SPEAKER VERIFICATION FOR ROMANIAN LANGUAGE

C.O. DUMITRU, Inge GAVĂT *

În acest articol se prezintă un sistem de verificarea vorbitorului cu fraze fixe pentru limba română. În sistemul propus extragerea parametrilor se bazează pe analiza cepstrală, obținându-se coeficienții mel cepstrali. Modelarea acustică a vorbirii este bazată pe modelele Markov ascunse dependente de context; pentru fiecare vorbitor se construieşte câte un model, prin antrenare cu date de la acel vorbitor. Pentru evaluarea performanțele sistemului, se testează fiecare model cu toți vorbitorii, determinandu-se astfel ratele de corectă şi falsă acceptare. Rezultatele sunt încurajatoare, fiind obținută o eroarea de falsă acceptare de 4%. Experimentele au fost făcute pe baza proprie de date, conținând semnal vocal de la zece vorbitori(8 vorbitori masculini şi 2 vorbitori feminini), fiecare vorbitor rostind 50 de fraze.

In this paper we present a speaker verification system with fixed phrase text for Romanian language. The feature extraction in the system is based on perceptual cepstral analysis, giving the mel-frequency cepstral coefficients (MFCC). The acoustical modeling of speech in the statistical framework is based on hidden Markov models (HMM) with context dependent modeling; a model for each speaker is built, by training with his own data. To evaluate the system performance, each model was tested with all speakers, obtaining the rates for correct and false acceptance. The results are promising, a false acceptance error of 4% being achieved. The experiments were made using our own database, containing speech data from ten speakers (8 male speakers and 2 female speakers), and each speaker reading 50 utterances.

Keywords: speaker verification, MFCC, HMM, monophone, triphone**.**

Introduction

Speaker verification is defined as a decision process determining if a speaker is who he claims to be. This is a different case than the speaker identification problem, which is deciding if a speaker is a specific person or is among a group of persons. In speaker verification, a person makes an identity claim entering an employee number, presenting his smart card or uttering a text. In text-dependent systems, the text is known and it can be fixed or optional, prompted visually or orally. The claimant speaks the text into a microphone. This signal is analyzed by a verification system that makes the binary decision to

* Prof., Ph.D., Dept. of Applied Electronics and Information Engineering, Faculty of Electronics Telecommunication and Information Technology, University "Politehnica" of Bucharest, ROMANIA, igavat@alpha.imag.pub.ro.

accept or reject the user's identity claim or possibly to report insufficient confidence and request additional input before making the decision [1], [2].

Speaker verification applications include access control, telephone banking, or telephone credit cards.

As automatic speaker verification systems gains widespread use, it is imperative to understand the errors made by these systems. There are two types of errors: the false acceptance of an invalid user and the false rejection of a valid user. It takes a pair of subjects to make a false acceptance error: an impostor and a target [3], [4].

In this paper we will use for speaker verification the pattern recognition approach, extracting features from speakers utterances and than evaluate the match with trained models.

For speaker verification, features that exhibit high speaker discrimination power, high inter-speaker variability, and low intra-speaker variability are desired [5], [6]. These conditions are fulfilled by the mel-frequency cepstral coefficients (MFCC), and therefore we chose them in our feature extraction part of the system.

The paper is structured as follows: chapter 1 is dedicated to the speech parameterization. In chapter 2, we present the HMMs with context dependent modeling. Databases and the system interface are exposed in chapter 3 and 4. Experimental results are explained in chapter 5. Conclusions and references are closing the paper.

1. Speech parameterization

Cepstral analysis is a very reliable method to speech parameterization and it can be realized applying the blocked and windowed time discrete signal *s (n)* to the processing chain depicted in Fig. 1. After the FFT, the modulus of the signal is calculated and the logarithm is taken, the result being proportional in fact to the power spectrum of the speech signal.

Through IFFT, the real cepstrum is obtained, the "filter" characterization being comprised near of the cepstrum origin. The re-sampling of the real cepstrum leads to the cepstral coefficients, which, alone or in addition with the energy E, and/or the first and second order differences constitute a feature vector successfully applied in speech recognition [7], [8].

In order to obtain a parametric representation with the mel-frequency cepstral coefficients and their first and second order variations, the power spectrum is processed using a set of filters with the transfer functions represented in the Fig. 2. That filter bank is a model for the critical band perception of the human cochlea [9], [10].

