DSP Based Speech Operated Home Appliances Using Zero Crossing Features

Deepali Y Loni *deepaliloni@rediffmail.com*
Assistant Professor
Department of Electronics
D.K.T.Es' Textile & Engineering Institute,
Ichalkaranji. Post code:416115, India.

Abstract

The main idea of this paper is to build a simple speech recognition system using Digital Signal Processor (DSP) that controls home appliances (i.e. turning on/off) by processing the spoken word. The method used is simple, involving a plain count of the frequency of zero crossings. Two features of zero-crossing are used namely: maxima & running sum that increases the accuracy of recognition. The DSP calculates the zero crossings of the spoken words and accordingly generates different analog signals at its output. These analog signals are further processed so as to operate the appliances. The words chosen for recognition are 'ONE', 'TWO' and 'THREE'.

The paper includes two approaches for implementation of speech recognition into DSP, using Matlab Simulink approach and secondly using Code Composer Studio (CCS). Moreover the first approach performs offline processing and the other performs real time processing of words. The results at the end describe the efficiency of the system.

Keywords: Speech Recognition, Matlab, Simulink, Zero Crossing.

1. INTRODUCTION

Speech recognition is a vast topic of interest and is looked upon as a complex problem. In a practical sense, speech recognition solves problems, improves productivity, and changes the way we run our lives. Reliable speech recognition is a hard problem, requiring a combination of many techniques; however modern methods have been able to achieve an impressive degree of accuracy [1]. Real-time digital signal processing made considerable advances after the introduction of specialized DSP processors. Suitable DSP Starter Kits, with specific DSP processor and related software tools such as assemblers, simulators and debuggers are available to make system design and application development easier. Digital Signal Processor TMS320C6713 enables to design a system with very high computational power and large memory space with minimal count of components what safes printed circuit board space and simplifies design [2,3].

Talking to appliances in a home has been a science fiction staple for decades. This paper proposes solution for providing speech interface to electronic devices in a home. The TMS320C6713 digital signal processor (DSP) and Microcontroller are used in this work. The DSP is used for implementing speaker dependent speech recognition system for capturing vocal commands for operating the appliances and the Microcontroller serves as the main interface between DSP and the control circuit handling the appliances.

2. THE PROPOSED SYSTEM MODEL

The system model of **Figure 1** consists of two parts. First part informs about interfacing and processing of the spoken word using floating point DSP Starter kit (DSK) TMS320C6713. DSK is used for the research module because it provides an efficient and stable DSP development environment and it is a robust, low-cost and easily available DSK in both universities and industry [4].

The second part consists of control circuitry that consists of hardware that operates the appropriate appliance through the command recognized by DSK.
The system model consists of:

2.1 The TMS320C6713 Processor for Speech Recognition

The system model uses Texas Instrument TMS320C6713 DSP Processor to perform the task of speech recognition. The Processor first filters the noise from input speech. It then processes and identifies the spoken word. For the identified word, the Processor generates an analog sinusoidal signal of certain frequency. For example, for the audio command 'ONE', the Processor generates a sinusoidal signal of frequency 400Hz. Thus sine waves of different frequencies are generated at output of Processor for each spoken word 'ONE', 'TWO' and 'THREE' as shown in the **Table 1**.

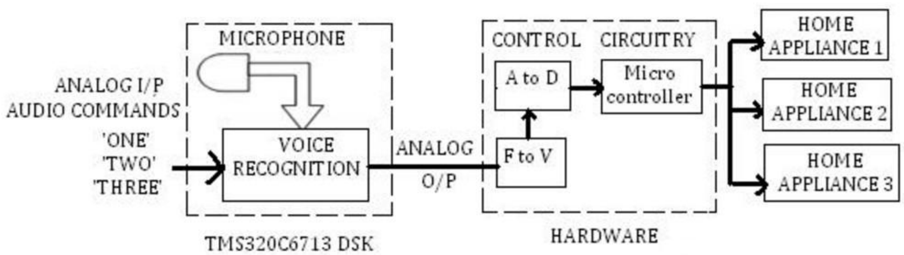


FIGURE 1: The proposed system model

Input audio command	Frequency of the analog signal generated at the output of processor
ONE	400Hz
TWO	800Hz
THREE	2000Hz

TABLE 1: Output of Processor

2.2 The Hardware Control Circuitry

The control circuitry consists of Frequency to Voltage (F to V) converter, Analog to Digital (A to D) converter, AT89C52 Microcontroller, relay drivers and relays.

The analog signals at the output of Processor are passed to F to V converter. The audio commands 'ONE',' TWO' and 'THREE' are mapped into three different voltage levels by F to V. These voltage levels are then digitalized by ADC0804. AT89C52 Microcontroller operates the appropriate relays scanning its digital input. These relays then in turn operate the corresponding appliances like; lamps, fan, dishwashers, clothes washers, dryers, microwaves, refrigerators, freezers, etc., switching them either ON or OFF. For example if the spoken word is 'ONE', then home appliance one will toggle (i.e. if it is ON then it will turn OFF and vice versa), if say the spoken word is 'TWO' then home appliance two will toggle.

