AFFINE PROJECTION METHODS IN FAULT TOLERANT ADAPTIVE FILTERING*

Robert A. Soni[1], *Kyle A. Gallivan*[2], *and W. Kenneth Jenkins*[3]

Coordinated Science Laboratory and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1308 W. Main St.
Urbana, IL 61801, U.S.A.
[1]rsoni@uiuc.edu, [2]gallivan@uiuc.edu, and [3]jenkins@uicsl.uiuc.edu

ABSTRACT

Reliable performance is very important for high speed channel equalizers and echo cancellers used in high speed communications channels. A common type of hardware fault occurs when the coefficients get "stuck" at an uncontrollable value. Such faults lead to larger overall mean square errors, and generally poor performance. Redundancy can provide the ability to compensate for these types of faults if the proper design is introduced into the adaptive filter structure. Unfortunately, this form of redundancy can lead to poor convergence performance for the adaptive filter after the fault occurrence. This paper examines the use of affine projection and *row projection* techniques to improve the convergence performance of the fault tolerant adaptive filtering structure. Algorithms are developed for two cases: fault knowledge and no fault knowledge incorporated in the adaptive filtering update. These algorithms are introduced in this paper and simulations are presented to illustrate the effectiveness of these approaches.

1. INTRODUCTION

Increasing concerns over the reliability of fast adaptive algorithms and their implementations has spawned interest and research in the development of algorithms which are capable of adapting to optimal or "nearly" optimal solutions in the presence of faults. Adaptive filter coefficients are made robust to faults by introducing redundant coefficients. In order to compensate for a single fault, a set of coefficients may be made fault-tolerant by adding a single additional adaptive filter coefficient. To compensate for up to R faults, R additional coefficients are added to the set of N coefficients.

However, the inputs to the R coefficients are different from the other N. They are typically composed of the linear combination of the N delayed input samples. In order to illustrate some of the possibilities, the output of the adaptive filter for the N-th order case with no faults can be expanded:

$$y(n) = W^H(n) \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} X(n) \qquad (1)$$

where $W(n)$ is a vector of adaptive filter coefficients and $X(n)$ is a vector of the input data. To incorporate protection for one fault, the following modified filtering structure may be used:

$$y(n) = W^H(n) \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ 1 & \cdots & 1 \end{bmatrix} X(n) \qquad (2)$$

where the vector of adaptive filter coefficients now has $N+1$ coefficients [3]. This approach may be generalized further by writing the output of the adaptive filter as $y(n) = W^H(n)\mathbf{U}^H X(n)$. The matrix $\mathbf{U}$ insures fault tolerance to R faults if all sets of any of the N columns of the $M \times N$ dimensional matrix $\mathbf{U}$ are linearly independent where $M = N + R$ [3]. This set may be constructed by taking the first N columns of the M dimensional DFT or DCT matrices. Alternatively, this can also be viewed as zero-padding the N dimensional input vector to M values, and taking the DFT or DCT, i.e. by taking the DFT of the vector $X_e(n)$ where $X_e(n)$ is given by:

$$X_e(n) = \left[X^T(n), 0, \ldots, 0\right]^T \qquad (3)$$

*This work was supported by the Joint Services Electronics Program (JSEP) under contract number N00014-96-1-0129. The opinions here do not necessarily represent those of the sponsoring agencies.

Note that many other constructions are possible for the matrix $\mathbf{U}$ [4].

