Speech Recognition Algorithm Implemented with a DSP

Authors: ***Constantin FILOTE***, e-mail: filote@eed.usv.ro
Adrian GRAUR, e-mail: adriang@eed.usv.ro
Gabriel ANTONESEI
Ovidiu OBADĂ
tel:(40) 30.216297, fax: (40) 30.520277,
Faculty of Electrical Engineering, *"Ştefan cel Mare"* University of Suceava,
1 University Street, 5800, Suceava, România.

Abstract: The aim of this paper is to describe a speech recognition algorithm. We describe the structure of a speaker-dependent system for sounds recognition. The implementation of this algorithm with a DSP allow to design and construct a vehicle that can be fully controlled from human generated sounds. More exactly we can command to our car to go forward, to left or right and to stop.

Keywords: Speech recognition , DFT algorithm, DSP.

1. INTRODUCTION

Most application of today involve the use of discrete time technological of processing continuous-time signals. One important class of signal processing problems is signal interpretation. The objective of the processing is not to obtain an output signals but to obtain a characterization of the input signal.

In a speech recognition system, the objective is interpret the input signal or extract information from it. Typically, such a system will apply preprocessing (filtering, parameter to estimation, Fourier transform etc.) followed by a pattern recognition system.

Speech recognition research and development has several goals. Simplifying the interface between users and machine is one major goal. Just as many users consider the mouse an improvement to the user interface on a personal computer, machine speech recognition and understanding has the potential to gre atly simplify the way people works with machines. Examples of this emerging technology include dialing telephones and controlling consumer electronic through voice-activation. As voice input and output become further integrated into the everyday machines, many advances will be possible.

Speech recognition systems fall into two categories:

- *Speaker dependent systems* that are used (and often trained) by one person (our car);
- *Speaker independent systems* that can be used by anyone.

2. VOICE PRODUCTION & MODELING

You can separate human speech production into two distinct sections: sound production and sound shaping. Sound production is caused by air passing across the vocal chords (as in "a", "e " or "o") or from a constriction in the vocal tract (as in "sss", "p" or "sh"). Sound production using the vocal chords is called voiced speech; unvoiced speech is produced by the tongue, lips, teeth, and mouth. In signal processing terminology, sound production is called excitation.

Sound shaping is a combination of the vocal tract, the placement of the tongue, lips, teeth, and the nasal passage. For each fundamental sound, or phoneme, the shape of the vocal tract is somewhat different, leading to a different sound. In signal processing terminology, sound shaping is called filtering.

3. ALGORITHM DESCRIPTION

The theory behind speech recognition is relatively simple. First, the DSP acquires an input sound (word) and compares it to a library of stored sounds. Then the DSP selects the library sound that most closely matches the unknown input sound. The selected sound is the recognition result. Systems that follow this model have two distinct phases: training phase and recognition phase.

3.1 Training phase

When you train a system to recognize sounds, you first create a library of stored sounds. Each sound to be recognized is stored in a library. Once the library is built, the system training is complete, and the task of recognition can begin.

3.2 Recognition phase

Figure 1 show a block diagram of the recognition algorithm:

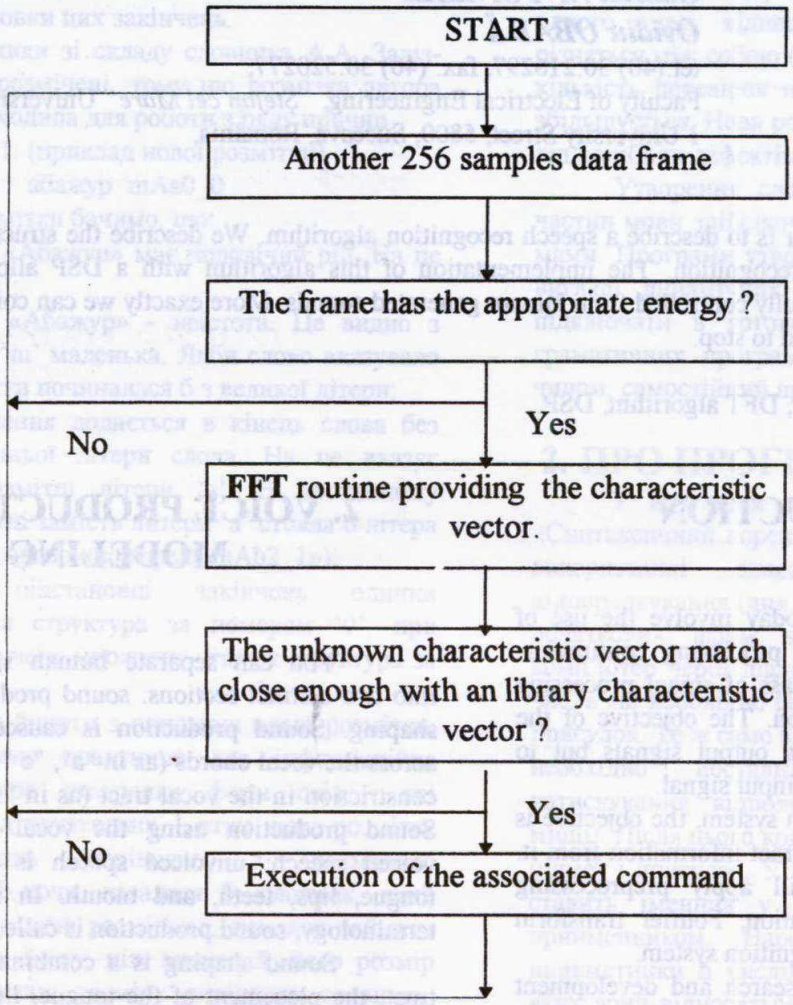


Figure 1 Speech Recognition Algorithm Diagram.

For the instant we resume to recognize fundamentals sounds such are the voices. We must mention that the same voice, but pronounced in two different words can be in her tour different. So, for the beginning we can recognize words by founding different voices.

The analogue input signal provided by a microphone is converted into a digital one as a result of hold and sampling process at 8 kHz frequency.

The processor receive this signal, split him in frames of 256 data (32 ms) and process this frames.

Because the microphone provide signals continuously and this signals are not all the time human sounds (like the noise, the smashes, etc.), it will be inefficient to process each frame. That's the reason why we make a test of this frames, more exactly we check if the energy of frame is high enough. If she is we go further, if not we take another frame.

Almost all speech recognition process use a parametric representation of speech rather than the waveform itself as the basis for pattern recognition. These parameters usually carry the information about the short time spectrum of the signal.

Among the most popular representation, produced by various forms of signal analysis, are spectral coefficients (**DFTC**), cepstral coefficients (**CEPC**), and linear predictive coding coefficients (**LPC**):

- Fourier analysis (**DFT**) yields discrete frequencies over time:

$$S(f) = \sum_{n=0}^{Ns-1} s(n) e^{-j2\pi n \frac{f}{f_s}} \quad (1)$$

where f_s is the sampling frequency and N_s is the length of the analysis sequence;

- linear predictive coding (**LPC**) yields coefficients of a linear equation that approximate the recent history of the raw speech values;

- cepstral analysis [1] calculates the inverse Fourier transform of the logarithm of the power spectrum of the signal:

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log\left|\hat{X}(e^{j\omega})\right| e^{j\omega n} d\omega \quad (2)$$

$$c(n) = \frac{1}{N_S} \sum_{k=0}^{Ns-1} \log|S(k)| e^{\frac{2\pi}{Ns} kn} \quad (3)$$

A signal observed for a finite interval of time (window) may have distorted spectral information in the Fourier transform. To avoid or minimize distortion, a signal is multiplied by a window-weighting function before the DFT is performed. Window choice is crucial for separation of spectral components. Our application use a 32 ms Hamming window to weight samples toward the center of the window.

The coefficients for the Hamming window [3] are obtained from the formula:

$$w(n) = \alpha + (1-\alpha)\cos\left(2\pi \frac{n}{N}\right) \quad (5)$$

commonly, $\alpha = 0.54$.

N = number of coefficients

Range: $$n = -\frac{N}{2} \text{ to } n = \frac{N}{2} - 1 \quad (6)$$

The DFT of Hamming Window Function is:

$$W(\theta) = \alpha D(\theta) + \frac{1}{2}(1-\alpha)\left[D(\theta - \frac{2\pi}{N}) + D(\theta + \frac{2\pi}{N})\right] \quad (7)$$

where,

$$\theta = 2\pi \frac{k}{N} \quad (8)$$

and

$$D(\theta) = exp\left(\frac{j\theta}{2}\right) \frac{sin\left(N\frac{\theta}{2}\right)}{sin\left(\frac{\theta}{2}\right)} \quad (9)$$

