Abstract:

This paper treats automatic vocal verification systems depending on text, for embedded systems. The use of vocal recognition is relevant, can be used for security, control to restricted areas and bank transactions by telephone.

In this perspective, during the work, will be detailed three methods of extraction of vocal characteristics: LPC, Cepstral and Mel-Cepstral. It noteworthy, that the LPC was developed and tested as a phase of the project of an IP core identity vocal, SPVR (Speaker Verification System).

Extraction methods of vocal characteristics for Embedded Systems

I. INTRODUCTION

This paper is about embedded systems for automatic verification of voice identity.

Voice identification can be used for security, control to restrict areas and banking transactions by phone. However, some factors can interfere in the recognition of the speaker, such as: noise and changes at the microphone position [4].

Feature extraction, consists in estimating relevant voice parameters. Defining a good feature set is essential [9]. This paper performs an analysis of three methods of feature extraction suited for dedicated IP cores: LPC (Linear Prediction Code), Cepstral and Mel-Cepstral, with respect to false acceptance rate, false rejection rate and the repetition rate of the speaker's sentence [9]. The LPC was developed and tested in FPGA, as a part of the SPVR project (Speaker Verification System) [5] of Federal University of Campina Grande, Brazil (UFCG).

Finally, in the next sections, will be presented theoretical methods of feature extraction, the development process of embedded system, IpProcess [6], the functional architecture of SPVR, and then, the results of extraction methods deployed.

II. THEORETICAL FUNDAMENTATION

This section defines the concepts that form part of the theoretical background necessary for understanding the research proposal presented in the introduction.

A. LPC, Cepstral, Mel-Cepstral

One of the simplest techniques of feature extraction is the LPC, providing a good reduced representation of speech parameters that can be transmitted at low rates in voice encoding, needing and low computing power for its calculation [13]. The LPC represents each voice sample as a linear combination of N previous samples, the higher the N value the more precise is the model.

The digital model of speech production that is right for using linear prediction is shown below:

![Figure 1. Block diagram of voice production based on the LPC model](image)

The vocal folds in this model would be equivalent to a pulse generator in that the intensity of the signal of the sound is determined by a gain G. The vocal tract is modeled as a linear filter H(z) also called synthesis filter. When a speech segment is not audible, it's considered that the vocal tract is excited by a noise, that equates to the key in figure one.

The parameters of the filter H(z) are estimated from an acoustic or deaf signal, using the
LPC technique in which the modulus of the spectrum of the impulse response, $H(z)$, corresponds to the spectral envelopment of the signal.

To obtain the LPC coefficients, several methods can be used, such as covariance [1], autocorrelation [7][12], inverse filter [11], spectral estimation [11], maximum likelihood [11] and inner product.

The Cepstrum is the transformation of the signal frequency spectrum at short time intervals. The Cepstral coefficients are used to describe the spectral envelope of the voice signal at short time intervals. One of the main advantages of Cepstral coefficients lies in the fact that these are generally uncorrelated [5].

There are two ways for obtain Cepstral coefficients: FFT (Fast Fourrier Transform) and LPC [14]. In the Cepstral analysis by FFT, FFT is directly applied to the voice signal. The i-th term, $c_e(n)$, it's calculated by [14]:

$$c_e(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log_{10} S_i(e^{j\omega}) d\omega$$

where $-\pi < n < \pi$ and $S_i$ shows the i-th block from the spectrum of power voice signal.

In the cepstral analysis by LPC, the $Z$ transform in the voice signal. The Cepstral coefficients spectrum, obtained from LPC analysis, $c_i$, can be recursively calculated by [14][8]:

$$c_i(1) = c_i(1),$$

$$c_i(n) = c_i(n) + \sum_{j=1}^{n-1} (1 - \frac{1}{n}) c_i(j) c_e(n-j),$$

where $n$ is coefficients index and $i$ is sample block's index.

Using this recursive relation leads us to efficient computation of cepstral coefficients, $c_e(n)$, and avoids factoring polynomial.

When a frame contains sound information, its first cepstral coefficients show a significant change in amplitude and periodically local maxima appear, we can check this in the figure below:

The same does not happen when frames are formed by components deaf of the signal, as shown in the figure 3. In this case, the local maxima leave to exist leaving only the first coefficients with significant amplitude.

