Lyapunov-Theory-Based Radial Basis Function Networks for Adaptive Filtering

Kah Phooi Seng, Zhihong Man, and Hong Ren Wu

***Abstract*—Two important convergence properties of *Lyapunov-theory-based adaptive filtering* (LAF) adaptive filters are first explored. The LAF *finite impulse response* and *infinite impulse response* adaptive filters are then realized using the *radial basis function* (RBF) *neural networks* (NNs). The proposed adaptive RBF neural filtering system possesses the distinctive properties of RBF NN and the LAF filtering system. Unlike many adaptive filtering schemes using gradient search techniques, a Lyapunov function of the error between the desired signal and the RBF NN output is first defined. By properly choosing the weights update law in the Lyapunov sense, the RBF filter output can asymptotically converge to the desired signal. The design is independent of the stochastic properties of the input disturbances and the stability is guaranteed by the Lyapunov stability theory. Simulation examples for nonlinear adaptive prediction of nonstationary signal and system identification are performed.**

***Index Terms*—Adaptive filtering, Lyapunov stability theory, radial basis function neural network.**

I. INTRODUCTION

Adaptive filtering has achieved widespread applications [1]. However, linear filtering is the most widely used and poor performance is usually observed for real world applications. Therefore the development of nonlinear filters is necessary. Among various adaptive nonlinear filters, neural networks are becoming increasingly popular. One type of nonlinear filters is multiplayer perceptron (MLP), but they are highly nonlinear in parameters. The parameter estimate may be trapped at a local minimum of the chosen optimization criterion when a gradient descent algorithm such as backpropagation (BP) is used. Other optimization techniques [2]–[4] are capable of achieving a global minimum but they require intensive computation. An alternative to highly nonlinear MLP is the *radial basis function* (RBF) *neural networks* (NNs). Their universal approximation property [5] and straightforward computation using a linearly weighted combination of single hidden-layer neurons have made RBF NN natural choices in many applications.

Due to the linear-weighted combiner, the weights can be determined using *least mean square* (LMS) and *recursive least square* (RLS) algorithms. However, these algorithms suffer from several drawbacks and limitations. The LMS is highly dependent on the autocorrelation function associated with the input signals and slow convergence. RLS, on the other hand, provides faster convergence but they rely on the implicit or explicit computation of the inverse of the input signal's autocorrelation matrix. This not only implies a higher computational cost, but it can lead to instability problems [6]. Methods of avoiding instability have been developed in [6]–[10], but the stability problem of adaptive filters with bounded input disturbances has not been solved. In addition, most of adaptive filtering schemes suffer so-called local minima problem, i.e., the optimization search may stop at a local minimum of the cost function in the parameter space if the initial values are arbitrarily chosen. For example, the cost function of a gradient search-based adaptive filtering system has a fixed structure in the parameter space after the expression of the cost function is chosen. The parameter update law is only a mean to search for the global minimum and independent of the cost function in the parameter space. Furthermore, if the disturbances encountered by the filter are random signals, the mathematics of stochastic processes need to be used for the optimization and parameter design.

Manuscript received May 22, 2000; revised January 5, 2001 and October 24, 2001. This paper was recommended by Associate Editor W.-S. Lu.

K. P. Seng is with the School of Engineering, Monash University, Selangor, Malaysia (e-mail: kp_seng@yahoo.com).

Z. Man is with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: ASZHMan@ntu.edu.sg).

H. R. Wu is with the School of Computer Science and Software Engineering, Monash University, Clayton VIC 3168, Australia (e-mail: hrw@csse.monash.edu.au).

Publisher Item Identifier 10.1109/TCSI.2002.801255.

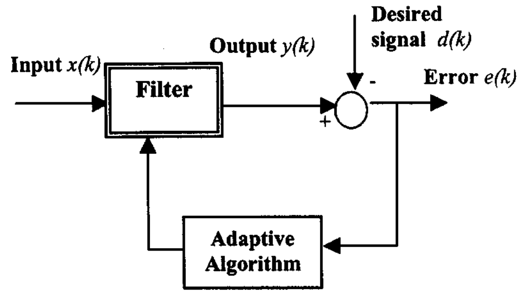

Fig. 1. Adaptive filtering problem.

To overcome the aforementioned problems, authors [11] proposed *Lyapunov-theory-based adaptive filtering* (LAF). In [11], a Lyapunov function of the error between the desired signal and the filter output is first defined, the filter weights are then adaptively adjusted based on *Lyapunov stability theory* (LST) so that the error can asymptotically converges to zero. The selected Lyapunov function for an LAF filter has a unique global minimum in the state–space. By properly choosing the parameter update law in Lyapunov sense, the filter output can asymptotically converge to the desired signal. Although the input signal of the adaptive filter is disturbed by the bounded random noises, only the input and output measurements are used for the filter design. Therefore, the design of the LAF filter is independent of the stochastic properties of the random input disturbances. Moreover, stability is guaranteed because the error dynamics asymptotically converge to zero. It can be seen that LST provides an optimization method in the state–space for the filter design. In this brief, two realizations of the LAF *finite-impulse response* (FIR) and *infinite-impulse response* (IIR) adaptive filters using the feedforward and recurrent RBF NNs are proposed. Before that, the error convergence rate of the LAF and the convergence region of the error for the modified LAF to avoid the singularities are explored. The proposed adaptive neural filtering system inherits the good properties of RBF NN and LAF filtering system.

