# Impaired Speech Evaluation using Mel-Cepstrum Analysis

Ovidiu Grigore. Corina Grigore. Valentin Velican.

***Abstract*—** The study presents a simple solution for identifying impaired speech pronunciations using the Mel-Cepstrum Coefficients as features. The pronunciation defect studied was rhotacism – wrongly pronounced 'r' - in the case of Romanian language. By comparing the timbre variation over the duration of the analyzed speech an average of 78% correct classification rate has been obtained.

***Keywords*—**pronunciation assessment, speech processing, software design, mel-cepstrum

## I. INTRODUCTION

THE increasing number of children with speaking problems (e.g. pronunciation) at early age generates a great request of speech therapy services. This phenomenon creates agglomeration of therapeutic interschool centres, and increasing cost of private speech therapy.

The missing model for the correct pronunciation outside the speech therapy centre makes all home-developed exercises to be empirical and not-systematic. Also, the lack of motivation makes all home-developed exercises to be frequently inconsistent when the parents cannot supervise the child. In such cases, untreated or inefficiently treated problems may influence the development of the suffering person, creating premises for more complicated disorders.

The speech therapy practice is based on correct diagnosis of the pronunciation problems and on their correction according to a precise methodology. The speech therapist needs a large amount of time to precisely identify not only problematic sounds, but also the problematic phonemic combinations. The correct "emission" of affected sounds is made both isolated and in syllables and words.

Because of that, a computer application can lighten the therapy and can also simplify the specialist effort, offering an etalon-mean for both inexperienced therapists and home conducted exercises.

This article presents a simple software application capable of classifying the correct or incorrect pronunciation of /r/ phoneme and the solution used to identify the differences in pronunciation. A brief problem formulation is subsequently presented, outlining the need for such an application and defending the reasons that make this application useful. Then, the method of identifying the mispronunciations is detailed, beginning with a highlight of the most important steps that were followed in developing the application. The core section of the article describes the implementation, presenting the mathematical tools used (Mel-Cepstrum) and the feature extraction method we identified. At the end the method was tested on a small batch of subjects and the results are presented in section V of the article.

## II. PROBLEM FORMULATION

Current speech therapy assumes the accurate diagnosis of the pronunciation troubles, but also their adjustment in accordance with a strictly defined methodology. The speech therapist needs a considerable period of time to precisely identify not only the specific sound but also especially phoneme combinations presenting problems. The imposition of the affected sounds is made not only for the isolate sounds, but also in syllables and words. Therefore, the traditional, empirical approach to the problem, where the experience of the specialist plays a key role in completing a therapy with good outcomes, can be avoided using an automated, objective in decisions, method. For this reason having such a product at one's disposal, which assist the pronunciation deficiency therapy, would considerably simplify the specialist's input, providing an efficient method for home exercise.

From another point of view, speech processing generally supposes that the analysed speech sequence is correct from the point of view of pronunciation. Therefore all existing methods apply on correct speech samples and also use vowels as primary feature selection sources, as these contain most of the information condensed in speech. It is clear then that the most important issue to address is the implementation of a novel method that should not only correctly analyse impaired pronunciation but also should use little information extracted from consonants or vowel-consonants groupings.

Finally, automated speech therapy in our country, Romania, lacks the support it needs. There were some individual, empirical attempts, undocumented in literature, to use the oscilloscope in the speech therapy activity, by requesting the affected child to superpose his own wrong pronunciation resulted curve over the etalon, produced by the speech therapist. Also, the SUVAG LINGUA equipment was used in the 70's to give correct models of the sounds to the children with articulated deficiency; this activity was

Manuscript received on October 15, 2010
This work was supported by CNCSIS-UEFISCSU, project no.846/2009
This work was presented at the WSEAS **CIRCUITS, SYSTEMS, SIGNALS (CSS) conference, Malta September 15-17, 2010.** having ID number: **ID 201-474**
Ovidiu Grigore is with Polytechnic University of Bucharest, Department of Applied Electronics and Information Engineering (e-mail: ovidiu.grigore@upb.ro)
Corina Grigore is with Coltea Hospital, Bucharest
Valentin Velican is with Polytechnic University of Bucharest, Department of Applied Electronics and Information Engineering (corresponding author, phone: 0040745765137, e-mail: valentin.velican@upb.ro)

important especially in the education of phonemic hearing [7]. Nowadays there is no more technological support for this kind of therapy and the mentioned methods are obsolete and hard to implement.

