Neural Network Program Package for Prosody Modeling

Jana TUČKOVÁ[1], *Jiří SANTARIUS*

[1]Dept. of Circuit Theory, Czech Technical University, Technická 2, 166 27 Prague 6, Czech Republic

tuckova@feld.cvut.cz, santarius@seznam.cz

Abstract. *This contribution describes the programme for one part of the automatic Text-to-Speech (TTS) synthesis. Some experiments (for example [14]) documented the considerable improvement of the naturalness of synthetic speech, but this approach requires completing the input feature values by hand. This completing takes a lot of time for big files. We need to improve the prosody by other approaches which use only automatically classified features (input parameters). The artificial neural network (ANN) approach is used for the modeling of prosody parameters. The program package contains all modules necessary for the text and speech signal pre-processing, neural network training, sensitivity analysis, result processing and a module for the creation of the input data protocol for Czech speech synthesizer ARTIC [1].*

Keywords

Speech synthesis, prosody modeling, artificial neural network, MATLAB.

1. Introduction

A probabilistic behavior of the speech signal is a motivation to use a statistical approach, for example, an artificial neural network (ANN) in the text-to-speech processing, especially for improving the quality of synthetic speech. The prosodic parameters play an important role in the full speech synthesis process. The quality (i.e. intelligibility and naturalness of the synthetic speech) is a very difficult task. The automatic system design for the preprocessing of the signal and text, training of the prosody by the ANN and prosody modeling are the goals of our research. We use a speech unit segmentation of a text for prosody modeling. The phonemes are the basic units in our neural network approach.

2. Speech Laboratory

Speech Laboratory is a complex system which was created as a user friendly application of the neural networks in prosody modeling of synthetic speech. The project consists of the tools necessary for utilizing neural networks in prosody modeling. Individual tools can be divided into three categories:

- **Pre-processing tools**: tools for data preparation, data creation and analysis.
- **Processing tools**: tools for working with neural nets, such as training, storage and analysis.
- **Post-processing tools**: tools for visualization, comparison and analysis of synthetic speech.

When artificial neural networks (ANN) are used as a prosody synthesizer, several things have to be done. The ANN is a simulation tool. Its use can be divided into two steps. In the first step, the ANN is created and trained on the input data. Prosodic parameters such as fundamental frequency (pitch period), phoneme duration and intensity have to be extracted from the acoustic **.wav* files. Extracted parameters create *target vectors* for the training process. In the second step, the ANN is used as a projection of input data extracted from the text into the outputs. These outputs directly represent prosodic parameters.

The following text describes how particular tools of the *Speech Laboratory* project can be used for Text-to-Prosody (TTP) synthesis. The following sections lead one through all the necessary steps needed for the successful use of neural nets in TTP synthesis.

2.1 PPL - Pitch Period Laboratory

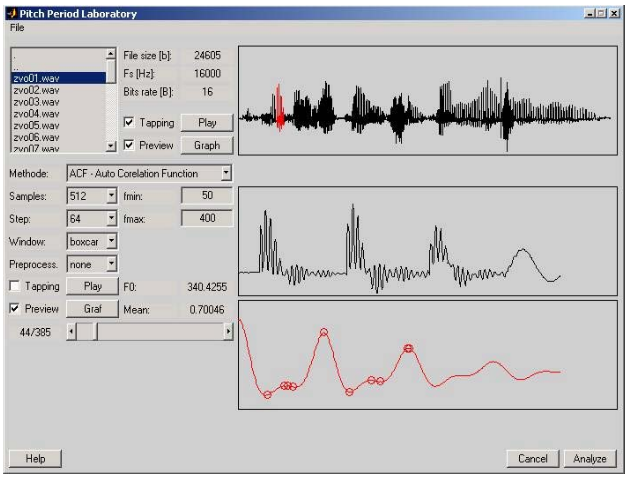
Let’s suppose, that we have the database, which consists of acoustic **.wav* files. The first step that has to be done is to detect the pitch period. A tool named *PPL* [5] is used for detecting this.

An analysis is done on segments with selectable length and overlap. Each segment may be multiplied by a selectable window function (Bartlett’s, Hamming’s, Blackman’s, …). Particular segments are then analyzed by one of two algorithms:

- FFT – Fast Fourier Transform,
- ACF – Autocorrelation Function.

