

Application of MFCC and Edge Detection for Remote Driven Vehicles through Matlab

T Ravi¹, K Pradeep Reddy², K Eepsita Saranya²

K L University, Guntur, India¹. B.Tech 4thyr K L University, Guntur, India
Email: ravi.blind@gmail.com, q4pradeep@gmail.com, +91 966 649 7297

Abstract

Speech recognition is a rapidly emerging technology in Human Computer Interaction HCI. It has many applications as we use it from search engines to the device control it serves many areas we interact every day from dawn to dusk. Along with the uses we have many limitations in speech processing such as Language barrier, Accent and Noise, so to implement the speech processing we have many challenges. To enable the advantages of this speech processing most of the leading software companies like Apple, Microsoft and Google are continuously evolving their speech enabled applications. The speech processing eases the physically challenged people's interaction with the devices and makes them productive. The idea of the automatically driven cars is introduced by Google and Audi, but they are not acceptable in most of the cases because of lacking trust in current technology. Thus we here worked on the remote driven vehicle in a more secured method using the Mel Frequency cepstral Coefficient. The secure driving of the vehicle can be ensured by the remote driver. This technique is very rapid and more reliable for the speech detection thus the remote driver can use the MFCC and the video from the vehicle needed to be broadcasted to the remote driver that can be done an IP camera running on a data network. And the instructions from the remote driver can be sent to the vehicle by an app created with python that connected to a micro controller. To minimize the limitations in the remote drive the vehicle must be enabled with automatic braking when obstacle approaches which can be done by the Ultrasonic sensors that do the distance estimation. The remote drivers usually have a very limited view of road they drive and they must get the precise edges of the road this can be achieved by processing the stream of images to calculate the edge in Matlab.

Keywords: MFCC, Remote Drive, IP Camera, Edge Detection, Matlab, DTMF, Edge Detection, Accident Avoidance

1. Introduction

For the last four decades there's been a great acceleration in improving the automatic speech recognition systems and now the ASR is applied in the fields like aviation for the unmanned drones, banking for voice based locking systems, search engines to interact quickly with the netizens. With the advancements that took place due to the rapid simulating software's like matlab, and wolfram's Mathematica, in speech recognition and processing we can develop a large number of systems that can use ASR, which users can control with speech and the speed of interaction thus enhances

But however the ASR based vehicle controlling systems not evolved due to many limitations in the current speech recognition algorithms. The main problem with the ASR systems is noise. Most of the systems we use in our daily life are out door, the outdoor environment is very badly influenced by the noise. Sometimes noise to message ratio will be very high where the message will be dropped. If we implement ASR based vehicle control in such scenario there is large scope for the system

Malfunctioning and another challenge to implement the ASR is accent and pitch of people. Asper the researches the fundamental frequency of an adult male lies in the range of 85Hz to 180Hz and in case of an adult female it lies in the range of 165Hz to 255 Hz. [1] [2]. Accent is the biggest critical issue in the design of Automatic Speech Recognition Systems this is because of the dialects in the natural spoken languages. For examples the Chinese language has many dialects Wu, Yue, Min etc., and however the official language is Putonghua or Mandarin [3]. However in spite of all these limitations the MFCC algorithm works quite well with the closer accents, thus we implemented the ASR Driven vehicle with this algorithm and coded the real-time application in matlab.

The design of ASR system involves the following stages [4], as Figure 1.

