[redacted]

Adaptive Noise Cancellation Using Recurrent Radial Basis Function Networks

S A Billings and C F Fung
Department of Automatic Control and Systems Engineering
University of Sheffield
P O Box 600
Mappin Street
Sheffield
S1 4DU

Research Report No 472

March 1993

Adaptive Noise Cancellation Using Recurrent Radial Basis Function Networks

S. A. BILLINGS AND C. F. FUNG

Department of Automatic Control and Systems Engineering

University of Sheffield

P. O. Box 600, Mappin Street

Sheffield S1 4DU

U. K.

Abstract

Radial basis function neural network architectures are introduced for the nonlinear adaptive noise cancellation problem. Both FIR and IIR filter designs are considered and it is shown that by exploiting the duality with system identification that the nonlinear IIR filter can be configured as a recurrent radial basis function network. Details of network training which is based on a combined k-means clustering and Givens routine, the inclusion of linear dynamic network links and metrics for performance monitoring are also discussed. Examples are included to demonstrate the degree of noise suppression that can be achieved based on the new design.

I. INTRODUCTION

EXTRACTION of a signal buried in noise is one of the benchmark problems in the area of signal processing. Adaptive noise cancellation provides a solution to this problem without incurring the computational expense associated with an optimal Kalman design. Examples where noise cancellation can be of benefit include the cancellation of broad band interference in the sidelobes of an antenna array, periodic interference in speech signals, various forms of interference in electrocardiographs and the elimination of tape hum or turntable rumble during playback.

When the signal and noise lie in different frequency bands with significant separation band-pass filtering can be applied to remove the major portion of the noise power and restore the signal of interest. Unfortunately, this idealistic situation rarely occurs in practice and either an optimal Kalman design or noise cancellation must be implemented. The principle of noise cancellation is based on the assumption that both the noisy signal and a filtered or distorted measurement of the noise are available. If the inverse of the filtered noise distortion can be estimated then the noise corrupting the signal can be predicted and consequently the noise can be canceled.

Noise cancellation has been widely studied in the literature but most of the emphasis has been on linear FIR designs implemented using the LMS algorithm [1][2]. FIR filters are constructed using lagged input variables only whereas IIR filters are based on both past inputs and outputs. IIR designs therefore involve fewer regressed variables and associated coefficients but usually at the expense of increased estimation complexity involving recursive least squares type estimators [3].

Adaptive linear designs can of course be applied to track and cancel nonlinear noise distortion but only if the nonlinearity is mild or the operating point changes relatively slowly. The development of nonlinear noise cancellation algorithms would therefore be desirable for many applications but only a few authors have addressed this problem. Most of the emphasis has been on Volterra series filters [4][5]. These can be considered as an extension of the classical linear FIR deigns but with both lagged linear inputs and polynomial functions of these. The advantage of this approach is that the linear FIR concepts can be utilized directly simply by extending the regression vector to include polynomial input terms. The disadvantage is that the number of filter parameters required can easily become very large [6]. For example to obtain a reasonable estimate of the first order kernel or impulse response function requires, typically, the estimation of 30 coefficients. The second order Volterra kernel will require the estimation of $(30 \times 30)/2$ points taking account of symmetry. So that even a simple system involving just a quadratic nonlinearity in the input and therefore only two Volterra kernels will require the estimation of almost 500 filter coefficients. If the system involves a nonlinear function of the output signal the third, fourth and higher order Volterra kernels may be required, in addition to the first two mentioned above, and so the number of filter coefficients required can become explosive. Whilst the number of coefficients required in linear FIR designs is often manageable nonlinear designs based on this approach can quickly become untenable because of the curse of dimensionality.

A lot of these problems can however be overcome by using nonlinear IIR filter designs based on the Nonlinear AutoRegressive Moving Average model with eXogenous inputs (NARMAX) [7][8][9]. The advantage of the NARMAX IIR design is that the use of lagged output variables reduces, often considerably, the number of coefficients which are required for an effective design. A system with a square law term in the output for example would necessitate hundreds and possibly thousands of coefficients if this were realized as an FIR filter. Whereas a NARMAX IIR design could accommodate the squared output term with just one extra coefficient. The balance between the simple estimation but excessive parameter set of the FIR case and the more complex estimation but considerably fewer coefficients of the IIR shifts decisively in favor of the NARMAX IIR design therefore when nonlinear systems are considered [10].

In the present study radial basis function (RBF) network architectures are introduced for the noise cancellation problem. It is shown that interpreting the noise cancellation concept in terms of a system identification problem [11][12] provides a framework for the design of neural network architectures which provide a solution to this problem. A nonlinear FIR design is considered initially. This is then extended by using a recurrent radial basis function network to the nonlinear IIR filter design case. Training of the network is achieved using a combined k-means clustering and Givens routine [13] and the effects of introducing linear dynamic network links is shown to be benefited. Model validation procedures are considered as a means of determining the effectiveness of the new filter designs and several examples are included to demonstrate the performance of the new designs.

II. LINEAR NOISE CANCELLATION FILTERS

The typical structure of an adaptive noise cancellation system is depicted in Fig. 1. As shown, in the primary channel, the signal of interest $s(i)$ is corrupted by an additive noise $n(i)$ at time step i resulting in the noisy signal $d(i)$. Assume that $\{s(i)\}$ and $\{n(i)\}$ are uncorrelated. The reference noise $x(i)$ is generated by feeding $n(i)$ through a measuring device of unknown transfer function $T(\cdot)$. By using $x(i)$ as an input to the filter $F(\cdot)$, an estimate $\hat{y}(i)$ of the noise can be obtained at the output of $F(\cdot)$. The recovered signal $\varepsilon(i)$, which is also the error between the noisy signal and the filter output, can be reconstructed based on this estimate and ideally this should be equal to $s(i-l)$. It can be shown that an optimal design yields the parameter set of $F(\cdot)$ such that the mean-square-error $E\{\varepsilon^2(i)\}$ is minimized [9]. For a physically realizable system, $F(\cdot)$ must be causal so that a number of delayed samples must be inserted in the primary channel. For an l-sample delay system, the output of the optimal filter $F(\cdot)$ will be

$$\hat{y}(i) = n(i-l) = T^{-1}(x(i)) = F(x(i)). \tag{1}$$

But, since $d(i) = n(i) + s(i)$ then

$$d(i-l) = n(i-l) + s(i-l) = T^{-1}(x(i)) + s(i-l) \tag{2}$$

such that

$$\varepsilon(i) = d(i-l) - \hat{y}(i) = s(i-l). \tag{3}$$

For a linear design of noise canceller, the structure of $F(\cdot)$ can be implemented using either an FIR or IIR filter design.

II.1. FIR Designs

The main feature of the FIR design is the all zero structure which gives rise to the property of global stability. Because of this advantage the FIR filter has been widely utilized in signal processing applications. The least mean squares (LMS) algorithm is usually employed for updating the FIR filter parameters although the recursive least squares (RLS) routine has recently been considered by some authors as an alternative solution [14][15].

Using the LMS algorithm the filter parameters can be updated at minimal computational cost but the convergence will generally be much slower than that of the RLS algorithm. The RLS algorithm exhibits a superior convergence rate at the expense of higher computational complexity. However, many fast variants of the RLS algorithm have been derived to minimize the computational burden [16][17][18][19][20].

When $F(\cdot)$ in Fig. 1 is chosen to be a linear FIR filter with transfer function

$$A\left(z^{-1}\right)=\sum_{j=0}^{N_x-1} a_j z^{-j} \tag{4}$$

the noisy signal can be expressed as

$$d(i-l)=F(x(i))+s(i-l)=A\left(z^{-1}\right)x(i)+s(i-l) \tag{5}$$

or in matrix form

$$d(i-l)=\Phi_x^{\mathrm{T}}(i)\Theta_x+s(i-l) \tag{6}$$

where

$$\Phi_x(i)=\left[x(i),\cdots,x(i-N_x+1)\right]^{\mathrm{T}} \tag{7}$$

and

$$\Theta_x=\left[a_0,\cdots,a_{N_x-1}\right]^{\mathrm{T}}. \tag{8}$$

Following the standard noise cancellation problem formulation and assuming that $\{s(i)\}$ and $\{x(i+k)\}$ are zero mean and uncorrelated for all integer k, either an LMS or a RLS algorithm can be applied to yield unbiased estimates of Θ_x in Eqn. (8).

II.2. IIR Designs

Linear IIR filters are based on a pole zero structure and therefore allow regressed input and output terms. The advantage of regressing the past output terms is that this often results in a considerable reduction in the size of the parameter set compared to an FIR design. But this has to be offset by the fact that the stability of an IIR filter can no longer be guaranteed and an inattentive parameter updating strategy may mistakenly cause the movement of a pole outside the unit circle and produce a divergence in filter parameters.

