A Systolic Array Realization of an LMS Adaptive Filter and the Effects of Delayed Adaptation

Hanan Herzberg, Raziel Haimi-Cohen, *Member, IEEE*, and Yair Be'ery, *Member, IEEE*

***Abstract*—This paper presents a design of a systolic array of an adaptive filter. The filter is based on the least mean square (LMS) algorithm, but due to the problems in implementation of the systolic array, a modified algorithm, a special case of the delayed LMS (DLMS), is used. The DLMS algorithm introduces a delay in the updating of the filter coefficients. The convergence and steady state behavior of the systolic array are analyzed. It is shown that the performance of the systolic array is similar to that of a conventional LMS implementation in a wide range of practical conditions.**

I. Introduction

ADAPTIVE filters have a wide range of applications [1], [2], such as echo cancellation, adaptive equalization, and noise cancellation. A great deal of research (e.g., [3]–[5]) has been conducted on the efficient implementation of an adaptive filter. One of the most common algorithms for adaptive filtering is Widrow's least mean square (LMS) algorithm [1]. An Nth order adaptive filter of this type may be represented by the following equations:

$$\text{Output: } y(n) = \sum_{k=0}^{N-1} w_k(n)x(n-k) \tag{1}$$

$$\text{Error: } e(n) = g(n) - y(n) \tag{2}$$

$$\text{Coefficient: } w_k(n+1) = w_k(n) + \mu e(n)x(n-k)$$

$$k = 0, 1, \cdots, N-1 \tag{3}$$

where

- $x(n)$ filter input (reference signal) at time n,
- $w_k(n)$ kth filter coefficient at time n ($k = 0, \cdots, N-1$),
- $y(n)$ filter output,
- $g(n)$ desired response (primary signal),
- $e(n)$ residual error, and
- μ step size (a small positive number).

This algorithm requires $2N$ multiply–add operations at each time interval n. Many applications (e.g., high rate data transmission as in the ISDN system [6]) require fast computation by adaptive filters of a large order. Serial processing implementation may be impractical because of available processing rates and a parallel configuration should be considered. Systolic arrays, which have recently been applied to fast digital signal processing [7], [8], are one of the most viable alternatives because of the great progress in VLSI technology.

Systolic-array design is essentially the decomposition of the system into simple processing units, interconnected by appropriate delays. The algebra of [9], which is a systematic method for performing this decomposition, together with a retiming process, leads to the satisfaction of the so-called "systolic conditions." This approach maintains the correctness of the design.

In this paper we utilize this algebra for the design of a systolic-array implementation for adaptive filters based on the LMS algorithm. However, since the LMS algorithm contains a feedback loop, the delays created in the decomposition and retiming process prohibit the exact implementation of the algorithm as given by (1)–(3). The design procedure of [9] leads to a systolic array which implements a special case of the so-called delayed LMS (DLMS) algorithm [10], [11]. The error $e(n)$ used in this algorithm is available only after the processing delay of the systolic array, and thus, the update of the coefficients is performed with this delay.

The formal design of an array consisting of $2N$ identical processing elements (PE) with maximum efficiency is presented in Section II. Discussion and conclusions are given in Section III. The conclusions are based on an analysis of the convergence and the steady state behavior of the implemented adaptive filter, using the theoretical results of [10], [11]. It is shown that under reasonable assumptions the systolic array has the same performance as the conventional implementation of the LMS algorithm.

II. Design of the Systolic Array

Fig. 1 shows a direct implementation of the LMS algorithm as stated by (1) and (3), where $r(n) = \mu e(n)$ and z^{-1} is a unit delay. Figs. 2–4 present a sequence of changes in the scheme of Fig. 1, which eventually enable the utilization of the algebra of [9]. Fig. 2 is a z-notation of Fig. 1. The adder of N inputs is split into N adders of two inputs and the input, $x(n)$, to the filter is presented as

Manuscript received October 2, 1989; revised August 9, 1991.

H. Herzberg is with the Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel.

R. Haimi-Cohen is with the DSP Group (Israel) Ltd., Givat-Shmuel, 51905 Israel.

Y. Be'ery is with the Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel, and with the DSP Group (Israel) Ltd., Givat-Shmuel, 51905 Israel.

IEEE Log Number 9202781.

