PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 36, NOS. 3-4, PP. 215-226 (1992)

NEURAL NETWORK CONTROLLED ADAPTIVE FILTERS

Béla PATAKI

Department of Measurement and Instrument Engineering
Technical University of Budapest
H-1521 Budapest, Hungary

Received: 27, November, 1992

Abstract

Recently SZTIPÁNOVITS has proposed [2,3] an adaptive processing system — an adaptive FIR filter structure — that consists of a resonator based digital filter (RBDF) and a neural network. The RBDF is a highly parallel structure with several structural and implementational advantages, which in the proposed case performs a recursive transformation as well. This paper focuses on the advantages and disadvantages of the proposed composite structure. Some improvements are suggested, for example extending the parallelism of the structure to the neural network as well, which results in a better convergence rate of the training procedure. On the other hand, the convergence rate can be improved by using the combination of the output error and the transform domain component error. Finally the structure is extended to adaptive IIR filtering problems which requires the modification of the resonator based IIR structure.

Keywords: neural networks, IIR filter, improved training rate, adaptive filters, resonator bank digital filters.

Introduction

Over the last years adaptive filtering has been an active area of research with important results and applications in fields of adaptive control, signal processing and communication [1]. (E.g. adaptive noise cancellation, channel equalization, etc.)

The general structure of an adaptive filter [1] is shown in *Fig. 1.* Usually we define the system decomposing it into two main subsystems: a linear or nonlinear time-varying filter, with predefined structure characterized by an adjustable parameter set ($\Theta(n)$) and an adaptation algorithm which adjusts the parameters of the filter.

The adaptation algorithm can get information for the proper parameter adjustment from several sources:

- from the input signal(s),
- from the output signal(s),

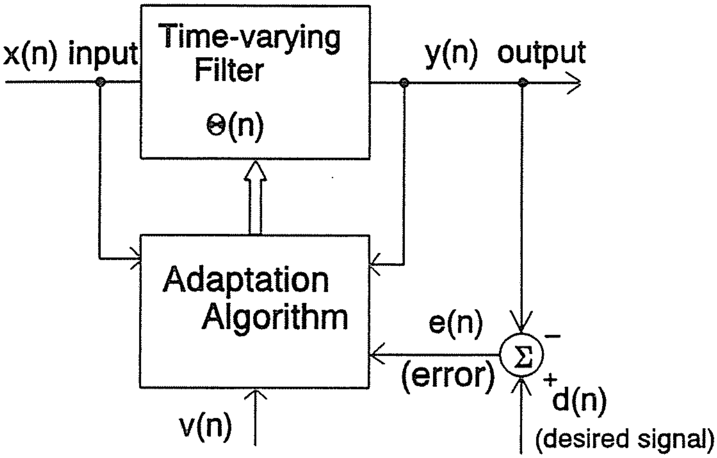

Fig. 1. General structure of adaptive filters

- from external signal(s), which are indirectly connected to the features of the ideal signal,
- from the desired (ideal) signal which is the purpose of adaptation.

In *Fig. 1* it is emphasized that usually the difference of the desired signal and the actual output signal (the error signal) drives the adaptation process.

Recently SZTIPÁNOVITS [2,3] has proposed new components for general adaptive FIR filter structures:

- resonator based digital filter (RBDF) is used as a time-varying filter component,
- single or multilayer neural network is used for adaptation.

Both components have promising features:

- the RBDF is structurally passive, provides minimum roundoff noise, can suppress zero-input limit cycles, etc.[4] In the implementational point of view it is a highly parallel structure which provides the adaptive system with substantial advantages,
- using neural networks for adaptation may solve problems, where usually no (or at least limited) *a priori* structural information of the system, signal, or process of interest is available. The necessary structural knowledge can be taught during the training process.

On the other hand, both components have drawbacks as well:

- the RBDF structure is suitable for both FIR and IIR filtering problems but its application in an adaptive IIR context is not straightforward because of stability problems,

- the neural network can exhibit poor convergence properties during training.