A full discussion of these properties including methods for monitoring the parameter updates and resetting any unstable effects is available in the literature [21][22][23]. Noise cancellation can be reconfigured as a system identification problem. Fig. 2 shows that with this interpretation $d(i-l)$ is the measured output, $x(i)$ the input, $s(i-l)$ an additive colored noise source and the identification problem involves adjusting $F(\cdot)$ to approximate $T^{-1}(\cdot)$. When $F(\cdot)$ is realized using an IIR filter with transfer function

$$\frac{A\left(z^{-1}\right)}{B\left(z^{-1}\right)}=\frac{\sum_{j=0}^{N_x-1} a_j z^{-j}}{1-\sum_{k=1}^{N_d} b_k z^{-k}}. \tag{9}$$

the noise corrupted signal can be expressed as

$$d(i-l)=\frac{A\left(z^{-1}\right)}{B\left(z^{-1}\right)}x(i)+s(i-l) \tag{10}$$

Eqn. (10) can be expanded and rearranged to yield

$$B\left(z^{-1}\right)d(i-l)=A\left(z^{-1}\right)x(i)+B\left(z^{-1}\right)s(i-l) \tag{11}$$

so that

$$d(i-l)=\Phi_x^{\mathrm{T}}(i)\Theta_x+\Phi_d^{\mathrm{T}}(i-l-1)\Theta_d-\Phi_s^{\mathrm{T}}(i-l-1)\Theta_d+s(i-l) \tag{12}$$

where

$$\Phi_d(i-l-1)=\left[d(i-l-1),\cdots,d(i-l-N_d)\right]^{\mathrm{T}} \tag{13}$$

$$\Phi_s(i-l-1)=\left[s(i-l-1),\cdots,s(i-l-N_d)\right]^{\mathrm{T}} \tag{14}$$

and

$$\Theta_d=\left[b_1,\cdots,b_{N_d}\right]^{\mathrm{T}} \tag{15}$$

By defining the matrices

$$\Phi_{xd}(i)=\begin{bmatrix}\Phi_x(i)\\ \Phi_d(i-l-1)\end{bmatrix} \tag{16}$$

and

$$\Theta_{xd}=\begin{bmatrix}\Theta_x\\ \Theta_d\end{bmatrix} \tag{17}$$

Eqn. (12) can be written as

$$d(i-l)=\Phi_{xd}^{\mathrm{T}}(i)\Theta_{xd}-\Phi_s^{\mathrm{T}}(i-l-1)\Theta_d+s(i-l). \tag{18}$$

Inspection of Eqn. (18) shows that the noise term will be colored and biased estimates of the parameters Θ_{xd} will be obtained if a standard RLS routine is applied. This problem often occurs in system identification and several recursive schemes such as extended least squares (ELS) [24], instrumental variables (IV) [25] and suboptimal least squares (SOLS) [26] have been designed to yield unbiased estimates in the presence of correlated noise. Among the various proposed schemes the SOLS algorithm provides the best match with the requirements of linear noise cancellation problems and can readily be extended to accommodate the nonlinear case [14].

If $n(i-l)$ could be monitored, Eqn. (18) could be rewritten as

$$d(i-l)=\Phi_x^{\mathrm{T}}(i)\Theta_x+\Phi_n^{\mathrm{T}}(i-l-1)\Theta_d+s(i-l) \tag{19}$$

where

$$\Phi_n(i-l-1)=\left[n(i-l-1),\cdots,n(i-l-N_d)\right]^{\mathrm{T}}. \tag{20}$$

By defining the matrix

$$\Phi_{xn}(i)=\begin{bmatrix}\Phi_x(i)\\ \Phi_n(i-l-1)\end{bmatrix} \tag{21}$$

Eqn. (19) can be written as

$$d(i-l)=\Phi_{xn}^{\mathrm{T}}(i)\Theta_{xd}+s(i-l) \tag{22}$$

Inspection of Eqn. (22) shows that now the signal of interest is no longer correlated with the elements of $\Phi_{xd}(i)$ and this is the key idea of the SOLS algorithm. In practice $n(i-l)$ is not available and so this has to be replaced by an estimate computed recursively as

$$\hat{y}(i)=\sum_{j=0}^{N_x-1}\hat{a}_j(i)x(i-j)+\sum_{k=1}^{N_d}\hat{b}_k(i)\hat{y}(i-k) \tag{23}$$

where $\hat{a}_j(i)$ and $\hat{b}_k(i)$ are the estimates of a_j and b_k at time step i respectively, $\hat{y}(i)$ denotes the prediction of $n(i-l)$ at time step i. The noise components contained in the regression vector are therefore replaced by the corresponding estimates of the noise free signals.

The basis of the SOLS algorithm therefore becomes the regression equation

$$d(i-l)=\hat{y}(i)+s(i-l)=\Phi_{x\hat{y}}^{\mathrm{T}}(i)\Theta_{x\hat{y}}(i)+s(i-l) \tag{24}$$

where

$$\Phi_{x\hat{y}}(i)=\left[x(i),\cdots,x(i-N_x+1),\hat{y}(i-1),\cdots,\hat{y}(i-N_d)\right]^{\mathrm{T}} \tag{25}$$

$$\Theta_{x\hat{y}}(i)=\left[\hat{a}_0(i),\cdots,\hat{a}_{N_x-1}(i),\hat{b}_1(i),\cdots,\hat{b}_{N_d}(i)\right]^{\mathrm{T}} \tag{26}$$

and unbiased estimates of the IIR filter parameters Θ_{xd} can be obtained without fitting a noise model. The modifications which are required to convert a standard RLS routine to operate as a SOLS algorithm are computationally simple and easy to implement. Convergence analysis by Moore [27] demonstrates that the SOLS procedure achieves global convergence in the presence of persistently excited input signals provided that $\left[B^{-1}\left(z^{-1}\right)-1/2\right]$ is strictly positive real [28].

III. Nonlinear Noise Cancellation Filters

If the noise distortion is only mildly nonlinear or the operating point moves slowly, an adaptive linear design might be sufficient [29]. But in general nonlinear distortion will necessitate the use of nonlinear noise cancellation filters.

III.1. The NARMAX Model

Consider the general single input, single output discrete time system

$$\hat{y}(i) = F(x(i)) \tag{27}$$

which is perturbed by additive noise $s(i-l)$ to yield the measured output $d(i-l)$

$$d(i-l) = F(x(i)) + s(i-l). \tag{28}$$

This can be expanded to yield

$$d(i-l) = dc + \hat{f}(\Phi_x(i), \Phi_d(i-l-1), \Phi_s(i-l-1)) + s(i-l) \tag{29}$$

where dc is a real valued constant and $\hat{f}(\cdot)$ is some nonlinear function. Models with this general structure are known as NARMAX models and it can be shown [7][30] that a wide class of nonlinear systems can be represented by models of this form. Since $\hat{y}(i)$ is a function of both input and output terms the NARMAX model can be viewed as a nonlinear IIR filter. The Hammerstein, Wiener, bilinear and Volterra models can all be shown to be special cases of the NARMAX model [30]. It is important to note that even though the noise is assumed to be additive in the noise cancellation problem this will induce multiplicative and cross product noise terms in the NARMAX model. This is indicated by the presence $\Phi_s(i-l-1)$ within $\hat{f}(\cdot)$ in Eqn. (29) and arises as a consequence of using an IIR design which employs past inputs and outputs in the regression.

III.2. Nonlinear FIR Filters

If the regression is restricted to past input terms the model of Eqn. (29) reduces to the Volterra functional expansion [31][32][33].

$$d(i-l) = dc + \sum_{n=1}^{K} \left\{ \sum_{k_n=0}^{N_x-1} \cdots \sum_{k_1=0}^{N_x-1} \left[h_n^{(k_1,\cdots,k_n)} \prod_{j=1}^{n} x\left(i-k_j\right) \Big] \Big] \right\} + s(i-l) \tag{30}$$

where $h_n^{(k_1,\cdots,k_n)}$ is called the nth order kernel and K is a sufficiently large positive integer. The Volterra series representation can be viewed as a special case of the NARMAX model where Eqn. (29) reduces to

$$d(i-l) = dc + \hat{f}\left(\Phi_x^{\mathrm{T}}(i)\right) + s(i-l) \tag{31}$$

This is just a nonlinear FIR filter. Expanding Eqn. (31) yields

$$d(i-l) = \Phi_x^{\prime\mathrm{T}}(i)\Theta_x^{\prime} + s(i-l) \tag{32}$$

where

$$\Phi_x^{\prime}(i) = \left[1, x(i), \cdots, x(i-N_x+1), x^2(i), \cdots, x(i)x(i-1), \cdots\right]^{\mathrm{T}} \tag{33}$$

and

$$\Theta_x^{\prime} = \left[dc, h_1^{(0)}, \cdots, h_1^{(N_x-1)}, h_2^{(0,0)}, \cdots, h_2^{(0,1)}, \cdots\right]^{\mathrm{T}}. \tag{34}$$

With this formulation the set of filter parameters can be updated using either the LMS or RLS algorithms [9] from linear estimation but with obvious redefinitions of the regression vector and parameter set. The advantage of the Volterra model and hence nonlinear FIR filter based designs for noise cancellation is that additive noise on the system translates into purely additive noise on the model. The complexity of induced multiplicative noise terms in Eqn. (29) is therefore avoided. But this has to be offset by the curse of dimensionality associated with nonlinear FIR designs which was discussed in the introduction. The explosion in the number of filter parameters which occurs for even simple nonlinear systems is a severe disadvantage of this class of filters.

III.3. Nonlinear NARMAX IIR Filters

General nonlinear IIR filter designs offer the advantage of small parameter sets at the expense of induced multiplicative noise terms. Both these properties arise as a direct consequence of exploiting the information contained in both past inputs and outputs. Fortunately by extending the SOLS concept to the nonlinear case the multiplicative noise terms can be avoided while retaining the concise IIR system representation.