3. PREPARATION OF DATABASE

In the first part of this work, Matlab is used to prepare database for offline processing of the words. These words are first recorded by means of microphone using wavrecord command in Matlab. The Matlab code is discussed below:

```
fs = 8000;
Y=wavrecord (5632, fs, 1, 16);
wavwrite(y, fs,16,'E:\database\three\three10');
```

It records 5632 samples of an audio signal, sampled at a rate of 8000 Hz using channel number 1of input channels from the audio device each of 16 bits. The speech is recorded for 5632 / 8000 = 0.704 seconds which is enough time to say a complete word. The Matlab command wavwrite is used to write the recorded data to .wav file. Ten sample words are stored for each audio command 'ONE', 'TWO' and 'THREE'.

4. USING ZERO-CROSSING FEATURE

The feature used to differentiate between the audio commands; 'ONE', 'TWO' and 'THREE' is zero crossings in the words. Zero-crossing rate is a measure of the number of time in a given time interval that the amplitude of the speech signal passes through a value of zero [1]. It means, a zero-crossing is said to have occurred in a signal when its waveform crosses the time axis or changes its algebraic sign. This feature has been used heavily in both speech recognition and music information retrieval.

5. METHODS ADOPTED FOR IMPLEMENTATION OF SPEECH RECOGNITION

The speech recognition using zero crossing features is implemented in this paper using two approaches.

5.1 Approach I: Implementation of Speech Recognition Using Matlab Simulink.

The Mathwork's Simulink is used to implement speech recognition system. There are many advantages in programming DSP algorithms using Matlab. These include ease of coding, able to use a powerful set of inbuilt functions and seamless link between Matlab and Simulink [5]. The use of Simulink enables the creation of sophisticated algorithms in an intuitive top-level design [6].

Figure 2 shows the complete Simulink model, build for the recognition of the three audio commands: ONE', 'TWO' and 'THREE'. The model recognizes the sample words from database prepared.

The block From Wave File reads data from stored sample words at a rate of 256 samples per frame. As the audio is recorded at 8 KHz, each sample word gets broken into 22 frames.

The Digital Filter Design block implements a Low Pass FIR Filter to filter the noise from the input signal. An offset of value 0.05 is added to the input signal before computing the zero crossing, in order to avoid the zero crossings due to signal noise. The zero crossing block outputs the number of times the signal crosses zero at its output port. The model further computes two features from the zero crossing block. They are: maximum value of zero crossings count from all 22 frames and the sum of zero crossings count in all 22 frames of the spoken word, using the Maximum and Running Sum blocksets. These computed features differ for every word and hence help to exactly identify the spoken word.

The Simulink model to understand the output of Maximum and Running Sum blockset for the word 'ONE' is shown in **Figure 3** and **4.** The Maximum block finds out frame with maximum zero crossings. Hence **Figure 3** displays the highest zero crossings from the 22 frames of the word. The running sum block computes the zero crossings in entire word by summing the zero crossings in all 22 frames. Hence **Figure 4** displays the sum of the zero crossings in the word 'ONE'.

