Two Stages Parallel LMS Structure: A Pipelined Hardware Architecture

G. Akkad[1], A. Mansour[1], B. ElHassan[2], E. Inaty[3] and R. Ayoubi[3]
[1]Lab-STICC, UMR 6285, ENSTA Bretagne, Brest, France
[2]Faculty of Engineering, Lebanese University, Tripoli, Lebanon
[3]Computer Engineering Department, University of Balamand, Koura, Lebanon
Ghattas.Akkad@ensta-bretagne.org

***Abstract*—Modern wireless communication systems have tighten the requirements of adaptive beamformers when implemented on Field Programmable Gate Array (FPGA). The set requirements imposed additional constraints such as designing a high throughput, low complexity system with fast convergence and low steady state error. Recently, a parallel multi-stage least mean square (pLMS) structure is proposed to mitigate the listed constraints. pLMS is a two stages least mean square (LMS) operating in parallel and connected by an error feedback. To form the total pLMS error, the second LMS stage (LMS_2) error is delayed by one sample and fed-back to combine with that of the first LMS stage (LMS_1). pLMS provides accelerated convergence while maintaining minimal steady state error and a computational complexity of order $O(N)$, where N represent the number of antenna elements. However, pipelining the pLMS structure is still difficult due to the LMS coefficient update loop. Thus, in this paper, we propose the application of the delay and sum relaxed look ahead technique to design a high throughput pipelined hardware architecture for the pLMS. Hence, the delayed pLMS (DpLMS) is obtained. Simulation and synthesis result, highlight the superior performance of the DpLMS in presenting a high throughput architecture while preserving accelerated convergence, low steady state error and low computational complexity. DpLMS operates at a maximum frequency of 208.33 MHz and is obtained at the cost of a marginal increase in resource requirements, i.e. additional delay registers compared to the original pLMS design.**

***Index Terms*—LMS, Parallel LMS, Relaxed Look-Ahead, FPGA, Antenna Array, Adaptive Beamforming.**

I. Introduction

To ease the spectral congestion, modern smart antennas [1] implement adaptive beamforming (ABF) techniques for directional signal transmission or reception [2]. ABF is achieved by applying a spatial filter with varying weights to the arrays output, modeled as a linear combination of the observed noisy signal. The filter weights are concurrently computed by an adaptive algorithm focusing the main beam towards a specific direction and attenuating interfering signals [2], as shown in figure 1. Modern wireless communication systems imposed challenging constraints on adaptive algorithms when implemented on hardware, i.e. Field Programmable Gate Array (FPGA). Such constraints are translated by the systems ability to accurately adapt, in real-time, to the ever changing signal condition and user mobility in a high throughput, low complexity architecture. Classical non-blind adaptive algorithms such as the least mean square (LMS) and recursive least square (RLS) iteratively minimizes the mean square error between the filtered output signal and a reference signal [4]. In contrast to the LMS with a linear computational complexity of order $O(N)$, where N represents the number of antenna elements, the RLS [5] presents an undesirable quadratic complexity of order $O(N^2)$. However the LMS still suffers from a trade-off between its convergence speed and its residual error floor [4]. Several variants of the classical LMS and RLS have been proposed to accelerate the convergence rate while trying to maintain an acceptably low error floor. However, such techniques resulted in a complex, non parallel, and difficult to be implemented in a pipeline architecture. These techniques include the cascaded least mean square-least mean square (LLMS) [4] and the recursive least mean square (RLMS) [3], [6]. The LLMS and RLMS presented a multi stage LMS-LMS and RLS-LMS structures cascaded by an estimate of the array steering vector [4], [6]. This technique shows superior performance over previous LMS variants at the cost of doubling the computational requirement of the classical LMS algorithm and introducing additional $N+1$ complex divisions. Furthermore, the cascade RLMS was simplified in [3] by eliminating the need to compute the array image cascading block, hence obtaining a parallel input RLMS structure. However, the complexity of the system remained of order $O(N^2)$ and the RLS still requires a division operation.

