

WIDELY-LINEAR RECURSIVE LEAST-SQUARES ALGORITHM FOR ADAPTIVE BEAMFORMING

Scott C. Douglas

Department of Electrical Engineering
Southern Methodist University
Dallas, Texas 75275 USA

ABSTRACT

Adaptive beamforming algorithms typically rely on a complex linear model between the sensor measurements and the desired signal output that does not enable the best performance from the data in some situations. In this paper, we present an extension of the well-known recursive least-squares algorithm for adaptive filters to widely-linear complex-valued signal and system modeling. The widely-linear RLS algorithm exploits a structured covariance matrix update that maintains information about the non-circularity of the input data to solve the widely-linear least-squares task at each snapshot. In addition, the WL-RLS algorithm can easily be switched between conventional and widely-linear complex modeling as needed. Application of the method to adaptive beamforming of mixed BPSK and QPSK signal transmissions shows that the system can extract all of the transmitted signal outputs in certain overloaded scenarios, and it performs up to 3dB better than the conventional RLS beamformer when the array is not overloaded.

Index Terms—adaptive arrays, adaptive filters, adaptive systems, least squares methods, recursive estimation

1. INTRODUCTION

Adaptive filters are used in a number of practical applications in digital communications, medicine, speech processing, and digital image processing. Adaptive filters enable one to implement a time-varying linear filter to model signal relationships, remove unwanted noise, or enhance a particular desired signal component when two signal sets, termed the input signals and the desired response signals, respectively, are available. The two most-popular adaptive filtering methods are the least-mean-square algorithm of Widrow and Hoff [1] and the recursive least-squares algorithm that is a special case of the well-known Kalman filter for state space estimation [2].

Usually, adaptive filters are designed to process real-valued signals, although complex extensions of some adaptive algorithms exist. These complex extensions have typically treated the input-output relationship as complex linear, which implies a certain structure on the input-output relationship of the system. For example, for an M -antenna adaptive beamformer with coefficients $\{g_1(n), \dots, g_M(n)\}$ at iteration n , the output signal of the filter at time k is typically

$$y_{1,n}(k) = \sum_{m=1}^M g_m(n)x_m(k), \quad (1)$$

where $x_m(k)$ is the m th sensor signal at time k . We can also consider the model in which the input signals are conjugated, given by

$$y_{2,n}(k) = \sum_{m=1}^M h_m(n)x_m^*(k), \quad (2)$$

where $\{h_1(n), \dots, h_M(n)\}$ are the coefficients of the adaptive beamformer. It is important to realize that for a given set of coefficients $\{g_m(n)\}$, it is typically impossible to choose values for $\{h_m(n)\}$ such that $y_{1,n}(k) = y_{2,n}(k)$; thus, the above two linear operations represent two different input-output relations that produce different output signals for the same complex input signals.

Linear adaptive systems involving complex-valued signals play an important role in beamforming for multi-port antenna arrays. Such systems exploit spatial diversity to build one or more signal estimates from a set of measured signals. For a uniform linear array (ULA), an appropriate signal model for a set of received signals $\{x_m(k)\}$, $1 \leq k \leq M$ is

$$x_m(k) = \eta_m(k) + \sum_{i=1}^N \delta_m(\phi_i)s_i(k) \quad (3)$$

$$\delta_m(\phi) = \exp\left(-j2\pi \frac{\Delta \sin(\phi)}{\gamma}(m-1)\right) \quad (4)$$

where $s_i(k)$ is the signal from the i th user at time k , ϕ_i is the angle of arrival of the i th narrowband signal with respect to the array normal, Δ is the inter-element antenna spacing, γ is the wavelength, and $\eta_m(k)$ is the complex-valued sensor noise at the m th sensor. Using either the structure in (1) or (2), it is possible to produce an output signal that estimates one of the user signals $s_i(k)$ or $s_i^*(k)$. Parallel versions of this structure can be used to estimate several user signals simultaneously.

