Filtered-X Radial Basis Function Neural Networks for Active Noise Control

Bambang Riyanto[*)], Lazuardi Anggono[*)] & Kenko Uchida[**)]

[*)] Dept. Electrical Eng., Institut Teknologi Bandung,
Jalan Ganesha 10, Bandung, Indonesia
[**)] Dept. Electrical Eng., Waseda University, Tokyo, Japan
E-mail : briyanto@lskk.ee.itb.ac.id

Abstract. This paper presents active control of acoustic noise using radial basis function (RBF) networks and its digital signal processor (DSP) real-time implementation. The neural control system consists of two stages: first, identification (modeling) of secondary path of the active noise control using RBF networks and its learning algorithm, and secondly neural control of primary path based on neural model obtained in the first stage. A tapped delay line is introduced in front of controller neural, and another tapped delay line is inserted between controller neural networks and model neural networks. A new algorithm referred to as Filtered X-RBF is proposed to account for secondary path effects of the control system arising in active noise control. The resulting algorithm turns out to be the filtered-X version of the standard RBF learning algorithm. We address centralized and decentralized controller configurations and their DSP implementation is carried out. Effectiveness of the neural controller is demonstrated by applying the algorithm to active noise control within a 3 dimension enclosure to generate quiet zones around error microphones. Results of the real-time experiments show that 10-23 dB noise attenuation is produced with moderate transient response.

Keywords: *Active Noise Control; Adaptive Nonlinear Control; DSP; Radial Basis Function Networks.*

1 Introduction

Basically, there are two approaches in control of acoustic noise: active and passive methods[1,2,4,5,12]. Active noise control (ANC), also referred to as active noise cancellation, has recently attracted much attention from engineers and scientists. This is due to the advancement of the low-cost digital signal processor and the fact that ANC offers advantages in terms of bulk and expenditure over the conventional utilization of passive dampers for attenuating low frequency acoustic noise. The ANC typically employs linear transversal filter algorithm for both identification and control, such as the well known Filtered-X Least Mean Square (FX-LMS) algorithm [4,5,7,12] where adaptive algorithm is used to adapt the FIR filter weights to minimize the mean square error between filter output and desired response[12]. However, such linear

filter, despite its success in a number of experiments, does not work well when there are significant nonlinearities. On the other hand recent development on neural networks has proved that neural networks with its learning capability can be effectively employed to approximate arbitrary nonlinear function, and therefore this method finds wide applications in identification and control of nonlinear and complex systems.

Even though the propagation media in ANC might be linear, other components (especially the power amp, speaker and microphone) of the secondary paths may exhibit noticeable non-linearity. An example of non-linearity is the exponential non-linearity. If the input signal is a pure sinusoidal tone, this kind of non-linearity causes harmonic frequencies of the input signal to appear at the output of the secondary path. Another possible non-linear behavior is saturation in ANC components or in some cases the ANC problem to be addressed is non-linear in nature. Non-linearities arising in the system can substantially reduce the performance of the linear filters employed [9]. To cope with the problem of non-linearity one needs an adaptable non-linear structure. One such structure which has been rigorously studied in the last few years is that of artificial neural networks (ANN) [8,10,13-15,20].

In this paper a new algorithm based on neural networks is proposed for ANC to overcome the problem of nonlinearities. Radial basis function (RBF) neural networks are employed and the corresponding algorithm suited for ANC is developed, by extending the results of [3] which rely on multilayer backpropagation networks. The resulting algorithm is called FX-RBF to represent that secondary path effect of ANC is accounted for in the adaptive algorithm. The secondary paths model will be a RBF neural network with a number of outputs and an input that is a connected to a tapped delay of a time-series. The resulting algorithm turns out to be the filtered-X version of the standard RBF learning algorithm. The new algorithm is based on the idea that instantaneous gradient of the error depends not only on the current values of networks weights, but also on their past values up to the number of tap used in the tapped delay lines

An experimental study is presented to actively control acoustic noise within a 3D enclosure by exploiting the long wavelength associated with low frequency sound. It works on the principle of destructive interference between the sound fields generated by the original “primary” sound source and that due to other “secondary” sources, whose acoustic outputs can be controlled. The objective of the experiment is to cancel the unwanted noise in specific areas within the enclosure known as quiet zones. To get significant noise attenuation in a number of quiet zones, multiple channels ANC employing multiple secondary source and error sensors is introduced.

Floating point TMS320C6701 Digital Signal Processors (DSP) is employed to implement FX-RBF nonlinear algorithm in real-time. Two major steps are employed in this experiment: 1) use the RBF nonlinear adaptive algorithm to identify nonlinear model of the secondary path, 2) use a copy of the identified model together with FX-RBF filter algorithm to perform a "destructive interference" via RBF neural controller in such a way that the error residue is minimized. We address centralized and decentralized controller configuration. In centralized configuration, cross-coupling effects between channels are accounted for. Due to its high computing load, consideration of the effects of cross-coupling channels in active noise control implementing RBF neural networks and its FX-RBF learning algorithm is impractical when performed in real-time. Instead, a decentralized configuration should be employed where cross-coupling effects between channels are neglected. While this configuration may slightly deteriorate performance of the ANC neural controller, it reduces significantly computing power needed. In this paper, real-time DSP implementation of the neural networks controller is used to cancel acoustic noise inside 3D enclosure to produce quiet zones around error microphones. We examine the attenuation level and transient response for both localized and spatially distributed noises. Some preliminary results appeared in [23]. In summary, contribution of this paper over existing literatures in this area[3,12-14,22] is twofold : First, theoretical development of a new algorithm for radial basis function neural networks suited for ANC application, and secondly, experimental study on application of the proposed algorithm to ANC inside 3D enclosure by using real-time DSP.

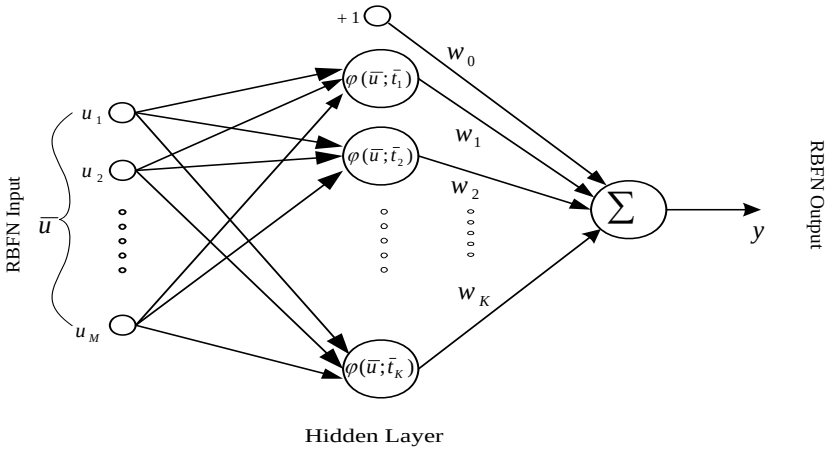
Outline of this paper is as follows. In Section 2, topology of RBF Networks and its computation are briefly presented. In Section 3, active method in noise control is addressed and FX-RBF learning algorithm is derived. DSP based real-time experiment set-up and its configuration are discussed in Section 4. Section 5 presents experimental results for various configuration and cases for both modeling and control tasks. Finally, conclusion is drawn in Section 6.

