

Romanian language and dialog systems

CATRINA CHIVU¹, CATALIN CHIVU², CATALIN GEORGESCU³

^{1,2}Department of Economic Engineering and Manufacturing Systems
"Transilvania" University of Brasov
ROMANIA

³Department of Mathematical Sciences
The University of South Dakota
UNITED STATES

¹catrina.chivu@gmail.com, ²catalin.chivu@gmail.com, ³Catalin.Georgescu@usd.edu

Abstract: - The automated dialog systems are based on natural language processing, which has as main purpose the development of language computing theory using knowledge specific to informatics field, as algorithms, data structures, knowledge representation models and automat reasoning models. For many specialists in the field, speech technology includes two main subfields: *vocal synthesis* and *speech recognition*. These two subfields are complementary.

The present paper presents such a system developed for Romanian language.

Key-Words: - speech recognition, vocal synthesis, programming, prosthetics, neural network, MBroli, SAPI

1 Introduction

The technologies of speech processing attend significant results in the last decades [8]. This development was facilitated by the progress in the field of acoustic and linguistic theory, mathematical modelling of producing and recognising the speech, signal processing, structural programming, VLSI technology and hardware architecture [5, 6].

The first subfield of speech technology is the *vocal synthesis*. There are many advantages and disadvantages in obtaining a synthetic voice from a text. The main application of this synthetic voice is the reading machine for blind people. The text is scanned and then converted to phonetic symbols and, in the final, results the synthetic voice. This result needs a synthesis program based on rules and so, the development of such a synthesis is the main objective in this field.

In the next section of the present paper is shown how can be done such a vocal synthesis, applied for Romanian language and based on two speech applications, SAPI 5 [3] and MBrola [4].

The second subfield of speech technology is *speech recognition*, and, as part of this subfield, the most important is continuous speech recognition.

Speech recognition is strong related to audible modelling, paralinguistic vocal processing (speaker checking, recognizing him, speaker identification, language recognition, speaker sex recognition and isolating the subjects), verifying the speaking, temporary marking, etc.

There are many fields where speech recognition may be useful [5]. The most important fields are: domestic, education, portable technologies, vehicle

control, aviation, telephony, telecommunications, hostile environment, prosthesis, etc.

Speech recognition is a multidisciplinary field that interact with acoustics, phonetics, linguistics, psychology, mathematics and statistics, computer technology, electronics and human anatomy.

The third section of the present paper is dedicated to continuous speech recognition with two applications for Romanian language.

2 Vocal synthesis

This section is dedicated to a study case that wants to determine the best vocal synthesis system, already designed and applied to American English, which can be used for Romanian language.

To do this were considered two such systems: SAPI 5 (Speech Application Programming Interface) [3], designed by Microsoft and MBroli, offered by MBrola [4].

To be able to obtain a synthetic voice, starting from a text, it is possible to follow two steps (figure 1): *sound concatenation* (the synthetic voice is obtained after the voice is recorded, then the phonemes are labelled and thus there are generated the basic phonemes for that language and finally, after concatenation between these phonemes and the phonetic transcription of the text, is obtained the synthetic voice) and by *vocal tract simulation* (requires a phonology study that determines the changes of the vocal tract parameters during speaking process; after these, the vocal tract being simulated as the text phonetic transcription says) [2].

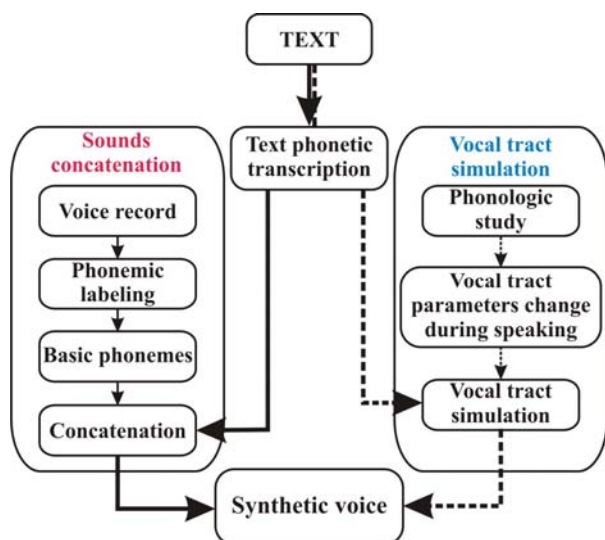


Fig. 1 The diagram of the synthetic voice generation

2.1 Romanian language with SAPI 5

One of the vocal synthesis systems, dedicated for English language but flexible enough to be adapted, was that offered by Microsoft as SAPI5 [3] toolbox. The system offers a satisfying quality of the synthetic voice. The disadvantage of this system is that it does not allow redefining the phonemic collection. The phonemic set is a minimal one and it is specific to American English. To reuse the vocal synthesis system, offered by SAPI, it has to be generated a rules set of SAPI phonemic collection adapted to that of the target language (in our case, Romanian language) [2].

Romanian language is a phonetic one, meaning that a word is read as it is written (with a few exceptions). Thus, it can be design, easily enough, a phonemic collection base (large enough, to describe the most sounds of Romanian language but, on the same time, small enough to maintain the simplicity of the system) [9].

Thus, after the authors defined the conventional base phonemes set, which it will be used to transcript the Romanian language, they designed a rules set, used to generate the Romanian language phonemes, using the American English language phonemes, given by SAPI. In the following are given some of these rules [2].