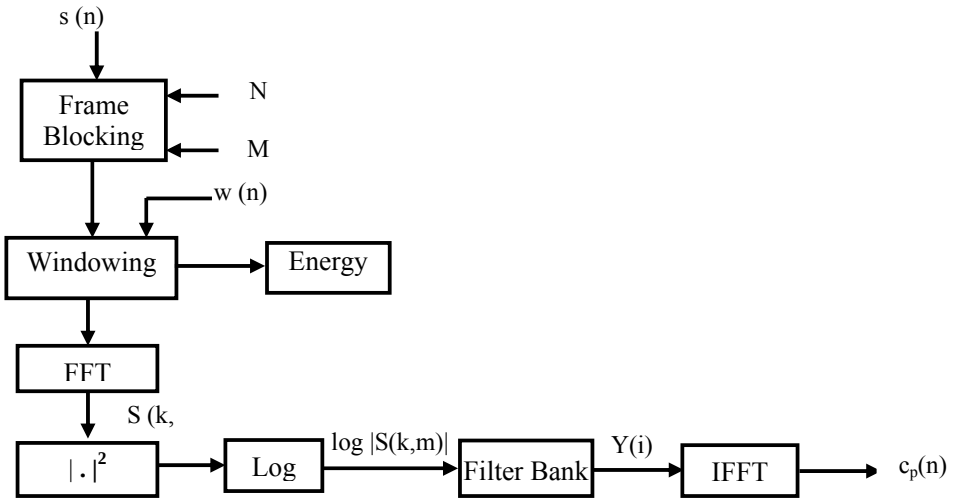

Fig. 1. Processing chain.

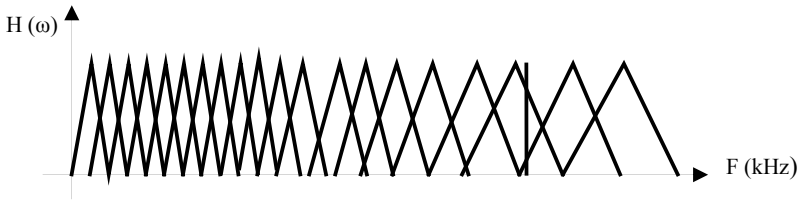

Fig. 2 The transfer functions of filters.

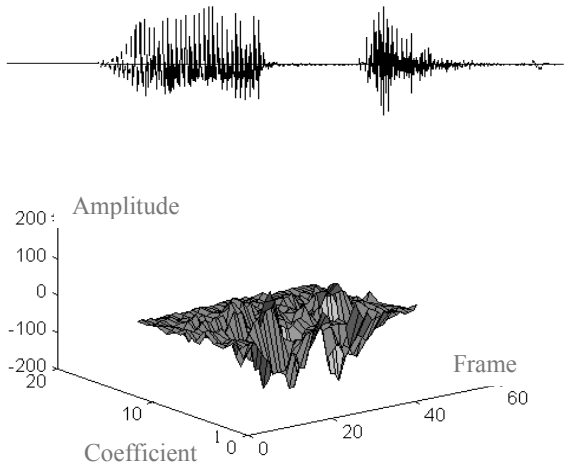

Fig. 3. The Romanian word “apa” and the cepstral coefficients.

The outputs of the filters are calculated by the formula (1), where i is the number of the filter.

$$Y(i) = \sum_{k=0}^{N/2} \log |S(k,m)| H_i(k \frac{2\pi}{N}) \quad (1)$$

The first and the second order variations of the mel-frequency cepstral coefficients are used for speech recorded in noisy environments or under the influence of stress or emotional factors [11], [12].

In Fig. 3 it is represented the variation of the 12 mel-frequency cepstral coefficients along 60 frames for the word "apa", calculated with the processing chain in Fig. 1.

2. Hidden Markov models (HMM)

HMMs are finite automates, with a given number of states; passing from one state to another, is made instantaneously at equally spaced time moments. At every passing from one state to another the system generates observations, into the automate two processes taking place: the transparent one and the hidden one, which can not be observed, first represented by the observations string (parameter sequence) and second, represented by the states string [7].

As concerns the HMMs, there are three main problems:

The first problem is the evaluation one. Given the model and the observation (parameter) sequence, we have to analyze if the sequence is produced by the given model. The probability to produce an observation sequence with a Markov model is calculated by the "forward" and "backward" algorithm.

The second problem is about establishing the correct state sequence. The "Viterby" algorithm is one of the most used algorithms for this purpose.

