Neurocomputing 440 (2021) 220–229

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A novel pipelined neural FIR architecture for nonlinear adaptive filter

Dinh Cong Le [a,b], Jiashu Zhang [a,*], Yanjie Pang [a]

[a] Sichuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu 610031, PR China
[b] School of Engineering and Technology, Vinh University, Viet Nam

ARTICLE INFO

Article history:
Received 8 September 2019
Revised 20 January 2020
Accepted 17 November 2020
Available online 1 December 2020
Communicated by Hong Xia

Keywords:
Neural networks
Nonlinear adaptive filter
Pipelined architecture

ABSTRACT

This paper presents a novel adaptive pipelined neural finite impulse response (PNFIR) filter for nonlinear signal processing. Unlike traditional pipelined recurrent neural network (PRNN), each module of the PNFIR filter is a simple architecture that includes a standard FIR filter followed by a nonlinear activation function. The complete design of proposed filter includes two subsections: The nonlinear part consists of neural FIR (NFIR) modules which is interconnected in a chained form and simultaneously executed in a parallel fashion; the linear subsection is a tapped-delay-line (TDL) linear combiner. Based on convex combination architecture, the adaptive algorithm derived from the gradient descent approach is utilized to update weights of the nonlinear and linear parts. Moreover, the analysis of stability conditions and computational complexity is also presented. Numerous simulation experimental results on nonlinear dynamic systems identification, speech signal and chaotic time series prediction show that the proposed PNFIR filter has simpler architecture, faster convergence rate, and lower computation complexity than the PRNN and joint process filter using pipelined feedforward second-order Volterra architecture (JPPSOV).

1. Introduction

Many physical signals encountered in practice are usually non-stationary and nonlinear (e.g., nonlinear colored signals, speech signals, chaotic signals). The processing task of such signals is increasingly difficult, since the mathematical models to describe them are often complicated to derive. Linear adaptive techniques based on the linear model do not perform well for such signals due to their nonlinearity nature. During the past decades, nonlinear adaptive filter drew attention of many researchers due to its superior performance when compared with the linear adaptive filter in many practical applications. Different nonlinear filter design approaches are suggested to model nonlinear systems [1,2]. However, to find a unified theory for modeling and characterizing them accurately is hard. So far, the most widely used nonlinear filters mainly have the following two categories: polynomial filters (PF) [1] and neural networks (NN) [2].

It is well known that NN are attractive owing to their learning and generalization abilities. They emerge as an effective and powerful tool to approximate nonlinear functions and expand signal processing horizons. Various types of NN such as radial basis function (RBF) [2–4], multilayer perceptron (MLP) [2,5], and recurrent neural network (RNN) [2,6] can be found in the literature to the nonlinear system identification [7,8], channel equalization [9–12], speech prediction [13–15], active noise control [16,17].

The adaptive nonlinear filters based on NN, especially RNN, exhibit a wide range of dynamics thanks to its recurrent architectures [6]. This advantage can enable RNNs to accurately model nonlinear dynamic systems, which are suitable for many practical applications. However, the major drawbacks of RNN filters are the heavy computational burden and slow convergence rate. To cope with these problems, a computationally efficient nonlinear predictor using the pipelined RNN (PRNN) architecture was proposed by Haykin and Li [18,19]. The application of PRNN have been already presented in the literature to prediction of speech, identification of nonlinear dynamic systems, communication systems, e.g,. see [20–23].

Volterra filter (VF) plays a very important role in polynomial filtering study due to its simple architecture and nonlinear processing capability. Moreover, the global convergence of VF is guaranteed because it can be considered as a straightforward generalization of linear adaptive filters. Therefore, the Volterra series-based model has been widely used in the applications such as the nonlinear system identification [24], channel equalizer [25], image processing [26], active noise control [27], acoustic echo cancellation [28], just to mention a few.

[*] Corresponding author.
E-mail address: jszhang@home.swjtu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.neucom.2020.11.036
0925-2312/© 2020 Elsevier B.V. All rights reserved.

However, since the VF requires a large number of coefficients to be able to accurately model nonlinear systems, its computational complexity becomes heavy burden. One main reason is that the coefficient of the Volterra filter increases geometrically as its memory length and/or order goes up. To mitigate this disadvantage, many researchers have developed low-complexity Volterra filters. Koh and Powers proposed an iterative factorization technique [29]; Lou et al. used a method based on multi-memory decomposition (MMD) [30]; Panicker and Mathews presented an adaptive parallel-cascade Volterra filter using the normalized least mean square (NLMS) [31]; Banat proposed a pipelined Volterra filter utilizing the recursive equation [32]. Especially, an efficient method, which using pipelined SOV architecture was proposed by Zhao and Zhang [33], is an adaptive joint process filter using pipelined feedforward second-order Volterra architecture (JPPSOV). Compared with the direct-form SOV, the JPPSOV exhibits better performance and lower computational complexity.

The VF and RNN using pipelined architecture can significantly reduce the computational complexity. However, because of their structural characteristics, the computational burden still prohibits many applications in practice. Moreover, these filters are slow in converging to the optimal state. Therefore, in order to offer a nonlinear filter which has simple architecture, low computational complexity and fast convergence rate, we propose a novel pipelined neural FIR (PNFIR) in this paper.

The rest of the paper is organized in the following manner. In Section 2, we present a motivation for the proposed filter. In Section 3, a brief summary of the neural FIR filter is given. Section 4 and 5 present the PNFIR architecture and adaptive algorithm, respectively. The analysis of stability conditions and computational complexity of the PNFIR filter is given in Sections 6 and 7. Section 8 illustrates the effectiveness of the proposed filter by comparing it with JPPSOV and PRNN. Finally, the conclusion is drawn in Section 9.

A. Notation

In this paper, vectors are represented by capital letters and scalars are denoted by lowercase letters. Time-varying vectors and scalars show discrete-time index in brackets. A regression vector is expressed as $X(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$, where N is the overall vector length and $x(n\text{-}i)$ is entry at the time instant n-i. All vectors are expressed as column vectors.

2. Motivation

By embedding memory into a static perceptron, Mandic et al. developed a series of neural FIR filters (dynamical perceptron) for nonlinear system identification and non-stationary signal processing [34–38].These nonlinear filters have low computational complexity, simple architecture, and effective nonlinear processing capability for small-scale applications which is almost equal to traditional neural networks'. However, its nonlinear processing capability is only based on the static nonlinearity of perceptron. Thus, when the task of interest is a difficult one, its nonlinear processing capability is limited. Indeed, this is the main reason that hinders the applications of the NFIR filter in practice.

In this paper, we present novel PNFIR architecture for nonlinear adaptive filter. Based on the nested nonlinearity of cascaded modules, the overall input–output relation of the nonlinear subsection is similar to that found in a multilayer perceptron. Consequently, the capability of nonlinear signal processing of the filter can be improved. In addition, thanks to the convex combination of the output of the TDL linear combiner and the output of the standard FIR filters, the PNFIR would adequately employ advantages of the nonlinear filter and characteristics of the linear filter to improve its performance. The main differences of the PNFIR compared with the previous pipelined architectures include the following:

- Each module in the nonlinear subsection of the PNFIR is a simple architecture that includes a standard FIR filter followed by a nonlinear activation function.
- The weights update method of the linear and nonlinear parts is based on convex combination architecture.

3. Brief of neural FIR (NFIR) filter

In order to tackle the disadvantages of the neural network (heavy computational complexity, complicated architecture, local minimum point...), Mandic et al. presented a series of adaptive neural FIR (NFIR) filters and realized them as a dynamical feedforward neuron based on the FIR architecture [34–38]. These filters have the simple architecture, low computational complexity, and theirs nonlinear processing capability for small-scale applications is close to that of traditional neural networks. An NFIR filter is depicted in Fig. 1. It consists of a standard FIR filter followed by a nonlinear activation function $f(.)$, typically a sigmoid.

