

A RECURRENT NEURAL FILTER FOR ADAPTIVE NOISE CANCELLATION

Paris A. Mastorocostas, Dimitris N. Varsamis, Constantinos A. Mastorocostas, Constantinos S. Hilas
Department of Informatics & Communications, Technological Educational Institute of Serres
62124, Serres
Greece

mast@teiser.gr, dvarsam@teiser.gr, costas_mast@teiser.gr, chilas@teiser.gr

ABSTRACT

This paper presents a dynamic neural filter for adaptive noise cancellation. The cancellation task is transformed to a system-identification problem, which is tackled by use of the Block-Diagonal Recurrent Neural Network. The filter is applied to a benchmark noise cancellation problem, where a comparative analysis with a series of other dynamic models is conducted, underlining the effectiveness of the proposed filter and its superior performance over its competing rivals.

KEY WORDS

block-diagonal recurrent networks, adaptive noise cancellation

1. Introduction

Extraction of an information signal buried in noise is one of the benchmark problems in the area of signal processing [1]. The issue of noise cancellation is encountered in many cases, including interference in electrocardiographs and periodic interference in speech signals. The most common method of signal estimation is to pass the noisy signal through a filter, which tends to suppress the noise while leaving the signal relatively unchanged. The filters applied to this problem are fixed or adaptive. The design of the former is based on prior knowledge of both the signal and the noise. Adaptive filters, on the other hand, have the ability to adjust their parameters automatically, requiring little or no prior knowledge of the signal or noise characteristics.

The problem of adaptive noise cancellation has been widely studied during the last decades and there exists a variety of filters in literature. Recently, neural networks have been employed to this area, exhibiting promising results [2]-[3]. In all cases, however, the suggested structures are static and the series-parallel identification approach is followed. Therefore, these models provide insufficient signal estimations when noise passes through nonlinear dynamic channels.

In an attempt to alleviate this problem, a number of recurrent neural and fuzzy-neural models have been suggested as dynamic adaptive noise cancellers [4]-[6].

These models are capable of effectively model the dynamics of a channel and exhibit superior cancellation performance compared to the aforementioned static neural models.

In this work an alternative recurrent structure is proposed as a noise cancellation filter. The filter is implemented by the Block-Diagonal Recurrent Neural Network [7], which has been proved in [8] to be an efficient identification tool.

The rest of paper is organized as follows: In Section 2 the transformation of the noise cancellation problem to a system identification problem is given. In the next section the proposed model is briefly described. Finally, Section 4 hosts the simulation results, where a comparative analysis with other recurrent neural and fuzzy models is conducted.

2. Transformation of the noise cancellation problem to a system identification problem

According to [1], a typical structure of a noise cancellation system is shown in Fig. 1 where additive noise, $n(k)$, corrupts the information signal, $s(k)$, resulting in the noisy signal, $d(k)$. The noise and information signals are assumed to be uncorrelated. The principle of noise cancellation is based on the assumption that both the noisy signal $d(k)$, and a filtered or distorted measurement of the noise, named *reference noise* $x(k)$, are available. Noise $x(k)$ is considered to pass through a channel with a transfer function $T(.)$. Under the assumption that the inverse of the filter noise distortion can be estimated, the noise corrupting the signal can be identified and cancelled. In this perspective, the problem of noise cancellation can be transformed to a system identification problem [4] as follows:

