

Manual Wheelchair Automator:
Design of a Speech Recognition System with Hidden
Markov Models and Joystick Steering Control

by

Hailun Huang

Electrical and Biomedical Engineering Design Project (4BI6)
Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada

Manual Wheelchair Automator:
Design of a Speech Recognition System with Hidden
Markov Models and Joystick Steering Control

by

Hailun Huang

Electrical and Biomedical Engineering
Faculty Advisor: Dr. Thomas Doyle

Electrical and Biomedical Engineering Project Report
submitted in partial fulfillment of the degree of
Bachelor of Engineering

McMaster University
Hamilton, Ontario, Canada

April 27, 2009

Copyright © April 27, 2009 by Hailun Huang

ABSTRACT

Manual Wheelchair Automator:
Design of a Speech Recognition System with Hidden
Markov Models and Joystick Steering Control

by
Hailun Huang
Electrical and Biomedical Engineering

The speech recognition system, as one of the steering control components of the Manual Wheelchair Automator (MWA), is designed to benefit the end users who have lost control of their upper extremities. An alternative joystick steering method is incorporated into the control system to provide the users with more options. In particular, the speech recognition system consists of two parts: the first part is a small vocabulary training section that constructs a model, and the second is a speech recognition section that uses this model. This speaker independent, discrete word speech recognition system is implemented on an 80 MHz, 32 bit microcontroller. Much research has been done on the existing techniques on implementing speech recognition. The result is that a Hidden Markov Models (HMMs)-based method with Viterbi re-estimation and Forward-Backward Model are both selected for model construction based on its reliability and relative efficiency. Our experimental results indicate that Forward-Backward re-estimation provides a better performance. Also, the speech recognition section adopts the Viterbi Algorithm to measure the likelihood of speech input to the model of the command words. The algorithm, design of the system, experimental data, and future improvements are presented in detail.

Key words: Manual Wheelchair Automater (MWA), speech recognition and training, Hidden Markov Models (HMMs), Viterbi Algorithm, Forward-Backward Algorithm, joystick, microcontroller, steering control system

ACKNOWLEDGMENTS

I would like to first thank my two supporting partners in this project, Ling Tsou and Erika Schimek for your company, support, advice and so much more. I could not have done it without you. I must also thank my course instructor and my project supervisor Dr. Doyle, who has given us the direction and provided assistance throughout the year. Diane Tait from St. Peter's Hospital lent our group a wheelchair for the project purposes. Your generosity is greatly appreciated. Last but not least, I thank my parents and my friends who have always trusted me and stand by me to help come this far.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
NOMENCLATURE	viii
1 Introduction	1
1.1 Background.....	1
1.2 Objectives	3
1.3 General Approach to the Problem	5
1.4 Scope of the Project.....	7
2 Literature Review.....	11
2.1 Speech Recognition with HMMs	11
2.2 HMM Recognition using Viterbi and Forward-Backward Algorithms	12
2.3 HMM Training using Viterbi and Forward-Backward Algorithm.....	13
2.4 Microcontroller Selection	14
3 Statement of Problem and Methodology of Solution	16
3.1 HMMs in Automatic Speech Recognition system	16
3.1.1 Discrete Markov Processes	17
3.1.2 Hidden Markov Models (HMM)	19
3.1.2.1 A HMM Example	20
3.1.2.2 Elements of an HMM.....	21
3.1.2.3 HMM Constraints for Speech Recognition Systems	24
3.1.2.4 The Three Basic Problems for HMMs	26
3.2 Joystick Steering Control.....	28
4 Experimental or Design Procedures.....	30
4.1 Top-Level Automatic Speech Recognition Design.....	30
4.1.1 HMM Recognition with Viterbi Algorithm	32
4.1.2 HMM Training	37
4.1.2.1 Viterbi Re-estimation.....	39
4.1.2.2 Forward-Backward Re-estimation	41

4.1.2.3 Laplace Smoothing	44
4.2 Joystick Steering Control Design	45
5 Results and Discussion.....	47
6 Conclusions and Recommendations	50
6.1 Conclusions.....	50
6.2 Recommendations	51
Appendix A: Raw Speech Signal Samples within a linear window	52
Appendix B: Viterbi Recognition results	55
Appendix C: HMM training and recognition implementation in C programming language	57
References	58
VITA.....	60

LIST OF TABLES

Table 1: Weather Expectation Probabilities	19
Table 2: Automatic Speech Recognition Rates	47
Table 3: HMM training run-time in sec	48
Table 4: Total MWA Project Cost	49

LIST OF FIGURES

Figure 1: Ideal MWA (8)	4
Figure 2: General MWA Block Diagram	5
Figure 3: MWA System Block Diagram	6
Figure 4: Functional scheme of an Automatic Speech Recognition System (6)	8
Figure 5: Overview of Automatic Speech Recognition and joystick steering control systems	10
Figure 6: PIC32 Starter Kit (16)	15
Figure 7: A Markov chain with 5 states (labeled S1 to S5) with selected state transitions (13)....	17
Figure 8: An N-state urn and candy model which illustrates the general case of a discrete symbol HMM (13)	21
Figure 9: 4-state Left-Right HMM with no skip transitions	25
Figure 10: Joystick Implementation.....	29
Figure 11: Block diagram of an isolated word HMM recognizer (13)	32
Figure 12: Viterbi Algorithm (19)	35
Figure 13: Optimum Search Possibilities	37
Figure 14: Block Diagram for HMM training	38
Figure 15: Trellis Illustration of Viterbi (18)	39
Figure 16: Viterbi Training Algorithm (20)	40
Figure 17: Illustration of the sequence of operations required for the computation of the joint event (13)	42
Figure 18: Joystick Command Mapping	46

NOMENCLATURE

MWA:	Manual Wheelchair Automator
HMMs:	Hidden Markov Models
LVQ:	Learning Vector Quantization
ADC:	Analog-to-Digital Converter
DTW:	Dynamic Time Wrapping

1 Introduction

1.1 Background

Since the first folding, tubular steel wheelchair was invented in 1932 and the first electric wheelchair was designed in the 1950's (1), the functionalities of the manual and electric wheelchair have been improved significantly. The electric wheelchairs have become more efficient, quieter and lower-maintenance in general. They also grant users more freedom with less assistance including in the control, styles, range or travel distance, maneuverability, seating and other user options. In contrast, the mobility of the manual wheelchair was limited by the user's physical condition and restricted his or her daily activities. This disparity in performance is reflected in a difference in cost: electric wheelchairs typically range between \$1600 and \$7500 (2), while the basic manual wheelchairs cost around \$100 to \$500 (3). After researching various existing motors, we noticed that a standard bike could be converted into an electric bike by installing hub motors, which cost around \$300 per wheel (4). The conversion is very convenient and allows a much greater range of travel. Also, cars' windshield wiper motors can also be used to provide as much torque as the hub motors with a more competitive cost. Therefore, a manual wheel chair can be possibly converted into an electric wheel chair by adding the suitable motors.

Our Manual Wheelchair Automator (MWA) project is inspired primarily by the high cost difference, above 1000 dollars, between the existing power wheelchairs and the manual wheel chairs. Nowadays, sales and services of electric wheel chairs have become quite widespread in North America. For countries like Canada, there are government programs that will fund disable people to purchase electric wheel chairs. However, in many other parts of the world, there are people who find electric chairs too expensive to be afforded. The high cost of electric wheel chairs limits the freedom and personal mobility of those individuals who cannot pay for a wheel chair.

The design also includes an option to disable the MWA, which allows the user to switch between manual and electric modes to offer more flexibility to end-users. This feature can greatly benefit the more active users. Those users still maintain most of their extremity control and they can exercise their arm with converting the wheel chair to manual. The electric wheel chair option provides them more freedom to travel a longer distance and require less assistance.

Moreover, in order to benefit the end users who have lost control of their upper extremities due to injury, illness or disability, a speech recognition system is designed to be one of the steering control components of the MWA. An alternative joystick steering method is incorporated into the control system to increase users' selections as well. The first attempts towards machine recognition of speech were made many years before the development of the first digital electronic computer (5). After generations of hardware device growth and ever increasing signal processing capability, speech recognition nowadays is regarded by market projections as one of the more promising technologies of the future (6). There are a variety of speech recognition systems that have become commercially available. Those products and the research about speech recognition are our references in designing and implementing our MWA speech recognition steering control.

This Automatic Speech Recognition technology is based on a stochastic signal model called Hidden Markov Models (HMMs). HMMs are one of the most powerful models that allow description of complex non-stationary phenomena ranging from speech to stock market behavior (6). Speech research has driven most of the engineering subject in the HMMs in the last three decades (7). In the speech recognition community, much research has produced efficient algorithms and techniques related to HMMs. More detail will be discussed in the following Literature Review and Statement of Problem and Methodology of Solution chapter.

1.2 Objectives

The primary aim of the MWA project is to convert a manual wheel chair into an electrical one under a competitive budget. As it is mentioned previously in the background section, a price difference of 1000 dollars or more exists between electric and manual wheel chairs. Our project cost has to be less than the difference in order to be viable while achieving the functionalities.

The second objective of this design is flexibility. The users can easily disassemble MWA to convert the wheel chair from electric to manual. It will greatly benefit the users with a higher mobility level. If they want to exercise their arms, it can be achieved by taking off the MWA. This is the reason why we have to ensure the design has to be mounted in a way that can be disassembled without other assistance.

Last but not least, this MWA system is designed to be user-friendly. Considering different people's needs, there are two steering control options available: Automatic Speech Recognition and joystick steering control. As for end-users with little control of their upper extremities, they can speak the designated command words to a microphone. The commands will be recognized as control signals to the wheel chair. Alternatively, the user can also be able to control the speed and the direction of the wheel chair using a joystick steering device. Please refer to Figure 1: Ideal MWA above for the ideal design for the MWA system.



Figure 1: Ideal MWA (8)

My tasks mainly concentrate on the back-end design of the speaker-dependent discrete Automatic Speech Recognition system using HMMs and the joystick steering control. Both systems are implemented an 80 MHz, 32 bit microcontroller. The Automatic Speech Recognition converting the speech signal to into digital signals, and then outputting to the microcontroller for further signal processing to identify command words. Similarly, the joystick steering control takes speed and direction signals from the joystick device and convert them into digital signal. My objectives are to ensure that my design and implementation of the two steering control devices are of sensitive reception of user input and accurate responses. The speech recognition rate for Automatic Speech Recognition system has to be as high as possible to ensure accuracy and reliability.

1.3 General Approach to the Problem

My foremost focuses are to design an Automatic Speech Recognition steering control system using HMMs and a joystick steering control handheld device that will allow the user to select the speed and direction of the wheel chair. They both provide the control signals to the propulsion system of the wheel chair. Please refer to Figure 2: General MWA Block Diagram for the relationship among those three components of the project.

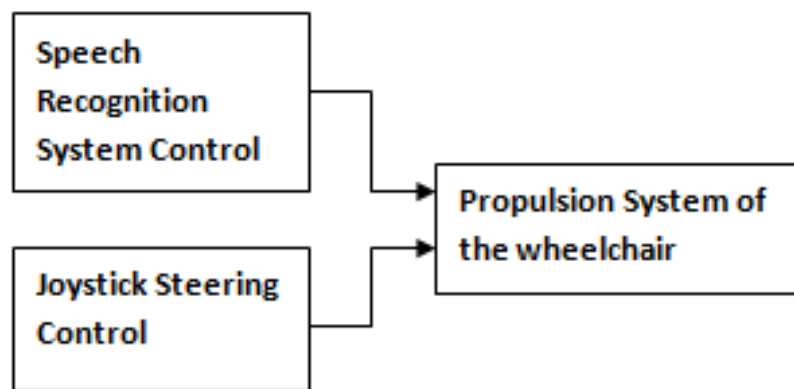


Figure 2: General MWA Block Diagram

AUTOMATIC SPEECH RECOGNITION

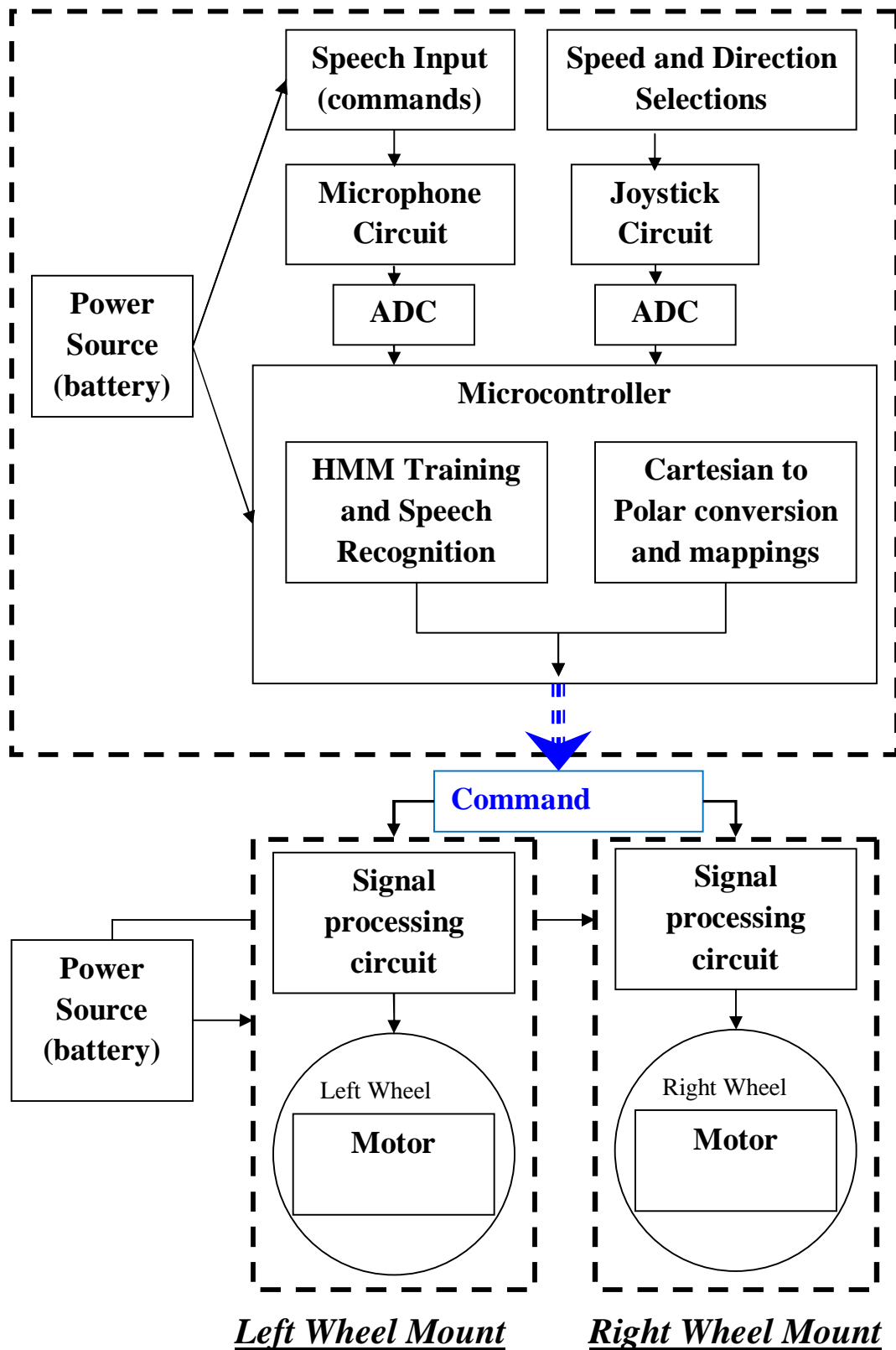


Figure 3: MWA System Block Diagram

More explicitly, the Automatic Speech Recognition and the joystick steering control handheld device each include an electric circuit that detects input signals and converts them from analog to digital. The output is sent to the microcontroller for further signal processing in the right format.