- Audio Recording
- Pre Emphasis
- Framing & Windowing
- FFT and DCT
- Extraction of Features MFCC

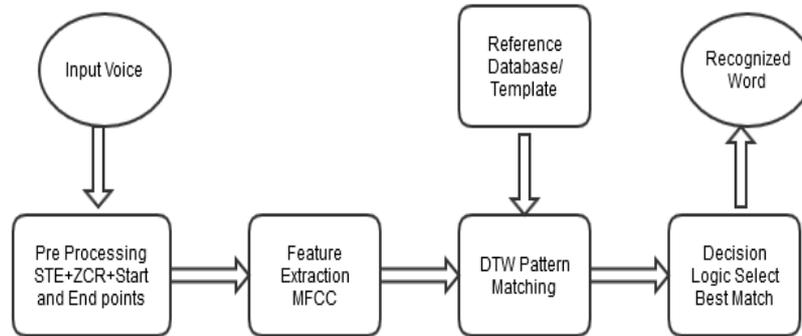


Figure 1. Block diagram of Speech Recognition System

In every basic Speech recognition system we use either the LPC or MFCC when implementing the system with the MFCC, we go ahead with 4 stages, Preprocessing of the recorded command, Feature extraction using the MFCC algorithm, Comparing the DTW pattern with the Template or Pattern database and finally a logic will select the closest match in the Database [5], as Figure 2.

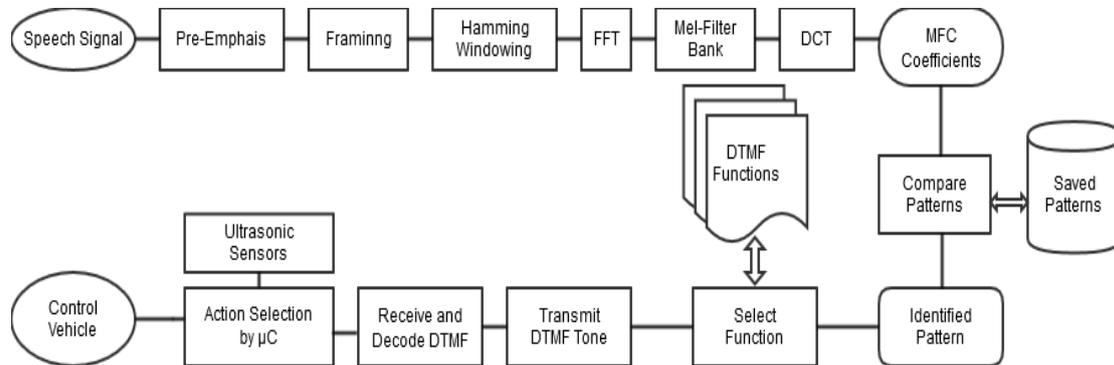


Figure 2. Showing the sequence of actions

2. Design Methodology

The flow chart in Figure 1 in the introduction shows the algorithm implemented for the 'recognition-phase' in the system that we are implementing. The command spoken by a user (an acoustic wave) is first subjected to the pre-processing phase. This pre-processing stage involves the removal of the DC offset in the acoustic wave, low pass filtering to remove the high frequency and undesirable noise from the wave, and pre-emphasis filtering to enhance the energy of the wave. This utterance is filtered using a fourth order Butterworth low pass filter and a pre-emphasis filter. A Hamming window of size equivalent to a window size of 20msec and an overlapping frame size of 10msec are implemented in this project. A Hamming window also preserves the spectral characteristic of the signal. The starting and ending point of this pre-processed utterance is determined from the Zero crossing rate and energy measures. The

original utterance in the time domain is read between the pre-determined 'Start' and 'Finish' point. The Mel-frequency cepstral coefficients (MFCC) are calculated for each window of the utterance. MFCC calculation is achieved by passing the Fourier transform of the utterance in the time domain through a bank of 24 triangular digital filters, designed around 'Mel' scale to extract 12 MFCC parameters and 1 energy parameter Cepstral parameters make up an orthogonal set, which makes it meaningful to compute the Euclidean distance between feature vectors. The Euclidean distance between the MFCC's and each of the library words is calculated to form a distance matrix, which is used by the 'Dynamic Time Warping' algorithm to adjust to the length of the orthogonal set of MFCC's for the utterance of the reference speaker. The optimal distance through this distance matrix, which is governed by a number of warping constraints, determines the overall similarity between the utterance and the library word. The library word that returns the minimum distance when compared against the utterance is assumed to be possibly the right word. This word is only returned as the right word if no other distance calculated is within 5 percent of this minimum distance.

