THE GRADIENT ADAPTIVE LATTICE ALGORITHM IN BLOCK FLOATING POINT FORMAT

Mrityunjoy Chakraborty and Abhijit Mitra

Department of Electronics and Electrical Communication Engineering,
Indian Institute of Technology (I.I.T.), Kharagpur, India.
E-mail: mrityun@ece.iitkgp.ernet.in, abhijit@ece.iitkgp.ernet.in

ABSTRACT

We present a novel scheme to implement the gradient adaptive lattice (GAL) algorithm using block floating point (BFP) arithmetic that permits processing of data over a wide dynamic range, at a cost significantly less than that of a floating point processor. Appropriate formats for the input data, the prediction errors and the reflection coefficients are adopted, taking care so that for the latter, they remain invariant to the coefficient updating process. Care is also taken to prevent overflow during prediction error computation and reflection coefficient updating by using an appropriate exponent assignment algorithm and an upper bound on the step size mantissa.

Keywords – Gradient adaptive lattice, Block floating point arithmetic, Exponent assignment.

1. INTRODUCTION

The block floating point (BFP) data format is an useful compromise between the floating point (FP) and the fixed point (FxP) schemes. In this format, the incoming data is partitioned into non-overlapping blocks and depending upon the data sample with highest magnitude in each block, a common exponent is assigned for the block. This permits an overall FP like representation of the data with fixed point (FxP) like computation within every block, thereby enabling the user to handle data over a FP like wide dynamic range with temporal and/or spatial complexities comparable to that of FxP based systems. In recent years, the BFP format has been used successfully for efficient realization of several forms of digital filters ([1]-[2], [4]-[5]). Some studies ([2] -[3]) have also been made to investigate the associated numerical error behaviour. Such efforts have, however, remained confined to the case of fixed coefficient digital filters only and were not extended to include adaptive filters that present more complex structures including error feedback. A BFP treatment to adaptive filters faces certain difficulties, not encountered in the fixed coefficient case, namely, (a) unlike a fixed coefficient filter, the filter coefficients in an adaptive filter can *not* be represented in the simpler fixed point form, as the coefficients in effect evolve from the data by a time update relation, and (b) the two principal operations in an adaptive filter, viz., filtering and weight updating, are mutually coupled, thus requiring an appropriate arrangement for joint prevention of overflow.

Recently, the BFP approach has been used to obtain an efficient realization of the NLMS based transversal adaptive filter [6]. In this paper, we consider a similar BFP treatment for efficient realization of the gradient adaptive lattice filter [7]. However, as the lattice is an order recursive structure unlike its transversal counterpart, new approaches are required to achieve this. This includes adoption of appropriate formats for representing the input, the prediction errors and the reflection coefficients, with the latter chosen carefully so that it remains invariant to the coefficient updating operation. Special arrangements are also made to prevent overflow during both prediction error computation and reflection coefficient updating – the former achieved via an appropriate exponent assignment algorithm and the latter by using a scaled format for the step size, with mantissa restricted to remain below certain upper bound. The proposed scheme employs mostly FxP type operations and can achieve considerable speed up over a FP based realization.

2. THE PROPOSED IMPLEMENTATION

For a Lth order linear prediction lattice filter, the pth stage, $p \in Z_L = \{1, ..., L\}$ is characterized by the following order update equations :

$$f_p(n) = f_{p-1}(n) - k_p(n) b_{p-1}(n-1) \quad (1)$$

$$b_p(n) = b_{p-1}(n-1) - k_p(n) f_{p-1}(n) \quad (2)$$

where $f_p(n)$ and $b_p(n)$ are the pth order forward prediction error (FPE) and backward prediction error (BPE) respectively and $k_p(n)$ is the time dependent reflection coefficient corresponding to the pth stage. The reflection coefficients

are updated in time using a gradient-based approach and there exist several forms of such update relationships [8]. For our treatment, we consider the simple and also popular case of unnormalized lattice recursion, for which the time update relationship for $k_p(n)$ is given by,

$$k_p(n+1) = k_p(n) + \mu_p(f_p(n)b_{p-1}(n-1) + b_p(n)f_{p-1}(n)) \quad (3)$$

where μ_p denotes the step size for the pth stage and should be chosen sufficiently small to guarantee convergence of the algorithm [7]. The order recursions (1) and (2) are initialized with the following zeroth order FPE and BPE values : $f_0(n) = b_0(n) = x(n)$.