II. PROBLEM FORMULATION

The typical structure of an adaptive filtering system is illustrated in Fig. 1, where $x(k)$ is the input signal, which is corrupted due to the nonlinearity of the communication channel and noises. The $y(k)$ is the filter output and $d(k)$ is the *desired response*. The difference between the desired response and the filter output is the error $e(k)$ in (2.1). The adaptive algorithm is generally designed to adaptively update the filter parameters so that a cost function of the error is minimized in the parameter space

$$e(k) = d(k) - y(k). \tag{2.1}$$

The basic principle of the LAF algorithm can be briefly introduced as follows. Consider the following FIR system:

$$y(k) = H^T(k)X(k) \tag{2.2}$$

where

$$H(k) = [h_k(0), h_k(1), \dots, h_k(N-1)]^T \tag{2.3}$$
$$X(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T. \tag{2.4}$$

The design of the LAF for FIR filter can be summarized in the following theorem.

Theorem 2.1 [11]: For the given desired response $d(k)$, if the weight vector $H(k)$ of the filter $y(k) = H^T(k)X(k)$ is updated as follows:

$$H(k) = H(k-1) + g(k)\alpha(k) \tag{2.5}$$

and

$$g(k) = \frac{X(k)}{\|X(k)\|^2}\left(1 - \kappa\frac{|e(k-1)|}{|\alpha(k)|}\right) \tag{2.6}$$

where $g(k)$ is the adaptation gain, $\alpha(k)$ is the *a priori* estimation error defined as

$$\alpha(k) = d(k) - H^T(k-1)X(k) \tag{2.7}$$

and

$$0 \le \kappa < 1 \tag{2.8}$$

then, the error $e(k)$ asymptotically converges to zero.

Proof [11]: To prevent the singularities of the gain $g(k)$ in (2.6) when $\|X(k)\|$ and $\alpha(k)$ approach zero, (2.6) is modified as (2.9)

$$g(k) = \frac{X(k)}{\lambda_1 + \|X(k)\|^2}\left(1 - \kappa\frac{|e(k-1)|}{\lambda_2 + |\alpha(k)|}\right). \tag{2.9}$$

The results in [11] provided only a basic idea of the Lyapunov filtering. Many problems, such as the analysis of convergence rate in Theorem 2.1 and the convergence region of the adaptive filter using the modified gain in (2.9) have not been investigated. In the following theorems, some important properties of the Lyapunov filtering systems are explored.

Theorem 2.2: Consider the FIR filtering system in (2.2). If the Lyapunov update law in (2.5)–(2.7) is used to update the filter parameters, the error $e(k)$ between the desired signal $d(k)$ and the filter output $y(k)$ converges to zero exponentially.

Proof: Appendix A: Expression (3.2) shows that the error $e(k)$ converges to zero exponentially and the convergence rate is controlled by the positive constant κ. The smaller κ is, the faster the error converges.

The convergence region of the adaptive filter using the modified gain in (2.9) is given by the following theorem.

Theorem 2.3: Consider the FIR system in (2.2). If the filter parameters are updated according to the modified adaptive laws (2.10)–(2.12),

$$H(k) = H(k-1) + g(k)\alpha(k) \tag{2.10}$$
$$g(k) = \frac{X(k)}{\lambda_1 + \|X(k)\|^2}\left(1 - \kappa\frac{|e(k-1)|}{\lambda_2 + |\alpha(k)|}\right) \tag{2.11}$$
$$\alpha(k) = d(k) - H^T(k-1)X(k) \tag{2.12}$$

then, the error $e(k)$ will converge to the ball centered at the origin of the error space with the radius

$$r_{e1} = \frac{-\frac{3\kappa\lambda_2\overline{\lambda}}{2} + \sqrt{\left(\frac{3\kappa\lambda_2\overline{\lambda}}{2}\right)^2 + 4\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)\frac{9}{4}\lambda_2^2}}{-2\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)} \tag{2.13}$$

where $\overline{\lambda}$ is a constant discussed in Appendix B.

Proof: Appendix B.

Remark 2.1: Similarly, for an IIR filtering system, the design principle of the Lyapunov filtering can also be implemented if $y(k)$, $H(k)$ and $X(k)$ are specified by (2.14)–(2.16).

$$y(k) = B^T(k)\tilde{X}(k) + A^T(k)\tilde{Y}(k-1) = H^T(k)X(k) \tag{2.14}$$

where

$$\begin{aligned} A(k) &= [a_1(k), a_2(k), \dots, a_{N-1}(k)]^T \\ B(k) &= [b_0(k), b_1(k), \dots, b_{N-1}(k)]^T \\ \tilde{X}(k) &= [x(k), x(k-1), \dots, x(k-N+1)]^T \\ \tilde{Y}(k) &= [y(k-1), y(k-2), \dots, y(k-N+1)]^T \\ H(k) &= [b_0(k), b_1(k), \dots, b_{N-1}(k), \\ &\qquad a_1(k), a_2(k), \dots, a_{N-1}(k)]^T \end{aligned} \tag{2.15}$$
$$\begin{aligned} X(k) &= [x(k), x(k-1), \dots, x(k-N+1), \\ &\qquad y(k-1), y(k-2), \dots, y(k-N+1)]^T. \end{aligned} \tag{2.16}$$

It is easy to prove that, for IIR filtering system in (2.14), the error can also exponentially converge to zero if the adaptive law in (2.5)–(2.7) is used with specified $y(k)$, $H(k)$, and $X(k)$ in (2.14)–(2.16), respectively.

