Adaptive Noise Cancellation with Fast Tunable RBF Network

Hao Chen, Yu Gong and Xia Hong
School of Systems Engineering, University of Reading, Reading, RG6 6AY, UK
E-mail: hao.chen@pgr.reading.ac.uk, {y.gong, x.hong}@reading.ac.uk

***Abstract*— This paper describes a novel adaptive noise cancellation system with fast tunable radial basis function (RBF). The weight coefficients of the RBF network are adapted by the multi-innovation recursive least square (MRLS) algorithm. If the RBF network performs poorly despite of the weight adaptation, an insignificant node with little contribution to the overall performance is replaced with a new node without changing the model size. Otherwise, the RBF network structure remains unchanged and only the weight vector is adapted. The simulation results show that the proposed approach can well cancel the noise in both stationary and nonstationary ANC systems.**

I. Introduction

The adaptive noise cancellation (ANC) [1] provides an efficient method to recover signal from noise in many applications including communications, echo cancellation, speech enhancement, antenna array processing, magnetic resonance imaging (MRI) and variety of electrocardiograms (ECGs) processing applications [2]–[5].

In the ANC system, adaptive algorithms such as the widely used least mean square (LMS) [6] and recursive least square (RLS) algorithms [7] are often used to "learn" the noise model from the input data or track the model variation. Both linear and non-linear ANC have been proposed. While the linear ANC provides a simple solution to some problems, it is often not adequate in practical systems which usually exhibit non-linear and non-stationary behaviors. Non-linear ANC systems are thus of particular importance.

The non-linear ANC systems have been well investigated. It is known that the ANC can be equivalent to the benchmark problem of system identification [8]. Therefore, many typical non-linear system identification approaches can be used in the ANC system. Of particular interest is the radial basis function (RBF) due to it simplicity and the ability to approximate any continuous function to an arbitrary degree of accuracy [9]. In the RBF network, not only do the outer layer weight coefficients need to be constantly adapted to track the variation of the in-coming data, but also the structure of the RBF network (including the number of nodes and the centres and width of each node) must also be carefully chosen. In [10], a recurrent RBF network was described for the ANC system, in which the centres of the RBF networks are chosen by the k-means clustering method. In [11], the ANC system with the enhanced dynamic fuzzy neural networks (EDFNN) learning approach (which is based on the DFNN algorithm for extended RBF networks [12]) was proposed where the number of RBF neurons and input/output space clustering can be adaptively determined.

Incremental RBF based on-line learning algorithms [13]–[15] with growing/pruning model size are well suitable for the ANC problem. These include the resource allocating network (RAN) [13], where the network model starts empty and grows with the input data based on the *nearest neighbor* method. Further in [14], a more compact model can be achieved by using the minimal RAN (M-RAN) algorithm which prunes the inactive kernel nodes based on relative contribution. More computationally efficient growing-and-pruning RBF (GAP-RBF) algorithm [15] was then proposed, in which only the nearest RBF node is considered for the model growing and pruning. The growing and pruning are based on the "significance" of the nodes which has a direct link to the learning accuracy. Some of incremental RBF approaches and its variants have been applied for ANC problem and presented a good performance, such as MRAN and direct link MRAN (DMRAN) in [16], [17]. While all of these RAN-based approaches can be used in the ANC system with adjustable model size, a common problem is that the structure of each individual node is not optimized. This makes the model size often increase with the number of the sample data, ending up with a very large ANC model, particularly so in the non-stationary environment where the noise model keeps changing with time [18]. This motivates us to propose a novel RBF model for the ANC system which can track the variation of the noise model with compact model size.

