ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Speech Operated System using DSP: A Review

Jeevanesh J. Chavathe[*1], P. V. Thakre[2]
[*1,2] Electronics and Telecommunication Dept., SSBT's COET Bambhori, Dist:Jalgaon, India
jeevanesh@gmail.com

Abstract

Speech is the most basic, common and efficient form of communication method for people to interact with each other. Today, speech technologies are commercially available for an unlimited but interesting range of tasks. These technologies enable machines to respond correctly and reliably to human voices, and provide useful and valuable services. This paper gives an overview of implementation of speech recognition using Digital Signal Processor (DSP) and comparison between DSPs used in.

Keywords: Speech recognition, Feature Extraction, Hidden Markov Model, MFCC, Speech operated systems.

Introduction

Speech recognition is vast and complex concept. This means to recognise the human speech and respond to spoken commands by machines or speech operated systems. This is more commonly known as automatic speech recognition (ASR) [4]. The goal of speech recognition is for a machine to be able to "hear", "understand" and "act upon" spoken information.

This paper is organized as follows. Section 2 presents the various methods used in speech recognition systems and Section 3 explains digital signal processors used in this system. Section 4 explains about various tools used in this system. Finally, the conclusion is summarized in Section 5 with future work.

Various Methods used in Speech Recognition

The automated recognition of human speech is immensely more difficult than speech generation. Speech recognition is a classic example of things that the human brain does well, but digital computers do poorly [2]. In general, it can be associated with three different fields: automatic, robust and distant speech recognition (DSR) [3]. The construction of optimal DSR systems must draw on concepts from several fields, including acoustics, signal processing, pattern recognition.

Obtaining the acoustic characteristics of the speech signal is referred to as Feature Extraction. Feature Extraction is used in both training and recognition phases.

It comprise of the following steps [2]: 1) Frame Blocking, 2) Windowing, 3) FFT (Fast Fourier Transform), 4) Mel-Frequency Wrapping, 5) Cepstrum (Mel Frequency Cepstral Coefficients).
The schematic diagram of the steps is depicted in Figure 1.

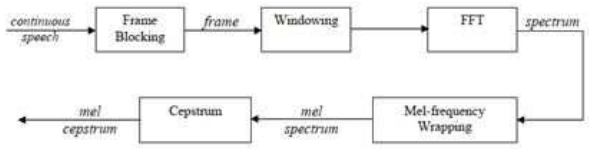

Figure.1 Feature Extraction Steps

Speech signals are processed in short time intervals. It is divided into frames with sizes generally between 30 and 100 milliseconds. Each frame overlaps its previous frame by a predefined size. The goal of the overlapping scheme is to smooth the transition from frame to frame.

The second step is to window all frames. This is done in order to eliminate discontinuities at the edges of the frames. If the windowing function is defined as w(n), 0< n< N-1 where N is the number of samples in each frame, the resulting signal will be y(n) = x(n)w(n). Generally hamming windows are used.

The next step is to take Fast Fourier Transform of each frame. This transformation is a fast way of Discrete Fourier Transform (DFT) and it changes the domain from time to frequency.

The Mel-Scale (Melody Scale) filter bank which characterizes the human ear perceiveness of frequency. It is used as a band pass filtering for this stage of identification. The signals for each frame is passed through Mel-Scale band pass filter to mimic the human ear.

As of the final step, each frame is inverse Fourier transformed to take them back to the time domain. Instead of using inverse FFT, Discrete Cosine Transform is used as it is more appropriate. The discrete form for a signal $x(n)$ is defined as,

ISSN: 2277-9655
Impact Factor: 1.852

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\Pi(2n-1)(k-1)}{2N}$$

$$w(k) = \begin{cases}\sqrt{1/N}, & k = 1\\ \sqrt{2/N}, & 2 \le k \le N\end{cases}$$

x : original signal,
y : Resulting Discrete Cosine Transformed signal,
N: number of samples.

A. Hidden Markov Model (HMM)

Hidden Markov Model, is doubly stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce sequence of observed symbols [5]. HMM which provides a highly reliable way for recognizing speech. The system is able to recognize the speech waveform by translating the speech waveform into a set of feature vectors using Mel Frequency Cepstral Coefficients (MFCC) technique [4].

