speech analysis ,voice control,
persons disability,
MSAA technology

Piotr PORWIK[*]

ISOLATED WORD DESCRIPTORS AS CONTROL PARAMETERS OF THE COMPUTER APPLICATIONS

This paper is an extended version of the MIT’06 conference contribution. During the conference, many inquiries about the used techniques were performed. Hence, in the paper some parts of investigations were explained and discussed, with greater accuracy. It is shown that the computer applications can be controlled by a human voice. The computer controlling processes are available by means of utterance of isolated words, where application events with the aid of user’s voice can be serviced. The voice usage can be convenient for blind or partially sighted users or for persons with limb paresis. The Microsoft application events, by means of the practicable Microsoft Windows firmware MSAA® technology can be analysed. Such technology, together with isolated word descriptors, as voice recognition system, has been presented.

1. INTRODUCTION

An operating system is important part of each computer system. The operating system is a software program that manages the hardware and software resources of a computer. The hardware (processor, memory and I/O devices) is making-up main resources of a computer system. The software (compilers, database systems, etc.) characterises the ways of these resources usage, to solve tasks assigned by users.

In many cases, software producers take into consideration specific needs of some users group – for example with dysfunctions. To this end many operating systems have options activated, mainly by persons with disabilities: magnifier function or built-in text-to-speech module. In particular, if a given application is controlled by Microsoft Windows operating system, the so-called Microsoft Active Accessibility” (MSAA) functions can be applied. It is technique allowing to control of application by reading active fields, which are actually presented on the computer screen. Because application objects can be controlled by programmer – it is possible to connect it with hardware/software speech analysers. In such case, utterance of isolated words of the user can be recognised and used in steering of user’s application.

Hence, the application management by means of user’s voice can be applied. All considerations presented in the paper concern applications where Polish language is used and appropriate isolated words are recognised. The elaborated software was dedicated Polish users, hence in the all discussed examples, Polish version of operating system is

[*] Institute of Informatics, Silesian University, Będzińska 39, 41-200 Sosnowiec, Poland

performed; with all reporting and operating messages displayed in Polish language. It can be noticed that the elaborated application controlling method (described in the paper) can be adapted to any other national language (without major obstacle).

2. MICROSOFT ACTIVE ACCESSIBILITY (MSAA)

In various modern computer applications, where graphical user interfaces (GUIs) are used, object-oriented programming is very often recommended. GUI is now firmly established, as the preferred user-interface for end users. In such applications, graphical user's window can be included: buttons, radio-buttons, check boxes, combo–boxes, icons, pull-down menus, text fields, etc.

In order to convey meaningful information from user interface, we must be able to access that information from the application. Solution of this problem is the Microsoft Active Accessibility (MSAA) technology that has been available as an add–on since Windows 98.

The MSAA technology provides users with consistent mechanisms of exchanging information between applications and Windows–consistent technologies. For example, the MSAA allows applications to present type, name, location and current state of all objects being displayed on the screen. The MSAA notifies about all Windows–connected events, which are resulting in changes of the user interfaces. Although, it is not only way to communicate, the MSAA allows programmers supporting a broader variety of applications without custom programming for each one.

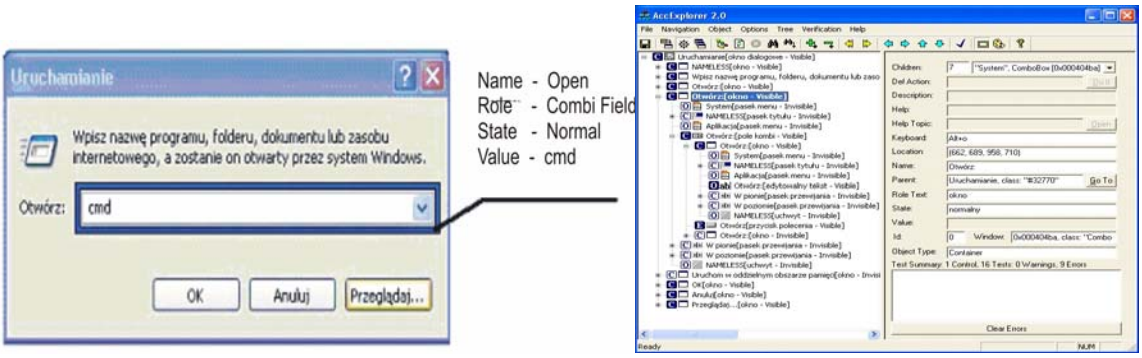
The number of applications, which support the MSAA is growing, although there are still many common applications, not supporting it. The important feature of the MSAA tool is possibility to use and to control these characteristic features in specific applications. A set of accessibility features, of application objects, provides persons with some disabilities; with facilities for using computers. The MSAA is a set of interfaces and APIs functions, which offer reliable mechanism~~s~~ for displaying and collecting information about Microsoft Windows-based user interfaces (UI).