To illustrate the concepts in the simplest way, consider a polynomial expansion of $\hat{f}(\cdot)$ in Eqn. (29) to yield

$$d(i-l)=dc+G_{xd}(\Phi'_x(i),\Phi'_d(i-l-1))+G_{xds}(\Phi'_x(i),\Phi'_d(i-l-1),\Phi'_s(i-l-1))+G_s(\Phi'_s(i-l-1))+s(i-l) \quad (35)$$

where

$$\Phi'_d(i-l-1)=\left[d(i-l-1),\cdots,d(i-l-N_d),d^2(i-l-1),\cdots,d(i-l-1)d(i-l-2),\cdots\right]^{\mathrm{T}} \quad (36)$$

$$\Phi'_s(i-l-1)=\left[s(i-l-1),\cdots,s(i-l-N_d),s^2(i-l-1),\cdots,s(i-l-1)s(i-l-2),\cdots\right]^{\mathrm{T}} \quad (37)$$

$G_{xd}(\cdot)$ is a function of $\Phi'_x(i)$ and $\Phi'_d(i-l-1)$, $G_{xds}(\cdot)$ represents all the cross product terms involving $\Phi'_s(i-l-1)$ and $G_s(\cdot)$ is a function of $\Phi'_s(i-l-1)$ only. $G_{xd}(\cdot)$, $G_{xds}(\cdot)$ and $G_s(\cdot)$ can, respectively, be viewed as operations of taking the linear combinations of their arguments. Eqn. (35) can therefore be expressed in matrix form as

$$d(i-l)=dc+\Phi'^{\mathrm{T}}_{xd}(i)\Theta'_{xd}+\Phi'^{\mathrm{T}}_{xds}(i)\Theta'_{xds}+\Phi'^{\mathrm{T}}_{s}(i)\Theta'_{s}+s(i-l) \quad (38)$$

where

$$\left.\begin{aligned}&G_{xd}(\Phi'_x(i),\Phi'_d(i-l-1))=\Phi'^{\mathrm{T}}_{xd}(i)\Theta'_{xd}\\&G_{xds}(\Phi'_x(i),\Phi'_d(i-l-1),\Phi'_s(i-l-1))=\Phi'^{\mathrm{T}}_{xds}(i)\Theta'_{xds}\\&G_s(\Phi'_s(i-l-1))=\Phi'^{\mathrm{T}}_{s}(i)\Theta'_{s}\end{aligned}\right\} \quad (39)$$

and Θ'_{xd}, Θ'_{xds} and Θ'_s are the parameter vectors corresponding to $\Phi'_{xd}(i)$, $\Phi'_{xds}(i)$ and $\Phi'_s(i)$ respectively.

For example consider a simple nonlinear quadratic system based on the notation in Fig. 2.

$$\left.\begin{aligned}&\hat{y}(i)=ax^2(i)+bx(i)\hat{y}(i-1)+c\hat{y}^2(i-1)\\&d(i-l)=\hat{y}(i)+s(i-l)\end{aligned}\right\}. \quad (40)$$

By eliminating $\hat{y}(i)$ in Eqn. (40), $d(i-l)$ can be expressed as

$$d(i-l)=ax^2(i)+bx(i)d(i-l-1)+cd^2(i-l-1)-bx(i)s(i-l-1)-2cd(i-l-1)s(i-l-1)+cs^2(i-l-1)+s(i-l) \quad (41)$$

Inspection of Eqn. (41) shows that higher order and cross product noise terms have been induced as a direct consequence of the presence of feedback terms involving $\hat{y}(i)$ in Eqn. (40). Under such conditions, biased estimates of the parameter set $\{a,b,c\}$ will result if a standard RLS procedure is applied. A comparison with Eqn. (35) shows that the following definitions hold.

$$\left.\begin{aligned}
&dc=0\\
&G_{xd}(\Phi'_x(i),\Phi'_d(i-l-1))=ax^2(i)+bx(i)d(i-l-1)+cd^2(i-l-1)\\
&G_{xds}(\Phi'_x(i),\Phi'_d(i-l-1),\Phi'_s(i-l-1))=-bx(i)s(i-l-1)-2cd(i-l-1)s(i-l-1)\\
&G_s(\Phi'_s(i-l-1))=cs^2(i-l-1)
\end{aligned}\right\} \quad (42)$$

Following the same argument as in Section II.2 if the output $n(i-l)$ in Fig. 2 could be monitored, Eqn. (38) could be expressed as

$$d(i-l)=dc+\hat{f}(\Phi_x(i),\Phi_n(i-l-1))+s(i-l) \quad (43)$$

so that all the cross product noise terms are avoided. Although $n(i-l)$ cannot be observed it can be replaced by the estimate calculated by

$$\begin{aligned}
\hat{y}(i)=dc&+\theta_x^{(0)}(i)x(i)+\cdots+\theta_x^{(N_x-1)}(i)x(i-N_x+1)+\theta_x^{(0,0)}(i)x^2(i)+\cdots+\theta_x^{(0,1)}(i)x(i)x(i-1)\\
&+\theta_{\hat{y}}^{(1)}(i)\hat{y}(i-1)+\cdots+\theta_{\hat{y}}^{(N_d)}(i)\hat{y}(i-N_d)+\theta_{\hat{y}}^{(1,1)}(i)\hat{y}^2(i-1)+\cdots+\theta_{\hat{y}}^{(1,2)}(i)\hat{y}(i-1)\hat{y}(i-2)+\cdots
\end{aligned} \quad (44)$$

Eqn. (44) can be written in matrix form to yield

$$\hat{y}(i)=\Phi'^{\mathrm{T}}_{x\hat{y}}(i)\Theta'_{x\hat{y}}(i) \quad (45)$$

where

$$\Phi'_{x\hat{y}}(i)=\left[1,x(i),\cdots,x(i-N_x+1),x^2(i),\cdots,x(i)x(i-1),\cdots,\hat{y}(i-1),\cdots,\hat{y}(i-N_d),\hat{y}^2(i-1),\cdots,\hat{y}(i-1)\hat{y}(i-2),\cdots\right]^{\mathrm{T}} \quad (46)$$

and

$$\Theta'_{x\hat{y}}(i)=\left[dc,\theta_x^{(0)},\cdots,\theta_x^{(N_x-1)},\theta_x^{(0,0)},\cdots,\theta_x^{(0,1)},\cdots,\theta_{\hat{y}}^{(1)},\cdots,\theta_{\hat{y}}^{(N_d)},\theta_{\hat{y}}^{(1,1)},\cdots,\theta_{\hat{y}}^{(1,2)},\cdots\right]^{\mathrm{T}}. \quad (47)$$

Eqn. (43) can then be modified to

$$d(i-l)=\Phi_{x\hat{y}}^{\prime\mathrm{T}}(i)\Theta_{x\hat{y}}^{\prime}(i)+s(i-l) \tag{48}$$

and the SOLS algorithm can be applied using an obvious extension to the notation for the linear case.

The SOLS solution matches the noise cancellation problem much better than either ELS or IV. ELS would operate on Eqn. (38) directly but now unbiased estimates of $\Theta_{xd}^{\prime}$ could only be obtained at the expense of using the full parameter vector $\Theta_{x\hat{y}}^{\prime}(i)$. The IV approach can be used if the model is linear but since this will generally produce biased estimates when the model is nonlinear [25] IV cannot be applied in the present case. Although all the simulation results [9][26] based on the SOLS algorithm have exhibited good convergence properties a theoretical convergence analysis is complex because the positive real condition which must hold in the linear case cannot easily be generalized for nonlinear models. Some of the concepts involved are discussed in [34] in the context of recursive prediction error estimation algorithms.

The analysis above was based on a polynomial expansion of $\hat{f}(\cdot)$ in Eqn. (29) to simplify the expansion. But the NARMAX model Eqn. (29) was derived for the case where $\hat{f}(\cdot)$ can take alternative nonlinear forms. While a polynomial expansion may be sufficient for many applications a much richer class of dynamic effects can be considered if $\hat{f}(\cdot)$ is expanded as a neural network [8].

IV. Radial Basis Function Filter Designs

Noise cancellation for a rich class of nonlinear systems can be studied by investigating the possibility of extending the SOLS idea to a neural network noise cancellation architecture.

The multi-layered perception (MLP) [35] and associated back-propagation training algorithm [36][37] have a very slow convergence rate, often stall at local minima and require the user to adjust the momentum rate, learning rate and number of hidden layers all of which can be very problem dependent [38][39]. This class of algorithms therefore does not appear to be well suited to adaptive applications and in the present study RBF networks will be studied.

RBF networks [40] consist of just two layers and provide an alternative to the MLP. A RBF network is illustrated in Fig. 3. There is only one hidden layer and this consists of an array of RBF nodes. Each node contains a parameter vector called a center. The node calculates the Euclidean distance between the center and the network input vector $\mathbf{v}(i)$ and the result is passed through a nonlinear function. The output layer is just a set of linear combiners.

