# Non-linear vector interpolation by neural network for phoneme identification in continuous speech

Yifan Gong, Jean-Paul Haton

INRIA

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.(1) 39 63 55 11

Rapports de Recherche

N° 1457

*Programme 3*
*Intelligence artificielle, Systèmes cognitifs et Interaction homme-machine*

# NON-LINEAR VECTOR INTERPOLATION BY NEURAL NETWORK FOR PHONEME IDENTIFICATION IN CONTINUOUS SPEECH

**Yifan GONG**
**Jean-Paul HATON**

**Juin 1991**

# Interpolation vectorielle non linéaire par réseau neuro-mimétique pour l'identification de phonèmes en parole continue

Yifan GONG et Jean-Paul HATON
CRIN/INRIA-Nancy, B.P. 239, 54506 Vandœuvre, France

résumé :

En décodage acoustico-phonétique de la parole, les corrélations entre les vecteurs de paramètres d'analyse sont supposés spécifiques des unités phonétiques. Nous proposons dans cet article des techniques d'interpolation non linéaires pour représenter ces corrélations et identifier les phonèmes. Le principe consiste à décomposer une séquence de trames de parole en deux parties et à construire une fonction permettant d'interpoler une partie de l'information à partir de l'autre. Trois types d'interpolateurs ont été développés en fonction des informations à interpoler. Pour la reconnaissance, chaque symbole phonétique est associé à un interpolateur linéaire qui lui a été adapté au cours d'une phase d'apprentissage. Les interpolateurs sont implantés à l'aide de réseaux multi-couches de type perceptron. Les tests effectués en utilisant des vecteurs de 16 coefficients cepstraux LPCC ont montré que les trois types d'interpolateurs donnaient des résultats comparables, le meilleur étant l'interpolateur par paires de vecteurs. Ce dernier modèle donne d'aussi bons résultats globaux que la technique de quantification vectorielle tout en étant bien meilleur pour la reconnaissance des consonnes plosives.

N° de programme INRIA : 3 (Intelligence artificielle, sciences cognitives et interaction homme-machine)

# Non-Linear Vector Interpolation by Neural Network for Phoneme Identification in Continuous Speech

Yifan GONG and Jean-Paul HATON
CRIN/INRIA-Nancy, BP 239, 54506 Vandœuvre, France

**abstract :**

The correlation between vectors in a sequence of analysis frames are supposed to be specific to phonetic units in acoustic-phonetic decoding of speech. We propose non-linear vector interpolation techniques to represent this correlation and to recognize phonemes. The interpolation is based on the decomposition of a frame sequence into two parts and on the construction of a function that interpolates one part using information from the second part. According to quantities to be interpolated, three families of interpolator models are developed. In a recognition system, each phonetic symbol is associated with a non-linear vector interpolator which is trained to give minimum interpolation error for that specific phoneme. Multi-layer feedforward neural networks are used to implement the non-linear vector interpolators. For a continuous speech phoneme spotting test using 16 LPCC-derived cepstrum coefficients as parametric vectors, the three categories of models gave compatible results. *Vector-pair* interpolator models yielded best recognition rate. Compared to a VQ-coded reference technique, this model gives close global recognition rate and significantly outperforms for plosive sounds.

# Contents

# 1 Introduction

We address in this paper the problem of phoneme recognition in continuous speech. Specifically, we are interested in giving the plausibility of each phoneme symbol at each analysis frame instant. This process is called also frame classification or phoneme spotting.

Because of the contexual variability, identifying phonemes in continuous speech is a very difficult task. For continuous speech, present phoneme recognition models are not totally satisfactory. Neural networks have shown powerful capabilities in capturing arbitrary relations and have given encouraging results in phoneme recognition. However, problems as time sequence modeling, context representation and connection to the speech input remain to be investigated.

The *acoustic image* of a phoneme can be represented by a parametric vector sequence. The vectors within the sequence as well as the components of a specific vector are correlated in a way specific to that phoneme. Information carried by part of the data can be used to predict other parts of the same phoneme. This idea has been exploited in speech analysis by means of linear scalar prediction [1].

In this paper we introduce the techniques of non-linear vectorial *interpolation* to represent this correlation in phoneme recognition. We propose to build phoneme models by minimizing training set mapping error using non-linear vector interpolation. The difficult problem of interpolation is solved by the use of artificial neural networks. We describe three types of non-linear interpolation models adapted to vector sequences of continuous speech. The three neural-network-based non-linear interpolation techniques have been compared using a continuous speech database. They gave good recognition results, compared to a previously developed VQ-based reference comparison phoneme recognition technique.

Neural networks have been widely used in speech recognition for vector (parameter) to symbol (phoneme or word) mappings. Recently, their application to vector-to-vector mappings instead of vector-to-symbol mappings has been receiving increasing attention. Particularly a family of *predictors*, Linked Predictive Neural Networks [2], Neural Prediction Models [3] and Hidden Control Neural Architecture [4] have been reported. While having very interesting properties, these models use only past speech frames at time $...n-2, n-1$ to predict frame $n$. However, it is known that during speech production, the realization of the current symbol may be influenced not only by the past symbols but also by future symbols.

The non-linear interpolators described in this paper can use information in frames $n+1, n+2, ...$ as well as in $...n-2, n-1$ to interpolate the frame $n$, making better use of contexual information in speech. Besides, they can be used to interpolate not only a vector frame, but also a set of vectors derived from an input frame vector sequence. Conceptually, these non-linear interpolators are thus more general than predictors.

The paper contains the mathematical description of the three vector interpolation models, the description of the implementation of non-linear interpolators by means of neural networks, and the experimental comparison of these models.

# 2 Basic principles

## 2.1 Introduction

The basic idea of the present work consists in using non-linear vector interpolation techniques to represent the correlation between feature vectors representing phonetic units. We build a set of phoneme models such that each model has its own non-linear vector interpolator trained to give minimum interpolation error for that specific phoneme. The trained interpolators contain therefore information about phoneme symbols and can be used for recognition. We assume that, when complex enough, these non-linear interpolators can deal with contextual deformation, event sequentiality and time warping problems.

## 2.2 Formulation of the problem

A spoken utterance is make up of a sequence of symbols. A symbol $s$ is observed as an acoustic image $I_s$, represented by a sequence of profiles $v_0^s, v_1^s, v_2^s, ....$ $I_s = v_0^s, v_1^s, v_2^s, ....$

For simplicity, in the following we will drop the supscript $s$. Symbols can belong to the phoneme set of a given language. Profiles are successive analysis frames represented in some parameter space such as spectrum, cepstrum or others.

We decompose $I$ into two sets of vectors, $I_a$ and $I_b$, which are not necessarily equal in size. We suppose that the two sets of vectors are redundant, so that an approximation of $I_b$, $\hat{I}_b$, can be computed by using information in $I_a$, $\hat{I}_b = f_s(I_a)$. The process of computing $I_b$ from $I_a$ is called interpolation and $f_s$ the interpolation function. If the correlation between $I_b$ and $I_a$ is specific to the symbol $s$, then $f_s$ is specific to $s$ and can be used to recognize $s$.

We denote the interpolation error as

$$e_s = ||\hat{I}_b - I_b|| = ||f_s(I_a) - I_b|| \tag{1}$$

Once an observation of symbol $s$ is given, $f_s$ can be adjusted so that $e_s$ is minimized. If $N_s$ realizations of symbol $s$ are available, the average error of interpolation $E_s$ can be defined as:

$$E_s = \frac{1}{N_s} \sum_{k=1}^{N_s} e_s^k = \frac{1}{N_s} \sum_{k=1}^{N_s} ||(f_s(I_a) - I_b)_k|| \tag{2}$$

where $(\cdot)_k$ means that expression $\cdot$ is evaluated for the $k^{th}$ realization. $\frac{1}{N_s}$ is a normalization factor; for simplicity, it will be dropped in the following. The interpolation model of symbol $s$ can be estimated by minimizing $E_s$ over all realizations, i.e.:

$$M_s = argmin_{f_s} E_s$$

## 2.3 Three families of models

The major problem in designing an interpolation system is to decide which quantities to be interpolated. The decomposition of $I_s$ into two vector sets can result in different

types of models. We have given particular interest to three families of models: *vector-pair*, *component-pair* and *vector-center*, depending on how $I$ is divided into $I_a$ and $I_b$.

*Vector-pair* is based on the assumption that consecutive vectors in the vector sequence of a symbol are closly correlated. *Component-pair* assumes that consecutive components of a vector in a suitable parametric space are dependent. *Vector-center* exploits the fact that the realization of a speech vector is conditioned by both its past and future. If only past is considered, then the system is reduced to a traditional predictor.

The following notations will now be used: $v_m^i$ denotes the $i^{th}$ component of vector $v_m$. $\mathcal{P}$ is the parameter vector space of dimension $N$, i.e. $\mathcal{P} = \mathcal{R}^N$. $L_s$ represents the number of frames for the acoustic image of symbol $s$. $(\cdot)_k$ will be used to indicate that the expression $\cdot$ is evaluated for the $k^{th}$ realization of a symbol. Besides, on the basis of the elementary model described in section 2.2, we introduce a weighting factor $w$ in the interpolation error for all three models.

### 2.3.1 Vector-pair models

In such models, the even-numbered profiles are used to interpolate the odd-numbered profiles. A model is obtained by minimization of a weighted average interpolation error. $I$ is decomposed into:

$$I_a = [v_{2m}^i] \quad \text{and} \quad I_b = [v_{2m+1}^i]$$

where $i = [0, N-1], m = [0, [\frac{L_s}{2}]]$. Eq-2 becomes

$$E_s^v = \sum_{k=1}^{N_s} \sum_{m=0}^{[\frac{L_s}{2}]-1} \sum_{i=0}^{N-1} |(f_{s,v}^{m,i}(I_a) - v_{2m+1}^i)_k|^2 w_v^m \tag{3}$$

where:

$$f_{s,v} : \mathcal{P}^{[\frac{L_s}{2}]} \rightarrow \mathcal{P}^{[\frac{L_s}{2}]}$$

$w_v^m$ weights the error as a function of frame index $m$ so that the center of the acoustic image plays a more important role than the sides of the image.

### 2.3.2 Component-pair models

In these models, the even-numbered components of profiles are used to interpolate the odd-numbered ones. The interpolation error is minimized over all profiles. $I$ is decomposed as

$$I_a = [v_m^{2i}] \quad \text{and} \quad I_b = [v_m^{2i+1}]$$

where $i = [0, [\frac{N}{2}] - 1], m = [0. L_s - 1]$. From Eq-2 we have

$$E_s^p = \sum_{k=1}^{N_s} \sum_{m=0}^{L_s-1} \sum_{i=0}^{[\frac{N}{2}]-1} |(f_{s,p}^{m,i}(I_a) - v_m^{2i+1})_k|^2 w_p^m \tag{4}$$

Let $\mathcal{P}_1 = \mathcal{R}^{[\frac{N}{2}]}$, $f_{s,p}$ is the mapping:

$$f_{s,p} : \mathcal{P}_1^{L_s} \rightarrow \mathcal{P}_1^{L_s}$$

The reason and choice for $w_p^m$ are the same as that for $E_s^v$.

### 2.3.3 Vector-center models

In these models, a selected profile is interpolated by the remaining profiles in the acoustic image. If the selected profile is at the center of the acoustic image, i.e. $q = [\frac{L_s}{2}]$, the model interpolates the central parameter vector $v_q$ using its left and right neighboring vectors. $I$ is divided into

$$I_a = [v_m^i] \quad \text{and} \quad I_b = [v_q^i]$$

where $i = [0, N-1], m \neq q$ and $m = [0, L_s - 1]$, and $q$ an arbitrary integer and $q \in [0, L_s - 1]$. Rewriting Eq-2 yields:

$$E_s^c = \sum_{k=1}^{N_s} \sum_{i=0}^{N-1} |(f_{s,c}^i(I_a) - v_q^i)_k|^2 w_c^i \tag{5}$$

where:

$$f_{s,c} : \mathcal{P}^{[\frac{L_s}{2}]-1} \rightarrow \mathcal{P}$$

$w_c^i$ weights the error according to the vector component index $i$.

# 3 Interpolation function

In section 2, we introduced $f_{s,x}, x \in \{v, p, c\}$ as interpolation function of the model type $x$ for the phoneme symbol $s$. When the structure of this function is specified, several multi-variable optimization techniques are available to obtain it. We foculize neural network based techniques.

Multi-layer artificial neural networks are known as being powerful in capturing non-linear relations. When complex enough, they are capable of being trained to achieve arbitrary mappings [5]. The neural networks are especially applicable for approximating non-linear multi-variate functions. In this section, we discuss a neural network implementation of the non-linear vector interpolators. Three-layer feedforward neural networks are used for all these models. Minimization of interpolation error over a training database is based on error-back propagation algorithm. This training procedure minimize the least-mean-square error by iteratively adjusting the weightings in the networks [6, 7].

The input layer of a neural network receives the vector set $I_a$. This layer has therefore $N_{in} = I_{max,in} \times M_{max,in}$ cells, where $I_{max,in}$ is the number of components of the input vector in $I_a$, and $M_{max,in}$ is the number of vectors in $I_a$. The output layer of a network produces the interpolated vector set $\hat{I}_b$. The desired output values are

$I_b$. The number of cells in this layer is $N_{out} = I_{max,out} \times M_{max,out}$, where $I_{max,out}$ is the number of components of the output vector in $I_b$, and $M_{max,out}$ is the number of vectors in $I_b$. Table 1 gives the summary of $N_{in}$ and $N_{out}$ for the three types of interpolators.

Values in both input and desired output vector sets are normalized to [0,1] before connecting to the neural networks.

| Parameter | vector-pair | component-pair | vector-center |
|---|---|---|---|
| $N_{in}$ | $N \times [\frac{L_s}{2}]$ | $[\frac{N}{2} \times L_s]$ | $N \times (L_s - 1)$ |
| $N_{out}$ | $N \times [\frac{L_s}{2}]$ | $[\frac{N}{2} \times L_s]$ | $N$ |

Table 1: parameters of three proposed interpolation models

The training phase consists in doing error-back propagation on models of each phoneme symbol. The models for different phoneme symbols are trained separately and thus have no mutual influence.

# 4 Experimentation

## 4.1 Speech data and phoneme models

In order to carry out an experimental comparison, we have choosen a speech database consisting of 17 French sentences, uttered 5 times by one male speaker and manually labeled [8]. This database corresponds to about six minutes of speech. Speech signals ware sampled at 16kHz and $16^{th}$ LPCC-derived cepstrum coefficients [9] were computed each 10ms using a window length of 25.6ms. These coefficients constitute the vector parameter space.

For spoken French 37 phoneme symbols are defined. For each of the three interpolator types, 37 phoneme models were therefore constructed. Since the models are trained separately, each phoneme model may have a specific length $L_s$. The length used for a phoneme model is

$$L_s = k_s \cdot d_s \tag{6}$$

where $d_s$ is the statistical average duration over all occurrences of the phoneme $s$ in a database containing ten male speakers, and $k_s \in [0, 1]$. Typically, $L_s$ ranges from 4 to 10 profiles.

Weighting functions $w$ depend upon phoneme durations and have the form of a Hamming window.

## 4.2 Evaluation method

Recognition consists in sliding all phoneme models over the test speech frame sequence and computing the interpolation error Eq-1 frame by frame. We thus obtain interpo-

lation error sequences $e_s(n) \in [0, \infty]$, for each phoneme $s$ at each time instant $n$, which has smaller value when the test speech is close to the model of phoneme $s$ at time $n$.

We define plausibility value $\mu_{s,n} \in [0, 1]$ from $e_s(n)$:

$$\mu_{s,n} = \frac{1}{1 + c \cdot e_s(n)}$$

where $c$ is a scaling factor. The higher the value of $\mu_{s,n}$, the more plausible that the current test profile with index $n$ is the acoustic image of symbol $s$.

We have compared the interpolator models to a method based on VQ-coded reference comparison developed in our previous work [10, 11, 12], using two evaluation methods. The first method is at the sentence recognition level. It consists in comparing global recognition rates. This method is pertinent with respect to the global recognition algorithm but does not provide much information about the recognition of individual phoneme at the frame level. The second method consists in comparing correct frame labelling rate at phoneme symbol level. The difficulty for this method is to define an appropriate criterion of the correctness.

In our experiments, the following procedure was used to implement the second method. For a given phoneme symbol $s$, all frame sequences manually labeled $s$ are first concatenated into one datafile. The recognition program is then run on that file, giving plausibility of all symbols at all time index. For each time index, the symbols are sorted according to their plausibility value. A frame is considered as correctly recognized if symbol $s$ is in the first two symbols with highest plausibility. The recognition rate is defined as the ratio of the number of correctly recognized frames to the total number of frames in the datafile.

For symbols whose acoustic realization is non-stationary, as plosives, only the center part of the frame sequence will be recognized even when using a perfect recognizer. This measure therefore would under-estimate recognition rate for non-stationary sounds.

## 4.3 Results and comments

### 4.3.1 Sentence level test

VINICS continuous speech recognition system [13] is used for sentence level test. The system is recognizes a vocabulary of 1000 words described by a semantic grammar of 1200 rules. Tested on 51 test sentences, *vector-pair* models outperforms *component-pair* and *vector-center* models, giving 90% sentence recognition rate. However, VQ-based reference comparison method yields 3% better global recognition rate than *vector-pair* models.

### 4.3.2 Phoneme level test

Table 2 (closed test) and Table 3 (open test) summarize the results. In these tables, recognition rates are averaged over six categories of phonemes, i.e.: vowel, plosive,

semivowel, fricative, nasal and liquid, and pause. In the tables the recognition rates are accompanied by the number of frames correctly recognized.

The three categories of models give comparable results. The proposed models give similar recognition rate as the VQ-based reference comparison method while significantly outperformed for plosives.

We computed the difference in recognition rate between closed test and open test. The result is given in Table 4. *vector-pair* interpolator models loss 3% of recognition rate for open test, compared to 6% of loss for VQ-based reference comparison. This indicates that this type of models has better learning ability.

Experiments show that all three types of model yield fast convergence during training, requiring about 5 minutes CPU time for each on a sun *sparc* work station.

### 4.3.3 Comments

Since the VQ-based reference comparison method uses training-token-specific duration whereas the non-linear interpolators use the average duration for all training tokens of same phoneme symbol, the former has more degree of freedom. If non-linear interpolators could use training-token-specific duration, then the result will be better.

Throughout the tests, we used only one utterance of the 17 sentences for training the interpolator models. These sentences are composed of about 450 phoneme symbols. This number is far from the one that allows presenting all phonemes in all possible contexts. We believe that for some phonemes, especially semivowels and some fricatives, training is insufficient.

Besides, the following points were observed:
**a.** The order in which training samples are presented to the network does not significantly influence the recognition results.
**b.** A number between 3 and 7 of cells in the hidden layer does not change notably the recognition result.
**c.** Fourier transform, cepstrum, or liftered cepstrum coefficients give similar result, the latter being the best.
**d.** Experiences on the value of $q$ in Eq-5 shows that $q = [L_s/2]$ gives best result for vector-center models.
**e.** A reduction of training phoneme duration $L_s$ in Eq-6 to 60% of labeled duration degrades vector-pair method significantly while giving the same performance for the other two methods.

# 5 Conclusion

We have shown in this paper that the intra-vector and inter-vector correlations of a sequence of parameter vectors of speech signal can be captured by non-linear interpolation techniques and exploited for speech recognition. Because of their simplicity and learning capability, neural networks are particularly appropriate for implementing such interpolators.

We have implemented and tested three types of non-linear interpolation models having the same characteristics: input-output mapping is performed from numeric vector space to numeric vector space, not to symbolic space. These models show good capability for capturing the temporal correlation of speech and for being trained separately on each phoneme symbol. Under our experimental test conditions, the vector-pair model gave best performances, both in recognition rate and in learning ability. The result is compatible with the best results obtained using a VQ-based reference comparison method. Plosive sound recognition using vector-pair interpolator models achieves significant better result of all compared models. In our laboratory, the non-linear vector interpolation models are being integrated into an operational continuous speech understanding system.

In non-linear vector interpolation, the choice of the quantity to be interpolated is still to be investigated. Experiments on different parameter spaces and on various weighting functions could also contribute to improving the recognition rate. Adding more speech-specific constraints to the interpolators provides more information for recognition. Better results can therefore be expected. We are currently studying new non-linear interpolation models which are more specific to speech signal processing.

While successful in speech recogniton, the neural network non-linear interpolation technique presented in this paper is not specific to speech processing. As a general approach for dynamic signal modeling, it will have impact on process identification, pattern recognition or other related fields.