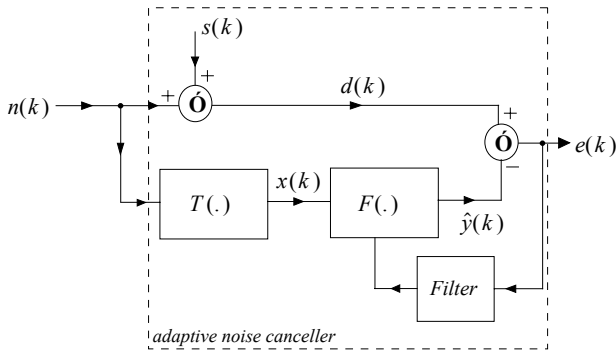


Fig. 1. The problem of adaptive noise cancellation

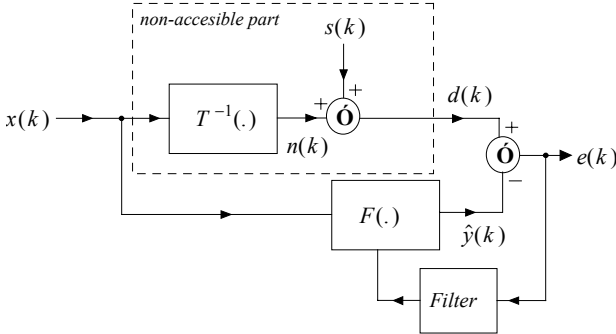


Fig. 2. Adaptive noise cancellation as a system identification problem

Let $F(\cdot)$ denote the transfer function of the system. It is derived from Fig. 2:

$$\hat{y}(k) = F(x(k)) = n(k) = \hat{T}^{-1}(x(k)) \quad (1)$$

and

$$e(k) = d(k) - \hat{y}(k) = s(k) + n(k) - \hat{y}(k) \rightarrow s(k) \quad (2)$$

Let $x(k)$, $d(k)$ be considered as the desired input and output, respectively, of the system $F(\cdot)$. According to (1) and (2), the error $e(k)$ will correspond to the information signal, which can be regarded as noise, additive to the output of the system $\hat{y}(k)$, as shown in Fig. 2.

It is derived from (2):

$$e^2(k) = (d(k) - y(k))^2 \Rightarrow E\{e^2(k)\} = E\{s^2(k)\} + E\{(n(k) - \hat{y}(k))^2\} + 2E\{s(k) \cdot n(k)\} - 2E\{s(k) \cdot \hat{y}(k)\} \quad (3)$$

The information signal is statistically uncorrelated to the noise $n(k)$ and its estimate $\hat{y}(k)$. Therefore the last two terms of (3) are equal to zero and (3) becomes

$$E\{e^2(k)\} = E\{s^2(k)\} + E\{(n(k) - \hat{y}(k))^2\} \quad (4)$$

Applying an optimization method, the parameters of the adaptive filter should be adjusted such that an error measure is minimized. Since the power of the information signal remains unchanged, minimizing the error measure leads to minimization of $E\{(n(k) - \hat{y}(k))^2\}$ and, according to (4), to minimization of the quantity $E\{(e(k) - s(k))^2\}$.

Thus, minimization of the total output power of the adaptive model leads to the optimal mean squared estimate of the information signal.

3. The proposed network and the modeling method

As shown in the previous section, the noise cancellation problem can be handled as a system identification problem. In this perspective, the model employed to perform system identification is the Block-Diagonal Recurrent Neural Network (BDRNN) [7]. The Block-Diagonal Recurrent Neural Network is a two-layer network, with the output layer being static and the hidden layer being dynamic. The hidden layer consists of pairs of neurons (blocks); there are feedback connections between the neurons of each pair, introducing dynamics to the network.

The operation of the BDRNN with m inputs, r outputs and N neurons at the hidden layer is described by the following set of state equations:

$$\mathbf{x}(k) = f_a(W \cdot \mathbf{x}(k-1) + B \cdot \mathbf{u}(k)) \quad (5a)$$

$$\mathbf{y}(k) = f_b(C \cdot \mathbf{x}(k)) \quad (5b)$$

where:

- f_a , f_b are the neuron activation functions of the hidden and the output layers, respectively. In the following, the activation functions are both chosen to be

$$\text{the sigmoid function } f(z) = \frac{1 - e^{-a_n \cdot z}}{1 + e^{-a_n \cdot z}}.$$

- $\mathbf{u}(k) = [u_i(k)]$ is a m -element vector, comprising the inputs of the network at time k .

- $\mathbf{x}(k) = [x_i(k)]$ is a N -element vector, comprising the outputs of the hidden layer. In particular, $x_i(k)$ is the output of the i -th hidden neuron at time k .