The proposed scheme is based on three simultaneous BFP representations - one for the reflection coefficients $k_p(n), p \in Z_L$, one for the given data $x(n)$ and another one for the FPE and the BPE, i.e., $f_p(n)$ and $b_p(n)$. These are as follows :

(a) **Format for the reflection coefficients**:

The proposed approach adopts a scaled representation for the reflection coefficients as given by

$$k_p(n) = \overline{k}_p(n).2^{\phi_p(n)} \quad (4)$$

where $\overline{k}_p(n)$ and $\phi_p(n)$ are respectively the time-varying mantissa and exponent which are updated at each index n, with the latter chosen to ensure that $|\overline{k}_p(n)| < \frac{1}{2}, n \in Z, p \in Z_L$. As shown later, $\phi_p(n)$, in our treatment, is a non-decreasing function of n. Further, the initial values of $\overline{k}_p(n)$ and $\phi_p(n)$ are taken as $\overline{k}_p(0) = 0$ and $\phi_p(0) = 1$ respectively.

(b) **BFP representation of the input data**:

In this, the input data $x(n)$ is partitioned into non-overlapping blocks of N samples each ($N \geq L$) with the ith block ($i \in Z$) consisting of $x(n)$ for $n \in Z_i' = \{iN, iN+1, ..., iN+N-1\}$. Further, the data samples of each block are scaled jointly by a common factor so as to have a uniform BFP representation. This means that, for each $n \in Z_i'$, $x(n)$ is expressed as

$$x(n) = \overline{x}(n).2^{\gamma_i} \quad (5)$$

where $\overline{x}(n)(= x(n).2^{-\gamma_i})$ represents the mantissa of $x(n)$ and γ_i is the common block exponent for the ith block, chosen so as to satisfy $\gamma_i \geq ex_i$, where $ex_i = \lfloor log_2 M_i \rfloor + 1$ and $M_i = max\{|x(n)| \mid n \in Z_i'\}$, i.e., ex_i is the exponent of the data sample with highest magnitude in the ith block. For such choice of γ_i, $|\overline{x}(n)| < 1$ for all $n \in Z_i'$. The block exponent γ_i is actually assigned as per an algorithm described later.

(c) **BFP format of the prediction errors**:

The FPE $f_p(n)$ and BPE $b_p(n)$ (for $p \in Z_L$) are expressed in the following format : $f_p(n) = \overline{f}_p(n).2^{\Gamma_{p,n}}$ and $b_p(n) = \overline{b}_p(n).2^{\Gamma_{p,n}}$ respectively, where the exponent $\Gamma_{p,n}$ is defined as

$$\Gamma_{p,n} = \gamma_i + \sum_{j=1}^{p} \phi_j(n), \quad (6)$$

and is updated order recursively as

$$\Gamma_{p+1,n} = \Gamma_{p,n} + \phi_{p+1}(n).$$

Next, from (1) and (2), it is seen that the computations at the p-th stage of the lattice involve two exponents : $\Gamma_{p-1,n}$ and $\Gamma_{p-1,n-1}$, the former coming from $f_{p-1}(n)$ and the latter from $b_{p-1}(n-1)$. In order to have a common exponent, we next rescale the delayed backward prediction error mantissas $\overline{b}_{p-1}(n-1)$, $p \in Z_L$, to

$$\tilde{b}_{p-1}(n-1) = \overline{b}_{p-1}(n-1).2^{-\alpha_{p-1,n}} \quad (7)$$

where the update factor $\alpha_{p-1,n}(= \Gamma_{p-1,n} - \Gamma_{p-1,n-1})$ is given by,

$$\alpha_{p-1,n} = \begin{cases} \theta_n + \sum_{j=1}^{p-1} \Delta\phi_j(n), & p \geq 2 \\ \theta_n, & p = 1 \end{cases} \quad (8)$$