The third problem is the parameter optimization of the model to describe as good as possible the observation sequence. Training allows optimal adaptation of the model parameters to training data by re-estimating them. The "Baum-Welch" algorithm is the most used model parameter re-estimation algorithm.

In Fig. 4 it is represented the left - right model (Bakis), which is considered the best choice for speech. For each phoneme, called monophone, such a model is constructed; a word string is obtained by connecting corresponding HMMs together in sequence, the extension to continuous speech being simply realized.

2.1. Context – dependent modeling.

In the simplified hypothesis of context-independent phoneme modeling each word results as a concatenation of the component phonemes; for each

phoneme a model is constructed. In Romanian language, as phoneticians claim, there are 34 phonemes, requiring 34 different models.

In real speech, the words are not simple strings of independent phonemes: as effect of co-articulation, the immediate neighbor – phonemes, for instance the preceding and the following one, affect each phoneme in the word.

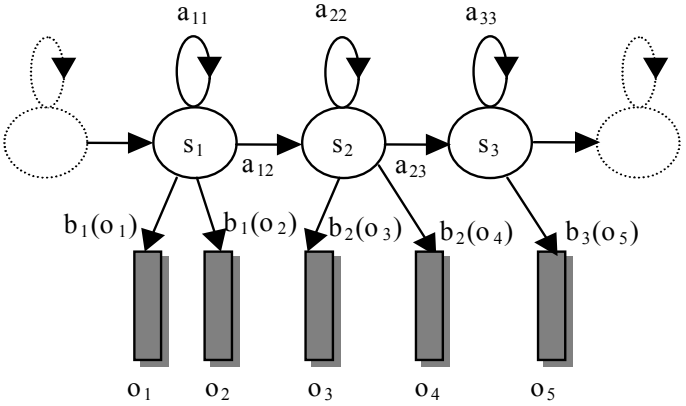

Fig. 4: Left-right model or Bakis model

This immediate neighbor – phonemes are called respectively the left and the right context; a phoneme constitutes with the left and right context a triphone. For example in the triphone "a - z + i_o", (SAMPA- Speech Assessment Methods Phonetic Alphabet [13] - transcription for the Romanian word "azi"), the phoneme "z" has as left context "a" and as right context "i_o", like is shown in Fig. 5.

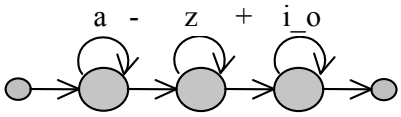

Fig. 5. The word internal triphone "a - z + i_o"

For each such a triphone a model must be trained: in Romanian that will give a number which equals $34^3 = 39304$ models, which is totally unacceptable for a real time system. In our speaker verification task we have modeled only internal – word triphones and the adopted state tying procedure has conducted to a controllable situation [14].

2.2. State Tying on Phonetic Decision Trees

If triphones are used in place of monophones, the number of needed models increases and it may occur the problem of insufficient training data. To

solve this problem, tying of acoustically similar states of the models built for triphones corresponding to each context is an efficient solution. For example, in Fig. 6a four models are represented for four different contexts of the phoneme "a", namely the triphones "k – a + S", "g – a + z", "n – a + j", "m – a + j".
In Fig. 6b, are represented the clusters formed with acoustically similar states of the corresponding HMMs.

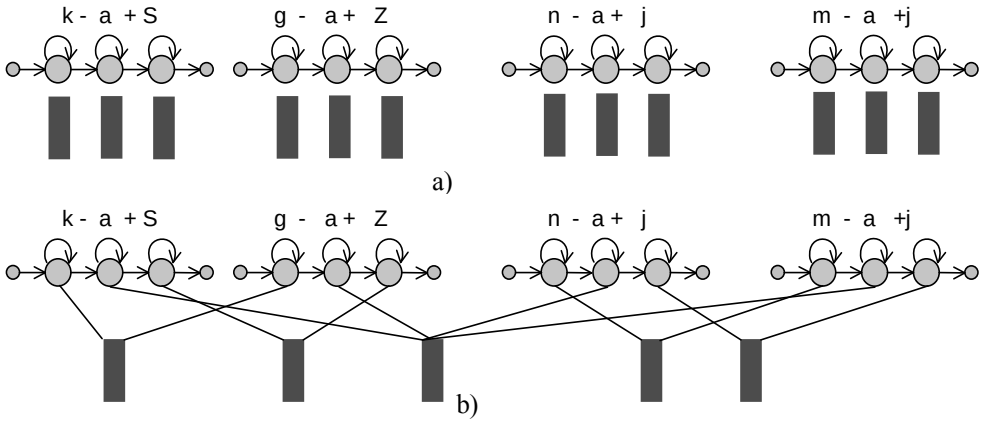

Fig. 6. a) Different models for triphones around the phoneme "a",
b) Tying of acoustically similar states.

