Development of Automatic Speech Recognition system for voice activated Ground Control System

Kavitha S, Veena S, R Kumaraswamy
PG Student, Principal Scientist, Professor & Head
SIT Tumakuru, CSIR- National Aerospace Laboratories Bengaluru, SIT Tumakuru
Kavithas.kavi08@gmail.com, veenas @ nal.res.in, hyrkswamy@gmail.com

***Abstract* - This paper gives details of the development of a speech recognition system for voice activated Ground Control Station (GCS). The speech recognition is implemented using MATLAB and the results are validated against the Hidden Markov Model Tool Kit (HTK), an open source tool for speech recognition. The menu items of Mission planner, a typical open source GCS used for flying of Micro Air Vehicles (MAV) are used for the experiments.**

Keywords – ASR, MFCC, HMM, HTK, MATLAB, MISSION PLANER, SPEECH CONTROLLED MAV

I INTRODUCTION

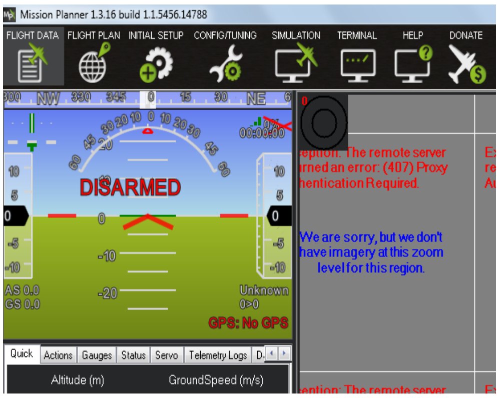
The **Micro Air Vehicles** (MAV) are employed in variety of missions due to their increased capability and relatively low cost with an aim of carrying out surveillance missions [1]. MAVs are remote controlled through a portable ground control station (GCS). Through the GCS, MAV hardware is configured and it also GCS gathers all the information about the MAV status and allows sending the commands according to the specified missions [2]. Therefore, GCS is software with multiple menu pages, operated by keyboard and/or mouse. This can be a cumbersome process at site of the mission. Mission Planner is user configurable software, installed on GCS computer which allows planning and control of flight operations. It provides the most complete operations and functionality for vehicle setup as well as pre-flight mission planner, in-flight monitoring and post flight log file analysis. All such features lot of user intervention and can become cumbersome at the site of mission. Therefore, in literature speech based control of MAVs is found to be an interesting alternative [3]. Thus, the use of speech-based input can become very good alternative to MAV operators to navigate through menus and select options more quickly and in a robust manner [4]. The recent document from DARPA also recognizes speech as an effective tool and emphasizes its need towards developing robust technology [5].

The Automatic Speech recognition involves speech feature extraction and its identification through modelling. Hidden Markov Model (HMM) is the most commonly used statistical modeling tool [6]. HMM can be implemented using an open source tool called HTK [7], the Hidden Markov Model tool kit. The HTK is an installable on Linux and it poses integration issues with the GCS software. Also, the source code of HTK is not available. Hence, it is important to have a complete source code that can be integrated with GCS.

In this paper, menu commands of the Mission planner, an open source GCS is considered for case study. The speech recognition algorithm is developed on MATLAB and the results are benchmarked with the results obtained from HTK tool kit.

This paper is organized as follows: section 2 gives a brief description of simple ASR system, introduction to HMM. Its implementation using HTK 3 and MATLAB will be discussed in section 3 and section 4, respectively. Section 5 covers the experimental results and section 6 presents conclusion.

II MISSION PLANNER



eption: The remote server rned an error: (407) Proxy hentication Required

We are sorry, but we don't have imagery at this zoom level for this region.

Figure 1 : Snapshot of Mission Planner Software

Figure 1 shows the screen shot of Mission Planner, which is an open source GCS for controlling of MAVs. It is compatible with Windows versions only. It serves as a configuration utility for MAV.