The difference between (1) and (2) suggests a model that uses both $\{x_m(k)\}$ and $\{x_m^*(k)\}$ as input signals. Such models are termed *widely linear*. The widely-linear model between $\{x_m(k)\}$ and $y_n(k)$ is

$$y_n(k) = \sum_{m=1}^M g_m(n)x_m(k) + h_m(n)x_m^*(k), \quad (5)$$

This model has twice the number of complex coefficients than either of the models in (1) or (2). More importantly, it has a modeling capability that is more capable than that of either (1) or (2) [4]–[8]. For example, using (5), it is possible to build beamformers that accurately estimate N user signals $\hat{s}_i(k)$ when there are more users N than antennas M , if two or more of the user signals are real-valued. Such performance overcomes a chief limitation of M -antenna array systems when one of the output signal models in (1) or (2) is used.

This paper describes a novel recursive least-squares algorithm for adapting the coefficients of the widely-linear model in (5) to minimize the well-known exponentially-weighted least-squares cost

$$\mathcal{J}(\mathbf{w}(n)) = \sum_{k=0}^n \lambda^{n-k} |d(k) - y_n(k)|^2, \quad (6)$$

where $d(k)$ is the complex-valued desired response signal, λ is a forgetting factor, and

$$y_n(k) = \mathbf{v}^H(k)\mathbf{w}(n), \quad (7)$$

$$\mathbf{v}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}^*(k) \end{bmatrix} \quad (8)$$

$$\mathbf{w}(n) = \begin{bmatrix} \mathbf{h}(n) \\ \mathbf{g}(n) \end{bmatrix} \quad (9)$$

$$\mathbf{g}(n) = [g_1(n) \cdots g_M(n)]^T \quad (10)$$

$$\mathbf{h}(n) = [h_1(n) \cdots h_M(n)]^T \quad (11)$$

$$\mathbf{x}(n) = [x_1(n) \cdots x_M(n)]^T. \quad (12)$$

This paper thus provides competing algorithms to previously-proposed approaches involving gradient descent, as exemplified in the augmented complex LMS algorithm [3]. The ACLMS algorithm updates the parameters of the widely-linear model as

$$\mathbf{g}(n) = \mathbf{g}(n-1) + \mu(n)(d(n) - y_{n-1}(n))\mathbf{x}^*(n) \quad (13)$$

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu(n)(d(n) - y_{n-1}(n))\mathbf{x}(n). \quad (14)$$

Because it is a stochastic gradient descent algorithm, ACLMS suffers from convergence issues not unlike the complex LMS and real-valued LMS algorithm in certain situations. It is expected that recursive least-squares formulations to the widely-linear signal estimation problem will have the same benefits and drawbacks as recursive least-squares approaches for real-valued signal processing when compared to gradient methods, namely, better estimation abilities at the price of increased computational complexity. It is our primary goal in this paper, therefore, to show a minimum-complexity implementation of the recursive least-squares approaches, one which we achieve by leveraging the block-conjugate structure of the associated data covariance matrices that appear in the derivations.

2. DERIVATION OF THE WIDELY-LINEAR RLS ALGORITHM

It is straightforward to show that the optimum solution to the minimization task of (6)–(12) is

$$\mathbf{w}(n) \equiv \begin{bmatrix} \mathbf{h}(n) \\ \mathbf{g}(n) \end{bmatrix} = \mathbf{R}_v^{-1}(n)\mathbf{f}_v(n), \quad (15)$$

where

$$\mathbf{R}_v(n) \equiv \begin{bmatrix} \mathbf{R}_x(n) & \mathbf{P}_x(n) \\ \mathbf{P}_x^*(n) & \mathbf{R}_x^*(n) \end{bmatrix} \quad (16)$$

$$= \sum_{k=1}^n \lambda^{n-k} \begin{bmatrix} \mathbf{x}(k)\mathbf{x}^H(k) & \mathbf{x}(k)\mathbf{x}^T(k) \\ \mathbf{x}^*(k)\mathbf{x}^H(k) & \mathbf{x}^*(k)\mathbf{x}^T(k) \end{bmatrix} \quad (17)$$

$$\mathbf{f}_v(n) \equiv \begin{bmatrix} \mathbf{f}_1(n) \\ \mathbf{f}_2(n) \end{bmatrix} = \sum_{k=1}^n \lambda^{n-k} \begin{bmatrix} \mathbf{x}(k)d(k) \\ \mathbf{x}^*(k)d(k) \end{bmatrix} \quad (18)$$

and

$$\mathbf{R}_x(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^H(k) \quad (19)$$

$$\mathbf{P}_x(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^T(k) \quad (20)$$

$$\mathbf{f}_1(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}(k)d(k) \quad (21)$$

$$\mathbf{f}_2(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}^*(k)d(k) \quad (22)$$