There are two parallel algorithms in microcontroller for speech signal and joystick signal processing. The speaker independent, discrete word speech recognition system consists of two parts at the back end. They are a small vocabulary training section that constructs a model and a speech recognition section that uses this model to score input command word. On the other hand, the joystick has two input signals which detect the x and y directions and speech information. They will be converted into digital signals by the Analog-to-Digital Converter (ADC) in the microcontroller. My program in microcontroller will transform the Cartesian x-y input into Polar coordinate. According to the magnitude and angle of the converted signal, the system is able to identify which direction the user selects. The above methodology is demonstrated in Figure 3: MWA System Block Diagram.

1.4 Scope of the Project

The MWA project consists of three distinct components: the first component is a conventional joystick steering control, the second component is a speech recognition steering control, and the third component is a motor unit. Erika Schimek is responsible for the motor unit design and it can be further divided into motor driver design and programming, motors and the rechargeable battery supplies. Ling Tsou and I completed the speech recognition system for steering control together, but we each concentrated on different stages of the implementation. Ling's concentration is on the database signal interface and feature extraction. I am mainly responsible for developing the HMMs including the initialization, training and recognition, which is indicated by the red dash line box in Figure 4: Functional scheme of an Automatic Speech Recognition System .

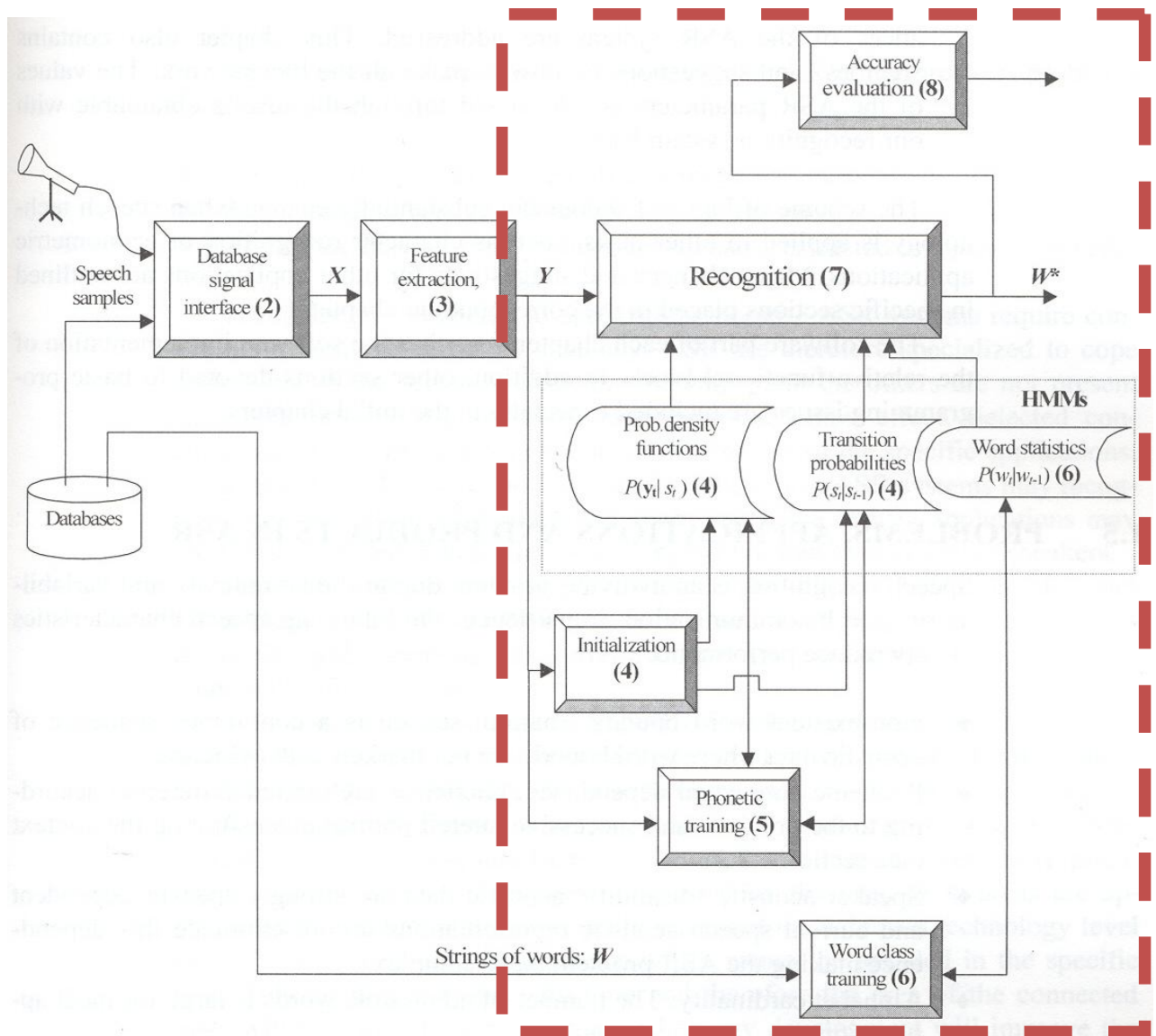


Figure 4: Functional scheme of an Automatic Speech Recognition System (6)

Based on Figure 4: Functional scheme of an Automatic Speech Recognition System above, Ling is responsible for microphone circuit design, block 2 and 3. Block 2, speech databases are essential to allow the Automatic Speech Recognition to grow in performance. The speech data are from a microphone in our design since we do not have other databases to store word samples. We also designed the vocabulary to include a small set of command word such as Go, Stop, Left, Right, Slow and Back. As for block 3, it processes the raw speech samples to extract significant speech features for recognition. The output of the feature extraction based on one command word is an

observation sequence, which will be the input for HMMs and block 7, speech recognition (6).

HMMs training estimates block 4 and block 6 probability parameters which will be discussed in more detail in the Design Procedure chapter later in the report. The Automatic Speech Recognition system in our project is dedicated for words. Therefore, the phonetic training does not apply to our design. After training the whole HMM, recognition, block 7, can be performed to obtain the more likely sequence of words \mathbf{W}^* , which is the recognized command word. This command word will be passed to Erika's part as her input to the propulsion system. The last block, block 8 is a set of procedures used to test the performances of the Automatic Speech Recognition system. It will be explained further in the Result and Discussion chapter.

In addition to building the HMM, another major focus of mine is to design and implement the joystick. The selected two axis mini-joystick takes the user selection of speed and direction and inputs this information to the microcontroller. After converting the signals from analog to digital by the built-in ADC in the microcontroller, there is an algorithm to convert the signal from Cartesian to Polar coordinates. The magnitude and angle will be extracted from the Polar coordinate. The magnitude represents the speed the user selects and the angle encodes the direction information. Finally, after a decision making algorithm, the magnitude and angle information will be mapped to the same command word used in Automatic Speech Recognition system. This way ensures the two steering control systems output a consistent command to the propulsion system. The following Figure 5: Overview of Automatic Speech Recognition and joystick steering control systems is an overview hardware implementation of both the Automatic Speech Recognition system and joystick steering control with a 9V external battery source.

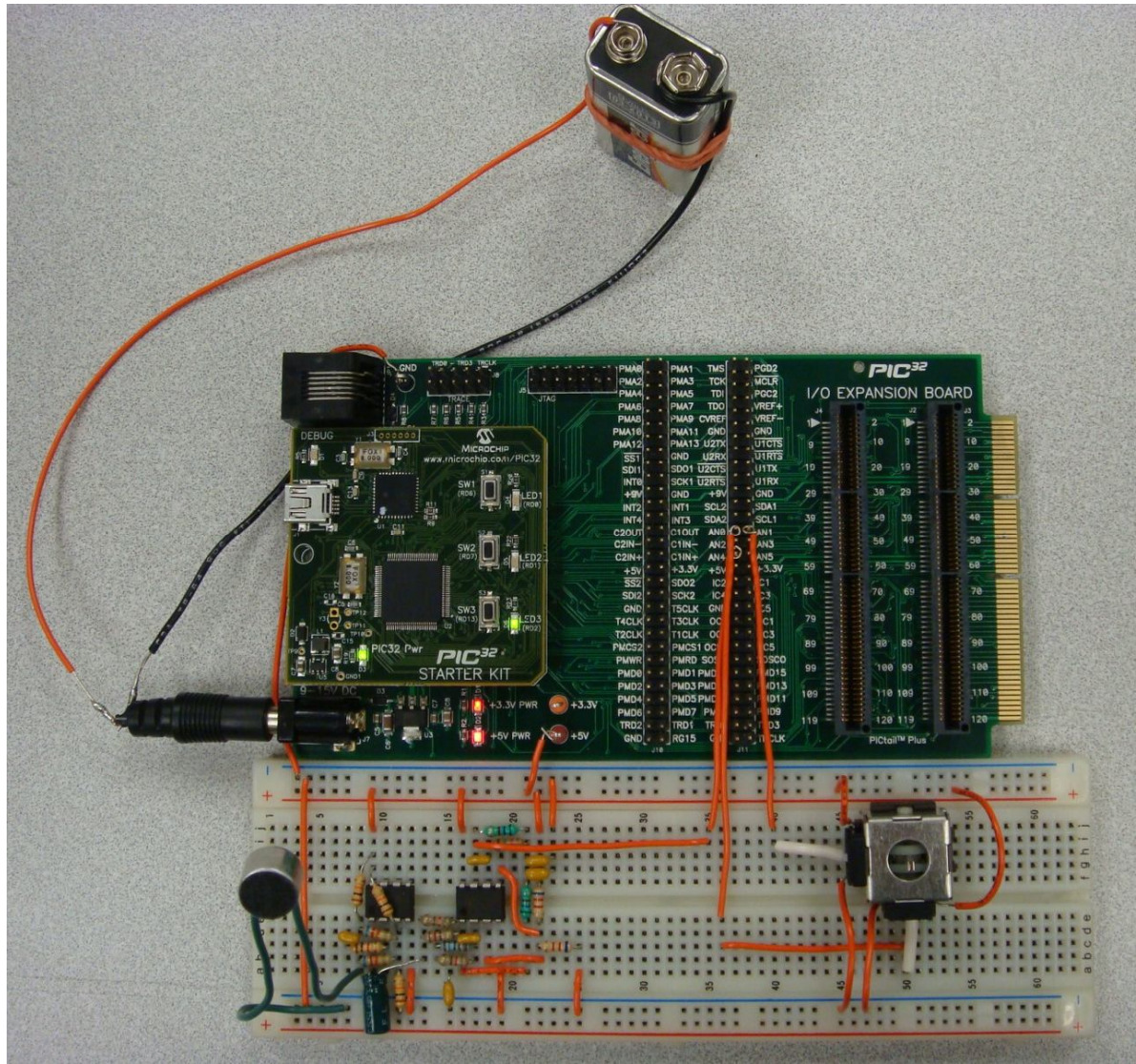


Figure 5: Overview of Automatic Speech Recognition and joystick steering control systems

2 Literature Review

2.1 Speech Recognition with HMMs

The thirty-year-old research activity in the speech recognition area has produced a well-consolidated technology that is also firmly supported theoretically (6). After reviewing several journals on its applications, we discovered that there exist different methods for speech recognition as listed below:

1. Traditional Dynamic Time Warping (DTW) (5),
2. Learning Vector Quantization (LVQ) neural network (9),
3. DTW combined with Continuous HMMs (10),
4. Hybrid HMM with Artificial Neural Networks algorithms (11).

In particular, DTW attempts to match stored patterns of a time-varying process to test templates created from the same time-varying courses (5). It requires a large amount of processing and significant storage. Another class of stochastic techniques in speech recognition is based on LVQ neural network, which has been a small part of a much more general research effort. Results of the applications of LVQ to speech research lag far behind those for HMMs because of the relative youth of the technology (7). Both LVQ and HMMs are stochastic algorithms holding considerable promise for speech recognition (12). In particular, LVQ is a vector quantizer with very powerful classification ability. HMMs, on the other hand, have the advantages that phoneme models can easily be concatenated to produce long utterance models, such as word or sentence models (12). Hence, HMMs are more suitable for our MWA steering control application since our input speech is restricted to a small set of command words. HMMs also possess a more solid foundation in theory and applications and it is the sole technique that gains the acceptance of the researchers to be the state of the art (11). The combined methods were not considered for implementing into our MWA speech recognition system as we wanted to focus on the basic algorithms. Therefore, HMMs was

selected based on its reliability compared to the other methods and its relatively efficient algorithm.

A HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters (5). The Markov Chain is a state model of a process where one can observe the state transitions. However, it is too simple to be applicable to speech because the features that can be estimated from speech, cannot detect accurately the nature of the observation (6). Therefore, the extended model, HMMs, is a doubly stochastic process, composed of a stochastic process describing the evolution of the states. The observations belong to each state are described by separate probabilistic functions (6). The challenge is to determine the hidden parameters of the probabilistic function from the observation parameters (5). HMM training is used to find the hidden parameters of the observation sequences provided by Ling's front-end process of speech recognition. The outcome of the HMMs training is the word models which will be in the other part of the HMMs recognition. The recognition part will find the likelihood of an incoming speech observation sequence (7).

2.2 HMM Recognition using Viterbi and Forward-Backward Algorithms

Between the two problems, HMM recognition is easier to solve than HMM training. This recognition method is used for the measure of likelihood of a given HMM. However, direct computation of the likelihood requires too much resource and is computationally infeasible (7). There are different algorithms available for reducing the search complexity. "Any path" (aka Forward-Backward Algorithm) and "Best path" (aka Viterbi Algorithm) are the two most widely used methods to perform HMM recognition. The reason for the name "any path" method is that the likelihood to be computed here is based on the probability that the observations could have been produced using any state sequence (path) through the model. The "Best path" insists that the likelihood should be

based on the best state sequence (7). Since the Viterbi approach is slightly less expensive to compute and it is functionally equivalent to the Forward-Backward Algorithm, it is selected to be used in the Automatic Speech Recognition system (7).

2.3 HMM Training using Viterbi and Forward-Backward Algorithm

The HMM training is very crucial for most applications of HMMs, since it allows us to optimally adapt the hidden model parameters to observed training data (13). Hence, it is extremely important for us to select the right method to perform HMM training. There are two commonly used solutions for HMM training, Viterbi Re-estimation and Forward-Backward Re-estimation. Similar to the HMM recognition, Viterbi Re-estimation start with random HMM parameters and uses Viterbi Algorithm to find the most probable path for each training sequence. Then it labels the sequence with that path. On the other hand, the Forward-Backward Re-estimation sums over all possible paths, rather than the single most probable one, to estimate expected probabilities (13).

Although the Forward-Backward algorithm is the most popular algorithm for training the discrete observation HMMs, Viterbi is a simpler and equally effective algorithm which yields comparable performance. Hence, Viterbi was initially chosen for HMM training estimations. However, in the design phase, its training results were found out to be non-satisfactory. Hence, HMM training using Forward-Backward Algorithm was also implemented later to produce more functional results. More detail can be found in the Experimental and Design Procedures Chapter.