Let $X(n)$ denotes the N input signal vector of the filter is given as follows

$$X(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T \tag{1}$$

The weight vector of this filter is defined as

$$W(n) = [w_1(n), w_2(n), \cdots, w_N(n)]^T \tag{2}$$

The cost function of the nonlinear gradient descent (NGD) algorithm for a structure shown in Fig. 1 is expressed by

$$J(n) = e^2(n) \tag{3}$$

where $e(n) = d(n) - y(n) = d(n) - f\left\{W^T(n)X(n)\right\}$ is the instantaneous output error; $d(n)$ is the desired response of the filter; $f\{.\}$ is the nonlinear activation function; and $(.)^T$ is the vector transpose. The gradient of $J(n)$ with respect to the weight vector $W(n)$ is given by

$$\begin{aligned} \nabla_{w(n)} J(n) = & \; 2e(n)\frac{\partial\left(d(n) - f\left\{W^T(n)X(n)\right\}\right)}{\partial W(n)} \\ = & \; -2e(n)f'\left\{W^T(n)X(n)\right\}X(n) \end{aligned} \tag{4}$$

By using the gradient estimation algorithm along the negative of the gradient of cost function with respect to weight vector $W(n)$, the weights updating equation is given as follows

$$\begin{aligned} W(n+1) = & \; W(n) - \tfrac{\eta}{2}\Delta W \\ = & \; W(n) + \eta e(n) f'\left\{W^T(n)X(n)\right\}X(n) \end{aligned} \tag{5}$$

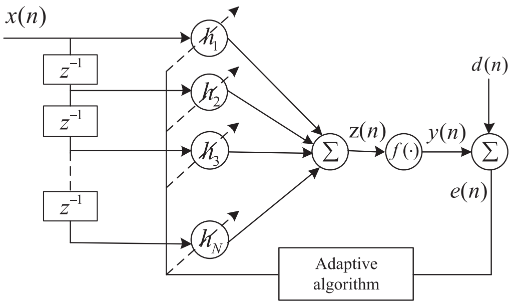

Fig. 1. Diagram of adaptive neural FIR filter.

where η denotes the learning rate of the algorithm.

Based on the characteristics of the activation function, it can be seen that when the external signal input is relatively small, it behaves as an approximate linearity; but when the input signal is large, its approximation tends to the limits 0 and 1 or -1 and 1 (depending upon the choice of activation function), which leads to the distortion at the output. Therefore, to ensure the effectiveness of the NFIR filter, the amplitude of external signal input should be selected appropriately.

4. The structure of PNFIR filter

Haykin and Li had proposed a nonlinear adaptive predictor based on pipelined architecture to overcome the computational complexity of the recurrent neural network (RNN) [18,19]. Following their work, many types of nonlinear models based on pipelined architecture have been proposed and successfully applied in practice [39–41]. Keeping the view of pipelined architecture, a novel PNFIR filter is proposed to improve performance of the NFIR nonlinear filter. The operation of the PNFIR is based on modularity principle.

The proposed PNFIR architecture is illustrated in Fig. 2. It consists of two subsections: nonlinear subsection is the NFIR modules performing a nonlinear mapping from the input space to an intermediate space, and linear subsection is a tapped-delay-line (TDL) filter performing a linear mapping from the intermediate space to the output space. Its output signal is a convex combination of the output of TDL linear combiner and output of standard FIR filters. Moreover, this new structure uses an overall feedback signal (the output of the first module) which performs as the guiding principle to improve nonlinearity [19].

4.1. Nonlinear subsection

The nonlinear subsection of the proposed structures is composed of M identical NFIR modules. Inputs of each module include L external input signals and a one-unit delayed signal of the previous adjacent module output. In the case of module M, input signals consist of L external input signals and a one-unit delayed version of signal feedback from module 1. All the modules operate similarly, which have exactly the same numbers of input signal. The detailed structure of the module i is depicted in Fig. 3.

Let $X_i(n)$ denote the $L + 1$ input signal vector of module i and is given as follows

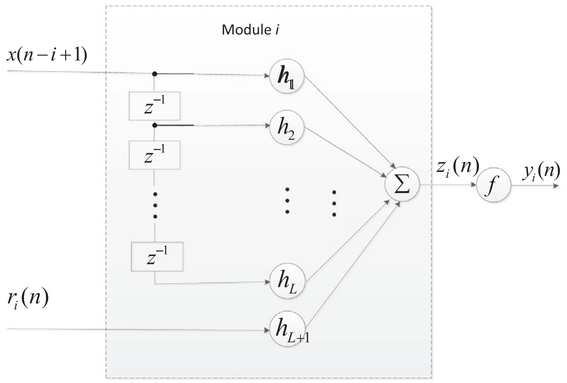

Fig. 3. Detailed construction of module i.

$$X_i(n) = [x(n-i+1), x(n-i), \dots,\ x(n-i-L+2), r_i(n)]^T, 1 \le i < M \tag{6}$$

where $r_i(n)$ is the previous adjacent module output or signal feedback. In the case of module M, it is an overall feedback from module 1.

$$r_i(n) = \begin{cases} y_1(n-1)\, i = M \\ y_{i+1}(n-1)\, i \neq M \end{cases} \tag{7}$$

The synaptic weight vector of module i in the PNFIR architecture is represented as

$$H_i(n) = \left[h_{i1}(n), h_{i2}(n), \cdots, h_{i(L+1)}(n)\right]^T,\ 1 \le i \le M \tag{8}$$

and the output $y_i(n)$ at the nth time point computed by passing $z_i(n)$ through a nonlinear activation function is written as

$$y_i(n) = f\{z_i(n)\} \tag{9}$$

where $f\{.\}$ denotes nonlinear activation function, and $z_i(n)$ is output of standard FIR filter is defined by

$$z_i(n) = H_i^T(n)X_i(n) \tag{10}$$

4.2. Linear subsection

In linear subsection, we define the weight vector $W(n)$ of TDL filter as follows

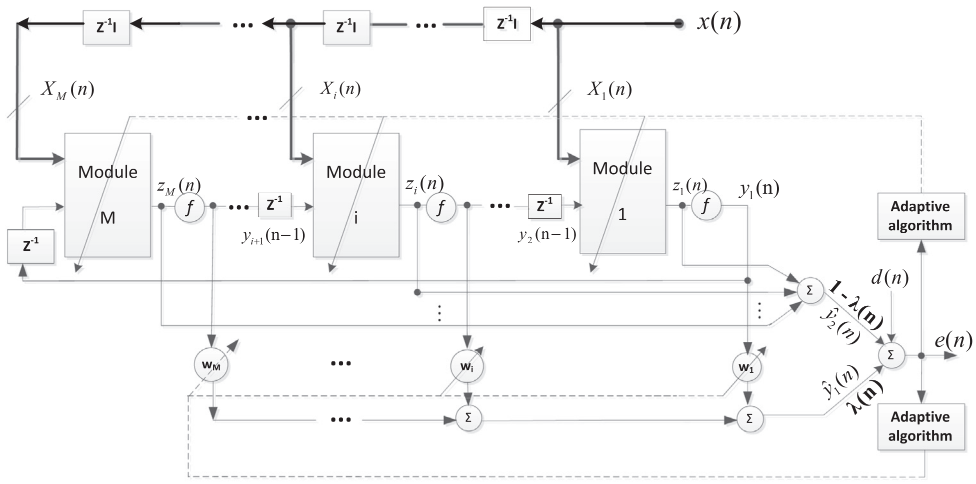

Fig. 2. The pipelined neural FIR filter.

$$W(n) = [w_1(n), w_2(n), \cdots, w_M(n)]^T \tag{11}$$

where M is the number of the modules in pipelined structure. From Fig. 2 we can easily see that the corresponding input vector of the TDL filter consists of the present output of each module, as shown by

$$Y(n) = [y_1(n), y_2(n), ..., y_M(n)]^T \tag{12}$$

Thus, the output of linear subsection can be expressed as

$$\hat{y}_1(n) = W^T(n)Y(n) \tag{13}$$

where $\hat{y}_1(n)$ can be considered the output of TDL linear combiner.

