Voice Recognition System Using Agilent DAQ and DSP

Wen Shengchao

***Abstract*—The project is an industrial collaborated project between NUS and Agilent Technologies Company, a simple discrete-word, speaker dependent voice recognition system installed in vehicle environment is developed and tested. The voice recognition system is designed based on the stochastic processing technique which uses Hidden Markov Model (HMM) method to construct model for each word/phrase. The whole system consists of 3 subsystems namely DAQ system, Voice recognition system and computer system interface. Agilent U2353A DAQ tool is used to get and manipulate the speech signal into the computer, then MFCC parameters in each frame is extracted from the speech data as features and a speech database is built based on MFCC, the recognition result is then transferred to the computer system interface and the interface will take carry out the respective action according to the result.**

I. INTRODUCTION

VEHICLES are highly desirable environments for speech recognition applications because the driver's eyes should remain on the road as much as possible and, ideally, both hands should remain on the steering wheel. It is necessary that voice recognition system can be designed for vehicles use to help the driver control the facilities in inside the vehicles so that they can focus on driving and hence to reduce the accident rate [1].

The objective of this project is to build a voice recognition system for such purpose. Due to the technology difficulties and short time constrain, a simple discrete word, speaker dependent recognition system simulation program which runs on PCs instead of real vehicles is developed.

II. VOICE RECOGNIITON TECHNOLGY REVIEW

A. Voice recognition techniques

Currently 3 types of recognition techniques are applied in modern recognition systems, namely: Template Matching, Acoustic-phonetic recognition and Stochastic processing. These approaches differs from each other in terms of speed, recognition accuracy and storage requirements [2].

A.1 Template Matching

Template Matching is a form of pattern matching. The speech data is represented as sets of feature vectors called templates. And each word or phase is stored as separate templates in the speech database. During the recognition stage, the speaker input speech is first convert into template and then compared with the all templates stored in the database. The best match template is selected as the recognition result. Template matching is based at the word level and contains no reference to the phoneme within the word, which make it only suitable for developing discrete-word recognition system [2].

A.2 Acoustic-phonetic recognition

Acoustic-phonetic recognition functions at the phoneme level, it is very good way for continuous speech recognition for large vocabulary because there are limited representations of phoneme for a language. For English language, there are only around 40 phonemes no matter how large is the vocabulary. However, the steps involved in the recognition process are more complicated than the template matching method. It contains steps such as feature-extraction, segmentation and labeling and word-level recognition.

A.3 Acoustic-phonetic recognition

Stochastic is a probabilistic algorithm which is about the process of making a sequence of nondeterministic selections from sets of alternatives. Similar to the template matching method, it also requires creating and storing each word model in the speech database, the difference is that stochastic processing does not involve direct matching between stored models and the input speech. On the contrary, it is based on complex statistical and probabilistic calculation and computation. And the statistic word models are stored as Hidden Markov Model (HMM) [3], which the key technology in speech recognition process.

B. Current state-of-the-art

Commercially, there are a lot of professional voice recognition systems that are available for PCs users. Voice recognition technology is applied in many areas such as voice control system, entertainment, robotic and handheld devices.

Voice Recognition Applications

- **Automotive**
  Navigation systems; Telematics units; Hands-free car-kits
- **Handhelds**
  Handsets ; SmartPhones ; PDA
- **Game consoles**
- **Other Devices**
  Military; Industrial (e.g. warehousing); Language Learning
- **Software**
  Windows Vista; Dragon NaturallySpeaking

Academically, there are several sophisticated open source toolkits that people can make use of and modify to build HMM for voice/speech recognition system for academic purpose. The most popular and famous toolkit is HTK, which was originally built by Cambridge University Engineering Department (CUED) in 1993. It consists of a batch of library

codes based on C programming language that people can modify and make use of. Another popular toolkit is CSLU Toolkit, which is a comprehensive suite of tools to enable exploration, learning, and research into speech and human-computer interaction. The toolkit consists of set of library files and graphic programming tools, which make it very easy to use to build new voice recognition interface based on HMM or Neural Network. In this project, HTK toolkits are used due to its great flexibility of making use of the source code.