The Mel's scale intends to approach human ear's characteristics. Human's perception from pure tones or voice signals does not follow a linear scale. For each tone with a frequency a mid value is associated in Mel's scale. Mel is a frequency unity. It's reference is defined at 1 kHz frequency, with intensity 40dB over the human's hearing threshold, as 1000 mels [9].

The mapping from Hertz frequency's to Mel's frequency follows this equation:

$$F_{\text{mel}} = 2595 \log_{10}(1 + \frac{F_{\text{linear}}(\text{Hz})}{700})$$

The values 2595 and 700 are empirically obtained, using these values we can verify the relationship between the linear scale and Mel scale from after figure.

In this section are presented the development methodology used to develop the IP core identity vocal (SPVR) and functional characteristics of embedded system.
A. Embedded Systems for Speaker Verification

For this system construction is needed that the kind of application be decided before, because embedded systems have architecture and limitations different from a computer. For this application a small scale of chip prototypes become a important fase of utilization of the FPGA (Field Programmable Gate Array).

The design flow comprises the following steps: specification of functional verification, testbench implementation, DUV (Designer Under Verification) implementation, synthesis, post-layout simulation, prototyping and silicon validation.

**Hardware specification:** high level documents even when no decisions have been made in relation to the implementation of the functionality.

**Specification of the functional verification:** Making a plan of verification that must contain information about the functionalities that will be “covered “, and types of stimulus used, the value range of stimuli, variable names.

**Testbench Implementation:** Construction of the ambient in which the pospective should be puted for verification, implemented in high level of abstraction, which is denominated transaction level. The testbench must have a module that will generate stimulus which will be used for verification. The answers for these stimuli must be compared with the ideal answers. A good testbench must be able to make this comparison automatically.

**DUV Implementation:** The lowest abstraction level code, signals level, written in RTL (Register Transfers Language) level, done using hardware description languages as in this case using System Verilog.

**Functional Verification:** Through the simulation, stimuli are inserted on the entry of the DUV and the are collected values on its output, comparing the collected values on this simulation with the ones obtained on the ideal model.

**RTL Synthesis:** Consists on converting the RTL description into a set of registers and combinational logic.

**Pos-Synthesis Simulation:** After the RTL code synthesis is necessary to make a new simulation. By the way this simulation considers hardware implementations problems such as, logic gate delay.

**Prototyping:** This phase occurs the netlist implementation obtained in synthesis fase in some hardware device. In this case the prototyping was made first in FPGA and after in silicon.

The adequate utilization of languages in the design flow can reduce costs, reduce time in the execution of the project and reduce the amount of design flaws. In this case System Verilog, was used because it's a hardware description language (HDL) as well as a hardware verification language (HVL).

In this paper BVM methodology (Brazil IP Verification Methodology) [10] was employed, which is derived from OVM (Open Verification Methodology) [10] for developing the speaker verification system (SPVR) and, it was verified that the Functional Verification phase consumes more than 70% total project time, but it's useful, because it minimizes errors, but can't prove non-existence from them [2].

B. SPVR (Speaker Verification System)

The IP core architecture developed (SPVR) is shown in figure 5.

![Figure 5. Architecture of SPVR](image)

In which first component of the architecture, Voice Detector (VD), has the functionality to eliminate parts of the speech signal that contain no significant information of the speaker, for example, ambient noise. Initially the Voice Detector receives characteristics of time and energy to start the detection of segments that present voice. After that, calculates the energy contained in a segment that is composed of a fixed number of speech samples.

To eliminate spectral tendencies coming from the speech coming from the lips has developed the Pre-Emphasis (PE), consisting of a filter, response of approximately +6 db/eighth, providing a leveling at spectrum. In this synthesis, the mathematical description of Pre-Emphasis is given by:

\[ S_p(n) = S(n) - \alpha S(n-1), \quad 0.9 \leq \alpha \leq 1, \]

wherein \( S_p(n) \) sample is pre-stressed and \( S(n) \) is the original sample.

The Windowing (WIN) divides the speech signal into segments or frames by multiplying the samples by values of the Hamming window, prioritizing the spectral characteristics at the center of the segment and associating lower weights to the extremities of the segment. After pre-processing the speech signal, step performed by the three previous modules, at that moment, it is calculated the module of the Fourier transform of the frames containing useful information, functionality module Fast Fourier Transform (FFT), that has passed its power
spectra for the module Mel-Cepstrais, with the purpose of obtaining coefficients mel-cepstrais of each speaker.