The key issue in developing a computationally-efficient algorithm for WL-RLS is to recognize that the inverse of $\mathbf{R}_v(n)$ has the same block conjugate structure as $\mathbf{R}_v(n)$; *i.e.*

$$\begin{bmatrix} \mathbf{R}_x(n) & \mathbf{P}_x(n) \\ \mathbf{P}_x^*(n) & \mathbf{R}_x^*(n) \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{C}(n) & \mathbf{D}(n) \\ \mathbf{D}^*(n) & \mathbf{C}^*(n) \end{bmatrix}. \quad (23)$$

To see this fact, consider the product

$$\begin{bmatrix} \mathbf{R}_x(n) & \mathbf{P}_x(n) \\ \mathbf{P}_x^*(n) & \mathbf{R}_x^*(n) \end{bmatrix} \begin{bmatrix} \mathbf{C}(n) & \mathbf{D}(n) \\ \mathbf{D}^*(n) & \mathbf{C}^*(n) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (24)$$

From this relation, four are revealed:

$$\mathbf{R}_x(n)\mathbf{C}(n) + \mathbf{P}_x(n)\mathbf{D}^*(n) = \mathbf{I} \quad (25)$$

$$\mathbf{R}_x(n)\mathbf{D}(n) + \mathbf{P}_x(n)\mathbf{C}^*(n) = \mathbf{0} \quad (26)$$

$$\mathbf{R}_x^*(n)\mathbf{C}^*(n) + \mathbf{P}_x^*(n)\mathbf{D}(n) = \mathbf{I} \quad (27)$$

$$\mathbf{R}_x^*(n)\mathbf{D}^*(n) + \mathbf{P}_x^*(n)\mathbf{C}(n) = \mathbf{0} \quad (28)$$

Clearly, the latter two matrix equations are simply conjugates of the first two matrix equations, which themselves are sets of $2L^2$ complex equations in $2L^2$ complex unknowns and are solvable. Thus, the inverse structure in (23) is valid, although it should be noted that $\mathbf{C}(n) \neq \mathbf{R}_x^{-1}(n)$, and other simple relations between $\mathbf{D}(n)$ and $\mathbf{P}_x(n)$ also do not hold. With this recognition, we see that

$$\mathbf{h}(n) = \mathbf{C}(n)\mathbf{f}_1(n) + \mathbf{D}(n)\mathbf{f}_2(n) \quad (29)$$

$$\mathbf{g}(n) = \mathbf{D}^*(n)\mathbf{f}_1(n) + \mathbf{C}^*(n)\mathbf{f}_2(n) \quad (30)$$

The value of $\mathbf{R}_v(n)$ can be updated as

$$\mathbf{R}_v(n) = \lambda\mathbf{R}_v(n-1) + \mathbf{v}(n)\mathbf{v}^H(n). \quad (31)$$

Using the matrix inversion lemma, we can express

$$\begin{aligned} \mathbf{R}_v^{-1}(n) &= \frac{1}{\lambda} (\mathbf{R}_v^{-1}(n-1) \\ &\quad - \mathbf{R}_v^{-1}(n-1)\mathbf{v}(n)c^{-1}(n)\mathbf{v}^H(n)\mathbf{R}_v^{-1}(n-1)) \beta_2 \end{aligned}$$

where

$$c(n) = \lambda + \mathbf{v}^H(n)\mathbf{R}_v^{-1}(n-1)\mathbf{v}(n). \quad (33)$$

Using the definition of $\mathbf{C}(n)$ and $\mathbf{D}(n)$ afforded by the relationship in (24), we have

$$\mathbf{C}(n) = \frac{1}{\lambda} (\mathbf{C}(n-1) - \mathbf{u}(n)c^{-1}(n)\mathbf{u}^H(n)) \quad (34)$$

$$\mathbf{D}(n) = \frac{1}{\lambda} (\mathbf{D}(n-1) - \mathbf{u}(n)c^{-1}(n)\mathbf{u}^T(n)) \quad (35)$$

$$\mathbf{u}(n) = \mathbf{C}(n-1)\mathbf{x}(n) + \mathbf{D}(n-1)\mathbf{x}^*(n), \quad (36)$$

