs for disabled and controlling various electronic devices and appliances at home using voice.



Different Modules has been written in Python which can be requested at author email for testing.

Fig 7. System Flow Diagram

Fig 7. Clearly Shows that first the voice is captured through smart-phone built in microphone in .3gpp extension and then it writes to the external storage(SD) of mobile phone and then it is transmitted to the laptop (using socket connection) on which our speech recognizer is running. When the .3gpp file is transmitted to the laptop from android phone, we cannot process .3gpp file from our recognizer directly. We have to first convert it in .wav file and then have to apply silence detection algorithm to remove silence and then pass it to the recognizer and this process continues. Note that we are calling Android Program as Server and Java program as client only because of Java program is sending the request to the android application and only has the permission to receive the file while Android application has the right to accept the connection and send file to the Java program. In our speech recognition program, training is done on training data once, but the program is running in infinite loop for the testing file. Whenever new file is received, it will get notified and process it and then recognize it. Now the question is how it will be notified when new file is received using java program in some directory. So, there is the Python program which is watching the directory in which Java program is receiving the new file and whenever any new file is added in directory, it will check the directory process the file and send it to the recognizer for testing. When we record wav file through laptop mic in wav format, there is much noise in the voice signal. And after capturing the signal when we applied silence removal on that signal, silence is not detected, it is recognizing silence as noise and silence removal program could not remove the silence. So, we write a simple application on android, which capture the voice from built in microphone of smart phone and without changing the signal, it writes the file in SD card. Android phone does automatically voice file saved in .3gpp extension and then we have sent this file to our laptop and then convert it to wav file using some library and then write it in new file with .wav extension. So, I decided to use smart-phone to record samples for training of my speech recognizer and for the testing. purpose. Whenever data is available on serial, first we will check that either it is a start of new message or not, by checking first character that if it is ‘S’ or not, if it is a start of message, it will reset the current index status to zero, which is showing that we have not parse message ID and Delay value yet for the new message structure. After checking for the start of message, it will parse MSG ID char by char and when it encounters “,” in this message, it will understand that MSD ID is parsed, current index status will be now “1” and then DELAY value will be parsed. While parsing DELAY, char by char, when it encounters “E”, that will show that it is the end of the message. Both MSG ID and DELAY value would be store in array name “Val”. It means at the index=0 of Val array, MSG ID will be stored, and at index=1 of Val array, DELAY value will be stored. When the message structure is parsed successfully, then HIGH the enable pins, because we know that we are using L293D IC and each IC has two enable pins, and to do run any motor of any IC, we have to HIGH the corresponding enable PINs. A lot of speech aware applications are already there in the market. Various dictations software has been developed by Dragon, IBM and Philips. Genie is an interactive speech recognition software developed by Microsoft. Various voice navigation applications, one developed by AT&T, allow users to control their computer by voice, like browsing the Internet by voice. Many more applications of this kind are appearing every day. Google Car developed by the big giant google is also a big progress in the field of speech in the field of speech recognition. But the thing is that, behind these apps there are giants like google, Microsoft, iphone, IBM. Such work at Universities lab will motivate graduates and postgrads to develop these kinds of application.

References

[1] L. R. Rabiner, B. H. Juang, “Fundamentals of Speech Recognition”, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] L. Diener, S. Bredehoeft and T. Schultz, "A comparison of EMG-to-Speech Conversion for Isolated and Continuous Speech," Speech Communication; 13th ITG-Symposium, Oldenburg, Germany, 2018, pp. 1-5.

[3] K. Darabkh, L. Haddad, S. Sweidan, M. Hawa, R. Saifan and S. Alnabelsi, "An efficient speech recognition system for arm-disabled students based on isolated words", Computer Applications in Engineering Education, vol. 26, no. 2, pp. 285-301, 2017. Available: 10.1002/cae.21884.

[4] Zakariyya Hassan Abdullahi, Nuhu Alhaji Muhammad, Jazuli Sanusi Kazaure, and Amuda F.A., “Mobile Robot Voice Recognition in Control Movements”, International Journal of Computer Science and Electronics Engineering (IJCSEE), vol. 3, Issue 1, pp. 11-16, 2015.

[5] Sneha V., Hardhika G., Jeeva Priya K., Gupta D. (2018) Isolated Kannada Speech Recognition Using HTK—A Detailed Approach. In: Saeed K., Chaki N., Pati B., Bakshi S., Mohapatra D. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 564. Springer, Singapore

[6] K. R, L. Birla and S. K, "A robust unsupervised pattern discovery and clustering of speech signals", Pattern Recognition Letters, vol. 116, pp. 254-261, 2018. Available: 10.1016/j.patrec.2018.10.035.

