Proceedings of the International Conference on Man-Machine Systems (ICoMMS)
11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA

Remotely Navigated Mobile Platform Using Voice Command

Khairulnizam Othman, K.Sundaraj
Universiti Malaysia Perlis (UniMAP)
School of Mechatronic Engineering
02600 Jejawi-Arau, Perlis, MALAYSIA
khairulnizam786@yahoo.com

Abstract- **Voice is one of convenient methods to communicate between human and robots. To command a robot task by voice, voices of the same number have to be able to be recognized But the more the number of recognition voice is, the higher the cost of recognition system is and the longer the time of recognition is. A large class of problems requires real-time processing of complex structured data in real-time. These are difficult problems to solve by state logic and compare techniques, since they require capturing complex structures of voice and relationships in massive quantities of low precision, ambiguous noisy data. In this paper, we develop a master processing unit that will process voice recognition which is a notebook. It will give command using serial data transfer to a microcontroller for further processing for remote navigate mobile Platform. Nonholonomic mobile robot like car is used for base. The aim is to build a prototype which can be analyze to create a mobile platform that can communicated with human by using voice command in the future.**

Keywords- **voice Recognition, Remote Navigated Mobile Platform, Nonholonomic, Car, Real Time.**

I. INTRODUCTION

Since human usually communicate each other by voices, it is very convenient if voice is used to command mobile robot. A lot of researches have been purpose. Some of them proposed algorithm of voice recognition and others concentrated on the application of voice recognition rather than voice recognition itself. Some researches discussed the effectiveness of voice input compare with other input device in robot systems [1-3].

Program develop to compare the every speech that say in the microphone to text then compare with a size for that word that came out. This is become easy by using Microsoft Foundation Class library speech. Application wizard has created this Speech application for us to reprogram and edit. This application not only demonstrates the basics of using the Microsoft Foundation classes but is also a starting point for writing your application. There couple of sub file for speech application then dialog class for put on view text. Followed by Comparison between list of order and dialog came out in text. Then communication between the main processes with our microcontroller happen using serial data transfer.

II. ARCHITECTURE

The speech SDK5.1 SAPI can be built in by using visual c++. The system uses Speech SDK5.1, a development component for the Speech procedures provided by the Microsoft Corporation .In this section, we will provide an overview of Speech SDK5.1, starting with an introduction of HMM-based recognizers.

Overview of an HMM-based Speech Recognition System

Speech SDK5.1 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, which is a type of statistical model. In HMM-based speech recognizers, each unit of sound (usually called a phoneme) is represented by a statistical model that represents the distribution of all the evidence (data) for that phoneme. This is called the acoustic model for that phoneme. When creating an acoustic model, the speech signals are first transformed into a sequence of vectors that represent certain characteristics of the signal, and the parameters of the acoustic model are then estimated using these vectors (usually called features). This process is called training the acoustic models[4]-[10].During speech recognition, features are derived from the incoming speech (we will use "speech" to mean the same thing as "audio") in the same way as in the training process. The component of the recognizer that generates these features is called the front end. These live features are scored against the acoustic model. The score obtained indicates how likely that a particular set of features (extracted from live audio) belongs to the phoneme of the corresponding acoustic model. The process of speech recognition is to find the best possible sequence of words (or units) that will fit the given input speech. It is a search problem, and in the case of HMM-based recognizers, a graph search problem. The graph represents all possible sequences of phonemes in the entire language of the task under consideration. The graph is typically composed of the HMMs of sound units concatenated in a guided manner, as specified by the grammar of the task. As an example, let look at a simple search graph that decodes the words "one" and "two". It is composed of the HMMs of the sounds units of the words "one" and "two":