The choice of the states and the clustering in phonetic classes are achieved by mean of phonetic decision trees. A phonetic decision tree, built as a binary tree is shown in Fig. 7 and has in the root node all the training frames to be tied, in other words all the contexts of a phoneme. To each node of the tree, beginning with the parent – nodes, a question q_i is associated concerning the contexts of the phoneme [15].

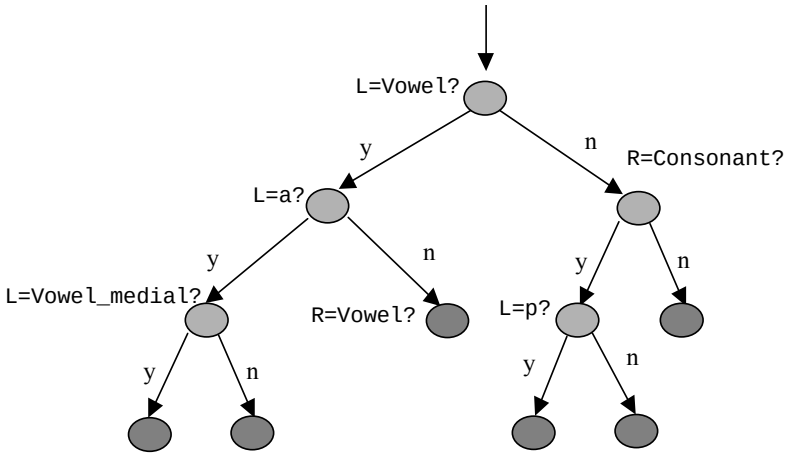

Fig. 7. Phonetic tree for phoneme m in state 2

Possible questions are, for example: is the right context a "consonant" (R = Consonant?), is the left context a phoneme "a" (L = a?); the first answer designates a large class of phonemes, the second only a single phonetic element. Depending on the answer, yes or no, child nodes are created and the frames are placed in them. New questions are further made for the child nodes, and the frames are divided again.

The questions are chosen in order to increase the *log likelihood* of the data after splitting. Splitting is stopped when increasing in *log likelihood* is less than an imposed threshold, resulting a leaf node. In such leaf nodes are concentrated all states having the same answer to the question made along the path from the root node and therefore states reaching the same leaf node can be tied as regarded acoustically similar. For each leaf node pair the occupancy must be calculated in order to merge insufficient occupied leaf nodes [16], [17].

A decision tree is built for each state of each phoneme. The sequential top down construction of the decision trees was realized automatically, with an algorithm selecting the questions to be answered from a large set of 130 questions, established after knowledge about phonetic rules for Romanian language [18].

3. Database

Our experiments were carried out in a speaker verification task.

For speaker verification, the database is constituted from 500 phrases, uttered by 10 speakers, 8 males (speaker 1, 2, 3, 4, 5, 7, 9, 10) and 2 females (speaker 6, 8), each speaker reading 50 utterances [19], [20]. The speaker 1 and the speaker 10 are the same. The database for training contains 480 phrases, each speaker reading 48 phrases. The testing database contains 20 phrases uttered by the same speakers, each reading the same 2 phrases.

Examples of testing phrases are:

- Zero unu doi trei patru cinci şase şapte opt nouă
- Salvează pe cartela sim

The training database contains over 250 distinct words, while the testing database contains 14 distinct words.

The data are in the typical "wave" format, sampled by 16 kHz, quantized with 16 bits, recorded in a laboratory environment with a desktop microphone.

4. Interface

In order to simplify our work during the experiments, a MATLAB interface *"Speaker verification"* represented in Fig. 8 was designed. The interface enables the choice of the speaker for training and of the speaker for testing. Waveform visualization, opening and playing the sound are also possible options.

5. Experimental results

In our experiments we trained with the embedded procedure the models for each speaker and tested the trained models with each of the ten speakers. The results of our experiments are synthetically represented in the confusion matrix in Table 1, where the success rate is pointed out. The success rate for acceptance is 97%.

In general, the system provides good results.