The results are saved in a text file. An example of the pitch period detection by the tool *PPL* is shown in Fig. 1.

The detected pitch period creates the first part of the target vector needed for the training process of ANN. The second important part is created by a vector of durations. The tool named *LABEL* enables the detection of durations.



.
..
zvo01.wav
zvo02.wav
zvo03.wav
zvo04.wav
zvo05.wav
zvo06.wav
zvo07.wav

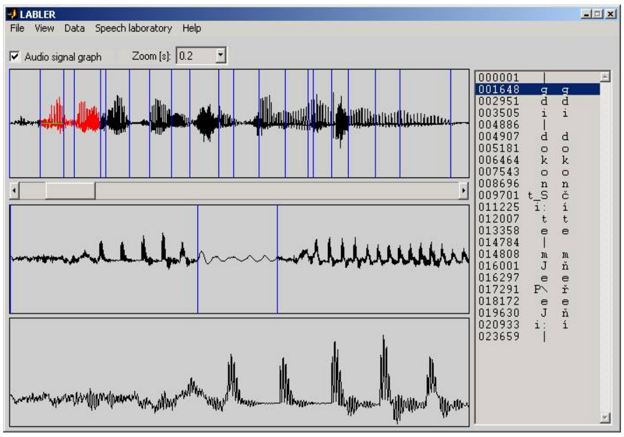
File size [b]: 24605
Fs [Hz]: 16000
Bits rate [B]: 16
☑ Tapping
☑ Preview
Methode: ACF - Auto Corelation Function
Samples: 512 fmin: 50
Step: 64 fmax: 400
Window: boxcar
Preprocess. none
□ Tapping F0: 340.4255
☑ Preview Mean: 0.70046
44/385

Fig. 1. The main window of the tool *PPL*. The tool *PPL* serves as a pitch period detector.

2.2 LABEL

The *LABEL* provides for the positioning of time marks into the acoustic continuance of *.wav file. Particular marks determine beginnings and endings of the phonemes. Marks are placed into the acoustic continuances manually. The distance between the marks determines the duration of the phoneme.

Fig. 2 shows the user interface of the *LABEL*. In order to be able to label the durations, the text representation of the acoustic form of exemplar database[1] has to be translated into the phonemes. This is performed by a tool named *DB*.



☑ Audio signal graph Zoom [s]: 0.2

```
000001   |
001648   g   g
002951   d   d
003505   i   i
004886   |
004907   d   d
005181   o   o
006464   k   k
007543   o   o
008696   n   n
009701 t_S   č
011225  i:   í
012007   t   t
013358   e   e
014784   |
014808   m   m
016001   J   ň
016297   e   e
017291  P\   ř
018172   e   e
019630   J   ň
020933  i:   í
023659   |
```

Fig. 2. The main window of the tool *LABEL*. The tool *LABEL* serves for the classification of phoneme's lengths.

2.3 DB – Data Base Maintain Tool

The *DB* manages the data of speech database. Particular vectors such as pitch period, intensity, duration, etc. are stored in separate files (*.f0, *.int, *.dur, etc.). The *DB* enables operations such as:

- Data conversion – conversion between several data standards, transcribes text into phonemes representation, etc.
- Data synchronization – synchronizes the data together (synchronization of a detected pitch period with phonemes).
- Data formatting – formats data according to the required rules. This feature enables one to generate special interface files to provide extra-project communication.
- Data classification – the *DB* supports some statistical classification methods.

The *PPL*, *LABEL* and *DB* support the creation of target vector that is essential for the process of the supervised ANN's training. The input vector is also very important. The IVL creates the input vector.

2.4 IVL – Input Vector Laboratory

The *IVL* generates input vector parameters. The input text is translated into phonemes [9] and proceeds according to the mask.[2] The key which determines positions and types of detected text parameters is called the mask. Output is represented by a data matrix which consists of particular input vectors.