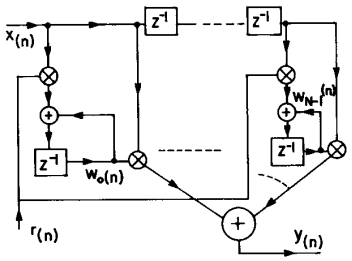


Fig. 1. Direct implementation of the LMS algorithm.

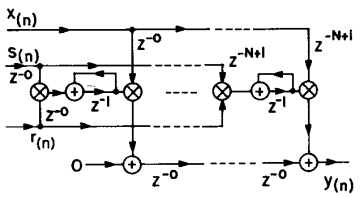


Fig. 2. Z-notation of the LMS algorithm.

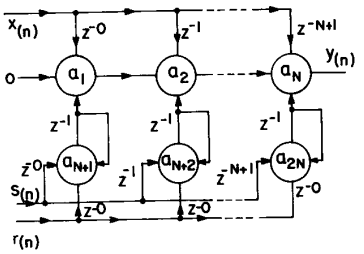


Fig. 3. Z-graph representation of the LMS algorithm.

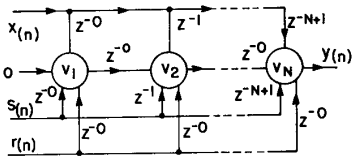


Fig. 4. Short form of the Z-graph for LMS algorithm.

two signals: $x(n)$, the input to the convolution process and $s(n)$, the input for updating the filter's coefficients. In the LMS algorithm $x(n)$ and $s(n)$ are the same, whereas in the systolic array $s(n)$ is a delayed version of $x(n)$ (as will be shown later). By combining each multiplier and adder into a multiply-add processor element (PE) a_i we get the Z-graph [9] representation in Fig. 3. Note that in this Z-graph, the same type of processors are used for computing the convolution and for updating the coefficients.

In order to utilize the algebra of [9] efficiently, we will use a short form of the Z-graph (Fig. 4). This is achieved by combining the two PE's a_i and a_{N+i} of Fig. 3 into one node with a single output, v_i, in Fig. 4. Each node in Fig. 4 is associated with a function that computes the output and updates the respective coefficient. Therefore, the dependence relations in the conventional adaptive filter can be described by the algebraic representation of [9] as follows:

$$V \leftarrow AV \text{ ``+'' } BI(n) \tag{4}$$

where

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}; \quad A = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ z^{-0} & 0 & \cdots & 0 & 0 \\ 0 & z^{-0} & 0 \cdots & 0 & 0 \\ & 0 & & & \vdots \\ & & \ddots & z^{-1} & 0 \end{bmatrix}$$

$$I(n) = \begin{bmatrix} x(n) \\ s(n) \\ r(n) \end{bmatrix}; \quad B = \begin{bmatrix} z^{-0} & z^{-0} & z^{-0} \\ z^{-1} & z^{-1} & z^{-0} \\ \vdots & \vdots & \vdots \\ z^{-N+1} & z^{-N+1} & z^{-0} \end{bmatrix}$$

and

$$y = C^T V \tag{5}$$

where

$$C^T = [0 \quad 0 \cdots 0 \quad z^{-0}].$$

Note that the "+" in (4) as well as the additions implied by the matrix multiplications represent the combination of information in terms of the algebra of [9] rather than the usual arithmetic addition.

Let D be a diagonal matrix $D = \text{diag}\,(z^{d(1)}, \cdots, z^{d(N)})$. This operator maps a vector V into another vector U, so that

$$U = DV; \quad V = D^{-1}U. \tag{6}$$

The matrix D delays the output of the PE v_i by $d(i)$, $i = 1, 2, \cdots, N$. By multiplying (4) and (5) by D and using (6) we get

$$U \leftarrow (DAD^{-1})U \text{ ``+'' } (DB)I(n) \tag{7}$$

$$y = (C^T D^{-1})U. \tag{8}$$

An important step in the design is the selection of the matrix D. The following algebraic manipulations are aimed at obtaining D such that, by substituting $u_i \leftarrow z^{-d(i)}v_i$ in the design, the delay of $y(n)$ will be minimal while the following systolic conditions will be satisfied [9]:

I) $d(i) + a(ij) - d(j) \geq 1$

$$\text{for } i, j = 1, 2, \cdots, N \tag{9}$$

where $A = (z^{-a(ij)})$.