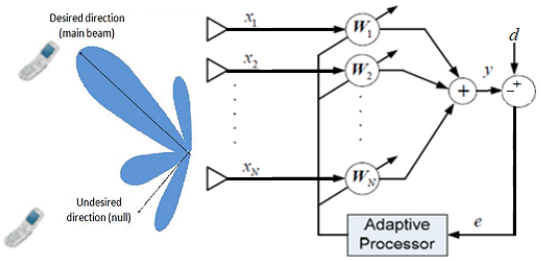

Fig. 1. Adaptive Beamforming System [3]

Thus, in this paper, we propose a pipelined delayed parallel two stages LMS structure (DpLMS) connected by error feedback [3], [4], [6], [7] for FPGA implementation. the pipelined hardware architecture for the pLMS is obtained through the application of the delay and sum relaxed look ahead technique. The proposed pLMS and DpLMS are implemented using infi-

nite and finite precision on software and FPGA, respectively, to assess their performance and resource requirements.

II. Mathematical Review

This section presents a brief background review of the LMS beamformer [8], [9] for narrow-band complex signals and a uniform linear array (ULA) of N equally spaced antenna elements [4]. Let the input vector, $\mathbf{x}(k) = [x_1(k), x_2(k), ..., x_N(k)]^T$, at the discrete time instant k to the narrow-band beamformer be defined by

$$\mathbf{x}(k) = \mathbf{a}_\mathrm{d} s_\mathrm{d}(k) + \sum_{\mathrm{l}=0}^{\mathrm{N}-1} \mathbf{i}_\mathrm{l}(k) + \mathbf{n}(k) \tag{1}$$

with $[.]^T$ is the matrix transpose, $s_d(k)$ and $\mathbf{i}_\mathrm{l}(k)$ are the desired and interfering signals with $l < N$, $\mathbf{a}_\mathrm{d}$ is the $N \times 1$ complex array steering vector for the desired signal, and $\mathbf{n}(k)$ is the complex additive white Gaussian noise (CAWGN) vector. A general form of $\mathbf{a}_d$ is given by

$$\mathbf{a}_d = [1, e^{-j\psi}, e^{-j2\psi}, ...e^{-j(N-1)\psi}]^T \tag{2}$$

where $j = \sqrt{-1}$ and

$$\psi = 2\pi \frac{B \sin(\theta)}{\lambda} \tag{3}$$

where the first antenna element acting as a reference, θ is the angle of arrival, B is the distance between consecutive antenna elements, and λ is the signal wavelength. The output of the beamformer subject to a linear combiner is given by

$$y(k) = \mathbf{w}^\mathbf{H}(k)\mathbf{x}(k) \tag{4}$$

where $[.]^H$ represents the matrix Hermitian transpose and $\mathbf{w}(k)$ is the array weight vector.

A. Least Mean Square Algorithm

The LMS algorithm is a recursive solution to the optimal Wiener filtering problem, by use of the steepest descent optimization method [10]. The LMS algorithm minimizes the mean square error (MSE) between the desired signal $\mathbf{d}(k)$ and filter output,and $y(k)$ [3]. LMS updates its weight vector $\mathbf{w}(k)$ as follows

$$e(k) = d(k) - y(k) \tag{5}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{x}(k) \tag{6}$$

where $*$ is the complex conjugate operator and μ controls the magnitude of the gradient descent steps, i.e. step size [8]. We assume the process to be wide sense stationary (WSS) and all signals are Gaussian with zero mean. Thus, the optimal weight, $\mathbf{w}_{op}$, [9] becomes

$$\mathbf{w}_{op} = \mathbf{R}^{-1}(k)\mathbf{p}(k) \tag{7}$$

Here $\mathbf{R}(k)$ and $\mathbf{R}^{-1}(k)$ are the input signals auto-correlation matrix and its inverse. $\mathbf{p}(k)$ is the cross correlation vector of the input $\mathbf{x}(k)$ and desired signal $d(k)$

$$\mathbf{R}(k) = E[\mathbf{x}(k)\mathbf{x}^\mathbf{H}(k)] \tag{8}$$

$$\mathbf{p}(k) = E[d^*(k)\mathbf{x}(k)] \tag{9}$$

Where $E[.]$ is the expectation operator.