2. RAPIDLY CONVERGING ADAPTIVE FAULT TOLERANT ALGORITHMS

Post fault convergence rates for LMS-based strategies for updating the filter coefficients will generally be poor for highly correlated inputs. One possible solution to this problem is to utilize a vector version of the normalized LMS algorithm to update the fault tolerant adaptive filtering structure. This method is related to the class of algorithms popularized as the "affine" projection methods [2],[5]. The authors [6],[7] have also developed algorithms which fall into this class by generalizing a classical form of optimization known as *row projection*. In general, each of these projection-based adaptive strategies can be posed as solving the following minimization problem:

$$\begin{array}{cc} \min\limits_{W(n)} & ||W(n) - W(n-1)||_2^2 \\ \textit{subject to} & D(n) = \mathbf{X}^H(n)\mathbf{U}W(n) \end{array} \quad (4)$$

where $D(n) = \mathbf{X}^H(n)\mathbf{U}W(n)$ represents the L^{th} order vector of error conditions, $E(n) = [e(n), \ldots, e(n-L+1)]$, set to zero. The vector $D(n)$ represents a vector of desired inputs, and the matrix $\mathbf{X}(n)$ represents a $N \times L$ matrix of the input signal. This problem can be solved using a Lagrange multiplier method of optimization. An update expression for the adaptive filter coefficients may be developed as:

$$\begin{aligned} W(n) &= W(n-1) - \mathbf{U}^H\mathbf{X}(n)[\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ &\quad \times[D(n) - \mathbf{X}^H(n)\mathbf{U}W(n-1)]. \end{aligned} \quad (5)$$

2.1. Fault Tolerant Adaptive Filtering Algorithms (Faults Known) - Implementation I

The algorithm described by equation 5 does not rely on knowledge of the exact fault locations. Conceivably, there might be a fault detection mechanism designed into the system [4]. Assuming that the overall goal of the algorithm is to compensate for faults occuring in up to R taps, the performance criterion of the vectorized NLMS algorithm may be re-stated to include the effects of the R faults.

$$\begin{array}{cc} \min\limits_{W(n)} & ||W(n) - W(n-1)||_2^2 \\ \textit{subject to} & D(n) = \mathbf{X}^H(n)\mathbf{U}W(n) \\ & \mathbf{E}^H W(n) = C \end{array} \quad (6)$$

where the matrix $\mathbf{E}$ is given by the following:

$$\mathbf{E} = [e_{k_1}, \ldots, e_{k_R}] \quad (7)$$

and where e_i is a vector with a one in the i'th position and zeros elsewhere, and $\{k_1, \ldots k_R\}$ is an indexed set of integers belonging to $[0, M-1]$ which defines the location of the R failed taps.

If the faults can be identified, the vectorized NLMS or the accelerated *row projection* methods developed by the authors in [6], [7] may be augmented directly to solve this alternative system. The vector NLMS with fault knowledge replaces $D(n)$ and $\mathbf{X}(n)$ with $\tilde{D}(n)$ and $\tilde{\mathbf{X}}(n)$, respectively where

$$\begin{aligned} \tilde{D}(n) &= \begin{bmatrix} D(n) \\ C \end{bmatrix} \\ \tilde{\mathbf{X}}^H(n) &= \begin{bmatrix} \mathbf{X}^H(n)\mathbf{U} \\ \mathbf{E}^H \end{bmatrix} \end{aligned} \quad (8)$$

The update of the adaptive coefficients in the vector NLMS case using fault knowledge becomes:

$$\begin{aligned} W(n) &= W(n-1) - \begin{bmatrix} \mathbf{U}^H\mathbf{X}(n) & \mathbf{E} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n) & \mathbf{X}^H\mathbf{U}\mathbf{E} \\ \mathbf{E}^H\mathbf{U}^H\mathbf{X} & \mathbf{E}^H\mathbf{E} \end{bmatrix}^{-1} \\ &\quad \times \begin{bmatrix} D(n) - \mathbf{X}_e^H(n)\mathbf{U}W(n-1)] \end{bmatrix} \end{aligned} \quad (9)$$

While this direct implementation is easy enough to visualize, it can be computationally expensive to implement. As an alternative, the conjugate gradient (CG) algorithm can be used to solve the system $\mathbf{A}X = B$ where $\mathbf{A} = \tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^H(n)$, and $B = \tilde{\mathbf{X}}(n)\tilde{D}(n)$ similar to the approach of Boray and Srinath [1]. A different implementation that directly incorporates the faults as constraining conditions may be also developed. This approach which solves the optimization problem stated in equation 6 is developed in the next section.