2.4 Microcontroller Selection

A small vocabulary, speaker independent, discrete word speech recognition system based on the System on Chip philosophy has been implemented on an 8-bit microcontroller (10). Their system adopts HMMs to develop the speech model and the recognition accuracy is nearly 97% with a vocabulary up to 30 phrases under normal conditions. Also, speech recognition tools for human-machine interaction in consumer equipments have become popular because of the development of high performance microcontroller devices at lower cost (14). Similarly, a speech recognition system with LVQ have developed onto a Mitsubishi M16C, a low-cost 16 bit microcontroller (about 5 euros), trained on a real environment (20dB of signal to noise ratio). Moreover, a single-chip speech recognition system has been implemented based on 8051 microcontroller core, which is composed of an 8-bit core, 512 bytes on-chip RAM, 8K bytes on-chip ROM. (15).

Besides the above literature research, we also know that the HMM solution requires heavy computations. A fast clock speed and a large amount of memory for a microcontroller system are required. At last, a Microchip 32 bits, 8 MHz core microcontroller starter kit with 32 Kbyte SRAM and 512 Kbyte FLASH memory is selected according to its cost and performance. It also contains the pulse width modulation feature, which is essential for motor implementation. Figure 6: PIC32 Starter Kit is the PIC32 starter kit.

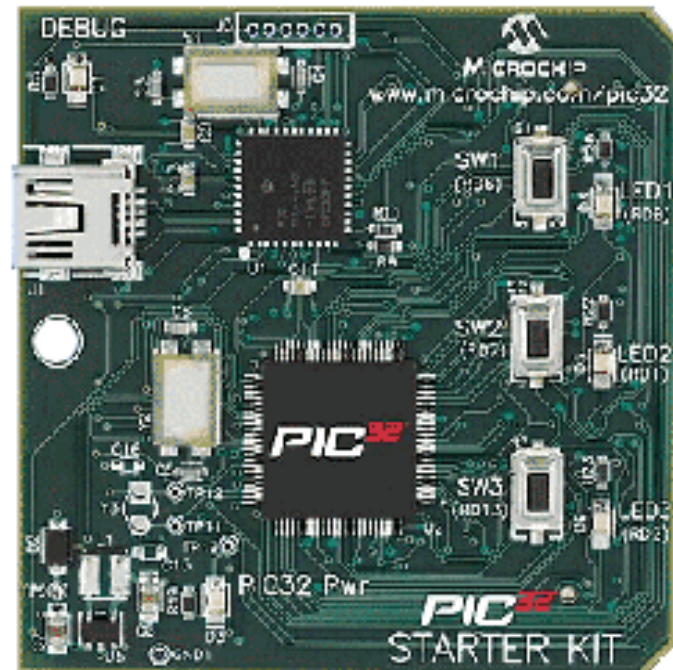


Figure 6: PIC32 Starter Kit (16)

3 Statement of Problem and Methodology of Solution

The overall problem of the MWA project is to convert a manual wheel chair into an electrical one with two types of steering controls, which are Automatic Speech Recognition system and joystick. My main focus of the problem is to implement the back end portion of the Automatic Speech Recognition system as well as to realize the joystick steering control.

3.1 HMMs in Automatic Speech Recognition system

In particular, the input of the back end of Automatic Speech Recognition system is an observation sequence (in recognition mode) or observation sequences (in training mode), which extract features from the speech signals. How to convert those observation sequences into a meaningful output signal that can be output to the propulsion system is the fundamental problem. The method chosen to solve this problem through comparing different literature is HMMs. (Please refer to the Literature Review for more detail as to the reason why HMMs is selected.) In fact, the back end of Automatic Speech Recognition system takes in the observation input, further processes it through HMMs and determines the most likely command word spoken by the user. The output command word will be passed to the propulsion system to produce wheel chair motions. The challenge of my tasks is essentially to design HMMs is to effectively represent the speech inputs and to recognize them.

Before going into the nuts and bolts of my actual design, more background of the theory of HMMs should be presented. Hidden Markov Models (HMMs) are inspired from the more than 90 years old mathematical model known as Markov Chain (11). A HMM is a statistical model in which the system being modeled is assumed to be a Markov process

with an underlying stochastic process that is not observed (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations (13).

3.1.1 Discrete Markov Processes

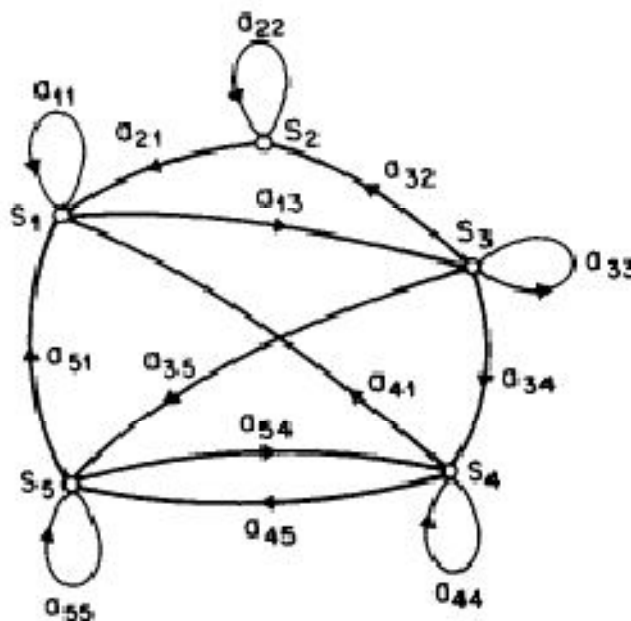


Figure 7: A Markov chain with 5 states (labeled S1 to S5) with selected state transitions (13)

The following system is in one of a set of N distinct states at any time, S_1, S_2, \dots, S_n , as demonstrated in Figure 7: A Markov chain with 5 states (labeled S1 to S5) with selected state transitions, where $N = 5$ for simplicity. At regularly spaced discrete time, the system undergoes a change of state or possibly stays at the same state depending on a set of probabilities associated with the state. The time instants linked to the state changes are denoted as $t = 1, 2, \dots$, and the actual state at time t is q_t (13). In general, a full probabilistic description of this n th-order system requires specification of only on the preceding n states (17). For speech recognition, the value of n is typically one as it is a

first-order Markov Chain system. Therefore, the probabilistic description for this first-order Markov Chain is truncated to just the current and the predecessor state (13):

$$\begin{aligned} P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots] \\ = P[q_t = S_j | q_{t-1} = S_i]. \end{aligned}$$

Furthermore, we only consider those processes in the above formula to be independent of time. Hence, the state transition probabilities a_{ij} becomes of the form (13):

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \leq i, j \leq N$$

With the state transition coefficients having the properties since they obey standard stochastic constraints (13):

$$\begin{aligned} a_{ij} &\geq 0 \\ \sum_{j=1}^N a_{ij} &= 1 \end{aligned}$$

The above stochastic process could be called an observable Markov model since the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event. To illustrate this idea with a concrete example, let us consider a simple 3-state Markov model of the weather. We assume that once a day (e.g., at noon), the weather is observed as being one of the following (13):

State 1: rainy

State 2: cloudy

State 3: sunny.

From the history of the weather of a town under investigation, we have the following table (Table 1: Weather Expectation Probabilities) of probabilities of having certain state of tomorrow's weather and being in certain condition today (11). Equivalently, those probabilities can be represented in state transition probabilities matrix A as:

Table 1: Weather Expectation Probabilities

Today	Tomorrow		
	Rainy (1)	Cloudy (2)	Sunny (3)
Rainy (1)	0.4	0.3	0.3
Cloudy (2)	0.2	0.6	0.2
Sunny (3)	0.1	0.1	0.8

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Given that the weather on day 1 ($t = 1$) is sunny (state 3), we can calculate for the following question: What is the probability according to the above model that the weather for next three days to be “sun-cloudy-rain”? To state more formally, we can define the observation sequence O as $O = \{S3, S2, S1\}$ corresponding to $t = 1, 2, 3$ and we wish to determine the probability of O , given the model. The probability can be expressed as:

$$\begin{aligned} P(O|Model) &= P[S3, S2, S1|Model] = P[S3] * P[S2|S3] * P[S1|S2] = \pi_3 * a_{33} * a_{32} * a_{21} \\ &= 1 * 0.8 * 0.1 * 0.2 = 0.016 \end{aligned}$$

where we use the notation

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N$$

to denote the initial state probabilities.

3.1.2 Hidden Markov Models (HMM)

In the particular problem presented in the previous section, the states were observable and they represented the weather conditions (sunny, cloudy and raining) as the Markov Chain is a state model of a process where one can observe the state transitions. However, this kind of model formulation is very limited due to the need of observable state sequence

which is unknown in most problems (11). For instance, it is too simple to be applicable to speech recognition because the features that can be estimated from speech, cannot detect accurately the nature of the observation (6). Therefore, it is extended to a HMM model, a doubly stochastic process, composed of a stochastic process describing the evolution of the states. The observations belong to each state are described by separate probabilistic functions (6). More precisely, the HMM is a probabilistic pattern matching technique in which the observations are considered to be the output of stochastic process and consists of an underlying Markov chain (11). The challenge is to determine the hidden parameters of the probabilistic function from the observation parameters.

3.1.2.1 A HMM Example

To help better understand the HMM, the Urns and Candies Model inspired by (13) will be shown in Figure 8: An N-state urn and candy model which illustrates the general case of a discrete symbol HMM. Assume that there are N large urns in a room and there are a large number of colored candies in each urn. The physical process for obtaining observation is as follows. Someone is in the room and based on some random process, he/she chooses an initial urn. From this urn, a candy is chosen at random, and its color is recorded as the observation. The candy is then put back in the urn which it was selected. Then a new urn is selected according to the random selection process associated with the current urn, and the candy selection process is repeated. The entire process generates a finite observation sequence of colors, which it is modeled as the observable output of an HMM.

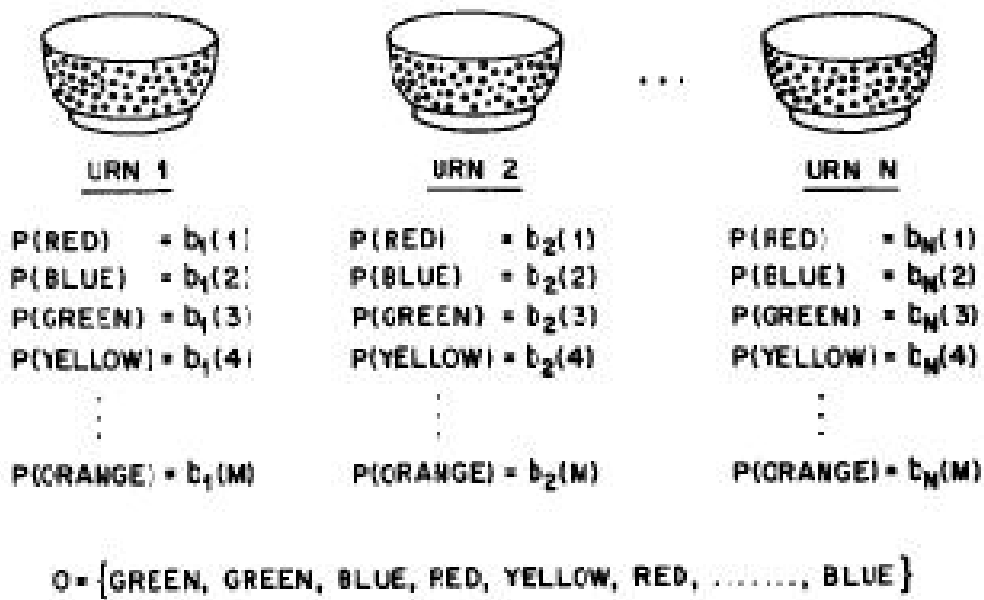


Figure 8: An N -state urn and candy model which illustrates the general case of a discrete symbol HMM (13)

The simplest HMM that corresponds to the urn and candy process is one in which each state corresponds to a specific urn, and for which a candy color probability is defined for each state. The choice of urns is dictated by the state transition matrix of the HMM (13).

3.1.2.2 Elements of an HMM

The above section should provide a clear idea as to what an HMM is and some simple applications. This section will be dedicated to formally define the elements of an HMM.

1. N , the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or sets of states of the model. For example, in the urn and candy model, the states corresponded to the urns. Generally the states are interconnected in such a way that any state can be reached from any other state, such as an Ergodic model (13). However, it is not the case for speech recognition. Due to the nature of the

speech signals, the state is only allowed to increase its index and the number of steps a state can skip to reach the next step is also limited depending on the applications. This is also known as Left-Right Model and more information will be discussed in the HMM Constraints for Speech Recognition Systems section.

2. M , the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. For the candy and urn model they were the colors of the balls selected from the urns. We denote the individual symbols as $V = \{v_1, v_2, \dots, v_m\}$ (13).
3. The state transition probability distribution $A = \{a_{ij}\}$ is the probability represents the random process going from one urn to another in the previous Urn and Candy example. It is also the same as the weather transition probability distribution in Markov Models where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N.$$

There are several types of common HMM models expressed by different transition probabilities. For the case where any state can reach any other state in a single step, also known as Ergodic process, we have $a_{ij} > 0$ for all i, j . For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs (13).

4. The observation symbol probability distribution in state j , $B = \{b_j(k)\}$, is a probabilistic matrix showing the percentage likelihood to pick certain colors of candies at a particular urn in our example. The colors of the candies are our observation symbols where (13)

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad \begin{matrix} 1 \leq j \leq N \\ 1 \leq k \leq M. \end{matrix}$$

5. The initial state distribution $\pi = \{\pi_i\}$, where(13).

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N.$$

This parameter is randomly selected in the example. However, it is meaningful in reducing the computational time. (Please refer to Design Procedures Chapter for more clarifications.)

In conclusion, a complete definition of an HMM requires specification of two model parameters (N and M), specification of observation symbols which is input to HMMs in our speech recognition system, and the specification of the three probability measures A, B and π . Therefore, a complete parameter set of the model can be represented as (13):

$$\lambda = (A, B, \pi)$$

It might not be very obvious how those HMM parameters related to the speech signal modeling, but this could be postulated by looking at the speech production mechanism. Speech is produced by the slow movements of the articulatory organ. The speech articulators taking up a sequence of different positions and consequently producing the stream of sounds that form the speech signal. Each articulatory position could be represented by a state of different and varying duration. Accordingly, the transition between different articulatory positions (states) can be represented by $A = \{a_{ij}\}$. The observations in this case are the sounds produced in each position and due to the variations in the evolution of each sound this can be also represented by a probabilistic function $B = \{b_j(k)\}$ (11). More background information about how the sound or raw speech signal looks like can be found in Appendix A: Raw Speech Signal Samples within a linear window.

The correspondence between the model parameters and what they represent in the speech signal is not unique and could be viewed differently. The important thing is to picture the physical meanings of the states and observations in each view (11).