III. Parallel LMS (pLMS)

As previously defined, pLMS is a multi stage LMS with parallel input where the overall error signal, $e_{pLMS}(k)$, is derived as a combination of individual stage errors [7] as shown in Fig. 2. As described in Fig. 2, the error signal of the

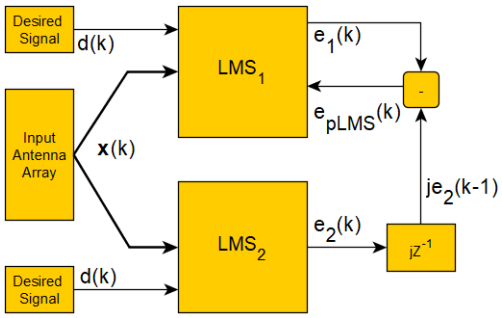

Fig. 2. pLMS Diagram [7]

LMS_2 stage, $e_2(k)$, is multiplied by j and subject to a one sample delay, represented by the block jZ^{-1}. The resulting error, $je_2(k-1)$ is then combined with that of LMS_1, consequently allowing parallel operation of both stages, hence the notation pLMS.

The pLMS algorithm is now defined by Algorithm 1 [7].

Algorithm 1 Parallel LMS (pLMS) [7]

Conditions:

$$e_2(-1) = e_2(0) = 0$$
$$d(-1) = d(0)$$
$$\mathbf{x}(-1) = \mathbf{x}(0)$$

pLMS:

LMS_1 :

$$y_{LMS1}(k) = \mathbf{w}_1^\mathbf{H}(k)\mathbf{x}(k)$$
$$e_1(k) = d(k) - y_{LMS1}(k)$$
$$e_{pLMS}(k) = e_1(k) - je_2(k-1)$$
$$\mathbf{w}_1(k+1) = \mathbf{w}_1(k) + \mu_1 e^*_{pLMS}(k)\mathbf{x}(k)$$

LMS_2 :

$$y_{LMS2}(k) = \mathbf{w}_2^\mathbf{H}(k)\mathbf{x}(k)$$
$$e_2(k) = d(k) - y_{LMS2}(k)$$
$$\mathbf{w}_2(k+1) = \mathbf{w}_2(k) + \mu_2 e^*_2(k)\mathbf{x}(k)$$

Where μ_1, $y_{LMS1}(k)$ and μ_2, $y_{LMS2}(k)$ are the first and second stage step size and output, respectively. The multiplication by the imaginary number j is used to protect against error nulls and delay in convergence caused by recurring samples and symbol re-transmission.

The overall mean square error, $\xi_{pLMS}(k)$ [7], is defined as

$$\begin{aligned}\xi_{pLMS}(k) =& E[|e_{pLMS}(k)|^2] \\ =& E[|e_1(k) - je_2(k-1)|^2] \\ =& E[|e_1(k)|^2 + je_1(k)e_2^*(k-1) \\ & - je_1^*(k)e_2(k-1) + |e_2(k-1)|^2] \end{aligned} \quad (10)$$

Where, $e_i(k) = d(k) - \mathbf{w}_i^{\mathbf{H}}(k)\mathbf{x}(k)$, i represents the stage identifier, $|.|$ signifies complex modulus and $*$ is the complex conjugate operator. Assuming both stages converge and since LMS_2 is the classical LMS its optimal weight is defined by (7). Thus, from [7], pLMS optimal weight vector, $\mathbf{w}_{opLMS}$ is

$$\begin{aligned}\mathbf{w}_{opLMS} =& \mathbf{w}_{op} + j\mathbf{R}^{-1}(k)E[d^*(k-1)\mathbf{x}(k)] \\ & - j\mathbf{R}^{-1}(k)E[\mathbf{x}(k)\mathbf{x}^{\mathbf{H}}(k-1)]\mathbf{w}_{op} \end{aligned} \quad (11)$$

Detailed algorithm derivation with stability and convergence behavior to determine the upper bound of the step size are presented in [7].