Detectable parameters can be divided into the following three categories:

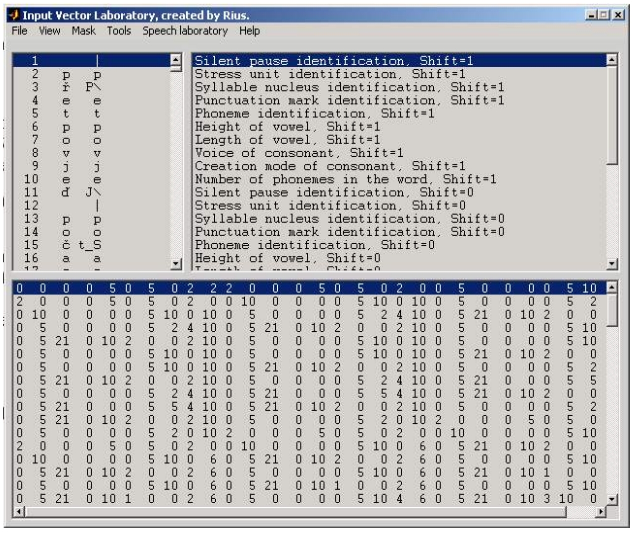
- Linguistic parameters – detection of features such as vocals, consonants, accents, prepositions, etc.
- Statistical parameters – detection of features such as the position in the sentence and in the word, the number of sentences, the number of words, the number of phonemes, etc.
- Informative – additional (optional) features for better orientation in a data vector.

The linguistic parameters are created only by properties of the Czech language which can have an important influence on prosody according to our previous experience and expert knowledge [2], [3], [14]. We will use a text and its speech signal for the ANN training. Target values are extracted from the speech signal for prosody modeling. The text will be available for the ANN input data for real prosody modeling. We cannot completely use all information extracted from a natural speech signal in automatic

[1] It is a database, which contains natural prosody labels of human speech.

[2] The mask represents the key, which encodes a type and an order of extracted text parameters.

input data creation. For example, the so-called prominence creation[3] cannot be differentiated automatically.



```
 1       |
 2   p   p
 3   ř   P\
 4   e   e
 5   t   t
 6   p   p
 7   o   o
 8   v   v
 9   j   j
10   e   e
11   ď  J\
12       |
13   p   p
14   o   o
15   č t_S
16   a   a
```

```
Silent pause identification, Shift=1
Stress unit identification, Shift=1
Syllable nucleus identification, Shift=1
Punctuation mark identification, Shift=1
Phoneme identification, Shift=1
Height of vowel, Shift=1
Length of vowel, Shift=1
Voice of consonant, Shift=1
Creation mode of consonant, Shift=1
Number of phonemes in the word, Shift=1
Silent pause identification, Shift=0
Stress unit identification, Shift=0
Syllable nucleus identification, Shift=0
Punctuation mark identification, Shift=0
Phoneme identification, Shift=0
Height of vowel, Shift=0
```

```
0  0  0  0  5 0  5  0 2  2 2  0  0  0  5 0  5  0 2  0 0  5  0  0  0 0  5 10
2  0  0  0  5 0  5  0 2  0 0 10  0  0  0 0  5 10 0 10 0  5  0  0  0 0  5  2
0 10  0  0  0 0  5 10 0 10 0  5  0  0  0 0  5  2 4 10 0  5 21  0 10 2  0  0
0  5  0  0  0 0  5  2 4 10 0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5 10
0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5 10 0 10 0  5  0  0  0 0  5 10
0  5  0  0  0 0  5 10 0 10 0  5  0  0  0 0  5 10 0 10 0  5 21  0 10 2  0  0
0  5  0  0  0 0  5 10 0 10 0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5  2
0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5  2 4 10 0  5 21  0  0 0  5  5
0  5  0  0  0 0  5  2 4 10 0  5 21  0  0 0  5  5 4 10 0  5 21  0 10 2  0  0
0  5 21  0  0 0  5  5 4 10 0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5  2
0  5 21  0 10 2  0  0 2 10 0  5  0  0  0 0  5  2 0 10 2  0  0  0  5 0  5  0
0  5  0  0  0 0  5  2 0 10 2  0  0  0  5 0  5  0 2  0 0 10  0  0  0 0  5 10
2  0  0  0  5 0  5  0 2  0 0 10  0  0  0 0  5 10 0  6 0  5 21  0 10 2  0  0
0 10  0  0  0 0  5 10 0  6 0  5 21  0 10 2  0  0 2  6 0  5  0  0  0 0  5 10
0  5 21  0 10 2  0  0 2  6 0  5  0  0  0 0  5 10 0  6 0  5 21  0 10 1  0  0
0  5  0  0  0 0  5 10 0  6 0  5 21  0 10 1  0  0 2  6 0  5  0  0  0 0  5 10
0  5 21  0 10 1  0  0 2  6 0  5  0  0  0 0  5 10 4  6 0  5 21  0 10 3 10  0
```

Fig. 3. User interface of the tool *IVL*.