The purpose of this paper is twofold:

- to improve the performance of the adaptive filter proposed by Sztipánovits (especially during the training phase),
- to extend the applicability to adaptive IIR applications.

The Time-varying Filter Section and the Adaptive FIR Filter

The proposed time-varying filter is a resonator based digital filter structure which has several structural and implementational advantages. The filter structure is based on some concepts of the observer theory. The key part of the filter is a conceptual state variable model of the input signal, where the state variables are the components of a discrete transformation (*Fig. 2*).

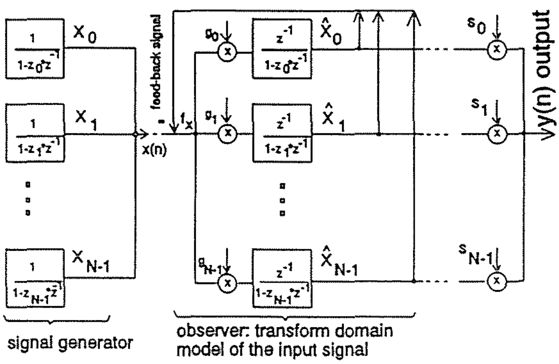

Fig. 2. Concept and structure of the resonator based digital filter

Parameters for the recursive Discrete Fourier Transformation:

$$z_m = e^{j\frac{2\pi}{N}m};$$

$$g_m = \frac{1}{N}e^{j\frac{2\pi}{N}m};$$

$$m = 0, 1, \ldots, N-1$$

Arbitrary discrete orthogonal transformation can be used but in most of the cases we prefer to use Fourier transformation, and that will be used throughout this paper. Discrete Fourier transformation is performed if we use N parallel first order complex resonators, where:

- the poles of the resonators (z_i =exp($2\pi i/N$), $i = 0, 1, 2, \ldots, N-1$ are the Nth roots of 1,

- the input weighting factors are proportional to the corresponding poles ($g_i = z_i/N, \quad i = 0, 1, 2, \ldots, N-1$).

In that case the outputs of the observer loop ($\hat{X}_0$, $\hat{X}_1$,...,$\hat{X}_{N-1}$) gives the recursive discrete Fourier transform (RDFT) of the last N samples of the input time series ($x(n-1), x(n-2), \ldots, x(n-N)$) [4]. The sum of the outputs of the observer loop ($\hat{X}_0+\hat{X}_1+\ldots+\hat{X}_{N-1}$) gives a one-step prediction of the input signal ($\hat{x}(n)$) based on the last N samples.

The outputs of the transformation loop always give a correct DFT of the last N input samples, but the one-step prediction is not always errorless. If the input signal is periodic, having time period of N (i.e. having harmonic components of the resonator pole frequencies of the loop), the $x(n)$ signal can be composed of the transformation components without error. In that case there is no difference between the input $x(n)$ and the feedback signal $fb(n)$, therefore the feedback difference signal $f_x(n) = x(n) - fb(n)$ is zero. The resonators work with zero input and their outputs will provide sinusoidal signals of the free running frequencies. It will be referred to as the case of correct signal model. If the input signal has different harmonics which are not embodied in the resonators, the input signal ($x(n)$) cannot be predicted from the N point time series, therefore cannot be reconstructed from the N point DFT of this record. In that case the feedback signal can follow the input with some error only, the feedback difference signal is not zero, it will tune the resonators to give frequencies different from the free-running ones. It will be referred to as the case of not correct signal model.

On the basis of the observer in *Fig.2* the realization of FIR and IIR filters is also possible. The transfer characteristics of the filter:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum\limits_{i=0}^{N-1} s_i \frac{g_i z^{-1}}{1 - z_i z^{-1}}}{1 + \sum\limits_{i=0}^{N-1} \frac{g_i z^{-1}}{1 - z_i z^{-1}}}, \tag{1}$$

For FIR filters, the transformation loop is fixed, only the output weighting factors have to be set for the proper FIR filter transfer characteristics. In that case the observer is dead-beat in N steps, where N denotes the transformation size. Because the resonator pole positions are on the unit circle the resonators work at the limit of stability. But the global observer system will be stable due to the global feedback of -1, which results in one zero for every pole at the same position and so stabilizes the transfer function. Because the zeros are provided by the same resonator parameter ($z_i = 0, 1, \ldots, N-1$) through the feedback the poles and zeros will effectively cancel the instability.