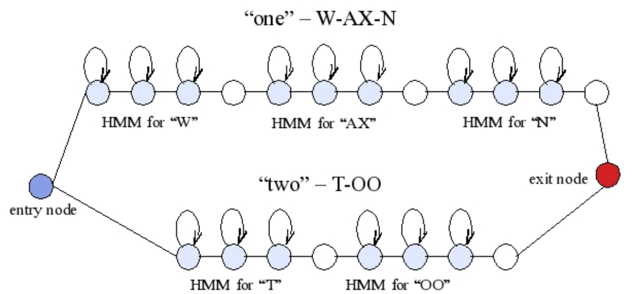


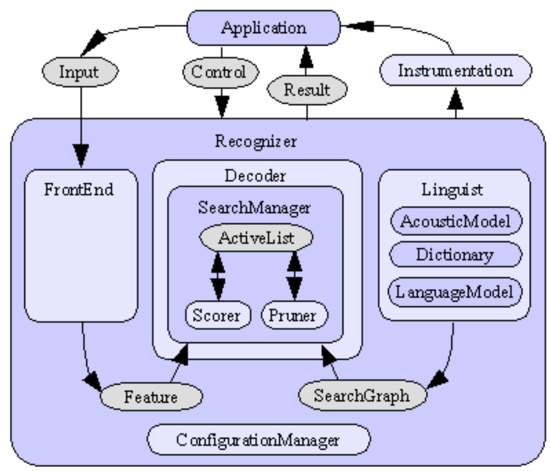
Fig. 1. Search Graph

Constructing the above graph requires knowledge from various sources. It requires a dictionary, which maps the word "one" to the phonemes W, AX and N, and the word "two" to T and OO. It requires the acoustic model to obtain the HMMs for the phonemes W, AX, N, T and OO. In Speech SDK5.1, the task of constructing this search graph is done by the linguist. Usually, the search graph also has information about how likely certain words will occur. This information is supplied by the language model. Suppose that, in our example, the probability of someone saying "one" (e.g., 0.8) is much higher than saying "two" (0.2). Then, in the above graph, the probability of the transition between the entry node and the first node of the HMM for W will be 0.8, while the probability of the transition between the entry node and the first node of the HMM for T will be 0.2. The path to "one" will consequently have a higher score. Once this chart is constructed, the sequence of parameterized speech signals (i.e., the features) is matched against different paths through the graph to find the best fit. The best fit is usually the least cost or highest scoring path, depending on the implementation. In speech SDK5.1, the task of searching through the chart for the best path is done by the search manager.

As you can see from the above chart, a lot of the nodes have self transitions. This can lead to a very large number of possible paths through the chart. As a result, finding the best possible path can take a very long time. The purpose of the prune is to reduce the number of possible paths during the search, using heuristics like pruning away the lowest scoring paths.

As we described earlier, the input speech signal is transformed into a sequence of feature vectors. After the last feature vector is decoded, we look at all the paths that have reached the final exit node (the red node). The path with the highest score is the best fit, and a result taking all the words of that path is returned.

Basic idea Speech SDK5.1 Architecture and Main Components



Fig, 2. Architecture Diagram

When the recognizer starts up, it constructs the front end (which generates features from speech), the decoder, and the linguist (which generates the search graph) according to the configuration specified by the user. These components will in turn construct their own subcomponents. For example, the linguist will construct the acoustic model, the dictionary, and the language model. It will use the knowledge from these three components to construct a search graph that is appropriate for the task. The decoder will construct the search manager, which in turn constructs the scorer, the pruner, and the active list.Most of these components represent interfaces. The search manager, linguist, acoustic model, dictionary, language model, active list, scorer, pruner, and search graph are all visual C++ interfaces. Microsoft Speech SDK 5.1 is based on Windows COM Development Kit. This SDK contains Voice Application Programming Interface (SAPI), Microsoft continuous speech recTTS376,engine0898pThis

