Blind Cancellation of Co-Channel Interference

Anuj Batra and John R. Barry

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250

Abstract — A co-channel system can be viewed as a multiple-input, multiple-output communication system. By extending the scalar constant-modulus algorithm to vector-valued signals, we develop several methods for blind cancellation of co-channel interference. In particular, to prevent the receiver from locking on to the same signal more than once, we propose the determinant constant-modulus algorithm. For the special case of constant-modulus signals and a memoryless channel, this method can outperform other known methods, as demonstrated by simulation.

I. Introduction

Co-channel interference (CCI) is the crosstalk caused by linear coupling among multiple channels, and the desire to eliminate this interference has motivated much research in the previous two decades [1–7]. In one form or another, CCI affects a wide variety of systems. For example, CCI in twisted-pair bundles results from linear coupling between neighboring pairs within a bundle [4]; CCI in dually polarized radio results from atmospheric rotations [5]; CCI in magnetic recording channels results from the coupling of data from adjacent tracks to the read head [6]; CCI in multiuser communications channels, where multiple users transmit at the same time and within the same frequency band via a common channel, results from the non-orthogonality of the signals at the receiver [7].

In each of the applications described above, a single receiver observes data from n different sources through m different sensors, resulting in an equivalent $m \times n$ multiple-input, multiple-output (MIMO) channel model. The block diagram of a MIMO channel is shown in Fig. 1. The channel input $\boldsymbol{x}_k$ is a sequence of vectors whose components represent the symbol sequences for each of n users. The channel transfer function is $\boldsymbol{H}(z)$, and the noise sequence $\boldsymbol{n}_k$ is assumed to be white and Gaussian with power spectral density $N_0\boldsymbol{I}$. For simplicity, we assume that $\boldsymbol{H}(z)$ is square, so that the dimension of the channel output $\boldsymbol{r}_k$ is n.

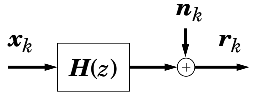

Fig. 1. A MIMO communications channel.

This research was supported by the Center for Research in Applied Signal Processing.

It has long been recognized that it is better to exploit the crosstalk than to treat it as background noise. In this paper we consider receivers consisting of a MIMO linear or decision-feedback equalizer followed by a vector decision device. Although a scalar equalizer mitigates intersymbol interference only (ISI), it should be emphasized that a MIMO equalizer mitigates both ISI and CCI.

Several prior works have developed detection strategies for MIMO channels when the channel responses are known [1–7], or for when a training signal is available [8]. But when the channel responses are unknown or time varying, and the receiver does not have access to a training signal, the receiver must be adapted according to a blind equalization algorithm. We propose several blind equalization algorithms for MIMO channels and compare their performance to a recent algorithm proposed by Oda and Sato [9].

In section II we review the vector LMS algorithm, a MIMO version of the least mean-squared (LMS) algorithm, and we discuss its convergence. In section III we propose several blind equalization algorithms for MIMO channels. In section IV we present some numerical results.

II. The Vector LMS Algorithm

A blind MIMO equalizer should be switched to a decision-directed mode once the ISI and CCI are mitigated to a sufficient degree that decisions based on the equalizer output are reliable. Under the assumption that all decisions are correct, the decision-directed equalizer is equivalent to an adaptive equalizer with training. In this section we review a MIMO version of the least-mean square (LMS) algorithm for adaptation with training.

A. Derivation of Vector LMS

The vector LMS algorithm can be derived following the approach of [10]. Consider the MIMO linear equalizer shown in Fig. 2-b below; it is a transversal filter with L matrix-valued taps described by the matrix $\boldsymbol{C}_k^T = [\boldsymbol{C}_{0,k}\ \boldsymbol{C}_{1,k}\ \ldots\ \boldsymbol{C}_{L-1,k}]$. Let $\boldsymbol{R}_k^T = [\boldsymbol{r}_k^T\ \boldsymbol{r}_{k-1}^T \ldots \boldsymbol{r}_{k-L+1}^T]$ be a vector of equalizer inputs, so that the filter output at time k is $\boldsymbol{y}_k = \boldsymbol{C}_k^T\boldsymbol{R}_k$.