II) All nonzero entries of any column of DAD^{-1} and DB must be different from each other.

In order to satisfy condition I, communication without a delay unit between the nodes must be prevented. This reduces the cycle time of the filter. In order to satisfy condition II global communication must be prevented.

From condition I we get

$$d(i+1) - d(i) \geq 1; \quad i = 1, 2, \cdots, N-1. \tag{10}$$

The columns of DAD^{-1} fulfill condition II for any $d(i)$, but for columns 1, 2 of DB, condition II implies

$$d(i) \neq d(i+1) + 1; \quad i = 1, 2, \cdots, N-1 \tag{11}$$

and for column 3

$$d(i) \neq d(j); \quad i \neq j; \quad i, j = 1, 2, \cdots, N. \tag{12}$$

Solving (10)–(12) yields

$$d(i+1) = d(i) + 1; \quad i = 1, 2, \cdots, N-1. \tag{13}$$

Because the minimal possible value of $d(1)$ is zero (for causality) we can choose the following D:

$$D = \begin{bmatrix} z^{-0} & & & \\ & z^{-1} & & 0 \\ & & \ddots & \\ 0 & & & z^{-N+1} \end{bmatrix}. \tag{14}$$

Hence, by (4), (5), and (14),

$$DAD^{-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ z^{-1} & 0 & \cdots & 0 & 0 \\ 0 & z^{-1} & 0 \cdots & 0 & 0 \\ & 0 & \ddots & & \vdots \\ & & & z^{-1} & 0 \end{bmatrix} \tag{15}$$

$$DB = \begin{bmatrix} z^{-0} & z^{-0} & z^{-0} \\ z^{-2} & z^{-2} & z^{-1} \\ \vdots & \vdots & \vdots \\ z^{-2N+2} & z^{-2N+2} & z^{-N+1} \end{bmatrix} \tag{16}$$

and

$$C^T D^{-1} = [0, 0, \cdots, z^{N-1}]. \tag{17}$$

Substituting (14)–(16) into (7), and (8), yields the systolic Z-graph of Fig. 5. The function associated with u_i is the same as that of v_i but delayed by $d(i)$. Practically, we can't implement z^{N-1} of (17) (this is the output segment in Fig. 5) by a causal system. Thus the output $y(n)$ must be delayed by $N-1$ in the systolic array (see Fig. 6). In Fig. 6 we replace the nodes u_i by the multiply–add processor elements a_i and a_{i+N} from Fig. 3.

Due to the delay by $N-1$ imposed on $y(n)$ by the systolic conditions, we have to add a delay of $N-1$ to $g(n)$ (the desired response in (2)). Hence the residual error $e(n)$ is delayed by $N-1$. Consequently, the input $r(n)$ is delayed by $N-1$, and for correct updating of the filter's coefficients we have to set

$$s(n) = x(n-N+1). \tag{18}$$

The final architecture of the systolic array for the adaptive filter is given in Fig. 7 with $2N$ processing elements

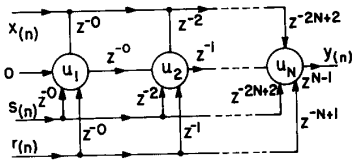


Fig. 5. Z-graph for the systolic array (short form).

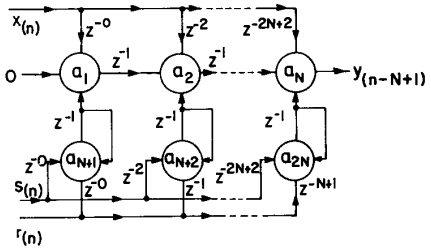


Fig. 6. Z-graph for the systolic array.

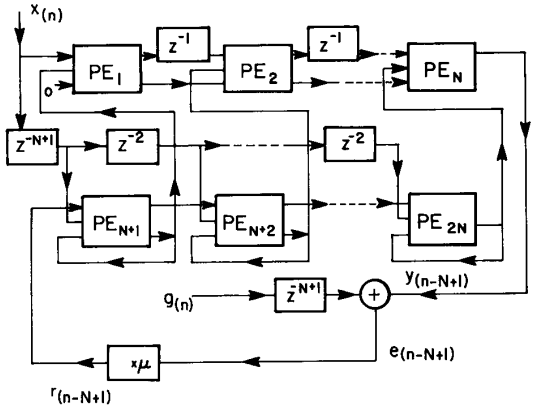


Fig. 7. A systolic array for adaptive filter.