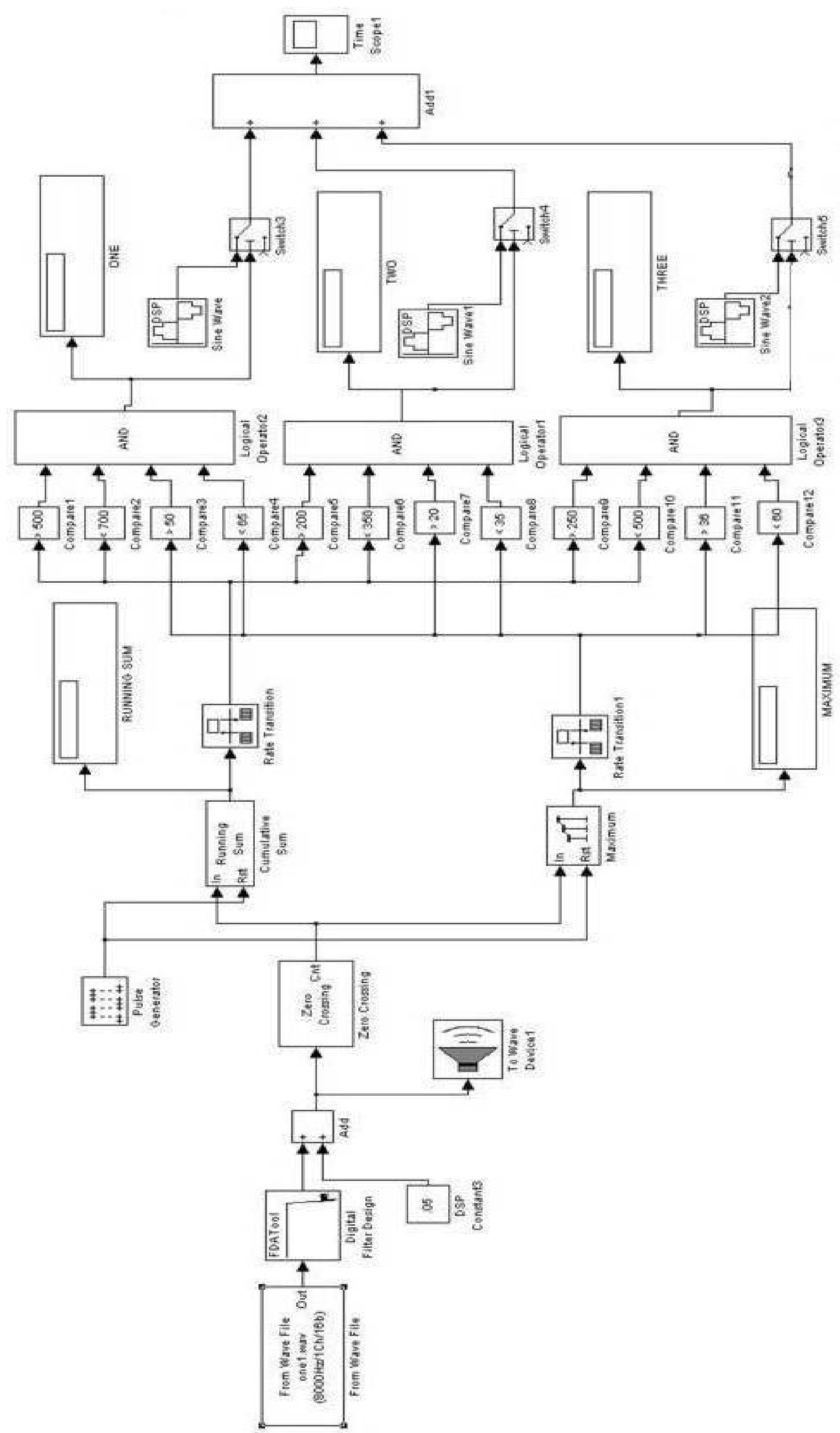


FIGURE 2: The Simulink Model recognizing commands 'ONE','TWO', &'THREE'.