The goal of the filter is to minimize the mean-squared error (MSE) cost function, defined by:

$$J = \mathrm{E}[\|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2], \tag{1}$$

where $\boldsymbol{x}_k$ is the desired signal. The complex gradient [11] of (1) with respect to $\boldsymbol{C}_k$ is:

$$\nabla J = 2\mathrm{E}[\boldsymbol{R}_k^{*}\boldsymbol{e}_k^T], \tag{2}$$

IEEE Global Telecommunications Conference, Singapore, vol. 1, pp. 157-162, November 1995.

where $\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{x}_k$ is the error between the filter output and the desired signal. Letting $2\boldsymbol{R}_k^* \boldsymbol{e}_k^T$ be a stochastic approximation of this gradient in the steepest descent algorithm leads to the vector LMS algorithm:

$$\boldsymbol{C}_{k+1} = \boldsymbol{C}_k - 0.5\,\mu \hat{\nabla}_k \tag{3}$$

$$= \boldsymbol{C}_k - \mu \boldsymbol{R}_k^* \boldsymbol{e}_k^T, \tag{4}$$

where μ is the step size. Note that, when $n = 1$, (4) reduces to the familiar scalar LMS algorithm [10].

Alternatively, substituting $\boldsymbol{e}_k = \boldsymbol{C}_k^T \boldsymbol{R}_k - \boldsymbol{x}_k$ into (2), the gradient can be rewritten as:

$$\nabla J = 2(\Phi \boldsymbol{C}_k - \boldsymbol{P}), \tag{5}$$

where $\Phi = \mathrm{E}[\boldsymbol{R}_k^* \boldsymbol{R}_k^T]$ and $\boldsymbol{P} = \mathrm{E}[\boldsymbol{R}_k^* \boldsymbol{x}_k^T]$. It follows that the optimal weights are given by:

$$\boldsymbol{C}_{opt} = \Phi^{-1} \boldsymbol{P}, \tag{6}$$

and the minimum mean-squared error is:

$$\xi_{min} = \mathrm{E}[\|\boldsymbol{x}_k\|^2] - \mathrm{tr}(\boldsymbol{P}^\dagger \boldsymbol{C}_{opt}), \tag{7}$$

where $(\cdot)^\dagger$ denotes conjugate transpose.

B. Convergence of Vector LMS

Because the vector LMS uses a time average to estimate an ensemble average, the coefficient trajectories are random in nature. A key performance measure is the rate of convergence of ξ_k, the average MSE, which can be expressed as:

$$\xi_k \approx \xi_{\min} + \zeta_k, \tag{8}$$

where $\zeta_k = \mathrm{tr}(\mathrm{E}[(\boldsymbol{C}_k - \boldsymbol{C}_{\mathrm{opt}})^\dagger \Phi (\boldsymbol{C}_k - \boldsymbol{C}_{\mathrm{opt}})])$ is the excess MSE at time k. Following the approach of [12], and assuming that $\Phi = \lambda \boldsymbol{I}$ is diagonal, it can be shown that the excess MSE obeys:

$$\zeta_{k+1} \approx \beta \zeta_k + nL\mu^2 \lambda^2 \xi_{\min}, \tag{9}$$

where $\beta = [1 - 2\,\mu\lambda + nL\mu^2\lambda^2]$. Differentiating β with respect to μ, we find the step size that maximizes the rate of decrease of the MSE is approximately:

$$\mu_{\mathrm{opt}} \approx \frac{1}{nL\lambda}. \tag{10}$$

This result agrees with the scalar theory when $n = 1$ [12]. As in the scalar case, the MSE converges to $2\xi_{min}$ asymptotically with $\mu = \mu_{\mathrm{opt}}$, and the step size must be less than $2\mu_{\mathrm{opt}}$ to ensure convergence. The choice of step size is a trade off between speed of convergence and excess asymptotic MSE.

C. Example of Vector LMS

It is important to emphasize that the vector LMS algorithm is not equivalent to n independent scalar LMS algorithms operating in parallel, because the vector LMS algorithm exploits the crosstalk between the different users, whereas a bank of scalar LMS algorithms effectively treats the interference as noise. To illustrate this point, consider the AWGN model of Fig. 1 with a channel transfer function of:

$$\boldsymbol{H}(z) = \boldsymbol{H}_0 + \boldsymbol{H}_1 z^{-1} = \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 0.7 & 0.4 \\ 0.1 & 0.65 \end{bmatrix} z^{-1}. \tag{11}$$

Assume the two users are independent BPSK transmitters. We consider three different adaptive filters to process the received signal $\boldsymbol{r}_k$, as shown in Fig. 2. The first (Fig. 2-a) is a conventional receiver consisting of a bank of two linear equalizers adapting independently according to the LMS algorithm. The second receiver (Fig. 2-b) is a MIMO linear equalizer (LE) employing the vector LMS algorithm. The third receiver (Fig. 2-c) is a MIMO decision-feedback equalizer (DFE) employing the vector LMS algorithm. In all cases we assume a training sequence is available, so that the receiver has knowledge of $\boldsymbol{x}_k$ for generating the error and for the feedback filter. In practice, after the training period has ended, the receiver uses the receiver decisions $\hat{\boldsymbol{x}}_k$ for these purposes.

