Proceedings of ITC-CSCC 2000, Pusan, Korea

Pipelined Adaptive Digital Filters Based on Affine Projection Algorithms with Order 2

Mitsuji Muneyasu, Takeshi Harada and Takao Hinamoto

Faculty of Engineering, Hiroshima University
1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527 Japan
E-mail: muneyasu@ecl.sys.hiroshima-u.ac.jp

Abstract

This paper proposes a pipelined adaptive filter based on affine projection algorithm with order 2. This filter gives a better convergence performance than that of LMS or NLMS pipeline algorithm and has same latency with the pipeline algorithm based on equivalent transformation. Compared to the critical path of the pipeline NLMS implementation, only 2 additions are increased in that of the proposed implementation.

1 Introduction

An adaptive digital filter has become a key technology for various fields like mobile equipment, hard-disk controllers and others [1]. In implementation of adaptive digital filters, the properties of high-speed processing and low power consumption are required. To achieve these requirements, pipeline implementation is one of an important technique. For example, the delayed least mean square (DLMS) algorithm which is a variation of LMS algorithm, has been applied to the pipeline implementation [2]. However this algorithm has a slow convergence property and increases the output latency [2]. Equivalent transformations for LMS or NLMS algorithm are proposed to solve this problem [3],[4]. By using this method, the same convergence property with ordinary LMS or NLMS algorithm is obtained.

On the other hand, the convergence property for colored inputs is also an essential factor in the adaptive algorithm. The equivalent transformation cannot improve the slow convergence property of LMS and NLMS algorithms for colored inputs.

In this paper, a pipeline architecture for an adaptive filter based on the affine projection algorithm (APA) with order 2. The same latency with the LMS or NLMS pipeline algorithm based on the equivalent transformation is given and a better convergence performance is also obtained. In the critical path of the proposed implementation, only 2 additions are increased compared to that of the pipeline NLMS implementation.

2 Equivalent Transformation of APA with Order 2

2.1 The APA with order 2

The update equation for adaptive filters by the affine projection algorithm with order 2 is defined as follows:

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\alpha e(n)}{\boldsymbol{x}^T(n)\boldsymbol{u}(n)}\boldsymbol{x}(n) \tag{1}$$

$$\boldsymbol{x}(n) = \boldsymbol{u}(n) - \frac{\boldsymbol{u}^T(n-1)\boldsymbol{u}(n)}{\|\boldsymbol{u}(n-1)\|^2}\boldsymbol{u}(n-1) \tag{2}$$

$$e(n) = d(n) - \boldsymbol{w}^T(n)\boldsymbol{u}(n) \tag{3}$$

where $\boldsymbol{w}(n)$ and $\boldsymbol{u}(n)$ mean the tap coefficient vectors and input vectors at time n, respectively. These vectors are given by

$$\boldsymbol{w}(n) = [w_0(n),\ w_1(n),\ \cdots,\ w_{N-1}(n)]^T, \tag{4}$$

$$\boldsymbol{u}(n) = [u(n),\ u(n-1),\ \cdots,\ u(n-N+1)]^T \tag{5}$$

where N is the number of taps of adaptive filters. $e(n)$ and $d(n)$ represent an error signal and a desired signal at time n, respectively and the parameter α satisfies $0 < \alpha < 2$. An output of adaptive filter $y(n)$ at time n, is given as

$$y(n) = \boldsymbol{w}^T(n)\boldsymbol{u}(n) \ . \tag{6}$$

2.2 Look-ahead transformation

For the pipelining of the calculation of $y(n)$, we introduce look-ahead transformation [3]. From Eqs. (1) and (2), the following equations are obtained:

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \frac{\alpha e(n-1)}{\boldsymbol{x}^T(n-1)\boldsymbol{u}(n-1)}\boldsymbol{x}(n-1) \tag{7}$$

$$\boldsymbol{x}(n-1) = \boldsymbol{u}(n-1) - \frac{q(n-1)}{p(n-2)}\boldsymbol{u}(n-2) \tag{8}$$

where $p(n) = \|\boldsymbol{u}(n)\|^2$ and $q(n) = \boldsymbol{u}^T(n-1)\boldsymbol{u}(n)$. Substituting Eqs.(7) and (8) to Eq. (6),

$$\begin{aligned} y(n) &= \boldsymbol{w}^T(n-1)\boldsymbol{u}(n) \\ &+ \frac{\alpha e(n-1)p(n-2)\boldsymbol{u}^T(n-1)\boldsymbol{u}(n)}{p(n-1)p(n-2)-q(n-1)^2} \\ &- \frac{\alpha e(n-1)q(n-1)\boldsymbol{u}^T(n-2)\boldsymbol{u}(n)}{p(n-1)p(n-2)-q(n-1)^2} \end{aligned} \tag{9}$$

is obtained. This operation is called as 1-step look-ahead transformation and the following equation is given by repeating 1-step look-ahead transformation n times:

$$y(n) = \frac{\alpha e(0)\boldsymbol{u}^T(0)\boldsymbol{u}(n)}{\|\boldsymbol{u}(0)\|^2} + \sum_{k=1}^{n-1} \frac{\alpha e(n-k)p(n-k-1)\boldsymbol{u}^T(n-k)\boldsymbol{u}(n)}{p(n-k)p(n-k-1)-q(n-k)^2} - \sum_{k=1}^{n-1} \frac{\alpha e(n-k)q(n-k)\boldsymbol{u}^T(n-k-1)\boldsymbol{u}(n)}{p(n-k)p(n-k-1)-q(n-k)^2} \quad (10)$$

where the initial conditions $\boldsymbol{w}(0) = \boldsymbol{0}$ and $\boldsymbol{x}(0) = \boldsymbol{u}(0)$ are assumed. This operation is called as n-step look-ahead transformation.

2.3 Pipelining of the calculation of $y(n)$

In this section, a technique for the pipelining of the calculation of $y(n)$ is shown.

Eq. (10) can be rewritten as

$$y(n) = y^{(0)}(n) + y^{(1)}(n) - y^{(2)}(n) \quad (11)$$

where

$$y^{(0)}(n) = \frac{\alpha e(0)\boldsymbol{u}^T(0)\boldsymbol{u}(n)}{\|\boldsymbol{u}(0)\|^2}$$

$$y^{(1)}(n) = \sum_{k=1}^{n-1} \frac{\alpha e(n-k)p(n-k-1)\boldsymbol{u}^T(n-k)\boldsymbol{u}(n)}{p(n-k)p(n-k-1)-q(n-k)^2}$$

$$y^{(2)}(n) = \sum_{k=1}^{n-1} \frac{\alpha e(n-k)q(n-k)\boldsymbol{u}^T(n-k-1)\boldsymbol{u}(n)}{p(n-k)p(n-k-1)-q(n-k)^2} .$$

First, noted that $\boldsymbol{u}(0) = [\, u(0),\ 0,\ \cdots,\ 0\,]$, $y^{(0)}(n)$ can be expressed by

$$y^{(0)}(n) = \frac{\alpha e(0)u(0)u(n)}{u(0)^2} = \frac{\alpha e(0)u(n)}{u(0)} . \quad (12)$$

Next, $y^{(1)}(n)$ and $y^{(2)}(n)$ can be rewritten as

$$y^{(1)}(n) = \sum_{i=0}^{N-1} \Bigg[h_i^{(1)}(n-i-1)u(n-i) + \left\{ r_i^{(1)}(n-i-1) + u(n-2i-1)u(n-i) \right\} \cdot \frac{\alpha e(n-i-1)p(n-i-2)}{p(n-i-1)p(n-i-2)-q(n-i-1)^2} \Bigg] , \quad (13)$$

$$y^{(2)}(n) = \sum_{i=0}^{N-1} \Bigg[h_i^{(2)}(n-i-1)u(n-i) + \left\{ r_i^{(2)}(n-i-1) + u(n-2i-2)u(n-i) \right\} \cdot \frac{\alpha e(n-i-1)q(n-i-1)}{p(n-i-1)p(n-i-2)-q(n-i-1)^2} \Bigg] \quad (14)$$

