We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



26

Ultimate Trends in Integrated Systems to Enhance Automatic Speech Recognition Performance

C. Durán
University of Pamplona
Colombia

1. Introduction

An automatic speech recognition (ASR) system can be defined as a mechanism capable of decoding the signal produced in the vocal and nasal tracts of a human speaker into the sequence of linguistic units contained in the message that the speaker wants to communicate (Peinado & Segura, 2006). The final goal of ASR is the man–machine communication. This natural way of interaction has found many applications because of the fast development of different hardware and software technologies. The most relevant are the access to information systems; an aid to the handicapped, automatic translation or oral system control. ASR technology has made enormous advances in the last 20 years, and now large vocabulary systems can be produced that have sufficient performance to be usefully employed in a variety of tasks (Benzeghiba et al., 2007; Coy & Barker, 2007; Wald, 2006; Leitch & Bain, 2000). However, the technology is surprisingly brittle and, in particular, does not exhibit the robustness to environmental noise that is characteristic of humans. Speech recognition applications that have emerged over the last few years include voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), domotic appliances control (e.g., "Turn on Lights" or "Turn off lights"), content-based spoken audio search (e.g., find a podcast where particular words were spoken), isolated words with a pattern recognition, etc. With the advances in VLSI technology, and high performance compilers, it has become possible to incorporate different algorithms into hardware. In the last few years, various systems have been developed to serve a variety of applications. There are many solutions which offer small-sized, high performance systems; however, these suffer from low flexibility and longer cycle-designed times. A complete software-based solution is attractive for a desktop application, but fails to provide an embedded portable and integrated solution.

Nowadays, High-end Digital Signal Processors (DSP's) from companies, such as; Texas Instruments (TI) or Analog Devices and High-performance systems like Field Programmable Gate Array (FPGA) from companies, such as; Xilinx or Altera, that provide an ideal platform for developing and testing algorithms in hardware.

The Digital signal processor (DSP) is one of the most popular embedded systems in which computational intensive algorithms can be applied. It provides good development flexibility

Open Access Database www.intechweb.org

Source: Speech Recognition, Technologies and Applications, Book edited by: France Mihelič and Janez Žibert,
ISBN 978-953-7619-29-9, pp. 550, November 2008, I-Tech, Vienna, Austria

and requires a relatively short application development cycle; therefore, the Automatic Speech Recognition on DSP Technology will continue as an active area of research for many years.

Speech recognition is a related process that attempts to identify the person speaking, as opposed to what is being said. The general-purpose speech recognition systems are generally based on Hidden Markov Models (HMM's). This is a statistical model which outputs a sequence of symbols or quantities. One possible reason why these models are used in speech recognition is that a speech signal could be viewed as a piece-wise stationary or short-time signal (Rabiner, 1989; Jou et al., 2001). Another of the most powerful speech analyses techniques is Linear Predictive Coding (LPC). The covariance analysis of linear predictive coding has wide applications, especially in speech recognition and speech signal processing (Schroeder & Atal, 1985; Kwong & Man, 1995; Tang et al., 1994). Real-time applications demand very high-speed processing, for example, for linear predictive coding analysis. An approach to acoustic modeling is the use of Artificial Neural Networks (ANN) (Lim et al., 2000). They are capable of solving much more complicated recognition tasks, but do not scale as well as LPC or HMM's when it comes to amplified vocabulary. Rather than being used in general purpose speech recognition applications, they can handle low quality, noisy data and speaker independence. Such systems can achieve greater accuracy like LPC or HMM's based systems, as long as there is training data and the vocabulary is limited. A more general approach using neural networks is phoneme recognition. This is an active field of research, but generally the results have been better than for LPC or HMM's. There are also LPC-ANN and HMM-ANN hybrid systems that use the neural network.

This chapter provides an overview of some applications with integrated systems, in order to improve the performance of the ASR systems. In the last section of this chapter the use of LPC-ANN hybrid system as an alternative in the identification of speech command and the implementation using Matlab® Released Software is described coupled with the DSP Hardware (DSK6416T Started Kit) developed by Texas Instruments.

2. VLSI technology

To learn the concept of integrated or embedded systems and their importance for ASR it is necessary to explain what Very Large Scale Integration (VLSI) technology is.

VLSI is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip (S. Kung, 1985; Barazesh et al., 1988; Cheng et al, 1991). VLSI began in the 70's when complex semiconductors were being developed. The processor is a VLSI device; however, the term is no longer as common as it once was since chips have increased into a complexity of millions of transistors. Today, in 2008, billions of transistor processors are commercially available; an example of which is the dual core processor called Montecito Itanium. This is expected to become more commonplace as a semiconductor.

The ASR has been an active area of research for many years; for this reason, with the advances in VLSI technology and high performance compilers, it has become possible to incorporate algorithms in hardware with great improvements in performance. In the last few years, various systems have been developed to cater to a variety of applications (Phadke et al., 2004; Melnikoff et al., 2002). Figure 1 shows an example of VLSI technology which possesses properties of low-cost, high-speed and massive computing capability; therefore, it is a suitable candidate in integrated Systems (e.g. The DSP) to enhance Automatic Speech Recognition Performance**.** Due to the fast progress of VLSI, algorithm-oriented architectural

array appears to be effective, feasible, and economical. Many algorithms can be efficiently implemented by array processors.

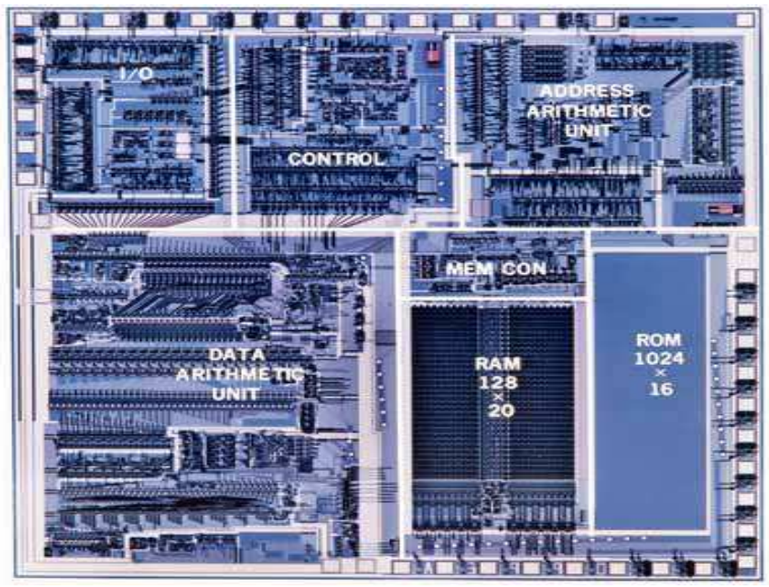


Fig. 1. DSP chip from VLSI (Bell Labs, device layout)

2.1 DSP chip and typical applications

This section describes the concepts of the DSP's device which can be applied to ASR. Digital signal processing chips (DSP's) were introduced in the early 80's and have caused a revolution in product design. Current major DSP manufacturers include Texas Instruments (TI), Motorola, Lucent/ Agere, Analog Devices, NEC, SGS-Thomson, and Conexant (formerly Rockwell Semiconductor).

DSP's are specifically designed to rapidly perform the sum of products' operation required in many discrete-time signal processing algorithms. They contain hardware parallel multipliers and functions implemented by microcoding in ordinary microprocessors that are implemented by high-speed hardware in DSP's. Since they do not have to perform any of the functions of a high end microprocessor, like an Intel Pentium Dual-Core, a DSP can be streamlined to have a smaller size, use less power, and have a lower cost (Tretter, 2008). The velocity of the latest generations of DSP's have increased to the point where they are being used in high speed applications like DSL, wireless base stations and hand sets. Other applications include the implementation of video coding motion (Louro et al., 2003), Real-time audio transmission (Pasero & Montuori, 2003), e-commerce security (Jiankun et al., 2001), digital signal Processing system design (Kehtarnavaz et al., 2004) and implementation of facial recognition algorithms (Batur et al., 2003). Some of the advantages resulting are the integrated digital circuits that are very reliable and can be automatically inserted in boards easily.

DSP's can implement complicated linear and nonlinear algorithms and easily switch functions by jumping to different sections of the program code, such as; the implementation of an Artificial Neural Network, or on the "On-Chip DSP" for the ASR. The complexity of the algorithms is only limited by the imagination of the programmer and the processing speed of the DSP. For the implementation of a DSP in speech recognition applications, it is

essential to define the following parameters, such as; DSP technology, size of the data set, number of samples, computing speed and pattern recognition methods. Analog circuits are designed to perform specific functions and lack the flexibility of the programmable DSP approach. Another advantage is that small changes in the DSP function can be made by varying a few lines of code in a ROM or EPROM, while similar changes may be very difficult with a hard-wired analog circuit. Digital signal processing algorithms were used long before the advent of DSP chips.

