

# WinSPT - A Software Tool for Speech Signal Processing

Athanasios Margaris<sup>1</sup>, Stauros Souravlas, Efthimios Kotsialos, and Manos Roumeliotis  
University of Macedonia, Department of Applied Informatics  
Egnatia 156 Str, GR 540 06, Thessaloniki, Greece  
emails: [amarg,sourstav,ekots,manos]@uom.gr

## Abstract

The objective of this paper is the presentation of a Windows application that can be used to perform the necessary pre-processing steps to be applied to a speech signal in order to be used in tasks such as speech synthesis and speech recognition. The described application has been implemented using the Visual C++ programming environment and it runs under the Microsoft Windows operating system.

## I. INTRODUCTION

Speech technology is the branch of science dealing with the processing of speech signals; this processing is a required step in tasks such as speech and speaker recognition, speech synthesis and speech understanding, and includes among others, the filtering of signals to remove the undesired frequency regions, the frame blocking stage to construct windowed speech frames with a length of a few hundred samples, and the estimation of the value of useful parameters such as the LPC and Cepstral coefficients used as feature vectors in speech recognition projects. There are many applications that perform speech processing tasks, such as the PRAAT software package of the University of Amsterdam [1], the Speech Analyzer and the CECIL applications created by SIL International [2] as well as the ELAN speech annotator [3] implemented by the Max Plank Institute for Psycholinguistics. A complete description of those applications as well as the full list of the supported features can be found in the literature.

The objective of the implementation of the WinSPT application is the derivation of training set files compatible with the Neural Workbench Simulator [4] created and used by the Neural Group staff of the Applied Informatics Department at the University of Macedonia. These files are composed of LPC and Cepstral coefficient vectors and are used in speech recognition projects based on back propagation neural networks.

## II. SPEECH PROCESSING

The objective of speech processing is the extraction of feature vectors capable of identifying isolated word frames. In this project, this technique is applied on speech samples recorded by an ordinary microphone and a standard on-board PC sound card. The sampling frequency of this recording technique is usually equal to 11.025 KHz, with each speech sample being recorded with a precision of about 8 bits. However, the WinSPT application is capable of loading any sound file saved in WAV file format. After the loading of the file in the computer memory, the following operations are applied to the speech signal [5]:

(1) Pre-emphasis: in this step the speech signal  $s(n)$  is passed through a first order FIR filter, to be spectrally flattened. The pre-emphasis filter used in this step is described by the equation

$$H(z) = 1 - \alpha z^{-1} \quad (1)$$

with the parameter  $\alpha$  having a value of 0.950. By applying this filter, the *DC* component of the signal is completely removed.

(2) Frame Blocking: in this step, the pre-emphasized signal produced in the previous stage is separated into frames of  $N$  samples, with the adjacent frames overlapping in time. The size of this overlap region is equal to  $M$  samples. If we denote the sample number with  $L$ , then the number of overlapping frames produced in this way is equal to  $W = (L - M)/(N - M)$ . The samples of these frames are stored in a two dimensional matrix *Win*, with

<sup>1</sup>Corresponding author

dimensions  $W \times N$ . In linear memory addressing, the expression  $Win[i][j] = Buf[i * (N - M) + j]$  holds, where  $Buf$  is the memory buffer that keeps the speech data.

(3) Windowing: in this step each of the speech frames is windowed in order to minimize the signal discontinuities at the borders of the frame. The window type used in this project is the Hamming window defined by the equation

$$h(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \quad n \in [0, N-1] \quad (2)$$

