A Recurrent Fuzzy Filter for Adaptive Noise Cancellation

Paris Mastorocostas, Dimitris Varsamis, Constantinos Mastorocostas, Constantinos Hilas
Dpt. of Informatics, Technological Educational Institute of Serres, 62124, Serres, GREECE
mast@teiser.gr

Abstract

This paper presents a recurrent fuzzy-neural filter for adaptive noise cancellation. The cancellation task is transformed to a system-identification problem, which is tackled by use of the Dynamic Neuron-based Fuzzy Neural Network. Extensive simulation results are given and performance comparison with a series of other dynamic fuzzy and neural models is conducted, underlining the effectiveness of the proposed filter and its superior performance over its competing rivals.

1. Introduction

Extraction of an information signal buried in noise is one of the benchmark problems in the area of signal processing. The issue of noise cancellation is encountered in many cases, including the cancellation of broad-band interference in the side-lobes of an antenna array, 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 issue of adaptive noise cancellation has been widely studied during the last decades and there exists a variety of filters in literature. Recently, fuzzy logic has been established as an effective tool for adaptive filtering, and several fuzzy filters have been proposed [1]-[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, the Dynamic Fuzzy Neural Network [4], has been suggested as a dynamic adaptive noise canceller. As shown in [4], DFNN is capable of effectively model the dynamics of a channel and exhibits superior cancellation performance compared to the aforementioned static fuzzy models.

In this work an alternative recurrent fuzzy structure is proposed as a noise cancellation filter. The filter is implemented by the Dynamic Neuron-based Fuzzy Neural Network, which has been proposed in [5] as an efficient identification tool.

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

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

According to [6], 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 noise 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

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 **IEEE**

can be transformed to a system identification problem [7] as follows:

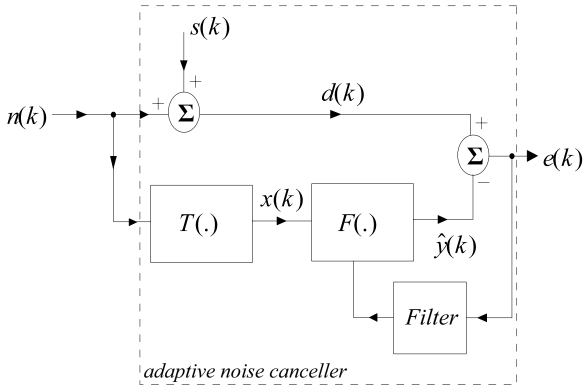


Figure 1. The problem of adaptive noise cancellation

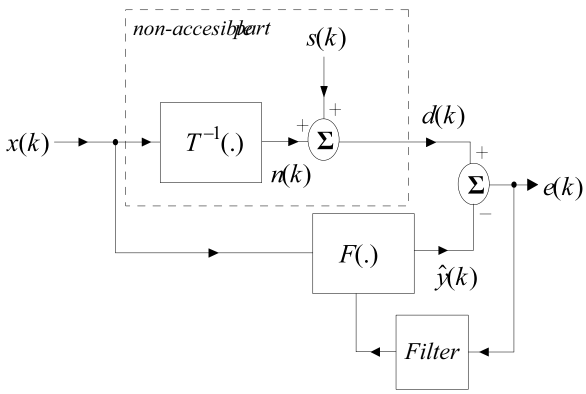


Figure 2. Adaptive noise cancellation as a system identification problem

Let $F(.)$ 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 $F(.)$. 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):