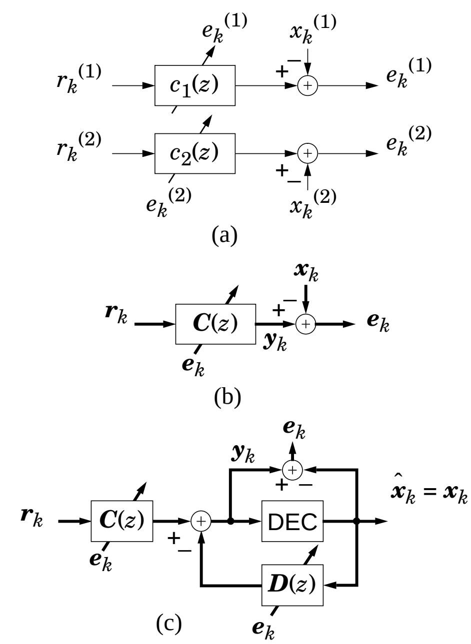

Fig. 2. Three adaptive receivers: (a) conventional receiver; (b) LE receiver; (c) DFE receiver.

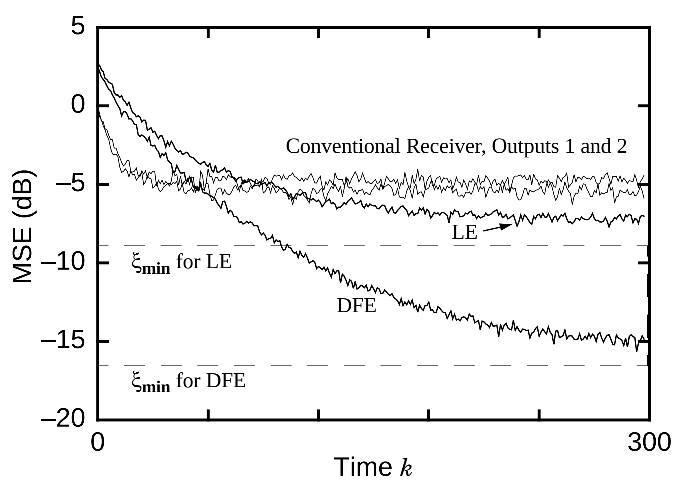

Fig. 3. Learning curves for the three adaptive receivers, averaged over 400 runs.

In Fig. 3 we show simulation results for each of the systems of Fig. 2, assuming a noise PSD of $N_0 = 0.01$. The step size was $\mu = 0.0595$ for the scalar equalizers of Fig. 2-a. The step size for the vector LMS was chosen to be $\mu_{opt}/2$ using (10), where λ is the average eigenvalue of Φ, or $\mu_{LE} = 0.0298$ for the LE and $\mu_{DF} = 0.0356$ for the DFE. The scalar equalizers in Fig. 2-a had five taps each, the LE in Fig. 2-b had five matrix taps, and the DFE in Fig. 2-c had three forward taps and two feedback taps.

From Fig. 3, we see that the conventional receiver performs poorly; the MSE for each user is –5 dB, resulting in a total MSE of –2 dB. In contrast, both the LE and DFE receivers are effective at suppressing the crosstalk, with the DFE outperforming the LE by 8 dB on this channel.

III. Blind Equalization For MIMO Channels

The vector LMS algorithm described above requires a reference signal (or a training signal) for acceptable performance, and hence it is unsuitable for blind equalization. In the following four subsections we propose four blind algorithms for adaptively suppressing both ISI and CCI, all of which can be viewed as extensions of the scalar constant-modulus algorithm (CMA) [13] to MIMO channels.

A. Pointwise CMA

The first algorithm is based on the observation that each component of the channel input will have a constant modulus for certain modulation schemes (such as PSK and 4-QAM), and nearly a constant modulus for other schemes. At perfect equalization, the equalizer output should equal the channel input. This suggests a cost function consisting of a sum of scalar cost functions:

$$J_p = \mathrm{E}\left[\sum_{i=1}^{n} \left(|y_k^{(i)}|^2 - M_i \right)^2 \right] = \mathrm{E}[\| \boldsymbol{y}_k \otimes \boldsymbol{y}_k^* - \boldsymbol{M} \|^2], \tag{12}$$

where $\boldsymbol{M} = [M_1 \dots M_n]^T$ and M_i is the modulus of the i-th transmitted symbol, and where the symbol $\otimes$ indicates a Schur (element-by-element) product. The complex gradient of this cost function with respect to $\boldsymbol{C}_k$ is:

$$\nabla J_p = 4\mathrm{E}[\boldsymbol{R}_k^* \boldsymbol{e}_k^T], \tag{13}$$

where the error signal is defined by:

$$\boldsymbol{e}_k = \boldsymbol{y}_k \otimes (\boldsymbol{y}_k \otimes \boldsymbol{y}_k^* - \boldsymbol{M}). \tag{14}$$