(4) LPC analysis: in this stage a vector of LPC or Cepstral Coefficients is calculated for each windowed speech frame. This stage is very important, since the calculated feature vector identifies the speech frame and therefore it is used during the recognition stage. In the first step, the autocorrelation coefficients are obtained by the equation:

$$r(p) = \sum_{n=0}^{N-1-p} x(n)x(n+p) \quad (3)$$

where the symbol  $x(i)$  denotes the speech sample located at the  $i_{th}$  frame position, while  $p$  is the number of LPC coefficients to be calculated, known as LPC order. After the calculation of the autocorrelation vector, the LPC coefficients can be calculated by applying the Durbin method [6] and the following recursive expressions developed by Furui [7]:

$$\begin{aligned} l = 0 & : E(l) = r(l) \\ l = 1 & : k(l) = \frac{r(l)}{E(l-1)}, \quad E(l) = E(l-1)[1 - k(l)^2], \quad \alpha_l^l = k(l) \\ l > 1 & : k(l) = \frac{1}{E(l-1)} \left( r(l) - \sum_{i=1}^{l-1} \alpha_i^{l-1} r(l-i) \right), \\ & E(l) = E(l-1)[1 - k(l)^2], \quad \alpha_m^l = \alpha_m^{l-1} - k(l)\alpha_{l-m}^{l-1}, \quad \alpha_l^l = k(l) \end{aligned}$$

where  $m \in [1, l] \forall l \in [1, p]$ , while the  $n^{th}$  LPC coefficient is denoted by  $\alpha_n = \alpha_n^p$ . Having calculated the LPC coefficients, the LPC Cepstrum Coefficients (or Cepstral Coefficients) can be obtained using the equations

$$c_m = \begin{cases} \sum_{k=1}^{m-1} \frac{kC_k\alpha_{m-k}}{m} + \alpha_m & m \in [1, p] \\ \sum_{k=1}^{m-1} \frac{kC_k\alpha_{m-k}}{m} & p < m < q \end{cases}$$

where  $q$  is the number of Cepstral coefficients, known as Cepstral order.

### III. THE WINSPT APPLICATION

The main advantage of the WinSPT application is that it supports the processing not only of a single file but also of a group of files in a batch mode operation. After the loading of the selected WAV file, the program automatically performs all the speech processing steps described above by using the parameter values defined by the user. A typical screenshot of the WinSPT application after the loading of some speech file is shown in Figure 1.

The most important features of the WinSPT application as they are shown in Figure 1, are the following:

(1) The program can work in data mode as well as in frame mode. In data mode the program plots the contents of the whole data file, while, in frame mode, the consecutive speech frames are plotted one after the other. The samples shown in red colors are associated only with the current frame, while, the samples plotted with blue color belong to the overlap regions of the current frame with the previous and the next frame. Each time the user is moving between frames by using the horizontal scrollbar, the LPC, Cepstral, and Fourier Coefficient diagrams are updated accordingly, to show the coefficients associated with the currently selected frame. The display of the horizontal axis of those diagrams that shows the zeroth value can be done by checking the "Show Axis" check box located below the main list box of the window. In both cases, as the user moves the mouse cursor to each one of the diagrams, the cursor position, as well as the speech sample of frame associated with it, are displayed to the appropriate labels that are updated in real time.