3. Command to DTMF Translation for Acoustic Transmission

The MFCC algorithm's output will be the command word that is spoken by the end user now we need a methodology to transmit this command to vehicle. We cannot use any cable connection as we are implementing on remote driven vehicle so we identified two ways to send the command one using the webserver and the other using the DTMF. And we implemented it using the DTMF due to ease in acoustic communication over long distance over telephone or sound broadcast server.

A dual tone multi-frequency tone is the sum of two sinusoidal waves or tones with two frequencies taken from two mutually exclusive groups. These frequencies are selected in such a way that every tune will be detected uniquely so that the harmonics will never be wrongly identified. In general the DTMF frequencies contain two groups of frequencies one is low group and the other is high group low group contains (697 Hz, 770 Hz, 852 Hz, 941 Hz) and high group contains (1209 Hz, 1336 Hz, 1477Hz). On total we can generate up to 12 tones [6]

Table 1. DTMF frequencies

Frequencies	1209 Hz	1336 Hz	1477Hz
697 Hz	1 Rotate Left	2 Forward	3 Rotate Right
770 Hz	4 Left	5 Stop	6 Right
852 Hz	7 Horn ON	8 Reverse	9 Lamp ON
941 Hz	* Horn OFF	0 Lock Car	# Lamp OFF

Tone= sum (sin (lfg(c); hfg(r))* (2*pi*t));

lfg = [697 770 852 941]; (Low frequency group)

hfg = [1209 1336 1477]; (High frequency group)

The generated tone can be played on matlab using the audioplayer command once the tone is generated the task is to send the tune to the vehicle this can be done in two ways one is to use the mobile call function to send the tune with a mobile on the vehicle with automatic call answering this method is not suitable because of the call charges cannot be implemented practically with ease. So the other method is to use a sound broadcast server like 'Sound wire' Server which is an open source software by George Labs this will broadcast the sound that is playing on the pc over a VOIP server the client listening in this server can retrieve the tune. The client is also designed by the George Labs and can be used for free with a limited usage. The server client need not be in the same network if the port is forwarded by the server, the only limitation is the client (android mobile) must enabled with a data network.

4. Vehicle Control using the Tune Received

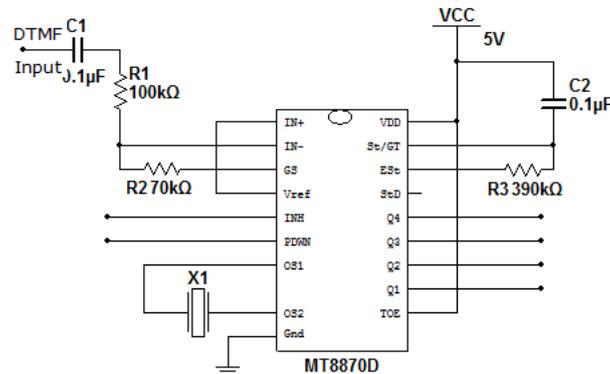


Figure 3. DTMF tune based on MT8870D

Once the DTMF tune is received by the vehicle the tune needs to be decoded and the decoded pattern should be sent for further control. The familiar DTMF decoding IC is MT 8870D runs on 5V.

The DTMF tune is fed as input the circuit shown and outputs Q_1 , Q_2 , Q_3 and Q_4 the output will be the binary equivalent of the number on typical keypad

Where * is mapped as 10, 0 is mapped as 11 and # is mapped as 12. This binary pattern is given to the microcontroller and the micro controller is programmed in such a way that it executes the actions according the commands by the user.