If we let $4\boldsymbol{R}_k^* \boldsymbol{e}_k^T$ be a stochastic approximation to this gradient in the steepest descent algorithm, we arrive at same update equation (4) as the vector LMS, but with $\boldsymbol{e}_k$ given by (14). We refer to this algorithm as the *pointwise* CMA, which reduces to the scalar CMA when $n = 1$. In fact, the pointwise CMA is equivalent to a bank of independent multichannel blind equalizers, with each multichannel receiver using the multichannel CMA proposed in [14]; in this configuration, only the modulus of $y_k^{(i)}$ is used to adapt the n scalar filters contributing to $y_k^{(i)}$, irrespective of the modulus of the other components.

The modulus parameters M_i should be chosen so that the gradient in (13) is equal to the zero matrix at perfect equalization, which gives $M_i = \mathrm{E}[|x_k^{(i)}|^4] / \mathrm{E}[|x_k^{(i)}|^2]$. With this choice, adaptation will cease on average after perfect equalization is achieved.

Let $\boldsymbol{F}(z) = \boldsymbol{C}(z)\boldsymbol{H}(z)$ be the overall system transfer function from the channel input to equalizer output. The primary drawback of the pointwise CMA is its susceptibility to the one-to-many problem, whereby more than one component of the equalizer output locks on to the same equalizer input. For example, when $n = 2$, the pointwise CMA cost function is minimized by an overall transfer function of:

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}. \tag{15}$$

This solution is unacceptable because the first channel input is mapped to both equalizer outputs, and the second channel input is ignored.

It can be shown that, under the assumption of an infinite-tap equalizer, the pointwise CMA cost function achieves a local minimum if and only if the overall transfer function has the form:

$$\boldsymbol{F}(z) = \begin{bmatrix} 0 \dots 0 & \pm z^{-N_1} & 0 & \dots & 0 \\ 0 & \dots & 0 & \pm z^{-N_2} & 0 \dots 0 \\ & & \vdots & & \\ 0 & \dots & 0 & \pm z^{-N_n} & 0 \quad \dots \quad 0 \end{bmatrix}, \tag{16}$$

where each row has only one nonzero entry. In other words, the cost function is minimized if and only if each component of the equalizer output is equal to a delayed and possibly sign-changed version of any component of the channel input. Observe that the unacceptable solutions are those for which $\boldsymbol{F}(z)$ is singular, since in this case one channel input is mapped to more than one equalizer output, while another channel input is dropped altogether.

Comparing (12) and (16), we see that all local minima of the pointwise CMA cost function are global minima.

B. Vector CMA

We now propose a second blind algorithm based on the observation that the norm of the input vector $\| \boldsymbol{x}_k \|$ is constant or nearly so. For example, $\| \boldsymbol{x}_k \|^2 = n$ when each of n users selects data from a BPSK constellation of $\{\pm 1\}$. At perfect equalization, the equalizer output will have the same norm as the channel input. This suggests a cost function of:

$$J_v = \mathrm{E}[(\| \boldsymbol{y}_k \|^2 - M)^2], \tag{17}$$

where M is a constant. The gradient of this cost function is again given by (13), where this time the error signal is:

$$\boldsymbol{e}_k = \boldsymbol{y}_k (\| \boldsymbol{y}_k \|^2 - M). \tag{18}$$

We refer to this algorithm as the *vector* CMA. The best modulus parameter M forces the gradient (13) to zero at perfect equalization, i.e., $M = \mathrm{E}[\| \boldsymbol{x}_k \|^4] / \mathrm{E}[\| \boldsymbol{x}_k \|^2]$. Like the pointwise CMA, the vector CMA reduces to the scalar CMA when $n = 1$. The vector CMA, pointwise CMA, and vector LMS algorithm are all described by the same update equation (4), the only difference being the definition of $\boldsymbol{e}_k$.