III. VOICE RECOGNITION SYSTEM IMPLEMENTATION

A. *System Overview*

As described above, the whole project consists of 3 subsystems. With the DAQ system process the speech input signal and then pass to the recognition system to recognize, according to the recognition result, the system interface will carry out actions such as play music, open door and so on.

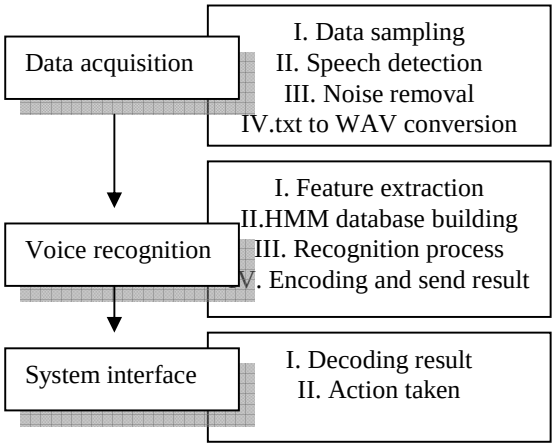


Fig.1 System organization

B. *Data Acquisition System (DAQ)*

The Agilent Technologies U2353A DAQ tool provides a sampling rate of 16 KHz with resolution of 16-bit [4]. The data it samples from the microphone is voltage level which represents the input speech signal, however, the input data format required by the recognizer designed from HTK toolkit is .WAV format. Hence, it is necessary to convert the voltage level from the text file to the WAV file format using the DAQ tool and the VEE program.

A.1 Speech Detection

Once the DAQ tool started, the data sampled keeps feeding into the system and will make the system busy with handling the data stream. Hence, speech detection has to be done to only get the speech data into the system. Each time the DAQ device samples 1 second of data into the computer by default. Normally the duration for a word will not exceed 1 second even though we speak it very slowly. The DAQ tools can begin to do the data writing action once it senses the maximum data value exceed certain threshold value. As shown in the following figure, the maximum noise level (H) in the quite environment is around 2.37V, so a bigger value such as 3.5V can be used as a threshold value to determine whether the input signal is quite region or voice region.

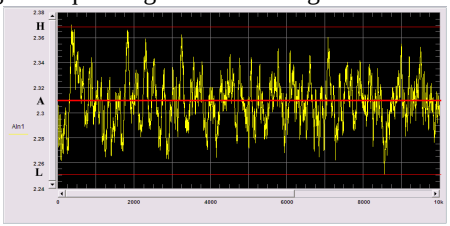


Fig.2 Waveform in quiet environment

Data reading and writing process will begin once the signal voltage from the DAQ device exceeds the threshold value. However, there maybe quite often the case that speech begins just at the end of the first 1 second duration, to guarantee the entire word/phrase signal is read into the file, an addition 1 second duration of sampling is necessary. Hence, for each word/phrase, the duration for them is 2 seconds, which contains 32000 sample data provided the sampling frequency is 16 KHz.

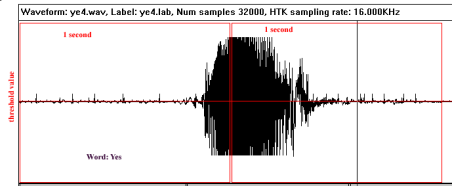


Fig.3 Voice starts at the end of the first 1second

A.2 Pre-noise removal action

As shown in Fig.2, the DAQ device produces certain level of DC voltage level such as 2.3V plus random noise signal with arbitrary frequency from the environment even when there is no speech input. To make the silence region produces not significant voltage level, the noise has to be removed before the speech is written into as .WAV file.