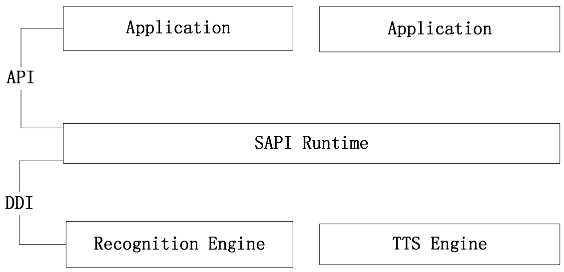


Fig. 3. SAPI 5.5 System structure diagram

Speech engine through the device driver interface (DDI) layer and SAPI runtime communication, application (Application) through the Application Programming Interface (API) layer and SAPI interaction[6]. Through the use of these API will be able to carry out speech recognition and speech synthesis aspects of software development.

We chose to use this method which is pc base for couple reason because it real time data transfer and suitable for real life application for Remote Navigated Mobile Platform using Voice Commands. Beside that it can be implemented easily future than voice command only. Furthermore we also has study supplementary hardware that can bring the same result, each of them has some disadvantage which make it not suitable for real life application. Below give a table of different method that used in speech recognition and difficulty for each system [11]-[13].

TABLE 1.
Speech Recognition Method

Product name	Model / Method	Disadvantage
Circuit SR-07	Train on keypad.	1.Limited command .(40)
RSC-4x Series	Phyton C compiler and assembler.	1. To complicate for user build the program. Because it more useful for signal processing.
Robust Speech Recognition	Hidden Markovl Mode	1. Big scale Hardware. 2. Can attenuate environments noise.
ATC18 64Kx32-bit	User decide	1. Not suitable for extend utilize which need big memory space.

III DRIVE AND CONTROL SYSTEM

Another important aspect is the control system design and frame work to support the implementation of our control schemes. The principal control system is PIC16F877 microcontroller which is additional as the main brain for the mobile platform. Voice recognition for command the robot for doing any task is given. The base is build with stepper motor to get an accurate position and smooth speed.

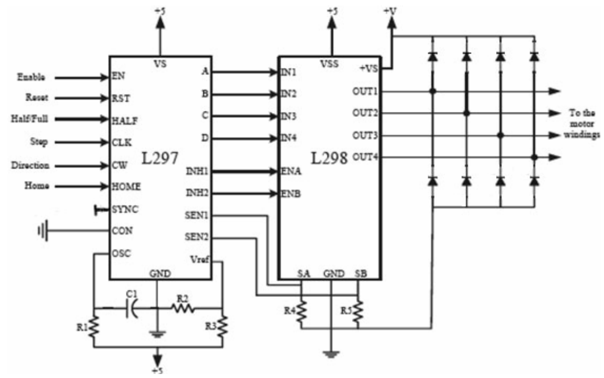


Fig. 4. L297 and L298 Stepper Motor Driver.

From figure above we can see that all control pin is broad out for programming purpose. This give a very effective control method which make our stepper motor easily control . This makes the movement of mobile robot also smoother. From The regular driver like ULN2003 the controlling method does not have enabled , home which indicate position, reset and control low chopper act on INH&INH2. Beside that the step clock frequency also easy to control.

TABLE 2
Controlling input /output

Pin Num	Name	Description
Pin 03	Home	Indicate when state = ABCD = 0101
Pin 18	Clock	Clock Pulse for speed control
Pin 10	Enable	Low signal causes INH1, INH2, A, B, C, D to low
Pin 11	Control	Low, chopper acts on INH1 & INH2. High,
Pin 17	CW /CCW	Direction control input
Pin 19	HALF/ FULL	High, Half step or low, full step
Pin 20	Reset	Active low resets counter ABCD = 0101

Differential drive

For kinematics analysis for Remotely Navigated Mobile Platform (RNMP), we can simplify with one of the simplest mobile robot constructions is a chassis with two fixed wheels. Understanding this construction helps you to grasp some basic kinematics of car-like robots. Usually differential drive mobile robots have an additional castor wheel as the third fulcrum. It is usually used for stability. Sometimes roller-balls can be used but from the kinematics point of view, there are no differences in calculations. As it can rotate freely in all directions, in our calculation we can omit the castor wheel because it only has a very little influence over the robot's kinematics In case of differential drive, to avoid slippage and have only a pure rolling motion, the robot must rotate around a point that lies on the common axis of the two driving wheels. This point is known as the instantaneous center of curvature (ICC) or the instantaneous center of rotation (ICR) .