The Mission Planner main menu buttons are as follows:

- ***Connect***: have parameters required to connect the Mission Planner to a MAV autopilot.
- ***Flight Data***: Information about Flight Data screens.

978-1-4673-6667-0/15/$31.00©2015 IEEE

- ***Flight Plan***: Have parameters for preparing flight plans for missions.
- ***Initial Setup***: Initial Setup screen.
- ***Configuration Tuning***: Parameters for Configuration of screens.
- ***Simulation***: Setting the mission planner to simulator mode to 'simulate' flying.
- ***Terminal***: Information about Terminal screens.
- ***Help***: Help on mission planner

Each of these menus has many sub menus. In this paper, only isolated words are considered for experiments.

III ASR SYSTEM DESCRIPTION

Figure 2 shows the basic block diagram description of a simple ASR system even though plenty of literature is available on this, for understanding purpose.

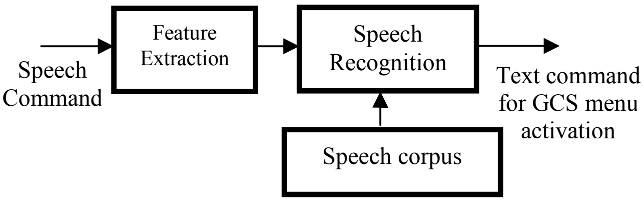


Figure 2: Illustrates the block diagram of ASR system

Each block of ASR system is explained as follows:

A. Feature extraction

Feature extraction aims at achieving compact coefficients from the speech signal by preserving temporal and/or spectral characteristics of speech. [8] Reports performances of Linear Prediction Coefficients (LPC), Linear Prediction Cepstrum Coefficients (LPC-CC), Linear Frequency Cepstrum Coefficients(LPC-CC), Reflection Coefficients (RC) and Mel-Frequency Cepstral Coefficients (MFCC) using template-based, dynamic time warping recognizer for monosyllabic word recognition and concludes that compared to other parameter types, MFCC parameters outperform. Hence, Cepstral analysis has been used extensively for feature extraction in speech recognition.

B. Speech recognition

This block measures the similarity between an input speech and a reference pattern/a model (obtained during training) and determines a reference/ a model, which best matches the input speech. Hidden Markov Model (HMM), a statistical model is the most popular model used for speech recognition [6] [9].

HMM is very powerful mathematical tool for modelling time series and are based Markov Chain. The most likely word with the largest probability is produced as the result of the given speech waveform. A natural extension of Markov chain is Hidden Markov Model (HMM), the extension where the internal states are hidden and any state produces observable symbols or evidences [6].

In HMM, the probability distributions A, B and π are written in a compact form denoted by lambda [6] as

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

Where π = initial state distribution vector.
A = State transition probability matrix.
B = continuous observation probability density function matrix.

For speech, a HMM model with transition from one state to next state and to itself is considered. Based on this **A** is initialized with the equal probability for each state. The initial state distribution vector is initialized with the probability to be in state one at the beginning, which is assumed in speech recognition theory. The matrix **B** has elements MFCC coefficients and delta coefficients weighed by multivariate single Gaussian distribution [9]. The system is solved as given in [6]. Instead of direct solution, the following three algorithms are used from the point of computational simplicity. The HMM has two phases training and testing. Training is carried out for each utterance by

- Baum-Welch forward backward method to estimate the probability of each observation sequence for the given model.
- Baum-Welch expectation modification algorithm re-estimates λ to maximise the probability of observation.

At the end of this step, a best possible model is generated for each word in the database..

The testing phase uses modified Viterbi algorithm generates **B** matrix for testing utterance. This **B** matrix with **A** and π matrices of each of the model are used to generate scores using alternative Viterbi algorithm [9].The HMM model with highest score represents the test utterance is corresponding to that model.