We understand that, in the non-stationary ANC system, because the noise model keeps varying, the "local characteristic" of the input data is more relevant than the "global characteristic" [19], [20]. This implies that the model size needs not to be large since the modeling needs to focus on the recent noise model, but not the older ones. Therefore, in this paper, we propose to fix the model size at a small number, and each RBF node has a tunable centre vector and an adjustable diagonal covariance matrix which can be optimized online one at a time. At each time step, the RBF weights are adapted using the RLS algorithm, and the modeling performance is monitored. If the RBF network performs poorly despite the weight adaptation, an insignificant node with little contribution to the overall system is identified and replaced by a new node

without changing the model size. The structural parameters of the new node including the centre vector and diagonal covariance matrix are iteratively optimized. Because the RBF network has tunable nodes, the model size can be much smaller than that of a conventional RBF network due to its structural flexibility. The algorithm is initially described for general on-line modeling systems in our recent submission [21], and is applied here particularly for the ANC system. Simulation results show that the proposed approach works very well in the ANC system both in stationary and non-stationary environment.

II. ANC SYSTEM WITH RBF

A typical ANC system is shown in Fig. 1, where the desired signal $s(t)$ is corrupted by a uncorrelated noise $n(t)$ to give the measured signal $d(t)$ in the primary channel, and the reference noise $x(t)$ is generated by passing $n(t)$ through a secondary channel with unknown transfer function $T(.)$. The task of the ANC is to adaptively adjust $F(.)$, based on the system residual $\varepsilon(t)$, to cancel the noise embedded in $d(t)$.

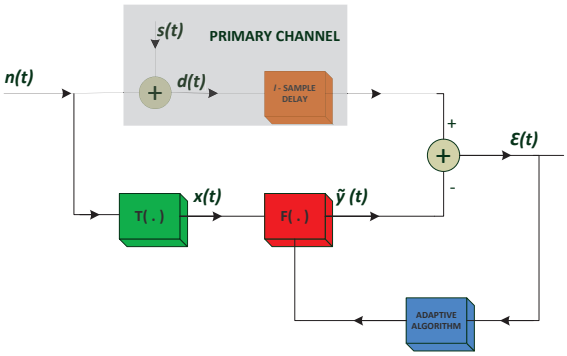

Fig. 1. The ANC System.

The ANC system can be equivalent to the system identification model as is shown in Fig. 2, where $x(t)$ is regarded as the system input, $d(t-l)$ is the measured system output with a delay l, $\hat{y}(t)$ is the prediction of the designed filter $F(.)$ by using $x(t)$ as the input. The task now is to adjust $F(.)$ to approximate $T^{-1}(.)$. Or the optimum $F(.)$ is obtained by minimizing the mean-square-ANC-output of $E[\varepsilon^2(t)]$, where $\varepsilon(t) = d(t-l) - \hat{y}(t)$

In this paper, we use the RBF network to realize $F(.)$ in Fig. 2. To be specific, there are M number of nodes. At time t, the input vector of the RBF network is given by

$$\mathbf{x}_t = [x_t(1), x_t(2), \cdots, x_t(N_x)]^{\mathrm{T}} \tag{1}$$

where N_x is the model input dimension or the number of input channels, and $x_t(i)$ is the input data from the ith input channel at time t.

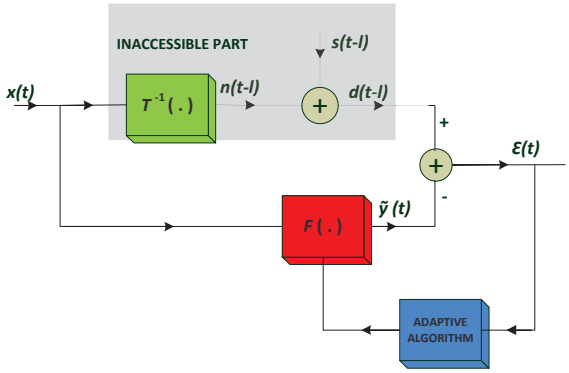

Fig. 2. System identification equivalent to the ANC.