where $\Delta\phi_j(n) = (\phi_j(n) - \phi_j(n-1))$ and θ_n takes the value $\Delta\gamma_i = \gamma_i - \gamma_{i-1}$ at $n = iN$, i.e., at the starting index of the ith block, $i \in Z$ and $zero$ otherwise. In practice, such rescaling is effected by passing each of the backward prediction error mantissas $\overline{b}_{p-1}(n-1)$, $p \in Z_L$ through a rescaling unit that applies $\alpha_{p-1,n}$ number of right or left shifts on $\overline{b}_{p-1}(n-1)$, depending on whether $\alpha_{p-1,n}$ is positive or negative respectively. The parameter $\alpha_{p-1,n}$ is again updated order recursively as,

$$\alpha_{p,n} = \alpha_{p-1,n} + \Delta\phi_p(n).$$

For the above description of $\Gamma_{p,n}$ and $\alpha_{p,n}$, the FPE and BPE mantissas can now be written as

$$\overline{f}_p(n) = \overline{f}_{p-1}(n).2^{-\phi_p(n)} - \overline{k}_p(n)\tilde{b}_{p-1}(n-1), \quad (9)$$

$$\overline{b}_p(n) = \tilde{b}_{p-1}(n-1).2^{-\phi_p(n)} - \overline{k}_p(n)\overline{f}_{p-1}(n). \quad (10)$$

Both the order update equations (9) and (10) above are based on FxP operations and thus it is required to ensure that no overflow arises during these computations. For this, we first consider $\tilde{b}_{p-1}(n-1)$ as given in (7). Noting that though in our treatment, $\Delta\phi_p(n) \geq 0$, since $\phi_p(n), p \in Z_L$ is a non-decreasing function of n as stated earlier, $\Delta\gamma_i$ can, however, be positive as well as negative and with negative $\Delta\gamma_i$, $\alpha_{p-1,n}$ in (8) can become negative at $n = iN$, thus giving rise to left shift operation on $\overline{b}_{p-1}(n-1)$ in (7) and to the possibility of overflow as a consequence. To ensure no overflow in $\tilde{b}_{p-1}(n-1)$, we need to maintain $|\tilde{b}_{p-1}(n-1)| < 1$. We meet this condition by first proposing an exponent assignment algorithm as follows :

Algorithm: *For any i-th block ($i \in Z$),*
if $ex_i \geq ex_{i-1}$,
choose $\gamma_i = ex_i$
else (i.e., $ex_i < ex_{i-1}$*)*
choose $\gamma_i = ex_{i-1}$.

Note that when $ex_i \geq ex_{i-1}$, we can either have $\gamma_{i-1} > ex_i \geq ex_{i-1}$ (Case A) implying $\Delta\gamma_i < 0$, or, $ex_i \geq \gamma_{i-1} \geq ex_{i-1}$ (Case B) meaning $\Delta\gamma_i \geq 0$. However, for $ex_i < ex_{i-1}$ (Case C), we always have $\Delta\gamma_i \leq 0$. It is, however, easy to see that for all $n \in Z'_{i-1}$, $|\overline{f}_0(n).2^{-\Delta\gamma_i}| = |\overline{b}_0(n).2^{-\Delta\gamma_i}| = |\overline{x}(n).2^{-\Delta\gamma_i}| < 1$, irrespective of whether $\Delta\gamma_i$ is positive or negative, as rescaling $\overline{f}_0(n)$ and $\overline{b}_0(n)$ by $2^{-\Delta\gamma_i}$ amounts to changing their exponent from γ_{i-1} to γ_i and from above, $\gamma_i \geq ex_{i-1}$.

Proposition 1: Given $n = iN, i \in Z$ and block length $N \geq$ filter order L, $|2^{-\Delta\gamma_i}.\overline{b}_{p-1}(n-r)| < 1, r = 1, 2, \ldots L-p+1$ and $|2^{-\Delta\gamma_i}.\overline{f}_{p-1}(n-l)| < 1, l = 1, 2, \ldots L-p+1$, for all $p \in Z_L$.