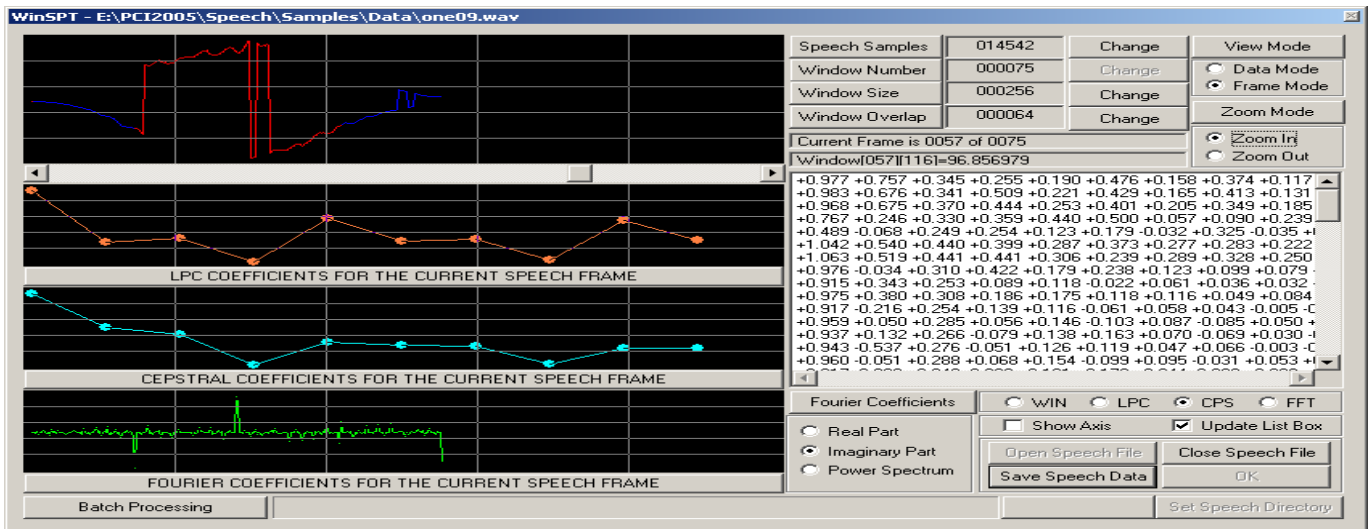


Fig. 1. A typical screenshot of the WinSPT application.

(2) The type of coefficients associated with the Fourier analysis is determined by the group of the three radio buttons located at the bottom of the window and labelled as "Real Part", "Imaginary Part", and "Power Spectrum". Recalling that each Fourier coefficient is a complex number in the form  $c = \alpha + i\beta$  it is clear that those three buttons allow the plotting of the real and the imaginary parts of the Fourier coefficients as well as the power spectrum defined the sequence  $C_i = |c_i| = \sqrt{\alpha_i^2 + \beta_i^2}$  ( $i = 1, 2, \dots, N$ ). A typical screen shot of these three different plot types is shown in Figure 2.

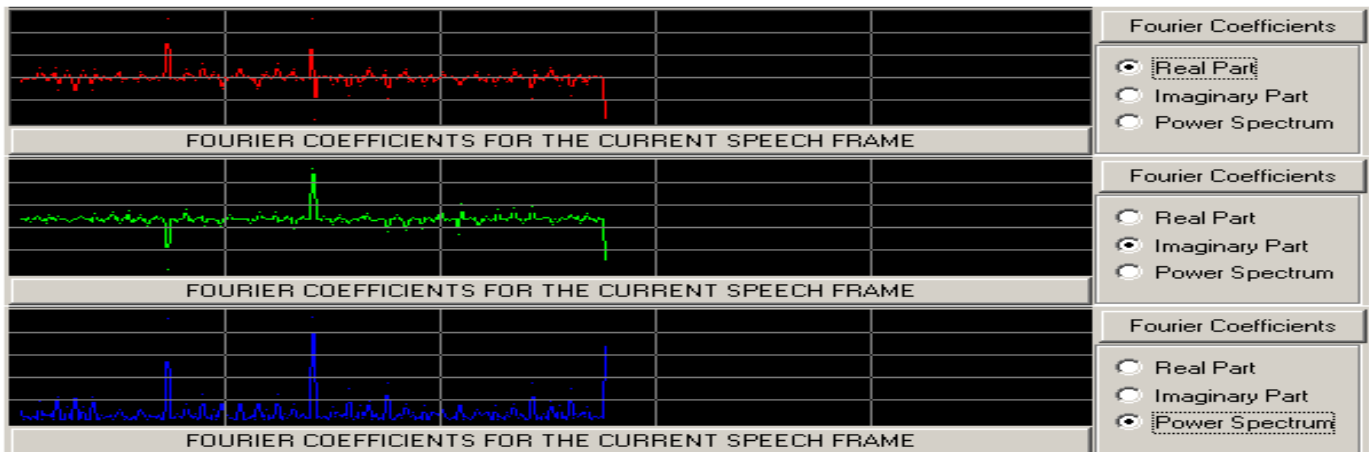


Fig. 2. Fourier coefficients data plotting

(3) The user has the ability to display the values of the parameters of the speech frames (namely the speech data as well as the LPC, Cepstral and the selected Fourier coefficient type) by checking the "Update List Box" check button. The type of data values to be displayed is determined by the group of the four radio buttons labelled "WIN", "LPC", "CPS" and "FFT" for the speech frames data, and the LPC, Cepstral and Fourier coefficients respectively.