- $\mathbf{y}(k) = [y_i(k)]$ is a r -element vector, comprising the outputs of the network at time k .

- $B = [b_{i,j}]$ and $C = [c_{i,j}]$ are $N \times m$ and $r \times N$ input and output weight matrices, respectively.

- $W = [w_{i,j}]$ is the $N \times N$ block diagonal feedback matrix. In particular,

$$w_{i,j} = \begin{cases} \neq 0 & \text{if } i = j \\ \neq 0 & \text{if } i \neq j \text{ and } i = j - 1 \text{ and } i \text{ is odd} \\ \neq 0 & \text{if } i \neq j \text{ and } i = j + 1 \text{ and } i \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

The feedback matrix, W , is block diagonal:

$$W = \text{diag} \left[W^{(1)}, \dots, W^{(N/2)} \right]; \text{ each diagonal element,}$$

corresponding to a block of recurrent neurons, has a block submatrix in the form

$$W^{(i)} = \begin{bmatrix} w_{2i,2i} & w_{2i,2i+1} \\ -w_{2i,2i+1} & w_{2i,2i} \end{bmatrix} = \begin{bmatrix} w_i^{(1)} & w_i^{(2)} \\ -w_i^{(2)} & w_i^{(1)} \end{bmatrix} \quad i = 1, 2, \dots, \frac{N}{2} \quad (6)$$

Equation (6) describes a special case of BDRNN, which has *scaled orthogonal sub-matrices*. As shown in the literature, this network exhibits superior modelling capabilities over other forms of BDRNN.

In view of the above, the state equations (5) for the Scaled Orthogonal BDRNN can take the following form:

$$x_{2i-1}(k) = f_a \left(\sum_{j=1}^m b_{2i-1,j} \cdot u_j(k) + w_i^{(1)} \cdot x_{2i-1}(k-1) + w_i^{(2)} \cdot x_{2i}(k-1) \right), \quad i = 1, \dots, \frac{N}{2} \quad (7a)$$

$$x_{2i}(k) = f_a \left(\sum_{j=1}^m b_{2i,j} \cdot u_j(k) - w_i^{(2)} \cdot x_{2i-1}(k-1) + w_i^{(1)} \cdot x_{2i}(k-1) \right), \quad i = 1, \dots, \frac{N}{2} \quad (7b)$$

$$y_l(k) = f_b \left(\sum_{j=1}^N c_{l,j} \cdot x_j(k) \right), \quad l = 1, \dots, r \quad (7c)$$

where $w_i^{(1)}$, $w_i^{(2)}$ are the feedback weights at the hidden layer.

The BDRNN is trained by the RENNCOM algorithm. The algorithm, entitled Recurrent Neural Network Constrained Optimization Method is fully described in [8] and a brief description is given in the sequel.

The training method aims at transforming the learning process to a constrained optimization problem, which has been solved using methods based on optimal control theory and the calculus of variations. For the case of a m -input- r -output BDRNN with scaled orthogonal feedback submatrices, the objective of the learning process is to adjust the network parameters so that a prescribed input/output mapping is captured. Learning is carried out in a parallel mode using a data batch $\{(\mathbf{u}(k), \hat{\mathbf{y}}(k)), k = 1, \dots, k_f\}$ comprising k_f input-output pairs. Two vectors are introduced:

a) the *state vector* $\mathbf{st}(t)$, defined as:

$$\mathbf{st}(k) = [x_1(k), \dots, x_N(k), y_1(k), \dots, y_r(k)]^T \quad (8)$$

comprising the outputs of the hidden and the output layer.

b) the *control vector* $\boldsymbol{\theta}$ comprising the synaptic and feedback weights

$$\boldsymbol{\theta} = \left[b_{1,1}, \dots, b_{N,m}, w_1^{(1)}, w_{\frac{N}{2}}^{(1)}, w_1^{(2)}, w_{\frac{N}{2}}^{(2)}, c_{1,1}, \dots, c_{r,N} \right]^T \quad (9)$$