IV. Delayed pLMS Hardware Architecture

In this section, we propose the application of the delay and sum relaxed look-ahead technique to present a pipelined, high throughput pLMS hardware implementation. The previous technique is applied individually for each LMS stage i.e., LMS_1 and LMS_2 stages [6], [11]. Thus, assuming WSS process, from (6), we start with a D_2 step look-ahead [11] in the weight update path and a delay relaxation of D_1 samples in the error path. Therefore, (6) becomes

$$\begin{aligned}\mathbf{w}(k+1) =& \mathbf{w}(k - D_2) \\ & + \mu \sum_{i=0}^{D_2-1} e^*(k - D_1 - i)\mathbf{x}(k - D_1 - i) \end{aligned} \quad (12)$$

The previous relaxation is possible given the signal is WSS and the gradient estimate does not change much over D_1 samples [11]. While the previous assumption allows presenting a pipelined architecture, its hardware overhead is $N(D_2 - 1)$ and becomes unacceptable for larger values of N and D_2 [11]. To reduce the resulting overhead, a sum relaxation of D_3 terms is applied where $1 \leq D_3 \leq D_2$. Equation (12) becomes

$$\begin{aligned}\mathbf{w}(k+1) =& \mathbf{w}(k - D_2) \\ & + \mu \sum_{i=0}^{D_3-1} e^*(k - D_1 - i)\mathbf{x}(k - D_1 - i) \end{aligned} \quad (13)$$

Finally, from (5), (13), assuming μ is small enough, and $\mathbf{w}(k - D_2 - 1)$ can be approximated by $\mathbf{w}(k - D_2)$, we obtain

$$e(k) = d(k) - \mathbf{w}^{\mathbf{H}}(k - D_2)\mathbf{u}(k) \quad (14)$$

Thus, a delayed pLMS (DpLMS) structure is now obtained by applying (13) and (14) to LMS_1 and LMS_2. Therefore, DpLMS is implemented using complex arithmetic a 18bits signed fixed point $Q2, 15$ format i.e. 1 signed bit, 2 integer bits and 15 precision bits with $D_1 = 4$, $D_2 = 2$ and $D_3 = 1$, i.e. six pipeline stages for $N = 8$ antenna elements. The top level hardware architecture is presented in Fig. 2. The resulting, architecture is shown in Fig. 3, where $\mathbf{x}_{1..4}(k)$, $\mathbf{x}_{5..8}(k)$, $\mathbf{w}_{1..4}(k)$ and $\mathbf{w}_{5..8}(k)$ are the DpLMS input and weight vectors formed of the first and last 4 elements of $\mathbf{x}(k)$and $\mathbf{w}(k)$, respectively. Z^{-1}, Z^{-D_1} and Z^{-D_1-1} represent the digital delay, i.e. registers of 1, D_1, and $D_1 - 1$ samples, respectively. The $*j$ and Conj block denotes multiplication by j and complex conjugation, respectively.

The linear combiner and weight update blocks are defined in Fig. 4 and Fig. 5, respectively.

From Fig. 4, we can notice that the multiplication and addition stages requires one clock cycle, each, for all parallel inputs with a computational complexity of $O(1)$, respectively. Each complex multiplier is formed of four real multipliers and one complex adder, i.e. it is equivalent two real adders. Additionally, from Fig. 5, the update term is obtained by a right shift of 5 bits i.e. $\mu = 2^{-5} = 0.313$ omitting the need for an additional multiplier. Both pipeline stages perform parallel operations and have a computational complexity of order $O(1)$.

Synthesis results of the delayed LMS (DLMS), pLMS and DpLMS are obtained for the Intel Stratix V 5SGXMABN3F45I4 model and shown in Table I. The look-up tables (LUTs) represents logic cells and a digital signal processing (DSP) block is formed of transistor level multipliers, adders and registers.

As presented in Table I, DpLMS presents a pipelined structure operating at a maximum frequency of 208.33 MHz.

Design	Clock (MHz)	LUTs	DSP Blocks	Registers
DLMS	208.33	773	32	1746
pLMS	208.33	1546	64	3488
DpLMS	208.33	1567	64	3636

TABLE I
8-Input DpLMS Beamformer Synthesis Results

In contrast to the pLMS, with the same operating frequency, the DpLMS presents a pipelined high throughput structure at the cost of a negligible increase in the resource usage, i.e. LUTs and registers. DpLMS can process one sample each $4.8ns$, i.e. one clock period compared to the original pLMS with one sample each $(D_1 + D_3) \times 4.8ns$, i.e. $28.8ns$.