where $h_i^{(1)}(n)$, $h_i^{(2)}(n)$, $r_i^{(1)}(n)$ and $r_i^{(2)}(n)$, $i = 0, 1, \cdots, N-1$ are defined by

$$h_i^{(1)}(n) = \sum_{j=1}^{n-i-2} \frac{\alpha e(n-j)p(n-j-1)u(n-i-j)}{p(n-j)p(n-j-1)-q(n-j)^2} \quad (15)$$

$$h_i^{(2)}(n) = \sum_{j=1}^{n-i-2} \frac{\alpha e(n-j)q(n-j)u(n-i-j-1)}{p(n-j)p(n-j-1)-q(n-j)^2} \quad (16)$$

$$r_i^{(1)}(n) = \begin{cases} \displaystyle\sum_{j=0}^{N-i-2} u(n-i-j-1)u(n-j), & (i = 0, 1, \cdots, N-2) \\ 0, & (i = N-1) \end{cases} \quad (17)$$

$$r_i^{(2)}(n) = \begin{cases} \displaystyle\sum_{j=0}^{N-i-2} u(n-i-j-2)u(n-j), & (i = 0, 1, \cdots, N-2) \\ 0, & (i = N-1) \end{cases} , \quad (18)$$

respectively.

In the following, we show the possibility of the pipeline implementation of this adaptive filter. $c_i^{(1)}(n)$ and $c_i^{(2)}(n)$ are defined as

$$c_i^{(1)}(n) = h_i^{(1)}(n-1)u(n) + \left\{ r_i^{(1)}(n-1) + u(n-i-1)u(n) \right\} \cdot \frac{\alpha e(n-1)p(n-2)}{p(n-1)p(n-2)-q(n-1)^2} , \quad (19)$$

$$c_i^{(2)}(n) = h_i^{(2)}(n-1)u(n) + \left\{ r_i^{(2)}(n-1) + u(n-i-2)u(n) \right\} \cdot \frac{\alpha e(n-1)q(n-1)}{p(n-1)p(n-2)-q(n-1)^2} , \quad (20)$$

respectively. From these equations, Eqs. (13) and (14) can be represented as

$$y^{(1)}(n) = \sum_{i=0}^{N-1} c_i^{(1)}(n-i), \quad y^{(2)}(n) = \sum_{i=0}^{N-1} c_i^{(2)}(n-i)$$

Therefore, $y_i^{(1)}(n)$ and $y_i^{(2)}(n)$ are rewritten by

$$y_i^{(1)}(n) = \begin{cases} y_{i+1}^{(1)}(n-1) + c_i^{(1)}(n), & \\ \quad (i = 0, 1, \cdots, N-1) & , \\ 0, \quad (i = N) & \end{cases} \tag{21}$$

$$y_i^{(2)}(n) = \begin{cases} y_{i+1}^{(2)}(n-1) + c_i^{(2)}(n), & \\ \quad (i = 0, 1, \cdots, N-1) & , \\ 0, \quad (i = N) & \end{cases} \tag{22}$$

respectively. From the definition of $y_i^{(1)}(n)$ and $y_i^{(2)}(n)$, we obtain $y^{(1)}(n) = y_0^{(1)}(n)$ and $y^{(2)}(n) = y_0^{(2)}(n)$.

Eqs. (21) and (22) mean that each calculation part of $y_i^{(*)}(n)$ is considered as a processing module whose input is $y_{i+1}^{(*)}(n-1)$, respectively. By connecting N modules in series and inserting a delay between each module, we can implement the calculation of $y_i^{(1)}(n)$ and $y_i^{(2)}(n)$ in a pipelined manner.