3.1.2.3 HMM Constraints for Speech Recognition Systems

As we briefly mentioned previously, HMM could have different constraints depending on the nature of the problem. As for speech recognition, the main constraints needed are concluded as follows:

1. First Order Markov Chain :

This constraint assumes the probability of transition to a state only depend on the current state (11).

$$\begin{aligned} P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots] \\ = P[q_t = S_j | q_{t-1} = S_i]. \end{aligned}$$

2. Stationary States' Transition:

This assumption testifies that the state transitions are time independent, and we will have (11):

$$a_{ij} = P(q_{t-1}=S_j / q_t=S_i) \text{ for all } t$$

3. Observations Independence:

This hypothesis illustrates that the observations come out within certain state depend only on the underlying Markov chain of the states, without considering the effect of the occurrence of the other observations. Although this assumption is a poor one and deviates from reality, it works fine in modeling speech signal (13). This assumption implies that:

$$P(O_t/O_{t-1}, O_{t-2}, \dots, O_{t-p}, q_t, q_{t-1}, q_{t-2}, \dots, q_{t-p}) = P(O_t/ q_t, q_{t-1}, q_{t-2}, \dots, q_{t-p})$$

4. Left-Right topology constraint:

Speech signals have been found to contain better observed properties of the signal being modeled as Left-Right type of HMM than the standard Ergodic model. The corresponding Left-Right state sequence associated with the model has the property that as time increases the state index increases (or stays the same), which means that

the states proceed from left to right. Clearly, the Left-Right type of HMM has the desirable property that it can readily model speech signals whose properties change over time (13). The fundamental property of all Left-Right HMM is that the state transition coefficients have the property:

$$a_{ij} = 0, \quad j < i$$

No transitions are allowed to states whose indices are lower than the current state. Furthermore, the initial state probabilities have the property (13):

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases}$$

Since the state sequence must begin in state 1 and end in state N , additional constraints are placed on the state transition coefficients to make sure that large changes in state indices do not occur with left-right models. Therefore, a constraint of the form (13):

$$a_{ij} = 0, \quad j > i + \Delta$$

In our speech recognition HMM implementation, the value of Δ equal to 1 is used. It implies that no jump of more than 1 state is permitted. Hence, the current state can only transit to a higher state or remain in the same state. The following Figure 9: 4-state Left-Right HMM with no skip transitions is equivalent to the state transition probability matrix. Zeros in the state transition matrix represent illegal state transitions.

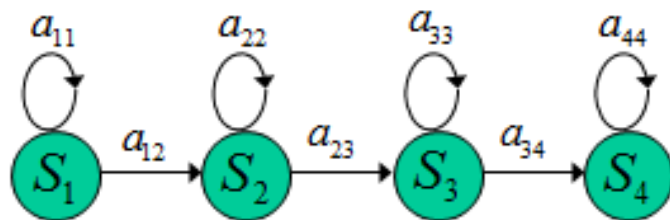


Figure 9: 4-state Left-Right HMM with no skip transitions

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

5. Probability Constraints:

The following constraints come from the fact that we are dealing with probabilities.

$$\sum_{j=1}^N \pi_j = 1$$

$$\sum_{j=1}^N \pi_j = 1$$

$$\int_{\mathcal{O}} b_i(\mathcal{O}_t) d\mathcal{O} = 1$$

If the observations are discrete then the last integration will be a summation.

3.1.2.4 The Three Basic Problems for HMMs

The general background of HMM has been explained in detail in the previous sections. Given the form of HMM, there are three basic problems of interest that must be solved for the model to be useful in real-world speech recognition applications. (13) These problems are directly referenced from (13) as following:

- Problem 1:** Given the observation sequence $O = O_1 O_2 \cdots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?
- Problem 2:** Given the observation sequence $O = O_1 O_2 \cdots O_T$, and the model λ , how do we choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal in some meaningful sense (i.e., best “explains” the observations)?
- Problem 3:** How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

To take those problems further, they will be analyzed on each of their purposes before discussing their solutions.

Problem 1 is essentially the HMM recognition problem. Given a model and a sequence of observations, how can we compute the probability that the observed sequence was produced by the model? One can also view the problem as the scoring of a given model matches with a given observation (13). The latter point of view can be extremely useful in the case which we are trying to choose among several competing models. The solution to Problem 1 allows us to select the best model among many that might have produced the given observation input (13).

Problem 2 is a decoding procedure to uncover the hidden part of the model, or to find the “correct” state sequence. This part is used in the training problem to study the behavior of each state from different aspects, such as states’ duration or spectral characteristics of each state. (11). However, there is no “correct” state sequence in reality because they are hidden from us and we can only estimate their behaviors. Hence, it becomes an optimization problem to find the best state sequence. There are several reasonable optimality criteria that were imposed to choose for the uncovered state sequence (13).

Problem 3 is the HMM training procedure to optimize the model parameters to obtain the best model that represents certain set of observations belonging to one spoken entity (11). This training problem is crucial for speech recognition because it allows us to create the best models for the speech input (13).

In particular to the solution of our MWA speech recognition system, for each of the six command words in our small vocabulary, we have a training sequence consisting of a number of repetitions of observation sequences of the word. Our goal is to build individual word models for the six command words in the small dictionary. This task is done by using the solution to Problem 3 to optimally estimate model parameters for each word model (13). To develop an understanding of the physical meaning of the model states, we use the solution to Problem 2 to segment each of the word training sequences into states, and then study the properties of the observations occurring in each state. The goal here would be to make refinements on the model so as to improve its capability of modeling the spoken word sequences (13). Finally, once the set of six HMMs has been designed and optimized, recognition of an unknown word is performed using the solution to Problem 1 to score each word model based upon the given test observation sequence, and select the word whose model score is highest.

Therefore, it should be clear that the solutions of the above three problems combined will complete a HMM, which is, in fact, the resolution for the speech recognition system. We will finally discuss the designs based on formal mathematical solution to each of the three fundamental problems for HMMs in the Experimental and Design Procedures chapter.

3.2 Joystick Steering Control

Let us review the alternative joystick steering control method of the MWA system. As the only control option other than speech recognition, the complexity of implementing the joystick control is much simpler than HMMs in speech recognition system. Please see Figure 10: Joystick Implementation below for the hardware implementation.

The mini-joystick with two channel inputs takes in x and y direction and speed information and encodes them as voltage variations. Those two channels of voltage signals get converted to digital by the Analog-to-Digital Converter built into the microcontroller. There are design algorithms in the micro-controller to extract the speed and direction information from the joystick input. Please refer to the Experimental and Design Procedures chapter for detail algorithms. Finally, the speed and direction will be mapped to the command words to output to the propulsion system.

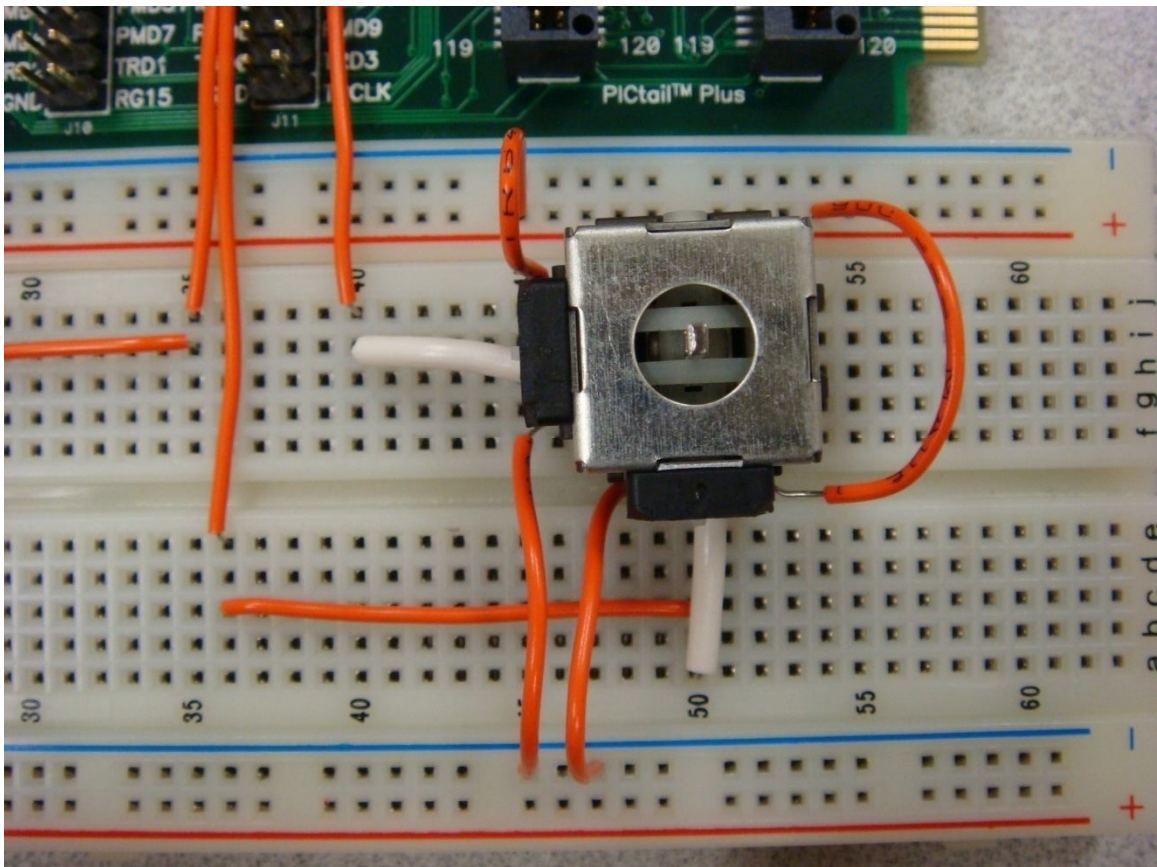


Figure 10: Joystick Implementation

4 Experimental or Design Procedures

4.1 Top-Level Automatic Speech Recognition Design

As mentioned previously, the solutions of the three problems proposed in the Statement of Problem and Methodology of Solution chapter are the major components that build up an HMM. In fact, HMM is our answer to implement the back-end of the Automatic Speech Recognition system. Problem 1 is essentially the HMM recognition problem and solutions to Problem 2 and 3 together can solve the HMM training task. They will be divided into several sub-sections in the rest of this chapter and detailed algorithms will be present.

First, an overview of the top level of the back-end of the speech recognition system design and the major parameters will be provided. A set of six command words is selected to be the users' vocabulary for controlling the MWA. They are "Go, Stop, Left, Right, Slow and Back". Therefore, this small vocabulary with isolated words enables us to use a discrete HMM instead of a continuous HMM. Since we assume that the primary user of the MWA is fixed usually, a speaker-independent HMM is not necessary in this case. It is also less complex to implement a speaker-dependent HMM as less training data are needed for building the HMM model. Also, a Left to Right topology is adapted for the HMM due to the nature of the speech signals as discussed beforehand.

As for selections of some major parameters of HMM, research was performed on the discrete word model. The reason why we select a word-based HMM instead of a phoneme-based one is that our six command words are all relatively short and within 2-3 phonemes. N , the number of states in the model, is chosen to be 5 after consulting with different sources (18). M , the number of distinct observation symbols per state, is dependent on the size of the codebook which Ling implemented. Due to the nature of the small vocabulary and short command words, M is determined to be 128.

The solutions of the HMM training and recognition are extremely computationally heavy in nature, which will be discussed more in the remainder of this chapter. Therefore, there exist various algorithms to help reduce the complexity of the computations. The most common ones including Viterbi Algorithm and Forward-Backward Algorithm, which is also known as Baum-Welch method or equivalently the EM (expectation-modification) method (13). Viterbi Algorithm was first selected to perform both HMM recognition and training due to its efficient algorithm. Unfortunately, the training result could not yield a satisfactory HMM model to pass on to recognition for identifying the command word. Therefore, Forward-Backward Algorithm was also implemented and it demonstrated a better result than the Viterbi Algorithm. The output models are used in HMM recognition and the system is able to identify the command word that is spoken by the user.

4.1.1 HMM Recognition with Viterbi Algorithm

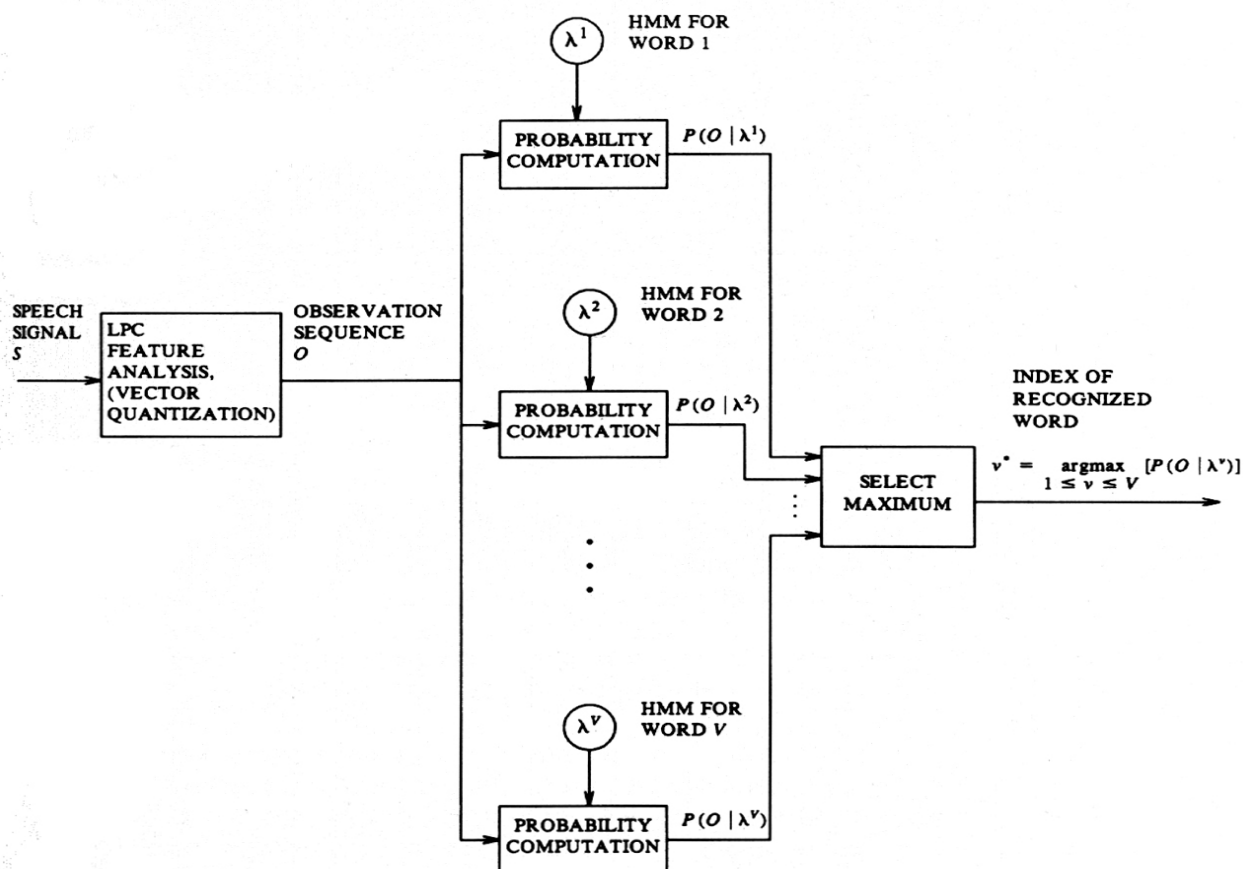


Figure 11: Block diagram of an isolated word HMM recognizer (13)

This part corresponds to Problem 1, which is to compute the most probable sequence of states (according to the HMMs that are trained) for a given sequence of outputs. Please refer to Figure 11: Block diagram of an isolated word HMM recognizer for the overview of HMM recognition process. Since it is a simpler problem than Problem 2 and 3, HMM training, we will first discuss its solution using the formal notations developed in the Statement of Problem and Methodology of Solution chapter.