V. Simulation Results and Discussion

A simulation of 500 realizations with 500 samples each is conducted for a uniform linear antenna array with $N = 8$ elements. The input signal is formed by a message signal and two interferes arriving at an angle of 20^o, 5^o and 65^o, respectively. The generated inputs are independent random complex Gaussian sequences of the form $v = \mathcal{N}(0, \sigma_r^2) + j\mathcal{N}(0, \sigma_c^2)$ where, $\mathcal{N}$ denotes normal, 0 mean, (Gaussian) distribution. σ_r^2 and σ_c^2 are the real and complex sequence variances, respectively. The final input signal is corrupted by CAWGN with a $SNR = 10dB$. The parameters and initial conditions at $k = 0$ are given as, $\mu_1 = \mu_2 = 0.03$, $d(-1) = d(0)$, $\mathbf{x}(-1) = \mathbf{x}(0)$, $\sigma_r^2 = 0.01$, $\sigma_c^2 = 0.04$. As for the RLS the initial parameters are the forgetting factor $\alpha = 0.99$ and initial matrix $\mathbf{P}(0) = 0.5I$, where I is an $N \times N$ identity matrix.

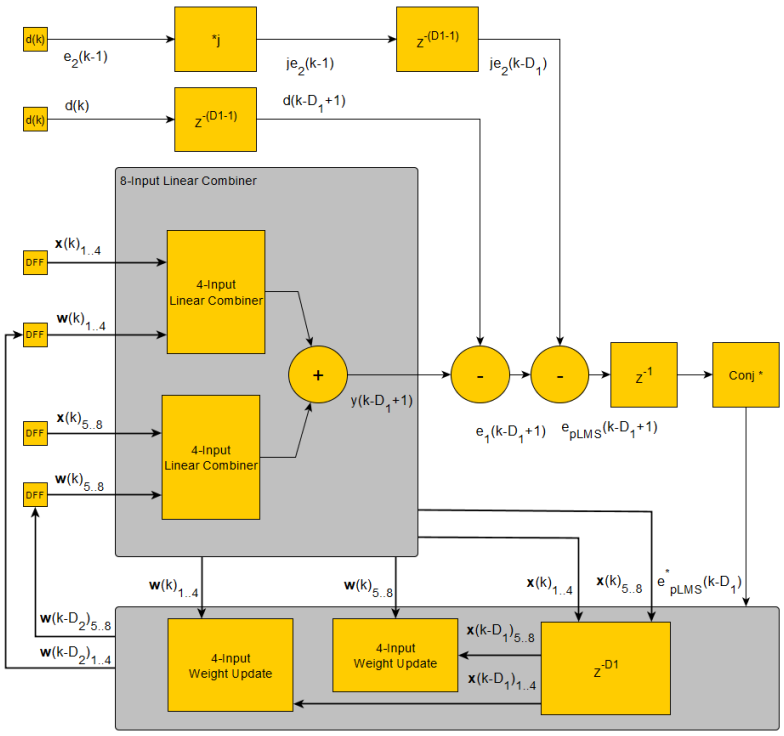

Fig. 3. 8-Elements DpLMS Hardware Architecture

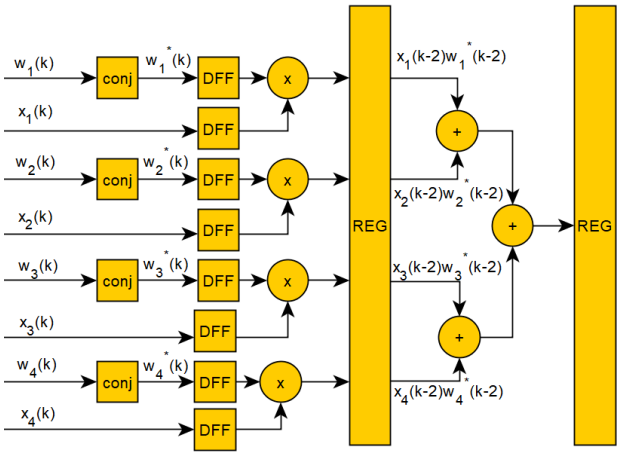

Fig. 4. 4-Input Linear Combiner Block

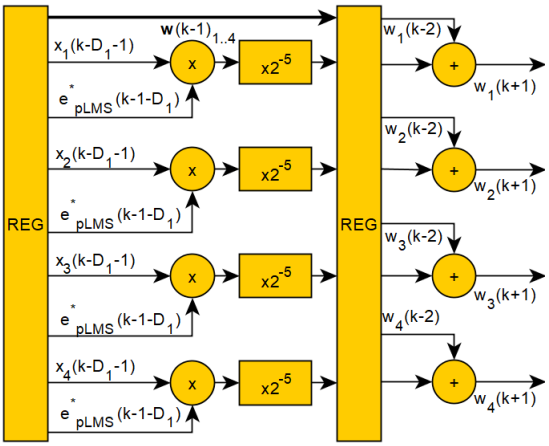

Fig. 5. 4-Input Weight Update Block

Simulation results describing the mean square error (MSE) convergence behavior are presented in Fig. 6 for the LLMS, pLMS, the DpLMS implemented in 18bits precision, RLS and LMS. As shown in Fig. 6 compared to LLMS, pLMS achieved the same accelerated convergence behavior with a lower residual error at steady state for a lower complexity architecture. Additionally, compared to the RLS with $O(N^2)$ complexity, the pLMS achieved accelerated and accurate behavior with a complexity of order $O(N)$. To better assess the behavior of the system and the effect of the sum relaxation, the beam pattern is simulated for the DpLMS with a SNR ranging from $1dB$ to $7dB$ with a step of $3dB$. The DpLMS resulting beam pattern are simulated for DpLMS . From Fig. 7 the DpLMS achieves a satisfactory behavior for SNR environments up to only $1dB$. Such degradation is a consequence of the sum relaxation adopted in (13) and can be omitted for large values of D_3, i.e. a high order moving average filter. However, as D_3 increase the hardware overhead increases. Moreover, Fig. 8 shows the finite and infinite precision beam radiation pattern.

Fig. 8 shows that the 18bits finite precision DpLMS, simulated on FPGA, presented the same radiation pattern and beam pointing accuracy as the infinite precision pLMS and DpLMS.