4.3. The PNFIR filter output

As shown in Fig. 2, the output of the PNFIR filter is the convex combination of the output of TDL linear combiner and standard FIR filters. Thus, its output at time n can be written as

$$\hat{y}(n) = \lambda(n)\hat{y}_1(n) + (1 - \lambda(n))\hat{y}_2(n) \tag{14}$$

where $\hat{y}_2(n)$ is sum of the outputs of standard FIR filters; $\hat{y}(n)$ is a prediction value of the actual desired sample $d(n)$; and $\lambda(n)$ is an adaptive convex combination parameter which lies between zero and one. In order to keep $\lambda(n)$ in the interval (0,1), it is defined via a nonlinear activation function as follows

$$\lambda(n) = \frac{1}{1 + \exp[-a(n)]} \tag{15}$$

where $a(n)$ is a variable parameter which can be adaptively adjusted. Convex combination parameter $\lambda(n)$ increases monotonically with $a(n)$. However, when $\lambda(n)$ is too close to 0 or 1, the $a(n)$ will stop updating. To deal with the disadvantage, the values of $a(n)$ can be limited to the interval [−4,4] [42].

5. Adaptive algorithm

Based on the architecture of the proposed PNFIR filter, the sum of the outputs of standard FIR filters can be computed as

$$\hat{y}_2(n) = \sum_{i=1}^{M} z_i(n) \tag{16}$$

Hence, the output of the PNFIR filter can be rewritten as

$$\begin{aligned}\hat{y}(n) &= \lambda(n)\hat{y}_1(n) + (1 - \lambda(n))\hat{y}_2(n)\\ &= \lambda(n)\sum_{i=1}^{M} w_i(n)y_i(n) + (1 - \lambda(n))\sum_{i=1}^{M} z_i(n)\end{aligned} \tag{17}$$

The error function $e(n)$ between the desired response $d(n)$ and estimated signal $\hat{y}(n)$ is given by

$$\begin{aligned}e(n) &= d(n) - \widehat{y}(n)\\ &= d(n) - \lambda(n)\sum_{i=1}^{M} w_i(n)y_i(n) - (1 - \lambda(n))\sum_{i=1}^{M} z_i(n)\end{aligned} \tag{18}$$

Therefore, the cost function of the PNFIR filter at the nth time point is defined by

$$\begin{aligned}J(n) &= e^2(n)\\ &= \left[d(n) - \lambda(n)\sum_{i=1}^{M} w_i(n)y_i(n) - (1 - \lambda(n))\sum_{i=1}^{M} z_i(n)\right]^2\end{aligned} \tag{19}$$

Taking as objective the minimization of the cost function $J(n)$, the parameters $H_i(n)$, $W(n)$ and $a(n)$ of the PNFIR filter are adjusted by the stochastic gradient estimation algorithm as the following formulas

$$H_i(n+1) = H_i(n) - \frac{\eta_H}{2}\nabla_{H_i(n)}J(n), \tag{20}$$

$$W(n+1) = W(n) - \frac{\eta_w}{2}\nabla_{W(n)}J(n) \tag{21}$$

$$a(n+1) = a(n) - \frac{\rho}{2}\nabla_{a(n)}J(n) \tag{22}$$

where η_H, η_W and ρ are learning rates of nonlinear subsection, linear subsection and variable parameter. $\nabla_{H_i(n)}J(n)$, $\nabla_{W(n)}J(n)$ and $\nabla_{a(n)}J(n)$ denote the gradient of $J(n)$ with respect to the weight vector $H_i(n)$, $W(n)$ and $a(n)$, respectively.

5.1. Nonlinear subsection

For the nonlinear subsection, the gradient of $J(n)$ with respect to the weight vector $H_i(n)$ is calculated as follows

$$\begin{aligned}\nabla_{H_i(n)}J(n) = &\; 2e(n)\tfrac{\partial e(n)}{\partial H_i(n)}\\ = &\; 2e(n)\frac{\partial\left(d(n)-\lambda(n)\sum_{i=1}^{M}w_i(n)y_i(n)-(1-\lambda(n))\sum_{i=1}^{M}z_i(n)\right)}{\partial H_i(n)}\end{aligned} \tag{23}$$

Substituting Eqs. (9) and (10) into Eq. (23) we can get

$$\begin{aligned}\nabla_{H_i(n)}J(n) &= -2e(n)\\ &\times\frac{\partial\left(\lambda(n)\sum_{i=1}^{M}w_i(n)f\left\{H_i^T(n)X_i(n)\right\}+(1-\lambda(n))\sum_{i=1}^{M}H_i^T(n)X_i(n)\right)}{\partial H_i(n)}\\ &= -2e(n)\left[\lambda(n)w_i(n)f'\left\{H_i^T(n)X_i(n)\right\}+(1-\lambda(n))\right]X_i(n)\end{aligned} \tag{24}$$

Therefore, the weight updating equation $H_i(n)$ of PNFIR filter in accordance with gradient descent approach is given as follows

$$\begin{aligned}H_i(n+1) &= H_i(n)\\ &+ \eta_H e(n)\left[\lambda(n)w_i(n)f'\left\{H_i^T(n)X_i(n)\right\} + (1 - \lambda(n))\right]X_i(n)\end{aligned} \tag{25}$$

5.2. Linear subsection

For the linear subsection, the gradient of $J(n)$ with respect to the weight vector $W(n)$ is obtained

$$\begin{aligned}\nabla_{W(n)}J(n) = &\; 2e(n)\tfrac{\partial e(n)}{\partial W(n)}\\ = &\; 2e(n)\frac{\partial\left(d(n)-\lambda(n)\sum_{i=1}^{M}w_i(n)y_i(n)-(1-\lambda(n))\sum_{i=1}^{M}z_i(n)\right)}{\partial W(n)}\\ = &\; 2e(n)\frac{\partial\left(d(n)-\lambda(n)W(n)Y(n)-(1-\lambda(n))\sum_{i=1}^{M}z_i(n)\right)}{\partial W(n)}\\ = &\; -2e(n)\lambda(n)Y(n)\end{aligned} \tag{26}$$

Therefore, the weight updating equation $W(n)$ of PNFIR filter in accordance with gradient descent approach is given as follows

$$W(n+1) = W(n) + \eta_w e(n)\lambda(n)Y(n) \tag{27}$$

5.3. Convex combination parameter

For the variable parameter $a(n)$, the gradient of $J(n)$ with respect to the parameter $a(n)$ is calculated as follows

$$\begin{aligned}\nabla_{a(n)}J(n) = &\; 2e(n)\tfrac{\partial e(n)}{\partial a(n)}\\ = &\; 2e(n)\frac{\partial\left(d(n)-\lambda(n)\sum_{i=1}^{M}w_i(n)y_i(n)-(1-\lambda(n))\sum_{i=1}^{M}z_i(n)\right)}{\partial a(n)}\\ = &\; -2e(n)\frac{\partial\left(\frac{1}{1+\exp[-a(n)]}\sum_{i=1}^{M}w_i(n)y_i(n)-\frac{1}{1+\exp[-a(n)]}\sum_{i=1}^{M}z_i(n)\right)}{\partial a(n)}\\ = &\; -2e(n)(\hat{y}_1(n) - \hat{y}_2(n))\lambda(n)[1 - \lambda(n)]\end{aligned} \tag{28}$$

Therefore, the update equation of the variable parameter is derived as follows

$$a(n+1) = a(n) + \rho e(n)(\hat{y}_1(n) - \hat{y}_2(n))\lambda(n)[1 - \lambda(n)] \tag{29}$$

6. The analysis of stability conditions

According to the adaptive filter theory [43], we know that the update Eqs. (25) and (27) will not ensure stability unless a strong condition is imposed on the step sizes η_H and η_w. In this section, we give the stability conditions for nonlinear and linear subsections of PNFIR filter.

