Contents lists available at ScienceDirect
Digital Signal Processing
www.elsevier.com/locate/dsp

A comparative survey of fast affine projection algorithms

Feiran Yang [a], Jun Yang [a,b,*]

[a] Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Adaptive filter
Affine projection
Low complexity
Matrix inversion

ABSTRACT

The affine projection (AP) algorithm is one of the most celebrated adaptive filtering algorithms, and it achieves a good tradeoff between the convergence rate and computational complexity. However, the complexity of the AP algorithm increases with the projection order. A wealth of fast AP algorithms have been proposed to reduce the complexity in the last two decades. However, those low-complexity methods have not been well analyzed and compared. To fill this gap, this paper reviews the fast AP algorithms, including both fast filtering approaches and efficient solutions of the linear system of equations. The advantages and disadvantages of each fast implementation version are clarified based on an extensive performance evaluation, which could be useful to engineers when selecting a suitable algorithm for their specific applications and could also be a starting point for experts in this field to develop better solutions.

1. Introduction

Since the least-mean-square (LMS) algorithm was invented by Widrow and Hoff [1], many adaptive filters have been developed and widely used in many applications [2–5]. A good adaptive filtering algorithm should have a fast initial convergence rate, low steady-state misalignment, good tracking capability, robustness to disturbances, low latency, low complexity, and good numerical stability. It is difficult to design one algorithm that attains all of those features. The classic time-domain adaptive algorithms, such as the LMS, the affine projection (AP), and the recursive least squares (RLS) algorithms, are generally applied in adaptive filters. The LMS algorithm and its normalized version, i.e., the normalized LMS (NLMS) algorithm, have low complexity, but the LMS algorithm suffers from slow convergence for highly correlated signals. In contrast, the RLS algorithm exhibits very good convergence performance, but its complexity is rather high for long adaptive filters. For instance, the most efficient version of the RLS algorithm, i.e., the stabilized fast transversal filter (FTF) algorithm, still requires $O(8L)$ operations (L being the length of the filter), and it suffers from numerical difficulties [3,5]. The AP algorithm [6–8] achieves a good tradeoff between the computational complexity and the convergence performance. The convergence rate of the AP algorithm is much faster than that of the LMS algorithm especially for correlated inputs,[1] and the complexity of the fast AP algorithm is only slightly higher than that of the LMS algorithm and is much lower than that of the RLS algorithm. Thus, the AP algorithm provides a link between the computationally intensive algorithms (e.g., the RLS algorithm) and the simple LMS algorithm.

Due to its good convergence and medium complexity, the AP algorithm has been widely used in various applications, such as echo cancellation [10–12], active noise control (ANC) [13–16], noise reduction [17], system identification [18,19], beamformer [20–22], acoustic feedback cancellation [23–25], among others. The statistical convergence behavior of the AP algorithm was extensively studied in [26–38]. A good overview of the AP algorithm can be found in [39].

However, the complexity of direct implementation of the AP algorithm increases as the projection order increases, and hence, its complexity is still prohibitive for a large projection order and a long adaptive filter. The complexity of the AP algorithm stems from three operations, i.e., calculation of the error vector, update of the weight vector, and the matrix inversion operation. In the last two decades, many efforts have been made to reduce the complexity of the AP algorithm, and a wealth of fast AP algorithms have been developed, which mainly resort to the time-shift properties of the input signal and the input signal matrix.

* Corresponding author at: Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China.
E-mail address: jyang@mail.ioa.ac.cn (J. Yang).

https://doi.org/10.1016/j.dsp.2018.09.004
1051-2004/

[1] As one of the reviewers mentioned that the convergence gain of the AP algorithm over the LMS/NLMS is not very impressive for a white Gaussian noise as input. For instance, it was found in [9] that the NLMS and AP have almost the same convergence rate when the input is white noise.

The first approach toward the complexity reduction of the AP algorithm is to update an auxiliary weight vector instead of the true weight vector [40–49]. By exploiting the time-shift structure of the input signal matrix, the adaptive weight vector can be expressed as the sum of an auxiliary weight vector and a matrix-vector product. If only the error signal is required, as in some applications such as echo control and ANC, it is sufficient to update the auxiliary weight vector with a complexity of $O(L)$ multiplications per sample. Fast filtering approaches have been presented in the literature [40,41,50,51]. An approximate filtering approach was presented in [40,41], where only the first element of the error vector is exactly computed, and the others are approximated using the previous error vector. The calculation of the error vector requires only L multiplications per sample instead of the needed PL multiplication in the original AP algorithm, where P stands for the projection order. Because the fast approximate filtering approach is based on an implicit "small regularization parameter" assumption [41], it is not exactly equal to the standard AP algorithm. To address this problem, a fast exact filtering approach to the AP algorithm was then presented in [50,51], where the calculation of the filtering-out vector requires $L + P^2$ multiplications per sample. Without considering the matrix inversion operation, the fast implementations of the AP algorithm can be classified into four categories:

- Fast approximate filtering with update of the auxiliary weight vector [40,41], which is called the fast affine projection (FAP),[2] requiring $2L$ multiplications per sample.
- Fast approximate filtering with update of the original weight vector, requiring $(P + 1)L$ multiplications per sample.
- Fast exact filtering with update of the auxiliary weight vector [51], requiring $2L$ multiplications per sample.
- Fast exact filtering with update of the original weight vector, requiring $(P + 1)L$ multiplications per sample [50].