## Acknowledgement:

The authors gratefully acknowledge the valuable contribution to this work of Odile Mella who recorded and labelled the speech database.

# References

[1] J. D. Markel and A. H. Gray Jr. *Linear Prediction of Speech.* Springer-Verlag, New York, 1976.

[2] J. Tebelskis and A. Waibel. Large vocabulary recognition using linked predictive neural networks. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1990*, volume 1, pages 437–440, Albuquerque, New Mexico, USA, April 1990.

[3] K. Iso and T. Watanabe. Speaker-independent word recognition using a neural prediction model. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1990*, volume 1, pages 441 444, Albuquerque, New Mexico, USA, April 1990.

[4] E. Levin. Word recognition using hidden control neural architecture. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1990*, volume 1, pages 433–436. Albuquerque, New Mexico, USA, April 1990.

[5] M. L. Minsky and S. Papert. *Perceptrons.* MIT press, 1969.

[6] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 3:422, April 1987.

[7] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3), September 1989.

[8] O. Mella and M. C. Haton. Méthologie d'étude de la pertinence de paramètres phonétiques et acoustiques pour la reconnaissance du locuteur. In *Actes du séminaire sur la variabilité du locuteur*, Marseille, 1989.

[9] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech.* Prentice-Hall, Englewood Cliffs, N.J., 1978.

[10] Y. Gong. *Contribution to automatic interpretation of uncertain signals.* PhD thesis, Université de Nancy 1, France, May 1988.

[11] Y. Gong and J.-P. Haton. Phôneme based continuous speech recognition without pre-segmentation. In *Proceedings of European Conference on Speech Technology*, volume 1, pages 121–124, Edinburgh, September, 1987.

[12] Y. Gong and J.-P. Haton. Signal-to-string conversion based on high likelihood regions using embedded dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2), February 1991.

[13] Y. Gong and J.-P. Haton. Continuous speech recognition based on high plausibility regions. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1991*, Toronto, Canada, May 1991.

| Method | Vowel | Plosive | SemiVowel | Fricative | Nasliq | Pause | Global |
|---|---|---|---|---|---|---|---|
| vector-pair | 79 (1606) | 32 (565) | 61 (77) | 76 (176) | 63 (433) | 100 (216) | 61 (3073) |
| component-pair | 52 (1058) | 12 (222) | 55 (70) | 67 (156) | 62 (427) | 90 (195) | 42 (2128) |
| vector-center | 68 (1393) | 24 (426) | 52 (66) | 77 (179) | 61 (422) | 94 (204) | 53 (2690) |
| reference | 84 (1698) | 24 (429) | 63 (80) | 77 (179) | 81 (560) | 98 (213) | 63 (3159) |
| Total Frames | 2019 | 1728 | 126 | 230 | 685 | 216 | 5004 |

Table 2: Comparison of three proposed methods (test phrases were used in training)

| Method | Vowel | Plosive | SemiVowel | Fricative | Nasliq | Pause | Global |
|---|---|---|---|---|---|---|---|
| vector-pair | 63 (583) | 43 (228) | 37 (28) | 52 (140) | 52 (182) | 100 (220) | 58 (1381) |
| component-pair | 29 (270) | 9 (48) | 16 (12) | 24 (65) | 45 (158) | 100 (220) | 32 (773) |
| vector-center | 56 (516) | 20 (107) | 71 (53) | 54 (147) | 50 (175) | 62 (137) | 48 (1135) |
| reference | 65 (597) | 22 (117) | 51 (38) | 64 (173) | 66 (231) | 88 (195) | 57 (1351) |
| Total Frames | 914 | 522 | 74 | 268 | 350 | 220 | 2348 |

Table 3: Comparison of three proposed methods (test phrases were not used for training)

| Method | Vowel | Plosive | SemiVowel | Fricative | Nasliq | Pause | Global |
|---|---|---|---|---|---|---|---|
| vector-pair | 16 | -11 | 24 | 24 | 11 | 0 | 3 |
| component-pair | 23 | 3 | 39 | 43 | 17 | -10 | 10 |
| vector-center | 12 | 4 | -19 | 23 | 11 | 32 | 5 |
| reference | 19 | 2 | 12 | 13 | 15 | 10 | 6 |

Table 4: Difference in recognition rate between closed test and open test

ISSN 0249-6399