DSP's have continually evolved since they were first introduced as VLSI improved technology since users requested additional functionality and as competition arose. Additional functions have been incorporated like hardware bit-reversed addressing for Fast Fourier Transform (FFT) and Artificial Neural Networks, hardware circular buffer addressing, serial ports, timers, Direct-Memory-Access (DMA) controllers, and sophisticated interrupt systems including shadow registers for low overhead context switching. Analog Devices has included switched capacitor filters and sigma-delta A/ D and D/ A converters on some DSP chips. Instruction rates have increased dramatically; the state-of-the-art DSP's, like the TMS320C5000 series are available with devices that can operate at clock rates of 200 MHz. Figure 2 presents a photograph of the first TMS 320 programmable DSP from Texas Instruments.

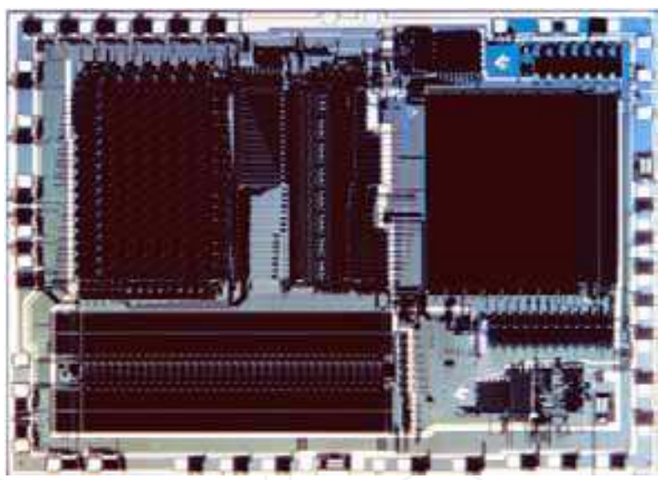


Fig. 2. First TMS 320 programmable DSP device, Texas Instruments, Inc.

The TMS320C6000 family has Very Long Instruction Word (VLIW) architecture, which has devices with clock rates up to 1 GHz (e.g. DSP TMS320C6416T of TI) the speed increase is largely a result of reduced geometries and improved CMOS technology.

In the last years, DSP manufacturers have been developing chips with multiple DSP cores and shared memory for use in high-end commercial applications like network access servers handling many voice and data channels. DSP chips with special purpose accelerators like Viterbi decoders, turbo code decoders, multimedia functions, and encryption/ decryption functions are appearing. The rapid emergence of broadband wireless applications is pushing DSP manufacturers to rapidly increase DSP speeds and capabilities so they do not have a disadvantage with respect to FPGA's.

In 1988, TI shipped initial samples of the TMS320C30 to begin its first generation TMS320C3x DSP family. This processor has a 32-bit word length. The TMS320C30 family has a device that can run at 25 million instructions per second (MIPs). The TMS320C31 has a

component that can perform 40 Million Instructions per second (MIPs). TI started its second generation DSP family with the TMS320C40 which contains extensive support for parallel processing. The last years ago TI introduced the TMS320C67x series of floating point and the TMS320C64x series of Fixed point DSP's, such as the TMS320C6713 and the TMS320C6416T. They were implemented on large main-frame computers and, later, on expensive "high-speed" mini-computers. DSP's of Floating-point and Fixed point are used in different applications because of their ease in programming. Some reasons for their applications are because they are smaller, cheaper, faster, and use less power. However, this is not much of a problem in speech recognition applications where the Levels of voice signal can be processed with any type of DSP.
Knowing that DSP's work well in the voice processing, these are used in a wide variety of offline and real-time applications. They are described in the following table:

Application	Description
Telecommunications	Telephone line modems, FAX, cellular telephones, speaker phones, ADPCM transcoders, digital speech interpolation, broadband wireless systems, and answering machines
Voice/Speech	Speech digitization and compression, voice mail, speaker verification, and automatic speech recognition (ASR).
Automotive	Engine control, antilock brakes, active suspension, airbag control, and system diagnosis.
Control Systems	Head positioning servo systems in disk drives, laser printer control, robot control, engine and motor control, and numerical control of automatic machine tools.
Military	Radar and sonar signal processing, navigation systems, missile guidance, HF radio frequency modems, secure spread spectrum radios, and secure voice
Medical	Hearing aids, MRI imaging, ultrasound imaging, and patient monitoring.
Instrumentation	Spectrum analysis, transient analysis, signal generators.
Image Processing	HDTV, pattern recognition, image enhancement, image compression and transmission, 3-D rotation, and animation

Table 1. Typical applications with TMS320C6000 family

Table 2 shows the following three categories of Texas Instruments DSP's. As can be seen for sound processing applications and speech recognition, it would be best to use a 32-bit DSP.

DSP's Categories	Characteristic	Applications
TMS320 C1s, C2x, C24x	Low Cost, Fixed-Point, 16-Bit Word Length	Motor control, disk head positioning, control
TMS320 C5x, C54x, C55x	Power Efficient, Fixed-Point, 16 Bit Words	Wireless phones, modems
TMS320 C62x (16-bit fixed-point) C3x, C4x, C64x, C67x (32–bit)	High Performance DSP's (32-bit floating-point and Fixed-Point)	Communications infrastructure, xDSL, imaging, sound, video

Table 2. Categories of Texas Instruments DSP's

3. ASR applications with integrated systems

ASR with integrated systems are employed in applications, such as; to enhance education for all students (Wald, 2005), methods to learn proper name pronunciations from audio samples (Beaufays et al., 2003), speech translation (Paulik et al., 2005) and classification techniques of speech parameters (Acevedo & Nieves, 2007). Many important results have been obtained through robust automatic speech recognition in car noise environments (Ding et al., 2006) and Statistical voice activity detection (Ramirez et al, 2005), while other applications employ powerful computers to handle complex recognition algorithms. There is clearly a demand for an effective solution on integrated systems like portable communication and various low-cost consumer electronic devices. Digital signal processor (DSP) is one of the most popular embedded platforms on which computationally intensive algorithms can be realized (Yuan et al., 2006).
This section shows a brief description of integrated systems and shows some important results using DSP technology, for example, to identify and classify voice commands or isolated words.

3.1 Isolated word recognition

The automatic speech recognition with integrated systems has a great performance for solving certain problems and limitations in human health service.
Nowadays, one of the problems that affects certain individuals is the lack of acoustic feedback (Deaf Speakers); thereby hindering their ability to communicate effectively (Kota et al., 1993). There is a practical need for the development of devices that can perform recognition of deaf speech in real time (The integrated systems can be a possible solution). Such devices could serve the communication needs of deaf speakers by correctly and reliably recognizing their speech and converting it into printed displays/ synthetic speech and used as voice input for a communication system. However, limitations on previously developed automatic speech recognition systems include limited vocabulary sets, intolerance to even slight variations in speech, and inability to operate in real time. Previous researchers have used a combination of dynamic time warping, template matching and HMM for recognition of deaf speech (Abdelhamied et al., 1990; Deller et al., 1988).
Reported recognition rates for isolated word recognition tasks have been in the range of 20% to 99.2%, and are highly dependent on the vocabulary, the extent of hearing loss of the speaker(s) and the performance of the recognition system itself. It is possible to select consistent acoustic features in deaf speech.
Artificial neural networks have been shown to perform pattern recognition, handle incomplete data and variability very well. It would seem appropriate that their use could enhance the performance of deaf speech recognizers by providing hybrid approaches of conventional signal processing and neural systems. In particular, the salient features of neural networks make them a useful tool in building better recognition systems for deaf speech. In this application three hundred utterances were recorded for each subject (age group 20-60 years old) in one session. Speech records were obtained from 6 deaf subjects. Speech intelligibility ratings were then obtained for each deaf speaker. One word list was selected from repetitions and randomized to avoid learning biases.
The entire ASR system including the preprocessor, data set, feature extractor and the neural network are implemented on the Texas Instruments' processor (TI) TMS320C30 DSP EVM (See Figure 3). The software for pre-processing and recognition tasks was written in SPOX

MI and C languages for maintaining modularity and portability. A time delay neural network model built for isolated digital recognition of normal speech was modified to incorporate the additional feature inputs to the network (Castro & Casacuberta, 1991). Time delays were built into the network structure to evolve time invariant feature extractors and feature integrators. The network was trained using general purpose backpropagation control strategy. A preliminary set of experiments were conducted with a vocabulary size of 20 words and dedicated networks for 2 speakers who had the highest and the lowest intelligibility in the speaker set. Six separate training schedules were developed each with a varying number of training tokens in the range 5-3. Each separate network was then tested for recognition rates with testing parameters.