Firstly, sample the environment signal for 2 seconds and then derive the average environment noise voltage level (A), which can be used as a threshold value to remove the noise signal. The highest voltage level (H) and lowest voltage level (L) from the quite environment should also be recorded. With these 3 parameters, actions can be done as follows and the data is written into a text file.

if S >L and S <H → $S = 0$
else → $S = (S - A)*10000$

A.3 TXT to WAV Conversion

WAVE file format has a collection of several different types of chunks [5]. A Format ("fmt ") chunk which contains important parameters describing the waveform, such as its sample rate is required. The Data chunk, which contains the actual waveform data from the .txt file, is also stored here.

Fig.4 is a graphical overview of a simple WAV file structure. It consists of a single WAVE containing the 2 required chunks, a Format and a Data Chunk as illustrated below. In the converting process, the data array is read from the .txt file and then directly written into the data chunk of the .WAV file to form the speech .WAV file.

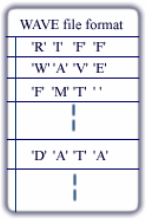


Fig.4 WAV files structure

C. *Voice Recognition System*

Mel Frequency Cepstrum Coefficients (MFCC) [6] is coefficient that represents the spectrum of spectrum of the speech signal. It is used as feature for each word/phrase to build a Hidden Markov Model (HMM) for that particular word/phrase.

B.1 Feature Extraction

MFCC has highly uncorrelated property and is used as features. The speech data is read from the .WAV file undergoes the pre-emphasis process to increase its energy level in high frequency component to a significant level to detect and analysis.

$$S'(n) = S(n) - 0.98 \times S(n-1)$$

The signal after the pre-emphasis step is then divided into frames and each frame has 512samples/frame. Hamming windowing function is then applied here due to its resolution in the frequency domain is relatively high and its spectral leak is small [7].

$$S_n' = \left\{0.54 - 0.46\cos\left(\frac{2\pi(n-1)}{N-1}\right)\right\} S_n$$

Fast Fourier Transform (FFT) is then done on the signal to derive the spectral of the speech signal. Fast Fourier Transform uses the Dnaielson-Lanczos algorithm [7] [8] to quickly generate the FFT result with computation order of $2N \lg N$, which is much faster than traditional DFT.

$$S(k) = \sum_{n=0}^{N-1} a_n e^{-2\pi ink/N} = \sum_{n=0}^{N/2-1} a_n e^{-2\pi i(2n)k/N} + \sum_{n=0}^{N/2-1} a_n e^{-2\pi i(2n+1)k/N}$$

$$= \sum_{n=0}^{N/2-1} a_n^{even} e^{-2\pi ink/(N/2)} + e^{-2\pi ik/N} \sum_{n=0}^{N/2-1} a_n^{odd} e^{-2\pi ink/(N/2)}$$

Human ear resolves frequencies non-linearly across the audio spectrum and experimental evidence suggests that designing a front-end to operate in a similar non-linear manner improves recognition performance [9]. The magnitude spectrum derived in the FFT step $|S(k)|$ is now scaled in frequency and magnitude using Mel filter bank [9] $H(k,m)$:

$S'(m) = \ln\left(\sum_{k=0}^{N-1} |S(k)| . H(k,m)\right)$ for m=1,2,..M, M is the number of filter banks and M<N. The Mel filter bank is a collection of triangular filters defined by the center frequencies $f_c(m)$ as shown.

$$H(k,m) = \begin{cases} 0 - - - (f(k) < f_c(m-1)) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} - - - (f_c(m-1) \le f(k) < f_c(m)) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)} - - - (f_c(m) \le f(k) < f_c(m+1)) \\ 0 - - - (f(k) > f_c(m+1)) \end{cases}$$

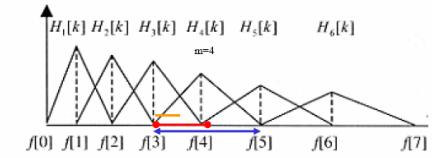


Fig.5 Filterbank

Cesprum is the spectrum of the spectrum of the signal, which is computed by treating the spectrum of the signal as a new signal and then applies the Fourier transform. Finally MFCC is obtained by computing DCT of $S'(j)$

$$C_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} S'(j) \cos\left(\frac{\pi i}{N}(j - 0.5)\right)$$