Using this information, programmers and users can represent the UI in alternative format, such as speech, Braille or voice command. Hence, control applications can remotely manipulate the user's interface. The elaborated MSAA technology can be used only *at present) in Windows-based environment.

The main idea of the MSAA interface is based on special functions designed for UI elements. If UI are accessible for the MSAA – they are redirected to *IAccessible* interface and *child* ID. It is enough to describe UI object.

The simple window (it is part of the global application) and the MSAA accessible features – as description of the window elements have been presented in Fig.1a.

Unfortunately, designing of the MSAA applications is difficult. Hence, appropriate tools are freely available as auxiliary applications. One from such applications is AccExplorer program. The AccExplorer allows us to indicate all features that are accessible for programmer. For any object that can be modified in user's application, appropriate MSAA properties are shown as tree of inherited features (Fig. 1b). If application object is indicated by means of the cursor, the AccExplorer window will be visible automatically on the screen and associated with appropriate the MSAA feature. Exhaustive considerations about the MSAA technology can be found in bibliography position [5].



Wpisz nazwę programu, folderu, dokumentu lub zasobu internetowego, a zostanie on otwarty przez system Windows.

Otwórz: cmd

Name - Open
Role - Combi Field
State - Normal
Value - cmd

Children: 7 "System", ComboBox [0x000404ba]
Def Action:
Description:
Help:
Help Topic:
Keyboard: Alt+o
Location: {662, 689, 958, 710}
Name: Otwórz
Parent: Uruchamianie, class: "#32770"
Role Text: okno
State: normalny
Value:
Id: 0 Window: 0x000404ba, class: "Combo
Object Type: Container
Test Summary: 1 Control, 16 Tests: 0 Warnings, 9 Errors

Fig.1 An example of the application window, where the MSAA accessible features can be analyzed and recognized by *IAccessibe* interface (a) and the MSAA assistant tool: the AccExplorer program (b)

3. HUMAN VOICE AS COMPUTER CONTROL TOOL

There are two areas of application for speech recognition systems:

Dictation – where translation of the spoken word into written text is carried out, and computer control – where computer and software applications are controlled by speaking commands.

Speech recognition is one of the desired supporting technologies used after the voice recognition, as a natural method of commands generating. For this reason computer control by means of human voice is very attractive for many users. Speech–recognition programs do not understand what words mean, but isolated words can be recognised and used as context tool. This information helps the computer with choosing the most likely word from the database. On the other hand, speech–recognition software works to the best of its ability when the computer has a chance to adjust to each new speaker. The process of teaching the computer to recognise voice is called training. Extraction of words from speaking commands is a very difficult task; therefore in proposed approach different methods of speech recognition have been implemented. The procedures for voice features extraction are as follows:

Linear Prediction Coefficients (LPC):

- Autocorrelation Coefficients (AC),
- Reflection Coefficients (REF),
- Linear Prediction-based Cepstral Coefficients (LPCC).

Mel-Frequency Cepstral Coefficients (MFCC).

Hidden Markov Models (HMM).

The diagram of recognition of isolated words has been presented in Fig. 2, where connections between blocks are shown. In the first stage, appropriate isolated words are expressed, registered and theirs parameters are computed. Each recognised word is stored in database.