//punctuations

if we have ‘ ’ then the equivalent is ‘ ’;
 if we have ‘ ! ’ then the equivalent is ‘ ! ’;
 if we have ‘ . ’ then the equivalent is ‘ . ’;
 if we have ‘ ? ’ then the equivalent is ‘ ? ’;
 if we have ‘ , ’ then the equivalent is ‘ , ’;
 if we have ‘ - ’ then the equivalent is ‘ - ’;

//vocals

if we have ‘ a ’ or ‘ A ’ then the equivalent is ‘ aa 1 ’;

if we have ‘ e ’ or ‘ E ’ then the equivalent is ‘ eh ’;
 if we have ‘ i ’ or ‘ I ’ then the equivalent is ‘ iy ’;
 if we have ‘ o ’ or ‘ O ’ then the equivalent is ‘ ow ’;
 if we have ‘ u ’ or ‘ U ’ then the equivalent is ‘ uw 1 ’;
 if we have ‘ ă ’ or ‘ Ă ’ then the equivalent is ‘ ax 1 ’;
 if we have ‘ î ’ or ‘ Î ’ then the equivalent is ‘ ax ’;

//consonants

if we have ‘ b ’ or ‘ B ’ then the equivalent is ‘ b ’;
 if we have ‘ c ’ or ‘ C ’ then
 {if we have posterior 1 ‘ h ’ or ‘ H ’ then
 //chi if we have posterior 2 ‘ i ’ or ‘ I ’ then the
 equivalent is ‘ k ih ’;
 //che if we have posterior 2 ‘ e ’ or ‘ E ’ then the
 equivalent is ‘ k y eh ’};
 //ci if we have posterior 1 ‘ i ’ or ‘ I ’ then the
 equivalent is ‘ ch ’
 //ce if we have posterior 1 ‘ e ’ or ‘ E ’ then the
 equivalent is ‘ ch eh ’} then the equivalent is ‘ k ’;
 if we have ‘ d ’ or ‘ D ’ then the equivalent is ‘ d ’;
 if we have ‘ f ’ or ‘ F ’ then the equivalent is ‘ f ’;
 if we have ‘ g ’ or ‘ G ’ then
 { if we have posterior 1 ‘ h ’ or ‘ H ’ then
 //ghi if we have posterior 2 ‘ i ’ or ‘ I ’ then the
 equivalent is ‘ g ih ’;
 //ghe if we have posterior 2 ‘ e ’ or ‘ E ’ then the
 equivalent is ‘ g y eh ’};
 //gi if we have posterior 1 ‘ i ’ or ‘ I ’ then the
 equivalent is ‘ d jh ih ’
 //ge if we have posterior 1 ‘ e ’ or ‘ E ’ then the
 equivalent is ‘ d jh ae ’} then the equivalent is ‘ g ’;
 if we have ‘ h ’ or ‘ H ’ then the equivalent is ‘ h ’;
 if we have ‘ j ’ or ‘ J ’ then the equivalent is ‘ jh ’;
 if we have ‘ k ’ or ‘ K ’ then the equivalent is ‘ k ’;
 if we have ‘ l ’ or ‘ L ’ then the equivalent is ‘ l ’;
 if we have ‘ m ’ or ‘ M ’ then the equivalent is ‘ m ’;
 if we have ‘ n ’ or ‘ N ’ then the equivalent is ‘ n ’;
 if we have ‘ p ’ or ‘ P ’ then the equivalent is ‘ p ’;
 if we have ‘ r ’ or ‘ R ’ then the equivalent is ‘ r ’;
 if we have ‘ s ’ or ‘ S ’ then the equivalent is ‘ s ’;
 if we have ‘ ș ’ or ‘ Ș ’ then the equivalent is ‘ sh ’;
 if we have ‘ t ’ or ‘ T ’ then the equivalent is ‘ t ’;
 if we have ‘ ț ’ or ‘ Ț ’ then the equivalent is ‘ t z ’;
 if we have ‘ v ’ or ‘ V ’ then the equivalent is ‘ v ’;
 if we have ‘ w ’ or ‘ W ’ then the equivalent is ‘ v ’;
 if we have ‘ x ’ or ‘ X ’ then the equivalent is ‘ k s ’;
 if we have ‘ y ’ or ‘ Y ’ then the equivalent is ‘ y ’;
 if we have ‘ z ’ or ‘ Z ’ then the equivalent is ‘ z ’;

The above set of rules allows the direct conversion of a Romanian written text into a phonetic transcription. This phonetic transcription is then vocal synthesizes by SAPI based on its phonemic collection. The result of using this set of rules is an intelligible synthetic voice but, comparing to the synthetic voice that reads in

English, the quality is lower. The cause of lower quality is the fact that the system does not allow to redefine the base phonemic set, which corresponds to American English, and there can not be totally described all the Romanian phonemic sounds. Moreover, the vocal synthesis, contrary to first impressions, it is a very difficult problem, even in the case when the specific phonemic set exists. The neighborhood adaptation of the phonemic sounds and the length of the sounds, in different contexts, are also aspects specific to each language.

2.2 Romanian language with MBrola

A vocal synthesis system, which supports prosodic variations, is that offered by MBrola. This synthesizer was designed thus is adaptable and usable for a lot of languages but, not for any. MBrola offers a data base that contains the phonemic inventories specific to many languages. The Romanian language is contained by this data base and thus, all it has to be done is to develop an application that uses the vocal synthesis system offered by MBrola, using the Romanian language as native language of the system [2].

The MBrola [4] system (the vocal synthesis system offered by MBrola) has its own conventional phonemic set, specific to Romanian language.