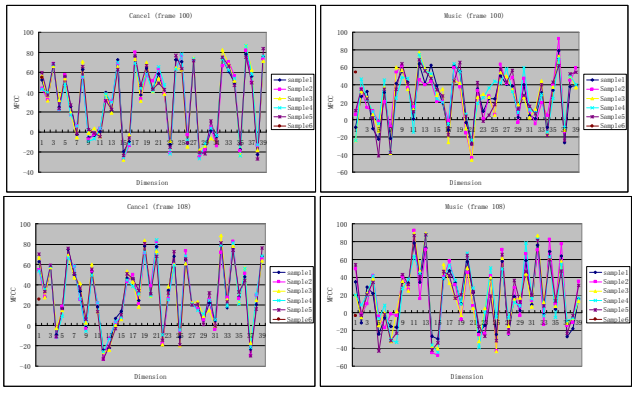
MFCC focuses only on each static state, which is not enough to represent the dynamical flow of the speech. It is important to estimate the change of coefficients from frame to frame so that the transitional probabilities calculated in the HMM building process can reflect the dynamically flow of the speech signal well [10]. The change of coefficients is denoted as delta cepstrum and delta-delta cepstrum.

$$\Delta C_k = C_{k+2} - C_{k-2} \text{ ---- (delta cepstrum)}$$

$$\Delta\Delta C_k = \Delta C_{k+1} - \Delta C_{k-1} \text{ ---- (delta-delta cepstrum)}$$

A 39-dimensional vector containing 13 MFCC, 13 delta cepstrum and 13 delta-delta cepstrum is used to express the voice signal in each frame. Individual MFCC are quite different between different words and remains similar for different samples of the same word at consecutive frames. Hence, it is possible to derive a suitable and relatively accurate MFCC set of representation for each word/phrase signal by estimating MFCC from a certain number of speech signal samples.

$$(C_1, C_2, ...C_{13}; \Delta C_1, \Delta C_2, ...\Delta C_{13}; \Delta\Delta C_1, \Delta\Delta C_2, ...\Delta\Delta C_{13})$$



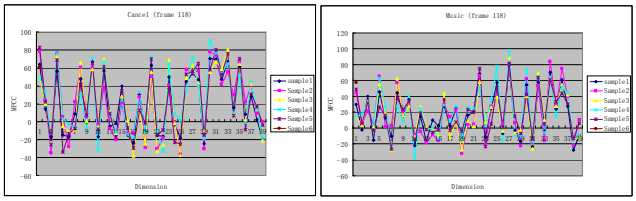


Fig.6 MFCC comparision

B.2 HMM database building

Hidden Markov Model (HMM) generally assumes that the sequence of the observed speech MFCC vectors are generated by the underlying states/symbols changes from state to state once every time unit. So, the observed speech MFCC vectors can be used as samples to estimate the underlying states/symbols for the speech signal.

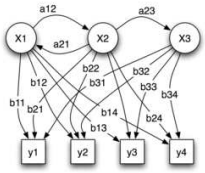


Fig.7 HMM representation

HMM can be represented mathematically as $M=\{A, B, \pi\}$

$$A=\begin{Bmatrix} a_{11}.....a_{1n} \\ a_{21}.....a_{2n} \\ ..... \\ a_{n1}.....a_{nn} \end{Bmatrix} \qquad S=\{(\mu_1,\Sigma_1),(\mu_2,\Sigma_2).....(\mu_n,\Sigma_n)\}$$

$$\pi=\{\pi_1,\pi_2,\pi_3.....\pi_n\}$$

A is the transition matrix whose (i,j)th element represents the probability of transiting from symbol in time step t to symbol in time step t+1: $a_{ij}=p(q_{t+1}=S_j \mid q_t=S_i)$. $B=b_i(O_t)$ represents the probability of the observation MFCC vector being generated in state Si [9].