The response of the RBF network can be expressed as

$$\hat{y}(i)=\sum_{j=1}^{N_h}\theta_j(i)g\left(\left\|\mathbf{v}(i)-\mathbf{c}_j(i)\right\|,\beta_j(i)\right) \tag{49}$$

where $\left\{\theta_j(i):1\le j\le N_h\right\}$ are the weights of the linear combiners, $\|\cdot\|$ denotes the Euclidean norm, $\left\{\mathbf{c}_j(i):1\le j\le N_h\right\}$ are the RBF centers, $\left\{\beta_j(i):1\le j\le N_h\right\}$ are positive scalars called widths, N_h is the number of nodes in the hidden layer and $g:\mathbf{R}_+\times\mathbf{R}\rightarrow\mathbf{R}$ is the radial basis function. The weight vector $\Theta(i)$ is defined as

$$\Theta(i)=\left[\theta_1(i),\cdots,\theta_{N_h}(i)\right]^{\mathrm{T}} \tag{50}$$

In the present study an m-input, single output RBF network will be sufficient and hence only one network output has been shown in Fig. 3 and Eqn. (49). Some commonly employed radial basis functions are listed below.

- thin-plate-spline function:

$$g(x)=x^2\ln x \tag{51}$$

- Gaussian function:

$$g(x,\beta)=e^{-\frac{x^2}{\beta^2}} \tag{52}$$

- multi-quadric function:

$$g(x,\beta)=\sqrt{x^2+\beta^2} \tag{53}$$

- inverse multi-quadric function:

$$g(x,\beta)=\frac{1}{\sqrt{x^2+\beta^2}} \tag{54}$$

where β is the width of the radial basis function. The width can either be fixed at some preset value or learnt during training. But in the present study the thin-plate-spline function which does not have a width parameter will be used. A detail discussion of the various effects of using different radial basis functions can be found in [41].

The radial basis functions form local patches whose positions are determined by the centers $\{\mathbf{c}_j(i)\ :\ 1\leq j\leq N_h\}$ and which are fitted together according to the weights $\{\theta_j(i)\ :\ 1\leq j\leq N_h\}$. By appropriately adjusting the centers and the weights of the network a wide class of highly nonlinear functions can be accurately approximated by the prescribed processing structure. Since it can be shown [42][43] that any continuous function can be uniformly approximated to an arbitrary accuracy by a RBF network this architecture should provide the basis for a new class of noise cancellation filters. This could be realized by replacing the filter $F(\cdot)$ in Fig. 1 with a RBF network and developing an appropriate class of adaptive training algorithms to update the weights and centers. Both FIR and IIR designs can be developed.

IV.1. FIR RBF Network Filters

The realization of FIR RBF filters is straightforward. The filter $F(\cdot)$ in Fig. 1 is replaced by a RBF network with output $\hat{y}(i)$ and with all the network nodes defined as lagged functions of $x(i)$. The FIR network can therefore be described as

$$\hat{y}(i)=\sum_{j=1}^{N_h}\theta_j(i)g\left(\left\|\Phi_x(i)-\mathbf{c}_j(i)\right\|\right) \tag{55}$$

and this can be considered as a simple extension of the Volterra FIR filter realization discussed in Section II. The main difference between the Volterra and the RBF FIR designs is that in the former case polynomic functions of input lags have to be explicitly computed to form the regression vector in Eqn. (33) whereas in the latter case only the lagged linear terms are required as input nodes to the network since functions of these are generated by the RBF neurons. But the problem of approximating a complex nonlinear expression in terms of an expansion of lagged inputs only remains and this will tend to produce large dimensional parameter sets or network weight vectors which will be disadvantageous to fast adaptive training.

IV.2. IIR Recurrent RBF Network Filters

The realization of IIR RBF filters follows the concepts introduced for the NARMAX IIR designs. The RBF network can be interpreted as an alternative expansion of $\hat{f}(\cdot)$ in the NARMAX model of Eqn. (29) so that $F(\cdot)$ in Fig. 1 can be approximated by

$$\hat{y}(i) = \sum_{j=1}^{N_h} \theta_j(i) g\left(\left\| \mathbf{\Phi}_{x\hat{y}}(i) - \mathbf{c}_j(i) \right\|\right) \tag{56}$$

The advantages of the nonlinear IIR design which were discussed in Section III carry over for the RBF based design. But just as in the NARMAX IIR design the advantage of a small parameter or network weight vector is achieved by using lagged inputs and outputs to define the RBF network input nodes. The problem associated with this is that the lagged outputs are highly correlated with the additive noise and straightforward network training based on this simple architecture would yield a biased or incorrect filter realization. Exploitation of the reduced size of the RBF networks and hence the potential for fast training and good adaptive tracking will therefore only be possible if the SOLS ideas can be extended to the RBF network architecture. This can be achieved by using a form of recurrent network where the noise corrupted system output signals are replaced by the network predicted outputs. This in effect means that the network predictions are fed back and assigned as network input nodes to produce what has become known as a recurrent network architecture. The IIR RBF noise cancellation filter realization, $F(\cdot)$ in Fig. 1, is therefore based on the map

$$\hat{f}: \mathbf{R}^{N_x+N_d} \times \mathbf{R}^{(N_x+N_d)\times N_h} \times \mathbf{R}^{N_h} \to \mathbf{R} \tag{57}$$

Before the recurrent network can be fully defined we need to introduce the network training algorithm.

IV.3. Recurrent Network Training

The strategy for updating the network parameters involves a combined form of supervised and unsupervised learning [13]. At each iteration a new recurrent hybrid algorithm is used which consists of updating the centers using a k-means clustering algorithm [44] and then updating the weights using a Givens SOLS algorithm [17]. These two sub-algorithms will be described sequentially below before combining them to form the hybrid training algorithm.

The k-means clustering algorithm is a procedure for classifying clustered data and locating the centers of gravity in a multi-dimensional space [44]. Given initial centers $\{\mathbf{c}_j(0) : 1 \le j \le N_h\}$ and an initial learning rate $\alpha(0)$ the k-means clustering algorithm operates, for each processing unit in the hidden layer, to compute the Euclidean distances $\{\rho_j(i) : 1 \le j \le N_h\}$ between the current input vector $\mathbf{v}(i)$ and the previous centers $\{\mathbf{c}_j(i-1) : 1 \le j \le N_h\}$,

$$\rho_j(i) = \left\| \mathbf{v}(i) - \mathbf{c}_j(i-1) \right\| \quad 1 \le j \le N_h \tag{58}$$

where $\mathbf{v}(i)$ is defined in terms of the current, lagged inputs and previous network predictions as

$$\mathbf{v}(i) = \left[x(i), \cdots, x(i - N_x + 1), \hat{y}(i-1), \cdots, \hat{y}(i - N_d) \right]^{\mathrm{T}} \tag{59}$$

with $m = N_x + N_d$. The key feature of the k-means clustering algorithm is that at each time step, only one of the centers will be altered. The updating of a center is based on the criterion of how far the current input vector is away from the last updated centers measured by the Euclidean distance. If $k = \text{argument}\left\{ \min\limits_{1 \le j \le N_h} \rho_j(i) \right\}$ then

$$\mathbf{c}_k(i) = \mathbf{c}_k(i-1) + \alpha(i)\left[\mathbf{v}(i) - \mathbf{c}_k(i-1) \right] \tag{60}$$

and the rest of the centers will be remain unchanged

$$\mathbf{c}_j(i) = \mathbf{c}_j(i-1) \quad \forall j \ne k. \tag{61}$$

The real valued scalar $\alpha(i) \in (0,1]$, called the learning rate, is given by

$$\alpha(i) = \frac{\alpha(i-1)}{\sqrt{1+\text{int}\left(\frac{i}{N_h}\right)}} \tag{62}$$

where $\text{int}(x)$ denotes the integral part of x. Other choices of updating rule for the learning rate are possible providing $\alpha(i)$ monotonically decreases to zero. Experience suggests that the choice of initial learning rate $\alpha(0)$ is not critical and good results are generally obtained with $\alpha(0)$ set close to unity. Once the centers have been updated the outputs of the hidden layer can be calculated using

$$\phi_j(i) = g\left(\left\|\mathbf{v}(i) - \mathbf{c}_j(i)\right\|\right) \quad 1 \le j \le N_h \tag{63}$$

to form the input vector to the output layer

$$\Phi(i) = \left[\phi_1(i), \cdots, \phi_{N_h}(i)\right]^{\mathrm{T}}. \tag{64}$$

The objective is to cluster the data into separate regions represented by the centers in a recursive manner. The initial centers should be selected from a subset of the input data within the estimate range. Experience indicates that initial centers can usually be randomly assigned in the vicinity of the domain of the input space.

Given a set of fixed centers determined by the k-means clustering algorithm, the weight vector $\Theta(i)$ defined in Eqn. (50) can be recursively updated using a Givens algorithm. To simplify the exposition it will be assumed initially that at time step i the input vector sequence $\Phi(1), \cdots, \Phi(i)$ and noise corrupted signal sequence $d(-l+1), \cdots, d(i-l)$, where $d(k) = 0$ for all $k < 0$, are available.