2 Radial Basis Function Neural Networks

Radial Basis Function Networks (RBFN)[10,22] is a multilayer feedforward neural networks consisting of one input layer, one hidden layer which performs nonlinear processing, and one output layer with linear weights as shown in Figure 1. Also shown in this figure, a Gaussian function which is commonly employed in RBFN as an instance of radial basis function. The structure is drawn from biological receptive fields to perform function mappings. Weights on the output layer are adapted via supervised learning[10]. Input-output mapping of RBFN is described through the following equation

$$y = \sum_{k=1}^{K} w_k \varphi(\bar{u}; \bar{t}_k) + w_0 \quad (1)$$

where $\varphi(\bar{u};\bar{t}_k)$ is a k^{th} radial (kernel) function which computes distance between input vector $\bar{u}$ and function center $\bar{t}_k$. Output signals of the hidden layer are nonlinear function of the distance. The activations of pattern units essentially characterize the distances of centers of radial basis functions of the pattern units



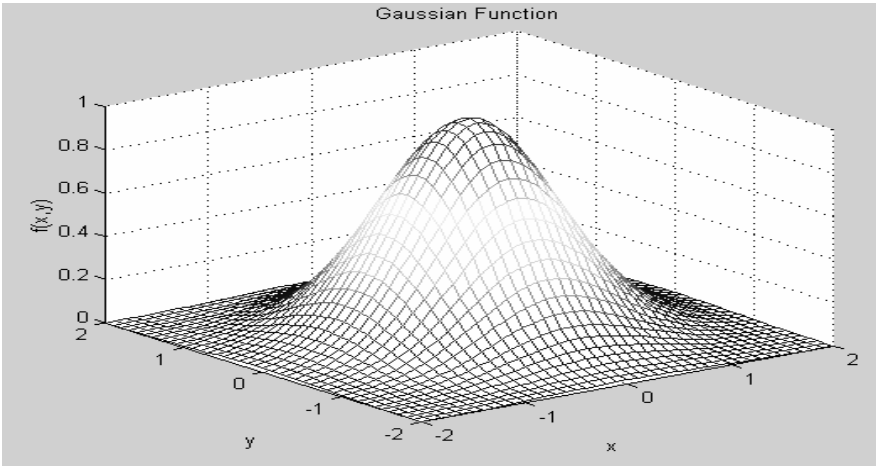


Figure 1 RBFN structure and Gaussian function.

from a given input vector. The radial basis functions thus produce localized, bounded, and radially symmetric activations—that is, activations rapidly decreasing with the distance from the function centers[10,22]. Scaling factor w_k represents weights connecting k^{th} node of the hidden layer and output node,

while w_0 represents the bias. *M* represents the number of inputs and *K* is the number of hidden neurons. A Gaussian function is chosen as the kernel

$$\varphi(\bar{u};\bar{t}_k) = \exp\left(-\frac{1}{\sigma_k^{\,2}}\left\|\bar{u}-\bar{t}_k\right\|^2\right) \tag{2}$$

where $\bar{t}_k$ is the center, σ_k is the width, and $\left\|\bar{u}-\bar{t}_k\right\|$ represents euclidean distance between $\bar{u}$ and $\bar{t}_k$.

Like a multilayer perceptron, RBFN has universal approximation ability. The advantages of RBFN are linearity in parameters and the availability of fast and efficient training methods. RBFN learns to approximate the desired input-output map represented by training data $\{\bar{u}_i, d_i\}$, where $\bar{u}_i$ is the input vector and d_i is the desired response (target), *i = 1, 2,…, N*. A number of learning methods exist to approximate the desired input-output maps[10,23]. In this paper we employ a hybrid learning approach which combines self-organized learning algorithm based on K-Means Clustering Algorithms and supervised learning algorithm based on stochastic gradient. The former is used to determine the center of Gaussian function, while the later is employed to adjust the output weights [10].

K-means clustering algorithm is used to cluster data into *k* number of clusters. Specifically, this algorithm places centers of radial basis function in the input space area where the data are significant. K-means clustering algorithm proceeds as follows :

1. *Initialization*, select randomly center values $\bar{t}_k(0)$; the only requirement is that values of $\bar{t}_k(0)$ must be different for each *k = 1, 2,…, K*. It is suggested that euclid norm of each center sufficiently small.
2. *Sampling*, take a sample vector $\bar{u}$ of input space with certain probability. Vector $\bar{u}$ represents input applied to RBFN.
3. *Similarity matching*, find center of the winner at n^{th} iteration, with minimum euclidean distance :

$$\tilde{k}(\bar{u}) = \arg\min_k \left\|\bar{u}(n)-\bar{t}_k(n)\right\|, \quad k = 1, 2, \ldots, K \tag{3}$$

4. *Updating*, adjust center position according to:

$$\bar{t}_k(n+1) = \begin{cases} \bar{t}_k(n) + \eta\left[\bar{u}(n)-\bar{t}_k(n)\right], & k = \tilde{k}(\bar{u}) \\ \bar{t}_k(n), & \textit{otherwise} \end{cases} \tag{4}$$

where η is learning rate parameter ($0 < \eta < 1$); or it can be adjusted from *1* to *0*, according to

$$\eta(n) = \frac{\eta(n-1)}{\left(1 + \frac{n}{\alpha . f_s}\right)^{\frac{1}{2}}} \tag{5}$$

where *n* is sample number, f_s is sampling frequency; and α is a constant chosen to determine rate of decrement of η .

3 Active Noise Control Using RBFN

The general structure of active noise control is shown in Figure 2. Referring to this figure, the complexity of ANC comes from the existence of secondary path from the control signal input to error measurement. This path consists of frequency response of the control actuator and error sensor, acoustic/structural characteristic and tapped delay line due to the distance between control actuator and error sensor. Intuitively, the existence of this path must be accounted for in the adaptation algorithm to maintain its stability. For ANC based on LMS, such consideration leads to a modification to standard LMS known as FX-LMS[12]. In this scheme, stability of adaptive algorithm is maintained by filtering signal reference to generate control signal through estimated model of the secondary path.

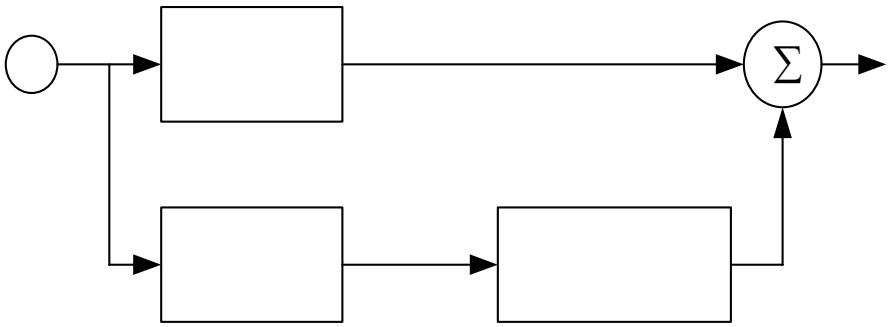


Figure 2 General feedforward ANC system.

Using similar argument, we propose Filtered-X RBF which modifies standard RBF learning algorithm by accounting for secondary path in the weights adaptation process. General structure of nonlinear ANC using Neural Networks is shown in Figure 4, where we use two neural networks. The first neural networks, referred to as controller neural networks, serves as controller, which receives input from reference signal and generates control signal to the

secondary path through actuator. The second neural networks, referred to as model neural networks, serves as secondary path model. Error signal in this figure is measured by error sensor and is employed to perform real-time adaptation of the controller neural networks. Tapped delay lines are inserted in both controller and model neural networks to capture the dynamic behavior of the plant.