For IIR filters fixed recursive DFT cannot be used because the pole positions of the resonators and the input weighting coefficients ($g_i = 0, ..., N-1$) will depend on the IIR filter pole-zero arrangement to be realized. In order to achieve good sensitivity properties the poles are on the unit circle and the input weighting factors are chosen $g_i = r_i z_i$, where r_i are real numbers and $r_0 + r_1 + ...r_{N-1} - 1$ [4].

The adaptive filter structure proposed by Sztipánovits is shown in *Fig. 3.*

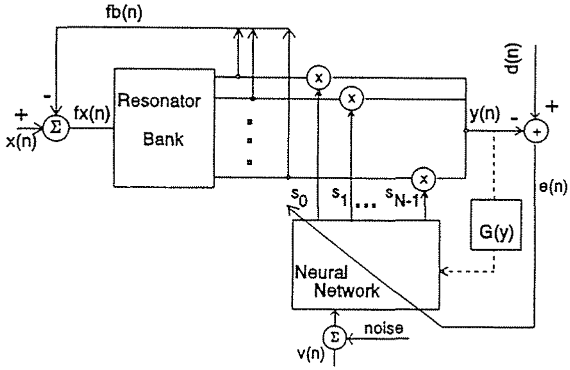

Fig. 3. Adaptive FIR filter structure

Only the output weighting factors of a fixed RDFT loop are adapted therefore only FIR filters can be implemented by this model. Two architectures were investigated with and without feedback $G(y)$ and it was shown that this type of feedback transformation helps to improve the noise rejection properties of the filter [2,3].

Improvement of the Convergence Properties

Using neural networks usually results in convergence rate problems during the training of the system. In this section some possibilities of the improvement of the training speed of the neural network are investigated.

The size (number of layers, number of nodes, number of connections) of the neural network is in strong correlation with the convergence rate; the higher is the complexity of the net the slower is the training rate. So it would be very advantageous to decrease the size of the neural network used. But there is a lower limit of the size — the neural network must be complex enough to model the complexity of the mapping of the input signal ($x(n)$) onto the output one ($d(n)$). The problem is that we usually do not know this complexity, but even if we know it, the complexity of the neural network should be higher for an efficient training. So we usually use a neural network which is redundant and thought to be complex enough. Therefore the direct decrease of the size of the neural network is not a feasible possibility.

But we can make use of the parallel structure of the filter; we can achieve the complexity needed by using independent smaller parallel subnetworks for every branch of the resonator bank. It is correct when our signal model is a correct one, in that case the signal components in the transform domain are independent of each other (at every output we have one component of the orthogonal discrete Fourier transform). By using parallel neural networks for every branch the complexity of the total neural network system remained nearly the same (the number of layers and nodes remained, only some connections were cancelled) but the training rate is improved, because smaller independent networks trained.

A second opportunity to improve the convergence properties is a consequence of the fact that the filter has parallel branches and the neural network is divided into parallel independent nets as well. Using backpropagation as training algorithm we try to optimize a scalar performance function, in most of the cases we use:

$$E = \frac{1}{M}\sum_{n=1}^{M}[d(n) - y(n)]^2. \tag{2}$$

We use the steepest descent gradient method; evaluating the gradient of the scalar cost function according to the parameter vector of the neural network. If $W^i_{p,q,r}$ is the input weight assigned to the pth input of the qth node in the rth layer of the ith subnetwork, for the sake of notational simplicity we will use W^i_t in referring to this single weight.

$$\frac{\delta E}{\delta W^i_t} = \frac{1}{M}\sum_{n=1}^{M} 2[d(n) - y(n)]\frac{\delta y(n)}{\delta W^i_t}. \tag{3}$$