C. Database creation

The database for training and testing HMM contains speech audio files and corresponding text transcriptions and is known as speech corpus. The mission planner commands fall into the category of isolated words or connected words.

IV HTK SYSTEM

The HTK is a collection of programming tools developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED) for creating and manipulating HMMs. HTK contains modules for building HMM and include tools for HMM training, recognition and evaluation [5]. Perl programming language has been used to call the modules from HTK to automate training and test procedures [10].

In our work, we considered HTK 3.4.1 which is commonly installed in Linux Environment like Ubuntu 14.04 with

WaveSurfer 1.8.5 tool for recording of speech files. The recorded speech files are converted to .htk files using the command HCopy. This command uses the configuration file myMfcc.conf which specifies the configuration parameters as shown in the Table 1 below.

Command line in linux:

```
HCopy –C myMfcc.conf wav-file mfc-file
```

HCopy will run for every pair of source waveform file (wav-file) and its corresponding MFCC file (mfc-file) resulting in conversion of all the .wav files into .htk files.

Table 1: Configuration parameters related to MFCC extraction process.

Configuration Parameters	Value
SOURCEKIND SOURCEFORMAT SOURCERATE	WAVEFORM WAV 625
TARGETKIND TARGETFORMAT	MFCC_D HTK
PREEMCOEF	0.97
TARGETRATE WINDOWSIZE USEHAMMING	100000 320000 TRUE
NUMCHANS NUMCEPS DELTA_WINDOW	26 12 5

To create HMM model for each word, same priori topology or prototype is defined [14]. The topology defines the following parameters:

~o <VecSize> <MFCC_D> - Header of the HMM description file, giving the size of coefficient vector and feature coefficient type (MFCC_D).

<NumStates> - Total number of HMM states, with 2 non-emitting states (first and last).

<State> and <NumMixes> - Each state observation function description. Single Gaussian Model is used to describe each state observation function. Such a Gaussian model is completely described by mean and variance vectors. Non emitting states are not described as they have no observation function.

<Stream> 1 - Number of MFCC sets per observation sequence.

<Mean> and <Variance> - Defines the mean vector with each element initialized to zero and variance vector with each element initialized to one of the current state observation function.

Mean and Variance coefficients will be trained later.

<TransP> - Transition square matrix of size equal to number of states is defined for each HMM word model. Zero values indicate that the corresponding transitions are not allowed and other values are randomly initialised (but each row of matrix must sum to 1).

Transition matrix is modified during training process.

Parameters of HMM must be initialised properly with training data for fast and optimal convergence of the training algorithm before starting the training process. Using HInit tool, all word models can be initialized by typing a command:

```
HInit protos mfcc1 mfcc2 mfcc3
```

where protos is the name of the file holding the prototype HMM and mfcc1, mfcc2 etc., are the names of the feature vector files of the training speech data [14].

The HRest tool estimates the optimum values for HMM parameters (transition probabilities, mean and variance). To train each HMM word model, the estimation has to be repeated for number of times. Each time, the iteration number and algorithm convergence through the change measure. When there is no further decrease in change measure from one iteration to another, it's the time to stop the iteration process or else the estimation can be stopped manually by pre-defining maximum number of iterations.

Recognition includes testing of an utterance against all the word models and is done using HVite tool. HVite is based o Viterbi word recognition algorithm. Recognition can be done both online and offline. In case of offline recognition, testing utterance is recorded as audio file before recognition and in online recognition; the speech signal to be recognized can be input directly on the fly.

V MATLAB IMPLEMENTATION OF ASR

MATLAB 2013 is used to implement the system. The feature extraction using MFCC coefficients is accomplished using the command

```
M=melcepst(s,fs)
```

where s is the speech input after pre-emphasis and voice activity detection, fs is the sampling frequency. The HMM equations given in [6] is implemented to generate and test the models.