The RBF network output is given by

$$f(\mathbf{x}_t) = \sum_{i=1}^{M} w_{t-1}(i) g_i(\mathbf{x}_t) \tag{2}$$

where $g_i(\mathbf{x}_t)$ is the output of the ith node, $w_{t-1}(i)$ is the weight coefficient for the ith node at time $t-1$. Letting $\mathbf{w}_{t-1} = [\omega_{t-1}(1), \cdots, \omega_{t-1}(M)]^{\mathrm{T}}$ be the weight vector and $\boldsymbol{\phi}_t = [g_1(\mathbf{x}_t), \cdots, g_M(\mathbf{x}_t)]^{\mathrm{T}}$ be the information vector, we can rewrite (2) as

$$f(\mathbf{x}_t) = \boldsymbol{\phi}_t^{\mathrm{T}} \mathbf{w}_{t-1} \tag{3}$$

The RBF residual error at time t is given by

$$\varepsilon(t) = y_t - \boldsymbol{\phi}_t^{\mathrm{T}} \mathbf{w}_{t-1} \tag{4}$$

where y_t the observed system output at time t. In this paper, we assume without losing generality that the RBF basis function is Gaussian so that

$$g_i(\mathbf{x}_t) = \exp\left(-(\mathbf{x}_t - \mathbf{c}_i)^{\mathrm{T}} \mathbf{H}_i (\mathbf{x}_t - \mathbf{c}_i)\right) \tag{5}$$

where $\mathbf{c}_i = [c_i(1), \cdots c_i(N_x)]^{\mathrm{T}}$ which is the centre vector of the ith node, $\mathbf{H}_i = \mathrm{diag}\{\sigma_i^2(1), \cdots, \sigma_i^2(N_x)\}$ which is the diagonal covariance matrix of the ith node, $c_i(j)$ and $\sigma_i^2(j)$ are the jth centre and variance of the ith RBF node respectively.

III. TUNABLE RBF NETWORK

At each time step, the weight vector is adapted by the RLS algorithm. The residual error of the network output $\varepsilon(t)$ is monitored. If the RBF network performs poorly, or the residual error is large, despite of the weight adaptation, an insignificant node with little contribution to the overall system is replaced with a new node without changing the model size. The structure of the new node is then optimized to "fit" for the current input data. At the same time, the weight vector should also be updated as otherwise it is only suitable for the old network before the node replacement.

Below we describe fast algorithms for the node structure and weight vector adaptation. We assume without losing generality that, at time t, the Mth node is replaced with a new node.

While the joint optimization of the structure of the new node and the weight vector is too complicated, we propose an iterative adaptation approach. At every iteration, either the structure of the new node or the weight vector is fixed first, and the other is updated, and vice versa. The iteration continues until the modelling residual error is sufficiently small or the maximum iteration number is reached.

A. Iterative node optimization

At every iteration, when the weight vector is fixed, the centre and the inverse of the variance for the new node are tuned by minimizing the 2-norm of the instantaneous error. The instantaneous error at time t can be expressed as

$$e_t = y_t - \sum_{i=1}^{M-1} w_{t-1}(i) g_i(\mathbf{x}_t) - w_{t-1}(M) g_M(\mathbf{x}_t), \tag{6}$$

From (5), the kernel function for the new node $g_M(\mathbf{x}(t))$ is obtained as

$$g_M(\mathbf{x}_t) = \exp\left(-\sum_{j=1}^{N_x} \eta_M(j) \cdot (x_t(j) - c_M(j))^2 \right), \tag{7}$$

where $\eta_M(j) = 1/\sigma_M^2(j)$ which is the inverse of the variance for the jth input of the Mth node. While the model size and the structure of the remaining nodes remain unchanged, the structural vector for adaptation is defined as

$$\Gamma = [c_M(1), \cdots, c_M(N_x), \eta_M(1), ..., \eta_M(N_x)]^{\mathrm{T}} \tag{8}$$