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

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

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

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 $E\left\{\left(n(k) - \hat{y}(k)\right)^2\right\}$ and, according to (4), to minimization of the quantity $E\left\{\left(e(k) - s(k)\right)^2\right\}$. 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 inference system 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 fuzzy inference system employed to perform system identification is the DN-FNN [5]. The classic TSK model consists of a set of linguistic IF-THEN rules with polynomial consequent parts:

$$\begin{aligned} &R^{(j)}: \ IF \ z_1 \ is \ A_1^j \ AND \ \ldots \ AND \ z_m \ is \ A_m^j \\ &THEN \ g_j = w_0^j + w_1^j u_1^j + \ldots + w_{n_j}^j u_{n_j}^j \end{aligned} \quad (5)$$

where, in the general case, the consequent part of each fuzzy rule comprises its own input vector $u^j = [u_1^j \ldots u_{n_j}^j]$ and the premise part input vector $z = [z_1 \ldots u_m]$ is common to all rules. If the consequent part input variables u_i^j are substituted by the recurrent functions $x_{ji}(k) = f\left(x_{ji}(k-i), u^j(k), u^j(k-1), \ldots, u^j(k-O_u)\right)$, then the rule outputs are linear combinations of the dynamic elements $x_{ji}(k)$, which have internal feedback. In this case (5) becomes

$$\begin{aligned} &R^{(j)}: \ IF \ z_1(k) \ is \ A_1^j \ AND \ldots AND \ z_m(k) \ is \ A_m^j \\ &THEN \ g_j(k) = \theta_{j1} \cdot x_{j1}(k) + \ldots + \theta_{jn_j} \cdot x_{jn_j}(k) \end{aligned} \quad (6)$$

Let the DN-FNN consist of r fuzzy rules, and be fed with k_f input-output data. The model output is given by

$$y(k) = \frac{\sum_{j=1}^{r} \mu_j(k) \cdot g_j(k)}{\sum_{l=1}^{r} \mu_l(k)} = \sum_{j=1}^{r} v_j(k) \cdot g_j(k) \quad (7)$$

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

where $v_j(k)=\frac{\mu_j(k)}{\sum_{l=1}^{r}\mu_l(k)}$ is the normalized degree of fulfillment for the j-th rule.

When a single-dimensional input vector $\boldsymbol{u}$ is considered, the functions of dynamic elements can be written

$$x_{ji}(k)=f\Big(\sum_{q=0}^{Ou}w_{1jiq}\cdot u(k-q)+w_{2ji}\cdot x_{ji}(k-i)+w_{0ji}\Big) \tag{8}$$

where $f(.)=\tanh(.)$ is the activation function. The neuron that implements this function is called ***d**ynamic **n**euron* (*DN*), having an output feedback structure. The *DN* is characterized by the orders of the infinite impulse response (IIR) synapse, O_u and *i*; thus the formalism $DN(O_u,i)$ fully determines a dynamic neuron.

As mentioned in [5], the DN-FNN is a generalized TSK dynamic model. The rules are not linked with each other in time, neither through external nor internal feedback; they are connected merely via the defuzzification part. The premise and deffuzification blocks are static while the consequent block is dynamic. Moreover, the linear constraint that the neurons of each consequent part should fulfill facilitates the transformation of the DN-FNN to an autoregressive model; therefore, the Dynamic Orthogonal Least Squares method (D-OLS), which constitutes an adaptation of the standard OLS algorithm to recurrent systems, can be applied to determine the structure and tune the parameters of the DN-FNN.

The D-OLS is fully described in [5]. In brief, the method aims at developing DN-FNN models by transforming them to autoregressive models. In the sequel, for a given partition of the input space, the modeling method builds the consequent parts of the fuzzy rules. From a set of candidate dynamic neurons $DN(O_u,i)$, $i=1,...,O_y$, the D-OLS selects the most significant neurons for each rule and calculate its coefficients. The method is a sequential procedure, where at each step the most significant consequent term (dynamic neuron) is extracted from a pool of candidate terms. When the process is completed, the selected dynamic neurons are appended to the consequent parts of the corresponding rules and their coefficients are calculated.