Fig. 3. Texas Instruments (TI) TMS320C30GEL Digital Signal Processor (DSP)

In this application the best results were obtained with the average intelligibility ratings of the two deaf speakers that were 75 % and 38 %. The neural network learned specific features of both speakers very well. A perfect recognition - l00% - of the training set was obtained within 4-5 training sessions (epochs).

3.2 Voice pattern recognition for statistical methods

This application describes the implementation of a pattern recognition algorithm in a Blackfin DSP ADSPBF533 EZ-KIT of Analog Devices, such as; figure 4 presents, which can identify voice patterns from speakers in real time using statistical methods. This system responds only to voice commands based on a statistical analysis of a spectrogram generated by the voice command (Gómez, 2007). The goal of the generation of the spectrogram is characterized by a group of objects with measurements or qualities, where the values tend to be similar. For objects in the same class the differences are minimal and therefore the characteristics are unchanged and irrelevant to those changes in the data provided.

To extract the characteristics of a sound signal it is important not to take it all as a single pattern, since each segment of sound in the time domain has a different parameter and the sum of these is necessary to give the difference at the moment of applying the pattern recognition method. The signal is segmented in parts and each division generates an FFT, where the matrix is gotten with values of volume, frequency and the image obtained in three dimensions which is called spectrogram-time or frequency sonogram (Proakis &

Manolakis, 1998). The features extraction for the reflection spectrum is applied at the moment a voice command is acquired or any word with three different magnitudes of time, level and frequency. The three magnitudes generated by the speaker are affected depending on their moods. The speaker pronounces the command and then the spectrum changes in spite of being the same command.

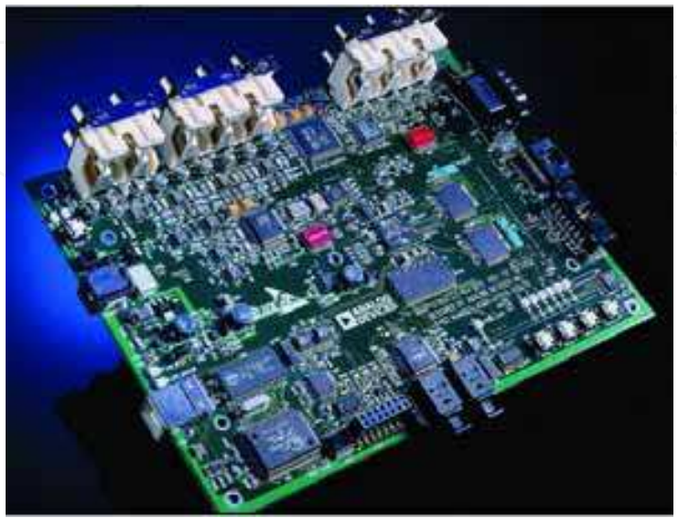


Fig. 4. ADSP-BF533 EZ-KIT Lite®, Analog Devices (ADI) evaluation system for Blackfin® embedded media processors.

Figure 5 shows the graphic of the spectrum reflecting the components in frequency and level. There are points in the categorization of the voice signal, where 9 points were extracted in the frequencies of 300Hz, 500Hz, 800Hz, 1.000Hz, 2.500Hz, 3.600Hz, 5.000Hz 7.000Hz and 8.000Hz. The magnitudes of these frequencies were carried to a vector, creating a database for each command voice.

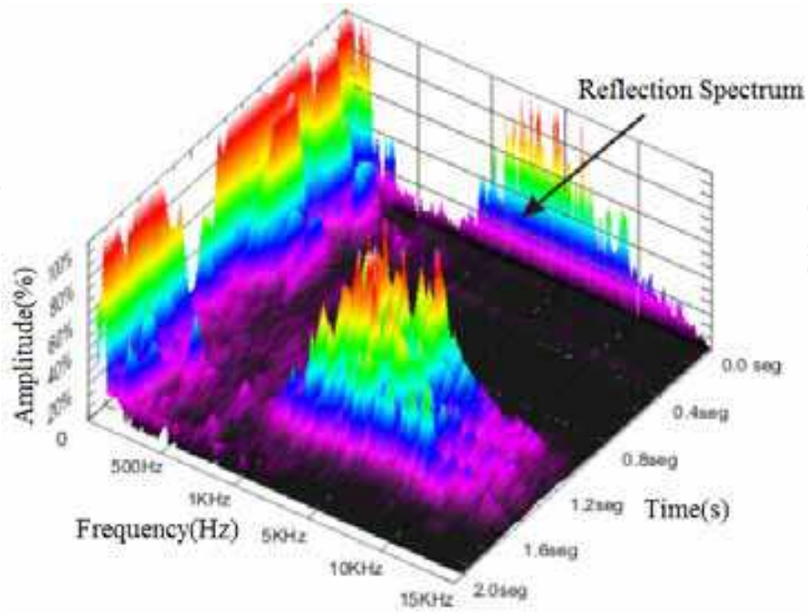


Fig.5. Spectrogram of voice command "Forward" showing the reflective spectrum.

After obtaining a characteristic vector from the dataset a linear regression is generated as a pattern recognition method. The linear regression analysis is a statistical technique for modeling of two or more variables. The regression can be used to relate a signal with another (Montgomery & Runger, 2006). The linear regression between two voice commands give as the results, a new vector of mean values or Mean Square Errors (MSE), and comparing two similar voice commands as the MSE goes to 0; see equation 1.
The residues of the MSE are determinate by:

$$MSE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(y_i - \hat{L}_i\right)^2 \quad (1)$$

Where, $\boldsymbol{L}$ is the regression estimated between two commands, $\boldsymbol{y}$ is the vector generated in the database with the voice commands and $\boldsymbol{n}$ is the length of the vector.
For the implementation of pattern recognition algorithm the development Kit ADSPBF533 EZ-KIT Lite of Analog Devices was utilized, which is composed of several peripherals, such as; ADC (Analog Digital Converters), DAC (Digital Analog Converters), RAM, EEPROM memory, etc. For the start-up of the device it is important to configure all ports and peripherals through the following functions: *i*) Configuration of general peripherals: This stage is composed of memory banks, the filter coefficients and the Fourier transform. *ii*) Configuration of the audio codec: The kit DSP includes an audio codec AD1836 designed for applications of high-fidelity audio; they are used in audio formats to 16 bits at a rate of frequency of 44.100Hz.
With the DSP system tests were made to determine their behavior and their effectiveness. The environmental conditions refer to external noise, location of the speaker regarding the microphone, types of microphones, parameters, compression and volume. Each voice command was assigned a binary code. The success rate of classification of each command was obtained with 50 words (10 each one) to verify coherence between the words pronounced and recognized (see table 3). The DSP acquires the voice command; the signals are filtered and the FFT is generating and calculating the matrix spectrogram, where the time, linear regression and MSE are calculated in milliseconds. The pattern recognition method with the spectrogram reflection technique applied on a DSP, delays 0.1 seconds in processing the information while a personal computer delays 2.5 seconds.

Command	Binary Code	Success rate (%)
Forward	000001	100
Stop	000010	95
Back	000100	100
Left	001000	98
Right	010000	100

Table 3. Success rate of the classification with the DSP Blackfin, to identify different voice commands with a single speaker.

4. Experimental framework

The following section describes an experimental setup where a LPC-ANN hybrid system was used as an alternative in the identification of voice commands from a speaker, and the implementation using Matlab Released 7.1 Software coupled with the DSP Hardware (i.e. The DSK6416T) developed by Texas Instruments.