The optimum Γ with fixed weight vector to minimizes e_t^2 can be found by the gradient descent search as

$$\Gamma_l = \Gamma_{l-1} - \lambda \frac{\boldsymbol{\nabla}}{\|\boldsymbol{\nabla}\|} e_t, \tag{9}$$

where subscript l represents the iteration step l, λ is a small positive step size, $\|.\|$ denotes Euclidian norm, and $\boldsymbol{\nabla}$ is the gradient vector which is given by

$$\boldsymbol{\nabla} = \frac{\partial(e_t)}{\partial \Gamma} = \left[\frac{\partial e_t}{\partial c_M(1)}, \cdots, \frac{\partial e_t}{\partial c_M(N_x)}, \frac{\partial e_t}{\partial \eta_M(1)}, \cdots \frac{\partial e_t}{\partial \eta_M(N_x)} \right]^{\mathrm{T}} \tag{10}$$

From (7), we can easily obtain that

$$\begin{cases} \frac{\partial e_t}{\partial c_M(i)} = -2\eta_M(i) w_{t-1}(M) g_M(\mathbf{x}_t)[(x_t(i) - c_M(i))] \\ \frac{\partial e_t}{\partial \eta_M(j)} = w_{t-1}(M) g_M(\mathbf{x}_t) \left[x_t(j) - c_M(j)\right]^2 \end{cases} \tag{11}$$

The computational cost of the above gradient descent search is in the order of $O(N_x)$ scaled by the iteration number.

B. Fast weight adaptation

At every iteration, when the node structure is fixed, the weight vector is calculated by the least-square (LS) method based on the recent input data. It is reasonable to use only the recent input data for the LS weight estimation because, when the node replacement occurs, the underlying system varies significantly (otherwise no node is replaced), and the previous data should be "forgotten" in order to capture the "local" characteristic. The LS estimate of $\mathbf{w}_t$ based on recent q input vectors is obtained by minimizing the least squares cost function as

$$J_t = \sum_{j=t}^{t-q+1} e_t^2(j) = \mathbf{e}_t^{\mathrm{T}} \mathbf{e}_t = (\mathbf{y}_t - \boldsymbol{\Phi}_t \mathbf{w}_t)^{\mathrm{T}} (\mathbf{y}_t - \boldsymbol{\Phi}_t \mathbf{w}_t), \tag{12}$$

where $\boldsymbol{\Phi}_t$ is a $q \times M$ matrix which is obtained similar to (3), and $\mathbf{y}_t$ consists of recent q number of observed system output. For a given $g_M(.)$, the LS solution of (12) is given by

$$\mathbf{w}_t = \mathbf{P}_t \boldsymbol{\Phi}_t^{\mathrm{T}} \mathbf{y}_t, \tag{13}$$

where $\mathbf{P}_t = (\boldsymbol{\Phi}_t^{\mathrm{T}} \boldsymbol{\Phi}_t)^{-1}$. For a given structure of the new node $g_M(.)$, (13) is the closed loop solution for $\mathbf{w}_t$. This further indicates that the structure (the centres and variances) of the new node and the weight vector can be optimized iteratively. We note that, in the iterative approach, (13) is performed at every iteration with the same input data but for the latest node structure. It is clear that the main computation cost in the weight adaptation comes from the matrix inversion $\mathbf{P}_t$. Fortunately, in the proposed scheme, only one insignificant node is replaced at a time without changing the model size. This means that only the last column of $\boldsymbol{\Phi}_t$ is changed at every iteration. Exploiting this property can significantly reduce the calculation of $\mathbf{P}_t$, which is accomplished by making use of the inverse of matrix block decomposition lemma to avoid the repetitive matrix inversions $\mathbf{P}_t$ at every iteration. To be specific, we can re-write $\boldsymbol{\Phi}_t$ and $\mathbf{P}_t$ in the forms of

