A Fault Tolerant Pipelined Adaptive Filter

Bhasker Allam, Martin Meyer, and James Leathrum

Electrical and Computer Engineering Department
Old Dominion University
Norfolk, VA 23529-0246

Abstract

A fault tolerant pipelined architecture for high sampling rate adaptive filters is presented. The architecture, which is based on the computational requirements of delayed LMS and adaptive lattice filters, offers robust peformance in the presence of single hardware faults, and software faults resulting from numerical instability. The reliability of the proposed system is analyzed and compared to the existing implementations strategies, and methods for fault detection, fault location, and recovery via hardware reconfiguration are discussed. Finally, simulation results illustrating recovery from processor fault are presented.

1 Introduction

The realization of transversal adaptive filters in real time is important in a variety of applications. Perhaps one of the most well known adaptive algorithms is the least mean squares (LMS) algorithm, which updates the weights of a transversal filter using an approximate technique of steepest descent [1]. Even though the LMS algorithm is relatively simple, in situations where high sampling rates or large numbers of filter taps are required, the computational demand placed on a single processor system becomes prohibitive. Therefore, recent attention has been given to implementations which introduce adaptation delay and use multiple processing elements and pipelining, to achieve the required processing speed [2]-[4]. Another important class of adaptive filters is the adaptive lattice. Due to their order recursive structure, both the gradient and the least squares lattice filters are natural candidates for pipelining [5, 6]. The above referenced implementations share a common structure shown in figure 1.

However, previously proposed linear processor arrays for delayed LMS and lattice systems lack any redundant connectivity, and are therefore intolerant

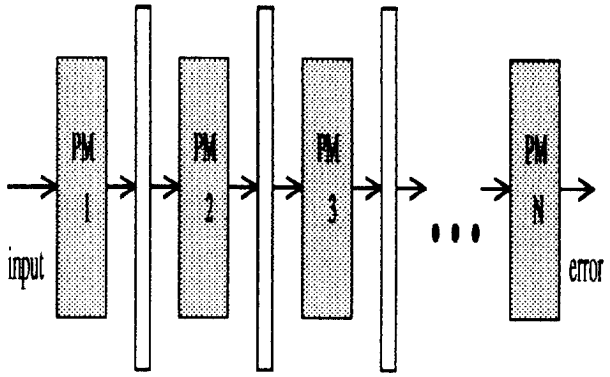

Figure 1: Pipelined adaptive filter structure (DLMS or Lattice)

of faults. In fact, as additional processors are added, the reliability of these systems decreases exponentially. Also, delay introduced to facilitate pipelining can increase the adaptive algorithms' sensitivity to roundoff error leading to problems with divergence when finite word lengths are used. This is pariticularly troublesome in the least sqaures lattice filter which suffers from known instability.

In this paper, a flexible fault tolerant pipelined architecture for adaptive filters is presented. The proposed array architecture is suitable for implementation of a variety of different adaptive algorithms, including delayed LMS, least squares lattice, and gradient lattice filters. Fault tolerance is acheived by introducing redundant links between processing modules, and employing a fault detection scheme which exploits the natural fault tolerance inherent in the adaptive filter.

1058-6393/95 $4.00 © 1995 IEEE

The system is shown to be robust in the presence of hardware faults, software faults due to accumulation of roundoff error, and a limited class of multiple hardware faults. Methods for fault detection and location are presented, equations for reliability are derived, and simulations results are presented which verify system performance in the presence of faults.

2 Fault Tolerant Pipeline

Before addressing the issues of fault location, fault detection, and hardware reconfiguration, certain assumptions must be made. These assumptions are consistent with those made in the literature [7]-[9].

processors in the system. The particular block affected may vary, but the result is always a divergence of the filters error signal. This provides the mechanism for fault detection. Fault location is a harder problem, and one which has recieved some attention outside the context of adaptive filtering [7]-[9]. The method proposed here for fault location is specific to the pipelined filter structure, and utilizes a processor self test sequence. Since a faulty processor cannot be trusted to evaluate its own intergrity, test results are passed to the right and are evaluated by the subsequent processors. In the diagram of figure 2, t_i is the evaluation of test results for PM_{i-1} which have been passed to PM_i. PM_i outputs $t_i = 1$ if it determines PM_{i-1} has failed its self test, and $t_i = 0$ if PM_i has completed its test succesfully. Finally, switching