In order to calculate the probability of the observation sequence $P(O|\lambda)$, $O = O_1 O_2 \dots O_T$, given the model λ . The most straightforward way is to go through every possible state

sequence of length T , the number of observations. Consider one such fixed state sequence (13):

$$Q = q_1 q_2 \dots q_T$$

where q_1 is the initial state. The probability of the observation sequence O for the state sequence above is (13):

$$P(O|Q, \lambda) = \prod_{t=1}^T P(O_t|q_t, \lambda)$$

where we have assumed statistical independence of observations. Therefore (13),

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdot \dots \cdot b_{q_T}(O_T).$$

The probability of such a state sequence Q can be written as (13):

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}.$$

For the joint probability of O and Q , the probability that O and Q occur simultaneously, is simply the product of the above two terms (13):

$$P(O, Q|\lambda) = P(O|Q, \lambda) P(Q, \lambda).$$

Therefore, the probability of O given the model λ is obtained by summing this joint probability over all possible state sequences q giving (13):

$$\begin{aligned} P(O|\lambda) &= \sum_{\text{all } Q} P(O|Q, \lambda) P(Q|\lambda) \\ &= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \\ &\quad \dots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned}$$

The following is how to interpret computations in the above equations. Initially (at time $t = 1$), we are in state q_1 with probability π_{q_1} , and generate the symbol O_1 (in this state) with probability $b_{q_1}(O_1)$. The clock changes from time t to $t + 1$ ($t = 2$) and we make a transition to state q_2 , from state q_1 with probability $a_{q_1q_2}$, and generate symbol O_2 with probability $b_{q_2}(O_2)$. This process continues in this manner until we make the last transition (at time T) from state q_{T-1} to state q_T with probability $a_{q_{T-1}q_T}$ and generate symbol O_T with probability $b_{q_T}(O_T)$ (13).

According to the previous formula, the number of calculations involved is on the order of $2T * N^T$. Since at every t , there are N possible states sequences. For our speech recognition HMM purposes (13), $N = 5$ (states), $T = 128$ (observations), there are on the order of $2 * 128 * 5^{128} \approx 10^{93}$. This calculation is computationally infeasible. Moreover, the total number of possibilities increases exponentially with the increasing number of states and observation instances. The Left-Right topology substantially reduces the number of possible paths over the full connection topology where Ergodic models in which every state could be reached from any other state at any instant (11).

To further reduce the computational cost, Viterbi Algorithm (see Figure 12: Viterbi Algorithm) was selected to achieve this objective. This technique greatly reduces the computational cost with simple iterative mathematical formulas (11).

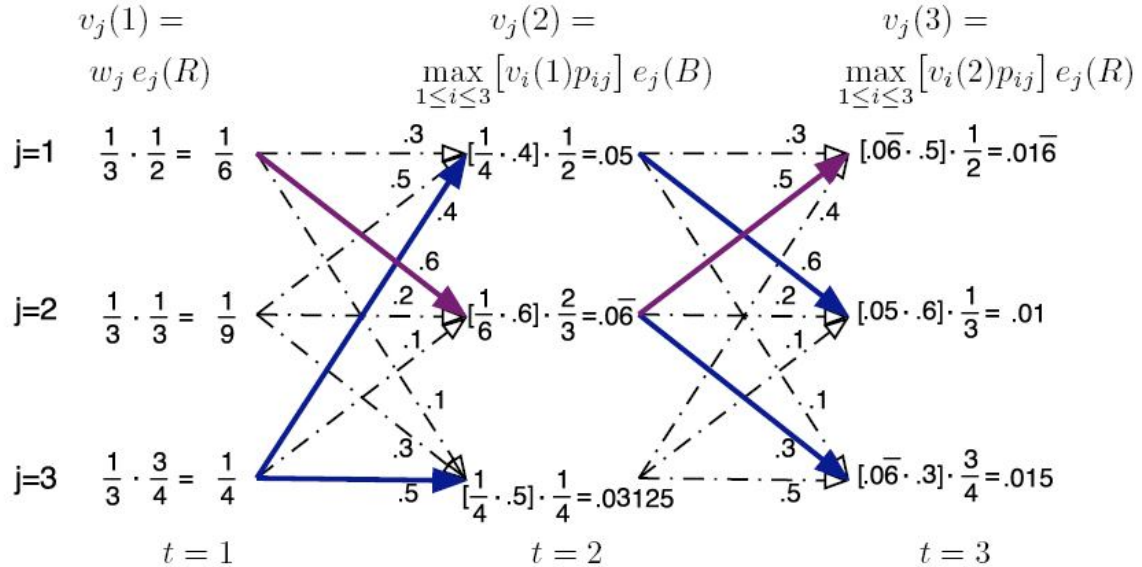


Figure 12: Viterbi Algorithm (19)

Viterbi Algorithm estimates the absolute best path by computing each local maximum (19). According to Figure 12: Viterbi Algorithm above, the probabilities going from the current state to the next state is calculated and listed below v_j of each step. The highest probability from each step will be recorded. At last, the best path will be the combination of each path with highest probability. To find the single best state sequence, $Q = \{q_1 q_2 \dots q_T\}$, for the given observation sequence $O = \{O_1 O_2 \dots O_T\}$, we need to define the quantity (13):

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \lambda]$$

If $\delta_t(i)$ is the best score (highest probability) along a single path, at time t , which accounts for the first t observations and ends in state S_i , we can induce (13):

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}).$$

The complete procedure used in the speech recognition implementation for finding the best state sequence is stated as follows:

1) Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

$$\psi_1(i) = 0.$$

2) Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T$$

$$1 \leq j \leq N$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T$$

$$1 \leq j \leq N.$$

3) Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)].$$

4) Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1.$$

The above step procedures for implementing Viterbi Algorithm are adopted from (13). In fact, this is also the solution to our Problem 2 mentioned previously, since it reveals the state sequence based on the input observation sequence and the model. Therefore, this Viterbi Algorithm can be in both HMM training and recognition.

The Viterbi algorithm involves multiplying many probabilities together. Since each of these numbers is smaller than one, possibly much smaller, the numbers in the process can become tiny enough to be indistinguishable from zero by a real computer. To avoid this problem, all the above calculations are implemented using log probabilities in the HMM speech recognition system. In other words, every value is its logarithm instead of its actual value. For instance, rather than storing probabilities p and q , $\log(p)$ and $\log(q)$ are saved. If the product pq of p and q is needed to be computed, the following rule is used:

$$\log(pq) = \log(p) + \log(q).$$

This HMM recognition with Viterbi Algorithm part of the speech recognition system is fully automated in the micro-controller. Therefore, this part fully functions in real time and the algorithm is efficient in order to be implemented into the micro-controller.

4.1.2 HMM Training

HMM Training part is the solution to Problem 3. It is also by far the most difficult problem of HMMs, which determines a method to adjust the model parameter $(A, B, \boldsymbol{\pi})$ to maximize the probability of the observation sequence given the model (13). There is no known way to analytically solve for the model which maximizes the probability of the observation sequence (13). In fact, there is no optimal way of estimating the model parameters, but we can choose $\lambda = (A, B, \boldsymbol{\pi})$ such that $P(O|\lambda)$ is locally maximized using an iterative procedure such as Forward-Backward or Viterbi Re-estimation. Figure 13: Optimum Search Possibilities shows that whether the HMM training converges at global maxima or local maxima depends on the initial guess of the model $\boldsymbol{\pi}$. Unfortunately, the current procedures cannot guarantee an globally maximal possibility.

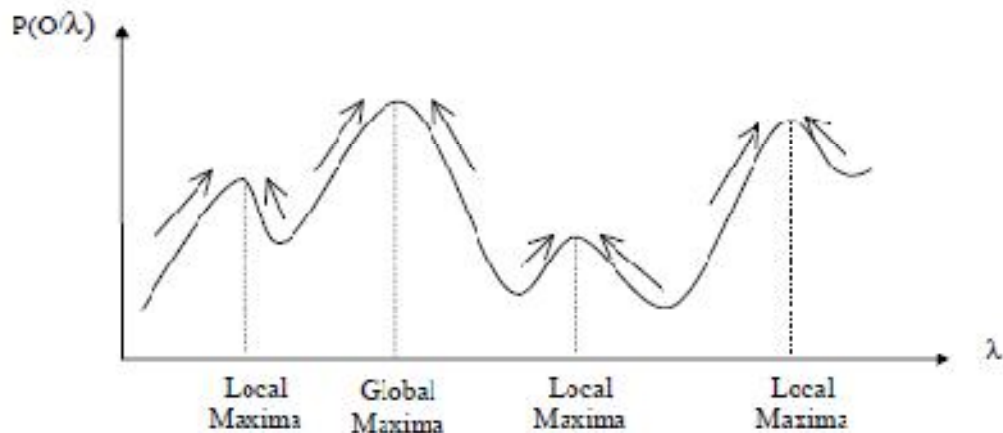


Figure 13: Optimum Search Possibilities

Besides the two algorithms selected for performing the HMM training, the actual design for our set-up is demonstrated in Figure 14: Block Diagram for HMM training. Based on

our research, 20 samples is collected to train each of the 6 command words. Each sample passes Ling's front-end of Automatic Speech Recognition system and outputs one observation sequence. For each word, 20 observation sequences will be passed into the HMM training to develop one HMM for the command word. In our design, the initial state π is arbitrarily picked to be the initial state due to the Left-Right Topology as mentioned before. Therefore, the complete HMM model $\lambda = (A, B, \pi)$ becomes $\lambda = (A, B)$.

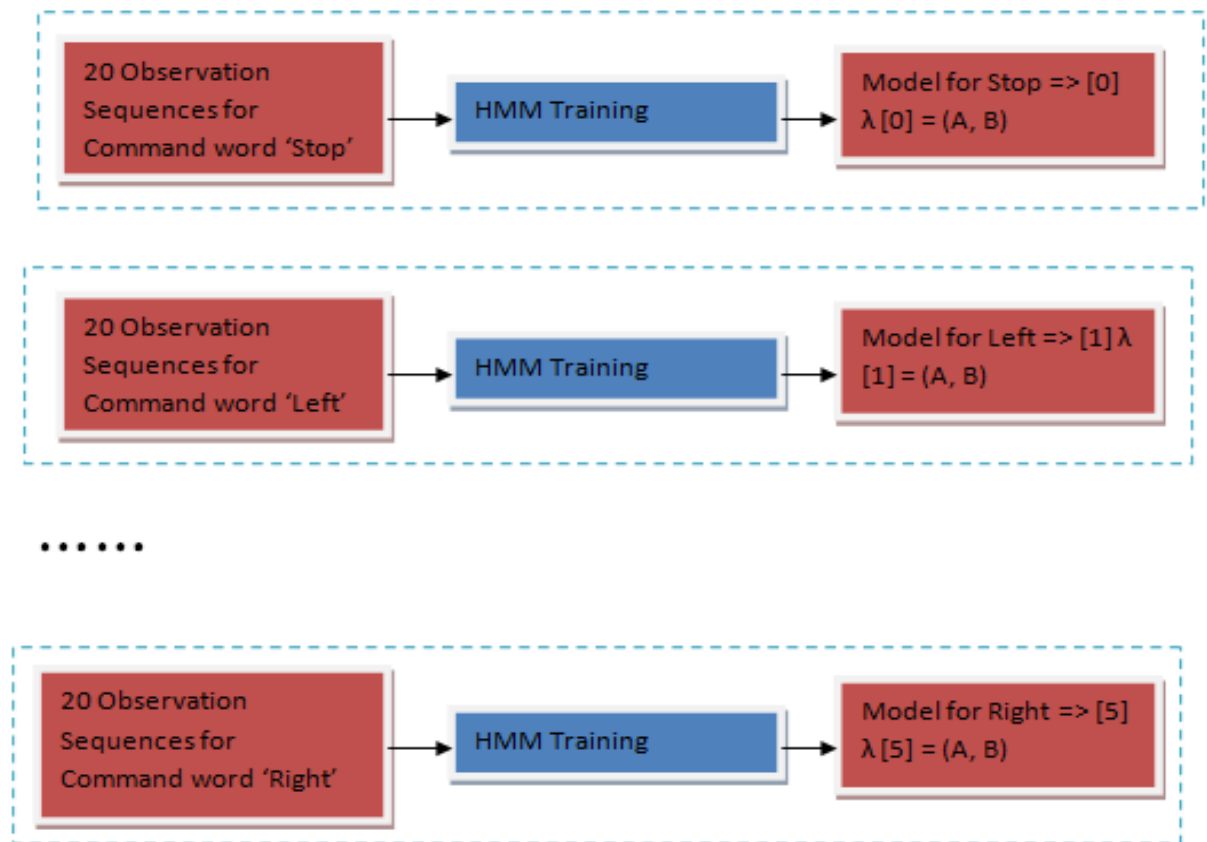


Figure 14: Block Diagram for HMM training

The two common algorithms for optimizing the model parameters will be discussed in the two sub-sections following. Due to the computational complexity of the HMM training problem and the memory limitation of our micro-controller, this portion of the Automatic Speech Recognition system is performed in a computer. Another important issue that was encountered in the Design process is that Viterbi Re-estimation which was

initially selected did not yield a satisfactory result. More information and analysis will be explained in the Viterbi Re-estimation section following. As a result, Forward-Backward Re-estimation is implemented and its output HMM model produces convincing speech recognition results.

4.1.2.1 Viterbi Re-estimation

Although the Forward-Back algorithm is the most popular algorithm for training the discrete observation HMM, a simpler but slightly less effective algorithm was selected to implement HMM training initially. This algorithm is based on the Viterbi decoding approach to recognition (7).

The following Figure 15: Trellis Illustration of Viterbi helps to explain the whole flow of the Viterbi training process. The state at each instant is represented by a small circle, and the arrows represent the state transition. It is a 10 state sequence with length equal to 14. It is equivalent to the A matrix below.