Remark 2.2: It is known that the gradient search-based optimization is indeed affected by the stochastic properties of the signals. From the introduction and the analysis of the error convergence properties of LAF, if the input disturbances are bounded random processes, the adaptive filtering algorithm can be directly designed using the input and output measurements based on the LST without considering the stochastic properties of the signals. This point is similar to the design of *Lyapunov-stability-based adaptive control systems* and *variable-structure control system* [12].

Remark 2.3: The LST provides an optimization method in state–space. It is different from the gradient search based methods. According to LST, the selected $V(k)$ is a Lyapunov function if and only is $\Delta V(k)$ is negative ($\Delta V(k) < 0$). For the Lyapunov adaptive filtering system, whether or not $\Delta V(k)$ is negative depends on the selection of the parameter update law. Only when the parameter update law is chosen in Lyapunov sense, $V(k) = e^2(k)$ is a Lyapunov function of the designed adaptive filtering system, which has a unique global minimum in this case. Therefore, the selection of the Lyapunov function and the parameter update law are not independent. The cost function and the Lyapunov function have many different characteristics, but they are all energy-like functions. One is considered in the state space, and another is considered in the parameter space. The corresponding optimization methods can be used for the design of adaptive filters with different requirements.Now

Remark 2.4: The LAF algorithm has the computational complexity, which is proportional to N multiplies per weight vector update ($\propto N$, N is filter order). It has less computational complexity compared with that of RLS that is proportional to N^2 due to the matrix multiplication. In contrast, LMS has the simplest computational complexity that is proportional to N.

III. Realization of Adaptive LAF Filters Using RBF Neural Networks

We consider the realization of adaptive FIR and IIR filters using RBF NNs. The advantages of this realization are that the RBF NNs that have some linear properties and the network parameters can be easily adjusted using LAF algorithm. In addition, the parallel structure of RBF NNs is suitable for the fast signal processing.

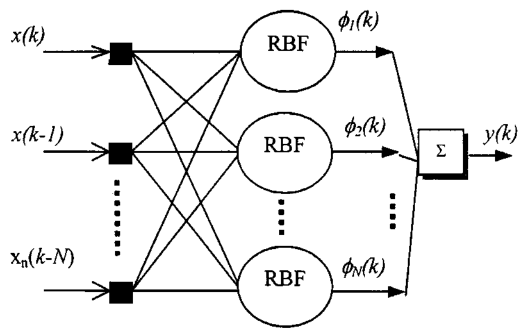

Fig. 2. Nonlinear feedforward RBF FIR filter.

A. Feedforward RBF Lyapunov Adaptive FIR Filter

Fig. 2 shows the realization of the adaptive FIR filter using the feedforward RBF. The RBF FIR filter output can be expressed as

$$y(k) = \sum_{i=1}^{N} w_i(k)\phi_i(k) \tag{3.1}$$

or

$$y(k) = W^T(k)\Phi(k) \tag{3.2}$$

where

$$W(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^T \tag{3.3}$$
$$\Phi(k) = [\phi_1(k), \phi_2(k), \ldots, \phi_N(k)]^T. \tag{3.4}$$

$\phi(k)$ is the Gaussian type of function defined as

$$\phi_i(k) = \exp\left(-\frac{\|X(k)-c_i\|}{\sigma_i^2}\right), \qquad i = 1, 2, 3, \ldots N. \tag{3.5}$$

$X(k) = [x(k), x(k-1), \ldots, x(k-N)]^T$, c_i is the center vector and σ_i is the width of Gaussian function. The width is controlled by the noise variance σ_n^2 and is usually set at $\sigma_i = 2\sigma_n^2$.

Using the results of Theorem 2.1 in Section II, we have the following update law for the RBF filter:

$$W(k) = W(k-1) + g(k)\alpha(k) \tag{3.6}$$
$$\alpha(k) = d(k) - W^T(k-1)\Phi(k) \tag{3.7}$$
$$g(k) = \frac{\Phi(k)}{\|\Phi(k)\|^2}\left(1-\kappa\frac{|e(k-1)|}{|\alpha(k)|}\right) \tag{3.8}$$

or

$$g(k) = \frac{\Phi(k)}{\|\Phi(k)\|^2+\lambda_1}\left(1-\kappa\frac{|e(k-1)|}{\lambda_2+|\alpha(k)|}\right). \tag{3.9}$$

Remark 3.1: It is easy to see that the stability analysis of the error dynamics, convergence analysis of the Lyapunov RBF FIR adaptive filter are same as the ones given in [11, Th. 2.1, 2.2] if we replace $X(k)$ by $\Phi(k)$.