Row projection methods may be used to express the adaptive filtering process as the solution of a constrained solution of a linear system. The linear system associated with the adaptive filtering process is the set of zero error conditions, $\mathbf{X}^H(n)W(n) = D(n)$, and the fault knowledge, $\mathbf{E}^H W(n) = C$. The authors have prescribed a number of methods to solve these types of linear systems using data reusing, and accelerated data reusing adaptive techniques [6],[7].

2.2. Fault Tolerant Adaptive Filtering Algorithms (Faults Known) - Implementation II

To formulate a solution for the optimization problem described in equation 6, the Lagrange multiplier method can be used. The following function can then be minimized:

$$\begin{aligned} J(W(n), \lambda_1, \lambda_2) &= ||W(n) - W(n-1)]||_2^2 + \\ &\quad \lambda_1^H[D(n) - \mathbf{X}^H(n)\mathbf{U}W(n)] \\ &\quad +\lambda_2^H[\mathbf{E}^H W(n) - C] \end{aligned} \quad (10)$$

Following the general prescription of Lagrange multiplier optimization, the derivatives of equation 10 are taken with respect to $W(n)$, λ_1, and λ_2 and set equal to zero.

$$\frac{\delta J(W(n),\lambda_1,\lambda_2)}{\delta W(n)} = W(n) - W(n-1) + \mathbf{U}^H\mathbf{X}(n)\lambda_1 + \mathbf{E}\lambda_2 \quad (11)$$

$$\frac{\delta J(W(n),\lambda_1,\lambda_2)}{\delta \lambda_1} = D(n) - \mathbf{X}^H(n)\mathbf{U}W(n) \quad (12)$$

$$\frac{\delta J(W(n),\lambda_1,\lambda_2)}{\delta \lambda_2} = \mathbf{E}^H W(n) - C \quad (13)$$

Substituting equation 11 into equation 12 and equation 13 yields the following two equations:

$$\begin{aligned} D(n) &= \mathbf{X}^H(n)\mathbf{U}[W(n-1) - \mathbf{U}^H\mathbf{X}(n)\lambda_1 - \mathbf{E}\lambda_2] \\ C &= \mathbf{E}^H[W(n-1) - \mathbf{U}^H\mathbf{X}(n)\lambda_1 - \mathbf{E}\lambda_2] \quad (14) \end{aligned}$$

Using the knowledge that the fault has occurred at the previous instant, $C - \mathbf{E}^H W(n-1) = 0$, and $\mathbf{E}^H\mathbf{E} = \mathbf{I}$, expressions for the optimal values of the Lagrange multipliers, λ_1^* and λ_2^* may be obtained as the following:

$$\begin{aligned} \lambda_1^* = & -[\mathbf{I} - [\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & \times\mathbf{X}^H(n)\mathbf{U}\mathbf{E}\mathbf{E}^H\mathbf{U}^H\mathbf{X}(n)]^{-1}[\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & \times[D(n) - \mathbf{X}^H(n)\mathbf{U}W(n-1)] \quad (15) \end{aligned}$$

$$\begin{aligned} \lambda_2^* = & \mathbf{E}^H\mathbf{U}^H\mathbf{X}(n)[\mathbf{I} - [\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & \times\mathbf{X}^H(n)\mathbf{U}\mathbf{E}\mathbf{E}^H\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & \times[\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & [D(n) - \mathbf{X}^H(n)\mathbf{U}W(n-1)] \quad (16) \end{aligned}$$

Using λ_1^* and λ_2^*, the optimal update expression for $W(n)$ becomes:

$$\begin{aligned} W(n) = & W(n-1) + [\mathbf{I} - \mathbf{E}\mathbf{E}^H]\mathbf{U}^H\mathbf{X}(n)[\mathbf{I} - \\ & [\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1}\mathbf{X}^H(n)\mathbf{U}\mathbf{E} \\ & \times\mathbf{E}^H\mathbf{U}^H\mathbf{X}(n)]^{-1}[\mathbf{X}^H(n)\mathbf{U}\mathbf{U}^H\mathbf{X}(n)]^{-1} \\ & \times[D(n) - \mathbf{X}^H(n)\mathbf{U}W(n-1)] \quad (17) \end{aligned}$$