$\pi=\{\pi_j = p(q_1=Sj \mid M)\}$ represents the probability of the first state is in state Sj given the HMM model M. Each state or symbol is formed by a 39-dimensional mean vector $\sum$ and a variance vector μ, and each state has a transition matrix A changes from states to states as time processing.

$$b_j(O_t)=\frac{1}{\sqrt{(2\pi)^n \mid \Sigma_j \mid}} e^{-\frac{1}{2}(O_t-\mu_j)'\Sigma_j^{-1}(O_t-\mu_j)}$$

A prototype HMM which contains the numbers of states that the model has (N), the mean vector ($\sum$), the variance vector (μ) and the transition matrix (A) is defined. Training data is then passed into the model for re-estimation purpose. According to Baum-Welch algorithm, the parameters are updated until the calculated probability produced by this word $P(O \mid M)$ is maximized.

The sampling data is passed into the prototype HMM, the parameters in the prototype HMM is then updated by applying Baum-Welch algorithm [10]. A new HMM model $M'=(A', B', \pi')$ with the updated parameters is obtained after Baum-Welch calculation. It can be derived that $P(O \mid M') > P(O \mid M)$. Hence, if M' is iteratively used to replace M and the above re-estimation is repeated, the probability of O being observed from the model is improved until it reaches certain limiting value. HMM model for the word is successfully built by then.

B.3 Recognition process

In the recognition process, a sequence of observed MFCC vectors $\{O_1,O_2,O_3...O_T\}$ generated from the input speech is passed into the HMM database, the probability that the observations are generated by each HMM model $p(O \mid M)$ is then calculated. The HMM model that produces the maximum probability is selected as the recognition result. $p(O \mid M)$ can be calculated by applying Forward Algorithm and Backward Algorithm.

$$p(O \mid M)=\sum_{i=1,2,...T}\pi_i b_i(O_1)a_{12}.....a_{T-1}a_T b_T(O_T)$$

Forward Algorithm: define a forward variable $a_i(t)$ as the probability of the partial observation sequence $\{O_1,O_2,O_3...O_t\}$, when it terminates at the state S_i. $a_i(t)=p\{O_1,O_2,.....O_t,q_t=Si \mid M\}$, the forward algorithm can be applied.

Step 1: Initialization

$$a_j(1)=\pi_j b_j(O_1),1\le j\le N$$

Step 2: Recursion

$$a_j(t+1)=b_j(O_{t+1})\sum_{i=1}^{N}a_i(t)a_{ij},1\le j\le N,1\le t\le T-1$$

Step 3: Termination

$$p(O \mid M)=\sum_{i=1}^{N}a_i(T)$$

Backward algorithm: define a backward variable $\beta_i(t)=p\{O_{t+1},O_{t+2},.....O_T \mid q_t=Si,M\}$, which demonstrated the probability of the partial observation sequence from back $\{O_{t+1},O_{t+1}...O_T\}$ given the current state is S_i.

Step 1: Initialization

$$\beta_i(T)=1.....(1\le i\le N)$$

Step 2: Recursion

$$\beta_i(t)=\sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\beta_j(t+1).....(1\le t\le T-1;1\le j\le N)$$

Step 3: Termination

$$p(O \mid M)=\sum_{i=1}^{N}\pi_i b_i(O_1)\beta_i(1)$$

Finally, the probability of the module producing the observed sequence vectors can be calculated by combining the forward parameters and the backward parameters to form the Forward-Backward parameters, which is called Forward-Backward algorithm [11]:

$$p(O,q_t=S_i \mid M)=a_i(t)\beta_i(t)\Rightarrow p(O \mid M)=\sum_{i=1}^{N}a_i(t)\beta_i(t)....(1\le t\le T)$$