Remark 3.2: The centers are either randomly selected from the data or determined using the k-means clustering [13], [14] algorithm if the number of hidden neurons needs to be relatively large to cover the entire input domain.

B. Recurrent RBF Lyapunov Adaptive IIR Filter

The basic structure of the adaptive IIR filter using the recurrent RBF NN is given in Fig. 3. The output is written as

$$y(k) = \sum_{i=1}^{N} w_i(k)\phi_i(k) + \sum_{j=1}^{M} w_{N+j}(k)\phi_{N+j}(k) \tag{3.10}$$

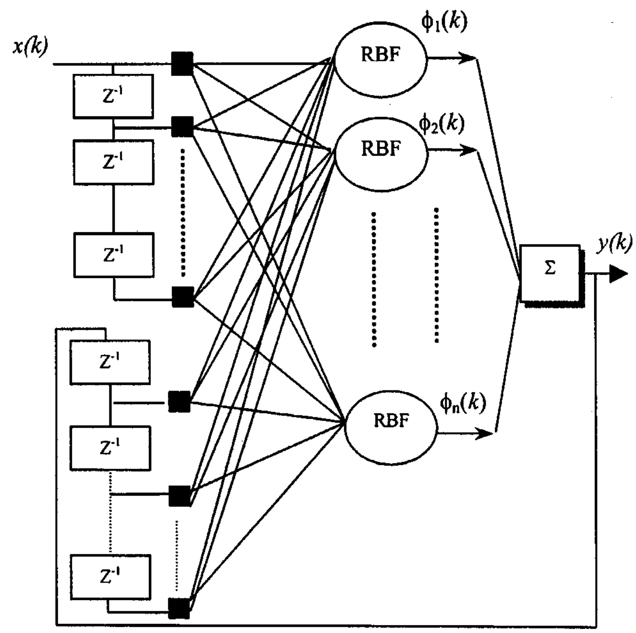

Fig. 3. Nonlinear recurrent RBF IIR filter.

or

$$y(k) = \theta^T(k)\Phi(k) \tag{3.11}$$

where

$$\theta(k) = [w_1(k), w_2(k), \ldots, w_N(k), w_{N+1}(k), \ldots, w_{N+j}(k)]^T \tag{3.12}$$
$$\Phi(k) = [\phi_1(k), \phi_2(k), \ldots, \phi_N(k), \phi_{N+1}(k), \ldots, \phi_{N+j}(k)]^T \tag{3.13}$$
$$X(k) = [x(k), x(k-1), \ldots, x(k-N), y(k-1), \ldots, y(k-N)]^T \tag{3.14}$$

and $\phi(k)$ is defined in (3.5).

Using the results in Sections II and III-A, the network parameters can be updated as follows:

$$\theta(k) = \theta(k-1) + g(k)\alpha(k) \tag{3.15}$$
$$\alpha(k) = d(k) - \theta^T(k-1)\Phi(k) \tag{3.16}$$
$$g(k) = \frac{\Phi(k)}{\|\Phi(k)\|^2}\left(1-\kappa\frac{|e(k-1)|}{|\alpha(k)|}\right) \tag{3.17}$$

and the modified $g(k)$ is given by

$$g(k) = \frac{\Phi(k)}{\|\Phi(k)\|^2+\lambda_1}\left(1-\kappa\frac{|e(k-1)|}{\lambda_2+|\alpha(k)|}\right). \tag{3.18}$$

IV. Simulation Examples

Nonstationary Time Series Prediction

Simulations have been done for a one-step ahead prediction of a nonlinear and nonstationary speech signal which is identical to that used by S. Haykin [15]–[17]. The signal is downloaded from the Internet [18] and is described as follows: S1 speech sample "When recording audio data … ," length 10 000, sampled at 8 kHz. The NN is expected to be able to track the nonstationary signal characteristic. An input order of $p = 50$, which is used in [16], and $c = 50$ centers are used for this

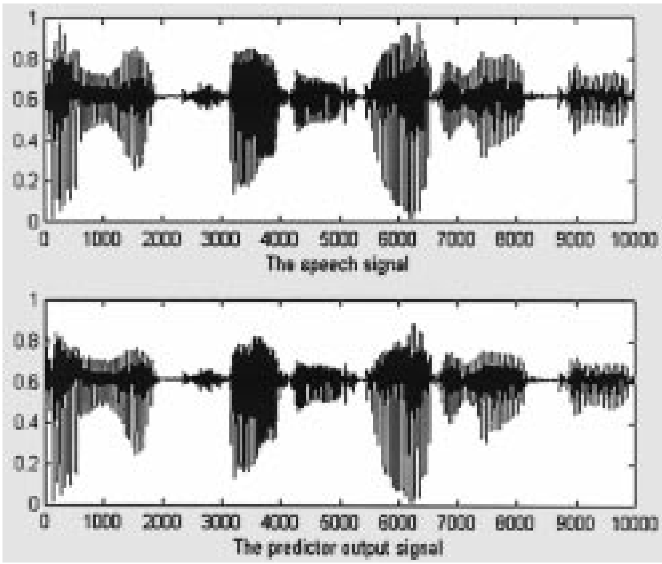

Fig. 4. RBF FIR predictor—the speech signal and predictor output.