The influence of the phonemes co-articulation is also very important from the point of view of prosody modeling and, therefore, we will use a moving window over several phonemes in the training process. We will try to ensure a suitable relation of numerical values among the different type of representatives of the parameters. Similar properties are assigned by neighboring numerical values [2], [3], [14].

The *IVL* is programmed in such a way that numbers and types of detectable parameters could be easily extended by new ones. An example of an *IVL* window is shown in Fig. 3.

The *PPL*, *Label*, *DB* and *IVL* give full support for data-creation that is important for the training process of ANN.

2.5 NNL – Neural Net Laboratory

The *NNL* is a tool that supports a graphical user interface for operations on ANNs. *NNL* covers operations such as creation, training, storage and analysis of ANNs. The tool supports a wide range of training algorithms. The fast error back-propagation algorithm (with a moment and adaptive learning rate), Levenberg-Marquardt algorithm, and the sigmoidal and linear activation function are implemented in the first experiments [13]. It is possible to add other algorithms and activation functions in the future.

The *NNL* supports a wide range of analysis algorithms. These algorithms are used for the optimization of neural network topology in order to improve the generalization ability of ANN [6], [7]. Currently the following algorithms are implemented.

- **Operation point analysis** – this algorithm classifies the position of the neuron's operation point. The classification is done on a base of mean value and standard deviation of neuron's output signal.

$$\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y[n] \qquad y_D = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(\bar{y} - y[n]\right)^2} \; . \tag{1}$$

- **Energy flow analysis** – this type of analysis detects the energy of neurons signal. The definition of neuron energy is given by equation (2).

$$E = \sum_{n=1}^{N}\left(\bar{y} - y[n]\right)^2 \; . \tag{2}$$

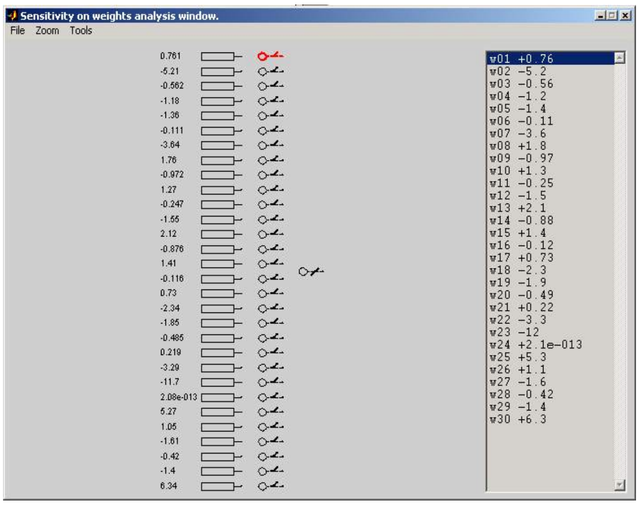
- **Sensitivity analysis** – this analysis uses sensitivities. According to the type of sensitivity used it will not only indicate the importance of input, weight, bias, but also the useless neuron could be detected. The definition of sensitivity [15] is given by equation (3).

$$S\left(y, x_i\right) = \frac{\partial y}{\partial x_i} \, . \tag{3}$$

An example of the ANN analysis window is shown in Fig. 4. Analyses are used for a detection of inappropriately trained neurons. Detected neurons are then retrained or pruned.

The *NNL* supports variety types of visualization plots. The *NNL* shows not only a network structure, but also plots graphs of signal continuances in various positions of ANN.