(4) The user can change the values of the sample number, the window size and the overlap size (the number of speech frames is estimated automatically by the equation  $W = (L - M)/(N - M)$  and it can not be changed by the user). The program can zoom in and zoom out to preview the whole data file with the zoom option to be available in both data and frame modes. This zoom capability is shown in Figure 3 for the combination Zoom In/Data Mode (Figure 3a), Zoom In/Frame Mode (Figure 3b) and Zoom Out/Frame Mode (Figure 3c).

(5) The user has the ability to process a set of speech files in one step through a batch operation; in this case the program performs the speech processing steps for each one of the selected files, and then, generates the final

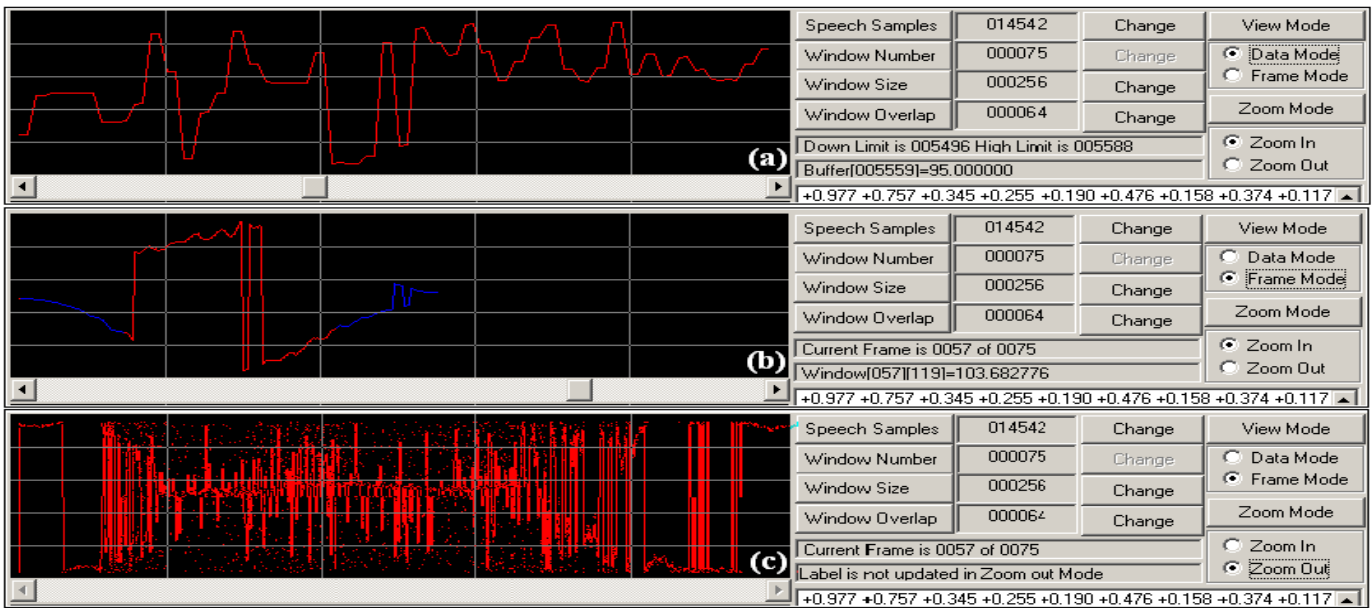


Fig. 3. The Zoom capabilities of the WinSPT application

estimates from the mean values of each parameter. For example, if the user specifies ten different speech files, then, the LPC coefficients will be estimated for each frame of each file, and then, the coefficients associated for some frame (for example the fifth frame) will be found by averaging the coefficient values of the fifth frame of each file. To avoid run time errors associated with the fact that the specified speech files may have different numbers of samples and therefore, different numbers of speech frames - the frame size is the same for all files - the number of frames for each file is estimated in advanced, and during the batch processing the minimum frame number is used as the number of frames of each speech file. The batch operation of the WinSPT application is shown in Figure 4.