For a data set including k_f pairs, the state equations (7) are written

$$\mathbf{f}(\mathbf{st}(k), \boldsymbol{\theta}(k)) = \boldsymbol{\theta}, \quad k = 1, \dots, k_f \quad (10)$$

$$f_a \left(\sum_{j=1}^m b_{2i-1,j} u_j(k) + w_i^{(1)} x_{2i-1}(k-1) + w_i^{(2)} x_{2i}(k-1) \right) - x_{2i-1}(k) = 0 \quad (11a)$$

$$f_a \left(\sum_{j=1}^m b_{2i,j} u_j(k) - w_i^{(2)} x_{2i-1}(k-1) + w_i^{(1)} x_{2i}(k-1) \right) - x_{2i}(k) = 0 \quad (11b)$$

$$f_b \left(\sum_{j=1}^N c_{l,j} x_j(k) \right) - y_l(k) = 0 \quad (k_f \times r \text{ equations}) \quad (11c)$$

The suggested algorithm is an iterative procedure which aims at achieving two objectives, simultaneously:

- It is desired that an error measure function, E , should be minimized, in order to perform the input-output mapping, successfully. The Mean Squared Error (MSE) is selected here and is defined by

$$E = \frac{1}{k_f} \sum_{k=1}^{k_f} \sum_{l=1}^r [y_l(k) - \hat{y}_l(k)]^2 \quad (12)$$

where $y_l(k)$ is the l -th model output, $\hat{y}_l(k)$ is the l -th desired (actual) output of the system at time step k .

- Secondly, a nonlinear multivariable function Φ , called the *pay-off function*, should be optimized. In the present case the main scope is to facilitate and accelerate the learning process. The following pay-off function is selected:

$$\Phi = (\boldsymbol{\theta} - \boldsymbol{\theta}_{cur})^T (\boldsymbol{\theta}_{cur} - \boldsymbol{\theta}_{prev}) \quad (13)$$

where $\boldsymbol{\theta}_{cur}$, $\boldsymbol{\theta}_{prev}$ are the control vectors at the current and the previous epochs, respectively. Maximization of the pay-off function at each epoch implies that the current and previous weight updates are highly aligned, thus avoiding zig-zag trajectories and improving the convergence speed.

Optimization of (12) and (13) is iteratively performed with respect to the decision variables, $\boldsymbol{\theta}$ and the state variables, \mathbf{st} , under the architectural constraints imposed by the state equations (11). In that sense, the learning process can be regarded as a constrained optimization problem.

Following the procedure dictated in [8], the weights are updated as follows, with t being the epoch index:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + d\boldsymbol{\theta} \quad (14)$$

The corrective vector $d\boldsymbol{\theta}$ is given by:

$$d\boldsymbol{\theta} = -\Delta^2 \cdot \frac{\sqrt{I_{EE} - (\delta E)^2}}{\sqrt{I_{\Phi\Phi} \cdot I_{EE} - I_{E\Phi}^2}} \cdot \left[\mathbf{A}_{\Phi}^T - (\mathbf{A}_E)^T \cdot \frac{I_{E\Phi}}{I_{EE}} \right] + \Delta^2 \cdot (\mathbf{A}_E)^T \cdot \frac{\delta E}{I_{EE}} \quad (15)$$

where the following factors are calculated:

$$I_{EE} = \mathbf{A}_E \Delta^2 \mathbf{A}_E^T \quad (16a) \quad I_{\Phi\Phi} = \mathbf{A}_{\Phi} \Delta^2 \mathbf{A}_{\Phi}^T \quad (16b)$$

$$I_{E\Phi} = \mathbf{A}_{\Phi} \Delta^2 \mathbf{A}_E^T \quad (16c) \quad \delta E = -\xi \cdot \sqrt{I_{EE}} \quad (16d)$$

and ξ is a constant over $[0,1]$.