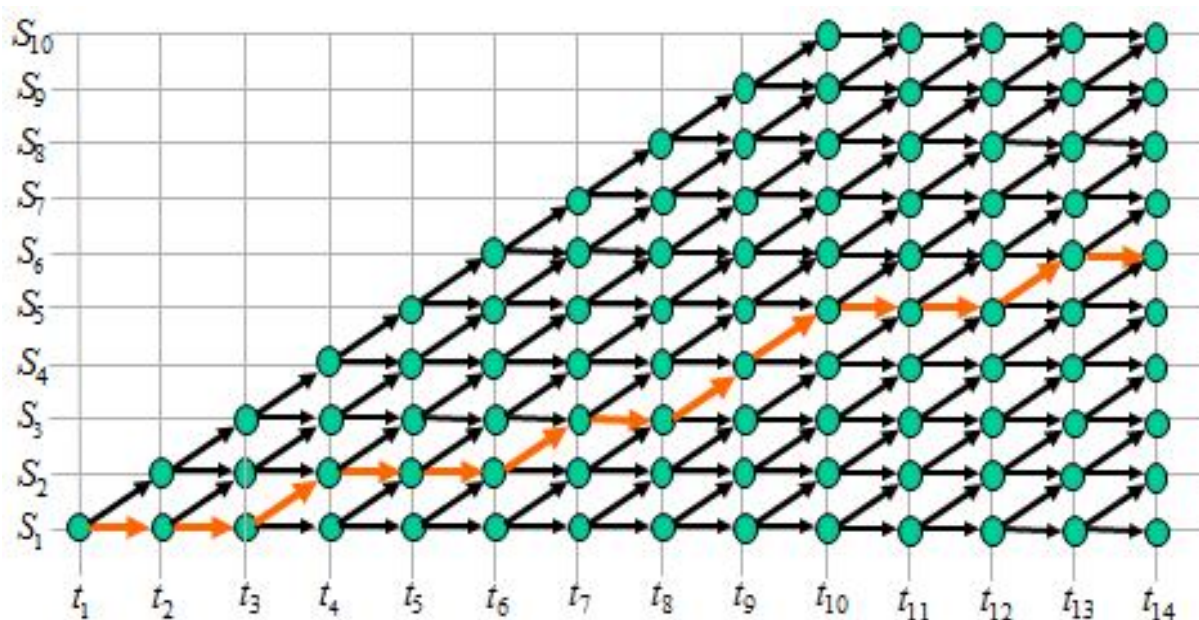


Figure 15: Trellis Illustration of Viterbi (18)

$$\mathbf{A} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Based on the Viterbi Algorithm explained previously in the Statement of Problem and Methodology of Solution chapter, the procedures for the implementation are as follows:

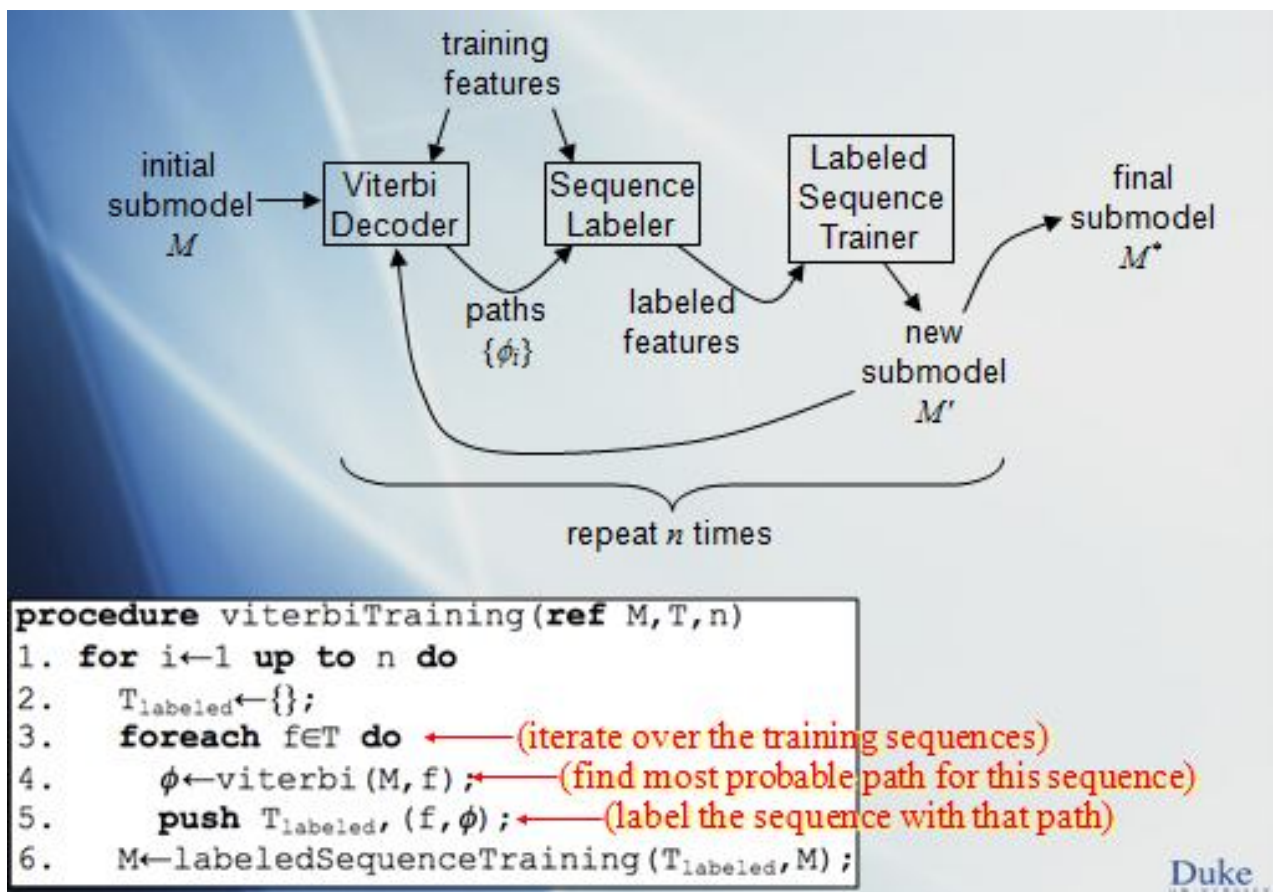


Figure 16: Viterbi Training Algorithm (20)

The Viterbi Re-estimation of HMM training was implemented as Figure 16: Viterbi Training Algorithm . After passing the HMMs developed by Viterbi training to the HMM recognition, the system results in a very low recognition rate. In order to improve the system performance, the most popular HMM training method: Forward-Backward Algorithm was also implemented. The Forward-Backward method yields significantly better HMMs results in a higher recognition rate.

4.1.2.2 Forward-Backward Re-estimation

Define $F(i,k)$ to represent the probability $P(x_0...x_{k-1}, q_i)$ that the machine emits the subsequence $x_0...x_{k-1}$ by any path ending in state q_i —i.e., so that symbol x_{k-1} is emitted by state q_i . Equivalently (20),

$$F(i,k) = \begin{cases} 1 & \text{for } k = 0, i = 0 \\ 0 & \text{for } k > 0, i = 0 \\ 0 & \text{for } k = 0, i > 0 \\ \sum_{j=0}^{|Q|-1} F(j,k-1)P_t(q_i | q_j)P_e(x_{k-1} | q_i) & \text{for } 1 \leq k \leq |S|, \\ & 1 \leq i < |Q| \end{cases}$$

Therefore, the probability of a state sequence S given a model M can be computed:

$$P(S | M) = \sum_{i=0}^{|Q|-1} F(i, |S|)P_t(q_0 | q_i)$$

Similarly, define $B(i,k)$ to be the probability that the machine M will emit the subsequence $x_k...x_{L-1}$ and then terminate, given that M is currently in state q_i (which has already emitted x_{k-1}) (20).

$$B(i,k) = \begin{cases} \sum_{j=1}^{|Q|-1} P_t(q_j | q_i)P_e(x_k | q_j)B(j,k+1) & \text{if } k < L, \\ P_t(q_0 | q_i) & \text{if } k = L. \end{cases}$$

The relationship between $F(i,k)$ and $B(i,k)$ is:

$$B(0,0) = P(S)$$

The Backward Algorithm alone can be calculated by the following procedure (20):

```

procedure unscaledBackwardAlg ( $Q, P_t, P_e, S, \lambda_{trans}, \lambda_{emit}$ )
1.  $len = |S|$ ;
2.  $\forall_{i \in [0, |Q|-1]} B[i][len] \leftarrow P_t(q_0 | q_i)$ ;
3. for  $k \leftarrow len-1$  down to 0 do
4.   for  $i \leftarrow 0$  up to  $|Q|-1$  do
5.      $sum \leftarrow 0$ ;
6.     for  $j \leftarrow 1$  up to  $|Q|-1$  do
7.        $sum \leftarrow sum + P_t(q_j | q_i) * P_e(S[k] | q_j) * B[j][k+1]$ ;
8.      $B[i][k] \leftarrow sum$ ;
9. return  $B$ ;

```

Graphically, the Forward-Backward Algorithm in state S_i at time t and state S_j at time $t + 1$ can be illustrated Figure 17: Illustration of the sequence of operations required for the computation of the joint event:

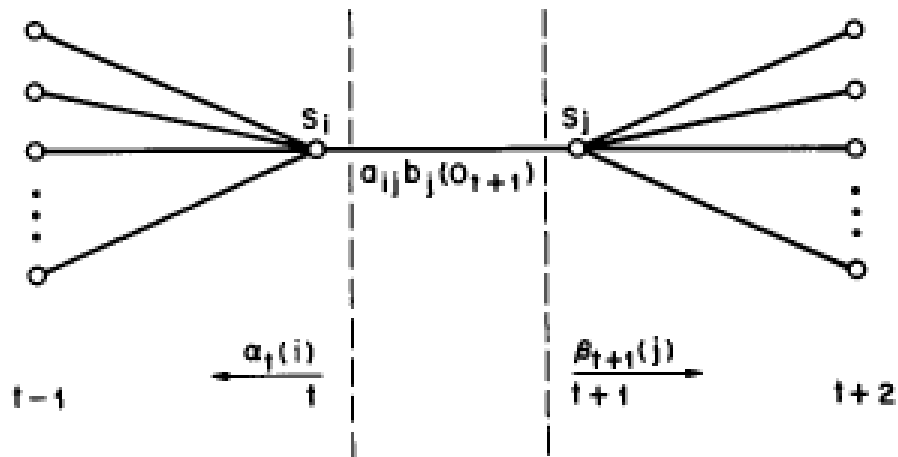


Figure 17: Illustration of the sequence of operations required for the computation of the joint event (13)

The following Forward-backward algorithm is adopted from (20).

Let us define some formal notations:

$$\begin{aligned}
 \mathbf{F}(i,k) &= P(x_0 \dots x_{k-1} | q_i) = P(M \text{ emits } x_0 \dots x_{k-1} \text{ by any path ending in state } q_i, \text{ with } x_{k-1} \text{ emitted by } q_i). \\
 \mathbf{B}(i,k) &= P(x_k \dots x_{L-1} | q_i) = P(M \text{ emits } x_k \dots x_{L-1} \text{ and then terminates, given that } M \text{ is in state } q_i, \text{ which has emitted } x_{k-1}). \\
 \mathbf{F}(i,k)\mathbf{B}(i,k) &= P(x_0 \dots x_{k-1} | q_i)P(x_k \dots x_{L-1} | q_i) = P(x_0 \dots x_{L-1} | q_i)^* \\
 \mathbf{F}(i,k)\mathbf{B}(i,k)/P(S) &= P(q_i, k-1 | S) \\
 \text{expectation}(E_{q_i, s}) &= \sum_{q_i, k} P(q_i, k | S) C(q_i, k) = \sum_{q_i, k} \frac{F(i, k+1)B(i, k+1)}{P(S)} C(q_i, k) \\
 \text{where } C(q_i, k) &= 1 \text{ if } q_i = q \text{ and } x_i = s; \text{ otherwise } 0. \\
 \text{expectation}(A_{i,j}) &= \sum_{q_m, q_n, k} \frac{F(m, k)P_i(q_n | q_m)P_\epsilon(x_k | q_n)B(n, k+1)}{P(S)} C(q_m, q_n, k) \\
 \text{where } C(q_m, q_n, k) &= 1 \text{ if } q_m = q_i \text{ and } q_n = q_j; \text{ otherwise } 0.
 \end{aligned}$$

(20)

Based on the above notations, a complete procedure for Forward-Backward Algorithm is shown as:

```

procedure baumWelch(ref M, T, n)
1.  $(Q, \alpha, P_t, P_e) \leftarrow M;$ 
2. for  $h \leftarrow 1$  up to  $n$  do
3.  $\forall_{i \in [0, |Q|-1]} \forall_{j \in [0, |Q|-1]} A[i][j] \leftarrow 0;$ 
4.  $\forall_{i \in [0, |Q|-1]} \forall_{k \in [0, |S|-1]} E[i][k] \leftarrow 0;$ 
5. foreach SET do
6.  $F \leftarrow \text{forwardAlgorithm}(M, S);$ 
7.  $B \leftarrow \text{backwardAlgorithm}(M, S);$ 
8.  $\text{updateCounts}(A, E, F, B, Q, S);$ 
9.  $\text{updateModel}(M, A, E);$ 

```

} compute Fwd & Bkwd DP matrices
 } accumulate expected counts for E & A

```

procedure updateCounts(ref A, ref E, F, B, Q, S)
1.  $P \leftarrow B[0][0];$ 
2. for  $i \leftarrow 1$  up to  $|Q|-1$  do
3. for  $k \leftarrow 0$  up to  $|S|-1$  do
4.  $E[i][S[k-1]] \leftarrow E[i][S[k-1]] + F[i][k] * B[i][k] / P;$ 
5. for  $j \leftarrow 0$  up to  $|Q|-1$  do
6.  $A[j][i] \leftarrow A[j][i] + F[j][k] * P_t(q_i | q_j) * P_e(S[k] | q_i) * B[i][k+1] / P;$ 
7.
8.  $A[i][0] \leftarrow F[i][|S|] * P_t(q_0 | q_i) / P;$ 

```

$\frac{F(i, k+1)B(i, k+1)}{P(S)}$
 $\frac{F(m, k)P(q_{k+1} | q_m)P_e(x_{k+1} | q_m)B(m, k+1)}{P(S)}$

Duke
UNIVERSITY

(20)

4.1.2.3 Laplace Smoothing

The purpose of Laplace Smoothing is to tune the HMM model. The first time after we implemented the complete HMM model, Viterbi recognizer did not identify the command word input. After research for the reasons, we found out that the problem is caused by the nature of the Viterbi Algorithm and our selection of the six command words. The commands are relatively short words whose lengths are either 2 or 3 phonemes. As a result, the length of the observation sequence is usually around 10-20 symbols/sample. The number of distinct observation symbols per state is defined to be 128 in our application. Therefore, for those observation symbols that did not appear in any of the 20 training sample/word, there exist zero probabilities in the observation symbol probability distribution matrix b . As we described previously, Viterbi Algorithm is based in logarithm. If any probability in the matrix is zero, its logarithm will be negative infinity. If there is any observation symbol that appears in the new observation sequence during

HMM recognition that is not in any of the 20 samples during training phases, the HMM recognizer with Viterbi Algorithm cannot yield a meaningful result. The whole process will get interrupted by the negative infinity. Hence, it results in low recognition rate, especially in a noisy environment.

The correction estimation of the probability of rare events is a primary concern in building language models (17). Laplace Smoothing is our correction method in this case to avoid estimating any probabilities to be zero, even for events never observed in the data. For HMMs, this is important since zero probabilities can be problematic for some algorithms, such as Viterbi Algorithm.

The actual implementation of Laplace Smoothing is relatively simple. A small number, $1/(N*M) = 1/(5*128)$ in our application, is added into the observation symbol probability distribution matrix b and then the b matrix multiplied some constant for normalization. This will ensure there is no zero probability to cause problems to our algorithm. Since it is such a small number, it would not affect the performance of the HMM recognizer. Laplace Smoothing greatly improves the performance of the whole Speech Recognition system.