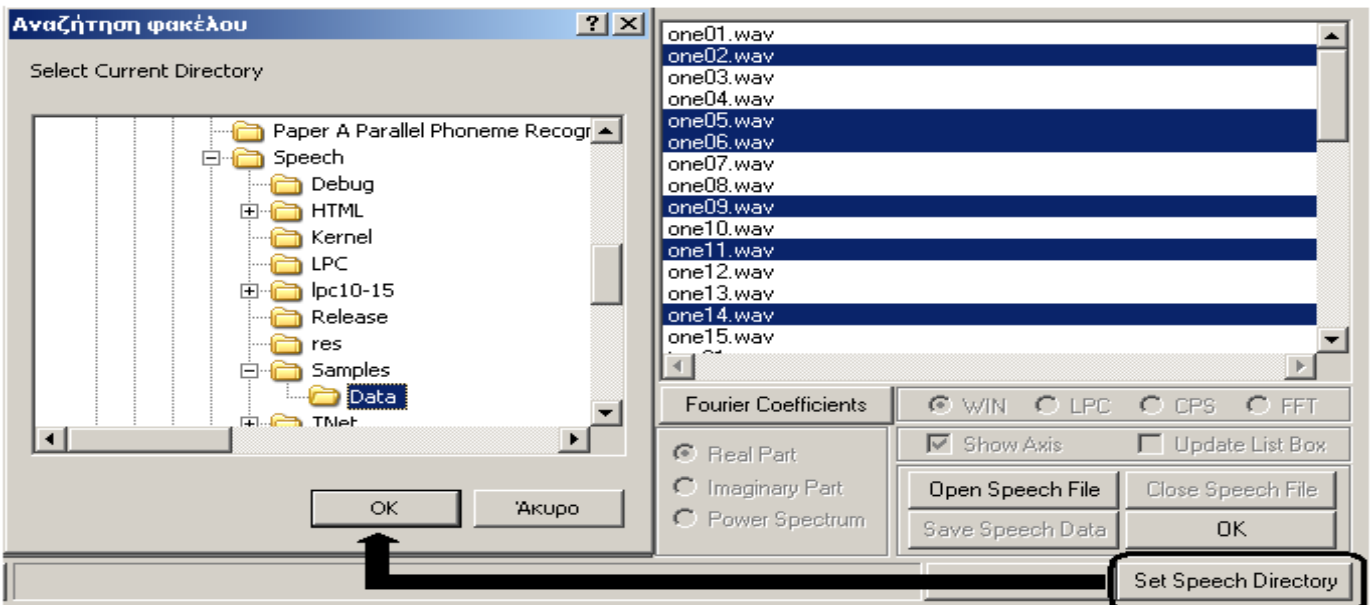


Fig. 4. The batch operation of the WinSPT application

In Figure 4, the "Set Speech Directory" push button has been used to specify the directory that contains the speech files to be processed and then, some of those files are used for further processing (the main list box of the application has been converted to a multi-selection list box to help the user select the desired files). Then, the "Batch Processing" push button shown in Figure 1 has to be pressed to perform the batch operation on the

selected files. A progress bar located to the right of this button shows the progress of the operation and it is a useful visual tool for time consuming operations. After the termination of the process, the user can save the results to an appropriately formatted binary data file with an extension of "SPH" by using the "Save Speech Data" push button.

#### IV. THE SPEECHTSET APPLICATION

SpeechTSet is not a part of WinSPT but a stand alone dialog based Windows utility that converts the SPH files produced by the WinSPT application to TRN training set files used by the Neural Workbench simulator. The main window of the SpeechTSet application is shown in Figure 5.