Figure 6 shows the Automatic speech recognition; it is constituted by two principal phases. The first phase is the training stage where each word or voice signal is acquired with the purpose to obtain a descriptive model from all the words used to build the model and train the network. As can be seen, in the recognition phase a new voice sample is acquired and is then projected onto the model to identify and classify the voice signal using the already trained network. The signal acquisition is obtained with the help of a high gain microphone and then the time is digitized by means of a computer audio card; in this process the voice signals obtained through the feature extraction techniques. With the feature extraction the spectral measurements become a set of parameters that describes the acoustic properties of phonetic units. These parameters can be: Cepstral coefficients, or the energy of the signal (i.e. extracting the energy from LPC), etc. Once the basic parameters are obtained, the aim is to identify the voice signal, applying the methods and algorithms that are translated into numerical values. For this, Neural Networks are used, such as; the Backpropagation or multilayer neural network specifically. Backpropagation was created by generalizing learning rules to multiple-layer networks and nonlinear transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can be approximate as a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as well as can be defined. A Backpropagation consists of three types of layers, namely: the input layer, a number of hidden layers; and the output layer. Only the units in the hidden and output layers are neurons and so it has two layers of weights (Cong et al., 2000; Kostepen & Kumar, 2000). In this experiment a Multilayer Perceptron (MLP) neural network is used, which has a supervised learning phase and employs a set of training vectors, followed by the prediction or recall of unknown input vectors.

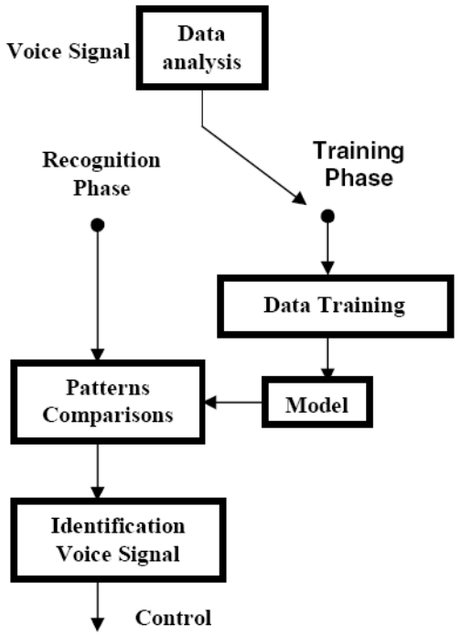


Fig.6. Block Diagram of Automatic Speech Recognition (ASR)

4.1 Data acquisition

Each one of the stages of the Automatic Speech Recognition is developed with the help of Matlab 7.1. The data acquisition control and signal processing are done with an audio digital

card, LPC and Neural Networks which are accomplished by a written-in-house software through of a Graphic User Interface (GUI). This software allows the voice signal to be acquired quickly in real-time. The software allows a model to obtain the training data under platform of Matlab-Simulink and the implementation on the DSP Hardware. To acquire the signal from the auxiliary input of the computer audio card, the function, "wavrecord" is used which corresponds to the acquisition time (i.e .in seconds), the sampling frequency (Fs) in Hz (e.g. 8000, 11025, 22050 and 44100) and the channel is obtained (i.e. Mono is Ch_1 and Ch_2 is Stereo). For example, to acquire a signal in mono-stereo with a period of one second of duration and the sampling frequency of 8000 Hz, it is possible to use the following command from the Workspace of Matlab:

```
>>Fs=8000
>>Y=wavrecord (1*Fs, Fs, 1)
```

To keep a signal in audio format (e.g. wav) the function "wavwrite" is used, where wavwrite (i.e. Y,Fs,NBITS,WAVEFILE) writes the data "Y" to a Windows file specified by the file name "WAVEFILE", with a sample rate of "Fs" in Hz and with "NBITS" number of bits. NBITS must be 8, 16, 24, or 32. Stereo data should be specified as a matrix with two columns. For NBITS < 32, amplitude values outside the range [-1, +1] are clipped. For example, in order to keep the previous sound, the following command will be used:

```
>> wavwrite (Y, Fs, 16,' close.wav')
```

Figure 7 shows the typical signal of the 'close' command, at the moment to acquire the voice signal from the auxiliary input. A total of 12,000 samples was obtained for the first 30 measurements and were processed later.

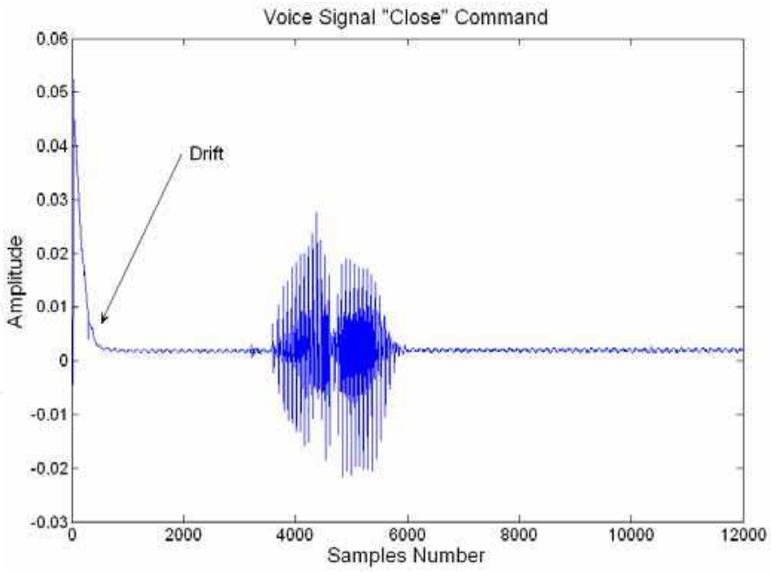


Fig. 7. Voice Signal of the "Close" Command

4.2 Signal processing

The next stage consists of acquiring the energy of the signals from LPC, with the possibility to acquire the principle parameters for network training. The signals processed were made

with algorithms of Matlab. The first samples of the signals acquired were reduced due to the drift that was generating from the audio board; therefore, the "baseline" of the signal was acquired (see figure 8).

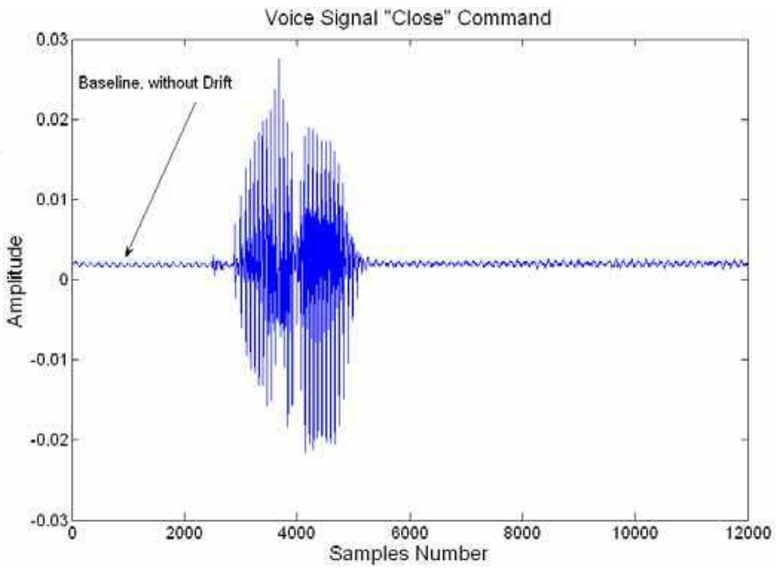


Fig. 8. Voice Signal without "drift" and with "baseline"

The figure below illustrates the acquisition of the first samples which do not contribute important information to the recognition process; therefore, this part was eliminated. In the second step the signal is pre-processed (i.e. Normalized) for 2,000 samples. The figure illustrates this process; where it represents the reduced graph with regard to the samples. This method was used with the aim so that the signal is always located at the same point and the acquisition of measurements must be repetitive.