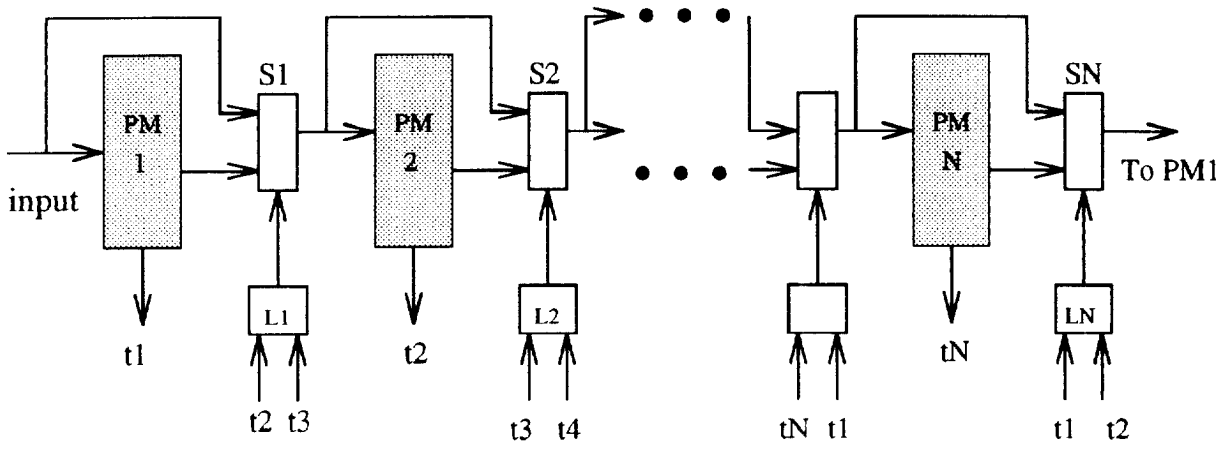

Figure 2: Fault tolerent pipelined adaptive filter.

assumption 1. No two faults can occur concurrently.

assumption 2. The switching logic is simple in comparison to the processing modules, and hence reliable in comparison.

assumption 3. Each processing module has the ability to perform a predetermined self-test sequence, but cannot be trusted to interpret the results.

A faulty processing module in a pipelined adaptive filter will cause a block of weights to take on unpredictable values. The number of weights affected depends on the order of the filter and the number of

hardware is included to dynamically reconfigure the pipeline. A block diagram of the system is shown in the figure 2.

The algorithm for fault detection, fault location, and reconfiguration is summarized below:

(1) Proceed with filter computations of error and weight updates.

(2) Monitor the squared error $e^2(n)$. If $e^2(n)$ exceeds a predetermined threshold for more than M consecutive samples periods, then a fault has occurred (either hardware or software). Proceed to step (3).

(3) Processors enter test mode, and pass test results to the right.

(4) Switch out the faulty processor by setting the input to each switch S_i to be $S_i = (t_i)(t_{i+1})'$.

(5) Return to step 1.

In this way, no faulty processor may erroneaoously switch out a good one. That is, if processor 1 is faulty, then processor 3 must determine that processor 2 is fault-free before processor 2 is able to switch out processor 1. If no processor is determined to be faulty, the end result is a system restart. In this way, the divergence caused by numerical instability is handled within the proposed framework. The sampling rate is not reduced due to reconfiguration, however in the case of a precossor fault, misadjustment will increase slightly due to a reduction in the number of filter taps being implemented. More about this is discussed in the next section.

