Integrated System Approach for the Automatic Speech Recognition using Linear predict Coding and Neural Networks

Cristhian Manuel Durán Acevedo, Martín Gallo Nieves
Department of Electronic Engineering, University of Pamplona, Pamplona, Colombia
cmduran@unipamplona.edu.co, magallo@unipamplona.edu.co

Abstract

In the present article we presented an automatic speech recognition approach to identify initially four voice words using the energy of the signal through of LPC (Linear Predict Coding) and finally neural networks as recognition and classification techniques of speech parameters. The identification of speech command was obtained with a Back propagation Multilayer Perceptron (MLP). The characteristics of the voice parameters in the time domain were processed, and the neural network was trained to classify and identify the speech commands.

The implementation of this system has been test in one Development Starter Kit (DSK), Digital Signal Processing Card of 1 GHz, with reference TMS320C6416T of Texas Instruments, utilized in this application.

1. Introduction

Automatic speech recognition (ASR) technology has made enormous advances in the last 20 years, and now large vocabulary systems can be produced that have sufficiently good performance to be usefully employed in a variety of tasks [1,2,3,4]. 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") and content-based spoken audio search (e.g. find a podcast where particular words were spoken). Voice recognition is a related process that attempts to identify the person speaking, as opposed to what is being said.

Nowadays general-purpose speech recognition systems are generally based on hidden Markov models (HMMs). This is a statistical model which outputs a sequence of symbols or quantities. One possible reason why HMMs are used in speech recognition is that a speech signal could be viewed as a piece-wise stationary signal or a short-time stationary signal. That is, one could assume in a short-time in the range of 10 milliseconds, speech could be approximated as a stationary process. Speech could thus be thought as a Markov model for many stochastic processes (known as states).

One of the most powerful speech analysis techniques is Linear Predictive Coding (LPC). The covariance analysis of linear predictive coding has wide applications, especially in speech recognition and speech signal processing [5,6,7]. Real-time applications demand very high speed processing speed for linear predictive coding analysis.

Another approach in acoustic modeling is the use of Neural Networks (NN). They are capable of solving much more complicated recognition tasks, but do not scale as well as LPC or HMMs when it comes to large vocabularies. 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 than LPC or HMMs 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 are better than for LPC or HMMs. There are also LPC-NN and NN-LPC hybrid systems that use the neural network.

In this work we used the LPC-NN hybrid system as alternative in the identification of speech command and the implementation using Matlab Released 7.1 Software [8] coupled with the Hardware DSP (Digital Signal Processing, DSK6416T) developed by Texas Instruments [9].

0-7695-2974-7/07 $25.00 © 2007 IEEE
DOI 10.1109/CERMA.2007.97

2. Experimental Setup

Figure 1 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 model descriptive from all the words used to build the model and train the network.

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.

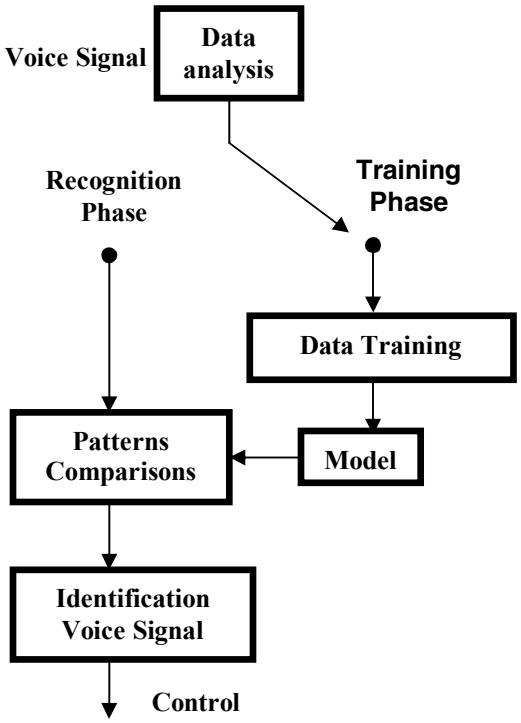

Figure 1. Block Diagram of a Automatic Speech Recognition (ASR)

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 is obtained the voice signals through of feature extraction techniques. With the feature extraction the spectral measures become a set of parameters that describe the acoustic properties of phonetic units. These parameters can be: Coefficients cepstral, the energy of the signal (e.g extracting the energy from LPC), etc. Once is obtained the basic parameters we intend to identify the voice signal, then we used methods and algorithms that translate us the numerical values for it we used Neural Networks such as the Backpropagation or multilayer neural network specifically.