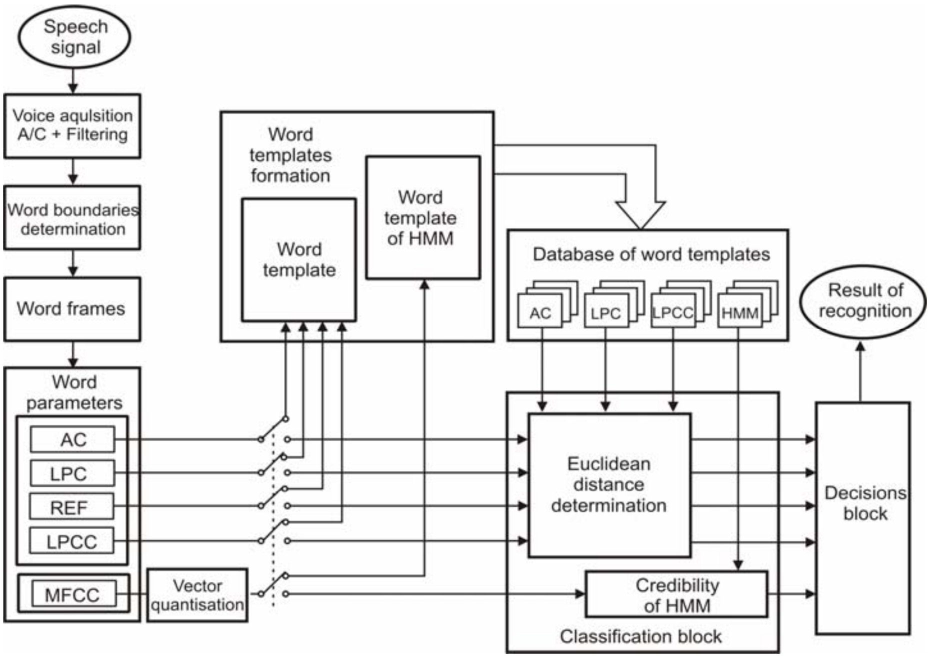


Fig. 2. Block scheme of the proposed system of isolated words recognition

The next section presents the experimental investigations. The effectiveness of speaker recognition by means of the above mentioned types of the features has been carried out.

4. ISOLATED WORD EXTRACTION FROM THE SPEECH SEQUENCE

The classical approach of speaker recognition can be performed through the use of short-term spectral templates. Such an approach involves applying an appropriate analysis of a spoken utterance to generate a sequence of short-term spectral feature vectors. Such templates have been found to contain significant voice characteristics, and may therefore be used an effective discrimination from amongst speakers. Amongst various types of speech features, LPCC and MFCC have been used; for better speaker recognition [1, 2, 3, 4]. Voice acquisition by PC-embedded sound card and microphone has been registered. Acquisition parameters can be individually selected, depending on sound cart quality. In presented experiment 8 kHz frequency sampling has been established, each voice sample in 8-bit resolution mode was registered. The voice recording time is 3*s*. Each record was three times registered; hence $3 \times 8000 = 24000$ samples per word have been stored in database.

4.1. PRELIMINARY VOICE FILTERING (PRE-EMPHASIS).

Registered by microphone speech signal is filtered. In this process, non-recursive Finite Impulse Response (FIR) filter has been used. Operation of the FIR filter can be described by equation:

$$\overline{y}(n) = y(n) - a \cdot y(n-1) \qquad (1)$$

where:

n – number of the current sample,
$\overline{y}(n)$ – sample of the signal after FIR filtration,
$y(n)$ – sample of the signal before FIR filtration,
a – filtration coefficient, $a = 0{,}937$.

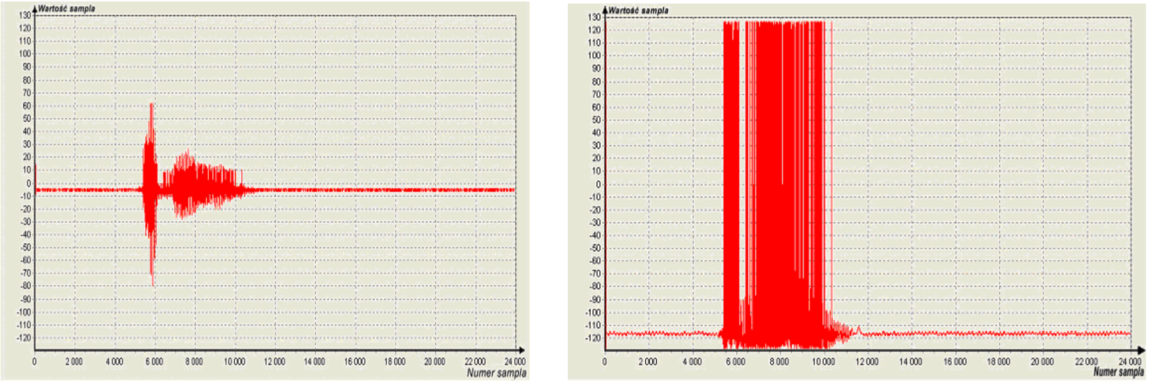


Fig. 3 The Polish word "Anuluj" (a) and this word after FIR filtering (b)

From Fig. 3 it can be observed that after FIR-type filtering, signal dynamics is reduced, high samples values are attenuated and low samples values, which occur in speech signal, are peaked.