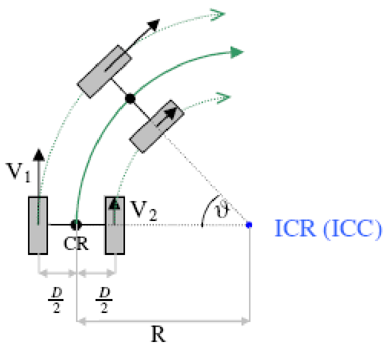


Fig. 5. The differential drive motion .

At each moment in time the left and right wheels follow a path (Fig. 1) that moves around the ICR with the same angular rate $\omega = \frac{dv}{dt}$,and thus:

$$\omega.R = v_{CR} \qquad \text{Equation 1}$$

$$\omega.(R + \frac{D}{2}) = v_1$$

$$\omega.(R - \frac{D}{2}) = v_2$$

Where *v1* is the left wheel's velocity along the ground, and *v2* is the right wheel's velocity along the ground, and *R* is the signed distance from the ICC to the midpoint between the two wheels. Note that *v1*, *v2*, and *R* are all functions of time. At any moment in time;

$$R = \frac{v_2 + v_1}{v_2 - v_1} \cdot \frac{D}{2} \qquad \text{Equation 2}$$

$$\omega = \frac{v_2 - v_1}{D} \qquad \text{Equation 3}$$

The velocity of the CR point, which is the midpoint between the two wheels, can be calculated as the average of the velocities v1 and v2:

$$V_{CR} = \frac{v_2 + v_1}{2} \qquad \text{Equation 4}$$

Mobile robot differential drive

Let's try to find formulas by means of which is possible to compute the actual position of the robot.

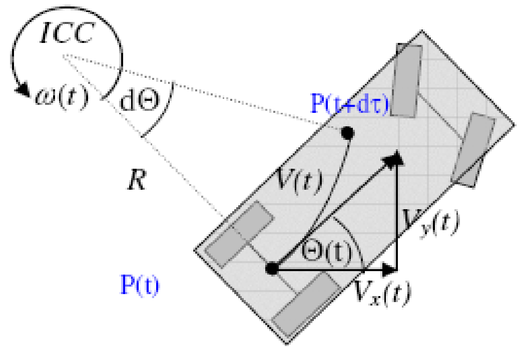


Fig. 6. Differential drive for four wheels mobile robot.

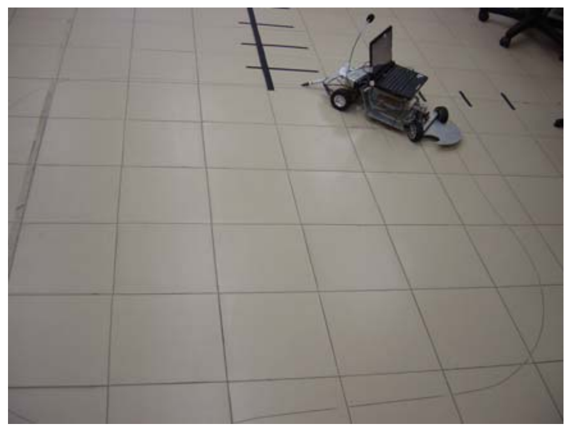


Fig. 7. Real application on floor

Suppose that a differential drive robot is rotating around the point ICC with an angular velocity *w(t).* During the infinite short time *dt* the robot will travel the distance from the point *P(t)* to *P(t+dt)* with a linear velocity *V(t)* (for more details about the velocity of the
CR point see section Differential Drive). *V(t)* has two perpendicular components, one along the *X* axis – *Vx(t),* and the other along the *Y* axis – *VY(t).* For infinite short time we can assume that the robot is moving along a straight line tangent in the point P(t) to the real trajectory of the robot. Based on the two components of the velocity *V(t),* the traveled distance in each direction can be calculated:

$$dx = V_x(t).d\tau \qquad \text{Equation 5}$$

$$dy = V_y(t).d\tau$$

Where:

$$V_x(t) = V(t).\cos[\Theta(t)] \qquad \text{Equation 6}$$

$$V_y(t) = V(t).\sin[\Theta(t)]$$
$$d\Theta = \omega(t).d\tau$$

IV. RESULT AND DISCUSSION

Table 3. Experiment speed and load for half and full step

Frequency(Hz)	RPM Half	Weight Half	RPM Full	Weight Full
5	0.75	103.00	1.5	64.75
10	1.50	103.02	3.05	64.76
20	3.00	101.04	6	64.77
30	4.75	95.16	9	64.78
40	6.05	90.25	12.25	56.90
50	7.54	79.46	15.05	49.05
60	9.17	70.63	18.44	40.22
70	10.67	63.77	21.6	35.32
80	12.17	60.82	24.42	23.54
90	13.75	57.88	25.33	17.66
100	15.30	51.01	30.25	13.73
110	16.83	44.15	32.5	9.81
120	18.00	24.53	36.42	8.83
130	20.7	18.63	37.45	6.87
140	21.42	11.77	40.45	4.91
150	23.00	9.81	45	2.94

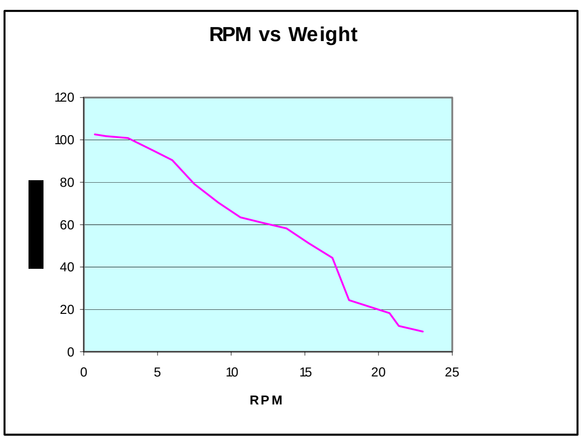


Fig. 8. Full Step

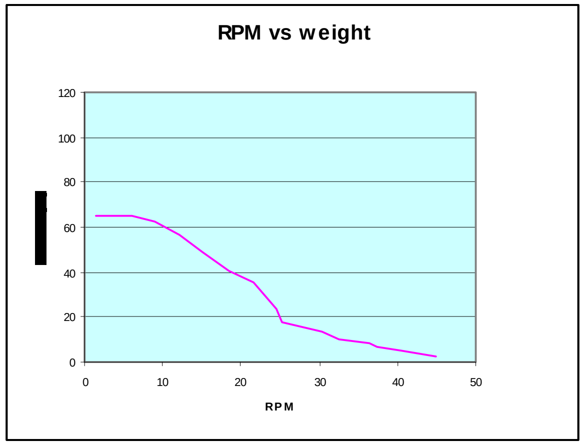


Fig. 9. Half Step

We choose to use full step mode because as you can see above graph clearly show the weight for it is higher while speed quite lower than half step . Hence these projects need some additional stack such as notebooks and battery. Therefore we choose 70 Hz as step clock for back wheel which give 10.67 RPM on half step mode. This can carry load up to 63.77 Newton or 6.5 Kilos. For this information we start to compare the theory and the real application on floor.

$$r = 0.045\ m$$
$$c = 2\lambda r$$
$$c = 0.28278\ m$$
$$10.67\ rpm = \frac{c \times 10.67}{60}$$
$$10.67\ rpm = 0.05029\ ms^{-1}$$