The above-mentioned tools enable a creation of prosodic database and ANN for the modeling of synthetic prosody. By using these tools, we are able to create a so-called TTP synthesizer. Unfortunately, the generated prosody is represented only by a set of vectors, which may be displayed on a graph. The conversion of the data to acoustic form is done by a tool named *Synth*.



```
w01 +0.76
w02 -5.2
w03 -0.56
w04 -1.2
w05 -1.4
w06 -0.11
w07 -3.6
w08 +1.8
w09 -0.97
w10 +1.3
w11 -0.25
w12 -1.5
w13 +2.1
w14 -0.88
w15 +1.4
w16 -0.12
w17 +0.73
w18 -2.3
w19 -1.9
w20 -0.49
w21 +0.22
w22 -3.3
w23 -12
w24 +2.1e-013
w25 +5.3
w26 +1.1
w27 -1.6
w28 -0.42
w29 -1.4
w30 +6.3
```

Fig. 4. ANN's weights sensitivity analysis of neuron n (1,1).

[3] The prominences demonstrate the different weights of a stress in a sentence.

2.6 Synth – Synthesizer

This tool enables one to change prosodic data into acoustic form. *Synth* supports two types of synthesis:

- **Wave representation** – *Synth* generates only waveforms [7] of a pitch period with respect to phonemes duration and intensity. Waves are generated according to equation (4)

$$y = I \sin\left(2\pi f(\tau)t\right) \tag{4}$$

where I is an intensity, $F_0\ (\tau\ )$ is a fundamental frequency contour and t is time.

- **Speech representation** – the *Synth* has an ability to send data via the Internet to the Czech Text – to - Speech (TTS) system ARTIC [1], [4] from the University of West Bohemia in Pilsen. A special file interface was created to enable an online access. This type of synthesis offers real synthetic speech with controlled prosody.

The *Synth* is more sophisticated. Apart from generating prosody by ANNs, *Synth* supports more types of prosody generation, such as monotonous, manual or Fujisaki's models. An example of *Synth* is shown in Fig. 5.

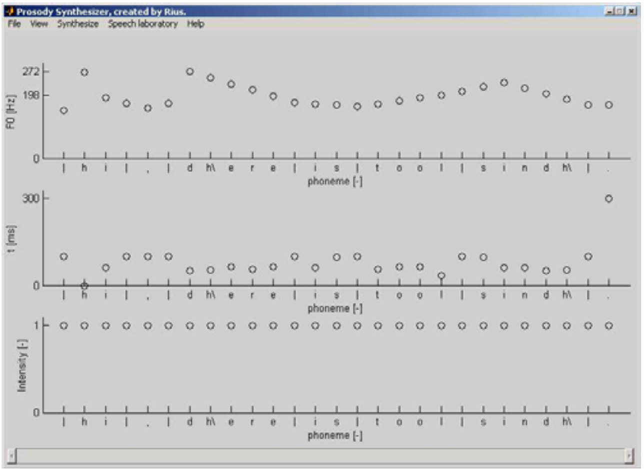

Prosody Synthesizer, created by Rius.
File
View
Synthesize
Speech laboratory
Help
F0 [Hz]
272
198
0
| h i | . | d h\ e r e | i s | t o o l | s i n d h\ | .
phoneme [-]
t [ms]
300
0
| h i | . | d h\ e r e | i s | t o o l | s i n d h\ | .
phoneme [-]
Intensity [-]
1
0
| h i | . | d h\ e r e | i s | t o o l | s i n d h\ | .
phoneme [-]

Fig. 5. An example of a synthesis by the tool *Synth*.

3. Software

This original software system in MATLAB, V6.5.0.180913a, Release 13 and NN-Toolbox, ver.4 for LINUX and Windows 98, 2000 and XP is presently under construction; this system will serve for automation of the data base creation, neural network training and graphical results processing [13].

4. Conclusion

The unsystematic errors during the hand labeling process caused by individual human abilities and the physical and mental state of the approbator can be eliminated by the automatic approach. It is possible to mention that a full automatic determination of the beginning and ending of speech units (labeling) is controversial. During the training of ANN, the automatically labeled signal needed for the determination of target values have to be checked and corrected by a person, but in the real synthesizer, utilization it is not possible. Therefore, the resulting signal can have some audible anomalies.

More details about the feature selection and ANN training were presented in [2], [4], [14].

We can state that ANN training can be successfully used for prosody modeling. The described software package enables an easier application of ANN training, and the package uses MATLAB with Neural Network Toolbox.

Acknowledgements

The research described in this paper was financially supported by the Czech Grant Agency under grant No. 102/ 02/0124.

