Estimation of Non-Stationary Noise for Audio Enhancement by Means of Recurrent Neural Networks

RAFAL KROLIKOWSKI
Sound & Vision Engineering Department
Technical University of Gdansk
Narutowicza 11/12; 80-952 Gdansk
POLAND

Abstract: - In the paper, some issues related to the problem of noise reduction in audio signals are addressed. Since under real conditions this kind of distortions is time varying, the noise is considered as non-stationary throughout the paper in which new methods of estimation of the distortion are introduced. On account of the non-stationary character of the noise, which can be described in terms of time sequences of some states, recurrent neural networks (RNNs) are employed as estimators of these sequences. This new approach is described in some details in the paper, and the supporting results of experiments are briefly described therein.

Key-Words: - audio enhancement, recurrent neural networks

1 Introduction

Noise and other distortions very often are present in various audio signals. Such a situation concerns old recordings, speech communication systems in a noisy environment (e.g. hand-free sets in car/plane cockpits), telecommunication systems, including teleconferencing, and many others. These distortions can be introduced in the process of sound acquisition (communication devices often work in poor acoustic conditions), may occur during a transmission, and also can be also caused by imperfections of audio media, especially old media. Therefore there is a need for robust and efficient methods of parasite noise reduction. The objectives of these methods are to improve the quality of audio and/or to enhance some information imparted in signals, what is very often related to the improvement of speech intelligibility.

Owing to the fact that noise shows a random character, it is practically impossible to achieve the exact clean signal, which makes the quality of audio deteriorated. Thus, this quality depends considerably on the closeness in estimation of distortions, and moreover - noise reduction itself can be considered as an estimation of the clean signal. Hence, numerous approaches & methods of audio enhancement have been proposed over the last dozen of years. The scope of the research encompassed mainly: Wiener- and Kalman adaptive filtration [21] [7], Bayesian estimation based on Hidden Markov Models (HMMs) [6], and spectral subtraction [17]. Besides, one can find perceptual methods, proposed by the author, which employ some properties of the human auditory system [2] [3]. However, the common trait of the afore-cited approaches is the need of noise estimation. The problem becomes more complex in the case of non-stationary one, i.e. when statistical characteristics of the noise vary in time. In such a case, it is additionally required to track the altering statistics. It should be mentioned that HMMs [13] [14] play a substantial role in modeling of non-stationary signals.

In turn, one can notice that the dynamically developing domain of neural processing can offer such capabilities as: non-linear approximation [23] (for estimation purposes), dealing with uncertain & incomplete information [4], modeling of time series [5], to name a few. Especially the latter feature is peculiar to RNNs, which structure equipped with time delay units enables efficient processing of time series, and thereby such networks are valuable for estimation of non-stationary noise. Moreover, if the delays refer to a time unit, one can find the analogy to HMMs of the first order. On account of the above, the proposed noise estimation was based on RNNs.

In the paper, for brevity of the description of audio enhancement the proposed neural estimation is based on spectral subtraction (SS). Nevertheless on the other hand, SS methods turn out to be both robust and simple, and are efficient in restoration of old recordings [20]. The presented approach can be extended to more advanced methods, as e.g. the one proposed by the author [2] [9].

On account of the nature the SS method and non-stationary noise there is a need to store spectra of the noise or other distributions of statistics. However in such a case, it becomes a problem to select correctly the relevant distribution (spectra). The author's proposal is to use soft-computing methods for this task in general, and RNNs in particular. In other words, sequences of noise spectra are assembled in a codebook, whereas the role of an access key is played by relevant vectors (codevectors) of parameters describing the noise. Entries into the codebook can be referred to as states, and hence - the selection of the relevant spectra can be considered as an estimation of state sequence representing the respective noise-stationary noise. In the paper, the author's proposal of the state sequence estimation driven by RNNs is presented.