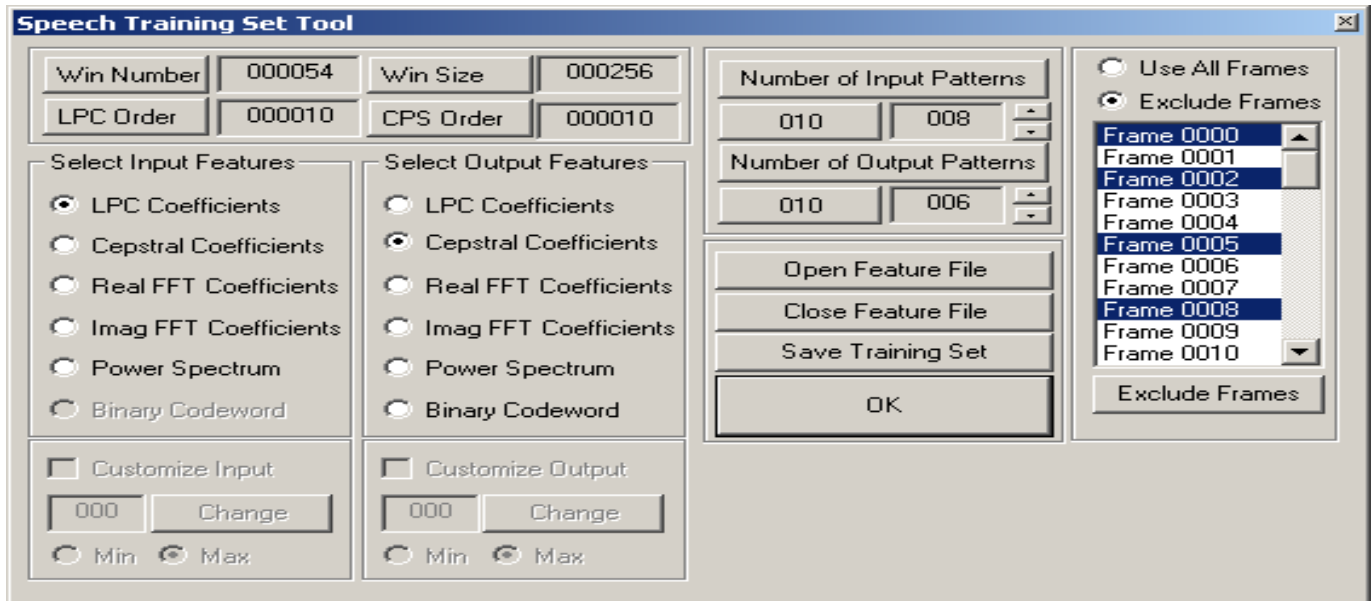


Fig. 5. The SpeechTSet main dialog box

In Figure 5, a SPH file with parameters  $W = 54$ ,  $N = 256$  and  $p = q = 10$  has been loaded into main memory. The user can select the type of the input and the output patterns of the training set to be created, by choosing one of the LPC coefficients, the Cepstral coefficients, as well as the real and the imaginary parts of the Fourier coefficients or the power spectrum values. If the user chooses the Fourier coefficients for the input and the output patterns, he can further customize his selection by choosing only the  $M$  minimum or maximum of them. Even though the default input and the output number is chosen automatically to be equal to the number of the selected features, the user can set a value less than this. In Figure 5, the number of the LPC coefficients is equal to 10 but the user has selected only the first 8 of them to be added to the training set file. Another very useful option provided by the program is the ability not to use the whole set of the available speech frames but to exclude some of them if necessary. To do this, the user has to select the undesired frames from the multi-selection list box and then, use the "Exclude Frames" push button to remove them from the frame list. After all these decisions and actions, the user can press the "Save Training Set" button to choose the training set file name, and finally, to create the training set.

The training set created in this way, is compatible with the Neural Workbench simulator; the contents of a typical training set created by the SpeechTSet application as they are shown from the Training Set Editor of Neural Workbench, are shown in Figure 6.