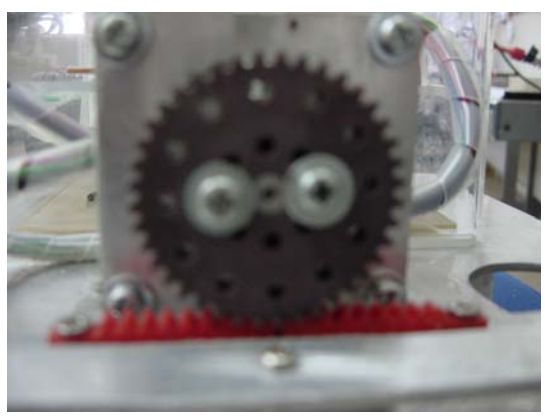


Fig. 10. Front wheel cornering motion is used Rack and Pinion Joint.

From figure 10 we can see the front tire use rack and pinion joint for control the front wheels angle when turn. Since the pinion gear has 40 teeth the angle for cornering is limited.

$$T = 40$$

$$\phi = \frac{360^{O}}{40}$$

$$\phi = 9^{O}$$

If we take in between gear teeth we can get small angle disarticulation for front is up to 4.5 degree.

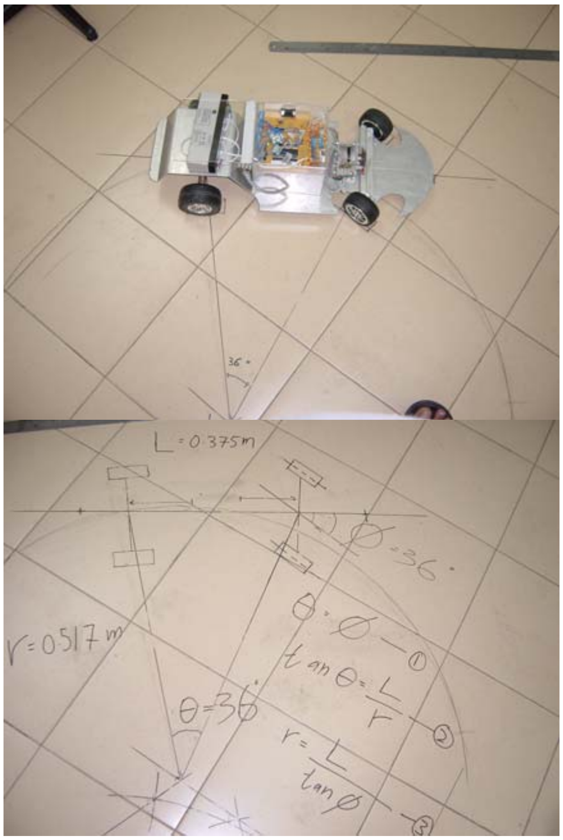


Fig. 8. The Simple car has three degree of freedom, but the velocity space at any configure is only two-dimensional.

TABLE 4
Result for angle displacement

Step	ϕ	R (m)	θ	T (time)
5	4.5	-	-	-
10	9.0	2.360	10	4.54
15	13.5	1.530	14	3.25
20	18.0	1.150	18	2.25
25	22.5	0.905	23	1.54
30	27.0	0.735	27	1.33
35	31.5	0.612	31	1.20
40	36.0	0.517	36.	1.08

From Table 4 , we can assume the ϕ equal θ. From this result we came out one formula.

A. Advance task Using Voice Command

Efficiency for voice in application is depending on the training that conducted. Meanwhile the distance between mouth and microphone must position at same distance for all time [14]-[17].

TABLE 5
Recognition Probability

Command	Number of testing	Recognition probability
Start	40	97.5%
Stop	40	97.5%
Go	40	97.5%
Back	40	97.5%
Right	40	97.5%
Left	40	95%
Line	40	97.5%

Recognition probability show a good result for voice command. However went we give two word at one command the probability became slighter. For check whether we give correct command there is COM window such as figure 11.