The intermediate variables $\mathbf{A}_\Phi = \left(\frac{\partial \Phi}{\partial \theta} \right)^T$ and

$\mathbf{A}_E = \left(\lambda^T \cdot \frac{\partial \mathbf{f}}{\partial \theta} \right)^T$ are calculated by use of the Lagrange

multipliers, which are given by:

$$\frac{\partial E}{\partial \mathbf{st}} + \lambda^T \cdot \frac{\partial \mathbf{f}}{\partial \mathbf{st}} = 0 \quad (17)$$

Matrix $\Delta = \text{diag} (\Delta_1, \Delta_2, \dots, \Delta_{N \times (m+r+1)})$ is the *Maximum Parameter Changes matrix (MPC)*. At each epoch the weights are changed by small amounts $d\theta$, so that the positive quadratic equation holds:

$$\sum_{i=1}^n \frac{(d\theta_i)^2}{\Delta_i^2} = 1 \quad \text{or} \quad d\theta^T (\Delta^2)^{-1} d\theta = 1 \quad (18)$$

where Δ_i is the MPC for θ_i . Equation (18) describes a hyper-ellipsoid centered at the point defined by the current control vector, whose axes are the Δ_i 's. Since the search space is defined in terms of Δ , the MPC's play an essential role in the learning process and is desired to be adaptable throughout the learning process. The adaptation mechanism is fully described in [8].

Summarizing, at each epoch $t = 1, 2, \dots$ the RENNCOM algorithm proceeds as follows: First, the current values of Δ 's are derived according to the suggested adaptation schedule. Next, parameter learning is performed, by calculating the optimal parameter changes. If the resulting error measure is smaller than a prescribed threshold, learning is completed, otherwise the whole process is repeated.

4. Simulation Results

In this section the proposed filter is applied to a noise cancellation problem, where the noise $n(k)$ passes through a nonlinear dynamic channel, producing the reference noise $x(k)$. The passage's dynamics is simulated by a second order nonlinear auto-regressive model with exogenous inputs (NARX) [4]:

$$x(k) = 0.25x(k-1) + 0.1x(k-2) + 0.5n(k-1) + 0.1n(k-2) - 0.2n(k-3) + 0.1n^2(k-2) + 0.08n(k-2)x(k-1) \quad (19)$$

The information signal $s(k)$ is a saw-tooth signal of unit magnitude, 50 samples period, as shown in Fig. 3a. The signal $s(k)$ is corrupted by a uniformly distributed white noise sequence varying in the range $[-2, 2]$ shown in Fig. 3b while the noise-corrupted signal $d(k)$ is depicted in Fig. 3c. The training data sets consist of 12000 pairs $[x(k), d(k)]$, while the testing set comprises 1000 data pairs.

The BDRNN comprises eight blocks of neurons in the hidden layer. Training lasts for 1000 epochs. The learning

parameters of the RENNCOM algorithm are shown in Table 1.

A time section of eight periods of the recovered signal is presented in Fig.3d. It is evident that even though the information signal has half the amplitude of the additive noise, the former is accurately identified, with the exception of a few high frequency components.

Table 1
The learning parameters of the RENNCOM method

a_n	n^+	n^-	Δ_0	ξ
2	1.05	0.95	1E-2	0.9

In the sequel, a comparative analysis is attempted between the suggested BDRNN filter and a class of noise cancellation filters including finite impulse response (FIR) and infinite impulse response (IIR) neural networks and a fuzzy inference system:

- A Dynamic Fuzzy Neural Network (DFNN), taken from [6].
- A recurrent three-layer neural network (FIRN), in the form of 1- H -1, having a linear input layer, and FIR synapses at the hidden and output layers, as described in [6]. The outputs of the hidden and output layers are given by the following formulas:

$$O_i^1(k) = \tanh \left(\sum_{q=0}^{O_u} [w_{iq}^1 u(k-q)] + w_i^3 \right) \quad i = 1, \dots, H \quad (20a)$$

$$y(k) = \tanh \left(\sum_{j=1}^H \sum_{q=0}^{O_y} [w_{jq}^4 O_j^1(k-q)] + w^6 \right) \quad (20b)$$