Backpropagation was created by generalizing learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as well defined by you. A Backpropagation consist 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 neurones and so it has two layers of weights. In this work we used a Backpropagation multilayer neural which has a supervised learning phase and employs a set of training vectors, followed by the prediction or recall of unknown input vectors.

2.1. Data acquisition

Each one of the stages of an Automatic Speech Recognition has been development with the help of Matlab 7.1. The data acquisition control and signal processing were realized with an Audio Digital Card, LPC and Neural Networks which were executed by written-in-house software through the use of a GUI (Graphic User Interface). This software allowed to acquire the Voice signal in real time to obtain pre-processed and processed results very fast. The goal to use this software was to obtain a model of data training under environment Matlab-Simulink and your later implementation on the Hardware DSP.

In order to acquire the signal for the auxiliary input of computer audio card, we used the function "*wavrecord"* which correspond to the acquisition time (e.g., seconds), the sampling frequency in Hz (e.g., 8000, 11025, 22050 and 44100) and the channel (Mono Ch_1 and Ch_2 to 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, we can use the following command from Matlab:

- Fs = 8000;
  y = wavrecord (1*Fs, Fs, 1);

To keep a signal in audio format (e.g., wav) we used the function *"wavwrite"* (Y, Fs, NBits, name_voice_Signal.wav) where the parameters correspond to writes data "Y" to a Windows WAVE file specified by the file name WAVEFILE, with a sample rate of F_S 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 example,

in order to keep the sound, previously the following command will be used:

- wavwrite *(y, Fs, 16,' close.wav');*

The figure 2 shows the typical signal of the command 'close', in the moment to acquire the voice signal from the auxiliary input. A total of 12000 samples were obtained for the first 30 measurements and later were processed.

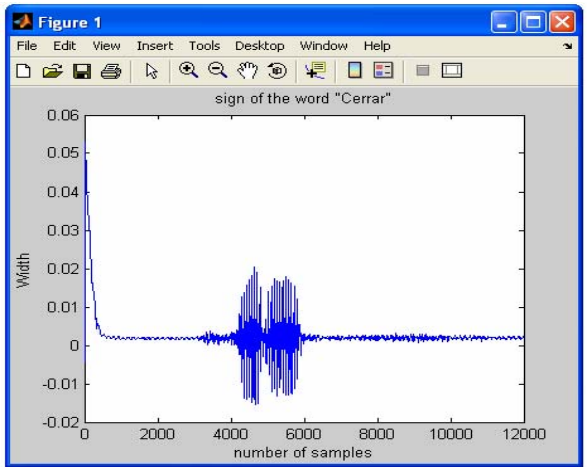

Figure 2. Voice Signal of the Command 'Close'

2.2. Signal Processing

We obtained the energy of the signals from LPC with the aim to get the principal parameters for the net training. The signals were processed with an algorithms made in Matlab.

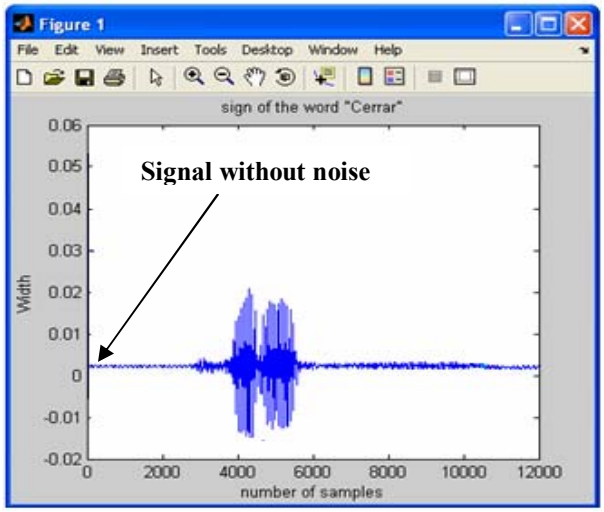

Figure 3. Signal without noise

The first samples of the complete signal and straightaway acquired from Matlab were reduced due to the noise than was generating of the audio card. The figure 3 and 4 shows the moment when start the acquisition and the first samples do not contribute with important information for the recognition process. Therefore, this part was eliminated.