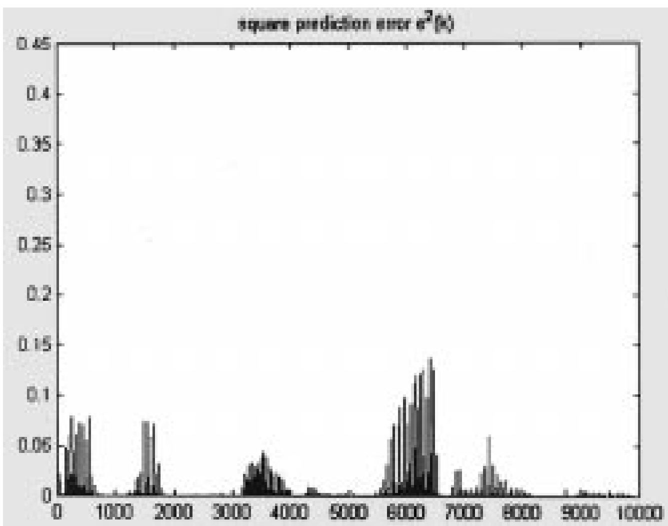

Fig. 5. RBF FIR predictor—the square prediction error, $e^2(k)$.

simulation. Fig. 4 shows the speech signal and the RBF FIR predictor output. Fig. 5 illustrates the square predictor error. For comparison to previous works [15]–[17], the performance measure we shall use is the *predicted signal-to-noise ratio* (PSNR) defined by

$$\mathrm{PSNR(dB)} \triangleq 10 \log_{10} \left(\tilde{\sigma}_s^2 / \tilde{\sigma}_e^2 \right) \tag{4.1}$$

where $\tilde{\sigma}_s^2$ and $\tilde{\sigma}_e^2$ are the actual and error signal powers estimated by $\tilde{\sigma}_s^2 \triangleq (1/N) \sum_{i=1}^{N} y^2(i)$ and $\tilde{\sigma}_e^2 \triangleq (1/N) \sum_{i=1}^{N} e^2(i)$.

For 10 000 speech samples, $\tilde{\sigma}_s^2$ is calculated to be 0.3854. The $\tilde{\sigma}_e^2$ is about 0.0019, yielding PSNR $= 23.11$ dB. The same speech signal has been used as part of three previous studies, the dynamic regularized RBF [16] based on the *regularized least-squares fitting* (RLSF), pipelined recurrent NNs (PRNN) [15], [17] which are another method of modeling nonstationary dynamics. The authors in [17] have done the simulations for PRNN and standard linear adaptive filters. While considerably different in details of their architectures and training methods, they do share the common principle of continuously adapting their network parameters to yield minimum squared prediction error and track nonstationary signal characteristics. Comparing their results, our PSNR is 8.7 dB better than the best PSNR of 14.71 dB listed in [16, Table IV] and 8.82 dB better than that of 13.59 dB listed in [17, Table II] for a hybrid *extended RLS* (ERLS)-trained PRNN followed by a 12th-order RLS filter. However, the computational complexity of our method and RBF NN with $p = 50$ inputs and $c = 50$ centers is less than that of [16] with $p = 50$, $c = 100$.

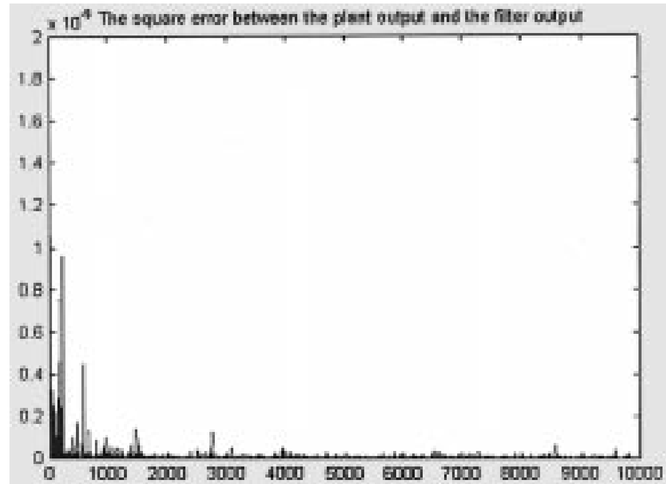

Fig. 6. RBF IIR (system identification)—squared error, $e^2(k)$ (y axis: $x10^{-9}$).

Nonlinear System Identification

A nonlinear single-input single-output (SISO) system is considered in this simulation example

$$y(k) = 0.0705x(k) - 0.141x(k-1) - 0.0705x(k-1) + 1.1993e^{-y^2(k-1)}y(k-1) - 0.5156e^{-y^2(k-2)}y(k-2).$$

The input signal that is random noise with zero mean value and variance 1 is used to excite the nonlinear plant. The recurrent RBF NN has five input and five hidden nodes and one output node with the feedback connections back to two nodes in the input layer. Fig. 6 has revealed the squared error, $e^2(k)$. The mean square error (MSE) of the proposed RBF IIR filter is 3.5519×10^{-12}. This extremely small MSE has indicated the proposed scheme has high tracking ability and can adaptively identify the nonlinear plant.