If $\mathbf{U}^H$ is chosen as the first N columns of the M dimensional *unitary* DFT, and the data vectors are zero-padded to enforce this choice, the update expression becomes:

$$\begin{aligned} W(n) = & W(n-1) + [\mathbf{I} - \mathbf{E}\mathbf{E}^H]\mathbf{U}^H\mathbf{X}_e(n)[\mathbf{I} - \\ & [\mathbf{X}^H(n)\mathbf{X}(n)]^{-1}\mathbf{X}_e^H(n)\mathbf{U}\mathbf{E}\mathbf{E}^H\mathbf{U}^H\mathbf{X}_e(n)]^{-1} \\ & \times[\mathbf{X}^H(n)\mathbf{X}(n)]^{-1}[D(n) - \mathbf{X}_e^H(n)\mathbf{U}W(n-1)] \quad (18) \end{aligned}$$

Further defining the projection matrices, $\mathbf{P}_{R(\mathbf{E})} = \mathbf{E}\mathbf{E}^H$, and $\mathbf{P}_{N(\mathbf{E}^H)} = [\mathbf{I} - \mathbf{E}\mathbf{E}^H]$, results in the following update expression:

$$\begin{aligned} W(n) = & W(n-1) + \mathbf{P}_{N(\mathbf{E}^H)}\mathbf{U}^H\mathbf{X}_e(n)[\mathbf{I} - \\ & [\mathbf{X}^H(n)\mathbf{X}(n)]^{-1}\mathbf{X}_e^H(n)\mathbf{U}\mathbf{P}_{R(\mathbf{E})}\mathbf{U}^H\mathbf{X}_e(n)]^{-1} \\ & \times[\mathbf{X}^H(n)\mathbf{X}(n)]^{-1}[D(n) - \mathbf{X}_e^H(n)\mathbf{U}W(n-1)] \quad (19) \end{aligned}$$

The update expression now appears similar to the update expression for the algorithm with no fault knowledge. The update in the coefficients is projected away from the span of the faulty coefficients. Potentially, this method might offer some benefit in terms of reduced computational complexity over the method in equation 9, but it may also require a more complicated implementation. While this expression is simplified, the resulting implementation can still be difficult.

Potentially, the *row projection* methods developed by the authors or the CG algorithm of Boray and Srinath [1] offers the simplest level of understanding to the system designer attempting to realize a fault tolerant system.

3. EXPERIMENTAL EVALUATION

In order to determine if knowledge of the faults made any difference in the performance of the adaptive filter convergence rate, the performance of the accelerated *row projection* method developed in [6] in the presence and absence of fault knowledge is examined. When the fault is known, the accelerated data reusing algorithms solve the optimization problem stated in equation 6. When the faults are unknown, the accelerated data reusing algorithms solve the optimization problem associated with equation 4. This is equivalent to holding the coefficients constant during the update.

A simple set of system identification experiments were performed where stuck at value faults occurred. The unknown system had eight taps and R was set to 2. This system could in theory compensate for up to two faults. The results of accelerated *row projection* algorithm where faults were both unknown and known for white noise are displayed in Figures 1 and 2.

The data from these plots does not indicate a significant disparity for the case of white noise input. But, the knowledge of faults does slightly improve the post-fault convergence rate after the second fault. However, it appears to lead to a slightly slower convergence rate after the first fault.