Let each spoken word be represented by a sequence of speech vectors or observations O, defined as [6],

$$O = o_1, o_2, \ldots\ldots, o_t$$

where o_t is the speech vector observed at time t. The isolated word recognition problem can then be regarded as that of computing,

$$\arg\max_i \{P(w_i \mid O)\}$$

where w_i is the i'th vocabulary word. This probability is not computable directly but using Bayes' Rule gives,

$$P(w_i \mid O) = \frac{P(O \mid w_i)P(w_i)}{P(O)}$$

Thus, for a given set of prior probabilities $P(w_i)$, the most probable spoken word depends only on the likelihood $P(O \mid w_i)$. Given the dimensionality of the observation sequence O, the direct estimation of the joint conditional probability $P(o_1, o_2, \ldots, \mid w_i)$ from examples of spoken words is not practicable.

However, if a parametric model of word production such as a Markov model is assumed, then estimation from data is possible since the problem of estimating the class conditional observation densities $P(O \mid w_i)$ is replaced by the much simpler problem of estimating the Markov model parameters.

In HMM based speech recognition, it is assumed that the sequence of observed speech vectors corresponding to each word is generated by a Markov model as shown in Figure 2.

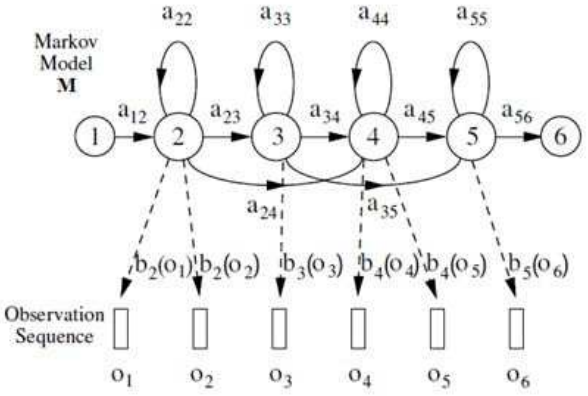

Figure.2 The Markov Generation Model

Given that X is unknown, the required likelihood is computed by summing over all possible state sequences X = $x(1), x(2), x(3), \ldots$, x(t), that is,

$$P(O \mid M) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)}$$

where $x(0)$ is constrained to be the model entry state and $x(T + 1)$ is constrained to be the model exit state. As an alternative to above equation, the likelihood can be approximated by only considering the most likely state sequence, that is,

$$P(O \mid M) = \max_X \left\{ a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)} \right\}$$

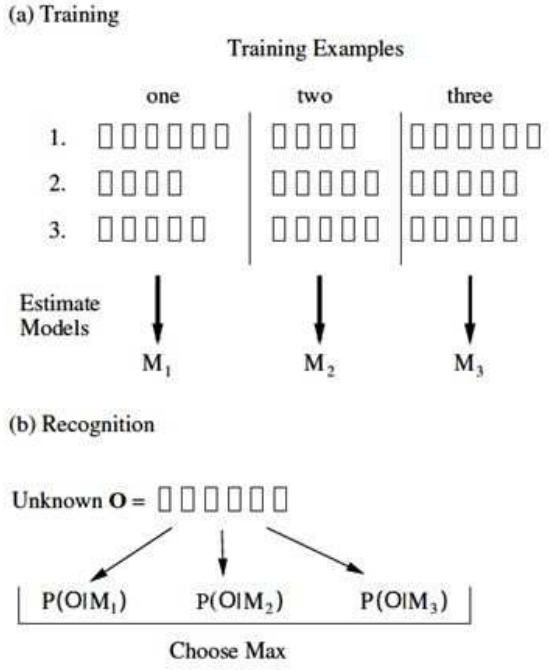

Figure.3 HMMs for Isolated Word Recognition

ISSN: 2277-9655
Impact Factor: 1.852

Figure 3 summarises the use of HMMs for isolated word recognition [6].

The main steps to perform the HMM based speech recognition system as follows [4]:

1. Receiving and digitizing the input speech signal.
2. Extracting features for all input speech signals using MFCC algorithm, and then converting and storing each signal's features into a feature vector.
3. Classifying the feature vectors into the phonetic based categories at each frame using HMM algorithm.
4. Finally, performing a Viterbi search which is an algorithm to compute the optimal (most likely) state sequence in HMM given a sequence of observed outputs.

Digital Signal Processors Used In Speech Recognition

The speech recognition can be implemented using DSP Processor Analog Devices's ADSP2181 [7] or Texas Instrument's TMS320C6713 [8].

A. Analog Devices's ADSP2181

Technical Specification of ADSP2181 are:

1) Processor ADSP2181 16-bit fixed point CORE operating at 5V, Internal memory DM-16KWords (16bits), PM- 6KWords(24bits), External memory-DM-16K Words (16-Bits) PM-16k Words (24-bits), Clock-24.576MHz, 2)UART-16C550 (19200 Baud rate used) 3) CODEC-HD44233 4) Power Supply-SMPS 5V, 500mA with EMI filter.