The output of the DN-FNN can be written in the form:

$$y(k)=\sum_{j=1}^{r}v_j(k)\cdot\Big[\theta_{j1}\cdot x_{j1}(k)+...+\theta_{jn_j}\cdot x_{jn_j}(k)\Big]= \\ =\sum_{j=1}^{r}\sum_{l=1}^{n_j}\Big[v_j(k)\cdot x_{jl}(k)\Big]\cdot\theta_{jl} \tag{9}$$

Introducing the following vectors

$$\boldsymbol{w}(k)=[v_j(k)\cdot x_{jl}(k)]=[w_1(k),w_2(k),....,w_Q(k)]^T$$

$$j=1,...,r,\quad l=1,...,n_j,\ Q=\sum_{j=1}^{r}n_j \tag{10}$$

$$\boldsymbol{b}(k)=[\theta_{jl}]=[b_1,b_2,...,b_Q]^T\quad j=1,...,r,\ l=1,...,n_j \tag{11}$$

(9) is written

$$y(k)=\sum_{j=1}^{Q}w_j(k)\cdot b_j \tag{12}$$

Equation (12) corresponds to an autoregressive model.

The classic Orthogonal Least Squares Method is applied to autoregressive models in the form of (8), selects the most significant terms w_j from the set of Q candidate terms and calculates the corresponding coefficients b_j. However, in the case of dynamic systems these terms are quite different from the respective terms in [8], where the classic OLS method is applied to static fuzzy systems. In [8] for a given data set, w_j was completely determined. In the case of dynamic systems, as shown in (10), each term w_j is the product of a normalized degree of fulfillment, $v_j(k)$, and a neuron output, $x_{jl}(k)$. For a given dataset, $v_i(k)$ is determined. In order to determine the neuron output $x_{jl}(k)$, the structure of the neuron should be selected (neuron type and feedback structure) and their parameters should be calculated. Since the neurons are dynamic, the values of w_j are calculated recursively. Thus they are related through the dynamics of the DN's. Therefore they are unknown and should be calculated as well. For the DN-FNN model under consideration, the type and structure of DN's have been determined in the previous subsection. The parameters of $x_{jl}(k)$ remain unknown and should be determined through a learning process, since different parameter values lead to different trajectories of $x_{jl}(k)$ within the time interval $[1,k_f]$ and, consequently, to different terms w_j. Therefore, at each step of the D-OLS process a learning algorithm should be applied. As described in [5], the D-FUNCOM learning method [4] is employed.

A detailed mathematical description of the method along with a presentation in pseudo-C code can be found in [5].

During the model generation process, the data are separated to: *a)* a training data set D_{tr}, used by the learning algorithm, and *b)* a validation data set D_{val}, considerably smaller than D_{tr}, used as a stopping criterion, in order to decide the number of dynamic neurons that should be included in the final model. At each step of the D-OLS algorithm the error measure for D_{val} is monitored. As learning proceeds and the error measure keeps reducing, new *DN*''s are added to the model. The D-OLS algorithm is terminated when increase of the error measure for D_{val} in two successive steps is observed. The last two selected *RF*'s are rejected.

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) [7]:

$$x(k)=0.25x(k-1)+0.1\,x(k-2)+0.5\,n(k-1)+0.1\,n(k-2)-$$
$$-0.2\,n(k-3)+0.1\,n^2(k-2)+0.08\,n(k-2)\,x(k-1) \quad (13)$$

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. Both the training and testing data sets consist of 1000 pairs $[x(k),d(k)]$, while the validation set consists of 500 data pairs.

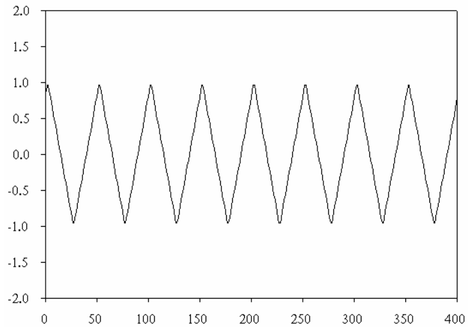
Following the procedure presented in [5], four fuzzy sets are assigned to the input. They are uniformly distributed along the input space, leading to a DN-FNN with four fuzzy rules. The consequent parts of the fuzzy rules will be a linear combination of the dynamic neurons that will be selected. Each rule comprises candidate dynamic neurons with maximum delay $d_{\max}=5$, therefore the size of the set of candidate consequent terms is $4\times 5=20$. The dynamic neuron outputs are given by (8). The order O_u is set to 1. The rest of the learning parameters are shown in Table I. At each step all candidate dynamic neurons are trained for 100 epochs. As mentioned in the previous section, the dynamic neuron introduces internal recurrence to the fuzzy model and is characterized by the orders O_u and i ($i=1,\ldots,d_{\max}$).