Taking into account that we evaluated the transform components of the incoming signal:

$$y(n) = \sum_{i=0}^{N-1} Y_i(n) = \sum_{i=0}^{N-1}[s_i(n)X_i(n)]. \tag{4}$$

It follows that:

$$\frac{\delta E}{\delta W^i_t} = \frac{1}{M}\sum_{n=1}^{M} 2\left[d(n) - \sum_{i=0}^{N-1} s_i X_i(n)\right] X_i(n)\frac{\delta s_i}{\delta W^i_t} \tag{5}$$

because the transform domain components X_i of $x(n)$ do not depend on the neural network parameters. Assuming that the signal concept of $x(n)$ is correct the components X_i are independent of each other since the transformation used is an orthogonal one. Therefore only the ideal output signal $d(n)$ builds connections among the components of the gradient vector. Let us take the transform domain components of $d(n)$ in the same transformation as we did for $x(n)$

$$d(n) = \sum_{i=0}^{N-1} D_i(n) \tag{6}$$

$$E = \frac{1}{M}\sum_{n=1}^{M}\left[\sum_{i=0}^{N-1}[D_i(n) - s_i X_i(n)]\right]^2 = \frac{1}{M}\sum_{n=1}^{M}\left[\sum_{i=0}^{N-1} E_i(n)\right]^2, \tag{7}$$

where

$$E_i(n) = D_i(n) - s_i X_i(n) \qquad i = 0, \ldots, N-1 \tag{8}$$

are the component errors in the transform domain. Assuming that the orthogonal transform of $d(n)$ gives a correct model of $d(n)$ as well:

$$\begin{aligned}\frac{\delta E}{\delta W_t^i} &= \frac{1}{M}\sum_{n=1}^{M} 2[D_i(n) - s_i(n)X_i(n)]X_i(n)\frac{\delta s_i(n)}{\delta W_t^i} \\ &= \frac{1}{M}\sum_{n=1}^{M} 2E_i(n) - X_i(n)\frac{\delta s_i(n)}{\delta W_t^i}\end{aligned} \tag{9}$$

If the above assumptions are valid the component errors could be used to train the independent neural subnetworks that will increase the convergence rate. (It can be considered as a type of the transform domain adaptation [6].)

Unfortunately if X_i and $D_i(i = 0, 1, \ldots, N-1)$ are not correct signal models of $x(n)$ and $d(n)$, respectively, the transform domain error evaluation gives a lower limit of the achievable accuracy. The goal of transform domain optimization is to modify X_i until $X_i = D_i$, but D_i cannot characterize $d(n)$ perfectly. You can measure how the transformations characterize the signals by measuring the power of the feedback difference signals. Therefore a combination of the errors seems to be effective. Define the ratio of the power of the output error and the transformation errors :

$$c = \frac{\overline{f_x} + \overline{f_d}}{\overline{(d-y)}} \tag{10}$$

where f_x and f_d are the feedback difference errors of the transformations of signals $x(n)$ and $d(n)$, respectively and the overbar notes the power of a signal. The following combination of the two errors was used in simulations:

$$e_0(n) = d(n) - y(n), \qquad E_i(n) = D_i(n) - s_i X_i(n) \tag{11}$$

$$E_i' = \frac{c}{1+c} e_0 + \frac{c}{1+c} E_i. \tag{12}$$

The system using the combination of the output error and the transform domain component error is shown in *Fig. 4.* For the sake of simplicity the error evaluation method for only one branch (*i*th branch) is shown. The shaded blocks perform recursive Fourier transformation, the dotted block is used for evaluating the combined error for training algorithm according to *Eqs.* (10)–(12).