To train the model neural networks to approximate the behavior of the secondary path (see Figure 3), standard RBF learning algorithm such as random fixed center based algorithm, stochastic gradient, or hybrid methods can be used [10,23]. Basically, a signal *y(n)* is applied to both secondary path and RBFN, producing *d(n)* and *y'(n)*, respectively. Difference between *d(n)* and *y'(n)* yields error which in turn is used to adjust weights of the RBFN based on the hybrid algorithm. On the other hand, training the controller neural networks can not be done using these standard methods due to the existence of tapped delay lines between controller and model neural networks. Using similar argument to FX-BP adaptive algorithm developed in [3], we introduce filtered-X version of the RBF learning algorithm.

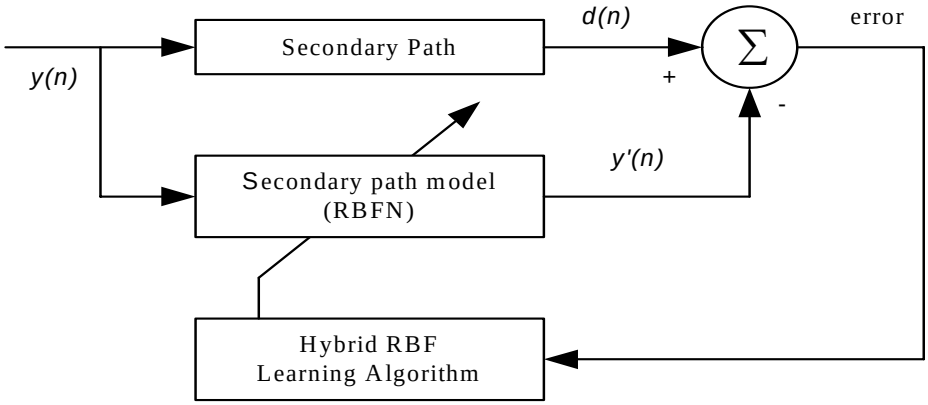


Figure 3 Secondary Path Neural Networks Modeling.

Filtered-X approach of the RBF learning is based on the fact that instantaneous quadratic error at time *n* is affected by weight values of the controller neural networks at the last $L + 1$ samples (weight values at time n, $n - 1$, …, $n - L$), where L is the number of tapped delay lines before the model neural networks. Thus,

$$\text{Instantaneous Gradient} = \frac{1}{2}\sum_{i=0}^{L}\frac{\partial e^2(n)}{\partial w_k(n-i)} \tag{6}$$

where :

$e^2(n)$ = quadratic error at time n
$w_k(n-i)$ = weight values of controller neural networks at time $(n-i)$.

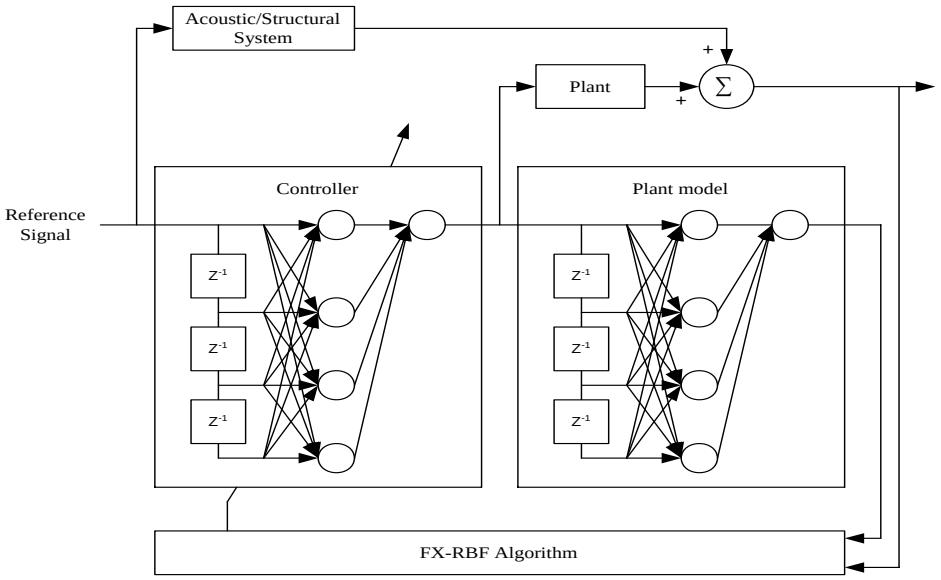


Figure 4 Block Diagram of ANC Using Neural Networks.

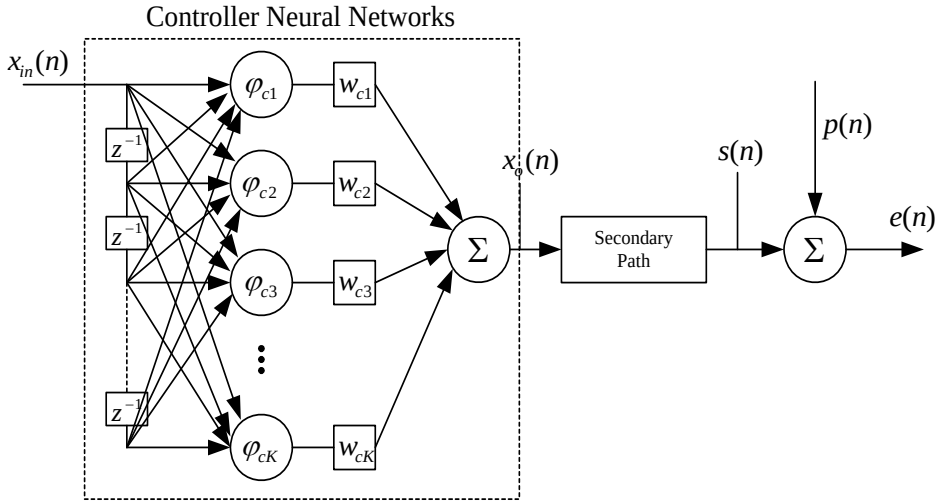


Figure 5 Feedforward Neural Control Scheme.

Development of FX-RBF algorithm to adapt ANC controller is done through Figure 5. In this figure, a sample of input reference signal at time n, $x_{in}(n)$, which correlates with propagating primary noise signal, $p(n)$, is used to generate control signal $x_o(n)$, through controller neural networks. This signal is modified through secondary path model to produce forward secondary signal $s(n)$ (see Figure 6).

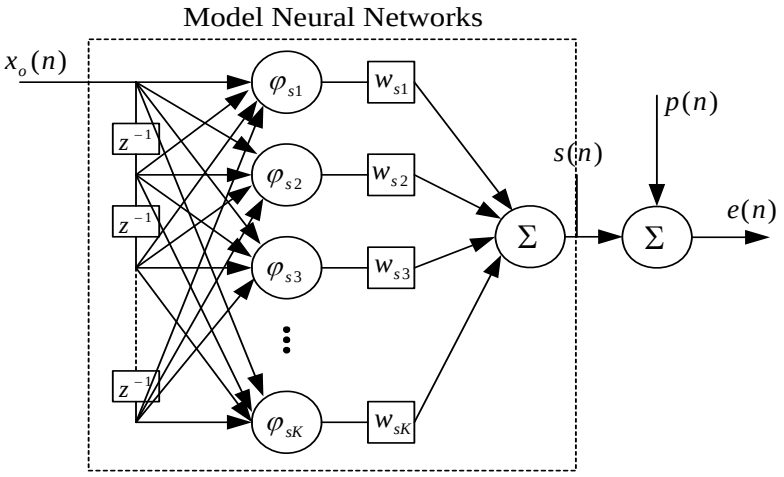


Figure 6 ANC Secondary Path model Using Neural Networks.