The formulation of the Givens RLS algorithm is based on solving the normal equation

$$\mathbf{X}^{\mathrm{T}}(i)\Lambda(i)\mathbf{X}(i)\hat{\Theta}(i) = \mathbf{X}^{\mathrm{T}}(i)\Lambda(i)\mathbf{d}(i-l) \tag{65}$$

using a Givens orthogonal transformation where

$$\mathbf{X}(i) = \left[\Phi(1), \cdots, \Phi(i)\right]^{\mathrm{T}} \tag{66}$$

$$\mathbf{d}(i-l)=\left[d(-l+1),\cdots,d(i-l)\right]^{\mathrm{T}} \tag{67}$$

$$\Lambda(i)=\begin{bmatrix}\lambda(i)\Lambda(i-1) & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix} \text{ with } \Lambda(1)=1. \tag{68}$$

and $\hat{\Theta}(i)$ is a least squares solution of $\Theta(i)$ at time step i. At each time step i, the forgetting factor $\lambda(i)$ is usually taken to be a value close to unity. For time varying systems a forgetting factor is usually employed in order to accommodate a continual tracking capability. By writing

$$\Lambda^{1/2}(i)\mathbf{X}(i)=\mathbf{Q}(i)\mathbf{S}(i) \tag{69}$$

where $\mathbf{Q}(i)$ is an $i\times N_h$ matrix with orthogonal columns and $\mathbf{S}(i)$ is an $N_h\times N_h$ unit upper triangular matrix so that

$$\mathbf{Q}^{\mathrm{T}}(i)\mathbf{Q}(i)=\mathbf{D}(i) \tag{70}$$

where $\mathbf{D}(i)$ is an $N_h\times N_h$ diagonal matrix the normal equation in Eqn. (65) can be rewritten as

$$\mathbf{D}^{1/2}(i)\mathbf{S}(i)\hat{\Theta}(i)=\mathbf{D}^{1/2}(i)\mathbf{q}(i) \tag{71}$$

where

$$\mathbf{q}(i)=\mathbf{D}^{-1}(i)\mathbf{Q}^{\mathrm{T}}(i)\Lambda^{1/2}(i)\mathbf{d}(i-l). \tag{72}$$

Solving Eqn. (71) is equivalent to solving Eqn. (65). The Givens RLS algorithm consists of the following computational steps:

- Prior to the starting of the algorithm all the variables are initialized as follows

$$\left.\begin{aligned}\mathbf{q}(0)&=\mathbf{0}\\ \mathbf{S}(0)&=\mathbf{I}\\ \mathbf{D}^{1/2}(0)&=\delta\mathbf{I}\end{aligned}\right\} \tag{73}$$

  where $\mathbf{I}$ is an $N_h\times N_h$ unit matrix and δ is a small positive number.

- At each time step, $\lambda(i)$ is updated according to

$$\lambda(i)=\lambda_0\lambda(i-1)+1-\lambda_0 \tag{74}$$

  $\lambda(0)$ and λ_0 are chosen to be just less than unity [12].

- Perform a Givens transformation

$$\begin{bmatrix} \sqrt{\lambda(i)}\mathbf{D}^{1/2}(i-1)\mathbf{S}(i-1) & \sqrt{\lambda(i)}\mathbf{D}^{1/2}(i-1)\mathbf{q}(i-1) \\ \Phi^{\mathrm{T}}(i) & d(i) \end{bmatrix} \xrightarrow{\text{Givens Transform}} \begin{bmatrix} \mathbf{D}^{1/2}(i)\mathbf{S}(i) & \mathbf{D}^{1/2}(i)\mathbf{q}(i) \\ \mathbf{0} & d'(i) \end{bmatrix} \quad (75)$$

where $d'(i)$ is a 'don't care' variable.

- Solve the upper triangular system in Eqn. (71) by back-substituting the vector-matrix equation formed by the submatrices $\mathbf{D}^{1/2}(i)\mathbf{S}(i)$ and $\mathbf{D}^{1/2}(i)\mathbf{q}(i)$ in Eqn. (75) to obtain $\hat{\Theta}(i)$.

The Givens transformation method applied to the least squares problem has been extensively studied by Gentlemen [17] who showed that this algorithm has numerical advantages over aforementioned algorithms based on a direct solution of the normal equations.

Recent studies [43] suggest that, the rate of convergence in network learning can be enhanced by modifying the network to allow direct link terms to contribute to the weighted sum at the network output node. In addition it is advantageous to allow a direct adjustable d.c. value link.

These modifications can easily be achieved by augmenting the vector $\Phi(i)$ to yield

$$\bar{\Phi}(i) = \begin{bmatrix} \Phi(i) \\ \mathrm{v}(i) \\ 1 \end{bmatrix}. \quad (76)$$

where $\mathrm{v}(i)$ is defined in Eqn. (59) and the unity entry provides the d.c. link. For notional convenience $\bar{\Phi}(i)$ will be relabelled as $\Phi(i)$ so that now

$$\Phi(i) = \left[\phi_1(i), \cdots, \phi_{N_h}(i), \mathrm{v}^{\mathrm{T}}(i), 1\right]^{\mathrm{T}} = \left[\phi_1(i), \cdots, \phi_{N_h}(i), v_1(i), \cdots, v_m(i), 1\right]^{\mathrm{T}}. \quad (77)$$

The network connection weight vector must also be redefined as

$$\Theta(i) = \left[\theta_1(i), \cdots, \theta_{N_h+m+1}(i)\right]^{\mathrm{T}}. \quad (78)$$

To ensure that the estimated network weights in the IIR RBF noise canceller are unbiased the Givens routine must operate in a manner analogous to the SOLS algorithm described in Section

III for the NARMAX IIR filter. This can be achieved by using the definition of $\mathbf{v}(i)$ used in the clustering routine to give, for an m-input, single output N_h-center network described by

$$\hat{y}(i) = \sum_{j=1}^{N_h} \theta_j(i) g\left(\left\|\mathbf{v}(i) - \mathbf{c}_j(i)\right\|\right) + \sum_{k=1}^{m} \theta_{N_h+k}(i) v_k(i) + \theta_{N_h+m+1}(i) \qquad (79)$$

where $g:\mathbf{R}_+ \to \mathbf{R}$ is defined in Eqn. (51). When Givens routine is applied to obtain an estimate of $\Theta(i)$ the predicted output $\hat{y}(i)$ at time step i can be computed by

$$\hat{y}(i) = \Phi^{\mathrm{T}}(i)\hat{\Theta}(i) \qquad (80)$$

where $\hat{\Theta}(i)$ is redefined as the least squares estimate of $\Theta(i)$ due to the redefinition of $\Theta(i)$ in Eqn. (78). Notice that now the RBF network inputs consist of lagged samples of the predicted network outputs $\hat{y}(i)$. The network predictions are therefore fed back to form part of the network input vector thus giving the required network architecture. A schematic diagram of recurrent RBF network is illustrated on Fig. 4.

IV.4. Summary of the RBF Recurrent Network Filters

The RBF noise canceller therefore consists of updating first the centers and then the network weights at each iteration. This combines to form a hybrid recurrent network training algorithm which can be summarized by the following steps:

- Prior to the starting the system initialize all components of the input vector $\mathbf{v}(0)$ to be zero and the parameters for Givens routine as specified in Eqn. (73). Randomly assign a value to each component for all the centers in the vicinity of the input space. Initialize the initial learning rate $\alpha(0)$ and the initial forgetting factor $\lambda(0)$.
- At time step i, collect samples of $x(i)$ and $d(i-l)$ where $d(k) = 0 \; \forall k < 0$ by definition.
- The noise cancellation system takes the current, lagged inputs and lagged outputs of the network to form the $m(= N_x + N_d)$-dimensional input vector $\mathbf{v}(i)$ defined in Eqn. (59).
- Update the learning rate $\alpha(i)$ using Eqn (62). The centers of the network $\{\mathbf{c}_j(i) \; : \; 1 \le j \le N_h\}$ are updated by the k-means clustering algorithm using Eqn. (60) and Eqn. (61).

- Calculate the outputs $\left\{\phi_j(i)\ :\ 1 \le j \le N_h\right\}$ of the hidden layer based on the updated centers and form the input vector to the output layer $\Phi(i)$ defined in Eqn. (77).
- Update the forgetting factor $\lambda(i)$ using Eqn. (74)
- Update the weight vector $\hat{\Theta}(i)$ using the Givens algorithm described in Eqn. (75).
- Compute the predicted output $\hat{y}(i)$ based on the updated centers and weight vector by using Eqn. (80).
- Generate the *i*th sample of the recovered signal $\varepsilon(i)$ using Eqn. (3).
- Store $\hat{y}(i)$ as a feedback signal for the next iteration and repeat the process from the second step above.

A schematic diagram of the noise cancellation system based on this algorithm is illustrated in Fig. 5.

V. Filter Validation

It can be shown [45] that if the canceller is operating correctly then the following correlation model validity tests should all be satisfied

$$\left.\begin{aligned}
\Psi_{x\varepsilon}(k) &= E\{x(i)\varepsilon(i+k)\} = 0 \\
\Psi_{x^{2'}\varepsilon}(k) &= E\left\{\left[x^2(i) - \left\langle x^2(i)\right\rangle\right]\varepsilon(i+k)\right\} = 0 \\
\Psi_{x^{2'}\varepsilon^2}(k) &= E\left\{\left[x^2(i) - \left\langle x^2(i)\right\rangle\right]\varepsilon^2(i+k)\right\} = 0
\end{aligned}\right\} \text{for all integer } k \tag{81}$$

where $\langle \cdot \rangle$ denotes the time average. On-line versions of these tests are also available [45]. In practice the model will be regarded as adequate if all the tests in Eqn. (81) fall within the 95% confidence bands at approximately $\pm 1.96/\sqrt{N}$ where *N* is the number of samples.