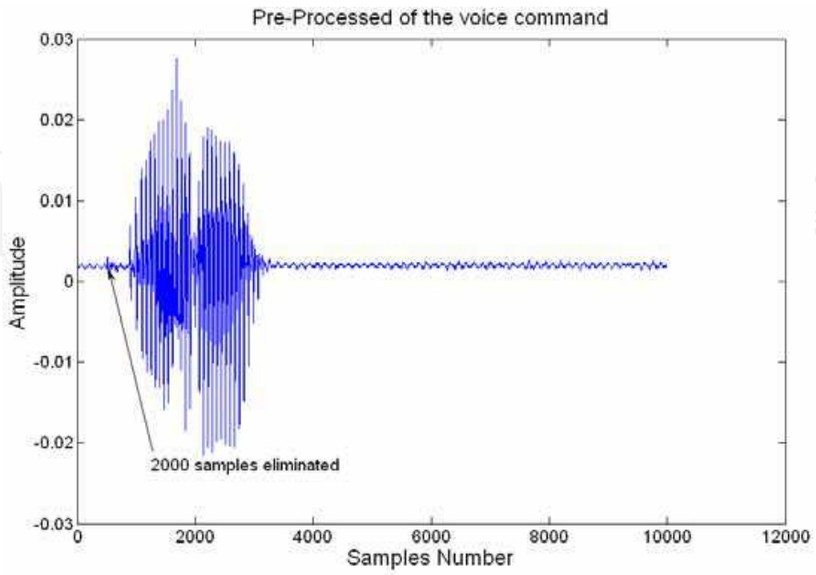


Fig. 9. Normalized Signal

In the next stage one proceeds to find the energy of the signal in the time domain. Two processes with the energy extraction algorithm were developed; in the first, the energy was obtained according to equation 2: In the second, the energy was normalized in as much in amplitude as in duration. 'E' is the energy and '*X*' is the voice signal. Figure 10 illustrates the energy for the signal of the "open" command. This process is the same for the open, close, lights and television commands.

$$E(t) = 10.\log \sum_{i=1}^{L} |X(i)|^2 \tag{2}$$

After finding the energy, data analysis techniques of one-dimension for discrete signals in time were applied. The characteristics of each one of the words that form the data set are a faculty of the speaker. A total of 120 measurements were acquired from 4 words (open, close, lights and television), from which 30 measurements correspond to each of them. In order to train the system, the neural network toolbox of Matlab was used to create the model. In this case two-layers of feed-forward network were created. The network's input corresponds to one data set of 120 measurements, from which the first layer has ten neurons and the second layer has one neuron. The network is simulated and the output is acquired through targets; finally the network was trained for 1000 epochs and the output acquired.

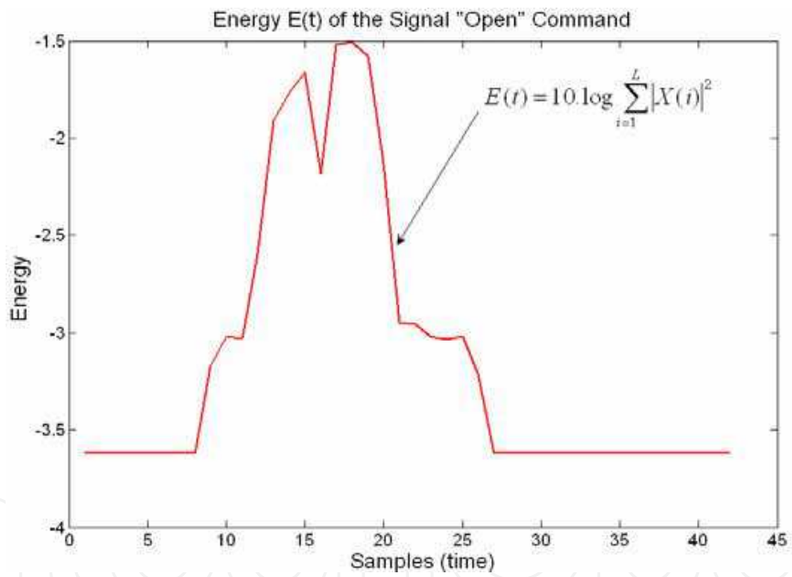


Fig. 10. Energy signal

Figure 11 shows three blocks in the recognition phase, where the first block acquires the voice signal, and afterwards the energy of the signal is obtained to be entered into the neural network. The function "wavrecord" is used to acquire the voice signal; subsequently, the energy of this command is obtained and it will be stored in the vector L, so that from Simulink of Matlab the network takes the value of this vector, making the pattern recognition.
Figure 12 shows the recognition algorithm; the first block takes the value of the vector L (i.e. Energy of the Signal) from the Workspace, the neural network finds the best success rate, while the third block is a function of the input value. The output network displays a number that can be a decimal value (i.e. Binary digits or LED (light emitting diodes) displays) for each of the commands, for example: 1, 2, 3, or 4, depending on the command input.

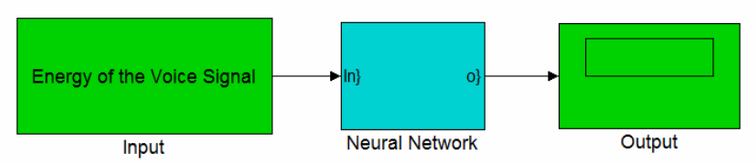


Fig. 11. Simulation of the Network

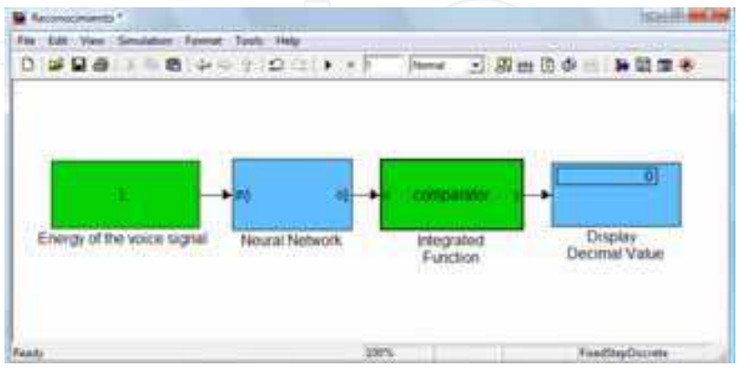


Fig. 12. Pattern Recognition System

4.3 DSP hardware (DSK TMS320C6416T of fixed point)

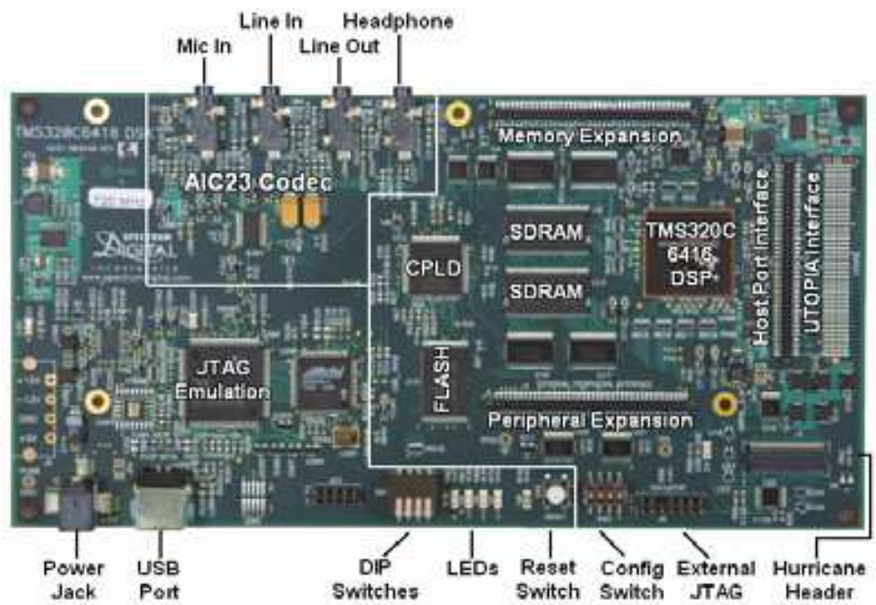


Fig. 13. C6416 DSK Board

The following section is important for the reader, for the reason that describes the DSP board which was used for the voice recognition in real time. The 6416 DSP Starter Kit (DSK) is a low-cost platform which lets customers evaluate and develop applications for the Texas Instruments C64x DSP family. The primary features of the DSK are: Speed 1 GHz, AIC23 Stereo Codec, four Position User DIP Switch and Four User LED's, On-board Flash and SDRAM. The figure 13 shows the main sections of the board.

The TMS320C6416 processor is the heart of the system. It is a core member of Texas Instruments'C64X line of "fixed point" DSP's whose distinguishing features are an extremely high performance 1 GHz VLIW DSP core and a large amount of fast on-chip memory (1Mbyte). On-chip peripherals include two independent external memory interfaces (EMIFs), 3 multi-channel buffered serial ports (McBSPs), three on-board timers and an enhanced DMA controller (EDMA). The 6416 represents the high end of TI's C6000 integral DSP line both in terms of computational performance and on-chip resources. The 6416 has a significant amount of internal memory so typical applications will have all code and data on-chip, especially designed for applications of speech recognition. External accesses are done through one of the EMIFs, either the 64-bit wide EMIFA or the 16-bit EMIFB. EMIFA is used for high bandwidth memories, such as; the SDRAM while EMIFB is used for non-performance critical devices, such as, the Flash memory that is loaded at boot time. A 32-bit subset of EMIFA is brought out to standard TI expansion bus connectors so additional functionality can be added on daughtercard modules.