The second step was to normalize the signal in the time in such a way is start from one fixed point, in this case from 2000 samples. The follows graphics shows this process; where the right figure represents the reduced graph. This method becomes true with the aim that the signal always be located in the same point. To obtain an important data set, the acquisition of measurements must be repetitive.

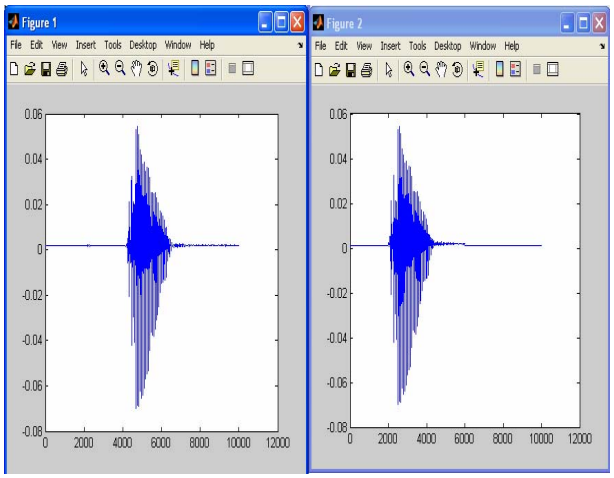

Figure 4. Signal preparation

In the next stage we proceed to find the signal energy in the time.

Two processes were developed in a algorithm of energy extraction: In the first, the energy was obtained according to the equation 1, the secondly, the energy gets back to normal so much in amplitude like in duration. Where, E is the energy and X is the voice signal. The figure 5 shows the signal energy of the "close command".

This process is true for commands To Open, to close, lights and television".

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

We apply the data analysis techniques of one-dimensional for discrete signals in the time, the characteristics in each one of the words that form the data set are faculty of speech. A total of 120 measures were acquired for 4 words (open, close, lights and television), which correspond each of them to 30 measures. In order to train the system the neural

network toolbox of Matlab was used to create the model. In this case two-layer feed-forward network is created. The network's input corresponds to one data set of 120 measures, the first layer has ten neurons; the second layer has one neuron where the network is simulated and its output acquired against the targets. Here the network is trained for 1000 epochs. Again the network's output is acquired.

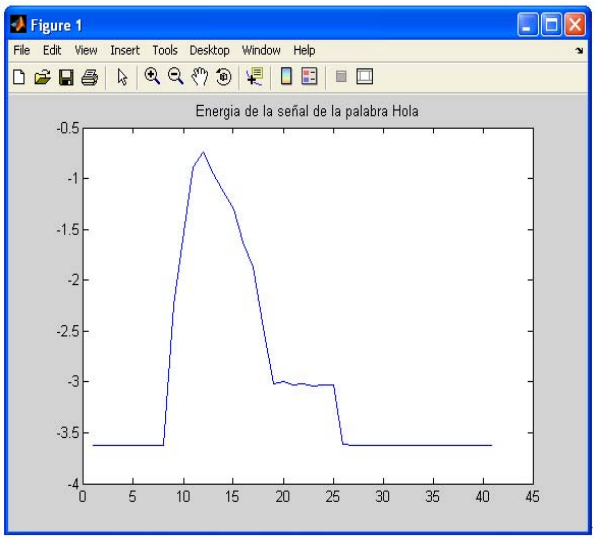

Figure 5. Energy of the signal

There are three different blocks in the recognition phase, to see figure 6, where the first one is the block to capture of the signal, and then we obtain their energy to be entered to the neural network input.

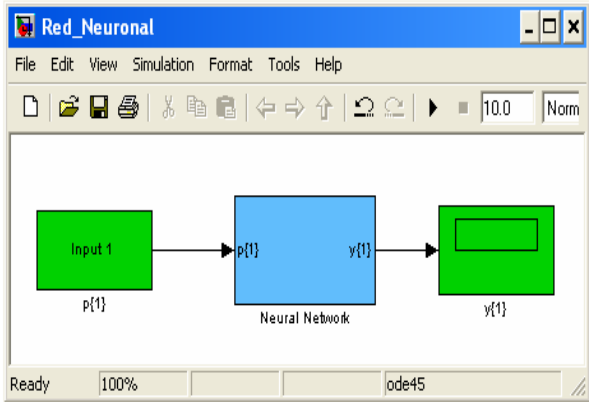

Figure 6. Simulation of the Net

The function "*wavrecord*" is used to acquire the voice signal, subsequently we obtain the energy of this command and this it will be kept in a vector L, so that from Simulink of Matlab the net takes the value of this vector to make the respective pattern recognition.