3 Performance and Simulation Results

A common measure of the fault tolerance of a system is reliability, which is defined as (1 - *probability of system failure*). The reliability of the N-element pipelined array shown in figure 1 is given by

$$R_{pipe} = R^N \tag{1}$$

where R is the reliability of each individual processing module. The system shown in figure 2 is tolerant of single faults, and will therefore operate reliably at a given sampling rate in the presensce of either a zero or one bad module. The overall reliabilty of the new system is then given by

$$R_{ftpipe} = R^N + N(1-R)R^{N-1} \tag{2}$$

This improvement is illustrated in figure 3, which shows the reliability of a pipelined system as a function of the pipeline length, where a value of $R = 0.98$ has been assumed. It should be noted that this is actually a conservative estimate of the actual reliability, since certain types of multiple faults can in fact be tolerated, as long as no two faults occur within the same sample period, and no two faults occur at consecutive processing modules. However, the extra component of reliability added by these cases is negligble for all but very large arrays, and is therefore ignored here.

To illustrate the reconfiuration of a representative system, the delayed LMS algorithm was selected for

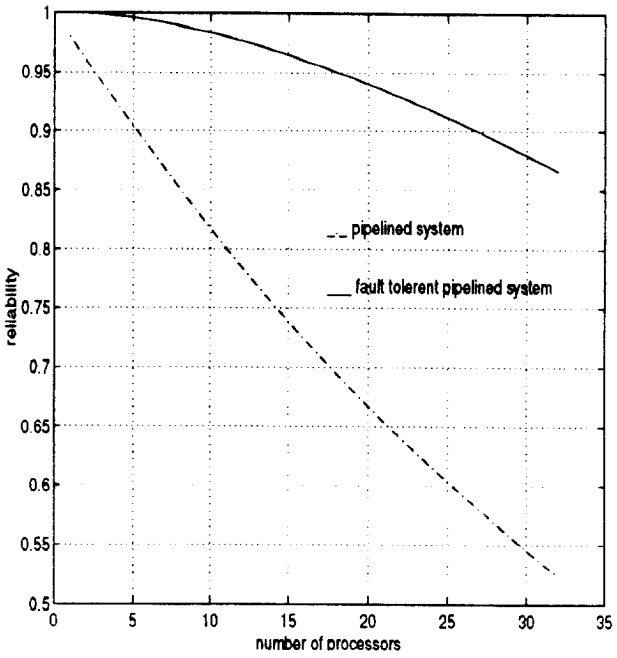

Figure 3: Reliability curves for piplined systems with and without fault tolereance

simulation. A 16-tap filter was configured in system identification mode and mapped onto an array of four processing modules. A colored input sequence is applied to the system, and additive noise is included to give SNR of 20dB. Each processor is responsible for computing weight updates and partial product updates for a block of four filter weights. The scalar updates to be performed for each weight by processor i at sample time n are given by

$$w_i(n-i) = w_i(n-i-1)+\beta e(n-N+1-i)x(n-N+1-2i) \tag{3}$$

$$y_i(n-i) = y_{n-1}(n-i) + x(n-2i)w_i(n-i-1) \tag{4}$$

where x is the input, e is the error signal, the w are the weights, and β is a constant gain. A detailed derivation of these is presented in [2]. Figure 4 shows a simulation of this system in response to a hardware fault. The region of initial convergence continues for 400 samples, until a fault occurs in PM2 at sample 400, causing weights 5, 6, 7, and 8 to take on random values resulting in error divergence. By sample 500, the system has reconfigured by switching PM2 out, and the resulting 12-tap filter reconverges. The results represent an ensemble average of 50 independent trials.

Because the reconfigured filter contains fewer weights than the original filter, the steady state mis-