The same experiment was repeated for the case where the input signal was colored noise. The resulting mean square error plots in Figures 3 and 4 illustrate the large disparity in the system that occurs when the knowledge of the faults is completely unknown. For this simple case, the post-fault convergence after the second fault illustrates that knowledge of the fault can

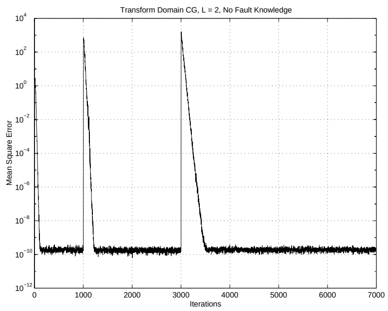

Figure 1: System Identification, White Noise Input, No Fault Knowledge

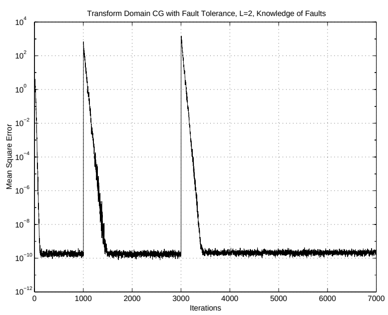

Figure 2: System Identification, White Noise Input, Fault Knowledge

improve convergence behavior over the system where no fault knowledge is assumed.

4. CONCLUDING REMARKS

The adaptive fault tolerant algorithms developed in this paper can compensate for the presence of up to R faults. It has been shown previously that fault tolerant algorithms can have their performance significantly degraded in the presence of colored or highly correlated inputs. New algorithms were introduced which

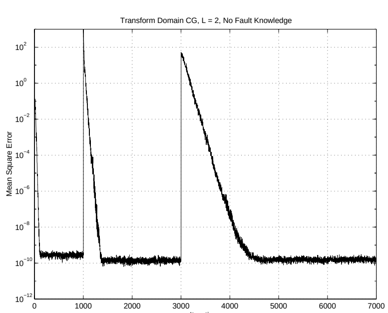

Figure 3: System Identification, Colored Noise Input, No Fault Knowledge

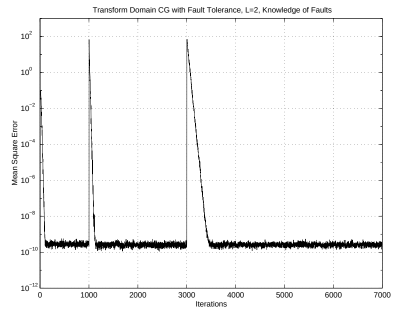

Figure 4: System Identification, Colored Noise Input, Fault Knowledge

featured improved performance over earlier fault tolerant algorithms. In particular, this paper specifically improved the performance of the quality of post-fault convergence rates with or without fault knowledge.

5. REFERENCES

[1] G. Boray and M. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Trans. Circuits Systems*, vol. 39, no. 1, pp. 1-10, Jan. 1992.

[2] S. Gay and S. Tavatia, "The fast affine projection algorithm," *Proc. ICASSP*, pp. 3023-3026, 1995.

[3] W. Jenkins, A. Hull, J. Strait, B. Schnaufer, and X. Li, *Advanced Concepts in Adaptive Signal Processing*, Kluwer Academic Publishers, Norwell, MA, 1996.

[4] B. Schnaufer and W. Jenkins, "Adaptive fault tolerance for reliable LMS adaptive filtering," *IEEE Trans. Circuits and Systems, Part II: Analog and Digital Signal Proc.*, vol. 44, no. 12, pp. 1001-1014, December 1997.

[5] D. Slock, "Underdetermined growing and sliding window covariance fast transversal filter RLS algorithms," *Proc. EUSIPCO '92*, Brussels, Belgium, pp. 1169-1172, Aug. 1992.

[6] R. Soni, K. Gallivan, and W. Jenkins, "Projection methods for improved performance in FIR adaptive filtering," *Proc. Midwest Symp. on Circuits and Systems*, Sacramento CA. Aug. 1997.

[7] R. Soni, K. Gallivan, and W. Jenkins, "Acceleration of Normalized Data Reusing Methods using the Tchebyshev and Conjugate Gradient Methods," *Proc. ISCAS, '98*, Monterrrey, CA, May 1998.