References

[1] MATOUŠEK, J., PSUTKA, J., KRUTA, J. Design of Speech Corpus for Text-to-Speech Synthesis. In *Proc. of. Eurospeech2001*. Denmark (Ålborg), 2001, vol. 3, pp. 2047–2050.

[2] TUČKOVÁ, J., ŠEBESTA, V. Influence of Language Parameters Selection on the Coarticulation of the Phonemes for Prosody Training in TTS by Neural Networks. In *Proc. of the Int. Conf.* on *Artificial Neural Nets and Genetic Algorithms (ICANNGA 2003).* France (Roanne), 2003, pp.85-90, ISBN: 3-211-00743-1 ,Springer-Verlag Wien-New York.

[3] TUČKOVÁ, J. *Úvod do teorie a aplikací umělých neuronových sítí.* Skripta FEL ČVUT v Praze, vydavatelství ČVUT, 2003, ISBN 80-01-02800-3.

[4] TUČKOVÁ, J., MATOUŠEK, J. Czech Language Features Selection and Prosody Modelling for Text-to-Speech Synthesis. In *ECMS 2003*. Czech Republic (Liberec), 2003, vol. 1, p. 98-102. ISBN 80-7083-708-X.

[5] SANTARIUS, J. PPL - Nástroj pro detekci základního hlasivkového tónu. In *Moderní směry výuky elektrotechniky a elektroniky*. Brno, 2002, vol. 1, p. 70-73. ISBN 80-214-2190-8.

[6] SANTARIUS, J. Statistical Methods for Optimalization of Neural Nets. In *Proc. of Workshop 2002.* Prague: CTU, 2002, vol. A, p. 486-487. ISBN 80-01-02511-X.

[7] SANTARIUS, J. Special Algorithms Used in Prosody Modelling. In *Proc. of the Polish-Hungarian-Czech Workshop on Circuit Theory, Signal Processing, and Applications.* Prague, 2003, vol. 1, p. 43-48. ISBN 80-01-02825-9.

[8] SANTARIUS, J., TIHELKA, J. Prosody Modelling of Synthetic Speech. In *ECMS 2003*. Czech Republic (Liberec), 2003, vol. 1, p. 89-92. ISBN 80-7083-708-X.

[9] SLEZÁK, J. Automatic Phonetic Transcription. In *ECMS 2003.* Czech Republic (Liberec), 2003, vol. 1, p. 93-97. ISBN 80-7083-708-X.

[10] PSUTKA, J. *Speech communication with a computer* (in Czech- Komunikace s počítačem mluvenou řečí). Academia, Praha, 1995, ISBN 80-200-0203-0.

[11] TRABER, Ch. *The implementation of the Text-to-Speech System for German*. PhD dissertation, ETH Zurich, Switzerland, 1995.

[12] PALKOVA, Z *Phonetics and phonologics of the Czech language* (in Czech: *Fonetika a fonologie češtiny*). Univerzita Karlova-Praha, 1994, ISBN: 80-7066-843-1.

[13] DEMUTH, H., BEALE, M. *Neural Network Toolbox. For Use with MATLAB*. User's Guide, ver.4, The MathWorks, Inc., MA 01760-2098

[14] TUCKOVA,J., SEBESTA,V. Data Mining Approach for Prosody Modelling by ANN in Text-to-Speech Synthesis.In *Proc. of the Int. Conf. IAESTED AIA2001*. Spain (Marbella), 2001, pp. 161-166, ISBN:0-88986-301-6.

[15] GÉHER, K. *Theory of Network Tolerances*. Akadémiai Kiadó, Budapest, 1971.

About the Authors...

Jana TUČKOVÁ received an Ing (M.SC.) degree in 1974, a CSc. (PhD) degree in 1981 at the Faculty of Electrical Engineering of the Czech Technical University in Prague and PGS in 1993 at EPFL, Lausanne, Switzerland. At present she is an associate professor in the Department of Circuit Theory at CTU in Prague. Her research is concentrated in some domains of the neural network applications, esp. in speech analysis and prosody modeling.

Jiří SANTARIUS received a Ing (M.SC.) degree in 2000 at the Faculty of Electrical Engineering, Czech Technical University, Prague. Since 2000 he is a PhD. student at the same university. He specializes in the application of artificial neural networks in the prosody modeling for synthetic speech.