4.2 Joystick Steering Control Design

The joystick was the only user interface for the steering control in our original design. Initially, I planned to design and build a joystick tailored to the application from basic potentiometers. After researching the current joystick designs, the problem has become time consuming and cost inefficient. The average price for potentiometers is around two to three dollars each. In order to build a joystick, we need two potentiometers for x and y axis flexibilities, RC components and a circuit board. The total cost of them will highly exceed the price of a mini-joystick. The joystick I identified and purchased on Digikey.com is \$7.34, with two axes and two potentiometers. It will be sufficient for our

implementation. As a result, my part was reduced to integrating the joystick into the rest of the control system.

The mini-joystick with two channel inputs takes in x and y direction and speed information and encodes them as voltage variations. Those two channels of voltage information are converted to digital by the Analog-to-Digital Converter built into the microcontroller. There is an algorithm in the micro-controller that converts the x and y Cartesian signal into Polar coordinates. The magnitude and angle information can be easily calculated in Polar coordinates. Finally, direction and speed can be decoded from magnitude and angle information and they are mapped to the same set of command words as the speech recognition system for outputting to the propulsion system. The mapping is done through matching several conditions dedicated to different command words. Please refer to Figure 18: Joystick Command Mapping for the mapping scheme.

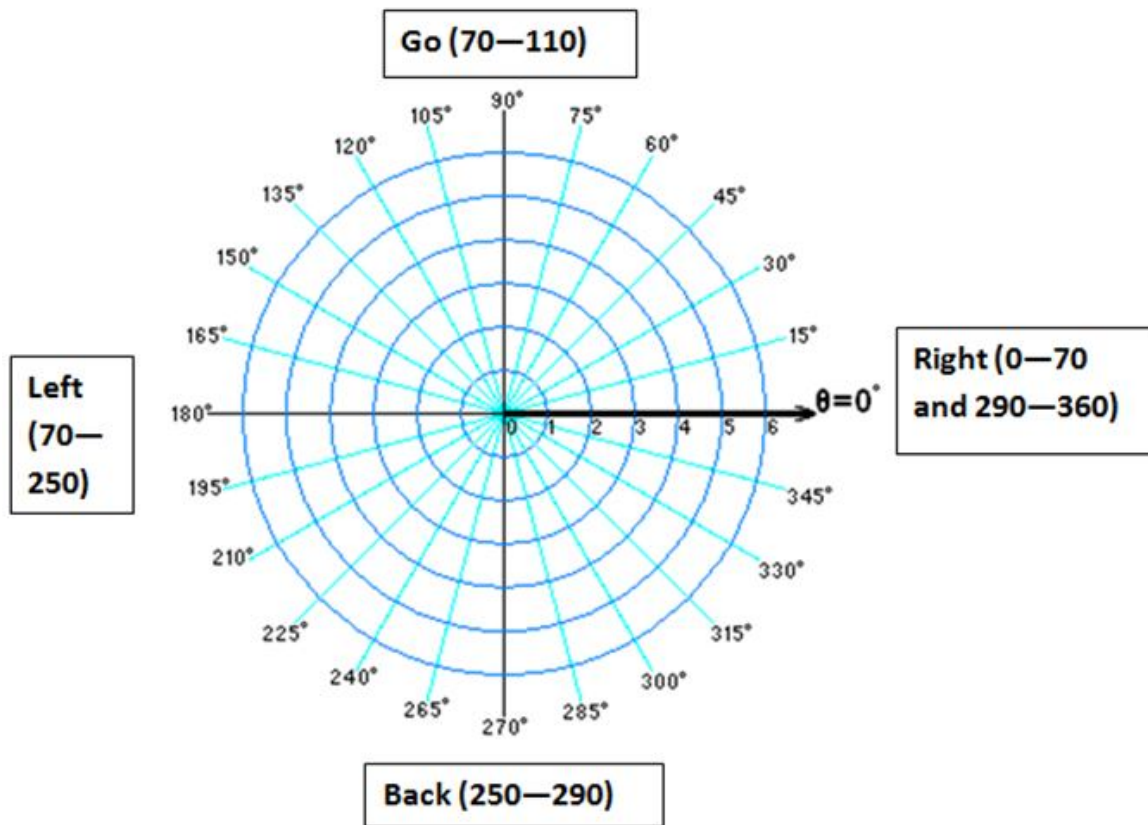


Figure 18: Joystick Command Mapping

5 Results and Discussion

For design purposes, the Automatic Speech Recognition and the joystick steering control systems are tested using a mapping scheme which maps the number of times a LED blinks to a specific command word that is identified. A set of testing command words: Go, Stop, Slow, Left, Right, Back are represented by the number of times a LED blinks. Similarly, the joystick outputs are also mapped to the same set of command words. Therefore, the same outcome for testing purposes can be observed as well.

The final result of the Automatic Speech Recognition system with speaker-dependent discrete HMMs performed with different types of codebook is summarized into Table 2: Automatic Speech Recognition Rates. Averaging those data, the overall recognition rate is 75.8%. We agree that this preliminary overall recognition rate is acceptable since there is space for improvement, such as to set the proper thresholds for recognition, to optimize the HMMs with different parameters, and to use other types of smoothing techniques. Due to time constraint, we did not have an opportunity to fully optimize the HMMs. I believe that the overall performance of the system can be greatly improved if we explore the options more completely.

Table 2: Automatic Speech Recognition Rates

Command Words	Combined Codebook	Codebook Without noise	Codebook With noise
Stop	16/20	15/20	24/29
Left	17/20	7/19	Nan
Back	17/20	17/19	13/19
Slow	(16-1)/20	15/20	(18-3)/19
Go	(14-1)/20	(17-1)/20	9/19
Right	18/20	17/20	16/19

As mentioned in the Experimental and Design Procedures chapter, the HMM training part of this Automatic Speech Recognition System is performed in a computer due to memory limitations of a micro-controller. Hence, only the HMM recognition portion is in real-time and the command word models have to be trained before recognition. Since the run time of performing the training is not in real time, the process run times for generating Table 2: Automatic Speech Recognition Rates result are recorded in

Table 3: HMM training run-time in sec:

Table 3: HMM training run-time in sec

Command Words	Combined Codebook	Codebook Without noise	Codebook With noise
Stop	46.4103	10.2025	15.5065
Left	13.8841	4.6488	19.9525
Back	12.8233	8.7361	39.0471
Slow	12.1213	7.7688	8.2525
Go	11.3569	4.8828	3.978
Right	8.9233	7.0512	27.9086

Compare the run-time values in

Table 3: HMM training run-time in sec, we notice that the Codebook Without noise has the lowest run-time in general. The reason behind is that with noise or a larger sample set codebook for the Combined Codebook case, there is more variance in the observation sequences. HMM training as an iterative process, will require more run time to converge. Therefore, HMM trainings for the Combined Codebook and Codebook with noise run slower than the one without noise.

For samples of intermediate results of the HMM training and recognition system, please refer to Appendix B: Viterbi Recognition results.

The alternative joystick steering control is observed to functions properly. Since the testing scheme is the same as the Automatic Speech Recognition system, we do not want to repeat listing the same result.

Last but not least, the total cost of the MWA project is calculated to be \$371.08. To be cost-competitive, as one of our primary objective, is clearly achieved. In the Objective chapter, \$1000 is stated as a standard to evaluate whether the project is under a competitive budget.

Table 4: Total MWA Project Cost

Item	Cost (\$)
Mini-joystick x1	7.34
Microphone x1	3.68
Micro-controller & Expansion Board x1	168.30
Breadboard x1	7.99
Circuit components (resistors, capacitors, op-amps, wires etc.)	20.00
Windshield Wiper Motor x2	90.00
8.5" Wheel x2	20.50
Lumber	3.00
Simpson BC40 Brackets x2	6.81
Right Angle Brackets x2	15.94
Plastic Clamps	1.13
Relays x7	14.28
MOSFET x2	2.95
2N2222 BJT x7	9.16
Total	371.08

6 Conclusions and Recommendations

6.1 Conclusions

The overall objective of the MWA project is to enable a simple conversion between a manual wheel chair and an electric wheel chair under a competitive budget. As

Table 4: Total MWA Project Cost is shown above, our actual cost is much less than the price difference between a typical manual and electric wheel chair. It indicates that our MWA project has a strong potential to be marketable. Although the MWA system is not at the stage to be a marketable device, this certainly appears to be achievable with some modifications.

Another aim for the MWA system is to be user-friendly. In particular, two types of steering control methods were designed and implemented successfully. The first control is the Automatic Speech Recognition system which achieves a 75.8% recognition rate. My focus is the back end process of this system which is divided into HMM training and recognition parts. The recognition part is implemented directly onto the micro-controller and is able to perform real-time speech recognition given the models. As for the HMM training, it is implemented in C programming language in a computer with the potential to be applied in a micro-controller. The alternative joystick steering control onto a micro-controller is also achieved. It can function in real-time as parallel to the Automatic Speech Recognition system.

In conclusion, the major objectives of the MWA project as well as my responsible parts are achieved. In order to improve the MWA system to be able to compete in the rapidly growing marketplace, more improvements listed in the following Recommendations chapter should be done.

6.2 Recommendations

Since my main responsibilities are to design the back-end of the Automatic Speech Recognition system and joystick steering control, I will focus on discussing their recommendations at the following.

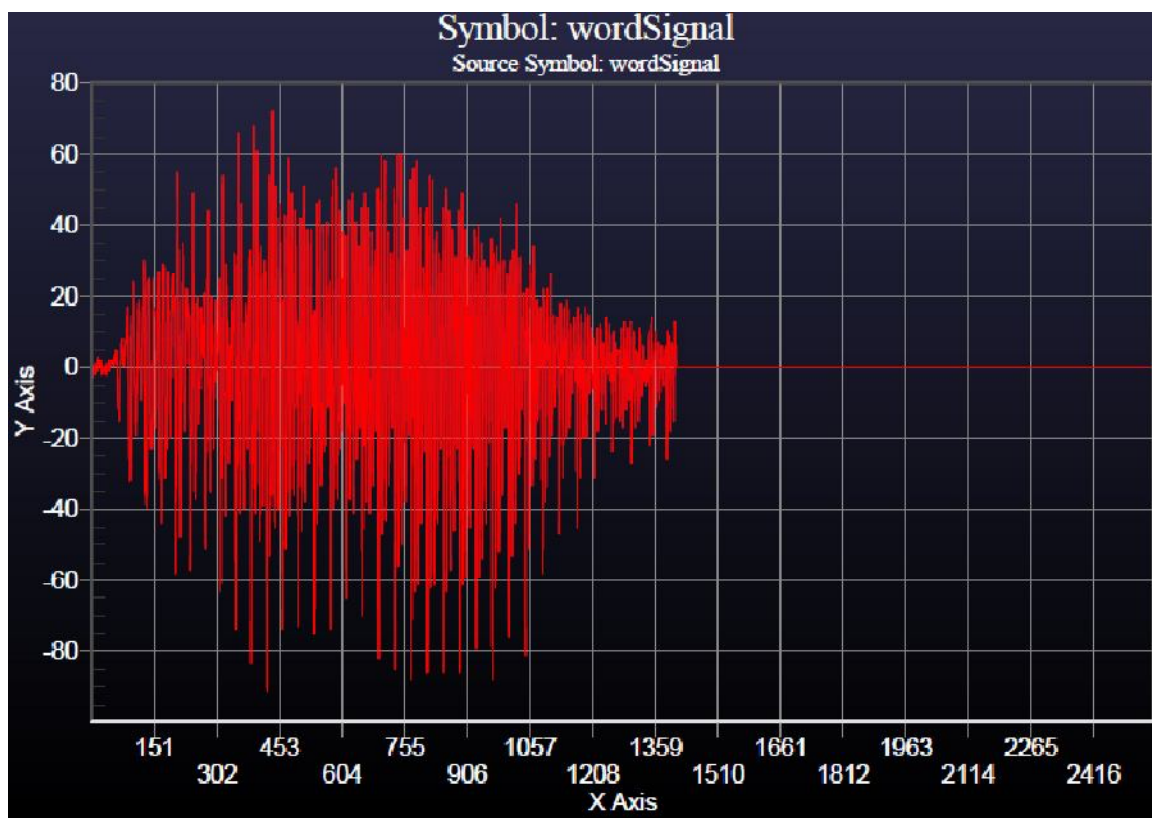
Recognition rate is one of our major concerns and this can be further improved by optimizing the threshold settings, HMMs with different parameters, and the smoothing techniques. In particular, we discovered that setting the proper threshold for each word is different as it depends on the features of the specific word. Therefore, optimal thresholds for different words should be found out by experiment

The micro-controller memory limitation restrains the HMM training to be performed on a computer. This constrain could be removed by purchasing external memory. Hence, the whole Automatic Speech Recognition system could be performed in real time given the hardware improvement. It will also help to increase the system speed as it eliminates the steps to transfer data from a computer onto a micro-controller.

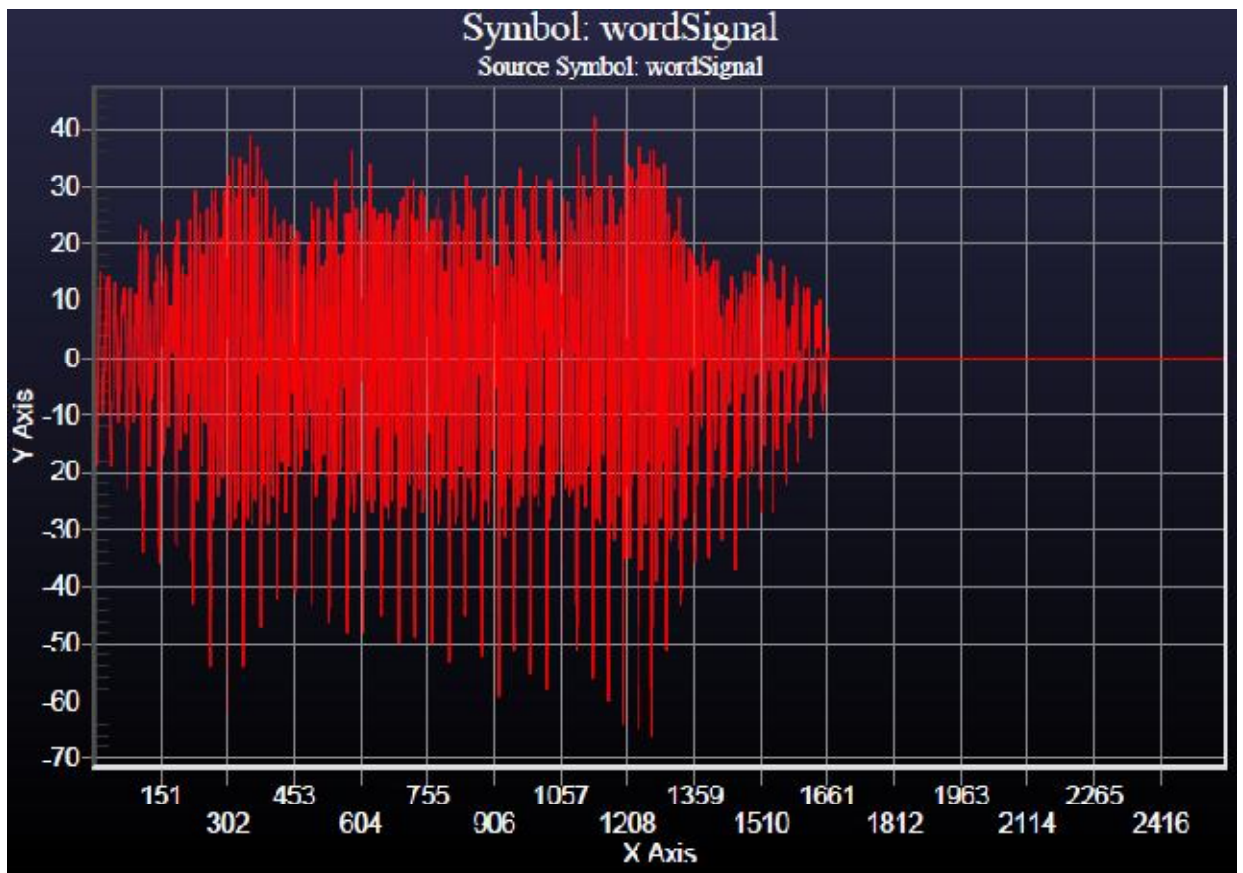
Last but not least, the production cost can be further reduced during massive production. The MWA control systems are implemented on a micro-controller starter kit, which is much more expensive than a micro-controller. Therefore, the cost of the current micro-controller and an extension board is \$168.30, which can be reduced to less than \$10. The cost of this MWA can potentially become more competitive.