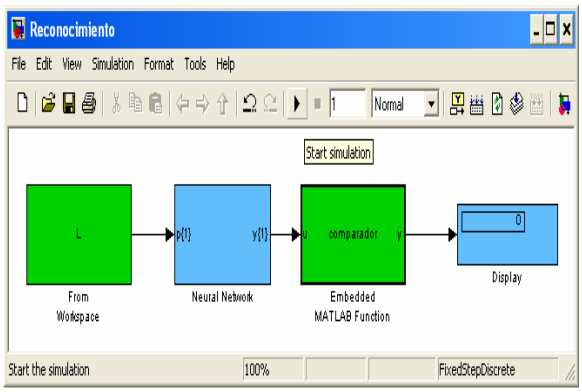

Figure 7. Pattern Recognition

The figure 7 shows the recognition algorithm, the first block takes the value of the vector L from *Workspace*, after we find the neural network that best make the recognition decision, while the third block is a function of the input value, and the output net is a number that can be a decimal value, for example: 1, 2, 3, or 4 depending of the command input.

2.3 Implementation on the Hardware DSP

The figure 8 shows a GUI, which each stage of the process is visualized from the capture of the signal until the pattern recognition. The GUI consists of three-button, the first one to capture the signal from the microphone, the next button to obtain the energy and the recognition of the voice command that it has been pronounced before. The energy that was obtained is entered to the net for the recognition.

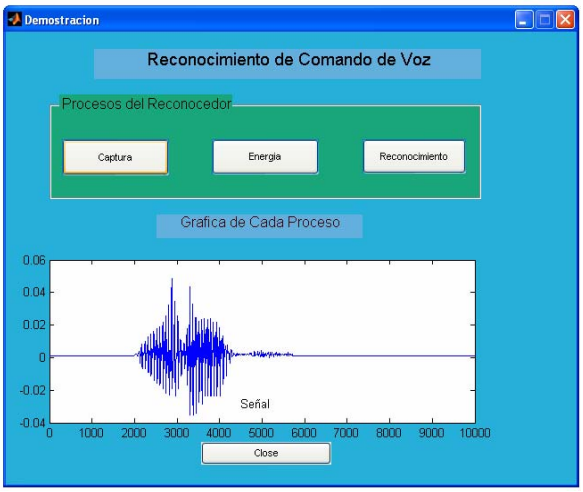

Figure 8. Interface of ASR

The last button of recognition to make the decision to know the speech command that was the one entered to the system. After having carried out the two

previous stages we will proceed to identify the command in the DSP target, this executes and building the model.

The figure 9 shows that automatically it generates the code and creates the model with respect to the pattern through of "*Code Composer Studio (CCS) Ver.3.0"* software developed by Texas Instruments. To achieve this, we used the power tool TLC (Target Language Compiler) from Simulink to interpreted language of Simulink Block to languaje C or assembler. This way we execute it in the DSK6416 of Texas Instruments, to see figure 10.