where we have used the properties that $\mathbf{C}^*(n) = \mathbf{C}^T(n)$ and $\mathbf{D}(n) = \mathbf{D}^T(n)$. Furthermore, by substitution of (23) into (33), we can obtain

$$c(n) = \lambda + \mathbf{x}^H(n)\mathbf{u}(n) + \mathbf{x}^T(n)\mathbf{u}^*(n) \quad (37)$$

$$= \lambda + 2\text{Re}\{\mathbf{x}^H(n)\mathbf{u}(n)\} \quad (38)$$

Now, noting that

$$\mathbf{f}_1(n) = \lambda\mathbf{f}_1(n-1) + \mathbf{x}(n)d(n) \quad (39)$$

$$\mathbf{f}_2(n) = \lambda\mathbf{f}_2(n-1) + \mathbf{x}^*(n)d(n) \quad (40)$$

we may substitute (39) and (40) into (34) and (36). For the first equation, we see that

$$\begin{aligned} \mathbf{C}(n)\mathbf{f}_1(n) &= \frac{1}{\lambda} \left(\mathbf{C}(n-1) - \mathbf{u}(n)c^{-1}(n)\mathbf{u}^H(n) \right) \\ &\quad \times (\lambda\mathbf{f}_1(n-1) + \mathbf{x}(n)d(n)) \end{aligned} \quad (41)$$

$$\begin{aligned} &= \mathbf{C}(n-1)\mathbf{f}_1(n-1) + \frac{1}{\lambda}\mathbf{C}(n-1)\mathbf{x}(n)d(n) \\ &\quad - \mathbf{u}(n)c^{-1}(n)\mathbf{u}^H(n)\mathbf{f}_1(n-1) \\ &\quad - \frac{1}{\lambda}\mathbf{u}(n)c^{-1}(n)\mathbf{u}^H(n)\mathbf{x}(n)d(n) \end{aligned} \quad (42)$$

and

$$\begin{aligned} \mathbf{D}(n)\mathbf{f}_2(n) &= \mathbf{D}(n-1)\mathbf{f}_2(n-1) + \frac{1}{\lambda}\mathbf{D}(n-1)\mathbf{x}^*(n)d(n) \\ &\quad - \mathbf{u}(n)c^{-1}(n)\mathbf{u}^T(n)\mathbf{f}_2(n-1) \\ &\quad - \frac{1}{\lambda}\mathbf{u}(n)c^{-1}(n)\mathbf{u}^T(n)\mathbf{x}^*(n)d(n) \end{aligned} \quad (43)$$

Combining these terms, it can be shown that

$$\begin{aligned} \mathbf{h}(n) &= \mathbf{h}(n-1) \\ &\quad + \mathbf{u}(n)c^{-1}(n) \left[d(n) - \mathbf{u}^H(n)\mathbf{f}_1(n-1) + \mathbf{u}^T(n)\mathbf{f}_2(n-1) \right] \end{aligned} \quad (44)$$

It can also be shown that

$$\begin{aligned} \mathbf{u}^H(n)\mathbf{f}_1(n-1) + \mathbf{u}^T(n)\mathbf{f}_2(n-1) \\ &= \mathbf{x}^H(n)\mathbf{h}(n-1) + \mathbf{x}^T(n)\mathbf{g}(n-1), \end{aligned} \quad (45)$$

such that the update for $\mathbf{h}(n)$ is

$$\mathbf{h}(n) = \mathbf{h}(n-1) + e(n)\mathbf{k}(n) \quad (46)$$

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{g}(n-1) - \mathbf{x}^H(n)\mathbf{h}(n-1) \quad (47)$$

$$\mathbf{k}(n) = c^{-1}(n)\mathbf{u}(n). \quad (48)$$

A similar update to that in (46) holds for $\mathbf{g}(n)$, except $\mathbf{k}^*(n)$ is used in place of $\mathbf{k}(n)$. Thus, the complete updates are

