DISTRIBUTED ADAPTIVE ALGORITHMS FOR LARGE DIMENSIONAL MIMO SYSTEMS

Barry D. Van Veen, Olivier Leblond, Vijay P. Mani, and Daniel Sebald

Electrical and Computer Engineering Department
1415 Engineering Drive
Madison, WI 53706
vanveen@engr.wisc.edu

ABSTRACT

A distributed algorithm for MIMO adaptive filtering is introduced. This algorithm distributes the adaptive computation over a set of linearly connected computational modules. Each module transmits data to and receives data from its nearest neighbor. A back-propagation LMS based algorithm is presented for adapting the parameters in each module. The performance surface is explored to identify upper bounds on each parameter and guidelines for choosing the LMS algorithm step sizes. An example illustrates application of the algorithm.

1. INTRODUCTION

Adaptive systems with large numbers of inputs and outputs present significant computational challenges since the number of adaptive filters is given by the product of the number of inputs and outputs. For example, many tens of inputs and outputs are often used in active noise control applications [1], [2].

In this paper we present a distributed algorithm for implementing adaptive MIMO systems based on splitting the adaptive algorithm over a set of linearly connected computational modules [3]. Each module is assumed to have at least one input signal, one output signal, and exchanges data with its two neighboring modules. The output of each module is based on a linear combination of its input signal and the data from adjacent modules. The data passed to the module on the right (left) is a linear combination of the input and data received from the module on the left (right). This approach distributes the computational burden over many local processors and improves the fault tolerance of the system. It also allows the number of inputs and outputs to be increased or decreased by simply adding or removing modules.

The paper is organized as follows. First, the function of each module is defined and the MIMO impulse response expressed in terms of the parameters associated with each module. An LMS based adaptive algorithm for adjusting the parameters in each module is then presented. Guidelines for selecting the initial conditions and step sizes associated with each parameter are given. The paper concludes with an example illustrating performance of the algorithm.

This work supported by Digisonix, Inc., Madison, WI and Nelson Industries, Inc., Stoughton, WI

2. MODULAR IMPLEMENTATION OF MIMO SYSTEMS

For simplicity of presentation we consider a J input, J output MIMO system consisting of M tap FIR filters. Let $u_i(n)$ denote the input from the i^{th} channel and $y_j(n)$ represent the output to the j^{th} channel. Define $\mathbf{u}_i(n) = [\, u_i(n)\; u_i(n-1) \dots u_i(n-M+1)\,]^T$ as a column vector of present and past inputs so that the j^{th} output is expressed as

$$y_j(n) = \sum_{i=1}^{J} \mathbf{h}_{i,j}^T \mathbf{u}_i(n), \qquad j = 1, 2, \dots, J \tag{1}$$

where $\mathbf{h}_{i,j}$ is a vector representing the impulse response from input i to output j. Equation (1) is rewritten in matrix form as

$$\begin{bmatrix} y_1(n) \\ y_2(n) \\ \vdots \\ y_J(n) \end{bmatrix} = \mathbf{H}^T \begin{bmatrix} \mathbf{u}_1(n) \\ \mathbf{u}_2(n) \\ \vdots \\ \mathbf{u}_J(n) \end{bmatrix} \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_{1,1} & \mathbf{h}_{1,2} & \dots & \mathbf{h}_{1,J} \\ \mathbf{h}_{2,1} & \mathbf{h}_{2,2} & \dots & \mathbf{h}_{2,J} \\ \vdots & & & \vdots \\ \mathbf{h}_{J,1} & \mathbf{h}_{J,2} & \dots & \mathbf{h}_{J,J} \end{bmatrix} \tag{3}$$