The phonemic alphabet is the following:

MBrola set of rules (Table 1) [1]:

<i>SAMPA Symbol</i>	<i>Spelling</i>	<i>SAMPA Symbol</i>	<i>Spelling</i>
i	vî	d	dar
i 0	câi	k	cal
e	fel	g	gât
a	cap	ts	țară
@	măr	tS	ceas
o	loc	dZ	geantă
u	sur	f	fată
l	fân	v	vale
<i>semi-vocals:</i>			s
j	doi	z	zi
e X	deasă	S	șa
w	sau	Z	țoi
o X	culoare	h	haină
<i>consonants:</i>			m
p	păr	n	nas
b	barbă	l	lapte
t	țun	r	râs

Starting with this phonemic collection the authors determined the set of rules used to convert a text, written in Romanian, into a phonemic transcription, accessible to the system. The set of

rules designed, based on the phonetic pronunciation rules for Romanian [1], is:

//signs

if there is ‘ ’ then the equivalent is ‘ _ 1’;

if there is ‘!’ then the equivalent is ‘’;

if there is ‘.’ then the equivalent is ‘ _ 20’;

if there is ‘?’ then the equivalent is ‘;;F=1.91’;

if there is ‘,’ then the equivalent is ‘ _ 10’;

if there is ‘-’ then the equivalent is ‘’;

//vowels

if there is ‘a’ or ‘A’ then the equivalent is ‘a 45’;

if there is ‘e’ or ‘E’ then

{if there is before a consonant then the equivalent is ‘e 50’ else the equivalent is ‘e 40’};

if there is ‘i’ or ‘I’ then

{if there is posterior ‘ ‘ and anterior a consonant then the equivalent is ‘j 20’

else the equivalent is ‘i 35’};

if there is ‘o’ or ‘O’ then the equivalent is ‘o 30’;

if there is ‘u’ or ‘U’ then the equivalent is ‘u 40’;

if there is ‘ă’ or ‘Ă’ then the equivalent is ‘@ 30’;

if there is ‘î’ or ‘Î’ or ‘â’ or ‘Ă’ then the equivalent is ‘1 60’;

//consonants

if there is ‘b’ or ‘B’ then the equivalent is ‘b 30’;

if there is ‘c’ or ‘C’ then

{if there is posterior 1 ‘h’ or ‘H’ then

//chi if there is posterior 2 ‘i’ or ‘I’ then the equivalent is ‘k 60 + i 30’

//che if there is posterior 2 ‘e’ or ‘E’ then the equivalent is ‘k 60 + e 30’};

//ci if there is posterior 1 ‘i’ or ‘I’ then the equivalent is ‘t 30 + S 30 + i 30’

//ce if there is posterior 1 ‘e’ or ‘E’ then the equivalent is ‘t 30 + S 30 + e 30’}

else the equivalent is ‘k 60’;

if there is ‘d’ or ‘D’ then the equivalent is ‘d 60’;

if there is ‘f’ or ‘F’ then the equivalent is ‘f 60’;

if there is ‘g’ or ‘G’ then

{if there is posterior 1 ‘h’ or ‘H’ then

//ghi if there is posterior 2 ‘i’ or ‘I’ then the equivalent is ‘g 30 + i 30’;

//ghe if there is posterior 2 ‘e’ or ‘E’ then the equivalent is ‘g 30 + e 60’};

//gi if there is posterior 1 ‘i’ or ‘I’ then the equivalent is ‘G 30 + i 60’

//ge if there is posterior 1 ‘e’ or ‘E’ then the equivalent is ‘G 30 + e 60’}

else the equivalent is ‘g 30’;

if there is ‘h’ or ‘H’ then the equivalent is ‘h 30’;

if there is ‘j’ or ‘J’ then the equivalent is ‘C 60’;

if there is ‘k’ or ‘K’ then the equivalent is ‘k 30’;

if there is ‘l’ or ‘L’ then the equivalent is ‘l 60’;

if there is ‘m’ or ‘M’ then the equivalent is ‘m 60’;

if there is ‘n’ or ‘N’ then the equivalent is ‘n 60’;

if there is 'p' or 'P' then the equivalent is 'p 30';
 if there is 'r' or 'R' then the equivalent is 'r 30';
 if there is 's' or 'S' then the equivalent is 's 60';
 if there is 'ș' or 'Ș' then the equivalent is 'S 60';
 if there is 't' or 'T' then the equivalent is 't 60';
 if there is 'ț' or 'Ț' then the equivalent is 't30+s30';
 if there is 'v' or 'V' then the equivalent is 'v 60';
 if there is 'w' or 'W' then the equivalent is 'v 60';
 if there is 'x' or 'X' then the equivalent is 'k30+s30';
 if there is 'y' or 'Y' then the equivalent is 'i 30';
 if there is 'z' or 'Z' then the equivalent is 'z 60';

Using the above presented set of rules, in the context of MBrolı vocal synthesizer, the authors of present paper obtained a synthetic voice with remarkable high quality.

2.3 Analysis of vocal synthesis performances for Romanian language

The voice quality, obtained by using a vocal synthesis system, can be better evaluated by listening and analyzing its understandability of the pronunciation details.

To facilitate an audio analysis of the two synthetic voices given by the two synthesizers (Microsoft SAPI 5TTS and MBrolı), the authors developed a concrete application that has as input a Romanian written text and as outputs the phonetic transcription of this text using SAPI and MBrolı rules. Moreover, the program is able to reproduce/pronounce, in both synthesizers, the written text. The modular diagram of this application is presented in figure 2 [2].

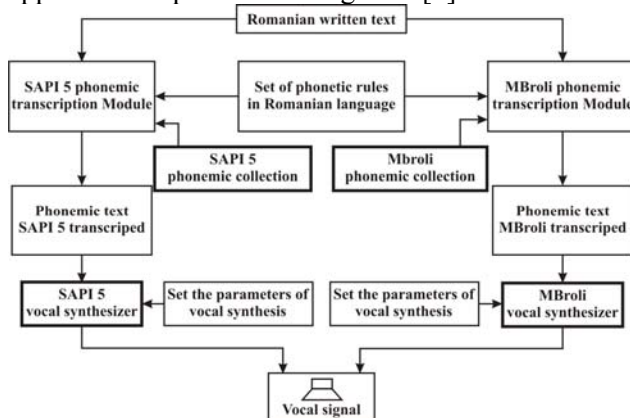


Fig. 2. Modular diagram of phonemic transcription application

The user may introduce a text written in Romanian (the letter *ă* is substituted by *ı*; letter *î*, by *j*; *â* by *j*; *ș* by *;* and letter *ț* by *“*). To the request of the user, one of the phonemic transcription modules will transform the introduced text, based on the

phonemic rules for Romanian language and the phonemic collection of the corresponding synthesizer. The interface of this application is presented in figure 3 [1].