The error signal is acoustically a superposisiton of the primary noise and control noise given by

$$e(n) = p(n) + s(n) \tag{7}$$

In subsequent discussion, it is assumed that the model neural networks approximate secondary path with good accuracy. The weights of controller neural networks are adapted according to

$$w(n+1) = \mathrm{w}(n) - \mu\Delta w(n) \tag{8}$$

where μ is convergence coefficient, such that the quadratic error criterion is minimized

$$\xi(n) = \frac{1}{2}e^2(n) \tag{9}$$

To derive FX-RBF, it is assumed that primary disturbance, $p(n)$ is not affected by the weights of controller neural networks.

According to equation (6):

$$\Delta w_{ok}(n) = \frac{1}{2}\sum_{i=0}^{L}\frac{\partial e^2(n)}{\partial w_{ok}(n-i)} \tag{10}$$

Using derivative chain rule and (7):

$$\frac{\partial e^2(n)}{\partial w_{ok}(n)} = 2e(n)\frac{\partial e(n)}{\partial w_{ok}(n)} = 2e(n)\frac{\partial(p(n)+s(n))}{\partial w_{ok}(n)} = 2e(n)\frac{\partial s(n)}{\partial w_{ok}(n)} \tag{11}$$

$$\frac{\partial s(n)}{\partial w_{ok}(n)} = \frac{\partial s(n)}{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}\frac{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}{\partial x_o(n)}\frac{\partial x_o(n)}{\partial w_{ok}(n)} \tag{12}$$

It follows from (1),

$$\frac{\partial s(n)}{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))} = w_{sk}(n) \tag{13}$$

and from (2)

$$\frac{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}{\partial x_o(n)} = -\frac{2}{{w_{is}}^2}\sum_{k=1}^{Ks}\left[\left(x_o(n)-t_{1k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]$$

(14)

where K_s is the number of neuron in the hidden layer of secondary model neural networks.

Now, using (1), it follows that

$$\frac{\partial x_o(n)}{\partial w_{ok}(n)} = \varphi(\bar{X}_{in}(n);\bar{t}_{ck}(n)) \tag{14}$$

Substituing equations (12), (13), and (14) into equation (11) leads to :

$$\frac{1}{2}\frac{\partial e^2(n)}{\partial w_{ok}(n)} = -\frac{2e(n)}{{w_{is}}^2}\left[\sum_{k=1}^{Ks}\left[w_{sk}\left(x_o(n)-t_{1k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]\right]\varphi(\bar{X}_{in}(n);\bar{t}_{ck}(n)) \tag{15}$$

Analogously, the derivative terms $\frac{\partial e^2(n)}{\partial w_{ok}(n-1)}$, $\frac{\partial e^2(n)}{\partial w_{ok}(n-2)}$,…., $\frac{\partial e^2(n)}{\partial w_{ok}(n-L)}$ can be derived by derivative chain rule, through the following equations:

$$\frac{\partial s(n)}{\partial w_{ok}(n-1)} = \frac{\partial s(n)}{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}\frac{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}{\partial x_o(n-1)}\frac{\partial x_o(n-1)}{\partial w_{ok}(n-1)} \tag{16}$$

$$\frac{\partial s(n)}{\partial w_{ok}(n-2)} = \frac{\partial s(n)}{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))} \frac{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}{\partial x_o(n-2)} \frac{\partial x_o(n-2)}{\partial w_{ok}(n-2)} \tag{17}$$

$$\frac{\partial s(n)}{\partial w_{ok}(n-L)} = \frac{\partial s(n)}{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))} \frac{\partial \varphi(\bar{X}_o(n);\bar{t}_{sk}(n))}{\partial x_o(n-L)} \frac{\partial x_o(n-L)}{\partial w_{ok}(n-L)} \tag{18}$$

Now, analogously to derivation of equation (15), we have

$$\frac{1}{2}\frac{\partial e^2(n)}{\partial w_{ok}(n-1)} = -\frac{2e(n)}{w_{is}^{\ 2}}\left[\sum_{k=1}^{Ks}\left[w_{sk}\left(x_o(n-1)-t_{2k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]\right]\varphi(\bar{X}_{in}(n-1);\bar{t}_{ck}(n-1)) \tag{19}$$

$$\frac{1}{2}\frac{\partial e^2(n)}{\partial w_{ok}(n-2)} = -\frac{2e(n)}{w_{is}^{\ 2}}\left[\sum_{k=1}^{Ks}\left[w_{sk}\left(x_o(n-2)-t_{3k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]\right]\varphi(\bar{X}_{in}(n-2);\bar{t}_{ck}(n-2)) \tag{20}$$

$$\frac{1}{2}\frac{\partial e^2(n)}{\partial w_{ok}(n-L)} = -\frac{2e(n)}{w_{is}^{\ 2}}\left[\sum_{k=1}^{Ks}\left[w_{sk}\left(x_o(n-L)-t_{(L+1)k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]\right]\varphi(\bar{X}_{in}(n-L);\bar{t}_{ck}(n-L)) \tag{21}$$

and using (10), (15), (19)-(21), the instantaneous gradient can be computed as

$$\Delta w_{ok}(n) = -\frac{2e(n)}{w_{is}^{\ 2}}\sum_{j=0}^{L}\left[\left[\sum_{k=1}^{Ks}\left[w_{sk}\left(x_o(n-j)-t_{(j+1)k}(n)\right)\varphi(\bar{X}_o(n);\bar{t}_{sk}(n))\right]\right]\varphi(\bar{X}_{in}(n-j);\bar{t}_{ck}(n-j))\right] \tag{22}$$

Equation (22) represents gradient of the instantaneous square error with respect to weights of neural networks controller in ANC structure shown in Figure 3, taking into account that the error at time *n* is not only affected by weight values of the controller neural networks at time *n*, but also at time *n - 1, n-2 …, n – L*, where *L* is the number of tapped delay lines inserted between model and controller neural networks. By substituting (22) into (8), neural network weights on the output layer of controller neural networks can be updated. The algorithm is referred to as Filtered-X RBF to indicate that it is filtered-X version of standard RBF learning algorithm. As described previously, center of radial basis function can be determined using K-means clustering algorithm. In summary, the model RBFN is trained separately using standard hybrid method based on input-output time-historical data of the secondary path. Then, using this neural networks model, controller neural networks is trained using FX-RBF, see Equation (8) and (22). Center of Gaussian functions for both model and controller neural networks are trained using K-means clustering algorithm.