Wide range of problems in accuracy arise when common automatic speech recognition systems are tested under operating conditions like Noise level, Distance between a speaker and a microphone, different speaker. Table I gives the accuracy of the implemented system under various conditions [7].

Table I System accuracy under various conditions of noise, speakers, microphone distance.

Training	Testing	Speaker	Mic. Distance	Accuracy %
L	L	Same	Less than 5cm	95
M	M			85
H	H			65
L	H			75
L	L	Different	Less than 5cm	85
L	L	Same	Less than 5cm	95
L	L		More than 5cm	50

L, M, H are the Low, Medium, High levels of noise.

On the basis of performance analysis carried out for system using ADSP2181 [7], it is concluded that the accuracy of the system is high when the system is trained and tested in the silent room, with microphone distance of less than 5cm.

B. Texas Instrument's TMS320C6713

Technical Specification of TMS320C6713 [9] are: 1) Highest-Performance Floating-Point DSP, Eight 32-bit Instructions/cycle, 32/64-bit Data Word, 225-, 200-MHz (GDP), and 200-, 167-MHz (PYP) Clock Rates, 2) Advanced Very Long Instruction Word (VLIW), Load-Store Architecture With 32 32-Bit General-Purpose Registers, 3) 32-Bit External Memory Interface (EMIF), 4) L1/L2 Memory Architecture, 4K-Byte L1P Program Cache (Direct-Mapped), 4K-Byte L1D Data Cache (2-Way) 4) Power Supply - 3.3-V I/Os, 1.2-V+ Internal (GDP & PYP).

The Speech recognition system explained in paper [8] used MFCC based approach is also compared with Cochlear based approach. Table II reports the recognition performance of the system for various combinations of training and testing datasets.

Table II Recognition accuracy for various combinations of Training and Testing Datasets [8].

Feature Input ↓	No. of Training Datasets → No. of Testing Datasets ↓	5	10	15	20
Using MFCC	5	78.0%	88.0%	92.0%	62.0%
	10	77.0%	90.0%	92.0%	65.0%
	15	75.3%	90.0%	93.3%	66.0%
	20	75.5%	91.5%	93.0%	69.5%
	25	73.6%	89.2%	93.2%	-
	30	72.0%	88.0%	-	-
	35	71.1%	-	-	-
Using Cochlear	5	92.0%	96.0%	100%	86.0%
	10	94.0%	97.0%	98.0%	82.0%
	15	92.7%	97.3%	98.7%	81.3%
	20	90.5%	97.0%	98.0%	81.5%
	25	88.0%	96.8%	98.0%	-
	30	88.0%	97.3%	-	-
	35	89.4%	-	-	-

It is observed from Table II that the recognition accuracy for Cochlear based approach is better when compared with MFCC features for all the combinations.

Table III Comparison between the recognition systems Implemented [8].

Details	MFCC Feature	Cochlear Feature
DSP Device	TMS320C6713	TMS320C6713
Clock Frequency	225MHZ	225MHZ
Memory Used (in Kilobytes)	511	284
Execution Time (No. of Clock cycles)	$167x10^6$	$345x10^6$
Recognition Accuracy	93.33%	98.67%
Number of Support Vectors	814	783

ISSN: 2277-9655
Impact Factor: 1.852

The hardware resources utilized and the performance of the real-time speech recognition system for MFCC feature inputs and feature inputs using Cochlear are reported in Table III.

It may be observed from Table V that the memory required for coding MFCC features is more than that required for Cochlear, whereas the execution time with MFCC features is lesser than that of Cochlear based approach [8].

Various Tools Used In Implementation Of Speech Recognition System

A. Approach – I : Using Matlab Simulink.

Matlab Simulink provides an exploratory and verification tool for both floating-point and fixed-point DSP systems and applications [10]. Simulink provides an environment where you model your physical system as a block diagram. You create the block diagram by using a mouse to connect blocks and a keyboard to edit block parameters [11].