## III. PROBLEM SOLUTION

When designing a speech processing system that intends to assist/replace real life practices, the most important design challenges come from the following development steps:

A. Understanding the pronunciation defects.
B. Creating a rich database of speech sample.
C. Choosing an appropriate feature extraction method.

Also it is important to take into consideration the available hardware that can support such an implementation. Due to the fact that nowadays personal computers are widespread and relatively low cost, it is important to develop an application capable of efficiently working on an ordinary PC with low end peripherals (microphones). Thus, high-end, costly (and probably hard to replace in case of malfunctions), DSP systems can be avoided in favor of (much) more popular and easy to use products.

### *A. Understanding the Pronunciation Defects*

To be able to develop an automatic system it is necessary to understand all the details of the defective speech problems, starting with analyzing the sounds situations that are wrong uttered and continuing even with the classical methodology that is used in this moment to correct the pronunciation.

Our implementation studies the defective pronunciation of /r/ in Romanian language. A strong rhotic consonant, this is one of the most common mispronounced sounds by persons with pronunciation problems; it's utterance ranging from children of young age to adults. The defect, known in general as "rhotacism" is in it's worst cases observed as a replacement of /r/ with other closely related sounds like /l/, /d/, /t/ (in other cases /r/ is not pronounced at all). [1] In the mild, linguistic acceptable mispronunciations, the phoneme is replaced with a guttural vibrating /r/, resembling the pronunciation of /r/ in French.

The shape of the correct /r/ phoneme resembles very much an amplitude modulated signal with an envelope frequency of 25-30Hz, as presented in (Fig.1). The shape of the envelope can be a very useful feature in the case of comparing correctly pronunciations with highly altered ones, the envelope of the latter being constant (or 'semi-constant') over time (Fig.2). This is due to the fact that severe defects tend to replace /r/ with a consonant like /l/ where a vowel-like component can be identified inside the phoneme. Therefore the sound has low varying amplitude unlike correct phoneme.

Yet in the case of mild pronunciation defects where /r/ is guttural (remember we define correctly according to Romanian language), the shape of the signal is similar to its correct counterpart. (Fig.3) In this case, extracting the shape of the envelope is no longer useful so spectrum based methods where frequency components are highlighted (or cepstrum analysis) are to be preferred in order to describe the timbre of the voice rather than the shape of the sound. Extracting the timbre is also a risky choice due to the fact that an adult's timbre is significantly different to the child's timbre. Therefore extracting features that describe timbre and comparing them are not sufficient.

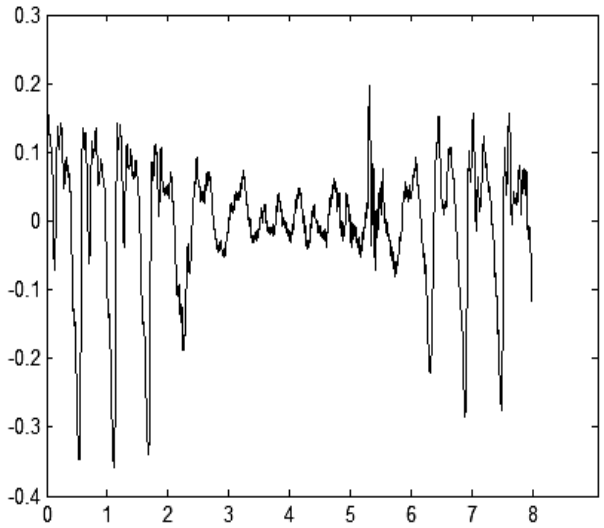


Fig. 1 Correctly pronounced 'r' in 'rac' (crab)