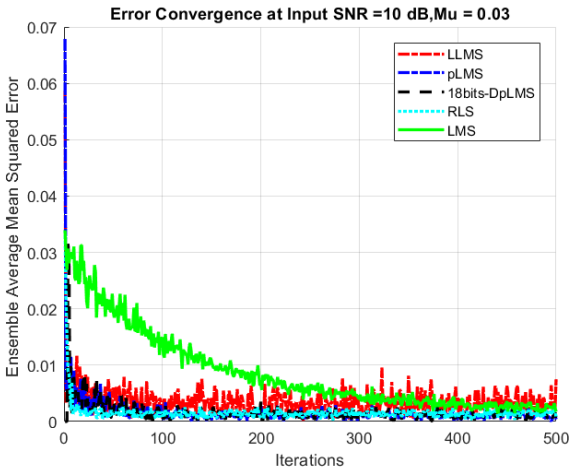

Fig. 6. pLMS MSE Convergence Behavior

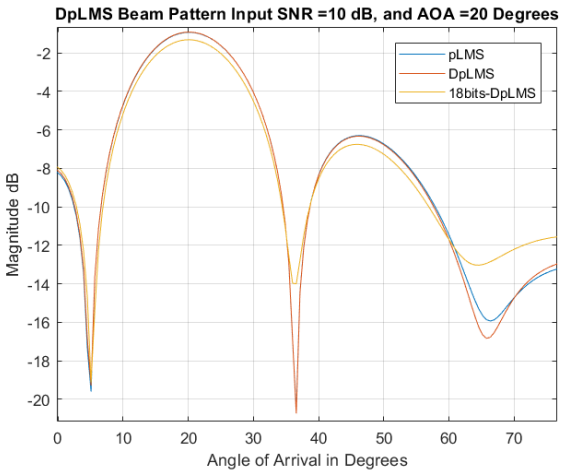

Fig. 8. DpLMS Finite and Infinite Precision Beam Radiation Pattern

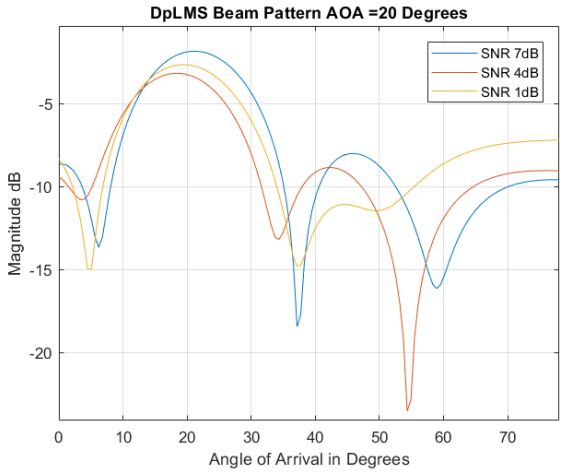

Fig. 7. DpLMS Beam Pattern for Different SNR

VI. Conclusion and Future Work

In this paper, we proposed a pipelined multi stage parallel least mean square (pLMS) structure for adaptive beamforming. pLMS is achieved by two least mean square (LMS) stages with parallel inputs connected by an error feedback. In order to present a pipelined hardware architecture and overcome the difficulties set forth by the coefficient update loop, we applied the delay and sum relaxed look-ahead technique. As such a delayed parallel least mean square (DpLMS) is obtained. Simulation and synthesis results have shown outstanding performance for the DpLMS presented by its accelerated convergence, low residual error and low computational complexity in finite precision arithmetic. Additionally, DpLMS presented the same beam radiation pattern as the infinite precision pLMS with a minor increase in resource requirements. Thus, future work includes performing stability and quantization effect analysis to determine the upper bound for the step size.

Acknowledgment

The authors are grateful to AID - DGA (l'Agence de l'Innovation de Défense à la Direction Générale de l'Armement – Minitère des Armées) & ANR (Agence Nationale de le Recherche en France) for supporting our ANR-ASTRID – Project (ANR-19-ASTR-0005-03).

References

[1] L. Xing, J. Zhu, Q. Xu, D. Yan, and Y. Zhao, "A circular beam-steering antenna with parasitic water reflectors," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, pp. 2140–2144, Oct 2019.

[2] A. Mansour, R. Mesleh, and M. Abaza, "New challenges in wireless and free space optical communications," *Optics and Lasers in Engineering*, vol. 89, pp. 95 – 108, 2017.