The coefficients, $h_i^{(1)}(n)$, $h_i^{(2)}(n)$, $r_i^{(1)}(n)$, $r_i^{(2)}(n)$, $p(n)$ and $q(n)$ in each module can be recursively calculated in the following:

$$h_i^{(1)}(n) = h_i^{(1)}(n-1) + \frac{\alpha e(n-1)p(n-2)u(n-i-1)}{p(n-1)p(n-2) - q(n-1)^2}$$

$$h_i^{(2)}(n) = h_i^{(2)}(n-1) + \frac{\alpha e(n-1)q(n-1)u(n-i-2)}{p(n-1)p(n-2) - q(n-1)^2}$$

$$r_i^{(1)}(n) = r_i^{(1)}(n-1) + u(n-i-1)u(n) - u(n-N)u(n-N+i+1)$$

$$r_i^{(2)}(n) = r_i^{(2)}(n-1) + u(n-i-2)u(n) - u(n-N-1)u(n-N+i+1)$$

$$p(n) = p(n-1) + u(n)^2 - u(n-N)^2$$

$$q(n) = q(n-1) + u(n-1)u(n) - u(n-N-1)u(n-N)$$

We can considerably reduce the amount of computation by using these formulas.

3 Proposed structure

In this section, a pipeline structure based on the formulas in the previous section is proposed. Fig. 1 shows the proposed pipeline structure. In this figure, the Unit B calculates $y^{(0)}(n)$, the calculation of $y^{(1)}(n)$ and $y^{(2)}(n)$ are implemented by the cascade connection of N modules which compute $y_i^{(1)}(n)$ and $y_i^{(2)}(n)$ and the outputs of each 0-th module become $y^{(1)}(n)$ and $y^{(2)}(n)$. The Unit A computes $\mu_1 = \alpha p(n-1)/\{p(n)p(n-1) - q(n)^2\}$ and $\mu_2 = \alpha q(n)/\{p(n)p(n-1) - q(n)^2\}$, simultaneously.

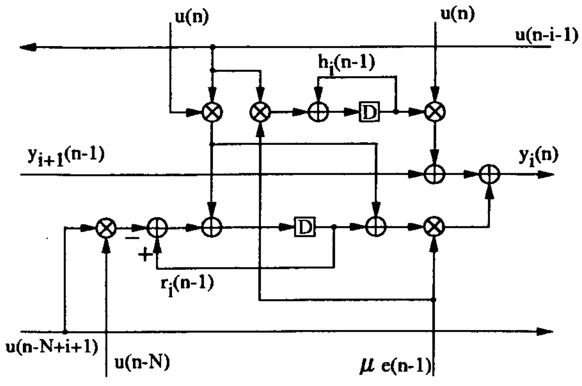

Figure 2: Structure of each module.

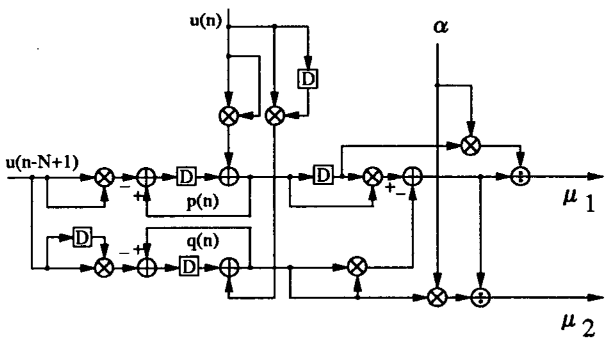

Figure 3: Structure of Unit A.

The structure of each module in Fig. 1 is shown in Fig. 2 and the Unit A and Unit B are shown in Figs. 3 and 4, respectively.

The output latency and the critical path of the proposed structure are as follows. In proposed structure, no delay exists between $u(n)$ and $y(n)$ from Figs. 1 and 2. As a result, the output latency always becomes 0.

Noted that Figs. 1, 2 and 3, the critical path t_{cp} is obtained as

$$t_{cp} = \begin{cases} 2t_{mult} + 5t_{add} & (t_{div} \leq 3t_{add}) \\ t_{div} + 2t_{mult} + 2t_{add} & (3t_{add} < t_{div}) \end{cases}$$

where t_{div}, t_{mult} and t_{add} mean the times required one division, multiplication and addition, respectively.