$$\boldsymbol{\Phi}_t = [\boldsymbol{\Phi}_{t,-M} \quad \mathbf{g}_M] \tag{14}$$

and

$$\mathbf{P}_t = \begin{bmatrix} \boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \boldsymbol{\Phi}_{t,-M} & \boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \mathbf{g}_M \\ \mathbf{g}_M^{\mathrm{T}} \boldsymbol{\Phi}_{t,-M} & \mathbf{g}_M^{\mathrm{T}} \mathbf{g}_M \end{bmatrix}^{-1} \tag{15}$$

respectively. Letting $\mathbf{P}_{t,-M} = \left(\boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \boldsymbol{\Phi}_{t,-M}\right)^{-1}$, and applying the inverse of matrix block decomposition lemma, we obtain

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^{\mathrm{T}} & c \end{bmatrix} \tag{16}$$

where

$$\begin{cases} \mathbf{A} &= \mathbf{P}_{t,-M} + \frac{1}{\kappa} \mathbf{P}_{t,-M} \boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \mathbf{g}_M \mathbf{g}_M^{\mathrm{T}} \boldsymbol{\Phi}_{t,-M} \mathbf{P}_{t,-M} \\ \mathbf{b} &= -\frac{1}{\kappa} \mathbf{P}_{t,-M} \boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \mathbf{g}_M \\ c &= \frac{1}{\kappa} \end{cases} \tag{17}$$

where $\kappa = \mathbf{g}_M^{\mathrm{T}} \mathbf{g}_M - \mathbf{g}_M^{\mathrm{T}} \boldsymbol{\Phi}_{t,-M} \mathbf{P}_{t,-M} \boldsymbol{\Phi}_{t,-M}^{\mathrm{T}} \mathbf{g}_M$. Because now only the structure of the Mth node is adapted and the structures for all others nodes remain unchanged, $\mathbf{P}_{t,-M}$ needs only be calculated once for every node replacement. Note that $\mathbf{g}_M$ needs to be iteratively calculated for the node optimization, so the computation cost for the LS weight calculation is $O(p)$ scaled by the number of iterations. There is also an one off computational cost to calculate $\mathbf{P}_{t,-M}$ which

is in the order of $O((M-1)^3)$.

IV. Simulations

In this section, computer simulations are given to compare the proposed algorithm [21] with some typical approaches including the RAN and GAP-RBF algorithms [13], [15]. All these approaches (including the proposed one) apply Gaussian nodes. In the proposed approach, the number of nodes is fixed at 10, the innovation length is simply set as $p = 1$ as a classic RLS algorithm. The performance for different approaches are compared based on the mean square error (MSE).

In this paper, the non-linear channel $T(.)$ in Fig.1 is described as

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + b_1 n(t-1) + b_2 n(t-2) + b_3 n(t-3) + c_1 n^2(t-2) + c_2 n(t-2)x(t-1), \quad (18)$$

where a_i, b_j and c_k are some coefficients which will be set below. The modelling input vector to $F(.)$ is given by

$$\mathbf{u}(t) = [x(t), x(t-1), x(t-2), \hat{y}(t-1), \hat{y}(t-2)]^{\mathrm{T}} \quad (19)$$

A. Non-linear stationary ANC problem

First we consider a stationary ANC model, where the parameters of the non-linear channel in (18) are fixed as $a_1 = 0.25$, $a_2 = 0.1$, $b_1 = 0.5$, $b_2 = 0.1$, $b_3 = -0.2$, $c_1 = 0.2$ and $c_2 = 0.08$. The desired signal is a sawtooth signal of unit magnitude with period of 50 samples.

Fig. 3 (a) and (b) show the corrupted signal $d(t)$ and the recovered signal $\varepsilon(t)$ during the last 500 samples for the proposed approach respectively, where it is clearly shown that the noise has been significantly cancelled.