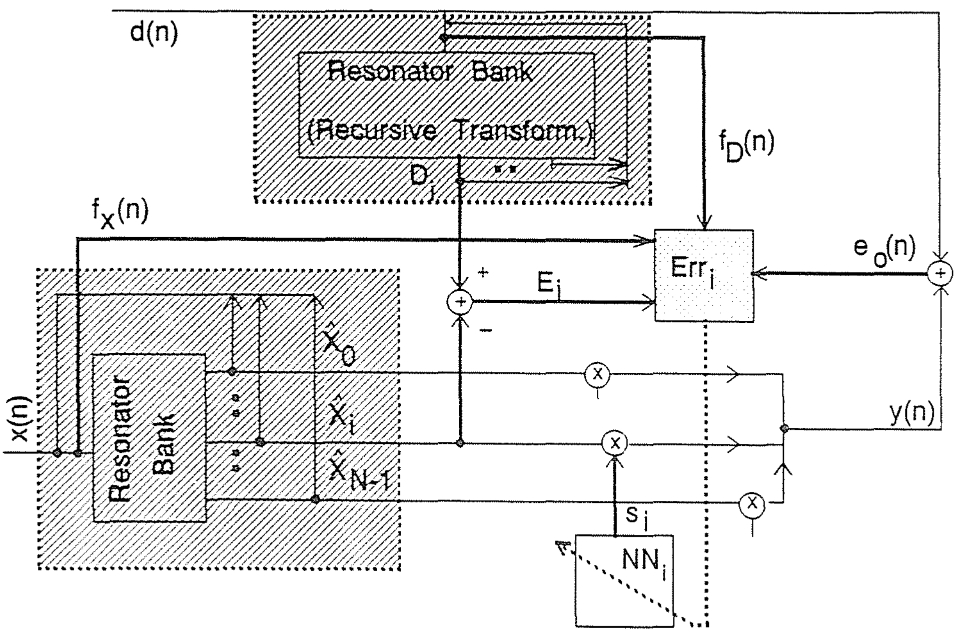

Fig. 4. Combined use of transform domain error and time domain error

Simulation results show that this system improves the learning rate, especially when the signal transform components give correct or near correct models of the signals. In *Fig.5* the result of modelling a 5th order FIR low-pass filter is shown. The cut-off frequency of the filter is $f_s/4$ (f_s is the sampling frequency), the input signal is the combination of sine waves of resonator pole position frequencies and white noise. The decrease of output relative error during the training is shown using the output error only (solid line) and the combination of output and transform domain errors (dotted line). The combined error results in a significant improvement of convergence rate.

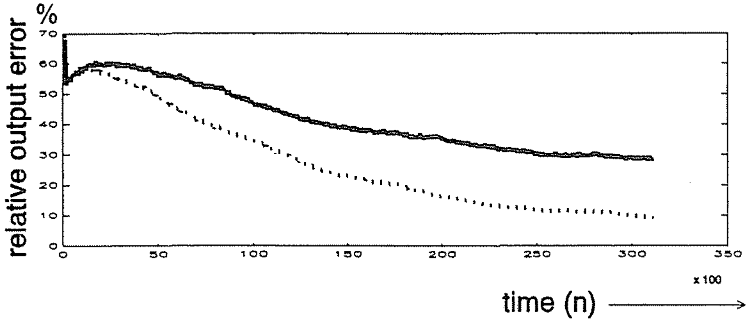

Fig. 5. Convergence of training algorithm using different errors

Extending the Adaptive Structure to IIR Filters

The resonators of the RBDF structure work at the limit of instability. The negative feedback of -1 stabilizes the loop. But for IIR filters the parameters of the loop especially the resonator pole positions should be changed as well. Because the adaptation algorithm of the pole positions includes the same pole positions, in its transfer function but it is out of the negative feedback loop, unfortunately the adaptation process will be unstable.