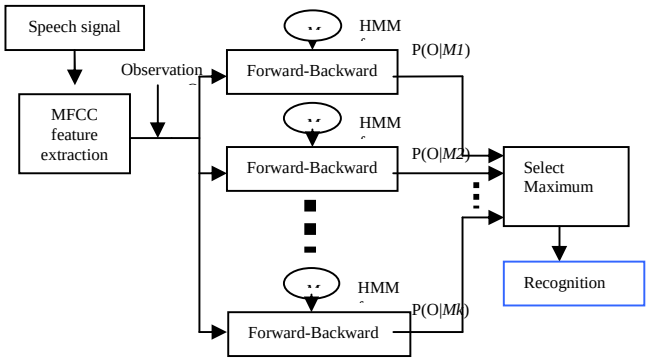


Fig.9 Recognition process [11]

D. Computer System Interface

A Recognition process

The recognition system designed in HTK runs in DOS command window, which is not user-friendly and provides poor interaction between human and the system. It is necessary to design a computer user interface so that driver can use in a more convenient way.

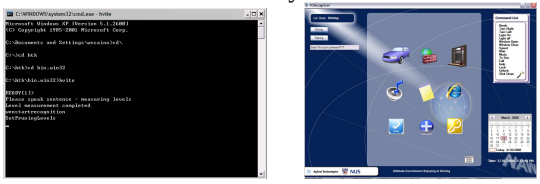


Fig.10 Voice recognizer interface

Communication between the recognizer and the system interface program has to be done so that the interface can take the action according to the recognition result. To achieve the communication step, share memory method is employed.

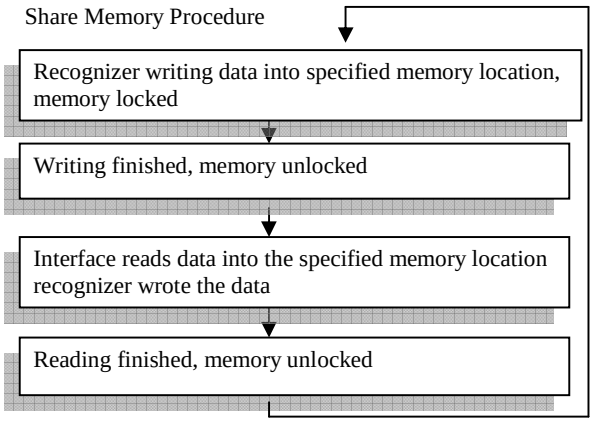


Fig.11 Share Memory process

B Encoding

In the share memory procedure, only integer can be written into the memory block and read by other programs. However, the results that the recognizer produces are word/phrase, which is in string format. To solve such problem, encoding has to be done to map the word string to integer respectively. Each word/phrase can be represented as an integer value and send to the interface, the interface will then decode the integer message according to the encoding table and take the respective action as required.

Table. I Encoding table

Word	File Name	Integer Code
Yes	ye	3
Cancel	ca	4
Music	mu	5
……	……	……

IV. RECOGNITION RESULTS

A Recognition accuracy analysis

Table. II Recognition result table

Word	Number	Fail	Accuracy
Yes	50	2	96%
Cancel	50	4	92%
Music	50	2	96%
Speed	50	5	90%
Phrase	**Number**	**Fail**	**Accuracy**
Window open	50	6	88%
Window close	50	7	86%
Light on	50	14	72%
Light off	50	13	74%

Obviously, the recognition result shows that discrete word model has higher recognition accuracy than phrase model. The reason behind is that phrase is constructing by two discrete words and a quiet interconnecting region. The duration of the quiet region is highly dependent on the speaking speed. The speaking speed for people is changing due to emotion change and so on. Hence, the duration of the quiet region is not fixed, which makes the modeling of this region difficult and not accurate. But this region is counted as part of the HMM model, so the inaccuracy of this part will results in the inaccuracy of the overall phrase HMM model accuracy.