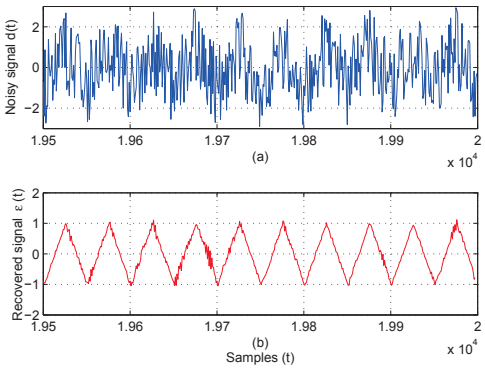

Fig. 3. **Stationary model**: Noisy *vs* recovered signals.

Fig. 4 compares the MSE learning curves for different approaches. The GAP-RBF has clearly better MSE performance than the RAN, and it also requires much fewer nodes. This is because the GAP-RBF can both grow and prune the model size, while the RAN can only increase the model. It is clear that the proposed approach has the best MSE performance with only 10 nodes.

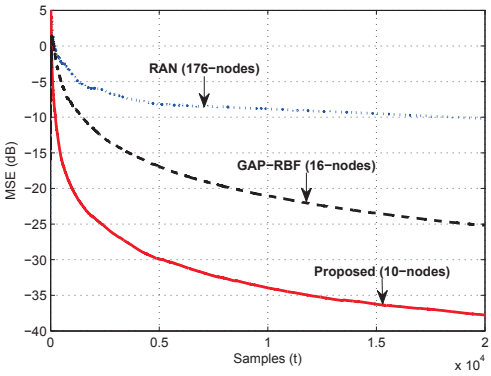

Fig. 4. **Stationary model**: MSE learning curves.

B. Non-linear and non-stationary ANC problem

In this simulation, we let the non-linear channel $T(.)$ vary with time. To be specific, the first coefficient in (18) is set varying with time as

$$a_1 = 0.2 + \left| 0.5 \cdot \sin\left(\frac{t}{1000\pi}\right) \right|, \quad (20)$$

and other coefficients are the same as those in stationary ANC model. The desired signal is also the sawtooth signal as the stationary case.

Fig. 5 (a) and (b) show the corrupted signal $d(t)$ and the recovered signal $\varepsilon(t)$ during the last 500 samples for the proposed approach respectively. It is clearly shown that the signal can still be well extracted from buried noise when the noise model is time varying.

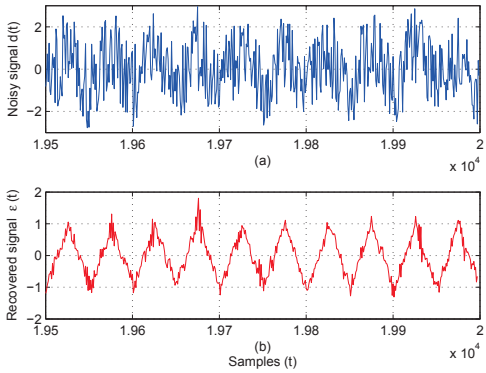

Fig. 5. **Nonstationary model**: Noisy *vs* recovered signals.

Fig. 6 compares the MSE leaning curves for different approaches. While the proposed approach has clearly the best performance, it is interesting to observe that both the RAN and GAP-RBF use significantly more nodes in the non-stationary case than in the stationary case, where particularly, the RAN algorithm ends up with a model size of 331.

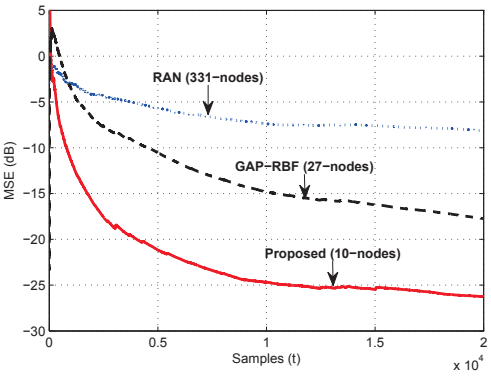

Fig. 6. **Nonstationary model**: MSE learning curves.