VI PERFORMANCE ANALYSIS

The database of utterances is created as follows and only isolated words from the Mission Planner menu are considered. Also, the model is developed for a speaker dependent system as the voice commands are issued by a single user.

Words to be identified: 12 words from the mission planner Menu: Actions, Config, Donate, Gauges, Help, Messages, Quick, Scripts, Servo, Simulation, Status and Terminal.

Training set: For each isolated word, 50 different utterances are recorded by a single speaker. Hence a set of 600 (12*50) audio files is given as source speech data for training. Utterances sourced from a multitude of users renders the model as speaker independent and less tightly fitted.

Speech signal is recorded using 'wavsurfer' tool at a sampling rate of 16 KHz.

Figure 3 gives MFCC and delta coefficients extracted for single frame of the word 'Actions'.

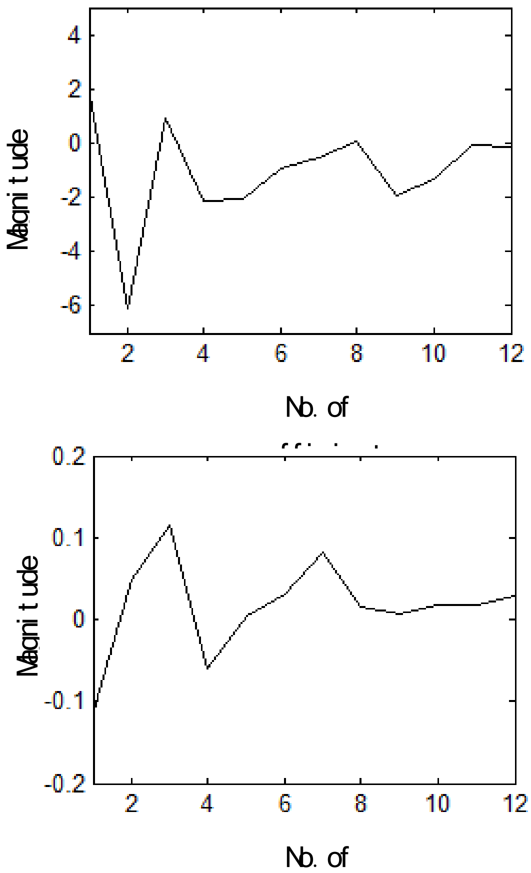


Figure 3: MFCC and Delta coefficients extracted for single frame.

These vectors are the feature vector of the utterance and are used to initialize matrix B. The forward and backward coefficients alpha and beta are computed to obtain the probability for the utterance 'Actions'. This procedure is followed for all the 50 utterances of 'Actions'. These alpha and beta values of each utterance is scaled and re-estimated to obtain maximum probability.

Consider Figure 4, here the maximum probability is obtained from 6th model. This model is the best model for the word 'Terminal'. Similarly for all the 12 words, best models are obtained.

To evaluate the models performance, testing is carried out i.e, how best the models recognize the given test utterance and the recognition results are defined in terms of score.

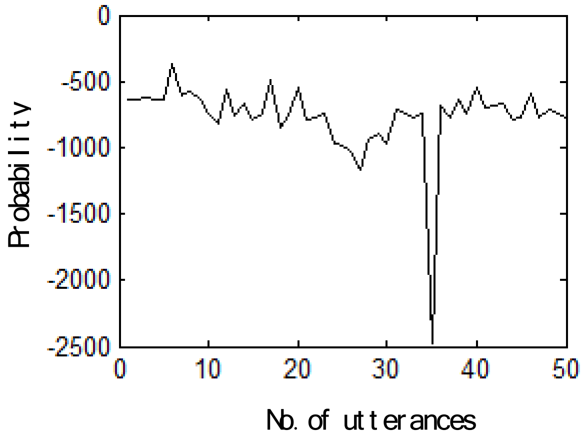


Figure 4: Probabilities of each utterances of word 'Terminal'.