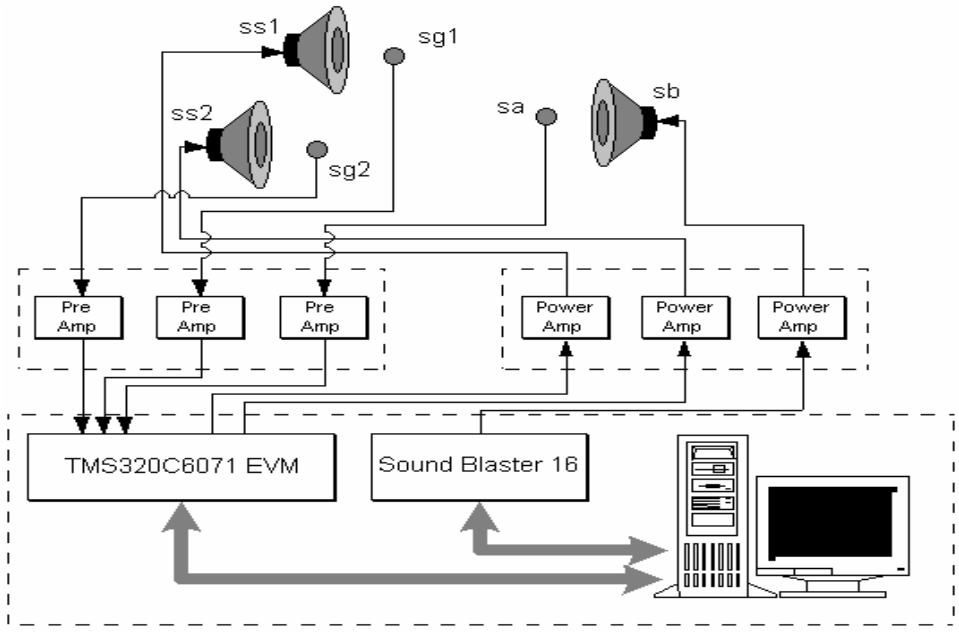


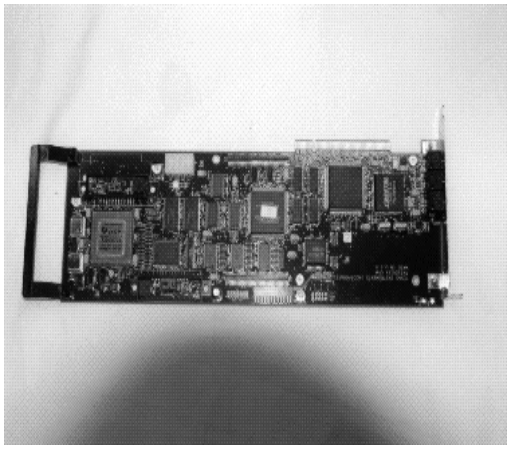
Figure 7 ANC Experiment Setup (ss: secondary sources, sg: error sensors, sb: primary noise source, sa: reference sensor).

4 Experiment Set-up and Configuration

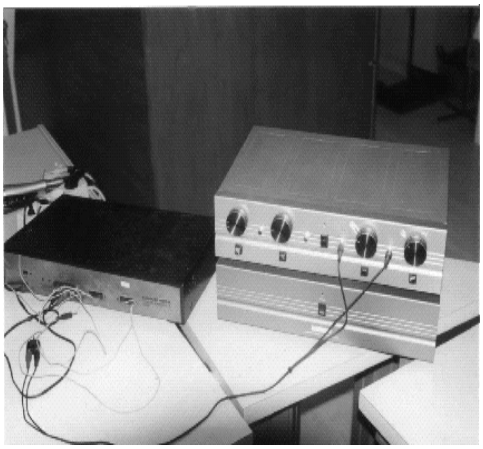
4.1 Experiment Setup

Experiment setup of multichannel ANC inside an enclosure is shown in Figure 7. In this setup, two error sensors (microphones), two actuators (6“, 8 Ohm speakers), and one reference sensor are used. The objective of the ANC is to obtain quiet zones around each error microphone. To implement active noise control computation and data acquisition, TMS320C6701 Evaluation Module DSP Board is used. The board is based on floating point TMS320C6701 DSP processor. The board is interfaced with computer host through PCI to enable real-time data exchange.

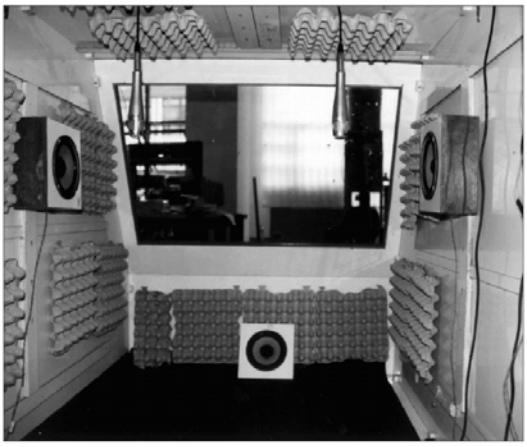
Primary noise is generated through Sound Blaster 16 and controlled from within Windows operating system. The output of sound blaster, as well as of DSP, are applied to power amplifiers, which in turn drive the speakers. Pre-amplifiers are used to amplify the signal measured by the microphones. Coding, debugging and real-time analysis are performed via DSP Code Composer Studio. In all experiments, we use 1000 Hz sampling rate. Photograph of ANC experiment components and setup is shown in Figure 8.



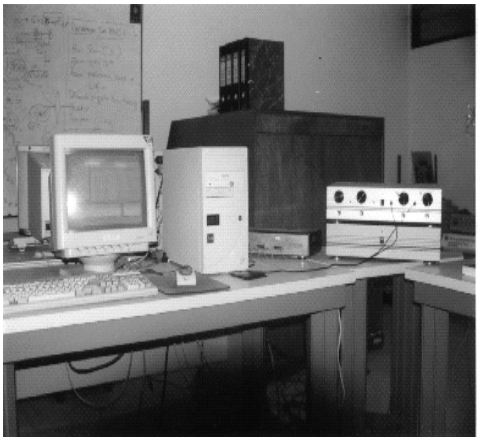
(a) DSP Board



(b) Pre-amp, Power amp, ADC/DAC



(c) 3D Enclosure



(d) ANC System

Figure 8 Photograph of ANC Experiment Components and Setup.

4.2 Experiment Configuration

As previously discussed, an identification process is performed to model secondary path of ANC prior to control operation. The secondary path includes DAC, speaker, acoustic path, microphone and ADC. Identification process for a single channel secondary path is illustrated in Figure 3. This process is done in real-time using DSP Board. For ANC with decentralized configuration this process is performed for each channel by ignoring coupling between channels, as illustrated in Figure 9 for two channels.

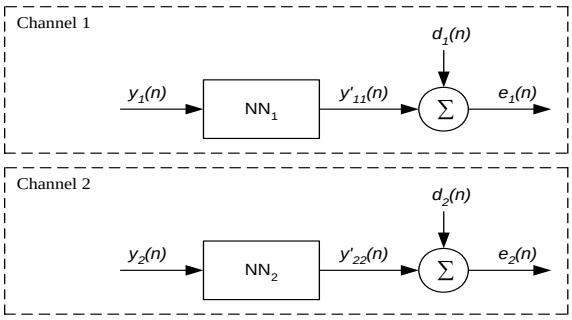


Figure 9 NN_1 and NN_2 Models for Each ANC Channel.

ANC experiments are performed for single channel as well as multichannel. In case of mutichannel configuration, decentralized controller structure is adopted as shown in Figure 9. In this structure, separate controller is implemented for each channel by ignoring cross-coupling between channels. However, all controllers and secondary models are realized in the same DSP board. Such structure is adopted to reduce computational load of the DSP.

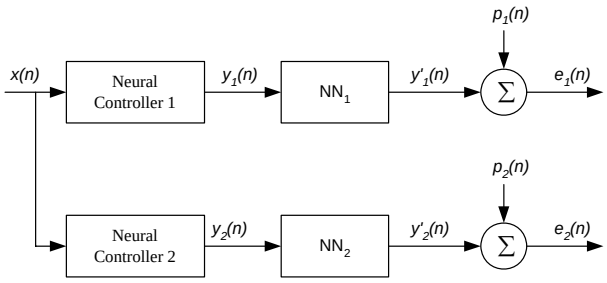


Figure 10 Decentralized ANC Structure.

The overall structure of ANC using RBF neural networks is shown in the Figure 3. In order to effectively cancel out the primary noise at the sensor, a reference signal with strong correlation to the primary noise is directly made available in a feedforward fashion.