As far as the application of RNNs is considered, in this work the focus is put on a general RNN proposed by Elman [5], despite that there is a number of other recurrent architectures [8]. An important class among RNNs constitute so called NARX networks (Nonlinear Auto Regressive with eXogenous inputs) [10] [16]. They are reported to be robust, more straightforward to converge during the training than general RNNs, and to be equivalent to a certain extent to general RNNs [18]. However due to their structure (only a single feedback loop between the output and input with a number delay taps in the loop) they seem to be unsuitable for the purposes of noise estimation. As regards a training algorithm - despite a number of various RNN training methods, including second-order methods [8], Conjugate Gradient Learning [1] and even genetic algorithms [11], the standard technique based on the Wiliams & Zipser approach [22] was used.

The RNN-based estimator was implemented and a number experiments were carried out. Some of the results are additionally discussed in the paper.

2 Audio Enhancement - Problem Statement

Generally, acoustic noise is considered as additive, although in telecommunication channels the distortions can be of the multiplicative type. Therefore the clean signal $x(m)$ in the presence of the noise $n(m)$ is observed as the noisy signal $y(m)$, and the relationship is modeled by the following equation:

$$y(m) = x(m) + n(m) \tag{1}$$

The counterpart of the above formula in the spectral domain is expressed as follows:

$$Y(\mathrm{j}\omega) = X(\mathrm{j}\omega) + N(\mathrm{j}\omega), \tag{2}$$

where $Y(\mathrm{j}\omega)$, $X(\mathrm{j}\omega)$, $N(\mathrm{j}\omega)$ are spectral representations of the signals: $y(m)$, $x(m)$, $n(m)$, respectively.

Since the exact noise $n(m)$ (or $N(\mathrm{j}\omega)$) is practically not known, only an approximated form $\hat{N}(\mathrm{j}\omega)$ can be estimated. Thus the clean signal also can be only estimated, what yields the following relationship:

$$\hat{X}(\mathrm{j}\omega) = Y(\mathrm{j}\omega) - \hat{N}(\mathrm{j}\omega), \tag{3}$$

or can be rewritten in more general form - the non-linear spectral subtraction as below [19]:

$$\left|\hat{X}(\mathrm{j}\omega)\right|^b = \left|Y(\mathrm{j}\omega)\right|^b - \alpha(\mathrm{j}\omega) \cdot \left|\hat{N}(\mathrm{j}\omega)\right|^b, \tag{4}$$

where the coefficients b and α control the subtraction process.

As can be derived from the above formula (4), the closeness of the clean signal estimation depends on the estimate $\hat{N}(\mathrm{j}\omega)$. Therefore the problem of noise reduction can be considered in terms of the closeness of non-stationary noise estimation: $N(t,\mathrm{j}\omega) \Rightarrow \hat{N}(t,\mathrm{j}\omega)$.

3 Estimation of Non-Stationary Noise

Since the noise estimation is based on a codebook solution driven by RNNs, the estimator works in the following modes:

- Building-up the Codebook Mode

  While building-up a codebook, some noise patterns $\tilde{n}(m)$ (reference noise) which are correlated to the noise $n(m)$ are expected to be available. In result, to the codebook two sequences of the length T are added: $\tilde{\boldsymbol{N}}(t,\mathrm{j}\omega) = \left\{\tilde{\boldsymbol{N}}'(t-\tau,\mathrm{j}\omega)\right\}_{\tau=0\ldots T-1}$ (representing the reference) and $\boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t) = \left\{\boldsymbol{v}^{\tilde{\boldsymbol{n}}}(t-\tau)\right\}_{\tau=0\ldots T-1}$ which stands for the sequence of codevectors. This process can be denotes by the following relationship:

$$\tilde{n}(m) \Rightarrow \boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t), \tilde{\boldsymbol{N}}(t,\mathrm{j}\omega) \tag{5}$$

In the codebook the *i*-th entry is described by the pair $\{\tilde{N}_i(t,\mathrm{j}\omega), \boldsymbol{v}_i^{\tilde{\boldsymbol{n}}}(t)\}$, where $\boldsymbol{v}_i^{\tilde{\boldsymbol{n}}}(t) \equiv \boldsymbol{v}^{\tilde{\boldsymbol{n}}}(t-\tau)$ and $\tilde{N}_i(t,\mathrm{j}\omega) \equiv \tilde{\boldsymbol{N}}'(t-\tau,\mathrm{j}\omega)$.