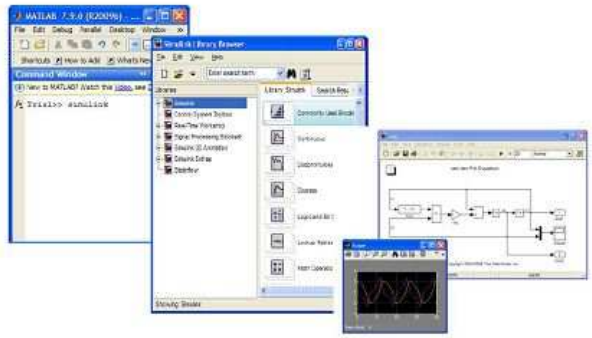

Figure.4 Launching Simulink

Steps to launch and use Simulink :

1. First of all, launch Matlab. Press the Simulink button or write "Simulink" in the command window as shown in figure 4.
2. You will see the Simulink window.
3. Open a new Simulink window by clicking on the New button.
4. You will see the blank workspace window.
5. Select a block from Simulink Library Browser and drag by mouse in blank workspce.
6. Connect the blocks with signals using mouse.
7. Also you can change simulation parameters by double click on block and Set.
8. Double click on the Scope to view the Simulation results.
9. In the model window, from the Simulation pull-down menu, Select Start to run the simulation and view the output in the Scope window.
10. Save the model by .mdl suffix, from the File pull-down menu and select Save.

B. Approach – II : Using Code Composer Studio.

The Code Composer Studio (CCS) software comes with the DSP Starter kit (DSK) and is used to download programs into DSK [1]. CCS includes tools for code generation, such as a C compiler, an assembler, and a linker.

The C compiler compiles a C source program with extension .c to produce an assembly source file with extension .asm. The assembler assembles an .asm source file to produce a machine language object file with extension .obj. The linker combines object executable file that can be loaded and run directly on C6713 DSP.
CCS or Code Composer studio by TI is the IDE we used for the Development Environment [12]. The main features of it are given in figure 5 :

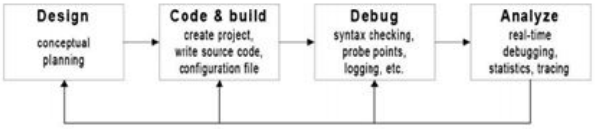

Figure.5 Features of CCS

Steps to launch and use Code Composer Studio :

Creation of Project : First of all launch CCStudio as shown in figure 6. Press File → New Project. Then give Location & Name to it and choose the target as 67XX. We then create new source file from File → New → Source File. We save it in the projects directory. We can also add C files to the project.

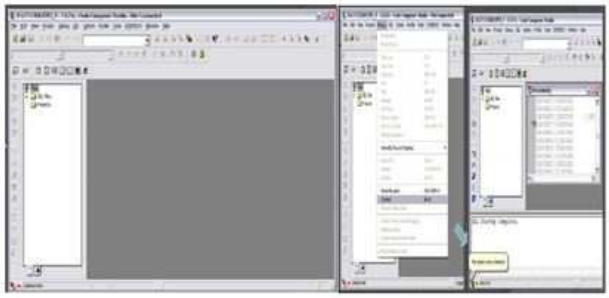

Figure.6 Launching CCStudio

Adding Libraries and Support Files : We add the following files to the project
C6713.cmd – Linker command file, we add from the support folder.
rts6713.lib – Run time support Library from lib folder under C6000 folder.
dsk6713 – DSK board support library from lib folder under C6000 folder.

ISSN: 2277-9655
Impact Factor: 1.852

csl6713 – Chip support library from lib folder under C6000 folder.
Vector_poll.asm or Vector_interrupt.asm – Interrupt as required.

Setting the Build Options : Select Build Option under project menu. Change the Options as given below in the compiler tab. Change the memory model to far and rts call to are far as shown in Figure 7. We change the option in the linker menu to accommodate the linker library files as shown in Figure 8.