But there are some false acceptance situations. The models trained with the first speaker accept the sixth one; we can consider the sixth speaker as an impostor. The models trained with the fifth and the sixth speaker respectively accept the seventh one; the models trained with the ninth speaker accept the third one. That is in connection with the fact that the acceptance level in success rate is very high, respectively 97%. There are chances to improve the situation by adding first order differences of the MFCCs to the feature vector or extending the training material. Informal tests made for extending the feature vector show some enhancements, the acceptance level becoming lower [21].

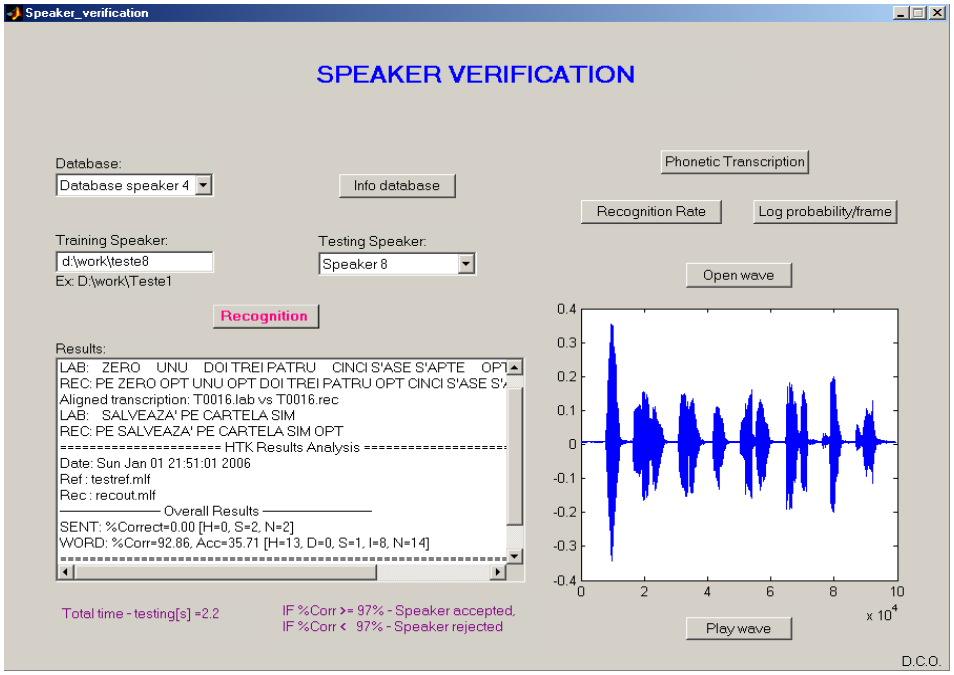

Fig. 8. Speaker verification interface.

Table 1

Experimental results

Train Speakers	Testing Speakers (Success Rate %)									
	1	2	3	4	5	6	7	8	9	10
1	100	71.43	71.42	42.86	78.57	100	92.86	78.57	57.14	100
2	85.71	100	85.71	71.42	85.71	78.57	92.86	78.57	85.71	78.57
3	92.86	85.71	100	92.86	92.86	64.29	92.86	57.14	92.86	78.57
4	50.00	64.29	92.86	100	71.43	42.86	100	28.57	85.71	64.29
5	78.57	71.43	92.86	71.43	100	85.71	100	50.00	92.86	78.57
6	78.57	50.00	57.14	50.00	71.43	100	78.57	78.57	78.57	92.86
7	92.86	71.43	71.43	71.43	71.43	64.29	100	64.29	64.29	92.86
8	92.86	78.57	78.57	42.86	71.43	92.86	71.43	100	71.43	92.86
9	71.43	85.71	100	78.57	85.71	57.14	92.86	64.29	100	85.71
10	100	78.57	85.71	64.29	85.71	92.86	92.86	71.43	92.86	100

Conclusions

The experiments we carry out with our speaker verification system lead to promising results.

In the system, feature extraction is based on perceptive cepstral analysis conducting to the mel-frequency cepstral coefficients (MFCCs).

The modeling technique applied to the statistical framework relies on HMMs, built for each monophones and triphones.

The success rate, actually the word recognition rate, must have a high threshold for acceptance (97%) and therefore some false acceptance situations appear (4% false acceptance error).

For our future work this performance can be enhanced, some informal tests giving good results by: enlarging the feature vector by adding first order differences to the MFCCs or by using PLP (Perceptual Linear Prediction) for feature vector [22].