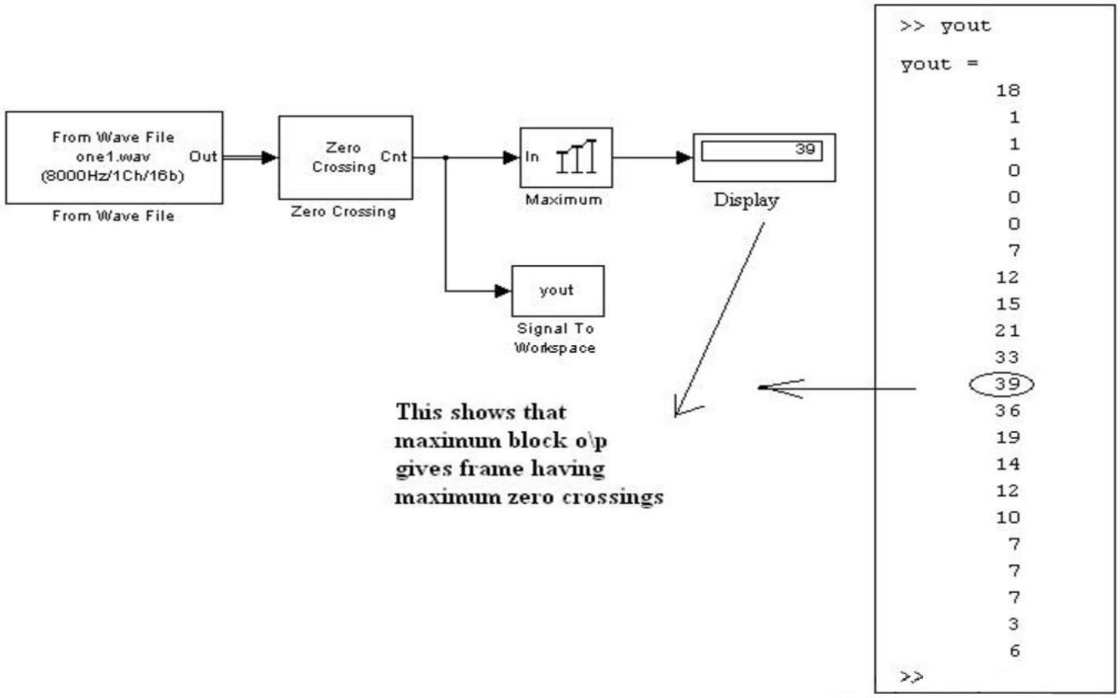


FIGURE 3: Maximum Block output of the Simulink Model

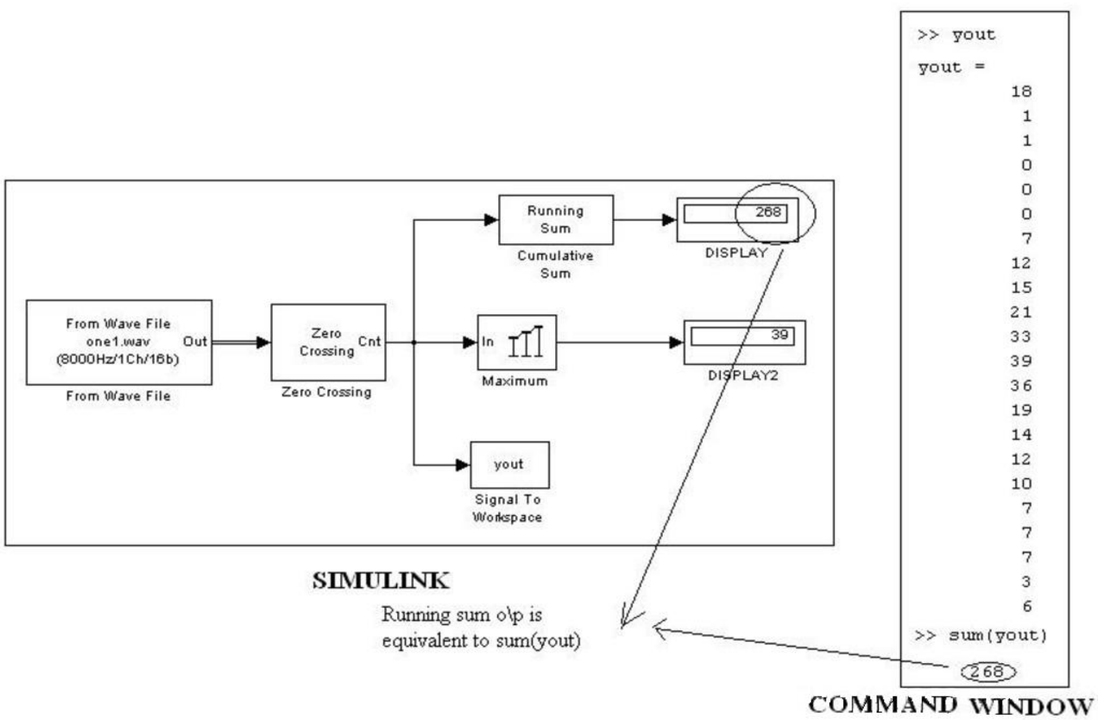


FIGURE 4: Running Sum Block output of the Simulink Model

Audio command	Simulink Blocksets			
	Maximum		Running sum	
	Min	Max	Min	Max
ONE	50	65	500	800
TWO	20	35	200	350
THREE	36	60	250	500

TABLE 2: Minimum & Maximum values of Maximum and Running Sum Blocks

These features are calculated for all the recorded commands in the database and it is observed that, each audio command lie within certain range of minima and maxima as shown in **Table 2.**

Table 2 shows that, the total zero crossings for audio command 'ONE' as computed by running sum block lies between 500-800 and maximum zero crossing observed in a frame lie between 50- 65. While for audio command 'TWO' and 'THREE' the cumulative zero crossings is between 200-350 and 250-500, and maximum zero crossing observed in a frame lie between 20-35 and 36-60 respectively. In addition Table **2** also reveals that the maximum zero crossing value for the audio command 'ONE' & 'THREE' have a common range, but their running sum ranges differ. Accordingly, the Simulink model of **Figure 2** is designed that check all states and accurately identify the command 'ONE'.

5.2 Approach II: Implementation of Speech Recognition 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 [8, 9]. 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 supports C-language coding. The C program for implementation of speech recognition algorithm in CCS is explained in the form of flowchart in **Figure 5.**