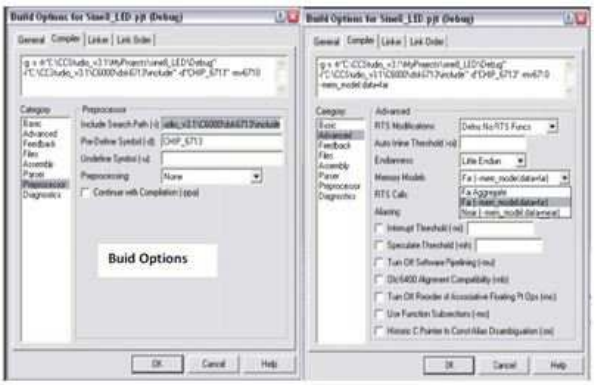

Figure.7 Setting the Build Options

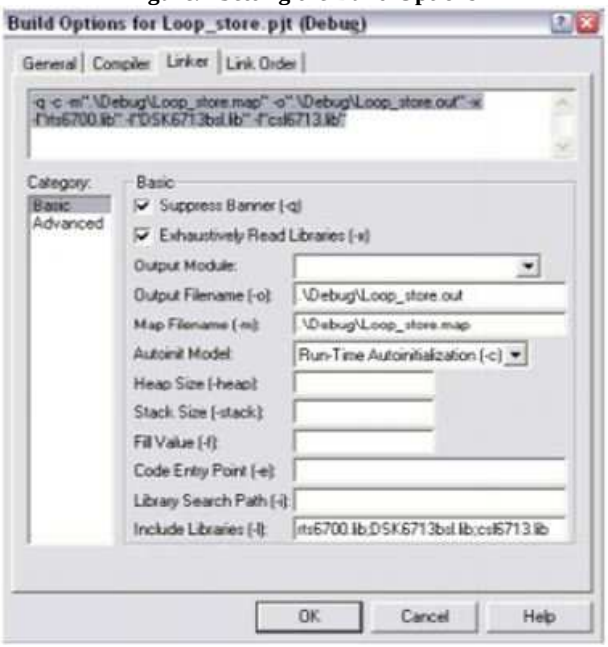

Figure.8 Changing the option in Linker menu

Compiling and Running the Program : Click Build from Project Menu. If error come the correct then rebuild all from project menu. After compiling a .out exec file will be created under a debug folder. It is dumped in the board by clicking Load Program from the File Menu. Else press ctrl+L to load the program on the board. Click Debug → Run to Run the program else click F5 to run. After running Click Halt from Debug Menu.

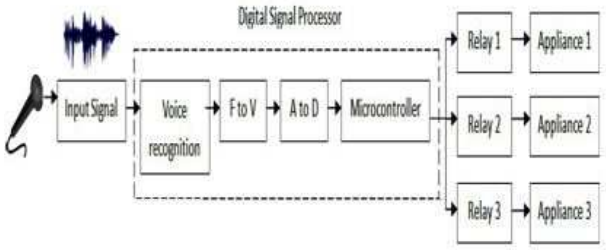

Figure.9 Proposed system model

In proposed system model, the voice command is given by microphone and this input signal (Speech Signal) is further proceed for voice recognition by DSP. Frequency to Voltage converter (F to V), Analog-to-Digital converter (A to D) and Microcontroller all are replaced by code implemented in DSP.

After recognising command from speaker, the home appliances toggle ON or OFF by using relays. All English digits from (Zero through Nine) are recognised and according to that spoken digit, the appliance switching either ON or OFF. For example, digit ONE (1) is allocated to Fan, spoken digit ONE is recognised and switching Fan ON; if again saying digit ONE, switching Fan OFF and vice versa. That means by single digit only we can switch ON or OFF the appliance.

References

[1] Deepali Y. Loni, "DSP Based Speech Operated Home Appliances Using Zero Crossing Features" Signal Processing: An International Journal (SPIJ), Volume (6) : Issue (2) : 2012.
[2] Volkan Tunali, "A speaker dependent, large vocabulary,isolated word speech recognition system for turkish" , Istanbul 2005.
[3] Dr. Matthias Woelfel, Dr. John McDonough, "Distant Speech Recognition", Ch. 1, Page 10.
[4] Ahmad A. M. Abushariah, Teddy S. Gunawan, Othman O. Khalifa, Mohammad A. M. Abushariah, "English Digits Speech Recognition System Based on Hidden Markov Models", International Conference on Computer and Communication Engineering (ICCCE 2010), 11-13 May 2010, Kuala Lumpur, Malaysia.
[5] Bhupinder Singh, Neha Kapur, Puneet Kaur, "Speech Recognition with Hidden Markov Model: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012.

[6] Steve Young and Gunnar Evermannl. The HTK Book. Microsoft Corporation, 2005.
[7] Kalpana Joshi, Nilima Kolhare & V. M. Pandharipande, "Implementation of Speech Recognition System Using DSP Processor ADSP2181", International Journal of Electronics Signals and Systems (IJESS) ISSN: 2231- 5969, Vol-1 Iss-3, 2012.
[8] J. Manikandan, B. Venkataramani, K. Girish, H. Karthic and V. Siddharth, "Hardware Implementation of Real-Time Speech Recognition System using TMS320C6713 DSP", 24th Annual Conference on VLSI Design, 2011.
[9] Texas Instruments, TMS320C6713 Floating-Point Digital Signal Processor Datasheet, SPRS186L − December 2001 − Revised November 2005.
[10] Woon S. Gan and Sen M. Kuo, "Transition from Simulink to MATLAB in Real-Time Digital Signal Processing Education".
[11] The Mathworks, DSP Blockset for use with Simulink, User's Guide, Version 5, 2003.
[12] R. Chassaing, DSP Applications Using C and the TMS320C6x DSK, Wiley, New York, 2002.