REFERENCES

1. *P. Joseph, Jr. Campbell*, Speaker Recognition, Proceedings of the IEEE, Vol. 85, No. 9, September 1997, pp. 1437-1462.
2. *X. Huang, A. Acero and H.W. Hon*, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice-Hall, USA, 2001.
3. *Y. Bennani and P. Gallinari*, Connectionist approaches for automatic speaker recognition, Proc. 1st ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland, April 1994, pp. 95–102.
4. *Y. Gu and T. Thomas*, A text-independent speaker verification system using support vector machines classifier, Proc. European Conference on Speech Communication and Technology (Eurospeech ’01), Aalborg, Denmark, September 2001, pp. 1765–1769.

5. *A. E. Rosenberg, I. Magrin-Chagnolleau, S. Parthasarathy, and Q. Huang*, Speaker detection in broadcast speech databases, Proc. International Conf. on Spoken Language Processing (ICSLP '98), Sydney, Australia, December, 1998.
6. *C. Huang, T. Chen, E. Chang*, Speaker Selection Training For Large Vocabulary Continuous Speech Recognition, Proc. ICLSP 2002, Vol. 1, pp. 609-612.
7. *I. Gavat & others,* Elemente de Sinteza si Recunoasterea Vorbirii, Printech, Bucuresti, 2000.
8. *S. Furui*, Digital Speech Processing, Synthesis and Recognition, 2-end, rev and expanded Marcel Dekker, N.Y., 2000.
9. *B. Gold, N. Morgan*, Speech and audio signal processing, John Wiley and Sons, N.Y., 2002.
10. *X. Huang, A. Acero, H.W. Hon*, Spoken Language Processing – A Guide to Theory, Algorithm, and System Development, Prentice Hall, 2001.
11. *B.A. Milner*, Comparison of Front-End Configurations for Robust Speech Recognition, ICLSP 2002 Proceedings, Vol. 1, pp. 797-800.
12. *R D.Vergin, O'Shaughnessy, A. Farhat*, Generalized Mel-Frequency Cepstral Coefficients for Large Vocabulary Speaker Independent Continuous Speech Recognition, IEEE Trans. Speech Audio Processing, Vol. 7, No.5, pp. 525-532, 1999.
13. SAMPA - Speech Assessment Methods Phonetic Alphabet, http://www.phon.ucl.ac.uk/home/sampa/home.htm
14. *J.J. Odell*, The Use of Decision Trees with Context Sensitive Phoneme Modeling, MPhil Thesis, Cambridge University Engineering Department, 1992.
15. *P.C. Woodland, J.J. Odell, V. Valtchev, S.J. Young,* Large Vocabulary Continuous Speech Recognition Using HTK, Proc. ICASSP 1994, Adelaide.
16. *S.J. Young*, The General Use of Tying in Phoneme-Based HMM Speech Recognizers, Proc. ICASSP'92, Vol. 1, pp. 569-572, San Francisco, 1992.
17. *S.J. Young*. *J.J. Odell*, *P.C. Woodland*, Tree Based State Tying for High Accuracy Modeling, ARPA Workshop on Human Language Technology, Princeton, 1994.
18. *E. Oancea, I. Gavat, C.O. Dumitru, D. Munteanu*, Continuous Speech Recognition for Romanian Language Based on Context-Dependent Modeling, Proc. COMMUNICATION 2004, Bucharest, Romania, pp. 221-224, 2004.
19. *I. Gavat, C.O. Dumitru, G. Costache, D. Militaru*, Continuous Speech Recognition Based on Statistical Methods, Proc. SPED 2003, April 10-11, Bucharest, Romania, pp. 115-126, 2003.
20. *I. Gavat, C.O. Dumitru, D. Duduleanu, G. Costache,* A Comparative Study of Features for Continuous Speech Recognition in Real Word Conditions, Proc. Etc 2002, September 19-20, Timisoara, Romania, Tom 47(61), Fascicola 1,2, Vol I, pp.102-107, 2002.
21. *C.O. Dumitru, I. Gavat, R. Vieru,* Speaker verification using HMM for Romanian Language, The 48th International Symposium - ELMAR 2006, Zadar, Croatia, June 7-10, pp.131-134, 2006.
22. *H.Hermansky*, Perceptual Linear Predictive Analysis of Speech, J. Acoust. Soc. America, Vol.87, No.4, pp. 1738-1752, 1990.