V. Conclusion

A voice command system for mobile robots is brilliant idea. In order to recognize voice commands more than the maximum number of recognizable voice for one voice recognition processor, all voice commands are grouped and organized such that the voice recognition processor always tries to find a similar voice pattern from one memory unit of which the voice number is not more than the maximum number of recognizable voice. In order to translate the gains made by research systems in recognition accuracy into practical use we need to filter continuous speech recognition designed for command. Then the send that command thru serial data transfer for remotely command the mobile platform. Thus, this system can be applied to many areas which require very high recognition rate. Perhaps in future human and robot can talk for each other.

Acknowledgment

The author would like to thank Ministry of Science, Technology and Innovation of Malaysia (MOSTI) for financial support. Special thank also goes Assoc Prof Dr. Kenneth Sundaraj for giving me the opportunity to develop this project. Thank You.

REFERENCES

[1] N, Yamasaki a d Y. Atlzai, "Active Interface for Human-

[2] Robot Interaction," Proceedings of the 2995 ZEEE Internutional Conference on Robotics and Automation, Nagoya, Japan, May, pp. 3103-3109, 1995.

[3] D. S. Lees and L. J. Leifer, "A Graphical Programming Language for Robots Operating in Lightly Structured Environments," Proceedings of the I993 IEEE International Conference on Robotics and Automation, San Diego, California, May, pp. 648453,1993.

[4] T. Nishimoto et. al., "proving Human Interface in Drawing Tool Using Speech, Mouse and Key-board," Proceedings of the 4th IEEE International Workshop on Robot and Human Communication, Tokyo, Japan, July, pp. 107-112, 1993.

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

[6] T.B. Hughes, H.-S. Kim, J.H. DiBiase, and H.F.Silverman, .Performance of an HMM. speech recognizer using a realtime tracking microphone array as input,. *IEEE Trans. Speech and Audio Process.*, vol. 7, no. 3, pp. 346.349, 1999.

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

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

[9] L.E. Baum and T. Petrie, " Statistical Inference for Probabilistic Functions of Finite State Markov Chains", Ann. Math. Stat., vol. 37 ,pp 1554-1563, 1966.

[10] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, " A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", Ann. Math. Stat., vol. 41, no.1, pp 164-171, 1970.

[11] L. E. Baum, "An Inequality and Associated Miximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes", Proc. Symp. On Inequalities, vol. 3, pp 1-7, Academic Press, New York and London 1972.

[12] A. Chandrakasan and S. Sheng and R. Brodersen. Low Power CMOS Digital Design. JSSC, V27, N4, April 1992, pp 473{484.

[13] ATMEL. ATMEL ATC18 64K x 32-bit Low-power Flash. 2002. http://www.atmel.com.

[14] CMU Speech. List of Speech Recognition Hardware Products.2002.http://www.speech.cs.cmu.edu/comp.speech/ Section6/Q6.5.html.

[15] C. Carter and R. Graham, .Experimental Comparison of Manual and Voice Controls for the Operation of In-Vehicle Systems,. In *Proceedings of the IEA2000/HFES2000 Congress*, Santa Monica, CA

[16] J.H.L. Hansen, .Analysis and Compensation of Speech under Stress and Noise for Environmental Robustness in Speech Recognition,. *Speech Communications*, Special Issue on Speech Under Stress, vol. 20(2), pp. 151-170, November 1996.

[17] A. Baron and P. Green, .Safety and Usability of Speech Interfaces for In-Vehicle Tasks while Driving: A Brief Literature Review,. Technical Report UMTRI-2006-5, Feb. 2006.

[18] J. H.L. Hansen, J. Plucienkowski, S. Gallant, R. Gallant, B. Pellom, and W. Ward, .CU-Move: Robust speech processing for in-vehicle speech systems,. in *ICSLP*, pp. 524.527,2000.