AFFINE-PROJECTION LEAST-MEAN-MAGNITUDE-PHASE ALGORITHMS USING A POSTERIORI UPDATES

Scott C. Douglas

Southern Methodist University
Department of Electrical Engineering
Dallas, Texas 75275 USA
sdouglas@smu.edu

Danilo P. Mandic

Imperial College London
Department of Electrical Engineering
London, SW7 2BT United Kingdom
d.mandic@imperial.ac.uk

ABSTRACT

The least-mean-magnitude-phase (LMMP) algorithm is useful for complex-valued signal processing applications where precise control of magnitude and/or phase error information can provide improved estimation performance. Because it is a gradient procedure, however, the convergence speed of the algorithm can be limited for correlated input signals. In this paper, we derive affine-projection least-mean-magnitude-phase (AP-LMMP) algorithms based on an *a posteriori* update relation that have improved convergence performance over that of the LMMP algorithm without significant increases in complexity. We employ different nonlinear lookahead approaches depending on the projection order to compute the magnitudes of the *a posteriori* output signals and use these to implement the coefficient updates. Simulations indicate that AP-LMMP algorithms can outperform other algorithms in situations where their use is appropriate.

Index Terms— Adaptive algorithms, adaptive filters, adaptive signal processing, adaptive systems, algorithm design and analysis.

1. INTRODUCTION

The complex least-mean-square (CLMS) algorithm extends the well-known LMS adaptive algorithm to complex-valued signals [1]. It is used in applications where algorithm simplicity is paramount and complex-valued data representations are relevant, such as in communications and array processing. In the CLMS algorithm, a single step size parameter μ is used to control the algorithm's performance. There are a number of practical situations where the CLMS algorithm is unable to adequately account for the uncertainties associated with the input and desired response signals. For example, carrier offset in communications tasks leads to phase uncertainties that can hamper the CLMS algorithm's performance [2, 3]. In [4], the least-mean-magnitude-phase (LMMP) algorithm was devised to address such situations. The LMMP algorithm is given by

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_{p,k}\left(d_k - \frac{|d_k|}{|y_k|}y_k\right)\mathbf{x}_k^* + \mu_{m,k}\left(\frac{|d_k|}{|y_k|}y_k - y_k\right)\mathbf{x}_k^* \quad (1)$$

$$y_k = \mathbf{w}_k^T\mathbf{x}_k, \quad (2)$$

where the vector $\mathbf{w}_k$ contains the L adaptive coefficients at time k, $\mathbf{w}_{k+1}$ is the updated coefficient vector, d_k is the desired signal sample, and $\mathbf{x}_k$ is the L-element input signal vector. In the algorithm, $\mu_{m,k}$ and $\mu_{p,k}$ are the magnitude and phase step sizes, respectively, at time k. By tuning $\mu_{m,k}$ and $\mu_{p,k}$, the LMMP algorithm can provide better performance than CLMS by appropriately emphasizing either the amplitude error or phase error of the desired signal within the coefficient updates. A stability analysis of the normalized version of the above algorithm was recently provided in [5].

As both CLMS and LMMP are gradient-based, they can suffer from poor performance when the input signal elements in $\mathbf{x}_k$ are correlated. Hence, it is desirable to develop modifications to address this input signal correlation structure. In [6], extensions of the CLMS algorithm were developed that can be viewed as complex-valued extensions of the affine projection algorithm for real-valued signals [7]–[11]. Affine projection algorithms employ N constraints on the *a posteriori* output errors, where $N \ll L$, and result in algorithms whose computational complexities are approximately $2NL$ or less, depending on the shift-input structure of $\mathbf{x}_k$. While useful, the algorithms in [6] do not directly control the amplitude error or phase error as does the LMMP algorithm. Affine projection extensions of the LMMP algorithm have not been explored in the literature.