Without loss of generality, we assume each module or node in the modular implementation has one input and one output, as illustrated in Fig. 1. The output for the i^{th} node is defined by

$$y_i(n) = \mathbf{w}_{i,i}^T \mathbf{u}_i(n) + \mathbf{w}_{i-1,i}^T \mathbf{s}_{i-1,i}(n) + \mathbf{w}_{i+1,i}^T \mathbf{s}_{i+1,i}(n) \tag{4}$$

where $\mathbf{w}_{i,i}, \mathbf{w}_{i-1,i}, \mathbf{w}_{i+1,i}$ are weight vectors, and $\mathbf{s}_{i-1,i}(n)$ and $\mathbf{s}_{i+1,i}(n)$ denote the P-dimensional data vector communicated to node i from nodes $i-1$ and $i+1$, respectively. That is, the output of each node is a linear combination of the input and the data received from adjacent nodes. The i^{th} node also determines data vectors $\mathbf{s}_{i,i-1}(n)$ and $\mathbf{s}_{i,i+1}(n)$ for adjacent nodes as follows

$$\mathbf{s}_{i,i-1}(n) = \mathbf{K}_{i,i-1}^T \mathbf{u}_i(n) + \mathbf{K}_{i+1,i-1}^T \mathbf{s}_{i+1,i}(n) \tag{5}$$

$$\mathbf{s}_{i,i+1}(n) = \mathbf{K}_{i,i+1}^T \mathbf{u}_i(n) + \mathbf{K}_{i-1,i+1}^T \mathbf{s}_{i-1,i}(n) \tag{6}$$

Here the matrices $\mathbf{K}_{i,i+1}$ and $\mathbf{K}_{i,i-1}$ are M by P while $\mathbf{K}_{i+1,i-1}$ and $\mathbf{K}_{i-1,i+1}$ are P by P. That is, the data shared to the left (right) is a linear combination of the input and the data received from the right (left). Our convention here is to use the first subscript in doubly subscripted quantities to represent "from", and the second subscript to denote "to". For example, $\mathbf{K}_{i-1,i+1}$ describes how data is communicated "from" node $i-1$ "to" node $i+1$.

The impulse response from any input to any output is generally a function of several $\mathbf{w}$ vectors and multiple $\mathbf{K}$ matrices. Using Eqs. (4), (5), and (6), it is a straightforward excercise to show that

$$\mathbf{h}_{i,j} = \begin{cases} \mathbf{w}_{i,i}, & i=j \\ \mathbf{K}_{i,i+1}\prod_{m=i+1}^{j-1}\mathbf{K}_{m-1,m+1}\mathbf{w}_{j,j-1}, & j>i \\ \mathbf{K}_{i,i-1}\prod_{m=i-1}^{j+1}\mathbf{K}_{m+1,m-1}\mathbf{w}_{j,j+1}, & j<i \end{cases} \quad (7)$$

If $J=4$, then the first two columns of $\mathbf{H}$ in Eq. (3) are

$$\begin{array}{cc} \mathbf{w}_{1,1} & \mathbf{K}_{2,1}\mathbf{w}_{2,1} \\ \mathbf{K}_{1,2}\mathbf{w}_{1,2} & \mathbf{w}_{2,2} \\ \mathbf{K}_{1,2}\mathbf{K}_{1,3}\mathbf{w}_{2,3} & \mathbf{K}_{2,3}\mathbf{w}_{2,3} \\ \mathbf{K}_{1,2}\mathbf{K}_{1,3}\mathbf{K}_{2,4}\mathbf{w}_{3,4} & \mathbf{K}_{2,3}\mathbf{K}_{2,4}\mathbf{w}_{3,4} \end{array}$$

and the last two columns of $\mathbf{H}$ are

$$\begin{array}{cc} \mathbf{K}_{3,2}\mathbf{K}_{3,1}\mathbf{w}_{2,1} & \mathbf{K}_{4,3}\mathbf{K}_{4,2}\mathbf{K}_{3,1}\mathbf{w}_{2,1} \\ \mathbf{K}_{3,2}\mathbf{w}_{3,2} & \mathbf{K}_{4,3}\mathbf{K}_{4,2}\mathbf{w}_{3,2} \\ \mathbf{w}_{3,3} & \mathbf{K}_{4,3}\mathbf{w}_{4,3} \\ \mathbf{K}_{3,4}\mathbf{w}_{3,4} & \mathbf{w}_{4,4} \end{array}$$

Note that there is full coupling between all inputs and outputs, since all the entries in $\mathbf{H}$ are nonzero.