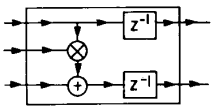


Fig. 8. Processing element (PE).

which are described in Fig. 8 (PE_N should be without the two delay units). The structure in Fig. 7 is based on that of Fig. 6 with additional parts for generating the residual error according to the aforementioned delay.

Let $X(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T$ and $W(n) = [w_0(n), w_1(n), \cdots, w_{N-1}(n)]^T$. Equation (18) implies the following modification in the vector form of (3):

$$W_k(n+1) = W_k(n) + \mu e(n-N+1) \cdot X(n-k-N+1). \tag{19}$$

This means that the updating of each of the filter coefficients is delayed by $N-1$. Therefore, the systolic array implements a special case of the delayed LMS (DLMS) algorithm [10], [11].

III. Discussion and Conclusion

For the following discussion we assume that $\{g(n)\}$ and $\{x(n)\}$ are wide-sense stationary stochastic processes and that $W(n)$ is independent of $X(n)$ (see [1]). We also assume that the distribution of $X(n)$ is either Gaussian (not necessarily white) or white binary (WB) (the entries of $X(n)$ are independent and have the values of $\{1, -1\}$ with equal probability). The Gaussian distribution is applicable to analog signals such as speech, whereas the WB distribution can be found in many digital communication applications (e.g., echo cancellation and adaptive equalization) in which scramblers are incorporated. Let

$$E\{X(n)X^T(n)\} = R = Q^{-1}MQ \quad \text{and} \quad E\{g(n)x(n)\} = p \tag{20}$$

where $M = \text{diag}\,(m_1, m_2, \cdots, m_N)$ $m_1 \geq m_2 \geq \cdots \geq m_N > 0$ are the eigenvalues of R and the columns of Q are the corresponding eigenvectors of R. Let the excess mean-square error (EMSE) be defined by EMSE $(n) \equiv E\{e^2(n)\} - \Phi$, where $\Phi = P^TR^{-1}P$ is the mean-square error corresponding to Wiener's optimal filter.

By using the results presented in [10], [11] for a DLMS algorithm with $d = N - 1$, we deduce that, for the systolic array, a sufficient condition for convergence, without oscillation, of the EMSE and of the mean of the coefficients is

$$\mu m_1 \leq 2\frac{(N-1)^{N-1}}{N^N} = 2(1/N)(1-1/N)^{N-1} \approx \frac{2e^{-1}}{N}. \tag{21}$$

Let z_i be the root with maximum absolute magnitude of $z^{N-1}(1-z) = \mu m_i$. From [10] and [11] we also deduce that if the EMSE converges its limit is given by

$$\text{EMSE}\,(\infty) = \frac{\sum_{i=1}^{N}(1-z_i)\Phi}{2 - \alpha\beta - \sum_{i=1}^{N}(1-z_i)} \tag{22}$$

where $1 - z_N \leq \beta \leq 1 - z_1$ and α depends on the distribution of $X(n)$; for the Gaussian case $\alpha = 2$ and for the WB case $\alpha = 0$.

As $\mu \to 0$, z_i $(i = 1, 2, \cdots, N)$ may be approximated by $1 - \mu m_i$, thus (22) is reduced to the common formula for the EMSE of the LMS adaptive filter (see [2]). Most practical implementations require μ to be sufficiently small, thus the delay in the adaptation has no significant influence on the performance of the filter.

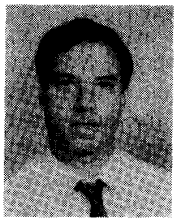
This paper has addressed the implementation of the LMS algorithm by a systolic array and the issues evolving from the interaction between algorithmic formulation and VLSI architectural constraints. In [3]–[5] we may find other approaches for the design of adaptive algorithms, including the LMS algorithm. However, in light of the quote from [3]: "the total processing latency must be less than one sample period," it is clear that the approach in [3] does not comply with the systolic conditions (see [9]) and hence the structures of [3]–[5] provide a slower throughput than the systolic array presented in this paper (Fig. 7). The difference in the throughput increases with the number of taps in the filter. The penalty for complying with the systolic conditions is the delay by $N - 1$ in updating each of the filter coefficients. However, as we have shown, for most of the practical applications, this delay has a negligible effect on the convergence and the steady state behavior of the adaptive filter.