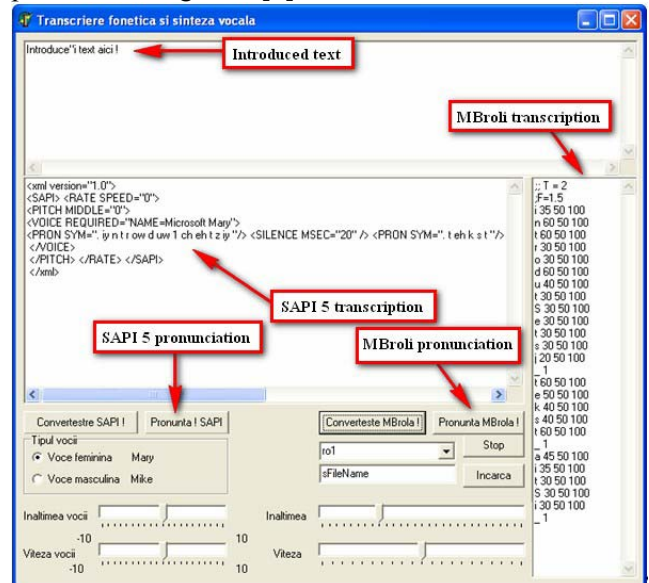


Fig. 3 Interface of phonetic transcription and vocal synthesis program

Another possibility to evaluate the voice quality is to compare the spectrogram of a synthetic voice to one of a real voice, both pronouncing the same text. The text should be chosen thus it is representative from the desired parameters point of view.

Thus, the authors decided to analyze the spectrograms of the vocal signals in three situations: natural voice (figure 4), MBrolı synthetic voice (figure 5) and SAPI 5 synthetic voice (figure 6), for the same word pronounced in Romanian “flexie”. The analysis was focused on some parameters: the evolution of fundamental frequency (voice pitch), prosody, secondary formants evolution, spectral form of phonemes.

Considering these parameters as outputs there were underlined some similarities and differences.

- The natural voice has more accentuated variations of the fundamental frequency (prosodic variations), while the MBrolı and SAPI voices have a stationary characteristic of fundamental frequency, so the prosodic variations are missing.
- The spectral form of the phonemes is quite the same for the three voices and so are the secondary formants.
- The transition from one phoneme to other: for the natural voice and the MBrolı voice this transition is a slowly one (there is an adaptation), as for SAPI voice the transition is a suddenly one.

- The quality of the inter-phonemic adaptation influences directly the naturalness of synthetic voice.

The good results obtained with MBroli synthesizer confirm the validity of the rules for the phonemic transcription. Even if the two synthesizers use the same transcription rules, the voice of SAPI synthesizer has a lower quality because the phonemic set is specific to American English language [7].

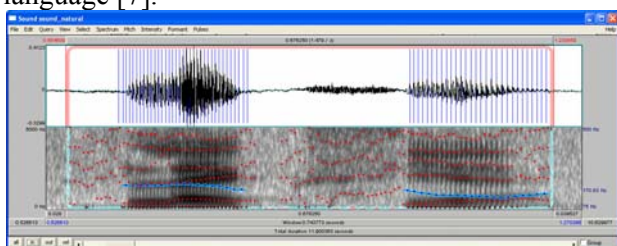


Fig. 4 Spectral analysis for word “flexie”
– natural voice –

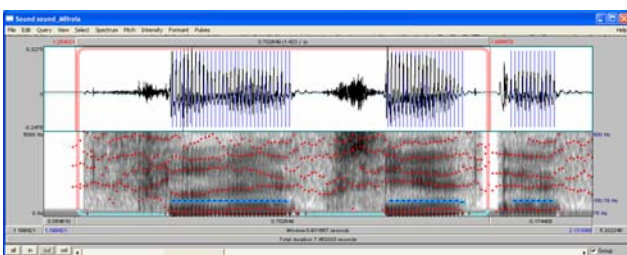


Fig. 5 Spectral analysis for word “flexie”
– MBroli synthetic voice –

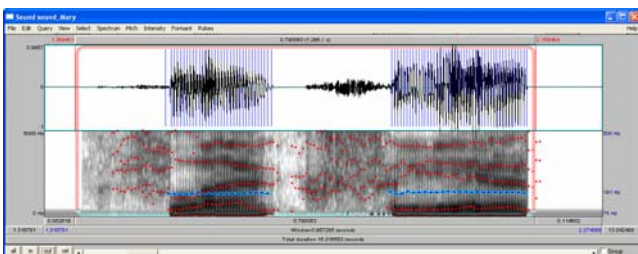


Fig. 6 Spectral analysis for word “flexie”
– SAPI 5 synthetic voice –

3 Continuous Speech Recognition

Speech recognition for Romanian language can be done using a speech recognition system specially designs for this task, which is based on neural network. Designing such a system implies to create a specific corpus that contains the training data set and also the testing data sets, to create (from zero) a recognition system and to train and to integrate this system in some specific applications (fig. 7).

The system designed by the authors is one for continuous speech recognition and that because the goal is to be a high applicable system.

The recognition engine, by itself, represents a program entity that is sold by the software companies, companies that invest fabulous sum of money, for many years, in specialists’ teams that should design, test and develop such systems. The product of such a company is a package of software in their executable form. Thus, the idea that one person (or one specialist) could develop by himself such a system is a utopia.