4.2. FRAME ENERGY AND NORMALISATION

In proposed solution speech signal is divided into frames and each frame includes $N = 300$ samples.

Number of samples is equal near of 37ms of speech recording. For each frame, energy of the signal is computed:

$$E(l) = \log\left(\sum_{n=1}^{N} \overline{y}_l^2(n)\right), \quad 1 \le l \le K \tag{2}$$

and in the next stage the energy $E(l)$ is normalised:

$$E_{Norm}(l) = \frac{E(l)}{\max\{E(1),\dots,E(K)\}} \tag{3}$$

where:

l – number of current frame,
K – number of all frames,
$E(l)$ – energy of l^{th} frame,
$E_{Norm}(l)$ – normalised energy of l^{th} frame.

Fig. 4a depicts an energy distribution of the word "Anuluj" and a normalised distribution of this same word (Fig. 4b). Additionally, in Fig.4 two horizontal lines are shown because after normalisation, the two (upper and lower) energy levels are established. The upper level has value 0,6 and determine beginning of the word. The lower level has value 0,5 and determine end of the word. The energy levels were experimentally selected. On the basis of energy levels, boundaries of the word can be fixed. Range of the word allows determining silent and noise places, which occur during utterance.

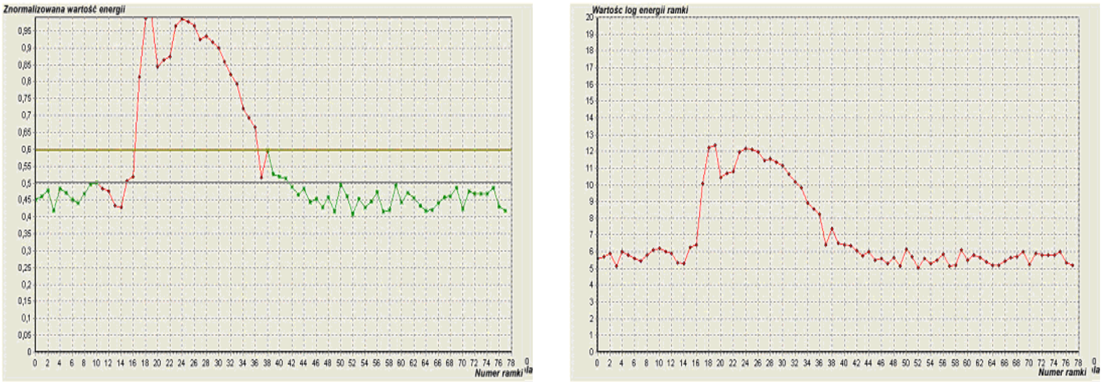


Fig. 4 Energy of the word Anuluj" in log scale (a) and distribution of the normalised energy of this word (b)

In this way, an isolated word (without a noise and silence) is determined (Fig. 5a). In this figure range of the word has been shown by means of the green colour. The red colour indicates noise or silence level. In the next stage, isolated word is again divided into frames (Fig. 5b). Many authors report, that the frame length should be of 20–40*ms* [2,3,5]. In proposed experiment time of the frame is 32*ms*. Because sampling frequency is equal to 8 kHz, hence in this time, N=256 samples is recorded. Such signal can be treated as quasi-stationary and such samples will be called as frames. In the next stage, for each frame the Hamming window is applied. Windowing operations are chosen to improve quality of a signal. When the FFT is used, one sample belongs to one window. In windowing process, samples at the boundaries are attenuated.