DSPs are frequently used in audio processing applications so the DSK includes an on-board codec called the AIC23. Codec stands for coder/ decoder. The job of the AIC23 is to code analog input samples into a digital format for the DSP to process; then decoded data comes out of the DSP to generate the processed analog output. Digital data is sent to and from the codec on McBSP2. The DSK has 4 LED´s and 4 DIP switches that allow users to interact with programs through simple LED displays and user input on the switches. Many of the included examples make use of these user interface options. The DSK implements the logic necessary to tie board components together in a programmable logical device called a CPLD. In addition to random glue logic, the CPLD implements a set of 4 software programmable registers that can be used to access the on-board LEDs and DIP switches as well as control the daughtercard interface.

The DSK uses a Texas Instruments' AIC23 (i.e. part #TLV320AIC23) stereo codec for input and output of audio signals, as shown in the next figure:

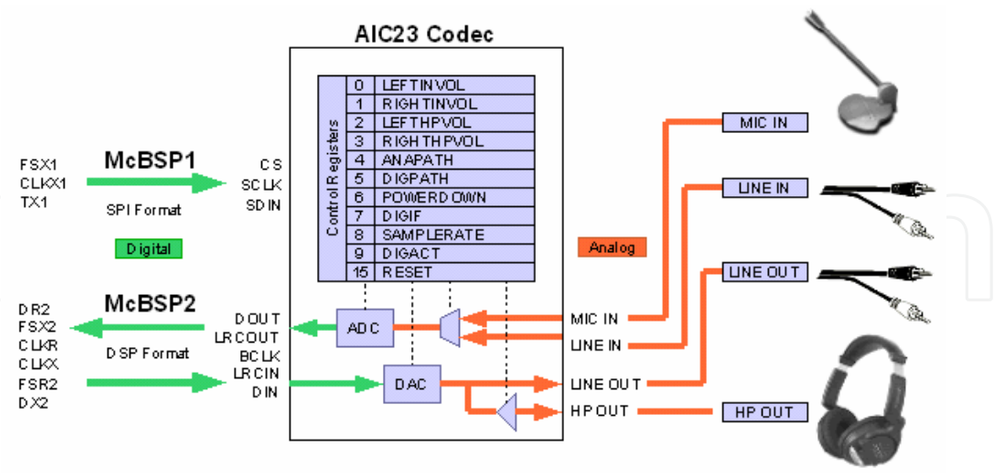


Fig. 14. (TLV320AIC23) audio stereo codec

The codec samples analog signals on the microphone or line inputs and converts them into digital data that can be processed by the DSP. When the DSP is finished with the data, it uses the codec to convert the samples back into analog signals on the line and headphone outputs so the user can hear the output.

The codec communicates using two serial channels; one to control the codec's internal configuration that registers, and one to send and receive digital audio samples. The AIC23 supports a variety of configurations that affect the data formats of the control and data channels.

4.4 Development tool

The following is the tool that provides the code needed to run the DSP Board. The development tool adapted to the DSP is the Code Composer Studio TI.

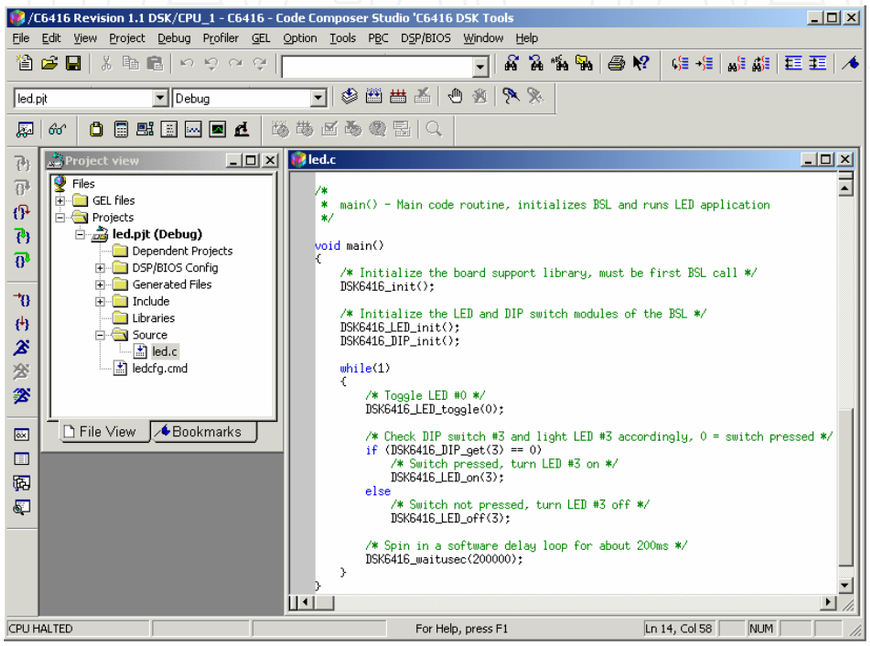


Fig. 15. Development Environment Tool: Code Composer Studio

It consists of an assembler, a C compiler, an integrated development environment (IDE, the graphical interface for the tools) and numerous support utilities like a hex format conversion tool. The DSK includes a special version of the Code Composer especially tailored to features on the 6416 DSK board. Other versions of the Code Composer are available that fully support each of TI's processor families on a wide variety of hardware targets.

The Code Composer IDE is the piece you see when you run the Code Composer. It consists of an editor for creating the source code, a project manager to identify the source files and options necessary for your programs and an integrated source level debugger that lets you examine the behavior of your program while it is running. The IDE is responsible for calling other components, like the compiler and assembler so that developers do not have to hassle running each tool manually.

The 6416 DSK includes a special device called a Joint Test Action Group (JTAG) emulator on-board that can directly access the register and memory state of the 6416 chip through a standardized JTAG interface port. When a user wants to monitor the progress of his program, the Code Composer sends commands to the emulator through its USB host interface to check on any data the user is interested in. This method is extremely powerful because programs can be debugged unobtrusively on real hardware targets without making any special provisions for debug-like external probes, software monitors or simulated

hardware. When designing your own hardware around the 6416, you can debug your application with the same wide functionality of the DSK simply by using the Code Composer with an external emulator and including a header for the JTAG interface signals. You should always be aware that the DSK is a different system from your PC; when you recompile a program with the Code Composer on your PC, you must specifically load it onto the 6416 on the DSK. Other things to be aware of are: 1) when you tell the Code Composer to run, it simply starts executing at the current program counter. If you want to restart the program, you must reset the program counter by using 'Debug' and 'Restart' or re-loading the program that implicitly sets the program counter. 2) After you have started a program running, it continues running on the DSP indefinitely. To stop it, you need to halt it with 'Debug' and 'Halt'.

4.5 Implementation on the DSP

Figure 16 shows a Graphic User Interface (GUI), where each stage of the process is visualized from the acquiring of the signal until the pattern recognition phase. The GUI consists of three-buttons; the first one to capture the signal from the microphone. The next button to obtain the energy and the recognition of the voice command of what has been pronounced before. The energy that was obtained is entered onto the net for its recognition.

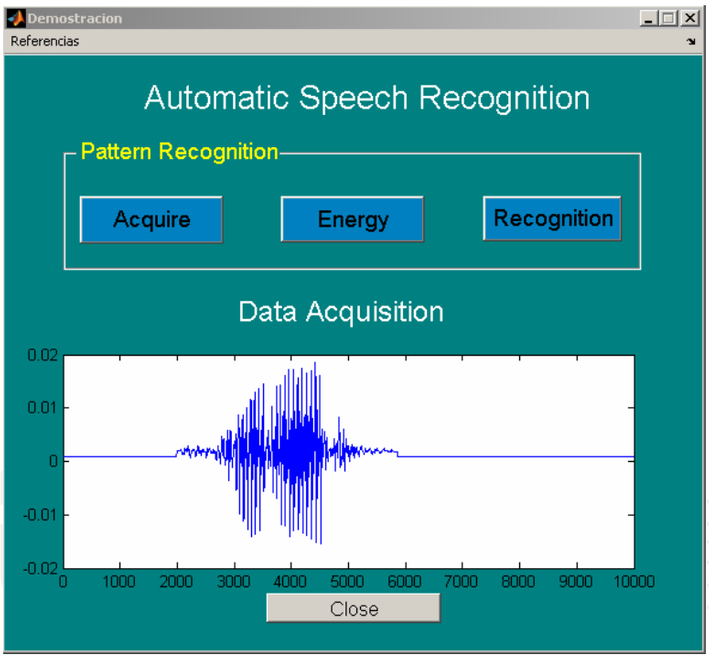


Fig. 16. ASR Interface

The last button makes the decision of the speech recognition command. After having carried out the two previous stages, it then, proceeds to identify the command with the DSP board; this started and builds the model.