[3] G. Akkad, A. Mansour, B. ElHassan, J. Srar, M. Najem, and F. LeRoy, "Low complexity robust adaptive beamformer based on parallel rlms and kalman rlms," in *27th European Signal Processing Conference (EUSIPCO)*, (Spain), Sep 2019.

[4] J. A. Srar, K. Chung, and A. Mansour, "Adaptive array beamforming using a combined lms-lms algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 58, pp. 3545–3557, Nov 2010.

[5] Jun Ma, K. K. Parhi, and E. F. Deprettere, "Annihilation-reordering look-ahead pipelined cordic-based rls adaptive filters and their application to adaptive beamforming," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2414–2431, Aug 2000.

[6] G. Akkad, A. Mansour, B. A. ElHassan, E. Inaty, R. Ayoubi, and J. A. Srar, "A Pipelined Reduced Complexity Two-Stages Parallel LMS Structure for Adaptive Beamforming," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–13, 2020.

[7] G. Akkad, A. Mansour, B. ElHassan, E. Inaty, and R. Ayoubi, "A multi-stage parallel lms structure and its stability analysis using transfer function approximation," in *Submitted to 28th European Signal Processing Conference (EUSIPCO)*, (Amsterdam), August 2020.

[8] R. K. Sarma, M. T. Khan, R. A. Shaik, and J. Hazarika, "A novel time-shared and lut-less pipelined architecture for lms adaptive filter," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–10, 2019.

[9] W. Zhao, J. Q. Lin, S. C. Chan, and H. K. . So, "A division-free and variable-regularized lms-based generalized sidelobe canceller for adaptive beamforming and its efficient hardware realization," *IEEE Access*, vol. 6, pp. 64470–64485, 2018.

[10] S. Haykin, *Adaptive Filter Theory (5th Ed.)*. Pearson, 2013.

[11] N. R. Shanbhag and K. K. Parhi, "Relaxed look-ahead pipelined lms adaptive filters and their application to adpcm coder," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, pp. 753–766, Dec 1993.