A primary drawback of the vector CMA cost function is that it is minimized by any unitary matrix $\boldsymbol{F}(z)$, such as:

$$\boldsymbol{F} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \qquad (19)$$

In addition, for the special case when each user employs the same constellation and this constellation has a constant modulus, the transfer functions described by (16) also minimize the vector CMA cost function. Thus, for constant-modulus signals, the vector CMA is also susceptible to the one-to-many problem.

When all users employ the same non-constant modulus constellation, the vector CMA cost function is minimized by some of the singular solutions of (16) but not others. In particular, the cost function is not minimized by any of the singular solutions that are memoryless, i.e., for which $N_1 = \ldots = N_n$. On the other hand, it is minimized by some singular solutions with memory, such as:

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ z^{-1} & 0 \end{bmatrix}. \qquad (20)$$

C. Combination of Pointwise and Vector CMA

Recognizing the drawbacks of both the pointwise and vector CMA cost functions, Oda and Sato proposed a linear combination that incorporates some of the better qualities of each [9]:

$$J_c = A\mathrm{E}[(\| \boldsymbol{y}_k \|^2 - M)^2] + B\mathrm{E}[\| \boldsymbol{y}_k \otimes \boldsymbol{y}_k^* - \boldsymbol{M} \|^2], \qquad (21)$$

where A and B are constants. We refer to the resulting algorithm as the *combination* CMA. As shown in section IV, the combination CMA can outperform both the pointwise CMA and vector CMA. Unfortunately, when the users employ the same constant-modulus constellation, all solutions described by (16) minimize (21), making the combination CMA susceptible to the one-to-many problem. Furthermore, even when the constellations do not have constant modulus, (21) is still minimized by solutions of the form (20).

D. Determinant CMA

We now propose the determinant CMA (DCMA), a blind equalization algorithm for memoryless channels having CCI but no ISI. In this case, the solutions which globally minimize the pointwise CMA cost function when $n = 2$ are:

$$\boldsymbol{F} = \begin{bmatrix} \pm1 & 0 \\ 0 & \pm1 \end{bmatrix}, \begin{bmatrix} 0 & \pm1 \\ \pm1 & 0 \end{bmatrix}, \text{ and } \begin{bmatrix} \pm1 & 0 \\ \pm1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & \pm1 \\ 0 & \pm1 \end{bmatrix}, \qquad (22)$$

where $\boldsymbol{F} = \boldsymbol{CH}$ is the overall transfer function, $\boldsymbol{C}$ is the equalizer, and $\boldsymbol{H}$ is the channel. The first two matrices describe acceptable solutions, because the data for both users can be recovered by changing the sign and reordering the signals as necessary. In contrast, the second two matrices are undesirable because they completely destroy the information from one of the users. Observe that the undesirable solutions have zero determinant, whereas the desirable solutions have determinant ±1. This suggests that we modify the pointwise cost function by penalizing those solutions for which the magnitude of the determinant is not unity:

$$J_d = AJ_p + B(|\det\boldsymbol{F}|^2 - 1)^2, \qquad (23)$$

where J_p is the pointwise cost function of (12), and A and B are constants. The gradient of J_d is:

$$\nabla J_d = 4A\mathrm{E}[\boldsymbol{R}_k^* \boldsymbol{e}_k^T] + 4B(\gamma^2 - 1)\gamma^2(\boldsymbol{C}^\dagger{}_k)^{-1}, \qquad (24)$$

where $\boldsymbol{e}_k$ is the pointwise CMA error signal of (14), and where $\gamma^2 = |\det\boldsymbol{F}|^2$. This gradient leads to the following update equation, which we call the determinant CMA (DCMA):

$$\boldsymbol{C}_{k+1} = \boldsymbol{C}_k - \mu A \boldsymbol{R}_k^* \boldsymbol{e}_k^T - B\mu(\gamma^2 - 1)\gamma^2(\boldsymbol{C}_k^\dagger)^{-1}. \qquad (25)$$

If $\det\boldsymbol{H} = 0$ then $\gamma = 0$, and the last correction term in (25) will be zero, so that the DCMA reverts back to the original pointwise CMA. This is not surprising in light of (23), which shows that $J_d = AJ_p + B$ when $\det\boldsymbol{H} = 0$, and hence the DCMA and pointwise cost functions are equivalent on singular channels.