- RNN Training Mode

  The task of RNNs is to correctly estimate the sequence $\boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t)$, thus this operation can be described by the relationship as below:

$$\boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t) \overset{RNN}{\Rightarrow} \hat{\boldsymbol{V}}^{\tilde{\boldsymbol{n}}}(t) \qquad (6)$$

- Run Mode

  Basing on the noisy observation $y(m)$ the respective sequence of the codevectors $\boldsymbol{V}^{y}(t)$ is computed and fed to a RNN in order to obtain the state sequence estimate $\hat{\boldsymbol{V}}^{\tilde{\boldsymbol{n}}}(t)$. This estimate compared to all codevectors $\boldsymbol{v}_i^{\tilde{\boldsymbol{n}}}(t)$ serves to generate the sequence $\boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t)$, and in consequence - the respective estimate $\hat{\boldsymbol{N}}(t,\mathrm{j}\omega)$ of the non-stationary noise.

$$y(m) \Rightarrow \boldsymbol{V}^{y}(t) \overset{RNN}{\Rightarrow} \hat{\boldsymbol{V}}^{\tilde{\boldsymbol{n}}}(t) \overset{codebook}{\Rightarrow} \boldsymbol{V}^{\tilde{\boldsymbol{n}}}(t), \hat{\boldsymbol{N}}(t,\mathrm{j}\omega) \qquad (7)$$

3.1 Codevectors Computation

In the given moment of time *t*, the elements of the codevector $\boldsymbol{v}^{\tilde{\boldsymbol{n}}}(t) = \left[v_1^{\tilde{\boldsymbol{n}}}(t)\ \ldots\ v_b^{\tilde{\boldsymbol{n}}}(t)\ \ldots\ v_B^{\tilde{\boldsymbol{n}}}(t)\right]^T$ are computed in the spectral domain in some subbands. Since in acoustics the octave bands play substantial role, the relevant parameters are computed in these bands. So that the elements of the codevector represent quantitatively & distinctively the noisy character of the reference signal, three kinds of parameters are introduced. The first two of them turned out to be very robust in perceptual coding schemes, including the MPEG standard [12], namely: the Spectral Flatness Measure [15] and the Unpredictability Measure [12]. The third kind of parameters regards the first statistical moments, i.e. the expected value and variance.

As regards the codevector $\boldsymbol{v}^{y}(t)$, its elements are computed in the same was those of $\boldsymbol{v}^{\tilde{\boldsymbol{n}}}(t)$.

3.1.1 Employment of *Spectral Flatness Measure*

The *SFM* parameter is defined as the ratio of the geometric to the arithmetic mean of the signal's power spectrum [15], and is expressed in dB. In the *b*-th subband, the parameter can be redefined as follows (time indices are dropped for brevity):

$$SFM_b = 10\log_{10}\frac{\left[\prod_{i=\inf(b)}^{\sup(b)} S_i\right]^{\frac{1}{N_b}}}{\frac{1}{N_b}\sum_{i=\inf(b)}^{\sup(b)} S_i} \qquad (8)$$

where $\sup(b)$ and $\inf(b)$ denote the spectral bin of the lowest and the highest frequency in the subband consisting of N_b components, whereas S_i stands for the magnitude of the *i*-th bin.