Figure 17 shows that the code automatically generates, creates the model with respect to the new voice command through "The Code Composer Studio (CCS)". To achieve this, the tool TLC was used (i.e. Target Language Compiler) from Simulink, to translate the Simulink Block to language C or assembler. Figures 18 and 19 illustrate the experimental way to implement the ASR System using the DSK6416 from Texas Instruments.

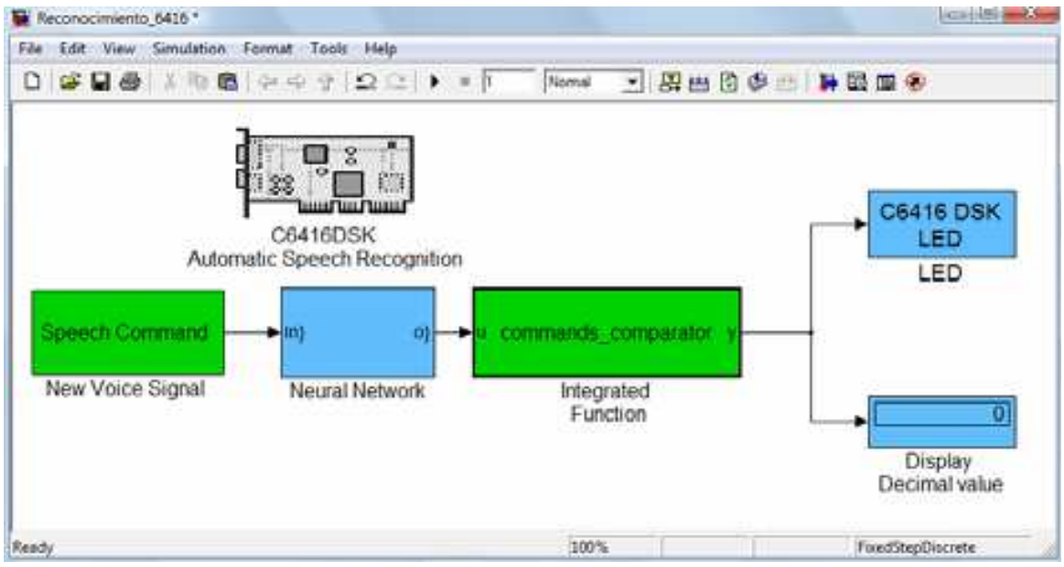


Fig. 17. Recognition model for the DSK6416

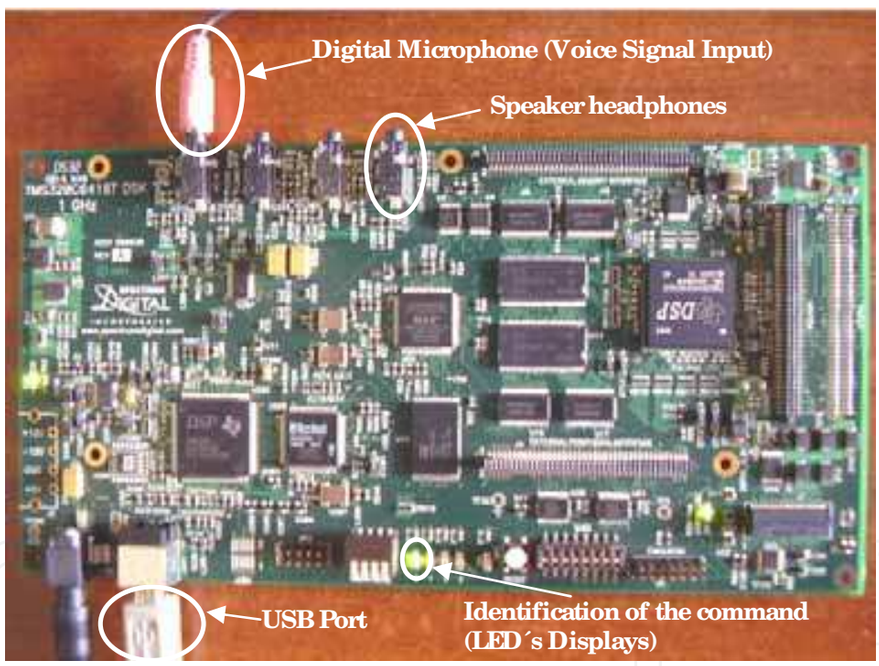


Fig.18. Connection of the DSP Board

The recognition system was subjected to a group of 4 habitual words (Open, Close, Lights and Television). The experiments were carried out pronouncing each one of the words 30 times and writing the successes and mistakes down obtaining 98% of a success rate of classification with only one error, which was detected with the "Ligths" command and located on the command "Close"; see Table 4. It is necessary to highlight that the tests were carried out for a single speaker and under conditions of absence of background noise and the pronunciation of the words that were made with the same characteristics with which it is possible to train the system.



Fig. 19. ASR with the DSP Hardware (DSK6416 TI)

Speech Command	1	2	3	4
Open	**30**	-	-	-
Close	-	**30**	**Error**	-
Lights	-	-	**29**	-
Television	-	-	-	**30**

Table 4. Results of the Classification with 4 speech commands

The results of this experiment concluded that is possible to implement a speech recognition algorithm (e.g. neural network) in a DSP, as a powerful tool or integrated system for automatic speech recognition.

5. Conclusions

This chapter has shown a review of the state of the art with some applications of the integrated systems, such as DPS´s to improve the performance of the ASR Systems. The chapter has summarized the VLSI technology and algorithm-oriented array architecture which appears to be effective, feasible, and economical in applications of speech recognition. DSP's are frequently used in a number of applications including telecommunications, automotive industries, control systems, medical-science, image processing, and now, in speech recognition. The potential of using statistical techniques and neural networks in speech recognition tasks (e.g. Isolated word recognition, solving limitations in human health, etc.) has been reviewed and preliminary results indicate that they have the potential to improve the performance of speech recognition systems. Dealing with the experimental

framework it is important to bring out that although initially tested with few words, in future work, it will be possible to make tests with a wider data set of voice commands for training.

6. References

Abdelhamied, K.; Waldron, M.; Fox, R.A. (1990). Automatic Recognition of Deaf Speech, *Volh Review*, volume: 2 pp. 121-30, Apr 1990.

Acevedo, C.M.D.; Nieves, M.G. (2007). Integrated System Approach for the Automatic Speech Recognition using Linear predict Coding and Neural Networks, *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pp. 207-212, 25-28 Sept. 2007, ISBN: 978-0-7695-2974-5, Cuernavaca.

Barazesh, B.; Michalina, J.C.; Picco, A. (1988). A VLSI signal processor with complex arithmetic capability, *Circuits and Systems, IEEE Transactions on*, Volume: 35, Issue: 5, May 1988, pp. 495-505, ISSN: 0098-4094.

Batur, A.U.; Flinchbaugh, B.E.; Hayes, M.H. (2003). A DSP-based approach for the implementation of face recognition algorithms, *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, Volume: 2, pp. 253-256, ISSN: 1520-6149.

Beaufays, F.; Sankar, A.; Williams, S.; Weintraub, M. (2003). Learning name pronunciations in automatic speech recognition systems, *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*, pp. 233- 240, 3-5 Nov. 2003, ISSN: 1082-3409.

Benzeghiba, M.; Mori, D.; Deroo, O.; Dupont, S.; Erbes, T.; Jouvet, D.; Fissore, L.; Laface, P.; Mertins, A.; Ris, C.; Rose, R.; Tyagi, V.; Wellekens, C. (2007). Automatic speech recognition and speech variability: A review, *Speech Communication*, Volume 49, Issues 10-11, October-November 2007, pp. 763-786, ISSN: 0167-6393.

Castro, M.J; Casacuberta, F. (1991). The use of multilayer perceptrons in isolated word recognition, *Artificial Neural Networks*, *Springer Berlin /Heidelberg*, pp. 469-476, ISBN: 978-3-540-54537-8.

Cheng, H.D.; Tang, Y.Y.; Suen, C.Y.; (1991), VLSI architecture for size-orientation-invariant pattern recognition, CompEuro '91. *Advanced Computer Technology, Reliable Systems and Applications. 5th Annual European Computer Conference. Proceedings*, pp. 63-67, ISBN: 0-8186-2141-9, 13-16 May 1991, Bologna, Italy.