3. LMS BASED ADAPTIVE ALGORITHM

In general, we consider the situation in which the errors are measured at the output of a second, known MIMO system. In an active noise control application, this second system represents the combined effects of the speakers and acoustic paths from each speaker to each error microphone. Let $\mathbf{c}_{k,l}$ denote the length N impulse response between the k^{th} output, $y_k(n)$, and the point at which the l^{th} error is measured. Let $d_l(n)$ denote the desired signal and define $\mathbf{y}_k(n) = [y_k(n)\ y_k(n-1)\ \ldots\ y_k(n-N)]^T$ in order to write the error as

$$e_l(n) = d_l(n) - \sum_{k=1}^{J}\mathbf{c}_{k,l}^T\mathbf{y}_k(n) \quad (8)$$

The cost function to be minimized is the mean-squared error (MSE)

$$C = \sum_{l=1}^{J} E\{e_l^2(n)\} \quad (9)$$

The MSE depends only indirectly on the parameters that we seek to adapt, the $\mathbf{w}$'s and $\mathbf{K}$'s. Furthermore, the MSE depends on products of $\mathbf{w}$'s and $\mathbf{K}$'s because each $\mathbf{h}_{i,j}$ in (1) is a product of one $\mathbf{w}$ and several $\mathbf{K}$'s. Hence, we have taken a back-propagation approach [4], [5] to develop an LMS based adaptive algorithm. First, the error is propagated back through the $\mathbf{c}_{k,l}$ filters using the temporal back-propagation algorithm described in [5]. Next, the error is propagated back through the $\mathbf{w}$'s and $\mathbf{K}$'s to obtain local errors for each parameter in the modular implementation. Note that in contrast to a neural network application of back propagation, the errors do not need to be back propagated through nonlinearities. Space constraints preclude full derivation of the algorithm, so we only present the result.

The $\mathbf{w}$'s are adapted as follows:

$$\begin{aligned} \mathbf{w}_{i,i}(n+1) &= \mathbf{w}_{i,i}(n) - \mu_1\delta_i^y(n)\mathbf{u}_i(n-N) \\ \mathbf{w}_{i+1,i}(n+1) &= \mathbf{w}_{i+1,i}(n) - \mu_2\delta_i^y(n)\mathbf{s}_{i+1,i}(n-N) \\ \mathbf{w}_{i-1,i}(n+1) &= \mathbf{w}_{i-1,i}(n) - \mu_2\delta_i^y(n)\mathbf{s}_{i-1,i}(n-N) \end{aligned}$$

where the local error $\delta_i^y(n)$ is given by

$$\delta_i^y(n) = -\sum_{l=1}^{J}[\ e_l(n-N)\ e_l(n-N+1)\ \ldots\ e_l(n)\ ]\mathbf{c}_{i,l}$$

The $\mathbf{K}$'s are adapted according to

$$\begin{aligned} \mathbf{K}_{i,i+1}(n+1) &= \mathbf{K}_{i,i+1}(n) - \mu_3\boldsymbol{\delta}_{i,i+1}^s(n)\mathbf{u}_i^T(n-N) \\ \mathbf{K}_{i,i-1}(n+1) &= \mathbf{K}_{i,i-1}(n) - \mu_3\boldsymbol{\delta}_{i,i-1}^s(n)\mathbf{u}_i^T(n-N) \\ \mathbf{K}_{i-1,i+1}(n+1) &= \mathbf{K}_{i-1,i+1}(n) - \\ &\quad \mu_4\boldsymbol{\delta}_{i,i+1}^s(n)\mathbf{s}_{i-1,i}^T(n-N) \\ \mathbf{K}_{i+1,i-1}(n+1) &= \mathbf{K}_{i+1,i-1}(n) - \\ &\quad \mu_4\boldsymbol{\delta}_{i,i-1}^s(n)\mathbf{s}_{i+1,i}^T(n-N) \end{aligned}$$