In this paper, we derive extensions of the LMMP algorithm that employ *a posterori* output signals within the coefficient updates themselves [12]–[15]. These algorithms can be viewed as extensions of the LMMP algorithm to the afffine projection algorithm class, with the additional constraint that the algorithms are implicitly regularized. The resulting algorithms have two step size parameters α and β that are easily understood given their relation to LMMP's step size parameters. Moreover, because they employ *a posteriori* updates, they are robust for a wide range of parameter choices. The algorithms employ novel lookahead mechanisms that enable their simple implementations despite the nonlinearity of the coefficient updates, resulting in different updates depending on whether $N = 1$ or $N > 1$. Simulations in an array processing example show the efficacies of the approaches.

2. AFFINE PROJECTION LMMP ALGORITHM: SINGLE DIMENSIONAL CASE

We first derive the affine projection LMMP algorithm based on an *a posteriori* update for $N = 1$ corresponding to a single constraint on the *a posteriori* output at time k. The *a posteriori* version of the LMMP algorithm is defined as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_p \left(d_k - \frac{|d_k|}{|y_{p,k}|} y_{p,k} \right) \mathbf{x}_k^* + \mu_m \left(\frac{|d_k|}{|y_{p,k}|} y_{p,k} - y_{p,k} \right) \mathbf{x}_k^* \quad (3)$$

$$y_{p,k} = \mathbf{w}_{k+1}^T \mathbf{x}_k, \quad (4)$$

where $y_{p,k}$ is the *a posteriori* output that depends on the current input signal vector $\mathbf{x}_k$ and the updated coefficient vector $\mathbf{w}_{k+1}$. Note that the above relation is not an update *per se,* as the right-hand-side requires the updated coefficient vector which is not generally available prior to the update. We can develop a mathematically-equivalent implementation of the above relation that allows $\mathbf{w}_k$ to be updated at each k. This realizable version uses the update relation to develop a lookahead value for the magnitude of $y_{p,k}$, from which an update relation can be derived. This version of the algorithm has a remapped form of the step size parameters, defined by

$$\beta = \frac{1}{\mu_m} \quad \text{and} \quad \alpha = \frac{\mu_p}{\mu_m} = \mu_p \beta. \quad (5)$$

With these choices, we can rewrite the *a posteriori* update as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha}{\beta} \left(d_k - \frac{|d_k|}{|y_{p,k}|} y_{p,k} \right) \mathbf{x}_k^* + \frac{1}{\beta} \left(\frac{|d_k|}{|y_{p,k}|} y_{p,k} - y_{p,k} \right) \mathbf{x}_k^*. \quad (6)$$

To determine this implementation, we first pre-multiply both sides of (6) by $\mathbf{x}^T$ to obtain

$$y_{p,k} = y_k + \left(\frac{\alpha}{\beta} d_k + \left(\frac{1-\alpha}{\beta} \right) \frac{|d_k|}{|y_{p,k}|} y_{p,k} - \frac{1}{\beta} y_{p,k} \right) ||\mathbf{x}_k||^2 \quad (7)$$

Moving the terms that depend on $y_{p,k}$ to the left-hand side, followed by taking absolute values of both sides, yields

$$\left| \left(1 + \frac{||\mathbf{x}_k||^2}{\beta} \right) - \left(\frac{1-\alpha}{\beta} \right) \frac{|d_k|}{|y_{p,k}|} ||\mathbf{x}_k||^2 \right| |y_{p,k}| = \left| y_k + \frac{\alpha}{\beta} d_k ||\mathbf{x}_k||^2 \right|. \quad (8)$$

Now, if

$$\beta > \left((1-\alpha) \frac{|d_k|}{|y_{p,k}|} - 1 \right) ||\mathbf{x}_k||^2, \quad (9)$$

then, we can simplify the above relation to

$$|y_{p,k}| = \frac{|\beta y_k + \alpha d_k ||\mathbf{x}_k||^2| + (1-\alpha)|d_k| \cdot ||\mathbf{x}_k||^2}{\beta + ||\mathbf{x}_k||^2} \quad (10)$$

Eq. (10) gives us a way to compute $|y_{p,k}|$ from y_k without having to compute the updated coefficient vector $\mathbf{w}_{k+1}$.

Now, we rewrite the update relation in (6) as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha}{\beta} d_k \mathbf{x}_k^* + \frac{1}{\beta} \left((1-\alpha) \frac{|d_k|}{|y_{p,k}|} - 1 \right) \mathbf{x}_k^* \mathbf{x}_k^T \mathbf{w}_{k+1}. \quad (11)$$

Moving the terms that depend on $\mathbf{w}_{k+1}$ to the left-hand side,

$$\left[\mathbf{I} - \frac{1}{\beta} \left((1-\alpha) \frac{|d_k|}{|y_{p,k}|} - 1 \right) \mathbf{x}_k^* \mathbf{x}_k^T \right] \mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha}{\beta} d_k \mathbf{x}_k^*. \quad (12)$$

Finally, we apply the matrix inversion lemma to the matrix in large brackets on the left-hand side of (12) to invert it on both sides of (12). After simplification, the final update is

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha \left(d_k - \frac{|d_k|}{|y_{p,k}|} y_k \right) + \left(\frac{|d_k|}{|y_{p,k}|} - 1 \right) y_k}{\beta + \left(1 + (\alpha - 1) \frac{|d_k|}{|y_{p,k}|} \right) ||\mathbf{x}_k||^2} \mathbf{x}_k^*. \quad (13)$$

Equations (2), (10), and (13) prove the implementation of the *a posteriori* relation in (3).

Remark 1: If the values of $|y_k|$ and $|y_{p,k}|$ are not too different, then the *a posteriori* LMMP algorithm is similar to the original LMMP algorithm with time-varying step sizes

$$\mu_{p,k} = \frac{\alpha}{\beta + \left(1 + (\alpha - 1) \frac{|d_k|}{|y_k|} \right) ||\mathbf{x}_k||^2} \quad (14)$$

$$\mu_{m,k} = \frac{1}{\beta + \left(1 + (\alpha - 1) \frac{|d_k|}{|y_k|} \right) ||\mathbf{x}_k||^2} \quad (15)$$

Thus, the new algorithm has a similar form to that of the original version.

Remark 2: We can view β as a type of relaxation parameter that is similar to regularization in other gradient-type algorithms. Values of β close to zero yield faster convergence of the magnitude error. As β becomes large, $|y_{p,k}|$ approaches $|y_k|$ such that there is less distance between the *a priori* and *a posteriori* outputs. Moreover, it has been found via simulation that the ratio β/α controls the steady-state misadjustment in situations where the system can model the desired signal accurately. From numerical studies, we have found a good relationship between β and α in such situations to be

$$\frac{\beta}{\alpha} \approx NE\{||\mathbf{x}_k||^2\} \quad (16)$$

where $N = 1$ for the single-dimensional projection update.

Remark 3: The parameter α determines the relative influence of the magnitude error versus the phase error in the AP-LMMP update. As α is defined as a ratio of step sizes, it is dimensionless, and its value should be considered along an exponential scale. Choosing $\alpha = 0.1$ causes magnitude errors to dominate the algorithm's behavior by roughly 10:1 with respect to phase errors, whereas $\alpha = 10$ similarly causes phase errors to dominate the algorithm's behavior. The choice $\alpha = 1$ corresponds to the NLMS algorithm with regularization parameter β. The choice of α should be made with regard to the overall goal of the AP-LMMP algorithm in the chosen application, considering the tradeoffs between the relative accuracy of magnitude and phase information contained within d_k and the competing goals of fast convergence, low steady-state misadjustment, and adequate tracking in time-varying situations. Moreover, while stability of the algorithm has not been proven, we have observe robust (non-divergent) behavior in numerical simulations for all $\beta > 0$ and choices of α in the range $0.001 < \alpha < 1000$.

Remark 4: There is an implementation issue with regard to $|y_{p,k}|$ computed using (10). It turns out that the positivity of the lookahead value is not guaranteed for all d_k, y_k, $||\mathbf{x}_k||^2$, α, and β, although (a) such situations appear to occur extremely rarely in practice and (b) they do not appear to cause anomalous behavior. Even so, such situations indicate that the *a posteriori* LMMP algorithm is not well-defined at such time instants. In such cases, we resolve the issue as follows: Whenever $|y_{p,k}|$ computed by (10) is negative for that particular k, we use the approximate version of the *a posteriori* LMMP algorithm given by (1), (14), and (15), which is simply the update in (13) where $|y_k|$ replaces $|y_{p,k}|$ wherever the latter value appears. This approach preserves the general form of the update without ignoring the data that is being presented to the algorithm at particular time instants.

3. AFFINE PROJECTION LMMP ALGORITHM: MULTIDIMENSIONAL CASE

We now consider the multidimensional extension of the *a posteriori* LMMP algorithm in the previous section for arbitrary N. Let

$$\mathbf{X}_k = [\mathbf{x}_{1,k}\ \mathbf{x}_{2,k}\ \cdots\ \mathbf{x}_{N,k}] \quad (17)$$

$$\mathbf{d}_k = [d_{1,k}\ d_{2,k}\ \cdots\ d_{N,k}]^T \quad (18)$$

$$\mathbf{y}_k = \mathbf{X}_k^T \mathbf{w}_k \quad (19)$$

$$\mathbf{y}_{p,k} = [y_{p,1,k}\ y_{p,2,k}\ \cdots\ y_{p,N,k}]^T = \mathbf{X}_k^T \mathbf{w}_{k+1} \quad (20)$$

be the multidimensional extensions of the respective quantities in the LMMP algorithm, and define the complex modulus operator on an arbitrary-sized matrix as $|\mathbf{d}_k| = [|d_{1,k}|\ \cdots\ |d_{N,k}|]^T$, for example.

To begin our derivation, define the $(N \times N)$ diagonal matrix $\mathbf{F}_k$ as in Eq. (29) in Table 1. Then, the *a posteriori* affine projection LMMP algorithm is defined by the relation

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{1}{\beta}\mathbf{X}_k^*\left(\alpha \mathbf{d}_k - \mathbf{F}_k \mathbf{X}_k^T \mathbf{w}_{k+1}\right). \quad (21)$$

As before, (21) is not in the form of an update, as $\mathbf{w}_{k+1}$ explicitly appears on the right-hand-side as well as in the entries of $\mathbf{F}_k$ through their dependence on the entries of $|\mathbf{y}_{p,k}|$.

Assume for the time being that we have a way to compute all N values of $|\mathbf{y}_{p,k}|$ used within the matrix $\mathbf{F}_k$. Then, we can rewrite the above relation as

$$\left[\mathbf{I} + \mathbf{X}_k^*\left(\frac{1}{\beta}\mathbf{I}\right)\left(\mathbf{F}_k \mathbf{X}_k^T\right)\right]\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha}{\beta}\mathbf{X}_k^*\mathbf{d}_k, \quad (22)$$

and by using the matrix inversion lemma approach, we can invert the matrix premultiplying $\mathbf{w}_{k+1}$ to yield the relation shown in Eq. (30) in Table 1. Thus, the affine projection version of the algorithm is available so long as we have a strategy for computing the entries of $\mathbf{F}_k$. We now consider a strategy for computing the values in $|\mathbf{y}_{p,k}|$ needed to compute the diagonal entries of $\mathbf{F}_k$.

Using the update relation in (21), we pre-multiply both sides of this relation by $\mathbf{X}_k^T$ to obtain

$$\mathbf{y}_{p,k} = \mathbf{y}_k + \frac{\alpha}{\beta}\mathbf{X}_k^T\mathbf{X}_k^*\mathbf{d}_k - \frac{1}{\beta}\mathbf{X}_k^T\mathbf{X}_k^*\mathbf{F}_k\mathbf{y}_{p,k}, \quad (23)$$

which we can modify via some simple algebra to obtain

$$\left(\beta + \mathbf{X}_k^T\mathbf{X}_k^*\right)\mathbf{y}_{p,k} = \beta\mathbf{y}_k + \alpha\mathbf{X}_k^T\mathbf{X}_k^*\mathbf{d}_k + (1-\alpha)\mathbf{X}_k^T\mathbf{X}_k^*|\mathbf{D}_k|\mathrm{sgn}(\mathbf{y}_{p,k}) \quad (24)$$

where $|\mathbf{D}_k|$ is a diagonal matrix whose mth diagonal entry is $|d_{m,k}|$ and sgn($\mathbf{y}$) normalizes the complex-valued elements of its argument to individual unit lengths, keeping their phase angles.

At this point, there is no simple way to compute the entries of $|\mathbf{y}_{p,k}|$ as all of the values are coupled in the relation above, and one cannot perform the same operations using the modulus operation as was done in the $N = 1$ case. As such, we resort to a nonlinear iterative approach to numerically estimate the entries of $\mathbf{y}_{p,k}$. Let $\mathbf{y}_{p,k}^{(i)}$ denote the ith iteration of this approach, and set $\mathbf{y}_{p,k}^{(0)} = \mathbf{y}_k$ Then, we update $\mathbf{y}_{p,k}^{(i)}$ as shown in Eq. (28) in Table 1, where the N-dimensional vector $\mathbf{g}_k$ and $(N \times N)$ matrix $\mathbf{H}_k$ are defined in Eqs. (26) and (27), respectively, and γ is a relaxation parameter. Note that $\mathbf{g}_k$ and $\mathbf{H}_k$ need only be computed once per k, and they both share a common $(N \times N)$ matrix $(\beta\mathbf{I} + \mathbf{X}_k^T\mathbf{X}_k^*)$ that only needs to be inverted once per k. Thus, the complexity of this iterative approach to finding $\mathbf{y}_{p,k}$ is of $\mathcal{O}(KN^2)$, where K is the number of iterations of the iterative approach. Thus, for typical choices where the parameter vector length L obeys both $L \gg K$ and $L \gg N$, the overall update complexity is $[2NL + (K+2N)N^2 + \mathcal{O}(KN)]$ not counting the computation of $\mathbf{X}_k^T\mathbf{X}_k^*$, which remains reasonable so long as both K and N are not too large. Choosing $\gamma = 0.5$ and $5 \leq K \leq 10$ yields reasonable behavior. The choice of N is governed by the input signal correlation.

Once the estimate of $\mathbf{y}_{p,k}$ is found, we calculate the entries of $|\mathbf{y}_{p,k}|$ and use these to compute $\mathbf{F}_k$ in the update in (21). Alternatively, one can avoid the iterative approach to computing the entries of $\mathbf{y}_{p,k}^{(i)}$ and approximate this by choosing $K = 0$ which sets $\mathbf{y}_{p,k}^{(K)} = \mathbf{y}_k$, using the *a priori* output

$$\mathbf{y}_k \;=\; \mathbf{y}_{p,k}^{(0)} \;=\; \mathbf{X}_k^T\mathbf{w}_k. \qquad (25)$$

If $K > 0$, do

$$\mathbf{g}_k \;=\; \left(\beta + \mathbf{X}_k^T\mathbf{X}_k^*\right)^{-1}\left(\beta\mathbf{y}_k + \alpha\mathbf{X}_k^T\mathbf{X}_k^*\mathbf{d}_k\right) \qquad (26)$$

$$\mathbf{H}_k \;=\; (1-\alpha)\left(\beta + \mathbf{X}_k^T\mathbf{X}_k^*\right)^{-1}\mathbf{X}_k^T\mathbf{X}_k^*|\mathbf{D}_k| \qquad (27)$$

For $i = 1, 2, \ldots, K$ and $\gamma = 0.5$, do

$$\mathbf{y}_{p,k}^{(i)} = (1-\gamma)\mathbf{y}_{p,k}^{(i-1)} + \gamma\left[\mathbf{g}_k + \mathbf{H}_k \mathrm{sgn}\left(\mathbf{y}_{p,k}^{(i-1)}\right)\right] \qquad (28)$$

end i

end if

$$\mathbf{F}_k \;=\; \mathrm{diag}\left\{1 + (\alpha-1)\frac{|d_{1,k}|}{|y_{p,1,k}^{(K)}|}, \ldots, 1 + (\alpha-1)\frac{|d_{N,k}|}{|y_{p,N,k}^{(K)}|}\right\} \qquad (29)$$

$$\mathbf{w}_{k+1} \;=\; \mathbf{w}_k + \mathbf{X}_k^*\left(\beta\mathbf{I} + \mathbf{F}_k\mathbf{X}_k^T\mathbf{X}_k^*\right)^{-1}(\alpha\mathbf{d}_k - \mathbf{F}_k\mathbf{y}_k). \qquad (30)$$

For the approximate version, choose $K = 0$.

Table 1. Affine-projection least-mean-magnitude-phase algorithm for projection order $N \geq 1$.

values in the computation of the magnitude and phase errors in the update. This approximate version has a complexity of $2NL + \mathcal{O}(N^3)$. Both versions of this algorithm for $K = 0$ and $K > 0$ are given in Table 1 and retain the structural simplicity of the original affine projection algorithm.

Remark 5: The algorithm in Table 1 is guaranteed to maintain all entries of $|\mathbf{y}_{p,k}^{(K)}| > 0$ as they are obtained by complex modulus operations. Even so, one should still avoid a divide-by-near-zero situation in cases where any entry of $|\mathbf{y}_{p,k}^{(K)}|$ is small relative to its corresponding $|\mathbf{d}_{p,k}|$ value. We have employed the test $|y_{p,i,k}^{(K)}| > 0.001|d_{i,k}|$ and replace the vector $\mathbf{y}_{p,k}^{(K)}$ with $\mathbf{y}_k$ if any one of these tests fails, similar to the recovery mechanism used in the $N = 1$ algorithm version.

Remark 6: The choices of $\gamma = 0.5$ and $K = 10$ were determined by experimentation but are justified as follows. For these choices, the initial condition remaining in the *a posteriori* output vector is reduced to approximately 2^{-10} or 60 dB below its initial value, and errors in the *a posteriori* estimates typically are between 20 dB and 70 dB below that of the errors in $\mathbf{e}_k = \mathbf{d}_k - \mathbf{y}_k$. Thus, the errors in the *a posteriori* output values are not dominant components in overall behavior.

4. NUMERICAL EVALUATIONS

The capabilities of the proposed methods are now explored via numerical simulations. Consider an $M = 10$-element uniform linear array with uncorrelated complex Gaussian sensor noise in which three 4QAM-modulated narrowband signals with SNRs of 30, 20, and 10 dB are impinging on the array at angles of 20, 25, and -45 degrees, respectively. Due to carrier offset [17], the 4QAM constellations are rotating 18 degrees in the complex plane between snapshots. The sensor signal snapshots are used as the input vector sequence $\mathbf{x}_k$. We use the first 4QAM signal as the desired signal d_k, and implement five different beamforming algorithms, choosing step size parameters to obtain the fastest convergence of the signal-to-interference ratio (SINR) for each algorithm [16]. One thousand simulations are run and the results averaged.

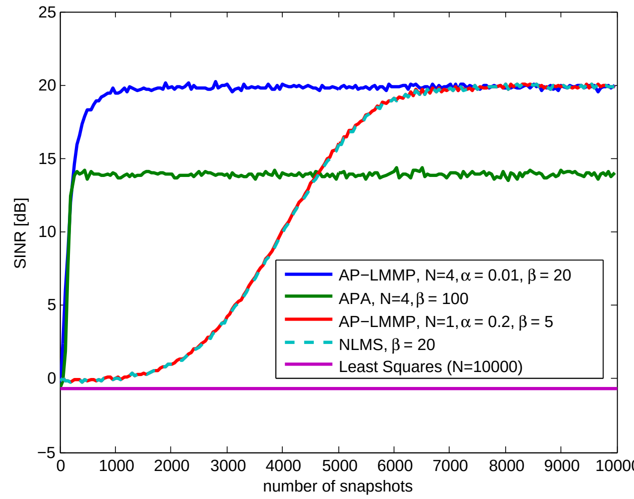

Fig. 1. Array processing example; see text for explanation.

Shown in Fig. 1 are the results of these evaluations. As can be seen, the AP-LMMP algorithm with $N = 4$ performs the best, providing the fastest convergence with the highest final SINR. The standard affine projection algorithm has the same convergence speed as AP-LMMP with $N = 4$ but cannot achieve a high SINR as its parameters cannot be adjusted to account for the carrier offset. The AP-LMMP with $N = 1$ and NLMS algorithms perform similarly, and both provide a high steady-state SINR but a slower convergence speed relative to the $N = 4$-based algorithms. The static solution provided by least-squares is unable to extract the desired source due to carrier offset. These results show the usefulness of the multidimensional AP-LMMP algorithm to independently manage amplitude and phase errors and simultaneously compensate for input signal correlation in an adaptive estimation task.