Cong, L.; Asghar, S.; Cong, B. (2000). Robust speech recognition using neural networks and hidden Markov models, *Information Technology: Coding and Computing*, *Proceedings. International Conference on*, pp. 350-354, ISBN: 0-7695-0540-6.

Coy, A.; Barker, J. (2007). An automatic speech recognition system based on the scene analysis account of auditory perception, *Speech Communication*, Volume 49 , Issue 5, May 2007, pp. 384-401, ISSN:0167-6393.

Deller, J.R.; Hsu, D.; Ferrier, L.J. (1988). Encouraging results in the automated recognition of cerebral palsyspeech, *Biomedical Engineering, IEEE Transactions on*, Volume: 35, Issue: 3, pp.218-220, Mar 1988, ISSN: 0018-9294.

Ding, P.; He, L.; Yan, X.; Zhao, R.; Hao, J. (2006). Robust Technologies towards Automatic Speech Recognition in Car Noise Environments, *Signal Processing, 2006 8th International Conference on*, pp. 16-20, ISBN: 0-7803-9737-1, Beijing.

Gómez,A.J. (2007). Diseño e Implementación de un Sistema de Reconocimiento de Patrones de Voz Basado en un DSP Blackfin, *XII simposio de tratamiento de señales, imágenes y visión artificial. STSIVA*, Barranquilla (Colombia).

Jiankun, H.Z.X.; Jennings, A.; Lee, H.Y.J; Wahyudi, D. (2001). DSP application in e-commerce security, *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, Volume: 2, pp. 1005-1008, ISBN: 0-7803-7041-4, Salt Lake City, UT, USA.

Jou, J.M.; Shiau, Y.H.; Huang, C.J. (2001). An efficient VLSI architecture for HMM-based speech recognition, Electronics, *Circuits and Systems, 2001. ICECS 2001 the 8th IEEE International Conference on*, Volume 1, pp. 469 – 472, ISBN: 0-7803-7057-0, 2-5 Sept. 2001.

Kehtarnavaz, N.; Kim, N.; Panahi, I. (2004). Digital signal processing system design: using LabVIEW and TMS320C6000, *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th*, pp. 10-14, ISBN: 0-7803-8434-2.

Kota, R.; Abdelhamied, K.A.; Goshorn, E.L. (1993). Isolated word recognition of deaf speech using artificial neural networks, *Biomedical Engineering Conference, 1993., Proceedings of the Twelfth Southern*, pp. 108-110, ISBN: 0-7803-0976-6, New Orleans.

Kostepen, M.H.; Kumar, G. (1991). Speech Recognition using Back-Propagation Neural Networks, *TENCON '91.1991 IEEE Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems*, Volume: 2, pp. 144-148, ISBN: 0-7803-0538-8, 28-30 Aug.

Kung, S. (1985). VLSI Array Processors, *ASSP Magazine, IEEE Signal Processing Magazine Inc*, Volume: 2, issue: 3, Part 1, July 1985, pp. 4-22, ISSN: 0740-7467.

Kwong, S.; Man, K.F. (1995). A Speech Coding Algorithm based on Predictive Coding, *Data Compression Conference, 1995. DCC '95. Proceedings*, pp. 455, ISBN: 0-8186-7012-6, 28-30 Mar Hong Kong.

Leitch, D.; Bain, K. (2000). Improving Access for Persons with Disabilities in Higher Education Using Speech Recognition Technology. *AVIOS Proceedings of The Speech Technology & Applications Expo*, pp. 83-86.

Lim, C.P.; Woo, S.C.; Loh, A.S.; Osman, R. (2000). Speech Recognition Using Artificial Neural Networks, wise, *First International Conference on Web Information Systems Engineering (WISE'00)*, Volume 1, pp. 419, ISBN: 0-7695-0577-5-1, 2000, IEEE Computer Society.

Louro, L.; Santos, P.; Rodrigues, N.; Silva, V.; Faria, S. (2003). DSP performance evaluation for motion estimation, *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, Volume: 2, pp. 137- 140, 1-4 July 2003, ISBN: 0-7803-7946-2.

Melnikoff, S.J; Quigley, S.F.; Russell, M.J. (2002), Speech Recognition on an FPGA Using Discrete and Continuous Hidden Markov Models, *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream, Springer Berlin / Heidelberg*, pp. 89-114, ISBN: 978-3-540-44108-3.

Montgomery, D.C.; Runger, G.C. (2006). Applied Statistics and Probability for Engineers, *Jhn Wiley & Sons*, ISBN-13: 978-0-471-74589-1.

Pasero, E.; Montuori, A. (2003). Neural network based arithmetic coding for real-time audio transmission on the TMS320C6000 DSP platform, *Acoustics, Speech, and Signal*

Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, Volume: 2, pp. 761-764, 6-10 April 2003, ISBN: 0-7803-7663-3.

Paulik, M.; Stuker, S.; Fugen, C.; Schultz, T.; Schaaf, T.; Waibel, A. (2005). Speech translation enhanced automatic speech recognition, *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pp. 121 – 126, 27 Nov- 1 Dec 2005, ISBN: 0-7803-9478-X.

Peinado, A.; Segura, J.C. (2006). Speech Recognition Over digital Channels Robustness and standards, *John Wiley & Sons Ltd*, ISBN-13: 978-0-470-02400-3, ISBN-10: 0-470-02400-3, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

Phadke, S.; Limaye, R.; Verma, S.; Subramanian, K, (2004), On Design and Implementation of an Embedded Automatic Speech Recognition System, *VLSI Design, 2004. Proceedings. 17th International Conference on*, pp. 127-132, ISBN: 0-7695-2072-3.

Proakis, J.G., Manolakis, D.K. (2006). Digital Signal processing (4th Edition), *Prentice Hall*; 4 edition (April 7, 2006), ISBN-13: 978-0131873742.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, Volume 77, Issue 2, pp. 257-286, ISSN: 0018-9219.

Ramirez, J; Segura, J.C.; Benitez, C.; Garcia, L.; Rubio, A.J. (2005). Statistical voice activity detection using a multiple observation likelihood ratio test, *IEEE Signal Processing Letters 12 (10)*, pp. 689–692.

Schroeder, M.R.; Atal, B.S. (1985). Code-excited Linear Prediction (CELP): High quality speech at very low bit rates, *IEEE International Conference on ICASSP '85, Acoustics, Speech, and Signal Processing*, Volume 10 pp.937-940, April 1985.

Tang, Y.Y.; Tao, L; Suen, C.Y. (1994). VLSI Arrays for Speech Processing with Linear Predictive, *Pattern Recognition, Conference C: Signal Processing, Proceedings of the 12th IAPR International Conference on*, pp. 357 – 359, ISBN: 0-8186-6275-1, University Chongqing, Oct 1994.

Tretter, S.A. (2008). Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6713™ DSK, *Springer Science Business Media, LLC*, ISBN: 978-0-387-74885-6.

Wald, M.; (2005).Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality, *Frontiers in Education, FIE '05. Proceedings 35th Annual Conference*, pp. S3G-22- S3G-25, 19-22 Oct. 2005, ISBN: 0-7803-9077-6.

Wald, M. (2006). Captioning for Deaf and Hard of Hearing People by Editing Automatic Speech Recognition in Real Time, *Proceedings of 10th International Conference on Computers Helping People with Special Needs ICCHP 2006, LNCS 4061*, pp. 683-690.

Yuan, M.; Lee, T.; Ching, P.C.; Zhu, Y. (2006). Speech recognition on DSP: issues on computational efficiency and performance analysis, *Microprocessors and Microsystems*, Volume 30, Issue 3, 5 May 2006, pp. 155-164, ISSN: 0141-9331.



Speech Recognition

Edited by France Mihelic and Janez Zibert

ISBN 978-953-7619-29-9

Hard cover, 550 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

Chapters in the first part of the book cover all the essential speech processing techniques for building robust, automatic speech recognition systems: the representation for speech signals and the methods for speech-features extraction, acoustic and language modeling, efficient algorithms for searching the hypothesis space, and multimodal approaches to speech recognition. The last part of the book is devoted to other speech processing applications that can use the information from automatic speech recognition for speaker identification and tracking, for prosody modeling in emotion-detection systems and in other speech processing applications that are able to operate in real-world environments, like mobile communication services and smart homes.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

C. Durán (2008). Ultimate Trends in Integrated Systems to Enhance Automatic Speech Recognition Performance, Speech Recognition, France Mihelic and Janez Zibert (Ed.), ISBN: 978-953-7619-29-9, InTech, Available from: http://www.intechopen.com/books/speech_recognition/ultimate_trends_in_integrated_systems_to_enhance_automatic_speech_recognition_performance

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen