Non-Linear Prediction Of Speech Using ANFIS: Comparison With Neural Nets.

A. Kaboli and M.H. Savoji
Electrical and Computer Engineering Faculty
Shahid Beheshti University
Evin Square, Tehran 1983963113, Iran.
e-mail: mh_savoji@yahoo.com

ABSTRACT
The neuro-fuzzy non-linear prediction of long segments of speech, as long as whole vowels, using ANFIS is reported in this paper and comparisons are made when neural nets are used for the same purpose.
Emphasis is put on the generalization properties of the trained fuzzy inference system when both intra-vowels and inter-vowels variability are considered. The data-base used is composed of Farsi vowels whose waveforms are sampled at 11 and 22 KHz and digitized at 8 and 16 bit resolution. The effects of sampling frequency and bit resolution on the working of ANFIS are also reported. It is shown that although results are qualitatively similar to those obtained using neural nets, ANFIS has the ability to train more quickly, in just a few epochs, and is more apt to tune in a given data set. The tuning is more pronounced when the input data is of wider bandwidth.

INTRODUCTION
The prediction of speech has applications in speech technology, specially coding. Linear prediction is used conventionally to reduce the redundancy of speech signal and decrease the coding bit-rate. Considering the non-linearity that exists in speech production should lead to lower dynamics of the signal to be coded with a consequent reduction in bit-rate and needed bandwidth. Artificial Neural Nets (ANN) are usually used to this effect and studies have shown that 2 to 3 dB further reduction in the coder gain factor is possible when short frames of speech are analyzed with ANN. However, this further reduction is not observed when segments as long as whole vowels are considered [1] e.g. in speech synthesis by waveform concatenation using LP-PSOLA method.
Neural Nets are attractive to use in non-linear problems because they are basically model-free estimators that can learn from experience. Similar to NN, fuzzy Systems can provide an estimation function without a mathematical model of how outputs depend on input data. This property gives opportunity to these systems to learn from experience with numerical or linguistic data [2]. Therefore, Neuro-Fuzzy computing i.e. soft computing approaches to system modeling has also attracted the attention of many researchers in the past recent years [3], [4]. This is because, in addition to the fact that they are model free, neuro-fuzzy methods possess both the low-level learning and computational power of neural networks and the advantages of high-level human like thinking of fuzzy systems making them a very powerful and versatile tool in non-linear modeling problems.

1- Neuro-Fuzzy Speech Prediction Using ANFIS.
The type of fuzzy model first suggested by Takagi and Sugeno [5], [6] uses fuzzy inputs and rules but its outputs are non-fuzzy sets. It provides a powerful tool for modeling complex non-linear problems when combined with a network structure as in Adaptive Network Fuzzy Inference System or ANFIS [7].
ANFIS can be applied to non-linear prediction of speech where past samples are used to predict the sample ahead. However, the number of ANFIS parameters augments exponentially when the number of input variables i.e. the number of past sample values that is necessary for a good prediction increases. The number of ANFIS parameters compares usually unfavorably with the number of parameters that characterize a NN. But the versatility of ANFIS suggests that it may succeed where neural nets have failed. That is why we applied it to non-linear prediction of long segments of speech.

1-1 ANFIS.
ANFIS is a class of adaptive multi-layer feed-forward networks that is functionally equivalent to a fuzzy inference system. It was proposed in an effort to formalize a systematic approach to generating fuzzy rules from an input-output data set. A typical fuzzy rule in a Sugeno fuzzy model has the format:

If x is A and y is B then z=f(x,y)

Where A and B are fuzzy sets in the antecedent; z=f(x,y) is a crisp function in the consequent. Usually this function is a polynomial of the input variables x and y, but it can be any other function that can approximately describe the output of the system within the fuzzy region specified by the antecedent of the rule.

When f is a constant we have the zero-order Sugeno fuzzy model that is functionally equivalent to a radial basis function network under certain constraints [8]. When f (x,y) is a first-order polynomial, the model is called first-order Sugeno fuzzy model and is what was originally proposed. Consider such a model that contains two rules:

Rule 1: If X is A_1 and Y is B_1, then

$$f_1 = p_1 x + q_1 y + r_1$$

Rule 2: If X is A_2 and Y is B_2, then

$$f_2 = p_2 x + q_2 y + r_2$$

Figure 1(a) illustrates graphically the fuzzy reasoning mechanism to derive an output f from a given input [x,y]. The firing strengths w_1 and w_2 are usually obtained as the product of the membership grades of the premise part, and the output f is the weighted average of each rule's output. Part (b) of Figure 1 shows the corresponding ANFIS structure where nodes within the same layer perform functions of the same type as detailed below. Note that O_i^j denotes the output of the i-th node in j-th layer.

Layer 1: Each node in this layer generates a membership grade of a linguistic label. For instance, the node function of the i-th node may be a generalized bell membership function:

$$O_i^1 = \mu_{A_i}(x) = 1/(1+|(x-c_i)/a_i|^{2b_i}) \quad (1)$$

Where x is the input to node i; A_i is the linguistic label (small, large, etc.) associated with this node; and $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the premise parameters.

Layer 2: Each node in this layer calculates the firing strength of a rule via multiplication:

$$O_i^2 = w_i = \mu_{A_i}(x)\ \mu_{B_i}(y),\ i = 1,2 \quad (2)$$

Layer 3: Node i in this layer calculates the ratio of the i-th rule's firing strength to the total of all firing strengths:

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \qquad (3)$$

Layer 4: Node i in this layer compute the contribution of i-th rule toward the overall output, with the following node function:

$$O_i^4 = \overline{w_i}\ f_i = \overline{w_i}\ (p_i x + q_i y + r_i), \quad (4)$$

Where $\overline{w_i}$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are referred to as the consequent parameters.

Layer 5: The single node in this layer computes the overall output as the summation of contribution from each rule:

$$O_1^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (5)$$

ANFIS in figure 1 (b) has, as the basic learning rule, the back-propagation gradient descent algorithm (the same used in feed-forward Neural Nets) which calculates the error signals recursively from the output layer backward to the input nodes.

From this architecture, it is seen that given the values of premise parameters, the overall output f can be expressed as a linear combination of the consequent parameters:

$$\begin{aligned} f &= \overline{w_1} f_1 + \overline{w_2} f_2 \\ &= (\overline{w_1} x) p_1 + (\overline{w_1} y) q_1 + (\overline{w_1}) r_1 \\ &\quad + (\overline{w_2} x) p_2 + (\overline{w_2} y) q_2 + (\overline{w_2}) r_2 \end{aligned}$$

Based on this observation J.S.R. Jang proposed a hybrid algorithm which combines the gradient descent and the least-squares method for an effective search of optimal parameters [9]. ANFIS is available in the Fuzzy Logic Toolbox of MATLAB [10].

2- Using ANFIS In MATLAB

Here, we recall some of the main functions available in the Fuzzy Logic Toolbox of MATLAB and explain their main features.

2-1 Genfis1 and Genfis2

The genfis1 function creates a Fuzzy Inference System (FIS) without using data clustering. In its general form it is called using the following format:

fismat=genfis1(data,nummfs,inmftype,outmftype)

where:

data is the training data matrix where all columns except the last one are input and the last column is the output data vector.

nummfs determines the number of membership functions for each input which can be an integer applied to all inputs or a matrix whose number of columns equals the number of inputs.

inmftype defines the type of the membership functions of the inputs. It can be a string of characters or an array of strings.

outmftype determines the type of the membership function for the output and, since the FIS structure here is of the Sugeno type, the system has only one output that can be either linear or constant.

The genfis2 function is similar to genfis1 but uses the subtractive clustering to create a fuzzy inference system.

2-2 The ANFIS Function

This function is used to train the created FIS. In its more general form, this function is called with the following parameters:

[fismat,error1,stepsize,fismat2,error2]=anfis(trndata,trnopt,dispopt,chkdata,optmethod)

where:

trndata is the training matrix.

fismat is the original FIS created by genfis1 for instance.

trnopt is a vector defining the training conditions such as the number of training epochs, the target error, the initial step size in updating the parameters, the rates of decrease and increase of the step.

chkdata is the data used to avoid over-fitting with the same format as trndata.

fismat1 is the optimum FIS returned whose parameters are calculated for the minimum training error.

error1 and error2 are the arrays of RMS errors during different training epochs for the training and check data.

fismat2 is the FIS returned whose parameters are set for the minimum error on the check data.

2-3 The Evalfis Function

This function permits to evaluate the trained FIS by the previous ANFIS function for a test data.

3- ANFIS Used For Non-Linear Prediction Of Farsi Vowels

The above functions have been used on a speech data-base of Farsi vowels to train and test ANFIS for non-linear prediction of long segments of speech.

3-1 The Speech Data-Base

The waveforms of Farsi (Persian) phrases and words uttered by two male speakers were recorded at 11 and 22 KHz sampling frequencies and digitized with 8 and 16 bits. Then words were segmented into syllables to be saved in separate files as items of our data-base. The phonetic description of the files' contents and other characteristics such as the speaker code and the code of microphone used were attached to each file. A search engine permits to extract all files with a specific phonetic content and other needed characteristics such as the sampling frequency or bit representation for different experiments.

3-2 The Experiments

The following experiments were conducted to examine the efficiency of ANFIS for the non-linear prediction of long segments of speech one sample ahead. Here, the generalization capabilities of the trained structure play an essential role. Since the number of rules in ANFIS and consequently the number of its parameters increases exponentially with the number of inputs, five was the maximum number of preceding samples that could be used.

In these experiments, the number of the membership functions for each input was set to two, justified by the input values being of both polarities, and the type chosen was Bell shaped that had given better results in preliminary tests. Therefore the total number of ANFIS parameters can be calculated as follows: Each membership function has three parameters, therefore for five inputs we have 5*2*3=30 premise parameters. On the other hand five inputs create 2^5 rules giving rise to 32*(5+1) =192 consequent parameters. Then, the total number of parameters is 192+30=222. Since the training data must be about 5 times the number of parameters [11], the vowels that could be used for training were limited to those at least 1110 samples long.

These experiments were conducted in 4 groups and in each group of experiments it was tried to use the results obtained before in order to achieve an overall assessment of the efficiency of ANFIS for the task in hand and compare the obtained results with those of neural nets reported elsewhere [1].

3-2-1 Defining The Best Five Inputs

In this group of experiments we tried to find out how the prediction error varies with regards to the interval between the five neighbouring samples used to predict the following one.

The experiments were conducted on the vowel O in different contexts. It was observed that increasing the interval between preceding samples from 1 to 3 led to a net increase in the prediction error. It was also observed that ANFIS was trained very quickly in the first few epochs and that increasing the training time did not result in an appreciable lower error. Figure 2 summarizes these results. Signals used were sampled at 22 KHz with 16 bit resolution. Figure 3 shows the

evolution of the prediction error versus the epoch number in both cases.
It was concluded that the five input samples used in prediction of the next one must be adjacent.

3-2-2 The Effect Of Sampling Frequency And Bit Resolution On The Generalization Of ANFIS For Vowels Of The Same Kind

Here we tried to study the effect of sampling frequency and bit resolution on the working of ANFIS. The experiments were conducted on two sets of vowels A and E along with validation. Using validation data during training permits access to the FIS that has resulted in minimum error on validation data. In opposition to neural nets, here ANFIS continues the training of the FIS using further the training data.
As regards the sampling frequency, it was observed that the FIS trained with training and validation data at 11 KHz (either with 8 or 16 bit resolution) could well be used for data sampled at 22 KHz but the reverse was not true.
As for the bit resolution, it was concluded that the prediction error was almost the same when 8 bit or 16 bit digitized signals were used in both training and test i.e. the FIS was immune to quantization noise; a well known property in neural nets.
Figure 4 compares the prediction error for vowel A at 8 and 16 bit resolution.

3-2-3 ANFIS Generalization Between Different Vowels When Training Is Controlled Using A Vowel Of The Same Kind

In this group of experiments, we tried to study the generalization between different vowels of a trained FIS when over-training was controlled by using validation data of the same kind. For example the FIS trained and checked with vowels A was tested using a vowel E.
Signals sampled at 11 KHz were used for training and validation whilst files of E vowels either sampled at 11KHZ or 22 KHz were employed for test. The Generalization of training of vowels A was quite good for vowels E sampled at 22KHz (8 bit) but not acceptable for signals of vowels E sampled at 11 KHZ (8bit). It was also noted that the prediction error when the FIS was trained for a vowel and tested with another depended also on the context of the test vowel. For instance, when vowel E in the context LI was used as opposed to E in the context WI, the error was lower for the LI file at 11 KHz (8 bit) than for the WI signal sampled at 22 KHZ (8bit).
Figure 5 shows the effect of the sampling frequency on the generalization in this case.

3-2-4 ANFIS Generalization Between Vowels When Training And Validation Include Vowels Of Different Kinds

In these experiments we tried to study further the inter-vowel generalization capabilities of ANFIS by combining, during training, different vowels of different sampling frequencies and bit resolutions. For example in one example only vowels A and E of 22 KHz and 16 bit were used for the training of the FIS that was then tested with different vowels. These results were compared with when training included vowels A and E of different sampling frequencies and bit resolutions.
We can summarize the general observations of these experiments as follows: When the test signals of different vowels were all of 22KHz and the training was carried out using vowels A and E only, the prediction error was notably higher than when the FIS was trained using a combination of different vowels. Here using vowels of different bit-rates during training did not have much effect on the results. But using training data that did not include the test vowel led to better results when files of 11 KHz were used in training as opposed to when solely signals of 22 KHz were employed. The inter-vowel generalization at 11 KHz got better, as in the case of signals of 22 KHz, when more varied combinations of vowels were used for training but the prediction was higher for these files than those of 22 KHz.
Figure 6 shows the effect of using a combination of different vowels during training in comparison to using only one vowel.

CONCLUSION

The following general conclusions can be made as regards the main results obtained in this work:

1- Although results are qualitatively similar to those obtained using neural nets, as far as the prediction error is concerned, they are achieved with less input data.
2- The computation time in the training phase is comparable to that of NN when the complexity of ANFIS structure is taken into account. The convergence, here, takes usually a few epochs to reach its steady state.
3- Best results are obtained when the past samples are adjacent and constitute the previous immediate neighbours.
4- Generalization of the network is quite good, for 22 KHz files, for both intra-vowels and inter-vowels variability.

5- The prediction error reduces when the network is trained on various input vowels as compared to when it is trained only with the same vowel suggesting that ANFIS, more than Neural Nets, has the capability to tune in a specific data set.
6- Using validation data is often of no consequence on generalization for same vowels confirming the tuning in of ANFIS to a given data.
7- For 22 kHz files, the bit resolution has no important effect on the prediction error.
8- The sampling frequency i.e. the input data bandwidth has a decisive role on the results in the sense that it permits fine tuning to a given data set.

REFERENCES

1. K Ashouri, M. Amini and M.H. Savoji; "Non-linear prediction of speech signal using artificial neural nets"; EURASIA-ICT-2002, IRAN, October 2002.

2. M.R. Emami, I.B. Turksen and A.A. Goldenberg; "Development of a systematic methodology of fuzzy logic modeling"; IEEE Trans. Fuzzy Syst. Vol. 6, N0. 3, August 1988.

3. J.S.R. Jang and C.T. Sun; "Neuro-fuzzy modeling and control"; Proc. Of the IEEE, March 1995.

4. J.S.R. Jang and C.T. Sun; "Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence"; Prentice Hall, 1997.

5. T. Takagi and M. Sugeno; "Fuzzy identification of systems and its application to modeling and control"; IEEE Trans. Syst. Man, Cybern., Vol. SMC-15, January 1985.

6. M. Sugeno and T. Yasukawa; "A fuzzy-logic-based approach to qualitative modeling"; IEEE Trans. Fuzzy Syst., Vol. 1, No. 1, February 1993.

7. J.S.R. Jang; "ANFIS: Adaptive Network –based Fuzzy Inference System"; IEEE Trans. Syst. Man, Cybern., Vol. 23, No. 3, May/ June 1993.

8. J.S.R. Jang and C.T. Sun; "Functional equivalence between radial basis function networks and fuzzy inference systems"; IEEE Trans. On Neural Networks, Vol. 4, No. 1, Jan. 1993.

9. J.S.R. Jang; "Fuzzy modeling using generalized neural networks and Kalman filter algorithm"; Ninth National Conf. on Artificial Intelligence (AAAI-91), July 1991.

10. J.S.R. Jang and N. Gulley; "The Fuzzy Logic Toolbox for use with MATLAB". The MathWorks Inc., Natick, Massachusetts, 1995.

11. Fuzzy Logic Toolbox for use with MATLAB – User's Guide (version 2). The MathWorks Inc., Natick, Massachusetts, 2000.

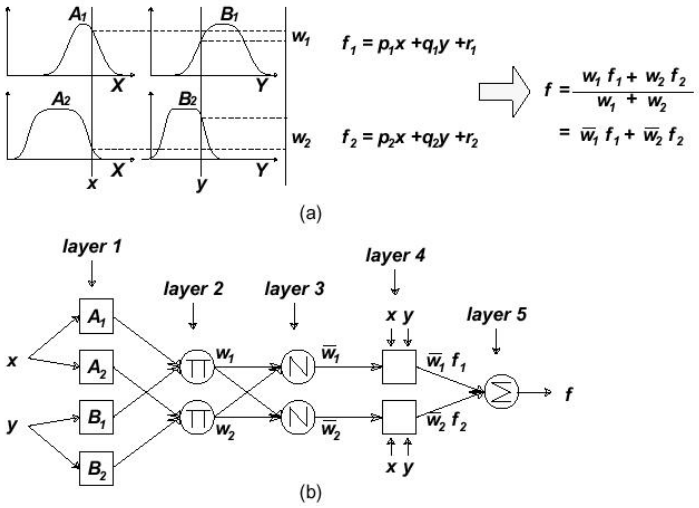

Figure 1. (a) First-order Sugeno fuzzy model; (b) corresponding ANFIS architecture.

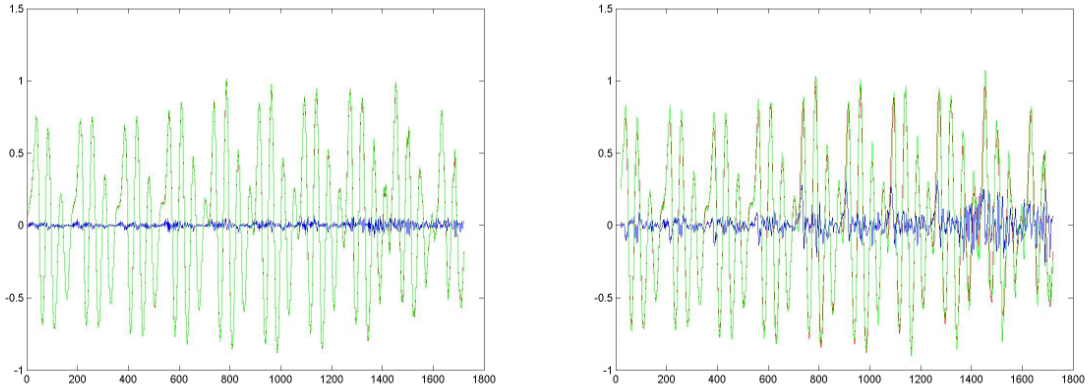

Figure 2. The predicted signal and error when samples are 1 interval distant (left) as compared to when the distance between them is 3 intervals (right).

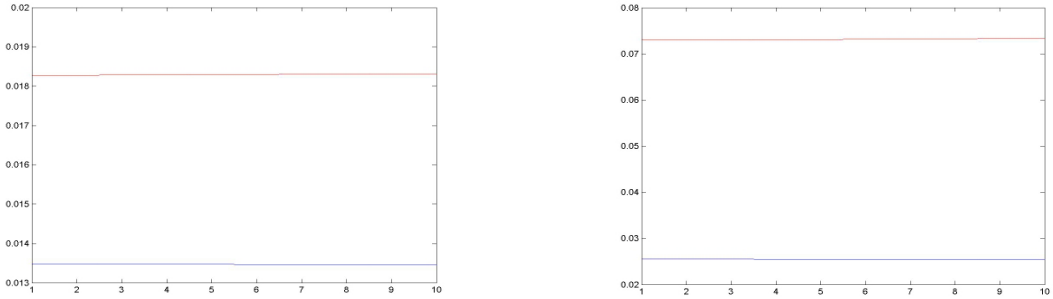

Figure 3. The evolution of the prediction error during training and test in both cases of 1 (left) and 3 (right) samples distance.