where the local error vectors are given by

$$\begin{aligned} \boldsymbol{\delta}_{J-1,J}^s(n) &= \delta_J^y(n)\mathbf{w}_{J-1,J}(n) \\ \boldsymbol{\delta}_{i,i+1}^s(n) &= \delta_{i+1}^y(n)\mathbf{w}_{i,i+1}(n) + \mathbf{K}_{i,i+2}^T(n)\boldsymbol{\delta}_{i+1,i+2}^s(n) \end{aligned}$$

for $i = J-2$ down to 1, and

$$\begin{aligned} \boldsymbol{\delta}_{2,1}^s(n) &= \delta_1^y(n)\mathbf{w}_{2,1}(n) \\ \boldsymbol{\delta}_{i,i-1}^s(n) &= \delta_{i-1}^y(n)\mathbf{w}_{i,i-1}(n) + \mathbf{K}_{i,i-2}^T(n)\boldsymbol{\delta}_{i-1,i-2}^s(n) \end{aligned}$$

for $i=3$ to J. Note that the updates for the $\mathbf{w}$'s and $\mathbf{K}$'s have a familiar form: the new value is equal to the old value minus a step size parameter times the product of the local error and the input to the parameter being adapted. The N sample delay associated with the input term in each update is a consequence of causality requirements when back propagating the error through the N tap filters $\mathbf{c}_{k,l}$.

4. CONVERGENCE ANALYSIS

The shape of the MSE as as a function of the $\mathbf{w}$'s and $\mathbf{K}$'s offers information about the convergence behavior of the algorithm. Substitute for $y_k(n)$ in (8) to obtain

$$e_l(n) = d_l(n) - \sum_{k=1}^{J}\mathbf{q}_{i,k}^{l\ T}(n)\mathbf{h}_{i,k} \quad (10)$$

where $\mathbf{q}_{i,k}^{l\ T}(n) = [\ u_i(n)\ u_i(n-1)\ldots u_i(n-N)\ ]\mathbf{c}_{k,l}$ represents a vector containing values of the i^{th} input filtered by

$\mathbf{c}_{k,l}$. Now define vectors $\tilde{\mathbf{q}}_l(n) = [\ \mathbf{q}_{1,1}^{l\ T}\ \mathbf{q}_{2,1}^{l\ T} \dots \mathbf{q}_{J,J}^{l\ T}\ ]$ and $\tilde{\mathbf{h}} = vec\{\mathbf{H}\}$ so that (10) can be rewritten as

$$e_l(n) = d_l(n) - \tilde{\mathbf{q}}_l(n)\tilde{\mathbf{h}} \tag{11}$$

Hence, the MSE (9) can be expressed in the form

$$C = \sigma^2 - 2\mathbf{p}^T\tilde{\mathbf{h}} + \tilde{\mathbf{h}}^T\mathbf{R}\tilde{\mathbf{h}} \tag{12}$$

where $\sigma^2 = \sum_{l=1}^{J} E\{d_l^2(n)\}$, $\mathbf{p} = \sum_{l=1}^{J} E\{\tilde{\mathbf{q}}_l(n)d_l(n)\}$, and $\mathbf{R} = \sum_{l=1}^{J} E\{\tilde{\mathbf{q}}_l(n)\tilde{\mathbf{q}}_l^T(n)\}$. Here $\mathbf{p}$ represents the cross-correlation between the filtered inputs and desired signals, while $\mathbf{R}$ is the correlation matrix associated with the filtered inputs.

First, note that the MSE is a quadratic function of $\tilde{\mathbf{h}}$, but it is not quadratic in the $\mathbf{w}$'s and $\mathbf{K}$'s since (7)indicates that $\tilde{\mathbf{h}}$ is a function of their products. Second, the global minimum is unique in $\tilde{\mathbf{h}}$ if $\mathbf{R}$ is nonsingular, which implies the filtered inputs represented by $\tilde{\mathbf{q}}_l(n)$ are persistently exciting. However, the global minimum is not unique in the $\mathbf{w}$'s and $\mathbf{K}$'s, again because $\tilde{\mathbf{h}}$ depends on products of $\mathbf{w}$'s and $\mathbf{K}$'s. Many different choices for the $\mathbf{w}$'s and $\mathbf{K}$'s can result in the same $\tilde{\mathbf{h}}$.