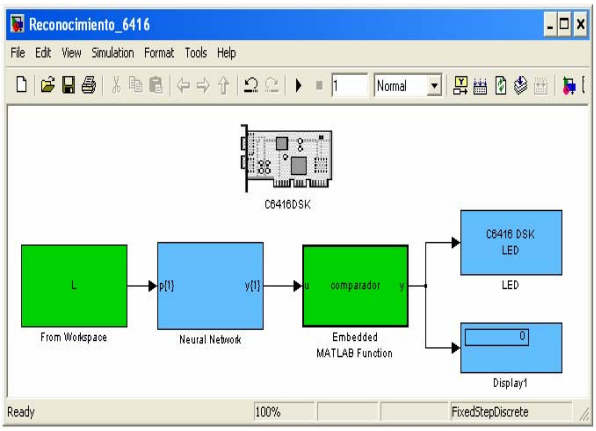

Figure 9. Recognition model to the DSK6416

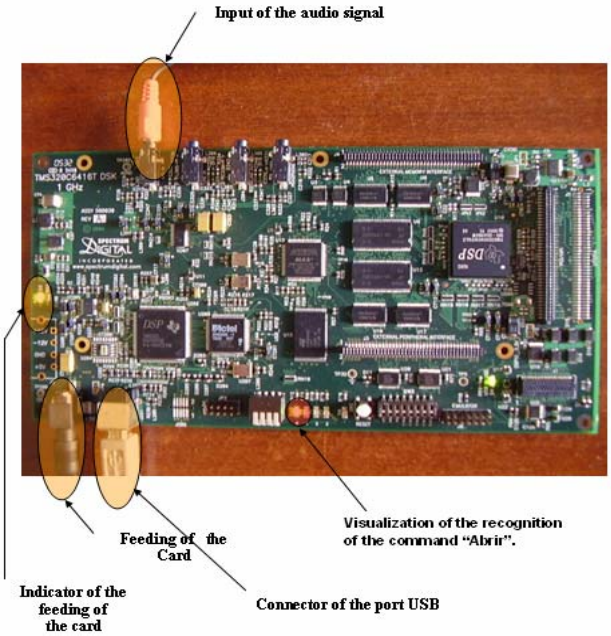

Figure 10. Speech Recognition on the Hardware DSP (DSK6416 TI)

3. Results

The recognition system was subjected for a group of 4 words (Open, Close, Lights and Television). The experiments were carried out pronouncing 30 times each one of the words and writing down the successes and mistakes, being obtained 98% of success rate of classification where only one error which was obtained in the "Ligths" command and located on the command "Close", to see the Table 1. 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 were made with the same characteristics with which is possible to train the automatic system.

Table 1. Results of Classification with 4 speech commands

Speech Command	1	2	3	4
To Open(1)	30	-	-	-
To Close(2)	-	30	Error	-
Ligths(3)	-	-	29	-
Television(4)	-	-	-	30

4. Conclusions

We demonstrate that is possible to implement a speech recognition algorithm such as Backpropagation neural network in a DSP (Digital Signal processing) DSK TMS320C6416T of Texas Instruments.

A technique of automatic speech recognition approach was developed which responds alone to the remarkable characteristics of the voice of a person in particular contemplating the recognition of isolated words.

Is important to say that although initially the tested were made with single 4 words, in a future work is possible to make tests with a large data set of voice command of training.

5. References

[1] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont *, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, C. Wellekens Multitel, Parc Initialis, *"Automatic speech recognition and speech variability: A review"* Avenue Copernic, B-7000 Mons, Belgium, In Press, accepted 6 February 2007.

[2] C. Andre, B. Jon "*An automatic speech recognition system based on the scene analysis account of auditory perception*" Department of Computer Science, University of She.eld, 211 Portobello Street, Sheeld S1 4DP, United Kingdom, 7 November 2006.

[3] Wald, M. *"Developments in technology to increase access to education for deaf and hard of hearing students"*. In: Proceedings of CSUN Conference Technology and Persons with Disabilities. California State University Northridge, 1999.

[4] Leitch, D. MacMillan, T. *"Improving Access for Persons with Disabilities in Higher Education Using Speech Recognition Technology"*. *Liberated Learning Project Year II Progress Report*. Saint Mary's University, Nova Scotia, 2001.

[5] M.R. Schroeder and B.S. Atal, *"Code-excited Linear Prediction (CELP):* High quality speech at very low bit rates", Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Processing, pp.937-940, 1985.

[6] S.Kwong and K.F Man, *"A Speech Coding Algorithm based on Predictive Coding*", Department of Computer Science, City University of Hong Kong 1995, pp.455.

[7] Coding Yuan Y. Tang, Tao L and Ching Y. Suenl "*VLSI Arrays for Speech Processing with Linear Predictive",* Centre for Pattern Recognition and Machine Intelligence Department of Computer Science Concordia University, Chongqing University Chongqing 630044, P. R. China.

[8] Neural Network Toolbox, Matlab_7.1, www.mathworks.com, Reference: Backpropagation feed-forward Multilayer.

[9] Digital Signal Processing, www.ti.com, Reference: DSK6416T-TMS320C6416T.