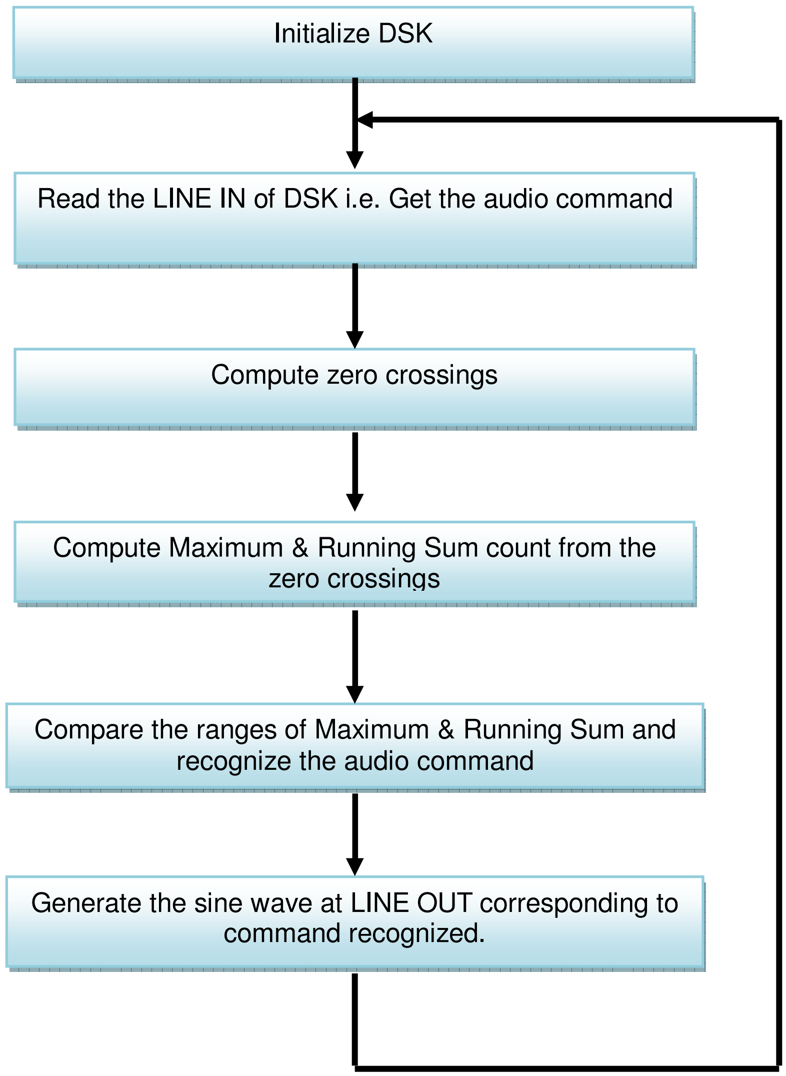


FIGURE 5: Flowchart of C program for implementation of speech recognition algorithm in CCS

6. EMBEDDING SPEECH RECOGNITION ALGORITHM INTO TMS320C6713 DSK

For standalone working of DSK TMS320C6713, the speech recognition algorithm must be embedded in its flash memory. As the paper proposes two approaches of speech recognition implementation, the methods of interfacing these algorithms to DSK are discussed below:

6.1 Embedding Matlab Simulink Speech Recognition Algorithm Into DSK.

With the advent of Matlab's Real-Time Workshop (RTW) it is possible to compile, load, and execute graphically designed Simulink models on an actual DSP plat-form.

Simulink uses graphical block diagrams to create models for real-time implementation of applications and then use Real-Time Workshop to generate C code targeted to the TI DSP board by mean Code Composer Studio (CCS IDE) [7]. For downloading this Simulink model of **Figure. 2** into DSK, it is essential to include C6713 DSK Board Support Library Blocks in the model so as to establish a communication with the codec of target C6713 DSK [10-13].

Steps involved in building and executing the Simulink model on C6713 DSK:
1) Configure Simulink parameters and configure Real-Time Workshop.
2) Run the CCS software in background.
3) Open the Simulink model of speech recognition and press CNTRL + B to build an equivalent 'C' language code in CCS. Simulink starts communicating with CCS and generates the C code that can be run by the C6713 DSP on DSK.
4) To stop model execution, click the Reset DSK block or use the Halt option in CCS.

6.2 Embedding the CCS Speech Recognition Algorithm Into DSK.

The approach using Simulink performs processing on stored audio words. For real time processing of audio words, this approach includes Real Time Data Exchange (RTDX). RTDX provides real-time, continuous visibility into the way target applications operates in the real world. RTDX allows transfer of data between the host computer and DSP devices without stopping target application.

Figure 6 illustrates how realtime audio signal can be passed on to DSK using RTDX. The audio command is recorded with PC-based audio input device i.e. microphone using wavrecord command of Matlab (running on the host PC) in real time. The array of recorded data is then sent to C6713 processor through RTDX. The C program for implementation of speech recognition algorithm (**Figure 5)** is transformed to executable file and is loaded on the C6713 DSP using CCS tools. The C source program (running on the DSK) then calculates the frame wise zero crossings and running sum of zero crossings. Depending upon its value, the DSK recognizes the spoken word 'ONE', 'TWO' or 'THREE' and sends the respective sine wave of particular frequency on LINE OUT port of DSK.

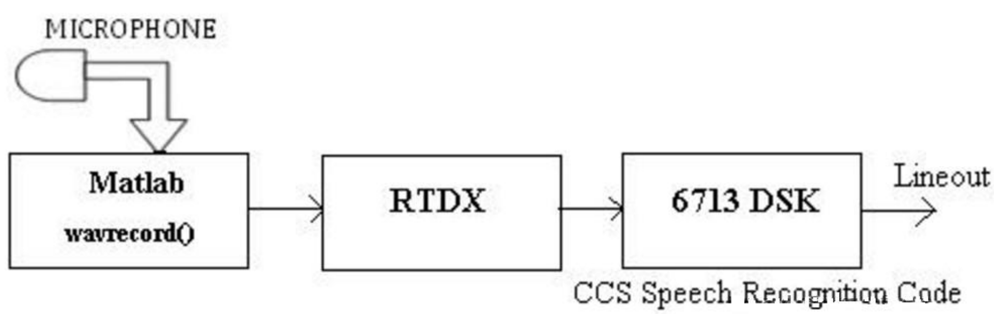


FIGURE 6: Interface between real time audio command and DSK using RTDX.

7. HARDWARE INTERFACE TO C6713 DSK

Once the DSK recognizes the spoken words, it generates a sinusoidal signal of a fixed frequency assigned to each word from the LINE OUT port. The hardware circuit interfaced to DSK operates the home appliances accordingly. The Block Diagram of the hardware circuit is shown in **Figure 7**.