Proof: For $p = 1$, the inequalities are satisfied trivially from above. Suppose the inequalities are satisfied for p up to some $q, 1 \leq q \leq L-1$. Then, for $p = q+1$ and for $n = iN, r = 1, 2, \ldots, L-q$, we can write from (10) and (7),

$$|2^{-\Delta\gamma_i}.\overline{b}_q(n-r)| = 2^{-\Delta\gamma_i}.|(\overline{b}_{q-1}(n-r-1).2^{-\alpha_{q-1,n-r}} .2^{-\phi_q(n-r)} - \overline{k}_q(n-r)\overline{f}_{q-1}(n-r))|.$$

For the stated choice of n and r, $\theta_{n-r} = 0$ and thus $\alpha_{q-1,n-r} = \sum_{j=1}^{q-1} \Delta\phi_j(n-r) \geq 0$. Using this, applying the triangle inequality to the RHS of the above equation and making use of the assumption on $\overline{b}_{q-1}(n-r-1)$ and $\overline{f}_{q-1}(n-r)$, we have,

$$|2^{-\Delta\gamma_i}.\overline{b}_q(n-r)| < 2^{-\phi_q(n-r)} + |\overline{k}_q(n-r)| < 1,$$

since the minimum value of $\phi_q(n-r)$ is one and $|\overline{k}_q(n-r)| < \frac{1}{2}$. In a similar way, it can be shown that for $n = iN$ and $l = 1, 2, ..., L-q$, $|2^{-\Delta\gamma_i}.\overline{f}_q(n-l)| < 1$. Hence proved. ■

Proposition 2: For $p \in Z_L$, $|\overline{f}_p(n)| < 1, |\overline{b}_p(n)| < 1$ and $|\tilde{b}_{p-1}(n-1)| < 1$.

Proof: It is enough to prove the above for a block, i.e., for $n \in Z'_i, i \in Z$. We prove this by induction. Assume the following be given : (i) $|\tilde{b}_{p-1}(n-1)| < 1, p \in Z_L$ for some $n \in Z'_i$. Note from Proposition 1 that this is already satisfied for $n = iN$, since at $n = iN$, $|\tilde{b}_{p-1}(n-1)| \leq |2^{-\Delta\gamma_i}.\overline{b}_{p-1}(n-1)| < 1$; (ii) $|\overline{f}_p(n)| < 1$ for p up to some $r, 0 \leq r \leq L-1$. For $r = 0$, $|\overline{f}_0(n)| = |\overline{x}(n)| < 1$ is always satisfied. For $p = r+1$, we have, from (9), $|\overline{f}_{r+1}(n)| \leq |\overline{f}_r(n)|.2^{-\phi_{r+1}(n)} + |\overline{k}_{r+1}(n)||\tilde{b}_r(n-1)| < 1$, since, as stated earlier, $|\overline{k}_{r+1}(n)| < \frac{1}{2}$ and $\phi_{r+1}(n) \geq 1$, meaning that $2^{-\phi_{r+1}(n)} \leq \frac{1}{2}$. In a similar manner, it can be shown from (10) that $|\overline{b}_{r+1}(n)| < 1$. For the $(n+1)$th index, if $n = iN + N - 1$, i.e., $n+1$ is the starting index of the $(i+1)$th block, then, from Proposition 1, $|\tilde{b}_{p-1}(n)| < 1, p \in Z_L$. If, however, $n < iN + N - 1$, then, for $p \geq 2$, from (7) and (8), $|\tilde{b}_{p-1}(n)| = |\overline{b}_{p-1}(n)|.2^{-\sum_{j=1}^{p-1}\Delta\phi_j(n+1)} < 1$. For $p = 1$, $|\tilde{b}_{p-1}(n)| = |\overline{b}_{p-1}(n)| = |\overline{x}(n)| < 1$. Hence proved. ■

From Proposition 2, it is thus clear that there will be no overflow in $\tilde{b}_{p-1}(n)$ as computed via (7) and in $\overline{f}_p(n)$ and $\overline{b}_p(n)$ as given by (9) and (10) respectively. For the above descriptions of $f_p(n)$, $b_p(n)$, $f_{p-1}(n)$ and $b_{p-1}(n-1)$, the reflection coefficient update equation (3) can now be written as $k_p(n+1) = \overline{v}_p(n).2^{\phi_p(n)}$ where

$$\overline{v}_p(n) = \overline{k}_p(n) + \overline{\mu}_{p,n}[\overline{f}_p(n)\tilde{b}_{p-1}(n-1) + \overline{f}_{p-1}(n)\overline{b}_p(n)] \quad (11)$$

where $\overline{\mu}_{p,n} = \mu_p.2^{2\Gamma_{p-1,n}}$. In other words, the proposed approach adopts a scaled representation for μ_p with $\overline{\mu}_{p,n}$ and $-2\Gamma_{p-1,n}$ denoting respectively the values of the time dependent mantissa and exponent at the nth index.