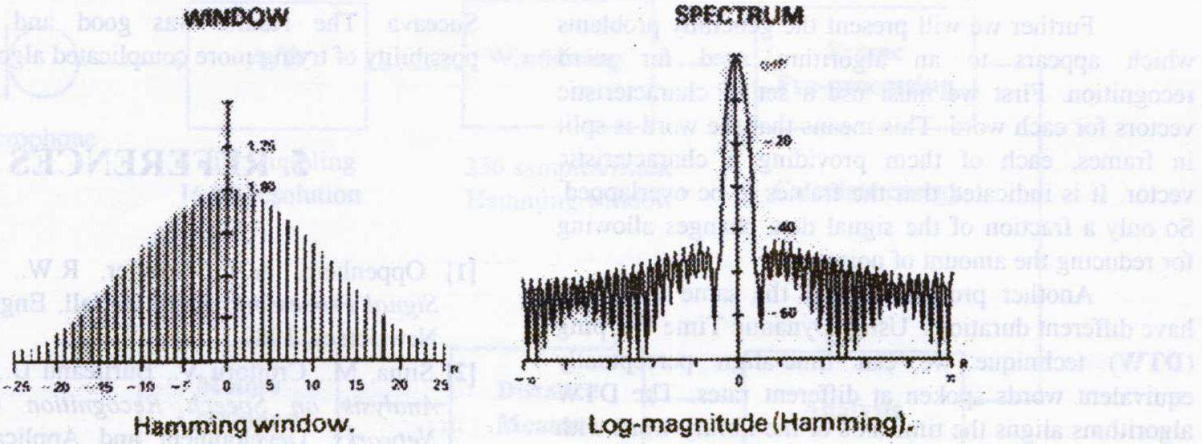


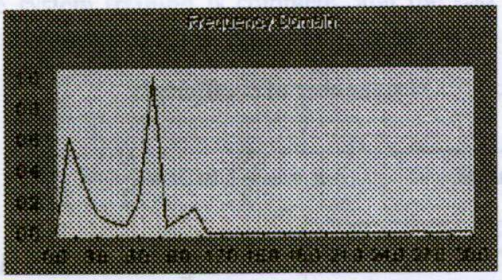
Figure 2 Characteristic of the Hamming Window.

Then we perform a 256 points FFT routine (Fast Fourier Transform). This routine provide a 128 points data vector containing information about the frequencies of the frame (in fact it contains the coefficients of the FFT). We do some operations with this vector in order to remove unnecessary information. We will name it the characteristic vector.

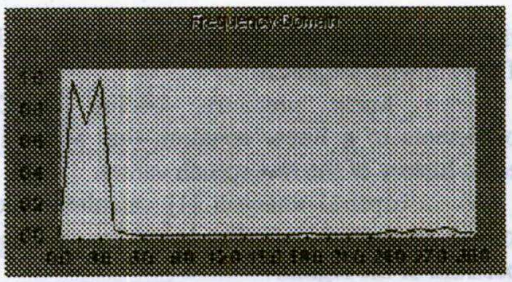
We compare this vector with the library created in the training phase. The distance between vectors can be determinate in many ways: the absolute distance, the euclidian distance, etc. I used the absolute distance.

Then the DSP selects the library vector that most closely matches the unknown vector and perform the associated command.

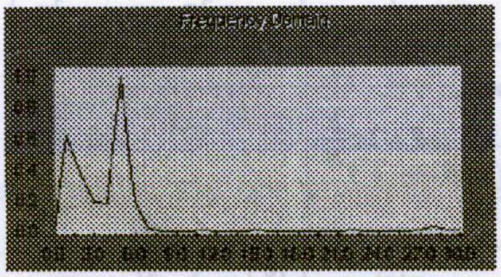
In figure 3 you can see the characteristic vectors of the voices that we use in our application.



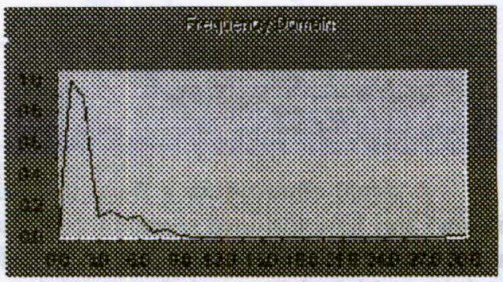
Characteristic vector of sound "a" from word UP



Characteristic vector of sound "i" form word PICK



Characteristic vector of sound "e" from word LEFT



Characteristic vector of sound "o" from word STOP

Figure 3 DFT caracteristic of some sounds.

Further we will present the generally problems which appears to an algorithm used for word recognition. First we must use a set of characteristic vectors for each word. This means that the word is split in frames, each of them providing a characteristic vector. It is indicated that the frames to be overlapped. So only a fraction of the signal data changes allowing for reducing the amount of noise.

Another problem is that the same word can have different durations. Using Dynamic Time Warping (**DTW**) technique we can time-align perceptually equivalent words spoken at different rates. The DTW algorithms aligns the time axis of the library word with the time axis of the unknown word.

4. CONCLUSIONS

The presented speech recognition algorithm a was implemented on an fully controlled by human sounds vehicle realized in the Digital Signal Processing Laboratory of Electrical Engineering Department from Suceava. The results was good and give us the possibility of trying more complicated algorithms.

5. REFERENCES

[1] Oppenhein, A.V., Schafer, R.W., *Discrete-time Signal Processing,* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[2] Sima, M., Croitoru V., Burileanu D., *Performance Analysis on Speech Recognition using Neural Networks,* Development and Application Systems (D&AS '98), Suceava, may 21-23, 1998, pp. 259-266.

[3] Higgins, R. J, *Digital Signal Processing in VLSI,* Prentice-Hall, Enlewood Cliffs, New Jersey, 1990.

[4] *Digital Signal Processing Applications using the ADSP-2100 Family*, Prentice-Hall, Inc. A division of Simon & Scuster Englewood Cliffs, New-Jersey 07632, 1992.