As DSK outputs sinusoidal signals of fixed frequency, the first block in the hardware circuit is an F to V converter. As per the digital signals received at its input port, Microcontroller operates the relay. As the Microcontroller cannot provide sufficient current to drive relays, a relay driver is essential interface between the two. The relays in turn operate the home appliances.

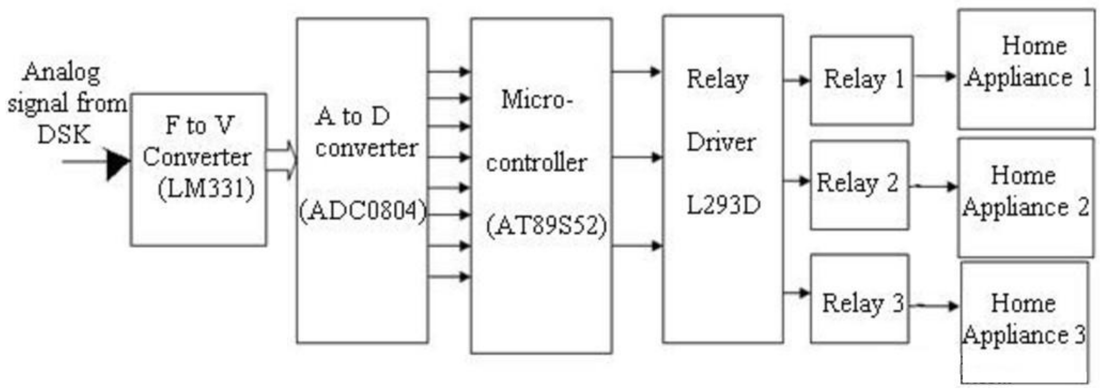


FIGURE 7: Block diagram of Hardware circuit interfaced to DSK

8. RESULTS

8.1 Simulink Result

The Simulink result for the audio command 'THREE' is shown in **Figure 8**. In **Figure 8,** the model processes the audio command 'THREE', generating the Running Sum value of 450 and Maximum value of 40. These values exactly lie in the ranges mentioned in **Table 2** that rightly identifies the audio command 'THREE'.

8.2 The DSK Result

The DSK generates sinusoidal signals at its LINE OUT port for every word recognized. The sinusoidal signals of 400Hz and 2000Hz obtained at the LINE OUT port for the audio command 'ONE ' and 'THREE' are shown in **Figure 9**.

8.3 Results Obtained From Hardware Circuit

Table 3 shows the output of F-V converter and the ADC ranges generated by the hardware in response to the word recognized by the DSK. The audio commands 'ONE', 'TWO' and 'THREE' are mapped into three different voltage levels by F to V converter. Accordingly, the ADC values lie in separate ranges. The Microcontroller operates the correct relay, which in turn switches the corresponding appliance.

9. CONCLUSION

The paper presents two approaches for implementing speech recognition algorithm; using Matlab Simulink approach and secondly using Code Composer Studio (CCS) using DSK TMS320C6713. The algorithms are based on zero crossing feature. The simulations as well as experimental results of the hardware circuit are included. These results indicate that the home appliances can be operated reliably with voice commands. The proposed method also finds promising applications in robot control and helpful for industries where there is immense danger in operating the system manually.