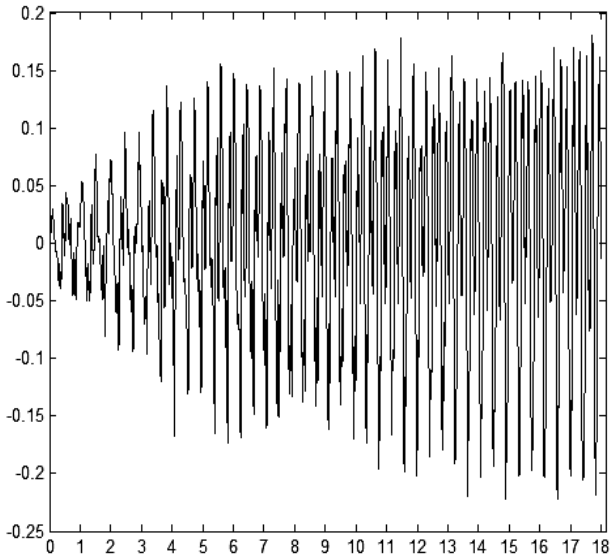


Fig. 2 Incorrectly pronounced 'r' in 'rac' (crab)

We therefore decided to study what happens to such features (those that describe the voice timbre) if we observed them over the duration of the pronounced phoneme.

### *B. Creating a Rich Database of Speech Samples*

To achieve a coherent analysis and extract the best possible features a diverse database should be build. In general, the speech material must contain utterances of isolated sounds

(phonemes) and of specific groups of sounds. These should be extracted from recordings of both correctly and incorrectly pronounced words containing the analyzed phoneme.

Due to the fact that subtle differences of pronunciation exist when /r/ is positioned differently inside a word or/and in relationship to a vowel [1] we decided to simplify our analysis and concentrate only on words that contain /r/ as the first letter.

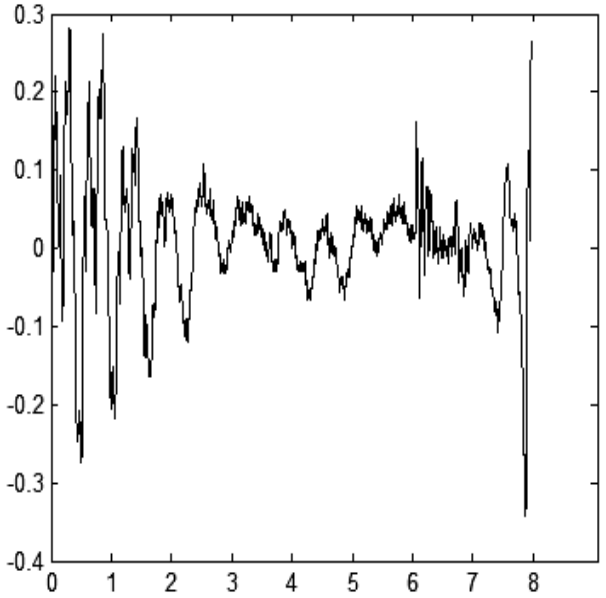


Fig. 3 Guttural pronounced 'r' in 'rac' (crab)

A database containing words like "rac" (crab), "ren" (reindeer), etc. was build by recording adults, 3 men and 8 women of whom 1 man and 3 women had different degrees of rhotacism. We recorded (at a sample rate of 44.1 KHz) ten words from each subject - each word pronounced at least three times - making sure we have all the possible vowels after the beginning /r/. Subjects were recorded using the same tools, in complete silence conditions, in order to benefit from the qualities of an as much as possible unaltered voice signal. The discussion in this direction can go on and is interesting to observe that in real life situations, where the speech therapy takes place, a suitable recording room is possibly unavailable.

Phonemes were afterwards extracted, taking care to isolate the /r/ as much as possible from the near vowel.

### C. Choosing the Appropriate Feature Extraction Method

Feature extraction is the fundamental part of any speech processing system. In this case correct and impaired speech must be differentiated by comparing data (features) extracted from the sound samples. Therefore the data extracted should describe the sound sample in an "as complete as possible" manner with detail on characteristics that differentiate correct of incorrect pronunciations and establish a resemblance between correct samples. For these reasons a feature extraction stage should receive great attention from the designer.

In our application we decided to use the *Mel-Cepstrum analysis.* This represents a tool mostly used today in speech / speaker recognition products. It is the corresponding implementation of the frequency cepstrum onto Mel frequency scale. The advantage of Mel Cepstrum over the "classical" implementation resides mostly in the difference between the Mel frequency scale and the Hertz frequency scale. This consists in the fact that Mel frequency scale approximates the human auditory system's response more accurately. Also Mel-Cepstrum has the advantage of representing speech amplitude in a compact form. [2].

In general, Cepstrum (both real and complex) is used in speech processing to describe the emission of voice accordingly with the source-filter model. Briefly, the algorithm is very simple: the Fast Fourier Transform of the voice signal is computed, then the logarithm of the resulting spectrum and finally the Inverse Fourier Transform. Due to the fact that the excitation of the vocal cords and the filtering of the vocal tract convolve in time domain and multiply in frequency domain, after computing the cepstrum these two separate in different regions: one associated with the filter (represented by the vocal tract) and one associated with the excitation source – the vocal chords. It is important to mention that the mark of the vocal chords is generally visible in vowels or any other strong sounds which contain a good amount of energy.

Mel-Cepstrum is an extension of the classical Cepstrum on the Mel frequency scale. It is extensively nowadays used in Automatic Speech Recognition [6] due to the ability of the Mel (*Mel*ody) scale to better and more efficiently represent the analyzed sound.

## IV. IMPLEMENTATION

The algorithm used in phoneme identification / classification was implemented in Matlab.

The general structure of our implementation is presented in (Fig.4). The chain of processing consists of a preprocessing stage, a feature extraction stage and a classification stage upon which a decision is made regarding the analyzed phoneme. This is the classic structure of classification problem where data is being processed in different stages in order to obtain the desired result. A detailed description follows.

### A. Preprocessing:

Preprocessing is the stage at which the phoneme is prepared for analysis. At this point attention was paid to carefully isolate the /r/ from the subsequent vowel as this could influence the results. Remember that a vowel leaves its mark on the cepstrum. The extraction was realized manually, using a free sound editor. The length of the speech samples, were in most cases of about 100ms. Due to the fact that sounds were recorded in a silent environment, no other pre-processing than the extraction of the affected phoneme was taken.

*B. Feature Extraction*

The feature extraction stage used an implementation of the Mel-Cepstrum according to [3]. We choose this due to the efficiency and popularity of the tool as features feeder for speech / speaker recognition applications.

Briefly, Mel-Cepstrum is computed by:

a) Taking the Fourier Transform of a weighted signal (in our case a Hamming window)
b) Mapping the power spectrum obtained, using triangular overlapped windows, onto the Mel-scale.
c) Taking the logarithm of the powers of each mel-frequency.
d) Taking the Discrete Cosine Transform of the above as if it were a signal.

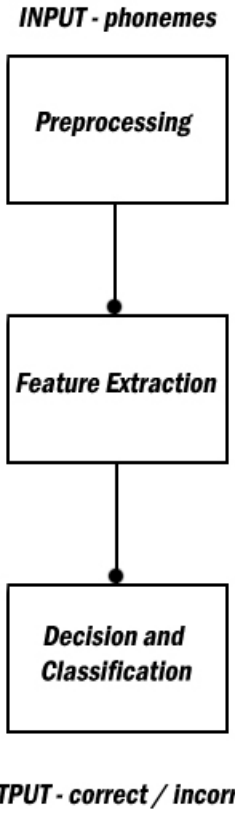


Fig. 4 Block diagram of the application

a) Speech is analyzed on small, overlapped, 15-30ms time intervals in order to maintain the stationary hypothesis of the input signal. On every such interval, before computing the spectrum, the data is being weighted using (in our case) a Hamming window in order to reduce spectral leakage:

$$X(k) = \sum_{n=0}^{N-1} w(n) \cdot x(n) \cdot \exp(-i\omega_k n) \qquad (1)$$

Where:

$$\omega_k = 2\pi \frac{k}{N} \qquad (2)$$

And $w(n)$ is the Hamming window:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad (3)$$

b) After the spectrum has been computed the the data is being mapped on the Mel-scale using triangular, overlapped filters as following:

*- decide on the number of filters*:
$m = 1, 2, \ldots, M$ is the number of bank filters, the same as the number of Mel-Cepstral coefficients ($L$).

*- compute the central frequencies (on the Mel scale) for the filters:*

$$\Delta\phi = \frac{(f_{\max})_{(Mel)} - (f_{\min})_{(Mel)}}{M+1}(Mel) \qquad (4)$$

$$\phi_c = m \cdot \Delta\phi \ (Mel) \qquad (5)$$

Where the Mel scale $(f_{\max})_{(Mel)}$, $(f_{\min})_{(Mel)}$ correspondent of the Hertz scale $f_{\max}$, $f_{\min}$ is computed using:

$$f(mel) = 2595\log_{10}\left(\frac{f(hertz)}{700} + 1\right) \qquad (6)$$

*- compute the bank filters using the converted central frequencies (Mel to Hz)*

$$H(k,m) = \begin{cases} 0 & \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)}, & f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)}, & f_c(m-1) \le f(k) < f_c(m) \\ 0 & f_c(m) \le f(k) < f_c(m+1) \\ & f(k) \ge f_c(m+1) \end{cases} \qquad (7)$$

Where the Hertz scale $f_c(m)$ correspondent of the Mel scale $\phi_c$ is computed using:

$$f(hertz) = 700(10^{f(mel)/2595} - 1) \qquad (8)$$

*- map the frequencies from Hertz to Mel scale using:*

$$X_1(m) = \sum_{k=0}^{N-1} |X(k)| \cdot H(k,m) \qquad (9)$$

c) Taking the logarithm of the above equation is the equivalent of taking the logarithm of the FFT in Hertz scale:

$$X'(m) = \log(X_1(m)) \qquad (10)$$

d) The final step in computing the Mel-Cepstrum is applying a DCT (Discrete Cosine Transform) on *X'*(*m*):

$$Ceps_{MFCC}(l) = \sum_{m=1}^{M} X'(m) \cdot \cos\left(l \frac{\pi}{M} \cdot (m - \frac{1}{2})\right) \qquad (11)$$

Where: $l = 1, .. , L$.

Example of a computed Mel-Cepstrum using 30 overlapped triangular windows, of a signal weighted by a 20ms Hamming window is presented in (Fig.5). Also in (Fig.6) the overlapped windows are presented.

Features extracted from each analyzed signal represent the standard deviation of homologous Mel-Cepstral coefficients throughout the phoneme.

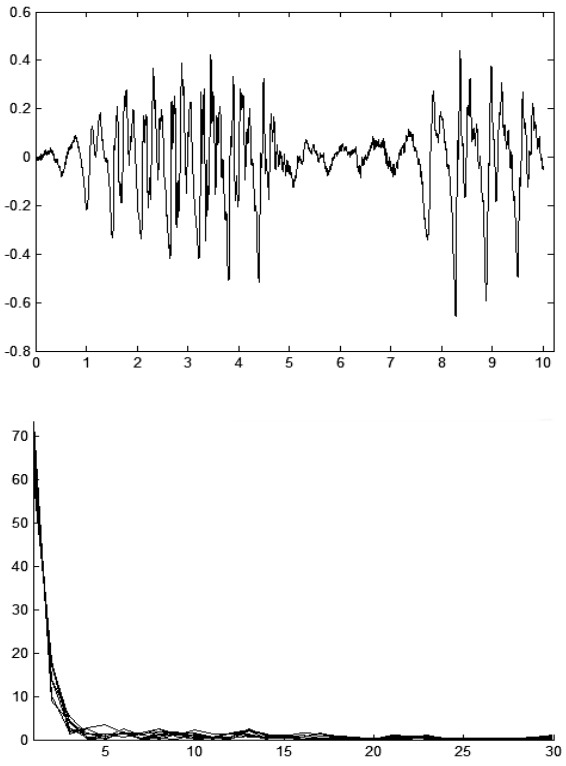

Fig. 5

/r/ phoneme as occurring in "ren" (above) and its corresponding Mel-Cepstrum.
Nine Hamming weighting windows of 20 ms were used.
The Mel-Cepstrum is computed for every such window.

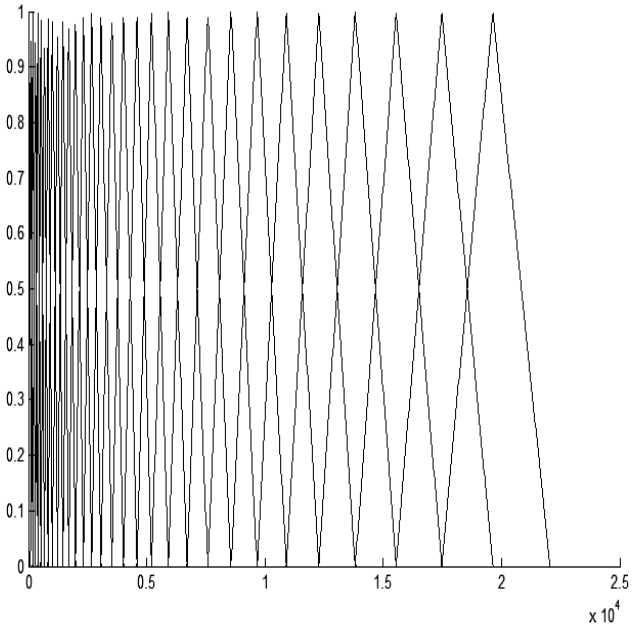

Fig. 6 Example of Overlapped Triangular Windows used in computing the Mel-Cepstrum

A bit more into detail, let us remember that analyzing a speech signal usually means analyzing short regions of only 5-20ms using weighted windows (overlapped or not). By doing so we preserve the stationary assumption.

On every such window, we compute the Mel-Cepstrum and keep only the coefficients from 6 to 10 corresponding to a region considered to mark the presence of the voice timbre. At the end of this step we shall have a set of 5 (from 6 to 10) Mel-Cepstrum coefficients corresponding to each Mel-Cepstrum computed. In all, 5**N*: five coefficients per each of the *N* analysing windows.

We now compute the standard deviation for the set of data comprised of every first Mel-Cepstrum coefficient from each of the *N* windows, then for every second and so on until fifth. The formula used is:

$$s = \left( \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \qquad (12)$$

Where

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad (13)$$

The result is a five element array (containing five standard deviations) that becomes the feature for the analyzed phoneme.

From our experiments it resulted that the timbre, over the duration of a phoneme, changes more rapidly in incorrect pronunciations compared to correct ones.

### C. Decision and Classification

Decision, in which of the two classes (correct / incorrect) an analyzed phoneme belongs, was made using a simple kNN (*k Nearest Neighbours*) classification algorithm [4].

This is an easy to implement method yet time consuming and computationally intensive in case of large set of data. Due to the fact that our database is not very large, we believed that a simple implementation is sufficient, in both computational workload and time, to give an idea on how well the feature extraction method works. We did not choose an improved version as presented in [12] because we considered the naïve implementation to be sufficient.

The algorithm is part of the supervised learning techniques and is used in many applications in the field of data mining, pattern recognition, etc. [12]. It is also a very popular algorithm according to the survey paper [11].

The algorithm presumes the existence of a set of known classes (learning / training data) already correctly classified. In our case we feed the algorithm with two classes of signals (one 'correct', one 'incorrect') comprised of manually selected phonemes on the correct / incorrect criteria. Thereafter the algorithm is fed with unclassified data (test data). For each element in this set, a distance (i.e. Euclidian) is computed to every element in the, known, learning classes. The resulting distances are ordered and using the first *k* nearest neighbours a decision is taken: the phoneme belongs to the class that gave the majority of the first *k* neighbours. [11]

More clearly, the algorithm is subsequently presented:

1. Determine the parameter *k* = number of nearest neighbors. This value is an input argument of the algorithm and, as it will be discussed, an important element that influences the overall classification rate. Generally this is chosen to be a value as high as possible.

2. Calculate the distance between the query-instance (the input data) and all the training samples. This is the measure upon which the degree of similarity between the input data and the training data is established. In our case, we used the Euclidian distance:

$$d(p,q)=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2} \qquad (14)$$

where *p, q* are two *n* dimensional arrays.

3. Sort the distances for all the training samples and determine the nearest neighbor based on the *k*-th minimum distance. This stage extracts the sorted array of the *k*-th nearest neighbors and their categories.

4. Since this is supervised learning, get all the categories of the training data for the sorted value which fall under *k*.

5. Use the majority of nearest neighbors as the prediction value. This is realized by simply counting the categories for each neighbor in the array determined at step 3.

6.

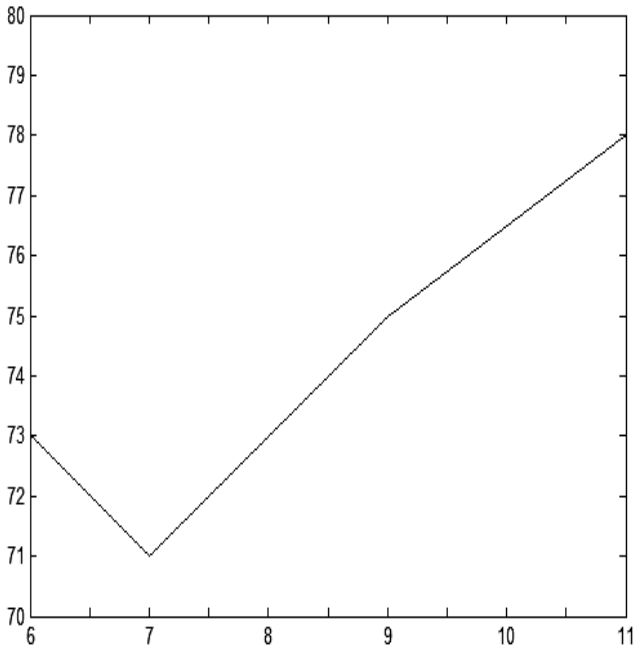


Fig. 7 Results.
Succesful classification rate (in percentage) versus *k* parameter of *k*NN algorithm

## V. RESULTS

The experiments were conducted using the parameters already above mentioned. Below, in (Fig. 7) a "succesfull classification" percentage is ploted versus the *k* parameter of the *k*NN algorithm (the number of distances that decide the phoneme's class).

What can be seen is that a 78% correct classification can be obtained once *k* is 11. Tests where not conducted above this value, however even higher correct classification percentages can be obtained (up to 95%) if the test and learning data is adapted to the sequence "/r/ - vowel". By this we refer to building both the data and learning sets using words that after

the beginning ”r” have the same vowel; yet in this case, *k* varies quite a lot from vowel to vowel.

For example, we achieved up to 90% for ”r-a-” using *k* between 3 to 5, which dropped to only 75% past this value. In the same way, for ”r-o-”, up to 83% was obtained when *k* was 7 – 11 and below this the correct classification rate dropped dramaticaly to only 50%.

This only suggests further analysis to adjust the *k* parameter of the algorithm accordingly to the type of phoneme combination processed.

The results above represent only the very few steps taken in this direction of research.

## VII. Further Development

For further development, with the goal in mind of building a fully functional system, capable of assessing in a complex manner the deficiencies of pronunciation, there are several paths to research:

- First of all, an analysis of different Mel-Cepstrum implementations (as in [4] or [5]) may be done in relationship with this kind of application. This should be studied in order to find the best solution for feature extraction.

- Secondly we discovered that the extraction of the phoneme influences the results quite a lot. A precise way to define the starting and the end point of the phoneme is to be defined in order to achieve not only a correct rule of separation but also an automated method based on current techniques. Speech segmentation is implemented in many speech processing systems; a great proportion of them using segmentation criteria based on energy analysis [8], [9] or, when it is available, on the speech sequence information. Due to the type of analysis we conduct, the difference in energy between the consonant and the subsequent vowel makes us presume that the technique will provide good result. In [10] an image segmentation algorithm for face detection for colored images is presented. The segmentation criterion is given by probability that a pixel belongs to the face hue. A similar criterion can be developed for the speech case if one can be able to find out the common characteristics of the pronunciation of the concerned sound on its whole domain of values. Problems may arise when the pronunciation is moving apart from the right model.