V. Conclusion

In this paper we propose a fast self-structured adaptive RBF network for the ANC system. The model size of the RBF network is fixed at a small number in order to capture the local characteristic of the non-stationary systems. The weight vector is normally adapted by the RLS while the modelling performance is monitored every time step. If the modelling residual becomes high, an insignificant node is replaced with a new node, of which the structural parameters and weights are optimized using proposed fast iterative algorithms including the gradient decent algorithm and LS estimator. The proposed approach is verified by numerical simulations in both stationary and nonstationary systems.

Acknowledgement

This research is sponsored by the UK Engineering and Physical Sciences Research Council and DSTL under the grant number EP/H012516/1.

References

[1] B. Widrow, J. Glover, J.R., J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, J. Eugene Dong, and R. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692 – 1716, Dec. 1975.

[2] E. R. Ferrara and B. Widrow, "Fetal electrocardiogram enhancement by time-sequenced adaptive filtering," *Biomedical Engineering, IEEE Transactions on*, vol. BME-29, no. 6, pp. 458 –460, Jun. 1982.

[3] M. Yelderman, B. Widrow, J. M. Cioffi, E. Hesler, and J. A. Leddy, "ECG enhancement by adaptive cancellation of electrosurgical interference," *Biomedical Engineering, IEEE Transactions on*, vol. BME-30, no. 7, pp. 392 –398, Jul. 1983.

[4] N. Thakor and Y.-S. Zhu, "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection," *Biomedical Engineering, IEEE Transactions on*, vol. 38, no. 8, pp. 785 –794, Aug. 1991.

[5] C. Chen, T.-D. Chiueh, and J.-H. Chen, "Active cancellation system of acoustic noise in MR imaging," *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 2, pp. 186 –191, Feb. 1999.

[6] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, Summer 1989.

[7] M. Birgmeier, "A fully kalman-trained radial basis function network for nonlinear speech modeling," in *Neural Networks, 1995. Proceedings, IEEE International Conference on*, vol. 1, Nov./Dec. 1995, pp. 259 – 264.

[8] B. Friedlander, "System identification techniques for adaptive noise cancelling," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, no. 5, pp. 699 – 709, Oct. 1982.

[9] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, Summer 1991.

[10] S. A. Billings and C. F. Fling, "Recurrent radial basis function networks for adaptive noise cancellation," *Neural Networks*, vol. 8, no. 2, pp. 273 – 290, 1995.

[11] M. J. Er, Z. Li, H. Cai, and Q. Chen, "Adaptive noise cancellation using enhanced dynamic fuzzy neural networks," *Fuzzy Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 331 – 342, Jun. 2005.

[12] S. Wu and M. J. Er, "Dynamic fuzzy neural networks-a novel approach to function approximation," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 30, no. 2, pp. 358 –364, Apr. 2000.

[13] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, Summer 1991.

[14] L. Yingwei, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function neural networks," *Neural Computation*, vol. 9, no. 2, pp. 461–478, Feb. 1997.

[15] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 6, pp. 2284 –2292, Dec. 2004.

[16] S. Yonghong, P. Saratchandran, and N. Sundararajan, "Minimal resource allocation network for adaptive noise cancellation," *Electronics Letters*, vol. 35, no. 9, pp. 726 –728, Apr. 1999.

[17] ——, "A direct link minimal resource allocation network for adaptive noise cancellation," *Neural Process. Lett.*, vol. 12, no. 3, Dec. 2000.

[18] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2165 – 2176, Aug. 2004.

[19] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[20] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Phys. Rev. Lett.*, vol. 59, no. 8, pp. 845–848, Aug. 1987.

[21] H. Chen, Y. Gong, X. Hong, and S. Chen, "A fast adaptive tunable RBF network for nonstationary systems," *Neural Networks and Learning Systems, IEEE Transactions on*, 2012, (Submitted).