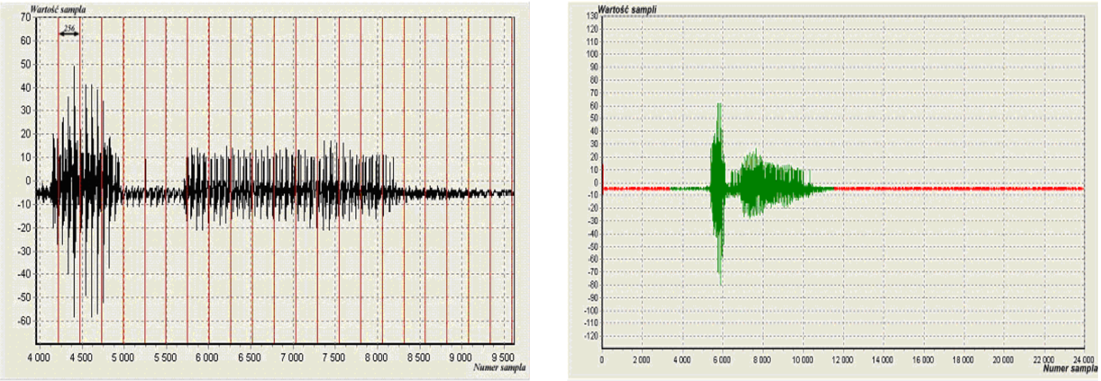


Fig. 5 Isolated word (green colour) (a) and the same re-scaled word, divided into N frames (b)

To reduce this effect, neighbour windows are overlapped. So, samples between two blocks are attenuated, but they belong to two blocks: their influence is still the same as samples that are not attenuated. Windowing for frame signal $\overline{y}(n)$ can be described as:

$$y^*(n) = \overline{y}(n) \cdot w(n),\; w(n) = 0{,}54 - 0{,}46 \cdot \cos\left(2\pi n/(N-1)\right), \qquad 0 \le n \le N \tag{4}$$

where: *w*(*n*) is the Hamming window.

For isolated words (Fig. 5a and 5b) their parameters have been determined.

5. USER VOICE PARAMETERS

In this paper different voice features are extracted on the basis of frame of the word (Fig.6b). The procedures for feature extraction are as follows [1,3]:

LPC – Linear prediction coefficients are obtained for each frame using the Levinson-Durbin recursive method [2,4]. The algorithm requires $O(n^2)$ operations, and is thus much more efficient for large n than standard Gaussian elimination, which requires $O(n^3)$ operations.

These coefficients are then converted to cepstral coefficients.

5.1. LPC COEFFICIENTS

As has been stated above, the speech signal is sampled and its frames were processed by the LPC algorithm. Every frame includes 256 samples. In other words, the LPC algorithm must extract the required parameters, characterizing 256 samples of speech, each of 32*ms*. Samples are high pass filtered and windowed using a Hamming window. The obtained samples are used as the input data of the autocorrelation detection blocks. The Levinson–Durbin algorithm is then used to solve a set of p linear equations. Reflection coefficients (REF) are also determined by the Levinson–Durbin algorithm.

The main stages of the Levinson–Durbin algorithm can be presented as follows:
Autocorrelation coefficients:

$$r(k) = \sum_{n=0}^{N-1-k} y(n)y(n+k),\ where\ k = 0,1,2,3,\ldots,p \tag{5}$$

Prediction coefficients:

$$\text{Initial values: } (m=1)\ E_0 = r(0),\ a_{11} = k_1 = r(1)/E_0,\ E_1 = E_0(1-k_1^2) \tag{6}$$

For $m \ge 2$:

a) $q_m = r(m) - \sum_{i=1}^{m-1} a_{i(m-1)} r(m-i)$,

b) $k_m = \dfrac{q_m}{E_{(m-1)}}$,

c) $a_{mm} = k_m$,

d) $a_{im} = a_{i(m-1)} - k_m a_{(m-i)(m-1)}$, for $i = 1, \ldots, m-1$

e) $E_m = E_{m-1}[1 - k_m^2]$

f) If $m < p$ then $m = m + 1$ and go to a). If $m = p$ then stop.

where:

$r(k)$, $0 \le k \le p$	– autocorrelation coefficients,
$a_i = a_{ip}$, $1 \le i \le p$	– prediction coefficients,
k_m, $1 \le m \le p$	– reflection coefficients,

5.2. MEL–SCALE CEPSTRAL COEFFICIENTS

As it has been said above, the speech signal is sampled and processed by the Levinson–Durbin algorithm. In this process a set of *p* linear equations is solved. The linear equations are functions of the sequence autocorrelation and the solution is the set of the coefficients *a*(*i*). From values *a*(*i*), cepstral coefficients can be calculated:

$$c(1) = a(1),\ c(k) = a(k) + \sum_{m=1}^{k-1} \frac{m}{k} \cdot c(m) \cdot a(k-m),\ 2 \le k \le p$$