3.1.2 Employment of *Unpredictability Measure*

Introducing denotations of the spectral magnitude prediction $\hat{r}_i(t)$ and the phase prediction $\hat{\phi}_i(t)$ of the *i*-th spectral component on the basis of their last two real values as below:

$$\begin{cases}\hat{r}_i(t) = r_i(t-1) + \left[r_i(t-1) - r_i(t-2)\right] \\ \hat{\phi}_i(t) = \phi_i(t-1) + \left[\phi_i(t-1) - \phi_i(t-2)\right]\end{cases}, \qquad (9)$$

the unpredictability measure $c_i(t)$ is defined as the Euclidean distance between the real values of $r_i(t)$, $\phi_i(t)$ and the predicted ones of $\hat{r}_i(t)$, $\hat{\phi}_i(t)$ according to the formula [12]:

$$c_i(t) = \frac{\left\|\left(\hat{r}_i(t), \hat{\phi}_i(t)\right) - \left(r_i(t), \phi_i(t)\right)\right\|}{r_i(t) + \left|\hat{r}_i(t)\right|}, \qquad (10)$$

and the unpredictability measure for the *b*-th subband is computed as follows:

$$C_b = \frac{1}{N_b}\sum_{i=\inf(b)}^{\sup(b)} c_i\ , \qquad (11)$$

where the denotations are as in the formula (8).

3.1.3 Employment of statistical moments

Assuming the denotations are as in the formula (8), in the *b*-th subband a pair of the expected value and the variance $\left(\mu_b, \sigma_b^2\right)$ is calculated as below:

$$\mu_b = \frac{1}{N_b}\sum_{i=\inf(b)}^{\sup(b)} S_i, \sigma_b^2 = \frac{1}{N_b}\sum_{i=\inf(b)}^{\sup(b)} \left[S_i - \mu_b\right]^2 \qquad (12)$$

3.2 Application of RNNs

As was mentioned, RNNs was employed as estimators of the state sequence $\hat{V}^{\tilde{n}}(t)$ given the sequence of codevectors $V^y(t)$ of the observed noisy signal $y(m)$. In turn, the training of RNNs is auto-associative, as can be concluded from the relationship (6).

3.2.1 Architecture of RNNs

In a given moment of time, a codevector $\boldsymbol{v}_i(t)$ consists of B elements given B octave subbands. Thus the input layer consists of B+1 neurons (including the dummy one). In turn, owing to the auto-association, the output layer is composed of B units. Only the size of the contex- and the hidden layer is variable and is set during experiments. The structure of the employed RNN is presented in Fig. 1, where $N = K = B$.

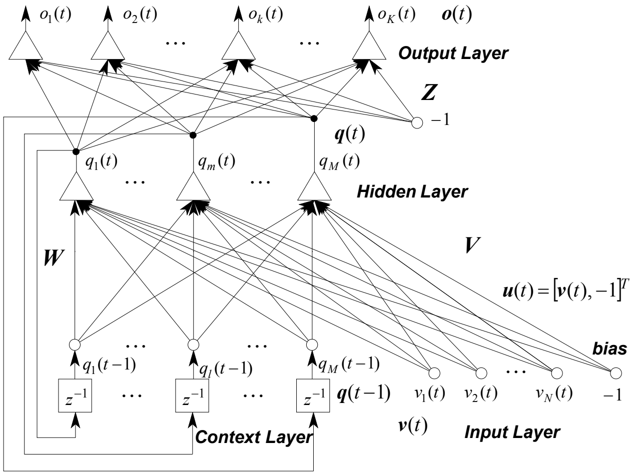

Fig. 1. Architecture of the employed RNN

As regards the neuron's activation function (NAF), continuous sigmoidal ones are recommended. However, in this case of estimation, the problem is that the values of codevector parameters exceed considerably the acceptable range of these functions, what makes the auto-association impossible. Fortunately, one can submit three solutions of this problem:

- all NAFs are linear
- only the output neurons are linear
- NAFs remain non-linear, but the parameters are normalized

The first solution can considerably deteriorate the convergence process while training. In turn, the third feasibility can be impractical, since in the case of the proposed codevector parameters the normalization could lead to very small values, and thereby - make the processing inefficient. Thus, the reasonable solution is to put NAFs only of the output units as linear functions.