One should modify the structure such that any IIR filter uses the same stable loop for transformation and the pole positions be unchanged. In that case new free parameters have to be introduced because the denominator of the transfer function has to be defined. (An IIR filter of order N has $2-N-1$ free parameters, compared to a FIR filter of the same order which has N free parameters only. On the other hand, the transformation is dead beat so it provides only a finite memory.) A possible solution could be found for that problem if we use the same recursive Discrete Fourier Transformation loop as in *Fig.1*, but we modify the output weighting by using first order IIR filters (in form of $b_1/(1+a_2z^{-1})$) instead of the complex numbers (s_i;

$i = 0, 1, \ldots, N-1$). The transfer function of the filter in that case:

$$H(z) = \frac{\sum_{i=0}^{N-1} \frac{b_{1i}}{1+a_{2i}z^{-1}} \frac{g_i z^{-1}}{1-z_i z^{-1}}}{1 + \sum_{i=0}^{N-1} \frac{g_i z^{-1}}{1-z_i z^{-1}}}, \tag{13}$$

$$H(z) = \frac{\sum_{i=0}^{N-1} [b_{1i} g_i z^{-1} \prod_{k=0, k\neq i}^{N-1} (1+a_{2i}z^{-1})(1-z_k z^{-1})]}{\prod_{k=0}^{N-1} (1+a_{2k}z^{-1})(1-z_k z^{-1}) + \sum_{i=0}^{N-1} g_i z^{-1}(1+a_{2i}z^{-1}) \prod_{k=0, k\neq i}^{N-1} (1+a_{2k})(1-z_k z^{-1})]} \tag{14}$$

The adaptation algorithm can set the parameters of the output weighting filters with some constraints to have a real-valued output function for $y(n)$. (Because the loop is similar to the FIR case, Discrete Fourier Transformation components will drive the output first order filters, so the input of the kth and $(N-k)$th filters are complex conjugate pairs; $X_k = X^*_{N-k}$. If we use in these branches first order filters having complex conjugate coefficients, the resulting output will be real valued. So $b_{1i} = b^*_{1(N-i)}$; $a_{2i} = a^*_{2(N-i)}$; $i = 0, \ldots, N-1$.

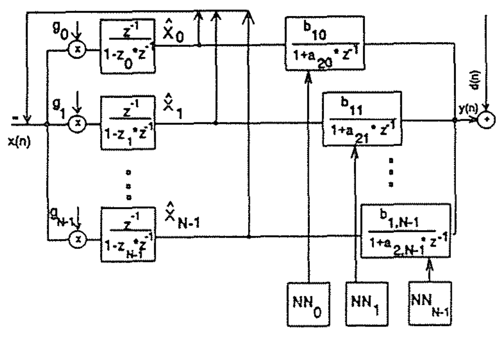

Fig. 6. Adaptive IIR filter structure

For the sake of simplicity in *Fig. 6* the transformation of the desired signal ($d(n)$) and the error evaluation process are not shown, they are the same as in *Fig. 4*.

The above adaptive IIR structure with the generalized error evaluation process was used in several computer simulations. In *Fig. 7* the ideal output signal ($d(n)$: solid line) and the actual output signal ($y(n)$: dotted line) are shown. The input signal ($x(n)$) was composed of some sinusoidal signals (some of them are in resonator positions others are between resonator positions) and a stochastic error term. The ideal output ($d(n)$) was a filtered version of the input signal by using two second order IIR low-pass filters. The filter parameters providing the ideal signal were changed periodically. The adaptive system had 16 parallel resonators (therefore 16 order IIR filter could be formed using that filter) and it was trained to follow these changes. In *Fig. 7* a part of the training is shown, the time window: $9300 < n < 9400$. The filter characteristics providing the ideal output is changed at $n = 9325$. It is clear that the adaptive filter could follow the change after a short transient.