When the step size is close to one, the error vector in the FAP algorithm can be approximated by a vector in which only the first entry is non-zero, namely, the simplified FAP (SFAP) algorithm [52–55]. The complexity in the SFAP algorithm can then be further reduced compared to the original FAP. Additionally, the pseudo AP (PAP) algorithm [56–69] will be briefly reviewed, although it is not a fast version of the AP algorithm; it is simply a variation of the AP algorithm. Specifically, we will prove that three versions of the PAP update equation in [56,58] and [60] are indeed mathematically equivalent.

The accuracy of the matrix inversion has a large effect on the convergence performance of the AP algorithm, and, indeed, it affects the stability of the entire algorithm. Four types of matrix inversions have been presented in the literature. The first type presents an exact solution of the matrix inversion, e.g., the LDL^T factorization [70,71], the displacement method [72–74], the RLS algorithm [75–79], the fast RLS (FRLS) algorithm [40,41,82,83], and Schur complement-based algorithm [80,81]. The second type approximates the correlation matrix as a Toeplitz matrix and then several fast algorithms, e.g., the Levinson recursion [84], the Durbin algorithm [85] and the Ratchet algorithm [86,87] can be adopted. However, it was observed in [88] that the condition number of the Toeplitz matrix may be much larger than that of the original correlation matrix, which leads to instability. Motivated by the schemes used in speech coders [89–92], a modified Toeplitz matrix structure was presented in [88] to improve the stability of the matrix inversion algorithm. The third type includes several iterative methods, such as the Gauss–Seidel (GS) [93–96], conjugate gradient (CG) [97–99] and the dichotomous coordinate descent (DCD) [100–107] algorithms. Additionally, an approximate matrix inversion method was presented using orthogonal transforms [108], and the discrete cosine transform (DCT) transform was explored [109–113]. The step size and the regularization parameter in the AP algorithm provide a tradeoff between the convergence rate and steady-state misalignment. A formula for choosing the constant regularization parameter of the AP algorithm was presented in [114–116], which is related to the signal-to-noise ratio (SNR). Many variable step-size and variable regularization-parameter AP algorithms have been developed [117–122]. The aforementioned fast algorithms hold for a variable step size, but not all fast implementation versions can adopt a variable regularization parameter. Various evolving-order AP algorithms have also been presented to achieve both fast convergence speed and small steady-state misalignment [123–129].

Block implementations of the AP algorithm were presented in [48,130–132]. A fast exact block implementation of the fast AP algorithm was proposed in [131], although this block version is not exactly equal to the standard AP algorithm because it is based on the fast approximate filtering scheme. Subsequently, a fast exact block version of the standard AP algorithm was presented in [132], which has a complexity that is comparable to the block version in [131]. The complexities of the two block versions could be reduced by employing the fast FIR filtering approach [133,134]. In this paper, however, we adopt the scheme in [48] to describe the block exact FAP algorithm because we found that this approach can be treated as a straightforward extension of the sample-by-sample FAP algorithm to the block case, and it is easy to follow. We then extend this approach to the fast exact filtering method by updating the auxiliary weight vector [51], which leads to a block fast exact AP algorithm that is equivalent to the method in [132]. The subband adaptive filtering technique has been widely used to reduce the complexity and improve the convergence rate [135]. The AP algorithm can be used in an individual subband to further whiten the received input signal with a relatively small projection order [136,137], which is promising in practice.

However, all of the aforementioned fast AP algorithms are scattered throughout the literature and have not been comprehensively compared and analyzed. Hence, engineers may still not know which fast version is optimal for their specific applications. We believe that the field has reached a maturity that allows us to write such a review on the low-complexity implementation of the AP algorithm. In this paper, we first present a thorough overview of all the fast implementations of the AP algorithm. We then evaluate the complexity and performance of each fast version and indicate their advantages and limitations. Necessary derivations are provided to make the paper logical and systematic so that both engineers and experts in this field can benefit from this contribution.

Throughout this paper, matrices and vectors are denoted by uppercase and lowercase bold fonts, respectively, e.g., $\mathbf{R}$ and $\mathbf{r}$, and scalars are denoted by italics. All vectors are defined as column vectors; row vectors are represented by transposition. The lowercase letter denotes the time index, which is placed in parentheses. The elements of the matrix and vector are denoted as $[\mathbf{R}]_{i,j}$ and $[\mathbf{r}]_i$. The p-th column of $\mathbf{R}$ is denoted as $\mathbf{R}^{(p)}$. Superscript T denotes the transpose operator, and $\mathcal{E}\{\cdot\}$ denotes the mathematical expectation. The notation diag$\{\cdot\}$ is used to form a diagonal matrix of the augment of the operator, and $\|\cdot\|$ denotes the Euclidean norm. The notation $\mathbf{I}$ denotes the identity matrix, and $\mathbf{0}$ stands for the zero matrix or the all-zero vector.

[2] Exactly speaking, the term 'FAP' can be used to describe any fast implementations of the AP algorithm. However, because FAP has been specifically associated with the fast approximate filtering approach without an explicit update of the weight vector in [41] and was followed in the literature, we use FAP to denote the fast algorithm in [40,41] and its variations for consistency.

Please cite this article in press as: F. Yang, J. Yang, A comparative survey of fast affine projection algorithms, Digit. Signal Process. (2018), https://doi.org/10.1016/j.dsp.2018.09.004