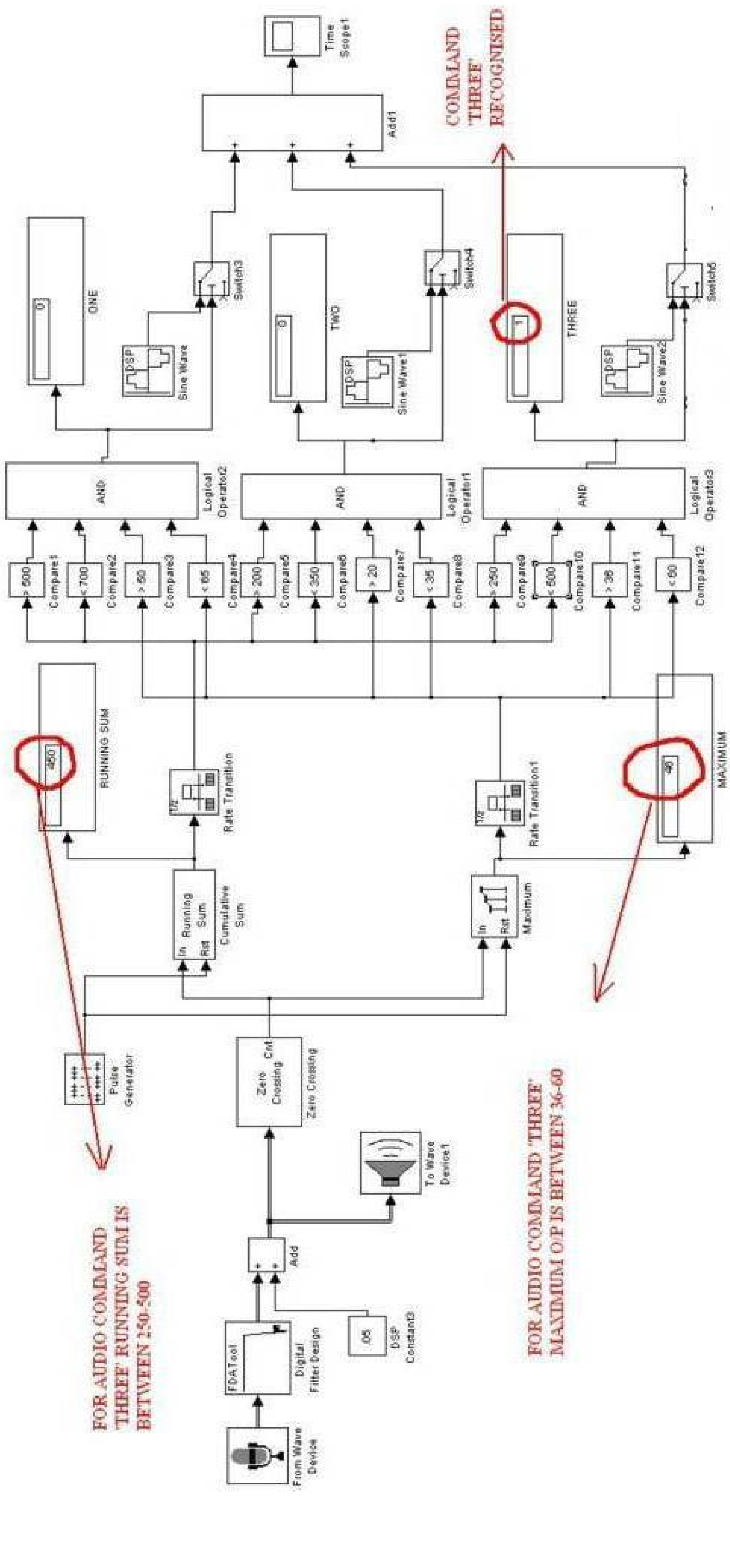
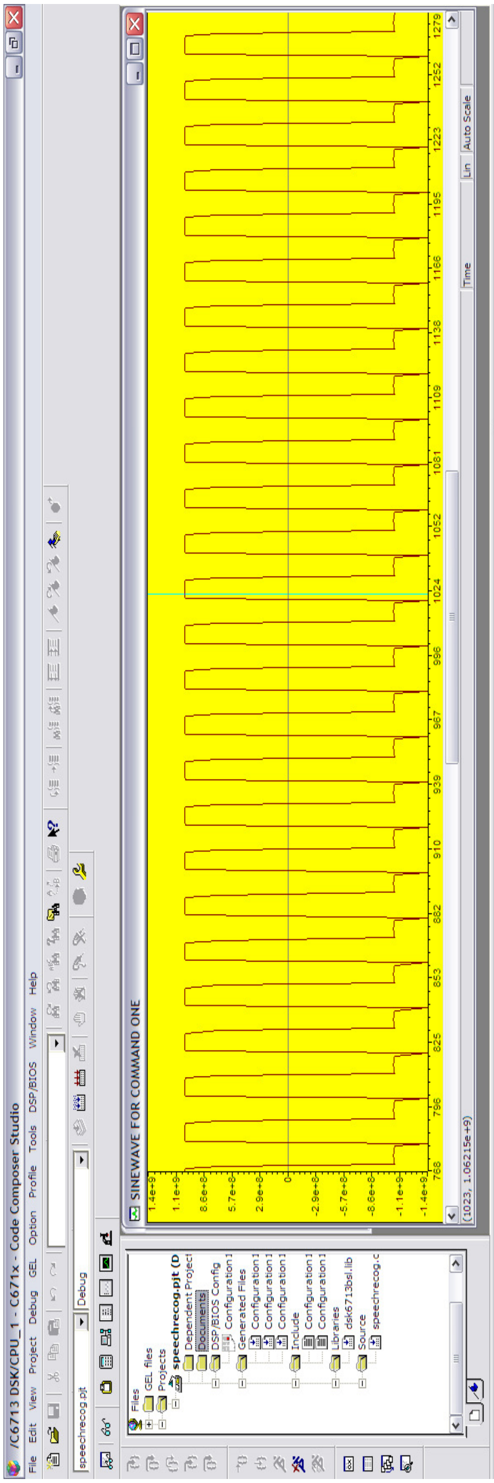
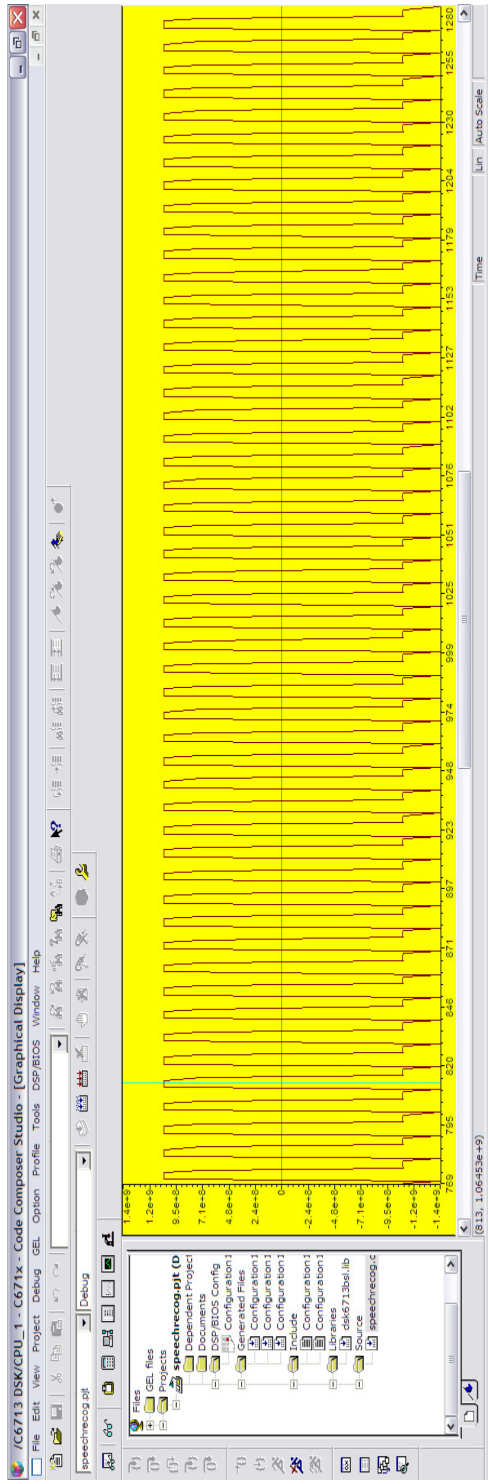