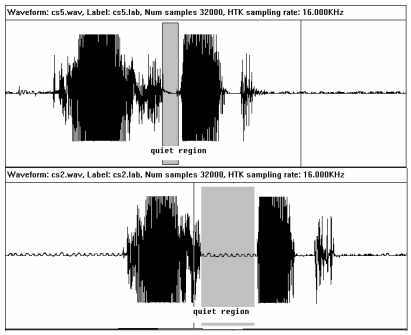


Fig.12 Phrase waveform (fast speaking V.S slow speaking)

This can be solved by reducing the duration of the quiet region as much as possible to make the phrase more like a discrete word. During the training step, the speed for speaking phrase should also be fast to minimize the quiet region. The speaking speed in the recognition should also be fast.

V. CONCLUSION

The system designed in this project is a simulation program runs in PCs instead of real vehicles. Future improvement can be made to enable system to control a car model, which requires hardware design. Another direction for the future development of speech recognition system is to expand its application to chip-level [12] to make it embedded inside handheld devices such as PDA, hand phones and so on.

ACKNOWLEDGMENT

Author thanks the following for their dedicated assistance in this project:

1. Asst. Prof. Ha Yajun and Asst Prof. Sadasivan Puthusserypady K for their supervision and suggestions to the project.
2. The Agilent engineering team for their support on DAQ hardware specified issues.
3. The labs at National University of Singapore, Mr Abdul Jalil bin Din (PCB fabrication lab), Ms Chia Meow Hua (Digital Electronics Laboratory),Mr Teo Seow Miang (Signal Processing & VLSI Laboratory), Mr Tan Chee Siong (Mechatronics & Automation Laboratory)
4. Mr Zhang Yi, NTU senior master student

REFERENCES

[1] M. C. McCallum, J. L. Campbell, J. B. Richman and J. L. Brown, "Speech Recognition and In-Vehicle Telematics Devices: Potential Reductions in Driver Distraction", International Journal of Speech Technology, vol. 7, no. 1, pp 25-33, Jan 04

[2] Judith A.Markowitz, "Using Speech Recognition", Prentice Hall PTR, 1996, pp.35—43.

[3] LAWRENCE R.RABINER, fellow IEEE, "A Tutorial n Hidden Markov Models and Selected Applications in Speech Recognition," , proceedings of the IEEE,Vol.77, No.2, February 1989, pp255-257.

[4] Agilent Technologies, "Agilent U2300A Series Multifunction USB Data Acquisition, programming guide", 1st edition, October 30, 2006, pp. 101–103.

[5] The Sonic Spot, "Wave File Format", Retrieved December 15, 2007 from http://www.sonicspot.com/guide/wavefiles.html

[6] Mark Hasegawa-Johnson, "Lecture 3: Acoustic Features", June 27, 2005, pp 3-5, http://www.ifp.uiuc.edu/speech/courses/minicourse/

[7] C.C.Kov, "Signals", 3rd edition, published by Prentice Hall, 2004, pp 55-58

[8] L. A. Webster , "THE POWER OF THREE: THE FRONT-END WITH SPECIAL EMPHASIS ON FFTs, LP TRANSFORMATIONS AND FEATURE SELECTION ", MS State Speech Conference , Spring 96. pp23-26, Available: http://www.ece.msstate.edu/research/isip/publications/courses/ece_8463/projects/1996_spring/conference/paper_webster.pdf

[9] Sigurdur Sigurdsson, Kaare Brandt Petersen and Tue Lehn-Schiøler, "Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music", pp 4-5

[10] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland, "The HTK Book", Cambridge University Engineering Department, 2003, pp 68-70

[11] Sadaoki Furui, "Digital Speech Processing, Synthesis, and Recognition", 2nd edition, Marcel Dekker, Inc, 2001, pp288-395

[12] J.N.Holmes, "Speech Synthesis and Recognition", Van Nostrand Reinhold Co.Ltd, 1988, pp 169--177