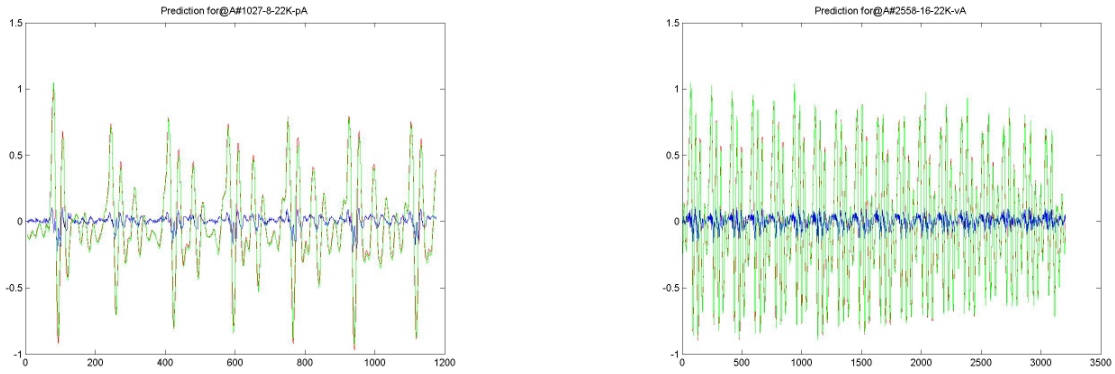

Figure 4. Comparison of prediction error for vowel A at 8 (left) and 16 bit (right) resolution (sampling frequency 22KHz in both cases).

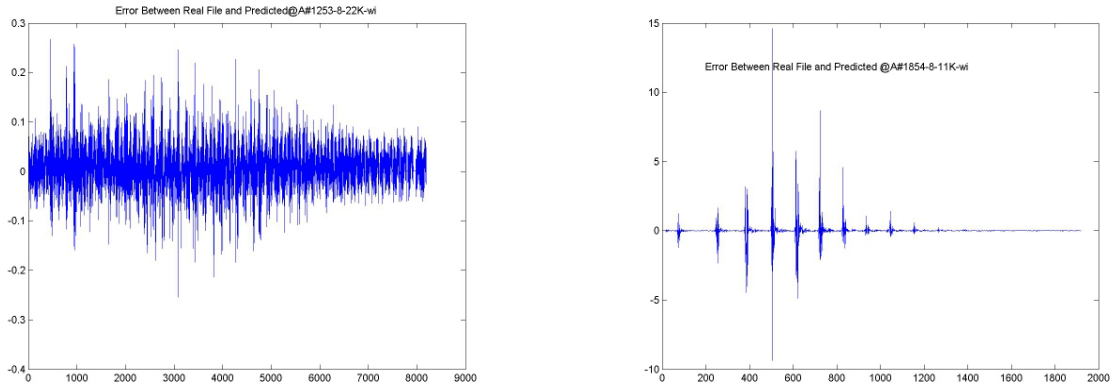

Figure 5. Comparison of prediction error of ANFIS trained for A vowel and tested with E vowel at 22 KHz (left) and 11KHz (right) sampling frequency.

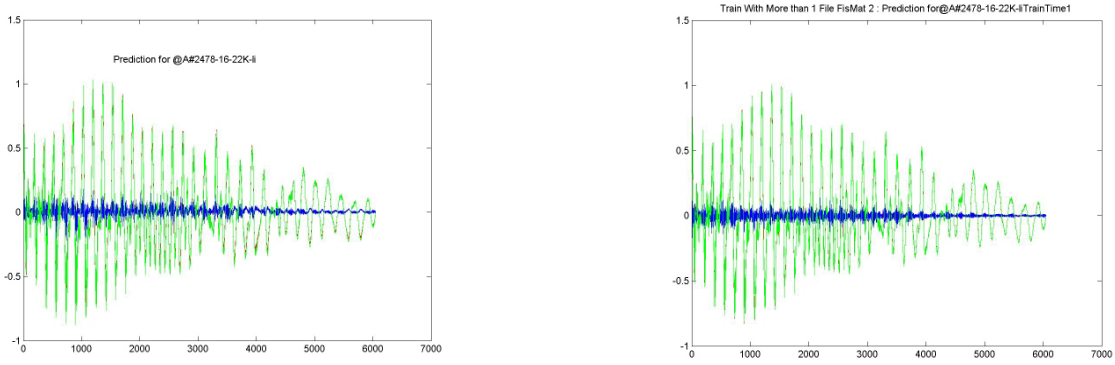

Figure 6. The predicted and error signals of the same vowel E when vowel A only is used in training (left) and when a combination of vowels is employed (right).