As stated in (25), the DCMA requires knowledge of $\gamma^2 = |\det\boldsymbol{F}|^2$ at the receiver. This is not problematic, because $|\det\boldsymbol{F}|^2$ is easily estimated. Without noise, the equalizer output is given by $\boldsymbol{y}_k = \boldsymbol{F}\boldsymbol{x}_k$, so that the autocorrelation matrix for $\boldsymbol{y}_k$ is:

$$\boldsymbol{P} = \mathrm{E}[\boldsymbol{y}_k \boldsymbol{y}_k^\dagger] = \sigma^2 \boldsymbol{F}\boldsymbol{F}^\dagger, \qquad (26)$$

where we have used the assumption that $\mathrm{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \sigma^2 \boldsymbol{I}$. It follows that $\gamma^2 = \det(\boldsymbol{P}/\sigma^2)$. Estimating $\boldsymbol{P}$ by time averaging results in the following simple estimate for $|\det\boldsymbol{F}|^2$ at time k:

$$\gamma_k^2 = \det\left(\frac{1}{k\sigma^2} \sum_{i=1}^{k} \boldsymbol{y}_i \boldsymbol{y}_i^\dagger\right). \qquad (27)$$

In a practical implementation, γ_k^2 should be substituted for γ^2 in (25). Simulation results indicate that the estimate in (27) converges to $|\det\boldsymbol{F}|^2$ fairly quickly, within 20-40 iterations on certain channels, so the time wasted while the receiver learns $|\det\boldsymbol{F}|$ will be negligible. The primary drawback of the DCMA is the increased complexity associated with estimating $|\det\boldsymbol{F}|$ and computing the matrix inverse.

IV. Numerical Results

We now present simulation results from four experiments. The DCMA is defined only for the memoryless channels of experiments 2 and 3, and it does not fail in either case. In contrast, we demonstrate failure for each of the other algorithms. In the first experiment, only the vector CMA fails; in the second experiment, both the vector and the pointwise CMA fail; and in the third and fourth experiments, the vector, pointwise, and combination CMA all fail.

A. Experiment 1: Channel with Memory

Consider a noiseless two-user system with the following channel matrix:

$$\boldsymbol{H}(z) = \begin{bmatrix} 1 & 0.7 \\ -0.3 & 0.9 \end{bmatrix} + \begin{bmatrix} 0.3 & 0 \\ 0.4 & 0.1 \end{bmatrix} z^{-1} . \tag{28}$$

Both users employ a 16-QAM constellation independently and uniformly. Assume the equalizer has seven taps, with the center tap initialized to the identity matrix, and assume a step size of 5×10^{-6}. The modulus parameters for two-user 16-QAM are $M_i = 13.2$ and $M = 23.2$.

In Fig. 4 we show simulation results for the vector CMA, pointwise CMA, and combination CMA. Both the pointwise and combination CMA successfully opened the eye and converged to an overall transfer function close to the identity matrix. The vector CMA, on the other hand, did not open the eye, but converged to the following transfer function (averaged from time 23000 to 25000):

$$\boldsymbol{F} = \begin{bmatrix} 0.89 & 0.47 \\ -0.48 & 0.87 \end{bmatrix} , \tag{29}$$

which is close to (19) with $\theta = -28^\circ$. The constellation for the first component of the vector CMA and pointwise CMA equalizer outputs are shown in Fig. 5, from time 23000 to time 25000. The constellation for the second components are nearly identical. The constellations for the combination CMA are similar to Fig. 5-b.