We compare the proposed structure with the other pipelined architecture based on DLMS [2], LMS [3], NLMS [4] and non-pipelined architecture based on the affine projection algorithm with order 2 in view of the critical path (CP), the output latency (OL) and the number of required operations. This comparison is summarized in Table 1. In this table, we assume that the number of taps is N and that $t_{div} \leq 3t_{add}$. From the comparison, the CP is

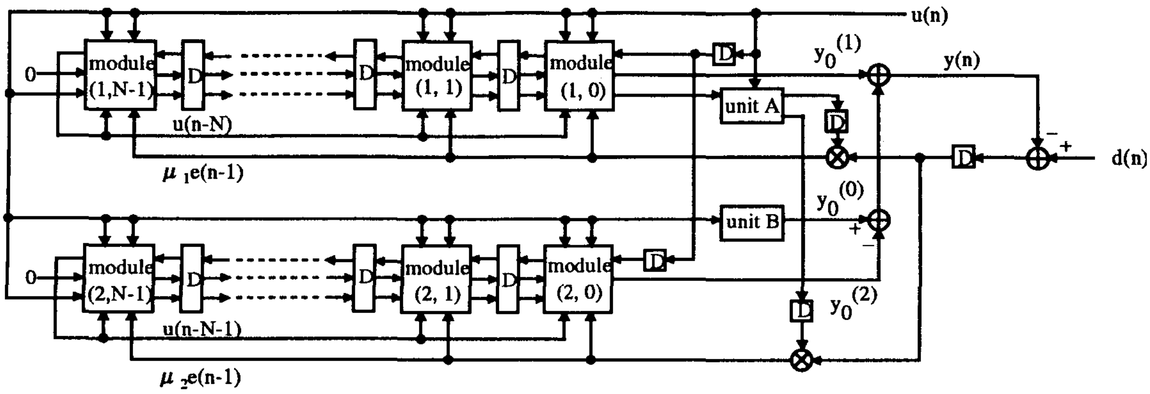

Figure 1: Pipeline structure.

Table 1: Comparison with other implementations

Architecture	CP	OL	Number of operators			
			Multiplier	Divider	Adder	Delay
DLMS	$t_{mult}+2t_{add}$	N	$2N+1$	0	$2N+1$	$8N-2$
LMS	$3t_{mult}+3t_{add}$	0	$5N-2$	0	$5N-2$	$5N-2$
NLMS	$2t_{mult}+3t_{add}$	0	$5N+2$	1	$6N$	$5N$
non-pipelined AP with order 2	$2t_{div}+(2N+3)t_{mult}$ $+(2N+1)t_{add}$	0	$4N+5$	2	$4N+3$	$3N+4$
proposed	$2t_{mult}+5t_{add}$	0	$10N+10$	2	$12N+2$	$10N+2$

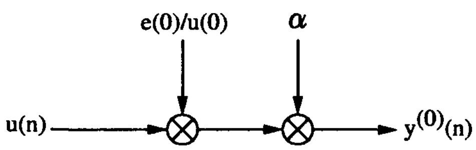

Figure 4: Structure of Unit B.

slightly longer than that of the other pipeline implementation, but it is independent on N. The OL is not increased. From this facts, it is known that the proposed architecture is effective compared to the non-pipelined one. From the point of the number of the operations, more number of operations for this architecture are required than the other implementations, because of the equivalent transformation and more complicated algorithm.

4 Conclusion

This paper has proposed the pipeline implementation of the adaptive filters based on the affine projection algorithm with order 2. In this technique, the equivalent transformation is used for the pipeline implementation. The critical path of the proposed implementation is almost same as that of the pipeline implementation based on the LMS or the NLMS algorithms and the output latency of the proposed architecture is also 0. In view of the convergence property for colored inputs, the proposed one is superior than the other algorithm.

References

[1] S. Haykin: "Introduction to Adaptive Filters", Macmillan publishing Company, New York, 1984.

[2] H.Herzberg, R.Haimi-Cohen, and Y.Be'ery: "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation", IEEE Trans. Signal Processing, vol.40, No.11, pp.2799-2803, Nov. 1992.

[3] A.Harada, K.Nishikawa and H.Kiya: "Pipelined architecture of the LMS adaptive digital filter with the minimum output latency", IEICE Trans. Fundamentals, E81-A, No.8, Aug. 1998.

[4] A.Harada, K.Nishikawa and H.Kiya: "A pipelined architecture for normalized LMS adaptive digital filters", IEICE Trans. Fundamentals, E82-A, No.2, Feb. 1999.