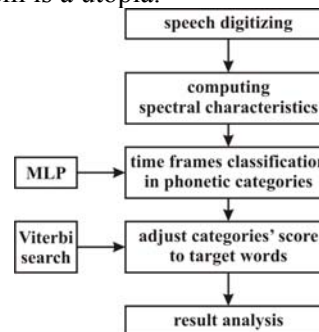


Fig. 7 Diagram of recognition process

The alternative solution is to consider an existing recognition system, which has satisfactory performances, even if it was already particularized for other language, and to adapt it and force it to recognize Romanian language. To adapt the system it should be replaced the grammar of the recognition system (dedicated to English applications) with other grammar, special designed for the desired application in Romanian language. To rewrite the grammar of the native language of the system for other language, different of the native one, implies, among redesigning the desired expression according to the rules assess by the system, the phonetic transcription of each Romanian word, which is part of the new grammar, based on the phonemic collection of the native language of the recognition system (in this situation the American English language).

Thus, considering the above presented SAPI 5 program the authors designed a grammar dedicated to Romanian language and for two particular applications: numerals and command of some practical applications.

3.1 Phonemic collection for Romanian language starting from SAPI 5 collection

Starting from SAPI 5 phonemic collection there were specified the rules of conversion for Romanian language. In the following are given these rules [1].

- The *ax* phoneme from English word *ago* is the same with that from Romanian word *hău*.
- The *b* phoneme from English word *big* is the same with that from Romanian word *stabil*.

- The *ch* phoneme from English word *chin* is the same with that from Romanian word *cină*.
- The *d* phoneme from English word *dig* is the same with that from Romanian word *dig*.
- The *eh* phoneme from English word *pet* is the same with that from Romanian word *pet*.
- The *er* phoneme from English word *fur* is almost the same with that from Romanian word *măr*.
- The *f* phoneme from English word *fork* is the same with that from Romanian word *foc*.
- The *g* phoneme from English word *gut* is the same with that from Romanian word *gaz*.
- The *h* phoneme from English word *help* is the same with that from Romanian word *han*.
- The *ih* phoneme from English word *fill* is the same with that from Romanian word *stiil*.
- The *jh* phoneme from English word *joy* is the same with that from Romanian word *imagine*.
- The *k* phoneme from English word *cut* is the same with that from Romanian word *cal*.
- The *l* phoneme from English word *lid* is the same with that from Romanian word *lin*.
- The *m* phoneme from English word *mat* is the same with that from Romanian word *mal*.
- The *n* phoneme from English word *no* is the same with that from Romanian word *nărav*.
- The *ow* phoneme from English word *go* is almost the same with that from Romanian word *nod*.
- The *p* phoneme from English word *put* is the same with that from Romanian word *puț*.
- The *r* phoneme from English word *red* is almost the same with that from Romanian word *rest*.
- The *s* phoneme from English word *sit* is the same with that from Romanian word *doșit*.
- The *sh* phoneme from English word *she* is the same with that from Romanian word *șiret*.
- The *t* phoneme from English word *talk* is the same with that from Romanian word *talc*.
- The *uh* phoneme from English word *book* is the same with that from Romanian word *suc*.
- The *v* phoneme from English word *vat* is the same with that from Romanian word *vară*.
- The *z* phoneme from English word *zap* is the same with that from Romanian word *zar*.
- The *zh* phoneme from English word *pleasure* is the same with that from Romanian word *jar*.

However, it can be observed that phoneme *î* or *â* (from Romanian word *cât*) is missing. That is why, this phoneme will be substitute by *ax* phoneme, presented above, which is the most closed, in pronunciation. The other Romanian phonemes that

have no correspondent could be described by using combinations of two or more English phonemes.

The *ch eh* phonemes are equivalent to the phoneme corresponding to *ce* group of letters from the word *cenușă*. The *ke eh* phonemes are equivalent to the phoneme corresponding to *che* group of letters from the word *cheie*. The *kih* phonemes are equivalent to the phoneme corresponding to *chi* group of letters from the word *șchiț*. The *d jh ae* phonemes are equivalent to the phoneme corresponding to *ge* group of letters from the word *gel*. The *gy eh* phonemes are equivalent to the phoneme corresponding to *ghe* group of letters from the word *ghem*. The *d jh ih* phonemes are equivalent to the phoneme corresponding to *gi* group of letters from the word *gin*. The *gih* phonemes are equivalent to the phoneme corresponding to *ghi* group of letters from the word *ghid*. The *tz* phonemes are equivalent to the phoneme corresponding to *ț* letter from the word *țap*.

3.2 Numerals speech recognition

The numeral is a very complex grammatical structure. The application designed allows recognition of numbers between zero and 999,999, pronounced according to the numerals pronunciation rules in Romanian or according to the current short forms. In figures 8 ÷ 10 are given the results of the recognition for some particular numbers [2].