V. Conclusion

In this brief, a new scheme called Lyapunov-theory-based RBF NNs has been proposed for adaptive filtering. The proposed scheme is based on the adaptive filter theory that the feedforward and recurrent RBF NNs are considered as the FIR and IIR filters, respectively. The weight parameters are adaptively adjusted using the LAF algorithm so that the dynamic error can converge to zero asymptotically. Two important convergence properties of the LAF are discussed. The design is independent of the stochastic properties of the input disturbances and the stability is guaranteed by the LST. The computational complexity of the LAF is less than that of RLS. The simulation results have demonstrated good convergence and tracking properties of the proposed RBF filters. Further research, based on this brief, is to use different Lyapunov functions and different adaptive laws to further improve the convergence and robustness properties of the Lyapunov filter or RBF NN filters with respect to the bounded random disturbances.

APPENDIX A

Proof of Theorem 2.1: Using (2.1) and (2.5)–(2.7), the error $e(k)$ can be expressed as

$$\begin{aligned} e(k) &= d(k) - y(k) = d(k) - H^T(k)X(k) \\ &= d(k) - [H^T(k-1) + g^T(k)\alpha(k)]X(k) \\ &= d(k) - H^T(k-1)X(k) - g^T(k)\alpha(k)X(k) \\ &= \alpha(k) - g^T(k)\alpha(k)X(k) \\ &= \alpha(k) - \frac{X^T(k)}{\|X(k)\|^2}\left(1 - \kappa\frac{|e(k-1)|}{|\alpha(k)|}\right)\alpha(k)X(k) \\ &= \alpha(k) - \left(1 - \kappa\frac{|e(k-1)|}{|\alpha(k)|}\right)\alpha(k) = \kappa\frac{\alpha(k)}{|\alpha(k)|}|e(k-1)| \\ &= \kappa|e(k-1)|\mathrm{sgn}(\alpha(k)) \end{aligned} \tag{A1}$$

$$\begin{aligned} \therefore |e(k)| &= \kappa|e(k-1)| \\ |e(1)| &= \kappa|e(0)| \\ |e(2)| &= \kappa|e(1)| = \kappa^2|e(0)| \\ &\vdots \\ |e(k)| &= \kappa^k|e(0)|. \end{aligned} \tag{A2}$$

APPENDIX B

Proof of Theorem 2.2: Define a Lyapunov function

$$V(k) = e^2(k). \tag{B1}$$

We then have $\Delta V(k) = V(k) - V(k-1)$

$$\begin{aligned} \Delta V(k) &= \alpha^2(k)(1 - g^T(k)X(k))^2 - e^2(k-1) \\ &= \alpha^2(k)\left[1 - \frac{\|X(k)\|^2}{\lambda_1 + \|X(k)\|^2}\left(1 - \kappa\frac{|e(k-1)|}{\lambda_2 + |\alpha(k)|}\right)\right]^2 \\ &\quad - e^2(k-1) \\ &= \alpha^2(k)\left[1 - \frac{\frac{\|X(k)\|^2}{\lambda_1}}{1 + \frac{\|X(k)\|^2}{\lambda_1}}\left(1 - \frac{\kappa}{\lambda_2}\frac{|e(k-1)|}{1 + \frac{|\alpha(k)|}{\lambda_2}}\right)\right]^2 \\ &\quad - e^2(k-1) \end{aligned} \tag{B2}$$

where

$$\frac{\|X(k)\|^2}{\lambda_1} < 1 \quad \text{and} \quad \frac{|\alpha(k)|}{\lambda_2} < 1. \tag{B3}$$

The following equations are obtained using Taylor series:

$$\frac{\frac{\|X(k)\|^2}{\lambda_1}}{1 + \frac{\|X(k)\|^2}{\lambda_1}} = \frac{\|X(k)\|^2}{\lambda_1} + O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) \tag{B4}$$

and

$$\frac{1}{1 + \frac{|\alpha(k)|}{\lambda_2}} = 1 - \frac{|\alpha(k)|}{\lambda_2} + O\left(\frac{|\alpha(k)|}{\lambda_2}\right) \tag{B5}$$

where

$$\begin{aligned} \left|O\left(\frac{\|X(k)\|^2}{\lambda_1}\right)\right| &= \left|\frac{\|X(k)\|^2}{\lambda_1 + \|X(k)\|^2} - \frac{\|X(k)\|^2}{\lambda_1}\right| \\ &= \frac{\|X(k)\|^4}{\lambda_1(\lambda_1 + \|X(k)\|^2)} \\ &= \frac{\|X(k)\|^2}{\lambda_1}\cdot\frac{\|X(k)\|^2}{\lambda_1 + \|X(k)\|^2} \le 1 \times \frac{1}{2} = \frac{1}{2} \end{aligned} \tag{B6}$$

and

$$\begin{aligned} \left|O\left(\frac{|\alpha(k)|}{\lambda_2}\right)\right| &= \left|\frac{\lambda_2}{\lambda_2 + |\alpha(k)|} - 1 + \frac{|\alpha(k)|}{\lambda_2}\right| \\ &= \frac{|\alpha(k)|^2}{\lambda_2(\lambda_2 + |\alpha(k)|)} \\ &= \frac{|\alpha(k)|}{\lambda_2}\cdot\frac{|\alpha(k)|}{(\lambda_2 + |\alpha(k)|)} \le 1 \times \frac{1}{2} = \frac{1}{2}. \end{aligned} \tag{B7}$$