- One other important issue to address in further studies is the amount and type of pre-processing done on the input signal. In real life usage the system will record speech in environments affected by ambient sounds/noise. It is therefore important to study the effect of the involuntary generated noise on the performance of the algorithm and the amount of filtering required.

- Finally, and probably the most important, the released application should implement all the methods described above, switching between different segmentation or feature extraction techniques (envelope detection/ timbre analysis) according to the input signal characteristics. Also the application should display in real time an etalon measure of correct pronunciation and the current performance of the subject related to that measure. Thus the subject has information about his current status and should try to reach the etalon presented.

## VIII. Conclusion

The results in our research indicate that a possible solution by which to differentiate correct and incorrect pronunciation is to analyze the variation of timbre over the length of the phoneme.

Using standard deviation of homologous Mel-Cepstrum coefficients as a feature extraction tool, we were able to construct a simple *k*NN classifier with up to 80% correct classification performance.

However, important dependencies were discovered during the study (between the algorithm used, the segmentation of the signal and the results) that lead to further possibilities of research in this field.

Also, ignored in this study, a feature extraction method based on the variation of the shape of the envelope of the analyzed sound can be used in order to differentiate correct pronunciation from highly altered ones. This is a much simpler approach needing less calculations and can be implemented together with more complex and robust methods as the one presented.

## References

[1] D.V.Popovici, C.Buica-Belciu, A.Iordan, ”Particularitatile Fonetice ale Pronuntiei Copiilor Dislalici”, in *O Scoala Deschisa,* 2/2009, Ed.SS6SN, pp. 116-124.

[2] B.Logan, ”Mel Frequency Cepstral Coefficients for Music Modeling”, Available: http://*citeseerx.ist.psu.edu/*.

[3] F.Zheng, G.Zhang, Z. Song, „Comparison of Different Implementations of MFCC”, in *J.Comput.Sci & Technol*, Vol. 16, No. 6, 2001.

[4] M.Skowronksi, J.Harris, ”Improving the Filter Bank of a Classical Speech Feature Extraction Algorithm” in *IEEE Intl Symposium on Circuits and Systems* Vol. 4, May 2003, pp. 281-284

[5] J-H.Lee, H-Y.Jung, T-W. Lee, S-Y. Lee, ”Speech Feature Extraction Using Independent Component Analysis” in *Acoustics, Speech and Signal Processing, 2000. ICASSP '00 Proceedings*, 2000

[6] J Ben J. Shannon, Kuldip K. Paliwal, ”A Comparative Study of Filter Bank Spacing for Speech Recognition” in *Microelectronic Engineering Research Conference*, 2003

[7] I. Moldovan, *Date privind raportul dintre capacitatea de pronuntare si cea de diferentiere la palatolalici. Elemente de psihopedagogia handicapatilor*. Bucuresti, Tipografia Universitatii din Bucuresti, 1990.

[8] T. Nagarajan, and H. A. Murthy, "Subband-Based Group Delay Segmentation of Spontaneous Speech into Syllable-Like Units" in EURASIP *Journal on Applied Signal Processing*. 2004, Vol.17, pp. 2614-2625.

[9] H. A. Murthy and B. Yegnanarayana, "Formant Extraction from Group Delay Function." in *Speech Communication*. 1991, Vol. 10, 3, pp. 209-221.

[10] B. Menser and F. Muller "Face detection in color images using principal components analysis" in *Image Processing and Its Applications*. 1999, Vol. 2, pp. 620-624.

[11] X. Wu, V. Kumar, J.R. Quinlan et al. "Top 10 algorithms in data mining" – survey paper, Available: http://citeseerx.ist.psu.edu/

[12] L. Baoli, Y. Shiwen, L.Qin, "An Improved *k*-Nearest Neighbour Algorithm for Text Categorization", Available: http://arxiv.org/ftp/cs/papers/0306/0306099.pdf