6.1. Nonlinear subsection

According to the rule of the Taylor expansion [44], the instantaneous error (18) can be expanded as follows

$$e(n+1) = e(n) + \frac{\partial e(n)}{\partial H_i(n)}\Delta H_i(n) + h.o.t \tag{30}$$

where *h.o.t* denotes the higher order terms of the rest of Taylor series expansion and can be ignored. Thus, for simplicity, we only consider the first and second terms of the instantaneous squared error as follows

$$e^2(n+1) - e^2(n) = [\frac{\partial e(n)}{\partial H_i(n)}\Delta H_i(n)]^2 + 2\frac{\partial e(n)}{\partial H_i(n)}\Delta H_i(n)e(n) \tag{31}$$

From (20), the weight correction is obtained by

$$\Delta H_i(n) = H_i(n+1) - H_i(n) = \frac{\eta_H}{2}\frac{\partial J(n)}{\partial H_i(n)} \tag{32}$$

During the training process, the error change $\Delta J(n)$ can be given by

$$\Delta J(n) = J(n+1) - J(n) = e^2(n+1) - e^2(n) \tag{33}$$

Substituting Eqs. (31) and (32) into Eq. (33), we can get the following relation

$$\begin{aligned} \Delta J(n) = & \left\{[\tfrac{\partial e(n)}{\partial H_i(n)}\Delta H_i(n)]^2 + 2\tfrac{\partial e(n)}{\partial H_i(n)}\Delta H_i(n)e(n)\right\} \\ = & \Delta H_i^2(n)\left\{[\tfrac{\partial e(n)}{\partial H_i(n)}]^2 + \tfrac{2}{\Delta H_i(n)}\tfrac{\partial e(n)}{\partial H_i(n)}e(n)\right\} \\ = & \Delta H_i^2(n)\left\{[\tfrac{\partial e(n)}{\partial H_i(n)}]^2 - \tfrac{4}{\eta_H\frac{\partial J(n)}{\partial H_i(n)}}\tfrac{\partial e(n)}{\partial H_i(n)}e(n)\right\} \end{aligned} \tag{34}$$

Therefore, a loosely defined restriction of convergence condition for nonlinear subsection of the PNFIR filter ($\Delta J(n) < 0$) can be derived as follows

$$\Delta H_i^2(n)\left\{\left[\frac{\partial e(n)}{\partial H_i(n)}\right]^2 - \frac{4}{\eta_H\frac{\partial J(n)}{\partial H_i(n)}}\frac{\partial e(n)}{\partial H_i(n)}e(n)\right\} < 0 \tag{35}$$

$$\Rightarrow \eta_H\left[\frac{\partial e_i(n)}{\partial H_i(n)}\right]^2 < \frac{4}{\frac{\partial J(n)}{\partial H_i(n)}}\frac{\partial e_i(n)}{\partial H_i(n)}e_i(n) \tag{36}$$

$$\eta_H < \frac{\left[\frac{4}{\frac{\partial J(n)}{\partial H_i(n)}}\frac{\partial e(n)}{\partial H_i(n)}e(n)\right]}{\left[\frac{\partial e(n)}{\partial H_i(n)}\right]^2} = \frac{4e(n)}{\frac{\partial J(n)}{\partial H_i(n)}\frac{\partial e(n)}{\partial H_i(n)}} = \frac{2}{\left[\frac{\partial e(n)}{\partial H_i(n)}\right]^2} \tag{37}$$

So, the range of learning rate of nonlinear subsection to guarantee a convergence condition of PNFIR filter is given as follows

$$\begin{gathered} 0 < \eta_H < \frac{2}{\sup\left[\frac{\partial e(n)}{\partial H_i(n)}\right]^2} \Rightarrow \\ 0 < \eta_H < \frac{2}{\sup\left\{\left[\lambda(n)w_i(n)f'(H_i^T(n)X_i(n)) + (1-\lambda(n)\right]X_i(n)\right\}^2} \end{gathered} \tag{38}$$

where "sup" denotes a supremum operator.

6.2. Linear subsection

Using the methods of analyzing stability condition and convergence performance of NLMS algorithm under small step size assumptions [43], we can present the range of learning rate to guarantee a convergence condition of linear subsection for PNFIR filter as follows

$$0 < \eta_w < \frac{2}{\gamma_{\max}(R_{YY})} \tag{39}$$

$$R_{YY} = E[Y(n)Y^T(n)] \tag{40}$$

where $\gamma_{\max}$ denotes the max eigenvalue of matrix R_{YY}

7. Computational complexity analysis

In this section, an analysis of the computational complexity of the proposed PNFIR filter is presented. Assuming L and M are the external input signals of each module and the number of modules, respectively. The computational complexity of the PNFIR filter is required the following major operations

a) Computing the estimation error in Eq. (18) requires $(ML + M + 2)$ multiplications and $(ML + 2)$ additions
b) Computing the update of coefficients $H_i(n)$ in Eq. (25) requires $(2L + 4)$ multiplications and $(L + 1)$ additions
c) Computing the update of coefficients $W(n)$ in Eq. (27) requires $(M + 2)$ multiplications and M additions
d) Computing the update of variable parameter $a(n)$ in Eq. (29) requires 4 multiplications and 3 additions

Accordingly, the computational complexity of the PNFIR requires about $2ML + 3M + 3L + 18$ arithmetic operations. Besides, referring to the studies in [37,18], and [33], we obtain the computational complexity of the NFIR, PRNN, and JPPSOV, respectively. Therefore, a brief comparison of the number of arithmetic operations for these adaptive filters is summarized in Table 1. Where M_1 and q are the number of modules and number of output layer neurons of PRNN, respectively; L_3 is the memory length of the NFIR; L_2 and M_2 are the external input signals of each module and number of modules of JPPSOV, respectively. Table 2 offers a computational complexity comparison for a special case in the simulation. Here the parameters are selected as in example 3 of the experiment on nonlinear system identification.

Table 1
Total computational requirements of NFIR, JPPSOV, PRNN and PNFIR.

Type of filter	Arithmetic operations
NFIR	$6L_3 + 7$
PNFIR	$2M(L+1) + 3M + 3(L+1) + 18$
PRNN	$M_1q^4 + 3(M_1 + 1)$
JPPSOV	$(2M_2 + 5)[(L_2^2 + 5L_2 + 6)/2] + 6M_2 + 1$

Table 2
Computational requirements of the filters in example 3 of the simulation section.

Type of filter	Parameter values	Arithmetic operations
NFIR	$L_3 = 2$	19
	$L_3 = 30$	187
PNFIR	$L = 2$, $M = 5$	72
PRNN	$q = 3$, $M_1 = 5$	423
JPPSOV	$L_2 = 2$; $M_2 = 5$;	181

From Tables 1 And 2, we realize that the computational complexity of the proposed PNFIR is higher than that of NFIR (L = 2), however the performance obtained of NFIR filter is very low (see the next section). Increasing the memory length of the filter (such as NFIR (L = 30)) is also not a feasible solution, because it significantly increases computational complexity but achieves trivial performance. And obviously, the computational requirement of PNFIR is significantly lower than that of the JPPSOV and PRNN.

8. Simulations

In order to evaluate the performance of the proposed PNFIR filter, a number of simulations are carried out to verify the effectiveness of our proposed method including the nonlinear dynamic systems identification, speech signal and chaotic time series prediction. The performance of the PNFIR filter is compared with NFIR, PRNN and JPPSOV filters in terms of convergence speed, steady-state error and signal prediction ability.

In the simulations, the nonlinear active function is chosen to be the logistic sigmoid function as follows

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{41}$$

and initial value of convex parameter $\lambda(0)$ and variable parameter $a(0)$ are set to 1 and 0, respectively.

8.1. Identification of nonlinear dynamic system

In the simulation, we have used the unknown system models [8] as follows

$$d(n) = \frac{d(n-1)}{1 + d^2(n-1)} + x^3(n) \tag{42}$$

where $d(n)$ and $x(n)$ are the output and input signal of the unknown system, respectively.

The measurement noise is assumed to be a mean-zero white Gaussian noise that was uncorrelated with the input signal, and the input signal to measurement noise ratio (SNR) is chosen to be 30 dB. The performance of the different filters will be measured in terms of the mean square error (MSE) which is obtained by averaging over 100 independent runs.

$$MSE = 10\log10(e^2(n)) \tag{43}$$

In this experiment we use two kinds of input signals: one is random sequence with uniform distribution, and the other is colored sequences. The range of input signal is chosen as (0, 0.6). The colored signal generated by passing white noise $u(n)$ through an AR model is given by [38]

$$\begin{aligned} x(n) = {} & 1.79x(n-1) - 1.85x(n-2) + 1.27x(n-3) \\ & - 0.41x(n-4) + u(n) \end{aligned} \tag{44}$$

where $u(n)$ is mean-zero white Gaussian noise with variance one.