Considerable information about the shape of the performance surface is obtained by examining the Hessian of the cost function. In particular, a positive definite Hessian implies a single global minimum, a semi-definite Hessian indicates the existence of a plateau, and a negative definite Hessian indicates the presence of a global maximum. If α and β are any two parameters (elements of the $\mathbf{w}$'s or $\mathbf{K}$'s), then the elements of the Hessian matrix are given by

$$\frac{\partial^2 C}{\partial\beta\partial\alpha} = \frac{\partial\tilde{\mathbf{h}}^T}{\partial\beta}\mathbf{R}\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha} + \left(\tilde{\mathbf{h}}^T\mathbf{R} + \mathbf{p}^T\right)\frac{\partial^2\tilde{\mathbf{h}}}{\partial\beta\partial\alpha} \tag{13}$$

While it can be shown that the first term is associated with a positive definite component of the Hessian, the second term may make the overall Hessian indefinite. However, this second term is zero under the following conditions:

- $\alpha = \beta$. This implies the diagonal elements of the Hessian are given by the quadratic form $\frac{\partial\tilde{\mathbf{h}}^T}{\partial\alpha}\mathbf{R}\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}$. This quantity is always positive since $\mathbf{R}$ is positive definite. Hence, the performance surface in any single parameter is quadratic with a single global minimum. Second, this implies the trace of the Hessian is always positive, so sum of eigenvalues is positive and the Hessian matrix cannot be negative definite. Thus, there are no local maxima. Saddle points exist when one or more eigenvalues are negative.
- When α and β are elements in a given $\mathbf{w}$ or $\mathbf{K}$ or elements in any $\mathbf{w}$ and $\mathbf{K}$ that do not appear in the same $\mathbf{h}_{i,j}$ (see (7)). In any subspace satisfying this condition, the performance surface is quadratic with a single global minimum. For example, $\mathbf{K}_{2,1}$ and $\mathbf{K}_{1,2}$ appear only in disjoint $\mathbf{h}_{i,j}$ and the performance surface is quadratic in the subspace defined by $\mathbf{K}_{2,1}$ and $\mathbf{K}_{1,2}$. Thus, the performance surface may be non-quadratic only in subspaces defined by parameters that interact in a single $\mathbf{h}_{i,j}$, such as $\mathbf{K}_{4,3}$ and $\mathbf{K}_{4,2}$, which both appear in $\mathbf{h}_{1,4}$.

We now turn our attention to selection of the step size parameters. Consider adapting a single parameter α while holding all other parameters fixed. The cost function is expressed in terms of α as $\alpha^2 r - 2p_o\alpha + \sigma_o^2$ where $r = \frac{1}{2}\frac{\partial^2 C}{\partial\alpha^2}$. Analysis of the standard LMS algorithm for adapting α indicates that the step size μ must satisfy $\mu < \frac{2}{r}$ to insure convergence, so the maximum step size decreases as r increases. An upper bound on r is obtained from (13)by noting

$$\begin{aligned}\frac{\partial^2 C}{\partial\alpha^2} &\le \left|\left|\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}\right|\right|_2^2 ||\mathbf{R}||_2 \\ &\le \left|\left|\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}\right|\right|_2^2 \operatorname{tr}\mathbf{R}\end{aligned} \tag{14}$$

However, $\left|\left|\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}\right|\right|_2^2$ depends on the values of the parameters that are fixed. As their magnitudes increase, $\left|\left|\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}\right|\right|_2^2$ generally increases without bound. This suggests that we place upper bounds on all parameters in order to insure stability with a fixed step size. Upper bounding all parameters also prevents excessive drift along any trajectory associated with minimum mean squared error.