5. EDGE Detection and Ultrasonic Sensors

Usually the remote driver have a very limited control and limited so that the edges of the road need to be determined so particularly to implement this we need an IP cam on the vehicle the cam feed must be retrieved by the matlab. But the matlab has a limitation towards listening to the IP cameras so we need to take the jpeg stream instead of the video stream periodically frames need to capture and the edges have to be calculated. And if the remote vehicle fails to receive the commands for odd reasons the vehicle needs to be controlled so as to avoid accidents thus we require the ultrasonic sensors to estimate the approaching vehicles. Edge can be defined as the path around an object in an image or more precisely can be defined as path that differentiates the rapid changing density pixels in an image [8].

6. Results and Discussion

Matlab Program with GUI to Generate the DTMF sequences for the control the vehicle remotely using buttons, as Figure 4. Performances as shown in Figures 5, 6 and 7.

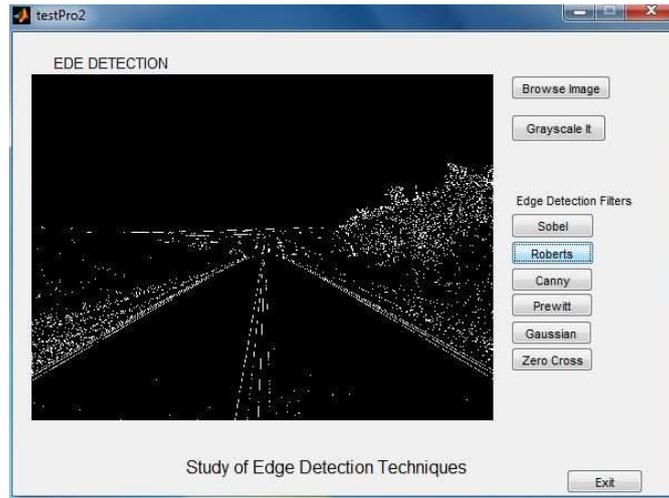


Figure 4. Matlab Program with GUI to Generate the DTMF sequences

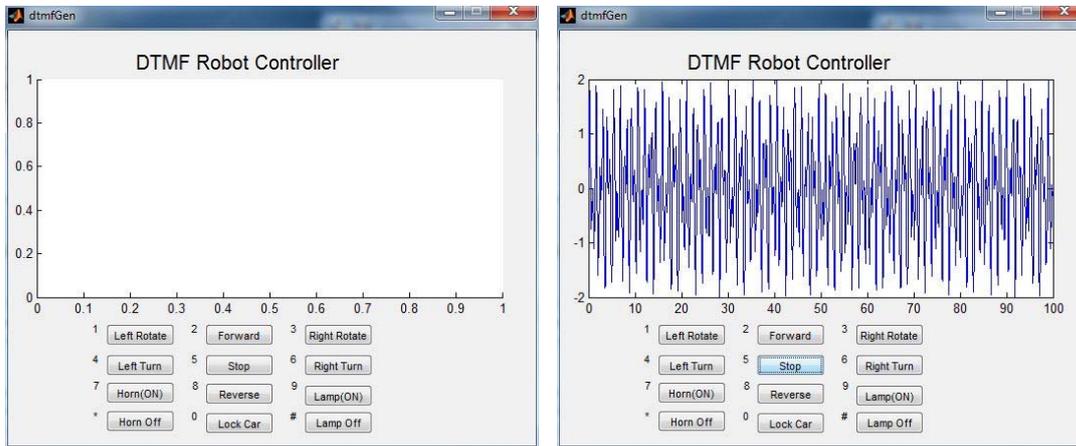


Figure 5. DTMF Robot Controller

Matlab Program with GUI to Generate the DTMF sequences for the control the vehicle remotely using Speech recognition.