3.2.2 Training of RNNs

The RNN training assumes the mean-square error as the error measure in the t-th moment of discrete time between the desired response $d_k(t)$ and the current output $o_k(t)$ of the k-th output neuron according to the following formulae:

$$E^{(t)} = \frac{1}{2}\sum_{k=1}^{K}\left[e_k^{(t)}\right]^2 \; ; \quad e_k^{(t)} = d_k^{(t)} - o_k^{(t)} \qquad (13)$$

Additionally, the input vector $\boldsymbol{u}(t)$ is the following augmentation of the codevector $\boldsymbol{v}(t)$:

$$\boldsymbol{u}(t) = \left[v_1(t) \; ... \; v_b(t) \; ... \; v_B(t) \; -1\right]^T \qquad (14)$$

Assuming the learning rate η, the equations for updating the weight values are based on the Wiliams & Zipser approach [22], and for the RNN as in Fig. 1 they assume:

$$\Delta Z_{km}^{(t)} = \eta e_k^{(t)} \frac{\partial f_k\left(\xi_k^{(t)}\right)}{\partial \xi_k^{(t)}} q_m^{(t)} = \eta e_k^{(t)} q_m^{(t)} \qquad (15)$$

$$\begin{aligned}\Delta W_{ml}^{(t)} &= \eta \sum_{k=1}^{K} e_k^{(t)} \frac{\partial f_k\left(\xi_k^{(t)}\right)}{\partial \xi_k^{(t)}} \sum_{j=1}^{M} Z_{kj}^{(t)} \Psi_{j,ml}^{(t)} = \\ &= \eta \sum_{k=1}^{K} e_k^{(t)} \sum_{j=1}^{M} Z_{kj}^{(t)} \Psi_{j,ml}^{(t)}\end{aligned} \; , \qquad (16)$$

$$\begin{aligned}\Delta V_{mn}^{(t)} &= \eta \sum_{k=1}^{K} e_k^{(t)} \frac{\partial f_k\left(\xi_k^{(t)}\right)}{\partial \xi_k^{(t)}} \sum_{j=1}^{M} Z_{kj}^{(t)} \Lambda_{j,mn}^{(t)} = \\ &= \eta \sum_{k=1}^{K} e_k^{(t)} \sum_{j=1}^{M} Z_{kj}^{(t)} \Lambda_{j,mn}^{(t)}\end{aligned} \; , \qquad (17)$$

where f_k denotes the NAF for the k-th output neuron (the NAF assumes the linear form), $q_m(t)$ is the output of the m-th hidden neural unit (often referred to as a state), and the auxiliary terms $\Psi_{j,ml}^{(t)}$ & $\Lambda_{j,mn}^{(t)}$ are defined as follows:

$$\Psi_{j,ml}^{(t)} = \frac{\partial f_j\left(\vartheta_j^{(t)}\right)}{\partial \vartheta_j^{(t)}}\left[\sum_{i=1}^{M} W_{ji}^{(t)} \Psi_{i,ml}^{(t-1)} + \delta_{jm} q_l^{(t-1)}\right] \quad (18)$$

$$\Lambda_{j,mn}^{(t)} = \frac{\partial f_j\left(\vartheta_j^{(t)}\right)}{\partial \vartheta_j^{(t)}}\left[\sum_{i=1}^{M} W_{ji}^{(t)} \Lambda_{j,mn}^{(t-1)} + \delta_{jm} u_n^{(t)}\right], \quad (19)$$

where δ_{jm} denotes the Kronecker's delta, f_j stands for the NAF for the j-th hidden neuron and the weighted sums $\vartheta_m^{(t)}$ & $\xi_k^{(t)}$ are computed as below:

$$\vartheta_m^{(t)} = \sum_{j=1}^{M} W_{mj}^{(t)} y_j^{(t-1)} + \sum_{j=1}^{N1} V_{mj}^{(t)} u_j^{(t)}, \quad (20)$$

$$\xi_k^{(t)} = \sum_{j=1}^{M} Z_{kj}^{(t)} q_j^{(t)} \quad (21)$$

4 Experiments

The developed neural estimator for non-stationary noise was implemented and tested. The verification experiments were carried out using old audio recordings (1924, wax media). For testing some excerpts of the length approx. 10-15 s were used. Their parameters were: the sampling frequency equal to 8 kHz, 16 bit/sample resolution and the mono track. For such parameters, the number of the octave bands exploited equaled to 9, and hence the size of codevectors were as below:

- 9 elements (the Spectral Flatness Measure)
- 9 elements (the Unpredictability Measure)
- 18 elements (statistical moments)

As refers RNNs, the number of hidden neurons was arbitrary set to 10 and 15 neurons. Bipolar continuous functions were used as NAFs except for the ones in the output layer. There were linear functions employed instead. The length of time sequences was set to 5. In result, the size of the codebook varied from 20 to 40 entries.

The experiments were carried with respect to the various kinds of codevectors. On the basis of some listening tests, the best quality of audio was achieved for the Unpredictability Measure, and the worst - for the Spectral Flatness Measure.

As an example of the efficiency of the proposed neural estimator, Fig. 2 presents spectrograms of a noisy audio and the enhanced one. The horizontal axis represents the time domain in seconds, whereas the vertical one - the frequency domain in Hz. In turn, the signal level is denoted by the intensity of grayness: the whiter color, the more intense is the audio.

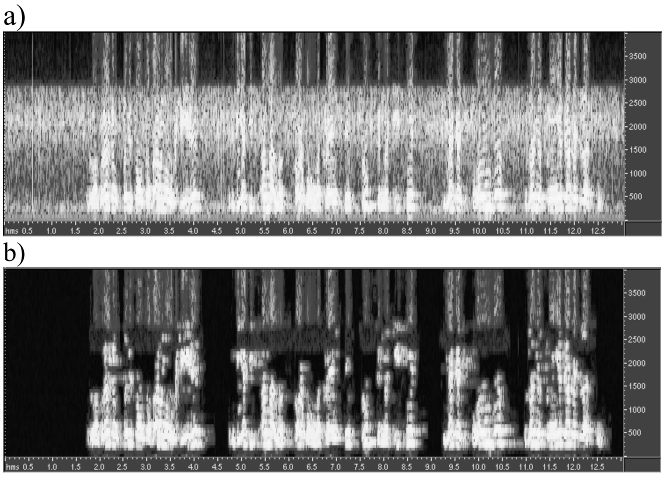

Fig. 2. Spectrograms of the example of noise reduction: (a) noisy signal, (b) restored audio.

5 Conclusions

In the paper, a RNN-based estimator for state sequence estimation was presented. Such a kind of neural network was chosen, since it enables processing time series and can deal with temporal relationships between input data. This feature makes RNNs very useful in the case of estimation of non-stationary noise, what was supported by a number of experiments. In these experiments, audio signals originated from old & original recordings dating from the 20-ties of the previous century.

The proposal of the estimation exploits a codebook-based solution which additionally require access keys - codevectors including some parameters describing quantitatively a noisy character of a given signal. It turns out that the choice of the codevector parameters can influence the quality of the enhanced audio. Three kinds of such parameters were proposed by the author and on the basis of the experiments' results - at least one of them shows to be quite efficient, namely the Unpredictability Measure.

Furthermore, although RNN-based processing is considered to be very time-consuming and computationally complex, the number of codevector parameters, the size of RNNs and the number of data vectors to be processed made the neural estimation acceptable fast to be put to use in practice.

Acknowledgements

The research is sponsored by: the Foundation for Polish Science and the State Committee for Scientific Research, Warsaw, Poland. Grant No. 8 T11D 028 19.

References:

[1] W. Chang, M. Mak, A Conjugate Gradient Learning Algorithm for Recurrent Neural Networks, *Neurocomputing*, Vol. 24, 1999, pp. 173-189.

[2] A. Czyzewski, R. Krolikowski, Noise Reduction in Audio Employing Auditory Masking Approach, *Proc. of the 106th Audio Engineering Society Conv.*, Munich, Germany, 1999, Preprint No. 4930.