Appendix A: Raw Speech Signal Samples within a linear window

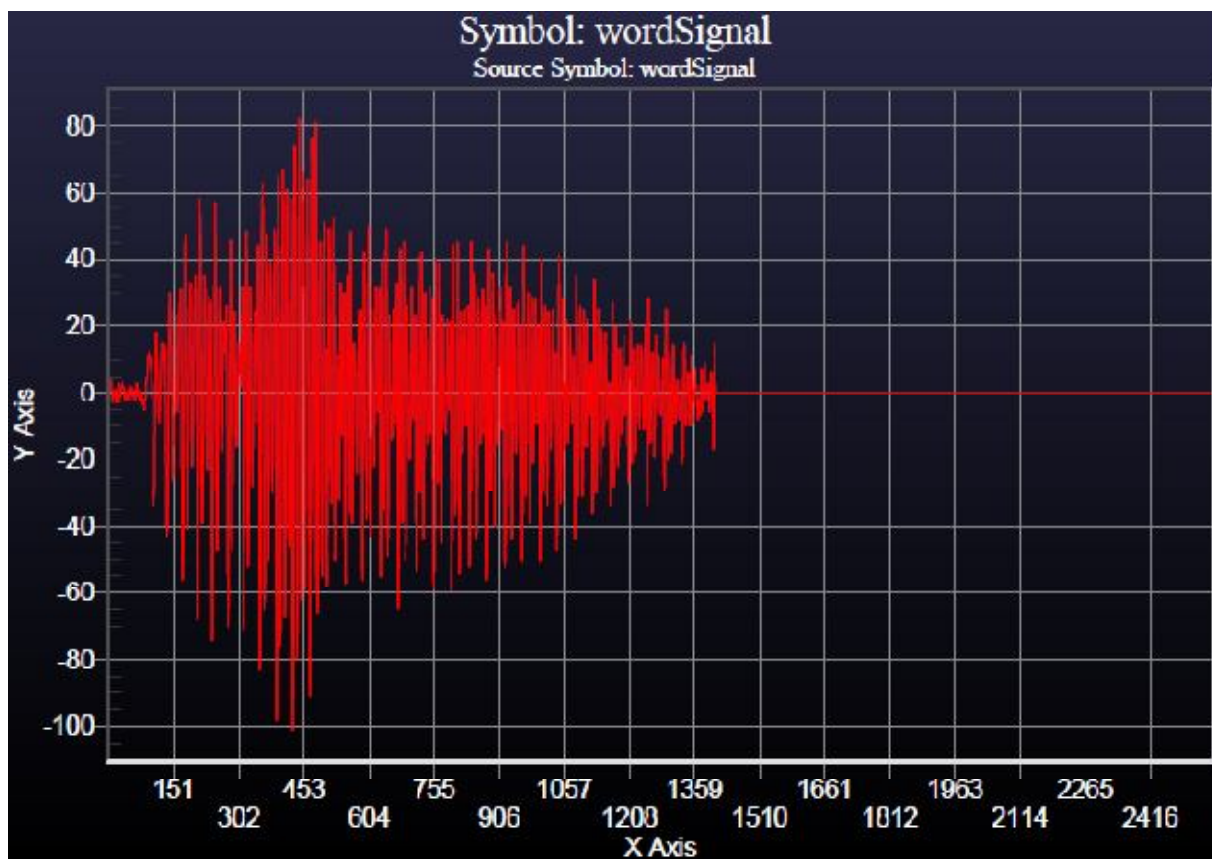
“Back”



“Left”



“Stop”



Appendix B: Viterbi Recognition results

HMM Training:

The transition probability matrixes (5*5) for the 6 HMM are at the following. Since the observation probability matrix is too big (5*128), it is not shown here.

A different index is assigned to one of the six command words to distinguish them.

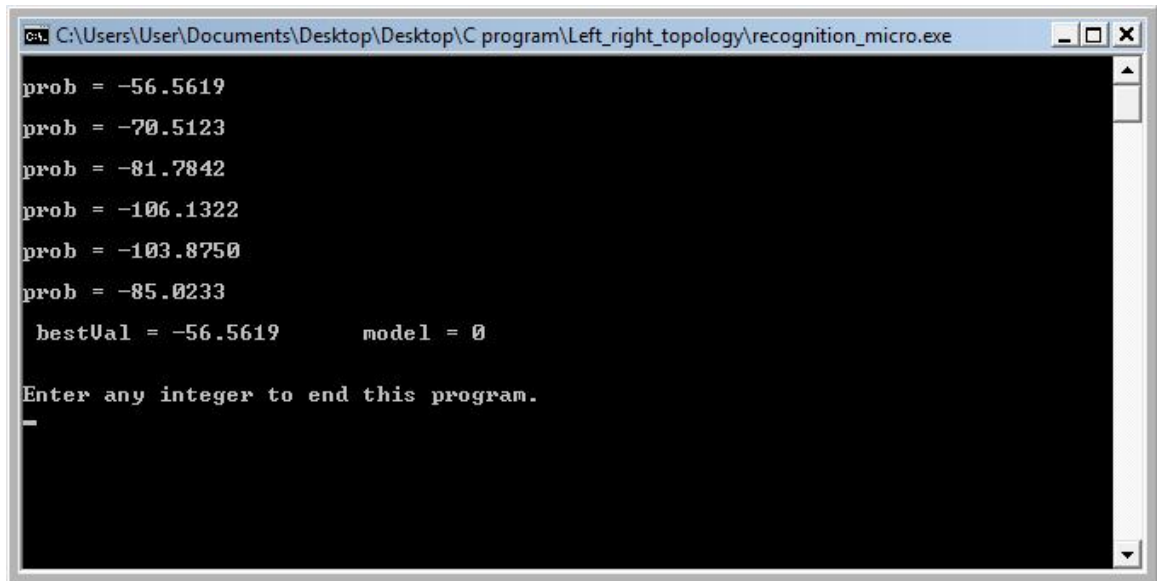
"stop" => [0]; "left" => [1]; "back" => [2]

"slow" => [3]; "go" => [4]; "right" => [5]

```
float transition[numModels][numStates][numStates] =
  {{{{0.6727,0.3273,0.0000,0.0000,0.0000},{0.0000,0.1374,0.8626,0.0000,0.0000},{0.0000,0.0000,0.7648,0.2352,0.0000},{0.0000,0.0000,0.0000,0.9968,0.0032},{0.0000,0.0000,0.0000,0.0000,1.0000}}},
  {{{{0.6837,0.3163,0.0000,0.0000,0.0000},{0.0000,0.2355,0.7645,0.0000,0.0000},{0.0000,0.0000,0.3347,0.6653,0.0000},{0.0000,0.0000,0.0000,0.7627,0.2373},{0.0000,0.0000,0.0000,0.0000,1.0000}}},
  {{{{0.7152,0.2848,0.0000,0.0000,0.0000},{0.0000,0.2809,0.7191,0.0000,0.0000},{0.0000,0.0000,0.7305,0.2695,0.0000},{0.0000,0.0000,0.0000,0.9872,0.0128},{0.0000,0.0000,0.0000,0.0000,1.0000}}},
  {{{{0.7232,0.2768,0.0000,0.0000,0.0000},{0.0000,0.2284,0.7716,0.0000,0.0000},{0.0000,0.0000,0.6841,0.3159,0.0000},{0.0000,0.0000,0.0000,0.9813,0.0187},{0.0000,0.0000,0.0000,0.0000,1.0000}}},
  {{{{0.6699,0.3301,0.0000,0.0000,0.0000},{0.0000,0.2019,0.7981,0.0000,0.0000},{0.0000,0.0000,0.4892,0.5108,0.0000},{0.0000,0.0000,0.0000,0.9203,0.0797},{0.0000,0.0000,0.0000,0.0000,1.0000}}},
  {{{{0.6721,0.3279,0.0000,0.0000,0.0000},{0.0000,0.3668,0.6332,0.0000,0.0000},{0.0000,0.0000,0.7457,0.2543,0.0000},{0.0000,0.0000,0.0000,0.9401,0.0599},{0.0000,0.0000,0.0000,0.0000,1.0000}}}}};
```


HMM Recognition:

Provided this model, a “stop” sequence “93, 93, 26, 44, 43, 43, 44, 43, 51, 51” is input into the HMM recognizer. The final output will be the index with the highest scoring. The following command window shows that model [0] which is “stop” has the highest probability scoring. Therefore, “stop” is the identified.

A screenshot of a Windows command prompt window. The title bar reads "C:\Users\User\Documents\Desktop\Desktop\C program\Left_right_topology\recognition_micro.exe". The window contains the following text:

```
prob = -56.5619
prob = -70.5123
prob = -81.7842
prob = -106.1322
prob = -103.8750
prob = -85.0233
bestVal = -56.5619      model = 0

Enter any integer to end this program.
_
```

Threshold Setting:

The importance of setting the correct threshold has been mentioned over and over again in the body of the report. How do we select the right threshold? The above command window can act as a good example to illustrate the idea. For “stop”, the highest probability scoring a right observation can give is -56.56 in this case. In order to avoid false recognition, the threshold should be set between the -56.56 and the second highest probability scoring -70.51 in this case. For different words, the probability scoring is not fixed. Therefore, an efficient threshold has to be determined according to each individual word.

Appendix C: HMM training and recognition implementation in C programming language

Due to the length of the C code, the computer program will be stored in a CD and be handed it with the hardcopy of report.

References

1. **Reinhardt, Stephen.** A History of Wheelchairs. [Online] June 13, 2008. [Cited: Oct. 5, 2008.] <http://electricwheelchairs.wordpress.com/2008/06/13/a-history-of-wheelchairs/>.
2. **Editor.** Electric Wheelchairs. *The Wheelchair Site* . [Online] March 26, 2008. [Cited: Oct. 5, 2008.] <http://www.thewheelchairsite.com/electric-wheelchairs.aspx>.
3. Discount Medical Supplies. *Centiva HomeCare, Inc.* [Online] 2008. [Cited: Oct. 6, 2008.] <http://www.home-medical-equipment-depot.com/servlet/the-Wheelchairs/Categories>.
4. Hi Speed Electric Bicycle Bike Hub. [Online] eBay Inc., Oct. 6, 2008. [Cited: Oct. 6, 2008.] http://shop.ebay.com/items/_W0QQ_nkwZHiQ20SpeedQ20ElectricQ20BicycleQ20BikeQ20HubQQ_armrsZ1QQ_fromZR40QQ_mdoZ.
5. **Reed, David Gerade.** *Speaker-dependent Isolated Word Recognition*. Hamilton : McMaster University, 1987.
6. **Claudio Becchetti and Lucio Prina Ricotti.** *Speech Recognition Theory and C++ Implementation*. New York : John Wiley & Sons. Inc., 1999. 0471977306.
7. **John R.Deller, Jr; John H.L. Hansen; John G. Proakis.** *Discrete-Time Processing of Speech Signals*. New York : Wiley-Interscience, 2000.
8. mobility. *how stuff works*. [Online] Oct. 30, 2008. [Cited: April 24, 2009.] <http://health.howstuffworks.com/shepherd-center6.htm>.
9. *Microcontroller Implementation of a Voice Command Recognition System for Human-Machine Interface in Embedded Systems*. **Carlos Bernal-Ruiz, Francisco E. Garcia-Tapias, Bonifacio Martin-del-Brio and Antonio Bono_nuez.** Zaragoza : IEEE, 2005. 07039402-X/05.
10. *Embedded Speech Recognition System on 8-bit MCU Core*. **Dong Wang, Liang Zhang, Jia Liu and Runsheng Liu.** Beijing : IEEE, 2004, Vols. V-301. 0780384849/04.
11. *The Concepts of Hidden Markov Model in Speech Recognition*. **Waleed H. Abdulla and Nikola K. Kasabov.** Otago, New Zealand : Department of Information Science, University of Otago, 1999.

12. *A Hybrid Speech Recognition System Using HMMs with an LVQ-trained Codebook*. **Hitoshi Iwamida, Shigeru Katagiri, Erik McDermott, and Yoh'ichi Tohkura**. Kyoto, Japan : The Acoustical Society of Japan, 1990, Vol. 11.
13. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. **Rabiner, Lawrence R.** Murray Hill, NJ, USA : IEEE, 1988.
14. *SPEECH RECOGNITION WITH LOW-COST*. **Carlos Bernal-Ruiz, Francisco E. García-Tapias, Bonifacio Martín-del-Brío**.
15. *SINGLE-CHIP SPEECH RECOGNITION SYSTEM*. **Shi Yuanyuan, Liu Jia and Liu Runsheng**. Beijing, P.R. China : IEEE, 2000.
16. PIC32. *Microchip*. [Online] Microchip, 2008. [Cited: April 24, 2009.] http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2591.
17. **Ben Gold, Nelson Morgan**. *Speech and Audio Signal Processing*. Toronto : John Wiley & Sons Inc., 2000.
18. **Irina Medvedev**. Isolated-Word Speech Recognition Using Hidden Markov Models. *MIT Education*. [Online] April 19, 2001. [Cited: April 26, 2009.] http://www.mit.edu/~6.454/www_spring_2001/irinam/seminar.ppt .
19. Hidden Markov Models and the Viterbi algorithm. *Cornell University*. [Online] [Cited: Oct. 5, 2008.] <http://people.ccmr.cornell.edu/~ginsparg/INFO295/vit.pdf>.
20. **Majoros, Bill**. Hidden Markov Models. *Comp Sci 261*. [Online] [Cited: March 23, 2009.]
21. **Rob Schapire**. COS 402: Artificial Intelligence. *Computer Science 402*. [Online] Princeton University, Nov. 30, 2007. [Cited: March 26, 2009.] <http://www.cs.princeton.edu/courses/archive/fall07/cos402/assignments/viterbi/>.

VITA

NAME: Hailun Huang

PLACE OF BIRTH: Shenzhen, Guangdong Province, P.R. China

YEAR OF BIRTH: 1985

SECONDARY EDUCATION: London International Academy (2003)

HONOURS and AWARDS: The Dr. Harry Lyman Hooker Scholarship 2008

The University (Senate) Scholarship 2007

The Nortel Networks Entrance Scholarship 2005, 2006

The George and Nora Elwin Scholarship 2005, 2006

Dean's Honor List 2005, 2006, 2007, 2008

etc.