[7] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286.

[8] A. Haridas, R. Marimuthu and V. Sivakumar, "A critical review and analysis on techniques of speech recognition: The road ahead", International Journal of Knowledge-based and Intelligent Engineering Systems, vol. 22, no. 1, pp. 39-57, 2018. Available: 10.3233/kes-180374.

[9] N. Sharmili, L. Tasnim, I. Shamsuddoha and T. Ahmed, Visual speech recognition using artificial neural networking", Hdl.handle.net, 2019. [Online]. Available: http://hdl.handle.net/10361/10183. [Accessed: 28- Mar- 2019].

[10] T. Seong, M. Ibrahim, N. Arshad and D. Mulvaney, "A Comparison of Model Validation Techniques for Audio-Visual Speech Recognition", IT Convergence and Security 2017, pp. 112-119, 2017. Available: 10.1007/978-981-10-6451-7_14 [Accessed 28 March 2019].

[11] L. R. Rabiner, “ A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, vol. 77, no. 2, pp. 257-286, 1989.

[12] Tatiana Alexenko, Megan Biondo, Deya Banisakher, Marjorie Skubic, “Android-based Speech Processing for Eldercare Robotics”, IUI '13 Companion Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion, pp.87-88, March2013.

[13] A. Takrim, M. Kamruzzaman, A. Zoha and S. Ahmmad, "Speech to Text Recognition", Dspace.uiu.ac.bd, 2019. [Online]. Available: http://dspace.uiu.ac.bd/handle/52243/383. [Accessed: 28- Mar- 2019].

[14] N. Le, S. Chen, T. Tai and J. Wang, "Single-Channel Speech Separation Based on Gaussian Process Regression", 2018 IEEE International Symposium on Multimedia (ISM), pp. 275-278, 2018. Available: 10.1109/ism.2018.00040 [Accessed 28 March 2019].

[15] H. Seki, T. Hori, S. Watanabe, J. Roux and J. Hershey, "A Purely End-to-end System for Multi-speaker Speech Recognition", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1805.05826. [Accessed: 28- Mar- 2019].

[16] A. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review", IEEE Access, vol. 7, pp. 19143-19165, 2019. Available: 10.1109/access.2019.2896880 [Accessed 28 March 2019].

[17] K. Naithani, V. M. Thakkar and A. Semwal, "English Language Speech Recognition Using MFCC and HMM," 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), San Salvador, 2018, pp. 1-7. Available: 10.1109/RICE.2018.8509046

[18] J.D. Ferguson, “Hidden Markov Analysis: An Introduction “, in Hidden Markov Models for Speech, Institute of Defence Analyses, Princeton, NJ, 1980.

[19] L. R. Rabiner, B. H. Juang, “An Introduction to Hidden Markov Models”, IEEE ASSP Magazine, pp. 4 – 16, Jan. 1986.

[20] T. Afouras, J. Chung, A. Senior, O. Vinyals and A. Zisserman, "Deep Audio-visual Speech Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1-1, 2018. Available: 10.1109/tpami.2018.2889052 [Accessed 28 March 2019].

Ayesha Shafique received BS. Degree in Computer Science from University of Karachi, in 2017. Her research interests include data mining, machine learning, and deep neural networks mainly in healthcare sector. Currently, she is working as a Data Science Lead at Ephlux.

Humera Tariq received the B.E (Electrical) degree from NED University of Engineering and Technology in 1999 and then continue her studies at University of Karachi for Masters of Computer Science (MCS) in 2001. She stands First Class First amongst the Evening batch of MCS 2001-2003. She joined MS leading to PhD program in 2009 and completed MS course work with CGPA 4.0. She started her PhD work in the field of image processing in 2011 under the supervision of Meritorious Professor Dr. S.M. Aqil Burney.

Farid Alvi is a senior SAP Consultant. Currently he is working in both industry and academia. He is a visiting faculty at University of Karachi, Department of Computer Science.

Usman Amjad received BS. Degree in Computer Science from University of Karachi, in 2008. He recently completed his PhD in Computer Science from University of Karachi. His research interests include soft computing, machine learning, artificial intelligence and programming languages. He was the recipient of the HEC Indigenous 5000 scholarship in 2013. Currently, he is working as AI solution architect at Datics.ai Solutions.