#### V. CONCLUSIONS

This paper presents a set of utilities used for the processing of speech data files; the main utility processes these files by applying a set of well known algorithms (pre-emphasis, filtering, frame blocking, Fourier, LPC and Cepstral analysis), while the second one converts the processed data produced in this way, to training set files compatible with Neural Workbench for further processing. There is no future work in this project; all the necessary operations have been implemented and the future research will be focused to the utilization of the processed data produced by them, to the various algorithms associated with the area of the speech technology.

| xxxxx    | Feature 01 | Feature 02 | Feature 03 | Feature 04 | Feat... |
|----------|------------|------------|------------|------------|---------|
| Input 01 |            |            |            |            |         |
| Input 02 |            |            |            |            |         |
| Input 03 |            |            |            |            |         |
| Input 04 |            |            |            |            |         |
| Input 05 |            |            |            |            |         |
| Input 06 |            |            |            |            |         |
| Input 07 |            |            |            |            |         |
| Input 08 |            |            |            |            |         |
| Input 09 |            |            |            |            |         |

| xxxxx     | Feature 01 | Feature 02 | Feature 03 | Feature 04 | Feat... |
|-----------|------------|------------|------------|------------|---------|
| Output 01 |            |            |            |            |         |
| Output 02 |            |            |            |            |         |
| Output 03 |            |            |            |            |         |
| Output 04 |            |            |            |            |         |
| Output 05 |            |            |            |            |         |
| Output 06 |            |            |            |            |         |
| Output 07 |            |            |            |            |         |
| Output 08 |            |            |            |            |         |
| Output 09 |            |            |            |            |         |

|          | Input 04 | Input 05  | Input 06 | Input 07  | Input 08 | Output 01 | Output 02 | Output 03 | Output 04 | Output 05 | Output 06 |
|----------|----------|-----------|----------|-----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Pat. 001 | 0.129586 | 0.017071  | 0.162285 | -0.108411 | 0.061780 | 0.523785  | 0.503201  | 0.209740  | 0.334300  | 0.128838  | 0.306837  |
| Pat. 002 | 0.215222 | -0.054827 | 0.117026 | -0.055569 | 0.059309 | 0.490572  | 0.435365  | 0.199202  | 0.390803  | 0.123780  | 0.288271  |
| Pat. 003 | 0.202332 | -0.026486 | 0.116243 | -0.049262 | 0.051374 | 0.468491  | 0.384057  | 0.202335  | 0.354498  | 0.155050  | 0.267349  |
| Pat. 004 | 0.139315 | 0.015145  | 0.032216 | -0.019491 | 0.031644 | 0.544505  | 0.369831  | 0.216611  | 0.317548  | 0.184223  | 0.203988  |
| Pat. 005 | 0.039719 | 0.069298  | 0.017407 | -0.022550 | 0.006524 | 0.614611  | 0.328794  | 0.222082  | 0.244896  | 0.187024  | 0.194175  |
| Pat. 006 | 0.017107 | 0.026238  | 0.014796 | 0.011853  | 0.008422 | 0.739539  | 0.366984  | 0.277114  | 0.246235  | 0.189326  | 0.190191  |

Fig. 6. Preview of the training set data from Neural Workbench

## REFERENCES

- [1] P.Boersma and D.Weenink (2006): PRAAT: Doing Phonetics by Computer, Software Application, Version 4.3.14, <http://www.praat.org/>.
- [2] SIL (Summer Institute of Linguistics) International, Speech Analyzer and WinCECIL, available from URL <http://www.sil.org/computing/speechtools/speechanalyzer.htm>
- [3] P.Wittenburg, H.Brugman, A.Russel, A.Klassmann and H.Sloetjes: ELAN - A Professional Framework for Multimodality Research, available from URL: <http://www.lat-mpi.eu/papers/papers-2006/elan-paper-final.pdf>
- [4] A. Margaris, E. Kotsialos, A. Styliadis, M. Roumeliotis, *Neural Workbench: An Object Oriented Neural Network Simulator*, Proceedings of International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI 2003), Acta Universitatis Apulensis (ISSN 1582-5329), Number 7/2004, Part B, pages 309-326, Alba Ioulia, Romania, 2003.
- [5] L. Rabiner, R. Schafer *Digital Processing of Speech Signals*, Pearson Education, 1978, ISBN 0-132-136031.
- [6] L. Rabiner, B-H Juang *Fundamentals of Speech Recognition*, Prentice Hall, 1993, ISBN 0-130-151572.
- [7] Furui S, *Digital Speech Processing, Synthesis and Recognition*, Marcel Dekker Inc, 1989.