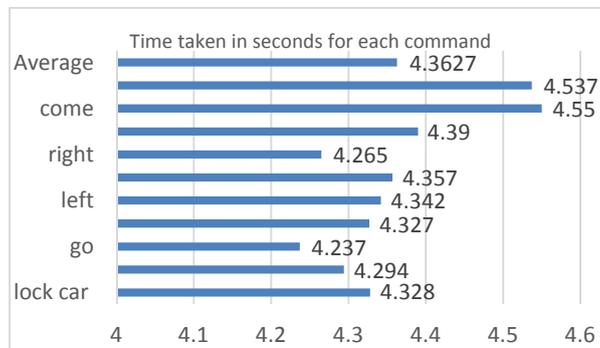


Figure 6. Time taken for each command

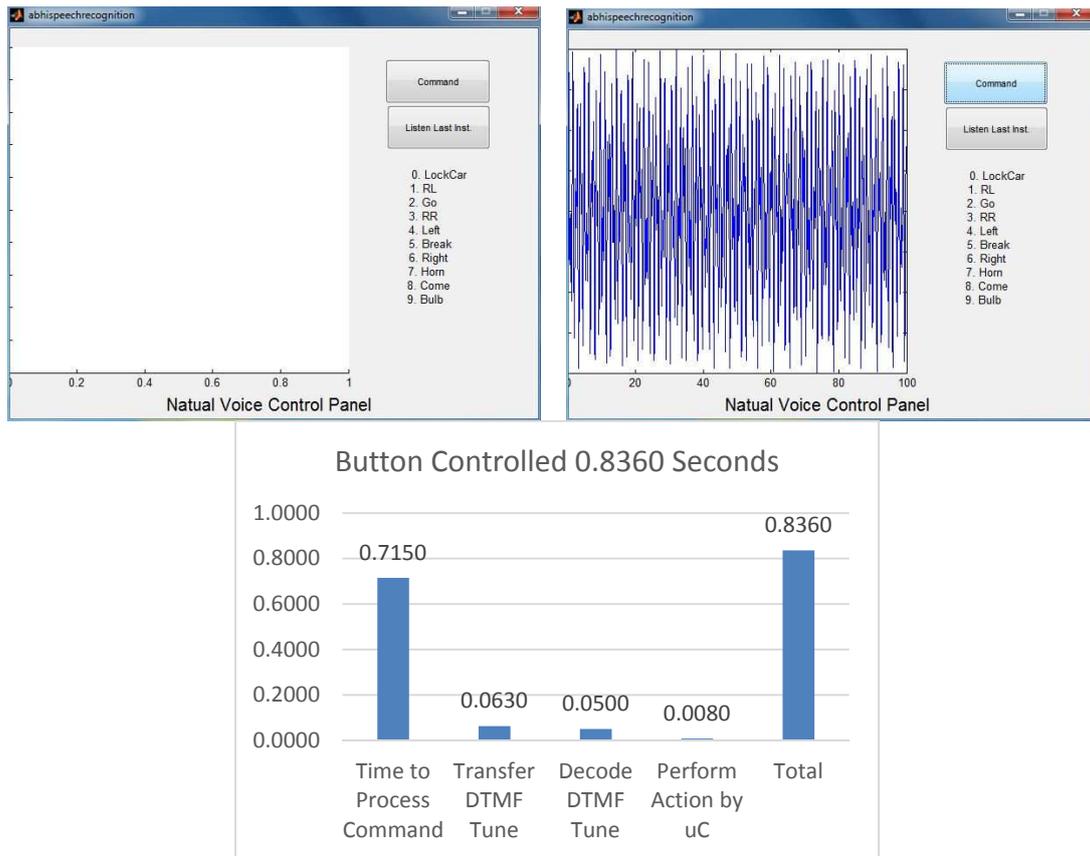


Figure 7. Natural voice control panel

The MFCC based ASR took about 4.3627 Seconds to which might be undesirable to use in some realtime scenario so we implemented another methodology of implementation using the press buttons for which the average time required is just 0.715 Seconds, as Figure 8.