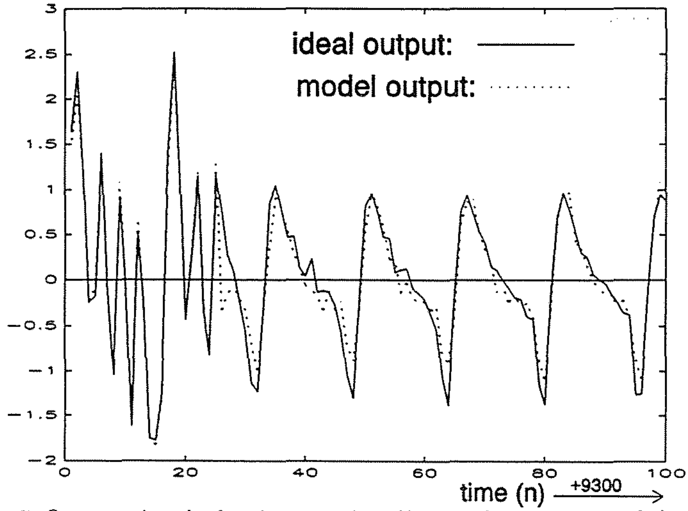

Fig. 7. Output signal of a time varying filter and the output of the model

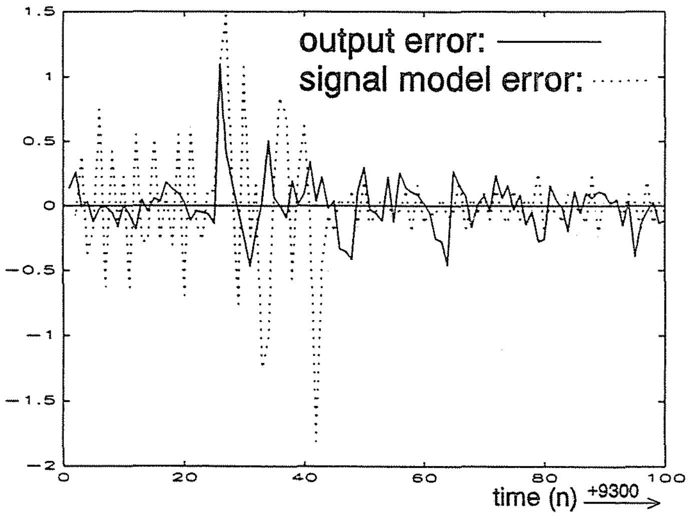

Fig. 8. Output error and feedback error signals

In *Fig. 8* the output error ($d(n) - y(n)$: solid line) and the feedback error of the transformation of the ideal signal ($f_d(n)$: dotted line) are shown in the same time window. You can see the transient after changing the filter providing the ideal output signal. In the period $9300 < n < 9325$ the low-pass filter having higher cut-off frequency is used, the output error is of lower level than the transformation error, because the high frequency noise could not be properly modelled by the transformation. In this part the output error is used dominantly for training. In the second part ($9325 < n < 9400$) when lower cut-off frequency filter is used, the output error is dominant over the error of transformation. In this part the transform domain component errors are used dominantly for training.

Conclusions

The promising new components for general adaptive FIR filter structures, proposed by Sztipánovits, have conceptual and implementational advantages. The proposed structure has drawbacks e.g. the poor convergence of the neural network during learning. In this paper some new methods for improving the convergence of the neural network were investigated. A structural modification of the resonator based filter was suggested as well to extend the applicability of the structure to adaptive IIR filters.

References

1. SHYNK, J. J.: Adaptive IIR filtering, *IEEE ASSP Magazine*, Vol. 6, pp. 4–21, 1989.
2. SZTIPÁNOVITS, J.: Adaptive Processing with Neural Controlled Resonator-Banks, *IEEE Trans. on Circuits and Systems,* Vol. CAS–37, No. 11, pp. 1436–1440, 1990.
3. SZTIPÁNOVITS, J.: Dynamic Backpropagation Algorithm for Neural Network Controlled Resonator-Banks, *IEEE Trans. on Circuits and Systems*, Vol. CAS–39, No. 2, 1992.
4. PÉCELI, G.: Resonator-Based Digital Filters, *IEEE Trans. on Circuits and Systems*, Vol. CAS–36. No. 1, pp. 156–159, 1989.
5. NARENDRA, K. S. – PARTHASARATHY, K.: Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks, *IEEE Trans. on Neural Networks* Vol. 2. pp. 252–262, 1991.
6. MARSHALL, D. F. – JENKINS, W. K. – MURPHY, J. J.: The Use of Orthogonal Transforms for Improving Performance of Adaptive Filters, *IEEE Trans. on Circuits and Systems* Vol. CAS–36, No. 4, pp. 474–483, 1989.