5 Experiment Results

In this section, we present experiment results for single channel as well as for multichannel ANC. Geometry of multichannel ANC is shown in Figure 11, where we use two error microphones, two speakers and one reference microphone. This construction is arranged to mimic a cabin of a vehicle. The position of actuators and sensors is determined around ear positions of passengers inside the cabin.

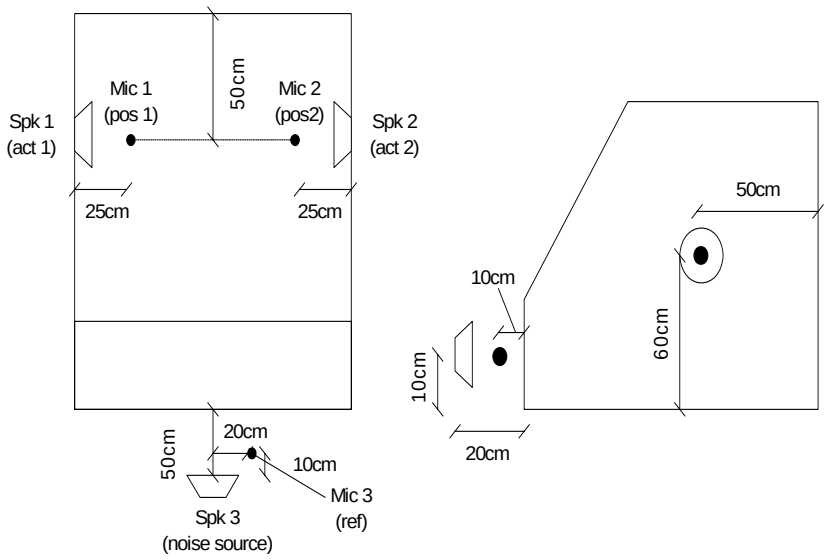


Figure 11 ANC Geometry Showing Position of Microphones and Speakers.

First, we present experiment results for neural modeling of secondary path of ANC. In this experiment the RBF weight adaptation is done in real-time on DSP. Figure 12 shows the error of secondary path identification from speaker 2 to microphone 1 with 170Hz excitation signal. As shown in this figure, the error tends to decrease with transient behavior occurring in the first 3.5 seconds. We observe from this figure that the RBF neural model approximate secondary path quite well. Similar observation can be drawn for neural modeling of secondary path between speaker 1 and microphone 2 as shown in Figure 13. We perform the neural modeling experiment for several cases by varying frequency of excitation signal, number of neurons in the RBF's hidden layer, and position of sensor and actuator. The overall results of modeling experiment are presented in Table 1. The number of neurons is chosen in such a way that the computation load for the DSP remains moderate, but high enough to carry out the required learning task effectively. Shown in this table are positions of sensor and actuator, frequency of excitation signal, number of neuron in the RBF's hidden layer, mean square error (MSE), signal to noise ratio (SNR), and settling time (number of samples required to reach steady state error). As can be seen from this figure for sensor 1 and actuator 1, there are cases (Experiment no. 1-3) where increasing the number of hidden neuron shall decrease MSE and improve SNR. However, further increase of number of hidden neuron may deteriorate performance of neural modeling as indicated by Experiment no. 4. In these particular cases, settling time is not significantly affected by the number of neuron in the hidden layer. Frequency content of the excitation signal is also

varied and the corresponding results are shown in Experiment no. 8-10 in Table 1. The results show that settling time is significantly affected by frequency content of the noise signal. Neural identification process is performed for other secondary channels as well to be used in centralized and decentralized configuration of ANC.

Next, we present experiment results for control task in RBF neural networks based ANC. Two ANC performance measures are used in the experiment : Total Attenuation (in dB), which calculates noise reduction over all frequencies, and Attenuation at (Main) Noise Frequency (in dB), which calculates noise reduction only at the main frequency of noise. Figure 14 shows the error residue of single channel ANC using RBF neural networks measured at error microphone 1 for acoustic noise with frequency of 170 Hz. In this experiment we use 3-10-1 topology for model neural networks and 3-48-1 topology for controller neural networks. The result shows that steady state error is achieved after 0.5 second. Frequency spectra of the primary noise, as compared to noise residue at error microphone 1, shows that attenuation of 42 dB is obtained at frequency of 170 Hz. To demonstrate generalization capability of RBF neural controller, acoustic noise of 210Hz is generated inside the enclosure. RBF neural controller trained for 170Hz is employed without retraining it for 210Hz noise. The results are shown in Figure 15. Note that a good attenuation level is obtained with moderate settling time, which shows generalization capability of RBF neural controller in ANC. The single channel ANC experiment is also carried out by generating noise signal with different frequency content (see Table 2). The frequencies of excitation signals are chosen to represent low frequency spectra where passive noise control does not perform effectively.

Finally, we present experiment results for multichannel ANC using RBF neural controller. Bottom rows of Table 2 show the results of decentralized and centralized multichannel ANC with 170Hz noise. The result shows that steady state error is achieved after 2 seconds. Though not shown, from frequency spectra of the primary noise as compared to noise residue at error microphone, attenuation level of 35-45 dB is obtained at frequency 170 Hz. Decentralized configuration is employed in this real-time experiment to reduce computation load experienced by DSP in centralized configuration. The experiment is also performed by using different noise geometry: spatially localized and spatially distributed. The overall experiment results for control task in ANC are shown in Table 2. In case of localized noise, in this particular experiment, increasing number of neuron in the hidden layer increases total noise attenuation as well as SNR, but decreases transient response. Overall, the results reported in this paper are better than those obtained in [18] using U-Filtered LMS algorithhm, which is based on linear (IIR) adaptive filter. Since neural networks can be viewed as a nonlinear adaptive filter, results of the modeling and control tasks reported in

this paper show that RBF can be effectively employed to model nonlinear dynamics of ANC secondary path and that Filtered-X RBF can be applied to compensate nonlinear phenomena which is inherently arising in a number of practical ANC systems. Further theoretical and experimental justification on such capabilities of neural networks can be found in [24] where recurrent neural networks with extended Kalman Filter algorithm is developed.

6 Conclusion

Nonlinear active noise control using Radial Basis Function Networks and its DSP real-time implementation have been presented. New algorithm which generalizes RBF learning suited for ANC application was proposed. Results of the real-time experiment showed that 10-23 dB total attenuation level is obtained at in the desired quiet zones. On-line adaptation of RBF weight has been possible because of the computing power of floating point DSP used and its support to multichannel application. Experimental results for single channel ANC and multi-channel secondary path modeling and control demonstrated that the neural networks along with DSP implementation provide effective method for controlling multi-channel acoustic noise under the presence of unknown nonlinear phenomena.

References

1. Beranek, L. L. & Ver, I. L., *Noise and Vibration Control Engineering: Principles and Applications*, New York: Wiley (1992).
2. Bies, D. A. & Hansen, C. H., *Engineering Noise Control: Theory and Practice*, 2ed, London: E & FN Spon (1996).
3. Bouchard, M., Paillard, B. & Le Dinh, C. T., *Improved Training of Neural Networks for the Nonlinear Active Control of Sound and Vibration*, IEEE Transactions on Neural Networks, Vol. 10, No. 2, pp. 391-401 (1999).
4. Elliot, S. J., *Down with Noise*, Proceedings of the IEEE (1999).
5. Elliot, S. J. & Nelson, P. A., *Active noise control*, IEEE Signal Processing Magazine, Vol. 10, No. 4, pp. 12-35 (1993).
6. Elliot, S. J. & Bouncher, C. C., *Interaction between multiple feedforward active noise control systems*, IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, pp. 521-530, 1994.
7. Fuller, C. R. & von Flotow, A. H., *Active control of sound and vibration*, IEEE Control Systems Magazine, Vol. 15, No. 6, pp. 9-19 (1995).
8. Funahashi, K., *On the approximate realization of continuous mappings by neural networks*, Neural Networks 2, pp. 183-192 (1989).
9. Haykin, S., Neural Networks: *A Comprehensive Foundation*, New York: Macmillan College Publishing Company, Inc. (1998).