These tests are designed to monitor the performance of the noise cancellation filter irrespective of the specific filter realization that is used. The tests have been formulated to detect any predictable linear or nonlinear effects in $\varepsilon(i)$ since the existence of such effects would mean that the canceller was not matched to the system. The tests should for example detect the incorrect use of a linear filter to cancel a nonlinear system, an inadequate NARMAX filter or an incorrectly trained RBF network filter.

VI. Simulation Examples

Three simulated examples which represent different types of nonlinearities will be considered. For ease of comparison a sawtooth signal of unit magnitude, period 50 samples as shown in Fig. 6 and a uniformly distributed white noise sequence varying in the range $[-2,2]$ as shown in Fig. 7 will be used in all the simulation examples. A plot of the noise corrupted signal is shown in Fig. 8. In each case a total of 20,000 samples were used and the results obtained from the first 1/10 (i.e. 2,000) were discarded in order to ensure that transient conditions are ignored in accessing the quality of the algorithms. As a measure of noise suppression the noise reduction factor (*NR*) is defined in terms of loss in average noise power at the output of the noise canceller [46] as

$$NR = 10\log_{10}\left(\frac{E\{n^2(i)\}}{E\{[s(i-l)-\varepsilon(i)]^2\}}\right)(\text{dB}). \tag{82}$$

From this definition it is clear that the larger the value of *NR* the greater amount of noise energy which has been canceled.

Example 1: The reference noise $x(i)$ was generated using the Nonlinear AutoRegressive model with eXogenous inputs (NARX) described by

$$x(i)=0.25x(i-1)+0.1x(i-2)+0.5n(i-1)+0.1n(i-2)-0.2n(i-3)+0.1n^2(i-2)+0.08n(i-2)x(i-1) \tag{83}$$

Notice that there is one sample delay in obtaining $x(i)$ from $n(i)$. Therefore $l=1$. Optimal noise cancellation will be achieved if the noise cancellation filter $F(\cdot)$ is implemented as a nonlinear IIR filter described by

$$\hat{y}(i)=2x(i)-0.5x(i-1)-0.2x(i-2)-0.2\hat{y}(i-1)+0.4\hat{y}(i-2)-0.2y^2(i-1)-0.16x(i-1)\hat{y}(i-1) \tag{84}$$

In the present simulation $F(\cdot)$ was constructed using a 20-center RBF network with linear dynamic links. The network input vector was defined as

$$\mathrm{v}(i)=\left[x(i),x(i-1)x(i-2),\hat{y}(i-1),\hat{y}(i-2)\right]^{\mathrm{T}} \tag{85}$$

giving $N_x=3$ and $N_d=2$ and the parameters in the hybrid algorithm were initialized to as

$$\left.\begin{aligned}\delta&=0.00001\\ \lambda(0)&=0.95\\ \lambda_0&=0.99\\ \alpha(0)&=0.9\end{aligned}\right\} \tag{86}$$

All the centers were randomly initialized in the region $[-2,3]$ for the first three components and $[-3,3]$ for the rest. A high degree of noise suppression at 22.41 dB was achieved. Fig. 9 demonstrates that after about 500 adaptations of the learning sequence, the noise canceller is performing well. Compared with the results obtained using a nonlinear FIR Volterra filter and the exact optimal nonlinear IIR filter Eqn. (84) described in [9] the level of noise reduction achieved using the RBF network implementation lies between these two approaches and suffers only a 2.5 dB degradation from the optimal implementation. A trace of the recovered signal sequence $\{\varepsilon(i)\}$ is shown in Fig. 10. The correlation tests illustrated in Fig. 11 are all within the 95% confidence bands confirming that the recurrent RBF filter is an adequate noise cancellation filter for this system.

Example 2: In this example, the reference noise $x(i)$ was generated by compressing the noise $n(i)$ according to the A-law given by

$$|x(i)|=\begin{cases}\dfrac{A|n(i)|}{2(1+\ln A)} & \text{if } \ 0\le\dfrac{|n(i)|}{2}\le\dfrac{1}{A}\\[2ex] \dfrac{1+\ln\left(\dfrac{A|n(i)|}{2}\right)}{1+\ln A} & \text{if } \ \dfrac{1}{A}\le\dfrac{|n(i)|}{2}\le 1\end{cases} \tag{87}$$

where $A = 87.6$, followed by a 64-level linear mid-riser quantizer.

The noise cancellation filter $F(\cdot)$ was implemented using a 10-center RBF network with linear network links. Since at time step i $x(i)$ corresponds exactly to $n(i)$ the structure of the input vector can be defined by

$$\mathrm{v}(i) = x(i) \tag{88}$$

that is $N_x = 1$, $N_d = 0$ and $l = 0$. The initial centers were randomly chosen in the range $[-2,2]$ and the network was initialized as in example 1. Fig. 12 demonstrates that the algorithm is capable of tracking the nonlinear dynamics of the system after a learning sequence of approximately 300 samples and a 25.48 dB reduction in the level of noise can be achieved. A trace of the recovered signal sequence $\{\varepsilon(i)\}$ is shown in Fig. 13. Model validation tests in Fig. 14 confirm that the 10-center RBF network is a suitable model for the nonlinear system under consideration.

Example 3: Consider the nonlinear system

$$x(i) = \tanh n(i) \tag{89}$$

This is a single input-single output system with continuous nonlinearity so that the input vector to the network can be defined by

$$\mathrm{v}(i) = x(i) \tag{90}$$

giving $N_x = 1$, $N_d = 0$ and $l = 0$. The noise cancellation filter $F(\cdot)$ was implemented using a 20-center RBF network with linear network links. The initial centers were randomly selected to be in the range $[-1,1]$ and the network was initialized as in Example 1. Fig. 15 shows that after a learning sequence of about 150 iterations, the noise canceller is capable of following the dynamics described in Eqn. (89). In the steady state condition a 30.73 dB reduction in noise level can be achieved for the present system. A trace of the recovered signal sequence $\{\varepsilon(i)\}$ is shown in Fig. 16. Fig. 17 shows that all the three correlation functions are within the 95% confidence bands indicating that the 20-center RBF network is an adequate model for the underlying nonlinear dynamics.

VII. CONCLUSIONS

The design of nonlinear FIR and IIR noise cancellation filters has been studied. New filter realizations based on both the traditional and a new recurrent RBF network have been introduced. Network training based on a combined k-means clustering and Givens implementation of the suboptimal least squares algorithm was discussed and the inclusion of linear dynamic link connections was shown to enhance convergence. Model validation methods were shown to provide a measure of network performance and the examples demonstrate the potential of the designs for noise cancellation of several nonlinear systems.

REFERENCES

[1]. B. Widrow and M. E. Hoff, "Adaptive Switching Circuits", *IRE WESCON Convention Record*, Pt. 4, New York: IRE, pp. 96-104, 1960.

[2]. P. L. Feintuch, "An Adaptive Recursive LMS Filter", *Proc. IEEE*, Vol. 64, pp. 1622-1624, Nov. 1976.

[3]. R. L. Plackett, "Some Theorems in Least Squares", *Biometrika*, Vol. 37, pp. 149-157, 1950.

[4]. G. L. Sicuranza, A. Bucconi and P. Mitri, "Adaptive Echo Cancellation with Nonlinear Digital Filters", *Proc. IEEE International Conference on ASSP*, San Diego, CA, USA, Vol. 1, pp. 3.10/1-44, Mar., 1984.

[5]. M. J. Coker and D. N. Simkins, "A Nonlinear Adaptive Noise Canceller", *Proc. IEEE International Conference on ASSP*, Pt. II, Denver, CO, USA, pp. 470-473, Apr., 1980.

[6]. S. A. Billings, J. O. Gray and D. H. Owens, *Nonlinear System Design*, Peter Peregrinus Ltd., UK, 1984.

[7]. S. A. Billings and J. Leontarities, "Identification of Nonlinear Systems Using Parameter Estimation Techniques", *Institute of Electrical Engineers Conference*, pp. 183-190, 1981.

[8]. S. Chen and S. A. Billings, "Representation of Nonlinear Systems: The NARMAX Model", *Int. J. Control*, Vol. 49, No. 3, pp. 303-344, 1985.

[9]. S. A. Billings and M. El-Hitmy, "Adaptive Noise Cancellation for a Class of Nonlinear IIR Filters", *J. Sound Vib.*, Vol. 140, No. 3, 1990, pp. 457-473.

[10]. S. A. Billings and W. S. F. Voon, "Piecewise Linear Identification of Nonlinear Systems", Int. J. Syst. Sci., Vol. 46, pp. 215-235, 1987.

[11]. T. Söderström and P. Stoica, "On Some System Identification Techniques for Adaptive Filtering", *IEEE Trans. Cir. & Syst.*, Vol. 35, No. 4, pp. 457-461, 1988.

[12]. L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT Press, 1985.

[13]. S. Chen, S. A. Billings and P. M. Grant, "A Recursive Hybrid Algorithm for Nonlinear System Identification Using Radial Basis Function Networks", *Int. J. Control*, Vol. 55, No. 5, pp. 1051-1070, 1992.

[14]. V. J. Mathews, "Adaptive Polynomial Filters", *IEEE SP Mag.*, pp. 10-26, July, 1991.

[15]. P. C. Young, *Recursive Estimation and Time Series Analysis*, Springer Verlag, 1984.

[16]. S. Y. Kung, H. J. Whitehouse and T. Kailath, *VLSI and Modern Signal Processing*, Prentice Hall, 1985.