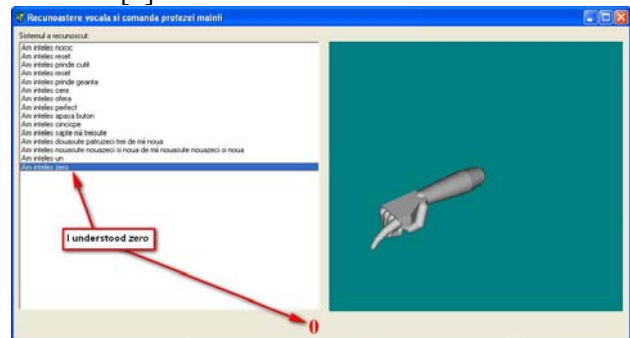


Fig. 8 Numeral 0 (zero)



Fig. 9 Numeral 999,999



Fig. 10 Numeral 243,009

In the following is presented a sequence of the grammar designed for numeral *zero* recognition. There are many rules and to recognize a complex numeral as in figure 10, for example, the program runs deep in the grammar.

```
<RULE NAME="1snumar" TOPLEVEL="ACTIVE">
<RULEREF NAME="snumar" />
<RULE NAME="snumar" TOPLEVEL="INACTIVE">
<RULEREF NAME="rzeci" PROPNAME="pzeci"
PROPID="PID_pzeci" />
<RULE NAME="rzeci">
<L PROPNAME="zeci" PROPID="PID_zeci">
<P VAL="0"/>/zero/zero/z ae r ow;/</P>
```

3.3 Vocal command of hand prosthesis

In the following is presented the structure of a complex system, which uses the speech recognition in Romanian language for control hand prosthesis [2]. The grammar of the speech recognition system was written thus it allows a variety of commands. The commands were classified on two levels of complexity [1].

The first level of complexity corresponds to cosmetic commands that are dedicated to the control of the functional parameters of the prosthesis, as gripping force, angular speed of joints (low level commands) [10]. Some of these commands imply the numeral, which also represents a complex grammatical section.

The second level of complexity corresponds to the simple direct commands (high level commands) that are dedicated to complex movements.

A very important goal is to create the high level commands as simpler as possible (for example: the command "*prinde creion*" – *catch the pencil* – that should determine the movement of the prosthesis in a predefined position, its rotation, the catch of the pencil and orientation of the pencil in the writing position). The simplicity is very important for the ergonomics of the prosthesis. As simpler the commands are, as efficient the speech recognition system becomes and the prosthesis is easier to be used. All the commands are in Romanian.

The grammar special created for the vocal command of prosthesis has a structural form, which gives a very high flexibility from composition of commands point of view. Thus, as example, it is presented a section of this grammar.

```
<RULE NAME="comenzi_rapide" TOPLEVEL="ACTIVE">
<L PROPNAME="comenzi_rapide" PROPID="PID_comenzi_rapide">
<RULEREF NAME="rscurte" PROPNAME="pscurte"
PROPID="PID_pscurte"/> </P> <P> <P>
<RULEREF NAME="rcompuse" PROPNAME="pcompuse"
PROPID="PID_pcompuse"/> </P> <O>
<RULEREF NAME="robiecte" PROPNAME="pobiecte"
PROPID="PID_pobiecte"/>
</O> </P> </L> </RULE>
```

This section defines a rule called "*comenzi rapide*" (*fast commands*) that has a list of phrases also called "*comenzi rapide*" (*fast commands*). If the rule has the parameter TOPLEVEL="ACTIVE" then the speech recognition system verifies each vocal expression if there is created based on the topology of this rule. After checking, the expression can be recognized as being or not being created based on the topology of this rule. The list contains two phrases. The first phrase is, in fact, a reference to other rule, called "*rscurte*" that describes the short commands that can be done by the prosthesis. The second phrase is composed from a compulsory one (in fact a reference to another rule called "*rcompuse*" that describes the compose commands) and an optional section (which is also a reference to the rule called "*robiecte*" that describes the list of objects of the compose commands).

A command to the system may have three solutions: may be recognized as a short command, may be recognized a compose command with or without target object or may be rejected by the system as not being part of the list of accepted command.

In figures 11 ÷ 14 are presented five of the implemented *short commands* and the position of the hand as response to the command. It can be given two commands consequently or, a *reset* command between two.

For the bellow command the grammar lines for the *short commands* are:

```
<RULE NAME="rscurte">
<L PROPNAME="scurte" PROPID="PID_scurte">
<P VAL="10"/>/noroc/noroc/n ow r ow k;/</P>
<P VAL="12"/>/ofera/ofera/ow f ae r ax 1;/</P>
<P VAL="14"/>/apasa buton/apasa buton/aa 1 p aa 1 s ax
1 b uw 1 t ow n;/</P>
<P VAL="13"/>/perfect/perfect/p ae r f ae k t;/</P>
```