The model-building process is presented in the sequel. The procedure stops at the eighth step, where an increase of the error measure for the validation data set in two successive steps is observed. Therefore, the final model includes the first six selected *DN*'s. The error sequences for the training and validation data sets are given in Table II, along with the kind of the selected neurons and the rules they are attached to. According to Table II, the following comments are in order:

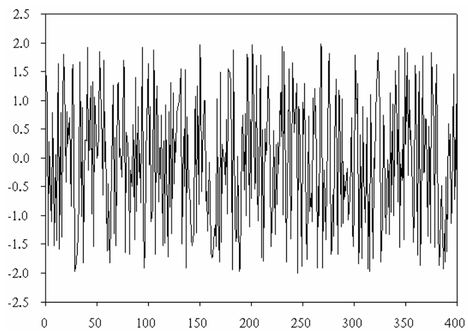
- All four rules are activated and the input space is fully covered by fuzzy sets, since terms from every rule are selected.
- Each rule requires dynamic neurons with different feedback, leading to the conclusion that the introduction of recurrence to the consequent part of the fuzzy rules is reasonable. Since each rule of the TSK model constitutes a local sub-model that operates in the region specified by the premise part of the rule, the identification problem in each region requires sub-models of various complexities.
- The evolution of the error measures with respect to the selected *DN*'s for the training and validation data sets is quite similar, a fact that can be attributed to the parallel learning mode. The final model attains MSE values of 0.01230, 0.01248 and 0.01241 for the training, validation and checking data sets, respectively, that are quite close to each other. Therefore, it can be argued that the model does not merely memorize the functional relationship of the input-output training data but identifies effectively the plant's dynamics, ensuring generalization.
- 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 D-FUNCOM method

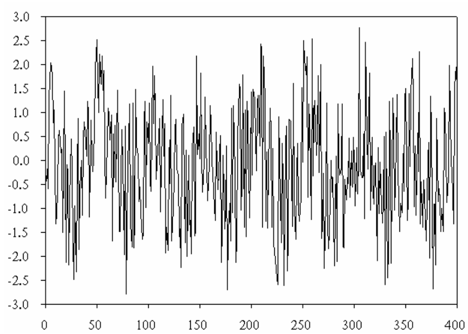
n^+	n^-	$\Delta_{\min}$	Δ_0	ξ
1.05	0.9	1E-4	0.1	0.9



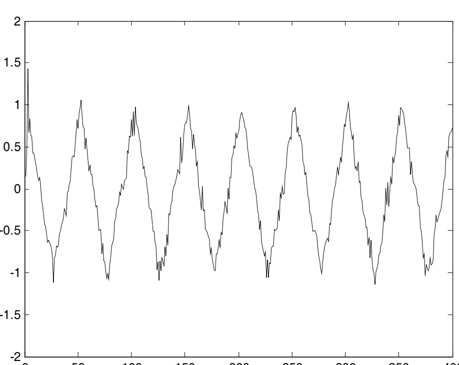
(a) Information signal *s(k)*



(b) Additive noise *n(k)*



(c) Noise corrupted signal *d(k)*



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

Figure 3.

Table 2. Ordering of the selected *DN's* and the corresponding error measures for the training and validation sets

No of *DN*	Rule	*DN*	*MSE* training	*MSE* validation
1	2	1	0.44330	0.45048
2	3	1	0.16180	0.16445
3	1	1	0.09981	0.10141
4	4	1	0.05100	0.05181
5	2	2	0.02450	0.02489
6	3	2	0.01230	**0.01248**
7	1	3	0.01092	0.01294
8	2	3	0.01078	0.01356