[3] A. Czyzewski, R. Krolikowski, Noise Reduction in Audio Signals Based on the Perceptual Coding Approach, *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 1999, pp. 147-150.

[4] A. Czyzewski, R. Krolikowski, Neuro-Rough Control of Masking Thresholds for Audio Signal Enhancement, *J. of NeuroComputing*, Vol. 36, 2001, pp. 5-27.

[5] J. Elman, Finding Structure in Time, *Cognitive Science*, Vol. 14, 1990, pp. 179-211.

[6] Y. Eprahim, D. Malah, B. Juang, On the Application of Hidden Markov Models for Enhancing Noisy Speech, *IEEE Trans. on Acoustics, Speech and Audio Processing*, Vol. 37, No. 12, December 1989, pp. 1846-1856.

[7] J. Gibson, B. Koo, Filtering of Colored Noise for Speech Enhancement and Coding, *IEEE Trans. on Signal Processing*, Vol. 39, No. 8, August 1991, pp. 1732-1742.

[8] M. Goudreau, C. Giles, S. Chakradhar, D. Chen, First-order vs. Second-order Single Layer Recurrent Neural Networks, *IEEE Trans. on Neural Networks*, Vol. 5, No. 3, 1994, pp. 511-513.

[9] R. Krolikowski, Exploitation of Self-Organising Maps for the Reduction of Non-Stationary Noise in Speech Signals, *CD-ROM Proc. of ICSC Neural Computation*, Berlin, Germany, 2000.

[10] T. Lin, B. Horne, P. Tino, C. Giles, Learning Long-term Dependencies in NARX Recurrent Neural Networks, *IEEE Trans. on Neural Networks*, Vol. 7, No. 6, 1996, pp. 1329-1351.

[11] M. Mak, K. Ku, Y. Lu, On the Improvement of the Real Time Recurrent Learning Algorithm for Recurrent Neural Networks, *Neurocomputing*, Vol. 24, 1999, pp. 13-36.

[12] MPEG-2, ISO/IEC 13818-3, Generic Coding of Moving Pictures and Associated Audio Information, Part 3: Audio, 1995.

[13] J. Picone, Continuous Speech Recognition Using Hidden Markov Models, *IEEE Audio, Speech and Signal Processing Magazine*, Vol. 7, No. 3, July 1990, pp. 26-41.

[14] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of IEEE*, Vol. 77, No. 2, February 1989, pp. 257-286.

[15] S. Shlien, Guide to MPEG-1 Audio Standard, *IEEE Trans. on Broadcasting*, Vol. 40, No. 4, December 1994, pp. 206-218.

[16] T. Siegelmann, B. Horne, C. Giles, Computational Capabilities of Recurrent NARX Neural Networks, *IEEE Trans. on Systems, Man and Cybernetics - Part B*: Cybernetics, Vol. 27, No. 2, 1997, pp. 208-228.

[17] B. Sim, Y. Tong, J. Chang, C. Tan, A Parametric Formulation of the Generalized Spectral Subtraction Method, *IEEE Trans. on Speech and Audio Processing*, Vol. 6, No. 4, July 1998, pp. 328-337.

[18] J. Sum, W. Kan, G. Young, A Note on the Equivalence of NARX and RNN, *Neural Computing & Applications*, Vol. 8, 1999, pp. 33-39.

[19] S. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*, Wiley & Teubner, New York 1997.

[20] S. Vaseghi, R. Frayling-Cork, Restoration of Old Gramophone Recordings, *J. of Audio Engineering Society*, Vol. 40, No. 10, October 1997, pp. 791-800.

[21] B. Widrow, S. Stearns, *Adaptive Signal Processing*, Prentice-Hall International Inc., New Jersey 1985.

[22] R. Wiliams, D. Zipser, A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, Vol. 1, 1989, pp. 270-280.

[23] J. Zurada, *Introduction to Artificial Neural Networks*, West Publishing Company, St. Paul New 1992.