(7)

$$c(k) = \sum_{m=k-p}^{k-1} \frac{m}{k} \cdot c(m) \cdot a(k-m),\ k > p$$

In practice, the MFCC parameters were computed. First signal is divided into short time windows, where the discrete Fourier transform (DFT) is computed:

$$s(k) = \sum_{n=0}^{N-1} w(n) y(n) \exp(-j2\pi kn / N) \quad k = 0,...,N-1 \tag{8}$$

and k corresponds to the frequency $f(k) = kf_s / N$, f_s is the sampling frequency in Hertz and $w(n)$ is a time–window. In experiments the Hamming window was chosen. The magnitude spectrum $|s(k)|$ is now scaled in both frequency and magnitude. First, the frequency is scaled logarithmically using the so–called Mel filter bank $H(k,m)$ and then the logarithm is computed, giving:

$$\overline{s}(n) = \ln\left(\sum_{k=0}^{N-1} |s(k)| \cdot H(k,m) \right) \quad n = 1,...,M \tag{9}$$

where M is the number of the filter banks and $M \square N$.

The Mel filter bank is a collection of triangular filters defined by the centre frequencies $f_c(m)$:

$$H(k,m)=\begin{cases} 0 & \text{for} \quad f(k)<f_c(m-1) \\ \dfrac{f(k)-f_c(m-1)}{f_c(m)-f_c(m-1)} & \text{for} \quad f_c(m-1)\le f(k)<f_c(m) \\ \dfrac{f(k)-f_c(m-1)}{f_c(m)-f_c(m+1)} & \text{for} \quad f_c(m)\le f(k)<f_c(m+1) \\ 0 & \text{for} \quad f(k)\ge f_c(m+1) \end{cases} \tag{10}$$

The centre frequencies of the filter bank are computed by approximating the Mel scale:

$$\Phi = 2595\cdot\log_{10}(1+f/700) \tag{11}$$

The Mel scale is computed, corresponding to algorithmic scaling of the repetition frequency:

$$\Delta\Phi = (\Phi_{\max}-\Phi_{\min})/(M+1) \tag{12}$$

where $\Phi_{\max}$ is the highest frequency of the filter bank on the Mel scale, computed from frequency $f_{\max}$ using equation (11), $\Phi_{\min}$ is the lowest frequency in Mel scale corresponding to frequency $f_{\min}$. The centre frequencies on the Mel scale are given by:

$$\Phi_{\mathrm{c}}(m) = m\cdot\Delta\Phi \tag{13}$$

To obtain the centre frequencies in Hertz, we use the inverse of equation (11):

$$f_c(m) = 700\cdot(10^{\Phi_c(m)/2595}-1) \tag{14}$$

Finally, the MFCCs are obtained by computing the DCT of $\overline{s}(n)$:

$$c_{mel}(k)=\sum_{n=1}^{M}\ln\left(\overline{s}(n)\right)\cdot\cos\left(\frac{\pi k}{L}(n-0{,}5)\right),\ k=1,2,\ldots,M \tag{15}$$

where: $c_{mel}(k)$ is the k^{th} MFCC.

Mentioned parameters (coefficients) constitute the vector of cepstral coefficients for the isolated word.

5.3. HIDDEN MARKOV MODELS

The speech recognition can be treated as stochastic process and can be described with the aid of the Hidden Markov Modelling technique (HMM). Speech is a continuous stream of acoustic information. Even if we assume that the talker must stop sometimes, the possible utterances vary in length and their number is practically unlimited. A possible solution is to trace the problem by the HMM technique. In this technique, for a given output sequence, the most likely set of state transition and output probabilities is searched. If isolated word is determined from speech sequence, then on the basis of mentioned algorithms HMM word models is stored in database and the highest probability of appearance of such word is calculated.

Appropriately ordered Mel coefficients form the data vector. In the next stage the vectors are quantized. Vector quantization is a non–parametric data reduction technique used to form a series of vectors known as code-words from a training set. This technique is well known in the research community [2]. In proposed approach codebook contains 64 code-words.