Example 1: Choice of the number of modules

The number of modules of the proposed PNFIR filter is an important parameter, which directly affects the performance and the computational cost of the filter. Fig. 4 shows the relation between the MSE and the number of modules for colored input signal with M increasing from 2 to 12, assuming that the external input signal L = 2, the learning rate of nonlinear and linear subsection of PNFIR filter are set to $\eta_H = 0.5$, $\eta_w = 0.01$, respectively; the parameter of convex function is chosen as $\rho = 0.01$. From Fig. 4, we can clearly see that the MSE is the lowest when M equals 5 to 8 and increases as M is greater than 8. According to the adaptive filtering theory [43], the main reason for the increase in MSE may be the overparameterized estimated problem. The choice of M controls the trade-off between computational complexity and performance, thus, we here choose M to be equal to 5.

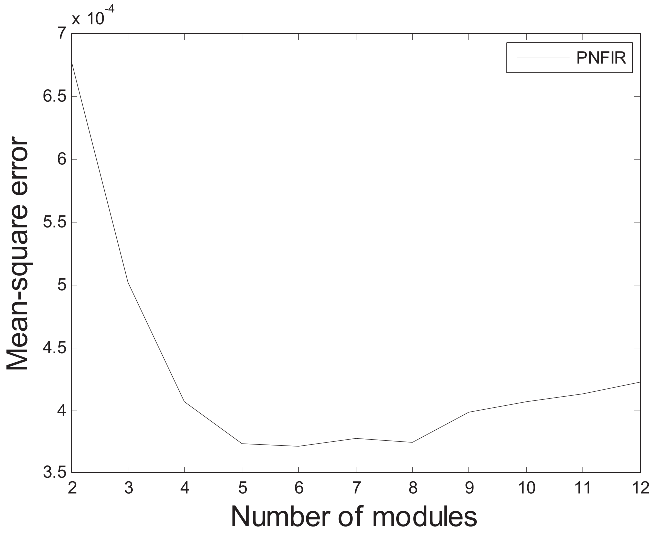

Fig. 4. Illustrating the relation between the mean-square error and the number of modules for the case of input colored noises.

Example 2: Choice of the number of external inputs

Fig. 5 describes the influence of the number of external input in the PNFIR filter on the MSE performance for colored input signal with L increasing from 1 to 10, assuming that the number of modules M = 5; the learning rate of PNFIR are set to $\eta_H = 0.6$, $\eta_w = 0.01$ and $\rho = 0.1$. For a specific level of L, Fig. 5 shows that the MSE decreases when L is chosen in the range of 2 to 4, and increases when L is greater than 4. The reason for increasing MSE (when $L > 4$) can be explained as the overparameterized estimated problem in adaptive filter processing. Similar to the parameter M, the parameter L is also chosen based on the compromise between computational complexity and performance, thus here we choose L = 2.

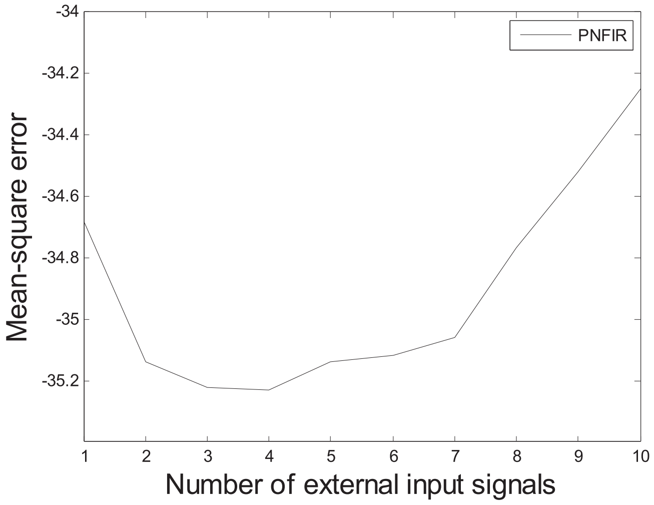

Fig. 5. Illustrating the relation between the MSE and the L for the case of input colored noises.

According to the aforementioned discussion and the results of the studies in [18,33], a summary of the parameters for PNFIR, PRNN, JPPSOV and NFIR filters is provided in Table 3.

Example 3: To further illustrate the function and advantage of the convex combination architecture, a simulation experiment have been conducted to compare the performance of the PNFIR and that of the PNFIR with $\lambda(n) = 1$(i.e., the linear and nonlinear parts of PNFIR are updated like the algorithm in [33]). In this experiment, the input is the random sequence with uniform distribution. The learning rate of the PNFIR filter and the PNFIR $(\lambda(n) = 1)$ filter are selected the same and are set to $\eta_H = 0.6$, $\eta_w = 0.01$; The learning rate of the parameter a(n) is chosen as $\rho = 0.6$. From Fig. 6, it is clear that the PNFIR architecture using convex combination algorithm achieves better performance than using the algorithm as in [33].

Example 4: In this example, we test the performance of the filters when the input is the random sequence as used in example 3. The learning rate of the PNFIR filter are set to $\eta_H = 0.6$, $\eta_w = 0.01$, and $\rho = 0.6$. The learning rate of nonlinear and linear subsection of PRNN filter are chosen as $\eta 1_H = 0.001$ and $\eta 1_w = 0.004$, respectively. The learning rate of nonlinear and linear subsection of JPPSOV filter are set to $\eta 2_H = 0.2$ and $\eta 2_w = 0.5$, respectively. The learning rate of NFIR is set to $\eta 3 = 0.031$ with $L_3 = 2$ and $\eta 3 = 0.61$ with $L_3 = 30$. Fig. 7 shows the averaged MSE performance curves for the random input signal.

Example 5: In this example, the performance of the NFIR, JPPSOV, PRNN, and PNFIR filters are investigated with the colored input signal. The learning rates of the filters are: for the PNFIR ($\eta_H = 0.5$, $\eta_w = 0.001$ and $\rho = 0.01$); for the PRNN ($\eta 1_H = 0.001$ and $\eta 1_w = 0.002$); for the JPPSOV ($\eta 2_H = 0.1$ and $\eta 2_w = 0.9$); for the NFIR ($\eta 3 = 0.1$ with $L_3 = 2$ and $\eta 3 = 0.9$ with $L_3 = 30$). Fig. 8 shows the averaged MSE performance curves for the colored input signal.

Table 3
The parameters of all the nonlinear filters.

Parameters	PNFIR	JPPSOV	PRNN	NFIR
Number of external input	2	2	2	2 and30
forgetting factor γ	–	–	0.97	–
Number of output layer neurons	1	–	3	1
Number of modules	5	5	5	–

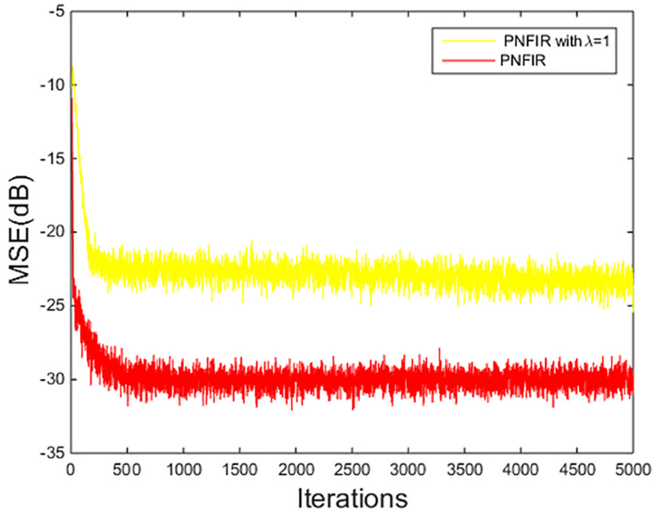

Fig. 6. Comparison of MSE of the PNFIR and PNFIR with $\lambda(n) = 1$.