An upper bound on the parameters must be chosen so that the region in which adaptation is permitted contains at least one minimum of the performance surface. To do so we assume that the maximum gain from input to output at any frequency is upper bounded by a constant c at the optimum. That is, the optimum filters satisfy $\max_\Omega |H_{i,j}(e^{j\Omega})| \le c$. Let K_1 and K_2 be the upper bound on the absolute value of any element of $\mathbf{K}$'s with subscripts offset by one and two, respectively, W_1 be the upper bound on the absolute value of any element of $\mathbf{w}_{j,j-1}$ or $\mathbf{w}_{j,j+1}$ and W_0 be the upper bound on the absolute value of any element in $\mathbf{w}_{j,j}$. By applying various matrix norm inequalities to (7), we may show that the condition $|H_{i,j}(e^{j\Omega})| \le c$ implies

$$W_0 \le \frac{c}{M} \tag{15}$$

$$W_1K_1 \le \frac{c}{PM} \quad \text{for } |i-j| = 1 \tag{16}$$

$$W_1K_1(PK_2)^{|i-j|-1} \le \frac{c}{PM} \quad \text{for } |i-j| \ge 2 \tag{17}$$

These conditions are satisfied with equality independent of i and j by choosing $W_0 = \frac{c}{M}$, $W_1 = K_1 = \sqrt{\frac{c}{PM}}$, and $K_2 = \frac{1}{P}$. Hence, we limit adaptation to the interior of a cube defined by the maximum gain, c, number of filter taps, M, and dimension of the data vector communicated between nodes, P.

Bounds on the step size may be determined from the upper bounds on the parameter values. Again applying matrix norm inequalities, we obtain upper bounds on $\left|\left|\frac{\partial\tilde{\mathbf{h}}}{\partial\alpha}\right|\right|_2^2$ and the corresponding step size bounds. Summarizing, the maximum step size for $\mathbf{w}_{i,i}$ is $\mu_1 < \frac{4}{\operatorname{tr}\mathbf{R}}$. The maximum step size for $\mathbf{w}_{i+1,i}$ and $\mathbf{w}_{i-1,i}$ takes the form $\frac{4}{c\eta(i)\operatorname{tr}\mathbf{R}}$, that for $\mathbf{K}_{i,i+1}$ and $\mathbf{K}_{i,i-1}$ takes the form $\frac{4M}{c\eta(i)\operatorname{tr}\mathbf{R}}$, and that for $\mathbf{K}_{i-1,i+1}$ and $\mathbf{K}_{i+1,i-1}$ takes the form $\frac{4M}{c^2\eta(i)\operatorname{tr}\mathbf{R}}$. In each case, $\eta(i)$ counts the number of $\mathbf{h}_{i,j}$ in which the parameter occurs. For example, $\mathbf{w}_{i,i-1}$ occurs in $J-i+1$ different $\mathbf{h}_{i,j}$ so $\eta(i) = J - i + 1$, while $\mathbf{K}_{i+1,i-1}$ occurs in $(J-i)(i-1)$ different $\mathbf{h}_{i,j}$ so in this case $\eta(i) = (J-i)(i-1)$. The dependence of $\eta(i)$ on i implies that the maximum step size is also a function of the node index.

The term $\mathrm{tr}\,\mathbf{R}$ may be upper bounded in terms of easily measured quantities by applying standard matrix norm inequalities. We obtain

$$\mathrm{tr}\,\mathbf{R} \leq NM \left(\sum_{i,j=1}^{J} \mathbf{c}_{i,j}^T \mathbf{c}_{i,j} \right) \sum_{k=1}^{L} \sigma_k^2 \tag{18}$$

where σ_k^2 is the power in the k^{th} input channel.

Several qualifications are in order regarding the step size bounds. First, the step size analysis did not consider the effect of the delay by N samples in the update term associated with backpropagation through the $\mathbf{c}_{i,j}$. Delays in the update have a destabilizing influence [6], and thus the upper bounds will need to be reduced to insure stability when $N \geq 1$. Second, the bounds were obtained by considering the worst case stability conditions for a single parameter. While this does not gaurantee stability in all directions of the parameter space, we have not yet encountered an example in which instability occured using these bounds.