The HMM is characterized as follows:

- *N* the number of states in the model. $S = \{s_1,...,s_N\}$ and the state at time *t* will be denoted as q_t.
- *M* the number of distinct observation symbols per state. $V = \{v_1,...,v_M\}$.
- The state transition probability distribution $A = \{a_{ij}\}$,
- where $a_{ij} = P[q_{t+1} = s_j \mid q_t = s_i]$, $1 \le i, j \le N$.
- The observation symbol probability distribution in the state j, $B = \{b_j(k)\}$,
- where $b_j(k) = P[v_k \text{ at } t \mid q_t = s_j]$, $1 \le j \le N$, $1 \le k \le M$.
- The initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P[q_1 = s_i]$, $1 \le i \le N$

Appropriate values of N, M, A, B, π can be used as a generator to give an observation sequence $O = O_1, O_2, ..., O_T$, where each observation O_t is one of the symbols from the set *V*, and *T* is the number of observations in the sequence. For convenience, compact notation of the HMM is denoted by $\lambda = (A, B, \pi)$ to indicate the complete parameters of the model [2]. Let us consider using the HMMs to build an isolated word recognizer. Let us assume that we have a vocabulary of *V* words to be recognized and that each word is to be modeled by HMM. For each word in vocabulary we have a training set of *K* occurrences of each spoken word (spoken by 1 talker). Each occurrence of the word constitutes an observation sequence $O = O_1, O_2, ..., O_T$. For each word *v* in the vocabulary, appropriate HMM λ^v is build. In other word, the model parameters (A, B, π) should be estimated that optimize the likelihood of the training set observation vectors for the v^{th} word. Hence, calculation of model likelihoods $P(O \mid \lambda^v)$, for all possible models is determined. In the next stage, selection of the word, whose model likelihood is highest, is performed $v^{\#} = \arg\max_{1 \le v \le V} \{P(O \mid \lambda^v)\}$.

$P(O \mid \lambda^v)$ can be determined by use of the forward-backward algorithms and the Baum–Welch (B–W) algorithm [2,3,4].

6. RECOGNITION EXPERIMENT

In the experiment it was investigated whether text utterance has actually been produced by the speaker (associated with the best–matched models from database) or by unknown speaker outside the registered set of the features.

Table 1.Experimental set up

Method	Parameter
Nature of speech data	Isolated words
Features under consideration	All LPC parameters, MFCC, HMM
Speaker modelling	Spiker_Demo-2.6 – polish male voice
Number of registered speakers	1
Number of unknown speakers	1
Number of known utterances	500
Training data duration	3 times specific word per 3 seconds
Performance measure	Identification error

Quality of identification by means of the next equation was checked:

$$IDE(ntification) = \{1 - [(correct\ identifications) / (total\ tests)]\} \times 100\% \qquad (16).$$

Table 2. Speaker's identification

Type of speaker	IDE (%)
Registered speaker	4
Unknown speaker	75

If word is correctly recognised, appropriate MSAA procedure is activated. On the basis of recognised words (speakers) voice control is performed, and MSAA procedures are activated. It follows from investigations that recognition level is satisfactory on condition that short training procedure will be carried out.

7. CONCLUSIONS

In this paper the MSAA Microsoft internal interface with multi-parameter descriptors of isolated words are proposed. Such technique can be used if applications are controlled by Microsoft Windows operating system. For isolated words its different descriptors are calculated and finally, global credibility of utterance is composed. For this reason words false accept rate achieves low values. This method can be used if short application tuning is performed: each new word should be uttered three times and stored in database. It is very simple procedure. Every user can extend the database because for any new words appropriate MSAA feature can be found. Hence, functionality of the proposed solution can be improved at any moment.

BIBLIOGRAPHY

[1] Deller J.R. et al. *Discrete-Time Processing of Speech Signals*, Macmillan Pub. Company, 2000.

[2] Huang X.D., Acero A., Hon H-W., *Spoken language processing*. New York, Prentice Hall, 2001.

[3] Rabiner L.R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proc. of the IEEE Conf., Vol. 77, No.2, 1989, pp. 257– 286.

[4] Rabiner L. R., Juang B.H., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[5] Zieliński T.P., *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*. WKŁ, 2005.

[6] *Microsoft Active Accessibility*. Version 2.0. On line Microsoft documentation.

[7] Porwik P., *User Voice Identification in Computer Applications*. Proc. of the XI Int. Conf. Medical Informatics & Technology, 2006, pp. 205–211.