The module Pattern Matching (PM) is responsible for calculating the distance, called distortion, between the coefficients Mel-Cepstrais and the codebook, by calculating the extent of distortion of the mean square error. The last module, the Decision Maker (DM), receives distances of the PM, calculates the euclidean median between the distances received and takes a decision based on the following rule: if the euclidean median is lower than the threshold $L_1$, the speaker is accepted. If the median gets higher than the threshold $L_2$, the speaker is rejected. However, if the median value is between $L_1$ and $L_2$, the speaker is considered unknown and is requested to repeat his vocal password. In the end after the description of the functionalities of each block from SPVR, the IP core layout was projected, as presented on figure 6:

![Figure 6. SPVR Layout](image)

For the synthesis was used the Quartus II 9.1 Build 222, even as for prototyping was used the Altera DE2 board and obtained the following results presented on table 1.

<table>
<thead>
<tr>
<th>Components</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic elements</td>
<td>30,819</td>
</tr>
<tr>
<td>Combinational functions</td>
<td>21,300</td>
</tr>
<tr>
<td>Dedicated logic register</td>
<td>19,627</td>
</tr>
<tr>
<td>Memory bits</td>
<td>69,376</td>
</tr>
<tr>
<td>Embedded Multiplier 9-bit elements</td>
<td>44</td>
</tr>
</tbody>
</table>

Table 1. Compilation Results

IV. RESULTS

Aiming to analyze the capacity of verification of the three feature extraction methods, initially were captured voice samples of 11 speakers. To realize the training phase and after obtaining the codebook, each speaker repeated his sentence 10 times obtained utterances of two training sentences that belongs to a speaker [9], with duration between 30 and 70 seconds.

![Figure 7. Utterance of the training sentence](image)

In the verification phase, the coefficients obtained from speakers during the training phase were compared with the vectors of coefficients obtained in this phase. Each speaker repeated their sentence ten times, but, the repetitions were stored in separate files.

Using Mel-Cepstral or Cepstral, no speaker is rejected when speaking his own sentence. Using LPC coefficients, this does no happen, the system rejects the speaker even when he is speaking his vocal keyword [9] in 4.55% of cases. This also depends on the acceptance thresholds that are used, as the developer of the system can raise the robustness, through the reduction of the lower threshold.

Based in simulations made in PC and the results shown before, the Mel-Cepstrais coefficients exhibit better results compared with LPC coefficients and Cepstral coefficient. This can be achieved through the use of the bank of filters whose center frequencies correspond to those in which they perceive the changes of tone and its triangular shape allows the stress components in center frequencies, attenuating the other.

V. CONCLUSION

The work aims a study of three methods of feature extraction, to make this possible we used three speaker verification systems. The systems were strictly equal the steps of: pre-processing and pattern generation. The only difference was in the method of extraction of features that could be: LPC, Cepstral or Mel-cepstral.

Based on simulations on PC and the results presented, the coefficients Mel-Cepstrais show better results when compared with the LPC coefficients and Cepstral coefficients. This can be achieved through the use of the bank of filters whose center frequencies correspond to those in which they perceive the changes of tone and its triangular...
shape allows the stress components in center frequencies, attenuating the other.

Although the results are considered very good you need to check the performance of the three methods in environments with a high noise level, and compare their performance when training and testing are made in different environments.

However, when the focus is passed on to the cost of making the implementation, it can be seen that a system using coefficients Mel-Cepstral have a higher cost than other systems of feature extraction (LPC and Cepstral).

Finally, compared to traditional systems, the IP core for speaker verification - SPVR - comprises a viable solution to the security context by providing not only greater reliability, as well as facility of reuse and, consequently, fast incorporation to other supplementary systems, a fact that is really relevant in the modern world. The succeed chip implementation and its high hit rate demonstrate the effectiveness of the design methodologies adopted. The entire digital signal processing demanded was part of the chip development, which eliminates the necessity of external signal processors and makes the SPVR a robust and a self-sufficient hardware for collaborative users.

REFERENCES