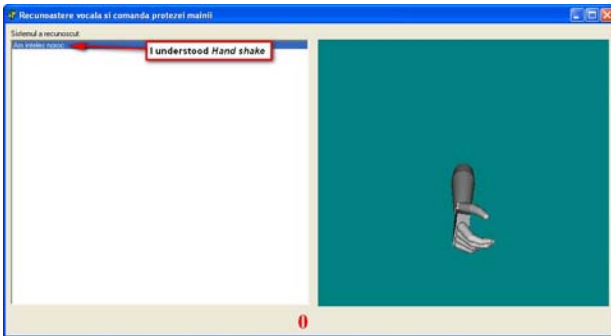


Fig. 11. Command *Noroc (Hand shake)*

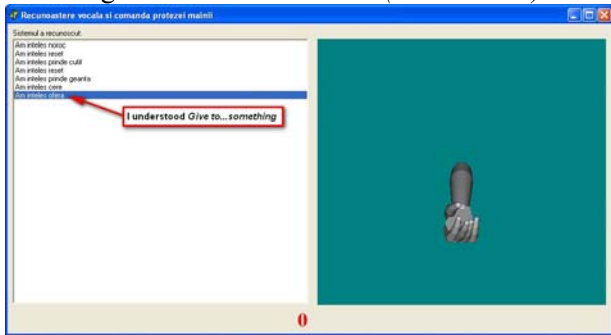


Fig. 12. Command *Ofera (Offer)*

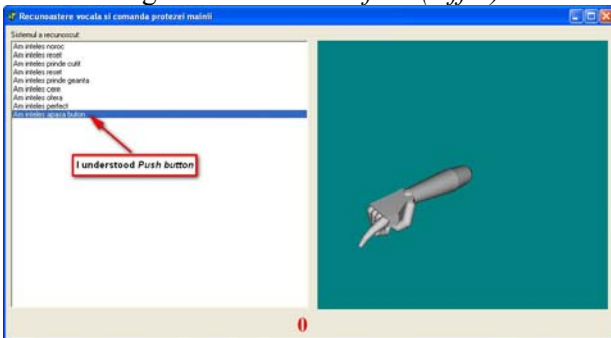


Fig. 13. Command *Apasa buton (Push button)*

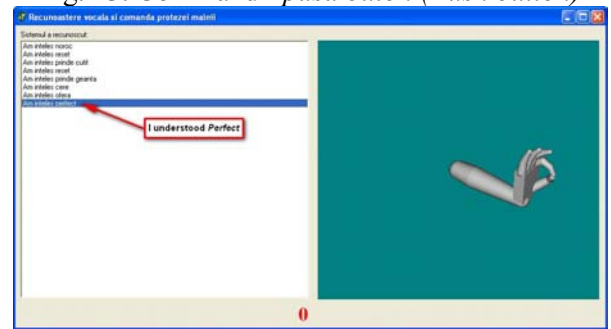


Fig. 14. Command *Perfect*

Rule called “*rscurte*” has a list of phrases (the properly commands). One of these phrases may correspond to a part of vocal expression. Each of these phrases has a numeric parameter, called VAL, which is returned to application by the recognition engine when the expression corresponds to a part of the analyzed vocal expression. The syntax of the rules includes the name of the commands, the returned text value and the desired pronunciation (phonetic transcription of the vocal segment that may correspond to the respective phrase).

3.4. Vocal command applied to automatic controlled slowly processes

The automatic controllers are based on a specific law and minimize the error of the system. Normally, an automatic controller has included the comparator, which compares the output with the input.

In the following it is considered a simple process controlled by an automatic controller that allows the operator to switch between the manual - automat operation modes. The process includes a D.C. motor that closes and opens a valve. The controlled parameter can be, depending on the situation, the pressure difference between input and output or the flow rate through the valve or the valve’s position.

The vocal command module of such a system is shown in figure 15.

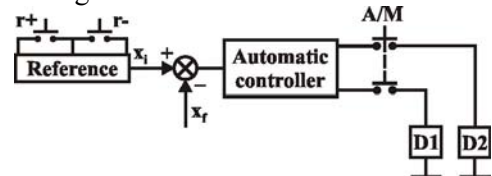


Fig. 15. Diagram of vocal command module

The elements presented in the above figure represent:

Referință (Reference) – device that gives the reference value for the system;

r+ – button that increases the reference value;

r- – button that decreases the reference value;

xi – reference value;

xf – value of controlled parameter.

Regulator automat (automatic controller)

The scheme from figure 15 allows to control the system both manually and automatic. The human operator is able to use, at any moment, the manually control to command the process. If the system is working in automatic operation mode and the human operator uses manual commands then, temporarily, the process switch on the manual command. In this situation, when the manual command stops, the system switches back on automatic operation mode. The automatic controller may be manually activated or inactivated by activating or inactivating the command on the D₁ and D₂ relays. If the automatic controller was inactivated the process control is a manual one obtained by using the *Scade (Decrease)* and *Crește (Increase)* buttons (figure 17).

Switching between manually and automatic command is done without any problem because the motor position (and valve position), which is steady during no supply time range, memorizes the output variable until is switched off the current command.

The reference can be a constant or manually modified by the human operator.

The purpose is to apply vocal command to such a system by maintaining active the switching possibility between automatic and manually control. This condition implies to include the Speech Recognition System (SRS) in the command and control system. Depending on the type of command that it receives (manual or vocal), SRS is able to command the switch on/off between manually and automatic control, to rotate the motor in one direction or the other with a specific angle, if the command is manually, or, if the command is an automatic one, to change the reference of the controller. The reference can be changed both manually and vocal. The vocal command module of such a system is shown in figure 16.

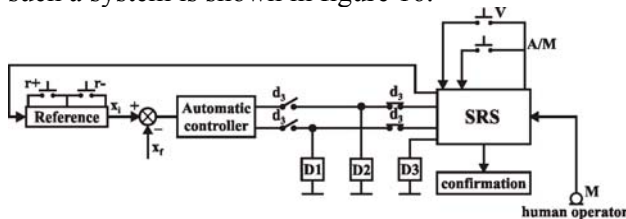


Fig. 16. Vocal command module of a discrete automatic control system

Using scheme presented in figure 16 there can be transmitted commands to the motor.

The notations from the above figure represent:

D_3 – relay winding (the relay command the switching on between Automatic and Manually operation mode),

d_3 – contacts of D_3 relay,

V – button that allows to activate or inactivate the vocal command,

A/M – button that switches on/off the Automatic – Manually operation mode.

Speech recognition system, used as part of the automatic control system, improves the ergonomics of the system. Also, it is necessary to create a grammar dedicated to this application. This grammar has to include all the possible forms for the commands transmitted by the human operator to the system. The vocal commands have to completely substitute the manual commands and, eventually to facilitate to system control.

In the case of manually command, the human operator has to be able to do the following manually operations:

- close the valve (proportionally to the time interval during which the close button is hold on);
- open the valve (proportionally to the time interval during which the open button is hold on);
- activate the vocal command (without inactivating the manual one);
- partially inactivate the vocal command (without inactivating the manual one);

- modify the reference;
- switch on the automatic command.

In the case of automatic command the human operator has to be able to do the following manually operations:

- activate the vocal command;
- partially inactivate the vocal command;
- choose the reference;
- switch on the manually command.