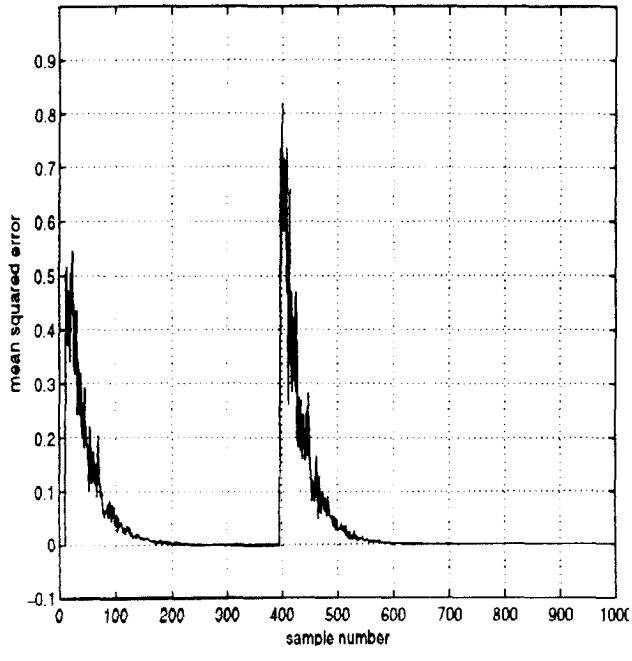

Figure 4: Simulation of a hardware fault in *PM2* and subsequent recovery

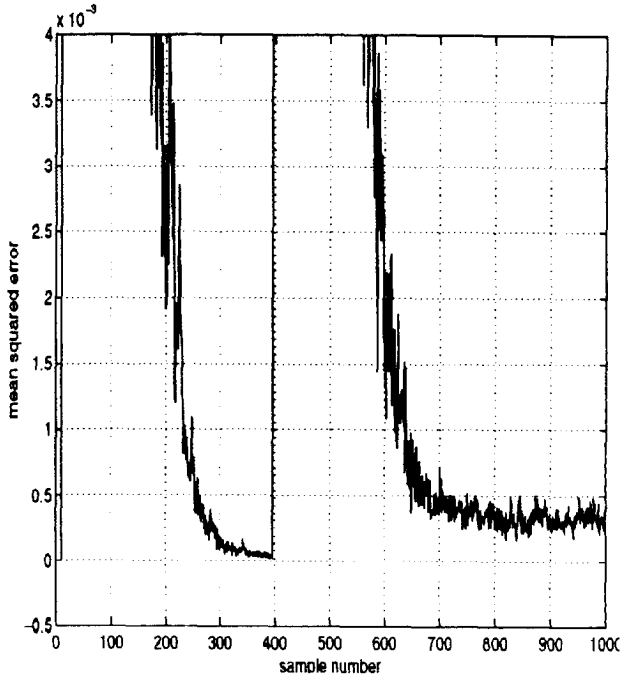

Figure 5: Illustraion of increase in steady state error with the decrease filter order

adjustment will increase. This behavior is illustrated in figure 5, which displays the same simulation shown in figure 4, where the axes have been scaled to focus on the region of convergence. In order to anaytically describe this behavior, an analysis of the steady state error with respect to the order of the filter is required.

The mimimum mean squared error achievable for an LMS adaptive filter is given by

$$\mathbf{J}_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_0 \tag{5}$$

$$\mathbf{J}_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \tag{6}$$

Using a unitary similarity transformation we can write

$$\mathbf{Q}^H \mathbf{R} \mathbf{Q} = \wedge \tag{7}$$

where $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_N]$, $\mathbf{q}_i$ is the eigen vector associated with eigenvalues λ_i of the autocorrelation matrix, and $\wedge = diag(\lambda_1, ..., \lambda_N)$. Equation (7) can also be written as,

$$\mathbf{R} = \mathbf{Q} \wedge \mathbf{Q}^{\mathbf{H}} \tag{8}$$

$$= \sum_{i=1}^{N} \lambda_i \mathbf{q}_i \mathbf{q}_i^H \tag{9}$$

Applying the property of autocorrelation matrix

$$\mathbf{R}\mathbf{q} = \lambda \mathbf{q} \tag{10}$$

to equation (9) we can easily show that,

$$\mathbf{R}^{-1} = \sum_{i=1}^{N} \lambda_i^{-1} \mathbf{q}_i \mathbf{q}_i^H \tag{11}$$

Substituting (11) into equation (6) we have,

$$\mathbf{J}_{min} = \sigma_d^2 - \mathbf{p}^H \sum_{i=1}^{N} \lambda_i^{-1} \mathbf{q}_i \mathbf{q}_i^H \mathbf{p} \tag{12}$$

The above equation can be further written as,

$$\mathbf{J}_{min} = \sigma_d^2 - \sum_{i=0}^{N-1} \frac{\mathbf{p}^H \mathbf{q}_i \mathbf{p}^H \mathbf{q}_i^*}{\lambda_i}$$

$$= \sigma_d^2 - \sum_{i=1}^{N} \frac{|\, \mathbf{q}^H \mathbf{p} \,|^2}{\lambda_i} \tag{13}$$

We know that the eigenvalues of $\mathbf{R}$ are real and non-negative, and moreover for practical signals they are always positive. Hence we can say that the subtractor of (13) is always positve, which implies that the $\mathbf{J}_{min}$ in equation (13) increases with the decrease in the order of the filter. This is illustrated in figure 6.