- A recurrent three-layer neural network (IIRN), in the form of 1- H -1, having a linear input layer, and Frasconi-Gori-Soda [9] neurons in the hidden and output layers. The outputs of these neurons are determined by the following formulas:

$$O_i^1(k) = \tanh \left(\sum_{q=0}^{O_u} w_{iq}^1 u(k-q) + \sum_{j=1}^{O_{y1}} w_{ij}^2 O_j^1(k-j) + w_i^3 \right) \quad i = 1, \dots, H \quad (21a)$$

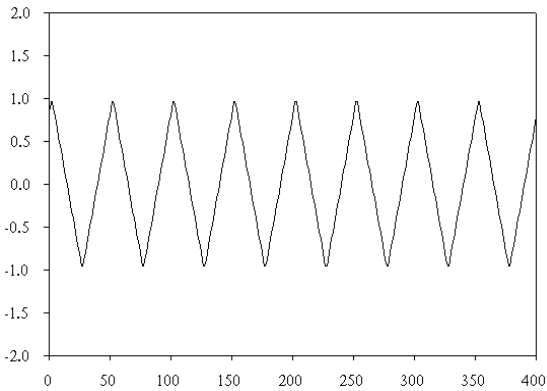
$$y(k) = \tanh \left(\sum_{j=1}^H \sum_{q=0}^{O_{y3}} w_{jq}^4 O_j^1(k-q) + \sum_{j=1}^{O_{y2}} w_j^5 y(k-j) + w^6 \right) \quad (21b)$$

For each of the above-mentioned models, exhaustive experimentation has been carried out in order to extract the most efficient structure, which is going to participate in the comparative analysis. Since the FIRN can be regarded as an IIRN without feedback, the D-FUNCOM [6] algorithm is chosen to be the training method for the neural network models. Thus, the structures of the suggested filter and those of the DFNN and the neural networks are evaluated using a similar learning method, since our concern in this paper is to investigate the performance of different models rather than focusing on the learning attributes of the training algorithm. The

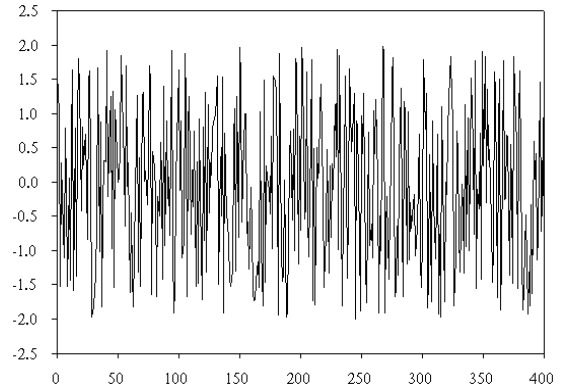
structural and learning characteristics of the competing filters are given in Table 2 while Table 3 hosts the simulation results.

Table 3
Comparative analysis

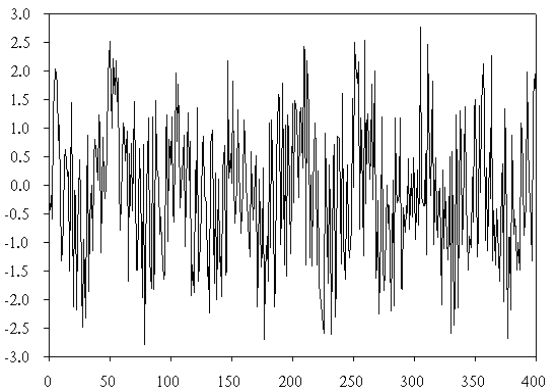
Model	MSE _{trn}	MSE _{st}	Parameters
BDRNN	0.01290	0.01240	64
DFNN	0.01360	0.01310	62
IIR	0.0157	0.01690	87
FIR	0.0617	0.06180	85