To satisfy $|\overline{k}_p(n+1)| < \frac{1}{2}$ for $p \in Z_L$, we first limit each $\overline{v}_p(n)$ so as to satisfy $|\overline{v}_p(n)| < 1$, $p \in Z_L$. Then, if each $\overline{v}_p(n)$ happens to be lying within $\pm\frac{1}{2}$, we make the assignments: $\overline{k}_p(n+1) = \overline{v}_p(n)$, $\phi_p(n+1) = \phi_p(n)$. Otherwise, we scale down $\overline{v}_p(n)$ by 2, in which case, $\overline{k}_p(n+1) = \frac{1}{2}\overline{v}_p(n)$, $\phi_p(n+1) = \phi_p(n) + 1$. Since each $|\overline{k}_p(n)| < \frac{1}{2}$ for $p \in Z_L$, from (11), it is sufficient to have $\overline{\mu}_{p,n}[|\overline{f}_p(n)||\tilde{b}_{p-1}(n-1)| + |\overline{f}_{p-1}(n)||\overline{b}_p(n)|] \leq \frac{1}{2}$ in order to satisfy the relation $|\overline{v}_p(n)| < 1$, $p \in Z_L$. Since $\overline{f}_p(n)$, $\overline{b}_p(n)$, $\overline{f}_{p-1}(n)$ and $\tilde{b}_{p-1}(n-1)$ have magnitudes less than one, this then results in the following global upper bound: $\overline{\mu}_{p,n} \leq \frac{1}{4}, p \in Z_L$.

Next, we evaluate $\overline{\mu}_{p,n}$ time-recursively as

$$\overline{\mu}_{p,n} = \overline{\mu}_{p,n-1}.2^{2\alpha_{p-1,n}}, \quad (12)$$

which follows from noting that

$$\overline{\mu}_{p,n}.2^{-2\Gamma_{p-1,n}} = \mu_p = \overline{\mu}_{p,n-1}.2^{-2\Gamma_{p-1,n-1}}$$

and also that $\Gamma_{p,n} = \Gamma_{p,n-1} + \alpha_{p,n}$. Finally, to ensure no overflow in $\overline{\mu}_{p,n}$ due to left shift of $\overline{\mu}_{p,n-1}$ and also to prevent stalling of the weight updating process that may occur when $\overline{\mu}_{p,n}$ becomes zero due to right shift of $\overline{\mu}_{p,n-1}$ under finite precision, an appropriately long register may be used to represent $\overline{\mu}_{p,n}$ and the initial value of $\overline{\mu}_{p,n}$ may be assigned by placing a binary 1 in the central bit location with other bits assigned binary 0.

It is also easily seen from above that $\phi_p(n)$ is a non-decreasing function of n and it saturates at a steady state value Φ_p, given by the peak value K_p of the $|k_p(n)|$-vs-n

trajectory as $\Phi_p = \lceil log_2 K_p \rceil + 1, K_p = max\{|k_p(n)| \; | n = 0, 1, 2, ...\}$. This implies that in the steady state, at all indices except for the starting index of each block, $\alpha_{p,n} = 0, 0 \leq p \leq L$ and thus the two operations, namely, (i) updating of $\Gamma_{p,n-1}$, $\alpha_{p-1,n}$ and $\overline{\mu}_{p,n-1}$, and (ii) rescaling of $\overline{b}_{p-1}(n-1)$ are no longer required.