FIGURE 8: Result of Simulink model for the audio word 'THREE'



(a)



(b)

FIGURE 9: Plots of sinusoidal signals generated at the output of DSK. . (a) For audio command 'ONE'. (b) For audio command 'THREE'.

Audio command	Sine Wave Frequency	F to V Converter output	ADC Range
ONE	400Hz	1.10V	0x30-0x3F
TWO	800Hz	2.65V	0x80-0x8F
THREE	2000Hz	4.04V	0xC0-0xDF

TABLE 3: Results obtained from hardware circuit**.**

10. REFERENCES

[1] S. K. Hasnain, A. Beg and S. Awan, "*Frequency Analysis of Urdu Spoken Numbers Using MATLAB and Simulink*" Journal of Science & Technology PAF KIET ISSN 1994- 862x, Karachi, Dec. 2007.

[2] F.Wallam, M. Asif, "*Dynamic Finger Movement Tracking and Voice Commands Based Smart Wheelchair",* International Journal of Computer and Electrical Engineering, Vol. 3, No. 4, August 2011,pp.497 502.

[3] H. Chin, J.Kim, I.Kim, Y.Kwon, K.Lee, S. Yang, "Realization of Speech Recognition Using DSP(Digita1 Signal Processor)", IEEE International Symposium On Industrial Electronics Proceedings 2001, pp.508-512.

[4] T. J. Ahmad, H. Ali, M. A. Ajaz, S. A. Khan, "A DSK Based Simplified Speech Processing Module for Cochlear Implant", IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. http://dx.doi.org/10.1109/ICASSP.2009.4959597

[5] W. S. Gan1 and S. M. Kuo2, "*Transition from Simulink to MATLAB in Real-Time Digital Signal Processing Education*", Int. Journal Engineering Education Vol. 21, No. 4, pp. 587±595, 2005.

[6] S. Gannot,V. Avrin. "A Simulink *A*nd Texas Instruments C6713 Based Digital Signal Processing Laboratory", 14th European Signal Processing Conference, Italy, September 4-8, 2006.

[7] J. Zapata,R. Ruiz, "Rapid Development of Real-Time Applications UsingMATLAB/Simulink on TI C6000-based DSP", Proceedings of the 5th WSEAS International Conference on Education and Educational Technology, Tenerife, Canary Islands, Spain, December 16-18, 2006,pp.104-110.

[8] M. T. Qadri, S. A. Ahmed,"Voice Controlled Wheelchair Using DSK TMS320C6711", IEEE Computer Society, International Conference on Signal Acquisition and Processing 2009, pp. 217-220. http://dx.doi.org/10.1109/ICSAP.2009.48

[9] R. Chassaing, "*Digital Signal Processing and Applications with the C6713 and C6416 DSK*", New Jersey, A John Wiley & Sons, Inc., Publication, 2005.

[10] The MathWorks, Signal Processing Toolbox User's Guide, Version 6, 2003.

[11] The MathWorks, DSP Blockset User's Guide, Version 5, 2003.

[12] The MathWorks, Embedded Target for the TI TMS320C6000™ DSP Platform, Version1, 2002.

[13] Texas Instruments, C6713 DSP Starter Kit. Available at: http://focus.ti.com/docs/toolsw/folders/print/tmds 20006713.html220