5. CONCLUSIONS

In this paper, we have described new algorithms for adaptive estimation of complex-valued signals that can be tuned to emphasize amplitude or phase information in the training data. The algorithms are simple and robust, employing *a posteriori* updates that allow a wide range of tuning parameters to be chosen. Novel methods for computing the magnitudes of the *a posteriori* output values are provided. Simulations show that the multidimensional algorithm has a fast convergence speed and a high steady-state accuracy through its tuning ability. The algorithms are useful for complex-valued signal processing tasks in communications and array processing.

6. REFERENCES

[1] B. Widrow, J. McCool, and M. Ball, “The complex LMS algorithm,” *Proc. IEEE,* vol. 63, pp. 719-720, Apr. 1975.

[2] M. Rupp, “On the separation of channel and frequency estimation,” *Proc. 32nd Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, vol. 2, pp. 1186-1190, Nov. 1998.

[3] A. Tarighat and A.H. Sayed, “Least mean-phase adaptive filters with applications to communications systems,” *IEEE Signal Processing Lett.,* vol. 11, no. 2, pp. 220-223, Feb. 2004.

[4] S.C. Douglas and D.P. Mandic, “The least-mean-magnitude-phase algorithm with applications in communications systems,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Prague, Czech Republic, pp. 4152-4155, May 2011.

[5] S.C. Douglas and D.P. Mandic, “Stability analysis of the least-mean-magnitude-phase algorithm,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Shanghai, China, pp. 4935-4939, May 2016.

[6] K.T. Wagner and M.I. Doroslovacki, “Complex proportionate-type affine projection algorithms,” *Proc. 47th Ann. Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, pp. 1510-1514, Nov. 2013.

[7] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electronics and Communications in Japan (Part I: Communications),* vol. 67, no. 5, pp. 19-27, May 1984.

[8] S. L. Gay, and S. Tavathia, “The fast affine projection algorithm,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, vol. 5, pp. 3023-3026, May 1995.

[9] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, “Fast projection algorithm and its step size control,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, vol. 2, pp. 945-948, May 1995.

[10] K. Maouche and D.T.M. Slock, “A fast instrumental variable affine projection algorithm,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Seattle, WA, vol. 3, pp. 1481-1484, May 1998.

[11] S. Haykin, *Adaptive Filter Theory,* 4th ed. (Upper Saddle River, NJ: Prentice-Hall, 2002).

[12] M. Rupp, “Contraction mapping: An important property in adaptive filters,” *Proc. 6th IEEE Digital Signal Processing Workshop,* Yosemite National Park, CA, pp. 273-276, Oct. 1994.

[13] A.H. Sayed and M. Rupp, “Error energy bounds for adaptive gradient algorithms,” *IEEE Trans. Signal Processing,* vol. 44, no. 8, pp. 1982-1989, Aug. 1996.

[14] S.C. Douglas and M. Rupp, “A posteriori updates for adaptive filters,” *Proc. 31st IEEE Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, vol. 2, pp. 1641-1645, Nov. 1997.

[15] M. Rupp and S.C. Douglas, “A posteriori analysis of adaptive blind equalizers,” *Proc. 32nd Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, vol. 1, pp. 369-373, Nov. 1998.

[16] D.G. Manolakis, V.K. Ingle, and S.M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing* (Norwood, Massachusetts: Artech, 2005, reissued).

[17] M. C. L. Siu, “Effects of local oscillator errors on digital beamforming,” M.S. thesis, Naval Postgraduate School, Monterey, CA, No. 2016-03, March 2016.

[18] A.H. Sayed, *Fundamentals of Adaptive Filtering* (Hoboken, NJ: Wiley, 2003).

[19] D.P. Mandic and S.L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models* (West Sussex, United Kingdom: Wiley, 2009).