For the manually command, the human operator has to be able to do the following vocal commands:

- close the valve (with predefined steps or percentage of the valve's full stroke);
- open the valve (with predefined steps or percentage of the valve's full stroke);
- switch on the vocal command (without inactivating the manual one), even if the vocal command is partially inactivated;
- partially inactivate the vocal command (without inactivating the manual one);
- change the reference (by increasing or decreasing with a predefined step or percentage);
- switch on the automatic command;
- activate the vocal confirmation of a correct recognition done by the SRS;
- inactivate the vocal confirmation of a correct recognition done by the SRS.

For the automatic command, the human operator has to be able to do the following vocal commands:

- activate the vocal command (without inactivating the automatic command), even if the vocal command is partially inactivated;
- partially inactivate the vocal command;
- choose the reference (by increasing or decreasing with a predefined step or percentage of the definition interval);
- switch on the manually command;
- activate the vocal confirmation of a correct recognition done by the SRS;
- inactivate the vocal confirmation of a correct recognition done by the SRS.

The vocal commands have to be part of the following possible operations:

- to close the valve with a predefined step the human operator has to say "închide robinet" ("close valve");
- to close the valve with percentage of the valve's full stroke, the human operator has to say "închide robinetul cu 'x' procente" ("close valve with 'x' percentage") or "închide robinetul cu 'x' la sută" ("close valve with 'x' per cent");
- to open the valve with a predefined step the human operator has to say "deschide robinet" ("open valve");

- to open the valve with percentage of the valve's full stroke, the human operator has to say "deschide robinetul cu 'x' procente" ("open valve with 'x' percentage") or "deschide robinetul cu 'x' la sută" ("open valve with 'x' per cent");
- to activate the vocal command, the human operator has to say "activează comanda vocală" ("activate vocal command");
- to inactivate the vocal command, the human operator has to say "dezactivează comanda vocală" ("inactivate vocal command");
- to activate the confirmation, the human operator has to say "activează confirmarea" ("activate confirmation");
- to inactivate the confirmation, the human operator has to say "dezactivează confirmarea" ("inactivate confirmation");
- to change the reference, by increasing the value with a predefined step, the human operator has to say "crește referință" ("increase reference")/ "incrementează referință" ("increment reference");
- to change the reference, by decreasing the value with a predefined step, the human operator has to say "scade referință" ("decrease reference") or "decrementează referință" ("decrement reference");
- to change the reference to a specific value, the human operator has to say "mută referința la 'x' la sută" ("change reference to 'x' per cent") or "mută referința la 'x' procente" ("change reference to 'x' percentage");
- to switch on to automatic command, the human operator has to say "treci pe comandă automată" ("switch to automatic command");
- to switch on to manually command, the human operator has to say "treci pe comandă manuală" ("switch to manual command").

In figure 17 is presented the interface of the program that implements the speech recognition to the command and control of a system with automatic control and that allows to switch on between the Automatic and Manual operation mode.

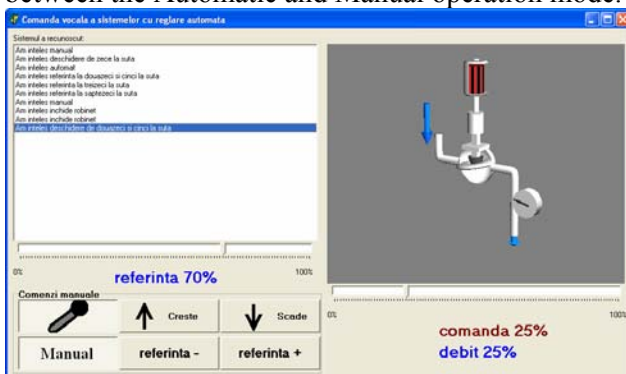


Fig. 17. Interface of the program that applies vocal command to an automatic control system

4 Conclusion

The originality of the research and the results regards:

- original solution in developing a new Speech Recognition System without starting it from zero;
- application in the automatic controlled slowly process that regards the vocal command for automatic opening and closing the valve;
- application for hand prosthesis: implementation of many vocal commands that allow the movement of the prosthesis accordingly to human hand movement.

By adapting the grammar and the phonemic rules of the speech recognition engine for American English, to Romanian language, was obtained a new speech recognition system. The validity of this adaptation was verified by the applications presented in the present paper.

References:

- [1] C. Chivu, *Contributions to vocal signal processing*, PhD Thesis, "Politehnica" University of Bucharest, Romania, 2006
- [2] C. Chivu, C. Chivu, *Automated dialog systems for Romanian language*, Computational methods and intelligent systems, Published by WSEAS Press, ISBN 978-960-6766-60-2, ISSN 1790-5117, pg. 83-91, 2008.
- [3] Microsoft Speech Application Programming Interface ("SAPI 5.1"), *Microsoft Speech SDK, version 5.1 Documentation*.
- [4] <http://tcts.fpms.ac.be/synthesis/mbrola.html> Faculté Polytechnique de Mons (Belgium).
- [5] A.M. Abdelatty, A. Spiegel, &co, An acoustic-phonetic feature-based system for the automatic recognition of fricative consonants, *In Proc. ICASSP-98*, pp. 961-964, 1998.
- [6] D. Tătar, *Artificial Intelligence: applications in natural language processing (In Romanian language)*, "Albastră" Publishing House, 2003.
- [7] L. Dybkjer, H. Hensen, W. Minker (editors), *Evaluation of Text and Speech Systems*, Springer Publishing House, 2007.
- [8] W. Chou, B.H. Juang (editors), *Pattern recognition in speech and language processing*, CRC Press Publishing House, 2003.
- [9] http://en.wikipedia.org/International_Phonetic_Alphabet
- [10] C. Chivu, C. Chivu, *Fuzzy control applied in prosthetics*, Advanced Topics on Fuzzy Systems, Published by WSEAS Press, ISBN 978-960-6766-57-2, ISSN 1790-5109, pg. 100-107, 2008.