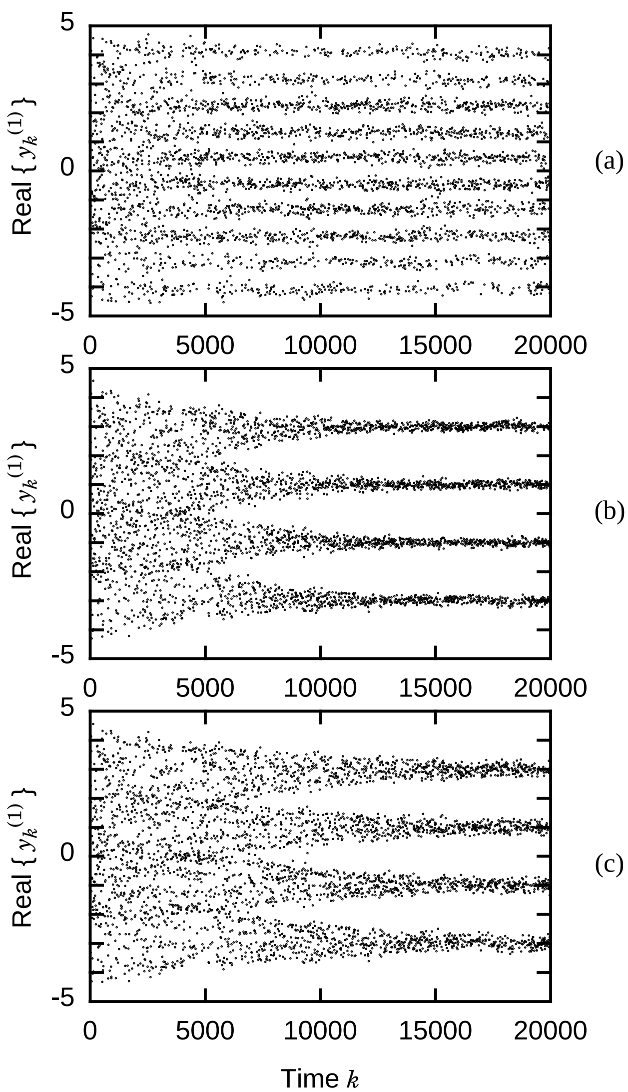

Fig. 4. Real part of first component of equalizer output for (a) vector CMA; (b) pointwise CMA; and (c) combination CMA.

B. Experiment 2: Memoryless Channel with 16-QAM

Consider the following memoryless channel:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 1 & 0.5 \end{bmatrix} . \tag{30}$$

Although there is no crosstalk for the first user, the crosstalk for the second user is particularly severe, with the interference power exceeding the signal power by 6 dB. This is a difficult channel to equalize, because the channel is close to the undesirable overall solution of (15).

As before, assume two independent 16-QAM transmitters. The equalizer, consisting of a single tap, was initialized to the identity matrix, and the step size $\mu = 4 \times 10^{-5}$. The simulation results are shown in Fig. 6, where we plot $|F_{ij}|$ (the magnitude of the (i,j)-th component of $\boldsymbol{F}$) versus time. We see that the pointwise CMA fails, with both equalizer outputs locking onto the first channel input, as in (15). In contrast, the combination and determinant CMA both converge to the identity matrix.

C. Experiment 3: Memoryless Channel with 4-QAM

We repeat experiment 2 with the memoryless channel of (30), but this time we assume the constellation for each user is 4-QAM. This constellation has a constant modulus, which makes the combination CMA susceptible to the one-to-many problem. Indeed, after convergence, the asymptotic overall transfer functions for the pointwise, combination, and determinant CMA were respectively:

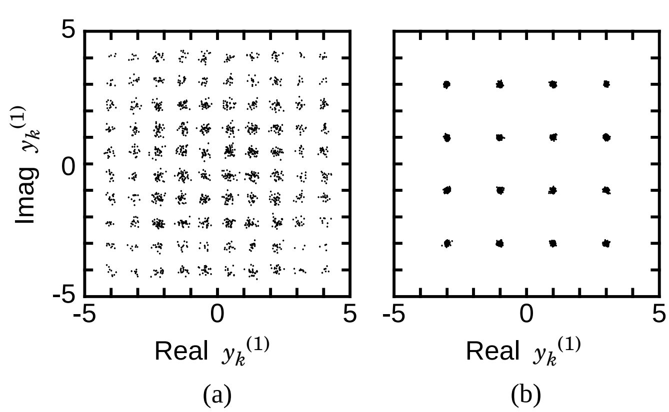

Fig. 5. User 1 constellation: (a) vector CMA; (b) pointwise CMA.

$$\boldsymbol{F}_p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \boldsymbol{F}_c = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \boldsymbol{F}_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (31)$$

Only the DCMA converged to the correct solution.

D. Experiment 4: Nearly Singular Channel with Memory

Consider the following channel:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ z^{-1} & 0.5 \end{bmatrix}. \quad (32)$$

The input constellations were BPSK {±1}. The equalizer had five taps, the center one initialized to the identity matrix. The step size was switched from 10^{-2} to 10^{-3} at time 5000. Both the pointwise and combination CMA converged to the undesirable solution of (20).

V. Conclusion

We have proposed several blind algorithms for adaptive suppression of co-channel interference and intersymbol interference. The proposed algorithms are straightforward extensions of the conventional CMA to vector-valued signals. To prevent multiple equalizer outputs from locking on to the same user signal, we proposed the determinant CMA for memoryless channels. Simulation results show that the DCMA can outperform the other algorithms considered on certain channels. Future work should extend this algorithm to channels with memory.