10. Hong, J. et al., *Modeling, identification, and feedback control of noise in an acoustic duct*, IEEE Transactions on Control Systems Technology, pp. 283-291 (1996).
11. Kuo, M. S. & Morgan, D. R., *Active Noise Control Systems: Algorithms and DSP Implementations*, New York: John Wiley & Sons, Inc. (1996).
12. Narendra, K. S. & Parthasarathy, K., *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, Vol 1, No. 1, pp. 4-27 (1990).
13. Narendra, K. S. & Parthasarathy, K., *Gradient methods for the optimization of Dynamical systems containing neural networks*, IEEE Transactions on Neural Networks, Vol. 2, pp. 252-262 (1991).
14. Narendra, K. S. & Mukhopadhyay, S., *Adaptive Control Using Neural Networks and Approximate Models*, IEEE Transactions on Neural Networks, Vol. 8, No. 3, pp. 475-485 (1997).
15. Nelson, P. A. & Elliott, S. J., *Active Control of Sound*, Cambridge: Academic Press (1995).
16. Snyder, D. S. & Tanaka, N., *Active Control of Vibration Using a Neural Networks*, IEEE Transactions on Neural Networks, Vol. 6, No. 4, pp. 819-828 (July 1995).
17. Widrow, B., *Inverse Adaptive Control*, Prentice Hall (1995).
18. Riyanto, B., *Decentralized Active Noise Control Using U-Filtered Algorithm: An Experimental Study*, Int. Conf. On Modeling, Identification and Control, Innsbruck, Austria (2000).
19. Riyanto, B., *On-Line Secondary Path Identification of Active Noise Control Using Neural Networks*, Int. Conf. Modeling and Simulation, Pittsburgh , USA (2000).
20. Riyanto, B., *Active Control of Acoustic Noise Using Adaptive Filter Algorithm: A Real-Time DSP Implementation Based on Feedforward Configuration*, ISASTI '98 (1998).
21. Riyanto, B. & Uchida, K., *Active Control of Acoustic Noise Using Adaptive H_∞ Filter Algorithm*, Movic 01, Osaka (2001).
22. Poggio, T. & Girosi, F., *Networks for Approximation and Learning*, Proceedings of the IEEE (1990).
23. Riyanto, B., Nasution, L. & Uchida, K., *Active Control of Acoustic Noise Using Radial Basis Function Networks*, International Conf. On Modeling, Identification and Control, Innsbruck, Austria (2001).
24. Riyanto, B., Yacoub, R. & Uchida, K., *Identification of Secondary Path in ANC Using Diagonal Recurrent Neural Networks with EKF Algorithm*, Proc. 5th ASCC, Melbourne(2004)

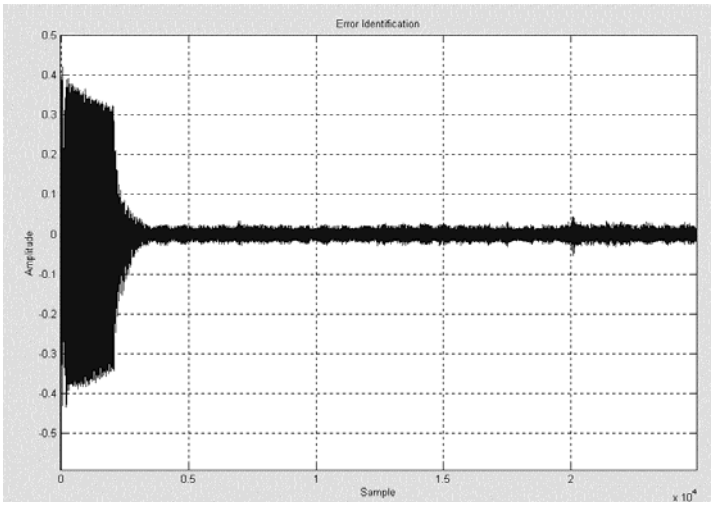


Figure 12 Error of Single Channel Identification of ANC Path between Microphone 1 and Actuator 2.

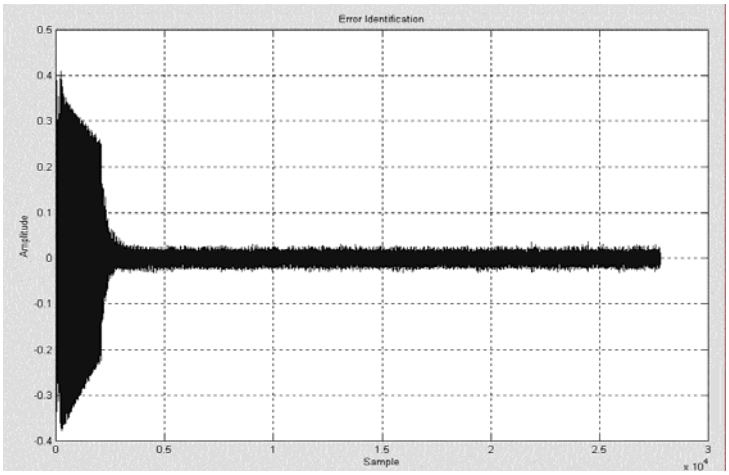


Figure 13 Error of Single Channel Identification of ANC Path between Actuator 2 and Microphone 1.

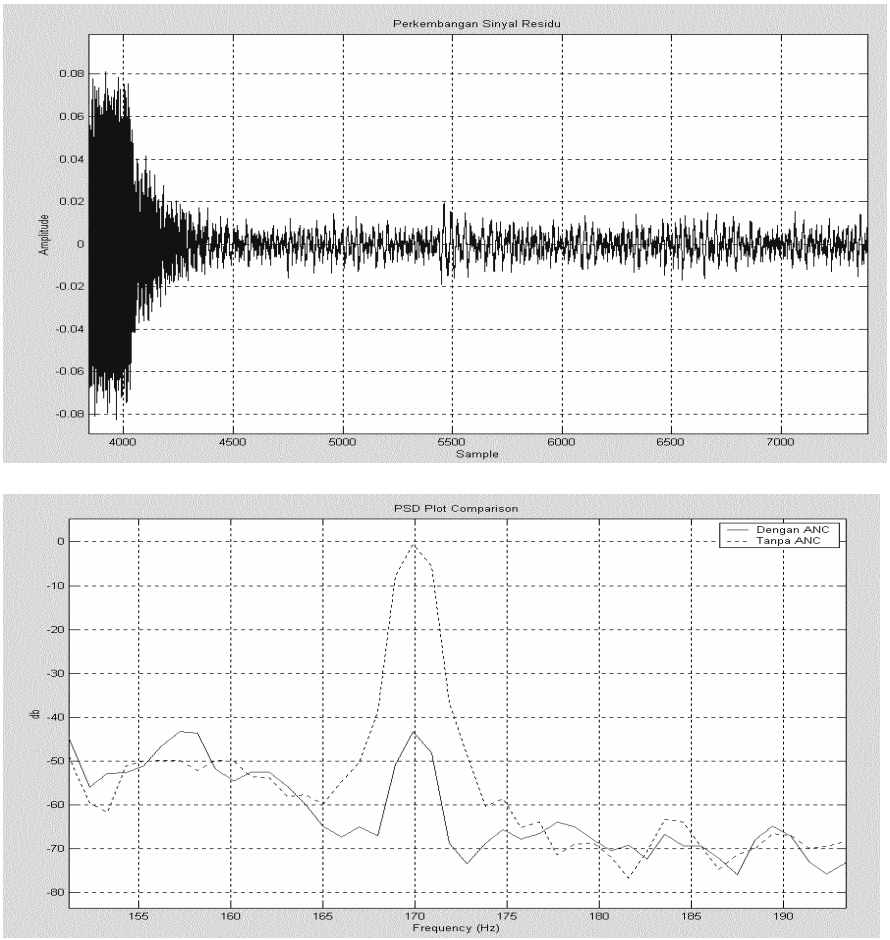


Figure 14 Experiment Result of Single Channel ANC with 170 Hz Noise (measured at microphone 1) : Residue Respose (top) and PSD Comparison between ANC Off and ANC On (bottom)