[17]. W. E. Gentleman, "Least Squares Computations by Givens Transformations without Square Roots", *J. Inst. Maths. Appls.*, Vol. 12, pp. 329-336, 1973.

[18]. A. J. Robins and P. E. Wellstead, "Recursive System Identification Using Fast Algorithms", *Int. J. Control*, Vol. 33, No. 3, pp. 455-480, 1981.

[19]. G. Caraynannis, D. Manolakis and N. Kalouptsidis, "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction", *IEEE Trans. ASSP*, Vol. 31, No. 12, pp. 1394-1402, 1983.

[20]. J. Cioffi and T. Kailath, "Fast Recursive-Least-Squares Transversal Filters for Adaptive Filtering", *IEEE Trans. ASSP*, Vol. 32, No. 4, pp. 304-338, 1984.

[21]. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[22]. E. Eweda and O. Macchi, "Convergence of the RLS and LMS Adaptive Filters", *IEEE Trans. Cir. & Syst.*, Vol. 7, pp. 799-803, 1987.

[23]. S. S. Haykin, *Adaptive Filter Theory*, 2nd Ed., Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[24]. B. Friedlander, "A Modified Prefilter for Some Recursive Parameter Estimation Algorithms", *IEEE Trans. Auto. Contr.*, Vol. 27, No. 1, pp. 232-235, 1982.

[25]. B. Friedlander, "Instrumental Variable Methods for ARMA Spectral Estimation", *IEEE Trans. ASSP*, Vol. 31, No. 2, pp. 404-415, 1983.

[26]. S. A. Billings and W. S. F. Voon, "Least Squares Parameter Estimation Algorithms for Non-Linear Systems", *Int. J. Syst Sci.*, Vol. 15, No. 6, pp. 601-615, 1984.

[27]. J. B. Moore, "Global Convergence of Output Error Recursions in Colored Noise", *IEEE Trans. Auto. Contr*, Vol. AC-27, No. 6, pp. 1189-1199, 1982.

[28]. S. Dasgupta and A. S. Bhagwat, "Conditions for Designing Strictly Positive Real Transfer Functions for Adaptive Output Identification", *IEEE Trans. Cir. & Syst.*, Vol. 34. No. 7, pp. 731-736, 1987.

[29]. B. Widrow, J. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong and R. C. Goodlin, "Adaptive Noise Canceling: Principles and Applications", *Proc. IEEE*, Vol. 63, No. 12, pp. 1692-1716, 1975.

[30]. J. Leontaritis and S. A. Billings, "Input Output Parametric Models for Nonlinear Systems, Part II: Deterministic Nonlinear Systems", *Int. J. Control*, Vol. 41, pp. 303-344, 1985.

[31]. S. A. Billings, "Identification of Nonlinear Systems: A Survey", *IEE Proc. Pt. D*, Vol. 127, No. 6, pp. 272-285, 1980.

[32]. V. Volterra, *Theory of Functionals*, Glasgow: Blackie, 1930.

[33]. M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley and Sons: New York, 1980.

[34]. S. Chen and S. A. Billings, "A Recursive Prediction Error Parameter Estimator for Nonlinear Models", *Int. J. Control*, Vol. 49, pp. 569-594, 1988.

[35]. F. Rosenblatt, *Principles of Neurodynamics*, New York: Spartan, 1962.

[36]. P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD Thesis, Harvard University.

[37]. S. A. Billings, H. B. Jamaluddin and S. Chen, "Properties of Neural Networks with Applications to Modelling Nonlinear Dynamical Systems", *Int. J. Control*, Vol. 55, No. 1, pp. 193-224, 1992.

[38]. S. A. Billings and S. Chen, "Neural Networks and System Identification", *Neural Networks for Control and Systems: Principles and Applications*, Ed. by K. Warwick, G. W. Irwin and K. J. Hunt, Peter Peregrinus Ltd., UK, pp.181-205, 1992.

[39]. K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. Neural Net.*, Vol. 1, No. 1, pp. 4-27, 1990.

[40]. D. S. Broomhead and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks", *Complex Syst.*, Vol. 2, pp. 321-355, 1988.

[41]. J. D. Powell, "Radial Basis Function Approximations to Polynomials", *Proc. 12th Biennial Numerical Analysis Conference*, Dundee, pp. 223-241, 1987.

[42]. K. Funahashi, "On The Approximate Realization of Continuous Mappings by Neural Networks", *Neural Net.*, Vol. 2, pp. 183-192, 1989.

[43]. T. Poggio and F. Girosi, "Networks for Approximation and Learning", *Proc. IEEE*, Vol. 78, No. 9, pp. 1481-1496, 1990.

[44]. Moody and C. Darken, "Fast-Learning in Networks of Locally-Tuned Processing Units", *Neural Comp.*, Vol. 1, No. 2, pp. 281-294, 1989.

[45]. S. A. Billings and F. A. Alturki, "Performance Monitoring in Nonlinear Adaptive Noise Cancellation", *J. Sound Vib.*, To Appear.

[46]. D. Graupe and A. J. Efron, "An Output-Whitening Approach to Adaptive Active Noise Cancellation", *IEEE Trans. Cir. & Syst.* Vol. 38, No. 1, pp. 1306-1313, 1991.