Acknowledgment

The authors were with Tadiran-Telecommunication Division Ltd., Israel, during the preparation of part of this paper.

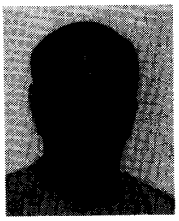
References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[2] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
[3] V. B. Lawrence and S. K. Tewksbury, "Multiprocessor implementation of adaptive digital filters," *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 826–835, June 1983.
[4] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 744–752, June 1981.
[5] T. M. Miller, S. T. Alexander, and L. J. Faber, "A SIMD multiprocessor ring architecture for LMS adaptive algorithm," *IEEE Trans. Commun.*, vol. COM-34, no. 1, pp. 89–92, Jan. 1986.
[6] D. G. Messerschmitt, "Design issues in the ISDN U-interface transceiver," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 8, pp. 1281–1293, Nov. 1986.
[7] S. Y. Kung, "VLSI array processors," *IEEE ASSP Mag.*, July 1985.
[8] S. Y. Kung, H. J. Whitehouse, and T. Kailath, *VLSI and Modern Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[9] H. T. Kung and W. L. Lin, "An algebra for VLSI algorithm design," in *Proc. Conf. Elliptic Problem Solvers* (Monterey, CA), Jan. 1983; also Res. Rep. CMU-CS-84-100, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburg, PA, Apr. 1983.
[10] R. Haimi-Cohen, H. Herzberg, and Y. Be'ery, "Delayed adaptive LMS filtering: Current results," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (New Mexico), Apr. 1990, pp. 1273–1276.
[11] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 1397–1405, Sept. 1989.



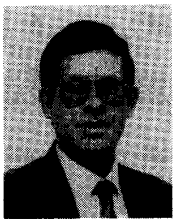
Hanan Herzberg was born in Israel in 1958. He received the B.Sc. and M.Sc. degrees in electrical engineering with distinctions from Tel Aviv University, Israel, in 1984 and 1987, respectively. He is presently working toward the Ph.D. degree in electrical engineering at the same university.

From 1984 to 1988 he was with the Research and Development Department, Communication Division, Tadiran, Ltd., Israel, where he is currently a consultant for digital communication and digital signal processing projects. Since 1988 he has also been a Teaching Assistant at Tel Aviv University. His current research interests include coding theory, digital communication, and digital signal processing.



Raziel Haimi-Cohen (S'82–M'84) was born in Israel in 1954. He received the B.Sc. degree in mathematics from Tel Aviv University, Tel Aviv, Israel, in 1974, the M.Sc. degree in applied mathematics from Cornell University, Ithaca, NY, in 1980, and the Ph.D. degree in electrical engineering from Ben-Gurion University, Beer-Sheva, Israel, in 1988.

Between 1980 and 1983 he was with the Brain and Behavior Research Unit at Hadassah University Hospital, Jerusalem, Israel, where he conducted research on the analysis and interpretation of neurobiological signals, which was also the subject of his Ph.D. dissertation. Between 1984 and 1986 he worked at the Telecommunication Division of Tadiran, Ltd., where he was involved in the development of an adaptive echo canceller. Since 1986 he has been with the DSP Group (Israel) Ltd., where he was Speech Recognition Manager between 1987 and 1989 and later a Senior Scientist, leading research and development projects in the areas of speech recognition and processing, digital signal processing, adaptive systems, and communication. He is currently the Advanced Technology Director at the DSP Group (Israel) Ltd.



Yair Be'ery (M'87) was born in Tel Aviv, Israel. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 1979, 1979, and 1985, respectively.

He is currently with the Department of Electronic Communications, Control, and Computer Systems, Tel Aviv University, Tel Aviv, Israel. From 1979 to 1985 he was involved in the research and development of digital signal processing systems and design of VLSI architectures for special purpose processors. He received the Eliyahu Golomb Award from the Ministry of Defense, Israel, in 1984, and the Rothschild Fellowship for postdoctoral studies at the Rensselaer Polytechnic Institute, Troy, NY, in 1986. His research interests include error control coding, VLSI architectures and systolic arrays, and adaptive algorithms for speech and data transmission.