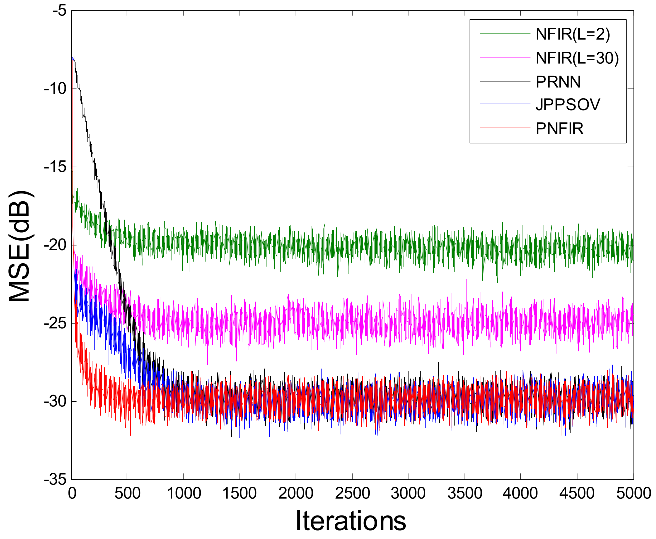

Fig. 7. Comparison of MSE for random input signal.

Figs. 7 and 8 show the MSE performance of the filters for the random input signal and the colored input signal, respectively. According to these two figures, we can see that in the same conditions, the PNFIR filter is superior to the NFIR filter. The NFIR filter can increase performance within a certain value by extending its memory length. Moreover, increasing the memory length would make its computational complexity become heavy. In addition, the PNFIR exhibits a faster convergence rate than that of the PRNN and JPPSOV filters.

8.2. Prediction of chaotic time series

To illustrate the effectiveness of proposed PNFIR filter for prediction of chaotic time series, a nonlinear signal predicting system is implemented in Fig. 9. Input signal $x(n)$ of system is the Mackey–Glass chaotic time series which are generated by a delay differential equation as follows [45]

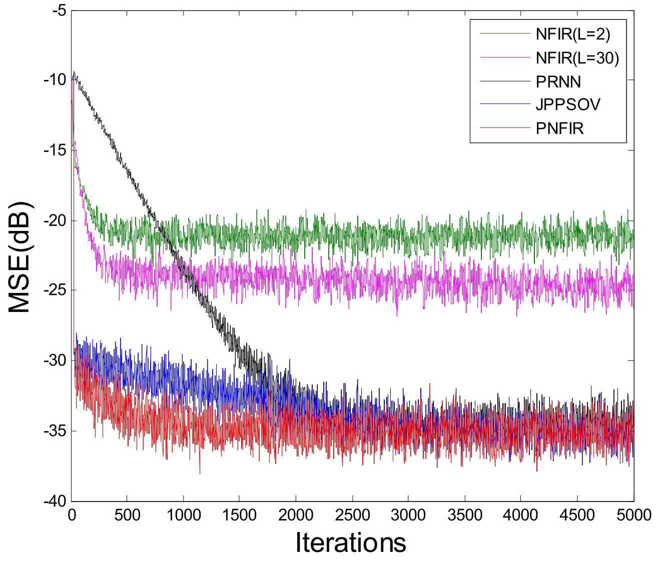

Fig. 8. Comparison of MSE for colored input signal.

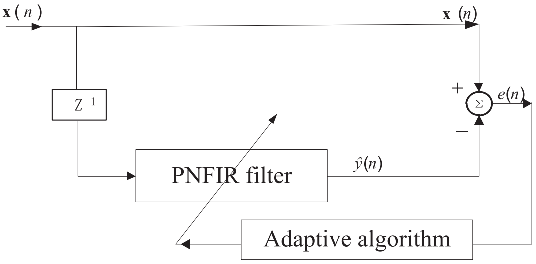

Fig. 9. Structure of PNFIR predictor.

$$\frac{dy(t)}{dt} = \frac{\alpha y(t-\tau)}{1 + y(t-\tau)^{10}} - by(t) \tag{45}$$

where τ is the delay time. The system is chaotic when $\tau > 16.8$. The data set was constructed using parameter $\alpha = 0.2$, $b = 0.1$, and $\tau = 17$.

The squared root of the MSE is used to evaluate the prediction accuracy, which is defined by

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y(t) - \hat{y}(t))^2} \tag{46}$$

where $\hat{y}(t)$ is the estimated value of $y(t)$ at time t.

In the simulation, to obtain the best result of nonlinear predictors, their parameters' values are chosen after several trials. The parameters of PNFIR are chosen as $\eta_H = 0.9$; $\eta_w = 0.9$ and $\rho = 0.9$. The learning rate of nonlinear and linear subsection of PRNN and JPPSOV filters are set to $\eta 1_H = 0.04$, $\eta 1_w = 0.9$ and $\eta 2_H = 0.9$, $\eta 2_w = 0.9$, respectively. The training chaotic time sequence length is 400. The range of input signal is chosen as (0, 0.1).

Figs. 10, 11 and Table 4 depict the results of the predictors, the predicted error and the final RMSE value of all of the nonlinear predictors, respectively. From these results, we can observe that the predicting capability of the PNFIR is better than that of JPPSOV, but slightly less than that of PRNN. However, it should be noted that in order to achieve such good predictability, PRNN has to pay a much more computational cost than PNFIR.

Therefore, these results clearly reveal that the PNFIR is capable of efficiently capturing the underlying dynamics from the chaotic time series.

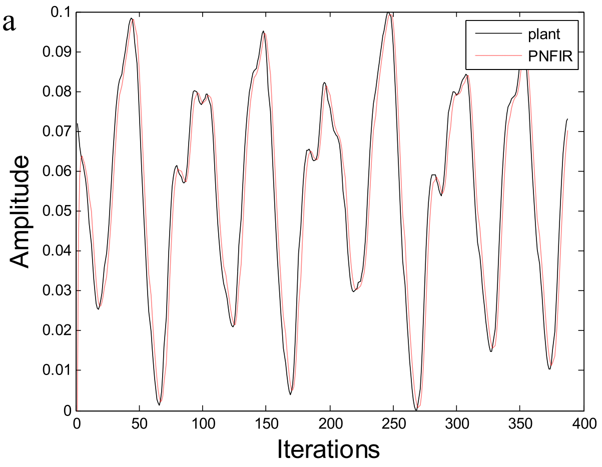

Fig. 10a. Predicted values of PNFIR.

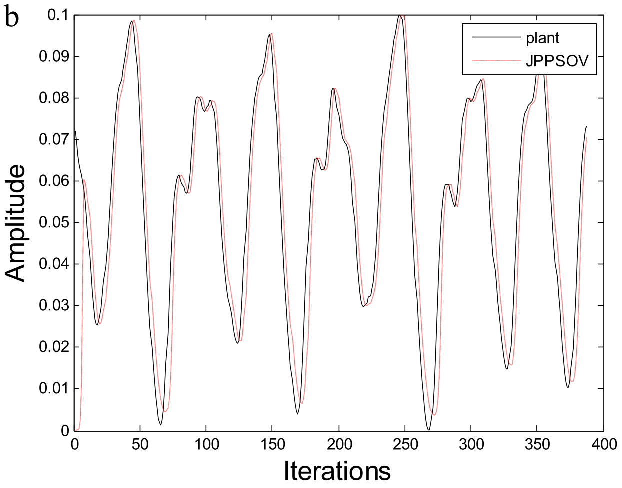

Fig. 10b. Predicted values of JPPSOV.

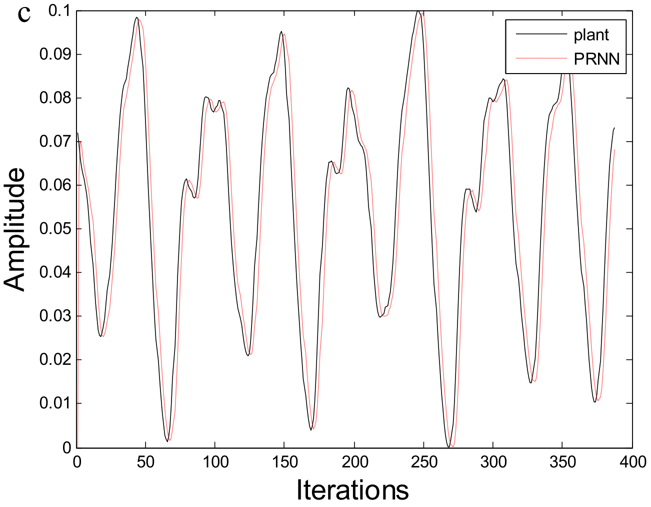

Fig. 10c. Predicted values of PRNN.