Then, the expression (B1) can be written as

$$\begin{aligned} \Delta V(k) &= \alpha^2(k)\Bigg[1 - \left(\frac{\|X(k)\|^2}{\lambda_1} + O\left(\frac{\|X(k)\|^2}{\lambda_1}\right)\right) \\ &\quad * \left(1 - \frac{\kappa|e(k-1)|}{\lambda_2}\left(1 - \frac{|\alpha(k)|}{\lambda_2} + O\left(\frac{|\alpha(k)|}{\lambda_2}\right)\right)\right)\Bigg]^2 \\ &\quad - e^2(k-1) \\ &= \alpha^2(k)\Bigg[1 - \left(\frac{\|X(k)\|^2}{\lambda_1} + O\left(\frac{\|X(k)\|^2}{\lambda_1}\right)\right) \\ &\quad * \left(1 - \frac{\kappa|e(k-1)|}{\lambda_2} + \frac{\kappa|e(k-1)||\alpha(k)|}{\lambda_2^2} - \frac{\kappa|e(k-1)|}{\lambda_2}O\left(\frac{|\alpha(k)|}{\lambda_2}\right)\right)\Bigg]^2 \\ &\quad - e^2(k-1) \\ &= \alpha^2(k)\Bigg[1 - \frac{\|X(k)\|^2}{\lambda_1} + \frac{\kappa\|X(k)\|^2|e(k-1)|}{\lambda_1\lambda_2} \\ &\quad - \frac{\kappa\|X(k)\|^2|e(k-1)||\alpha(k)|}{\lambda_1\lambda_2^2} \\ &\quad + \frac{\kappa\|X(k)\|^2|e(k-1)|}{\lambda_1\lambda_2}O\left(\frac{|\alpha(k)|}{\lambda_2}\right) \\ &\quad - O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) + \frac{\kappa|e(k-1)|}{\lambda_2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) - \frac{\kappa|e(k-1)||\alpha(k)|}{\lambda_2^2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) + \frac{\kappa|e(k-1)|}{\lambda_2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right)O\left(\frac{|\alpha(k)|}{\lambda_2}\right)\Bigg]^2 \\ &\quad - e^2(k-1) \\ &\le \alpha^2(k)\Bigg[1 + \frac{\kappa\|X(k)\|^2|e(k-1)|}{\lambda_1\lambda_2} \\ &\quad + \frac{\kappa\|X(k)\|^2|e(k-1)|}{\lambda_1\lambda_2}O\left(\frac{|\alpha(k)|}{\lambda_2}\right) \\ &\quad - O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) + \frac{\kappa|e(k-1)|}{\lambda_2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) - \frac{\kappa|e(k-1)||\alpha(k)|}{\lambda_2^2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right) + \frac{\kappa|e(k-1)|}{\lambda_2} \\ &\quad * O\left(\frac{\|X(k)\|^2}{\lambda_1}\right)O\left(\frac{|\alpha(k)|}{\lambda_2}\right)\Bigg]^2 \\ &\quad - e^2(k-1). \end{aligned} \tag{B8}$$

It is noted that

$$\begin{aligned}\Delta V(k) &< \lambda_2^2 \left[1 + \frac{\kappa|e(k-1)|}{\lambda_2} + \frac{\kappa|e(k-1)|}{2\lambda_2} + \frac{1}{2} + \frac{\kappa|e(k-1)|}{2\lambda_2} + \frac{\kappa\lambda_1|e(k-1)|}{2\lambda_2} + \frac{\kappa|e(k-1)|}{4\lambda_2}\right]^2 - e^2(k-1) \\ &= \lambda_2^2\left[\frac{3}{2} + \frac{\kappa|e(k-1)|}{2\lambda_2}\left(2+1+1+\lambda_1+\frac{1}{2}\right)\right]^2 - e^2(k-1) \\ &= \lambda_2^2\left[\frac{3}{2} + \frac{\kappa|e(k-1)|}{2\lambda_2}\left(4\frac{1}{2}+\lambda_1\right)\right]^2 - e^2(k-1) \end{aligned} \tag{B9}$$

and let

$$\overline{\lambda} = 4\,\tfrac{1}{2} + \lambda_1. \tag{B10}$$

$$\begin{aligned}\Delta V(k) &< \lambda_2^2\left[\frac{3}{2} + \frac{\kappa\overline{\lambda}}{2\lambda_2}\,|e(k-1)|\right]^2 - e^2(k-1) \\ &= \lambda_2^2\left[\frac{9}{4} + \frac{3\kappa\overline{\lambda}}{2\lambda_2}\,|e(k-1)| + \frac{\kappa^2\overline{\lambda}^2}{4\lambda_2^2}\,|e^2(k-1)|\right]^2 - e^2(k-1) \\ &= -\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)e^2(k-1) + \frac{3\kappa\lambda_2\overline{\lambda}}{2}\,|e(k-1)| + \tfrac{9}{4}\,\lambda_2^2. \end{aligned} \tag{B11}$$