Lastly, note that we cannot use initial conditions of all zeros since then $\mathbf{s}_{i-1,i}(n)$ and $\mathbf{s}_{i+1,i}(n)$ are zero and the $\mathbf{K}$'s and $\mathbf{w}$'s (except for $\mathbf{w}_{i,i}$) never adapt to a nonzero value. All zeros represents a saddle point in the performance surface. Thus, the gradient is very small in the vicinity of the origin, and adaptation is slow. This suggests that fastest convergence is obtained by choosing initial conditions away from the origin.

5. EXAMPLE

The adaptive algorithm is demonstrated using a $J = 5$ node system based on $M = 5$ tap filters and $P = 5$ data values communicated between nodes. We chose $N = 0$ in order to eliminate the destabilizing influence of delays in the updates associated with backpropagation through the $\mathbf{c}_{i,j}$. Hence, each $\mathbf{c}_{i,j}$ is a scalar and was chosen as a random number uniformly distributed between zero and one. The input to each channel is independent white noise of unit variance. The desired signals $d_l(n)$ were generated by passing the input through a 5 input, 5 output system composed of five tap FIR filters. Hence, in this case the MSE is driven to zero if the adaptive algorithm converges to an optimal solution. The maximum gain at any frequency, c, was set at one.

Figure 2 depicts the MSE as a function of the number of adaptive algorithm iterations for thirty trials with different initial conditions. The step sizes for each parameter were set at the upper bound presented in the previous section. The initial conditions for each parameter were chosen as the maximum permissible values with the sign randomized, that is, the values $\pm W_0, \pm W_1, \pm K_1, \pm K_2$, respectively. The results indicate that the algorithm converges consistently, independent of the initial conditions.

6. REFERENCES

[1] S.J. Elliot and P.A. Nelson, "Active Noise Control," *IEEE Signal Processing Magazine*, vol. 10, pp.12-35, October 1993.

[2] S.R. Popovich, "Multi-Channel Active Attenuation System with Error Signal Inputs," US Patent 5,216,722.

[3] B.D. Van Veen, O.E. Leblond, and D.J. Sebald,"Adaptive Acoustic Attenuation System Having Distributed Processing and Shared State Nodal Architecture," US Patent (applied).

[4] S. Haykin, *Neural Networks*, Macmillan, 1994.

[5] E.A. Wan, "Time Series Prediction by Using a Connectionist Network with Internal Delay Lines," in *Time Series Prediction: Forecasting the Future and Understanding the Past,* Eds. A.S. Weigend and N.A. Gershenfeld, Addison Wesley, 1993.

[6] D.R. Morgan and C. Sanford, "A Control Theory Approach to the Stability and Transient Analysis of The Filtered-X LMS Adaptive Notch Filter," *IEEE Trans. Signal Proc.,* vol. 40, pp. 2341-2346, 1992.

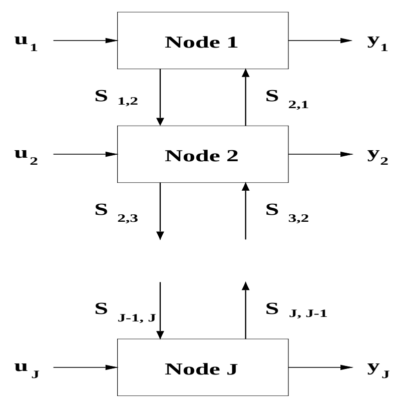

Figure 1: Modular MIMO Adaptive Filter.

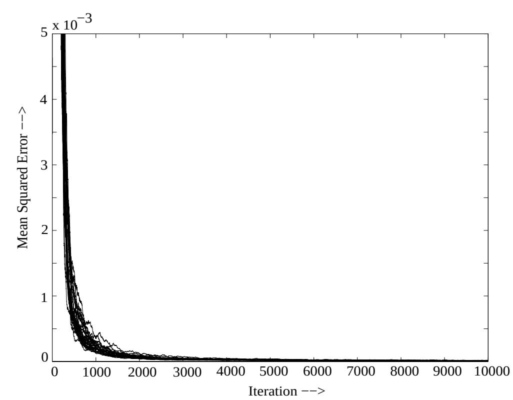

Figure 2: Convergence of the Adaptive Algorithm for Different Initial Conditions.