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

- A Dynamic Fuzzy Neural Network (DFNN), taken from [4].
- 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 [4]. The outputs of the hidden and output layers are given by the following formulas:

$$O_i^1(k) = \tanh\left(\sum_{q=0}^{Ou}\left[w_{i\,q}^1\, u(k-q)\right] + w_i^3\right) \quad i = 1,\ldots,H \quad (14a)$$

$$y(k) = \tanh\left(\sum_{j=1}^{H}\sum_{q=0}^{Oy}\left[w_{j\,q}^4\, O_j^1(k-q)\right] + w^6\right) \quad (14b)$$

- 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:

$$O_i^1(k) = \tanh\left(\sum_{q=0}^{O_u} w_{iq}^1 u(k-q) + \sum_{j=1}^{Oy_1} w_{ij}^2 O_i^1(k-j) + w_i^3\right) \quad (15a)$$

$$y(k) = \tanh\left(\sum_{j=1}^{H}\sum_{q=0}^{Oy_3} w_{jq}^4 O_j^1(k-q) + \sum_{j=1}^{Oy_2} w_j^5 y(k-j) + w^6\right) \quad (15b)$$

where $i = 1,\ldots,H$

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 algorithm is chosen to be

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 **IEEE**

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 common 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 III while Table V presents the simulation results.

Based on the results cited in Table IV, it becomes evident that the dynamic nature of the channel is clearly reflected to the results, since the FIRN neural network fail 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. Additionally, the model-building process has led to a considerably smaller model, requiring half the parameters of DFNN and nearly one third of the parameters of the neural models. This fact is due to the local modeling approach and the inference characteristics of the fuzzy systems. It becomes evident that the suggested neuro-fuzzy dynamic model constitutes an effective noise cancellation tool, with a reduced parameter set.

Table 4. Comparative analysis

Model	MSE training	MSE testing	Parameters
DN-FNN	0.01230	0.01241	30
DFNN	0.01360	0.01310	62
IIR	0.01570	0.01690	87
FIR	0.06170	0.06180	85

Acknowledgement

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

References

[1] J.-S.R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New Jersey, 1997.

[2] C.-T. Lin and C.-F. Juang, "An Adaptive Neural Fuzzy Filter and its Applications," *IEEE Trans. Syst., Man, and Cybern. - Part B*, IEEE, August 1997, pp. 635-656.

[3] L.-X. Wang and J.M. Mendel, "Fuzzy Adaptive Filters with Application to Nonlinear Channel Equalization," *IEEE Trans. Fuzzy Systems*, IEEE, August 1993, pp. 161-170.

[4] P.A. Mastorocostas and J.B. Theocharis, "A Recurrent Fuzzy Neural Model for Dynamic System Identification," *IEEE Trans. Syst., Man, and Cybern. - Part B*, IEEE, April 2002, pp. 176-190.

[5] P.A. Mastorocostas and J.B. Theocharis, "An Orthogonal Least Squares Method for Recurrent Fuzzy-Neural Modeling," *Fuzzy Sets and Systems*, December 2003, pp. 285-300.

[6] B. Widrow, et al, "Adaptive Noise Cancellation: Principles and Applications," *IEEE Proceedings*, IEEE, December 1975, pp. 1692-1716.

[7] S.A. Billings and C.F. Fung, "Recurrent Radial Basis Function Networks for Adaptive Noise Cancellation," *Neural Networks*, Elsevier, 1995, pp. 273-290.

[8] L.-X. Wang and J.M. Mendel, "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning," *IEEE Trans. Neural Networks*, IEEE, September 1992, pp. 807-814.

[9] P. Frasconi, M. Gori, and G. Soda, "Local Feedback Multilayered Networks," *Neural Computation*, MIT Press, vol. 4, pp. 120-130, 1992.

Table 3. Characteristics of the comparing filters

Model	Learning Method	Iterations	Model's characteristics				
DFNN	D-FUNCOM	1000	*H=4*	*Ou=2*	$Oy_1=1$	$Oy_2=2$	$Oy_3=1$
FIRN	D-FUNCOM	1000	*H=12*	*Ou=2*		*Oy=2*	
IIRN	D-FUNCOM	1000	*H=12*	*Ou=2*	$Oy_1=1$	$Oy_2=2$	$Oy_3=1$