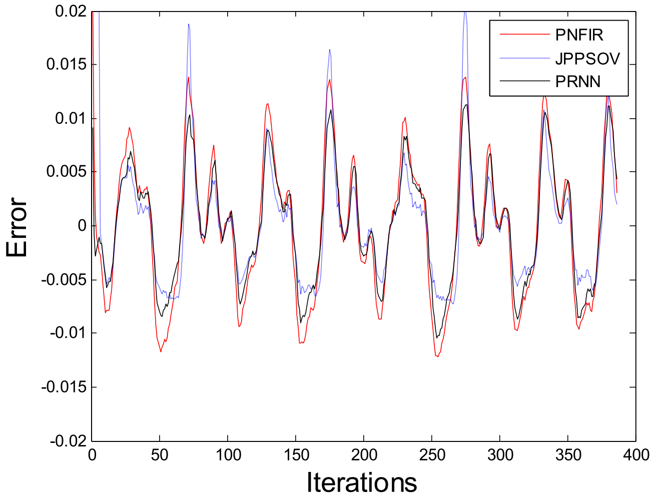

Fig. 11. Comparison of predicted errors.

Table 4
Comparing the RMSE for all the nonlinear filters.

Prediction model	RMSE
PNFIR	0.0078
PRNN	0.0064
JPPSOV	0.0090

8.3. Speech adaptive prediction

To evaluate the performance of the proposed PNFIR in the prediction of speech signals, we applied the nonlinear adaptive predictor to the case of speech signals which is made up of 4000 points. The measure used to evaluate the performance of the predictor is the one-step forward prediction gain defined as

$$R_p = 10\log 10\left(\frac{\hat{\sigma}_x^2}{\hat{\sigma}_e^2}\right) \tag{47}$$

where $\hat{\delta}_x^2$, $\hat{\delta}_e^2$ are the estimated variance of the input speech signal and error signal, respectively. They are calculated as follows

$$\hat{\sigma}_x^2 = \frac{1}{N}\sum_{i=1}^{N} x^2(i); \quad \hat{\sigma}_e^2 = \frac{1}{N}\sum_{i=1}^{N} e^2(i)$$

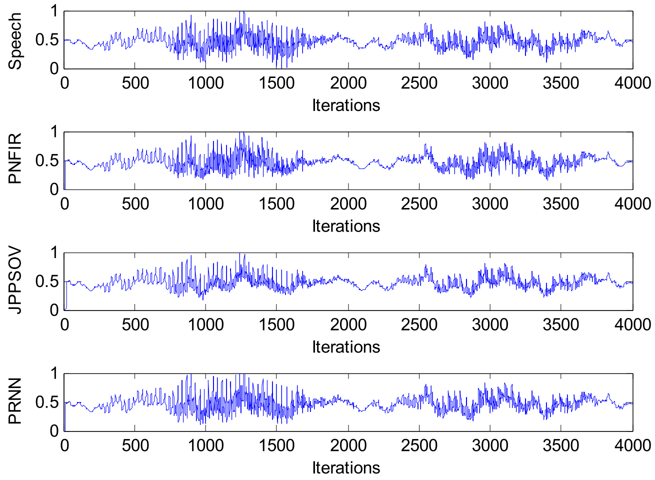

Fig. 12. Original speech signals, predicting values by PNFIR, PRNN and JPPSOV.

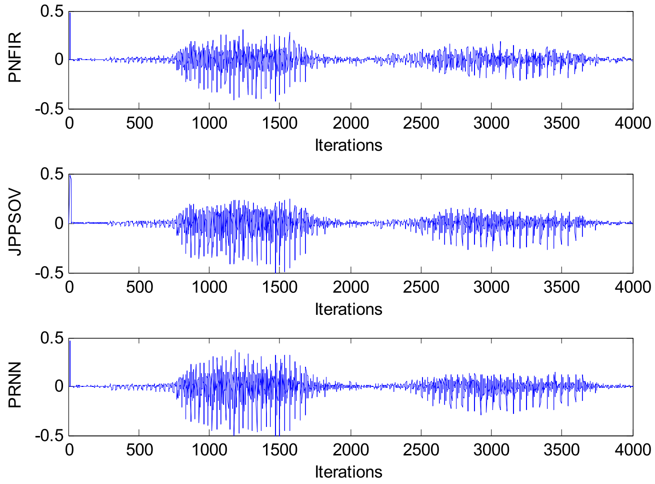

Fig. 13. Predicting error by PNFIR, PRNN and JPPSOV.

Table 5
Comparing of the R_P for nonlinear filters.

Prediction model	R_P
PNFIR	18.1017
JPPSOV	16.6485
PRNN	16.5766

The SNR of observation noise is chosen to be 50 dB. The parameters of PNFIR are chosen as $\eta_H = 0.8$, $\eta_w = 0.7$ and $\rho = 0.1$. The learning rate of nonlinear and linear subsection of PRNN and JPPSOV are set to $\eta 1_H = 0.9$; $\eta 1_w = 0.9$ and $\eta 2_H = 0.6$; $\eta 2_w = 0.1$, respectively.

Fig. 12 shows four curves of 4000 samples of the speech signals including original speech, predicting by the PNFIR, predicting by the PRNN and predicting by the JPPSOV. Fig. 13 depicts corresponding predicting errors. The results obtained from Figs. 12 and 13 show that the performance of PNFIR is better than that of PRNN and JPPSOV.

In addition, according to the one-step forward prediction gain R_p of all the predictors (Table 5), it can be seen that the gain R_p of PNFIR is better than that of the JPPSOV and PRNN. Thus, the PNFIR also has a good tracking capability for speech signals.

9. Conclusions

A novel nonlinear adaptive filter named pipelined neural FIR (PNFIR) is presented in the paper. Since the PNFIR filter includes a number of neural FIR modules, its architecture is simple and easy to implement. Moreover, based on nested nonlinearity of cascaded modules and the advantages of convex combination function, the nonlinear processing capability of PNFIR can be improved. The simulation results demonstrate that the proposed PNFIR filter exhibits faster convergence rate, lower computational complexity and better ability to efficiently approximate nonlinear systems than PRNN and JPPSOV filters. However, it should be noted that the PNFIR filter is more efficiently used under the amplitude of external input signal is small. In conclusion, those characteristics are believed to make the PNFIR filter be a potential choice for the problems of nonlinear signal processing.

CRediT authorship contribution statement

Dinh Cong Le: Data curation, Writing - original draft, Software, Writing - review & editing. **Jiashu Zhang:** Conceptualization, Methodology, Supervision, Validation. **Yanjie Pang:** Visualization, Data curation, Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by National Science foundation of P. R. China (Grant: 61671392, 62071396).

References

[1] V.J. Mathews, G.L. Sicuranza, Polynomial Signal Processing, Wiley, New York, 2001.
[2] S. Haykin, Neural Networks – A Comprehensive Foundation, 2th ed., Prentice-Hall, Englewood Cliffs, NJ, 1994.

[3] J.J. Rubio, I. Elias, D.J. Cruz, J. Pacheco, Uniform stable radial basis function neural network for the prediction in two mechatronic processes, Neurocomputing 227 (2017) 122–130.

[4] C. Zhang, H. Wei, L. Xie, Y. Shen, K. Zhang, Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework, Neurocomputing 205 (2016) 53–63.

[5] R.M. Ehsan, S.P. Simon, P.R. Venkateswaran, Day-ahead forecasting of solar photovoltaic output power using multilayer perceptron, Neural Comput. Appl. 28 (2017) 3981–3992.

[6] D. Mandic, J. Chambers, Recurrent Neural Networks for Prediction, Wiley, New York, 2001.

[7] H.M.R. Ugalde, J.C. Carmona, J.R. Reyes, V.M. Alvarado, J. Mantilla, Computational cost improvement of neural network models in black box nonlinear system identification, Neurocomputing 166 (2015) 96–108.