3. DISCUSSIONS AND CONCLUSIONS

The BFP-based implementation of the GAL algorithm, as proposed above, relies mostly on FxP arithmetic and thus enjoys less processing time than its FP-based counterpart. For example, to compute the FPE and the BPE for all the L stages in the steady state (i.e., after $\phi_p(n), p \in Z_L$ attain their saturation values), the proposed scheme requires a total of $2L$ "Multiply and Accumulate (MAC)" operations (FxP) and $2L$ shifts (assuming availability of single cycle barrel shifters). [Add to this an additional L shifts and L exponent additions that may be necessary at the starting index of each block.] In contrast, in a FP-based realization, this would require the following additional operations: (a) $2L$ exponent comparisons, (b) $2L$ shifts, and (c) $4L$ exponent additions. Similar advantages exist in reflection coefficient updating also. In both cases, the number of additional operations required under FP-based realization increases linearly with the order L of the lattice. A comparative study of the two approaches reveals that given a fixed point processor with single cycle MAC and barrel shifter units, the proposed scheme is about *four times faster* than a FP-based implementation. The storage requirement in the FP scheme is also higher as it needs to store several exponent values at each index.

The proposed algorithm has also been simulated in finite precision to examine the effects on convergence behaviour. A second order autoregressive signal $x(n)$ was generated by using the following model : $x(n) = 1.55x(n-1) - 0.8x(n-2) + z(n)$, with a zero-mean, white input $z(n)$ of variance 0.02. Block formatting of $x(n)$ was carried out for a block length of 10. The signed mantissas for each data sample within every block and also for each reflection coefficient, FPE and BPE were represented using 12 (11+1) bits, while 4 (3+1) bits were allocated to represent the respective exponents each. The algorithm was simulated with $L = 2$ and $\mu_p = 2^{-6}, p = 1, 2$ which also provides the initial values for $\overline{\mu}_{1,n}$ and $\overline{\mu}_{2,n}$. The simulation results are displayed by plotting $k_1(n)$ and $k_2(n)$ against n and are shown in Fig. 1. Clearly, the reflection coefficients converge after about 300-500 samples which is very much comparable with the case of an infinite precision based realization (not shown here). Also note that both $k_1(n)$ and $k_2(n)$ have magnitudes less than one right from $n = 0$ onwards and thus $\phi_p(n), p = 1, 2$ saturates at $\Phi_p = 1$ at $n = 0$ itself.

4. REFERENCES

[1] K. R. Ralev and P. H. Bauer, "Realization of Block Floating Point Digital Filters and Application to Block Implementations," *IEEE Trans. Signal Processing*, vol. 47, no. 4, pp. 1076-1086, April 1999.

[2] K. Kalliojärvi and J. Astola, "Roundoff Errors in Block-Floating-Point Systems," *IEEE Trans. Signal Processing*, vol. 44, no. 4, pp. 783-790, April 1996.

[3] P. H. Bauer, "Absolute Error Bounds for Block Floating Point Direct form Digital Filters," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1994-1996, Aug. 1995.

[4] D. Williamson, S. Sridharan and P. G. McCrea, "A new approach to block floating point arithmetic in recursive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 719-722, July 1985.

[5] F. J. Taylor, "Block Floating Point Distributed Filters," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 300-304, Mar. 1984.

[6] A. Mitra and M. Chakraborty, "Realization of the NLMS based Transversal Adaptive Filter Using Block Floating Point Arithmetic," in *Proc. 2003 IEEE Int. Symp. Circuits, Syst. (ISCAS)*, Bangkok, May 2003, pp. IV452-IV455; also to appear in *IEEE Signal Processing Letters*, March 2004.

[7] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[8] Y. Iiguni, H. Sakai and H. Tokumaru, "Convergence Properties of Simplified Gradient Adaptive Lattice Algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1427-1434, Dec. 1985.

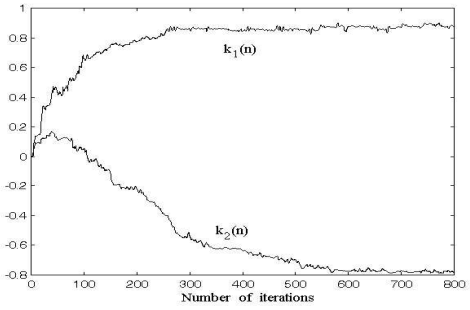

Fig. 1. Convergence results for $k_1(n)$ and $k_2(n)$ when implemented under the precision of 12 bits for mantissa and 4 bits for exponent with $\mu_p = 2^{-6}$.