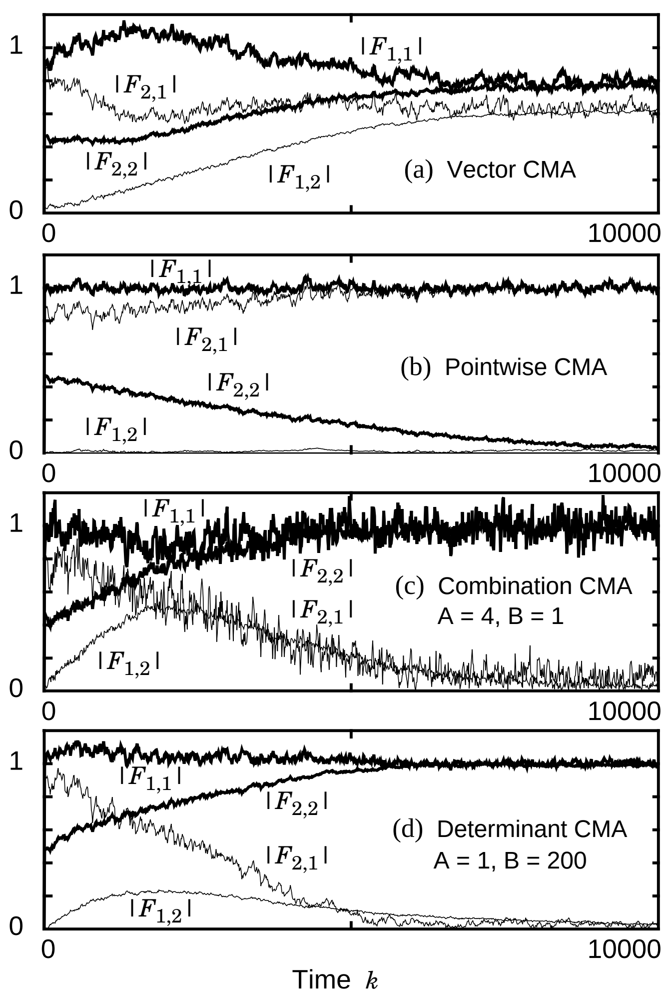

Fig. 6. Simulation results for experiment two.

VI. Acknowledgment

The authors are grateful to Michael Minardi for his help in deriving the local minima for pointwise CMA and for his useful comments.

VII. References

[1] W. van Etten, "An Optimum Linear Receiver for Multiple Channel Digital Transmission Systems," *IEEE Trans. Comm.*, pp. 828-834, Aug. 1975.

[2] J. Salz, "Digital Transmission Over Cross-Coupled Linear Channels," *AT&T Techn. J.*, 64:6, pp. 1147, July-Aug. 1985.

[3] A. Duel-Hallen, "Equalizers for Multiple-Input/Multiple-Output Channels and PAM Systems with Cyclostationary Input Sequences," *IEEE J. Select. Areas in Comm.*," 10:3, pp. 630-639, Apr. 1992.

[4] M. L. Honig, P. Crespo, and K. Steiglitz, "Suppression of Near- and Far-End Crosstalk by Linear Pre- and Post-Filter," *IEEE J. Select. Areas in Comm.*, 10:3, pp. 614-629, Apr. 1992.

[5] M. Kavehrad and J. Salz, "Cross-Polarization Cancellation and Equalization in Digital Transmission over Dually Polarized Multipath Fading Channels," *Bell Sys. Techn. J.*, 64:10, pp. 2211-45, Dec. 1985.

[6] P. A. Voois and J. M. Cioffi, "Multichannel Signal Processing for Multiple-Head Digital Magnetic Recording," *IEEE Trans. Magnetics*, 30:6, pp. 5100-5114, Nov. 1994.

[7] S. Verdu, "Recent progress in multi-user detection," in *Advances in Communications and Signal Progressing*, W. A. Porter and S. C. Kak, Eds. Springer-Verlag, New York (1989).

[8] K. D. Senne, "Adaptive Linear Discrete-Time Estimation," Technical Report No. 6778-5, Systems Theory Laboratory, Stanford University, June 1968.

[9] H. Oda, Y. Sato, "A Method of Multi-Dimensional Blind Equalization," *Intern. Symp. on Info. Theory*, San Antonio, TX, p. 327, 1993.

[10] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, New Jersey (1985), pp. 99-103.

[11] S. Haykin, *Adaptive Filter Theory*, Second Edition, Prentice-Hall, New Jersey (1991), p. 162.

[12] E. A. Lee and D. G. Messerschmitt, Digital Communication, Second Edition, Kluwer, Boston (1994), p. 537.

[13] J. R. Treichler, B. G. Agee, "A New Approach to Multipath Correction of Constant Modulus Signals," *IEEE Trans. Acous., Speech, and Sign. Proc.*, 31:2, pp. 459-472, Apr. 1983.

[14] J. R. Treichler, M. G. Larimore, "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm," *IEEE Trans. Acous., Speech, and Sign. Proc.*, 33:2, pp. 420-431, Apr. 1985.