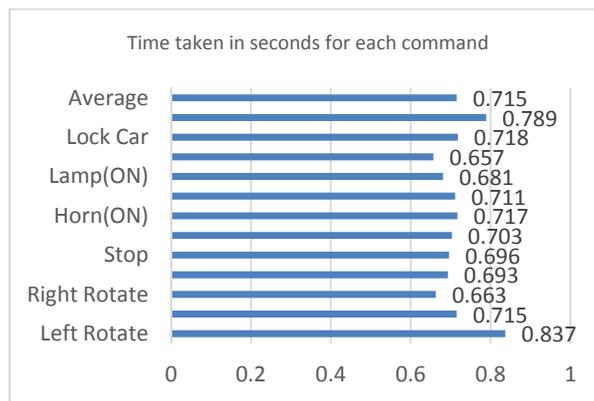


Figure 8. Time take for each command

We have studied all the time complexities for every command using the matlab software package and the timings are as below, as Figure 9:

- Average time required to receive the DTMF tone by the vehicle 0.063 seconds
- Average time required to decode the DTMF tone by the vehicle 0.050 seconds
- Average time required to perform action by uC 0.008 seconds

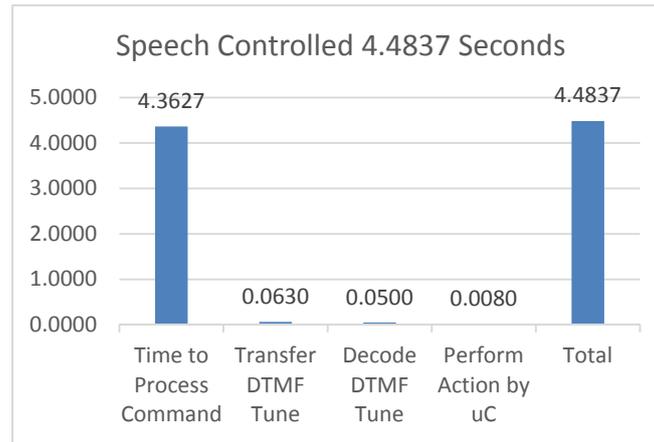


Figure 9. Response of the speech controlled

7. Conclusion

We implemented a secure system using MFCC Automatic Speech Recognition and system will survive against the communication failures through the Ultra Sonic sensors. However system that we implemented requires a faster CPU to minimize the time required for the MFCC. All the timings in the paper are tested under Intel i3 @ 2.1 GHz with single core (2 cpus). The time further can be decreased using more cores and high performance CPU but which on other hand is expensive. The algorithm can be further optimized to decrease the time utilization.

References

- [1] Titze IR. Principles of Voice Production. Prentice Hall (currently published by NCVS.org) ISBN 978-0-13-717893-3. 1994: 188.
- [2] Baken RJ. Clinical Measurement of Speech and Voice. London: Taylor and Francis Ltd. ISBN 1-5659-3869-0. 1987: 177.
- [3] Accent Detection and Speech Recognition for Shanghai Accented Mandarin ;Yanli Zheng¹, Richard Sproat¹, Liang Gu², Izhak Shafran³, Haolang Zhou³, Yi Su³, Dan Jurafsky⁴, Rebecca⁴ Starr, Su-Youn Yoon¹ University of Illinois¹, IBM T.J.Watson Research Center² Johns Hopkins University⁴, Stanford University⁴.
- [4] Speaker Adaptable Voice Controlled Model-Vehicle using Energy Threshold and MFCC Parameters. K.S. Anantha krishnan*, Ryan Andrew,*School of Electrical & Information Engineering, University of South Australia.
- [5] Recognizing voice commands for robot using MFCC and DTW; Nidhi Desai, Prof.Kinnal Dhameliya, Prof. Vijayendra Desai, CGPIT Bardoli, India; ISSN (Print): 2319-5940. *International Journal of Advanced Research in Computer and Communication Engineering*. Vol. 3, Issue 5
- [6] <http://in.mathworks.com/products/demos/signaltlbx/dtmf/dtmfdemo.html> taken from the internet Copyright 1988-2004 The MathWorks, Inc. Published with MATLAB® 7.0.4
- [7] <http://in.mathworks.com/help/images/edge-detection.html>