For further analysis, we consider the following parabolic function:

$$\Delta\overline{V}(k) = -\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)e^2(k-1) + \frac{3\kappa\lambda_2\overline{\lambda}}{2}|e(k-1)| + \frac{9}{4}\,\lambda_2^2. \tag{B12}$$

If κ is small enough in the sense that

$$\frac{\kappa^2\overline{\lambda}^2}{4} < 1 \qquad \text{or} \qquad \frac{\kappa\overline{\lambda}}{2} < 1 \tag{B13}$$

and

$$\kappa\left(\tfrac{9}{2} + \lambda_1\right) < 2. \tag{B14}$$

$\Delta\overline{V}(k)$ in (B1) is a concave down parabolic function. Also, for the given λ_1 and λ_2, the small positive number κ satisfies the following inequality:

$$0 \le \kappa < 2\big/\left(\tfrac{9}{2} + \lambda_1\right). \tag{B15}$$

Solving the quadratic equation $\Delta\overline{V}(k) = 0$, we obtain the two roots as follows:

$$r_{e1,2} = \frac{-\frac{3\kappa\lambda_2\overline{\lambda}}{2} \pm \sqrt{\left(\frac{3\kappa\lambda_2\overline{\lambda}}{2}\right)^2 + 4\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)\frac{9}{4}\,\lambda_2^2}}{-2\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)}. \tag{B16}$$

The root r_{e1} is considered because $|e(k-1)| \ge 0$

$$r_{e1} = \frac{-\frac{3\kappa\lambda_2\overline{\lambda}}{2} + \sqrt{\left(\frac{3\kappa\lambda_2\overline{\lambda}}{2}\right)^2 + 4\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)\frac{9}{4}\,\lambda_2^2}}{-2\left(1 - \frac{\kappa^2\overline{\lambda}^2}{4}\right)}. \tag{B17}$$

Hence, the error $|e(k-1)|$ should satisfy the following equality:

$$|e(k-1)| > r_{e1} \tag{B18}$$

in order to make $\Delta V(k) < \Delta\overline{V}(k) < 0$. Then, the error will converge to the ball centered at the origin of the error space with radius r_{e1} in (B16).

References

[1] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[3] K. S. Narenda and M. A. L. Thathachar, *Learning Automata—An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[4] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[5] J. Park and I. W Sandberg, "Universal approximation using RBF networks," *Neural Comput.*, vol. 3, pp. 246–257, 1991.
[6] M. Mueller, "Least-squares algorithms for adaptive equalizers," *Bell Syst. Tech. J.*, vol. 60, pp. 1905–1925, 1981.
[7] S. Ljung and L. Ljung, "Error propagation properties of least squares adaptation algorithms," *Automatica*, vol. 21, pp. 157–167, 1985.
[8] D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commum.*, vol. COMM-26, pp. 1439–1446, Oct. 1978.
[9] M. Verhaegen, "Roundoff error propagation in four generally applicable, recursive least squares estimation schemes," *Automatica*, vol. 25, no. 3, pp. 437–444, 1989.
[10] M. M. Chansarkar and U. B. Desai, "A robust recursive least squares algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 1726–1735, July 1997.
[11] M. ZhiHong, H. R. Wu, W. Lai, and T. Nguyen, "Design of adaptive filters using Lyapunov stability theory," in *Proc. 6th IEEE Int. Workshop Intelligent Signal Processing and Communication Systems*, vol. 1, 1998, pp. 304–308.
[12] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[13] C. F. Fung, A. B. Steve, and W. Luo, "On-line supervised adaptive training using radial basis function networks," *Neural Netw.*, vol. 9, no. 9, pp. 1597–1617, 1996.
[14] S. A. Billings and C. F. Fung, "Recurrent radial basis function networks for adaptive noise cancellation," *Neural Netw.*, vol. 8, no. 3, pp. 273–290, 1995.
[15] S. Haykin, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Trans. Signal Processing*, vol. 43, pp. 526–535, Feb. 1995.
[16] P. Yee and S. Haykin, "A dynamic regularized RBF networks for nonlinear, nonstationary time series prediction," *IEEE Trans. Signal Processing*, vol. 47, pp. 2503–2521, Sept. 1999.
[17] J. Baltersee and A. Joonathon, "Nonlinear adaptive prediction of speech with a pipelined recurrent neural network," *IEEE Trans. Signal Processing*, vol. 46, pp. 2207–2216, Aug. 1998.
[18] Institute for Integrated Signal Processing Systems.. ISS, Aachen, Germany. [Online]. Available: http://www.ert.rwth-aachen.de/Presonen/balterse.html