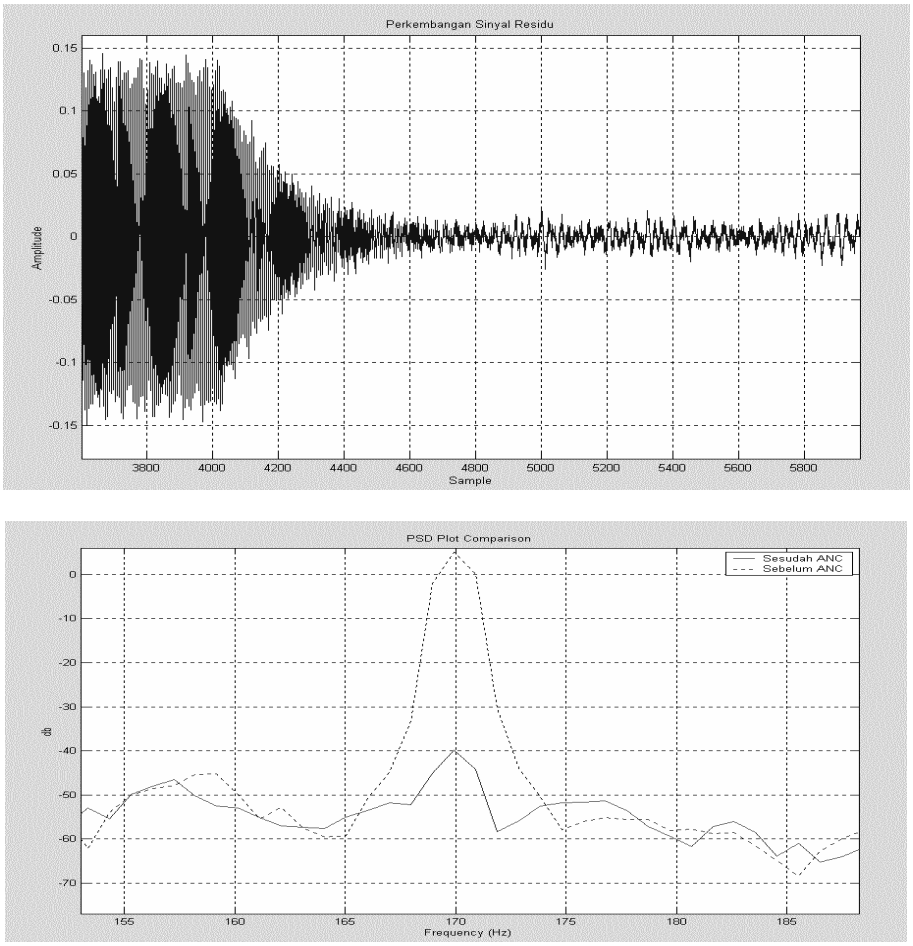


Figure 15 Experiment Result of Single Channel ANC with 210 Hz Noise (measured at microphone 1) without Retraining: Residue Respose (top) and PSD Comparison between ANC Off and ANC On (bottom).

Secondary Path Neural Modeling									
No	Act Position	Sensor Position	Excitation Signal	Number of Neuron		MSE	SNR(db)	Transient (sample)	Transient (second)
				In	Hn				
1	1	1	170 Hz + Hr	6	10	2.13910E-04	27.0968	3500	3.5
2	1	1	170 Hz + Hr	6	15	5.39340E-05	34.0814	3500	3.5
3	1	1	170 Hz + Hr	6	6	7.62020E-04	22.1088	3500	3.5
4	1	1	170 Hz + Hr	6	30	8.01400E-05	31.2088	3500	3.5
5	1	2	170 Hz + Hr	6	10	1.03650E-04	28.8105	3500	3.5
6	2	1	170 Hz + Hr	6	10	1.12920E-04	29.3433	3500	3.5
7	2	2	170 Hz + Hr	6	10	1.60590E-04	28.7643	3500	3.5
8	1	1	100+170 Hz	6	10	1.90000E-03	12.2698	8000	8
9	1	1	100+170 Hz	12	50	2.51900E-04	20.9193	2500	2.5
10	1	1	100+170 Hz	12	70	2.92300E-04	20.6543	4500	4.5
11	1	1	170 Hz + Hr	6	10	1.95070E-04	26.8607	3000	3
		2				1.94860E-04	26.8661	4000	4
12	2	1	170 Hz + Hr	6	10	4.65590E-05	29.8405	3800	3.8
		2				4.65630E-05	29.8429	3500	3.5

Table 1 Experiment Results of RBF Neural Networks Modeling.

Sensor and Actuator Position	170 Hz Localized Noise			
	Number of Neuron in Controller NN	Total Attenuation (dB)	Attenuation at Noise Frequency (dB)	Transient (second)
Actuator 1 & Sensor 1	3 – 48 – 1	20.0912	42	0.5
Actuator 1 & Sensor 2	3 – 48 – 1	18.0555	63	2
Actuator 2 & Sensor 1	3 – 48 – 1	21.8274	48	0.5
Actuator 2 & Sensor 2	3 – 48 – 1	17.0939	42	0.75
Actuator 1 & Sensor 1	3 – 5 – 1	10.7633	11	>21
Actuator 1 & Sensor 1	3 – 10 – 1	20.2227	28	16
Actuator 1 & Sensor 1	3 – 15 – 1	21.7436	28	13
Actuator 1 & Sensor 1	3 – 48 – 1	170 Hz Spatially Distributed Noise		
		17.9843	45	1
Actuator 1 & Sensor 1	3 – 48 – 1	210 Hz Localized Noise		
		21.6	40	2
Actuator 1 & Sensor 1	3 – 48 – 1	(100 Hz + 170 Hz) Localized Noise		
		10.7974	100 Hz	170 Hz
			0	27
Decentralized Multi-Channel (1 sensor , 2 actuators)				
Controller	Neuron Size	170 Hz Localized Noise		
1	3 – 15 – 1	21.6752	38	2
2	3 – 15 – 1			
		170 Hz Spatially Distributed Noise		
1	3 – 15 – 1	17.6863	45	1.5
2	3 – 15 – 1			
Centralized Multichannel (1 sensor, 2 actuators)				
Controller	Neuron Size	170 Hz Localized Noise		
1	3 – 20 – 1	22.855	44	0.6
		170 Hz Distributed Noise		
1	3 – 20 – 1	19.8381	35	0.5

Table 2 Experiment Results of RBF Neural Networks Based ANC.