4 Conclusions

In this paper, a fault tolerent algorithm is proposed and an architecture is presented which implements the

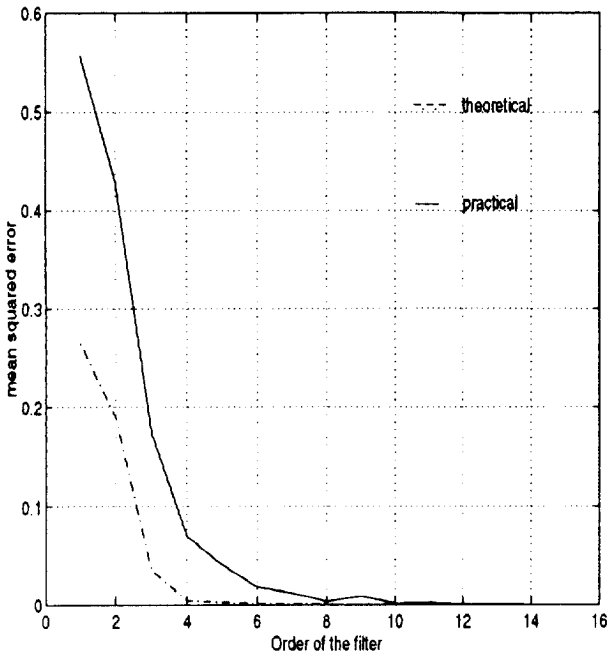

Figure 6: Misadjustment curve with respect to the order of the filter.

proposed algorithm. The algorithm includes a fault detection scheme, a fault location scheme and a stretegy for recovery via hardware reconfiguration. The increase in the steady state misadjustment, due to the decrease in the order of the filter is also analyzed.

References

[1] B. Widrow, "Adaptive Filters," in *Aspects of Network and system theory*, R. E. Kalman and N. de Claris, eds.; New York: Holt, Rinehart, and Winston, 1971, pp. 653-687.

[2] M. D. Meyer and D. P. Agarwal, "A High Sampling Rate Delayed LMS Adaptive Filter," *IEEE Trans. on Circuits and Systems II* pp. 1234-1236, March 1994

[3] M. D. Meyer and D. P. Agrawal, "A Modular Pipelined Implementation of a Delayed LMS Transversal Adaptive Filter," *Proceedings*, 1990 IEEE Symposium on Circuits and Systems, vol. 3, pp. 1712-1715.

[4] N. R. Shanbhag and K. K. Parhi, "A Pipelined LMS Adaptive Filter Architecture," *Proceedings*, 25th Asilomar Conference on Signals, Systems and Computers, 1991, pp. 668-672.

[5] T. H. Y. Meng and D. G Messerschmitt, "Arbitrily High Sampling Rate Adaptive Filters," *IEEE Trans. on Acous., Speech, and Signal Processing*, vol. ASSP-35, pp. 455-470, April 1987.

[6] M. D. Meyer and D. P. Agrawal, "Adaptive Lattice Filter Implementations on pipelined Multiprocessor Architectures," *IEEE Trans. on Communications*, vol. 69, no. 1, pp. 1234-123, January 1990.

[7] Arnold L. Rosenberg, "The Diogenes Approach to Testable Fault Tolerant Arrays of Processors," *IEEE Trans. on Computers*, vol. C-32, October 1983.

[8] J. H. Kim, "On-line Detection of Errors in Homogeneous Multiprocessor Systems," *Proceedings*, 1986 Real Time System Symposium, pp. 55-62, December 1986.

[9] J. H. Kim and Sudhakar M. Reddy, "On the Design of Fault-Tolerant Two Dimensional Systolic Arrays for Yield Enhancement," *IEEE Trans. on Computers*, vol. 38, No. 4, April 1989.