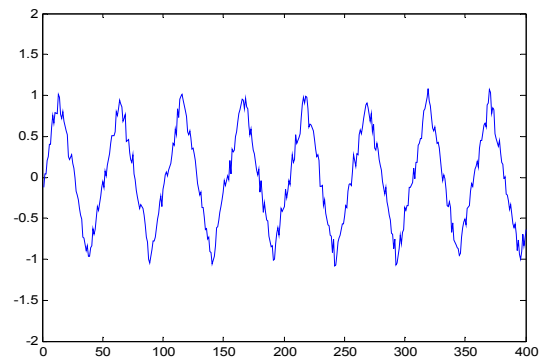
(a) information signal $s(k)$



(b) additive noise $n(k)$



(c) noise corrupted signal $d(k)$



(d) recovered signal $\hat{s}(k)$

Fig. 3

Table 2
Characteristics of the comparing filters

Model	Learning Method	Iterations	Model's characteristics				
			$H=4$	$O_u=2$	$O_{y_1}=1$	$O_{y_2}=2$	$O_{y_3}=1$
DFNN	D-FUNCOM	1000	$H=4$	$O_u=2$	$O_{y_1}=1$	$O_{y_2}=2$	$O_{y_3}=1$
FIRN	D-FUNCOM	1000	$H=12$	$O_u=2$	$O_y=2$		
IIRN	D-FUNCOM	1000	$H=12$	$O_u=2$	$O_{y_1}=1$	$O_{y_2}=2$	$O_{y_3}=1$

5. Conclusion

A new dynamic filter, with a relatively simple structure, for adaptive noise cancellation has been proposed. The

Based on the results cited in Table 3, it becomes evident that the dynamic nature of the channel is clearly reflected to the results, since the FIRN neural network fails in sufficiently tracking the passage dynamics, $T^{-1}(\cdot)$. Moreover, the proposed filter exhibits superior performance compared to IIRN and FIRN, and a similar performance compared to DFNN, requiring the same number of parameters with respect to DFNN. Therefore, it can be argued that the suggested dynamic neural model constitutes an effective noise cancellation tool, with a reduced complexity compared to its competing rivals.

cancellation problem has been transformed to a system identification problem, tackled by an identifier based on the Block-Diagonal Recurrent Neural Network. The proposed filter has been applied to a noise cancellation problem, where the noise passes through a nonlinear dynamic channel. A comparative analysis with a series of

filters has been conducted, underlining the effectiveness of the proposed noise canceller.

Acknowledgements

This work was supported in part by the European research project Archimedes II.

References

- [1] B. Widrow, et al, Adaptive Noise Cancellation: Principles and Applications, *IEEE Proceedings*, 63, 1975, 1692-1716.
- [2] I. Cha, & S.A. Kassam, Interference Cancellation Using Radial Basis Function Networks, *Signal Processing*, 47, 1995, 247-268.
- [3] S.A. Vorobyov, & A. Cichocki, Hyper Radial Basis Function Neural Network for Interference Cancellation with Nonlinear Processing of Reference Signal, *Digital Signal Processing*, 11, 2001, 204-221.
- [4] S.A. Billings, & C.F. Fung, Recurrent Radial Basis Function Networks for Adaptive Noise Cancellation, *Neural Networks*, 8(2), 1995, 273-290.
- [5] G. Kechriotis, E. Zervas, & E.S. Manolakos, Using Recurrent Neural Networks for Adaptive Communication Channel Equalization, *IEEE Transactions on Neural Networks*, 5(2), 1994, 267-278.
- [6] P.A. Mastorocostas, & J.B. Theocharis, A Recurrent Fuzzy Neural Model for Dynamic System Identification, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 32(2), 2002, 176-190.
- [7] S.C. Sivakumar, W. Robertson, & W.J. Phillips, On-Line Stabilization of Block-Diagonal Recurrent Neural Networks, *IEEE Transactions on Neural Networks*, 10(1), 1999, 167-175.
- [8] P.A. Mastorocostas, & J.B. Theocharis, A Stable Learning Algorithm for Block-Diagonal Recurrent Neural Networks: Application to the Analysis of Lung Sounds, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 36(2), 2006, 242-254.
- [9] P. Frasconi, M. Gori, & G. Soda, Local Feedback Multilayered Networks, *Neural Computation*, 4, 1992, 120-130.