[8] R. Kumar, S. Srivastava, J.R.P. Gupta, A. Mohindru, Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates, Neurocomputing 287 (2018) 102–117.

[9] S. Panda, P.K. Mohapatra, S.P. Panigrahi, A new training scheme for neural networks and application in non-linear channel equalization, Appl. Soft Comput. 27 (2015) 47–52.

[10] S. Chen, B. Mulgrew, P.M. Grant, Clustering technique for digital communication channel equalization using radial basis function networks, IEEE Trans Neural Netw. 4 (4) (1993) 570–579.

[11] S. Chen, B. Mulgrew, Overcoming co-channel interference using an adaptive radial basis function equalizer, Signal Process. 28 (1992) 91–107.

[12] G. Kechriotis, E. Zervas, E.S. Manolakos, Using recurrent neural network for adaptive communication channel equalization, IEEE Trans Neural Nets. 5 (2) (1994) 267–278.

[13] J. Baltersee, J.A. Chambers, Nonlinear adaptive prediction of speech with a pipelined recurrent neural network, IEEE Trans. Signal Process. 46 (8) (1998) 2207–2216.

[14] D.G. Stavrakoudis, J.B. Theocharis, Pipelined recurrent fuzzy neural networks for nonlinear adaptive speech prediction, IEEE Trans. Syst. Man Cybern. B 37 (5) (2007) 1305–1320.

[15] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.

[16] D.C. Le, J. Zhang, Y. Pang, A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition, Appl. Acoust. 132 (2018) 19–25.

[17] M. Bouchard, B. Paillard, C.T.L. Dinh, Improved training of neural networks for the nonlinear active noise control of sound and vibration, IEEE Trans. Neural Netw. 10 (1999) 391–401.

[18] S. Haykin, L. Li, Nonlinear adaptive prediction of nonstationary signals, IEEE Trans. Signal Process. 43 (2) (1995) 526–535.

[19] L. Li and S. Haykin, A cascaded recurrent neural networks for real-time nonlinear adaptive filtering, in Proc. IEEE Int. Conf. Neural Netw., San Francisco, CA, (1993) 857–862.

[20] H. Zhao, J. Zhang, Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network, Neurocomputing 72 (2009) 3046–3054.

[21] D.G. Stavrakoudis, J.B. Theocharis, Pipelined recurrent fuzzy neural networks for nonlinear adaptive speech prediction, IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 37 (5) (2007) 1305–1320.

[22] J.S. Zhang, Y.J. Pang, Pipelined robust M-estimate adaptive second order Volterra filter against impulsive noise, Digit. Signal Process. 26 (2014) 71–80.

[23] P.R. Chang, J.T. Hu, Optimal nonlinear adaptive prediction and modeling of MPEG video in ATM networks using pipelined recurrent neural networks, IEEE J. Sel. Areas Commun. 15 (6) (1997) 1087–1100.

[24] T.G. Burton, R.A. Goubran, F. Beaucoup, Nonlinear system identification using a subband adaptive Volterra filter, IEEE Trans. Instrum. Meas. 58 (5) (2009) 1389–1397.

[25] D.G. Lainiotis, P. Papaparaskeva, A partitioned adaptive approach to nonlinear channel equalization, IEEE Trans. Commun. 46 (10) (1998) 1325–1336.

[26] M.H. Asyali, M. Juusola, Use of Meixner functions in estimation of Volterra kernels of nonlinear systems with delay, IEEE Trans. Biomed. Eng. 52 (2) (2005) 229–237.

[27] L. Tan, J. Jiang, Adaptive Volterra filters for active control of non-linear noise processes, IEEE Trans. Signal Process. 49 (8) (2001) 1667–1676.

[28] F. Kuech, W. Kellermann, Nonlinear line echo cancellation using a simplified second order Volterra filter, in: Proc. IEEE ICASSP, (2002) 1117–1120.

[29] T. Koh, E.J. Powers, Second-order Volterra filtering and its application to nonlinear system identification, IEEE Trans. Acoust. Speech Signal Process. 33 (6) (1985) 1445–1455.

[30] Y. Lou, C.L. Nikias, A.N. Venetsanopoulos, Efficient VLSI array processing structures for adaptive quadratic digital filters, Circuits Syst. Signal Process 7 (2) (1988) 253–273.

[31] T.M. Panicker, V.J. Mathews, G.L. Sicuranza, Adaptive parallel-cascade truncated Volterra filters, IEEE Trans. Signal Process. 46 (1998) 2664–2673.

[32] M.M. Banat, Pipelined volterra filter, Electron. Lett. 28 (13) (1992) 1276–1277.

[33] H.Q. Zhao, J.S. Zhang, A novel adaptive nonlinear filter-based pipelined feedforward second-order Volterra architecture, IEEE Trans. Signal Process. 57 (1) (2009) 237–246.

[34] D.P. Mandic, J.A. Chambers, Relations between the a priori and a posteriori errors in nonlinear adaptive neural filters, IEEE Trans. Neural Computation. 12 (2000) 1285–1292.

[35] D.P. Mandic, NNGD algorithm for neural adaptive filters, Electron Lett. 39 (9) (2000) 845–846.

[36] D.P. Mandic, A.I. Hanna, M. Razaz, A normalized gradient descent algorithm for nonlinear adaptive filters using gradient adaptive step size, IEEE Signal Process Lett. 8 (11) (2001) 295–297.

[37] S.L. Goh, D.P. Mandic, Nonlinear neural FIR filter with an adaptive activation function, J. Automat. Control Univers. Belgrade 13 (1) (2003) 1–5.

[38] A.I. Hanna, D.P. Mandic, Nonlinear FIR adaptive filters with a gradient adaptive amplitude in the nonlinearity, IEEE Signal Process Lett. 9 (8) (2002) 253–255.

[39] D.C. Le, J.S. Zhang, D. Li, Hierarchical partial update generalized functional link artificial neural network filter for nonlinear active noise control, Digit. Signal Process. 93 (2019) 160–171.

[40] J.S. Zhang, H.Q. Zhao, A novel adaptive bilinear filter based on pipelined architecture, Digit. Signal Process. 20 (1) (2010) 23–38.

[41] S. Zhang, J.S. Zhang, Y.J. Pang, Pipelined set-membership approach to adaptive Volterra filtering, Signal Process. 129 (2016) 195–203.

[42] J. Arenas-García, V. Gomez-Verdejo, A.R. Figueiras-Vidal, New algorithms for improved adaptive convex combination of LMS transversal filters, IEEE Trans. Instrum. Meas. 54 (6) (2005) 2239–2249.

[43] S. Haykin, Adaptive Filter Theory, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.

[44] D.P. Mandic, J.A. Chambers, Towards an optimal learning rate for backpropagation, Neural Process. Lett. 11 (1) (2000) 1–5.

[45] M.Z. Shi, M. Han, Support vector echo-state machine for chaotic time-series prediction, IEEE Trans. Neural Netw. 18 (2) (2007) 359–372.

Dinh Cong Le received the B.S. degree from Hanoi National University, Hanoi, Vietnam, in 2001, the M.S. degree from Le Quy Don Technology University, Hanoi, Vietnam, in 2011, and the Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2019. He is currently a lecturer in School of Engineering and Technology, Vinh University, Vietnam. His current research interests include adaptive filtering and nonlinear signal processing for communication.

Jiashu Zhang received the B.S. degree from University of Electronic Science and Technology of China, Chengdu, China, in 1987, the M.S. degree from Chongqing University, Chongqing, China, in 1990, and the Ph.D. degree from University of Electronic Science and Technology of China, Chengdu, China, in 2001.In 2001, he joined the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Currently, he is a professor and the director of Sichuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, China. He has published more than 200 journal and conference papers. His research interests include nonlinear and adaptive signal processing, biometric security and privacy, and information forensic and data hiding.

Yanjie Pang received the B.S .degrees in computer science in School of Information Science & Technology from Southwest Jiaotong University, Chengdu, China in 2005. She is currently working towards the Ph.D. degree in the field of nonlinear signal processing at the Si-Chuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, China. Her current research interests include adaptive filtering, nonlinear signal processing for communication, nonlinear systems and active noise control.