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{g}(n-1) - \mathbf{x}^H(n)\mathbf{h}(n-1) \quad (49)$$

$$\mathbf{u}(n) = \mathbf{C}(n-1)\mathbf{x}(n) + \mathbf{D}(n-1)\mathbf{x}^*(n) \quad (50)$$

$$\mathbf{k}(n) = (\lambda + 2\text{Re}\{\mathbf{x}^H(n)\mathbf{u}(n)\})^{-1}\mathbf{u}(n) \quad (51)$$

$$\mathbf{C}(n) = \frac{1}{\lambda} \left(\mathbf{C}(n-1) - \mathbf{u}(n)\mathbf{k}^H(n) \right) \quad (52)$$

$$\mathbf{D}(n) = \frac{1}{\lambda} \left(\mathbf{D}(n-1) - \mathbf{u}(n)\mathbf{k}^T(n) \right) \quad (53)$$

$$\mathbf{g}(n) = \mathbf{g}(n-1) + e(n)\mathbf{k}^*(n) \quad (54)$$

$$\mathbf{h}(n) = \mathbf{h}(n-1) + e(n)\mathbf{k}(n) \quad (55)$$

Equations (49)–(55) are the widely-linear RLS algorithm. Several remarks about this algorithm can be made:

Remark #1: The WL-RLS algorithm reduces to the conventional RLS algorithm for complex linear input-output relationships if we set $\mathbf{D}(n) = \mathbf{0}$ and either $\mathbf{g}(n) = \mathbf{0}$ or $\mathbf{h}(n) = \mathbf{0}$ for all n , and we modify the Kalman gain as

$$\mathbf{k}(n) = (\lambda + \text{Re}\{\mathbf{x}^H(n)\mathbf{u}(n)\})^{-1}\mathbf{u}(n) \quad (56)$$

In such cases, $\mathbf{u}(n) = \mathbf{C}(n-1)\mathbf{x}(n)$ and $\mathbf{C}(n) = \mathbf{R}_x^{-1}(n)$. This result means that, depending on how large the magnitudes of the entries of $\mathbf{g}(n)$ are with respect to $\mathbf{h}(n)$, we may switch from the

WL-RLS algorithm to the conventional RLS algorithm by setting either $\mathbf{g}(n)$ or $\mathbf{h}(n)$ to zero and not updating the zeroed coefficient vector, removing the entries of $\mathbf{D}(n)$ in the updates, and defining $\mathbf{k}(n)$ as above. Similarly, if we have either $\mathbf{g}(n)$ or $\mathbf{h}(n)$ active in the estimation process and wish to switch to the widely-linear model, we can switch back to the complete algorithm by allowing $\mathbf{D}(n)$ to be non-zero again and updating all parameters according to the WL-RLS algorithm.

Remark #2: The computational complexity of the WL-RLS algorithm is about twice that of the conventional RLS algorithm when applied to complex data.

Remark #3: The conventional real-valued RLS algorithm is known to have issues of numerical stability when implemented in finite-precision arithmetic [9]. These issues can typically be addressed by maintaining a minimum-parameter implementation of the system, in which symmetry of the inverse autocorrelation matrix at time n is maintained. Although a stability analysis of the WL-RLS algorithm has not been performed, we suspect that it, too, suffers from similar numerical instabilities. Fortunately, a minimum-parameter implementation of WL-RLS can be obtained by maintaining the symmetries of $\mathbf{C}(n)$ and $\mathbf{D}(n)$ as

$$\mathbf{C}(n) = \mathbf{C}^H(n) \quad (57)$$

$$\mathbf{D}(n) = \mathbf{D}^T(n) \quad (58)$$

by only propagating the unique values within each of these matrices and copying the appropriate matrix entries to their associated identical value locations. Our implementations make use of this structure, and we have not observed any numerical issues in any of our simulations of the approach.

Remark #4: The performance of the WL-RLS algorithm is *identical* to that of two real-valued RLS algorithms in which the input signal for each algorithm is the vector $[\text{Re}\{\mathbf{x}^T(n)\} \text{Im}\{\mathbf{x}^T(n)\}]^T$ and the desired responses are $\text{Re}\{d(n)\}$ and $\text{Im}\{d(n)\}$, respectively. Moreover, the computational complexity of the two approaches, in terms of numbers of real-valued multiplies, are also about the same, if symmetries in appropriate matrices are taken into account. Thus, there is no performance or computational gain to be obtained by using WL-RLS over a conventional two-channel real-valued RLS algorithm involving common input signals. The convenience of the widely-linear processing model, however, is in its partitioning of the input data statistics into circular and non-circular portions, and the partitioning of the model into conventional complex and widely-linear portions.