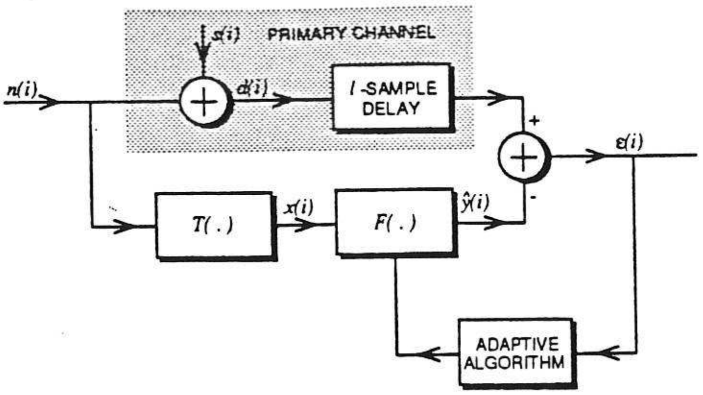


Fig. 1. Noise cancellation problem

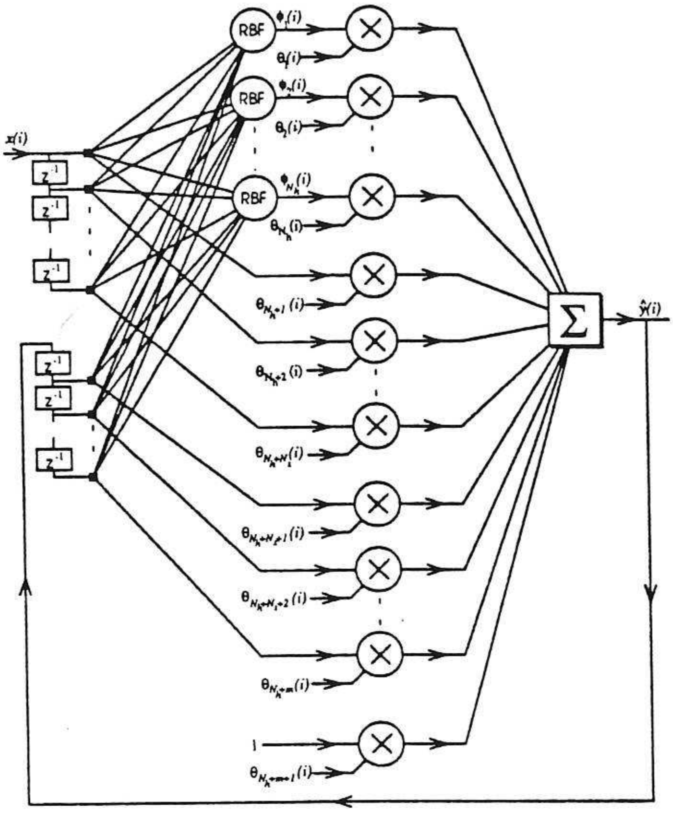


Fig. 4. Recurrent RBF network with linear links

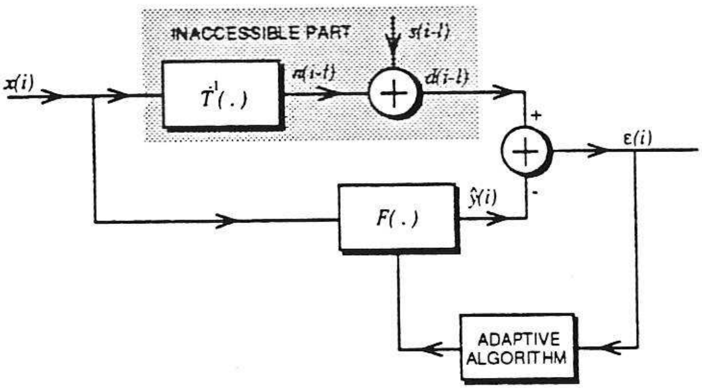


Fig. 2. System identification interpretation of noise cancellation problem

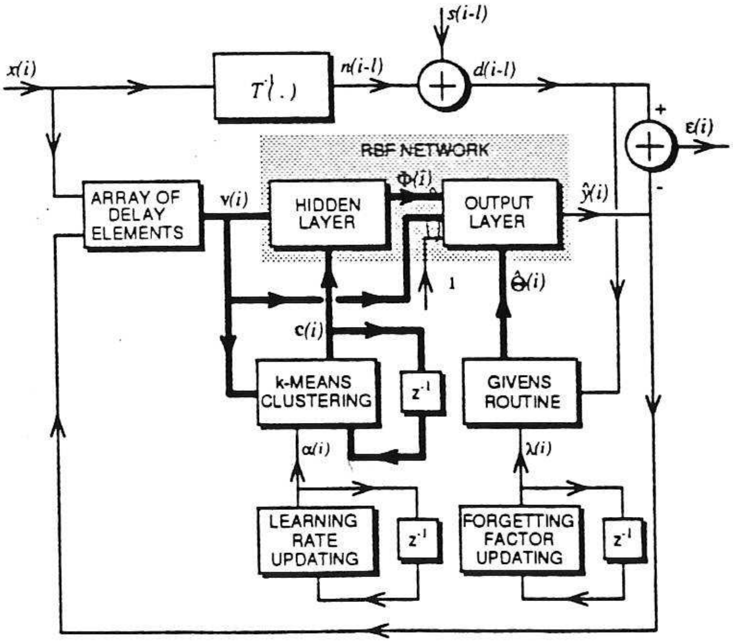


Fig. 5. Adaptive noise cancellation system

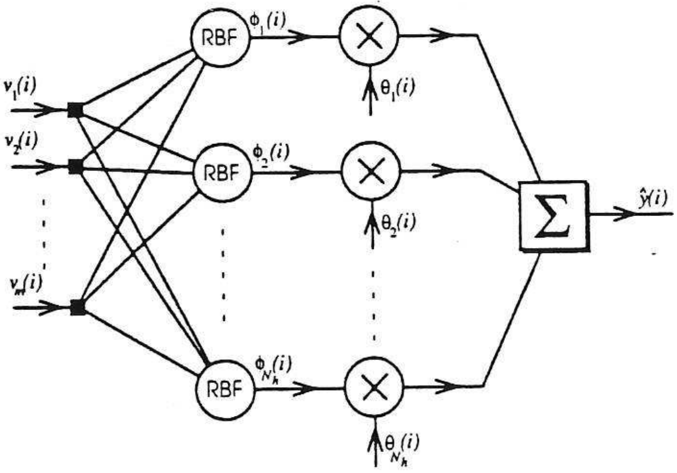


Fig. 3. Radial basis function network

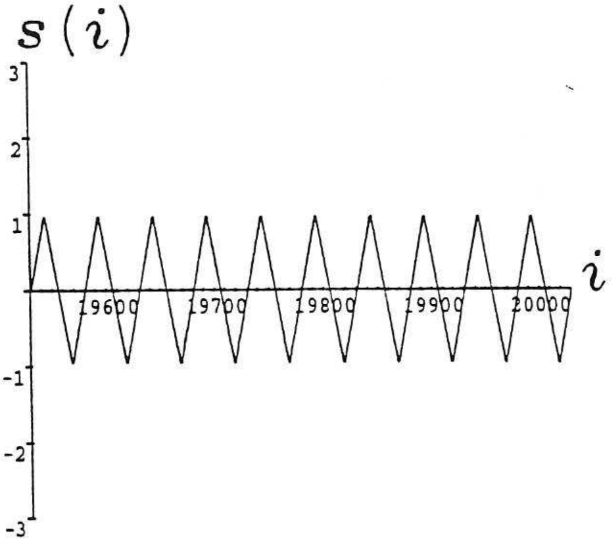


Fig. 6.

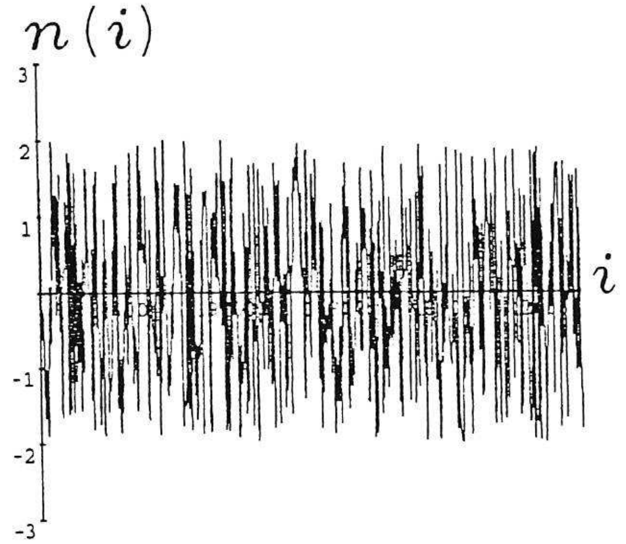


Fig. 7.

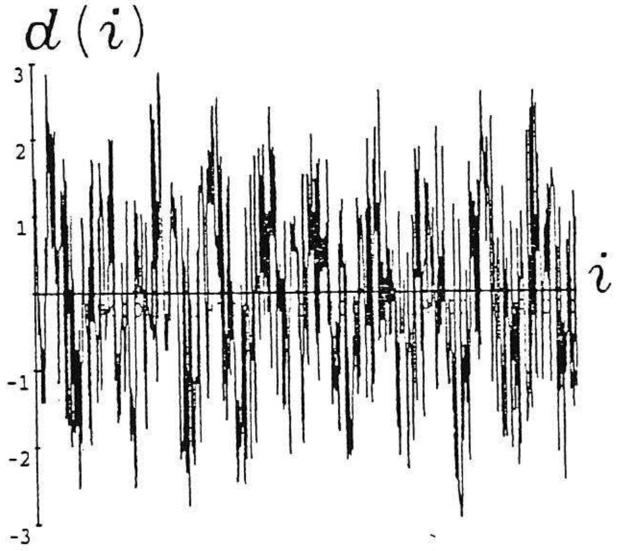


Fig. 8.

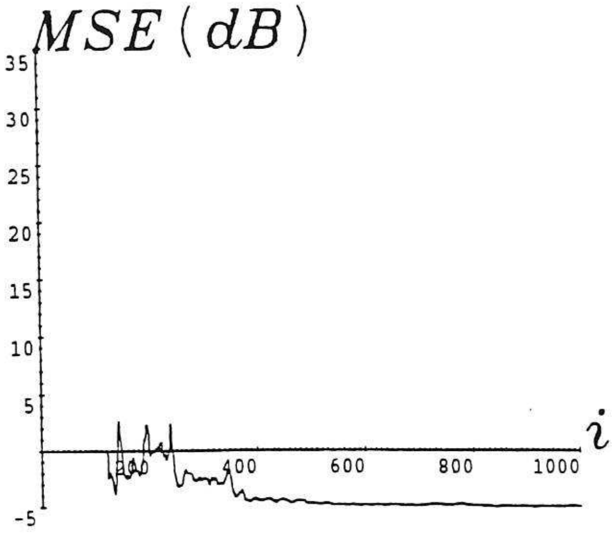


Fig. 9.

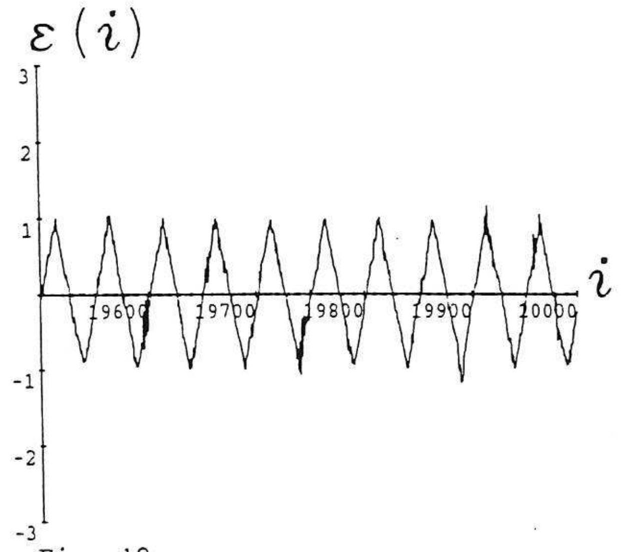
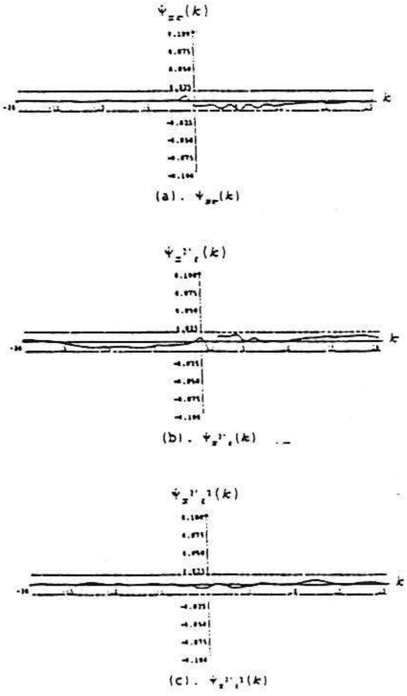


Fig. 10.



(a). $\Psi_{x\varepsilon}(k)$

(b). $\Psi_{x^{2}\varepsilon}(k)$

(c). $\Psi_{x^{2}\varepsilon^{2}}(k)$

Fig. 11.