Accuracies of ASR systems developed in MATLAB and HTK are compared for 5 states as shown in Table 2. The performance of MATLAB is better compared to HTK except for the case of small utterances. When number of states are increased to 10 as shown in Table 3, performance of both become comparable, however with MATLAB still has recognition problem for small words, since innately, HMM's are less suitable for smaller words. Alternatively, Neural Networks provide a better result in such cases.

Table 2: Comparison of Recognition Performance for 5 states model.

Word Model	**% of Recognition accuracy with 5 states**	
	HTK Toolkit	**MATLAB**
Actions	100%	100%
Config	53.33%	86.67%
Donate	60%	100%
Gauges	20%	93.33%
Help	33.33%	60%
Messages	100%	100%
Quick	73.33%	40%
Scripts	100%	93.33%
Servo	100%	100%
Simulation	100%	100%
Status	100%	100%
Terminal	100%	86.67%

Table 3: Comparison of Recognition Performance for 10 states model.

Word Model	% of Recognition accuracy with 10 states	
	HTK Toolkit	MATLAB
Actions	100%	100%
Config	93.33%	93.33%
Donate	100%	93.33%
Gauges	100%	80%
Help	93.33%	40%
Messages	100%	100%
Quick	93.33%	0%
Scripts	100%	93.33%
Servo	100%	100%
Simulation	100%	100%
Status	100%	100%
Terminal	100%	100%

VII Conclusion

This paper brings out the application of ASR in command and controlling of Micro Air Vehicles (MAV). To realize this, the ASR system needs to be integrated to the Ground Control Station (GCS) software. To facilitate this, the ASR code has been developed on MATLAB and the future plan is to convert it into a high level language code (C/C++) for integration with mission planner. The performance of this system is validated with the results obtained from HTK and it can be concluded that performance of both the systems are comparable.

REFERENCES

1] Davis, Kosicki, Boroson and Kostishack, “Micro Air Vehicles for Optical Surveillance”, *The Lincoln Laboratory Journal 197*, Volume 9, Number 2, 1996.

2] Natarajan, G. (2001), “Ground Control Stations for Unmanned Air Vehicles”, DEF SCI J, 51(3).

3] Mark Draper1, Gloria Calhoun1, Heath Ruff2, David Williamson1 and Timothy Barry2, “Manual Versus Speech Input For Unmanned Aerial Vehicle Control Station Operation”, *Proceedings of the Human Factors & Ergonomics Society's 47th Annual Meeting,* October, 2003, pp. 109-113 (Clearance ASC:03-1666, 19 Jun 2003).

4] Kumar, P. Sathish, Suraj, S. Subramanian, R. Venkata, Raghavan and Vinay .V, “Voice Operated Micro Air Vehicle*”, International Journal of Micro Air Vehicles*, Jun2014, Vol. 6 Issue 2, p129.

5] *Defense Advanced Research Projects Agency.* Submitted to the Subcommittee on Terrorism,Unconventional Threats and Capabilities House Armed Services Committee United States House of Representatives, March 13, 2008.

6] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition”, *Proceedings of the IEEE*, Vol.77, No.2, pp.257-286, February 1989.

7] Young, S. Evermann, G. Gales, M. Hain, T. Kershaw, D. Liu, X. Moore, G. Odell, J. Ollason, D. Povey, D. Valtchev and V. Woodland, *The HTK Book Version 3.4,* Cambridge University, Cambridge 2006, Available on the World Wide Web: *http: ==htk.eng.cam.ac.uk.*

8] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, No. 4, pp. 357-366, 1980.

[9] Mikael Nilsson and Marcus Ejnarsson, “Speech Recognition using Hidden Markov Model”*,* Degree of Master of Science in Electrical Engineering, Department of Telecommunications and Signal Processing, Blekinge Institute of Technology Ronneby,March 2002.

10] Nicolas Moreau, *HTK (v.3.1): Basic Tutorial*, 02.02.2002.