3. BEAMFORMING EXAMPLES

We now show the advantage of WL-RLS in a beamforming context. Let $\mathbf{x}(k)$ fit the narrowband array model in (4), where the normalized sensor spacing is $\Delta/\gamma = 1/2$, the number of sources is $N = 4$, and the number of elements is $M = 3$. Because $M \leq N$, it is not possible to resolve each source using a traditional complex least-squares beamformer using either $y_{1,n}(k)$ or $y_{2,n}(k)$ in (1) or (2). It is possible, however, to resolve all four sources using $y_n(k)$ in (5) if two or more of the sources are real-valued. Let $s_1(k)$ and $s_2(k)$ be two independent rotated BPSK signals, and let $s_3(k)$ and $s_4(k)$ be two independent QPSK signals. Furthermore, let $\{\phi_1, \phi_2, \phi_3, \phi_4\} = \{-45^\circ, 8^\circ, -13^\circ, 30^\circ\}$, and let the complex Gaussian measurement noise be such that the signal-to-noise ratio (SNR) of each source in each sensor is 25dB. We apply four versions of the complex RLS as well as the WL-RLS algorithms to this data, where $d_i(k) = s_i(k)$, $\lambda = 0.99$, $\mathbf{C}(0) = 100\mathbf{I}$, and $\mathbf{D}(0) = \mathbf{0}$.

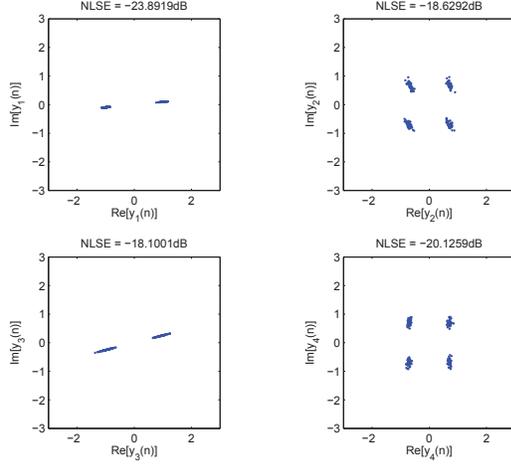


Fig. 1. Output signal constellations for the WL-RLS beamformers.

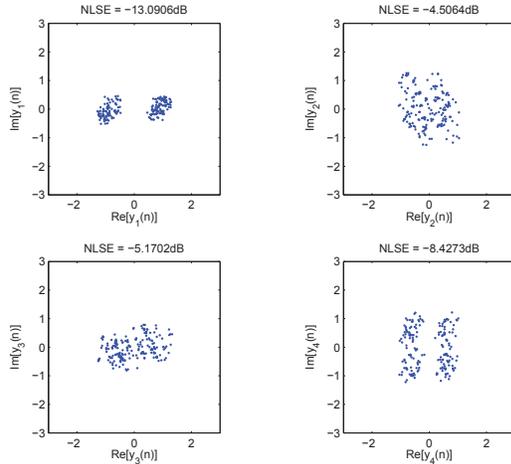


Fig. 2. Output signal constellations for the RLS beamformers.

Shown in Fig. 1 are the baseband output signal constellations for $20 \leq n \leq 200$ using the widely-linear RLS algorithm, in which the BPSK and QPSK output signal patterns are readily evident. Fig. 2 shows the corresponding outputs for the standard RLS algorithm, in which the BPSK and QPSK output signal patterns are not resolved for sources 2 and 3 and are barely resolvable for sources 1 and 4.

Fig. 3 shows the convergence of the squared errors for the normalized complex LMS, normalized augmented complex LMS, RLS, and WL-RLS algorithms as computed from 100 different data sets with these particular statistics, where we have chosen $\mu_{LMS} = \mu_{ACLMS} = 0.2$ to try to obtain the fastest convergence from the gradient approaches without significant noise enhancement. Clearly, ACLMS converges much more slowly than the other approaches, and while LMS converges quickly, its final NLSE is too large to be useful. Only the WL-RLS algorithm obtains both fast convergence and a low NLSE at convergence in this example.

Table 1 shows the average NLSE as a function of the number of antenna elements for the RLS and WL-RLS algorithms for each of the source types in this scenario, where $2 \leq M \leq 8$. As can be seen, the WL-RLS algorithm outperforms the RLS algorithm for $M \leq N$. When $M > N$, the BPSK NLSE is approximately 3dB better for the former algorithm, and the QPSK NLSEs are approximately the

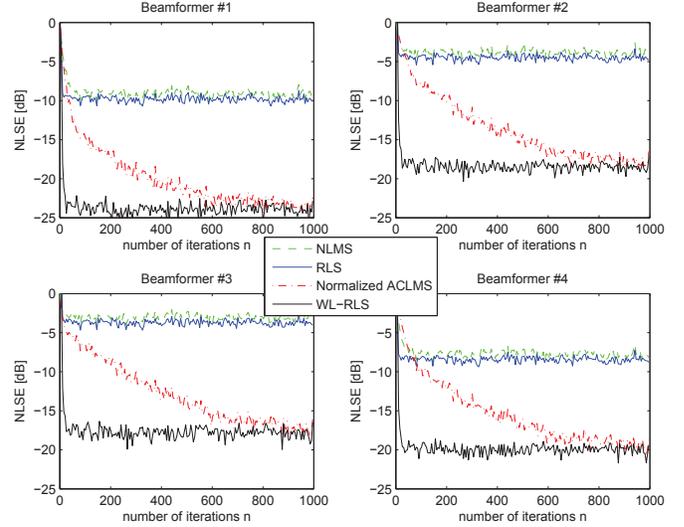


Fig. 3. Average NLSEs for the four beamformers in the data example.

Table 1: Steady-State NLSEs (dB) vs. number of antenna elements

| Algorithm (Signal Type) | M | | | | | |
|----------------------------|------|-------|-------|-------|-------|-------|
| | 2 | 3 | 4(N) | 5 | 6 | 8 |
| RLS (BPSK) | -3.0 | -5.9 | -21.9 | -24.8 | -24.7 | -24.4 |
| WLRLS(BPSK) | -4.7 | -20.1 | -24.9 | -27.6 | -27.5 | -27.0 |
| RLS (QPSK) | -5.8 | -10.1 | -24.6 | -27.6 | -27.8 | -27.3 |
| WLRLS(QPSK) | -7.5 | -25.4 | -26.8 | -27.5 | -27.5 | -27.3 |

same. The 3dB performance improvement in BPSK signal estimation provided by widely-linear modeling is well-understood [6].

4. CONCLUSIONS

This paper presents a conventional recursive least-squares algorithm for widely-linear signal and system modeling. The algorithm leverages the block conjugate structure of the augmented data correlation matrix to minimize the computational complexity of the coefficient updates. Numerical examples show that the proposed method outperforms both gradient approaches and conventional RLS approaches in adaptive beamforming tasks.

5. REFERENCES

- [1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," 1960 IRE WESCON Convention Record, pp. 96-104, 1960.
- [2] T. Kailath, *Linear Systems*, (Englewood Cliffs, NJ: Prentice Hall, 1980).
- [3] S. Javidi, M. Pedzisz, S.L. Goh, and D.P. Mandic, "The augmented complex least mean square algorithm with application to adaptive prediction problems", *Proc. IAPR Workshop Cognitive Information Processing Syst.*, Santorini, Greece, Session P, June 2008.
- [4] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 2030-2033, Aug. 1995.
- [5] W. Gerstacker, R. Schober, and A. Lampe, "Receivers with widely linear processing for frequency-selective channels," *IEEE Trans. Commun.* vol. 51, pp. 1512-1523, Sept. 2003.
- [6] P.J. Schreier, L.L. Scharf, and C.T. Mullis, "Detection and estimation of improper complex random signals," *IEEE Trans. Inform. Theory*, vol. 51, pp. 306-312, Jan. 2005.
- [7] D. Mattered, L. Paura, and F. Sterle, "Widely linear decision-feedback equalizer for time-dispersive linear MIMO channels," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2525-2536, July 2005.
- [8] P. Chevalier and F. Pison, "New insights into optimal widely linear array receivers for the demodulation of BPSK, MSK, and GMSK signals corrupted by noncircular interferences-application to SAIC," *IEEE Trans. Signal Processing*, vol. 54, pp. 870-883, Mar. 2006.
- [9] G.E. Bottomley and S.T. Alexander, "A theoretical basis for divergence of conventional recursive least squares filters," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Glasgow, UK, vol. 2, pp. 908-911, May 1989.